



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 АССЕМБЛЕРА 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 VABS.

Любой из указанных номеров должен реально существовать, иначе будет сообщение об ошибке:

U: NO SUCH LI NE

(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 даёте 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: "Mi ssi ng 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: "Mi ssi ng 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:

T: "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 скроллировать. 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$.
```

Печатавая 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 позволят 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 UDG. 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 UDG заслать оператором POKE другие числа, то компьютер будет искать конструкцию символов UDG 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 UDG в форме треугольника и разместили его в новом адресе - 60 000 (строка 10)

В строке 20 Вы переадресовали системную переменную UDG на новый набор (на адрес 60 000).

Вы по-прежнему не можете иметь в одном наборе символов UDG более 21, но зато можете теперь иметь в памяти компьютера столько наборов, сколько хотите, переключая системную переменную UDG на адреса, с которых начинаются эти наборы.

Попробуем теперь, используя самодельные наборы UDG оживить Ваши программы небольшими цветными картинками. Используем для каждой картинке набор из 20 символов UDG, 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 UDG 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".

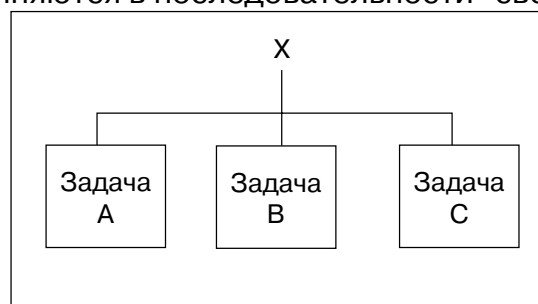


Рис. 1

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).

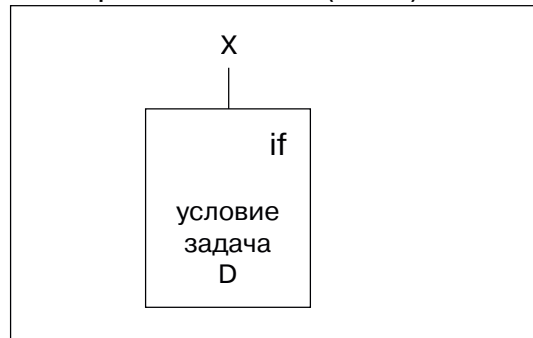


Рис.2

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.

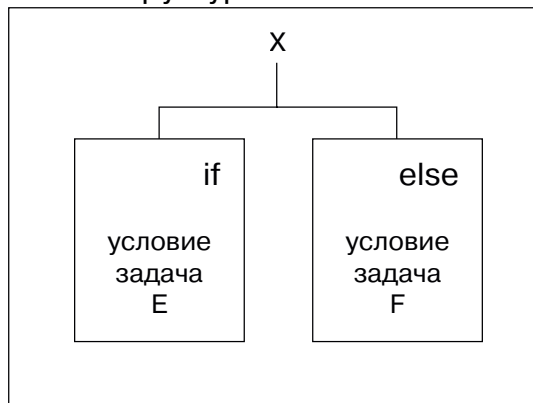


Рис.3

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.

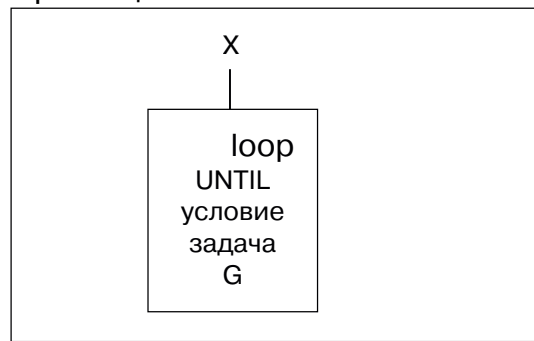


Рис.4

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.

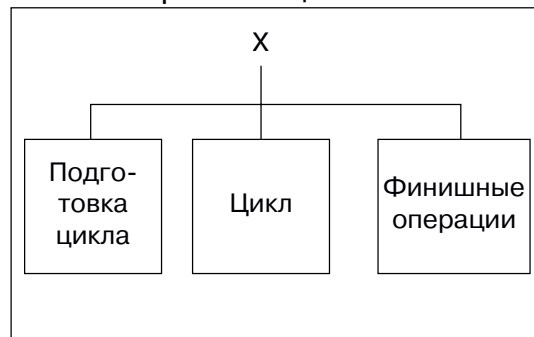


Рис.5

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.

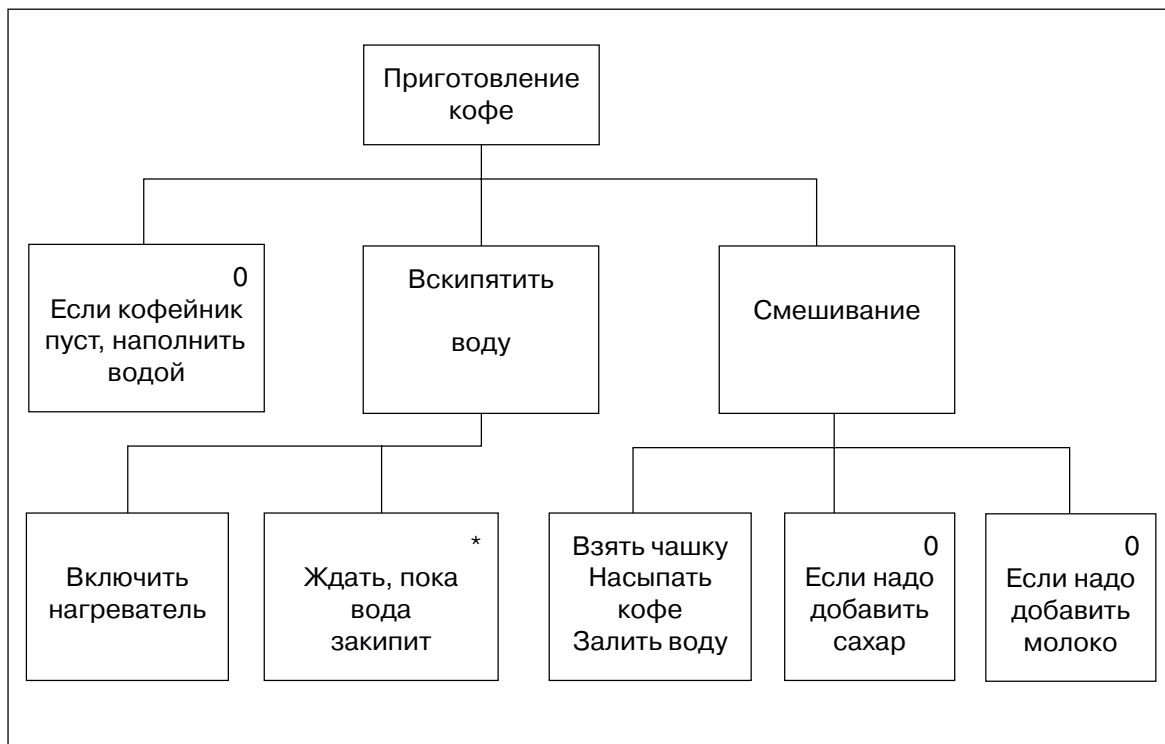


Рис.6

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.

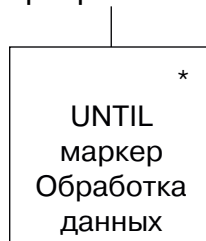
Как структурировать программу?

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).

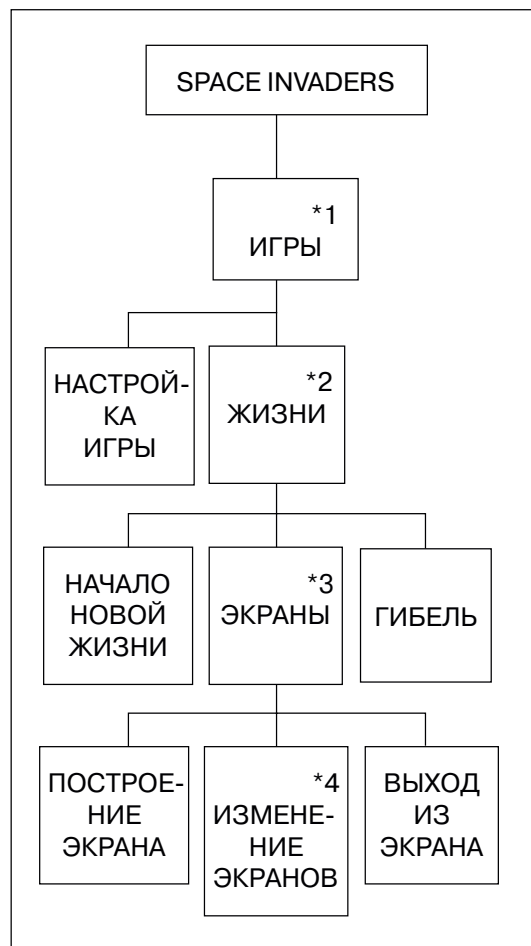


Рис. 7

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.

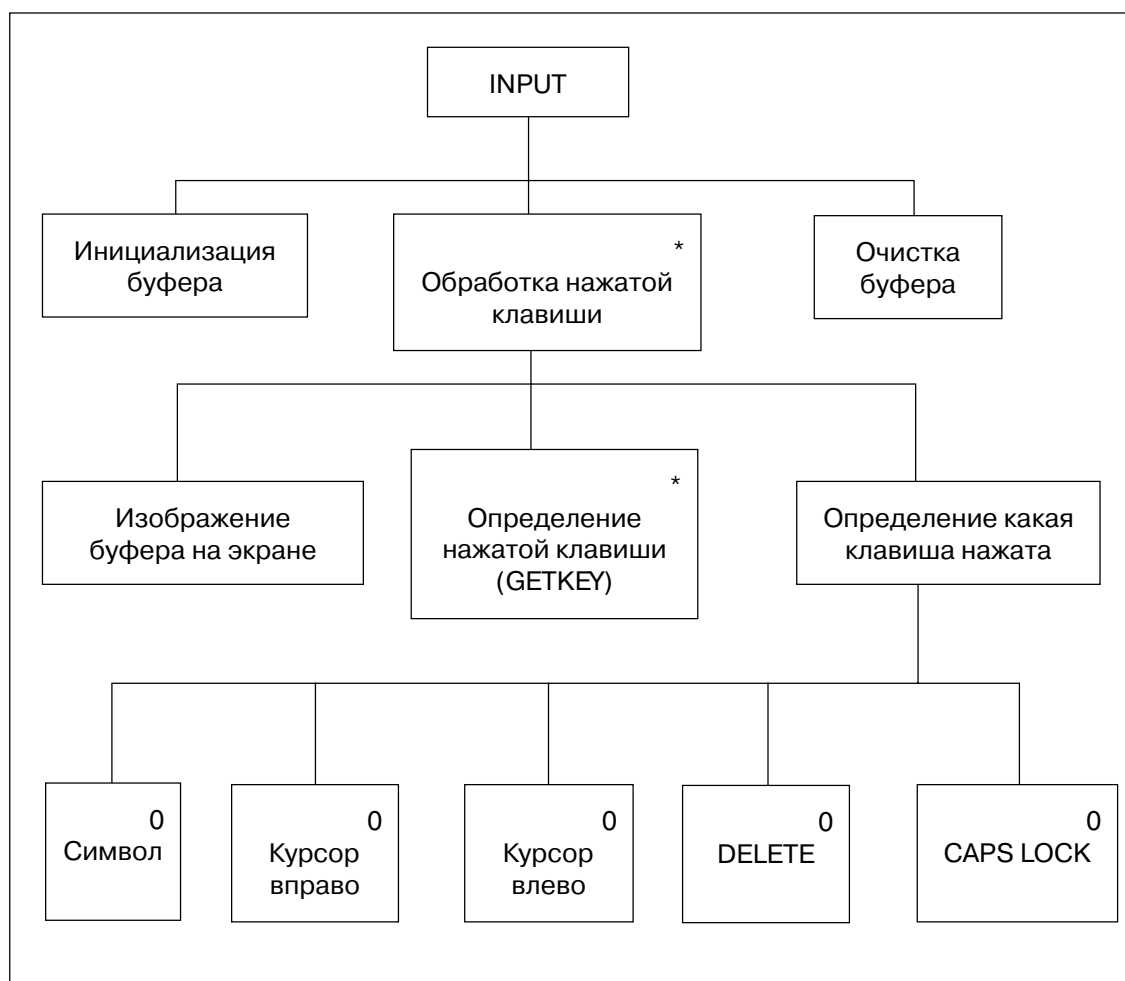


Рис.8

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	; Смещение 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:

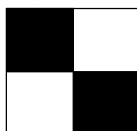
1	0
3	2

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 отыскивает 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.Konopnickiej 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.

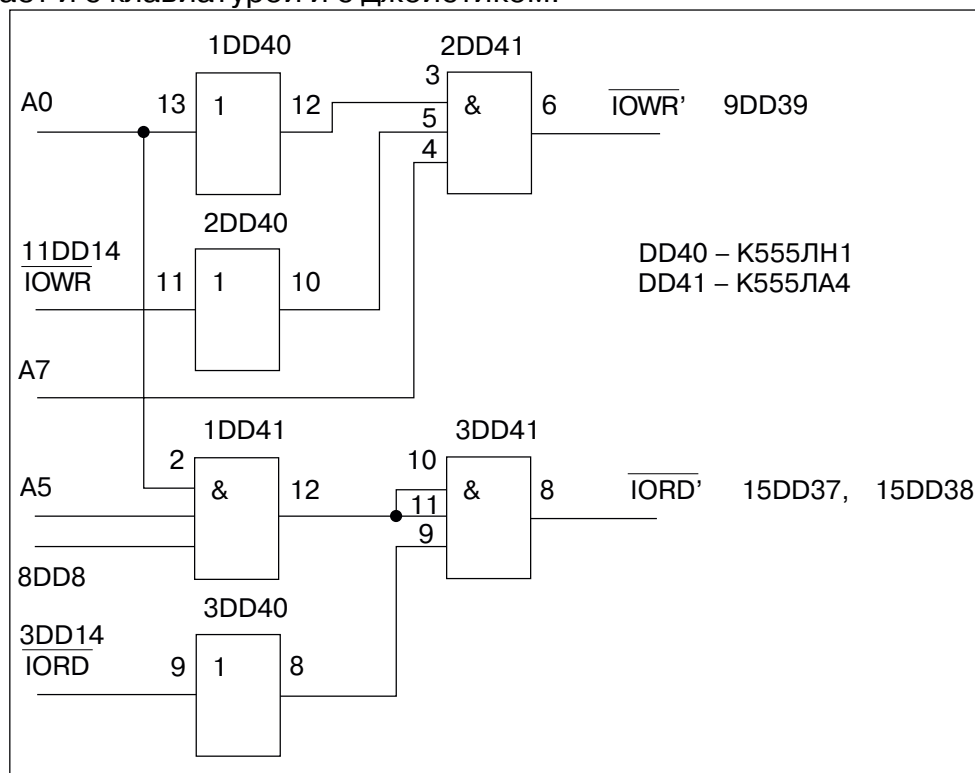


Рис. 1

