



ОТ **А** ДО

З Х

© Малое предприятие "ФОРУМ", 1992 г.

Введение

16К байтная программа - монитор спектра - является сложной программой, написанной в машинных кодах Z-80, общая структура монитора достаточно ясна, и в нем можно выделить три главные части:

- 1) программа ввода/вывода
- 2) интерпретатор BASIC
- 3) обработка выражения

Однако, эти части также являются у MNT слишком емкими для изложения, и эту книгу программ разобьем на 10 частей, каждую из которых опишем ниже.

Программы запуска таблицы

Программа монитор запускается различными программами, которые вызываются одним единственным байтом команды RST, используются все коды RST, например, RST 0008 используя для сообщения о синтаксических или таймерных ошибках.

Таблицы в этой части программы MNT содержат расширенные форматы знаков и коды клавиатуры.

Программы клавиатуры

Клавиатура сканируется каждые 1/50 секунды, и программа клавиатуры возвращает требуемые (необходимые) коды символов, все клавиши клавиатуры повторно опрашиваются, и, если они были нажаты, программа клавиатуры (драйвер клавиатуры) принимает их во внимание.

Программа управления динамиком (драйвер динамика)

"Спектр" имеет единственный "бортовой" динамик, и нота генерируется повторным использованием соответствующей команды вывода (OUT), главной целью программы управления должно быть обеспечение того, чтобы подавалась необходимая нота необходимой высоты тона в течение требуемого промежутка времени.

Программа управления кассетным магнитофоном

В "спектруме" есть широкий набор кодов и высокий уровень управления кассетным магнитофоном, что является одной из наиболее ценных особенностей машины.

BASIC-программы или блоки данных рассматриваются одинаковым образом и сопровождаются заголовком (семнадцать байтов), который записывается первым. Этот заголовок описывает блок данных, которые пишутся за ним.

Одним из недостатков этой системы является то, что она не дает возможности записывать программы с какой-либо защитой вообще. (Защита от несанкционированного доступа).

Программа управления дисплеем и принтером

Все остальные программы ввода/вывода "спектрума" являются через "каналы и области информационных потоков".

В стандарте "спектрума" "ввод" возможен только от клавиатуры, но "вывод" может быть направлен на принтер, в верхнюю часть TV-дисплея. Главной программой "вывода" в этой части программы - MNT является EDITOR (Редактор), который позволяет пользователю вводить символы в нижнюю часть TV-дисплея.

Программа драйвер печати является довольно медленной программой, так как эта же программа используется для всех случаев. Например, добавлением единственного байта в "дисплейный файл" и включает в себя рассмотрение текущего состояния OVER и INVERSE в каждом случае.

Программы исполнения

В этой части программы MNT должна располагаться процедура инициализации и главные циклы исполнения интерпретатора BASIC:

В "спектруме" строка BASIC, возвращенная редактором, проверяется на правильность синтаксиса и затем записывается в область программы, если эта строка начиналась с номера строки. В противном случае исполняется сразу. Исполнение может в свою очередь привести к исполнению следующей команды, (наиболее ясно показано в случае RUN).

Интерпретация команд и строк BASIC

Монитор рассматривает строку BASIC как набор утверждений, в свою очередь, каждое утверждение - как начало отдельной команды. Для каждой команды существует "управляющая (командная) программа", выполнение которой в машинных кодах осуществляет интерпретацию.

Оценка выражения

"Спектрум" имеет возможность наиболее всесторонней оценки выражения и принимает в расчет широкий диапазон различных типов переменных, функций и операций. Эта часть монитора является довольно медленной, т. к. должны быть рассмотрены все возможные альтернативы.

Особенно хорошо реализуется обработка строк последовательностей. Все простые строки (последовательности) управляются "динамически", и старые копии исправляются, когда они являются избыточными. Это является средством для того, чтобы избежать "замусоривания" (загромождения).

Арифметические программы

"Спектрум" имеет два формата представления чисел. Целые числа в диапазоне -65535 до +65535 представляются в "интегральном" или "укороченном" формате и в то же время все другие числа представляются в пятибайтном формате с плавающей запятой. К сожалению, настоящая версия монитора имеет две ошибки:

1. Ошибка в "делении", после чего 34-бит результата деления теряется.
2. Величина - 65536 иногда берется в "коротком формате" и иногда в формате с ПЗ, и это ведет к нарушениям вычислений (ошибкам).

Вычислитель с ПЗ

Калькулятор (вычислитель) "спектрум" оперирует с числами и строками, и его действия определяются литерами. Следовательно, он может быть рассмотрен как внутренний язык стекового действия, реализованный в калькуляторе.

Эта часть программы MON содержит программы для всех математических функций SIN X, EXP X, LN X & ATN X, получены путем развития полиномов Чебышева, и более полно изложены в приложении.

В целом 16 К программа монитор представляет собой крайне широкий диапазон (ряд) различных команд функций BASIC. Программисты, однако, всегда предпочитают краткость, и, следовательно, программа написана скорее для краткости, чем ради быстродействия.

Дисассемблер. Программы перезапуска и таблицы

Запуск

Маскируемое прерывание запрещается, и в пару регистров (DE) засылается старший адрес доступного пользователю ОЗУ.

000 START

DI

Запретить "прерывание клавиатуры"

XOR A +00

Для "запуск"(но +FF для "NEW")

LD DE +FFFF

Старший адрес ОЗУ

JP 11CB,START/NEW

Переход по адресу 11 CBH

Перезапуск по "ошибке"

Указатель ошибки устанавливается для указания мест ошибки.

0008 ERROR-1

LD HL,(CH-ADD)

Значение адреса, кот. достигнут интерпретатором, копируется в указатель ошибки

LD (X-PTR),HL

JR 0053, ERROR-2

до продолжения процедуры

Перезапуск "печать символа"

Регистр A содержит код символа, выводимого на печать

0010 PRINT-A-1

JP 15F2,PRINT-A-2

Переход по адресу 15F2H

DEFB +FF, +FF, +FF, +FF, +FF неиспользуемые ячейки

Перезапуск "получение символа"

Извлекается содержимое ячейки, адрес которой находится в CH-ADD. Возврат происходит, если полученный символ разрешен для печати. В противном случае, содержимое CH-ADD инкрементируется, и процедура повторяется.

0018 GET-CHAR

LD HL,(CH-ADD) Извлечь значение, адресуемое CH-ADD

LD A,(HL)

001C TEST-CHAR

CALL 0070,SKIP-OVER Определить является ли
символ разрешенным для печати
RET NC Возврат, если символ печатный

Перезапуск "получение следующего символа"

Точка входа для вычисления с плавающей точкой находится по адресу 335B.

0028 FR-CALC

JP 335B, CALCULATE Переход по адресу 335B

DEFB +FF, +FF, +FF, +FF, +FF Неиспользуемые ячейки

Перезапуск THE "MAKE BC SPACES"(Резервирование "BC" ячеек памяти)

Эта программа резервирует свободные ячейки в рабочей области. Число ячеек задается в регистровой паре BC.

0030 BC-SPACES

PUSH BC Сохранить число ячеек

LD HL,(WORKSP) Прочитать адрес начала рабочей

PUSH HL области и сохранить его прежде,

JP 169E, REZERVE чем продолжить далее

Программа THE "MASKABLE INTERRUPT "ROUTINE"(Маскируемое прерывание)

Увеличение счетчика реального времени и сканирование клавиатуры всякий раз, когда происходит маскируемое прерывание.

0038 MASK-INT

PUSH	AF	Сохранить текущее значение
PUSH	HL	регистров,
LD	HL,(FRAMES)	младшие два байта системного
INC	HL	счетчика инкрементирующегося через
LD	(FRAMES),HL	каждые 20 МС старший байт
LD	A,H	счетчика инкрементируется только
OR	L	тогда, когда значение младших двух
JR	NZ,0048, KEY-INT	байтов равно нулю.
INC	(FRAMES-3)	
0048	KEY-INT	
PUSH	BC	Сохранить текущее значение регистров
PUSH	DE	
CALL	028, KEYBOARD	Сканировать клавиатуру
POP	DE	Восстановить содержимое регистров
POP	BC	
POP	HL	
POP	AF	
EI		Разрешить маскируемое прерывание
RET		перед возвратом из подпрограммы

Программа "ERROR-2" (Ошибка-2)

Интерпретатору возвращается адрес входа в таблицу (DEFB), указывающий на ошибку, которая произошла, код ошибки из таблицы "DEFB" загружается через ячейку ERR-NR. Машинный стек очищается до перехода вперед, чтобы очистить стек калькулятора.

0053 ERROR-2.

POP	HL	Адрес из стека указывает на код
LD	L,(HL)	
0055		
LD	(ERR-NR),L	ошибки, загружаемой в ERR-NR
LD	SP, (ERR-SP)	очистить машинный стек до выхода
JP	1605, SET-STK	через SET-STK (стек калькулятора)
DEFB	+FF, +FF, +FF, +FF	Неиспользуемые ячейки

DEFB +FF, +FF, +FF

Программа "NON-MASKABLE INTERRUPT"
(Немаскируемое прерывание)

Эта программа не используется в стандартном "спектруме", но этот код дает возможность осуществить системный сброс с последующей активизацией сигнала MNI. Системная переменная по адресу 5CBO, называемая здесь NMIADD, должна быть равна 0 для осуществления сброса (RESET).

0066	RESET	
PUSH	AF	Сохранить содержимое регистров
PUSH	HL	
LD	HL, (NMIADD)	Два байта NMIADD должны быть
LD	A, H	равны 0, чтобы произошел
		системный сброс
OR	L	
JR	NZ, 0070, NO-RESET	Замечание: это должна была
		быть "JRZ"
JP	(HL)	Переход на запуск (START)
0070	NO-RESET	
POP	HL	Восстановить содержимое регистров
POP	AF	
RETN	Возврат	

Подпрограмма "CH-ADD+1".

Адрес, хранящийся в CH-ADD извлекается, инкрементируется и записывается обратно. Читается содержимое ячейки, адрес которой хранится в CH-ADD, точки входа TEMP-PTR1 и TEMP-PTR2 используются в CH-ADD для установки временного периода.

0070	NO-RESET	
POP	HL	Восстановить содержимое регистров
0074	CH-ADD+1	
LD	HL, (CH-ADD)	Прочитать адрес
0077	TEMP-PTR1	
INC	H	Увеличить указатель

0078	TEMP-PTR2	
LD	(CH-ADD),HL	Установить CH-ADD
LD	A,(HL)	Прочитать в аккумулятор адресуемое значение,
RET		Возврат

Подпрограмма "Переходов (SKIP-OVER)"

При входе в подпрограмму проверяется код символа в регистре A, и определяется, является ли он разрешенным к печати. При обнаружении специальных кодов содержимое пары HL инкрементируется один раз или дважды, а содержимое CH-ADD изменяется соответственно.

007D SKIP-OVER

CR	+Z1	Возврат, если код простого символа
RET	NC	Флаг C сброшен
CP	+0D	Возврат, если обнаружен конец строки
RET	Z	
CP	+10	Возврат, если код от +00 до +0F
RET	C	Флаг C установлен
CP	+18	Возврат, если код от +18 до -20
CCF		Флаг C установлен
RET	C	
INC	HL	Увеличение SKIP-OVER
CP	+16	Переход, если код от +10 до +15
JR	C, 0090, SKIRS	(INK до OVER)
INC	HL	Увеличение SKIP-OVER (AT&TABL)
0089 SKIPS		
SCF		Возврат, флаг C установлен
LD	(CH-ADD),HL	CH-ADD содержит соответствующий адрес
RET		Возврат

Программы клавиатуры "THE KEYBOARD ROUTINES"Подпрограмма "Сканирования клавиатуры"
(THE KEYBOARD SCANNING)

Эта очень важная подпрограмма вызывается как главной подпрограммой клавиатуры, так и программой INKEY (в сканировании).

При всех запросах регистр E при возврате содержит значение в диапазоне +00 до +127, которое различается для каждой из сорока клавиш, или значение +FF, если клавиша не нажата.

Регистр D при возврате содержит величину, которая фиксирует, какая единственная клавиша регистра нажата. Если нажаты обе клавиши, то D и E регистры при возврате содержат величины для CAPS SHIFT и SYMBOL SHIFT клавиши соответственно. Если не было нажато ни одной клавиши, то регистровая пара DE при возврате содержит +FFFF. Флаг нуля устанавливается при возврате, если было нажато более двух или ни одна из пары нажатых клавиш не является клавишей регистра.

028E KEY-SCAN

LD L, +2F Начальной величиной клавиши для каждой строки Д.Б. +2F, +2E... +28 (восемь строк)
LD DE, +FFFF установка DE в состояние "нет ввода"
LD BC, +FEFE C - адрес порта
B - счетчик

Затем переходим в цикл, делается восемь шагов с присвоением на каждом шаге различных начальных значений клавиатуры и сканированием различных строк по пять клавиш (Первая строка - пять клавиш) первая строка - CAPS SHIFT, Z, X, C, V)

0296 KEY-LINE

IN A, C Чтение из заданного порта
SRL A Нажатая клавиша в строке установит
AND +1F соотв. бит (от бита 0 до бита 4)
JR Z, 02AB, KEY-DONE Переход вперед, если ни одна из пяти клавиш в строке не нажата
LD H, A Код клавиши (биты) передает-
LD A, L ся в регистр H при получении величины

инициализированной клавиши.

029F KEY-3KEYS

INC D Если на клавиатуре нажато три клавиши, то регистр D уже не содержит кода +FF и осуществляется возврат (если это случилось).

RET NZ

02A1 KEY BITS

SUB +08 повторно вычитается из текущей

SRL H величины кода клавиатуры, до тех пор

JR NC,02A1,KEY-BITS пока бит клавиши не будет найден

LD D,E в D регистр копируется ранее

определенная величина кода клавиши.

LD E,A в E регистр засылается новая величина

кода клавиши.

JR NZ,029F,KEY-3KEYS Если обнаружена вторая или

возможно третья нажатая клавиша в этой

строке, то программа возвращается назад.

02AB KEY-DONE

DEC L Строка должна сканироваться так, что

RLC B начальный код клавиатуры уменьшает-

JR C,0296,KEY-LINE ся для каждого следующего цикла.

Теперь делается четыре проверки

LD A, D Принимается некое значение кода кла-

виатуры, которое до сих пор имел ре-

INC A гистр D, содержащий +FF, т. е. нажата

RET Z только одна клавиша или нет нажатия

CP +28 Принимается код клавиатуры для пары

RET Z клавиш, если в D находится код CAPS SHIFT.

CP +19 Принимается код клавиатуры для пары

RET Z клавиш, если в D находится SYMBOL SHIFT.

LD A, E Возможно наличие SYMBOL SHIFT в ре-

LD E, D гистре E, так что этот случай дол-

LD D, A жен быть учтен.

CP +18

RET Возврат с установкой флага нуля

Если это был SYMBOL SHIFT и другая клавиша, в противном случае - сброс.

Подпрограмма "Клавиатура" (KEYBOARD)

Эта подпрограмма вызывается в любом случае, когда происходит маскируемое прерывание. В обычном режиме это случается каждые 20 мс. Целью этой подпрограммы является сканирование клавиатуры и декодирование значения клавиш. Полученный код будет, если повторный опрос позволит, передан в системную переменную LAST-K.

Когда код переносится в эту системную переменную, бит 5 флагов устанавливается в активное состояние, указывая, что нажата новая клавиша.

02BF KEYBOARD

CALL 028E, KEY-SCAN Запись значения клавиши в регистровую пару DE, по медленным
RET NZ возвратом, если установлен флаг нуля.

Двойная система "KSTATE системные переменные" (KSTATE1-KSTATE3 и KSTATE4-KSTATE7) используется с этого момента. Два набора позволяют обнаружить нажатие новой клавиши (используется один набор), в то время, пока еще система находится внутри периода повторного опроса предыдущего нажатия (детальное описание в следующем наборе).

Набор будет освобожден для ввода новой клавиши, если клавиша нажимается на время около 1/10-тысячной секунды, т. е. на пять циклов опроса клавиатуры.

LD HL, KSTATE Старт с KSTATE0

02C6 K-ST-LOOP

BIT 7, (HL) Переход вперед, если набор свободен,
т.е. KSTATE0/4

JR NZ, 02D1, K-CH-SET Содержит +FF

INC HL Однако, если набор не свободен,

DEC (HL) уменьшает "свой" счетчик пяти
DEC HL вызовов" и когда он достигает

JR NZ, 02D1, K-CH-SET значения 0, набор считается

LD (HL), +FF свободным.

После обработки первого набора изменяется счетчик и рассматривается второй набор.

02D1 K-CH-SET

LD	A,L	Запись младшего байта адреса
LD	HL, +KSTATE4	и возврат назад, если должен
CP	L	быть рассмотрен еще и второй
JR	NZ,02C6,K-ST-LOOP	набор.

Теперь возврат, если значение клавиши фиксирует "нет нажатия" или нажата только клавиша регистра.

CALL	031E,K-TEST	Делает необходимую проверку и
RET	NC	возврат, если необходимо,

изменяет также значение клавиши в "главный (основной) код".

"Дребезг" клавиатуры теперь отделяется от нового нажатия клавиш.

