

«ИНФОРКОМ»

ZX-РЕВЮ 1991

Выпуск 01-12

VERSION 1.00

PDF version by NUK, km, Василий Кормилицын, Kendo Anoubis.

Feedback: nuk@mail.ru

ZX-PEBIO № 1'1991 "The Origin"		Completed	
ZX-PEBIO № 2'1991 "To The Stars"		Completed	
ZX-PEBIO № 3'1991 "One Step From Earth"		Completed	
ZX-PEBIO № 4-5'1991 "Generation Z"		Completed	
ZX-PEBIO № 6'1991 "Nuclear Power"		Completed	
ZX-PEBIO № 7-8'1991 "A Human Work"		Completed	
ZX-PEBIO № 9'1991 "The Undead"		Completed	
ZX-PEBIO № 10'1991 "The Murder Machine"		Completed	
ZX-PEBIO № 11-12'1991 "Level Complete"		Completed	
Спектрум в школе	1 ¹	Completed	by NUK
MEGA-BASIC	3	Completed	by NUK
128K	4	Completed	by NUK
Раздел для начинающих	5	Completed	by NUK
MACHINE CODE	6	Completed	by NUK
Секреты ПЗУ	7	Completed	by NUK
Маленькие хитрости	8	Completed	by NUK
Жанры игровых программ.	9	Completed	by NUK
FIGHTER PILOT	11	Completed	by NUK
IS CHESS 48K	12	Completed	by NUK
MOON LANDING	13	Completed	by NUK
Наш конкурс	14	Completed	by NUK
TOP TEN	14	Completed	by NUK
Расширение памяти компьютера "Спектрум"	15	Completed	by NUK
Раздел для начинающих	23	Completed	by NUK
MACHINE CODE	24	Completed	by NUK
MEGA-BASIC	25	Completed	by NUK
128K	26	Completed	by NUK
Секреты ПЗУ	27	Completed	by NUK
Маленькие хитрости	29	Completed	by NUK
FORUM	30	Completed	by NUK
ELITE	31	Completed	by NUK
ADVENTURE LESSONS	35	Completed	by NUK
ZX MODEM	37	Completed	by NUK
Раздел для начинающих	45	Completed	by NUK
MEGA BASIC	47	Completed	by km
128 K	48	Completed	by NUK
Профессиональный подход (Стив Тернер)	49	Completed	by Kendo Anoubis
Маленькие хитрости	51	Completed	by NUK
FORUM	52	Completed	by km
CONQUEST	53	Completed	by km
POKES (Manic Miner)	54	Completed	by km
ADVENTURE LESSONS	55	Completed	by km
CENTI	57	Completed	by NUK
BEST GAMES	58	Completed	by NUK
ZX Модем	59	Completed	by NUK
Раздел для начинающих	67	Completed	by NUK
MEGA BASIC	68	Completed	by NUK
Профессиональный подход (Стив Тернер)	70	Completed	by NUK
Секреты ПЗУ	71	Completed	by NUK
FORUM	80	Completed	by NUK
Маленькие хитрости	82	Completed	by NUK
ADVENTURE LESSONS	83	Completed	by km
G.A.C.	85	Completed	by km
SCRABBLE	87	Completed	by km
FORUM	89	Completed	by km
TOBRUK	91	Completed	by NUK
LEGIONS OF DEATH	91	Completed	by NUK
BATTLEFIELD GERMANY	92	Completed	by NUK
SAMURAI	92	Completed	by NUK
BATTLE OF BRITAIN	93	Completed	by NUK
VULCAN	93	Completed	by NUK
DESERT RATS	93	Completed	by NUK
INVASION	94	Completed	by NUK

¹ Номер страницы в оригинальном издании

Буферизация	95	Completed	by km
Рекламное приложение	103	Completed	by km
Наш конкурс(подборка игровых программ)	105	Completed	by km
Рекламное приложение	107	Completed	by km
Раздел для начинающих	111	Completed	by NUK
MEGA BASIC	112	Completed	by NUK
128 K	114	Completed	by Василий Кормилицын
Forum	115	Completed	by Василий Кормилицын
Маленькие хитрости	116	Completed	by km
Профессиональный подход (Стив Тернер)	117	Completed	by km
В Вашу записную книжку (интерфейсы)	120	Completed	by Василий Кормилицын
Adventure Project	122	Completed	by km
SHERLOCK	125	Completed	by km
MIAMI VICE	128	Completed	by km
PRESIDENT	129	Completed	by NUK
FALKLANDS82	129	Completed	by NUK
AUSTERLIZ	130	Completed	by NUK
Цена идеи	131	Completed	by Василий Кормилицын
Спектрум в школе	133	Completed	by NUK
BETA BASIC	134	Completed	by NUK
Маленькие хитрости	140	Completed	by NUK
Профессиональный подход (Стив Тернер)	143	Completed	by NUK
Вывод на экран и принтер	148	Completed	by NUK
FORUM	155	Completed	by NUK
PSI CHESS	158	Completed	by km
FORUM (ELITE, SABOTEUR-2)	161	Completed	by km
POKES	164	Completed	by NUK
THUNDERBIRDS	166	Completed	by km
HAMMERFIST	168	Completed	by NUK
MYTH	168	Completed	by NUK
KLAX	168	Completed	by NUK
OVERLORD	168	Completed	by km
OVERLORDS	170	Completed	by km
ZX LPRINT III	171	Completed	by NUK
Секреты ПЗУ	177	Completed	by NUK
BETA BASIC	180	Completed	by NUK
Цена идеи	183	Completed	by Василий Кормилицын
Профессиональный подход (Стив Тернер)	187	Completed	by km
EQUINOX	189	Completed	by km
ADVENTURE PROJECT	193	Completed	by Василий Кормилицын
FORUM	195	Completed	by km
Спектрум в школе	199	Completed	by NUK
BETA BASIC	201	Completed	by NUK
Секреты ПЗУ	203	Completed	by NUK
SECAM CODER	207	Completed	by NUK
ZX LPRINT III	210	Completed	by NUK
FORUM (ELITE)	211	Completed	by NUK
LAST NINJA2	216	Completed	by NUK
BATMAN2	216	Completed	by NUK
GUNFRIGHT	218	Completed	by NUK
POKES	219	Completed	by NUK
ADVENTURE GAMES	220	Completed	by NUK
FORUM	222	Completed	by NUK
BETA BASIC	223	Completed	by NUK
КАНАЛЫ И ПОТОКИ	227	Completed	by NUK
MACHINE CODE	240	Completed	by NUK
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	244	Completed	by NUK
ADVENTURE PROJECT	248	Completed	by NUK
FORUM	252	Completed	by km
PIJAMARAMA	255	Completed	by NUK
AMAUROTE	256	Completed	by km
FREDDY HARDEST	257	Completed	by km
SILENT SERVICE	258	Completed	by km
HUNTER KILLER	261	Completed	by km

Содержание

СПЕКТРУМ В ШКОЛЕ	17
ОРГАНИЗАЦИЯ СТРУКТУР ДАННЫХ В УЧЕБНЫХ ПРОГРАММАХ	17
MEGA-BASIC	20
ОСНОВНЫЕ ХАРАКТЕРИСТИКИ.....	20
1. КЛАВИАТУРА.....	20
РЕДАКТИРОВАНИЕ	21
128K.....	23
РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ.....	25
MACHINE CODE	28
1. ПРОГРАММА КОМПРЕССИИ	28
СЕКРЕТЫ ПЗУ	31
1. ПРОЦЕДУРЫ ИНИЦИАЛИЗАЦИИ	32
МАЛЕНЬКИЕ ХИТРОСТИ	35
ЖАНРЫ ИГРОВЫХ ПРОГРАММ.....	37
ADVENTURE	37
ARCADE	38
ARCADE/ADVENTURE	38
ACTION	39
TRADITIONALS.....	39
SIMULATIONS.....	39
STRATEGY	39
BUSINESS/MANAGEMENT	40
PUZZLE	40
FIGHTER PILOT	41
РЕЖИМЫ ПРОГРАММЫ.	41
УПРАВЛЕНИЕ САМОЛЕТОМ.....	42
БОРТОВЫЕ ПРИБОРЫ.	43
СВОДКА КОМАНД	44
ПОЛЕТНЫЕ ДАННЫЕ.....	44
IS CHESS 48K.....	45
MOON LANDING	47
НАШ КОНКУРС.....	51
TOP TEN	51
РАСШИРЕНИЕ ПАМЯТИ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА "СПЕКТРУМ"	52
РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ.....	64
БЕЙСИК ШАГ ЗА ШАГОМ	64
MACHINE CODE	67
2. ПРОГРАММА ДЕКОМПРЕССИИ	67
MEGA-BASIC	69
<i>Клавиши, определяемые пользователем (UDK)</i>	<i>69</i>
<i>Клавиши управления.....</i>	<i>69</i>
2. ВЫВОД НА ЭКРАН	69
Окна	69
Команды CLS и CLW_.....	70
Команды PAN и SCROLL	70
Команда MODE_.....	70
128K.....	71

ВПЕЧАТЫВАНИЕ СТРАНИЦ.	71
ЭКРАННАЯ ПАМЯТЬ	72
ПЕРЕКЛЮЧАТЕЛЬ РЕЖИМОВ	72
СЕКРЕТЫ ПЗУ	74
2. ОБСЛУЖИВАНИЕ КЛАВИАТУРЫ.	74
3. УПРАВЛЕНИЕ ЗВУКОВЫМ ДИНАМИКОМ	77
Подпрограмма <i>BEEPER</i>	77
Подпрограмма <i>BEEP (03F8-046B)</i>	78
ЛИТЕРАТУРА	78
МАЛЕНЬКИЕ ХИТРОСТИ	79
FORUM	81
ОБЗОР ФЕВРАЛЬСКОЙ ПОЧТЫ	81
ИГРОВОЕ ПРИЛОЖЕНИЕ	83
ELITE.....	83
1. ВСТУПЛЕНИЕ.	83
2. НАЧАЛО РАБОТЫ.	84
3. РАБОТА С КАССЕТОЙ.	84
4. СТРАНИЦА СОСТОЯНИЯ ИГРЫ.	84
5. БОЕВОЙ РЕЙТИНГ.	85
6. НАВИГАЦИЯ.	85
7. УПРАВЛЕНИЕ НАВИГАЦИЕЙ.	85
8. ОРБИТАЛЬНАЯ СТАНЦИЯ "КОРИОЛИС".	87
9. ОТХОД ОТ ОРБИТАЛЬНОЙ СТАНЦИИ.	87
10. ПРИБОРНАЯ ПАНЕЛЬ.	87
11. ПЕРЕЛЕТ.	88
12. МЕЖГАЛАКТИЧЕСКИЙ ПЕРЕЛЕТ.	89
13. ТЕХНИКА СТЫКОВКИ С ОРБИТАЛЬНОЙ СТАНЦИЕЙ.	89
14. БОЕВЫЕ ДЕЙСТВИЯ.	89
15. ВОЗМОЖНЫЕ ПРОТИВНИКИ.	90
Охотники за призами.	90
Пираты.	90
Полиция.....	90
Корабли Таргона и "Таргоны".	90
16. Другие космические объекты.....	91
Отшельники.	91
Корабли поколений.	91
Космические платформы.	91
17. СНАРЯЖЕНИЕ И ВООРУЖЕНИЕ.....	91
Лазеры.	92
Ракеты.	92
Энергетические бомбы.	92
18. Защита.....	92
19. СНАРЯЖЕНИЕ.	93
20. ТОРГОВЛЯ.	93
21. ПОЛЕЗНЫЕ СОВЕТЫ ПО ТОРГОВЛЕ.	94
22. ДРУГИЕ СПОСОБЫ ОБОГАЩЕНИЯ.	94
Охота за призами.....	94
Пиратство.....	94
Разработка астероидов.	94
Поиск грузовых контейнеров.....	94
ADVENTURE LESSONS	95
УРОК 1. ОБЩИЙ АНАЛИЗ	95
УРОК 2. ПЕРЕМЕЩЕНИЕ В ИГРОВОМ ПРОСТРАНСТВЕ.....	97
ТЕХНИЧЕСКОЕ ПРИЛОЖЕНИЕ	99
ZX MODEM	99
Формат передачи.	101
Программная поддержка.	101
Проверка работы	101
РЕКЛАМНОЕ ПРИЛОЖЕНИЕ	105

ПЛЮС	107
РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ.....	109
БЕЙСИК ШАГ ЗА ШАГОМ	109
Работа с символьным набором.	109
MEGABASIC.....	115
Команда FONT_.....	115
Команды CHR\$, VDU, DOWN.	115
Команды SPRINT и PRINTER.....	115
ГРАФИКА.....	115
Команды CHANGE и SWAP.	115
Команда FADE_.....	116
Команда INVERT.....	116
Команда DEFG_.....	116
Команды GET_ и PUT_.....	116
Команда SPUT_.....	116
128 К	118
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	121
ЧАСТЬ 1.	121
ЦВЕТОВЫЕ АТТРИБУТЫ	121
1. Избегайте фона.....	121
2. Только два цвета	121
3. Много PAPER и мало INK	122
ЗВУК.....	122
СКОРОСТЬ РАБОТЫ ПРОЦЕССОРА	122
УПРАВЛЕНИЕ ИГРОЙ	122
ПАМЯТЬ.....	123
МАЛЕНЬКИЕ ХИТРОСТИ	125
Числа.....	125
Несколько приемов.....	126
FORUM	128
ИГРОВОЕ ПРИЛОЖЕНИЕ	130
CONQUEST	130
Введение.....	130
Правила уровня 1.....	130
Уровень 2.....	131
Уровень 3.....	132
Уровень 4.....	132
Уровень 5.....	132
Правила усложненной игры.....	132
КАК САМОСТОЯТЕЛЬНО ВНЕСТИ ИЗМЕНЕНИЯ В ПРОГРАММУ "MANIC MINER"	134
I. Порядок внесения изменений в программу.....	134
II. Дополнительные возможности.	134
III. Выбор экрана с клавиатуры.	135
ADVENTURE LESSONS	136
УРОК 3	136
Осмотр и анализ обстановки.....	136
УРОК 4. РАБОТА С ОБЪЕКТАМИ.....	136
CENTI.....	140
BEST GAMES	142
FULL PRICE TOP 30	142
BUDGIES TOP 30	142
FULL PRICE 20	143
BUDGET 10.....	144
ZX-REVIEW TOP TEN	144
ZX MODEM	145

РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ.....	152
МЕГА-БЕЙСИК	155
УПРАВЛЕНИЕ ВЫПОЛНЕНИЕМ ПРОГРАММЫ	155
Процедуры.	155
REPEAT-UNTIL.....	155
Процедурный стек	155
ОТЛАДКА И РЕДАКТИРОВАНИЕ ПРОГРАММ.....	156
Операторы <i>BRANCH_</i> , <i>TRON_</i> , <i>TROFF_</i> и <i>SPEED</i>	156
Команды <i>AUTO</i> и <i>DELETE</i>	156
Команды <i>BRON_</i> и <i>BROFF_</i>	156
Звук.	156
<i>PLAY_n,l,s,d,f</i>	157
<i>ISG</i>	157
МАШИННЫЕ КОДЫ И МЕГА-БЕЙСИК	157
Команды <i>DOKE</i> и <i>CALL</i>	157
<i>FRONT-PANEL</i>	158
НОВЫЕ КОМАНДЫ.	158
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	160
ОРУДИЯ ПРОИЗВОДСТВА.	160
1. Редактор/Ассемблер.....	160
2. Монитор.....	161
3. Внешняя память.	161
4. Принтер	161
5. Книги по Z-80.	162
СЕКРЕТЫ ПЗУ	164
ПРОГРАММЫ, ОБСЛУЖИВАЮЩИЕ РАБОТУ С МАГНИТОФОНОМ.	164
FORUM	181
МАЛЕНЬКИЕ ХИТРОСТИ	187
ADVENTURE LESSONS	188
9. Книги, записки, объявления и пр.	188
10. Музыкальные инструменты.....	188
11. Растения.	188
12. Веревка, канат (<i>ROPE</i>).....	188
13. Лестница - <i>LADDER</i>	188
5. ВЗАИМОДЕЙСТВИЕ С ДРУГИМИ ПЕРСОНАЖАМИ.	189
6. ИСПОЛЬЗОВАНИЕ ОТГРУЗКИ ПРОГРАММЫ НА ЛЕНТУ.	191
7. ПОЛЕЗНЫЕ ПРАКТИЧЕСКИЕ ПРИЕМЫ.	191
ЗАКЛЮЧЕНИЕ.....	192
GRAPHIC ADVENTURE CREATOR	193
ЛОГИКА.	194
ГРАФИКА.	195
SCRABBLE	196
ВСТУПЛЕНИЕ.....	196
I. ПРАВИЛА ИГРЫ.	196
Игра.	196
Счет.....	197
II. ИГРА С КОМПЬЮТЕРОМ.	198
Главное меню.	198
Как делать ход.	199
Ход пустой фишкой.	199
Запрещенные ходы.	199
Оспаривание хода.	199
Окончание игры.	199
Откладывание игры.....	199
Загрузка отложенной программы.....	199
Стратегия.	199

Словарь	200
FORUM	201
STRATEGIC GAMES	206
TOBRUK	206
LEGIONS OF DEATH	207
BATTLEFIELD GERMANY	209
SAMURAI	210
BATTLE OF BRITAIN	211
VULCAN	212
DESERT RATS	213
INVASION	214
<i>Ваши боевые части</i>	<i>214</i>
1. Радары	214
2. Минные части	214
3. Воздушная пехота	214
4. Парящая броня	214
5. Страйдеры	214
<i>Боевые части инопланетян</i>	<i>215</i>
1. Тяжеловооруженные части	215
2. Части быстрого реагирования	215
3. Дроиды	215
<i>Игра</i>	<i>215</i>
<i>Дополнительные возможности</i>	<i>215</i>
<i>Полезные советы</i>	<i>216</i>
<i>Счет</i>	<i>216</i>
<i>Управление игрой</i>	<i>216</i>
<i>Расписание клавиатуры</i>	<i>216</i>
ТЕХНИЧЕСКОЕ ПРИЛОЖЕНИЕ	217
БУФЕРИЗАЦИЯ	217
НАШ КОНКУРС	228
ЗВЕЗДНЫЕ ВОЙНЫ	229
ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 1	229
ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 2	230
ВОЗДУШНЫЕ СРАЖЕНИЯ	230
СПОРТИВНЫЕ ИГРЫ	230
КОСМИЧЕСКИЕ ВОЙНЫ	231
БЕГИ И СТРЕЛЯЙ ВО ВСЕ, ЧТО ДВИЖЕТСЯ	231
ЛАБИРИНТ-1	232
ПИЛОТ БУДУЩЕГО	232
NO NAME-1	233
ЛАБИРИНТ-2	233
РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ	240
MEGA BASIC	242
Новые сообщения об ошибках	243
128 К	248
Страница ОЗУ	248
FORUM	251
МАЛЕНЬКИЕ ХИТРОСТИ	252
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	255
РАЗРАБОТКА ПРОГРАММЫ	255
<i>Этап 1. Постановка задачи</i>	<i>255</i>
<i>Этап 2. Проектирование программы</i>	<i>256</i>
<i>Этап 3. Кодирование</i>	<i>256</i>
<i>Этап 4. Отладка</i>	<i>257</i>
<i>Что такое хорошо и что такое плохо?</i>	<i>258</i>

В ВАШУ ЗАПИСНУЮ КНИЖКУ	262
ИНТЕРФЕЙСЫ ДИСКОВОДОВ.	262
ИНТЕРФЕЙСЫ ДЖОЙСТИКА.	263
ЗВУКОВЫЕ УСТРОЙСТВА.	263
ПРОЧИЕ УСТРОЙСТВА.	264
ADVENTURE PROJECT	267
SHERLOCK	273
Правила "инглийского" языка.	273
Обсуждение.	274
Обобщение команд.	275
Сокращения.	275
Движение и перемещение.	276
Перемещение в заданное место.	276
Кэбы и поезда.	276
Вход и выход.	276
Преследование других людей.	277
Хождение по улицам Лондона.	277
Ход времени.	277
День и ночь.	277
Сон.	277
Ускорение времени.	277
Поведение персонажей.	277
Сотрудничество с другими персонажами.	277
Деньги.	278
Стратегия игры.	278
Повторение команд.	278
Изучение текущей ситуации.	278
Прерывание игры и выгрузка позиции.	279
MIAMI VICE	280
Управление.	280
Стрельба из окна машины.	280
Вход в здание.	281
Управление внутри зданий.	281
Специальные службы.	281
Расписание.	281
Время.	282
Очки.	282
STRATEGIC GAMES	284
PRESIDENT	284
FALKLANDS 82	285
AUSTERLITZ	286
ЦЕНА ИДЕИ	289
Установление приемлемой цены.	291
Факторы определения цен.	291
СПЕКТРУМ В ШКОЛЕ	293
BETA BASIC	296
ВСТУПЛЕНИЕ.	296
1. Команда ALTER.	297
2. Команда: AUTO.	298
3. Команда BREAK.	298
4. Команда CLOCK.	298
5. Команда: DEF PROC.	300
6. Команда: DELETE.	300
7. Команда: DO	300
8. Команда: DPOKE	301
9. Команда EDIT.	302
10. Команда: ELSE.	302
11. Команда: END PROC	303
12. Команда: EXIT IF	303

13. Команда: GET	303
14. Команда: KEYWORDS	303
15. Команда: LOOP	304
16. Команда: ON	304
17. Команда: ON ERROR	304
18. Команда: PLOT	305
19. Команда POP	306
МАЛЕНЬКИЕ ХИТРОСТИ	308
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	316
Элементы структурированной программы	316
1. Последовательные команды	316
2. Условное ветвление.	317
3. Циклы (итерации)	317
Объединение элементов	318
Как структурировать программу?	319
Процедура INPUT	321
СЕКРЕТЫ ПЗУ	327
СИСТЕМНЫЕ ПЕРЕМЕННЫЕ	327
Флаговые переменные.	329
Подпрограмма "CURSOR LEFT".	331
Подпрограмма "CURSOR RIGHT".	331
Подпрограмма "CARRIAGE RETURN"	331
Подпрограмма "PRINT COMMA".	331
Подпрограмма "PRINT A QUESTION MARK".	331
Подпрограмма "CONTROL CHARACTERS WITH OPERANDS".	331
Подпрограмма PRINTABLE CHARACTER CODES.	333
Подпрограмма "POSITION STORE".	333
Подпрограмма "POSITION FETCH".	333
Подпрограмма "PRINT ANY CHARACTERS"	333
Подпрограмма "PRINT ALL CHARACTERS"	335
Подпрограмма "SET ATTRIBUTE BYTE"	336
Подпрограмма "MESSAGE PRINTING".	336
Подпрограмма "PO-SAVE".	337
Подпрограмма "TABLE SEARCH".	337
Подпрограмма "TEST FOR SCROLL".	338
Подпрограмма "TEMPORARY COLOR ITEMS"	339
Подпрограмма "CLS COMMAND".	339
Подпрограмма "SCROLLING".	340
Подпрограмма "CLEAR LINES".	341
Подпрограмма "CL-ATTR".	341
Подпрограмма "CL-ADDR".	341
Подпрограмма "COPY COMMAND".	342
Подпрограмма "COPY-BUFF".	342
Подпрограмма "CLEAR PRINTER BUFFER".	342
Подпрограмма "COPY-LINE".	342
FORUM	344
АВТОСТАРТ ПРОГРАММ В МАШИННЫХ КОДАХ.	344
ДЕКОДЕР РАДИОПЕРЕДАЧИ.	344
К ВОПРОСУ О СОВМЕСТИМОСТИ	345
COPY-COPY	346
ОРГАНИЗАЦИЯ КЛУБОВ	349
PSI CHESS	350
ВСТУПЛЕНИЕ	350
ИГРА.	351
ДЕТАЛЬНОЕ ОПИСАНИЕ ПРОГРАММЫ.	352
FORUM	358
"Взрыв сверхновой".	360
"Cloaking device".	361
Е.С.М. "JAMMER".	361

Загадочные явления.	361
SABOTEUR-2.....	362
POKES	364
СЛОВО ЭКСПЕРТАМ	372
THUNDERBIRDS 1-4	372
Миссия 1. MINE RESCUE.....	373
Миссия 2. SUBMARINE CRASH	373
Миссия 3. BANK JOB.	374
Миссия 4. EVIL HOOD	374
Запуск игры	375
Экран	375
Управление.....	375
Применение предметов	376
Сила	376
Некоторые советы	376
Пароли	377
HAMMERFIST	378
MYTH.....	379
KLAX	379
OVERLORD.....	380
Настройка программы.	381
Некоторые советы.	382
OVERLORDS.....	383
Порядок игры.....	383
Цель игры.	384
Фигуры.	384
Территория.....	384
Сражения.	384
Пополнение армии.	384
Информация на экране.	384
Карта.	385
Повторение игры.	385
Запись игры на ленту.	385
Правила ведения войны.	385
Дополнительная информация при работе с двумя "Спектрами".	385
ZX LPRINT III	386
СЕКРЕТЫ ПЗУ	395
ПРОЦЕДУРЫ РЕДАКТОРА.	395
Подпрограмма "KEYBOARD INPUT".....	399
Подпрограмма "LOWER SCREEN COPYING"	400
Подпрограмма "SET-HL AND SET-DE".	401
Подпрограмма REMOVE-FP.....	401
BETA BASIC	402
20. Команда: PROC	402
21. Команда: RENUM.....	402
22. Команда: ROLL	403
23. Команда SCROLL	404
24. Команда: SORT.....	404
25. Команда: TRACE	406
26. команда: UNTIL.....	406
27. Команда USING.....	406
28. Команда WHILE.....	407
29. XOS,XRG,YOS,YRG	407
ЦЕНА ИДЕИ	409
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	418
ОШИБКИ В ПРОЕКТЕ.	419
1. Размер программы.	419
2. Быстродействие программы.	419

3. Концепция программы.	419
4. Цели программы.	419
5. Интерфейс "человек-компьютер".	419
6. Структура программы.	419
ОШИБКИ В ПРОГРАММИРОВАНИИ.	419
Пути сокращения количества ошибок.	420
<i>Для тех, кто пишет на БЕЙСИКе.</i>	420
ПРОБНЫЙ ЗАПУСК.	421
ПОЛЬЗОВАТЕЛЬСКАЯ ПРОВЕРКА.	422
ПОСЛЕДНИЙ БАРЬЕР.	422
ПЛАН ТЕСТИРОВАНИЯ.	422
СЛОВО ЭКСПЕРТУ.	423
EQUINOX.	423
<i>Настройка программы.</i>	424
TELEPORTER TERMINAL.	425
TRANS-LEVEL TELEPORTER.	425
CREDIT DISPENSER.	425
DISPOSAL CHUTE.	426
MAGNETIC LIFT.	426
Экран программы.	426
Работа с предметами и описание некоторых уровней.	427
Уровень 3.	429
Уровень 4.	429
Полезные советы.	431
ADVENTURE PROJECT.	433
РАСЧЕТ ХАРАКТЕРИСТИК ПЕРСОНАЖЕЙ.	433
FORUM.	437
Привидения в ОЗУ.	437
Что такое стек?	438
Зачем нужны прерывания?	438
К вопросу о совместимости.	440
Интерфейс MIDI.	440
Нужна помощь.	441
Компьютер в моем городе.	442
Наш каталог.	443
ИНФОРКОМ ПРЕДЛАГАЕТ...	443
Просим откликнуться.	443
Что бы это значило?	443
СПЕКТРУМ В ШКОЛЕ.	444
Графики функций.	444
Исследование матриц.	445
BETA BASIC.	449
Часть 2. Функции.	449
Общий обзор.	449
1. Функция: FN C\$	449
2. Функция FN D.	449
3. Функция: FN H\$	450
4. Функция FN I.	450
5. Функция: FN M.	451
6. Функция: FN N.	451
7. Функция FN P.	451
8. Функция: FN S\$	452
9. Функция FN T\$ ()	452
10. Функция: FN U\$	452
Приложение А.	452
Набор символов.	452
Приложение Б.	453
Сообщения.	453
СЕКРЕТЫ ПЗУ.	454

Подпрограмма <i>INITIALIZATION</i>	454
Подпрограмма <i>"MAIN EXECUTION LOOP"</i>	456
Тексты сообщений.....	457
РАЗДЕЛ ИСХОДНОЙ ИНФОРМАЦИИ О КАНАЛАХ.....	458
Исходная информация о потоках.....	459
Подпрограмма <i>WAIT-KEY</i>	459
Подпрограмма <i>"MAIN PRINTING"</i>	460
Подпрограмма <i>"CHAN-OPEN"</i>	460
Подпрограмма <i>CHAN-FLAG</i>	460
Подпрограмма <i>"MAKE-ROOM"</i>	461
Подпрограмма <i>POINTERS</i>	461
Подпрограмма <i>"COLLECT A LINE NUMBER"</i>	462
Подпрограмма <i>RESERVE</i>	462
Подпрограмма <i>SET-MIN</i>	462
Подпрограмма <i>INDEXER</i>	463
Подпрограмма <i>"CLOSE #"</i>	463
АППАРАТНОЕ ОБЕСПЕЧЕНИЕ	465
SECAM CODER.....	465
КОДЕР СЕКАМ.....	465
ZX-LPRINT III.....	471
FORUM	472
СООБЩЕНИЯ ОБ ОШИБКАХ.....	472
<i>DTR+</i>	472
<i>"Секреты ПЗУ" №4,5</i>	472
К ВОПРОСУ О СОВМЕСТИМОСТИ.....	472
ELITE.....	474
2. <i>Заправка топливом от звезды</i>	474
3. <i>Новые возможности управления</i>	475
4. <i>Космические платформы</i>	475
5. <i>Невидимые планеты</i>	476
6. <i>Интересные идеи</i>	476
7. <i>Корабли поколений</i>	476
9. <i>Тактико-технические данные и внешний вид кораблей</i>	476
ADDER.....	477
ASP MK II.....	477
COBRA MK III.....	478
FER-DE-LANCE.....	478
KRAIT.....	478
PYTHON.....	479
SIDEWINDER SCOUT SHIP.....	479
THARGOID INVASION SHIP.....	479
VIPER.....	480
<i>Домашнее задание</i>	480
TRANTOR.....	480
НАШ КОНКУРС.....	481
СЛОВО ЭКСПЕРТА.....	482
LAST NINJA 2.....	482
<i>Управление игрой</i>	482
<i>Примечания</i>	483
BATMAN-2.....	484
<i>Сюжет</i>	484
<i>Настройка программы</i>	484
<i>Игра</i>	485
<i>Управление игрой</i>	487
GUNFRIGHT.....	488
<i>Структура экрана</i>	488
<i>Управляющие клавиши</i>	489
<i>Полезные советы</i>	489
POKES	490
ADVENTURE GAMES	493

FORUM	498
БЕТА-БЕЙСИК 1.8	500
Дополнение к инструкции БЕТА-БЕЙСИК 1.0	500
1. Команда <i>CLOCK</i>	500
2. Команды управления курсором.	500
3. Команда: <i>DEF KEY</i>	501
4. Команда: <i>FILL</i>	502
5. Команда: <i>JOIN</i>	502
6. Команда: <i>KEYIN</i>	503
7. Команда: <i>LIST</i>	503
8. Команда <i>POKE</i>	503
9. Команды: <i>ROLL</i> и <i>SCROLL</i>	505
10. Команда <i>SPLIT</i>	505
ФУНКЦИИ	505
1. Функция: <i>AND</i>	506
2. Функция <i>BIN\$</i>	506
3. Функция: <i>COSE</i>	507
4. Функция: <i>FILLED</i>	507
5. Функция: <i>MEMORY\$</i>	507
6. Функция <i>MOD</i>	507
7. Функция: <i>OR</i>	508
8. Функция: <i>RNDM</i>	508
9. Функция <i>SCRN\$</i>	508
10. Функция: <i>SINE</i>	508
11. Функция: <i>XOR</i>	508
КАНАЛЫ И ПОТОКИ	509
Прочие каналы.	512
Область информации о каналах	514
Использование процедур ПЗУ	516
И снова о потоках	516
Как закрыть канал.	520
Файлы последовательного доступа на <i>RAM</i> -диске	521
<i>RAM</i> -диск	522
Структура каталога.	525
Канал "V"	525
Структура блока информации о канале "V".	527
MACHINE CODE	532
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	538
ADVENTURE PROJECT	546
ОБУЧАЮЩИЕ ПРОГРАММЫ	554
FORUM	555
К ВОПРОСУ О СОВМЕСТИМОСТИ.	555
ПРОБЛЕМЫ <i>ELITE</i>	557
СОВЕТЫ И СЕКРЕТЫ	559
НЕПРОБИВАЕМАЯ ЗАЩИТА	560
PYJAMARAMA	562
СЛОВО ЭКСПЕРТАМ	566
<i>AMAUROTE</i>	566
Город	566
Насекомые.	566
Бомбы.	567
Радио	567
Радар.	567
Управление	567
<i>FREDDY HARDEST</i>	568

Оружие.....	568
Город.....	569
Полезный совет.....	569
Экран.....	569
Очки.....	569
FREDDY HARDEST II.....	569
Игра.....	569
Заправка.....	570
Открытие ангара.....	570
Код доступа.....	570
Взлет.....	570
Полезные советы.....	570
Инопланетяне.....	570
Экран.....	571
Управление.....	571
Очки.....	571
SILENT SERVICE.....	571
Настройка программы.....	572
1. Экраны карт.....	573
2. Боевая рубка.....	573
3. Перископ.....	574
4. Палуба.....	575
5. Приборная панель.....	576
6. Экран повреждений.....	576
7. Экран побед.....	576
I. Торпедно-пушечная практика.....	576
II. Нападение на караван.....	576
III Боевое патрулирование.....	577
Полезные советы.....	577
HUNTER-KILLER.....	578
Подводные лодки типа "S".....	578
Управление подводной лодкой.....	578
Работа с перископом.....	579
Радар, гидролокатор, эхолот.....	579
Штурманская рубка.....	580
Стрельба торпедами.....	581
Прочие опасности.....	581
Управляющие клавиши.....	581



НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ИНФОРКОМ", 107311, МОСКВА, Б-241.

ИНФОРКОМ поздравляет всех любителей замечательного бытового персонального компьютера "СПЕКТРУМ" с Новым Годом, благодарит тех, кто выразил ему свое доверие, выписав персональный ньюс-леттер ZX-РЕВЮ и представляет первый выпуск.

Мы глубоко ценим то доверие, которое Вы нам оказали. Мы знаем, что работаем с простыми тружениками, не имеющими лишних средств, чтобы выписывать "кота в мешке", тем дороже нам ваши письма и слова благодарности, которых мы получили немало. Для нас это стимул, чтобы сделать "РЕВЮ" как можно более содержательным.

Мы разделили "ZX-РЕВЮ" на отдельные выпуски, чтобы привлечь как можно больше людей с разными интересами и с ограниченными финансовыми возможностями. Но сегодня в качестве рождественского сюрприза мы отправляем полный комплект всем, в том числе и тем, кто подписался только на приложения.

Как знать, ведь интересы у людей меняются, завтра Вам может стать нужным то, что сегодня неинтересно, может быть у Вас появятся новые знакомые, друзья, и это будет интересно им.

Одним словом, если Вы решитесь выписать полный комплект, то можете произвести доплату, условия приведены на стр. 8.

Мы можем предположить, что некоторым может не понравиться то, что мы предлагаем. Может быть это не то, чего они ждали. Для нас самая лучшая реклама - "довольный клиент", поэтому пожалуйста запечатайте этот номер в конверт и направьте в наш адрес наложенным платежом, оценив в сумму ваших затрат на подписку плюс стоимость ваших почтовых расходов. Мы его примем назад, а Вы полностью компенсируете затраты.

Прекрасно сознавая, что наше полиграфическое исполнение оставляет желать лучшего, мы приложим все усилия, чтобы компенсировать его недостатки интересным и полезным содержанием.

У нас нет пока возможности использовать более мощную полиграфическую базу, но мы знаем из ваших писем, что тысячи любителей по всей стране нуждаются в наших материалах сегодня, а не через год или два. Мы надеемся, что искренние поклонники синклер-совместимых компьютеров проявят к нам снисхождение в этом вопросе.

Если полиграфия пока от нас не зависит, то от нас зависят содержание, и каждая страница РЕВЮ эквивалентна примерно шести страницам стандартного машинописного текста, к концу года Вы будете обладателем материалов объемом до 1500 стандартный страниц, если будете аккуратно подшивать и хранить наши выпуски.

Мы получаем много писем, в которых наши читатели просят разработать те или иные остро необходимые им программы, выполнить все просьбы мы не в состоянии, сколько бы ни работали. Решить все проблемы можно только всем миром. Поэтому основную задачу ИНФОРКОМ видит в том, чтобы научить всех желающих тому, как взять максимум возможного от компьютера, как наиболее эффективно использовать то богатейшее программное обеспечение, которое уже создано.

Подчеркивая, что цели народного образования является для нас основными, мы начинаем свои публикации с рубрики "СПЕКТРУМ В ШКОЛЕ".

Дело, которое мы начали, является для нас новым, нет еще необходимого опыта, и мы открыты для контактов с теми, кто может нас в чем-то поправить и что-то нам подсказать.

Мы не прекращаем прием подписки. Те, кто присоединятся в 1991 году, получают и все ранее вышедшие номера, подробности на с. 8.

Просим извинить за длинное вступление, но это самый первый выпуск и в дальнейшем обещаем не занимать место несущественным содержанием.

СПЕКТРУМ В ШКОЛЕ

ОРГАНИЗАЦИЯ СТРУКТУР ДАННЫХ В УЧЕБНЫХ ПРОГРАММАХ

Одна из задач, с которыми сталкивается программист, пишущий обучающие или экзаменующие программы состоит в том, чтобы включить в программу рисунки, схемы, чертежи, поясняющие учебный материал. Это может быть сделано разными способами. Можно, например, нарисовать нужные иллюстрации в графическом редакторе типа ARTSTUDIO ИЛИ ARTIST, сохранить их на ленте вместе с программой, загружать вместе с ней в отведенные участки памяти (LOAD ""CODE адрес, длина) и вызывать по мере необходимости. Но поскольку в "СПЕКТРУМе" каждый цветной экран занимает 6912 байтов, а черно-белый - 6144 байта, то такой подход довольно расточителен. Можно готовую картинку компрессировать с помощью специальной процедуры (одна из возможных приведена в этом номере), что дает возможность еще в 2-3 раза увеличить количество рисунков, находящихся в памяти компьютера, но предел все равно близок.

Очень часто фирмы, выпускающие программное обеспечение применяют следующий прием. Они отводят под изображение верхнюю треть экрана (восемь строк). Тогда рисунок, даже и не компрессированный, занимает лишь 2046 байтов в черно - белом варианте или 2304 в цветном, это уже лучше, и этим методом пользуются, если есть необходимость в большом количестве картинок с высоким разрешением, как например в игровых программах.

Другой способ исполнения несложных поясняющих схем на экране состоит в использовании графики пользователя (UDG). Правда, графические символы, задаваемые пользователем, могут быть вами уже заняты, например для изображения русских букв. Тогда вводят дополнительные банки UDG и переключают их по мере необходимости. Мы об этом еще будем писать в последующих выпусках.

Простейший и наиболее экономичный способ рисования несложных схем состоит в использовании операторов БЕЙСИКа:

PLOT - для изображения точек;

DRAW - для изображения линия и дуг окружностей;

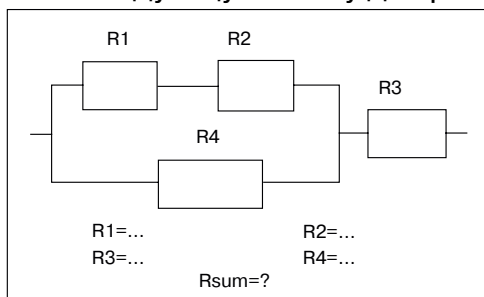
CIRCLE- рисование окружностей;

PRINT - для печати текста на схемах.

Однако, если Вы, скажем, готовите экзаменационные билеты по физике и Вам их надо, например, 99 штук, каждый со своим рисунком, то можно представить, во что превратится программа. Это будет бесконечная путаница из PLOT, DRAW, . . . А если завтра Вам надо будет подготовить пакет билетов по химии, то всю программу придется переписывать сначала?

Вот здесь Вам и могут пригодиться наши предложения по организации структур данных, которые позволят хранить изображение в строке DATA очень рациональным использованием памяти. Более того, создавая новый пакет билетов по другому материалу, Вам придется менять в основном только строки DATA, остальная же часть программы остается без существенных изменений.

Предположим, Вам надо ввести следующую схему для расчета сопротивления цепи:



На базе такой схемы Вы можете организовать целую серию вопросов, меняя данные при R1 - R4 и анализируя ответ ученика.

Допустим, что в Вашей программе 99 разных схем и пусть для их хранения отведены строки, начиная с 8000, пусть приведенная схема имеет номер $n=73$. Тогда фрагмент программы будет выглядеть так:

```
1 DIM a(3)
996 LET n=73
997 RESTORE 8000+n*10
998 GO SUB 2000
999 STOP
```

Мы здесь поставили "заглушку". Вы же сможете организовать логику, как Вам надо.

```
2000 REM
```

Эта подпрограмма анализирует строку DATA, на которую указал оператор RESTORE и вызывает рисующие подпрограммы. Свою работу подпрограмма заканчивает, когда встретит символ "/", который должен быть последним в строке DATA. Переменная datnum указывает сколько числовых параметров должно быть введено. Например для оператора PLOT m, n - их два, а для CIRCLE m, n, r - три.

```
2010 CLS
2020 READ a$
2100 IF a$ <> "p" THEN GO TO 2200
2110 LET datnum=2
2120 GO SUB 3000
2200 IF a$ <> "l" THEN GO TO 2300
2210 LET datnum=2
2220 GO SUB 4000
2300 IF a$ <> "c" THEN GO TO 2400
2310 LET datnum=3
2320 GO SUB 5000
2400 IF a$ <> "d" THEN GO TO 2500
2410 LET datnum=3
2420 GO SUB 6000
2500 IF a$ <> "t" THEN GO TO 2600
2510 GO SUB 7000
2600 IF a$ = "/" THEN RETURN
```

Возврат в вызывающую программу, если нарисован весь рисунок.

```
2610 GO TO 2020
```

Возврат к изображению очередного элемента.

```
2700 REM подпрограмма вводит числовые параметры для операторов PLOT, DRAW, CIRCLE, PRINT AT.
2710 FOR i=1 TO datnum
2720 READ a(i)
2730 NEXT i
2740 RETURN
3000 REM изображается точка
3020 GO SUB 2700
3020 LET m=a(1)
3030 LET n=a(2)
3040 PLOT m, n
```

Подпрограмма напечатает точку в позиции экрана с координатами m,n, где m-координата по горизонтали (от 0 до 255), а n-координата по вертикали от 0 до 175.

```
3050 RETURN
4000 REM изображение линии
4010 GO SUB 2700
4020 LET m=a(1)
4030 LET n=a(2)
4040 DRAW m,n
4050 RETURN
5000 REM изображение окружностей
5010 GO SUB 2700
5020 LET m=a(1)
5030 LET n=a(2)
5040 LET r=a(3)
5050 CIRCLE m, n, r
```

```

5060 RETURN
6000 REM изображение дуг
6010 GO SUB 2700
6020 LET m=a(1)
6030 LET n=a(2)
6040 LET alfa=a(3)
6050 DRAW m,n,alfa
6060 RETURN
7000 REM Печать текста на экране, начиная с заданной позиции m, n, где m - номер строки от 0
      до 21, а n - номер столбца от 0 до 31.
7010 GO SUB 2700
7020 READ t$
7030 LET m=a(1)
7040 LET n=a(2)
7050 PRINT AT m,n; t$
7060 RETURN
6000 REM здесь в строках DATA хранятся схемы, чертежи и рисунки для Вашей программы.
6010 DATA .....
6730 DATA "p",0,119, "l",30,0, "l",0,80, "l",20,0, "l",0,6, "l",32,0, "l",0,-12, "l",-32,0,
      "l",0,6
6731 DATA "p",62,139, "l",40,0, "l",0,6, "l",32,0, "l",0,-12, "l",-32,0, "l",0,6
6732 DATA "p",154,139, "l",20,0, "l",0,-20, "l",20,0, "l",0,6, "l",32,0, "l",0,-12, "l",-
      32,0, "l",0,6
8733 DATA "p",226,119, "l",20,0, "p",30,119, "l",0,-20, "l",50,0, "l",0,6, "l",32,0, "l",0,-
      12, "l",-32,0, "l",0,6, "p",112,99, "l",68,0, "l",0,20
8734 DATA "t",2,7, "R1", "t",2,16, "R2", "t",5,25, "R3", "t",7,11, "R4", "t",14,6, "R1=...",
      "t",14,16, "R2=...", "t",16,6, "R3=...", "t",16,16, "R4=...", "t",16,11, "Rsum=?",
      "/"

```

Тот подход к организации данных, который мы Вам здесь продемонстрировали, может пока показаться не очень удобным, но его преимущества раскроются, когда Вам надо будет ввести в программу другой рисунок и организовать на его базе серию учебных задач.

Предположим теперь что Вам надо проиллюстрировать задачу о бросании камня. Пусть рисунок к ней имеет номер $n=74$. Поменяйте в строке 996 значение n , а далее вся переделка сведется к тому, чтобы ввести несколько новых строк DATA.

```

8740 DATA "p",0,80, "l",255,0, "p",30,60, "l",180,0,-2, "p",120,30, "l",0,49, "p",119,127,
      "p",121,127, "p",119,62, "P",12,1,62, "p",50,80, "d",-10,13,1,2, "p",20,90, "l",20,20
8741 DATA "p",36,109, "p",39,108, "p",30,60, "l",0,-32, "p",210,80, "l",0,-32, "p",30,50,
      "l",160,0, "p",33,51, "p",32,49, "p",208,51, "p",208,49
8742 DATA "t",8,16, "H=?", "t",14,14, "L=?", "t",10,7, "alfa=...", "t",6,0, "V=...", "/"

```

В строках с 1000 по 2000 Вы можете организовать ввод ответов от ученика, проверку их правильности и принятие решения. Решений может быть несколько. Если ответ верен, то либо дать поощрение, либо перейти к очередному вопросу, либо и то и другое вместе. Если неверен, то либо дать подсказку, например показать расчетную формулу, либо дать порицание. Все подсказки и пр. сообщения тоже должны храниться где-то в строках DATA и вызывать их надо с использованием оператора RESTORE перед READ.

Например:

```
RESTORE 9000+n*10+q
```

здесь n - номер билета, а q - флажок, который зависит от правильности ответа.

$q=0$ - ответ верен, поощрить.

$q=1$ - ответ неверен, дать вторую попытку и пожурить.

$q=2$ - ответ опять неверен, дать подсказку.

$q=3$ - ученик просто не знает материала, сделать вывод.

Организовать в таком духе диалог ученика и компьютера Вы сможете всего за несколько дней. Не ставьте для начала сверхзадач, и Вы увидите, что компьютер станет прекрасным инструментом в школе.

Мы не сомневаемся, что Вы сумеете найти новые более эффективные решения по структурированию данных. Мы специально ограничились не самыми головоломными примерами, чтобы разбудить Вашу творческую мысль, и если нам это удалось, то значит наша цель достигнута.

MEGA-BASIC

Сегодня ИНФОРКОМ начинает печатать серию статей, посвященных работе с наиболее популярными программами. Открывает эту серию описание работы с языком программирования YS MEGABASIC. Аббревиатура YS возникла потому, что в Англии этот язык был впервые представлен на страницах журнала YOUR SPECTRUM, позже на его страницах неоднократно помещались интересные прикладные программы, написанные на этом языке, видимо в силу конкурентной борьбы работа с этим языком практически никогда не освещалась на страницах других изданий, как SINCLAIR USER, ZX-COMPUTING, SINCLAIR PROGRAMS, и у нас в стране внедрение этого языка в практику программирования несколько затянулось, а напрасно. По нашему мнению, этот язык наилучшим образом подходит для создания обучающих программ по сравнению с другими известными для "СПЕКТРУМа" языками программирования, он обладает рядом значительных преимуществ, на которые мы укажем.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

1. Программа занимает порядка 20K и располагается в верхних областях ОЗУ.
2. Уникальной особенностью является то, что кроме основного шрифта она поддерживает одновременно еще и два подзагружаемых. Обратите внимание те, кто пишет программы обучения английскому языку. Одновременно на экране Вы можете иметь и русский в английский текст и транскрипцию.
3. Каждый из этих шрифтов может одновременно(!) в пределах одного экрана изображаться четырьмя размерами от 16 до 64 знаков в строке.
4. Дополнительно к этому Вы можете легко печатать символами любого размера, хоть во весь экран, можете делать из них "бегущую строку" (как насчет исполнения титров и заставок к видеофильмам, конечно если в Вашем компьютере есть видеовыход, как в фирменном?)
5. Возможно создание и использование до 10 окон в пределах экрана. Произвольный "скроллинг" изображения по горизонтали и вертикали в окне. Уникальная возможность иметь на экране одновременно и листинг программы в одном окне и результат ее работы в другом.
6. Широкие возможности структурного программирования благодаря наличию понятия "процедура". Большие программы сразу перестают быть проклятием БЕЙСИКа, появляется возможность коллективного написания программ, более того, созданные Вами процедуры Вы можете переносить из программы в программу без изменения, создав себе тем самым библиотеку и сократить тем самым время на разработку новых программ процентов на 60, а может быть и больше.
7. Новые операторы цикла.
8. Дополнительные отладочные операторы.
9. Улучшенные возможности управления звуком.
10. Встроенный монитор для легкого внедрения в машинный код с целью проверки и внесения изменений.

Обращаем Ваше внимание на то, что MEGABASIC поддерживает и все команды и функции стандартного встроенного БЕЙСИКа и, поскольку приступая к новому языку, Вы по крайней мере с ними должны быть знакомы, мы абсолютно не будем рассматривать стандартные команды, а остановимся только на новых.

ИНФОРКОМ выражает глубокую признательность Пашорину В.И. из города Балашова Саратовской обл., который выполнил перевод с английского языка фирменной инструкции. Техническое редактирование и вступительная статья - наши.

1. КЛАВИАТУРА

После запуска программы Вы увидите короткое сообщение и мигающий квадрат в левом нижнем углу экрана. Это новый курсор, он показывает, где будет помещаться вводимая информация. В МЕГАБЕЙСИКе для ввода информации может использоваться

весь экран, а не только две нижние строки, как при работе со стандартным БЕЙСИКОМ, содержащимся в ПЗУ компьютера.

Попытайтесь записать несколько символов с клавиатуры, и Вы заметите, что привычные Вам ключевые слова, соответствующие нажатой клавише не появляются на экране, а появляются только одиночные символы. Теперь Вам придется записывать все операторы полностью, так, как это делается на большинстве компьютеров.

Например, чтобы набрать слово PRINT, вам уже недостаточно просто нажать на клавишу "P", а придется набрать 'P','R','I','N','T' полностью. Впрочем, некоторые операторы, пришедшие из БЕЙСИКа, можно записывать и в сокращенной форме, но она должна заканчиваться точкой. Ниже мы привели список этих ключевых слов, указав где должна стоять точка.

A.TTR	BE.EP	B.IN
BO.RDER	BR.IGHT	CH.R\$
CI.RCLE	CLE.AR	CL.OSE#
C.ODE	CON.TINUE	DA.TA
D.EF FN	DR.AW	ER.ASE
E.XP	FL.ASH	F.ORMAT
GO S.UB	G.O TO	I.NKEY\$
INP.UT	INV.ERSE	L.EN
LI.NE	LL.IST	LP.PRINT
LO.AD	ME.RGE	M.OVE
NE.XT	N.OT	OP.EN#
OV.ER	PA.PER	PAU.SE
PE.EK	PL.OT	P.OINT
PR.INT	RA.NDOMIZE	RE.AD
RES.TORE	RET.URN	R.ND
SA.VE	S.GREEN\$	ST.R\$
T.AB	TH.EN	U.SR
V.AL\$	VE.RIFY	

При записи операторов типа GO TO не забывайте делать пропуски. Компьютер поймет GO TO, но при записи GOTO выдаст сообщение об ошибке SYNTAX ERROR.

Нижняя строка показывает режим курсора:

'L' - литерный строчный CAPS OFF

'C' - литерный прописной CAPS ON

'G' - графический

'E' - расширенный EXTENDED

РЕДАКТИРОВАНИЕ

В программе реализован довольно необычный режим редактирования, который имитирует полноэкранный редактор. Для этого в работе участвуют два курсора. Первый - входной курсор, а второй - COPY-курсор.

Клавиши управления редактором:

КЛАВИША	НАЗНАЧЕНИЕ
EDIT	-копирование очередного символа в буфер редактора.
TRUE VIDEO	-удаление всей строки.
INV. VIDEO	-удаление символа справа от курсора.
CURS. LEFT	-перемещение курсора влево на один символ.
CURS. RIGHT	-вправо на один символ.
CURS. UP	-вверх на один символ.
CURS. DOWN	-вниз на один символ.
DELETE	-удаление символа слева от курсора.
<=	-перемещение курсора в начало строки.
<>	- удаление всех символов от курсора до начала строки.
>=	-перемещение курсора в конец строки.
SCREEN\$	-листинг программы.
OR	-перемещение указателя текущей строки вверх и листинг.
AND	-то же, но вниз.

STOP	-перемещение COPY-курсора влево.
NOT	-то же, но вниз.
STEP	-то же, но вверх.
TO	-то же, но вправо.
AT	-копирование символа от COPY-курсора к выходному курсору. Эта команда может использоваться только при работе с символами стандартного размера, но не в режимах 64 знака в строке и не с символами двойной ширины и высоты.
OVER	-перемещение COPY-курсор на следующее окно.
INVERSE	-возвращение COPY-курсора в левый верхний угол окна.

В режиме редактирования на экране присутствуют три окна.

0 - окно индикации информации, поступающей от пользователя и выдачи сообщений об ошибках.

1 - окно автоматического листинга программы.

2 - окно выходной информации.

Подробнее об окнах мы напишем в следующем выпуске. В первый момент после загрузки МЕГАБЕЙСИКа Вам может показаться, что у Вас только одно окно, но это не так, поскольку окна имеют право перекрывать друг друга.

В отличие от встроенного редактора, в МЕГАБЕЙСИКе Вы можете напрямую вызывать нужную, а не только текущую строку на редактирование. Это делается подачей команды EDIT n. Здесь n-номер нужной Вам строки.

(Продолжение в следующем выпуске)

128K

ВОПРОС НАШЕГО ЧИТАТЕЛЯ:

Уважаемый ИНФОРКОМ! В своей разработке "Большие Возможности Вашего СПЕКТРУМа" Вы пишете о том, что в 128 килобайтных моделях область памяти, которая в 48 килобайтных машинах отводилась для буфера принтера, теперь занята новыми системными переменными. Эта область расположена в адресах с 23296 по 23551. Но как же так, ведь у компьютеров с памятью 128K размер ПЗУ равен 32K? Почему же в его пределах может существовать область системных переменных?

Вопросы, подобные приведенному, нам задают многие читатели. И это не случайно. Если информационный голод по 48-килобайтным машинам еще как-то утоляется, в том числе и нашими стараниями, то про модели 128K этого пока сказать нельзя. Многие просят рассказать об организации памяти в этом компьютере, а она действительно несколько необычна, дело в том, что SINCLAIR RESEARCH, приступая к этой разработке поставила задачу добиться в первую очередь максимальной совместимости со своими предыдущими моделями SPECTRUM 16 и SPECTRUM 48 как по программному обеспечению, так и по имевшемуся периферийному окружению. Именно это требование и наложило серьезный отпечаток на организацию памяти в новой модели.

В нескольких номерах ZX-РЕВЮ мы поделимся с Вами тем, что знаем по этому вопросу.

Во-первых, основным отличием этой модели от своих предшественников является очевидно значительное увеличение как оперативной, так и постоянной памяти. Фактически "СПЕКТРУМ 128" имеет 32K ПЗУ и 128K ОЗУ, т.е. всего 160 килобайт. Отнюдь немалое количество для домашнего ПК! И организована эта память в страницы. Страница - это просто отдельный блок памяти. Каждая страница имеет 16 килобайт и, тем самым, в 128-килобайтных машинах можно выделить 2 страницы ПЗУ и восемь страниц ОЗУ.

$$2 \times 16K = 32K \text{ (ПЗУ)}$$

$$8 \times 16K = 128K \text{ (ОЗУ)}$$

Те, кто читал уже наши разработки, знают, что в обычном "СПЕКТРУМЕ-48" память непрерывна и имеет 65535 байтов (64K). В шестнадцатеричной системе ее адреса записываются от 0000 до FFFF, причем первые 16K, т.е. от 0000 до 3FFF занимает ПЗУ, а остальное - ОЗУ. Понятно, что если мы хотим записать адреса в компьютере, имеющем больший объем памяти, то четырех знаков шестнадцатеричных чисел нам уже недостаточно, и необходимо вводить пятый знак, тогда адреса могут записываться, скажем 13579 или, скажем, 6ABCD.

Для того, чтобы разобраться как все это работает, давайте рассмотрим сначала организацию ПЗУ, ответив на вопрос нашего читателя.

ПЗУ организовано в две страницы, каждая из которых имеет по 16K. Первая страница имеет адреса от 00000 до 03FFF и называется ПЗУ-0, или "Новым ПЗУ". Когда Вы впервые включаете компьютер в сеть, или нажимаете кнопку RESET, если у Вас она есть, то происходит старт с нулевого адреса именно этого ПЗУ. "Новое ПЗУ" действительно совершенно новое, его не было в моделях 48K.

Вторая страница ПЗУ называется ПЗУ-1 и имеет адреса от 10000 до 13FFF. Его иногда называют "Старым ПЗУ". Хотя в него и были внесены небольшие изменения, но практически оно соответствует тому ПЗУ, которое стоит в 48-килобайтных компьютерах. Обратите внимание, что теперь уже адресное пространство не является непрерывным, т.е. есть очевидный зазор между концом нулевого ПЗУ 03FFF и началом первого ПЗУ - 10000.

Сердцем компьютера является, как известно процессор. Синклер-совместимые компьютеры собраны на процессоре Z-80, который может работать с 16-битной адресацией. Другими словами, адресное пространство, которое в состоянии обслужить этот процессор, составляет 2 в шестнадцатой степени байтов, т.е. 64 килобайта, точнее 65536 адресов, никаких других адресов вне этих процессор не знает и знать не может.

Поэтому в командах процессора все адреса могут выражаться только четырьмя разрядами шестнадцатеричных чисел, но никак не пятью. Например, команда CALL может быть записана только как CALL pnnn. но никак не как CALL xnnnn. Точно также и все регистры процессора, даже двойные никак не могут содержать числа больше 4-х разрядов (в шестнадцатеричном виде), где же выход? Как работать с адресами ПЗУ типа 03463 или 12466?

А выход очень простой, он состоит в том, что команда CALL 2466 означает то же самое, что и команда CALL 02466, и то же самое, что и CALL 12466. Все зависит только от того, какая страница ПЗУ подключена в данный момент - нулевая или первая. Процессору все равно, с какой работать, но только с одной. Обе вместе не могут быть подключены, или, как говорят "впечатаны", никогда.

Поскольку существуют только два ПЗУ, то конечно такие адреса, как скажем A2468 не могут существовать, но те, кто разрабатывают разнообразные периферийные устройства с "теневым" (замещающим) ПЗУ могут рассматривать их адресные пространства как 2-ую страницу ПЗУ, третью и т.д. Надо только позаботиться, чтобы она была правильно впечатана в нужное время и в нужное место. О том как впечатываются страницы в 128-килобайтных "СПЕКТРУМах" мы расскажем в следующем выпуске, а сейчас остановимся на страничной организации ОЗУ.

Адресное пространство ОЗУ составляют адреса от C000 до FFFF и, так же как и для ПЗУ, сюда может быть впечатана только одна страница, страницы ОЗУ нумеруются от 0 до 7, всего их восемь и, соответственно, адреса ОЗУ могут быть представлены для соответствующих страниц как:

ОЗУ-0 0C000 - 0FFFF;

ОЗУ-1 1C000 - 1FFFF;

ОЗУ-6 6C000 - 6FFFF;

ОЗУ-7 7C000 - 7FFFF.

То ОЗУ, которое в данный момент впечатано, считается текущим ОЗУ или, иначе говоря, текущей страницей, номер текущей страницы таким образом, задается числом от 0 до 7, а с прочими числами, входящими в адрес, процессор работает как с обычными четырехразрядными шестнадцатеричными числами.

Теперь внимательный читатель наверное уже почувствовал, что здесь что-то не так.

Мы сказали выше, что все адреса от x0000 до x3FFF принадлежат ПЗУ, а все адреса от xC000 до xFFFF принадлежат ОЗУ, а куда же подевались адреса от 4000 до BFFF, т.е. целых 32К?

Ответ достаточно неожиданный. Этот участок разбит на два блока по 16К.

Первый - от 4000 до 7FFF.

Второй - от 8000 до BFFF.

За первым постоянно закреплена пятая страница ОЗУ, а за вторым - страница 2. И так, вторая и пятая страницы имеют постоянное закрепленное за ними место, да еще могут впечатываться в верхние 16К компьютера, как и все другие страницы. Возникает интересная ситуация, когда адрес 4000 означает то же самое, что и 5C000. Соответственно тождественны и адреса:

5000 и 5D000 6000 и 5E000

7000 и 5F000. То же относится и к странице 2.

Таким трюком К.Синклер создает иллюзию непрерывности адресного пространства в 128-килобайтных машинах, этим закладывается возможность программной совместимости с предыдущими моделями. С точки зрения процессора теперь любой доступный ему адрес от 0000 до FFFF физически что-то означает. В каждом адресе всегда есть ячейка памяти, причем ячейки от 0000 до 3FFF представляют ПЗУ, а остальные - ОЗУ. Да и ОЗУ простирается непрерывно от 4000 до FFFF, все как в обычных "СПЕКТРУМах".

На этой мы прервемся до следующего номера, в котором расскажем о том, как выполняется впечатывание страниц ПЗУ и ОЗУ, а заодно расскажем о том малоизвестном факте, что компьютеры 128К имеют две экранных области вместо одной.

РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ

Мы знаем, что особую сложность для начинающих представляет работа с операторами БЕЙСИКа READ и DATA, не говоря уже о RESTORE. Профессионалы, если вспомнят свои первые шаги, конечно с нами согласятся, причин здесь две. Во-первых работа этих операторов неочевидна, то есть требует специальных разъяснений, а во-вторых в БЕЙСИКе есть замечательный оператор INPUT, который позволяет удобно организовывать ввод данных, и начинающие предпочитают пользоваться им.

Сегодня мы предлагаем Вашему вниманию несложную программу, позволяющую воспроизвести на компьютере несколько простых мелодий. Работа звуковой динамика здесь основывается на использовании оператора BEEP m, n. Параметр m означает продолжительность звукового сигнала в секундах, а параметр n - высоту тона (частоту звукового сигнала).

Эти параметры вводятся с помощью операторов READ и DATA. Они хранятся в строках DATA и считываются оттуда оператором READ. Операторы READ и DATA работают в паре. Когда программа встречает очередной оператор READ, например READ Q, R, S - она обращается к ближайшему неиспользованному до сих ПОР оператору DATA и вводит из него столько чисел, сколько нужно оператору READ, в нашем случае - три (Q, R и S). Если в строке DATA столько чисел нет, то программа идет к следующему блоку DATA и читает данные из него. После того, как данные прочитаны, они считаются использованными и больше к ним так просто не вернуться.

После совместной работы READ и DATA возможны три варианта. Первый, когда сколько данных потребовалось для READ, столько и нашлось в DATA, это нормальный вариант, все в порядке. Бывает, что операторы READ уже взяли все, что им надо, а в строках DATA еще не все данные использованы. Это возможно, никакой ошибки здесь нет. Хуже, когда READ просит какие-то данные, а строки DATA уже исчерпаны, в этом случае появляется сообщение типа 80 OUT OF DATA. Оно означает, что тому оператору READ, который стоит в 80-й строке, не хватило данных в DATA. Скорее всего причина ошибки в той, что Вы по невнимательности пропустили какое-то число, набирая строку DATA.

Если Вы подумали, что данные в строках DATA могут быть использованы только раз и потом они недостижимы, то это не совсем так. Дело может поправить оператор RESTORE NNNN, где NNNN - номер строки, например RESTORE 2500. Этот оператор "восстанавливает" строки DATA, начиная со строки NNNN, после чего следующий оператор READ будет читать данные из этой строки и последующих и т. д.

С помощью оператора RESTORE можно не только "восстанавливать" строки DATA, но с тем же успехом и вообще манипулировать данными, определяя какая строка когда будет вводиться.

Предлагаемая программа даст Вам представление о том, как все это происходит.

* * *

Введите программу и нажмите RUN и ENTER. Вы можете остановить программу с помощью CAPS SHIFT + SPACE (или клавиши BREAK).

```
10 LET menu = 40 : LET dstart = 300 : LET dplay = 400
```

Здесь мы присвоили переменным значения номеров часто используемых строк, чтобы потом обращаться к строкам не по номеру, а по имени. Все-таки в имени есть какая-то информация, и если так делать почаще, то программа становится понятнее. Бывает трудно разобраться в своей же программе, написанной пару месяцев назад.

```
20 BORDER 4: PAPER 3: INK 9
```

```
30 BRIGHT 1: CLS Выставили цвета экрана.
```

```
40 REM menu
```

```
50 RESTORE dstart
```

```
55 FOR n = 1 TO 4: READ a$
```

Прочитали из строки DATA названия четырех мелодии.

```
60 PRINT AT 5 + 2*n, 1; PAPER 7; n;"="; a$;
```

Распечатали их на экране.

```
70 NEXT n
```

```
90 INPUT p
```

Запрос номера мелодии, которую Вы хотите прослушать.

```
100 IF p<1 OR p>4 THEN CLS: GO TO menu
```

В программе всего 4 мелодии и если нажата клавиша с номером, которого нет, возврат в меню.

```
110 IF p<>INT(p) THEN CLS: GO TO menu
```

Если введено не целое число, то тоже возврат в меню.

```
114 LET p$=STR$(p)
```

```
115 IF CODE(p$)<48 OR CODE(p$)>57 THEN CLS: GO TO menu
```

Если нажата не цифра, а иная клавиша, то тоже возврат. Коды цифр лежат в интервале от 46 до 57. Остальные символы - не цифры.

```
130 CLS
```

```
140 INPUT PAPER 7;"how many?"; v
```

Сколько раз повторить мелодию?

```
150 LET v = INT v
```

```
160 IF v<1 THEN GO TO menu
```

```
180 RESTORE dstart + p
```

Очередной READ устанавливается на название избранной мелодии.

```
190 READ a$:
```

Теперь рассчитаем величину "k", необходимую для центрирования названия мелодии на экране.

```
200 LET b$ = "□"
```

Это графический символ. Он вводится в графическом режиме нажатием CAPS SHIFT+8. Переход в графический режим выполняется нажатием CAPS SHIFT + 9 (появятся курсор G).

```
310 LET k=INT((32-LEN a$)/2):
```

```
315 LET s$=" "
```

Здесь один пробел,

```
220 PRINT AT 11,k-2; PAPER 7; b$; s$;; a$; s$; s$; b$
```

Нарисуем рамку вокруг названия.

```
240 FOR n = k-1 TO LEN a$+k+2
```

```
250 PRINT AT 10,n; PAPER 7; s$;
```

```
    AT 12, n; s$;
```

```
    AT 10, k-2; b$;
```

```
    AT 12, k-2; b$;
```

```
    AT 10, LEN a$+k+2;b$;
```

```
    AT 12, LEN a$+k+2;b$;
```

```
255 NEXT n
```

```
260 FOR n=k-2 TO LEN a$+k+2: PRINT AT 9,n; INK 0; b$; AT 13, n; b$
```

```
265 NEXT n
```

```
270 RESTORE dplay + p 275 READ q, r, s
```

Ввели параметры мелодий:

q - количество нот;

r - ритм (темп);

s - пауза между повторами.

Далее идет само исполнение избранной мелодии,

```
280 FOR m = 1 TO v
```

```
281 RESTORE dplay + P*100
```

```
282 FOR n = 1 TO q
```

```
283 READ a,b
```

```
284 BEEP a/r,b
```

```
285 NEXT n
```

```
286 PAUSE S
```

```
290 NEXT m
```

```
299 RUN
```

```
300 REM -----
```

```
301 DATA "Melody one"
```

```
302 DATA "Melody two"
```

```
303 DATA "Melody three"
```

```
304 DATA "Melody four"
```

```
400 REM -----
```

```
401 DATA 23,6, 16
```

402 DATA 73,4,75
403 DATA 36,3,1
404 DATA 33,3,33
490 REM ---Melody one----
500 DATA 2,0,2,0,2,4,2,7
501 DATA 6,12,10,9,2,9,8,5
502 DATA 2,7,2,9,16,7,2,0
503 DATA 2,0,2,4,1,7,1,7
504 DATA 6,7,10,2,2,4,2, 5
505 DATA 2,4,2,2,14,0
590 REM ---Melody two----
600 DATA 2,2,2,7,3,7,1,7
601 DATA 1,7,3,11,2,7,2,9
602 DATA 3,9,1,9,1,9,3,12
603 DATA 2,9,2,11,2,9,2,7
604 DATA 2,14,2,12,2,11,2,11
605 DATA 3,9,1,9,4,9,1,2
606 DATA 1,2,2,7,2,7,2,7
607 DATA 1,7,3,11,2,7,2,9
608 DATA 3,9,1,9,1,9,3,12
609 DATA 1,9,1,9,1,11,3,14
610 DATA 2,12,1,11,3,14,2,12
611 DATA 3,11,1,7,2,9,4,7
612 DATA 2,2,3,7,1,7,3,7
613 DATA 1,7,3,11,2,7,3,9
614 DATA 1,9,2,9,1,9,3,12
615 DATA 1,9,1,9,1,11,3,14
616 DATA 2,12,1,11,3,14,2,12
617 DATA 3,11,1,7,2,9,4,7
690 REM ----Melody three----
700 DATA 1,0,1,0,2,5,2,5
701 DATA 2,9,1,0,1,9,2,10
702 DATA 2,10,2,14,1,10,1,10
703 DATA 2,9,2,9,2,12,1,9
704 DATA 1,5,2,7,2,7,2,12
705 DATA 2,0,2,5,1,5,1,5
706 DATA 2,9,1,12,1,9,2,10
707 DATA 2,10,2,14,2,10,2,9
708 DATA 1,9,1,9,1,0,2,9
709 DATA 2,7,2,5
790 REM ----Melody four----
600 DATA 2,7,2,11,2,11,6,11
601 DATA 2,7,2,9,1,11,1,9
602 DATA 6,7,2,7,2,11,2,14
603 DATA 6,14,2,9,2,11,8,14
604 DATA 2,14,2,16,2,14,6,11
605 DATA 1,9,1,7,2,14,2,11
606 DATA 6,2,1,2,1,2,2,4
607 DATA 2,7,6,9,2,2,2,4,6,7

MACHINE CODE

В данной статье разобран оригинальный метод выполнения компрессии и декомпрессии экранной области "СПЕКТРУМА". Подробные комментарии позволяют использовать этот материал не только как полезную процедуру, но и как пособие тем, кто самостоятельно изучает программирование в машинных кодах и на языке АССЕМБЛЕРА.

ИНФОРКОМ выражает глубокую признательность И. Девятко из г. Нальчика, разработавшему и прокомментировавшему эти процедуры.

Техническое редактирование выполнено ИНФОРКОМом.

Экранная область в памяти компьютера начинается с адреса 16384 и занимает 6912 байтов. Если ваша программа содержит несколько картинок (экранов), то хранить их в памяти - дело довольно расточительное, поэтому желательно провести компрессию (архивацию) экрана перед отправлением его в память на хранение и, соответственно надо выполнять декомпрессию перед выдачей картинки на экран.

Если внимательно проанализировать содержимое экранной области памяти, то можно увидеть, что некоторые значения повторяются многократно. Предлагаемый здесь алгоритм основан на том, что если соседние ячейки не повторяются, то их значения просто переписываются в новую область, иначе переписываются первые два числа из группы повторяющихся, а за ними ставится двухбайтное число, показывающее количество повторений.

1. ПРОГРАММА КОМПРЕССИИ

LD HL, 16364 ; В HL -счетчик в экранной области.
LD BC, 6911 ; - в BC -количество байтов экрана.
LD DE, ADDR ; - в DE - адрес, в котором организуется хранение экрана.

Величину ADDR выберите сами. Экран содержит 6912 байтов, тем не менее мы поместили в BC число 6911, и было это сделано вот почему. В дальнейшем мы организуем в BC счетчик повторений одинаковых байтов, т.е. эта регистровая пара будет работать "за двоих". Возникает необходимость как-то сообщить программе о том, что экранная область кончилась, для этого с последним 6912-ым байтом мы сделаем несложную манипуляцию. Мы сохраним его на стеке, а вместо него поставим инвертированную копию предпоследнего байта. Теперь мы можем компрессировать 6911 байтов, твердо зная, что 6912-ый снимем со стека. На АССЕМБЛЕРЕ это выглядят так:

ADD HL, BC ; HL = HL+BC, теперь в HL находится адрес последнего байта.
LD A, (HL) ; Значение последнего байта засылается в аккумулятор и
PUSH AF ; отправляется на стек.
DEC HL ; HL=HL-1. теперь в HL находится адрес предпоследнего байта.
LD A, (HL) ; Его значение поступает в аккумулятор.
CPL ; инвертируется и
INC HL ; копируется в последний
LD (HL), A ; байт.

После этого последний байт равен 255-предпоследний.

XOR A ; Сброс флага переноса.

Это необходимо потому, что далее мы применяем команду SBC, ведь простого вычитания SUB для двухбайтных чисел не бывает.

SBC HL, BC ; HL=HL-BC - установка в HL адреса начала компрессируемой области.

Мы могли бы эти операции и упростить, если бы пользовались готовыми значениями адресов, но мы этого не делали, чтобы процедура была универсальной и могла бы применяться и для компрессирования других областей памяти.

L0 LD A, (HL) ; В акк-р идет байт из исходной области
LDI ; и переносится по адресу, на который указывает DE.
DEC C ; это проверка BC
INC C ; на 0 и переход на
JR HZ, L1 ; метку LK (конец),
DEC B ; если 0

```
INC B          ; достигнут.
JR Z, LKON     ;
.....
```

Обычно проверку на ноль делают не так, а проще:

```
LD A, B
OR C
```

В нашем случае этого делать нельзя, т. к. регистр А уже занят и пришлось бы его сохранять на стеке:

```
PUSH AF
LD A, B
OR C
POP AF
```

- но это уже работает не так быстро, как приведенный нами метод.

```
L1  CP (HL)          ; Следующий байт равен предыдущему?
    JR NZ, L0        ; Если нет, то на L0 (копирование).
    LDI              ; Иначе копируем
    PUSH BC          ; еще раз и запоминаем BC на стеке.
    LD BC, 0         ; организуем в BC счетчик повторов.
L3  CP (HL)          ; следующий байт равен предыдущему?
    JR NZ, L2        ; Если нет, то на L2 (восстановление прежнего состояния)
    INC BC           ; Нарастиваем счетчик
    INC HL           ; и переходим к следующему байту.
    JR L3            ; Повторяем проверку
L2  LD A, C          ; записываем в
    LD (DE), A       ; приемник количество
    INC (DE)         ; повторений,
    LD A, B          ; взятое из BC.
    LD (DE), A       ;
    INC DE           ; DE=DE+1. После этого в DE содержится адрес
                    ; первого пустого места в приемнике.
    EX (SP), HL      ; HL сохр. на стеке и из стека установленное
                    ; значение BC восстанавливаем в HL,
    XOR A            ; Сброс флага C.
    SBC HL, BC       ; HL=HL-BC Теперь мы имеем в HL количество байтов,
                    ; которое осталось компрессировать.
    EX (SP, BC)      ; двухходовая переброска
    POP BC           ; из HL в BC через стек с восстановлением HL.
    LD A, B          ; проверка BC на 0
    OR C             ; (на конец работы)
    JR Z, L0         ;
LK  POP AF           ; Восстановление последнего байта и
    LD (DE), A       ; напр. в приемник.
    INC DE           ; DE=DE+1
    PUSH DE          ; Переброска из DE в
    POP BC           ; BC через стек.
    RET              ; Выход.
```

Длина сжатого блока равна содержимому BC минус начальный адрес. Мы специально поместили результат в BC, чтобы из БЕЙСИКа по команде PRINT USR его можно было бы легко выдать на экран.

ПРИМЕР

Рассмотрим конкретный пример. Адрес трансляции программы примем 30000, а адрес хранения компрессированного экрана - 32766.

Адрес	Маш. код	Метка	Мнемоника
7530	21 00 40		LD HL, 16384
7533	01 FF 1A		LD BC, 6911
7536	11 00 80		LD DE, 32768
7539	09		ADD HL, BC
753A	7E		LD A, (HL)
753B	F5		PUSH HL
753C	2B		DEC HL
753D	7E		LD A, (HL)
753E	2F		CPL

753F	23		INC HL
7540	77		LD (HL), A
7541	AF		XOR A
7542	ED 42		SBC HL, BC
7544	7E	L0	LD A, (HL)
7546	ED A0		LDI
7540	0D		DEC C
7549	0C		INC C
754A	20 04		JR NZ, L1
754C	05		DEC B
754D	04		INC B
754E	28 20		JR Z, LK
7550	BE	L1	CP (HL)
7551	20 F2		JR NZ, L0
7553	ED A0		LDI
7554	C5		PUSH BC
7555	01 00 00		LD BC, 00
7556	BE	L3	CP (HL)
7559	20 04		JR HZ, L2
755B	03		INC BC
755C	23		INC HL
755D	18 F9		JR L3
755F	79	L2	LD A, C
7560	12		LD (DE), A
7561	13		INC DE
7562	78		LD A, B
7563	12		LD (DE), A
7564	13		INC DE
7565	E3		EX (SP), HL
7566	AF		XOR A
7567	ED 42		SBC HL, BC
7569	E3		EX (SP), HL
756A	C1		POP BC
756B	78		LD A, B
756C	B1		OR C
756D	20 D5		JR NZ, L0
756F	F1	LK	POP AF
7570	12		LD (DE), A
7571	13		INC DE
7572	C9		RET

Загрузите в экран какую-либо картинку, вызовите из БЕЙСИКа компрессирующую программу PRINT USR 30000 и запомните число, которое получили, его надо будет указывать при декомпрессии.

Зная конечный адрес и начало, можно вычислить и длину:

длина = результат-32768

Теперь можно выгрузить блок кодов: SAVE"имя"CODE 32768,длина.

Процедуру декомпрессии мы дадим в следующем выпуске.

СЕКРЕТЫ ПЗУ

Надо ли объяснять какое значение имеет постоянное запоминающее устройство (ПЗУ) компьютера. В системе Синклер-Спектрум объем ПЗУ составляет 16 килобайт, расположено оно в нижних адресах памяти от адреса 0000 до 3FFF, что в десятиричной системе может быть записано как от 0 до 16383.

Содержимое ПЗУ составляет одна очень сложная и сравнительно крупная программа. Ее называют монитором, и многое в компьютере происходит под ее управлением. Она организует опрос клавиатуры, ввод/вывод программ с магнитофона, управляет выдачей информации на экран и принтер, весь диалог, который Вы в БЕЙСИКе ведете с компьютером, на самом деле ведется с этой программой. Монитор может еще очень и очень многое. Те, кто хорошо знают эту программу, могут использовать ее процедуры в своих программах, что позволяет сокращать время при разработке новых программ и повышать их эффективность.

Мы имеем очень много писем с просьбой рассказать о ПЗУ компьютера, объяснить назначение и особенности работы некоторых входящих в него процедур. Выполняя многочисленные просьбы наших читателей, мы начинаем сегодня печатать с продолжением свою разработку "СЕКРЕТЫ ПЗУ".

Задача, которую мы поставили, выглядит серьезной. Дело в том, что основным первоисточником является книга д-ра Яна Логана и д-ра Фрэнка О'Хары "THE COMPLETE SPECTRUM ROM DISASSEMBLER". Эта книга вполне может считаться библией программиста на "СПЕКТРУМЕ", но ее внушительный объем (порядка 450 машинописных страниц в русскоязычном переводе) делает невозможным для нас ее распространение в рамках разумных цен.

Пытаясь совместить желаемое и возможное, мы сделаем все, что сумеем, чтобы дать Вам в этом году полный обзор системного ПЗУ компьютера, для большинства наиболее употребимых процедур мы дадим не только точки входа, но и условия обращения к ним, некоторые наиболее интересные и поучительные конструкции будут детально разобраны и прокомментированы, а для прочих мы укажем начало, конец и конечно же назначение процедуры.

Программы, содержащиеся в ПЗУ очень широко общаются с областью системных переменных "СПЕКТРУМА". Названия системных переменных при этом мы будем указывать точно так, как это сделано в книге Векерса "Программирование на языке БЕЙСИК для компьютера СИНКЛЕР ZX СПЕКТРУМ", те же названия системных переменных были применены нами в нашей разработке "Большие возможности вашего СПЕКТРУМа".

Поскольку изучая ПЗУ вам необходимо знать основы программирования в машинных кодах и на языке АССЕМБЛЕРА, реальную помощь Вам может оказать наша разработка "Первые шаги в машинных кодах", а если и после этого какие-либо нюансы будут ускользать от вас, то многие программистские приемы освещены в нашей книге "Практикум по программированию в машинных кодах". Эти разработки - базовые. Не сделав их, мы не смели и приближаться к тому, чтобы начать освещение системного ПЗУ компьютера.

Монитор "СПЕКТРУМа" содержит три основные части:

1. Процедуры управления вводом/ выводом.
2. Интерпретатор БЕЙСИКа.
3. Вычислительные процедуры.

Поскольку эти части весьма значительны, мы рассмотрим монитор, условно выделив в нем 10 разделов.

1. Раздел инициализации.
2. Обслуживание клавиатуры.
3. Обслуживание звукового динамика.
4. Работа с магнитофоном.
5. Поддержка экрана и принтера.
6. Процедуры исполнения команд.

7. Интерпретатор команд и БЕЙСИК-строк.
8. Расчет выражений.
9. Арифметические процедуры.
10. Встроенный калькулятор.

1. ПРОЦЕДУРЫ ИНИЦИАЛИЗАЦИИ

0000-0007 - START.

Процедура запускается при включении компьютера в сеть, после нажатия кнопки RESET, после команды RANDOMIZE USR 0. Она выставляет вершину физической памяти компьютера FFFF в регистре DE, обнуляет аккумулятор и делает переход к процедуре START/NEW по адресу 11CB.

0006-000F - ERROR-1.

К этой процедуре происходит обращение, когда интерпретатор БЕЙСИКа фиксирует ошибку. Процедура извлекает адрес возникновения ошибки из системной переменной CH_ADD и выставляет его в указателе (системной переменной) X_PTR, после чего делает переход на процедуру обработки ошибки ERROR-2 по адресу 0053.

0010-0012 - PRINT-A-1.

Точка входа для печати символа на экране или на принтере, при входе код печатаемого символа должен содержаться в регистре A процессора. Процедура не делает ничего, просто передает управление процедуре PRINT-A-2, которая находится по адресу 15F2 и действительно занимается печатью.

0013-0017 - не используются.

0018-001B - GET-CHAR.

Процедура принимает из системной переменной CH_ADD код символа, который в настоящий момент обрабатывается интерпретатором БЕЙСИКа и помещает его в аккумулятор процессора.

001C-001F - TEST-CHAR.

Проверяет, соответствует ли данный код печатному символу или нет. Если да, то происходит возврат в вызывающую процедуру. Если нет, то работа продолжается. Для проверки TEST-CHAR использует обращение к процедуре SKIP-OVER, находящейся в 007D.

0020-0024 - NEXT-CHAR.

Увеличивает CH_ADD на единицу, т.е. переходит к следующему символу БЕЙСИК-строки и снова идет на проверку в TEST-CHAR (001C).

0025-0027 - не используются.

0028-002A - FP-CALC.

Вызов калькулятора.

002B-002F - не используются.

0030-0037 - BC-SPACES.

Предназначена для выделения в рабочей области БЕЙСИКа свободного места заданного размера. Размер предварительно должен быть установлен в регистровой паре BC. Собственно здесь вычисления не производятся. Это только точка входа. Управление передается на процедуру RESERVE (169E).

Широко используется БЕЙСИКОМ, особенно при редактировании, например, когда Вы вставляете пропущенные строки или операторы.

0038-0047 - MASK-INT.

Очень важная процедура. 50 раз в секунду процессор выполняет прерывание своей работы и обращается сюда. При работе в БЕЙСИКе это происходит всегда, а при работе в маш. кодах только если прерывание включено. Здесь происходит увеличение системной переменной FRAMES, которая является как Вы внутренними часами компьютера и начинается выполнение KEY-INT.

0046-0052 - KEY-INT.

Отсюда 50 раз в секунду вызывается процедура сканирования клавиатуры KEYBOARD (02BF) с целью определения была ли нажата какая-либо клавиша, когда часы выставлены и клавиатура проверена, управление вновь передается процессору для продолжения той

работы, которой он занимался до прерывания.

0053-0054 - ERROR-2.

Если БЕЙСИК-интерпретатор встретил ошибку, то ее код отправляется на стек, здесь код ошибки снимается со стека и помещается в регистр L процессора.

0055-005E - ERROR-2.

Код ошибки передается из регистра L в системную переменную ERR_NR, а на стек процессора помещается адрес, определяющий куда надо переходить для обработки ошибки. Он содержится в системной переменной ERR_SP. Возврат выполняется не сразу, а после перехода к процедуре SET-STK (16C5), где очищается стек калькулятора (не путать со стеком процессора, который именуют также машинным стеком).

Дополнительно сообщим, что использование новых нестандартных команд, как например команд МЕГА-БЕЙСИКа, ЛАЗЕР-БЕЙСИКа, БЕТА-БЕЙСИКа и т. п. организовано именно на этой системной переменной ERR_SP. Интерпретатор встречает незнакомый или неожиданный символ, и идет на обработку ошибки по выставленному в ERR_SP адресу, а там его ждут процедуры, которые отличают настоящую ошибку от допустимого расширения, предусмотренного программой.

005F-0065 - не используются.

0066-006F – RESET.

Процедура обработки немаскированного прерывания. Сразу скажем, что в фирменной ПЗУ здесь содержится ошибка.

При получении прерывания по линии INT, процессор обращается к адресу 0066 (это заложено в устройстве процессора), и здесь он должен найти адрес процедуры, которая будет это прерывание обрабатывать.

PUSH AF ; Запомнили на стеке

PUSH HL ; содержимое этих пар.

LD HL, (5CB0) ; В регистр HL загружается содержимое системной переменной, находящейся по указанному адресу – 5C80.

Не ищите эту системную переменную в литературе. Поскольку данная процедура не работает из-за содержащейся в ней ошибки, то и про эту системную переменную нигде не пишут, а говорят, что адрес 5CB0 (23728) якобы не используется. Должна же эта переменная называться NMIADD и содержать адрес, по которому находится программа, обрабатывающая немаскируемое прерывание.

LD A,H ; Проверка содержимого

OR L ; HL на 0.

JR NZ, 0070 ; ОШИБКА! Здесь должно быть JR Z,0070. Должно быть так, что если в NMIADD стоит 0, то переход в 0070 и оттуда возврат, ничего не сделав. А если не 0, то переход по адресу из NMIADD.

JP (HL) ; На самом же деле все происходит наоборот: если 0, то происходит переход по адресу 0, т.е. системный сброс компьютера, а если не 0, то возврат, ничего не сделав.

0070-0073 - NORESET.

POP HL ; Восстановление содержимого

POP AF ; этих пар.

RETN ; Возврат после обработки немаскированного прерывания.

0074-007C - CH_ADD+1.

Процедура выполняет переход к следующему интерпретируемому символу, увеличивает на единицу системную переменную CH_ADD и вводит новый символ в аккумулятор,

007D-0094 - SKIP-OVER.

При входе сюда в регистре A содержится код символа, который сейчас обрабатывает интерпретатор БЕЙСИКа. Проверяется является ли он нормальным печатным символом или управляющим кодом, в первом случае - возврат, а во втором код анализируется и производится переход к очередному символу или даже через один с выставлением флага переноса или без. Все зависит от того, какой символ был найден.

0095-0204 - таблицы токенов.

Вы, конечно знаете, что ключевые слова в БЕЙСИКе на "СПЕКТРУМЕ" набираются не по буквам, а закреплены за своими клавишами. Здесь в ПЗУ находятся токены этих ключевых слов. Чтобы отделять их друг от друга последний символ каждого токена инвертирован, включением седьмого бита, иначе говоря, к коду последнего символа

прибавлено число 128.

0205-026B - таблицы клавиатуры.

Этих таблиц - шесть для разных режимов, в которых может находиться клавиатура. Они определяют какой символ, токен или управляющий код закреплен за данной клавишей.

Более подробно:

0205-022B - когда клавиатура находится в режимах L или C.

022C-0245 - клавиатура в режиме E. Для буквенных клавиш без нажатия CAPS SHIFT.

0246-02SF - клавиатура в режиме E. Для буквенные клавиш с нажатой клавишей CAPS SHIFT.

0260-0269 - управляющие коды, закрепленные за цифровыми клавишами при нажатой клавише CAPS SHIFT.

026A-0283 - символы и токены, закрепленные за клавишами при нажатии клавиш вместе с SYMBOL SHIFT.

0284-026B - клавиатура в режиме E. Для цифровых клавиш с нажатой клавишей SYMBOL SHIFT.

МАЛЕНЬКИЕ ХИТРОСТИ

Если у Вас еще небольшой опыт работы со "СПЕКТРУМом", то эти маленькие хитрости могут Вам доставить удовольствие. Считайте их головоломками. Давайте сделаем так. Сначала попробуйте угадать, что будет напечатано по команде PRINT X в приведенных примерах. Потом проверьте себя на компьютере и попробуйте объяснить почему так происходит, если не сумеете, не беда. В следующем выпуске мы дадим объяснение и расскажем как все это можно применить в программах.

I.
10 LET a=5: LET b=8
20 LET x=10
30 LET x=x+(a>b)
40 PRINT x

II.
10 LET a=15: LET b=8
20 LET x=10
30 LET x=x+(a>b)
40 PRINT x

III.
10 LET a=13: LET b=8
20 LET x=a=b
30 PRINT x

IV.
10 LET x=0
20 LET x= NOT NOT x
30 PRINT x

V.
10 LET x=1
20 LET x= NOT NOT x
30 PRINT x

VI.
10 LET x=10
20 LET x= NOT NOT x
30 PRINT x

VII.
10 LET a=5: LET b = 8
20 LET x=(a=b)=1
30 PRINT x

VIII.
10 LET a=5: LET b=8
20 LET x=(a=b)=0
30 PRINT x

ПОДПИСКА ПРОДОЛЖАЕТСЯ

Дорогие друзья! Вы только что прочитали основной выпуск нашего "Персонального письма" любителям бытового персонального компьютера "ZX-Spectrum".

Надеемся, что сейчас у Вас есть четкое представление о его целях, задачах и содержании.

Кроме основного выпуска мы предлагаем Вашему вниманию также и три приложения: "Игровое Приложение", "Техническое приложение" и "Рекламное Приложение".

Если Вы подписались не на полный комплект, то имеете возможность произвести доплату до стоимости полного комплекта и в самые кратчайшие сроки начать получать "ZX-РЕВЮ" полностью, напоминаем, что цена получения полного комплекта в течение года составляет 70 (семьдесят) руб. в ценах 1990 года. Доплата производится на наш расчетный счет. Одновременно просим выслать нам сообщение о произведенной доплате с указанием Вашего регистрационного номера, это позволит Вам получать полный комплект уже со второго выпуска. Идя навстречу многочисленным просьбам читателей, желающим комплектовать свои подшивки из наших выпусков мы будем делать сквозную нумерацию страниц в ZX-РЕВЮ как по всем приложениям, так и по всем выпускам, т.е. например первая страница основного выпуска за февраль месяц будет иметь номер 23 и т. д.

Если Вы не являетесь зарегистрированным получателем ZX-РЕВЮ, но желаете войти в их ряды, то можете сделать это в любой удобный для Вас момент, но теперь мы принимаем заказы только на полный комплект. Вы получите все выпуски, вышедшие до того, как Вы сделали заказ и будете ежемесячно получать по новому выпуску в течение всего года.

Своих постоянных заказчиков "ИНФОРКОМ" будет также регулярно оповещать об имеющихся и вновь сделанных разработках.

Заказы направляйте по адресу 107241, Москва, Б-241, а/я 37 "ИНФОРКОМ". Здесь же укажите когда и откуда был сделан перевод.

Для организаций при оплате по безналичному расчету действует повышенный тариф - 252 руб. за полный годовой комплект.

Этот тариф, не распространяется на школы, дома Пионеров и т.п. организации, для них тариф - 70 р.

Оплату для организаций и индивидуальных заказчиков производить по адресу: Фрунзенское отделение Жилсоцбанка г. Москвы МФО 201412 Р/С 500461778, кооператив "ИНФОРКОМ".

ЖАНРЫ ИГРОВЫХ ПРОГРАММ.

Богатейшее игровое программное обеспечение для ПК "ZX-Spectrum" дает обширный простор для анализа и классификации игровых программ.

Мы не будем выдумывать ничего нового, классификация давно существует. Она была поддержана такими крупными журналами как "SINCLAIR USER" и стала традиционной. Для отечественных любителей игровых программ этот вопрос до сих пор оставался закрытым. Наши журналы обходят игровые программы, сторонясь их как заразы. В лучшем случае упоминают о том, что вроде бы и среди этих программ тоже бывают невредные, развивающие реакцию и сообразительность.

Надо честно сказать, что все выдающиеся программисты, завоевавшие известность в восьмидесятые годы, начинали с игровых программ. В своих статьях они пишут, что и сейчас не прочь покопаться в машинном коде других авторов, от души аплодируя удачным находкам.

Итак, если начать рассматривать жанры игровых программ, нам надо сначала отправиться в недалекое прошлое.

ADVENTURE.

Представьте себе вычислительный центр середины семидесятых годов, огромные блоки дорогостоящей аппаратуры, трудные ночные часы дежурств (ведь машины дорогие и должны работать 24 часа в сутки). Время тянется медленно, а спать нельзя.

В этих условиях и появились на свет первые игровые программы. Сначала их писали подпольно, по ночам, когда начальство не видит. Машины не имели графических дисплеев, и программы были чисто текстовыми. По имени первой игровой коммерческой программы этого жанра ("Adventure"- "Приключение") и было названо все направление.

Перевести на русский язык этот жанр как "приключенческий" мы не можем, да и не имеем права, уж очень не похожи эти игры на "приключенческие", поэтому за ними так и закрепилось название "адвентюры".

Итак, это текстовые игры. В некоторых случаях их оснащают графикой (Worm In Paradise-фирма LEVEL NINE), а иногда и очень высококачественной графикой (Kayleth - фирма ADVENTURE SOFT, Aftershock - фирма INTERCEPTOR), но графика здесь роли не играет, часто мешает углубиться в содержание игры. Главное здесь диалог. Чем лучше программа, тем лучше она Вас понимает, тем интереснее с ней общаться. Чем мудрее Вы, тем быстрее Вы находите, что программа понимает, а что нет, тем быстрее у Вас налаживается контакт. А длиться он может долгие месяцы. Хорошие адвентюрные программы имеют многие сотни и даже тысячи пунктов, в которых Вы можете побывать.

Многочисленные опросы общественного мнения показали, что если в первые два года работы с компьютером пользователи редко отдают предпочтение этому жанру, то уже после третьего года эти игры вместе со стратегическими становятся основными. Они подкупают неисчерпаемостью сюжетов, романтическим духом, богатством фантазии, на базе одного научно-фантастического рассказа можно сделать игру продолжительностью в год. Еще одно преимущество у этого жанра в том, что эти игры сближают людей, трудно удержаться, чтобы не поделиться своим открытием с друзьями, когда Вы обнаружите, например, что стоит со старого сундука стереть пыль, как он отодвинется и откроет подземный ход. Точно так же трудно долго пребывать в безвыходном положении и не иметь во-время нужной подсказки от друга. Поэтому в эти игры часто играют подолгу и коллективно (причем сразу в несколько), периодически обмениваясь открытиями.

Мы хорошо знаем, что среди широких масс в нашей стране эти игры не пользуются авторитетом, и причина здесь в кажущемся плохом знании языка. Это предрассудок. На самом деле дело не в языке, а в незнании элементарных принципов этих игр. Ведь есть простейшие приемы, которые позволяют их освоить, вам достаточно тридцати английских слов, чтобы начать освоение, а половину из них Вы и так знаете, даже если никогда не изучали английский, например (North, East, South, West, Up, Down, Press...) а дальше

развитие придет в игре. Мы сознательно уделяем этому жанру особое внимание, потому, что он учит:

- активному владению языком;
- правописанию слов;
- овладению клавиатурой, а это пригодится многим;
- умению мыслить логически;
- умению мыслить нестандартно.

Мы беремся, уделяя на одной странице игрового приложения ежемесячно, в течение полугода обучить Вас всем приемам и хитростям которые знаем сами, и вскоре в нашем "ZX-РЕВЮ" может появиться уголок, в котором Вы сможете обмениваться своими достижениями.

Достойнейшими представителями игр этого жанра во всем мире считаются игры фирмы LEVEL NINE (мы конечно говорим только о "СПЕКТРУМе").

ARCADE

В начале восьмидесятых появились первые персональные компьютеры, оснащенные сначала черно-белыми, а потом и цветными дисплеями. И сразу у программистов, пишущих игровые программы, встала задача как передать детали и нюансы нашей повседневной жизни в игровой экран. Как изобразить дождь? А если дождь радиоактивный? Как передать инопланетян, желающих захватить нашу планету? Причем имеется в виду не только как передать их облик, но и как передать их поведение?

Так появились первые аркадные игры. Аркадные значит идиллические. Здесь в идеализированном виде представляются взаимоотношения людей и природы и т.п. Вспомните Manic Miner и Jet Set Willy, Saboteur и Bruce Lee - это все аркадные игры.

Фирм, пишущих программы этого жанра великое количество, у нас же широко известны IMAGINE, OCEAN, GREMLIN GRAPHICS.

ARCADE/ADVENTURE

Два основных наиболее популярных жанра породили третий - аркадные адвентюры. которые во многом превзошли своих предшественников.

От аркадных игр было взято управление героями на экране, а от адвентюр - использование разнообразных предметов, которые могут быть найдены по ходу игры для достижения конечной цели и взаимодействие с другими персонажами. Обратите внимание, что предметы служат не просто, чтобы их брать и получать за это очки, а именно чтобы в одной месте их взять, а в другом применить. Часто разные предметы служат разным целям. Иногда надо иметь комбинацию предметов, чтобы достичь результата (чтобы сварить уху, надо иметь и костер, и котелок, и воду и рыбу в одно время и в одном месте).

Первой программой этого жанра была программа Attic Atac (1983) а первооткрывателем стала фирма ULTIMATE. Успех превзошел все ожидания. Открытием стало не только создание нового жанра (тогда об этом еще не знали), но и введение псевдо-трехмерной графики. Далее последовали Underground (1984), Sabre Wulf (1984) и Knight Lore (1984). Момент появления Knight Lore стал звездным часом фирмы. Удивительно, но факт - трехмерный формат изображения, введенный в этой игре, стал неофициальным стандартом, и то, что мы видим в более поздних играх других фирм (Batman(OCEAN), Movie (IMAGINE), Head Over Heels (OCEAN) , Sweevo's World (GARGOYLE GAMES), Prodigy (ELECTRIC DREAMS) и др.), - развитие плодотворной идеи, впервые открытой фирмой ULTIMATE.

Развивая идею Knight Lore, фирма ULTIMATE выпустила еще Alien 8 (1985) и Pentagram(1986), но увы достичь такого успеха ей больше не удалось.

У нас аркадно-адвентюрные игры давно пользуются заслуженным признанием, особенно игры фирмы MICRO-GEN Pijamarama, Everyone's a Wally, Three Weeks In Paradize, Herbert Dummy Run.

Нельзя не отметить и такие захватывающие игры как Dizzy и Dizzy 2 фирмы CODEMASTERS, Sceptre Of Bagdad (ATLANTIS), Firelord (HEWSON CONSULTANTS).

В нашей стране распространены несколько сот игр этого жанра, и как правило это

высококачественные программы.

ACTION

Дословно на русский язык название этого жанра наверное лучше не переводить, но к нему хорошо подходит синоним - боевик. Из названия ясно - содержание этих игр - побегать, полетать, пострелять во все, что движется. Цель проста а средства ее исполнения могут иногда требовать огромной реакции и выдержки. Как правило, игры этого жанра самостоятельно не существуют. Хотя какой-то, пусть самый примитивный сюжет игра имеет, поэтому почти во всех случаях эти игры можно характеризовать как ARCADE/ACTION.

Если нужны примеры - Commando фирмы ELITE, Green Beret фирмы IMAGINE (оригинал - фирмы KONAMI для компьютеров системы MSX), Uridium - HEWSON CONSULTANTS и многие сотни других.

TRADITIONALS.

Очевидным шагом в дальнейшем развитии игровых программ стало привлечение в их орбиту традиционных игр. Таких как шахматы, шашки, нарды, бильярд, рулетка, многочисленных карточных игр и т.п. О сути этого жанра говорить не нужно, она ясна.

В "ZX-РЕВЮ" мы уделим особое внимание этим играм, поскольку некоторые из них, особенно шахматные программы требуют наличия специальной инструкции для полноценной работы.

SIMULATIONS

Имитаторы появились не сразу, а только по достижении определенного уровня в технике программирования, поскольку для создания хорошего имитатора, моделирующего управление сложным техническим объектом, необходимо определенное быстроедействие программы.

На сегодняшний день в "Спектруме" охвачены имитаторами наверное уже все мыслимые транспортные средства - самолеты пассажирские (Flight Simulation - PSION) и боевые (Fighter Pilot - DIGITAL INTEGRATION), вертолеты (Tomahawk - DIGITAL INTEGRATION), лодки моторные (Power Boat simulation - MASTERTRONIC) и подводные (Hunter Killer - PROTEC, Silent Service - MICROPROSE), воздушные шары, яхты и даже паровозы (Southern Belle - HEWSON CONSULTANTS). О многочисленных имитаторах автомобилей мы даже и не говорим. Им посвящен наш отдельный альбом описаний "HURRY UP", - см. рекламный листок "ИНФОРКОМ предлагает...".

Полезность программ этого жанра для молодежи трудно оспаривать. В то же время, как нам известно, многие сталкиваются с трудностями при их освоении. Трудности вызваны отсутствием хорошей документации по управлению программой. Мы уделим особое внимание программам - имитаторам, и в каждом номере игрового приложения к "ZX - ревью" будем подробно описывать хотя бы одну.

Несколько в стороне стоят спортивные имитаторы. Это и имитаторы отдельных видов спорта и имитаторы спортивных игр. Программы очень неровны по уровню их исполнения. Мало кому нравится стучать по клавишам в беге на стометровку в программах Decathlon и Supertest фирмы OCEAN зато как здорово сделана стрельба из лука и пистолета.

Практически никуда не годятся многочисленные имитаторы бокса типа Frank Bruno Boxing - ELITE, но пользуются огромной популярностью имитаторы карате Fist, International Karate, Sai Combat и многие другие.

Многочисленные имитаторы спортивных игр типа футбола, волейбола, баскетбола, регби, крикета, бейсбола, хоккея и др. скорее служат пополнению коллекции, чем увлекательному отдыху. Изящество техники программирования, к сожалению в них не материализовалось в изящество игры, и мало кто общается с ними более одного раза.

STRATEGY

К этому жанру относятся обычно программы, требующие от играющего

стратегического мышления в управлении войсками. Наиболее ранней игрой этого жанра является Battle 1917 фирмы CCS, написанная на Бейсике. В последние годы наибольших успехов здесь добилась фирма LOTHLORIEN, которой принадлежат такие программы как Austerlitz, Confrontation, Overlords и многие другие. Продолжает работать и фирма CCS - Desert Rats и др. Очень интересна на своих высших уровнях сложности программа CONQUEST фирмы SHEETANSOFT.

Как правило, эти программы не требуют особой сноровки. Действие развивается неспешно.

Выше ценится талант полководца чем быстрота реакции, но есть некоторые игры, в которых кроме стратегического планирования и управления войсками Вам предлагается также и пострелять, например Beach Head фирмы US GOLD, в зарубежных журналах такие программы относятся к жанру STRATEGY/ WAR.

BUSINESS/MANAGEMENT

Это деловые игры, они имитируют реальную ситуацию, в которой может оказаться государственный деятель (Dictator - DK'TRONICS, President - ADDICTIVE GAMES), бизнесмен (Millionaire - INCENTIVE), управляющий, например менеджер футбольной команды (Football Manager 1 и 2 - ADDICTIVE GAMES), менеджер команды автогонщиков (Formula One Manager - CRL) и другие.

Успех в этих программах, также как и в стратегических зависит от умения правильно анализировать ситуацию, принимать верные решения. Кроме развития навыков менеджмента эти игры улучшают также и знание английского языка, поскольку значительная часть информации выдается на экран в текстовом виде.

В последние несколько лет этот жанр сильно обогатился за счет программ с детективным содержанием (Mugsy и Mugsy Revenge - MELBOURNE HOUSE, Murder Miami - CRL, The Fourth Protocol 1,2 и 3 - Century/Hutchinson и многие другие). Сейчас в них все более широко применяются иконно-оконный метод диалога. Например, если Вы хотите позвонить по телефону, то указываете курсором на "иконку" с изображением телефона, а если рассмотреть подозрительный предмет поближе, то на изображение увеличительного стекла, в ответ программа высвечивает "окно", в котором Вы можете выбрать нужное Вам действие более конкретно, например: "Позвонить полковнику".

PUZZLE

Игры - головоломки, в первую очередь к ним относятся всевозможные двумерные и трехмерные лабиринты. Конечно лабиринт - неперенный атрибут большинства аркадных и адвентюрных программ, но разница есть. Здесь лабиринт существует не как элемент игры, а как сама игра. Выйти из него - единственная задача. Из наиболее совершенных по графическому исполнению стоит отметить Bobby Bearing фирмы EDGE, Triaxos фирмы ARIOLASOFT.

Кроме лабиринтов к головоломкам можно отнести и разного рода задачи на комбинаторику.

В последние годы появилась целая серия программ с неплохой трехмерной графикой, сделанных по типу Marble Madness фирмы ATARI-SOFT. Отметим в качестве оригинальной позиционной головоломки программу Kirel фирмы ADDICTIVE GAMES.

Мы осветили в этом кратком обзоре все основные жанры игровых программ. Сразу оговоримся, что мы не претендуем на истину в последней инстанции. Никаких утвержденных стандартов нет, и мы выражали только наше субъективное мнение. Если у Вас есть другое, оно имеет такое же право на существование.

Использование единой терминологии позволит упростить взаимопонимание всех любителей Синклер-компьютеров нашей страны.

Отметим, что многие, в том числе как правило самые лучшие программы никогда не втиснуть в рамки одного жанра. Попробуйте, например, дать точное определение лучшей программе десятилетия Elite фирмы FIREBIRD. У Вас получится нечто вроде: - ARCADE/ADVENTURE/ACTION/STRATEGY/ BUSINESS.

Впрочем, об этой уникальной программе мы еще поговорим на страницах последующих выпусков "ZX-РЕВЮ".

Внимание!

Мы начинаем давать подробные описания работы с некоторыми игровыми программами. В основу их выбора положены два принципа. Во-первых, это программы достаточно сложные для самостоятельного освоения без специальных знаний, а во-вторых, это программы достаточно распространенные, чтобы Вы могли их найти своими силами, или они у Вас уже есть.

Тем не менее, мы имеем в виду возможность выпуска в будущем сборника программ на кассете С-90, описания к которым давались в ZX-РЕВЮ. Безусловно, зарегистрированные получатели ZX-РЕВЮ будут при этом иметь ощутимые, а может быть и исключительные льготы.

FIGHTER PILOT

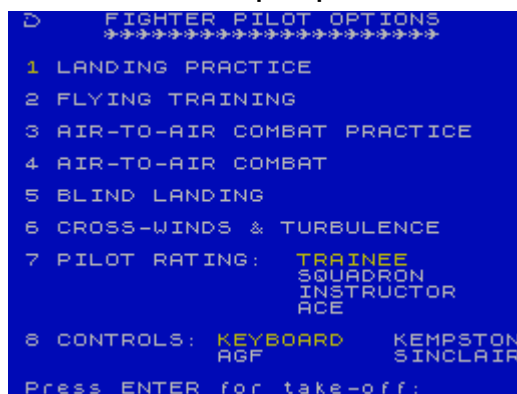


Начиная с программы "FIGHTER PILOT", мы подчеркиваем, что это одна из тех программ, которые стали классическими. Техника управления самолетом, реализованная здесь, часто используется и в других аналогичных программах, поэтому освоив одну эту программу, Вы сделаете шаг вперед по освоению целого направления.

"FIGHTER PILOT" - это имитатор в режиме реального времени, на базе истребителя F-15 ("EAGLE").

Программа представляет наиболее полный набор характеристик, необходимых для авиационных имитаторов, включая сюда трехмерный обзор из кабины пилота, полную имитацию маневров самолета, воздушный бой, порывы ветра, воздушные ямы и слепую посадку. Для каждого из режимов программа предоставляет возможность тренировки, а также позволяет выбирать уровень сложности.

Режимы программы.



1. LANDING PRACTICE - отработка посадки.

Ваш самолет находится в 6 милях от взлетно-посадочной полосы (ВПП) базы (BASE). Шасси выпущено и готово к посадке. Используя регулятор тяги, закрылки и элероны выберите требуемую скорость полета и скорость снижения. Наведение самолета на ВПП может выполняться с помощью системы инструментальной посадки (ILS) или с помощью

бортового компьютера (FLIGHT COMPUTER). После посадки сбросьте тягу до нуля и включите тормоза.

2. FLYING TRAINING - отработка полета.

Исходное положение - самолет на ВПП базы. Взлетайте, для этого дайте полную тягу и по достижении скорости отрыва переложите элероны на подъем (кл. "6"). Максимальное ускорение при взлете можно достичь, если развивать тягу с включенными тормозами, и стартовать отключением тормозов. Сразу после отрыва убирайте шасси. С выпущенными закрылками скорость отрыва меньше, маневрирование на ВПП выполняется с помощью хвостового руля.

3. AIR TO AIR COMBAT PRACTICE - отработка воздушного боя.

Исходное положение на расстоянии двух миль от вражеского самолета, сзади, на той же высоте, выберите боевой режим (COMBAT MODE) и включите бортовой компьютер. Он выдает Вам данные о цели. Противник летит со скоростью 550 узлов и не отстреливается. Маневрируйте, когда увидите противника и стреляйте, когда он проходит через прицел.

4. AIR TO AIR COMBAT - воздушный бой.

В этом Режиме вам поручается охрана четырех аэродромов BASE, TANGO, DELTA и ZULU от вражеских самолетов. Ваша миссия начинается со старта с базы BASE, используйте радар и бортовой компьютер для определения положения вражеского самолета. После выявления цели выводите свой самолет на перехват. Визуальный контакт возможен при расстоянии между вами не более 1 мили и разнице высот не более 5000 футов. Бой начинается с того, что противник начинает маневрировать, повреждения вашего самолета изображаются изменением цвета символа самолета, изображенного на экране радара. Допускаются только три попадания в Ваш самолет. Если Вы хотите выйти из боя вследствие многочисленных повреждений или полного расхода боеприпасов, то уводите свой самолет. Противник перестает атаковать, если расстояние между вами превышает 1 милю или разница высот 5000 футов и вернется к своей главной задаче - налету на аэродромы. Теперь Вы можете вернуться на ближайший аэродром для ремонта, дозаправки и пополнения боекомплекта.

5. BLIND LANDING - слепая посадка

Этот режим имитирует посадку и взлет в тумане. На экране нет изображения, пока самолет находится выше 50 футов. Горизонт вообще не виден. Ориентируйтесь с помощью радара, бортового компьютера и карты.

6. CROSSWINDS AND TURBULENCE - порывы ветра и воздушные ямы.

В этом режиме самолет получает случайные аэродинамические возмущения. Полет и навигация усложняются. Этот режим рекомендуется опытным пилотам.

7. PILOT RATING - уровень сложности.

Вы можете выбрать уровень от новичка до аса. Он не влияет на управление самолетом, но влияет на поведение самолета-соперника.

8. CONTROLS - управление игрой.

Можно задать тип джойстика, которым Вы работаете. AGF соответствует клавишам 5,6,7,8 и 0.

Управление самолетом.

В полете самолет управляется элеваторами, элеронами и хвостовым рулем. Элеваторы управляются клавишами "6" ("вверх") и "7" ("вниз"). Элероны - "5" и "8" вызывают крен самолета влево или вправо. Управление хвостовым рулем - "CAPS SHIFT" и "7".

Во время полета действие этих органов управления может меняться.

Скорость поворота, крена и пр. изменяются в соответствии с длительностью нажатия клавиши, что дает максимальное приближение к реальным условиям полета.

Управление тягой - "Q" и "A".

"Q" - увеличение тяги и "A" - уменьшение, изменение тяги влияет не только на скорость самолета, но и на угол тангажа, величина тяги, необходимая чтобы удерживать заданную скорость, зависит от высоты и от угла тангажа. Так, например, при посадке на малых скоростях, чтобы сохранить подъемную силу, нос самолета должен смотреть вверх,

от этого увеличивается лобовое сопротивление и тяга должна быть повышена.

Максимальная скорость возрастает с высотой, т. к. там плотность воздуха ниже.

Управление закрылками - "W" ("вверх") и "S" ("вниз"). Управление закрылками влияет также на угол тангажа. Выпуск закрылков при скорости выше 472 узлов приводит к их разрушению.

Шасси поднимается и опускается клавишей "U".

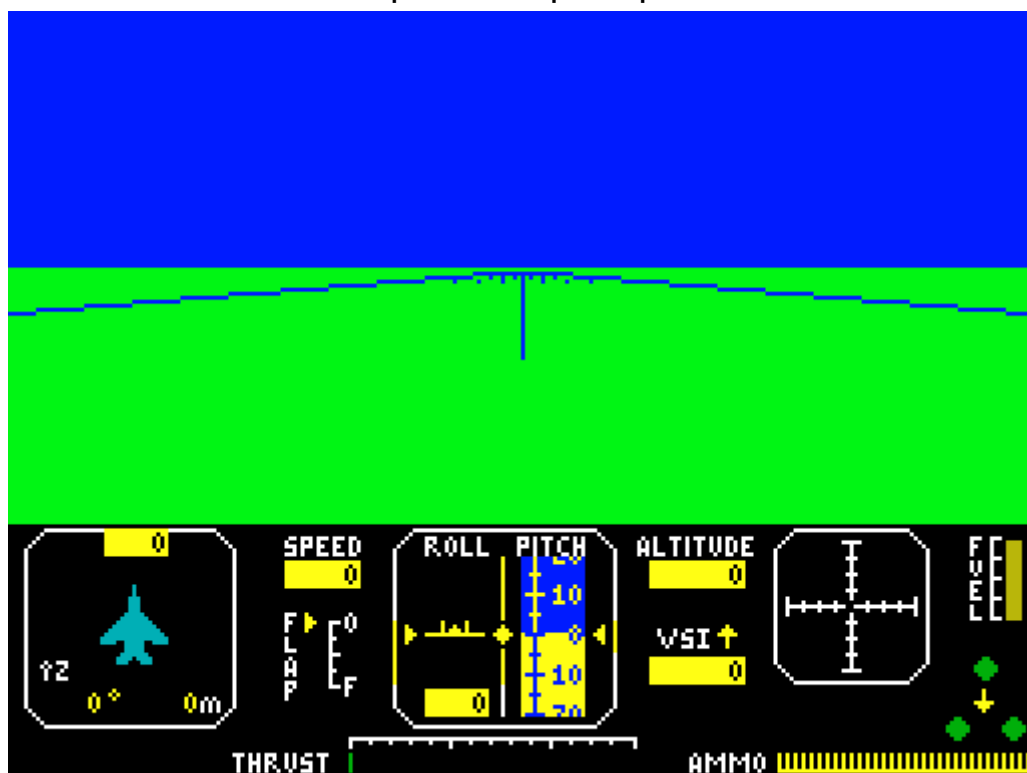
Тормоза - "B". Они включены, когда клавиша нажата, при этом на приборной панели горит надпись "BRAKES".

Огонь из пушки - клавишей "O" с клавишей "C".

Состояние боезапаса изображается в правом нижнем углу дисплея. Три светящихся символа над индикатором боезапаса указывают на наличие в воздухе самолетов противника. Число справа - количество сбитых самолетов.

Замечание: Управление истребителем - это довольно сложная задача, которая требует практики, особенно в боевом режиме.

Бортовые приборы.



Авиагоризонт - расположен в центре приборной панели и показывает углы крена и тангажа Вашего самолета. Широко используется при выполнении сложных маневров самолета, когда Вы теряете из виду горизонт. Угол крена показывается наклоном "символа" самолета, а угол тангажа - на "движущейся ленте". Лента имеет голубой цвет для положительных углов (нос вверх) и желтый для отрицательных.

Скорость, слева от авиагоризонта даются показания скорости высота - дается относительно земли в футах.

Индикатор скорости подъема (VSI) - дает показания скорости набора высоты или снижения в фут/сек. Стрелка показывает вверх когда самолет идет вверх и наоборот. При посадке скорость снижения должна быть примерно 20 футов в секунду.

Закрылки (FLAPS) - их угол выхода меняется от 0 до полного (FULL). С убранными закрылками минимальная скорость горизонтального полета - 130 узлов, с выпущенными 120.

Тяга (THRUST) - индикатор в виде полосы в нижней части приборной панели.

Локатор и компас - эти приборы находятся на левом краю панели. Здесь же изображается отсчет Вашего курса относительно радиомаяка. При смене маяка (кл. "N") меняется и отсчет.

Боевой режим. Переход в него клавишей "C". При этом включается система

прицеливания, задействуются пушки, локатор и бортовой компьютер переключаются на самолет противника. Этот режим индицируется на экране вашего радара светящимся символом. Дистанция и курс теперь относятся к самолету противника, чтобы узнать его высоту, включите бортовой компьютер.

Система инструментальной посадки/бортовой компьютер - находится справа от датчика высоты. Это прибор двойного назначения.

Применяется при посадке и в воздушном бою, переключение между режимами выполняется клавишей "SYMBOL SHIFT".

а) Система инструментальной посадки.

Помогает наводить самолет на ВПП при посадке. Чтобы спускаться по правильной траектории, надо держать светящийся квадрат в центре прибора, это даст угол снижения 3 градуса.

б) Бортовой компьютер. Задействуется клавишей "SYMBOL SHIFT"

Изображается точный отсчет в футах относительно любой ВПП, имеющей радиомаяк в радиусе 10 миль. Расстояния относятся к тому радиомаяку, который изображается в данный момент на Вашем локаторе, Если ВПП была разрушена противником или находится слишком далеко, то компьютер не работает и показывает желтые и черные полосы.

В боевом режиме компьютер выдает высоту самолета противника.

В бою старайтесь держать высоту, равную высоте противника.

Топливо (FUEL) - здесь изображается количество оставшегося топлива.

Шасси (UNDERCARRIAGE) - индикатор находится ниже датчика топлива. Он показывает три красных точки и стрелку вверх, когда шасси поднято и три зеленых и стрелку вниз, когда шасси выпущено.

Карта. Включается и выключается клавишей "M".

СВОДКА КОМАНД

5 - "влево"

6 - "вверх"

7 - "вниз"

8 - "вправо"

0 - "огонь" (работает только в боевом режиме)

CAPS SHIFT - руль влево;

Z - руль вправо;

Q - увеличить тягу;

A - уменьшить тягу;

W - закрылки убрать;

S - закрылки выпустить;

U - управление шасси;

B - включение тормозов;

N - переход на другой радиомаяк;

M - карта;

SYMBOL SHIFT - выбор системы инструментальной посадки или бортового компьютера

C - перевод в боевой режим;

H - пауза;

J - отбой паузы;

XVII - возврат в меню.

ПОЛЕТНЫЕ ДАННЫЕ.

Скорость отрыва:

- с выпущенными закрылками: 130 узлов;

- с убранными закрылками: 140 узлов.

Минимальная скорость горизонтального полета:

- с выпущенными закрылками: 120 узлов;

- с убранными закрылками: 130 узлов.

Максимальная допустимая скорость для закрылков:

- полностью выпущены - 352 узла;
 - промежуточные положения - 472 узла.
- Максимальная допустимая скорость для шасси:
- в полете - 300 узлов;
 - на земле - 250 узлов.

Характеристики:

Максимальная скорость:

- на уровне моря - 802 узла;
- на высоте 60000 футов - 1439 узлов.

Максимальная высота - 65000 футов.

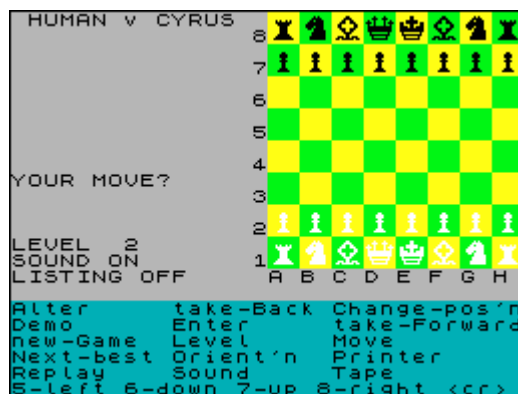
Динамика посадки:

Тяга	74%	62%
Закрылки	+	-
Шасси	+	-
Угол тангажа	+3 град.	+6 град.
Скорость снижения	-9 фут/сек	-12 фут/сек
Скорость	125 узл.	135 узл.

IS CHESS 48K

После загрузки программы перед Вами высвечивается следующая информация:

1. Шахматная доска с исходной позицией.
2. Играющие: человек против компьютера (HUMAN V. CYRUS)
3. Исходная установка:
 - уровень игры - 2 (LEVEL 2)
 - звук включен (SOUND ON)
 - принтер отключен (LISTING OFF)
4. Главное меню.
5. Вопрос "Ваш ход?" (YOUR MOVE?).



Одна из клеток доски мигает. Это курсор.

С помощью курсорных клавиш (5 -влево, 6 - вниз, 7 - вверх, 8 -вправо) установить курсор на ту фигуру, которой Вы будете ходить. Нажать клавишу "ENTER". Поместить курсор на то поле, на которое предполагается сделать ход. Нажать "ENTER".

Если Ваш ход сделан не по правилам, появится надпись "ILLEGAL MOVE", и ход надо повторить. Рассмотрим опции главного меню:

A - ALTER изменение цвета полей и фигур (если у Вас черно-белый экран, то Вы можете менять оттенки серого цвета).

D - DEMO - демонстрационная игра.

G - NEW GAME начать новую игру.

N - NEXT BEST дать другой ход (наилучший из рассмотренных).

R - REPLAY - повторить партию.

B - TAKE BACK - взять ход назад (или несколько ходов назад).

E - ENTER - переключение на игру "человек против человека" (HUMAN V HUMAN).

L - LEVEL - изменение уровня игры.

O - ORIENT'N - изменение ориентации доски.

S - SOUND - вкл./выкл. звука.

C - CHANGE POSITION - применяется при расстановке заранее заданной позиции.

F - TAKE FORWARD MOVE - восстановить взятый назад ход или несколько ходов.

P - PRINTER - печать на принтере.

T - TAPE - запись позиции на магнитофон или загрузка отложенной позиции с магнитофона.

Рассмотрим более подробно содержание некоторых опции.

1. A - изменение цвета. Вам предлагается последовательно выбрать:

- цвет черных фигур?

- цвет белых фигур?

- цвет черных полей?

- цвет белых полей? Для этого Вы можете в ответ на каждый из этих запросов нажать клавиши от 0 до 7:

0 - черный

1 - синий

2 - красный

3 - пурпурный

4 - зеленый

5 - синий

6 - желтый

7 - белый

Или:

8 - оставить цвет без изменения:

D - вернуть исходные цвета, любая другая клавиша - возврат к главному меню.

2. D - демонстрация.

В этом случае играет компьютер против компьютера (CYRUS V CYRUS). Для прекращения демонстрационной игры нажмите любую клавишу.

3. G - начать новую игру.

Эта команда требует подтверждения.

"Y" - да

"N" - нет (если Вы набрали команду "G" по ошибке).

4. R - переигровка. Для прекращения нажмите любую клавишу.

5. B - взятие хода назад. Многократный нажатием клавиши можно взять назад несколько ходов. Обратная команда - F.

6. L - установка уровня. Всего 3 уровней игры от 1 до 8 и 3 уровня для решения комбинационных задач и анализа позиций от P1 до P3.

7. C - расстановка (изменение позиции).

По этой команде вызывается дополнительное меню.

A - ALL CLEAR - убрать все фигуры с доски.

O - ORIENT'N ориентация доски.

T - переключатель цвета расставляемых фигур. WHITE белые. BLACK черные.

E - Выход из этого меню. Возврат в главное. Эта опция не срабатывает, если на доске расставлена позиция, не соответствующая правилам игры или не расставлено никакой (по крайней мере должны быть черный и белый короли).

S - включение/выключение звука

G - начать новую игру (требуется подтверждения "Y").

U - удалить данную фигуру.

Расстановка выполняется так:

Клавишей "Т" выбираете цвет расставляемых фигур.

Затем подведите курсор к клетке, где хотите установить фигуру и нажмите нужную клавишу:

слон - В ладья - R

ферзь - Q король - К

Закончив расстановку белых фигур, переключитесь (Т) на черные и расставьте их, затем (Е) перейдите в главное меню. Стирайте неправильно установленные фигуры клавишей U или А (удаление всех фигур с доски).

(работа с магнитофоном).

S - записать на магнитофон.

L - загрузить с магнитофона.

Любая другая клавиша - возврат в главное меню.

Прервать запись или загрузку можно клавишей BREAK.

MOON LANDING

Существует немалое количество всевозможнейших версий программ управления посадкой корабля на Луне. У этой несложной, но увлекательной программы есть то неоценимое преимущество, что Вы можете ее набрать сами.

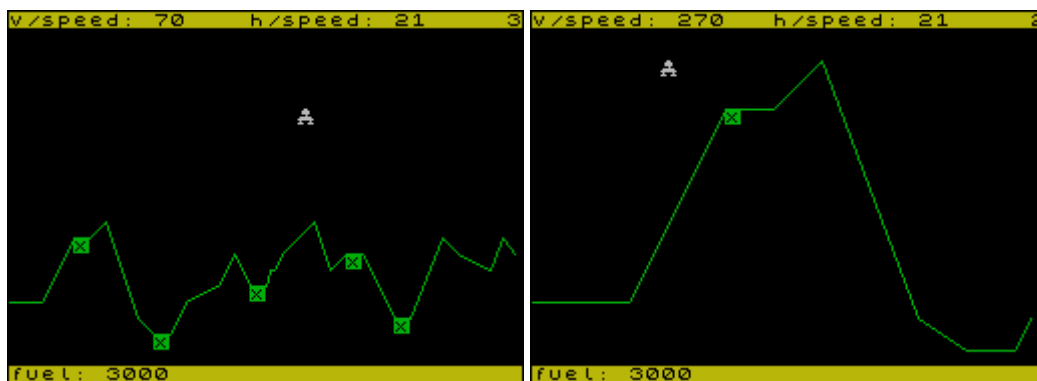
Нам бы очень хотелось сделать все сообщения в программе на русском языке, но поскольку мы не можем знать как активируется русский шрифт в компьютере у каждого конкретного пользователя, мы даем эту распечатку под базовую фирменную машину. Вы сами сможете внести необходимые изменения в соответствии с особенностями своей машины, делая перед PRINT переход на русский шрифт и поменяв текст, стоящий в кавычках на русскоязычный.

Цель программы - совершить посадки на 5-ти лунных базах. Управляющие клавиши - "9" - тяга, "7", "8" - поворот аппарата. Особый интерес представляет то, что при приближении к поверхности Луны изменяется масштаб изображения и Вы можете более тонко управлять посадкой.

Существенным является ограниченность запаса топлива. Чем дольше Вы маневрируете перед посадкой, тем труднее Вам будет выполнить всю задачу.

Не отчаивайтесь, если после того, как Вы наберете программу, она не заработает. Программа достаточно трудоемкая для набора, и как бы Вы аккуратны ни были, 5-7 ошибок Вы все равно сделаете. Зато Вас впереди ждет удовольствие отладить программу. Вы убедитесь, что это интереснее и полезнее, чем просто в нее поиграть.

Во время набора не забывайте периодически делать SAVE. Это избавит Вас от многих хлопот.



```

10 REM Настройка переменных
20 PAPER 0: BORDER 0: CLS : INK 4
30 LET lv=.5: LET sc=0: LET z =3: LET l=0: LET vs=30: LET hs=21
35 GO SUB 9000
36 LET Pad=0: LET f=3000: DIM C(30)
40 LET ls=9530: LET h = 0: LET d = 0: LET line=2: LET column=10: LET drift=3: LET gs=147:
    LET down=4: LET s=0
60 GO SUB 9500
70 PRINT AT 14,4; FLASH 1;"x"; AT 20,9;"x"; AT 17, 15; "x"; AT 15, 21; "x"; AT 19, 24; "x"
100 REM Начало работы программы
110 INK 7
115 PRINT AT 0,0; INVERSE 1; INK 6;"v/speed: ";VS;" "; AT 0, 15; "h/speed: ";hs;" ";
    AT 0,31 ;z;#0; INVERSE 1; INK 6; AT 0,0; "fuel: "; f; " "
116 IF INT line=h-2 AND (INT column=d OR INT (column+.5)=d) THEN GO TO 600
120 PRINT AT line,column; CHR$ gs
125 IF ATTR (line+SGN vs,column + SGN drift)=4 OR ATTR (line, column + SGN drift)=4 THEN GO
    TO 800
130 LET pc=column: LET pl=line
140 LET column=column+SGN drift
145 LET line=line+lv*SGN vs*(ATTR (line+ SGN vs,column) <> 132)
150 LET drift = drift+(INKEY$="9" AND gs=146)-(INKEY$="9" AND gs=144)
155 LET down=SGN (down+(INKEY$="9" AND gs=145)-(INKEY$="9" AND gs=147)-down*(INKEY$ = ""))
160 LET gs=gs+(INKEY$="7")-(INKEY$ = "8")
170 IF gs=148 THEN LET gs=144
180 IF gs=143 THEN LET gs=147
190 IF INKEY$ ="9" THEN LET f=f-10
195 IF INKEY$ ="9" THEN BEEP 08,20
200 IF column >= 30 THEN GO TO 700
210 IF column >= 0 AND column<1 THEN GO TO 760
220 IF line=1 THEN LET line=2
230 LET vs=vs+5+15*down: LET hs=drift*7
235 IF S <> 0 THEN GO TO 250
240 IF line >= 12 AND s=0 THEN GO TO 505
250 IF s=1 AND line <= 2 THEN RESTORE 9530: LET column=(column /3.1) +22*
    (ls=9580)+19*(ls=9570)+ 11*(ls=9560)+6*(ls=9550) : LET ls=9530: CLS : INK 4: LET
    line=11: LET s=0: GO TO 60
270 IF l = 6 THEN BEEP .08,40: LET l=0
275 LET l=l+1
280 IF line <= 2 THEN LET line = 2
290 IF f = 0 THEN GO TO 950
400 PRINT AT pl,pc;" "
410 GO TO 100
500 REM Изменение сценария
505 LET column=INT column
510 LET ls=9530+10*(column>0 AND column<7)+20*(column>6 AND column<13)+30*(column>12 AND
    column< 18)+40*(column>17 AND column<22)+50*(column>21 AND column<31)
520 CLS : INK 4: LET line=2: RESTORE ls
530 GO SUB 9500
540 LET h=6*(column>0 AND column<7)+21* (column>6 AND column<13) +16*(column>12 AND
    column<18)+9*(column>17 AND column<22)+21* (column>21 AND column<31)
550 LET d=12*(column>0 AND column<7)+9*(column>6 AND column<13)+ 12*(column>12 AND
    column<18)+16*(column>17 AND column<22)+ 6*(column>21 AND column<31)
560 PRINT AT h,d; FLASH 1;"x"
565 LET column=column-6*(column>21)-5*( column>17) -5* (column>12) -6*(column>6)
570 LET column=column*3.1: LET s=1
580 GO TO 100
600 REM Посадка
620 IF gs <> 147 OR vs>10 THEN GO TO 800
630 IF c(line+column) <> 1 THEN GO TO 680
640 PRINT AT 21,0; "RESTRICTED LANDING PAD. GO AWAY."
650 FOR i=1 TO 500
660 IF INKEY$="9" THEN LET h=0: GO TO 100
670 NEXT i: GO TO 800
680 PRINT AT 10,7; "GREAT LANDING"; AT 11, 6;"CONGRATULATIONS!"

```


[illegible]

Комментарий к программе.

10 - 70 Настройка программных переменных.

100 - 410 Главная программа.

115 - Печать сообщений на экране

116 - Проверка посадки.

120 - Изображение корабля.

125 - Проверка аварии.

130 - 145 Расчет положения корабля.

150 - 155 Расчет горизонтального и вертикального перемещений.

160 - 180 Определение, в какую сторону направлен корабль.

190 - Если двигатель включен, то запас топлива уменьшается.

200 - 210 Проверка не находится ли корабль на краю экрана.

220 - Ограничение по вертикали.

230 - Пересчет вертикальной и горизонтальной скоростей.

235 - 250 Проверка на необходимость перестроения ландшафта.

290 - Проверка на конец топлива.

400 - Стирание корабля.

500 - 580 Перестроение ландшафта в новом масштабе,

600 - 696 Проверка условий посадки.

700 - 795 В увеличенном масштабе переход из одной зоны в другую.

800 - 895 Изображение аварийной посадки и возврат к началу.

900 - 940 Пересчет очков после удачной посадки, переход к началу на более высоком уровне сложности.

950 - 960 Действия, если кончилось топливо.

9000 - 9060 Графика UDG для изображения корабля.

9500 - 9999 Данные для построения лунного ландшафта.

* * *

Программные переменные.

lv - уровень сложности.

sc - счет.

z - количество оставшихся попыток.

vs - вертикальная скорость.

hs - горизонтальная скорость.

pad - количество площадок, на которые еще надо сесть.

f - остаток топлива.

ls - номер строки, в которой хранятся данные для изображения ландшафта.

h - вертикальная координата площадки в экране с большим масштабом.

d - то же, но по горизонтали.

line - номер строки для изображения корабля.

column - номер столбца для изображения корабля.

drift - смещение корабля. Положительное - вправо.

gs - код символа UDG, которым изображается корабль.

down - смещение корабля по вертикали. При положительном - вниз.

s - равно 1, если изображение выполнено в крупном масштабе, иначе 0.

pl - номер строки, в которой был корабль перед данной.

pc - номер столбца экрана, в котором был корабль в предыдущий момент времени.

Массив [c] - Если $c(\text{line} + \text{column})$ равно единице, то на данной площадке Вы уже сядились.

НАШ КОНКУРС

"ИНФОРКОМ" с удовлетворением отмечает, что конкурс на лучший тематический сборник игровых программ, объявленный нами в индивидуальных письмах к своим читателям, набирает силу.

В течение ноября - декабря мы получили около трех десятков заявок на участие в конкурсе. Благодарим всех, кто на него откликнулся.

Вместе с тем, должны сказать честно, что не все письма соответствуют требованиям конкурса, в частности, есть непроработанные проекты сборников без полного указания названия фирмы, года выпуска программы и размера программы.

В некоторых сборниках составители указывают вместо фирмы, выпустившей программу, фамилию хаккера, который ее взломал, т.е. снял с нее фирменную защиту и оставил свою "визитную карточку". Например, Билл Гильберт и т.п.

Есть, увы, несколько писем, в которых названия программ указаны неверно.

Мы понимаем, что есть очень много прекрасных программ, в которые можно с интересом играть, но в которых нельзя найти ни названия фирмы, ни года выпуска. Но ведь Вас никто и не заставляет включать в список такую программу. Наличие в вашем списке таких пунктов говорит о том, что это механическое перечисление программ, но не творчески проработанный сборник.

Напоминаем, что призом в конкурсе является не денежное вознаграждение, а заказ на подробную разработку сборника с буклетом описаний с соответствующим объявленным вознаграждением. И конечно же мы не сможем предоставить такой контракт тем, кто не сумел выполнить несложного списка в строгом соответствии в указанными требованиями.

Мы будем до марта месяца с.г. принимать ваши заявки, в апрельском выпуске "Игрового Приложения к ZX-РЕВЮ" будут опубликованы те, которые не противоречат условиям конкурса, а далее читатели сами выберут три лучших сборника, авторы которых подучат контракты. Конечно проекты будут напечатаны нами без указаний авторов, а под девизами, которые они сами себе избрали, дав названия присланный сборникам. И последнее. Мы дадим эти списки в "игровом приложении" не вместо основного материала, это было бы нечестно по отношению к нашим читателям, а дополнительно к обязательному материалу.

TOP TEN

Поскольку это всего лишь первый выпуск, то у нас пока нет писем от Вас, по которым мы могли бы составить "ЛУЧШУЮ ДЕСЯТКУ".

Тем не менее, в какой-то степени мы можем судить о популярности и известности тех или иных игровых программ хотя бы по тем вариантам, которые нам были присланы на конкурс.

Проанализировав эти варианты, мы сумели выделить наиболее популярную десятку за январь месяц.

Вот она:

1 Chimera	FIREBIRD	1985
2 Exolon	HEWSON CONS.	1987
3 Saboteur 1	DURELL	1986
4 Elite	FIREBIRD	1984
5 Fred	INDESCOMP	1986
6 Saboteur 2	DURELL	1987
7 Barbarian	PALACE	1987
8 Xecutor	SOFT. INTERN.	1987
9 Manic Miner	BUG-BYTE	1983
10 Pijamarama	MICRO-GEN	1985

Комментировать полученный список не беремся. Видимо, пока было слишком мало материала для анализа.

ЭТОТ РАЗДЕЛ ВЕДЕТ НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС",
СПЕЦИАЛИЗИРУЮЩИЙСЯ В ВОПРОСАХ АППАРАТНОГО ОБЕСПЕЧЕНИЯ СИНКЛЕР-
СОВМЕСТИМЫХ КОМПЬЮТЕРОВ И СХЕМОТЕХНИЧЕСКИХ РЕШЕНИЯ ПЕРИФЕРИЙНЫХ
УСТРОЙСТВ

127566, МОСКВА И-566, НТК "ПЛЮС"

РАСШИРЕНИЕ ПАМЯТИ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА "СПЕКТРУМ"

Фирма SINCLAIR, разрабатывая персональный компьютер "СПЕКТРУМ", видимо решила сделать все возможное, чтобы он был максимально дешевым. Это дало возможность в короткие сроки захватить рынок, но при этом значительная доля затрат пользователя была перенесена в сферу программного обеспечения и в широкий спектр периферийных устройств, которые очень часто бывают сравнимы по цене со стоимостью самого компьютера, хотя и значительно повышает его потребительские свойства.

В техническом приложении к ZX-РЕВЮ мы будем давать всевозможные схемотехнические решения, направленные на повышение возможностей Вашего компьютера. Мы часто будем опираться на западные публикации, но по сравнению с ними наша задача сильно осложнена тем, что мы не можем опираться на некоторую стандартную фирменную модель, снабженную стандартным разъемом. Это связано с тем, что в нашей стране парк самодельных Синклер-совместимых компьютеров за последние два года чрезвычайно вырос и давно обошел количество фирменных машин, имеющих на руках у населения.

В центральных регионах страны наибольшее распространение получили варианты "Московский", "Ленинградский", "Балтика", "128К", "Пентагон", "Дубна" и др. варианты. В восточных регионах распространен "Новосибирский", а в западных - "Львовский" и "Минский" варианты, все варианты перечислить невозможно.

В одних случаях вариации достаточно сильно отличаются друг от друга, но есть немало и таких, которые получили новые названия от своих "авторов" только потому, что он перерисовал принципиальную схему в другом формате или, скажем, заменил какие-то микросхемы на те, которые выпускаются на родном предприятии.

При таком огромном количестве разновидностей моделей мы не в состоянии охватить их все, но надеемся на вашу помощь. Обобщение этих материалов позволит нам в наших разработках наилучшим образом давать рекомендации, универсальные для большинства пользователей.

В первом выпуске "технического приложения" мы предлагаем вашему вниманию обзор архитектуры памяти компьютера и описание некоторых возможностей ее расширения.

Микропроцессор Z-80, как известно, может обслуживать адресное пространство размером в 64К. В стандартном компьютере оно распределено следующим образом:

FFFF	ОЗУ -- 32К
8000 7FFF	ОЗУ 16К
4000 3FFF	ПЗУ 16К
0000	

Здесь и далее мы будем давать адреса в шестнадцатеричном коде. В низших адресах от 0000 до 3FFF располагается ПЗУ компьютера объемом 16К, содержащее операционную систему "СПЕКТРУМа". Память в адресах с 4000 до 7FFF выполнена на 8 микросхемах динамической памяти DRAM типа 4116 с организацией 16К*1. Советским аналогом является микросхема 565РУ3 или 565РУ6 (то же, но с одним источником питания). Дополнительная память 32К в адресах с 8000 по FFFF представлена восемью микросхемами DRAM 64К*1, у которых (у всех одинаково) отбракована одна из половин - левая, правая, верхняя или нижняя. Советским аналогом являются "половинки" микросхем 565РУ5 -

565PY5D1 и 565PY5D2.

Что дает использование отдельных полей памяти? Этот вопрос повторяется очень часто, поэтому мы вкратце его осветим.

Контроллер дисплея в фирменном "СПЕКТРУМе" выполнен на одной БИС программируемой логической матрице ULA (Uncommitted Logic Array).

СПРАВКА "ИНФОРКОМа"

Хитрость микросхем типа ULA состоит в том, что фирма-изготовитель выпускает их в незавершенном виде, т.е. выпускает полуфабрикат, который далее поступает на доработку в "секретный" цех, где их доводят персонально под требования заказчика, в данном случае под фирму Sinclair Research Ltd. Сэр Клайв Синклер был вынужден прибегнуть к такому методу защиты своего компьютера наученный горьким опытом со своей предыдущей машиной ZX-81, которая была мгновенно пиратски растиражирована по всей Европе.

Надо сказать, что эта защита свою роль выполнила блестяще. Прошел не один год, пока научились эмулировать эту микросхему с помощью аналогичных конструкций, выполненных на нескольких десятках микросхем.

Подробнее об этом Вы можете прочитать в нашем переводе Яна Скэйлса "Путеводитель по периферии ZX-"Спектрум". (См. информ. листок "ИНФОРКОМ ПРЕДЛАГАЕТ", который получают все зарегистрированные получатели "ZX-РЕВЮ").

Такой же метод защиты Синклер применил и в своем Интерфейсе-1, который выполняет функции контроллера локальной сети, контроллера микродрайва и последовательного порта RS232. ИНФОРКОМ располагает информацией о том, что в г. Свердловске успешно "раскололи" и эмулировали микросхему ULA, на базе которой создан этот интерфейс. Называют и конкретный адрес. Мы пока не можем ни подтвердить ни опровергнуть эту информацию. Готовы опубликовать рекламу авторов в "Рекламном приложении" бесплатно.

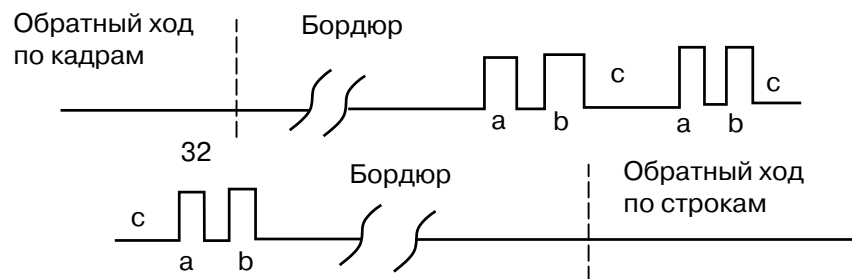
В самодельных компьютерах функции ULA выполняет "рассыпуха" из 20-40 микросхем, причем многие варианты самодельного "Спектрума" различаются именно построением контроллера дисплея и количеством микросхем, затраченных на него.

Оперативная память дисплея Спектрума расположена в адресах:

4000 - 57FF память точек экрана,

5800- 5AFF - память атрибутов.

Через каждые 20 мс контроллер дисплея Спектрума аппаратным образом обязан считать информацию из этой области для регенерации изображения на экране дисплея, что он и делает 64 раза во время длительности каждой строки - 32 раза считывает по 8 точек строки, состоящей из 256 точек и чередует это с 32-кратным считыванием атрибутов знакомест, в состав которых входит данная строка. Понятно, что во время этих считываний микропроцессор не может обращаться к микросхемам памяти, в которых расположена память дисплея. В фирменной машине этот конфликт решен так, что если на момент аппаратной регенерации изображения на экране дисплея процессор, выполняя программу, обратится к этим микросхемам, то на него аппаратно перестают подаваться тактовые импульсы. Это, конечно, замедляет выполнение программы, но не очень сильно, т.к. во-первых, не замедляется работа с "верхними" 32К памяти, которые физически выполнены на других микросхемах, во-вторых, процессор "свободно" работает с "нижними" 16К - обращаясь к ПЗУ "Спектрума" и, в-третьих, он имеет возможность работать даже и с экранной областью памяти, так как регенерация изображения происходит неравномерно, что показано на диаграмме.



Работа процессора приостанавливается только при одновременном обращении в моменты а) и б), когда а) - происходит считывание в буфер байта точек экрана, б) - считывается байт атрибутов знакомест. Все остальное время - во время обратного хода по кадрам, по строкам, во время вывода бордюра в верхней, нижней, левой и правой частях экрана, а также внутри основной рамки изображения в моменты с), процессор работает без ограничений.

Эти вышеперечисленные во-первых. Во-вторых и в-третьих и объясняют довольно высокую скорость работы графики, хорошее звуковое сопровождение и другие прелести "Спектрума".

Из самодельных аналогов наиболее полно копируют идеологию и архитектуру оригинального "Спектрума" те, что используют отдельную память, например, "Московский вариант". В вариантах с общим полем памяти используются микросхемы 64К*1, нижние 16К которых как бы "затеняются" ПЗУ, использование общего поля памяти сокращает общее количество микросхем на 13-15 корпусов, уменьшает потребление компьютера, его размеры, т.е. обладает несомненными достоинствами. Сложность пожалуй лишь в том, что для сохранения прежней скорости работы программ приходится повышать частоту обращения к памяти, что повышает требования к быстродействию микросхем памяти их согласованию с другими микросхемами и т.п. Процессор в таких вариантах схем приостанавливается или остановкой тактов или с помощью сигнала WAIT.

Следует также отметить, что оригинальный "Sinclair 128+" тоже выполнен по схеме с общим полем памяти, и в первые годы после его появления на рынке пользователи часто отмечали некоторую несовместимость с ним программ, написанных ранее для "Spectrum 16K" и "Spectrum 128K", но в программах, написанных позже, программисты видимо учли архитектуру 128-го и поэтому с несовместимостью программ практически сталкиваться не приходится.

Теперь рассмотрим некоторые возможности изменений и дополнений в устройстве памяти Спектрума.

Нужно отметить, что всякие изменения и дополнения в архитектуре не могут быть учтены фирменным программным обеспечением (исключение составляет расширение памяти до 128К, выполненное по тому же принципу адресации, что и в оригинале) и могут представлять для Вас интерес в том случае, если Вы готовы отражать эту новую архитектуру и в программах.

Один из способов расширения памяти - это введение страничной адресации. Таким способом можно увеличить память до очень значительной величины. Например, если использовать хотя бы один порт ввода/вывода, то записывая в него байт данных, уже можно адресовать 256 страниц, дешифрируя этот байт данных.

Для управления включением страниц можно использовать сигнал CAS - сигнал выборки, проходящий на 15-ый вывод 6-ми микросхем памяти. Все остальные микросхемы памяти соединяются параллельно основной памяти, за исключением только сигнала CAS. Работать в данный момент будет та страница, до которой доходит сигнал CAS, прохождением которого управляет дешифратор адреса страницы.

На схеме 1 показано, как к основной памяти 32К (для компьютеров с отдельными полями памяти (КРППП)) можно присоединить 15 страниц по 32К каждая.

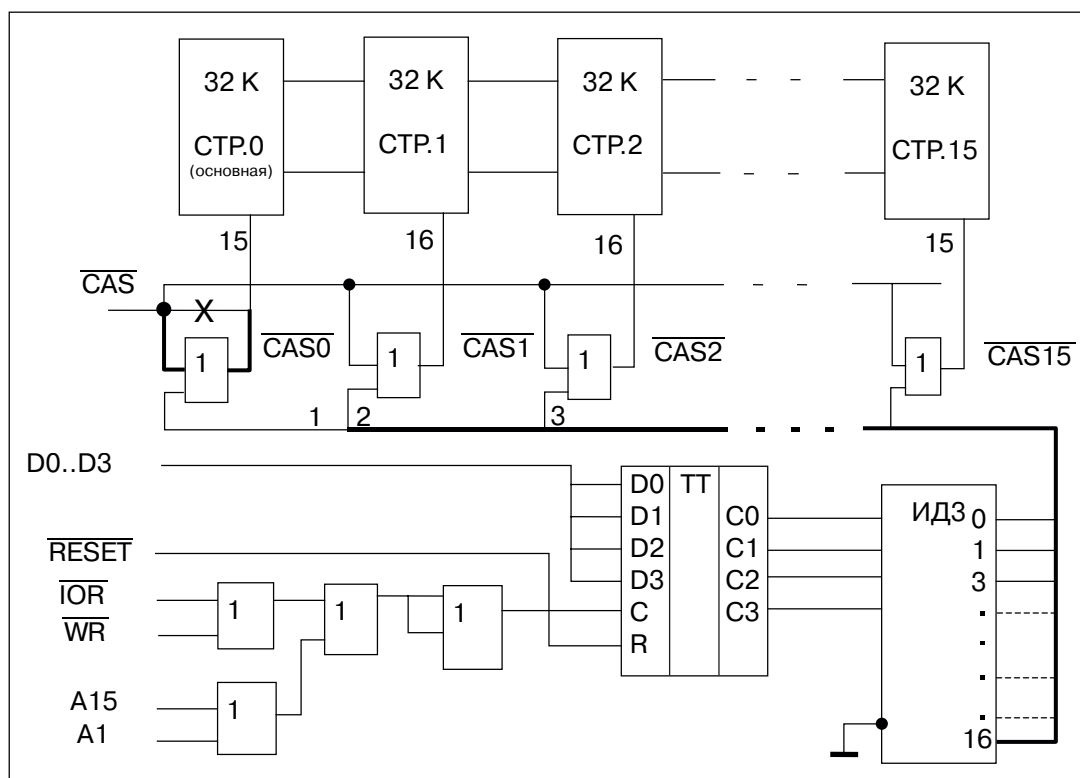


СХЕМА 1.

Разумеется, если присоединять так много микросхем, то нужно буферизировать все шины, т.е. эта схема дана как пример одного из подходов к расширению памяти. На схеме 2 показано конкретно, как увеличить память на 32К для фирменного "SP" или для КРПП. Для этого надо "половинки" микросхем заменить на целые 64К*1.

Для получения старшего адреса динамической памяти DRAM7 на один вход мультиплексора подается A14, а другой, в зависимости от того, какая из половинок используется, соединен с землей или плюсом источника питания. Вот на него и нужно подать выход триггера, который является портом ввода/вывода с адресом 7FFD. Адрес Вы можете выбрать по своему усмотрению, изменив соответственно дешифрацию. Пунктиром обведена часть схемы компьютера.

Конденсатор обеспечивает начальную установку банка 0.

Переключение банка данных происходит по команде из Бейсика OUT 32765,X, где X - любое число от 0 до 255. При этом включение дополнительных 32К можно контролировать по светодиоду. Обратное переключение по команде IN 32765,X.

В машинных кодах нужно выполнить цепочку команд

```
LD BC, 7FFD
OUT(C), A
```

Перед заменой банка памяти необходимо перенести стек вниз командой CLEAR 32767, иначе после замены нарушится нормальная работа процессора.

Для компьютеров с общим полем памяти, у которых нижние 16К "затеняются" ПЗУ было бы интересно иметь возможность пользоваться этими 16К с 0000 по 3FFF. Например, можно было бы скопировать в эту область памяти ПЗУ и пользоваться ею при отключенном ПЗУ. Это позволило бы вносить в ПЗУ всевозможные изменения. Например, уменьшить некоторые константы в подпрограммах ввода/вывода данных на магнитофон, что позволило бы, при хорошем качестве магнитофона и ленты, хранить данные с большой плотностью и быстрее выгружать и загружать их, или можно было бы оперативно изменять шрифты, хранящиеся в ПЗУ, загружать русский шрифт, использовать свободные 1170 байт ПЗУ для хранения нужных подпрограмм, а также проводить работу по усовершенствованию самой операционной системы Спектрума или интерпретатора Бейсика.

Для решения этой задачи можно порекомендовать следующий подход, необходимо отыскать на схеме сигнал ROMCS (RDRAM) т.е. сигнал, осуществляющий выбор микросхемы или микросхем (если ПЗУ выполнено не на одной микросхеме).

Этот сигнал обычно формируется из сигналов A14, A15, RD, MREQ. Когда эти сигналы в логическом нуле - выбрано ПЗУ. ROMCS (RDRAM) обычно приводит на 20-ый или 22-ой вывод микросхемы ПЗУ.

Найдя схему формирования сигнала ROMCS (RDRAM), нужно осуществить в ней доработки для перехвата сигнала A15. Например, в схеме "Ленинградский вариант" сигнал RDRAM формируется так, как показано на схеме 3.

Видно, что когда не выбрано ПЗУ, то выбирается ОЗУ, поэтому, если перехватить сигнал A15, то при любом адресе памяти выбираться будет ОЗУ, в том числе и та область, которая расположена "под ПЗУ", перехват можно осуществить следующим образом.

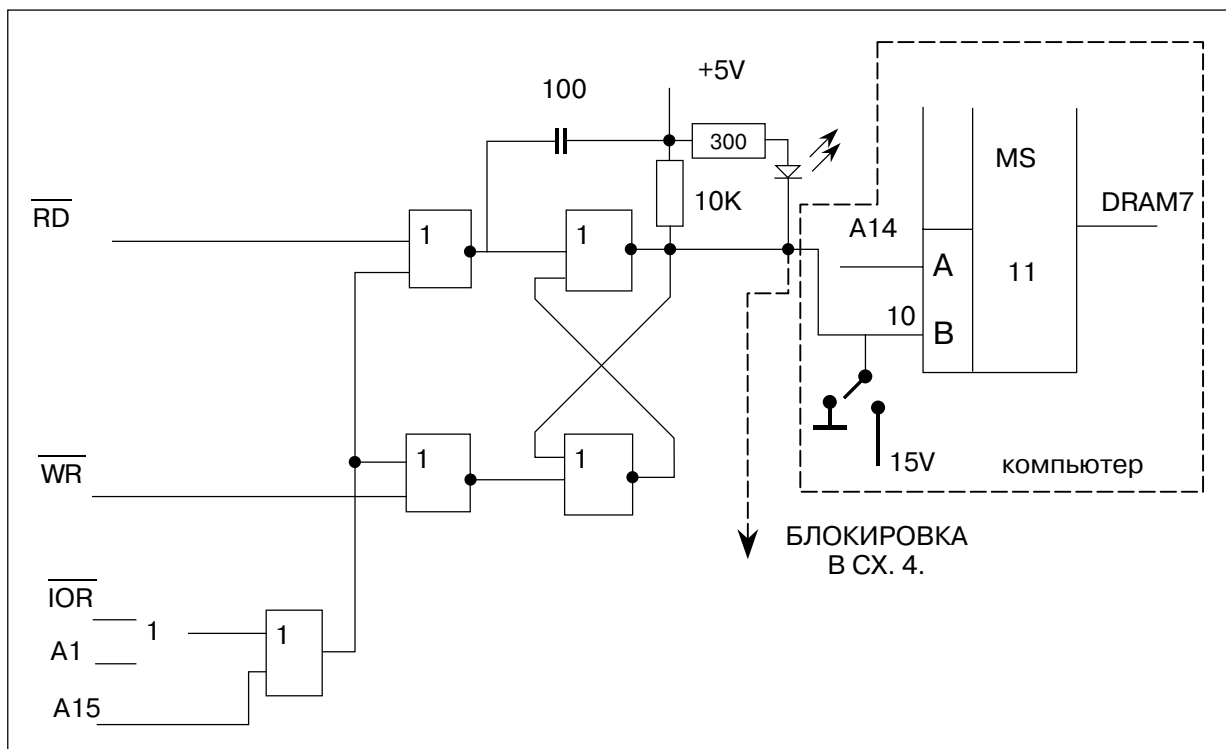


СХЕМА 2.

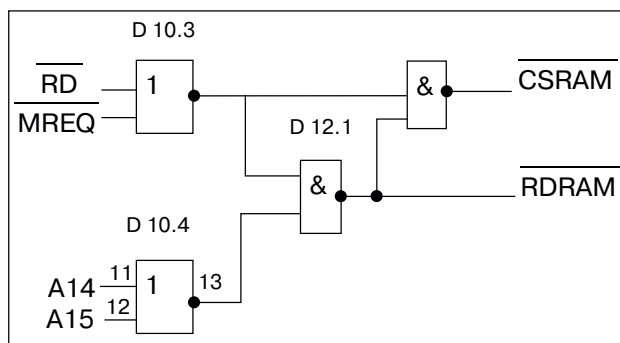


СХЕМА 3.

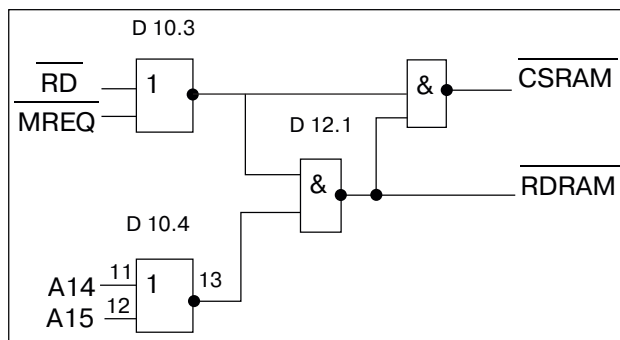


СХЕМА 4.

Сигнал на блокировку приходит со схемы 2. Это тот же выход триггера, что идет на мультиплексор для выбора дополнительного блока памяти.

В "Новосибирском варианте" перехват можно осуществить на входе в PLM 556PT1 - DD10, вывод 9.

В варианте "Балтика" это вывод 3 микросхемы D34, и т. д.

В заключение приведем конкретную схему расширения памяти "Ленинградского варианта" Спектрума до 128 К, программно совместимую с фирменным "128+".

1. Сигнал CHSCR подключить к 13 D17, предварительно отключив его от "земли".

2. Сигнал A14(ROM) подключить к 27 ножке ПЗУ 27256, установленной на место микросхемы D29. Это ПЗУ должно содержать в себе и "старое" и "новое" ПЗУ, в отличие от ПЗУ 27128 "Спектрума 48".

3. Сигналы A14' и A15' подключаются к 10 и 13 выводам D16, которые предварительно надо отключить от адресных линий A14 и A15.

4. Сигнал CAS0 подключается к 15-ым выводам микросхем ОЗУ, установленным на плате, от которых предварительно отключается сигнал CAS.

5. Сигнал CAS1 подключается к 15-ым выводам второй линейки микросхем ОЗУ, напаянных на микросхемы ОЗУ, установленные в плате.

Карта прошивки микросхемы DD2 556PT4 (с) 1990 "Сделай сам" & МОА

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	C	D	E	F	C	D	E	F	5	5	5	5	5	5	5
10	A	A	A	A	A	A	A	A	8	9	A	B	4	5	6
20	4	5	6	7	4	5	5	7	5	5	5	5	5	5	5
30	6	6	6	6	6	6	6	6	4	5	6	7	4	5	6
40	C	D	E	F	C	D	E	F	D	D	D	D	D	D	D
50	E	E	E	E	E	E	E	E	C	D	E	F	C	D	E
60	C	D	F	F	C	D	E	F	D	D	D	D	D	D	D
70	E	E	E	E	E	E	E	E	C	D	E	F	C	D	E
80	8	8	8	8	8	8	8	8	9	9	9	9	9	9	9
90	A	A	A	A	A	A	A	A	B	B	B	B	B	B	B
A0	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5
B0	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7
C0	C	D	E	F	C	D	E	F	D	D	D	D	D	D	D
D0	E	E	E	E	E	E	E	E	C	D	E	F	C	B	E
E0	C	D	E	F	C	D	E	F	D	D	D	D	D	D	D
F0	E	E	E	E	E	E	E	E	C	D	E	F	C	D	E

Предлагаю игровые и прикладные программы к компьютеру ZX Спектрум.

Обращаться по адресу: 258600 Черкасская обл. г.Ватутино, ул.Ломоносова, 4
Свичару Эдуарду Степановичу.

АЛАТЫРЬ

МЦ "Алатырь" приступает к выпуску набора радиоконструктора. В набор входят:

- а) печатная плата для компьютера Спектрум;
- б) клавиатура для того же компьютера;
- в) корпус.

Обращаться по адресу: 429600 Чув.АССР
Алатырь, ул.Октябрьская, 7

РАДАР

Предлагаем системную плату для ПЭВМ "Синклер" с набором кнопок (40 шт.) или с пленочной клавиатурой
(по договоренности и по выбору).

Стоимость:

- 620 руб.- за наличный расчет
(наложенным платежом);
- 680 руб.- за безналичный расчет.

Наш адрес: 423034 г.Чебоксары,
Московский пр. дом 40 Производственный
кооператив "РАДАР".

Продается ПК "Синклер" в комплектации с описанием и 2-мя кассетами с программами (можно на тему заказчика).
Стоимость 1100 руб.

Обращаться по адресу: 1
60028 Вологда-28, 1 мкр-н, д.23, кв.11
Сабинин Евгений Михайлович.

Программирую все виды РПЗУ и ППЗУ включая зарубежные серии 27XXX любого объема, на следующих условиях:

1. Срок исполнения партии до 100 шт. - 24 часа с момента получения заказа.
2. Материал заказчика.
3. Вместе с партией высылается образец записанной микросхемы (эталон).
4. Без эталона заявки не принимаются.
5. Стоимость услуги за 1 штуку:

155РЕЗ - 50 коп.

556РТ4 - 1 руб.

573РФ2 - 3 руб.

573РФ4,6; 2764 - 5 руб.

27256 - 10 руб.

Остальные в зависимости от емкости.

Обращаться по адресу: 620102
г.Свердловск, ул.Ясная 32. корп.2 кв.142
Борнуков М.М.

ИКАР

Малое предприятие научно-производственная фирма "Икар" производит:

- продажу (с возможной установкой и гарантийным обслуживанием) бытового компьютера (БК) "Икар", программно совместимого с компьютером ZX-Spectrum.

Компьютер "Икар" разработан и изготовлен в НПО "Электроприбор".

- продажу, разработку и изготовление аппаратного и математического обеспечения АСУ ТП, АВС, АРМ, компьютерных классов, периферийного оборудования на базе БК "Икар" и (или) ЭВМ IBM PC/AT(XT).

Обращаться по адресу:
310002 Харьков, ул.Иванова, 24 НПФ
"Икар" тел.47-41 37

Вышлю наложенным платежом
запрограммированные ППЗУ для ПК
Радио-86-РК, Специалист, Спектрум,
Орион-128, а также комплектующие
изделия для их сборки и ремонта.

Обращаться: 454091 Челябинск а/я 16287

АРСЕНАЛ

Экспериментальное объединение
"Арсенал", включающее в себя оборонные
предприятия, приступившие к конверсии,
лучших разработчиков страны в области
ЭВМ, предлагает организациям и частным
лицам компьютеры нового поколения,
совместимые с серией "СПЕКТРУМ".

Компьютеры "Компаньон" и "Магик" -
самые дешевые в стране, а по надежности
значительно превосходят зарубежные
аналоги.

Компьютеры "Арсенала" - это компьютеры
для всех. Диапазон их возможностей
универсален и устроит любого - от
первоклассника до академика. Столь же
универсальна и сфера использования этих
компьютеров - от учебных компьютерных
классов и игровых залов до процессов
управления производством и подготовки
научно-технической документации.

Компьютеры и информационные сети на их
основе продаются по наличному и
безналичному расчету и устанавливаются
"под ключ". По истечении гарантийного
срока предоставляются льготы для
приобретения нового компьютера.

Наш адрес: 125190 Москва, а/я 179 Факс:
(095) 2002216, 2002217 Телекс: 411700
ТОМ Телефон: 278-50-38
(звонить с 11.00 до 15.00, кроме субботы и
воскресенья, время московское).

МАСТЕР

АСП "Мастер" предлагает программы на
импортных аудиокассетах С-90 для
компьютеров совместимых с СИНКЛЕР-
системой.

Каталог высылается бесплатно. Цена
кассеты 49 руб.

Обращаться по адресу: 353410-
Краснодарский край, г.Анапа, 11 отд.связи
а/я 6, АСП "Мастер".

Имею программу англо-русский словарь
"SLOVAR SM+"

1660 англ. слов, количество сравниваемых
букв задается пользователем или машиной
(по умолчанию), переводить можно как по
одному слову, так и группами (до 250 слов
и более).

Собственная разработка.

692438 Приморский край, Дальнегорский
район, п. Краснореченск, ул.Гастелло, 13,
кв.8 Солдатову В.Ю.

"ЭЛИКОН-50"

Опытный завод Грозненского НПО
"Промавтоматика" освоил выпуск двух
модификаций бытовых компьютеров,
совместимых с системой SINCLAIR,
"Эликон-050"

с выносным БП по цене 1400 руб.
и со встроенным БП по цене 1500 руб.,
а также оригинального джойстика
"ELICON WORLD MASTER"
по цене 50 руб.

Заказы отправляйте по адресу: 364002
г.Грозный, пр. Кирова 2

РАДУГА

У нас самые низкие цены! Мы тиражируем
каталоги, программы.

Напишите по адресу:

Московская обл., г.Луховицы, ул.
Тимирязева д.7 кв.96 Силкину О.В., и в
день получения Вашего письма Вам будет
выслан перечень услуг кооператива
"Радуга".

ИНФОРКОМ

ищет зарубежные журналы,
посвященные персональному
компьютеру ZX-SPECTRUM:

SINCLAIR USER
ZX-COMPUTING
YOUR SPECTRUM
SINCLAIR PROGRAMS и др.

Возможные варианты:

- покупка по цене, назначенной Вами;
- обмен оригиналами;
- обмен копиями;
- обмен на время, с целью копирования.
- и др. условия по Вашему желанию.

При обращении просим предоставить
список имеющихся у Вас журналов с
указанием названия, года, номера, а также
указать оригинал/копия.

ВАЖТЕК просьба не предлагать. 107241,
Москва, Б-241, а/я 37, НТК "ИНФОРКОМ"

БИС

В канун Нового Года "ИНФОРКОМ"
получил из г. Сыктывкара
презентационный выпуск бюллетеня БИС,
выпускаемого научно-производственным
кооперативом "СБИС". Бюллетень БИС -
это информационное приложение к
технической документации для
персональных компьютеров БК-0010.
Конечно такая его специализация не
соответствует интересам большинства
наших читателей, но тем не менее,
учитывая невысокую стоимость (один
рубль за выпуск объемом 12 стр.), и
хороший стиль изложения, он вполне
может служить повышению общего уровня
информированности любителей
персональных компьютеров страны. Кроме
того, он содержит конкретные адреса
клубов любителей ПК, в том числе и
системы "СИНКЛЕР", а также адреса
комиссионных магазинов, где если
повезет можно приобрести что-либо
полезное.

Нам выпуск понравился, благодарим
сотрудников "СБИС" и рекомендуем его
вниманию наших читателей.

Адрес НПК "СБИС"

167024, г.Сыктывкар, а/я 430, НПК "СБИС"

Хотел бы обмениваться программами и
документацией к ПК "Спектрум". Возможна
купля-продажа наложенным платежом.

Обращаться по адресу: 610004 Киров а/я
1199 Таранцев И.Е.

Имею около 300 программ для "ZX-
Spectrum". Обменяюсь с другими
пользователями.

Обращаться по адресу: 620120.
г.Свердловск-45, ул.Ленина 24-40
Буденко А.В.

Предлагаю усилитель записи (схему или
готовый блок), позволяющий производить
высококачественные записи на кассетном
магнитофоне при использовании любой
исправной кассеты. Схема незаменима
при тиражировании программ.

327018 Г.Николаев, пер. Кобера 13, кв.26.
Шилкин А.А.

Прошу помочь в приобретении книги Яна
Логана и Фрэнка О'Хара "Диссасемблер
ROM Спектрум" или ей подобной в полном
объеме.

184364 Мурманская обл. пос.Мурташи,
ул.Тягунова д.2 кв.7 Кириллову С.В.

Остро нуждаюсь в описании компилятора
языка "СИ" фирмы HISOFT V11 1984г.

156901 Костромская обл., г.Волгореченск,
ул. им. Ленинского комсомола, 47-21
Закорюкину Ю.Е.

Ищу ZX-Микродрайв, ZX-Интерфейс-1 к
нему и картриджи за разумную цену.

656004, г.Барнаул, 4 а/я 502

ПЛЮС
НАУЧНО ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС"
ПРЕДЛАГАЕТ ВАШЕМУ ВНИМАНИЮ СВОИ УЧЕБНО-МЕТОДИЧЕСКИЕ РАЗРАБОТКИ

Методическая разработка (МР) N1.

Любители компьютерных игр найдут в ней несколько вариантов схем подключения джойстика. Наличие нескольких вариантов позволяет Вам выбрать наиболее приемлемый по наличию элементной базы.

Те, у кого есть принтер, смогут подключить его с помощью одной из схем последовательного или параллельного интерфейса принтера. Приведенные программы поддержки принтера (драйверы) позволят Вам распечатывать как текстовую, так и графическую информацию. Схемы интерфейсов и программы поддержки предназначены для работы с любыми принтерами, оборудованными одним из стандартных интерфейсов: CENTRONICS (ИРПР-М), ИРПР (IFSP), RS-232C (V.24).

Для занимающихся компьютерной графикой и живописью предназначена схема светового пера и распечатка работающего с ним графического редактора LP48K.

Схема программатора для ПЗУ с УФ стиранием позволит любителям технического творчества самостоятельно программировать такие микросхемы как 2716, 2764, 27128, 27256, 573 PФ2, 5, 4, 5, 8.

Объем МР1 - 34 страницы, стоимость - 30 рублей.

По времени начала ее распространения (с сентября 1990г.) в адрес НТК не прекращается поток писем с положительными отзывами. Все схемы, входящие в разработку уже повторены многими пользователями, которые отмечают простоту изложения материала, доступность элементной базы и высокую повторяемость схем.

Методическая разработка (МР) N2.

Вторая методическая разработка МР2 состоит из двух частей. Первая часть посвящена контроллеру накопителя на гибких магнитных дисках НГМД. Контроллер полностью совместим с фирменной системой BETA-DISC INTERFACE. Он выполнен на базе микросхемы 1818ВГ93 (WD1293) и поддерживает работу до 4х НГМД - 3-х или 5-ти дюймовых, односторонних и двусторонних, 40 и 80 дорожечных. Разработка содержит подробное описание схемы, сигналов, принципов работы интерфейса, рекомендации по сборке и наладке, систему команд дисковой операционной системы TRDOS.

О преимуществах работы с НГМД по сравнению с работой с магнитофонной лентой знают все, это новый уровень работы с компьютером, и мы надеемся, что наша разработка поможет Вам перейти на этот уровень.

Вторая часть МР2 посвящена стыковке компьютера с бытовыми телевизорами. Наиболее качественное цветное изображение получается при стыковке компьютера по R-G-B входу телевизора. При стыковке по антенному входу с кодировкой по системе SEKAM или PAL происходит некоторая потеря качества цветного изображения, но зато телевизор не требует переоборудования. В некоторых случаях может удовлетворить стыковка через видеовход с чернобелыми или цветными телевизорами без передачи информации о цвете, т.е. в черно-белом изображении.

Вторая часть МР2 содержит схемы всех перечисленных способов стыковки компьютера с телевизорами, причем схемы предназначены не только для полупроводниковых, но и для ламповых телевизоров.

Стоимость МР2 - 25 руб.

Методическая разработка (МР) N3.

Третья методическая разработка (МР3) подготовлена по многочисленным письмам и пожеланиям пользователей "Спектрума". В нее входит принципиально новый способ русификации "Спектрума", выгодно отличающийся от известных способов.

Знакогенератор кириллицы размещается в неиспользуемой области ROM, там же

размещены некоторые подпрограммы, которые добавляет в операционную систему "Спектрума" два новых регистра клавиатуры R и N к уже имеющимся K, L, C, E, G. Таким образом, в любой строке могут быть представлены буквы как русского, так и латинского алфавита одновременно, т.е. например, операторы Бейсика на английском языке, а текст на русском.

Также приведены дополнения к программам поддержки интерфейсов принтеров, опубликованных в нашей МР1 для вывода на печать смешанного русско-латинского текста,

Кроме этого, в МР3 по просьбам пользователей включены схемы интерфейсов джойстиков, рассчитанных на двух игроков INTERFACE 1 и SINCLAIR, а так же схема программируемого джойстика, позволяющая запрограммировать все положения рукоятки и кнопки в соответствии с клавишами клавиатуры и более простой вариант механически программируемый (путем перестановки перемычек) джойстик.

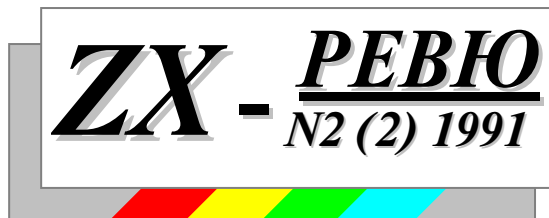
Стоимость МР3 - 25 руб.

Все программы, входящие в разработки, приведены в виде листинга и блоков шестнадцатиричных кодов.

Свои заявки на приобретение МР1-МР3 Вы можете направлять по адресу:

127566, Москва, И-566, НТК "Плюс". Оплата производится только наложенным платежом.

Просьба при всех обращениях к нам прикладывать к письму конверт с обратным адресом. В заявке необходимо указать свой полный адрес, фамилию, имя, отчество.



НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ИНФОРКОМ", 107241, МОСКВА, Б-241

ДОРОГОЙ ЧИТАТЕЛЬ

ИНФОРКОМ представляет Вам второй выпуск ZX-РЕВЮ. Истекший месяц принес нам очень много отзывов со всех концов страны. Мы благодарим всех, кто прислал свои теплые отзывы в наш адрес, но должны сообщить, что перед нами возникла проблема. С самого начала своей работы "ИНФОРКОМ" всегда отвечал на все письма своих клиентов, независимо от того, что там было - вопрос, просьба, консультация или критика (если был приложен заполненный конверт). Нашими бесплатными консультациями воспользовались только за последний год более двух тысяч человек. Увы, теперь, с выпуском ZX-РЕВЮ наша почта выросла в несколько раз, и теперь мы не можем гарантировать ответ даже на пятую часть поступающих к нам писем.

Вместе с тем, очень часто вопросы наших читателей повторяются и поэтому те из них, которые представляют общественный интерес, мы будем освещать на страницах ZX-РЕВЮ.

Судя по тому, что процентов 80 наших читателей, выписавших неполный комплект, после получения первого выпуска произвели доплату до полного комплекта и сообщили нам об этом, содержанием они остались довольны. Мы постараемся не обмануть их ожиданий.

Вместе с тем, почтовые переводы по-прежнему поступают крайне медленно. Специально поставленное для такой работы подразделение Моспочтамта (12-ый цех) задерживает Ваши переводы по 3 - 4 месяца. Мы в феврале получали переводы, отправленные в августе - октябре. Поэтому, чтобы не подвести тех наших читателей, которые дооформили заказ до полного, но сведения об этом до нас еще не дошли, мы 2-ой и 3-ий выпуски, также как и 1-ый вышлем ВСЕМ В ПОЛНОМ СОСТАВЕ.

Среди читателей ИНФОРКОМа очень много военнослужащих Советской Армии и Военно-морского флота. Много также курсантов, есть и ветераны Великой Отечественной, не говоря уже о наших воинах-интернационалистах. Всех их мы сердечно поздравляем с праздником 23 февраля и сообщаем, что, по нашим данным, именно им принадлежит заслуга в продвижении бытовой компьютерной техники и компьютерной грамотности в особо удаленные уголки нашей Родины.

РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ

БЕЙСИК ШАГ ЗА ШАГОМ

Этот раздел для начинающих подготовлен нашим читателем из г. Дубна тов. Скитевым Б.Н. Мы планируем сделать эту рубрику постоянной и приносим ему благодарность за предоставленный материал.

```
10 PRINT "*";  
20 GO TO 10
```

Попробуйте вместо точки с запятой ввести запятую или вообще ничего не вводить.

```
10 FOR A = 1 TO 704  
20 PRINT "*";  
30 NEXT A
```

Экран может содержать не более 704 символов, так как 32x22=704


```

40 PAUSE
50 FOR S = 5 TO 16
60 PRINT AT S, 10; " "
70 NEXT S

```

Добавим к предыдущей программе эту, вырежем на экране окно. Отметим, что в строке 60- 12 пробелов.

```

10 BORDER 1
20 INK RND * 7
30 PRINT "SPECTRUM";
40 GO TO 10

```

Выбираем цвета бордюра (возможны цифры от 0 до 7). Компьютер случайным образом выбирает цвет знаков.

```

10 LET A = RND*255
20 LET B = RND*175
30 PLOT INK RND*6;A,B
40 GO TO 10

```

Заполняем экран точками, координаты точек генерируются случайным образом командой RND.

Мозаика из разноцветных квадратов.

```

20 BORDER 1
30 INK RND * 6
40 PRINT "■":
70 GO TO 20

```

Квадрат в строке 40 рисуем в графическом режиме.

```

10 LET C = 0
50 IF C = 703 THEN STOP
60 LET C = C + 1

```

Если Вы хотите, чтобы программа остановилась сразу после заполнения экрана, то введите эти строки просто дописав их.

```

10 BORDER 0
20 PAPER 5
30 CLS
40 LET X = 1
50 FOR L = 0 TO 21
60 FOR C = 1 TO 7
70 PRINT INK C; AT L,C+X; "■"
80 NEXT C
90 LET X = X + 1
95 NEXT L

```

Рисуем "лесенку", цвет бордюра и экрана можно выбрать любым.

```

10 CLS
20 FOR N = 1 TO 1000
30 LET C = RND*6
40 LET X = RND*21
50 LET Y = RND*31
60 LET A$ = CHR$(RND*133)+32)
70 PRINT AT X,Y; INK C: A$
80 NEXT N

```

Заполнение экрана случайным образом, случайными символами (конфетти). Программа остановится после того, как выведет на экран 1000 символов.

Бегущая буква.

```

10 FOR F = 0 TO 3;
20 PRINT AT 10, F; "A"
30 PAUSE 5
35 BEEP .01,20

```

```
40 PRINT AT 10,F:" "
50 NEXT F
```

Буква пробегает только одну строку 10. Попробуйте заставить букву пробегать другую строчку или несколько строк.

Горизонтальные цветные полосы.

```
10 FOR N = 1 TO 2
20 FOR M = 0 TO 7
30 DIM P$ (32)
40 PRINT PAPER M; P$
50 NEXT M
60 NEXT N
```

Вертикальные цветные полосы.

```
10 FOR F = 7 TO 0 STEP -1
20 PRINT INK F: " "
30 NEXT F
40 GO TO 10
```

Вводим 4 квадрата в графическом режиме.

Геометрические узоры.

```
10 FOR A = 127 TO 254 STEP 4
20 CLS
30 INK RND * 6
40 PLOT 60,40
50 DRAW 126,88, PI*(2*A-1)
60 NEXT A
```

Звук и цвет.

```
10 FOR F = 1 TO 30
20 BEEP .1, RND * 20 + 20
30 BORDER RND * 7
40 NEXT F
```

Звуки.

```
10 FOR X = 12 TO 36
20 BEEP .01,X
30 BEEP .01,24-X
40 NEXT X
```

Клавиатура - музыкальный инструмент.

```
10 LET P = CODE INKEY$
20 IF P = 0 THEN GO TO 10
30 BEEP .04, (P-30)/2
40 GO TO 10
```

Эта программа позволяет Вам самим рисовать на экране.

```
10 CLS
20 PLOT 25,25
30 LET X = 5
40 INPUT K$
50 IF K$ = "Q" THEN DRAW 0,X
60 IF K$ = "A" THEN DRAW 0,-X
70 IF K$ = "P" THEN DRAW X,0
80 IF K$ = "O" THEN DRAW -X,0
90 IF K$ = "C" THEN CLS
100 GO TO 40
```

Клавиша Q - вверх, клавиша A - вниз,
клавиша P - вправо, клавиша O - влево,
клавиша C - стереть все.

MACHINE CODE

2. ПРОГРАММА ДЕКОМПРЕССИИ

Продолжение. Начало на стр. 6

Если Вы внимательно разобрали программу компрессии, то Вам не составит труда понять работу и этой программы, т.к. здесь зачастую повторяются некоторые фрагменты, которые мы подробно рассмотрели в прошлом выпуске.

Приступаем к работе.

	LD HL, 32768	; В HL-адрес скомпрессированного блока.
	LD BC,DLIN	; DLIN - длина блока, взятая из программы компрессии.
	LD DE, 16334	; Адрес экрана в паре DE.
L0	DEC C	; Проверка BC на 0.
	INC C	; Если 0, т.е. все
	JR NZ,L1	; коды декомпресси-
	DEC B	; рованы, то
	INC B	; возврат.
	RET Z	; -
L1	LD A,(HL)	; В акк-р байт из скомпрессированной области.
	LDI	; Перезапись байта в экран.
	CP (HL)	; следующий = предыдущему?
	JR NZ,L0	; если нет, то на L0
	LDI	; перезаписать еще 1 байт в экран.
	PUSH BC	; BC сохраняем в стеке.
	LD C,(HL)	; берем новое зна-
	INC HL	; чение из области
	LD B,(HL)	; скомпрессированной,
	INC HL	; оно эквивалентно числу копирования
		; байта A в экран.область.
L4	DEC C	; проверка BC на 0.
	INC C	;
	JR NZ,L2	;
	DEC B	;
	INC B	;
	JR Z,L3	; если 0, то идти на восстановление старого BC,
		; т.к. мы уже скопировали A определенное число раз.
L2	LD (DE),A	; A копировать в экран еще раз.
	INC DE	; счетчик адресов в экране увеличивается на 1.
	DEC BC	; счетчик количества оставшихся
		; копий аккумулятора уменьшить на 1
	JR L4	; идти на проверку BC.
L3	POP BC	; восстановить прежнее BC.
	DEC BC	; увеличить BC на Е т.к. мы уже сняли 2 байта,
		; которые указывали кол-во копий A.
	JR L0	; идти на L0

Теперь оттранслируем эту программу с адреса 31000.

Поле сод. памяти метки мнемокод
памяти

7916	21 00 80	LD HL,32768	
791B	01 91 0C	LD BC,DLIN	; Здесь в качестве параметра DLIN стоит число 0C91, что конечно же является частным случаем. У Вас может стоять другое число, зависящее от того, что Вы получите после того, как отработает программа компрессии.
791E	11 00 40	LD DE,16384	
7921	0D	L0 DEC C	

79E2	0C		INC C
7923	20 03		JR NZ, L1
7925	05		DEC B
7926	04		INC B
7927	C8		RET Z
7928	7E	L1	LD A, (HL)
7929	ED A0		LDI
792B	BE		CP (HL)
792C	20 F3		JR NZ, L0
792E	ED A0		LDI
7930	C5		PUSH BC
7931	4E		LD C, (HL)
7932	23		INC HL
7933	46		LD B, (HL)
7934	23		INC HL
7935	0D	L4	DEC C
7936	0C		INC C
7937	20 04		JR NZ, L2
7939	05		DEC B
793A	04		INC B
793B	38 05		JR Z, L3
793D	12	L2	LD (DE), A
793E	13		INC DE
793F	0B		DEC BC
7940	18 F3		JR L4
7942	C1	L3	POP BC
7943	0B		DEC BC
7944	0B		DEC BC
7945	18 DA		JR L0

Загрузите в память блок кодов, сброшенный программой компрессии, сделайте RANDOMIZE USR 31000. На экране та же картинка, но на ленте она занимает меньше места.

В заключении хочется отметить, что данный алгоритм не всегда приводит к требуемому результату (когда много последовательностей, где одинаковых 2-3 байта). В этом случае блок может даже увеличиться, но такие ситуации крайне редки, и как показала практика применения, обычно эффективность работы данного алгоритма компрессии высока и размер компрессированного блока более чем в два раза меньше исходного.

MEGA-BASIC

Продолжение. Начало на стр. 3.

Клавиши, определяемые пользователем (UDK)

В МЕГАБЕЙСИКе есть возможность запрограммировать верхний ряд клавиш для выполнения определенных функций. Для этого применяется команда KEY_, за которой должны быть записаны числовое и символьное выражения. Числовое выражение определяет какая из клавиш программируется, а символьное - задает значение этой клавише. Записывая ENTER (CHR\$13) в конце символьного выражения, Вы обеспечиваете тем самым автоматическое выполнение назначенной команды после нажатия заданной клавиши UDK. В МЕГАБЕЙСИКе уже встроены некоторые UDK:

VERIFY - здесь в символьном выражении уже записана последовательность 'RUN+CHR\$13'. При нажатии этой клавиши выполняется очередная программа.

VAL\$ - в символьном выражении здесь записано 'LOAD ""': RUN+CHR\$13'. При нажатии этой клавиши включается загрузка и автоматический запуск очередной записанной на кассете программы.

Клавиши управления

Во время выполнения программы клавиша "ПРОБЕЛ" может быть использована как новая SHIFT-клавиша для выполнения следующих функций:

CONTROL F- вызов фронт-панели.

CONTROL E- останов выполнения программы, выдача на экран сообщения и переход к режиму редактирования строк.

CONTROL R- останов программы, выставление начальных значений PAPER, INK, BORDER и переход к редактору строк.

2. Вывод на экран

Окна

Размер символов, их форма, а также способ их вывода на экран значительно расширяют в МЕГАБЕЙСИКе возможности "СПЕКТРУМа". Например, в МЕГАБЕЙСИКе возможно ограничить площадь экрана для выходной информации. Эта площадь может быть любого размера. Отдельные участки экрана определяются как "окна". Каждое окно имеет свой номер. Язык позволяет Вам работать с 10 окнами, которые пронумерованы от 0 до 9, и все могут быть использованы в программах. Правда, как мы уже отмечали выше, некоторые окна уже закреплены за определенными операциями, например все операторы PRINT в программе используют для выдачи информации окно 2, но у Вас есть возможность перейти на другое окно с помощью команды CURRENT_. За этой командой должно идти числовое выражение, которое определяет номер назначенного окна. Если результат этого числового выражения оказывается больше 9, то выдается сообщение о том, что окно задано неправильно:

ILLEGAL WINDOW

Окна могут иметь любой размер и занимать произвольное положение на экране. Для определения размера и положения окна используется команда WINDOW_y,x,d,w

y -номер строки верхнего левого угла окна (0...23);

x-номер столбца верхнего левого угла окна (0...63);

d-высота окна в строках;

w-ширина окна в столбцах;

Таким образом, в МЕГАБЕЙСИКе весь экран разбивается на 24 строки и 64 столбца. Функции ATTR и SCREEN\$ используют старую координатную систему, а PRINT AT - новую. Кроме того, координаты PRINT AT задаются относительно левого верхнего угла окна, а функции SCREEN\$ и ATTR по-старому используют абсолютные координаты экрана. Если

координаты окна превышают возможные размеры экрана, то Вы получите сообщение, что окно слишком велико, т.е. если 'y+d>24' или если 'x+w>64', то следует сообщение

WINDOW TOO LARGE,

Если же значения d и w меньше нуля, то следует сообщение

WINDOW TOO SMALL.

После выполнения команды WINDOW, очередной позицией для вывода на экран становится верхний левый угол окна.

Команды CLS и CLW_

Команду CLS, используемую для чистки всего экрана, нельзя использовать для чистки только одного окна. Для этого необходимо использовать команду CLW_. За этой командой должны быть записаны одно или два числа. При двух записанных числах первое определяет номер окна, а второе - режим CLW_:

CLW_n,0 - затухевка окна цветом PAPER;

CLW_n,1 - затухевка окна цветом INK;

CLW_n,2 - инвертирование. Цвет PAPER меняется на цвет INK;

CLW_n,3 - очистка атрибутов без изменения содержимого окна. Команда CLW_ всегда использует текущие атрибуты. Так, например, если текущим является окно 3, то после выполнения команды CLW_0,0 окно 0 будет затухевано с использованием атрибутов окна 3.

Команды PAN и SCROLL.

В YSMB имеется возможность перемещать пиксел за пикселом очередное окно экрана во всех направлениях. Команда PAN_ позволяет перемещать окно по горизонтали, а команда SCROLL_ - по вертикали. После этих команд должны быть записаны два числа: первое определяет цвет кромки экрана (1-цвет INK, 0-цвет PAPER), а второе показывает, в каком направлении будет смещение и на какую величину. Для команды PAN_: если второе число положительное, то перемещение выполняется вправо, если отрицательное, то влево. Для команды SCROLL_: положительное число - перемещение вверх, отрицательное - вниз. Возможно также содержимое экрана прокручивать. Для этого используют команды PANW_ и SCROLLW_. Параметры этих команд точно такие, как и для команд PAN_ и SCROLL_. Все эти команды действуют только на дисплейный файл.

Задание окон для конкретных целей осуществляется по команде FX_, за которой следует 2 числа: первое указывает, для каких целей предназначено это окно, а второе - определяет номер окна.

FX_0,n - определяет окно n для входной информации и сообщения об ошибках;

FX_1,n - определяет окно n для автоматического листинга;

FX_2,n - определяет окно n для выходной информации;

FX_3,n - определяет окно n для фронт-панели.

К примеру, запись FX_0,5 определяет, что окно 5 будет использоваться для входной информации и сообщения об ошибках. При желании наберите и запустите следующую программу для демонстрации окон.

```
10 BORDER 0
20 CURRENT_0: WINDOW_0,0,15,32
30 PAPER 3: INK 7: CLW_0
40 CURRENT_1: WINDOW_0,0,32,22,32
50 PAPER 1: INK 7: CLW_0
60 CURRENT_2: WINDOW_15,0,7,32
70 PAPER 7: INK 1: CLW_0
```

Команда MODE_

Используя команду MODE, можно получить 4 различных размера символов, выводимых на экран:

MODE_n,1 - каждый символ записан в матрицу 4x8 пиксел;

MODE_n,2 - обычный размер символов, т. е. 8x8 пиксел;

MODE_n,3 - символы двойной высоты 8x16 пиксел;

MODE_n,4 - символы размером 16x16 пиксел.

Как Вы видите, за командой MODE_ должно быть записано одно или два числа.

Первое число указывает номер окна, а второе - размер символов. Для каждого окна могут быть заданы свои размеры символов. При использовании команды `MODE_n,4` возможно получение символов с оттенками, так как в этом случае каждый пиксел символа занимает 4 пиксела экрана. Задание интенсивности тени осуществляется по команде `STIPPLE_`, после которой записывают число (0...15) определяющее интенсивность тени.

(Продолжение следует)

128K

Продолжение, начало на стр. 4

Впечатывание страниц.

Следующий вопрос, который мы рассмотрим, - это как дать компьютеру указание о том, какую страницу ПЗУ или какую страницу ОЗУ нужно впечатать.

Это выполняется с помощью команды `OUT`. Ключом же к этой операции служит внешний порт `7FFD`. Вы можете представить себе его как некий дополнительный регистр процессора, поскольку там может находиться число от 0 до 255. Как в ячейку памяти, туда можно отправить новое число, но в отличие от нее, считать число оттуда нельзя. В машинном коде выдача байта `n` по этому порту запишется так:

```
LD    A, n
LD    BC, 7FFD
OUT   (C), A
```

Число, выдаваемое через этот порт и определяет какая страница ПЗУ и ОЗУ будет впечатана. На рис.1 показано, какие функции выполняют отдельные биты байта, выдаваемого на порт `7FFD`.

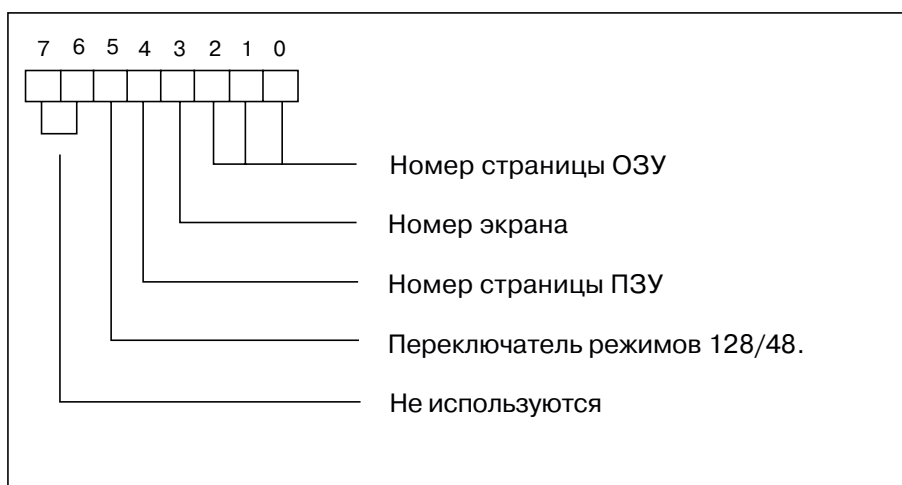


Рис.1

Как видно из рисунка, младшие три бита задают страницу ОЗУ (от 0 до 7). Четвертый бит задает страницу ПЗУ (0 или 1). Про прочие биты мы поговорим чуть позже.

В то же время, использование порта `7FFD` доставляет одну маленькую проблему. Выдачей байта на этот порт можно переключать страницы, но поскольку нельзя прочитать значение, там находящееся, то и нельзя определить, какие же именно страницы в данный момент впечатаны.

Чтобы уйти от этой трудности, `SPECTRUM 128` оставляет копию байта, выданного на этот порт в одной из своих новых системных переменных (которых не было в `SPECTRUM 48`). Эта системная переменная называется `BANK_M` и хранится в ячейке памяти с адресом `5B5C`. Теперь Вы можете, прочитав содержимое этого адреса, определить, какие страницы у Вас впечатаны.

В то же время, это означает, что когда Вы выдаете байт на порт `7FFD`, Вам следует

применять меры и для того, чтобы отправить в BANK_M его копию, в противном случае, когда управление от Вашей программы будет передано ПЗУ, произойдет зависание или сброс компьютера.

Кроме того, процедуры обработки прерываний, расположенные в адресах 00038 и 10038 требуют, чтобы по крайней мере бит 4 системной переменной BANK_M был выставлен точно. Это означает, что прежде чем заниматься переключением страниц ПЗУ, Вы должны отключить обработку прерываний. Вдобавок к этому, при выходе из процедуры обработки прерываний значение, содержащееся в BANK_M выдается на порт 7FFD, так что если они не совпадают, то возможны неприятности. Это означает, что при переключении страниц ОЗУ (или смене экранов, о чем чуть позже) Вы должны либо отключать прерывания, либо сначала загружать нужный байт в BANK_M, а только потом выдавать его на порт 7FFD и не наоборот. Если этого не сделаете, то при первой же обработке прерываний компьютер зависнет.

Экранная память

Как и в "Спектруме 48", здесь экран хранится, начиная с адреса 4000H. В предыдущем выпуске мы говорили, что за экраном постоянно закреплена пятая страница ОЗУ, т.е. 4000H - это то же самое, что и 5C000.

В свою очередь "Спектрум 128" имеет не одну, а две экранные области памяти, хотя даже сопроводительные руководства по эксплуатации не всегда об этом упоминают.

Оба экрана имеют в памяти свои адреса, но очевидно, что выдаваться на экран они могут только по одному. Традиционная экранная область, выдаваемая при нормальных обстоятельствах, называется SCREEN0, а альтернативная SCREEN1 и хранится SCREEN1 в адресах, начиная с 7C000, т.е. начиная с адреса C000 на странице 7.

Та экранная область, которая в данный момент отображается на экране телевизора, называется активной. Обычно активным экраном является SCREEN0, но Вы можете это изменить из машинного кода включением третьего бита в системной переменной BANK_M и выдачей соответствующего байта на порт 7FFD. После этого на экране будет изображаться содержимое экранной области памяти SCREEN1.

Черно-белая часть изображения хранится в адресах, начиная с 7C000 по 7D7FF, а атрибуты цвета в адресах с 7D800 по 7DAFF.

Очевидно, что страница 7 должна быть впечатана, чтобы экран SCREEN1 мог сменить экран SCREEN0 но совсем нет необходимости, чтобы страница 7 оставалась впечатанной, чтобы SCREEN1 оставался активным. Может быть впечатана любая страница ОЗУ и, тем не менее, экран SCREEN1 будет изображаться на экране до тех пор, пока он остается активным и убрать его можно будет только активизацией экрана SCREEN0.

Можно переключать текущий экран и из Бейсика, не выходя в машинный код. Команда POKE 23388, 24 активизирует SCREEN1, а команда POKE 23388, 16 активизирует SCREEN0.

Если Вы попробуете это сделать, то получите черный экран, т.к. все байты атрибутов у экрана SCREEN1 содержат нули. К сожалению, из Бейсика нельзя печатать на экране SCREEN1, это можно делать только процедурами, записанными в машинном коде. Тем не менее, следующая техника может быть полезной при загрузке программ:

```
POKE 23388, 24:  
LOAD "" SCREEN$  
POKE 23388, 16
```

Работает эта программа так: экран гаснет и начинается загрузка в память "картинки", но при этом на экране ничего не изображается. После окончания загрузки вся картинка высвечивается на экране целиком и мгновенно.

Имейте также в виду, что любое сообщение Бейсика об ошибке (и даже 0: OK) автоматически реактивирует SCREEN0, поэтому если Вы забудете отключить SCREEN1, то компьютер во многих случаях сделает это за Вас.

Переключатель режимов

Порт 7FFD имеет еще один бит, который мы не рассмотрели - это бит 5, который в режиме 128K должен быть всегда выключен. Если он включен, то компьютер "запирается" в

режиме 48K, при этом постоянно включена первая страница ПЗУ и постоянно активен экран SCREEN0. В то же время, включение бита 5 еще не означает, что этого достаточно для перехода в режим 48K, поскольку машинный стек содержит адреса возврата, относящиеся к ПЗУ0, а поскольку компьютер "заперт", то страница 0 ПЗУ не может быть впечатана.

"Спектрум 128" имеет две новые команды Бейсика - это команды PLAY и SPECTRUM. Команда PLAY служит для воспроизведения музыки но поскольку отечественные модификации 128-го не имеют музыкального процессора, а владельцы фирменных машин имеют фирменные инструкции и могут все прочесть сами, то мы на ней останавливаться не будем, а рассмотрим команду SPECTRUM.

Эта команда имеет двойной эффект. Во-первых, по ней компьютер переключается в режим 48K, а во-вторых выдается сообщение 0:OK, при этом останавливается исполнение текущей программы.

СЕКРЕТЫ ПЗУ

Продолжение. Начало на стр.

2. Обслуживание клавиатуры.

Все процедуры, которые входят в этот раздел, служат для того, чтобы установить факт нажатия клавиши (или двух, если одна из них SHIFT) и после преобразований выдать код символа (токена или код управления), соответствующий данной клавише, в зависимости от того, в каком режиме эта клавиша была нажата (K,L,C,E).

Прежде, чем подробно рассматривать входящие сюда процедуры, кратко обрисуем логику их работы.

1. Сначала выполняется сканирование клавиатуры по 8-ми полурядам [1, стр. 90], а внутри каждого полуряда - по 5-ти столбцам (от внешней клавиши к внутренней). Сканируются 40 основных клавиш. Определяются порядковые номера нажатых клавиш (от 0 до 27H).

2. Клавиша считается нажатой только если продолжительность ее удержания составила не менее 5-ти циклов прерываний, а поскольку в течение этих 5-ти циклов могла быть нажата и другая клавиша, то организуются две группы системных переменных. В первой группе ведется счет 5-ти циклов, а во второй "ловится" следующая клавиша. Далее наоборот.

3. По порядковому номеру клавиши из таблицы 0205-022B определяется "главный код" нажатой клавиши. Это как бы код символа (буквы или цифры), написанного на этой клавише. Причем он соответствует заглавной букве (с CAPS SHIFT). Тот факт, что клавиша могла быть нажата без CAPS SHIFT или с SYMBOL SHIFT или в режиме "E", здесь пока не учитывается.

4. В зависимости от того, в каком режиме находилась клавиатура, по "главному коду" из таблиц 0205-028D определяется "окончательный код" клавиши, который уже представляет код символа (буквы или цифры), код токена ключевого слова или управляющий код (код управления курсором, код управления цветом, позицией печати и т.п.). Коды "Спектрума" см. [3].

По тексту статьи многократно встречаются ссылки на различные системные переменные. Их список и назначение можно найти в [3].

028E-02BE - Подпрограмма сканирования клавиатуры.

В результате работы этой подпрограммы регистр E возвращает значение от 0 до 27H, которое указывает на то, какая клавиша из 40 основных была нажата. Если никакая, то FFH.

Регистр D возвращает значение, которое указывает на то, какая SHIFT-клавиша была при этом нажата. Если никакая, то FFH. Если одновременно были нажаты обе SHIFT-клавиши, то D содержит значение для CAPS-SHIFT, а E - для SYMBOL-SHIFT. Если никакие клавиши не нажаты, то DE возвращает FFFFH.

Если по ошибке нажаты три клавиши, или если нажаты две клавиши ни одна из которых не SHIFT, то флаг нуля (Z-флаг регистра F) выключается.

Клавиатура условно разбивается на 8 рядов по 5 клавиш в каждом ряду.

Сначала клавишам присваивается значение 2F, а затем, сканируя по рядам, а внутри каждого ряда по столбцам, из этого числа вычитается по 8 за каждый столбец и по 1 за каждый ряд, тем самым на выходе для любой нажатой клавиши регистр E будет содержать соответствующий ей номер от 0 до 27H. В регистре D будет 28, если одновременно нажат CAPS SHIFT или 19, если нажат SYMBOL SHIFT.

Эта подпрограмма включает в себя следующие процедуры:

028E-0295 - KEY-SCAN

Выполняет начальные установки HL,DE,BC.

0296-029E - KEY-LINE

Чтение данных с порта клавиатуры (FE, см. [1, стр. 89]), определение столбца нажатой

клавиши.

029F - 02A0 - KEY-3KEYS

Выход, если нажаты сразу три клавиши.

02A1-02AA - KEY-BITS

Вычисление номера нажатой клавиши по ряду и столбцу.

02AB-02BE - KEY-DONE

Возврат к 028E, если не все ряды просканированы. Выход с установлением номера нажатой второй клавиши, если все.

02BF-030F - подпрограмма обслуживания клавиатуры.

Эта подпрограмма вызывается всякий раз, когда происходит маскируемое прерывание, т.е. нормально - каждые 20 мкс. определяет, какая клавиша была нажата и вводит ее окончательный код в системную переменную LAST-K, при этом включает бит 5 системной переменной FLAGS, что указывает на факт нажатия новой клавиши.

02BF-02C5 - KEYBOARD

Это точка входа. Отсюда вызывается KEY-SCAN, т.е. подпрограмма сканирования клавиатуры.

02C6-02D0 - K-ST LOOP

Проверка первой группы системных переменных KSTATE0...KSTATE3. Если она свободна, то переход на 02D1, а если занята, то счетчик 5-ти циклов уменьшается на 1.

02D1-02F0 - K-CH-SET

Здесь выполняется обработка системных переменных KSTATE0...KSTATE7.

Интересной особенностью является то, что они разделены на две группы - KSTATE0...KSTATE3 и KSTATE4...KSTATE7. Использование двух групп позволяет производить определение нажатой клавиши в одной группе, в то время как другая группа определяет удержание предыдущей клавиши. Группа освобождается для приема новой клавиши только если клавиша была нажата в течение не менее чем 5-ти циклов опроса клавиатуры, т.е. в течение не менее 1/10 доли секунды.

Пока одна из групп не освободится, переход к приему новой клавиши не произойдет.

02F1-0307 - K-NEW

Отсюда начинается обработка новой нажатой клавиши. Сначала инициализируются системные переменные KSTATE0...KSTATE3 (или KSTATE4...KSTATE5, в зависимости от того, какая группа свободна к приему).

В KSTATE0 (KSTATE4) поступает "главный" код нажатой клавиши.

В KSTATE1 (KSTATE5) устанавливается число 5 (для счетчика пяти повторений).

В KSTATE2 (KSTATE6) вводится значение системной переменной REPDEL (нормально 0.7 сек.).

В KSTATE3 (KSTATE7) после декодирования засылается "окончательный" код нажатой клавиши, зависящий от того, в каком регистре находилась клавиатура в момент нажатия. Декодирование выполняется вызовом процедуры K-DECODE по адресу 0333.

0308-030F - K-END

Финишные операции. В системную переменную LAST-K засылается код символа (токена), соответствующего нажатой клавише. Включается бит 5 системной переменной FLAGS как сигнал о том, что новый код принят.

0310-031D K-REPEAT

Вспомогательная процедура, вызывается более мощной K-CH-SET (02D1).

Процедура обрабатывает длительно нажатую клавишу. Если клавиша нажата более 0.7 сек. в первый раз (установлено системной переменной SEPDEL), то ее действие повторяется. В последующем ее действие повторяется, если она остается нажатой более 0.1 сек. (установлено системной переменной REPPER).

031E-032B - K-TEST

Вспомогательная процедура. Вызывается более мощной K-CH-SET (02D1). Служит для анализа результатов работы KEY-SCAN и выяснения по состоянию регистров D и E, была ли вообще нажата какая-либо клавиша, и если да, то с SHIFT или нет.

На выходе из нее регистр D обнулен, а информация о SHIFT-клавише, хранившаяся в нем, сохраняется теперь в регистре B.

032C-0332 - K-MAIN

Используя данные, содержащиеся в DE по результатам работы KEY-SCAN (0201) эта процедура отыскивает в таблицах клавиатуры (0205-028B) код символа (токена), соответствующий нажатой клавише.

Подпрограмма декодирования клавиатуры.

При входе в эту подпрограмму в регистре E содержится "главный" код нажатой клавиши, значение переменной FLAGS - в регистре D, значение MODE - в регистре C, а информация о нажатой SHIFT-клавише - в регистре B.

0333-0340 - K-DECODE

Распределяет обработку по процедурам, в зависимости от того, цифровая или буквенная клавиша была нажата и от состояния режима клавиатуры (MODE).

Если была нажата цифровая клавиша, то далее следует переход к процедуре K-DIGIT (0367).

Если клавиша буквенная, то в режиме "E" переход к процедуре K-E-LET (0341), а в режимах "K", "L", "C" - к процедуре K-KLC-LET(034F).

0341-0349 - K-E-LET

Если клавиатура находилась в режиме "E", то для поиска "окончательного кода" устанавливается базовый адрес таблицы 022C-0245, если никакая SHIFT-клавиша не была нажата, в противном случае поиск производится по таблице 0246-025F. После установки базового адреса для поиска в таблице вызывается процедура

K-LOOK-UP (034A).

034A-034E - K-LOOK-UP

По установленному базовому адресу (в регистре HL) и "главному" коду (в регистре DE) производит выборку из таблиц клавиатуры 022C-0245; 0246-025F; 0260-0269; 026A-0283; 0284-028D "окончательного" кода.

034F-0363 - K-KLC-LET

Выполняет поиск окончательного кода по главному коду для режимов клавиатуры "K", "L", "C".

Если клавиатура находилась в режиме "K", вызывается подпрограмма K-TOKENS (0364), где к "главному" коду прибавляется число A5H (десятиричное 165).

В режиме "L" сначала решается вопрос о том, была задействована при этом клавиша SYMBOL SHIFT или нет. Если да, то в HL вводится базовый адрес таблицы 026A-0263 и вызывается K-LOOK-UP. Если нет, то к главному коду добавляется 20H (десятиричное 32) - этот случай соответствует вводу строчной буквы.

0364-0366 - K-TOKENS

Определяет "окончательный" код для токенов, который отличается от "главного" на A5H (десятиричное 165).

Пример:

Буква A - код 65 десятиричный.