(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):

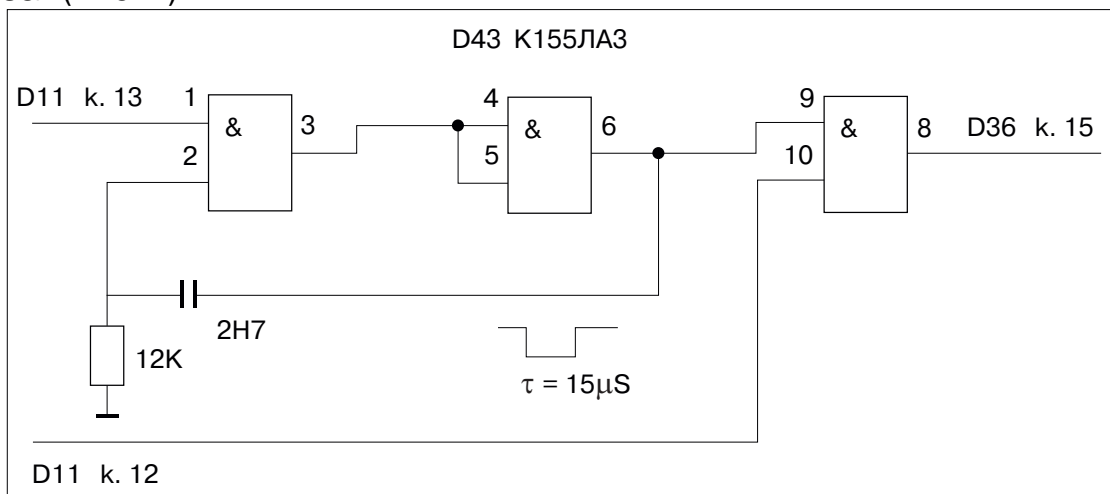


Рис.2

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):

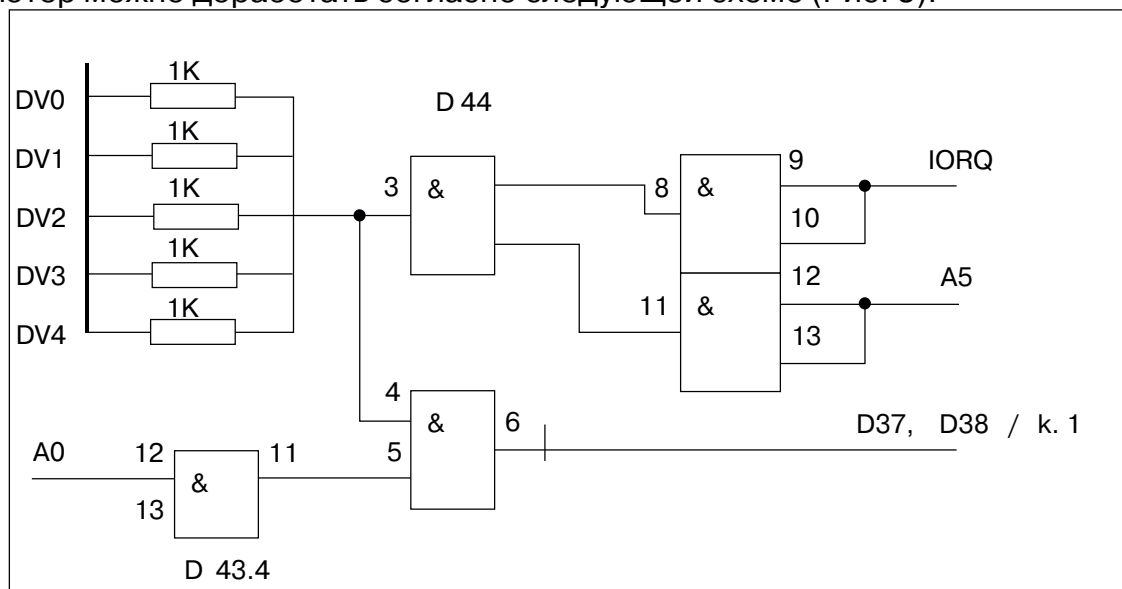


Рис.3

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):

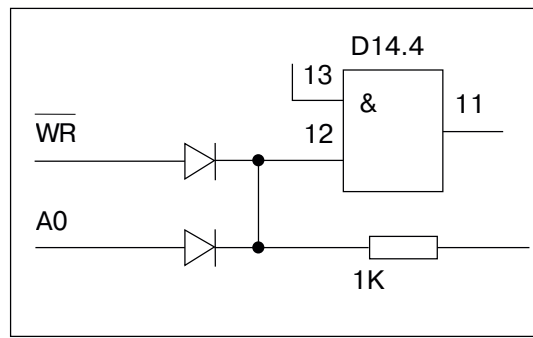


Рис.4

"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 Ивано-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.

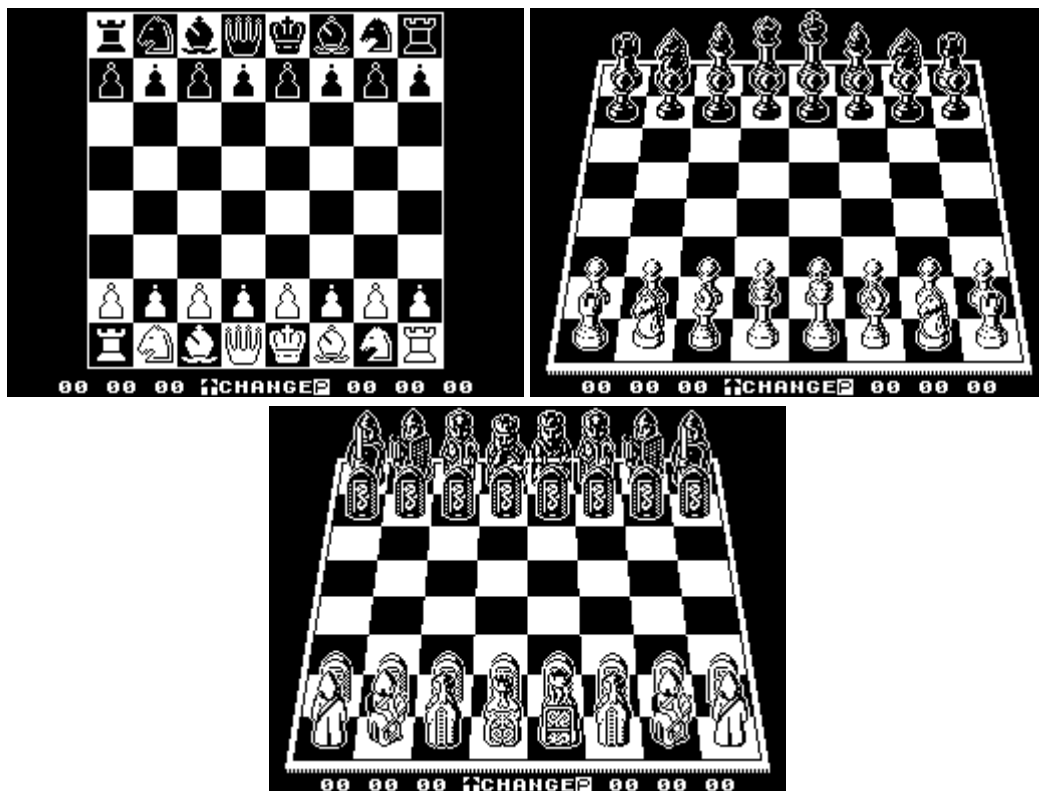
Когда программа загружена впервые, Вы попадаете в режим трехмерной графики, предполагается, что Вы играете белыми против компьютера, уровень силы игры компьютера установлен в A1 (ответный ход выполняется немедленно).

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 настроенный 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".

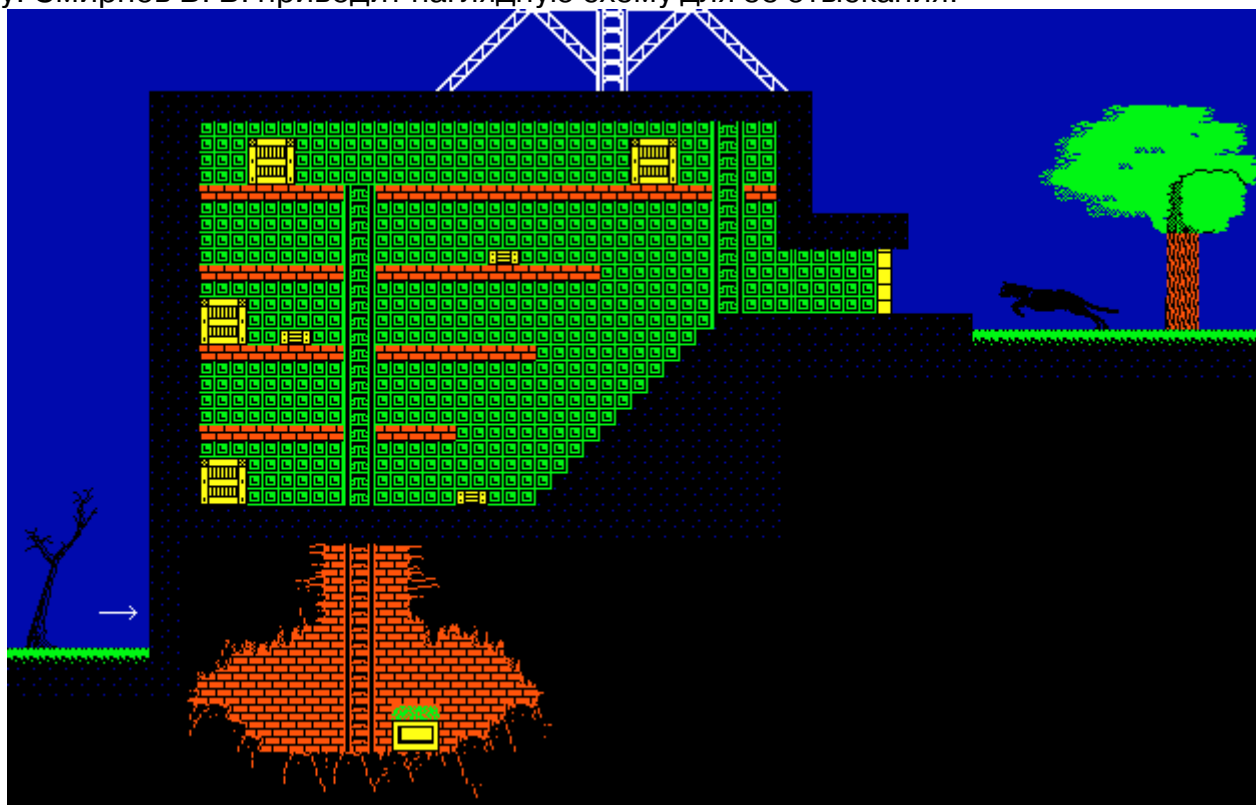
1. Мы написали, что астероиды желательно уничтожать, и за это выдается "наибольшая" премия. Большинство читателей правильно поняли. Конечно надо читать "небольшая" премия.

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

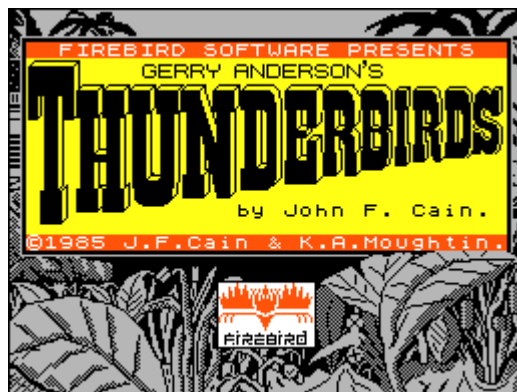
```
1 CLEAR 65399:LET A=65400
2 READ N:IF N>256 THEN RANDOMIZE USR 65400
3 POKE A,N:LET A=A+1: GO TO 2
4 DATA 221,33,0,64,17,96,198,62,255,55,205,86,5,33,0,0
5 DATA 34,175,105,175,60,177,105
6 DATA 50,161,105
7 DATA 62,255,50,97,106
8 DATA 62,243,50,182,105
9 DATA 62,243,50,183,105
10 DATA 195,14,241
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.