LD	HL, +KSTATE0	Смотрим первую строчку в KSTATE0
CP	(HL)	переход вперед, если состояние ко-
JR	Z,0310,K-REPEAT	дов обнаруживается повторно.
EX	DE,HL	Спасение адреса KSTATE0.
LD	HL, +KSTATE4	Теперь рассмотрим KSTATE4.
CP	(HL)	Переход вперед, если состояние ко-
JR	Z,0310, K-REPEAT	дов обнаруживается повторно.

Но новая клавиша не будет введена до тех пор, пока один из наборов системных переменных KSTATE не будет свободен.

BIT	7, (HL)	Рассмотрим второй набор
JR	NZ,02F1,K-NEW	Переход вперед, если свободно.
EX	DE, HL	Теперь рассмотрим первый набор
BIT	7, (HL)	Продолжаем, если набор "свободен"
RET	Z	Но с выходом из программы
		"клавиатура", если нет

Новая клавиша введена, но перед тем, как системная переменная LAST-K может быть заполнена, KSTATE системная переменная использованного набора должна быть инициализирована и код клавиши должен быть декодирован.

02F1 K-NEW

LD E,A

Код записывается в регистр E и B

LD	(HL), A	KSTATE/4, Счетчик на пять для этого
INC	HL	набора устанавливается в "5"
LD	(HL), +05	
INR	HL	Третья системная переменная набора
LD	A, (REPDEL)	содержит значение REPDEL
LD	(HL), A	(обычно 0,7 сек)
INC	HL	точка В KSTATE 3/7

Декодирование "Основного кода" зависит от текущего состояния режима, бита 3 флагов и "Бита регистра".

LD	C, (MODE)	Загрузка режима
LD	D, (FLAGS)	Загрузка флагов
PUSH	HL	Спасение указателя на время, когда
CALL	0333, K-DECODE	"Основной код" декодируется.
POP	HL	
LD	(HL), A	Значение окончательного кода
		спасается в KSTATE 3/7, откуда он
		выбирается в случае повторения.

Следующие три строки инструкции являются общими для управления и для "Новой клавиши", и для "Повторной клавиши".

0308 K-END

LD	(LAST-K), A	Вводится окончательное значение
SET	5, (FLAGS)	кода в LAST-K и сигнал "Новая
		клавиша"
RET		Окончательный возврат

Подпрограмма "Повторный набор"

(REPEATING KEY SUBROUTINE)

Клавиша будет введена повторно в первом случае после периода задержки (REPDEL) • (Обычно 0,7 сек) и в последующем случае после периода задержки (Обычно 0,1 сек).

0310 K-REPEAT

INC	HL	установить "счетчик пяти вызовов"
LD	(HL), +05	В "5"
INC	HL	Адресовать третью системную

DEC	(HL)	переменную REPDEL/REPPER значение и уменьшить его.
RET	NZ	Выход из подпрограммы KEYBOARD, если период задержки не окончился
LD	A,(REPPER)	Однако, как только это произойдет,
LD	(HL),A	период задержки для следующего повтора Д.В. установлен REPPER
INC	HL	повторение принимается, как оконча-
LD	A,(HL)	тельное значение кола.
JR	O308, K-END	извлекается из KSTATE 3/7 и записывается в K-END

Подпрограмма "K-TEST"

Значение клавиши проверяется, и осуществляется возврат, если "Нет нажатия" или "Только регистр", в противном случае определяется "Основной код" для этой клавиши.

031E K-TEST

LD	B,D	Сканировать байт сдвига (регистра)
LD	D,+00	Очистить D-регистр для будущей работы.
LD	A,E	Запести номер клавиши
CP	+27	теперь повторить, если была только
RET	NC	клавиша "CAPS SHIFT" или не было нажатия
CP	+18	Переход вперед, если только не был
JR	NZ, 032C, K-MAIN	SYMBOL SHIFT клавиши "E"
BIT	7,B	Однако, прием SYMBOL SHIFT и другой
RET	NZ	клавиши: возврат только с SYMBOL SHIFT
"Основной код"		находится индексированием основной
таблицы клавиши		
LD	HL,-0205	Базовый адрес таблицы
ADD	HL, DE	Индекс в таблице и вызов "Основного
LD	A,(HL)	кода"
SCF		Установить сигнал "Допустимый
RET		ввод" перед возвратом.

Подпрограмма "Декодирование клавиатуры"
(THE KEYBOARD DECODING SUBROUTINE)

При вызове подпрограммы в регистре "Е" находится "Основной код", значение ФЛАГОВ в регистре "D", значение РЕЖИМА в регистре "С" и "БАЙТ РЕГИСТРА" в регистре "В". Рассматривая эти четыре значения и обращаясь, при необходимости, к шести таблицам кодов, клавиш получают "Окончательный код". Он возвращается в регистре "А".

0333 K-DECODE

LD A,E Копирование "Основного кода"
 CP +3 Переход вперед, если рассматривается
 JR C,0367,K-DIGIT цифровая клавиша, а также SPACE,
 ENTER и оба сдвига (регистра)
 DEC C Уменьшение значения режима
 JR M,034F,K-KLC-LET Переход вперед, как требуется,
 JR Z,0341,K-E-LET для режимов "К", "L", "С", "Е"

Остается только "Графический режим" и "Окончательный код" для буквенных клавиш в графическом режиме вычисляется из "Основного кода".

ADD A,+4F Добавить смещение для получения
 требуемого адреса памяти
 RET возврат с "Окончательным кодом"

Буквенные клавиши в расширенном режиме рассматриваются ниже.

0341 K-E-LET

LD HL,+01EB Базовый адрес для таблицы "2".
 INC B Переход вперед для использования
 JR Z,034A,K-LOOK-UP этой таблицы, если клавиша РЕ-
 ГИСТРА не нажата
 LD HL,+0205 в противном случае использовать
 базовый адрес для таблицы "3".

Все таблицы клавиши "В-Ф" обслуживаются нижеследующей циклической программой, во всех случаях "Окончательный код" определяется и возвращается.

034A K-LOOK-UP

LD D,+00 Очистить D-регистр
 ADD HL,DE индексировать адрес требуемой

таблицы
и извлечь "Окончательный код",
затем вернуться.

LD A,(HL)
RET

Теперь рассмотрим клавиши букв в режимах "К", "L" или "С", но, во-первых, имеем дело со специальными кодами SIMBOL SHIFT (Символьного регистра).

034F K-KLC-LET

LD HL,+0229 Базовый адрес для таблицы 5
BIT 0,B Вернуться назад, если используется
JR Z,034A,K-LOOK-UP SIMBOL SHIFT клавиша и
буквенная клавиша.

BIT 3,D Переход вперед, если в текущий момент
JR Z,0364,K-TOKENS времени работаем в "К" режиме.
BIT 3,(FLAGS2) Если CAPS LOCK установлен, то
RET NZ возврат с "Основным кодом".
INC B Таким же образом идет возврат, если
RET NZ нажат CAPS SHIFT.
ADD A,20 Однако, в последнем случае необходимо
RET затем добавить +20 к "Основному коду"
для получения скорректированного
"Окончательного кода".

Значение "Окончательного кода" для знаков
определяется добавлением +5 к "Основному коду".

0364 K-TOKENS

ADD A,+A5 Добавить необходимое смещение и
RET вернуться

Теперь рассмотрим цифровые клавиши, SPACE, ENTER и оба регистра

0367 K-DIGIT

CP +30 Продолжить только с цифровыми клавишами,
RET C т. е. возврат с SPACE (+20), ENTER (+0D) и обоими регистрами (+0E).

DEC C Теперь делятся цифровые клавиши на 3
группы в соответствии с режимом.

JP M,039D,K-KLC-DGT Переход в режимы "К", "L" и "С",

JR NZ,0389,K-GRA-DGT а также в режим "G", продолжение с режимом "E".

LD HL,-0254 Базовый адрес таблицы "6".
BIT .5,B Используем эту таблицу для
SIMBOL SHIFT

JR Z,034A,K-LOOK-UP и цифровой клавиши в расширенном режиме.

CP +38 Переход вперед с цифровой клавишей "8"

JR NC,0382,K-8-&-9 и "9"

Цифровые клавиши от "0" до "7" в расширенном режиме дают либо "Код цвета бумаги" либо "Код цвета чернил" в зависимости от используемого CAPS SHIFT.

SUB +20 Уменьшить диапазон от +30 до +37,
получая диапазон от +10 до +17

INC B Возврат с "Кодом цвета бумаги", если

RET Z CAPS SHIFT не использован.

ADD A,+08 Но если используется, то диапазон будет
RET от +18 до +1F, фиксируя "Код цвета чернил"

Цифровые клавиши "8" и "9" определяют коды "Яркости" и "Мигания".

0382 K-8-&-9

SUB +36 +38& +39 переводятся в 02& +03

INC B возврат с этими кодами, если CAPS SHIFT

RET Z не используется (это коды "Яркости"),

ADD A,+FE вычитание "2", если CAPS SHIFT
использован

RET дается +00& +01 (как коды "Мигания")

Цифровые клавиши в графическом режиме используются для получения блока графических символов (+80 до +8F), графического кода (+0F) и кода стирания (+0C).

LD HL,+0230 Базовый адрес таблицы "4"

CP +39 Используя эту таблицу непосредственно

JR Z,034A,K-LOOK-UP для цифровой клавиши "9",
которая дает графику

CP +30 и цифровую клавишу "0", которая

JR Z,034A,K-LOOK-UP дает стирание.

AND	+07	Для клавиши от "1" до "8" делаем диапазон
ADD	A,+80	от +80 до 87
INC	B	Возврат со значением из указанного
RET	Z	диапазона, если одна из клавиш регистра не была нажата.
XOR	+0F	Но если клавиша регистра нажата, то
RET		взять диапазон от +88 до +8F.

В конце рассмотрим работу цифр, клавиш в режимах "K","L","C".

039D K-KLC-DGT

INC	B	Прямой возврат, если ни одна из клавиш
RET	Z	регистра не нажата (Окончательный код от +30 до +39).
BIT	5,B	Использовать таблицу "4", если была
LD	HL,+0230	также нажата CAPS SHIFT
JR	NZ,034A, K-LOOK-UP	

Теперь могут быть найдены коды для различных цифровых клавиш при нажатой клавише регистра SYMBOL SHIFT.

SUB	+10	Уменьшить диапазон для получения диапазона от +20 до +29.
CP	+22	Отделить символ "@" от всех остальных
JR	Z,0382,K-@-CHAR	
CP	+20	символ "-" также должен быть выделен.
RET	NZ	Возврат с "Окончательным кодом" +21 и от +23 до +29.
LD	A,+5F	Присвоить "-" код +5F
RET		
03B2	K-@-CHAR	
LD	A,+40	
RET		

Звуковые программы (THE LOUDSPEAKER ROUTINES)

Две подпрограммы, описанные в этой части, представляют собой подпрограмму BEEPER, которая управляет динамиком, и управляющую программу BEEP.

Динамик активируется путем перевода бита D4 В"0" при действии команды OUT, использующей порт "254", если разряд D4 находится в состоянии "1" (высоком состоянии), то динамик не работает, следовательно, "Сигнал" может быть получен путем изменения уровня разряда D4 порта "254".

Теперь рассмотрим ноту "СИ - средняя", которая имеет частоту 261.63 ГЦ, соответственно, для получения этой ноты динамик должен быть попеременно включен и выключен каждый 1/523.26 тысячную секунды, в "спектрume" системные такты равны 3,5 мГЦ и нота "си-средняя" потребует, чтобы необходимая команда вывода (OUT) выполнялась каждые 6,689 тактов, последнее значение, немного уменьшенное для получения необходимой высоты звука, представляет собой "длину времени цикла" в подпрограмме BEEPER.

Подпрограмма "BEEPER"

При вызове этой подпрограммы в регистровой паре DE хранятся значения "F", "T", означающие, что нота данной частоты "F" должна звучать в течение "T" секунда, и в регистровой паре HL хранится величина, равная числу тактов в цикле, деленному на 4, т. е. для ноты "си-средняя" продолжительностью 1 сек DE содержит +0105 (прерывание с частотой 261.63 ГЦ) и HL содержит +066A (Получено из 6.689/4-30.125)

03B5 BEEPER

DI		Запретить прерывание на время действия "Сигнала".
LD	A,L	Временное спасение L.
SRL	L	каждая "1" в регистре L является
SRL	L	счетчиком "4" тактов.
CPL		
AND	+05	Возврат к начальному заключению регис-
LD	C,A	тра L и определение сколько было по-
LD	B,+00	теряно при получении INT(L/4)
LD	IX,+03D1	Базовый адрес времени цикла.
ADD	IX,BC	Изменить длину времени цикла, исполь-
		зовать предыдущую стартовую точку для
		каждой "1", вычитаемой при получении
		INT(L/4)

LD	F,(BORDER)	Вызывает текущий граничный цвет
AND	+38	и перезаписывает его в биты 2,1 и 0
RRCA		биты регистра A
RRCA		
OR	+08	Обеспечивает отключение выхода MIC.

Теперь входим в цикл генерации звука, регистр HL содержит L "Длину времени цикла" с "16" тактами для каждой "1" в регистре L и "1024" тактами для каждой "1" в регистре H.

03D1	BE-IX +3	
NOP		Добавить "4" такта для каждой
03D2	BE-IX +2	
NOP		предыдущей точки входа, которая
03D3	BE-IX +1	
NOP		используется
03D4	BE-IX +0	
INC	B	Значение регистров BC получено из HL
INC	C	регистров - см. ниже.
03D6	BE-HL-LP	
DEC	C	"Время цикла", т.е. "BC" "4" такта.
JR	NZ,03D6,BE-HL-LP	
LD	C,+3F	(Заметим, что точка половины периода
DEC	B	C, будет равна "L+1")
JP	NZ,03D6,BE-H&L-LP	

Динамик сейчас попеременно включается и выключается.

XOR	+10	Установить бит "4"
OUT	(+FE),A	Осуществляет операцию вывода, оставляя границу неизменной
LD	B,H	Переустановка регистра B.
LD	C,A	Спасти регистр A.
BIT	4,A	Переход, если эта точка полупериода
JR	NZ,03F2,BE-AGA1N	

После первого цикла регистровая пара DE проверяется.

LD	A,D	Переход вперед, если последний полный проход уже сделан.
OR	E	

JR Z,03F6,BE-END Вызов ранее спасенных величин.
 LD A,C Переустановка регистра C.
 LD C,L
 DEC DE Уменьшить счетчик проходов.
 JP (IX) Возврат назад к стартовой ячейке цикла.
 Устанавливаются параметры для второго полупериода.
 03F2 BE-AGAIN
 LD C,L Переустановка регистра C
 INC C Добавить 1/16 тактов, т. к. эта часть короче.
 JP (IX) Возврат назад.
 По завершении "Сигнала" маскируемого прерывания Д.Б. разрешены.
 03F6 BE-END
 EI Разрешение прерываний
 RET Окончательный возврат

Управляющая программа "BEEP"

Это подпрограмма вызывается с двумя числами в стеке калькулятора, верхнее число фиксирует "Высоту" ноты и нижнее число фиксирует "Длительность".

03F8 BEEP

RST 0028, FP-CALC Калькулятор с плавающей запятой используется для манипуляции с двумя величинами - "Т" и "Р" (время и высота).
 DEFB +31, DUPLICAT "Т","Р","Р".
 DEFB +27, INT "Т","Р","I" (где I=INT P)
 DEFB +C0,ST-MAM-0 "Т","Р","I"(MAM 0 содержит 1)
 DEFB +03, CUBTRACT "Т","Р"(где Р является функциональной частью от Р)
 DEFB +34, STK-DATA Стек-десятичное значение "К".
 DEFB +EC, EXPONENT+7C 0,0577622606, который является остатком
 DEFB +6C, +98, +1F, +F5 после 12 2 -1.

DEFB	+0,4, MULTIPLY	"T",PK
DEFB	+A1, STK-ONE	"T",PK,1
DEFB	+0F, ADDITION	"T",PK,+1
DEFB	+38, END-CALC	

Теперь выполним несколько проверок I, целой части "Высоты"

LD	HL, +5C92	
LD	A, (HL)	
AND	A	
JR	NZ, 046C, REPORT-8	Проверка 1
INC	HL	
LD	C, (HL)	
INC	HL	
LD	B, (HL)	
LD	A, B	
RLA		
SBC	A, A	
CP	C	
JR	NZ, 046C, REPORT-8	
INC	HL	
CP	(HL)	
JR	NZ, 046C, REPORT-8	
LD	A, B	
ADD	A, +3C	
JR	F, 0425, BE-I-OK	
JR	PO, 046C, REPORT-B	

Заметим: диапазон +70 до +127 будет отброшен позже, теперь М.Б. определенная скорректированная частота для "Высоты" 1.

0425 BE-I-OK

LD B, +FA

0427 PE-OCTAUT

INC B

SUP +0C

JR NC, 0427, BE-OCTAUE Коррекция 1

ADD A,+0C
 PUSH BC
 LD HL,+046E
 CALL 3406,LOC-MEM
 CALL 3384,STACK-NUM

Значение "ATN"

Вводится в стек калькулятора,
 далее учитывается дробная
 часть "Высоты".

RST 0028,FR-CALC
 DEFB +04,MULTIRLY
 DEFB +38,AND-CALC

Окончательная частота F определяется путем
 модификации "Последнего значения" в соответствии с
 номером октавы.