Токен NEW - 230;

Буква J - код 74,

Токен LOAD - 239, 0367-0381 K-DIGIT

Здесь рассматривается случай нажатия цифровой клавиши, а также клавиш ENTER (0DH), SPACE (20H) и обеих клавиш SHIFT (0EH). В последних трех случаях выполняется возврат, а для цифровой клавиши происходит следующее:

Если она нажата в режимах "K", "L", "C" - переход на процедуру K-KLC-DGT (039D).

Если в режиме "G" - переход в K-GRA-DGT (0389).

Если в режиме "E", то проверяется клавиша SYMBOL SHIFT. Если она нажата, т.е. вводится токен, то вводится базовый адрес таблицы 0284-028D и выполняется переход на K-LOOK-UP (034A) для вычисления "окончательного" кода. Если же SYMBOL SHIFT не нажата,

то для клавиш 0...7 это означает ввод цветового кода PAPER или INK в зависимости от состояния CAPS SHIFT. Особый случай здесь представляют клавиши 8 и 9, которые в режиме "E" задают цветовые атрибуты BRIGHT и FLASH, что и анализируется в следующей процедуре K-8-&-9.

0382-0388 - K-8-&-9.

0389-039C - K-GRA-DGT.

Здесь устанавливаются коды символов блочной графики (символы с 80H по 8FH), а также коды перехода в графический режим GRAPHICS (0FH), и код DELETE (0CH).

039D-03B1 - K-KLC-DGT.

Выдает "окончательный" код цифрового символа, записанного на клавише, но для символа "@" выполняет переход на 03B2.

03B2-03B4 - K-@-CHAR.

Выдает "окончательный" код символа @.

3. Управление звуковым динамиком

Динамик управляется состоянием 4-го бита в команде OUT, использующей порт 254 (FE HEX).

Рассмотрим ноту "ДО" первой октавы, которая имеет частоту 261,63 Гц. Чтобы динамик воспроизвел эту ноту, данный бит должен поочередно включаться и выключаться каждую 1/523.26 долю секунды.

В стандартном "Спектруме" применен кварц с частотой 3,5МГц и для получения ноты "ДО" необходимо, чтобы команда OUT выдавалась процессором после каждых 6689 тактов. Задержку для этого выполняет замедляющий цикл, организованный в подпрограмме BEEPER.

Подпрограмма BEEPER

При входе в эту подпрограмму пара DE должна содержать произведение частоты звукового сигнала на время его звучания (в секундах):

$f \times t$

Пара HL, в свою очередь, должна содержать рассчитанную продолжительность замедляющего цикла, измеренную в тактах, деленных на четыре, минус число 30.125 которое компенсирует естественную задержку, возникающую при исполнении машинокодовых команд.

Пример:

для воспроизведения ноты "ДО" в течение 1 секунды.

$f = 261.63 \text{ Гц}; t = 1 \text{ сек},$

$\text{INT}(f \times t) = 261 = 0105 \text{ HEX}$

в регистре DE - 0105.

$6689/4 - 30,125 = 1642 = 066A \text{ HEX}.$

В регистре HL - 066A.

Подпрограмма BEEPER содержит следующие процедуры.

03B5-03D0 - BEEPER.

Выполняет исходные установки и начинает отсчет 16-х долей потребного количества тактов процессора (вместо 4-х долей). Возникающий при этом остаток от 0 до 3 компенсируется последующим переходом в соответствующую точку от 03D1 до 03D4.

03D1- BE-IX+3 здесь происходит

03D2- BE-IX+2 компенсация

03D3- BE-IX+1 возникшего

03D4- BE-IX+0 остатка.

03D6-03F1 - BE-H&L-LF,

Здесь организуется первый полуцикл колебания динамика.

03F2-03F5 - BE-AGAIN.

Организация второго полуцикла.

03F6-03F7 - BE-END.

Завершение подпрограммы.

Подпрограмма ВЕЕР (03F8-046B)

Эта подпрограмма составлена частично в машинном коде, а частично в коде калькулятора. Служит для выдачи звукового сигнала заданной высоты и заданной продолжительности из интерпретатора БЕЙСИКа.

Математическими расчетами здесь занимаются процедуры, записанные в кодах калькулятора, а необходимыми проверками - машинокодовые процедуры. Точка входа - 03F8. После исполнения всех необходимых расчетов, подпрограмма ВЕЕР использует для выдачи звука подпрограмму ВЕЕЕР как процедуру низшего уровня.

При входе в подпрограмму на стеке калькулятора должна находиться высота ноты, а под ней число, представляющее длительность звучания. Оба числа должны быть записаны в интегральной упакованной форме, иначе будет вызвана процедура 046C для выдачи сообщения об ошибке. Подробнее о работе калькулятора, об операциях со стеком и об интегральной и интегральной упакованной формах записи чисел с плавающей точкой см. в наших разработках [1,2,3].

046C-046D - REPORT-B.

Процедура выдает сообщение об ошибке "INTEGER OUT OF RANGE", если числа на вершине стека калькулятора заданы неверно. Обработка ошибки выполняется вызовом RST 0008 с подачей кода-перехватчика (hook-кода)0A.

046E-04A9 - SEMI-TONE TABLE.

Здесь в этой таблице записаны в интегральной (пятибайтной) форме рассчитанные значения частот для 12-ти основных музыкальных полутонов первой октавы (ДО, ДО-диез....СИ).

04AA-04C1 - "PROGRAM NAME".

Эта процедура не имеет отношения к "Спектруму". Она осталась в ПЗУ как наследство от его предшественника компьютера ZX-81 и не была удалена своевременно при разработке ПЗУ.

* * *

Продолжение следует.

В следующем выпуске мы начнем рассмотрение процедур, обслуживающих ввод/вывод с кассетного магнитофона.

Литература

- [1]. Первые шаги в машинном коде Z-80.
- [2]. Практикум по программированию в машинных кодах Z-80.
- [3]. Справочник по программированию в машинных кодах Z-80.
- [4]. Большие возможности Вашего "Спектрума".

МАЛЕНЬКИЕ ХИТРОСТИ

В прошлом выпуске в этом разделе мы предложили Вашему вниманию несколько любопытных задач. Все они используют способность "Спектрума" выполнять логические вычисления. Сегодня же подробно разберем пути использования этих способностей компьютера в ваших программах.

Рассмотрим, например, такой фрагмент какой-либо программы.

```
100 IF INKEY$="8" THEN LET x=x+1
110 IF INKEY$="5" THEN LET x=x-1
```

Такая логика часто встречается, когда Вы хотите, например, чтобы при нажатии клавиши "8" объект двигался по экрану вправо, а при нажатии клавиши "5" - влево. Занимает такой фрагмент 48 байтов и далеко не оптимален.

Рассмотрим другой пример:

```
100 LET x=x+(INKEY$="8")-(INKEY$="5").
```

Это совершенно то же самое, но занимает всего 29 байтов и, самое главное, намного быстрее работает.

Вся хитрость в том, что те выражения, которые записаны внутри скобок, являются логическими. Если результат логического отношения - "ИСТИНА", то логическое выражение имеет значение 1, а если "ЛОЖЬ" - 0.

Поскольку функция INKEY\$ не может считывать две клавиши одновременно, то либо была нажата клавиша "8", либо "5", либо ни одна из них. Тогда либо первая скобка равна 1, либо вторая, либо ни одна. Фактически мы в этом примере прибавили к числовой переменной x результат логического выражения. Надо сказать, что не всякий язык программирования позволил бы такую вольность. Паскаль бы, например не позволил, да и БЕЙСИК не всякий, но БЕЙСИК "Спектрума" - позволяет.

Продолжим с нашим примером. При изображении на экране движущегося объекта бывает важно так ограничить его перемещение, чтобы он не вышел за пределы экрана. Обычно это делают так:

```
120 IF X>31 THEN LET x=31
130 IF X<0 THEN LET X=0
```

Вот дополнительный расход 54 байтов, а все можно сделать проще и лучше путем:

```
100 LET x=x+(INKEY$="8" AND x<31)-(INKEY$="5" AND x>0)
```

Здесь требуется всего 46 байтов вместо 102!

Существенная проблема с использованием функции INKEY\$ состоит в том, что с ее помощью трудно организовывать перемещение по диагонали, поскольку она читает только одну нажатую клавишу. Решить проблему можно с помощью оператора IN, читая клавиатуру, как внешний порт:

```
10 LET x=x-(IN 49150 + IN 57342 <>510)-(IN 64510 + IN 65022<>510): LET y=y+(IN 65278+IN
32706<>510)-(IN 63486+IN 61438<>510)
```

Клавиатура делится на 4 части. Верхний ряд служит для перемещения вверх, нижний - вниз, а левая и правая половины средних рядов - влево и вправо.

Одновременным нажатием двух клавиш можно получить движение по диагонали.

Другой пример:

```
10 LET x=a=b
```

На первый взгляд в этой записи мало смысла, но это только на первый взгляд. Здесь ясно видны два знака равенства, а вот означают они совершенно разные вещи. Первый - часть оператора присвоения, то есть вместе с LET он присваивает переменной x результат логического выражения a=b. А второй знак равенства - это логическое отношение "РАВНО" и, тем самым, является частью логического выражения a=b, которое может быть либо истинным, либо ложным и, соответственно, иметь в качестве результата либо 0, либо 1. Вся запись заняла 11 байтов, в то время как его традиционный эквивалент имел бы 38 байтов:

```
10 IF a=b THEN LET x=1
20 IF a<>b THEN LET x=0
```

А если надо, чтобы при a=b, x равнялся бы десяти, тогда так:

```
10 LET x=(a=b)*10
```

В программах такую систему иногда используют для организации подсчета очков. Например, Вам надо, чтобы при правильном ответе (при нажатии заданной клавиши n) к счету добавлялось бы 50 очков:

```
10 LET s=s+(INKEY$=STR$ n)*50
```

Не забудьте только, что число n надо сначала преобразовать в стринг через STR\$, поскольку иначе возникнет несовпадение типов, ибо INKEY\$ - всегда стринг (строковая, иногда еще говорят литеральная, переменная).

Очень часто в программах для создания мультипликационных эффектов используют быстрое чередование двух каких-то символов, например символов из области графики пользователя. Первый, допустим, изображает бабочку со сложенными крыльями, а второй - с расправленными. Удобнее всего для этого пользоваться оператором NOT.

```
510 LET a=1
520 LET a= NOT a
530 PRINT AT 10,15; CHR$(144+a)
540 IF INKEY$<>"" THEN RETURN
550 GO TO 520
```

Цикл между строками 520 и 550 будет продолжаться до тех пор, пока Вы не нажмете какую-либо клавишу и не выйдете через строку 540 в вызывающую подпрограмму.

Вы, должно быть знаете, что числа в БЕЙСИКе являются страшными расточителями памяти, поэтому там где можно от них уйти с помощью логических выражений, это лучше делать.

Вместо

```
10 IF x=1 THEN ...
```

Можно

```
10 IF x THEN ...
```

Вместо

```
20 IF x=0 THEN ...
```

Можно

```
20 IF NOT x THEN ...
```

А вот такую головоломку:

```
30 IF x<>1 AND x<>0 THEN LET x=1
```

можно упростить до:

```
30 LET x= NOT NOT x
```

Следующий пример хоть и не дает такой ощутимой экономии памяти, как большинство предыдущих, но зато демонстрирует оригинальный прием организации переключателей в программе и вообще открывает доступ новым идеям:

```
10 IF r=1 THEN PRINT "x"
20 IF r=2 THEN PRINT "y"
30 IF r=3 THEN PRINT "z"
```

Можно записать как:

```
10 PRINT ("x" AND r=1)+("y" AND r=2)+("z" AND r=3)
```

Еще одним полезным способом применения логических операторов может быть имитация структурных переключателей типа ON...GO TO... Такой оператор есть во многих языках программирования, в том числе и в некоторых диалектах БЕЙСИКа. Например, Вам необходимо иметь:

```
100 IF INKEY$="1" THEN GO TO 1000
110 IF INKEY$="2" THEN GO TO 2000
120 IF INKEY$="3" THEN GO TO 3000
```

Это можно легко исполнить путем:

```
100 GO TO VAL INKEY$*1000
```

Этот прием, однако, не сработает, если строки, на которые должен быть сделан переход, расположены нерегулярно, а также если по ошибке будет нажата не та клавиша. Если вам, например нужны переходы на строки 70, 1100 и 60, то это можно сделать следующей строкой:

```
100 GO TO (600 AND INKEY$="1")+ (1010 AND INKEY$="2") + (-40 AND INKEY$="3")+100
```

Внимательный читатель заметил, что мы число 100 вынесли отдельно. Это сделано для того, чтобы если нажата не та клавиша или не нажата никакая, то вновь выполнялась бы сотая строка.

FORUM

ОБЗОР ФЕВРАЛЬСКОЙ ПОЧТЫ

Вот с какой проблемой столкнулся наш читатель Клейменов из Целиноградской области. Его Синклер модели "Дубна 48K" не работает с программой TF COPY.

Мы вряд ли можем здесь чем-то помочь, но останавливаемся на данном факте не случайно. При таком огромном разнообразии модификаций Синклер-совместимых компьютеров, какое сейчас сложилось у нас в стране, не будет ничего удивительного, если начнут расти как снежный ком проблемы совместимости программного обеспечения. И самым радикальным лекарством против этого мог бы быть всеобщий форум, где каждый может высказаться и будет услышан. Если Вы поняли, что мы имеем в виду, то ждем Ваших писем - будем вместе анализировать и обобщать что не работает, как не работает и на каких моделях компьютеров.

Сразу три письма с просьбой помочь разобраться с игрой SABOTEUR 2. Может быть кто-либо из читателей поможет? Черкните пару строк для форума.

Татьяне Петровне К. очень нужно для занятий с детьми что-нибудь по Синклер-ЛОГО. Откликнется кто может, мы ей переправим. Одновременно пользуясь случаем поздравляем ее и всех других наших читательниц с приближающимся 8 Марта.

Кстати, дорогие товарищи учителя, если Вы сами друг другу не будете помогать, то никто Вам не поможет. "ИНФОРКОМ" готов Вас всех перезнакомить, пишите.

Как всегда в почте есть несколько писем о том, что в городе N-ске наши разработки можно купить на местном толчке и притом дешевле.

Упреки не очень заслуженные, потому что мы работаем над новыми разработками, а ребята с толкучки - нет. Мы до 85% отчисляем государству, а "толкучка" - нет. Трудно было бы ожидать, что мы сможем конкурировать с черным рынком. Хотя если совсем плохо станет и перестанем писать мы и другие такие как мы - откуда на толкучке возьмутся новые работы?

Владимир К. из Полтавы написал о том, что его друг получил рекламный лист кооператива "Электрон" (Зеленоград), в котором этот "Электрон" предлагает наши разработки, рекламируя их нашими же словами.

Уважаемый Владимир, нам все это хорошо известно, этим занимается не один десяток кооперативов, ну что с этим можно сделать? А "Электрон" довольно известная фирма, заслужившая репутацию серьезными разработками для гораздо более серьезных машин и мы полагаем, что его руководитель спросит как положено с того бездельника, который не сумел найти десятка слов для своего рекламного листка без унылого списывания с нашего.

Кстати, спасибо Вам Владимир за то, что Вы нам этот листок переслали.

Хотим обрадовать тех, кто интересовался русификацией МАСТЕРФАЙЛа. Мы получили несколько сообщений от тех, кто это сделал. Контакт установлен, теперь есть шанс, что в РЕВЮ появится статья для всех желающих сделать это самим.

Из города П. сообщают, что местный кооператив К. тиражирует наши сборники программного обеспечения и продает их по демпинговым ценам.

Это замечательно. Мы ведь не можем удовлетворить всех желающих, потому и цены на наши сборники так завышены. Честно говоря, мы на то и рассчитывали, что кто-то после нас сможет и сам заработать и спрос удовлетворить. А вот если этот кооператив К. напишет нам, то мы еще и рекламу ему в ZX-РЕВЮ дадим, и много за нее не попросим.

Как обычно много писем по программному обеспечению. У многих не работает копировщик защищенных программ LERM-7. Да, к сожалению, как выяснилось, эта программа крайне чувствительна к аппаратному окружению и отрицательно относится к архитектуре отечественных самодельных версий "Спектрума". Теперь в своих рекламных листах мы об этом предупреждаем. А на компьютерах 128K она не работает даже и на фирменных.

Многих заставил поволноваться пакет LASER BASIC. Дело в том, что после первых настроечных операций программа "сбрасывается" и появляется исходное сообщение Sinclair Research Ltd.

Вопросов по этому поводу так много, что видимо надо дать пояснение. Программа не сбрасывается. Просто из памяти удаляется БЕЙСИК-загрузчик, чтобы освободить область БЕЙСИКа для Вашей работы. Основная же машиннокодочная часть программы находится выше уровня RAMTOP, который был выставлен оператором CLEAR в БЕЙСИК-загрузчике и потому сброшена командой NEW быть не может. Попробуйте и Вы увидите, что после этого "сброса" работают операторы и функции ЛАЗЕР-БЕЙСИКа, а они начинаются с " точки" или с " вопросительного знака".

Другой "дежурный" вопрос - почему программа ULTIMON не загружается, а название программы читается не как Program, а как bytes?

Причина в том, что ULTIMON - не имеет традиционного БЕЙСИК-загрузчика и представляет собой блок машинных кодов. Машинный код, в отличие от БЕЙСИКа загружается не командой LOAD"", а командой LOAD ""CODE addr. Здесь addr - адрес, начиная с которого этот код должен быть загружен. Если его не указывать, то блок будет загружен в то место памяти, из которого он был когда-то выгружен, т.е. в данном случае неизвестно куда. Надо этот адрес знать, а поскольку мы пока инструкцию к этой программе не рекламировали, то говорить об этом рановато.

Зато в этом месяце мы начинаем распространение описания к программе для автоматизированной разработки дизайна игровых программ H.U.R.G., о которой нас тоже уже давно спрашивают.

И последнее, многие спрашивают описание БЕТА-БЕЙСИКа. Не спешите его приобретать. Нам хорошо известны те полуграмотные описания, которые ходят по стране. Вот закончим с МЕГА-БЕЙСИКом и дадим в "ZX-РЕВЮ" настоящее описание версий 1.0, 1.8 и 3.0. Это все уже готово и скоро будет в Вашем распоряжении. Материал огромный - около 150 машинописных страниц. Только это уже оправдало бы затраты на Вашу подписку.

А сейчас позвольте с Вами на месяц расстаться. До следующей почты.

* * *

Если Вы не являетесь зарегистрированным получателем ZX-РЕВЮ, но желаете войти в их ряды, то пока еще сделать это не поздно. Вы получите все выпуски, вышедшие до того, как Вы сделали заказ, и будете ежемесячно получать по новому выпуску в течение всего года.

Советуем поспешить. Свободное количество экземпляров уже ограничено, и есть признаки того, что к концу года ZX-РЕВЮ станет раритетом.

Пока еще стоимость подписки не изменилась и составляет на год (а на меньшие сроки мы заказы не принимаем) для частных лиц - 70 рублей.

Своих постоянных заказчиков "ИНФОРКОМ" будет также регулярно оповещать об имеющихся и вновь сделанных разработках.

Заказы направляйте по адресу 107241, Москва, Б-341, а/я 37 "ИНФОРКОМ". Здесь же укажите, когда и откуда был сделан перевод.

Для организаций при оплате по безналичному расчету действует повышенный тариф - 252 руб. за полный годовой комплект.

Этот тариф не распространяется на школы, Дома Пионеров и т.п. организации, для них тариф - 70 р.

Оплату для организаций и индивидуальных заказчиков производить по адресу:

Фрунзенское отделение Жилсоцбанка г. Москвы МФО 201412 Р/С 500461778, кооператив "ИНФОРКОМ".

Игровое приложение

Февраль 1991

ELITE



Сегодня мы большую часть игрового приложения посвятим самой известной компьютерной программе "ELITE" фирмы FIREBIRD.

Успех, выпавший на долю этой программы вполне заслужен. Несколько лет зарубежные Синклер-журналы были заполнены статьями, так или иначе связанными с "Элитой". "Программой жизни" называют ее на Западе, имея по-видимому в виду трудности выживания в космическом мире. Вы сами кузнец своего счастья, и от той стратегии, с которой Вы вступаете в космическое сообщество, будет зависеть Ваш успех, или наоборот разорение и физическая гибель.

Вы можете быть космическим рейнджером, кровожадным пиратом, мирным торговцем, разработчиком полезных ископаемых, а можете просто чувствовать себя туристом, путешествуя по неизведанным мирам и во всех случаях игра будет проходить для Вас по-разному.

Фирменный проспект с описанием программы занимает примерно 60 страниц, а к программе прилагается еще и небольшой фантастический роман, в котором описаны межгалактические приключения главного героя. Перед нами стояла достаточно сложная задача вместить весь этот поток информации в наши ограниченные рамки.

Надо сказать, что фирма, выпустившая программу, сделала не совсем обычный ход. В программе есть события, даже не упомянутые в обширном описании. Успех был потрясающий. В течение нескольких лет тысячи любителей этой игры писали статьи и заметки о том, что им удалось обнаружить, возникло даже целое направление исследования, связанное с изучением этой игры. Не открывая всех вскрытых нюансов этой игры, мы скажем только что где-то по ходу игры Вам могут быть предложены дополнительные секретные миссии, если Вы будете того достойны.

1. ВСТУПЛЕНИЕ.

Итак, Вы начинаете свою боевую и деловую карьеру. Для начала Вы располагаете кораблем "Кобра-МК-3" и суммой в 100 кредитов (CR). Корабль оснащен топливом, достаточным для гиперперехода на 7 световых лет и вооружен тремя самонаводящимися ракетами, а также пульсирующим лазером на носовой надстройке. В исходном состоянии Вы пришвартованы к орбитальной станции около планеты LAVE.

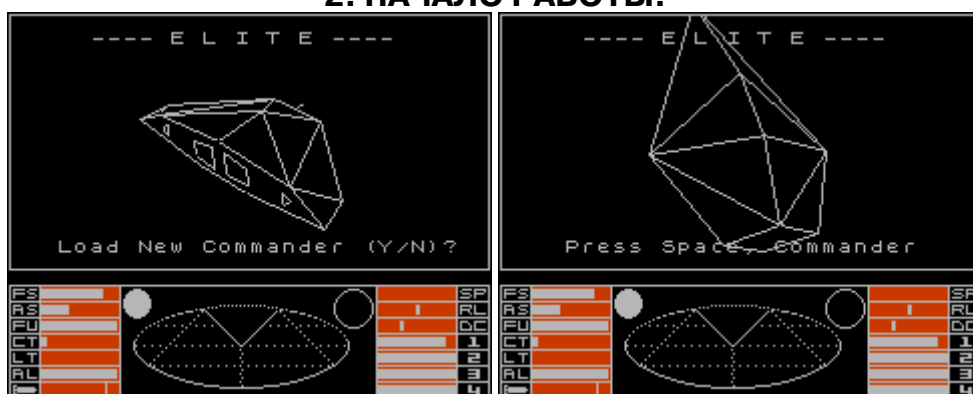
"Кобра-МК-3" - наилучший из транспортно-боевых кораблей среднего радиуса действия, средней грузоподъемности. Это идеальный корабль для начинающих космических бизнесменов и бойцов.

Корабль высокоманевренный, хорошо держит курс на гиперпереходах. Он имеет систему защитных полей, питающихся от 4-х энергетических блоков.

Пилотские лицензии выдаются только на планете LAVE, поэтому это единственное место, где разрешено тренироваться в стыковке и расстыковке с орбитальной станцией. Воспользуйтесь этой возможностью.

Грузовой отсек корабля может содержать 20 однотонных контейнеров. Этот объем можно увеличить за счет расширения грузового отсека, что не влияет на маневренность корабля. Контейнеры (ТС) прикреплены магнитным путем к грузовым кронштейнам.

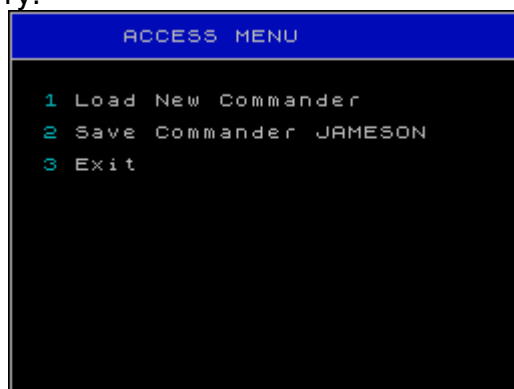
2. НАЧАЛО РАБОТЫ.



После завершения загрузки, Вы увидите вращающийся космический корабль. Это "Кобра МК-3", идентичный вашему. Не обращайтесь пока внимания на подпись "LOAD NEW COMMANDER (Y/N)?", а нажмите клавишу SPACE для перехода на вторую титульную страницу. С этого момента начинается игра, на экране появляется информация, описывающая текущую игровую ситуацию.

3. РАБОТА С КАССЕТОЙ.

Пока Вы находитесь в пристыкованном к космической станции состоянии, Вы можете записать свое положение на ленту.



Нажмите SYMBOL SHIFT, появится меню.

1. Загрузить отложенную игру.
2. Записать игру на ленту. Имя.
3. Выход.

Это же меню появилось бы, если бы Вы нажали клавишу "Y" в ответ на вопрос "LOAD NEW COMMANDER (Y/N)?"

Загрузка отложенной игры:

От Вас запросят имя. После загрузки Вы сразу попадете на вторую страницу.

Запись на ленту:

От Вас запросят имя. После выгрузки Вы попадете на страницу, изображающую текущее состояние игры.

4. СТРАНИЦА СОСТОЯНИЯ ИГРЫ.

На этой странице сверху изображается имя командира. Далее идет следующая информация:

PRESENT SYSTEM - планетная система, в которой Вы находитесь.

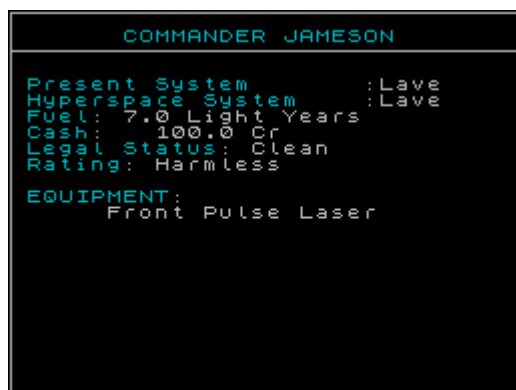
HYPERSPACE SYSTEM - указывает, на какую планетную систему в данный момент настроена система гиперперехода.

FUEL: 3.5 LIGHT YEARS - топливо: 3.5 световых года.

CASH: 100 CR - наличные средства: 100 кредитов.

LEGAL STATUS: - Ваш правовой статус. В нормальном состоянии он CLEAR, т.е. у галактической полиции к Вам претензий нет. Но по мере совершения правонарушений, он может меняться до OFFENDER и даже до FUGITIVE. В последнем случае Вы можете быть атакованы полицейскими кораблями.

RATING: - Здесь изображается Ваш боевой рейтинг. HARMLESS - низшая ступень, обозначает вашу полную безвредность для кого-либо. По мере накопления боевого опыта Ваш рейтинг будет меняться.



5. БОЕВОЙ РЕЙТИНГ.

Для того, чтобы стать бойцом класса "Элита", Вы должны проявить высокое мастерство и огромное терпение, т.к. только с накоплением материальных средств в результате квалифицированных торговых операций Вы сможете приобрести дополнительное вооружение и оборудование: лучевой лазер, дополнительные ракеты, энергетические бомбы, галактический гипер-привод, стыковочный компьютер и пр. Вы будете жить со своим боевым рейтингом. Начав как HARMLESS Вы перейдете к MOSTLY HARMLESS. Далее, по мере накопления боевого опыта, Вы станете POOR, затем AVERAGE, затем ABOVE AVERAGE, COMPETENT, DANGEROUS и, наконец, DEADLY. И лишь очень немногие станут ELITE.

Ваши победы будут автоматически фиксироваться и передаваться в Галактический Федеральный Центр, пропорционально будет меняться и рейтинг.

Управляйте кораблем аккуратно, действуйте мудро. Помните, что другие пилоты могут попытаться увеличить свой рейтинг нападением на торговые корабли или на полицейские корабли класса VIPERS. Если Вы примете такую тактику (или если начнете торговлю запрещенным товаром), то Ваш боевой рейтинг может расти, но при этом будет и изменяться и правовой статус: CLEAN - OFFENDER - FUGITIVE

Вы можете стать врагом номер один и не сомневайтесь, Вас не оставят в покое.

6. НАВИГАЦИЯ.

В Галактическую Кооперацию входят более 2000 обитаемых планет, разбросанных по 8-ми галактикам. Политическая власть на планете - это важный фактор, который должен учитываться при навигации. Поскольку многие планеты находятся в состоянии анархии и посещать их на плохо вооруженной корабле очень опасно. Важным также является экономический профиль планеты.

Ваша стратегия будет зависеть от ваших целей и задач, от того, насколько Вы готовы рисковать. Чем больше Вы рискуете, путешествуя к опасным планетам или торгуя запрещенными товарами, тем быстрее Вы заработаете средства и оснастите свой корабль, но тем быстрее Вы и погибнете. Вскоре Вы поймете, что жизнь в межгалактическом пространстве - это вопрос баланса и путь пиратства, ведущий к быстрому повышению рейтинга быстро приводит и к финалу.

В этой игре от Вас, конечно потребуются способности бойца, но это не основное. Умение размышлять и принимать правильные решения здесь важны не менее.

7. УПРАВЛЕНИЕ НАВИГАЦИЕЙ.

Клавиша I - карта Галактики.

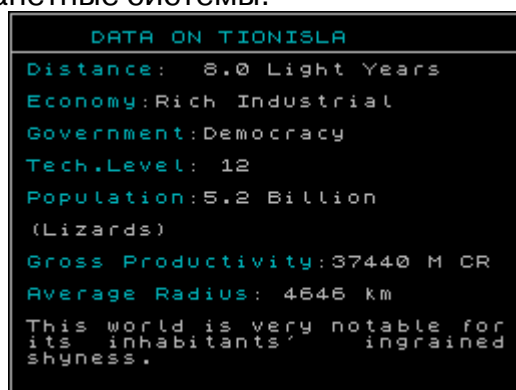
При нажатии этой клавиши на экране появляются все обитаемые планетные системы

Галактики. Здесь же обозначены Ваши координаты.



Управляя курсором, Вы можете просмотреть данные на эти системы в поисках выгодной торговой операции. Управление курсором: S-вверх, X-вниз, N-влево, M-вправо.

Клавиша P - данные на планетные системы.



Нажатием этой клавиши вызывается информация по какой-либо планете. Имейте в виду, что Космическая Администрация не несет ответственность за точность предлагаемой информации, но опытный бизнесмен сам разрабатывает для себя критерии возможности использования информации.

Пример информации.

DISTANCE: 3.6 LIGHT YEARS - расстояние до системы 3.6 световых года.

ECONOMY: AVERAGE AGRICULTURAL - экономика: среднеразвитая, аграрная.

GOVERNMENT: DEMOCRACY - тип правительства: демократия.

TECH. LEVEL: 8 - технологический уровень: 8.

POPULATION: 4.1 BILLION - население: 4,1 млрд.

GROSS PRODUCTIVITY: 13120 MCR - совокупный общественный продукт: 13120 млн. кредов.

AVERAGE RADIUS: 6155 км - средний радиус.

Ниже приводится краткий комментарий по основным особенностям планеты.

Теперь вновь нажмите клавишу I и вызовите карту Галактики.

Малый курсор (крестик) указывает на ту планету, на которую Вы настраиваетесь.

Большой курсор указывает ваше местоположение. Окружность вокруг него показывает, куда Вы можете совершить гиперпереход с имеющимся у Вас запасом топлива.

Клавиша B - возврат малого курсора к большому.



Клавиша O - вызов локальной карты. Это подробная карта, показывающая

расположение планетных систем, ближайших к месту Вашей стыковки. Поскольку "Кобра МК-3" имеет топлива для гиперперехода не более, чем на 7 световых лет, цель перелета надо выбирать очень тщательно, проверяйте себя по предлагаемой информации на звездные системы.

Если курсор выйдет за пределы экрана, возвращайте его клавишей "В".

Клавиша D - эта клавиша показывает насколько далеко находится система, ближайшая к малому курсору.

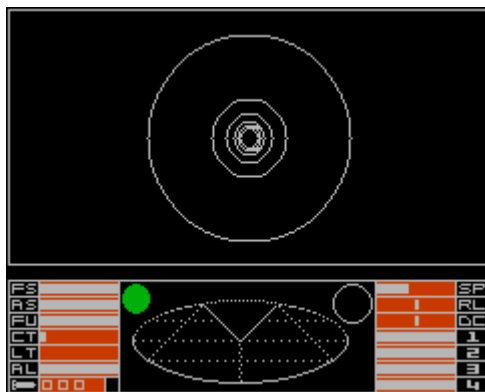
8. ОРБИТАЛЬНАЯ СТАНЦИЯ "КОРИОЛИС".

Каждая из обитаемых планет имеет несколько космических станции "Кориолис" на орбите. Эти станции являются "нейтральной" территорией, которая управляется одновременно Галактическим Центром Развития и Администрацией планеты. В последние годы появились станции в виде додекаэдра, их поэтому называют "Додо".

Все орбитальные станции имеют мощные защитные поля от атак пиратов и неквалифицированной стыковки, а также флот полицейских кораблей "VIPER". Дозаправка кораблей, а также разгрузка-погрузка производятся на этих станциях. Каждая станция имеет диаметр 1 км и может принимать одновременно до 2000 кораблей.

9. ОТХОД ОТ ОРБИТАЛЬНОЙ СТАНЦИИ.

Он выполняется клавишей 1. После этого в течение 10 сек. корабль проходит туннель в защитных полях станции.



"Кобра МК-3" - высокоманевренный корабль, но он требует наличия определенной практики.

N - вращение корабля против часовой стрелки.

M - вращение корабля по часовой стрелке.

X - вверх.

S - вниз.

SPACE - увеличить скорость.

SYMBOL SHIFT - уменьшить скорость.

При выполнении маневров пользуйтесь компасом, который имеется на приборном дисплее. Это окружность в правом верхнем углу дисплея: внутри имеется светлая точка. Она указывает положение Вашего корабля. Когда точка четкая и находится в центре окружности, космическая станция перед вами.

Потренируйтесь в управлении кораблем.

Вам будут встречаться здесь другие корабли. Никто не нападет здесь на Вас, так как Вы находитесь под защитой станции. Если же Вы начнете здесь стрельбу, то будете отвечать в полной мере.

10. ПРИБОРНАЯ ПАНЕЛЬ.

В центре панели имеется дисплей в виде эллипса. Он показывает трехмерное изображение того, что находится в непосредственной близости от вашего корабля. Поле зрения сделано так, как будто наблюдатель находится сзади и сверху от Вашего корабля.

Слева и справа от эллипса имеются две окружности. Та, что справа - это компас, о нем см. выше. Окружность слева имеет цвет. Зеленый обозначает, что никакой опасности для Вас нет. Желтый указывает на возможную, а красный на реальную опасность. Мигающий

красный означает критическое состояние. Это может быть, если к Вашему кораблю приблизится вражеский, если Вы очень близко подлетите к Солнцу и т.п.

Слева и справа от радара имеются шкалы различных приборов.

FS и AS - состояние энергетических источников защитных полей.

FS - переднего поля.

AS - заднего поля.

FU - уровень топлива.

CT - температура, она возрастает при приближении к звезде.

LT - температура лазера. Она повышается во время стрельбы. При перегреве лазер сам отключается, чтобы не вывести из строя систему.

AL - высотомер. Летать слишком низко над планетой опасно.

SP - скорость. При подходе к планете ее надо держать максимальной. При маневрах у орбитальной станции - низкой. В последний момент причаливания - минимальной.

RL - скорость поворота вправо / влево.

DL - скорость поворота вниз/вверх.

1,2,3,4 - состояние энергетических отсеков. Уровень снижается тогда, когда защитные поля подавляют огонь противника, когда интенсивно работает лазер или противоракетная система ECM. При опустошении четвертого отсека появляется надпись "ENERGY LOW".

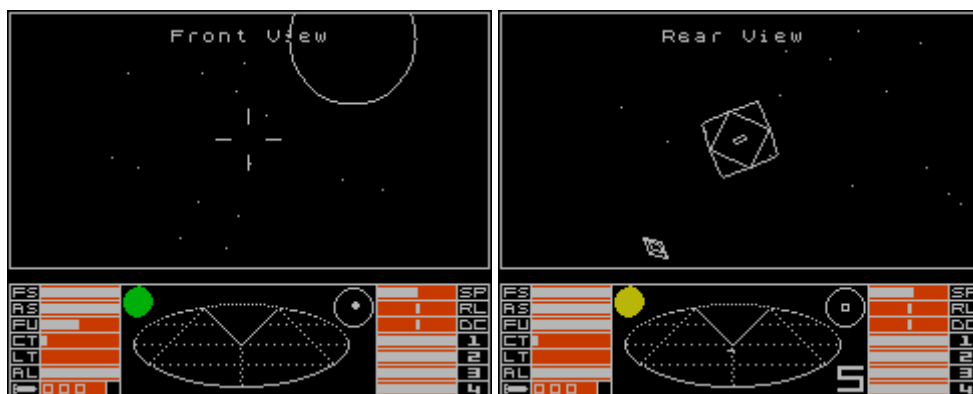
В нормальных условиях отсеки заполняются постепенно сами за счет аккумуляции поверхностного излучения.

В нижней части левой колонки изображен символ ракеты. Эта шкала указывает количество ракет на борту и состояние в котором они находятся.

"Зеленый" - ракеты нацелены; "Черный" - ненацелены; "Красный" - сняты с предохранителя и готовы к стрельбе.

11. ПЕРЕЛЕТ.

Отчалив от орбитальной станции Вы оказались на низкой орбите возле планеты LAVE. Уменьшите Вашу скорость до абсолютного минимума, прежде чем вводить координаты гиперперехода.

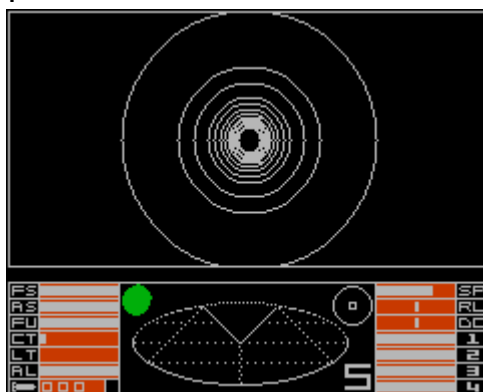


Во время полета клавиши 1,2,3,4 дают вид пространства вблизи корабля.

1 - впереди; 2 - сзади; 3 - слева 4 - справа.

Клавиша H - гиперпереход.

Нажмите "I" или "O" и вызовите карту Галактики. Подведите курсор к требуемой планете и нажмите "I", а затем "H".



После гиперперехода Вы окажетесь на некотором расстоянии от цели. Мы Вам рекомендуем добраться до орбиты как можно быстрее. До предела увеличьте скорость, можете воспользоваться J-двигателем. Для этого нажмите клавишу J. Но имейте в виду, что система J-транспортировки не работает из-за интерференции, если рядом с Вами есть звезда, планета или другой корабль. В этом случае Вы услышите звуковой сигнал.



Вскоре Ваш бортовой компьютер примет позывные радиомаяка орбитальной станции. Компас перенастроится с планеты на станцию. На дисплее появится буква S, означающая что корабль находится под защитой станции.

12. МЕЖГАЛАКТИЧЕСКИЙ ПЕРЕЛЕТ.

Клавиши "G", затем "H".

Межгалактический гиперпереход - довольно энергоемкое мероприятие и его можно совершить только с планет, технологический уровень которых не ниже 10. Его можно выполнить только один раз. После перелета Вы находитесь в новой Галактике, соседней с данной.

Всего Галактик 8, поэтому чтобы вернуться в Галактику 1, Вам надо выполнить 8 межгалактических переходов.

13. ТЕХНИКА СТЫКОВКИ С ОРБИТАЛЬНОЙ СТАНЦИЕЙ.

Стыковка - очень сложный процесс, если Ваш корабль не оснащен специальным стыковочным компьютером (включается клавишей C).



Мы рекомендуем следующую последовательность действий.

Подлетите к станции. Входной туннель всегда обращен лицом к поверхности планеты. Пролетите рядом со станцией и двигайтесь к планете. Следите при этом за высотой. Теперь развернитесь, Вы будете смотреть в сторону входного туннеля. Стыкуйтесь на абсолют¹ совпадения скорости вращения корабля и станции вокруг своей оси. Если Вы заденете стенки туннеля, это может быть смертельным, но может только повредить корабль и груз. При очень низкой скорости даже если Вы и заденете стенки, то все может кончиться благополучно, хотя с потерей энергии защитных полей.

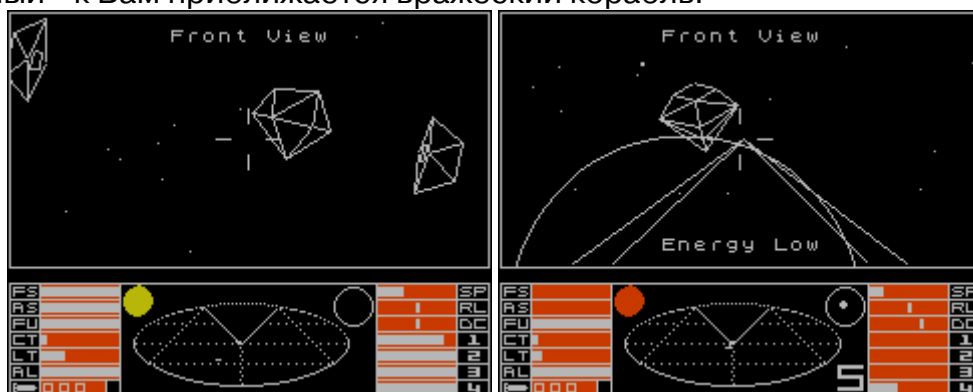
14. БОЕВЫЕ ДЕЙСТВИЯ.

Не все корабли в открытом космосе пираты, но если Вы уничтожите пиратский корабль или корабль пришельцев с Таргона, Вы получите приличную премию.

¹ В оригинале пропущена строка (прим. OCR).

"Кобра МК-3" - отличный высокоманевренный боевой корабль. У него довольно небольшой радиус разворота, хотя с увеличением скорости он тоже увеличивается. Корабль может за счет скорости и маневренности уйти от большинства противников, но он не уйдет от ракеты.

Одним из признаков опасности является то, что Вы не можете совершить J-переход, значит рядом кто-то есть. Другой признак - показание индикатора на дисплее. Если он желтый или красный - к Вам приближается вражеский корабль.



Мигающие штрихи на каком-либо экране означают, что по Вам открыли огонь из лазеров. Защитные поля отражают этот огонь с легким скрежешущим звуком. Если звук стал низким, осторожно - огонь пробил защитное силовое поле.

О приближении ракеты свидетельствуют сообщения на экране бортового компьютера.

Многие космические пираты не отличаются храбростью и скрываются встретив отпор. Некоторые же не знают страха и стоят до конца, в частности корабли с Таргона.

15. ВОЗМОЖНЫЕ ПРОТИВНИКИ.

Охотники за призами.

Если Вы не получили правовой статус FUGITIVE, то они Вас не тронут. Их не интересуют такие незначительные корабли, как Ваш. Они пользуются чрезвычайно эффективным кораблем "Фер де Лэнс". Их можно найти за пределами орбитальных станций, особенно в районе "опасных" планет, где они охотятся за пиратами. Их боевой рейтинг обязательно "DEADLY" или "ELITE". Они редко объявляют себя пролетающим мимо кораблям и при слишком близком приближения обычно убивают.

Пираты.

Пираты могут пользоваться кораблями самых разных типов, поэтому визуально их идентифицировать довольно трудно. Они могут быть где угодно в космосе, но тяготеют к планетам с нестабильной политической структурой, в частности к анархическим и феодальным.

Как их распознать?

Маленькие корабли, оказавшиеся вдали от планетных систем - это обычно пираты.

Большие корабли, окруженные ворохами более мелких - это тоже как правило пираты.

Корабли, отказывающиеся отвечать на запрос - пираты.

В районе планет, управляемых автократией или кланами, пиратам иногда платят за то, чтобы они не трогали пролетающие торговые корабли. Пиратство - это могучий и очень сложный бизнес. Начинающий пилот вскоре сумеет распознавать пиратские корабли по косвенным данным, если хочет выжить.

Полиция.

Полиция обычно использует маленький корабль "Вайпер" очень быстрый и маневренный. Эти корабли несут постоянное дежурство на станциях "Кориолис" и "Додо" и готовы атаковать пирата в течение секунд. Они не арестовывают - они убивают. Полиция имеет много подразделений. Одни занимаются пиратами, другие - наркотиками, третьи - работорговлей, но все они оснащены этими же кораблями.

Корабли Таргона и "Таргоны".

Таргондяне - злейшие враги человечества. В восьми галактиках насчитывается по

крайней мере 50 зон, где непрерывно с переменным успехом ведутся боевые действия. Таргондяне - высокоразвитая цивилизация. Это насекомовидная раса. Они ведут войну в настоящее время и с 17 различными другими формами разумной жизни.

Все пилоты Таргона бесстрашны и безжалостны в бою. По классу большинство из них соответствует разряду "ELITE".

Очень часто небольшие корабли совершают диверсионные рейды. Это чрезвычайно быстрые корабли, которые обязательно имеют противоракетную систему ECM. Кроме того, большинство их боевых кораблей несет на себе несколько маленьких дистанционно управляемых боевых катеров, снабженных лазером. Это так называемые "Таргоны".

Галактические Боевые Силы ведут разработку своего дистанционно управляемого корабля и готовы платить большие деньги за доставленный к ним "Таргон".

Ходят слухи, что таргондяне научились "парить" в гиперпространстве и могут там перехватывать проходящие корабли.

16. Другие космические объекты.

Отшельники.

Состарившиеся пираты и изгнанники с планет очень часто поселяются на одиноких астероидах. Обычно они выбирают астероид покрупнее и устанавливают маяк, предупреждающий разведчиков полезных ископаемых.

Они пользуются защитой всегалактического права, а так как полезные ископаемые с большинства астероидов уже выбраны, то совсем никому не интересны.

Корабли поколений.

До того как были изобретены гипердвигатели человечество отправлялось к звездам на крупных кораблях, где сменялось не одно поколение во время перелета. И сейчас в космосе летают до 70 тыс. таких кораблей. На многих сменилось уже 30 поколений. Наказание за нападение на такой корабль - всеобщее презрение и полное изгнание.

Космические платформы.

Эти корабли можно найти в тех районах, где была война или естественная катастрофа. Это гигантские корабли-заводы. Они имеют десятки миль в длину. На них живут люди, уцелевшие от катастрофы. Они имеют тяжелое вооружение и флоты истребителей. Избегайте их любой ценой.

17. СНАРЯЖЕНИЕ И ВООРУЖЕНИЕ.

Его можно приобретать только на планетах с соответствующим технологическим уровнем

Наименование.	Технол. уровень	Цена
FUEL (топливо)	любой	перемен.
MISSILE (ракета)	любой	30
LARGE CARGO BAY (груз. отсек)	любой	400
ECM SYSTEM(противоракет. система)	2	600
PULSE LASER (пульсир. лазер)	3	400
BEAM LASER (лучевой лазер)	4	1000
FUEL SCOOPS (заборники топлива)	5	525
ESCAPE CAPSULE (спасат. капсула)	6	1000
ENERGY BOMB (энергетич. бомба)	7	900
EXTRA ENERGY UNIT (доп. энергет. блок)	8	1500
DOCKING COMPUTER (стыковочн. комп.)	9	1500
GALACTIC HYPERDRIVE (галактич. гиперпривод)	10	5000
MINING LASER (технолог. лазер)	10	800
MILITARY LASER (военный лазер)	10	6000

EQUIP SHIP	
1 Fuel	4000.00
2 Missile	4000.00
3 Large Cargo Bay	4000.00
4 E.C.M. System	4000.00
5 Extra Pulse Lasers	1000.00
6 Extra Beam Lasers	1000.00
7 Fuel Scoops	1000.00
8 Escape Pod	1000.00

Item?

Лазеры.

Вы начинаете игру с одним пульсирующим лазером, установленным впереди. Прицелов в боковых и в заднем экране таким образом пока нет. Когда Вы наберете достаточно средств, можете установить лазеры и там. Со временем сможете и заменить свои пульсирующие лазеры на более мощные лучевые.

Стрельба из лазера выполняется клавишей А. При покупке лучевого лазера взамен пульсирующего, стоимость последнего учитывается. Военные лазеры - чрезвычайно эффективное оружие.

Ракеты.

Их может быть не более четырех. Это тоже очень эффективное оружие. Когда ракета идет на Вас, бортовой компьютер предупреждает Вас об этом. Если у Вас нет системы ЕСМ, то уйти от ракеты можно только маневрированием. Попадание ракеты почти полностью уничтожает энергию силовых защитных полей.

Ракета может быть запущена по цели из любого экрана - 1,2,3,4. Нацеливание выполняется клавишей Т. Один из индикаторов ракет загорается зеленым цветом. Захват и сопровождение цели выполняется автоматически, когда цель оказывается около перекрестия прицела.

Индикатор становится красным.

Пуск - клавишей F. Снять ракету с прицеливания до пуска можно клавишей U.

Энергетические бомбы.

Такая бомба - оружие однократного применения. Включается клавишей U. Она уничтожает все объекты (корабли, ракеты, астероиды) в пределах видимости.

18. Защита.

Самое лучшее средство защиты - это находиться под защитой орбитальной станции.

Защитные поля и энергетические блоки. В исходном положении у Вас есть носовое и кормовое защитные поля. После того, как поле пробито вражеским огнем, начинается расход энергии из блоков. Далее вражеский огонь может повредить груз, вооружение, а затем и уничтожить Ваш корабль.

Применение лазера или системы ЕСМ также снижает ваши запасы энергии. Бортовой компьютер держит Вас в курсе состояния ваших запасов энергии.

Вы можете купить более современный дополнительный энергетический блок, который удвоит темп подзарядки.

Система ЕСМ - клавиша E.

Это противоракетная система. Ее можно использовать неоднократно. При нажатии клавиши E уничтожаются все ракеты в пределах видимости (в том числе и Ваши). Корабли противника тоже могут иметь эту систему. Бортовой компьютер информирует о ее наличии изображением большой буквы E.

Спасательная капсула - клавиша Q. При нажатии Q во время боя Вы будете спасены и доставлены на ближайшую орбитальную станцию. Ваш счет в банке останется, но груз будет потерян.

К капсуле прилагается страховой полис, по которому Вам выдадут новый корабль с

аналогичным оснащением. Так как регистрация в полиции привязана к Вашему кораблю и есть возможность очистить таким образом и свой правовой статус. Это недоработка полиции, но Вам она может пригодиться.

19. СНАРЯЖЕНИЕ.

Топливо. Оно достижимо на всех планетах. Заправляться можно только до полного объема (не меньше).

Заборники топлива (они же и грузозаборники). Они позволят Вам восполнять запасы топлива без дозаправки в результате пролета на большой скорости вблизи звезды.

С помощью этого устройства можно принимать на борт грузовые контейнеры и топливо с побежденных кораблей. Для этого держите их в нижней части экрана и двигайтесь по направлению к ним.

Дополнительный грузовой отсек. Он увеличивает Вашу грузоподъемность от 20 до 35 тонн.

Стыковочный компьютер. Позволяет стыковаться с орбитальной станцией автоматически. Включается клавишей С.

Межгалактический гиперпривод. Включается клавишами G, затем H. Одноразового применения.

Технологический лазер. Применяется для разрушения астероидов, состоящих из полезных ископаемых. Обломки собираются с помощью грузозаборников, которые предварительно должны быть установлены.

20. ТОРГОВЛЯ.

Среди предметов торговли есть запрещенные, которыми Вам лучше не заниматься. В таблице они отмечены знаком "*".

Вам может показаться странным, что работорговля измеряется в тоннах, но сюда входит система криоподвески, необходимая, чтобы доставить товар живым.

Товар	Средняя цена
FOOD (пищевые продукты)	4.4/т
TEXTILES (текстиль)	6.4/т
RADIOACTIVES (радиоакт. руды)	21.2/т
*SLAVES (рабы)	8.0/т
LIQUOR/WINES (спиртное)	25.2/т
LUXURIES (предметы роскоши - парфюмерия, кофе и др.)	91.2/т
*NARCOTICS (наркотики, табак)	114.8/т
COMPUTERS (компьютеры)	84.0/т
MACHINERY (заводские и сельские машины)	56.4/т
ALLOYS (промышл. металлы)	32.8/т
*FIREARMS (ручное оружие, легкая артиллерия)	70.4/т
FURS (меха)	56.0/т
MINERALS (неочищенная порода со следами редких элементов)	8.0/т
GOLD (ЗОЛОТО)	37.2/кг
PLATINUM (платина)	65.2/кг
GEM-STONES (ювелир. камни)	16.4/г
ALIEN ITEMS (предметы вне земных цивилизаций - артефакты и др.)	27.0/т

Таблица товаров, имеющих в продаже на планете, вызывается клавишей К. Указываются: наименование товара, единица измерения, цена, количество, предлагаемое к продаже.

LAVE MARKET PRICES			
PRODUCT	UNIT	UNIT PRICE	QUANTITY FOR SALE
Food	t	3.00	17t
Textiles	t	10.00	18t
Radioactives	t	100.00	26t
Slaves	t	100.00	14t
Liquor/Wines	t	20.00	39t
Luxuries	t	44.44	8t
Narcotics	t	100.00	-
Computers	t	100.00	-
Machinery	t	100.00	10t
Alloys	t	33.33	25t
Firearms	t	50.00	-
Furs	t	100.00	-
Minerals	t	10.44	61t
Gold	kg	64.44	14kg
Platinum	kg	64.00	17kg
Gem-Stones	g	15.00	12g
Alien Items	t	51.00	-

Если Вы не хотите ничего покупать, то нажмите ENTER. Вы окажетесь в таблице INVENTORY, которая показывает количество имеющегося у Вас груза, топлива и наличных средств.

Если хотите что-либо купить, нажмите "2", появится таблица и от Вас будут запрашивать каждый товар по очереди. Если Вы его брать не хотите, нажимайте ENTER, если хотите, наберите необходимое количество.

Клавиша "3" вводит Вас в режим продажи товара.

21. ПОЛЕЗНЫЕ СОВЕТЫ ПО ТОРГОВЛЕ.

Ваш корабль может иметь 4 лазера, 4 ракеты и одну энергетическую бомбу. С полным вооружением Вы можете торговать даже в местах наводненных пиратами. Но мы настойчиво рекомендуем не делать этого, если Ваш рейтинг ниже "DEADLY". Не приближайтесь к планетам, отмеченным в информатории как анархические или феодальные - "ANARCHY" или "FEUDAL".

Выбирайте вариант для торговли, думайте!

1. Что может быть нужным планете?
2. В чем состоит ее социальная функция?
3. Не ввозите предметы роскоши в голодные миры.
4. Если доход того не стоит, везите свой товар в другое место.

22. ДРУГИЕ СПОСОБЫ ОБОГАЩЕНИЯ.

Охота за призами.

Галактические страховые компании выплачивают большие премии за уничтожение пиратских кораблей. Обычно для этой цели надо перелететь к месту возможного пребывания пиратов и ждать нападения.

Пиратство.

Оно широко распространено во всех восьми галактиках. Многие из пиратов не закоренелые преступники, а просто разорившиеся бизнесмены. Чтобы выжить в этих условиях нужен немалый боевой опыт.

Если пиратский корабль имеет FUEL SCOOPS (грузозаборники), он может собирать грузовые контейнеры с разбитых кораблей и торговать ими.

Разработка астероидов.

В них немало средств, но чтобы их разрабатывать, надо иметь технологический лазер и грузозаборники для сбора обломков. Имейте в виду, что поскольку астероиды мешают навигации, то за его уничтожение Вам будет выплачена наибольшая премия.

Поиск грузовых контейнеров.

В результате боевых действий или естественных катастроф в космосе остаются несобранными контейнеры с товарами. Вы можете их бесплатно собирать для себя, если Вы установили грузозаборники. Но смотрите, есть вероятность, что по ошибке Вы подберете контейнер с запрещенным товаром.

ADVENTURE LESSONS

Сегодня мы начинаем серию уроков, посвященных работе с игровыми программами жанра ADVENTURE. Причины, которые позволяют нам особо выделить программы этого жанра из обширного многообразия игровых программ были вкратце затронуты в статье "Жанры игровых программ", опубликованной в прошлом выпуске. Напомним основные преимущества адвентюрных программ:

1. Эти программы являются удобным средством для совершенствования и приобретения навыков в английском языке.

2. Работа с этими программами проходит в диалоговом режиме и способность установить контакт с программой, вести с ней диалог может Вам еще многократно в жизни пригодиться, причем при работе с совсем даже не игровыми программами.

3. Работа с такими программами требует от пользователя гораздо более интеллектуального подхода, чем элементарные "стрелялки". Они развивают словарь, учат гибкому мышлению, требуют логического подхода, умения анализировать ситуацию и делать правильные выводы. Аналитичность - главная особенность этих программ.

4. От хорошо поставленной текстовой диалоговой программы до обучающей системы всего лишь шаг. А оттуда рукой подать и до экспертных систем.

5. В этих программах большую роль играет общекультурный аспект. Очень часто сюжет программы базируется на известном литературном произведении, на старинных балладах и легендах.

6. И, наконец, надо честно сказать, что в этом жанре программы написанные для "Спектрума" достигли наивысших успехов. Понятно, что имитаторы автомобилей и самолетов на IBM всегда будут более совершенными, чем на "Спектруме", аркадные игры с высокоразвитой графикой тоже.

Но с ADVENTURE GAMES ситуация совсем иная. Что могут предложить суперкомпьютеры этому жанру? - Графику? Музыку? Невероятный объем? - А зачем?

Весь интерес состоит в диалоге, в логике игры и графика и музыка здесь часто просто досадная помеха, а невероятные объемы тоже не нужны, ведь и так, чтобы пройти скромную программу фирмы LEVEL 9 нужны месяцы.

Одним словом, можно предполагать, что с дальнейшим развитием технических средств все виды игровых программ будут развиваться и становиться все более и более изощренными, но текстовые игры для "Спектрума" уже достигли такого высокого уровня, что грех был бы не использовать это обстоятельство.

Работа с текстовыми играми может быть условно разбита на несколько элементов, на которых мы ее и освоим.

1. Общий анализ.
2. Перемещение в игровом пространстве.
3. Осмотр и анализ обстановки.
4. Работа с объектами.
5. Взаимодействие с другими персонажами.
6. Использование отгрузки программы на ленту.
7. Полезные практические приемы.

Урок 1. Общий анализ

Итак, Вы загрузили некоторую текстовую программу. Обычно цель, которая стоит перед Вами записана в нескольких строчках в самом начале. Может быть Вам надо спастись с необитаемого острова, может быть починить свой космический корабль, а может быть, если Вы заколдованы в лягушонка, то вернуть себе человеческий облик путем серии превращений в других животных. На худой конец, если цель неизвестна, все равно можете приступить к исследованию, как если Вы оказались на совершенно чуждой Вам планете, а жить то как-то все равно надо.

Во-первых, надо установить диалог с компьютером. Диалог происходит путем подачи команд и получения ответа. Ничего, если поначалу самым типичным ответом будет ответ

типа "Не понимаю", "Не знаю", "Пардон?.." и т.п.

Основой команды, которую Вы можете дать компьютеру почти всегда является глагол, а их-то как оказывается не так уж и много в компьютерном языке. Глагол может сопровождаться существительным, а может и нет, это уже по смыслу. Если в начале игры Вам сообщили, что Вы сидите в кресле, то первое что Вы можете сделать - это посмотреть по сторонам - LOOK. Следующим Вашим шагом может быть попытка встать - STAND, после чего невинно снова сделать LOOK и посмотреть что изменилось.

Эти команды состояли лишь из глагола, но если Вы например хотите что-то взять (TAKE), то придется добавлять существительное, например взять карандаш TAKE PENCIL.

Заметьте, что от Вас совсем не требуется быть семи пядей во лбу, соображая что же именно Вам лучше взять и как оно по-английски запишется, ведь в ответ на команду LOOK компьютер сам Вам перечислил, что Вы перед собой видите. Это может быть и карандаш, а может быть и пальма. Ее конечно взять нельзя, но можно попробовать на нее влезть (CLIMB), чтобы оттуда сделать LOOK.

Сравнительно редко в команду может входить и прилагательное, тогда команда будет записана тремя словами. Такое происходит, когда надо более детально охарактеризовать предмет.

TAKE RED PENCIL - взять красный карандаш.

TAKE GOLD KEY - взять золотой ключ.

Итак, ваша первая задача - установить те команды, которые понимает программа, а поскольку основой команды является глагол, то значит надо установить известные ей глаголы.

Как правило программа по-разному реагирует на команды, которые ей непонятны (нет их в ее словаре) и те, которые понятны, но она не может их выполнить в силу каких-то причин. Например Вы видите перед собой пальму и пытаетесь влезть на нее, а программа упорно говорит "Can't" (НЕ МОГУ). А потом оказывается, что пальма-то была нарисованной.

Попробуйте дать ей команду ABCDEFGH и посмотрите, что она ответит. Например она ответит What? Для Вас это означает, что на незнакомый глагол она отвечает именно так. Теперь если в игре Вы встретите от нее вопрос What? то будете знать, что Вы употребили слово, которое ей неизвестно.

Можно прозондировать и не только глаголы, но и существительные. Например дайте машине нелепую команду HAMMER (молоток) и посмотрите, что она ответит. Если ответит What? - значит такого слова вообще не знает и будьте спокойны, молотков в игре Вы не встретите, а если ответит, к примеру Can't - значит команду не поняла, но само такое слово ей известно и где-то оно Вам пригодится.

Кстати, обратите внимание, что далеко не все то, что программа Вам сообщает в ответ на команду LOOK ей известно и входит в ее словарь. Начинающих часто шокирует, что только что программа написала, что Вы видите окно, а на просьбу его открыть отвечает, что слово окно ей вовсе неизвестно. Такое бывает, описания обстановки хранятся отдельно от действующего словаря и совсем не все то, что попадает в обстановку может иметь участие в игре.

Так надо аккуратно прощупать словарь, который программа понимает. Здесь же неплохо и определиться насколько можно сокращать слова. Дело в том, что большинство текстовых программ понимают слова по неполной записи. Иногда по трем буквам (со слабым словарем), но чаще - по четырем.

Тогда вместо длинной записи BREAK MIRROR - разбить зеркало, можно будет писать проще: BREA MIRR. Правда обратите внимание на то, что уж если пишете слово, то либо сокращенно, либо нет. Записать сокращение до 5-ти букв MIRRO - нарваться на непонимание.

Желательно также сразу проверить и действие еще нескольких полезных команд, таких как INVENTORY, HELP, SAVE, LISTEN.

INVENTORY - инвентарь. По этой команде программа распечатывает список тех предметов, которые Вы при себе имеете. Очень часто она открывает Вам нечто, о чем Вы и не догадывались или, наоборот, оказывается, что у Вас нет чего-то, что точно должно быть.

Эта команда почти стандартная, ее понимают практически все программы и часто ее можно записывать не полностью - INV или еще проще I.

HELP - помощь. В большинстве случаев программы дают подсказки, но не во всяком месте и не во всякое время. Если установите, что программа эту команду понимает, пользуйтесь ею регулярно, рано или поздно она Вам пригодится.

SAVE - по этой команде может производиться отгрузка состояния игры на ленту. Если программа такой возможности не имеет выключите ее и больше не включайте - это очень плохая программа, есть сотни гораздо более интересных. Соответственно, если она понимает SAVE, то должна понимать и LOAD (иногда впрочем вместо LOAD служит RESTORE).

LISTEN - слушать. Это аналог LOOK (смотреть), хотя встречается много реже, но зато может быть необходим для успешного завершения игры. В программе CASTLE, например, нам удалось в блужданиях по лабиринту с помощью этой команды услышать морзянку, которая содержала необходимый ключ к победе.

Урок 2. Перемещение в игровом пространстве.

Основным содержанием игр этого жанра является конечно путешествие. Вы перемещаетесь из точки в точку с помощью специальных команд и всюду занимаетесь исследованием. Все смотрите, испытываете, пробуете. Количество мест, в которых Вы можете побывать часто измеряется многими сотнями (отдельные пункты называют локациями). Вы можете на первом этапе своего ознакомления с игрой походить из точки в точку и никаких существенных действий не делать. В современных хорошо подготовленных программах смерть Вам от этого грозить не будет, хотя правда от пустого блуждания и эффективность тоже невелика. Напротив в ранних, более слабых программах блуждание по локациям нередко приводит к гибели в лапах какого-нибудь дракона и т.п.

Первое, что Вам совершенно необходимо в путешествии - это карта. Некоторые полагают, что карта - это технический прием, сродни отмычке, но это не так. Игру средней силы без карты не осилить, а если и осилить, то потеряв массу времени на бесполезные шаги.

Перемещение в программах выполняется командой GO (ИДТИ) с указанием направления, например:

GO NORTH - идти на север.

Поскольку такой способ перемещения стал уже стандартным, то и необходимость в команде GO в общем-то отпала и можно дать команду проще - NORTH, а в большинстве случаев вообще - N.

Совершенно аналогично для других стран света:

EAST - E - идти на восток

SOUTH - S - идти на юг

WEST - W - идти на запад.

Не забывайте также о возможности перемещения вверх или вниз. Эти направления тоже надо проверять, даже если по команде LOOK Вам и не сообщили, что туда есть проход.

UP - U - вверх

DOWN - D - вниз

В некоторых играх с довольно развитой географией встречаются и промежуточные направления движения. Это также нужно проверить и убедиться, понимает их программа или нет.

NE - идти на северо-восток

SE - идти на юго-восток

NW - идти на северо-запад

SW - идти на юго-запад

Итак, наиболее традиционный способ перемещения ходьбой ясен. Но иногда приходится преодолевать препятствия, тогда потребуется прыжок JUMP. Перемещаться можно также и бегом RUN, но в программах это встречается редко. Более того, иногда предполагается, что RUN выполняется сломя голову в бегстве от опасности и тогда по

результатам этой команды Вы можете оказаться в незнакомой месте с потерей ориентировки, чего конечно при простом GO на один пункт не бывает.

В путешествии бывает нужным войти куда-то (в лифт, в туннель, дверь), причем направление может быть неизвестным. В этом случае как правило используют глагол ENTER и соответственно случается:

ENTER LIFT
ENTER TUNNEL
ENTER DOOR

Перемещение с помощью технических средств обычно в программах упрощено до предела. Так, чтобы поехать на машине или на катере Вам может быть достаточно запустить мотор:

START MOTOR
START ENGINE ("KAYLETH")

Для поездки на лифте обычно достаточно нажать кнопку:

PRESS BUTTON

Редко, но встречается в программах передвижение с помощью животных (лошади, дракона например в NEVERENDING STORY).

В этом случае обычно используется глагол RIDE, например RIDE HORSE.

Переходя из пункта в пункт никогда не верьте тому, что сообщает Вам программа. Всегда давайте команду LOOK, не пренебрегайте и такими нетривиальными вещами как посмотреть вверх и посмотреть вниз LOOK UP и LOOK DOWN. В 999 случаях из 1000 это ничего не даст, но когда даст эффект, тогда Вы поймете, чего Вы чуть было не пропустили.

Помните, что самое интересное скрыто из описания локации, хотя какой-то намек все-таки обычно оставляют.

Пробуйте во время блужданий отыскивать проходы там, где они только могут быть - под коврами, за гобеленами, за картинами, за зеркалами, в шкафах и шифоньерах.

Часто срабатывают такие вещи как

ENTER MIRROR - войти в зеркало ("HI-MAN")
ENTER WARDROBE - войти в шкаф ("WITCH'S CAULDRON").

Теперь нам осталось только осветить такие экзотические способы перемещения как по воде и по воздуху.

По воздуху это обычно FLY - летать.

По воде - ROW - грести и SWIM - плыть.

Дополнительно еще укажем что иногда встречаются неожиданные телепортирующие устройства, которые могут требовать наличия некоторого кода или пароля, без которого не работают или работают неправильно, но это уже особый случай ("WORM IN PARADIZE", "KAYLETH" и др.), конечно этот код или пароль Вы должны отыскать сами по ходу программы.

ТЕХНИЧЕСКОЕ ПРИЛОЖЕНИЕ

ФЕВРАЛЬ 1991

ЭТОТ РАЗДЕЛ ВЕДЕТ НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС",
СПЕЦИАЛИЗИРУЮЩИЙСЯ В ВОПРОСАХ АППАРАТНОГО ОБЕСПЕЧЕНИЯ СИНКЛЕР-
СОВМЕСТИМЫХ КОМПЬЮТЕРОВ И СХЕМОТЕХНИЧЕСКИХ РЕШЕНИЙ ПЕРИФЕРИЙНЫХ
УСТРОЙСТВ

127566, МОСКВА И-566, НТК "ПЛЮС"

В течение последнего года и НТК "Плюс" и НТК "ИНФОРКОМ" получили десятки писем от читателей с просьбой рассказать о модемах. Сегодня мы выполняем эту просьбу и даем описание простейшего устройства этого типа.

Распечатку программной поддержки с шестнадцатиричным загрузчиком и контрольными суммами мы приведем в следующем выпуске.

ZX MODEM

С тех пор, как компьютеры вошли в нашу жизнь, мы не перестаем слышать о таком их великом достоинстве, как возможность устанавливать связь, обмениваться информацией, пользоваться удаленными банками данных и т.п. Например, эта страшная история о том, как компьютеры между собой играют в шахматы по телефону, а хозяева недоумевают по поводу огромных счетов за телефон или перерасхода электроэнергии. Или, что еще страшнее, как с домашнего компьютера ребенок по телефону соединяется с центральным компьютером оборонного ведомства и, думая, что играет в компьютерную войну, начинает настоящую Третью Мировую. Тем не менее, вот уже несколько месяцев или даже лет Вы владеете компьютером, а никакой связи почему-то не устанавливается, и вообще ничего страшного не происходит. А все потому, что Ваш компьютер до сих пор не оборудован модемом. Что такое модем? Если Вы в курсе дела, то следующие несколько абзацев Вам будут неинтересны.

Как Вы знаете, информация в компьютерах передается в двоичном виде, т.е. 0 или 1. Это означает, что по проводникам проходят прямоугольные импульсы.

Чем более прямоугольны и менее синусоидальны эти импульсы, тем шире их частотный спектр, например у прямоугольного импульса частотный спектр бесконечен. Зато конечен и весьма ограничен спектр частот телефонного канала.

По стандарту, принятому во всем мире, один телефонный канал занимает полосу частот от 300 Гц до 3400 Гц, т.е. равную 3100 Гц и сигналы, передаваемые по такому каналу, должны вписываться в эту полосу. Например, этой полосы достаточно для разборчивой передачи человеческой речи.

Если по телефонному каналу передавать сигналы с частотами, выходящими за пределы этой полосы, то они во-первых не дойдут до Вашего абонента, во-вторых в силу несовершенства полосовых фильтров они влезут в соседние телефонные каналы и либо будут мешать чужим абонентам, либо чужие абоненты Вас подслушают, это кому что больше нравится.

Вот для того, чтобы преобразовать компьютерные широкополосные импульсы в посылки подходящие для передачи по телефонному каналу и для обратного преобразования и служат модемы. Модем - словообразование от двух слов - модулятор и демодулятор. Модулятор получает последовательность нулей и единиц и выдает в телефонный канал синусоидальные сигналы звуковых частот, например 1200 Гц для передачи нулей. Демодулятор, соответственно, получает эти сигналы из телефонной линии и передает в память компьютера информацию в двоичном виде.

Разновидностей модема великое множество, и различаются они в основном по сложности и, соответственно, в цене. Самый простой модем, так называемый ручной (неавтоматический и неинтеллектуальный), выполняет описанную выше функцию. Он представляет собой некую коробочку, соединяемую с компьютером обычно с помощью

интерфейса RS-232C, который давно уже стал стандартом для связи с модемом. Еще более простой модем типа того, что опубликован в этом номере, обходится и без интерфейса связи. Он соединен непосредственно с шинами компьютера, а весь обмен происходит под управлением специальной программы. Это, конечно, отнимает у компьютера уйму времени и он ничем другим заниматься в это время уже не может, но в некоторых случаях годится и такой модем. Более сложные модемы берут на себя больше функций и меньше отвлекают компьютер. Например, они сами отвечают на вызов, или сами посылают вызов абоненту. Кроме того, модем может иметь возможность набирать номер телефона абонента.

Интеллектуальный модем имеет гораздо больше возможностей. Он имеет стандартизованный набор команд, которые получает от компьютера и выполняет самостоятельно, может иметь буферную память, самотестируется, поддерживает какой-либо протокол связи, настраивается в разные режимы работы, позволяет одновременно передавать и получать информацию и даже одновременно говорить по этому же телефонному каналу с Вашим абонентом. Возможности коммуникаций также сильно зависят и от программного обеспечения, используемого в вашем компьютере. Сильное программное обеспечение поможет Вам добиться удивительных результатов даже при использовании самых примитивных модемов, и связи по отечественным телефонным линиям. От качества телефонной линии сильно зависит достоверность передаваемой информации или время затраченное на передачу при полной достоверности информации. Помеха в линии может исказить передаваемые данные. Поэтому приходится применять какие-либо протоколы обмена. Самое простое и достаточно эффективное средство - передача битов паритета - битов контроля четности или нечетности информации. Бит паритета передается вместе с байтом данных и проверяется на приемной стороне. При несовпадении делается вывод о недостоверности информации. При более сложных протоколах связи поток данных разбивается на пакеты, т.е. части, длиной допустим по 256 байт, подсчитывается контрольная сумма и пакет передается с контрольной суммой в конце. После приема информации вновь подсчитывается контрольная сумма, если она совпадает с переданной, то пакет квитируется, т.е. подтверждается его получение. В противном случае передается требование повторить пакет. Такой же способ применяется например в радиолучительской пакетной связи. Более сложные протоколы связи занимают не только обнаружением, но и исправлением ошибок передачи данных с помощью например кодов Хемминга.

В этом номере мы предлагаем Вашему вниманию схему ZX модема. ZX модем подключается непосредственно к шинам вашего Спектрума, без использования промежуточного интерфейса. Кроме аппаратной части необходимо загрузить и запустить программу поддержки ZX модема, написанную в машинных кодах.

ZX модем - так называемый модем прямого включения, т.е. он включается непосредственно в телефонную линию, в отличие от модемов с акустической связью, где информация передается через телефон и микрофон телефонной трубки. При непосредственном включении модема необходимо соблюдать определенные стандарты, принятые в телефонной связи. В схеме ZX модема связь с телефонной линией осуществляется через трансформатор, выполняющий функции гальванической развязки. Тем не менее, остается под вопросом правомерность непосредственного включения в линию без особого разрешения предприятия связи. Надеемся, что Вы не столкнетесь с трудностями в этом вопросе. В любом случае, ничто не мешает Вам передавать Ваши данные по двухпроводной линии любой длины, связывающей Ваш модем с модемом Вашего абонента.

Схема ZX модема приведена на рис. 1. На рис. 2 изображена печатная плата. Если у Вас есть такая возможность, то можно изготовить печатную плату. В противном случае можно выполнить монтаж проводниками. Из рисунка печатной платы видно, что еще нужно будет установить несколько перемычек.

Трансформатор можно намотать на сердечнике с подходящим сечением, например от трансформатора транзисторного приемника.

Если Ваш Спектрум снабжен разъемом, то можно впаять ответный разъем и на плате

модема. Предлагаемая конфигурация разъема рассчитана на стыковку с фирменным Спектрумом.

Формат передачи.

Формат передачи и несущей частоты кода ZX MODEMa зависит исключительно от программного обеспечения, так что если будет нужно, его можно сильно изменить новым решением программной поддержки. Это позволит Вам при наличии навыков программирования полностью перестроить ZX модем, в результате чего Вы сможете стыковаться с модемами и компьютерами любого типа. Например, Вы могли бы перегонять тексты, подготовленные дома, на компьютер, установленный у Вас на работе с целью успешного решения производственных задач на дому.

В этом описании приведена программа, которая использует формат передачи от 600 до 1200 бит в секунду в форме: один стартовый бит, 8 битов данных, один бит четности и один столовый бит.

Частота передачи - 1200 Гц для бита единицы и 2400 Гц для нулевого бита.

Программная поддержка.

ZX MODEM пользуется программой в машинных кодах, которая занимает 1730 байтов и начинается с адреса 63000. Она служит для приема и передачи желаемого количества байтов из памяти "Спектрума".

Программа обеспечивает исполнение двух команд.

Первая - TRANSMIT, по ней передается желаемое количество байтов. После этой команды стоит число, которое означает количество байтов, которые будут посылаться (похожа на команду SAVE).

Другая команда - RECEIVE, после которой следует адрес, в который должны быть помещены данные. Возможно управление приемом и передачей байтов и из БЕЙСИКа, через функцию DEF FN.

Во время приема ZX MODEM выполняет проверку четности принимаемых байтов и, в случае ошибки, оповещает о числе неправильно принятых байтов.

Технические испытания показывают, что на 100 000 переданных байтов чаще всего появляется не более одной ошибки. Поэтому предлагается передача по частям (по 5 Кбайт) с повторением блока, если возникла ошибка.

Когда наберете программу, которую мы приведем в следующем выпуске, обязательно исполните копию на кассету с помощью команды:

```
SAVE "modem" CODE 63000,1734
```

- иначе при возникновении ошибки рискуете остаться ни с чем.

Прежде, чем начнете записывать коды в память, опустите системную переменную RAMTOP командой

```
CLEAR 62999.
```

Программа запускается командой

```
RANDOMIZE USR 63000.
```

Следует напомнить, что программа написана исключительно для модели SPECTRUM 48K и программа не настраивается.

Проверка работы

Модем подстыкуйте к выходному разъему "Спектрума" и, после того, как считаете программу с кассеты, наберите команду

```
TRANSMIT 0,100.
```

Когда нажмете ENTER, начнется пересылка данных, и на модеме загорится зеленый светодиод CTS (Clear to send - готовность пересылки) и красный светодиод DCD (Data carrier detect - есть передача данных).

После этой проверки модем можно подсоединить к телефонной линии или с помощью обычной двухпроводной связи соединить с другим модемом.

Модем подключается параллельно телефонному аппарату.

Во время передачи данных переключатель на модеме должен быть включен.

ВНИМАНИЕ!

Имейте в виду, что если после обмена данными переключатель будет включен, то телефонная линия будет все время занята, даже после разрыва связи на телефонном аппарате.

Имейте в виду, что модем должен подключаться и отключаться (подсоединяться и отсоединяться) только когда "Спектрум" отключен от сети, то есть не под напряжением. Это важно и для всех других аппаратных приставок. В противном случае компьютер легко может выйти из строя.

Телефонная связь устанавливается обычным способом, для чего служит телефонный аппарат. После того, как Вы и Ваш абонент, с которым Вы обмениваетесь данными, подключите свои модемы и запустите нужную программу, выполните старт-предписание (TRANSMIT, RECEIVE).

Во время самого обмена данными нет необходимости опускать телефонную трубку, но рекомендуется обеспечивать тишину в зоне трубки, иначе может появиться ошибка в приеме данных.

ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ

IC1 - 74LS01 (555ЛА8)

IC2 - LM324 (1401УД2)

IC3 - 74LS02 (555ЛЕ1)

IC4 - 74LS00 (555ЛА3)

TR1 - КТ3102

R1 - 820 Ом

D1 - КД522

R2 - 10K

LEDr - АЛ307

R3 - 200K

LEDg - АЛ307

R4 - 100k

C1 - 100 нФ

R5 - 56K

C2 - 100 нФ

R6 - 62K

C3 - 10,0

R7 - 680K

C4 - 1H5

R8 - 100K

C5 - 1H8

R9 - 100K

C6 - 1 нФ

R10 - 1.5K

C7 - 120 пф

R11 - 820 Ом

C8 - 470 пф

R12 - 1M

C9 - 1 нФ

R13 - 10K

C10 - 1 нФ

C11 - 100 нФ

C12 - 10,0

C13 - 10,0

Трансформатор.

S = 0,3 кв. см

1-2 = 450 витков D=0.07 мм

3-4 = 450 витков D = 0.07 мм

5-7-6 = 2x250 витков D=0,07 мм

(Окончание в следующем выпуске)

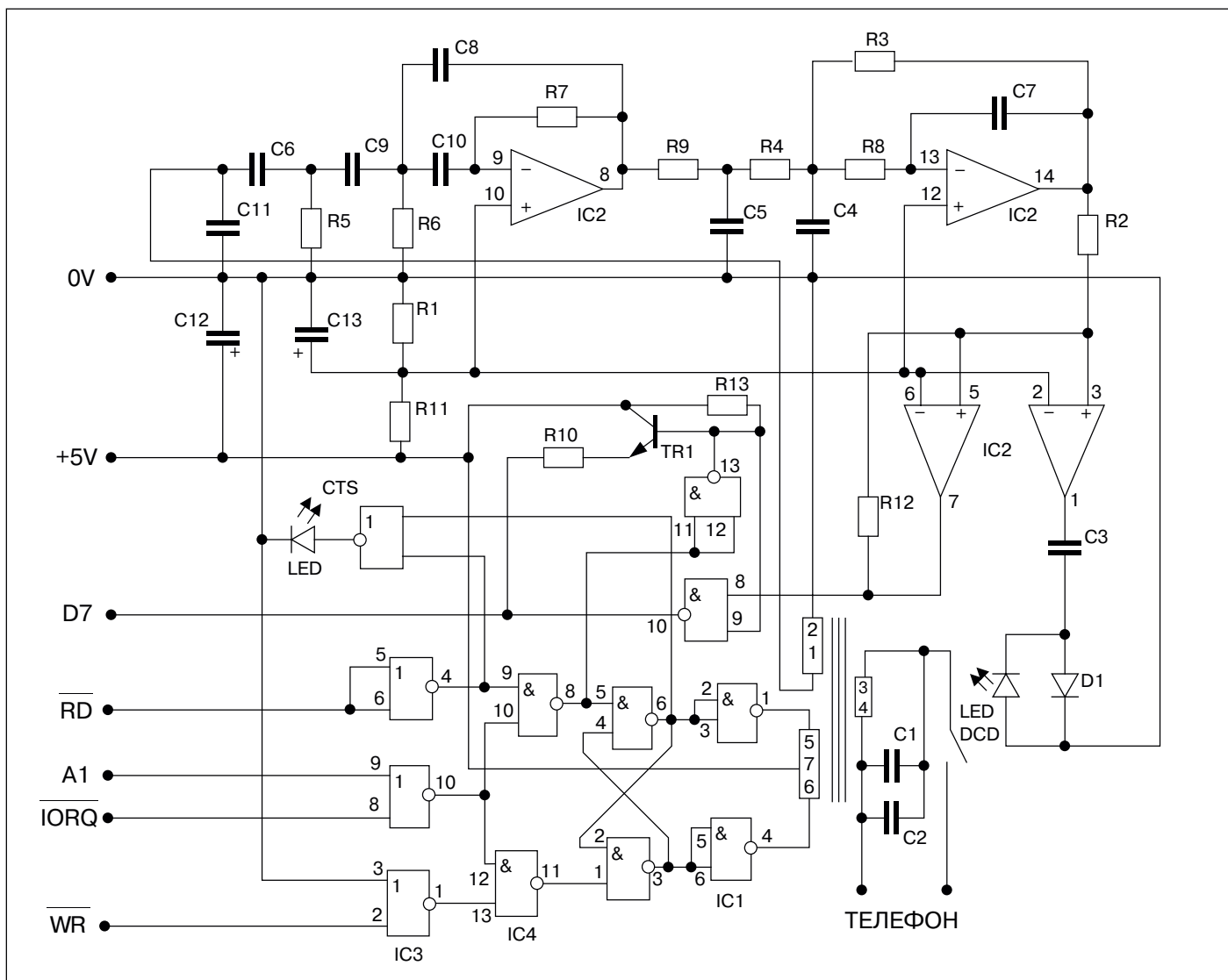


Рис.1
ZX - MODEM. Принципиальная схема.

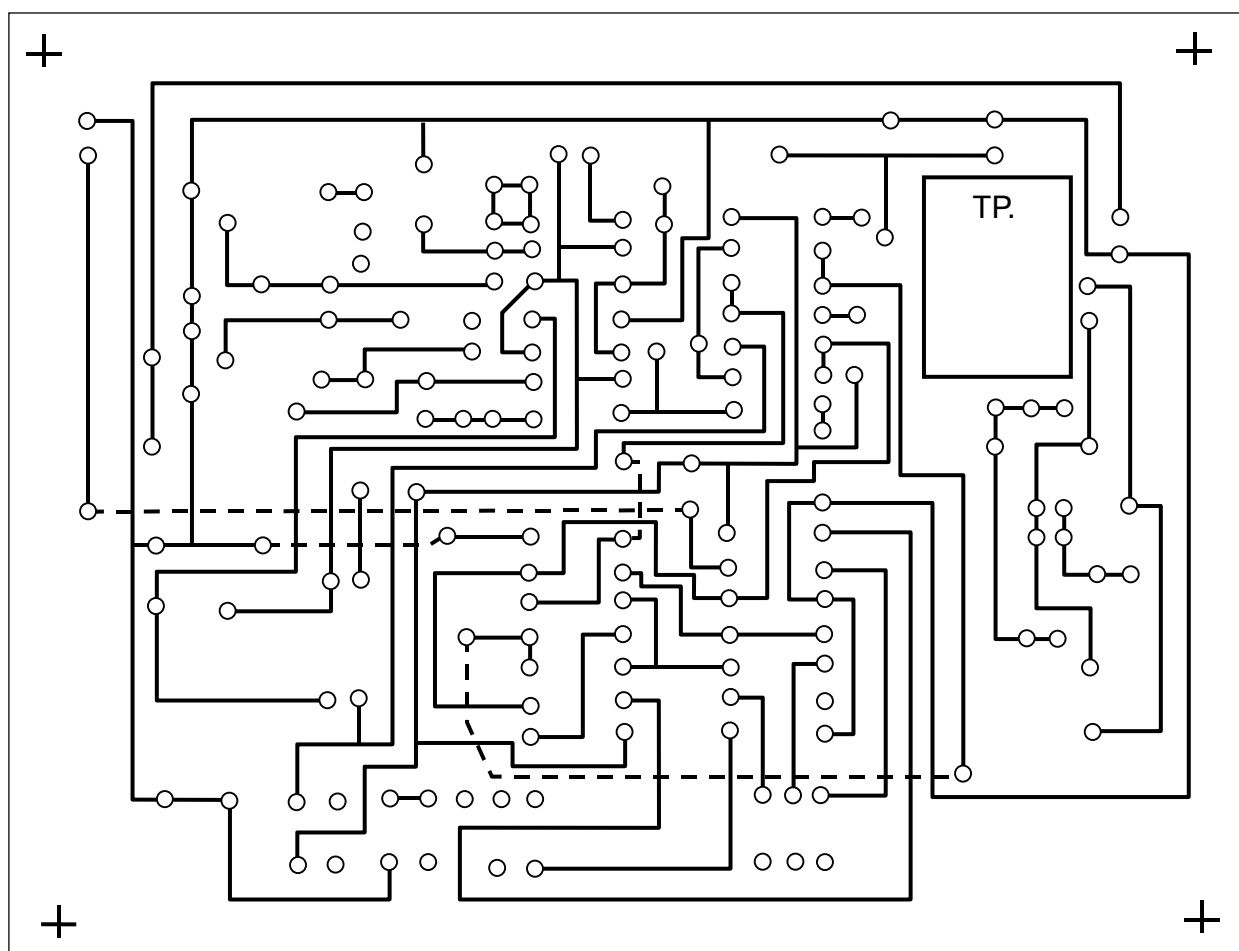
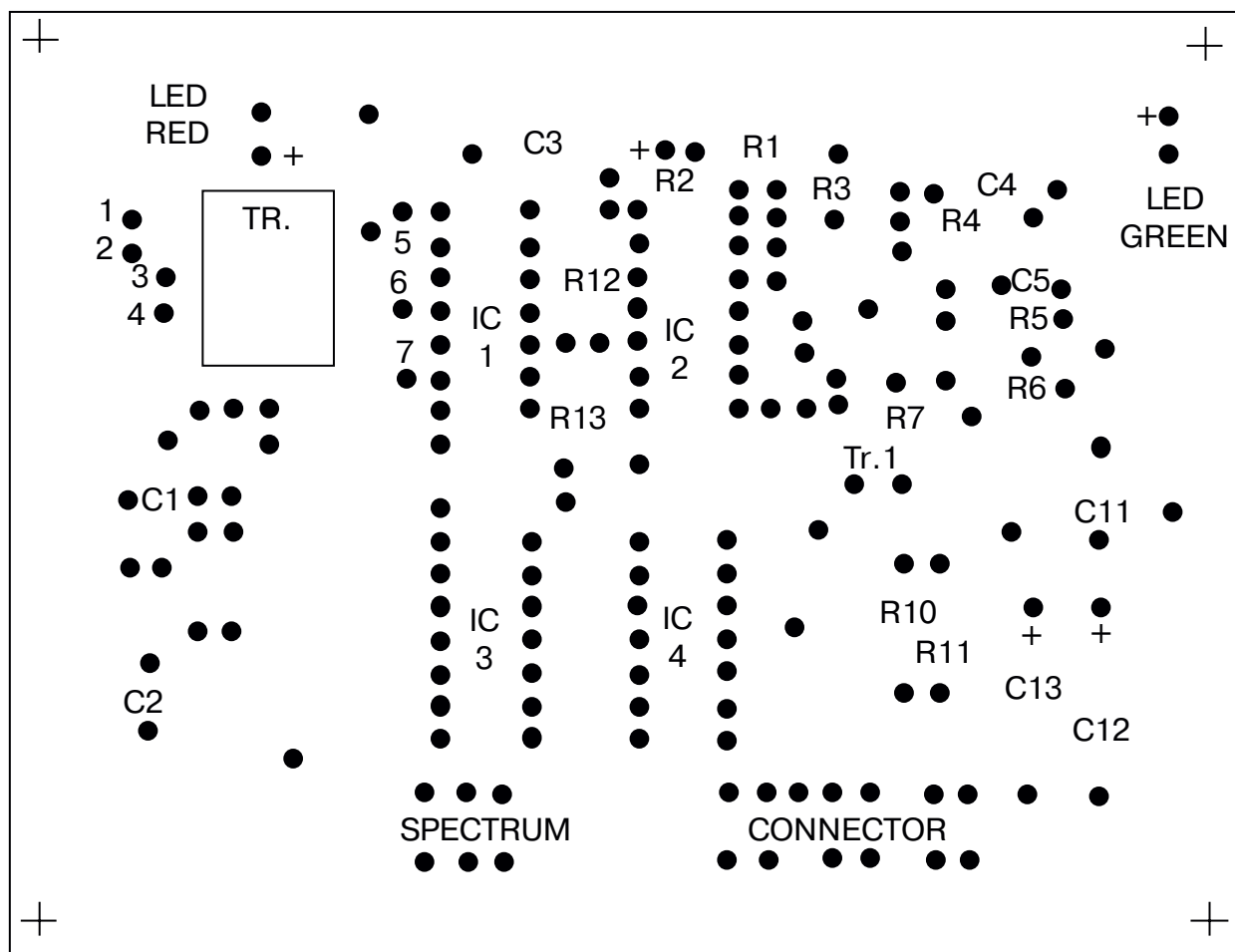


Рис. 2
ZX - MODEM. Печатная плата

РЕКЛАМНОЕ ПРИЛОЖЕНИЕ

ФЕВРАЛЬ 1991

НТК "ИНФОРКОМ" не может принимать на себя никакую ответственность за точность и достоверность сведений, содержащихся в объявлениях читателей.

ИНФОРКОМ

ищет зарубежные журналы, посвященные персональному компьютеру ZX-SPECTRUM:

SINCLAIR USER

ZX-COMPUTING

YOUR SPECTRUM

SINCLAIR PROGRAMS и др.

Возможные варианты:

- покупка по цене, назначенной Вами;
- обмен оригиналами;
- обмен копиями;
- обмен на время, с целью копирования.
- и др. условия по Вашему желанию.

При обращении просим предоставить список имеющихся у Вас журналов с указанием названия, года, номера, а также указать оригинал/копия.

ВАЖТЕК просьба не предлагать.

107241, Москва, Б-241, а/я 37, НТК "ИНФОРКОМ"

Купим патентно чистые игровые программы для Синклер-совместимых ПК емкостью ОЗУ 48К.

С предложениями обращаться по адресу: 600009, г.Владимир, ул.Батурина, 28, Владимирское КБ радиосвязи.

Вышлю наложенным платежом:

- 1) описание пакета MONS/GENS - 20 руб.
- 2) руководство пользователя SPECTRUM48 - 15 руб.
- 3) SPECTRUM изнутри - 15 руб.

Обращаться по адресу:

313510 Харьковская обл., пос.Краснопавловка, д.10, кв.43 Шпаковскому С.В.

Предлагаю описания TASWORD, BETABASIC и др. Цены низкие.

692438 Приморье, Краснореченск, Гастелло 13-8 Солдатову

Обменяюсь или запишу программы к ПК "СПЕКТРУМ" (система Синклер). Имею более 400 Фирменных программ на кассетах. Вышлю каталог.

Мой адрес: 211440 г.Новополоцк, Витебской обл, п/о N8, а/я 43, Геннадью.

Продаю компьютер "ZX-SPECTRUM" (вариант "Ленинград-2") с техописанием и программным обеспечением (10 кассет МК-60). Гарантийный срок - 3 года.

Цена (без ПО) - 900 руб.

Обращаться по адресу: 256300 Киевская обл, г.Борисполь, ул.Котовского 3 кв.77

Кысиль Григорий Николаевич.

Продам компьютер "ZX-SPECTRUM" ленинградский вариант, а также отлаженную плату компьютера данной версии.

Адрес: 140520 Московская обл., п.Белоомут-2 а/я 1, Чаплыгин В.Д.

Куплю принтер к ПК "SPECTRUM". Обменяюсь программами к "SPECTRUM", имею около 300.

Обращаться по адресу:

460048, г.Оренбург, пр.Победы, д.144"А", кор.2, кв.572 Щедринову Ю.Н.

Обменяю или куплю кварцы 8 и 14 МГц. Имею на обмен кварцы 1, 2, 3.2, 5, 6 и 10 МГц.

Куплю м/сх 555АГ3, ТМ9, 155ЛП11.

Обращаться по адресу:

111555, Москва, ул.Молостовых д.11 кор.1 кв.87, Константинову Л.М. (тел.303-81-03).

Предлагаю программы и литературу в ZX-Spectrum. В коллекции около 1000 программ. Каталог высылается бесплатно. Возможен обмен.

164413, п/о Катунино, Архангельской обл., ул. Катунина, д.4, кв.20 Костюченко Ю.К.

По самым низким ценам предлагаю широкий ассортимент описаний программного обеспечения (в том числе и игр) и документации к ZX-Spectrum.

Каталог высылается бесплатно и оперативно. 310085 г.Харьков, Продольная 3 кв.4 Фонарев О.В.

Запишу программы к компьютеру ZX-Spectrum на кассеты заказчика. Имею более 400 фирменных программ. Качество гарантирую. Каталог бесплатно

211440 г.Новополоцк-9, Витебской обл, а/я 6, Саша

ВЫ ИНТЕРЕСУЕТЕСЬ КОМПЬЮТЕРОМ ZX-SPECTRUM?

ВАМ НЕОБХОДИМО ПРИОБРЕСТИ НАШ КОНТРОЛЛЕР ПРОФЕССИОНАЛЬНОГО
ДИСПЛЕЯ!!!

ВЫ ПОЛУЧИТЕ:

- на экране телевизора, черно-белого или цветного дисплея 24 строки по 80 символов, восемь стандартных цветов ZX-Spectrum, две градации яркости, атрибут мигания;
- возможность выбрать с помощью переключателя матрицу знакогенератора 8x8 или 8x14;
- полную программную совместимость с интерпретатором БЕЙСИКа, ассемблерами GENS, ZEUS, MRS, отладчиками MONS (который теперь можно загружать в область экрана с адреса #4000), MONITOR48, трансляторами ПАСКАЛЯ, СИ, ЛИСПа, ПРОЛОГа и любыми другими программами, работающими через процедуру RST 0010;
- новые качества любимых программ, возможность с помощью 4 операторов РОКЕ задать произвольное окно;
- легкость подключения: достаточно вставить контроллер в системный разъем, не меняя ни схему машины, ни ПЗУ;
- за дополнительную плату базу данных и текстовый процессор, адаптированные к работе с контроллером.

НАШ КОНТРОЛЛЕР СОХРАНЯЕТ ВАШЕ ЗРЕНИЕ,
ЭКОНОМИТ ВРЕМЯ ПРИ НАПИСАНИИ ПРОГРАММ
С ОКОННЫМ ИНТЕРФЕЙСОМ!!!

Пишите: 270026, г. Одесса-26, а/я МП "КИТ"
Звоните: в г. Одессе 23-69-54 с 10 до 14

Если Вы хотите вдвое сократить время ввода/вывода программ, уменьшить износ головок магнитофона и обойтись вдвое меньшим количеством кассет, Вас заинтересует

ДВУХРЕЖИМНЫЙ
МОДУЛЬ
ТУРБООБМЕНА
С МАГНИТОФОНОМ

Наш модуль позволяет наравне со стандартным форматом реализовать формат двойной плотности (3000 бод), не занимая при этом ОЗУ, чего невозможно достичь другими, чисто программными способами. Модуль может быть подключен к любому варианту "ZX-SPECTRUM" (Синклер), не затрагивает его программную совместимость. При копировании и трансформации программ из одного формата в другой, не требует специальных копировщиков, позволяя переключать режимы во время работы компьютера. Модуль высылается наложенным платежом.

Стоимость модуля с документацией 42 рубля.

Наш адрес: 270013 Одесса, а/я 12, "КРОСС"

ПЛЮС

НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС" - ПРЕДЛАГАЕТ ВАШЕМУ ВНИМАНИЮ СВОИ УЧЕБНО-МЕТОДИЧЕСКИЕ РАЗРАБОТКИ

Методическая разработка (МР) N1.

Любители компьютерных игр найдут в ней несколько вариантов схем подключения джойстика. Наличие нескольких вариантов позволяет Вам выбрать наиболее приемлемый по наличию элементной базы.

Те, у кого есть принтер, смогут подключить его с помощью одной из схем последовательного или параллельного интерфейса принтера. Приведенные программы поддержки принтера (драйверы) позволят Вам распечатывать как текстовую, так и графическую информацию. Схемы интерфейсов и программы поддержки предназначены для работы с любыми принтерами, оборудованными одним из стандартных интерфейсов: CENTRONICS (ИРПР-М), ИРПР (IFSP), RS-232C (V.24).

Для занимающихся компьютерной графикой и живописью предназначена схема светового пера и распечатка работающего с ним графического редактора LP48K.

Схема программатора для ПЗУ с УФ стиранием позволит любителям технического творчества самостоятельно программировать такие микросхемы как 2716, 2764, 27128, 27256, 573 РФ2, 5, 4, 6, 8.

Объем МР1 - 34 страницы, стоимость - 30 рублей.

Со времени начала ее распространения (с сентября 1990г.) в адрес НТК не прекращается поток писем с положительными отзывами. Все схемы, входящие в разработку уже повторены многими пользователями, которые отмечают простоту изложения материала, доступность элементной базы и высокую повторяемость схем.

Методическая разработка (МР) N2.

Вторая методическая разработка МР2 состоит из двух частей. Первая часть посвящена контроллеру накопителя на гибких магнитных дисках НГМД. Контроллер полностью совместим с фирменной системой BETA-DISC INTERFACE. Он выполнен на базе микросхемы 1818ВГ93 (WD1793) и поддерживает работу до 4-х НГМД - 3-х или 5-ти дюймовых, односторонних и двусторонних, 40 и 80 дорожечных. Разработка содержит подробное описание схемы, сигналов, принципов работы интерфейса, рекомендации по сборке и наладке, систему команд дисковой операционной системы TRDOS.

О преимуществах работы с НГМД по сравнению с работой с магнитофонной лентой знают все, это новый уровень работы с компьютером, и мы надеемся, что наша разработка поможет Вам перейти на этот уровень.

Вторая часть МР2 посвящена стыковке компьютера с бытовыми телевизорами. Наиболее качественное цветное изображение получается при стыковке компьютера по R-G-B входу телевизора. При стыковке по антенному входу с кодировкой по системе СЕКАМ или PAL происходит некоторая потеря качества цветного изображения, но зато телевизор не требует переоборудования. В некоторых случаях может удовлетворить стыковка через видеовход с чернобелыми или цветными телевизорами без передачи информации о цвете, т.е. в черно-белом изображении.

Вторая часть МР2 содержит схемы всех перечисленных способов стыковки компьютера с телевизорами, причем схемы предназначены не только для полупроводниковых, но и для ламповых телевизоров.

Стоимость МР2 - 30 руб.

Методическая разработка (МР) N3.

Третья методическая разработка (МР3) подготовлена по многочисленным письмам и пожеланиям пользователей "Спектрума". В нее входит принципиально новый способ русификации "Спектрума", выгодно отличающийся от известных способов.

Знакогенератор кириллицы размещается в неиспользуемой области ROM, там же

размещены некоторые подпрограммы, которые добавляют в операционную систему "Спектрума" два новых регистра клавиатуры R и N к уже имеющимся K, L, C, E, G. Таким образом, в любой строке могут быть представлены буквы как русского, так и латинского алфавита одновременно, т.е. например, операторы Бейсика на английском языке, а текст на русском.

Также приведены дополнения к программам поддержки интерфейсов принтеров, опубликованных в нашей МР1 для вывода на печать смешанного русско-латинского текста.

Кроме этого, в МР3 по просьбам пользователей включены схемы интерфейсов джойстиков, рассчитанных на двух игроков - INTERFACE 1 и SINCLAIR, а так же схема программируемого джойстика, позволяющая запрограммировать все положения рукоятки и кнопки в соответствии с клавишами клавиатуры и более простой вариант - механически программируемый (путем перестановки перемычек) джойстик.

Стоимость МР3 - 30 руб.

Все программы, входящие в разработки, приведены в виде листинга и блоков шестнадцатичных кодов.

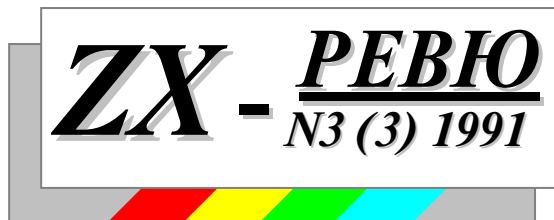
Свои заявки на приобретение МР1-МР3 Вы можете направлять по адресу:

127566, Москва, И-566, НТК "Плюс". Оплата производится только наложенным платежом.

Просьба при всех обращениях к нам прикладывать к письму конверт с обратным адресом. В заявке необходимо указать свой полный адрес, фамилии, имя, отчество.

PS. Мы приносим извинение всем нашим заказчикам по поводу задержки МР2 и МР3. Ввиду непрерывных изменений правил игры мы постоянно сталкиваемся с трудностями в реализации наших планов и, тем не менее, продолжаем рекламировать нашу деятельность, рассчитывая в ближайшее время удовлетворить Ваши заявки. Это в наших интересах. Мы обслуживаем посредством наложенного платежа, но есть несколько заказчиков, выславших нам предоплату за эти разработки. Мы приносим им особые извинения и обратимся к ним письмом персонально к каждому.

С уважением! Члены НТК "ПЛЮС"



НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ИНФОРКОМ", 10724-1, МОСКВА, Б-241 а/я 3

ДОРОГОЙ ЧИТАТЕЛЬ

Этот номер несколько отличается от всех предыдущих. В нем нет ставшей традиционной рубрики "СЕКРЕТЫ ПЗУ". Дело в том, что очередной материал этой рубрики посвящен процедурам, обеспечивающим ввод/вывод с магнитофона. Материал очень важный и сложный. В один номер его не вместить, а делить пополам мы не стали, чтобы не нарушить логичность изложения.

Мы приняли такое решение - следующий номер будет двойным, т.е. 4-ый и 5-ый выпуски мы вышлем вместе. Ждите их ориентировочно 20 мая.

С уважением.

"ИНФОРКОМ"

РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ

БЕЙСИК ШАГ ЗА ШАГОМ

Продолжение, начало на стр 23.

Работа с символьным набором.

Вывод символьного набора.

```
10 FOR F = 32 TO 255
20 PRINT CHR$ A;" ";
30 NEXT A
```

Вывод символа по его номеру от 32 до 255.

```
10 INPUT A
20 PRINT A,CHR$ A
30 GO TO 10
```

Вывод таблицы кодов.

```
10 FOR A = 32 TO 127
20 PRINT A,CHR$ A
30 NEXT A
```

Заполнение экрана графическими символами.

```
10 FOR N = 1 TO 704
20 PRINT INK RND*6;CHR$(131+(RND>.5));
30 NEXT N
```

Увеличенный рисунок любого символа с клавиатуры (для продолжения - нажатие любой клавиши за исключением "N").

```
10 CLS
20 INPUT "ВВЕДИ ЗНАК",A$
30 LET X=15616+(CODE A$-32)*8
40 LET Z = 0
50 FOR I = X TO X + 7
60 LET K = PEEK I
70 FOR J = 7 TO 0 STEP -1
80 PRINT AT Z,J; "■" AND (K-2*INT(K/2))=1
```

```

90 LET K = INT (K/2)
100 NEXT J
110 LET Z = Z + 1
120 NEXT I
130 PAUSE 0
140 GO TO 10

```

Беспорядочное заполнение экрана.

```

10 LET X = 16
20 LET Y = 11
30 PRINT AT Y,X; "*"
40 LET R = INT(RND*4)
50 LET X = X +(R=0)-(R=1)
60 IF X>31 OR X<0 THEN LET X=0
70 LET Y = Y +(R=2)-(R=3)
80 IF Y>21 OR Y<0 THEN LET Y=0
90 GO TO 30

```

Занавес.

```

10 OVER 1
20 FOR C = 0 TO 2
30 FOR A = 0 TO 1
40 FOR B = 0 TO 255 STEP 10
50 BEEP .01,B/10
60 PLOT 0,175*A
70 DRAW B,175-350*A
80 PLOT 255,175*A
90 DRAW -B,175-350*A
100 NEXT B: NEXT A: NEXT C

```

Зачерчивание экрана линиями.

```

10 BORDER 2
20 FOR I = 0 TO 50

```

Построение вертикальных линий

```

30 LET N = INT(RND*Z55)
40 PLOT N,0
50 LET M = INT (RND*127)
60 IF(M+N)>255 THEN GO TO 50
70 DRAW M,175
130 NEXT I

```

Добавив эту часть программы, мы будем строить и горизонтальные линии.

```

80 LET N = INT(RND*175)
90 PLOT 0,N
100 LET M = INT(RND*88)
110 IF(M+N)>175 THEN LET M = -M
120 DRAW 255,M

```

Прыгающий НЛО.

```

10 LET B=INT(RND*30)
20 LET A=INT(RND*18)
30 PRINT AT A,B+1;"0"
40 PRINT AT A+1,B;"<8>"
50 PRINT AT A+2,B+1;"H"
60 PAUSE 10
70 CLS
80 PRINT AT A+1,B+1;"0"
90 PRINT AT A+2,B;"<8>"
100 PRINT AT A+3,B+1;"H"
110 PAUSE 10
120 CLS
130 GO TO 10

```

Попробуйте после строки 70 ввести:

```

75 GO TO 10

```

Узор.

```
10 LET A$=""
20 FOR X = 1 TO 7
30 LET A$=A$+CHR$(RND*14+129)
40 NEXT X
50 INK RND*6
60 BORDER RND*7
70 FOR N=0 TO 87
80 PRINT A$
90 NEXT N
```

Мозаика.

```
10 BORDER 0: CLS
20 LET H = 16
30 LET V = 11
40 LET X = INT(RND*3-1)
50 LET Y = INT(RND*3-1)
60 INK RND*7
70 FOR Z=1 TO 20
80 PRINT AT V,H; CHR$ 143
90 LET H=H+X
100 LET V=V+Y
110 IF H < 0 THEN LET H=31
120 IF H > 31 THEN LET H=0
130 IF V < 0 THEN LET V=21
140 IF V > 21 THEN LET V=0
150 NEXT Z
160 GO TO 20
```

Попробуйте присвоить строке 70 номер 35, а строку 70 убрать.

Далее все программы - "звучащие". Это варианты использования оператора BEEP. Большинство из этих программ можно использовать в своих программах.

Звуки.

```
10 FOR Y = -2 TO 4
20 FOR X = 0 TO 6
30 BEEP .02,X+Y
40 NEXT X
50 NEXT Y
```

Измените .02 на .2 или на .002

Полный диапазон звуков Вашего компьютера.

```
10 FOR X = -60 TO 69
20 BEEP .02,X
30 NEXT X
10 FOR A = .01 TO .1 STEP .01
20 FOR B=46 TO 32 STEP -4
30 BEEP .15-A,B
40 NEXT B
50 NEXT A

10 INPUT B
20 FOR Q=20-B*19 TO 1+B*19 STEP -1+B*2
30 BEEP .01,0
40 NEXT Q
50 GO TO 10
```

"B" находится в диапазоне -1 до 3.

Имитатор сирены. Обратите внимание на разницу в наборе первой цифры в 20 и 30 строках (варианты эквивалентны).

```
10 FOR N = 1 TO 10
20 BEEP .15,17
```

```

30 BEEP 0.15,12
40 PAUSE 3
50 NEXT N

```

Имитация прыгающего шарика.

```

10 FOR N = 85 TO 0 STEP -2
20 BEEP .01,55
30 FOR A = N TO 0 STEP -1
40 NEXT A
50 NEXT H

```

```

10 FOR N = 1 TO 25
20 LET P = INT(RND*40) - 30
30 BEEP .05,P
40 BEEP .05,P+7
50 BEEP .05,P+4
60 NEXT N

```

```

10 FOR P=1 TO 48 STEP .5
20 BEEP .01,P
30 BEEP .01,P-6
40 NEXT P

```

Далее идут программы, основой которых является использование операторов PLOT, DRAW, CIRCLE.

Рамка.

```

10 FOR N = 9 TO 175
20 PLOT 0,N
30 DRAW 8,0
40 PLOT 247,(175-N)
50 DRAW 8,0
60 NEXT N
70 FOR N = 9 TO 255
80 PLOT N,175
90 DRAW 0,-8
100 PLOT (255-N),0
110 DRAW 0,8
120 NEXT N

```

Рамка.

```

10 FOR F = 0 TO 31
20 PRINT AT 0,F;" "
30 PRINT AT 21,F;" "
40 NEXT F
50 FOR F = 0 TO 21
60 PRINT AT F,0;" "
70 PRINT AT F,31;" "
80 NEXT F

```

Спираль. Попробуйте развернуть спираль в другую сторону.

```

10 PLOT 128,88
20 FOR M=1 TO 176 STEP 4
30 DRAW 0,M
40 DRAW -(M+1),0
50 DRAW 0,-(M+2)
60 DRAW (M+3),0
70 NEXT M

```

Цветные окружности.

```

10 CLS
20 CIRCLE INK RND*6; FLASH RND; 120+RND*8, 80*RND*6, RND*80
30 BEEP .1, RND*60

```



```
40 IF RND>.9 THEN GO TO 10
50 GO TO 20
```

Попробуйте убрать из строки 20 INK и FLASH. Попробуйте убрать строку 30.

Вложенные прямоугольники.

```
10 LET X = 0
20 FOR N = 170 TO 0 STEP -4
30 LET X = X + 2
40 PLOT X,X
50 DRAW N+80,0
60 DRAW 0,N
70 DRAW -(N+80),0
80 DRAW 0,-N
90 NEXT N
```

Предлагается четыре варианта построения круга.

Первые три строки в программах построения круга одинаковы, поэтому при наборе следующих программ эти можно оставить.

```
10 LET X= 128
20 LET Y=88
30 LET R=30
40 LET A=40
50 FOR F=-A TO A
60 LET C=R*SQR ABS(1-F*F/(A*A))
70 PLOT X+F,Y+C
80 DRAW 0,-2*C
90 NEXT F
```

```
10 LET X=128
20 LET Y=88
30 LET R=30
40 FOR F=-R TO R
50 PLOT X+F,Y+SQR ABS(R*R-F*F)
60 DRAW 0,-2*SQR ABS(R*R-F*F)
70 NEXT F
```

```
10 LET X=128
20 LET Y=88
30 LET R=30
40 FOR F=0 TO 2*PI STEP PI/180
50 PLOT X,Y
60 DRAW R*COS F,R*SIN F
70 NEXT F
```

```
10 LET X=128
20 LET Y=88
30 LET R=30
40 FOR F=1 TO R
50 CIRCLE X,Y,F
60 NEXT F
```

Ромб.

```
10 LET X=128
20 LET Y=88
30 LET R=30
40 LET A=PI/3
50 FOR F=-R TO R
60 PLOT X+F,Y-R
70 DRAW R*COS A,R*SIN A
80 NEXT F
```

```
10 LET X=128
20 LET Y=88
30 LET R=30
```

```
40 FOR F=-R TO R
50 PLOT X+F,Y
60 DRAW 0,2*R
70 NEXT F
```

Треугольник.

```
10 BORDER 1
20 PAPER 6
30 INK 2
40 CLS
50 FOR X= -60 TO 60
60 PLOT 128,150
70 DRAW X,-120
80 NEXT X
```

Убывающий квадрат.

```
10 FOR N=80 TO 0 STEP -2
20 PLOT 50+N,N
30 DRAW N,0
40 DRAW 0,N
50 DRAW -N,0
60 DRAW 0,-N
70 NEXT N
```

Квадраты.

```
10 BORDER 0
20 PAPER 0:CLS
30 FOR X=7 TO 0 STEP -1
40 INK X
50 FOR L=11-X TO 11+X
60 FOR C=16-X TO 16+X
70 PRINT AT L,C; "■"
80 NEXT C
90 NEXT L
100 NEXT X
```

Вложенные квадраты.

```
10 LET X=0
20 FOR N=120 TO 0 STEP -4
30 LET X=X+2
40 PLOT 50+X,X
50 DRAW N,0
60 DRAW 0,N
70 DRAW -N,0
80 DRAW 0,-N
90 NEXT N
```

Сетчатое поле для настройки монитора.

```
10 BORDER 2: PAPER 0
20 INK 7: CLS
30 FOR N=0 TO 255 STEP 16
40 PLOT N,0
50 DRAW 0,175
60 NEXT N
70 FOR N=0 TO 175 STEP 16
80 PLOT 0,N
90 DRAW 255,C
100 NEXT N
110 PLOT 255,C
120 DRAW 0,175
130 DRAW -255,0
140 PLOT 136,88
150 CIRCLE 136,88,88
```

MEGABASIC

Продолжение. Начало на стр. 3, 25.

Команда FONT_

МЕГАБЕЙСИК может оперировать с тремя наборами символов. Выбор набора выполняется с помощью команды FONT_ после которой записывают число, указывающее на выбранный набор:

FONT_0 - обычные символы компьютера ZX SPECTRUM;

FONT_1 - символы, подобные применяемым в компьютерах фирмы BBC, например в "MICRO ELECTRON";

FONT_2 - символы, подобные применяемым в компьютерах фирмы AMSTRAD, например в CFC 464.

Новые символьные наборы хранятся в ОЗУ: 1ый - с адреса 46000, а 2ой - с адреса 45000. Адрес произвольного символа из набора может быть найден по формуле $S+8*(CODE\ A\$-32)$, где S - начальный адрес символьного набора. Поскольку адреса Вам известны, Вы можете сами загружать туда какие Вам нужно наборы, предварительно приготовив их, например с помощью программы ARTSTUDIO.

Команды CHR\$, VDU, DOWN.

Символьный набор ZX SPECTRUM содержит печатаемые и управляющие символы. Несколько новых управляющих символов входит в МЕГАБЕЙСИК.

CHR\$1...CHR\$4 - выбор режима очередного окна;

CHR\$7 - установка символа в позицию курсора;

CHR\$24...CHR\$31 - выбор окна для выходной информации. CHR\$24 - окно 0, CHR\$31 - окно 7.

Новая команда VDU_ эквивалентна PRINT CHR\$. При записи VDU_65,56 на экране появится "AB". Строка символов может быть выведена в нижнюю часть экрана по команде DOWN_u,x,a\$, где u и x - соответственно строка и столбец, на которые выводится строка символов, a\$ - строка выводимых символов.

Команды SPRINT и PRINTER.

Командой SPRINT_x,y,a,b,a\$ можно вывести на экран символы любого размера. Здесь x,y - позиция первого символа на экране; a и b - коэффициенты увеличения символа в направлении x и y соответственно; a\$ - строка выводимых символов. Если достигнут правый край экрана, то запись появляется с левой стороны.

Команда PRINTER, обеспечивает вывод на принтер. Если после этой команды будет записан 0, то вывод будет выполняться только на экран, любое другое число - на принтер. Адрес подпрограммы вывода на принтер, написанной в машинных кодах - 59934. Обратите внимание, что в МЕГАБЕЙСИКЕ нет команд CLEAR#, OPEN#2 и CLOSE#2.

ГРАФИКА.

Команды CHANGE и SWAP.

В МЕГАБЕЙСИК включены некоторые команды, которые позволяют пользователю непосредственно манипулировать с файлом атрибутов. Команда CHANGE_ позволяет изменить определенную часть байтов атрибутов. После этой команды записывают 2 числа, представляющие собой маску и данные. Маска определяет какой из битов каждого байта файла атрибутов должен быть изменен. Данные указывают в чем должно состоять это изменение:

CHANGE_1 - логическое NOT. Все 1 в каждом байте становятся 0, а 0 становятся 1;

CHANGE_2 - логическое AND между всеми байтами файла атрибутов и значением маски;

CHANGE_3 - логическое OR между байтом атрибутов и байтом данных.

Команда SWAP_ - позволяет поменять одни атрибуты на другие. После команды записываются два числа: первое число определяет значение новых атрибутов, а второе - значение тех атрибутов, которые должны измениться.

Команда FADE_

С помощью этой команды можно получить особые эффекты. Введите и запустите следующую короткую программу:

```
10 FOR A=0 TO 703
20 POKE 22528+A,PEEK A
30 NEXT A
40 FADE_0
```

Первые три строки этой программы обеспечивают заполнение файла атрибутов случайными числами, в 40-й строке по команде FADE_0 выполняется сравнение значения каждого байта файла атрибутов с числом, записанным после этой команды. Если эти значения равны, то байт остается неизменным, если же нет, то значение байта в файле атрибутов изменяется, пока не станет равным записанному числу.

Команда INVERT.

Это простая команда, по которой инвертируется весь экран, цвет INK изменяется на цвет PAPER, а цвет PAPER - на цвет INK.

Команда DEFG_.

Способ, позволяющий создавать в стандартном БЕЙСИКе новые символы графики пользователя (UDG) довольно громоздок. В МЕГАБЕЙСИКЕ же эта задача решается с помощью одной команды DEFG_, после которой записывается символ (любая из литер от a до u) а затем 8 чисел, разделенных запятыми. Символ определяет местонахождение UDG, а числа - его форму.

Команды GET_ и PUT_.

В МЕГАБЕЙСИКЕ можно запомнить часть изображения экрана в ОЗУ, и при необходимости вернуть это изображение на экран в любую позицию. Для передачи части экрана в ОЗУ используют команду

GET_0, a, y, x, d, w.

Здесь: a - адрес, начиная с которого хранится в ОЗУ изображение экрана, y,x - соответственно номер строки и номер столбца верхнего левого угла сохраняемого участка экрана, d,w - длина и ширина этого участка. Команды GET_ и PUT_ применяют такую же координатную систему, что и команды ATTR и SCREEN\$. Их координаты абсолютны и не зависят от положения верхнего левого угла окна на экране. Число байтов, занимаемых изображением экрана в ОЗУ, определяется по формуле $64 * w * d$.

Команда PUT_ прямо противоположна по действию команде GET_. Записывается она в следующей форме: PUT_f,a,y,x,d,w. Здесь f определяет способ восстановления изображения экрана:

f=0 - изображение из ОЗУ переписывается в дисплейный файл без изменений;

f=1 - восстановление изображения осуществляется через логическое OR значений дисплейного файла и сохраняемого в ОЗУ изображения;

f=2 - восстановление через логическое XOR;

f=4 - тоже, что и f=0, но вместо хранимых в памяти используются значения текущих атрибутов;

f=5 - тоже, что и f=1, но используются текущие атрибуты;

f=6 - тоже, что и f=2, но используются текущие атрибуты.

Переменные a,x,y,d,w аналогичны определяемым в команде GET.

Команда SPUT_.

Это вариация команды PUT_, позволяющая не только восстановить изображение экрана, но и увеличить его:

SPUT_a, x, y, b, c, w, d

Здесь: a - начальный адрес блока в ОЗУ; x,y - координаты верхнего левого угла экрана при восстановлении изображения: b,c - коэффициента увеличения по оси x и y соответственно. При этой команде всегда используются текущие атрибуты, а координатная система точно такая же, как и для команды SPRINT.

Продолжение в следующем выпуске.

128 K

Продолжение. Начало стр. 4, 26.

Итак, переход из режима 128K в режим 48K может программно выполняться достаточно просто, чего не скажешь об обратном переходе. Во всех "фирменных" инструкциях по работе со 128-килобайтными машинами написано, что из режима 48K в режим 128K войти невозможно, кроме как выключив и снова включив компьютер. Тем не менее, если из машинного кода напрямую работать с пятым битом порта 7FFD, то такой переход выполнить все-таки можно, для чего и служит приведенная ниже программа, написанная на языке АССЕМБЛЕРА.

Включите компьютер и войдите в режим 128K. После этого можете командой SPECTRUM перейти в режим 48K, но если Вы сразу войдете в 48K, то программа работать не будет. Прежде чем загрузить программу опустите RAMTOP командой CLEAR 32767. После загрузки Вы имеете в своем распоряжении две команды.

RANDOMIZE USR 32781.

Эта команда равноценна команде SPECTRUM - она выполняет переход в 48K, но при этом не генерирует на экране сообщение 0:OK, т.е. исполнение программы не прерывается и она может продолжать исполняться со следующего оператора, хотя и в другом режиме.

Противоположная команда -

RANDOMIZE USR 32814.

Эта команда выполняет возврат в режим 128K и опять же продолжает исполнение программы со следующего оператора.

Хотя есть и некоторые ограничения на такую работу с компьютером. Естественно, поскольку 128-й компьютер хранит системные переменные в области буфера ZX-принтера, то ничего в эту область засылать нельзя, иначе работа станет невозможной. Нельзя соответственно использовать и команды ZX-принтера - COPY и LPRINT, если Вы находитесь в режиме 48K. Из режима 128K - можно.

Такая организация работы может быть полезной тем, кто имеет компьютер 128K со встроенным последовательным портом RS232, если программная поддержка этого порта выполняется только в режиме 128K, а работать надо с программами (например с текстовым редактором), которые в режиме 128K не работают.

F5	SWAP_2	ORG 8000 PUSH AF	Сохранение на стеке регистров A, B, C и F.
C5		PUSH BC	
01FD7F		LD BC, 7FFD	В BC загружается порт 7FFD.
3A5C5B		LD A, (BANK_M)	В акк-р - текущая страница.
E61F		AND 1F	Выключается бит 5, что "отпирает" переключение режимов.
C3085B		JP 5B08, SWAP+8	Переход на финальную часть процедуры переключения режимов (размещена в ПЗУ).
ED7B3D5C	MODE_48	ORG 800D LD SP, (ERR_SP)	Машинный стек опустошается до адреса возврата по ошибке.
E1		POP HL	Ввод адреса возврата по ошибке.
210313		LD HL, 1303, MAIN-4	На стек идет адрес возврата при ошибке в режиме 48K.
E5		PUSH HL	
21761B		LD HL, 1B76, STMT_RET	На стек идет адрес возврата после исполнения оператора.
E5		PUSH HL	
2A455C		LD HL, (PPC)	В HL - номер текущей строки.
22425C		LD (NEWPPC), HL	Он считается номером следующей строки.
3A475C		LD A, (SUBPPC)	В аккумулятор - номер теку-

3C	INC A
32445C	LD (NSPPC), A
0600	LD B, 00
11BE15	LD DE, 15BE
1810	JR MODE_CHANGE

щего оператора.
Переход к следующему оператору в строке.
Выключение 4-го бита.
DE указывает на информацию канала "P" в режиме 48K.
Переход на переключение режима.

3EC3	MODE_128	ORG 802E
32005B		LD A, C3
210080		LD (SWAP), a
22015B		LD HL, SWAP, 2
0610		LD (SWAP+1), HL
115480		LD B, 10
		LD DE, 8054, P_128

В SWAP устанавливается инструкция JP.
Адрес перехода в SWAP указывает на данную программу.
Включение 4-го бита.
DE указывает на информацию канала "P" в режиме 128K.
Вызов переменной FLAGS,
Выключение 4-го бита.
Переключение 4-го бита в соответствии с действующим режимом.
Запоминание измененной переменной FLAGS,
HL указывает на начало области информации о каналах.

3A3B5C	MODE_CHANGE	LD A, (FLAGS)
E6EF		AND EF
B0		OR B

323B5C		LD (FLAGS), A
2A4F5C		LD HL, (CHANS)

010F00		LD BC, 000F
09		ADD HL, BC

HL указывает на информацию о канале "P".

EB		EX DE, HL
----	--	-----------

DE указывает на информацию о канале "P".

0E04		LD C, 04
------	--	----------

В BC - количество байтов, подлежащих копированию.

EDB0		LDIR
------	--	------

Копирование данных в область информации канала "P"

C9		RET
----	--	-----

Возврат

345B	P_128	ORG 6054
2F5B		DEFW 5B34, POUT
		DEFW 5B2F, PIN

Адрес выходной информации канала "P".
Адрес входной информации канала "P".

Тогда находясь в режиме 128K Вы открываете поток и устанавливаете параметр скорости обмена командой **FORMAT "P",n** , а затем переходите в режим 48K для работы, скажем со своим редактором. Теперь когда Вам надо что-то напечатать Вы переходите в режим 128K с немедленным возвратом в 48K:

RANDOMIZE USR 32814: LPRINT "что угодно" :RANDOMIZE USR 32781

Приведенная здесь программа является релоцируемой, то есть ее можно перемещать в другие удобные Вам адреса, надо только подправить адресацию в двух инструкциях, которые используют абсолютную адресацию к меткам **SWAP-2** и **P-128**.

И в завершение этого выпуска для тех, у кого 128-ая модель управляет двумя джойстиковыми портами без внешнего интерфейса дадим схемы распределения управляющих битов этих портов, а в следующем выпуске начнем рассматривать работу электронного диска 128-ой модели.

Первый джойстик опрашивается по порту **EFFE**, схема приведена на рис. 1, а второй - по порту **F7FE** - схема на рис. 2.

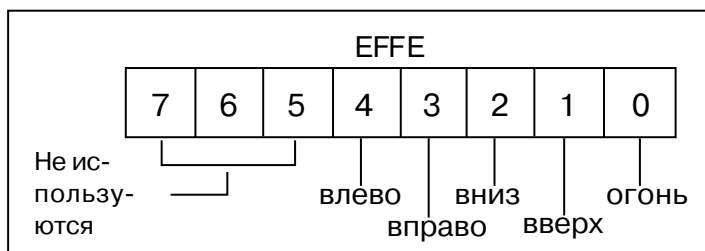


Рис.1

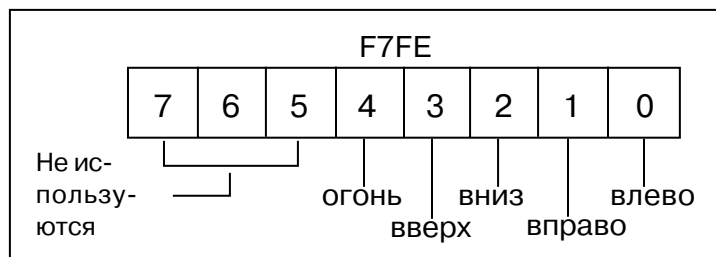


Рис.2

Пример загрузки данных с джойстикового порта в аккумулятор процессора для последующей обработки выглядит так:

```
LD BC, F7FE
IN A, (C)
```

Продолжение в следующем выпуске.

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД

Сегодня мы начинаем новый раздел, в котором слово будет предоставлено лучшим программистам мира, чей опыт может очень пригодиться тем, кто самостоятельно на своем домашнем компьютере решает важнейшую для страны задачу полной компьютеризации.

Мы открываем этот раздел циклом статей выдающегося английского программиста Стива Тернера (STEVE TURNER). Вы, конечно, знакомы с ним хотя бы по широко распространенной программе "QUAZATRON".

ЧАСТЬ 1.

Когда мне было пятнадцать лет, я вступил в один из первых британских компьютерных клубов. Машина, с которой мы работали, имела 1024 байта памяти и была размером с хороший стол. Прежде чем она могла запустить программу, в неё надо было ввести компилятор языка программирования, который хранился в виде огромной катушки с бумажной лентой. Самой главной и трудной задачей было отыскать у ленты начало и перемотать её по окончании работы.

Выдавал результаты своей работы компьютер на телетайп, а что касается ввода, то ввод с бумажной ленты был, пожалуй единственным способом, да правда была возможность вводить двоичные числа по одному за раз при помощи переключателей.

Вот так и началась моя страсть к компьютерам. С тех пор я работал со множеством машин - от огромных стационарных до крошечного ZX80. Работал также и на многих языках программирования, таких как КОБОЛ, ФОРТРАН, АЛГОЛ, освоил несколько машинных кодов. Семь лет занимался коммерческим программированием, достиг должности руководителя проекта и бросил все в первые дни после появления "Спектрума", чтобы начать собственное дело по разработке игровых программ и ни разу об этом не пожалел.

Меня часто спрашивают "Как начать программировать?", "Как изучить машинный код?", "Как написать игровую программу?" В ближайшие месяцы я постараюсь ответить Вам на эти вопросы и заодно попробую показать работу профессионального программиста игровых программ изнутри. Я дам несколько подсказок и полезных советов как начинающим, так и опытным программистам как по БЕЙСИКУ, так и по АССЕМБЛЕРУ.

ЦВЕТОВЫЕ АТТРИБУТЫ

Если Вы хотите взять от машины максимум возможного, то Вы должны хорошо представлять ее ограничения. Чем ограничены возможности дизайна? В чем этот компьютер особенно хорош, а чем он плох?

Самым сильным ограничивающим фактором в любой графической работе со "Спектрумом" является разрешающая способность цветовой карты экрана. Это означает, что если Вы недостаточно тщательно продумаете дизайн экрана, то вокруг нарисованных Вами объектов будут располагаться разноцветные квадраты и общее впечатление от графики будет ужасно примитивным. Есть несколько путей избежать этой проблемы.

1. Избегайте фона

Фон экрана лучше оставлять неокрашенным или почти неокрашенным, а ещё лучше делать его черным. В этом случае Вы можете спокойно перемещать по экрану цветные объекты и всё будет в порядке до тех пор, пока Ваши цветные объекты не начнут накладываться друг на друга. Такая техника хорошо подходит для космических игр, для аркадных игр направления "платформы и лестницы" В своей программе AVALON я использовал конфликт атрибутов в свою пользу. Когда главный герой перемещается, то цветовые атрибуты создают как бы сияние вокруг него, получается, что он как бы освещает сцену.

2. Только два цвета

Применение цвета в главной игровой зоне надо ограничить использованием не более двух цветов. У "Спектрума" достаточно хорошее графическое разрешение, которое

позволяет использовать полутоновую штриховку, чтобы получить ряд оттенков и текстур. Примеры такого подхода – KNIGHT LORE, FAIRLIGHT и мой QUAZATRON.

Вы можете несколько компенсировать отсутствие палитры цветов на основном игровом поле благодаря использованию цвета в статических (неподвижных) композициях вокруг игрового поля.

3. Много PAPER и мало INK

Цветовое решение всей игровой зоны надо делать в одном цветовом тоне и это должен быть цвет INK. Движущиеся предметы тогда разрабатываются цветами PAPER, и их перемещение по фону, выполненному INK не будет представлять таких сложностей.

ЗВУК

Отсутствие в "Спектруме" звукового сопроцессора означает, что если Вы хотите извлечь из компьютера звук, то Вам придется использовать время работы основного процессора. Чтобы одновременно мог воспроизводиться звук и работала бы программа, процессор надо разделять. Основной путь для этого - использование прерываний процессора. Конечно, самые лучшие музыкальные решения возможны в статические моменты игры, когда всю производительность процессора можно направить на обслуживание звуковых процедур. Проект игры должен обязательно учитывать те моменты, когда игра имеет естественную паузу.

СКОРОСТЬ РАБОТЫ ПРОЦЕССОРА

Самая выдающаяся особенность "Спектрума" - это скорость, с которой его процессор Z-80 может исполнять команды из своего большого набора, что позволяет программными средствами компенсировать отсутствие видеопроцессора и звукового процессора. Тем не менее, важно представлять себе ограниченность его быстродействия. Если, например, Вы хотите полностью перестроить весь экран (как будто Ваш герой вошел в другую комнату), то наилучшая скорость, которую Вы сможете достигнуть — пять экранов в секунду, но этого недостаточно для гладкой мультипликации.

Программа должна преодолеть это другими, более изощренными средствами или уменьшением размеров игрового поля. Поскольку в "Спектруме" процессор отвечает за всё, необходимо довести эксплуатацию его до предела путем эффективного программирования.

Есть один параметр, который определяет, насколько компьютер хорош для игрового программирования. Это отношение скорости работы процессора к размеру видеопамати. Чем больше этот параметр, тем лучше этот компьютер подходит для создания игр. Более высокое разрешение экрана тормозит видеоигру.

Примечание "ИНФОРКОМА": Для "Спектрума" этот параметр удивительно высок. При сравнительно стандартной скорости процессора он имеет беспрецедентно малую экранную память, что надо конечно отнести к гениальности К. Синклера. Практическим результатом, вытекающим из этого стало то, что это основное ограничение не сдерживало для "Спектрума" конверсию программ, выполненных для компьютеров других моделей. Да, есть ещё ограничения по объему памяти, но их преодолеть проще. Они решаются за счет объема данных. Упрощается музыка, упрощается графика, но логика и динамика игры сохраняются. Нормальная ситуация на рынке программного обеспечения, когда для "Спектрума" есть практически все, что есть для "Атари", "Коммодора", "Амстрада", "MSX" и других машин, но не наоборот. Только сейчас, в последние годы 32-разрядные компьютеры типа ATARI ST превосходят "Спектрум" в отношении скорость/объем экранной памяти. Из этого простого соотношения по-видимому и родился феномен "Спектрума")

УПРАВЛЕНИЕ ИГРОЙ

Отсутствие в стандартной версии компьютера встроенного порта джойстика означает, что игра должна управляться либо от клавиатуры, либо от джойстика произвольного типа. Поскольку на клавиатуре очень трудно играть, управляя объектом сразу по четырем

направлениям и при этом ещё стреляя, то стратегия управления игрой должна быть очень тщательно продумана, а динамика игры должна быть оптимально сбалансирована с динамикой управления. Помните, что метод управления оказывает решающее влияние на притягательность программы.

ПАМЯТЬ

В "Спектруме" есть области памяти, в которых программы работают с разной скоростью. Наиболее "быстрая" область нижние 16К ОЗУ (Это справедливо для машин с раздельным полем памяти, см. Техническое Приложение, вып. 1 - прим "ИНФОРКОМА"). Поэтому наиболее рационально процедуры, от которых требуется высокая скорость работы, размещать именно здесь. Здесь они работают на 30% быстрее. Здесь же желательно хранить и графические образы объектов, которые часто нужны в работе на экране. А прочие данные можно хранить и в верхних областях. Вы даже можете затереть область системных переменных, если в своей программе полностью перехватите управление прерываниями для опроса клавиатуры, отключившись от стандартных процедур ПЗУ. Лично я считаю непрактичным создавать программу с длиной объектного кода более 15К, а остальная часть памяти в моих программах начинена графикой, данными и некоторыми инструментальными средствами.

Одним из своих инструментальных средств я готов поделиться. Эта маленькая программа написана на БЕЙСИКе и служит для удаления строк из программы. Я вообще очень широко использую БЕЙСИК во время подготовки данных, текста и графики для своих игр.

Слейте эту программу с какой-либо своей с помощью команды MERGE. Надо только чтобы Ваша программа начиналась с номера строки 10 или больше. Запустите полученную программу - RUN. А теперь по запросу можете ввести номера начальной и конечной строк, которые Вы хотите удалить. Только не удаляйте первую строку, а то тот машинный код, который в ней записан погибнет и компьютер "зависнет" или "сбросится".

Чтобы сделать себе такую программу, Вам надо:

Набрать приведённый ниже текст, обратив особое внимание на том, что в строке 1 после REM должно идти по крайней мере 18 каких-нибудь символов. Здесь вместо них разместится машинный код процедуры удаления. Постарайтесь также набрать строку DATA без ошибок.

Выгрузить программу (SAVE) на тот случай, если есть какая-нибудь ошибка.

Дать команду GO TO 200, по которой машинный код будет перегружен в строку 1 и установлен за оператором REM.

Запустите программу и по запросу введите номера строк, предназначенных на удаление FROM 200 TO 999. Теперь, когда машинный код перегружен в 1-ую строку, они больше не нужны.

Теперь программа готова для работы. Выгрузите её на ленту в окончательном виде и можете сливать со своей программой, с которой Вы работаете через MERGE.

РАСПЕЧАТКА ПРОГРАММЫ

```
1 REM 123451234512345123
2 LET ADD=5+PEEK 23635+(256*PEEK 23636)
3 INPUT "FROM";FF;" TO";TO
4 LET TO=TO+1
5 LET X=INT (FF/256): POKE ADD+2,X: POKE ADD+1, FF-(X*256)
6 LET X=INT (TO/256): POKE ADD+9,X: POKE ADD+8, TO-(X*256)
7 RANDOMIZE USR ADD: STOP

200 REM MACHINE CODE LOADER
210 DATA 33,0,0,205,110,25,229,33,01,00,205,110,25,209,205,229,25,201
215 LET ADD=5+PEEK 23635+(256*PEEK 23636)
220 FOR A=0 TO 17
230 READ B
```

```
240 POKE ADD+A, B  
250 NEXT A
```

Для тех же, кто любит сам разбираться с программами, в следующем выпуске мы дадим небольшой комментарий к машинному коду этой процедуры.

(Продолжение следует)

МАЛЕНЬКИЕ ХИТРОСТИ

Экономия памяти - проблема, с которой редко сталкиваются начинающие программисты, но по мере того, как начинает расти их уровень подготовки, начинают расти и размеры программ и ужесточаются требования по экономному расходованию байтов.

48 килобайт - много это или мало? Для начинающих, конечно очень много, но посмотрите фирменные программы, выполненные лучшими программистами мира. Это сотни экранов графики, десятки страниц текста. Если бы только знать, какими трудами это дается!

Мы начнем с малого. У английских программистов есть хорошая поговорка - "Позаботься о своих байтах, а килобайты о себе сами позаботятся."

Итак, из 48 К "Спектрума" около 7К уходят на память экрана, а есть еще область системных переменных, область машинного и программного стеков, а также стека калькулятора, область графики пользователя UDG. Одним словом реально для написания программы и размещения данных Вы располагаете объемом чуть больше 40 К.

Это по-прежнему немало, но представьте себе, что Вы пишете базу данных или текстовый редактор, в которых Вы отводите как можно больше памяти под хранение информации, скажем 30К, тогда на Вашу программу остаются крохи. То же самое, если Вы пишете игру и хотите иметь в ней скажем 20 игровых экранов (как в программе MANIC MINER). Даже если Вы скомпрессируете каждый экран до 1.5К, все равно графика займет порядка 30К.

И это еще не все. Дело в том, что при программировании на БЕЙСИКе информация хранится как бы дважды. Во-первых, текст вашей программы находится в области, которая начинается с адреса, на который указывает системная переменная PROG (23635,23636). Обычно, если никакая периферия не подключена, то эта область начинается с адреса 23755. Во-вторых, значения всех переменных и, соответственно и массивов, которые Вы используете в своей программе, хранятся в области переменных, которая начинается с адреса, на который указывает системная переменная VARS (23627,33628). Адрес, на который указывает VARS - величина переменная, в отличие от PROG и зависит от размера текста программы. Это естественно, ведь область переменных расположена после программной области и чем длиннее вторая, тем выше адрес, с которого начинается первая.

Как определить адрес начала области, зная содержимое системной переменной Вы наверное знаете:

```
LET ad=PEEK 23627 +256*PEEK 33628
```

Экономия памяти должна проводиться грамотно. При этом не должна нарушаться удобочитаемость программ. Те приемы, которые мы здесь дадим, помогут Вам прояснить и некоторые туманные моменты, ставящие в тупик начинающих, когда они прорабатывают листинги БЕЙСИК-программ, написанных другими авторами. Наиболее частый вопрос, который мы встречаем из этой области:

"Скажите, почему в БЕЙСИК-загрузчиках программ иногда встречается запись типа LET a=VAL"15", ведь гораздо проще и короче написать LET a=15?"

С этого мы и начнем.

Числа

Числа являются самыми большими расточителями памяти в "Спектруме". Если Вы думаете, что число .134 занимает три байта, а число .13 - два байта, то Вы глубоко заблуждаетесь. Они хранятся в памяти в пятибайтной (интегральной) форме, но и это еще не все затраты. Давайте посмотрим как число .134 записано в программной строке. Напишите:

```
999 LET a=.134:LET p=PEEK 23635 + 256*PEEK 23636: FOR f=p TO p+17: PRINT f,PEEK f : NEXT f
```

Теперь дайте команду RUN и Вы увидите на экране две колонки цифр. Левая показывает адреса, в которых хранится первый оператор строки 10 LET a=.134:, а правая

показывает код символа, содержащегося в каждом адресе.

Адрес	Код	Значение
23755	3	Номер строки:
23756	231	$3+231*356=999$
23757	78	Длина строки:
23758	0	$78+0*256-78$
23759	211	LET
23760	97	a
23761	61	=
23762	46	. (точка)
23763	49	1
23764	51	2
23765	52	3
23766	14	Код, означающий что за ним следует число, записанное в интегральной форме.
23767	126	Это и есть число 0.134 в 5-ти байтной (интегральной) форме.
23768	9	
23769	55	
23770	75	
23771	199	
23772	58	: (двоеточие)

Из всего этого вытекает, что числа расходуют на 6 байтов больше, чем это кажется на первый взгляд.

Несколько приемов

Может быть обойтись без чисел, чтобы не тратить на них память? Для ряда чисел это сделать можно путем применения вместо них функций. Например, вместо нуля можно использовать NOT PI. Вместо семи байтов здесь расходуется всего 2. PI - это число "пи", которое уже хранится в ПЗУ компьютера, а функция NOT дает 0 для любого аргумента, отличного от нуля.

Вместо 1 применяют функцию SGN PI, поскольку число "пи" всегда положительное, эта функция всегда дает 1. Экономия здесь также 2 байта вместо 7.

С той же эффективностью применяют INT PI вместо числа 3.

Другие числа получают путем комбинирования приемов, причем здесь есть множество возможных путей. Например, если Вами уже введена переменная L, равная 1 путем 10 LET L=SGN PI, то дальше Вы можете это использовать например так:

```
20 LET z=L+L
30 LET f = z + z
```

Эта запись "съедает" 22 байта. Ее традиционной альтернативой, расходующей 30 байтов является:

```
20 LET z=2
30 LET f=4
```

Другой способ сокращения расхода памяти для произвольных чисел - применение функции VAL, что позволяет экономить три байта на каждом ее применении. Эта функция заменяет строковую переменную ее значением. Поскольку в памяти компьютера под строковую переменную отводится только один байт на символ, а на число - 5 байтов, то это и является основным источником экономии.

Посмотрите сами как в памяти хранятся строки:

```
10 LET z= 2 и 10 LET z=VAL"2"
```

Другая "маленькая хитрость" связана с использованием функции CODE для получения целых чисел от 32 до 164. Функция CODE возвращает код символа, стоящего после нее. Так, LET s= CODE "1" сделает s равным сорока девяти, поскольку код единицы в системе ASCII равен 49. Возникающую при этом экономию можете теперь подсчитать сами.

Вы наверное знаете, что после 164-го кода в Синклер-компьютерах расположены уже не символы, а токены ключевых слов, по токену на каждый код. Прелесть в том, что даже такое длинное слово как RANDOMIZE занимает только один байт и выражается одним кодом - 249. Теперь было бы желательно использовать возможности функции CODE, чтобы получать числа от 165 до 255. Действительно, запись типа LET s=CODE "RANDOMIZE" присвоит переменной s значение 249. Вот только вся беда, что сделать такую запись Вам не удастся.

Попробуйте, и у Вас ничего не выйдет, потому что когда Вы наберете LET s = " , у Вас на экране будет курсор L или C, а Вам надо курсор K, чтобы ввести ключевое слово RANDOMIZE. Набирать по буквам его нельзя, ведь это же токен. Здесь Вам поможет еще одна "маленькая" хитрость", связанная с использованием оператора THEN, поскольку после него появляется необходимый Вам курсор K. Делайте так: наберите

```
LET s = CODE "THEN RANDOMIZE",
```

а теперь сдвиньте курсор влево и с помощью DELETE удалите THEN.

FORUM

С мартовской почтой пришло еще несколько интересных писем от наших читателей, касающихся вопросов совместимости версий компьютера. Товарищ Бочанцев из Владивостока разобрался, почему на некоторых моделях "Спектрума" самодельной сборки не идет программа "BOMB JACK". Причина - в измененном в связи с русификацией ПЗУ компьютера.

Товарищ Панышин из Свердловска исследовал вопрос, почему на многих "нефирменных" компьютерах не идет программа "ARKANOID-I". Дисассемблирование программы позволило ему установить, что при чтении порта номер FF должны выполняться следующие условия:

1. Должен поступать код FF.
2. На 5-ый разряд этого порта необходимо подать синхросерию.

На своем компьютере (Ленинградский вариант) он произвел следующие доработки.

Входы 3,6,10,13 микросхем D37,D38 (КП11) должны быть в состоянии логической единицы, в том числе и незадействованные. Если джойстик подключается через инверторы, то входные контакты необходимо заземлить.

На 6-ую ногу микросхемы D38 (пятый разряд порта FF) он подал синхросерию, взяв ее с 8-ой ножки триггера D8.2 (сигнал INT). В этом случае не пропадает часть элементов экрана во время игры. От частоты поданной синхросерии зависит скорость игры.

Он высказал предположение, что по всей видимости машинный код программы проверяет некоторые особенности конструкции "фирменного" компьютера.

Мы связались со специалистами НТК "ПЛЮС" по этому вопросу и получили следующий комментарий. (Физику происходящих процессов мы опускаем, приводя только логику).

1. Не вполне корректно считать порт FF - портом Кемпстон-джойстика. Стандартно это порт 31, а в схеме "ЛЕНИНГРАД" просто дешифрация выполнена довольно грубо, хотя это мало где проявляется.

2. Наш читатель совершенно прав в том, что при опросе порта FF, как впрочем и вообще любого нечетного порта ввода фирменный компьютер должен выдавать FF, если физическое внешнее устройство или его интерфейс отсутствуют. Но дело в том, что это происходит не всегда! Есть отклонение, которое и было принято за синхросерию. Поставьте такой эксперимент:

10 PRINT IN 255: GO TO 10

Компьютер начнет распечатывать числа 255, 255, 255... , но вдруг в этой последовательности проскакивает код 56 (на фирменной машине), а затем опять 255, 255 и т.д. В чем же тут дело?

(Кстати, если внешнее устройство физически подключено, но не задействовано, например если компьютер имеет встроенный интерфейс для подключения джойстика, то будут выдаваться нули).

Код 56 в двоичной записи имеет вид 00111000. Это не что иное, как цветовой атрибут, соответствующий черному цвету символов на белом поле. Справа налево:

000 = 0 - цвет INK - черный;

111 = 7 - цвет PAPER - белый;

0 - BRIGHT - выключен;

0 - FLASH - выключен.

¹ данные, считанные с внешнего порта, к которому ничего не подключено?

Здесь Вам поможет описание, данное в 1-ом выпуске "РЕВЮ" на стр. 16 (левая колонка). В фирменном компьютере микросхема ULA занимается среди прочих вопросов и регенерацией (освежением) памяти дисплея. Она выполняет это по двухимпульсной схеме.

¹ В оригинале пропущена строка (Прим. OCR)

По первому регенерируются данные экрана, по второму - атрибуты. В момент этой регенерации ULA отключает процессор. После того, как она закончила эту операцию, процессор включается и "подхватывает" с шины данных то, что там выставила последний раз ULA. Вот и получается, что код, отличный от 255 иногда может прочитаться с внешнего незадействованного порта. Именно это и интерпретировал наш читатель как наличие непонятной синхросерии.

Конечно программа проверяла не весь байт, а только один бит (в данном случае пятый) - ей этого вполне достаточно.

Мы благодарим товарища Панышина от своего имени и от имени тысяч читателей, которым исследование, начатое им, поможет лучше понять работу своего компьютера.

Остается открытым вопрос зачем это было нужно проверять в программе. Гипотезу, что таким образом "фирмачи" хотели бороться с "самодельщиками" отбросим сразу. Поскольку нигде в мире кроме СССР самодельных "Синклеров" не бывает, а в те годы не было и у нас. Скорее всего это оригинальный метод выполнения синхронизации динамики перестроения экрана с динамикой управления игрой. Это может быть полезным, если например программисты хотели, чтобы угол отражения шарика от ракетки был бы не просто равен углу падения, но еще зависел бы и от скорости и направления движения ракетки в момент касания (как бы "подкрутка" шарика).

Вместе с тем, обращаем внимание на такую деталь. Конечно, разобраться с пределами совместимости очень важно, но многие наши читатели не специалисты в электронике. Не менее полезно вскрыв в программе такие "жучки" сделать просто их обход посредством нескольких операторов РОКЕ. В этом случае достижение одного могло бы сразу служить всем. И мы рассчитываем на то, что наши читатели будут писать о найденных решениях.

ПОСЛЕДНИЙ МЕСЯЦ

Апрель - последний месяц, в течение которого Вы еще можете сделать заказ на полный комплект ZX-РЕВЮ. Вы получите все выпуски, вышедшие до того, как Вы сделали заказ, и будете ежемесячно получать по новому выпуску в течение всего года.

Стоимость подписки на год (а на меньшие сроки мы заказы не принимаем) составляет для частных лиц 90 рублей.

Своих постоянных заказчиков "ИКФОРКОМ" также регулярно оповещает об имеющихся и вновь сделанных разработках.

Заказы направляйте по адресу 107241, Москва, Б-241, а/я 37, "ИНФОРКОМ". Здесь же укажите, когда и откуда был сделан перевод.

Для организаций при оплате по безналичному расчету действует повышенный тариф - 252 руб. за полный годовой комплект. Этот тариф не распространяется на школы, Дома Пионеров и т.п. организации, для них тариф - 90 р.

Оплату для организаций и индивидуальных заказчиков производить по адресу: Фрунзенское отделение Жилсоцбанка г.Москвы МФО 201412, р/с 500461778, кооператив "ИНФОРКОМ".

Игровое приложение

CONQUEST

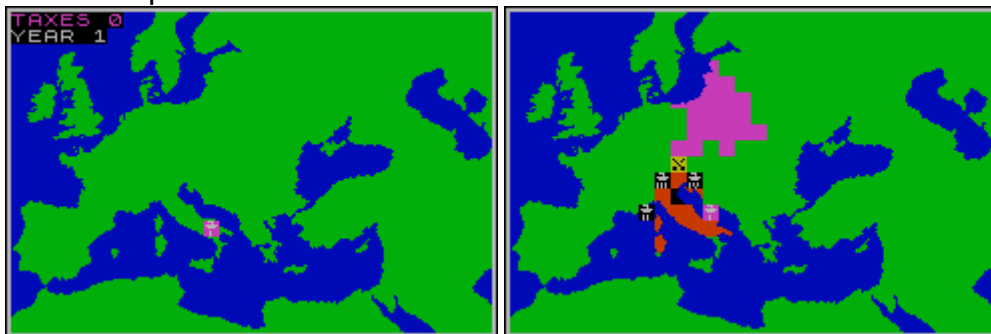


Введение.

"CONQUEST" - увлекательная, стратегическая игра, имеющая сложную систему правил. В программе 6 уровней сложности. Это позволяет наилучшим образом подготовить игрока к трудной стратегической борьбе на верхних уровнях. На каждом уровне добавляются некоторые новые правила, которые потребуют от Вас повышенного мастерства.

Игра построена по мотивам завоевательных походов древнего Рима. Кроме стратегического мышления она потребует от Вас и тактических навыков - умения принимать оптимальное решение в сложной ситуации. Особенностью игры является также то, что при передвижении своих легионов Вам надо будет хорошо помнить оперативную расстановку сил и, тем самым, программа тренирует зрительную память.

Мы рекомендуем хотя бы по разу сыграть на каждом уровне, прежде чем переходить к следующему. Не судите об игре по первым уровням. Они разработаны специально, чтобы Вы привыкли к основным правилам.



Правила уровня 1.

На экране изображена карта Средиземноморья. Она разбита на квадраты (20x30). Существуют два типа квадратов (полей) - суша и вода. Суша - это квадраты, на которых есть хотя бы немного зеленого цвета, по ним разрешено перемещение. Вода - это поля, закрашенные только в синий цвет, движение по ним невозможно.

В игре есть возможность наложения сетки на игровое поле и возможность выделения области "воды" для координации ваших перемещений.

Игра начинается с выбора места для будущей столицы (этот квадрат закрашивается черным цветом).

Столица может содержать одну армию (или легион), который может перемещаться в четырех направлениях курсорными клавишами 5,6,7,8. Это 1-й легион. Им командует император, он окрашен в пурпурный цвет. Когда он перемещается на зеленое поле, он завоевывает его и поле становится красным. Налоги с пяти завоеванных территорий позволяют создать новый легион - номер 2. Он возникает в столице, а командует им один из имперских генералов. Легион окрашен в черный цвет и имеет римскую цифру, обозначающую его номер. По мере захвата очередных пяти полей создаются все новые и

новые легионы. По мере утраты территорий, легионы также могут исчезать, ибо их не на что содержать.

Через несколько лет, таким образом, империя начинает приобретать какую-либо форму.

Обычные легионы (II,III,IV и т.д.) могут перемещаться только на одно поле за один ход, в то время как императорский легион (I) может перемещаться на неограниченное количество полей, но только в пределах империи.

Если нажать 0 вместо курсорной клавиши, легион остается на месте. Максимально возможное количество легионов - 16.

Завоевание зеленых полей всегда бывает успешным, но варвары могут совершать набеги на границы империи и отвоевывать отдельные поля, не защищенные в данный момент гарнизоном.

По мере развития игры, эти атаки становятся все более частыми. Направления атак варваров - такие же, как и направления движения легионов. (Они не могут нападать по диагонали).

Цель игры - завоевать 100 полей и как можно быстрее. После окончания игры рассчитывается Ваш фактор мастерства. При этом учитывается затраченное время, уровень сложности и сложность исходной позиции (местоположение столицы). Достигнутый Вами рейтинг сравнивается с достижениями выдающихся полководцев, такими как Александр Македонский, Наполеон и др. Таковы правила игры на первом уровне.

После просмотра первой части инструкции, Вам предлагается нажать:

"R" - повторить инструкцию.

"L" - загрузить основную программу.

"C" - продолжить просмотр инструкции.

Уровень 2.

Пройдя успешно уровень 1, Вы, наверное, поняли значение таких стратегических факторов, как центральное положение столицы империи, расположение линии легионов по границе империи, а также научились использовать естественные условия - моря и края игрового поля в качестве границ.

На уровне 2 прибавляются некоторые суровые реальности войны. Легионы по-прежнему непобедимы, но их командиры могут погибать в боях (примерно 10% вероятности).

Перед каждым очередным ходом Вы можете запросить послужной список для каждого из генералов. Это выполняется клавишей "R".

Если генерал погиб, то это никак не отражается на легионе, только новому офицеру, взявшему командование на себя, присваивается нулевой боевой опыт. Если же погибнет император, то императором провозгласит себя наиболее опытный из генералов. Обычно другие генералы резко против этого возражают, в результате чего начинается гражданская война. В этом случае примерно половина легионов может отколоться и пойти за мятежными генералами (окрашиваются в зеленый цвет). В этом состоянии они неподвижны, привести их в повиновение можно только победив легионом, оставшимся лояльным, если у его командира послужной список больше или равен списку командира мятежного легиона.

Если мятежный легион побежден, то его командира казнят, легион становится лояльным, а во главе его становится офицер, которому присваивается нулевой послужной список. Если же мятежный легион окажется сильнее, то лояльный легион погибает и должен быть сформирован заново в столице империи. Это единственный случай, когда легион полностью может погибнуть.

Гражданская война может вспыхнуть также, если какой-либо из генералов переживет 10 кампаний и решит, что он достаточно велик, чтобы сместить императора. В этом случае император может быть и жив.

Надо регулярно проводить проверку послужных списков генералов. Разумеется, боевой опыт императора может быть выше 10 кампаний без возникновения гражданской войны.

Мятежные легионы, хоть и не могут перемещаться, т.е. не повинуются указаниям, все же свою службу по охране границ от варваров выполняют. Это правила второго уровня. Далее предлагается нажать R,L,C (см. выше).

Уровень 3.

Играя на уровне 2, Вы вероятно научились вовремя отзываться генералов, когда их рейтинг достиг 9 и экономно использовать императора, чтобы не вызвать гражданской войны.

На третьем уровне добавлена еще одна возможная причина гражданской войны.

Если император в течение нескольких лет воздерживается от боев, он может стать непопулярным в армии и пасть жертвой покушения (assasination).

Далее "R", "L" или "C".

Уровень 4.

Здесь добавляются враждебные империи, их может быть три. Они окрашены в пурпурный, желтый и голубой цвет. Впервые они появляются через несколько лет после основания Вашей столицы и начинают развиваться за счет захвата территорий варваров. Если они входят в контакт с Вашей территорией, то могут захватить участки, не занятые гарнизонами. Враждебные империи не могут захватывать участки, на которых стоят лояльные императору легионы или крепости, но во время гражданской войны они могут захватить любые крепости или победить мятежные легионы. В этом смысле они более опасны, чем варвары.

Крепость может быть построена на месте легиона вместо его хода. Это делается нажатием клавиши F. При этом рейтинг командира легиона повышается на 1. Это может стать причиной гражданской войны. Если крепость захвачена противником, соответственно изменяется цвет фона поля. Всего в игре может быть построено до 10 крепостей. Далее "R", "L" или "C".

Уровень 5.

Здесь появляются эпидемии (PLAGUE). Они возникают примерно каждые 10 лет и уничтожают все на пути своего распространения. Легионы, охваченные эпидемией уничтожаются и вместо них в столице формируются новые. Это еще одна возможность легиону полностью исчезнуть.

Таковы правила базовой игры. Далее "R", "L" и "C".

Правила усложненной игры.

Здесь остаются в силе правила уровня 5. Добавляется возможность постройки морских портов. Этим обеспечивается возможность быстрой переброски легионов морем, необходимое при эпидемии или во время гражданской войны. Порты строятся так же, как и крепости. Клавиша - "P". Возможна постройка 20 портов, но для содержания каждого необходимы налоги с 10 клеток, они вычитаются из общей суммы. Можно построить дополнительные крепости (до 50), для каждой необходимы налоги с двух клеток. Помните, что для содержания 16 легионов вам необходимы 75 единиц налогов (первый легион содержится за счет столицы).

В начале каждого года изображается размер налогов. При этом расходы на порты и крепости уже вычтены, так что по его величине Вы можете судить о количестве доступных легионов.

Когда легион вступает на поле, отмеченное как порт, он погружается на корабль (корабль имеет тот же символ, что и порт, но окрашен в цвет легиона). Теперь он имеет неограниченные возможности перемещения в пределах открытого моря. Движение по морю прерывается достижением квадрата с участком суши. Если на нем построен порт, то движение по морю может быть продолжено.

Так, можно делать каналы на суше, если устанавливать порты в одну линию. Порт можно построить на любом участке суши. Это уже Ваша задача, как Вы будете обеспечивать выход из порта в море.

Это все правила игры.

Нажмите "L", а затем любую клавишу и загрузите основную программу.

После ее загрузки появится вопрос, не желаете ли Вы продолжить начатую ранее игру? Да - "Y", нет - "N".

Если "Да", то запустите ленту, затем нажмите любую клавишу.

Если "Нет", то появится запрос уровня сложности (от 1 до 5).

Введите уровень. На экране появится краткая сводка правил для данного уровня.

После этого появится меню:

S - запись состояния игры на ленту;

G - наложение сетки на карту;

O - изображение участков открытого моря;

R - просмотр послужного списка генералов;

F - построить крепость;

D - показать крепости;

5 - движение на запад;

6 - движение на юг;

7 - движение на север;

8 - движение на восток;

0 - без движения.

Далее идет запрос, нужна ли вам распечатка игры на принтере (Y/N)?

Затем идет ввод координат столицы. Сначала по вертикали (от 1 до 20), затем по горизонтали (от 1 до 30). И запрос - согласны ли Вы с этим ("IS THIS O.K?"). Ответ - "Y/N".

Как самостоятельно внести изменения в программу "MANIC MINER"



Одним из распространенных хобби в среде любителей персональных компьютеров является внесение изменений в готовую фирменную программу с целью настроить какие-либо игровые параметры по своему вкусу. Как правило это делается с помощью операторов POKE. Для тех, кто не знает как это делается, мы рассмотрим достаточно подробно технологию адаптации одной из популярнейших программ - MANIC MINER.

I. Порядок внесения изменений в программу.

Самый простой метод состоит в использовании оператора "MERGE". Для этого наберите "MERGE" или MERGE "MANIC MINER". Это позволит загрузить первую часть программы (но не стартовать). В этом месте надо остановить магнитофон.

Введите требуемые команды POKE, оформив их в качестве строки с номером 25, а затем запустите программу (RUN), например:

```
25 POKE 33885,7:POKE 35136,0:POKE 36924,0:POKE 35925,0
```

Единственная проблема состоит в том, что каждый раз, загружая программу, Вам предстоит переделывать загрузчик.

Вы можете переделанный таким образом загрузчик выгрузить на ленту (SAVE). Теперь его можно использовать совместно с исходной лентой. Сначала загружаете модифицированный загрузчик, а затем заменяете кассету в магнитофоне на исходную.

Если Вы хотите скопировать оставшиеся блоки программы на ту же ленту, где уже размешен новый загрузчик, то выполняйте следующие шаги:

1. LOAD "" CODE
SAVE "MMM" CODE 22784,256
2. CLEAR 30000
LOAD "" CODE
SAVE "MML" CODE 32768,33768

II. Дополнительные возможности.

Нижеперечисленные команды POKE позволят Вам изменить отдельные аспекты программы.

1. Выбор произвольного экрана.

```
POKE 33885,7
```

Подробности см. в разделе III.

2. Увеличение количества "жизней".

```
POKE 34269,K
```

Сколько попыток на прохождение игры имеет Ваш герой определяет параметр K.

3. "Бессмертие". Это другой метод продления жизни.

```
POKE 35136,0
```

После гибели Вы возвращаетесь к началу этого же самого экрана, при этом количество оставшихся "жизней" не изменяется. Для возврата к титульному экрану нажмите "BREAK".

4. "Последний экран".

```
POKE 36916,0      POKE 36924,0  
POKE 36918,0      POKE 36925,0  
POKE 36918,0
```

После того, как Вы проходите последний экран, происходит "нечто", но оно не может произойти в режиме демонстрации или если ранее был произведен выбор экрана.

Команды POKE от 36916 до 36918 позволяют этому событию происходить в демонстрационном режиме, а команды POKE 36934 и POKE 36925 - после выбора произвольного экрана.

5. Подача воздуха.

```
POKE 34795,0      POKE 34798,0
```

POKE 34796,0 POKE 34799,0
POKE 34797,0 POKE 34600,0

Этими командами устраняется утечка воздуха из пещер, что дает Вам возможность потратить больше времени на завершение экрана.

6. Изменение титров.

Высшей степенью адаптации программы является изменение титров бегущей строки в начале игры.

40224 - 40234 ...

40240 - 40259 ...

40288 - 40447 ...

III. Выбор экрана с клавиатуры.

Разработчики программы MANIC MINER оставили в ней интересный "жучок", который позволяет "своим" людям сразу войти в любой экран.

Нажмите "ENTER", запуская игру и, когда появится первый экран, наберите 6031769. В нижнем левом углу экрана появится изображение ботинка.

Чтобы выбрать какой-либо конкретный экран, нажмите (одновременно) указанные ниже клавиши, а затем (не отпуская их) нажмите клавишу "6". Для вызова первого экрана можно сразу нажать "6".

Экран	Клавиши	Экран	Клавиши
2	1	12	121
3	2	13	34
4	12	14	134
5	3	15	234
6	13	16	1234
7	23	17	5
8	123	18	15
9	4	19	25
10	14	20	125
11	24		

С другой стороны, войти в режим выбора произвольного экрана можно и командой POKE 33885,7 см. раздел II.

ADVENTURE LESSONS

Продолжение. Начало см. стр.35.

УРОК 3

Осмотр и анализ обстановки

Итак, Вы научились перемещаться в игровом пространстве. Начните просто с прогулки. Походите в разные стороны, посмотрите вокруг. На каждом шагу давайте команду LOOK, смотрите внимательно. Если программа имеет графику, то на картинке может быть изображено что-то, не вошедшее в текстовое описание. Обязательно рисуйте карту. В принципе чем больше локаций доступно для Вас, тем лучше у Вас развязаны руки и тем больше шансов, что Вы найдете что-то нужное. Но в игре всегда основная часть проходов закрыта от любопытного путешественника. Двери можно пробовать открыть OPEN, хотя без ключа это редко удастся.

Очень часто какие-то предметы загораживают нужный проход. Проверяйте команды MOVE, SHIFT, SLIDE. Например:

MOVE CHEST - отодвинуть сундук;

SHIFT MIRROR - сдвинуть зеркало;

SLIDE PANEL - сдвинуть панель.

В абсолютном большинстве игр Вы с помощью этих команд найдете множество новых проходов, комнат, этажей, подвалов и чердаков.

УРОК 4. РАБОТА С ОБЪЕКТАМИ.

Правильная работа с объектами, которые Вы можете встретить в своем путешествии, позволит Вам значительно сэкономить время на исследование и получить больше удовольствия от игры.

Обычно программа буквально насыщена всевозможными предметами. Что это может быть заранее мы сказать не можем и здесь Вам придется воспользоваться англо-русским словарем, если Вы слабовато знаете английский, но те действия, которые с предметами можно выполнить - отнюдь не бесконечны и осветить их вполне в наших силах.

Первое, что Вы можете сделать с предметов - это взять его (TAKE или GET). Например TAKE LASER - взять лазер. Часто встречающаяся ошибка - забывают взять предмет и пытаются его как-то использовать. Естественно, программа отвечает, что это она сделать не может.

Очень интересная команда TAKE ALL (взять все). Ее удобство не только в том, что Вы экономите время, не перечисляя предметы по одному. Второе достоинство ее в том, что Вы можете стать обладателем какого-либо предмета, взять который Вы бы сами и не догадались.

Например, в каменной пустыне Вы находите цветок (FLOWER). По команде TAKE FLOWER Вы возьмете только цветок. С другой стороны, после команды TAKE ALL, Вы можете с помощью команды INVENTORY обнаружить, что стали счастливым обладателем и цветка и небольшого камня (STONE). Для Вас, конечно, это сигнал о том, что где-то в программе он может пригодиться. Камнем можно что-либо разбить (BREAK) или в кого-нибудь его бросить (THROW). Правда, может оказаться, что камень драгоценный и это надо проверить командой EXAMINE STONE (проверить камень).

Итак, после того, как предмет взят, надо обязательно выполнить команду EXAMINE. Она позволит Вам узнать, например, что на рукоятке меча имеется какая-то надпись, что в ряду кнопок на пульте есть одна красная, что в рукоятке лазерного пистолета есть пустое отверстие и многое многое другое. Если программа не понимает слово EXAMINE (что бывает крайне редко), попробуйте TEST (испытать), на худой конец LOOK.

В программах фирмы "MELBOURNE HOUSE" применяют еще команду ANALIZE. Это почти то же, что и EXAMINE, но она позволяет проверить предмет на расстоянии, не вступая с ним в прямой контакт. Это может быть полезным в условиях, когда взять предмет в руки или хотя бы войти в комнату, в которой он расположен смертельно опасно. Например, если в этой комнате находится охранник этого предмета.

Команда EXAMINE (EXAM) считается в этих играх основной. От того, что Вы узнаете, выполнив ее и от того, какие выводы Вы при этом сделаете, зависит успех всей игры.

А как быть, если объект нельзя взять в руки, например, стол, комод, комнату. Для них тоже можно применять команду EXAM, но есть еще полезная команда SEARCH (обыскать). Например, SEARCH ROOM - обыскать комнату.

Противоположностью командам TAKE и GET являются команды DROP (бросить) и PUT (положить). Во-первых, просто физически Вы можете не унести все предметы, которые собрали, тогда что-то надо выложить (PUT). Так же, как и команда TAKE ALL, интересна команда DROP ALL (бросить все). Периодически ее можно выполнять, после чего стоит сделать LOOK, чтобы посмотреть, что из этого вышло. Дело в том, что среди предметов могут быть волшебные, при бросании которых могут происходить чудесные превращения. Помните, как в русской народной сказке: "... Бросил Иванушка гребень на дорогу и обернулся гребень лесом непроходимым ...". А зеркальце обернулось мостом через реку.

После DROP могут быть и иные превращения. Какой-то предмет может разбиться и открыть Вам нечто совсем иное.

Команда PUT позволяет выкладывать предметы в те места, в которых им положено быть. Например, если Вам надо сварить волшебный эликсир, то Вы должны положить его компоненты скажем в горшок (PUT ASH INTO POT - положить пепел в горшок), а затем поставить горшок на огонь (PUT POT INTO FIRE).

Иногда при этом возникают и неправильные лексические конструкции, но они работают: PUT WATER INTO BOTTLE - положить(?) воду в бутылку. Впрочем работает и GET WATER INTO BOTTLE - набрать воду в бутылку.

Кстати, это пример того, что иногда Вы не можете взять какой-либо объект (например, воду), если у Вас нет бутылки (BOTTLE), ведра (BUCKET), чашки (CUP) и т.п. Более изящная команда для наполнения емкостей - FILL (например, FILL BOTTLE WITH WATER - наполнить бутылку водой).

Однако вернемся к тому, что Вы взяли (TAKE) некий предмет и хорошо его рассмотрели (EXAM). Следующая Ваша задача научиться его применять. Применять предмет, конечно лучше всего по назначению, а также в соответствии с той информацией, которую Вы получили по команде EXAM. Поскольку предметов может быть очень много, то и способов их применения немало. Здесь Вам поможет логическое мышление и житейский опыт. На основные же способы мы укажем.

1. Еда, пища, напитки. Их можно есть (EAT) или пить (DRINK) например:

EAT FOOD - съесть пищу

DRINK MILK - выпить молоко

2. Предметы одежды, головные уборы и т.п. Их можно одеть на себя и в дальнейшем носить.

WEAR COAT - надеть пальто

WEAR HAT - надеть шляпу.

Команду WEAR можно применять не только к одежде, но и к некоторым видам оружия и к другим предметам, которые можно носить на одежде (значки, медальоны и др.)

WEAR SWORD - перепоясаться мечом

WEAR BADGE - приколоть значок

3. Оружие.

В зависимости от вида оружия могут применяться и разные команды.

Ножом (KNIFE) или топором (AXE) можно резать или рубить (CUT).

Саблю или меч надо обнажить, прежде чем ринуться в атаку на врага: DRAW SWORD.

При стрельбе из огнестрельного оружия применяют команды FIRE и SHOOT. И та и другая имеют значение - "стрелять", но разница есть.

Если после FIRE стоит дополнение, которое указывает из чего стрелять, то после SHOOT стоит дополнение, указывающее во что стрелять.

FIRE LASER - выстрелить из лазерного пистолета.

SHOOT WOLF - выстрелить в волка.

Кстати, если в программе участвует лазерное оружие, то стрельба из него обозначается более простой командой BLAST:

BLAST DOOR - расстрелять дверь из лазерного оружия.

Иногда, прежде чем оружие применять, его нужно зарядить (LOAD), например LOAD RIFLE - зарядить ружье. Правда, если LOAD используется для загрузки отложенной программы, то зарядить оружие можно командой CHARGE.

4. Двери, окна, сундуки и др. Все эти предметы характерны тем, что их можно открыть (OPEN) и закрыть (CLOSE).

OPEN CHEST - открыть сундук.

CLOSE WINDOW - закрыть окно.

Поскольку эти объекты могут иметь замок, то открыть их может оказаться непросто, надо сначала открыть замок (UNLOCK):

UNLOCK CHEST - отпереть сундук

LOCK DOOR - запереть дверь.

Правда, для того, чтобы отпереть что-либо, надо иметь еще и ключ (KEY). Если он у Вас есть, то команда может пройти, а если нет, то только в том случае, когда объект закрыт на ручную задвижку. Невредно сделать EXAM LOCK.

Правда, во многих программах здесь бывает маленькая хитрость. По результатам EXAM Вам могут сообщить, что замок, задвижка или дверные петли заржавели. В этом случае их надо смазать.

Естественный выход - найти масло (OIL). Оно может быть где-то лежит само по себе, а может быть для этого вам потребуется найти масленку (OIL CAN) или вскрыть масляную лампу (лампаду) - OIL LAMP.

Не забывайте также и о возможности открывать закрытые объекты силой или с помощью оружия, например:

BREAK LOCK - сломать замок.

Правда, срабатывает это редко.

5. Ключи.

С ними все просто. Обычно если Вы обладаете нужным ключом (KEY), то дверь открывается просто (OPEN DOOR), но иногда ключ надо вставить (INSERT KEY) и/или повернуть (TURN KEY). На необходимость этих действий может указывать сообщение после EXAM DOOR, что дверь имеет отверстие для ключа (KEY HOLE).

6. Отверстие, углубление. Если проверенный Вами объект имеет отверстие или углубление, то можно предположить, что туда надо что-то вставить (INSERT). Это может быть ключ (KEY). Для лазерного оружия может быть необходима батарейка (BATTERY). Для двигателей и других технических устройств иногда надо вставить предохранитель (FUSE). В играх с фантастическим сюжетом, особенно если имеете дело с инопланетной техникой, может быть необходимым вставить что угодно и куда угодно. Все надо проверять. Если в программе Вам встретилась монета (COIN), то теперь ищите щель (SLOT), в которую эту монету можно вставить (INSERT) для того, чтобы привести в действие какую-то автоматику. Правда деньгами можно и расплачиваться (PAY) или их дарить (GIVE), но об этом мы поговорим позже, когда будем обсуждать взаимоотношения с другими персонажами.

7. Кнопки, тумблеры, клавиши и т.д.

Самое лучшее, что можно сделать с кнопкой - нажать ее (PRESS BUTTON), правда можно при этом погибнуть, если это, скажем система автоликвидации космического корабля. Кроме команды PRESS для этой цели может применяться иногда команда PUSH - толкнуть. Все это относится и к клавишам (KEY). Тумблеры обычно переключаются командой TURN. В некоторых случаях для достижения успеха достаточно до чего-то только дотронуться (TOUCH). Это может быть какой-то выступ в стене, элемент орнамента у картины, металлическая пластинка на двери и т.п.

Вообще старайтесь с незнакомыми предметами почаще проверять команды PUSH, PULL, PRESS, TOUCH. Множество потайных ходов скрываются в программах и остаются неисследованными потому, что Вы не догадались прикоснуться к металлической шишечке у кровати или, скажем, не потянули (PULL) свисающую веревку или шнурок.

8. Если у Вас есть лопата (SHOVEL), то естественно желание где-то что-то раскопать (DIG). Пробуйте копать в разных местах. Иногда удастся копать ножом, кинжалом. Возможен такой диалог между Вами и программой:

```
>DIG (копать)
-DIG WITH WHAT? (чем копать?)
>DIG WITH KNIFE (копать ножом)
-O.K.
```

Продолжение в следующем номере.

CENTI

Для тех, кто имеет желание и возможность самостоятельно заниматься набором программ, мы сегодня даем распечатку программы CENTI. Это разновидность одной из первых компьютерных программ CENTIPEDE.



Наберите программу, затем запустите ее (RUN), после чего программа выдаст вам номера строк, в которых при наборе были сделаны ошибки. Исправив их, отгрузите программу на ленту

SAVE "CENTI" CODE 60000, 2688.

Полученный код загружайте и запускайте в игру следующим БЕЙСИК - загрузчиком:

10 CLEAR 59999: LOAD ""CODE: RANDOMIZE USR 60000

Если хотите поэкспериментировать с программой, то вот несколько POKES.

POKE 61056,n – количество "жизней".

POKE 60511,n - количество грибов. Чем меньше n - тем больше грибов.

POKE 63125,n - количество пауков.

```
5 CLEAR 59999: LET add=60000
10 FOR f=100 TO 510 STEP 10
15 PRINT AT 0,0; f
20 READ a$
25 LET val=VAL a$(1 TO 4)
30 LET a=0
35 FOR g = 5 TO LEN a$ STEP 2
40 LET num=(CODE a$(g)-48-(7 AND a$(g)>"9"))*16+(CODE a$(g+1)-48-(7 AND a$(g+1)>"9")): POKE
    add, num: LET a=a+num: LET add=add+1: NEXT g
45 IF a<>val THEN PRINT "ERROR!!!!!! line ";f: BEEP 2,10: STOP
50 NEXT f: STOP
100 DATA "7589F3AFD3FE32485CFD3 653473E02CD0116DD212DF1DD7708DD7
    709CDEAEBO6DB21D1F37ED72310FBDD5 607DD5E06010800CD6FECDD5609DD5E0 8011900CD6FEC3E"
110 DATA "9026C708DD7E11CDACF40 8DD7E12CDACF4CD1CEC3EE7DBFE2FE61
    F28F73E4708DD7E11CDACF408DD7E12C DACF43EF7DBFE1F380CCD01ECCD1CECC D13ECC37AEA1F38"
120 DATA "608304DD3611021F3804D D3611031F380ADD3612042101002271E
    B1F380ADD36120521D0072271EB3EEFD BFE1FD220EBE608C29FEADD361206218 8132271EBC39FEA"
130 DATA "7201CD31ECCD13ECCD25E CCDEAEBCFD213A5CCD3DEC214BEE06237
    ED72310FBDD5E021600011B01CD6FECDD5609DD5E08010D01CD6FEC010000AFD BFE2FE61F20050B"
140 DATA "786378B120F3DD5E06DD5 607010001CD6FEC01D0070B78B120FB3
    E7FDBFE1F380D3EFEDBFE1F3806FD213 A5CFBC93EBFDBFEE6102006CD1CECCD1 3ECDD7E11FE02CC"
150 DATA "968328F2CDA6ECCDA3F2C D49F3CD7EEF1E433E14CDABEDCD40F1C
    364EBFD213A5CCD8EF2E1DD3502C226E BCDEAEBCDD5607DD5E06EBDD5609DD5E0 8A7ED52DA7AEA19"
160 DATA "7492EBDD7308DD7209C37 AEA2100401101403600010018EDB0231
    30100033642EDB0C9CDEAEB21E0EF014 D017ED70B2378B120F8C9AFDBFE2FE61 F28F9C9AFDBFE2F"
170 DATA "5811E61F20F8C9113AF22 1F7ED015400EDB0C9112DF1217EEE010
    800EDB0C9011F012A785C545D2329291 92929291922785C0C3E20B920070E000 43E15B8C87CFE14"
180 DATA "77013E171E4230043E161 E04CDABED18D1DDE5C5EB0605DD219CE
    CDD5E00DD56013EFF3CB7ED5230FA195 0C11E42CDABED0CC542DD23DD2310E1C 1DDE1C91027E803"
190 DATA "656064000A000100FD213 AF2DD5605FD4E00FD4601783CCA63ED3
    E171E42CDABEDFD7E02FE80283CA7204 679A7200D04FD350278FE18207206101 86E0DCDE1EDBB28"
```

[illegible]

BEST GAMES

Сегодня мы даем несколько списков лучших игр по итогам 90-го года. Парад открывают два листа из журнала "YOUR SINCLAIR", в каждом по тридцать программ.

Зарубежные журналы подводят итоги в двух категориях. Первая категория - программное обеспечение, продаваемое за полную стоимость. Обычно цена на такую программу составляет от 10 до 15 фунтов стерлингов.

Вторая категория - бюджетное программное обеспечение. Здесь цена программы - примерно 3 ф. ст. Не стоит думать, что игры FULL PRICE чем-то лучше, чем BUDGET. Прямой зависимости здесь нет. Речь просто идет о разных способах распространения и рекламы. Различаются, конечно и тиражи.

FULL PRICE TOP 30

- 1 "Shadow Warriors" - OCEAN
- 2 "Turrican" - RAINBOW ARTS
- 3 "Chase HQ" - OCEAN
- 4 "Manchester United" - KRISALIS
- 5 "Robocop" - OCEAN
- 6 "World Cup Soccer'90" - VIRGIN
- 7 "Rainbow Island" - OCEAN
- 8 "Emlyn Hughes International Soccer Manager" - AUDIOGENIC
- 9 "Football Manager World Cup Edition" - ADDICTIVE
- 10 "Batman (The Movie)" - OCEAN
- 11 "Adidas Championship Football" OCEAN
- 12 "Fighter Bomber" - ACTIVISION
- 13 "Italy 1990 Winners Edition" - US GOLD
- 14 "Italy 1990" - US GOLD
- 15 "International 3D Tennis" - PALACE
- 16 "Gazza's Super Soccer" - EMPIRE
- 17 "Cricket Master" - CHALLENGE SOFTWARE
- 18 "Escape from Planet of the Robot Monsters" - TENGEN(DOMARK)
- 19 "Heroes of the Lance" - US GOLD
- 20 "World Cup'90 Compilation" - EMPIRE
- 21 "Turbo OutRun" - US GOLD
- 22 "Lords Of Chaos" - BLADE
- 23 "Castle Master" - DOMARK
- 24 "Dan Dare III" - VIRGIN
- 25 "Treble Champions" - CHALLENGE SOFTWARE
- 26 "Project Stealth Fighter" - MICROPROSE
- 27 "The Cycles" - ACCOLATE
- 28 "Ghouls and Ghosts" - US GOLD
- 29 "Hard Drivin" - DOMARK
- 30 "Operation Thunderbolt" - OCEAN

BUDGIES TOP 30

- 1 "Fantasy World Dizzy" - CODEMASTERS
- 2 "Pro Boxing" - CODEMASTERS
- 3 "Quattro Adventure" - CODEMASTERS
- 4 "Paperboy" - ENCORE
- 5 "A Question of Sport" - ENCORE

- 6 "Treasure Island Dizzy" - ENCORE
- 7 "Matchday II" - HIT SQUAD
- 8 "Rastan" - HIT SQUAD
- 9 "Hong Kong Phooey" - HI-TEC
- 10 "World Cup Challenge" - PLAYERS
- 11 "Indiana Jones and the Temple of Doom" - KIXX
- 12 "Rock Star Ate My Hamster" - CODEMASTERS
- 13 "Jack the Nipper II" - KIXX
- 14 "Wonderboy" - HIT SQUAD
- 15 "Roadblasters" - KIXX
- 16 "Yogi's Great Escape" - HI-TECH
- 17 "Ruff and Reddy" - HI-TECH
- 18 "Cauldron 1&2" - HI-TECH
- 19 "Daley Thompson's Olympic Challenge" - HIT SQUAD
- 20 "Big Trouble in Little China" - ALTERNATIVE
- 21 "The Munsters" - ALTERNATIVE
- 22 "Football Champions" - CULT
- 23 "Pro Golf" - ATLANTIC
- 24 "4x4 Off Road Racing" - KIXX
- 25 "Salamander" - HIT SQUAD
- 26 "Quattro Sports" - CODEMASTERS
- 27 "Vindicator" - THE HIT SQUAD
- 28 "Pro Tennis Simulation" - CODEMASTERS
- 29 "Daley Thompson's Supertest" - HIT SQUAD
- 30 "Prison Riot" - PLAYERS

Теперь два списка из журнала SINCLAIR USER.

FULL PRICE 20

- 1 "Shadow Warriors" - OCEAN
- 2 "Turrican" - RAINBOW ARTS
- 3 "Chase H. Q." - OCEAN
- 4 "Manchester United" - KRISALIS
- 5 "Robocop" - OCEAN
- 6 "Lords of Chaos" - BLADE
- 7 "Batman the Movie" - OCEAN
- 8 "Emlyn Hughes International Soccer" - AUDIOGENIC
- 9 "Rainbow Islands" - OCEAN
- 10 "Midnight Resistance" - OCEAN
- 11 "Italy 1990 Winners Edition" - US GOLD
- 12 "Football Man World Cup" - ADDICTIVE
- 13 "Bomber" - ACTIVISION
- 14 "Adidas Championship Football" - US GOLD
- 15 "Gazza's Super Soccer" - EMPIRE
- 16 "Battle of the Bulge" - CCS
- 17 "World Cup Soccer 90" - VIRGIN
- 18 "Escape from Planet of the Robot Monsters" - TENGEN(DOMARK)
- 19 "Heroes of the Lance" - US GOLD
- 20 "International 3D Tennis" - PALACE

BUDGET 10

- 1 "Rastan" - HIT SQUAD
- 2 "Quattro Adventure"-CODEMASTERS
- 3 "Fantazy World Dizzy" - CODEMASTERS
- 4 "Pro Boxing Manager" - CODEMASTERS
- 5 "Treasure Island Dizzy" - CODEMASTERS
- 6 "Match Day 2" - HIT SQUAD
- 7 "Paperboy" - ENCORE
- 8 "D.Thompson's Olympic Challenge" - HIT SQUAD
- 9 "Pro Golf" - ATLANTIC
- 10 "Indiana Jones and the Temple of Doom" - KIXX

И, в заключение, наша лучшая десятка, составленная по письмам читателей.

ZX-REVIEW TOP TEN

- 1 "Elite" - FIREBIRD
- 2 "Exolon"- HEWSON CONSULTANTS
- 3 "Saboteur 2" - DURELL
- 4 "Into the Eagles Nest" - PANDORA
- 5 "Saboteur 1" - DURELL
- 6 "Barbarian" - PALACE
- 7 "Spy Hunter" - US GOLD
- 8 "Down to Earth" - FIREBIRD
- 9 "Army Moves" - IMAGINE
- 10 "Pyjamarama" - MIKROGEN

ЭТОТ РАЗДЕЛ ВЕДЕТ НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС",
СПЕЦИАЛИЗИРУЮЩИЙСЯ В ВОПРОСАХ АППАРАТНОГО ОБЕСПЕЧЕНИЯ СИНКЛЕР-
СОВМЕСТИМЫХ КОМПЬЮТЕРОВ И СХЕМОТЕХНИЧЕСКИХ РЕШЕНИИ ПЕРИФЕРИЙНЫХ
УСТРОЙСТВ

127566, МОСКВА И-566, НТК "ПЛЮС"

ZX MODEM

Сегодня мы продолжаем разговор о ZX-модеме, начатый в прошлом выпуске, и приводим распечатку машинного кода программы, поддерживающей работу модема.

Перед началом ввода машинного кода Вам надо набрать программу БЕЙСИК-загрузчика:

```
1 CLEAR 62999: LET add=63000: DIM a(10)
2 DEF FN A(a$)=(CODE a$(1)-48-(7 AND a$(1)>"9"))*16+(CODE a$(2)-48-(7 AND a$(2)>"9"))
10 FOR f=101 TO 317
12 RESTORE f
14 PRINT AT 0,0; f
15 READ a$
16 LET sum=0
20 FOR i=1 TO 2
22 LET b$ = a$(2*i-1 TO 2*i)
24 LET a(i)=FN A(b$)
26 NEXT i
30 LET add=a(1)*256+a(2)
32 LET sum=a(1)+a(2)
40 FOR i = 3 TO 10
44 LET b$=a$(2*i-1 TO 2*i)
46 LET a(i)=FN A(b$)
48 LET sum=sum+a(i)
50 POKE add, a(i)
52 LET add=add+1
54 NEXT i
60 LET b$=a$(21 TO)
62 LET cs=FN A(b$)
64 LET cs1=sum-256*INT(sum/256)
66 IF cs<>cs1 THEN PRINT "ERROR!!!!!! line "; f:BEEP 2,10: STOP
68 NEXT f: STOP
```

Комментарии

Строка 10 - главный цикл ввода данных.

Строка 20 - ввод двух байтов адреса.

Строка 30 -определение адреса.

строка 40 - ввод 8-ми кодов.

Строка 60 - ввод контрольной суммы.

Приведенные ниже строки DATA напечатаны вразрядку. Это сделано для удобочитаемости данных. Вам же при наборе следует набирать их подряд. Например:

```
101 DATA "F618C3B6F8000054524166"
```

Никаких пробелов в строке DATA быть не должно. После того, как программа будет набрана, запустите ее (RUN). Все сделанные при наборе ошибки будут выявлены путем проверки контрольных сумм. Исправьте их и, когда убедитесь, что ошибок больше нет, отгрузите программу на ленту командой SAVE "modem" CODE 63000,1734

```
101 DATA "F618 C3 B6 F8 00 00 54 52 41 66"
102 DATA "F620 4E 53 4D 49 54 20 01 52 14"
103 DATA "F628 45 43 45 49 56 45 20 01 F0"
104 DATA "F630 00 20 43 6F 70 79 72 69 BC"
105 DATA "F638 67 68 74 20 20 44 2E 4A 6D"
```

```
106 DATA "F640 6F 76 61 6E 6F 76 69 63 9B"
107 DATA "F648 20 20 31 39 38 35 20 20 95"
108 DATA "F650 20 20 20 20 20 20 20 41 67"
109 DATA "F658 6C 6C 20 72 69 67 68 74 64"
110 DATA "F660 73 20 72 65 73 65 72 76 80"
```

111 DATA "F668 65 64 20 20 20 20 20 20 E7"
112 DATA "F670 20 20 20 84 8C 8C 20 88 0A"
113 DATA "F678 20 88 20 20 20 20 88 20 3E"
114 DATA "F680 88 20 8C 88 20 8C 8C 20 8A"
115 DATA "F688 84 8C 8C 20 88 20 88 20 8A"
116 DATA "F690 20 20 20 20 84 82 20 81 AD"
117 DATA "F698 89 20 20 20 20 20 8B 89 CB"
118 DATA "F6A0 8A 85 20 85 20 8A 20 8A 9E"
119 DATA "F6A8 85 8C 88 20 8B 89 8A 20 15"
120 DATA "F6B0 20 20 20 84 8E 8C 20 89 4D"
121 DATA "F6B8 81 88 20 20 20 20 8A 20 E1"
122 DATA "F6C0 8A 81 8C 89 20 8E 8C 82 92"
123 DATA "F6C8 85 8C 8C 20 8A 20 8A 20 CF"
124 DATA "F6D0 20 20 20 20 20 20 20 20 C6"
125 DATA "F6D8 20 20 20 20 20 20 20 20 CE"
126 DATA "F6E0 20 20 20 20 20 20 20 20 DE"
127 DATA "F6E8 20 20 20 20 20 20 20 20 DE"
128 DATA "F6F0 20 43 43 49 54 54 2D 56 00"
129 DATA "F6F8 32 34 20 73 74 61 6E 64 8E"
130 DATA "F700 61 72 64 20 20 20 20 20 CE"
131 DATA "F708 20 20 20 20 20 20 20 20 FF"
132 DATA "F710 20 4E 4F 20 73 74 6F 70 AA"
133 DATA "F718 20 62 69 74 73 20 3B 31 6D"
134 DATA "F720 20 73 74 61 72 74 20 62 E7"
135 DATA "F728 69 74 20 20 20 20 20 20 BC"
136 DATA "F730 20 50 61 72 69 74 79 3A FA"
137 DATA "F738 20 20 20 4F 44 44 3B 45 E6"
138 DATA "F740 56 45 4E 20 28 50 20 74 4C"
139 DATA "F748 6F 20 73 65 6C 65 63 74 4E"
140 DATA "F750 29 42 61 75 64 20 72 61 DF"
141 DATA "F758 74 65 3A 36 30 30 3B 31 64"
142 DATA "F760 32 30 30 20 28 42 20 74 07"
143 DATA "F768 6F 20 73 65 6C 65 63 74 6E"
144 DATA "F770 29 31 32 30 30 20 48 7A 35"
145 DATA "F778 20 48 49 47 48 20 62 69 9A"
146 DATA "F780 74 3B 32 34 30 30 20 48 54"
147 DATA "F788 7A 20 4C 4F 57 20 62 69 F6"
148 DATA "F790 74 20 20 20 20 20 20 20 DB"
149 DATA "F798 20 20 20 20 20 20 20 20 8F"
150 DATA "F7A0 20 20 20 20 20 20 20 20 97"
151 DATA "F7A8 20 20 20 20 20 20 20 20 9F"
152 DATA "F7B0 20 20 20 20 20 20 20 20 A7"
153 DATA "F7B8 20 20 20 20 20 20 20 20 50 DF"
154 DATA "F7C0 72 65 73 73 20 20 20 20 F4"
155 DATA "F7C8 20 20 20 20 20 20 20 20 BF"
156 DATA "F7D0 20 20 20 20 20 20 20 20 C7"
157 DATA "F7D8 20 20 20 20 20 20 20 20 CF"
158 DATA "F7E0 20 20 20 20 20 20 20 20 D7"
159 DATA "F7E8 20 20 20 20 20 20 20 20 DF"
160 DATA "F7F0 20 20 A3 20 66 6F 72 20 51"
161 DATA "F7F8 74 72 61 6E 73 6D 69 74 61"
162 DATA "F800 20 3B 20 52 20 66 6F 72 2C"
163 DATA "F808 20 72 65 63 65 69 76 65 03"
164 DATA "F810 20 20 28 54 72 61 6E 73 78"
165 DATA "F818 6D 69 74 20 78 2C 79 29 C0"
166 DATA "F820 20 3B 20 28 52 65 63 65 3A"
167 DATA "F828 69 76 65 20 78 29 20 20 65"
168 DATA "F830 20 20 20 20 20 20 20 20 28"
169 DATA "F838 20 20 20 20 20 20 20 20 30"
170 DATA "F840 20 20 20 20 20 20 20 20 38"
171 DATA "F848 20 20 20 20 20 20 20 20 40"
172 DATA "F850 20 78 20 3D 20 73 74 61 A5"
173 DATA "F858 72 74 20 61 64 64 72 65 56"
174 DATA "F860 73 73 20 20 20 20 20 20 FE"
175 DATA "F868 20 20 20 20 20 20 20 20 60"

176 DATA "F870 20 79 20 3D 20 6C 65 6E BD"
177 DATA "F878 67 68 74 20 20 20 20 20 53"
178 DATA "F880 20 20 20 20 20 20 20 20 78"
179 DATA "F888 20 20 20 20 20 20 20 20 80"
180 DATA "F890 20 20 20 20 20 20 20 20 88"
181 DATA "F898 20 20 20 20 20 20 20 20 90"
182 DATA "F8A0 20 20 20 20 20 20 20 20 98"
183 DATA "F8A8 20 20 20 20 20 20 20 20 A0"
184 DATA "F8B0 20 00 20 20 20 20 21 B2 1B"
185 DATA "F8B8 5C 3E 17 77 23 3E F6 77 A6"
186 DATA "F8C0 CD 81 FA AF 32 8C 5C 06 CF"
187 DATA "F8C8 16 3E 0D D7 10 FB 01 FE 02"
188 DATA "F8D0 FB ED 78 CB 67 28 E9 CB 36"
189 DATA "F8D8 5F 28 E5 CD 4B FB 20 EE 5D"
190 DATA "F8E0 C9 21 2A 59 7E FE 78 28 61"
191 DATA "F8E8 11 CD 13 F9 3E 78 CD 0D 5A"
192 DATA "F8F0 F9 3E 38 23 23 CD 0B F9 6E"
193 DATA "F8F8 AF C9 CD 3F F9 3E 38 CD B0"
194 DATA "F900 0D F9 3E 78 23 23 CD 0B D3"
195 DATA "F908 F9 AF C9 77 23 77 23 77 1D"
196 DATA "F910 23 77 C9 3E 02 32 8E FC 68"
197 DATA "F918 3E 04 32 A3 FC 3E 02 32 96"
198 DATA "F920 ED FB 3E F5 32 F4 FB 3E 93"
199 DATA "F928 81 32 00 FC 3E CD 32 FA 07"
200 DATA "F930 FB 32 06 FC 3E B9 32 0C 8D"
201 DATA "F938 FC 3E 96 32 CA F9 C9 3E FD"
202 DATA "F940 01 32 8E FC 3E 02 32 A3 0B"
203 DATA "F948 FC 3E 01 32 ED FB 3E 14 E8"
204 DATA "F950 32 F4 FB 3E 46 32 00 FC 1C"
205 DATA "F958 3E 3A 32 FA FB 32 06 FC 24"
206 DATA "F960 3E 7D 32 0C FC 3E 1E 32 DC"
207 DATA "F968 CA F9 C9 21 0A 59 7E FE ED"
208 DATA "F970 78 28 11 CD 95 F9 3E 78 2B"
209 DATA "F978 CD 0D F9 3E 38 23 23 CD CD"
210 DATA "F980 0B F9 AF C9 CD AF F9 3E A8"
211 DATA "F988 38 CD 0D F9 3E 78 23 23 88"
212 DATA "F990 CD 0B F9 AF C9 3E E4 32 26"
213 DATA "F998 71 FC 3E EC 32 74 FC 3E 08"
214 DATA "F9A0 EA 32 D6 FB 3E E2 32 DD B5"
215 DATA "F9A8 FB 3E 0A 32 C9 F9 C9 3E DF"
216 DATA "F9B0 E4 32 74 FC 3E EC 32 71 FC"
217 DATA "F9B8 FC 3E EA 32 DD FB 3E E2 FF"
218 DATA "F9C0 32 D6 FB 3E 09 32 C9 F9 F7"
219 DATA "F9C8 C9 09 1E 3E 02 CD 01 16 D5"
220 DATA "F9D0 11 31 F6 1A FE 00 28 08 49"
221 DATA "F9D8 FE 01 28 54 D7 13 18 F3 41"
222 DATA "F9E0 3A CA F9 FE 1E 28 05 CD EC"
223 DATA "F9E8 E1 F8 18 06 21 2A 59 CD 49"
224 DATA "F9F0 FA F8 3A C9 F9 FE 09 28 06"
225 DATA "F9F8 05 CD 6B F9 18 06 21 0A 70"
226 DATA "FA00 59 CD 84 F9 CD 87 FA FE E9"
227 DATA "FA08 74 28 17 FE 62 CC E1 F8 BA"
228 DATA "FA10 FE 70 CC 6B F9 FE 00 28 CE"
229 DATA "FA18 EB FE 72 28 0C FE 20 C8 87"
230 DATA "FA20 18 5F 11 1D F6 0E 00 18 DB"
231 DATA "FA28 AA 11 27 F6 0E 01 18 A3 C4"
232 DATA "FA30 21 00 00 CD 87 FA FE 0C A3"
233 DATA "FA38 28 47 FE 0D 28 27 F5 D7 C7"
234 DATA "FA40 F1 FE 2C 28 14 D6 30 FE 95"
235 DATA "FA48 0A F2 81 FA E5 D1 06 09 7E"
236 DATA "FA50 19 10 FD 16 00 5F 19 18 16"
237 DATA "FA58 DA CD E8 FA 79 FE 00 0E 60"
238 DATA "FA60 0A 28 CD 18 1C F5 11 80 13"
239 DATA "FA68 5A 06 20 3E 78 12 13 10 CD"
240 DATA "FA70 FC F1 79 FE 0A F5 CC F1 8A"

241 DATA "FA78 FA F1 CA 17 FC FE 01 28 61"
242 DATA "FA80 15 CD AF OD C3 CB F9 E5 84"
243 DATA "FA88 21 3B 5C CB 6E 28 FC 3A D1"
244 DATA "FA90 08 5C CB AE E1 C9 F3 06 0A"
245 DATA "FA98 00 CD 55 FB CD 64 FB C5 A0"
246 DATA "FAA0 3E 2C D7 AF CD BD FA 3E 4C"
247 DATA "FAA8 20 D7 3E 45 D7 3E 52 D7 5A"
248 DATA "FAB0 3E 52 D7 3E 4F D7 3E 52 05"
249 DATA "FAB8 D7 3E 20 D7 E1 11 01 00 B1"
250 DATA "FAC0 D5 11 0A 00 D5 11 64 00 F4"
251 DATA "FAC8 D5 11 E8 03 D5 11 10 27 B0"
252 DATA "FAD0 D5 06 05 AF D1 E5 ED 52 4E"
253 DATA "FAD8 FA E1 FA 3C 08 F1 08 18 FC"
254 DATA "FAE0 F4 C6 30 E1 D7 10 EC C9 41"
255 DATA "FAE8 7C 32 27 FC 7D 32 26 FC 84"
256 DATA "FAF0 C9 3A 27 FC 57 3A 26 FC C3"
257 DATA "FAF8 5F 19 7C 32 2D FC 7D 32 F0"
258 DATA "FB00 2C FC C9 DD 2A 0B 5C DD 37"
259 DATA "FB08 6E 04 DD 66 05 DD 5E 0C 04"
260 DATA "FB10 DD 56 0D C9 CD 03 FB CD AC"
261 DATA "FB18 E8 FA D5 E1 CD F1 FA 06 69"
262 DATA "FB20 00 5F DB FD E6 80 BB 20 93"
263 DATA "FB28 F6 10 F7 06 0A C3 17 FC 06"
264 DATA "FB30 CD 03 FB F3 06 0A CD 55 1B"
265 DATA "FB38 FB CD 64 FB C5 E5 C1 E1 A6"
266 DATA "FB40 11 01 00 A7 ED 52 F8 01 2C"
267 DATA "FB48 00 00 C9 C5 01 FE 7F ED 3C"
268 DATA "FB50 78 CB 47 C1 C9 CD 4B FB 72"
269 DATA "FB58 C8 DB FD E6 80 BB 28 F5 31"
270 DATA "FB60 5F 10 F2 C9 DD E5 FD E5 29"
271 DATA "FB68 E5 DD E1 FD 21 00 00 E5 09"
272 DATA "FB70 21 00 00 01 00 00 16 80 23"
273 DATA "FB78 DB FD E6 80 5F 06 00 DB F1"
274 DATA "FB80 FD E6 80 BB 20 11 10 F7 D1"
275 DATA "FB88 E1 FD E5 C1 DD E5 D1 FD 97"
276 DATA "FB90 E1 DD E1 ED 52 FB C9 CB F8"
277 DATA "FB98 22 DA 7C FB 3E E9 B8 F2 D7"
278 DATA "FBA0 AF FB AF B9 C2 C4 FB 0E 3C"
279 DATA "FBA8 01 CD EC FB C3 76 FB AF 3B"

280 DATA "FBB0 B9 CA 76 FB 23 CD F8 FB 82"
281 DATA "FBB8 3E 09 BD CA D4 FB 37 CB 52"
282 DATA "FBC0 1C C3 76 FB 23 CD 04 FC FB"
283 DATA "FBC8 3E 09 BD CA DB FB A7 CB D9"
284 DATA "FBD0 1C C3 76 FB AF B4 E2 E2 42"
285 DATA "FBD8 FB 18 OD AF B4 EA E2 FB 1D"
286 DATA "FBE0 18 06 E1 77 23 C3 6F FB A1"
287 DATA "FBE8 FD 23 18 F6 06 01 CD 10 F5"
288 DATA "FBF0 FC 10 FB 06 14 10 FE C9 E3"
289 DATA "FBF8 06 01 3A 10 FC 10 FB 06 51"
290 DATA "FC00 46 10 FE C9 06 01 3A 10 6A"
291 DATA "FC08 FC 10 FB 06 7D 10 FE C9 65"
292 DATA "FC10 C5 06 A0 10 FE C1 C9 F3 02"
293 DATA "FC18 C5 06 04 D9 CD 8D FC D9 EB"
294 DATA "FC20 10 F9 C1 10 F3 21 00 00 0A"
295 DATA "FC28 CD A1 FC 11 00 00 AF CB 19"
296 DATA "FC30 46 CC A1 FC C4 8D FC CB F3"
297 DATA "FC38 4E CC A1 FC C4 8D FC CB 03"
298 DATA "FC40 56 CC A1 FC C4 8D FC CB 13"
299 DATA "FC48 5E CC A1 FC C4 8D FC CB 23"
300 DATA "FC50 66 CC A1 FC C4 8D FC CB 33"
301 DATA "FC58 6E CC A1 FC C4 8D FC CB 43"
302 DATA "FC60 76 CC A1 FC C4 8D FC CB 53"
303 DATA "FC68 7E CC A1 FC C4 8D FC B6 4E"
304 DATA "FC70 EB EC A1 FC E4 8D FC 13 60"
305 DATA "FC78 ED 52 CD 8D FC EB C2 28 DE"
306 DATA "FC80 FC CD 8D FC DB FD FB C9 6A"
307 DATA "FC88 CD D6 FC 18 02 06 01 DB 1F"
308 DATA "FC90 FD CD B2 FC D3 FD CD B9 5A"
309 DATA "FC98 FC 10 ED C9 CD D4 FC 18 0B"
310 DATA "FCA0 03 F5 06 02 DB FD CD C2 03"
311 DATA "FCA8 FC D3 FD CD CA FC 10 EC FF"
312 DATA "FCB0 F1 C9 C5 06 6B 10 FE C1 6B"
313 DATA "FCB8 C9 C5 06 69 10 FE 3E 00 FD"
314 DATA "FCC0 C1 C9 C5 06 33 10 FE C1 13"
315 DATA "FCC8 AF C9 C5 06 2C 10 FE C1 02"
316 DATA "FCD0 3E 00 00 C9 C5 C1 3E 00 97"
317 DATA "FCD8 3E 00 C9 00 C9 C1 00 00 65"

НТК "ИНФОРКОМ" не может принимать на себя никакую ответственность за точность и достоверность сведений, содержащихся в объявлениях читателей.

ИНФОРКОМ

ищет зарубежные журналы, посвященные персональному компьютеру ZX-SPECTRUM:

SINCLAIR USER
ZX-COMPUTING
YOUR SPECTRUM
SINCLAIR PROGRAMS и др.

Возможные варианты:

- покупка по цене, назначенной Вами;
- обмен оригиналами;
- обмен копиями;
- обмен на время, с целью копирования.
- и др. условия по Вашему желанию.

При обращении просим предоставить список имеющихся у Вас журналов с указанием названия, года, номера, а также указать оригинал/копия.

БАЙТЕК просьба не предлагать.

107241, Москва, Б-241, а/я 37, НТК "ИНФОРКОМ"

ПРЕДЛАГАЮ

программы на кассетах и дискетах (игровые, тестовые, обучающие, демонстрационные, сервисные). Возможна разработка программ на заказ как для частных лиц, так и для организаций.

Обращаться по адресу: 129224 МОСКВА, И-224. а/я 116 Самойлову С.В.

Ищу принтер с интерфейсом ИРПР (IFSP) за разумную цену, а также информацию по ПК "Львів"

266024, г. Ровно-24, а/я 166, Юркиву А.Я.

Предлагаю программы и литературу к "ZX-Spectrum". В комплекте около 1000 программ. Каталог высылается бесплатно. Возможен обмен.

164413. п/о Катунино, Архангельской обл., ул.Кагунина, д.4, кв.20, Костюченко Ю.К.

Куплю принтер к ПК "Спектрум". Обменяюсь программами для "Спектрума", имею около 300. 460048. г.Оренбург, пр.Победы, д.144-А", кор.2, кв.572

Владельцам "ZX-Spectrum"

Предлагаю программные средства и комплект документации для изготовления дополнительной платы (всего 6 микросхем) к различным моделям самодельных компьютеров типа "ZX-Spectrum", позволяющий работать с обычной, удвоенной и утроенной плотностью записи на любом типе магнитофонных кассет.

При утроенной плотности на кассете МК-60 помещаются до 35-ти объемных игровых программ, время загрузки игры не более 2-х минут.

На ускоренных режимах загружается 90% игровых программ и все известные автору системные. Кроме того, системные программы автоматически поддерживают выбранную скорость работы с магнитофоном. Подключение дополнительной платы гарантированно не нарушает работоспособность программ. Оригинальный входной компаратор значительно повышает надежность считывания. За справками обращаться по адресу:

314032, г.Полтава, ул. Степного Фронта д.28, кв.144 Сороке Сергею Васильевичу.