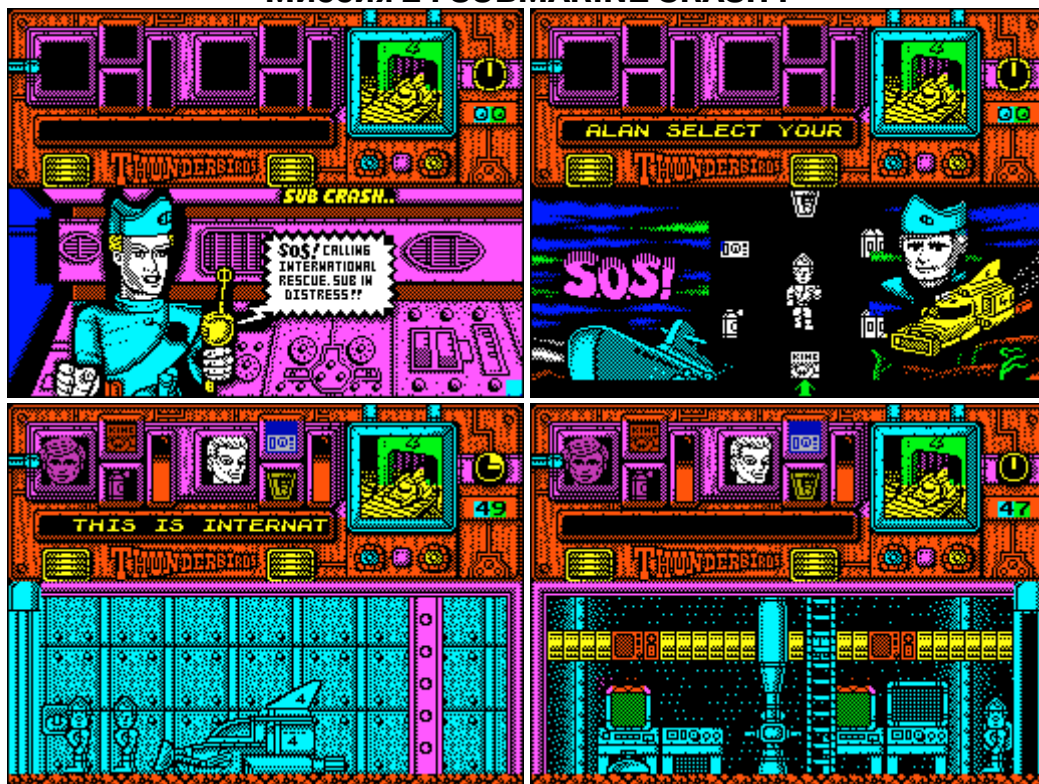
Миссия 1. MINE RESCUE.