POP AF Извлечь номер октавы
 ADD A,(HL) Умножить "Последнее значение" на 2
 LD (LD),A для получения номера октавы.
 RST 0028,FP-CALC "T","F"
 DEFB +CO,ST-MEM-0 Частота дается для момента
 в MEM-0.

DEFB +02,DELETE

Теперь внимание переключается на "Время".

DEFB +31 DUPLICAT "T","T"
 DEFB +38,END-CALC
 CALL 1E94, FIND-INT1 Значение "INT T" должно быть в
 CP +08 диапазоне +00 до +0A
 JR NC,046C,REPORT-B

Число общих циклов в "Сигнале" задается величинами
 "F","T" и таким образом это значение теперь определяется.

RST 0028,FR-CALC "T"
 DEFB +E0,GET-MEM-0 "T","F"
 DEFB +04,MULTIPLY

Результат остается в стеке калькулятора на время
 вычисления длины "Времени цикла", необходимого для
 "Сигнала".

DEFB +E0, GET-MEM-0
 DEFB +34,STK-DATA


```

DEFB +80,FOUR BYTES
DEFB +43,EXPONENT+93
DEFB +55,9F,+30,(+00)
DEFB +01,EXCHANGE
DEFB +05
DEFB +34,STK-DATA
DEFB +35,EXPONENT +85
DEFB +71,(+00,+00,+00)
DEFB +03
DEFB +38,END-CALC

```

Заметим, что величина "437,500/F" дает длину "Полупериода" ноты. Это значение теперь можно перенести в необходимые регистры.

```

CALL 1E99,FIND-INT2    Значение "Времени цикла"
                        упаковывается в регистровую
PUSH BC                пару BC и спасается.

```

Заметим, что если величина цикла слишком большая, то будет зафиксирована ошибка (возвращаемая как ERROR-1), посредством этого исключается значение "Высоты" в диапазоне +70 до +127.

```

CALL 1E99, FIND-INT 2  Значение "F", "T" упаковывается в
                        регистровой паре BC
POP    HL              Переписать "Время цикла" в
                        регистровую пару HL
LD     D,B             Переписать значение "F","T" в
LD     E,C             регистровую пару DE.

```

Однако, перед выводом сигнала проверим значение "F","T"

```

LD     A,D             Возврат, если "F" "T" даст результат
OR     E               "Не требуется циклов".
RET    Z
DEC    DE              Уменьшить число циклов и возврат
JR     03B5,BEEPER     в подпрограмму BEEPER (делая по
                        крайней мере один проход)

```

Сообщение В - число вышло за диапазон.

046C REPORT-B

RST 0008,ERROR-1 Вызов программы обработки ошибки
 EFB +0A

Таблица "Полутон" (THE "SEMI-TON")

Эта таблица содержит частоты двенадцати полутонов в октаве,

		Частота, гц	Нота
046E DEFB	+89,+02,+D0,+12,+86	261.65	C
DEFB	+89,+0A,+97,+60,+75	277.18	C#
DEFB	+89,+12,+D5,+17,+1F	293.66	D
DEFB	+89,+1B,+90,+41,+02	311.13	D#
DEFB	+89,+24,+D0,+53,+CA	329.63	E
DEFB	+89,+2E,+9D,+36,+B1	349.23	F
DEFB	+89,+38,+FF,+49,+3E	369.99	F#
DEFB	+89,+43,+FF,+6A,+73	392	G
DEFB	+89,+4F,+A7,+00,+54	415.30	G#
DEFB	+89,+5C,+00,+00,+00	440	A
DEFB	+89,+69,+14,+F6,+24	466.16	A#
DEFB	+89,+76,+F1,+10,+05	493.88	B

Подпрограмма "Имя программы"(ZX81)

Следующая подпрограмма применяется в ZX81 и не была изменена для "спектрума".

04AA DEFB +CD,+FB,+24,+3A
 DEFB +3B,+5C,+87,+FA
 DEFB +8A,+1C,+E1,+D0
 DEFB +E5,+CD,+F1,+2B
 DEFB +62,+6B,+0D,+F8
 DEFB +09,+CB,+FE,+C9

Программа управления магнитофоном THE CASSTTE HANDLING ROUTINE

16 К байтовая программа-монитор имеет 'широкий ряд программ для управления интерфейсом магнитофона, в результате эти программы формируют управляющие процедуры: спасение (SAVE), проверка (VERIFY) и склейка (MERGE).

Точкой входа в программу является SAVE-ETC(0605), однако, перед этим, подпрограммы занимаются текущим спасением и загрузкой (или проверкой) байтов.

В любом случае байты, управляющие этими подпрограммами, описываются регистровой парой DE, содержащей "Длину" блока, регистровой парой, содержащей "Базовый адрес" и регистром A, содержащим +00 для управляющего блока или +FF для блока программ/данных.

Подпрограмма SA-BYTES

Эта подпрограмма вызывается для спасения управляющей информации (C 098A) и затем текущего блока программ/данных (C 099E).

04C2 SA-BYTES

LD	HL,+053F	Предварительная загрузка машинного
PUSH	HL	стека адресом-SA/LD-RET (Возврат)
LD	HL,+1F80	Эта константа даст лидеру (драйверу)
		около 5 сек для "Заголовок"
BIT	7,A	переход вперед, если спасается
JR	Z,04D0,SA-FLAG	"Заголовок"
LD	HL,+0C98	Эта константа даст лидеру около 2 сек
		для блока программ/данных

04D0 SA-FLAG

EX	AF,A'F'	спасается флаг
INC	DE	"Длина" увеличивается и "Базовый
DEC	IX	адрес" уменьшается с учетом флага.
DI		Запрещается маскируемое прерывание
		на время спасения (записи).
LD	A,+02	Сигнал "MIC ON" (Включение
		микрофона) и бордюр должны быть
RED		(Красный)
LD	B,A	Запись значения в "B".

Теперь начинается цикл для генерации импульсов лидера, длительность каждого из импульсов - "MIC ON" (Микрофон включен) и "MIC OFF" (Микрофон выключен)-2, 168

тактов, цвет бордюра меняется от RED к CYAN при каждом переключении (По фронту).

Заметим: фронт означает переход от состояния "вкл" к состоянию "выкл" или наоборот.

04DB SA-LEADER

DJNZ	04D8,SA-LEADER	Основной период
OUT	(+FE),A	Микрофон вкл./выкл., бордюр
XOR	+0F	RED/CYAN на каждом шаге.
LD	B,+A4	Основная временная константа
		(период)
DEC	L	Уменьшить младший счетчик.
JR	NZ,04D8,SA-LEADER	Возврат назад для следующего импульса.
DEC	E	Сделать проверку для следующего прохода (уменьшить на 13 тактов)
DEC	H	Уменьшить старший счетчик
JP	P,04D8,SA-LEADER	Возврат для следующего импульса вплоть до завершения лидера.

Теперь посылается синхрои́мпульс.

LD B,+2F

04EA SA-SYNC-1

DJNZ	04EA,SA-SYNC-1	Отключение микрофона на 667 тактов от "Вывода до вывода" (OUT TO OUT)
OUT	(+FE),A	Включение микрофона и RED (Красный)
LD	A,+0D	сигнал "выкл.микрофон и CYAN"
LD	B,+37	Микрофон включен на 735 тактов от

04F2 SA-SYNC-2

DJNZ	04F2,SA-SYNC-2	вывода до вывода.
OUT	(+FE),A	Теперь микрофон выключен и бордюр установлен CYAN.

Флаг "Заголовок или программа/данные" будет первым байтом, который должен спастись.

LD	BC,+3B0E	+3B является временной константой;
		+0E сигнализирует "Микрофон"

выключен и желтый".

EX AF,A'F' Извлечь флаг и поместить его в
LD L,A регистр L для "Пересылки".
JP 0507,SA-START Переход вперед в цикла спасения.

Начинается цикл спасения байта, флаг является первым байтом, который должен быть спасен: за ним следует текущий байт и последним байтом посылается байт паритета, значение которого зависит от всех ранее записанных байтов.

04FE SA-LOOP

LD A,D Проверить счетчик "Длины" и переход,
OR E когда он становится равным нулю.

JR Z,050E,SA-PARITY

LD L,(IX+00) Извлечь следующий байт, который
должен быть спасен.

0505 SA-LOOP-P

LD A,H Вызвать текущий "Паритет".

XOR L Включить очередной байт.

0507 SA-START

LD H,A Сохранить "Паритет"; заметим, что
значение "Флага" инициализирует
"паритет".

LD A,+01 Сигнал "вкл.мик. и голубой".

SCF Установить флаг переноса, он будет
действовать, как маркер для 8 битов
байта

JP 0525,SA-8-BITS Переход вперед.

После посылки байта паритета он переписывается в регистр L для спасения.

050E SA-PARITY

LD L,H Получение окончательного значения
"Паритета"

JP 0505,SA-LOOP-P Возврат назад.

Следующий внутренний цикл генерирует импульсы, цикл начинается в SA-BIT-1 с фиксации типа бита, который должен быть спасен, посредством флага переноса для каждого бита делается два прохода цикла, тем самым

осуществляя "Отключение импульса" и "Включение импульса". Импульс для восстановления байта несколько короче, 855 тактов.

0511 SA-BIT-2

LD A,C

Перейти сюда на втором проходе и получить "Мик.выкл. и желтый".

BIT 7,B

Установить флаг в "0" для обозначения второго прохода.

0514 SA-BIT-1

DJNZ 0514,SA-BIT-1

Основное время цикла (таймер); всегда равное 801 такту на втором проходе.

JR NC,051C,SA-OUT

Переход по короткой ветке, если спасается "0".

LD B,+42

Однако, если спасается "1", то добавляется

051A SA-SET

DJNZ 051A,SA-SET

855 тактов.

051C SA-OUT

OUT (+FE),A

На первом проходе "Мик.вкл. и голубой" и на втором проходе "Мик.выкл. и желтый"

LD B,+3E

установить константу времени для второго прохода.

JR NZ,0511,SA-BIT-2

Возврат назад в конец первого прохода;

DEC B

в противном случае генерация 13 тактов

XOR A

Очистить флаг переноса и заслать в A +01 (Мик.вкл. и голубой) перед "8-битовым циклом".

INC A

"8-битовый цикл" начинается с полным байтом в регистре L и после каждого "спасенного" бита, до тех пор, пока "Маркер" циклов во флаге переноса не зафиксирует "Пустой" L-регистр (L-регистр - пустой).

0525 SA-8-BITS

RL L

Сдвинуть седьмой бит в перенос и,

соответственно, "Маркер" сдвинуть влево.

JP	NZ,0514,SA-BIT-1	Спасти бит до тех пор, пока не окончится байт.
DEC	DE	Уменьшить "Счетчик"
INC	IX	Увеличить "Базовый адрес".
LD	B,+31	Установить константу времени (период) для первого бита следующего байта.
LD	A,+7F	Возврат (K SA/LD-RET), если нажата клавиша BREAK.
IN	A,(+FE)	
RRA		
RET	NC	
LD	A,D	В противном случае, проверка "Счетчика"
INC	A	и возврат назад даже, если он достиг "0"
JP	NZ,04FE,SA-LOOP	(т. к. нужно послать еще байт "паритета")
LD	B,+3B	выход, когда "Счетчик" достигает
053C	SA-DELAY	значения "FFFF, но сначала нужно
DJNZ	053C,SA-DELAY	дать короткую задержку.
RET		

Заметим: нулевой бит должен записываться сигналом "Мик.выкл." в течение 855 тактов с последующим сигналом "Мик.вкл." в течение 855 тактов, единичный бит требует сигналов двойной длительности. Заметим также, что нет паузы между синхронизатором и первым битом флага или между байтами.

Подпрограмма "SA-LD-RET"

Подпрограмма является общей как для спасения, так и для загрузки, бордюр устанавливается в свой начальный цвет и клавиша BREAK тестируется в последний раз.

053F SA/LD-RET

PUSH	AF	Спасти флаг переноса. (Он восстанавливается после ошибки загрузки.)
------	----	---

LD A,(BORDER) Извлечь начальный цвет бордюра из
 AND +39 системной переменной.
 RRCA Сдвинуть цвет в биты 2,1 и 0
 RRCA
 OUT (+FE),A Установить бордюр в начальный цвет.
 LD A,+7F Прочитать клавишу "BREAK"
 IN A,(+FE) в последний раз.
 RRA
 EI Разрешить маскируемые прерывания.
 JR C,0554,SA/LD-END Переход до тех пор, пока не будет
 сделан сброс (BREAK)
 Сообщение D-BREAK-CONT повторяется

0552 REPORT

RST 0008,ERROR-1 Вызов программы обработки ошибки.
 DEFB +0C

Продолжение:

0554 SA/LD-END

POP AF Восстановить флаг переноса
 RET Возврат в вызывающую программу.

Подпрограмма "LD-BYTES"

Подпрограмма вызывается при загрузке управляющей информации (из точки 076E) и далее при загрузке и проверке, текущего блока данных (из точки 0802).

0556 LD-BYTES

INC D Сброс флага нуля (D не может хранить +FF)
 EX AF,A'F' Регистр A содержит +00 для заголовка и
 +FF для блока данных.
 Флаг переноса сбрасывается при проверке и
 устанавливается при загрузке
 DEC D Возврат в начальное значение.
 DI Запрет маскируемых прерываний.
 LD A,+0F Бордюр делается белым (WHITE).
 OUT (+FE),A Занесение в стек адреса возврата
 LD HL,+53F

PUSH	HL	SA/LD-RET.
IN	A,(+FE)	Чтение порта "254"
RRA		Сдвиг полученного байта, но выделяется
AND	+20	только бит EAR.
OR	+02	Сигнал "красного" бордюра,
LD	C,A	Сохранить значение в регистре C (+22 для "выкл." и "+02" для "вкл.").
CP	A	Установить флаг нуля.

Первая фаза чтения ленты включает в себя выделение пульсирующего сигнала (т.е. фронт вкл/выкл или выкл/вкл).

056B LD/BREAK

RET NZ Возврат, если нажат ключ "BREAK"

056C LD-START

CALL 05E7,LD-EDGE-1 Возврат со сбросом флага переноса, если не обнаружен фронт в течение 14000 тактов, но, если будет обнаружен фронт, то бордюр устанавливается CYAN.

Следующий шаг включает в себя ожидание, в течение которого проверяется, продолжается ли пульсация.

LD HL,0415 Длительность этого периода ожидания

DJNZ 0574,LD-WAIT будет равна почти 1 сек.

DEC HL

LD A,H

OR L

JR NZ,0574,LD-WAIT

CALL 05E3,LD-EDGE-2 Продолжать только, если обнару-

JR NC,056B,LD-BREAK жено два фронта внутри временного интервала.

Теперь принимаем только "Сигнал лидера".

0530 LD-LEADER

LD B,+9C Временная константа.

CALL 05E3,LD-EDGE-2 Продолжать, если обнаружено

JR NC,056B,LD-BREAK два фронта внутри временного интервала.

LD A,+C6 Фронты должны определять на
 CP B интервале около 3000 тактов один от
 JR NC,056C,LD-START другого
 INC H Считать (Нарастивать) число парных
 JR NZ,0580,LD-LEADER фронтов в регистре H до тех
 пор, пока не будет обнаружено
 256 пар.

После получения "Лидера" выделим, синхроимпульс.

058F LD-SYNC

LD B,+C9 Временная константа (период).
 CALL 05E7,LD-EDGE-1 Рассматривается каждый фронт
 JR NC,056B,LD-BREAK до тех пор, пока два фронта
 LD A,B не закончатся вместе, это могут
 CP +D4 быть передний и задний фронта
 JR NC,058F,LD-SYNC синхроимпульса "OFF" (выкл.)
 CALL 05E7,LD-EDGE-1 Необходимо найти задний фронт
 импульса "ON" (вкл.)

RET NC Возврат по нулю флага переноса

Байты заголовка или программ/данных теперь могут
 быть загружены или проверены, но первым байтом является
 байт флага.

LD A,C Цвет бордюра с этого момента Д.Б.
 XOR +0,3 голубой и желтый.

LD C,A

LD H,+00

LD B,+B0

Установить байт паритета в нуль.

Установить временную константу
 для байта флагов.

JR 05C8, LD-MARKER Переход вперед в цикле загрузки
 байта.

Цикл загрузки байта используется для извлечения
 байтов, флаговый байт идет первым, затем следуют байты
 данных и последний байт паритета.

05A9 LD-LOOP

EX A,F,A',F'

Вызвать флаги.

JR NZ,0583,LD-FLAG

Переход вперед только, когда

		обрабатывается первый байт.
JR	NC,058D,LD-VERIFY	Переход вперед, если проверяется лента.
LD	(IX+00),L	Сделать очередную загрузку, когда потребуется.
JR	05C2,LD-NEXT	Переход вперед для загрузки следующего байта.
05B3	LD-FLAG	
RL	C	Временно сохранить флаг переноса.
XOR	L	Возврат, если флаг типа не соответствует первому байту на ленте (флаг переноса сброшен)
RET	NZ	Восстановление флага переноса.
LD	A,C	
RRA		
LD	C,A	
INC	DE	Увеличить счетчик, чтобы компенсировать его уменьшение после перехода.
JR	05C4,LD-DEC	
Если осуществляется проверка блока данных, то вновь введенный байт проверяется на соответствие эталонному байту.		
05BD	LD-VERIFY	
LD	A,(IX+00)	Вызвать эталонный байт.
XOR	L	Сравнить его с введенным байтом.
RET	NZ	Возврат, если "Нет равенства" (Флаг переноса сбросить).
Теперь новый байт с ленты.		
05C2	LD-NEXT	
INC	IX	Увеличить "Адрес назначения".
05C4	LD-DEC	
DEC	DE	Уменьшить "Счетчик"
EX	AF,A'F'	Спасти флаги.
LD	B,+B2	Установить временную константу.
05C8	LD-MARKER	
LD	L,+01	Очистить "Объектный регистр" за исключением "Маркерного" бита.