Не забудьте вложить в письмо конверт с обратным адресом.

Продам компьютер БК-0010,01 в полной комплектации, с игровыми, системными, прикладными программами (свыше 250 программ). Срок эксплуатации компьютера - 3 месяца.

380002, г. Тбилиси, ул. Ленинградская, 5 Иванишвили Александр Ильич.

ПРОДАЮТСЯ:

ПК "Сиклер" (48К), декодеры ПАЛ, микросхемы серий К140, 142, 155, 176, 555, 561, КР 565, КР 580, КР 1533 и др. Заказы рассматриваются при получении почтового перевода на сумму 1 руб.

Обращаться:

220070, г.Минск-70, а/я 61

Загорский Александр Николаевич

МАСТЕР

АСП "МАСТЕР" предлагает компьютер "Сириус" - обладатель диплома ВДНХ СССР 1990 г.

- * Цветная графика, великолепный звук, джойстик.
- * Встроенный блок питания, высокая клавиатура.
- * Простота, в обращении и безотказность в работе.
- * Совместимость с известным английским компьютером ZX-Spectrum и русификация.
- * Существует более 5000 игровых и системных программ (к компьютеру прилагаются 3 кассеты С-90).

Цена компьютера 1500 руб.

Заказ высылать по адресу:

353410 краснодарский край, г. Анапа, ул. Краснодарская 25, компьютерный центр

"АСП"

Оплата производится почтовым переводом по адресу:

353410 Краснодарский край, г. Анапа,

р/с N 000609903 в Анапском Коммерческом банке, компьютерный центр.

В заказе указать, когда и откуда сделан перевод, квитанцию сохранить.

INFORCOM

Писать:

107241, МОСКВА-Б241 а/я 37 НТК "ИНФОРКОМ"

НТК "ИНФОРКОМ" приглашает откликнуться всех, имеющих способности к дизайну, художественной композиции и просто умеющих рисовать.

Если Вы готовы уделить несколько вечеров в месяц творческой работе и желаете заработать несколько десятков рублей, напишите нам, приложите конверт с обратным адресом, и мы вышлем Вам условия возможного сотрудничества.

Работа носит долговременный характер. Работа связана с дизайном экрана пэвм. Работы много и хватит на всех

Оснастите Ваш "Спектрум" двухрежимной ТУРБО-ПРИСТАВКОЙ

и Вы сможете записать на кассету около 45 больших игровых программ, каждая из которых будет надежно записана и считана

ВДВОЕ БЫСТРЕЕ

обычного, причем порядок загрузки и запуска программ при этом остаются таким же, как обычно.

Подключение приставки не влияет на работоспособность программ и объем ОЗУ; сохраняется возможность работы в режиме стандартной скорости, переключение режимов чисто, аппаратное и возможно в любой момент работы компьютера.

Для эффективного использования приставки не требуются специальные программы, в том числе и копировщики. Приставка может быть подключена к любому варианту "Спектрума".

Стоимость приставки с документацией - 54 руб. Оплата - наложенным платежом.

Время - деньги. Вы сэкономите и то и другое, если направите заказ по адресу. 270013, г.Одесса, аб. ящ. 12.

SPRITE

Предлагает программное обеспечение для ПЭВМ IBM PC/AT/XT, СПЕКТРУМ, КОРБЕТ, а также литературу по ПК "ЯМАХА" (MSX2). Каталог высылается бесплатно.

Обращаться по адресу: 474950 Каз. ССР, Тургайская обл. г. Державинск-1, 3/69 "СПРАЙТ".

ПЛЮС
НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС" - ПРЕДЛАГАЕТ ВАШЕМУ ВНИМАНИЮ
СВОИ УЧЕБНО-МЕТОДИЧЕСКИЕ РАЗРАБОТКИ

Методическая разработка (МР) N1.

Любители компьютерных игр найдут в ней несколько вариантов схем подключения джойстика. Наличие нескольких вариантов позволяет Вам выбрать наиболее приемлемый по наличию элементной базы.

Те, у кого есть принтер, смогут подключить его с помощью одной из схем последовательного или параллельного интерфейса принтера. Приведенные программы поддержки принтера (драйверы) позволят Вам распечатывать как текстовую, так и графическую информацию. Схемы интерфейсов и программы поддержки предназначены для работы с любыми принтерами, оборудованными одним из стандартных интерфейсов: CENTRONICS (ИРПР-М), ИРПР (IFSP), RS-232C (V.24).

Для занимающихся компьютерной графикой и живописью предназначена схема светового пера и распечатка работающего с ним графического редактора LP48K.

Схема программатора для ПЗУ с УФ стиранием позволит любителям технического творчества самостоятельно программировать такие микросхемы как 2716, 2764, 27128, 27256, 573 PФ2, 5, 4, 6, 8.

Объем МР1 - 34 страницы, стоимость - 30 рублей.

Со времени начала ее распространения (с сентября 1990г.) в адрес НТК не прекращается поток писем с положительными отзывами. Все схемы, входящие в разработку уже повторены многими пользователями, которые отмечают простоту изложения материала, доступность элементной базы и высокую повторяемость схем.

Методическая разработка (МР) N2.

Вторая методическая разработка МР2 состоит из двух частей. Первая часть посвящена контроллеру накопителя на гибких магнитных дисках НГМД. Контроллер полностью совместим с фирменной системой BETA-DISC INTERFACE. Он выполнен на базе микросхемы 1818ВГ93 (WD1793) и поддерживает работу до 4-х НГМД - 3-х или 5-ти дюймовых, односторонних и двусторонних, 40 и 80 дорожечных. Разработка содержит подробное описание схемы, сигналов, принципов работы интерфейса, рекомендации по сборке и наладке, систему команд дисковой операционной системы TRDOS.

О преимуществах работы с НГМД по сравнению с работой с магнитофонной лентой знают все, это новый уровень работы с компьютером, и мы надеемся, что наша разработка поможет Вам перейти на этот уровень.

Вторая часть МР2 посвящена стыковке компьютера с бытовыми телевизорами. Наиболее качественное цветное изображение получается при стыковке компьютера по R-G-B входу телевизора. При стыковке по антенному входу с кодировкой по системе СЕКАМ или PAL происходит некоторая потеря качества цветного изображения, но зато телевизор не требует переоборудования. В некоторых случаях может удовлетворить стыковка через видеовход с чернобелыми или цветными телевизорами без передачи информации о цвете, т.е. в черно-белом изображении.

Вторая часть МР2 содержит схемы всех перечисленных способов стыковки компьютера с телевизорами, причем схемы предназначены не только для полупроводниковых, но и для ламповых телевизоров.

Стоимость МР2 - 30 руб.

Методическая разработка (МР) N3.

Третья методическая разработка (МР3) подготовлена по многочисленным письмам и пожеланиям пользователей "Спектрума". В нее входит принципиально новый способ русификации "Спектрума", выгодно отличающийся от известных способов.

Знакогенератор кириллицы размещается в неиспользуемой области ROM, там же размещены некоторые подпрограммы, которые добавляют в операционную систему

"Спектрума" два новых регистра клавиатуры R и N к уже имеющимся K, L, C, E, G. Таким образом, в любой строке могут быть представлены буквы как русского, так и латинского алфавита одновременно, т.е. например, операторы Бейсика на английском языке, а текст на русском.

Также приведены дополнения к программам поддержки интерфейсов принтеров, опубликованных в нашей МР1 для вывода на печать смешанного русско-латинского текста.

Кроме этого, в МР3 по просьбам пользователей включены схемы интерфейсов джойстиков, рассчитанных на двух игроков - INTERFACE 1 и SINCLAIR, а так же схема программируемого джойстика, позволяющая запрограммировать все положения рукоятки и кнопки в соответствии с клавишами клавиатуры и более простой вариант - механически программируемый (путем перестановки перемычек) джойстик.

Стоимость МР3 - 30 руб.

Все программы, входящие в разработки, приведены в виде листинга и блоков шестнадцатиричных кодов.

Свои заявки на приобретение МР1-МР3 Вы можете направлять по адресу:

127566, Москва, И-566, НТК "Плюс". Оплата производится только наложенным платежом.

Просьба при всех обращениях к нам прикладывать к письму конверт с обратным адресом. В заявке необходимо указать свой полный адрес, фамилии, имя, отчество.

PS. Мы приносим извинение всем нашим заказчикам по поводу задержки МР2 и МР3. Ввиду непрерывных изменений правил игры мы постоянно сталкиваемся с трудностями в реализации наших планов и, тем не менее, продолжаем рекламировать нашу деятельность, рассчитывая в ближайшее время удовлетворить Ваши заявки. Это в наших интересах. Мы обслуживаем посредством наложенного платежа, но есть несколько заказчиков, выславших нам предоплату за эти разработки. Мы приносим им особые извинения и обратимся к ним письмом персонально к каждому.

С уважением! Члены НТК "ПЛЮС"



НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ИНФОРКОМ", 107241, МОСКВА, Б-241 а/я 37

Раздел для начинающих

Продолжение, начало на с. 23,45.

Сегодня мы продолжаем печатать небольшие программы для начинающих, присланные нашим постоянным читателем из г. Дубна, тов. Скитевым В.Н.

Спирали.

```
10 LET N=36:LET A=0:LET D=2*PI/N
20 CIRCLE 128,88,80
30 FOR I=1 TO N
40 LET X =40*COS A
50 LET Y=40*SIN A
60 PLOT 128, 88:DRAW X,Y,-PI:DRAW X,Y,PI
70 LET A=A+D
80 NEXT I
```

Звезда.

```
10 PLOT 128,174
20 DRAW 55,-160
30 DRAW -137,100
40 DRAW 164,0
50 DRAW -135,-100
60 DRAW 53,160
```

Парабола.

```
10 PLOT 1,88
20 DRAW 254,0
30 FOR A=1.5 TO .4 STEP -.1
40 PLOT 200,5
50 DRAW 0.165,A
60 LET C=(165/2) / (TAN A)
70 DRAW -C,-(165/2)
80 DRAW C,-(165/2)
90 NEXT A
```

Восход.

```
10 BORDER RND*6
20 INK RND*7
30 PAPER RND*6
40 CLS
50 LET Z=RND*10+2
60 FOR X=0 TO 174 STEP Z
70 PLOT 126,0
80 DRAW -126,X
90 BEEP .01,X/3
100 NEXT X
110 FOR X=-127 TO 127 STEP Z
120 PLOT 128,0
130 DRAW X,175
140 BEEP .01,60
```



```

150 NEXT X
160 FOR X=174 TO 0 STEP -Z
170 PLOT 128,0
180 DRAW 127,X
190 BEEP .01,X/3
200 NEXT X
210 PAUSE 200
220 GO TO 10

```

50-100 левая часть экрана

100-150 центральная часть экрана

160-200 правая часть экрана

Полное окрашивание экрана монитора. (2-красный, 4-зеленый, 1-синий).

```

10 CLS
20 BORDER 2: PAPER 2
30 FOR N = 0 TO 21
40 PRINT "
50 NEXT N

```

В строке 40 в кавычках 32 пробела.

Программы, представленные ниже достаточно сложны, поэтому сначала введите основную часть программы, а затем после проверки вводите следующую часть.

Пирамиды.

```

10 BORDER 0
20 PAPER 1
30 INK 6: CLS
50 FOR Y = 0 TO 20 STEP 2
60 PLOT 0, Y
70 DRAW 255,0
80 NEXT Y
90 FOR N=100 TO 220 STEP 30
100 FOR X=-10-N/10 TO 10+N/10
110 PLOT N, 35+N/10
120 DRAW X,-N/4
130 NEXT X
140 NEXT N

```

Продолжение пирамид.

```

190 FOR F=1 TO 30
200 LET X=RND*255
210 LET Y=RND*104+71
220 LET L=INT(175-Y)/8
230 LET C=INT(X/8)
240 PLOT 0,0
250 DRAW OVER 1; X,Y
260 BEEP .01,X/4
270 PLOT 0,0
280 DRAW OVER 1; X,Y
290 PRINT AT L,C; "*"
300 NEXT F

```

Мыльный пузырь.

```

10 BORDER 1
20 FOR Z=3 TO 10
30 LET H=INT(RND*25)
40 LET V=INT(RND*21)
50 PRINT INK 2;PAPER 6; FLASH 1; AT V,H;"X"
60 NEXT Z
70 LET X=1: LET Y=1
80 PRINT AT V,H;" "
90 LET V=V+Y

```

```

100 LET H=H+X
110 IF H=0 OR H=31 THEN LET X=-X: BEEP .02,24
120 IF V=0 OR V=21 THEN LET Y=-Y: BEEP .02,12
130 IF SCREEN$(V,H)="X" THEN PRINT INK 1; PAPER 5; AT V,H;"*": BEEP .5,25:STOP
140 PRINT AT V,H;"o"
150 PAUSE 2
160 GO TO 80

```

С очередной почтой поступила и довольно поучительная программа "ЦВЕТОМУЗЫКА", позволяющая получить на экране разноцветные узоры, изменяющиеся в такт музыке.

Ее прислал наш читатель из г. Минска Кипорук А.А.

Пользоваться программой очень просто. После того, как программа считана с магнитофона и запущена, вставьте в магнитофон кассету с музыкальной записью и включите его. Динамику изображения можно менять, изменяя уровень входного сигнала.

```

5 REM Настройка переменных.
10 PAPER 0: BORDER 0: CLS
15 LET a=65022: LET s0=255: LET t=0: LET x=0: LET y=21: LET c=1: LET n=2: LET d=0: LET d1=0:
  LET d2=0
19 REM Анализ изменения сигнала на входе
20 LET s= IN a
25 INK 0: GO SUB 5000: INK c
30 IF s<>s0 THEN GO SUB n*1000
40 LET t=t+1
50 GO TO 20
999 REM Вывод узора 1-го типа.
1000 IF t=1 THEN INK 5: GO SUB 5000
1003 LET x=x+(x<31 AND t=4) - (x>0 AND t=3)
1006 LET y=y+(y<21 AND t=2) - (y>0 AND t=1)
1010 IF t>=5 THEN LET y=INT (RND*21): LET x=INT (RND*30)
1020 IF t>6 THEN CLS:LET n=1+INT(RND*3): GO TO 1055
1025 IF t<>4 THEN INK c: GO TO 1050
1030 LET c=1+INT (RND*7)
1040 INK c: BRIGHT INT (RND*2)
1050 PRINT AT y,x;"█";AT y,30-x;"█";AT 21-y,x;"█"; AT 21-y,30-x;"█"
1055 LET s0=s: let t=0
1060 RETURN
1999 REM Вывод узора 2-го типа
2000 IF t=1 THEN LET c=5: GO SUB 5000: GO TO 2900
2003 INK 0: GO SUB 5100
2005 IF t>5 THEN CLS: LET n=1+INT (RND*3): GO TO 2900
2010 BRIGHT INT (RND*2): INK t
2020 LET d1=t+INT (RND*4): LET d2=d1+INT (RND*5)
3040 GO SUB 5100
2900 LET s0=s: LET t=0
2910 RETURN
2999 REM Вывод узора 3-го типа
3000 IF t=1 THEN LET c=5:GO SUB 5000
3005 IF t>5 THEN CLS: LET n=1+INT(RND*3): LET x=20: LET y=30:GO TO 3900
3010 IF t>4 THEN CLS
3050 LET c=1+INT (RND*7)
3060 LET x=INT (RND*256): LET y=INT (RND*176)
3070 INS c: PLOT x,y
3080 IF t<3 THEN DRAW -INT (RND*X),-INT (RND*y)
3090 IF t>2 THEN CIRCLE 20+INT (RND*210),20+INT (RND*130,INT (RND*20)
3900 LET s0=s: LET t=0: LET x=15: LET y=10
3910 RETURN
4999 REM Вывод угловых квадратов
5000 PRINT AT 0,0;"█"; AT 0,31;"█"; AT 21,0;"█"; AT 21, 31;"█"
5010 RETURN
5100 PRINT AT 10+d1,15+d2;"█" AT 10+d1,15-d2;"█"; AT 10-d1,15=d2;"█";AT 10-d1,15-
  d2;"█"
5120 RETURN

```

Мега-Бейсик

(Продолжение)

Управление выполнением программы

Процедуры.

Мега-Бейсик обеспечивает работу с процедурами, вызов которых осуществляется по их имени, точно так же, как и после команды, после имени процедуры может быть записана серия арифметических выражений или чисел. Эти выражения определяют необходимые переменные для работы в процедуре.

Недостатком процедур в Мега-Бейсике является то, что они не могут работать с локальными переменными: их переменные являются общими для всей программы.

Используя процедуры, можно разделить всю программу на ряд частных задач. Каждая процедура отлаживается отдельно, а затем все они объединяются в программу. Начало процедуры задается символом '@', следом за которым записывается имя процедуры. После имени процедуры записываются необходимые параметры. Окончание процедуры обозначается оператором `ENDPROC_`. Как только будет достигнут этот оператор, управление выполнением программы будет передано оператору, записанному за оператором вызова процедуры.

После оператора `ENDPROC_` можно записать имя процедуры:

```
9000 @DISPLAY_A,A$
9010 PAPER A: INK 9
9020 MODE_4: STIPPLE_5
9030 PRINT A$
9040 ENDPROC_DISPLAY
```

Это пример процедуры. В строке 9010 определяется имя процедуры `DISPLAY` и, как видно, для ее работы необходимо определить две переменные - `A` и `A$`. В строке 9030 определяется размер и оттенок символов, выводимых на экран. Для вызова процедуры необходимо записать, например:

```
DISPLAY_2,"MEGABASIC"
```

Процедуры могут вызываться и из процедур.

REPEAT-UNTIL

Мега-Бейсик обеспечивает работу с циклами типа `REPEAT-UNTIL`. Оператор `REPEAT` определяет начало цикла. После оператора `UNTIL` должна быть записана односимвольная переменная. Если значение этой переменной равно нулю, то цикл прерывается и выполнение программы переходит к оператору, записанному после последнего оператора `REPEAT`, если же эта переменная отлична от 0, то выполняется следующий оператор программы. Допускается вложение до 10 циклов типа `REPEAT-UNTIL`.

Процедурный стек

Для хранения номеров строк операторов, к которым выполняется возврат после использования процедур и циклов `REPEAT-UNTIL`, применяется специальный стек. Когда вызывается процедура, номер строки, следующий за оператором вызова, записывается в стек, что позволяет системе знать адрес возврата после окончания процедуры.

Аналогично и для циклов типа `REPEAT-UNTIL`. Здесь также в стек записываются номера строки и оператора, к которым осуществляется возврат после оператора `UNTIL`.

После выполнения оператора `ENDPROC_` и перехода к строке, номер которой хранился в стеке, этот номер удаляется со стека. Любая попытка записать в стек больше 10 номеров заканчивается сообщением `"PROC STACK OVERFLOW"` (Процедурный стек переполнен).

Команды `POP` и `PUSH`. Оператор `POP_` служит для "выталкивания" значений со стека, оператор `PUSH_` для записи значения в стек. За оператором `PUSH_` следуют 2 числа: номер

оператора в строке и номер строки. Для очистки стека перед началом выполнения программ необходимо записать команду PCLEAR.

Команда BRANCH_. По этой команде после выполнения каждой программной строки будет вызываться подпрограмма, записанная в строках основной программы. Начальная строка этой подпрограммы указывается после команды BRANCH, а конец подпрограммы определяется оператором ENDPROC_.

Оператор MTASK_. Это оператор многозадачного режима. Он позволяет выполнять программу одновременно из двух разных мест. Считается, что программа как бы разделена на две части: первая начинается после строки, где записан оператор MTASK_, а вторая - со строки, номер которой записан после оператора MTASK_. Мега-Бейсик выполняет поочередно строки из каждой части программы.

Отладка и редактирование программ

Операторы BRANCH_, TRON_, TROFF_ и SPEED_.

Оператор BRANCH_ оказывается весьма полезным при отладке программ. Если, к примеру, Вас интересует какая-то переменная, то Вы можете с помощью этой команды вызывать подпрограмму печати значения этой переменной после каждой программной строки. Другой способ проверить выполнение программы - выводить на экран номер выполняемой строки, что возможно по команде TRON_. При этом номер выполняемой строки программы будет выведен в нижний угол экрана, команда TROFF_ отменяет такой режим. Связана с этими двумя командами также команда SPEED_, после которой записано число, определяющее скорость выполнения программных строк. Максимально возможное число - 255. При этом выполнение программы будет приостановлено до тех пор, пока пользователь не нажмет любую клавишу.

Команды AUTO и DELETE_.

При вводе больших программ весьма полезным может быть автоматический ввод номера следующей строки после нажатия клавиши ENTER. Такой режим работы возможен при применении команды AUTO_, после которой следует 2 числа, разделенных запятой. Первое число задает начальный номер строки, с которого начнется автоматическая нумерация, а второе - шаг нумерации.

Остановить режим автонумерации можно так: EXTEND SYMB.SHIFT 'L'. Часто бывает необходимо удалить целый блок программных строк. Эту задачу можно решить, используя команду DELETE_. После этой команды записывают два числа, являющиеся соответственно начальным и конечным номерами строк удаляемого блока.

Команды BRON_ и BROFF_.

Желающие защитить свои программы от любопытных глаз, найдут полезным возможность отключать клавишу BREAK. это может быть сделано при применении команды BROFF_. Возвращение клавише ее возможностей осуществляется по команде BRON_.

Команда RESTART. Многие версии языка Бейсик имеют команду ON ERROR GO TO, позволяющую не прерывать выполнение программы в случае обнаружения ошибки, а осуществлять обход строки с ошибкой. Аналогичная команда имеется и в Мега-Бейсике. Это команда RESTART_, после которой записывается номер строки, к которой осуществляется переход в случае обнаружения ошибки. По команде RESTART_OFF такой режим отменяется. После обработки ошибки с помощью команды RESTART, некоторая полезная информация сохраняется в ячейках памяти: 59873/4 - номер строки, в которой обнаружена ошибка; 59875 - номер оператора внутри строки с ошибкой; 59862 - код обнаруженной ошибки.

Звук.

Мега-Бейсик открывает для Вас новые возможности создания звуков на Вашем ZX SPECTRUM. Для этого необходимо использовать команду PLAY_ и генератор звука ISG

PLAY_n,l,s,d,f

Здесь:

n=0 - ноты;

n=1 - белый шум;

l - длительность звучания на каждом шаге;

s - значение начальной звуковой частоты;

d - число шагов;

f - изменение частоты после каждого шага.

ISG

Если команда PLAY_ только улучшает возможности команды BEEP, то использование звукового генератора ISG позволит получить новые эффекты. Дело в том, что при выполнении программы в ZX SPECTRUM происходят прерывания 50 раз в секунду. Во время обслуживания прерывания специальная программа в машинных кодах производит сканирование клавиатуры. В это же время и работает ISG. В Мега-Бейсике имеется ряд команд, которые могут управлять этим генератором.

SOUND_n,a,b,c,d - команда для работы с звуковым буфером.

Здесь:

n=0 очистка звукового буфера перед очередной загрузкой;

n=1 добавление новой информации в звуковой буфер;

a=0 воспроизведение нот;

a=1 воспроизведение шума;

b - шаг звуковой частоты;

c - число шагов в последовательности нот;

d - число повторений последовательности нот.

Другие команды:

SOFF - отключение ISG;

SON - включение ISG;

SREP_n - количество повторений звуковой последовательности; если n=0, то данные, записанные в звуковом буфере будут использоваться только один раз, в противном случае - многократно.

Когда выполняется команда PLAY, ISG автоматически отключается. Для включения ISG применяйте команду SON. Более сложные звуковые эффекты можно получить, если замедлить выполнение программы. Этот короткий фрагмент программы демонстрирует работу ISG:

```
10 SOUND_ 0,0,1,20,255
20 SREP_1
30 SON
40 MODE_4: STIPPLE_6: FONT_2
50 VDU_(128+RND*15)
60 PAPER RND*7: INK 9
70 GO TO 50
```

Машинные коды и МЕГА-БЕЙСИК

Команды DOKE и CALL.

Мега-Бейсик позволяет включать в текст основной программы и фрагменты в машинных кодах. Мега-Бейсик занимает в ОЗУ участок памяти, начиная с адреса 45000.

Поэтому все программы в машинных кодах должны быть размещены в ОЗУ до этого адреса. Для того, чтобы записать в память двухбайтное число, можно воспользоваться командой DOKE, которая поддерживается в Мега-Бейсике. После этой команды должны быть записаны два числа: первое определяет адрес, куда записываются данные, а второе -

записываемое число.

Для вызова программ в машинных кодах в Мега-Бейсике используется команда **CALL_**. После этой команды записывается число, указывающее адрес вызываемой подпрограммы, а затем произвольные числа, которые передаются в машинный стек и затем используются как параметры в подпрограмме.

FRONT-PANEL.

Фронт-панель позволяет пользователю вносить изменения в ячейки памяти и регистры микропроцессора путем записи шестнадцатеричных чисел и является по сути встроенным монитором Мега-Бейсика.

Фронт-панель активизируется при выполнении команды **MON**, либо после **'SPACE F'** при выполнении программы.

Фронт-панель использует окно 3, в котором после активизации фронт-панели Вы увидите столбцы **HEX**-чисел. Слева - регистры процессора и их значения, а справа - распечатка участка ОЗУ. Звездочка указывает на текущий регистр, а инверсная полоса - текущую ячейку памяти. Все изменения производятся с помощью однобуквенных команд, после которых могут быть записаны до трех шестнадцатеричных чисел:

SPACE - возврат в Бейсик;

R nn - запись в текущий регистр 16-битового **HEX**-числа;

L nn nn nn - перемещение участка памяти. Здесь:

1-е число определяет начальный адрес перемещаемого блока;

2-е число задает новый начальный адрес;

3-е число определяет длину перемещаемого блока в байтах;

M nn - смещение указателя ячейки памяти на адрес **nn**;

S - установка точки выхода в программе для текущей ячейки;

K - продолжение выполнения программы после **S**;

U - восстановление точки выхода;

I nn nn nn - заполнение блока ОЗУ. Здесь:

1-е число - начальный адрес блока;

2-е число - длина блока;

3-е число - байт, которым заполняется блок;

J nn - вызов подпрограммы в машинных кодах, расположенной по адресу **nn**;

ENTER - смещение указателя ячейки памяти на один байт вперед;

"-" - смещение указателя ячейки памяти на один байт назад.

Новые команды.

Ниже приводится перечень новых команд Мега-Бейсика. Здесь литера после команды определяет число или числовое выражение, а литера со знаком **\$** - символьное выражение.

AUTO_a, b - автоматическая нумерация вводимых строк. Для выключения такого режима необходимо нажать **"="** и **ENTER**.

BACKUP - эта команда используется для копирования файлов. После ввода этой команды на экране появляется сообщение **"START TAPE"**

Как только на входе **EAR** появится сигнал, оно сменится новым сообщением **"OK searching"** и будет идти загрузка очередного файла с кассеты. После окончания загрузки на экране появляется сообщение:

"Start tape, then press any key".

Когда Вы будете готовы записать этот файл на кассету, нажмите любую клавишу и файл переписется. После этого система запросит Вас о еще одной копии. Если Вам она нужна, то нажмите клавишу **"Y"**, если же нет, то нажмите **"N"**. После этой команды необходимо выполнить команду **NEW**.

BRANCH_n - по этой команде будет вызываться подпрограмма после выполнения каждой программной строки.

BROFF - выключение клавиши **BREAK**.

BRON - включение клавиши **BREAK** после команды **BROFF**.

CALL_a(n,n...) - вызов программы в машинных кодах, по адресу 'a'.

После адреса в скобках могут быть записаны числа, которые записываются на машинный стек для использования в подпрограмме.

CHANGE_a,b - это команда для изменения атрибутного файла. Здесь 'a'- маска, а 'b'- данные.

CLW_(n),h - эта команда используется для очистки окон. Если записаны два числа, то первое определяет, какое из окон очищается. Если же записано одно число, то очищается текущее окно.

(Окончание следует)

Профессиональный подход

Сегодня мы продолжаем публиковать заметки знаменитого английского программиста Стива Тернера, автора не менее знаменитой игры OUAZATRON с полезными советами для тех, кто делает первые шаги в освоении "Спектрума".

Итак, следующая глава:

Орудия производства.

Для того, чтобы взять максимум возможного от микрокомпьютера, совершенно необходимо начинать программировать в машинном коде. "Спектрум" в состоянии выполнять в среднем порядка трехсот тысяч команд машинного кода в секунду. Когда же Вы программируете на Бейсике, то строки Вашей программы на самом деле ведь не исполняются компьютером. Вместо этого исполняется большая программа, называемая интерпретатором Бейсика, которая спрятана в ПЗУ. Эта программа преобразует Вашу строку в машинный код и исполняет его, поэтому для исполнения одной небольшой Бейсик-строки компьютер вынужден делать много тысяч разных операций.

Многие любители находят, что переход от программирования на Бейсике к машинным кодам представляет совершенно невообразимую по сложности и очень скучную задачу. Я называю это явление "Барьером машинного кода". В жизни существует много барьеров и они преодолимы. Как и звуковой барьер, барьер машинного кода преодолим с помощью правильной технологии и грамотной проработки проекта.

1. Редактор/Ассемблер.

Многие начинают освоение машинного кода с применения Бейсика для того, чтобы включить в свою программу машинно-кодovou процедуру путем применения операторов POKE и списков DATA.

Да, конечно, это самый простой способ, чтобы вставить в готовую Бейсик-программу готовую и отлаженную процедуру, но этот путь никак не годится для проработки, написания и отладки этой процедуры. Вы ведь работаете с числами, беря их из DATA и засылая их в нужные адреса POKE. Это очень неудобно, поскольку смысл этих чисел неочевиден. Гораздо удобнее применить программу Ассемблер, тогда создаваемый машинный код становится наглядным, его легче проследить и, конечно, отладить.

Более того, хорошая практика применения меток в Ассемблере сделает программу даже более читаемой, чем при программировании на Бейсике.

Сами посудите, оператор Ассемблера CALL SOUND больше говорит о своем назначении, чем строка Бейсика GO TO 9700.

Когда выбираете для себя программу Ассемблер, обратите внимание на его систему редактирования. К сожалению немногие программисты знают о хороших стандартах.

Ввод и редактирование программ должны быть такими же простыми, как печать писем в текстовом редакторе. Желательно, чтобы редактор Ассемблера позволял делать следующие операции:

- перемещение блоков;
- удаление блоков;
- копирование блоков.

Без этих функций программирование будет не только утомительным, но самое главное, подверженным многочисленным ошибкам. Я рекомендую работать с полноэкранным редактором, в котором Вы можете использовать весь экран, а редактор сам оформит подготовленный Вами текст в удобочитаемые колонки.

Управление курсором, удаление и вставка символов должны выполняться точно также, теми же клавишами, что и при обычном программировании в Бейсике, иначе Вы будете испытывать неудобство. К сожалению, многие пакеты игнорируют этот элементарный здравый смысл.

Следующей чертой, на которую надо обратить внимание - это требование, чтобы

Ассемблер после ассемблирования кода мог выполнить распечатку объектного кода в установленном формате. Стандартов в общем-то нет, но примером того, как она может выглядеть, служит процедура, приведенная в конце данной главы. (В следующем номере ZX-РЕВЮ - "ИНФОРКОМ").

Некоторые Ассемблеры распечатывают листинг до того, как рассчитаны адреса меток. Это не очень хорошо. И, что еще важно здесь учесть, это те ограничения, которые накладывает Ваше аппаратное обеспечение. Так, если Вы используете какую-то дисковую систему, то Ассемблер должен ее поддерживать.

2. Монитор.

Под "монитором" здесь понимается не дисплей, а дисассемблирующая программа. Их часто называют "мониторами", т.к. под их "управлением" производятся отладочные запуски разрабатываемой Вами программы. Я знаю некоторых программистов, работающих над игровыми программами, которые не пользуются мониторами. Очень не советую так поступать. Раз уж Вы работаете с компьютером, так используйте его возможности до конца. Пусть он поможет Вам в выполнении отладки Вашей программы.

Лучше всего выбирать такой Ассемблер, который имеет родственный ему монитор, тогда Вы сможете совместно их использовать. Конечно, монитор должен иметь возможность загружаться и без Ассемблера, поскольку в паре они занимают так много места в памяти, что вам остается пространство только на некрупные процедуры. Лично я предпочитаю использовать монитор минимальных размеров.

Необходимыми функциями монитора являются:

- просмотр содержимого ячеек памяти (адресов);
- просмотр содержимого регистров процессора;
- возможность изменения содержимого ячеек памяти;
- установка точек прерывания (Breakpoint).

Точки прерывания позволяют вам смело запускать не отлаженную процедуру. Когда программа дойдет до этой точки, она остановится, передаст управление монитору, и Вы сможете проверить содержимое регистров, адресов и т.п., а потом продолжить работу. Постепенно перемещая точку прерывания, Вы сможете найти, где в программе у Вас происходит сбой и внести изменения.

Точка прерывания выполняет функции, похожие на действие оператора Бейсика STOP.

Если же у Вас нет принтера, чтобы после ассемблирования распечатать объектный код, то Монитор становится для Вас абсолютно необходимым инструментом.

3. Внешняя память.

Если Вы собираетесь разрабатывать большие программы, то быстро работающее устройство для записи программы себя обязательно окупит. Когда я работал над программой QUAZATRON, то выгружал версии его десяти основных модулей более шестисот раз. А уж сколько раз я выполнял пробные запуски, об этом я и говорить не смею. Но дело не только в этом. Когда программа зависает или сбрасывается, Вам надо успеть войти в работу как можно быстрее, иначе нить рассуждений может быть потеряна и затраты времени станут еще больше. Лично у меня в среднем два раза в день компьютер "вылетает" только в связи с проблемами "питания".

Микродрайвы вполне удовлетворяют необходимым требованиям и дешевы, но лично мне больше нравится работать с дисковой системой. Хотя должен сказать, что для этого мне пришлось адаптировать под дисковод мой Ассемблер, а это была отнюдь непростая задача.

4. Принтер

С каждым годом принтеры становятся все дешевле и все доступнее. Для программиста принтер нужен прежде всего, чтобы распечатать текст программы, а его возможности по исполнению качественных документов отходят на второй план. Поэтому можно выбирать самый дешевый принтер из доступных.

Конечно подключается принтер к компьютеру через интерфейс и поэтому надо либо

выбирать стандартный интерфейс и, соответственно при выборе Ассемблера надо убедиться, что он его поддерживает, либо при нестандартном интерфейсе возникнет необходимость адаптировать Ассемблер под работу с Вашим интерфейсом.

5. Книги по Z-80.

О машинном коде процессора Z-80 написано уже немало книг. Наилучшим справочным руководством общепризнанно считается монография Л. Левенталя "ASSEMBLER LANGUAGE PROGRAMMING". В этой книге каждой команде процессора отведена отдельная страница, приведены подробнейшие данные об условиях работы каждой команды (состояния флагов, время исполнения и т.п.). Кроме этого, книга изобилует мно-¹ освоение машинного кода, то Вам необходимо что-нибудь попроще. Вы можете обратиться в ближайшую библиотеку и подобрать себе необходимые книги. Вы можете также обнаружить, что фирменное руководство к "Спектруму" в своих приложениях содержит очень много полезной информации.

Изучение машинного кода рано или поздно приведет Вас к необходимости понимания организации работы самого компьютера. Вам потребуется литература по ПЗУ компьютера. Наилучшим справочником по этому вопросу может служить книга доктора Яна Логана и доктора Фрэнка О'Хары "Полный дисассемблер ПЗУ" ("THE COMPLETE SPECTRUM ROM DISASSEMBLY").

* * *

Комментарий ИНФОРКОМа

В этом вопросе Стив Тернер вплотную подошел к тому, чем занимаемся мы, и в ответ на его предложение обратиться в ближайшую библиотеку мы считаем необходимым сделать ряд замечаний.

Конечно же в ближайшей библиотеке Вы почти ничего, а скорее всего вообще ничего не найдете. Если Вы живете в Москве, то еще сможете получить в читальном зале ГПНТБ на Кузнецком Мосту русский перевод книги Виккерса, являющейся официальным фирменным руководством к "Спектруму", а может быть даже заказать его ксерокопию, но что делать если Вы живете в другом городе?

Здесь Вам поможет "ИНФОРКОМ". Мы уже писали о нашем трехтомнике для изучающих программирование в машинных кодах. Более подробную информацию можете найти в рекламном приложении на стр. 104.

Уверены, что с его помощью Вы сможете решить основные проблемы.

По вопросам, касающимся ПЗУ компьютера, мы можем Вас отослать к постоянному разделу "Секреты ПЗУ", в котором пусть не столь полно, как в книге Я.Логана и Ф.О'Хары, но зато более популярно мы рассказываем о программах, входящих в ПЗУ.

Кроме того, по индивидуальным заказам мы подготовили и уже разослали желающим копию английского издания этой книги. У нас еще осталось около 40 экземпляров и, если Вам она необходима, то можете успеть ее заказать. Поставку не гарантируем, Вам может не хватить, т.к. сроки приема заказов уже кончились, и потому это единственное, что мы высылаем наложенным платежом. Пожалуйста, не надо высылать предоплату, а только запрос. Цена - 112 руб.

Теперь относительно работ Лэнса Левенталя. Кроме упомянутой выше книги "Z-80 ASSEMBLY LANGUAGE PROGRAMMING" (600 стр.) у него есть еще логическое продолжение, написанное в соавторстве с Уинтропом Сэвиллем - "Z-80 ASSEMBLY LANGUAGE SUBROUTINES" (500 с.)

В этой книге рассматриваются многочисленные примеры создания процедур в машинных кодах.

Мы в принципе могли бы рассмотреть вопрос о том, чтобы сделать несколько копий и той и другой книги и разослать их наложенным платежом, аналогично тому, как мы это сделали с книгой Логана и О'Хары. Правда, надо иметь в виду, что при нынешних ценах на копирование и их внушительном объеме стоить это будет очень дорого, о чем мы объявим в случае наличия пожеланий.

¹ В оригинале пропущена строка (Прим. OCR)

С книгой же Виккерса все гораздо проще. На сегодняшний день она распространена очень широко и ее русский перевод можно купить на всех рынках страны. Ее распространяют многие кооперативы. Мы ее не распространяем, поскольку на русском языке мы предлагаем только то, что сделали сами или то, что сделано по нашему заказу, а к данной книге это не относится.

(Продолжение следует)

Секреты ПЗУ

Программы, обслуживающие работу с магнитофоном.

Этот раздел ПЗУ занимает адреса с 04C2 по 09F3. Основная задача содержащихся здесь процедур - исполнение четырех команд - SAVE, LOAD, VERIFY и MERGE. Они выполняют все необходимые для этого функции:

- генерируют пилоттон (широкие полосы на экране телевизора, появляющиеся перед загрузкой/выгрузкой каждого блока программы;
- записывают и считывают "хэдер" (17-байтный заголовок, который предшествует загрузке/выгрузке каждого программного блока (program), а также блоков "машинных кодов (bytes) и блоков данных (character array или number array);
- записывают и считывают сам программный блок.

Общим в работе всех подпрограмм, обслуживающих команды SAVE, LOAD, VERIFY и MERGE является то, что адрес блока, для обработки которого они предназначены, должен быть предварительно установлен в регистровой паре IX, а длина этого блока - в регистровой паре DE. При этом аккумулятор (регистр A) содержит 0, если обрабатывается 17-байтный "хэдер", а если обрабатывается сам программный/кодовый блок или блок данных, то в аккумуляторе находится число 255 (0FFH).

Начальная точка входа в этот раздел ПЗУ находится в адресе 0605H, после чего исполнение разветвляется, в зависимости от того, с какой командой из SAVE, LOAD, VERIFY и MERGE Вы работаете.

Основные наиболее крупные блоки этого раздела такие:

0462-053E.

Подпрограмма "SA-BYTES".

Выполняет передачу байтов, содержащихся начиная с адреса, на который указывает содержимое IX, в количестве, указанном в DE, на внешний порт магнитофона. Эта же подпрограмма управляет изменением цвета бордюра при загрузке/выгрузке и звуковым динамиком.

053F-0555.

Подпрограмма "SA/LD-RET".

Небольшая подпрограмма, общая для операций SAVE и LOAD. Выполняет необходимые финишные операции после окончания загрузки/выгрузки.

0556-0604.

Подпрограмма "LD-BYTES".

Выполняет прием байтов с внешнего входного порта магнитофона при операциях LOAD, VERIFY, MERGE. Как и для программы "SA-BYTES" в паре IX находится адрес, начиная с которого располагается загружаемый (проверяемый) блок, а в паре DE - его длина.

0605-096F.

Главная рабочая область данного раздела.

Поскольку адрес 0605 является точкой входа в пакет процедур, обслуживающих работу с магнитофоном, здесь размещается основной логический блок, который и управляет прохождением исполнения поданной Вами команды.

0970-09A0.

Подпрограмма "SAVE CONTROL".

Выполняет вспомогательные действия при выгрузке.

Осуществляет связь с пользователем, например выдает сообщение "Start tape, then press any key" и ожидает нажатия клавиши.

09A1-09F3.

Это не подпрограмма. В этих адресах хранятся сами тексты сообщений (кроме сообщений об ошибках), которые могут появиться при работе с магнитофоном.

Например, хорошо знакомое сообщение, появляющееся при начале загрузки "Program:" или "Bytes:" и др.

Поскольку процедуры, связанные с обслуживанием магнитофона имеют особую важность для пользователей, мы их рассмотрим более подробно, раскрыв логику их работы и взаимосвязь. Эти данные могут пригодиться тем, кто сам организует нестандартный ввод/вывод, занимается разработкой загрузчиков, защищающих от копирования через копировщики, а также исследует возможность снятия таких защит.

Многие так или иначе работают вопросами ускорения ввода/вывода аппаратно. Поможет эта информация и тем, кто занимается разработкой или анализом копирующих программ.

На приведенных ниже блок-схемах показана взаимосвязь между отдельными процедурами, входящими в этот раздел, а далее прокомментированы эти процедуры.

Условные обозначения:

— переход, ветвление (команда Ассемблера JP);

= вызов с возвратом (команда CALL или RST).

Тем, кто не знает машинного кода или языка Ассемблера Z-80 и по этой причине будут испытывать трудности при чтении материала, очень решительно советуем приобрести наш трехтомник по машинному коду для начинающих.

SAVE_ETC: 0605-0620

Это главная точка входа в программу для всех четырех команд LOAD, SAVE, VERIFY, MERGE. Чтобы установить какая конкретно из четырех обрабатывается, в системной переменной T_ADDR (23668 DEC) выставляется соответствующее число:

SAVE - 00; LOAD - 01;

VERIFY - 02; MERGE - 03.

Процедура выполняет подготовку к выгрузке или загрузке "хэдера". При работе вызывает процедуры:

- EXPT_EXP (1C8C) - для передачи числовых параметров "хэдера" на стек калькулятора;

- SYNTAX_Z (2530) - для проверки в каком режиме работает БЕЙСИК - в режиме исполнения команды или проверки синтаксиса. В последнем случае выполняется переход вперед на процедуру SA_DATA.

SA_SPACE: 0631-0636

Резервирует место в рабочей области БЕЙСИКа для размещения "хэдера", для чего обращается к процедуре BC_SPACES(0030).

SA_BLANK: 0639-0641

Выполняет проверку параметров имени.

Вызывает процедуру STK_FETCH (2BF1) для ввода со стека калькулятора параметров имени.

Если имя пустое или длиннее десяти символов, что допустимо только для LOAD, VERIFY и MERGE, то переход на SA_NULL, если с именем все в порядке - переход на SA_NAME.

REPORT_F: 0642-0643

Если имя неправильное, вызывается ERROR_1 (0008) с кодом перехватчиком 0E, что означает "invalid file name".

SA_NULL: 0644-064A

Если имя - пустое, переход на процедуру SA_DATA. "Обрезает" имя, если оно более 10

СИМВОЛОВ.

SA_NAME: 0648-0651

Размещает имя в рабочей области БЕЙСИКа.

SA_DATA: 0652-0669

Определяет с каким типом объекта производится операция.

Если это не блок данных, т.е. либо экран, либо программа, либо машинный код, то переход на процедуру SA_SCR\$ и далее.

Если это блок данных, то переход на SA_VOLD, если имеем дело с массивом данных, имеющимся в памяти, что бывает при операциях SAVE и VERIFY, или переход на SA_NEW, если обрабатываем новый блок данных.

Вызываемые процедуры:

- GET_CHAR (0018) - проверяет код первого символа для идентификации "DATA" это или нет;

- REPORT_C (1C8A) - вызывается при попытке сделать MERGE для блока данных, что конечно же невозможно. А оттуда запускается ERROR_1 (0008) с кодом перехватчиком 0B, что означает ошибку "Nonsense in BASIC";

- NEXT_CHAR(0020)- подготовка к приему нового символа.

- LOOK_VARS (28B2) - в таблице переменных разыскивает массив данных.

REPORT_2: 0670-0671

Сообщение об ошибке, если соответствующий массив данных в памяти не существует (не разыскан). - Variable not found.

SA_V_OLD: 0672-0684

Обрабатывается массив, существующий в памяти.

Вызываемые процедуры: SYNTAX_Z - см. выше.

Если идет проверка синтаксиса, то переход на SA_DATA_1.

SA_V_NEW: 0685-068E

Определяется тип данных - символьные или числовые.

SA_V_TYPE: 068F-0691

Подготавливает для выгрузки или загрузки код типа данных в качестве первого байта "хэдера".

SA_DATA_1: 0692- 069F

Здесь анализируется та часть команды загрузки/выгрузки, которая следует за оператором DATA и за именем массива. Следующим символом должна идти открывающая скобка "(" . Если это не так, то возврат на SA_V_OLD, а оттуда на сообщение об ошибке REPORT_C.

Вызываемые процедуры:

- NEXT_CHAR(0020) - см. выше.

- CHECK_END(1BEE) - обеспечивает выдачу сообщения об ошибке, если вся строка до конца не прошла успешную проверку на синтаксис.

SA-SCR: 06A0-06C2

Проверяет, является ли обрабатываемый блок экраном. Если нет, то переход на SA_CODE и далее, при попытке сделать MERGE для экрана выдает сообщение об ошибке REPORT-C. Если все в порядке, подготавливает регистры процессора к работе с экранной областью и переходит на SA_TYPE-3.

SA_CODE: 06C3-06E0

Проверяет, является ли обрабатываемый блок блоком машинных кодов. Если нет, то очевидно это БЕЙСИК (больше ничего не остается) и тогда переход на SA_LINE. При попытке сделать MERGE выполняет переход на REPORT_C. Если идет операция SAVE, то проверяется есть ли параметры начала блока и длины. Выгружать коды только по имени нельзя. В этом случае также переход на REPORT_C.

Вызываемые процедуры:

- NEXT_CHAR - см. выше;
- PR_ST_END (2048) - проверяет закончен ли оператор. Если да, то возвращает 0 и работа продолжается, если нет, то строка не закончена. ПЗУ находится не в режиме исполнения, а в режиме проверки синтаксиса и следует переход на процедуру SA_CODE_1.
- USE_ZERO (1CE5) - помещает 0 на стек калькулятора в качестве адреса начала блока. Работа продолжается переходом на SA_CODE_2.

SA_CODE_1: 06E1-06EF

Анализирует синтаксис, дает переход на REPORTS при попытке сделать SAVE без указания начального адреса и длины блока.

Вызываемые процедуры:

- EXPT_1NUM (1C82) - процедура калькулятора. Рассчитывает значение последнего выражения. Результат становится последним вложением на вершине стека калькулятора, которое далее считается адресом начала обрабатываемого блока кодов.
- GET_CHAR (0018) - см. выше. Продолжение работы - SA_CODE_3.

SA_CODE_2: 06F0-06F4

Посредством вызова процедуры USE_ZERO (1CE6) помещает 0 на вершину стека калькулятора в качестве параметра длины обрабатываемого блока машинного кода. Продолжение работы - SA_CODE_4.

SA_CODE_3: 06F5-06F8

С помощью процедуры калькулятора EXPT-1NUM рассчитывает значение последнего выражения, помещает его на вершину стека и считает его длиной блока кодов.

Начиная с этого времени все параметры хранятся в той части рабочей области БЕЙСИКа, которая отведена для хранения "хедера".

SA_CODE_4: 06F9-0709

С помощью процедуры калькулятора FIND_INT2 (1E99) преобразует параметр "длина" блока из интегральной формы, применяющейся в расчетах калькулятора в двухбайтную, приемлемую для машинного кодирования. Потом то же самое для параметра "начальный адрес".

SA_TYPE_3: 0710-0715

Подготавливает для выгрузки или загрузки код типа данных в качестве первого байта "хедера". В данном случае и для экрана и для блока маш. кодов этот тип - 3.

SA-LINE: 0716-0722

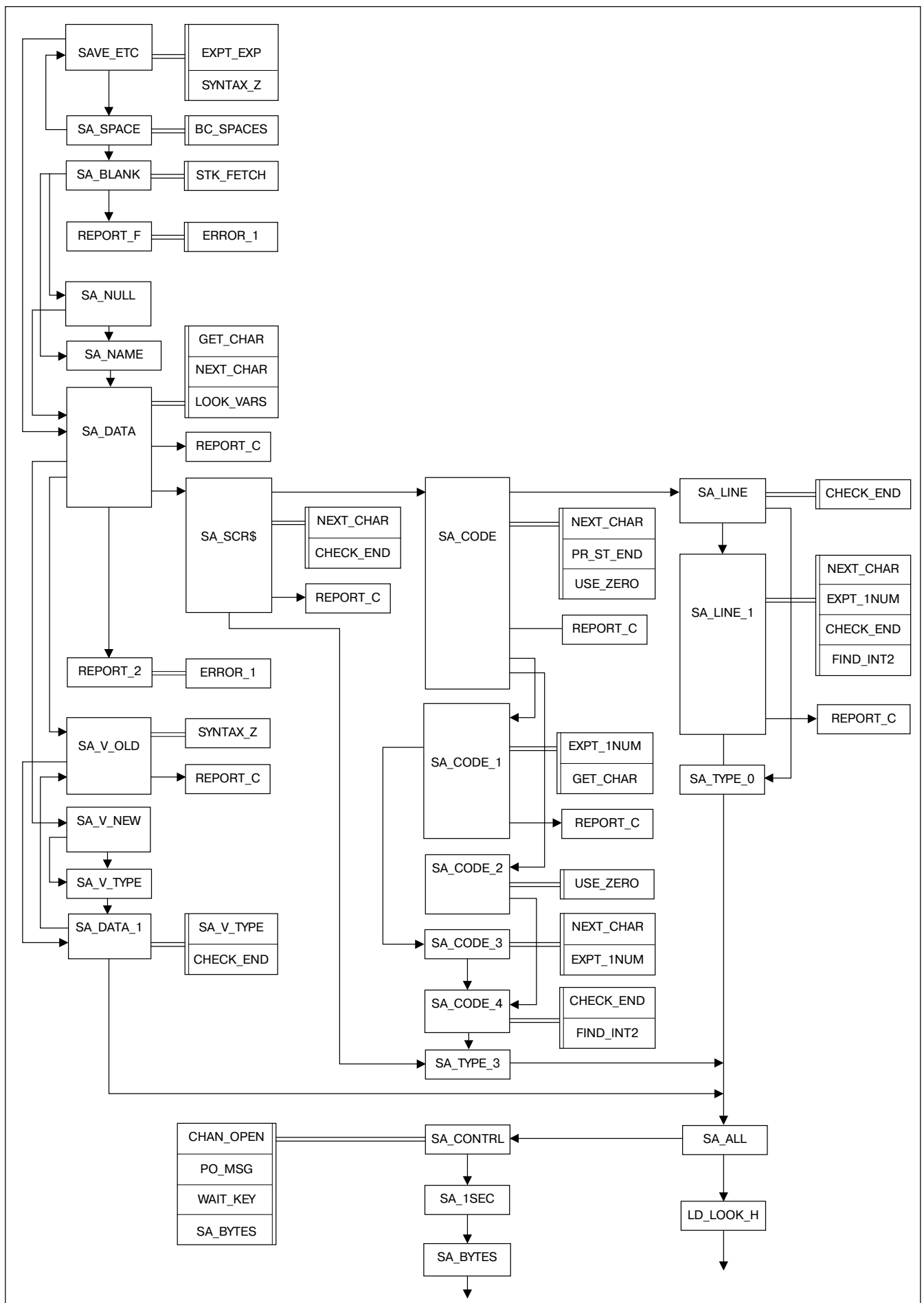
Проверяет присутствует ли оператор LINE в командной строке. Если да, то переход на SA_LINE_1. Туда же переход в режиме проверки синтаксиса, если по результатам вызова CHECK_END (1BEE) установлено, что строка не завершена.

Продолжение работы - переходом на SA_TYPE_0.

SA_LINE_1: 0723-0739

Если не в режиме SAVE, то оператору LINE здесь делать нечего и выполняется переход на REPORT_C.

Принимается число, стоящее после LINE, помещается на вершину стека калькулятора, переводится в двухбайтную форму и запоминается.



SA_TYPE_0: 073A-0759

Записывает в рабочую область БЕЙСИКа, отведенную для "хэдера" информацию о программе и ее переменных:

тип (в данном случае он равен нулю), а также содержимое системных переменных PROG и E-LINE.

Итак, в результате этих операций в рабочей области БЕЙСИКа сформировалась область информации по "хэдеру" загружаемого или выгружаемого блока. На начало этой области указывает число, находящееся в регистровой паре IX процессора, а далее информация распределена следующим образом:

IX+00 - тип (данные, коды, Бейсик).

IX+01 - IX+0A - имя (десять символов). Если имени нет, то по адресу IX+01 должно быть число FF.

IX+0B - IX+0C - количество байтов в блоке данных.

IX+0D - IX+10 - прочая информация, зависящая от типа.

SA_ALL: 075A-0766

Здесь программа начинает вводить различия для разных операций. Для операции SAVE управление передается на SA_CONTRL, а для остальных операций продолжается исполнение процедурой LD_LOOK_H.

Напоминаем, что различает программа о какой операции идет речь по содержимому системной переменной T-ADDR(23668-десятиричный адрес).

LD_LOOK_H: 0767-0789

Процедура выполняет поиск заголовка ("хэдера") на ленте и печать его на экране. Поиск организован в цикле, пока "хэдер" не будет найден.

Если заголовок найден, но его тип не совпадает с ожидаемым - переход на LD_TYPE. Если совпадает, то дается команда на сравнение десяти символов имени и выполняется прямой проход на LD_TYPE.

Вызываемые процедуры: - LD_BYTES (0556) – выполняет загрузку встреченного заголовка. Подробнее об этой подпрограмме ниже.

CHAN_OPEN (1601) - открывает канал "S" для печати имени на экране.

LD_TYPE: 078A-07A5

Печатает на экране сообщение Program или Bytes, Number array, Character array. Печать выполняется вызовом предназначенной для этого процедуры PO_MSG(0C0A).

Если в "хэдере" прочитан несуществующий тип - 4 и более, возврат на LD_LOOK_H.

Если принятое имя надо сравнить с имеющимся в компьютере, то переход на LD_NAME, но если в памяти содержится пустое имя, то прямой проход туда же с указанием, что сравнивать их не надо.

LD_NAME: 07A6-07AC

Сравнивает имена символ за символом и переходит на LD_CH_PR. Если имена сравнивать не надо, сразу выполняется переход туда же.

LD_CH_PR: 07AD-07CA

Печатает символ за символом, используя для этого вызываемую процедуру PRINT_A_1 (0010). После печати очередного символа возвращается за новым в LD_NAME.

Если в имени оказалось менее 10 символов, то это не имя и тогда выполняется возврат в процедуру LD_LOOK_H для поиска на ленте другого блока.

Далее исполнение разветвляется в зависимости от того, с какой из загрузочных команд мы имеем дело.

VERIFY - VR_CONTRL(07CB)
LOAD - LD_CONTRL(0808)
MERGE - ME_CONTROL(08B6).

VR_CONTRL: 07CB-07E8

Процедура принимает длину загружаемого блока, сравнивает ее с заказанной и переходит на VR_CONT_1. Если длина не была указана при подаче команды, то сразу переходит туда же. Если длина читаемого блока не совпадает с заказанной длиной, переход на REPORT_R (0806).

VR_CONT_1: 07E9-07F3

Выставляет адрес начала загружаемого (проверяемого блока) и переходит на VR_CONT_2. Если при подаче команды этот параметр был опущен, то принимается тот, который был прочитан в заголовке загружаемого блока и сразу выполняется переход туда же.

VR_CONT_2: 07F4-07FF

Принципиально верификация мало чем отличается от загрузки. Просто при верификации байт считанный с магнитфонного порта не запоминается в памяти, а только сравнивается с тем, который там находится. Поэтому значительная часть процедур, обрабатывающих режим VERIFY используется также и в режиме LOAD, если речь идет о загрузке блоков маш. кодов (Bytes) или экранов (Screen\$), но только не БЕЙСИКа.

Различие начинается здесь, в этой процедуре.

Процедура выставляет флаг C, если идет режим LOAD и снимает его, если режим VERIFY.

VR_CONT_3: 0800-0801

Выставляет FF в аккумуляторе (подготовительная операция перед загрузкой блока).

LD_BLOCK: 0802-0605

Вызывает процедуру LD_BYTES(0556), выполняющую загрузку. О ней мы поговорим ниже. Если загрузка прошла нормально (а проверяют это проверкой флага C) - "выход". Если нет, переход на REPORT_R.

REPORT_R: 0806-0807

При появлении ошибки вызывает процедуру ERROR_1(0008) с кодом перехвата 1A, что означает Tape loading error.

LD_CONTRL: 0808-0818

Отсюда начинается управление загрузкой БЕЙСИКа.

Возможны два варианта - загружается БЕЙСИК-программа или именованный массив данных.

Напомним, что управлением загрузкой прочих типов занималась ветвь программы, предназначенная для верификации - VR_CONTRL и др.

Из "хэдера" принимается длина загружаемого массива. Если она не указана - переход на LD_CONT_1, а если указана - на LD_CONT_2.

LD_CONT_1: 0819-0824

Проверяет, достаточно ли в существующей рабочей области БЕЙСИКа места под загружаемую переменную или массив. Если достаточно, то переход на LD_DATA(082E).

Если нет - продолжение работы.

LD_CONT_2: 0825-082D

Резервирует под переменную дополнительно 5 байтов и проверяет хватает ли на это

оперативной памяти. Проверка выполняется вызовом TEST-ROOM(1F05).

LD-DATA: 082E-084B

Здесь рассматривается загрузка массива.

Если идет загрузка БЕЙСИК-программы, то сразу переход на процедуру LD_PROG.

Если загружается новый массив, то сразу делается переход на процедуру LD_DATA_1.

Если такой массив уже существует, резервируется дополнительное место под описание массива и работа продолжается.

Вызываемая процедура:

- RECLAIM_2 (19E8) – выполняет резервирование памяти.

LD_DATA_1: 084E-0872

Отводит место под содержимое элементов массива в конце области переменных БЕЙСИКа.

Резервирование места в этой области выполняет вызываемая процедура MAKE_ROOM (1655).

Далее производится сама загрузка блока данных (массива). Это выполняется переходом на процедуру LD_BLOCK (0802), которую мы уже рассмотрели.

LD_PROG: 0873-08AC

Отсюда начинаются операции по загрузке БЕЙСИК-программы вместе с переменными.

Сначала с помощью процедур RECLAIM_1(19E5) и MAKE_ROOM(1655) отводится место в памяти. Выставляются некоторые системные переменные. Если программа загружается без номера строки автостарта, то выполняется переход на LD_PROG_1, а если с номером, то выставляются соответствующие системные переменные и потом делается переход туда же.

LD_PROG_1: 08AD-08B5

"Оформляет" БЕЙСИК-программу как блок данных, выставляя соответствующие значения в регистрах процессора и переходит для загрузки на LD_BLOCK(0802) и далее, о чем мы уже говорили.

ME_CONTRL: 08B6-08D1

Отсюда начинается управление исполнением команды MERGE.

Работа происходит в три приема. Сначала подзагружаемые БЕЙСИК-строки загружаются как блок данных, затем производится слияние БЕЙСИК-строк, а затем сливаются новые переменные.

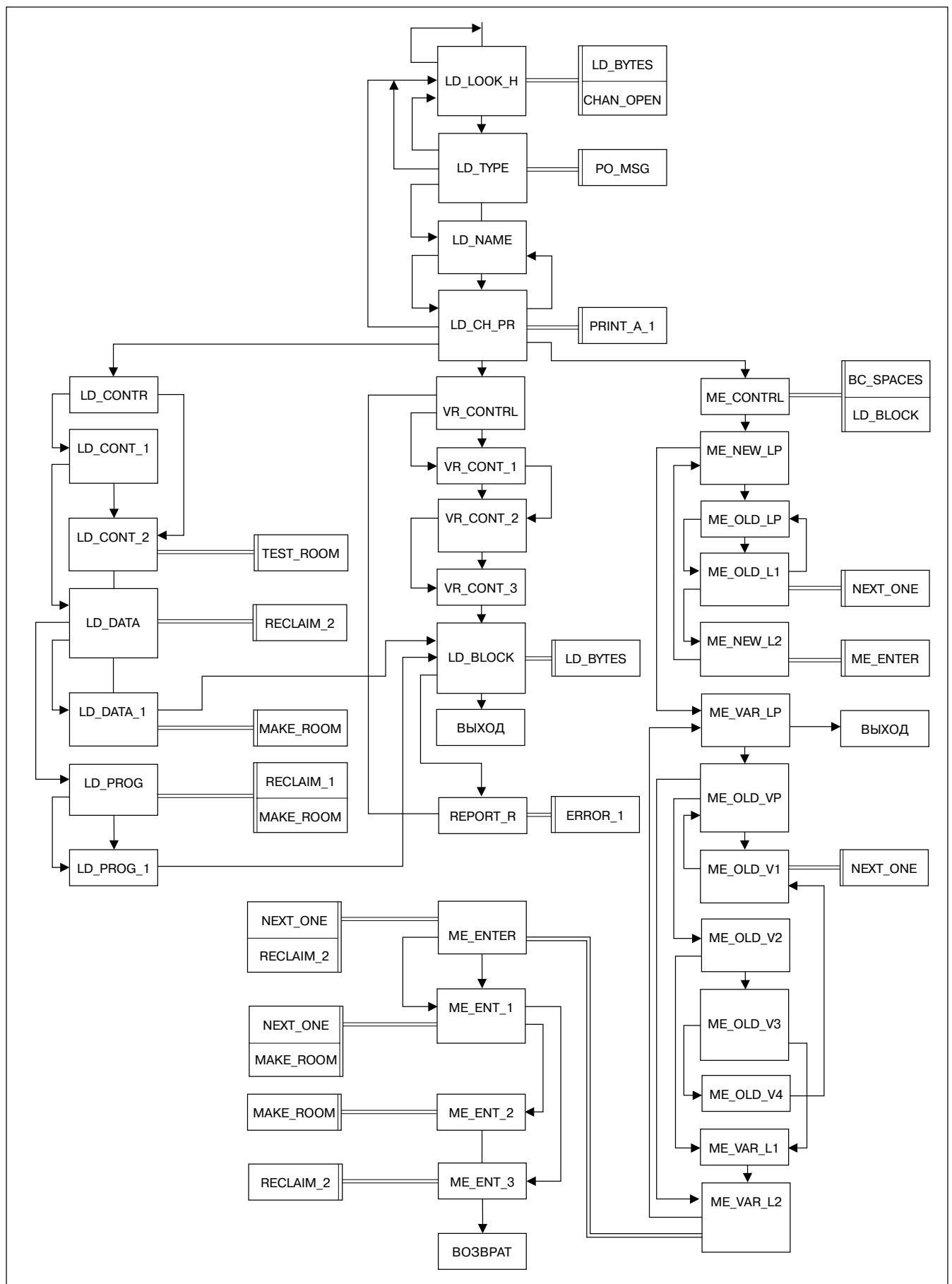
Процедура ME_CONTRL сначала принимает параметры загружаемого блока и резервирует в рабочей области необходимое место посредством вызова BC_SPACES(0030), затем подготавливает загрузку блока и выполняет ее вызовом LD_BLOCK (см. выше).

ME_NEW_LP: 08D2-08D6

Это начало внешнего цикла, в котором один за другим вводятся номера строк новой программы и проверяются. Когда все строки просмотрены, выполняется переход на ME_VAR_LP(08F0).

ME_OLD_LP: 08D7-08DE

Это начало внутреннего цикла, в котором рассматриваются строки "старой" программы. Проверяется и сравнивается с номером "новой" строки сначала старший байт номера. Если они не совпадают, то сразу переход к следующей процедуре, в противном случае проверяется еще и младший байт.



ME_OLD_L1: 08DF-08EA

Когда по результатам сравнения включится флаг CARRY регистра F это означает, что подходящее место для вставки "новой" строки найдено и выполняется переход на ME_NEW_L2.

В противном случае отыскивается с помощью вызываемой процедуры NEXT_ONE адрес, в котором начинается следующая строка "старой" программы и выполняется возврат к началу цикла (ME_OLD_LP) для рассмотрения очередной строки "старой" программы.

ME_NEW_L2: 08EB-08EF

Вызовом процедуры ME_ENTER (см. ниже) выполняется вставка "новой" строки между строками "старой" программы и происходит переход в начало внешнего цикла - к процедуре ME_NEW_LP.

ME_VAR_LP: 08F0-08F8

Примерно так же, как выше происходило слияние строк, здесь начинается слияние переменных.

Это начало внешнего цикла. Сначала производится проверка на конец работы. Если все имена переменных обработаны - "ВЫХОД". В противном случае продолжение работы.

ME_OLD_VP: 08F9-0901

Это начало внутреннего цикла, в котором сканируется область переменных "старой" программы и просматриваются их имена.

Если найден маркер конца области, то выполняется переход на ME_VAR_L2.

Имена переменных старой и новой программы сравниваются сначала по первой букве. Если они совпадают, то сразу делается переход на ME_OLD_V2, где их рассмотрят более подробно.

ME_OLD_V1: 0901-0908

Если имена не совпадают, то с помощью вызываемой процедуры NEXT_ONE (19B8) устанавливается адрес следующей "старой" переменной и происходит возврат к началу внутреннего цикла ME_OLD_VP.

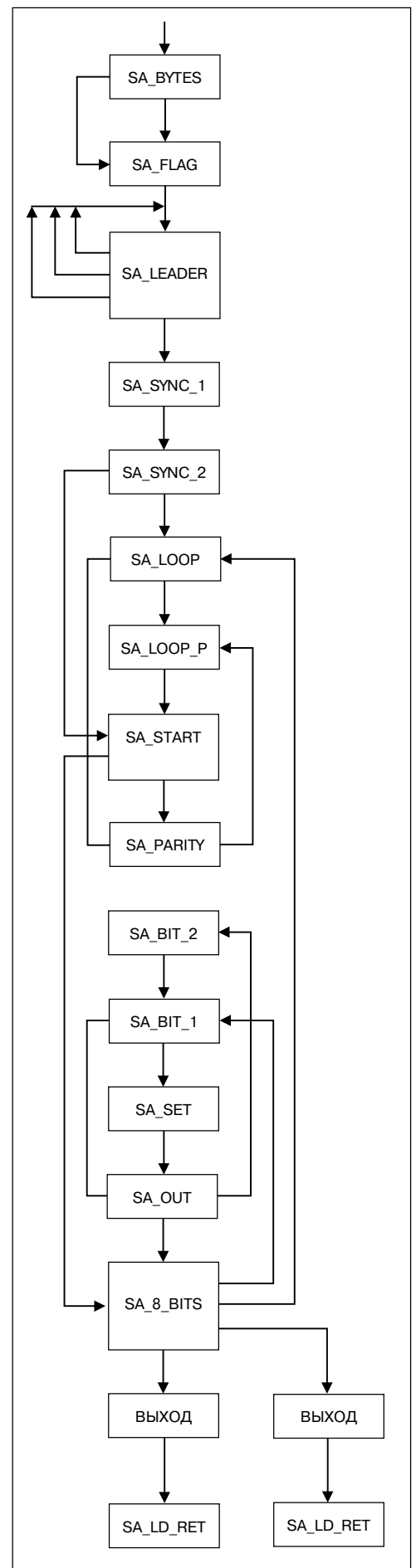
ME_OLD_V2: 0909-0911

Если имена состоят из одной буквы, а они совпадают, то выполняется переход на ME_VAR_L1, если же имя "длинное", то с переменной займется следующая процедура.

ME_OLD_V3: 0912-091D

Сравниваются вторые и последующие буквы имен, для чего в процедуре организован внутренний цикл. Если какая-то из букв не совпадает, то выполняется переход на ME_OLD_V4.

Если установлено, что имена совпали, то выполняется переход вперед на ME_VAR_L1.



ME_OLD_V4: 091E-0920

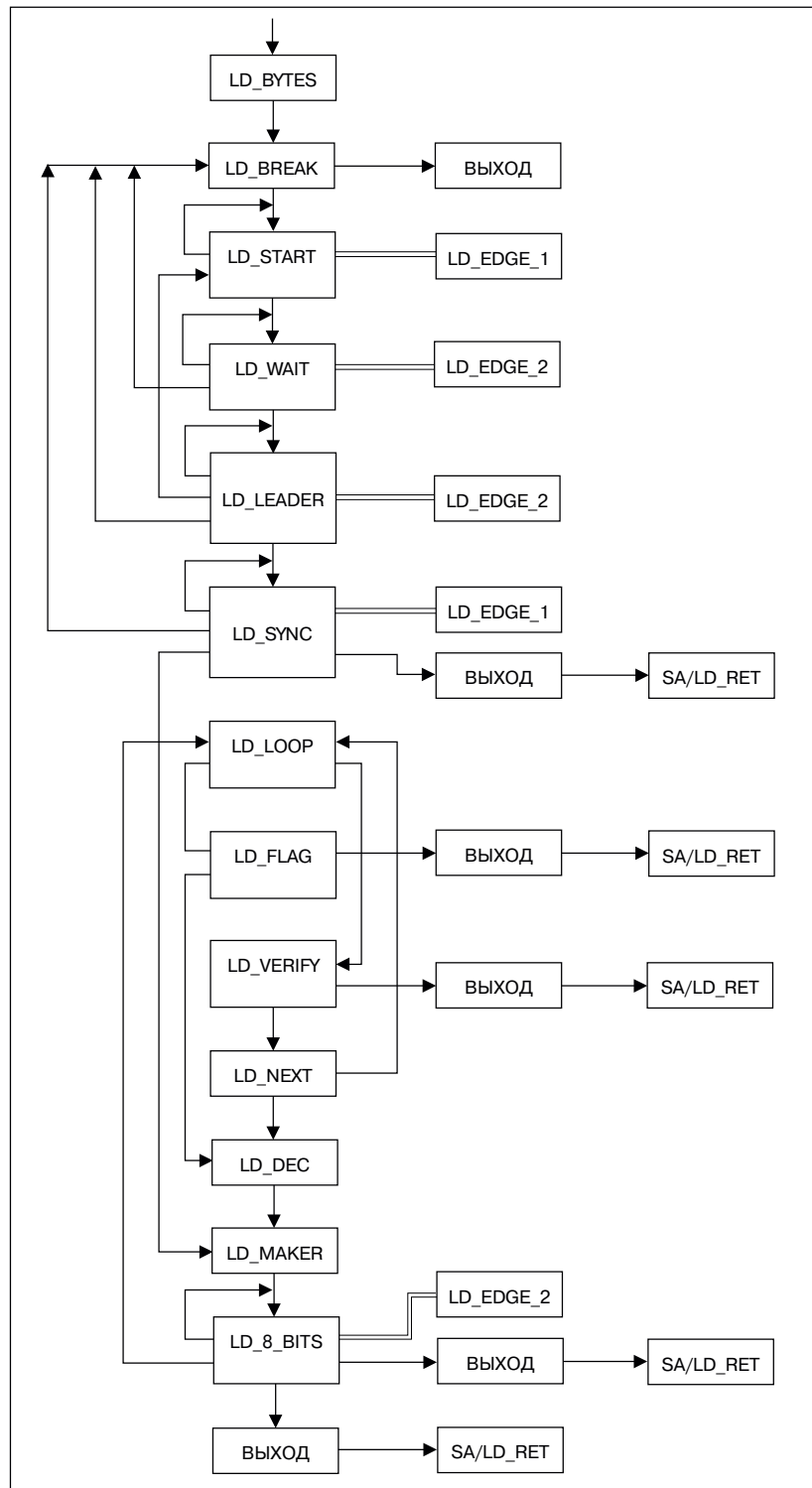
Если имена в предыдущей процедуре не совпали, то отсюда выполняется возврат в начало внутреннего цикла для выбора следующей переменной - ME_OLD_V1.

ME_VAR_L1: 0921-0922

Сюда Вы попадаете, если переменные совпали (по имени). Здесь в регистр А засылается код FF, что в дальнейшем будет означать - "заменить содержимое".

ME_VAR_L2: 09E3-092B

Сюда Вы попадаете, если достигнут конец области переменных и есть переменные, которым не нашлось пары. В результате регистр А содержит код + 80, что означает "должны быть добавлены переменные".



Процедура вызывает подпрограмму ME_ENTER для добавления новой переменной и возвращается к началу внешнего цикла ME_VAR_LP.

ME_ENTER: 09EC-093D

С этой процедуры начинается пакет, предназначенный для "сливания" строк БЕЙСИКа и программных переменных. Регистры процессора и флаги несут в себе информацию, необходимую для выполнения этих операций:

Флаг CARRY (C)

Выключен - слияние БЕЙСИК-строки. Включен - слияние переменной.

Флаг ZERO (Z)

Выключен - добавление. Включен - замена.

Регистровая пара HL

Указывает на адрес начала новой строки (переменной).

Регистровая пара DE

Указывает на адрес ее места назначения.

Процедура ME-ENTER проверяет флаг Z и если речь идет о добавлении, а не о замене, сразу передает управление на ME_ENT_1.

Вызываемыми процедурами NEXT_ONE и RECLAIM_2 (см. выше) запрашивает и заносит в регистры информацию об адресе старой переменной (строки).

ME_ENT_1: 093E-0954

Проводит подготовительные операции, например резервирует место для новой строки. Переходит на ME_ENT_2, если добавляется переменная и на ME_ENT_3, если строка.

ME_ENT_2: 0955-0957

Резервирует место для новой переменной.

ME_ENT_3: 0958-096F

Выполняет добавление/замену. Удаляет из рабочей области копии "новых" строк и переменных. Переустанавливает системную переменную PROG и заканчивает работу, выполняя возврат в вызывающую процедуру.

SA_CONTRL: 0970-0990

Мы возвращаемся к операции выгрузки - SAVE. Данная процедура служит для управления процессом выгрузки.

Сначала выполняется вызов CHAN_OPEN(1601), что открывает канал "K", который служит для вывода информации в нижние две строки экрана.

Затем вызовом PO_MSG(0C0A) выполняется печать в этих строках системного сообщения Start tape and press any key.

Далее процедура WAIT_KEY(15D4) организует ожидание нажатия клавиши.

И, наконец, вызовом SA_BYTES, о чем мы еще поговорим ниже выполняется выгрузка "хэдера" на ленту. Он выгружается таким образом, что первый байт указывает на тип выгружаемого блока, а последний байт является байтом чётности.

SA_1_SEC: 0991-09A0

Выдерживает одну секунду и, после подготовки, выполняет переход на SA_BYTES для выгрузки самого блока.

09A1 - 09F3 - сообщения.

Здесь хранятся тексты системных сообщений, выдаваемых на экран во время работы с магнитофоном. Последний байт каждого сообщения инвертирован (к нему прибавлено число +80H).

09A2 - Start tape, then press any key.

09C1 - # Program:

09CB - # Number array:

09DA - # Character array:

09EC - # Bytes:

Мы здесь употребили знак # вместо символа "возврат каретки" CHR\$ 13.

Теперь мы рассмотрим вызываемые процедуры, выполняющие выгрузку байтов - SA_BYTES и их загрузку - LD_BYTES.

SA_BYTES: 04C2-04CF

Сначала процедура выставляет на машинном стеке (не путать со стеком калькулятора) адрес 055F. Это означает, что по окончании операции, при выходе из нее по команде RET, будет сделан переход по этому адресу, где расположены финишные процедуры, связанные с окончанием загрузки/выгрузки SA/LD_RET и другие.

Устанавливаются постоянные, обеспечивающие продолжительность пилоттона перед "хэдером" продолжительностью 5 сек (в HL устанавливается число 1F80). Если SA_BYTES была вызвана для выгрузки "хэдера", то отсюда сразу следует переход на процедуру SA_FLAG. Если для выгрузки блока, то выставляется продолжительность пилоттона примерно 2 сек (в HL число 0C96).

SA_FLAG: 04D0-04D7

Отключает маскированное прерывание, загружает в аккумулятор 02, что означает цвет бордюра - красный и состояние динамика - включен.

SA_LEADER: 04D8-04E9

Здесь организуются выдача импульсов, формирующих сигнал пилоттона. Это сделано введением трех циклов. Продолжительность каждого импульса (включен/выключен) - 2048 тактов процессора - первый цикл, выполненный на базе команды DJNZ. Количество фронтов импульсов определяется содержимым регистровой пары HL. После каждого прохода L уменьшается на единицу - это второй цикл на базе команды JR NZ. Если L упадет до нуля, то снимается единица из H, что вызывает появление третьей ветви возврата к началу процедуры по команде JP P. Движение по третьей ветви конечно более длинное, чем по второй, поэтому в ней вводится компенсация DEC B.

Мы специально так подробно комментируем эту очень короткую процедуру, с которой Вы теперь разберетесь даже имея минимальные навыки по машинным кодам, "прощелкав" командой PEEK содержимое указанных адресов (если нет под рукой дисассемблера). Дело в том, что создание собственного пилоттона, отличного от стандартного - излюбленный прием тех, кто делает процедуры, защищенные от копирования. Впрочем, это только один из множества приемов.

SA_SYNC_1: 04EA-04F1

Вырабатывается синхронизирующий импульс. На входе в процедуру стоит задержка, обеспечивающая отключение динамика в течение 667 тактов, при этом цвет бордюра - красный, затем командой OUT по порту FE выдается сигнал на динамик и переустанавливается содержимое аккумулятора для изменения цвета бордюра на голубой.

SA_SYNC_A: 04FE-04FD

Окончание формирования синхроимпульса. На входе в процедуру стоит задержка, обеспечивающая включенное состояние динамика в течение 735 тактов. Затем командой OUT (FE), А выключается динамик и включается голубой цвет бордюра (в аккумуляторе - 0D).

Тем, кто не знает как производится управление внешним портом FE при вводе и выводе, мы можем только порекомендовать заглянуть на стр. 89-91 нашей книги "Первые шаги в машинных кодах" (т.1 трехтомника). Там все очень популярно написано. Мы не случайно называем этот трехтомник базовым - мы будем ссылаться на него еще не раз.

Далее управление передается процедуре SA_START.

SA_LOOP: 04FE-0504

Процедура является вершиной цикла, связанного с выгрузкой пакета байтов.

Проверяется не достиг ли нуля счетчик выгружаемых байтов (пара DE). Если да, то перед концом работы надо выдать байт четности, для чего переход на SA_PARITY. Если нет - продолжить работу.

SA_LOOP_P: 0505-0506

Фиксируется текущее значение четности.

SA_STAST: 0507-050D

Подготавливается побитная выдача на порт магнитофона восьми битов, составляющих байт. Включается флаг переноса (CARRY), служащий маркером для 8-ми битов байта. По нему процедура SA_8_BITS будет распознавать, что все 8 битов отправлены. В аккумуляторе выставляется 01 (бордюр красный, динамик включен).

Управление передается процедуре SA_8_BITS.

SA_PARITY: 050E-0510

Сюда Вы попадаете, если блок фактически выгружен и осталось отправить завершающий байт четности. Он вводится в регистр H и управление передается назад в SA_LOOP_P для последнего прохода.

SA_BIT_2: 0511-0513

Каждый бит выгружается после двух проходов по нижележащей петле. На первом проходе вырабатывается как бы передний фронт импульса, на втором – задний. Процедура SA_BIT_2 - вершина второго прохода. Здесь устанавливается желтый цвет бордюра и сигнал на отключение динамика.

SA_BIT_1: 0514-0519

Вершина петли первого прохода. Здесь расположен замедляющий цикл, обеспечивающий задержку на время 801 такта процессора. Далее если выгружаемый бит равен нулю, то переход на SA_OUT.

SA_SET: 051A-051B

Если выгружаемый бит равен единице, то здесь выполняется принудительная задержка еще на 855 тактов в дополнение к имевшимся ранее 801. Поэтому частота следования импульсов нулей в "Спектруме" примерно в два раза выше частоты единиц.

SA_OUT: 051C-0524

Выдает на порт FE при первом проходе 01 (динамик включен, бордюр-синий), а при втором проходе 0E (динамик выключен, бордюр желтый). После первого прохода переход на SA_BIT_2, после второго - на SA_8_BITS.

SA_8_BITS: 0525-053B

Организует побитную выдачу восьми битов байта. Отправляет бит за битом в SA_BIT_1, пока не достигнут ранее установленный маркер.

Путем опроса внешнего порта клавиатуры (тоже FE) проверяет нажатие клавиши BREAK и в этом случае выполняет выход и, соответственно, переход по содержимому машинного стека на SA/LD_RET.

Проверяет счетчик выданных байтов (DE) и, если он не достиг FFFF, обращается назад к процедуре SA_LOOP. Обращаем внимание, что когда счетчик равен нулю, то нужен еще один проход для выдачи байта четности, отсюда и взялось число FFFF, которое в двоичной дополнительной арифметике тождественно минус единице.

SA_DELAY: 053C-053D

Короткая задержка перед выходом.

Теперь рассмотрим работу пакета процедур, непосредственно занимающихся

загрузкой байтов.

При входе сюда аккумулятор содержит 00, если загружается "хэдер" или FF, если блок.

Флаг C включен при операциях загрузки и выключен при верификации.

LD_BYTES: 0556-056A

Процедура вызывается как при загрузке информации из "хэдера", так и при загрузке/верификации самого блока.

Выполняет подготовительные операции, отключает маскируемое прерывание, устанавливает цвет бордюра белым (0F на порт FE). Устанавливает на машинный стек адрес возврата 053F (SA/LD_RET), считывает байт, поступивший с порта FE, но рассматривает только бит EAR (бит 6) путем ротации RRA и маскирования AND 20, подготавливает переключение бордюра на красный, включает флаг нуля (Z).

LD_BREAK: 056B

Проверяет факт нажатия клавиши BREAK по состоянию флага Z. Если он выключен, выход на SA/LD_RET.

LD_START: 056c-0573

Вызывает процедуру LD_EDGE_1 для поиска фронта сигнала. Если в течение заданного времени он не отыскивается (свидетельствует об этом выключенный флаг переноса), то выполняется возврат на LD_BREAK. Если найден, бордюр становится голубым.

LD_WAIT: 0574-057F

Чтобы убедиться, что сигнал не случайный, здесь выполняется задержка примерно на 1 сек. И повторяется попытка найти фронт вызовом LD_EDGE_2. Если его нет, то возврат на LD_BREAK.

LD_LEADER: 0580-058E

Вызывает LD_EDGE_2 для поиска фронта сигнала. Если он не поступил в отведенное время, возврат на LD_BREAK. Если поступил, то по содержимому регистра В определяется время, прошедшее с момента приема предыдущего фронта. Если оно больше 3000 тактов, то выполняется возврат на LD_START.

В регистре Н подсчитывается количество пар фронтов. Только когда их насчитывается 256 пилоттон считается принятым и - работа может быть продолжена.

LD_SYNC: 058F-05A8

За пилоттоном идут две части синхроимпульса. Любой принятый фронт считается за передний фронт импульса "on", но только если в заданном интервале времени поступит еще один фронт, он будет принят за задний, иначе возврат на LD_BREAK.

Теперь ожидается импульс "off". Если он не поступил - выход на SA/LD_RET с выключенным флагом переноса (ошибка ввода).

Теперь здесь перед загрузкой "хэдера" или блока данных подготавливаются регистры к приему первого байта, свидетельствующего о типе данных и последнего байта (байта четности). Работа продолжается переходом на LD_MARKER.

LD_LOOP: 05A9-05B2

Вершина цикла загрузки байтов. Отсюда управление распределяется в следующем порядке:

При загрузке первого (флагового) байта - на LD_FLAG.

При верификации - на LD_VERIFY.

При загрузке последующих байтов - на LD_NEXT.

LD_FLAG: 05B3-05BC

Здесь проверяется первый загруженный байт, определяющий тип загружаемых

данных. Если он не совпадает с тем, что было указано в команде, выполняется выход на SA/LD_RET. В противном случае - продолжение работы - переход на LD_DEC.

LD_VERIFY: 05BD-05C1

В аккумулятор вводится текущий байт из памяти. В регистре L - байт, принятый с ленты. Если они не совпадают - выход (SA/LD_RET).

LD_NEXT: 05C2-05C3

Переход к очередной ячейке памяти.

LD_DEC: 05C4-05C7

Уменьшает на единицу счетчик загружаемых, байтов, вводит в регистр В постоянную времени, сравнением с которой впоследствии определяют что было загружено - ноль или единица.

LD_MARKER: 05C8-05C9

Очищает регистр L, в котором будет "собираться" из битов загружаемый байт и включает в нем младший бит. После приема 8-ми битов он смещаясь влево отойдет в позицию флага С и это будет свидетельствовать о том, что байт принят полностью. Таким образом он выполняет роль маркера.

LD_S_BITS: 05CA-05E2

Организуется цикл по загрузке 8-ми битов от младшего до старшего. Цикл заканчивается, когда включится флаг С.

Загрузка битов выполняется вызовом LD_EDGE_2. Там "ловится" пара фронтов, замеряется интервал между ними. Сравнивается с величиной, установленной в В. Если больше, то принята единица, а если меньше, то ноль.

В регистре Н фиксируется текущее значение четности.

По содержимому DE определяется весь ли блок обработан и если да, то выход на SA/LD_RET, а если нет, то возврат в вершину цикла - в процедуру LD_LOOP.

Далее следуют процедуры, занимающиеся отысканием на ленте фронтов импульсов и определением временного интервала между ними, по которому и судят о том, что это - ноль, единица, пилоттон или вообще случайный импульс.

LD_EDGE_2: 05E3 - 05E6

Служит для отыскания пары фронтов, первого путем вызова LD_EDGE_1, а второго путём прямого прохода туда же.

LD_EDGE_1: 05E7-05E8

Устанавливает в аккумуляторе величину первичной задержки.

LD_DELAY: 05E9-05EC

Выполняет эту задержку в течение 358 тактов перед входом в сравнивающую процедуру.

LD_SAMPLE: 05ED-0604

Организуется цикл, в котором с порта FE "ловится" фронт импульса. Регистр В выполняет роль счетчика проходов (таймера). Если он достиг нуля, а фронта нет - это означает "время истекло" и выполняется возврат с включенным флагом нуля (Z). Одновременно по сигналу с этого порта определяет факт нажатия BREAK-клавиши и выходит с выключенными флагами переноса (C) и нуля, если она нажата.

Если в отведенное время фронт найден, то меняет цвет бордюра и состояние звуковой динамика, и производится возврат в вызывающую процедуру.

Нам осталось только рассмотреть несколько финишных процедур, занимающихся оформлением окончания процессов загрузки/ выгрузки/ верификации/ слияния.

SA/LD_RET:053F-0551

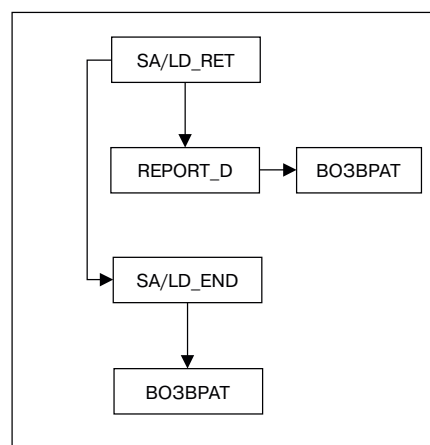
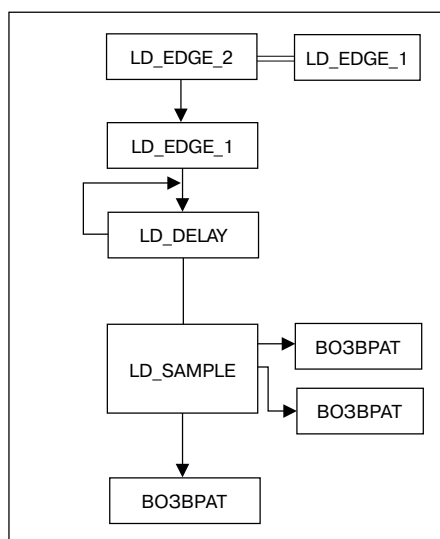
Восстанавливает исходный цвет бордюра, в последний раз проверяет нажатие клавиши BREAK, разрешает маскируемое прерывание. Если BREAK не нажата, о чем свидетельствует включенный флаг C, то переходит на SA/LD_END.

REPORTED: 0552-0553

Если нажата BREAK-клавиша, вызывается процедура обработки ошибок ERROR_1 (0008) с кодом перехватчиком 0C, что означает выдачу сообщения: BREAK-CONT repeats.

SA/LD_END: 0554-0555

Восстанавливает со стека флаги и возвращает в вызывающую программу.



FORUM

Мы продолжаем начатый разговор по письмам читателей о пределах совместимости отечественных модификаций "Спектрумов" с фирменным программным обеспечением. Пусть не сразу, пусть постепенно, но надеемся с Вашей активной помощью внести какую-то ясность в этот вопрос.

Тов. Змерзлюк из г. Краснокаменска сообщает, что на его компьютере модели "Дубна-48" не работают программы TF-COPY, COPY-86/M, LERM-7, а также программы, имеющие нестандартный загрузчик типа BOMB JACK, BOULDER DASH и др. Он был бы очень рад узнать о причине этого явления.

Завод "ТЕНЗОР" на запрос товарища Клейменова по этому поводу любезно сообщил, что ПК "Дубна-48" может не работать с теми программами, которые выполняют проверку содержимого ПЗУ.

Что по этому поводу можно сказать? В своей книге "Первые шаги в машинном коде Z-80" мы высказали отношение к доработкам ПЗУ и предложили выполнять их посредством замещающего, "теневого" ПЗУ. В рекомендациях мы основывались на многолетнем опыте, накопленном в Европе при работе с этим и другими подобными компьютерами. Сэр Синклер чего только не модифицировал в своих машинах, но вот ПЗУ - не трогал, хотя хорошо знал о том, сколько в нем ошибок. Да и фирма "АМСТРАД", перекупив у него все права на производство Синклер-совместимых моделей, тем не менее, даже начав выпускать модели 128K, обеспечила возможность работы в режиме 48K почти со 100%-ной совместимостью. Фирма "Таймекс" в своей модели 2048, серьезно переделав архитектуру компьютера, не тронула ПЗУ, а к модели 2068, которая была исполнена с расчетом на работу в дисковой системе под управлением CP/M, что потребовало полной замены ПЗУ, прилагает специальный кэтриндж Синклер-совместимости, который вставляется в специальный отсек. Есть непроверенные сведения, что так же поступили и с компьютером UNIPOLBRIT в Польше.

Мы предлагаем всем, кто столкнулся с подобными проблемами и умеет читать машинный код фирменных программ обратить усилия на то, чтобы вскрыть в них места, в которых они выполняют проверку ПЗУ и организовать обход, по возможности несложный. Мы же с радостью напечатаем сообщения о ваших открытиях.

Конечно же проблема эта относится совсем не только к "Дубне", а и к десяткам других моделей. Нужна координация усилий и очень своевременной выглядит идея научно-производственной фирмы "Икар" о создании Всесоюзной ассоциации производителей аппаратного, программного и информационного обеспечения компьютеров, совместимых с "ZX Spectrum" (ВАПРОС).

Не работает "BOMB JACK" на версии компьютера "АИОА" (Козлов, Сургут).

Товарищ Смирнов из Ленинграда пишет о том, что у него на компьютере версии BALTIC не идет программа "ELITE", т.е. она стартует, но через три-четыре минуты "зависает". То же проявляется и на других, проверенных им компьютерах этой версии. Кто что-то знает по этому вопросу, отзовитесь!

Поступили сведения о том, что то же происходит и с программой "THE LAST WORD TWO" после проведения в ней русификации. Она стартует, после набора нескольких символов зависает. Поскольку сведения поступили примерно от пяти человек из разных мест (Витебск, Мурманск и др.) и совершенно совпадают, можно предположить, что это не случайность. С другой стороны, есть уже сотни сообщений о том, что русификация прошла успешно. По-видимому, сказывается какая-то особенность применяемого компьютера. Делать выводы пока рано. Не хватает статистики, ждем Ваших наблюдений.

Интересная информация поступила от товарища Ермакова о появлении модификации компьютера "Синклер-90". Судя по приведенной им информации, ПЗУ компьютера настолько развито, что обеспечивает очень мощный мониторинг исполняемых программ и даже доступ к регистрам процессора.

Нам об этом ничего не известно, но если кто-то из авторов разработки читает эти строки, то мы надеемся на то, что он сочтет возможным дать на страницах "ZX-РЕВЮ" несколько уроков по работе с новой системой.

Крайне интересный вопрос задал в своем письме товарищ Максимов из г. Кемерово. Он посетовал на то, что нет единого стандарта в присвоении букв русского алфавита клавишам "Спектрума". Действительно, вопрос весьма важен, т.к. от него может зависеть совместимость разрабатываемого Вами программного обеспечения.

Мы решали этот вопрос, когда несколько лет назад выполняли русификацию мощного текстового редактора "THE LAST WORD TWO". Примененный там подход мы изложили в разработке, посвященной этой программе, но поскольку не все ее имеют, попробуем осветить его здесь.

Во-первых, Вы конечно знаете, что русификацию можно проводить двумя основными методами. Первый основан на использовании символов графики пользователя. Эти символы имеют номера со 144-го по 164-ый включительно. Если Вы с их помощью набираете русский текст, то вводите его в графическом режиме (курсор G). Понятно, что 21 символа Вам недостаточно, чтобы задать все буквы русского алфавита (хотя конечно есть способы создания и переключения многочисленных банков UDG, о чем мы еще в свое время поговорим на наших страницах), поэтому приходится использовать и те английские буквы, которые имеют с русскими одинаковое написание, да и вообще ограничиваться русификацией только прописных букв. Стандартов здесь нет и быть не может, поскольку Вы и так ограничены в своем выборе. Проводите замену как вам удобно. Хотя должны сказать, что на заре Синклер-машин в СССР, где-то году в 84-м или 85-м появилась нехитрая программа ABC (неизвестного автора). Мы приводили ее в нашей разработке "Большие возможности Вашего Спектрума". Поскольку за столько лет эта программа стала общераспространенной, то может быть имеет смысл использовать заданные в ней соотношения:

"Q" - "Ы"	"D" - "Д"
"E" - "Э"	"F" - "Ф"
"R" - "Ь"	"G" - "Г"
"T" - "У"	"H" - "Ч"
"U" - "Ю"	"J" - "Ж"
"I" - "И"	"L" - "Л"
"O" - "У"	"C" - "Ц"
"P" - "П"	"B" - "Б"
"A" - "Я"	"N" - "Ш"
"S" - "Й"	"M" - "Щ"

Совсем другое дело, если Вы проводите русификацию полной заменой знакогенератора, здесь мы уже вторгаемся в поле действия международного стандарта ASCII и для начала маленькая историческая справка.

Набор символов компьютера содержит 256 символов, начиная с нулевого и до 255-го. Стандарт ASCII определяет символы с нулевого по 31-ый как управляющие коды и с 32-го по 127-ой как буквы английского алфавита, цифры и знаки препинания. Символы со 128-го по 255-ый этот стандарт не определяет, разрешая размещать в них графические символы или национальные шрифты. Здесь бы и разместить русский шрифт, как это делается, например в IBM-совместимых и других компьютерах, но для "Спектрума" из этого ничего не выйдет. Дело в том, что К. Синклер воспользовался предоставленной ему здесь свободой и уже разместил начиная со 128-го по 143-ий символы блочной графики, со 144-го по 164-ый отдал Вам в виде символов UDG, а со 165-го по 255-ый использовал для определения токенов ключевых слов, что дает Вам возможность набирать такие слова как LOAD одним нажатием клавиши. Так что в "Спектруме" LOAD - это не слово, а всего лишь один символ (239) и потому это токен, так же и SAVE - это символ 248. Подробности найдете в нашем "Справочнике по программированию в машинных кодах, Том 3 из трехтомника".

Вывод один, необходимо производить переключение шрифтов с английского на русский и наоборот. Неважно каким способом это сделано - программно или аппаратно введением дополнительного регистра, но если и русский и английский шрифты занимают одни и те же символы с 32-го по 127-ой, то на наш взгляд лучше всего делать замену так,

чтобы используемый Вами принтер тоже мог бы легко переключаться одной командой.

Наиболее распространенными для работы с 8-разрядными компьютерами в СССР являются принтеры ROBOTRON (ГДР) и D-100 (Польша). И в том и в другом реализован стандарт КОИ-7.

Для них соответствие клавиш должно быть таким (мы начинаем с 64-го кода, поскольку цифры и знаки препинания совпадают):

Код	Англ	Рус	Код	Англ	Рус
64	@	ю	96	фунт	Ю
65	A	а	97	a	А
66	B	б	98	b	Б
67	C	ц	99	c	Ц
68	D	д	100	d	Д
69	E	е	101	e	Е
70	F	ф	102	f	Ф
71	G	г	103	g	Г
72	H	х	104	h	Х
73	I	и	105	i	И
74	J	й	106	j	Й
75	K	к	107	k	К
76	L	л	106	l	Л
77	M	м	109	m	М
78	N	н	110	n	Н
79	O	о	111	o	О
80	P	п	112	p	П
81	Q	я	113	q	Я
82	R	р	114	r	Р
83	S	с	115	s	С
84	T	т	116	t	Т
85	U	у	117	u	У
86	V	ж	118	v	Ж
87	W	в	119	w	В
88	X	ь	120	x	Ь
89	Y	ы	121	y	Ы
90	Z	з	122	z	З
91	[ш	123	}	Ш
92	\	э	124		Э
93]	щ	125	{	Щ
94	↑	ч	126	~	Ч
95	-	ъ	127	"копирайт"	

Если предположить, что кому то из наших ассов удалось или удастся так переработать ПЗУ "Спектрума", что токены будут сделаны "прозрачными", т.е. они как бы есть, но для пользователя их как бы и нет, и удастся привязать русские литеры к верхней половине таблицы кодов ASCII, то тогда имеет смысл привязывать русские буквы к клавишам так, как это сделано на стандартной пишущей машинке.

Английский шрифт - "QWERTY..."

Русский шрифт - "ЙЦУКЕН..."

В этом случае можно было бы без перешивки ПЗУ или без программной перекодировки использовать принтеры EPSON-совместимые или IBM-совместимые, поскольку в них национальные шрифты лежат в верхней половине таблицы кодов.

Товарищ Максимов высказал еще пожелание внести ясность в вопрос стандартизации джойстиков, и он абсолютно прав, вопрос назрел, но поскольку "ИНФОРКОМ" аппаратными вопросами не занимается, то мы предпочли бы выслушать мнение специалистов, а потом его довести.

В одном из прошлых выпусков "ZX-РЕВЮ" мы сообщили, что нашим читателям нужны рекомендации по русификации программы "MASTERFILE 09". Семя упало в благодарную почву и несколько человек откликнулись. Спасибо им за это.

Должны сказать, что русификация программ бывает полной и неполной. При неполной встраивается русский шрифт и появляется возможность вводить информацию на русском языке, но программные сообщения, меню, запросы и т.п. продолжают печататься хоть и русскими буквами, но на английском языке. Это наиболее быстрый способ и к возникающему при этом неудобству активный пользователь быстро привыкает и перестает его замечать.

При полной русификации заменяются и все программные сообщения. Высшим шагом является такая переделка программы, когда после русификации работают правильно и такие режимы как сортировка по новому алфавиту.

В нашей почте есть несколько писем с идеями неполной русификации. Мы их сегодня дадим в качестве "неотложной помощи", есть также предложения о разработке статьи с рекомендациями по полной русификации. С этими авторами мы выходим на контакт и будем готовить материал для последующих выпусков.

Товарищ Козлов из г. Сургут любезно поделился собственным опытом русификации МАСТЕРФАЙЛа.

1. С помощью графического редактора "ART/STUDIO" он сформировал набор русских знаков и выгрузил его на ленту (768 байтов).

2. Загрузил его командой:

```
CLEAR 56559:LOAD""CODE 56560,768
```

3. Загрузил MASTERFILE 09:

```
LOAD "MF 09 LEER"
```

4. После загрузки MF вошел в БЕЙСИК-область из главного меню.

5. Работая в БЕЙСИКе ввел новые строки в программу "MF 09 LEER":

Строка 2:

```
POKE VAL "23606",VAL"0":
```

```
POKE VAL "23607",VAL"60":
```

```
GO TO VAL "1"
```

Здесь указатель набора символов (системная переменная CHARS) устанавливает набор знаков на "Спектрумовский" и выполняется возврат в "MF".

Строка 3:

```
POKE VAL "23606",VAL"240":
```

```
POKE VAL "23607",VAL"219":
```

```
GO TO VAL "1"
```

Здесь CHARS устанавливается на новый набор знаков и выполняется возврат в MF. Строка 4030.

Здесь необходимо поменять операнды во втором операторе SAVE.

```
4030 ... SAVE "MFcode" CODE VAL "56560",VAL"8976".
```

6. Выгружается измененная программа на ленту с помощью¹

Перед загрузкой программы для работы теперь надо давать прямую команду CLEAR 56559.

Адаптированная таким образом программа позволяет формировать файлы либо русским, либо латинским шрифтом, но не одновременно.

Работать с программой просто. После загрузки Вы выходите из Главного Меню в БЕЙСИК, а там даете прямую команду GO TO 2. Если работать будете с латинским шрифтом или GO TO 3, если с русским.

Примерно так же, используя БЕЙСИК-область произвел изменения и т. Соболев из Москвы. Поскольку он ввел русский знакогенератор через операторы READ и DATA, то его прием может пригодиться тем, у кого нет графического редактора "ARTSTUDIO", хотя конечно работать с ним приятнее, чем набивать 96 строк DATA, и его надо иметь, тем более

¹ В оригинале пропущена строка (Прим. OCR)

что генерация шрифта это всего лишь один процент его богатейших возможностей.

Вот что пишет товарищ Соболев. Он поставил задачу ввести русский шрифт не вторгаясь в машиннокодую часть программы и использовал для этого возможность МАСТЕРФАЙЛа исполнять БЕЙСИК-программу пользователя нажатием клавиши V в главном меню.

Сначала надо загрузить первый блок программы (mf lader) и прервать дальнейшую загрузку нажатием BREAK. После этого оператором CLEAR изменяется установка системной переменной RAMTOP: 10 CLEAR 56319: LOAD "" Сохраняется эта часть на ленте командой SAVE "mf lader" LINE 10. Запускаем измененный блок командой RUN. Начинает загружаться следующий блок MF 09 LEER. После его загрузки также прерываем дальнейшую работу командой BREAK и изменяем следующие строки программы:

```
4035 LOAD ""CODE: LOAD""CODE: GO TO VAL "1"
9000 INPUT "(RUS-LAT)-->(R/L?)"; n$: IF n$="r" OR n$="R" THEN POKE 23607,219: GO TO 9050
9010 IF n$="l" OR n$="L" THEN POKE 23607,60: GO TO 9050
9020 GO TO 9000
9050 GO TO 1
```

Измененную программу записываем на ленту:

```
SAVE "MF09LEER" LINE 4035
```

Теперь создадим новый знакогенератор с помощью приведенной ниже программы:

```
1 CLEAR 56319
5 LET A=10: LET B=11: LET C=12
10 LET D=13: LET E=14: LET F=15
12 LET x=56320
15 READ a$
20 FOR n=1 TO 15 STEP 2 30 LET W = VAL a$(n)
40 LET w=w*16: LET w=w + VAL a$(n+1)
50 POKE x,w: LET x=x+1
60 NEXT n
70 GO TO 15
```

(Пробелы в строках DATA даны для удобочитаемости. Вам их давать при наборе не надо).

100 DATA "0000 0000 0000 0000"	131 DATA "3C66 060C 1800 1800"
101 DATA "1818 1818 1800 1800"	132 DATA "0000 CEDB FBDB CE00"
102 DATA "6C6C 0000 0000 0000"	133 DATA "0000 3C06 3E66 3F00"
103 DATA "6C6C FE6C FE6C 6C00"	134 DATA "043C 607C 6666 3C00"
104 DATA "187E E07C 0EFC 3800"	135 DATA "0000 6666 6666 7F03"
105 DATA "00C6 CC18 3066 C600"	136 DATA "0000 7E66 6666 FFC3"
106 DATA "386C 3877 DECE 7B00"	137 DATA "0000 3C66 7E60 3E00"
107 DATA "1818 3000 0000 0000"	138 DATA "0018 7EDB DBDB 7E18"
108 DATA "000C 1818 1818 0C00"	139 DATA "0000 7C50 6060 6000"
109 DATA "0030 1818 1818 3000"	140 DATA "0000 C66C 386C C600"
110 DATA "0066 3CFF 3C66 0000"	141 DATA "0000 6666 6E76 6600"
111 DATA "0018 187E 1818 0000"	142 DATA "1300 6666 6E76 6600"
112 DATA "0000 0000 1818 3000"	143 DATA "0000 666C 7866 5600"
113 DATA "0000 007E 0000 0000"	144 DATA "0000 0E1E 3666 C600"
114 DATA "0000 0000 0018 1800"	145 DATA "0000 C6EE D6C6 C600"
115 DATA "0306 0C18 3060 C000"	146 DATA "0000 6666 7E66 6600"
116 DATA "7CC6 CED6 E6C6 7C00"	147 DATA "0000 3C66 6666 3C00"
117 DATA "1838 1818 1818 3C00"	148 DATA "0000 7E66 6666 6600"
118 DATA "3C66 061C 3066 7E00"	149 DATA "0000 3E66 3E36 6600"
119 DATA "3C66 061C 0666 3C00"	150 DATA "0000 7C66 7C60 6000"
120 DATA "0E1E 3666 7F06 0F00"	151 DATA "0000 3C66 6066 3C00"
121 DATA "7E60 7C06 0666 3C00"	152 DATA "0000 7E16 1816 1800"
122 DATA "1C30 607C 6666 3C00"	153 DATA "0000 6666 663E 063C"
123 DATA "7E66 060C 1618 1800"	154 DATA "0000 DB7E 187E DB00"
124 DATA "3C66 663C 6666 3C00"	155 DATA "0000 7C66 7C63 7E00"
125 DATA "3C66 663E 060C 3600"	156 DATA "0000 6060 7C66 7C00"
126 DATA "0018 1800 1618 0000"	157 DATA "0000 C6C6 F6DA F600"
127 DATA "0018 1800 1818 3000"	158 DATA "0000 3C66 0C66 3C00"
128 DATA "0C18 3060 3018 0C00"	159 DATA "0000 D6D6 D6D6 FE00"
129 DATA "0000 7E00 007E 0000"	160 DATA "0000 7C06 3E06 7C00"
130 DATA "6030 180C 1830 6000"	161 DATA "0000 D6D6 D6D6 FF03"

162 DATA "0000 6666 3E06 0600"	179 DATA "7CC6 C6C6 C6C6 7C00"
163 DATA "0000 0000 0000 000F"	180 DATA "FEC6 C6C6 C6C6 C600"
164 DATA "CEDB DBFB DBDE CE00"	181 DATA "7EC6 C67E 3666 C600"
165 DATA "0E1E 3666 FEC6 C600"	182 DATA "FCC6 C6FC C0C0 C000"
166 DATA "FCC0 C0FC C6C6 FC00"	183 DATA "7CC6 C0C0 C0C6 7C00"
167 DATA "CCCC CCCC CCCC FC06"	184 DATA "FC30 3030 3030 3000"
168 DATA "7E65 6666 6666 FFC3"	185 DATA "C6C6 C67E 06C6 7C00"
169 DATA "FCC0 C0F8 C0C0 FE00"	186 DATA "DBDB 7E18 7EDB DB00"
170 DATA "187E DBDB DB7E 1800"	187 DATA "F8CC CCFC C6C6 FC00"
171 DATA "7C60 6060 6060 6000"	188 DATA "C0C0 C0FC C6C6 FC00"
172 DATA "C356 3C18 3C66 C300"	189 DATA "C6C6 C6F6 DADA F600"
173 DATA "C6C6 CEDE F6E6 C600"	190 DATA "3C66 061C 0666 2C00"
174 DATA "D6C6 CEDE F6E6 C600"	191 DATA "D6D6 D6D6 D6D6 FE00"
175 DATA "C6CC D8F8 CCC6 C600"	192 DATA "7CC6 061E 06C6 7C00"
176 DATA "0E1E 3666 C6C6 C600"	193 DATA "D6D6 D6D6 D6D6 FF03"
177 DATA "C6EE FED6 C6C6 C600"	194 DATA "6666 663E 0606 0600"
176 DATA "C6C6 C6FE C6C6 C600"	195 DATA "3C42 99A1 A199 423C"

После запуска программы командой RUN компьютер сформирует новый знакогенератор и разместит его начиная с адреса 56320.

Выгрузите его на ленту прямой командой SAVE "RUS"CODE 56320,768

В заключение необходимо с помощью какой-либо копирующей программы собрать полученные блоки программы MASTERFILE09 в следующей последовательности: "mf lader" "MF 09 LEER" "RUS" "MF mcode"

Переключение на русский шрифт или наоборот выполняется выбором опции V в главном меню. В ответ на запрос "R/L?" Вам надо нажать клавишу R или L, в зависимости от того, какой шрифт Вам нужен.

Теперь небольшой финансовый вопрос. Многие читатели заметили в мартовском выпуске, что цена на подписку увеличилась на 20 руб. и прислали вопросы - надо ли производить доплату? А некоторые даже и перевели деньги!

Уважаемые товарищи! Если Вы подписались, то ничего вам доплачивать не нужно, что бы мы с ценой ни делали. Это имеет отношение только к новым подписчикам. Мы и дальше может быть будем отслеживать конъюнктуру рынка, но Вы страдать не должны. Единственное, когда Вы можете произвести доплату - это если хотите расширить условия подписки. Например подписавшись на игровое приложение Вы хотели бы получать выпуски целиком - вот тогда пожалуйста, но и то это дело добровольное. Мы выслали первые три выпуска всем в полном составе, независимо от того, кто на какой раздел подписался и дали возможность их оценить.

Теперь у нас будет головная боль как вернуть деньги тем людям, кто прислал по 20 рублей. Может быть Вы учтете их, если будете заказывать что-либо новое, хотя должны предупредить, что до сентября месяца мы новых изданий не планируем. Мы их готовим, но консервируем.

В заключение мы благодарим всех читателей, приславших письма для "ФОРУМа" или по выступлениям "ФОРУМа" и приносим извинения, что очень многим мы не в состоянии ответить, т.к. пишем очень много. Но читаем и принимаем к сведению абсолютно все.

До встречи в июне!

ИНФОРКОМ.

Маленькие хитрости

Продолжение. Начало на с. 51.

Сегодня мы продолжаем разговор; о простейших приемах, позволяющих экономить объем памяти, необходимой для написания вашей программы. Если Вы даже и не пишете еще¹

Еще большую экономию при работе с числами и другими типами переменных можно получить, если Вы знаете заранее, что изменяться в программе эти переменные не будут или если Вам эти изменения безразличны.

Когда Вы запускаете программу (RUN), то все переменные, которые есть в программе, копируются дополнительно в область переменных (VARS), т.е. они хранятся дважды. Почему бы их не хранить только в области переменных, конечно после того, как программа Вами отлажена и работает. Сделать это можно так. Попробуйте дать прямую команду

```
LET a=10
```

Прямая команда вводится без номера строки и выполняется компьютером немедленно. Начиная с этого момента компьютер "помнит", что а равно десяти, хотя такой строки в программе и нет. Проверьте его командой PRINT а. Убедились? Помнит компьютер это число потому, что значение а хранится в области переменных. Посмотрите, как оно там хранится:

```
10 LET p = PEEK 23627+ 256* PEEK 23628: FOR f=p TO p+17: PRINT f,PEEK f : NEXT f
```

Вот только запускать эту строку командой RUN уже нельзя потому, что RUN перед началом работы удаляет из памяти все переменные и очищает стек. Общее правило - если Вы применяете переменные, которые хранятся в памяти, но не хранятся в программе, т.е. не выставляются программой, то вместо RUN можно использовать только GO TO, в данном случае запуск произведете командой GO TO 10.

Второе золотое правило - всегда выгружайте такую программу только с номером строки автостарта - SAVE "имя" LINE n. Тогда после загрузки программа начнет работать со строки n так, как будто бы прошла команда GO TO n, а не RUN.

И, наконец, простейшая рекомендация - если какое-то число используется в программе многократно, то лучше обозвать его именем какой-либо переменной и дальше вызывать по этому имени. И последнее - имена переменных из одной буквы расходуют значительно меньше места, чем из нескольких, хотя это нарушает читаемость программ. Поэтому такое изменение надо вводить, когда программа уже полностью отлажена. При этом можно удалить из программы и строки REM.

¹ В оригинале пропущена строка (Прим. OCR)

ADVENTURE LESSONS

Окончание, начало на с. 35,55

9. Книги, записки, объявления и пр.

Их, конечно, если они встречаются по ходу игры, можно взять и рассмотреть (LOOK) и/или прочитать (READ).

READ BOOK - прочесть книгу

READ NOTE - прочесть записку

READ ADVERT - прочесть объявление

Почти всегда в них находится какая-то информация, жизненно важная для прохождения программы. Хотя иногда фирма, выпустившая программу, вставляет в них ради шуток информацию рекламного характера.

10. Музыкальные инструменты.

Особый случай могут представлять музыкальные ноты. Их можно, конечно, прочитать (READ MUSIC), но можно и сыграть на музыкальном инструменте.

PLAY MUSIC - сыграть музыку.

Можно и указать на чем сыграть, например, PLAY PIANO - сыграть на фортепиано.

Для духовых инструментов обычно применяется команда BLOW.

Например:

BLOW WHISTLE - подуть в свисток;

BLOW HORN - подуть в рожок.

Такой командой можно разбудить спящего персонажа, оживить статую, вызвать коня и т.п.

11. Растения.

Понятно, что если перед вами дерево, то на него можно взобраться - CLIMB TREE (если это не священное дерево). Но особый и более интересный случай представляют менее значительные растения. И здесь есть несколько вариантов.

Во-первых, слабенький зеленый росток, очевидно, нуждается в поливке. Если у Вас есть вода, то полейте его - POUR PLANT - полить росток; POUR WATER - полить водой.

Посмотрите, что из этого выйдет. Вполне возможно, что на Ваших глазах он вырастет, распустится и даст какие-то плоды, которые можно где-то использовать. Кстати, вообще надо помнить, что в работе с растениями самым интересным могут быть не растения, а их плоды.

Во-вторых, буйная трава (очевидно не нуждавшаяся в поливе) может что-то скрывать. Можно обыскать это место командой SEARCH. Можно попытаться здесь копнуть, чтобы понять отчего это вдруг вокруг все нормально, а здесь выросла буйная трава. Слабонервным лучше этого не делать. Бывает, откапываются останки ("PRICE OF MAGIC").

12. Веревка, канат (ROPE).

С ними все просто. Их можно потянуть (PULL ROPE), а также привязать - TIE ROPE. По ним можно и взобраться наверх - CLIMB ROPE.

13. Лестница - LADDER.

Совершенно очевидное применение - CLIMB LADDER.

В заключение этой главы мы должны сделать еще пару замечаний общего характера об использовании различных предметов.

Есть очень хорошая команда - USE (использовать), если программа ее понимает. Предположим, что Вы найдете предмет, который вам совершенно не известен, непонятен и что с ним делать Вы не представляете, например Вы нашли ABCDEF. Попробуйте его

применить командой USE:

USE ABCDEF

Если это ни к чему не привело, но Вы убедились, что команду USE программа понимает, то попробуйте повторять ее в разных местах. Может быть, где-то Вы и получите неожиданный результат.

Конечно, при использовании разных предметов надо руководствоваться в первую очередь здравым смыслом, но можно их использовать и не по назначению. Очень простой пример. Во многих программах встречаются лабиринты. Хорошо, если лабиринт обычный. Рисуя карту, Вы пройдете его рано или поздно. Но встречаются, и нередко, очень иррациональные лабиринты. Сделав по нему хотя бы шаг, а затем вернувшись на шаг назад, Вы уже можете не попасть в исходную точку. Там могут быть и коридоры с односторонней проходимостью и переходы на другие этажи. В общем, возможны совершенно нелогичные вещи. Чтобы осилить такой лабиринт, надо использовать имеющиеся у Вас предметы. Бросили предмет, сделали шаг вперед, потом назад и посмотрите (LOOK). Если Ваш предмет лежит там же, то здесь пока все в порядке. Так шаг за шагом "раскалывают" любые лабиринты, а они, как показывает практика, совсем не так страшны, как кажется на первый взгляд.

И последнее. Кажущееся многообразие предметов и действий, которые с ними можно совершить, не очень должно Вас пугать. Дело в том, что хорошая программа так или иначе намекает Вам о том, чего она от Вас ждет. Как догадаться, что в заброшенной часовне надо поднять подгнившую половую доску, чтобы обнаружить тайник? - так ведь Вам говорили, что когда Вы вошли в дверь, доска скрипела.

Когда, закончив игру, Вы пройдете ее еще раз, то увидите, что на все (даже самые головоломные) действия в игре были подсказки и намеки. Ну, а уж если этого не было, то знайте, что именно здесь и проходит граница между великими шедеврами и посредственной подделкой.

Поскольку на рынке программного обеспечения есть несколько программ, позволяющих создавать адвентюрные игры автоматизированным путем (в порядке их появления - QUILL, G.A.C. - GRAPHIC ADVENTURE CREATOR и P.A.W.S. - PROFESSIONAL ADVENTURE WRITING SYSTEM), то неискушенные программисты часто используют их для быстрого создания неполноценной продукции, которая тысячами экземпляров заваливает фирмы, выпускающие программное обеспечение и редакции Синклер-журналов. Хотя, конечно, у нас нет и мысли опорочить программы QUILL, G.A.C. и P.A.W.S. Они являются прекрасным инструментальным средством и, конечно, они дали возможность многим начинающим программистам попробовать свои силы, создать что-то свое и благодаря им некоторые смогли впоследствии достичь высоких результатов.

Тем не менее надо помнить, что если с их помощью не требуется быть программистом, чтобы создать шедевр, то зато надо безусловно иметь писательский талант и нестандартную фантазию, а и то и другое требует многих месяцев и лет напряженной тренировки, так что сразу шедевр не получается.

5. Взаимодействие с другими персонажами.

Правильная работа с другими персонажами, также как и работа с объектами, может очень облегчить прохождение программы, в то время как неправильная может испортить все впечатления от программы.

Во первых, всех встреченных Вами персонажей (неважно людей или животных) можно условно разделить на четыре категории.

1. Друзья.
2. Враги.
3. Нейтральные.
4. Неопределенные.

Если с первыми тремя категориями все ясно, то к четвертой мы (то есть "ИНФОРКОМ") относим таких, которые по ходу развития событий могут стать как друзьями, так и врагами - все зависит от того, как Вы будете себя с ними вести.

Итак, рассмотрим основные команды, которые используются при работе с персонажами.

1. Возьмите себе за правило при встрече нового персонажа здороваться. Делается это с помощью команды SAY (сказать). Например, SAY "HELLO" - сказать "ПРИВЕТ".

Это не поможет, если встреченный персонаж Ваш убежденный враг, но в прочих случаях это позволит Вам получить ответ, а из него какой-то намек на то, в каком направлении вести дальнейшую беседу.

2. Предположим, контакт установлен и на Ваше приветствие встреченный персонаж не откусил Вам голову, а ответил что-нибудь или хотя бы пробурчал, следующий шаг - вытрясти из него информацию или еще что-нибудь, наиболее часто применяется команда:

ASK ABOUT ... - спросить о ...

Предположим, Вам надо переправиться через реку и вам встретился лодочник. После приветствий спросите его о лодке:

ASK ABOUT BOAT

Он может прямо запросить оплату, может сразу перевезти Вас, а может и сказать, что некогда ему здесь сидеть и отвечать на глупые вопросы.

Также можно спрашивать и об оружии у оружейника, о хлебе у булочника и так далее.

3. Как правило, за услуги надо платить. Платят чаще всего деньгами. Соответствующие команды - PAY (платить) или GIVE (давать). Например,

PAY MONEY - заплатить деньги.

GIVE COIN - дать монету.

Платить можно не только деньгами, можно расположить персонаж к себе с помощью подарка:

GIVE GOLD - дать золото

GIVE BOOK - дать книгу.

Самое интересное, что в мире сказочных приключений действуют совсем иные соотношения ценностей. Правильно подаренный рваный башмак может сыграть большую роль, чем мешок с золотом.

4. Иные взаимоотношения с врагами. С ними можно либо сражаться, либо убегать.

Убегать просто - RUN.

Сражаться - FIGHT.

Можно еще употребить команду ATTACK – атаковать.

В некоторых играх бывают ситуации, когда разных врагов лучше атаковать разным оружием. Например, против бегемота лучше применить топор (AXE), а против вампира - вязанку чеснока (GARLIC), в этом случае в команде указывают чем атаковать соперника:

ATTACK WITH DAGGER - кинжалом

ATTACK WITH SWORD - мечом

ATTACK WITH AXE - топором и т.п.

5. В некоторых играх общаться с другими персонажами можно и по телефону.

Если Вы хотите позвонить по телефону, можете использовать для начала команду USE PHONE - воспользоваться телефоном, это если из контекста программы само собой разумеется, кому Вы звоните.

Бывает, что по ходу программы Вам подсказывают какой-либо номер (например, Вы его прочитали в найденной в собственном кармане записной книжке). Набор номера выполняется командой DIAL.

Например:

DIAL NUMBER - набрать номер. Сам номер может потом набираться по запросу от программы.

Если Вам известно, кому Вы хотите позвонить конкретно, то можете проверить команду CALL. В принципе CALL по-английски это звать, вызывать, но позвонить по телефону это тоже CALL. Например, CALL PRESIDENT позвонить президенту.

6. Использование отгрузки программы на ленту.

Нормальная программа, предназначенная для длительной работы, обязана иметь возможность отгрузки состояния игры на ленту. Обычно это делается командой SAVE.

Естественное назначение этой команды - отложить незаконченную игру до следующего сеанса, в то же время, в адвентюрных играх эту команду применяют просто для закрепления достигнутого успеха.

Всегда, когда идете на рискованное дело, делайте SAVE, это позволит Вам в случае гибели не начинать все сначала, а загрузить отложенное состояние и сэкономить на этом много времени.

Адвентюрная программа, которая не имеет опции SAVE - не программа, имеющая право считаться адвентюрной.

7. Полезные практические приемы.

Здесь мы рассмотрим некоторые моменты, не вошедшие в основной текст нашей статьи по причине либо их нетипичности, либо нетрадиционности.

1. Есть очень хорошая команда - BOM. Введена в практику фирмой "LEVEL 9". Это сокращение от BACK ONE MOVE - ОДИН ХОД НАЗАД. Она позволяет взять один ход назад, чтобы избежать нежелательного исхода. К сожалению, она редко применяется в играх, но если кто-то из наших читателей, наигравшись досыта, начнет сам разрабатывать такие игры, то он должен знать, что за наличие такой команды в игре ему многократно скажут спасибо.

2. В программах, связанных с магией и чародейством, вам может быть придется иметь дело с заклинаниями. Заклинание по-английски SPELL. Сами заклинания и их назначение Вам придется найти по ходу игры. Они могут быть в старинных книгах, их можно купить, наконец они могут быть выцарапаны на стене, вышиты на ковре или вырезаны на картинной раме.

Произносятся заклинания командой CAST.

Приведем пример, Вы нашли на задней стенке шкафа надпись FLY (летать). Полагая, что это заклинание, Вы даете: CAST FLY или SAY FLY, после чего оказывается, что Вы можете теперь летать.

Диалог может быть и таким:

> CAST SPELL - произнести заклинание

- WHICH SPELL? - какое заклинание

> DIE - умереть

- O.K.

После этого Ваш соперник погибает.

3. При работе с программой никогда не используйте ругательных выражений, примеры мы по понятным причинам не приводим.

Умная программа их понимает. В лучшем случае она скажет, что джентльмены так не выражаются, а бывает сбрасывается совсем. Если даже она и не сбрасывается, то как знать, может быть в одну из ячеек памяти пошла информация о том, что Вы неприятный партнер и на что это в дальнейшем повлияет?

4. Если в работе с программой Вы зашли в безнадежный тупик, а поэкспериментировать в общем-то хочется, то дадим Вам несколько советов.

Первое - попробуйте с помощью дисассемблера (например MONITOR 48 или MONITOR 16) пробежаться по машинному коду программы в поисках ее словаря. Если он не зашифрован, то это поможет Вам установить список глаголов и существительных. Глаголы - это те команды, которые программа понимает, а существительные - возможные объекты, которые могут встретиться в ходе игры.

Есть интересный топорный прием, позволяющий поэкспериментировать с программой. Отгрузите игру в каком-то месте на ленту. Теперь снова подзагрузите ее, но при загрузке отложенного состояния прервите загрузку, остановив магнитофон. В девяти случаях из десяти загрузка не пройдет. Но иногда эффект бывает поразительным. Все предметы в программе путаются. Вы можете приобрести что-то, о чем и не знали, за

стойкой бара вместо бармена может оказаться инопланетянин, до которого еще надо было бы лететь на другую планету. У Вас может оказаться баснословное количество денег и теперь Вы можете купить что угодно и все испытать, все попробовать. Как Вы понимаете, результат непредсказуем, но интересен, тем более, что не стоит особого труда, если получится.

Более тонкая химия - это взять отгруженный блок, а он всегда очень короткий, загрузить его в дисассемблер и поменять в нем несколько байтов, а затем посмотреть, что получилось. Правда, почти всегда этот блок бесхэддерный (без заголовка), и потому в дисассемблер загружаться не будет, но для такого дела можете "пришить" к нему заголовок от какого угодно другого блока или сделайте его сами. Дайте команду

```
SAVE "abcd" CODE m,n
```

m, n - почти что угодно. Например, m - 30000, а n - 500.

Когда пройдет первая порция широких полос и маленькая пачка (17 байтов) узких, выключите магнитофон, не выгружая всего остального - вот Вам и "хэддер"

Сшивание делайте в любом копировщике.

Кстати, копировщик COPY-COPY может загружать и бесхэддерные блоки и просматривать их командой LIST и вносить в них изменения командой POKE. Впрочем, мы об этом писали в нашей базовой разработке "Большие возможности Вашего Спектрума" еще полтора года назад и наши старые читатели это знают.

ЗАКЛЮЧЕНИЕ.

Мы обещали, что уделив по одной странице в месяц, беремся обучить Вас за полгода. Уделив по две страницы, мы уложились за три месяца. Конечно не все команды (глаголы), которые понимают текстовые программы, мы сюда включили, но примерно 85% основных - наверняка. Начинать Вы уже можете. А если кто знает еще хорошие широко встречающиеся команды - пишите, мы напечатаем, ведь могли же мы что-то упустить.

Чтобы мы хотели получить взамен?

Только одно - заинтересованность. Чтобы те, кто сегодня начинают работать со "Спектрумом" не бежали как от чумы от текстовых программ. Помните, что через три года общения с компьютером для 75% пользователей этот жанр становится основным. Вот облегчить этот путь мы и постарались.

Вы приобретете:

1. Практическое применение знаниям английского языка.
2. Полезный и увлекательный досуг.
3. Новых друзей и знакомых.

Почти никому не удастся пройти мощную игру в одиночку. Только обмениваясь достижениями Вы сможете преодолеть нормальную программу средней силы.

GRAPHIC ADVENTURE CREATOR

В мире программного обеспечения время от времени выпускаются программы, вносящие значительный вклад в написание игровых программ. Одной из таких программ была программа QUILL. Она открыла возможность самим создавать адвентюрные игры для тех, у кого хорошо развито воображение, есть терпение и совершенно нет навыков в программировании.

Конечно, большинство программ, созданных в системе QUILL, оставляют желать лучшего, но самые лучшие из них (например, TOWER OF DISPAIR) показывают каких успехов можно достичь.

Несмотря на то, что программа QUILL очень долго доминировала на рынке, у нее все-таки есть свои ограничения, и программа GAC (Graphic Adventure Creator), выпущенная фирмой INCENTIVE SOFTWARE явилась для нее серьезным конкурентом. Она предоставляет следующие дополнительные возможности:

- анализ сложных предложений;
- многозначные команды;
- компрессия текста;
- широкий спектр графических возможностей.

Пакет GAC включает в себя кроме самой программы еще и небольшой полезный файл Quickstart, являющийся как бы образцовой адвентюрной микроигрой.

Конечно, как и положено серьезной системе, к пакету должно прилагаться объемистое руководство по работе с ним, которого у нас в "Инфорконе" к сожалению нет, но если есть у кого-то из наших читателей, мы с радостью бы его приобрели. Поэтому нам приходится сегодня ограничиваться лишь обзорной статьей на эту тему, тем более, что многие читатели просят именно обзорных статей, для расширения общего кругозора по интересным программным продуктам.

Наше общение с любителями адвентюрных программ, показало, что при наличии упорства с этой программой можно разобраться и без учебника, а та информация, которую мы здесь дадим, поможет сделать первый шаг. Мы будем рады получать письма от тех, кто пробьется в этом направлении через первые трудности и поможет другим, ценным было бы также исследование возможности русификации этой программы. Общие соображения говорят о том, что это не должно быть очень трудно, поскольку в программе где-то есть свой знакогенератор.

После загрузки программы Вам выдается обширное меню различных опций. Самое мудрое, что здесь можно сделать - это загрузить прилагаемый файл "Quickstart".

Это даст полезную базу, от которой можно развивать работу. В этом файле уже есть небольшой словарь, включающий в себя такие совершенно необходимые в любой текстовой программе команды, как GET, LOOK, EXAMINE, INVENTORY и т.п. Там же есть и небольшая система программных сообщений типа:

YOU CAN'T - Вы не можете.

WHAT NOW? - Что дальше?

YOU ARE CARRYING... - Вы несете...

Конечно их можно переделывать как хотите.

Словарь входит в программу в трех отдельных файлах:

- VERBS (глаголы),
- NOUNS (существительные),
- ADVERBS (наречия).

Процесс заполнения и редактирования словаря изумительно прост. Каждому глаголу, существительному или наречию присваивается номер от 1 до 255. Программа может теперь элементарно работать и с синонимами, поскольку Вы можете разным словам, имеющим одинаковое значение, присвоить и одинаковые номера.

Способность работать с наречиями существенно отличает GAC от QUILL, поскольку теперь Вы можете различать такие команды как:

EXAMINE BOX - проверить ящик и

EXAMINE BOX CAREFULLY - тщательно проверить ящик.

Ввод текстов сообщений и описаний обстановки тоже доставляет удовольствие. Вводя текст Вы можете не думать, как он расположится на экране, не будет ли разрыва в слове при переносе. Просто набирайте текст, и все. Программа сама все сделает за Вас. Правда, программа немного путается со знаками препинания. Считается общепринятым после знака препинания давать пробел, поэтому если знак препинания оказывается последним на строке, то в начале следующей строки этот пробел давать не надо. Здесь программа этого не учитывает и левый край текста может получиться неровным. На каждом шагу работы программа дает вам необходимые запросы, что очень упрощает работу. Так, например, когда Вы закончите описание какой-либо локации, программа запросит от Вас ввода соединений этой локации с другими. Например, если из данной точки можно пойти на север и попасть в пункт номер 15, то Вы вводите:

NORTH 15

Таким простым путем задается карта игры. Изменения в процессе работы вводятся тоже очень легко.

ЛОГИКА.

Когда введен текст, описания локаций, сообщения, карта и имена объектов (существительные), можно приступать к разработке логики игры. Именно здесь и выявляется вся сила этой программы и именно здесь и наступают основные трудности для Вас.

Существуют три типа условных конструкций, для каждого типа есть свой файл и доступ к ним открывается из главного меню.

Условия высшего приоритета (High Priority Condition) проверяются, перед тем как играющий вводит команду.

Условия низшего приоритета, проверяются после того, как игрок команду подал.

Локальные условия относятся к тем специфическим действиям, которые могут происходить в данной локации, в которой сейчас находится игрок.

Логические операторы, которые могут быть при этом созданы и рассмотрены, очень широки и многообразны. Достаточно сказать, что Вы имеете возможность задавать и проверять 255 флагов (маркеров), с помощью которых программа отслеживает важнейшие события, такие например, как открыта дверь или закрыта. С их же помощью можно оценивать и более сложные вещи:

- сколько весит объект;
- есть он у Вас в наличии или нет;
- есть он в заданной комнате или нет и т.п.

Нижеприведенный пример служит для иллюстрации языка программы.

Предположим, что где-то по ходу игры Вам может встретиться гангстер NICK, вооруженный пистолетом (GUN). В этом случае программа в ответ на команду игрока:

TELL NICK TO GIVE ME THE GUN

(сказать Нику, чтобы отдал пистолет) программа должна, во-первых, дать нужное ответное сообщение и, во-вторых, добавить пистолет к списку объектов, которыми Вы владеете.

Условие в программе будет выглядеть так:

IF (VERB 24 AND NOUN 3 AND NOUN 5 AND RES? 6 AND HERE 3) SET 6 MESS 6 5 TO 3
GET 5 END

Если это грубо перевести на русский язык, то получится следующее:

Если команда содержит глагол (VERB) номер 24 (tell) и существительное номер 3 (Nick), а также существительное номер 5 (GUN) и если при этом флаг номер 6 выключен (RESET) и Ник находится здесь же рядом с Вами (HERE 3), то включить флаг 6 (чтобы можно было забрать пистолет только один раз), напечатать нужное сообщение (MESSAGE 6), образовать пистолет как объект в данной локации (5 TO 3) и взять его, добавив к списку предметов, которыми владеет игрок (GET 3).

Если Вы зададите таким образом логику работы по отдельным командам типа

UNLOCK THE DOOR - отпереть дверь

OPEN DOOR - открыть дверь

GO INTO DOOR - войти в дверь,

то и многозначные команды типа: UNLOCK THE DOOR, OPEN IT AND GO IN - будет тоже выполняться!

Обратите внимание, что программа понимает местоимение IT и использует его вместо последнего встречавшегося существительного DOOR.

Чего правда не хватает программе - это она не умеет работать с командами DROP ALL и TAKE ALL.

ГРАФИКА.

Адвентюрная игра может быть сделана с графическими иллюстрациями, любую картинку из загруженных Вы можете включить и выключить в любом месте игры по желанию.

Картинка изображается в окне, занимающем верхние 2/3 экрана, что пожалуй многовато, поскольку внизу остается недостаточно места для детального описания обстановки, и если описание слишком длинное, то оно начинает скроллироваться до того, как игрок успеет его прочитать.

Программа может рисовать эллипсы, прямоугольники, линии, точки. Можно заполнять фигуры гладким цветом или несложной текстурой. Есть очень хорошие редакторские возможности. Одним словом, создать можете любую иллюстрацию из тех, какие Вам позволят создать собственные таланты. Приличная картинка (не хуже, чем те, которые Вы можете увидеть в программе HOBBIT) занимает примерно 0.5 К.

Замечательной способностью программы является возможность "сливать" (MERGE) изображения. Поэтому Вы можете один раз изобразить "стандартное" дерево и всякий раз, когда надо, вызывать и накладывать его на картинку. То же самое часто используется для стилизованных орнаментов вокруг картинки.

Единственное ограничение - это то, что, как дерево нарисовано на экране, так оно всегда и будет изображаться - в одном и том же месте. Передвигать рисунки программа не умеет.

Заканчивая обзор этой замечательной программы, нам бы хотелось обратить внимание наших читателей на одно существенное обстоятельство.

Мы все время говорили о программе GAC как об инструментальном средстве для создания игр, в данном случае текстовых. Совсем не надо иметь богатую фантазию для того, чтобы сообразить, что это почти готовая система для создания обучающих программ, способных и дать материал с иллюстрацией и поставить вопрос и проанализировать ответ, а затем сделать вывод о том, чтобы похвалить/ пожурить/ подсказать.

Причем работать с этой системой (создавать учебные комплексы) может преподаватель, хорошо знающий свой предмет, но не знающий программирования. Вы же, конвертировав GAC и сделав из него систему, можете наоборот не быть преподавателем и не знать никаких предметов. Вот Вам и путь к быстрому решению проблем компьютеризации образования. Нам часто пишут молодые люди, освоившие или осваивающие программирование и спрашивают совета куда приложить свои усилия. Возьмите GAC, разберитесь с ним так, чтобы ориентироваться в этой программе, как в своей, замените генератор шрифта на русский, адаптируйте GAC под другой круг задач, напишите достойную инструкцию по работе с ним, и Вы сделаете такой вклад в общее дело, который конечно же не останется незамеченным. Даже если у Вас из этого ничего не получится, Вы приобретете опыт и при этом у Вас появятся новые идеи. Дерзайте, и никакие Ваши усилия не пропадут даром, а мы дадим рекламу Вашим достижениям на страницах "ZX-РЕВЮ".

SCRABBLE



Эта игра должна быть Вам хорошо известна. Поскольку впервые она была представлена в журнале "Наука и жизнь" еще в шестидесятых годах под названием "ЭРУДИТ", а в последующие годы было выпущено множество настольных вариантов.

Игра вполне оправдывает свое название, поскольку от играющего требуется умение комбинационно мыслить и иметь солидный запас слов.

Компьютерная версия игры, рассчитанная на тех, кто знает или осваивает английский язык, имеет особенную прелесть, поскольку вдвойне будет способствовать повышению Вашей эрудиции.

Вступление.

Эта инструкция состоит из двух разделов. В первом описаны правила игры "SCRABBLE", а во втором - описание, как играть в нее на компьютере.

I. Правила игры.

SCRABBLE - это буквенная игра. Количество играющих - от 1 до 4 человек. Цель игры состоит в построении взаимопересекающихся слов наподобие кроссворда. Игра ведется на специальной игровой доске, где размещаются фишки с буквами, имеющими различную ценность.

Каждый игрок стремится получить максимальное количество очков на своем ходе и должен для этого учитывать неодинаковую ценность различных букв, а также использовать премиальные клетки игровой доски (они выполнены цветными). Итоговый счет может быть в диапазоне от 500 до 700 очков, в зависимости от мастерства играющих.

Игра.

Каждый игрок в начале игры имеет на руках 7 фишек. После каждого хода он получает дополнительно такое количество фишек, какое необходимо для дополнения их общего количества, находящихся у него на руках, до семи.

1. Первый играющий берет две или более букв, составляющих какое-либо слово и размещает их так, чтобы слово проходило через центр доски и читалось слева направо или сверху вниз.

Разнесение слов по диагонали не разрешается.

2. Игрок заканчивает свой ход подсчетом очков и объявлением счета, после чего он подбирает количество фишек до семи. (Конечно в компьютерной версии все за него сделает компьютер).

3. Второй игрок и все последующие игроки добавляют одну или более букв к уже выставленным на доску так, чтобы они составили слово. Все фишки, выставленные в один ход, должны быть выстроены в один вертикальный или горизонтальный ряд. Они должны составлять одно законченное слово. Если, в то же время, они соприкасаются с фишками в соседних рядах, то и с ними они должны составлять слова (по принципу кроссворда). Игрок получает очки за все построенные или измененные в результате хода слова.

4. Новые слова могут быть составлены:

- добавлением одной или более букв к слову или к буквам уже имеющимся на доске.
- размещением слова под прямым углом к уже имеющемуся слову. При этом новое слово должно использовать хотя бы одну букву ранее имевшегося слова.
- размещением слова параллельно имевшемуся, но при этом соседствующие клетки также должны составлять слова в перекрестном направлении.

5. Буквы после их установки перемещать нельзя.

6. В наборе букв есть две пустые фишки, которые могут быть использованы игроком по желанию вместо любой буквы. Выставляя пустую фишку, игрок должен указать какой буквой он ее назначает, после чего это назначение остается на все время игры и никем из играющих не может быть переопределено.

7. Каждый игрок может вместо хода произвести замену любого числа имеющихся у него букв. Сначала он сбрасывает ненужные ему фишки, затем вытягивает новые, после чего сброшенные фишки смешиваются с содержимым "котла". Дальнейший ход он делает в порядке очереди.

8. Разрешенными считаются словарные слова, за исключением имен собственных, слов иностранных языков (не английского), а также слов с апострофом, в случае сомнений в правописании слова справляйтесь по словарю. Любое слово может быть оспорено до того, как следующий игрок сделает свой ход. Если слово признается недействительным, игрок обязан забрать назад свои фишки и теряет ход.

9. Игра продолжается до тех пор, пока все фишки не будут вытащены из "котла" и один из играющих не выставит все имеющиеся у него на руках буквы или до тех пор, когда дальнейшие ходы станут невозможными для всех играющих.

Счет.

10. Ценность букв неодинакова. Чем чаще данная буква встречается в словах английского языка, тем меньше ее ценность, т. е. тем меньше очков Вы получите за то, что нашли способ ее применения. Таблицу расценок букв для справки Вы всегда сможете вызвать во время игры, кроме этого, Вы можете вызвать и таблицу, показывающую распределение фишек по их количеству. Обратите внимание на то, что ценность пустой фишки равняется нулю.

11. Счет подсчитывается после каждого хода. Очки образуются суммированием оценок всех букв, входящих в построенное или измененное слово плюс премиальные очки за размещение фишек на премиальных полях.

12. Поля премиальных букв:

- голубые поля удваивают ценность букв, помещенных на них;
- синие поля утраивают ценность помещенных на них букв;

13. Поля премиальных слов:

- зеленые поля удваивают ценность всего слова, проходящего через них;
- красные поля, соответственно, ее утраивают;
- если в слове есть премиальные буквы, то сначала прибавляются премиальные очки, а только затем ценность слова удваивается или утраивается;
- если слово проходит через два зеленых или красных поля, то оно удваивается или утраивается дважды (т. е. увеличивается в 1 или 9 раз соответственно);
- обратите внимание на то, что центральное поле имеет зеленый цвет и следовательно, цена первого слова двойная.

14. Премиальные поля действуют только на тот ход, в котором они используются. В последующих ходах ценность букв и слов не увеличивается.

15. Если пустая фишка попадает на поле премиального слова, то ценность слова удваивается или утраивается, несмотря на то, что ценность пустой фишки равна нулю.

16. Если в результате хода образуются два или более слов, то их ценность суммируется. Ценность общей буквы (даже если она размещена на премиальном поле) учитывается для каждого слова независимо.

17. Если игрок в один ход выставит все свои семь фишек, то он получает дополнительные призовые 50 очков.

18. В конце игры производится окончательный подсчет очков. Финальный счет для каждого из игроков уменьшается на величину, равную сумме цен фишек, оставшихся несыгранными у него на руках. Если один из играющих израсходовал все свои фишки, то его очки увеличиваются на величину, равную сумме очков всех фишек, оставшихся несыгранными у других партнеров.

II. Игра с компьютером.

После загрузки программы Вам надо ответить на ряд вопросов, чтобы определить условия игры.

1. Какой у Вас телевизор? (цветной - нажмите С, а если черно-белый - нажмите В.)

2. Не хотите ли Вы загрузить предварительно отложенную игру? Да - "Y", нет - "N".

3. Выберите количество играющих: от 1 до 4.

Теперь ответьте на вопросы для каждого играющего отдельно.

4. Не хотите ли Вы, чтобы в качестве этого играющего выступал компьютер? Да - "Y", нет - "N".

5. Если "Да", то на каком уровне сложности должен играть компьютер (нажмите от 1 до 4. Уровень 4 - самый сильный).

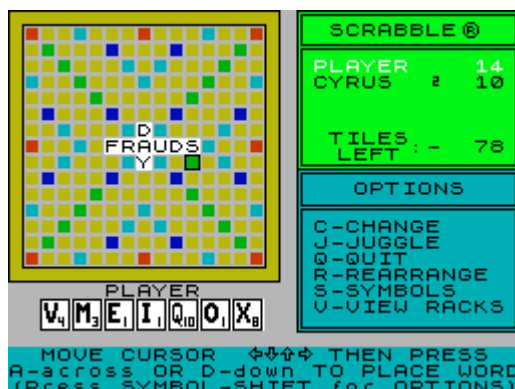
6. Введите имя игрока (максимум 8 букв для людей. 7 букв для компьютерных игроков).

И, наконец:

7. Хотите ли Вы видеть как компьютер думает? Нажмите "Y" или "N". Если Вы выберете "N", то пробные ходы компьютера не будут изображаться на экране, а также не будут изображены фишки, имеющиеся у него на руках.

8. Надо ли Вам, чтобы компьютер делал паузу между своими ходами? (Y/N)

После этого компьютер определит (случайным образом) кто из играющих будет ходить первым, нарисует игровую доску и раздаст (совершенно случайным образом) фишки.



На экране будет изображена доска, справа от нее написаны имена игроков и счет, а также список возможных действий.

Там же изображаются фишки игрока, который в данный момент должен делать ход.

Главное меню.

V - VIEW RACKS

Этот выбор позволяет увидеть фишки, имеющиеся на руках у всех игроков. Возврат в главное меню выполняется одновременным нажатием SYMBOL SHIFT и 0.

S - SYMBOLS

По этой опции вызывается другой экран, на котором указаны премиальные поля. Возврат через SYMBOL SHIFT и 0.

R - REARRANGE - позволяет переместить фишки, находящиеся у Вас на руках. После нажатия "R" просто наберите их на клавиатуре в том порядке, в каком Вы хотите их видеть.

C - CHANGE - Вы можете поменять любые (или все) свои фишки вместо хода. После нажатия "C" Вы можете нажать на клавиатуре те буквы, которые хотите поменять. Если ошибетесь, пользуйтесь клавишей DELETE. В конце нажмите ENTER. Если хотите просто пропустить ход, нажмите ENTER без набора букв.

J - JUGGLE - это инструкция для компьютера - перемешать фишки ходящего игрока в случайном порядке.

T - TILE VALUES - по этой команде изображается алфавит с указанием ценности каждой буквы. Возврат через SYMBOL SHIFT и O.

D - DISTRIBUTION - изображает доступное количество каждой из букв в начале игры.

Q - QUIT - прерывание игры. После этого можно начать новую игру или выгрузить игру в промежуточном состоянии на чистую ленту.

Как делать ход.

После того, как Вы придумали слово, Вам надо указать компьютеру, где его разместить.

Перемещайте курсор по доске с помощью курсорных клавиш 5,6,7,8 к полю, с которого начнется слово. Теперь нажмите "A", если слово пойдет слева направо, или "D", если слово пойдет сверху вниз. Далее выстраивайте свое слово простым нажатием клавиш на клавиатуре. Когда закончите, нажмите ENTER.

Если ход был выполнен по всем правилам, компьютер подсчитает очки и спросит, устраивает ли Вас это или попробуете другой ход. Ответьте "Y" или "N". Если Вы ответите "Y", компьютер выдает вам новые фишки взамен сыгранных. Теперь ход перейдет к следующему играющему.

Ход пустой фишкой.

Чтобы пойти пустой фишкой, сначала нажмите пробел (SPACE). После этого нажмите любую букву, которая Вам нужна, на месте пустой фишки.

Запрещенные ходы.

Если Вы сделаете ошибку или попытаетесь сыграть не по правилам или допустите грубую ошибку, компьютер сообщит об этом и ход можно будет повторить.

Оспаривание хода.

После того, как Вы выставите слово, компьютер проверит его по своему словарю, в котором более 11 тыс. слов. Если в нем такого слова нет, то он попросит подтвердить правильность этого слова.

Да - "Y". Нет - "N". Если Вы нажмете "Y", то компьютер Вам поверит и слово будет засчитано.

Проверку может сделать и другой играющий. Он может запретить Вам нажимать "Y", если Вы ошиблись.

Вам проверять компьютер не надо. Его словарь был проверен экспертами.

Окончание игры.

В конце игры компьютер подведет итог с учетом того, какие фишки остались на руках у игроков.

Откладывание игры.

Незаконченную игру можно отложить. Воспользуйтесь выбором QUIT в главном меню, затем пользуйтесь указаниями на экране. Не забудьте о необходимости использования чистой кассеты. Выгружается не вся программа, а только блок данных.

Загрузка отложенной программы

Отложенную игру можно дозагрузить после загрузки главной программы. Можно также загрузить отложенную игру после команды QUIT.

Стратегия.

Компьютер в игре руководствуется определенной стратегией. Так, например, даже на самом сильном уровне (4) он может не выставлять максимально возможное слово для того, чтобы сберечь какую-либо полезную букву, например "S". В то же время, компьютер не применяет принципов негативной стратегии (например, он старается избегать открывания

для Вас премиальных полей). Он выиграет у Вас и без этих ухищрений.

Словарь

Словарь компьютера содержит более 11000 слов. Это беспримерное достижение для сравнительно малой машины. Для того, чтобы он мог помнить максимальное количество высокоценных слов, многие простые слова, встречающиеся очень часто в повседневной жизни и хорошо известные Вам, были опущены. Поэтому пожалуйста не смущайтесь, если компьютер будет оспаривать очевидное слово. Просто подтвердите его нажатием клавиши "Y".

FORUM

Мы рады сообщить нашим читателям, что объявленная нами атака на игру SABOTEUR-2, производства фирмы DURELL вылилась в самый настоящий коллективный штурм, о результатах которого мы Вам сейчас и расскажем. Мы получили десятки писем от энтузиастов и поклонников этой игры и можем сказать, что практически все они написаны с большим знанием дела. Приятно сознавать, что у сложной зарубежной программы нет шансов противостоять дружному натиску наших экспертов, но самое приятное - это то, что эти письма дополняют друг друга и сами эксперты из сегодняшнего обзора найдут для себя что-то новенькое, что не удалось обнаружить им и что нашли другие.

Лабиринт состоит из трех соединенных подвалами зданий. Он содержит три лифта и пульта управления к ним. Главное помещение лабиринта - туннель, проходящий две трети лабиринта и выходящий наружу на уровне земли.

По всем этажам лабиринта разбросаны ящики, в которых можно найти ключи, кинжалы, шурикены (метательные звезды ниндзя), а также фрагменты перфолент.

По комнатам зданий ходят охранники с пантерами. Они хорошо вооружены. У некоторых есть даже огнеметы. Для борьбы с ними Вы можете использовать найденное оружие или применять приемы каратэ (удары рукой или ногой).

В правом здании находятся ракеты и два пульта управления. Первый пульт - для загрузки перфоленты, второй - отключение электронной защиты.

При загрузке игры внимательно рассмотрите здание, изображенное на заставке, это поможет Вам ориентироваться в игре. Приготовьте карандаш и бумагу, желательно по ходу игры рисовать лабиринт.

После загрузки игры нажмите "ENTER" - перед вами главное меню программы. Вы можете выбрать себе нужные органы управления, а также миссию ("M"). Всего в этой игре, как сообщают читатели, до девяти миссий. После завершения очередной миссии Вам выдается пароль на следующую.

Вот расписание миссии, как оно составлено по письмам читателей.

1. ENTER - пройти по зданию, ликвидировать несколько охранников и покинуть здание на мотоцикле.

2. JONIN - пройти по зданию, ликвидировать несколько охранников и покинуть здание через туннель (пешком!).

3. KIME - найти два фрагмента перфоленты и покинуть здание на мотоцикле.

4. KUI KIRT - собрать пять фрагментов перфоленты, отключить электронную защиту, покинуть здание на мотоцикле.

5. SAIMENJITSU - то же, но собрать 9 фрагментов перфоленты.

6. GENIN - собрать 9 фрагментов перфоленты, ввести их в пульт управления ракетой, отключить электронную защиту и покинуть здание на мотоцикле.

7. MI LU KATA - собрать 11 фрагментов перфоленты, ввести в пульт управления и покинуть здание через туннель.

8. DIM MAK - собрать 14 фрагментов перфоленты, ввести их в пульт управления ракетой и покинуть здание через туннель на мотоцикле.

На каждом последующем уровне время, отведенное на выполнение задания, сокращается на 50%.

По всей видимости существует какое-то стандартное описание игры. Потому что до этого момента письма читателей в основном совпадают, но самое интересное наступает дальше.

Товарищ Скуратов из Муромы сообщает о девятом (!) уровне, пароль которого - SATORI (об этом же сообщает т. Каневский из Киева), а также о наличии в игре комнаты на нулевом этаже в правой части здания, после пребывания в которой Ваша энергия больше не расходуется. Можете падать с любых высот и смело сражаться с охраной. Сложность отыскания этой комнаты в том, что туда ведет невидимая лестница.

Товарищ Боярский из Ленинграда приводит и схему для отыскания этой комнаты:



Товарищи Подопригора из Днепропетровска и Шилин из Симбирска тоже отмечают сложность отыскания этой комнаты и приводят несколько POKES для облегчения выполнения миссий:

POKE 37122,0 (время)

POKE 61340,201 (энергия).

Товарищ Каневский дополняет эти данные следующими:

37121,0; 37122,0 - время

61338,182; 61382,182 - энергия,

Товарищ Дейна из Новгорода подчеркивает разницу в том, как завершаются миссии - уехать на мотоцикле (через левый выход) или уйти пешком - через правый. Когда прибывает мотоцикл - появляется сообщение BIKE ARRIVED.

Он же делится несколькими полезными приемами, которые могут пригодиться в игре.

1. Ввод перфоленты в пульт управления ракетой.

На высоких уровнях (с четвертого) появляется зеленый код. Это означает, что Вам надо добраться до правого здания, добраться до него можно по канату или пешком через нижние этажи. По канату надо проходить не останавливаясь. Если в длинном обходном пути через нижние этажи не хватает энергии, можно отдохнуть на дереве, в лифте или на ровной плоской площадке.

Когда переберетесь в правое здание, надо подойти к пульту с левой стороны у второго яруса ракеты, чтобы он загорелся и нажать FIRE. Если у Вас есть необходимое количество перфолент, то появится сообщение CODE ACCEPTED.

2. Полезный совет для рукопашного боя. Если у Вас мало сил и Вы не уверены, что Вам удастся убежать, сделайте следующий фокус: подойдите к краю экрана так, чтобы половина вашего туловища исчезла, развернитесь и сядьте. Теперь охранник не достанет Вас ни ножом, ни ногой и Вы можете сделать с ним, что хотите. Для пантеры это не проходит, но сидя ее легко убить. Правда сидя нельзя метнуть предметы.

Многokrатно пройдя игру, товарищ Дейна убежденно считает ее одной из лучших игр для "Спектрума" и сообщает, что есть непроверенные сведения о том, что где-то в программе есть комната, в которой можно остановить время. Ему ее найти не удалось. Но ведь должен же кто-нибудь что-то об этом знать?

Как мы и ожидали и, честно говоря рассчитывали, волну писем вызвала печать инструкций к программе "ELITE". Это не случайно, мы ведь сказали, что программа намного глубже, чем то, что написано в инструкции. Фирма специально оставила в ней неосвещенными многие вопросы и программа несколько лет держала в напряжении любителей этой игры. Ей посвящались рубрики в Синклер-журналах и систематические исследования.

Как оказалось, у нас тоже очень много межгалактических ассов, готовых немедленно ринуться в космические битвы. Правда, некоторые обратили внимание на наши слова о том, что фирменная инструкция имеет громадные размеры и мы дали ее в сокращении. Заподозрив нас в утаивании информации, они просят выслать им ксерокопии полной инструкции.

Конечно мы не ангелы и у нас бывают ошибки, но сознательный брак мы не делаем. Если мы что и выкинули из оригинального текста, так это только лирику (правда не всю, - нужно было бы еще больше).

Учитывая нынешний дефицит в услугах по копированию и очень высокие цены (для нас - от 75 к. до рубля за страницу) мы конечно никаких выкопировок для огромной массы желающих делать не можем, да это и не нужно. Давайте лучше вместе разбираться с этой программой.

Теперь об ошибках. В главе 13 выпали строки там, где описана стыковка с орбитальной станцией, что ввело в заблуждение т. Рутковского и некоторых других. Принося извинения, даем подробную консультацию на сей счет. После того, как Вы развернулись носом ко входу в станцию, установите сначала минимальную скорость сближения. Затем совместите оси корабля и станции с тем, чтобы во-первых они совпадали, а во-вторых корабль двигался строго перпендикулярно поверхности станции. Лучше всего оси совмещать парными импульсами. Нажали на клавишу "вверх" на долю секунды. Отпустили, подождали некоторое время и нажмем на клавишу "вниз" на такое же время. Когда оси совмещены, начинайте сближение. Когда приблизитесь достаточно, увидите, что входное отверстие имеет прямоугольную форму, этот прямоугольник вращается, т.к. станция вращается вокруг своей оси. Чтобы нормально пристыковаться, Вам надо тоже закрутить свой корабль вокруг его оси (клавишами "вправо/влево").

Если скорости вращения совпадут, то прямоугольное отверстие "застынет" на экране. Можете причаливать. Если Вы увидите, что полностью остановить прямоугольник нельзя, если он "бьет" после синхронизации вращения, значит Ваши оси не совпадают. Чем ближе к станции, тем сильнее биения. Подход надо повторить. Мы делали так. Почти у самого входа на станцию гасим скорость почти до нуля, делаем крутой разворот по тангажу на 180 градусов (клавишей "вверх" или "вниз"), переключаем экран на задний. Когда аппертура станции (приемное отверстие), появится в центре заднего экрана, останавливаем отработку тангажа подачей противоположного импульса и резко набираем скорость, удаляясь от станции, отойдя на приличное расстояние, опять сбрасываем скорость, разворачиваемся по тангажу на 180, переключаемся на передний экран, ловим аппертуру в центре экрана, сближаемся, тормозимся, отрабатываем угол крена ("влево"/"вправо"), синхронизируемся и, если все в порядке, стыкуемся, иначе - еще раз. Через неделю Вы будете это делать вдохновенно и артистично.

Если все это кажется Вам очень сложным, то пусть Вас утешает мысль, что это очень похоже на управление реальным космическим кораблем, и здесь Вы хоть чуть-чуть почувствуете как работают наши космонавты на ручной стыковке с "Салютом".

Теперь ответы на некоторые вопросы.

Товарищ Зайцев из Воронежа заслуженно получил ранг COMPETENT, но дальше дело не движется, ранг не меняется, а порой даже сбрасывается до "HARMLESS", в самом худшем подозревается собственный компьютер ("Балтика").

Что касается снижения ранга, то это что-то странное, а в остальном мы можем предположить, что видимо настало время проявить свои силы в исполнении специальных миссий. Откроем секрет, которого нет и не было в инструкции. По мере вашего роста в

глазах "Спектрума", он может Вам дать спецзадание. На "Сп. 48" их три, а в версии 128K их шесть. Первое задание - "Взрыв сверхновой" - Вам предложат спасти население некой планеты, которой угрожает гибель, и щедро за это расплатятся. Но миссии выдает компьютер сам, Вам их вызвать нельзя.

Ряд вопросов, на которые нет ответа:

Как поменять корабль на корабль другой марки? Как доставить "Таргон" в исследовательский центр? Как получить пилотскую лицензию? Если записано, что "корабли, отказывающие отвечать на запрос - пираты...", - то как послать запрос? И другие подобные. На все эти вопросы ответ - НИКАК. Эти вопросы возникли потому, что мы не всю "лирику" выкинули из фирменной инструкции. Это ведь только антураж, игра ведь аркадная, и эти события - ПОДРАЗУМЕВАЮТСЯ. Вывод Вы должны были сделать совсем другой. Если хотите строить в космосе честную карьеру и не заниматься пиратством, то при сближении с космическим телом не открывайте огонь первым. Ждите, когда корабль себя проявит как враг, тогда с ним и расправляйтесь.

Есть еще серия вопросов типа: Сколько энергетических отсеков можно купить одновременно? Одноразовая ли спасательная капсула? Можно ли энергетической бомбой уничтожить станцию? и другие подобные. Ответ на все эти вопросы должен быть таким - ПРОВЕРЬТЕ И УЗНАЕТЕ. Дело не в том, что нам жалко дать такую информацию, но на ход игры она влияния не оказывает, а существует очень хороший принцип компьютерных "хакеров" - "ВСЕ, ЧТО МОЖНО ПРОВЕРИТЬ, НУЖНО ПРОВЕРИТЬ". Только так можно стать экспертом в любой программе - будь то примитивная "стрелялка" или новый язык программирования.

С интересной мыслью обратился наш читатель Довженко В.П. из Киева. Он предложил использовать в игре джойстик, что гораздо удобнее, для этого надо после загрузки программы не нажимать клавишу BREAK, а нажать подключенный джойстик влево. Если же в этот момент нажать клавишу "курсор влево", то можно будет управлять курсорными клавишами.

В нашей почте лежит обширнейшее письмо с килобайтами ценнейшей информации от т. Дейна. Если их все собрать вместе и обработать, то получился бы бестселлер под названием "О чем умолчал ИНФОРКОМ". Нам самим эта работа очень понравилась, и мы сообщаем адрес автора. От себя можем добавить, что если многое из того, о чем он пишет, не явилось более поздними "доработками" хакеров, то его сведения превосходят то, что вскрыли западноевропейские пилоты.

173024, Новгород, ул. Попова, дом 15, корп. 1, кв. 39. Дейна А. А.

Сообщаем также адрес еще одного эксперта по проблемам "Элиты", изъявившего желание вести консультирование и обобщать поступающую информацию:

325026, Херсон, пер. Комбайновый, д. 11, кв. 3, Жаров Р.М.

Если Вы, уважаемые читатели, разделяете нашу мысль о том, что "ФОРУМ" заработал и набирает обороты, то может быть начнем новые атаки? В план очередного штурма включены программы по письмам наших читателей:

TOTAL ECLIPSE (*INCENTIVE*)

MOVIE (*IMAGINE*)

BATMAN (*OCEAN*)

THE GREAT ESCAPE (*OCEAN*)

THEIR FINEST HOUR (*CENT.COM.*)

SHADOWFIRE (*BEYOND*)

STRIKE FORCE COBRA (*PIRANHA/MACMILLAN*)

Есть идеи? Пишите, только просьба: одно письмо - одна проблема. Иначе с письмами не управиться, не срабатывает принцип распределения обязанностей между сотрудниками при обработке почты.

Представьте себе, если в одном письме заказ на новые разработки, просьба о досылке какого-нибудь выпуска "РЕВЮ", утерянного по вине почты, вопросы по работе с какой-нибудь программой, предложение своих услуг и полезные идеи для "ФОРУМа". Нам очень трудно работать таким письмом, неясно кому поручить его обработку.

Настала пора подводить итоги конкурса, который мы объявили в конце прошлого года. Напомним, что это конкурс на лучшую тематическую подборку игровых программ для "Спектрума".

Воспользовавшись тем, что этот выпуск имеет двойной объем, и в рекламном разделе образовалось место, мы привели списки поступивших проектов в "Рекламном приложении" и представляем их на Ваш суд.

STRATEGIC GAMES

TOBRUK

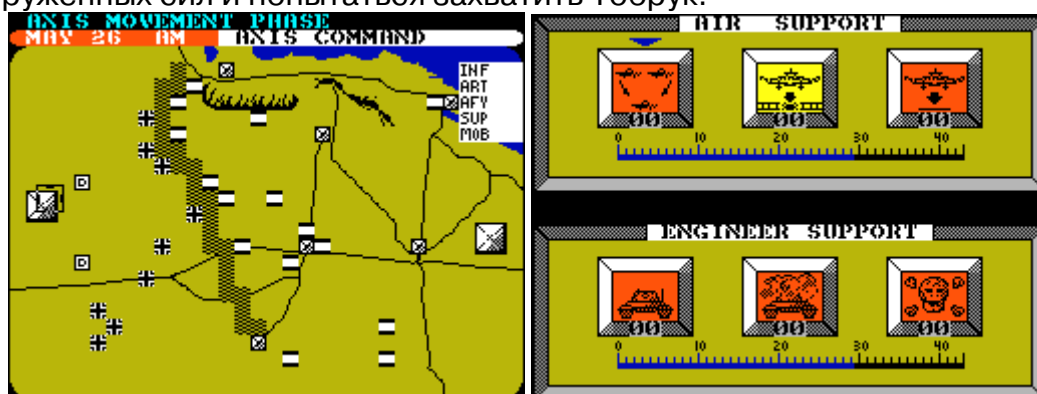
"PSS" 1987



Танковые сражения в пустыне - вот тема стратегической игры TOBRUK. В основу игры положены реальные события боев второй мировой войны в Северной Африке.

Тобрук - маленький городок на Средиземноморском побережье Ливии. В январе 1941 г. он был занят английскими войсками и с апреля по декабрь 1941 г. выдержал длительную осаду объединенных немецких и итальянских войск.

Играть можно как вдвоем, так и против компьютера. В последнем случае компьютер исполняет роль союзных войск, а Вам предоставляется испытать себя в качестве германских вооруженных сил и попытаться захватить Тобрук.



Ко времени начала немецкого наступления союзные войска не успели закончить минирование подступов к городу. Тем не менее, даже незавершенная линия обороны поставила германское командование перед тройной проблемой. Что делать:

- прорывать оборону невзирая на потери?
- искать обходные пути и возможности для маневра?
- полагаться на длительную работу инженерных войск по подготовке прорыва?

Основной же целью их плана наступления являются шесть оазисов, через которые идет снабжение войск союзников.

Как и в большинстве стратегических игр, игра проходит в виде последовательности ходов. Во время хода войска могут перемещаться и атаковать армии противника. Если с помощью джойстика или курсорных клавиш установить курсор на боевом соединении, то на экране появляется окно, в котором показаны:

- наименование соединения;
- количество пехоты (INF);
- артиллерии (ART);
- бронетехники (AFV);
- условия снабжения (SUP);
- и фактор мобильности (MOB).

Фактор мобильности определяет на сколько полей может быть передвинуто это соединение за очередной ход. Естественно, на него оказывает влияние рельеф местности,

но также и качество снабжения.

Обе воюющие стороны очень сильно зависят от условий снабжения (тылового обеспечения). Снабжение осуществляется боеприпасами, горючим и подкреплениями. В принципе игра в Тобрук, в отличие от многих других стратегических игр - это в первую очередь игра в организацию тылового обеспечения.

Союзники получают свое обеспечение через сеть транспортных путей, опирающихся на шесть ключевых оазисов и вынуждены отступать по мере их потери. Германские же войска имеют два мобильных снабженческих подразделения, которые могут питать войска, расположенные от них на расстоянии не более шести игровых полей. Поэтому если Вы держите свои войска в компактных группировках, у Вас больше шансов выиграть сражение. Но имейте в виду, что Вы должны обеспечить самую лучшую защиту отрядам снабжения, иначе грузовики будут уничтожены и игра проиграна.

В отличие от некоторых стратегических игр в Тобруке есть командная стадия игры. На этой стадии Вы можете назначить одну из шести возможных диспозиций своим воздушным силам и инженерным подразделениям. Так, например, воздушным силам может быть отдан приказ на завоевание превосходства в воздухе или на поддержку наземных атак. В свою очередь инженерным подразделениям может быть поставлена основная задача на восстановление пораженной бронетехники или на расчистку минных полей или на организацию противодействия вражеским танкам. Каждая из диспозиций изображается цветным квадратом на экране, цвет которого меняется в зависимости от того, как идет выполнение данной задачи. Красный цвет означает, что в выполнении данной задачи противник Вас обошел, желтый - Ваши успехи примерно равны, а зеленый - успех на вашей стороне. Если в обеспечении наземных подразделений Вы вышли вперед, то имеете право на проведение дополнительной атаки или воздушного рейда против вражеского соединения. Если в расчистке минных полей Вы получите зеленый квадрат, то один участок будет освобожден от мин. Когда расчистите их достаточно, сможете порадовать противника неожиданной атакой.

Как и некоторые другие игры фирмы PSS, Тобрук имеет в своем сценарии аркадные вставки. Это как бы миниигра в большой игре. В ней Вы можете проверить свои силы в качестве командира танка на поле боя. Вы можете выбрать один из четырех постов управления танком и вооружением:

- общий обзор (карта);
- пулемет;
- орудие (башня);
- ходовая часть (движение).

Ваша задача подавить как можно больше целей, располагая ограниченным боекомплектom. Работа требует недюжинной реакции, поскольку постоянно требуется переключаться между четырьмя экранами и делать это надо очень быстро, пока Вас не разбили. Ваш успех в этой аркадной вставке определяет результат сражения между вашими соединениями и соединениями противника. Пока у Вас нет необходимого опыта, лучше избежать этих аркадных вставок, благодаря тому, что их можно отключать по желанию.

Даже и без них Тобрук выглядит одной из наилучших стратегических игр.

LEGIONS OF DEATH

"LOTHLORIEN" 1987
LEGIONS ◁ OF
◁ DEATH ▷

Для сюжета этой интересной игры фирма выбрала один из наименее исследованных периодов военной истории. Действие происходит в Средиземноморье где-то между 264 годом до нашей эры и 146 годом до н.э. и относится к противостоянию флотов Рима и Карфагена. Те, кто увлекаются древней историей, уже поняли, что речь идет о Пунических войнах. Исторический аспект таков: Рим и Карфаген схватились в жестокой войне за обладание торговыми путями по Средиземному морю. К успеху в этой длительной и изнурительной войне придет тот, кто обеспечит контроль за основными проливами.

"Легионы смерти" - игра для двоих, но в нее можно играть и в одиночку против компьютера. Ваша задача - разгромить корабли противника, но в то же время, Вы должны обеспечить надлежащую охрану собственных портов. Успех в игре зависит от Ваших способностей проводить военные операции на обширных просторах, а также от умения принять правильное решение в сложных ситуациях, когда время на решение ограничено, а от правильности стратегии зависит многое.

Оригинальный подход, необычный для стратегических игр состоит в том, что Вам самому предлагается назначить условия победы. Это может быть количество захваченного золота, количество уничтоженных кораблей противника или размеры захваченных территорий. Благодаря такой возможности у Вас появляется способ регулировать продолжительность игры и уровень ее сложности.

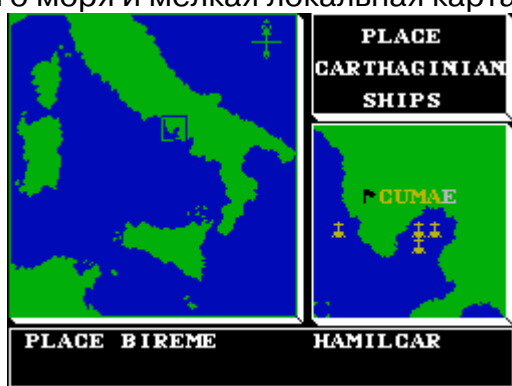
"Легионы смерти" можно было бы с полным правом назвать "Легионами пиктограмм". Пиктограммы были изобретены, чтобы обеспечить пользователю простоту и наглядность ввода в игру информации, но когда их становится очень много, то работа с ними может порождать трудности. Впрочем, это не упрек в адрес "Легионов смерти", потому что программа очень и очень глубока и без них с ней не разобраться.

Игра состоит из нескольких стадий. На первой стадии Ваша задача - построить и оснастить флот, а также нанять экипажи. Для всего этого Вы имеете 1000 единиц золотого запаса. Выбор корабля выполняется из пяти предлагаемых - от быстороходной, но слабозащищенной биремы до могучего и неповоротливого гептареса. После того, как корабль выбран, он оснащается парусами, надстройками и пр. Среди модификаций Вы можете выбрать даже и специальный прочный настил, идущий вдоль борта, предназначенный для упрощения десантирования на вражеский корабль в абордажном бою (CORVUS). Правда надо иметь в виду, что он точно также упрощает и вторжение на Ваш собственный корабль.



На этой стадии игры так много возможностей для выбора, что она может и затянуться, особенно если Вы сели в первый раз и еще не научились ловко порхать между пиктограммами.

На следующей стадии Вам надо расположить свои суда по Средиземному морю. Здесь все проходит просто, особенно если учесть, что в Вашем распоряжении две карты - одна крупная карта Средиземного моря и мелкая локальная карта зоны боевых действий.



После этого наступает самая трудная часть игры - сражение, и здесь опять Вы перемещаете свои суда, отдаете им приказы с помощью пиктограмм. Когда два судна вступают в ближний бой, то в нижней трети экрана детально изображается картина схватки.

Эта игра определенно доставит удовольствие тем, кто ищет разнообразие после многочисленных сухопутных, боев. Совершенно очевидный исторический дух присутствует в программе и доставит радость тем, кому надоели сюжеты Второй Мировой или Наполеоновских нашествий. Начинаям игра против компьютера может показаться скучноватой, в то время как опытные стратеги найдут в глубинах этой игры то, чего им давно не доставало. Управление пиктограммами поможет начинающему очень быстро "включиться" в игру и начать ее освоение, но по мере все более глубокого погружения техника их использования становится все сложнее и сложнее и игра требует уже профессионального подхода. В общем, в этой игре найдут свои положительные моменты любители всех категорий, а заодно это поможет пробудить интерес и желание узнать побольше об эпохе Пунических войн.

BATTLEFIELD GERMANY

"PSS" 1987



Это полноэкранный имитатор войны, которая не должна никогда произойти. Речь идет о войне между НАТО и Вооруженными силами Варшавского Договора. История возникновения гипотетического конфликта достаточно интересна и поучительна, но имеет малое отношение к самой игре и мы ее приводить не будем.

Вы можете сами выбрать за какую сторону Вы будете играть, если игра ведется вдвоем. Если же Вы играете против компьютера, то придется вам воевать за НАТО, но не расстраивайтесь - все равно Вам не победить ВС ВД.

В игре задействовано невероятно большое количество соединений. Среди них есть и пехота и мотопехота и бронетанковые войска, а также парашютные, вертолетные части и даже горные стрелки. Все это перемещается по карте Европы, которая имеет возможность скроллить во всех направлениях. Очень удобно, что в правом верхнем углу размещено окно, в котором в укрупненном масштабе показана карта Европы с нанесенной на ней стратегической обстановкой. Соединения на этой карте показаны в виде мигающих точек. На основной карте соединения представлены пиктограммами. Информацию по каждому соединению можно получить, если ввести в его бокс курсор.



Игра разыгрывается в шесть или в двенадцать ходов. Каждый ход может занять до часа времени и состоит из двух этапов.

На первом этапе выполняется снабжение войск. Компьютер выполняет это автоматически, а на втором производится управление войсками - их перемещение и нанесение ударов. Цель - захват территории. Компьютер играет очень жестко и наказывает

за любые стратегические или тактические промахи. Если Вы не чрезвычайно опытный игрок в такие настольные игры, привыкший к шестигранным игровым полям, то у Вас очень мало или почти никаких шансов противостоять компьютеру. Эта игра - для экспертов, начинающим лучше сначала отточить силы на ТОБРУКе или чем-нибудь подобном.

SAMURAI

"CRL" 1986



Это игра для одного игрока, имеющая три различных сценария и три уровня сложности. Вы управляете сборным отрядом из воинов различного боевого амплуа. Ваша задача - уничтожить вражеские силы, охраняющие замок.

Управляется игра методом пиктограмм и начинается с того, что Вы собираете под свои знамена войско, выбирая себе воинов четырех типов. Каждый из них чего-то "стоит", а ваши возможности расплаты ограничены.



В Вашем распоряжении:

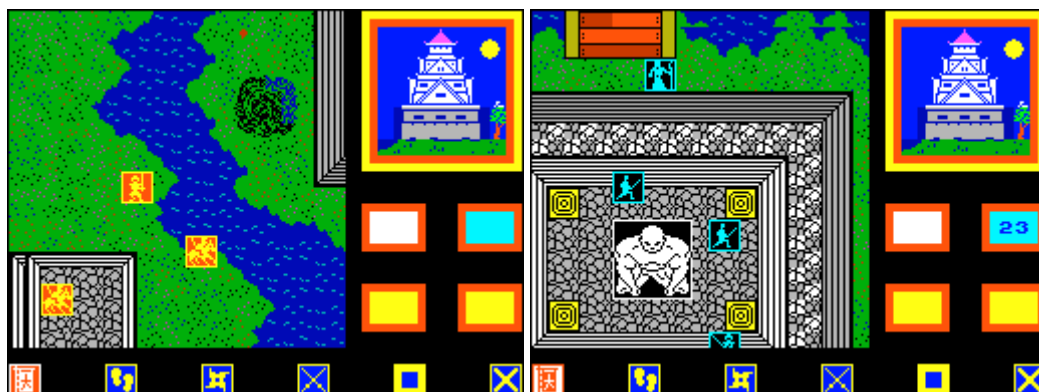
Ашигари - легкая пехота.

Самураи – традиционные японские воины.

Конные самураи.

Смертоносные ниндзя.

Механика игры очень проста и состоит в перемещении и нападении. Особенно аккуратно надо распоряжаться с ниндзя, поскольку это единственные воины, способные наносить удар на расстоянии. Самая лучшая тактика их применения - диверсионная "бей и беги".



Игра настолько проста, что ее можно смело рекомендовать начинающим карьеру в стратегических играх. Во многом она напоминает игру SWORDS OF BANE фирмы CCS.

BATTLE OF BRITAIN

"PSS" 1987



Эта игра относится к событиям Второй Мировой войны, а именно к 1940-ому году. Планируя вторжение на Британские острова, Гитлер отдал приказ командующему имперскими люфтваффе Герингу подавить Английские Королевские Воздушные силы и завоевать превосходство в воздухе.

Учитывая крайнюю скудность снабжения английских воздушных сил, почти любая тактика Геринга принесла бы успех, если бы она упорно проводилась в действие, но тому показалось, что гораздо надежнее переключить свой воздушный флот на ночные бомбардировки мирных городов. Конечно к подавлению английских ВВС это не привело по той простой причине, что "Спитфайры" и "Харрикейны" по ночам просто никогда не летали.



В игре Вы можете выбрать один из трех сценариев. Тренировочный режим поможет Вам освоить методику управления воздушными силами путем имитации легкого воздушного налета. В сценарии "Блицкриг" немецкая авиация бросает все, что есть в ее распоряжении, но только в течение одного дня. И, наконец, есть сценарий затяжной кампании, рассчитанной на тридцать дней.

В игре есть также аркадные вставки, в которых Вы можете попробовать посбивать "Мессершмиты", не вставая с собственного стула. Если Вы входите в такую аркадную вставку, то успех Ваш в стрельбе по "Мессершмитам" будет рассматриваться и как успех в локальных воздушных боях и будет отражаться на ходе всей кампании.

Конечно это хорошо для любителей аркадных "стрелялок", но настоящим любителям стратегических операций лучше этот режим сразу отключить, чтобы не мешал заниматься серьезным делом. Успех в игре зависит от Вашей способности быстро принимать ответственные решения в сложных ситуациях.

Как только появятся немецкие самолеты, Вам надо поднять в воздух свои эскадрильи. После боя или продолжительного патрулирования в воздухе эскадрильи надо посадить, чтобы они могли дозаправиться и довооружиться. Если Вы упустите момент, когда их надо возвращать, они расплывутся по ближайшим аэродромам и пройдет некоторое время, пока Вы сможете их снова собрать и привести в боеготовность.

Дополнительную сложность привносят погодные условия, т.к. в плохую погоду некоторые аэродромы могут быть закрыты для приема самолетов.

Игра несложна в работе, но чем-то напоминает искусство жонглирования. Каждый знает, что подбросить в воздух одновременно несколько мячей нетрудно, но вот как сделать, чтобы все они там и оставались?

VULCAN "CCS" 1987



Эта игра имитирует события боевых действий в Тунисе в 1942-1943 годах и является достойным образцом среди игр этого жанра.

Управление игрой - через систему меню.



В игре есть пять сценариев, рассчитанных на то, чтобы ее продолжительность составляла от получаса до шестнадцати часов. Вы можете выбрать за кого Вы будете воевать - за немецко-итальянские войска (Axis) или за англо-американские (Allied). Играть, как и в большинстве игр можно против партнера, а можно и против компьютера.

Существует и версия для 128-го компьютера, в которой есть дополнительные возможности:

- серьезный информационный блок;
- несколько исторических сценариев, разработанных по принципу "Что было бы, если бы...";
- а также устранена необходимость подзагрузки данных после каждого сеанса игры, что необходимо в модели 48K по причине нехватки памяти. В игре есть очень приятная особенность, которой лишены настольные аналоги военных игр. Здесь можно сделать перемещение войск противника скрытым от Вас. Вы обнаруживаете вражескую часть только буквально столкнувшись с ней.

Существуют три способа перемещения своих частей.

Нормальное перемещение - moving. Нападение - assault. Перемещение походной колонной - travel.

При нападении соединение идет до достижения своей цели, а при походном марше может двигаться только по дорогам, но зато с удвоенной скоростью. Правда в этих условиях

оно очень легко уязвимо для вражеских атак.

Если Вы не хотите перемещать часть, то можете выбрать один из двух режимов:

- обороняться (hold position);

- закрепляться (fortify). Рельеф местности играет в игре очень большую роль.

Естественно, он сказывается на скорости перемещения частей, но не только. От него во многом зависит насколько успешно пройдет атака или насколько успешно Вы атаку отразите. Успех каждого из многочисленных сражения между вошедшими в контакт противостоящими частями также зависит от многих факторов: от силы частей, от условий их снабжения и даже от погодных факторов. Снабжение своих войск - важнейший элемент Вашей стратегии. Необходимо также хорошо использовать авиацию как для нанесения ударов по врагу, так и для решения разведывательных задач. Эта игра может быть рекомендована вниманию даже самых искушенных игроков, она всем доставит удовольствие.

DESERT RATS

"CCS" 1987



Программа "Крысы пустыни" больше подойдет тем, кто ставит игровые аспекты выше боевых. Это скорее шахматы, в которых вместо фигур - танковые полки, а вместо доски - разбитая на клетки карта ливийской пустыни. Карта очень гладко скроллирует, предоставляя Вам игровое пространство внушительных размеров. Можно отметить очень четкую графику.



В игре есть шесть сценариев, относящихся к боям в Северной Африке 1941-1942 годов, начиная с высадки дивизий Роммеля в Триполи и кончая битвой за Эль Аламейн.

Успех в битвах зависит от правильной расстановки сил и от умения обеспечить на должном уровне снабжение боевых частей.

Это не игра в режиме реального времени. В ней реализован двухшаговый метод. Сначала Вы выдаете каждой части приказ на данный ход, а потом сидите и смотрите, что из этого вышло. На ваших глазах перемещаются Ваши полки и полки противника, входят в столкновения, стирают друг друга с лица земли.

Играть можно как против партнера, так и против компьютера. Конечно поначалу у Вас почти нет шансов как-то противостоять Роммелю, но если Вы играете с другом, то может быть Вам и удастся сбросить его войска в море.

Для тех, кто знаком с такими играми, эта игра не испортит общего впечатления от жанра, а для тех, кто еще не начал, она могла бы стать прекрасным началом.

INVASION

"MASTERTRONIC" 1987



Если предыдущие стратегические программы, рассмотренные в данном разделе проработаны нами на уровне обзора, то нижеследующее описание является полноценной инструкцией по работе.

Эта стратегическая программа необычна тем, что боевые действия в ней ведутся между фантастическими вооруженными силами.

История конфликта примерно такова.

Колонизируя новую планету, поисковый отряд столкнулся с инопланетной цивилизацией, которая занимаясь тем же, установила здесь станцию управления погодой и занимается тем, что создает условия жизни совершенно неприемлемые для землян. Ваша задача - разгромить войска противника, разрушить эту станцию и, по возможности, живым и невредимым вернуться домой.

Основные факторы в игре - стратегия, скорость и тактика. На каждый Ваш ход пришельцы ответят так, чтобы помешать Вам разрушить их станцию управления погодой и выиграть сражение. Вам необходимо тактически правильно спланировать кампанию и все время быть начеку. Противник только и ждет Вашей ошибки и зачастую второго шанса у Вас не будет.

Под Вашим управлением находятся различные боевые части. Используйте их рационально, поскольку у противника тоже есть соответствующие аналоги и вскоре они их проявят.

Ваши боевые части

1. Радары.

Они управляются клавишей R и являются как бы Вашими "глазами". Карта на экране изображает то, что находится вокруг радара. Радаров у Вас четыре и, нажимая клавишу "R", можно между ними переключаться.

2. Минные части.

Они представляют из себя дистанционно управляемые ядерные взрывные устройства. Взрывное устройство, взорванное рядом с космическим кораблем или заводом полностью выводят их из строя. Поражаются и все находящиеся в смежных полях боевые части. Но для того, чтобы взорвать вражескую станцию управления погодой, необходимо взорвать два рядом стоящих взрывных устройства.

3. Воздушная пехота.

Это очень быстрые и маневренные соединения, имеющие среднюю защиту и огневую мощь. Оно может атаковать только смежные с ним части, но может преодолевать кислотные озера и минные поля.

4. Парящая броня.

Эти части имеют примерно те же характеристики, что и воздушная пехота, но у них радиус действия больше.

5. Страйдеры.

Это медленные части со средней степенью защиты. Они могут атаковать только

смежные с ними позиции.

Боевые части инопланетян

1. Тяжеловооруженные части.

Это медленно перемещающиеся войска, но с тяжелым вооружением, мощной броней и довольно большим радиусом действия.

2. Части быстрого реагирования.

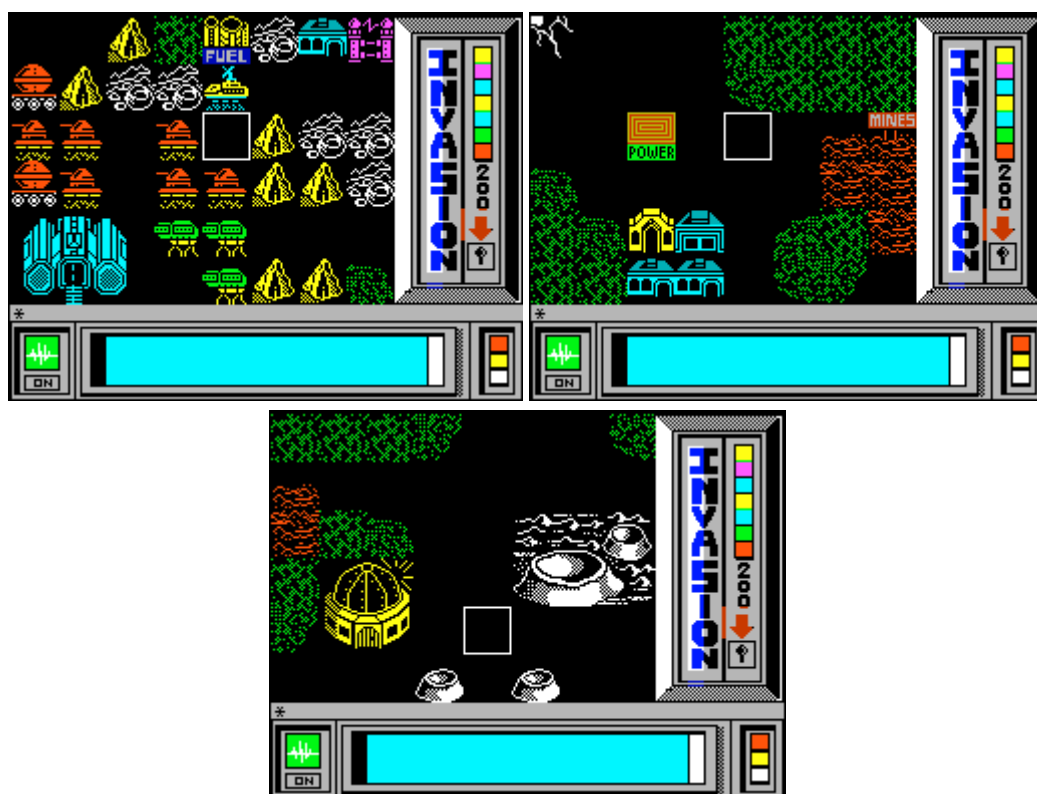
Быстрые воздушные части с низкой защищенностью и малой огневой мощностью. Они имеют возможность преодолевать кислотные озера и минные поля.

3. Дроиды.

Аналог земных частей "Парящая броня".

Игра

Главная цель - уничтожить станцию управления погодой. Для этого Вам надо взорвать два из трех Ваших взрывных устройств в прилегающих к ней позициях. Если Вы сумеете это сделать, то вам надо как можно быстрее покинуть планету на одном из двух возможных космических кораблей, прихватив с собой как можно больше своих частей. Если Вы оставите второй корабль неразрушенным при оставшихся в целости вражеских войсках, то инопланетяне стартуют вслед за Вами и собьют вас, поэтому надо либо уничтожить все вражеские части, либо взорвать вражеский корабль последним оставшимся взрывным устройством.



Инопланетяне могут уничтожить космический корабль, поместив на соседние поля три боевых части и дав им команду на самоликвидацию. Если оба корабля и станция управления погодой уже уничтожены, то Вам остается только закончить игру или вести войну до полного уничтожения.

Дополнительные возможности

1. Боевые части могут скрываться на заводах и фабриках и вести огонь оттуда.
2. Части могут частично восстанавливать свои повреждения, если их поместить на поле с топливным устройством.
3. Войска могут погружаться на космический корабль только с полей, смежных с дверьми.
4. При взрыве минного устройства уничтожаются все смежные с ним части - не только

вражеские, но и Ваши. При поражении вражеским огнем они также взрываются.

Полезные советы

1. Защищайте свои минные части. Без них Вы не сможете победить. Особенно тщательно оберегайте их от дальнего огня противника.
2. Следите за сообщениями телепринтера.
3. Постоянно следите за собственными частями. Небрежность может превратить их в артиллерийскую мишень.
4. Всегда имейте в виду основную цель.

Счет

В конце игры Вы получите итоговую таблицу. Наибольшие очки даются за уничтожение станции управления погодой или вражеского корабля противника. Те же очки можно получить и за уничтожение собственного корабля, если Вам удастся улететь на вражеском.

Начисляются очки также и за каждую уничтоженную вражескую часть и вывезенную на корабле свою.

Управление игрой

Если поместить курсор на свою часть и нажать "огонь", то в специальном окне на экране появится сообщение о ее состоянии. Если текущее положение Вас устраивает, нажмите "огонь" еще раз, в противном случае переместите курсор в новое место и вновь нажмите "огонь". После этого курсор автоматически укажет на новую часть, хотя Вы можете ее и пропустить нажатием клавиши N.

Ваши части защищаются от огня самостоятельно и сами выбирают цель для стрельбы. Вы не можете выбирать цели.

Расписание клавиатуры

- N - переход к следующей части;
- R - переход на следующий радар;
- F - переключатель "Пауза";
- V - выгрузка игры на ленту;
- E - подрыв взрывного устройства;
- T - запуск космического корабля, если на нем установлен курсор.

ТЕХНИЧЕСКОЕ ПРИЛОЖЕНИЕ

ЭТОТ РАЗДЕЛ ВЕДЕТ НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС",
СПЕЦИАЛИЗИРУЮЩИЙСЯ В ВОПРОСАХ АППАРАТНОГО ОБЕСПЕЧЕНИЯ СИНКЛЕР-
СОВМЕСТИМЫХ КОМПЬЮТЕРОВ И СХЕМОТЕХНИЧЕСКИХ РЕШЕНИЙ ПЕРИФЕРИЙНЫХ
УСТРОЙСТВ

137566, МОСКВА И-566, НТК "ПЛЮС"

ВНИМАНИЕ

В программе поддержки ZX-Модема обнаружены опечатки. Первым об их обнаружении сообщил товарищ Ключник из Полтавы.

Строка 46:

Напечатано: LET a(1) = FN(b\$)

Должно быть: LET a(1) = FN A(b\$)

Строка 202:

Напечатано:

F940 01 32 8E FC 3E 02 32 A3 09

Должно быть:

F940 01 32 8E FC 3E 02 32 A3 0B

Строка 210:

Напечатано:

F980 0B F9 AF C9 CD AF F9 3E AB

Должно быть:

F980 0B F9 AF C9 CD AF F9 3E A8

Строка 238:

Напечатано:

FA60 0A 2B CD 18 1C F5 11 80 13

Должно быть:

FA60 0A 28 CD 18 1C F5 11 80 13

БУФЕРИЗАЦИЯ

Многие наши читатели являются убежденными поклонниками Синклер-совместимых компьютеров. Хорошо понимая и осознавая ограниченность этих машин, они также понимают, что в настоящее время нет никакой другой разумной альтернативы домашнему компьютерному творчеству. В нем они находят источник постоянного самосовершенствования и самоутверждения.

Совершенствуя себя, развиваясь и образовываясь, многие задаются вопросами совершенствования и своего компьютера, ведь как бы хорош он ни был, пытливый человек всегда найдет как сделать что-то лучше, иначе, не так как у всех. Одному из направлений возможного совершенствования Вашей машины и посвящена сегодняшняя статья "Технического приложения".

Фирменные и всевозможные варианты схем самодельных Спектрумов за редким исключением имеют общий существенный недостаток. Этот недостаток заключается в том, что микропроцессор Z80 - сердце Спектрума - не защищен никакими шинными формирователями, усилителями и другими буферными элементами. В фирменном компьютере это обусловлено малым количеством микросхем, подключенных к шинам микропроцессора (всего около 20), но и там любая фирменная периферия, подключаемая к краевому разъему платы, вынуждена на входе иметь буферные элементы.

Самодельные компьютеры и без того используют предельные по нагрузочной способности режимы микропроцессора, а если встает вопрос о дополнительной периферии, то тут уже без усилителей шин не обойтись.

Кстати, с перегрузкой шин микропроцессора в самодельных компьютерах очень часто связаны всевозможные неполадки. Иногда не удается запустить правильно собранный из исправных элементов компьютер из-за разброса параметров даже для исправных микросхем. Например, Вы вынимаете микропроцессор из панельки и втыкаете его в исправный компьютер, он оказывается в порядке, а на Вашей плате - "не тянет".

Иногда компьютер работает, но часто "сбоит", сбрасывается, зависает, то ли после прогрева, то ли от помех по сети, то при нажатии клавиш. (Кстати, если работа компьютера нарушается при нажатии клавиш, то по ним можно вычислить перегруженную шину и буферизировать хотя бы одну ее). В общем, весь этот "полтергейст" может быть связан с превышением предельного уровня нагрузки микропроцессора. Разработчики фирмы Синклер руководствовались, видимо, соображениями дешевизны, когда не заботились о периферии.

СПРАВКА ИНФОРКОМА

Разработчики фирмы "СИНКЛЕР" руководствовались соображениями эффективности технологии производства, ставя перед собой задачу всемерного повышения надежности главной платы в автоматизированном производстве. Поскольку надежность микросхем обеспечивалась автоматизированным входным контролем, надежность готовой платы зависела в основном от количества паяных и тем более непаяных контактных соединений. Были приняты все меры, чтобы свести их к минимуму. В этом случае ответственность за надежность работы машины совместно с периферией перекладывалась на плечи разработчика периферии.

Хотя конечно же такой подход дал возможность значительно снизить цены на готовую машину.

Интересно отметить, что в результате стоимость самой примитивной периферии стала сравнимой со стоимостью компьютера.

Например, стоимость нехитрого интерфейса джойстика на одной микросхеме составляла до 20% стоимости компьютера, но его делал не Синклер. Зато его компьютеры оставались самыми дешевыми в мире.

Пользователи же самодельных Спектрумов, особенно те, кто не слишком озабочен их серийным производством, могут себе позволить "обвесить" компьютер буферными элементами и затем вешать на него всю мыслимую периферию, схемы которой сыплются на головы пользователя, в том числе и стараниями специалистов из НТК "ПЛЮС".

Кстати сказать, когда много лет назад, в 1984 году разрабатывался один из первых вариантов самодельных Спектрумов, впоследствии окрещенный "Московским" вариантом, то в схему и плату компьютера были заложены буфера на K555АП6. Эти буфера можно по желанию устанавливать при работе с периферией или обходиться без них, заменив их перемычками. Они не окружают микропроцессор, а работают только на внешний разъем.

Этот подход мы и рассмотрим в нашей статье. Он подходит нам потому, что не потребует вмешательства в плату и схему компьютера и обращается лишь к внешнему разъему. Правда при этом внутренние шины компьютера не будут буферизованы.

Для буферизации внутренних шин компьютера можно было бы посоветовать следующий способ. Удалить микропроцессор и заменить его на небольшую плату, на которой установлен как сам микропроцессор так и буферные элементы, например 4 штуки K555АП6. Если у Вас запаяна панелька под микропроцессором, то такой эксперимент провести будет не очень сложно. Правда есть одно проверенное правило - лучшее враг хорошего. Поэтому если Ваш компьютер работает вполне устойчиво, не сбоит и вообще не вызывает устойчивых подозрений своей работой, можно обойтись без "хирургического вмешательства", а для подключения разветвленной периферии воспользоваться внешней буферизацией.

Схемы подключения буферных элементов для конкретного микропроцессора достаточно стандартны и изобретать велосипед мы не стали, а взяли эти схемы из зарубежной печати. При этом мы исходили из доступной отечественной базы.

Шины микропроцессора делятся на входные, выходные и двунаправленные. Проще всего с входными шинами, такими как /WAIT¹, /NMI, /INT, /RESET и т.д. Их вообще не нужно буферизировать.

Выходные сигналы, требующие усиления - это 16 линий шины адреса - A0-A15, а также сигналы управления, такие как /RD, /WR, /IORQ, /MREQ и т.п.

Немного сложнее обстоит дело с двунаправленной шиной данных D0-D7, потому что для этого требуется применение двунаправленных микросхем и управление переключением направления передачи данных.

На рис.1 и 2 приведены схемы буферов шины адреса и управления, выполненные на распространенных микросхемах K589HP12. Здесь и далее обозначения сигналов со штрихом относятся к усиленным сигналам. Номиналы резисторов не критичны и могут быть в пределах 10-20 Ком.

Более сложная схема буфера шины данных приведена на рис. 3.

Правда, если заменить доступные микросхемы K589HP16 на более дефицитные, например, на восьмиразрядный буфер K555АП6, то схема будет по крайней мере выглядеть более простой, да и количество паек уменьшится. Сложность управления буферами объясняется стремлением достичь полной универсальности в их работе.

Здесь приняты меры к тому, чтобы не было конкуренции на шине данных "внутри" компьютера между "внутренними" источниками и буфером, включенным на ввод данных по сигналу "на запись" (-WR без инверсии) и вовремя не переведенным в высокоимпедансное состояние.

Для этого служит сборка 8И-НЕ, выполненная на K555ЛА2. Сборка "отлавливает" те моменты, когда внешние устройства (устройства находящиеся за буфером) не выбраны, т.е. находятся в третьем состоянии и отключает буфера от шины. Кроме того, этот "диспетчер шины данных" отключает /CS на буферах еще и тогда, когда от внешних устройств поступает байт FF (все единицы, ни одного нуля на шине данных), что вообще говоря неправильно, но в этом случае роль байта FF выполняет высокоимпедансный выход буфера и резисторы R1-R8. Кстати, эти резисторы можно не ставить потому, что они как правило уже есть в компьютере. Чтобы это проверить, нужно прочитать данные из незадействованного адреса внешнего порта, и если они равны FF(255), то это говорит о подтянутости шины данных к плюсу.

Далее по схеме на рис.3. Если перемычка Е запаяна, то буфер полностью универсален. При удалении перемычки буфер становится зависим от сигнала /IORQ. Т.е. буфер будет работать только с внешними устройствами, адресованными как порты ввода/вывода. От перемычек ABCD зависит то, какие сигналы используются в управлении буфером - непосредственно от микропроцессора (в таком случае желательно в качестве инверторов применить микросхемы из серии ТТЛШ для уменьшения нагрузки на микропроцессор) или взятые после буфера сигналы управления.

Если Вы используете буфер шины управления, то есть смысл использовать усиленные сигналы /RD и /WR. Опыт показывает, что возникающая при буферировании задержка около 30 Нс в данном случае не существенна.

Оставшиеся незадействованными два элемента 2 ИЛИ-НЕ можно использовать для выработки сигнала очень полезного при работе с периферией, связь с которой устанавливается по прерыванию. Этот сигнал INTA (аналогичен сигналу INTE микропроцессора KP580ИК80 (I8080)) и служит для подтверждения полученного прерывания. У микропроцессора Z80 нет такого сигнала, но сигналы /M1 и /IORQ свидетельствуют о выборке вектора прерывания с шины данных, поэтому их комбинация может служить в этом качестве.

На рис.4 представлена схема дешифратора адресов портов ввода-вывода. Это также необходимая вещь, если Вы присоедините к компьютеру периферию. Дело в том, что в фирменном и почти во всех самодельных компьютерах Спектрум очень грубо дешифрируется адресное пространство ввода/вывода, а именно - все четные адреса (при

¹ Инверсные сигналы обозначены префиксом / (Прим. NUK)

A0=0) задействованы под внутренний порт с адресом 254, который "занимается" бордюром экрана, магнитофоном, звуком и клавиатурой. Кроме того, по концепциям фирмы под периферию, выпускающуюся для Спектрума заняты адресные линии A5 и A6.

Некоторые пользователи жалуются на неработающую периферию, в том числе и сделанную по нашим схемам (НТК "ПЛЮС"). Часто причина этого в том, что в схеме уже стоит какой-либо грубо адресованный порт, и в результате возникают конфликты на шине, т.е. выбирается одновременно более одного источника.

Если Вы хотите избежать этих трудностей при Ваших экспериментах, используйте более точный дешифратор адресов. Дешифратор, изображенный на схеме 4, рассчитан на 8 периферийных устройств и при желании может быть расширен дальше. Следующий дешифратор K555ИД7 можно присоединить, используя в качестве сигналов выбора вывод 8 микросхемы DD3 и адресную линию A10 и т.п.

Адресные линии A5, A6 не используются, чтобы не конкурировать с наработанными периферийными устройствами с "кемпстоновской" адресацией.

На рис.5 приведена для примера буферная схема для шин адреса, данных и управления, выполненная на микросхемах KP589BA86. Адресную шину можно буферизировать полностью, добавив еще один элемент аналогично DD1.

Буфер в шине управления на DD3 тоже не нуждается в комментариях. Особенностью этой схемы является управление переключением направления буфера данных на DD2.

В связи с тем, что направление данных в микросхемах KP589BA86 нельзя изменять во время их активного состояния, как это делалось в K589АП16, в схему введен RS триггер.

Как Вы могли заметить, данная схема буферизации не универсальна, она рассчитана на работу с портами ввода/вывода и управляется сигналами выбора /CS тех внешних устройств, которые подключены к вашему компьютеру.

Если таких устройств несколько, то для объединения их сигналов выбора /CS можно применить сборку, например 8И-НЕ, как показано на рисунке. Теперь при адресации к любому из внешних портов любой из сигналов /CS переведет буфер шины данных, выполненный на микросхеме DD2 из высокоимпедансного состояния в активное. Направление передачи данных через буфер в это время удерживается RS триггером, включенным сигналом /WR или /RD в одно из двух состояний. Заметим, что так как на сборку сигналов CS заводятся сигналы выбора только от устройств, подключенных после буфера, то при выборе какого-либо внутреннего порта буфер будет находиться в высокоимпедансном состоянии и таким образом не возникает конфликта на шине данных.

Приведенные здесь схемы вполне работоспособны, но мы ставили своей целью осветить общие подходы к буферизации шин микропроцессора. Пользователи Спектрума могут использовать эти подходы при разработке собственных схем внешних устройств. Применяемые Вами в качестве буферных элементов микросхемы могут быть очень разнообразными, в зависимости от возможности их приобретения, потребляемого тока, размеров и т.д.

Кроме упомянутых в статье микросхем, в качестве буферов часто применяются K155ЛН1, 2, 3, 4, 5 - буфера без выводов разрешения; K155ЛН6, K155ЛП7, ЛП8, ЛП10 - буфера с выводом разрешения; K531АП3, АП4, K155ИП6, ИП7, K555АП5, АП6 - двунаправленные буфера с выводами разрешения и т.д. и т.п.

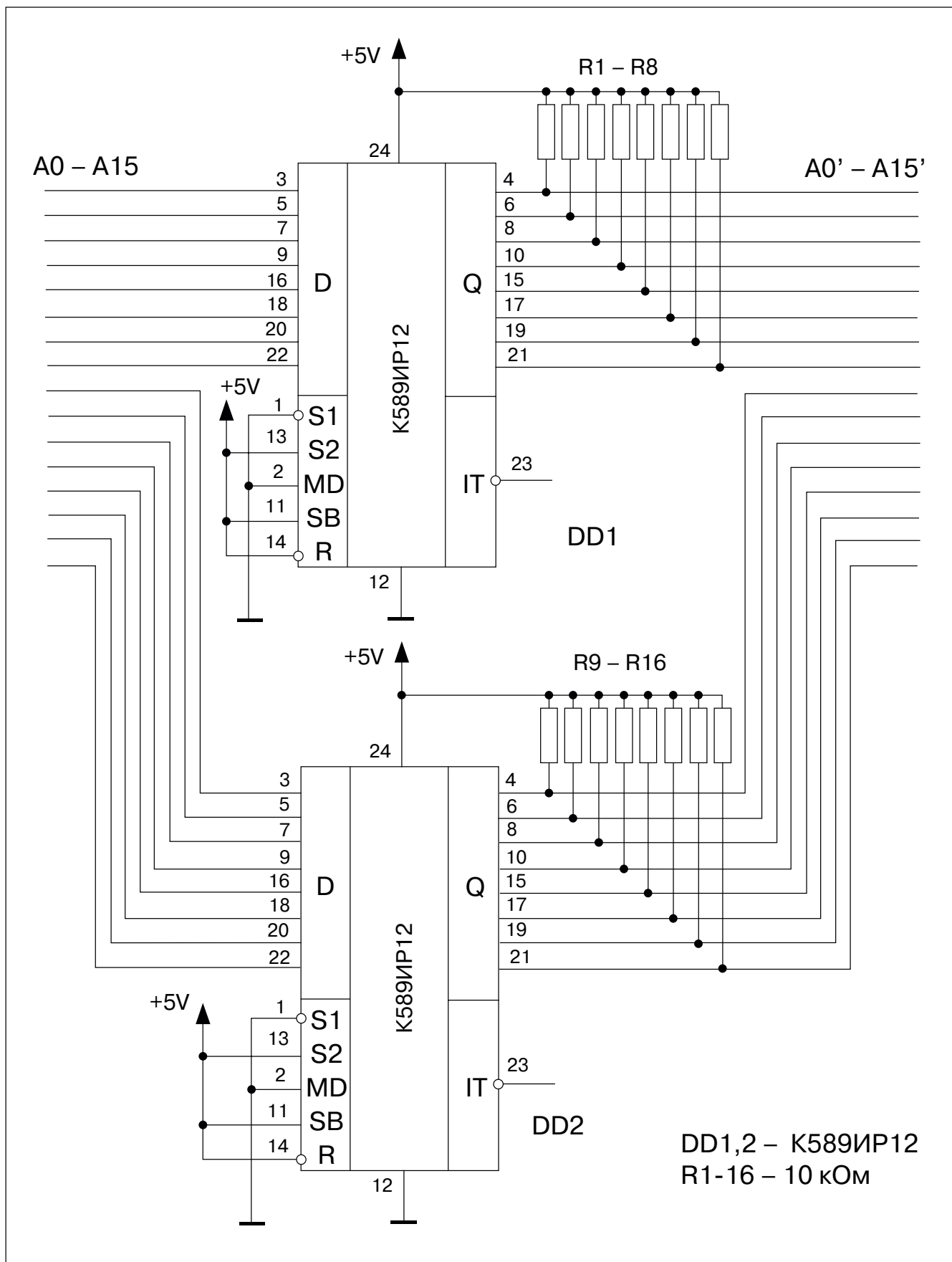


Рис. 1. Буфер шины адреса A0 - A15

В заключение рассмотрим одну важную особенность фирменных компьютеров SPECTRUM, которая в последние годы стала иногда использоваться разработчиками фирменных программ для Спектрума, что привело к проявлению неполной совместимости самодельных компьютеров с оригиналом. Речь идет об устройстве шины данных компьютера. Мы уже обсудили тот факт, что при чтении данных из несуществующего порта (или из неиспользуемого адреса порта), мы получаем данные FFH (или 255D). Это связано с тем, что тогда, когда не выбрано ни одно устройство, подключенное к шине данных, то все они находятся в высокоимпедансном состоянии. Но шина данных через резисторы 10-20 К подключена к плюсу источника питания. Поэтому все разряды шины данных принимают состояние логической единицы, что в результате дает данные FF(255).

Т.е.

```
10 PRINT IN 1
20 GO TO 10
```

даст результат:

```
255
255
```

```
...
```

```
255
```

```
SCROLL?
```

Однако если эту программку запустить на фирменном Спектруме сразу же после включения или сброса, то мы увидим, что среди результата 255 иногда будет "проскальзывать" результат 56.