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 .



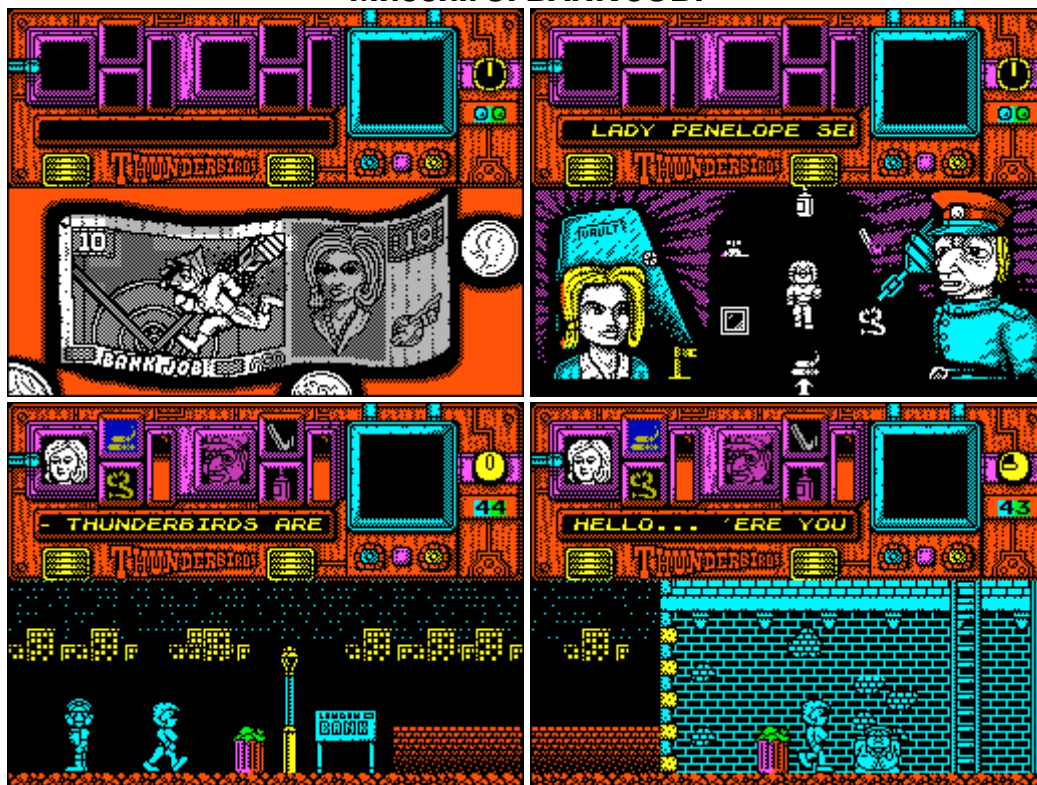
О помощи взывает полузатопленная атомная подводная лодка, в которой в