Цикл "LD-8-BITS" используется для формирования байта в регистре L.

05CA LD-8-BITS

CALL 05E3,LD-EDGE-2 Найти длину импульсов "вкл" и "выкл" следующего бита.

RET NC Возврат, если время периода закончилось (Флаг переноса сброшен).

LD A,+CB Сравнить длину С 2400 тактами; сбросить флаг переноса для "0" и установить его для "1".

CP B Добавить новый бит в регистр L.

RL L

LD B,+B0 Установить временную константу для следующего бита.

JP NC,05CA,LD-8-DITS Возврат назад, пока не будут извлечены все биты.

Байт паритета будет модифицироваться при каждом новом вводе байта.

LD A,H Вызвать байт паритета и учесть новый байт.

XOR L

LD H,A Спасти байт паритета.

Повторять цикл до обнаружения "0" в счетчике, в этой точке байт паритета должен быть равен нулю.

LD A,D Продолжить цикл, если регистровая пара DE не содержит "0".

OR E

JR NZ,05A9,LD-LOOP

LD A,H Извлечь байт паритета.

CP +01

RET Возврат с установкой бита переноса, если байт паритета равен "0", (Флаг переноса сбрасывается, если есть ошибка).

Подпрограммы "LD-EDGE-2" и "LD-EDGE-1"

Эти две подпрограммы формируют наиболее важную часть операций загрузка/проверка, при входе в подпрограмму в регистре В находится постоянная времени, а в регистре С

предварительно установленное значение цвета бордюра и тип фронта.

Возврат из подпрограммы осуществляется с установкой флага переноса, если обнаружено требуемое число "Фронтов" в заданный промежуток времени; и изменение значения регистра В показывает, сколько времени прошло до обнаружения "Фронта" (Фронтов), флаг переноса сбрасывается, если обнаружена ошибка, флаг нуля сигнализирует о нажатии клавиши "BREAK" или об установке сигнала "Время истекло".

Вход в подпрограмму LD-EDGE-2 используется, когда требуется определить длину импульса и LD-EDGE-1 используется для определения времени до следующего фронта.

05E3 LD-EDGE-2

CALL 05E7,LD-EDGE-1 При работе дважды выдает LD-EDGE-1;

RET NC Возвращение, если обнаружена ошибка.

05E7 LD EDGE-1

LD A1+16 Ждать 358 тактов перед входом в цикл проверки.

05E9 LD-DELAY

DEC A

JR NZ 05E7,LD-DELAY

AND A

Теперь начинается цикл проверки, значение регистра В увеличивается на каждом проходе, "Конец интервала времени" получается, когда регистр В достигает "0", 05ED LD-SAMPLE.

INC B Счетчик циклов.

RET I Возврат со сбросом переноса и установкой "0", если "Конец интервала".

LD A1+7F Чтение из порта +7FFE,

IN A1(+FE) т.е. BREAK AND EAR.

RRA Сдвиг байта.

RET NC Возврат со сбросом переноса и сбросом флага нуля, если была

XOR	C	нажата клавиша "BREAK"
		Теперь проверка байта
AND	+20	"Последнего типа фронта";
JR	Z,05ED,LD-SAMPLE	возврат назад до тех пор,
		пока он не изменится.
Обнаружен новый "Фронт" в течение допустимого интервала поиска, тогда изменяем цвет бордюра и устанавливаем флаг переноса.		
LD	A,C	Изменить "Тип последнего
C	PL	фронта" и цвет бордюра.
LD	C,A	
AND	+0,7	Сохранить только цвет бордюра.
OR	+0,8	Сигнал "Мик.выкл."
OUT	(+FE),A	Изменить цвет бордюра
		(RED/CYAN или голубой/желтый),
SCF		Сигнал обнаружения перед
		возвратом.
RET		

Заметим: подпрограмма LD-EDGE-1 занимает 465 тактов, плюс дополнительные 38 тактов для каждого безуспешного цикла проверки, например, при ожиданий синхроимпульса (см.LD-CYNC В 050F) принимаются во внимание десять дополнительных проходов в цикле проверки, поиск следующего фронта тем самым осуществляется внутри, приблизительно 1100 тактов (465+10x58+Вход), это дает обнаружение для синхроимпульса "выкл.", который приходит после длинных "лидирующих импульсов".

Управляющие программы "Спасение, загрузка, проверка и склейка"

("THE SAVE, LOAD, VERIFY & MERGE" COMMAND ROUTINES)

Для всех четырех команд используется точка входа SAVE-ETS. Значение хранится в T-ADDR, однако различается для всех команд. Первая часть программы связана со структурой "Управляющей информации" в рабочей области.

0605 SAVE-ETS

POP AF

Уменьшить адрес -SCAN-LOOP.

LD	(T-ADDR-LO)	Уменьшить T-ADDR-LO на +E0,
SUB	+E0	задавая +00 для спасения, +01 для
LD	(T-ADDR-LO),A	загрузки, +02 для проверки, +03
		для склейки.
CALL	1CBC,EXPT-EXP	Переместить параметры "Имя" в
		стек калькулятора.
CALL	2530,SYNTAX-Z	Переход вперед, если проверяется
JR	Z,0652,SA-DATA	синтаксис.
LD	BC,+0011	Предоставить семнадцать ячеек
LD	A,(T-ADDR-LO)	для управления спасением или
AND	A	тринадцать для других команд.
JR	Z,0621,SA-SPACE	
LD	C,+22	
	0621 SA-SPACE	
RST	0030,BC-SPACE	Резервируется рабочая область.
PUSH	DE	Копирование начального адреса в
POP	IX	регистровую пару IX.
LD	B,+0B	Имя программы может занимать
LD	A,+20	до десяти символов и вводится в
	0629 SA-BLANK	
LD	(DE),A	приготовленную область.
INC	DE	
SR	NZ 0629, SA-BLANK	
LD	(IX+01),+FF	Не имеющее силы имя это только
CALL	2BF1,STK-FETCH	для +FF извлекаются параметры
		имени и проверяется их длина.
LD	HL,+FFF6	Это означает "-10".
DEC	BC	Переход вперед, если длина имени
ADD	HL,BC	не слишком длинная (т.е. не более
INC	BC	10 символов).
JR	NC,0648,SA-NAME	
LD	A,(T-ADDR-LO)	Допустимо, для программ загрузки,
AND	A	проверки и склейки с "неиме-
JR	NZ,0644,BA-NULL	ющими силу" именами или
		слишком длинным именем.

Сообщение F - неправильное имя файла.

0642 REPORT-F

RST 0008,ERROR-1 Вызов программы обработки
ошибки.

DEFB +0E

Продолжить обработку имени программы.

0644 SA-NULL

LD A,B

Переход вперед, если имя превышает
допустимую длину.

OR C

JR Z,0652,SA-DATA

LD BC,+000A

Отсечь более длинные имена.

Теперь имя переносится в рабочую область (Вторая
ячейка далее).

0648 SA-NAME

PUSH IX

Скопировать начальный адрес в ре-
гистровую пару HL.

POP HL

INC HL

Перейти к следующей ячейке.

EX DE,HL

Переключить следующие указатели
и скопировать имя.

LDIR

Теперь рассматривается множество различных
параметров, которые следуют за командой, начинается
обработка "XXX" имя "данные".

0652 SA-DATA

RST 0018,GET-CHAR

Является ли настоящий код
знаком "Данные"?

CP +E4

JR NZ,06A0,SA-SCRS

Переход, если нет.

LD A,(T-ADDR-LO)

Невозможно получить
"Склею-имя-данные".

JP Z,1C8A,REPORT-C

RST 0020,NEXT-CHAR

Увеличить CH-ADD.

CALL 28B2,LOOK-VARS

Просмотр области переменных
для массива.

SET 7,C

Установить бит 7 в имени массива.

JR NC,0672,SA-V-OLD

Переход, если обработка
существующего массива.

LD HL,+0000 Сигнал "Использование нового массива".

LD A,(T-ADDR-LO) Рассмотреть значение T-ADDR и

DEC A выдать ошибку, если хотят спасти

JR Z,0685,SA-V-NEW или проверить новый массив.

Сообщение 2 - переменная не найдена (VARIABLE NOT FOUND).

0670 REPORT-2

RST 0008,ERROR-1 Вызов программы обработки ошибки.

DEFB +01

Продолжить обработку текущего массива.

0672 SA-V-OLD

JR NZ,1C8A,REPORT-C Заметим, это не исключает простые строки.

CALL 2530,SYNTAX-Z Переход вперед, если проверяется синтаксис.

JR Z,0692,SA-DATA-1

INC HL Указать на "Младший байт длины" переменной.

LD A,(HL) Младший байт длины передается

LD (IX+0B),A в рабочую область; затем старший байт.

INC HL

LD A,(HL)

LD (IX+0C),A

INC HL

Следующий шаг после байтов длины. (Адресовать ячейку после байтов длины).

Следующая часть является общей для "Старого" и "Нового" массивов. Заметим: синтаксическая ошибка.

0685 SA-V-NEW

LD (IX+0E),C Скопировать имя массива.

LD A,+01 Принять массив номеров.

BIT 6,C Переход, если это так.

JR 068F,SA-V-TYPE

INC A
068F SA-V-TYPE
LD (IX+00),A

Это массив символов.

Спасти "Тип" в первой ячейке управляющей области.

Последняя часть оператора проверяется перед присоединением к другим ветвям.
0692 SA-DATA-1

EX DE,HL Спасти указатель в DE.
RST 0020,NEXT-CHAR Является ли следующий символ ")"
CP +29
JR NZ,0672,SA-V-OLD Дать сообщение "С", если "Нет".
RST 0020,NEXT-CHAR Повышается CH-ADD.
CALL 1BEE,CHECK-END Переход к следующему оператору, если проверяется синтаксис.
EX DE,HL Возврат указателя в регистровую
JR 075A,SA-ALL пару перед переходом вперед (указатель фиксирует начало оглавления текущего массива).

Теперь рассмотрим "SCREEN\$".

06A0 SA-SCR\$

CP +AA Является ли текущий код знаком "SCREEN\$".

JR NZ,06C3,SA-CODE Переход, если нет.
LD A,(T-ADDR-LO) Это невозможно для получения.
CP +03 "MERGE NAME SCREENS".

JR Z,1C8A,REPORT-C
RST 0020,NEXT-CHAR Повышается CH-ADD.
CALL 1BEE,CHECK-END Переход к следующему оператору, если проверяется синтаксис.

LD (IX+0B),+00 Область дисплея и область
LD (IX+0C),1B атрибутов занимают +1 и 00 ячеек,
LD HL,+4000 и эти ячейки начинаются с адреса 4000.
LD (IX+0D),L Эти элементы перемещаются в
LD (IX+0E),H область заголовка в рабочих ячейках.
JR 0710,SA-TYPES-3 Переход вперед.

Теперь рассмотрим "CODE"

0603 SA-CODE

CP	+AF	Является ли текущий код знаком "CODE".
JR	NZ,0716,SA-LINE	Переход, если нет.
LD	A,(T-ADDR-LO)	Это невозможно для получения "MERGE NAME CODE".
CP	+03	
JP	Z,1C8A,REPORT-C	
RST	0020,NEXT-CHAR	Повышается CH-ADD.
CALL	2048,PR-ST-END	Переход вперед, если оператор не окончен.
JR	NZ,06E1,SA-CODE-1	
LD	A,(T-ADDR-LO)	Невозможно получение "SAVE NAME CODE" самим собой.
AND	A	
JP	Z,1C8A,REPORT-C	
CALL	1CE6,USE-ZERO	Занести ноль в стек калькулятора для "Старта".
JR	06F0,SA-CODE-2	Переход вперед. Ищем стартовый адрес.
06E1 SA-CODE-1		
CALL	1C82,EXPT-1NUM	Извлечь первый номер.
RST	0018,GET-CHAR	Является ли текущий символ "." или нет
CP	+2C	
JR	Z,06F5,SA-CODE-3	переход, если "да" - номер был "Стартовым адресом".
LD	A,(T-ADDR-LO)	Невозможность образования "SAVE NAME CODE"
AND	A	делает невозможным получение "Старта" и "Длины".
JP	Z,1C8A,REPORT-C	
06F0 SA-CODE-2		
CALL	1CE6,USE-ZERO	Занести ноль в стек калькулятора для "Длины".
JP	06F9,SA-CODE-4	Переход вперед.

Извлечь длину после ее определения.

06F5 SA-CODE-3

RST 0020,NEXT-CHAR Повышается CH-ADD.

CALL 1C82,EXPT-1NUM Получить "Длину".

Теперь сохраним параметры в области заголовка рабочего пространства.

06F9 SA-CODE-4

CALL 1BEE,CHECK-END Переход в следующее состояние, если проверяется синтаксис.

CALL 1F99,FIND-INT2 Упаковать "Длину" в регистровую пару BC и сохранить ее.

LD (IX+0B),C

LD (IX+0C),B

CALL 1E99,FIND-INT2 Упаковать "Стартовый адрес" в регистровую пару BC и сохранить его.

LD (IX+0D),C

LD (IX+0E),B

LD H,B

LD L,C

Перенести указатель в регистровую пару HL.

"SCREEN\$" и "CODE" обозначаются, как тип 3.

0710 SA-TYPE-3

LD (IX+00),+03

Ввести номер "Типа"

JR 075A,SA-ALL

Подсоединение других ветвей (путей) программы.

Теперь рассмотрим "LINE" и "Нет дальнейших параметров".

0716 SA-LINE

CP +CA

Является ли текущий код знаком "LINE"?

JR Z,0723,SA-LINE-1

Переход, если это так.

CALL 1BEE,CHECK-END

Переход к следующему состоянию, если проверяется синтаксис.

LD (IX+0E),80

Когда нет дальнейших параметров, вводится +80,

JR 073A,SA-TYPE-0

Переход вперед.

Извлечь "Номер строки", который должен следовать за "LINE".

0723 SA-LINE-1

LD A,(T-ADDR-I.O) Учитывает "SAVE NAME LINE
NUMBER".

AND A (Спасение имени строки).

JP NZ,1C8A,REPORT-C

RST 0020,NEXT-CHAR Повышается CH-ADD.

CALL 1C82,EXPT-1NUM Передать номер в стек
калькулятора.

CALL 1BEE,CHECK-END Переход к следующему оператору,
если проверяется синтаксис.

CALL 1E99,FIND-INT2 Упаковать "Номер строки"

LD (IX+0E),B в регистр

"LINE" и "Нет дальнейших параметров" обозначаются,
как тип "0".

073A SA-TYPE-0

LD (IX+00),+00 Ввод номер "Типа".

Параметры, которые описывают программу и ее
переменные, определяются и сохраняются в заголовке
рабочей области.

LD HL,(E-LINE) Указатель конца области
переменных.

LD DE,(PROG) Указатель начала программы BASIS.

SCF Теперь выполняем вычитание для
SBC HL,DE определения длины "Программы" +

LD (IX+0B),L "переменные"; Сохраним результат.

LD (IX+0C),H

LD HL,(VARS) Повторим действие, но на этот

SBC HL,DE раз только для длины "Программы".

LD (IX+0F),L

LD (IX+10),H

EX DE,HL Перенести "Указатель" в
регистровую пару HL, как обычно.

В любом случае управляющая информация теперь
приготовлена.

Ячейка "IX+00" содержит номер типа.

Ячейки "IX+01 до IX+0A" содержат имя (+FF в "IX+01", если не имеет силы "Запрещающая комбинация").

Ячейки "IX+0B и IX+0C" содержат число байтов, которые должны находиться в блоке данных.

Ячейки "IX+0D до IX+10" содержат различные параметры, которые интерпретируются в зависимости от типа.

Программа продолжается с первой задачи до отселения SAVE от LOAD, VERIFY и MERGE.

075A SA-ALL

LD	A,(T-ADDR-LO)	Переход вперед, когда обрабаты-
AND	A	вается команда "Спасение".
JP	Z,0970,SA-CONTRL	

В случае загрузки (LOAD), проверки (VERIFY) или склейки (MERGE) первых семнадцать байтов и "Заголовок" содержат подготовленную информацию, как описано выше; и теперь наступило время для "Извлечения" заголовка с ленты.

PUSH	HL	Спасти указатель
		"Места назначения".

LD	BC,+0011	Сформировать в регистровой паре.
ADD	IX,BC	Базовый адрес "Второй
		управляющей области".

Теперь переходим в цикл; выходим из цикла только когда "Заголовок" загружен.

0767 LD-LOOK-H

PUSH	IX	Скопировать базовый адрес.
LD	DE,+0011	Загрузить семнадцать байтов.
XOR	A	Сигнал "Заголовок".
SCF		Сигнал "Загрузка".
CALL	0556,LD-BYTES	Теперь следите за заголовком.
POP	IX	Восстановить базовый адрес.
JR	NC,0767,LD-LOOK-H	Выполнять цикл до
		получения результата.