Оказалось, что это число зависит от атрибутов экрана. Например, если задать цвет фона (PAPER) черный, а цвет символов (INK) - белый, то это число, иногда считываемое в программе, будет равно 7, т. е. это именно байт атрибутов, считываемый из области 768 байтов атрибутов, которая расположена после области 6144 байтов графической информации экрана (Мы исходим из того, что карта памяти компьютера Вам хорошо знакома, хотя бы по книге ИНФОРКОМА "Большие Возможности Вашего СПЕКТРУМа").

Остается выяснить, каким образом, считывая данные из порта с адресом 1 (этот адрес взят для примера, можно взять любой адрес порта, не используемый в данной конфигурации), мы получаем в результате данные из области атрибутов экрана.

Для этого нужно рассмотреть устройство внутренней шины данных фирменного компьютера Спектрум. У фирменного Спектрума непосредственно к шине данных подключены: микропроцессор, ПЗУ, поле памяти 16К (нижнее), дополнительное поле памяти 32К (верхнее) и внутренние регистры микросхемы ULA - программируемой логической матрицы, выполняющей в Спектруме функции контроллера дисплея, генератора тактовой частоты процессора и порта ввода/вывода с адресом 254 (а на самом деле с любым четным адресом), для ввода и вывода данных магнитофона, вывода звука, ввода данных от клавиатуры, вывода цвета бордюра.

Причем ULA работает автономно совместно с областью памяти 16К, в которой как раз и выделено место для памяти экрана 6912 байт (растровая графика плюс атрибуты). Периодически, в соответствии с частотой кадров и строк телевизионной развертки, ULA вырабатывает адреса области экрана и считывает 32 раза в течение каждой строки по два байта - один байт из области растровой графики и один из области атрибутов экрана, так как микропроцессор также имеет доступ к этой области памяти, то, если не принять специальные меры, может возникнуть конфликт на шине данных.

Чтобы этого не произошло, в фирменном компьютере приняты следующие меры. При одновременном обращении к полю памяти 16К со стороны ULA и микропроцессора, ULA останавливает тактовую последовательность, поступающую к микропроцессору Z80 и, таким образом, Z80 "ждет" возможности работать дальше, т.е. ULA имеет приоритет на работу с этими микросхемами памяти. В то же время, независимо от работы ULA с памятью экрана, микропроцессор может продолжать работать с дополнительной памятью 32К, с ПЗУ и внешними портами.

Чтобы при этом не возникал конфликт на шине данных, шина данных разделена. ULA и поле памяти 16К отделены от остальных пользователей шины данных резисторами 410 Ом.

При этом ULA и микропроцессор по отдельности пользуются каждый своим "куском" шины данных, не слишком мешая друг другу. Теперь вернемся к работе нашей программы.

Когда микропроцессор считывает данные из порта 1, то состояние шины данных на участке, "принадлежащем" микропроцессору, определяется резисторами 8.2 Ком, которыми линии подтянуты к плюсу источника питания. В то же время, к микропроцессору через резисторы 470K поступают данные, считываемые ULA из памяти экрана при его регенерации и при совпадении моментов считывания эти данные воспринимаются микропроцессором, как данные из порта 1. Т.е. в фирменном компьютере при считывании с внешнего порта с неиспользуемым адресом можно получить данные из области атрибутов экрана.

Это обстоятельство используется в некоторых фирменных программах возможно для более точной синхронизации события программы и изображения на экране, во всяком случае, то обстоятельство, что в самодельном компьютере шина данных полностью развязана и на ней всегда присутствует лог."1" по всем линиям при невыбранных устройствах, является причиной несовместимости некоторых программ.

Например, здесь нужно упомянуть третий выпуск ZX-РЕВЮ (раздел "ФОРУМ"), в котором наш читатель рекомендовал как сделать работоспособной программу ARKANOID-1. В самодельных компьютерах с отдельным полем памяти для достижения подобного результата можно было бы еще посоветовать соединить входы и выходы микросхем памяти поля 16К с помощью резисторов 470 Ом.

Ввиду большого разнообразия разработок самодельных компьютеров трудно дать рекомендации всем пользователям Спектрума, но мы думаем, что своими публикациями мы хотя бы чуть-чуть поможем Вам разобраться в некоторых тонкостях работы Вашего компьютера.

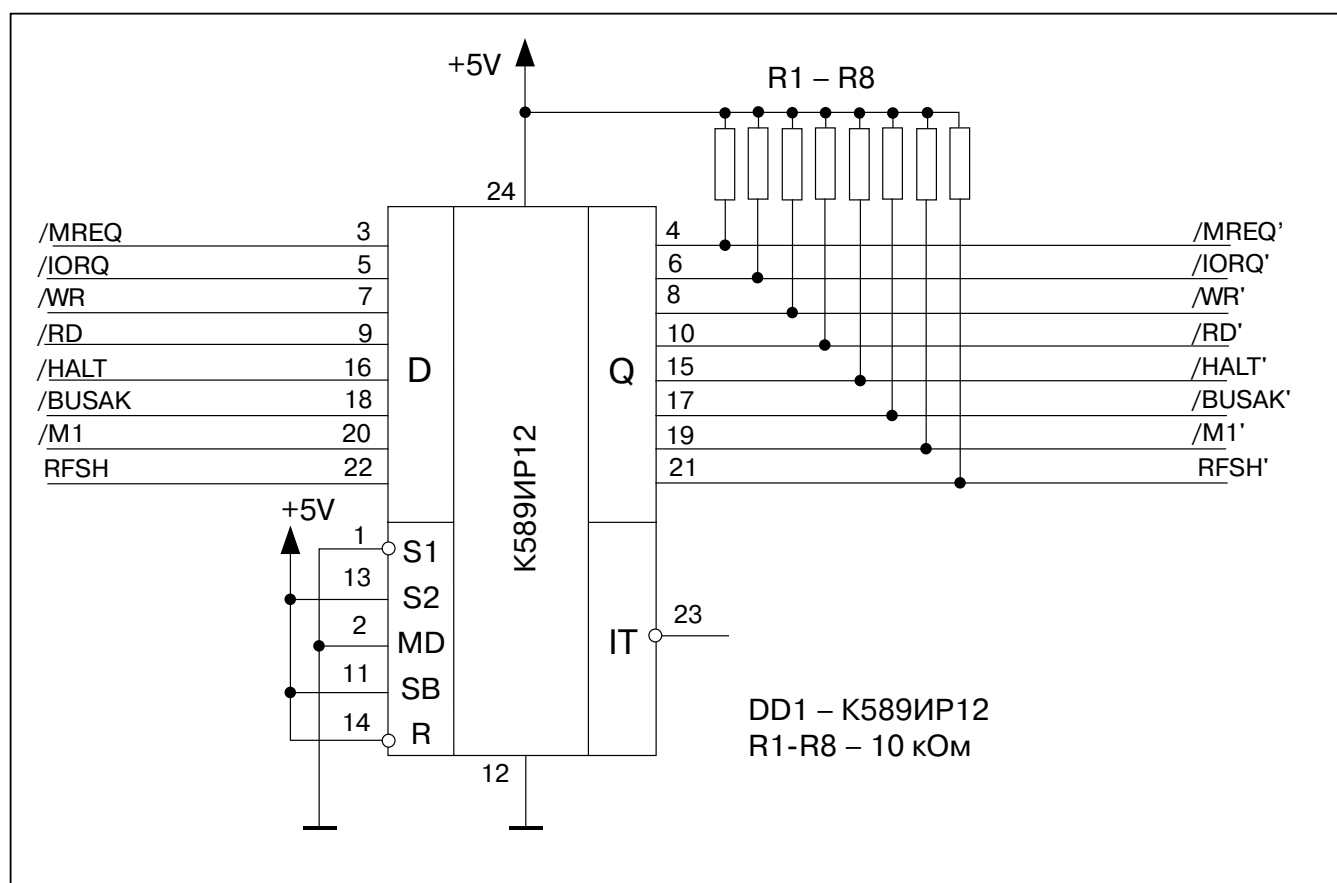


Рис.2. Буфер шины управления.

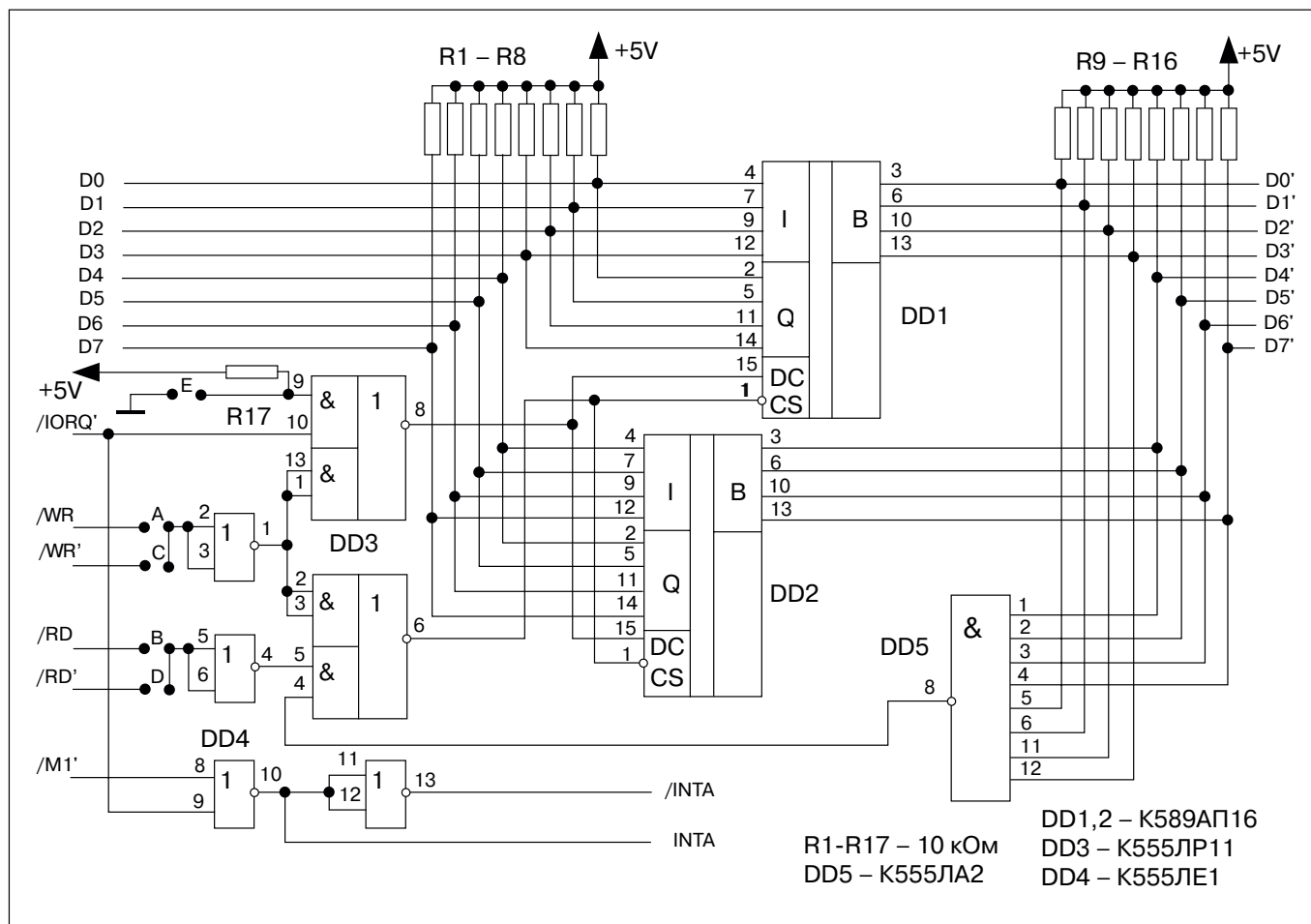


Рис.3. Буфер шины данных.

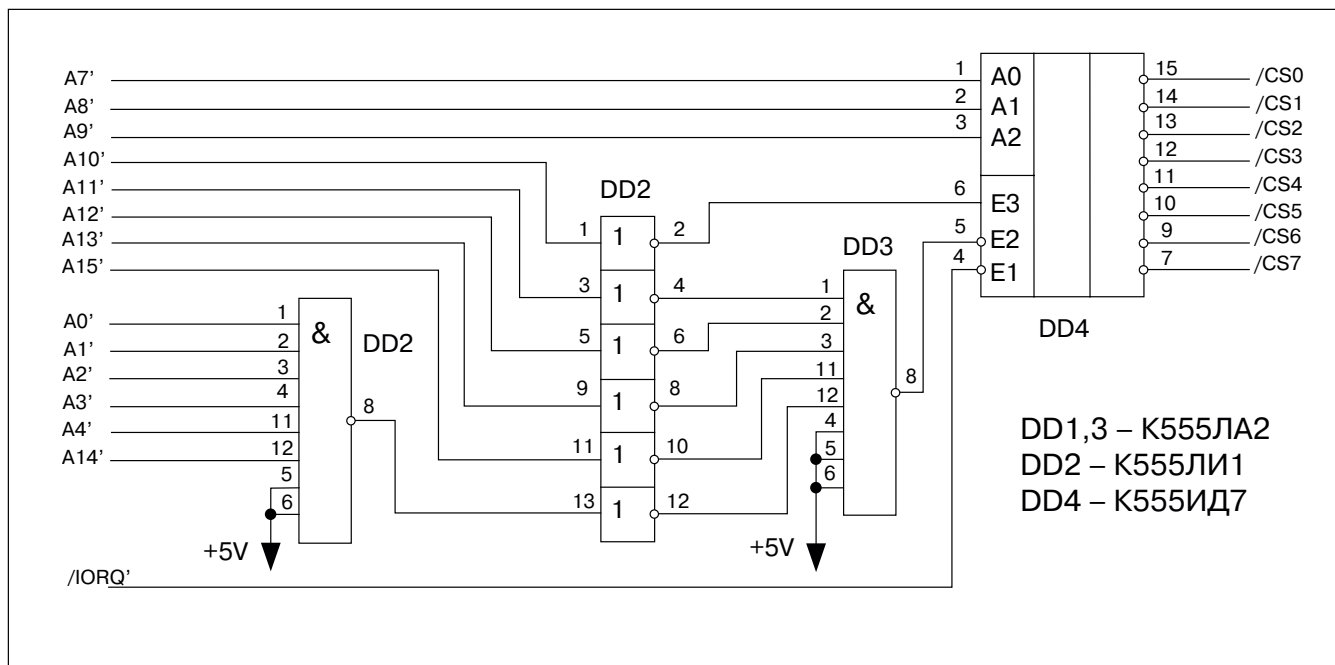


Рис. 4 Дешифратор адресного пространства ввода/вывода

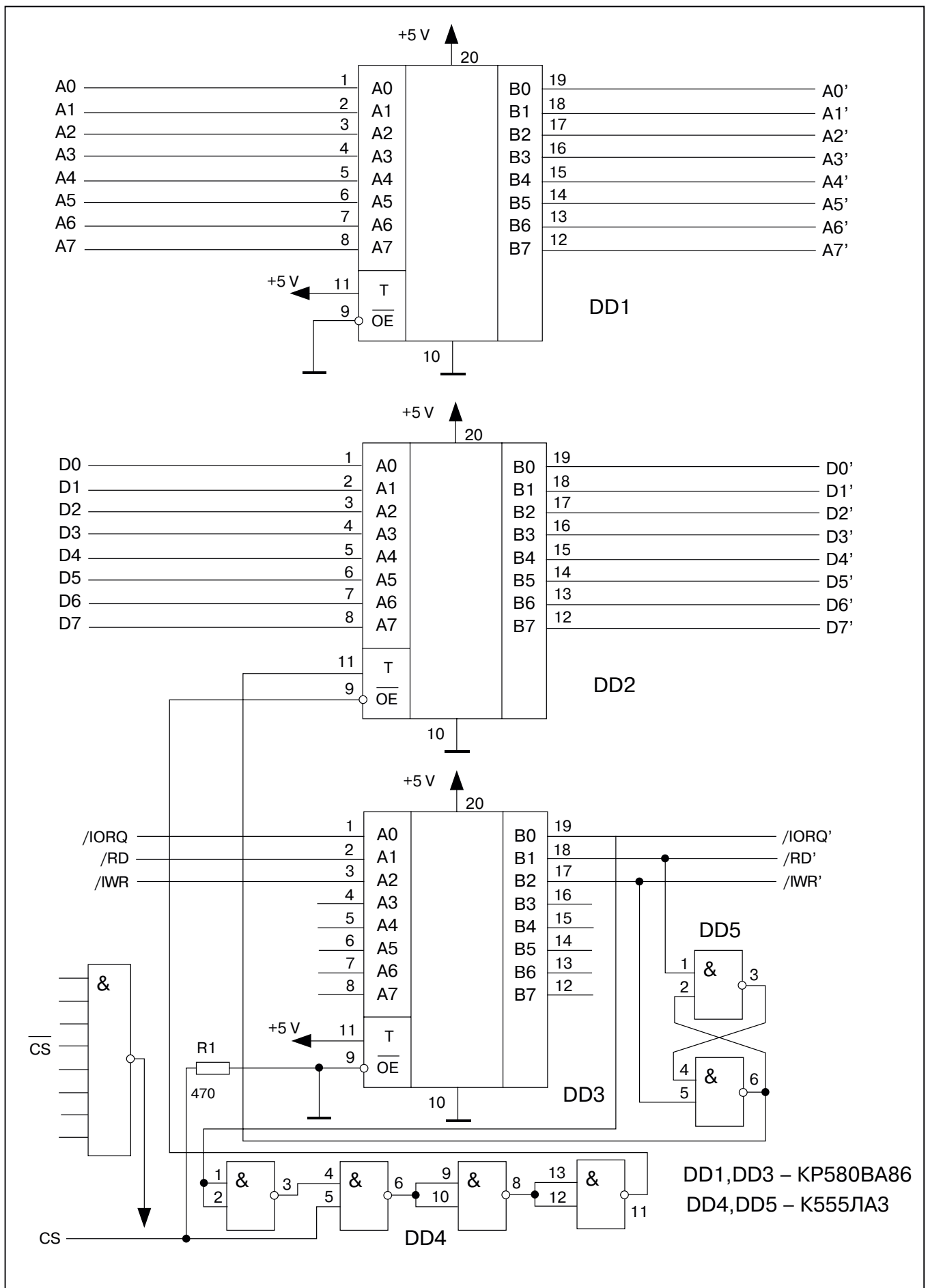


Рис.5. Универсальный буфер

ZX-РЕВЮ РЕКЛАМНОЕ ПРИЛОЖЕНИЕ АПРЕЛЬ-МАЙ 1991

НТК "ИНФОРКОМ" не может принимать на себя никакую ответственность за точность и достоверность сведений, содержащихся в объявлениях читателей.

Имею 1000 программ для ZX-Spectrum.
Предлагаю обмен. Всем желающим вышлю каталог.

614002 Пермь, ул.Фонтанная 9-105
Носкову Владимиру Анатольевичу.

Предлагаю:
Запись на ГМД размер 5.25" различных
игровых и прикладных программ.
ГМД - мои или заказчика. Каталог вышлю.

326520, Херсонская обл, г.Цурюпинск,
пер.Зеленый- 2
Алексеев А.А.

Обменяю системные, прикладные,
сервисные, учебные и игровые
программы.

Телефон в Бердске: 5-32-62. Адрес:633190
Новосибирская обл, г.Бердск,
ул.К.Маркса, 66 кв.86 Кобылинский Юрий.

Ищу принтер с интерфейсом для ПК
"Спектрум".

334264 Крым, г.Гурзуф, ул.Соловьева 6
кв.33 Сорокину В.В.

Предлагаю в обмен на кассеты,
дискеты и радиодетали более 800
системных и игровых программ к ПК типа
"ZX SPECTRUM". Запись на кассетах
заказчика.

Обращаться по адресу:
460048 Оренбург, пр.Победы д.144"А"
корп.2 к.572 Щедринов.

Предлагаю ППЗУ (573РФ5) с
тестами для проверки ОЗУ и отладки плат
"Синклер-48К" (30 руб.) и "Синклер-128К"
(35 руб.).

Обращаться по адресу: 617740
г.Чайковский, ул.Кабалевского 28-75
Дубинец Игорь Николаевич

Всем владельцам 43-х корпусного
варианта ПК "Spectrum" (вариант Зонova)
предлагаю несложную аппаратную
доработку. Она превратит его из обычной
игрушки в очень нужный и серьезный
прибор.

Доработка позволяет:

1. Подключить к компьютеру помимо
джойстика, "естественную" периферию -
принтер и контроллер дисководов.

2. Подключить любые разработки
НТК "ПЛЮС" (программатор, все типы
джойстиков, световое перо, интерфейсы
RS232 и параллельный для принтера.)

3. Кроме того, в распоряжении
пользователя останется еще 32700 портов
ввода-вывода, т.е. неограниченное
количество аппаратуры, которая может
быть подключена к шине данных
компьютера.

Все вышеперечисленное в
оригинале невозможно. Стоимость схемы
и описания (включая пересылку) - 27
рублей.

Оплата наложенным платежом.

Гарантирую четкую и безотказную
работу всем приславшим заказ.

Письма-заявки присылать по адресу:
460040 Оренбург, пр.Гагарина д.23, кв.24
Чалову Юрию Анатольевичу.

Куплю принтер к ПК "SPECTRUM".

Обращаться по адресу: 338042 Донецкая
об. г.Горловка, ул.Бессонова 33/310
Павловскому Г.В. (тел.3-96-08)

Запишу программы к ПК "Спектрум".
Имею около 400 фирменных программ.
Качество гарантирую. Вышлю каталог.
Возможен обмен программами.

Адрес: 330001, г.Запорожье-1, а/я
160. Юра.

Предлагаю на обмен программы 87-89 г.г. для компьютера ZX-Spectrum.
Куплю джойстики - аналог VS-125.

Адрес: 678300 ЯСССР, п.Сангар,
ул.Алексеева 17-9. Нечетайленко А.В.

Вышлю наложенным платежом
печатные платы ПК "Синклер".

Варианты: ленинградский,
московский, "Пентагон"

- 90 руб. за шт.

Микросхемы, проверенные в ПК
KP565PY5

- 180 руб. за 8 шт.

Документация к печатным платам
10-15 руб.

Обращаться по адресу: 606400
Нижегородская об. г.Балахна-2, до
востребования Пухову Юрию Викторовичу.

Предлагаю программы на кассетах
SONY, BASF, TDK по 40 руб. за кассету с
20-ю программами. Кассеты вышлю
наложенным платежом. Каталог вышлю
бесплатно.

Обращаться по адресу: 364052 ЧИССР.
г.Грозный, гор-к Иванова дом 106 кв.10
Кирилину Андрею Николаевичу.

Предлагаю ежемесячное издание
"48Кбайт". Стоимость годовой подписки 60
рублей, объем издания 22-30 страниц.
Запись игровых и системных программ,
(в коллекции около 2 тысяч программ),
большой выбор описаний игровых и
системных программ, в т.ч. и авторских.
Каталоги и рекламные листки высылаются
бесплатно.

Обращаться по адресу: 164413.
Архангельская об, п/о Катунино,
ул.Катунина 4-20. Костюченко Ю.К.

Предлагаю программы "BIORITMICS",
"TESTS", "CHIROMANT". Авторские
разработки. Объем каждой около 40К.
Адрес: 640020 г.Курган, ул.Климова д.56,
кв.17. Сорокин Алексей Вячеславович.

Продаю компьютеры "ZX-SPECTRUM"
(вариант "Ленинград-2" с кодером: RGB,
BRIGHT, BEEP CEKAM-6 канал) с
техописанием по ценам ниже рыночных.

Гарантийный срок - 3 года.

Продаю программы, сгруппированные по
тематике, на 20 кассетах МК-60.

256300 Киевская обл. г.Борисполь,
ул.Котовского д.3. кв.77

Кысиль Григорий Николаевич.

Продам Спектр-128 с любой
периферией (монитор, дисковод, принтер)
и программным обеспечением. Форма
оплаты любая.

Обращаться по адресу: 117463,
Москва, В-463, а/я 97 Шаликов В.В.

С высоким качеством, оперативно и
дешево запишу на кассету заказчика или
свою описания игровых и системных
программ и программы.

Обращаться по адресу: 626440
Нижевартовск. ул.Чапаева 85"Б"-188.
Карша С.Р.

Продаю, покупаю и обмениваюсь
программами ZX SPECTRUM. Обмен и
продажу обеспечиваю своими кассетами.

Обращаться по адресу: 245110
Сумская об, г.Шостка, ул.Франко, 24.
Луговому Владимиру Викторовичу.

Продам ИС серий: 142; 155; 555;
561; 580; 1810. Налаженные платы "ZX-
Spectrum" (6 штук). Чистящие дискеты (15
штук).

Обращаться по адресу: 634001
Томск, пер.Островского 8-60. Тристан С.В.

Предлагаю обмен программами
владельцам "Спектрума" и других
совместимых ПК, проживающим на
Дальнем Востоке и в Приморье.

Адрес: 692900, г.Находка-18,
ул.Постышева д.49-22 Евтушенко В.А.

НАШ КОНКУРС

Сегодня мы подводим итоги ранее объявленного конкурса на лучшую тематическую подборку игровых программ.

На конкурс мы получили 53 сборника от 30 читателей. К сожалению, большинство или невнимательно читали его условия или не смогли их правильно выполнить. На Ваш суд выносятся только 11 проектов 7-ми авторов.

Напоминаем условия конкурса:

1. Списки должны содержать программы, объединенные единым тематическим содержанием.

С этим условием как-то справились все, поскольку абсолютного критерия нет. А уж насколько успешно это получилось - судить Вам.

2. Количество программ в подборке должно быть таким, чтобы полностью заполнять 90- минутную кассету. С этим не справился только один конкурсант из п. Сельстрой Ленинградской обл.

3. Программы, вошедшие в список, должны иметь фирменное название, название фирмы-производителя и год выпуска.

На этом "срезалось" большинство. Очень многих подвел Bill Gilbert и другие подобные "хакеры" и пиратские фирмы, ворующие чужое программное обеспечение и оставляющие на нем свою "марку".

Многие написали названия фирм не для всех программ, включенных в сборники, поскольку взять их было негде, а может быть просто забыли.

4. В списки не должны включаться программы, ранее объявленные в наших сборниках "Детский Альбом-1", "Детский Альбом-2" и "HURRY UP".

Несколько человек, включивших в свои сборники программы типа Bruce Lee, Robin in the Wood, River Raid, Batty и т.п. не были допущены к конкурсу по этой причине. Правила есть правила и надо было внимательней читать условия.

Огромным подвохом для многих оказалась игра Green Beret. Эта программа оригинально разработанная японской фирмой KONAMI и конвертированная для "Спектрума" фирмой IMAGINE пользуется по-видимому очень большой популярностью среди "пиратов", поскольку каждый, кто включал ее в список, приводил самые невероятные названия фирмы-разработчика.

Призом в конкурсе является контракт на разработку инструктивного буклета к сформированной подборке программ. Стоимость 1-го контракта 1200 руб., второго - 1000 руб., третьего - 800 руб.

Многие читатели указывают, что далеко не всегда можно в программе точно установить фирму производителя и год выпуска, и это ограничение очень сокращает возможный круг поиска. Это действительно так, но вот вопрос включать или не включать такую программу в список решали Вы сами.

Нашей задачей было найти людей, умеющих добывать информацию, а не просто имеющих и любящих игровые программы.

Но и мы вынесли для себя кое-что полезное. Теперь мы лучше представляем сколько же ходит по стране "изувеченных" программ с неграмотно переделанным названием и с указанием сомнительного "авторства".

Чтобы как-то помочь в преодолении такого положения, а заодно поощрить участников, мы поступим так:

Всем участникам конкурса, в том числе и тем, чьи предложения не были допущены к заключительному этапу, в начале сентября мы бесплатно вышлем готовящийся сейчас каталог-справочник по программному обеспечению, включающий около 5000 программ. Он окажет Вам необходимую помощь в Вашей работе с компьютером. Будет он также чрезвычайно полезным для проведения взаимных обменов программами и систематизации своей библиотеки.

Итак, представляем присланные на конкурс сборники, они приведены под теми девизами, которые им дали сами авторы. В тех случаях, когда такого девиза не было, мы будем писать NO NAME 1, NO NAME 2 и т. д.

В обсуждении итогов могут принять участие все желающие. Надо только написать нам и указать какой сборник Вы ставите на 1-е место, какой на второе, а какой на третье. За первое место будет начислено 5 очков, за второе - 3, за третье - 1.

Если найдете в списках "дезинформацию", то можете начислить этому автору 10 штрафных очков.

Мы допускаем, что все-таки где-то в представленных списках "дезинформации" просочилась, поскольку мы тоже многого не знаем, хотя что смогли, мы постарались отсеять.

В целом мы оцениваем предварительные результаты первого этапа как крайне неудовлетворительные.

Возможно, что условия оказались слишком жесткими для основной массы любителей. Посмотрим, как он будет развиваться дальше, когда Вы отберете три лучших сборника и мы предложим их авторам написать серьезный буклет описаний в обоснование своего проекта.

Не удовлетворившись такими итогами, мы начинаем новую кампанию - набор в заочные творческие группы "ЭКСПЕРТОВ" и "ОБОЗРЕВАТЕЛЕЙ" с очень высоким уровнем оплаты но и очень высокими требованиями. Желающие попробовать свои силы могут нам написать.

Подробности в следующем объявлении "Рекламного Приложения".

ЗВЕЗДНЫЕ ВОЙНЫ

1	Elite	FIREBIRD	1884
2	Cybernoid 2	HEWSON	1988
3	Task Force	PREMIER	1989
4	Star Wars	DOMARK	1987
5	Phantis 1	DINAMIC	1987
6	Dark Star	DESIGN DESIGN	1986
7	Zynaps	HEWSON	1987
8	Xecutor	ACE	1987
9	Sabotage	ZEPPELIN	1987
10	Uridium	HEWSON	1987
11	Empire	DOMARK	1988
12	Virus	TELECOM	1988
13	Nemesis	KONAMI	1987
14	StarFight	HYBRID	1987
15	Cyberun	A. C. G.	1985
16	Death Star	SYSTEM 3	1985
17	EarthLight	TELECOM	1988
18	Unitrax	DOMARK	1987

ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 1

1	Freddy Hardest 1	DINAMIC	1987
2	Freddy Hardest 2	DINAMIC	1987
3	Dan Dare	VIRGIN GAMES	1986
4	Quartet	ACTIVIZION	1986
5	Defender of Earth + 3 levels	ENIGMA VARIATIONS	1990
6	Return of Jedi	DOMARK	1987

7	Side Arms	CAPCOM	1987
8	Zub	MASTERTRONIC	1987
9	Forgotten Words + levels	U. S. GOLD	1987
10	North Star	GREMLIN	1988
11	Dark Fussion + 4 levels	GREMLIN	1988
12	The Ice Temple	BUBBLEBUS	1986
13	Dynatron	MASTERTRONIC	1987
14	Hypper Active	SPECIAL FX SOFT	1988

ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 2

1	Alien 8	ULTIMATE	1985
2	Venom Strikes Back	GREMLIN	1988
3	Enigma Force	BEYOND	1985
4	Gun Runner	HEWSON	1987
5	People from Sirius	U. S. GOLD	1987
6	Omega One	MASTERTRONIC	1987
7	Shadowfire	BEYOND	1985
8	Blind Panic	MARTECH	1988
9	Catch 23	MARTECH	1987
10	Phantis 1	DINAMIC	1987
11	Phantis 2	DINAMIC	1987
12	Crosswize	FIREBIRD	1989
13	Prodigy	ELECTRONIC DREAMS	1986
14	Future Knight	GREMLIN	1986
15	Amaurote	MASTERTRONIC	1987
16	Nodes of Yesod	ODIN	1985
17	Antiriad	PALACE	1986
18	Equinox	MIKROGEN	1986

ВОЗДУШНЫЕ СРАЖЕНИЯ

1	Ace	CASCADE	1986
2	Ace 2	CASCADE	1987
3	Cobra Force	PLAYERS PREMIER	1989
4	A.T.F.	DIGITAL INTEGRATIONS	1988
5	Sanxion	THALAMUS	1989
6	Thunder Blades + 5 levels	U.S. GOLD	1988
7	Flyer Fox	GAME GEMS	1984
8	Tomahawk	DIGITAL INTEGRATIONS	1985
9	Top Gun	OCEAN	1986
10	Skyranger	BE BE SOFT	1986
11	Operation Hormuz	DURELL	1985
12	Air Wolf	ELITE	1988
13	Typhoon + levels	OCEAN	1989
14	1943 of Midway	CAPCOM	1988

СПОРТИВНЫЕ ИГРЫ

1	Pro Ski	CODEMASTERS	1987
2	World Games + 8 levels	ЕПУХ	1987

3	Uchi Kata 2	MARTECH	1988
4	Indoor Soccer	CODEMASTERS	1988
5	Supertest 1	OCEAN	1986
6	Show Jumping	ALLIGATO	1986
7	Basket Master	DINAMIC	1987
8	Passing Shot	IMAGE WORKS	1989
9	Xeno	A&F	1986
10	Basketball	IMAGINE	1985
11	Skate of Die	KINETIC DESIGN	1989
12	International Soccer	AUDIOGENIC	1988
13	Frank Bruno's Boxing	ELITE	1988
14	Play for Your Life	IMAGINE	1988

КОСМИЧЕСКИЕ ВОЙНЫ

1	Ad Astra	GARGOYLE GAMES	1984
2	Code-Name Mat	MICROMEGA	1984
3	Code-Name Mat 2	DOMARK	1985
4	Death Star	SYSTEM 2	1985
5	Dark Star	DESIGN	1989
6	Elite	FIREBIRD	1986
7	Exolon	HEWSON	1987
8	Lunar Jetman	ULTIMATE	1983
9	Jetpac	ULTIMATE	1983
10	Luna Crabs	MICROMEGA	1983
11	Mercenary	NOVAGEN	1987
12	Planetoids	PSION	1982
13	Quazatron	PSION	1985
14	Space Wars	HEWSON	1983
15	Trantor	PROBE	1987
16	Venom	GREMLIN	1988
17	Zub	M. A. D. GAMES	1986
18	Zaxxon	U. S. GOLD	1986
19	Academy	CRL	1986
20	D.N.A. Warrior	ARTRONIC	1989
21	R-Type	IREM CORP.	1987
22	Thunderceptor	OCEAN	1989

БЕГИ И СТРЕЛЯЙ ВО ВСЕ, ЧТО ДВИЖЕТСЯ

1	Attic Attack	ULTIMATE	1983
2	Android-1	VORTEX	1983
3	Android-2	VORTEX	1983
4	Avenger	GREMLIN	1986
5	Black Magic	PARAGON	1987
6	Batman	OCEAN	1986
7	Elevation Act	TAITO	1987
8	Kane	BYTE	1986
9	Eagle Nest	PANDORA	1986
10	Fred	INDESCOMP	1985
11	Gunfright	ULTIMATE	1985
12	Ghost'n Goblin	JAPAN CAPSULE	1986

13	Game Over	DYNAMIC	1985
14	Ice Palace	ELITE	1988
15	Ikari Warriors	ELITE	1986
16	Laser Tag		1987
17	Saboteur-1	DURELL	1986
18	Saboteur-2	DURELL	1987
19	Tutankhamoun	MICROMANIA	1983
20	Dan Dare	VIRGIN	1986
21	Cosmo Police	OCEAN	1986
22	Martianoids	ULTIMATE	1987

ЛАБИРИНТ-1

1	Alien 8	ULTIMATE	1985
2	Knight Lore	ULTIMATE	1984
3	Pentagram	ULTIMATE	1986
4	Batman	OCEAN	1987
5	Nosferatu	MACMILLAN	1986
6	Chimera	FIREBIRD	1985
7	Demon's Revenge	FIREBIRD	1989
8	Heartbroken	DYNAMIC	1989
9	Sheer	K.J. BEZANT	1983
10	Panama Joe	SINCLAIR/HUDSON	1986
11	Gauntlet + 21 block	GREMLIN	1985
12	Three Weeks in Paradize	MIKROGEN	1986
13	Pyjamarama	MIKROGEN	1986
14	Nipper 2	GREMLIN	1987
15	Solomon's	PROBE	1987
16	Starquake	S. J. CROW	1985
17	Black Magic	PARAGON	1987

ПИЛОТ БУДУЩЕГО

1	Academy	PETE COOKE	1986
2	Death Star	ATARI CORP.	1985
3	Elite	FIREBIRD	1987
4	Freddy 1	IMAGINE	1988
5	Freddy 2	IMAGINE	1988
6	Chronos	GREMLIN	1987
7	Equinox	MICRO-GEN	1986
8	Dan Dare	VIRGIN	1987
9	L. Force	FTL	1986
10	Trantor	PROBE	1988
11	Amaurote	MASTERTRONIC	1987
12	Zub	MASTERTRONIC	1987
13	Exolon	HEWSON	1987
14	Marsport	GARGOYLE	1985
15	Star Raiders 2	ATARI	1987
16	Earthlight	TARZAN BOY	1989

NO NAME-1

1	Mr. Wimpy	OCEAN	1984
2	Space Raiders	PSION	1982
3	Falcon Patrol 2	VIRGIN	1985
4	Cosmic Pirate	B. C. G.	1983
5	Zipper Flipper	R. E. D.	1984
6	Equinox	MICRO-GEN	1986
7	The Fall Guy	ELITE	1986
8	Roller Coaster	ELITE	1985
9	Rockman	MASTERTRONIC	1985
10	Painter	A+F SOFTWARE	1984
11	Buck Rogers	O. S. A.	1985
12	PSST	A. C. G.	1983
13	Penetrator	C. B. S.	1982
14	Zoom	IMAGINE	1983
15	Trail Blazer	Mr. CHIP S.	1986
16	Twister	SYSTEM 3	1986
17	Hunchback	OCEAN	1983
18	The Devils Crow	MASTERTRONIC	1985
19	Magico la ta Peto	TURBO	1985
20	Ball Breaker	CRL GROUP	1987
21	Crystal Castle	ANDROMEDA	1986
22	PI-Balled	AUTOMATA	1984
23	Armageddon	OCEAN	1983

ЛАБИРИНТ-2

1	Eagles Nest	PANDORA	1988
2	Spirits	TOPO SOFT	1987
3	Great Escape	OCEAN	1986
4	Robocop	OCEAN	1987
5	Game Over 1	DINAMIC	1985
6	Rat Race	OCEAN	1985
7	Scooby Doo	ELITE	1986
8	Atic Attack	ULTIMATE	1983
9	Sabre Wolf	ULTIMATE	1984
10	Knight Lore	ULTIMATE	1984
11	Alien 8	ULTIMATE	1985
12	On the Run	ULTIMATE	1986
13	Chimera	FIREBIRD	1983
14	Droid	FIREBIRD	1986
15	Rygar	U.S. GOLD	1988
16	Hysteria	SOFTWARE PROJ.	1987
17	Dan Dare	VIRGIN	1986
18	Olli & Lisa	ISI	1986
19	Dynamite Dan	MIRRORSOFT	1985

ДЛЯ РАБОТЫ ПО ТЕМАТИКЕ ИГРОВОГО
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ "ИНФОРКОМ" ПРИГЛАШАЕТ К СОТРУДНИЧЕСТВУ
"ХАККЕРОВ" ДВУХ КАТЕГОРИИ

ЭКСПЕРТОВ и ОБОЗРЕВАТЕЛЕЙ

1. Задачи ЭКСПЕРТА.

Экспертом может быть, любой человек, способный с глубоким знанием дела разобраться с интересной программой, литературно изложить ее цели и задачи, порядок ее настройки, порядок работы с ней, дать полезные советы и подсказки. Одним словом - задача ЭКСПЕРТА - дать глубокий всесторонний анализ игры. Понятно, что не всякая программа заслуживает такого подхода. Подобрать ее - тоже задача эксперта. ЭКСПЕРТ отвечает за качество самой программы. Исследование неинтересной "стрелялки" может оказаться малонужным. Особым приоритетом будут пользоваться программы, требующие специальных знаний - адвентюрные, аркадно-адвентюрные, стратегические, менеджмент, имитаторы сложных технических устройств, аналоги традиционных настольных игр и т.п.

2. Задачи ОБОЗРЕВАТЕЛЯ.

Обозревателем может быть только человек, имеющий доступ к самым новым появляющимся в стране программам. В отличие от ЭКСПЕРТА, от него не требуется глубокого уровня проработки программ, а только рецензия (аннотация), написанная интересно и на достаточном художественном уровне, но программа, в отличие от ЭКСПЕРТА, обязательно должна быть новой.

В отличие от ЭКСПЕРТА, ОБОЗРЕВАТЕЛЬ не несет ответственности за качество программы, которую он освещает, то есть его обзор может иметь и негативное содержание, поскольку отрицательная рецензия - тоже рецензия.

3. Порядок выполнения работ.

Вы пишете нам письмо, в котором указываете чем Вы хотите заняться - обзором или экспертной проработкой. Указываете также какие программы Вы берете на себя. Обобщив полученные сведения, поступившие от разных людей, чтобы избежать, дублирования, мы дадим Вам "добро", указав какие из предложенных Вами программ Вам поручаются.

Получив Вашу законченную работу, мы либо принимаем ее в свой портфель и высылаем Вам купон на оплату, либо не принимаем и тогда возвращаем ее Вам с замечаниями которые позволят Вам быстрее набрать необходимый опыт и включиться в производительную работу.

4. Система оплаты.

Оплата и для ЭКСПЕРТОВ и для ОБОЗРЕВАТЕЛЕЙ одинакова - 15 рублей за одну машинописную страницу материала, напечатанного через 1,5 интервала (40 строк) с возможностью ее повышения до 20 рублей по мере приобретения опыта.

Если Вы не имеете возможность присылать материал в машинописном виде, то мы примем и разборчивую рукопись, но уровень оплаты на 30% ниже.

Если Вы имеете доступ к IBM-совместимой технике, то можете присылать материал на дискетах. В этом случае уровень оплаты - на 30% выше. Кодировка символов русского алфавита - альтернативная. Дискеты мы конечно вернем.

5. Система связи.

Связь только, по почте. Для ускорения – ставьте пометку на конверте "О" или "Э". Не посвящайте письмо разнородным вопросам. Принцип "одно письмо - одна проблема" - и оно попадет в нужные руки без блужданий по многочисленным столам.

6. Примеры.

В "Игровом приложении" данного выпуска Вы найдете цикл, посвященный

стратегическим играм. В этом цикле мы рассматриваем игры TOBRUK, LEGIONS OF DELATH, VULCAN, SAMURAI и др. как обзоры. Это примерно то, что мы хотели бы получать от ОБОЗРЕВАТЕЛЯ. Более широко рассмотренная программа, INVASION - это примерно то, что мы хотели бы получать от ЭКСПЕРТА.

7. Нежелательные контакты.

Если Вы располагаете описаниями, распространяемыми в среде любителей, и рассчитываете использовать их в своей работе, то будет лучше, если Вы воздержитесь от контактов с нами. Нас интересуют только Ваши авторские разработки и у нас очень мало времени на переписку.

8. Наши интересы.

"ИНФОРКОМ" никогда не занимался достаточно серьезной работой в области игрового программного обеспечения. В то же время, у нас есть многочисленные обращения читателей с просьбой о проведении обзоров новых игровых программ и глубокой проработки известных. Обращаясь к Вам с таким предложением, мы рассчитываем сделать ZX-РЕВЮ содержательнее и интереснее, а также подготовить определенный контингент соавторов для длительной работы. Может быть те из Вас, кто сумеют захватить целые направления, в дальнейшем получат постоянную рубрику.

Мы также рассчитываем, что сделанные по нашему заказу и оплаченные нами разработки не появятся нигде, кроме как у нас, причем это может быть не обязательно будет ZX-РЕВЮ. При достаточном материале это может быть особый сборник.

9. Персональные приглашения.

Объявление касается всех желающих, но мы хотели бы особо отметить, сотрудничество некоторых товарищей и пригласить их персонально:

Богачева В. В. , Брунова О. В. , Дейну А. А. , Жукова К. А. , Климова Д. В. , Котина Д. , Кочнева С. В. , Линькова И. П. , Сипухина Р. И. , Сорокина А. В. , Татевосяна С. , Фокина А. М. , Хахонова А. Н. , Шилина Д. П.

И особо пригласить в качестве ОБОЗРЕВАТЕЛЯ Деменина В. А.

ПЛЮС

НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС" ПРЕДЛАГАЕТ ВАШЕМУ ВНИМАНИЮ СВОИ УЧЕБНО-МЕТОДИЧЕСКИЕ РАЗРАБОТКИ

Методическая разработка (МР) N1.

Любители компьютерных игр найдут в ней несколько вариантов схем подключения джойстика. Наличие нескольких вариантов позволяет Вам выбрать наиболее приемлемый по наличию элементной базы.

Те, у кого есть принтер, смогут подключить его с помощью одной из схем последовательного или параллельного интерфейса принтера. Приведенные программы поддержки принтера (драйверы) позволят Вам распечатывать как текстовую, так и графическую информацию. Схемы интерфейсов и программы поддержки предназначены для работы с любыми принтерами, оборудованными одним из стандартных интерфейсов: CENTRONICS (ИРПР-М), ИРПР (IFSP), RS-232C (V.24).

Для занимающихся компьютерной графикой и живописью предназначена схема светового пера и распечатка работающего с ним графического редактора LP48K.

Схема программатора для ПЗУ с УФ стиранием позволит любителям технического творчества самостоятельно программировать такие микросхемы как 2716, 2764, 27128, 27256, 573 PФ2, 5, 4, 6, 8.

Объем МР1 - 34 страницы, стоимость - 30 рублей.

Со времени начала ее распространения (с сентября 1990г.) в адрес НТК не прекращается поток писем с положительными отзывами. Все схемы, входящие в разработку уже повторены многими пользователями, которые отмечают простоту изложения материала, доступность элементной базы и высокую повторяемость схем.

Методическая разработка (МР) N2.

Вторая методическая разработка МР2 состоит из двух частей. Первая часть посвящена контроллеру накопителя на гибких магнитных дисках НГМД. Контроллер полностью совместим с фирменной системой BETA-DISC INTERFACE. Он выполнен на базе микросхемы 1818ВГ93 (WD1793) и поддерживает работу до 4-х НГМД - 3-х или 5-ти дюймовых, односторонних и двусторонних, 40 и 80 дорожечных. Разработка содержит подробное описание схемы, сигналов, принципов работы интерфейса, рекомендации по сборке и наладке, систему команд дисковой операционной системы TRDOS.

О преимуществах работы с НГМД по сравнению с работой с магнитофонной лентой знают все, это новый уровень работы с компьютером, и мы надеемся, что наша разработка поможет Вам перейти на этот уровень.

Вторая часть МР2 посвящена стыковке компьютера с бытовыми телевизорами. Наиболее качественное цветное изображение получается при стыковке компьютера по R-G-B входу телевизора. При стыковке по антенному входу с кодировкой по системе СЕКАМ или PAL происходит некоторая потеря качества цветного изображения, но зато телевизор не требует переоборудования. В некоторых случаях может удовлетворить стыковка через видеовход с чернобелыми или цветными телевизорами без передачи информации о цвете, т.е. в черно-белом изображении.

Вторая часть МР2 содержит схемы всех перечисленных способов стыковки компьютера с телевизорами, причем схемы предназначены не только АЛЯ полупроводниковых, но и для ламповых телевизоров.

Стоимость МР2 - 30 руб.

Методическая разработка (МР) N3.

Третья методическая разработка (МР3) подготовлена по многочисленным письмам и пожеланиям пользователей "Спектрума". В нее входит принципиально новый способ русификации "Спектрума", выгодно отличающийся от известных способов.

Знакогенератор кириллицы размещается в неиспользуемой области ROM, там же

размещены некоторые подпрограммы, которые добавляют в операционную систему "Спектрума" два новых регистра клавиатуры R и N к уже имеющимся K, L, C, E, G. Таким образом, в любой строке могут быть представлены буквы как русского, так и латинского алфавита одновременно, т.е. например, операторы Бейсика на английском языке, а текст на русском.

Также приведены дополнения к программам поддержки интерфейсов принтеров, опубликованных в нашей МР1 для вывода на печать смешанного русско-латинского текста.

Кроме этого, в МР3 по просьбам пользователей включены схемы интерфейсов джойстиков, рассчитанных на двух игроков - INTERFACE 1 и SINCLAIR, а так же схема программируемого джойстика, позволяющая запрограммировать все положения рукоятки и кнопки в соответствии с клавишами клавиатуры и более простой вариант - механически программируемый (путем перестановки перемычек) джойстик.

Стоимость МР3 - 30 руб.

Все программы, входящие в разработки, приведены в виде листинга и блоков шестнадцатеричных кодов.

Свои заявки на приобретение МР1-МР3 Вы можете направлять по адресу:

127566, Москва, И-566, НТК "Плюс". Оплата производится, только наложенным платежом.

Просьба при всех обращениях к нам прикладывать к письму конверт с обратным адресом. В заявке необходимо указать свой полный адрес, фамилию, имя, отчество.

СКАЖИТЕ, ВЫ ЧУВСТВУЕТЕ, КАК С ПРИОБРЕТЕНИЕМ КОМПЬЮТЕРА ВАША ЖИЗНЬ
СТАЛА СПОКОЙНЕЕ И ВЫ ПОЧУВСТВОВАЛИ СЕБЯ УВЕРЕННЕЕ?

НЕТ?

ТОГДА ВАМ НУЖЕН МАЛЕНЬКИЙ АУТОТРЕНИНГ.

Внимательно прочитайте следующие строки.

* * *

Я чувствую себя прекрасно. Мой кругозор сильно расширился. У меня появились интересные друзья и знакомые. Мне не надо думать о завтрашнем дне, работа с компьютером решит все мои проблемы.

На работе ожидается сокращение? Это прекрасно! Тем, кто останется, повысят зарплату. Меня оно не коснется, все знают, что я один из тех, кто приобщился к таинственному компьютерному знанию.

Пусть беспокоится тот, кто забыл, когда последний раз освоил что-нибудь сам, без указки начальства.

Жизнь становится дороже день ото дня? Наверное так и должно быть, но я спокоен. Мои проекты, мои идеи тоже дорожают день ото дня.

Нет уверенности в завтрашнем дне? А зачем она мне? И завтрашний и послезавтрашний дни у меня давно спланированы. Я думаю о двадцать первом веке, а завтрашний день пусть сам подстраивается под мои планы.

Нет финансовой стабильности? Пусть об этом беспокоятся те, кто вкладывает средства, в сбербанк или в акции каких-нибудь предприятий. Я вкладываю их в знания, а их не обменяют как ставшие ненужными купюры, да и процент они дадут побольше.

Жизнь становится суетливее? Возможно, но когда я часами трясусь в переполненном автобусе, моя голова работает над новыми идеями и дает отдачу не хуже, чем дома, в мягком кресле.

Нечем себя занять, по телевизору нечего смотреть, с семьей некуда пойти? Мне бы Ваши проблемы. Мы с семьей можем по сорок восемь часов в день что-то осваивать, над чем-то работать. Дети, правда увлекаются игрушками, но через пол-года это пройдет, они поймут, что самая интересная игра - это игра в сыщиков. Они начнут пылливо разбираться с работой программ, и все секреты будут им открыты. Прекрасное будущее им гарантировано и без армии грабителей-репититоров.

* * *

НУ КАК, ПОМОГ ВАМ АУТОТРЕНИНГ?

Неужели нет? Тогда с Вами все ясно!

Вы просто не приобрели наш трехтомник для изучения программирования в машинных кодах и на языке АССЕМБЛЕРА. Очень напрасно. Тысячи читателей уже это сделали, а кому же хочется быть в последних рядах на спринте в 21-ый век?

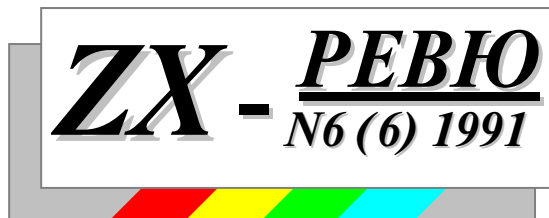
А ведь мы рассчитываем делать новые книги, в которых будем опираться на эти как на базовые.

Заплатив шестьдесят (пока) рублей Вы приобретаете не только доступный учебник, Вы приобретаете уверенность в завтрашнем дне. Это дорого? Да, недешево. Знания стоят денег. Только незнание ничего не стоит, но платить за него придется дороже.

Будут меняться поколения компьютеров. Сокращая по обходной дуге разрыв в программировании между нами и американцами, мы будем все быстрее и быстрее осваивать новые технологии. Сегодня СИНКЛЕР, завтра IBM, послезавтра что угодно. Но первый шаг ведь делать надо! И поверьте, это шаг именно по той тропинке, которая сократит в десятки раз сложный путь по освоению новых компьютерных систем будущего.

Пишите, адрес Вам известен.

С наилучшими пожеланиями,
ИНФОРКОМ.



НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ИНФОРКОМ". 107241, МОСКВА, Б-241, А/Я 37

Дорогие читатели.

Сейчас Вы держите в руках шестой выпуск ZX-PEBYU, а это означает, что половина пути нами пройдена, можно подводить первые итоги.

Судя по тем письмам, которые мы получаем, ZX-PEBYU находит добрый отклик среди любителей "Спектрума", и мы с большим интересом продолжаем это начинание.

Но не все так хорошо, как хотелось бы. Нас не устраивает содержание рекламного приложения, надо сказать правду - оно не пользуется популярностью, а в то же время читатели просят давать больше познавательных материалов.

Начиная с этого номера мы ликвидируем этот раздел, увеличив объем материалов, представляющих всеобщий интерес. Конечно, те немногие объявления, которые уже оплачены нашими читателями, мы еще дадим, но больше плату за объявления не принимаем.

Это не значит, что мы лишаем Вас связи друг с другом. Во-первых, кому есть что сказать, может это сделать и через раздел "ФОРУМ". Во-вторых, в порядке исключения мы все-таки будем бесплатно иногда давать некоторые объявления наших читателей, но при условии, что они будут иметь общественный интерес, при наличии свободного места и в порядке очереди. В первую очередь просьбы об экстренной помощи и объявления от инвалидов, поскольку они наиболее нуждаются в общении с единомышленниками.

Разумеется, в скромных пределах мы оставляем за собой возможность представлять свои разработки и разработки НТК "ПЛЮС" (это условие нашего сотрудничества).

Кстати, в этом выпуске нет технического раздела, но это явление временное, связанное с отпускным периодом, далее он будет восстановлен.

Нет в этом выпуске и раздела "Секреты ПЗУ". Большой объем очередной главы, посвященной процедурам, выполняющим печать на экране и принтере вынуждает нас опять сделать очередной выпуск двойным.

Все чаще поступают вопросы о подписке на 1992 год. Мы пока не готовы сообщить ничего конкретного. Ясно только одно - будет подписка не будет, а "ZX-PEBYU" все равно будет.

Те, кто поверил нам в этом году и подписался, будут получать его и в будущем, пусть не в форме подписки, а в какой-то другой, но мы знаем, что нам работа нужна. Вы ее ждете и те, кто нами зарегистрирован, всегда будут своевременно проинформированы.

По этой причине мы и не стали прекращать прием подписки в этом году. Напоминаем, что вновь подписавшиеся получают все вышедшие выпуски. Стоимость подписки за весь год - 90 руб.

Пока же мы формируем портфель материалов на 1992-ой год и, если он окажется очень толстый, рассмотрим вопрос о значительном увеличении нашего объема.

И, в заключение, просьба ко всем нашим читателям. Мы хотели бы выписать наиболее популярные местные ежедневные или еженедельные газеты нашей страны, чтобы лучше знать чем живут регионы. Не можете ли Вы дать нам совет, что наиболее охотно читают в Вашей местности?

Может быть Вы даже сможете выслать экземпляр такого издания для ознакомления, из него мы сразу узнаем и координаты редакции.

Надеемся на вашу подсказку.

С уважением, "ИНФОРКОМ".

Раздел для начинающих

Мы продолжаем публикацию заметок для начинающих нашего читателя из г. Дубна, т. Скитева.

Сегодня раздел посвящен графике пользователя, так называемой UDG-графике.

При работе с графикой пользователя обычно сталкиваются с определенными трудностями. Для графики отводится 21 символ от "A" до "U" при условии печати этих знаков в графическом режиме (курсор G).

Каждый знак состоит из 6 строчек по 8 символов в каждой. Рисуем матрицу 8x8, и в ней рисуем то изображение, которое хотели бы видеть. Заменяем закрашенные квадраты единицами, а незакрашенные - нулем. Затем выберем символ, который хотим заменить своим, например "A", и наберем следующую программу.

```
10 POKE USR "A",BIN 00111100
20 POKE USR "A"+1,BIN 01011010
30 POKE USR "A"+2,BIN 01111110
40 POKE USR "A"+3,BIN 11100111
50 POKE USR "A"+4,BIN 10100101
60 POKE USR "A"+5,BIN 10100101
70 POKE USR "A"+6,BIN 10100101
80 POKE USR "A"+7,BIN 00100100
```

Если после выполнения программы перейдем в графический режим (а это выполняется одновременным нажатием клавиш CAPS SHIFT и 9) и напечатаем литеру "A", то увидим изображение паука. Это изображение будет находиться в памяти компьютера до тех пор, пока не будет выключено питание или нажата кнопка "RESET".

Программа заняла восемь строчек, а заменила только одну литеру. Можно сделать эту программу по-другому, используя оператор "READ" и список "DATA".

Правда, предварительно необходимо провести следующую работу - перевести изображение из двоичного кода (а мы создавали его в двоичном коде 0,1) в десятичный код. Для этого можно, например, воспользоваться оператором "BIN". Дав прямую команду PRINT BIN 00111100, получим результат - 60.

Затем следующая строка PRINT BIN 01011010 дает 50 и т.д. Таким образом определяем значения всех строк и заносим их в список DATA. Теперь программа выглядит так:

```
10 FOR X=0 TO 7
20 READ Y
30 POKE USR "A"+X,Y
40 NEXT X
50 DATA 60,90,126,231,165,165,165,36
```

Программа делает то же самое, что и предыдущая.

Продолжение паука (падающий паук).

```
110 FOR X=0 TO 7
120 READ Y
130 POKE USR "┘"+X,Y
140 NEXT X
150 DATA 16,16,16,16,16,16,16,16
160 FOR L=0 TO 20
170 PRINT AT L,3: INK 0; "┘"
```

Здесь черточка под буквой "T" означает, что это вовсе не буква, "T", а тот знак, который закреплен за клавишей "T" в графическом режиме. Пожалуйста не ошибитесь при наборе. Аналогично будем поступать и далее.

```
180 PRINT AT L+1,3: INK 2;"┘"
190 NEXT L
```

Продолжение паука (убегающий паук).

```
210 PAUSE 10
220 FOR C=3 TO 30
230 PRINT AT 21,C; " "
240 PRINT AT 21,C+1: INK 2;"┘"
```


250 NEXT C

В разработке "Большие возможности Вашего Спектрума" есть вариант перевода компьютера на русский шрифт, создав русские символы в графическом режиме.

Работая с графикой пользователя можно создавать и более сложные рисунки, состоящие из нескольких символов.

Попробуйте ввести следующую программу.

```
10 FOR I=0 TO 47
20 READ N
30 POKE USR "A"+I,N
40 NEXT I
50 DATA 0,2,3,7,15,19,83,126
60 DATA 0,64,64,224,240,200,202,126
70 DATA 127,115,120,31,15,4,4,28
80 DATA 254,206,30,243,240,32,56,0
90 DATA 127,187,120,27,15,4,28,0
100 DATA 254,254,30,216,240,32,32,56
```

Выполним эту программу, результатом будет изменение в графическом режиме символов A,B,C,D,E,F. Если будет сообщение - OUT OF DATA, внимательно проверьте Ваш листинг. В каждой строке DATA должно быть 8 чисел. Теперь можно командой NEW очистить память, листинг исчезнет, но результат работы программы останется (до команды RESET).

































Введем следующую программу. Символы, стоящие в кавычках, будем вводить в графическом режиме.

Прыгающие фигурки.

```
110 BORDER 5; PAPER 5: CLS
120 PLOT 60,71: DRAW 135,0
130 PAUSE 50
140 PRINT AT 10, 14; "AB"; AT 10,18;" "
150 PRINT AT 11,14; "CD"; AT 11,18; "AB"
160 PRINT AT 12,14; " "; AT 12,18; "EF"
165 BEEP .05,0
170 PAUSE 50
180 PRINT AT 10,14;" "; AT 12,18;" "
190 PRINT AT 12,14; "EF"; AT 12,18;" "
205 BEEP .05,20
210 GO TO 130
```

По такому принципу можно создавать достаточно сложные рисунки состоящие из большого количества символов до 21.

Эта программа рисует шахматное поле. В строках 50, 60 после кавычек делаем два пробела, затем два черных квадрата в графическом режиме и так далее, всего в строке 32 символа. А в строках 70, 80 сначала два квадрата, затем два пробела и т.д.

```
10 CLS
20 BORDER 0
30 PAPER 7
40 FOR N=1 TO 6
50 PRINT " "         "
60 PRINT " "         "
70 PRINT "         "
80 PRINT "         "
90 NEXT N
```

Используя эту программу, а также программы, которые мы приводили ранее, сетка на экране, вертикальные цветные полосы, полное окрашивание экрана в основные цвета (R,G,B) и так далее, можно сделать тестовую программу для монитора. Для этого необходимо сделать программу меню, один из вариантов можно найти во втором выпуске "ZX-РЕВЮ" в разделе маленькие хитрости, а в листингах программ соответственно изменить номера.

MEGA BASIC

Сегодня мы заканчиваем печать инструкции по работе с языком программирования MEGA-BASIC и приводим окончание списка дополнительных команд, а также перечень системных сообщений программы.

Начиная со следующего выпуска мы последовательно начнем рассматривать версии другого, не менее мощного языка - BETA BASIC.

* * *

CURRENT_n - по этой команде текущим становится окно n.

DEFG_A\$,n... - эта команда позволяет определить символ UDG. Форма UDG задается 8 числами, записанными после a\$, которое в свою очередь определяет адрес ОЗУ, где этот символ будет храниться.

DELETE_a,b - эта команда используется для удаления блока программных строк. Здесь a - номер первой строки, а b - номер последней строки удаляемого блока. Если a>b, то на экране появится сообщение "B Integer out of range".

DOKE_a,b - эта команда эквивалентна следующим:

POKE a,b-256*INT(b/256)

POKE a+1,INT(b/256)

DOWN_y,x,a\$ - по этой команде строка символов a\$ выводится в нижнюю часть экрана.

EDIT_n - по этой команде n-я строка переписывается в область редактора строк и активизируется редактор,

ENDPROC_n - эта команда определяет конец процедуры, имя процедуры может быть записано после команды.

EXAMINE - по этой команде на экран выводится содержимое заголовка вводимой с кассеты программы, т.е. имя, длина в байтах и начальный адрес в ОЗУ вводимого файла.

FADE_n - по этой команде каждый раз уменьшается на 1 содержимое каждого байта атрибутов экрана. Этот процесс повторяется до тех пор, пока значение каждого байта не станет равным n.

FONT_n - эта команда позволяет выбрать необходимый символьный набор из имеющихся в памяти. Команда не действует в режиме MODE 1, так как в этом режиме используется собственный символьный набор.

FX_n,m - команды выбора окна на экране.

GET_0,a,y,x,d,w - эта команда используется для сохранения в памяти изображения экрана.

INVERT - по этой команде инвертируется весь экран; цвет INK меняется на цвет PAPER, а цвет PAPER на цвет INK.

KEY_n,A\$ - эта команда используется для задания функциональных клавиш. Здесь n - номер клавиши, а A\$ - строка символов, закрепленных за этой клавишей.

MODE_(n),a - эта команда используется для выбора размеров выводимых на экран символов. Если после команды записаны два числа, то первое определяет номер окна.

MON - по этой команде вызывается фронт-панель. При этом пользователь может не только узнать содержимое ячеек памяти и регистров процессора, но и изменить их. Возврат в Мега-Бейсик осуществляется нажатием SPACE.

MTASK_n - по этой команде программа может выполняться одновременно с двух точек.

PAN_n,m - эта команда позволяет перемещать попиксельно вправо и влево содержимое текущего окна экрана без изменения значений атрибутов.

PCLEAR - по этой команде очищается стек PROCEDURE/REPEAT-UNTIL.

PLAY_n,l,s,d,f - эта команда используется для исполнения звуковых эффектов.

POP_n - по этой команде в стек PROCEDURE/REPEAT-UNTIL записывается число n.

PRINTER_n - эта команда используется для вывода на принтер или другие периферийные устройства.

PUSH_n,m - эта команда используется для записи значения в стек PROCEDURE/REPEAT-UNTIL.

PUT_1,a,y,x,d,w - эта команда используется для восстановления изображения, хранящегося в памяти, на экран. Команда противоположна команде GET.

REPEAT - оператор, определяющий начало цикла REPEAT-UNTIL. Если стек PROCEDURE/REPEAT-UNTIL заполнен, то на экране появляется сообщение "PROC stack overflow"

RESTART_n - эта команда эквивалентна команде ON ERROR GO TO, которая существует в других версиях BASIC.

RESTART_OFF - использование этой команды приводит к отмене режима программной обработки ошибок.

SCROLL_n,m - по этой команде осуществляется перемещение вверх или вниз значений текущего окна.

SOFF - команда выключения ISG.

SON - команда включения ISG.

SPEED_n - команда управления скоростью выполнения программы при включенном режиме трейсирования TRON.

SPRINT_x,y,a,b,c\$ - это команда вывода на принтер строки символов, размер которых задается пользователем.

SFROFF_n,m - выключение спрайтов.

SPRON_n,m - включение спрайтов.

SPUT_x,y,a,b,c,w,d - эта команда подобна команде PUT, но в данном случае размеры изображения могут быть увеличены в двух направлениях заданием коэффициентов c,b.

SREP_n - это команда повторения звуков, информация о которых записана в звуковом буфере.

STIPPLE_n - эта команда позволяет создавать теневые эффекты для символов, записанных в режиме MODE 4.

SWAP_n,m - это команда для манипуляций с файлом атрибутов. При этом значение байта файла атрибутов меняется с n на m.

TRON - команда включения режима трассировки. При трассировке номер выполняемой программной строки выводится в нижнем левом углу экрана.

TROFF - команда выключения режима трассировки.

UNTIL_a - эта команда определяет конец цикла REPEAT-UNTIL.

VDU a,(a...) - эта команда эквивалентна PRINT CHR\$.

WINDOW_y,x,d,w - по этой команде задается размер и расположение на экране текущего окна.

Новые сообщения об ошибках

FX NOT IMPLEMENTED - ошибка, связанная с командой FX..., Вы попытались использовать несуществующее назначение.

ILLEGAL WINDOW - ошибка команд FX или CURRENT. Вы попытались использовать более чем 10 окон.

LINE NOT FOUND - ошибка команды EDIT. Вы попытались редактировать строку, которой нет в программе.

MISSING PARAMETERS - некорректная запись чисел или параметров в командах.

WINDOW TOO LARGE - эта запись может появиться при использовании команды WINDOW, если размеры окна слишком большие.

WINDOW TOO SMALL - эта запись появляется, если один из размеров окна равен нулю.

X TOO LARGE или Y TOO LARGE - выход за пределы экрана.

В нашей почте лежат письма от читателей, которые либо начали работу с МЕГАБЕЙСИКом, либо собираются ее начать, но хотели бы видеть образец программы, написанной на этом языке.

Самое простое, что можно им порекомендовать - это просмотреть листинг файла DEMO, который прилагается к программе. Но поскольку файл DEMO предназначен прежде

всего для того, чтобы наилучшим образом представить программу в работе, а не в просмотре, он выглядит не очень читабельным.

Идя навстречу высказанным пожеланиям, мы приводим распечатку программы, предназначенной для исследования оператора PLAY. Программа взята из журнала YOUR SPECTRUM (N15, июнь, 1985 г.).

Прелесть программы в том, что во-первых Вы наглядно увидите насколько же текст ее более понятен и компактен по сравнению с обычным Бейсиком, а во-вторых, сможете неплохо разобраться с командой PLAY, прощупать ее возможности.

После этой команды следуют несколько параметров и влияние их на работу команды на первый взгляд совсем неочевидно.

Наберите эту программу и запустите. Вертикальным перемещением курсора Вы сможете выбрать тот параметр, который хотите поменять, а горизонтальным перемещением - изменить этот параметр. После этого нажмите "пробел" и слушайте, что у Вас получилось.

```
1000 REM PLAY EXPLORER
1010 REM BY L A PRIVETT
1020 REM MARCH 1985
1030 :
1040 PCLEAR
1050 SETSCREEN
1060 SETDISPLAY
1070 SETVALUE
1080 BOXIN
1090 SCANKEYS
1100 STOP
1110 :
1120 :
1130 :
```

Вот собственно и вся программа - она занимает 7 строк, начиная со строки 1040 - по строку 1100.

Все остальное - вспомогательные процедуры.

```
2000 @SETSCREEN
2010 WINDOW_0,0,22,64
2020 PAPER 0
2030 INK 6
2040 BRIGHT 1
2050 OVER 0
2060 INVERSE 0
2070 MODE_2
2080 CLS
2090 DRAW 255,0
2100 DRAW 0,175
2110 DRAW 255,0
2120 DRAW 0,-175
2130 FONT_1
2140 LET a$="PLAY_ EXPLORER"
2150 SPRINT_20,10,1,2,A$
2160 INK 3
2170 ENDPROC_SETSCREEN
2180 :
2190 :
2200 :
3000 @SETDISPLAY
3010 LET A$="A B C D E"
3020 LET B$="= = = = ="
3030 INK 5
3040 DOWN_5,8,A$
3050 DOWN_5,12,B$
3060 FONT_2
```

```

3070 SPRINT_10,150,2,2,"PLAY"
3080 PRINT AT 2,40;"USE CURSOR"
3090 PRINT AT 3,40;" KEYS TO "
3100 PRINT AT 4,40;"SELECT AND"
3110 PRINT AT 5,40;" CHANGE "
3120 INK 6
3130 PLOT 0,36
3140 DRAW 255,0
3150 INK 6
3160 PRINT AT 8,40;"SPACE FOR"
3370 PRINT AT 9,40;" PLAY "
3180 BEEP .1,0
3200 ENDPROC_SETDISPLAY
3210 :
3220 :
3230 :
4000 @SETVALUE
4010 DIM M(5)
4020 DIM V(5)
4030 DIM P(5)
4040 FOR F=1 TO 5
4050 LET P(F)=3+F*2
4060 LET V(F)=0
4070 NEXT F
4080 RESTORE 4180
4090 FOR F=1 TO 5
4100 READ Q
4110 LET M(F)=Q
4120 PRINT AT 3+F*2,16;V(F)
4130 PRINT AT 3+F*2,24;M(F)
4140 NEXT F
4150 LET PRE=1
4160 LET POS=1
4180 DATA 1,255,255,255,255
4200 ENDPROC_SETVALUE
4210 :
4220 :
4230 :
5000 @BOXIN
5010 PLOT 20,60: DRAW 111,0
5020 DRAW 0,79: DRAW -111,0
5030 DRAW 0,-79
5040 PLOT 95,0: DRAW 0,36
5050 PLOT 95,19: DRAW 160,0
5060 MODE_1
5070 PRINT AT 18,26;"A B";
5080 PRINT " C D";
5090 PRINT " E"
5100 INK 4
5110 PRINT AT 15,25;"MAX"
5120 PRINT AT 15,14;"ACTUAL"
5130 MODE_2
5140 INK 2
5150 PRINT AT 12,40;"Q TO QUIT"
5160 INK 6
5170 PRINT AT 15,40;"L.PRIVETT"
5200 ENDPROC_BOXIN
5210 :
5220 :

6000 @SCANKEYS
6010 LET Z$ = " "
6020 OVER 1
6030 PAPER 1
6040 INVERSE 1

```

```
6050 PRINT AT P(1),6;Z$
6060 REPEAT
6070 LET FINISH=0
6080 IF IN 63486=239 THEN LEFT
6090 IF IN 61438 = 251 THEN RIGHT
6100 IF IN 61438=247 THEN UP
6110 IF IN 61438=239 THEN DON
6120 IF IN 64510=254 THEN QUIT
6130 IF IN 32766=254 THEN NOISE
6150 BEEP .005,-16
6160 UNTIL_FINISH
6200 ENDPROC_SCANKEYS
6210 :
6220 :
```

```
7000 @DON
7010 LET POS=POS+1
7020 IF POS>5 THEN LET POS=5
7030 POSITION
7040 ENDPROC_DON
7070 :
7080 :
7100 @UP
7110 LET POS=POS-1
7120 IF POS<1 THEN LET POS=1
7130 POSITION
7140 ENDPROC_UP
7170 :
7180 :
```

```
7200 @POSITION
7210 PRINT AT P(PRE),6; Z$
7220 PRINT AT P(POS),6; Z$
7230 LET PRE=POS
7240 ENDPROC_POSITION
7250 :
7260 :
```

```
7300 @LEFT
7310 LET S=POS
7320 LET V(S)=V(S)-1
7330 IF V(S)<0 THEN LET V(S)=0
7340 OVER 0
7350 PRINT AT P(S),16;V(S); " "
7360 OVER 1
7370 ENDFROC_LEFT
7380 :
7390 :
```

```
7400 @RIGHT
7410 LET S=POS
7420 LET K=M(POS)
7430 LET V(S)=V(S)+1
7440 IF V(S)>K THEN LET V(S)=K
7450 OVER 0
7460 PRINT AT P(S),16; V(S); " "
7470 OVER 1
7480 ENDPROC_RIGHT
7490 :
7500 :
```

```
8000 @QUIT
8010 LET FINISH=200
8020 PAPER 0
8030 INK 7
```

```
8040 OVER 0
8050 CLS
8060 ENDPROC_QUIT
8070 :
8080 :

8100 @NOISE
8110 LET A=V(1): LET B=V(2)
8120 LET C = V(3): LET D=V(4)
8130 LET E=V(5): MODE_1
8140 OVER 0
8150 INK 7
8160 PAPER 0
8170 PRINT AT 20,26;A
8180 PRINT AT 20,32;B
8190 PRINT AT 20,38;C; " "
8200 PRINT AT 20,46;D; " "
8210 PRINT AT 20,54;E; " ";
8220 MODE_2
8250 PAPER 1
8260 OVER 1
8270 INK 6
8280 PLAY_A,B,C,D,E
8310 ENDPROC_NOISE
```

128 К

Итак, сегодня мы продолжаем разговор о некоторых особенностях компьютеров с памятью 128К. Мы уже рассмотрели вопросы, связанные с особенностями организации оперативной памяти и знаем, как подключаются дополнительные страницы при программировании в машинных кодах.

Здесь мы рассмотрим работу системы организации памяти в виде так называемого "ЭЛЕКТРОННОГО ДИСКА" (RAM-диска).

Итак, в Бейсике Вы имеете возможность выгружать программы, данные или машинный код на электронный диск примерно таким же образом, как это делается при выгрузке на ленту. Разница, конечно, в скорости этой операции, да и команда немного отличается. При выгрузке на квазидиск - это SAVE!, а при выгрузке на ленту, как Вы знаете, просто SAVE.

Аналогичным образом работают и команды LOAD!, VERIFY!, MERGE!.

Очевидным недостатком электронного диска является то, что все, что на нем хранится, стирается из памяти при выключении питания.

В нашей статье мы рассмотрим организацию системы "электронного диска" с точки зрения логики машинного кода.

Необходимо отметить, что термины "электронный диск", "квазидиск", "RAM-диск" - конечно же очень неточные, поскольку никаких дисков в компьютере нет. Есть 128 килобайт оперативной памяти, а называется система ее использования "электронным диском" только потому, что она имитирует работу как бы с внешним дисководом.

Страница ОЗУ.

В предыдущих выпусках мы с Вами уже говорили о страничной организации памяти и говорили о страницах ОЗУ и страницах ПЗУ. Сейчас мы коснемся страниц ОЗУ несколько глубже. Как Вы помните, в компьютере есть 8 страниц ОЗУ, пронумерованных от 0 до 7, объемом по 16К каждая, что и дает в сумме 128К.

Сейчас же мы добавим к этому, что эти страницы ОЗУ разделены на две группы, которые назовем:

ОСНОВНАЯ ПАМЯТЬ и ПАМЯТЬ ЭЛЕКТРОННОГО ДИСКА.

На Рис. 1 показано распределение страниц ОЗУ по этим группам.



Рис.1

Как видно из рис. 1, основная память содержит три страницы ОЗУ по 16К каждая, что в сумме составляет 48К. Здесь находится память экрана, системные переменные, рабочая БЕЙСИК-программа со своей областью переменных, область информации о каналах, стек калькулятора, графика пользователя и все прочее, чему положено быть. Одним словом, как видите, это раздел 128-го мало отличается от карты памяти стандартной 48-килобайтной

машины. Все нововведения коснулись другого раздела ОЗУ - области памяти электронного диска.

Основная память ОЗУ состоит из страниц 5,2,0 (именно в этом порядке). Страница 5 постоянно впечатана, начиная с адреса 4000H, страница 2 - постоянно, начиная с адреса 8000, а страница 0 - при нормальных обстоятельствах, начиная с адреса 0C000H. Это означает, что для того, чтобы иметь доступ к одной из этих страниц, Вы должны впечатать ее вместо нулевой страницы, начиная с адреса 0C000H (а ее сохранить и впоследствии восстановить). Итак, нельзя иметь одновременный доступ ко всей памяти электронного диска.

Чтобы упростить дальнейшие рассуждения, введем понятие код страницы ОЗУ. Это не то же самое, что номер страницы. На рис. 1 показано, как кодируются страницы основной памяти и памяти электронного квазидиска. Итак, код страницы - число от 0 до 5.

В прошлых выпусках мы говорили о том, что для обозначения памяти выше 64 килобайт четырех шестнадцатеричных разрядов уже недостаточно и ввели пятый разряд - так, что адрес 1C000H обозначает первую страницу ОЗУ и, соответственно, адрес C000H на этой странице, соответственно аналогично можно поступить и с кодами страниц. Рассмотрим начало области памяти электронного диска.

Оно существует на странице ОЗУ с номером 1. Она же - страница с кодом 0. Тогда можно указать на адрес, принадлежащий этой странице, указав на код страницы [0] и на адрес, например C000.

Чтобы избежать путаницы кодов страниц с номерами страниц, мы будем писать код в квадратных скобках.

Итак, начало области электронного диска имеет абсолютный адрес 1C000H, а в терминологии кодов страниц - адрес [0]C000.

Чтобы это было понятнее, вот несколько примеров:

1FFFF	=	[0]FFFF
3C001	=	[1]C001
4D800	=	[2]D800
6EE00	=	[3]EE00
7EBEC	=	[4]EBEC
4000	=	[5]4000
ABCD	=	[5]ABCD
F000	=	[5]F000

Понятно, что одна пара регистров микропроцессора не в состоянии хранить число, большее по размеру, чем четыре шестнадцатеричных разряда, поэтому, чтобы работать с такими адресами в машинном коде, необходимо наличие трех регистров, вместо обычных двух. Причем неважно, в какой системе Вы работаете - с адресацией через номера страниц или через их коды. Обычно, работая в системе электронного диска, удобнее пользоваться кодовой адресацией, а при работе со страничной организацией памяти (как в прошлых выпусках) - удобнее пользоваться абсолютной адресацией по номеру страницы.

В качестве примера рассмотрим такой случай. Нам надо сохранить адрес [4]EBEC в тройном регистре ANL. Тогда регистр A должен содержать 04, а пара HL - число EBEC.

Для чего вообще нужен был переход от номеров страниц к их кодам? Для упрощения операций адресации в машинных кодах. Номера страниц связаны с физической организацией памяти, и потому идут, как видно из рис. 1 не подряд. Коды же страниц - чисто логическая конструкция, и потому идут подряд и манипулировать с ними проще.

Например:

Пусть ANL содержит адрес какого-либо байта в памяти электронного диска. Как отыскать адрес следующего за ним байта (Рис.2):

INC HL	- Увеличили адрес на единицу
BIT 7,H	- Проверили старший разряд на 0. Поскольку в нормальных условиях там всегда единица, то появление нуля будет свидетельствовать о том, что страница исчерпана.
RET NZ	- Если не 0, то все в порядке и возврат.
LD HL,C000	- В противном случае переходим на новую страницу и выставляем ее начальный физический адрес C000.
INC A	- Увеличиваем на единицу номер страницы.
RET	- Возврат.

Рис.2

А теперь представьте, во что бы превратилась эта нехитрая процедура, если бы мы использовали не коды страниц, а их физические номера. Ведь нельзя же было бы использовать простую и быструю операцию INC A.

На этом мы прервемся до следующего выпуска, в котором подробно рассмотрим карту памяти RAM-диска и разберем несколько примеров по работе с RAM-диском из машинного кода.

FORUM

Мы вновь возвращаемся к вопросу о пределах совместимости отечественных моделей "Спектрума" и будем возвращаться к нему еще не раз, поскольку считаем его одним из наиболее важных.

На наш взгляд существенный вклад в прояснение ситуации внес т. Симаков из г. Красноярск. Его письмо мы приводим почти полностью и надеемся, что оно заинтересует наших читателей.

Сам он работает с компьютером, собранным по схеме "Балтика" и практически не имеет проблем. Нормально работают и "ELITE" и "TLW2" о которых мы упоминали ранее, но в его компьютере стоит ПЗУ 1982 г., а версии 1987 г. и 1989 г. работают менее надежно.

Знакомые товарища Симакова, владеющие моделью "Дубна" неоднократно обращались к нему с просьбой разобраться с причинами незагружаемости ряда фирменных программ. Было установлено, что от этого страдают программы, имеющие нестандартный загрузчик, т.е. не использующие для загрузки процедуры, размещенные в ПЗУ, а какие - то свои из предварительно загруженного блока (кстати он-то загружается нормально, поскольку его загрузкой управляет ПЗУ).

Сами эти загрузчики похожи на стандартный, но имеют какие-либо особенности, например для вывода счетчика на экран. При этом в них установлены определенные константы, задающие временные интервалы между считыванием отдельных битов с магнитофонного порта (см. РЕВЮ, с. 78-79).

Но все дело в том, что значения этих констант рассчитаны исходя из заданной скорости работы микропроцессора, а в "Дубне" (по крайней мере в попавшихся ему экземплярах) был использован процессор UA880 вместо Z80, имеющий меньшую тактовую частоту.

Разработчики компьютера по-видимому это учли и скорректировали эти константы в ПЗУ компьютера, но загрузка под управлением внешнего загрузчика не идет.

Вскрыв программу и заменив нестандартный загрузчик на обращение к ПЗУ (0556H), ему удалось "реанимировать" несколько программ, например "Exolon", "Freddy Hardest" и ряд других.

По этой же причине по-видимому не работают и TF-COPY и C0FY-86/M, поскольку они тоже применяют свой собственный загрузчик.

* * *

Мы благодарим т. Симакова за информацию и ждем других сообщений по этим вопросам.

МАЛЕНЬКИЕ ХИТРОСТИ

Так сложилось, что экономия памяти стала основной темой этого раздела. По-видимому, это как раз тот вопрос, в котором без хитростей не обойтись. Очень обидно бывает, когда к концу Вашей программы Вы обнаруживаете, что места для чего-то не хватает.

Вторая причина такой направленности нашего раздела в том, что "хитрости" - это нестандартные, а потому нередко не очень удобные приемы и, разумеется, их не применяют без достаточных на то оснований. Острая необходимость сэкономить память - как раз достойное основание.

Но как бы там ни было, будете Вы применять эти приемы или нет - дело Ваше, но узнать побольше о своем компьютере Вы наверное сможете, а сейчас для нас это самое главное.

Какие же программы в первую очередь нуждаются в "хитростях"? В первую очередь текстовые, а именно:

- адвентюрные игры;
- обучающие программы;
- деловые игры.

То есть, это программы с большим количеством текстовых блоков.

Рассмотрим пример:

```
10 PRINT "Нажмите любую клавишу".
```

Занимает такая строка 29 байтов в программной области Бейсика (PROG) и ничего не занимает в области Бейсик-переменных (VARS), но это только пока. После запуска программы всякий раз, когда Вы будете выполнять эту строку, в области VARS под нее будет расходоваться память.

Если Вы предполагаете использовать ее не один раз, то можете поступить так:

Сбросьте программу - NEW, а затем наберите

```
LET a$="нажмите любую клавишу".
```

Если теперь Вы дадите команду LIST, то не получите ничего и это не удивительно, ведь у Вас нет никакой программы.

Теперь наберите:

```
10 PRINT a$
```

а затем дайте прямую команду GO TO 10 и Вы увидите, как Ваше сообщение появится на экране. Волшебство? Нисколько! Просто Ваше сообщение находится в области переменных и хранится там в переменной a\$. В области VARS оно занимает 24 байта, а в области PROG всего 8, т.е. в сумме 32 - Вы сэкономили 21 байт.

Если бы вместо GO TO 10 Вы использовали RUN, то получили бы сообщение "VARIABLE NOT FOUND", потому как RUN очищает область программных переменных. Работая таким образом, Вы обязаны использовать GO TO и более того, если в программе есть переменные, содержащиеся только в области VARS, то выгружать ее можно только с номером строки автостарта, чтобы она сама стартовала без необходимости давать RUN.

Кстати, такой прием позволяет немного сбить с толку малоопытного, но любопытного "расколыщика". Ведь по команде LIST, которую он освоил в первый день после приобретения компьютера, он увидит далеко не всю информацию.

"Расколыщика" поопытнее можно перехитрить например другим приемом.

Скажите, пожалуйста, вооружившись всеми своими знаниями, что бы Вы подумали, увидев в первых строках программы запись типа:

```
10 LET start=CODE "A"  
20 GO TO start
```

Наверное, Вы открыли бы справочник (если Вы его у нас приобрели) или хотя бы страницу 81 прошлого выпуска "ZX-РЕВЮ" и узнали бы, что код буквы А равен 65. То есть эта запись как бы эквивалентна GO TO 65. Далее Вы просмотрели бы строку 65, где настраивается вся работа программы и пошли бы по ложному пути.

Хитрость может быть например в том, что программист в 10-й строке написал вовсе

не букву А, не верьте своим глазам. Это вполне может быть графический символ из графики пользователя UDG, закрепленный за клавишей А, который пока ничей от нее по образу не отличается, но имеет совсем другой код - 144, и тогда запись эта эквивалентна GO TO 144. А может быть это какой-то другой код, если он предварительно "перевесил" образ буквы "А" на клавишу "F", тогда это будет код 149.

При всей примитивности этого примера, он дает Вам представление о мотиве, которым нередко пользуются профессионалы и, когда он подмешан в густую кашу других элементарных в общем-то приемов, то сильно портит кровь любопытным, особенно если они не знают с чего начать.

А теперь рассмотрим некоторые другие приемы экономии памяти.

В своих текстовых строках Вы можете использовать управляющие символы, это такие:

CHR\$ 5 - "запятая PRINT"

CHR\$ 16...CHR\$ 21 - цветовые атрибуты

CHR\$ 8 - BACKSPACE (смещение курсора влево на одну позицию).

Пожалуй о "запятой PRINT" надо поговорить особо. Дело в том, что в наборе символов "Спектрума" существуют две запятые. Они выглядят (на экране) совершенно одинаково, но имеют разное назначение.

Во-первых, это обычная запятая, символ номер 44 (CHR\$44), которая не является управляющим кодом, а просто обычным символом.

Во-вторых - это запятая оператора PRINT. Это уже управляющий код. Номер этого символа - 6 (CHR\$6). Вот пример ее применения:

```
10 LET a = 5
20 LET b = 10
30 PRINT a,b
```

Вы, конечно, знаете, что делает этот управляющий код, а если не знаете, то попробуйте и увидите. Попробуйте также

```
30 PRINT a;b
```

И еще попробуйте

```
30 PRINT a'b
```

Итак, этот код дает компьютеру команду - выполнить печать на правом поле экрана, а если Вы и так на нем, то перейти на новую строку.

Кстати, напомним, что для того, чтобы просто перейти на следующую строку можно использовать символ " '" (апостроф).

Итак, предположим, что Вам надо выдать на экран какое-то сообщение длиной более 32 символов. Вам надо строку делить, а делить как попало Вы не хотите - Вам надо это сделать в строго определенном месте.

Вариант первый - строку подбивают дополнительными пробелами (расход памяти).

Вариант второй - использовать апостроф (' ') - расход 3-х байтов.

Это можно сделать, если вставить CHR\$6 внутрь строки, что в общем-то сделать непросто и дело вот в чем.

Предположим, Вам надо набрать:

```
PRINT "Это первая печатная строка а это вторая".
```

Если Вы поставите запятую перед "а", то это будет обычная запятая (CHR\$44).

Если Вы хотите вставить туда CHR\$6, то пришлось бы закрывать кавычки:

```
PRINT "Это первая печатная строка", "а это вторая".
```

А здесь есть расход трех байтов.

Можно, однако, этого избежать следующим путем:

- наберите

```
PRINT "это первая печатная строка
```

- перейдите в режим E (CAPS SHIFT + SYMB SHIFT)

- держа клавишу CAPS SHIFT, нажмите "6".

Курсор приобретет желтый цвет. Это произошло потому, что Вы своими действиями ввели два управляющих кода (CHR\$16 + CHR\$6): CHR\$16, как известно управляет цветом INK, а CHR\$6 после него означает "желтый".

Нам с Вами символ CHR\$6 нужен, а вот CHR\$16 - нет, его надо удалить. Удалим его с помощью DELETE (CAPS SHIFT + "0") и курсор сразу "прыгнет" на другое поле. Это значит,

что код CHR\$6 воспринят. А теперь закончим наше сообщение:

а это вторая"

Пока не нажимайте ENTER.

Давайте убедимся, что зазор между "строка" и "а" реально существует. Прогоните курсор влево к началу строки (CAPS SHIFT + "5") и смотрите на него. Видите, как он перепрыгнул через зазор? Вы видите этот зазор? Конечно видите, а вот для компьютера его как бы и не существует, поэтому и нет расхода памяти на заполнение пробелами этого зазора.

Теперь нажмите ENTER и давайте расстанемся до следующего выпуска, который будет большим и даст возможность рассмотреть ещё несколько приемов, связанных с внедрением в текстовую строку управляющих символов.

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД

(Продолжение. Начало на стр. 49, 70)

Разработка программы.

Мой подход к разработке программ базируется на той профессиональной подготовке, которую я получил на фирмах ICL и IBM.

Сам процесс разработки не зависит от того, какой язык программирования я применяю. Конкретное же содержание каждого этапа может изменяться в зависимости от того, какими инструментальными средствами я располагаю на своей машине и, в определенной степени, от типа разрабатываемой программы.

Этап 1. Постановка задачи.

Для чего предназначена программа, что она будет делать? Для игровых программ есть несколько важных моментов, которые должны быть определены уже на ранней стадии:

- каким будет экранный интерфейс (какой метод будет применяться для связи между программой и пользователем);
- тема игры;
- каков предполагаемый размер игры и сколько памяти можно отвести под хранение экранов и таблиц данных.

Выходная информация:

Это прежде всего экраны, но сюда можно отнести и распечатки на принтере, если они предполагаются, здесь же должны быть намечены те выгрузки на ленту или диск, которые могут потребоваться по ходу игры. И, конечно же, должен использоваться звук. Любая программа, разумно его использующая, только выигрывает от этого.

Я говорю о выходной информации, не сказав о входной вовсе не потому, что считаю ее более важной. Просто Вы не определитесь с тем, что надо дать программе, если не знаете, что Вы будете от нее брать.

Для игровых программ стоит рассмотреть в качестве выходных характеристик такие очевидные параметры, как:

- степень использования цвета;
- предполагаемая скорость;
- притягательность и т.п. Составьте список того, что Вы хотели бы иметь в своей программе и, по мере ее разработки, периодически к нему возвращайтесь для сверки.

Входная информация:

Сюда относятся те данные, которые поступают в программу во время игры от клавиатуры или от джойстика. Если необходимо во время игры делать подзагрузку, то сюда же можно отнести данные, считываемые с ленты или диска.

От того, как происходит управление игрой в большинстве случаев зависит судьба этой игры - успех или неудача.

Применение джойстика всегда вносит в игру ограничение в связи с тем, что у него только одна кнопка. Ах, если бы у него было две функциональных кнопки! Тогда первая исполняла бы какое-то действие, а вторая - переключала бы вид этого действия. Например, первая означала бы "огонь", а вторая переключала бы оружие: нож, пистолет, автомат, гранатомет и т.п.

Однако кнопка у нас только одна и все функции, которые Вы хотели бы иметь, надо все-таки как-то к джойстику привязать. Здесь очень легко можно зайти так далеко, что игра окажется трудной в освоении.

Тщательно продумывайте технологию управления игрой. Не хватайтесь за первый попавшийся вариант. Взвесьте несколько альтернативных, выберите лучший и будьте готовы в любой момент от него отказаться, если окажется, что для других людей он слишком сложен.

Файлы и таблицы данных:

На этой стадии достаточно только сделать грубую прикидку того, какие данные могут потребоваться программе во время работы. Оцените их размер, предварительно наметьте метод, которым данные будут выбираться из таблиц.

Например, если в игре есть много комнат, надо наметить, как эти комнаты будут нумероваться и как будут кодироваться двери, связывающие разные комнаты между собой.

Первый этап должен закончиться твердым ответом на вопрос: возможно ли создание той программы, которую я задумал или все надо начать сначала. Обычно к этому моменту мой стол уже завален грудой набросков, эскизов, таблиц, которые смогут пригодиться на следующем этапе.

Этап 2. Проектирование программы.

Теперь настало время решать как Вы будете программировать все то, что задумали и наметили на первом этапе.

Работа начинается с подчистки и увязки результатов первого этапа. Я предпочитаю сначала строго расписать все таблицы данных, которые были намечены и заканчиваю эту работу присвоением имен переменным, которые будут хранить эти данные. Обратите внимание на то, что значительная часть пути уже пройдена, а компьютер Вам до сих пор еще не был нужен. Именно в этом и состоит суть моего подхода. Тщательная проработка структур данных на бумаге до начала программирования позволяет разбить предполагаемую программу на несколько независимых частей, с каждой из которых работать проще, чем со всей программой в целом. Каждая часть, в свою очередь делится на процедуры. Каждая процедура выполняет отдельную конкретную задачу, достаточно простую, чтобы можно было запрограммировать ее за один прием, без необходимости делить на части.

Правильное проектирование программы - самый критический фактор для успеха или провала всей Вашей работы. Плохо спроектированные или вообще неспроектированные программы занимают столько времени на программирование и отладку (если вообще отлаживаются до конца), что когда они готовы, то обычно быстрое действие их работы бывает неудовлетворительным. Вам очень дорого будет стоить пренебрежение этим этапом. Начинающему не терпится начать программирование, он полагает, что методом проб и ошибок, внося коррективы в программу по ходу разработки, решит все проблемы. Опытный программист, наоборот, отдает половину своего времени именно этой работе без компьютера, потому что знает, что когда коррекций в программе становится слишком много, то они начинают влиять друг на друга и быстро губят всю работу. Более того, еще на докомпьютерном этапе он уже наметит, что будет подлежать последующим уточнениям и коррекциям с целью наилучшей играбельности программы и так организует программу, чтобы эти уточняемые параметры не влияли друг на друга.

Итак, одним словом, для всех языков программирования желательно, а при программировании в машинном коде (или на Ассемблере) абсолютно необходимо расчленив задачу на множество микрозадач, каждую из которых можно запрограммировать так, чтобы текст этой микропрограммы мог сформироваться и уместиться в Вашей голове.

Этап 3. Кодирование.

На этом этапе записывается процедура в машинных кодах (на ассемблере), но сначала надо завести рукописный справочник по всем переменным, используемым в Вашей программе.

Если Вы полагаете, что и так все запомните, то лучше не экспериментируйте, для этого надо иметь необычную память. По каждой переменной я записываю имя, размер, содержание, примечание.

Например:

Имя	Размер	Содержание	Примечание
BOMB	1 байт	0 – нет 1 - есть	Наличие бомбы устанавливается в процедуре GETBOMB. Изменяется в процедуре USEBOMB.

Кроме справочника по переменным, надо вести еще и справочник по процедурам:

- имя процедуры;
- что делает;
- содержание регистров на входе;
- содержание регистров на выходе;
- какие переменные использует.

Например:

Имя	Что делает	Вход	Выход
FIRE	Инициализирует пакет процедур, управляющих стрельбой.	A - вид оружия 0...3 0 - нож 1 - пистолет 2 - автомат 3 - гранатомет BC,DE,HL – не определены F - флаг C=0 – выполнить стрельбу - флаг C=1 – поменять оружие - прочие флаги не определены	A – сила выстрела от 0 до 255
Используемые переменные:		WEAPON 1 байт CHARGE 3 байта	Вид оружия Количество зарядов на каждый вид оружия

Конечно, если Вы пишете программу с помощью Ассемблера, то эту информацию можно включать в исходный текст в виде комментария, но во-первых, лучше иметь эти листки на каждую процедуру под рукой, во-вторых, Вы быстрее приобретете профессиональный стиль и, в-третьих, это серьезный шаг вперед на пути создания собственной библиотеки процедур, которая позволит Вам в несколько раз сократить время на разработку будущих проектов.

Не надо пытаться написать сразу всю программу. Начните с программирования самых малых, не взаимодействующих с другими процедур. Их Вы быстро напишете и испытаете.

Этап 4. Отладка.

Я предпочитаю всякую процедуру проверять сразу же после того, как написал. Как правило, для этого необходимо приписывать несколько строк, чтобы обеспечить те данные, которые эта процедура должна получать из главной программы. Иногда даже необходимо сразу писать несколько процедур, испытывать и отлаживать совместно, если они друг без друга не отлаживаются, но чем их меньше, тем лучше. Есть закон, согласно которому если размер текста возрастает вдвое, то количество ошибок в нем возрастет в четыре раза. Есть и другой закон - если количество ошибок возрастает вдвое, то время на их отыскание и устранение тоже возрастает в 4 раза. Так что сами думайте, какой размер должны иметь Ваши элементарные процедуры. Заканчивается проверка и отладка процедуры тогда, когда Вы уверены, что в проведенном независимом испытании она работает и делает все, что положено. Правда, к сожалению, это еще не дает 100%-ной уверенности в том, что она также надежно будет работать в комплексе с другими при запуске всей программы в целом. Но если этап проектирования был хорошо проработан, то Вы сможете подкорректировать ее работу в составе всей программы без нарушения работы других блоков.

В основном эти проблемы совместной работы отлаженных порознь процедур порождаются взаимообменом данными между процедурами, который происходит через переменные, содержащиеся в заданных ячейках памяти, через регистры процессора или через флаги процессора. Эти ошибки будут минимальными, если Вы в свое время четко расписали для каждой процедуры, что она имеет на входе, что выдает на выходе и с какими регистрами работает.

Чтобы минимизировать вред от ошибок, связанных с взаимодействием между процедурами, я обычно "прикрепляю" новую к уже отлаженной. Так, постепенно, отлаженные процедуры создают среду для проверки и отладки новых.

Что такое хорошо и что такое плохо?

Хорошее программирование означает понятное программирование и минимум ошибок. Помните, что Вам многократно предстоит возвращаться назад и что-то переделывать, поэтому пожалуйста не оставляйте сами себе ловушек. Вот примерный список приемов плохого стиля:

1. Использование "хитрых", головоломных приемов.

Я считаю, что в принципе этого делать не надо. Исключение - тот случай, когда это абсолютно необходимо для достижения максимальной скорости работы, но и здесь это может быть в минимальном количестве особо важных (по быстродействию) процедур.

2. Злоупотребление применением стека.

Лучше оставить стек в покое. Особенно плохо применять его для хранения программных переменных. Вызов переменной по имени (в Ассемблере) - намного понятнее, быстрее и создает меньше условий для появления ошибок.

3. Неправильное использование процедур.

Каждая процедура должна иметь один вход и один выход. Если Вам их нужно больше, то вернитесь на этап 2 и повторите проектирование.

Никогда не имитируйте вызов процедуры (CALL) путем перехода (JP) с последующей манипуляцией стеком. Прием слишком головоломный, чтобы пользоваться им часто. Всегда завершайте процедуру естественным путем (команда RET в конце процедуры).

4. Злоупотребление переходами (JP).

Команда JP, как и GO TO в Бейсике очень затрудняет читаемость программы. При широком применении переходов JP Вы затрудняете себе отладку. Используйте JP только для организации циклов и для исполнения условного ветвления (IF... THEN...).

5. Работа с адресами.

Программируя на Ассемблере, не задавайте адреса, а определяйте только имена переменных, процедур и т.п. Если Вы будете вызывать процедуру по адресу, то все будет хорошо, пока Вам не придется ее переместить в другое место и тогда вся программа рассыплется как картонный домик.

К чертам хорошего стиля я отношу:

1. Определение имен со смыслом (название отражает содержание).

2. Комментарий к каждой процедуре.

3. Комментария к каждому "головоломному" (нестандартному) приему.

4. Применение регистров процессора по назначению:

A - аккумулятор;

B - восьмиразрядный обратный счетчик;

BC - шестнадцатиразрядный обратный счетчик;

HL - для хранения адреса или как двухбайтный аккумулятор.

DE - для хранения адреса в операциях, в которых необходимо использование двух адресов.

IX, IY - для выборки элемента из таблицы с использованием индексной адресации.

ПРИМЕЧАНИЕ ИНФОРКОМА.

Если Ваша программа предполагает обращение к системному ПЗУ "Спектрума", то лучше в регистр IY ничего не засылать, а использовать его для выбора данных из таблицы системных переменных "Спектрума", - прием, ставший стандартным.

Регистр же IX в этом случае наиболее часто используют в операциях загрузки/выгрузки.

В качестве примера распечатки программы в машинных кодах, обещанной в прошлом выпуске "ZX-РЕВЮ", мы приводим процедуру Стива Тернера для печати текстовых сообщений в любой точке экрана с точностью до пиксела (а не только в координатах знакомест).

Она может применяться в играх, например для выдачи на экран текущего счета.

Данный пример распечатает на экране две строки: EXAMPLE OF TEXT TABLE. Первая

строка содержится в относительном адресе 00AB, а вторая - в 00B3. Обратите внимание на то, что перед строкой, выводимой на печать, должен стоять байт, в котором указана длина этой строки:

00AA - 07 (7 символов в слове "EXAMPLE")

00B2 - 0D (13 символов в строке "OF TEXT TABLE").

(Продолжение в следующем выпуске)

0000		00010	; Пример программы
0000		00020	; печати текста
0000	3E01	00030	LD A, 1 ; Номер сообщения
0002	160A	00040	LD D, 10 ; Координата "у" места печати
0004	1E0A	00050	LD E, 10 ; Координата "х"
0006	CD0A00	00060	CALL TEXT
0009	C9	00070	RET
000A		00080	; Подпрограмма TEXT
000A		00090	; A – номер сообщения
000A		00095	; DE – координаты печати.
000A		00100	TEXT
000A	21AA00	00110	LD HL, TXADD;
000D	0600	00120	LOOK LD B, 0
000F	1803	00130	JR GOTRY
0011	4E	00140	LONG LD C, (HL) ; поиск текста
0012	23	00150	INC HL
0013	09	00160	ADD HL, BC
0014	3D	00170	GOTRY DEC A
0015	20FA	00180	JR NZ, LONG
0017	CD1B00	00190	CALL PRINT
001A	C9	00200	RET
001B		00210	
001B		00220	
001B		00230	; Подпрограмма PRINT
001B	22A800	00240	PRINT LD (INPUT), HL
001E	7A	00250	LD A, D
001F	E638	00260	AND 38H ; определяем ряд
0021	87	00270	ADD A, A
0022	87	00280	ADD A, A
0023	83	00290	ADD A, E ; прибавили "х"
0024	5F	00300	LD E, A ; младший байт
0025	70	00310	LD A, D
0026	1F	00320	RRA
0027	1F	00330	RRA
0028	1F	00340	RRA
0029	E618	00350	AND 018H ; определяем треть экрана
002B	4F	00360	LD C, A
002C	7A	00370	LD A, D
002D	E607	00380	AND 07H ; строка в ряду
002F	81	00390	ADD C
0030	C640	00400	ADD 40H ; старший байт
0032	57	00405	LD D, A
0033	2AA800	00410	LD HL, (INPUT)
0036	ED53A400	00420	LD (OUTLIN), DE
003A	7E	00430	LD A, (HL)
003B	A7	00440	AND A
003C	C8	00450	RET Z ; неверное значение
003D	4F	00460	LD C, A
003E	ED53A600	00470	PROLET LD (OUTPUT), DE
0042	23	00480	INC HL
0043	22A800	00490	LD (INPUT), HL
0046	7E	00500	LD A, (HL)
0047	FEFF	00510	CP OFFH
0049	2016	00520	JR NZ, CHAR
004B		00530	; переход к новой строке

004B		00540		; под первой позицией
004B		00550		; печати
004B	ED5BA400	00560		LD DE, (OUTLIN)
004F	7B	00570		LD A, E
0050	C620	00580		ADD 20H
0052	5F	00590		LD E, A
0053	D0	00600		RET NC
0054	7A	00610		LD A, D ; другая треть экрана
0055	C608	00620		ADD 8
0057	FE58	00630		CP 58H
0059	D0	00640		RET NC
005A	57	00650		LD D, A
005B	ED53A400	00660		LD (OUTLIN), DE
005F	183C	00670		JR REJOIN
0061		00680		
0061	6F	00690	CHAR	LD L, A ; поиск символа
0062	2600	00700		LD H, 0
0064	29	00710		ADD HL, HL
0065	29	00720		ADD HL, HL
0066	29	00730		ADD HL, HL
0067	ED5B365C	00740		LD DE, (CHASET)
006B	19	00760		ADD HL, DE
006C	ED5BA600	00770		LD DE, (OUTPUT)
0070	7A	00780		LD A, D
0071	FE58	00790		CP 58H
0073	D0	00800		RET NC
0074	0608	00810		LD B, 8
0076	7E	00820	EIGHT	LD A, (HL)
0077	12	00830		LD (DE), A ; выдает
0078	23	00840		INC HL ; восемь
0079	7A	00850		LD A, D ; строк
007A	E607	00860		AND 07 ; символа
007C	3C	00870		INC A
007D	FE08	00880		CP 8
007F	200E	00890		JR NZ, NONEW
0081	7A	00900		LD A, D
0082	E6F8	00910		AND 0F8H
0084	57	00920		LD D, A
0085	7B	00930		LD A, E
0086	C620	00940		ADD 20H
0088	5F	00950		LD E, A
0089	3005	00960		JR NC, NEXTSL
008B	7A	00970		LD A, D
008C	C608	00980		ADD 8
008E	57	00990		LD D, A
008F	14	01000	NONEW	INC D
0090	10E4	01010	NEXTSL	DJNZ EIGHT
0092	ED5BA600	01020		LD DE, (OUTPUT)
0096	1C	01030	NEXTLT	INC E
0097	2004	01040		JR NZ, REJOIN
0099	7A	01050		LD A, D
009A	C608	01060		ADD A, 8
009C	57	01070		LD A, D
009D	2AA800	01080	REJOIN	LD HL, (INPUT)
00A0	0D	01090		DEC C
00A1	209B	01100		JR NZ, PROLET
00A3	C9	01110		RET
00A4	01120			
00A4	0000	01130	OUTLIN	DW 0 ; начальная строка
00A6	0000	01140	OUTPUT	DW 0 ; адрес печати
00A8	0000	01150	INPUT	DW 0 ; адрес текста
5C36		01160	CHASET	EQU 23606 ; указывает на
00AA		01170		; адрес шрифта в ПЗУ ми-
00AA		01180		; нус 256 байтов
00AA		01190		; Таблица текста

00AA	07	01200	TXADD	DEFB	7
00AB	4558414D	01210		DEFM	'EXAMPLE'
00AF	504C45				
00B2	0D	01220		DEFB	13
00B3	4F462054	01230		DEFM	'OF TEXT TABLE'
00B7	45585420				
00BB	5441424C				
00BF	45				
00000	TOTAL ERRORS				

В ВАШУ ЗАПИСНУЮ КНИЖКУ

Поскольку в этом номере нет технического приложения, мы начинаем новый цикл, который может заинтересовать всех тех, кто умеет держать в руках паяльник.

Проведя анализ многочисленных зарубежных источников, мы выбрали несколько десятков оригинальных периферийных устройств и в ближайших выпусках дадим о них краткую информацию.

Пусть это будет ответом на довольно многочисленные просьбы наших читателей порекомендовать им, какими разработками стоит заняться. Посмотрев, что бывает для "Спектрума", Вы можете быть определите и область приложения собственных сил.

Сразу оговоримся, что наши источники информации имели рекламный характер и потому более подробной технической информацией по этим устройствам мы не владеем, тем более что как Вы конечно знаете, "ИНФОРКОМ" не занимается аппаратными вопросами.

Для справки в обзоре приводим также названия фирм-производителей и цену в английских фунтах стерлингов.

Интерфейсы дисководов.

1. Disciple.

Фирма: Miles Gordon.

Цена: 69.95 ф. ст.

Поддерживает до двух дисководов с объемом памяти до 800К на каждом. Загрузка программы 48К занимает 3,5 секунды. Совместим с любыми стандартными дисковыми (одно- и двухсторонними, одинарной и двойной плотности; с 40 и 80 дорожками: диаметром 3.0; 3.5; и 5.25 дюйма).

Имеет кнопку snapshot для остановки программы в любом месте и мгновенной выгрузки ее на диск.

Имеет встроенный интерфейс "Центроникс" для подключения принтера, поддерживает при работе с принтером команды LLIST, LPRINT, COPY.

Имеет встроенные порты как под Кемпстон-джойстик, так и под Синклер-джойстик.

Операционная система позволяет "Спектрumu" служить головной станцией для обслуживания принтера, двух дисководов и еще 64-х "Спектрумов".

2. Beta 128 Disk Interface.

Фирма: TECHNOLOGY RESEARCH

Цена: 109.25 ф. ст.

Может поддерживать до 4-х дисководов практически любого типа. Основное достоинство - в наличии кнопки "magic button" для остановки работающей программы и сброса ее на диск.

В нашей стране наибольшее распространение нашли версии именно этого интерфейса.

Основной недостаток - операционная система не обходит испорченные сектора, поэтому для серьезной работы может применяться только с большой осторожностью.

3. Delta Disk Interface.

Фирма: TECHNOLOGY RESEARCH

Цена: 149.00 ф. ст.

Дополнительно к предыдущей модели имеет встроенный интерфейс для принтера "Центроникс" и расширение памяти ОЗУ до 128К.

4. Discovery 1

Фирма: OPUS

Цена: 99 ф. ст. (с дисководом 3,5 дюйма).

Самая дешевая и популярная модель. Отличается высокой надежностью и нашла наибольшее применение в зарубежных странах.

Может поддерживать до двух дисководов. Хотя и поставляется с дисководами 3,5 дюйма, но может дооснащаться дисководами 5,25 дюйма.

Интерфейс, блок питания и до двух дисководов размещаются в едином блоке, от него же питается и сам компьютер.

Дополнительно интерфейс имеет порт джойстика (Кемпстон) и принтера (Центроникс), а также выходной порт для подключения прочей аппаратуры.

Интерфейсы джойстика.

1. Single Port.

Фирма: DK'Tronics

Цена: 9.95 ф. ст.

Имеет один порт для стандартных джойстиков, имитирующий команду IN 31 (Кемпстон-интерфейс).

2. Dual Port.

Фирма: DK'Tronics

Цена: 13.00 ф. ст.

Имеет два порта. Первый - для кемпстон-джойстика, второй - имитирует клавиши 6,7,8,9,0 - Синклер (правый).

3. Programmable Joystick Interface.

Фирма: DK'Tronics

Цена: 14.95 ф. ст.

Позволяет использовать джойстик с любой игрой. Поставляется с программным обеспечением, позволяющим запрограммировать его от клавиатуры.

4. Comcon Programmable Joystick Interface.

Фирма: Comcon

Цена: 19.95 ф. ст.

Программируемый интерфейс для джойстика, очень легко и просто перепрограммируется путем установки шести перемычек в позиции, необходимые для имитации джойстиком действия определенных клавиш. Может поддерживать джойстики с двумя независимыми кнопками, например одна кнопка "огонь", а вторая - "бомбометание". Выбор настройки возможен в любое время, в том числе и когда игра уже полностью загружена.

5. Fox Programmable Joystick interface.

Фирма: Fox Electronics Ltd.

Цена: 24.95 ф.ст.

Программируемый интерфейс. Совместим с любыми программами. Одновременно может хранить в памяти расположение клавиш для 16 программ. Имеет встроенный источник питания на батареях, поэтому не теряет настройку и после отключения. Совместим со всеми джойстиками "Атари"-типа, поддерживает функцию "автострельба".

Звуковые устройства.

1. Digital Sound Sampler.

Фирма: Datel Electronics

Цена: 49.99 ф.ст,

Анализатор звука (оцифровщик, дигитайзер). Позволяет Вам переводить в "двоичную" цифровую форму любые звуки и сигналы и записывать их в память компьютера. Воспроизводится звук может в любой тональности, как вперед, так и назад, с восходящим

тоном или с нисходящим, с эхоэффектом, в замкнутых циклах и т.п.

Поставляется с пакетом программного обеспечения. Кроме перечисленных, имеет еще ряд возможностей.

2. Sound Sampler.

Фирма: CHEETAN

Цена: 45.95 ф.ст.

Анализатор звука.

3. Sweet Talker.

Фирма: CHEETAN

Цена: 24.95 ф. ст.

Синтезатор речи.

4. Mega Sound.

Фирма: CHEETAN

Цена: 10.95 ф. ст.

Усилитель звука.

5. Specdrum.

Фирма: CHEETAN

Цена: 29.95 ф. ст.

Синтезатор звука ударных инструментов.

6. MIDI Interface.

Фирма: CHEETAN

Цена: 49.95 ф. ст.

Интерфейс MIDI для подключения внешних электромузыкальных инструментов.

7. Micon

Фирма: XR1 SYSTEMS

Цена: 109.00 ф. ст.

Это интерфейс типа MIDI высокой сложности. Позволяет подключать внешнее музыкальное оборудование, регулировать звук не только по высоте, но и по длительности и окрасу, причем все это одновременно по восьми каналам. К интерфейсу прилагается математическое обеспечение.

Прочие устройства

1. Snapshot

Фирма: DATEL ELECTRONICS

Цена: 24.99 ф. ст.

Программа "замораживается" нажатием кнопки. Ее можно выгрузить, можно ввести в нее необходимые изменения через POKE. После загрузки программа будет стартовать точно с того места, в котором она была выгружена.

2. Robotics and Model Control (ROBOTEK)

Фирма: DATEL ELECTRONICS

Цена: 29.99 ф. ст.

Имеет 4 независимых выхода для подключения роботов, двигателей, световых источников, бытовой техники и т.п. Имеет 8 независимых входов от чувствительных устройств и т.п. Легкое и удобное в работе устройство поставляется со всеми необходимыми шнурами.

3. Lightwriter

Фирма: DATEL ELECTRONICS

Цена: 14.99 ф. ст.

Это световое перо поставляется с необходимым комплектом математического обеспечения и с соответствующим интерфейсом. С его помощью можно рисовать на экране, в том числе кривые и замкнутые линии, стирать изображения, заполнять контуры, менять цвета INK и PAPER. Все необходимые функции пера выбираются из экрана с помощью самого пера.

4. Kempston Mouse

Фирма: KEMPSTON

Цена: 69.95 ф.ст.

Манипулятор мышь предназначенный для выполнения графических работ, в частности, для изображения гладких кривых.

Мышь совместима с Кемпстон-интерфейсами. К сожалению, математического обеспечения под нее кроме адаптированной версии Артстудии нет, хотя возможно, что Артист-2 тоже с ней совместим. Мышь использует команду IN для определения своего положения относительно курсора на экране.

х - координата: IN 64479

у - координата: IN 65503

Мышь имеет две клавиши.

5. Star Mouse

Фирма: SAGA SYSTEM

Цена: 49.95 ф.ст.

Эта мышь немного меньше, чем ее кемпстоновская сестра. Она имеет только одну клавишу. Ее можно разбирать и чистить.

Единственный пакет графического обеспечения, совместимый с ней - это управляемый через пиктограммы CAD пакет. Панель пиктограмм на экране дает возможность выполнять операции черчения и рисования.

Фирма сделала возможность работать при помощи этой мыши со своим редактором "THE LAST WORD". Эта мышь имеет меньшие возможности, чем предыдущая, но с учетом значительно более низкой цены выглядит привлекательнее.

6. AMX Mouse

Фирма: AMX

Цена: 69.95 ф.ст.

Эта мышь решительно выигрывает по сравнению с двумя предыдущими. Пакет, ее сопровождающий, выглядит внушительно. Многие фирмы, выпускающие математическое обеспечение, неоднократно заявляли о готовности работать под нее, по крайней мере в базах данных, электронных таблицах и графических пакетах. "Артстудия" имеет опцию для этой мыши. Мышь подключается через прилагаемый интерфейс, имеющий одновременно порт "Центроникс" для подключения принтера.

Это единственная мышь из всех, имеющая три клавиши. Они могут независимо программироваться, их потенциал очень велик.

Солидный пакет прилагаемых программ включает в себя "AMX ART" - программу для управления аппаратным обеспечением; "MOUSE CONTROL" - программу для добавления в Бейсик новых 23-х команд, связанных с управлением мышью; "ICON DESIGNER", "CALCULATOR" и "PUZZLE".

7. Datapen

Фирма: DATAPEN MICROTECHNOLOGY

Цена: 20.95 ф.ст.

Это светоперо имеет характеристики, которых нет у других. Операционная кнопка находится на перо, что дает возможность не использовать клавиатуру. Оно может работать в

любых условиях внешнего освещения. Работает с точностью до пикселя.

Имеет 20 заранее определенных команд, позволяющих наносить геометрические объекты (треугольники, линии, окружности и т.п., а также текст и символы графики пользователя). Работает с цветом, имеется накладываемая сетка, легкость стирания, возможность выгрузки рисунков на ленту.

8. Videoface

Фирма: DATA-SCIP

Цена: 69.00 ф.ст.

Устройство предназначено для оцифровывания изображений. Совместимо с микродрайвом, а также дисковыми системами "Опус" и "Бета".

Время оцифровки экрана - 0. 27 сек. Все математическое обеспечение управляется от меню. Имеет возможность хранить в памяти 6 полных экранов. Может применяться в развлекательных и профессиональных целях.

Необходимые шнуры прилагаются.

9. Scorpion

Фирма: MICRO-ROBOTICS

Цена: 249.00 ф.ст.

Этот контроллер содержит 24К ОЗУ и 32К ПЗУ. Он открывает необъятные возможности по части подключения аналоговых, цифровых источников сигналов, также для управления сервомеханизмами, светодиодными экранами, источниками инфракрасного излучения и многое др. Если у Вас в голове есть какой-нибудь проект по использованию "Спектрума", Вы без сомнения найдете возможность применить "Скорпион".

Существенный недостаток - очень высокая цена.

10. Spectrum Doctor Screenex

Фирма: SCREEN MICROCOMPUTERS DISTRIBUTION

Цена: 57.43 ф.ст.

Устройство предназначено в первую очередь для профессионалов.

"Доктор" подключается к внешнему разъему "Спектрума" и прогоняет через компьютер серию тестов. Телевизор Вам уже не нужен, т.к. на панели прибора имеются 15 светодиодов, показывающих состояние систем компьютера. Выполняются проверки питания, адресной шины и шины данных, блока ULA и процессора.

Проверяет наличие неправильных байтов в ПЗУ и исправность ОЗУ. Для проверки ОЗУ низших 16К и верхних 32К выполняются отдельно.

Скорость и простота диагностирования таковы, что блок может быть полезен тем, кто выполняет ремонтные работы.

ADVENTURE PROJECT

Игра - дело серьезное, а игра адвентюрная - дело серьезное вдвойне. Если Вы полагали, что дав серию статей по методике работы с адвентюрными игровыми программами, мы с ними закончили, то Вы чуть-чуть ошибаетесь.

Сегодня мы начнем новый цикл статей и покажем на примере адвентюрных программ как могут применяться принципы искусственного интеллекта. А это уже согласитесь дело совсем нешуточное и, возможно, знание этих элементарных базовых принципов поможет когда-нибудь в будущем в Вашей практической работе.

Первые адвентюрные игры в основном представляют собой просто путешествие по игровым полям (локациям), информация о которых содержится в памяти ЭВМ. Но с течением времени программы стали усложняться, и в первую очередь усложнение пошло как по пути введения в игру элементарной графики, так и по пути введения некоторых базовых концепций, связанных с искусственным интеллектом. Первыми образцами такого подхода стали The Hobbit и Sherlock фирмы MELBOURNE HOUSE и в какой-то степени Valhalla фирмы LEGEND.

Такой подход позволил дать игровым персонажам возможность вести в общем программном окружении как бы собственную жизнь. В программах появился эффект "реального времени". И Ваш главный герой, которым Вы управляете, получает возможность общаться с другими персонажами и "жить" в игровом пространстве.

И, как Вы увидите, делается это совсем несложно. Все, что для этого надо - это определить правила поведения героев и правила их реакции на игровые ситуации, возможные в ходе игры.

Вы, должно быть, знаете, что когда дело доходит до обработки правил, то очень хорошо подходит язык ПРОЛОГ, который кстати для "Спектрума" тоже реализован в версии "ZX-Микропролог", но мы в рамках данной статьи будем обсуждать базовые концепции на примерах, написанных на БЕЙСИКЕ.

Конечно, БЕЙСИК далеко не идеален, но зато понятен всем, а уж применять основные принципы Вы можете где хотите и на том языке, какой Вам будет удобнее - ПАСКАЛЬ, СИ и др.

Мы не будем останавливаться на методике разработки сценария игры - это важный момент, но не тема данной статьи. Для этого Вам необходимы фантазия, вдохновение, художественный вкус. Скажем только, что как правило ситуации реальной жизни имеют более мощный эффект, чем надуманные конструкции.

Когда сценарий игры продуман, на бумаге расписывается карта локаций, по которым могут путешествовать персонажи.

Программа может иметь несколько игровых пространств, перемещение между которыми выполняется подъемом по лестницам, спуском в ущелья и т.п. В таком случае локации помечаются сначала меткой, указывающей на уровень, а затем на координаты позиции. Эти метки локаций могут не иметь никакого отношения к тому, с какими описаниями они появятся в игре, но служат как бы путеводителем программисту и дизайнеру программы, особенно если это разные люди.

Предположим, что в одном академическом НИИ, расположенном на морском берегу, группа налетчиков похитила уникальный опытный экземпляр часов, способных управлять развитием Вселенной. Преступники скрылись на моторной яхте, а часы перепрограммировали на обратный ход. Время пошло вспять.

Вас зовут Инф. Вам, с Вашими помощниками Орой и Комом, надо найти налетчиков и вернуть часы до того, как обратный ход времени превратит Вас в ребенка или вообще уничтожит Вас как еще не появившегося на свет.

Игра начинается на берегу моря. Двигаясь вправо, Вы попадаете в моторную лодку. Остальные направления показаны на рис.1.

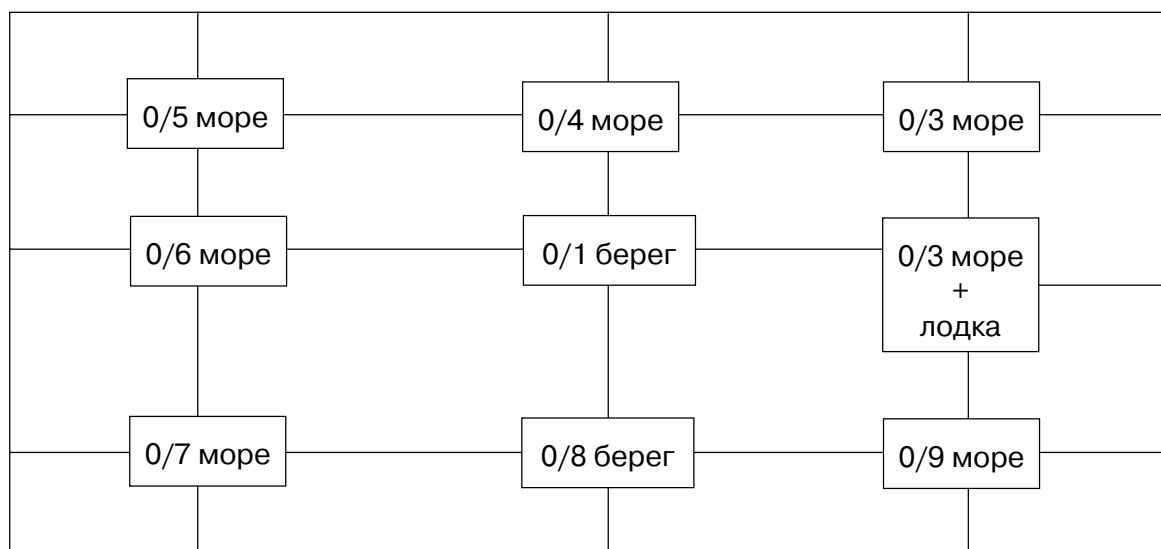


Рис. 1

На этом этапе Вам для описания локации достаточно одного-двух слов, но Вы-то конечно помните, что в игре каждая локация будет появляться со своим каким-то достаточно подробным описанием.

Лучше всего завести отдельную картотеку. Пометьте каждую карточку шифром, определяющий номер локации. Затем дайте краткое описание, полное описание и список объектов (предметов), которые могут быть здесь обнаружены.

Карточка может выглядеть примерно так, как показано на рис.2.

0/2 МОРЕ

Вы упали в ледяную воду. От холода у Вас начинают неметь конечности. Еще немного, и вы утонете.

+ ЛОДКА

Вы взобрались на борт моторной лодки, слегка покачивающейся на волнах.

ПРЕДМЕТЫ:

- ключ зажигания;
-

Рис. 2

В качестве уровня игрового пространства здесь взят 0. Всякий уровень, находящийся выше, будет обозначаться положительным числом, а уровень ниже - отрицательным. Например, когда Вы настигнете налетчиков и подниметесь на борт яхты, уровень будет +1, а когда спуститесь в трюмное помещение, то -1.

Когда Вы перейдете от разработки проекта непосредственно к программированию, то обнаружите, что локации и объекты могут конкурировать друг с другом и сделать программирование более сложным, чем ему положено быть. Например, лодка может поставить Вас перед проблемой, которая хоть и не очевидна, но оказывает влияние на структуру программы и ход игры. Проблема в том, чем считать лодку - локацией или объектом, поскольку она перемещается.

Если ее считать объектом, то можно ею не заниматься до тех пор, пока не дойдете до размещения объектов. Описание локации в этом случае такое, как на рис.2.

Учет лодки как объекта, имеет определенные преимущества, поскольку тогда ее можно перемещать по ходу игры. Более того, с ней могут перемещаться и персонажи. С точки зрения гибкости игры гораздо удобнее иметь лодку и перемещаться с нею по

локациям, чем описывать путь по фиксированным локациям пространства игры. При таком подходе единственная структура данных, необходимая нам пока - это обычный массив.

На рис.3 показан текст программы, описывающей приведенную выше карту в виде двумерного массива размером 3x3. Каждый элемент массива имеет место для 10 символов для краткого описания локации.

```
10 DIM g$(3,3,10)
20 FOR i=1 TO 3
30 FOR j=1 TO 3
40 READ g$(i,j)
50 NEXT j: NEXT i
60 DATA "Море", "Море", "Море"
70 DATA "Море", "Берег", "Море"
80 DATA "Море", "Берег", "Море"
90 REM PRINT ROUTINE
100 FOR i=1 TO 3
110 FOR j=1 TO 3
120 PRINT g$(i,j); " ";
130 NEXT j
140 PRINT
150 NEXT i
```

Рис.3

Если Вы пишете программу в машинном коде, Вы сведете все данные в единую таблицу, а на конкретный элемент будете указывать с помощью индекса (смещения).

Конечно, Вы понимаете, что приведенный выше пример расточительно использует ограниченную память компьютера. Так, семь раз повторяется локация с описанием "море" и два раза "берег". Вы, конечно, найдете способы сэкономить здесь немалый объем. Надо также иметь в виду и ограничения по скорости, если работаете на БЕЙСИКе. "Спектрум" может затратить пол-минуты на то, чтобы обработать массив всего лишь 100x100.

Теперь перейдем к проектированию персонажей. Этот процесс состоит из трех основных этапов:

- характеристика;
- изменчивость;
- коммуникабельность.

Последние два взаимосвязаны. Действительно, взаимодействие между персонажами по ходу игры неминуемо приводит к изменению их действий. Любые взаимодействия отражаются на атрибутах, присвоенных персонажам, а те в свою очередь приводят к изменениям во взглядах на внешний мир и, соответственно к изменению поведения по отношению к другим персонажам.

Действия и перемещения одного персонажа вызывают перемещения других персонажей (и объектов!) в игровом пространстве. Например, в зависимости от смелости одного персонажа он может взять или не взять какой-либо объект на борту вражеской яхты (пищу или, скажем оружие), а это действие может заставить страдать другой персонаж в определенное время (от голода или от безоружности). По этой причине фактор изменчивости является важнейшим при проектировании игры с элементами ИИ (искусственного интеллекта).

Первый этап - задание характеристик персонажам понятен. Его сложность зависит от того, насколько сложной Вы планируете сделать программу. Уже давно в играх сложилась определенная система атрибутов, которая включает в себя:

- силу;
- интеллект;
- ловкость;
- пси-энергию;
- стойкость.

Если с первыми тремя все понятно, то пси-энергия характеризует магические способности героя, а стойкость - способность выдержать определенное количество ударов, не будучи травмированным.

Вы видите, что такая система атрибутов нацелена на игры, в которых сражение - это естественное состояние героев, для чего собственно и идет игра.

Эта система не годится для более интеллектуальной программы с более сложным поведением персонажей, поскольку никаких качеств характера героя она не описывает, а потому не сможет определять его самопроизвольное поведение в игровой среде, что необходимо для программы с искусственным интеллектом.

Желательно, чтобы в программе герои могли бы испытывать чувства, например, такие, как боль, страх, ненависть и другие, неотъемлемо присущие человеку.

Вернемся к нашему проекту программы. Предположим, пока Ком остается в лодке, Вы с Орой ныряете в пучину моря, где во мраке видите очертания затонувшей яхты и силуэты двух чудовищ, охраняющих ее - Крама и Брога. Они сильны, но малоподвижны. В то же время, у них очень неплохо с интеллектом. Настолько неплохо, что они не будут ввязываться в драку или продолжать ее, если у них нет стопроцентной уверенности закончить ее без особого ущерба для себя. Атрибуты, которыми может быть в дальнейшем описано поведение всех персонажей, могут быть такими, как показано на рис.4.

Атрибут	Изменчивость	Диапазон изменения +10/-10
Эмоции	x	стабильные / нестабильные
Энергия	x	энергичный / вялый
Сила	x	большая / малая
Интеллект		гений / тупица
Совесь	x	большая / отсутствует
Образованность		высокая / низкая

Рис. 4

Отмеченные знаком "x" атрибуты могут меняться по ходу игры. Диапазон изменения от "-10" до "+10". Так, например, если Ваша энергичность имеет атрибут "-5", то в очередной ситуации, в которой она будет учитываться, она зачтется не в Вашу пользу. Если по ходу программы Вы будете делать ошибки, то атрибуты эмоций, силы и совести также будут изменяться.

Ключ к использованию этих атрибутов в системе искусственного интеллекта состоит в том, как они взаимосвязаны, как влияют один на другой и как влияют на поведение персонажей. Нет больше однозначно хороших или однозначно плохих героев. Теперь Вы можете покорять врагов, а можете даже делать из них друзей.

Конечно, по ходу игры эти атрибуты могут изменяться сотни раз, но в начале игры они должны быть заданы, например, так, как показано на рис.5.

Атрибут	Инф	Ора	Крам	Брог
Эмоции	5	-2	-3	2
Энергия	4	-1	6	-3
Сила	3	2	4	4
Интеллект	7	2	4	4
Совесь	6	5	-4	-3
Образованность	6	6	1	2

Рис. 5

Определив систему атрибутов, Вы теперь можете выработать критерии поведения своих персонажей, образовав комплексные атрибуты, см. рис.6.

Характеристика персонажа	От чего зависит
Гнев	Энергия, Интеллект, Сила
Мужество	Энергия, Сила, Совесть
Трусость	Энергия, Сила, Совесть
Счастье	Интеллект, Эмоции, Энергия
Настроение	Эмоции, Энергия, Совесть

Рис. 6

Мы позже займемся выводением формул для расчета характеристик персонажей через их атрибуты, но сейчас имейте в виду, что все здесь зависит от Вас. Вы сами можете задать эти формулы. Например:

мужество = энергия + сила + совесть

или скажем:

мужество = 2*энергия + 5*сила + 3*совесть.

Вы же можете вычислить и другие какие-либо характеристики, не указанные выше, например, способность к самопожертвованию.

Скоро Вы увидите, что зная атрибуты персонажей, Вы сможете на глаз предсказать их поведение в той или иной ситуации, а также что произойдет, если например Инф совершит какое-то действие.

Благодаря такому подходу, персонажи могут приобрести некоторую автономию. Они смогут сами "решать", что им делать в той или иной ситуации, а не быть марионетками в руках играющего. Такая игра становится намного интереснее, неожиданнее, да и вообще меняется сама ее природа.

После того, как данные по персонажам определены в исходной таблице (рис.5), можно приступить к программированию. Предельно простой сценарий показан на рис.7.

```

100 LET c$="":LET p$="":LET n$=""
110 REM заполняем банк атрибутов
120 DIM a(4,6):REM - 4 персонажа, 6 атрибутов
130 RESTORE 190
140 FOR i=1 TO 4
150 FOR j=1 TO 6
160 READ a(i,j)
170 NEXT j
180 NEXT i
190 DATA 5,4,3,7,6,6: REM - Инф
200 DATA -2,-1,2,2,5,-6: REM -Ора
210 DATA -3,6,4,4,-4,1: REM - Крам
220 DATA 2,-3,4,4,-3,2: REM - Брог
230 REM - начало экспериментального сценария
240 CLS
250 PRINT "Мрачные очертания проступают"
260 PRINT "перед Инфом и Орой."
270 PRINT "Они видят корпус яхты, на"
280 PRINT "которую указал локатор."
290 REM - ввод противников
310 IF a(3,2) OR a(3,3) OR a(3,5)<-1 THEN LET c$="Брог":
    LET f$="одно чудовище охраняет":
    GO TO 340
320 IF a(4,2) OR a(4,3) OR a(4,5)<-1 THEN LET c$="Крам":
    LET f$="одно чудовище охраняет":
    GO TO 340
330 IF c$="" THEN PRINT "яхта появляется перед Вами":

```

```

PRINT "дверь в рубку открыта":
GO TO 380
340 REM
350 PRINT "Корпус яхты прорисовывается":
360 PRINT "в темноте"
370 PRINT f$;"дверь в рубку.":
GO TO 400
380 PRINT "Вы проходите в помещения"
390 STOP
400 REM - решение стоит ли сражаться
410 IF a(1,2) AND a(1,3) AND a(1,5)<-8 THEN LET p$="Ора":
LET n$="Инф":
GO TO 440
420 IF a(2,2) AND a(2,3) AND a(2,5)<-8 THEN let p$="Инф":
LET n$="Ора":
GO TO 440
430 LET p$="Инф и Ора": STOP
440 REM - один из них уплывает
450 PRINT n$; "Холодеет при мысли о схватке и"
460 PRINT "уплывает к лодке."
470 PRINT p$; "остается с чудовищем"
480 PRINT "один на один": STOP

```

Рис. 7

Вам должно быть понятно из этого примера как учитывается влияние атрибутов на поведение персонажей. Возьмем чудовищ.

Строка 310. Если у Крама не все в порядке с силой или с энергией или с совестью, он не выползает на поле битвы.

То же самое делает и Брок в строке 320.

Если ни тот, ни другой не охраняют вход, то через строку 330 Вы с Орой проникаете на яхту.

Аналогично и с положительными героями. Их поведение рассматривается в строках 400.

Есть только одна особенность. Чудовища анализируются оператором OR, а ваши герои оператором AND. Это сделано специально, чтобы у первых было больше оснований не вступать с Вами в схватку, ведь если ИЛИ энергии, ИЛИ силы, ИЛИ совести у них мало, они уйдут.

Для вас же наоборот, надо, чтобы ОДНОВРЕМЕННО И того, И другого И третьего не хватало, чтобы бросить товарища в беде. Конечно, это менее вероятно и возможно лишь в совсем отчаянном положении, когда действительно сражаться бесполезно, а надо думать о конечной задаче.

Вам должно быть ясно, что приведенный пример страшно расточителен по расходу памяти и предельно примитивен. Конечно, на БЕЙСИКе реально такие вещи не пишут. Но как иллюстрация концепции он годится и должен быть понятен. Степень сложности, которую Вы определите для себя - дело Ваше. Вы можете анализировать сотни факторов, составлять любые уравнения для расчета критериев - это ваше право.

И напоследок обратим здесь еще внимание на один аспект взаимосвязи. Предположим, что у Брока был кинжал. Что тогда, если он не вышел на битву? Вам стало легче от того, что Вы не пострадали, но зато Вы лишились возможности завладеть кинжалом и впоследствии его использовать. Вы даже никогда и не узнаете, чего лишились - вот что самое интересное. Это тот случай, когда программа начинает жить собственной жизнью, независимо от Вас.

(Продолжение в следующем выпуске)

Развивая начатый разговор об приключенческих играх, мы сегодня даем две полных инструкции по работе с программами этого жанра. Первая программа - Sherlock - классический представитель этого направления, давно завоевавшая устойчивую популярность. Вторая программа - Miami Vice. Ее особенность в том, что это одна из программ, объединивших в себе приключенческий жанр с жанром деловой игры (MANAGEMENT) на базе криминального сюжета. В последние годы таких программ стало немало - это The Sidney Affair (INFOGRAMES), Vera Cruz Affair (INFOGRAMES), The Fourth Protocol (CENTURY), и многие многие другие.

SHERLOCK



Фирма MELBOURNE HOUSE.

Год выпуска: 1985.

Игра относится к жанру приключенческих (текстовых). Действие происходит в Лондоне конца XIX века и его ближайших окрестностях. Вы играете роль Шерлока Холмса и должны раскрыть ряд преступлений. Если Вы погибнете, игра прекратится.

В своих путешествиях Вы встретите многих персонажей, окажетесь в странных ситуациях. Люди и предметы откроются Вам совсем не такими, какими казались на первый взгляд. Ваша работа похожа на работу кукловода - Вы будете контролировать все, что Холмс будет делать или говорить. Свои команды Вы будете отдавать компьютеру на простом "Инглийском" языке (это разновидность английского). Словарь у компьютера большой. Он содержит свыше 800 слов, может выполнять 53 разных действия.

Вы вводите свои указания в виде простых предложений. Верхняя часть экрана служит для изображения места действия и описания ситуации. Графическое изображение появляется, когда Вы входите в новое место, но при повторном входе его уже не будет. Чтобы вызвать рисунок, надо дать команду LOOK (смотреть).

Здесь же описываются все действия, совершаемые Вами или другими персонажами. You take the note (Вы берете записку). Watson examines the pistol (Уотсон изучает пистолет). Нижняя часть экрана служит для связи между Вами и компьютером. Все, что Вы наберете, будет показано здесь заглавными буквами. Здесь же изображаются запросы и подсказки от компьютера, например: I don't understand the word "DOR" (Я не понимаю слово "DOR").

Стирание неправильного символа - CAPS SHIFT + 0.

Стирание всей командной строки - SYMBOL SHIFT + 0.

Правила "инглийского" языка.

1. Каждое предложение должно иметь глагол. Оно может состоять только из глагола.

SIT - сидеть.

CLIMB - взбираться.

Значение глагола может быть изменено добавлением наречия:

FOLLOW QUICKLY - быстро преследовать.

QUICKLY COUNT THE MONEY - быстро пересчитать деньги.

2. Глагол должен стоять первым.

Правильно:

OPEN THE WINDOW CAREFULLY WITH THE STICK - открыть окно с помощью палки.

Неправильно:

WITH THE STICK CAREFULLY OPEN THE WINDOW - палкой осторожно открыть окно.

3. Прилагательные должны стоять перед существительными.

OPEN THE GREEN DOOR - открыть зеленую дверь.

4. Предлоги, такие как WITH (с); UNDER (под); ON (на); OFF (из) и др. обычно должны стоять перед существительными.

OPEN THE DOOR WITH THE KEY - открыть дверь с помощью ключа.

PICK UP THE NOTE - поднять записку.

Иногда они могут стоять и после существительного. Допускаются, например:

TURN THE LIGHT ON - включить свет.

PICK THE NOTE UP - поднять записку.

5. Можно увязывать несколько команд в одну с помощью союза AND (и), например:

PICK UP THE NOTE AND TAKE THE LAMP OUT OF THE HOUSE - поднять записку и вынести лампу из дома.

TAKE THE MONEY AND RUN - взять деньги и убежать.

6. Можно применять запятые и точки, но фраза не должна быть длиннее 128 символов.

TAKE MONEY, RUN - взять деньги, убежать.

PICK UP THE NOTE. TAKE THE LAMP OUT OF THE HOUSE - поднять записку. Вынести лампу из дома.

Всегда имейте в виду, что в ответ на Ваши действия другие персонажи действуют независимо. Если Вы дадите слишком много инструкций в одной команде, то результат может оказаться неожиданным.

Шерлок умеет "говорить". Это позволяет Вам инструктировать других персонажи. Вы можете, по крайней мере дать ему команду сказать кому-либо, что Вы хотите, чтобы тот сделал. Разумеется, поскольку эти персонажи независимы, то они сами решают делать ли им то, что Вам нужно.

SAY TO WATSON "READ NOTE" - сказать Уотсону: "прочитай записку".

Заметьте, что то, что Шерлок должен сказать, заключено в кавычки. К тому, что он говорит применимы также все правила "английского" языка. Старайтесь не говорить очень много одному персонажу за один раз, т.к. ему это может надоесть, и он не захочет Вам помогать.

Другие персонажи, в отличие от компьютера, не сообщат Вам о том, что они не понимают того, что им говорят, а просто сочтут Ваше поведение довольно странным. И если им Ваши действия покажутся бессмысленными, вряд ли они станут помогать Вам.

Если Вы, например, начали свой диалог с Уотсоном так: SAY TO WATSON "HELLO" - сказать Уотсону "привет", - то далее уже не надо повторять команду SAY TO WATSON. Дальнейшая беседа идет просто путем применения кавычек. "READ THE DAILY CHRONICLE" - прочитай "Дэйли Хроникл". Если же после этого Вы захотите обратиться еще к кому-либо, то опять надо будет начать с SAY TO.

Обсуждение.

Холмс не только может давать указания другим персонажам, но может обсуждать обстоятельства дела вместе с Уотсоном и Лестрейдом. При этом он может отвечать на вопросы, которые они ему ставят. То есть могут быть составлены предложения, которые не вызывают немедленных действий со стороны персонажей. Лестрейду и Уотсону можно, например, предложить о чем-то подумать.

Такие заявления в адрес других персонажей не могут начинаться с глагола (чтобы отличаться от команд) и должны быть построены по одной из двух форм. Первая форма

описывает взаимоотношения между субъектом и объектом: BROWN KILLED SMITH - Браун убил Смита. MAJOR FOULKES HAS THE PISTOL - Майор Фоулкс имеет пистолет.

Вторая форма содержит глагол TO BE (быть) для описания какого-либо аспекта. THE GUN IS GREEN - Пистолет зеленый. WATSON IS INNOCENT - Уотсон невиновен.

Очень часто Холмсу удается извлекать необходимую ему информацию из свидетелей. Это можно сделать командой TELL ME ABOUT ... - Расскажите мне о ...

Естественно, в начале перечня должна стоять инструкция SAY TO. Например, SAY TO WATSON "TELL ME ABOUT THE PISTOL" - сказать Уотсону "Расскажите мне о пистолете".

Вы можете расспрашивать кого угодно и о чем угодно - об информации, о людях, о предметах, об их адресах и об алиби.

Обобщение команд.

Вам может быть неудобным давать описание всем предметам, которые Вы хотите, например, поднять. Для этого Вы можете обобщать свою команду, используя слова ALL (все), EVERYTHING (все), EXCEPT (за исключением). Например:

ALL BOTTLES - все бутылки;

EVERYTHING EXCEPT GREEN BOTTLES - все, кроме зеленых бутылок;

EXAMINE EVERYTHING - обследовать все;

OPEN ALL EXCEPT THE GREEN DOOR - открыть все, кроме зеленой двери;

CLOSE ALL DOORS EXCEPT THE GREEN ONE - закрыть все двери, кроме зеленой.

Здесь также можно использовать союз AND (и) для получения более сложных структур: TAKE THE NOTE AND THE LAMP OUT OF THE HOUSE - вынести записку и лампу из дома.

DROP THE SHORT AND THE LONG ROPES - бросить короткую и длинную веревки.

Сокращения.

Вполне возможно быть максимально кратким в диалоге с компьютером без нарушения взаимопонимания. Конечно, чем короче Ваше предложение, тем больше вероятность непонимания.

Рассмотрим игровую ситуацию. На экране горит описание You are in your sitting room. To the west there is your door to the north is a green door. - (Вы находитесь в гостиной. К западу от Вас расположена Ваша дверь, а к северу - зеленая дверь). Предположим, что Вы решили дать команду:

OPEN THE GREEN DOOR - открыть зеленую дверь, но для краткости набрали только слово:

OPEN - открыть. Для Вас ситуация ясна, но компьютер ответит:

OPEN WHAT? - что открыть?

Если теперь Вы дадите команду OPEN ALL - открыть все, то увидите результат своих действий:

You open your door. - Вы открываете Вашу дверь.

You open the plain chest. - Вы открываете шкаф.

Если теперь Вы опять дадите команду OPEN, то получите ответ:

I SEE NOTHING TO OPEN - я не вижу, что можно открыть.

Если бы Вы сразу дали команду

OPEN YOUR DOOR - открыть Вашу дверь, то сразу бы и получили результат: YOUR DOOR IS OPEN - Ваша дверь открыта.

Как видите, за краткость иногда приходится платить.

Описывая объект, Вы можете давать его название и при нем определения (если они есть). Предположим, что Вы видите вкусное пенящееся пиво в бутылке. Тогда можно дать одну из следующих команд:

DRINK BEER - пить пиво;

DRINK DELICIOUS BEER - пить вкусное пиво;

DRINK FOAMING BEER - пить пенящееся пиво;

DRINK DELICIOUS FOAMING BEER - пить вкусное пенящееся пиво.

Во всех случаях Вы утолите жажду. Нельзя использовать местоположение предмета в

качестве его определения:

DRINK BEER IN BOTTLE - пить пиво в бутылке. Это неправильно, т.к. BOTTLE - не прилагательное.

Нельзя также делать что-либо с какой-либо вещью более чем одним образом. Например, можно сказать:

PUT THE ROPE ON THE CHAIR - положите веревку на стул, или:

PUT THE ROPE ON THE TABLE - положите веревку на стол.

Но нельзя сказать: PUT THE ROPE ON THE TABLE AND CHAIR - положите веревку на стол и на стул. И тем более нельзя: PUT THE ROPE ON EVERYTHING - положите веревку на все.

Движение и перемещение.

После того, как Вы приходите на новое место, на экране появляется изображение и игра останавливается на время, чтобы Вы могли рассмотреть экран. Нажатие любой клавиши продолжит игру.

Существует несколько способов для перемещения из одного места в другое.

1. С помощью курсорных клавиш 5,6,7,8 - соответственно: запад, юг, север, восток. При этом эта клавиша должна быть первой в команде. Этим методом нельзя двигаться вверх (UP) или вниз (DOWN), а также например на северо-восток (NORTHEAST).

2. В восьми направлениях, а также вверх и вниз можно двигаться, набирая команду по буквам:

E

EAST - восток

GO EAST - идти на восток

QUICKLY GO EAST - идти на восток быстро.

Перемещение в заданное место.

Если Вы знаете, куда Вам надо пройти, то можете давать команду, используя место назначения:

GO TO LODGINGS - идти к домику.

Если Вам надо переместиться в удаленное место, то Вы можете пользоваться кэбом или поездом, но для этого Вам нужны деньги. Чтобы успеть на поезд, Вы должны знать время его отправления.

Кэбы и поезда.

Кэб можно нанять на большинстве лондонских улиц. Но извозчик ничего не знает, кроме названия улиц. Так, чтобы поехать на вокзал Виктория, надо дать команду: SAY TO CABBIE "GO TO BUCKINGHAM PALACE ROAD" - сказать извозчику "ехать на Бэкингем Палас Роуд".

Чтобы попасть на поезд, Вы должны приехать на нужный вокзал и пройти на нужную платформу. Во время движения в кэбе или на поезде время идет в реальном масштабе. В это время Вы можете обсуждать с Уотсоном обстоятельства дела или отдыхать.

Вход и выход.

Если Вы знаете свое местоположение, например перед входом в строение, можете задавать команду на вход например так:

ENTER LODGINGS - войти в домик

GO INTO LODGINGS - пройти в домик.

Можете проходить через открытые двери:

GO THROUGH THE DOOR - пройти в дверь.

Можно также проходить через окна (WINDOWS) и т.п. Сквозь них можно также смотреть (LOOK). Это может быть очень полезным, если Вы хотите знать, куда попадете, если пройдете в этом направлении. Бывает можно увидеть, кто Вас там ждет!

Преследование других людей.

Если Вы знаете, куда отправился тот, с кем Вы хотите поговорить, то можете отправиться за ним следом, например:

FOLLOW WATSON - идти за Уотсоном.

Хождение по улицам Лондона.

В этой игре Ваши возможности хождения по Лондону ограничены. Это согласуется с характером Холмса и вызвано ограничением памяти компьютера. Он никогда не ходил пешком туда, куда можно доехать в кэбе.

Ход времени.

Течение времени оказывает большое влияние на игру. Даже когда Холмс ничего не делает, а только рассуждает, время идет и другие персонажи действуют.

Игра идет в реальном времени. Оно изображается на экране телевизора и все время персонажи независимо действуют. Взаимосвязь событий во времени - критический фактор в данной игре.

Единственно, когда компьютер не работает - это во время набора очередной команды.

Течение времени может быть ускорено командой WAIT (ждать).

День и ночь.

Поскольку время идет, день сменяется ночью. Если Вы попадете без фонаря в темное место, то ничего не увидите. К счастью, в некоторых местах есть искусственное освещение. В темноте Вы можете потерять ориентиры, ждать в темноте опасно.

Сон.

Холмс, когда глубоко погружается в дело, может подолгу обходиться без сна, но такой энергии нет у окружающих его людей. Им приходится спать. Однако помните, что большинство наиболее темных дел вершится под покровом ночи.

Ускорение времени.

Если Вы дадите команду WAIT (ждать), то некоторое время ничего не будете делать. Будьте осторожны, последствия могут быть непредсказуемы. Команда WAIT имеет два формата:

WAIT - по ней Вы ничего не делаете в течение 5 минут.

WAIT UNTIL ... - ждать до ... - по этой команде Холмс ничего не делает до заданного времени.

Например.

WAIT UNTIL 10 - ждать до 10 часов;

WAIT UNTIL 11 PM - ждать до 11 часов вечера;

WAIT UNTIL 9:30 AM - ждать до 9:30 утра.

Поведение персонажей.

Все персонажи игры живут собственной жизнью и ведут себя относительно независимо. Их поведение зависит как от Ваших действий, так и от действия других персонажей и поэтому в казалось бы равных ситуациях оно может быть различным. Каждая ситуация в игре является уникальной и однозначного решения игры не может быть. Повторяя игру снова и снова, Вы будете сталкиваться с новыми ситуациями. Играть в нее можно не один раз. Всякий раз это будет новая игра.

Сотрудничество с другими персонажами.

Многие проблемы игры могут быть разрешены только если Вы проявите готовность к сотрудничеству с другими героями. Столько много дел надо сделать Холмсу, что без чужой помощи не обойтись. Хотя он и работает независимо от полиции, тем не менее ему нужна ее помощь для отыскания дополнительных улик и получения информации.

Деньги.

Без них Вы далеко не уйдете. Вам надо покупать билеты на поезд, оплачивать информацию и другие услуги. Фунты вводятся в следующем формате:

.. 99 - 99 фунтов

Для фунтов и шиллингов:

.. 99 19S. или .. 99 19/-

Только для шиллингов:

10/- или 10S.

Для шиллингов и пенсов:

13/11

Только для пенсов:9d.

Команда на уплату денег:

PAY или PAY TO:

PAY TO WATSON .. 1 - уплатить Уотсону один фунт.

PAY 6D TO THE CABBIE - уплатить 6 пенсов извозчику.

Стратегия игры.

Мы Вам советуем во время игры создать собственную карту, чтобы Вы могли легко возвращаться назад.

Когда Вы впервые приходите на какое-то место, компьютер дает Вам его подробное описание и список возможных выходов. При повторном посещении описание очень краткое. Если Вам нужны подробности, надо дать команду LOOK.

В игре имеются и определенные физические законы.

1. Вы не можете поднять предмет, который превышает Ваши силы, а также если Вы уже несете тяжелый груз. Это же относится и к остальным персонажам игры, но если они крепче Вас, то и поднять смогут больше.

2. Вам не обязательно владеть предметом (нести его), которым Вы собираетесь как-то воспользоваться. Если камень лежит на земле, Вы можете давать команду:

THROW STONE AT WINDOW - бросить камень в окно.

Исключение составляют предметы, которыми уже владеют кто-нибудь из других персонажей игры.

3. Некоторые предметы могут играть роль емкостей или контейнеров. Это чемоданы, бочки и т.п.

Нельзя положить в контейнер предмет, если он слишком велик для этого. Очевидно нельзя этого сделать и если емкость закрыта.

4. Некоторые емкости могут быть прозрачными, а некоторые нет. Вы можете увидеть, что находится внутри прозрачного контейнера, а для непрозрачного надо его сначала открыть. Для этого может потребоваться ключ.

5. Некоторые предметы могут ломаться. Вы должны быть осторожны в обращении с ними. Например, если Вы дадите команду сломать дверь с помощью бутылки, результат может быть совсем противоположным - разобьется бутылка.

Повторение команд.

Нажатие @ приводит к повторению ранее поданной команды.

Изучение текущей ситуации.

1. LOOK. Можно давать сокращенно "L".

2. LOOK THROUGH объект. По этой команде дается изображение и описание того, что можно увидеть через открытую дверь или, скажем, окно.

3. INVENTORY. По этой команде выдается описание всего того, что Вы имеете в своем распоряжении. Сокращение - "I".

4. EXAMINE объект. Эта команда позволяет более детально рассмотреть любой предмет, находящийся в пределах вашей досягаемости.

Прерывание игры и выгрузка позиции.

1. PAUSE - игра приостанавливается до тех пор, пока не будет нажата какая-либо клавиша.
2. SAVE - по этой команде производится выгрузка незаконченной игры на ленту. После выгрузки игра продолжается как обычно.
3. LOAD - загрузка ранее отложенной позиции.
4. QUIT - прекращение текущей игры.
5. PRINT - по этой команде включается принтер, если он подсоединен.
6. NOPRINT - эта команда отбивает команду PRINT.

MIAMI VICE



Фирма: "OCEAN"

Год выпуска: 1987

Цель игры - сорвать крупную сделку по переправке контрабанды.

Кроккет и Таббс получили сведения о том, что утром в четверг готовится доставка в город крупной партии контрабанды стоимостью миллион долларов. Получатель - некто мистер "J" - гангстер со стажем, ныне занимающийся респектабельной деятельностью и имеющий влиятельных покровителей.

Известно только, что сделка состоится в одном из портовых заведений и что произойдет она между полуночью воскресенья и утром четверга.

Ваша задача - выехать на место и внедриться в сеть, начав с мелких агентов, обычных завсегдатаев портовых баров. Если вести себя с ними правильно, то через них можно выйти на поставщиков. Еще одно звено в цепи - содержатели различных казино. Если их сразу не пристрелить, то через них можно получить много различной информации.

Управление

1. Интерфейс-2 (Синклер-джойстик)

2. Кемпстон-джойстик

3. Курсор-джойстик

4. Клавиатура

Управление клавиатурой

X – влево C – вправо

M – вниз L - вверх

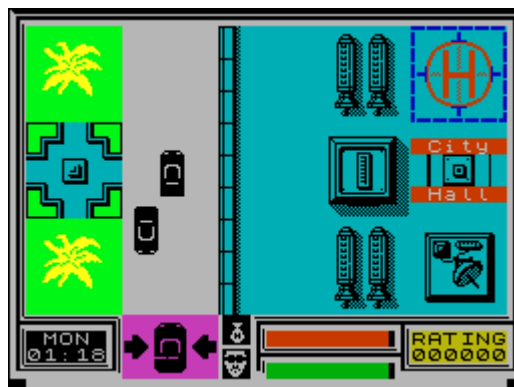
от Y до P – огонь

CAPS LOCK - остановка игры, запуск - повторным нажатием.

E - выход из игры (работает только после паузы).

Стрельба из окна машины.

Сначала нужно перейти в режим стрельбы. Это выполняется кнопкой "Огонь". При этом высвечивается автомобиль на дисплее. Чтобы выстрелить в указанном направлении нажмите "Огонь" еще раз.



Чтобы изменить направление стрельбы, пользуйтесь соответствующими клавишами.

Примечание: Пассажир не может стрелять из машины влево.

Вход в здание.

Остановите машину на дороге. Нажмите и отпустите "Огонь", теперь нажмите "Влево". Дисплей автомобиля изменится. На нем изображается символ автомобиля с двумя мигающими стрелками (справа и слева). Если в машине есть вещественные доказательства, то справа будет мигать стилизованное изображение мешка, а если в машине задержанный, то изображение человека. Чтобы выйти из машины и войти в помещение, нажмите "Огонь" в тот момент, когда стрелка на дисплее есть.

Чтобы не входить, нажмите "Огонь", когда стрелки на экране нет.

Примечание: Если в машине есть улики или задержанный и Вы выйдете из машины и войдете в здание, то они будут утрачены.

Управление внутри зданий.

Вверх - проход через дверь.

Вниз - режим "Выбор".

Влево, вправо - соответственно.

"Огонь" - стрельба в текущем направлении.

Улики (сумки с контрабандой) собираются простым перемещением на них.

При встрече с агентом, его имя загорается на экране сообщений. Чтобы задержать, его надо догнать. Иногда это удается сделать только после перестрелки (если он останется жив). Если же он сдается добровольно, то автоматически включается режим "Выбор" для заключения с ним сделки. Выбирайте тот или иной вариант из предлагаемых нажатием клавиши "Огонь".

Если допрос прошел успешно, Вы должны узнать имя или адрес явки, или время, или сумму сделки.

Чтобы получить очки, надо доставить арестованного гангстера в городское управление полиции (CITY HALL).

Специальные службы.

Они не имеют экранов, показывающих их внутреннее устройство, но компьютер издает характерный звук при посещении этих зданий.

Полицейское управление (CITY HALL).

Сюда доставляются арестованные и улики. Если машина с ними попадает в аварию, то они теряются.

Вы не получите очков за преступника, который откажется говорить, но получите за имевшееся при нем вещественное доказательство. Кроме того, после допроса в полицейском управлении Вам передадут его показания (примерно через 3 часа после доставки).

Госпиталь (HOSPITAL) - его можно посещать один раз в день для восстановления сил.

Расписание.

Ваш шеф снабдил Вас расписанием возможных появлений членов банды в местах явок, но оно действует только на понедельник.

Имеются 2 возможных варианта на первую половину дня и четыре варианта на вторую. Все встречи в течение 12 часов идут по одному расписанию, хотя заранее неизвестно, по какому именно.

Первая половина суток (A.M.)

Кто	Где	Когда	
		или	или
BLADES	SAMS BAR	2:00	9:20
PADDY	ISLAND BAR	3:20	8:00
MAC	FATS BAR	4:40	6:40
RONNIE	DIXIE BAR	6:00	5:20
BONZO	LAMP BAR	7:20	4:00
CHICO	SURFER BAR	8:40	2:40

Вторая половина суток (P.M.)

Кто	Где	Когда			
		или	или	или	или
DINO	JOES BAR	0:20	9:40	6:00	5:20
SHARKY	VINES BAR	1:20	8:40	7:00	4:20
TOOTS	SAMS BAR	2:20	7:40	8:00	3:20
HUGGY	ISLAND BAR	3:20	6:40	9:00	2:20
DUKE	FATS BAR	4:20	5:40	10:00	1:20
FRANKIE	DIXIE BAR	5:20	4:40	1:00	10:20
EDDIE	LAMP BAR	6:20	3:40	2:00	9:20
HAWKER	SURFER BAR	7:20	2:40	3:00	8:20
REEMO	JOES BAR	8:20	1:40	4:00	7:20
SNOWY	VINES BAR	9:20	0:40	5:00	6:20

Время

1 секунда реального времени = 1 минута игры.

Если Вы приедете на явку до того, как там появились члены банды, то они засекут Вашу машину у входа и успеют скрыться. Встреча будет сорвана.

Если Вы прибудете в течение 4-х минут после начала встречи, то они сумеют разбежаться, но оставят вещественные доказательства.

Если Вы прибудете от 4-х до 6-ми минут после назначенного времени, то сумеете захватить гангстера вместе с уликами.

Если Вы приедете от 8-ми до 12-ти минут после начала сбора, они успеют разойтись, но если Вы поспешите, то успеете "сесть на хвост" отъезжавшей с преступниками машины (она красного цвета).

После 12-ти минут будет слишком поздно.

Если Вы не сумеете схватить м-ра "J", то игра начнется сначала, с полуночи следующего воскресенья и до четверга. Все встречи членов банды будут идти по тому же расписанию, кроме тех, кто был арестован или убит.

Очки.

Игра заканчивается автоматически при задержании м-ра "J" или гибели Крокетта или Таббса.

Очки начисляются за:

- доставку улик в полицейское управление;
- доставку гангстеров, если они заговорили (но если это была не ложь);
- уничтожение автомобилей (красного цвета) с пытающимися скрыться преступниками. Очки снимаются за:

- аварию автомобиля;
- уничтожение автомобилей обычных граждан;
- за каждую рану Крокетта и Таббса;
- за неудачную попытку подкупа;
- за несостоявшийся арест, если Вы объявили преступнику, что он арестован.

Если Вы не задержите м-ра "J" к полудню четверга, то потеряете очки.

Запишу или обменяю программы к "ZX-Spectrum". В коллекции более 500 программ. Цены - низкие. Каталоги выгружу на ленту.
341048, УССР, Мариуполь - 48, а/я 48010, Александру.

Обмен и запись программ к Spectrum. Вышлю тематический каталог. К запросу приложить конверт с обратным адресом. 630102, Новосибирск, ул. Богаткова, д.50, кв.29. Ескевич Алекс Александрович.

Предлагаю программы и литературу по "Спектруму". Возможен обмен. В коллекции 500 программ. Каталог высылается бесплатно. Пишу программы на заказ для организаций и частных лиц.

Обращаться по адресу: 301650, Тульская обл., г. Узловая, ул. Завенягина, д.26, кв.43, Панарьину Артуру Васильевичу

Владельцам ПК-01 "ЛЪВІВ"

Если Вы хотите знать все секреты своего компьютера и быстро научиться программировать, приобретайте кассету с записью системных и игровых программ.

Вы найдете практические советы по программированию и защите программ.

Содержание:

- все о ПК " ЛЪВІВ";
- универсальный отладчик;
- АССЕМБЛЕР;
- ДИСАССЕМБЛЕР;
- редактор текстов;
- графический редактор;
- монитор;
- копировщик программ;
- BS монитор;
- ОЗУ-V2;
- RENUM (расширение БЕЙСИКа);
- шахматы;
- шашки;
- и другие системные и игровые программы.

Всего 170 рублей, и Вы будете обладать большими возможностями.

Оплату за кассету выполнять почтовым переводом по адресу:

41651С, г. Ахтубинск-1, ул. Волгоградская, д. 14, кв. 12.

Мызину Вячеславу Николаевичу.

STRATEGIC GAMES

PRESIDENT

ADDICTIVE 1986

Автор игры - Кевин Томс. Ему же принадлежит и ставшая очень популярной менеджерская игра FOOTBALL MANAGER.

Такая игра, как PRESIDENT - единственное средство проверить Ваши способности по управлению современным государством в одиночку. Вам предстоит принимать все ответственные решения: как распределить продукты питания, где производить поиск нефти, как и когда наносить военные удары по соседним государствам.

Игра как бы разделена на множество разделов, в каждом из которых Вы принимаете решение посредством выбора из альтернативных вариантов. О том, какой эффект имеет Ваше решение, Вы узнаете в конце месяца, когда Вам сообщат как относятся к Вашему правлению мировое сообщество и основные политические партии Вашей страны.

OPINION POLLS	
YOUR PARTY	50%
MODERATE PARTY	25%
EXTREMIST PARTY	25%
<FIRE> TO CONTINUE	

NEXT ELECTION IN 23 MTHS	
INCOME	
OIL CONTRACTS	= 0
SPOT MARKET	= 0
GOLD SALES	= 0
TOTAL	0
EXPENDITURE	
IMPORT COSTS	= 35
HEALTH COSTS	= 30
GOLD PURCHASES	= 0
TOTAL	65
BAL. OF PAYMNTS = K\$	-65
<FIRE> TO CONTINUE	

Каждые два года (через двадцать четыре месяца) проводятся новые президентские выборы. Конечно, ловкий игрок будет принимать самые крутые и непопулярные решения сразу после очередных выборов, а в оставшийся срок правления станет завоевывать голоса и доверие избирателей специально для этого рассчитанными взвешенными решениями.

В управление Вам досталась довольно типичная средиземноморская страна, в процветании которой решающую роль играет нефть. На экране изображена карта Вашей страны с нанесенными на нее символами танковых дивизий, нефтеперерабатывающих заводов, нефтехранилищ, посевов зерновых и т.п.



Там же показаны транспортные пути. Вы можете искать нефть и ставить нефтяные вышки, прокладывать дороги и регулировать транспортировку продуктов и товаров.

Время от времени беспокойные соседи будут наносить по Вам воздушные удары, имея в виду в первую очередь поражение нефтяных полей. Интенсивность этих налетов можно снизить тщательно продуманной стратегией размещения зенитных батарей. Могут вражеские силы проводить и наземные атаки, в этом случае Вам предстоит управление своими танковыми частями.

IMPORTS		K\$	VALUE
OIL PRODUCTION EQUIP.	=	15	
MILITARY HARDWARE	=	0	
FOOD	=	20	
TOTAL VALUE	=	35	
CURRENCY VALUE INDEX	=	99%	
TOTAL COST			
$100 \times (\text{TOTALVALUE}) / (\text{C.V. IND.})$		=	35
BANK BALANCE	=	K\$	5965
<FIRE> TO CONTINUE			

OIL SUPPLY CONTRACT	
COUNTRY:	DENMARK
THIS CONTRACT REQUIRES THE	
SUPPLY OF	1 KBARRELS
PER MTH	
AT A UNIT PRICE OF K\$	57
LENGTH OF CONTRACT	= 3 MTHS
N.B. FAILURE TO SUPPLY IN	
FULL WILL RESULT IN IMMEDIATE	
CONTRACT TERMINATION	
SPOT MARKET PRICE	= K\$ 50
KBARRELS REMAINING AFTER	
CONTRACT SALES	= 2
DO YOU ACCEPT?	YES NO

У Вас есть масса возможностей проиграть. При слабой армии вражеские самолеты разнесут заводы и погубят урожай. При чрезмерных затратах на армию население начнет погибать от малярии и туберкулеза, а укреплять армию можно, например, импортом товаров военного назначения.

Это весьма сложная игра. Графика, правда, не потрясает, но вопрос, насколько графика нужна стратегическим играм - достаточно спорный, а в отношении стратегического менеджмента эта игра не может не вызывать восхищения.

FALKLANDS 82

PSS 1986



Сюжет игры относится к Англо-Аргентинскому кризису 1982 года. Эти события еще совсем свежи в памяти большинства наших читателей, и мы останавливаться на них не будем. Скажем только, что операция, начиная с высадки в Сан-Карлосе и до захвата Порты Стенли заняла 25 дней. Аналогично и Вам предстоит повторить это достижение за 25-30 игровых ходов (в зависимости от избранного уровня сложности).

Экран воспроизводит карту северо-восточной части Фолклендских островов. Ваши боевые части показаны в виде элементарных блоков. Достаточно простыми графическими средствами показаны горы и поселки. Аргентинские части на карте не показаны - они скрыты, и Вы их увидите только когда столкнетесь.

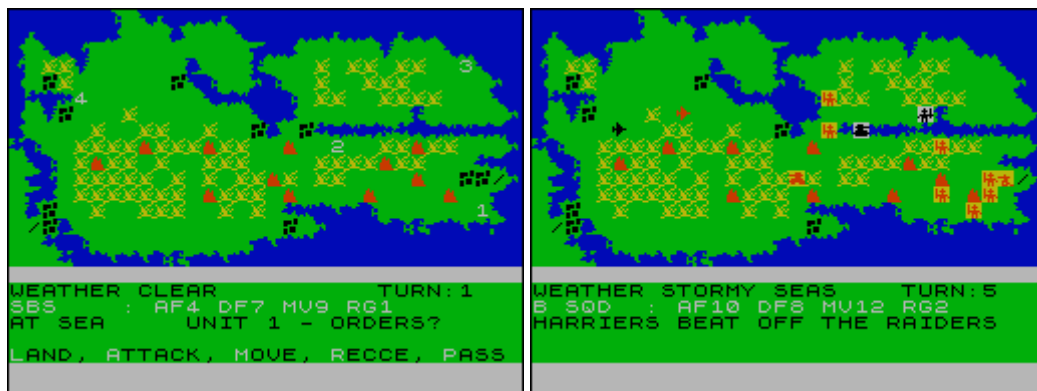
Правила игры очень несложны. Каждая часть имеет 4 показателя:

- фактор атаки (AF);
- фактор защиты (DF);
- скорость перемещения (MV),
- дальность действия (RG).

Фактор атаки характеризует силу боевой части.

Фактор защиты - примерно то же, но на него оказывает влияние рельеф местности, впрочем он же влияет и на скорость перемещения.

Дальность действия для артиллерии - 6 полей, для танков и моточастей - 2, для пехоты - 1.



Вы можете использовать поддержку своей авиации морского базирования, а также огневую поддержку корабельных орудий. В самом начале игры Вы должны назначать сколько-то кораблей для этой задачи, но если назначите слишком много, а аргентинцы атакуют головной флот, то самолеты не поддержат наземные операции, т.к. будут отражать морские атаки.

Поддержка с моря и с воздуха зависит от погоды, а погода с каждым днем становится все хуже и хуже. Поэтому постарайтесь разгромить все, что можно, как можно раньше. Вызов поддержки выполняется вместо хода Вашего полка.

Поскольку на каждом квадрате игрового поля не может быть более одного полка, возможны пробки. Надо аккуратно планировать атаку.

Цель игры не состоит в разгроме аргентинских частей. Вам надо захватить как можно больше населенных пунктов. Если потеряете время в бесконечных стычках, то можете не успеть выполнить главную задачу.

Игра идеальна для начинающих своей простотой. К недостаткам можно отнести не очень хорошую сбалансированность. На высоком уровне сложности аргентинцы сильны настолько, что это решительно не соответствует реальности. На самом же низком уровне играть так легко, что неинтересно.

AUSTERLITZ LOTHLORIEN1986.



Игра разработана для версии компьютера с оперативной памятью 128К и относится к одной из лучших стратегических игр этой фирмы.

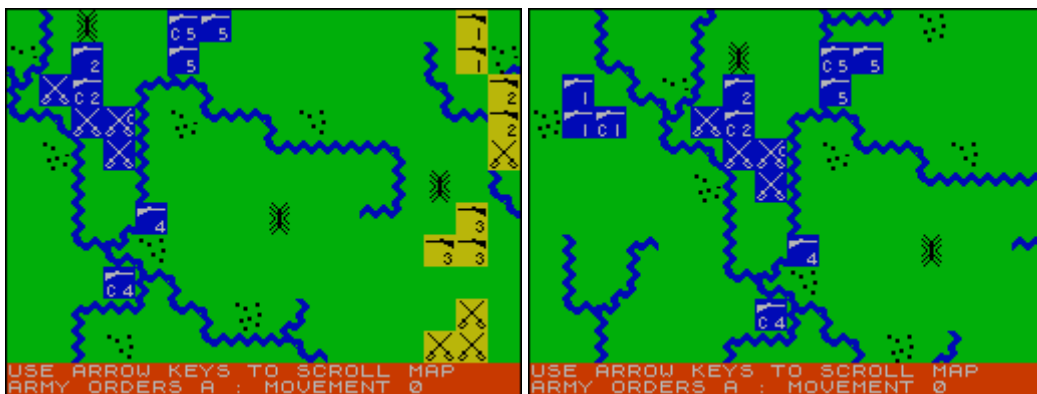
В основу игры положена битва при Аустерлице, в которой Наполеон разгромил значительно превосходящие его объединенные войска Австрии и союзной с ней России. Сражение произошло в 1805 году.

Успех был достигнут тремя факторами. Во-первых, прекрасная обученность войск и, соответственно, высокий моральный фактор.

Во-вторых, стратегический гений Наполеона, талантливое манипулирование войсками.

И, наконец, в третьих, более опытные генералы, по сравнению с войсками союзников. Тактические решения по ходу сражения принимались на месте, оперативно и безошибочно.

Ваша задача - повторить успех Наполеона как в стратегическом, так и в тактическом плане. На тот случай, что вам это удастся, в игре предусмотрены повышенные уровни сложности.



В игре есть две характерные черты, отличающие ее от прочих игр этого жанра.

Во-первых, предполагается, что командиры Ваших частей хорошо подготовлены и во многих случаях решения по отдельным деталям боевых ситуаций принимается и без Вас. Можете даже полностью передать им управление войсками.

Во-вторых, для максимального приближения к реальности существуют скрытые перемещения войск противника. Так, по мере перемещения частей Вы можете терять их из виду и вновь обнаруживать только в результате прямого столкновения.

Механика игры несложна и соответствует большинству игр этого жанра. На результативность боевых ударов оказывают большое влияние фактор местности и фактор морального состояния войска. Местность, во-первых, определяет на какое расстояние то или иное подразделение может быть передвинуто за один ход и, во-вторых, влияет на силу подразделения, находящегося в обороне. Понятно, что горный хребет оборонять проще, чем заболоченную долину.

Моральный фактор учитывается так:

Боевая мощь частей рассчитывается из условия 500 человек - одна единица силы. Тогда часть в составе 4500 человек имеет 9 единиц. При максимальном моральном факторе к ее силе прибавляется еще 5 единиц, что эквивалентно увеличению численности как бы на 3000 человек.

"Аустерлиц" - великолепная игра. Возможно, Наполеон не одержал бы той блистательной победы, если бы его противники промоделировали сражение на "Спектрe".

Техника управления игрой чрезвычайно похожа на технику игры "OVERLORD" той же фирмы, в которой Вы решаете задачу высадки союзных войск в Франции в 1944 году и которую мы очень подробно разберем в следующем выпуске "ZX-РЕВЮ".

* * *

Тем, кто хочет попробовать свои силы в адвентюрных программах!

По просьбам желающих приобщиться к работе с программами этого жанра мы подготовили два сборника программ. Не можем правда сказать, чтобы эти просьбы были многочисленными, но зато они соответствуют нашему основному направлению на расширение интереса к "умным" играм.

Среди программ, представленных в этих сборниках, есть и классические образцы, есть и шедевры, есть и несложные программы, удобные для начинающих.

Считайте, что это как бы маленькая ретроспектива жанра, конечно, здесь не охвачена и десятая часть того, что было разработано, но здесь охвачены основные направления, представлены ведущие фирмы, а если окажется, что это кому-то интересно, то мы ведь можем и продолжить такие сборники.

Если вам удастся пройти какую-нибудь игру от начала и до конца, черкните нам об этом записку, мы будем использовать Ваш опыт для помощи тем, кто "застрянет" по дороге, эти игры проходятся только коллективно.

Цена сборников и порядок оплаты Вам известны, поскольку каждый зарегистрированный получатель "ZX-РЕВЮ" получает вместе с ним ежемесячно и информационный листок "Информком предлагает..."

ADVENTURE-1

1. ZZZZ	MASTERTRONIC	1987
2. Robin of Sherlock-1	SILVERSOFT	1985
3. Robin of Sherlock-2	SILVERSOFT	1985
4. Robin of Sherlock-3	SILVERSOFT	1985
5. Red Moon	LEVEL 9	1985
6. Heavy on the Magic	GARGOYLE	1986
7. Bored of the Rings	DELTA 4	1985
8. Sherlock	MELBOURNE HOUSE	1984
9. Gremlins	ADVENTURE INT.	1983
10. Slane King	MARTECH	1987
11. Rebel Planet	U.S.GOLD	1986
12. He Man	ADVENTURE SOFT	1986
13. Witch's Cauldron	MIKROGEN	1987
14. Planet of Death	ARTIC	1982
15. Inca Curse	ARTIC	1982
16. Ship of Doom	ARTIC	1982
17. Espionage Island	ARTIC	1982
18. Neverending Story 1	OCEAN	1985
19. Neverending Story 2	OCEAN	1985
20. Neverending Story 3	OCEAN	1985
21. Aftershock	INTERCEPTOR	1986
22. Wulfan	MASTERTRONIC	1988

ADVENTURE-2

1. Hunchback 1	OCEAN	1986
2. Hunchback 2	OCEAN	1986
3. Hunchback 3	OCEAN	1986
4. Red Hawk	MELBOURNE HOUSE	1986
5. Sorderon's Shadow	BEYOND	1985
6. Stormbringer	MASTERTRONIC	1987
7. Vera Cruz 1	INFOGRAMES	1985
6. Vera Cruz 2	INFOGRAMES	1985
9. Sidney Affair	INFOGRAMES	1987
10. Master of Magic	MASTERTRONIC	1987
11. Hampstead	MELBOURNE HOUSE	1986
12. Spellbound	BEYOND	1984
13. Doomdark's Revenge	BEYOND	1985
14. Worm in Paradise - 1 (с графикой)	LEVEL 9	1985
15. Worm in Paradise - 2 (то же, но без графики, с расширенным текстом)	LEVEL 9	1985
16. Heartland	ODIN	1987
17. Sam Cruise	MICROSPHERE	1987

ЦЕНА ИДЕИ

Многие из читателей ZX-РЕВЮ в свободное от основной работы время работают над какими-то разработками, связанными со своим компьютером. Рано или поздно наступит момент, когда они предложат свою работу потребительскому рынку, а может быть коммерческая организация обратится к ним с заказом на исполнение работы, проведение исследования и т. п.

Как себя повести в этой ситуации, как установить цену на свою разработку, как получить оплату - сразу или по частям?

Этим вопросам мы посвящаем статью "Цена идеи". Страна входит в рынок. Мало кто знает, что это такое, и что будет с обычными товарами в рыночной экономике, но совсем никто не знает что будет с ценообразованием на идеи, не имеющие аналогов, на научные разработки, которые нельзя взвесить и пощупать.

Сразу оговоримся, что материал, который мы здесь даем, уникальный. Еще не пришло время его проверить практикой, и потому мы не претендуем ни на какую методичность, а просто постараемся простыми словами изложить те проблемы, которые надо решать и наметим основные принципы их решения.

Как бы мы ни старались упростить изложение, все равно без некоторых экономических терминов нам не обойтись.

Итак, предположим, Вы стали автором идеи (научной, или программной, или методической разработки, а может быть и технической, например Вы разработали принципиальную схему и описание периферийного устройства).

Товар Ваша идея или не товар?

Еще недавно она не была товаром, а если и товаром, то не Вашим. Она принадлежала той организации, в которой Вы родили эту идею, а еще точнее - никому, поскольку все прекрасно знают о муках изобретателей и рационализаторов. Они зачем-то что-то изобретают, чтобы отрывать от дела важных людей и портить им кровь своими просьбами. А уж о программных или о методических разработках мы и не говорим - раз нет изделия в металле, значит и нет предмета изобретения, а значит и вообще ничего нет.

Никто не пробовал запатентовать методику (мы имеем в виду у нас, а не за рубежом)?

Сейчас можно определенно сказать, что Ваши идеи тот же товар, и он может быть продан и оплачен, но как?

Если бы Вы производили обувь, все было бы просто. Взяли стоимость материалов, прибавили трудозатраты, налоги, накладные расходы, ожидаемую прибыль - вот Вам и цена. Это расчет от себестоимости, здесь он совершенно неприменим.

Вопросы продажи идеи надо решать с других позиций. Продажа идеи - это своего рода продажа лицензии и если Вы сделали программу и предлагаете ее фирме, которая берется ее распространять, то Вы становитесь ЛИЦЕНЗИАРОм, а фирма, приобретающая у Вас право на распространение - ЛИЦЕНЗИАТОм и вот на особенностях их взаимоотношений, мы и остановимся. Суть не меняется, если Вы сами распространяете свою программу. Все равно - всякий купивший ее у Вас покупает тем самым лицензию на ее применение. А цена на лицензию определяется никак не от себестоимости, возьмем для примера "Кока-Колу". Изобретена почти сто лет назад, затраты на ее изобретение окупались уже миллион раз - а попробуйте купить лицензию, что это будет стоить?

Основной вопрос, который надо решить при определении цены, заключается не в том, сколько получит лицензиар за свою продукцию, а в том насколько выгодно ее применение лицензиату.

Иными словами, проблема состоит не в том, сколько стоит идея, а в том, сколько за нее готовы заплатить. (Иногда к оценке стоимости идеи пытаются подойти с точки зрения затрат лицензиара на ее разработку или возможных затрат лицензиата на создание альтернативной идеи. Такой подход нельзя признать продуктивным. Заинтересованность лицензиата в покупке идеи проистекает не из затрат лицензиара, а из выгоды, которую эта сделка принесет ему). Поэтому определение размера компенсации наиболее

целесообразно производить исходя из доли лицензиата в прибыли лицензиара, т.е. оплата лицензии должна производиться исходя из предполагаемого конечного результата ее использования.

Ценообразование в этой области осложняется еще и тем, что идеи могут еще не иметь адекватного материального выражения. Практическое использование их, как правило, требует финансовых вложений. Кроме того, будущая рентабельность неизвестна, да и вообще проблематична. Как ни однозначна ситуация с использованием идей, рыночная ситуация еще более непонятна, спрос на новый продукт может быть загадкой, размеры рынка неизвестны. Тем не менее, несмотря на данные трудности, продавец и покупатель должны достичь соглашения, установив приемлемую для обоих цену.

Итак, коль скоро мы установили, что размер компенсации за эксплуатацию идеи (разработки) должен зависеть от прибыли покупателя, то разумно и стоимость приобретения лицензии (разработки) определять исходя из доли лицензиара в будущей прибыли лицензиата и выражать ее в виде процента от прибыли.

Этот процент от прибыли называется СТАВКОЙ РОЯЛТИ. Обычно этот процент составляет от 0,5 до 15% реальной прибыли, а от чего он зависит мы рассмотрим немного позже.

Конечно получать оплату за свою разработку в виде процента от прибыли не очень удобно. Прежде всего Вы рискуете нарваться на организацию, которая не приложит сил к продвижению Вашей идеи и Вы ничего не получите, зато они, затормозив Вашу разработку, приложат максимум усилий к продвижению своих, устранив тем самым опасность, что Вы продадите свою идею ближайшим конкурентам.

Рискуете Вы также тем, что Ваша разработка как бы гениальна она не была, может просто не пойти на рынке, хотя бы потому, что рынок не готов к ее приему. Тогда лицензиар потерпит убытки, а Вы не получите за свою идею ни копейки.

Гораздо удобнее разработчику получить компенсацию за идею сразу и целиком, не дожидаясь прибыли. Конечно, в этом случае общий размер суммы должен быть существенно ниже, хотя бы за исключение риска не получить ничего, но и не только за это. Об этом мы тоже поговорим позже.

Такая единовременная выплата при приобретении вашей идеи называется ПАУШАЛЬНОЙ СУММОЙ.

Конечно, договор о продаже может быть комбинированным и предполагать некоторую единовременную ПАУШАЛЬНУЮ сумму и последующие отчисления роялти.

Дело не в том, в какой форме будет выплачиваться вознаграждение, суть в том, что при любой форме единственный способ прикинуть каким-то образом размер этой ПАУШАЛЬНОЙ СУММЫ (или ее первого взноса) можно тоже только на основе ставки РОЯЛТИ и расчеты и в том и в другом случае проводятся от определения этой ставки.

Применение в договоре и той и другой формы оплаты имеют свои достоинства и недостатки. Перечислим их хотя бы кратко, Вы конечно понимаете, что достоинства с точки зрения лицензиара могут быть недостатками с точки зрения лицензиата и наоборот. Чтобы как-то согласовать их интересы мы будем считать достоинствами те факторы, которые способствуют справедливому заключению договора, а недостатками те, которые ему препятствуют. Тем самым мы приводим противоположные интересы сторон к общему для них интересу - заключению справедливого контракта.

Преимущества оплаты лицензии паушальной суммой (а также оплаты в течение непродолжительного времени) следующие:

- цена разработки известна заранее;
- предотвращается инспектирование лицензиаром коммерческих операций лицензиата (а ведь они могут содержать коммерческую тайну);
- непредвиденное увеличение цен не повысит размер выплат;
- цены различных идей становятся более сопоставимыми, т.к. отпадает необходимость оценки возможного объема и времени производства и продаж;
- паушальная сумма, как правило, бывает ниже приведенных роялти, т.к. все бремя

риска берет на себя лицензиат.

Но есть и недостатки:

- оплата производится до получения результата;
- лицензиар оказывается не заинтересованным в успехе лицензиата, т.к. уже получил всю причитающуюся ему сумму заранее; лицензиар может продать аналогичную лицензию конкуренту по другой цене, что приведет к снижению прибыли лицензиата, а сама мысль об этом может сорвать заключение договора.

Соответственно, преимущества текущей ставки роялти заключаются в следующем:

- бремя риска распределяется между лицензиаром и лицензиатом;
- снижая ставку роялти, лицензиар может помочь лицензиату захватить большую часть рынка;
- если лицензиар не выполняет каких-то обязательств, выплата роялти может быть приостановлена. Прекращение платежа также наступит при ликвидации лицензиатом контракта;
- текущие ставки облегчают финансово-кредитное положение лицензиата;
- возможна дифференциация ставок роялти в зависимости от места (продажа в своем регионе или в соседнем) и времени продаж;
- текущая ставка в любой момент может быть с согласия лицензиара преобразована в паушальную сумму;
- ставки роялти могут быть изменены при обнаружении недостатков в приобретенной разработке.

Недостатки текущих ставок следующие:

- повышение цен в результате региональной инфляции или регулирующей политики властей приводит к увеличению выплат лицензиару без каких-либо усилий с его стороны;
- доход лицензиара за период контракта невозможно определить с необходимой точностью.

Установление приемлемой цены.

На этот вопрос нет легкого ответа. Соглашение должно обеспечить адекватную награду лицензиару за его творческий потенциал, усилия и затраты. Но цена не должна быть настолько высокой, чтобы не быть не выгодной, не угрожать получению приемлемой прибыли лицензиатом.

Разработчики новых технологий, авторы идей часто считают возможным запрашивать за них довольно высокие цены, т.к. они считают, что если бы не их усилия, то этих идей или технологий, а следовательно, и возможностей их применения не было бы вообще.

Лицензиаты, наоборот, считают, что т.к. на них по сравнению с изобретателем (автором идеи) ложится основное финансовое бремя, т.е. и риск некупаемости вложенных средств, то они имеют право на большую долю дохода. Обе позиции обоснованы, но могут быть причиной одинаковых ошибок, являющихся результатом переоценки своего положения.

Необходимость, а не жадность является основной силой, направляющей заключение сделки, однако этот принцип, хотя и является важным, еще не приводит к установлению приемлемой для обеих сторон цены. Необходим учет многих других факторов.

Факторы определения цен.

Стоимость новой идеи, научной разработки в наиболее полной степени определяется предполагаемым доходом, который может получить владелец в течение их жизненного цикла. Этот доход в свою очередь определяется рыночным спросом на данную разработку и уровнем роялти, уплачиваемой лицензиару в течение всего срока лицензионного соглашения. Поэтому вопрос об уровне роялти является очень важным. Обычно ставки роялти составляют от 0.5 до 15 процентов от прибыли с продажи данной разработки. Оценить ставку роялти в этих пределах можно исходя из девяти перечисленных ниже

факторов, в той или иной мере присутствующих во всех сделках по продаже научных разработок.

Каждый из этих факторов влияет на коммерческую привлекательность разработки для потенциальных лицензиатов и поэтому должен оказывать влияние на ставку роялти. В каждом факторе мы постарались учесть только одну из значимых переменных, характеризующих коммерческую выгодность идеи.

В следующем выпуске мы рассмотрим эти факторы, приведем конкретные практические примеры из нашей жизни, разберем возможные расчетные формулы и дадим ряд практических советов для успешного продвижения Ваших идей и разработок.

А пока желаем Вам успешной работы над Вашими проектами.

ИНФОРКОМ

(Окончание в следующем выпуске)

ИНФОРКОМ

Начал выпуск обучающих программ для IBM совместимых персональных компьютеров

Просим откликнуться все организации, имеющие такие машины и заинтересованные в получении более подробной информации. Основные особенности наших программ:

- имеют очень широкие сферы применения;
- могут быть использованы немедленно после поставки (не требуют подготовки);
- эффективность применения проверена практикой;
- относительно невысокие цены.

Банк разработок непрерывно пополняется.

программы поставляются на высококачественных дискетах фирмы Kodak с защитным тефлоновым покрытием.

Программы разработаны с использованием инструментальных программных средств, приобретенных с официальной лицензией.

Посредническим организациям предоставляем лицензию на право перепродажи и значительные оптовые скидки.

107241 Москва, Б-241, а/я 37 "ИНФОРКОМ"



НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ИНФОРКОМ". 107241, МОСКВА Б-241 а/я 37

СПЕКТРУМ В ШКОЛЕ

Много программ написано для игры на "Спектруме". К Вашим услугам самые разнообразные игровые жанры. Вы можете тренировать свою реакцию, изобретательность, исследовательские способности.

В этой статье мы и коснемся вопроса исследований. Очень многим решение головоломок, научная и исследовательская работа дают не меньшее удовольствие, чем самая азартная игра. А если Вы еще не пробовали свои силы в исследованиях, попробуйте. Хуже Вам от этого не будет. Опыт пригодится, а там кто знает, может быть Вы станете и автором открытия.

Задолго до того, как были изобретены ЭВМ, математики выполняли свои расчеты на бумаге. Появление таблиц логарифмов и других справочных таблиц сильно упростило их работу, но вывод новых математических соотношений по-прежнему оставался крайне утомительным делом. Например, вычисление числа "пи" с точностью до 200-300 знаков занимало несколько месяцев кропотливой работы. Точно то же и с простыми числами. Отыскание вручную семизначного числа может занять несколько недель расчетов.

Простое число - это число, которое делится нацело только само на себя или на единицу. Вы знаете, конечно, что к простым относятся числа

1, 3, 5, 7, 11, 13, 17, 19, 23...

А вот попробуйте вручную с помощью карандаша и бумаги определить, является ли число 1333 простым или нет? И заодно засекайте время, которое на это потребуется. А потом попробуйте то же самое узнать с помощью приведенной ниже программы. С ее помощью Вы можете проверить числа до 4 294 967 295 (предел работы "Спектрума" с целыми числами). Если проверяемое число не является простым, то программа выдаст его простые множители.

Если Вам удастся найти простое число, в котором есть несколько нечетных цифр, то попробуйте поменять местами эти нечетные знаки и посмотреть, не будет ли простым число, полученное в результате таких перестановок. Исследуйте, сколько времени занимает проверка числа и как оно зависит от длины этого числа.

А вот еще одна проблема для исследования. Взяв число, являющееся квадратом целого числа, попробуйте получить его в виде суммы двух простых чисел. Например: $25 = 23 + 2$. А вот число 36 может быть получено даже тремя способами:

$$5 + 31 = 36$$

$$7 + 29 = 36$$

$$17 + 19 = 36$$

Может быть для больших чисел можно найти другой рекорд?

Еще вопрос. А можно ли суммой трех простых чисел образовать квадрат целого числа и всегда ли?

Развлечения с числами вдохновляли математиков много столетий. Можно сказать, что задача выведения формулы для генерации простых чисел заняла больше человеко-часов работы, чем любая другая математическая проблема. Даже и сегодня формулы для вычисления всех простых чисел нет. Появляются все новые и новые формулы, но находятся простые числа, которые в них не укладываются.

Великий Эйлер когда-то предложил формулу $X^2 + X + 41$, но к сожалению, среди тех

чисел, которые она дает, есть не только простые числа.

Другой хорошо известной формулой стала формула генерации чисел Мерсена (по имени французского монаха, занимавшегося этой проблемой в 17 веке). Эта формула: $2^p - 1$, где p - в свою очередь тоже простое число.

Самое интересное, что число $2^{67} - 1$ более двухсот лет считалось простым, пока в 1903 году один американский профессор не показал, что оно делится на 193707721 и на 761836257287.

После 1952 года к поискам и проверкам чисел Мерсена подключились компьютеры.

1952 г. - $2^{2281} - 1$

1962 г. - $2^{4423} - 1$

1968 г. - $2^{11213} - 1$

1971 г. - $2^{19937} - 1$

Метод, которым ищутся простые числа, остался неизменным почти 2000 лет и носит название "решето Эратосфена" по имени Эратосфена из Александрии. Метод прост и заключается в следующем:

Записывается ряд положительных чисел. Затем слева направо в нем вычеркивается каждое второе число, затем каждое третье, каждое четвертое и т.д. Таким образом, все числа, которые могут быть представлены в виде произведения других, выпадают, а оставшиеся и являются простыми числами.

С простыми числами можно сделать еще ряд интересных наблюдений, например то, что первые пять цифр после запятой числа "пи" образуют простое число.

3.14159... - 14159

Проверьте и убедитесь.

Проверьте числа, образованные одинаковыми цифрами, например

111, 1111, 11111, 7777 и т.п.

Какие из них являются простыми? И в заключение три проблемы:

1) сколько полиндромов, являющихся простыми числами Вы можете найти? Полиндромами называют числа, которые слева направо и справа налево читаются одинаково, например:

3156513

2) Используя каждую цифру 1, 2, 3, 4, 5, 6, 7, 8, 9 и только по одному разу образуйте три простых числа, имеющих минимальную сумму. Например, простые числа 941, 653 и 827 дают в сумме 2421, но это не минимальная сумма.

3) Не сможете ли Вы с помощью "Спектрума" составить магический квадрат, состоящий только из простых чисел. Напоминаем, что в магическом квадрате сумма цифр по всем вертикалям, горизонталям и диагоналям одинакова.

* * *

В предлагаемой программе мы все текстовые сообщения, стоящие в кавычках и в строках REM дали на русском языке. Вы, наверное сами разберетесь, чем их заменить, если в Вашем компьютере нет русского шрифта или вообще пропустите. На итог работы это никак не повлияет.

```
10 PRINT "Программа для проверки простых чисел"
```

```
20 PRINT "Введите Ваше число" ' ' "Для прекращения работы введите <s>" ' ' "Нажмите любую клавишу"
```

```
40 DIM L(400)
```

```
60 REM *** Блок ввода ***
```

```
70 INPUT B$: CLS
```

```
80 INPUT "Введите число"; N$
```

```
90 IF N$="s" OR N$="S" THEN STOP
```

```
100 LET number = VAL N$
```

```
110 IF number - INT (number)<>0 THEN PRINT "Число не целое": GO TO 80
```

```
130 REM ***Блок инициализации ***
```

```
140 IF number=2 OR number=3 THEN GO TO 290
```

```
150 LET index=1: LET temp=number: LET flag=0
```

```
160 REM ***Главный блок ***
```

```
170 IF 2*INT(temp/2)=temp THEN
```

```
LET L(index)=2:
```

```

    LET flag=1:
    LET index=index+1:
    LET temp=INT(temp/2):
    GO TO 170
180 LET factor = 3
190 IF factor*INT(temp/factor)=temp THEN
    LET L(index)=factor:
    LET index=index+1:
    LET flag=1:
    LET temp=INT(temp/factor):
    GO TO 190
210 IF factor^2>temp AND flag=0 THEN GO TO 290
220 IF factor^2>temp AND flag=1 THEN
    LET L(index)=temp:
    LET index=index+1:
    GO TO 245
230 LET factor=factor+2: GO TO 190
240 REM *** Блок печати ***
245 PRINT number; " не простое число" ' ' "Его множители:-" ' '
250 FOR j=1 TO index-1
260 PRINT L(j)
270 NEXT j
280 PRINT: GO TO 80
290 PRINT " ";number;" простое число": PRINT: GO TO 80

```

BETA BASIC

Мы начинаем печатать инструктивных материалов по работе с языком программирования БЕТА-БЕЙСИК, разработанным и выпущенным фирмой BETASOFT.

На сегодняшний день уже выпущено несколько версий этого языка. Наиболее распространенной является версия 3.0, но мы начнем описание языка с версии 1.0 (хотя она никем и не применяется) потому, что в версиях этого языка сохранена преемственность, т.е. каждая новая версия добавляет какие-то новые команды и начинать сразу с высоких версий не очень удобно в плане стройности изложения материала. Вы же, со своей стороны, должны знать, что все возможности старых версий сохранены в новых, а на дополнения мы укажем по мере перехода к их рассмотрению.

Вступление.

Бета-Бейсик (1.0) по сравнению со стандартным Бейсиком "Спектрума" имеет 26 новых ключевых слов и 10 новых функций. Кроме того, некоторые из существовавших команд были существенно улучшены. Введены также ряд возможностей повышающих удобство программирования, например, мигающий указатель строки и BREAK для процедур, записанных в машинных кодах.

Существуют две версии этого выпуска. Одна для 48К, другая для 16К. (Версия 16К будет работать и на 48К компьютере, но большая часть программы останется неиспользованной).

Загрузите программу:

LOAD "bb16" или "bb48".

По команде LOAD загрузятся и автостартуют две строки Бейсика (строка 0 и строка 1). Строка 0 содержит определение функции Бета-Бейсика, а строка 1 резервирует память для основной части Бета-Бейсика, которая выполнена в машинных кодах и имеет длину 5.3К, а затем ее загружает.

По команде CLEAR понижается RAMTOP до 27168 или до 59936, в зависимости от применяемой Вами версии. Загрузка закончена, когда на экране появляется сообщение "копирайт". Если это не так, значит произошла ошибка ввода попробуйте еще раз.

Строка 1 сама себя уничтожит после окончания загрузки, а строка 0 останется. Другим указанием на то, что Бета-Бейсик успешно загрузился, является мигающий указатель текущей строки. При нажатии клавиши воспроизводится звуковой сигнал. Если он Вам не нужен, то дайте команду POKE 23609,0.

Строка 0 - не проявляется по команде LIST, если в компьютере присутствуют другие строки Бейсик-программы. Теперь Вы можете загрузить (MERGE) любую Бейсик-программу. Нельзя давать LOAD, а только MERGE, если Вы хотите, чтобы новые функции работали. Имейте в виду, что вместо графики пользователя (UDG) будут воспроизводиться новые ключевые слова Бета-Бейсика. Чтобы вернуть графику UDG, Вам надо будет дать команду KEYWORDS 0 (см. ниже).

При работе Вы почувствуете некоторое увеличение скорости вычислений, особенно на длинных программах, в которых есть много операторов GO TO и GO SUB.

Команда RETURN также выполняется намного быстрее.

Команда NEW вычищает всю программу, кроме строки 0, и одновременно выполняет CLEAR.

После того, как Вы напишете программу, содержащую новые ключевые слова, Вы можете выгрузить ее на ленту обычным образом. Когда Вам понадобится ее загрузить, загрузите сначала Бета-Бейсик (если он уже не присутствует в компьютере), а затем загружайте Вашу программу. При выгрузке программы в нее вошла и строка 0, поэтому теперь Вам нет необходимости загружаться через MERGE, достаточно дать LOAD.

Если Вы забыли предварительно загрузить Бета-Бейсик, то новые ключевые слова будут изображены в виде одиночных символов и появится сообщение "NONSENSE IN BASIC".

Чтобы это исправить, дайте:

```
MERGE "bb48": GO TO 1.
```

Здесь GO TO 1 нужен, чтобы обеспечить автостарт, т.к. нормальный автостарт отменяется загрузкой через "MERGE".

Все новые ключевые слова, кроме EDIT берутся через графический режим (CAPS SHIFT+9). В графическом режиме клавиши дают эти ключевые слова, а не символы графики пользователя. Графический режим действует на одно нажатие и автоматически снимается. Правила синтаксиса для новых ключевых слов приведены ниже. То, что заключено между знаками "// //" (двойная косая черта), может быть при вводе опущено. При вводе, как и обычно, выполняется проверка синтаксиса. Если встречается ошибка, то может быть выдано одно из стандартных сообщений "Спектрума", а может быть выдано сообщение Бета-Бейсика.

В нижеприведенных примерах мы старались не давать более чем по одному новому ключевому слову на один пример. Это нужно для лучшего усвоения, но зато многие наши примеры лишены возможной элегантности. Относитесь к этому с пониманием.

РАСКЛАДКА КЛАВИАТУРЫ									
DEF PROC	PROC	END PROC	RENUM		AUTO	DELETE	KEYWORDS		EDIT
1	2	3	4	5	6	7	8	9	0
POP		ELSE	ROLL	TRACE		USING	EXIT IF	ON	DPOKE
Q	W	E	R	T	Y	U	I	O	P
ALTER	SCROLL	DO		GET		WHILE	UNTIL	LOOP	
A	S	D	F	G	H	J	K	L	
		CLOCK			ON ERROR	SORT			
	Z	X	C	V	B	N	M		

1. Команда ALTER.

Структура:

ALTER //описание атрибутов// TO описание атрибутов.

Клавиша: A.

Эта команда позволяет выполнять обширные манипуляции с атрибутами экрана, а именно с INK, PAPER, FLASH и BRIGHT для каждой символьной позиции. В своей простейшей форме команда ALTER может выполнить изменение цвета экрана INK или PAPER по всему экрану без (!!!) очищения экрана:

```
100 PRINT AT 10,10;"TEST":PAUSE 50:ALTER TO PAPER 1
```

По этой команде свет фона изменится на синий. Вы можете воспользоваться также комбинацией атрибутов:

```
ALTER TO PAPER 2, INK 7,FLASH 1
```

Можно сделать так, чтобы программа различала какие атрибуты заменять, а какие нет:

```
ALTER INK 1 TO INK 7
```

По этой команде все, что написано синим цветом изменится на белый. Теперь можете выполнять графические изображения цветом INK, совпадающим с цветом PAPER так, что они на экране будут не видны, а потом с помощью команды ALTER "проявлять" изображение.

Возможны и комплексные построения типа:

```
ALTER INK 3, BRIGHT 1, PAPER 7 TO INK 5, FLASH 1
```

Например:

```
100 LET A=2: LET B=4  
110 FOR L=1 TO 5
```

```

120 FOR N=1 TO 16: PRINT INK A;PAPER B; "XXXX"; PAPER A; INK B; "0000": NEXT N
130 LET temp = A: LET A=B: LET B=temp: NEXT L
140 LET t=30
150 ALTER INK A TO INK B: PAUSE t
160 ALTER PAPER A TO INK A: PAUSE t
170 ALTER IKK A TO PAPER B: PAUSE t
180 ALTER INK B TO PAPER A: PAUSE t
190 LET t=t-t/10+1: GO TO 150

```

2. Команда: AUTO

Структура:

AUTO //номер строки// //,величина шага//

Клавиша: 6

Команда AUTO включает автоматическую нумерацию строк, что сильно облегчает ввод программ с клавиатуры. Если параметр величины шага опущен, то он принимается равным 10. Если и номер строки опущен, то в качестве него принимается текущий номер программной строки плюс десять. Этот режим отключается по команде AUTO 3, или если номер строки превысит 9984, или после любого системного сообщения. Если Вы хотите в режиме AUTO выпустить какой-то блок номеров строк, то просто сотрите предложенный Вам номер и вместо него наберите нужный Вам. Дальнейшая автонумерация продолжится с тем же шагом, что и ранее.

Примеры:

AUTO - действует от текущей строки +10, шаг = 10

AUTO 100 - от строки 100, шаг равен 10

AUTO 100,5 - от строки 100, шаг = 5.

3. Команда BREAK

Клавиши: CAPS SHIFT + SPACE (не в графическом режиме)

BREAK - это не ключевое слово.

Обычный BREAK, который поддерживается стандартным Бейсиком, вполне годится для многих целей, хотя те, кто занимаются программированием в машинных кодах, испытывают неудобства, когда их процедура оказывается замкнутой в бесконечном цикле, который нельзя прервать через BREAK и программа "зависает". Если Вы работаете в Бета-Бейсике и пользуетесь при этом функцией CLOCK (см. ниже), то Вы можете организовать систему для поддержки BREAK в машинных кодах. Так, например, если Вы нажмете CAPS SHIFT + SPACE на время большее 1 сек, то поддерживающая система решит, что Вы "зависли" и прервет работу, даже если программа работала правильно.

Замечание для программистов в машинных кодах: Если Вы хотите пользоваться этой функцией BREAK, не отключайте прерывания (команда ACCEМБЛЕРа DI - DISABLE INTERRUPTS).

4. Команда CLOCK

Структура:

CLOCK число

или

CLOCK строка

Клавиша: C

Команда CLOCK управляет работой таймера, который действует от системы прерываний компьютера. CLOCK может изображать в правом верхнем углу текущее время в часах, минутах и секундах. Можно сделать так, что при достижении заданного времени компьютер будет издавать звуковой сигнал (будильник) или будет выполняться переход к какой-либо подпрограмме GO SUB. Часы работают и в то время, когда Вы пишете программу, и во время ее прогона (RUN). Они не работают только во время подачи звукового сигнала (BEEP) и во время операций с магнитной лентой. Чтобы указать какие именно возможности используются, после CLOCK нужно поставить число в диапазоне от 0 до 7, которое можно выбрать по нижеприведенной таблице:

Число	Переход по GO SUB	Включение звукового сигнала	Изображение времени на экране
0	-	-	-
1	-	-	+
2	-	+	-
3	-	+	+
4	+	-	-
5	+	-	+
6	+	+	-
7	+	+	+

Часы начнут работать сразу после того, как один из этих режимов будет задан, начиная с "00:00:00". Дайте CLOCK 1, чтобы увидеть время. Чтобы установить правильное время, надо после CLOCK дать строку, например:

```
CLOCK "09:29:55"
```

В принципе, здесь двоеточие можно и не давать. Если строка содержит менее чем 6 цифр, все остальные полагаются равными нулю. Так,

```
CLOCK "xyz 10" дает "10:00:00".
```

Если в строке первым символом стоит "а" или "А", то остальные числа воспринимаются как данные для установки звукового сигнала.

CLOCK "A06:20" выполнит установку будильника на двадцать минут седьмого. В установленное время прозвучит звуковой сигнал, если он был взведен после CLOCK числом 2,3,6 или 7 (см. таблицу выше).

Можно, если Вы задали число 4,5,6 или 7, в назначенное время выполнить переход к какой-либо подпрограмме GO SUB, но это возможно только при прогоне программы. Если Вы сидите и работаете над составлением программы, это произойти не может.

По достижении установленного времени оператор CLOCK ждет окончания вычисления последней строки, что может занять некоторое время, особенно если это INPUT или PAUSE и только после этого переходит к указанной подпрограмме. Никакого GO SUB из программы, написанной в машинных кодах здесь выполняться не может. Номер строки, к которой выполняется переход, должен стоять после CLOCK и должен быть в диапазоне от 8 до 9999. В подпрограмме надо использовать имена переменных, отличных от имен в главной программе, иначе эти переменные могут подвергнуться несанкционированным изменениям.

Если Вам нужно, чтобы подпрограмма хранила данные, а в программе есть CLEAR или RUN, то эти данные нужно перенести предварительно в соответствующую область памяти через POKE.