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.



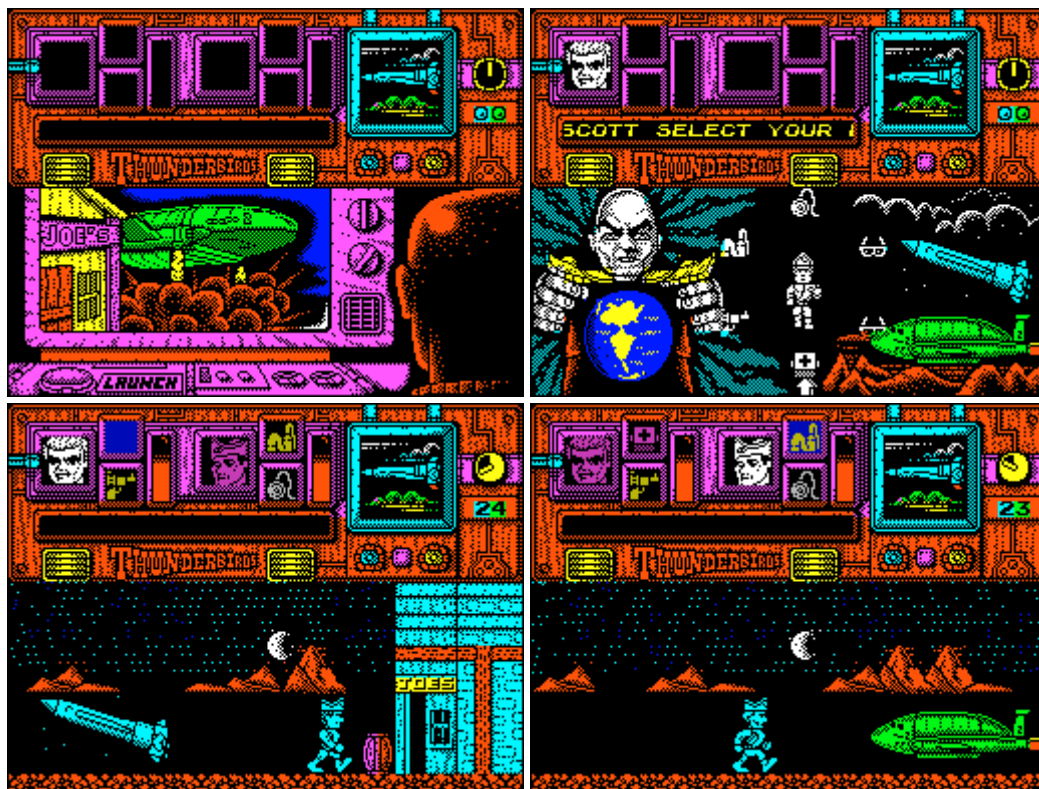
Здесь двум спасателям мистеру Паркеру и леди Пенелопе дается задание выкрасть из тщательно охраняемого частного банка секретные бумаги, имеющие большое государственное значение.