Новый "Заголовок" теперь выводится на экран, но программа продолжается только, если "Новый" заголовок совпадает со "Старым" заголовком.

LD	A,+FE	Гарантирует, что канал "S" свобо-
----	-------	-----------------------------------

CALL 1601,CHAN-OPEN ден (Открыт).
 LD (SCR-CT),+03 Установить счетчик вращения
 (Роллинг).
 LD C,+80 Сигнал "Имя не совпадает".
 LD A,(IX+00) Сравнить "Новый" тип со "Старым"
 CP (IX-11) типом.
 JR NZ,078A,LD-TYPE Переход, если "Типы" не
 совпадают.
 LD C,+F6 Если они совпадают, то сигнал -
 "десять символов совпадают".

078A LD-TYPE

CP +04 Очистка "Заголовка" является бес-
 JR NC,0767,LD-LOOK-H смыленным, если "Тип 4 или
 более".

Печатаются сообщения: - "PROGRAM" (Программа),
 "NUMBER ARRAY" (Цифровой массив), "CHARACTER ARRAY"
 (Массив символов), "BYTES"(Байты).

LD DE,+09C0 Базовый адрес блока сообщения.
 PUSH BC Спасти регистр Ц в то время, когда
 CALL 0C0A,PO-MSG печатается необходимое сообщение,
 "Новое имя" печатается и
 сравниваются "Старое" и "Новое" имена.

PUSH IX Сделать регистровую пару DE
 POP DE указателем "Нового типа" и
 LD HL,+FFF0 регистровую пару HL "Старого имени".

ADD HL,DE
 LD B,+0A Должны рассматриваться десять
 символов.

LD A,(HL) Переход вперед, если необходимо
 INC A проводить сравнение.

JR NZ,07A6,LD-NAME

LD A,C Но, если "Старое имя" является
 ADD A,B "Запрещенным", то сигнал "Десять
 LD C,A символов уже проверены".

Цикл печати символов "Нового имени", имя выведено, "Счетчик" достигает значения нуля.

07A6 LD-NAME

INC	DE	Рассмотрим каждый символ "Нового имени" по очереди.
LD	A,(DE)	
CP	(HL)	Сравнение символа с соответствующим символом "Старого имени".
INC	HL	
JR	NZ,07AD,LD-CH-PR	Не увеличивать счетчик, если нет совпадения.
INC	C	

07AD LD-CH-PR

RST	0010,PRINT-A-1	Печать "Нового" символа.
DJNZ	07A6,LD-NAME	Цикл для десяти символов.
BIT	7,C	Принять имя только, если счетчик достиг нуля.
JR	NZ,0767,LD-LOOK-H	
LD	A,+0D	Следует "Новое имя" с "Блоком возврата".
RST	0010,PRINT-A-1	

Получен корректный заголовок, и пришло время рассмотреть отдельно 3 команды: загрузка (LOAD), проверка (VERIFY) и склейка (MERGE).

POP	HL	Извлечь указатель.
LD	A,(IX+00)	"SCREEN\$" и CODE" обрабатываются командой проверка.
CR	+03	
JR	Z,07CB,VR-CONTRL	
LD	A,(T-ADDR-LO)	Переход вперед, если используется команда загрузки.
DEC	A	
JR	Z,0808,LD-CONTRL	
CP	+02	Переход вперед, если используется команда склейки; продолжение с командой проверки.
JR	Z,08D6,ME-CONTRL	

Управляющая программа "VERIFY" (Проверка)

Процесс проверки включает в себя загрузку блока данных, по одному байту, но байты не хранятся, только проверяются, эта программа используется для загрузки блоков данных, которые описаны с атрибутами "SCREEN\$" и "CODE".

07CB VR-CONTRL

PUSH	HL	Смести "Указатель".
LD	L,(IX-06)	Извлечь число байтов, описанное в
LD	H,(IX-05)	"Старом" заголовке.
LD	E (IX+0B)	Извлечь также число байтов из
LD	D,(IX+0C)	"Нового" заголовка.
LD	A,H	Переход вперед, если "Длина" не
OR	L	определена.
JR	Z,07E9,VR-CONT-1	Например, только "Загрузка имя код".
SBC	HL,DE	Лать сообщение R, если обнаружена
JR	C,0806,REPORT-R	попытка загрузить блок больший, чем требуется.
JR	Z,07E9,VR-CONT-1	Получено, что длины равны.
LD	A,(IX+00)	Сообщение R выдается также, если
CP	+03	сделана попытка сравнить блоки разной длины
JP	NZ,0806,REPORT-R	("Старая" длина больше, чем. "Новая" длина).

Программа продолжается рассмотрением "Указателя места назначения".

07E9 VR-CONT-1

POP	HL	Извлечь "Указатель", т.е. "Старт".
LD	A,H	Этот "Указатель" будет использован,
OR	L	если он не равен нулю, в случае нуля
JR	NZ,07F4,VR-CONT-2	"Начало" (Старт) находится в
LD	L,(IX+0D)	"Новом" заголовке.
LD	H,(IX+0E)	

Теперь рассматривается флаг проверка/загрузка и делается текущая загрузка.

07F4 VR-CONT-2

PUSH HL

Перенос "Указателя" в регистровую

POP IX пару IX
 LD A,(T-ADDR-LO) Переход вперед, если не
 CP +02 используется команда проверка; с
 SCF флагом переноса, сигнализирующим
 JR NZ,0800,VR-CONT-3 "Загрузку".
 AND A Сигнал "Проверка".
 0800 VR-CONT-3
 LD A,+FF Сигнал "Прием только блока данных",
 перед загрузкой блока.

Подпрограмма "LOAD A DATA BLOCK"

(Загрузка блока данных)

Подпрограмма является общей для всех программ загрузки, в случае загрузки и проверки она действует, как полный возврат из программ управления магнитофоном, но в случае склейки блок данных должен быть еще и склеен.

0802 LD-BLOCK

CALL 0556,LD-BYTES Загрузка/проверка блока данных,
 RET C возврат, если нет ошибки.

Сообщение R-лента загружается с ошибкой. (TAPE LOADING ERROR)

0806 REPORT-R

RST 0008,ERROR-1 Вызов программы.

DEFB +1A Обработка ошибок.

Программа управления "Загрузка" (LOAD)

Эта программа управляет загрузкой программ Бейсика и их проверкой или загрузкой массива.

0808 LD-CONTRL

LD E,(IX+0B) Извлечь число байтов, полученных в

LD D,(IX+0C) "Новом заголовке".

PUSH HL Спасти "Указатель места назначения".

LD	A,H	Переход вперед, если только не
OR	L	обнаружена попытка загрузить
JR	NZ,0819,LD-CONT-1	предварительно не описанный массив.
INC	DE	Добавить три байта к длине для
INC	DE	имени младшего байта и старшего
INC	DE	байта длины или новой переменной.
EX	DE,HL	
JR	0825,LD-CONT-2	Переход вперед.

Теперь посмотрим, есть ли достаточно места в памяти для нового блока данных.

0819 LD-CONT-1

LD	L,(IX-06)	Извлечь размер "Программы + пере-
LD	H,(IX-05)	менных или массива".
EX	DE,HL	
SCF		Переход вперед, если нет превышения
SBC	HL,DE	"требуемого над имеющимся";
JR	C,082E,LD-DATA	принимается в расчет имеющаяся в настоящее время в распоряжении память.

Делаем текущий анализ памяти.

0825 LD-CONT-2

LD	DE,+0005	Увеличить на пять байтов.
ADD	HL,DE	
LD	B,H	Переслать результат в регистровой па-
LD	C,L	ре BC и повести тестирование.
CALL	1F05,TEST-ROOM	

Теперь имеем дело с загрузкой массива.

082E LD-DATA

POP	HL	Извлечь новый "указатель".
LD	A,(IX+00)	Переход вперед, если идет загрузка
AND	A	программы Бейсик.

JR Z,0873,LD-PROG
 LD A,H Переход вперед, если загружается
 OR L новый массив.
 JR Z,084C,LD-DATA-1
 DEC HL Извлечь "Длину" массива путем склейки
 LD B,(HL) байтов в области переменных.
 DEC HL
 LD C,(HL)
 DEC HL Адресовать его старое имя.
 INC BC Добавить три байта к длине - одно для
 INC BC имени, два - для длины.
 LD (X-PTR),IX Временно спасти регистровую пару IX
 CALL 19E8,RECLAIM-2 на время регенерации старого
 массива.

LD IX,(X-PTR)

Теперь подготовлено пространство для нового массива в
 конце области переменных.

084C LD-DATA-1

LD HL,(E-LINE) Найти указатель в маркере конца обла-
 DEC HL сти переменных - "80 байт".
 LD C,(IX+0B) Извлечь "Длину" нового массива.
 LD B,(IX+0C)
 PUSH BC Спасти эту длину.
 INC BC Добавить три байта - один для имени и
 INC BC два - для "Длины".
 INC BC
 LD A,(IX+03) "IX+0E" старого заголовка задает имя
 массива.
 PUSH AF Имя спасается на время вычисления
 CALL 1655,MAKE-ROOM величины памяти, доступной
 INC HL пользователю, в результате "BC"

POP	AF	помещается перед "Новым 80ым байтом".
LD	(HL),A	Вводится имя.
POP	DE	Извлекается "Длина", и эти два байта
INC	HL	также вводятся.
LD	(HL),E	
INC	HL	
LD	(HL),D	
INC	HL	HL теперь указывает на первую ячейку памяти, которая должна быть заполнена с магнитной ленты.
PUSH	HL	Этот адрес переносится в регистровую
POP	IX	пару IX; устанавливается флаг переноса;
SCF		сигнал "Блок данных"; и блок
LD	A,+FF	загружается.

Теперь имеем дело с загрузкой программы Бейсика и ее переменных.

0873 LD-PROG

EX	DE,HL	Спасти "Указатель места назначения".
LD	HL,(E-LINE)	Найти адрес маркера конца текущей области переменных - "80 байт".
DEC	HL	
LD	(X-PTR),IX	Временное спасение IX.
LD	C,(IX+0B)	Извлечь "Длину" нового блока данных.
LD	B,(IX+0C)	
PUSH	BC	Сохранить копию "Длины" на время, пока программа и области переменных
CALL	19E5,RECLAIM-1	исправляются.
POP	BC	
PUSH	HL	Спасти указатель в области программы
PUSH	BC	и длину нового блока данных.
CALL	1655,MAKE-ROOM	Сделать доступной достаточную область памяти для новой

программы и ее переменных.

LD	IX,(X-PTR)	Восстановить пару регистров IX.
INC	HL	Системная переменная VARS тоже дол-
LD	C,(IX+0F)	жна быть установлена для новой
LD	B,(IX+10)	программы.
ADD	HL,BC	
LD	(VARS),HL	
LD	H,(IX+0E)	Если номер строки был определен,
LD	A,H	тогда он тоже должен быть рассмотрен.
AND	+C0	
JR	NZ,08AD,LD-PROG-1	Переход вперед, если "Нет номера";
LD	L,(IX+0D)	в противном случае установить:
LD	(NEWPPC),HL	NEWPPC & NSPPC.
LD	(NSPPC),+00	

Теперь может быть загружен блок данных.

08AD LD-PROG-1

POP	DE	Извлечь "Длину".
POP	IX	Извлечь "Старт".
SCF		Сигнал "Загрузка".
LD	A,+FF	Сигнал только "Блок данных".
JP	0802, LD-BLOCK	Теперь загрузка блока

Программа управления "MERGE"(Склейка)

В этой программе существуют три главные части:

1. Загрузка блока данных в рабочую область.
2. Склейка строк новой программы со старой программой.
3. Склейка новых переменных со старыми переменными.

Итак, начнем с загрузки блока данных.

08B6 ME-CONTRL

LD	C,(IX+0B)	Извлечь "Длину"
----	-----------	-----------------

LD	B,(IX+0C)	блока данных.
PUSH	BC	Спасти "Длину"
INC	BC	Теперь сделать доступными "Длина+1"
RST	0030,BC-SPACES	ячеек памяти в рабочей области.
LD	(HL),+80	Разместить маркер конца в следующей ячейке.
EX	DE,HL	Перенести указатель "Старта" в регистровую пару.
POP	DE	Извлечь первоначальную "Длину".
PUSH	HL	Спасти "Старт".
PUSH	HL	Теперь установить регистровую пару IX
POP	IX	для текущей загрузки:
SCF		Сигнал "Загрузка".
LD	A,+FF	Сигнал "Только блок данных".
CALL	0802,LD-BLOCK	Загрузка блока данных.

Строки новой программы склеиваются со строками старой программы.

POP	HL	Извлечь "Старт" новой программы.
LD	DE,(PROG)	Загрузить DE "Стартом" старой программы.

Переход в цикл, имеющий дело со строками новой программы.

0802 ME-NEW-LP

LD	A,(HL)	Извлечь номер строки и проверить его.
AND	+C0	
JR	NZ,08F0,ME-VAR-LP	Переход, когда окончатся все строки.

Теперь переходим во внутренний цикл, имеющий дело со строками старой программы.

08D7 ME-OLD-LP

LD	A,(DE)	Извлечь старший байт номера строки и
INC	DE	сравнить его.

CP (HL) Переход вперед, если нет сравнения, но
 INC HL в любом случае увеличиваются оба
 JR NZ,08DF, ME-OLD-L1 указателя.
 LD A,(DE) Повторить сравнение для младшего
 CP (HL) байта номера строки.
 08DF ME-OLD-L1
 DEC DE Теперь уменьшить указатель.
 DEC HL
 JR NC,08EB,ME-NEW-L2 Переход вперед, если все
 корректно для строки новой программы.
 PUSH HL В противном случае найти адрес начала
 EX DE,HL следующей старой строки.
 CALL 1988,NEXT-ONE
 POP HL
 JR 0807,ME-OLD-LP Повторить цикл для каждой
 "Старой строки".
 08EB ME-NEW-L2
 CALL 092C, ME-ENTER Ввести "Новую строку" и
 JR 08D2, VT-NEW-LP повторить внешний цикл.
 Таким же образом переменные новой программы
 "Склеиваются" с переменными старой программы.
 Цикл преобразует каждую новую переменную.
 08F0 ME-VAR-LP
 LD A,(HL) Извлечь по очереди каждую
 LD C,A переменную и проверить ее.
 CP +80 Возврат, когда все переменные
 RET Z рассмотрены.
 PUSH HL Спастись текущее состояние нового указателя.
 LD HL(VARS) Извлечь VARS (для старой
 программы).

Теперь переход во внутренний цикл для поиска области текущих переменных.

08F9 ME-OLD-VR

LD A,(HL) Извлечь для каждой переменной и
 CP +80 проверить его.
 JR Z,0923,ME-VAR-L2 Переход вперед как только
 определен маркер конца.
 CP C Сравнить имена (первые байты).
 JR Z,0909,ME-OLD-V2 Переход вперед для дальнейшего
 рассмотрения имени; возврат, если нет
 совпадения.

0901 ME-OLD-V1

PUSH BC Спаси новое имя переменной на время
 CALL 1988, NEXT-ONE пока определяется следующая
 POP BC "Старая" переменная.
 EX DE,HL Восстановить указатель в регистровой
 JR 08F9,ME-OLD-VP паре DE и повторить цикл.

Новые и старые переменные сравниваются по первым байтам, но переменные с длинными именами должны проверяться полностью.

0909 ME-OLD-V2

AND +E0 Рассматриваются только 7,6 и 5 биты.
 CP +A0

JR NZ,0921,ME-VAR-L1 Принимать во внимание все
 типы переменных, исключая "Перемен-
 ные с длинным именем".

POP DE Принять DE, как указатель первого
 PUSH DE символа "Нового имени".
 PUSH HL Спаси указатель для "Старого имени".

Переход в цикл сравнения (букв) длинных имен.

0912 ME-OLD-V3

INC HL Модификация "Старого" и "Нового"

INC DE указателя.
 LD A,(DE) Сравнить две буквы.
 CP (HL)
 JR NZ,091E,ME-OLD-V4 Переход вперед, если нет совпадения.
 RLA Работа в цикле до тех пор, пока не
 JR NC,0912,ME-OLD-V3 будет определен "Последний символ".
 POP HL Извлечь указатель начала "Старого"
 JR 0921,ME-VAR-L1 имени и переход вперед - результат успешный.
 091E ME-OLD-V4
 POP HL Извлечь указатель и переход назад -
 JR 0901, ME-OLD-V1 результат отрицательный.
 Попадаем сюда, если было обнаружено совпадение.
 0921 ME-VAR-L1
 LD A,+FF Сигнал "Замены" переменной.
 Сюда попадаем, если нет совпадения (Содержимое +80 - переменная должна быть "Суммирование").
 0923 ME-VAR-L2
 POP DE Извлечь указатель "Нового" имени.
 EX DE,HL Обмен регистров.
 INC A Если устанавливается флаг нуля, то "Замена", если сброшен, то "Добавление".
 SCF Сигнал "Обработки переменных".
 CALL 092C,ME-ENTER Теперь обеспечить вход.
 JR 08F0,ME-VAR-LP Работа в цикле, чтобы рассмотреть следующую новую переменную.