Возможные применения включают в себя различные вариации на тему будильника - включение мелодии, изображения на экране, отбивание часов и т.п. (Текущее время можно взять в качестве функции и использовать в подпрограмме - см. FN T\$). Энтузиасты от электроники могут сделать так, что компьютер будет принимать данные через внешний порт, скажем один раз в час или в минуту и что-нибудь с ними делать. Пример:

```
8909 STOP
9010 LET pointer=PEEK 37000: POKE pointer+27001, IN 127
9020 LET pointer=pointer+1: IF pointer>100 THEN LET pointer = 0
9030 POKE 27000,pointer: LET Z$: FN T$( )
9040 LET hours = VAL Z$(1 TO 2): LET mins = VAL Z$(4 TO 5)
9050 LET mins = mins+1: IF mins = 60 THEN LET hours = hours+1:LET mins =0
9060 CLOCK "A"+STR$ hours+STR$ mins: RETURN
```

Не делайте RUN. Вместо этого дайте прямую команду:

```
CLEAR 26999: POKE 27000,0: CLOCK 9000: CLOCK 5
```

Установите "будильник" на какое-то время, чуть попозже текущего.

```
CLOCK "A xxxx"
```

Теперь запустите (RUN) какую-либо программу. Процедура CLOCK активизируется

каждую минуту: по ней считывается значение с порта 127 и отправляется на хранение в адрес, на который указывает непрерывно увеличивающийся указатель (он сбрасывается на нуль после превышении 100).

Если у Вас нет устройств, управляемых через порты, воспользуйтесь строками 9040-9060 или какими-либо другими, которые будут что-либо делать через регулярные интервалы. Эти строки взводят "будильник" на одну минуту после текущего времени, прежде чем возвращаться в главную программу.

5. Команда: DEF PROC

Структура: DEF PROC имя процедуры

Клавиша 1 (та же, что и DEF FN).

См. также PROC, END PROC

По этой команде задается имя процедуры. DEF PROC должен быть первым ключевым словом в строке (хотя наличие пробелов или определений цвета разрешается).

Использованное имя процедуры должно соответствовать правилам для имен переменных - первый символ должен быть буквой, а за ним строка букв или цифр или знак \$. Пробелы не учитываются и буквы верхнего и нижнего регистра эквивалентны. Процедура может иметь имя, совпадающее с именем переменной: путаницы не произойдет.

6. Команда: DELETE

Структура: DELETE // номер строки// TO // номер строки//

Клавиша: 7

По этой команде удаляются все строки в указанном блоке программы. Если начальный номер строки опущен, то по умолчанию принимается номер строки, следующей за нулевой. Если опущен конечный номер, то принимается номер последующей строки программы.

DELETE TO 100 - удаляются все строки, начиная со строки, следующей за нулевой до строки 100 включительно.

DELETE 100 TO - удаляется строка 100 и все последующие.

DELETE 100 TO 100 - удаляется только строка 100

DELETE 0 TO 0 - удаляется только нулевая строка.

DELETE TO - удаляется вся программа, кроме нулевой строки.

Последний из примеров не эквивалентен NEW, т.к. в этом случае не вычищаются переменные. Удаляются только строки программной области PROG, но не переменные из области VARS.

Любой из указанных номеров должен реально существовать, иначе будет сообщение об ошибке:

U: NO SUCH LINE

(такой строки нет).

DELETE можно включать в текст программы, хотя с определенной предусмотрительностью. Так, например, если DELETE входит в состав процедуры, подпрограммы или цикла и уничтожает строки, расположенные выше, то программа остановится, так как адреса перестанут соответствовать положению в программе. Если DELETE должен уничтожить сам себя, то он должен быть последней строкой удаляемого блока.

Одно из возможных практических приложений состоит в очистке данных DATA после их использования через READ, чтобы освободить память для переменных (каждое число в DATA занимает по меньшей мере 8 байтов).

7. Команда: DO

Структура:

DO

или

DO WHILE условие

или

DO UNTIL условие

Клавиша: D

WHILE - клавиша J

UNTIL - клавиша K

См. также LOOP, EXIT IF

Команды DO и LOOP работают совместно и служат для организации циклов. Совместно со своими квалификаторами WHILE и UNTIL они обеспечивают управляющую структуру, значительно превосходящую возможности стандартного Бейсика. Сама по себе команда DO служит как бы маркером, которому передает управление соответствующий оператор LOOP. Пример:

```
10 DO
20 PRINT "HELLO";: PAUSE 50
30 LOOP
```

По этой программе будет печататься слово "HELLO" до тех пор, пока Вы не нажмете BREAK.

Цикл DO может управляться через WHILE и некоторое условие. Если указанное условие справедливо, то все строки между DO и LOOP будут выполняться, т.е. цикл работает, пока условие выполняется. Как только условие станет ложным, то управление сразу будет передано оператору, следующему за LOOP.

DO UNTIL - имеет прямо противоположное значение. Цикл DO LOOP выполняется, пока условие при операторе UNTIL ложно, то есть до тех пор, пока оно не станет справедливым.

```
10 LET total = 0
20 DO UNTIL total>100
30 INPUT "ENTER A NUMBER";x
40 LET total=total+x: PRINT total
50 LOOP
60 PRINT "THAT IS OVER A HUNDRED"
```

Строку 20 можно заменить строкой:

```
20 DO WHILE total <=100
```

Пары DO LOOP можно вкладывать друг в друга так же, как это делается для циклов FOR-NEXT. (в нижележащем примере мы наметили буквами пары для наглядности, в программе же этого делать нельзя).

```
DO(A):DO (B):LOOP (B):LOOP (A)
```

Здесь один цикл (B) внутри другого (A).

```
DO(A):DO(B):LOOP(B):DO(C): LOOP(C):LOOP(A)
```

Здесь два цикла (B) и (C) внутри цикла (A).

Поскольку адреса цикла хранятся в памяти, "выйти" из середины цикла нельзя иначе, как с помощью операторов EXIT IF или POP, иначе нарушится работа стека.

Если в Вашей программе цикл DO не закрыт соответствующим оператором LOOP, компьютер выдаст сообщение:

```
S "MISSING LOOP"
```

(S "Отсутствует LOOP").

Управляющие структуры, очень похожие на DO-LOOP существуют и в некоторых других алгоритмических языках и имеют различные названия. Например:

REPEAT(выражение) UNTIL(условие)

эквивалентно:

DO(выражение) LOOP UNTIL(условие)

и

WHILE (Условие) (выражение) ENDWHILE (или WEND)

эквивалентно:

DO WHILE(условие)(выражение) LOOP

8. Команда: DPOKE

Структура:

DPOKE адрес, число

Клавиша: P

См. также FN P (адрес).

DPOKE означает двойной POKE. Ему эквивалентна следующая конструкция обычного Бейсика:

POKE адрес, число - INT (число/256)*256

POKE адрес+1,INT(число/256)

Другими словами, младший байт двухбайтного числа засылается (POKE) в определенный адрес, а старший байт - в следующий адрес. Таким образом хранятся коды от 0 до 65535. Многие системные переменные имеют такой двухбайтный формат и работа с ними через DPOKE значительно упрощается.

Эквивалентный двойной PEEK выполняется через FN P (адрес).

9. Команда EDIT.

Структура:

EDIT // номер строки//

Клавиша: 0 (не в графическом режиме!)

Оператор EDIT не то же самое, что стандартное CAPS SHIFT + 1.

Это ключевое слово. Его цель состоит в отказе от утомительного процесса: LIST - BREAK - CAPS SHIFT + 1

Чтобы быть действительно усовершенствованием, EDIT должен быть ключевым словом, берущимся без регистра, но все буквенные клавиши уже заняты, а цифровые нужны для нумерации строк. Однако, поскольку обычно номер строки не начинается с нуля, то его можно использовать для этих целей. Ключевое слово EDIT вызывается, если клавиша 0 нажимается после предшествовавшего нажатия ENTER.

Если Вы дадите EDIT+ номер строки + ENTER, то эта строка немедленно появляется, готовая к редактированию. Если номер строки опущен, то Вам будет подана текущая строка. Процесс редактирования был еще более упрощен путем задействования курсорных клавиш "вверх" и "вниз" для редактирования строки. Они позволяют мгновенно перемещать курсор в начало или в конец строки.

Другой способ вызова EDIT - клавиша 5 в графическом режиме.

10. Команда ELSE

Структура: ELSE (выражение)

Клавиша: E

ELSE - это часть конструкции IF-THEN. Обычно, когда значение выражения, стоящего после IF, является ложным, вычисления передаются строке, следующей за оператором THEN. Однако, если конструкция IF-THEN имеет связанный с нею оператор ELSE, то управление передается на оператор, следующий за ELSE.

С другой стороны, если условие, стоящее после IF является справедливым, то все действия выполняются только до ELSE, а затем управление передается на строку, следующую за строкой, содержащей ELSE.

Например:

```
10 INPUT "Give me a number";X
20 PRINT "Does that number = 1?"
30 PAUSE 50
40 IF X = 1 THEN PRINT "True": ELSE PRINT "False"
50 GO TO 10
```

Заметьте, что перед оператором ELSE всегда должно стоять двоеточие. Ситуация усложняется, когда у Вас есть несколько IF и ELSE в одной строке. Примеры ниже показывают как сочетаются различные ELSE со своими IF. Буквенные метки в скобках служат только для наглядности.

IF-THEN (A):IF-THEN (B):ELSE (B)

IF-THEN (A):IF-THEN (B):ELSE (B):ELSE (A)

IF-THEN (A):ELSE (A):IF THEN (B):ELSE (B)

11. Команда: END PROC

Клавиша: 3

См. также DEF PROC, PROC

Этим оператором отмечается конец именованной процедуры, что позволяет компьютеру избежать вычислений между ее началом DEF PROC и концом END PROC, если процедура не была вызвана командой PROC.

Когда процедура вызвана, то END PROC отмечает место конца вычислений и управление передается строке, стоящей после PROC. (Более подробно об этом см. PROC).

Использование END PROC, без предварительного задания DEF PROC вызывает сообщение об ошибке

W: "Missing DEF PROC"

(W: "Отсутствует DEF PROC")

12. Команда: EXIT IF

Структура: EXIT IF (условие)

Клавиша: i

См. также DO, P LOOP EXIT IF - это часть структуры цикла DO-LOOP (сначала прочитайте об этом). EXIT IF применяется, когда надо выйти из середины цикла до его нормального завершения. Если установленное условие справедливо, то выполняется переход к следующему за LOOP, в противном случае ничего не происходит.

```
100 DO: PRINT "Line 100":PAUSE 20
110 EXIT IF INKEY$="STOP"
120 PRINT "LINE 120":PAUSE 20:LOOP
130 PRINT "Out of loop"
```

Вы будете находиться в цикле, пока не нажмете STOP (SYMBOL SHIFT + A). Заметьте, что текст строки 120 не печатается, когда выполняется выход из цикла. Если в программе будет отсутствовать закрывающий оператор LOOP, то Вы получите сообщение об ошибке:

S: "Missing LOOP"

(S: "отсутствует LOOP")

13. Команда: GET

Структура:

GET числовая переменная

GET строковая переменная

Клавиша: G

Так же, как и INKEY\$, GET позволяет читать клавиатуру без использования ENTER.

Разница между GET и INKEY\$ состоит в том, что GET ждет нажатия клавиши прежде, чем продолжить работу. Когда GET применяется со строковой переменной, он "снимает" символ.

```
10 GET A$: PRINT A$: GO TO 10
```

Теперь Ваш компьютер работает как пишущая машинка. Можете переключать регистры обычным порядком. CAPS SHIFT + 5 дает шаг назад без стирания, а ENTER эквивалентен PRINT. Более изощренная версия выглядит так:

```
10 GET A$: PRINT A$: FLASH 1;"B"; FLASH 0; " ";CHR$8;CHR$8;: GOTO 10
```

Если GET используется с числовой переменной, т.е. GET X или GET клавиша, то переменной присваивается 1, если будет нажата клавиша "1" и так далее до 9. Если будет нажата "A", то 10, если "B", то 11 и т.д.

Особенно полезен оператор GET в программах, где управление выполняется от меню. См. также ON.

14. Команда: KEYWORDS

Структура:

KEYWORDS 1 или

KEYWORDS 0

Клавиша: 8

Эта команда позволяет делать выбор, что печатать - ключевые слова Бета-Бейсика или символы графики пользователя UDG.

KEYWORDS 0 - символы UDG

KEYWORDS 1 - ключевые слова Бета-Бейсика.

Несмотря на то, что распечатка Вашей программы будет выглядеть очень сомнительно, работать все будет нормально. Единственное ключевое слово, которое нельзя будет заменить графикой - это само слово "KEYWORDS". Да это собственно и не нужно, потому что графический символ на клавише 8 эквивалентен пробелу.

15. Команда: LOOP

Структура:

LOOP WHILE условие или

LOOP UNTIL условие

Клавиша: L (UNTIL - клавиша K; WHILE - клавиша J)

См. также DO, EXIT IF

LOOP - это часть структуры циклов DO-LOOP - см. сначала DO. Сам по себе оператор LOOP заставляет программу возвращаться к соответствующему DO. Квалификаторы WHILE (пока) и UNTIL (до тех пор) используются для того, чтобы возврат от LOOP к DO был условным.

LOOP WHILE условие - повторяет цикл только если условие справедливо.

LOOP UNTIL условие - наоборот, если условие не выполняется.

Если Вы используете окончание цикла (LOOP без его начала DO), то получите сообщение об ошибке:

Т: "LOOP WITHOUT DO"
(LOOP без DO)

16. Команда: ON

Структура:

GO TO ON число; номер строки, номер строки, номер строки....

или

GO SUB ON число; номер строки, номер строки, номер строки....

Клавиша: O (буква "O")

Оператор ON позволяет выполнять переход к конкретному номеру строки из списка в зависимости от величины выражения, непосредственно стоящего за ON.

В принципе это возможно делать и в стандартном Бейсике, например, так:

```
10 INPUT a: GO TO a*100+100,
```

но здесь номера строк перехода являются членами одной закономерной последовательности. А если они произвольные, то:

```
10 INPUT a: GO TO ON a; 90, 135, 60, 40
```

```
20 PRINT "ENTER 1 TO 4": GO TO 10
```

В этом примере к строке 90 будет выполнен переход, если a=1; если a=2, то к строке 135 и т.д.

Если "a" не принадлежит диапазону от 1 до 4, то переход не состоится, управление будет передано следующей строке, которая вернет Вас назад и даст еще раз попробовать. INPUT здесь может быть заменен на GET, чтобы более элегантно организовать меню в программе.

17. Команда: ON ERROR

Структура:

ON ERROR номер строки

Клавиша: N

Эта команда требует в случае появления ошибки перехода к строке, номер которой задан в команде. Такой переход возможен после любого сообщения об ошибке как от стандартного Бейсика, так и от Бета-Бейсика. Исключение составляют сообщения: 0, "OK" и 9, "Stop statement", которые конечно же ошибками не являются. Отключается такой режим

подачей ON ERROR 0. Кроме того, он отключается во время работы подпрограммы, которой было передано управление в результате ошибки, но вновь включается после возврата в главную программу.

В подпрограмме могут быть использованы три переменные: line, stat и error.

line - номер строки, в которой произошла ошибка;

stat - номер выражения в строке;

error - код ошибки.

Все это не ключевые слова, поэтому их надо набирать по буквам.

Ниже приведен пример фрагмента программы, который печатает точки. Подпрограмма работает так, что если точка выходит за пределы экрана, то программа ее пропускает.

```
100 ON ERROR 5000
110 FOR n=1 TO 10: INPUT "x coord ";x;"y coord ";y
120 PLOT x,y :NEXT n
4990 STOP
5000 IF ERROR = 11 AND line = 120 THEN RETURN: ELSE POP: CONTINUE
```

Примечания:

Команда STOP предназначена для того, чтобы предотвратить случайный вход в подпрограмму обработки ошибок. Номера строки и ошибки проверяются потому, что "INTEGER OUT OF RANGE" - довольно часто встречающаяся ошибка.

Возврат в главную программу из строки 5000 будет происходить в точку, следующую за тем местом, где произошла ошибка, поэтому RETURN будет выполняться к NEXT n. Если же номер ошибки не тот, который был оговорен, будет выполнен оператор CONTINUE.

Команда POP удаляет со стека адрес возврата в главную программу, иначе стек может оказаться запертым.

Есть одна "ошибка" с которой надо работать несколько иначе. Она связана с использованием команды BREAK и называется "BREAK INTO PROGRAM" (прерывание программы). Т.к. функция ON ERROR отключена во время работы подпрограммы обработки ошибок, то результатом нажатия BREAK будет то, что программа остановится после первого же выражения подпрограммы, т.к. клавишу BREAK Вы еще не успели отпустить. Поэтому, если Вы хотите, чтобы после BREAK Ваша подпрограмма выполнила какие-то действия, то необходимо ввести какую-то задержку в первом же выражении в Вашей подпрограмме. За время этой задержки Вы успеете отпустить BREAK. В качестве задержки подойдет BEEP, причем Вы можете использовать частоту звукового сигнала, которая почти не слышна.

```
100 ON ERROR 5000
110 PRINT "round and";:PAUSE 10: GO TO 110
4990 STOP
5000 IF ERROR = 21 THEN BEEP 1,69: BORDER RND*7: RETURN: ELSE POP: CONTINUE
```

Если Вы забудете ограничить подпрограмму обработки ошибок каким-либо конкретным типом ошибки, то можете зависнуть в бесконечном цикле. Например, что делать, если Вы не сможете выйти из "STOP in input". В этом случае Вам может помочь система, поддерживающая BREAK в Бета-Бейсике, иначе можете выключать компьютер.

18. Команда: PLOT

Структура:

PLOT X координата, Y координата //: строка//

Клавиша: обычная клавиша для функции PLOT (не в графическом режиме!)

Как Вы уже увидели из структуры команды, оператор PLOT в Бета-Бейсике имеет дело не только с включением и выключением пикселей, но и со строковыми данными.

Координаты выражения при PLOT указывают на левый верхний угол первого символа строки. Строки не должны превышать 32 символа. Если они длиннее, то напечатаны не будут. Могут быть использованы INVERSE, OVER, INK и т.д. Строки не должны содержать CHR\$ 13 (управляющий код ENTER), иначе некоторые из них не смогут быть напечатаны. Если строка выходит за пределы правого поля экрана, то она "возвращается" с левой стороны. Если она выходит за пределы экрана сверху или снизу, то будет выдано сообщение "INTEGER OUT OF RANGE". Позиция точки, используемая для операции DRAW в качестве

стартовой не изменяется при помещении строки на экран.

Изменяя координаты тех точек, куда Вы будете помещать (PLOT) символы, Вы сможете добиться более гладкого перемещения, чем при использовании традиционного "PRINT AT".

```
100 FOR X=16 TO 224: PLOT X,X/2;"<>": NEXT X
```

Попробуйте в эту строку добавить STEP 2, или STEP 3 для получения различных эффектов. Поскольку "<>" имеет бордюр вокруг символа цвета PAPER шириной по крайней мере в один пиксел, то он автоматически будет стирать сам себя по мере перемещения на один пиксел за один шаг. Некоторые буквы, например "Т" имеют пикселя цвета "INK", достигающие краев элементарного символьного квадрата 8x8. Это означает, что при движении в некоторых направлениях они оставляют "след", если Вы не примете специальных мер для его стирания.

Если Вы будете конструировать свой собственный шрифт, то очень неплохо оставлять у символов бордюр в один пиксель со всех сторон.

Возможность выполнять PLOT для строк применяется для нанесения меток на диаграммы и графики, т.к. в этом случае Вы можете пользоваться общей координатной системой для построения графика и для нанесения надписей на нем.

Нижеприведенный пример показывает возможность выполнения подстрочных и надстрочных индексов на "Спектруме".

```
100 LET A$="DO you know how many molecules/cm+3-Na-2+PO-4+has?" (Знаете ли Вы сколько  
молекул/куб-см имеет Na2PO4 ?)
```

```
110 LET X = 0: LET Y = 160  
120 FOR C=1 TO LEN A$  
130 IF A$ (C) = "-" THEN LET Y=Y-3: NEXT C  
140 IF A$ (C) = "+" THEN LET Y=Y+3: NEXT C  
150 PLOT X,Y: A$(C): LET X=X+8  
160 IF X>=248 THEN LET X=0: LET Y=Y-12  
170 NEXT C
```

В программе знаки "+" и "-" играют роль непечатных управляющих кодов, которые сдвигают позицию печати вниз или вверх на три пикселя.

Поэкспериментируйте со строкой 150 с тем, чтобы X возрастало больше или меньше, чем на 8 пикселей. Можете при этом попробовать PLOT OVER 1.

Если среди Вас есть энтузиасты, попробуйте использовать PLOT для того, чтобы обеспечить выравнивание строк по правому полю. Вам потребуется отыскать пробелы между словами и увеличить их или уменьшить так, чтобы строка содержала точно целое число слов.

19. Команда POP

Структура:

POP //числовая переменная//

Клавиша: Q

По этой команде устраняется адрес, находящийся на вершине программного стека. Это тот стек, который обслуживает GO SUB, DO-LOOP, PROC-ENDPROC (не путать с машинным стеком и со стеком калькулятора).

Указанный номер строки присваивается переменной, если она есть при POP. POP позволяет выходить из подпрограмм или циклов без их естественного завершения. Для этого POP снимает ставший ненужным адрес возврата со стека и предотвращает "зависание" программы. Просто POP уничтожает этот адрес, а POP с переменной, например "а" присваивает ей это значение. Тогда процедура или подпрограмма может "запомнить" откуда ее вызывали. Если Вы впоследствии пожалеете о том, что воспользовались POP, то сможете исправить положение: GO TO а+1

Хотя это и не эквивалентно RETURN, т.к. через GO TO можно войти только в первый оператор строки.

```
100 GO SUB 500  
110 STOP  
500 POP а
```

```
510 PRINT "Subroutine was called from line "; a (Подпрограмма вызывалась из строки "a").  
520 GO TO a+1
```

Если Вы замените строку 520 на 520 RETURN, то получите: "RETURN Without GO SUB", т.к. требуемый адрес возврата на стеке уже находится.

Использование POP, когда на стеке уже нет данных, дает сообщение об ошибке:

```
V: NO POP data
```

(Продолжение в следующем выпуске)

Маленькие хитрости

Продолжаем начатый в прошлом выпуске разговор об использовании управляющих кодов в операторе PRINT с целью существенной экономии памяти как в программной области, так и в области программных переменных.

Вы можете таким же образом использовать символ CHR\$6 в начале строки, в конце, можете давать их подряд, чтобы в тексте сообщения образовались пустые строки (хотя для этого есть и другие "хитрости").

Аналогичным же образом в текстовые строки вставляют и цветовые атрибуты.

Возьмем строку:

```
10 PRINT INK 1;PAPER 6; BRIGHT 1; FLASH 1; "TEST".
```

Она занимает 48 байтов, из которых 31 можно сэкономить (в нашем справочнике по программированию в машинных кодах, т.3 трехтомника, мы приводим таблицу всех символов "Спектрума", в том числе и управляющих кодов). Действуйте так:

```
10 PRINT
  перейти в режим E
  нажать CAPS SHIFT
  нажать 1
  перейти в режим E
  нажать 6
  перейти в режим E
  нажать 9
  перейти в режим E
  нажать CAPS SHIFT
  нажать 9
  набрать "TEST"
  нажать ENTER
```

Когда Вы все это сделаете, ярко-синее слово TEST будет изображено на желтом фоне и будет при этом мигать.

Правда, после этого будут точно так же мигать теми же цветами и все последующие строки листинга. Можете на это не обращать внимания, поскольку на работу программы после RUN это никак не влияет, но если все-таки Вас это раздражает, то можете погасить в листинге:

```
  перейти в режим E
  нажать 8
  перейти в режим E
  нажать CAPS SHIFT
  нажать 8
  перейти в режим E
  нажать 7
  перейти в режим E
  нажать CAPS SHIFT
  нажать 0
```

Но это будет стоить Вам еще 8 байтов. С другой стороны, эти цветовые проблемы не возникли бы, если бы Вы строку держали бы только в области VARS. И, напоследок, еще один прием. Вы можете напечатать один символ поверх другого. Часто это бывает полезно, если Вы хотите сделать текст на экране подчеркнутый.

Для этого могут применяться управляющие коды CHR\$8 - BACKSPACE и CHR\$21 - OVER. Попробуйте:

```
LET a$="A"+CHR$8*CHR$21+"_"
```

Дайте команду PRINT a\$ и Вы увидите на экране подчеркнутую букву "а".

Поэкспериментируйте с компьютером, посмотрите как в оператор PRINT входят цветовые коды после перехода в режим к и нажатия цифровой клавиши (по разному с CAPS

SHIFT и без).

Еще один путь экономии памяти состоит в использовании токенов ключевых слов, правда это нельзя сделать с русскоязычными текстами, поскольку все токены - это английские слова. Тем не менее, интересен сам прием.

Например, в Вашей программе есть такое сообщение:

```
20 PRINT "NEW DATA"
```

Оно может использоваться, например, в Вашей базе данных. Занимает такая запись 16 байтов, но если вместо слов NEW и DATA использовать токены NEW и DATA, то расход можно уменьшить до 10 байтов, поскольку всякий токен, даже такой длинный как RANDOMIZE занимает только один байт.

Однако, если Вы попытаетесь набрать такую строку с помощью токенов, то у Вас ничего не получится. Как только Вы откроете первую кавычку, на экране возникнет курсор L и машина будет ждать от Вас ввода букв. Это ведь и естественно. Внутри кавычек ведь пишется текст, набираемый по буквам, а никак не операторы БЕЙСИКа. Вам надо обхитрить компьютер, чтобы он все-таки выдал Вам курсор K. Это можно сделать так:

```
20 PRINT "THEN NEW THEN DATA"
```

- а теперь удалите операторы THEN стиранием.

Если Вам хочется отдохнуть после напряженной работы с компьютером, попробуйте поэкспериментировать, какой длины Вы сможете составить предложение, используя только токены и знаки пунктуации. Вот удивительно глупый пример:

```
RUN VAL, COS LEN FLASH!
```

Теперь продолжим наши эксперименты со строками. Предположим, что Вам надо дать сообщение: "Вы находитесь в коридоре, ведущем ", а затем варианты:

```
"на север";  
"на восток";  
"на юг";  
"на запад";  
"вверх";  
"вниз".
```

Вам надо сделать 6 различных сообщений. Можно задать эти сообщения так:

LET a\$="Вы находитесь в коридоре, ведущем ", а затем:

```
10 PRINT a$; "на север"
```

Не забудьте только в a\$ после слова "ведущем" оставить один пробел.

Тексты значительно проще читаются, если между строками на экране оставить пустые строки. Это можно сделать, если применять CHR\$6 (запятую оператора PRINT), о чем мы писали в прошлом выпуске, но можно вставить и символ CHR\$13 (ENTER).

Попробуйте, например:

```
LET a$=CHR$13+"ZX Spectrum" + CHR$13  
PRINT "Test 1";a$;"Test 2"
```

Вы увидите, что фразы отделены пробельными строками.

Очень часто в своих программах Вы под главным окном программы делаете небольшое окно для коммуникации с пользователем. Это означает, что Вам приходится использовать строки типа:

```
10 PRINT AT 20,4;"Подумайте и ответьте:"
```

Конечно, если эта фраза используется очень часто, то ее можно присвоить переменной, что даст определенную экономию, но еще больше можно сэкономить с помощью CHR\$22.

CHR\$22 - управляющий код AT (управляет позицией печати) тоже можно вставить в программных переменных VARS, как и любой другой управляющий символ. Вот пример такой строки:

```
LET a$=CHR$22+CHR$20+CHR$4+"Подумайте и ответьте:"
```

Если теперь дать PRINT a\$, то это сообщение появится в двадцатой строке, начиная с четвертого столбца.

Если же Вам надо много разных сообщений печатать в одном месте экрана, то можно управляющие коды выделить в отдельную строковую переменную:

```
LET p$=CHR$22+CHR$20+CHR$4,
```

а затем применять например так:

```
10 PRINT p$;"Ваше сообщение..."
```

Конечно и само сообщение тоже можно выразить переменной. Если же Вы хотите, чтобы все Ваши сообщения печатались в заданном месте в заданном цвете, то и цветовые атрибуты можно задать управляющими кодами и "пристегнуть" их к печатаемой строке.

Напечатав сообщение в заданном месте, Вы может быть захотите его стереть. Обычно это делают так:

```
10 PRINT AT 20,0;" (32 пробела)"
```

Эти пробелы - расточительный расход памяти. Можно было бы сделать так:

```
10 PRINT AT 20,0, ,
```

В конце здесь стоит не кавычка - это две запятые (CHR\$6).

Еще лучше этот пример запишется:

```
LET d$=CHR$22+CHR$20+CHR$0 + ' ' (2 запятые оператора PRINT)",
```

а затем давать PRINT d\$, когда надо стереть сообщение.

О том, как вставить "запятые PRINT" (CHR\$6) внутрь текстовых кавычек, мы с Вами говорили ранее.

Используя серию символов "запятая PRINT" Вы можете стирать несколько строк, но для этого можно использовать также оператор INPUT AT.

Количество строк, подлежащих очистке, отсчитывается снизу вверх, а текущая позиция печати должна быть над областью, подлежащей очистке, иначе экран начнет скроллить. Например, чтобы очистить четыре нижние строки. Попробуйте:

```
10 PRINT AT 17,0;:INPUT AT 4,0;
```

Кстати, если Вы применяли печать сообщения оператором

```
PRINT #0;"сообщение..."
```

и теперь имеете трудности с его очисткой, это можно сделать просто:

```
INPUT;!
```

Все то, что мы говорили о символах управления позицией печати CHR\$22, применимо и к управляющему символу CHR\$23 (TAB). Он управляет величиной табуляции. Но надо помнить, что TAB влияет на весь ряд, в том числе и на атрибуты, и результат может быть очень странным.

Теперь перейдем к "хитростям", связанным с элементарной графикой. Кое-что интересное при очень малом расходе памяти можно получить, если использовать самые обычные символы в сочетании с управляющими кодами.

Вот небольшой пример:

```
10 LET h$="
```

Режим E

CAPS SHIFT

2

\$" (ENTER)

```
20 LET b$=
```

"Режим E CAPS SHIFT E

Режим E CAPS SHIFT 7

Режим E 1

=====

Режим E 6

Режим E CAPS SHIFT 3

режим E CAPS SHIFT 9

@

Режим E CAPS SHIFT 7

Режим E 1

Режим E CAPS SHIFT 8

Режим E CAPS SHIFT 2

Режим E 7 0 " (ENTER)

Строка 30 вводится обычным порядком:

```
30 LET s$=CHR$22+CHR$17+CHR$22+b$+CHR$22+CHR$18+CHR$8+b$+CHR$22+CHR$19*CHR$22+h$
```

Если Вы внимательно следите за нашими выпусками и читаете этот раздел, то Вы

должны знать, что обозначают странные записи в строках 10 и 20.

Запустите эту программу (RUN). Сначала Вы получите только сообщение O.K., а теперь дайте команду

```
CLS:PRINT s$
```

и Вы увидите на экране изображение священного меча с рубином на рукоятке, причем сделан он только из символов ASCII. Его изображение хранится в s\$, и всякий раз, когда его надо изобразить, достаточно дать команду

```
PRINT s$.
```

Печатаемые символы, заложенные в ПЗУ "Спектрума" конечно можно достичь каких-то элементарных результатов, но почему бы не попробовать воспользоваться еще и командой OVER.

Эта команда позволяет напечатать один символ поверх другого и получить что-то новое.

Работает она несколько необычно.

INK поверх INK дает PAPER

PAPER поверх PAPER тоже дает PAPER, но

INK поверх PAPER дает INK.

Благодаря этому, если напечатать один символ поверх другого, можно получить совершенно неожиданные новые символы (нечто вроде псевдо-графики пользователя). Попробуйте напечатать * поверх > (получится нечто вроде аэроплана). Буква "С" поверх буквы "О" даст изображение пули, а * поверх восклицательного знака - пламя ракетного двигателя. Попробуйте поэкспериментировать сами, поле исследований большое, ведь возможны десятки тысяч комбинаций.

Теперь посмотрим, что можно сделать с графикой пользователя UDG. Если с ней заняться серьезно, то можно внутри Вашего компьютера найти совсем новый "Спектрум".

Поскольку каждый из символов UDG имеет довольно малые размеры, то для получения чего-то существенного их надо использовать группами. Тогда можно получать картинку любого размера.

В Вашем распоряжении 21 символ UDG, закрепленные за клавишами A - U. Даже если не заниматься манипуляциями с памятью машины, которые позволяют сильно расширить этот набор, все равно это не так мало, как кажется. Крепко подумав, можно некоторые символы UDG использовать не для одной цели.

В книге Виккерса (официальная инструкция по работе со "Спектрумом", широко распространенная на наших "рынках") в главе, посвященной UDG-графике, есть пример построения шахматных фигур. Правда они получаются такими маленькими, что к реальному делу их трудно приспособить, а вот если бы увеличить их размер хотя бы в два раза было бы совсем другое дело. Итак, существуют 6 различных шахматных фигур. Если на каждую использовать по 4 символа UDG, то потребуется 24 символа, а у нас есть только 21. Решение очень простое - сделаем основание каждой фигуры стандартным (нижние два символа), а варьировать будем только верх. Тогда нам необходимо:

$6 \times 2 + 2 = 14$ (символов).

Остался еще запас, можно даже для пешек сделать основание другим способом (поменьше размером).

Но на самом деле Вы вовсе и не ограничены 21 символом. Можете иметь их столько, сколько пожелаете, надо только сообщить об этом Вашем желании компьютеру и расплатиться по 8 байтов за каждый новый символ. Среди многочисленных системных переменных "Спектрума" есть одна, которая называется UDG и располагается в адресах 23675 и 23676.

В этих ячейках можно найти информацию об адресе первого байта, с которого начинается графика символов UDG.

Если Вы дадите команду:

```
PRINT USR "A",
```

то получите число 65368, поскольку "A" - первый символ графики UDG, то это число есть адрес первого символа.

Если Вы дадите команды:

PRINT PEEK 23675

и

PRINT PEEK 23676,

то получите 2 числа: 88 и 255 - это тоже информация об адресе первого символа UDГ. Просто таким образом в своей памяти компьютер хранит длинные двухбайтные числа.

Умножьте 255 на 256 и прибавьте 88. Вы получите тот же адрес - 65368.

Если в системную переменную UDГ записать оператором POKE другие числа, то компьютер будет искать конструкцию символов UDГ в других областях памяти. Можете выполнять это так часто, как пожелаете. Наберите программу, приведенную ниже. Обратите при этом внимание на то, что символы с чертой наверху - это графические символы и набирать их надо в графическом режиме (курсор G).

```
10 FOR f=60000 TO 60007
  READ a: POKE f,a:
NEXT f:
DATA 1, 3, 7, 15, 31, 63, 127, 255
20 POKE 23675,96:
  POKE 33676,234:
  PRINT "AAAAA"
```

Вы создали символ UDГ в форме треугольника и разместили его в новом адресе - 60 000 (строка 10)

В строке 20 Вы переадресовали системную переменную UDГ на новый набор (на адрес 60 000).

Вы по-прежнему не можете иметь в одном наборе символов UDГ более 21, но зато можете теперь иметь в памяти компьютера столько наборов, сколько хотите, переключая системную переменную UDГ на адреса, с которых начинаются эти наборы.

Попробуем теперь, используя самодельные наборы UDГ оживить Ваши программы небольшими цветными картинками. Используем для каждой картинке набор из 20 символов UDГ, тогда картинка будет иметь вид:

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T

Это вполне приемлемый размер, способный порадовать глаз будущему пользователю Вашей программы.

Теперь нарисуйте на миллиметровке или на бумаге в клетку саму Вашу картинку размером 40x32 точки. Продумайте, какие цвета INK и PAPER Вы будете использовать для каждого знакоместа. Можете использовать и FLASH и BRIGHT.

Теперь рисунок, полученный для символов "A", "B" ... "T" надо занести в память компьютера (мы работаем с 20 символами, о 21-ом мы как бы "забыли", он нам не нужен).

Как это делается, Вы можете прочитать в предыдущем номере "ZX-РЕВЮ" в разделе "Для начинающих". Написано об этом и в нашей разработке "Большие возможности Вашего Спектрума", правда мы сейчас ее снимаем с распространения в связи с тем, что готовим новое издание, исправленное и дополненное.

Конечно, формировать наборы UDГ вручную - дело крайне утомительное. Лучше воспользоваться какой-либо специально предназначенной для этой цели программой, а их немало.

Например, есть пакет HORIZONS, который поставляется вместе с фирменным "Спектрумом". В нем есть программа CHARACTER GENERATOR, служащая для этой цели. Конечно, фирменной машины у Вас может и не быть, но этот пакет программ был самым первым, попавшим к нам в страну и распространен более широко, чем что-либо еще.

Давайте используем для этой цели графический редактор "ART-STUDIO". Его у Вас не может не быть, иначе непонятно, зачем Вам компьютер. В нем есть возможность конструировать наборы символов, хотя с ним придется тоже применить "маленькие хитрости".

Войдите в режим "FONT". Вы увидите 96 символов "Спектрума". Это не UDG-символы, а символы основного набора, тем не менее и они нам подойдут. Нарисуйте первые 20 символов из набора, как Вам надо. Теперь дайте команду "SAVE FONT". Присвойте набору имя, например "MYFONT1". Он выгрузится на ленту под этим именем и займет 768 байтов (96 символов по 8 байтов). Но Вам-то ведь нужны только первые 20 символов, т.е. 160 байтов. Поэтому лишние 608 байтов надо "отрезать".

Сделаем, например, так:

- выйдите из "ARTSTUDIO"

- загрузите сформированный блок "MYFONT1" в адрес, в котором хотели бы хранить свой набор UDG, например, в адрес 60 000:

```
LOAD "MYFONT1" CODE 60000,768
```

А теперь выгрузите на ленту его первые 160 байтов, обрезав тем самым остальные:

```
SAVE "MYFONT1" CODE 60000,160
```

Теперь у Вас на ленте есть первый набор, загружать его очень просто.

```
LOAD "" CODE
```

Он всегда будет загружаться в адрес 60 000.

Аналогично можете себе наделать еще сколько хотите разных наборов, меняя их имя.

Предположим, что Вы наделали таких наборов штук 30, т.е. заготовили 30 картинок. Теперь их можно "сшить" в один сплошной блок машинного кода, начинающийся, скажем с адреса 60 000. Для этого может пригодиться следующая программа:

```
10 LET g$=CHR$22+CHR$10+CHR$0+ "ABCDE"+CHR$13+"FGHIJ"*CHR$13+ "KLMNO"+CHR$13+"PQRST"
11 REM *буквы, надчеркнутые сверху, вводятся в графическом режиме
15 LET a=60000:LET c=1
20 INPUT "name of font"; t$
30 IF t$="" THEN GO TO 200
31 IF LEN t$>10 THEN GO TO 20
32 LET c=c+1
35 LOAD t$ CODE a,168
40 RANDOMIZE a: POKE 23675,PEEK 23670: POKE 23676,PEEK 23671: PRINT g$
50 LET k=0: FOR f=22848 TO 22944 STEP 32
60 FOR n=0 TO 4: LET k=k+1: POKE f+n,7
70 INPUT "Attribute value?",V
75 IF V<1 OR V>255 THEN GO TO 70
80 POKE f+n,V: POKE a+160+k,V
90 NEXT n:NEXT f: LET a=a+180
100 PAUSE 100:CLS: GO TO 20
200 INPUT "name for your picture?", t$: IF t$="" OR LEN t$>10 THEN GO TO 200
210 SAVE t$ CODE 60000,c*180
220 VERIFY t$ CODE
```

Программа работает следующим образом. Сначала она просит Вас ввести имя первого набора символов UDG и загружает его. После этого программа запрашивает от Вас ввода цветовых атрибутов для каждого из 20 символов, из которых сделана картинка (программная переменная "V"). Если Вы не знаете, как кодируются цветовые атрибуты, воспользуйтесь таблицей 1.

Например, если Вы хотите иметь желтое изображение (INK=6) на синем фоне (PAPER=1), то должны в ответ на запрос дать число 14.

Если хотите, чтобы символ был ярким (BRIGHT), добавьте к полученному числу еще 64, а если хотите чтобы он еще и мигал (FLASH), то прибавьте еще 128. Но пожалуйста, не забывайте, что в пределах одного знакоместа (8x8 пикселей) можно иметь только один цвет INK и один цвет PAPER.

Табл. 1

		PAPER							
		0	1	2	3	4	5	6	7
INK	0	-	8	16	24	32	40	48	56
	1	1	-	17	25	33	41	49	57
	2	2	10	-	26	34	42	50	58
	3	3	11	19	-	35	43	51	59
	4	4	12	20	28	-	44	52	60
	5	5	13	21	29	37	-	53	61
	6	6	14	22	30	38	46	-	62
	7	7	15	23	31	39	47	55	63

Закончив работу с первым набором, можете приступить ко второму и т.д. Когда закончите, в ответ на вопрос об имени вашего набора нажмите 0 и Вам предложат выгрузить весь блок целиком, начиная с адреса 60000.

Каждый из наборов при выгрузке займет по 180 байтов, из них 160 - черно-белая информация (20 символов x 8 байтов) и еще 20 байтов займут цветовые атрибуты - по одному байту на каждый символ (на одно знакоместо).

Нам осталось только научиться использовать сформированный графический файл в своих программах. Для этого может пригодиться следующая программа. Наберите ее и запустите командой RUN 9999.

```
100 LET f$="001 START": GO SUB 1000: STOP
1000 LET c = VAL f$ (TO 3)*180+59820: RANDOMIZE c:
    POKE 23675,PEEK 23670:
    POKE 23676,PEEK 23671:
    PRINT g$'f$(4 TO):
    LET c=c+160:
    FOR f=22848 TO 22944 STEP 32:
    FOR n=0 TO 4:
    POKE f+n, PEEK c:
    LET c=c+1:
    NEXT n:
    NEXT f:
    RETURN
```

9998 REM *Символы от А до Т в следующей строке являются графическими и набираются в графическом режиме

```
9999 LET g$=CHR$22+CHR$10+CHR$0+"ABCDE"+CHR$13+"FGHIJ"+CHR$13+"KLMNO"+CHR$13+"PQRST"
```

В строке 9999 формируется строковая переменная g\$, содержащая в виде 20-ти символов UDG Вашу картинку. Картинка будет печататься в строке 10, начиная с нулевой колонки на экране Вашего телевизора. Место печати задается в строке 9999 кодами

CHR\$22 (AT)

CHR\$10 (номер строки)

CHR\$0 (номер колонки).

Конечно, Вы можете изменить место печати, изменив номер строки, но номер колонки пусть останется равным нулю. Дело в том, что Ваша картинка имеет ведь 4 строки, а переход к печати следующей выполняется кодом CHR\$13 (ENTER), после которого позиция печати выставляется всегда в нулевой колонке. Если захотите печатать не с нулевой колонки, то придется в строке 9999 убрать все CHR\$ 13 и заменить их на соответствующие последовательности CHR\$ 22 + CHR\$ R + CHR\$ C, где R - номер строки, а C - номер колонки.

Придется также при изменении позиции печати пересчитать адреса цветовых атрибутов, но о них чуть ниже.

Когда после запуска программы получите сообщение O.K, можете стереть строку 9999. Теперь картинка содержится в переменной g\$, которая хранится в области переменных VARS и строку 9999 можно удалить для экономии места в программной области PROG.

Переменная f\$ содержит сразу и номер Вашей картинки (сначала это 001) и название

(START).

Всю работу выполняет подпрограмма, начинавшаяся в строке 1000. Во-первых, берется номер картинки и умножается на 180. Когда к результату прибавили 59820, получается, что первая картинка находится по адресу 60000, вторая - 60180 и т.д. Оператором RANDOMIZE по этому значению выставляется другая системная переменная - SEED, поэтому когда Вы будете командой PEEK "вытаскивать" графику UDG, Вы найдете нужные адреса в этой системной переменной.

Так программа печатает Вашу картинку в текущих цветовых атрибутах экрана (как они были установлены до того). Раскрашивание же картинки производится в той же строке 1000. Файл экранных атрибутов начинается с 22528 и занимает 768 байтов (24 ряда по 32 байта).

Поскольку Ваша картинка печатается в экранной строке 10, то атрибуты начинаются с $22528 + 10 * 32 = 22848$, что Вы и видите в программной строке 1000. Закрашиванием Вашей картинки занимаются в строке 1000 два цикла - по f и по n.

Заканчивая материал этого месяца, должны сказать, что, конечно, Вы можете делать в своей программе картинки и поменьше размером и побольше, но разберитесь с этим сами. Мы дали основные идеи, а простор для исследований перед Вами необъятный.

Профессиональный подход

(Продолжение)

Продолжая печатать рекомендации известного британского программиста Стива Тернера для тех, кто делает первые шаги в написании больших и сложных программ в машинных кодах, мы поговорим о структурном программировании.

Структурное программирование - это метод организации программного проекта и самого процесса программирования. Существуют и другие методы, но этот наилучшим способом обеспечивает понимание главной задачи - что программа должна выполнять. После этого гораздо проще решать практический вопрос - как это сделать.

Я вижу следующие преимущества в структурном программировании:

1. Огромная неохватная задача разбивается на несколько простых подзадач.
2. Этот метод дисциплинирует образ мышления и порядок программирования.

Каждая процедура выполняет свою частную, отдельную задачу.

3. Разные процедуры нередко имеют сходные черты, что очень упрощает программирование.

4. Значительно упрощается процесс отладки, т.к. снижается количество ошибок.

Примечание "ИНФОРКОМА"

Стив Тернер не привел на наш взгляд самого главного достоинства структурного программирования - возможности организации коллективного программирования, а в наши дни 95% программ создаются коллективно.

При таком подходе каждый член команды может заниматься своим блоком процедур, и результат после "сшивки" должен работать, если распределение работ было произведено правильно.

Более того, достижения одного сразу становятся достижениями всех членов команды и прогресс в разработке быстро нарастает.

Правда, при коллективном подходе кроме просто структурирования программы до уровня мелких процедур необходимо еще выполнять анализ структуры с целью максимального выявления общих мест и производить согласование процедур, написанных разными программистами, а может быть и в разное время и для разных проектов.

Элементы структурированной программы.

Любую задачу, будь то какая-либо программа или инструкция по изготовлению садовой теплицы можно разделить на 3 типа команд.

1. Последовательные команды.

Одна команда идет за другой.

Компьютер выполняет свои команды последовательно. На рис. 1 это показано на структурной диаграмме.

Каждый квадрат представляет отдельную задачу.

Последовательность их исполнения - слева направо, в отличие от блок-схем, в которых отдельные блоки исполняются в последовательности "сверху-вниз".

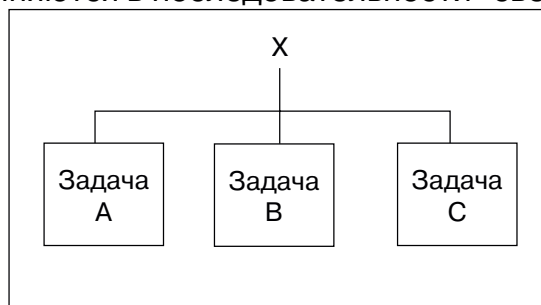


Рис. 1

2. Условное ветвление.

Если <условие> исполнить <задача>

IF <условие> DO <задача>

Все программные языки имеют средства для того, чтобы исполнять какую-либо задачу только если это необходимо. В языках высокого уровня операторы могут выполняться или не выполняться в зависимости от какого-либо условия. В языках низкого уровня (например в АССЕМБЛЕРЕ) есть инструкции, после исполнения которых определяется, какая инструкция будет выполняться следующей. Это переходы (JP и JR), вызовы (CALL) и возвраты (RET). Рекомендуется всячески избегать излишних переходов и, если использовать переходы, так только условные, например JR NC, JR Z,... и т.п.

На рис.2 представлен элементарный блок "IF" (если).

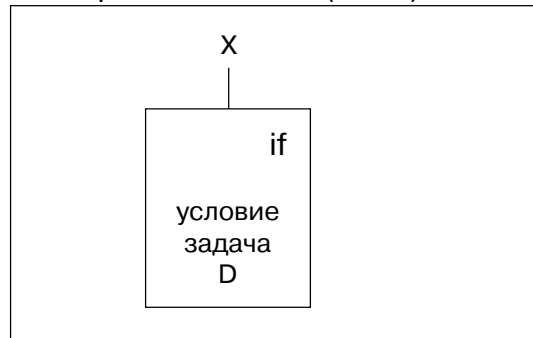


Рис.2

Имейте в виду, что как бы ни было сложно и многовариантно исходное условие, его всегда можно раздробить на серию таких элементарных IF.

На рис. 3 показана комплексная структура IF.

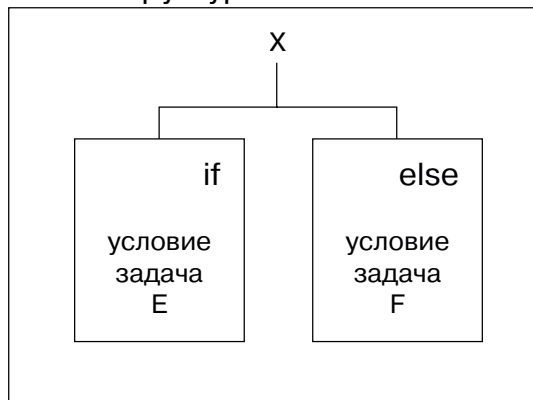


Рис.3

Здесь представлено условие IF, что делать, если оно выполняется (задача E) и что делать в ином случае - ELSE (задача F).

3. Циклы (итерации).

Они позволяют выполнить задачу многократно, пока соблюдается какое-то условие.

DO <задача> UNTIL <условие>

Задача может быть выполнена столько раз, сколько это необходимо. В Бейсике этим занимаются операторы FOR...NEXT. На АССЕМБЛЕРЕ это можно сделать двумя командами перехода. Одна ставится в конце задачи и выполняет переход к началу, а вторая - условие выхода из образовавшейся петли. Если первый переход может и не быть условным, то уж второй - обязательно условный. Часто, когда я пишу программу, ставлю условие выхода последней командой в цикле. В этом случае можно обойтись одним переходом вместо двух. Тогда программа автоматически "вываливается" из цикла, когда выходное условие перехода перестает выполняться. По этому же принципу работает и команда АССЕМБЛЕРА DJNZ.

На рис. 4 показан элементарный цикл.

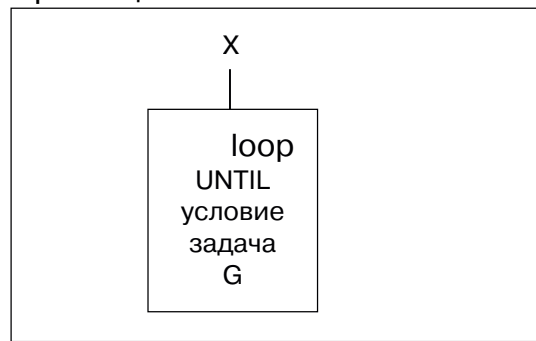


Рис.4

Работая с циклами, я всегда выполняю дополнительные операции:

- 1). Инициализация переменных перед циклом. Установка исходных значений в регистрах и т.п.
- 2). Программирование цикла.
- 3). Финишные операции после завершения цикла.

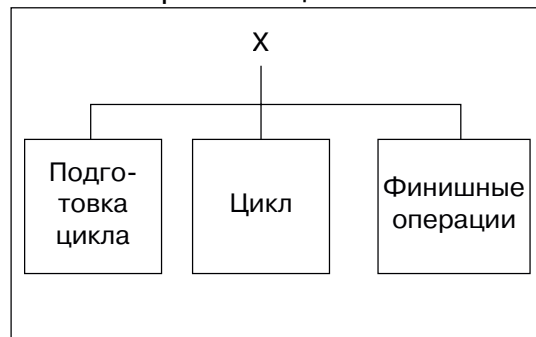


Рис.5

Объединение элементов.

Рассмотрим как объединяются эти три основных типа структурных элементов в единое целое на примере задачи о чашке кофе. На рис. 6 показана последовательность действий при выполнении этой задачи. (Судя по схеме, Стив Тернер пьет растворимый кофе или просто не хочет раскрывать глубоких секретов приготовления кофе натурального - Прим. "Инфоркома").

В правом верхнем углу блоков с условиями типа IF поставлен символ "O", а в блоках с циклами символ "***".

Предположим, что нам надо запрограммировать кухонного робота на выполнение этой задачи. Если он уже это делал, то нам достаточно одного верхнего блока "приготовить кофе", но если эта задача выполняется впервые, то она разбивается на ряд микро-задач, объединенных связями, как показано на рис. 6. Вы видите, что сначала главная задача разбивается на три под-задачи, которые далее еще более детализируются.

NB! Обратите внимание на то, что каждый горизонтальный уровень на схеме представляет полное решение задачи, но только на разных уровнях различается детальность описания. Именно в этом состоит основное преимущество структурной схемы, по сравнению со столь привычными для нас блок-схемами.

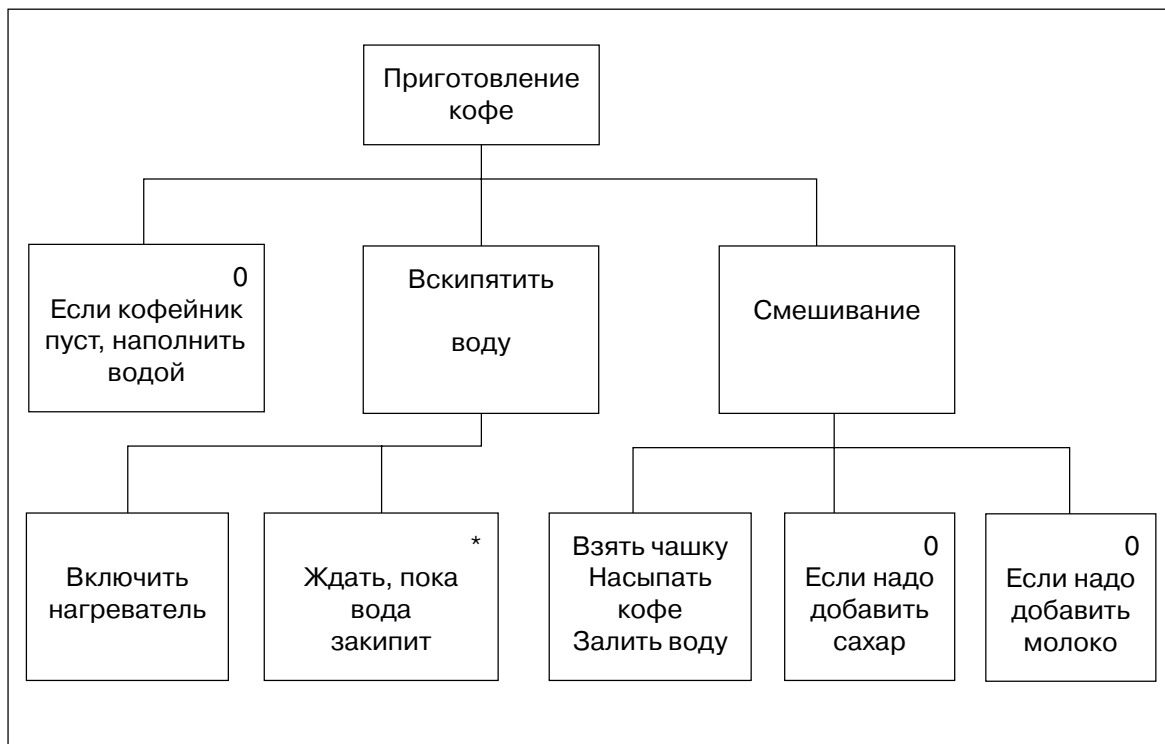


Рис.6

Можно идти от уровня к уровню все ниже и ниже, все больше и больше дробя элементарные задачи. До каких пор это делается? До тех пор, пока мы не придем к элементарным задачам, каждая из которых уже знакома нашему гипотетическому кухонному роботу.

Такой подход к подготовке программы удобен еще тем, что любой свой проект Вы всегда можете уместить на одном стандартном листе бумаги и окинуть взглядом. Одного листа бывает достаточно для размещения до восьми уровней. А как быть, если Вам их не хватило? Все равно весь проект уложится в одну страницу, просто в нижнем уровне будут блоки, которые представляют собой тоже самостоятельные задачи, и в свое время для них будут созданы свои структурные диаграммы на своих отдельных листах.

Итак, если Вы хотите, чтобы в Вашем рисунке одновременно сочетались детальность и компактность, то структурные диаграммы гораздо удобнее традиционных блок-схем.

Как структурировать программу?

Может быть сначала это кажется несколько необычным, но поверьте, потратить усилия стоят. Это вполне окупится.

Как только Вы освоите структурное программирование, то сразу увидите поразительную вещь, что в мире существуют всего лишь несколько видов программ! Вы увидите, что основные структуры из раза в раз повторяются и повторяются.

Как только Вы твердо определитесь с тем, что Ваша программа должна делать, можете приступить к ее структурированию. Если, например, Ваша программа предназначена для обработки каких-то данных (например текстовых файлов), то обычно структура такой программы отражает структуру данных и поэтому имеет смысл сначала разобраться с ними.

Например, если в конце текстовых записей в файле стоит маркер, говорящий о том, что записи закончились, то определенно в программе будет блок типа:



В то же время, если взять игровые программы, они редко развиваются вокруг одного файла данных. Здесь целесообразнее строить структуру не от структуры данных, а от тех процессов, которые будет выполнять программа.

Начните с того, что перечислите главные задачи, выполняемые программой. После этого попробуйте рассортировать их по порядку очередности исполнения.

Следующий шаг - наметьте основные циклы повторений исполнения этих главных задач.

Теперь начинайте чертить структурную диаграмму, правильно выбирая уровень, на который "подвешивается" каждая из перечисленных задач.

Вам знакома игра SPACE INVADERS? На рис. 7 приведена структурная диаграмма этой игры и, надо сказать, в эту структуру уложились бы еще сотни разных "стрелялок".

Попробуем шаг за шагом подойти к этой структуре.

1. Запишем основные задачи.

Перемещение вашей базы.

Огонь.

Перемещение кораблей противника.

Построение игровой страницы (экрана).

Ведение счета очков.

Инициализация волны вражеского налета.

Три жизни.

Начать новую игру по окончании.

На этом этапе мы пока не задумывались над очередностью исполнения этих задач.

2. Наметим основные циклы.

Здесь очень важно правильно соблюсти порядок.

После хорошего размышления у нас может получиться нечто следующее:

Игра SPACE INVADERS состоит из ряда игр, каждая из которых состоит из трех попыток (три жизни), каждая из которых включает несколько страниц (экранов), каждый из которых может многократно изменяться.

3. Теперь будем рисовать структурную диаграмму.

Начнем с верхнего бокса, в котором запишем имя программы. Теперь, используя список циклов, будем продвигаться вниз на очередной уровень и внимательно продумывать, что должно быть сделано до цикла и что после.

Иногда Вам может показаться, что бывает что-то такое, что не вписывается в структуру ни на каком уровне. Например, после определенного количества выстрелов может появиться премиальный корабль, за сбивание которого даются повышенные очки.

Если бы мы сочли эту задачу достаточно важной для рассмотрения, то должны были бы встроить еще один уровень где то между уровнем организации экранов и уровнем изменения экранов. С другой стороны, если этот элемент игры не считать таким уж очень важным, поскольку он не оказывает существенного влияния на код остальной части игры, я бы добавил его в нижний уровень, сделав условным (IF).

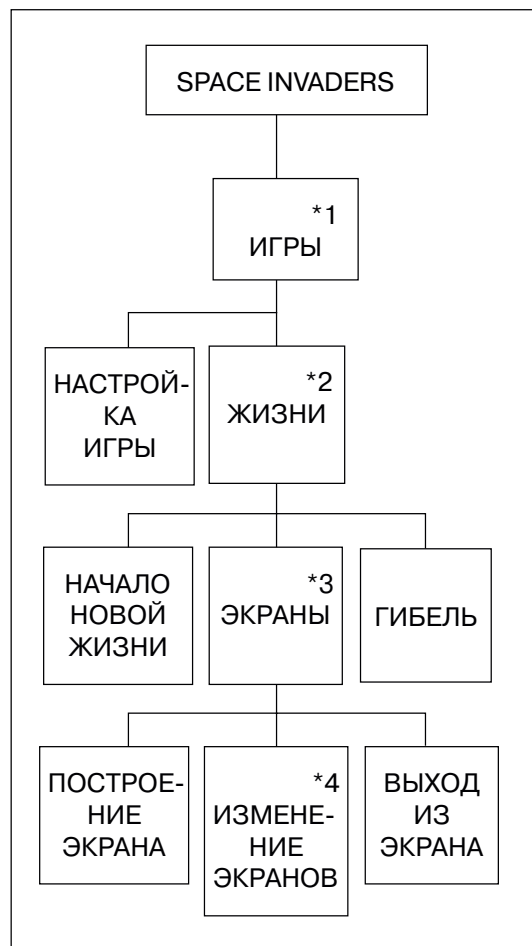


Рис. 7

Продолжительность циклов:

1. Бесконечно или до отказа от продолжения игры.
2. Три раза.
3. До гибели.
4. До гибели или по окончании волны атаки.

Очень часто в структуру приходится вводить новые уровни и перечерчивать диаграмму. Мне иногда даже приходится это делать, когда я уже активно занимаюсь программированием. И если текст моей программы строго соответствует разработанной структуре, то сделать это совсем несложно.

Обратите внимание на то, что самый последний цикл (самого низкого уровня) в своем конце должен содержать не только условие проверки окончания самого себя, но и условия проверки окончания всех остальных вышележащих циклов.

Процедура INPUT

В завершение данного выпуска я предлагаю процедуру, которая в машинных кодах выполняет то, что делает команда БЕЙСИКа INPUT.

Если бы Вы знали, как много существует коммерческих программ, в которых при вводе данных по запросу от программы нельзя стереть допущенную ошибку или вставить пропущенную букву! Если Вы используете эту процедуру вместе с процедурой PRINT, напечатанной в прошлом выпуске, то сможете в машиннокодовых программах выполнять INPUT и PRINT, как если Вы работаете в Бейсике, только конечно гораздо быстрее.

Я привожу здесь же в учебных целях и структурную диаграмму этой процедуры (рис. 8). Посмотрите на нее и постарайтесь разыскать показанные на ней блоки в программном коде.

Для того, чтобы воспользоваться процедурой, Вам надо сначала напечатать сообщение (запрос) от программы. Это можно сделать процедурой прошлого выпуска PRINT. Затем надо в регистровой паре DE установить координаты экранной позиции, в которой выполняется INPUT (D - координата y в пикселях; E - координата x в знаках).

Предполагаемая длина вводимой информации задается в регистре А (от 1 до 32 символов).

Обратите внимание на то, что буфер INPUT сделан в том же формате, который поддерживается подпрограммой PRINT. Первый байт буфера содержит длину самого буфера.

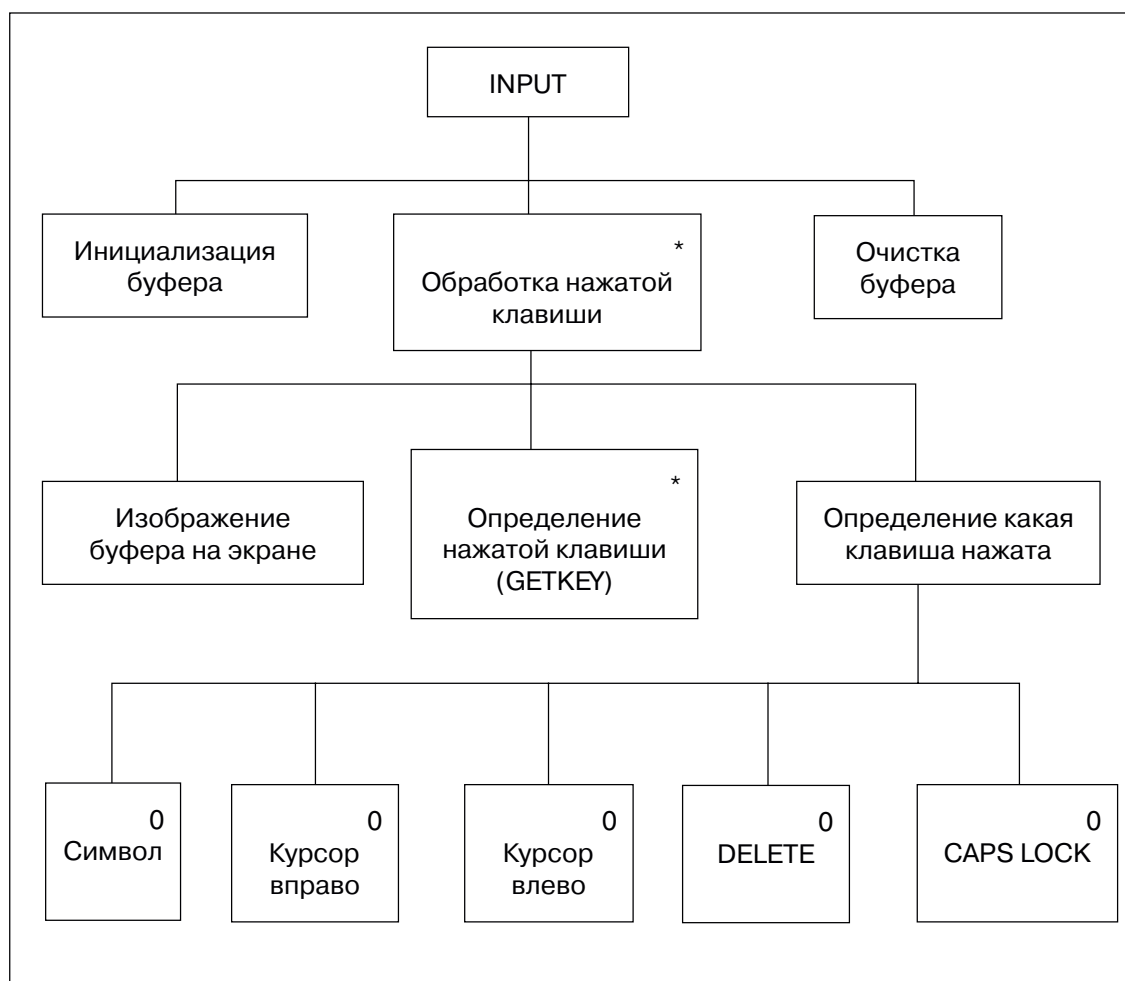


Рис.8

0000	00010	
0000	00020	
C000	00030	ORG 0C000H
C000	00040	; Пример программы INPUTX
C000 110000	00050	LD DE, 0
C003 2124C0	00060	; Координаты x, y
C006 CD88C1	00070	LD HL, MESS1
C009 111000	00080	CALL PRINT
C00C 3E10	00090	; Печать сообщения
C00E CD42C0	00100	LD DE, 10
C011	00110	; Координаты x, y
C011	00120	LD A, 16
C011	00130	; Длина поля
C011	00140	CALL INPUTX
C011	00150	
C011 110010	00160	
C014 2134C0	00170	; Данные для ввода нахо-
C017 CD88C1	00180	; дятся в буфере и готовы
C01A 110E10	00190	; к использованию
C01D 214DC1	00200	
C020 CD88C1	00210	LD DE, 1000H
C023 C9	00220	LD HL, MESS2
C024	00230	CALL PRINT
C024	00240	; Печать сообщ.
C025 454E5445	00250	LD DE, 100EH
C029 5220594F		LD HL, LENGH
		CALL PRINT
		; Печать резуль.
		RET
		DB 15
		DM 'ENTER YOUR NAME'

C02D	5552204E		
C031	414D45		
C034	0D	00260	MESS2 DB 13
C035	594F5552	00270	DM 'YOUR NAME IS'
C039	204E414D		
C03D	45204953		
C041	20		
C042		00280	
C042		00290	;-----
C042		00300	
C042		00310	; Условия вызова процедуры
C042		00320	; INPUTX:
C042		00330	; A - длина вводимых данных 1-32
C042		00340	; D - у координата (0 - верх)
C042		00350	; E - x координата
C042		00360	
C042		00370	INPUTX ;Начало процедуры
C042		00380	;Инициализация буфера
C042	324DC1	00390	OK LD (LENGTH),A
C045	6F	00400	LD L,A
C046	ED5375C1	00410	LD (OUTPOS),DE
C04A	AE	00420	XOR A
C04B	3E73C1	00430	LD (CURSOR),A
C04E	67	00440	LD H,A
C04F	114EC1	00450	LD DE,BUFFER
C052	19	00460	ADD HL,DE
C053	2270C1	00470	LD (BUFEND),HL
C056	3A4DC1	00480	LD A,(LENGTH)
C059	47	00490	LD B,A
C05A	214EC1	00500	LD HL,BUFFER
C05D	367F	00510	LD (HL),CURS
C05F	23	00530	INC HL
C060	3E2D	00530	LD A,ULINE
C062	77	00540	CLEAR LD (HL),A
C063	23	00550	INC HL
C064	10FC	00560	DJNZ CLEAR
C066	213B5C	00570	LD HL,FLAGS
C069	CBAE	00580	RES 5,(HL)
C06B		00590	
C06B		00600	;Обработка принятой клавиши
C06B		00610	;в ожидании клавиши ENTER
C06B		00620	
C06B		00630	ACCEPT
C06B	ED5BT5C1	00640	LD DE,(OUTPOS)
C06F	214DC1	00600	LD HL,LENGTH
C072	CD88C1	00660	CALL PRINT
C075		00670	
C075		00680	;Прием очередной клавиши
C075		00690	
C075	213B5C	00700	GETKEY LD HL,FLAGS
C078	CB6E	00710	BIT 5,(HL)
C07A	28F9	00720	JR Z,GETKEY
C07C	3A065C	00730	LD A,(LASTK)
C07F	CBAF	00740	RES 5,(HL)
C061	FE06	00750	CP 6
C083	38F0	00760	JR C,GETKEY
C085	FE0D	00770	CP 0DH
C087	CA2FC1	00780	JP Z,ENTER
C08A	FE08	00790	CP 8
C08C	283A	00800	JR Z,CRLEFT
C08E	FE09	00810	CP 9
C090	284F	00820	JR Z,CRRIGT
C092	FE0C	00830	CP 0CH
C094	2669	00840	JR Z,DELETE
C096	FE06	00850	CP 6

C098 CA25C1	00860	JP Z,CAPSLK	
C09B FE20	00870	CP 20H	
C09D 38D6	00880	JR C,GETKEY	
C09F FE80	00890	CP 80H	
C0A1 30D2	00900	JR NC,GETKEY	
C0A3	00910		
C0A3	00920	;Если принят печатный символ	
C0A3	00930		
C0A3 3274C1	00940	LD (LETTER),A	; Временный стек
C0A6 3A73C1	00950	LD A,(CURSOR)	
C0A9 47	00960	LD B,A	
C0AA 3A4DC1	00970	LD A,(LENGTH)	
C0AD 90	00980	SUB B	
C0AE 8805	00990	JK Z,GETKEY	; Буфер полон
C0B0 4F	01000	LD C,A	
C0B1 78	01010	LD A,B	
C0B2 3C	01020	INC A	; Сдвиг курсора
C0B3 3273C1	01030	LD (CURSOR),A	
C0B6	01040		
C0B6 ED5B70C1	01050	LD DE,(BUFEND)	
C0BA 2A70C1	01060	LD HL,(BUFEND)	
C0BD 2B	01070	DEC HL	
C0BE 0600	01080	LD B,0	
C0C0 EDB8	01090	LDDR	
C0C2 3A74C1	01100	LD A,(LETTER)	
C0C5 12	01110	LD (DE),H	
C0C6 18A3	01120	JR ACCEPT	
C0C8	01130		
C0C8	01140	;Если курсор влево	
C0C8	01150		
C0C8 3A73C1	01160	CRLEFT LD A,(CURSOR)	
C0CB A7	01170	AND A	
C0CC 28A7	01180	JR Z,GETKEY	
C0CE 3D	01190	DEC A	
C0CF 3273C1	01200	LD (CURSOR),A	
C0D2 214EC1	01210	LD HL,BUFFER	
C0D5 5F	01220	LD E,A	
C0D6 1600	01230	LD D,0	
C0D8 19	01240	ADD HL,DE	
C0D9 7E	01250	LD A,(HL)	
C0DA 367F	01260	LD (HL),CURS	
C0DC 23	01270	INC HL	
C0DD 77	01280	LD (HL),A	
C0DE C36BC0	01290	JP ACCEPT	
C0E1	01300		
C0E1	01310	;Если курсор вправо	
C0E1	01320		
C0E1	01330	CRRIGT	
C0E1 3A4DC1	01340	LD A,(LENGTH)	
C0E4 4F	01350	LD C,A	
C0E5 3A73C1	01360	LD A,(CURSOR)	
C0E8 B9	01370	CP C	
C0E9 CA75C0	01360	JP Z,GETKEY	
C0EC 3C	01390	INC A	
C0ED 3273C1	01400	LD (CURSOR),A	
C0F0 214EC1	01410	LD HL,BUFFER	
C0F3 5F	01420	LD E,A	
C0F4 1600	01430	LD D,0	
C0F6 19	01440	ADD HL,DE	
C0F7 7E	01450	LD A,(HL)	
C0F6 367F	01460	LD (HL),CURS	
C0FA 2B	01470	DEC HL	
C0FB 77	01480	LD (HL),A	
C0FC C36BC0	01490	JP ACCEPT	
C0FF	01500		

C0FF	01510	; Если DELETE	
C0FF	01520		
C0FF	01530	DELETE	
C0FF 3A73C1	01540	LD A, (CURSOR)	
C102 A7	01550	AND A	
C103 CA75C0	01560	JP Z, GETKEY	
C106 5F	01570	LD E, A	
C107 3A4DC1	01560	LD A, (LENGTH)	
C10A 93	01590	SUB E	
C10B 3C	01600	INC A	
C10C 4F	01610	LD C, A	
C10D 7B	01620	LD A, E	
C10E 3D	01630	DEC A	; Курсор влево
C10F 3273C1	01640	LD (CURSOR), A	
C112	01650		
C112 214EC1	01660	LD HL, BUFFER	
C115 1600	01670	LD D, 0	
C117 19	01680	ADD HL, DE	
C116 54	01690	LD D, H	
C119 5D	01700	LD E, L	
C11A 1B	01710	DEC DE	
C11B 0600	01720	LD B, 0	
C11D EDB0	01730	LDIR	
C11F 3E2D	01740	LD A, ULINE	
C121 12	01750	LD (DE), A	
C122 C36BC0	01760	JP ACCEPT	
C125	01770		
C125	01780	; Если CAPS LOCK	
C125	01790		
C125 216A5C	01800	CAPSLK LD HL, 5C6AH	
C128 3E08	01810	LD A, 8H	
C12A AE	01820	XOR (HL)	
C12B 77	01830	LD (HL), A	
C12C C36BC0	01840	JP ACCEPT	
C12F	01850		
C12F	01860	; Если ENTER	
C12F	01670		
C12F	01860	ENTER	
C12F 114EC1	01890	LD DE, BUFFER	
C132 214EC1	01900	LD HL, BUFFER	
C135 3A4DC1	01910	LD A, (LENGTH)	
C138 47	01920	LD B, A	
C139 7E	01930	CLEAN LD A, (HL)	
C13A 23	01940	INC HL	
C13B FE7F	01950	CP CURS	
C13D 2808	01960	JR Z, IGNORE	
C13F FE2D	01970	CP ULINE	
C141 2002	01980	JR NZ, PACK	
C143 3E20	01990	LD A, SPACE	
C145 12	02000	PACK LD (DE), A	
C146 13	02010	INC DE	
C147 10F0	02020	IGNORE DJNZ CLEAN	
C149 3E20	02030	LD A, SPACE	
C14B 12	02040	LD (DE), A	
C14C C9	02050	RET	
C14D	02060		
C14D	02070		
C14D	02060	; Переменные буфера	
C14D 0E090			
C14D 00	02100	LENGTH DB 0	
C14E	02110	BUFFER DS 34	; Резервирует 34 б
C170	02120		
C170 0000	02130	BUFEND DW 0	
002D	02140	ULINE EQU ' - '	
007F	02150	CURS EQU 127	

0020	02160	SPACE	EQU ' '
C172	02170		
C172	02180	;Настройка переменных печати	
C172	02190		
5C3B	02200	FLAGS	EQU 5C3BH
5C6A	02210	FLAGS2	EQU 5C6AH
5C08	02220	LASTK	EQU 23560
c172 00	02230	COUNT	DB 0
C173 00	02240	CURSOR	DB 0
C174 00	02250	LETTER	DB 0
C175 0000	02260	OUTPOS	DW 0
C177	02270		

СЕКРЕТЫ ПЗУ

(Продолжение).

Продолжая начатый разговор о постоянном запоминающем устройстве Вашего компьютера, мы рассмотрим сегодня процедуры, управляющие выдачей информации на экран и на принтер.

Специально оговорим, что под словом "принтер" здесь понимается ZX-принтер - периферийное печатающее устройство, разработанное К. Синклером для компьютеров ZX-Spectrum. Его характерными особенностями является то, что он поддерживается командами интерпретатора БЕЙСИКа LLIST, LPRINT, COPY. За ним закреплён постоянный канал - "P". Он печатает в строку 32 символа, конструкция которых полностью совпадает с конструкцией символов на экране (8x8 пикселей).

Скорее всего Вы применяете для работы с компьютером точечно-матричный принтер, для которого в ПЗУ никакой поддержки нет. В этом случае Вы обеспечиваете физическую подстыковку с помощью специального интерфейса (последовательного RS232 или параллельного "Центроникс"), а программное управление - с помощью специальной небольшой программы - драйвера, которая зависит от того, какой интерфейс Вы подключаете.

Неплохо обо всем этом написано в разработке НТК "ПЛЮС" - "Интерфейсы "Спектрума".

Процедуры, содержащиеся в этом разделе ПЗУ очень активно используют в своей работе некоторые системные переменные компьютера и флаги, поэтому прежде чем переходить к подробному описанию раздела мы считаем необходимым дать сводку этих системных переменных и расписать побитно назначение флагов. Это значительно упростит Вам работу с этой информацией и позволит нам в тексте ссылаться на эти переменные по имени, не расшифровывая их назначение.

Системные переменные.

TVDATA - 23566,23567.

Переменная содержит те параметры, которые сопровождают операторы INK, PAPER, FLASH, BRIGHT, INVERSE, OVER, AT и TAB. Если параметр один (для INK и пр.) - он содержится в младшем байте - по адресу 23566. Если же параметров два (AT и TAB) - первый содержится в младшем байте, а второй - в старшем.

DF-SZ - 23659.

Здесь хранится количество строк нижней части экрана.

S-POSN - 23688,23669.

Здесь хранятся номер колонки позиции печати и номер строки печати (при печати в главную часть экрана).

DF-CC - 23684,23685.

Здесь хранится адрес позиции печати в дисплейном файле (при печати в главную часть экрана). Напоминаем, что дисплейный файл состоит из двух областей:

Область черно-белой графики занимает 6144 байта и располагается в адресах 16384 - 22527.

Область цветовых атрибутов занимает 768 байтов и размещается в адресах 22528 - 23295.

S-POSN-L - 23690,23691.

То же, что и S-POSN, но для нижнего окна экрана.

DF-CCL-23666,23687.

То же, что и DF-CC, но для нижнего окна экрана.

ECHO-E - 23682,23683. Здесь хранятся номер колонки позиции печати и номер строки печати (при печати в нижнюю часть экрана в операциях INPUT).

P-POSN - 23679.

Здесь хранится номер позиции печати на принтере.

PR-CC - 23680.

Здесь хранится младший байт адреса текущего печатаемого символа в буфере принтера. Старший байт хранить не надо, поскольку он и так известен, т.к. весь буфер принтера имеет всего 256 байтов. Напомним, что он располагается в адресах 23296 - 23551.

MEMBOT - 23696 - 23727.

Это область памяти калькулятора. В тридцати байтах могут разместиться 6 ячеек памяти от M0 до M5 (по 5 байтов на каждое число или строковую переменную). Обычно калькулятор работает храня данные на своем стеке, но это не всегда удобно и иногда эти ячейки используются.

UDG - 23675,23676.

В этих ячейках хранится двухбайтный адрес, начиная с которого расположена конструкция символов графики пользователя UDG.

CHARS -23606,23607.

В этих ячейках хранится двухбайтное число, указывающее на 256 байтов ниже, чем адрес с которого начинается размещение образов символов в ПЗУ компьютера.

Указание не на начало набора символов, а на 256 байтов ниже сделано потому, что как известно печатные символы начинаются с кода номер 32, а поскольку на каждый символ отводится по 8 байтов памяти, то для того, чтобы по номеру символа можно было сразу отыскать адрес расположения его шаблона и сделан сдвиг на 256 байтов.

ATTR-P - 23693.

Содержит информацию о постоянных атрибутах экрана.

Биты 0, 1, 2 - INK.

Биты 3, 4, 5 - PAPER.

Бит 6 - BRIGHT.

Бит 7 - FLASH.

ATTR-T - 23695.

То же, что и ATTR-P, но для временных атрибутов.

MASK-P - 23694.

Эта системная переменная нужна для реализации режимов INK 8, PAPER 8, BRIGHT и FLASH 8 на постоянной основе.

Биты 0, 1, 2 - включены, если установлено INK 8. Биты 3, 4, 5 - включены, если установлено PAPER 8. Бит 6 включен в режиме BRIGHT 8. Бит 7 включен в режиме FLASH 8.

MASK-T - 23696.

То же, что и MASK-P, но для временных атрибутов.

Для тех, кто не все знает о "Спектруме" дадим некоторые пояснения:

"Временными" атрибутами называются те, которые действуют на один оператор печати. Они вводятся в команде PRINT через точку с запятой. А далее будут действовать

постоянные атрибуты.

Например:

10 INK 5: PAPER 1

20 PRINT INK 7; PAPER 2; "ZX"

Здесь в строке 10 операторами INK и PAPER вводятся постоянные цветовые атрибуты. Они действуют на всю программу или до тех пор, пока не будут изменены другими такими же операторами.

В строке же 20 действующий оператор - PRINT, а INK и PAPER только устанавливают временные атрибуты, которые действуют только на эту строку.

Теперь несколько слов об INK 9 и PAPER 9. В принципе все знают, что "Спектрум" оперирует с 8-ю основными цветами, пронумерованными от 0 до 7. Тем не менее команды INK 8, PAPER 8 и даже FLASH 8 и BRIGHT 8, а также INK 9 и PAPER 9 могут иметь место.

INK 8, PAPER 8, BRIGHT 8, FLASH 8 устанавливают текущие атрибуты "прозрачными". Это означает, что при печати на экране атрибуты в знакоместе устанавливаются такими, какими они были в этом знакоместе до печати, очень удобно, если Вы не знаете, какими они были, но хотите их не изменять.

INK 9, PAPER 9 - устанавливают атрибуты "контрастными". Так, INK 9 установит цвет символов таким образом, чтобы он наилучшим образом контрастировал с текущим цветом PAPER.

Флаговые переменные.

Состояние битов расписывается справа налево, т.е. бит 0 - младший, а бит 7 - старший.

FLAGS - 23611.

Бит 0 - включен, если перед печатью не требуется ведущий пробел.

Бит 1 - включен, если ZX-принтер подключен.

PFLAG - 23697.

Карта битов этой флаговой переменной характеризует установку атрибутов OVER, INVERSE, INK 9, PAPER 9. Включенное состояние этих битов означает:

Бит 0 - режим OVER 1 (временный)

Бит 1 - режим OVER 1 (постоянный)

Бит 2 - INVERSE 1 (временный)

Бит 3 - INVERSE 1 (постоянный)

Бит 4 - INK 9 (временный)

Бит 5 - INK 9 (постоянный)

Бит 6 - PAPER 9 (временный)

Бит 7 - PAPER 9 (постоянный).

TV-FLAG

Бит 0 - включен, если печать выполняется в нижнюю часть экрана (канал "К").

Бит 3 - включен, если идет редактирование программной строки.

Бит 4 - включен при автоматическом листинге.

Бит 5 - включен, если при нажатии клавиши должна очиститься нижняя часть экрана.

FLAGS-2

Бит 1 - выключен, если буфер принтера пуст.

Основными блоками этого раздела являются:

09F4 - 0A10 - PRINTOUT - выполняет функции диспетчера.

0A11 - 0A22 - таблица управляющих символов (кодов).

0A23 - 0A3C - CURSOR LEFT - управляет перемещением курсора влево.

0A3D - 0A4E - CURSOR RIGHT - управляет перемещением курсора вправо.

0A4F - 0A5E - CARRIAGE RETURN - обработка кода "возврат каретки" (ENTER).

0A5F - 0A68 - PRINT COMMA - Обработка кода CHR\$ 6.

0A69 - 0A6C - PRINT QUESTION MARK - печать вопросительного знака.

0A6D - 0AD8 - CONTROL CHARACTERS WITH OPERANDS - обрабатывает те управляющие коды, которые требуют наличия после себя определенного операнда.

0A09 - 0ADB - PRINTABLE CHARACTER CODES - выполняет печать печатаемых символов.

0ADC - 0B02 - POSITION STORE - запоминание координат знакоместа в системных переменных.

0B03 - 0B23 - POSITION FETCH - ввод координат позиции печати из соответствующих системных переменных.

0B24 - 0B7E - PRINT ANY CHARACTERS - печать символов, токенов, символов блочной графики и символов графики пользователя.

0B7F - 0BDA - PRINT ALL CHARACTERS - печать шаблонов символов

0BDB - 0C09 - SET ATTRIBUTE BYTE - обработка атрибутов.

0C0A - 0C3A - MESSAGE PRINTING - печать токенов ключевых слов и системных сообщений.

0C3B - 0C40 - PO-SAVE - вспомогательная подпрограмма.

0C41 - 0C55 - TABLE SEARCH - подпрограмма поиска в таблицах.

0C56 - 0D4C - TEST FOR SCROLL - проверяет необходимость "прокрутки" экрана.

0D4D - 0DCA - TEMPORARY COLOUR ITEMS - обработка временных установок атрибутов.

0D6B - 0DFD - CLS COMMAND - выполняет очистку экрана.

0DFE - 0E43 - SCROLLING - выполняет "прокрутку" экрана.

0E44 - 0E87 - CLEAR LINES - выполняет очистку заданного количества строк.

0E88 - 0E9A - CL-ATTR - определяет адрес байта в файле атрибутов экрана по координатам знакоместа.

0E9B - 0EAB - CL-ADDR - определяет адрес байта в экранном файле по координатам знакоместа.

0EAC - 0EDF - COPY COMMAND - обрабатывает команду COPY.

0EDF - 0EF3 - COPY-BUFF - передает данные из буфера на принтер.

0EDF - 0EF3 - CLEAR PRINTER BUFFER - выполняет очистку буфера принтера.

0EF4 - 0F2B - COPY LINE - копирует строку пикселей (в поддержку COPY COMMAND).

Теперь рассмотрим процедуры, входящие в эти подпрограммы подробно.

09F4 - 0A10 - PRINT-OUT.

При входе в данную процедуру в регистре А должен содержаться код печатного символа, управляющий код или код токена ключевого слова.

Процедура вызывает PO-FETCH для ввода координат позиции печати, а затем определяет что же за символ ей предстоит напечатать.

Если это печатный символ - переход на PO-ABLE (0AD9).

Если управляющий код от 00 до 05 - переход для печати вопросительного знака на PO-QUEST (0A69).

То же, если управляющий код от 18 до 1F.

Для прочих управляющих кодов (от 6 до 17) выполняется следующая операция:

Код символа прибавляется к базовому адресу таблицы управляющих кодов (0A11-0A22). Базовым служит адрес 0A0B. В этой таблице содержатся "смещения" адреса, с которого начинается процедура обработки данного кода относительно положения этого кода в таблице и результат отправляется на стек. Далее выполняется переход на процедуру PO-FETCH, заготовленный на стеке адрес процедуры обработки данного управляющего кода обеспечивает переход к ней после возврата по RET из PO_FETCH.

(Примечание: все управляющие коды Вы можете просмотреть в "Справочнике по программированию в машинных кодах Z-80" - т. 3 нашего трехтомника.)

0A11 - 0A23 - таблица управляющих кодов.

Здесь расположены "смещения" адресов для процедур обработки управляющих кодов, начиная с CHR\$ 6 и кончая CHR\$ 23 (TAB). Ячейки 0A19 и 0A1A - не используются.

Подпрограмма "CURSOR LEFT".

0A23 - 0A38 - PO-BACK-1.

При входе в эту процедуру в регистре В содержится текущий номер строки, а в регистре С - номер колонки.

Проверяется не достигнут ли левый край. Если нет, то переход на выходную процедуру PO-BACK-3.

Если да, то проверяется подключен ли принтер по первому биту FLAGS, в этом случае переход на PO-BACK-2.

Если край достигнут, переход на новую строку, проверка на то не последняя ли это строка и переход на PO-BACK-3.

0A36 - 0A39 - PO-BACK-2.

Регистр С устанавливается на крайнюю левую колонку.

0A3A - 0A3C - PO-BACK-3.

Переход на CL-SET (0DD9) для печати и последующего возврата через PO-STORE (0ADC).

Подпрограмма "CURSOR RIGHT".

0A3D - 0A4E - PO-RIGHT.

Процедура вызывает системную переменную P-FLAG, сохраняет ее на стеке, включает в ней первый бит (аналогично команде БЕЙСИКа) OVER 1; и вызовом процедуры PO-CHAR (0B65) печатает в той же позиции пробел (CHR\$ 32) и восстанавливает со стека P-FLAG перед возвратом.

Процедура имеет ошибку. Дело в том, что возврат надо было бы делать через PO-STORE, чтобы запомнить в системных переменных координаты позиции печати.

Подпрограмма "CARRIAGE RETURN".

0A4F - 0A5E - PO-ENTER.

Если печать идет на принтер (проверяется по FLAGS), в буфер принтера выдается символ CHR\$ 13 переходом на COPY-BUFF (0ECD).

Если на экран - проверка на необходимость скроллинга вызовом PO-SCR (0C55) и переход на новую строку.

Выход выполняется переходом на CL-SET (0DD9).

Подпрограмма "PRINT COMMA".

0A5F - 0A68 - PO-COMMA.

Процедура выполняет ввод текущих координат позиции печати вызовом PO-FETCH (0B03), чего в принципе делать и не обязательно, поскольку это было уже сделано в PRINT-OUT (09F1).

Позиция печати в регистре А проходит сравнение по AND 10H, в результате чего устанавливается либо нулевая, либо шестнадцатая колонка печати.

Выход выполняется переходом на PO-FILL (0AC3).

Подпрограмма "PRINT A QUESTION MARK".

0A69 - 0A6C - PO-QUEST.

При попытке распечатать непечатный символ должен выдаваться знак "?". Его код (3F) вводится в аккумулятор, а для печати выполняется переход на PO-ABLE (0AD9).

Подпрограмма "CONTROL CHARACTERS WITH OPERANDS".

Некоторые управляющие коды требуют после себя наличия операндов. Так, после кодов INK или PAPER должен идти байт, указывающий на то, какой цвет устанавливается. А такой код как AT требует даже два операнда для указания координат позиции печати.

0A6D - 0A74 - PO-TV-2.

Эта процедура обрабатывает второй операнд для операций AT и TAB и делает следующие операции:

Загружает в DE 0A87 - это адрес процедуры PO-CONT. Это нужно чтобы после PO-CHANGE обеспечить выход через PO-CONT.

Операнд направляется из аккумулятора в старший байт системной переменной TV-DATA и выполняется переход на PO-CHANGE (0A80).

0A75 - 0A79 - PO-2-OPER.

Сюда мы попадаем, если идет обработка управляющих кодов AT и TAB, т.е. тех, после которых идут два операнда.

Загружается в DE 0A6D - адрес процедуры PO-TV-2. Это нужно чтобы после PO-CHANGE сделать второй проход, поскольку операндов два. Далее переход на PO-TV-1.

0A7A - 0A7C - PO-1-OPER.

Это точка входа для управляющих кодов, после которых идет один операнд. Здесь только устанавливается в DE адрес 0A87 для обеспечения выхода через PO-CONT.

0A7D - 0A7E - PO-TV-1.

Операнд направляется из аккумулятора в младший байт системной переменной TV-DATA.

0A80 - 0A86 - PO-CHANGE.

Установленный выше адрес перегружается из DE в HL для обеспечения перехода на PO-CONT, если один операнд или на PO-TV-2, если есть еще и второй.

0AB7 - 0AAB - PO-CONT.

Продолжает работу, когда операнды приняты и введены в TVDATA.

В качестве выходного адреса устанавливается 09F4 (PRINT-OUT). Это делается тем же способом, в DE вводится 09F4 и вызывается PO-CHANGE для переброски его в HL.

Если обрабатывается управляющий код от 16 до 21 (от INK до OVER), то переход на CO-TEMP-5 (2211), где изменяются цветовые системные переменные, а если код равен 23 (TAB) - переход на PO-TAB (0AC2).

Далее идет обработка кода 22 (AT). Изменяются координаты позиции печати (они хранились в BC). Если координаты выходят за допустимые пределы - переход на PO-AT-ERR (0AAC). Проверяется работа с принтером. Если печать идет на принтер - переход на PO-AT-SET (0ABF) иначе продолжение работы.

0AAC-0ABE - PO-AT-ERR.

Сюда мы попадаем двумя путями. Во-первых, из предыдущей процедуры, если позиция печати выходит за допустимые пределы и, во вторых, естественно продолжая нормальную работу.

В первом случае сразу выполняется переход на REPORT-B (1E9F) для печати сообщения Integer out of range.

Во втором случае проверяется не идет ли печать в нижнее окно экрана (по биту 0 переменной TV-FLAG). Если да, то переход на PO-SCR (0C55). Если печать идет в основную часть экрана, то проверяется сравнением с DF-SZ есть ли место для печати. Если его нет, то переход на REPORT-5 (0C86) для сообщения out of screen.

0ABF - 0AC1 - PO-AT-SET.

Выполняется возврат путем перехода на 0DD9 (CL-SET).

0AC2 PO-TAB.

Начинается обработка управляющего кода 22 (TAB).

0AC3 - 0ACF - PO-FILL.

Вызовом PO-FETCH (0B03) вводятся текущие координаты позиции печати. Вычисляется новая позиция печати. В регистре D устанавливается необходимое число пробелов.

0AD0 - 0AD8 - PO-SPACE.

В цикле вызывается процедура PO-SAVE для печати пробелов. Цикл заканчивается, когда содержимое D становится равным нулю, далее - возврат.

Подпрограмма PRINTABLE CHARACTER CODES.

0AD9 - 0ADB - PO-ABLE.

Это только точка входа. Дальнейшая работа отсюда передается на PO-ANY (0B24) посредством ее вызова CALL. После возврата из нее работу продолжает PO-STORE.

Подпрограмма "POSITION STORE".

0ADC - 0AEF - PO-STORE.

Если печать идет на принтер (проверяется по FLAGS) - переход на PO-ST-PR (0AFC).

По системной переменной TV-FLAG проверяется в какую часть экрана идет печать - в главную или в нижние две строки. В последнем случае переход на PO-ST-E (0AF0).

Далее выполняется запоминание позиции печати в системных переменных S-POSN и DF-CC и возврат.

0AF0 - 0AFB - PO-ST-E.

Запоминает координаты позиции печати в системных переменных S-POSN-L, ECHO-E, DF-CCL и выполняет возврат.

0AFC - 0B02 - PO-ST-PR.

Если печать идет на принтер, координаты позиции печати запоминаются в переменных P-POSN и PR-CC. Далее возврат.

Подпрограмма "POSITION FETCH".

Эта подпрограмма выполняет функции, противоположные предыдущей. Она вводит координаты позиции печати в регистры процессора из соответствующих переменных.

0B03 - 0B1C - PO-FETCH.

Если печать на принтере, переход на PO-F-PR (определяется по FLAGS), в противном случае выясняется в какую часть экрана идет печать (определяется по TV-FLAG).

Если печать производится в главный экран, то координаты позиции печати берутся из S-POSN и DF-CC, а если в "системном окне", т.е. в нижних двух строках экрана - из переменных S-POSNL и DF-CCL.

Далее - возврат.

0B1D - 0B23 - PO-F-PR.

Для печати на принтере координаты позиции печати снимаются из системных переменных P-POSN и PR-CCL. Далее возврат.

Подпрограмма "PRINT ANY CHARACTERS".

0B24 - 0B37 - PO-ANY.

Это начало основной последовательности процедур, предназначенных для исполнения печати символов и токенов на экране или на принтере.

Сначала выполняется распределение по процедурам, в зависимости от того, что за символ должен печататься.

Если это одиночный символ - переход на PO-CHAR (0B65).

Если это токен ключевого слова или символ графики пользователя - переход на PO-T&UDG (0B52).

Если это символ блочной графики, вызывается через CALL процедура PO-GR-1, выполняющая его конструирование. Поскольку PO-GR-1 "портит" координаты позиции печати, содержащиеся в HL, то FO-FETCH вызывается еще раз.

Для самой же печати графического символа выполняется переход на PR-ALL (0B7F).

0B38 - 0B3D - PO-GR-1.

Графические символы конструируются с использованием ячеек памяти калькулятора M0 и M1, для чего сначала в регистре HL устанавливается адрес системной переменной MEMBOT, определяющей начало памяти калькулятора. Далее вызывается процедура PO-GR-2. Поскольку сама PO-GR-2 стоит сразу за PO-GR-1, то фактически по ней происходит два прохода.

0B3E - 0B4B - PO-GR-2.

На входе в эту процедуру регистр B содержит код символа блочной графики (от 128 до 143), он был установлен в PO-ANY.

Конструкция шаблонов символов блочной графики в "Спектруме" сделана очень интересно. В принципе она заложена уже в самом номере символа (а точнее - в его четырех младших битах). Если представить символ блочной графики как выполненный из четырех блоков, каждый из которых имеет размер 4x4 пикселя, то получится следующее:

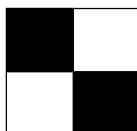
1	0
3	2

Здесь цифрами показаны номера младших битов кода символа, отвечающие за цвет своего участка. Возьмем для примера символ с кодом 134. Разложим его на биты: 134 - 1000 0110.

Тогда четыре младших бита дают рисунок:

1	0
0	1

Закрасим единицы черным цветом и получим рисунок:



Этим декодированием и занимается процедура PO-GR-2, но только для первой строки символа (из восьми).

0B4C - 0B51 - PO-GR-3.

Здесь организовывается цикл (4 прохода), в результате которого определяются уже 4 верхние строки символа.

Поскольку, как мы уже говорили, процедура PO-GR-2, а вслед за ней и PO-GR-3 фактически проходятся дважды, то на втором проходе конструируются нижние четыре строки шаблона символа.

Заканчивается возвратом.

0B52 - 0B5E - PO-T&UDG.

Здесь мы можем рассмотреть как конструируются изображения токенов ключевых слов и шаблоны символов графики пользователя (UDG).

Сначала выполняется селекция. Если речь идет о печати токенов, то этим будет заниматься PO-T (0B5F), а для символов UDG вводится адрес, начиная с которого в памяти компьютера расположены их шаблоны. Адрес расположен в системной переменной UDG.

Для выполнения их печати выполняется безусловный переход на PO-CHAR-2.

0B5F - 0B64 - PO-T.

Отсюда для печати токенов вызывается PO-TOKENS (0C10), а по окончании печати безусловным переходом через PO-FETCH (0B03) выполняется возврат в вызывающую программу.

0B65 - 0B69 - PO-CHAR.

Сюда передается управление из PO-ANY, если печатается обычный символ. На стеке сохраняются координаты позиции печати, а затем в BC устанавливается базовый адрес таблицы символов. Он берется из системной переменной CHARS.

0B6A - 0B75 - PO-CHAR-2.

Выключается нулевой бит FLAGS, что означает: "перед печатью дать пробел", а затем проверяется не "пробел" ли должен печататься. Если нет, то переход на PO-CHAR-3, а если да, то ведущий пробел не нужен и бит 0 FLAGS принудительно включается.

0B76 - 0B7E - PO-CHAR-3.

В регистре HL код печатаемого символа умножается на 8, а затем суммированием с базовым адресом таблицы символов определяется адрес, начиная с которого в ПЗУ расположены восемь битов, определяющих его конструкцию.

Подпрограмма "PRINT ALL CHARACTERS".

Эта подпрограмма служит для печати графических шаблонов символов размером 8x8. При входе сюда в регистровой паре DE содержится базовый адрес таблицы символов, в HL - адрес в дисплейном файле, в который помещается графическое изображение символа; пара BC содержит номер строки и столбца позиции печати. В-строка, С - столбец.

Процедура проверяет не достигнут ли конец строки. В зависимости от результата реализуется один из двух переходов на PR-ALL-1.

Далее проверяется куда идет печать. Если на принтер - вызывается процедура для копирования буфера принтера COPY-BUFF (0ECD), после чего прямым переходом мы идем к PR-ALL-1.

0B93 - 0BA3 - PR-ALL-1.

Если начинается новая строка, то вызовом PO-SCR (0C56) проверяется не надо ли делать "прокрутку" экрана.

Далее проверяется необходимость режима печати OVER по системной переменной P-FLAG. Если он включен, в регистре B выставляется FF, а если нет - 00 перед переходом на PR-ALL-2.

0BA4 - 0BB5 - PR-ALL-2.

Здесь проверяется необходимость режима печати INVERSE по системной переменной P-FLAG. Если он включен, в регистре C выставляется FF, а если нет - 00.

Подготавливая печать в аккумуляторе выставляется число 8 по числу строк в символе. Далее проверяется куда идет печать. Если на экран - переход на PR-ALL-3, а если на принтер - принудительно включается первый бит системной переменной FLAGS-2, что означает "буфер принтера не пустой".

0BB6 - PR-ALL-3.

Здесь только обмен содержимым между регистровыми парами DE и HL - вспомогательная операция.

0BB7 - 0BC0 - PR-ALL-4.

Из ПЗУ или ОЗУ принимается первая строчка шаблона символа (из восьми) и

обрабатывается в соответствии с установками OVER и INVERSE (определено в BC).

Если печать идет на принтер - переход на PO-ALL-6, а если на экран - на PR-ALL-5.

0BC1 - 0BD2 - PR-ALL-5.

Организует цикл по восьми строкам символа. Возвращается в PR-ALL-4 для печати второй и последующих строк (до восьмой).

После того, как весь символ напечатан, начинается обработка цветовых атрибутов, для этой цели вызывается PO-ATTR (0BDB).

Далее возврат.

0BD3 - 0BDA - PO-ALL-6.

Если печать выполняется на экран, то каждая следующая строка шаблона символа имеет адрес в дисплейном файле на 256 больше, чем предыдущая (это связано с особенностью формата экранного файла "Спектрума".) Просмотреть эту особенность и понять ее нетрудно простым экспериментом

```
FOR i=16384 TO 22527:
```

```
POKE i,255:
```

```
NEXT i
```

Кстати, этот эксперимент поможет Вам увидеть, что в области памяти экрана существуют три области, каждая по 2048 байтов.

Отслеживается такой шаг на 256 байтов командой INC D в процедуре PR-ALL-4. Когда же печать ведется на принтере, то этого делать не надо. Здесь приращение только на 32, что и выполняется увеличением содержимого регистра E на 20H за каждый проход.

Таким образом, эта процедура представляет собой путь обхода ненужной команды INC D, стоящей на выходе PR-ALL-4, если печать ведется на принтере.

Подпрограмма "SET ATTRIBUTE BYTE"

Рассмотрев печать графического изображения символа, перейдем к тому, как устанавливаются в оперативной памяти значения цветовых атрибутов экрана. В работе этих процедур активно используются системные переменные ATTR-T, MASK-T и P-FLAG.

0BDB - 0BF9 - PO-ATTR.

Сначала старший байт адреса позиции печати в дисплейной памяти делится на 8 для определения координаты знакоместа, а затем выполняется операция AND 03 для определения трети экрана, в которой выполняется печать. Вычисляется адрес, соответствующий заданному знакоместу в файле атрибутов и принимаются во внимание ATTR-T, MASK-T, P-FLAG.

Если бит 6 P-FLAG включен, то обрабатывается команда PAPER 9.

0BFA - 0C07 - PO-ATTR-1.

Обрабатывается ситуация INK 9, если бит 4 P-FLAG включен.

0C08 - 0C09 - PO-ATTR-2.

Это выходная процедура. В нужный адрес файла атрибутов заносится байт атрибутов знакоместа и выполняется возврат.

Подпрограмма "MESSAGE PRINTING".

Эта процедура служит для печати сообщений или токенов ключевых слов. Очень интересная процедура, которой Вы можете пользоваться достаточно широко, программируя в машинных кодах. При входе в нее в регистровой паре DE содержится базовый адрес таблицы, в которой размещены сообщения, а в аккумуляторе - номер сообщения в таблице.

0C0A - 0C0F - PO-MSG.

Здесь выполняется обнуление старшего байта адреса, находящегося на вершине машинного стека. Это подготовительная операция для того, чтобы в последующем

различать печать сообщений от печати токенов ключевых слов, ибо первые не должны требовать вставки перед ними ведущего пробела, а вторые - должны. Это станет ясным в процедуре PO-EACH. Заканчивается процедура переходом на PO-TABLE.

0C10 - 0C13 - PO-TOKENS.

Процедура работает при печати токенов. Здесь в регистровую пару DE вводится базовый адрес таблицы токенов - 0095.

0C14 - 0C21 - PO-TABLE.

Вызовом PO-SEARCH по номеру сообщения (токена) отыскивается адрес его размещения в таблице.

Переходом на PO-EACH выполняется печать тех сообщений и токенов, которые не требуют ведущего пробела, для прочих выполняется вызов PO-SAVE для печати пробела.

0C22 - 0C34 - PO-EACH.

Отсюда рекуррентно начинает печататься сообщение или токен символ за символом. Очередной печатаемый символ выставляется в аккумуляторе из DE. Если появляется символ, у которого включен 7-ой бит (символ больше 128), то это означает, что данный символ в сообщении или в токене последний. Его выключают, чтобы символ мог быть напечатан вызовом PO-SAVE (0C3B).

Тем не менее символ вновь восстанавливают из DE и вновь проверяют седьмой бит. Если он выключен, то символ не последний и идет переход к началу PO-EACH для печати очередного символа. В противном случае рассматривается вопрос о необходимости ведущего пробела.

Если печатается сообщение, в регистре D помещается 0, а если токен, то его номер в таблице токенов.

Проверяется код печатаемого символа. Если он равен 43 (символ \$), то переход на PO-TRSP.

Если он знак препинания, цифра и т.п., т.е. меньше чем 41 (код буквы "a"), то выполняется возврат в вызывающую процедуру, в противном случае - проход на PO-TRSP.

0C35 - 0C3A - PO-TRSP.

Процедура выполняет подготовку печати ведущего пробела. Проверяется что печатается. Если это текстовое сообщение или токен RND, INKEY\$ или PI, то ведущий пробел не нужен и выполняется возврат, иначе в аккумулятор загружается код "пробела".

Подпрограмма "PO-SAVE".

0C3B-0C40 - PO-SAVE.

Процедура исполняет печать символов рекуррентно. Она вызывает PRINT-A-1 (0010), которая вызывает PRINT-OUT и т.д. вплоть до самой PO-SAVE. Эта рекурсия прерывается только в PO-EACH или в PO-TRSP когда проходит последний символ (с включенным седьмым битом).

Подпрограмма "TABLE SEARCH".

При входе в нее в регистровой паре DE содержится базовый адрес таблицы, в которой размещены сообщения, а в аккумуляторе номер сообщения в таблице.

0C41 - 0C43 - PO-SEARCH.

Выполняет подготовительные операции для поиска.

0C44 - 0C55 - PO-STEP.

Проверяются байты, составляющие текст. По седьмому биту определяется конец каждой записи, отсчитывается номер нужной записи и в регистровой паре DE выставляется ее адрес.

Процедура выполняет еще одну важную операцию. Если перед печатью сообщения

(токена) может быть необходим пробел, то взводится флаг C, что необходимо для рассмотренной ранее процедуры PO-TABLE.

Подпрограмма "TEST FOR SCROLL".

Эта подпрограмма вызывается в тех случаях, когда может возникнуть необходимость в скроллинге экрана. Это происходит в трех случаях:

- при обработке символа CHR\$ 13 ("возврат каретки");
- при печати с указанием AT в нижней строке экрана (INPUT-строке);
- когда текущая строка печати заполнена и необходим переход к следующей строке.

При входе в эту подпрограмму в регистре В содержится номер текущей строки печати.

0C56 - 0C85 - PO-SCR.

Из процедуры выполняется немедленный выход, если печать должна идти не на экран, а на принтер.

В начале своей работы процедура выставляет на стеке адрес 0D09, поэтому при завершении работы по команде RET автоматически будет стартовать CL-SET, расположенная по этому адресу.

Если речь идет о печати в системное окно оператором INPUT AT (проверяется по нулевому биту TV-FLAG), то выполняется переход на PO-SCR-4.

Номер строки сравнивается с DF-SZ. Если он больше, то выход с переходом через адрес на стеке в CL-SET, а если меньше, то переход на REPORT-5 для печати сообщения.

При их равенстве встает вопрос о необходимости скроллинга.

Если компьютер находится в режиме автоматического скроллинга (проверяется по 4-му биту TV-FLAG) переход на PO-SCR-2.

Уменьшается на единицу счетчик строки и если достигнут ноль, переход на PO-SCR-3. В противном случае вызовом CHAN-OPEN (1601) открывается канал "K" и принудительно включается бит 4 в TV-FLAG. Напомним, что канал "K" осуществляет связь с клавиатурой. Возвратом выполняется переход в CL-SET.

0C86 - 0C87 - REPORT-5.

Вызывается процедура обработки ошибок ERROR-1 с кодом перехвата 04, что вызывает печать сообщения об ошибке "Out of screen".

0C88 - 0CD1 - PO-SCR-2.

Уменьшается на единицу содержимое системной переменной SCR-CT. Если достигнут ноль, то надо давать на экране запрос "scroll?". В противном случае работа продолжается переходом на PO-SCR-3.

Для печати сообщения открывается канал "K" вызовом CHAN-OPEN (1601) и вызывается процедура печати сообщений PO-MSG (0C0A).

Включается бит 5 TV-FLAG. Это является сигналом для очистки нижней части экрана после нажатия клавиши.

Включается бит 3 FLAGS (означает: "курсор - L").

Включается бит 5 FLAGS (означает: "Клавиша не нажата").

Вызывается процедура WAIT-KEY (15D4), ожидавшая нажатия клавиши.

После того, как клавиша нажата, она анализируется. В четырех случаях, когда Вы отказываетесь от скроллинга нажатием "N", "n", "BREAK" и "STOP", вызывается REPORT-D (0D00) для остановки и печати сообщения "BREAK-CONT repeats".

Нажатие любой другой клавиши означает продолжение скроллинга экрана.

Перед выходом на PO-SCR-3 открывается канал "S" вызовом CHAN-OPEN (1601). Напоминаем, что канал "S" осуществляет связь с экраном.

0CD2 - 0CEF - PO-SCR-3.

Теперь можно делать "скролинг". Это выполняется вызовом CL-SC-ALL (0DFE).

Для отыскания адреса в экранной памяти, соответствующего текущему номеру строки, вызывается CL-ADDR (0E9B).

0CF0 - 0CF7 - PO-SCR-3A.

Управляет цветовыми атрибутами нижних двух строк экрана при скроллинге.

0CF6 - 0CFF - здесь хранится текст сообщения "scroll?". Как и положено, последний байт, содержащий знак вопроса инвертирован.

0D00 - 0D1B - REPORT-D.

Вызывается процедура обработки ошибок ERROR-1 с кодом перехвата 0C, что вызывает печать сообщения "BREAK - CONT repeats".

0D02 - 0D1B - PO-SCR-4.

Сюда мы попадаем, если печать должна быть выполнена в нижней части экрана (в окне системных сообщений). При попытке использовать более двух строк переход на REPORT-5. Если скроллинг не нужен - возврат. В противном случае подготавливается возможность скроллинга. Вызывается TEMPS (0D4D) для того, чтобы работать с временными цветовыми атрибутами.

0D1C - 0D2C - PO-SCR-4A.

Определяется необходимость скроллинга только нижних двух строк или всего экрана. В первом случае - переход на PO-SCR-4B, во втором случае - прямой проход туда же. Разница в содержимом регистра В, где установлено количество строк, подлежащих скроллингу. В аккумуляторе же содержится количество шагов скроллинга.

0D2D - 0D4C - PO-SCR-4B.

Шаг за шагом вызывается CL-SCROLL (0E00) до тех пор, пока содержимое аккумулятора не уменьшится до нуля.

По окончании цикла выполняются финишные операции по установке значений системных переменных, вызов процедуры CL-SET (0DD9) и возврат.

Подпрограмма "TEMPORARY COLOR ITEMS".

Это одна из важнейших подпрограмм. Она используется в тех случаях, когда необходимо использовать постоянные атрибуты в качестве временных.

0D40 - 0D57 - TEMPS.

Здесь устанавливаются ATTR-T и MASK-T для нижней части экрана.

0D58 - 0D65 - TEMPS-1.

Здесь устанавливаются ATTR-T и MASK-T для главной части экрана.

0D65 - 0D6A - TEMPS-2.

Здесь устанавливается P-FLAG.

Подпрограмма "CLS COMMAND".

Подпрограмма предназначена для исполнения очистки экрана. Сначала выключаются все пиксели экрана, затем атрибуты экрана устанавливаются в соответствии с содержимым ATTR-P, а затем обрабатывается нижняя часть экрана.

0D6B - 0D6D - CLS.

Здесь только вызов CL-ALL (0DAF) для очистки дисплейного файла.

0D6E - 0D86 - CLS-LOWER.

Вызовом TEMPS (0D4D) устанавливаются цветовые атрибуты. В регистр В вводится из

DF-SZ количество строк в нижней части экрана. Для их очистки вызывается процедура CL-LINE (0E44). Решается вопрос об атрибутах этих строк. В принципе они берутся из ATTR-T, но для двух самых нижних строк они выставляются равными цвету бордюра из системной переменной BORDER.

Организуется внешний цикл по установке атрибутов в строках. Счетчик цикла - в регистре В.

0D87 - 0D88 - CLS-1.

Это вершина внутреннего цикла по установке атрибутов в позициях строки. Счетчик цикла - в регистре С.

0D89 - 0D8D - CLS-2.

Тело внутреннего цикла.

0D8E - 0D93 - CLS-3.

Завершение внешнего цикла и принудительная установка в DF-SZ числа 2, поскольку в нормальных условиях для окна системных сообщений отводятся две строки.

0D94 - 0D9F - CL-CHAN.

Выполняются финишные вспомогательные операции, а именно:

- открывается канал "К" и далее: 0BA0 - 0DAE - CL-CHAN-A.
- устанавливается в качестве выходного адреса 09F4 (PRINT-OUT) и в качестве входного - 10A8 (KEY-INPUT).

0DAF - 0DD8 - CL-ALL.

Открывает канал "S", устанавливает в качестве временных постоянные атрибуты экрана и выполняет очистку 24-х строк (вызовом CL-LINE (0E44) с установленным в регистре В числом строк, подлежащих очистке.) Продолжает работу в CL-SET.

0DD9 - 0DED - CL-SET.

При входе в эту процедуру в паре ВС хранятся номер строки экрана и номер позиции печати. Если печать идет на принтер, то только в регистре С содержится координата позиции печати.

Если печать идет на принтер, управление передается на CL-SET-2 а если в главную часть экрана - на CL-SET-1.

0DEE - 0DF3 - CL-SET-1.

Вызовом CL-ADDR (0E9B) формирует в HL адрес, соответствующий началу строки, номер которой содержится в В.

0DF4 - 0DFD - CL-SET-2.

В отличие от предыдущей процедуры здесь адрес в HL формируется не для экранного файла, а для буфера принтера.

Выход оформляется безусловным переходом на PO-STORE (0ADC) для запоминания координат позиции печати.

Подпрограмма "SCROLLING".

Количество строк, подлежащих скроллингу содержится в регистре В.

0DFE - 0DFF - CL-SC-ALL.

Это точка входа в процедуру после запроса "scroll?". Здесь в регистр В засылается 017H, что означает "скроллировать весь экран".

0E00 - 0E04 - CL-SCROLL.

Это главная точка входа. Вызовом CL-ADDR (0E9B) определяется адрес в дисплейном файле, соответствующий строке.

0E05 - 0E0C - CL-SCR-1.

Проблемой при скроллинге является то, что формат экрана в "Спектруме" разбит на три отдельных участка, в каждом по 8 экранных строк. Сложность состоит в переходе из трети в треть.

В этой процедуре выполняется переход на CL-SCR-3, если скроллингу подлежат более чем 8 строк, т.е. предполагается такой переход. Если нет, то прямой проход на CL-SCR-2.

0E0D - 0E18 - CL-SCR-2.

Процедура обеспечивает переход из трети в треть посредством переброски информации в оперативной памяти командой LDIR.

0E19 - 0E43 - CL-SCR-3.

Здесь выполняется скроллинг в пределах одной трети экрана. Сначала передвигается черно-белая графика, а затем атрибуты, заканчивается процедура проходом на CL-LINE для очистки нижней строки экрана.

Подпрограмма "CLEAR LINES".

Эта подпрограмма выполняет очистку n нижних строк экрана, где n - это содержимое регистра В.

0E44 - 0E49 - CL-LINE.

Вызовом CL-ADDR (0E9B) в паре HL формируется адрес, соответствующий номеру строки.

0E4A - 0E4C - CL-LINE-1.

Подготовка к очистке пиксельных строк.

0E4D - 0E7F - CL-LINE-2.

Очищаются строки в следующем порядке: первая пиксельная строка первого символа; первая пиксельная строка всех символов ряда, переход на следующий ряд; если происходит переход между третями экрана - переход на CL-LINE-2; очистка первой строки второго и последующего рядов; переход на CL-LINE-1 для очистки в том же порядке второй пиксельной строки и последующих до восьми; очистка области атрибутов.

Атрибуты в главной части экрана берутся из ATTR-P, а для нижних двух строк устанавливаются по цвету бордюра - из BORDER.

0E80 - 0E87 - CL-LINE-3.

Завершает установку атрибутов и выполняет возврат.

Подпрограмма "CL-ATTR".

0E88 - 0E9A - CL-ATTR.

Эта процедура выполняет две функции:

- по заданному адресу графики в дисплейном файле ищет и возвращает в регистровой паре DE соответствующий ему адрес в файле атрибутов;
- по заданному номеру строки, содержащемуся в регистре В, возвращает в ВС количество символов, которое имеется от начала этой строки до конца экрана.

Подпрограмма "CL-ADDR".

0E9B - 0EAB - CL-ADDR.

Процедура возвращает в паре HL адрес в дисплейном файле по заданному в регистре В номеру строки.

Подпрограмма "COPY COMMAND".

Суть команды COPY состоит в копировании графики экрана на ZX-PRINTER.

0EAC - 0EB1 - COPY.

Копирование производится построчно (176 строк). В этой процедуре настраиваются регистры перед входом в цикл.

0EB2 - 0EC8 - COPY-1.

Организуется цикл для 176 строк и в цикле вызывается COPY-LINE (0EF4) для копирования строк.

0EC9 - 0ECC - COPY-2.

Окончание цикла и переход на финишную процедуру COPY-END.

Подпрограмма "COPY-BUFF".

Подпрограмма вызывается в тех случаях, когда необходимо переслать содержимое буфера принтера на ZX-принтер.

0ECD - 0ED2 - COPY-BUFF.

Подготовка работы.

0ED3 - 0ED9 - COPY-3.

Организуется цикл, который в раз обращается к COPY-LINE (0ED3) для копирования информации из буфера на принтер.

0EDA - 0EDE - COPY-END.

Выключается двигатель принтера.

Подпрограмма "CLEAR PRINTER BUFFER".

0EDF - 0EE6 - CLEAR-PRB.

Подготовка к очистке буфера принтера.

0EE7 - 0EF4 - PRB-BYTES.

Последовательная байт за байтом очистка буфера. Выход переходом на CL-SET и далее PO-STORE.

Подпрограмма "COPY-LINE".

0EF4 - 0EFC - COPY-LINE.

При входе в процедуру регистровая пара HL содержит начальный адрес 32-ух байтов, составляющих пиксельную строку, а регистр B содержит номер этой пиксельной строки.

Из аккумулятора на порт FB выдается 00 для включения мотора. Если до конца печати осталось две строчки, на порт FB выдается 02 -сигнал замедлить мотор.

0EFD - 0F0B - COPY-L-1.

Вызовом BREAK-KEY (1F54) проверяется нажатие клавиши BREAK и если она не нажата - продолжение работы переходом на COPY-L-2.

Если нажата, выдается 04 на порт FB (остановить мотор), очищается буфер принтера вызовом CLEAR-PRB (0EDF) и работа прерывается вызовом ERROR-1 (0008) с кодом перехвата 0C (BREAK - CONT repeats).

0F0C - 0F13 - COPY-L-2.

Проверка опросом порта FB на наличие принтера. Если принтер не подключен, немедленный возврат.

0F14 - 0F17 - COPY-L-3.

Начало внешнего цикла. Здесь в регистр E засылается очередной байт из буфера принтера и устанавливается параметр цикла в регистре В - 8 по числу графических строчек в печатной строке

0F18 - 0F1D - COPY-L-4.

Начало внутреннего цикла. Здесь печатаемые байты раскладываются на биты. В результате этой раскладки очередной печатаемый бит выставляется старшим битом в аккумуляторе.

0F1E - 0F2B - COPY-L-5.

В цикле выполняется печать. Печать выполняется побитно выдачей содержимого аккумулятора на порт FD. При этом младшие биты управляют мотором, а самой печатью (есть точка/ нет точки) управляет бит 7.

(Продолжение следует).

FORUM

Как всегда принося глубочайшие извинения за то, что мы сейчас совершенно не в состоянии охватить и двадцатую часть поступавшей корреспонденции, мы рассмотрим лишь несколько писем из тех, которые представляют на наш взгляд массовый интерес.

Автостарт программ в машинных кодах.

Никольский А.Л. из города Клина спрашивает о том, как сделать программу в машинных кодах автостартующей без Бейсик-заголовка.

Задача довольно полезная. Это во-первых подход к защите, поскольку программу, записанную в машинном коде трудно прервать, а во-вторых такая задача встает например перед теми, кто занимается переделкой программ под игровые автоматы.

Ситуация, когда программа накладывается на область стека в данном случае неинтересна, поскольку читатель знает, что в этом случае в те ячейки стековой области, которые определяют куда выполняется переход при обработке ошибок, может быть записан нужный ему адрес.

В этом случае может быть реализован например такой подход, связанный с присоединением БЕЙСИК-области к машинному коду.

Предположим, что Вам надо автостартовать программу, записанную в машинном коде, начинающуюся с адреса 26500, занимающую 1500 байтов и стартовую с 29000.

Загрузите программу в компьютер и наберите следующие строки на БЕЙСИКе:

```
10 SAVE "myprog" CODE 23552,6448
20 RANDOMIZE USR 29000.
```

Эта программа делает следующее. При исполнении строки 10 системные переменные настраиваются так, что в качестве следующей исполняемой строки в них запоминается строка номер 20.

Строка 10 выгружает на ленту в виде единого блока кодов и системные переменные и область Бейсика и Ваш машинный код.

Если теперь Вы загрузите этот блок машинного кода командой `LOAD "" CODE`, то по системным переменным пройдет переход к строке 20, а она стартует Вашу программу. Вы даже и не заметите, что работал Бейсик.

Конечно это только идея и дальше Вы можете ее развивать как угодно, организовав в стартовавшем машинном коде старт других блоков, перемещение кода и т.п. Кстати, заметим, что при работе таким методом перекрытие кодом области стека является наоборот критическим, т.к. после загрузки не стартует нормальный Бейсик, а происходит сброс, вот и приходится делать загрузку в одно место, а потом, когда программа стартовала, перемещать ее через `LDIR` или `LDDR` в ту область, где ей на самом деле предстоит работать.

Декодер радиопередачи.

К нам обратился читатель из г.Белозерска Брестской области Давиденко А.В. с сообщением, что он принимает по УКВ трансляцию программного обеспечения из Польши. По всей видимости эта информация может быть полезна многим нашим читателям, проживающим в западных областях. Трудность состоит в том, что программы закодированы. Для их декодирования служит "Радиодекодер ПЛЮС", а взять его негде.

Ниже мы приводим программу из журнала "БАЛТЕК" N5, 1989г. "Decoder Transmisji RadioweJ Plus" (Сокращенно DTR+).

Мы не очень сильны в польском языке и приводим статью в кратком изложении.

1. Надо набрать Бейсик-загрузчик (строки 100 - 120) листинга номер 1.
2. Выгрузить загрузчик на ленту командой `SAVE "DTR+" LINE 1`
3. Набрать листинг номер 2 и, запустив его, выгрузить на ленту блок машинных кодов.
4. Выгруженные загрузчик и блок маш. кодов представляют собой готовую программу, которая стартует после загрузки автоматически. Она предложит Вам небольшое меню.

LOAD - считывание с ленты закодированной программы (она должна быть записана на магнитофон из эфира).

SAVE - выгрузка на ленту программы в декодированном виде.

RESET - выход.

Приводится адрес для справок:

Rozglosnia Harcerska

00-950 Warszawa

ul. M. Kopernickiej 6

с пометкой "RADIOKOMPUTER".

Сообщается также, что передачи студии "Радиокомпьютер" ведутся на УКВ и на средних волнах по четвергам 16.00-16.20 и по вторникам 22.50 - 24.00. Трансляция программных блоков 23.25 - 24.00. (По всей видимости время варшавское).

Листинг-1.

```
100 CLEAR 64000: LOAD "DTR+"CODE
110 CLS: PRINT ,,"  DECODER TRANSMISJI RADIOWEJ",,"  L-LOAD",,"  S-SAVE",,"  BREAK-STOP",,"
    R-RESET"
120 RANDOMIZE USR 64940
```

Листинг-2.

```
150 CLEAR 64000: LET S=0: FOR a=64940 TO 65534: READ b: POKE a,b: LET s=s+b: NEXT a
160 IF s=69549 THEN SAVE "DTR+" CODE 64940,595: STOP
170 PRINT "Ошибка ввода данных": STOP
200 DATA 1, 16, 13, 205, 10, 32, 17, 196, 253, 205, 55, 254, 62, 16, 50, 71, 80, 62, 8, 50, 39, 87, 24, 48, 62
210 DATA 83, 90, 79, 75, 60, 13, 13, 32, 32, 32, 32, 32, 32, 83, 119, 105, 101, 98, 111, 100, 122, 105, 110, 32
220 DATA 32, 49, 57, 56, 56, 114, 174, 194, 9, 204, 9, 219, 9, 237, 9, 0, 0, 0, 0, 0, 0, 253, 54, 56
230 DATA 0, 253, 203, 48, 222, 49, 229, 253, 62, 56, 50, 130, 88, 50, 98, 88, 205, 91, 254, 245, 205, 65, 254,
    241, 254
240 DATA 82, 202, 0, 0, 254, 76, 40, 76, 254, 83, 204, 156, 255, 24, 221, 1, 21, 0, 205, 10, 32, 24, 13, 203, 39
250 DATA 60, 33, 228, 253, 35, 61, 32, 252, 94, 35, 86, 62, 32, 215, 26, 19, 203, 127, 40, 249, 203, 191, 215,
    201, 1
260 DATA 21, 0, 17, 32, 0, 24, 6, 1, 8, 0, 17, 159, 1, 205, 10, 32, 62, 32, 215, 27, 122, 179, 32, 248, 201
270 DATA 33, 8, 92, 54, 0, 126, 254, 0, 40, 251, 201, 205, 73, 254, 253, 54, 56, 0, 62, 248, 250, 98, 88, 243, 17
280 DATA 5, 0, 33, 239, 253, 14, 101, 205, 231, 254, 33, 69, 95, 237, 91, 241, 253, 25, 218, 144, 255, 33, 248,
    92, 14
290 DATA 86, 205, 231, 254, 1, 8, 0, 205, 10, 32, 253, 54, 57, 13, 33, 248, 92, 34, 237, 253, 229, 237, 91, 239, 253
300 DATA 167, 237, 82, 40, 49, 253, 53, 57, 202, 144, 255, 225, 229, 126, 254, 4, 210, 144, 255, 205, 39, 254,
    225, 6, 10
310 DATA 35, 126, 254, 32, 56, 4, 254, 128, 56, 2, 62, 32, 215, 16, 241, 62, 13, 215, 35, 94, 35, 66, 1, 5, 0
320 DATA 9, 25, 24, 194, 253, 53, 56, 251, 62, 7, 211, 254, 195, 252, 253, 217, 49, 227, 253, 22, 11, 46, 50, 62, 25
330 DATA 38, 1, 205, 76, 253, 48, 243, 45, 32, 244, 62, 22, 22, 13, 205, 93, 255, 217, 185, 317, 32, 228, 22, 14, 62
340 DATA 19, 24, 2, 62, 17, 205, 93, 255, 217, 119, 169, 79, 35, 27, 122, 179, 217, 32, 240, 217, 175, 185, 32,
    110, 201
350 DATA 71, 16, 254, 203, 18, 63, 203, 26, 63, 8, 203, 122, 32, 1, 122, 211, 254, 62, 127, 219, 254, 31, 48, 94, 1
360 DATA 64, 37, 219, 254, 161, 167, 32, 4, 16, 246, 34, 67, 95, 201, 36, 3, 205, 35, 255, 62, 28, 37, 32, 248, 96
370 DATA 205, 35, 255, 120, 132, 254, 44, 201, 205, 74, 255, 48, 41, 62, 25, 205, 84, 255, 56, 249, 63, 25, 205,
    84, 255
380 DATA 111, 62, 26, 205, 84, 255, 189, 38, 3, 48, 1, 37, 62, 24, 46, 128, 205, 76, 255, 203, 29, 62, 25, 38, 3
390 DATA 48, 245, 125, 201, 203, 66, 194, 232, 254, 205, 73, 254, 17, 37, 21, 205, 31, 254, 195, 223, 254, 253,
    203, 56, 126
400 DATA 203, 252, 253, 237, 115, 61, 92, 62, 248, 50, 130, 88, 17, 162, 9, 205, 31, 254, 205, 91, 254, 205, 65,
    254, 33
410 DATA 248, 92, 34, 237, 253, 237, 91, 239, 253, 167, 237, 82, 202, 252, 253, 221, 42, 237, 253, 205, 246, 255,
    17, 17, 0
420 DATA 62, 0, 205, 194, 4, 231, 42, 237, 253, 205, 246, 255, 221, 94, 11, 221, 86, 12, 213, 1, 17, 0, 231, 9, 221
430 DATA 229, 62, 255, 205, 194, 4, 225, 209, 25, 24, 197, 17, 255, 255, 122, 179, 200, 27, 24, 250
```

К вопросу о совместимости.

Обсуждая в наших выпусках вопросы совместимости различных моделей отечественных аналогов "Спектрума", мы несколько раз касались и компьютера "Дубна".

Мы получили исчерпывающую информацию от главного инженера завода "ТЕНЗОР", выпускающего этот широко известный компьютер.

Некоторая несовместимость вызвана тем, что компьютер был разработан в 1987-88

г.г., а в то время микросхемы ОЗУ с достаточным быстродействием представляли значительный дефицит. В связи с этим разработчики были вынуждены применить микропроцессор с тактовой частотой ниже, чем у фирменного прототипа и, как следствие, им пришлось "перетрясти" процедуры ПЗУ, обслуживающие ввод/вывод с магнитофона, чтобы обеспечить совместимость программного обеспечения при загрузке с ленты.

В результате не загружаются программы, имеющие нестандартные загрузчики. По этой же причине не работают некоторые копировщики, такие как TF-COPY и COPY-86M.

Завод рекомендует использовать могучую копирующую программу COPY-COPY, а для тех, кто умеет читать машинный код сосредоточить усилия на замену спецзагрузчиков в программах на стандартный.

Понимая в то же время, что эта особенность компьютера создает определенные неудобства, завод подготовил к выпуску новую модель "Дубна-М". В нем будет стандартная частота, а также встроенные интерфейсы - дисковод и принтера (последовательный и параллельный). Разработчики надеются, что эта модель удовлетворит самые взыскательные вкусы.

"ИНФОРКОМ" не может не согласиться с рекомендацией шире использовать копировщик COPY-COPY (второе название - PIRATE 02). Дело в том, что эта программа предоставляет пользователю такой спектр дополнительных возможностей, которые нельзя не использовать. В своей разработке "Большие возможности Вашего Спектрума" мы два года назад дали инструкцию по работе с этой программой. Учитывая, что эта наша разработка многократно копировалась и переиздавалась разными организациями, она должна быть широко распространена, но на тот случай, что кто-то из наших читателей ее не имеет, мы приведем эту инструкцию чуть ниже.

Мы также должны сказать, что судя по письмам наших читателей программ, не загружающихся в "Дубну" не так много - около двух десятков. Правда надо признать, что есть психологический эффект, согласно которому неискушенный пользователь всегда уверен, что самые лучшие программы - это именно те, которые у него не идут и поэтому мы просим тех, кто снял спецзагрузчики и адаптировал программы под "Дубну" дать рекомендации о том как конкретно это сделать. Остальным же будет очень полезно покопаться в этих загрузчиках, освоив на этой интересной задаче программирование в машинных кодах, благо исходной информацией мы обеспечиваем (см. раздел "Секреты ПЗУ" в ZX-РЕВЮ и наш трехтомник для начинающих осваивать программирование в машинных кодах).

Дополнительно сообщим, что в 92-ом году в ZX-РЕВЮ мы напечатаем несколько книг, имеющих большую ценность для тех, кто собирается осваивать машинный код или уже освоил, но хочет эффективно применять.

COPY-COPY

Все команды являются ключевыми словами "Спектрума" и потому не набираются по буквам. Они требуют завершения нажатием клавиши ENTER.

LOAD - выполнить загрузку очередной записи (очередного блока).

LOAD n to m - выполнить загрузку записей, начиная с номера n до номера m. При этом ранее существовавшие записи с номером n и выше будут уничтожены, т.е. эту команду применяют и для очистки памяти от ненужных записей. Так, LOAD 1 уничтожит все ранее существовавшие записи и подготовит компьютер к загрузке новых.

Возможные вариации этой команды:

LOAD n TO

LOAD TO m

SAVE - выполнение выгрузки записи (записей) на ленту. Может иметь форму:

SAVE n to m

SAVE n TO

SAVE TO m

SAVE n TO m STEP k

Параметр k после оператора STEP задает величину паузы между блоками при

выгрузке (в секундах).

VERIFY - проверка выгруженных записей. Имеет все те же формы, что и LOAD.

Программа позволяет загружать блоки в заданные адреса. Это часто применяется для последующего просмотра машинного кода и внесения в него изменений. На этом нередко основывается внесение изменений в игровые программы.

LOAD AT addr - загрузка блока, начиная с заданного адреса addr. Так, например, LOAD AT 16364 - загрузка в экранную память.

LIST addr - выдача на экран содержимого памяти, начиная с заданного адреса (просмотр машинного кода).

POKE addr, n - поместить в заданный адрес байт n.

И, наконец, самая мощная команда - COPY.

COPY 16384 позволяет копировать блоки длиной до 49K. Есть программы, имеющие цельный блок длиной порядка 49к и не поддающиеся компрессии. Таковы, например FIRELORD или, скажем URIDIUM фирмы Hewson Consultants. в режиме COPY 16384 программа удаляет саму себя из памяти и оставляет лишь несколько байтов, служащих для выгрузки загруженного блока. Выгрузка выполняется нажатием CAPS SHIFT. Возможна только однократная выгрузка и после нее копировщик следует перезагрузить.

В нашей почте сегодня есть несколько рекомендаций для повышения совместимости компьютеров "Ленинград-1".

Сизой В.М. и Лавринов В.В. из г. Днепропетровска сообщают нам, что на этой модели компьютера при работе с рядом программ может мигать бордюр, при этом раздается треск в динамике. Это, например, программа "IKARI WARRIORS" и другие. Это обусловлено тем, что все выходные порты от 0 до 255 воспринимаются микросхемой DD39 как порт 254 и в тех программах, где есть обращение к другим портам, возникают помехи.

В свою очередь входные порты воспринимаются:

нечетные - как порт 31 (джойстик)

четные - как порт 254 (магнитофон, клавиатура).

Предлагаемая ими схемная доработка (рис. 1) позволяет сделать так, что все четные выходные порты, начиная со 128-го, воспринимаются как порт 254, а входные порты, имеющие в обоих адресах A0 и A5 единицы, воспринимаются как несуществующие. Для работы программы "ARKANOID" в схему введен синхросигнал с 8DD8. Благодаря этому программа работает и с клавиатурой и с джойстиком.

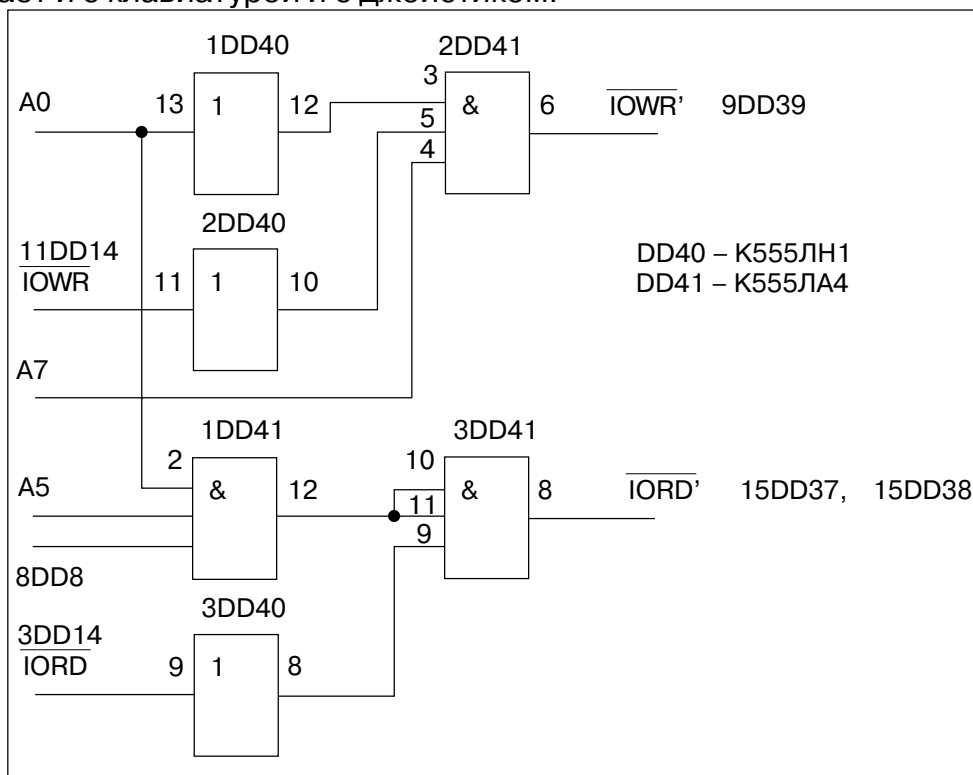


Рис. 1

(О проблемах программы ARKANOID мы уже писали в ZX-РЕВЮ - "ИНФОРКОМ").

Полезными советами делится с владельцами компьютера "Ленинград-1" и другой наш читатель из Львовской области Кондачков С.Н.

1. Он пишет, что подключить этот компьютер к цветным телевизорам с модулем МЦ-2, МЦ-3 весьма трудно. Это связано с тем, что гасящие строчные импульсы данного компьютера очень коротки по длительности. Парировать этот недостаток можно добавив следующий узел (Рис. 2):

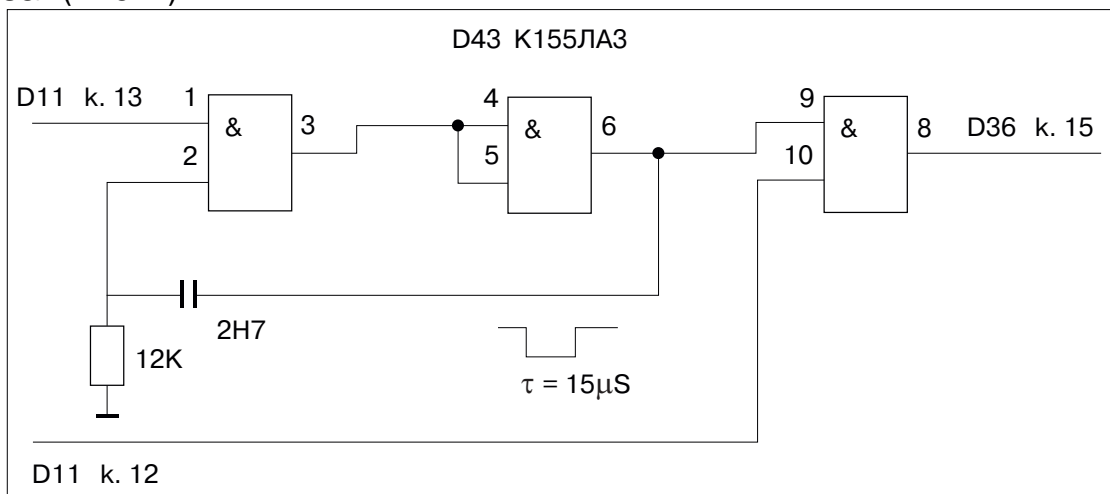


Рис.2

2. При подключении джойстика с нормально разомкнутыми контактами не загружаются некоторые программы (ANDROID, HOT SHOT и другие). Причина опять же в очень упрощенной дешифрации входного порта джойстика. Кроме того, в некоторых программах джойстик не работает (WEST BANK, SPLAT).

Компьютер можно доработать согласно следующей схеме (Рис. 3):

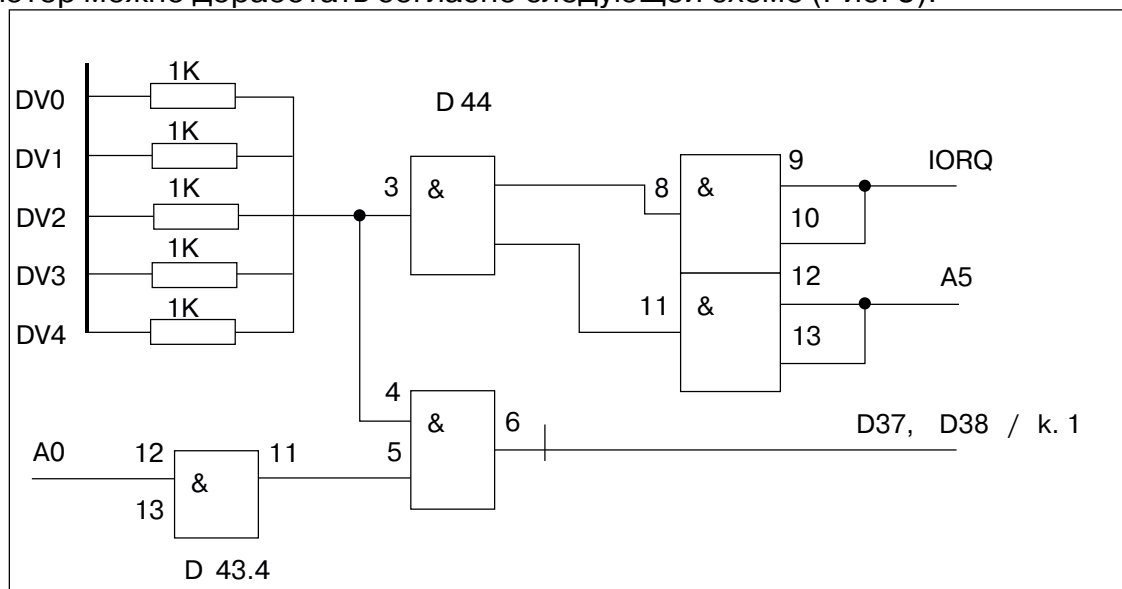


Рис.3

Недостатки устраняются полностью.

3. В некоторых играх при нажатии на кнопку "Огонь" в такт с выстрелом очень неприятно мерцает бордюр. И это связано с грубой дешифрацией порта. Предлагаемая доработка заключается в подключении двух диодов и одного резистора (Рис. 4):

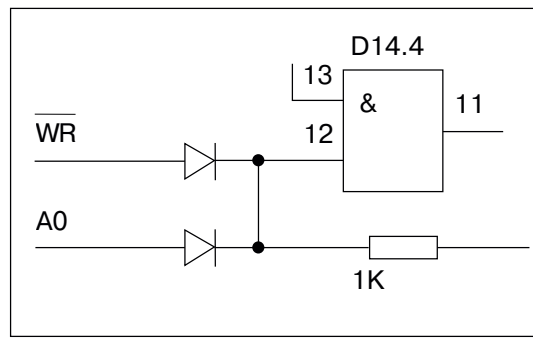


Рис.4

"ИНФОРКОМ" благодарит от своего имени и от имени многочисленных читателей всех друзей, приславших свои материалы в раздел "форум" и мы очень надеемся, что они будут полезны людям, разбросанным по просторам нашей страны и сражающимся со своим компьютером один на один.

Организация клубов.

Мы искренне были бы рады сделать все от нас зависящее, чтобы упростить сложный путь в компьютеризацию, сплотить любителей этого замечательного компьютера, и давно ждали писем от каких читателей об организации клубов любителей "Спектрума".

Наконец-то такие письма пришли.

1. Организуется клуб в городе Львове. Имеется договоренность с одним из Домов культуры. В группе организаторов есть конструкторы, разрабатывавшие "Львовский" вариант. Заинтересованных жителей Львова и области, а также Тернопольской и Иваново-Франковской областей просят обращаться по адресу:

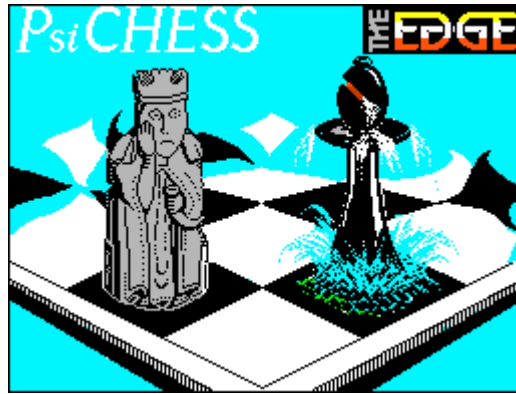
290066, г. Львов-66, а/я 8175 Барышневу Александру Борисовичу

2. Действует детский клуб любителей компьютера ZX-Spectrum в г.Одессе и его участники хотели бы переписываться с другими клубами нашей страны и, по возможности, с зарубежными.

Контактный адрес:

270114, Одесса а/я 292, клуб. Руководитель Макаров А.В.

PSI CHESS



Программу PSI-CHESS мы включили в сборник "Коллекция ZX-РЕВЮ" и сегодня даем ее полное фирменное описание. Программа достаточно трудна для самостоятельного разбора, поскольку ее система управления очень насыщена и надо считать это описание необходимым для нормальной работы с ней.

Еще раз особо подчеркиваем, что данное описание - фирменное. Не со всеми содержащимися в нем утверждениями мы согласны. Так, например, мы проводили соревнования между программами PSI-CHESS, I.S.CHESS, CHESS (PSION), COLOSSUS CHESS 4.0 и SUPERCHESS 3.5.

Эти партии позволяют сделать удивительные открытия и ставят под сомнение расхожие мнения о силе шахматных программ. Мы не будем развивать эту тему дальше, оставив удовольствие заняться исследованиями Вам самим. Попробуйте, но, правда, Вам придется решить вопрос о том, как задать уровень игры в разных программах, чтобы они примерно соответствовали по силе друг другу. Мы в испытаниях устанавливали уровни таким образом, чтобы программы думали примерно одинаковое время, рассчитывая в среднем по два часа на партию.

Вступление.

PSI-CHESS - наиболее совершенная шахматная программа написанная для микрокомпьютеров. Она не только может воспроизводить трехмерное изображение игровой доски и фигур, но и играет в силу, к которой приближаются очень немногие программисты.

PSI-CHESS понимает абсолютно все правила шахматной игры.

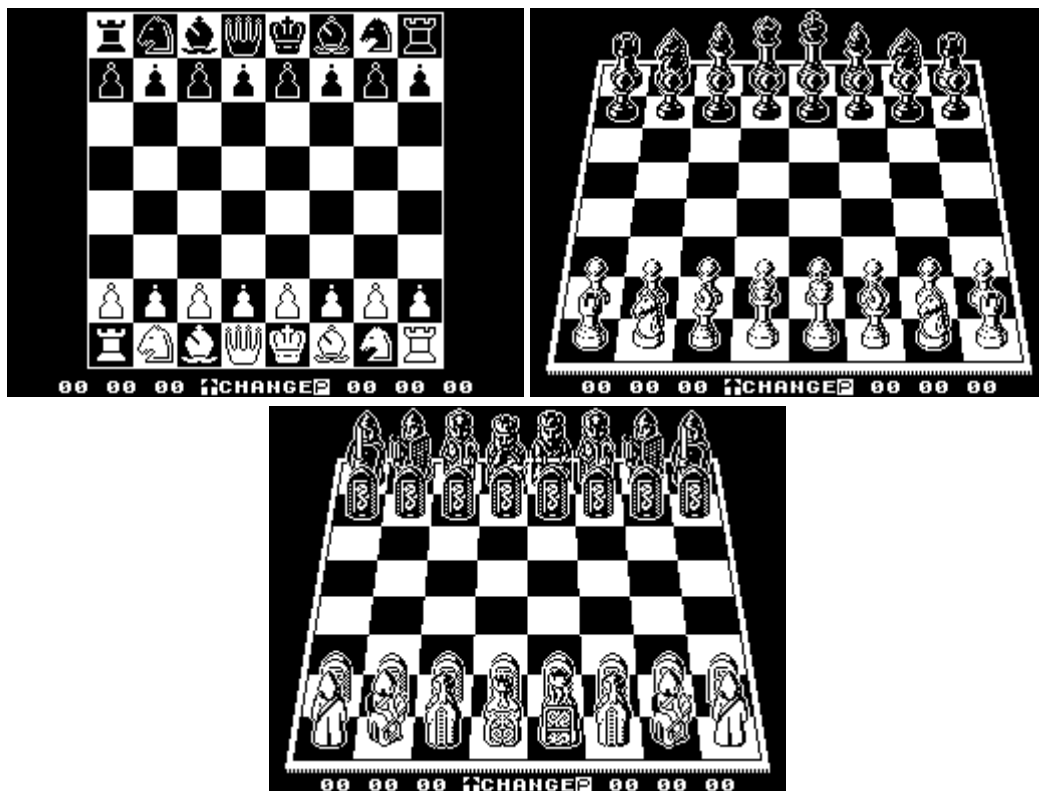
Когда программа загружена впервые, Вы попадаете в режим трехмерной графики, предполагается, что Вы играете белыми против компьютера, уровень силы игры компьютера установлен в A1 (ответный ход выполняется немедленно).

В нижней части экрана Вы видите слово "CHANGE" ("изменять"). Оно означает, что в этом режиме Вы можете переключать экраны, менять исходную настройку, менять режимы:

- Scoresheet (таблица ходов)
- Position (позиция)
- Game (игра).

Рассмотрим возможные экраны: Нажмите S и одновременно 1. Вы попадете в режим "таблицы ходов". Здесь приводится не только запись партии, но и изображаются основные параметры, на которые настроена игра. Так, например, в верхней части экрана Вы увидите кто играет белыми и кто черными, а также уровень сложности - "A1" (об этом чуть позже).

Теперь нажмите S и одновременно 2. Вы увидите двумерное изображение доски. Фигуры изображены символически. К этому режиму мы также вернемся несколько позже.



Нажатие S и 3 переводит Вас в исходный трехмерный экран с классическими фигурами.

Нажмите S и 4 - Вы увидите трехмерное изображение доски с фигурами, выполненными в стиле старинных шахмат.

Теперь попробуйте вращать шахматную доску, это выполняется нажатием клавиш O (буква O) и одной из цифр - от 1 до 4.

Игра.

Чтобы сразу начать игру, Вам надо нажать "M" и "1" одновременно. Вы увидите, что на дисплее появилась буква "G" ("GAME" - игра). Программа ждет Вашего первого хода. Вы также видите, что пошли Ваши часы (они слева).

Ввод ходов очень простой и может выполняться двумя способами. Это можно делать традиционной алгебраической нотацией или пользоваться курсором на экране для выбора фигуры, которой Вы собираетесь ходить и выбора поля, на которое Вы хотите пойти.

В алгебраическом методе Вы должны набрать букву и цифру, указывающие на то поле, на которое Вы хотите пойти. Вертикали пронумерованы от "A" до "H", а горизонтали от "1" до "8".

Наберите "E2-E4", а если бы играли черными, то могли бы дать "E7-E5" (о смене цвета мы будем говорить ниже).

По второму методу Вы можете работать с курсором. Это выполняется нажатием "J" и цифры "1", "2" или "3".

J1 - подразумевается имитация левого Синклер-джойстика;

J2 - правого Синклер-джойстика;

J3 - работа с Кемпстон-джойстиком.

Это не означает, что вам необходимо иметь джойстик. Синклер-джойстик вполне замещается клавишами:

- | | |
|------------|------------|
| J1 | J2 |
| 1 - влево | 6 - влево |
| 2 - вправо | 7 - вправо |
| 3 - вниз | 8 - вниз |
| 4 - вверх | 9 - вверх |
| 5 - выбор | 10 - выбор |

В этом режиме работать крайне просто. На экране появляется стрелочный курсор. Его надо поместить на нижнюю часть шахматной фигуры и нажать клавишу "выбор". Теперь переместите курсор на поле, на которое хотите пойти и опять нажмите "выбор", фигура передвинется, если ход, которого Вы требуете, разрешен. Если же ход запрещенный, то в нижней части экрана появится сообщение: "Illegal move"

Если Вы хотите перейти к нормальному способу управления фигурами, то в любой момент можете нажать "K".

Закончив игру, если хотите, можете сыграть еще раз. Для этого нажмите I, на экране появится слово "CHANGE". Теперь нажмите "I" и "1" вместе. Фигуры будут вновь расставлены на доске. Теперь опять дайте "I", чтобы появилось слово "CHANGE", а затем "M" и "1" одновременно, как и раньше.

Детальное описание программы.

PSI-CHESS имеет огромное количество возможностей и основная прелесть ее состоит в том, что Вы можете ими не пользоваться, если хотите просто играть в шахматы или заниматься их изучением.

Как Вы уже видели выше, играть очень просто, и по мере усиления Вашей игры, Вы можете повышать уровень силы игры программы.

Изменение параметров игры.

Когда на экране есть слово "CHANGE", Вы можете вносить изменения в настройку игры. Полный список возможностей программы для быстрого освоения Вы найдете в конце данного руководства.

Изменение цветов экрана.

Это делается нажатием "X", "C" или "B" с одной из клавиш от 0 до 9. Эти клавиши меняют "Спектрумовские" цвета INK, PAPER и BORDER.

Звук.

Вы можете получать звуковые эффекты при нажатии клавиш. Для этого служит клавиша "Z" с клавишами от "1" до "8". "Z" и "1" дают тихий короткий звук, а "Z" и "9" - вызывают более громкий сигнал.

Вы поймете, что это свойство программы неоценимо, когда будете играть на более высоких уровнях, т. е. когда компьютер думает по несколько минут над каждым ходом. Звук будет сигнализировать в этом случае, что сейчас Ваша очередь ходить.

Режимы экрана.

Об этом мы уже говорили, это клавиша "S" и клавиша от "1" до "4" одновременно.

Настройка игры.

Режим "S" - таблица ходов.

Настройку программы, по-видимому, лучше всего выполнять, когда на экране изображена таблица ходов. Это режим вызывается нажатием "S"+"1" в тот момент, когда горит слово "CHANGE". Если это не так, то перейдите в режим "CHANGE" нажатием "I".

В таблице Вы увидите имена игроков, список последних 15 ходов, уровень силы игры и часы.

Выбор цвета фигур - черные или белые.

Здесь у Вас есть 4 возможных выбора.

1. Пользователь (USER) - белые, компьютер - черные.
2. Пользователь - черные, компьютер - белые.
3. Компьютер играет и за черных и за белых.
4. Два человека играют друг против друга.

Выбор осуществляется после перехода в настроечный режим совместным нажатием

"M"+"3". В нижней части экрана загорается S. Слова "CHANGE" в это время на экране быть не должно, поэтому если оно есть, нажмите "I", чтобы режим работал нормально.

Находясь в режиме S, можете переключить играющих с помощью клавиши "T". "T"+"1" переключает того, кто играет белыми, а "T"+"2" - того, кто играет черными. "T"+"3" - переключает того, кто делает следующий ход.

На экране тот, чей ход обозначается надписью "to move".

Обратите внимание, что хотя мы и использовали для этой настройки экран, изображающий таблицу ходов, тем не менее все это же самое можно сделать и в любом другом экране, хотя подтверждения правильности своих действий Вы там не увидите.

Настройка часов.

Это также выполняется в режиме S, но находиться в экране таблицы ходов не обязательно.

Часы начинают идти как только начинается игра и уже не останавливаются и не устанавливаются в исходное положение, даже если Вы уже играете вторую, третью партию и т.д. Тем не менее, если Вы хотите обнулить показания часов перед началом игры или установить их на заданное время, это можно сделать следующим образом:

1) Установка на нуль - клавиши "T"+"4".

2) Установка заданного времени:

В режиме "S" для белых фигур нажмите "W" + цифру от "1" до "5" для уменьшения показаний часов в разрядах или "W" + цифру от "6" до "0" для увеличения показаний в разрядах.

Для черных процедура такая же, но вместо "W" надо нажимать "R".

Клавиша "5" - секунды;

клавиша "4" - десятки секунд;

клавиша "3" - минуты;

клавиша "8" - десятки минут;

клавиша "1" - часы.

Аналогично работают клавиши "6"-"0".

Обратите внимание, что изменение показаний в секундах и минутах может влиять на показания в минутах и часах соответственно.

Используя часы, Вы можете задавать себе время на обдумывание больше или меньше, чем оппоненту.

Играя в режиме "BLITZ"(Блиц), Вы можете устанавливать каждому игроку по 5 минут или давать одному из них фору.

Уровни игры.

Изменение уровня игры также выполняется в режиме "S" (вход в него через "M"+"3"). Настройка уровня видна только в экране, изображающем таблицу ходов, хотя изменять уровень можно и в любом другом экране.

PSI-CHESS имеет 7 уровней для обычной игры, 6 уровней для игры по переписке и три специальных уровня, по крайней мере два из которых впервые введены в программу для бытового компьютера.

Когда программа начинает работу, она настроена на уровень A1. Чтобы перейти к A2, нажните "A" и "2" и т.д.

Аналогично выбирается уровень игры по переписке "C1"... "C6".

Разница в силе игры зависит от того, сколько времени программа расходует на обдумывание хода, от этого зависит в первую очередь глубина просмотра ходов вперед, а также степень "изоэкономности" тех процедур, к которым программа имеет доступ на данном уровне игры.

На своем самом низком уровне (для начинающих) программа отвечает практически немедленно, здесь программа не использует свое знание дебютной теории и не форсирует мат в выигрышных ситуациях.

С другой стороны, на самом высоком уровне для игры C6 обдумывание занимает

очень много времени и программа использует все процедуры для наилучшей игры по переписке (не турнирной).

Краткие сведения об уровнях игры:

уровень	время ответа	
A1	немедленно	для играющих нерегулярно
A2	3 сек.	средняя сила
A3	10 сек.	для опытных игроков
A4	20 сек.	для сильных игроков
A5	40 сек.	клубный стандарт
A6	1 мин. 30 сек.	
A7	3 мин.	турнирная игра
C1	10мин.	для игры по переписке
C2...C6	свыше 10 мин.	

Указанное время является средним. В зависимости от конкретной ситуации оно может изменяться в ту или иную сторону.

Специальные уровни.

В программе есть три специальных уровня игры:

B3 - Beginner - начинающий

B2 - Balance - баланс

B1 - Blitz - блиц.

Эти уровни вызываются нажатием "B" и "1" или "2" или "3".

B1 - Блиц.

На этом уровне вся игра должна быть сыграна в заданное время. Обычное время - 5 мин. Поэтому, прежде чем входить в этот режим, надо соответственно установить часы. Это лучше всего сделать путем сброса часов "T4", а затем установкой минутного разряда - W8 и R8.

Программа не дает сообщения о том, что время истекло, т.е. "технически" Вы можете продолжать игру и далее.

В этом режиме программа варьирует свое время ответа в зависимости от того, сколько времени у нее осталось. Если компьютер просрочит время и игра будет продолжена, то дальше он работает в режиме немедленного ответного хода.

B2 - Баланс.

В этом режиме программа меняет свой уровень игры и, соответственно, время на обдумывание хода так, чтобы оно соответствовало Вашему. Если Вы будете думать долго, то и она тоже, и наоборот. На практике показания часов могут немного расходиться, в пределах нескольких минут.

B3 - Новичок.

Это уникальный уровень игры. Здесь компьютер специально играет слабо. Он не пользуется стандартными началами, не ищет матовых окончаний и не обращается ко многим встроенным процедурам. Таким образом, этот уровень идеален для абсолютного новичка, который не в состоянии играть в силу компьютера на так называемом "слабейшем" уровне A1.

Таблица ходов.

Вы можете просмотреть запись партии "S"+"1". Программа держит в памяти 60 ходов и одновременно изображает 15 из них.

"Q"+"1" - первые 15 ходов

"Q" + "2" - ходы 15...30

"Q" + "3" - ходы 31...45

"Q" + "4" - ходы 46...60

Кроме этого, есть еще режим "Q5". Он показывает ту четверть списка, в которую входит последний сыгранный ход.

Программа позволяет также просматривать партию по записи как вперед, так и назад и воспроизводить "разыгрывание" в двухмерном и трехмерном виде.

Это выполняется нажатием клавиш "Z" и "X" в режиме "S". Для этого нажатием "I" вызовите "CHANGE", затем выберите режим "S" нажатием "M" + "3", а затем "Z" или "X" для просмотра партии вперед и назад. В экране таблицы ходов Вы будете при этом просто пролистывать запись, а в двумерном или в трехмерном экране увидите перемещение фигур.

Эта возможность в программе является очень удобной, если Вы хотите взять ход назад, сделав грубую ошибку. Можно также проанализировать проигранную партию и определить, в каком же ходе была допущена ошибка.

Устранение строки с часами:

Если Вам мешает эта строка, то Вы можете ее убрать нажатием клавиш "L" + "2". Возврат строки - "L" + "2".

Вращение доски в трехмерном режиме.

Вы можете вращать доску в обоих трехмерных экранах: (S3) и (S4). Это выполняется нажатием буквы "O" с цифрой от -"1" до "4".

Загрузка и выгрузка игры.

PSI-CHESS позволяет выгружать позицию на ленту или загружать отложенную позицию с магнитофона. Все факторы, сопутствующие игре, будут выгружены также (например, делали король и ладья ходы или еще нет), т.е. после загрузки отложенной партии программа сможет оценить правильно выполняется рокировка или нет.

Для записи позиции на ленту убедитесь в том, что программа находится в режиме S и что магнитофон включен на запись. После пуска ленты нажмите "S" + "8". Выгрузка выполняется немедленно.

Загрузка выполняется аналогичным образом. Нажмите "S" + "9".

Эти режимы являются незаменимыми, например, при игре по переписке.

Режим "P" (Позиция).

Вход в этот режим выполняется из режима "CHANGE" нажатием "M" + "2". В нижней части экрана появится символ "P".

Расстановка фигур.

По-видимому, наиболее часто расстановка начинается с установки исходной позиции нажатием в режиме "P" клавиш "P" + "1". При этом индикатор хода "TO MOVE" устанавливается на очередность хода белых.

В то же время, есть и другие возможности расстановки. Существуют две базовые методики расстановки. В первом случае Вы вводите положение фигур алгебраической нотацией, во втором случае Вы расставляете символы фигур на двумерной доске. Вы можете также снять фигуру с доски, если она стоит неправильно.

Алгебраическая расстановка.

В этом случае Вы вводите буквы, обозначающие фигуры и их координаты. Обозначения следующие:

S - удаление фигуры;

W - белая фигура;

B - черная фигура;

P - пешка;

N - конь;

В - слон;
Р - ладья;
Q - ферзь;
К - король.

Для того, чтобы установить, например, белого короля на E7, наберите WKE7.

Удаление ошибочно поставленной фигуры с поля D3 выполняется так: SD3.

Такую расстановку можно выполнять как на двумерном, так и на трехмерном экране. Однако, если Вы в трехмерном режиме установите одну фигуру на другую, забыв ее предварительно снять с доски, то на экране возникнет "мешанина" из двух изображений. Если теперь перейти в двумерную графику и опять вернуться в трехмерную, то непорядок устранился, можно устранить его и изменением ориентации. Естественно, останется фигура, которая была поставлена последней.

Расстановка с помощью курсора.

Этот метод также реализуется из режима "Р", но в двумерном экране и вызывается: "S"+"2".

В этом методе Вы можете избрать тип джойстика "J"+"цифра". По обеим сторонам доски появятся символы фигур. Расстановка крайне проста. Вы просто подводите курсор к нужному символу и нажимаете кнопку. Так же, как и в игре, Вы можете обходиться без джойстика, т.к. Синклер-джойстик имитирует работу цифровых клавиш.

Итак, если Вы хотите начать игру сначала или почти сначала, то клавишами "Р"+"1" Вы можете установить исходную позицию начала партии. Если же вам нужно установить эндшпильную позицию, то можете воспользоваться "Р"+"2" или "Р"+"3". В первом случае устанавливаются только короли и ладьи, во втором - только короли.

Обратите внимание, что в этих расстановках предполагается, что короли и ладьи еще не ходили. Если же они уже ходили, то после установки надо их снять и опять поставить на то же место.

Расстановка предлагает вам ряд интересных вариаций. Например, Вы можете прервать игру, изменить ее характер перестановкой фигур и опять продолжить партию.

Расстановка не ограничивает количество фигур, которые Вы можете расставить на доске.

В заключение обратите внимание на стрелку в нижней части экрана, она указывает, куда направлено движение белых фигур. Это нужно в том случае, если в финальной стадии игры Вы в результате поворотов доски потеряете ориентацию.

Продвижение пешки.

Когда в игре пешка достигнет восьмой горизонтали, она должна быть превращена в фигуру. В нижней части экрана появится запрос:

Q R B N ?

Вам в ответ надо нажать одну из этих клавиш, чтобы пешка превратилась в ферзя, ладью, слона или коня, соответственно.

Решение задач.

Хотя PSI-CHESS и не создавалась для этих целей, но на уровне A4 она может искать мат в два хода, а на уровне C1 - в три хода. Для этого надо переключить в режиме S клавишами T1 и T2 программу так, чтобы она играла сама против себя.

Внимание!

S1 служит для загрузки отложенной позиции.

S0 служит для загрузки другой формы фигур (эта функция запроектирована для более поздних версий программы, но пока не реализована).

Если Вы по ошибке нажмете эти клавиши, то не сможете выйти из состояния ожидания загрузки. В этом случае придется отключать питание и перезагружать игру.

Основные команды.
Режим "CHANGE".

B0...B7	- цвет бордюра
C0...C7	- цвет фона
X0...X7	- цвет фигур
L1...L2	- устранение изображения часов
O1...O4	- ориентация доски
Q1...Q4	- выбор части записи партии
Q5	- та часть записи, в которую входит последний ход
S1...S4	- выбор экрана
S1	- таблица ходов
S2	- двумерная доска
S3	- классические трехмерные шахматы
S4	- старинные трехмерные фигуры
Z1...Z8	- громкость звука
I	- вкл/выкл режима "CHANGE"
M1...M3	- основные режимы:
M1	- игра (G)
M2	- расстановка позиции (P)
M3	- настройка программы (S)

Режим настройки "S".

A1...A7	- уровни обычной игры
B1	- "блиц"
B2	- "баланс"
B3	- "новичок"
C1...C6	- уровни для игры по переписке
S0	- загрузка новой формы фигур (пока не реализовано)
S8	- запись отложенной партии
S9	- загрузка отложенной партии
R0...R9	- настройка часов черных
W0...W9	- настройка часов белых
T1	- переключение играющего белыми
T2	- переключение играющего черными
T3	- переключение очередности хода
T4	- установка часов в ноль
Z	- просмотр партии вперед
X	- просмотр партии назад

FORUM

Наша статья с описанием программы "ELITE" не прошла незамеченной для межзвездных асов.

Как мы и предполагали, игра эта всколыхнула читающую, летающую и стреляющую массу. Почта полна исследований самых необычных особенностей этой программы, и нам выпала весьма трудная, хотя и интересная задача каким-то образом систематизировать огромный материал.

Во-первых, мы хотим поздравить наших читателей с тем, что объединенными усилиями им удалось разыскать и выполнить все три встроенные в программу и не документированные в официальной инструкции специальные миссии, хотя не все пилоты знали о том, что то, что они сделали это была миссия. Итак, вот эти миссии в том порядке, в каком они должны следовать:

1. Взрыв "сверхновой".
2. "Cloaking device".
3. E.C.M. System "Jammer".

По материалам зарубежной печати скажем, что в 128-килобайтной версии программы миссий должно быть шесть.

Наиболее полное исследование провел эксперт из Новгорода Александр Дейна. Мы давали его адрес в 4/5 выпуске "РЕВЮ". Он ведет обширную переписку по проблемам программы, т.е. организовал как бы заочный клуб. Основное слово в сегодняшнем обзоре принадлежит ему.

Итак, Вы начинаете свой боевой путь на планете LAVE. Сначала у Вас только 100 кредитов и грузоподъемность корабля всего лишь 20 тонн. Ваши первые действия должны быть направлены на обогащение.

Если вспомнить, что торговля наркотиками дает мафии максимальную прибыль, можно понять, что и в этой игре это быстрейший путь к обогащению, но... на Вашем пути стоит полиция. Вы очень быстро получите статус FUGITIVE, и Вам придется сражаться с полицейскими кораблями. С малым боевым опытом и всего лишь с одним пульсирующим лазером очень маловероятно, что Вам удастся долго продержаться. Правда, статус может постепенно и очиститься. Обычно это происходит после двух-трех гиперперелетов без запрещенных товаров. Мало кто из начинающих пилотов не пробовал запретный плод хотя бы ради эксперимента. Но игра глубже, чем это кажется при беглом знакомстве. Дело в том, что максимальный доход от торговли обычно нужен начинающим, не имеющим хорошего снаряжения, но ведь они не имеют и боевого опыта, а потому быстро гибнут в схватках с полицией. Опытный боец, в свою очередь, может не опасаться полиции, он сильнее ее, но он и не будет торговать гадостью - у него множество гораздо более эффективных и благородных методов составить капитал.

Прежде чем лететь куда-то с грузом товара, надо навести справку об избранной Вами планете. Если там демократический строй или это корпоративное государство - можете лететь смело - пиратов там не будет. Если же это феодальная планета или там у власти диктатура или, еще хуже, анархия - Вам надо хорошо подумать, сражаться с 4-5 противниками имея только пульсирующий лазер - безумие. Другое дело, когда у Вас лучевой лазер, дополнительные энергетические блоки и солидный боевой опыт - Вы специально будете искать такие планеты.

Надо обратить внимание и на уровень развития планеты. Если Вы летите на планету с более высоким уровнем развития, то выгодно везти еду, радиоактивные руды, минералы, спиртное и меха. Обратное хорошо возить, например, компьютеры. Интересная особенность - на феодальных планетах сумасшедшие цены. По видимому, это связано с тем, что пробиться на них мирному торговцу очень непросто.

Ваш первый путь должен лежать на планету LEESTI - это самая безопасная из ближайших планет. К тому же 11-й уровень ее развития обещает неплохую прибыль в

торговле. Сделав гиперпереход, Вы приближаетесь к планете джамп-прыжками, когда рядом нет других объектов. Когда загорится буква S, станция взяла Вас под свое наблюдение, не вздумайте в это время воевать, кроме как с пиратом.

Как отличить пирата от мирного торговца, полицейского или от астероида?

Последние всегда появляются в переднем экране. Пираты же появляются, как правило, под большим углом к Вашей трассе. На радаре их изображение напоминает "кочергу". Чем выше ее вертикальная часть, тем больше этот угол.

(Примечание "ИНФОРКОМа": Мы предпочитаем не гадать, а дожидаться, когда противник сам откроет огонь, и после этого принимать решение - поражать его или уходить маневрами или же спастись на станции).

Действуйте так. Клавишами N,M разверните корабль, чтобы враг находился снизу или сверху вертикальной линии, проходящей через центр радара, а клавишами X,S приведите его в передний экран. Наберите скорость и, когда пират увеличится в размерах, тормозите и атакуйте ракетой или лазером.

За поражение пирата получите премию в размере 4, 5, 17.5, 20 или 50 кредов. Чем серьезнее противник, тем больше премия. Не удивляйтесь, если два-три дня Вам придется учиться четко управлять своим кораблем. В конце концов это лучшая игра десятилетия, и она рассчитана на то, что может быть Вы будете к ней возвращаться годами.

Влетев на станцию, Вы сможете выгодно продать свой товар и купить компьютеры, следующей вашей целью может стать планета DISO (меха) и опять LEESTI, где при удачной рыночной конъюнктуре вам может хватить денег на покупку груза/топливо-заборников FUEL SCOOPS. Может быть летать по этому маршруту придется не раз. Более серьезные путешествия сможете совершать потом, когда оснаститесь нормальным вооружением.

Вот последовательность, в которой надо приобретать оснащение: FUEL SCOOPS - DOCKING COMPUTER - EXTRA BEAM LASER - LARGE CARGO BAY - EXTRA ENERGY UNIT.

(Примечание "ИНФОРКОМа": такой порядок может свидетельствовать об агрессивной настроенности командора. Мы предлагаем иной порядок: DOCKING COMPUTER - EXTRA ENERGY UNIT - EXTRA BEAM LASER - LARGE CARGO BAY - FUEL SCOOPS, но приветствуем и другие варианты. Ведь это говорит о том, что реализуя разные стратегии выживания в космосе, мы имеем дело как бы с разными вариантами игры).

Имея FUEL SCOOPS, Вы можете расстреливать пролетающие навстречу корабли и подбирать грузовые контейнеры с товаром. Мирные торговцы очень удобны своей незащищенностью и содержательностью трюмов.

Приобретая хорошее вооружение, Вы можете торговать и на планетах с нестабильной политической системой. Но особого смысла в этом нет, торговать лучше с мирными планетами, а к феодалам и анархистам лучше летать для набора боевого опыта и повышения рейтинга. Причем здесь, как правило, чем ниже уровень развития, тем больше пиратов. Скорее всего, Вам придется сражаться с 3-4 пиратами, заходящими с двух-трех направлений, но их может быть и больше. Встречалось до 12 пиратов, и спасала от них только энергетическая бомба.

Наилучшая тактика в борьбе против группы следующая:

Остановиться, выбрать цели и выпустить одну-две ракеты. Развернуться в сторону наибольшего скопления противников и набрать скорость, при приближении остановиться и спокойно расстреливать один корабль. Если Вы хороший боец, то Вам удастся уничтожить один или даже два корабля, прежде чем остальные начнут стрелять Вам в спину. Тогда надо вновь развернуться, набрать скорость, затормозиться и расстреливать следующего. Приоритет надо отдавать самым активным пиратам.

(Примечание "ИНФОРКОМа": Признавая успешность этой тактики, мы должны дополнить ее методом ухода из под огня. Когда Вы видите, что запасы энергии на поддержание силовых полей на три четверти израсходованы, надо улетать. Перед стартом полезно выпустить ракету по последней мишени и резко набрать скорость. Как только она достигнет максимума, делайте торможение и поворот на 90 градусов в любую сторону. Разгон - торможение - поворот, этот цикл надо повторять, пока не восстановится уровень энергии в отсеках, а заодно остынет лазер. Не беда, если при этом потеряются один-два

противника. Это не Вы их потеряли, а они Вас. Восстановив энергию, можно продолжить бой. Самое главное - не увлекаться. Уйти можно из очень трудных ситуаций - важно это сделать вовремя.)

Хорошей практикой перед боем с превосходящими силами является предварительная настройка системы гиперперехода на ближайшую звезду (если до нее хватает топлива), тогда в безнадежной ситуации можно улизнуть от противника клавишей "H". Можно воспользоваться энергетической бомбой или спасательной капсулой, но это стоит несколько дороже.

Если на Вас напал корабль странной дискообразной формы - считайте, что Вам повезло - это Таргон. Начинающие пилоты обычно в этом случае погибают до того, как сумеют понять, откуда их поразили. Ракеты его не берут. Его энергетическое поле настолько сильно, что даже энергетическая бомба берет его не всегда, надо сначала повредить его защитное поле несколькими попаданиями лазера. Тем, кто мастерски управляет кораблем, удастся "положить" ненадолго плоскость Таргона в плоскость экрана, тогда его расстрелять проще, и можно это сделать даже пульсирующим лазером.

Пилоты Таргона имеют класс "ЭЛИТА". Кроме этого, Таргон выпускает от трех до шести маленьких беспилотных радиоуправляемых кораблей - Таргоидов. Здесь Вам надо знать, что при уничтожении материнского корабля Таргоиды замолкают, и Вы можете уничтожить их по одному, получив за каждый по 5 кредитов, либо подобрать их с помощью FUEL SCOOPS. Каждый корабль - это одна тонна Alien Items, то есть от 30 до 70 кредитов.

(Наши люди искали Alien Items на планетах в восьми галактиках два года. Спасибо эксперту - он открыл нам глаза. - ИНФОРКОМ).

Успешная битва с Таргоном очень сильно повышает Ваш рейтинг. Рейтинг повышается тем быстрее, чем чаще Вы встречаете сообщение "БРАВО КОМАНДОР". Не всякая победа оценивается такой похвалой из межгалактического центра. Многие сетуют на то, что их рейтинг медленно растёт. Здесь есть особенность. Путь от HARMLESS до COMPETENT можно пройти за один день, но путь от COMPETENT до DANGEROUS может занять недели. С Таргонами Вам может "повезти" еще больше, если они перехватят Вас в гиперпространстве. Отбившись от одной атаки и продолжив гиперпереход, Вы встречаете новую волну, и так без конца. На этом можно быстро заработать хороший рейтинг, но скорее всего, что Вы погибнете. Самое надежное известное средство уйти от них - сделать перелет в другую галактику.

"Взрыв сверхновой".

Вам может встретиться в программе ситуация, когда во время гиперперехода концентрические кольца на экране станут красными, а после перехода все на экране будет красным. После прилета на базу Вы можете продать все свои товары, купить горючее или снаряжение, но при попытке купить товар Вы получите сообщение о том, что Вас просят помочь спасти жителей планеты, поскольку им угрожает взрыв сверхновой. Не вздумайте отказаться и не пытайтесь покупать товары, после перелета Ваш корабль погибнет даже если Вы в последний момент передумаете и возьмете беженцев. Вам надо продать все товары (они все равно пропадут) и ответить согласием на просьбу о помощи. Тогда после обычного гиперперехода Вы окажетесь в другой галактике и, когда высадите спасенных жителей, получите небольшое вознаграждение в алмазах.

(Некоторые наши читатели наблюдали взрыв сверхновой, не получая предложения о спасении людей. В этом случае их выбрасывало в другую галактику даже если межгалактического двигателя у них не было. Мы можем только предположить, что какое-то обстоятельство, по-видимому, вызвало такие события. Может быть миссию не предложили именно потому, что у них не было межгалактического двигателя, может быть правовой статус повлиял, может быть такое предложение делают только тем, кто достиг определенного класса, может быть для этой миссии необходим дополнительный грузовой отсек, одним словом, этот вопрос до конца еще не исследован. - ИНФОРКОМ.)

"Cloaking device".

Это устройство, которое позволяет сделать Ваш корабль невидимым для противника. Устройство не документировано в фирменном описании и отыскание его - вторая миссия программы. Вот что об этом пишет наш читатель из г. Хабаровска М. Стинов.

После того, как Вы достигли рейтинга DANGEROUS, Вам могут встретиться корабли-невидимки. Они появляются на экране лишь на короткие промежутки времени и потому поразить их - задача очень непростая. После нескольких удачных попаданий такой корабль может стать видимым и, если Вам удастся его уничтожить и подобрать разлетающиеся контейнеры, то Вы сможете приобрести специальное устройство - CLOAKING DEVICE, делающее Вас невидимым при нажатии "Y".

Это дает возможность посещать планеты со сложным социальным климатом с минимальным риском быть сбитым. Если в полете Вам встретится армада вражеских кораблей и нет уверенности в победе, нажимайте "Y". Главный экран корабля при этом окрашивается в синий цвет, правда при этом сильно расходуется энергия защитных полей и после того, как опустошится третий энергетический отсек, устройство отключится. Поэтому желательно при встрече с неприятелем не использовать это устройство как можно дольше.

Устройство невидимости позволяет точно поймать в прицел вражеский звездолет и уничтожить его. Особенно удачно это получается из кормового экрана во время преследования.

E.C.M. "JAMMER".

О миссии, связанной с противоракетной системой JAMMER пишет наш читатель из Москвы Кочнев С. В. Когда у него был рейтинг DEADLY, ему предложили уничтожить станцию на планете ISREM, которая использовалась таргондянами для колонизации планеты. При этом уже за то, что он ознакомился с текстом миссии, ему была передана противоракетная система JAMMER.

Самым сложным оказалось найти эту планету, а станция взорвалась от первой же ракеты. Пользуясь отгрузками, он взрывал станцию несколько раз и обратил внимание на то, что по возвращении титулы, которые он получал менялись.

Заканчивая этот обзор программы "ELITE", мы приведем некоторые вопросы наших читателей, которые хотелось бы вынести на обсуждение:

1. Кто видел корабли поколений и космические платформы?
2. Кто видел орбитальные станции типа "Додо"?
3. Кто видел и где планету с названием Raxhla? (Дело в том, что к фирменной программе прилагается научно-фантастическая приключенческая повесть, написанная по мотивам ELITE. Так вот, главная цель героя этой повести и состоит в том, чтобы отыскать в глубинах космоса эту планету, скрывающую важную тайну).
4. Не удалось ли кому-нибудь подзарядиться топливом от звезды?
5. Не были ли обнаружены невидимые планеты?
6. Есть сообщения о том, что иногда в космосе встречаются враждебные орбитальные станции, встречающие огнем Ваш корабль. Что бы это могло значить и как можно их подавить?

Возможно, что многие из этих вопросов и не имеют ответа. В нашей почте есть свидетельства о том, что кое-что из того, что фирма привела в своем описании к игре, относится к версиям программы, написанным для других моделей компьютеров. Так, например, на IBM существует возможность нажатием клавиши I получать справку о приближающемся объекте, а заодно и нацеливать на него ракеты (об этом пишет читатель из Киева Каневский С. Н.).

Загадочные явления.

Пилот класса DANGEROUS С. Ю. Ветров из Ленинграда сообщает о любопытном факте. После загрузки программы в ответ на запрос Load New Commander (Y/N)? нажатием Y он переходит в Access Menu, где нажимает клавишу "2" (Save Commander), вводит свое имя

и имитирует выгрузку на ленту. Затем выходит из меню (клавиша 3) и оказывается в галактике N 47!? с полным комплектом вооружения, с астрономической суммой кредитов и с товарами общим весом много сотен тонн (при максимуме - 35!?) да к тому же не на планете, а где-то в космосе между Tigeap и Reseus.

Об этом же пишет Смирнов В. Б. из Санкт-Петербурга и некоторые другие наши читатели.

По всей видимости речь идет о какой-то неустойчивости в работе программы, а может быть конкретной версии, и мы не стали бы и упоминать об этом факте, но среди пилотов циркулируют слухи о существовании 47-ой галактики, а это все-таки горячий след.

И, наконец, несколько слов о наших опечатках в инструкции к программе "ELITE".

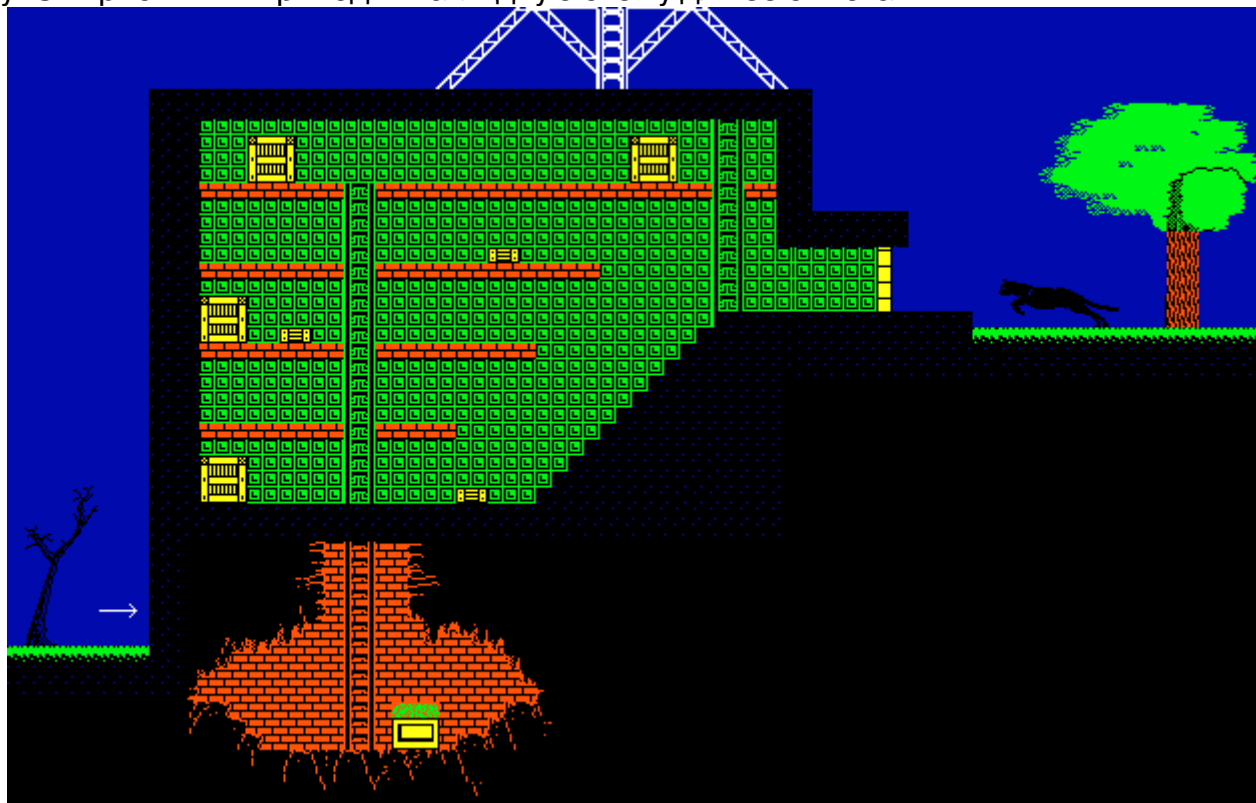
1. Мы написали, что астероиды желательно уничтожать, и за это выдается "наибольшая" премия. Большинство читателей правильно поняли. Конечно надо читать "небольшая" премия.

2. Более серьезная опечатка - клавиша управления энергетической бомбой - "W", а не "U".

SABOTEUR-2

Теперь несколько слов о программе SABOTEUR-2. В общем атаку на эту программу можно считать законченной, осталось только сделать одно уточнение и два дополнения.

Во-первых, мы ошибочно сказали, что потайная комната расположена на нулевом этаже в правой части здания. На самом деле она находится в самой левой части здания сверху. Смирнов В. Б. приводит наглядную схему для ее отыскания.



Дополнительно он сообщает: в игре есть система телепортации. Надо встать рядом с деревом и прыгнуть на стену в направлении, указанном стрелкой. В результате половина вашего тела будет скрыта в стене. Если после этого прыгнуть или пойти вправо, то Вы немедленно окажетесь в правой нижней части здания. Это же обнаружил и Свириденко А. В. из Алушты.

Колотников В. М. из Самары отвечая на вопрос "Форума" о наличии комнаты, где останавливается время, сообщает, что таких комнат две, но время так не останавливается, а прибавляется. Для этого надо так же, как и в игре Saboteur-1, сесть на поезд и нажать клавишу "влево" или "вправо" в зависимости от расположения комнаты.

Заканчивая данный раздел, мы предлагаем по просьбам читателей еще два объекта для коллективной атаки.

Во-первых, это программа TRANTOR. Ветров С. Ю. сообщает, что в этой игре на некоторых этажах есть компьютеры, которые требуют ввода какого-то кода. Если кто-то знает как работать с этой программой, отзовитесь.

Вторая просьба высказана в письмах нескольких читателей - помочь разобраться с программой EQUINOX. Будут также с благодарностью приняты POKES.

По самым низким ценам мы тиражируем программы на компакт-кассетах, дисках 5.25", описания, каталоги.

Напишите по адресу;

140500, г. Луховицы, Московской области, ул. Первомайская, 42, Черкасову А. А.

В день получения Вашего письма Вам будет выслан перечень услуг кооператива "СЕТЬ".

Записываю программное обеспечение (около 2 тыс. программ) для ZX-Spectrum. Каталог с краткими аннотациями всех программ высылается бесплатно.

254108, Киев-108, а/я 21, Шевлякову А. В.

Запись и обмен программ к ZX-Spectrum. Вышлю тематический каталог. К запросу приложить конверт с обратным адресом.

630102, Новосибирск, ул. Богаткова, 50, кв. 29, Ескевичу Алексу Александровичу.

Предлагаю запись программ на дискеты заказчика (размер 5,25) Возможен обмен. 252148, г. Киев, ул. Семьи Сосниных, д. 2-Б, кв. 85, Довженко В. П.

Предлагаю описания:

COPY-COPY - 0р. 66 коп.

GENS 3 - 6р. 83 коп.

MONS 3 - 3р. 28 коп.

Языков: СИ, Макроассемблер, Паскаль, DBASE и др. для IBM по согласованной стоимости.

Провожу расчет пола ребенка (90% совпадений). Необходимые данные - даты рождения обоих родителей. Срок исполнения 1 неделя. Цена услуги - 2р. 50к.

Оплата - предварительная. 195427, г. Ленинград, Светлановский пр-т, д. 73, кв. 17, Смирновой Татьяне Викторовне.

POKES

Многие игры на компьютере "ZX-SPECTRUM" оказываются очень сложными для прохождения всех этапов или уровней, иногда особо настойчивым любителям (или профессионалам) приходится потратить на одну игру несколько дней.

Увидеть или услышать окончание игры очень заманчиво. В этой Вам помогут РОКЕ, приведение ниже.

Для того, чтобы "обессмертить" игру, Вам необходимо, в некоторых случаях раскрыть "загрузчик" к программе, а в некоторых случаях воспользоваться "COPY-COPY". В "Загрузчик" коды необходимо вносить перед последним "RANDOMIZE". Если Вам встретится такая строчка:

```
40 DRAW USR"adress1",USR"adress2"
```

то ее надо разбить:

```
40 RANDOMIZE USR"adress1"
```

```
50 POKE adress,n
```

```
60 RANDOMIZE USR"adress2"
```

Т.е. РОКЕ нужно вносить перед последним стартом кодового блока. В каких случаях Вы будете пользоваться копировщиком "COPY-COPY" определите экспериментальным путем сами.

Приведенный ниже список составил наш читатель из г. Одесса Букалов О.В. Отмеченные "звездочкой" POKES были им проверены.

Мы рассматриваем этот список как базовый и далее будем его пополнять по мере поступления информации.

"А"

1.ACE	32506,0;32507,0;32508, 0;
2.AD ASTRA	38591,0;28592,0;28593,0;
3.AGENT X	26099,0;25917,0;
4.AGENT X II. 1	57821,0;
2	62499,0;
3	50561,0;
5.AIR WOLF II	53471,0;
*6. ALIEN 8	43735,201;(43255,201)
7.ALIEN HIGHWAY	39443,0;39142,0;35125,0;
6.ALIENS	31014,0;30738,0;34464,195;
9.AMAUROTE	46192,0;
10.ANDROID	52250,32;
11.ARKANOID	33702,0;
*12.ARKANOID-2	33054,1;37484,162; 35418,182; или 35417,0;37483,0;
13.ARMY MOVES 2. 1	54597,0;
14.ARMY MOVES 2. 2	53772,0;
15.ATHENA	50267,0; 55268,61;51212,0;
*16.ATIC ATTACK	36519,0;
17.ATTACK OF KILLER TOMATOES	25323,0;49433,81;42160,201;37002,0;
19.ALIEN SINDROME	47503,182;

"В"

1.BASIL...	41296,0;41968,201;
2.BATMAN	36798,0;
3.BATTY	47633,0; или 48437,183;
*4.BIONIC COMMANDO	34690,0;
*5.BOOTY	58294,0;
6.BOULDER I	36610,0;
7.BOULDER II	26028,0;26029,0;26030,0;

8. BOULDER III	26011,0;26012,0; 26013,0;
9. BUBBLER	57517,0;
10. BOBBY BEARING	28094,36; 29750,182-Время; 32173,0-K
11. BUBBLE BOBBLE	43871,52;
12. BOMB JACK II	25379,0; или 31060,0;
13. BARBARIAN (palace)	51005,255;
14. BARBARIAN (psugnosis)	27680,0;
*15. BALL BREAKER II	38874,0;35936,0;35729,99;
16. BEACH HEAD	45465,0;
17. BOUNTY BOB	50955,X;
18. B.C. BILL	47589,201;
19. BLOOD AXE	26582,0;27957,0;

"C"

1. CAR WARS	32337,0;
2. CATCH	46813,0;61635,0;
3. CAULDRON	40060,0;
4. CAULDRON II	52133,0;
5. CHILLER	34025,0;
6. CHUCKIE EGG	42873,0;
7. CHUCKIE EGG II	35453,0;
8. COBRA	36515,183;
*9. CRITICAL MASS	56879,0;
10. CYBERNOID I	24917,255;
11. CYBERNOID II	32202,0;
12. COSMIC CRUISER	26567,0;28523,0;
13. CYCLONE	37536,0;16547,24;
*14. CRYSTAL CASTLES	63733,182;
15. COMANDO	27652,175;27653,264;27654,0;
16. COMBAT LYNX	42525,0;42526,0;42527,0;
17. CON-QUEST	38003,201;
*18. CURSE OF SHERWOOD	63033,0;64613,0;
19. CLIFF HANGER	25692,0;
*20. CHRONOS	53407,201;
*21. CPT. KELLY	42982,0 - Энергия; 47145,0;42517,0;

"D"

1. DAN DARE	23974,168;
2. DAN DARE II	23453,237; 23450,212;
3. DEATH WISH III	38678,183; 39353,183;43301,183;
4. DEFLECTOR	34473,8;41784,0; 42707,201;
5. DIZZY	54216,0;
6. DRILLER	46246,0;
7. DRUID	24890,201;
8. DRUID II	30012,58;
9. DRAGON'S LAIR	51867,0;
10. DONKEY KONG	35370,0;
11. DIMAMITE DAN II	33097,62;33098,192;33099,50;33100,217;33101,91;33102,33; 33103,16; 33104,4;33105,0; 33106,0; 34032,24;43333,201;
*13. DVILLER	46246,0;
*14. DRAGON NINJA	41770,0;

"E"

1. ELITE	46848,201;
----------	------------

*2.ENDURO RACER	43542,0;43643,0;
*3.ELEVATOR ACTION	43630,0; или 43820,0;
4. ESKIMO EDDIE	24686,24;24687,76;
5.ERIC & THE FLOATERS	33245,0;
*6. EXOLON	38120,201;31646,0;35456,0;42338,0;36845,0;38221,0;

"F"

1. FAT WORM	30624,201;
2. FIRE LORD	34509,0; или 39974,0;39975,195;
*3. FREDDY HARDEST I	64011,24;
*4. FREDDY HARDEST II	61607,183;
5. FUTURE KNIGHT	31683,0;
6. FAIRLIGHT II	30429,0;32027,24; 31341,0;
7. FIREFLY	44997, 255;
8. FLYING SHARK	54462,201;54379,3; или 43006,62;43007,2;43008,167;46980,0; 48981,0;48982,0;43006,0;43007,0;43008,0;48630,0;48631,0; 48532,0;48605,0;48606,0;48607,0;
9.FRANK.2000	36047,0;36048,0;36049,0;
*10.FUTURE PROJECT	27662,0;
11.FALL GUY	27235,0;
12.FROST BYTE	33805,182;36560,182;36687,0;
13.FRED	31171,0;

"G"

1.GHOSTS'N GOBLINS	33352,201;
*2.GOONIES	33400,183;
3.GREET GURIANOS	34962,0;
*4.GREEN BERET	42076,0;
5.GRYZOR	33015,255;
6.GREAT ESCAPE	41953,183;
7.GARFIELD	33595,0;
8.GYROSCOPE	53887,201;59149,0;
*9.GOTHIK	43934,58;42110,56;
10.GAINT'S REVENGE	24504,0;
*11.GO TO HELL	63275,5;62254,0;
12.GLUG-GLUG	34139,0;34176,0;

"H"

1.HEAD OVER HEELS	40756,0;40757,0;43132,0;42195,0;35315,0; или 38752,0;38473,201;
2.HYSTERIA	44588,201;
3.HEIST 2012	36106,12;36190,12;
4.HUNGRY HORACE	26426,0;
5.HEAVY OF THE MAGIC	33102,0;33103,0; 33201,48;
6.HUNCH BACK	24760,55;
7.HYDROFOOL	25859,201;RANDOMIZE USR 64071
8.HOPPI'N MAD	41966,0;

"I"

1.INDIANA JONES	33948,0;
2.I'BALL II	43384,0; 49483,0; или 49165,0;49483,0;
3.IKARI WARRIORS	40272,0;
4.IMPOSSABALL	41165,0;37706,0; 37539,0;
5.INTO THE EAGLE'S NEST	36640,0;36641,0; 40518,0; 40513,0;

6.IMPLOSION 33538,182;

"J"

1.JACK THE NIPPER 44278,58;44385,58;
2.JACK THE NIPPER II 43251,0;23739,244; - перед randomize
3.JAIL BREAK 50651,0;
4.JOE BLADE 65032,50; 65029,50;
5.JUDGE DREDD 24936,24;
6.JAWS 33904,182;
7.JET SET 35899,0;

"K"

1.KARNOV 32972,0;
2.KNIGHT MARE 38686,16;38693,16;
3.KOKOTONYWILF 42214,255;
4.KRAKOUT 45565,0;
5.KAI TEMPLE 47783,0;47624,0;
6.KILLER RING 33636,0;
7.KOSMIC KANGAROO 36212,0;
8.KNIGHT TYME 27313,0;32007,183;
9.KNIGHT LORE 49629,175;53567,175;44947,0;50206,0;

"L"

*1.LIGHT FORCE 40725,255;
2.LEGEND OF KAGE 30609,255;
3.LAZER WHEEL 32849,0;
4.LIVING DAYLIGHTS 38913,201;
*5.LAZY JONES 56693,0;
6.LUNAR JETMAN 23439,201;36963,0;
7. LEGEND OF THE AMAZON 57960,0;
8. LAST NINJA-2 36578,175;36579,60;41855,0;

"M"

1.MANIC MINER 35136,0;(36160,0);
*2.METROCROSS 43006,195;44490,0;
*3.MAD MAX 58472,12;
4.MOON ALERT 42404,255;
5.MR.WIMPY 33693,0;
6.MUTANT MONTY 54933,0;
7.MARBLE MADNESS 39579,0;
*8.MOTOS 42241,0;
9.MARIO BROS 44079,0;
10.MR FREEZE 33823,0;33824,0;
11.MASK II 41560,0;41561,0; 44944,0;45325,0; 45326,0;43122,0;
12.MOVIE 64275,195;
13.MARSPORT 34379,24;34057,201; 43865,24;36607,24; 36587,24;44955,24;

"N"

*1.NEBULUS 32913,0;43650,0; или 32921,0;
2.NEMESIS 51949,0;
3.NORTH STAR 44433,0;
4.NOSFERATU 32499,0;39791,0; или 36792,201;
5.NEXOR 36212,201;
6.NEMESIS THE WARLOCK 31851,0;31658,0;

7.NAVY MOVES I	49962,0;
8.NAVY MOVES II	54047,0;55805,0; 58837,0;

"O"

1.OUTRUN	39204,0;
2.ORBIX	65529,191;32127,0;32188,0;
3.OVERLANDER	29521,0;
*4.OCTAN 58309,0;	

"P"

1.PSYTRON	28625,0;26143,255;26144,0;
2.PUD PUD	49287,0;
3.PUNCHY	45632,0;
*4.PENTAGRAM	49917,0; 50751,0;
*5.PROHIBITION	25421,182;
6.PANZADROME	25657,1;30410,0;32597,0;28658,0;28854,0;
7.PIPELINE II	32511,18;
8.PSYCHO SOLDIER	33372,62;33373,4;33374,0;
*9.PROFANATION	44015,201;

"Q"

1.QUAZATRON	56243,0;58244,0; 58245,0;58267,0;58268,0;58269,0;
-------------	---------------------------------------------------

"R"

1.RENEGADE	41048,185;
*2.ROGUE TROOPER	30374,0;
3.RAMBO	27401,52;60263,0;
4.RYGAR	51216,0;61577,0;
5.REVOLUTION	35653,183;47111,0; 34831,K-(0,7)
6.ROBIN OF THE WOOD	46690,0;

"S"

1.SABRE WOLF	44786,0;
*2.SABOTEUR II	37121,0;37122,0 - время 61338,162;61362,182 - жизнь
3.SAMANTHA FOX	20728,62;20729,6; 20730,50;20731,112; 20732,91;20733,195; 20734,224;20735,81;
4.STAR QUAKE	50274,0; (через COPY-COPY)
5.SIR LANCELOT	33892,0;33693,0;
6.SPACE HARRIER	41499,N; или 46543,195;46544,14; 46545,182;46570,195; 46571,14;46572,132;
7.SUB NIGHTMARE	43435,0;
8.STYX	31753,0;31764,0; 31765,0;31766,0;
9.STARION	46271,0;46272,0; 46273,0;
10.STOP THE EXPRESS	34464,183;
11.SEAS OF BLOOD	34242,195;34247,195;30241,201;
12.SORCERY	49823,0;
13.SURVIVOR	37734,0;37735,0;-энергия 36047,0;36048,0;-энергия
14.SPIRITS	51754,0;
15.STAR WARS	45268,0;
*16.SLAP FIGHT	46672,0;48873,0;48674,0;57115,201;

"T"

1.THE BUDS & THE BEES 37229,0;
*2.TIRNANOG 34202,255;
*3.THUNDERBIRDS 58927,0;56926,0;58929,0;58930,0;
4.TERRA CRESTA 35050,0;35051,0; 35052,0;
5.TUTANKHAMUN 27783,0; 27279,X-(0,255);
6.TORMOIL 57557,0;

"V"

*1.VIXEN (A,B,C) 51794,175
*2.VIGILANT 48735,0;

"W"

1.WAR CARS 32337,0;

"X"

1.XEVIOUS 53592,255;

"Z"

*.ZYNAPS 43592,24;
2.ZYTHUM 54786,0;54790,0;54791,0;51270,0;51271,0;

Ниже приведены некоторые варианты загрузчиков с POKE для различных версий:

AMAUROTE

(2-ой вариант)

```
10 CLEAR 24999:LOAD ""CODE:RANDOMIZE USR 5e4
20 POKE 23570,16:LOAD""CODE:FOR I = 1 TO 4:READ A
30 FOR J=A TO A+2:POKE J,0: NEXT J: NEXT I
40 RANDOMIZE USR 36924: PRINT USR 26600
50 DATA 42912,42929,42455,38551
```

ASTRO BLASTER

```
10 CLEAR 32757:POKE 23613,87:SAVE "ASTRO" CODE 23570,760
20 POKE 27422,0: POKE 26396,255
```

DUET

```
10 CLEAR 24229: LOAD""CODE: POKE 48270,57: RANDOMIZE USR 5e4
```

FRANK.2000

```
1 LOAD ""CODE:RANDOMIZE USR 27100
2 POKE 36047,0:POKE 36048,0:POKE 36049, 0
3 RANDOMIZE USR 36000
```

GLUG-GLUG

```
10 LOAD""CODE:POKE 23313,201:RANDOMIZE USR 23297
20 POKE 34139,0:POKE 34176,0:RANDOMIZE USR 34349
```

GUN RUNNER

(для фирменного варианта)

```
1 BORDER 0:POKE 23693,0:POKE 23624,0:CLEAR 25317
2 LOAD""CODE:POKE 64531,68:RANDOMIZE USR 64512:POKE 65120,12
3 FOR I=23308 TO 23323:READ A:POKE I,A:NEXT I:RANDOMIZE USR 65082
4 DATA 175,50,19,192,50,132,190,50,169,192,50,69,205,195,196,187
```

(для других вариантов)

```
POKE 49171,0:POKE 46772,0: POKE 49321,0:POKE 52549,0: RANDOMIZE USR 48070
```

GHOST'N GOBLINS

```
10 CLEAR 24791:LOAD""CODE:LOAD "" CODE:RANDOMIZE USR 24830: LOAD""CODE 16464:POKE  
36058,0:POKE 36059,0: POKE 18345,x-1
```

HEAVY ON THE MAGIC (2-ой вариант)

```
10 CLEAR 24499:POKE 23570,16:LOAD""CODE:LOAD ""CODE:POKE 40207,24:POKE 33130,201:POKE  
33169,201:POKE 33240,203:RANDOMIZE USR 18434
```

HYDROFOOL

(для фирменного варианта)

```
1 CLEAR 84769:LOAD ""CODE:POKE 64066,185:POKE 64069,248:POKE 64070,255  
10 FOR I=65528 TO 65535:READ A:POKE I,A:NEXT I:RANDOMIZE USR 63500  
20 DATA 62,201,50,3,101,195,71,250
```

ICE ATTACK

```
20 CLEAR 24999:POKE 23800,195: RANDOMIZE USR 23760:POKE 53111,60:RANDOMIZE USR 23803
```

JACK THE NIPPER

(2-ой вариант)

```
10 CLEAR 24791:LOAD""CODE:RANDOMIZE USR 24630:POKE 44278,58:POKE 44285,56:LOAD ""CODE 16464:  
RANDOMIZE USR 24833
```

LIGHT FORCE

```
10 CLEAR 24791:LOAD ""CODE: LOAD ""CODE: RANDOMIZE USR 24830: POKE 40721,201:POKE  
40725,0:LOAD "" CODE 16464:RANDOMIZE USR 24833
```

NETHER EARTH

```
10 FOR I=16384 TO 16411:READ A: POKE I,A:NEXT I:RANDOMIZE USR 16384  
20 DATA 49,255,67,221,33,0,91,17,0,165,62,255,55,205,86,5,33,0,0,34,129,202,34,131,202,  
195,0,166
```

RANARAMA

```
20 CLEAR 24999:POKE 23800,195: RANDOMIZE USR 23760:POKE 57421,0: POKE 57436,205: POKE  
57572,201: POKE 59821,0:POKE 59836,0: RANDOMIZE USR 23803
```

RENEGADE

```
100 MERGE "" :POKE 23800,201:CLEAR 24699:RANDOMIZE USR 23760  
110 POKE 23800,16  
120 POKE 41047,182 - жизнь  
130 POKE 30301,195 - время  
140 RANDOMIZE USR 23600
```

THANATOS

```
10 CLEAR 24791:LET A=24:POKE 23570,16:LOAD""CODE:LOAD""CODE  
20 RANDOMIZE USR 24830:LOAD""CODE 16464  
30 POKE 56056,201:POKE 56549,A:POKE 55816,A:POKE 58021,A:POKE 58220,A:POKE 58788,A:  
POKE 59174,A:POKE 59240,A:POKE 57603,A:POKE 57604,A:PRINT USR 24833
```

TRAP DOOR

```
1 CLEAR 24791:LOAD""CODE:LOAD""CODE  
2 RANDOMIZE USR 24830  
3 POKE 44108,201 - время для английской версии  
4 POKE 41283,201 - время для пол. версии  
5 LOAD "" CODE 16464:RANDOMIZE USR 24833
```

URIDIUM

(1-ый вариант)

```
1 FOR A=23296 TO 23327:READ S: POKE A,S:NEXT A:RANDOMIZE USR 23296  
2 DATA 221,33,0,64,17,0,27,55,62,255,205,86,5,1,0,0,205,61,31,221,  
33,0,64,17,0,192,62,255,194,4,201
```

(2-ой вариант)

```
1 FOR F=43773 TO 43814:READ A:POKE F,A: NEXT F
2 DATA 62,126,219,254,230,31,254,31,203,17,62,251,219,254,31,63,203,17,62,253,219,254,
    31,53,203,17,62,223,219,254,47,31,31,203,17,23,203,17,0,0,0,0
```

WAY OF THE EXPLODING FIST

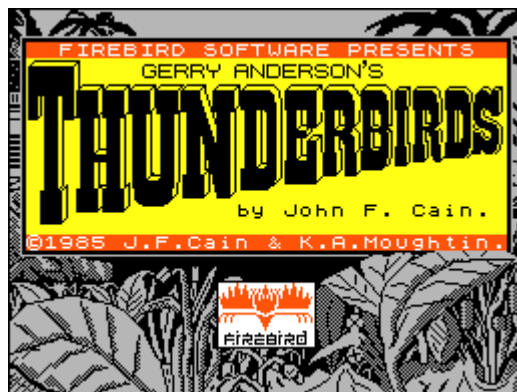
```
1 CLEAR 65399:LET A=65400
2 READ N:IF N>256 THEN RANDOMIZE USR 65400
3 POKE A,N:LET A=A+1: GO TO 2
4 DATA 221,33,0,64,17,96,198,62,255,55,205,86,5,33,0,0
5 DATA 34,175,105,175,60,177,105
6 DATA 50,161,105
7 DATA 62,255,50,97,106
8 DATA 62,243,50,182,105
9 DATA 62,243,50,183,105
10 DATA 195,14,241
11 DATA 999
```

СЛОВО ЭКСПЕРТАМ

Сегодня мы начинаем публиковать проработки наших экспертов и обозревателей по игровым программам, полученные в ответ на обращение в N 4-5 ZX-РЕВЮ. Мы открыты для такого сотрудничества и тех, кто хочет попробовать свои силы приглашаем внимательно ознакомиться с условиями работы, приведенными там и обращаться с предложениями.