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

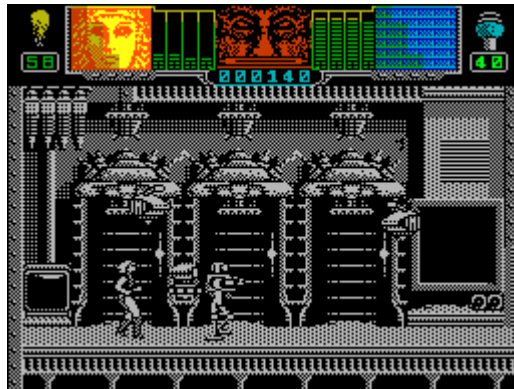
ACTION - казалось бы, что нового можно предложить в этом жанре игр?

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.

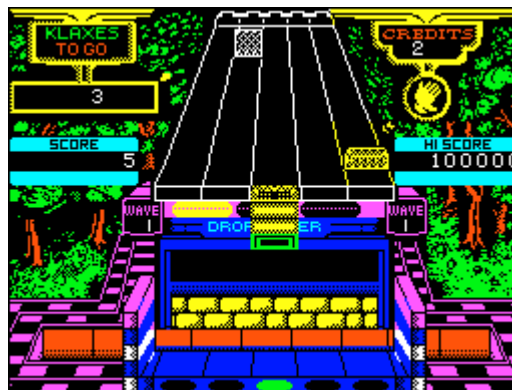


Обозреватель FREE group

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.

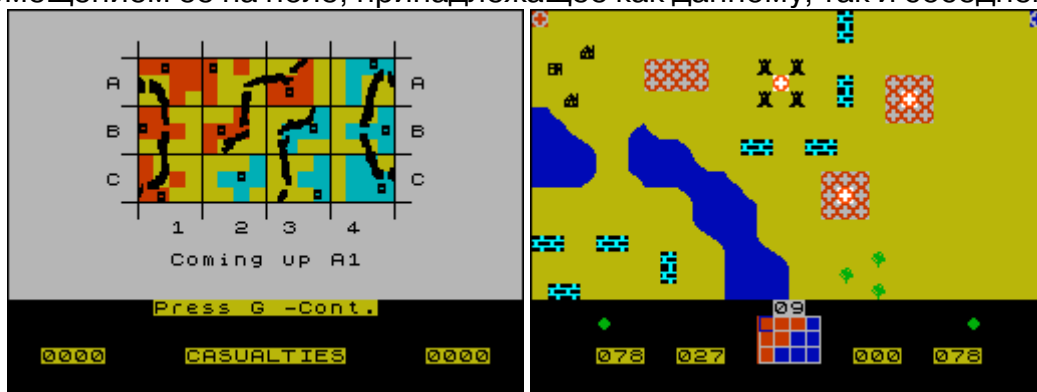
Игра имеет дополнительный режим, в котором два отдельных "Спектрума" соединяются между собой в локальную сеть.

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 "Спектрумами".

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 "к" 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;

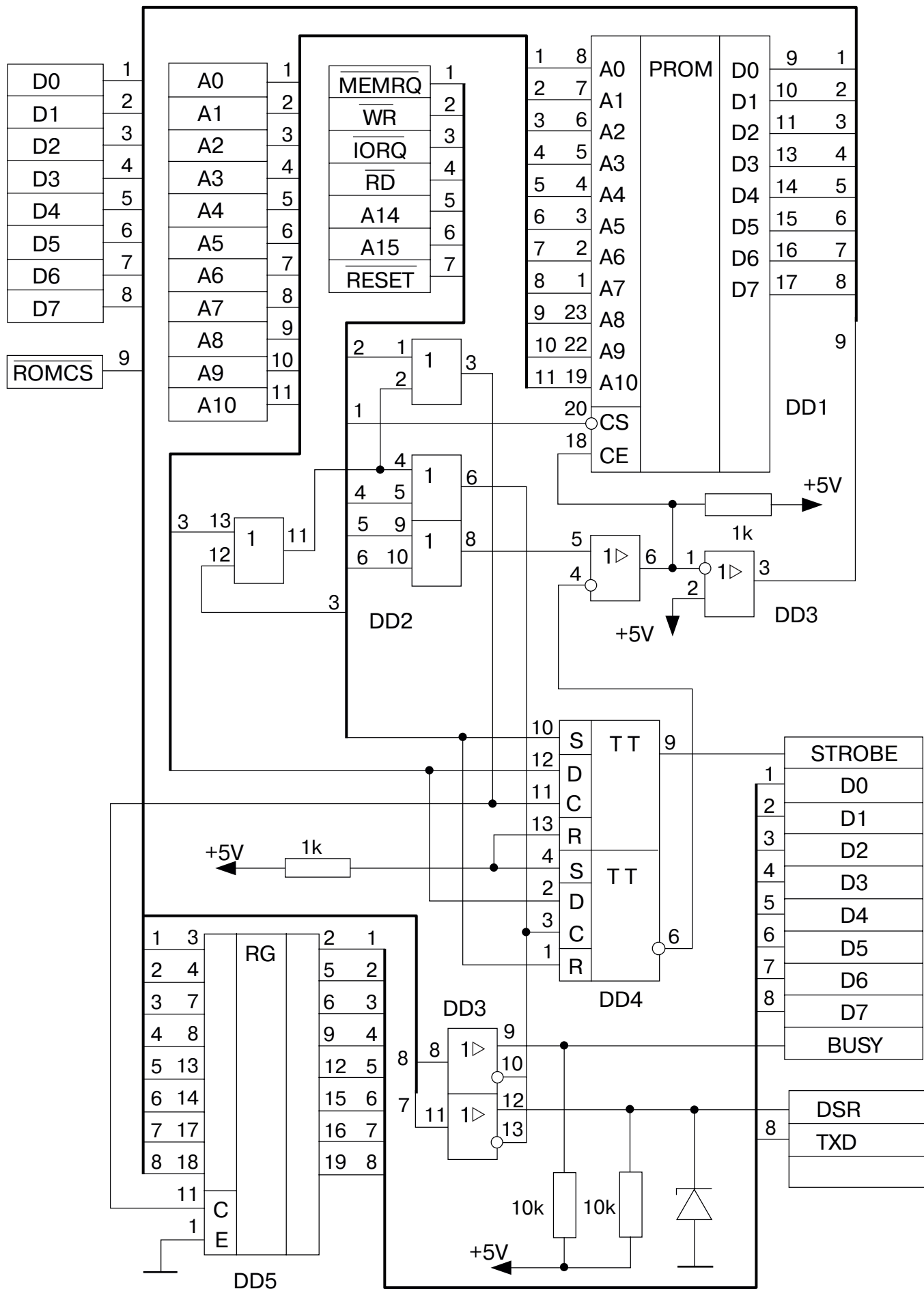
- третий провод соединяет интерфейс с принтером по "земле".