Подпрограмма "MERGE A LINE OR VARIABLE"(Склейка строки или переменной)

В эту подпрограмму входим со следующими параметрами:

Флаг переноса: сброшен - склейка строки Бейсика;

установлен - склейка переменной.

Флаг нуля: сброшен - должно быть "Добавлено"

установлен - должна быть "Замена".

HL регистровая пара - указывает на начало нового входа.

DE - регистровая пара указывает на то, где должна быть проведена склейка.

092C ME-ENTER

JR NZ,093E,ME-ENT-1 Переход, если обработка типа
"Добавление".

EX AF,A,'F' Спасты флаги.

LD (X-PTR),HL Спасты "Новый" указатель на

EX DE,HL время, пока "Старая" строка или

CALL 19B8,NEXT-ONE переменная исправляются.

CALL 19E8,RECLAM-2

EX DE,HL

LD HL,(X-PTR)

EX AF,A,'F' Восстановить флаги.

Теперь может быть определен новый вход.

093E ME-ENT-1

EX AF,A,'F' Спасты флаги.

PUSH DE Сделать копию указателя
"Места назначения".

CALL 19B8,NEXT-ONE Определить длину "Новой"
переменной/строки.

LD (X-PTR),HL Спасты указатель "Новой"
переменной/строки.

LD	HL,(PROG)	Извлечь PROG - чтобы избежать потери.
EX	(SP),HL	Спасти PROG в стеке и извлечь "Новый" указатель.
PUSH	BC	Спасти длину.
EX	AF,A,'F'	Восстановить флаги.
JR	C,0955,ME-ENT-2	Переход вперед, если добавляется новая переменная.
DEC	HL	Новая строка добавляется перед ячейкой "Места назначения".
CALL	1655,MAKE-ROOM	Подготовить память для новой строки.
INC	HL	
JR	0958,ME-ENT-3	Переход вперед.
	0955 ME-ENT-2	
CALL	1655,MAKE-RQOM	Подготовить память для новой переменной.
	0958 ME-ENT-3	
INC	HL	Указывает на первую новую ячейку.
POP	BC	Восстанавливает длину.
POP	DE	Восстанавливает PROG и хранит ее в соответствующем месте.
LD	(PROG),DE	
LD	DE,(X-PTR)	Извлекает также "Новый указатель".
PUSH	BC	Опять спасает длину и "Новый"
PUSH	DE	указатель.
EX	DE,HL	Переключает указатели и копирует
LDIR		"Новую" переменную/строку - в подготовленную память.
"Новая" переменная/строка теперь должна быть перемещена из рабочей области.		
POP	HL	Извлечь "Новый" указатель.

POP	C	Извлечь длину.
PUSH	DE	Спасти "Старый" указатель (указывает на ячейку после "добавления" переменной/строки).
CALL	19E8,RECLAM-2	Переместить переменную/строку из рабочей области.
POP	DE	Возврат со "старым" указателем в регистровой паре DE.
RET		

Подпрограмма управления "SAVE" (Спасение)

Действия по спасению программы или блока данных являются очень простыми.

0970 SA-CONTRL

PUSH	HL	Спасти указатель.
LD	A,+FD	Обеспечить, чтобы канал был открыт
CALL	1601,CHAN-OPEN	(Свободен).
XOR	A	Сигнал "Первое сообщение".
LD	DE,+09A1	Печать сообщения - начало ленты,
CALL	0C0A,PO-MSG	затем нажать любую клавишу.
SET	5,(TV-FLAG)	Сигнал "Очистка необходимого экрана".
CALL	1504,WAIT-KEY	Ожидание нажатия клавиши.

При нажатии клавиши заголовок спасается.

PUSH	IX	Спасение базового адреса "Заголовок" в машинном стеке.
LD	DE,+0011	Должно быть спасено 17 байтов.
XOR	A	Сигнал "Этот заголовок".
CALL	04C2,SA-BYTES	Послать "Заголовок" с лидирующим байтом "Типа" и завершающим байтом паритета.

Теперь следует короткая задержка перед спасением блока программ/данных.

POP	IX	Восстановить указатель "Заголовок".
LD	B,+32	Задержка на пятьдесят прерываний.
0991 SA-1-SEC		
HALT		т.е. одна секунда.
DJNZ	0991, SA-1-SEC	
LD	E,(IX+0B)	Извлечь длину блока данных, который
LD	D,(IX+0C)	должен быть спасен.
LD	A,+FF	Сигнал "Блок данных".
POP	IX	Извлечь "Указатель начала блока" и
JP	04C2,SA-BYTES	спасти блок.

Сообщения магнитофона

Каждое сообщение дается с инвертированием последнего символа (+80H).

09A1 DEFB	+80	- прошел байт инициализации.
09A2 DEFM		- включить ленту, затем нажать любую клавишу
09C1 DEFM		- "возврат каретки" - программа.
09CB DEFM		- "возврат каретки" - массив чисел.
09DA DEFM		- "возврат каретки" - массив символов.
09EC DEFM		- "возврат каретки" - байты.

Программы обработки экрана и принтера

Программы вывода на печать "PRINT OUT"

Все типы печати - печать на полный экран, печать в нижней части экрана и печать на АЦПУ управляются этими программами.

При входе в программу вывода на печать в регистре А содержится код управляющего символа, символ для печати или знак для печати.

09F4 PRINT OUT

CALL	0B03,PO-FETCH	Текущая позиция печати.
CP	+20	Если код является разрешенным
JR	NC,0AD9,PO-ABLE	для печати символом, то переход.
CP	+06	Напечатать знак вопроса для кодов
JR	C,0A69,PO-QUEST	в диапазоне +00 - +05.
CP	+18	То же самое для кодов +18 - +1A
JR	NC,0A69,PO-QUEST	
LD	HL,+0A0B	База таблицы управления.
LD	E,A	Переместить код в регистровую
LD	D,+00	пару DE.
ADD	HL,DE	Индексировать таблицу и извлечь
LD	E,(HL)	смещение.
ADD	HL,DE	Добавить смещение и сделать пере-
PUSH	HL	ход в соответствующую программу.
JR	0B03,PO-FETCH	

Таблица управляющих символов

Адрес	Смещение	Символ
0A11	4E	команда печати
0A12	57	редактор
0A13	10	курсор влево
0A14	29	курсор вправо
0A15	54	курсор вниз
0A16	53	курсор вверх
0A17	52	удалить
0A18	37	вход
0A19	50	не используется
0A1A	4F	не используется
0A1B	5F	управ.лентой
0A1C	5E	управл.бумагой
0A1D	5D	управл. мерцанием
0A1E	5C	управл. яркостью
0A1F	5B	управл. негативом
0A20	5A	управл. перекрытием
0A21	54	управл.вводом

0A22 53 управл.табуляцией

Подпрограмма "Курсор влево"

При входе в подпрограмму в регистре В содержится текущий номер строки и в регистре С - текущий номер колонки.

0A23 PO-BACK-1

INC	C	.Смещение влево на одну колонку.
LD	A,+22	Изменять до тех пор, пока не будет
CP	C	достигнута крайняя левая позиция.
JR	NZ,0A3A,PO-BACK-3	
BIT	1,(FLAGS)	Если работаем с принтером, переход
JR	NZ,0A38,PO-BACK-2	вперед.
INC	B	Подняться на одну строку вверх.
LD	C,+02	Установить значение колонки.
LD	A,+18	Проверить на код верхней строки.
CP	B	Заметим: выход должен быть +19.
JR	NZ,0A3A,PO-BACK-3	Изменять до тех пор, пока не
		будет достигнут верх экрана.
DEC	B	Неприемлемо, поэтому опустить вниз
		строку.

0A38 PO-BACK-2

LD	C,+21	Установить левую колонку.
----	-------	---------------------------

0A3A PO-BACK-3

JP	0DD9,CL-SET	Сделать прямой возврат через CL-SET и PO-STOP.
----	-------------	--

Подпрограмма "Курсор вправо"

Подпрограмма осуществляет действие, идентичное команде Бейсика PRINT OVER 1; CHRS# 32;

0A3D PO-RIGHT

LD	A,(P-FLAG)	Извлечь P-FLAG и спасти его в
PUSH	AF	машинном стеке.
LD	(P-FLAG),+01	Установить P-FLAG в положение
		OVER1 (через один).

LD	A,+20	"Пробел".
CALL	0B65,PO-CHAR	Печать символа.
POP	AF	Извлечь старое значение P-FLAG.
LD	(P-FLAG),A	
RET		Конец.

Заметим: программист забывает о выходе через PO-STORE.

Подпрограмма "Возврат каретки"

Если распечатка выводится на принтер, то символ возврата каретки означает, что буфер принтера должен быть очищен, если распечатка выводится на экран, то проводится текст на ролинг перед тем, как уменьшается номер строки.

0A4F PO-ENTER

BIT	1,(FLAGS)	Переход вперед, если управление
JP	NZ,0ECD,COPY-BUFF	принтером.
LD	C,+21	Установить левую колонку.
CALL	0C55,PO-SCR	Ролинг, если необходимо.
DEC	B	Теперь опустить строку.
JP	0DD9,CL-SET	Сделать косвенный возврат через
		CL-SET и PO-STORE.

Подпрограмма "Печать запятой"

Изменяется значение текущей колонки, и в регистр A заносится +00 (для TAB0) или +10 (для TAB 16).

0A5F PO-COMMA

CALL	0B30,PO-FETCH	Повтор?
LD	A,C	Текущий номер колонки.
DEC	A	Передвинуть позицию на две
DEC	A	колонки и затем проверить.
AND	+10	Регистр должен содержать +00 или +10
JR	0AC3,PO-FILL	выход через PO-FILL.

Подпрограмма "Печать знака вопроса"

Знак вопроса печатается всякий раз, когда делается попытка распечатать запрещенный код.

0A69 PO-QUEST

LD A,+3F

Символ "?".

JR 0A09,PO-ABLE

Теперь печать этого символа.

Программа "Управляющие символы с операндами"
(THE CONTROL CHARACTERS WITH OPERANDS
ROUTINE).

Управляющие символы от INC до OVER требуют единственного операнда, тогда как управляющие символы AT и TAB требуют по два операнда, программа управляет кодом управляющего символа, спасенным в TVDATA-LO, первый операнд в TVDATA-H1 или в регистре A.

0A60 PO-TV-2.

LD DE,+0A87

Спаси первый операнд в TVDATA-

LD (TVDATA-H1),A

H1 и изменить адрес выхода программы.

JR 0A80,PO-CHANGE

Изменяется на PO-CONT(+0A87).

Переход сюда, когда обработка символов AT и TAB.

0A75 PO-2-OPER

LD DE,+0A6D Код символа будет спасен в TVDATA-LD

JR 0A7D,PO-TV-1

и адрес "Выхода" программы

изменяется на PO-TV-2(+0A6D).

Переход сюда, когда обработка цветового номера - от INC до OVER.

0A7A PO-1-OPER

LD DE,+0A87

"Выход" программы должен быть изменен на PO-CONT (+0A87).

0A7D PO-TV-1

LD (TVDATA-L0),A Спаси код управляющего символа.

Текущий адрес "Выхода" программы временно изменяется.

0A80 PO-CHANGE

LD HL,(CURCHL)

HL будет указывать на адрес "Выхода" программы.

LD (HL),E

Вводим новый адрес "Выхода"

INC HL

программы и посредством этого

LD (HL),D добиваемся того, чтобы следующий код
RET символа был рассмотрен как операнда.

Теперь операнды для продолжения программы.

0A87 RO-CONT

LD DE,+09F4 Восстановить первоначальный адрес

CALL 0A80,PO-CHANGE для PRINT-OUT (+09F4).

LD HL,(TVDATA) Извлечь управляющий код и первый
операнда, если требуются два операнда.

LD D,A Последний операнда и код управления

LD A,L перемещаются.

CP +16 Переход вперед, если обработка от INC

JP C,2211,CO-TEMPS до OVER.

JR NZ,0AC2,PO-TAB Переход вперед, если обработка TAB.

Теперь имеем дело с управляющим символом AT.

LD B,H Номер строки.

LD C,D Номер колонки.

LD A,+1F Реверс номера колонки, т.е. +00 - +1F

SUB C становится +1F - +00.

JR C,0AAC,PO-AT-ERR Должно быть в диапазоне.

ADD A,02 Добавить в смещение для получения в

LD C,A "C" +21 - +22.

BIT 1,(FLAGS) Переход вперед, если управление

JR NZ,0ABF,PO-AT-SET принтером.

LD A,+16 Реверс номера строки, т.е. +00 - +15

SUB B становится +16 - +01.

0AAC PO-AT-ERR

JP C,1E9F,REPORT-B Соответствующий переход вперед,
но "Если".

INC A Диапазон +16 - +01 становится +17

LD B,A +02.

INC B Теперь +18 - +03.

BIT 0,(TV-FLAG) Если печать в нижней части экрана, то

JP NZ,0C55,PO-SCR посмотреть нужен ли ролинг.

CP (DF-CZ) Дать сообщение 5 - вывод экрана, если

JP C,0C86,REPORT-5 необходимо.

0ABF PO-AT-SET

JP 0DD9,CL-SET Возврат CL-SET и PO-STORE.

И управляющий символ TAB.

0AC2 PO-TAB

LD A,H Извлечь первый операнд.

0AC3 PO-FILL

CALL 0B03,R0-FETCH Текущая позиция печати.

ADD A,C Добавить значение текущей колонки.

DEC A Определение, сколько "пробелов" по мо-

AND +1F дулю 32 потребуется и возврат, если

RET Z результат равен нулю.

LD D,A Использовать D как счетчик.

SET 0,(FLAGS) Запретить "Управляющий пробел"

0AD0 PO-SPACE

LD A,+20 Печать "D-число" пробелов.

CALL 0C3B,PO-SAVE

DEC D

JR NZ,0AD0,PO-SPACE

RET Теперь конец.

Разрешенные к печати символьные коды

Необходимый символ (или символы) печатаются при вызове PO-ANY с последующим вызовом PO-STORE.

0AD9 PO-ABLE

CALL 0B24,PO-ANY Печать символа (0B) и продолжение в PO-STORE.

Подпрограмма "Сохранение позиции" (POSITION-STORE)

Новые значения позиций "Строки и колонки" и адрес пикселя сохраняются в соответствующих системных переменных.

0ADC PO-STORE

BIT 1,(FLAGS) Переход вперед, если обработка

JR NZ,0AFC,PO-ST-PR принтера.

BIT 0,(TV-FLAG) Переход вперед, если обработка ниж-

JR NZ,0AF0,PO-ST-E ней части экрана.

LD	(S-POSH),BC	Спасти значения, которые имеют
LD	(DF-CC),HL	отношение к главной части экрана.
RET		Затем возврат.
0AF0 PO-ST-E		
LD	(S-POSN),BC	Спасти значения, которые имеют
LD	(ECHO-E),BC	отношения к нижней части экрана.
LD	(DF-CCL),HL	
RET		Теперь возврат.
0AF0 PO-ST-PR		
LD	(S-POSN),C	Спасти значения, которые имеют
LD	(PR-CC),HL	отношение к буферу принтера.
RET		Теперь возврат.

Подпрограмма "Извлечения позиции"

Текущие позиционные параметры извлекаются из соответствующих системных переменных.

0B03 PO-FETCH

BIT	1,(FLAGS)	Переход вперед, если обработка
JR	NZ,0B1D,PO-F-PR	принтера.
LD	BC,(S-POSN)	Извлечь значения, имеющие
LD	HL,(DF-CC)	отношение к главной части экрана и
BIT	0,(TV-FLAC)	возврат, если это было задумано.
RET	Z	
LD	BC,(S-POSN)	В противном случае извлечь значе-
LD	HL,(DF-CC)	ния, имеющие отношение к нижней
		части экрана.

0B1D PO-F-PR

LD	C,(P-POSH)	Извлечь значения, имеющие
LD	HL,(PR-CC)	отношение к буферу принтера.
RET		

Подпрограмма "Печать какого-либо символа"
(PRINT ANY CHARACTERS)

Имеем дело с обычными символьными кодами, кодами знаков, определенными пользователем графическими кодами (UDG) и графическими кодами.

0B24 PO-ANY

CP	+80	Переход вперед с обычными
JR	C,0B65,PO-CHAR	символьными кодами.
CP	+90	Переход вперед с кодами
JR	NC,0D52,PO-T&UDG	знака и UDG кодами.
LD	B,A	Занести графический код.
CALL	0B38,PO-GR-1	Создать графическую форму
CALL	0B03,PO-FETCH	HL нарушен, поэтому опять
		извлечение.
LD	DE,+5CD2	Занести в DE точку начала
		формирования графики, т.е.
		MEMBOT.
JR	0B7F,PO-ALL	Переход вперед для печати
		графического символа.

Графические символы формируются в области памяти калькулятора, т.е. MEM-0 и MEM-1.

0B38 PO-GR-1

LD	HL,+5C92	Это MEMBOT.
CALL	0B3E,PO-GR-2	В действительности текущая
		программа вызывается дважды.