THUNDERBIRDS 1-4

Grandslam LTD 1989г.



Эксперт Порядин С. В.,
Марийская ССР

Игры сериала Thunderbirds можно отнести к аркадно-адвентурному жанру. Весь сериал состоит из четырех отдельно загружаемых игр, в которых нужно, поочередно управляя двумя спасателями из прославленного интернационального спасательного отряда "Thunderbirds", выполнить определенные задания (миссии). На одну игру - одна миссия. Следовательно всего нужно пройти четыре миссии. На каждое задание отводится определенное количество времени. Если время истекает, а основная цель миссии не достигнута, то игра заканчивается. С переходом к выполнению следующей миссии время, отводимое для достижения цели уменьшается, но у Вас появляется значительный опыт в управлении работой спасателей.



Прохождение миссий строго последовательно, так как по окончании миссий Вам выдается пароль на вход в следующую. Несмотря на то, что мы приведем в конце описания эти пароли, мы настоятельно рекомендуем проходить миссии последовательно, так как сложность прохождения миссий от первой к последующим увеличивается. Если же Вы попытаетесь сразу начать с выполнения, например, третьей миссии, то весьма вероятно, что дальше нескольких шагов в первых попытках не уйдете и потратите на прохождение этой миссии больше времени, чем если бы проходили все миссии последовательно.

Теперь о самих миссиях.

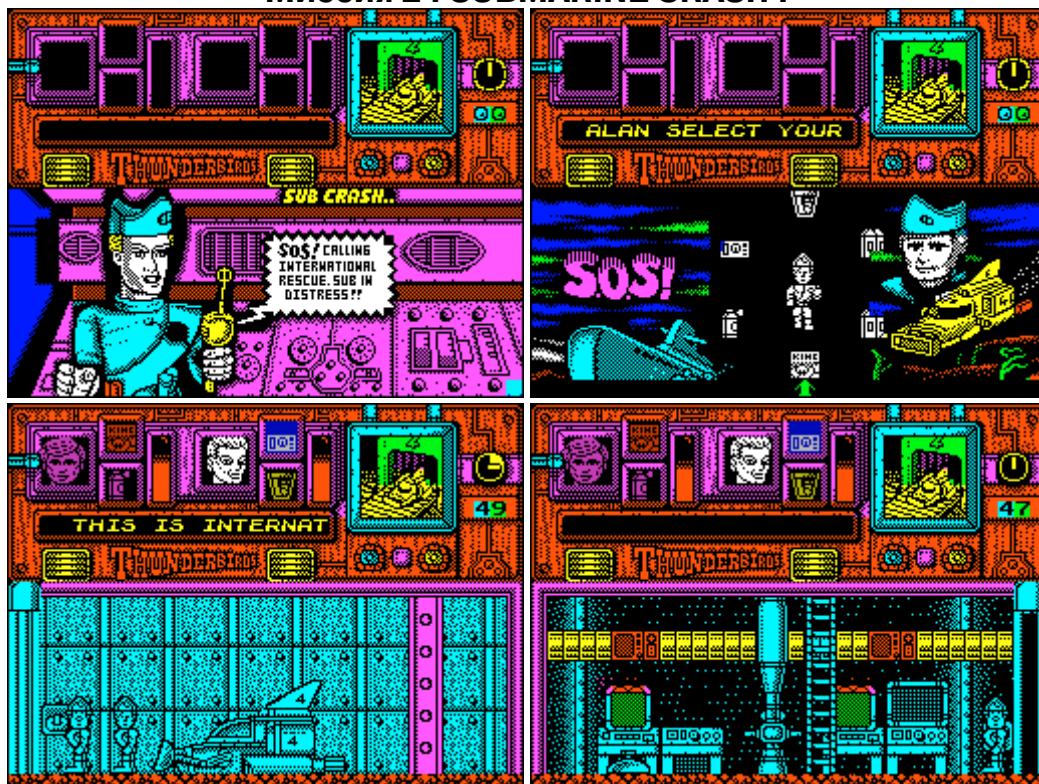
Миссия 1. MINE RESCUE.



Двое спасателей Алан и Брайн получают задание вызволить попавшего в беду шахтера, которого завалило в шахте вместе с вагонеткой. Алан пытается пробраться к шахтеру с поверхности, а Брайн - из подрытого туннеля. В пути спасателям грозят всяческие опасности: затопляющая шахту вода, срывающиеся с потолка сталактиты и темные помещения, где у спасателей быстро тают силы (конечно же не от страха). Кроме этого встречаются различные препятствия: ямы, заклинивший лифт, заваленный проход и так далее.

Но, в конце концов, спасатели применяют всю свою находчивость и некоторые предметы, имеющиеся у них, и смогут выполнить задание и перейти к следующей миссии.

Миссия 2. SUBMARINE CRASH.



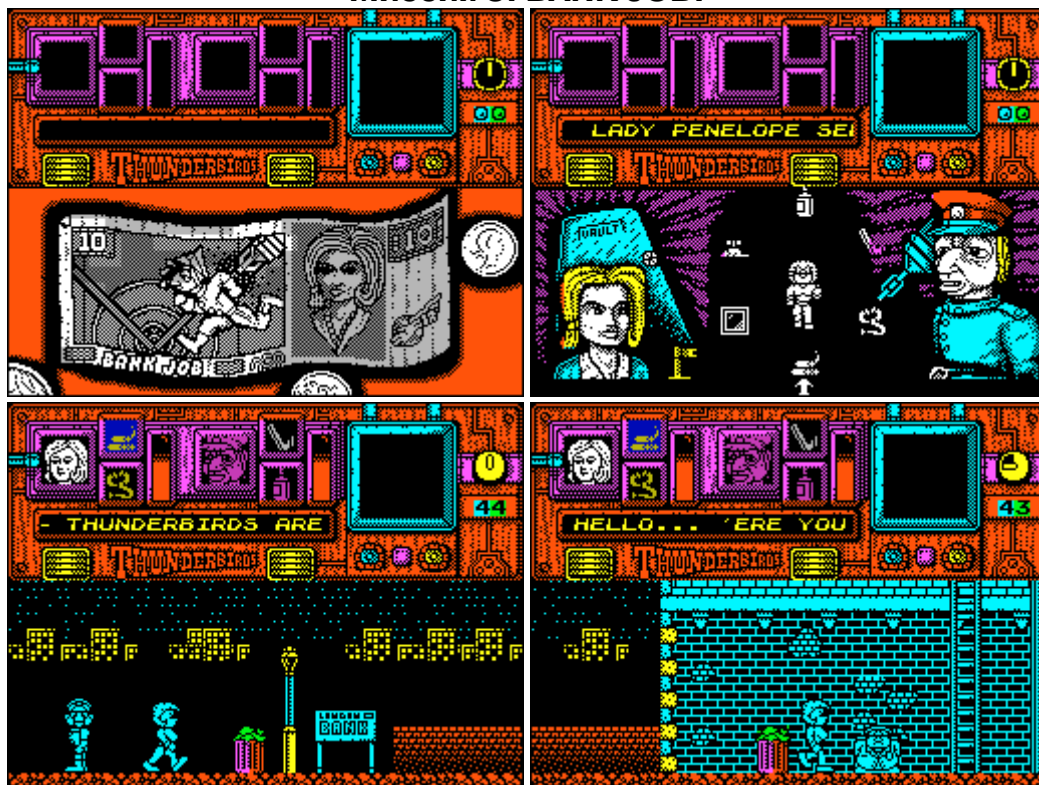
О помощи взывает полузатопленная атомная подводная лодка, в которой в

довершение всех бед произошла утечка радиации, и почти весь экипаж покинул ее.

Спасателям предстоит откачать воду из герметично закрывающихся отсеков и запустить заглушенный реактор. Причем запустить реактор под силу только двум операторам. Реактор запускается с двух пультов, в которые нужно вставить определенные магнитные карточки. Если были применены не те карточки, которые нужно, то срабатывает защита.

В затопленный отсек из "сухого" соседнего попасть невозможно. Поднять герметичную перегородку можно после затопления отсека, в котором Вы находитесь, водой. В этой миссии спасателям угрожает радиация, забортная вода и акулы обитающие в море.

Миссия 3. BANK JOB.



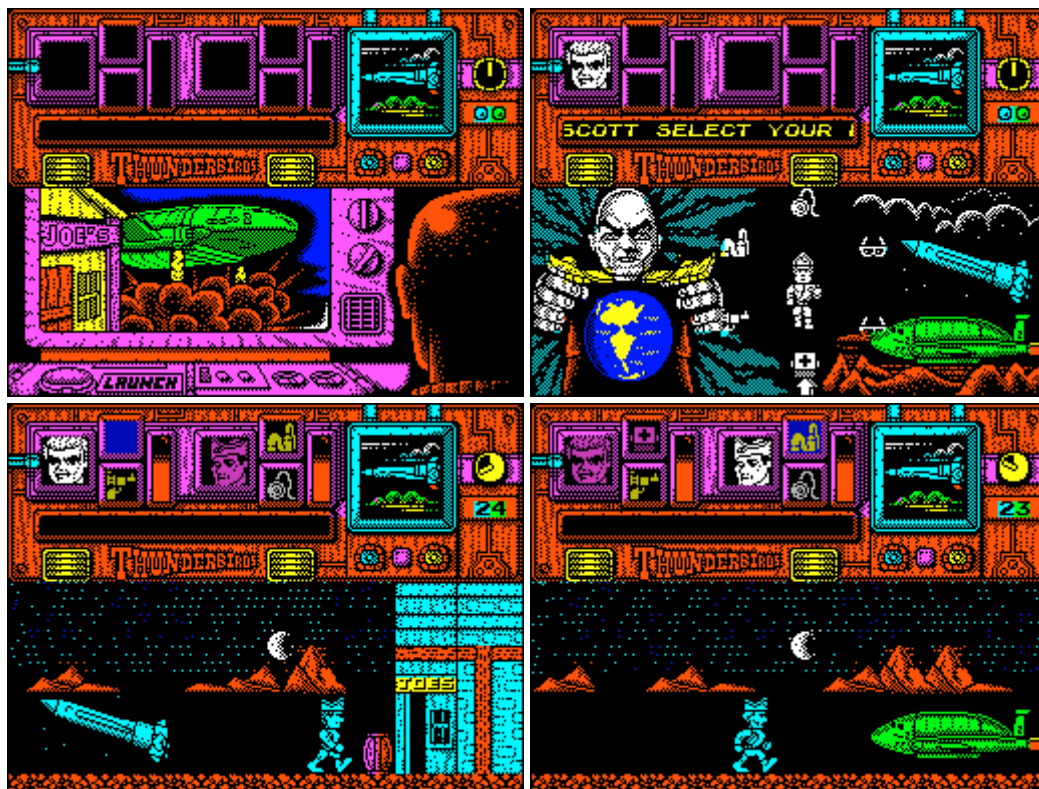
Здесь двум спасателям мистеру Паркеру и леди Пенелопе дается задание выкрасть из тщательно охраняемого частного банка секретные бумаги, имеющие большое государственное значение.

Спасателям будет мешать охрана, электронная защита, им также следует опасаться крыс, обитающих в подвалах банка. Но спасатели как всегда с честью выйдут из этой ситуации и приступят к выполнению четвертого задания, завершающего данный сериал.

Миссия 4. EVIL HOOD

В этой миссии спасателям предстоит устранить опасность нависшую над всей Землей, а именно обезвредить готовую к старту ракету с шестимегатонной термоядерной боеголовкой нацеленной на Лондон. Ракету нужно заминировать заложив мину в специальный отсек. Ракета расположена в подземной шахте, вход в которую замаскирован в обыкновенном доме. Ракету охраняет электронная защита и военные роботы

Эта миссия является самой сложной и на прохождение ее выделяется в четыре раза меньше времени, чем на прохождение первой миссии



Теперь ознакомившись с тем, что нужно сделать перейдем к рассмотрению вопросов как это сделать

Запуск игры

Сразу после загрузки программа предлагает выбрать один из джойстиков или клавиатуру затем она запросит подтверждение правильности выбора. После утвердительного ответа Вы выбираете по два предмета с которыми вступит в игру каждый спасатель. Предметов больше, чем Вы можете взять поэтому нужно выбрать самое необходимое. Изображения выбираемых предметов можно перемещать с помощью клавиш вправо и влево или джойстиком. Выбор предмета - нажатием клавиши огонь.

После выбора предметов для обоих спасателей Вы приступаете непосредственно к игре.

Экран

Вы увидите разделенный на поля экран. В нижней части экрана находится игровое поле, где перемещается главный герой, где находятся различные предметы, среди которых он действует.

В правом верхнем углу изображены часы и счетчик, указывающий сколько времени Вам осталось для выполнения конкретной миссии. Вверху экрана, над игровым полем, изображены два портрета спасателей участвующих в выполнении данной миссии. Портрет спасателя которым непосредственно управляет игрок, выделен цветом. Переключение осуществляется нажатием клавиши "ПРОБЕЛ".

Рядом с каждым портретом в двух окнах находятся стилизованные изображения предметов которые имеются в распоряжении каждого спасателя. Активный предмет, которым спасатель может воспользоваться выделен цветом.

Переключение активных предметов осуществляется нажатием клавиш "1" и "2".

Справа от изображения предметов находится индикатор, показывающий уровень сил каждого спасателя.

Над игровым полем наводится информационная строка, наблюдение за которой может облегчить прохождение игры.

Управление

Передвижение главного героя осуществляется очень просто: клавишами

"Z", "X", "K", "O", " " или джойстиком. Но кроме этого главному герою потребуется применять предметы или просто подбирать различные вещи, лежащие на земле.

Подобрать предмет можно подойдя к нему и нажав клавишу "вниз", но при этом нужно помнить, что одновременно у спасателя может находиться только два предмета. Таким образом, Вы можете, подобрав один предмет, оставить на земле другой, более нужный.

Нажав клавишу "H", Вы получите в информационной строке названия предметов, которые имеет спасатель, находящийся под вашим контролем.

Одновременно нажав клавиши "Caps Shift" и "Q" Вы начнете игру заново.

Переключение спасателя, действиями которого Вы управляете - клавишей "ПРОБЕЛ". Переключение активных предметов - клавишами "1" и "2".

Как отмечено выше, Вы поочередно управляете двумя спасателями, которые по сюжету игры могут находиться вместе или раздельно. Но ни одно задание нельзя выполнить действуя только одним спасателем. При гибели одного из них игра заканчивается, и Вам предлагается повторить попытку.

Применение предметов

Применение предмета осуществляется или автоматически (в случае акваланга, фонарика, репеллента) или с помощью одновременно нажатых клавиш "огонь" и "вправо" ("влево"). Задействовать конкретный предмет можно только в определенных условиях или в определенном месте. При этом использованный предмет может остаться у Вас, но может и исчезнуть. Так, взорвав динамитную шашку, Вы у себя ее, естественно, больше не обнаружите. Почти все предметы, которые встречаются Вам по ходу игры, находят в ней свое применение. Некоторые предметы можно использовать только совместно, по отдельности они никакой пользы принести не могут. Например: детонатор и обрывки медного провода помогут спасателю подорвать заряд, заложенный в завал, преграждающий ему дорогу.

В первых играх сериала предмет можно использовать только в одном (нужном) случае и после использования надобность в этом предмете, как правило, отпадает. Но на последнем уровне игры использовав некоторые предметы неосмотрительно можно даже навредить себе или просто лишиться необходимого в дальнейшем предмета.

Сила

Начиная игру, каждый спасатель имеет один и тот же уровень сил, который по ходу игры может уменьшаться. Увеличить уровень сил невозможно и, израсходовав все силы, спасатель погибает, но если быть осторожным и действовать осмотрительно, то игру можно закончить с таким же уровнем сил, с каким начали ее.

От воздействия различных факторов силы расходуются с различной скоростью. Под воздействием радиации силы расходуются медленно, но у самого реактора подводной лодки смерть наступает почти мгновенно.

В темных помещениях и в воде силы расходуются очень быстро. Охранник в банке и акула убивают моментально, как и защита в ракетной шахте.

Силы могут уменьшаться при падениях в ямы, попадании в спасателя сорвавшимся в пещере сталактитом и от других различных факторов.

Некоторые советы

Теперь несколько конкретных советов, которые облегчат прохождение игр сериала.

В миссии 1 шахту начинает заливать вода, в которой Брайн может утонуть, но воду можно откачать с помощью помпы. Помпу можно отремонтировать с помощью гаечного ключа.

В некоторых случаях можно обойтись без применения определенного предмета, но это приведет к замедлению прохождения миссии. Например, в случае с заржавевшей вагонеткой, ее можно смазать из масленки и она покатится быстрее.

В миссии 2 применение противорадиационных таблеток позволяет сохранить силы спасателям.

В миссии 3 охранника в банковском помещении можно обмануть действуя

попеременно обоими спасателями, избегая встречи с ним и зная, что охранник по лестницам перемещаться не может. Электронную защиту главного сейфа можно отключить тоже только совместными действиями обоих спасателей. Защита отключается путем набора определенного кода с помощью переключателей, находящихся по обе стороны помещения, где расположен главный сейф.

В миссии 4 военного робота, охраняющего вход в ракетную шахту, можно застрелить из пистолета, но это не является оптимальным вариантом. Обезвредить робота можно совсем необычным способом. Каким? - догадайтесь сами, ведь в этом и состоит вся прелесть игры.

Пароли

Теперь о паролях для входа в миссии 2, 3, 4 - это слова: RECOVERY; ALOYSIUS; ANDERSON соответственно. Но, конечно, будет лучше, если их Вам будет сообщать компьютер после окончания каждой миссии. Ну а теперь вставьте кассету в магнитофон и загрузите игру.

УДАЧИ ВАМ СПАСАТЕЛИ!

HAMMERFIST

VIVID IMAGE 1990 г.



Обозреватель Носков В. А.
г. Пермь

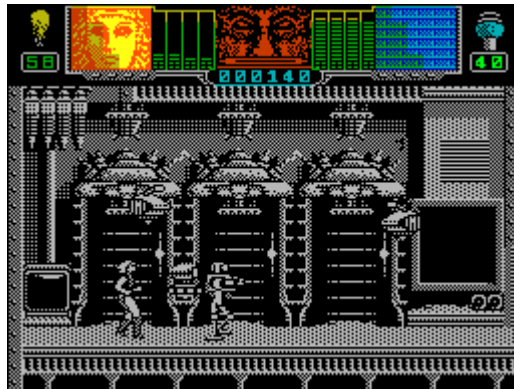
ACTION - казалось бы, что нового можно предложить в этом жанре игр?

Первое, что бросается в глаза это тщательно проработанная графика, плавное перемещение персонажей по экрану, фантастический сюжет. Смело можно ставить HAMMERRFIST в один ряд с DAN DARE, MASK III, FREDDY HARDEST и др.

Ваш герой - робот и даже не один, а два. Одним нажатием клавиши "вверх" он превращается то в сурового мужчину, готового своими воистину железными кулаками снести все на своем пути, то в спортивного вида женщину, предпочитающую через препятствия перепрыгивать.

Действие происходит в далеком будущем, на дне океана. Подводная база так и кишит враждебно настроенными роботами, мутантами, чудовищами и прочей нечистью.

Если Вы думаете, что Вам предстоит расчищать дорогу и идти вперед, то серьезно заблуждаетесь. Предстоит немало поломать голову, чтобы найти изящный обход очередного препятствия. Вам встретятся и двери, открывающиеся только после набора определенного количества очков.



Когда же убив выходной, доконав клавиатуру своего компьютера и вспоминая (добрыми словами) авторов этой игры Вы наконец-то выберетесь из базы, то окажется, что самое интересное еще впереди.

Сначала управление игрой кажется несколько инерционным, но через несколько часов игры это чувство проходит.

MYTH



SYSTEM 3 1989г.

Обозреватель FREE group

г. Тернополь

Эта игра - прекрасная реклама для фирмы SYSTEM 3. Имеет 5 догружаемых уровней. Великолепная плавная графика хоть и мелковата, но зато очень подробна. Движения персонажей реалистичны. Сюжет программы основан на древних легендах и играть в MYTH очень интересно.

К сожалению за неделю труда Вы пройдете эту программу "от корки" до "корки", но никогда не сотрете со своей кассеты - это точно.

Это программа, которая понравится каждому. Не пожалейте труда и найдите MYTH.

KLAX

TENGEN 1990г.

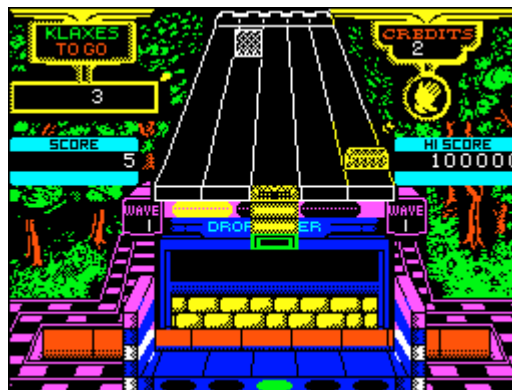


Обозреватель FREE group

г. Тернополь

Для "Спектрума" написано множество головоломок. Исторически одной из ведущих является TETRIS, но на сей раз ему по-видимому становится тесно.

Как игра KLAX более привлекателен и сделан лучше. Кроме того, 100 уровней, не считая догружаемых, гарантируют, что программа Вам не скоро наскучит. Каждый уровень состоит из нескольких подуровней, притом с разными заданиями: собирание вертикалей, горизонталей, диагоналей. В KLAX нужно играть на цветном телевизоре, поскольку цвет играет важную роль, хотя есть опыт проведения до 60-го уровня и на черно-белом телевизоре.



разгром войск противника. Поскольку высадка идет постепенно, то обе эти фазы действуют одновременно. Роль высадки сводится к своего рода снабжению.

Как и во всякой стратегической игре к успеху приводит серия удачных маневров, основной целью которых является создание локального перевеса сил и уничтожение частей противника при сравнительно малых потерях, но сделать это отнюдь не просто.

Игра начинается при примерном равенстве сил противоборствующих сторон, но ваша задача существенно усложнена необходимостью производить высадку соединений отдельными частями и тут же освобождать плацдарм для высадки новых.

Во-первых, Вам трудно сконцентрировать войска в начале игры, в то время как противник действует крупными ударными группировками, грамотно учитывая рельеф местности для быстрой переброски войск.

Во-вторых, сила частей в атаке конечно меньше, чем в обороне. Вам предстоит непрерывно атаковать значительно превосходящие силы противника, чтобы расчистить плацдарм для новых сил вторжения.

Все, что Вы ни делаете - делается в начале игры неоптимально, и когда наконец прорыв произойдет и части развернутся к бою, (если Вас сразу не сбросили в море), Вы обнаружите, что Ваши силы уже существенно уступают германским и спасти Вас сможет только удачное маневрирование.

Настройка программы.

После загрузки Вам предлагается выбрать исходный уровень сложности от 1 до 3, хотя мы не почувствовали существенной разницы. Без достаточного опыта трудно победить и на 1-ом уровне, опытный игрок побеждает и на 3-ем.

После этого Вам предлагается загрузить отложенную игру (клавиша "L") или начать новую (любая другая клавиша).

Перед Вами карта северного побережья Франции. Гладкий скроллинг выполняется курсорными клавишами.

Ваша первая задача - наметить пункты высадки десанта, возможны 12 пунктов (клавиши от A до L), но Вы должны из них выбрать только 5. Это очень ответственный этап. Многое будет зависеть от правильности выбранного Вами стратегического решения.

Закончив выбор пунктов вторжения, Вы получаете выходной запрос, который будет повторяться после каждого игрового шага:

S - Save Game (выгрузить позицию)

Continue - продолжить (любая другая клавиша).

Затем Вы попадаете в главное меню:

ARMY ORDERS - "A" (отдать приказ по армии).

MOVEMENT - "O" (привести войска в движение согласно приказу).

Выбрав "A", Вы получаете изображение союзных армий, - их 7. Американских - 5, британских - 1 и один канадский экспедиционный корпус.

Перемещение курсора между армиями - "P" и "Q", а выбор нужной армии - "A". Клавиша "E" - возврат.

Каждая армия состоит из 6-ти частей (полков, бригад). Перемещение курсора - "P" и "Q". Выбор - "U". Клавиша "E" - возврат.

Возврат всегда выполняется в предыдущее меню.

Ваша задача - выбрав боевую часть, назначить ей плацдарм для высадки (от A до L), разумеется из числа намеченных ранее.

Если выбор сделан правильно, часть займет исходное положение.

Здесь есть особенности.

Во-первых, высадка может начинаться только с пехотных частей (они изображены символом винтовки). Поэтому сразу привести в действие например танковый полк Вам не удастся.

Во-вторых, часть займет исходное положение только если плацдарм высадки свободен как от своих частей, так и от противника. На первом шагу это неважно, но далее Вам придется как можно быстрее освобождать плацдарм.

Последовательным нажатием "Е" Вы можете вернуться к главному меню, которое теперь выглядит так:

ORDER - "O" (отдача приказа)

DETAIL - "D" (состояние войск)

TERRAIN - "T" (расположение частей)

Клавиша "O" позволяет отдать приказ по армии. Сначала Вы выбираете армию, которой хотите отдать приказ. За семью союзными армиями закреплены клавиши от 1 до 7. Обратите внимание, что приказу подчиняются только те части, которые уже высадились на континент.

Приказ армии отдается тремя перемещениями курсора. Сначала установив курсор X Вы отмечаете то место, куда должен переместиться центр Вашей армии. Затем курсором R задаете перемещение правого фланга, а курсором L - левого.

Пока части не перемещаются. Они только получают приказ, исполнение начнется потом, когда Вы в вышестоящем меню нажмете клавишу O (MOVEMENT).

Это очень оригинальное управление перемещением армий указанием желаемого места расположения центра и флангов. Тем самым Вы как бы не управляете отдельными частями, а управляете армией в целом. Реакция же отдельных частей может быть неожиданной. Это зависит от ее силы, морального состояния, рельефа местности, наличия контакта с противником и наконец от того, свободен ли маршрут. Лишь в самых простых ситуациях Вы получите тот результат, который задали, как правило же реальная картина будет лишь приближаться к Вашим планам и степень этого приближения зависит от Вашего мастерства. После задания направления перемещения Вам предложат указать режим боевых действий армии:

"A" - атаковать;

"D" - обороняться;

"E" - выход.

Клавишей "D" (DETAILS) Вы можете просмотреть состояние своих частей и частей противника:

G - германские армии;

A - союзные армии.

Клавишей "T" (TERRAIN) Вы можете определить местоположение своей армии (хотя маловероятно, чтобы Вы ее потеряли).

Закончив отдачу приказов по армиям, Вы возвращаетесь в верхнее меню и клавишей O (MOVEMENT) приводите войска в движение.

Каждый игровой шаг состоит из четырех этапов.

1) Передвижение союзных войск.

2) Сражение (COMBAT) в точках контакта с противником.

3) Передвижение германских войск.

4) Сражение.

Все они выполняются компьютером автоматически. Наблюдая со стороны за ходом сражения, Вы вносите коррективы в свои решения и планируете приказ на очередной шаг.

Некоторые советы.

1) Игра может иметь большое количество стратегий. Можно производить высадку на рядом расположенные плацдармы. Тогда первая фаза пройдет относительно успешно, но противник успеет стянуть мощные силы к этому направлению и во второй фазе Вам придется плохо.

Можно рассредоточить силы по далеко отстоящим плацдармам и тогда противник не сконцентрирует войска, но в первой фазе Вы потеряете при высадке очень много частей.

Победная стратегия лежит по-видимому в умелом комбинировании разных идей и гибком реагировании на ход сражения.

2) "Проутюжив" плацдарм, противник уничтожает его и закрывает тем самым для Вас возможность высадки новых частей в этом месте. Потеряв все плацдармы, Вы проигрываете игру.

С другой стороны, Ваши войска могут, действуя с суши, открывать новые пункты для высадки десанта и возвращать уничтоженные противником.

3) Не рекомендуем высаживать части одной армий на далеко отстоящих друг от друга плацдармах. Кроме того, что это вызывает раздробление и ослабление сил, это еще влечет совершенно непредсказуемое их перемещение в ответ на ваши приказы.

4) Противник очень чутко и быстро реагирует на ваши действия, умело маневрирует войсками. Такая его чувствительность дает вам возможность вводить его в заблуждение нанесением вспомогательных и отвлекающих ударов и ложными маневрами.

Можно скрыть направление главного удара и провести высадку очень организованно.

В бою же на обширных пространствах Вы наверняка сумеете переиграть компьютер, если Ваши армии не будут разбросаны в беспорядке.

5) Через пару дней Вы обнаружите, что использование различных пунктов высадки десанта ведет к различному характеру боевых действий. В одном случае это затяжная позиционная борьба, в другом остродинамическое маневренное сражение, в третьем - глубинные рейды.

Может быть Вы научитесь жертвовать отдельными частями для нарушения согласованности в действиях противника.

Во всех случаях это говорит о богатстве стратегий в игре и о том, что она долго будет Вас занимать.

6) И последнее. Основной Вашей проблемой является эффект "бутылочного горла", когда Вашим частям придется маневрировать на ограниченных пространствах. Поэтому обратите самое большое внимание на то, в каком порядке Вы отдаете своим армиям приказы на перемещение, двигаться они будут в том же порядке, и очень важно, чтобы на пути частей не стояли части другой армии, блокируя и запирая их.

OVERLORDS



"OVERLORDS" - это стратегическая игра для двух играющих. В ее основу положены известные (к сожалению не у нас) настольные игры.

Оба игрока играют одновременно. Существуют несколько возможных режимов игры:

- оба игрока играют на клавиатуре (клавиши задаются);
- оба игрока играют джойстиком (это возможно конечно только для Синклер-джойстиков);
- один игрок работает с клавиатурой, другой - с джойстиком.

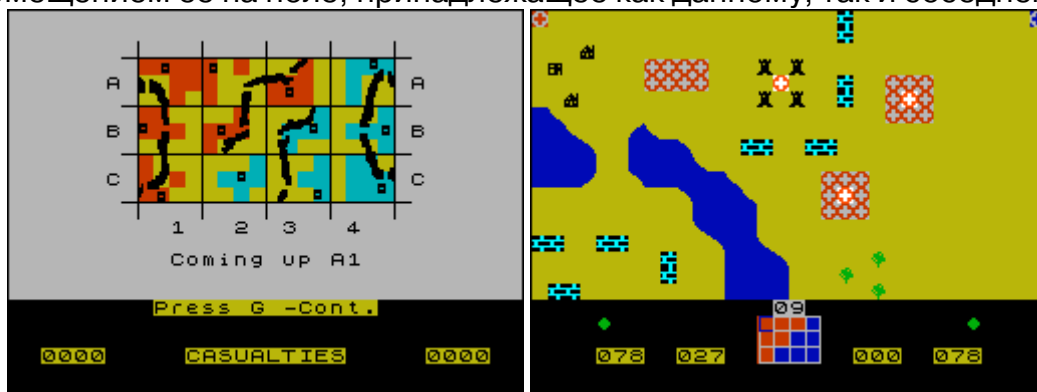
Игра имеет дополнительный режим, в котором два отдельных "Спектрума" соединяются между собой в локальную сеть.

Порядок игры.

Карта игрового поля довольно велика и разнесена на 12 отдельных экранов (6 в укороченной версии игры). Поля каждого экрана несколько перекрывают экраны, смежные с ним.

Первый играющий играет красными фигурами, второй - синими. На каждом экране имеется курсор для каждого игравшего (соответствующего цвета). Оба игрока могут двигать фигуры одновременно. Курсор противника не оказывает влияния на перемещение ваших фигур и наоборот, фигуры, находящиеся за пределами экрана не могут перемещаться, но в

рамках экрана ограничений на передвижение нет. Вывод фигуры в соседний экран выполняется помещением ее на поле, принадлежащее как данному, так и соседнему экрану.



Цель игры.

Победителем считается игрок, первым уничтоживший неприятельского короля (KING) или захватившего оговоренное число замков (CASTLE) - от 7 в простейшей версии игры до 12.

Фигуры.

Каждая армия имеет фигуры трех типов (они изображаются одинаковыми символами):

1. Солдаты. Это основная фигура, их большинство.
2. Генералы (отличаются от солдат тем, что изображены более ярко), их количество - примерно 1/8 всей армии.
3. Король - он один. Изображение короля - обратно изображению солдат, первоначально короли соперников расположены на экранах A2 и C3.

Территория.

Местность имеет графические элементы, такие как дома, амбары, церкви, деревья и т.п. Фигуры не могут перемещаться по этим объектам. Реки и озера не пересекаются.

В каждом экране имеются замки. Их надо захватывать, что выполняется помещением фигур в центр всех четырех башен замка.

Сражения.

Когда фигуры противников входят в соприкосновение, между ними происходит сражение. Победитель определяется сравнением сил сторон. При этом учитывается суммарная сила всех фигур, имеющих контакт с той, которая вступила в сражение.

В обороне все фигуры имеют равную силу, равную 1. В атаке генералы имеют силу, равную 3.

Пополнение армии.

По мере развития игры, стороны могут рекрутировать себе войска. Компьютер оценивает вашу популярность у местного населения и наличие людских ресурсов.

Генералы, находящиеся в замках, могут рекрутировать солдат, а король в замке может рекрутировать генералов.

Информация на экране.

Ниже игрового поля на экране имеется информационная панель для каждого игрока.

1. Зеленый цвет показывает, что игрок может продолжать двигать свою армию на экране. При красном цвете клавиша, управляющая перемещением войск, становится неоперабельной.

2. Число, показывающее относительную силу армии.

3. Число, которое показывает количество фигур игрока на данном экране.

Кроме этого, в центре имеется информационная панель для обоих игроков.

1. Счетчик времени (7 периодов по 30 сек.)

По обоюдному соглашению игроков они могут отключить счетчик времени клавишей "G".

2. Сетка, показывающая 12 экранов. Текущий экран выполнен мигающим. Изображается также цветовой код, показывающий кому из играющих принадлежат какие замки на каждом из экранов.

Карта.

Прежде, чем изображать каждую зону, компьютер показывает карту местности, всю игровую область и расположение армий.

Повторение игры.

Чтобы переиграть игру, используйте выбор "H-EXIT", когда Вам будет предложено, и затем нажмите клавишу "H" еще раз. Компьютер хранит в памяти серию моментальных "снимков" карты, на которых доказано как развивалась игра.

Запись игры на ленту.

После выхода из игры через выбор "H-EXIT", игроки могут выгрузить/загрузить игру.

Правила ведения войны.

Наперед заданных правил игры нет. Игроки свободны в их выборе. Единственное условие - это то, что пользование клавишами "G" и "H" должно происходить по обоюдному согласию.

Дополнительная информация при работе с двумя "Спектрами".

1. Программа рассчитана на то, что компьютеры сопрягаются в сеть через фирменный интерфейс (ZX-ИНТЕРФЕЙС-1).

Обратите, пожалуйста, внимание:

1. Рекомендуются загружать игру в ОБА "Спектрума".

2. Если Вы загружаете отложенную игру, то в ОБА "Спектрума" должна быть загружена идентичная копия.

3. Для некоторых интерфейсов нажатие клавиши "SPACE" может вызвать сброс программы. Если Вы работаете с двумя компьютерами, не нажимайте клавишу "SPACE".

Порядок работы после загрузки программы.

S - записать игру на ленту;

L - загрузить с ленты отложенную позицию;

G - начать игру;

H - переиграть.

Если Вы нажмете G (начать игру), то подучите вопрос: "Играете ли Вы на двух соединенных вместе компьютерах?". Отвечайте - "n" ("нет").

Далее вам предложат назначить устройство управления игрой для 1-го играющего:

1. Клавиатура (клавиши уже заданы);

2. ZX - интерфейс-2 (Синклер-джойстик);

3. Кемпстон-джойстик;

4. Задать свои собственные клавиши.

Если Вы нажмете "4", то компьютер запросит от Вас ввод управляющих клавиш.

RIGHT - вправо LEFT - влево

UP - вверх DOWN - вниз

"CARRY" - этой клавишей фигура "подхватывается" для перевода ее на другое поле.

Вслед за этим аналогичным путем запрашиваются данные для второго играющего.

Затем следует запрос:

"SELECT SPEED OF PLAY" (выберите скорость игры: 1 - медленно; 5 - быстро).

Следующий выбор - выбор цели игры.

1. Игра до победы над королем противника.

2. Игра до захвата 12 замков.

3. "Укороченный" вариант.

ZX LPRINT III

Сегодня в нашем техническом разделе научно-технический кооператив "ПЛЮС" представляет Вам разработку универсального интерфейса для поддержки принтера. Это аналог зарубежного интерфейса ZX-LPRINT-III, адаптированный под отечественную элементную базу.

Прежде, чем предоставить слово НТК "ПЛЮС" мы дадим небольшую справку из зарубежных рекламных изданий, добавив от себя, что то, что в ней сказано о зарубежных принтерах, относится и к наиболее распространенным в нашей стране ROBOTRON, D100 и др., причем с любым интерфейсом (как с последовательным, так и с параллельным).

ZX LPRINT III Interface

Фирма: EUROELECTRONICS

Цена: 29.95 ф. ст.

Позволяет подключать "Спектрум 16", "48", "+" или "128" (в режиме 48) почти к любому матричному принтеру или типа "ромашка". Поддерживает команды LPRINT, LLIST. Команда COPY работает без вспомогательного математического обеспечения на принтерах: EPSON, STAR, BROTHERS, WALTERS, MANNERSMAM TALLY, TAXAN KAGA, CANON, SHINWA, CITIZEN. На SEICOSHA GP-700 команда COPY работает в полном цвете. На EPSON и совместимых может выполнять копию экрана в двух размерах. На CGP115, MCP40, MCP80, SCP80 копия может быть четырехцветной. Совместим с ZX-интерфейсом-1, модемом VTX 5000 и дисковыми интерфейсами OPUS DISCOVERY и BETA.

Дополнительно добавим, что этот интерфейс давно и прочно завоевал популярность в мире и не удивительно, что многие фирмы, выпускающие прикладное программное обеспечение, ориентировались на него. Так, например, текстовый редактор THE LAST WORD TWO в своей поставочной версии рассчитан на работу именно с ним. Графический редактор ARTSTUDIO в числе прочих поддерживает работу и с этим интерфейсом. Одним словом, это тот интерфейс, который позволяет решить большинство практических проблем возникающих перед нашими читателями, когда они начинают подумывать о работе с принтером.

Адрес НТК "ПЛЮС" Вам должен быть известен:

127566, Москва, И-566, НТК "ПЛЮС"

- а сейчас предоставляем слово его сотрудникам.

Вопросы стыковки компьютера с принтером уже неоднократно освещались в самой разнообразной литературе, в том числе и в нашей методической разработке N1 "Интерфейсы Спектрума", но судя по почте наших читателей, интерес к этой теме не ослабевает.

В самом деле, отсутствие у базовой модели "Спектрума" встроенного интерфейса и программы поддержки принтера порождает у пользователей этого замечательного компьютера немало проблем. Положение осложняется еще и тем, что трудно найти универсальное решение, способное удовлетворить всех пользователей. В стране развелось невообразимое множество самодельных компьютеров, добавьте к этому разнотипность имеющихся принтеров и интерфейсов. Мы полагаем, что продолжая развивать эту тему в своих публикациях, мы поможем Вам найти подходящее именно Вам решение этого вопроса или хотя бы подготовим Вас к решению его своими силами.

Рассмотрим некоторые аспекты, связанные с подключением принтера к "Спектруму". Как Вы понимаете, кроме аппаратных средств, то есть самого интерфейса, необходима также и программная поддержка этого интерфейса, то есть программа драйвер. Как же этот вопрос решен в фирменном компьютере?

Для вывода на печать текста и изображений БЕЙСИК "Спектрума" имеет три команды - LPRINT, LLIST и COPY. Первые две команды почти полностью аналогичны командам печати

на экране с той разницей, что при их выполнении открывается канал "P", в то время как по командам LIST и PRINT открываются каналы "S" или "K" в зависимости от того в главную часть экрана идет печать или в окно системных сообщений (в нижние две строки).

Открытие этого канала изменяет местонахождение программы поддержки и направляет поток данных на вывод с помощью программы, обслуживающей фирменный ZX-принтер. Работа ZX-принтера аналогична работе телевизора с механической разверткой. Головка с единственной иглой постоянно перемещается взад и вперед по бумаге и, в зависимости от наличия точки (для экрана - пиксела) печатает ее на бумаге. Бумага подается после каждого прохода на 1/8 строки.

В более поздних моделях, работающих с термобумагой, бумага прижата к проворачивающемуся на 1/8 строки барабану с помощью головки, имеющей 256 нагревательных элементов с малой инерционностью нагрева и остывания. Здесь сразу производится печать 256 позиций пикселей. И в том и в другом случае печать производится на узкой бумажной ленте и представляет собой копию экрана, т.е. 22 строки по 32 символа или для графической копии 256X176 точек.

Программа поддержки ZX-принтера находится в ПЗУ (см. раздел "Секреты ПЗУ" в данном номере "ZX-РЕВЮ"), что освобождает пользователя от необходимости загружать ее с магнитофона и значительно экономит время.

Т.к. в нашей стране ZX-принтеры не слишком распространены, то пользователь вынужден решать задачу подключения реально существующего у него принтера. Для этого приобретается какой-либо подходящий интерфейс и загружается программа поддержки. Программа может быть размещена в любом месте памяти, где она не мешает работе других программ и не уничтожается при их работе. Например часто ее располагают в так называемом буфере принтера - это 256 байтов, следующие сразу за областью файла экранных атрибутов. Буфер принтера начинается с адреса 5B00H (23296). Разместить там драйвер возможно, если он не превышает 256 байтов. Далее изменяется стартовый адрес, хранящийся в системных переменных в соответствии с номером канала, который Вы открываете для вывода на печать. Теперь после команды начала печати компьютер будет обращаться к программе поддержки, которую Вы ему назначили и которая рассчитана именно на ваши аппаратные средства. Таков в общих чертах подход к этой проблеме в фирменном "Спектруме".

(Примечание "ИНФОРКОМа": Полагая, что вопросы работы с каналами и потоками компьютера тоже должны быть освещены поподробнее, мы запланировали дать в начале 1992 года обзорную статью "Каналы и потоки".)

Среди разновидностей отечественных моделей компьютера есть такие, в которых заложена аппаратная часть интерфейса принтера, например установлен порт пользователя, выполненный на KP580BB55.

В этом случае лишь необходимо подзагрузить программу поддержки этого порта. Иногда такую программу вписывают в свободную область ПЗУ компьютера. Это почти идеальный вариант, если эта программа подойдет для вашего принтера, если эта область не занята под кириллицу и если Вы готовы мириться с тем, что Ваше ПЗУ после этого не будет соответствовать фирменному со всей головной болью, возникающей на почве несовместимости.

Сегодня мы предлагаем Вашему вниманию вариант интерфейса принтера, адаптированного с фирменного ZX-LPRINT III- интерфейса. В некоторых моделях самодельных компьютеров он уже встроен и их владельцы смогут его опознать. Особенностью его является то, что программы поддержки записаны во входящую микросхему ПЗУ объемом 2К и потому не требуется загрузка с магнитофона.

Эта микросхема подменяет собой ПЗУ "Спектрума" тогда, когда выполняются команды печати.

Получив команду LPRINT, LLIST или COPY, процессор обращается к подпрограмме работы с ZX-принтером, который адресуется портом FB(HEX), при этом срабатывает дешифратор интерфейса ZX LPRINT и ПЗУ "Спектрума" заменяется "теневым" двухкилобайтным ПЗУ.

Для отключения ПЗУ "Спектрума" используется сигнал /ROMCS. В фирменном компьютере такой сигнал уже имеется и никаких доработок не требуется. В самодельных компьютерах использование этого сигнала потребует некоторой квалификации и знания принципиальной схемы. Подход к этой проблеме мы излагали в ZX-РЕВЮ N1 на стр. 17. В нашем случае надо найти сигнал выбора микросхемы ПЗУ, разрезать этот проводник и в разрыв включить резистор 680 Ом. Сигнал /ROMCS нашего интерфейса подключается после резистора к CS микросхемы ПЗУ "Спектрума".

Таким образом, при выборе ПЗУ интерфейса на CS ПЗУ "Спектрума" будет подано напряжение логической единицы, переключаящее ее в третье состояние.

Надеемся, что пользователи, взявшиеся за изготовление этого интерфейса, смогут разобраться с сигналом /ROMCS, в случае же затруднений мы постараемся дать более подробные рекомендации.

Кроме того, что интерфейс не требует программной поддержки, он имеет еще ряд достоинств. В нем может программно переключаться режим работы:

- по протоколу "Центроникс" (параллельный);
- RS232 (последовательный).

В последовательном режиме можно настраивать скорость передачи данных от 75 до 9600 бод.

Интерфейс также поддерживает графический режим работы, причем программно настраивается на один из нескольких типов графических принтеров.

После включения компьютера с присоединенным к нему интерфейсом, его надо инициализировать. Это делается командой CHR\$ 13 (Возврат каретки). Для этого дайте команду LPRINT ENTER. После инициализации интерфейс входит в текстовый режим печати 80 знаков в строке.

Для работы в графическом режиме одной из следующих команд выбирается тип графического принтера:

1. Seikosha GP 250x -
LPRINT CHR\$0; CHR\$1
2. Seikosha 100a -
LPRINT CHR\$0; CHR\$2
3. Epson (модели с высоким разрешением), Star DP510, Gemini и STX 80 -
LPRINT CHR\$0: CHR\$3
4. Walters WM80, Shinwa, CTI CP80 –
LPRINT CHR\$0: CHR\$4
5. То же, что и п. 3, но уменьшенная копия -
LPRINT CHR\$0; CHR\$5
6. Microline 80 и 82 и им подобные, CG 115 и MCP в 4-х цветах.
LPRINT CHR\$0; CHR\$6
7. Переход в текстовый режим -LPRINT CHR\$0; CHR\$0

Выбираемая команда и тип принтера отображаются на экране, перебор осуществляется клавишей SPACE, выбор - ENTER.

Интерфейс выполняет также следующие команды:

LPRINT CHR\$ 3 -

возврат каретки с переводом строки;

LPRINT CHR\$ 2 -

LPRINT CHR\$ 1; "A" -

здесь A - от 1 до 8. Столько знаков после этой команды будут восприниматься принтером как команды;

LPRINT CHR\$ 4; -

только один знак интерпретируется как команда принтеру;

LPRINT CHR\$ 5; -

знакогенератор компьютера отключается, все знаки после этого будут интерпретированы как коды ASCII, а не распечатываться как ключевые слова "Спектрума". Выход из этого режима и возврат к знакогенератору компьютера производится командой

COPY.

LPRINT CHR\$ 0; "S" -

переключение режима работы из параллельного ("Центроникс") в последовательный (RS232);

LPRINT CHR\$ 0; "P" -

обратное переключение режима работы интерфейса.

POKE 23697,n -

n задает число знаков в строке при печати. При инициализации устанавливается значение 80 знаков в строке.

POKE 23728,n -

n задает скорость передачи при работе в последовательном режиме. выбирается из следующей таблицы:

скорость, бод	n
75	1
110	2
150	3
300	4
600	5
1200	6
2400	7
4800	6
9600	9

В интерфейсе использованы следующие микросхемы:

DD1 - 573 РФ2, 573 РФ5

DD2 - 555 ЛЛ1

DD3 - 155 ЛП8

DD4 - 555 ТМ2

DD5 - 555 ИР23

стабилитрон - КС147А;

На схеме не указаны выводы питания для DD2,3,4:

- вывод 7 - 0 V, вывод 14 - +5 V DD 1 - вывод 12 - 0 V; 24 - + 5V не забудьте соединить вывод 21 с "землей".

DD 5 - вывод 10 - 0 V; 20 - +5V

Связь по последовательному интерфейсу RS232 осуществляется по упрощенному DTR протоколу по трем проводам:

- выход TXD - интерфейса соединяется со входом RXD принтера;

- вход DSR интерфейса - с выходом DTR принтера;

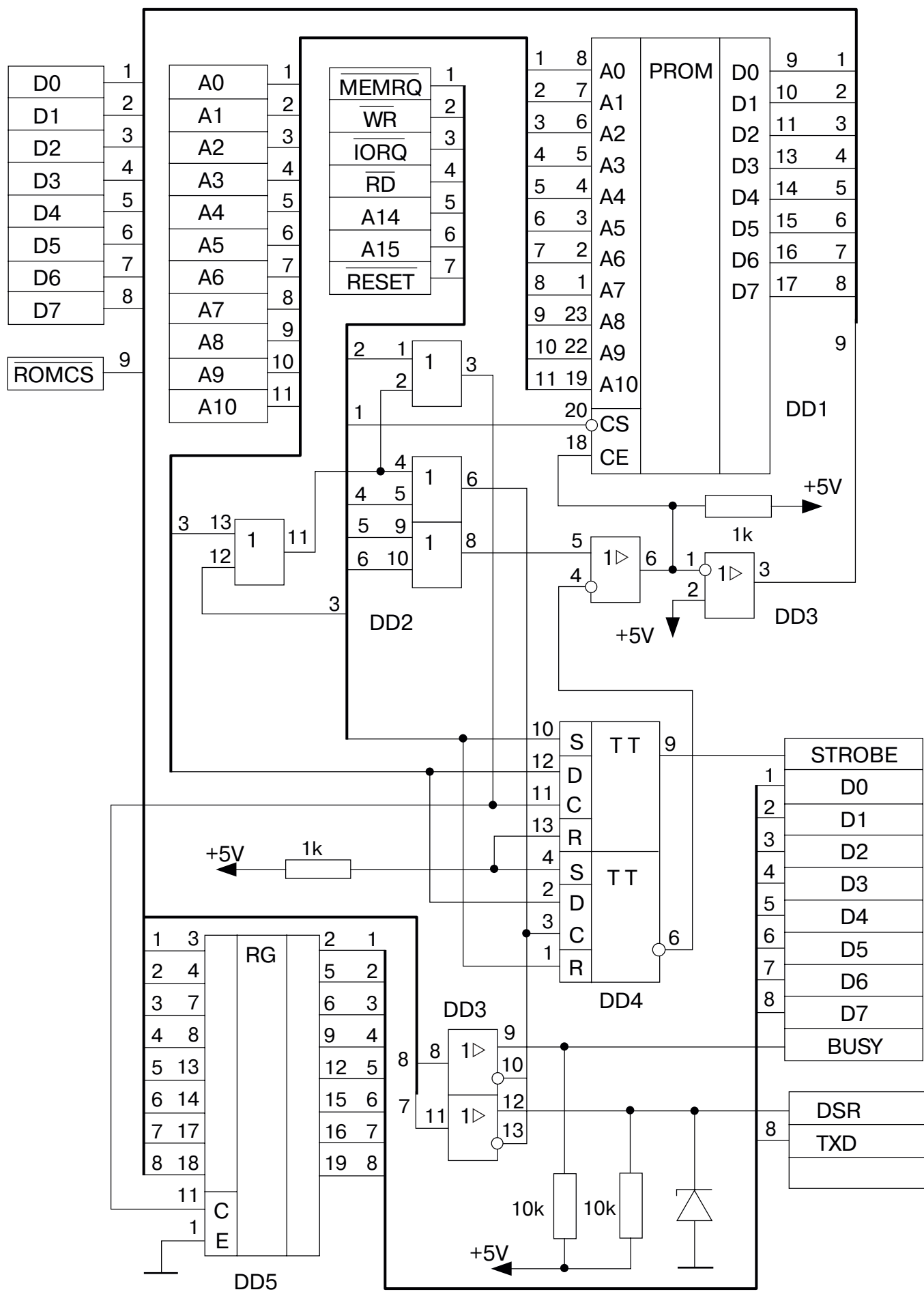
- третий провод соединяет интерфейс с принтером по "земле".

Подробнее об этом написано в МР N1 НТК ПЛЮС "Интерфейсы персонального компьютера Спектрум", стр. 14.

Обращаем Ваше внимание на то, что в данном интерфейсе выход данных TXD осуществляется в TTL уровнях и, если Ваш принтер не работает в этом режиме, необходимо дополнить интерфейс передатчиком шины с двуполярным выходом.

В параллельном режиме интерфейс соединяется с принтером не менее, чем 12 проводами - восемь шин данных D0 - D7, сигналы STROBE и BUSY и "земля" (см. МР N1, стр. 9).

В заключение добавим, что интерфейс активно использует для перемещения тела программы адресное пространство буфера принтера с 5B00 по 5BFF, которое в связи с этим нужно оставить свободным.



Ниже мы приводим распечатку содержимого ПЗУ интерфейса с контрольными суммами.

Если Вы пожелаете перед программированием ПЗУ завести код в память компьютера и сохранить его на ленте, а заодно и проверять правильность ввода по контрольным суммам, то можете воспользоваться той программой, которую мы давали в N3 из "ZX-РЕВЮ", внося в нее незначительные изменения. Эти изменения отмечены в листинге знаком !!!

Загрузчик рассчитан на то, что Вы будете собирать машинный код, начиная с адреса 60000. Если это не так, подправьте строки 1 и 30.

```
1 CLEAR 59999: LET add=60000: DIM a(10)                                     !!!
2 DEF FN A(a$)= (CODE a$(1)-48-(7 AND a$ (1) >"9"))*16+(CODE a$(2)-48-(7 AND a$(2)>"9"))
10 FOR f=101 TO 356                                                         !!!
12 RESTORE f
14 PRINT AT 0,0;f
15 READ a$
16 LET sum=0
20 FOR i=1 TO 2
22 LET b$=a$(2*i-1 TO 2*i)
24 LET a(i)=FN A(b$)
26 NEXT i
30 LET add=a(1)*256+a(2)+60000                                             !!!
32 LET sum=a(1)+a(2)
40 FOR i=3 TO 10
44 LET b$=a$(2*i-1 TO 2*i)
46 LET a(i)=FN A(b$)
48 LET sum=sum+a(i)
50 POKE add,a(i)
52 LET add=add+i
54 NEXT i
60 LET b$=a$(21 TO)
62 LET cs=FN A(b$)
64 LET cs1=sum-256*INT(sum/256)
66 IF cs<>cs1 THEN PRINT "ERROR!!!!!! line ";f:BEEP 2,10: STOP
68 NEXT f: STOP
```

Если Вы воспользовались этой программой, то данные перед вводом Вам надо оформить как строки DATA для ввода в память компьютера. Например первые две строки должны выглядеть только так:

```
101 DATA "00000000c33800C0970968"
102 DATA "0008217BCD224F5B211371"
```

```
0000 00 00 C3 38 00 CD 97 09 68
0008 21 7B CD 22 4F 5B 21 13 71
0010 5B 22 51 5B 3E 4C 32 F4 E9
0018 5B 3E 07 32 F2 5B 21 4B A3
0020 0A 22 2D 5B C3 B9 0D 4D AA
0028 4C 49 4E 45 38 30 12 00 CA
0030 A0 00 CD 12 0E C3 8B 0D 18
0038 D3 FB 18 0E CD 70 09 01 73
0040 6E 5B 70 2B 71 21 B1 5C 43
0048 CB AE C3 4D 00 0E 7B ED 47
0050 40 FB C9 3E 23 21 B1 5C E3
0058 CB AE C3 00 5B CD 5C 09 21
0060 18 D3 5F DB 7B FB 7B 21 97
0068 FA 5B CB 26 38 62 FE 06 4C
0070 38 31 28 15 FE 17 28 04 57
0078 FE 16 20 33 D1 E1 D9 E1 4B
0080 F5 1F 78 30 0A 79 47 18 1E
0088 06 F5 FD 7E 45 CB 3F FD 4A
0090 96 46 30 15 CD F8 1F F1 86
0098 FE 06 C8 3E 20 CD 59 5B 43
00A0 10 F9 C9 F3 DB FB C3 51 4F
```

```
00A8 0A D1 C8 47 18 ED 00 FE 95
00B0 A5 30 50 FE 90 30 7A FE 0B
00B8 80 30 4D FE 0D 20 26 E5 EB
00C0 FD 36 46 0B CD 6E 5B 21 FB
00C8 B1 5C CB 7E E1 3E 0A C8 0F
00D0 F5 CD 54 1F D2 00 0D DB BF
00D8 7B E6 80 20 F4 F1 D3 FB 8C
00E0 D3 7B D3 FB C9 FE 20 38 1B
00E8 BA CD 90 5B FE 60 28 B3 93
00F0 18 DE F5 FD 7E 46 FD BE 57
00F8 45 3E 0D D4 59 5B FD 34 41
0100 46 F1 C9 D6 A5 C3 10 0C 5B
0108 CD 90 5B CD E7 5B 21 92 83
0110 5C 22 FE 5B 47 CD 38 0B 3F
0118 0E 08 2A FE 5B 06 08 CB 8B
0120 06 17 23 10 FA 00 00 CD 38
0128 6E 5B 0D 20 ED C9 0F 18 FC
0130 9F CD 90 5B D6 90 CD E7 A2
0138 5B 2A 7B 5C 06 03 A7 17 5C
0140 10 FC 4F 09 22 FE 5B 18 38
0148 CF F5 21 F5 5B 06 04 23 AB
```

0150 7E CD 6E 5B 10 F9 F1 C9 28
0158 1B 4B 08 00 F3 01 FA 00 B5
0160 11 00 5B 21 62 08 ED B0 F5
0168 21 B1 5C CB 76 C4 7D 09 22
0170 01 0F 00 2A 4F 5C 09 36 95
0178 FC 23 36 0E C9 11 6E 5B 7F
0180 01 10 00 21 89 09 ED B0 E2
0188 C9 F3 F5 DB FB F1 CD BB 89
0190 0F F5 DB 7B F1 FB C9 F3 93
0198 01 A2 00 11 00 5B 21 D0 99
01A0 09 ED B0 21 B1 5C CB 76 B6
01A8 11 13 5B C4 80 09 01 0C 82
01B0 00 21 C4 09 11 ED 5B ED E5
01B8 B0 21 B1 5C CB 76 C8 AF 4F
01C0 32 1F 5B C9 1B 4D 08 1B C1
01C8 41 08 1B 4B 00 03 1B 40 D6
01D0 DB 7B FB CD 28 5B FB C9 36
01D8 CD 13 5B 21 B1 5C CB 7E 8B
01E0 3E 0A C8 F5 CD 54 1F D2 F8
01E8 00 0D DB 7B E6 80 20 F4 C6
01F0 F1 D3 FB D3 7B D3 FB C9 95
01F8 01 00 00 C5 01 4E 07 CD E2
0200 71 5B C1 C5 16 08 C5 D5 0C
0208 78 CD B0 22 D1 47 04 7E BB
0210 07 10 FD 00 CB 13 00 00 04
0218 00 C1 04 15 20 E8 00 00 FC
0220 00 00 7B CD 13 5B 00 00 D8
0228 00 0C 79 28 04 C1 4F 18 03
0230 D2 D1 3E 0D CD 08 5B 3E 8E
0238 BF B8 38 02 18 BD 01 55 16
0240 02 21 A2 5B C5 06 00 09 36
0248 C1 7E CD 13 5B 23 10 F9 F0
0250 C9 7B 2E 18 26 35 FE 05 3A
0258 28 3D FE 04 28 31 FE 03 1B
0260 28 47 FE 02 28 4A FE 00 41
0268 28 4D FE 01 28 16 FE 60 7A
0270 CA 53 08 D6 10 FE 06 30 B1
0278 1B 22 0A 5B 21 94 0A 22 FD
0280 45 5B 18 3C 22 0A 5B 21 1E
0288 BA 0B 22 45 5B 18 31 21 7B
0290 FA 5B 36 80 C3 47 0B FD AF
0298 7E 30 F5 E6 E0 32 FE 5B 8E
02A0 F1 F6 E0 FD 77 30 C3 3C 0C
02A8 08 21 B1 5C CB FE 18 E4 A5
02B0 21 B1 5C CB BE 18 DD 22 80
02B8 0A 5B 21 15 0B 22 45 5B 22
02C0 21 B1 5C CB EE C3 4D 00 B9
02C8 DB 7B AF CD 01 16 AF 11 73
02D0 12 5B CD 0A 0C DB FB C3 BB
02D8 5D 08 A0 16 01 00 45 4E 89
02E0 54 45 52 2D 41 43 43 45 06
02E8 50 54 2F 53 50 41 43 45 29
02F0 2D 4E 45 58 54 20 12 01 91
02F8 45 50 53 4F 4E 20 20 12 D1
0300 80 00 F3 01 3A 00 21 C8 9A
0308 0A 11 00 5B ED B0 C9 CD B4
0310 02 0B C3 00 5B 7B 21 B1 8B
0318 5C CB AE F5 FD 7E 30 E6 76
0320 1F 5F F1 FE 00 28 28 FE DE
0328 01 28 45 FE 02 28 4A FE 09
0330 03 28 4F FE 04 28 54 FE 29
0338 05 28 59 FE 06 28 5E FE 49
0340 53 28 63 FE 50 28 6C 21 24
0348 B1 5C CB AE C3 4D 00 7B 5C
0350 FD 77 30 CD 02 0B 21 6A 5C

0358 0B 22 10 5B 21 2D 0C 11 5E
0360 30 5B 01 06 00 ED B0 C3 55
0368 00 5B CD 32 0D C3 8B 0D 2D
0370 7B F6 20 FD 77 30 C3 6C D7
0378 0C 7B F6 40 FD 77 30 C3 9F
0380 9B 0C 7B F6 60 FD 77 30 9F
0390 30 C3 A0 0D 7B F6 A0 FD 41
0398 77 30 C3 E0 0D 7B F6 C0 23
03A0 FD 77 30 C3 FB 0D 21 B1 E4
03A8 5C CB F6 D3 FB CD 5A 0D CA
03B0 C3 4D 00 21 B1 5C CB B6 72
03B8 18 F3 7B E6 0F FE 09 30 6D
03C0 1A 47 21 FA 5B 36 00 37 07
03C8 CB 1E 10 FB 2A 6C 08 22 7F
03D0 0A 5B 21 51 0A 22 45 5B 76
03D8 C3 4C 0B 3E 08 18 E2 CD 02
03E0 97 09 21 F0 0B 11 28 5B 33
03E8 01 3D 00 ED B0 C3 43 0C D8
03F0 01 00 00 C5 C5 CD 38 25 A8
03F8 CD F1 2B 0D C1 20 10 FE E0
0400 20 38 0C FE 60 20 04 3E 28
0408 23 18 06 FE 80 38 02 3E 43
0410 20 CD 13 5B 04 78 FE 20 09
0418 30 04 C1 47 18 D5 C1 3E 44
0420 0D CD 08 5B 0C 3E 17 B9 7B
0428 D8 06 00 18 C6 54 45 58 D9
0430 54 20 20 CD 97 09 3E 1B 8E
0438 32 45 5B 3E 03 32 70 5B 4C
0440 CD 46 0C C3 00 5B 01 0A 8C
0448 00 21 52 0C 11 F0 5B ED 14
0450 B0 C9 1B 4C 02 1B 47 01 99
0458 00 1B 4C 03 3E 47 32 F7 74
0460 5B 3E 00 32 F8 5B 3E 08 C8
0468 32 F9 5B C9 CD 02 0B 21 B6
0470 95 0C 22 10 5B 21 82 0C 51
0478 11 30 5B 01 07 00 ED B0 BD
0480 18 C1 53 45 49 4B 32 35 F0
0488 30 CD 5C 09 CD 5C 0C 3E 61
0490 1F 32 BF 5B C9 CD 89 0C 2A
0498 C3 8B 0D CD 02 0B 21 D0 C2
04A0 0C 22 10 5B 21 53 0D 11 CF
04A8 30 5B 01 07 00 ED B0 18 F4
04B0 92 CD 5C 09 21 08 0F 22 D2
04B8 F6 5B 3E 01 32 EC 5B 3E 03
04C0 3E 32 CB 5B 21 CB FF 22 67
04C8 C3 5B 3E 1F 32 BF 5B C9 5C
04D0 CD B1 0C 18 C3 CD 97 09 A6
04D8 3E 1B 32 45 5B 21 52 01 7B
04E0 22 6F 5B 21 51 01 22 2D 92
04E8 5B 3E 07 32 35 5B 21 CB 3A
04F0 1B 22 4E 5B 21 CB FB 22 E3
04F8 50 5B 21 08 0F 22 F3 5B 4F
0500 18 AD 14 07 D6 04 8C 03 4E
0508 C0 01 D8 00 6D 00 33 00 46
0510 17 00 0A 00 FD 7E 76 FE 25
0518 0A D4 50 0D FE 00 CC 50 72
0520 0D 3D 87 4F 06 00 21 02 6E
0528 0D 09 5E 23 56 ED 53 FC 56
0530 5B C9 CD 5C 09 21 06 01 B3
0538 22 B6 5B 3E C3 32 B8 5B B6
0540 21 39 5B 22 B9 5B 3E C9 37
0548 32 E7 5B C9 A7 DB 7B FB 82
0550 3E 06 C9 53 45 49 4B 31 BF
0558 30 30 FD 7E 30 E6 E0 FE 2C
0560 00 CA 32 0D FE 20 CA 89 DF

0568 0C FE 40 CA B1 0C FE 60 9C
0570 CA 5C 09 FE 80 CA 5C 09 51
0578 FE A0 CA 5C 09 FE C0 CA D2
0580 12 0E CD 5C 09 3E 18 32 5F
0588 0A 5B C9 21 FA 5B 36 00 67
0590 2A 5D 5C ED 5B 59 5C A7 1C
0598 ED 52 DA 4D 00 C3 C1 0E 95
05A8 5B 01 05 00 ED B0 18 5F 22
05B0 CD 97 09 21 00 02 22 F5 5C
05B8 5B 3E 04 32 35 5B 3E F5 4F
05C0 32 43 5B 3E F1 32 46 5B 97
05C8 21 CB 13 22 47 5B 3E CD 9B
05D0 32 56 5B 21 13 5B 22 57 C0
05D8 5B 18 34 57 4D 20 38 30 B0
05E0 CD 02 0B 3E 3E 32 35 5B FD
05F0 32 F6 5B 3E 32 32 F8 5B 6D
05F8 C3 00 5B CD 02 0B 21 32 48
0600 08 22 10 5B 21 27 08 11 FC
0608 30 5B 01 0A 00 ED B0 C3 04
0610 00 5B CD 5C 09 21 21 0E F3
0618 11 A9 5B 01 2E 00 ED B0 FF
0620 C9 21 B3 5B D6 80 85 6F 68
0628 7E 18 10 80 82 81 83 88 62
0630 8A 89 8B 84 86 85 87 8C 76
0638 8E 8D 8F CD 6E 5B C9 CD 14
0640 97 09 CD 47 0E 18 C8 21 09
0648 53 0E 11 28 5B 01 66 00 AA
0650 ED B0 C9 21 88 5B CD 6D FA
0658 5B 01 FF 00 C5 16 03 CD 64
0660 77 5B 04 CD 77 5B 05 0D ED
0668 15 20 F4 37 CB 1B 37 CB B6
0670 1B 7B FE E0 30 02 CB B7 9E
0678 CD 13 5B 04 04 78 FE B0 E7
0680 28 04 C1 47 18 D6 D1 3E B7
0688 0D CD 08 5B 3E 00 B9 28 EA
0690 04 AF 47 18 C7 21 8B 5B 76
0698 06 03 7E CD 13 5B 23 10 93
06A0 F9 C9 C5 D5 78 CD B0 22 19
06A8 D1 47 04 7E 07 10 FD CB 27
06B0 1B C1 C9 1B 38 1D 1E 1B 04
06B8 36 7E CD 13 5B 23 10 F9 D9

06C0 C9 FD 7E 30 06 05 CB 3F 4F
06C8 10 FC 3C 5F FE 07 30 16 C0
06D0 CD F0 0E 01 FE 7F ED 78 84
06D8 1F 30 0F 01 FE BF ED 78 5F
06E0 1F 38 F0 C3 4D 00 1E 00 5B
06E8 18 E6 CD F0 0E C3 15 0B 9A
06F0 21 FF FF 2B 7D B4 20 FB 8C
06F8 C9 3E 7F DB FE 1F D8 3E 92
0700 FE DB FE 1F C9 00 00 00 C6
0708 00 00 00 00 DB FB 87 F3 5F
0710 E1 E5 3E 0E BC 20 62 3E A5
0718 B7 BD 20 5D C1 C1 E1 CD 40
0720 70 09 FD 7E 30 E6 E0 FE 0F
0728 00 CA DF 0B FE 20 CA 33 FE
0730 0C FE 40 CA D5 0C FE 60 8A
0738 CA 05 08 FE 80 CA B0 0D 1B
0740 FE A0 CA EB 0D FE C0 CA 2F
0748 3F 0E FD 7E 30 E6 1F 4F 9B
0750 3A FE 5B E6 E0 A9 FD 77 CD
0758 30 C3 AD 0B 7A F5 21 B1 4B
0760 5C CB 6E CC 5A 0D FD 7E AA
0768 45 FE 21 20 05 3E 50 FD 83
0770 77 45 CD 14 0D F1 C3 00 D5
0778 5B 2A 4F 5C 0E 0F 06 00 D2
0780 09 4E 23 46 21 FC 0E ED 5F
0788 42 28 D1 FD 7E 30 E6 1F 7A
0790 FD 77 30 21 B1 5C 36 00 9F
0798 CD 5A 0D FD CB 30 8E 3E 97
07A0 50 FD 77 45 AF 32 FA 5B E6
07A8 C1 C1 C1 C1 C1 3E 0D FD BC
07B0 36 76 06 F5 CD 14 0D F1 3D
07B8 C3 4D 00 C5 E5 F5 EE FF 5B
07C0 F5 CD F9 0E D2 4C 05 DB 8E
07C8 FB CB 77 28 F4 06 08 3E 74
07D0 FF D3 FB F1 CD EE 0F CB 2A
07D8 0F D3 FB 10 F7 CD EE 0F 8D
07E0 3E 00 D3 FB CD EE 0F CD 8A
07E8 EE 0F F1 E1 C1 C9 F5 2A 67
07F0 FC 5B 2B 7C B5 20 FB F1 B6
07F8 C9 C9 00 00 00 00 00 D8 69
0800 00 00 00 08

ИНФОРКОМ сообщает

Вместе с этим номером "ZX-РЕВЮ" Вы получите вместо привычного информационного листа бланк-заказ для подписки на 1992 год.

Этот лист составлен в расчете тех, кто не знает, что такое ZX-РЕВЮ. Вам же, нашему постоянному читателю, мы хотели бы сказать еще несколько слов о том, каким будет ZX-РЕВЮ в будущем году и чем оно будет отличаться от РЕВЮ-91.

Мы сохраним общую направленность нашего издания, его дух и стиль, останется тем же объем ZX-РЕВЮ, пока не изменятся и его цена, хотя если верить прогнозам специалистов о развитии общей экономической ситуации в стране, возможно в ближайшие месяцы придется ее значительно поднять, впрочем это будет зависеть и от количества подписчиков - чем их больше, тем проще выдержать пресс инфляции.

Кое-что в ZX-РЕВЮ мы изменим. Во-первых, изменится его структура. Мы не будем выделять отдельно Основной выпуск, Игровое приложение, Техническое и Рекламное приложения. Структура будет цельной.

Во-вторых, мы намерены уменьшить раздробленность материала, то есть каждый выпуск будет содержать меньшее количество рубрик, но в среднем рубрики будут иметь больший объем, чтобы уменьшить количество продолжений для крупных материалов.

На сегодняшний день портфель материалов будущего года в значительной степени сформирован и мы уверены в том, что по содержательности 1992-ой год должен превзойти год 1991-й. Значительно возрастет объем материалов для тех, кто начинает осваивать работу с машинным кодом, в том числе и для графических приложений. В обширных материалах немало полезных идей найдут и те, кто уже освоил и применяет машинный код в своей работе.

Вы знаете, что мы прекратили прием объявлений в Рекламное приложение и надо честно сказать, что получили за это несколько сердитых писем от читателей. Тем не менее истина остается истиной - объявлений мы получали очень и очень мало и нам обидно тратить площадь, которую могли бы занять более полезным и поучительным материалом. Мы надеемся, что данный выпуск наглядно показывает, что отказ от принудительного разделения ZX-РЕВЮ на приложения дает ощутимый эффект для читателей, хотя несколько объявлений (ранее оплаченных) мы не смогли не дать. Вместе с тем, мы понимаем, что нельзя лишать читателей возможности установления контактов друг с другом и приняли компромиссное решение;

Объявления давать будем, но бесплатно, скромно (в малом объеме) и не все, а только те, которые имеют общественный интерес. Приоритеты - следующие:

1. В первую очередь объявления от тех, кто может оказывать скорую техническую и консультационную помощь по аппаратным вопросам. Это очень необходимо многим читателям.

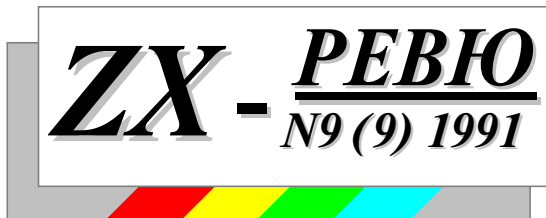
2. Объявления об организации клубов и о взаимодействии между клубами.

3. Объявления от инвалидов.

4. Объявления от учителей и преподавателей, связанные с применением компьютера в учебном процессе.

5. Информационные сообщения некоммерческого содержания от предприятий-производителей компьютеров,

6. Информационные сообщения некоммерческого содержания об опыте использования "Спектрума" в промышленных системах.



МП "ИНФОРКОМ", 107241, МОСКВА, Б-241, а/я 37

СЕКРЕТЫ ПЗУ

(продолжение)

Процедуры редактора.

Редактор вызывается в двух случаях: из главного вычислительного блока - когда пользователь вводит в систему БЕЙСИК-программу и из процедур, занимающихся обработкой команда INPUT.

0F2C-0F2F EDITOR

Это точка входа. Здесь запоминается на машинном стеке содержимое системной переменной ERR-SP (23613,23614).

0F30-0F37 ED-AGAIN

В ERR-SP устанавливается адрес 107F, где расположена процедура обработки ошибок, возникающих при редактировании строк.

0F38-0F6B ED-LOOP

Организуется цикл, принимающий код нажатой клавиши и выполняется распределение работы по процедурам, в зависимости от того, что это за код.

Опрашивается клавиатура вызовом WAIT-KEY (15D4). Принятый код находится в аккумуляторе. Устанавливается продолжительность звукового сигнала по системной переменной PIP(23609) и его высота. Вызовом BEEPER (03B5) выдается звуковой сигнал, свидетельствующий о том, что клавиша нажата, после чего на стеке выставляется адрес начала этой процедуры (0F38), что и обеспечит работу в цикле.

Далее анализируется принятый от клавиши код.

Если он больше 16H, т.е. это печатный или графический символ или токен ключевого слова - переход на ADD-CHAR (0F81).

Если он равен 6, то это запятая оператора PRINT и выполняется переход туда же.

Если он меньше 10H, но больше 6, то это клавиша редактирования и переход выполняется в ED-KEYS (0F92).

Если код больше или равен 10H, но меньше 16H, то это цветовой код и переход на ED-CONTR (0F60). Обратите внимание, что после цветового кода должен идти еще один код (операнд), определяющий значение данного атрибута цвета.

Последний случай - если код равен 16H (AT) или 17H (TAB), после которых должны следовать два операнда, определяющие координаты позиции печати. Если идет обработка БЕЙСИК-строки (а это определяется по выключенному седьмому биту системной переменной FLAGX), то их не надо исполнять и выполняется переход на процедуру ED-IGNORE (101E).

Если же идет обработка INPUT, то их надо исполнить. Принимается первый операнд вызовом WAIT-KEY (15D4) и работа продолжается в следующей процедуре.

0F5C-0FB0 ED-CONTR

Здесь обрабатываются управляющие коды цвета, а также AT и TAB.

Вызовом WAIT-KEY (15D4) вводится операнд (для кодов управления цветом он первый, а для AT и TAB - уже второй).

Вызовом MAKE-ROOM (1655) выделяется место в области памяти БЕЙСИКа для вставки этих управляющих кодов с операндами в БЕЙСИК-строку. Работа заканчивается переходом на ADD-CH-1 (0F8B)

0F81-0F8A ADD-CHAR

Задача - ввести принятый с клавиатуры символ в редактируемую Бейсик-строку или INPUT-строку. Выключением нулевого бита системной переменной MODE (23617) устанавливается курсор "K". Адрес курсора в экранном файле вводится из K-CUR(23643,23644) в регистровую пару HL и вызовом процедуры ONE-SPACE (1652) выделяется один байт памяти в редактируемой области для ввода туда принятого кода.

0F8B-0F91 ADD-CH-1

В выделенное место вставляется принятый код, запоминается положение курсора и выполняется возврат. Поскольку на стеке был выставлен адрес ED-LOOP (0F38), то это означает повторение цикла.

0F92-0F9F ED-KEYS

Сюда мы попадаем, если была нажата клавиша редактирования. Здесь в зависимости от того, что это была за клавиша, выполняется переход на соответствующую процедуру. Адрес перехода вычисляется с помощью нижележащей таблицы, в которой ведены "смещения" относительно базового адреса.

В качестве базового вводится адрес 0F99. Поиск по таблице выполняется прибавлением базового адреса к коду клавиши редактирования. Из полученного адреса извлекается величина "смещения", которая прибавляется к полученному адресу и дает адрес обрабатывавшей процедуры. Этот адрес выставляется на стеке и командой RET (возврат) осуществляется переход по вычисленному адресу процедуры.

0FA0-0FA8 - таблица клавиш редактирования.

Символ	смещение	адрес процедуры
07 EDIT	09	0FA9
08 курсор влево	66	1007
09 курсор вправо	6A	100C
0A курсор вниз	50	0FF3
0B курсор вверх	B5	1059
0C DELETE	70	1015
0D ENTER	7E	1024
0E SYMBOL SHIFT	CF	1076
0F GRAPHICS	D4	107C

0FA9-0FF2 ED-EDIT

Если Вы находитесь в режиме редактирования, то нажатие клавиши EDIT переводит текущую БЕЙСИК-строку в нижнюю часть экрана для редактирования.

Если Вы находитесь в режиме ввода по INPUT, то нажатие этой клавиши выполняет сброс введенной информации для повторного набора.

В регистровую пару HL вводится содержимое системной переменной E-PPC (23625,23626), т.е. номер текущей БЕЙСИК-строки. Далее, если обрабатывается INPUT-строка (определяется по 5-ому биту FLAGX) - переход на CLEAR-SP (1097) для очистки, а если БЕЙСИК-строка, работа продолжается.

Вызовом LINE-ADRR (1966) вводится начальный адрес этой строки.

Вызовом LINE-NO (1695) по адресу определяется номер строки.

Выполняется проверка номера строки на ноль. Если это так, то следует переход на CLEAR-SP (1097) для очистки области редактирования. (Нулевая строка не редактируется!)

Определяется длина строки и вызовом TEST-ROOM (1F05) выясняется, есть ли место для ее копирования из Бейсик-области в область редактирования.

В регистре HL устанавливается начальный адрес строки, а на стеке - адрес процедуры, обрабатывающей текущий канал ввода-вывода (из системной переменной CURCHL (23633,23634)).

Вызовом CHAN-OPEN (1601) открывается канал "R". В отличие от прочих стандартных каналов "K", "S" и "P", он является внутренним и не связан с внешними периферийными устройствами. Он связывает программную область БЕЙСИКа (где хранится программа) с областью редактирования.

Вызовом OUT-LINE (1855) выполняется "печать" БЕЙСИК-строки. Поскольку открыт канал "K", то эта "печать" представляет из себя ни что иное, как копирование текущей строки из программной области в область редактирования.

Далее выставляется положение курсора - в строке редактирования после ее номера и вызывается CHAN-FLAG (1615), которая восстанавливает предыдущее значение канала ввода/вывода и выставляет в соответствии с ним значения флаговых переменных.

0FF3-1000 ED-DOWN

Если выполняется обработка INPUT-строки - переход на ED-STOP (1001). В противном случае вызовом LN-FETCH (190F) определяется номер следующей строки и переходом на ED-LIST (106E) запускается автоматический листинг.

1001-1006 ED-STOP

В системной переменной ERG NR (23610) выставляется код 10H (что означает STOP IN INPUT) и выполняется переход на ED-ENTER (1024).

1007-100B ED-LEFT

Курсор перемещается влево посредством вызова ED-EDGE (1031). Далее работа продолжается переходом на ED-CHR (1011).

100C-1010 ED-RIGHT

Проверяется текущий символ. Если это CHR13 ("возврат каретки"), то возврат по RET. Если же нет, то увеличивается на единицу адрес положения курсора.

1011-1014 ED-CUR

Адрес курсора запоминается в системной переменной K-CUR (23643,23644) и выполняется возврат по RET.

1015-101D ED-DELETE

Речь идет о стирании символа, находящегося слева от курсора. Курсор передвигается влево вызовом ED-EDGE (1031). Текущий символ удаляется вызовом RECLAIM-2 (19E8).

101E-1023 ED-IGNORE

Процедура вызывается из ED-LOOP (0F38). Двойным вызовом WAIT-KEY (15D4) игнорируются два вводимых кода.

1024-1025 ED-ENTER

Речь идет о завершении ввода строки. Со стека снимаются адреса ED-LOOP (обеспечивающий цикл ввода символов) и ED-ERROR (устанавливающий месторасположение процедуры обработки ошибок, возникающих при редактировании).

1026-1030 ED-END

В системной переменной ERR-SP выставляется исходный адрес процедуры обработки ошибок. Если при редактировании не было ошибок, выполняется возврат в вызывающую программу, а если были, то переход на их обработку.

1031-103D ED-EDGE

Здесь речь идет о перемещении курсора влево. Во внимание должны быть приняты следующие обстоятельства:

- не достигнуто ли уже начало строки;
- не следует выставлять курсор между управляющими кодами и сопровождающими их операндами.

Вызовом SET-DE (1195) инициализируется регистр DE. В нем выставляется адрес системной переменной E-LINE, если идет редактирование БЕЙСИК-строк или адрес WORKSP, если идет обработка команды INPUT.

Сначала проверяется левый край. Если он достигнут - возврат на ED-LOOP (через адрес на стеке).

103E-1050 ED-EDGE-1

Далее обеспечивается неотделение управляющих кодов от их параметров. Вводится код текущего символа. Если это не код от INK до TAB - переход на ED-EDGE-2 (1051). Иначе пропускается один параметр.

1051-1058 ED-EDGE-2

Если есть еще параметры, переход на ED-EDGE-1 (103E), в противном случае возврат.

1059-106D ED-UP

Обслуживается код CHR\$ 11 ("курсор вверх").

Если идет обработка INPUT-строки, то этой клавише делать нечего - сразу выполняется возврат. Работа продолжается, если речь идет о строке БЕЙСИК-программы.

Вызовом LINE-ADDR (196E) и далее LINE-NO (1695) определяется номер строки, в которой находился курсор. Вызовом LN-STORE (191C) в системную переменную E-PPC (23625,23626) заносится этот номер.

106E-1075 ED-LIST

Вызовом AUTO-LIST (1795) выполняется перепечатаывание листинга на экране. Работа процедуры заканчивается переходом на процедуру, открывающую канал ввода/вывода CHAN-OPEN (1501) при установленном в аккумуляторе номере потока, равном 0, что открывает канал "K".

1076-107B ED-SYMBOL

Сюда мы попадаем, если был нажат SYMBOL SHIFT. Если речь идет о БЕЙСИК строке из программной области, то его быть не должно и переходом на ED-ENTER (1024) выполняется возврат.

Если же речь идет об INPUT-строке, то это какое-то ключевое слово и работа продолжается.

107C-107E ED-GRAPH

Для приема символа (графического или берущегося при нажатой клавише SYMB SHIFT) выполняется переход на ADD-CHAR (0F81).

107A-1096 ED-ERROR

Сюда мы попадаем в случае возникновения какой-либо ошибки при редактировании.

При работе с любым иным каналом, кроме "K" выполняется переход на конец редактирования ED-END (1026). В противном случае вызовом BEEPER (03B5) выдается звуковой сигнал, свидетельствующий об ошибке. Его длительность берется из системной переменной RASP (23606). Далее выполняется переход на ED-AGAIN (0F30).

1097-10A8 CLEAR-SP

Процедура служит для очистки области редактирования, что выполняется вызовом подпрограммы RECLAIM-1 (19E5) после того, как подпрограмма SET-HL (1190) выставит нужным образом регистровые пары HL и DE.

Здесь же устанавливаются в исходное положение системные переменные K-CUR (23643,23644) и MODE (23617).

Подпрограмма "KEYBOARD INPUT".

Эта подпрограмма имеет очень важное значение. Она возвращает в регистре A процессора код каждой нажатой клавиши. Но основное значение этого блока процедур состоит в том, что именно этот блок стандартно привязан к входному каналу "K" (клавиатура), о чем мы будем еще говорить в следующих выпусках.

10AB-10D7 KEY-IBPUT

Сначала проверяется 3-ий бит системной переменной TV-FLAG (23612). Если он включен (включается в WAIT-KEY (15D4)), то это сигнал на копирование строки из области редактирования или из буфера INPUT в нижнюю часть экрана, что и делается вызовом ED-COPY (111D).

По 5-ому биту системной переменной FLAG (23611) определяется, была ли нажата клавиша. Если нет - возврат.

Если была, ее код считывается из системной переменной LAST-K (23560).

По 5-му биту TV-FLAG (23612) устанавливается, подлежит ли переделке нижняя часть экрана. Если да, то это делается вызовом CLS LOWER (0D6E).

Далее работа распределяется по процедурам. Если принятый код является печатным символом - переход на KEY-DONE (111B). Если это управляющий код - переход на KEY-CONTR (10FA).

Если принятый код свидетельствует о включенном режиме CAPS LOCK или несет информацию о режиме, в котором находится клавиатура, выполняется переход на KEY-M&CL (10DB).

Далее обрабатываются коды FLASH, BRIGHT и INVERSE и работа продолжается переходом на KEY-DATA (1105).

10DB-10E5 KEY-M&CL

Не оставляя обработку режима CAPS LOCK процедурам, обслуживающим клавиатуру, здесь при обработке ввода через INPUT выставляются необходимые флаги и выполняется переход на KEY-FLAG (10F4).

10E6-10F3 KEY-MODE

Точно также обрабатывается переключение клавиатуры в другой режим.

10F4-10F9 KEY-FLAG

Заключительные операции по установке флагов и возврат.

10FA-1104 KEY-CONTR

Здесь обрабатываются управляющие коды (кроме FLASH, BRIGHT и INVERSE), т.е. либо INK, либо PAPER. Какой из них конкретно – определяется по тому, была ли нажата при вводе этого кода клавиша SHIFT (по третьему биту кода). И в том и в другом случае переход на KEY-DATA (1105).

1105-110C KEY-DATA

Принятый управляющий код запоминается в системной переменной K-DATA (23565), в регистр DE устанавливается адрес 110D (зачем - будет ясно чуть ниже) и выполняется переход на KEY-CHAN (1113) для смены адреса процедуры, выполнявшей обработку информации, поступавшей из входного канала "K". Там он будет установлен на KEY-NEXT (110D).

110D-1112 KEY-NEXT

Поскольку после управляющего кода должен идти операнд, поэтому и возникла необходимость переключить канал сюда. Здесь этот операнд принимается в аккумулятор и вновь в регистр DE засылается адрес 10A8 (подготовка к восстановлению исходного адреса процедуры, обрабатывающей информацию от канала).

1113-111A KEY-CHAN

Здесь выполняется переключение каналов.

111B-111C KEY-DONE

Финишная процедура. По возвращении из нее флаг CARRY включен, если код от клавиши был успешно принят, а сам этот код находится в аккумуляторе.

Подпрограмма "LOWER SCREEN COPYING"

Эта подпрограмма также представляет из себя пакет процедур. По своему назначению она является вспомогательной и вызывается для работы тогда, когда надо взять редактируемую строку из области редактирования или из буфера INPUT и поместить в нижнюю часть экрана.

111D-1150 ED-COPY

Основная процедура этого блока.

Вызовом TEMPS (0D4D) вводит постоянные цветовые атрибуты.

Устанавливает необходимые флаги.

Запоминает на стеке системные переменные S-POSNL(23690,23691), ERR-SF(23613,24614).

Устанавливает в ERR-SP в качестве адреса процедуры обработки ошибок значение 1167, что соответствует процедуре ED-FULL.

Запоминает на стеке содержимое ECHO-E.

Вызовом SET-HL (1195) инициализирует регистр HL.

Вызовом OUT-LINE2 (187D) печатает строку.

Вызовом OUT-CURS (18E1) печатает курсор.

Вызовом TEMPS (0D4D) вновь вводятся постоянные цветовые атрибуты.

1150-115D ED-BLANK

Если напечатанная строка не равна по длине экранной строке, то это имеет некрасивый вид, поскольку могут различаться по цвету то, что печатается и то, что в этом месте экрана было до печати.

Поэтому необходимо дополнить напечатанную строку до полной посредством печати пробелов в установленном цвете.

Если в этом нет необходимости, то сразу выполняется переход на ED-C-DONE (117C).

Для исполнения печати пробелов - переход на ED-SPACES (115E).

115E-1166 ED-SPACES

Вызовом процедуры PRINT-OUT (09F4) выполняется печать пробела. Далее следует переход на ED-BLANK (1150), что обеспечивает работу в цикле.

1167-117B ED-FULL

Здесь обрабатываются возникшие ошибки. Вызовом BEEPER (03B5) выдается звуковой сигнал с продолжительностью, установленной в системной переменной RASP (23608) и следует переход на финиш ED-C-END (117E).

117C-117D ED-C-DONE

При нормальном завершении работы перед выходом необходимо снять запомненные

на стеке параметры и восстановить их в регистрах процессора, что здесь и делается.

117E - 118F ED-C-END

Восстанавливаются значения системных переменных ERR-SP (23613,24614), ECHO-E (23682, 23663) и др., после чего выполняется возврат в вызывающую процедуру.

Подпрограмма "SET-HL AND SET-DE".

Здесь содержатся всего две вспомогательные процедуры.

1190-1194 SET-HL

В регистровую пару HL засылается содержимое системной переменной WORKSP (23649,23650) минус единица. Т.е. после этого регистр HL указывает на конец рабочего пространства.

1195-11A6 SET-DE

В режиме редактирования здесь в регистровую пару DE устанавливается из системной переменной E-LINE (23641,23642) адрес начала области редактирования и выполняется возврат.

Если Вы не находитесь в режиме редактирования, здесь устанавливается значение взятое из WORKSP (23649,23650).

Подпрограмма REMOVE-FP

Эта подпрограмма имеет сомнительное отношение к процедурам редактора, поскольку в нем нигде не вызывается и наверное осталась в этом блоке процедур по каким-то историческим причинам.

Ее назначение - удалить из БЕЙСИК-строки все числа, записанные в формате с плавающей точкой (в пятибайтной интегральной форме). Кстати, об этом формате Вы можете почитать в нашем трехтомнике, поскольку в рамках материалов "Секреты ПЗУ" мы до этого дойдем еще не скоро.

В двух словах: действительные числа с плавающей точкой хранятся в БЕЙСИК-строках в виде пяти байтов, перед которыми стоит код 0E.

Эта процедура отыскивает такой код в БЕЙСИК-строке и, проходя шесть раз, удаляет шесть ненужных байтов - пять байтов числа и шестой байт - сам код 0E.

Удаление байтов выполняется вызовом процедуры RECLAIM-2 (19E8).

А сейчас мы прощаемся до следующего выпуска, в котором начнем рассматривать наиболее важный раздел ПЗУ - блок исполняющих процедур.

BETA BASIC

Мы продолжаем печатать инструктивных материалов по работе с языком программирования BETA BASIC. Начало - см. стр. 134.

20. Команда: PROC

Структура: PROC имя

Клавиша: 2

См. также DEF PROC, END PROC.

PROC - это сокращение от слова процедура.

Процедуры похожи на GO SUB с именем, но имеют то преимущество, что Вам не надо беспокоиться о том, где в программе находится определение процедуры. Компьютер найдет ее где угодно, при условии, что DEF PROC - первое выражение в строке (это ограничение несколько увеличивает скорость вычислений).

Имя процедуры должно удовлетворять тем же условиям, что и имя переменной. Определение процедуры может иметь длину в сколько угодно строк - оно заканчивается оператором END PROC. Так же, как и DEF FN, DEF PROC игнорируется программой, если они не были вызваны через PROC. Программа в своих вычислениях просто перепрыгивает через них. Все переменные, используемые в основной программе, доступны и для процедуры, а все переменные, которые создает или изменяет процедура, доступны для главной программы.

```
10 FOR n=1 TO 20:PROC draw square: NEXT n
200 DEF PROC draw square: LET side=RND*20+20
210 PLOT RND*215,RND*135:DRAW side,0:DRAW 0,side
220 DRAW-side,0:DRAW 0,-side:END PROC
230 PRINT "Finished"
```

Идеальная "структурированная" программа должна содержать серию заданных процедур, каждая из которых выполняет определенную работу и может быть оттестирована независимо. Они вызываются в нужном порядке из основного тела программы, например:

```
100 PROC Set up board (установить игровое поле)
110 PROC Play game (играть в игру)
120 PROC Show Score (показать счет)
130 STOP
```

Если процедура не была задана, то использование PROC вызовет сообщение:

W: "Missing DEF FROC" (W: "Отсутствует DEF PROC")

Это же произойдет, если Вы используете END PROC, не задав DEF PROC. исключение составляет тот случай, когда в стеке есть доступный адрес возврата. END PROC не знает, что это не адрес PROC и "благополучно" возвратится к указанной строке. Если Вы забудете дать END PROC, то программа выдаст сообщение:

X: "No END PROC" (X: "Отсутствует END PROC")

Программа будет пытаться "перепрыгнуть" через определение процедуры, но не найдет где она кончается.

21. Команда: RENUM

Структура: RENUM // (начало TO конец) // //LINE новое начало// // STEP шаг//

Клавиша: 4

Просто RENUM используется для перенумерации всей программы таким образом, что номер первой строки равен 10 и интервал между строками также равен 10. Примеры.

RENUM (130 TO 220) - перенумерация указанного блока;

RENUM (130 TO) - перенумерация строк со 130 до конца программы;

RENUM (TO 100) - перенумерация всех строк программы, кроме нулевой, до строки 100 включительно.

Если блок нельзя перенумеровать так, чтобы после его перенумерации строки не "влезли" бы в строки остальной части программы, то Вы получите сообщение:

G: NO ROOM for line" (G: "нет места для строки")

В данном контексте оно имеет несколько иное значение, чем в обычном стандартной Бейсике.

Если Вам надо задать специфический номер первой строки, то используйте LINE. (Это ключевое слово, т.е. набирается не по буквам).

Шаг между строками задается с помощью STEP. Это тоже ключевое слово.

Примеры

```
RENUM  
RENUM LINE 100 STEP 20  
RENUM (1540 TO) LINE 2000  
RENUM (100 TO 176) LINE 230 STEP 5
```

Вы понимаете, что после перенумерации могут "пострадать" операторы перехода GO TO или GO SUB. Поэтому БЭТА-БЕЙСИК при перенумерации выявляет такие строки и изменяет их. Вместе с тем, могут быть особые случаи, например когда Вы делаете вычисляемый GO TO.

Сначала программа делает поиск выражений, которые могут нуждаться в перенумерации, например GO TO L*100. Такое выражение может указывать на строку, которую Вы собираетесь перенумеровать. Этот случай может оказаться слишком сложным для обычной процедуры перенумерации, поэтому перенумерация прервется с сообщением:

Y: "TOO hard" (Y: "слишком сложно")

Это дает вам возможность переделать выражение, может быть с использованием оператора ON (см. выше). Можете также на первых порах просто изменить это выражение как-нибудь, чтобы удовлетворить RENUM временно. Например, спрячьте недостойное выражение в кавычки

```
PRINT "GO TO L*100"
```

или поставьте перед ним REM

```
REM GO TO L*100.
```

После того, как все недопустимые выражения будут спрятаны, дайте еще раз команду RENUM. Теперь будут изменены в программе все ссылки на перенумерованные строки, включая: GO TO, GO SUB, RESTORE, RUN, ON, ON ERROR, TRACE, LIST, LLIST, LINE, DELETE.

Функцию CLOCK это не касается. Ее надо переделать самому! (Это потому, что после CLOCK может быть не только номер строки, но и число, показывающее режим настройки будильника).

Примечание. RENUM организует промежуточную таблицу для хранения данных, которая размещается в экранной области памяти. Она чистится после окончания перенумерации, поэтому не пугайтесь, глядя на экран.

22. Команда: ROLL

Структура: ROLL направление; // X координата, Y координата; ширина, длина //

Клавиша: R

См. также SCROLL

ROLL смещает весь экран или заданное окно на один пиксел вверх, вниз, влево или вправо. Все, что выйдет за границы окна, появится из противоположной границы. Другими словами, команда не уничтожает на экране ничего, она просто переорганизует информацию (в отличие от SCROLL). Направление ROLL задается числом 5, 6, 7 или 8 после команды. Поскольку каждая команда ROLL выполняет незначительное смещение, то лучше всего ее использовать в цикле. Нарисуйте график (DRAW) или дайте LIST, затем попробуйте:

```
100 FOR d=5 TO 8: FOR p=1 TO 100  
110 ROLL d: NEXT p: NEXT d: STOP
```

ROLL по диагонали может быть выполнен последовательной подачей ROLL вверх и ROLL вправо.

ROLL для окна экрана вводится подачей после кода направления еще четырех параметров: координат X и Y левого верхнего угла окна (в той же системе координат, что и для PLOT и DRAW), а также ширины окна (в символах) и его длины (в пикселах). Ширина может быть от 1 до 32, а длина от 1 до 176. ROLL может быть очень эффективной командой в играх для организации перемещения объектов или ландшафта. Чрезвычайно интересный, хотя и вызывающий легкое головокружение эффект может быть получен организацией

нескольких взаимопересекающихся окон таким образом, что экран начинает двигаться и дергаться в разных направлениях.

Интересный эффект дают следующие программы:

```
100 LIST:LIST:LIST
110 ROLL 5;0,175;32,88 :ROLL 6;0,175;16,176
120 ROLL 8;0,87;32,88 :ROLL 7;128,175;16,176
130 GO TO 110
```

или

```
200 FOR N=1 TO 7:LIST:NEXT N
210 FOR L=1 TO 175:ROLL 5;0,175;32,L:NEXT L
```

23. Команда SCROLL

Структура: SCROLL // направление//;/// X координата, Y координата; ширина, длина//

Клавиша: S

См. также ROLL

SCROLL по синтаксису очень похожа на ROLL. Основное отличие состоит в том, что SCROLL можно использоваться без параметров. В этом случае передвигается целиком весь экран вверх на один символ.

Если после SCROLL дать 5,6,7,8, то весь экран будет смещен на один пиксел в соответствующем направлении. Все, что выйдет за пределы экрана, пропадет совсем.

Новый экран, который "вытягивается" с противоположной стороны - чистый.

Можно передвигать информацию в части экрана - в окне. Для этого после команды должны стоять: код направления движения;

- координаты X и Y левого верхнего угла окна (в той же системе, что и для DRAW и PLOT);

- ширина окна (в единицах символов);

- его длина (в пикселах). SCROLL так же, как и ROLL, широко применяются в играх для создания движущихся изображений. Опробуйте примеры, приведенные для ROLL с командой SCROLL и сравните результат.

А вот пример программы, которая печатает строку и прогоняет ее по экрану:

```
100 LET A$="A NICE LONG STRING.." (красивая длинная строка)
110 FOR C=1 TO LEN A$
120 PRINT AT 10,31; INK 7; A$(C)
130 FOR P=1 TO 8: SCROLL 5;0,95;32,8: NEXT P
140 NEXT C
150 FOR P = 1 TO 255: SCROLL 5;0,95;32,8: NEXT P
```

Текст будет печататься символ за символом в одном и том же месте экрана белым цветом INK. Цвет может быть и другим, но он должен совпадать с цветом PAPER. Буквы будут медленно "выплывать" из квадрата, в котором они "невидимы".

Внутренний цикл FOR-NEXT сдвигает символ на одну позицию влево прежде, чем напечатается второй символ.

Строка 150 перемещает текст по экрану.

Примечание:

Если Вам покажется неудобным смешивать координатные системы "PRINT AT" и PLOT, то Вы можете воспользоваться усовершенствованной командой PLOT Бета-Бейсика и заменять строку 120 на:

```
120 PLOT 248,95; A$ (C)
```

Этот же принцип может быть использован, чтобы сделать текст привлекательным:

```
200 DATA "LONG AGO, IN A DISTANT GALAXY"
210 DATA "LOTS MORE TEXT, LOTS MORE..."
300 FOR L=1 TO 2: READ A$
310 PRINT AT 21,0; INK 7; A$
320 FOR P=1 TO 8: SCROLL 7: NEXT P: NEXT L
330 FOR P=1 TO 176: SCROLL 7 : NEXT P
```

24. Команда SORT

Структура:

SORT строковый массив или

SORT числовой массив или
SORT строковая переменная
Клавиша: M

Команда **SORT** организует строки, числа или буквы в восходящем или нисходящем порядке. Сначала обсудим ее применение со строковыми массивами. Вот программа, генерирующая 100 десятибуквенных случайных строк.

```
100 DIM A$(100,10)
110 FOR S=1 TO 100:FOR L=1 TO 10
120 LET A$(S,L)=CHR$(RND*25+65)
130 NEXT L:NEXT S: GO TO 200
140 SORT A$
200 FOR S=1 TO 100:PRINT A$(S): NEXT S
```

После того, как массив будет сгенерирован, что займет некоторое время, он будет распечатан. Теперь дайте **GO TO 140**, и он будет отсортирован и вновь распечатан (**RUN** не давайте). На сортировку уйдет примерно 1/5 секунды. Оно может немного возрасти, если Вы будете использовать более длинные строки. Количество строк более критично. 200 займут 0,7 сек., а 400 - 3 секунды.

Строки сортируются в соответствии с кодами символов. Пробел предшествует "A", а "A", в свою очередь предшествует "a". Если бы в строке 140 было записано

```
SORT INVERSE a$,
```

то порядок мог бы быть противоположным.

Попробуйте. Можно сортировать и отдельные блоки строк, например первые 20:

```
SORT A$(1 TO 20)
```

Команда **SORT A\$(30 to)** выполнит сортировку от 30-ой строки и до конца. Можно выполнять сортировку и не по первому символу. Так, команда

```
SORT A$( )(2 TO)
```

выполнит сортировку всего массива по второму и последующим символам.

Команда **SORT** дает возможность создавать быстроработающие базы данных. В этом контексте назовем весь массив файлом, а каждую отдельную строку - записью. Области каждой строки могут быть использованы для различных типов информации - это поля. Вам для этого могут понадобиться относительно длинные строки. Файл имен, адресов и других данных, например возраста, может быть организован так, что например первые 20 символов в каждой записи (строке) будут содержать имя персоны, следующие 20 - адрес, а последние - возраст. Поскольку любой возраст укладывается в диапазон от 0 до 255, то можно использовать: **LET A\$(S,41);CHR\$ age**, чтобы поместить возраст в запись. Такое хранение данных несложно и занимает мало памяти.

Предположим, что Вам надо хранить еще что-либо более сложное, например, банковский счет. Если Вы используете:

```
LET A$(S,41 to 46)=STR$ balance,
```

то информация будет помещена в строку, но она будет выровнена по левому полю, т.е. если вам надо ввести "9", то число "9" запишется в позиции 41, также как и цифра "9" от числа 900. В результате по этому полю не может правильно работать сортировка.

Поэтому вам нужно аккуратно записывать все данные с тем, чтобы десятичные точки располагались в одну колонку, тогда у каждой строки единицы, десятки и сотни будут записаны в одних и тех же позициях. Для этого можно пользоваться форматизирующей функцией **FN U\$**:

```
LET A$(S,41 TO 46)=FN U$("000.000",balance)
```

Полное описание этой функции Вы найдете, когда прочитаете **FN U\$, USING, FN C\$**.

Теперь Вы можете сортировать свои записи по возрасту, по счету в банке или в алфавитном порядке.

Обратите внимание на то, что поскольку код "1" меньше, чем код "2", то меньшие числа будут по результатам сортировки размещаться раньше. Можете пользоваться **SORT INVERSE**, чтобы большие числа шли раньше.

Сортировка работает также с обычными строками и с одномерными массивами.

```
INPUT S$: SORT S$: PRINT S$
```

Если в **INPUT** Вы дадите "Fred Bloggs", то на печати получите "BF degglors". Это не

выглядит очень полезным, но это дает возможность работы с числовыми данными, которые наиболее экономично хранятся в качестве строковых переменных. Например;

```
LET S$(position)=CHR$(data)
```

Это может также применяться к обычным одномерным или двумерным числовым массивам, которые используют тот же синтаксис, что и строковые массивы. Двумерный числовой массив можно рассматривать как таблицу, в которой первая размерность - ряд, а вторая - столбец.

Сортировка чисел идет в четыре раза медленнее, чем строк, потому что SORT должна проверить два различных числовых формата, а также учесть возможность наличия как положительного, так и отрицательного числа.

25. Команда: TRACE

Структура: TRACE номер строки.

Клавиша: T

Эта команда позволяет выполнять отладку Бейсик-программ. Она распечатывает текущую строку, текущее выражение в строке, а также избранные переменные, причем выполняет это с уменьшенной скоростью.

TRACE позволяет выполнять немедленный переход, до начала каких либо вычислений, к специальной подпрограмме. Эта подпрограмма может иметь две переменные (не ключевые слова) line и stat.

line - это номер строки.

stat - это номер выражения в строке.

Речь идет о строке, выполнение которой сейчас начнется. TRACE отключается во время работы своей подпрограммы, но снова включается после RETURN. Содержание этой подпрограммы - Ваше дело! Например:

```
9000 PRINT INVERSE 1 ;line; ":"; stat: RETURN
```

Включите эту процедуру в программу и вставьте инструкцию: TRACE 9000

TRACE вставляется с того места, с которого Вы хотите начать отладку. Если бы хотите отключить TRACE в каком-то месте программы, вставьте TRACE 0.

Использование TRACE дает печать выражений по мере их исполнения в формате, который используют системные сообщения. Чтобы отличить их от системных сообщений, воспользуйтесь оператором INVERSE. Трейсирование отключается после RUN, CLEAR и TRACE 0. Если Вы хотите замедлить процесс, вставьте PAUSE или другое средство по Вашему усмотрению. Переменные также могут выводиться на печать. Если Вам это нужно, то объявите их где-нибудь в начале программы, иначе может быть сообщение "Variable not found" ("переменная не найдена"). Чтобы не перепутать результаты печати от TRACE и результаты печати программы, мы Вам советуем использовать PRINT AT, но при этом надо запоминать место текущей печати. Иначе печать из главной программы все равно будет мешать. Это можно сделать например так:

```
9000 LET col=PEEK 23688:LET row = PEEK 23689
```

```
9010 PRINT AT 0,0; line; ":"; stat;"      ", "A$ =" ; A$; "      "
```

```
9020 POKE 23688,col:POKE 23689,row:RETURN
```

По этой программе печатается номер исполняемой строки и переменная A\$. Пробелы необходимы, чтобы исключить наложение печатаемых данных. Подпрограмма запоминает системные переменные, характеризующие позицию печати с тем, чтобы оно изменялось инструкцией PRINT AT только временно.

26. команда: UNTIL

Структура: UNTIL условие

Клавиша: K

См. DO и LOOP.

Задаёт условия выполнения циклов DO-LOOP.

27. Команда USING

Клавиша: U

Пример: PRINT USING format строка,число.

См. также FN U\$

Оператор USING, как и функция FN U\$ (см. далее) задают спецификацию формата, в котором должны печататься числа. По желанию вместо USING можно использовать FN U\$. Желаемый формат задается строковой переменной, в которой используются знаки # вместо ведущих пробелов, нули вместо предшествующих нулей и либо то, либо другое вместо цифр, стоящих после десятичной точки.

```
100 FOR n=1 TO 20:LET X=RND*100
110 PRINT X,USING"###,##";X:NEXT n
```

Обратите внимание, насколько аккуратнее получается печать отформатированных чисел, поэкспериментируйте с различными форматобразующими строками. Некоторые возможные форматирующие строки и результат их действия на число 12.3456 показаны ниже:

" # # . # "	12.3
" # # # . # "	12.3
" # # # # . # # "	12.35
" 000.00 "	012.35
" 00 "	12
" R00.00 "	R12.35
" 0.00 "	%..3

Предпоследний пример показывает, что в форматирующей строке можно использовать не только # и 0, но и другие символы. В этом случае они напрямую передаются на печать.

Последний пример показывает печать знака %, который говорит о переполнении заказанного формата.

FN U\$ применяется таким же образом, но:

вместо PRINT USING A\$; число

надо: PRINT FN U\$ (A\$,число).

28. Команда WHILE

Структура: WHILE условие

Клавиша: J

Позволяет вводить условие для завершения циклов DO-LOOP.

29. XOS,XRG,YOS,YRG

Эти четыре слова не являются ключевыми - это специальные переменные, которые позволяют изменять начало отсчета координат и масштаб координатной сетки, используемой командами PLOT, DRAW и CIRCLE.

XOS и YOS задают x-координату и y- координату начала системы отсчета, а XRG и YRG - масштаб по осям X и Y.

Эти переменные устанавливаются через CLEAR и RUN. очень часто изменение начала отсчета системы координат бывает более удобным, чем изменение выражений PLOT, например:

```
LET XOS = 126: LET YOS = 88
```

Эти команды сдвигают начало системы координат в центр экрана.

XRG нормально имеет значение 256 (т.е. Вы можете поместить до 256 точек на оси X). YRG нормально установлено в 176.

```
10 GO SUB 100: REM normal
20 LET XRG = 128: GO SUB 100
30 LET YRG = 88: GO SUB 100
40 LET XRG=256:GO SUB 100:STOP
100 CLS:PLOT 0,0:DRAW 50,10: DRAW 0,50
110 DRAW 50,0:DRAW 0,-50: PAUSE 100: RETURN
```

Сначала изображается нормальный квадрат, затем сжатый по оси X, далее сжатый по обеим осям и, наконец, он сжимается только по оси Y.

Нижеследующий пример показывает изображение графика функции синус.

```
100 LET XRG = 2*PI
110 LET YRG = 2.2
120 LET YOS = 1.1
```

```
130 FOR n=0 TO 2*PI STEP 2*PI/256  
140 PLOT n,SIN n:NEXT n
```

Заметьте, что позиция начала координат тоже масштабируется.

При использовании функции CIRCLE величина радиуса не масштабируется, поэтому получить растянутую окружность таким способом не удастся.

На этом мы пока прощаемся до следующего выпуска, в котором рассмотрим дополнительные функции Бета-Бейсика.

ЦЕНА ИДЕИ

Сегодня мы продолжаем (и завершаем) несколько необычный для нас раздел "Цена идеи". Начало см. в №6. стр. 131.

Чтобы до конца быть честными, мы должны признаться, что начало статьи вызвало неоднозначную оценку некоторых читателей, около десяти писем пришло от тех, кто счел, что мы уходим в сторону от главной линии и задумали что-то нехорошее.

Это не совсем так. Все гораздо проще. Дело в том, что мы уже несколько месяцев печатаем заметки Стива Тернера "Профессиональный подход", и этот материал не вызывает у читателей сомнений. А одна из ближайших статей его цикла как раз и посвящена вопросам маркетинга собственных разработок. Беда лишь в том, что его материал, как бы хорош он ни был, применим только в Англии и абсолютно неприменим у нас.

Мы много занимаемся маркетингом. Абсолютно все наши разработки проходят маркетинговое исследование у специалистов и нам есть что сказать по этому поводу, тем более что двухлетний опыт кое-чего стоит, вот мы и взяли на себя труд довести то, что хотел, но не мог Вам порекомендовать Стив Тернер.

Кстати, мы не занимаемся маркетингом чужих разработок (т. е. не приобретаем разработки с целью перепродажи) просто потому, что имеем много своих и достаточно загружены перспективными работами. Но мы охотно проводим маркетинговые исследования для других фирм, например можем Вам абсолютно точно сказать по какой цене Вам надо продавать ту или иную разработку, чтобы удовлетворить поставленной задаче. Можем точно предсказать количество покупателей, которое Вы приобретете при заданной Вами цене и избранном плане рекламной кампании, можем дать оптимальный (наиболее дешевый и наиболее эффективный) план рекламной кампании. Такие услуги стоят очень недешево, но заказчик их мгновенно оправдывает, можно ведь только поражаться тому, сколь много теряют разработчики только из-за неправильной оценки предполагаемого объема поставок и связанного с этим неверного определения цены единицы изделия.

Итак, в прошлом выпуске мы подошли к тому, что для того, чтобы подсчитать компенсацию, которую должен получить разработчик идеи от фирмы, принявшей на себя ее реализацию, надо определить ставку роялти, т. е. тот процент от дохода фирмы, который должен получать разработчик (по данному виду товара).

Мы также указали, что этот процент зависит от девяти факторов и находится в пределах от 0. 5% - до 15%.

Первый вопрос от умудренного читателя почему только 15%, а почему не половину? Здесь все очень просто - это мировой опыт с ним приходится считаться. Просто в любой стране есть налоги на прибыль и другие. И при увеличении ставки роялти выше 15% предприниматель-лицензиат попадает в положение, когда ему уже невыгодно заниматься никаким делом.

Сделаем прикидку. Предположим, что нормально преуспевающая фирма работает с уровнем рентабельности 30%-50%. Так оно в большинстве случаев и происходит, выше не дают подняться налоги на сверхрентабельность, а ниже фирма не сможет нормально развиваться (жить сможет, но не сможет расти) и тогда - застой и то, что мы имеем сейчас в экономической сфере.

Примем в среднем - 40%. Это значит, что на 100 рублей продаж себестоимость (т.е. затраты) составляют примерно 70 рублей и прибыль примерно 50 рублей.

$$30:70*100\%=42\%.$$

Проводя Вам отчисления в размере 15% от объема продаж фирма увеличит себестоимость на 15 рублей, т. е. доведет ее до 85 р. и рентабельность составит:

$$15:85*100\%= 17\%$$

Это еще терпимо, хоть и с трудом, если фирме больше просто нечего делать или если от вашей идеи она рассчитывает получить какие-то косвенные преимущества, например

нанести поражение конкуренту или сделать себе имидж.

При отчислении же 20%:

Себестоимость - 90 руб.

Прибыль - 10 руб.

Рентабельность - 11%

Это уже близко к невозможному.

Теперь рассмотрим, от чего зависит процентная ставка роялти и попробуем ее оценить.

1. Важность разработки. Использование новых разработок может принести сравнительно небольшое изменение существующих технологий или продукта, а может привести к технологическому прорыву. Это довольно широкий диапазон. Если, например, оборудование телевизоров стереозвуковой системой находится в нижней части спектра, то разработка волоконной оптики - где-то посередине, а транзисторов и сверхпроводящих керамических материалов - на самом верху.

2. Степень патентной защищенности, возможность соблюдения коммерческой тайны.

Защищенность, предоставляемая патентами или соблюдением коммерческой тайны, состоит из трех отдельных элементов:

2.1 - "открываемость" - относительная легкость или сложность, с которой конкурент может открыть, как функционирует Ваша технология.

2.2 - "повторяемость" - характеризует, насколько легко конкурент сможет повторить технологию после того, как он поймет, в чем заключается ваша идея.

2.3 - "избегаемость" или возможность достижения того же полезного эффекта без необходимости повторения запатентованной разработки. Фотоаппараты "Поляроид", видимо, представляют собой один из наиболее защищенных патентов в истории. Хотя выяснить химический состав пленки и проявителей, используемых фирмой при "мгновенной" съемке, было очень просто, но вот повторить эти составы оказалось не такой уж легкой задачей.

3. Влияние конкурентной ситуации на рынке на возможность использования новых разработок.

Имеются в виду те трудности, с которыми столкнется лицензиат при выходе на рынок с Вашей продукцией. Конечно, чем они выше, тем меньшую сумму он готов Вам заплатить за разработку.

Против Вас здесь работают следующие явления:

3.1 - если в сфере действия вашего нового товара (услуги) при увеличении объема производства однозначно снижаются и удельные затраты. Это не ошибка, действительно этот фактор работает против Вас. Ведь начиная производство Ваш клиент будет находиться в трудных условиях, а когда нарастит обороты и достигнет больших объемов и экономии затрат, то это будет не Ваша заслуга, а его, ведь он это сделает своими трудовыми и материальными затратами.

3.2 - если есть наличие множества различных товаров, удовлетворяющих одну и ту же функциональную потребность;

3.3 - если данная сфера характеризуется приверженностью покупателей одному сорту товара или верностью фирме, конечно в этих условиях труднее выйти на рынок. Кстати, если Ваш лицензиат известная фирма, пользующаяся доверием клиентов, то это тоже не Ваша заслуга, а фирмы;

3.4 - необходимость крупных капитальных вложений при выходе на рынок;

4. Необходимость инвестиций в дальнейшие НИОКР.

Создание прототипов, разработка планов производства и выход на рынок, пробные продажи, а также дизайн и привязка научных разработок к техническим возможностям - все это необходимо для того, чтобы технология получила реальную коммерческую ценность. Но это потребует времени и финансовых ресурсов, поэтому чем больше подобных работ лицензиар уже выполнил, тем большую цену он может запросить.

5. Стадия разработки.

В зависимости от стадии практической готовности разработки риск принятия ее к реализации может быть больше или меньше, снижение риска должно оказывать повышающее влияние на оценку технологии лицензиатом. Соответственно, цена должна увеличиваться, когда автор патента или идеи берет часть риска на себя, доводя разработку до уровня, максимально приемлемого к запуску в производство.

6. Конкурентные преимущества использования новых разработок.

Они определяются ценой, внешним видом, простотой использования и другими качественными параметрами конечного изделия.

Изделия, предоставляющие отчетливые преимущества по одному или нескольким параметрам, дадут возможность повысить цену на разработку.

7. Возможность рентабельности разработки.

Как отмечалось ранее, цена идеи или разработки - это часть прибыли, которую лицензиат надеется получить в результате ее внедрения. Понятно, что чем больше будет эта прибыль, тем больше будет и цена идеи как в абсолютном, так и относительном выражении, при увеличении рентабельности лицензиаты обычно с большим желанием идут на увеличение цены разработки.

8. Инновационная значимость, технологии и идеи, имеющие широкий спектр использования, обладают большей ценностью. Так, транзистор способствовал развитию нескольких новых отраслей, созданию тысяч новых продуктов, существенному улучшению существующих изделий, микросхемы также революционизировали практически все стороны экономики. Очевидно, что цена технологии, способных породить большое количество практических результатов, будет стоять намного выше, чем технологии и идеи с ограниченными направлениями и будущими возможностями применения.

9. Другие факторы. Рассмотрим три возможных фактора, которые снижают или увеличивают стоимость научной разработки для лицензиата, соответствующим образом изменяя и цену.

9. 1. Исключительность.

Если лицензиат приобретает исключительное право использования разработки или идеи, то получение этого конкурентного преимущества увеличивает цену или ставку роялти.

9. 2. Авансовые платежи.

Владелец новой технологии может потребовать уплату комбинированной цены, состоящей из части, подлежащей немедленной оплате и ставки роялти. Это особенно важно, когда лицензиат - многопродуктовая фирма. Лицензиар рискует, что внимание лицензиата может быть отвлечено необходимостью развития производства других продуктов. Авансовый платеж одновременно стимулирует лицензиата к внедрению и снижает риск, который несет лицензиар, ставка роялти при этом, конечно, снижается.

9. 3 Гарантии использования разработки.

Они предназначены для стимулирования заинтересованности внедряющей фирмы во внедрении Вашей разработки или идеи. Они выполняют схожую с авансовыми платежами функцию. Отсутствие же этих гарантий приводит к увеличению бремени риска, которое несет лицензиар, что увеличивает ставку роялти.

Итак, мы рассмотрели какими факторами определяется размер ставки роялти в

процентах от объема продаж.

Теперь, зная предполагаемый объем продаж и цену единицы продукции, (а это выявляется маркетинговыми исследованиями, хотя это лишь очень малая часть того, что называют маркетинговыми исследованиями) можно рассчитать и величину единовременной выплаты разработчику за коммерческую эксплуатацию его идеи.

$$S = R \cdot C \cdot N / 100$$

S - размер паушальной суммы;

R - ставка роялти в %;

C - расчетная цена единицы продукции;

N - объем продаж за период жизни товара (в рассматриваемой нами области этот период надо считать равным одному году - очень быстро меняется материальная база и конъюнктура рынка, кстати, отсюда и проистекает требование высокой рентабельности 30%-50%, т.к. часто приходится переоснащаться.)

По поводу этой формулы надо сделать ряд замечаний. Как Вы видите, размер причитающейся Вам суммы будет зависеть от планового объема продаж и от цены товара, а ни то ни другое от Вас не зависит и определяется только мощностью фирмы, которой Вы предлагаете свою разработку.

То, что фирма занизит цену, чтобы обидеть Вас, маловероятно, т.к. она накажет саму себя.

Возможно обратное - фирма излишне завысит цену так, что упадет объем продаж, зачем она это сделает? А хотя бы потому, что имеет ограниченные мощности и не справится с большим объемом заказов тогда Вам надо просто поискать более мощную фирму и предложить свою разработку ей.

Для нас важно, что Вы располагаете методикой расчета положенной Вам ставки роялти, а фирма - методикой расчета объемов поставок. На переговорах с фирмой Вы перемножаете одно на другое и определяете, что из этого вытекает, а далее можете попробовать переговорить и с другой фирмой.

ПРИМЕРЫ.

Теперь рассмотрим несколько реальных жизненных примеров. Предположим, что несколько человек выполнили несколько разработок в области программного, информационного и аппаратного обеспечения и планируют продать это какой-либо фирме (малому предприятию, кооперативу, заводу).

Разработка А.

Инженер разработал схему и исполнил опытный образец интерфейса локальной сети, который позволяет подключать два и более компьютера для совместной работы. Интерфейс имеет собственное ПЗУ, в котором "защита" его программная поддержка, оно подключается как "теневое" и не нарушает аппаратно-программную совместимость компьютера.

Образец предназначен для работы с компьютером ABCD.

Перед ним два варианта: А1 - продать схемное решение заводу выпускающему компьютеры ABCD. Понятно, что в этом случае оно останется привязанным только к этой модели.

А2 - продать идею фирме, занимающейся информационным обеспечением. Хотя за идеи платят меньше, чем за готовый образец, но зато потребителей будет больше, поскольку идею можно привязать к любой модели компьютера, а не только к ABCD.

Разработка В.

Программист, исходя из своих личных потребностей адаптировал программу THE LAST WORD 2 таким образом, что этот редактор стал работать не только с русским и латинским шрифтом, но еще и с таджикским.

Удовлетворив свои потребности, он решил еще и продать свою разработку фирме,

распространяющей программное обеспечение, поскольку справедливо полагает, что такая вещь должна послужить еще многим людям.

Разработка С.

Программист разработал программу, обучающую чему-то очень нужному. Пробный маркетинг среди друзей показал, что действительно программа нужна многим, он обращается к фирме с предложением продать право на тиражирование на ленте. Это вариант С1.

Фирма подумала и, понимая нужность и полезность этого предприятия, предложила выпустить эту программу в виде статьи в своем информационном сборнике, а не связываться с кассетой, это вариант С2.

Разработка D.

Ученый в течение двадцати лет занимался тем, что изучал методику лечения разных болезней лекарственными растениями, до этого этим тридцать лет занимался его отец. Собрана уникальная информация. Более 1000 рецептов. В сложных ситуациях предлагаются комплексные решения. И он предлагает свои материалы фирме для подготовки базы данных и распространения ее на дискетах для IBM-совместимых компьютеров. ("Синклер" здесь уже слабоват).

Итак, для рассмотрения влияния факторов на ставку роялти, договоримся, что будем оценивать каждый фактор от 0 до 15 баллов. Всего факторов - 9. Найдем коэффициенты К1 ... К9, возьмем среднее значение и пусть оно нам даст прикидочную ставку роялти, конечно это не точное значение, но хотя бы какая-то основа для переговоров с фирмой.

Окончательную же паушальную сумму определим после того, как фирма объявит ориентировочную продажную цену и планируемый объем продаж, поскольку наша методика расчета ставки роялти такова, что она объективно учитывает интересы фирмы, как и Ваши, то можете не сомневаться - занижать цены и объем продаж фирма не будет, ей нет смысла срывать наметившуюся с вами выгодную договоренность. Другое дело, если ее мощности или загруженность текущими работами не позволяют оптимально распорядиться вашей разработкой.

1. Важность разработки.

Все разработки в данном случае безусловно важны. Можно предположить, что разработка D существенно более важна чем остальные, если какая-то из них и предопределяет возможный прорыв, так это она.

Разработка В по-видимому находится в нижней части спектра, а А по-видимому превосходит С, т.к. обучение возможно и без компьютера, а вот соединить два компьютера без интерфейса - удастся вряд ли.

В то же время, вариант А1 направлен на усовершенствование только одной модели, а вариант А2 - любой модели компьютера и потому его важность значительно выше.

Примем в дальнейшую проработку следующий расклад:

Вариант	К1	Вариант	К1
A1	4	C1	3
A2	10	C2	3
B	1	D	11

2. Защищенность.

Здесь рассмотрим все три входящих фактора и возьмем средний между ними.

2.1. Открываемость.

Здесь все просто. Поскольку в вариантах А2, С2 и D речь идет о широком распространении информации, то и открывать уже нечего - этот фактор нулевой.

В остальных случаях он также близок к нулю.

Вариант	K21	Вариант	K21
A1	1	C1	1
A2	0	C2	0
B	2	D	0

2. 2. Повторяемость.

Реально неповторимой можно считать разработку D. Остальные повторимы - разница только в затраченном времени и в профессиональной подготовке. Поскольку варианты A2 и C2 открыты полностью, то здесь этот фактор - нулевой.

Вариант	K22	Вариант	K22
A1	2	C1	1
A2	0	C2	0
B	2	D	15

2. 3. Избегаемость.

Здесь нет нулевых факторов, поскольку даже для открытой информации нужны усилия для обхода. В то же время, вариант D уже не является абсолютно защищенным, поскольку есть и другие методики применения лекарственных растений, да можно лечить и другими методами.

Вариант	K23	Вариант	K23
A1	2	C1	2
A2	2	C2	2
B	3	D	10

В среднем фактор защищенности будет иметь значения:

Вариант	K2	Вариант	K2
A1	1,7	C1	1,3
A2	0,7	C2	0,7
B	2,3	D	8,3

3. Конкурентная ситуация.

Здесь также рассмотрим ряд составляющих факторов.

3.1. Соотношение между постоянными и переменными затратами.

Только в вариантах B и C1 практически все затраты переменные и не снижаются с ростом производства. В вариантах A1 и D есть значительные постоянные предварительные затраты, причем в варианте A1, связанном с переналадкой производственной базы, они особо критичны. В вариантах A2 и C2 они есть, но пренебрежимо малы.

Вариант	K31	Вариант	K31
A1	0	C1	15
A2	15	C2	15
B	15	D	5

3.2. Наличие аналогов.

Выходя на переговоры с Вами, фирма не знает стопроцентно, есть аналоги или нет и определенно, рискует. Ясно только, что в вариантах A2 и C2 наличие аналогов ее не волнует, а наиболее критичен этот фактор для варианта A1, поскольку если она пойдет по неверному пути, и конкурент примет в проработку более удачный вариант, то это ударит не по интерфейсу, а по всему компьютеру в целом и по перспективам развития целого предприятия.

Вариант	K32	Вариант	K32
A1	5	C1	7
A2	15	C2	15
B	7	D	8

3.3. Приверженность фирме.

Очень субъективный фактор, у нас в стране пока нет никаких приверженностей определенному сорту или приверженности определенной фирме.

Тем не менее, можно сказать, что если все равно где и у кого приобретать информацию, то программы приобретать уже сложнее, а перед приобретением компьютера люди очень долго и тщательно взвешивают варианты.

Вариант	K33	Вариант	K33
A1	5	C1	10
A2	15	C2	15
B	10	D	15

3. 4. Необходимость крупных капиталовложений при выходе на рынок.

Варианты A2 и C2 почти не требуют капиталовложений для выхода на рынок, варианты B и C1 требуют небольших капиталовложений в рекламу. Вариант D требует огромных затрат на рекламу, вариант A1 тоже, но меньше, зато там необходимы затраты на предпродажную подготовку (дизайн, оформление, подготовка сопроводительной документации и пр.)

Вариант	K34	Вариант	K34
A1	5	C1	10
A2	15	C2	15
B	10	D	5

В среднем фактор конкуренции будет иметь значение:

Вариант	K3	Вариант	K3
A1	3.7	C1	10.5
A2	15	C2	15.0
B	10	D	8.2

4. Инвестиции в НИР и ОКР.

Варианты B и C1 не требуют инвестиций, программы берутся в чистом виде и размножаются.

Варианты A2 и C2 требуют незначительных затрат на редактирование и подготовку материалов.

Вариант D требует определенных усилий на разработку базы данных.

Наибольших работ требует вариант A, вносящий изменения в технологический цикл изготовления компьютера, но все же это не строительство нового цеха.

Вариант	K4	Вариант	K4
A1	4	C1	15
A2	15	C2	15
B	15	D	12

5. Стадия разработки.

Во всех случаях речь идет о полностью готовой разработке и можно принять $K5=15$ для всех вариантов, кроме варианта A1, где надо все-таки выполнить привязку к технологическому процессу. Для этого варианта примем $K1=7$.

6. Конкурентные преимущества. Оценка субъективная, т.к. зависит от положения фирмы на ее рынке. Можно только предположить, что на рынке производителей компьютеров конкуренция более жесткая, чем на смежных и поэтому вариант A1 дает заводу определенные преимущества. Пользующаяся ограниченным спросом и выступавшая вне конкуренции программа B не даст преимуществ, как впрочем и D. Оригинальность открытых идей в разработках A2 и C2 конечно даст преимущества информационной фирме.

С другой стороны, распространение программы B позволяет фирме предлагать в пакете с ней еще и собственную разработку, посвященную смежным вопросам или по крайней мере рассчитывать на то, что ее начнут спрашивать, поэтому вариант B здесь ненулевой.

Вариант	K6	Вариант	K6
A1	7	C1	3
A2	15	C2	15
B	2	D	0

7. Рентабельность.

A1 - рентабельность средняя, ведь речь идет о доработке достаточно дорогой модели компьютера, на которую и так уже падает немало затрат.

У прочих вариантов рентабельность высокая, причем в вариантах В и С1 она немного ниже, чем в вариантах А2, С2, D, т.к. есть затраты на носитель информации (кассету).

Вариант	K7	Вариант	K7
A1	7	C1	12
A2	15	C2	15
B	12	D	15

8. Инновационная значимость.

Некоторые перспективы, что разработка вызовет появление новых есть у интерфейса A1 и у обучающей программы C1. В первом случае возможны разработки специальных программных комплексов для работы в локальной сети, во втором случае возможно развитие идеи в смежные сферы.

То, что появятся новые статьи после публикации A2 и C2 - практически несомненно и здесь этот уровень выше.

Узкая специализация В и D не позволяют надеяться на быстрое развитие направлений.

Вариант	K8	Вариант	K8
A1	3	C1	3
A2	10	C2	10
B	0	D	0

9. Прочие факторы.

9. 1. Исключительность. Везде речь идет об исключительном праве на реализацию идеи. Примем всюду $K91=15$

9.2. Авансовые платежи. Во всех случаях автор идеи желает получить всю причитающуюся ему сумму сразу, до начала эксплуатации разработки, это снижает его роялти. $K92 = 0$.

9. 3. Гарантийные обязательства.

Здесь в них нет нужды, поскольку автор уже получил всю рассчитанную сумму. $K93 = 0$. Среднее значение $K9=5$.

Подсчитаем по среднему значению всех факторов размер средней ставки роялти для всех вариантов.

Вариант	R	Вариант	R
A1	4.7%	C1	7.4%
A2	11.1%	C2	12.1%
B	7.0%	D	8.3%

Конечно, результат достаточно условен, но дает Вам хорошую ориентировку перед началом переговоров с фирмой.

Он же позволяет Вам подумать над тем, насколько правильно Вы предполагаете начать распространение своего продукта. Может быть его надо предлагать совсем в другой форме, а не так, как Вы планировали.

Окончательные результаты можно оценить, зная предполагаемый объем продаж и цену единицы продукции. Рассмотрим примеры.

ВСЕ КОЛИЧЕСТВЕННЫЕ ДАННЫЕ, ПРЕДСТАВЛЕННЫЕ В ПРИМЕРАХ - УСЛОВНЫЕ И ИМЕЮТ ТОЛЬКО УЧЕБНО-МЕТОДИЧЕСКОЕ ЗНАЧЕНИЕ

ИХ СОВПАДЕНИЕ С РЕАЛЬНОЙ СИТУАЦИЕЙ ДОЛЖНО РАССМАТРИВАТЬСЯ КАК ЧИСТО СЛУЧАЙНОЕ

Вариант А

Если завод, с которым Вы ведете переговоры, в состоянии поднять цену на свои компьютеры скажем рублей на 50, введя Вашу доработку, и выпускает в год 10000 изделий, то паушальная сумма равна:

$$S = 10000 * 50 * 0.047 = 22350 \text{ руб.}$$

Если информационная фирма, которой Вы предложите эту разработку имеет 10000 клиентов и в среднем рассчитывает, что 10% из них заинтересуются предлагаемой доработкой по цене 20 руб., то она сможет вам предложить:

$$S = 1000 * 20 * 0.111 = 2220 \text{ руб.}$$

Вариант В

Маркетинговые исследования показывают, что такая программа может иметь спрос в ограниченном регионе, в котором фирма имеет всего 200 клиентов. Предполагается, что 10% смогут приобретать ее по цене 30 руб. за экземпляр.

$$S = 20 * 30 * 0.07 = 42 \text{ руб.}$$

Не очень много, но ведь программа разрабатывалась не для коммерческих целей.

Вариант С

Как показали исследования, можно рассчитывать, что из 10 000 клиентов эту программу готовы приобрести 10% при цене программы 7 рублей.

$$S = 1000 * 7 * 0.074 = 520 \text{ руб.}$$

Рассмотрим встречный вариант, предложенный фирмой - включение ее в сборник. В этом случае фирма удовлетворит все 10 000 клиентов и в среднем каждому программа обойдется в 50 копеек.

$$S = 10000 * 0.5 * 0.121 = 605 \text{ руб.}$$

Как видите, вариант, предложенный фирмой, интереснее и менее хлопотный.

Вариант D

Возможность приобретения такой программной разработки ограничена. База данных по лекарственным растениям безусловно интересна тысячам граждан, но они не имеют дома IBM-совместимых машин. Организаций, работающих в этом направлении, и оснащенных требуемой техникой - немного, и для того, чтобы выйти на них, требуются серьезные усилия по рекламе, что конечно поднимет цену готового продукта.

Предположим, что в покрытие этих затрат установлена цена в размере 4 000 рублей и рекламная кампания привела к заключению 20 договоров (очень оптимистическая оценка).

$$S = 4000 * 20 * 0.083 = 6640 \text{ руб.}$$

Конечно, это недостаточная оценка полувекового труда авторов.

Причина скорее всего в том, что рынок еще не созрел для приобретения этой разработки в той форме, в которой она предлагается (в виде базы данных). И концепция маркетинга с самого начала была порочной.

Имеет смысл подумать о заключении договора с крупным издательством о публикации книги массовым тиражом.

При цене книги 20 руб. можно рассчитывать, что она разойдется в количестве 50000 экземпляров.

$$S = 5000 * 20 * 0.063 = 83\,000 \text{ руб.}$$

Данная оценка ближе к истине. Правда, в этом случае стоило бы пересчитать ставку роялти. По всей видимости она будет несколько иной.

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД

Мы продолжаем печатать заметок известного британского программиста Стива Тернера и сегодня мы поговорим о самом щепетильном вопросе, связанном с программированием - об ошибках в программах. (Начало см. на стр. 49, 70, 117, 143)

Многие из Вас, возможно, видели отрывки из американского фильма "Звездные войны". Наиболее захватывающие сцены космических сражений в этом фильме моделировались с помощью компьютеров и, надо сказать, что суммарные трудозатраты программистов на подготовку всего обширного комплекса программного обеспечения составили 3 тысячи человеко-лет.

Спрашивается - как же можно разрабатывать такие грандиозные проекты и создавать такие обширные системы без ошибок, способных свести на нет огромные трудозатраты.

Да, конечно, судьба планеты не зависит от правильности работы Вашей программы, но в принципе те же проблемы стоят и перед Вами. Почему в программах всегда есть "жучки"? Почему программисты так много ошибаются? Можно ли избежать ошибок или, по крайней мере, постараться удалить их из готового продукта?

Ошибки делают все. Одна медсестра сказала, что если бы она делала столько ошибок, сколько делают программисты, никто из ее пациентов не выжил бы.

Я думаю, что она на самом деле тоже делает немало ошибок. Все дело в том, что они незаметны и не являются очень критическими, а к тем действиям, которые могут быть критическими, ее ведь просто не допускают - их делают дипломированные врачи. В нашей повседневной жизни мы совершаем массу действий, не являющихся критическими и поэтому не замечаем ошибок. И действительно, в любом деле между полным успехом и полным провалом есть еще огромный спектр возможных результатов, включающий в себя все виды нашей естественной человеческой неаккуратности.

Если Вы во время ходьбы споткнетесь, то Ваш мозг автоматически даст команду на корректировку - он ведет постоянный контроль за положением Вашего тела. И только когда Вы ведете работу, не осуществляя такого постоянного контроля - тогда и начинаются все неприятности. Хотя даже и в этом случае рядом может оказаться кто-то другой, кто заметит ошибку и исправит Вас.

Совсем по-другому обстоят дела в программировании, ведь компьютеру все равно правильную или неправильную команду он встречает. Он все равно старается сделать все, что может, чтобы ее исполнить и не заботится о последствиях. Мы по-человечески закладываем в программы массу естественных ошибок, а компьютер слепо все это обрабатывает.

Как правило, ошибки в программировании имеют отдаленные последствия. Время между совершением ошибки и ее проявлением измеряется часами, днями и даже годами, но суть та же. Все мы делаем ошибки постоянно. Разница в том, что ошибки программиста имеют отложенное действие, как бомба с часовым механизмом.

Можно ли перестать делать ошибки? Нет, быть совершенством невозможно. Единственное, что можно сделать - постараться свести их количество к минимуму.

Путь к уменьшению количества ошибок лежит через дисциплину программирования и самоконтроль, а дисциплина и свободное творчество находятся в постоянном конфликте.

Вопрос исключения ошибок, таким образом, сводится к самоконтролю, а самый лучший способ - это расставить ловушки на всем пути. Причем если Вы думаете, что надо исключить ошибки только в программировании, то заблуждаетесь. Их надо исключить еще раньше - уже на этапе проектирования программы. Согласитесь, что нет ничего обиднее, чем узнать, что вся Ваша концепция порочна и неверна. Конечно, мы не можем охватить все случаи, в которых могут возникнуть ошибки, но на наиболее ответственные этапы мы все-таки укажем.

Ошибки в проекте.

1. Размер программы.

Подготавливая проект, оцените размеры всех входящих в него данных: файлов, таблиц, буферов и т.п. Оценить размер программного блока - сложнее, но надо опираться на собственный предыдущий опыт и постоянно контролировать процесс. Попробуйте составить список подпрограмм, которые войдут в Вашу программу. Обязательно оцените средний размер памяти, отпущенный Вам на среднюю подпрограмму.

Когда будете программировать, то постоянно сверяйтесь - укладываетесь ли Вы в этот размер.

2. Быстродействие программы.

Часто скорость является критическим фактором, особенно это относится к играм, имеющим движущиеся объекты. Оцените время, необходимое для выполнения наиболее критических с этой точки зрения операций, например операции по перестроению изображения на экране. Прикиньте, сколько кадров в секунду Вы можете достичь, проверьте себя на простейшем тесте.

Если что-то не получается - подумайте о том, чтобы использовать другие команды процессора или другие методы адресации. Если и это не помогает - может быть можно уменьшить экранную область? Вы же ведь видели не раз, как в программах например используется для динамической графики только верхняя треть экрана, а остальное поле занято статическим, не изменяющимся рисунком.

3. Концепция программы.

Проверяйте концепцию программы на простом примере на БЕЙСИКе, можно это делать и вообще на листе бумаги. Попробуйте объяснить суть задуманной Вами программы кому-нибудь и подумайте над высказанными Вам соображениями. Представьте себя на месте будущего пользователя Вашей программы и задайте себе вопрос: "А что бы я сделал, если бы захотел сыграть в эту игру не по правилам, заложенным программистом?".

4. Цели программы.

Очень часто (особенно если это не игровая, а деловая или прикладная программа) проработка концепции еще не ведет к успеху. Надо абсолютно точно знать, чего же Вы хотите достичь. Прежде, чем начинать проработку проекта, надо составить список целей, для которых создается программа, и, по мере развития проекта, постоянно с этим списком сверяться.

5. Интерфейс "человек-компьютер".

Если Вы предполагаете использовать в программе новый для Вас метод управления, его необходимо проверить на модели. А не слишком ли он сложен? Сможет ли его освоить и применять с удовольствием средний пользователь?

6. Структура программы.

Проверьте всю структуру программы, пройдясь по ней с карандашом в руках вдоль и поперек. Расскажите ее кому-нибудь из знакомых. Объясните, как все работает и что за чем следует. Обязательно постарайтесь получить вопрос типа: "А что будет, если...?", - и ответьте на них. Проиграйте всю программу в уме, следя за блоками структурной диаграммы. Постарайтесь в этот момент стать глупой машиной.

Наметьте для себя точки в программе, в которых устанавливаются основные переменные и в которых они изменяются.

Ошибки в программировании.

Я выписал 10 наиболее часто встречающихся ошибок в программах, написанных в машинном коде. Они расписаны по убыванию вероятности их появления. Есть прямой смысл иметь этот список перед глазами. Если что-то у Вас идет не так, 90%, что Ваша ошибка есть в этом списке.

1. Переменная или регистр процессора не установлены так, как это надо перед

началом операции.

2. Перепутаны условия перехода и т.п., например, написано JR C вместо JR NC.

3. В операциях с регистрами процессора перепутаны операнды, например: LD A,B вместо LD B,A

4. Перепутаны данные и адреса. Например LD HL,1000 вместо LD HL,(1000)

5. Неверно обрабатывается счетчик цикла. Всякий счетчик должен быть:

- установлен;

- вызван;

- изменен;

- сохранен.

6. Не рассмотрен "нулевой" вариант. Не учтено, что будет, если переменная или счетчик будут равны нулю. А если пользователь по запросу введет нуль?

7. Регистр процессора или переменная используются для какой-то второй цели, а их содержимое не сохранено. Наиболее часто это происходит, когда подпрограмма вызывает другие подпрограммы и т. д. или когда Вы вставляете внутрь своей программы новые строки.

8. Арифметические ошибки. Особенно часто возникают, когда в операции участвуют 16-битные и 8 битные числа.

9. Взаимодействие между процедурами. Должно быть строго определено что и откуда процедура принимает и что и куда она возвращает.

10. Перепутаны шестнадцатеричные и десятичные числа.

Пути сокращения количества ошибок.

1. Сначала проектирование - программирование потом.

2. Структурируйте программу.

3. Документируйте программу (процедуры и переменные).

4. Старайтесь писать как можно меньше машинного кода. Максимально используйте свои старые процедуры, копируйте похожие куски программы.

5. Пишите в одной и той же манере. Например, организуйте циклы одним и тем же способом и изменяйте счетчик цикла в одном и том же месте.

6. Будьте проще. Пишите простой код. Не пользуйтесь без особой нужды головоломными приемами.

7. Без нужды не трогайте стек.

8. Новые процедуры сначала расписывайте на бумаге.

Одним из неудобств является то, что на экране Вы можете одновременно видеть лишь очень малый кусок программы. Я считаю, что программисту, работавшему над крупным проектом, принтер необходим. После написания новой процедуры я ее распечатаю и делаю ее "прогон" в уме. Конечно, все возможные варианты ее работы отследить невозможно, но по крайней мере первый прогон по ней и последний я анализирую обязательно.

В вычислительных операциях я прикидываю минимальное и максимальное возможные значения.

В операциях сдвига и в операциях с битами приходится на бумаге рисовать карту битов в байте. Конечно это я делаю не всегда, а когда компьютер не желает исполнять свеженаписанную процедуру.

Как-то раз я написал процедуру, которая сразу заработала. Я был ошеломлен настолько, что не поверил сам себе. Два часа я разбирал ее на части в уверенности, что так не бывает. И нашел в ней такое ... - самую настоящую мину с часовым механизмом, которая только ждала своего часа.

Для тех, кто пишет на БЕЙСИКе.

Как найти ошибку в БЕЙСИК-программе? Что делать, если программа отказывается работать.

1. Начните искать ошибку в строках, которые Вы изменяли последними. Обычно

ошибка в них.

2. Постарайтесь локализовать место ошибки путем запуска не всей программы, а только ее части. Ее начало определит команда `RUN` и или `GO TO`, а конец можете обеспечить вставкой `STOP`.

Начинайте это делать с достаточно большого блока и уменьшайте его постепенно.

3. Если в программе есть или создаются массивы данных, то можете написать маленькую программку для проверки этих массивов. То ли в них содержится, что должно быть.

4. Оговорите суть проблемы со своим другом. Обычно, когда Вы умеете хорошо объяснить суть проблемы, то и решение уже рядом.

5. Попробуйте вжиться в проблему настолько, чтобы засыпать с нею в уме. Нередко утром Вас осеняет простое решение. Конечно это стоит делать только для особо важных и сложных проблем.

6. Распечатайте листинг проблемной области. Прокрутите работу этого блока всухую (в уме). Наметьте точки, в которых Вы вставите `STOP` (точки прерывания). Когда программа встанет в точке прерывания, прямой командой `PRINT` проверьте значения важнейших переменных.

А что делать, если программа не Ваша, если Вы набрали ее из распечатки в журнале, и она не работает?

Надо сразу сказать, что проблема эта довольно типичная. Большинство ошибок возникает при наборе, многие вызваны непропечаткой символов, а есть и такие, которые заложены уже в приведенный листинг. Для всех категорий ошибок есть масса причин, вызывающих их появление, и это конечно проблема, но решение ее - довольно полезная задача. Все программисты знают, что настоящий профессионализм приобретается не тогда, когда находишь собственные ошибки, а тогда, когда чужие.

1. Запускайте программу по частям, используя оператор `STOP` и проверяя переменные после остановки. Если все в порядке, запускайте программу дальше оператором `CONT`.

2. Найдя строки, в которых происходит ошибка, проверьте правильность своего набора. Обратите внимание на соответствие прописных и строчных букв, на похожесть буквы `I` и цифры `1`, а также буквы `B` и цифры `8`. Постарайтесь понять, что делает каждый оператор и проследить из каких других частей программы могла прийти ошибка вместе с какой-либо переменной. Если Вы поймете, что делают строки программы, то сможете обнаружить даже и те ошибки, которые содержатся в приведенном листинге.

3. Особое внимание обратите на ошибки, связанные с `DATA` и с `POKE` - их наиболее трудно отыскивать, часто здесь приходится не просто проверять правильно ли Вы набрали текст программы, но и разбираться с тем, что и для чего они вводят.

4. Внутри циклов используйте оператор `PRINT`, чтобы следить за ходом изменения параметра цикла и основных переменных.

5. Ищите улики. Пробуйте разные значения в `INPUT` и следите за тем, какие подпрограммы их обрабатывают. Старайтесь точно определить условия, в которых они начинают неправильно работать.

Пробный запуск.

Несмотря ни на какие предпринимаемые меры предосторожности, некоторые ошибки тем не менее все же доживают до тестового запуска процедуры. Тестирование должно быть систематический. Проводите его малыми порциями. Когда компьютер "зависнет", Вы будете знать, где собака зарыта.

Постепенно наращивайте размер программы путем "прикрепления" новых блоков к уже проверенным и надежно работающим. Если какой то блок не зависит от работы прочих - испытывайте его в первую очередь и только потом проверяйте его в составе всей программы.

Точно так же я работаю и на БЕЙСИКе, только там я запускаю группы строк, проверяя содержимое переменных.

Пользовательская проверка.

Этот этап служит не столько для отыскания ошибок, сколько для проверки соответствия ее конечным целям. Кроме того, очень часто другие пользователи делают такие вводы в программы, которых программист и не предполагал и не предвидел. И это позволяет вскрыть еще немало ошибок.

На этом этапе я провожу еще и окончательную настройку программы, чтобы достичь наилучшей "играбельности".

Последний барьер.

Когда моя игра готова к выпуску, я провожу еще одну - последнюю проверку.

Я читаю всю программу в распечатке, процедуру за процедурой. Это чтение на понимание. Я должен убедить себя в том, что понимаю, что делает каждая команда. При этом обычно вскрывается и еще какая-нибудь шальная ошибка. Далее уже слово за публикой.

План тестирования.

Чтобы правильно выполнить тестирование написанных процедур, нужна систематичность. Лучше всего для начала подготовить план.

Каждому тесту дается свой справочный номер, рядом с которым выписываемся список данных, подлежащих проверке. Очень часто эти данные - это то, что вводит пользователь при работе с программой, например команды, поступающие от джойстика.

Можно быть еще более систематичным и расписать возможные условия, в которых будет работать та или иная процедура. При этом можно использовать распечатку программы, пометив из ней все группы команд, находящиеся между условными конструкциями типа JR Z, DJNZ, JP C и т.п.

Чтобы тест был всеобъемлющим, определите условия, при которых те или иные блоки должны работать. Когда же дело дойдет до реального теста, большинство возможных тестов Вы вычеркнете из своего листинга без необходимости специально задавать условия и прогонять блоки. Останутся самые необходимые пункты, которые нельзя упустить.

Убедитесь, что каждый из намеченных к тестированию блоков реально выполняет свои функции. Это не значит, что достаточно посидеть и посмотреть, как он работает. Решите заранее для себя, в каких точках Вы будете прерывать работу процедуры и проверять то ли содержится в регистрах процессора, что там должно быть и те ли значения имеют переменные, какие положено. Конечно, не каждую группу команд стоит так проверять. Это можно оставить для наиболее головолomных. А вот конец каждой процедуры - это вполне удобное место, в котором стоит это сделать.

В следующем выпуске я приведу образец плана тестирования программы, а также приложу распечатку собственной программы MONITOR (Real Time Monitor - монитор, работающий в режиме реального времени), с помощью которой очень удобно делать многочисленные проверки.

(Продолжение следует)

Слово эксперту

С большой радостью мы отмечаем, что не успели мы объявить атаку на программу EQUINOX, как получили ответ от эксперта, сделавшего на наш взгляд великолепную проработку.

Так держать, эксперты!

EQUINOX
(РАВНОДЕНСТВИЕ)
Mikro-Gen 1986г.



Эксперт Кочнев С. В.
г. Москва.

Программа, строго говоря, относится к жанру аркадных адвентюр, но учитывая необходимость постоянно отбиваться от противника, справедливо будет определить ее жанр как ARCADE/ADVENTURE/ACTION. Недюжинная реакция и умение метко стрелять здесь понадобятся не менее, чем способность мыслить логически, поэтому эту игру можно рекомендовать и любителям боевиков.



Программа оснащена очень высококачественной графикой и, хотя общее число локаций всего 128 экранов, не считая комнат межуровневой телепортации, что по сравнению с некоторыми программами не так уж велико, отличная графика с лихвой

компенсирует этот недостаток. В программе задействовано немало число технических устройств. Одни работают бесплатно, другие - только после уплаты соответствующей суммы. Такое разнообразие делает программу одинаково привлекательной для любителей разных жанров.

Действие происходит в "далекой-далекой" Галактике, на астероиде, ставшем лунной обитаемой планеты. На спутнике велась добыча жизненно важной для обитателей этой планеты руды, но внезапно, в день равноденствия, наступающего здесь раз в несколько тысяч лет, в недрах астероида проснулось страшное животное, занесенное вместе с астероидом из соседней звездной системы.

Чудовище выпустило на поверхность разработок свои органы чувств и оказалось способным силой своего воображения создавать фантомов, способных убивать. Оставшиеся в живых рабочие бежали, бросив шахты. Вооруженные Силы планеты в состоянии уничтожить чудовище, применив ультразвуковое оружие, но бежавшие рабочие оставили в спешке канистры с ядерным топливом, применявшимся в силовых агрегатах машин, небезопасными. Поэтому применение оружия грозит взрывом всему астероиду.

Положение осложняется еще и тем, что ядерные канистры сами по себе неустойчивы и взорвутся, если их вовремя не поместить в специальную камеру.

Ваша миссия: ликвидировать опасность взрыва. Управляя дроидом, Вы должны найти на каждом уровне шахты ядерную канистру и в установленное время по пневмолиниям отправить ее в специальную камеру хранения.

Настройка программы.

Сразу после загрузки программы Вы получаете главное меню:

1. DEFINE KEYS - задание клавиш.
2. INSTRUCTION - инструкция.
3. START GAME - запуск игры.
4. JOYSTICKS - джойстики.

Если Вы будете работать на клавиатуре, то для выбора удобных для Вас клавиш вызовите опцию главного меню "DEFINE KEYS".

Получите следующее меню:

SELECT KEYS FOR:

(выберите клавиши для:)

LEFT - движение влево

RIGHT - движение вправо

THRUST - включение двигателя

USE - пользование предметами

FIRE – огонь

PAUSE - пауза

Если Вы используете джойстик, то вызвав опцию "JOYSTICK", попадете в меню выбора органа управления:

1. KEYBOARD - клавиатура
2. KEMPSTON - кемпстон-джойстик
3. INTERFACE 2 - интерфейс-2, т.е. синклер-джойстик (правый).

Функции "USE" на джойстике соответствует нажатие рукоятки "вниз".

Следует иметь в виду, что если поиграв на джойстике, Вы решили вернуться на клавиатуру, то клавиши Вам придется задавать заново. Если же Вы сразу решили использовать клавиатуру, то выбирать ее опцией "JOYSTICK" не надо.

Знающие английский могут вызвать на экран краткую инструкцию к программе. Для возврата к главному меню нажмите любую клавишу.

Подождите пока запускать игру, лучше послушайте музыку (если не хотите, нажмите любую клавишу, кроме управляющих главным меню). По ее окончании программа

последовательно выдает на экран внешний вид и краткое описание основных видов технических устройств, с которыми Вам предстоит столкнуться. Здесь мы даем инструкции к этой технике в порядке вывода их компьютером на экран.

TELEPORTER TERMINAL

Система местной (т.е. в пределах данного уровня) телепортации, состоящая из пары устройств, бесплатно работать не будет. Поэтому сначала следует отыскать и опустить в телепортер монету, для чего следует поместить дроида внутрь устройства и нажать клавишу "USE". Звуковой сигнал свидетельствует о приеме монеты.

Для телепортации необходимо совместить дроида с шаром, для чего следует включить двигатель. Вы окажитесь у второго телепортера. За одну монету Вы можете дважды поменять свое местоположение (локацию). Однако имеется одна тонкость: системе безразлично, будете ли Вы телепортироваться "туда-обратно" или два раза "туда" (есть и такие хитрые лабиринты). В любом случае за одну монету она обслужит Вас дважды. Вы можете опустить в один из терминалов и большее число монет. При этом соответственно возрастет и количество возможных телепортаций. Во время местной телепортации Вы можете брать с собой любые предметы.

Примечание: не следует пытаться опускать в терминал батарейки, ключи, динамитные шашки и прочие несвойственные монетоприемнику предметы - устройство крепкое, выдерживало и не такое.

TRANS-LEVEL TELEPORTER

Устройство межуровневой телепортации. Вход только при наличии специальной магнитной карточки. В комнате телепортера слева вверху находится индикатор уровня шахты, на котором Вы в данный момент находитесь. В центре, на стене, расположены клавиши с цифрами, обозначающими номера уровней. Для телепортации на нужный Вам уровень необходимо подлететь к соответствующей клавише и выполнить функцию "USE". Раздастся звуковой сигнал и индикатор укажет, на каком уровне Вы теперь находитесь. Можно выходить.

Следует, однако, иметь в виду, что если номер выбранного Вами уровня окажется больше числа, написанного на карточке, то устройство работать откажется. Таким образом, если Вы на первом уровне войдете внутрь с карточкой N1, то телепортироваться вообще никуда не сможете.

Т.к. Вы можете иметь при себе только один предмет, а для входа в TRANS LEVEL TELEPORTER необходимо иметь при себе магнитную карточку, то ничего, кроме нее перенести на другой уровень Вы не сможете.

Топливо во время перемещения в комнате телепортера не расходуется, время не останавливается. Лазер во избежание несчастного случая или умышленного вредительства блокируется.

CREDIT DISPENSER

Дословно - распорядитель кредитов.

К делу не относится, но любопытно: в игровой технике (мы не имеем в виду "Морской бой" и прочие шедевры, сделанные видимо специально, чтобы не возбудить в советском человеке жажду крови) количество опущенных в автомат монет индицируется как "CREDIT". Есть, правда, в зависимости от достоинства монеты и другие соотношения, скажем 1 COIN - 1/2 CREDIT и другие. В любом случае, CREDIT - это число игр, оплаченных заранее. У нас это вполне прижилось и ожидающиеся своей очереди "профи", скажем у имитатора мотогонок "SUPER HANG-ON" (кстати его автор - фирма "SEGA" отлично конвертировала для SPECTRUMа самую длинную и сложную из трасс - PART AMERICA) спросит у играющего: "Сколько кредитов поставил?" В нашем случае, будь у системы TELEPORTER TERMINAL дисплей, то опустив в ее монетоприемник пару монет, мы бы увидели сообщение: "CREDIT:4", что соответствует четырем телепортациям.

Не верьте названию этого странного устройства! Просто так "раздавать кредиты" он не будет. CREDIT DISPENSER меняет монету на жизнь вашего дроида! Вообще, на каждом уровне находится только минимально необходимое число монет для выполнения данной миссии. Поэтому если Вы совершите ошибку, Вам будет просто нечем платить системе телепортации. В этом случае ничего другого не остается, кроме как обменять одного из дроидов на монету. Для этого надо опустить дроида на горловину аппарата (при этом у Вас не должно быть с собой никаких предметов) и нажать клавишу "USE". Если это была последняя попытка, игра прекратится. Старайтесь не делать ошибок в ходе игры, тогда и пользоваться CREDIT DISPENSER'ом не придется. Кроме того, он имеется не на всех уровнях. Если Вы попадете в такую ситуацию, Вам останется только выйти из игры, нажав "CAPS SHIFT" + "BREAK".

DISPOSAL CHUTE

Распределительное устройство вакуумной системы сообщения, пронизывает все уровни шахты. В нашем случае может служить для отправки ядерных канистр в хранилище. Работает бесплатно. Чтобы воспользоваться DISPOSAL CHUTE необходимо ввести Вашего дроида (разумеется с канистрой на борту) в приемник устройства и выполнить функцию "USE". Вы увидите прибытие контейнера в камеру хранения. Вход в приемник возможен как слева, так и справа, поэтому если труба пневмолинии преградила Вам путь, пересекайте ее в месте приемного устройства.

MAGNETIC LIFT

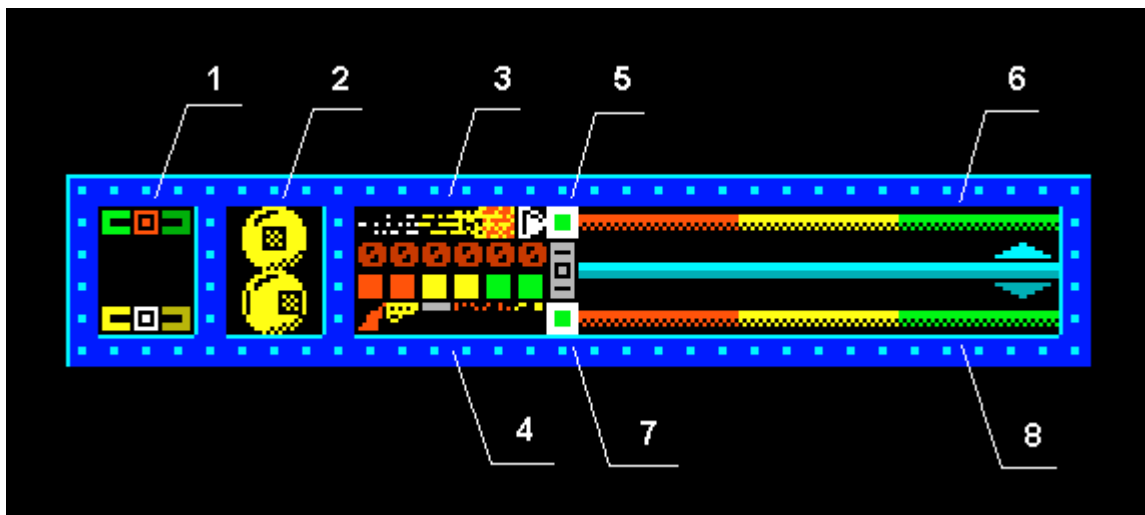
Остроумно придуманная система подъема металлических тел, состоящая из постоянного магнита и металлического контура, образующего шахту лифта и предназначенного для поддержания силовых линий магнитного поля вдали от его источника, а также экранизации магнита (без этого он бы действовал на все металлические тела, в том числе и на не находящиеся в шахте лифта). В контуре на каждом из этажей, которые пересекают лифт, сделаны разрезы для входа и выхода. Силовые линии, пронизывающие контур, настолько мощны, что в этих местах наблюдается непрерывный электрический разряд. Магнитный лифт начинает свое действие сразу после входа дроида внутрь шахты. Вам остается только не прозевать нужный этаж. Кроме несомненных положительных свойств - экономия топлива и времени, у лифта есть и очень серьезный недостаток: работать он может, разумеется, только в одну сторону. Это не имеет значения на первом уровне, но на остальных уровнях лабиринт построен так, что если Вы по ошибке неправильно им воспользуетесь, например забудете захватить нужный предмет, то за обратный путь Вы можете заплатить не только временем, топливом и энергией лазера, но и деньгами. Отметим также, что пересечь шахту лифта в горизонтальном направлении, кроме ее верхней точки, возможно только в одном направлении, так как подъем начинается сразу после входа в шахту и поэтому входы в MAGNETIC LIFT расположены на разной высоте. Иначе говоря, магнитный лифт обладает односторонней проходимостью как в вертикальном, так и в горизонтальном направлениях.

Если эти объяснения показались Вам слишком запутанными, посмотрите на данный ниже план второго уровня, и Вы сразу все поймете.

Работу технических устройств, а также работу с предметами мы рассмотрим ниже, параллельно с разбором методов прохождения некоторых уровней.

Экран программы.

После запуска программы Вы увидите сверху на экране таблицу, отражающую основную текущую информацию.



где:





- 1 - содержимое грузового отсека;
- 2 - количество оставшихся у Вас дроидов;
- 3 - очки;
- 4 - время;
- 5 - индикатор включения двигателя;
- 6 - указатель уровня топлива;
- 7 - индикатор включения лазера;
- 8 - указатель состояния энергетического отсека.

Стрелки на индикаторах топлива и энергии, по мере их расхода, движутся справа налево, т.е. крайне правое положение соответствует их максимальному запасу.

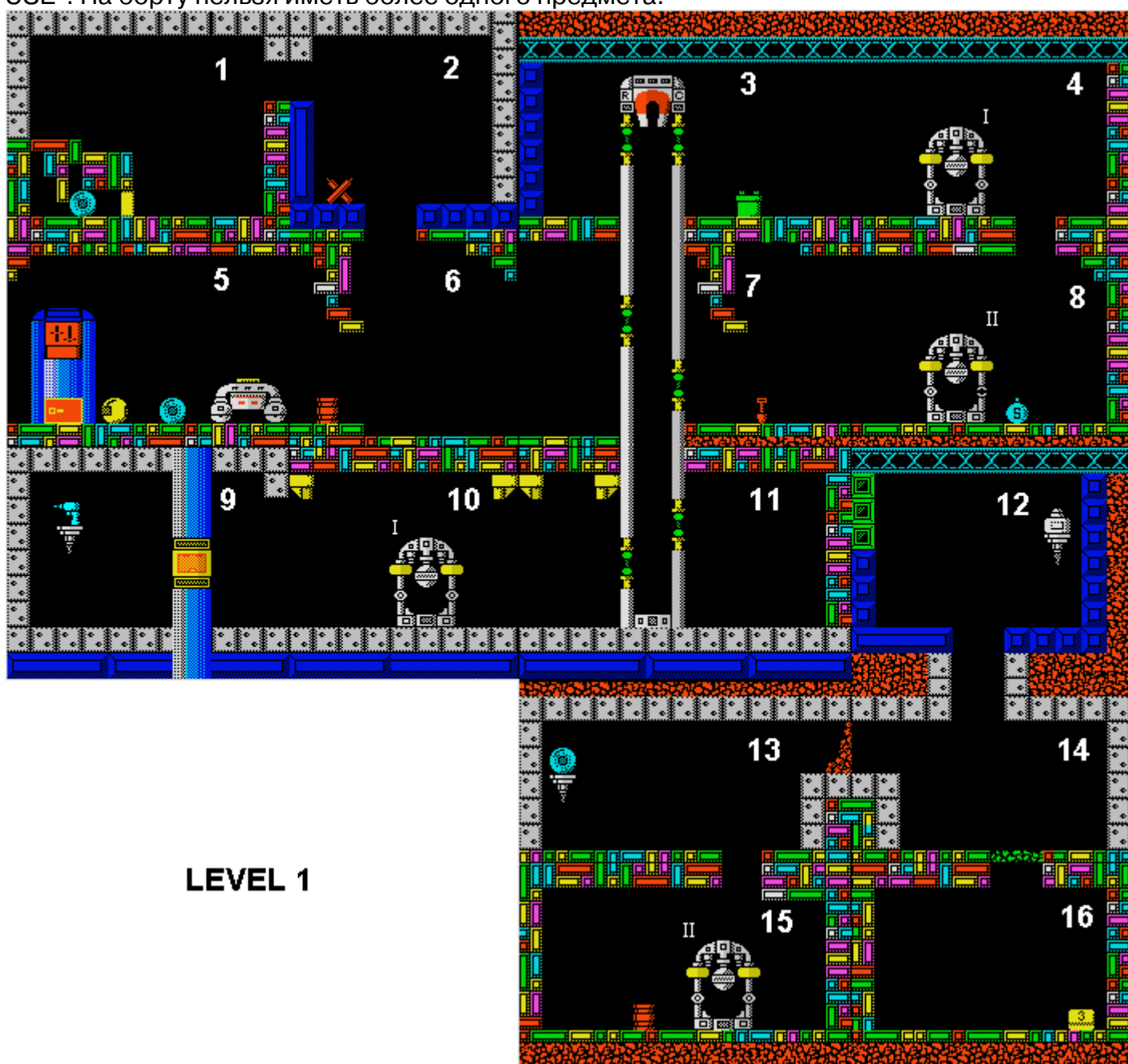
Работа с предметами и описание некоторых уровней.

Сначала рассмотрим основные предметы, с которыми Вам предстоит работать на всех уровнях, а также приведем их условное обозначение, т.е. дадим легенду к нашим картам.

Обозначение	Название	Назначение
	Ядерная канистра (ЯК)	Опустить в приемник DISPOSAL CHUTE
	Магнитная карточка (МК)	Вход в межуровневый телепортер. Номер указывает на максимальный уровень телепортации.
	Монета (М)	Оплата работы TELEPORTER TERMINAL'a
	Батарейка	Подзарядка энергетического отсека
	Канистра с топливом	Заправка двигателя
	Дрель (Д)	Вскрытие сейфов

	Динамитная шашка (ДШ)	Расчистка проходов
	Граната	Уничтожает всех фантомов в комнате
	Ключ (К)	Отпирание дверей
	Предохранитель (ПР)	Отключение систем электронной защиты

Примечание: Для захвата, обмена и использования всех предметов служит клавиша "USE". На борту нельзя иметь более одного предмета.



Мы умышленно не даем план первого уровня, так как он достаточно прост для того, чтобы Вы потренировались на нем сами.

В правом верхнем углу каждой комнаты находится число, соответствующее ее

порядковому номеру. Нумерация, разумеется, чисто условная и служит для упрощения записи плана прохождения уровня. Символы, обозначающие технические устройства, достаточно похожи на их внешний вид, поэтому мы не стали делать для них пояснительную таблицу.

Выйдя из TRANS-LEVEL TELEPORTER'а в комнате 5, возьмите монету, двигайтесь 6-7, далее на лифте -3-4, опустите монету и телепортируйтесь -10-9, возьмите дрель, -10-11-7 (выход налево, не прозевайте, иначе придется возвращаться /3-4-8-7/) -6-2-1, используя дрель, достаньте монету, -2-6-7-3-4-8. Опустите монету, но не телепортируйтесь (кстати, обратите внимание на невозможность 6-7-8, это и есть односторонняя проходимость) 7-6 (а так можно!) -2, возьмите динамитные шашки, -6-7-3-4-8,Т,-15-13, расчистите взрывчаткой завал, возьмите монету, -15, опустите монету, -13-14-12, возьмите канистру, -14-13-15 (Внимательно следящие наверное уже отметили, что на данный момент на системе TELEPORTER TERMINAL (I) можно произвести 1Т, а на системе TELEPORTER TERMINAL (II) – 3Т). Т, -8-7-3-4, Т, -10-9, опустите канистру в DISPOSAL CHUTE. На этом Ваша миссия на втором уровне считается выполненной. На шкале времени появляется сплошная полоса, т.е., здесь опасность взрыва ликвидирована. Но Вам нужно идти дальше, для этого нужна магнитная карточка с номером следующего уровня. -10-11-7, возьмите ключ, -8,Т,-15-13-14. Здесь мы встречаемся с нестандартным применением ключа - в качестве ершика для прочистки горизонтального прохода. -16, возьмите карточку, -14,-13-15,Т,-8. На этом Ваш кредит на обеих системах телепортации иссяк (можете проверить), но они нам больше не нужны. Возвращайтесь к TRANS-LEVEL TELEPORTER'у: -7-6-5 и входите.

Как видите, пройти уровень не так уж и просто, есть над чем поломать голову. Здесь мы не давали указаний по поводу использования батареек и бочек с топливом: их, а также гранату, применяйте по собственному усмотрению.

Уровень 3

Здесь мы встречаемся с тремя новыми устройствами - транспортером (7-8-9), односторонним проходом (1-4) и системой силовой защиты (1). Для снятия последней в ее щит необходимо вставить предохранитель (ПР). Т.к. подробное описание метода прохождения занимает довольно много места, а Вы уже имеете довольно полное представление о работе технических устройств, то в подробных комментариях больше нужды нет. Перейдем к сокращенной записи. Символ ""¹ будет обозначать "взять", "" - "использовать" и "" - "обменять". Отметим теперь касательно транспортера: он является горизонтальным аналогом лифта, ехать на нем можно только в одном направлении, перейти в соседнюю комнату можно только по ходу движения.

Способ прохождения:

12-11-10-7-4 .. К-7-10-11-14-.. К 15 .. Пр 14-11-10-7-4-5-2-1..Пр .. М-4-7-10-11-14-13-16-.. М; Т-3.. ЯК 6-9-8-7-10-11-14-13 .. ЯК-16 Т-3-6 .. МК-9-8-7-10-11-12

Как видите, запись получилась достаточно компактной. Постарайтесь предложить свой алгоритм, хотя бы не намного отличающийся от данного. Сделайте то же для второго уровня.

Уровень 4

Здесь Вам встретятся прессы (комнаты 1 и 8). От Вас им ничего не нужно, постарайтесь только под них не попасть.

Разобрав предыдущие примеры, Вы можете найти алгоритм прохождения этого уровня самостоятельно. Сделайте это и сравните с алгоритмом данным нами. Не спутайтесь, если внешне они будут отличаться. Принцип всегда будет одинаков.

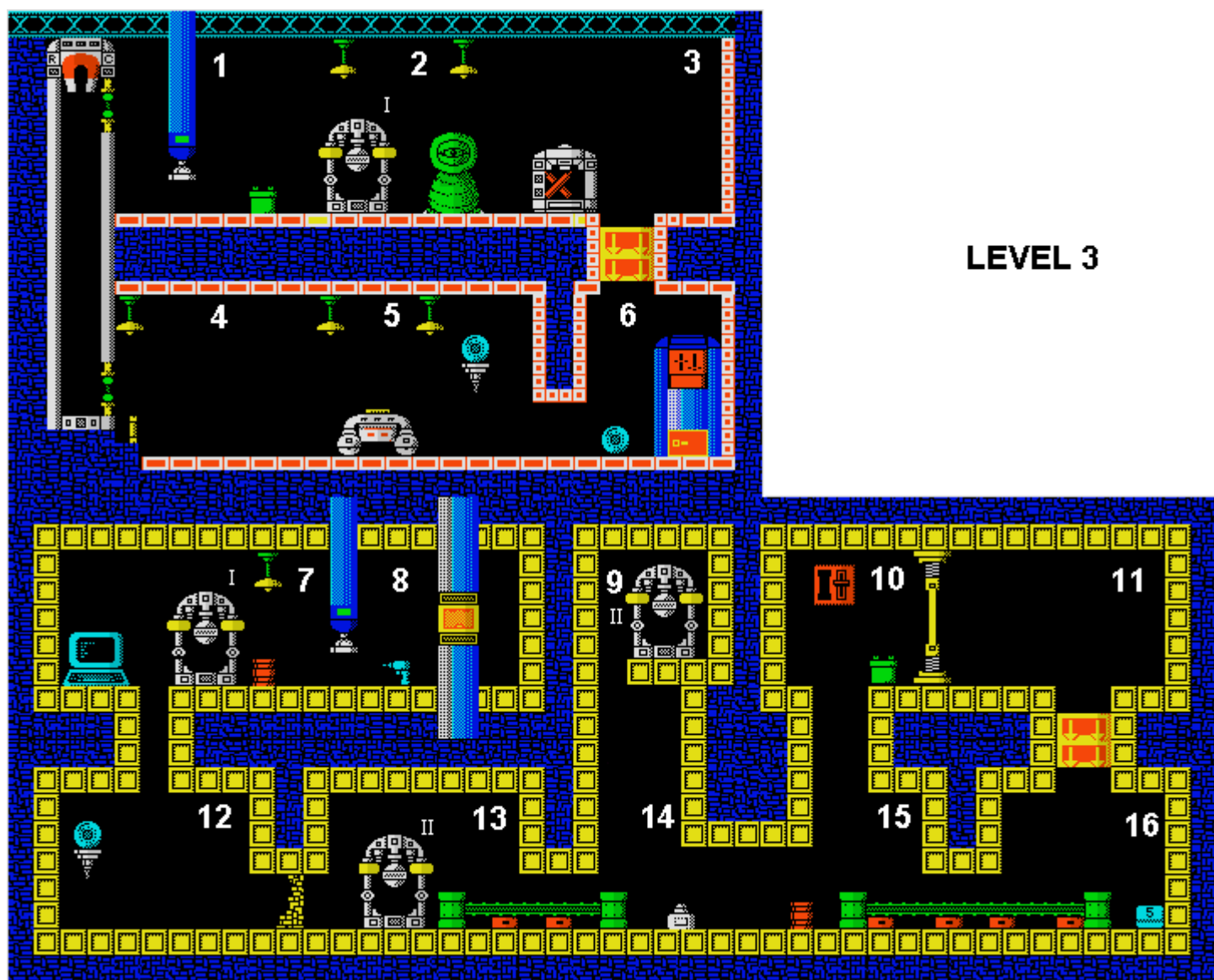
Способ прохождения:

6 ... М-5-4-1-2 ... М-3-6-5 ... М-4-1-2 .. М-3-6-5-1 ... Пр-1-2 Т-7-8 Пр.. Д-7 Т-2-3 ..

¹ В оригинале пробелы. (Прим. NUK)

Не имея желания лишать Вас удовольствия самостоятельно пройти оставшиеся уровни (а половину мы уже разобрали), рассматривать мы их не будем.





Полезные советы

У игры EQUINOX есть очень интересное свойство. Впрямую мы о нем не сообщали, но если Вы сами догадались о сказанном ниже, значит не зря потратили время на изучение этого описания и нестандартное мышление Вам не чуждо. Дело в том, что в этой игре возможен переход на следующий уровень без выполнения предыдущего, ведь для этого нужна только карточка с соответствующим номером.

Это очень облегчает работу, ведь, как Вы уже успели понять, сразу программу не пройти, прежде надо разработать план действий. Мы рекомендуем Вам сначала не выполнять миссии ни на одном из уровней, этим Вы сэкономите свое время, а возможно и попытки. На первоначальном этапе Вы должны составить точные планы всех уровней, здесь Вам поможет клавиша "PAUSE". Следующий этап - чисто аналитический - разработка алгоритма прохождения. Кстати, Вам понадобятся два алгоритма: для простого перехода на следующий уровень и для выполнения миссии. После этого можете возвращаться к игре и спасать астероид.

Топлива и энергии, как правило, хватает, но все же старайтесь их использовать рационально. Например, если Вы видите, что Вам скоро потребуется заправка, на борту у Вас ничего нет, то взятую бочку не надо использовать сразу! Продолжайте движение, клавишу "USE" нажимайте только тогда, когда кончится топливо или потребуются место в грузовом отсеке, ведь содержимого канистры хватает, чтобы полностью заправить топливный отсек, и его избыток будет просто выброшен. Это справедливо и для батареек. Если у Вас на борту что-то есть и необходима дозаправка, то подлетите к канистре, обменяйте ее на Ваш предмет, отлетите от этого места и используйте канистру, после чего заберите предмет.

Выбросить ни один из предметов нельзя, можно либо обменять его на другой, либо использовать. Во избежание тесного контакта с призраками, комнату старайтесь "проскакивать" побыстрее, гранату используйте там, где это затруднительно. И последнее: не пугайтесь, увидев незнакомый предмет или устройство, может быть это только графическое решение?!

GOOD LUCK!

ADVENTURE PROJECT

Расчет характеристик персонажей.

Задание начальных атрибутов Ваших персонажей - это только начало трудного пути установления правил, по которым они будут взаимодействовать друг с другом.

В предыдущей статье (№6, с.122) мы начали рассматривать проект, связанный с похищением экспериментального образца "Часов Вселенной" и остановились там, где Инф, Ора и Ком обнаружили затонувшую яхту, на которой налетчики вывезли уникальный прибор. Инф и Ора ныряют в море и находят, что яхта охраняется двумя чудовищами - Крамом и Боргом. Дело идет к тому, что столкновения не миновать.

Сражение всегда было важнейшим элементом адвентюрных программ, но редко когда удается создать напряженный дух схватки. Помочь приблизить ситуацию к реальной могут те принципы искусственного интеллекта, о которых мы и говорим.

Общий подход в программе состоит из двух этапов:

1. На основе характеристик персонажей рассчитывается желательность схватки. Либо та, либо другая сторона могут в последний момент отказаться от боя.

2. Задаются начальные условия боя:

- в программных переменных устанавливаются имена тех, кто участвует в битве и какими преимуществами они располагают (наличие оружия или магических объектов).

Эти этапы взаимосвязаны. Компьютер рассчитывает желательность схватки, а затем задает ее начальные условия. Например, Инф не станет наносить удара по Оре. Имеет смысл вообще исключить такую возможность. Вы можете, скажем, встроить в программу блок, который будет игнорировать те команды пользователя, которые не согласуются с личными характеристиками персонажей и выдавать на экран какое-то сообщение.

В нашем примере сражение, скорее всего, неминуемо, поскольку проникнуть на яхту герои все-таки должны для успешного выполнения цели игры.

Итак, программа, занимавшаяся имитацией схватки, должна решать следующие 6 задач:

1. Сколько героев вступают в схватку?
2. Сколько противников им противостоят?
3. Соотношения в силе между персонажами.
4. Имеется ли оружие и как оно учитывается.
5. Пути выхода из битвы.
6. Условия победы.

С первыми двумя проблемами из перечисленных мы с Вами разобрались в прошлом выпуске, когда персонажи решали, стоит ли им вступать в бой.

Соотношения боевых характеристик персонажей зависят от их атрибутов, о которых мы тоже говорили в шестом выпуске. Нам необходимы формулы для того, чтобы из исходных атрибутов (эмоции, энергия, сила, интеллект, совесть, образованность) рассчитать производные характеристики, необходимые для битвы, такие, как гнев, мужество, решительность, трусость, самолюбие. Физическая сила может не рассчитываться, а браться из атрибутов.

Гнев, например, может формироваться из эмоций, силы и интеллекта. Мужество из энергии, силы и совести.

Если первичные атрибуты установлены в диапазоне (-10... +10), Инф, Ора, и чудовища могут иметь факторы, приведенные на рис. 1.

Инф не имеет отрицательных параметров и потому очевидно готов к схватке. Оре не так повезло, у нее может быть самолюбие как у львицы, но мужества и силы у нее, как у овечки.

Тем не менее, недостатки Оры могут быть парированы, если мы рассмотрим их противников. Брэг и Крам - это пара хоть и могучих, но трусливых и неповоротливых монстров, мало способных выиграть хоть какое-то сражение.

Программа, приведенная на рис. 2 - это упрощенная процедура, описывающая схватку. Входящие формулы, описывающие такие факторы, как гнев или мужество, должны быть рассчитаны в другой подпрограмме, до битвы. Так гнев, например, может рассчитываться по формуле:

$$\text{LET гнев} = (\text{INT}(\text{эмоции} * \text{интеллект} + \text{сила}) / \text{NOATT}) + 1$$

Формула рассчитывает среднее значение атрибутов, заключенных в скобках. NOATT - количество атрибутов, в нашем случае это 3.

Функция INT служит для округления результата. Тот недостаток, что "Спектрум" всегда при вычислении целой части числа выполняет округление вниз, можно парировать за счет прибавления единицы к полученному результату.

Когда все атрибуты рассчитаны. Вы можете перейти к сценарию битвы. Листинг, приведенный на рис.2, соответствует атрибутам, приведенным на рис.1, которые передаются через массив C, организуемый в строках 1...90.

Группа операторов IF...THEN в строках 110-120 занимается расчетом, имеют ли Инф и Ора достаточно сил для продолжения битвы.

Брог и Крам не имеют такой возможности последнего выбора, но Вы всегда можете изменить программу и поставить столько проверок, сколько хотите.

Переменная N\$ в строках 110 и 120 выполняет роль статусной переменной. Она выставляется в строке 2 и до строки 110 является пустой.

Если кто-то из героев ранен и его сила падает ниже -9, его имя помещается в переменную N\$ и передается в подпрограмму 2000. Здесь чудовища могут его убить, если их атрибуты им это позволяют.

Теперь рассмотрим действия, выполняемые в строках 160...190. Строка 160 сравнивает гнев, силу и мужество Инфа и Крама. Если показатели Инфа лучше, то он выигрывает раунд и оператор LET в конце строки выполняется таким образом, что его гнев возрастает, а сила уменьшаются, далее управление передается строке 180 для оценки второй пары.

Если Крам выигрывает раунд, то управление переходит на строку 170. где изменяются его атрибуты.

Строка 190 изменяет атрибуты Брома, если Ора проигрывает в схватке.

Когда эти расчеты и изменения произведены, программа возвращается на строку 100, откуда весь процесс вновь повторяется.

Подпрограмма в строке 2000 вычисляет, может ли противник убить героя, сила которого упала ниже -9. Если это произошло, то к имени героя, хранящегося в переменной N\$, добавляется символ #.

Атрибут	Инф	Ора	Кран	Брог
Ловкость	5	9	-3	-3
Гнев	3	2	7	7
Самолюбие	5	9	9	6
Интеллект	7	2	4	4
Мужество	7	-4	-2	-3
Сила	4	-3	7	5

Рис. 1

```
1 REM Подпрограмма "СРАЖЕНИЕ"  
2 LET N$ = ""  
5 RESTORE 60  
10 DIM C(4,6)  
20 FOR k = 1 TO 4  
30 FOR m = 1 TO 6  
40 READ C(k, m)  
50 NEXT m: NEXT k  
60 DATA 5,3,5,7,7,4:REM ИНФ  
70 DATA 9,2,9,2,-4,-3:REM ОРА  
80 DATA -3,7,9,4,-2,7:REM КРАМ
```

```

90 DATA -3,7,6,4,-3,5:REM БРОГ
110 IF C(1,6)<=-9 THEN PRINT "Инф не может больше сражаться": LET N$="ИНФ": GO SUB 2000
120 IF C(2,6)<=-9 THEN PRINT "У Оры нет больше сил": LET N$="Ора": GO SUB 2000
125 LET G$=""
130 IF C(3,6)<=-9 OR C(4,6)<=-9 THEN GO SUB 3000
140 IF G$="STOP" THEN STOP
160 IF (C(1,1)+C(1,2)+C(1,6)) > (C(3,1)+C(3,2)+C(3,6)) THEN LET C(1,2)=C(1,2)+1: LET
    C(3,6)=C(3,6)-1: GO TO 200
170 LET C(3,2)=C(3,2)+1: LET C(1,6)=C(3,6)-1
180 IF (C(2,1)+C(2,2)+C(2,6))(C(4,1)+C(4,2)+C(4,6)) THEN LET C(2,2)=C(2,2)+1: LET
    C(4,6)=C(4,6)-1: GO TO 180
190 LET C(4,2)=C(4,2)+1: LET C(2,6)=C(2,6)-1
200 GO TO 100
2000 REM Гибель героя.
2010 IF C(3,1)>4 AND C(3,5)>5 AND C(3,6)>3 THEN PRINT "Крам убивает Инфа": LET N$ = N$+"#"
2020 IF C(4,1)>4 AND C(4,5)>5 AND C(4,6)>3 THEN PRINT "Брог убивает Ору ": LET N$=N$+"#"
2030 IF N$="Инф#" THEN LET C(2,1)=C(2,1)-2: LET C(2,2)=C(2,2)+1: LET C(2,5)=C(2,5)-1
2040 IF N$="Ора#" THEN LET C(1,1)=C(1,1)-2: LET C(1,2)=C(1,2)+1: LET C(1,5)=C(1,5)-1
2050 RETURN
3000 REM Гибель монстра
3010 IF C(3,6)<=-9 THEN PRINT "Крам убит": LET G$="STOP"
3020 IF C(4,6)<=-9 THEN PRINT "Брог убит": LET G$="STOP"
3030 RETURN

```

Строки 2030 и 2040 используют новое значение N\$, чтобы изменить атрибуты оставшегося в живых персонажа. В приведенном примере его гнев увеличивается, а самолюбие убывает, такое взаимовлияние довольно просто, но наглядно демонстрирует возможности системы, в которой один персонаж оказывает влияние на других.

Последняя подпрограмма, начинавшаяся в строке 3000, проверяет, не пора ли погибать монстрам. То же уменьшение сил, которое убивает положительного героя, может убить и отрицательного.

Переменная G\$, введенная в строке 125, участвует в этой процедуре. G\$ принимает значение "STOP", если Брог умирает и после возвращения в главную часть программы заканчивается ее работа в строке 140. Когда монстры уйдут, Вы можете провести своих героев на яхту.

Приведенная выше распечатка текста программы может быть еще и еще более усовершенствована.

Так, в строках 110 и 120 персонаж выбывает из борьбы только если его силы падают ниже -9. Вы можете принять здесь в рассмотрение и фактор мужества, используя для этого логический оператор OR.

Еще одним дополнением может быть, например, кинжал, лежащий у двери капитанской рубки. Здесь возможны варианты. Может быть, один из героев подберет кинжал, пока другой ведет схватку, а может быть Вы решите, что его подбирать надо только когда оба противника подавлены.

Добавив несколько новых строк и строк PRINT. Вы придадите программе элемент неожиданности, совершенно очевидный дух искусственного интеллекта, а Инф и Ора получает дополнительный шанс остаться в живых.

Такие объекты, как кинжал, легко хранить в памяти программы и несложно организуется их перемещение их вместе с персонажами. В распечатке на рис. 3 показан один из методов хранения объектов в локациях до того, как персонажи их подобрали. Каждая локация имеет группу предметов и хранит их в переменной, скажем B\$. Как только изменяется локация, изменяется и содержимое переменной. Если в данной локации предметов нет, то возвращается пустая строка. Размещение многих объектов в одной строковой переменной вместо массива позволяет значительно снизить расход памяти компьютера.

Формат, в котором названия объектов хранятся в b\$, довольно очевиден и может обслуживаться парой строк.

Переменная начинается с астериска (*), им же отделяются объекты друг от друга.

Первая буква в названии объекта указывает на его тип.

О – оружие

П - обычный предмет

В - враждебное существо.

Если бы в Вашей программе участвовали товарно-денежные отношения, можно было бы ввести и еще один класс ценностей, отнеся к ним драгоценные камни, золото, монеты, украшения и пр. Очевидный класс составляют магические предметы.

Следующие символы содержат информацию о названии объекта.

```
10 LET b$="*пята*окинжал*вБрог*вКран"  
20 INPUT a$  
25 LET c$a$(1 TO 3)  
30 LET k=1  
50 IF b$(k)="*" THEN GO TO 90  
60 IF k=LEN (b$) THEN PRINT "Объект не найден": GO TO 20  
70 LET k=k+1  
80 GO TO 50  
90 LET m=k+2: LET c=m+2  
100 IF c$b$ (m TO c) THEN GO TO 120  
110 GO TO 70  
120 IF b$(k+1)="о" THEN PRINT a$; " : "; "Вы нашли оружие"  
130 IF b$(k+1)="п" THEN PRINT a$; " : "; "Вы нашли предмет"  
140 IF b$(k+1)="в" THEN PRINT a$: " : ": "Перед Вами появился противник"  
150 GO TO 20
```

Переменная b\$ задается в строке 10.

В строке 20 вводится название предмета и в строке 25 оно "урезается" до трех символов. Цикл в строках 50...80 ищет появление астериска. Когда он найден, выполняется переход на строку 90. где символ за символом сканируется символьная переменная.

В строках 120. .. 140 определяется тип объекта и информация об этом доводится до играющего.

Если же этот объект не найден в b\$, то сообщение об этом генерируется в строке 60.

А теперь продолжим наш проект. Когда Инф и Ора обнаружили кинжал, подняли его, победили чудовищ и прошли в дверь, их ослепил яркий свет. Они оказались в подземном городе короля Кельроса. Их захватили в плен и повели на допрос к королю.

Самое первое средство, которое стоит попробовать для выхода из такой ситуации - попытаться договориться. В следующей части мы рассмотрим технику общения между персонажами.

(Продолжение следует).

FORUM

В нашей почте сентября лежит сразу несколько писем, которые могли бы претендовать на приз "Лучший вопрос месяца", если бы мы осмелились устроить такой конкурс. Но наши возможности имеют вполне осязаемый предел и не позволяют так рисковать.

Привидения в ОЗУ

Во-первых, поступило обращение от военнослужащего из г. Мурманска т. Новикова В. П. сомневающегося в надежности своей машины. Вот в двух словах суть его проблемы.

Экспериментируя с оператором РЕЕК, он обнаружил, что значительная часть верхнего пространства памяти компьютера содержит нули, что и не удивительно - память девственно чиста, но на самом верху, в районе адреса 65300 содержатся какие-то числа.

Попытка их "очистить", т.е. принудительно обнулить, в одних случаях ни на что не влияет, в других выводит компьютер из строя (конечно не физически, а только программно). Более того, многие из них просто нестабильны. Если в ОЗУ уже размещена какая-то Бейсик-программа, то эти числа меняются без видимой закономерности, что затрудняет ему борьбу с ними за очищение памяти от "мусора".

Прежде всего, мы должны успокоить нашего читателя - с компьютером все в порядке, просто он провел эксперименты с одним из разделов оперативной памяти компьютера. Те, кто имеет нашу разработку "Большие возможности Вашего Спектрума", могут найти в ней карту памяти и увидеть, что в исходном (после включения) состоянии этот участок памяти отводится для хранения символов графики пользователя (UDG), а также для организации в нем вспомогательных служебных структур - стеков. Это стек GO SUB и машинный стек. Содержимое этой области находится под управлением ПЗУ компьютера и потому может изменяться, даже если пользователь об этом и не догадывается.

Напомним для начинающих карту памяти "Спектрума". Нижние 16К занимает ПЗУ (содержащее необходимые процедуры операционной системы, о чей мы достаточно подробно пишем в разделе "Секреты ПЗУ"). Далее примерно 7К занимает экранная область памяти, потом идет буфер принтера, системные переменные и пр. И наконец - свободное пространство (более 2/3 общей памяти). Это "свободное пространство" подвержено постоянным изменениям. Это не чистый лист бумаги, на котором можно что-то написать и стереть. Оно непрерывно "пульсирует", то увеличиваясь, то уменьшаясь, выделяя свои просторы для временного использования по мере необходимости для различных нужд. Контроль же за всеми этими процессами ведется в установках значений системных переменных и в данных на стеках, находящихся на вершине памяти (вот почему они могут меняться и почему их нельзя "портить").

Возьмем простой пример, Вы нажали клавишу EDIT и видите как строка из листинга Вашей программы на экране копируется в нижнюю часть экрана, где Вы можете вносить в нее свои изменения и дополнения. Вам и невдомек, что в это же время в "свободной" области памяти происходят бурные события. Выделяется место под область редактирования, передвигаются вверх буфер INPUT и стек калькулятора, копируется содержимое Вашей строки из программной области в область редактирования и прочее и прочее.

Вы нажали ENTER и отредактированная строка заняла свое место в программе. И это опять "раздвигание" программной области, чтобы втиснуть измененную строку, смещение всего, что было выше ее вверх, копирование, сворачивание области редактирования и, наоборот, сдвиг вниз буфера INPUT и стека калькулятора.

Кстати, обратите внимание на то, что после такого сдвига вниз освободившиеся ячейки станут "свободными", но не нулевыми, как после включения компьютера. Компьютер не тратит драгоценное время на очистку (если обнуление это очистка) освободившихся областей памяти. Ему достаточно указать в нужных адресах, что то, что там находится - "чисто и свободно", не важно, равно оно нулю или нет.

То же происходит и со стеками, находящимися в верхних адресах компьютера. Они могут расти (растут они вниз) или сжиматься, но по мере их сжатия освобождающиеся ячейки памяти присовокупляются к свободному пространству, но не обнуляются. Там остается "эхо" предшествующих событий и держится оно до тех пор, пока стек, разрастаясь сверху вниз вновь не проутюжит эти адреса или пока снизу не подступят рабочие области по мере развития Вашей Бейсик-программы.

Что такое стек?

Продолжая эту тему, ответим также на вопрос В. Сивцева из Воронежа о том, что такое стеки GO SUB и машинный стек, о которых так часто упоминается.

Большинство наших читателей конечно знает, что делает команда GO SUB - вызывает исполнение подпрограммы. Когда ПЗУ встречает в Вашей программе команду GO SUB, то сначала запоминается то место, в котором она встретилась - номер строки и номер оператора в строке.

Когда же встретится команда RETURN, то эти запомненные данные и послужат для точного возврата именно туда, откуда подпрограмма вызывалась.

Стек GO SUB - и есть тот участок памяти, в которой это все и запоминается. Он растет сверху вниз и работает по принципу "последним пришел - первым уйдешь". Т.е. при команде RETURN со стека снимаются данные, поступившие в последней GO SUB. Легко представить себе, что вызвав через GO SUB подпрограмму, Вы в ней опять применяете GO SUB для вызова другой и так далее. При этом стек будет расти все больше и больше. Совершенно очевидно, что возвращаясь из этих подпрограмм через RETURN, Вы снимете все заложенные на стек данные, он сожмется и придет в исходное состояние. Кстати, если Вы из подпрограмм будете возвращаться через GO TO, то такой нежный баланс на стеке будет нарушен и работоспособность Вашей программы будет висеть на волоске.

Машинный стек находится чуть ниже стека GO SUB и непосредственно граничит с верхней границей свободного пространства. Соответственно, когда стек GO SUB расширяется и сжимается, то машинный стек копируется то вниз, то вверх.

Машинный стек аналогичен стеку GO SUB по природе, но с ним работает не Бейсик-программа, а сам процессор Z-80. Там процессор запоминает адреса процедур, когда они вызывают другие процедуры и наоборот, снимает эти адреса при возврате. Вторая его функция - временное место хранения содержимого регистров процессора или каких-либо ячеек памяти, когда их нужно освободить для каких то срочных дел, но нельзя их потерять. Для этого в системе команд Z-80 есть даже две группы команд - PUSH (команды этой группы отправляют на стек данные на хранение), и команды POP (они восстанавливают данные, снимая их со стека).

Так, например, PUSH BC отправит на стек содержимое регистров В и С, а команда POP HL снимет то, что последних помещалось на стек (2 байта) и зашлет их в регистры H и L.

Обратите внимание еще и на тот факт, что если Вы никогда в машинном коде (на Ассемблере) не программировали и не собираетесь, то и машинный стек Вам вроде бы не нужен? - Ничего подобного!

Вполне достаточно того, что любое Ваше действие на компьютере (а также и бездействие) сопровождается исполнением сотен процедур, находящихся в ПЗУ, которые работают в машинном коде и без этого стека жить никак не смогут.

Зачем нужны прерывания?

Еще один вопрос, интересующий десятки наших читателей в последнее время - это использование прерываний в "Спектруме". Что такое прерывания? Зачем они нужны?

Надо сказать, что кое-что об этом есть в нашем трехтомнике по программированию в машинных кодах, но видимо недостаточно. В нашем портфеле сейчас заготовлен обширный материал на эту тему, выполненный Баяновым К. К. из Томска по нашему заказу, но на финише года мы должны в первую очередь завершить начатые статьи и циклы, поэтому этот материал дадим в следующем году, а пока ограничимся несколькими словами о прерываниях.

Система прерываний - это средство, с помощью которого процессор может прекратить исполнение одной задачи и перейти к другой, а потом опять вернуться туда, где прервал работу.

Все компьютеры имеют систему прерываний. На самых совершенных машинах работа может быть организована например так, что процессор исполняет свою главную задачу, но иногда обращается к вспомогательным, которые в свою очередь распределены по приоритетам (по степени важности) и более приоритетные выполняются в первую очередь. По мере исполнения каждой работы компьютер возвращается к предыдущей незаконченной задаче.

Процессор Z-80 может исполнять прерывания двух основных типов.

Первый тип - немаскированное прерывание (NMI). Оно так названо потому, что программист не может его предотвратить или направить на исполнение своей задачи (маскировать). Процессор неотвратно исполняет это прерывание, когда оно происходит.

Маскируемое прерывание более интересно, т.к. может быть использовано и потому, что оно имеет три разных режима (IM0, IM1 и IM2).

В нулевом режиме процессор просто ждет команды от внешнего устройства на то, чтобы прервать свою работу и обратиться к исполнению программы, заложенной в известном месте памяти. В фирменной машине и в большинстве самодельных аналогов нет средств для работы в этом режиме.

Первый режим (IM1) - основной. "Спектр" конструктивно исполнен так, чтобы всегда работать в нем. В этом режиме процессор прерывает работу, запоминая в регистре PC адрес, в котором он прервался, и обращается в ПЗУ по адресу 0038H. Процедура ПЗУ, расположенная в этом месте обеспечивает "тикание" внутренних часов "Спектра" и сканирование клавиатуры в поисках нажатой клавиши. Если клавиша была нажата, тут же изменяются значения флагов в системных переменных (см. раздел "Секреты ПЗУ") и далее работа продолжается.

Такое "тикание" происходит 50 раз в секунду (это связано с тем, что в нашей электрической сети частота тока - 50 Гц). Если бы Вы жили в Англии или в США, Ваш компьютер сканировал бы клавиатуру чаще - 60 раз в секунду.

Самый серьезный, хотя и реже используемый режим - второй (IM2). В нем после прерывания процессор обращается не в одну и ту же точку ПЗУ, как в IM1, а туда, куда задаст сам программист. Адрес перехода по этому прерыванию вычисляется следующим образом.

Он состоит из двух байтов. Старший байт процессор берет из регистра I (который так и называется - вектор прерываний). Программист может задать его так, как ему надо. Второй байт адреса (младший) процессор принимает от внешнего периферийного устройства, выдавшего этот сигнал прерывания.

Адрес = $256 \times (\text{содержимое I}) + \text{то, что поступило по шине данных}$. Далее компьютер смотрит, что содержится по этому адресу (и в следующем) и определяет, куда ему перейти на исполнение новой задачи.

Новый адрес = $\text{содержимому адреса} + 256 \times \text{содержимое (адрес+1)}$.

Пример:

Регистр I содержит число 143. От периферии поступило число 27, расчет адреса дает:
 $256 \times 143 + 27 = 36635$ (и 36636)

Если в этих ячейках содержатся числа 137 и 93, то будет сделан переход на адрес:
 $137 + 256 \times 93 = 23945$

Таким образом, программист в нужном ему месте может организовать таблицу переходов, содержащую 126 адресов. Длина этой таблицы будет 256 байтов. Далее он выставит регистр I так, чтобы он всегда указывал на начало этой таблицы, и положение адреса перехода по прерыванию будет зависеть от внешнего устройства. Так, можно обеспечить работу компьютера со 128-ью разными внешними устройствами, каждое из которых обслуживается своей личной программой (драйвером).

Пока мы на этом остановимся, а в будущем году дадим развернутый материал на эту тему, причем приложим к нему несколько образцов программ-драйверов.

К вопросу о совместимости.

Нам было очень приятно получить письмо из Елгавы (Латвийская республика) от нашего читателя Масолова Ю. В.

Вы помните, что в начале года на наших страницах проводилось исследование почему не работает на многих самодельных машинах программа "Арканойд". Было установлено, что процессор в фирменной машине при чтении внешнего порта "подхватывает" байт атрибутов и в этом вся проблема и заключалась.

Нашему читателю удалось таким же образом запустить TOP GUN, DUET и многие другие программы.

Конечно же, нам очень приятно, что проблемы, поднятые на наших страницах, приводят в итоге к практическим решениям.

Вместе с тем, как указано в письме, есть еще проблемы с прерываниями INT. В разных схемах этот сигнал формируется по разному. В результате опытов удалось установить, что этот сигнал должен появляться в НАЧАЛЕ КАДРОВОГО СИНХРОИМПУЛЬСА и сбрасывается через M1 и MREQ, а если прерывания запрещены, то через 10 мксек. Если же сигнал начинается в другое время, имеются следующие проблемы:

1. Дрожат спрайты на экране (ARKANOID и др.).
2. Исчезает курсор в экране (ARTSTUDIO).
3. Линии в бордюре не совпадают с фоном экрана.

О необходимости разобраться с сигналом INT говорит и наш читатель Хвиш А. А. из г. Луцка. Им также установлено, что этот сигнал оказывает большое влияние на программную совместимость с родительской машиной. Так, на его "Балтике", как и на машинах его близких друзей, отказывались работать программы "Samantha Fox", "Turbo Comp" и др.

Идею по доработке компьютера он нашел в минском журнале "Радиолюбитель" и, как пишет наш читатель, "Теперь компьютер работает отлично".

В этом журнале есть и немало прочей ценной информации для любителей нашей системы. Так, например, сообщается о том, что минский кооператив "Сонет" поставляет желающим ПЗУ 27256 с записью в старшей 16-килобайтной области системы "Disk-monitor", копировщика ZX-COPY, программы "Assembler" и базовой системы ввода/вывода (BIOS) CP/M.

Данный пакет дает возможность не только останавливать любые программы, вносить в них изменения и запускать с места останова, но и писать свои. Выгрузка программ на ленту или диск выполняется несложно с помощью контроллера. Загрузчик BIOS делает возможной работу в среде CP/M, причем без каких-либо существенных аппаратных изменений.

Принцип работы заключается в переписывании верхних 16К в область теневого ОЗУ.

Наш читатель опробовал этот пакет на своем "Балтике" и остался очень и очень доволен. К сожалению, у нас нет информации от кооператива "Сонет" по этому вопросу, но если кто-то из приближенных к нему читателей может содействовать установлению такой связи, мы с радостью опубликуем более широкую информацию об их разработках, ведь судя по нашей почте, очень многих интересует работа в среде CP/M.

Но есть и новые вопросы, новые проблемы.

Во-первых, у нашего постоянного читателя из г. Киева Довженко В. П. не идет программа PSI-CHESS на Львовском варианте компьютера. Что это - случайность или за этим что-то стоит?

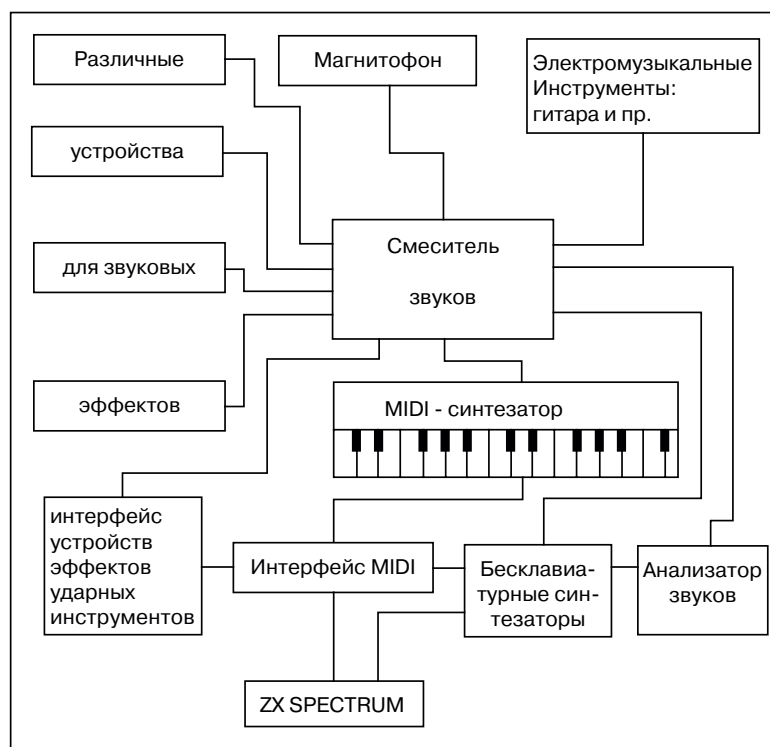
Есть также сообщения и о том, что скорость игр, запущенных на "Ленинградской" версии заметно ниже, чем на других версиях, например на "Балтике". И это скорее всего не случайность.

Интерфейс MIDI

Еще один часто встречающийся вопрос касается интерфейса MIDI, который служит для подключения к компьютеру музыкальных инструментов, синтезаторов и прочих музыкальных устройств.

Читателей интересует все: описания, схемы, прошивка ПЗУ, программная поддержка.

К сожалению, мы не располагаем по этому вопросу никакой информацией. Может быть в стране просто еще нет разработок, привязанных к отечественной элементной базе, но если все же что-то есть, то мы с радостью приобрели бы любую информацию, столь нужную нашим читателям.



Для тех, кто пока просто не задумывался над этим вопросом, мы дадим блок-схему подключения музыкальной системы к "Спектруму", в основу которой положен интерфейс "MIDI" и синтезатор звуков, например YAMAHA DX-100. Надеемся, это зрелище впечатлит тех, кто может заняться такой разработкой. Из схемы наглядно видно, к каким системам открывается доступ, если есть в наличии MIDI-интерфейс. Схема взята из журнала Sinclair User.

Заканчивая сегодня раздел "Форум", мы хотим порекомендовать Вам достать и прочитать журнал "Компьютер" (N1 за 1991 год). Этот совместный советско-польский журнал формируется по материалам польского ежемесячного журнала "Kompjuter" и выходит раз в квартал. Журнал известен среди пользователей компьютеров разных систем, в основном он посвящен IBM-совместимым машинам, но данный номер был почти полностью предоставлен системе "СИНКЛЕР" и содержит весьма интересные и поучительные материалы.

Мы не знаем, как дело обстоит во всей стране в целом, но в Москве и Ленинграде он еще есть в продаже и приобретается без особых проблем.

