



# **ZX SPECTRUM**

.....  
**СБОРНИК ОПИСАНИЙ**



## ВВЕДЕНИЕ

### Фирма "СИНКЛЕР РИСЕРЧ ЛИМИТЕД"

Фирму "СИНКЛЕР РИСЕРЧ ЛИМИТЕД" основал в 1979 г. ее президент сэр Клайв Синклер с целью изобретения, разработки и сбыта электронной продукции.

Основная деятельность фирмы лежит в сфере компьютеров, где она является признанным мировым лидером в области персональных микроЭВМ. Фирмой уже произведено более 4 млн. микроЭВМ. Ежемесячно выпускается 200000 машин.

Начав выпуск плоскоэкранный карманного телевизора, фирма внедрилась во второй главный рынок сбыта. Ведущие сейчас на фирме разработки включают компьютеры следующего поколения, периферийные устройства ЭВМ, а также продукцию широкого применения на твердотельной технике.

МЕТАЛАБ, прогрессивная научно-исследовательская база фирмы СИНКЛЕР, отведена сугубо под исследования широкого ряда идей для последующего их внедрения в перспективную продукцию по самому последнему слову техники.

База нашей фирмы находится в Кембридже, но конторы имеются также в Лондоне, Франкфурте, Сингапуре и в Бостоне, США. Кроме того, дополнительная научно-исследовательская лаборатория действует также в Винчестере, графство Гемпшир.

### Продукция фирмы

#### Общие сведения по компьютерам

Персональные микроЭВМ делают непосредственный вклад в образование школьников, помогая им понять принцип действия компьютеров и методику их программирования. При наличии подходящего программного обеспечения компьютер является бесценным вспомогательным средством изучения всех предметов школьной программы.

В домашних условиях компьютеры можно использовать для ведения домашнего бюджета, составления списков покупок, описи имущества, хобби, обработки текстов, а также для всех задач, вечно приносимых домой с работы.

Персональные компьютеры нашли широкое применение в учреждениях, начиная с управления базой данных и обработки текстовой информации и кончая графическим прогнозированием и оперированием данных.

#### ZX SPECTRUM.

Эта микроЭВМ, производство которой начато в 1982 г., пригодна для широкого применения в быту, на небольших предприятиях, а также в процессе обучения. Емкость определяемого пользователем ОЗУ составляет 48К байт (самая высокая для компьютеров этого класса). Технические возможности этой модели включают мощный язык Синклер-Бейсик в ПЗУ, цветную графику высокого разрешения, генератор



звукового сигнала, а также клавиатуру на 40 нажимных клавиш. Исключительно обширная номенклатура программного обеспечения включает более 5000 наименований. Периферийные устройства, техническая литература и журналы — все это помогло превратить эту модель в одну из наиболее отличительных микроЭВМ на мировом рынке.

## ZX SPECTRUM +

МикроЭВМ ZX SPECTRUM+, промышленное производство которой начато в 1984 г., имеет определяемую пользователем память емкостью 48 К, а также полную профессиональную клавиатуру типа пишущей машинки. В клавиатуру этой модели помимо стандартных клавиш модели ZX SPECTRUM внесено 17 дополнительных клавиш, а также пробельная клавиша. Модель ZX SPECTRUM+ обеспечивает полную совместимость по всей номенклатуре программного обеспечения и периферийного аппаратного обеспечения.

## QL

МикроЭВМ SINCLAIR QL, введенная в производство в 1984 г., предназначена для серьезного применения в быту, на небольшом производстве или в обучающем процессе. В основе этой модели лежит 32-битовый микропроцессор. Емкость ОЗУ — 128 К. В модель включено 2 внутренних микропривода, и каждый из них обеспечивает дополнительную энергозависимую память емкостью 100 К. Помимо сложного и весьма мощного нового языка Синклер-Бейсик, резидентного в ПЗУ, SINCLAIR QL поставляется в комплекте с собственным программным обеспечением, направленным на пользователя-неспециалиста, пользующегося микроЭВМ либо в быту, либо на рабочем месте. Комплект включает 4 программы: по обработке текста, по планированию, по оперированию данными и графики. Каждая из программ является лучшей в своей области.