0B3E PO-GR-2

RR	B	Определить бит 0 (и позже бит 2)
SBC	A,A	графического кода.
AND	+0F	Регистр A будет содержать +00 или
		+0F в зависимости от значения бита
		в коде.
LD	C,A	Спасти результат в C.
RR	B	Определить бит 1 (и позже бит 3) в
SBC	A,A	графическом коде.
AND	+F0	Регистр A будет содержать +00 или

OR	C	+F0, два результата складываются
LD	C,04	(логически), а регистр A содержит
		половину формы символа
0B4C PO-GR-3		
LD	(HL),A	и должен быть использован 4 раза.
INC	HL	
DEC	C	Это делается для верхней половины
JR	NZ,084C,PO-GR-3	формы символа, а затем для нижней.
RET		
Теперь выделяются коды знаков и UDG символы.		
0B52 PO-T&UDG		
SUB	+A5	Переход вперед с кодами знаков.
JR	NC,0B5F,PO-T	
ADD	A,+15	UDG кодами являются +00 - +0F.
PUSH	BC	Спасти значение текущей позиции в
		машинном стеке.
LD	BC,(UDG)	Извлечь базовый адрес области UDG и
JR	0B6A,PO-CHAR-2	переход вперед.
0B5F PO-T		
CALL	0C10,PO-TOKENS	Теперь печать знака и возврат
JR	0B03,PO-FETCH	через PO-FETCH.
Определяется форма требуемого символа.		
0B65 PO-CHAR		
PUSH	BC	Текущая позиция спасается.
LD	BC,(CHARS)	Извлекается базовый адрес
		символьной области.
0B6A PO-CHAR-2		
EX	DE,HL	Спасается адрес печати.
LD	HL,+5C3B	Флаги.
RES	0,(HL)	Делается поправка на
		управляющий пробел.
CP	+20	Переход вперед, если символ не
JR	NZ,0B76,PO-CHARS-3	является "пробелом".
SET	0,(HL)	Запретить, если это так.
0B76 PO-CHARS-3		

LD	H,+00	Теперь переслать код символа
LD	L,A	в регистровую пару HL.
ADD	HL,HL	В действительности код символа
ADD	HL,HL	умножается на 8.
ADD	HL,HL	
ADD	HL,BC	Определяется базовый адрес
		формы символа.
POP	BC	Извлекается текущая позиция
EX	DE,HL	и базовый адрес помещается в
		регистровую пару DE.

Подпрограмма "Печать всех символов"

(PRINT ALL CHARACTERS)

Подпрограмма используется для печати всех "8X8" битовых символов. На входе регистровая пара DE содержит базовый адрес формы символа, HL регистр адрес места назначения и регистровая пара BC - текущие значения "Строки и колонки".

0B7F PR-ALL

LD	A,C	Извлечь номер колонки.
DEC	A	Сдвинуть на одну колонку вправо.
LD	A,+21	Переход вперед до тех пор, пока не
JR	NZ,0B93,PR-ALL-1	зафиксируется новая строка.
DEC	B	Спуститься на одну строку.
LD	C,A	Номер строки является +21.
BIT	1,(FLAGS)	Переход вперед, если обработка экрана.
JR	Z,0B93,PR-ALL-1	
PUSH	DE	Спасти базовый адрес на время, пока
CALL	0ECD, COPY-BUFF	буфер принтера выводится
		(опустошается).
POP	DE	
LD	A,C	Скопировать номер новой колонки.
0B93	PR-ALL-1	
CP	C	Проверить, используется ли новая
PUSH	DE	строка, если "да", то посмотреть нужен

CALL Z,0C55,PO-SCR ли роллинг.
POP DE

Теперь рассмотрим текущее состояние "INVERS & OVER".

PUSH BC Спастн значение позиции и адрес места назначения в машинном стеке.
PUSH HL

LD A,(P-FLAG) Извлечь P-FLAG и прочитать бит 0.

LD B,+FF Подготовить "OVER-MASK" в регистре B, т.е. OVER 0=+00 & OVER1=+FF.

RRA

JR C,0BAA,PR-ALL-2

INC B

0BA4 PR-ALL-2

RRA

RRA

SBC A,A

LD C,A

LD A,+08

AND A

Прочестъ бит 2 P-FLAG и подготовить "INVERSE MASK" в регистре C т.е. INVERSE 0=+00 & INVERSE 1=+FF.

BIT 1,(FLAGS)

JR Z,0BD6,PR-ALL-3

SET 1,(FLAGS2) Сигнал "Буфер принтера" не пустой.

SCF Установить флаг переноса, чтобы показать, что используется принтер.

0BE6 PR-ALL-3

EX DE,HL

Обмен адреса места назначения и базового адреса перед переходом в цикл.

Теперь может быть напечатан символ, делается восемь проходов цикла - один для каждой "Пиксель-строки".

0BB7,PR-ALL-4

EX AF,A,'F'

Флаг переноса устанавливается, когда используется принтер, спасти этот флаг в 'F'.

LD A,(DE)

Извлечь текущую "Пиксель-строку".

AND	B	Использовать "OVER-MASK" и затем
XOR	(HL)	произвести операцию или (XOR)
		результата и "Пиксель-строки" формы
		символа.
XOR	C	В завершении рассмотреть
		"INVERSE-MASK".
LD	(DE),A	Ввести результат.
EX	AF,A,'F'	Извлечь флаг принтера и переход
JR	C,0BD3,PR-ALL-6	вперед, если необходимо.
INC	DE	Увеличить адрес места назначения.
0BC1	PR-ALL-5	
INC	HL	Увеличить "Пиксель-строку" формы
		символа.
DEC	A	Уменьшить счетчик и вернуться в
JR	NZ,0BB7,PR-ALL-4	цикла до получения "0".

Таким образом, напечатан символ и. установлен необходимый байт атрибутов.

EX	DE,HL	Занести в регистр и содержимое
DEC	H	старшего адреса для области символа.
BIT	1,(FLAGS)	Установить байт атрибута только, если
CALL	Z,0BDB,PO-ATTR	идет вывод на экран.
POP	HL	Восстановить первоначальный адрес
POP	BC	места назначения (знако-места) и
		значения позиции.
DEC	C	Уменьшить номер колонки и увеличить
INC	HL	адрес знакоместа перед возвратом.
RET		

Когда используется принтер, адрес знакоместа должен быть увеличен на +20,0BD3 PO-ALL-6

EX	AF,A,'F'	Спасти еще раз флаг принтера.
LD	A,+20	Необходимое увеличение значения.
ADD	A,E	Добавить значение и занести
LD	E,A	результат назад в регистр E.
EX	AF,A,'F'	Извлечь флаг.
JR	0BC1,PR-ALL-5	Возврат назад в цикл.

Подпрограмма "Установка байта атрибутов"

(SET ATTRIBUTE BYTE)

Соответствующий байт атрибутов определяется и извлекается, новое значение формируется путем преобразования старого значения, ATTR-T, MASK-T и P-FLAG. В заключение это новое значение копируется в область атрибутов. (***T-временное, ***- P-постоянное).

0BDB PO-ATTR

LD	A,H	Старший байт знакомства делится на 8 и
RRCA		суммируется с +03 для определения, какая
RRCA		треть экрана адресуется, т.е. 00,01
AND	+03	или 02
OR	+58	Теперь формируется старший байт для
LD	H,A	области атрибутов.
LD	DE,(ATTR-T)	D содержит ATTR-T и E содержит MASK-T.
LD	A,0BDB	Старое значение атрибутов.
XOR	E	Принимается во внимание значение
AND	D	MASK-T и ATTR-R.
XOR	E	
BIT	6,(P-FLAG)	Переход вперед, пока не достигнем стр.9
JR	Z,0BFA,PO-ATTR-1	(PAPER 9).
AND	+C7	Ранее установленный цвет бумаги
		игнорируется, и в зависимости от того,
BIT	2A	какого цвета чернила (белые или черные),
JR	NZ,0BFA,PO-ATTR-1	новый цвет бумаги будет
XOR	+38	черным (000) или белым(111)
0BFA PO-ATTR-1		
BIT	4,(P-FLAG)	Переход вперед, пока не достигнем
JR	Z,0C08,PO-ATTR-2	INK 9 (чернила 9).
AND	+F8	Ранее установленный цвет чернил
BIT	5,A	игнорируется, и в зависимости от того,
JR	NZ,0C08,PO-ATTR-2	какой цвет бумаги - белый или
		черный, новый цвет чернил будет

XOR	+07	черным (000) или белым(111);
0C08 PO-ATTR-2		
LD	(HL),A	Вводим новый атрибут
RET		и возврат.

Подпрограмма "Печать сообщения"
(THE 'MESSAGE PRINTING')

Программа используется для печати сообщений и знаков, регистр A содержит входной номер сообщения или знака в таблице, регистровая пара DE содержит базовый адрес таблицы.

0C0A PO-MSG

PUSH	HL	Старший байт последнего обращения к
LD	H,+00	машинному стеку делается равным нулю
EX	(SP),HL	для того, чтобы подавить (отбросить)
		дополняющие пробелы (см.ниже).

JR 0C14,PO-TABLE Переход вперед.

Сюда попадаем, когда работаем с расширенными кодами знаков.

0C10 PO-TOKENS

LD	DE,+0095	Базовый адрес знаковой таблицы.
PUSH	AF	Спасти код в стеке (диапазон +00-+5A, RND-COPY).

Используется таблица и печатается соответствующий вход.

0C14 PO-NABLE

CALL	0C41,PO-SEARCH	Определение необходимого входа.
JR	C,0C22,PO-EACH	Печать сообщения/знака.
LD	A,+20	Должен быть напечатан "пробел"
BIT	0,(FLAGS)	перед сообщением/знаком.
CALL	Z,0C3B,PO-SHAW	

По очереди печатаются символы сообщения/знака.

0C22 PO-EACH

LD	A,(DE)	Собрать код.
AND	+7F	Аннулировать "Инвертированный бит".
CALL	0C3B,PO-SAVE	Печать символа.

LD	A,(DE)	Повторно собрать код.
INC	DE	Увеличить указатель.
ADD	A,A	"Инвертированный бит" передается во
JR	NC,0C22,PO-EACH	флаг переноса и сигнализирует о
		конце сообщения/знака, в противном
		случае осуществляется возврат назад.
		Теперь смотрим, требуются ли "дополняющие пробелы".
POP	DE.	Для сообщений -D содержит +00; для
		знаков - D содержит +00 - +5A.
CP	+48	Переход вперед, если последний символ
JR	Z,0C35,PO-TRSP	был "&".
CP	+82	Возврат, если последний символ был
RET		любой другой до "A".
	0C35 PO-TRSP	
LD	A,D	Проверить значение D и возврат, если
CP	+03	он фиксирует сообщение, RND,
RET	C	INKEY & или P1
LD	A,+20	Требовать "Дополнительные пробелы".

Подпрограмма "PO-SAVE"

Эта подпрограмма позволяет осуществить повторную печать символов, соответствующие регистры спасаются на время вызова подпрограммы "PRINT-OUT".

0C3B PO-SAVE

PUSH	DE	Спасти регистровую пару DE.
EXX		Спасти HL и BC.
RST	0010,PRINT-A-1	Печать одного символа.
EXX		Восстановить HL и BC.
POP	DE	Восстановить DE.
RET		Окончание.

Подпрограмма "Изучение таблицы"(THE TABLE SEARCH)

Подпрограмма возвращает в регистровой паре DE адрес начального символа требуемого входа, и флаг переноса сброшен, если должен быть учтен "Лидирующий (управляющий) пробел".

0C41 PO-SEARCH

PUSH	AF	Спасти "Входной номер".
EX	DE,HL	HL теперь содержит базовый адрес.
INC	A	Задать диапазон +01 - ?

0C44 PO-STER

BIT	7,(HL)	Ждать, пока не получим "инвертированный символ".
INC	HL	

JR Z,0C44,PO-STER

DEC	A	Перебор входов, пока не будет найден
-----	---	--------------------------------------

JR NZ,0C44,PO-STER соответствующий.

EX	DE,HL	DE указывает на начальный символ.
POP	AF	Извлечь "Входной номер" и осуществить возврат с битом переноса для первых тридцати двух входов.
CP	+20	
RET	C	
LD	A,(DE)	Однако, если начальным символом является буква, то может быть
SUB	+41	необходим "Управляющий пробел".
RET		

Подпрограмма "Проверка для роллинга"(TEST FOR SCROLL)

Эта подпрограмма вызывается всякий раз, когда может быть необходимо вращение дисплея (Роллинга), это возможно в трех случаях:

- 1 - когда обрабатывается символ "Возврат каретки";
- 2 - когда используется AT в строке INPUT;
- 3 - когда текущая строка заполнена и следующая строка должна быть использована.

На входе регистр В содержит номер строки в процессе проверки.

0C55 PO-SCR

BIT	1,(FLAGS)	Прямой возврат, если используется принтер.
RET	NZ	
LD	DE, +0DD9	Предварительная загрузка в машинный стек адреса "CL-SET".
PUSH	DE	
LD	A,B	Передача номер строки.
BIT	0,(TV-FLAG)	Переход вперед, если рассматривается "INPUT... AT...".
JR	NZ,0D02,PO-SCR-4	
JR	(DF-SZ)	Возврат через SL-SET, если номер строки больше, чем значение DE-SZ; дать сообщение 5, если он меньше; в противном случае продолжать, переход вперед, если только мы не имеем дело с "Автоматической распечаткой".
JR	C,0C86,REPORT-5	
RET	NZ	
BIT	4(TV-FLAG)	
JR	Z,0C88,PO-SCR-2	
LD	E,(BREG)	Извлечь счетчик строк.
DEC	E	Уменьшить этот счетчик.
JR	Z,0CD2,PO-SCR-3	Переход вперед, если листинг должен "Вращаться".
LD	A, +20	В противном случае открыть канал "К",
CALL	1601,CHAN-OPEN	восстановить указатель стека,
LD	SP,(LIST-SP)	флаг окончания автоматической распечатки через CL-SET.
RES	4,(TV-FLAG)	
RET		
Сообщение 5 - вывод экрана		
OC86 REPORT-5		
RST	0008,ERROR-1	Вызов программы обработки ошибки.
DEFB	+04	
Теперь посмотрим, нужен ли быстрый "Роллинг"?		
OC88 PO-SCR-2		
DEC	(SCR-CT)	Уменьшить счетчик вращений и приступить к вызову только, если получится "0".
JR	NZ,0CD2,PO-SCR-3	
Приступить к получению вызываемого сообщения.		
LD	A, +18	

SUB	B	
LD	(SCR-CT),A	
LD	HL,(ATTR-T)	Спасаются текущие значения ATTR-T и MASK-T.
PUSH	HL	
LD	A,(P-FLAG)	Спасается значение P-FLAG.
PUSH	AF	
LD	A+FD	Открывается канал "К".
CALL	0601,CHAN-OPEN	
XOR	A	Сообщение "Роллинг"?
LD	DE,0CF8	Идентично сообщению "0". Это
CALL	0C0A,PO-MSG	сообщение теперь распечатывается.
SET	5,(TV-FLAG)	Сигнал "Очистить нижнюю часть экрана после нажатия клавиши".
LD	HL,+5C3B	Это флаги.
SET	3,(HL)	Сигнал "Режим L".
RES	5,(HL)	Сигнал "Еще нет клавиши".
EXX		<i>Заметим:</i> DE должен быть также спасен.
CALL	15D4,WAIT-KEY	Извлечь клавиши.
EXX		Сохранить регистры.
CP	+20	Теперь переход вперед, на
JR	Z,0D00,REPORT-D	сообщение D- "BREAK-CONT REPERATS", если были нажаты
CP	+E2	клавиши "BREAK", "STOP", "N" или
JP	Z,0D00,REPORT-D	"H"; в противном случае
CR	+20	интерпретировать нажатие клавиши как необходимое "Вращение"
CP	+6E	дисплея (перелистывание).
JR	Z,0D00,REPORT-D	Открыть канал "S".
LD	A,+FE	
CALL	1601,CHAN-OPEN	
POP	AF	Восстановить значение P-FLAG.
LD	(P-FLAG),A	
POP	HL	Восстановить значение ATTR-T
LD	(ATTR-T),HL	и MASK-T.

Теперь дисплей "вращается" (Роллинг).

0CD2 PO-SCR-3

CALL	0DFE,CLSC-ALL	Вращается весь дисплей.
LD	B,(DF-SZ)	Определяются и спасаются номера
INC	B	строки и колонки для начала строки
LD	C,+21	над нижней частью дисплея.
PUSH	BC	
CALL	0E9B,CL-ADDR	Теперь определяется
LD	A,H	соответствующий байт атрибутов
		для области символов.

RRCA

RRCA

RRCA

AND +03

OR +58

LD H,A

Запрашиваемая строка будет иметь значения атрибутов "Нижней части", и новая строка в нижней части дисплея может иметь значение "ATTR-P", таким образом значения атрибутов изменяются.

LD	DE,+5AE0	DE указывает на первый байт атрибутов нижней строки.
----	----------	--

LD	A,(DE)	Извлекается значение.
----	--------	-----------------------

LD	C,(HL)	Значение "Нижней части".
----	--------	--------------------------

LD	B,+20	Здесь тридцать два байта.
----	-------	---------------------------

EX	DE,HL	Обмен указателей.
----	-------	-------------------

0CF0 R0-SCR-3A

LD	(DE),A	Сделать 1-ый обмен и затем продолжать.
----	--------	--

LD	(HL),C	Используются те же значения для 32-х
----	--------	--------------------------------------

INC	DE	байтов атрибутов, обрабатываемых
-----	----	----------------------------------

INC	HL	двух строк.
-----	----	-------------

DJNZ 0CF0,PO-SCR-3A

POP	BC	Номера строки и колонки нижней строки
		"Верхней части" извлекаются перед
		возвратом.