Мы очень часто получаем письма с просьбой посоветовать над чем бы поработать нашим читателям, умеющим программировать и желающим приложить свои усилия с пользой для себя и общества.

Рецепт очень прост - в нем всего пять слов: "Найдите потребность и удовлетворите ее". (Рецепт взят из недавно вышедшей книги "Как стать предприимчивым и богатым. Из американских рецептов." М., "Молодая гвардия", 1991)

Нужна помощь.

Требуются обучающие программы для ПК "Спектрум". 183060, Мурманск-60, ул. Мира, д. 5/4, кв. 221, Гершковичу М. Б.

Для применения в медицинской практике требуется тех. документация и

Компьютер в моем городе.

Сегодня мы обращаемся ко всем нашим читателям с предложением принять участие в крупной акции по созданию летописи развития движения самостоятельного компьютерного творчества в нашей стране.

То явление, которое мы сейчас переживаем, связанное с огромным проникновением компьютерных знаний в самые широкие круги, уникально. Это подлинно народная компьютерная революция. И все Вы ее творцы и активные участники.

Народ сам, не спросив никого, и не дожидаясь милости от государственных комитетов по информатике и по народному образованию оснастил себя колоссальным парком машин.

Сегодня в стране по нашим оценкам 1.5 - 2 миллиона пользователей "Спектрума". Можно сколько угодно восхищаться гением сэра Клайва Синклера, создателя этого компьютера, но мы то абсолютно точно знаем сколько у нас своих гениев. Тех, кто "на коленке", разрабатывал и из немыслимых деталей собирал советские "Спектрумы", когда официальные журналы упивались достоинствами "Микроши".

А через что прошли заводы, которые начали выпускать первые "Спектрумы"! Каким мужеством обладали их руководители, не побоявшиеся срыва планов основного производства и взявшиеся за совершенно новое, рискованное и не пользующееся официальной поддержкой дело!

Нет в мире другой страны, в которой были бы еще самодельные "Спектрумы". А ведь сколько разных версии разработано и освоено Вами!

Можно только поражаться, какой же волей должен обладать народ, который без малейшей поддержки и даже вопреки ограничениям на ввоз процессоров Z-80 (и такое было!) поднял в стране уровень компьютерной грамотности и довел ее до миллионов.

Мы обращаемся с просьбой ко всем. Пока все это живо в памяти, именно в эти трудные для страны дни давайте запишем все, что мы знаем и помним об этом пути.

Пройдут годы и мы уверены: об этом периоде будут говорить с должным уважением. Давайте соберем наши общие знания в единую летопись.

Пишите все, что знаете. Когда и как в Вашем городе появились первые компьютеры? Какие модели? Чем сейчас живет город? Кто выпускает машины в Вашем регионе? Какие программы пользуются наибольшим авторитетом и какие модели машин? Кто и как их разрабатывал, кто что предложил нового? Где Вы собираетесь и как? Сколько Вас? Сколько в городе машин и каких? Есть ли учебные классы и школы, оснащенные компьютерами?

Нам нужны и адреса производителей "Спектрумов". Самых крупных мы, конечно знаем, но ведь их сейчас сотни.

Если Вы живете не в городе, а в поселке или селе - это не имеет никакого значения - нам важно знать все.

Если Вы лишь недавно приобрели компьютер и не имеете информации о прошлом - напишите, как Вы видите день сегодняшней. Ведь завтра и он станет историей.

Нам обычно пишут, когда с компьютером что-то плохо. Почему бы не написать, когда с ним все хорошо (и сказать, что это за модель и какой завод ее сделал).

Особо ценными были бы для нас заметки и воспоминания тех, кто были первыми и разрабатывали первые версии.

Одним словом, нас интересует ВСЕ! Мы будем скрупулезно накапливать и хранить всю информацию. Как знать, ведь если этого не сделать сейчас, она может быть безвозвратно утрачена! Много ли сейчас узнаешь о тех людях, которые делали первый российский (украинский, белорусский...) автомобиль?

У нас есть мечта - когда-нибудь создать музей бытовых персональных компьютеров. Чтобы лет через двадцать все могли прийти и увидеть, как это начиналось.

Пока это только мечта, но если Вы, уважаемый читатель, внимательно следите за нашим развитием, то знаете, что мы хоть и медленно, но неуклонно идем своим путем. И мы

обязательно создадим такой музей, когда наберем для этого достаточно сил.

Обращаясь к Вам с такой просьбой мы преследуем еще одну цель.

Мы хотим выявить среди Вас тех, кто умеет вести маркетинговые исследования, обладает для этого достаточной коммуникабельностью, контактностью, способностью собирать и анализировать информацию.

Крупные разработчики и изготовители компьютеров, периферийных устройств, поставщики программного и информационного обеспечения нуждаются в надежных маркетинговых исследованиях, прежде чем предпринимать какие-либо шаги. До сих пор мы неплохо удовлетворяли их потребности, но количество заказов растет и мы будем формировать коллектив внештатных исследователей, которые по нашему заказу смогут проводить такие оплачиваемые работы.

Наш каталог.

Вышел из печати и сейчас рассылается подписчикам наш каталог игровых и прикладных программ для "Спектрума". Это получился солидный труд объемом 150 страниц, включающий более 5000 программ.

Мы высылаем его тем, кто в свое время (три месяца назад) подписался на предложенных нами условиях. Мы полагаем, что он обрадует всех и они не пожалеют о том, что перечислили средства "ИНФОРКОМу".

Для тех, кто в свое время не оформил подписку на него, сообщаем, что установлена цена в размере 45 рублей. Заказы принимаются.

Мы надеемся на то, что этот каталог станет общепринятым средством для общения любителей между собой при обменах, переписке, составлении собственных каталогов и т. п.

Мы также планируем один раз в год дополнять его новым содержанием и держать его актуальность на должном уровне.

ИНФОРКОМ ПРЕДЛАГАЕТ...

Наши читатели привыкли ежемесячно вместе с очередным выпуском "ZX-РЕВЮ" получать информационный листок, начинающийся с этих слов.

В этом месяце Вы не получите такого листа. Причина в том, что мы ведем пересмотр цен в связи с общей экономической ситуацией и должны какое-то время подождать, пока пройдут заказы по отправленным ранее листкам с устаревшими ценами.

Просим откликнуться.

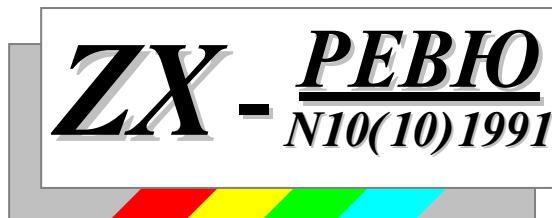
Мы убедительно просим откликнуться нашего читателя, приславшего для нас на дискете 5,25" рисунки CLOTH, ROBOT и SUNRISE. При передаче между подразделениями утерян конверт с Вашим обратным адресом, а нами для Вас кое-что подготовлено.

Что бы это значило?

Любителям поломать голову предлагаем набрать нехитрую программу и попытаться объяснить как же она работает.

```
10 FOR k = 72 TO 79
20 POKE 23681,k
30 LPRINT "GOOD LUCK TO YOU"
40 NEXT k
```

Тем, кому этот пример доставил удовольствие, рекомендуем в 1992 году читать в "ZX-РЕВЮ" сериал В. С. Михайленко (Минск), посвященный методам защиты программ от копирования и борьбы с ними. Пример взят оттуда.



Спектрум в школе

Сегодня мы печатаем материал, подготовленный участниками компьютерного кружка под руководством С. В. Митина (г. Ульяновск).

Первая программа может пригодиться на уроках математики. Она предназначена для построения графиков.

Графики функций.

```

10 REM ** Построение графиков **
20 REM ** Начальная установка **
110 CLS
120 LET sx=176
130 LET sy=160
140 LET hy=sy/2
150 CLS
160 PRINT AT 8,10; "Построение графиков"
170 PRINT
180 PRINT "1. Y=X*X*SIN(1/X)"
190 PRINT
200 PRINT "2. Y=X*SIN(1/X)"
210 PRINT
220 PRINT "3. Y=SQR(X*X+2)"
230 PRINT
240 PRINT "4. Y=COS(X*EXP(-X/5))"
250 PRINT
260 PRINT "5. Y=6+2*X*X-X*X*X*X"
270 PRINT
280 PRINT FLASH 1; TAB 5;"Укажите номер уравнения"
300 INPUT N
310 IF N=1 THEN DEF FN A(x)=x*x*SIN(1/x)
320 IF N=2 THEN DEF FN A(x)=x*SIN(1/x)
330 IF N=3 THEN DEF FN A(x)=SQR(x*x+2)
340 IF N=4 THEN DEF FN A(x)=COS(x*EXP(-x/5))
350 IF N=5 THEN DEF FN A(x)=6+2*x*x-x*x*x*x
360 PRINT
370 PRINT "Область изменения X"
400 INPUT "Ввести наименьшее и наибольшее значение ";a,b
450 IF a>b THEN PRINT FLASH 1; "Ошибка - попробуйте еще."
460 IF a>b THEN GO TO 370
500 PRINT "Вычисление области значений Y."
520 LET c=(b-a)/100
530 LET m=1.0E-30
540 FOR x=a TO b STEP c
550 IF x=0 THEN GO TO 580
560 LET y=ABS(FN A(x))
570 IF m<y THEN LET m=y
580 NEXT x
590 REM *Готов построить графики
600 PAUSE 500: CLS
700 LET c=c/10
720 FOR x=a TO b STEP c
730 IF x=0 THEN GO TO 790
740 LET y=FN A(x)

```

```

750 LET u=sx*(x-a)/(b-3)
760 LET v=hy+hy*y/m
770 IF v<0 OR v>sy THEN GO TO 790
780 PLOT u,v
790 NEXT x
800 REM *окончание и повторение*
810 LET g$=INKEY$
820 IF g$="" THEN GO TO 810
830 CLS
850 PRINT "Повторить? Y/N"
860 LET g$=INKEY$
870 IF g$<>"Y" AND g$<>"y" AND g$<>"N" AND g$<>"n" THEN GO TO 860
880 IF g$="Y" OR g$="y" THEN GO TO 150
890 STOP

```

С помощью этой программы Вы можете анализировать графики и каких-то своих функций. Для этого надо изменить расчетную формулу, по которой она будет вычисляться (строки 310 - 350), а также поменять запись формулы в меню (строки 180 - 260). Обратите особое внимание, чтобы при обработке Вашей функции не происходило деления на ноль. Если в формуле есть деление, то либо область изменения знаменателя не должна содержать ноль, либо перед расчетом надо проверять знаменатель и, если он равен нулю, обрабатывать этот случай особо или вообще его пропускать, не рассматривая.

Поэкспериментируйте с предложенной Вам программой, попробуйте менять значение с в строке 700. Вместо с/10 попробуйте с/5 и с/20.

Графики получается симпатичнее, если левое значение x взять отрицательным, а правое положительным.

А вот еще одна небольшая программа для урока математики:

Исследование матриц.

```

10 REM **Исследование матриц**
100 REM **Начальная установка**
110 CLS
120 LET sx=176: LET sy=160
130 LET rt=.6
140 LET hx=sx/2: LET hy=sy/2
170 REM ** Ввод матрицы**
200 PRINT
210 PRINT "Введите свою матрицу"
220 PRINT
230 INPUT"Строка 1, столбец 1":m1
240 PRINT
250 INPUT"Строка 1, столбец 2":m2
260 PRINT
270 INPUT"Строка 2, столбец 1":m3
280 PRINT
290 INPUT"Строка 2, столбец 2":m4
320 PRINT
330 PRINT"Ваша матрица: "
340 PRINT TAB 4; m1; TAB 10;m2
350 PRINT TAB 4; m3; TAB 10;m4
360 PRINT
370 LET det=m1*m4 - m2*m3
380 PRINT "Определитель D=";det
390 PRINT "Хотите продолжить Y/N"
400 PRINT
430 LET g$=INKEY$
440 IF g$<>"Y" AND g$<>"y" AND g$<>"N" AND g$<>"n" THEN GO TO 430
450 IF g$="N" OR g$="n" THEN GO TO 180
460 PRINT "Ладно, но немного терпения, пожалуйста"
470 PAUSE 250
480 LET dd=ABS(det)
490 IF dd<1 THEN LET dd=1: BEEP .1,2

```

```

500 CLS
600 REM ** Картинка **
610 FOR x=0 TO 15 STEP 7.5
620 FOR y=0 TO 15 STEP 1/dd
630 GO SUB 910
640 NEXT y
650 NEXT x
660 FOR y=0 TO 15 STEP 15
670 FOR x=0 TO 15 STEP 1/dd
680 GO SUB 910
690 NEXT x
700 NEXT y
710 LET y=12
720 FOR x=5 TO 10 STEP 1/dd
730 GO SUB 910
740 NEXT x
800 REM *окончание и повторение*
810 PAUSE 0
820 PRINT "Повторить? Y/N": PAUSE 500
830 CLS
840 LET g$=INKEY$
850 IF g$<>"Y" AND g$<>"y" AND g$<>"N" AND g$<>"n" THEN GO TO 840
860 IF g$="Y" OR g$="y" THEN GO TO 180
870 STOP
900 REM ** вычерчивание **
910 LET u= (m1*x+m2*y)*rt+bx
920 IF u<0 OR u>sx THEN RETURN
930 LET v=hy+m3*x+m4*y
940 IF v<0 OR v>sy THEN RETURN
950 PLOT u,v
960 RETURN

```

Сейчас, когда страна вступает на путь рыночных отношений, молодежи могут пригодиться знания, связанные со способами и методами вложения и приумножения капитала. Итак, сегодня мы рассмотрим проект "ИНВЕСТОР".

Для справки: ИНВЕСТОР - человек, который вкладывая свой капитал способствует развитию какого-то предприятия и резонно рассчитывает получить за это часть прибыли. От того, как он сумеет определить перспективность того или иного предприятия, зависит получит ли он прибыль.

В реальной жизни он руководствуется при этом своими знаниями (знания никогда не бывают лишними), опытом, интуицией, личными убеждениями, но может привлекать консультантов и, конечно, рискует.

Итак, неизвестный благодетель предоставил Вам 1000 долларов с условием, что Вы вложите их в развитие трех фирм X, Y, Z. Вы обязаны вкладывать ВСЕ имеющиеся у Вас деньги в течение 10 лет. Для этого Вы раз в год сообщаете компьютеру о том, какую сумму Вы вкладываете в то или иное предприятие.

У Вас есть мощный компьютер и свой человек в каждой из трех фирм, так что Вы в состоянии оценить прибыль (убыток) на один доллар в каждой фирме в зависимости от состояния рынка.

Всего возможны три состояния, и Вам компьютер подсказывает с какой вероятностью наступит то или иное состояние, а вот решение по тому, в какую фирму сколько вкладывать примете Вы сами. Возможная прибыль или убыток на один доллар выводится на экран в виде таблицы, например:

Фирма	Состояния рынка		
	ST 1 0.222	ST 2 0.413	ST 3 0.365
X	.4	-.1	.3
Y	-.2	.1	.1
Z	-.1	.4	-.3

Здесь вероятность того, что рынок окажется в 1-ом состоянии - 0.222, (т.е. в 22.2% случаях). Если оно произойдет, то в предприятии X каждый вложенный доллар принесет 0.4 доллара прибыли, в фирме Y Вы понесете убыток 0.2 долл., а в фирме Z - 0.1 \$. Таким образом, если Вы вложили в X, Y и Z соответственно по 300, 300 и 400 долларов, а состояние рынка было ST 1, то результат Вашей инвестиции будет: $300 \cdot 0.4 + 300 \cdot (-0.2) + 400 \cdot (-0.1) = 20$ долл.

Теперь Вы будете обладателем 1020 долларов и в конце года будете снова принимать решение, куда Вам их вложить. Достаточно указать сколько будет вложено в X и Y, а остаток компьютер сам переправит в Z.

Цена игры - порядка 5%. Так что, если Вы будете играть оптимально, то должны в среднем приумножать свой капитал на 5% в год. Играя рискованно, Вы можете разбогатеть или разориться чуть быстрее.

```
10 REM  ИНВЕТОР
100 REM  **Начальная установка**
110 CLS
120 DIM a(4,3,10)
130 FOR k=1 TO 10
140 FOR i=1 TO 4
150 FOR j=1 TO 3
160 READ a(i,j,k)
170 NEXT j
180 NEXT i
190 NEXT k
200 LET w=1000
210 LET year=1
300 REM  ** ЦИКЛ **
310 CLS
315 PRINT TAB 12; "ИНВЕТОР"
320 PRINT TAB 13; "ГОД: "; year
330 PRINT
340 PRINT "Капитал в начале года равен ";w
350 PRINT
360 PRINT TAB 8; "Состояние рынка"
370 PRINT TAB 8;"ST 1   ST 2   ST 3"
380 LET K=INT(RND*10+1)
390 FOR i=1 TO 4
400 IF i=2 THEN PRINT "X";
410 IF i=3 THEN PRINT "Y";
420 IF i=4 THEN PRINT "Z";
430 FOR j=1 TO 3
440 IF i=1 THEN PRINT TAB (3+6*j); a(i,j,k);
450 IF i>1 THEN PRINT TAB (3+6*j); a(i,j,k);
460 NEXT j
470 IF i=1 THEN PRINT
480 PRINT
490 NEXT i
500 PRINT
510 INPUT "Сколько Вы хотите вложить в X?"; x
540 LET x=INT(x+0.5)
550 IF x<0 OR x>w THEN PRINT FLASH 1; "Ошибка"
560 IF x<0 OR x>w THEN GO TO 510
570 INPUT "Сколько Вы хотите вложить в Y?"; y
590 LET y=INT(y+0.5)
600 IF y<0 OR y>(w-x) THEN PRINT FLASH 1: "Ошибка"
610 IF y<0 OR y>(w-x) THEN GO TO 570
620 LET z=w-x-y
630 PRINT "Остаток ";z;" вложен в Z"
640 REM  ** Состояние рынка **
650 LET q=RND
660 LET r=3
670 IF q<a(1,1,k)+a(1,2,k) THEN LET r=2
680 IF q<a(1,1,k) THEN LET r=1
690 PRINT " Состояние рынка "; r
```

```

700 REM ** Прибыль и убыток **
710 LET w= w+a(2,r,k)*x + a(3,r,k)*y + a(4,r,k)*z
720 LET w=INT(w+.5)
730 PRINT "Ваш капитал теперь равен: ";w
740 PRINT
750 PRINT "Чтобы продолжить нажмите Y"
760 LET g$=INKEY$
770 IF g$ <> "y" AND g$ <> "Y" THEN GO TO 760
780 LET year=year+1
790 IF year<11 THEN GO TO 300
800 REM *Окончание и повторение*
810 PRINT
820 PRINT "Вы начали с 1000, а теперь у Вас уже: ";w
830 PRINT
840 LET b$ = " *Потрясающий результат* "
850 IF w<2500 THEN LET b$= " *Отлично* "
860 IF w<2000 THEN LET b$= " *Хорошо * "
870 IF w<1750 THEN LET b$= " *Неплохо* "
880 IF w<1400 THEN LET b$= " *Жить можно* "
890 IF w<1000 THEN LET b$= " *В другой раз повезет * "
900 PRINT b$
910 PRINT
1000 REM ** Повторение **
1010 PRINT "Повторить? Y/N"
1020 LET g$ = INKEY$
1030 IF g$<>"Y" AND g$<>"y" AND g$<>"N" AND g$ <>"n" THEN GO TO 1020
1040 IF g$="Y" OR g$ = "y" THEN GO TO 200
1050 STOP
1100 REM ** Данные **
1110 DATA .478, .337, .185, .3,
      -.7, .7, -.3, .7,
      -.3, .2, .1, -.5
1120 DATA .106, .53, .364, -.5,
      -.3, .7, .5, -.5,
      .7, .3, .5, -.7
1130 DATA .415, .347, .237, .3,
      -.7, .7, -.3, .7,
      -.3, .4, 0, -.5
1140 DATA .247, .346, .38, .5,
      -.7, .4, -.3, .7,
      -.3, -.5, .2, .3
1150 DATA .437, .438, .125, .6,
      -.6, .4, -.6, .8,
      -.3, -.3, .5, -.3
1160 DATA .304, .435, .261, .8,
      -.7, .4, -.6, .4,
      .2, -.2, .6, -.6
1170 DATA .422, .39, .188, -.5,
      .8, -.3, .7, -.5,
      -.3, -.3, .1, .7
1180 DATA -.214, .357, .429, .3,
      -.2, .1, -.3, .4,
      -.1, .1, -.2, .2
1190 DATA .222, .413, .365, .4,
      -.4, .3, -.2, .1,
      .1, -.1, .4, -.3
1200 DATA .093, .463, .444, 0,
      -.6, .7, -.6, 0,
      .2, .2, .8, -.8

```

"ИНФОРКОМ" сыграл в эту игру десятков раз, быстро обнаружил оптимальную стратегию и ни разу не показал результата хуже "хорошо". Поздравляем авторов программы с очень хорошо сбалансированной игрой.

BETA BASIC

Часть 2. Функции.

Общий обзор.

В Бета-Бейсик добавлены десять новых функций. Их определение дано в строке 0, которая не выводится по команде LIST. Там размещено указание на блок, записанный в машинных кодах, размещенный в верхних областях памяти и выполняющий вычислительную работу. Эти функции работать не будут и вся система "зависнет", если основная машиннокодовая часть Бейсика не будет присутствовать в памяти. С другой стороны, если не будет присутствовать нулевая строка, то Вы получите резолюцию:

"FN Without DEF" ("функция не определена")

В этом случае остальная часть Бета-Бейсика будет работать нормально, Вы только не сможете использовать новые функции. При выгрузке программы на ленту выгружается и нулевая строка, поэтому, если Вы загрузите (LOAD) программу, написанную в Бета-Бейсике, то строка 0 в ней будет присутствовать. Однако, загрузка программы, записанной не в Бета-Бейсике, сотрет строку 0. Чтобы избежать этого, сначала дайте NEW, что очистит компьютер, но строку 0 оставит нетронутой, а затем загрузите новую программу через MERGE.

Нулевую строку можно удалить, например так: DELETE 0 to 0. Это позволит сократить объем памяти (строка 0 занимает до 221 байта).

1. Функция: FN C\$

Структура: FN C\$ (число)

См. также FN N(строка)

Эта функция конвертирует целые числа (от 0 до 65535) в двухсимвольные строки, что позволяет гораздо более экономно хранить числовые данные. Эквивалентом в Бейсике является:

```
LET A=INT(число/256): LET B = число - A*256 LET C$ = CHR$ A + CHR$ B
```

Результат при попытке распечатать эту строковую переменную может быть такой:

```
K,"Invalid colour"
```

Это происходит потому, что она может содержать коды управления цветом при печати.

FN N (двухсимвольная строка) обычно применяется для перевода числовых строк в числа. Поскольку здесь для хранения числа нужно всего только два байта, в то время как в обычном формате - пять, то для Вас имеет прямой смысл использовать эту функцию, если Вы имеете дело с очень большими массивами данных, причем данные не обязательно должны быть целыми.

Если у Вас есть число типа 87.643, Вы можете его умножить на 100 и получить 8764,3. Взяв целую часть 8764, Вы применяете к ней FN C\$, получаете строку из двух символов. Последующее применение к ней FN N и деление на 100 даст 87.64. Это неплохая точность для большинства практических приложений.

```
100 DIM A$ (500,2)
110 FOR E=1 TO 500: LET A$(E)= FN C$(E*10): NEXT E
120 FRINT "Array created-press any key to print it"
    (массив создан, нажмите любую клавишу).
130 PAUSE 0
140 FOR E=1 TO 500: PRINT E, FN N (A$(E)): NEXT E
```

Это массив занимает только 1К памяти, в то время как в обычном формате он бы занимал 2.5К.

Команда SORT работает с такими массивами правильно.

2. Функция FN D

Структура: FN D (строка)

См. также FN H\$(число).

Эта функция дает десятиричный эквивалент для строковой переменной, которая представляет собой допустимое шестнадцатиричное число. Строка может иметь от 1 до 4 символов.

FN D ("FF") = 255

FN D ("10") = 16

FN D ("4000") = 16384

FN D ("e") = 14

Регистр (верхний или нижний) для букв не играет роли.

Чтобы выполнить POKE с шестнадцатиричным вводом, Вы можете сделать:

```
INPUT A$: POKE address, FN D(A$)
```

Если строка не имеет символов или имеет более 4-х символов или имеет неразрешенные символы (т.е. символы кроме 0...9, A...F, a...f), то выдается сообщение:

"invalid argument" (неправильный аргумент)

3. Функция: FN H\$

Структура: FN H\$ (число)

См. также FN D (строка)

По этой команде десятиричный числовой аргумент переводится в шестнадцатиричный строковый. Он будет иметь 2 символа, если число было по абсолютной величине меньше 255 или 4 символа, если оно по абсолютной величине больше. Если число больше, чем 65535, то выдается сообщение:

B, "integer out of range"

("целое число выходит за допустимые пределы")

FN H\$ (32) = "20"

FN H\$ (255) = "FF"

FN H\$ (512) = "0200"

FN H\$ (-1024) = "FC00"

Возможность работы с отрицательными числами должна быть очень полезной для тех, кто программирует в машинных кодах для выполнения относительных переходов.

Если Вы хотите просмотреть содержимое памяти в шестнадцатиричных кодах, Вы можете воспользоваться:

```
100 INPUT "Start address?"; addr
110 PRINT FN H$ (addr); " ";
    FN H$ (PEEK addr)
120 LET addr = addr + 1:GOTO 110
```

Если Вы хотите задать стартовый адрес в шестнадцатиричной форме, можете изменить строку 100

```
100 INPUT "Start address?":A$: LET addr = FN D(A$)
```

4. Функция FN I

Структура: FN I (старт, строка A, строка B).

Функция FN I проверяет строку A, начиная с позиции "старт" в поисках строки B. В других версиях Бейсика эту функцию называют INSTRING. Если строка найдена, то выдается позиция в строке A первого символа строки B, иначе выдается 0.

Строка A может быть любой длины, но строка B должна иметь не более 255 символов, иначе пройдет сообщение "invalid argument".

Если "старт" равен нулю, то Вы получите сообщение "Subscript out of range".

Если строка B длиннее, чем A, то результат - 0, то же - если "старт" больше, чем длина A\$ или если обе строки имеют длину, равную нулю.

Вы можете заменить некоторые символы в искомой строке на знак #, что означает "все равно".

Например:

```
PRINT FN I (1,A$, "SM # TH")
```

отыщет положение в A\$ таких строк как "SMYTH", "SMATH" и т.д.

Единственный случай, когда символ # выступает сам за себя, т.е. рассматривается

буквально - это когда он стоит первым в искомой строке.

Возможность задания "старта" для поиска может быть полезной, если Вы предполагаете, что искомая строка содержится не один раз. В приведенном примере в A\$ разыскивается каждое употребление "TEST".

```
100 DIM A$(1000)
110 FOR n=1 TO RND*10+3
120 LET pos = RND*995
130 LET A$(pos TO pos+3)="TEST"
140 NEXT n
150 PRINT "TESTs hidden in A$ - press any key to find them"
    (в A$ спрятаны символьные строки TEST. Нажмите любую клавишу, чтобы их отыскать).
160 PAUSE 0
170 LET loc = 1
180 LET loc + FN I (loc, A$, "TEST")
190 IF loc <> 0 THEN FRINT loc: LET loc=loc+1 : GO TO 180
200 PRINT "Finish"
```

Строка A\$ просматривается, начиная с 1-ой позиции (loc = 1), а затем с каждой позиции, в которой была найдена строка "TEST". Когда FN I выдает 0, все появления "TEST" были найдены.

FN I может применяться для проверки вводимых строк в обучающих или игровых программах. Представьте, например, что "Спектрум" задал Вам вопрос, правильный ответ на который "NAPOLEON". Те, кто введут "NAPOLEON " (обратите внимание на пробел в конце слова) или "NAPOLEON BONAPARTE", получат ответ, что они неправы и будут законно разочарованы такой недружественностью со стороны программиста, написавшего программу. А это происходит, когда сравнение того, что введено и того, что должно быть, проводится простейшим способом. Вы можете обеспечить проверку на содержание требуемой строки во вводимой.

```
INPUT A$: IF FN I (1,A$,C$)<>0 THEN PRINT "CORRECT"
```

Другое применение этой функции может быть найдено в "разупаковке" одной длинной строки, состоящей из многих символьных строк различной длины. Один из возможных приемов состоит в том, что резервируется определенный набор символов (например CHR\$ 1...CHR\$31) для выполнения функций "маркера". CHR\$ 1 отмечает начало первой подстроки из длинной строки, CHR\$ 2 - начало второго и т.д. Тогда:

```
PRINT A$ (FN I(1,A$,CHR$ n) + 1 TO FN I (1,A$, CHR$ (n+1))-1)
```

выдает n-ую строку из A\$.

Здесь возможны и многие другие приемы, но основное преимущество состоит в том, что основная строка будет быстро просмотрена на предмет входа в нее составных подстрок, причем сильно экономится память, если входящие подстроки имеют разную длину.

5. Функция: FN M

Структура: FN M ()

Эта функция выдает объем свободной памяти. В скобках ничего ставить не надо. Попробуйте:

```
PRINT FN M( ): DIM A$ (100): PRINT FN M( )
```

Это очень простая функция, которая состоит в основном из обращения в ROM. При отсутствии Бета-Бейсика можете применять:

```
PRINT 65535 - USR 7962
```

6. Функция: FN N

Структура: FN N (строинг)

См. также FN C\$ (число)

Конвертирует двухсимвольную строку в целое число от 0 до 65535. Эквивалент:

```
LET number = 256*CODE C$(1)+CODE C$(2)
```

Если в строке не два символа, то получите сообщение:

```
"invalid argument" ("неправильный аргумент")
```

7. Функция FN P

Структура: FN P (адрес)

FN P - это двойной PEEK - указанного адреса и следующего за ним байта. Эквивалент:
LET value = PEEK (address) + 256*PEEK (address+1)

Заметьте, что младший байт идет первый, как это принято в машинном кодировании и в таблице системных переменных.

DPOKE позволяет делать двойной POKE так же, как FN P - двойной PEEK.

8. Функция: FN S\$

Структура: FN S\$ (число, строка).

В других версиях Бейсика эту функцию часто называют "STRING\$". Ее результат - это количество повторений символьной строки.

FN S\$ (32,"-") =32 знака минус

FN S\$ (4,"AB") ="ABABABAB"

PRINT FN S\$(704,"X") = целый экран знаков "X"

PRINT FN S\$ (3,"A"+CHR\$13)= A

A

A

FN S\$ работает быстрее, чем цикл FOR-NEXT и занимает меньше места, чем прямой ввод строки, если строка длиннее, чем 14 символов.

9. Функция FN T\$ ()

См. также CLOCK

Эта функция выдает текущее время. Если часы не запущены, то: FN T\$ () ="00:00:00"

Если часы были запущены, то независимо от того, выводится ли их показание на дисплей, FN T\$ будет выдавать постоянно меняющееся значение:

```
100 CLOCK 1
```

```
110 LET N$ = FN T$( ): PRINT N$
```

```
120 PRINT "Hours =" ;N$(1 TO 2); "Mins=" ;N$ (4 TO 5)
```

```
130 GO TO 110
```

Обычно неплохо передавать результат FN T\$ переменной, чтобы "замораживать" его в какой-то момент.

10. Функция: FN U\$

Структура: FN U\$ (форматирующая строка, число)

См. также USING

Выдает строку, эквивалентную "числу", отформатированную так, как указывает "форматирующая строка". Количество ведущих нулей или пробелов может быть задано.

USING выполняет аналогичное действие, но применяется только с оператором PRINT, в то время как FN U\$ можно использовать с любой командой, допускающей работу со строками. Для более подробного объяснения см. USING.

Приложение А

Набор символов.

Здесь представлено дополнение к набору символов стандартного "Спектрума", вводимых программой BETA-BASIC.

Код	Клавиша	Символ
128	8	KEYWORDS
129	1	DEF PROC
130	2	PROC
131	3	END PROC
132	4	RENUM
133	5	EDIT
134	6	AUTO
135	7	DELETE

*136-143

144	A	ALTER
*145	B	B
146	C	CLOCK
147	D	DO
148	E	ELSE
*149	F	F
150	G	GET
*151	H	H
152	I	EXIT IF
153	Y	WHILE
154	K	UNTIL
155	L	LOOP
156	M	SORT
157	N	ON ERROR
158	O	ON
159	P	DPOKE
160	Q	POP
161	R	ROLL
162	S	SCROLL
163	T	TRACE
164	U	USING

* - осталось без изменений.

Приложение Б

Сообщения.

Ниже перечислены сообщения, которых нет в стандартной версии Бейсика для "Спектрума". Исключение - G, но оно имеет другой смысл.

G: NO ROOM FOR LINE

Перенумерация строк программы так, как это было заказано, приводит к тому, что одна или несколько новых строк попадают в блок, не подлежащий перенумерации или строка в результате перенумерации приобретает номер больше 9999.

S: MISSING LOOP

Оператор EXIT IF или оператор цикла DO с условием (UNTIL или WHILE) не содержит оператора конца цикла LOOP.

T: LOOP WITHOUT DO

Оператор конца цикла LOOP не имеет соответствующего открывающего DO.

U: NO SUCH LINE

Оператор DELETE использован с номером строки, которой нет в программе.

V: NO POP DATA

Была попытка снять данные с программного стека, обслуживающего операторы GO SUB, DO LOOP или PROC, в то время как стек пуст, т.е. операции GO SUB, DO LOOP и PROC еще не встречались в программе.

W: MISSING DEF PROC

Была вызвана незаданная процедура PROC или встретился оператор END PROC без соответствующего DEF PROC.

X: NO END PROC

При попытке "перепрыгнуть" через DEF PROC (не во время вычисления процедуры PROC) программа не нашла соответствующего END PROC.

Y: TOO HARD

При перенумерации строк в программе было найдено указание на номер строки, записанный в виде вычисляемого выражения.

(Продолжение следует)

Секреты ПЗУ

Подпрограмма INITIALIZATION.

Здесь располагается пакет процедур, запускаемых на исполнение после включения питания компьютера.

11B7-11CA - NEW

Выполняются необходимые подготовительные операции для работы команды NEW. Регистры процессора выставляются в соответствии со значениями системных переменных RAMTOP, RASP, PIP, PRAMT, UDG. В аккумуляторе устанавливается FF.

Установленные значения запоминаются в регистрах путем перехода к альтернативному набору.

11CB-11D9 - START/NEW

Это точка входа. Сюда Вы попадаете после включения питания компьютера. В этом случае в аккумуляторе содержится 0, а в паре DE - FFFF. Если же Вы попали сюда после исполнения команды NEW, то в аккумуляторе - FF, а в паре DE - значение (RAMTOP).

По порту FE выдается байт 7, что означает - бордюр белый. Инициализируется регистр I.

11DA-11DC RAM-CHECK

Отсюда начинается проверка ОЗУ.

11DC-11E1 RAM-FILL

Сначала в каждый байт ОЗУ засылается число 02. Засылка этого числа идет сверху вниз. Если первое включение, то с адреса FFFF до адреса 3FFF, а если по команде NEW, то с адреса (RAMTOP) до 3FFF.

11E2-11EE - RAM READ

Проверяется действительно ли в ячейках ОЗУ содержится число 02.

Проверка идет снизу вверх. Сначала в нижней ячейке содержимое уменьшается на 1. Если достигнут 0, значит память неисправна и следует переход на 11EF (RAM-DONE)- при этом регистр HL, содержащий адрес проверяемой ячейки становится как бы указателем верхнего физического предела памяти.

Вновь содержимое уменьшается на единицу. Должен быть 0. Если это не так - ошибка и прямой проход на RAM-DONE. HL указывает на вершину работоспособной памяти. Если все в порядке, переход в начало RAM-READ для проверки следующей ячейки. Когда все ячейки проверены - тоже переход на RAM-DONE.

11EF-1218 RAM-DONE

Из альтернативного набора в регистрах процессора восстанавливаются запомненные там значения системных переменных. Это важно при выполнении команды NEW, а для первого включения (системного рестарта) - безразлично.

Если идет инициализация после NEW выполняется переход на 1219 (RAM-SET), в противном случае работа продолжается.

Значение P-RAMT выставляется по содержимому HL (по результатам проверки).

В паре DE выставляется адрес 3EAF, который относится к таблице знакогенератора и соответствует концу шаблона буквы "U" (21-ая буква).

В паре BC выставляется число 00A8, что соответствует размеру памяти, отводимой под 21 символ UDG (каждому по 8 байтов).

Изображения двадцати одного символа копируются из знакогенератора ПЗУ в область графики пользователя UDG. это их первоначальная установка. Далее пользователь

сможет задавать их такими, какими ему надо.

1219-12A1 - RAM-SET

Эта весьма объемная процедура является общей для первого включения компьютера и для исполнения команды NEW.

Выставляется значение RAMTOP.

В системной переменной CHARS, от которой зависит, в каких адресах памяти расположен знакогенератор, выставляется исходное значение 3C00.

В ячейке, на которую указывает RAMTOP, устанавливается значение 3E, а адрес нижележащей ячейки засылается в регистровую пару SP. Тем самым на вершине памяти организован машинный стек.

Адрес ячейки, находящейся еще на два байта ниже, засылается в системную переменную ERR-SP. Теперь в этих ячейках можно размещать адрес процедуры, к которой должен происходить переход при ошибке.

Инициализация продолжается:

1. Включается режим обработки прерываний IM1.

2. В индексный регистр IX засылается адрес 5C3A. Он служит как базовый для работы с таблицей системных переменных. Пока Вы работаете в БЕЙСИКе, это значение хранится там всегда, его менять нельзя.

3. В системную переменную CHANS засылается адрес 5CB6, используемый как базовый адрес таблицы, в которой содержится информация о каналах.

4. Начальная информация о каналах (15 байтов) пересылается из таблицы, находящейся в ПЗУ по адресу 15AF в область, на которую указывает установленное значение CHANS.

5. Системная переменная DATADD устанавливается так, чтобы указывать на последний байт таблицы информации о каналах.

6. Следующий за ним адрес используется в качестве исходного значения системных переменных PROG и VARS.

7. В него засылается число 80H, которое является меткой конца области переменных (VARS).

8. Следующий за ним адрес используется для установки системной переменной E-LINE.

9. В него засылается число 0D, т.е. в исходном состоянии строка редактирования содержит только байт "возврат каретки" (ENTER).

10. В следующий адрес засылается байт 80H, который является меткой конца области редактирования.

11. Адрес следующего байта используется для установки конца рабочей области БЕЙСИКа (системной переменной WORKSP) и для установки основания и вершины стека встроенного калькулятора (системных переменных STKBOT и STKEND, которые пока совпадают).

12. В системные переменные, отвечающие за установку цветовых атрибутов ATTR-P, ATTR-T, BORDER, засылается байт 38H, что означает:

FLASH 0; BRIGHT 0;
PAPER 7; INK 0.

13. Инициализируются системные переменные REPDEL и REPPER.

14. В системных переменных KSTATE-0 и KSTATE-4 устанавливается значение FF.

15. Исходная информация о потоках (14 байтов) пересылается из таблицы, находящейся в ПЗУ по адресу 15C6 в область системных переменных (начиная с адреса STRMS).

16. Вызовом CLEAR-PRB (0EDF) очищается буфер принтера.

17. В системной переменной DF-SZ устанавливается число 02, что означает "две строки в системном окне".

16. Вызовом CLS (0D6B) выполняется очистка экрана.

19. Выполняется печать системного сообщения (с) 1982 Sinclair Research Ltd, для чего в паре DE выставляется адрес этого сообщения - 1538 и вызывается процедура PO-MSG (0C0A).

20. Для продолжения работы выполняется безусловный переход на процедуру MAIN-1 (12A9).

Подпрограмма "MAIN EXECUTION LOOP".

Этот пакет процедур управляет режимом редактирования, исполнением прямых команд и генерацией системных сообщений.

12A2 - 12A8 MAIN-EXEC

В нижней части экрана устанавливаются 2 строки и вызовом AUTO-LIST (1795) включается режим автоматического листинга.

12A9-12AB - MAIN-1

Вызовом SET-MIN (16B0) устанавливается минимальный размер всех областей, находящихся выше E-LINE.

12AC-12CE - MAIN-2

Вызовом CHAN-OPEN (1601) при содержимом аккумулятора, равном нулю, открывается канал "K".

Вызовом EDITOR (0F2C) включается режим редактирования для создания БЕЙСИК-строки.

Вызовом LINE-SCAN (1B17) текущая строка проверяется на правильность синтаксиса. Проверкой седьмого бита системной переменной ERR-NR устанавливается правильность синтаксиса. Если он включен - все правильно и происходит переход на MAIN-2 (12CF).

Если есть ошибка, устанавливается с каким каналом идет работа (по 4-му биту системной переменной FLAGS2). Если это не канал "K" - переход на MAIN-4 (1303).

Если это канал "K", вызовом REMOVE-FP (11A7) удаляются все числа, записанные в интегральной форме, в системной переменной ERR-NR устанавливается FF и выполняется возврат к началу MAIN-2 (12AC).

12CF-1302 - MAIN-3

Проверка строки на синтаксис прошла успешно.

В системную переменную CH-ADD устанавливается адрес начала области редактирования. Вызовом E-LINE-NO (19FB) выясняется номер этой строки. Если это полноценное число - переход на MAIN-ADD (155D). В противном случае вызовом RST 0018 вводится первый байт этой строки. Если это 0D ("возврат каретки") - вновь переход на MAIN-EXEC (12A2). В противном случае речь идет о выполнении прямой команды.

Если необходимо, вызовом CL-ALL (0DAF) очищается область экрана и вызовом CLS-LOWER (0D6E) - очищается нижняя часть экрана. Устанавливается необходимый режим скроллинга и вызовом LINE-RUN (1B6A) запускается исполнение строки.

После окончания интерпретации и исполнения строки выполняется возврат на 1303 (MAIN-4).

1303-1312 - MAIN-4

Отсюда начинается подготовка к печати сообщения по результатам работы строки или, если ранее была выявлена ошибка в синтаксисе, по результатам ее обработки.

Включаются маскируемые прерывания, подготавливается прием нажатой клавиши, очищается буфер принтера, в аккумулятор засылается содержимое ERR-NR плюс единица.

1313-133B - MAIN-G

Обнуляются системные переменные FLAGX, DEFADD, старший байт X-PTR.

Поток 0 подключается к каналу "K".

Вызовом SET-MIN (16B0) очищаются рабочая область и стек калькулятора.

Вызовом CLS-LOWER (0D6E) очищается нижняя часть экрана.

Проверяется код сообщения, подлежащего печати. Если он от 0 до 9 - переход на MAIN-5 (133C).

Если он больше 9, то к нему прибавляется число 7, чтобы после цифр 0...9 следовали буквы A...R. (Код цифры 9 равен 57. Следующий за ним код 58 не означает букву "а", а соответствует двоеточию ":". Код же буквы "А" равен 65, отсюда и возникает смещение на 7 единиц).

133C-1372 - MAIN-6

Вызовом OUT-CODE (15EF) печатается код сообщения. Вызовом RST 0010 вслед за ним печатается пробел.

В регистровой паре DE устанавливается адрес 1391 - базовый адрес таблицы, содержащей сообщения БЕЙСИКа.

Вызовом PO-MSG (0C0A) с кодом сообщения, установленным в аккумуляторе, печатается необходимое сообщение.

Следом за ним печатается номер строки (вызовом OUT-NUM1 – 1A1B), затем знак ":" (вызовом RST 0010) и наконец номер оператора в строке (вызовом OUT-NUM1 - 1A1B).

Вызовом CLEAR-SP (1097) очищается область редактирования.

Если программа завершилась успешно (в ERR-NR по-прежнему содержится FF) выполняется переход на MAIN-9 (1386).

Проверяется код сообщения. Если он равен 9 (STOP statement) или 15 (BREAK into program), то продолжение работы должно идти со следующего оператора и выполняется переход на MAIN-6 (1373). В прочих случаях - с того же места - переход на MAIN-7 (1376).

1373-1375 - MAIN-6

Здесь увеличивается на единицу системная переменная SUBPPC (номер оператора в строке).

1376 - 1383 - MAIN-7

Подготавливается перенос содержимого из системных переменных NEWPPC и NSPPC в системные переменные OLDPPC и OSPPC.

1384-1385 - MAIN-8

Выполняется этот перенос.

1366-1390 - MAIN-9

Заключительные операции.

Тексты сообщений.

В каждом сообщении последний символ - инвертирован, т.е. код символа увеличен на 80H, что служит маркером, разделяющим сообщения друг от друга.

1391	80H	- маркер.
1392	Report 0	- "O.K."
1394	Report 1	- "NEXT without FOR"
13A4	Report 2	- "Variable not found"
13B6	Report 3	- "Subscript wrong"
13C6	Report 4	- "Out of memory"
13D2	Report 5	- "Out of screen"
13DF	Report 6	- "Number too big"
13ED	Report 7	- "RETURN without GOSUB"
1401	Report 8	- "End of file"
140C	Report 9	- "STOP statement"
141A	Report A	- "Invalid argument"

142A	Report B	- "Integer out of range"
143E	Report C	- "Nonsense in Basic"
144F	Report D	- "BREAK - CONT Repeats"
1463	Report E	- "Out of DATA"
147F	Report G	- "No room for line"
148F	Report H	- "STOP in INPUT"
149C	Report I	- "FOR without NEXT"
14AC	Report J	- "Invalid I/O device"
14BE	Report K	- "Invalid color"
14CC	Report L	- "BREAK into program"
14DE	Report M	- "RAMTOP no good"
14EC	Report N	- "Statement lost"
14FA	Report O	- "Invalid stream"
1508	Report P	- "FN without DEF"
1516	Report Q	- "Parameter error"
1526	Report R	- "Tape loading error"

В этой же таблице есть еще два "сообщения":

1537	", "	- запятая с пробелом
1539	"(c) 1982 Sinclair Research Ltd"	- исходное системное сообщение.

1555 - 155C - REPORT-G

Процедура служит для печати сообщения "No room for line". Печать выполняется переходом на процедуру MAIN-G (1313) с установленным в аккумуляторе кодом 10H.

Подпрограмма "MAIN-ADD". Этот пакет процедур служит для "пристегивания" новой БЕЙСИК-строки к строкам существующей БЕЙСИК-программы. Если новая строка имеет номер, который уже есть у строки БЕЙСИК-программы, то происходит замена последней. Если новая строка не имеет номера, то ее ввод в программу не производится.

155D - 157C - MAIN-ADD

Номер новой строки делается текущим засылкой его в E-PPC, на стеке запоминается адрес процедуры обработки возможной ошибки REPORT-G (1555), определяется длина вводимой строки, вызовом LINE-ADDR (196E) проверяется нет ли строки с таким же номером и если нет, то переход на MAIN-ADD1 (157D).

Если уже есть, то ее длина определяется вызовом NEXT-ONE (19B8) и она удаляется вызовом RECLAIM-2 (19E8).

157D-15AA - MAIN-ADD1

Если новая строка содержит только номер и "возврат каретки" - переход на MAIN-ADD2 (15AB).

К длине новой строки прибавляются 4 байта (2 для номера строки и 2 для ее длины). Вызовом MAKE-ROOM (1655) под нее резервируется место и выполняется копирование. В E-PPC вводится номер этой строки и в саму строку вводятся два байта с ее номером и два байта с ее длиной.

15AB-15AE - MAIN-ADD2

Если строка пустая, отсюда делается безусловный переход на MAIN-EXEC (12A2) для включения автоматического листинга.

Раздел исходной информации о каналах.

Первоначально компьютер имеет четыре канала:

- "K" - связь с клавиатурой;
- "S" - связь с дисплеем;
- "R" - связь программной области с областью редактирования (внутренний канал);

- "P" - связь с принтером.

В нижележащей таблице содержится следующая информация по каждому из каналов:

- адрес процедуры, обрабатывающей вывод;

- адрес процедуры, обрабатывающей ввод;

- наименование канала (одна литера).

Адрес	Содерж.	Наименование
15AF	F4 09	PRINT-OUT (09F4)
	A8 10	KEY-INPUT (10A8)
	48	"K"
15B4	F4 09	PRINT-OUT (09F4)
	C4 15	REPORT-J (15C4)
	53	"S"
15B9	81 0F	ADD-CHAR (0F81)
	C4 15	REPORT-J (15C4)
	52	"R"
15BE	F4 09	PRINT-OUT (09F4)
	C4 15	REPORT-J (15C4)
	50	"P"
15C3	80	маркер конца таблицы.

15C4-15C5 - REPORT-J

Вызывает RST 0008 с кодом перехвата 12H для генерации сообщения "Invalid I/O device".

Исходная информация о потоках.

"Спектрум" в исходном состоянии имеет семь задействованных потоков, которые ведут к четырем вышеперечисленным каналам. Номера этих потоков - от FD до 03.

Адрес	Данные	Комментарий.
15C6	01 00	Поток FD - подключен к каналу "K"
15C8	06 00	Поток FE (канал "S")
15CA	0B 00	Поток FF (канал "R")
15CC	01 00	Поток 00 (канал "K")
15CE	01 00	Поток 01 (канал "K")
15D0	06 00	Поток 02 (канал "S")
15D2	10 00	Поток 03 (канал "P")

Примечание ИНФОРКОМа:

Как мы уже упоминали, в начале следующего года мы дадим статью, подробно разбирающую концепцию потоков и каналов в "Спектруме" и откроем перспективы использования их в Ваших программах.

Подпрограмма WAIT-KEY.

Этот блок процедур осуществляет управление процедурами ввода информации.

15D4-15DD - WAIT-KEY

Подготовительные операции, связанные с проверкой необходимости очистки нижней части экрана.

15DE-15E3 - WAIT-KEY1

Вызывается процедура подготовки ввода через INPUT-AD (15E6). Если положенный код принят - включается флаг переноса (CARRY) и по нему выполняется возврат. Если никакая клавиша не нажата, то и флаг CARRY и флаг ZERO равны нулю, в этом случае повтор исполнения WAIT-KEY1, в прочих случаях произошла ошибка и прямой проход на REPORT-8.

15E4-15E5 - REPORT-8

Вызывает RST 0008 с кодом перехвата 07H для генерации сообщения "End of file".

15E6-15EE - INPUT-AD

Подготовительные операции перед безусловным переходом на процедуру CALL-SUB (15F7).

В регистре HL устанавливается базовый адрес таблицы информации о каналах.

Подпрограмма "MAIN PRINTING"

15EF-15F1 - OUT-CODE

Подпрограмма переводит целое число от 0 до 9 в значение его кода ASCII.

15F2-15F6 - PRINT-A-2

Подготовительная операция. В регистре HL устанавливается базовый адрес таблицы информации о каналах.

15F7-1600 CALL-SUB

По заданному каналу для ввода или вывода информации вызовом CALL-JUMP (162C) выполняется установка соответствующих данному каналу флаговых системных переменных. Далее возврат в вызывающую процедуру.

Подпрограмма "CHAN-OPEN".

Подпрограмма служит для открывания канала ввода/вывода. При входе в нее в регистре A должен содержаться номер задействованного потока - от FD до 03. В зависимости от данных по потокам тот или иной канал делается текущим.

1601-160D - CHAN-OPEN

Базовым адресом для потока 00 является 5C16 (23574). Содержимое аккумулятора удваивается и прибавляется к этому адресу. Полученный адрес указывает на место расположения данных по данному потоку. Вводятся оба байта. Если они оба равны нулю - переход на REPORT-0. Если все в порядке – переход на CHAN-OP-1 (1610).

160E-160F - REPORT-0

Вызывает RST 0008 с кодом перехвата 17H для генерации сообщения "Invalid stream".

1610-1614 - CHAN-OP-1

Данные по потоку уменьшаются на единицу и прибавляются к базовому адресу области информации о каналах. В результате суммирования получается адрес, по которому можно найти адрес процедуры, занимающейся обработкой данного канала ввода/вывода.

Подпрограмма CHAN-FLAG.

Процедуры этой подпрограммы выставляют флаговые системные переменные в соответствии с тем, какой канал задействован.

1615-162B - CHAN-FLAG

Для каждого из стандартных трех внешних каналов есть для этого своя процедура. Обратиться к ней можно, вычислив ее адрес с помощью специальной таблицы (16ED). Здесь вводится базовый адрес этой таблицы - 163D и вызовом процедуры INDEXER (16DC) из нее выбирается величина смещения, соответствующая текущему каналу. Если такого канала в таблице нет - возврат. Далее величина смещения прибавляется к базовому адресу и получается адрес необходимой процедуры - это либо CHAN-K, либо CHAN-S, либо CHAN-P.

162C- CALL-JUMP

Безусловный переход по вычисленному адресу.

162D-1633 - таблица " смещений" для перехода к процедурам установки флагов.

1634-1641 - CHAN-K

Здесь выставляются флаги системных переменных TV-FLAG, FLAGS, FLAGS2 так, как это нужно для работы на ввод от клавиатуры и на вывод в нижнюю часть экрана дисплея. Работа завершается переходом на CHANS-S-1.

1642-1645 - CHAN-S

1646-164C - CHAN-S-1

Здесь выставляются флаги системных переменных TV-FLAG и FLAGS так, как это нужно для работы на вывод в главную часть экрана дисплея. Работа завершается переходом на TEMPS (0D4D).

164D-1651 - CHAN-P

Здесь выставляется бит 1 системной переменной FLAGS так, как это нужно для работы на вывод на принтер. Работа завершается возвратом в вызывающую процедуру.

Подпрограмма "MAKE-ROOM".

Эта чрезвычайно важная подпрограмма вызывается очень часто. Ее задача - "раздвинуть" БЕЙСИК-код с тем, чтобы в образовавшееся место вставить новую строку. При вызове этой подпрограммы регистровая пара HL должна содержать адрес точки, следующей за той, после которой необходимо выделить место, а в регистровой паре BC содержится длина необходимого пространства.

1652-1654 ONE-SPACE

Это точка входа для того случая, когда надо выделить только один байт. Здесь в BC устанавливается значение 0001.

1655-1663 - MAKE-ROOM

Вызовом TEST-ROOM (1F05) проверяется имеет ли компьютер достаточно свободной памяти, чтобы выполнить такую операцию.

Вызовом POINTERS (1664) изменяются системные переменные, которые служат указателями на различные разделы БЕЙСИК-области памяти.

Выполняется перемещение вверх всей БЕЙСИК-информации, лежащей выше точки, в которую будет делаться вставка.

После выхода из этой подпрограммы регистры процессора имеют следующее содержание:

- в HL содержится адрес, предшествующий началу новой выделенной области;
- содержимое DE указывает на конец выделенной области.

Подпрограмма POINTERS.

При всех операциях с БЕЙСИК-строками, когда различные разделы БЕЙСИК-области то растягиваются, то сжимаются, за их положение отвечают соответствующие системные переменные, например VARS и др. (их всего 14). Необходимо своевременно изменять содержимое этих указателей.

В BC содержится длина блока, вовлеченного в манипуляции, а HL указывает на адрес, предшествующий тому, с которого начинаются перемещения.

1664-166A - POINTERS

Здесь устанавливается адрес VARS (5C4B), с которого и начинаются в области

системных переменных указатели, а в аккумуляторе выставляется 0EH (14 DEC) - их количество.

166B-167E - PTR-NEXT

Организуется цикл (14 проходов) для корректировки системных указателей. Если какой-либо из них не нуждается в переработке - переход на PTR-DONE (167F).

Последовательно изменяются: VARS, DEST, CHANS, CURCHL, PROG, NXTLIN, DATADD, E-LINE, K-CUR, CH-ADD, X-PTR, WORKSP, STKBOT, STKEND.

167F-168E - PTR-DONE

Если не все указатели еще обработаны, то отсюда выполняется переход в начало цикла PTR-NEXT (166B).

Подпрограмма "COLLECT A LINE NUMBER".

Процедура служит для выдачи номера строки. На входе пара HL содержит адрес, а на выходе номер строки содержится в паре DE. Если номер строки является недопустимым, то выдается нулевой номер.

168F-1690 - LINE-ZERO

Здесь стоят два нулевых байта - они нужны впоследствии для обнуления содержимого DE.

1691-1694 - LINE-NO-A

Подготовка к обнулению DE.

1695-169D - LINE-NO

Это обычная точка входа. Старший байт номера строки проверяется на допустимость и, если он не пригоден, переход на LINE-NO-A (1691). Далее возврат в вызывающую процедуру.

Подпрограмма RESERVE.

Эта подпрограмма служит для резервирования места в памяти компьютера между вершиной рабочего пространства (WORKSP) и основанием стека калькулятора (STKBOT). Используется процедурой RST 0030.

169E-16AF - RESERVE

В своей работе программа вызывает MAKE-ROOM (1655)

По окончании работы в регистровой паре DE содержится адрес первого байта вновь выделенной области, а пара HL указывает на последний байт.

Подпрограмма SET-MIN.

Эта подпрограмма устанавливает область редактирования и те области, которые следуют за ней, в минимальный размер. Фактически она выполняет очистку этих областей.

16B0-16BE - SET-MIN

Вводит данные из E-LINE, помещает в область редактирования только один байт - "возврат каретки" и следом за ним конечный маркер +80H. Следующий за ним адрес помещается в системную переменную WORKSP.

16BF-16C4 - SET-WORK

Если Вам надо очистить только рабочую область и стек встроенного калькулятора, то эта точка входа служит для этой цели.

16C5-16D3 - SET-STK

Это точка входа, если очистить надо только стек калькулятора.

Здесь же инициализируется системная переменная MEM (в ней устанавливается адрес 5C92).

По окончании следует возврат в вызывающую программу.

16D4 - 16DA - REC-EDIT

Процедура служит для удаления из памяти редактируемой строки. Это выполняется вызовом RECLAIM-1 (19E5) при установленном в регистровой паре DE адресе, который хранится в E-LINE.

Подпрограмма INDEXER.

Подпрограмма используется в некоторых случаях, когда надо провести выборку какого-либо значения из таблиц. Точкой входа является 16DC.

16D8-16DB - INDEXER-1

Переход к просмотру следующего адреса.

16DC-16E4 - INDEXER

Вводится первый байт. Если он равен нулю (конечный маркер) - возврат. Затем он сравнивается с искомым, содержащимся в регистре C. Если они не совпадают - переход на следующий адрес и переход на INDEXER-1 (16D8).

Подпрограмма "CLOSE #".

Процедуры, содержащиеся здесь, позволяют пользователю закрыть открытые потоки. Однако потоки 00, 01, 02 и 03 являются стандартными и постоянными - их закрыть нельзя.

16E5-16FB - CLOSE

Вызовом STR-DATA (171E) вводятся текущие данные по потоку.

Вызовом CLOSE-2 (1701) устанавливается код канала, к которому подключен текущий поток.

Если речь идет о потоках, открытых пользователем, 04...0F, выполняется переход на CLOSE-1 (16FC), а для стандартных потоков 00...03 вводятся исходные данные из таблицы 15C6.

16FC-1700 - CLOSE-1

Для пользовательских потоков здесь в качестве данных вводятся нули и выполняется возврат в вызывающую программу.

1701-1715 - CLOSE-2

Здесь устанавливается, к какому каналу подключен поток путем выбора из таблицы (1716). Выбор выполняется вызовом процедуры INDEXER (16DC). Из таблицы же извлекается и величина "смещения" для расчета адреса подпрограммы, закрывающей поток. В нашем случае для всех трех стандартных потоков этот адрес будет один и тот же - 171C (процедура CLOSE STREAM), но это не всегда так. Если Вами подключена периферия, обслуживаемая своим ПЗУ, замещающим (затеняющим) ПЗУ компьютера, то там для прочих потоков могут устанавливаться иные процедуры.

1716-171B - таблица смещений для расчета перехода на процедуру закрывания потока.

171C-171D - CLOSE-STR

Финишная процедура.

171E-1724 - STR-DATA

Снимая номер потока с вершины стека калькулятора, процедура возвращает в регистровой паре BC данные по этому потоку. Это выполняется вызовом STK-TO-A (1E94).

Принятый номер потока проверяется. Если он меньше 10H, следует переход на STK-DATA-1 (1727). Если же больше 0F, чего быть не может, - прямой проход на REPORT-0.

1735-1736 - REPORT-0

Вызывает RST 0008 с кодом перехвата 17H для генерации сообщения "Invalid stream".

1727-1735 - STK-DATA1.

По номеру потока, используя адрес в области системных переменных 5C10 в качестве базового, извлекаются текущие данные по данному потоку и помещаются в пару BC. Далее - возврат.

(Продолжение следует).

Аппаратное обеспечение

SECAM CODER

Сегодня, как и всегда, наш технический раздел ведут специалисты НТК "ПЛЮС".

Их работы стали настольной книгой многих любителей "Спектрума" и в первую очередь тех, кто не просто эксплуатирует компьютер в собственное удовольствие, а серьезно думает об его усовершенствовании.

Огромная потребность в их разработках до сих пор не позволяла удовлетворить всех желающих и внимательные читатели должны были отметить, что "ПЛЮС" до сих пор никогда не просил за свои работы предоплату, а отправлял их наложенным платежом - это мера защиты от переизбытка заказов.

Сегодня с любезного согласия НТК "ПЛЮС" мы публикуем информационное сообщение о работах НТК "ПЛЮС", а перед ним даем главу из их методической разработки N2 (MP-2), посвященную SEKAM-кодеру. Надо сказать, что эта глава составляет не более 10% этой разработки и мы надеемся, что ее публикация не подорвет интереса читателей к приобретению самой разработки.

Кодер SEKAM

Многие пользователи "Спектрумов" интересуются схемой формирования полного цветного ТВ сигнала, кодированного по системе SEKAM. Их привлекает возможность подключения к видеовходу или антенному входу ТВ без необходимости влезать в его внутренности.

Нами была опробована одна из распространенных схем SEKAM кодера, выполненного на микросхемах генераторов, управляемых напряжением, K531ГГ1. Эта схема приведена ниже. Следует отметить, что схема SEKAM кодера очень сложна в настройке, критична к разбросу параметров элементов. Высокого качества изображения добиться очень трудно, сильно влияние помех по всем цепям. Необходимо также иметь специальные приборы для настройки ТВ техники, например, осциллограф с блоком выбора строки и т.п. Кроме того, сама система SEKAM не обеспечивает высокого качества изображения от компьютера в силу своих особенностей. Например, ужасно выглядят горизонтальные цветные линии шириной в одну строку. В силу указанных причин SEKAM кодеры не имеют большой популярности среди пользователей Спектрума.

Прежде чем перейти к подробному описанию схем SEKAM кодера, немного вспомним об основных принципах формирования ПЦТС в системе SEKAM. Для получения полной гаммы цветов в телевидении используются три первичных света R,G,B. Названия происходят от аббревиатуры английских слов Red, Green, Blue - красный, зеленый, синий. При сложении трех первичных цветов с определенными весовыми коэффициентами, мы получаем белый цвет $Y = 0.3R + 0.59G + 0.11B$. Соответственно, при изменении весов первичных цветов, получаются все цвета спектра. Для того, чтобы не передавать избыточной информации по каналам ТВ вещания, вместе с импульсами синхронизации и яркостной составляющей, которые передаются и для ЧБ телевидения, передают дополнительно две цветоразностные составляющие:

$$R-Y = 0.7R - 0.59G - 0.11B$$

$$B-Y = -0.3R - 0.59G + 0.89B.$$

На приемной стороне в соответствии с этими соотношениями восстанавливаются исходные первичные цвета R, G, B.

Цветоразностные сигналы передаются с помощью двух частотно-модулированных (ЧМ) цветовых поднесущих. В каждой строке передается только одна поднесущая, т.е. информация об одном из цветоразностных сигналов R-Y или B-Y. Поскольку для восстановления первичных цветов R, G, B нужно иметь R-Y и B-Y одновременно, то в

телевизорах используют задержку цветоразностного сигнала предыдущей строки из 64 мкс (на длину строки) и, таким образом, восстанавливают первичные цвета. Для того, чтобы цвета в телевизоре не перепутались, после каждого кадрового синхронизирующего импульса в течение 9 строк передается импульсы цветовой синхронизации, которые и задают очередность следования цветоразностных сигналов.

Следовательно, задача SEKAM кодера - сформировать сигнал яркости, подмешать к нему синхросмесь, затем в каждую строку добавить цветовую поднесущую, промодулированную по частоте сигналом R-Y или B-Y поочередно и, кроме того, в начале каждого кадра передавать сигналы цветовой синхронизации. В этом случае телевизор сможет восстановить исходное цветное изображение.

Для передачи составляющих R-Y используется частота 4406.25 ± 2 кГц (частота покоя в "красной" строке), B-Y - 4250 ± 2 кГц (частота покоя в "синей" строке). В зависимости от передаваемого цвета меняется амплитуда цветоразностного сигнала, соответственно изменяется и частота поднесущей этого сигнала, т.е. происходит частотная модуляция. В таблице 1 приведены частоты поднесущих по "красной" и "синей" строкам для 8 основных цветов.

Таблица 1

Сигнал цветности				
Цвет полос	Красная строка		Синяя строка	
	Девияция кГц	Частота кГц	Девияция кГц	Частота кГц
Белый	0	4406	0	4250
Желтый	-46	4360	-230	4020
Голубой	280	4686	78	4328
Зеленый	234	4640	-152	4098
Пурпурный	-234	4172	152	4402
Красный	-280	4126	-76	4172
Синий	46	4452	230	4480
Черный	0	4406	0	4250

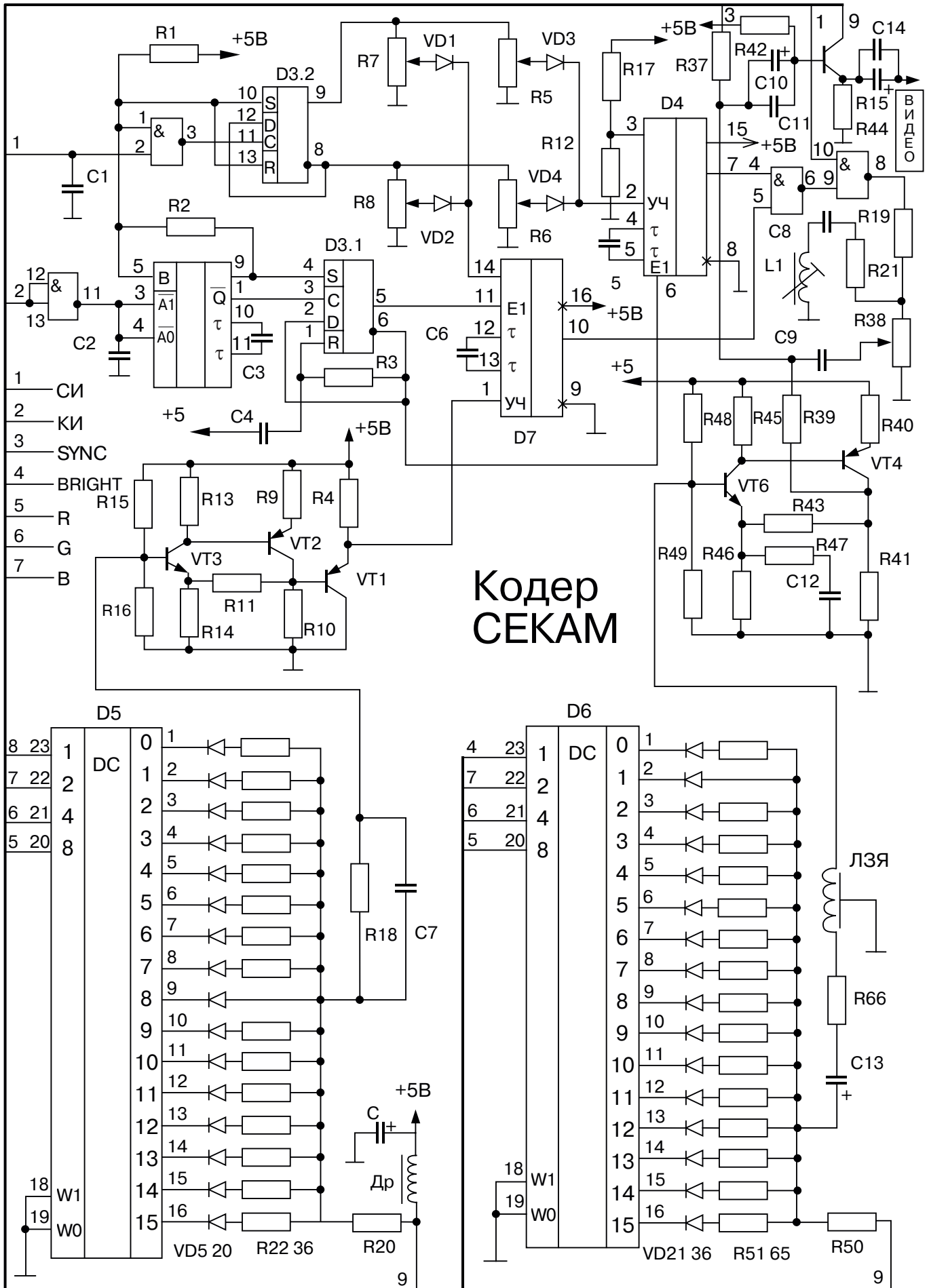
В заключение обзора системы SEKAM добавим, что составляющую R-Y принято передавать в инверсном виде, т.е. Y-R.

Теперь перейдем к схеме SEKAM кодера. Схема была разработана для работы с компьютером, поэтому формирование ПЦТС в ней значительно упрощено (без ущерба для работоспособности). В основном это упрощение касается формирования цветоразностных составляющих, сигнала яркости и сигналов цветовой синхронизации. Рассмотрим вначале схему формирования цветоразностных сигналов Y-R, B-Y.

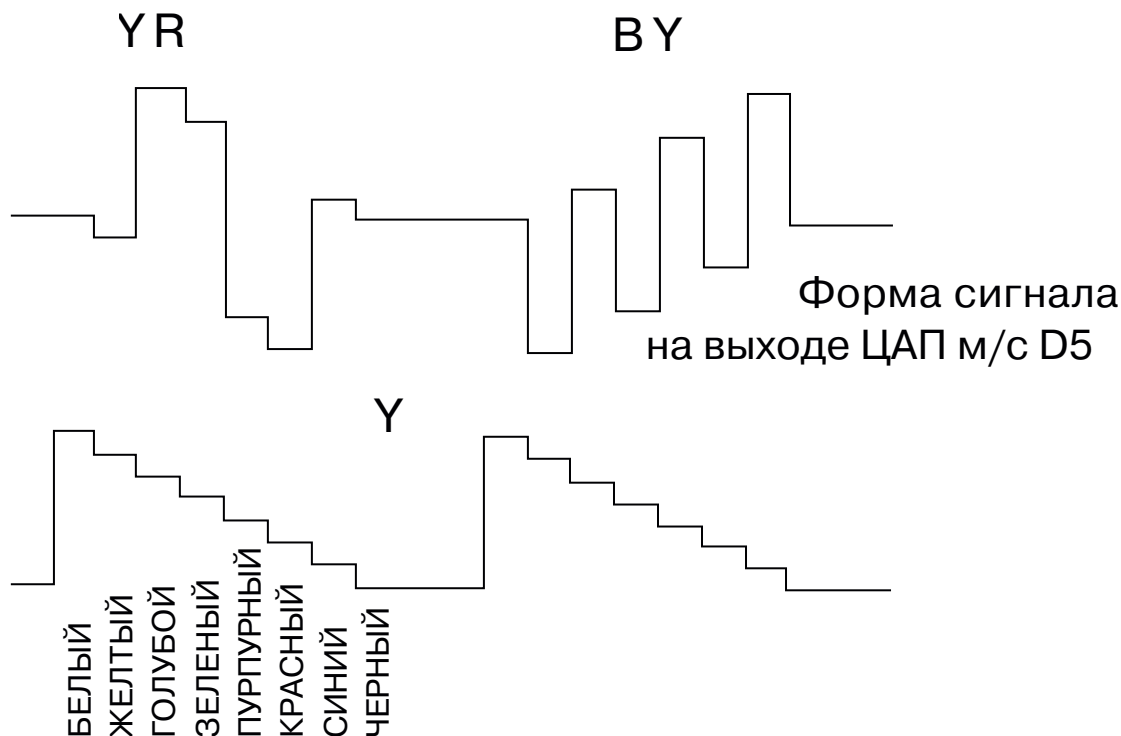
Поскольку от компьютера поступают дискретные сигналы R, G, B, то для формирования цветоразностных сигналов применена цифровая микросхема D5 - дешифратор 4 на 16 - 155ИДЗ, в качестве цифроаналогового преобразователя (ЦАП). В соответствии с принципами его работы, при одной из 16 комбинаций 0/1 на входе, на одном из 16 выходов появляется лог. "0" и подключается один из весовых резисторов. Резисторы должны быть подобраны очень точно для точного повторения формы сигналов Y-R, B-Y.

Сигналы первичных цветов R, G, B подключены к трем старшим разрядам дешифратора, на младший разряд подается сигнал строчной частоты, деленной пополам на триггере D3.2, поэтому четные резисторы подключаются во время одной строки, а нечетные во время другой. Соответственно, на выходе ЦАП поочередно появляются сигналы Y-R, B-Y.

Аналогичным образом устроен и ЦАП яркостной составляющей на микросхемах D6. На его разряды подаются сигнал управления яркостью BRIGHT, первичные цвета R, G, B и на выходе получаем сигнал яркости Y. Схема использовалась с компьютером, в котором сигнал BRIGHT в состоянии лог. "0" при команде с компьютера BRIGHT 1 и наоборот. Если в Вашем компьютере яркость выше при сигнале BRIGHT в лог. "0", то поменяйте местами резисторы R51 с R52, R53 с R54 и т.д., т.е. четные с нечетными до R65 включительно, или инвертируйте сигнал BRIGHT.



Сформированный сигнал яркости Y поступает далее на линию задержки яркости (1 мкс, т.к. сигналы цветности пройдут более долгий путь обработки), затем усиливается (VT6, VT4) и поступает на оконечный каскад смесителя (VT5). Формы сигналов на выходах ЦАП приведены на рис.



Сигналы цветности с выхода ЦАП также усиливаются в схеме на VT1-VT3 и поступают на вход управления частотой (УЧ) генератора управляемого напряжением (ГУН) на микросхемах D7-531ГГ1.

Эта микросхема кроме входа УЧ имеет также вход переключения диапазона (Д). К этому входу поочередно, через строку, подключаются резисторы R7, R8 (т.к. D3.2 работает как делитель строчной частоты на два). Эти резисторы настраиваются таким образом, чтобы в одной строке ГУН вырабатывал частоты "красной" строки (R8), а в другой - "синей" (R7) в соответствии с таблицей 1. Схема на одновибраторе D1 и триггере D13.1 вырабатывает импульс, начинающийся через 6 строк после окончания кадрового импульса КИ и заканчивающийся еще через 9 строк. Этот импульс с прямого выхода триггера запрещает работу ГУН D7 на время передачи пакета цветовой синхронизации и на то же время разрешает работу второго ГУН на D4, который и вырабатывает этот пакет. Этот второй ГУН работает под управлением того же D3.2 и вырабатывает частоты покоя "синей" (подстраивается R5) и "красной" (R5) строк.

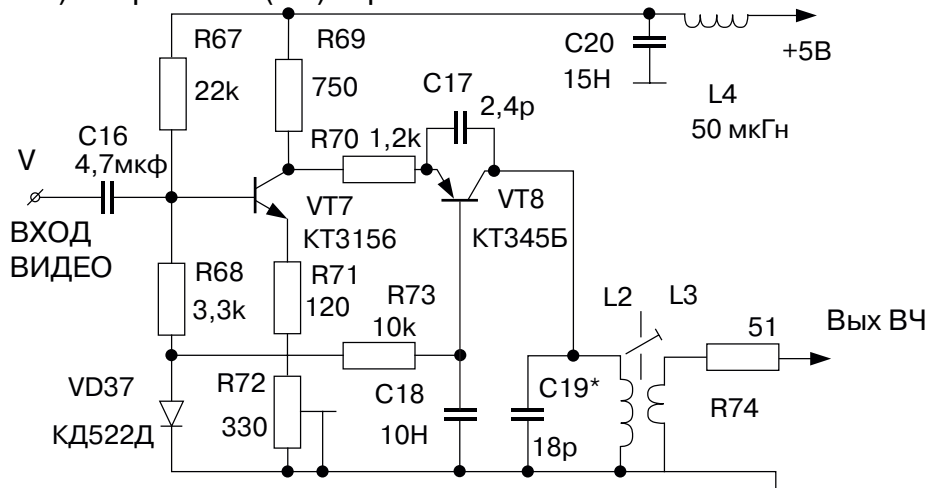


Схема ВЧ модулятора

Сигналы цветности с обоих ГУН собираются на микросхеме D2.2 и с выхода 8 D2.3 поступают через R19 на контур предискажений L1, C8, настроенный на частоту 4286 кГц и

далее через R38, C9 на вход смесителя на VT5. На VT5 происходит смешивание сигналов яркости, цветности и синхронизации и с выхода смесителя мы получаем ПЦТС системы СЕКАМ.

При монтаже и наладке схемы особое внимание следует обратить на разделение цепей питания как можно большего числа каскадов друг от друга. Не жалейте дросселей и блокировочных емкостей. Частоты ГУНов настраиваются с помощью частотомера с большой точностью. Схемы ЦАП конечно громоздки и сигналы Y-R, B-Y и Y можно было бы сформировать иначе, но преимуществом данной схемы их формирования является высокая стабильность и возможность особо точной настройки формы выходных аналоговых сигналов. Необходимость такой настройки продиктована также нелинейностью характеристики ГУН.

* * *

Всех желающих получить наши методические разработки мы приглашаем обратиться письменно по адресу:

1127566, Москва, И-566, НТК "ПЛЮС"

В последние месяцы "ПЛЮС" получал много писем от пользователей "Спектрумов" с жалобами на несвоевременное выполнение заказов и следует признать их справедливость. Действительно, заказов оказалось так много, что ни основного тиража, ни проведенных допечаток не хватило, чтобы удовлетворить все просьбы.

Сейчас "ПЛЮС" рад сообщить, что принятыми мерами он смог решительно исправить положение и в настоящее время исполняются последние из ранее поступивших заказов. Более того, учитывая, что спрос на подобную литературу очень и очень высок, был создан солидный задел и "ПЛЮС" готов в сжатые сроки удовлетворить новых клиентов.

Напоминаем содержание разработок НТК "ПЛЮС".

Методическая разработка (МР) N1.

"Интерфейсы персонального компьютера "Спектрум" "

Любители компьютерных игр найдут в ней несколько вариантов схем подключения джойстика. Наличие нескольких вариантов позволяет Вам выбрать наиболее приемлемый по наличию элементной базы.

Те, у кого есть принтер, смогут подключить его с помощью одной из схем последовательного или параллельного интерфейса принтера. Приведенные программы поддержки принтера (драйверы) позволят Вам распечатывать как текстовую, так и графическую информацию. Схемы интерфейсов и программы поддержки предназначены для работы с любыми принтерами, оборудованными одним из стандартных интерфейсов: CENTRONICS (ИРПР-М), ИРПР (IFSP), RS-232C (V.24).

Для занимающихся компьютерной графикой и живописью предназначена схема светового пера и распечатка работающего с ним графического редактора LP48K.

Схема программатора для ПЗУ с УФ стиранием позволит любителям технического творчества самостоятельно программировать такие микросхемы как 2716, 2764, 27128, 27256, 573 РФ2, 5, 4, 6, 8.

Объем МР1 - 34 страницы.

Со времени начала ее распространения (с сентября 1990г.) в адрес НТК не прекращается поток писем с положительными отзывами. Все схемы, входящие в разработку уже повторены многими пользователями, которые отмечают простоту изложения материала, доступность элементной базы и высокую повторяемость схем.

Методическая разработка (МР) N2.

Вторая методическая разработка МР2 состоит из двух частей. Первая часть посвящена контроллеру накопителя на гибких магнитных дисках НГМД. Контроллер полностью совместим с фирменной системой BETA-DISC INTERFACE. Он выполнен на базе микросхемы 1818ВГ93 (WD1793) и поддерживает работу до 4-х НГМД - 3-я или 5-ти дюймовых, односторонних и двусторонних, 40 и 80 дорожечных. Разработка содержит подробное описание схемы, сигналов, принципов работы интерфейса, рекомендации по сборке и наладке, перечень команд дисковой операционной системы TRDOS.

О преимуществах работы с НГМД по сравнению с работой с магнитофонной лентой знают все, это новый уровень работы с компьютером, и мы надеемся, что наша разработка поможет вам перейти на этот уровень.

Вторая часть МР 2 посвящена стыковке компьютера с бытовыми телевизорами. Наиболее качественное цветное изображение получается при стыковке компьютера по R-G-B входу телевизора. При стыковке по антенному входу с кодировкой по системе SEKAM или PAL происходит некоторая потеря качества цветного изображения, но зато телевизор не требует переоборудования. В некоторых случаях может удовлетворить стыковка через видеовход с чернобелыми или цветными телевизорами без передачи информации о цвете, т.е. в черно-белом изображении.

Вторая часть МР2 содержит схемы всех перечисленных способов стыковки компьютера с телевизорами, причем схемы предназначены не только для полупроводниковых, но и для ламповых телевизоров всех типов.

В разработке описаны способы подключения ко всем модификациям цветных телевизоров семейств:

УЛПЦТ-59/61, УЛПЦТ(И)-59/61, УПИМЦТ, 4УПИЦТ, 2УСЦТ, 3УСЦТ, 4УСЦТ, 4ПИЦТ-25-IV-1,2, 1УПЦТ-25, ПИЦТ-32, 3ПЦТ-32, 4УПЦТ-32-1,2, УПИЦТ-32-IV, ПИЦТ-32-10,

а также даны рекомендации для подключения ко всем моделям ч/б телевизоров.

Объем МР1 - 45 страниц.

Методическая разработка (МР) N3.

Третья методическая разработка (МР3) подготовлена по многочисленным письмам и пожеланиям пользователей "Спектрума". В нее входит принципиально новый способ русификации "Спектрума", выгодно отличающийся от известных способов.

Знакогенератор кириллицы размещается в неиспользуемой области ROM, там же размещены некоторые подпрограммы, которые добавляют в операционную систему "Спектрума" два новых регистра клавиатуры R и r к уже имеющимся K, L, C, E, G. Таким образом, в любой строке могут быть представлены буквы как русского, так и латинского алфавита одновременно, т.е. например, операторы Бейсика на английском языке, а текст на русском.

Также приведены дополнения к программам поддержки интерфейсов принтеров, опубликованных в МР1, для вывода на печать смешанного русско-латинского текста.

Кроме этого, в МР3 по просьбам пользователей включены схемы интерфейсов джойстиков, рассчитанных на двух игроков - INTERFACE 2 и SINCLAIR, а также схема программируемого джойстика, позволяющая запрограммировать все положения рукоятки и кнопки в соответствии с клавишами клавиатуры и более простой вариант - механически программируемый (путем перестановки перемычек) джойстик.

Все программы, входящие в разработки, приведены в виде листинга и блоков шестнадцатичных кодов с контрольными суммами. Свои заявки (желательно на открытках) направлять по адресу:

137566, Москва, И-566,
НТК "ПЛЮС"

Мы немедленно вышлем Вам бланк-заказ с указанием цены и порядка оплаты.

В заявке укажите полный адрес с почтовым индексом, фамилию, имя, отчество.

ZX-LPRINT III

Есть несколько неточностей в статье НТК "ПЛЮС", освещающей работу интерфейса ZX LPRINT-III.

Пугач Г.И. из г. Сумы спрашивает: куда выходят выводы 12 и 2 триггера ТМ-2.

"ПЛЮС" сообщает:

"По нашей вине не обозначены некоторые провода в жгутах. Так, от 10 и 1 ноги триггера DD4 в жгут идет провод N7 к RESET, а от 12 и 2 ног триггера DD4 провод N8 идет к A7".

Есть и ошибка по тексту. Так, на стр. 172 (3-я колонка), где описано подсоединение выводов DD1 напечатано:

"... не забудьте соединить вывод 21 с "землей" ".

Правильно должно быть:

"...не забудьте соединить вывод 21 с "+5V" ".

Тот же читатель сообщает о том, что единственный справочник, в котором ему удалось что-то разыскать об "экзотической" микросхеме K155ЛП8 - это справочник Якубовского и др. (1989 г.). В нем утверждается, что у вышеупомянутой ЛП8 выводы 8 и 9 вообще не используются, вопреки схеме интерфейса.

Ответ гласит:

"155ЛП8 обозначена верно, схема проверена, работает. В справочнике Якубовского приведена цоколевка (если присмотреться внимательнее к рис. 62 на стр. 87) для микросхемы 133ЛП8. Это планарная микросхема и у нее 15 выводов, а у 155ЛП8 - 14 выводов. Паяйте смело - пашет как зверь!"

Фомин В.А. из Челябинска имеет принтер CPF-136. Его интересует будет ли он работать с интерфейсом ZX-LPRINT-III (ведь он не входит в перечисленные). В своем принтере он не нашел графический режим и не нашел также больших букв русского алфавита.

Ответ:

Этот принтер скорее всего совпадает по кодам управления с EPSON-совместимыми принтерами. Возможно, что для этого Вам надо предварительно установить DIP-переключатели на режим EPSON (см. в инструкции к принтеру).

После изготовления интерфейса и опробования в текстовом режиме Вы можете путем перебора проверить все графические типы.

Что же касается больших букв русского алфавита, то раз есть малые, то должны быть и большие. Попробуйте распечатать знакогенератор по программе:

```
10 FOR a=32 TO 127
20 LPRINT CHR$ a; : NEXT a
30 FOR a=160 TO 255
40 LPRINT CHR a$; : NEXT a
```

Кстати, положение DIP-переключателей влияет и на выбранный шрифт. Имеет смысл поэкспериментировать с теми из них, которые за шрифт отвечают. Не забудьте после каждого изменения положения переключателей заново инициализировать принтер выключением и включением питания.

FORUM

Сообщения об ошибках.

DTR+

В номере 7/8 мы по просьбе нашего читателя публиковали программу декодера радиопередачи "DTR+" из журнала "БАЙТЕК".

В строке 330 допущена ошибка. Вместо числа 253 там должно быть 255. Ошибка вызвана непропечаткой в "БАЙТЕКе".

Некоторые читатели столкнулись с ней, а причину нашел А. Радомский из г. Львова, уже пользующийся приемом программ по радио и исследовавший этот вопрос.

По своему практическому опыту он рекомендует подавать сигнал одновременно на магнитофон (с уровнем +3... +6 dB) и на вход компьютера (при работе программы DTR+ в режиме LOAD).

Желательно использовать монофонический радиоприемник, т.к. у него более узкая полоса пропускания по ПЧ и уровень возможных помех по соседнему каналу ниже. Регулятором тембра необходимо уменьшить усиление по ВЧ. Сигнал на компьютер и на магнитофон подавать с выхода УНЧ (после регулятора тембра).

От имени наших читателей мы благодарим людей неравнодушных и готовых помочь и поделиться знаниями и опытом.

"Секреты ПЗУ" N4,5

В этом номере есть досадная опечатка: процедура выгрузки байтов SA-BYTES находится по адресу 04C2. У нас же ошибочно напечатано 04B2. Исправьте, пожалуйста в своих подшивках.

К вопросу о совместимости.

Интересное исследование для владельцев 128-килобайтных машин провел наш читатель из г. Фрязино Моск. обл. Губанов М.И. Он обратил внимание на то, что многие программы, особенно прошедшие через руки Билла Гильберта, работающие на Sp-48, отказываются работать на Sp-128 (имеется в виду одна из старых разработок, а не "Пентагон").

Симптомы следующие:

Нормально загружается и стартует первый блок (программа на БЕЙСИКе), читается следующий заголовок, но сообщение Bytes:... или аналогичное не появляется и больше вообще никаких процессов не происходит.

Если "маленькая" ПЗУ (интерфейс принтера) отсутствует или неправильно подключена, то в режиме 48K после чтения заголовка второго блока происходит все, что угодно - рестарт, мигание или произвольное окрашивание экрана и т.п. - короче сбой системы.

При внимательном рассмотрении оказалось, что такие программы содержат в себе нечто вроде POKE 23570,16. Как правило, эта команда записана в первом блоке (БЕЙСИКовском), но это может делаться и из машинных кодов.

Разобравшись, что она делает, товарищ Губанов установил, что она вносит изменения в таблицу данных о потоках STRMS.

Эта таблица определяет к какому каналу ввода/вывода подключен тот или иной поток. Здесь вам поможет наш раздел "Секреты ПЗУ". См. в данном номере таблицу исходных данных по потокам, находящуюся по адресу 15C6. Ячейка 23570 относится к потоку FE, а он исходно подключен к каналу "S" - главная часть экрана.

Таким образом, POKE 23570,16 это простейший способ блокировать появление на экране сообщений БЕЙСИКа путем переадресовки их на принтер, где они и должны успешно

Это изменение влияет таким образом, что весь выход, который первоначально должен был идти на экран, направляется на принтер. А именно: - на принтер отсылаются сообщения о найденных заголовках блоков программы - bytes: и т.п.

Как правило, этот оператор может быть безболезненно удален. Естественно, если он размещен в машинных кодах, сделать это будет посложнее. Проще всего забить соответствующее место пустой командой NOP (0).

Возможны несколько - как программных, так и аппаратных путей борьбы с этим злом.

1. Перед загрузкой такой программы включить принтер. Недостаток очевиден - не у всех он есть и к тому же надо помнить для каких программ это следует делать.

2. Доработать интерфейс принтера так, чтобы в случае выключенного или неподсоединенного принтера в компьютер поступал бит готовности, т.е. вместо принципа "если готов, то ноль" использовать принцип "если не готов, то единица". Достоинство способа в том, что сразу все такие программы начинают работать. Недостаток - надо иметь исправный интерфейс, а он есть не у всех.

3. Просто удалить оператор POKE... или его аналог. Недостаток - посреди красивой картинки будут возникать мерзкие надписи "Bytes:...". Их действие можно ослабить, спрятав их путем предварительной установки цветов INK и PAPER, но все равно это не то, что нужно.

4. И последний способ - поставить вместо зловерного оператора POKE 23739,111 (по принципу "клин клином вышибают?" - Инфорком.)

Здесь достигается тот же эффект, что и у Билла Гильберта, но несколько иным способом. В таблице информации о каналах, на которую указывает системная переменная CHANS изменяется адрес подпрограммы вывода символа на экран таким образом, чтобы он указывал на адрес другой "подпрограммы", состоящей из единственного оператора - RET. Таким образом, то, что в нормальных "человеческих" программах печатается на экране, а у Б. Гильберта отправляется на принтер, будет теперь отправляться в "никуда".

В исходном состоянии CHANS указывает на 23734, а на информацию по каждому каналу отводится 5 байтов. Таким образом, 23739 - ячейка, содержащая первый байт информации по второму каналу - "S". А именно - младший байт адреса процедуры, обеспечивающей вывод данного канала. Поместив туда число 111 (6FH), товарищ Губанов тем самым назначил в качестве процедуры, обслуживающей вывод на экран не C9F4 (PRINT-OUT), а 095F. Там действительно находится команда RET, в чем может убедиться всякий, читающий "Секреты ПЗУ".

Инфорком.

Этот способ потенциально опасен тем, что если программа использует стандартные каналы "Спектрума" (данные из таблицы CHANS) для вывода своих собственных сообщений, и при этом не проверяет эту таблицу и не устанавливает ее содержимое, то возможны неприятности. В таком случае можно в начале программы-загрузчика сохранять где-либо значение РЕЕК 23739, а после загрузки всех блоков программы с заголовками восстанавливать его.

И, в заключение об одной трудности. Загрузчики со следами Билла Гильберта достаточно трудно переделывать средствами БЕЙСИКа. Лучше воспользоваться программой COPY-COPY и с ее помощью внести изменения в числа. Но изменения надо вводить в число, записанное в интегральной (пятибайтной) форме. Можно заменить оператор POKE на REM. Тогда все, что стоит за ним, не будет иметь значения, хотя и в этом есть опасность, т.к. далее в этой строке могут быть необходимые для работы программы операторы. Можно заменить адрес 23570 на любой, меньший 16363 (адресовать POKE в ПЗУ - тогда он не пройдет), например на 0. Но это опять же надо делать с интегральной формой записи числа.

Естественно, может возникнуть вопрос: как же работают эти программы, требующие

принтера, на моделях 48K, к которым принтер не подключен или если у них даже нет интерфейса?

Все очень просто. Неподключенный принтер в моделях 48K вызывает игнорирование всех обращений к нему. То же самое можно сделать на SP-128 в режиме 48K, если совершенно отказаться от принтера в этом режиме и сделать соответствующие аппаратные изменения. Это наименее эстетичный из аппаратных методов.

ELITE

Как всегда, в нашей почте немало писем посвящено этой замечательной игре. По-видимому, это неисчерпаемая тема, которая еще долго будет привлекать к себе внимание пытливых исследователей.

Мы как смогли расклассифицировали проблемы, ставшие объектом изучения в этом месяце и вот что у нас получилось.

1. Планету RAXXLA не удалось обнаружить пока никому. Мы ничего не утверждаем - возможно такой планеты и нет. Мы просто знаем по зарубежным журналам, что искали ее во многих странах и очень активно. А сама эта планета, как мы уже сообщали, является главной целью героя захватывающей фантастической повести, которая прилагается к дистрибутивной кассете при покупке этой программы за рубежом.

Кстати, поиск планет многие ведут с помощью клавиши "R". Конечно, это возможно, но наверное те планеты, которые не включены в межгалактический справочник таким путем не отыскать, а RAXXLA тем и примечательна, что ее местоположение известно только пилотам ELITA, удаляющимся на покой. С этой планеты (судя по повести) возможны путешествия в любые точки пространства. И конечно ее нет ни в каких справочниках, а случайно забредшие к ней корабли уже назад не возвращаются. Свою тайну пилоты берегут свято.

Интересный эксперимент провел Иванов Д.В. из г. Мурманска. Он попытался отыскать планету ISREM (база Таргонов) без объявления миссии. Ее нигде нет!

А вот Плотникову А.В. (DEADLY, 8 мес.) из г. Мончегорска, Мурманской обл. при объявлении миссии E.C.M. JAMMER дали задание уничтожить базу Таргонов на планете DIZAEN, а не IZREM. Так что возможно, что эта база и не имеет постоянного места расположения.

2. Заправка топливом от звезды.

Информация по технологии этого мероприятия приведена в десятках писем и мы не сможем даже всем выразить признательность.

Основная трудность состоит в том, чтобы не перегреть свой корабль.

Надо приблизиться к звезде так, чтобы датчик температуры поднялся примерно на 50% своего предела и подождать, пока появится надпись FUEL SCOOPS ON. Далее можно подождать, пока заправка закончится, внимательно следя за датчиком температуры, а можно развернуться и начать на малой скорости удаляться от звезды.

Кочнев С.В. (ELITE) подчеркивает, что заправка - это длительный и опасный процесс, поэтому следует запастись терпением.

Для ускорения подхода к звезде он рекомендует смело пользоваться JUMP-двигателями, поскольку они отключатся на безопасном расстоянии и сгореть с ними нельзя.

Другая опасность состоит в том, что многие пираты, которых не принимают орбитальные станции, тоже заправляются возле звезды и здесь можно столкнуться с необходимостью сражаться в трудных условиях ограниченного маневра. Противники могут подлетать к звезде гораздо ближе Вас. Любителям острых ощущений он рекомендует по получении 3-ей миссии попробовать заправиться топливом от звезды в системе, где расположена база Таргона.

И еще одна "маленькая хитрость". После заправки, когда Вы прилетите на станцию,

Вас будут пытаться "надуть". В отделе "снаряжение" в графе "топливо" будет проставлена какая-то цена, как будто корабль не заправлен. Если Вы попытаетесь заправиться еще и здесь, Вас "надуют" на эту сумму, не дав взамен ни гектолитра горючего.

3. Новые возможности управления.

Группа пилотов из Новосибирска KVL, KI&SH, Rg, Fed в составе:

В. Кладов (DEADLY, 4 мес.);

А. Шадрин (DEADLY, 4 мес.);

В. Ржевцев (DEADLY, 3 мес.);

Ф. Замараев (DEADLY, 1 год) - исследовали дополнительные возможности управления программой и систематизировали их. Мы и раньше получали информацию по этим возможностям, но с полной систематизацией встречаемся впервые.

Все функции дополнительного управления выполняются в режиме паузы. Переход в режим паузы - нажатием CAPS SHIFT, дополнительные функции можно выбирать по одной или группами.

Q - QUIET - режим "без звука", сохраняется только тихий шорох, сопровождающий выбор самих дополнительных функций.

S - SOUND - обратное включение звука.

1 - EXIT - переход в заставку Press SPACE Commander, после чего восстанавливается последнее записанное состояние.

Все остальные функции действуют по принципу "переключателя", т.е. повторное нажатие клавиши вызывает отключение функции.

K - KEYBOARD - включение клавиатуры в режиме SXNM/5678 (возможно, если после Press SPACE, Commander была нажата стрелка влево).

Y - KEYALT - реверсирование клавиш управления "вверх"/"вниз". Теперь S будет вызывать движение носа корабля вверх, а X - опускание вниз.

B - BACK - реверсирование управления. Следует включать при переходе на задний экран, чтобы не переучиваться управлению кораблем.

D - DYNAMIC - полученная в результате нажатия клавиш скорость изменения угла тангажа и угла крена сохраняется и после отпускания клавиши. Режим полезен при посадке на станцию вручную.

R - REACTION - управление становится более инерционным и приближенным к реальности.

F - Find - поиск и перехват таргонов в гиперпространстве. Дело в том, что ситуация "везения", когда корабль перехвачен таргонами в гиперпространстве, есть результат неаккуратной работы с клавишами, когда происходит случайное включение этой функции. На самом деле таргондяне не умеют перехватывать корабли класса "Кобра". Совсем наоборот: включив режим "F", Ваш корабль начинает перехватывать группы таргонских кораблей (от 3 до 5). Этот режим специально можно использовать для повышения рейтинга, очистки статуса и для зарабатывания капитала.

4. Космические платформы.

Группа KVL, KI&SH, Rg, Fed кажется решила и эту проблему, более того, пилоты научились их специально разыскивать.

Платформа (как они полагают) - это ни что иное, как пиратская станция "Кориолис".

Она не всегда вращается и располагается от планеты несколько дальше обычного. Для ее поиска необходимо идти на планету и когда размер планеты по вертикали достигнет примерно высота экрана, начать пеленг, отворачивая от курса на 90 градусов. Если компас начнет сбиваться, надо продолжать лететь в этой направлении. Даже по достижении прямой видимости такой "базы" компас будет продолжать "прыгать", а буква S исчезать при маневрах корабля. Садиться на такую базу - бесполезно, корабль будет атакован и захвачен пиратами. Уничтожить платформу даже имея ECM System Jammer не удастся.

5. Невидимые планеты.

Судя по сообщениям наших читателей, их немало во Вселенной. Их видели в 5-ой, 6-ой, 8-ой галактиках и, по видимому, они есть и в других.

Так, Минеев А.В. (Владивосток) обнаружил в 47-ой галактике 8 таких планет и приводит их координаты. Координаты довольно условные, поскольку имеют отношение к карте, если нарисовать ее в масштабе 250х140 мм.

1. VEUSRI	124,47
2. MAGEON	142,115
3. SOONTI	142,116
4. ATCEVE	124,46
5. LAGERESO	82, 92
6. VEDIMAER	168,65
7. RAARBICE	169,66
8. VECEER	50, 2

Нам показалось, судя по этому списку, что они появляются парами (по крайней мере 6 из 8). Может быть, это не случайность и у двух остальных тоже есть партнер?

Эту мысль подтверждает и группа KVL,KI&SH,Rg,Fed. Вот что они пишут:

Невидимые звезды обычное явление. Для поиска их на галактической карте необходимо ставить перекрестье прицела на более или менее свободный от звезд участок и нажимать "D". Поиск цели системой наведения идет, скорее всего, по спирали и до нахождения ближайшей звезды локатор успевает обшарить квадрат с центром в установленной позиции.

Невидимые звезды - это двойные звезды. И здесь возможны варианты. Если координаты их в точности совпадают, то на локальной карте можно увидеть колечко и одна из звезд в этом случае недоступна.



Ceedra

Обычно же мы имеем:



Эта идея группы KVL... хороша еще и тем, что проливает свет на природу звезд, обозначенных кольцом на локальной карте, о которых пишут многие корреспонденты.

6. Интересные идеи.

Группа KVL..... предлагает интересное занятие - преследовать убегающие корабли. Особенно интересно преследовать подраненный Fer de Lance (правда в сопроводительном рисунке они изобразили не его, а ASP MK II, догнать который очень непросто. Во избежание подобных недоразумений мы сегодня даем рисунки всех известных кораблей). Во время одной из таких попыток была встречена армада более чем из десяти кораблей, которая мгновенно уничтожила и преследователя и преследуемого. Может быть где-то рядом и была настоящая космическая платформа?

7. Корабли поколений.

Пока никто из наших читателей таких объектов не встречал, но группа KVL... сообщает, что есть сведения о том, что они сохранились в варианте игры, взломанном JOYSTICK CLUBом, а вот в версии Родионова их нет. По слухам это действительно очень большие корабли, похожие по форме на "Кобру", но с характерными очертаниями дюз.

8. Станции "Додо" никто пока не встретил.

9. Тактико-технические данные и внешний вид кораблей.

До сих пор мы не касались вопроса как выглядят те или иные корабли, с которыми можно встретиться в космосе.

Честно говоря, это наш промах. Это следовало бы сделать уже давно. Во-первых,

пилоты ищут новые космические объекты и им надо знать, что перед ними. Во-вторых, судя по нашим письмам, среди пилотов разгорелись дискуссии на эту тему и к нам обращаются для вынесения окончательного суждения.

Ниже мы даем ВСЕ ДОКУМЕНТИРОВАННЫЕ в материалах фирмы корабли. Если Вам встретится нечто иное - фиксируйте, это может быть интересно, если это не астероид.

Для справки мы приводим не только внешний вид кораблей, но и их тактико-технические данные - это поможет Вам сравнить возможности потенциальных противников со своей "Коброй МК III"

Условные обозначения:

LM - Light Mach (скорость света)

CF - Curve Factor (относительная кривизна разворота) - служит для измерения маневренности корабля.

ADDER

Размер: 45/8/30 футов

Грузоподъемность: 2т

Вооружение:

Лучевой лазер Ingram

Ракеты Geret Starseeker

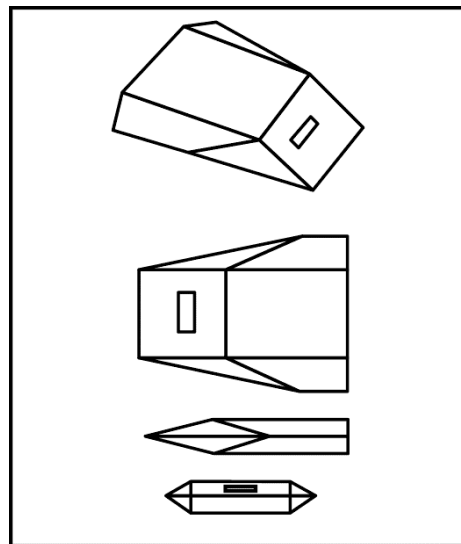
Скорость: 0.24 LM

Маневренность: CF 4

Экипаж: 2 чел.

Способность гиперперехода: да

Год разработки: 2914.



Разработан и выпускается компанией, работавшей без лицензии. Местоположение штаб-квартиры неизвестно. Уникальная технология изготовления складных крыльев позволяет кораблям этого класса совершать посадки и на поверхности планеты. Используются в основном контрабандистами.

Вооружение слабое. Несет только одну ракету.

ASP MK II

Размер: 70/20/65 футов

Грузоподъемность: -

Вооружение:

Взрывной лазер Hassoni-Kruger

Ракеты Geret Starseeker

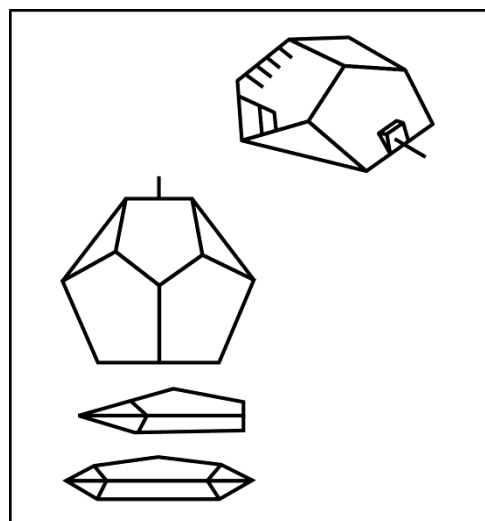
Скорость: 0.40 LM

Маневренность: CF 4

Экипаж: 2 чел.

Способность гиперперехода: да

Год разработки: 2878.



Основная боевая единица галактического флота. Разработана и изготавливается на государственных заводах. Имеет уникальную систему камуфлирования под окружающую обстановку.

Предназначен для разведывательных операций и сопровождения караванов.

Прекрасная маневренность, высокая скорость, мощный лазер и способность маскироваться делает его очень предпочтительным (но и труднодоступным) для пиратов. Большие габариты позволяют нести мощные установки генерации силовых полей, но пусковая установка для ракет только одна.

COBRA MK III

Размер: 65/30/130 футов

Грузоподъемность: 20Т

Вооружение:

Лучевой лазер Ingram

Ракеты Lance & Ferman

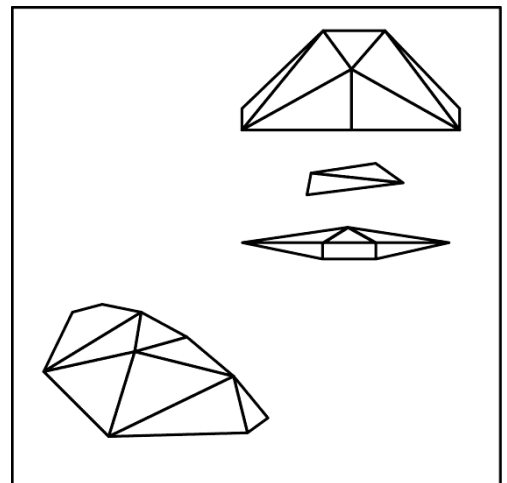
Скорость: 0.30 LM

Маневренность: CF 8

Экипаж: 1 или 2 чел.

Способность гиперперехода: да

Год разработки: 3100.



Корабль оснащен генераторами кормового и носового защитных полей. Боевые надстройки позволяют размещать до 4-х лазеров и пусковых установок ракет.

Наиболее популярен среди независимых торговцев, желающих совместить боевую мощь с приличной грузоподъемностью.

FER-DE-LANCE

Размер: 65/20/45 футов

Грузоподъемность: 2т

Вооружение:

Лазерная система Ergon

Ракеты IFS

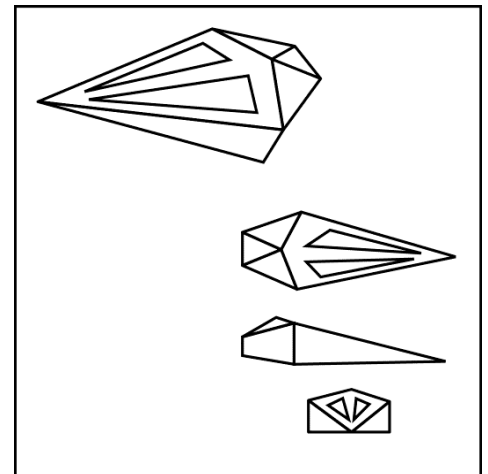
Скорость: 0.30 LM

Маневренность: CF 5

Экипаж: 12 чел.

Способность гиперперехода: да

Год разработки: 3100.



Корабль наиболее широко используется состоятельными охотниками за призами и независимыми компаниями в деловых операциях.

Изысканный корабль, пригодный как для деловых вояжей бизнесменов, так и для боевого применения и для комфортабельного отдыха.

В ущерб грузоподъемности оснащен наиболее совершенным навигационным оборудованием, системами защиты и нападения.

Внутренняя отделка выполнена из самых дорогих материалов.

Fuel-scoops являются стандартным элементом серии - корабль полностью автономен.

KRAIT

Размер: 80/20/90 футов

Грузоподъемность: 10т

Вооружение:

Лазерная система Ergon

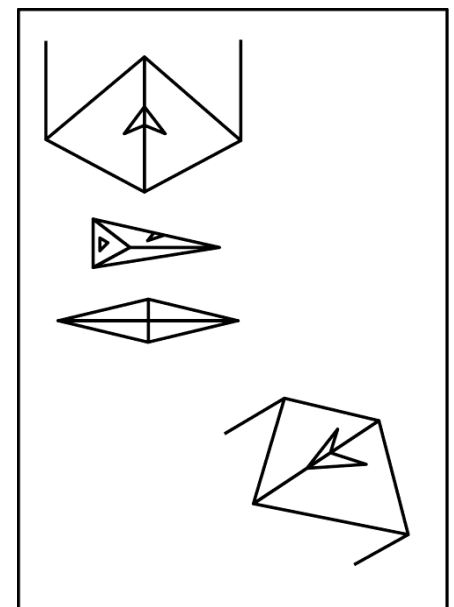
Скорость: 0.30 LM

Маневренность: CF 8

Экипаж: 1 чел.

Способность гиперперехода: нет

Год разработки: 3027.



Небольшой надежный одноместный истребитель. В последние годы был вытеснен более совершенной версией

МАМБА, но в отдаленных секторах космоса он еще встречается.

Запасные части давно не выпускаются и пилоты нередко добывают их посредством пиратства (технический каннибализм).

PYTHON

Размер: 130/40/80 футов

Грузоподъемность: 100 т

Вооружение:

Пульсирующий лазер

Скорость: 0.20 LM

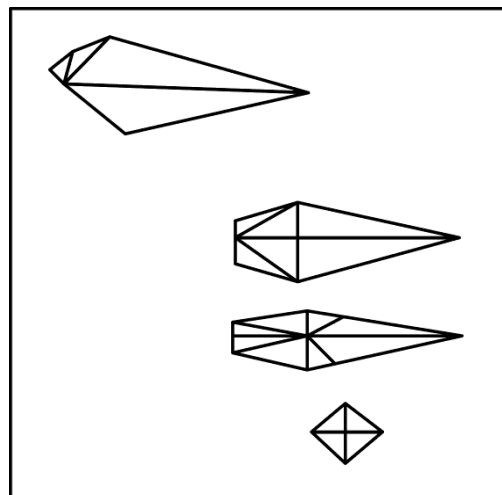
Маневренность: CF 3

Экипаж: 20 - 30 чел.

Способность гиперперехода: да

Год разработки: 2700.

Один из самых крупных торговых кораблей. Малую скорость и плохую маневренность компенсирует мощнейшими установками защитных полей и мощным лазером. Редко подвергается атакам пиратских кораблей. Широко используется свободными предпринимателями в качестве временного склада в челночных операциях и промежуточной базы отдыха.



SIDEWINDER SCOUT SHIP

Размер: 35/15/65 футов

Грузоподъемность: -

Вооружение:

Спаренная лазерная пушка

Ракеты

Скорость: 0.37 LM

Маневренность: CF 9

Экипаж: 1 чел.

Способность гиперперехода: нет

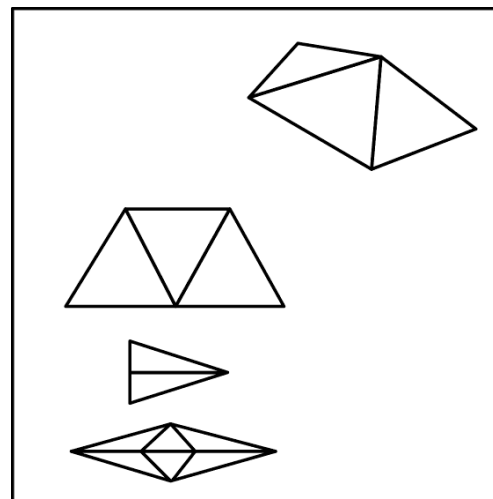
Год разработки: 2962.

Разработан для межгалактических Вооруженных сил в качестве многоцелевого корабля поддержки. Широко используется в разведывательных операциях в атмосфере планет и для воздушно-космической поддержки наземных частей.

Излюбленный корабль для пиратов.

Не имеет способности к гиперпереходу и переносится в грузовых отсеках более крупных кораблей.

Небольшая боевая надстройка не позволяет установить Fuel Scoops и несет только одну ракетную установку.



THARGOID INVASION SHIP

Размер: 130/40/180 футов

Грузоподъемность: не определена

Вооружение: многообразное

Скорость: 0.20 LM

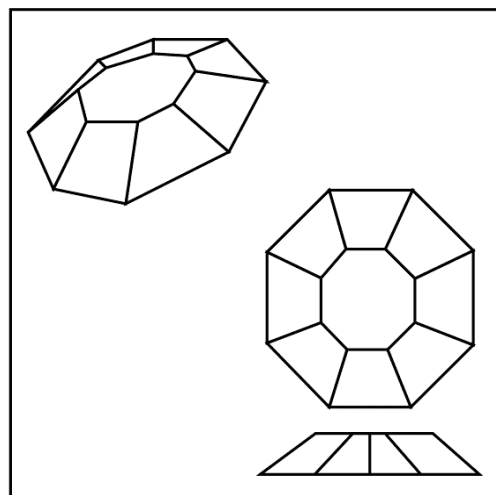
Маневренность: CF 6

Экипаж: 150 чел.

Способность гиперперехода: да

Год разработки: нет данных

Для своих габаритов имеет чрезвычайно высокую



скорость и маневренность. Обязательно оснащены системой ЕСМ (эта система разработана на Таргоне и лишь недавно была скопирована с подбитых Таргонских кораблей).

Базовый корабль обычно несет также эскадрилью малых истребителей с дистанционным управлением.

VIPER

Размер: 55/80/50 футов

Грузоподъемность: -

Вооружение:

Пульсирующий лазер

Ракеты

Скорость: 0.32 LM

Маневренность: CF 7.4

Экипаж: от 1 до 10 чел.

Способность гиперперехода: нет

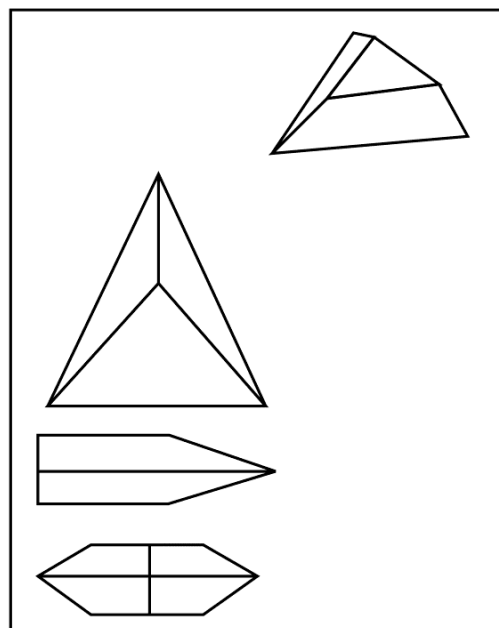
Год разработки: 2762

Малый, надежный, высокоманевренный истребитель-перехватчик. Изготовлен по заказу полицейских сил для патрулирования.

Используется также и вооруженными силами в операциях по сопровождению караванов.

Истребитель одноместный, но можно кратковременно разместить до 10 пассажиров.

Не имеет грузового отсека, но может исполнять роль тягача в случае необходимости транспортировки грузов массой до 140000 т.



Домашнее задание.

Тем, кому надоело летать без дела, мы предлагаем разведывательную миссию.

Ваша задача:

В одной из восьми галактик проложить маршрут через 20 планет таким образом, чтобы суммарный технологический уровень этих планет был максимальным.

Разумеется, расстояние между смежными точками маршрута должно быть таким, чтобы его можно было преодолеть за один гиперпереход.

В качестве подтверждения того, что Вы действительно пролетели по этому маршруту, Вам надо указать каким товаром наиболее целесообразно загружаться в каждой точке этого маршрута. Вы понимаете, что это зависит от разницы в цене между соседними пунктами.

№ п/п	Планета	Тех.уровень	Товар
1	GAREDO	11	FURS
2	NIROGO	7	ALLOYS
.....			
20	CELESTI	14	-
Сумма:		198	

Мы опубликуем в 1-ом номере 1992 года 5 наиболее технологически развитых маршрутов, а их авторам в качестве приза вышлем ксерокопию повести "ELITE: THE DARK WHEEL" на английском языке. Автор - Роберт Холдсток.

TRANTOR

По просьбе нашего читателя мы объявляли атаку на программу TRANTOR. Писем пришло немного. Наиболее полной информацией располагает активно сотрудничающий с нами TiMiSOFT (Т. Саркисов из г. Калуга). Вот что он пишет:

В программе есть несколько уровней. Заходя на каждом уровне в эти компьютеры, можно получить буквы кода. К концу игры их скопится приличное количество, т.е. надо играть с карандашом в руке.

На верхнем уровне имеются два главных компьютера. В первый надо ввести все

собранные буквы (делать это надо быстро и четко) и идти ко второму. В нем набирается второй код, состоящий из слов: KEMPSTON, JOYSTICK, SPECTRUM, SOFTWARE, KEYBOARD, COMPUTER, CASSETE, SINCLAIR, GRAPHICS, HARDWARE, TERMINAL, PRINTERS, CONTROLS, WARGAMES, WARRIORS, MEGAGAME.

К сожалению, в письме не указано надо ли набирать все эти слова или какое-то одно из них является кодом? И настораживает слово CASSETE - все слова имеют по 8 букв, а в этом только 7. Может быть CASSETTE? Впрочем, это достаточный материал для продолжения поиска пытливому исследователю.

На этом мы не прощаемся. Раздел "Форум" в данном номере исчерпан, но не исчерпана и десятая часть писем читателей. Наиболее интересные мы осветили в последующих разделах - POKES и ADVENTURE GAMES.

НАШ КОНКУРС

Наши читатели помнят конкурс на составление лучшей тематической подборки игровых программ. Список допущенных ко второму туру сборников мы публиковали в N4/5.

Сегодня мы можем подвести итоги и рады представить Вам победителей.

Первое место и заказ на разработку подробного комплекта описаний к играм, включенным в список стоимостью 1200 рублей завоевал сборник "Воздушные сражения".

Второе место и аналогичный заказ стоимостью 1000 рублей получает сборник "Приключения Астронавтов-2"

Оба эти сборника собрали одинаковое количество очков от читателей и мы имели бы трудность в распределении мест, если бы не одно обстоятельство - у них один автор - Фокин А.М. из С.-Петербурга. Поздравляем его с двойной победой и просим сосредоточиться над выполнением заказа.

Третье место и контракт на сумму 800 рублей получает сборник "Лабиринт-1". Автор сборника - Д. Котин из г. Кингисепп Ленинградской обл.

Мы будем рады видеть победителей в числе экспертов на страницах "ZX-РЕВЮ".

Все остальные участники конкурса, в том числе и недопущенные ко второму туру, получают поощрительный приз - наш каталог программного обеспечения для "ZX-Spectrum". Мы благодарим участников. Вот они:

Андреев Г.В. (С.Петербург),
Богачев В.В. (С.Петербург),
Брунов О.В. (С.Петербург),
Буденко А.В. (Екатеринбург),
Венско М.В. (Минская обл.),
Дзреев К.К. (Ростов на Дону),
Держкявичюс Р.Б. (Каунас),
Жуков К.А. (Новосибирск),
Зубков С.С. (Моск. обл.),
Ивахненко И.А. (Рязань),
Климов Д.В. (Ленингр. обл.),
Куншиков М.А. (Свердловская обл.),
Лебедев А.Н. (Новосибирск),
Линьков И.П. (Томск),
Панфилов Д. (Ленингр. обл.),
Сипухин Р.И. (Самара),
Смирнов (Оренбург),
Сорокин А.В. (Курган),
Татевосян С. (Ереван),
Хахонов А.Н. (Ленингр. обл.),
Шевелев Э.В. (Серпухов),
Шилин Д.П. (Ульяновск),
Щербак Е.Ф. (С.Петербург),
Яцула И.А. (Алма-Ата).

СЛОВО ЭКСПЕРТА

LAST NINJA 2

System Three 1988г.



Эксперт Саркисов Т.И.
г. Калуга

Игра относится к жанру ARCADE/ACTION.

Незадачливый нинзя, прогуливаясь по городу, зашел в театр, очутился за кулисами, и, встретив наглого, вооруженного противника, понял, что к нему здесь относятся не очень-то дружелюбно.

Наш герой даже не предполагал, что защищаясь и ударив чужака первый раз, он вступил в неравный бой с целым синдикатом, производящим и сбывающим наркотики. По ходу игры нинзя найдет план - карту, с помощью которой можно добраться до нарколаборатории.

Путь не легкий! Но тому, кто познал нелегкое искусство нинзюцу, сделать это будет проще. Вперед!

Управление игрой:

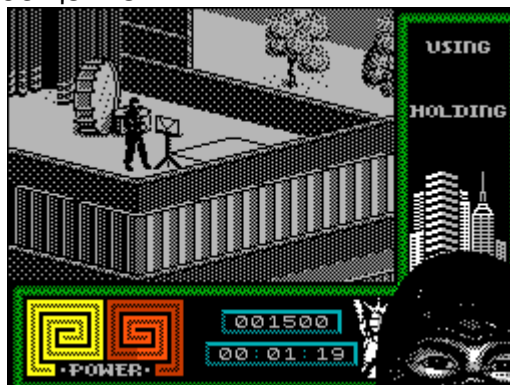
После включения компьютера нажать 0 и выбрать клавиши управления или джойстик.

MODE 1: курсор-джойстик;

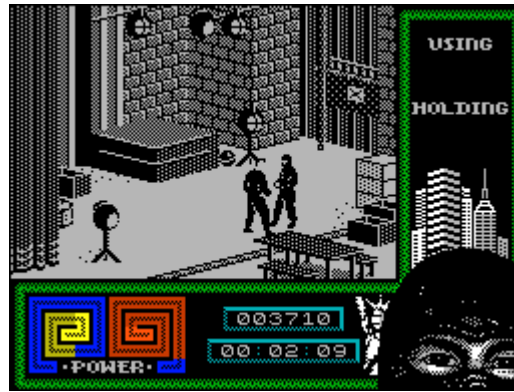
MODE 2: синклер-джойстик 1;

MODE 3: синклер-джойстик 2;

Переключение с джойстика на джойстик выполняется клавишей J, при этом на экране появляется соответствующее сообщение.



Вы начинаете игру в театре, Ваши карманы и руки пусты, о чем свидетельствует левая часть экрана. Позже, когда Вы возьмете какой-либо предмет, нажатием клавиши CAPS SHIFT сможете просмотреть содержимое своих карманов. Если же станете обладателем оружия, оно появится на экране в соответствующем окне. Использовать его можно клавишей "BREAK/SPACE".



Итак, идем налево в темный проход, прижимаясь к левой стене. В следующем экране постарайтесь боевыми приемами обезвредить охранника (прим. 1). Внимательно осмотрев его комнату, Вы увидите маленькое мигающее окошечко. Его нужно выбить ударом руки и вернуться в предыдущий экран. Там Вы увидите открытый люк. Далее все просто - надо в него пройти.

Победа над вторым охранником позволит Вам взять ключ и пройти через темную дверь в левой стене - она ведет в парк.

Встреча с полицейским тоже не предвещает ничего хорошего. Мафия может многое и полиция уже подкуплена. Справиться с ним непросто, но может быть Вам повезет.

Поблуждав по аллеям парка и обезвредив несколько псевдополицейских, Вы можете несколько пополнить свой инвентарь. Здесь Вы найдете карту, палку, нунчаки (прим. 2) и "хот дог" (прим. 3).

Палка лежит около карты. Забравшись по решетке и спрыгнув вниз, Вы без труда ее найдете.

Нунчаки лежат у дверей двух сараев в разных концах парка. "Хот-дог" находится в будке с надписью "C25".

Обыщите также ящик, стоящий возле дорожки. В нем Вы найдете боевые звезды нинзя - шурикены (прим. 4).

Теперь Вы в полном вооружении.

Подойдите к выходу из парка. Чтобы открыть ворота, воспользуйтесь ключом.

Путь свободен. Идите прямо и, дойдя до обрыва, остановитесь! Ждите лодку. Запрыгните в нее, а из нее - на края обрыва и опять идите прямо.

Здесь важно не останавливаться. Вас может настигнуть пчелиный рой и это будет стоить энергии. Бегом по дорожке прямо, потом налево и вверх к реке. Встав на самом краю обрыва, можно перепрыгнуть на остров. Теперь Ваша цель - подтолкнуть лодку, чтобы она поплыла. Подойдите к кустам и толкайте (клавиши "влево" и "удар"). Лодка поплыла, но не отчаивайтесь, Вы ее еще догоните! Прыгайте назад и, спасаясь от надоедливых пчел, бегите влево к обрыву. Подгадав момент, прыгайте в лодку и, проплыв на ней чуть-чуть, прыгайте дальше на уступ.

1-ый блок Вами пройден, но впереди еще четыре: улица, канализационная сеть, склад и офис.

Примечания.

1. Из практики прохождения игры можно сделать вывод, что самым эффективным приемом является удар ногой. Выполняется нажатием клавиш "назад" ("вправо" или "влево" в зависимости от того, куда лицом Вы повернуты) и "удар".

2. Нунчаки и палку удобнее использовать ударом вперед ("вперед" + "удар"), но это не самый эффективный удар. Более эффективен, хотя и менее удобен удар наотмашь ("назад" + "удар").

3. "Хот-дог" - американская сосиска, запеченная в тесте. Она поможет Вам продлить жизнь, когда Вы уже потеряете предыдущие четыре попытки.

4. Шурикены лучше не использовать на первом уровне - их стоит поберечь. Используются они клавишами "влево" ("вправо") и "удар", но предварительно надо переключить оружие с помощью клавиши BREAK/SPACE.

BATMAN

"A BIRD IN THE HAND"

1. KEYBOARD (клавиатура)

2. KEMPSTON (кемпстон-джойстик)
3. SINCLAIR (синклер-джойстик)
4. CURSOR (курсор-джойстик)
5. NEW KEYS (изменить клавиши)

Вы можете выбрать орган управления и одновременно начать игру, нажав клавиши 1-

4. Для изменения клавиш управления нажмите "5":

UP	- вверх	LEFT	- влево
DOWN	- вниз	RIGHT	- вправо
FIRE	- огонь		

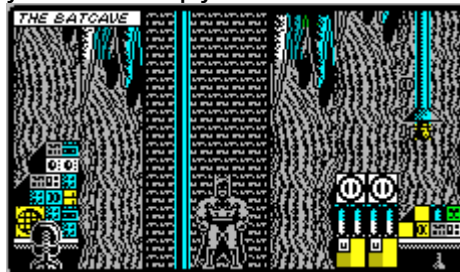
Если Вы не хотите менять клавиши, пользуйтесь следующими:

Z - вправо	C - вверх
X - влево	V - вниз
B - огонь	

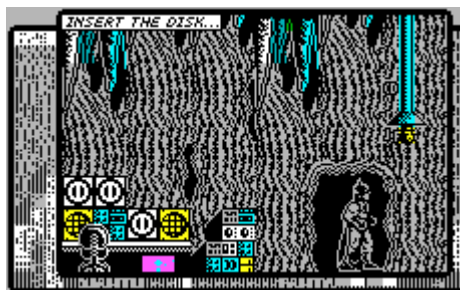
Игра.

Условно всю игру можно разделить на 2 части:

- 1) Поиск дискеты (game disk) и пропуска (pass) в особняк Шутника.
- 2) Проникновение к главному компьютеру и его отключение.



Игра начинается в пещере летучих мышей (the batcave) - это как бы "штаб-квартира" Бэтмена. Здесь ваша задача собрать все предметы, два из которых нужно использовать сразу: починить инструментом сломанное оборудование и вставить личную дискету в компьютер.



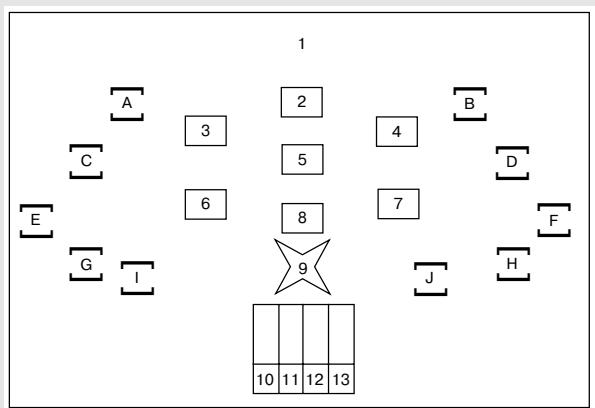
После этого Вы должны найти выход (way out) наверх. Если это вам удалось, Вы окажетесь на улице, где можете столкнуться с наемными убийцами - людьми Джокера. Вы можете наносить им удары или метать бумеранг (если он у Вас есть), но лучше всего не вступать с ними в борьбу, чтобы не терять напрасно силы. Уклониться от пуль наемников можно только присев (одновременно нажав клавиши "вниз" и "вправо" (влево)).

Теперь Вам нужно совершить "прогулку" по крышам домов. Попасть туда можно по замаскированной лестнице, прикрепленной к торцу здания. Всего лестниц три, две ведут на крышу одного, третья - на крышу другого здания. На одной из крыш есть закрытая дверь. Используйте отмычку и входите внутрь. Вы попадете в огромное помещение со множеством комнат и переходов. Где-то здесь спрятана дискета и находится тайник с пропуском. Но сначала Вы должны включить лифт с помощью специального ключа. Попробуйте отыскать комнату с тайником сами. Вот небольшая подсказка: Вам не обойтись без каната. Если же Вам удастся найти эту комнату, на стене Вы увидите большую мишень с портретом Бэтмена - это и есть тайник. Как его открыть - догадайтесь без подсказок. Если это Вам удастся, и в Ваших руках окажется пропуск, можете себя поздравить - с первой частью игры Вы справились успешно.

Теперь Ваш путь лежит в особняк Шутника - двухэтажное белокаменное здание, в котором находится главный компьютер города.

Проникнув в особняк, Вы очутитесь в крошечной тьме - в доме отключен свет. Здесь Вам пригодится карманный фонарик. Теперь ищите компьютер. Сделать это весьма непросто из-за множества запутанных входов и выходов, которые могут ввести Вас в заблуждение (в результате Вы либо попадете в тупик, либо будете по несколько раз возвращаться в одно и то же место). Во избежание этого постарайтесь выработать свою систему поиска (например, по каким-то характерным деталям запоминайте комнаты, где Вы уже побывали или которые ведут в тупик и т.п.) - это поможет Вам быстрее достичь успеха. Если Вы найдете комнату, в которой появится надпись "Three steps to hell" ("в трех шагах от ада"), не пугайтесь - это проделки Джокера, а надпись означает, что Вы на верном пути, и, чтобы попасть к компьютеру, Вам осталось пройти три комнаты. Что делать дальше, Вы знаете.

Условная схема меню:



1. Количество набранных очков (в процентах).
 2. Работа с магнитофоном. (Запись на ленту и загрузка отложенной игры).
 3. Выложить предмет.
 4. Использовать предмет.
 5. Продолжить игру.
 6. Изменить цвет графики.
 7. Изменить цвет фона.
 8. Начать игру сначала.
 9. Курсор в виде контура летучей мыши.
 10. Уровень энергии.
 11. Работа сердца.
 12. Время.
 13. Мышечная сила.
- A-J - содержимое карманов Бэтмэна.

Когда Вы вставите дискету в компьютер, на экране появится надпись: "Computer stopped" ("Компьютер остановлен") и будет показан Ваш результат игры в процентах. Максимальный результат, которого Вы можете достичь - 100%, но для этого Вы должны не только выполнить основную задачу, но и найти верное применение всем предметам, каким возможно. Всего Вы можете найти 24 предмета:

- | | |
|----------|-----------------------------|
| BAT DISK | - личная дискета Бэтмэна |
| BATARANG | - бумеранг |
| BOMB | - бомба |
| CAKE | - торт |
| DART | - дарт (метательная стрела) |
| DOOR KEY | - дверной ключ |
| EGG | - куриное яйцо |
| GAMEDISK | - дискета игры |
| KNIFE | - нож |
| LIFT KEY | - ключ к лифту |
| LOCKPICK | - отмычка |

MAGNET	- магнит
NANA	- бананы
NOSE	- фальшивый нос
PASS	- пропуск
POP	- кукуруза
ROPE	- канат
SHOE	- кроссовка
SWEET	- конфета
TAPE	- видеолента
TOAST	- тост (поджаренный сухарь)
TOOL	- инструмент
TORCH	- карманный фонарь
TRUMPET	- дудка

Чтобы взять предмет, нажмите одновременно "вниз" и "огонь".

Думается, что Вам будет интересно самим узнать назначение каждого предмета. На всякий случай вот несколько подсказок:

1) Продукты питания восстанавливают ваши силы, но очки за их использование не начисляются.

2) Два предмета - бомба и нож - не только не приносят пользы, но и отнимают энергию и время, поэтому не рекомендуется не только использовать, но и подбирать их.

3) Некоторые предметы можно использовать где и когда угодно, другие - только в определенном месте.

Управление игрой.

В любой момент игры Вы можете войти в режим пиктографического меню, с помощью которого управляется программа (одновременно нажав клавиши "вниз" и "огонь").

Управление осуществляется курсором в виде летучей мыши.

Например, для того, чтобы воспользоваться каким-либо предметом, сначала подведите курсор к нему и нажмите "огонь" (предмет начнет мерцать), затем переместите курсор к пиктограмме в виде кулака - "использование предмета" и вновь нажмите "огонь". Если предмет использован верно, Вам начислят очки, после чего Вы автоматически вернетесь в текущий момент игры. Если же ничего не произошло, значит предмет использовать нельзя.

Если Вы захотите начать игру сначала, войдите в режим меню, подведите курсор к соответствующей пиктограмме и нажмите "огонь". Компьютер осведомится: "Quit? (y/n)" (Вы уверены? (да/нет). Нажмите "Y" или "N" соответственно.

Полезные советы.

1. В особняке Шутника Вы можете встретиться с самим хозяином, наряженным в клоунский костюм и с сигарой во рту. Не пытайтесь его убить - он неуязвим, а Вы только потеряете силы и время.

2. Пингины - убийцы обезвреживаются ударом ноги вниз (одновременное нажатие "вправо-влево", "вниз" и "огонь").

3. Если Вы нашли какой-нибудь полезный предмет (например, дискету), а места в ваших карманах нет, значит, какой-то предмет оттуда нужно использовать немедленно.

4. Не ленитесь почаще проверять, нельзя ли использовать какой-нибудь предмет - так Вы быстрее добьетесь успеха.

5. Не забывайте время от времени восстанавливать свои силы, съев что-нибудь из продуктов.

Удачи Вам!

GUNFRIGHT
(РЕВОЛЬВЕРНЫЕ СТРАСТИ)
Ultimate 1985г.



Эксперт Ескевич А. А.

Игра совмещает в себе два жанра - ARCADE и ACTION, но поскольку в ней есть возможность пользоваться некоторыми объектами (например деньгами), в ней есть и элемент ARCADE/ADVENTURE.

Сюжет игры относится ко второй половине XIX века. Действие происходит в одном провинциальном американской городке в южном, а может быть и западном штате.

Обстановка в городе сложилась не самая лучшая. Выходить на улицу с каждым днем становится все опаснее. Темные личности, разыскиваемые властями, все чаще посещают этот город. На стенах вывешиваются объявления о розыске бандитов (WANTED FOR...) и в этой обстановке Вас назначают шерифом.



Найти и уничтожить бандитов - Ваша непосредственная задача. За каждого Вы получите солидное вознаграждение и чем опаснее бандит, тем больше платят за его голову.

Найти преступника в запутанном лабиринте улиц Вам помогут некоторые благонамеренные обитатели города. Вы их легко узнаете - подпрыгивая от волнения, протянутой рукой они указывают Вам, где находится преступник.

По городу бродят и другие населяющие его жители подозрительного вида. Не надо сталкиваться с ними (а также и с колючими кактусами). За убийство же случайного прохожего вам придется уплатить еще и крупный штраф (от 100 до 150 долларов в зависимости от уровня жизни на данный момент).

Ускорить свое передвижение Вы можете взяв лошадь (50 - 75 долларов). На ней Вы неуязвимы для прохожих, но при столкновении с бандитом или кактусом потеряете не только лошадь, но и жизнь.

Приобретение патронов для револьвера тоже стоит денег (5 - 10) долларов за штуку.

Итак, цель игры можно сформулировать так: обезвредить как можно больше головорезов при минимальных финансовых затратах.

Структура экрана.

Слева изображен портрет разыскиваемого преступника, под которым помещена

сумма, назначенная за его уничтожение. Внизу указана Ваша наличность, количество патронов в барабане кольта, текущая цена на патроны (BULLET), лошадь (HORSE), и сумма штрафа (FINE) за непреднамеренное убийство случайных прохожих.

Настройка программы.

После загрузки Вы получаете меню:

1. ONE PLAYER (один игрок).
2. KEYBOARD (клавиатура).
3. START GAME (начать игру).

Вы можете выбрать игру вдвоем, нажав клавишу "1" и изменить управление игрой, нажав на клавишу "2". Вы можете выбирать джойстик KEMPSTON, CURSOR, SINCLAIR(INTERFACE-2).

Игру Вы начинаете с суммой 250 долларов.

Первая фаза - получить первоначальный капитал. На экране появляется прицел и сверху начинают падать мешки с деньгами. Ваша цель - настрелять как можно больше денег.

Закончив "отстрел", можете начинать розыск бандита. Пользуйтесь указаниями прохожих, но избегайте возможных столкновений с людьми и с кактусами. Следите за изменениями обстановки. Заметив бандита, ни в коем случае не подходите к нему вплотную, а стреляйте с расстояния.

Если выстрел был точен, заиграет музыка и начнется заключительная фаза. На экране появится увеличенная фигура головореза, перемещающаяся случайным образом. Надо быстро навести прицел и выстрелить. Если опоздаете и бандит откроет стрельбу раньше, Вы будете убиты первым же выстрелом.

Если преступник поражен, то после музыкальной паузы Вы получите вознаграждение.

На этом игра не кончается. В город пожаловал еще более хитрый и опасный бандит, и теперь Ваша задача усложняется.

За пять обезвреженных преступников Вы получаете премию BONUS. Как и вначале игра, Вам предоставляется возможность "настрелять" столько мешков с деньгами, сколько позволят реакция и сноровка.

Бывает, что все деньги кончились, а преступник все никак не отыскивается. В этом случае Вам может повезти и Вы найдете мешок с деньгами.

За пятнадцать убитых преступников Вы получите благодарность, после чего игра начинается сначала.

Управляющие клавиши.

а) При патрулировании города:

(управление шерифом)

Два верхних ряда - "Огонь"

"X", "V", "B", "M" - "Налево"

"C", "N" - "Направо"

от "A" до "ENTER" - "Вперед"

б) при стрельбе:

верхний ряд - "Огонь"

второй ряд - "Вверх"

третий ряд - "Вниз"

"X", "V", "B", "M" - "Влево"

"C", "N" - "Вправо"

Клавиши "CAPS" или "BREAK" - "Пауза".

Полезные советы.

1. При стрельбе по мешкам не стоит манипулировать прицелом по всему экрану. Достаточно опустить его вниз и перемещать вправо/влево.

2. При входе в дом на всякий случай выстрелите - это может спасти Вашу жизнь.

3. Если столкновение с прохожим неизбежно, разумнее застрелить его и заплатить штраф, чем потерять жизнь.

4. Если у Вас нет денег, а Вы нашли лошадь - можете ехать бесплатно.
5. Сидя на лошади не топчитесь на месте - время езды ограничено.
6. Если прохожий, указывающий где укрылся преступник, начинает поворачиваться то в одну, то в другую сторону, будьте начеку - преступник совсем рядом.
7. Следите за тем, чтобы в барабане вашего кольта всегда оставалось не менее двух патронов.

Удачи Вам!

POKES

Мы развиваем тот материал, который был напечатан в N7/8. Сегодня этот блок составлен исключительно по письмам наших читателей.

Д.В. Иванов (Мурманск) приводит еще пару POKES для "обессмерчивания" CAULDRON-2.

Для блока длиной 41986 - POKE 52974,0.

Для блока длиной 41447 - POKE 52716,0.

Последний был им проверен.

С очень важной информацией обратился уже известный Вам эксперт Саркисов Т.И. (TiMiSOFT). Начиная с этого года в Польше начал выходить журнал "TOP SECRET", предназначенный для любителей компьютерных игр. Несколько новых POKES приведены им из этого журнала. Все эти POKES были им лично проверены.

BEYOND THE ICE P.	- 38379,0
BOSCONIAN	- 33066,255
BUGGY BOY	- 39086,0
GEMINI WING	- 41740,24: 40076,0
GUNFIGHTER	- 65535,0
MASK III	- 46045,58: 47778,58: 49872,58: 45589,201
MASTERS OF UNIVERSE	- 42173,0: 51406,0
MICKEY MOUSE	- 40873,0: 40012,0: 40091,0: 40114,0
MOON ALERT	- 39754,0: 42654,195
MUTANT MONTY	- 54933,0
PACLAND	- 35141,0
PSYCHO SOLDIER	- 40123,0
REBEL SQUAD	- 46840,295
SKATE CRAZY	- 46473,201: 46646,126: 46409,201
SCEPTRE OF BAGDAD	- 59858,0
STORMLORD	- 58105,0: 56877,0: 34650,0
TANK COMMAND	- 65535,0
TRANTOR	- 54236,0: 56596,0
XECUTOR	- 47216,201: 47320,201

TiMiSOFT не ограничивается этой информацией и любезно открывает еще несколько секретов любителям компьютерных игр.

AUF WIEDERSEHEN MONTY.

Программу следует загрузить следующим образом:

LOAD "" : REM MONTY

После загрузки и запуска программы поднимите предмет, лежащий слева и Ваш Монти становится бессмертным.

BUGGY BOY.

От финиша Вас отделяют лишь миллиметры, а счетчик опустился до нуля. Не берите это близко к сердцу и не бросайте джойстик. Нажмите "вверх" и автомобиль с разгона пересечет финишную черту.

GEMINI WINGS.

Пароли уровней:

- | | |
|--------------|-------------|
| 1. THE START | 5. SKULLDUG |
| 2. EYEPLANT | 6. BIGMOUTH |
| 3. WHATWALL | 7. CREEPISH |
| 4. GOODNITE | 8. FINALFXS |

KOSMOS.

Во время игры перейдите в меню и нажмите клавиши SYMBOL SHIFT и "K". Это дает бессмертие и доступ ко всем предметам.

LED STORM.

Когда закончится отсчет, наберите еще немного очков и два раза нажмите BREAK. игра будет прервана и после повторного старта Вы обнаружите, что имеете на счетчике 300000 очков.

NONTETTAQUEOS.

Находясь в БЕЙСИКе, введите:

INK 0: BORDER 0: PAPER 7: INVERSE 1: BRIGHT 1

Затем два раза нажмите ENTER и загрузите игру через LOAD "". Титульная заставка будет выглядеть немного странно, но зато противники в игре исчезнут наверняка.

Кулаков Ф.В. (Москва) нашел интересный "жучок" в игре TARGET RENEGADE. Суть его состоит в следующем:

Надо набрать такое количество очков, чтобы попасть на "Доску почета". Здесь, когда вам надо будет ввести свое имя, нажмите какую-либо клавишу и, удерживая ее, нажимайте другую. Это надо сделать столько раз, чтобы появившаяся в "доске почета" запись перекрыла показанный Вами результат на 2-3 символа (не больше).

После небольшой паузы появится место, в котором Вы погибли. Это будет черно-белое, не очень хорошее изображение. Вскоре Вы "воскресаете" и после короткого боя перейдете в другой экран, где изображение нормальное. Теперь Вы бессмертны и все это удалось сделать без введения в программу POKES.

Несколько POKES, найденных самостоятельно прислал нам С.В. Сыромолотов (Магаданская область). Нас особенно порадовало, что он разобрался с ассемблером и научился разбираться в фирменных программах и писать свои процедуры по нашему трехтомнику "Первые шаги в машинных кодах". (Все-таки очень приятно, что книга, написанная нами, работает и помогает людям в их увлечении. Как-то чувствуешь себя увереннее и хочется работать еще больше).

A.FORCE 2	- 50262,N - попытки.
GAME OVER 1	- 31870,N - попытки.
	- 31880,1 - гранаты.
GR. BERET	- 46828,0: 46829,0: 46830,0 - гранаты.
OP.GUNSHIP	- 56304,0 - бомбы.
	- 55976,0 - ракеты.
	- 57907,201: 53912,201: 57914,0: 57915,0: 57916,0: 57901,0:
	57902,0: 57903,0 -бессмертие.
SAVAGE-1	- 35454,N - попытки.
SAVAGE-2	- 29177,60 - бессмертие
SAVAGE-3	- 44251,0 - энергия.

Свою существенную лепту вносит в общее дело и Шапка В.Н. из г. Киева. Вот присланный им список:

A.FORCE 2	- 50145,36 - энергия
AMAZON WOMEN	- 57590,183
ARCADIA	- 25776,0
AUF WIEDERSEHEN MONTY.	- 42160,201: 37002,0
AIRWOLF	- 45982,0
ALCHEMIST	- 31055,0
BOUNDER	- 36610,0
BLACK HAWK	- 34695,183

BRUCE LEE	- 51795,0
BLADE ALLEY	- 58201,0
BATTLE RACE	- 44641,0
COOKIE	- 35730,52
CAVERN FIGHTER	- 31663,0: 31684,0
CYBERUN	- 36168,175
DEFENDA	- 37630,52: 37283,0
DEFENDER	- 30822,255: 37815,255
DEATHCHASE	- 26463,0
FANTASTIC VOYAGE	- 54277, 0: 54492,0
FRANK'N'STEIN	- 28287,255
FAIRLIGHT	- 51893,0: 62797,24 63478,24: 58813,62 58814,6
FINDERS KEEPERS	- 33969,0: 34252,0
GROUND ATTACK	- 24872,0
JUMPING JACK	- 30094,182
JET PACK	- 25020,255: 26075,0
NIGHT SHADE	- 58056,0: 57499,0: 53442,0: 53443,12: 51105,0
ORION	- 37319,201
1942	- 47001,255
PROJECT FUTURE	- 27662,0
PAPER BOY	- 48023,201
RIVER RESCUE	- 33199,255
SCOOBY DOO	- 64027,86: 64028,5: 29614,0
SWEWO'S WORLD	- 33219,0
SPINDIZZY	- 48272,201:48401,201
SAI COMBAT	- 63364,201: 37421,1
SABOTEUR	- 29894,0 - энергия
SCUBA DIVE	- 55711,N - попытки
TERMINIUS	- 45583,0: 47023,0
THE PYRAMID	- 44865,0
TLL	- 55006,0
UNDERWURLDE	- 59376,0: 59380,0
V.C.M.	- 23296,255
WAR	- 38394,0
WIZARD'S LAIR	- 25522,N - попытки
ZIP ZAP	- 54065,0
ZOOM	- 24743,0
ZAXXON	- 48825,X - попытки

Adventure Games

Как оказалось, самое трудное в этих играх - начало. Вот и сегодня т. Иванов (Мурманск) спрашивает как начать игру в AFTERSHOCK и как выйти из дома в программе SHERLOCK.

В AFTERSHOCK оглядитесь по сторонам (LOOK), попробуйте все предметы (EXAM), можете взять с собой то, что возьмется (TAKE).

Пройдите на восток (E), а оттуда на все четыре стороны (N,E,S,W). Где можно попробуйте вверх (U) или вниз (D). Попад в лифт осмотритесь - если что-то увидите и не сможете достать, принесите из офиса стул и влезьте на него (CLIMB ON CHAIR). Вот Вы уже и начали игру. И нажимайте когда надо HELP (H) - программа может быть что-то и подскажет.

Теперь что касается Шерлока Холмса:

День 1. Понедельник, 8:00.

Холмс сидит вместе с Уотсоном в своем особняке на Бейкер-Стрит, 221Б. Уотсон держит в руках газету.

READ CHRONICLE - говорит Холмс Уотсону, после чего проходит в гардеробную (plain door). Пройти в каком-либо направлении, если на пути есть дверь, можно только предварительно открыв дверь:

OPEN PLAIN DOOR

Здесь хранятся два комплекта маскировки (disguise). Один комплект - для маскировки под старика, другой - под китайца. Их надо взять с собой. Чтобы их взять, их надо одеть, например:

WEAR OLD MANS DISGUISE

а потом снять - TAKE OFF.

Взяв оба комплекта, Холмс возвращается в гостиную. Здесь он должен взять лампу - TAKE OIL LAMP и перед выходом на улицу пригласить с собой Уотсона - SAY TO WATSON "FOLLOW ME".

Выход прост - открыть дверь (your door), выйти в холл, спуститься по лестнице, открыть входную дверь (front door) и выйти на улицу.

Итак, время 8:16 и Ваши приключения только начинаются - счастливого пути.

В нашей почте есть еще одна просьба относительно адвентюрных игр. К.Г. Терпогосов из г. Алма-Ата прошел 26% программы SPELLBOUND и "застрял". Может быть кто-нибудь ему поможет? Для тех, кто еще не пробовал эту игру он приводит свой дебют:

Действие происходит в обширном, многоэтажном строении. Этажи соединены лифтом, на крыше - сад, а на каждой этаже помногу комнат. В комнатах разбросаны разные предметы, которые наш герой может брать.

Многие из них могут быть полезными, на одних можно что-то прочитать, другие могут дать возможность телепортироваться, третьи могут использоваться для вызова других персонажей и отдания им команд и т.д.

Персонажи встречаются в самых разных местах. У многих из них тоже есть какие-либо предметы. Они могут их дать, а могут и отказать в Вашей просьбе (в зависимости от ситуации). Но они могут брать предметы у главного героя.

Цель игры - найти колдуна (Gimbol The Wizard), каким-то образом его победить и расколдовать остальных персонажей.

Список действующих лиц:

Gimbol the Wizard - колдун, которого надо найти.

Thor - бесполезный человек, предметов не имеет, требует чтобы его постоянно кормили командой "EAT AND DRINK", в противном случае умирает от истощения и игра заканчивается.

Florin the Dwarf - пещерный гном. Очень хороший человек, имеет при себе топор как всякий уважающий себя гном. С его помощью главный герой и сам может подкормиться.

Orik the Cleric - роль не ясна, предметов не имеет.

Samsun¹ the Strong - имеет очень полезную вещь - Elf Gorn (рожок эльфов), с помощью которого можно вызывать других персонажей. Есть у него еще некая платформа (Platform), которую никогда не отдает.

Elrand Halfelven - имеет два предмета непонятного назначения.

Lady Rosmar - имеет два предмета непонятного назначения, один из которых никогда не отдает.

The Banshee - дух, который появляется, если разбить бутылку (Glowing Bottle).

Полезные предметы:

Glowing Bottle - бутылка, которую надо обязательно иметь с собой, чтобы пройти через комнату "Little Bottle". Взять ее надо на полке с цветами в Plant Room.

Teleport Key - ключ телепортации. При его наличии можно мгновенно перемещаться в ту комнату, где лежит Teleport Pad.

Bottle of Liquid. - Лежит в саду на крыше возле лифта. Если дать эту бутылку Флорину, а затем забрать, энергия возрастает до максимума. Периодически это стоит делать.

Wand of Command - "волшебная палочка", позволяющая командовать персонажами. Лежит под деревом в саду.

Mirror - зеркало, отображающее уровень энергии и наличие предметов у главного героя.

Elf-gorn - если в него протрубить, то можно вызвать к себе других героев.

Главный герой - Magic Knight может одновременно нести не более 5 предметов.

В игре есть еще разные предметы, но назначение их не вполне ясно.

Дебют товарища Терпогосова выглядит так:

Из первой комнаты (Start Room) взяв Teleport Key он телепортировался на Teleport Pad в Plant Room. Здесь, взобравшись на полку с цветами, он взял Glowing Bottle, снова телепортировался, взял Teleport Pad и отправился налево для встречи с Флорином. По дороге выбросил объявление Advert чтобы освободить руки для более полезных предметов.

Встретив Флорина, оставил около него Teleport Pad и пошел опять налево. Преодолевав "горку", на вершине которой взял зеркало (Mirror) и прошел к лифту в левой стене библиотеки. Вызвав лифт, он проехал на крышу (Roof), где взял Bottle of Liquid и, затем пройдя через весь сад (Roof Garden), взял волшебную палочку (Wand of Command) и опять телепортировался к Флорину.

Обменявшись с ним бутылкой "Bottle of Liquid" и пополнив тем самым запас энергии, он дал команды "WAKE UP" и "EAT AND DRINK", чтобы тот проснулся и поел.

Затем, захватив с собой Teleport Pad, он двинулся по знакомому пути к лифту. Там оставил Teleport Pad и проехал на 2-ой этаж для встречи с Samsun the Strong, у которого надо забрать Elf Gorn. Идти от него надо направо до стены, где можно разбить Glowing Bottle и вызвать таким образом Banshee. Протрубив в рожок (Elf Gorn) можно вызвать Флорина и Тора. Тор встанет у стены и обеспечит проход главному герою, который должен собрать все предметы, имеющиеся в помещении, и разбудить находящегося здесь Орика. Ему можно передать имеющиеся лишние предметы.

Далее можно телепортироваться в лифт на Teleport Pad, вызвать туда всех персонажей и накормить их.

2-ой, 3-ий этажи и крыша уже осмотрены. Теперь можно заняться остальными помещениями.

На 1-ый этаж (Ground Floor) и в подвал (Basement) лифт не ходит из-за того, что якобы не исправен. Впрочем, намекают на то, что есть возможность его починить, если исправить щит управления, неизвестно где находящийся.

На 1-ом этаже есть некая "Грибная комната" (Mash Room). За ней идет Gas Room,

¹ Samsun(Прим.OCR)

через которую нельзя пройти. Там видны какие-то предметы и намекают, что проход возможен, но надо знать как.

На 4-ом этаже в первой же комнате есть стена, которая идет не до самого потолка, через нее можно перебрасывать предметы персонажам, находящимся за стеной (их туда можно вызвать посредством рожка). Однако перебросить Teleport Pad не удастся, т.к. она слишком тяжела и персонажи, в которых она брошена, погибают.

В лифте лежит шар, который применяется для определения местоположения персонажей, кроме этого на нем можно прочитать различные двусмысленные подсказки. Иносказательно изъясняется также и Banshee (по команде HELP).

Таков дебют, который Вы можете развить или опровергнуть. Возможно, что в нем есть ошибки.

Итак, ко всем любителям адвентюрного жанра обращено предложение прорваться как можно дальше.

Определенно есть смысл этим заняться, поскольку игра имеет еще два продолжения - KNIGHT TYME и STORMBRINGER и может долго доставлять удовольствие.

Это очень умная и безусловно полезная игра.

Ждем ваших сообщений.



МП "ИНФОРКОМ" Москва, 107241, а/я 37

Внимание!!!

"ИНФОРКОМ" сообщает, что начиная с 01.01.92г. изымаются из обращения все бланки заказов, выславшиеся в 1991 году. Заказы и оплата по ним не принимаются и не исполняются.

Начиная с 01.01.92 мы рассылаем бланки двухмесячного действия. Заказы и оплата по ним принимаются только в течение их срока действия.

Заказы, присланные не на наших бланках не исполняются. Рекомендуем Вам перед тем, как что-то заказывать, обратиться к нам письмом для получения бланка и уточнения текущих цен.

"ИНФОРКОМ"

"ЖИВАЯ НИТЬ"

Дорогие друзья!

"ZX-РЕВЮ" отмечает год своего существования. Можно подводить какие-то итоги.

И самый первый итог: мы живы, и собираемся жить дальше. Более того, почти все наши постоянные читатели подписались и на 1992 год. В этом мы видим признание того, что делаем нужное дело.

Мы рады, что смогли предоставить Вам возможность подписаться на 92-ой год без повышения цены, несмотря на то, наши затраты многократно возросли.

Вспомните, что мы писали Вам более года назад: "ZX-РЕВЮ" - это не журнал и не газета. Это персональное письмо." Сегодня мы к этому добавляем "Это живая нить, связывающая нас с Вами."

У "ZX-РЕВЮ" своя особенная миссия и мы рады Вам сообщить, что по-нашему мнению цель, поставленная полтора года назад, успешно достигнута: в стране сложился дружественней многотысячный коллектив поддержки "ИНФОРКОМа". Нам кажется, что мы не ошибаемся в своих оценках - мы судим по тысячам очень и очень теплых писем.

Мы рады Вам сообщить, что начиная с 1 января 1992 года мы уже не считаем Вас просто клиентами и просто читателями, мы вступаем с Вами в ПАРТНЕРСКИЕ ОТНОШЕНИЯ. Организационная, материальная, программная базы для этого созданы.

В тысячах городов, больших и малых, есть наши представители. Более того, они нас ЗНАЮТ, а мы ЗНАЕМ их. Мы не раз к Вам обращались с просьбами и Вы откликались. Мы же, как могли, выполняли Ваши просьбы и пожелания. Мы хорошо друг друга знаем. Знаем кто что может, а что нет. И теперь мы начинаем новый этап совместной работы. Надеемся, что он продлится долгие годы.

Мы решительно заявляем, что сегодня "ИНФОРКОМ" выходит в стране на лидирующие позиции в области разработки обучающих программ для IBM-совместимых компьютеров. Любой сомневающийся в этом может попробовать предложить нам что-нибудь более значительное и удобное в работе, чем наш программный комплекс "АНГРАМ" (см. последнюю страницу).

Мы полтора года готовили те проекты, к распространению которых начинаем сегодня привлекать и Вас. Пробный маркетинг этих программных продуктов уже проведен и поддерживающая реклама по стране уже прошла. Мы точно знаем что и как стоит и как и кем покупается. Но сколь бы ни были наши программы хороши и необходимы, мы не сделаем

самый решительный шаг без Вашей помощи. Каждый из Вас, кто хочет, может стать дистрибутором нашей фирмы в своем городе, поселке, на своем предприятии или в своем учебной заведении. Совместными усилиями мы не только поможем друг другу, мы дадим стране то, чего она не дожидется в полной мере от Академии Педагогических наук.

Итак, приступаем к делу.

Отныне в каждом номере на последней странице мы будем давать развернутое описание какой-либо нашей системы или программы.

Зная нужды своего региона или знакомых вам предприятий Вы всегда сможете выбрать из наших продуктов что-то, что можно предложить в той или иной организаций для приобретения.

Условия предельно просты. (Подробности на стр. 263)

Вы можете ничего не знать об IBM-совместимых компьютерах, но если Вы знаете НАС и верите НАМ, а те, кому наши разработки необходимы, знают ВАС и верят ВАМ - это уже победа. Вам должно очень крупно не повезти, если Вы не сумеете десятки раз окупить тех средств, которые затратили на "ZX-РЕВЮ". Но цель не только в этом.

Наши планы и наши проекты не кончаются ни в 92-ом ни в 93-ем годах. Ведутся уже работы над продуктами 94-го года. Мы не оставим ни одного учебного предмета, не охваченного нашей программной поддержкой. Складывается мощная структура из преподавателей, программистов, методистов, менеджеров, специалистов по маркетингу и опираться она будет на Вас - региональные дистрибуторские сети.

Где бы Вы ни жили и кем бы ни работали, можете быть нашим опорным пунктом и мы на Вас рассчитываем.

Вы же можете рассчитывать на достойные комиссионные отчисления, программно-информационную поддержку, постоянное обновление ассортимента нашей программной продукции и ее самую высокую конкурентоспособность, и самое главное - наши планы работ рассчитаны на очень и очень многие годы. Наше развитие имеет гигантские перспективы, приглашаем Вас развиваться в своих регионах параллельно с нами не меньшими темпами.

Если Вы еще школьник и у Вас нет нужных контактов - не отчаивайтесь. Мы думаем и о завтрашнем дне. Придет и Ваше время. Может быть через десять лет нам еще дороже будут наши проверенные годами партнерские связи. Пока читайте "ZX-РЕВЮ", набирайтесь опыта и знаний. Мы не живем одним днем и рассчитываем свои шаги на далекое будущее.

Итак, всем желающим войти в нашу структуру - путь открыт. Читайте последние 2 страницы - там написаны условия работы по проекту "АНГРАМ". В следующем номере представляем многоцелевую информационно-поисковую систему "РЕГИСТРАТУРА", которая найдет свое место в любом учреждении, на любом заводе - где угодно, где бы ни стоял IBM-совместимый компьютер.

ИНФОРКОМ.

FORUM

От имени группы читателей к нам обратился С.Е. Троицкий из Москвы с просьбой прокомментировать явление, с которым они столкнулись в своей работе и дать рекомендации по путям его обхода.

Проблема связана с организацией работы в циклах, когда параметр цикла задан числом с плавающей точкой. Суть проблемы хорошо демонстрируется на двух следующих примерах.

```
10 FOR i=0 TO 1 STEP 0.01
20 IF i <> 0.5 THEN PRINT i
30 NEXT i
```

По этой программе компьютер печатает все значения от 0 до 1, в том числе и 0.5 (?!).

```
10 FOR i=0 TO 1 STEP 0.01
20 IF i=0.5 THEN PRINT 1
30 NEXT i
```

Здесь компьютер ничего не печатает (!?).

Примеры можно продолжать, но суть ясна - что-то не так и надо что-то делать.

Нас в этом вопросе заинтересовала прежде всего его академичность. Проблема в общем-то типовая и, что самое интересное - она имеет отношение не только к "Спектруму", а вообще к любым видам вычислительной техники и знать пути обхода ее надо каждому, кто собирается применять компьютер в инженерных вычислениях.

Итак, начнем. Вы знаете, что компьютер хранит числа в байтах, каждый из которых состоит из восьми битов. Для целого числа от 0 до 65535 достаточно и двух байтов, но для действительных чисел с плавающей точкой этого недостаточно - здесь "Спектрум" использует 5 байтов на каждое число (другие компьютеры могут использовать другое количество байтов, но суть проблемы не меняется). Запись действительного числа в 5-ти байтной форме называется интегральным представлением числа. Хотите узнать все подробно - читайте наш трехтомник по программированию в машинных кодах - там это доступно изложено.

Хоть 5 байтов это и немало - целых 40 битов, тем не менее точность, с которой число может быть занесено в память, все равно не бесконечная. Так, для "Спектрума" эта точность - 8 значащих цифр.

После всего вышесказанного Вы не будете удивляться, если число 0.01 будет храниться в памяти компьютера как 0.0099999999.

Теперь дело за процедурами. Процедуры, выполняющие сложение, после 50 проходов по Вашему циклу могут привести к тому, что вместо 0.5 параметр цикла *i* может иметь что угодно от 0.049999949 до 0.049999999, но никак не 0.5. И процедура сравнения в строке 20 однозначно ответит, что эти числа "НЕ РАВНЫ". Тогда в действие вступит процедура печати, которая в первом примере должна напечатать 1. Она округлит имеющийся результат и выдаст 0.5.

Другое дело имеет ли она право выполнять такое округление. На вычислительных системах, предназначенных для выполнения инженерных и физических расчетов конечно никто такого права компьютеру не даст. "Спектрум" же - бытовая машина и домохозяйке (школьнику, студенту, поэту, журналисту, рабочему и др.) гораздо приятнее видеть на экране 0.5, чем громоздкое 0.049999949 или еще того лучше 4.9999949 E-02.

Теперь другое дело - как Вам это обойти, чтобы сравнение все-таки работало и работало правильно.

Во-первых, давайте договоримся, что такие операции нужны прежде всего в инженерных расчетах, а они, как известно, выполняются всегда с ограниченной точностью. Например, вычисляя длину окружности по ее радиусу и закладывая радиус равным, например 0.25 м Вы тем самым утверждаете, что радиус вам известен с точностью равной двум значащим цифрам. Какой же Вам смысл получать длину окружности с точностью до 8 значащих цифр, если Вы знаете, что все равно цифрам от третьей до восьмой веры нет?

Во вторых, давайте введем на время Ваших вычислений понятие точности.

Предположим для приведенных выше примеров Вас устраивает точность порядка 0.00001. И теперь давайте с ней работать. Обозначим эту нормативную (необходимую Вам) точность переменной eps (обычно в инженерных расчетах погрешность обозначают греческой буквой "эпсилон").

Теперь Ваша программа будет иметь следующий вид:

```
5 LET eps = 0.00001
10 FOR i = 0 TO 1 STEP 0.01
20 IF ABS(i-0.5) > eps THEN PRINT i
30 NEXT i
```

или

```
5 LET eps = 0.00001
10 FOR i = 0 TO 1 STEP 0.01
20 IF ABS(i-0.5) < eps THEN PRINT i
30 NEXT i
```

Итак, рецепт прост: "Если ваши числа представляют числа с плавающей точкой, то вместо того, чтобы сравнивать их одно с другим, надо вычислять разность сравниваемых чисел и сравнивать абсолютную величину этой разности с разумно заданной Вами величиной допустимой погрешности eps."

Подобными приемами надо пользоваться всегда, когда Вы ведете инженерные расчеты и выделяете какое-то сравнение. Это не зависит от того "Спектрум" это или "Эльбрус" или "Коммодор". Это стиль инженерных вычислений и его надо выдерживать, чтобы быть гарантированно защищенным от возможных ошибок, связанных с формами представления чисел в известной или неизвестной Вам машине.

Ну и наконец последнее замечание. Использовать в качестве параметра цикла число с плавающей точкой - это "дурной тон". На большинстве компьютеров это приводит как минимум к замедлению скорости вычислений и делать это просто не принято. Параметр цикла надо делать целым числом. И тогда наши примеры будут выглядеть так:

```
5 LET eps = 0.00001
10 FOR i=0 TO 100
15 LET a=i/100
20 IF ABS(a-0.5) > eps THEN PRINT a
30 NEXT i
```

или

```
5 LET eps = 0.00001
10 FOR i=0 TO 100
15 LET a=1/100
20 IF ABS(a-0.5) < eps THEN PRINT a
30 NEXT i
```

БЕТА-БЕЙСИК 1.8

Дополнение к инструкции БЕТА-БЕЙСИК 1.0

Окончание. Начало см. стр. 134, 100, 201.

В данной статье мы заканчиваем печать материалов по БЕТА-БЕЙСИКу версий 1.0 и 1.6.

Версия 1.8 настолько похожа на версию 1.0, что ее можно дать как дополнение к версии 1.0.

Нам еще осталось рассмотреть версию 3.0. Она имеет значительные отличия от двух предыдущих версий и будет нами рассмотрена полностью от начала и до конца. Этим мы с Вами займемся в первой половине следующего года.

Версия 1.8 языка Бета-Бейсик имеет значительно больше команд и функций, чем выпуск 1.0.

Программа теперь имеет больший размер и занимает 9.5K вместо 5.6K у версии 1.0. В этой версии нет модификации для 16K компьютеров. Значение RAMTOP в этой версии установлено в 55800 и загрузчик содержит теперь две самоуничтожающиеся Бейсик-строки (вместо одной). Строка 1 содержит процедуру SAVE для выгрузки текущей Бейсик-программы с последующим блоком машинных кодов Бета-Бейсика. Если Вы хотите воспользоваться ею, то загрузите загрузчик через "MERGE", чтобы исключить автостарт, затем остановите ленту и дайте:

```
CLEAR rt: LOAD "" CODE
```

Теперь вновь запустите ленту. Когда вторая часть будет загружена, RUN вызовет автовыгрузку программы на ленту.

Чтобы выгрузить на микродрайв, дайте RANDOMIZEUSR 58419 (это инициализация Бета-Бейсика), а затем измените строки 1 и 2 прежде, чем давать RUN:

```
1 LET rt = DPEEK (23730): RANDOMIZEUSR 59904: SAVE * "m"; 1; "run" LINE 2: POKE DPEEK  
(23631)+2, 181: SAVE * "m"; 1; "BB" CODE rt + 1, 65367 - rt: STOP  
2 CLEAR rt: LOAD * "m"; 1; "BB" CODE: CLS: RANDOMIZEUSR 58419: DELETE 1 TO 2
```

1. Команда CLOCK.

В этой версии программы часы задействуются, хотя и не изображаются на экране, уже во время загрузки Бета-Бейсика. в это же время включается усовершенствованный режим BREAK.

2. Команды управления курсором.

CHR\$ 8 (курсор влево)

CHR\$ 9 (курсор вправо)

CHR\$ 10 (курсор вниз)

CHR\$ 11 (курсор вверх)

Бета-Бейсик позволяет использовать эти символы при печати. Например, PRINT CHR\$ 10 сдвинет позицию печати на одну строку вниз. Эти же коды работают с оператором PLOT при печати на экране строк.

В стандартном Бейсике CHR\$8 (курсор влево) имеет "жучок". Сдвигая курсор назад, нет возможности войти в верхнюю строку экрана из какой-либо нижележащей строки.

А если Вы уже находитесь на верхней строке то движением назад (влево) можете вообще покинуть экран и войти в программу. В этой версии Бета-Бейсика ошибка исправлена.

CHR\$ 9 стандартного Бейсика также имеет "жучок", мешающий его работе. Здесь он также исправлен.

CHR\$ 10 и CHR\$ 11 в стандартном Бейсике печатают "?" от которого нет никакой пользы. Сейчас они работают, как положено.

Пример, приведенный ниже, показывает, как четыре управляющих символа включаются в строки, что дает возможность создания сложных форм с помощью PRINT или

PLOT.

```
10 LET a$="1234"+ CHR$ 8 + CHR$ 10 + "5" + CHR$ 8 + CHR$ 10 + "678" + CHR$ 8 + CHR$ 11 + "9"  
20 PRINT AT 10,10; a$  
30 PAUSE 100: CLS  
40 FOR n = 32 TO 255  
50 PLOT n,n/2; a$: NEXT n
```

С применением графики UDG здесь можно получать очень эффектные изображения.

3. Команда: DEF KEY

Структура: DEF KEY символ; строка

или DEF KEY символ: оператор: оператор:...

Клавиша: CAPS SHIFT + 1

Бета-Бейсик позволяет воспроизводить строковые переменные или целые строки программы при нажатии цифровой или буквенной клавиши. Причем они могут либо сразу идти в компьютер, либо оставаться в нижней части экрана до нажатия ENTER. Последний случай выполняется введением управляющего символа ":". Он должен стоять последним символом в строке или стоять после последнего выражения в строке.

Попробуйте:

```
DEF KEY "i"; "HELLO":
```

Теперь нажмите "SYMBOL SHIFT" и "SPACE" одновременно. Курсор изменится на мигающую звездочку. Если Вы нажмете "I", то в нижней части экрана появится "HELLO". Поскольку все прочие клавиши заданы не были, то при нажатии любой другой клавиши Вы получите ее нормальные значения.

```
DEF KEY "a"; "GOODBYE"
```

В этом примере строка "GOODBYE" присваивается клавише "a" (или "A", что одно и то же). пока она не выполняется, поскольку после выражения не стоит ":". Когда же Вы нажмете "SYMBOL SHIFT" + "SPACE", строка будет введена, а когда нажмете "a", она исполнится.

Вы можете с помощью нехитрого приема записывать строки в программу одним нажатием клавиши. Например, Вам надо, чтобы после нажатия клавиши "a\$ в программу вводилась строка 10 REM hello.

Вы можете задать клавишу "a" следующим образом:

```
DEF KEY "a": "10 REM hello".
```

Когда Вы попытаете это сделать, Вас может постигнуть легкое разочарование, поскольку Вы не сможете получить ключевое слово REM, т.к. курсор в этот момент будет иметь вид не "K", а "L". Как с этим бороться мы уже писали раньше в разделе "Маленькие хитрости", стр. 52. Используем для этого особенность оператора THEN. Наберем начала

```
DEF KEY "a"; "10 THEN REM hello"
```

Теперь установим курсор справа от REM и помощью DELETE сотрем THEN - Вас останется то, что надо.

Таким приемом можно на любую клавишу "подвесить" часто повторяющиеся при наборе программ строки и облегчить себе программирование.

Клавише может быть присвоено новое значение в любое время, старое будет забыто. Если использовалась пустая строка или не было задано никаких операторов после определения клавиши, то клавиша не будет иметь определения.

DEF KEY ERASE сотрет все определения клавиш, которые хранятся выше RAMTOP и защищены от стирания другими путями, в том числе и NEW. Процедура SAVE в загрузчике Бета-Бейсика запишет все определения клавиш на ленту вместе с программой. (Запись идет от RAMTOP до конца Бета-Бейсика).

Для тех, кто программирует в машинных кодах, возможно, будет интересна следующая информация.

RAMTOP автоматически понижается для того, чтобы выделить место для занесения определений клавиш. Если Вы сами используете CLEAR (число) для изменения RAMTOP, то это может помешать воспользоваться заданными клавишами.

Эту проблему можно обойти, если сделать так, чтобы при переустановке RAMTOP синхронно перемещалась бы и область определения клавиш. Нижеследующая процедура сдвигает RAMTOP и определения клавиш вниз на заданное количество шагов, чтобы

избежать этой проблемы. Между концом области определения клавиш и старым RAMTOP, т.е. 55800 создается зазор, при желании его можно изменить.

space - размер памяти, на которую понижается RAMTOP;

oldrt - старый адрес RAMTOP;

newrt - новый адрес RAMTOP.

В процедуре используются функции MEMORY\$() и STRING\$(), которые мы еще не рассматривали и о которых мы сообщим ниже.

```
10 INPUT "Spaces?"; space
20 LET oldrt = DPEEK (23730)
30 DPOKE 23728,oldrt
40 CLEAR (oldrt - space)
50 LET oldrt = DPEEK (23726)
60 LET a$ = MEMORY$ ( ) (oldrt TO 55600)
70 LET newrt = DPEEK (23730)
80 LET space = oldrt - newrt
90 POKE newrt,a$
100 POKE (55801 - space),STRING$ (space,CHR$ 0)
```

Всегда оставляйте нулевой байт после последнего определения клавиш в качестве "маркера" конца.

4. Команда: FILL

Структура: FILL x,y или FILL // INK цвет//; x,y

или FILL //PAPER цвет //; x,y

Клавиша: F

По этой команде происходит заполнение области PAPER цветом INK, если была выбрана команда FILL или FILL INK или происходит заполнение области INK цветом PAPER, если была избрана команда FILL PAPER. Заполнение начинается с точки, имеющей координаты x,y. Если Вы попытаете заполнить область цветом INK, а точка x,y уже имеет цвет INK, то ничего не получится. В отличие от стандартного Бейсика, если Вы опустите номер цвета, то команда FILL сработает с текущим цветом INK или PAPER.

```
10 FOR N = 1 TO 6
20 CLS
30 CIRCLE INK N; 128, 88, N*10
40 FILL INK N; 128,88
50 NEXT N
```

Возможно использование более сложных форм FILL:

```
FILL INK 2; PAPER 1; FLASH 1; x,y
```

В этом случае первое слово после FILL указывает какой цвет INK или PAPER Вам нужен, а остальные изменяют атрибуты заполняемой области.

Поскольку количество цветов, допустимых для одного знакоместа, ограничено двумя, то необходимо тщательно продумывать условия стыковки двух областей с разным цветом INK. Необходимо, чтобы стык проходил по границам знакомест.

FILL будет работать с любыми формами. Попробуйте этот пример, который заполнит весь экран, за исключением областей внутри букв.

```
PRINT STRING$(704,"Q"):FILL 0,0
```

Метод работает быстрее, если имеется большой объем доступной памяти. Если Вы попытаете заполнить большую пустую область, например весь экран, Вы увидите паузу в те моменты, когда FILL производит проверку хранимых данных в поисках того, что можно без ущерба вычистить.

Количество пикселей, заполненных по FILL, можно определить с помощью функции FILLED ().

FILL можно прервать в любой момент с помощью BREAK.

5. Команда: JOIN

Структура: JOIN //номер строки//

Клавиша: SYM SHIFT + G (см. также SPLIT.)

По этой команде объединяются вместе две строки. Если номер не указан, то

объединяется строка, около которой стоит указатель, со следующей за ней строкой, если она есть.

Вторая строка теряет свой номер и присоединяется к предыдущей через разделитель ":",

6. Команда: KEYIN

Структура: KEYIN строка.

Клавиша: SYM.SHIFT + 4

По этой команде вводится строка так, как если бы она была впечатана с клавиатуры. Это позволяет программам самозаписываться.

Полный список возможных приложений этой функции выходит за пределы, рассматриваемые фирменной инструкцией по работе с Бета-Бейсиком. Очевидно, ее возможности столь широки и необычны, что требуют специального исследования. Указано только, что наиболее очевидным приложением является возможность автоматического написания в листинге программы строк DATA (очень утомительное занятие для программиста, и если у него есть массив данных в памяти машины, имеет смысл подготовить специализированный модуль, занимающийся тем, чтобы формировать строки DATA из массива.)

Пример

```
10 LET a$="100 DATA": REM применяем ключевое слово DATA. Как это делать через THEN мы писали
    выше
20 FOR N = 0 TO 9
30 LET A$=A$ + STR$(PEEK N) + ", "
40 NEXT N
50 LET A$=A$(1 TO LEN(A$)-1): REM отсечение последней запятой
60 KEYIN A$
```

Вы увидите, что после RUN к программе будет добавлена еще одна строка. (Строка совершенно бесполезная, поскольку дает данные первых 10 байтов памяти ПЗУ, но зато иллюстрирует сам принцип).

Если Вы чувствуете в себе смелость, можете использовать SCREEN\$ и KEYIN для того, чтобы с их помощью написать полноэкранный редактор Бейсика!!!

Примечание: Он может быть довольно медленным.

Примечание 2: KEYIN не может быть непосредственной (прямой командой), а только частью программы.

7. Команда: LIST

Структура: LIST номер строки TO номер строки

или LLIST номер строки TO номер строки

Первый или второй номер строки может быть опущен. Если опущен первый, то он принимается за 0, если второй, то предполагается номер последней строки.

LIST 20 TO 100

LIST TO 200

LLIST 100 TO 180

Если оба номера совпадают, то распечатывается только одна строка.

Очень эффективно можно эту команду использовать совместно с TRACE.

```
9000 LIST line TO line: RETURN
```

Эта строка распечатывает каждую строку по мере ее исполнения после того, как Вы вставите TRACE 9000 в программу (в данной примере "line" - это переменная, создаваемая процедурой TRACE).

К сожалению, каждая строка будет иметь при себе мигающий курсор.

8. Команда POKE

Структура: POKE адрес, строка

Это обычное ключевое слово.

Бета-Бейсик 1.8 разрешает выполнять POKE с символьными строками так же, как и с числами, что в сочетании с функцией MEMORY\$ дает возможность быстрых манипуляций

большими блоками памяти. (Здесь надо отметить, что если Ваша программа может выйти из строя в результате нерасчетливого POKE, то при работе с длинными символьными строками это становится много более вероятным).

Давайте посмотрим на один эффект:

```
10 LET screen = 16384
20 POKE screen, STRING$(6144, "U").
```

Функция STRING\$ была рассмотрена в инструкции к версии 1.0 под именем FN S\$.

По этой программе экран заполняется "U", но поскольку вследствие своего положения это уже не "U", а код от "U", т.е. 85, а в двоичной форме "01010101", то экран заполняется полосами.

```
30 LET attributes = 22528
40 POKE attributes, MEMORY$(1 TO 704)
```

Эта программа копирует начало области памяти ROM в файл атрибутов, где эта операция производит интересный эффект.

Теперь давайте запишем простую программу, которая создаст на экране рисунок, сохранит его в памяти как символьную строку и затем через POKE вернет его на экран.

```
10 CIRCLE 128,88,70
20 FILL 128,83
30 LET a$=MEMORY$(16384 TO 23255):
40 CLS: PRINT "HIT ANY KEY":PAUSE 0
50 POKE 16384,a$
```

Память компьютера может содержать несколько таких рисунков, что даст вам возможность заменять их и демонстрировать последовательно. Чтобы можно было разместить в памяти большее количество рисунков, удобно считывать в строковую переменную третью часть экрана, туда же Вы можете добавить и третью часть файла атрибутов. Чтобы сохранить в памяти три трети экрана в виде строк, пользуйтесь:

Первая треть экрана.

```
LET a$=MEMORY$(16384 TO 16431)
```

Вторая треть экрана.

```
LET a$=MEMORY$(18432 TO 20479)
```

Нижняя треть экрана.

```
LET a$=MEMORY$(20480 TO 22527)
```

Памяти достаточно, чтобы создать удовлетворительный мультфильм с использованием последовательных POKE строк. Вы можете применять массив (например DIM a\$(10,2048)) чтобы хранить эти данные.

Конечно, существует значительно больше потенциальных возможностей, кроме развлечений с экранной памятью. Можно очистить (CLEAR) большую область памяти и загнать туда на хранение целую программу:

```
CLEAR 33900,
```

затем запустите (RUN) следующую программу:

```
10 POKE 34000, MEMORY$(33552 TO 33800)
20 ... остальная часть программы
```

Теперь Вы можете дать NEW и стереть свою программу, а затем, когда она Вам вновь понадобится, вернуть ее назад:

```
POKE 23552, MEMORY$(34000 TO 44246)
```

Эту команду можно "спрятать" от NEW на клавише заданной пользователем. Мы уже говорили, что задания клавиш после NEW не уничтожаются, т.к. хранятся выше уровня, установленного RAMTOP. Клавиша задается после CLEAR и до RUN, например:

```
DEF KEY "J": POKF 23552, MEMORY$(34000 TO 44248).
```

После того, как программа будет возвращена, она продолжит работу с того места, в котором она была "спрятана", т.к. системные переменные были сохранены вместе с Бейсиком, а в них запоминается состояние программы во время ее работы.

Немного труднее организовать хранение двух программ, нужно больше памяти, нужно также предоставить место в БЕЙСИК-области для их запуска, но это тоже возможно.

И, дополнительно, еще одна "маленькая хитрость". Чтобы выгрузить на ленту блок в машинных кодах с Бейсиковской программой, присвойте его символьной строке с использованием MEMORY\$, а затем сделайте SAVE для Бейсик-программы, включив в нее и

эту строковую переменную. При этом предусмотрите, чтобы после автостарта POKE возвращал бы этот блок на его место при загрузке БЕЙСИКа.

Примечание: Если хотите избежать затирания каких-то важных данных после такого POKE, не забывайте предварительно выставлять CLEAR. Еще полезный совет; POKE машинного кода совершенно безопасно производится в область экрана (хотя это некрасиво выглядит) или в область буфера принтера (если Ваш код не более 256 байтов и если Вы не работаете со 128-ой моделью).

9. Команды: ROLL и SCROLL

Эти команды могут иметь довольно сложный синтаксис. Мы настойчиво рекомендуем изучить сначала их работу по инструкции к Бета-Бейсику 1.0.

В этой версии ROLL может иметь структуру:

ROLL код направления //,пиксели// //; x,y; ширина, длина//

Коды направления 1...4 перемещают только атрибуты, коды 5...8 - только пиксельную информацию, коды 9...12 - и то и другое вместе.

Поскольку атрибуты могут перемещаться только на величину знакоместа 8x8, то целесообразно при совместном перемещении двигать пиксельную информацию на 8 пикселей за шаг.

Параметр "пиксели" можно задать самому. По умолчанию он принимается равным 1. Его значение не должно превышать 255 для горизонтального перемещения и 176 для вертикального.

ROLL 5 или ROLL 5,1 передвинет экран влево на один пиксель, а ROLL 5,2 - на два пикселя. При вертикальном ROLL или SCROLL скорость перемещения примерно пропорциональна количеству пикселей, сдвигаемых за один раз. В горизонтальном направлении наилучшую скорость дает перемещение на 4 пикселя или 8, т.к. при этом процессор Z-80 применяет инструкции сдвига на байт или полубайт (ниббл).

10. Команда SPLIT

Это не ключевое слово. В действительности вводится символ <>.

Клавиша: SYM.SHIFT + W (не в режиме "G").

Команда служит для деления строк программы на несколько частей.

Если в строку, содержащую несколько операторов, при редактировании ввести символ "<>" в качестве первого символа какой-либо инструкции, то после ENTER в программу эта строка пойдет только до знака "<>", а оставшаяся часть строки останется в нижней части экрана в области редактирования.

Теперь знак "<>" можно убрать и он заменится тем же номером строки. Курсор стоит справа от номера строки в положении, которое позволяет вам легко изменить номер на новый прежде чем нажмете "ENTER".

Если Вы введете:

```
10 PRINT "hello": GO TO 10: <> PRINT "goodbye"
```

и нажмете ENTER, то в листинге появится:

```
10 PRINT "hello": GO TO 10
```

а в нижней части экрана останется:

```
10 (курсор) PRINT "goodbye"
```

ФУНКЦИИ

В новой версии Бета-Бейсика 1.8 по сравнению с версией 1.0 добавлено много новых функций. Ниже в таблице они приведены как бы в виде ключевых слов, но таковыми не являются, а набираются через FN, с последующей буквой и знаком \$ или скобкой (.

Функция	Клавиша	Версия
AND	FN A(1.8
BIN\$	FN B\$	1.8
CHAR\$	FN C\$	1.0
COSE	FN C	1.8
DEC	FN D(1.0

DPEEK	FN P(1.0
FILLED	FN F(1.8
HEX\$	FN H\$	1.0
INSTRING	FN I(1.0
MEM	FN M(1.0
MEMORY\$	FN M\$	1.8
MOD	FN V(1.8
NUMBER	FN N(1.0
OR	FN O(1.8
RNDM	FN R(1.8
SCRN	FN K\$	1.8
SINE	FN S(1.8
STRING\$	FN S\$	1.0
TIME\$	FN T\$	1.0
USING\$	FN U\$	1.0
XOR	FN X(1.8

Те функции, которые вошли в описание версии Бета-Бейсик 1.0, здесь не рассматриваются.

1. Функция: AND

Структура: AND (число, число)

Клавиши: FN A (число, число).

Эта функция записывается как обычное ключевой слово, но отличается от стандартного в листинге программы своим синтаксисом. Оно выдает результат побитной операции логического AND двух чисел, которые должны быть в диапазоне от 0 до 65535. каждый бит будет равен "1" только если соответствующий бит и в первом и во втором числе равнялся единице. Пониманию этого очень способствует новая функция BIN\$.

```
BIN$(254)="1111 1110"
```

```
BIN$(120)="0111 1000"
```

```
BIN$(AND(254,120))="01111000"
```

Вы можете использовать AND для того, чтобы "маскировать" нежелаемые биты.

Например:

```
PRINT AND (BIN 00000111,ATTR(line,column))
```

Эта строка дает цвет INK для позиции "line,column", путем маскировки остальных битов атрибутов. Вы могли воспользоваться числом "7" вместо "BIN 0000 0111".

В нижеследующем примере программа напишет слово "Bang", если была нажата клавиша F, причем возможно ее сочетание с любыми клавишами (см. т. 1 нашего трехтомника по программированию в машинных кодах, где каждая клавиша рассматривается в качестве серии внешних портов).

```
10 IF AND (BIN 0000 1000, IN 65022) 0 THEN PRINT "bang!";
```

```
20 GO TO 10
```

2. Функция BIN\$

Структура: BIN\$ (число)

Клавиша: FN B\$ (число)

Дает двоичный эквивалент числа R в качестве восьми-символьной строки. Если число меньше, чем 256 или в качестве шестнадцати-символьной строки, если число лежит между 256 и 65535.

Эта функция полезна для понимания машинных кодов и операций с битами для функций AND, OR и XOR.

Она может быть также полезной при проверке генератора символов из ПЗУ, области графики пользователя, области атрибутов, системных переменных и клавиш клавиатуры.

```
10 PRINT AT 10,10; BIN$(IN 65022):GO TO 10
```

Если Вам хочется иметь в строке какие-то символы, отличные от "0" и "1", то дайте POKE 62865 или 62869 с желаемым символом.

3. Функция: COSE

Структура: COSE (число)

Клавиши: FN C (число)

Функция выдает косинус "числа" с меньшей точностью, чем стандартная - точность до 4-х значащих цифр, но в шесть раз быстрее.

4. Функция: FILLED

Структура: FILLED()

Клавиша: FN F()

Функция дает количество пикселей, заполненных последней командой FILL.

Например:

```
10 PLOT 0,0: DRAW 9,0: DRAW 0,9
20 DRAW -9,0: DRAW 0,-9
30 FILL 5,5
40 PRINT FILLED ( )
```

Одна сторона квадрата - 10 пикселей. (Заметьте, что если бы мы вместо 9 использовали бы 1 в функции DRAW, то сторона квадрата равнялась бы двум пикселям). Это потому, что реальные линии имеют на компьютере толщину в один пиксель. Внутренняя сторона построенного квадрата имеет 8 пикселей, поэтому FILLED даст нам 64.

Если бы мы дали FILL PAPER:5,5 то квадрат удалился бы с экрана и функция FILLED дала бы 100.

Разница между 100 и 64, равная 36 - это количество пикселей, составляющих периметр.

5. Функция: MEMORY\$

Структура: MEMORY\$ ()

Клавиши: FN M\$()

Выдает всю память в качестве символьной строки. На самом деле сюда не включается первый байт компьютера (адрес 0), поэтому MEMORY\$() (1)- это тоже самое, что и PEEK 1. По техническим соображениям сюда не входят также три последних байта памяти, поэтому результат работы функции на самом деле имеет длину LEN 65532.

Совместно со способностью Бета-Бейсика делать POKE для символьных строк, эта функция дает программисту возможность перемещать большие области памяти с высокой скоростью. Для более полного описания этого аспекта см. POKE.

Другое приложение MEMORY\$ состоит в том, что она позволяет делать быстрый поиск в памяти, используя функцию INSTRING.

Несмотря на то, что область памяти, в которой производится поиск, можно ограничить например вырезкой MEMORY\$(23759 TO)).

INSTRING работает так быстро, что как правило это не имеет смысла делать.

```
10 REM ASDFG
20 PRINT INSTRING (1, MEMORY$( ), "ASDFG")
```

Эта программа отыщет тот адрес, где в REM содержится строка "ASDFG". Вместо 1 мы могли бы поставить DPEEK(23635), где содержится переменная PROG, указывающая на адрес начала Бейсик-программы. Тогда поиск проводился бы от начала программы, а не от начала ПЗУ.

Поскольку Бета-Бейсик позволяет выполнять POKE для строк, то поиск строк и их замена выполняются очень просто, если Вы хорошо представляете, что Вы делаете. Может быть Вы не пожелаете, чтобы программа допускала замену строки другой, если та имеет большую длину.

6. Функция MOD

Структура: MOD (число,число).

Клавиши: FN V (число,число).

Функция дает остаток деления одного числа на другое.

```
MOD(10, 3) = 1
MOD(66, 16) = 2
```

$\text{MOD}(125, 35, 5) = 18.5$

Ниже приведен пример программы, предотвращающей печать (PLOT) за пределами экрана.

```
10 FOR n=0 TO 400
20 PLOT MOD(n, 256), MOD(n, 176)
30 NEXT n
```

7. Функция: OR

Структура: OR (число, число).

Клавиши: FN O (число, число).

Эта функция записывается как обычное ключевое слово, но отличается от стандартного OR другим синтаксисом. Она выполняет логическую побитную операцию OR для двух чисел, которые должны быть в диапазоне от 0 до 65535. Если бит равен 1 в первом числе или во втором, то соответствующий бит результата тоже будет равен 1. Чтобы в результате бит равнялся нулю, он должен быть равным нулю в обоих числах.

8. Функция: RNDM

Структура: RNDM (число).

Клавиши: FN R (число).

Если число равно "0", то RNDM дает случайное число от 0 до 1, как и RND. Однако она работает в два с половиной раза быстрее. Если число не равно нулю, то функция дает случайное число в диапазоне от 0 до этого числа. Это также выполняется в два с половиной раза быстрее, чем $\text{RND} * (\text{число})$.

```
10 PLOT RNDM(255), RNDM(175)
20 GO TO 10
```

RANDOMISE (число) устанавливает RNDM в определенное место в последовательности случайных чисел так же, как и для RND.

9. Функция: SCRN\$

Структура: SCRN\$ (ряд, столбец)

Клавиши: FN K\$

Работает почти как обычная функция SCREEN\$, за исключением того, что распознает символы графики, как обычные символы. "Жучок", который содержится в функции SCREEN\$ "Спектрума", здесь также устранен.

Сначала введите KEYWORDS 0, затем попробуйте пример, приведенный ниже. Он создает символы графики пользователя со случайным рисунком, а затем читает их с экрана.

```
10 FOR a=USR "a" TO USR "u"+7
20 POKE a, RND*255:
30 NEXT a
50 LET a$=""
60 FOR c=0 TO 31
70 LET a$=a$+SCRN$(0, c)
80 NEXT c
90 PRINT a$
```

"Спектрумовская" блочная графика не понимается. Если Вам это нужно, запрограммируйте символы графики пользователя так, чтобы они выглядели, как блочная графика.

10. Функция: SINE

Структура: SINE (число).

Клавиши: FN S (число).

Дает синус "числа" с меньшей точностью, чем стандартная функция, хотя 4 значащих цифры есть. Зато работает в шесть раз быстрее.

11. Функция: XOR

Структура: XOR (число, число).

Клавиши: FN X (число, число). Эта функция выполняет XOR (исключающее "или") для двух чисел. Числа должны быть в диапазоне от 0 до 65535. Если бит равен "0" или "1" в обоих числах, то он будет равен "0" в результате. Если бит равен "1" только в одном из чисел, то он будет равен "1" в результате.

КАНАЛЫ И ПОТОКИ

Сегодня в номере нет традиционного раздела "Секреты ПЗУ". Причина проста - сейчас мы рассматриваем тот раздел, который очень широко оперирует с такими образованиями, как потоки и каналы и мы уже не можем двигаться дальше, не рассказав Вам о том, что это такое, ведь раньше в своих изданиях мы на этом вопросе не останавливались.

Мы обещали Вам, что в начале следующего года дадим широкую статью на эту тему, но время идет, планы меняются и в связи с тем, что материал по ПЗУ данного выпуска обязательно требует предварительного обсуждения потоков и каналов, мы пропустили эту статью вперед и даем ее всю целиком, в одном номере, без продолжений, чтобы не ущемить интересы тех немногих читателей, кто не подписался на будущий год.

В двух словах: каналы и потоки позволяют скрыть от пользователя сложную логику работы программного обеспечения и сделать для него возможность простыми командами выполнять сложные действия.

Это не более чем еще один прием организации программ, еще один вклад в копилку Ваших знаний о компьютере "Спектрум".

Для многих начинающих пользователей "СПЕКТРУМа" такие понятия как "каналы" и "потоки" могут звучать как некоторые непонятные жаргонные обозначения, но на самом деле за ними скрывается интересная концепция, которая может позволить Вам взять от компьютера то, что другими путями сделать непросто.

Итак, как всегда, начнем с самого простого. Вы начинаете использовать каналы и потоки уже тогда, когда даете команды PPINT или INPUT.

Так, PRINT на самом деле обозначает PRINT #2, хотя #2, как правило опускают. Точно так же INPUT на самом деле INPUT #0. Кстати и LPRINT на самом деле то же самое, что и PRINT #3.

Число, которое стоит после символа # является номером потока.

Не любое число может быть использовано в качестве номера потока. Так, если Вы захотите после включения "Спектрума" напечатать что-то по шестому потоку и дадите команду PRINT №6, то получите сообщение об ошибке "0: Invalid stream" ("ошибочно задан поток"). Это происходит потому, что поток номер шесть еще не подключен ни к какому каналу (потоки 0...3 подключены стандартно, здесь Ваше участие не требуется, этим занимаются программы, "зашитые" в ПЗУ), поэтому прежде чем двигаться дальше, давайте разберемся с каналами.

Листинг 1.

37	CLOSE_NEW	ORG B000 SCF	; Сигнал на выгрузку дан- ных из буфера.
F5	CLOSE_CL	PUSH AF	; Сохранение номера потока.
06		EX AF,A'F'	; Для сохранения флага C.
CD2117		CALL 1721,STR_DATA_A	; Это обращение во вторую половину процедуры STR_DATA-1 (171E). На выходе HL содержит базо- вый адрес таблицы STRMS, а BC - адрес данных для данного потока.
78		LD A,B	; Проверка
B1		OR C	; BC на ноль.
CB		RET Z	; Возврат, если поток уже закрыт.
E5		PUSH HL	; Сохранение на стеке адреса таблицы STRMS.

21E2A3	LD HL,A3E2	;Число A3E2 взято потому, ;что если к нему прибавить ;5C1E (адрес первого ;пользовательского потока ;в STRMS, то возникнет ;переполнение и включится ;флаг переноса).
09	ADD HL,BC	;Проверка на переполнение.
E1	POP HL	;Возврат адреса STRMS со ;стека.
D0	RET NC	;Возврат, если не включен ;флаг переноса, следова- ;тельно переполнения не ;было и поток не пользова- ;тельский.
DD2A4F5C	LD IX,(CHANS)	;Установка в IX начала об- ;ласти C.I.A.
DD09	ADD IX,BC	;Установка адреса блока
DD2B	DEC IX	;информации о канале.
DD7E05	LD A,(IX+05)	;Ввод идентификатора ;пользовательского канала.
FE34	CP 34	;Возврат, если канал не
C0	RET NZ	;пользовательский;
DD7E06	LD A,(IX+06)	;Ввод идентификатора ;пользовательского канала.
FE12	CP 12	;Возврат, если канал не
C0	RET NZ	;пользовательский;
3600	LD (HL),00	;Обнуление данных
23	INC HL	;по потоку
3600	LD (HL),00	;в таблице STRMS.
C5	PUSH BC	
DD6E07	LD L,(IX+07)	;В HL устанавливается ад-
DD6608	LD H,(IX+08)	;рес процедуры, выгружаю- ;щей буфер.
08	EX AF,A'F''	;Вызов флага C.
DC2C16	CALL C,162C,CALL JUMP	;Выгрузка буфера.
DDE5	PUSH IX	;Переброска из
E1	POP HL	;IX в HL.
DD4E09	LD C,(IX+09)	;В регистре BC установ-
DD460A	LD B,(IX+0A)	;ливается длина блока.
C5	PUSH BC	
CDE819	CALL 19E8,RECLAIM_2	;Удаление блока из памяти.
3E10	LD A,10	;10 - количество потоков.
21165C	LD HL,5C16	;5C16 - адрес STRMS 00.
5E	LD E,(HL)	;Теперь HL указывает на
23	INC HL	;данные по только что пе-
56	LD D,(HL)	;ремещенному потоку, а DE
E3	EX (SP),HL	; - по следующему.
A7	AND A	;Сброс флага переноса.
ED42	SBC HL,DF	;Проверка передвигался ли
19	ADD HL,DE	;блок после стирания.
300B	JR NC,CLOSE NEXT	;Если не передвигался и в ;его STRMS_n изменения не ;вносятся - переход.
EB	EX DE,HL	;Нет операции SRC DE,BC, ;потому переброс в HL.
A7	AND A	;Сброс флага переноса.
ED42	SBC HL,BC	;Теперь в HL содержится ;измененное значение ука- ;зателя STRMS_n.
EB	EX DE,HL	;Перевод его в DE.
E3	EX (SP),HL	;Установка
2B	DEC HL	;нового
73	LD (HL),E	;значения
23	INC HL	;указателя

72		LD (HL), D	; в таблице STRMS.
E3		EX (SP), HL	
E3	CLOSE_NEXT	EX (SP), HL	
23		INC HL	; HL указывает на следующий
			; указатель.
3D		DEC A	; Уменьшение параметра
			; цикла.
20E6		JR NZ, CLOSE_LOOP	; Возврат к началу цикла.
F1		POP AF	; Балансировка стека.
C9		RET	; Возврат.

Можете представить себе канал в качестве некоего устройства, которое используется для ввода или вывода информации. Так, например Ваш телевизор - это канал, поскольку на его экране можно печатать символы.

Надо, правда оговориться, что канал - это не всегда техническое устройство. Каналом может быть например файл на диске (или в памяти компьютера). В файл ведь тоже можно заносить информацию, можно ее оттуда и принимать. Если Вы выполните печать чего-то в файл, то ни на экране, ни на принтере Вы результатов этой печати не увидите, но впоследствии, когда будете просматривать этот файл, увидите, что информация туда вошла, то есть произошла печать (вывод).

Еще лучше представить концепцию каналов и потоков на примере морской бухты. Представьте себе побережье с многочисленными заливами и бухтами. Со стороны суши в них впадают реки, речки и ручьи. Так вот заливы и бухта - это те самые КАНАЛЫ, а ручьи и речки, впадающие в них, - это ПОТОКИ, подключенные к КАНАЛАМ. Их можно и переподключить. Если Вы постройте на ручье плотину, образуется водохранилище, уровень воды поднимется и Вы сможете отвести ручей (ПОТОК) в другой залив (КАНАЛ).

Эта аналогия хороша тем, что помогает преодолеть имеющуюся в русском языке небольшую мнемоническую путаницу. Дело в том, что мы обычно под словом "канал" понимаем что-то узкое, длинное, по чему перемещаются какие-то суда, грузы и течет вода. В общем, отождествляем канал с потоком. К сожалению это неверно. "Канал" в компьютере - не средство транспортировки данных - это именно залив, бухта, (экран, принтер, файл на диске, участок в области памяти и т.п.) в которые вливаются потоки данных (ручьи).

В "Спектруме" каждый канал имеет имя, которое выражено одной буквой алфавита. Так, экран дисплея - это канал "S" потому, что по-английски Screen - экран.

Оператор OPEN служит для того, чтобы подключить поток к каналу (канал к потоку). Так, Вы можете в БЕЙСИКе дать команду OPEN#6,"S" и тем самым подключите шестой поток к экрану и тогда команда PRINT# 6 будет печатать Ваш текст на экране точно так же, как это делает обычная команда PRINT.

Листинг 2.

		ORG B061	
3E10	CLEAR_NEW	LD A, 10	
3D	CLEAR_LOOP	DEC A	
F5		PUSH AF	
A7		AND A	; Выключение флага C - сиг-
			; нал о том, что данные в
			; буферах уничтожаются.
CD01B0		CALL B001, CLOSE_CL	; Удаление канала, подклю-
			; ченного к данному потоку.
F1		POP AF	
20F7		JR NZ, CLEAR_LOOP	; Возврат для повтора.
C9		RET	

Листинг 4.

Вектор переходов для ПЗУ "ZX-Spectrum-128".

		ORG B6FC	
C3AC05	V_ERROR	JP 05AC	; Генерация сообщений об ошибках.

C3641C	V_PAGE	JP 1C64	; Переключение текущей страницы ОЗУ.
C3971D	V_NEWCAT	JP 1C97	; Создание новой записи в каталоге.
C3F31C	V_SPACE	JP 1CF3	; Проверка на достаточность ; памяти RAM-диска.
C3121D	V_FIND	JP 1D12	; Поиск имени файла в каталоге.
C3561D	V_CATEND	JP 1D56	; Оформление последней ; записи каталога.

Листинг 5.

Вектор переходов для ПЗУ "ZX-Spectrum+2"

		ORG B6FC	
C3CB05	V_ERROR	JP 05CB	; Генерация сообщений об ошибках.
C3631C	V_PAGE	JP 1C83	; Переключение текущей страницы ОЗУ.
C3B61C	V_NEWCAT	JP 1CB6	; Создание новой записи в каталоге.
C3121D	V_SPACE	JP 1D12	; Проверка на достаточность ; памяти RAM-диска.
C3311D	V_FIND	JP 1D31	; Поиск имени файла в каталоге.
C3751D	V_CATEND	JP 1D75	; Оформление последней ; записи каталога.

Аналогично Вы можете представить, что клавиатура - это устройство, предназначенное для ввода информации и потому это тоже канал. Он имеет имя "K" (от слова Keyboard = клавиатура). К этому каналу можно подключить поток точно так же, как мы это делали с экраном. Можете дать команду OPEN #n,"K" и подключить поток n к каналу "K".

Всего Вы можете иметь не более 16 потоков, пронумерованных от 0 до 15. Шестнадцатого потока не существует и, если Вы попытаетесь его использовать, то получите сообщение об ошибке.

Мы уже сказали о том, что потоки от 0 до 3 являются стандартными и организованы без нашего участия. Они стандартно подключены к стандартным каналам.

Потоки 0 и 1 подключены к каналу "K", поток 2 - к каналу "S", а поток 3 - к каналу "P". Канал "P" - принтер (Printer).

Каналы "S" и "P" предназначены только для вывода (ручей может только впадать в залив), поэтому например PRINT #2 или PRINT #3 возможны, а INPUT #2 или INPUT #3 - невозможны.

В отличие от них канал "K" может использоваться и для ввода и для вывода (из этого озера река может вытекать, как Нева из Ладоги).

INPUT #0 - это самая обычная команда INPUT.

Возможен и вывод:

PRINT #0 обеспечит печать Вашего сообщения в нижних двух строках экрана, которые выполняют роль "системного окна". Это те самые две строки, в которых появляется информация при работе команды INPUT.

В "родном" "Спектруме-48" есть еще один стандартный канал - "R", но из Бейсика он не достижим и мы пока его отставим, вернемся к нему позже.

Прочие каналы.

После подключения дополнительной периферии, располагающей своим ПЗУ, к стандартным каналам могут добавляться дополнительные. Так, например, подключение интерфейса-1 ("ZX-Interface One") создает несколько новых каналов:

- "M" - канал микродрайва;
- "N" - канал локальной сети;
- "T" - канал принтера для печати листинга программ (коды выше 165 интерпретируются как токены ключевых слов "Спектрума");

- "B" - канал принтера для печати данных (все коды интерпретируются по значению).

Вот практически и все каналы, возможные для "Спектрума" на стандартном оборудовании. Но это не значит, что у Вас нет дополнительных возможностей. Вы можете создавать свои каналы в оперативной памяти и эффективно использовать их. Этим мы с Вами и займемся.

Процедура служит для открывания нового пользовательского канала. Это выполняется путем создания блока информации о канале. При входе в эту процедуру регистр A'(альтернативный) должен содержать номер назначенного потока для данного канала, регистр A(основной) - код имени данного канала (код ASCII), BC - длину блока информации о канале, DE - адрес процедуры ввода (INPUT), HL - адрес процедуры вывода (PKINTJ, а регистр IX - адрес процедуры, используемой для выгрузки данных из буфера или адрес 0052 HEX, если работа с буфером не нужна.

		ORG B06D	
C5	OPEN_NEW	PUSH BC	;Запоминаем на стеке
DDE5		PUSH IX	;входные значения
F5		PUSH AF	;регистров процессора.
D5		PUSH DE	
E5		PUSH HL	
C5		PUSH BC	
08		EX AF,A'F'	;A-номер потока.
CD2117		CALL 1721,STR_DATA_A	;Загрузка в BC данных по ;этому потоку из системной ;таблицы STRMS.
78		LD A,B	
B1		OR C	;Проверка адреса на ноль.
C1		POP BC	
2602		JR Z,OPEN_NEW_2	;Переход, если поток не ;подключен.
CF17		RST 08	;Вызов процедуры обработки
		DEFB 17	;ошибок с кодом перехвата
			;17, что означает:
			; "0: invalid stream" ("Не-
			;верно задан поток").
E5	OPEN_NEW_2	PUSH HL	
2A535C		LD HL,(PROG)	; (PROG) - начало БЕЙСИКа.
2B		DEC HL	;Теперь HL указывает на
			;байт, содержащий маркер
			;80H и отмечающий конец
			;области информации о каналах.
CD5516		CALL 1655,MAKE_ROOM	;Выделение места под новый
23		INC HL	;блок информации о канале.
			;Теперь HL указывает на
			;начало нового информаци-
			;онного блока.
22515C		LD (CURCHL),HL	;Канал делается текущим и
E5		PUSH HL	;адрес начала его информа-
DDE1		POP IX	;ционного блока переводит-
			;ся в IX.
23		INC HL	;HL - указывает на 2-ой
			;байт блока.
ED4B4F5C		LD BC,(CHANS)	;BC указывает на начало
			;области CHANS.
A7		AND A	;Обнуление флага C как по-
			;дготовка к операции SBC.
ED42		SBC HL, BC	;Вычисленное значение
			;является тем адресом, ко-
			;торый должен быть занесен
			;в таблицу данных по
			;потокам STRMS.
EB		EX DE,HL	;Теперь оно - в DE,
E1		POP HL	;HL содержит адрес требу-
			;емой STRMS_n.
73		LD (HL),E	;Отправляем данные
23		INC HL	;по потоку на свое место
72		LD(HL),D	;в STRMS_n.
DDE5		PUSH IX	

E1	POP HL	; HL-адрес блока информации
		; о канале.
D1	POP DE	; DE - адрес процедуры
		; вывода.
C1	POP BC	; BC -адрес процедуры ввода.
CDAEBO	CALL BOAE, OPEN_STORE	; Сохранение этой инфор-
		; мации в блоке информа-
		; ции о канале.
F1	POP AF	; A содержит имя канала.
77	LD (HL), A	; Записали его в блок.
23	INC HL	; Напомним: 1234 - сигнал
3634	LD (HL), 34	; о том, что данный канал -
23	INC HL	; пользовательский, а не
3612	LD (HL), 12	; стандартный.
23	INC HL	;
D1	POP DE	; DE-адрес процедуры, за-
		; крывающей буфер.
C1	POP BC	; BC-длина блока информации
		; о канале.
73	LD (HL), E	; Помещение в блок информа-
23	INC HL	; ции о канале адреса
72	LD (HL), D	; закрывающей процедуры.
23	INC HL	
71	LD (HL), C	; Помещение в блок информа-
23	INC HL	; ции о канале длины блока
70	LD (HL), B	; информации.
23	INC HL	
C9	RET	; Возврат в вызывающую
		; процедуру.