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 №3 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.

СПЕКТРУМ В ШКОЛЕ	1
БЕТА BASIC	4
ВСТУПЛЕНИЕ.....	4
1. Команда ALTER.....	5
2. Команда: AUTO.....	6
3. Команда BREAK.....	6
4. Команда CLOCK.....	6
5. Команда: DEF PROC.....	8
6. Команда: DELETE.....	8
7. Команда: DO	8
8. Команда: DPOKE	9
9. Команда EDIT.....	10
10. Команда: ELSE.....	10
11. Команда: END PROC.....	11
12. Команда: EXIT IF	11
13. Команда: GET	11
14. Команда: KEYWORDS.....	11
15. Команда: LOOP.....	12
16. Команда: ON	12
17. Команда: ON ERROR.....	12
18. Команда: PLOT	13
19. Команда POP.....	14
МАЛЕНЬКИЕ ХИТРОСТИ	16
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	24
Элементы структурированной программы.....	24
Объединение элементов.....	26
Как структурировать программу?	27
Процедура INPUT	29
СЕКРЕТЫ ПЗУ	35
СИСТЕМНЫЕ ПЕРЕМЕННЫЕ.....	35
Флаговые переменные.....	37
Подпрограмма "CURSOR LEFT".....	39
Подпрограмма "CURSOR RIGHT".....	39
Подпрограмма "CARRIAGE RETURN".....	39
Подпрограмма "PRINT COMMA".....	39
Подпрограмма "PRINT A QUESTION MARK".....	39
Подпрограмма "CONTROL CHARACTERS WITH OPERANDS".....	39
Подпрограмма PRINTABLE CHARACTER CODES.....	41
Подпрограмма "POSITION STORE".....	41
Подпрограмма "POSITION FETCH".....	41
Подпрограмма "PRINT ANY CHARACTERS".....	41
Подпрограмма "PRINT ALL CHARACTERS".....	43
Подпрограмма "SET ATTRIBUTE BYTE".....	44
Подпрограмма "MESSAGE PRINTING".....	44
Подпрограмма "PO-SAVE".....	45
Подпрограмма "TABLE SEARCH".....	45
Подпрограмма "TEST FOR SCROLL".....	46
Подпрограмма "TEMPORARY COLOR ITEMS".....	47
Подпрограмма "CLS COMMAND".....	47
Подпрограмма "SCROLLING".....	48
Подпрограмма "CLEAR LINES".....	49
Подпрограмма "CL-ATTR".....	49
Подпрограмма "CL-ADDR".....	49
Подпрограмма "COPY COMMAND".....	50
Подпрограмма "COPY-BUFF".....	50
Подпрограмма "CLEAR PRINTER BUFFER".....	50
Подпрограмма "COPY-LINE".....	50
FORUM	52

АВТОСТАРТ ПРОГРАММ В МАШИННЫХ КОДАХ.	52
ДЕКОДЕР РАДИОПЕРЕДАЧИ.	52
К ВОПРОСУ О СОВМЕСТИМОСТИ.	53
COPY-COPY.	54
ОРГАНИЗАЦИЯ КЛУБОВ.	57
PSI CHESS.	58
ВСТУПЛЕНИЕ.	58
ИГРА.	59
ДЕТАЛЬНОЕ ОПИСАНИЕ ПРОГРАММЫ.	60
FORUM.	66
"Взрыв сверхновой".	68
"Cloaking device".	69
Е.С.М. "JAMMER".	69
Загадочные явления.	69
SABOTEUR-2.	70
POKES.	72
СЛОВО ЭКСПЕРТАМ.	80
THUNDERBIRDS 1-4.	80
Миссия 1. MINE RESCUE.	81
Миссия 2. SUBMARINE CRASH.	81
Миссия 3. BANK JOB.	82
Миссия 4. EVIL HOOD.	82
Запуск игры.	83
Экран.	83
Управление.	83
Применение предметов.	84
Сила.	84
Некоторые советы.	84
Пароли.	85
HAMMERFIST.	86
MYTH.	87
KLAX.	87
OVERLORD.	88
Настройка программы.	89
Некоторые советы.	90
OVERLORDS.	91
Порядок игры.	91
Цель игры.	92
Фигуры.	92
Территория.	92
Сражения.	92
Пополнение армии.	92
Информация на экране.	92
Карта.	93
Повторение игры.	93
Запись игры на ленту.	93
Правила ведения войны.	93
Дополнительная информация при работе с двумя "Спектрами".	93
ZX LPRINT III.	94