RET

Сообщение "Вращение?"

0CF8

DEFB +80 Выставить маркер-через шаг.

DEFB +73,+63,+72,+6F S-C-R-O.

DEFB +6C,+6C,+BF I-I-? (Инвертированные).

Сообщение D-BREAK-CONT повторяется.

0D00 REPORT-D

RST 0008,ERROR-1 Вызов программы обработки ошибки.

DEFB +0C

Нижняя часть дисплея обрабатывается следующим образом.

0D02 PO-SCR-4

CP +02 Ошибка "Вывода на экран" выдается,

JR C,0C86,REPORT-5 если нижняя часть получается

ADD A,(DF-SZ) "слишком большой" и делается

SUB +19 возврат, если вращение не нужно.

RET NC

NEG Регистр A теперь содержит "Число необходимых вращений".

PUSH BC Номера строки и колонки спасаются.

LD B,A Спасаются "Число вращений"

LD HL,(ATTR-T) ATTR-T,MASK-T и P-FLAG.

PUSH HL

CALL 0D04,TEMPS Должны использоваться

"Постоянные цветовые атрибуты".

LD A,B Извлекается "Число вращений".

Нижняя часть экрана теперь вращается "А" число раз.

0D1C PO-SCR-4A

PUSH AF Спасти "Число".

LD HL,+5C6B Это DF-SZ.

LD B,(HL) Значение B DF-SZ увеличивается, ре-

LD A,B гистр B используется для хранения

INC A прежнего значения и регистр A для

LD (HL),A нового значения.

LD HL,+5C89 Это S-POST-H1 (Старший).

CP (HL) Переход осуществляется только,

JR	C,0D20,PO-SCR-48	если нижняя часть дисплея должна вращаться (B=старое DF-SZ)
INC	(HL)	В противном случае S-POSN-HI
LD	B,+18	увеличивается и вращается весь дисплей (B=18).
0D2D PO-SCR-4B		
CALL	0E00,CL-SCROLL	Вращение "В"линий.
POP	AF	Извлечь и уменьшить "Счетчик вращения".
DEC	A	
JR	NZ,0D1C,PO-SCR-4A	Возврат назад до тех пор, пока не окончится.
POP	HL	Восстановить значение P-FLAG.
LD	(P-FLAG),L	
POP	HL	Восстановить значение ATTR-T в MASK-T.
LD	(ATTR-T),HL	
LD	BC,(S-POSN)	В случае, если S-POSN изменилось, вызывается CL-SET для передачи извлеченного значения В DF-CC.
RES	0,(TV-FLAG)	
CALL	0DD9,CL-SET	Сброс флага, чтобы индицировать конец.
SET	0,(TV-FLAG)	
POP	BC	Обработки нижней части экрана:
RET		извлечь номер строки и колонки и, затем возврат

Программа "Временные цветовые характеристики" (TEMPORARY COLOUR ITEMS)

Это наиболее важная подпрограмма, она используется всякий раз, когда "Постоянные" элементы должны быть скопированы в область "Временных" переменных, сначала рассмотрим ATTR-T и MASK-T.

0D4D TEMPS

XOR	A	Сброс А в +00.
LD	HL,(ATTR-P)	Текущие значения ATTR-P и MASK-P
BIT	0,(TV-FLAG)	извлекаются.
JP	Z,0D5B,TEMPS-1	Переход вперед, если обработка

LD	H,A	основной части экрана.
LD	L,(BORDCR)	В противном случае использовать +00
0D58	TEMPS-1	и заменить значение в BORDCR.
LD	(ATTR-T),HL	установить ATTR-P и MASK-P
LD	HL,+5C91	следующим рассматривается P-FLAG.
JR	NZ,0065,TEMPS-2	Это P-FLAG.
LD	A,(HL)	Переход вперед, если имеем дело с
RRCA		нижней частью экрана (A = +00).
		В противном случае извлечь
		значение P-FLAG и
		сдвинуть нечетные биты в четные
		биты.
0D65	TEMPS-2	
XOR	(HL)	
LD	(HL),A	
RET		

Программа "Команда CLS" (Очистка дисплея)

В первый момент очищается весь дисплей - все пиксели сбрасываются и байты атрибутов устанавливаются равными значению в ATTR-P- затем преобразуется часть дисплея.

0D6B	CLS	
CALL	0DAF,CL-ALL	Очистка всего дисплея.
0D6E	CLS-LOWER	
LD	HL,+5C3C	Это флаг TV(TV-FLAG).
RES	5,(HL)	Сигнал "Не очищать нижнюю
		часть экрана после нажатия
		клавиши".
SET	0,(HL)	Сигнал "Нижняя часть".
CALL	0D40,TEMPS	Использовать постоянные
		значения, т.е. ATTR-T копируется
		с BORDCR (клавиатуры)
LD	B,(DF-SZ)	Нижняя часть экрана теперь очи-

CALL 0E44,CL-LINE щается с этими значениями.

За исключением байтов атрибутов для строк "22" и "23" байты атрибутов для строк в нижней части дисплея необходимо сделать равными ATTR-P.

LD HL,+5AC0 Байт атрибутов в начале строки "22".

LD A,(ATTR-P) Извлечь ATTR-P.

DEC B Счетчик строк.

JR 0D8E,CLS-3 Переход вперед в цикл.

0D87 CLS-1

LD C,+20; +20 символов на строку.

0D89 CLS-2

DEC HL Возврат вдоль строки с установкой

LD (HL),A байтов атрибутов.

DEC C

JR NZ,0D89,CLS-2

0D8E CLS-3

DJNZ 0D87,CLS-1 Повторение цикла до завершения.

Теперь может быть зафиксирован размер нижней части дисплея.

LD (DF-SZ)+02 Размер будет равен двум строкам.

Теперь оставим полученный результат для последующего выполнения задач "Общего назначения".

0D94 CL-CHAN

LD A,+FD Открыть канал "К".

CALL 1601,CHAN-OPEN

LD HL,(CURCHL) Извлечь адрес текущего канала и

LD DE,+09F4 задать адрес вывода

+09F4(PRINT-OUT) и адрес

AND A входа +10AB(KEY-INPUT) (Вывод на принтер, ввод с клавиатуры)

0DA0 CL-CHAN-A

LD (HL),E

INC HL

LD (HL),D

INC HL

LD DE,+10AB
 CCF
 JR C,0DA0,CL-CHAN-A Сначала адрес вывода, потом - ввода.
 LD BC,+1721 При обработке нижней части дисплея
 "Нижняя строка печати" будет строка "23"
 JR 0DD9,CL-SET Возврат через CL-SET.

Подпрограмма "Очистка всей области дисплея"
(CLEARING THE WHOLE DISPLAY AREA)

Эта подпрограмма вызывается:

- 1 - из программы CLS (Очистка экрана);
- 2 - из основной программы исполнения;
- 3 - из программы автоматического листинга.

0DAF CL-ALL

LD HL,+0000 Системная переменная COORDS
 LD (COORDS),HL сбрасывается в нуль.
 RES 0,(FLAGS2) Сигнал "Экран чист".
 CALL 0D94,CL-CHAN Выполнить задачи общего
 назначения.
 LD A,+FE Открыть канал "S".
 CALL 1601,CHAN-OPEN
 CALL 0D4D,TEMPS Использовать "Постоянные.
 значения".
 LD B,+18 Теперь "Очистить" 24 линии
 дисплея.
 CALL 0E44,CL-LINE
 LD HL,(CURCHL) Учесть, что текущий адрес
 выхода равен 09F4
 LD DE,+09F4
 LD (HL),E (PRINT-OUT)
 INC HL
 LD (HL),D
 LD (SCR-CT),+01
 LD BC,+1821 Предустанов, счетчика
 перемещения (роллинга) по мере
 того, как верхняя часть дисплея

обрабатывается "Верхней строкой печати" будет строка "0".

Продолжение в CL-SET.

Подпрограмма "CL-SET"

При вызове подпрограммы регистровая пара BC содержит номера строк и колонки области символов или регистр C содержит номер колонки внутри буфера принтера, затем определяется соответствующий адрес первого бита символа. Подпрограмма возвращается через PO-STORE для того, чтобы сохранить значения необходимых системных переменных.

0DD9	CL-SET	
LD	HL,+5B00	Начало буфера принтера.
BIT	1,(FLAGS)	Переход вперед, если обработка
JR	NZ,0DF4,CL-SET-2	буфера принтера.
LD	A,B	Переместить номер строки.
BIT	0,(TV-FLAG)	Переход вперед, если обраба-
JR	Z,0DEE,CL-SET-1	тывается основная часть
		дисплея.
ADD	A,(DF-SZ)	Верхняя строка нижней части
SUB	+18	дисплея называется "Строкой
		+18" и она должна быть
		преобразована.
0DEE	CL-SET-1	
PUSH	BC	Спасаются номера строки и
		колонки.
LD	B,A	Номер строки переносится.
CALL	0E9B,CL-ADDR	В HL формируется адрес начала
		строки.
POP	BC	Извлекаются назад номера
		строки и колонки.
0DF4	CL-SET-2	
LD	A,+21	Номер колонки теперь
		преобразуется и переносится
SUB	C	в регистровую пару DE.

LD	E,A	
LD	D,+00.	
ADD	HL,DE	Формируется необходимый адрес: адрес и номера
JP	0ADC,PO-STORE	строки и колонки сохраняются посредством перехода в PO-STORE.

Подпрограмма "Вращение" (SCROLLING)

При вызове основной подпрограммы в регистре В содержится число строк дисплея, которые должны "Вращаться".

0DFE CL-SC-ALL

LD	B,+17	Точка входа после "SCROLL?"
----	-------	-----------------------------

Основная точка входа - из вышеописанных программ и когда осуществляется вращение для INPUT...AT.

0E00 CL-SCROLL

CALL	0E9B,CL-ADDR	Найти начальный адрес строки.
LD	C,+08	Это восемь пиксельных строк в заполненной строке.

Теперь. вход в основной цикл вращения. Регистр В содержит номер верхней строки, которая должна вращаться. Регистровая пара HL - начальный адрес этой строки в дисплейном файле и регистр С - счетчик строки пикселей.

0E05 CL-SCR-1

PUSH	BC	Спасти оба счетчика.
PUSH	HL	Спасти начальный адрес.
LD	A,B	Переход вперед, если в настоящий момент не имеем дело с "третьей" дисплея.
AND	+07	

LD	A,B
----	-----

JR	NZ,0E19,CL-SCR-3
----	------------------

Пиксельные строки верхней строки "Трети" дисплея должны быть перенесены через 2К границы (каждая треть = 2К).

0E0D CL-SCR-2

EX	DE,HL	Результатом этого преобразования является то, что HL остается без изменений, а
LD	HL,+F8E0	
ADD	HL,DE	DE указывает на требуемое место назначения.
EX	DE,HL	
LD	BC,+0020	Это +20 символов.
DEC	A	Уменьшаем счетчик, пока работаем с одной строкой.
LDIR		Теперь переносим тридцать два байта.

Теперь можно вращать пиксельные строки внутри трети, регистр A на первом проходе содержит +01,+07,+09 -+0F или +11-+17.

0E19 CL-SCR-3

EX	DE,HL	DE опять используется как указатель требуемого места назначения.
LD	HL,+FFE0	
ADD	HL,DE,	На этот раз только на 32 ячейки ниже.
EX	DE,HL	
LD	B,A	Спасти номер строки в "B".
AND	+07	Теперь определить сколько символов осталось в "трети".
RRCA		
RRCA		
RRCA		
LD	C,A	Загрузить "Итог" в регистр C.
LD	A,B	Извлечь номер строки.
LD	B,+00	BC содержит "Итог" и пиксельная строка каждого символа "Вращается".
LDIR		
LD	B,+07	Теперь подготовим увеличение адреса для того, чтобы перейти через границу "Трети".
ADD	HL,BC	Увеличить HL на +0700.
AND	+F8	Возврат назад, если рассмотрена какая-либо строка.
JR	NZ,0E0D,CL-SCR-2	либо "Треть" (конец "Трети").

Теперь определим завершено ли восемь циклов - по одному на каждую пиксельную строку.

POP	HL	Извлечь первоначальный адрес.
-----	----	-------------------------------

INC	H	Адрес следующей пиксельной строки.
POP	BC	Извлечь счетчики.
DEC	C	Уменьшить счетчик пиксельных строк
JR	NZ,0E05,CL-SR-1	и возврат назад до тех пор, пока не будут сдвинуты все восемь линий.

Затем возвращаются байты атрибутов. Заметим, что регистр В все еще содержит число строк, которые должны вращаться, а регистр С - содержит нуль.

CALL	0E88,CL-ATTR	Определяется необходимый адрес в области атрибутов и число симво- лов в строках "В".
------	--------------	--

LD	HL,+FFE0	Сдвиг всех байтов атрибутов на
ADD	HL,DE	тридцать две ячейки.

EX	DE,HL
----	-------

LDIR		Вращаются байты атрибутов.
------	--	----------------------------

Очистка нижней строки дисплея.

LD	B,+01	В регистр В загружается +01 и переход в CL-LINE.
----	-------	---

СОДЕРЖАНИЕ

Введение	3
Дисассемблер, программы перезапуска и таблицы . . .	6
"Запуск"	6
Перезапуск по "ошибке"	6
Перезапуск "печать символа"	6
Перезапуск "получение символа"	7
Перезапуск "получение следующего символа"	7
Перезапуск THE "MAKE BC SPACES" (резервирование "BC" ячеек памяти)	7
Программа THE "MASKABLE INTERRUPT" ROUTINE" (Маскируемое прерывание)	7
Программа "ERROR-2" (Ошибка-2)	8
Программа "NON-MASKABLE INTERRUPT" (Немаскируемое прерывание)	9
Подпрограмма "CH-ADD+1"	9
Подпрограмма "Переходов (SKIP-OVER)"	10
Программы клавиатуры "THE KEYBOARD ROUTINES" . . .	11
Подпрограмма "Сканирования клавиатуры" (THE KEYBOARD SCANNING).	11
Подпрограмма "Клавиатура" (KEYBOARD)	13
Подпрограмма "Повторный набор" (REPEATING KEY SUBROUTINE)	15
Подпрограмма "K-TEST"	16
Подпрограмма "Декодирование клавиатуры"	17
Звуковые программы (THE LOUDSPEAKED ROUTINES) . . .	20
Подпрограмма "BEEPER".	21
Управляющая программа "BEEP".	23
Таблица "Полутон" (THE "SEMI-TON")	27
Подпрограмма "Имя программы" (Zx81)	27
Программы управления магнитофоном (THE CASSETTE HANDLING ROUTINES)	27
Подпрограмма "SA-BYTES"	28
Подпрограмма "SA-LD-RET".	32

Подпрограмма "LD-BYTES"	33
Подпрограммы "LD-EDGE-2" и "LD-EDGE-1"	37
Управляющие программы "Спасение, загрузка, проверка и склейка"	39
Управляющая программа "VERIFY" (Проверка).	49
Подпрограмма "LOAD A DATA BLOCK" (Загрузка блока данных).	51
Программа управления "Загрузка" (LOAD)	51
Программа управления "MERGE" (Склейка)	55
Подпрограмма "MERGE A LINE OR VARIABLE" (Склейка строки или переменной)	60
Подпрограмма управления "SAVE" (Спасение)	62
Сообщения магнитофона	63
Программы обработки экрана и принтера	
Программы вывода на печать "PRINT OUT"	63
Таблица управляющих символов.	64
Подпрограмма "Курсор влево"	65
Подпрограмма "Курсор вправо"	65
Подпрограмма "Возврат каретки".	66
Подпрограмма "Печать запятой".	66
Подпрограмма "Печать знака вопроса".	66
Программа "Управляющие символы с операндами"	67
Разрешенные к печати символьные коды.	69
Подпрограмма "Сохранение позиции" (POSITION- STOPE)	69
Подпрограмма "Извлечения позиции".	70
Подпрограмма "Печать какого-либо символа" (PRINT ANY CHARACTERS)	71
Подпрограмма "Печать всех символов" (PRINT ALL CHARACTERS).	73
Подпрограмма "Установка байта атрибутов" (SET ATTRIBUTE BYTE)	76
Подпрограмма "Печать сообщения" (THE MESSAGE PRINTING)	77
Подпрограмма "PO-SAVE"	78
Подпрограмма "Изучение таблицы"	

(THE TABLE SEARCH)	79
Подпрограмма "Проверка для роллинга"	
(TEST FOR SCROLL)	79
Программа "Временные цветовые характеристики" (TEMPORARY COLOUR ITEMS) . . .	84
Программа "Команда CLS" (Очистка дисплея)	85
Подпрограмма "Очистка всей области дисплея". . . .	87
Подпрограмма "CL-SET"	88
Подпрограмма "Вращение" (SCROLLING)	89

"От А до ZX"
часть 5
"ПЗУ часть I"

Фирма "ПРИСТ"

Подписано к печати 29.02.92

Формат 60x84 1/16 Бумага газетная

Объем 6.0 Печ. лист

Тираж 10.000 экз. Заказ N **483**

Малое предприятие "Форум"

105215, Москва, а/я 16

Отпечатано на ротапринте в типографии ОХО

Миннефтегазпрома СССР.

113035, Москва, набережная М.Тореза, д.26/1

PS
Print