Область информации о каналах

Итак, как же все это работает? Вся концепция потоков и каналов базируется на области памяти, называемой "область информации о канале" (CHANNEL INFORMATION AREA).

Эта область памяти расположена непосредственно перед БЕЙСИК-областью, чуть ниже нее. То есть лежит между системными переменными и БЕЙСИКом. Начинается она с адреса, хранимого в системной переменной CHANS (23631, 2 байта)(5C4FH) и заканчивается байтом, в котором стоит маркер 80H. Далее уже идет Ваша БЕЙСИК-программа, на которую указывает PROG (23635, 2 байта)(5C53H).

Для того, чтобы создать свой канал Вам практически надо переорганизовать данные в этой области, может быть и раздвинуть эту область, сдвинув вверх маркер и поменяв значение PROG и разместить где-либо в памяти две процедуры. Одну - для обеспечения ввода в Ваш канал (INPUT) и вторую - для обеспечения вывода (PRINT).

Каждый канал должен иметь блок информации о канале (CHANNEL INFORMATION BLOCK). Этот блок и располагается в области информации о каналах.

В этом блоке содержится вся информация, необходимая для того, чтобы канал мог функционировать. Четыре стандартных канала "K", "S", "P" и "R" имеют каждый по пятибайтному блоку, но это исключение. Все остальные каналы, в том числе и те, которые создадите Вы, должны иметь не менее, чем по 11 байтов в этом блоке на каждый канал.

По четырем стандартным каналам эти блоки выглядят так:

Байты 0 и 1 - адрес процедуры вывода (PRINT#).

Байты 2 и 3 - адрес процедуры ввода (INPUT#).

Байт 5 - имя канала (код одной буквы "K", "S", "R" или "P").

При подключении стандартной периферии, например ИНТЕРФЕЙСа-1 этот блок имеет длину 11 байтов и адресация к ним производится помещением в регистровую пару процессора IX базового адреса начала блока.

IX+0 (2 байта) - адрес процедуры обработки ошибок 0008.

IX+2 (2 байта) - адрес процедуры обработки ошибок 0008.

Листинг 7.

В области информации о каналах данная процедура выполняет поиск блока с заданным именем.

Данная процедура используется при работе процедуры V_OPEN (B966), см. Листинг 20.

DD2A4F5C	SEARCH_CH_ALL	ORG B594 LD IX, (CHANS)	; В IX - базовый адрес на- ; чала области информации ; о каналах.
011400		LD BC, 0014	
57		LD D, A	; D содержит ASCII-код ; имени канала.
DD09	SCH_CH_LOOP	ADD IX, BC	; IX указывает на адрес ; очередного блока.
DD7E00	SEARCH_CH	LD A, (IX+00)	
FE80		CP 80	; Напомним, что 80H - мар- ; кер, отмечающий конец ; области информации о ; каналах.
37		SCF	; Включение флага C.
C8		RET Z	; Выход с включенным C-фла- ; гом, если блок канала C ; таким именем не найден.
DD7E04		LD A, (IX+04)	; A содержит ASCII-код ; имени канала.
BA		CP D	; Сравнили с искомым.
C8		RET Z	; Выход, если найден.
DP4E09		LD C, (IX+09)	; BC - длина блока инфор-
DD460A		LD B, (IX+0A)	; мации о канале.
18EA		JR SCH_CH_LOOP	; Переход в начало цикла ; для продолжения поиска.

Листинг 8.

Нижеприведенная процедура служит для обслуживания такой "хитрой" конструкции как "трехбайтный" регистр. Ранее мы писали, что в 128-килобайтных моделях для хранения адреса недостаточно двух байтов и приходится привлекать еще и кодовый номер страницы. Такой "адрес" хранится в трех регистрах процессора, например, B, H, L.

Процедура выполняет операцию декремента (DEC BHL).

2B	DEC_BHL	ORG B70E DEC HL	; Декремент пары HL.
78		LD A, B	; в A был код страницы.
FE05		CP 05	; Возврат, если это обычное
C8		RET Z	; ОЗУ.
CB74		BIT 6, H	; Возврат, если не было пе-
C0		RET NZ	; рехода между страницами.
CBF4		SET 6, H	
05		DEC B	
C9		RET	

IX+4 (1 байт) - имя канала.

IX+5 (2 байта) - адрес процедуры PRINT#, которая находится в "теневом" ПЗУ интерфейса.

IX+7 (2 байта) - адрес процедуры INPUT#, которая находится в "теневом" ПЗУ интерфейса.

IX+9 (2 байта) - длина блока информации о канале (не менее 000B).

IX+0B (длина любая) - любая дополнительная информация.

Блоки информации о пользовательских каналах тоже лучше создавать и обслуживать,

используя адресацию через индексный регистр IX. Тогда такой блок имеет следующий вид:

- IX+0 (2 байта) - адрес процедуры вывода (PRINT#).
- IX+2 (2 байта) - адрес процедуры ввода (INPUT#).
- IX+4 (1 байт) - имя канала,
- IX+5 (2 байта) - число 1234, свидетельствующее о том, что этот канал не стандартный, а пользовательский.
- IX+7 (2 байта) - адрес закрывающей процедуры (CLOSE#).
- IX+9 (2 байта) - длина блока информации о канале (не менее 000B).
- IX+0B (длина любая) - любая дополнительная информация.

Использование процедур ПЗУ

Процедура вывода, содержащаяся в ПЗУ (RST 10H) и процедура ввода INPUT_AD (15E6H) работают со стандартными каналами. Поэтому при выводе происходит сразу обращение в первые два байта блока информации о каналах для поиска адреса печатающей процедуры, а при вводе - обращение в 3-ий и 4-ый байты для поиска адреса читающей процедуры.

Если Вы используете стандартную периферию, например Интерфейс-1, то в этих байтах содержится адрес 0008 (адрес процедуры обработки ошибки). При попадании туда происходит "впечатывание" страницы "теневого" ПЗУ вместо стандартного, а уже из него выполняется выбор процедур печати и чтения (байты IX+5 ... IX+8).

И снова о потоках

Теперь, когда мы разобрались с каналами и поняли, что это не так сложно, что это всего лишь еще один способ организации памяти компьютера и взаимодействия процедур друг с другом, вернемся к потокам.

Листинг 9.

Данная процедура предназначена примерно для тех же целей, что и знаменитая команда процессора LDIR и служит для быстрого перемещения значительных блоков памяти в пределах RAM-диска.

Поскольку адресация в дополнительной памяти 128-килобайтных машин невозможна через двухбайтный регистр, эта процедура введена для обслуживания трехбайтных конструкций типа BHL, B'H'L', CDE и др.

Процедура выполняет три функции:

- 1) Выполняет декремент BHL и B'H'L';
- 2) перегружает байт из BHL в B'H'L';
- 3) если BHL не равен CDE, переход на пункт 1.

		ORG B71A	
CD0EB7	V_TRANSFER	CALL B70E, DEC_BHL	; Декремент BHL.
D9		EXX	
CD0EB7		CALL B70E, DEC_BHL	; Декремент B'H'L'.
D9		EXX	
78		LD A, B	; В A стал код страницы
			; источника.
CDFFB6		CALL B6FF, V_PAGE	; Впечатывание страницы 03У
7E		LD A, (HL)	; Прием перебрасываемого
			; байта и
F5		PUSH AF	; сохранение его на стеке.
D9		EXX	
7B		LD A, B	; В A стал код страницы
			; назначения.
CDFFB6		CALL B6FF, V_PAGE	; Впечатывание страницы 03У
F1		POP AF	; Восстановление пересылае-
77		LD (HL), A	; мого байта и пересылка.
D9		EXX	
78	V_TRANSFER_2	LD A, B	; В A стал код страницы
			; источника.

B9	CP C	
20E6	JR NZ, V_TRANSFER	; На повтор цикла.
ED52	SBC HL, DE	; Включение флага Z, если
		; все байты переданы.
19	ADD HL, DE	; Эта операция на Z флаг
		; не влияет.
20E1	JR NZ, V_TRANSFER	; На повтор цикла, если не
		; все байты переданы.
C9	RET	

Листинг 10.

Эта вспомогательная процедура рассчитывает адрес при страничной организации памяти, отстоящий от адреса, содержащегося в комплексе AHL на величину, содержащуюся в паре BC.

Предполагается, что величина в BC не превосходит размера одной страницы – 4000H (16 килобайт).

		ORG B73A	
09	ADD_AHL_BC	ADD HL, BC	; 16-разрядное сложение.
FE05		CP 5	; Сравниваем код страницы с
			; числом 5.
C8		RET Z	; Возврат, если это стандартное 03У.
CB74		BIT 6, H	; Возврат, если не было пе-
C0		RET NZ	; рехода между страницами.
CBFC		SET 7, H	
CBF4		SET 6, H	
3C		INC A	; Увеличиваем код страницы,
			; если был переход.
C9		RET	

Среди системных переменных компьютера есть переменная STRMS. Ее адрес - 23568 (5C10H). Ее назначение - указание на адреса каналов, подключенных к потокам. Длина этой системной переменной - 38 байтов и если говорить откровенно, то никакая это не переменная, а самая настоящая указательная таблица, в которой каждому потоку отданы два байта, содержащие адрес канала, к которому данный поток подключен.

Правда, у внимательного читателя может сразу возникнуть вопрос - почему же потоков 16, да на каждый по 2 байта, итого 32, а STRMS содержит 38 байтов.

Дело в том, что есть еще три потока, которые из БЕЙСИКа вам недоступны - только из машинного кода. Эти потоки занимаются своими "внутренними" делами при работе ПЗУ. Это "минус третий" поток (FD), "минус второй" (FE) и "минус первый" (FF). Таким образом, таблицу STRMS можно представить, как 19 двухбайтных системных переменных:

```

STRMS_FD 5C10 (23568)
STRMS_FE 5C12 (23570)
STRMS_FF 5C14 (2357E)
STRMS_00 5C16 (23574)
STRMS_01 5C16 (83576)
.....
STRMS_0F 5C36 (23606)

```

Поток FD подключен к каналу "K" и не должен переподключаться. Аналогично поток FE подключен к каналу "S". Интересен поток FF, подключенный к "внутреннему" "Спектрумовскому" каналу "R", который отвечает за динамическое копирование информации из одних областей памяти компьютера в другие, производя при этом "раздвигание" информации для вставки новой в середину имеющейся. Мы об этом писали в разделе "Секреты ПЗУ", когда рассматривали процедуры БЕЙСИКовского редактора.

Итак, с помощью таблицы переменных STRMS выполняется привязка потоков к каналам. Если какая-либо переменная из набора STRMS содержит 0000, то это означает, что к данному потоку не подключен ни один канал, иначе говоря, канал закрыт. Если же там не ноль, значит канал открыт и данный поток подключен к этому каналу. Фактически же поток подключен к тому каналу, информационный блок которого начинается с адреса, на который

указывает системная переменная CHANS (23631) плюс величина, содержащаяся в STRMS для данного потока минус единица:

$(CHANS) + (STRMS_n) - 1$

Для тех, кто не знаком с программированием в машинных кодах, укажем, что круглые скобки в этой формуле означают, что речь не идет об адресе самой системной переменной CHANS, а об адресе, на который она указывает, т.е. который в ней хранится, то же и (STRMS_n).

Листинг 11.

Процедура предназначена для поиска в каталоге RAM-диска файла с именем, которое задано в блоке информации о канале. Если файл с таким именем в каталоге не существует, выдается системное сообщение об ошибке. При успешном окончании работы процедуры на выходе в индексной паре IX устанавливается адрес, указывающий на требуемую информацию в каталоге. При входе там должен быть выставлен адрес места расположения блока информации о канале.

DDE5	FIND_FILE	ORG B747	
E1		PUSH IX	;Переброска адреса блока
010E00		POP HL	;через стек.
		LD BC,000E	;Имя в блоке начинается с
			;байта IX+0E.
09		ADD HL,BC	;HL указывает на имя.
0E0A		LD C,0A	;Длина имени - 10 знаков.
11675B		LD DE,5B67,NSTR_1	;В системной переменной
			;NSTR_1 должно содержать-
			;ся имя искомого файла.
EDB0		LDIR	;Переброска имени из блока
			;информации в системную
			;переменную.
CD08B7		CALL B708,V_FIND	;Вызов процедуры ПЗУ для
			;поиска по каталогу.
C0		RET NZ	;Выход, если файл найден.
CDFCBP		CALL B6FC,V_ERROR	;Вызов системной процеду-
23		DEFB 23	;ры генерации сообщения об
			;ошибке с кодом перехвата
			;23H (h "File does not
			;exist") - "Файл не су-
			;ществует."

Листинг 12.

Данная процедура выполняет важную задачу. Мы организовали в блоке информации о канале "V" полукилобайтный буфер и эта процедура устанавливает взаимное соответствие между данным буфером и областью оперативной памяти на RAM-диске. Эта область представляет тем самым как бы последовательность полукилобайтных секторов.

На входе в эту процедуру необходимо, чтобы в IX был установлен адрес начала блока информации о канале "V".

На выходе из этой процедуры устанавливаются:

- в регистрах BHL - адрес первого байта, следующего за последним сектором, заполненным на RAM-диске (для данного файла);
- в регистрах CDE - адрес начала сектора на RAM-диске;
- в регистрах B'H'L' адрес байта, следующего за буфером канала "V" (в блоке информации о канале).

DDE5	V_MATCH	ORG B75D	
DD7E0D		PUSH IX	;Сохранили на стеке адрес
F5		LD A,(V_CHREC)	;блока информации о канале
CD47B7		PUSH AF	;и номер записи в файле.
		CALL B747, FIND_FILE	;Теперь IX указывает на
			;запись о данном файле в
			;каталоге, а

C1		POP BC	; В - на номер записи в файле.
CB20		SLA B	; Удвоение содержимого В.
0E01		LD C, 01	; В BC теперь 200H*(номер записи)+1
37		SCF	; Сигнал "конец блока" и
08		EX AF, A'F'	; сохранение его.
DD6E0D		LD (L, SF_LEN)	; В AHL устанавливается
DD560E		LD (H, SF_LEN+1)	; длина
DD7E0F		LD (A, SF_LEN+2)	; файла.
A7		AND A	; Обнуление флага переноса,
			; чтобы не исказить следу-
			; ющую операцию.
ED42		SBC HL, BC	; Вычитание с "займом"
DE00		SBC A, 00	; Вычитание "займа" из A,
			; теперь в AHL - размер
			; "остатка" файла.
A7		AND A	; проверка на ноль.
2008		JR NZ, V_M_N_EOF	; Переход, если старший
			; байт AHL еще не обнулен.
110102		LD DE, 0201	
ED52		SBC HL, DE	; Сравнение размера "остат-
19		ADD HL, DE	; с длиной буфера. Если бо-
3604		JR C, V_M_EOF	; льше, то включится флаг
			; переноса и тогда переход.
210002	V_M_N_EOF	LD HL, 200	; Длина записи max 200H.
08		EX AF, A'F'	; Запомнили выключенный
			; флаг C, означающий "Не
			; конец записи".
EB	V_M_EOF EX	DE, HL	; DE - длина записи.
DD6E0A		LD L, (SF_START)	; В AHL устанавливается
DD660B		LD H, (SF_START+1)	; адрес начала
DD7E0C		LD A, (SF_START+2)	; файла.
CD3AB7		CALL B73A, ADD_AHL_BC	; CM. B73A.
C5		PUSH BC	; 200*(номер записи)+1
D5		PUSH DE	; Длина записи.
E5		PUSH HL	; Адрес сегмента и код
F5		PUSH AF	; его страницы.
42		LD B, D	
4B		LD C, E	
CD3AB7		CALL B73A, ADD_AHL_BC	; В AHL помещается адрес
			; очередного сектора на
			; RAM-диске.
47		LD B, A	
F1		POP AF	
4F		LD C, A	; В CDE адрес сегмента
D1		POP DE	; и код его страницы.
D9		EXX	; Включение альтернативных
			; регистров.
D1		POP DE	; D'E' - длина записи.
C1		POP BC	; B'C'=200*(номер записи)+1
DDE1		POP IX	; В IX - адрес блока инфор-
			; мации о канале.
DD7319		LD (V_RECLRN), E	; Заносим длину записи в
DD731A		LD (V_RECLRN+1), D	; V_RECLRN.
DDE5		PUSH IX	; в HL - адрес блока инфор-
E1		POP HL	; мации о канале.
011B00		LD BC, 001B	; В блоке информации о ка-
			; нале до буфера 1B байтов.
09		ADD HL, BC	; H'L' указывает на буфер
			; канала "V".
19		ADD HL, DE	; H'L' указывает на байт,
			; следующий за текущей
			; записью.

0605	LD B, 05	; B'=05 - сигнал "страница
		; стандартного ОЗУ".
D9	EXX	; Основной набор регистров.
DDCB188E	RES 1, (V_CHFLAG)	; Сигнал "не последний
		; блок в файле."
08	EX AF, A'F'	; Напомним - флаг C несет
		; информацию о конце файла.
D0	RET NC	; Возврат, если блок не по-
		; следний.
DDCB18CE	SET 1, (V_CHFLAG)	; Сигнал "конечный блок
		; файла".
C9	RET	

Из всего вышесказанного вытекает одна тонкость. Оказывается с программистской точки зрения проще создать и открыть новый канал, чем закрыть его. Действительно, если Вы решили создать канал, Вы в конце области информации о каналах, на которую указывает (CHANS), припишете блок информации о канале (11 байтов или более) и для желаемого Вами потока #n запишете в соответствующую переменную STRMS_n указание на этот блок.

А что же, если Вы хотите закрыть канал? Тогда в соответствующее значение STRMS_n Вы запишете нули, но надо еще уничтожить блок информации о канале. Хорошо, когда он - последний. Убрали его и все. А если он находится в середине, тогда придется все вышестоящие блоки сдвигать вниз с изменением при этом содержимого STRMS_n+1, STRMS_n+2

Как закрыть канал.

Итак, если мы хотим научиться сами создавать свои каналы, подключать к ним потоки, т.е. открывать эти каналы и использовать их для своих целей, то прежде всего надо научиться закрывать свои каналы, для чего и служит предлагаемая ниже процедура CLOSE_NEW (листинг 1).

При входе в эту процедуру регистр A микропроцессора должен содержать номер подключаемого потока. Если канал уже закрыт или если канал не является каналом, созданным пользователем, то процедура выполняет немедленный возврат и ничего не делает.

С другой стороны, если канал открыт (поток подключен) и это пользовательский канал, процедура "вычищает" память, занятую блоком информации о канале, сдвигает вышестоящие блоки вниз ("убирает мусор") и перестраивает системные переменные в таблице STRMS.

При работе процедуры, учитывается также состояние флага переноса (флаг C флагового регистра F). Если при входе в процедуру через адрес CLOSE_CL флаг C включен, значит в буферах есть данные и перед тем, как закрыть канал надо их очистить, выдав все данные, чтобы информация не пропала. Если флаг C выключен, то наличие данных в буферах можно проигнорировать.

Если однозначно необходимо буфер отгрузить, то входить в процедуру надо через точку входа CLOSE_NEW.

Ниже представлена также вспомогательная процедура CLEAR CHANS (листинг 2), служащая для закрывания всех пользовательских каналов. При вызове этой процедуры все данные, оставшиеся в буферах, уничтожаются.

Листинг 13.

Процедура настраивает буфер канала, подготавливая к работе с файлом, открытым для чтения.

		ORG B7C5	
CD5DB7	V_ASSIGN	CALL B75D, V_MATCH	
CD30B7		CALL B730, V_TRANSFER_2	; Копирует данные в буфер.
DD360B00	V_BUF_EXIT	LD(V_CHBYTE), 00	; обнуляет указатель дан-
DD360C00		LD(V_CHBYTE+1), 00	; ных в буфере.
C9		RET	

Эти программы нерелоцируемы, то есть их загружать можно только в те адреса памяти, для которых они написаны, и которые стоят в директиве АССЕМБЛЕРА ORG. Это сделано потому, что за этими процедурами пойдут другие и все они должны работать совместно, как единое целое.

Файлы последовательного доступа на RAM-диске

Теперь, когда мы научились, в принципе, закрывать созданные каналы, можно попробовать их создавать. В качестве примера мы рассмотрим идею о том, как владельцы 128-килобайтных машин могут использовать их потенциал, прибегнув к концепции потоков и каналов.

Что еще можно делать с помощью каналов и потоков? Мы назовем несколько направлений. Например, организуется нестандартная печать, скажем 45, или 55 или 60 символов в строке. С их же помощью организуется, например, работа с "окнами" так, как это сделано в "МЕГАБЕЙСИКе" или, скажем, в "ЛАЗЕР-БЕЙСИКе". Вы можете расширить БЕЙСИК своего компьютера, придумав для него несколько новых команд. И самое интересное, пожалуй, - это организация обмена между разными машинами. Мы бы с удовольствием привели пример именно из этой области, но для этого пришлось бы освещать работу и этой (другой) машины, что никак не входит в наши планы. Приходится ограничиться вопросами обмена данными между различными областями одной машины.

Итак, в качестве примера мы создадим канал, позволяющий работать с файлами последовательного доступа на RAM-диске 128-килобайтных компьютеров.

Примечание.

Это, конечно не значит, что Вам надо завтра сразу все бросить и побежать покупать 128-килобайтную машину, пока Вам кто-то не сумеет объяснить, что же хорошего, кроме лишнего десятка "игрушек" Вы получите от этих дополнительных 80 килобайтов, к которым у процессора нет и не может быть прямой адресации.

Кстати, уникальной особенностью этих машин (в импортном исполнении) является трехголосый звуковой синтезатор, ради которого все и затевалось. Пока у нас в стране нет никаких аналогов этой микросхемы, все попытки "продать" пользователю 128-килобайтную модель по цене хотя бы на 200 рублей дороже 48-килобайтного аналога - это просто эксплуатация его доверчивости и неинформированности.

Эта задачка так и осталась бы академической, но есть и факты: темп продаж 128 килобайтных машин на Западе так и не достиг и 10% темпа продаж обычных 48-килобайтных машин в период их расцвета. Даже попытка встроить дисковод в "ZX-Spectrum+3" ничего кроме очередного разочарования не дала и количество программ, поддерживаемых этой версией, можно пересчитать на пальцах одной руки.

Вообще история сэра К. Синклера это такой роман с потрясающими взлетами и глубокими падениями, изучая который можно уберечь пользователя от излишних тысячных затрат, а производителя - от многомиллионных.

Листинг 15.

Данная процедура принимает из "V"-канала единичный символ и помещает его в аккумулятор процессора.

CD005B	V_INKEY	ORG B803	
2A5A5B		CALL 5B00,SWAP	; "Впечатывание" ПЗУ-0.
E5		LD HL,(RETADDR)	
D9		PUSH HL	
C5		EXX	
D5		PUSH BC	
E5		PUSH DE	
DD2A515C		PUSH HL	
		LD (IX,CURCHL)	; IX указывает на начало
			; блока информации о
			; канале.

DDCB1846 2804 CDFCB6 1D	V_ERROR_1	BIT 0, (V_CHFLAF) JR Z, V_INKEY_2 CALL B6FC, V_ERROR DEFB 1D	; Если это READ-файл - ; то перевод. Иначе ; генерация сообщения об ; ошибке с кодом перехвата ; 0D ("b: Wrong file type") ; "Неверный тип файла". ; В DE - позиция следующего ; читаемого байта. ; Проверка на содержание в ; блоке метки "конец файла" ; В HL - длина текущей за- ;писи в файле.
DD5E0B DD560C DDCB184E 280F DD6E19 DD661A A7 ED52 2004 CDFCB6 07	V_INKEY_2	LD E, (V_CHBYTE) LD D, (V_CHBYTE+1) BIT 1, (V_CHFLAG) JR Z, V_INKEY_RD LD L, (V_RECLen) LD H, (V_RECLen+1) AND A SBC HL, DE JR Z, V_INKEY_RD CALL B6FC, V_ERROR DEFB 07	; Проверка на достижение ; последней записи. ; Генерация сообщения об ; ошибке с кодом перехвата ; 07 ("8: End of file") ; "Конец файла".
DDE5 E1	V_INKEY_RD	PUSH IX POP HL	; HL указывает на блок информации ; по каналу "V".
011B00 09 19		LD BC, 001B ADD HL, BC ADD HL, DE	; HL указывает на буфер. ; HL указывает на очеред- ; ной читаемый символ.
7E		LD A, (HL)	; В аккумулятор поступает ; символ от INKEY\$.
F5 13 DD730B DD7E0C 15 15 2006 DD340D CDC5B7 F1		PUSH AF INC DE LD (V_CHBYTE), E LD (V_CHBYTE+1), D DEC D DEC D JR NZ, V_INKEY_EXIT INC (V_CHREC) CALL B705, Y_ASSIGN	; Передвинули указатель. ; Запомнили положение ; указателя. ; Если D=2 значит пора об- ; новлять бУФер.
	V_INKEY_EXIT	POP AF	; Увеличили номер записи.
37 E1 D1 C1 D9 E1 225A5B	V_INPUT_EXIT V_EXIT	SCF POP HL POP DE POP BC EXX POP HL LD (RETADDR), HL	; В A - только что ; прочитанный байт.
C3005B		JP 5B00, SWAP	; В HL - адрес возврата в ПЗУ-0. ; Настройка системной пере- ; менной на адрес возврата. ; "Впечатывание" ПЗУ-1 и ; возврат.

Те, кто знаком с такими носителями информации как флоппи-диск или микродрайв, представляют, что такое файлы последовательного доступа, открытые для чтения (READ-файл) или для записи (WRITE-файл). В файл, открытый для записи, Вы можете "впечатывать" текст, затем этот файл может быть закрыт (CLOSE) и снова открыт (OPEN) для чтения. Вводить информацию (текстовую и числовую) в такой файл Вы сможете прямо из БЕЙСИКа.

Нужны ли Вам "файлы последовательного доступа" или нет - дело Ваше, но показать, каким путем идея потоков и каналов используется в приложениях мы должны, а Вы сами разберетесь, как Вам поступить.

RAM-диск

Файлы, размещенные в RAM-диске, который нередко еще называют виртуальный диск, работают точно так же, как и файлы, размещенные например на гибких дисках,

когда Вы открываете (OPEN) на RAM-диске файл для записи, то в верхних страницах памяти компьютера создается файл с заданным именем. В него можно вводить какие-то данные. Когда файл закрыт (CLOSE), в него нельзя ввести ничего, но можно открыть файл для чтения и тогда из него можно ввести в какую-либо БЕЙСИК-переменную то, что там содержится посредством INPUT.

Как и при работе с обычным диском, файл на RAM-диске должен обязательно быть закрыт после того, как закончился ввод в него или вывод из него. Если файл, открытый для записи, не будет закрыт, то есть вероятность того, что часть данных (а может быть и все) будут безвозвратно утрачены, поскольку специальный буфер не будет очищен (отгружен в файл). Последствия для незакрытого READ-файла не столь неприятные, хотя надо признать, что каждый канал, обслуживающий RAM-диск, занимает пол-килобайта памяти и освободить их, не закрыв файл, нельзя. Поэтому ВСЕГДА ЗАКРЫВАЙТЕ ФАЙЛ, КОГДА ЗАКОНЧИЛИ РАБОТУ С НИМ.

Листинг 16.

Эта крупная и важная процедура выполняет операцию вставки байтов в уже существующий на RAM-диске файл. Те файлы, которые в результате такой вставки должны быть передвинуты, перемещаются и переиндексируются.

При входе в эту процедуру комплекс регистров ANL должен содержать адрес (с кодовой страницей), в который выполняется вставка байта (байтов), а пара BC - длину вставляемого блока.

	ORG B85F	
C5	V_MAKEROOM	PUSH BC
E5		PUSH HL
F5		PUSH AF
AF		XOR A
		; Обнуление аккумулятора,
		; выключение флага C.
67		LD H, A
6F		LD L, A
		; Обнуление HL.
ED42		SBC HL, BC
9F		SBC A, A
		; в результате в ANL - ми-
		; нус число вводимых байтов
CD05B7		CALL B705, V_SPACE
		; Проверка на достаточность
		; памяти для вставки.
F1		POP AF
		; Восстановление
E1		POP HL
		; исходных
C1		POP BC
		; данных.
C5		PUSH BC
E5		PUSH HL
F5		PUSH AF
3E04		LD A, 04
		; 4-ая страница 03У содер-
CDFFB6		CALL B6FF, V_PAGE
		; жит каталог.
DD2A835B		LD IX, (SF_NEXT)
		; IX указывает на конец
		; каталога.
DD6E0A		LD L, (SF_START)
		; ANL указывает на первый
DD660B		LD H, (SF_START+1)
		; свободный байт простран-
DD7E0C		LD L, (SF_START+2)
		; ства на RAM-диске.
F5		PUSH AF
E5		PUSH HL
		; Запомнили этот адрес.
CD3AB7		CALL B73A, ADD_ANL_BC
		; ANL указывает на первый
		; свободный байт который
		; будет после операции
		; вставки блока байтов.
		; теперь этот адрес в BHL.
47		LD B, A
D9		EXX
		; Теперь он в B'H'L'.
E1		POP HL
		; BHL - адрес первого сво-
C1		POP BC
		; бодного байта (старый).
F1		POP AF
		; ADE - адрес куда идет
D1		POP DE
		; вставка.
4F		LD C, A
		; Теперь это - CDE.
D5		PUSH DE
		; Запомнили
F5		PUSH AF
		; этот адрес.

CD30B7		CALL B730, V_TRANSFER_2	; Перемещение байтов.
3E04		LD A, 04	; 4-ая страница ОЗУ содер-
CDFFB6		CALL B6FF, V_PAGE	; жит каталог.
C1		POP BC	; BDE - адрес куда идет
D1		POP DE	; вставка.
DD6E0A	V_MR_LOOP	LD L, (SF_START)	; AHL указывает на первый
DD660B		LD H, (SF_START+1)	; свободный байт (старый)
DD7E0C		LD L, (SF_START+2)	; на RAM-диске.
B8		CP B	
382E		JR C, V_MR_FOUND	; Переход, если файл нахо-
			; дится до точки вставки
			; (проверили старший байт).
ED52		SBC HL, DE	; Переход, если файл нахо-
19		ADD HL, DE	; дится до точки вставки
3829		JR C, V_MR_FOUND	; (проверили младшие
			; байты).
E3		EX (SP), HL	; В DE теперь количество
EB		EX DE, HL	; вставляемых байтов.
19		ADD HL, DE	
3005		JR NC, V_MR_ADDR	
CBFC		SET 7, H	
CBF4		SET 6, H	
3C		INC A	; В AHL новый адрес файла
EB	V_MR_ADDR	EX DE, HL	
E3		EX (SP), HL	
EB		EX DE, HL	; BDE - позиция вставки.
DD750A		LD (SF_START), L	; Откорректировали каталог
DD740B		LD (SF_START+1), H	; под новые адресные данные
DD770C		LD (SF_START+2), A	; файла.
C5		PUSH BC	
011400		LD BC, 0014	
DD09		ADD IX, BC	; IX - указывает на следую-
			; щий файл.
C1		POP BC	; BDE - позиция вставки.
DD7510		LD (SF_END), L	; Откорректировали каталог
DD7411		LD (SF_END+1), H	; под новые адресные данные
DD7712		LD (SF_END+2), A	; следующего файла.
18C6		JR V_MR_LOOP	; Возврат к началу цикла
			; для работы с этим файлом.
DD6E0D	V_MR_FOUND	LD (L, SF_LEN)	
DD660E		LD (H, SP_LEN+1)	
DD7E0F		LD (A, SF_LEN+2)	; AHL-старая длина файла.
EB		EX DE, HL	
E3		EX (SP), HL	
EB		EX DE, HL	; DE-число вставляемых
			; байтов.
19		ADD HL, DE	
CE00		ADC A, 00	; AHL-новая длина файла.
DD750D		LD (SF_LEN), L	; (Откорректировали каталог
DD740E		LD (SF_LEN+1), H	; под длину нового
DD770F		LD (SF_LEN+2), A	; файла.
78		LD A, B	; A-код страницы места
			; вставки.
42		LD B, D	; BC-число вставляемых
4B		LD C, E	; байтов.
E1		POP HL	; AHL - адрес, куда был
			; вставлен блок.
C9		RET	

Как только файл на RAM-диске открыт, он будет внесен в каталог, который Вы можете вызвать из БЕЙСИКа командой CAT!

Принцип, по которому работает приведенная ниже программа, основан на концепции организации электронного диска (RAM-диска) в расширенной памяти 128-килобайтных машин. Вы уже знаете из предыдущих наших материалов, что в этой области памяти можно

хранить программы, данные или машинный код в течение того времени, пока компьютер остается включенным. RAM-диск значительно быстрее любого физического магнитного диска, но зато он "погибает" при выключении питания машины и своевременно должен быть отгружен.

Все это относится и к файлам последовательного доступа, о которых мы ведем речь.

Память на электронном диске организована с помощью каталога. Каталог является как бы указателем к тому, что есть на этой диске. Сам каталог размещается на седьмой странице дополнительной памяти и фактически организован по принципу стека, начинающегося по адресу 7EBFF и расширяющегося вниз. Каждая запись в каталоге занимает 20 байтов. Ниже приведены значения каждого из этих 20 байтов. IX указывает на начало записи. В конце каталога стоит 20-байтный маркер, обозначающий "конец каталога" (хотя на самом деле из него используются только три байта). Системная переменная, имеющаяся в 128-килобайтных машинах SFNEXT, указывает на этот маркер и служит как бы "указателем стека". Таким образом, ведя каталог в требуемом формате, мы можем управлять RAM-диском из машинного кода.

Структура каталога.

IX+00 SF_NAME Имя файла.

IX+0A SF_START Адрес начала файла (с кодом страницы).

IX+0D SF_LEN Длина файла вместе с заголовком.

IX+10 SF_END Адрес байта, следующего за окончанием файла (с кодом страницы).

IX+13 SF_FLAG Флаговый байт. Выключен, если каталог не завершен (т.е. как правило он выключен).

Файлы RAM-диска начинаются на первой странице ОЗУ и развиваются вверх, проходя через страницы ОЗУ - 3,4,6 и 7. Необходимо принимать во внимание, что они в своем развитии не должны перекрыть область, отведенную под каталог и, тем самым, погубить всю информацию.

Чтобы избежать конфуза с такой "странной" нумерацией страниц, введено понятие "кодовой страницы", о чем мы писали в статьях "128 K" (см. стр. 114,115 ИНФОРКОМ). Это означает, что пока один регистр процессора запоминает "код страницы", другой в это время хранит ее реальный физический адрес. Номера этих "кодовых страниц" - 0,1,2,3 и 4, а номер 5 относится к тем файлам RAM-диска, которые размещены в обычных первых 48 килобайтах компьютера.

Канал "V"

Мы назовем наш пользовательский канал, с помощью которого мы будем управлять файлами на RAM-диске, латинской буквой "V" (от слова "виртуальный") и создадим его так, чтобы это управление можно было делать средствами обычного БЕЙСИКа.

Для организации такого канала нам потребуется блок информации о канале размером более 500 байтов, хотя большую его часть будет занимать не сама информация, а буфер. То, что Вы будете засылать в файл на RAM-диске, на самом деле будет поступать не туда, а в этот буфер и перебрасываться в файл только по полному заполнению буфера (или при закрытии канала).

Вам надо иметь в виду, что адрес места расположения файла на RAM-диске не является величиной постоянной. Все файлы могут мигрировать по мере стирания каких-то файлов или по мере вставки новой информации в готовые файлы. Таким образом, они могут перемещаться всякий раз, когда мы с ними работаем. Использование буфера позволяет нам сделать этот процесс непостоянным и, тем самым, сэкономить машинное время.

Ниже приведена структура блока информации о канале "V".

Обратите внимание на то, что используемые в этом блоке переменные V_CHREC, V_RECLEN и первый бит V_CHFLAG используются только при работе с файлами, открытыми для чтения (READ). Прочие же используются с файлами обоего типа.

Листинг 17.

Приведенная ниже процедура V_STORE выполняет перенос содержимого буфера канала "V" из блока информации о канале в соответствующее ему место на RAM-диске.

DD4E0B	V_STORE	ORG B8FE	
DD460C		LD C, (V_CHBYTE)	
DDES		LD B, (V_CHBYTE+1)	; BC-число байтов в буфере.
		PUSH IX	; сохранили адрес блока
C5		PUSH BC	; информации о канале.
			; Сохранили число байтов в
			; буфере.
CD47B7		CALL B747, FIND-FILE	; IX-адрес информации о
			; файле в каталоге.
C1		POP BC	; Восстановили BC.
DD6E10		LD L, (SF_END)	; В AHL выставляем адрес
DD6611		LD H, (SF_END+1)	; первого байта за данным
DD7E12		LD H, (SF_END+2)	; файлом в каталоге.
CD5FB6		CALL B85F, V_MAKER00M	; Выделение места в ОЗУ.
DDE1		POP IX	; Восстановили BC.
CD3AB7		CALL B73A, ADD_AHL_BC	; В AHL выставляем адрес
			; первого байта за пересы-
			; лаемым блоком байтов.
C5		PUSH BC	
47		LD B, A	; То же в BHL,
D9		EXX	; То же в B'H'L'.
DDE5		PUSH IX	; Переброска IX
E1		POP HL	; в HL.
011B00		LD BC, 001B	; HL указывает теперь
09		ADD HL, BC	; на начало буфера.
C1		POP BC	; Восстановили BC.
E5		PUSH HL	
09		ADD HL, BC	; HL указывает теперь
			; на байт за буфером.
D1		POP DE	; Начало буфера.
010505		LD BC, 0505	; Напомним: 05 - код
			; страницы стандартного
			; ОЗУ, в котором и раз-
			; мещен блок информации
			; о канале "V".
CD30B7		CALL B730, V_TRANSFER_2	; Копирование буфера
76		LD A, B	; A=5
CDFFB6		CALL B6FF, V_PAGE	; "Впечатывание" стан-
			; дартного ОЗУ.
C3CBB7		JP B7CB, V_BUFF_EXIT	; Приведение указателя
			; буфера в исходное со-
			; стояние и "выход".

Листинг 18.

Процедура, отвечающая за вывод информации. Она выполняет "печать" по каналу "V". Сначала символ, содержащийся в аккумуляторе помещается в буфер и только потом переправляется на RAM диск.

CD005B	V_PRINT	ORG B92B	
2A5A5B		CALL 5B00, SWAP	; Впечатывание страницы 0.
E5		LD HL, (RETADDR)	
D9		PUSH HL	; Сохранили адрес возврата.
c5		EXX	
D5		PUSH BC	
E5		PUSH DE	
DD2A515C		PUSH HL	
		LD IX, (CURCHL)	; IX указывает на адрес ин-
			; формации о канале.
DDCB1846		BIT 0, (V_CHFLAG)	
CA18B8		JP Z, B818, V_ERROR	; Ошибка, если это файл для
			; чтения.
DD5E0B		LD E, (V_CHBYTE)	; DE - количество байтов
DD560C		LD D, (V_CHBYTE+1)	; в буфере.
DDE5		PUSH IX	

E1	POP HL	; HL - указывает на блок ; информации о канале.
011B00	LD BC, 001B	
09	ADD HL, BC	; HL-начало буфера.
19	ADD HL, DE	; HL-первый свободный байт.
77	LD (HL), A	; Байт - в буфер.
13	INC DE	; Новое число б-в в буфере.
DD730B	LD (V_CHBYTE), E	; Запомнили новое число
DD720C	LD (V_CHBYTE+1), D	; байтов в буфере.
15	DEC D	; Если буфер заполнен,
15	DEC D	; он опорожняется
CCF2B8	CALL Z, B8F2, V_STORE	; в файл на RAM-диске.
A7	AND A	; Выкл. флага переноса.
C354B8	JP B854, V_INOUT_EXIT	; Переход на выходную ; процедуру.

Структура блока информации о канале "V".

IX+00 V_OUT - адрес процедуры вывода информации из файла на RAM-диске (B92B)

IX+02 V_IN - адрес процедуры ввода информации в файл на RAM-диске (B7D4).

IX+04 V_NAME - имя канала ("V")

IX+05 V_IDEN - идентификатор пользовательского канала ("1234").

IX+07 V_CLOSE - адрес процедуры, закрывающей файл (B960).

IX+09 V_LEN - Длина блока информации о канале (021B).

IX+0B V_CHBYTE- указатель буфера

IX+0D V_CHREC - номер записи в файле.

IX+0E V_CHNAME - имя файла.

IX+18 V_CHFLAG- вспомогательные флаги:

Бит 0 - выключен для файла, открытого для чтения и включен для файла, открытого для записи. Бит 1 - включен, если в конце текущей записи стоит маркер "конец файла", иначе выключен. Биты 2...7 - не используются.

IX+19 V_RECLEN- длина текущей записи в буфере.

IX+1B V_BUFFER- буфер, хранящий текущую запись. Ниже мы привели пакет процедур, необходимых для организации вышеописанной концепции. Чтобы открыть файл последовательного доступа на RAM-диске, необходимо предварительно в аккумуляторе процессора выставить номер потока, подключенного к каналу. Имя же самого файла хранится в системной переменной N_STR1 по адресу 5B67 (23399). Далее вызывается процедура V_OPEN (B988). Есть и другие точки входа. Это OPEN_4, OPEN_5, CLOSE_4 и CLOSE_5. Вызовом OPEN_4 Вы открываете на RAM-диске файл с условным именем FILE_1 и подключаете его к четвертому потоку.

Точно так же вызовом OPEN_5 открывается файл последовательного доступа с условным именем FILE_2 и подключается к пятому потоку.

Процедуры CLOSE_4 и CLOSE_5 закрывают эти файлы и отключают эти потоки.

Таким образом, вновь созданный нами канал "V может легко быть задействован из БЕЙСИКа. Нижеприведенная программа (Листинг 3) демонстрирует файлы RAM-диска в работе сначала в качестве WRITE-файлов, а затем в качестве READ-файлов.

Конечно, Вы вовсе не обязаны называть свои файлы именами FILE_1 и FILE_2, также как и не обязаны подключать к ним только потоки #4 и #5. Для этого в пакете процедур имеется более общая точка входа V_OPEN, при входе в которую, как уже было указано, регистр A процессора должен содержать номер подключаемого потока, а системная переменная N_STR1 - имя файла (если имя меньше положенных 10 символов, пробелы делаются последующими).

Листинг 19.

Данная процедура предназначена для закрывания "V" канала. Основным в ней является опорожнение буфера, если в нем что-то есть для файла, открытого для записи. Для файла, открытого для чтения, содержимое буфера может игнорироваться.

CD005B	V_CLOSE	ORG B960	
2A5A5B		CALL 5B00, SWAP	; Впечатывание страницы 0.
E5		LD HL, (BETADDR)	
DDE5		PUSH HL	; Запомнили адрес возврата.
2A3D5C		PUSH IX	
		LD HL, (ERR_SP)	; HL указывает на адрес
			; процедуры обработки
			; ошибки.
E5		PUSH HL	; Запомнили указатель.
21FEFF		LD HL, FFFE	
39		ADD HL, SP	; В HL помещен указатель
			; стека минус 2.
E33D5C		LD (ERR_SP), HL	; Новый адрес возврата.
DDCB1846		BIT 0, (V_CHFLAG)	; Если файл является файлом
C4F2B8		CALL NZ, B8F2, V_STORE	; открытым для записи.
			; Здесь отгружается буфер.
E1		POP HL	; Напомним, что данный
			; адрес при работе V_STORE
			; служил адресом возврата
			; при любой ошибке.
223D5C		LD (ERR_SP), HL	
DDE1		POP IX	
2158E7	V_OC_EXIT	LD HL, 2758	
D9		EXX	
C358BB		JP B858, V_EXIT	; Переход на выход.

Листинг 20.

Эта процедура открывается наиболее универсальной точкой входа. Процедура предназначена для открывания канала. При входе в процедуру аккумулятор процессора должен содержать номер потока, подключенного к данному каналу, а в десяти байтах системной переменной N_STR1 должно быть записано имя открываемого файла. Если имя файла менее десяти символов, то пробелы выполняются как следующие.

CD005B	V_OPEN	ORG B988	
2A545B		CALL 5B00, SWAP	; Впечатывание страницы 0.
E5		LD H, (RETADDX)	; Установка адреса возвра-
F5		PUSH HL	; та в ПЗУ-0.
3E56		PUSH AF	; Запомнили номер потока.
CD94B5		LD A, 56	; CHR\$ 55 = "V"-имя канала.
		CALL B594	; Проверка на существование
			; такого канала.
381F		JR C, V_OP_OK	; Переход, если его нет.
DDES	V_OP_LOOP	PUSH IX	
E1		POP HL	; HL указывает на область
			; информации по уже сущес-
			; твующему каналу.
010E00		LD BC, 000E	; HL указывает
09		ADD HL, BC	; на имя файла.
11675B		LD DE, N_STR1	; DE-адрес имени файла.
060A		LD B, 0A	; 0AH=10 DEC-длина имени.
1A	V_OP_NAME	LD A, (DE)	
13		INC DE	
2006		JR NZ, V_OP_RETRY	; Переход, если имя не
			; совпадает.
10F8		DJNZ V_OP_NAME	; Переход на вершину цик-
			; ла для проверки всех
			; десяти символов.
CDFCB6	V_OP_ERROR	CALL B6FC, V_ERROR	; Генерация сообщения
20		DEFB 20	; "e: File already exists"
			; "Файл уже существует".
1656	V_OP_RETRY	LD D, 56	; CHR\$ 56 = "V"-имя канала.
CDAAB5		CALL B5AA	; Поиск еще одного канала
			; с тем же именем "V".

30E1 CD08B7	V_OP_OK	JR NC V_OP_LOOP CALL B708,V_FIND	; Возврат, если найден. ; Поиск имени файла на ; RAM-диске.
F5 2811		PUSH AF JR Z,V_OP_CONT	; Сохранение флага Z. ; Переход, если файл не ; найден, т.е. Вы открыва- ; ете файл "для записи".
DD6E0A DD660B DD7E0C CDFFB6		LD L,(SF_START) LD H,(SF_START+1) LD L,(SF_START+2) CALL B6FF,V_PAGE	; AH указывает на начало ; файла с данным именем ; на RAM-диске. ; Выбор страницы, содержа- ; щей первый байт файла. ; A - код типа файла.
7E FE04 20DE		LD A,(HL) CP 04 JR NZ, V_OP_ERROR	; На обработку ошибки, если ; это не READ-Файл.
3E05 CDFFB6	V_OP_CONT	LD A,05 CALL B6FF,V_PAGE	; "Впечатывание" стандарт- ; ного ОЗУ. ; Возврат флага Z.
F1 08 F1 08 F5		POP AF EX AF,A`F` POP AF EX AF,A`F` PUSH AF	; A-номер потока. ; A`-номер потока. ; Сохранение флага Z, кото- ; рый указывает тип Файла. ; CHR\$ 56 = "V"-имя канала, ; длина блока информации о ; канале "V".
3E56 011B02		LD A, 56 LD BC,021B	; DE-адрес процедуры ввода. ; HL-адрес процедуры вывода ; IX-адрес закрывающей про- ; цедуры.
11D4B7 212BB9 DD2160B9		LD DE,B7D4,V_INPUT LD HL,B92B,V_PRINT LD IX,B9650,V_CLOSE	; Вызов калькулятора. ; Это уже не команды ма- ; шинного кода Z80 - это ; команды кода калькулятора ; о котором мы писали в ; нашем трехтомнике по про- ; граммированию в машинных ; кодах.
EF6DB0 DEFB 6D DEFB B0		RST 28	; Здесь на стек калькулято- ; ра помещается адрес про- ; цедуры OPEN_NEW (B06D). ; Адрес блока информации о ; канале переводится из IX ; в HL.
DDE5 E1		PUSH IX POP HL	
010B00 09 70 23 70 23 70 23 EB 21675B 0E0A EDB0		LD BC,000B ADD HL, BC LD (HL),B INC HL LD (HL),B INC HL LD (HL),B INC HL EX DE, HL LD HL,5B67,N_STR1 LD C,0A LDIR	; B HL адрес V_CHBYTE. ; Обнулили V_CHBYTE. ; Обнулили V_CHREC. ; B DE адрес V_CHNAME. ; B HL адрес имени файла. ; 0AH=10 DEC-длина имени ; копирование имени файла ; в блок информации о канале.
F1 2809		POP AF JR Z,V_OP_WRITE	; Восстановление флага Z. ; Переход, если файл - ; "для записи".
DDCB1686 CDC5B7		RES 0,(V_CHFLAG) CALL B7C5,V_ASSIGN	; Сигнал: "Файл для чтения" ; Привязка буфера к RAM ; диску.

1840		JR V_OP_EXIT	; Переход на выходную
			; процедуру.
DDCB18C6	V_OP_WRITE	SET 0, (V_CHFLAG)	; Сигнал: "Файл для записи"
CD02B7		CALL B703, V_NEWCAT	; Создание новой записи в
			; каталоге.
21FFFF		LD HL, FFFF	
7C		LD A, H	; В аккумуляторе минус 1.
CD05B7		CALL B705, V_SPACE	; Проверка на достаточность
			; места для вставки байта.
3E04		LD A, 04	; Напомним: каталог - на
			; странице 4.
CDFFB6		CALL B6FF, V_PAGE	; Выбор страницы 4.
DD6E0A		LD L, (SF_START)	; ANL указывает на первый
DD660B		LD H, (SF_START+1)	; свободный байт (старый)
DD7E0C		LD L, (SF_START+2)	; на RAM-диске.
F5		PUSH AF	; Запомнили код страницы.
CDFFB6		CALL B6FF, V_PAGE	; Выбор страницы, содержа-
			; щей первый свободный байт
F3		POP AF	; ANL указывает на первый
			; свободный байт.
3604		LD (HL), 04	; Записали 04 в качестве
			; байта "тип файла".
010100		LD BC, 0001	
CD3AB7		CALL B73A, ADD_HL_BC	; ANL указывает на первый
			; свободный байт.
5F		LD E, A	; То же и EHL.
3E04		LD A, 04	
CDFFB6		CALL B6FF, V_PAGE	; Выбор страницы 4.
DD7510		LD (SF_END), L	; Откорректировали каталог
DD7411		LD (SF_END+1), H	; под конец адресных данных
DD7312		LD (SF_END+2), E	; данного файла.
CD0BB7		CALL B70B, V_CATEND	; Оформили конец каталога.
3E05		LD A, 05	
CDFFB6		CALL B6FF, V_PAGE	; Выбрали стандартное ОЗУ.
C381B9	V_OP_EXIT	JP B981, V_OC_EXIT	; Переход на выходную
			; процедуру.

Листинг 3

```

1000 REM WRITE FILE DEMO
1010 RANDOMIZE USR 47713: REM открываем OPEN#4, Файл "FILE_1"
1020 RANDOMIZE USR 47720: REM открываем OPEN#5, Файл "FILE_2"
1030 FOR i=1 TO 512
1040 INPUT "": PRINT i
1050 FRINT #4; 2*i
1060 PRINT #5; i*i
1070 NEXT i
1080 RANDOMIZE USR 47736: REM закрываем CLOSE#4, Файл "FILE_1"
1090 RANDOMIZE USR 47740: REM закрываем CLOSE#5, Файл "FILE_2"
1100 REM READ FILE DEMO
1110 RANDOMIZE USR 47713: REM OPEN#4
1120 RANDOMIZE USR 47720: REM OPEN#5
1130 FOR i=1 TO 512
1140 INPUT #4; a
1150 INPUT #5; b
1160 PRINT a,b
1170 NEXT i
1180 RANDOMIZE USR 47736: REM CLOSE#4
1190 RANDONIZE USR 47740: REM CLOSE#5
1200 STOP

```

Ниже приведены листинги процедур, предназначенные для манипуляции с файлами на RAM-диске 128-килобайтных моделей.

Надо сразу оговориться, что существуют две основные отличающиеся друг от друга версии ПЗУ 128-килобайтных машин. Отличия касаются нулевой страницы ПЗУ. Первая

версия - это более ранняя модель "ZX-Spectrum+128", а вторая - "ZX-Spectrum+2". Поскольку в ПЗУ этих машин адреса некоторых процедур отличаются, то чтобы не писать два отдельных пакета для той машины и для другой, вводится таблица адресов переходов (вектор переходов). Вы можете использовать либо ту, либо другую. Если ПЗУ Вашей модели отличается и от той и от другой версии, то Вы имеете в принципе возможность (если у Вас хватает информации о точках входа своего ПЗУ) подкорректировать эти таблицы (Листинги 4,5) так, как Вам угодно. Во всем остальном приведенный пакет процедур идентичен и для той и для другой модели.

Листинг 21.

Последняя процедура служит для интеграции всего вышеприведенного пакета процедур с БЕЙСИКом компьютера. Приведенные в ней строки являются образцами и Вы можете переделать их под свои конкретные требования. OPEN_4 открывает файл FILE_1 и подключает к нему поток номер 4. OPEN_5 подключает поток 5 к открываемому файлу FILE_2. CLOSE_4 и CLOSE_5 соответственно закрывают четвертый и пятый потоки.

		ORG BA4D	
46494C4531	FILE_1	DEFM "FILE1"	
2020202020		DEFM " "	; Имя файла+5 пробелов.
46494C4532	FILE_2	DEFM "FILE2"	
2020202020		DEFM " "	; Имя файла+5 пробелов.
3E04	OPEN_4	LD A, 04	; A - номер потока.
214DBA		LD HL, FILE_1	; HL-адрес имени файла.
1805		JR OPEN_4_5	; Переход на открывание ; потока.
3E05	OPEN_5	LD A, 05	; A - номер потока.
2157BA		LD HL, FILE_2	; HL-адрес имени файла.
11675B	OPEN_4_5	LD DE, NSTR1	
010A00		LD BC, 000A	
EDB0		LDIR	; Кодирование имени файла ; в системную переменную.
C388B9		JP B988, V_OPEN	; Переход на открывание ; канала.
3E04		LD A, 04	
1802		JR CLOSE_4_5	; Переход на открывание ; канала.
3E05		LD A, 05	
C300B0		JP B0000	; Закрываем каналы.

MACHINE CODE

Сегодня своими приемами создания игровых программ в машинных кодах делится наш постоянный читатель Антон Александрович Яицкий.

Нам особенно приятно напечатать эту статью потому, что это первая ласточка от наших отечественных программистов. Ну не все же время читать Стива Тернера. Ведь есть же и у нас прекрасные профессионалы, имеющие свой опыт, свои секреты и приемы и, главное, готовые поделиться с теми, кто делает первые шаги в освоении глубин компьютера.

При создании программ в машинных кодах (например игровых) с использованием АССЕМБЛЕРА Z-80 очень трудно бывает сосредоточиться на самой программе, т.к. приходится рассеивать внимание на создании различных подпрограмм.

Есть весьма простой способ обойти это препятствие. И его алгоритм легко применим.

1) Сначала пишутся независимые подпрограммки - модули, которые являются законченными, т.е. оканчиваются кодом 201 (RET -возврат).

2) Потом они компилируются в машинный код с помощью компилятора (СИ, Ассемблер и т.п.) и размещаются в памяти таким образом, чтобы их координаты были известны.

3) На БЕЙСИКе пишется программа, которая управляет их работой. Это главная программа и ее сразу трудно написать в машинных кодах, поскольку ее приходится постоянно изменять, переделывать, корректировать и т.п.

4) Когда увязка процедур с помощью головной БЕЙСИК-программы закончена, переносим ее на бумагу со всеми точными данными - паузами, переходами, определениями. Это и будет алгоритм основного блока программы, другими словами "телом" будущей игры.

5) Далее переводим эту программу в язык низкого уровня, пользуясь ею как алгоритмом. Т.к. процедуры, вызываемые ею, уже отлажены, то ее отлаживать не так сложно. Во всяком случае нетрудно обнаружить место происхождения той или иной ошибки, раз алгоритм (сценарий игры) перед Вами и процедуры нижних этажей проверены и должны работать. Будут, конечно, ошибки, допущенные при переводе Бейсика в машинный код, но Вы с ними справитесь.

Располагать в памяти компьютера "тело" главного управляющего блока следует так, чтобы вместе с подготовленными ранее подпрограммами она представляла единое целое. После старта она будет вызывать в нужной последовательности те же процедура, что и БЕЙСИК-программа, повторяя ее алгоритм.

Какие при этой должны быть учтены нюансы?

1) Во-первых, есть разница в быстродействии. БЕЙСИК-программа работает гораздо медленнее, чем машинный код. Вследствие этого откомпилированная программа будет работать несколько не так, как ранее отлаженная в БЕЙСИКе ее предшественница. Это может повлиять на динамику управления программой и т.п. Поэтому заранее такое явление необходимо учитывать.

Надо предварительно продумать и определить, для каких частей Вашей программы зависимость от скорости БЕЙСИКа будет наиболее критической. Их лучше сразу написать на Ассемблере или СИ и откомпилировать, а из БЕЙСИКа вызывать через RANDOMIZE USR addr, где addr - адрес расположения данной подпрограммы.

2) Могут возникать ошибки при исполнении переходов. В Ассемблере адрес относительного перехода до инструкции JR n не должен отстоять более, чем на 128 байтов от точки исполнения перехода. Поэтому, если в независимых процедурах использовать эту команду желательно, то при написании головного блока ее использовать нельзя. Следует пользоваться абсолютным переходом JP nn или вызовом CALL.

Какие же модули в первую очередь нуждаются в быстродействии? Это прежде всего все выводы на экран (например спрайтов). Это смена экрана по команде LDIR (LDDR). Это процедуры компрессии/декомпрессии экранов и процедуры, управляющие

вводом/выводом информации.

Когда все будет сделано, Вам останется только дать команду: RANDOMIZE USR begin.

По всей видимости надо дать и несколько примеров перевода операторов БЕЙСИКа в стандартные конструкции в машинных кодах.

1) Часто встречающаяся в БЕЙСИКе команда PAUSE (пауза). Рассмотрим например PAUSE 200 (пауза на 4 секунды).

Для этой цели предлагается процедура WAIT. Она требует, чтобы заранее в регистр A (аккумулятор) было заслано число, определяющее длину этой паузы.

```
LD A, 200
CALL WAIT
```

Процедура WAIT будет ждать 4 секунды и вернется туда, откуда был сделан вызов. А вот как она выглядит:

```
WAIT      LD HL, 23672
          LD (HL), 0
WAIT1     CP (HL)
          RET Z
          JR WAIT1
```

Здесь использован младший байт системной переменной FRAMES (23672), которая как известно служит как бы внутренними часами компьютера.

2) Так же полезной может быть и процедура TXT, обеспечивающая вывод текстового сообщения на экран.

В вызывающей процедуре нужно сделать:

```
.....
          CALL TXT
          DEFB 22
          DEFB y
          DEFB x
          DEFM "ZX SPECTRUM"
```

Вместо y и x Вы подставите те значения, которые обычно в БЕЙСИКе Вы пишете после PRINT AT (x,y). Сама же печатающая процедура может выглядеть например так:

```
TXT      POP HL
TXLOOP   LD A, (HL)
          CP 0
          JR Z, TXEND
          RST 16
          INC HL
          JR TXLOOP
TXEND    INC HL
          PUSH HL
          RET
```

В данном случае процедура использует для печати процедуру ПЗУ RST 16. Использование процедур ПЗУ следует применять шире - это позволяет значительно экономить память и упрощает программирование.

Такой способ написания программ значительно облегчает трудоемкое занятие отслеживания правильности программирования и отладки.

* * *

Наш комментарий

В заключение работы Антона Александровича нам хотелось бы высказать ряд и своих соображений, а также внести некоторые дополнения, полезные тем, кто сейчас делает первые шаги в машинном коде или собирается их делать.

Конечно, приемов организации программирования в машинном коде может быть столько же, сколько и программистов, но как нам кажется, у предлагаемой системы есть ряд очень весомых преимуществ, которые должны заинтересовать наших читателей.

1. Доступность.

Пожалуй, самое приятное в ней для тех, кто освоил БЕЙСИК, и никак не может решиться на освоение АССЕМБЛЕРа - то, что прямо сейчас он скажет сам себе "Все, я уже программирую на Ассемблере", после чего вставит вместо оператора PAUSE в своих программах команду RANDOMIZE USR pause. Затем по адресу pause запишет несколько

байтов из приведенного выше образца и встанет из-за компьютера с гордо поднятой головой. Да, действительно он уже программирует с использованием машинного кода и не заметил, как преодолел сложный психологический барьер.

Постепенно подготавливая одну за другой процедуры, можно незаметно перейти к качественно новому уровню работы с машиной.

2. Наглядность.

Это обстоятельство даже не нуждается в комментариях.

3. Структурность.

Важная особенность состоит в том, что Вы можете готовить программы коллективно.

Предположим, что вместе с группой единомышленников Вы решили подготовить интересную игровую программу. В одиночку на это уйдет год и Вы решили сократить время, распараллелив работу между людьми и поручили каждому подготовку процедур и данных по одному направлению:

- один пишет все процедуры, связанные с выдачей на экран текстовых сообщений;
- другой пишет все процедуры, связанные с управлением программой от клавиатуры или от джойстика;
- третий - управление работой звукового динамика;
- четвертый - процедуры, связанные с экраном - построение экрана, помещение спрайтов, перемещение спрайтов, компрессия и декомпрессия экранов; пятый вообще ничего писать не умеет (он - художник) и рисует в редакторе ARTSTUDIO экран за экраном; а шестой - пишет музыку.

Последний - руководитель проекта. Он собирает все то, что сделано остальными, объединяет разнородные куски в единое целое с помощью своего главного БЕЙСИКовского блока. Он координирует работу остальных, назначает адреса, через которые передаются параметры из одной процедуры в другую, отводит те или иные размеры памяти участникам проекта по их запросу.

Вот Вы уже и создали свою фирму. Теперь пару лет кропотливой работы и можете выходить на международный уровень. В Вашей фирме найдется место и художнику и композитору и специалисту по иностранным языкам и менеджеру и даже специалисту по проведению маркетинговых исследований. Добавьте отдел распространения, рекламный отдел, бухгалтерию и отдел кадров и назначьте себя Генеральным Директором или хотя бы президентом.

4. Из того факта, что программа сделана из многочисленных мало зависящих друг от друга модулей вытекает и возможность создания частных и коллективных библиотек.

Когда Ваша группа закончит работу над своим проектом и решит создать новый, то окажется, что значительный пакет процедур может быть перенесен из предыдущего проекта без каких-либо изменений.

Поэтому сразу надо начинать работу не над одной программой, а над сериалом и, прорабатывая процедуры, учитывать возможность их использования и в других программах тоже.

Так у Вас будет создана библиотека. Придет время и Вы напишете на БЕЙСИКе нехитрую программу, которая прокрутит на магнитофоне одну-две кассеты с Вашими библиотеками, выберет из них те процедуры, которые Вам нужны в сегодняшнем проекте, загрузит их в память компьютера и объединит в единый модуль.

И в заключение несколько практических советов "ИНФОРКОМа".

I

Мы рекомендуем всем, кто пишет в машинных кодах, завести для себя единый стандартный блок, разместив его в раз и навсегда отведенной для него области памяти. Мы называем этот блок - керналь (KERNAL). Вот для чего он нужен.

Вы, конечно, понимаете, что абсолютное большинство созданных Вами стандартных процедур должны обмениваться друг с другом и с головной программой какими то данными. Например, процедура, которая занимается переброской экрана должна как минимум получить два параметра - адрес, начиная с которого хранится блок (2 байта) и длину этого блока (2 байта). Более того, если экран хранится в сжатом виде, то его еще

надо декомпрессировать. Поэтому мало принять эти два параметра - их еще надо передать в процедуру декомпрессии. Так отведите раз и навсегда 4 байта памяти для хранения параметров, используемых процедурой переброски экрана. Пусть это будут скажем адреса с 60000 по 60003. далее отведите 4 байта для параметров, передаваемых в процедуру, выполняющую музыкальное сопровождение - с 60004 по 60007. и т.д. У Вас образуется некий блок размером 1...2 К, который Вы всегда сможете вставлять в любые свои программы. Сегодня Вы не знаете, в каких адресах у Вас завтра будет находиться звуковоспроизводящая процедура, но Вам об этом и не надо думать. Размещайте ее где угодно. Ей это все равно, если и сегодня и завтра и всегда она будет получать свои данные из одних и тех же адресов и отправлять результат работы в одни и те же ячейки. Вот данные в этих ячейках могут меняться, но сами адреса - никогда.

Более того, если Вы делаете программу, в которой нет ни стрельбы ни бомбежки, а в Вашем кернале есть ячейки, отвечающие за передачу параметров в процедуры бомбометания и стрельбы, пусть они там и будут. Оставьте их, чтобы не сдвигать раз и навсегда сделанный кернал. Невелика потеря памяти, зато в конечном итоге экономятся годы труда.

И, наконец, в кернале можно хранить не только данные. В нем же можно хранить и адреса каких-то процедур. Посмотрите, как организован вектор переходов в статье "Потоки и каналы". Одним словом, принцип керналя - тот же, по которому организован раздел системных переменных в "Спектруме", только там он сделан К. Синклером в поддержку его ПЗУ, а Вы сделаете его в поддержку всех своих программ и он будет постепенно год от года наращиваться по мере накопления вами библиотек и личного опыта.

II

И последний практический совет. Он касается хранения текстовых сообщений (впрочем то же относится и к хранению в памяти экранов и прочих данных).

В целях упрощения подачи материала наш автор в примере процедуры TXT смешал в процедуре и машинный код и данные. То есть текстовое сообщение "ZX SPECTRUM" им размещено там же в процедуре, где и сам код. Адрес начала сообщения, как Вы видите из примера, передается через стек командой POP HL (а значит кто-то должен был заранее позаботиться и заготовить его там). Длина сообщения не определена, вместо этого маркером конца сообщения служит CHR\$ 0 (операция проверки на конец - CP 0).

В качестве примера этот вариант годится, да видимо именно так и надо представлять идею, чтобы не усложнять ее излишними деталями, но теперь, в качестве комментария, мы должны сказать, что при написании реальных программ делают не так.

Размещать текстовые сообщения в процедурах вообще- то нельзя. Дело в том, что текст сообщения, как и любые данные, Вам потом может захотеться изменить, и тогда если изменится его длина, то после его вставки в процедуру переместятся команды, и в результате "поплывут" все команды относительных переходов JR, что очень неприятно.

Поэтому все сообщения хранят в едином блоке (таблице), отведя ей какой-то фиксированный участок памяти, например с 53000 и выше (только последите, чтобы он не перекрыл кернал или другую какую-либо таблицу). Более того, сам адрес начала этой таблицы текстовых сообщений можно хранить в кернале, отведя ему раз и навсегда 2 байта.

Теперь Вы можете разместить в таблице сколько хотите сообщений.

Например:

58000 - Сообщение1

58007 - Сообщение2

.....

59567 - Сообщение250

Вызывается процедурой TXT то или иное сообщение по его номеру, который должен быть установлен предварительно в аккумуляторе. Например вызов 15-го сообщения пройдет так:

```
LD A,15
CALL TXT
```

Теперь осталось выяснить каким образом процедура TXT найдет пятнадцатое сообщение, если адрес его ей не известен. Во-первых, ей известно начало таблицы

сообщений (его можно взять из керналя). Во вторых, вводится еще одна таблица - указательная таблица, в которой хранится величина смещения начала каждого сообщения от начала таблицы сообщений. На каждое смещение надо дать 2 байта и тогда если в Вашей программе 250 сообщений, то указательная таблица займет 500 байтов. Предположим, что мы ее разместим, начиная с адреса 57500.

Дальше логика такова. Номер сообщения Вам известен - 15. Перед ним стоят 14 сообщений. Поскольку в указательной таблице на каждое сообщение уделено 2 байта, то его надо удвоить - получим 28. Начало указательной таблицы Вам известно - 57500 (кстати его тоже надо брать из керналя, ведь оно может меняться от программы к программе). Теперь определяем сумму $57500+28=57528$. Значит в адресах 57539 и 57530 хранится величина смещения в таблице сообщений. Определим ее. Пусть мы получили 1560. Теперь из керналя вводим базовый адрес таблицы сообщений. У нас это 58000. " Прибавляем к нему величину смещения 1560 и получаем 59560 - адрес, начиная с которого содержится ваше искомое сообщение.

А вот как это выглядит на практике см. Листинг-1.

И ряд дополнительных практических замечаний:

1. Хранение и вызов экранов по их номеру организуется точно так же.
2. Атрибуты, с которыми печатается текст (цвет фона, символа, признаки мигания, яркости, координаты позиции печати и т.п. хранятся вместе с текстом).
3. Управляющие коды, такие как CHR\$ 13, CHR\$ 6 и т.п. хранятся вместе с текстом.
4. В приведенных примерах не задана длина текстового сообщения. Компьютер определяет, что оно закончилось по появлению кода CHR\$ 0 (NOP). Это не вполне экономично, т.к. сколько у Вас в программе сообщений, столько байтов Вы потратите на хранение этих маркеров. Обычно применяют другой способ - такой, какой применен в ПЗУ компьютера. Последний символ каждого сообщения увеличивают на 128, т.е. включают старший (седьмой бит). При проверке же на конец сообщения проверяют седьмой бит и если он включен, то печатают последний символ и выполняют возврат.
5. При переброске экранов удобнее не маркировать конец экрана, а хранить в указательной таблице длину каждого закомпрессированного экрана или вычислять ее через разность соседних значений в указательной таблице.
6. При переброске спрайтов указательная таблица вообще не нужна. Спрайты имеют как правило стандартный размер, скажем 32 байта каждый. Тогда по номеру спрайта всегда в таблице спрайтов можно найти его начало и конец. Адрес начала таблицы спрайтов хранится в кернале. Там же может храниться и максимальное число спрайтов в Вашей программе.
7. После всего вышесказанного Вы можете представить структуру своей будущей программы примерно в таком виде: (см. рис. 1).

.....
Головной блок.
Рабочие процедуры.
Область Ваших программных переменных.
Указательная таблица экранов.
Таблица закомпрессированных экранов.
Таблица графики спрайтов.
Таблицы шрифтов (если они нужны)
Указательная таблица сообщений
Таблица текстовых сообщений
К Е Р Н А Л Ь
.....

Рис. 1

Обратите внимание на то, что еще до начала написания программы Вы должны примерно прикинуть распределение памяти и решить сколько байтов и на что Вы отведете. При программировании же должны тщательно следить за тем, как Вы укладываетесь в намеченные пределы. Впрочем, это было лучше описано у Стива Тернера в статье "Профессиональный подход".

Обратите внимание также на то, что адреса границ этих программных областей - это тоже вещь, которую можно хранить в кернале.

Мы надеемся, что материал этой статьи не вызовет больших сложностей в понимании у начинающих, а профессионалам напомним, что мы ориентируемся в первую очередь не на них.

Зато спешим порадовать всех. В развитие данной темы в будущем году (причем в первом полугодии) мы напечатаем замечательную книгу Стюарта Николса "Применение АССЕМБЛЕРА для создания игровых и других скоростных программ на компьютере "ZX SPECTRUM".

Перевод подготовлен по нашему заказу специально для "ZX-РЕВЮ" Пашориным Валерием Ивановичем (г. Балашов, Саратовской обл.)

В значительной степени книга и посвящена вопросам конверсии БЕЙСИКа в машинный код.

Листинг 1.

Предположим, что в кернале базовый адрес указательной таблицы хранится, скажем, в 61245, а базовый адрес таблицы сообщений - в 61247.

TXT	AND A	;Этой безобидной операцией мы сбрасываем ;флаг переноса без изменения аккумулятора ;(это делается на всякий случай).
	LD HL,(61245)	;В пару HL загружается начальный адрес ;указательной таблицы
	ADD A,A	;Удвоение содержимого аккумулятора. Обра- ;тим внимание на то, что если в нем было ;установлено число большее, чем 127, то ;после удвоения аккумулятор переполнится ;и включится флаг переноса.
	LD E,A	;Удвоенное содержимое аккумулятора пере-
	LD D,00	;сылается в регистровую пару DE.
	JR NC,BYPASS	;Если аккумулятор не переполнялся, то ;следующую операцию надо обойти.
	INC D	;В противном случае надо увеличить регистр ;D на единицу.
BYPASS	ADD HL,DE	;Теперь в DE хранится удвоенный номер на- ;шего сообщения. Прибавим его к начальному ;адресу указательной таблицы, хранящемуся ;в HL.
	LD E,(HL)	;Ищем в нем смещение. Младший байт смеще- ;ния отправляем в регистр E.
	INC HL	;Переходим к старшему байту.
	LD E,(HL)	;Отправляем его в регистр D.
	LD HL,(61247)	;В пару HL загружается начальный адрес ;таблицы Ваших текстовых сообщений.
	ADD HL,DE	;Прибавив его к взятой из таблицы величине ;смещения, находим адрес начала сообщения
TXLOOP	LD A,(HL)	;Загружаем в аккумулятор первый символ ;сообщения.
		;Далее все также, как у автора статьи.
	CP 0	
	JR Z,TXEND	
	RST 16	
	INC HL	
	JR TXLOOP	
TXEND	INC HL	
	RET	

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД

Окончание. Начало на стр. 49,70,117,143,167.

В прошлый раз мы с Вами рассмотрели вопрос как избегать ошибок при программировании, как их находить и устранять. Сегодня мы заканчиваем печать статей Стива Тернера под общим заголовком "Профессиональный подход". Правда это не значит, что мы навсегда прощаемся с этим выдающимся программистом. Может быть ему не раз еще удастся поделиться с нами маленькими хитростями и своими профессиональными секретами.

Итак, вернемся к плану тестирования готовой программы с целью поиска в ней возможных ошибок "мин замедленного действия".

Ниже приведена программа "Монитор реального времени" (MONITR) и на примере тестирования ее самой же и показано, как можно составить план тестирования.

В отличие от большинства других программ-мониторов эта очень мала и позволяет запускать Вам Вашу программу под ее управлением. Во время работы Вашей программы Вы сможете вести контроль за своими переменными и отслеживать их изменение. Ее применение не исключает возможность применения внешнего более общего монитора. Наоборот, монитор реального времени служит как бы дополнением к нему. Чтобы в дальнейшем не путаться между двумя мониторами, давайте будем называть монитор реального времени резидентным монитором, а главный монитор - просто монитором.

Когда Ваша игровая программа работает, MONITR непрерывно ее сопровождает, но может ее прервать и передать управление главному монитору для глубокого анализа состояния программы в момент прерывания.

Вы можете значительно ускорить процесс тестирования программы благодаря тому, что Вам предоставлена возможность выставлять значения своих переменных перед тестовым запуском. Это неоценимо важно при настройке игровых программ. Вы сможете вносить изменение в какие-либо программные константы, от которых зависит скорость передвижения персонажей, интенсивность стрельбы, позиционирование объектов на экране и многое тому подобное. Все это позволяет достичь максимальной играбельности готового программного продукта.

	MONITR	DL	; Определение метки.
3A27B1		LD A, (MONON)	; Включение.
A7		AND A	; Проверка "Включен/выключен?"
200F		JR NZ, ISON	; Переход, если включен.
01FEF7		LD BC, F7FE	; Задание адреса порта
			; клавиатуры. Выставлен верхний
			; ряд, левый полуряд.
ED78		IN A, (C)	; Прием информации от клавиатуры.
F6FE		OR FE	; Маскирование 7-ми младших
			; битов.
FEFF		CP FF	; Проверка не была ли нажата
			; клавиша "1".
C8		RET Z	; Поскольку MONON=0 и клавиша "1"
			; не нажата - возврат.
3E01		LD A, 01	; Если клавиша "1" была нажата,
3227B1		LD (MONON), A	; изменяем значение MONON.
210040	ISON	LD HL, 4000	; организация замедляющего цикла.
2D	DELAY	DEC L	
20FD		JR NZ, DELAY	
25		DEC H	
20FA		JR NZ, DELAY	
	KEYSCAN	DEFL	; Определение метки.
01FEEF		LD BC, EFFE	; Установка адреса порта
			; клавиатуры. Выставлен верхний
			; ряд, правый полуряд.

ED78		IN A, (C)	;Чтение данных с порта. Поступил ;байт, имеющий вид: ;111???? - здесь и далее: ;1 - бит включен; ;0 - выключен; ;? - не определен.
F6E2		OR E2	;Включение тех битов, которые ;нас не интересуют. ;E2 HEX = 1110 0010 BIN, имеем ;в итоге: 111? ??1?
6F 01FEF7		LD L, A LD BC, F7FE	;Сохранили значение. ;Установка адреса порта ;клавиатуры. Выставлен верхний ;ряд, левый полуряд.
ED76		IN A, (C)	;Чтение данных с порта. Поступил ;байт, имеющий вид 111? ????
17		RLA	;Теперь байт имеет раскладку: ;11?? ???1. Мы гарантированно ;включили младший нулевой бит, ;взяв его из 7-го бита. ;DF HEX = 1101 1111, имеем в ;итоге: 11?1 1111.
F6DF		OR DF	
A5 FEFE		AND L CP FE	;В аккумуляторе будет FE тогда и ;только тогда, когда нажата ;клавиша 0 и никакие другие ;клавиши верхнего ряда не ;нажаты.
2003 C30FF7		JR NZ, NOQUIT JP EXITAD	;Переход для продолжения работы. ;Если была нажата клавиша "0", ;здесь выполняется переход в ;главный монитор по адресу 0FF7, ;заданному константой EXITAD. ;Если Вы не пользуетесь ;монитором фирмы PICTURESQUE, ;задайте здесь свой адрес.
FEFF CA36B0	NOQUIT	CP FF JP Z, OPTYPE	;Если никакая клавиша не нажата, ;остаемся в режиме 1.
1F 1F		RRA RRA	;Биты, выключаемые при нажатии ;клавиш "0" и "9" нас не интере- ;суют и мы их "прокручиваем".
0E0		LD C, 0	;C - выполняет роль счетчика ;оборотов байта.
1F	FINDK	RRA	; "Крутим" байт, пока не найден ;нулевой бит, т.е. нажатую ;клавишу.
3003		JR NC, KEYNO	;Как только найдется нулевой ;бит, выполняется переход в ;KEYNO.
18FA		JR FINDK	;Переход на очередное ;вращение байта.
79	KEYNO	LD A, C	;Сколько оборотов сделали? ;Отсюда и будем вычислять какая ;клавиша была нажата. ;позиция курсора MONDIG.
2128B1 1129B1		LD HL, MONDIG LD DE, MONADD	
FE03		CP 3	;3 оборота - клавиша "5".
2811		JR Z, LEFTCR	;Переход на "курсор влево".
FE02		CP 2	;2 оборота - клавиша "6".
2316		JR Z, DOWN	;Переход на "вниз".
FE01		CP 1	;1 оборот клавиша "7".
2816		JR Z, UP	;Переход на "вверх". ;В противном случае начинаем ;обработать режим "курсор

7E		LD A, (HL)	; вправо".
3C		INC A	; Ввели позицию курсора.
FE07		CP 07	; Увеличили ее.
3028		JR NC, OPTYPE	; Позиция не может быть более 6.
77		LD (HL), A	; Продолжение работы в режиме 1.
1825		JR OPTYPE	; Запомнили позицию курсора в
7E	LEFTCR	LD A, (HL)	; MONDIG.
FE00		CP 0	; Продолжение работы в режиме 1.
2820		JR Z, OPTYPE	
3D		DEC A	
77		LD (HL), A	
1B1C		JR OPTYPE	; Продолжение работы в режиме 1.
0EFF	DOWN	LD C, FF	; Ввод начального значения.
1802		JR TURNBT	; Переход на изменение.
0E01	UP	LD C, 1	; Ввод начального значения
	TURNBT	DEFL	; Определение метки.
7E		LD A, (HL)	; Ввод позиции курсора.
47		LD B, A	; Запомнили ее в В.
2600		LD H, 00	
CB2F		SRA A	; Деление на 2, поскольку инди-
			; кация каждого байта на экране
			; занимает 2 позиции, то поделив
			; номер позиции пополам, получим
			; номер байта.
6F		LD L, A	
19		ADD HL, DE	; HL указывает на младший или
			; старший байт адреса, на который
			; настроен монитор.
78		LD A, B	; Старший или младший
E601		AND 01	; полубайт?
79		LD A, C	
2006		JR NZ, TOMONB	
17		RLA	; Если младший, то вращаем его
17		RLA	; четыре раза.
17		RLA	
17		RLA	
E6F0		AND F0	
86	TOMONB	ADD A, (HL)	; Установленное значение отпра-
77		LD (HL), A	; ляется на хранение в
			; программную переменную.
112E31	OPTYPE	LD DE, (MONLIN)	; MONLIN указывает на начало
			; выходного буфера.
3A29B1		LD A, (MONADD)	
67		LD H, A	
3A2AB1		LD A, (MONADD+1)	; Теперь в HL –
6F		LD L, A	; анализируемый
3A2CB1		LD A, (MONOP)	; адрес.
A7		AND A	; A содержит код операции.
2806		JR Z, PEEK	; Проверка на 0.
FE20		CP 20H	; Переход на процедуру ис-
2808		JR Z, POKE	; полнения режима PEEK.
			; Если там "пробел" –
180A		JR NULLOP	; Переход на процедуру ис-
			; полнения режима POKE.
7E	PEEK	LD A, (HL)	; В противном случае обраба-
			; тывается режим 0.
322BB1		LD (MONVAL), A	; Снимается значение из ана-
			; лизируемого адреса
1804		JR NULLOP	; и запоминается в программ-
3A2BB1	POKE	LD A, (MONVAL)	; ной переменной MONVAL.
77		LD (HL), A	; Продолжение работы.
			; Засылка байта в адрес.

2129B1	NULLOP	LD HL, MONADD	; В HL - адрес выходного буфера. ; В нем хранится образ экранной строки.
CD07B1		CALL HEX	; Для первой пары позиций вызывается конвертер перевода 16-тиричного кода в ASCII.
23		INC HL	; Пропуск второй позиции 1-ой пары.
CD07B1		CALL HEX	; Для второй пары позиций.
13		INC DE	; Вводится пробельная позиция.
23		INC HL	;
CD07B1		CALL HEX	; 3-я пара.
13		INC DE	; Пробел.
23		INC HL	
CD07B1		CALL HEX	; Обрабатывается последняя позиция.
3E20		LD A, 20H	; 20H - это код пробела.
3237B1		LD (BLANK), A	; Подготовка к "стиранию"
3A28B1		LD A, (MONDIG)	; Номер позиции.
210058		LD HL, 58000	; Адрес в дисплейном файле.
0620		LD B, 20H	; Счетчик цикла.
360F	COLMON	LD (HL), 0F	; Выставление
23		INC HL	; экранных
10FB		DJNZ COLMON	; атрибутов.
210056		LD HL, 58000	; Адрес в дисплейном файле.
FE04		CP 04	; 4-ая позиция пробельная и
3806		JR C, YE	; не обрабатывается.
23		INC HL	
FE04		CP 06	; 6-ая позиция пробельная и
3801		JR C, YE	; не обрабатывается.
23		INC HL	
1600	YE	LD D, 0	
5F		LD E, A	
19		ADD HL, DE	
3630		LD (HL), 30H	; Изменение цвета в позиции курсора.
3A29B1	PUTMON	LD A, (MONADD)	; Подготовка
67		LD H, A	; к печати
3A2AB1		LD A, (MONADD+1)	; содержимого
6F		LD L, A	; ближайших
1138B1		LD DE, MONDMP	; восьми байтов.
13		INC DE	
0604		LD B, 04	; Сначала 4 байта.
CD07B1	HEXDMP	CALL HEX	; Вызов 16-ричного конвертера.
23		INC HL	
10FA		DJNZ HEXDMP	; Основание цикла.
0604		LD B, 04	; еще 4 байта.
13		INC DE	; Пробельная позиция.
CD07B1	NEXTWO	CALL HEX	; Вызов 16-ричного конвертера.
23		INC HL	
10FA		DJNZ NEXTWO	; Основание цикла.
11000		LD DE, 0000H	; Установка начальной позиции печати строки на экране.
212DB1		LD HL, MONOUT	
CD4AB1		CALL PRINT	; Вызов процедуры, печатающей результаты работы программы на экране. Можете здесь поставить свой адрес.
C9		RET	; Конец работы и возврат в исполняемую программу.

Конвертер 16-ричного кода в код символа ASCII.

AF	HEX	XOR A	; Обнуление аккумулятора.
ED6F		RLD	; Вращение полубайта, подхватывая его значение из адреса (HL)
12		LD (DE), A	
13		INC DE	
ED6F		BLD	; Второй полубайт.
12		LD (DE), A	
ED6F		RLD	; Восстановление значения в адресе, на который указывает HL, в исходное состояние.
1B		DEC DE	
1B		DEC DE	
0E02		LD C, 2	; Ввод параметра цикла.
1A	DIGIT	LD A, (DE)	
FE0A		CP 0AH	
3004		JR NC, DOLET	; Если число больше 9 и должно изображаться буквой - переход на DOLET.
C630		ADD 30H	; Перевод цифры в код ASCII.
1802		JR DONE	; Конец.
C637	DOLET	ADD 37H	; Перевод буквы в код ASCII.
12	DONE	LD (DE), A	
13		INC DE	
0D		DEC C	
20F0		JR NZ, DIGIT	; Возврат к началу цикла.
C9		RET	; Возврат в вызывающую программу.

Таблица программных переменных.

00	MONON	DB 0	; 0 - выкл. ; 1 - вкл.
00	MONDIG	DB 0	; Позиция курсора.
00AF	MONADD	DW AF00	; Адрес вызова внешнего монитора.
00	MONVAL	DB 0	; Значение анализируемого адреса
00	MONOP	DB 0	; Код режима работы.
1C	MONOUT	DB 1C	; Количество символов в выходном буфере.
00000000	MONLIN	DEFM	; Содержимое выходного буфера для адреса и содержимого байта.
20000020			
00			
20	BLANK	DB 20H	; Символ "пробел".
20000000	MONDMP	DEFM	; Содержимое выходного буфера для восьми ближайших байтов
00000000			
00			
20000000			
00000000 00			

Чтобы резидентный монитор сопровождал Вашу игру, Вам надо в самой игре сделать периодический его вызов. Лучше всего это делать где-то в главном управляющем цикле. Я, например, этот вызов (CALL) вставляю в ту процедуру, которая занимается сканированием клавиатуры в поисках нажатой пользователем клавиши, так что у меня он работает даже когда игра стоит в ожидании чего-то.

Первое, что делает резидентный монитор - он обеспечивает выход из вашей игры, если нажата клавиша "0" и вход в главный монитор. Этот вход делается по методу установки точки прерывания, чтобы отработав в главной мониторе, можно было бы вновь вернуться и продолжить игру.

Я не знаю каким монитором в качестве главного пользуетесь Вы и поэтому не знаю какие команды он вставляет, когда устанавливает точку прерывания, но Вы можете легко это сами установить.

Работая со своим монитором, (например MONS 3) установите где-либо точку

прерывания, а потом посмотрите, что он туда подставил. Это будет либо команда CALL, либо JUMP. В зависимости от прочитанного адреса перевода замените в нижеприлагаемой программе монитора реального времени переменную EXITAD на то, что Вы там нашли. В приведенной распечатке там стоит адрес F70F (63247) - адрес вызова программы MONITOR 48 фирмы PICTURESQUE, которым я и пользуюсь.

Начинать процесс мониторинга игры надо из главного монитора. Стартуйте его, чтобы правильно прошла настройка его собственных переменных, а потом из него пойдите в тестируемую программу.

Поместить вызов резидентного монитора Вы можете в любом цикле Вашей программы, в зависимости от того, что Вы тестируете в данный момент. Он может вам даже дать возможность прервать исполнение зацикленной программы.

Что позволяет Вам делать резидентный монитор? Во время работы Вашей программы он изображает в верхней части экрана:

- адрес контролируемой Вами ячейки;
- содержимое данной ячейки;
- код режима работы;
- содержимое ближайших восьми байтов.

Печать на экране выполняется с помощью процедуры PRINT, которую мы рассматривали ранее (стр. 119).

На экране информация выглядит так:

```
6000  00  1  00010203  04050607
|      |      |
адрес  |      |  содержимое 8-ми
содержимое |      |  соседних байтов
           |      |
           |      |  код операции
```

Управляющие клавиши:

- 0 - прерывание работы резидентного монитора и переход в главный монитор.
- 1 - включение резидентного монитора.
- 5 - курсор влево (выбор цифры, подлежащей изменению);
- 8 - курсор вправо (выбор цифры, подлежащей изменению);
- 6 - изменение цифры вниз;
- 7 - изменение вверх.

Так, перемещаясь между цифрами, изображаемыми на экране, Вы можете их изменять. Например, Вы можете изменить адрес, содержимое которого изображается на дисплее. Вы можете изменить режим работы резидентного монитора.

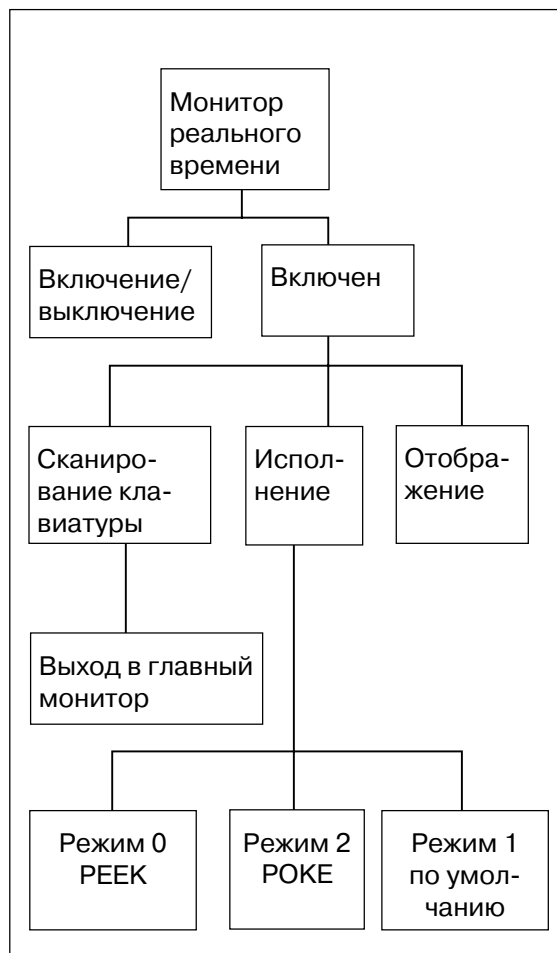
Всего он имеет три режима.

Режим - 0. В этом режиме происходит считывание (PEEK) содержимого заданного адреса и изображение его на экране.

Если Вы не успеваете отслеживать происходящие изменения, то "замедлить" работу программы можно перейдя в режим 1. Это основной режим, принимаемый по умолчанию. В этом режиме можно задать байт и задать адрес для исполнения POKE.

Режим 2 служит для засылки выставленного байта в выставленный адрес. После выполнения этого вернитесь в режим 0. Оставлять резидентный монитор в этом режиме опасно.

Для справки я привожу структурную схему монитора реального времени.



Теперь на примере программы MONITR рассмотрим план тестирования реальной программы. Ниже перечислены точки, в которых можно выполнить тестирование и намечено, что в этих точках подвергается проверке.

1. MONITR.

Никаких действий не должно выполняться, если переменная MONON равна нулю, т.е. если не нажата клавиша "1".

2. MONITR.

При нажатии "1" MONON должна принимать значение = 1.

3. MONITR.

Если MONON = 1 процедура должна запускаться и работать при нажатии любой клавиши, кроме "0".

4. DELAY.

Проверяется не слишком ли быстро работает программа.

5. KEYSKAN.

Должна прерываться работа при нажатии клавиши "0".

6. KEYNO.

Не должно происходить никаких изменений при нажатии любых клавиш, кроме "5", "6", "7", "8".

7. KEYNO.

При нажатии клавиши "8" (курсор вправо) должно возрасть значение переменной MONDIG, но не более 6.

8. LEFTCR.

При нажатии клавиши "5" должно уменьшаться значение MONDIG, но не ниже, чем до 0.

9. DOWN.

При нажатии клавиши "5" значения цифры на экране должны убывать.

10. UP.

При нажатии клавиши "6" значения цифры на экране должны возрастать.

11. TURNBT.

Проверяется изменение второй цифры в изображаемом на экране содержимом байта.

12. TURNBT.

Проверяется изменение второй пары цифр в изображаемом адресе.

13. OPTYPE.

Проверяется влияние переменной MONOP на режим работы программы. Выполняется ли PEEK, когда MONOP равно 0? Выполняется ли POKE, когда MONOP = 2? При любых других значениях ничего не должно выполняться.

14. PEEK.

Читается ли содержимое памяти?

15. POKE.

Вводится ли в память установленное значение байта?

16. HEX.

Вывод на экран соответствует шестнадцатеричному представлению чисел?

17. NULLOP.

Формат печати информации на экране соответствует требуемому?

18. COLMON.

Есть ли курсор на экране? он передвигается? Он перепрыгивает через промежутки между числами? После прохождения курсора по числу его цвет восстанавливается исходный?

19. PUTMON.

Проверяется правильность изображения на экране данных по 8-ми соседним байтам. Также проверяется их занесение в буфер MONDMP.

20. NEXTWO.

Окончательно проверяется качество печати.

И, в заключение, пару интересных наблюдений.

Может быть Вы заметили, что план тестирования программы в значительной степени отражает структурную диаграмму, и это не случайно. Если бы было желание дробить диаграмму до более мелких деталей, Вы бы увидели, что они вообще идентичны. Каждому блоку на диаграмме соответствовал бы один тест. Отсюда важный вывод - еще и не приступив к работе над программой, занимаясь только ее проектированием и исполнением структурных диаграмм, Вы уже закладываете основы и возможность будущей отладки и задаете надежность будущей программы.

Как говорится, что заложите, то и получите и остается только еще раз повторить: "Сначала тщательное проектирование - программирование потом".

И второе. Несмотря на то, что список возможных тестов оказался достаточно длинным проверка всего того, что в нем понаписано занимает у меня 30 секунд! (Конечно если все работает, но ведь и в этом тоже надо убедиться).

ADVENTURE PROJECT

Окончание.

Начало см. на стр. 122, 193.

Взаимодействие между играющим и программой - это один из важнейших вопросов, оказывающих влияние на качество программы. Между тем в программах адвентюрного жанра это взаимодействие организуется посредством текстового ввода (довольно архаичная, хотя и любимая многими манера).

Большинство адвентюрных программ используют технологию интерпретаторов для выяснения того, какую же команду хотел дать пользователь. В минимальной виде он включает конструкции типа Глагол + Существительное. В этом случае глагол выполняет функции команды, а существительное указывает на объект, к которому эта команда относится и, тем самым, служит как бы параметром при команде. Например программа поймет команду GO NORTH (Идти на север), но вполне может не понять команду PLEASE GO NORTH (Пожалуйста иди на север) или, еще того хуже, GO DOWN THE ROAD TO THE NORTH (Идти по дороге на север).

Таким образом, словарь который понимает программа, конечно ограничен. Раздвинуть пределы этих ограничений - это основная цель каждой фирмы, занимающейся выпуском адвентюрных программ. В то время как лучшие экземпляры достигают нескольких тысяч слов, типовым размером словаря минимальной (учебной) программы являются примерно 200 слов, которые организуются в программе в виде базы данных.

Вместе с тем, организация базы данных и работы с ней, это еще не все те проблемы, которые требуется решить. Прежде всего, если Вы хотите, чтобы Ваши персонажи могли взаимодействовать друг с другом и обмениваться сообщениями, необходимо подумать над специальными приемами. Они ведь не смогут общаться по системе глагол + существительное.

Мы в своем проекте пока обходились без общения между персонажами и вернемся к Оре и Инфу, которые победили монстров в глубине океана, проникли на затонувшую яхту и оказались в подземном мире короля Кельроса.

Под охраной подземных жителей их провели в столицу страны и теперь они ждут допроса у короля.

Теперь необходимо взводить программные средства для общения с персонажами. Дело-то ведь не в том, что Вы хотите сообщить им, куда им следует пойти, им еще надо провести переговоры с королем.

Чтобы не изобретать велосипед, мы вернемся к технике, которая уже была опробована в адвентюрных играх.

Когда в конце 60-х на гигантских вычислительных машинах, имевших огромные размеры памяти, но совершенно негибкие и неэффективные операционные системы, появились первые текстовые игры, они были по сегодняшним меркам невероятно примитивными и имели словарь порядка 100 слов. Как же они работали?

Приняв команду, например GO NORTH, они делили ее на две части и в каждой оставляли только два первых символа, отрезая остальные. Так что у Вас оставалось GO NO, после чего начиналась трансляция.

Трансляция проводилась одним из двух возможных способов. Либо была одна центральная база данных, в которую были засланы все возможные слова, которые программа могла интерпретировать, либо в соответствии со сценарием программы она разбивалась на сцены и для каждой сцены хранился свой допустимый набор слов. Второй способ требовал больше памяти, но зато был гибче первого, поскольку в первом случае приходилось в одной базе хранить слова самых разных типов, участвующих в программе: команды перемещения, боевых действий, названия предметов и др.

Второй способ хоть и требует большего расхода памяти, имеет предпочтение в большем просторе алгоритмизации и кодирования, особенно на БЕЙСИКе и ограничения, накладываемые на программу здесь не столь жестки.

Например, в какой-то локации Вам встретилась кошка. Вполне может быть, что Вы захотите ее погладить и программа может это предусмотреть, но если Вы захотите ее накормить, а вокруг нет и не было никакой еды, то и предусматривать здесь нечего.

Если же Вы окажетесь в сцене, где будет еда, то связанные с ней слова будут предусмотрены, но зато не надо обрабатывать слова, имеющие отношение к контактам с кошкой, а при наличии рядом и кошки, можете ее и накормить, но это уже как бы третья сцена: "кошка + еда".

Преимущество такого подхода еще и в том, что Вы подгоняете словарь под контекст реальной ситуации и он может быть хорошо продуман. При образовании же центральной базы данных Вам необходимо иметь в голове столько возможных вариантов, что все равно что-то не предусмотрите и пользователь будет разочарован.

С тех пор прошли годы, было изобретено множество новых приемов. Техника программирования становится все изощреннее и все ближе подходит к искусству, по так или иначе новые приемы зачастую не снижают естественных ограничений, а нередко и добавляют их.

Например, такими технологиями управления адвентюрными играми является управление через меню, пиктограммы или с помощью ключевых слов. Например Вы хотите куда-то пройти: - нажимаете клавишу G. На экране появляется запрос "Куда?". Вы нажимаете клавишу N (На север) и т.д. а если Вам надо что-то взять (TAKE), то первым нажатием будет "T". Такая техника обычно используется в программах, имеющих центральную базу данных, а не систему, опирающуюся на сценарии.

Обе системы имеют свои за и против и нам надо определиться с двумя вопросами - "Что мы хотим иметь?" и "Как мы это будем делать?" и именно в таком порядке, а не наоборот. Нередки еще программисты, которые садятся писать что-то, полагая, что по ходу дела все прояснится и пути откроются туда, куда надо сами собой.

Нам нужен метод коммуникации достаточно удобный, чтобы не раздражать пользователя и достаточно простой, чтобы не раздражать программиста (об элегантности мы здесь не говорим, поскольку мы обсуждаем идеи, а не пишем реальную программу).

Мы вам предложим две системы обеспечения коммуникации. Первая простая, но ограниченная, а вторая - позволяет вводить команды от пользователя полными предложениями, но нуждается в корректировке при переходе от сцены к сцене, т.е. одни действия возможны в одной сцене, но невозможны в другой.

Система ввода через меню или ключевые слова может быть использована в простейшей подпрограмме ввода и будет работать таким образом, чтобы введенные слова одинаково интерпретировались во всех сценах Вашей игры. Пример для ключевых слов GO ("G") и TAKE ("T") приведен на листинге 1.

Пример открыт для расширения и Вы можете добавить к нему столько вариаций, сколько захотите. Все, что Вам требуется - это строка, перехватывающая ввод после строки 30.

В качестве примера объект введен в переменную в строке 5. В реальной программе в каждой сцене здесь должны вводиться соответствующие сцене объекты. Например, в сцене могут быть ведро и лопата на берегу моря. Это может быть кинжал, лежащий на палубе яхты и т.п.

Строка 10 принимает от пользователя команду в виде одного символа и помещает его в переменную a\$. Строки 20 и 30 проверяют что это за команда - GO или TAKE. Если принятый символ не является ни "g" ни "t", тогда компьютер печатает в нижней части экрана "Не понимаю!" и ждет другой команды.

Если была нажата клавиша "g", то это команда GO и выполняется переход на строку 60 для анализа этой команды.

Строка 60 - начало блока управления сценой. Переменные n,s,e,w,u,d - соответствуют тем шести направлениям, в которые может захотеть переместиться играющий. Содержатся же в этих переменных номера строк, к которым следует перейти, чтобы отработать поданную пользователем команду. Ноль указывает на то, что в данном направлении перемещение невозможно.

Строка 70 вызывает подпрограмму, выполняющую ввод направления перемещения от пользователя и проверку на допустимость этого направления. Если оно недопустимо, в соответствующую переменную вводится ноль.

После возврата в главный блок проверяется значение переменной В. Если это ноль, пользователю выдается сообщение о невозможности перемещения в данном направлении и ему предлагается изменить команду.

Если же все прошло нормально, Вы переходите к следующей сцене по номеру строки, содержавшемуся в В.

Подпрограмма, занимающаяся командой TAKE расположена начиная со строки 100. Игрока спрашивают о том, какой предмет он хочет взять и вводят его в а\$.

Переменная о\$ хранит информацию о том объекте, который может быть взят в данной сцене. Если о\$ = "" (пустая строка), программа сообщит о том, что нет объектов в данной комнате.

Если а\$ не совпадает с о\$, пользователя предупреждают о том, что данный предмет взять нельзя, он не здесь. Если же взять предмет удалось, вам об этом сообщают и данный предмет заносится в Ваш инвентарный список (переменная i\$, которая отслеживает пополнение Вашей коллекции). Символ "*" служит для отделения объектов друг от друга и список может быть просмотрен, например с помощью программы, о которой мы уже говорили (см. стр. 194).

Листинг 1.

```
5 LET o$="SWORD":LET i$=""
10 PAUSE 0: LET a$=INKEY$: IF a$="" THEN GO TO 10
20 IF a$="g" THEN GO TO 60
30 IF a$="t" THEN GO TO 100
40 PRINT #0;"Я Вас не понимаю"
50 GO TO 10
55 REM Начало процедуры перемещения.
60 LET n=0: LET d=0: LET s=4000: LET w=5000: LET e=5050: LET u=4050
70 GO SUB 1000
80 IF B=0 THEN FRINT "Вы не можете идти в этом направлении": GO TO 10
90 GO TO B
100 INPUT "Что взять?"; a$
110 IF a$="" THEN FRINT "Здесь нет предметов, которые можно было бы взять.": GO TO 10
115 IF a$<>o$ THEN PRINT "Этот предмет находится не здесь.": GO TO 10
120 PRINT "вы берете ";o$
130 LET i$=i$+"*"+o$
140 GO TO 10
1000 INPUT "Куда идти? ";b$: LET b$=b$(1)
1010 IF b$="n" THEN LET b=n: RETURN
1020 IF b$="s" THEN LET b=s:RETURN
1030 IF b$="e" THEN LET b=e:RETURN
1040 IF b$="w" THEN LET b=w:RETURN
1050 IF b$="u" THEN LET b=u:RETURN
1060 IF b$="d" THEN LET b=d:RETURN
1080 RETURN
4000 CLS: PRINT "Вы находитесь в тронном зале короля Кельроса. Сверкающий трон поражает своим великолепием. Во взгляде властелина читается жестокая решимость. "
4010 GO TO 10
4050 CLS: PRINT "Оттолкнув стражу, Вы бросаетесь вверх по лестнице и, преодолев ее, попадаете в пустой коридор. Все боковые двери закрыты. Дальше дороги нет. Погоня настигает, сопротивляться бесполезно. "
4060 GO TO 10
5000 CLS: PRINT "Воспользовавшись секундным замешательством стражи с криком 'Бежим!' Вы хватаете Ору за руку и бросаетесь к выходу. "
5010 GO TO 10
5050 CLS: PRINT "Собрав все силы, Вы вырываетесь из рук стражи и бежите к ближайшей двери, но она оказывается закрытой. Погоня настигает Вас, под тяжелыми ударами Вы падаете и теряете сознание."
5060 GO TO 10
```

Это то, что касается первой технологии коммуникации.

Теперь рассмотрим второй прием, позволяющий вводить длинные предложения.

Итак, наши герои предстали перед повелителем подземного царства, который восседает на троне, а рядом с ним на столике лежит цель Вашего приключения - похищенные часы, управляющие ходом развития Вселенной. Вы должны вернуть бесценный прибор любыми усилиями.

Фактически перед ними три пути. Либо попытаться убежать, либо схватить часы и попытаться убежать, либо попробовать договориться мирным путем.

Переведя все это на язык компьютера, мы получаем: GO (идти) или TAKE AND GO (взять и идти), или SAY TO (сказать).

Рассмотрим эти варианты и прикинем как их можно реализовать в нашей экспериментальной программе. Итак, если мы хотим, чтобы пользователь мог вводить свои команды длинными предложениями нормальным человеческим языком, нам необходимы программные средства, чтобы выделить из его заявления ту ключевую информацию, которая действительно играет роль.

В нашем примере мы рассмотрим именно эти три функции GO, TAKE и SAY TO. Давайте сразу отметим, что в любом языке одно и то же действие можно выразить по-разному, но тем не менее возможный набор синонимов все-таки ограничен. Так, например вместо "взять", можно употребить "схватить", "украсть", "похитить" и даже что-либо типа "умыкнуть", но надо все-таки признаться, что последнее очень маловероятно. Скорее всего пользователь будет применять TAKE или GET и их и надо искать в его команде. Если же программа пишется на русском языке, то имеет смысл настраивать ее на "взять" и на "хват" (от слов схватить, захватить). Если же он употребит иные слова, то ему можно сообщить, что мысль его в общем-то понятна, но не мог ли бы он попробовать ее выразить другими словами. Итак, мы остановимся на TAKE и GET. Точно также и с перемещением GO. Способов перемещения не так много - GO, RUN, MOVE (идти, бежать, передвигаться) и пожалуй из общеупотребимых это все. Направления же возможного движения вообще известны однозначно: СЕВЕР, ЗАПАД, ЮГ, ВОСТОК, ВВЕРХ, ВНИЗ (NORTH, SOUTH, WEST, EAST, UP, DOWN). Только уж если Вы захотите создать шедевр искусственного интеллекта, Вы сделаете так, что программа будет понимать вперед и назад (FORWARD, BACK).

Отсюда начинается самая трудная часть работы. Для процедуры, описывающей перемещение, Вам необходимо, чтобы она просмотрела введенную фразу и нашла есть ли в ней GO, затем определила есть ли в ней указание на направление и еще раз проверила нет ли других указаний.

При таком подходе пользователь сможет ввести команду: "Весело идти по дороге вымощенной желтым кирпичом на север и потом повернуть на запад" и будет правильно понят.

Если выделить в такой команде ключевые слова, то она будет выглядеть так:

(...ИДИТИ) (...СЕВЕР...) (...И...) (...ЗАПАД...). GO...NORTH...AND...WEST

GO - основной носитель команды в этой фразе и искать этот глагол надо в первую очередь. Если он не найден, то значит надо проверить прочие глаголы-команды, которыми может воспользоваться играющий.

Если же он найден, то ищется направление, в данном случае NORTH, затем проверяется нет ли других команд в этой фразе (ищется AND) и если и это слово найдено, тогда дешифрируется и вторая часть команды.

В смысле алгоритма это означает, что сначала ищется подстрока "GO" в строке введенной пользователем. Подпрограмму поиска подстроки в строке может написать наверное всякий, владеющий хоть каким-нибудь языком программирования, но тем не менее в прилагаемом примере (Листинг 2) есть один вариант.

Если она находит GO, то выполнит переход в подпрограмму, которая занимается вопросами перемещения и проанализирует остальную часть фразы.

Аналогично может строиться и строка команды TAKE (GET).

Например:

GET...CLOCK...AND...TORCH...

ВЗЯТЬ...ЧАСЫ...И...ФАКЕЛ...

Теперь рассмотрим команду SAY TO (сказать). На первый взгляд она может

представлять проблему. Нет, особых проблем нет, решается все той же техникой, хотя немного посложнее. Именно ее-то мы и рассмотрим в конкретном примере (Листинг 2), не отвлекаясь на TAKE, GO и т.п.

Во-первых, установим, что команда должна состоять из трех основных разделов и программа должна с ними соответственно и разобраться. На первом этапе компьютер выделяет ключевое слово SAY. Делается это точно также, как для GET и GO. Далее управление передается подпрограмме, обслуживавшей SAY, которая анализирует оставшуюся часть команды.

Следом за командой SAY TO естественно должно стоять имя того, к кому речь обращена. на втором этапе программа проверяет присутствует ли данный персонаж в данной сцене. Если его нет, выдается соответствующее сообщение.

Если он есть, проверяется намерен ли он разговаривать. Это не значит, конечно, что программа должна анализировать его настроение (хотя и это возможно, раз у нас есть критерии оценки его состояния), все гораздо проще. Вы ведь не можете готовить программу предусмотреть диалоги с абсолютно всеми персонажами и поэтому отделяетесь от диалога с тем, для кого он Вами не предусмотрен. сообщениями типа "Он сейчас не в настроении для беседы и не отвечает на Ваше обращение" или, скажем, "Вам лучше обратиться к кому-нибудь другому по этому вопросу."

Если диалог для этого персонажа Вами предусмотрен, тогда декодируется остальная часть Вашей команды. Это скорее всего либо просьба (указание), либо вопрос. Кстати, отличить их легко по наличию или отсутствию вопросительного знака в ходе прямой речи. Но сейчас речь не об этом. То, что один персонаж говорит другому, стоит в кавычках и, выделив строку, стоящую в кавычках, Вы получаете как бы независимую команду, с которыми мы уже умеем работать.

Конструкция получается типа:

СКАЗАТЬ КОРОЛЮ "...ДАТЬ...ЧАСЫ..

" SAY TO KING "...GIVE....CLOCK..."

Возможно, конечно, что Вы захотите дать пользователю возможность строить очень головоломные конструкции типа: "Идти на север и потом повернуть на восток. Взять веревку и ведро и спросить у старика с палкой: "Где колодец?"."

Все это вполне реализуется серией условных переходов с логическим оператором IF. Компьютер воспринимает точку как разделитель команд. Выполняет все, что положено до точки. Потом то, что осталось до следующей точки и так до конца команды. Союз "И" (AND) выполняет, правда, двойную функцию. Он может объединять в одной команде два объекта, на которые направлено действие, а может и объединять два действия. Различить то и другое несложно, если посмотреть какого типа ключевые слова стоят по обе стороны от "И". Если это существительные (объекты), то это первый случай, а если одно из них глагол (команда из словаря команд), то это второй случай - тут-то и действует серия IF. Тогда компьютер разобьет Вашу длинную команду на серию коротких и, пока будет исполнять первую, остальные поддержит временно в каком-то буфере (в какой-либо переменной).

Тогда Ваша длинная последовательность упростится до:

ИДТИ...СЕВЕР...И...ВОСТОК...

ВЗЯТЬ...ВЕРЕВКА...И...ВЕДРО...

СПРОСИТЬ СТАРИКА...

...КОЛОДЕЦ ?

Когда Вы программируете на английском языке, мало проблем с приставками, окончаниями слов и т.п., но на русском это проблема, которую программа, ищущая подстроку в строке, должна решить за счет сокращения словарных слов до 3-х 4-х коренных букв и их-то и искать.

ИДТИ...СЕВЕ...И...ВОСТ...

ВЗЯТ...ВЕРЕ...И...ВЕДР...

СПРО...СТАР...

...КОЛО?

Практика показывает, что если Вы делаете словарь игры примерно на 200 слов, то сокращение до трех букв проходит нормально. Если же словарь подходит к тысяче слов, то безусловно четыре буквы необходимы.

В нижеприведенном образце программы рассмотрен вариант обработки SAY TO. Программа ждет, что Вы обратитесь к королю. Он правда не очень разговорчив и подарив ему свой меч SAY TO KING 'TAKE MY SWORD', Вы уловите момент, когда он отвлечется рассматриванием подарка, схватите часы и броситесь бежать. А уж как вам удастся вывести героев на поверхность - это дело Ваше.

Листинг 2

```
5 DIM m(10):DIM c$(10,32)
10 CLS
11 REM Ввод команды играющего.
20 INPUT a$
21 REM Первый этап анализа команды.
30 LET nend=2
31 LET c$(1)="G"
32 LET c$(2)="S"
33 LET m(1)=100
34 LET m(2)=200
35 LET e=20
36 REM Правильнее конечно было бы вводить эти данные через READ/DATA, но так для примера
    нагляднее.
37 GO SUB 1000
38 GO TO gt
100 LET nend=1
110 LET c$(1)="G0"
120 LET m(1)=300
130 LET e=200
140 GO SUB 1000
150 GO TO gt
200 LET nend=1
210 LET c$(1)="SAY TO"
220 LET m(1)=400
230 LET e=20
240 GO SUB 1000
250 GO TO gt
300 REM Здесь расположен блок анализа перемещения. Мы не будем его рассматривать в данном
    примере.
310 REM .....
398 REM .....
399 STOP
400 REM Второй этап анализа команды SAY TO.
410 LET nend=1
420 LET c$(1) = "TO KING"
430 LET m(1)=500
440 LET e = 900
450 GO SUB 1000
460 GO TO gt
500 REM Поиск прямой речи. Предположим, что она должна быть ограничена слева и справа не
    кавычками, а апострофами.
510 LET nend=1
520 LET c$(1)="'"
530 LET m(1)=600
540 LET e = 910
550 GO SUB 1000
560 GO TO gt
600 REM Найден первый апостроф! Теперь можно найти и второй.
605 LET a$=a$(k+1 TO)
610 LET nend=1
620 LET c$(1)="'"
630 LET a(1)=700
640 LET e=920
650 GO SUB 1000
```

```

660 GO TO gt
700 REM Высказывание определено и теперь можно его анализировать.
705 LET a$a$( TO K)
710 LET nend=1
720 LET c$(1)="TAKE"
730 LET m(1)=800
740 LET e=930
750 GO SUB 1000
760 GO TO gt
800 REM Король понял, что он может принять от Вас что-то в подарок, осталось только понять
    что именно.
805 LET a$a$( TO K)
810 LET nend=2
820 LET c$(1)="SWORD"
830 LET m(1)=890
840 LET e=940
850 GO SUB 1000
860 GO TO gt
890 PRINT "король благожелательно улыбнулся, беря Ваш меч в свои руки и внимательно
    рассматривая его. Внимание охраны несколько рассеялось, все внимательно смотрят на
    короля."
899 STOP: REM Наступает момент, когда Вы можете схватить часы и попытаться убежать.
900 PRINT "Попробуйте обратиться к другому персонажу"
905 GO TO 30
910 PRINT "Ваше обращение к персонажу - это прямая речь и ее надо ограничить апострофами
    слева и справа."
915 GO TO 20
920 PRINT "Вы забыли закрыть прямую речь справа."
925 GO TO 20.
930 PRINT "король грозно смотрит на Вас и о чем-то шепчется со своими воинами. Пока Вы не
    нашли нужных слов для спасения собственной жизни. ""
935 GO TO 20.
940 PRINT ""Я не понимаю, что можешь предложить мне ты, несчастный чужестранец?'- грозно
    спрашивает король. Стража напряглась, в зале воцарило напряженное ожидание. ""
945 GO TO 20
1000 REM Анализатор подстроки в строке.
1005 LET kend=LEN (a$a)
1006 LET ktest=LEN(c$a)
1010 FOR k=1 TO kend
1020 FOR n=1 TO nend
1030 IF a$a(k TO k+ktest-1)=c$(n) THEN LET gt=m(n):GO TO 1070
1040 LET gt=e
1050 NEXT n
1060 NEXT k
1070 RETURN

```

Еще раз обращаем Ваше внимание на то, что приведенный пример лишен элегантности именно потому, что это учебный пример. В Вашей реальной программе Вы конечно же должны использовать тот факт, что строки 100...200...300... и т.д. имеют одинаковую структуру. Введя подпрограмму, которая будет универсальной для всех этих блоков и вводя данные не через LET, а через READ-DATA, Вы во много раз сократите объем Вашей программы.

Заканчивая этот материал, мы еще раз хотим вам напомнить, что мы не ставили целью научить Вас составлять игровые программы. Это дело очень и очень сложное, к которому надо идти кропотливым трудом не один год. Мы только напомним, что основное направление деятельности "ИНФОРКОМа" - обучающие программы, а хорошая обучающая программа строится по тем же самым принципам, что и игровая.

Сейчас очень много классов в наших школах оборудуются "Синклер"-совместимыми машинами. С ними работают тысячи учителей. Им надо дать в руки инструмент быстрого реагирования, чтобы они уже завтра могли хоть что-то давать своим ученикам, обучая их не "информатике", как она понимается в наших нелепых учебниках, а именно "компьютерной грамотности". Умению ставить перед собой задачи и находить пути их решения. Вот для них-то идеи, изложенные в эти статье, и должны пригодиться в первую очередь.

Эти идеи и концепции - канва, на которую каждый нанесет то, что ему сегодня нужнее и важнее, завтра он сам определится, что для него самое важное и подберет другую канву или создаст свою, оснастит ее программно-инструментальными средствами, наполнит своими библиотеками и обогатит личным опытом, но все делается постепенно и начинать надо с простого.

Обучающие программы

Очень и очень многие пишут о том, как нужны им обучающие программы. Сегодня для них радостное сообщение пришло из Бухары. Наш читатель Подкорытов Л.К. выполнил русификацию пакета TUTOR. Для тех, кто не знает, сообщаем - это 40 уроков АССЕМБЛЕРА и этот пакет был бы замечательным дополнением к нашему трехтомнику по программированию в машинных кодах. Мы, к сожалению, перегружены сейчас собственными проектами и не сможем принять маркетинг этого пакета на себя, но полагаем, что те, кому он нужен, смогут обратиться к т. Подкорытову напрямую по адресу:

705022, г. Бухара-22, ул. Ульянова, д.82. кв.115.

Для тех, кто тоже работает в этом направлении мы всегда дадим "Зеленую улицу". И вот Вам адреса для потенциального партнерства:

ИЩЕМ ОБУЧАЮЩЕ ПРОГРАММЫ ДЛЯ ПК "SPECTRUM" ПО СПЕЦИАЛЬНОСТЯМ:

автодело, медицина, радиомонтаж, кулинария, вычислительная техника и программирование, психология, строительство, торговля, швейное дело.

Нужны программы по "Правилам дорожного движения".

Покупаем хорошие программы по всем предметам за курс начальной, восьмилетней и средней школы.

Обработаем поступающую информацию и вышлем всем желающим.

Наш адрес: 659700, г. Горно-Алтайск, ул. Маресьева 11/1. Межшкольный учебно-производственный комбинат. Сарычевой Т.А.

ДЛЯ ЭКСПЛУАТАЦИИ В ВУЗЕ

требуются обучающие программы для ПК "Спектрум".

453135, Стерлитамак-25, а/я 75 Вережкину Игорю Дмитриевичу.

И еще одна нужная вам информация:

Малое предприятие "РИТМ" просит отозваться всех программистов, пишущих игровые программы для "Спектрума".

Контактный адрес: 400009, Волгоград, пр. Ленина, д.129, кв.45. Харионовскому А.В.

FORUM

Очень интересный вопрос, имеющий некоторое историческое содержание задает наш читатель из г. С.-Петербурга Тасев А.Д.

"Меня и многих моих знакомых - почитателей "Спектрума" интересует вопрос, связанный с полем экрана компьютера. Если взять такие известные компьютерные системы, как MSX, IBM и многие другие, так там исходный экран - черный, а текст - белый. Это естественно, ведь это меньше нагружает глаза.

Почему же К.Синклер в своем компьютере ввел белый фон с черным текстом? Что за этим стоит? Или это только рекламный трюк? Да и вообще какое сочетание является более эргономичным для глаз, если работа идет с цветным дисплеем?"

При кажущейся простоте вопрос читателя не праздный - за ним стоит идеология архитектуры всей системы.

Во-первых давайте признаем, что действительно белый текст на черном фоне гораздо более эргономичен и это важно. Одно дело - если пользователь работает с компьютером 2 часа в день и совсем другое, если 14.

Теперь давайте вспомним, что в MSX, IBM и многих других системах, о которых говорит наш читатель, четко разделены режимы работы. Есть текстовый режим и соответственно текстовый экран и графический режим и несколько графических экранов (с разным разрешением).

И там и там основным является текстовый режим - все сделано для него, а графические режимы - более позднее добавление, не вполне рационально организованные и не столь удобные, как хотелось бы.

Попробуйте-ка включив IBM или "Ямаху" (MSX) и не изменяя режимы и не загружая никакие программы, нарисовать хотя бы окружность. Не тут-то было. Ничего не выйдет.

Кстати, программистам и фирмам, выпускающим программное обеспечение, не очень удобно такое разделение режимов.

Итак, каким должен быть экран у компьютера, изначально предполагавшегося для работы в текстовом режиме? - Конечно, белым по черному.

А для графики? Наверное наоборот, ведь не очень привычно рисовать на черном экране.

Весь успех Синклера на том и был основан, что он дал людям компьютер с достаточно удобным и экономичным экраном, работающий одновременно и в текстовом и в графическом режиме, да еще и с палитрой цветов. При этом на все уходит менее 7 килобайт. Если все мы сейчас и наслаждаемся этой машиной, то скорее всего именно поэтому. Где сейчас стандарт MSX, который в зарубежной прессе в 1986 году называли "бурно развивающимся", а в 1987 году - "загнивающим". Если бы не эта "хитрость" с экраном, был бы сейчас "Спектрум" там же, где и сотни его конкурентов 1982-84 годов.

Итак, совмещение алфавитно-символьного экрана и многоцветного графического экрана с очень малым расходом памяти - вот то гениальное решение, которое и привело эту машину в Ваш дом, и вот почему у нее белый экран.

Ну, а что касается наиболее эргономичного сочетания цветов? Вопрос мнений. Авторитеты пишут, что желтым по синему. Может быть еще рассмотрен вариант белым по синему. Это если дело касается "Спектрума".

К вопросу о совместимости.

Рассматривая проблемы совместимости отечественных модификаций "Спектрума", мы уже говорили о том, что, как установлено, определенное влияние имеет формирование сигнала INT в Вашем компьютере. Сегодня наш корреспондент из г. Йошкар-Ола Сергей Викторович Полубарьев делится некоторыми рекомендациями по обеспечению параметров этого сигнала.

Вот что он пишет:

В результате анализа работы фирменного "ZX Spectrum+128" установлено следующее:

1. Длительность сигнала /INT (вывод 16 Z80A) равна 8 мксек и не зависит от текущего состояния процессора в том числе и сигналов /M1 и /MREQ.

2. Некоторое несовпадение линии на бордюре с линиями в экранном поле может наблюдаться и на этом компьютере (обнаружено на игре "Monte Carlo Casino"). При этом бордюрные линии смещаются на 1 пиксел вниз относительно экранных.

В наиболее распространенных версиях самодельных "Синклеров" нормализацией сигнала INT занимается RC-цепочка, дифференцирующая длинный кадровый синхроимпульс. Ниже дается описание этого узла для версий "Москва" (с отдельным полем памяти), Ленинград-1 и "Пентагон 48K" (дисковый вариант).

В рассмотренной нами версии "Москва" места для RC-цепи не предусмотрено. Если на Вашей компьютере она отсутствует, придется собрать ее навесным монтажом на свободных элементах DD54 (K555ЛН1, на плате располагается между рядами ОЗУ 565РУ6 и РУ5). Схема приведена на рис.1. Фрагмент, к которому относятся доработки, выделен пунктиром.

При указанных номиналах деталей обеспечивается длительность сигнала около 9 мксек, что достаточно для нормальной работы компьютера. В других перечисленных моделях RC-цепочка уже есть и достаточно уточнить номиналы ее деталей.

Необходимо отметить также, что встречаются варианты RC-цепочки, выполненные на резисторе с большим номиналом (30...50 кОм) и с очень малой емкостью (30...300 пФ). Такая схема в принципе работоспособна, но может стать причиной частых сбоев из-за высокой чувствительности к импульсным помехам (в практике автора такое бывало). Более надежно применять низкоомный резистор (0.5...3 кОм).

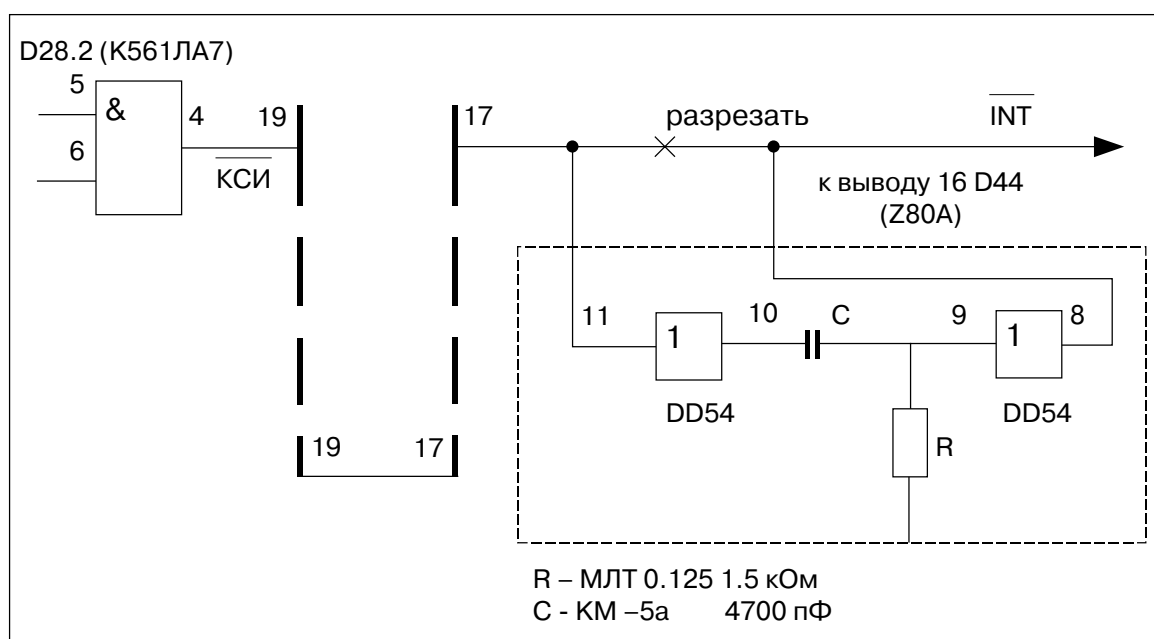


Рис.1

Если у Вас нет хорошего широкополосного осциллографа, то оценить необходимую емкость конденсатора можно прикидочным путем. Это делается так:

1) Постепенно снижая емкость конденсатора, определите порог, за которым прекращается опрос клавиатуры. Включенный компьютер выводит на экран "(C) 1982...", но не реагирует на нажатие клавиш.

2) Увеличьте полученное пороговое значение примерно в 1,5 ... 2 раза и впаяйте в схему конденсатор необходимой емкости.

Обычно этот метод неплохо срабатывает, а получаемая при этом длительность сигнала достаточно оптимальна для Вашего экземпляра компьютера.

В заключение для менее опытных пользователей небольшая справка по местонахождению дифференцирующей цепочки в других упомянутых моделях компьютера.

1) "Пентагон 48K" (дисковый вариант) - элементы С5, R13, R15, исходный сигнал

снимается с вывода 8 микросхемы D5.2 (K555ЛИЗ).

2) "Ленинград-1" - элементы C3, R2, далее сигнал инвертируется на D1.6 и поступает на вывод 16 Z80A. Исходный сигнал снимается с инверсного выхода триггера D8.2 (K555ТМ2).

Если обозначения элементов на Вашей схеме не совпадают с указанными, то их легко найти, проследив, откуда приходит сигнал на вход /INT процессора Z80A.

Сведениями по другим моделям автор пока не располагает.

Описанная методика проверена при наладке нескольких компьютеров каждой из версий. Компьютер версии "Москва", отлаженный, как описано выше, работает в течение полугода нормально. Без проблем запускается ARKANOID и другие игры. Единственная игра, не работающая на этой машине - "MOVIE" (причины пока не выяснены, но предполагается, что дело в отсутствии "подхвата" атрибутов экрана, т.к. резисторы 470 Ом на поле памяти 16K не установлены). - см. ZX-РЕВЮ N 3, стр.52 (FORUM).

Проблемы ELITE.

Нашему читателю из г.Челябинска Кислову Л.Н. кажется удалось в открытом космосе столкнуться с космической платформой. Причем произошло это не где-нибудь в глубинах отдаленных галактик, а у самой первой планеты - LAVE. Вот выдержки из его боржурнала.

Однажды, решив начать свою карьеру с нуля, я, имея рейтинг DANGEROUS, перезагрузил программу и начал игру с HARMLESS. При перелете с планеты LAVE на планету REORTE я столкнулся с летающим объектом необычной формы (бесформенный многогранник). Объект появился на переднем экране и шел мне навстречу. Приняв его за астероид, я начал обстрел издалека, надеясь на небольшую премию.

Каково же было мое удивление, когда "астероид" встретил меня ответным огнем. Несмотря на то, что боевой опыт у меня есть, поразить его пульсирующим лазером не было никакой возможности. Объект имел прекрасную защиту и, несмотря на многократные, длительные и точные попадания, не уничтожался. Во-вторых, он имел огромную скорость и постоянно маневрировал.

Самое интересное, когда от объекта начали "отслаиваться" корабли по 5-6 шт. за раз. (Наш читатель приводит их рисунок - это "Сайдуиндеры". - "ИНФОРКОМ").

Не обращая на них внимание, я продолжал преследовать объект. Примерно через минуту, поняв, что прекращать преследование я не собираюсь, объект выпустил новую партию кораблей - еще около 5 штук.

Общими усилиями они довели мою энергию почти до нуля, и я решил воспользоваться ракетами (обычно я ими не пользуюсь - считаю дорогими и малоэффективными). Первая ракета пропала неизвестно куда. Остальные тоже. Вылетев, они просто исчезали, не взрываясь. Последняя же ракета и не стартовала - появилась надпись о ее повреждении Missile Crashed.

Вскоре меня добились, и я так и не знаю, что же это было.

Версия программы характерна тем, что в стартовой экране выдается сообщение PHESS SPACE COMMANDER, а не LOAD NEW COMMANDER (Y/N).

* * *

Дорогие читатели. Вас, конечно не удивит, если "ИНФОРКОМ" признается, что получает сотни писем ежемесячно с описанием межзвездных баталий. Мы могли бы открыть серию романов, но должны честно сказать, что такого явного и яркого свидетельства, наличия космических платформ, как это, к нам еще не поступало.

* * *

Интересный эффект в этой программе обнаружил другой наш читатель, тоже из города Челябинска - Прыкин А. В (стаж полетов более года).

Он умеет перелетать со станции на станцию без затрат топлива и без боевых действий, воспользовавшись "жучком", имеющимся в программе, причем, как ему кажется, это не ошибка программы, а хитрая уловка фирмы.

Вот как это делается.

Перед стартом со станции выберите любую планету в данной галактике. Теперь стартуйте. Выйдя в космос, развернитесь на 180 градусов и вновь идите на стыковку со станцией. Когда сближение произойдет, перед самым входом на станцию, нажмите "H". Если выбранная Вами планета находилась в пределах круга, то Вы окажетесь на ней. Если же Вы выбирали планету вне пределов круга, ограничивающего гиперпереход, то нажимать надо "GH" - результат - тот же.

У "ИНФОРКОМа" сразу возникает вопрос: "А нельзя ли аналогичным способом перелетать и между галактиками?" Чем не предмет для исследования?

* * *

Интересное направление исследований начали вести сразу несколько наших корреспондентов. Они повели атаку на тот блок состояния, который выгружается на ленту при отгрузке программы.

Вот что пишет пилот Иванищенко (г. Орск).

Этот блок имеет всегда длину, равную 102 байтам. Методом проб удалось узнать назначение некоторых байтов в этом блоке. Прочие - пока не установлены, исследование продолжается.

В таблице приведены:

I - номер байта,

II - первоначальная установка,

III - назначение.

I	II	III
1	-	Имя командира
...	-	(10 символов в
10	-	кодах ASCII)
11	0	?
...	0	?
20	0	?
21	232	Сумма денег
22	3	в
23	0	кредах

Далее следует информация о количестве загруженных товаров

24	0	Food
25	0	Textiles
26	0	Radioactives
27	0	Slaves
28	0	Liquour/wines
29	0	Luxurious
30	0	Narcotics
31	0	Computers
32	0	Machinery
33	0	Alloys
34	0	Firearms
35	0	Furs
36	0	Minerals
37	0	Gold
38	0	Platinum
39	0	Gem-stones
40	0	Alien Items

Далее следует информация о снаряжении и бортовом вооружении

47	70	Топливо
48	3	Ракеты
49	0	Система E.C.M.
50	0	Extra Pulse Laser
51	0	Beam Laser
52	0	Fuel Scoops
53	0	Escape Pod
54	0	Energy Bomb
55	0	Energy Unit
56	0	Docking Comp.
57	0	Galactic Hyperdrive
58	0	Military Laser
59	0	Mining Laser

60 – 102 не установлены

Дополнительную информацию дает А.И. Ивлев из Симферополя. Он исследовал байты 42,13,44,45. Там находится информации о вооружении четырех боевых надстроек корабля (носовая, кормовая и две бортовых). При этом:

- 0 - отсутствие оружия;
- 1 - Pulse Laser;
- 2 - Beam Laser;
- 3 - Military Laser;
- 4 - Mining Laser.

Если заслать туда число, большее 4-х, можно получить интереснейшие эффекты. Например такую чепуху, как "Front Energy Bomb", причем со свойствами и боевого и технологического лазера одновременно. Очень удобно - стреляет как боевой, а обломки от астероидов летят, как от технологического.

* * *

И, наконец, вопрос к читателям, который задает наш давний партнер из г.Батуми - Гогуа А.В.: "Нет ли у кого версии программы ELITE, работающей в дисковой операционной системе". (Очевидно "самоделки", сделанные с помощью Magic-Button здесь не имеются в виду - "ИНФОРКОМ").

Советы и секреты

Просматривая код хорошо всем известной программы фирмы MIKROGEN "Three Weeks in Paradise" наш читатель Сукач К.М. (г.Херсон) нашел послание, предназначенное любителям взламывать чужие программы. Содержание его интересно, поэтому любопытным предлагается небольшая программа для его прочтения.

Так как основной игровой блок не имеет "хэдера", то сначала надо его сделать командой

```
SAVE "PARADISE!" CODE 26490, 38582
```

Затем надо считать этот заголовок и после него загрузить игровой блок.

После загрузки надо набрать небольшую программу-просмотрщик.

```
5 CLS
10 FOR n=63923 TO 65072
20 PRINT CHR$( PEEK N);
30 NEXT n: PAUSE 0: GO TO 5
```

И запустить ее <RUN>.

На экране появится текст послания.

Фильков Андрей Петрович (Москва) и другие сообщают о том, как в программе EQUINOX, экспертную проработку которой мы недавно давали, обессмертить своего робота. Суть такова:

На первом уровне нужно найти карту PETE, вернуться с ней к месту старта и находясь в левом верхнем углу, нажать одновременно клавиши R,N,C.

А в программе EXOLON можно получить бессмертие, если в режиме переопределения управляющих клавиш (DEFINE KEYS) ввести буквы Z,O,R,B,A.

* * *

Юрий Евгеньевич Горяинов из г.Славутич Киевской обл. сообщает о том, что в программе EQUINOX оказывается есть еще и встроенный МОНИТОР. Мы, правда, не знаем построила ли его фирма или просто в спешке забыла убрать (и такое бывает) или его построил кто-то из тех бойцов невидимого фронта, снимавших с программы фирменную защиту, но он там есть. Впрочем, откуда берутся в программах вмонтированные мониторы Вы можете узнать, если прочтете в данном номере статью Стива Тернера "Профессиональный подход".

Чтобы его вызвать, надо убрать автостарт программы и после загрузки дать команду RUN. МОНИТОР отображает на экране содержимое всех регистров процессора и ячеек памяти, позволяет записывать в них данные как в 16-ричном виде, так и в десятиричном и в символьной виде. Вот его команды:

- 1 - переключение DEC и HEX систем счисления;
- 2 - переключатель "символьное"/"числовое" представление;
- 3 - запись числа в регистр (через ENTER);
- 4 - запись числа по адресу (первому на странице);
- 5 - листание вниз на 64 адреса;
- 6 - листание вниз на 1 адрес;
- 7 - листание вверх на 1 адрес;
- 8 - листание вверх на 64 адреса;
- 9 - выбор регистра процессора (вперед);
- 0 - выбор регистра процессора (назад);
- SPACE - занесение начального адреса страницы.

Непробиваемая защита

"Очень многие сейчас занимаются "хаккерством", упражняясь во взломе и изменении программ" - пишет наш читатель из г.Таганрога Василенко В.И. - И дело это не столь сложное, как могло бы показаться. Имея "библию" Яна Логана и Ф. О'Хары "The Complete Spectrum ROM Disassembly" это сделать можно за полчаса, но увы не со всякой программой. Как подойти к защитам, в которых циклически изменяется информация, например как показано ниже?"

"ИНФОРКОМ" напоминает о том, что Вы можете заказать у нас качественно сделанную копию Мельбурнского издания книги Логана и О'Хары. Должны также добавить, что книга конечно хороша, но для неподготовленного читателя может быть сложноватой и наш комментарий к ней, который мы ведем уже год в виде сериала "Секреты ПЗУ" будет более чем полезен.

Пример такой защиты с циклической перекодировкой блока:

LD HL,nn	(адрес начала блока)
LD BC,nn	(длина блока)
LD A,n	(ввод ключа-декодера)
XOR (HL)	(декодирование)
LD (HL),A	(замена закодированного значения на раскодированное)
INC HL	(переход к очередному байту)
DEC BC	(уменьшение счетчика)
LD A,C	(проверка счетчика на ноль)
OR C	
JR NZ,n	(повтор декодирования)
DEFB	
DEFB	
закодированный блок	

DEFB

В данном примере байт за байтом декодируется блок, размещенный сразу за декодирующей процедурой. Декодирующей функцией служит XOR. После окончания процесса декодирования исполнение переходит к первому байту из декодируемого блока. Вроде бы все просто, но вдруг! О ужас! Оказывается, что раскодированный блок содержит в самом начале примерно такую же процедуру, а дальше за ней опять "абракадабра". Раскрыв и запустив декодирующую процедуру, мы декодируем и ее, но дальше опять все то же и так до 100 - 200 раз. Что же делать, нельзя ли как-то это преодолеть?

Мы должны откровенно признаться, что бессильны чем-то помочь нашему читателю и привели здесь логику работы защиты во-первых для тех, кто не знает этого приема, а во-вторых в надежде на то, что отзовется кто-нибудь, кто решил такую проблему, хотя мы признаться мало в это верим. Дело вот в чем.

Несколько лет назад на Западе было объявлено о создании интегрального нераскалываемого загрузчика ALKATRAZ LOADER. Он работал именно по этому принципу. Была объявлена громадная премия тому, кто сумеет его "расколоть" и, насколько нам известно, эту премию никто не востребовал. Конечно "расколка" аппаратными средствами не признавалась.

Кодирующая программа продавалась фирме, желавшей защитить свою продукцию. После работы кодировщика к программе приписывался загрузчик и теперь никто, в том числе и фирма, защищавшая свою продукцию, не могла снять защиту.

Самое суровое, что этой защитой защищалась даже не сама программа, а другие защищающие процедуры. Возьмем, к примеру, шедевр адвентюрных игр - программу KAYLETH, написанную по повести Айзека Азимова. Там "Алькатрасом" защищался блок, обеспечивающий ускоренную нестандартную загрузку и, к тому же, между блоками была встроена пауза с замеряемой продолжительностью и с замеряемым уровнем шумов. Копировщики эту программу естественно не берут, а прямое переписывание с магнитофона на магнитофон удастся только на очень дорогих высококачественных аппаратах, да и то работоспособной остается максимум вторая копия.

Вопрос поднятая нашим читателем достаточно интересен. Мы бы с радостью опубликовали исследования наших авторов, работающих в этом направлении. Наш собственный опыт увы исчерпывается одной-двумя безуспешными попытками. Так, например, раскрывая программу Pinball Wizard, мы прошли тяжелым трудом 20-25 этапов защиты, но наткнулись на то, что сравнительно простые декодирующие функции типа XOR (HL) или DEC A сменились гораздо более крутыми:

```
LD B, (IY+n)
LD A, (IX+n)
XOR B
```

Если Вы с самого начала не отслеживали содержимое регистров IX и IY, то после труднейшего дня кропотливой работы очень приятно столкнуться с такой функцией.

По всей видимости для снятия защит такого типа необходимо специальное программно-инструментальное средство, возможно что-то типа резидентного монитора.

Для тех читателей, кто готов осудить нас за подстрекательство к "хаккерству" скажем, что мы со всем уважением относимся к авторским правам, (сами ведь тоже пишем программы для IBM-совместимых машин) но считаем допустимым и даже полезным для самообразования "влезть" в код чужой программы, если при этом:

- программу не "портят" в угоду личным амбициям;
- не меняют данных об авторстве программы;
- внесенные изменения оставляют для себя и не отравляют жизнь будущим пользователям, распространяя их.

Одним словом, мы относимся к этому как к спорту, который как любой вид спорта может быть чистым, а может быть и грязным. И зависит это не от спорта, а от спортсмена.

По всей видимости в условиях отсутствия нормального авторского законодательства и нормального легального распространения программного обеспечения основным критерием, которым стоит руководствоваться, должно быть **УВАЖИТЕЛЬНОЕ И КОРРЕКТНОЕ ОТНОШЕНИЕ К АВТОРУ ПРОГРАММЫ.**

PYJAMARAMA

Фирма: MIKRO-GEN

Год выпуска: 1984



Многие любят игры аркадно-адвентюрного жанра. В них можно играть неделями. К сожалению, сложность многих программ не позволяет начинающим получить все то удовольствие от игры, на которое они вправе рассчитывать.

В последнее время наши эксперты обратили пристальное внимание на игры этого жанра. Вместе с тем, нам бы хотелось дать им какое-то направление в работе. Дело в том, что подробно расписанный порядок действий может отбить интерес к игре и здесь нужен взвешенный подход.

Сегодня в качестве одного из возможных образцов инструкции к аркадно-адвентюрной программе мы предлагаем описание программы PYJAMARAMA.

Должны сразу оговориться, что это описание достаточно хорошо распространено по стране. Мы его сделали в те далекие годы, когда никакого "ИНФОРКОМА" еще не было и в проекте. Вполне может быть, что оно вам знакомо и мы просим Вас не воспринимать его как несвежий материал. Его цель - дать определенное направление всем экспертам, которые работают по нашему заданию в этом жанре.

Цель игры: Вам надо разбудить Уолли, главного героя многих игр фирмы MIKRO-GEN. Для того, чтобы его разбудить, надо завести будильник.

Данная инструкция построена таким образом, чтобы читатель не потерял интерес к игре. Для этого информация, необходимая для полного завершения игры сообщается постепенно, небольшими разделами. Ознакомившись с очередным разделом, попробуйте свои силы и только если у Вас все равно ничего не получается, читайте очередной раздел.

1.



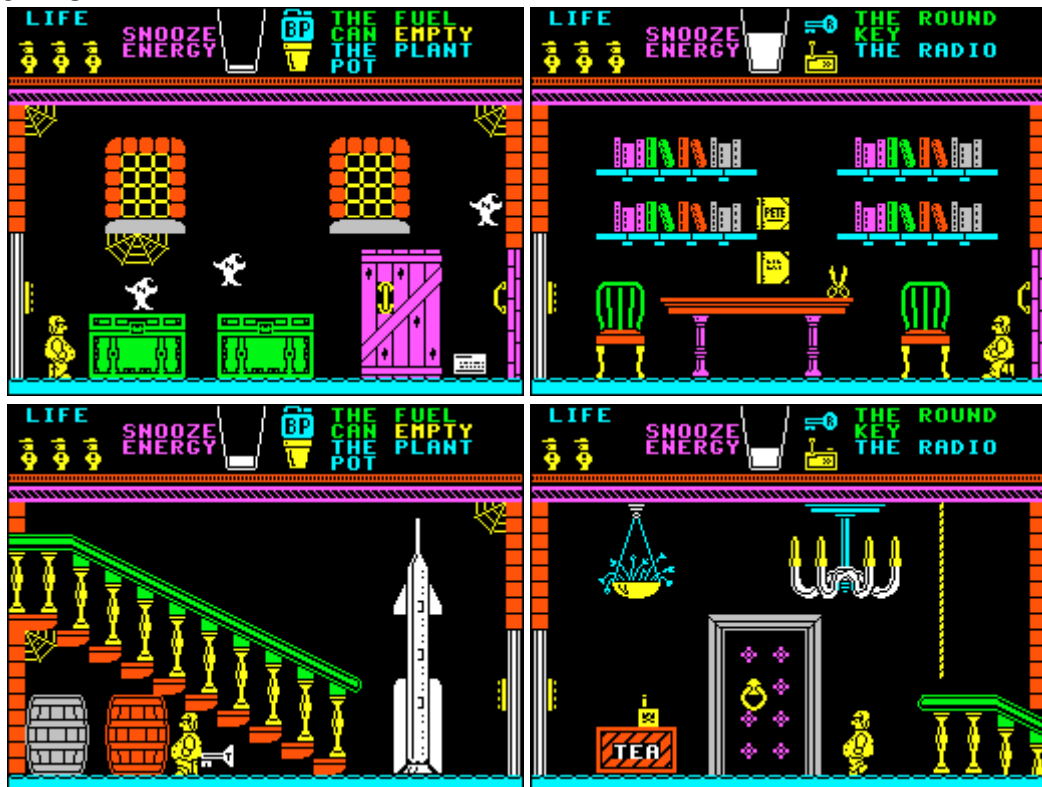
Для того, чтобы найти ключ от будильника, надо слетать на Луну. Ключ находится за решеткой с магнитным замком, поэтому надо иметь при себе магнит. Проход к ключу охраняют инопланетяне, которых можно пройти только, если доставить на Луну заряженный лазерный пистолет. Ракета запускается только если у Вас есть при себе полная канистра топлива. Так как более двух предметов иметь при себе одновременно нельзя, то надо в первый рейс на Луну доставить туда заряженный лазерный пистолет, а только во втором рейсе с магнитом идти за ключом...

2.

Итак, три основные проблемы - заправить ракету топливом, найти и зарядить лазерный пистолет, взять магнит (он на кухне под столом, закрыт на ключ).



Кроме обычного передвижения по дому, можно пользоваться лифтом. Вход в него из библиотеки. По дороге, проходя через комнату, в которой лежит пляжный мяч, надо подпрыгнуть и переключить кнопку вызова лифта так, чтобы стрелка смотрела вниз. В лифте есть четыре бокса, прыжком в которые лифт отправляется на соответствующий этаж (лампочка наверху у включенного бокса загорается). Первый этаж - подвал. Второй этаж - исходный, с него начинается игра. На третьем этаже находится ракета. На четвертом этаже спальня, где спит Уолли, ванная комната, комната видеоигр и прочие комнаты. Кстати, если Вы войдете в комнату видеоигр и пройдете все уровни игры, Вы получите дополнительную жизнь.



Как заправить ракету топливом? Для этого надо в комнате с щелкающими капканами взять канистру. Затем на первом этаже найти топливную помпу. Если подойти к ней с пустой канистрой, она наполнится.

Где взять лазерный пистолет? Лазерный пистолет лежит в бильярдной. Туда можно войти только если у Вас есть специальный квадратный ключ. Дверь в бильярдную на третьем этаже. (С четвертого этажа спуститься по лестнице вниз), но без ключа она не открывается. Проход к ключу следующий: на первый этаж, затем вторая комната с привидениями направо, затем по ящикам выпрыгнуть в зарешеченное окно, далее направо и вниз, там горит огонь. Чтобы его пройти, надо иметь огнетушитель. Далее вниз и налево.

Как взять магнит? Чтобы взять магнит, надо иметь специальный ключ от ящика. Ключ висит под потолком в той комнате, где есть воздушный шарик. Первое - шарик Вас

поднимет наверх, если Вы перережете веревочку ножницами, второе - поднявшись наверх, Вы сможете прыжками подойти к ключу, но до него не допрыгнете. Для этого надо под ключ подставить ящик. Это делается переключением кнопки "HELP" в комнате на четвертом этаже рядом с ванной так, чтобы стрелка смотрела вниз. Помните, что эта клавиша может самопроизвольно выключаться, так что может быть Вам придется сбегать наверх еще раз.

3.

Как отключить капканы, чтобы взять канистру? для этого надо иметь с собой ведро воды. Внеся его в комнату, можно там его и оставить вместо канистры, тогда капканы перестанут работать.

Ведро берется по дороге в библиотеку, прыжком со стула. Заполнить его водой можно в ванной комнате на четвертом этаже, пройдя под краном. Надпись "EMPTY" (пустое) сменится надписью "FULL" (полное). Помните, что если Вы где-либо выпустите ведро из рук (поменяете на какой-либо предмет), вода прольется и оно опять станет пустым.

Где взять огнетушитель? Огнетушитель находится в лифте. Над ним надпись "SMASH GLASS" (разбей стекло). Разбить стекло нужно молотком в прыжке.

Как взять ножницы? Ножницы находятся на столе в библиотеке, но справа со стула до них не допрыгнуть, а слева не пройти - мешают движущиеся книги. Чтобы взять ножницы, надо иметь с собой мотоциклетный шлем для защиты (он на третьей этаже лежит на сундуке), а также библиотечную книгу (она на четвертом этаже, рядом с кушеткой).



4.

Где взять молоток? Молоток лежит в туалете. Вход в туалет рядом со входом в биллиардную, но эта дверь закрыта. Вход платный, надо при себе иметь мелкую монету - пенс.

Как взять шлем? Шлем берется, если скатиться по перилам с четвертого этажа на третий, при этом надо иметь при себе ключи зажигания. Ключи зажигания находятся на пути к квадратному ключу (с первого этажа через зарешеченное окно), но берутся только в обмен на водительские права, они лежат в подвале.

Как взять библиотечную книгу? Для этого надо поменять ее на читательский билет, который находится на третьем этаже на стене в комнате по соседству со шлемом.

Где взять пенс? Надо сначала в столовой взять фунт стерлингов и идти с ним на четвертый этаж. Там надо найти разменный автомат с надписью "CHARGE" (размен). Прыжком поменять фунт на пенсы.

Как зарядить лазерный пистолет? Пистолет заряжается от батарейки, которая лежит в комнате на первом этаже. Проход в эту комнату из той комнаты, в которой стоит топливная помпа. Но дверь открывается только трехгранным ключом. Он на третьем этаже рядом с ракетой. (При взятии этого ключа желательно, чтобы капканы уже были обездвижены, т.к. путь в лифт может быть закрыт, а путь наверх, через комнату с капканами, будет непроходим при работающих капканах). Отправляясь за батарейкой, на этаж N1, перед входом в лифт убедитесь, что и пистолет и ключ с Вами.

Войдя в комнату с батарейкой, немедленно (!) прыгайте на ящик и на стол, но не проскочите мимо батарейки.

Как запрыгнуть на перила, чтобы взять шлем? Проход четвертого этажа представляет

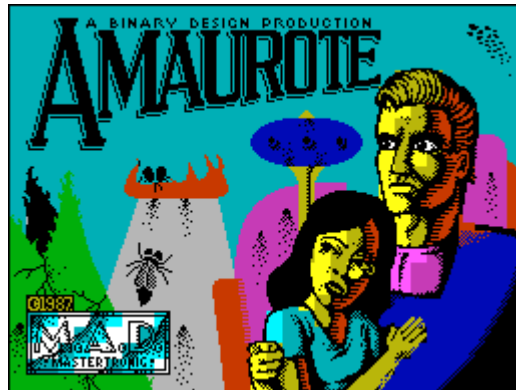
собой движущийся конвейер. Передвигаться по нему в противоположном направлении можно только прыжками. С движущегося конвейера запрыгнуть на перила можно, изъяв привод. Он лежит на сундуке в углу. Для этого, приехав на лифте на четвертый этаж и выйдя из двери, сразу прыгайте на ящик и берите привод.

В процессе игры Вы можете подкреплять силы Уолли пищей, которая иногда появляется в некоторые места (шоколад, джем, яблоко и пр.). Желательно пользоваться этим рационально. Чтобы не расходовать еду, надо просто через нее перепрыгнуть.

В комнатах есть и другие предметы, которые также могут быть взяты, хотя непосредственного участия в игре не принимают.

Слово экспертам

АМАУРОТЕ
М. А. Д. 1987 Г.



Эксперт Борисов П.М.
г. Норильск

Программа относится к аркадно-адвентюрным играм с элементами ACTION.

Ваша задача - очистить город от вредных инопланетных насекомых. Для этого Вам выдали в полное распоряжение боевую машину "Арахнус-4", оснащенную совершенным радарным оборудованием.

У Вас есть двусторонняя радиосвязь с базой и 30 инсектицидных бомб. Мы надеемся, что с их помощью Вы не разрушите город до основания. Вам также ассигнована сумма в 5 млн. долларов.

Население покинуло город, армия также бессильна, теперь вся надежда на Вас.

Город.

Город разделен на 25 районов, в каждом из которых насекомые организовали свои колонии. У Вас есть карта и Вы можете входить в эти районы в любом порядке. Каждый район представлен кружком и, перемещая "Арахнус-4", Вы производите выбор района, при этом на экране зажигается название. Когда Вам удастся уничтожить всех насекомых в районе, Вы вернетесь к карте.



Насекомые.

Существует три типа насекомых:

МАТКА. Это Ваша основная цель, но к ней очень нелегко пробиться, т.к. она находится под постоянной охраной. Она выполняет две функции. Во-первых, она отдает приказания трутням, которые работают на основе той информации, которую доставляют разведчики и, во-вторых, выводит новых насекомых.

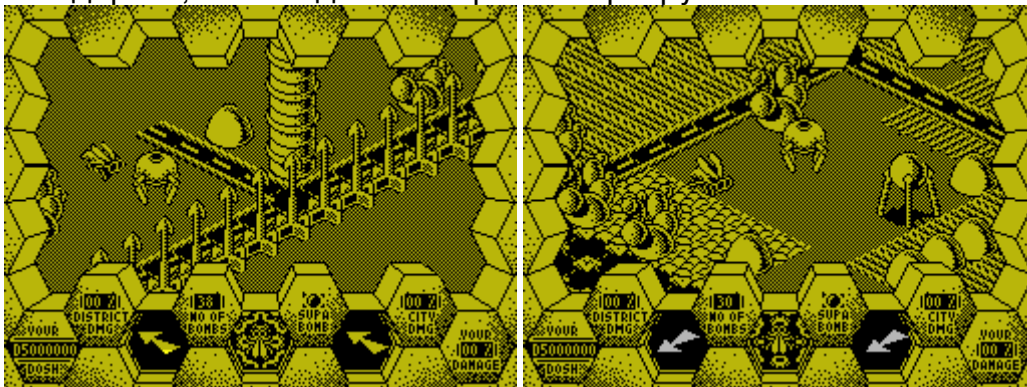
Каждый раз, когда Вы убиваете очередное насекомое, она выводит новое, но не сразу. Время, которое для этого необходимо, зависит от того, когда в последний раз ее кормили.

РАЗВЕДЧИКИ. Они летают по городу в поисках потенциальной пищи и непрошенных гостей (это касается и Вас). Если разведчики увидят Вас, они немедленно помчатся с донесением к матке и тогда у Вас будут большие осложнения.

ТРУТНИ. Их задача доставлять пищу и защищаться. Они довольно тупы, строго подчиняются данным им приказам, но не сдаются в бою и не отступают.

Бомбы.

Бомба запускается нажатием кнопки "огонь". Направление стрельбы - это направление Вашего последнего движения. Бомба летит до столкновения с насекомым или со зданием. Помните, что от Вас требуют очистить город от насекомых, а не разрушить его. Не поддаются уничтожению только периметр города и сама матка. Для ее уничтожения Вам потребуется супербомба. С собой ее у Вас нет, но Вы можете сделать заказ в любое время. Эти бомбы очень дороги, но обладают невероятной разрушительной силой.



После того, как бомба запущена, Вы не можете запустить вторую, пока первая не взорвется.

Число, которое показывает оставшееся количество бомб, в это время имеет красный цвет, даже если бомба находится за пределами экрана.

Радио

Задействование радио открывает перед Вами следующее меню:

1. Заказать бомбы.
2. Заказать супербомбу.
3. Ремонт.
4. Выход из города.

Бомбы доставляются в город по воздуху и Вы можете разыскать их с помощью своего радара. Если Ваш "Арахнус-4" претерпел большие повреждения, можете вызвать новый, но это стоит денег.

Радар.

Вы можете запрограммировать свой радар на отыскание ближайшего насекомого, матки или доставленных бомб. Для этого нажмите необходимую клавишу и следуйте за указателями.

Управление.

Вправо-вверх - Y...P
Вправо-вниз - H...ENTER
Влево-вверх - Q...T
Влево-вниз - A...G
"Огонь" - B...SPACE
Радио - CAPS SHIFT
Изменение цвета - V
Для радара:
Поиск насекомых - Z
Поиск матки - X
Поиск бомб - C

FREDDY HARDEST

DINAMIC 1987



Эксперт Смирнов В. Б.

г. С. -Петербург.

Далекая галактика, незнакомая звезда... Одна из планет излучает загадочные радиосигналы.

Космический путешественник Фредди Хардест получает специальное задание Всегалактического Центра Космических Исследований узнать причину появления этих сигналов. И вот он, вдалеке от родной звезды, подлетает на своем космическом челноке к чужой планете. Бортовой компьютер сигнализирует о выходе на круговую орбиту, а затем беспристрастная техника фиксирует приближение инопланетного космического корабля. Фредди обеспокоен: корабль не отвечает на запросы, чужак открыл огонь по челноку Фредди. Завязался космический бой, который, впрочем, длился недолго. Фредди сумел уничтожить врага, но и сам еле дотянул до враждебной, как оказалось, планеты. Тщательно перевязав раны, отважный путешественник отправился в путь не зная, что его ждет впереди.



С этого момента Вы вступаете в игру. Вы должны помочь Фредди избежать мести инопланетян и улететь с чужой планеты.

Незадолго до гибели корабля бортовой компьютер выдал информацию о климате и населении планеты, а также о расположении городов инопланетян. Проведя расчеты, Фредди определил, что в пяти милях от места падения находится крупное поселение с космической базой. Тут же возник план: добраться до города и, если представится возможность, угнать космический корабль. Планета населена рептилиями негуманоидного типа. Они имеют явное намерение уничтожить Фредди любой ценой. Ваша задача - помочь ему.

Оружие.

Итак, цель игры - добраться до города инопланетян. Выполнение задания осложняют снующие повсюду рептилии, которые, однако, не вооружены, в то время как у Фредди имеется лазерный пистолет новейшей конструкции. Для того, чтобы им воспользоваться, надо нажать клавишу "вниз" и, удерживая ее, клавишу "огонь". Пистолет обладает бесконечным, т. е. не уменьшающимся зарядом лазера.

Город.

Дойдя до лестницы, ведущей под землю, Вы должны высоко подпрыгнуть над ней и ... можете загружать FREDDY HARDEST II.

Полезный совет.

Если у Вас нет желания сражаться с летающими (но не прыгающими) рептилиями, Вы можете при их полете присесть. В этом случае они пролетают над Вами, не задевая.

Экран.

В верхней части экрана отображается игровая ситуация, в нижней расположено информационное табло. Рассмотрим его структуру.

Слева на табло изображен улыбающийся Фредди. Правее указано количество оставшихся кредитов (в начале их дается пять). Далее расположен символ пистолета. Ниже выводится количество очков.

Три квадрата, находящиеся в правой части информационного табло, не используются в первой части игры FREDDY HARDEST.

Очки.

За каждую уничтоженную рептилию Вы получаете 200 очков.

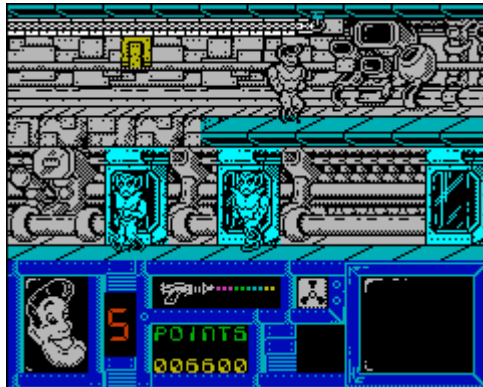
FREDDY HARDEST II



Эта игра является продолжением игры FREDDY HARDEST. После того как Фредди добрался до города инопланетян, ему надо улететь с чужой планеты. Корабль космического путешественника разбит и у Фредди есть один выход: угнать один из боевых кораблей инопланетян.

Игра.

Ангары космических кораблей находятся в нижнем ярусе города. Корабли различаются по цвету. Существует четыре вида кораблей: зеленый (green), красный (red), синий (blue) и белый (white). Для побега Вы можете воспользоваться любым из них.



Для того, чтобы стартовать с планеты, необходимо подготовить корабль к полету. Подготовка заключается в проведении трех операций:

1. Заправить корабль ядерным топливом.
2. Открыть ангар заправленного корабля.
3. Определить код доступа в корабль.

Для выполнения этих заданий Вам необходимо научиться пользоваться служебными

компьютерами инопланетян. Они расположены на разных этажах города. Для получения информации от компьютера, надо встать рядом с ним и нажать клавишу "вверх". При этом на информационном табло в среднем квадрате начнет мерцать изображение дискеты, а в большом появится сообщение. Если Вы уже получаете информацию от компьютера, но передумали это делать (например, если к Вам приближается инопланетянин), нажмите клавишу "вниз".

Надо заметить, что не все компьютеры исправны. Некоторые выводят сообщение "Out of order" - неисправен.

Теперь подробнее о выполнении заданий.

Заправка.

Для начала вам надо найти контейнер с топливом. Он выглядит, как мерцающий квадрат с нарисованным на нем знаком радиации (т.к. топливо ядерное). С контейнером идите в комнату, где на полу написана буква N (от Nuclear - ядерный). Таких комнат в городе две. Встаньте на участок пола с изображением буквы. При этом контейнер окажется стоящим на полу. После этого подойдите к стоящему здесь же компьютеру и включите его. Контейнер опустится вниз. Далее по автоматическим линиям он будет доставлен к кораблю и загружен в него. На табло появится сообщение о цвете заправленного корабля. Например: Green ship full - зеленый корабль заправлен.

Открытие ангара

Для того, чтобы открыть ангар корабля, который Вы только что заправили, надо найти компьютер, управляющий открыванием данного ангара. Тут Вам придется побегать в его поисках.

При открывании ангара Вам выдается сообщение вида: Red on to hyperspace - Ангар красного корабля открыт.

Код доступа

Определение кода доступа в корабль производится также с помощью компьютера, который вам надо найти. При этом выводится такое сообщение: White captain, Code AUDAX - капитану белого корабля: код AUDAX.

Взлет

Надо заметить, что не обязательно выполнять все задания в приведенной порядке.

Итак, Вы заправили корабль, открыли ангар и знаете код доступа. Теперь спускайтесь в самый низ, к ангарам. Отыскать нужный ангар Вам помогут две буквы, написанные у каждого корабля, это сокращение от названия цвета, например: GR - GREEN (зеленый). После того, как Фредди зайдет в ангар, бортовой компьютер проверит наличие на корабле топлива и открыт ли ангар для вылета. Если все в порядке, у Вас запросят код доступа в корабль. Если Вы ответите правильно, то появится сообщение "Все системы готовы. Приготовиться к старту." и Вы можете наблюдать, как Фредди вырывается на космический фарватер.

Полезные советы

1. Для ускорения передвижения по городу можно пользоваться "темными" комнатами. Снаружи они выглядят, как большие черные раковины. Зайдя в такую раковину и выйдя с другой стороны, Вы окажетесь в другой части города.

2. Кроме того, можно пользоваться лифтами. К сожалению, они перемещаются лишь в пределах двух этажей.

Инопланетяне

С ними можно бороться двумя способами. Первый - бить их руками и ногами, а второй - стрелять в них из лазерного пистолета. Несмотря на кажущееся преимущество пистолета, первый способ предпочтительнее, т. к. им можно уничтожать все виды инопланетян, в то время как вторым только один вид - больших рептилий. К тому же, заряд пистолета во время

стрельбы уменьшается.

Экран

В отличие от первой части игры, во Freddy Hardest II используются три квадрата, находящиеся в правой части информационного табло. В маленьком квадрате индицируется наличие у Фредди контейнера с топливом. В среднем квадрате в момент получения информации от компьютера мерцает изображение дискеты. И, наконец, в большой квадрат компьютер выводит сообщения.

Длина линии, расположенной рядом со стволом пистолета, указывает оставшийся заряд. Когда Вы не пользуетесь оружием, оно автоматически подзаряжается.

Управление

Клавиши управления Вы выбираете сами перед началом игры. Для захода в ангар, включения компьютера, захода в "темные" комнаты, запрыгивания на горизонтальные и вертикальные веревки, используйте клавишу "вверх".

Если Вы нажмете клавишу "огонь" во время движения, то Фредди сделает удар рукой, а если Вы просто стоите, то - ногой. Когда Фредди висит на горизонтальном канате, удар производите клавишей "вверх".

Для использования лифта встаньте на него и нажмите клавишу "вверх" или "вниз".

Очки

Подсчет очков такой же, как во Freddy Hardest I, т.е. 200 очков за рептилию.

SILENT SERVICE.

MICROPROSE 1989

Эксперт Морозов С. Ф. г. Хабаровск



Эта игра относится к имитаторам и представляет имитатор подводной лодки.

Вы можете либо просто потренироваться в уничтожении вражеских кораблей, либо вести настоящую подводную войну. Для этих целей Вам предоставлена подводная лодка по номером 104. Это подлодка типа "К" - крейсерская, она обладает довольно мощным вооружением:

- носовая пушка с 80 снарядами;
- 6 носовых торпедных аппаратов с 14 запасными торпедами;
- 4 кормовых торпедных аппарата с 10 запасными торпедами.

Мощный дизель позволяет развивать скорость до 20 узлов в час (надводную) и до 10 узлов (подводную). Двойные переборки позволяют опускаться до 500 м.

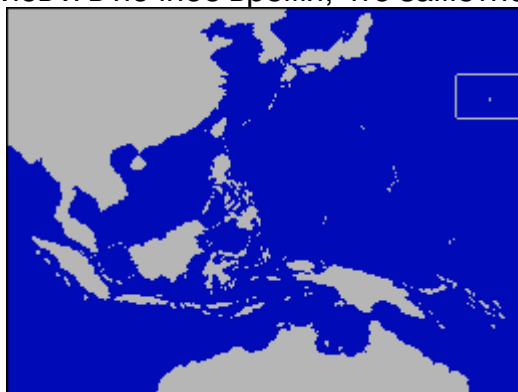
Вам предлагаются 3 варианта ведения боевых действий.

1. Практика

Вам следует начать с этой миссии. Перед Вами будет находиться неохранный вражеский караван. Расстрелять его из пушки или торпедами не составит никакого труда, но именно здесь и осваиваются навыки. Начиная с мичмана, Вы вполне сможете дослужиться и до адмирала (присваивается и такое звание), став грозой морей.

2. Разгром вражеского конвоя.

В этой миссии Вам уже будут противостоять вражеские "охотники", которые будут атаковать Вас глубинными бомбами. Вот, где Вы вспомните, что Ваш корабль - подводный. Военные действия будут вестись и в ночное время, что заметно осложнит их.



3. Для самых отважных есть третья миссия - военное патрулирование.

Здесь Ваше мастерство как капитана субмарины проявится в полной мере. Вам придется не только топить корабли, но еще и искать их. Поможет Вам чутье, но и, конечно, смекалка. В погоне за транспортом будут меняться день и ночь. Вас будут пытаться протаранить и подбить глубинными бомбами, повредить лодку огнем пушек. Вы же, топя корабли и зарабатывая тоннаж, будете двигаться к главной цели - уничтожить противника и стать адмиралом.

Все действия происходят в районе Центральной Америки и Вашими главными врагами будут японские корабли. Что же, мы знаем, что американцы сумели одержать победу в этой войне, того же желаем и Вам. А чтобы победить, надо хорошо разобраться с игрой, что мы сейчас и сделаем.

Настройка программы

В этой игре нельзя использовать джойстик или что-либо другое, вся игра идет под управлением от клавиатуры. После загрузки перед Вами основное меню, где перечислены данные миссии. Нажав клавишу "1", "2" или "3", Вы выберете ту или иную миссию. Начинать, конечно, следует с первой. Итак, по порядку.

Если Вы выберете торпедно-пушечную практику (клавиша "1"), то окажетесь в дополнительном меню. Это меню, по сути дела - выбор уровня. Вначале Вы можете задать свой рейтинг от мичмана до капитана (клавиши "1" - "4"), затем можете еще более усложнить (или облегчить) жизнь, выбирая: имеются ли ограничения видимости, может ли конвой идти зигзагами, "опытный" ли охотник и (или) конвой и т.д. Управление происходит клавишами "Y" - вверх, "N" - вниз, "M" - выбор. Но для начала советуем сразу нажать "SPACE" и перейти к игре. Потом, по мере накопления опыта, Вы всегда сможете выбрать нужный Вам уровень.

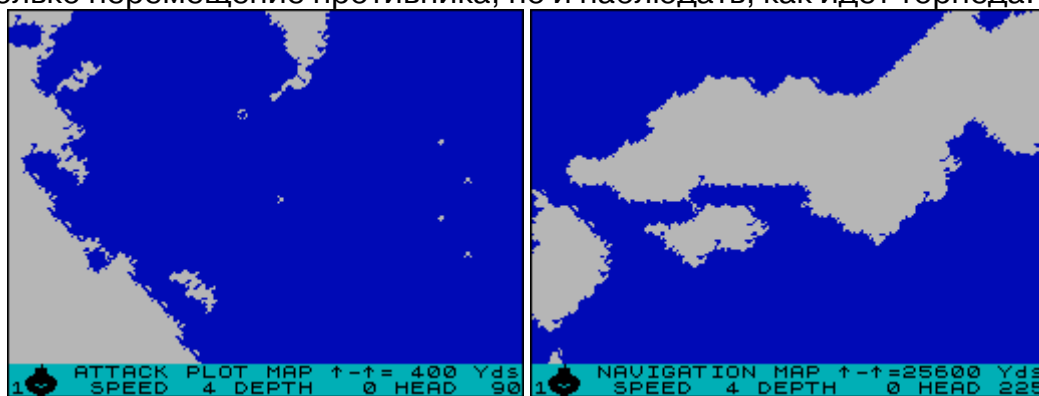
Но если Вы в основном меню выберете конвойную акцию (клавиша "2"), то перед Вами станет задача выбора одного из шести мест прохождения конвоя и времени суток. За этим последует дополнительное меню, действия в котором аналогичны. И, наконец, решившись на военное патрулирование, предстоит выбирать начальное положение подлодки (одно из пяти) и год войны. После этого Вы должны будете выбрать тип вражеского противолодочного корабля (клавиши "1" - "4"). Если Вы неудачно выберете тип японского "охотника" (то есть уровень не будет соответствовать типу корабля), то появится надпись "Потренируйтесь". Но какая бы надпись ни появилась, все равно Вы попадете в дополнительное меню, где следует действовать аналогично первым двум случаям.



Теперь нам надо освоить технику управления подводкой. Самое сложное - освоить клавиши, сама же игра происходит в нескольких экранах, с ними разобраться легко.

1. Экраны карт

Сразу после выбора игры перед Вами на секунду появится карта побережья Центральной Америки, с ограничением того района, где Вам предстоит топить корабли. (Это относится к торпедной практике и конвойной миссии, режим же боевого патрулирования отличается, его мы рассмотрим ниже). После этого Вы окажетесь в экране-1 - навигационной карте, она имеет очень крупный масштаб. Поэтому лучше сразу переключиться на экран-2 - (карту патрулирования), у нее масштаб мельче и все цели довольно хорошо просматриваются. Переключение экранов производится клавишей "Z". Вы можете войти также в экране-3 (карта торпедной атаки). В этом экране Вы сможете наблюдать не только перемещение противника, но и наблюдать, как идет торпеда.



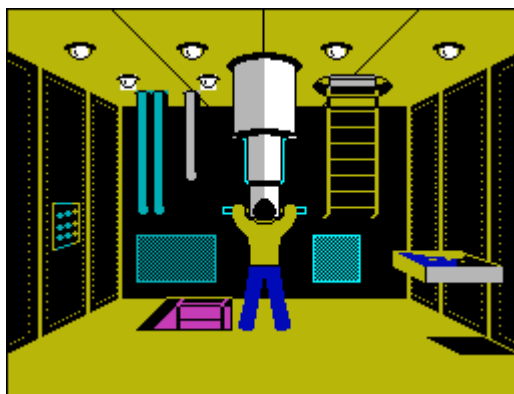
Эти экраны незаменимы при подводном положении субмарины, когда нет возможности наблюдать за вражескими кораблями через перископ и маневрировать. Обратное переключение карт осуществляется клавишей "X". Под картами - две информационные строчки, которые информируют Вас о скорости судна, глубине погружения, об угле движения подлодки, а также о том, какая карта на экране.

2. Боевая рубка

Определив местонахождение японских судов, можно переключиться на командный пост. Это делается клавишами "M" или "SPACE". Перед Вами будет внутреннее помещение корабля и капитан, находящийся у перископа. В боевой рубке Вы можете:

1. Посмотреть в перископ (если он поднят и глубина не превышает 44 метра¹). Осуществляется нажатием клавиши "H".
2. Подняться на палубу (если субмарина в надводном положении). Клавиши "Y"+"M".
3. Обратиться к картам. Клавиши "O"+"M".
4. Осмотреть приборы, показывающие состояния всей лодки. Клавиши "9"+"M".
5. Осмотреть повреждения лодки, клавиши "H"+"M".
6. Вывести список потопленных судов и их тоннаж, а также количество оставшихся торпед. Клавиши "H"+"O"+"M".
7. Возврат в меню (кроме режима боевого патрулирования). Клавиши "H"+"9"+"M".

¹Здесь и далее глубины указаны в футах. (Прим. NUK)



Это все, что касается действий в боевой рубке. Теперь рассмотрим эти действия по порядку.

3. Перископ

Выбрав этот режим в боевой рубке, Вы увидите то, что находится в поле зрения перископа, а также данные от системы наведения торпед. Они загораются лишь в случае захвата цели и содержат:

1. Расстояние до цели.
2. Скорость цели.
3. Угол упреждения торпеды.
4. Угол подлодки к цели.
5. Курс цели.

И опять же две последние строчки занимает информация о скорости, глубине погружения, курсе, а также дополнительная информация, которая будет рассмотрена ниже. Вначале Вам следует найти цели, что осуществляется поворотом субмарины. Он может быть выполнен при помощи руля или двигателя (для этого лодка должна иметь ход). Поворот влево - клавиша "С": один раз - поворот рулем, два раза - двигателем. Так же происходит поворот вправо - клавиша "V". В нижней строчке появляется соответствующая надпись.



Остановка осуществляется клавишей "ENTER". Можно поднять или опустить перископ - клавиша "Р". Появляется соответствующее сообщение. Скорость лодки задается клавишами "2"-"5".

Погружение осуществляется клавишей "D", при этом раздается предупредительный звуковой сигнал и выдается сообщение. Если в этот момент перископ был поднят, то Вы будете продолжать наблюдение до глубины 44 м. Затем, как если бы перископ был бы опущен, Вы переходите в боевую рубку к карте.

Для остановки погружения Вам надо, нажав "9"+"M", перейти в экран приборов, и нажатием "ENTER", остановить погружение.

Дальнейшие действия могут быть например такими:

- поднять перископ (если он был опущен);
- осуществить всплытие (клавиша "S") до разрешенной глубины, после чего вновь можно производить наблюдение.

Скорость при подводном положении, конечно, меньше. Полная остановка с выключением механизмов осуществляется клавишей "1".

Можно дать задний ход. Для этого, дав одну из четырех скоростей, надо нажать

клавишу "R". Теперь, поймав цель в перекрестие прицела, можно получить информацию от системы целеуказания.

Можно также идентифицировать тип судна, нажав "I". Можно поворачивать лишь перископ клавишами "9", "0", торпеда пойдет по заданному таким образом курсу.

Дальность действия торпеды - 3000 ярдов, поэтому, когда до цели осталось меньше 3000 ярдов и она поймана в прицел, Вы можете смело атаковать торпедой (клавиша "T").

В надводном положении можно использовать и пушку.

Прицеливание осуществляется клавишами "J" (увеличение угла прицеливания) и "K" (уменьшение), стрельба - клавиша "G". Если Вы догоняете цель, но она приближается слишком медленно, обладая хорошим ходом, то можно использовать 2-ю, 3-ю или 4-ю временную шкалу (клавиша "F"). Переход на нормальную шкалу происходит при выхождении цели из прицела или по нажатию клавиши "N".

Это практически все клавиши, используемые а данном экране. Есть правда еще клавиша "A". Она начинает работать в случае, если в дополнительном меню Вы выбрали режим ручной установки угла упреждения торпеды. Тогда при нажатии на "A", появляется угол упреждения, который можно увеличивать и, уменьшать клавишами "9", "0". Устанавливается он так:

1) От полученного значения отнять градусную меру курса Вашей субмарины.

Полученное значение и следует вводить. Выход осуществляется нажатием на "M".

Пауза – клавиша "W".

Все команды данного экрана рассмотрены, но есть еще несколько моментов. При попадании торпеды в цель появляется сообщение о том, что сонар зафиксировал взрыв. Такая же надпись появляется при попадании снарядом из пушки.

Не все корабли тонут от одной торпеды, поэтому Вам придется стрелять несколькими.

Зарядка торпед происходит автоматически через некоторый промежуток времени.

При опустошении торпедных аппаратов появляется сообщение.

При повороте перископа от оси субмарины более чем на 90 градусов, в ход вступают кормовые торпедные аппараты.

Скорость заднего хода значительно меньше скорости движения вперед.

Возврат в боевую рубку осуществляется нажатием на "SPACE".

Когда вражеский корабль тонет, то появляется информация.

Это самый главный экран, и на его освоение стоит потратить время.

4. Палуба

В этот экран Вы можете войти только в надводном положении. Вы оказываетесь над боевой рубкой. Открывающийся вид гораздо шире, чем тот, который возможен при работе с перископом. При этом появляется сообщение об условиях видимости, а также об угле наблюдения. Конечно, этот режим ничего практически не дает в военном отношении, но зато делает игру несколько более разнообразной.



В этом режиме Вы можете проводить практически все те же действия, что и в боевой рубке.

Внизу также присутствуют информационные строки. Если Вы попытаетесь стрелять торпедами, то попадете в боевую рубку. Стрельба из пушки разрешается. Возврат в боевую рубку - клавиша "SPACE".

5. Приборная панель

На этом экране присутствуют все приборы, необходимые для плавания: глубиномер, датчик скорости хода, компас, часы, указатель положения перископа, а также счетчик торпед в торпедных аппаратах и запасных и счетчик снарядов для пушки.



Внизу информационные строки.

Возврат в рубку - "SPACE".

6. Экран повреждений

Если Вам все-таки не удалось избежать повреждений, то Вы сможете осмотреть их в этом экране. Самое опасное повреждение - это нарушение герметичности лодки. Лодка набирает воду и начинает погружаться. Когда глубина достигает 460 м, то не выдерживают переборки и Вы погибаете.



Плохо, когда повреждаются:

- 1) носовые и кормовые торпедные аппараты, так как Вы не сможете стрелять из них;
- 2) перископ, потому что под водой Вы уже не сможете атаковать противника.

Постарайтесь избежать повреждения вообще. Конечно они со временем устраняются но заметно осложняют ведение боевых действий. Вода откачивается автоматически, а иные повреждения могут восстанавливаться через несколько минут - в зависимости от их сложности. Возврат в рубку - "SPACE".

7. Экран побед

Здесь показывается потопленные Вами корабли, их общий тоннаж, а также количество оставшихся торпед. По окончании игры здесь же выводится Ваш рейтинг в зависимости от тоннажа потопленных кораблей. Возврат в рубку или меню - "M".

Теперь, после того, как рассмотрены все экраны, можно более конкретно поговорить о трех миссиях.

I. Торпедно-пушечная практика

В этой миссии вам надо потопить четыре судна противника: один танкер, два торговых судна и одно пассажирское. Цели не двигаются. Поэтому особых сложностей на этом этапе Вы не встретите.

II. Нападение на караван

В этой миссии Вам будет поручено уничтожения вражеского конвоя, в который входит от одного до шести судов. В этом режиме появляются японские "охотники" и сторожевики.

Вам придется немало потрудиться, чтобы разгромить конвой.

В этих двух миссиях после потопления всех кораблей, Вы попадете в экран побед, а затем в меню. Вы также можете прервать игру и возвратиться в меню, как это описано в режиме боевой рубки.

III Боевое патрулирование

В этом режиме сложности возникают после первых же действий. После того, как Вы выберете тип военного патрулирования, Вам будет предложено идентифицировать вражеского "охотника". Это первые буквы А, В, С, D. Вам же надо нажать на клавиши ("1"- "4"), чтобы показать этот тип. Если Вы неправильно это сделаете, то Вам посоветуют потренироваться. Чтобы Вы не тратили время зря, ниже мы приводим характерные особенности "охотников".

А - мачта на корме, одна труба;

В - нет кормовой мачты, одна труба;

С - мачта, две трубы;

D - нет мачты и носовой пушки.

После того, как Вы правильно идентифицировали "охотника", Вы окажетесь в дополнительном меню.

Игра начинается с сообщения: "Топлива на 50 дней: нажмите клавишу <огонь> для начала патрулирования". Нажмите на "М", перед Вами возникнет карта побережья Центральной Америки с мигающими тремя точками и одной неподвижной, эти три точки - Ваши базы, а неподвижная точка - Ваша субмарина. Теперь, передвигая эту точку клавишами "Y", "H", "9" и "0", ищите вражеский караван. Как только Вы его найдете, все точки на экране остановят свое мигание. Нажмите на "М", и Вы окажетесь в экране карт, дальше действуйте аналогично первым двум миссиям - топите корабли. После того, как все японские корабли уничтожены, Вы вновь можете попасть в режим патрулирования - для этого нажмите клавиши "H"+"9"+"М". Вам сообщается количество оставшегося топлива. Когда его будет совсем мало, Вы сможете возвратиться на базу. Для этого подведя субмарину к мигающей точке (в режиме патрулирования) и, сделав накладку точек, нажмите "М". Вы окажетесь в режиме экрана побед, там отобразится количество Ваших побед, а также присвоенное Вам звание. После этого Вы окажетесь в главном меню. Как видите, в этом режиме, чтобы возвратиться в меню, недостаточно нажать "H"+"9"+"М", - сначала надо вернуться на базу.

Полезные советы

Когда на Вас идет "охотник", и Вы точно знаете, что он Вас заметил, то надо подпустить его на расстояние около 1000 ярдов и пускать торпеду или стрелять из пушки. Если Вы пустите торпеду раньше, то есть шанс, что он увернется.

Догоняя вражеские корабли, дайте полную скорость, можно включить ускоренную временную шкалу. Но следите за расстоянием, чтобы избежать столкновения.

Старайтесь, чтобы перископ был поднят, так как тогда Вы при срочном погружении все-таки сможете наблюдать картину боя и при погружении не потеряются координаты угла наблюдения.

Если Вы все же не смогли поразить "охотник", и он начинает кружить над Вами, то по карте наблюдайте за ним (подняв перископ и опустившись на глубину около 100 м). Когда Вы увидите, что он направляется к Вам, резко всплывайте на 40 метров (чтобы видеть в перископ) и атакуйте.

Если в режиме боевого патрулирования Вы обнаружили цель, но это один-два корабля, и Вы не хотите тратить на них время, то сразу же переключайтесь на патрулирование. Если Вы промедлите, то Вам придется топить и эти корабли.

Чтобы остановить погружение лодки при ее затоплении водой, используйте всплытие, это может замедлить погружение.

Некоторую сложность представляет прицеливание пушкой, а также поиск больших конвоев. Но это останется маленькой тайной. Опытные капитаны смогут дойти и до этого.

Дерзайте!

Любителям имитаторов подводных лодок может быть интересна программа HUNTER-KILLER. В данном случае это не проработка наших экспертов - это наш перевод фирменной инструкции.

HUNTER-KILLER

PROTEK 1983 г.



Вы командир британской подводной лодки типа "S" и выполняете ответственное задание вблизи берегов Германии и Дании во время Второй мировой войны. В зоне Вашего поиска где-то находится вражеская подводная лодка, ее надо отыскать и потопить. Выходить за пределы зоны нельзя, или Вы будете потоплены вражескими миноносцами. Побережье заминировано, так что если Вы не сядете на мель, то подорветесь на mine.

Подводные лодки типа "S"

Лодки этой серии были наиболее совершенными в британском флоте в годы второй мировой войны. Их длина - 217 футов, водоизмещение до 1000 тонн, численность экипажа - 44 человека, мощность дизельных двигателей - 1900 л.с., что давало скорость в надводном плавании до 16 узлов. Мощность электромоторов - 1300 л.с., скорость в подводном положении - 9 узлов. Максимальная глубина погружения - 300 футов.

Управление подводной лодкой

Подводная лодка управляется, как и обычный корабль. Переключивание руля вправо вызывает поворот вправо, аналогично и влево.

Можно управлять и джойстиком. Максимальный угол поворота руля - 70 град. Чувствительность руля зависит от скорости, чем быстрее Вы плывете, тем быстрее выполняется поворот.

Лодка имеет два двигателя - дизельный и электрический, но поскольку дизель нуждается в воздухе, его нельзя использовать в подводном плавании. Здесь надо переключаться на электрические двигатели. При погружении с включенными дизелями загорается контрольная лампочка на панели и, если не принять меры, двигатели необратимо выходят из строя. Работа электрических двигателей зависит от заряда батарей, которые подзаряжаются от дизелей во время надводного плавания. Следите за зарядом батарей, своевременно их подзаряжайте.

Дизельные двигатели - значительно более мощные, чем электрические. Переключение двигателей выполняется клавишей "E". Индикатор в боевой рубке показывает, какой двигатель включен.

Для погружения лодки необходимо выполнить две операции. Во-первых, надо забрать балласт "N", создав нулевую плавучесть. По мере заполнения цистерн водой изменяется состояние индикатора в левом верхнем углу боевой рубки. После этого лодка начнет медленно погружаться. Ускорить погружение можно переложив горизонтальные рули клавишей "B" или джойстиком "назад".

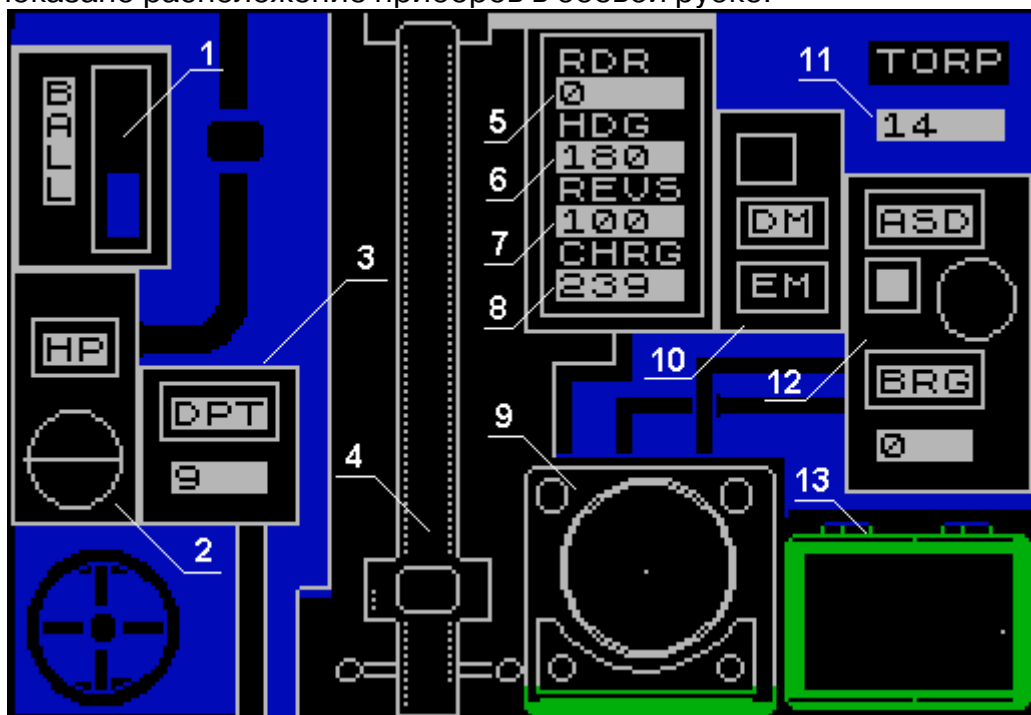
Внимание! Без выполнения обоих этих действий, Вам не удастся погрузиться быстро.

Останавливается погружение клавишей "7" или джойстиком "вперед". Необходимо также немного продуть балласт, пока индикатор уровня балласта не установится в центре.

Для всплытия надо выполнить противоположные действия. Продуть балласт - "B" и

переложить горизонтальные рули - "7"

На рис. 1 показано расположение приборов в боевой рубке.



1. Уровень воды в балластных цистернах.
2. Угол отклонения горизонтальных рулей.
3. Глубиномер.
4. Перископ.
5. Угол поворота руля.
6. Направление носа лодки.
7. Обороты двигателя.
8. Заряд батарей.
9. Экран радара.
10. Индикатор двигателя.
11. Счетчик оставшихся торпед.
12. Дисплей гидролокатора.
13. Эхолот.

Работа с перископом

Перископ находится в боевой рубке. Его подъем или опускание выполняется клавишей "P", а клавиша "V" позволяет осмотреть поверхность моря через перископ. Если в зоне видимости имеется цель, ее можно увидеть. Чем глубже находится лодка, тем выше уровень воды на экране перископа. Глубже 38 футов перископом пользоваться нельзя вообще. Оптимальная глубина для работы с перископом - 34 фута.

Вращается перископ клавишей "O" (по часовой стрелке с шагом 36 град.) или клавишей "I" (против часовой стрелки с шагом 6 град.).

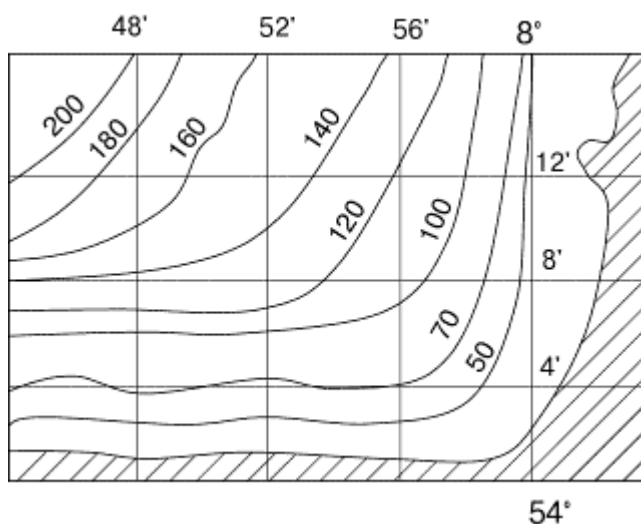
Когда Вы работаете с перископом, в нижней части изображены четыре отсчета. Первый показывает угол поворота перископа. Второй - угол, в направлении которого установлен нос корабля, оно не обязательно должно совпадать с направлением его движения. Третий - расстояние до цели, оно изображается только если противник находится в поле зрения. Четвертый - счетчик торпед.

Для возврата в боевую рубку нажмите клавишу "V".

Радар, гидролокатор, эхолот

Гидролокатор показывает направление на цель в градусах (0 гр. - север), если в радиусе 4 мили есть цель. Он включается автоматически при погружении на глубину более 10 футов. На поверхности автоматически включается радар, его радиус действия 22 мили.

Эхолот доказывает глубину под килем. На рис.2 показана карта глубин в диапазоне от 0 до 200 футов.



Штурманская рубка

Штурманская рубка находится справа от боевой рубки и вызывается клавишей "С". Боевая рубка сдвигается влево, при этом эхолот и гидролокатор остаются на экране и продолжают работать.

На экране изображается также карта, на ней нанесено очертание побережья, показаны границы минных полей (на рис.3), направления сторон горизонта, координатная сетка в градусах и минутах. В левой верхней части карты изображен кружок с маленькой линией, она указывает направление течения. Ваша позиция изображается миниатюрной подводной лодкой, под картой имеются отсчеты скорости течения, показания лага, т.е. Ваша скорость относительно воды, а также координаты последнего положения противника. Координаты даются только в минутах, показания в градусах очевидны. Эти координаты изображаются только в том случае, если противник находится в надводном положении и начинает игру в восточной части зоны. Его курс случаен, но с преимущественным движением на запад. Выход из штурманской рубки и переход в боевую рубку - клавишей "Х".



1. Эхолот.
2. Дисплей гидролокатора.
3. Направление течения.
4. Позиция лодки.

5. Мины.
6. Побережье.
7. Скорость течения.
8. Скорость лодки относительно воды.
9. Координаты противника (в угловых минутах).

Стрельба торпедами

Торпедами можно стрелять только вперед, т.к. у Вас есть только носовые торпедные аппараты. Клавишей "Т" запускается первая торпеда. Вторая и последующие в этом залпе запускаются клавишей "F". Счетчик торпед отсчитывает количество торпед в залпе. Залпы можно растягивать по времени, но не по углу.

Торпеды идут со скоростью 45 узлов и имеют дальность действия - 3 мили. Приблизительно следить за торпедой можно по пузырьковому следу. Если очевидно, что залп проходит мимо, его надо прервать клавишей "А".

Торпеды не могут быть выпущены, если руль не установлен точно прямо или если глубина погружения больше, чем возможная для работы с перископом. Если в момент стрельбы угол поворота перископа отличен от нуля, он автоматически устанавливается на нуль.

Прочие опасности

За Вами постоянно охотятся самолеты. Если один из них будет отмечен поблизости, Вы услышите звук. У Вас есть около двух минут, чтобы погрузиться на глубину ниже 30 футов. Если Вы опоздаете, Вас потопят.

После первой загрузки игры Вам будет предложена практика в стрельбе по неподвижной цели.

Кроме этой игры для одного компьютера существует вариант "HUNTER 2" для игры с двумя компьютерами, соединенными вместе в сеть. В этом случае подводной лодкой противника управляет второй человек с другой клавиатуры. Основные отличия:

- противник может быть как в подводном, так и в надводном положении;
- в этой игре нет воздушных атак;
- значительно снижена емкость батарей обоих противников, чтобы вызвать более частую необходимость всплытия для подзарядки.

Управляющие клавиши

- 5 - руль влево
- 6 - горизонтальный руль вниз (погружение)
- 7 - горизонтальный руль вверх (всплытие)
- 8 - руль вправо
- А - прекратить залп торпедами
- В - продуть балласт
- С - пройти в штурманскую рубку
- Е - переключить двигатели
- F - пуск второй и последующих торпед
- І - вращение перископа против часовой стрелки на 6 град
- Ј - уменьшить обороты двигателей
- К - увеличить обороты двигателей
- N - заполнение балластных цистерн
- О - поворот перископа по часовой стрелке на 36 град.
- Р - подъем/опускание перископа
- Т - нацеливание торпеды и пуск первой торпеды
- У - осмотр поверхности моря через перископ
- Х - выход из штурманской рубки.

Уважаемый читатель

Мы предлагаем вам лично подключиться к распространению обучающих программ для IBM-совместимых компьютеров, а также информационно-поисковых систем, разработанных и выпущенных нашей фирмой.

Основой Вашей деятельности является тот факт, что не все организации, предприятия, учебные заведения и пр. знают о наших программных разработках, а они достойны самого широкого распространения и внедрения.

С другой стороны, мы не знаем всех тех, кто в них нуждается и готовы ВАМ ЛИЧНО или ВАШЕЙ ОРГАНИЗАЦИИ заплатить за точную информацию о потенциальном покупателе, если она приведет к заключению сделки.

Размер оплаты:

- для частных лиц - 8 процентов от суммы сделки (оформляются трудовым соглашением и переводятся по почте или в Сбербанк).

- для организаций - 20% (оформляются договором на распространении научно-технической продукции).

Разумеется, мы рассчитываем на то, что Вы достойно представите нашу фирму потенциальному заказчику и со своей стороны готовы сделать все, чтобы Вы могли представлять нас с гордостью.

Начиная с данного номера "ZX-РЕВЮ" мы начинаем печатать аннотации на наши разработки и будем в каждом номере освещать по одной программе (пакету, комплексу). Сегодня мы начали с самой значительной работы, завершенной на сегодняшний день - комплексу из 22-ух программ "АНГРАМ" (см. следующую страницу).

Вы всегда сможете выбрать для себя, распространением какого продукта вам наиболее интересно заниматься. Следите за нашей информацией.

"ЗЕЛЕНЫЙ ПАКЕТ"

Мы можем значительно поднять Ваш уровень представительства, предложив Вам специально подготовленный комплект материалов, который мы назвали "зеленый пакет".

Вот что в него входит:

1. Инструкция коммерческого представителя.
2. Заверенное письмо-поручение на ведение переговоров от имени фирмы.
3. Два экземпляра проекта "Трудового Договора", предусматривающего отчисление вам 8% от объема продаж, произведенных по результатам Вашего исследования местного рынка. Эти экземпляры должны быть вами заполнены и возвращены нам на утверждение. После утверждения Вам будет возвращен один экземпляр.
4. три полных комплекта коммерческих предложений "ИНФОРКОМа" на текущий период (обновляются один раз в два месяца). По мере их израсходования, вышлем дополнительно по запросу необходимое количество комплектов.
5. По мере наращивания Вашей деятельности и аналогичной деятельности ваших коллег из других регионов, мы будем обеспечивать Вас небольшим бюллетенем по обмену опытом. Наверняка в работе откроются какие-либо нюансы, которые сейчас нельзя предвидеть, но по мере накопления общего опыта можно будет использовать для блага всех.
6. Рабочие или демонстрационные образцы программного обеспечения.

Мы надеемся на Вашу успешную деятельность, но, к сожалению, не в состоянии обеспечить всех желающих данными образцами бесплатно. Они могут быть переданы Вам на ответственное хранение.

Получить желаемые образцы Вы можете по цене "зеленого пакета", которая складывается из:

- 10% текущей цены программного средства, если в качестве образца представлена полноценная рабочая программы или бесплатно, если представлена демо-версия;
- + стоимость дискет;

+ стоимость почтовых операций.

+ полная стоимость дополнительной документации, если она есть.

При любом - успешном или неуспешном характере Вашей деятельности, Вы можете вернуть нам "зеленый пакет" в любое удобное для Вас время вместе с полученными от нас образцами программ на тех же дискетах. Этот возврат Вы должны сделать посредством наложенного платежа в наш адрес, оценив почтовое отправление в ту же сумму, за которую Вы приобретали этот пакет, в данном случае Ваш риск ограничивается только почтовыми расходами.

Мы гарантируем выкуп данного почтового отправления на почте.

У активно работающих дистрибуторов мы выкупим все высланные образцы и далее будем их высылать бесплатно.

Итак, для представленного в данном номере "ZX-РЕВЮ" программного комплекса "АНГРАМ" в составе "зеленого пакета" поставляется рабочая программа "АНГРАМ-1" (урок первый) и цена составляет:

1300 руб. * 10%	=	120 руб.
Дискета	=	25 руб.
План-проспект	=	350 руб.
Почтовые расходы	=	3 руб.
Итого:		498 рублей

(Нас часто спрашивают, чем вызвана несоразмерно высокая цена плана-проспекта. Упреждая недоуменные вопросы укажем, что он является по сути готовым ТЗ и, если Вам понятно, что такое любопытство возможных конкурентов, то Вы поймете, почему мы не даем его бесплатно. Для Вас он послужит подтверждением уровня Вашего представительства при переговорах с потенциальными клиентами, мы уже начали маркетинг "АНГРАМа" и знаем, как он впечатляет клиентов.)

Образец будет Вам выслан на дискете фирмы "КОДАК" с защитным тефлоновым покрытием. Их внешний вид действует на клиентов завораживающе и будет содействовать успеху Вашего представительства.

Бланк для заказа "зеленого пакета" Вы получите вместе с данным номером "ZX-РЕВЮ". Там же указано как и куда производить оплату.

Если Вы не являетесь зарегистрированным у нас подписчиком "ZX-РЕВЮ", значит Вы не получите этот бланк и лишены возможности получить "зеленый пакет" и включиться в нормальную представительскую деятельность. Рекомендуем Вам для начала обратиться к нам с письмом для регистрации.

1. Вы знаете, что мы очень перегружены письмами и не можем реагировать на все обращения и вопросы, но уведомляем вас, что штаты для работы с дистрибуторами уже выделены, и ваши письма будут обработаны мгновенно. Выделен также телефонный номер для Вас.

2. Наша политика по выпуску обучающего программного обеспечения ориентирована на многие годы, ее основы уже заложены и работы давно ведутся. Если Вы готовы на ближайший десяток лет включиться в нашу структуру - мы даем вам "зеленый свет" и снабжаем Вас "зеленым пакетом".

3. Теперь все зависит только от Вас - от Вашей мобильности, контактности, оперативности, от умения общаться с людьми и от знания психологии.

Мы очень на Вас рассчитываем. Нам нужна прямая реклама, доходящая до конечного потребителя, стоимость рекламы в газетах достигла сотен тысяч рублей и пользоваться их услугами мы больше не можем - нам пришлось бы еще в два раза поднять цену своих программ за счет покупателей.

"АНГРАМ" - наша гордость.

"АНГРАМ" - это полный курс английского языка для IBM-совместимых компьютеров.

"АНГРАМ" - это 22 урока английского языка продолжительностью по 16 часов каждый.

"АНГРАМ" - это 22 дискеты 5,25 дюйма емкостью по 360 килобайт.

"АНГРАМ" - это полный курс грамматики английского языка. Это десять тысяч упражнений, десятки тысяч примеров.

"АНГРАМ" - это мощная лексическая поддержка - это десятки тысяч слов, осваиваемых во время занятий с компьютером. В комплексе нет ни одного слова, точный перевод которого Вы не смогли бы получить по ходу занятия.

"АНГРАМ" - это мощная поддержка фонетики. В комплексе нет ни одного слова, к КОТОРОМУ не было бы подробного описания правил произношения.

"АНГРАМ" - это полная гарантия того, что за 500 часов занятия каждый может изучить язык в достаточном объеме для чтения книжно-журнальной и специальной литературы на понимание.

"АНГРАМ" - это более 30 тысяч часов работы профессоров, доцентов, квалифицированных преподавателей и программистов.

"АНГРАМ" - не имеет аналогов ни у нас в стране, ни за рубежом - это комплекс, который дешевле приобрести за любую цену, чем пытаться превзойти.

"АНГРАМ" - ЭТО ВЫЗОВ ВСЕЙ СИСТЕМЕ ОБРАЗОВАНИЯ. МЫ ДОРОГО ЗАПЛАТИМ ТОЛЬКО ЗА ВОЗМОЖНОСТЬ УВИДЕТЬ ЧТО-ЛИБО ПОДОБНОЕ

Один только план-проспект комплекса, содержащий лишь перечень его разделов занимает 120 машинописных страниц.

И ЭТО ДАЛЕКО НЕ ВСЕ !!!

Невероятно - но факт! вся основная работа с комплексом выполняется пятью клавишами. Нет никаких проблем для самостоятельного освоения программ теми, кто не имеет профессиональных навыков работы с компьютером.

Программы комплекса могут быть использованы в учебных заведениях любого типа. Пользователь сам выбирает желаемую глубину проработки материала. Каждая программа гибко настраивается на работу от двух часов (беглый просмотр) до 16 часов. Все учебные материалы построены на реалиях жизни в США. Как припарковать машину, как вести себя в универсаме, как снять квартиру, материалы по истории США и многое другое.

Условия, в которых оказывается обучаемый, работая с нашим комплексом, напоминают те, в которых оказывается человек, впервые попавший за рубеж. Обучение в бою. Методической основой комплекса "АНГРАМ" является контекстная подача материала.

Во время работы учащийся может несколькими нажатиями клавиш получить контекстный перевод любого слова из имеющихся на экране, узнать, как оно произносится.

В течение всего занятия в любую минуту учащийся может получить объективную оценку своих успехов.

Стоимость комплекса из двадцати двух программ на период январь-февраль 1992 года - 17 500 рублей.

ЭТОТ КОМПЛЕКС ДЕШЕВЛЕ ПРИОБРЕСТИ, ЧЕМ ПЫТАТЬСЯ ПРЕВЗОЙТИ ЕГО ВНЕДРЕНИЕ ЗАНИМАЕТ МИНУТЫ. СЛУЖИТЬ ОН БУДЕТ ГОДЫ

Стоимость одной ОТДЕЛЬНОЙ ПРОГРАММЫ на период январь-февраль 1992 года - 1 200 рублей.

(Покупателям всего комплекса план-проспект прилагается бесплатно!)

ВНИМАНИЕ !

СТОИМОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ может непрерывно изменяться в связи с общей экономической ситуацией. Указанные значения действительны только для заказов, оплаченных в период января - февраля 1992 года, во всех прочих случаях прежде чем оформлять заказ и производить оплату необходимо обратиться к нам для уточнения текущего уровня цен.

ГРАФИК ПОСТАВКИ.

Программы отправляются заказчиком, оплатившим заказ, по следующему графику:

январь -92	- 4 программы.
февраль -92	- 3 программы.
март -92	- 3 программы.
апрель -92	- 3 программы.
май -92	- 3 программы.
июнь -92	- 4 программы.
июль -92	- 2 программы.

Полное окончание поставок программ комплекса - не позднее июля 1992 года.

СПОСОБ ПОСТАВКИ.

Все программы доставляется ценной бандеролью в надежной упаковке, исключающей повреждение при почтово-транспортных операциях.

ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА

Гарантийный свидетельством при поставке программного продукта является картонный альбом, в который вложены дискеты, с указанной на нем датой продажи. При его отсутствии поставка выполняется с заверениям гарантийным талоном. Гарантиями обеспечивается:

- бесплатная замена поставочных дисков, неработоспособных в состоянии поставки (в течение месяца после поставки);
- замена с минимальной оплатой при выходе программ из строя по вине пользователя (механическое или электромагнитное повреждение, поражение вирусом на машине пользователя и т.п.) или по истечении месяца после поставки. Минимальная оплата не превышает стоимость дисков + 5% текущей стоимости программного обеспечения + стоимость почтово-транспортных расходов и согласовывается с потребителем.
- соблюдение сроков поставки (полная поставка - до 31.07.92).

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К АППАРАТНО-ПРОГРАММНОМУ ОКРУЖЕНИЮ

1. Полная аппаратно-программная совместимость с IBM PC XT/AT. Надежность функционирования на отечественных модификациях не гарантируется и не обсуждается.
2. Наличие "жесткого" диска ("Винчестера") емкостью не менее 2 МБ.
3. Наличие не менее одного дисковода гибких дисков 5,25 дюйма. На дискетах 3,5 дюйма программы не поставляются. Перенос с дискет 5.25" на 3,25" не обеспечивается (программы защищены от копирования).
4. Операционная система - MSDOS 3.20 и выше.
5. Русификация компьютера в стандарте гост (кодировка альтернативная).
6. Требования к монитору - не специфицируются.
7. Принтер - не требуется.

ПОРЯДОК ОФОРМЛЕНИЯ ЗАКАЗА.

а) направить в наш адрес письмо-заказ с указанием необходимого программного продукта; приложить копию платежного поручения;

Наш адрес: 107241, Москва, Б-241, а/я 37, "ИНФОРКОМ" б) произвести предварительную оплату платежным поручением на наш р/с:

Наш р/с N 500461778 во Фрунзенском коммерческом банке г.Москвы. МФО 201412.