## ZX 81.

Эта микроЭВМ появилась в продаже в 1981 г. В ее основу положена новаторская конструкция из 4-х микросхем. Емкость ПЗУ этой модели составляет 8 К с мощным языком Синклер-Бейсик. Модель исключительно портативна (габариты: 16x17x4 см и массой 350 г.). Миллионы пользователей по всему миру получили первые навыки работы на микроЭВМ именно на этой модели.

## ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА

Микропривод ZX для моделей SPECTRUM и SPECTRUM+ обеспечивает недорогое массовое запоминающее устройство с быстрым доступом. Микропривод хранит не менее 85 К байт данных на магнитной кассете, предоставляя преимущества работ на гибких дисках со значительным сокращением денежных затрат.

Интерфейс-1 ZX для моделей SPECTRUM и SPECTRUM+ управляет микроприводом и поддерживает до 8-ми устройств. Интерфейс-1 включает также интерфейс



RS 232, позволяя таким образом подключение микроЭВМ SPECTRUM к другим компьютерам и периферийным устройствам.

Интерфейс-2 ZX обеспечивает быстрозагружаемое ПЗУ, средства кассетного программного обеспечения и координатной ручки для микроЭВМ SPECTRUM и SPECTRUM+.

Программное обеспечение.

Фирма СИНКЛЕР РИСЕРЧ предлагает широкий ассортимент деловых, общеобразовательных и игровых программ, разработанных для фирмы рядом ведущих международных фирм, специализирующихся на разработке программного обеспечения.

Плоскоэкранный телевизор.

Исключительно портативный телевизионный приемник с плоским экраном 5 x 5 см, габаритами 14 x 9 x 3 см и весом 280 г. был введен в производство в 1983 г. Ключевыми элементами проектирования в этом телевизоре являются уникальная ЭЛТ и одинарная интегральная схема, обрабатывающая почти все схемные решения.

Миниатюрный телевизор обеспечивает качество изображения значительно выше своих соперников. Эта модель направлена на потребителя, часто находящегося в пути и желающего знать, что творится в мире или же в спортивных событиях — будь то дома, на работе, автомобиле, на борту парохода или самолета.

## Краткое описание языка BASIC.

Персональный компьютер ZX SPECTRUM и аналогичный ему ZX SPECTRUM PLUS в стандартном исполнении обеспечиваются интерпретатором языка BASIC, хранящимся в ПЗУ. Эта версия BASIC является ограниченным подмножеством языка BASIC MICROSOFT с графическим расширением. Описываемая версия интерпретатора поддерживает внешнюю память на магнитной ленте.

Важной задачей при работе с этим компьютером является ввод текста программы. Каждая клавиша может действовать в одном из 5-ти режимов, обеспечивая ввод ключевых слов БЕЙСИКа, отдельных символов и знаков, а также управляющих слов. Различие между ZX SPECTRUM и ZX SPECTRUM+ состоит в отличии клавиатуры, дополненной в последнем отдельными клавишами для наиболее употребительных управляющих символов.

## ОПИСАНИЕ МАССИВОВ:

Массивы могут иметь произвольную размерность и минимальное значение индексов 1. В строковых массивах последний индекс определяет максимальную длину текста, хранящегося символ за символом в очередных элементах массива. Возможно повторное описание массива с ранее использованным идентификатором.



## ИДЕНТИФИКАТОРЫ:

Идентификатором является последовательность символов, состоящая из букв и цифр и начинающаяся с буквы. Прописные и строчные буквы не различаются. В зависимости от типа переменной или функции идентификаторы:

- числовых переменных могут иметь произвольную форму;
- строковых переменных могут быть только односимвольными;
- управляющих переменных — только односимвольными;
- числовых и строковых массивов — только односимвольными.

Совпадающие идентификаторы переменных, массивов и функций обозначают разные объекты. Ключевые слова могут быть частью идентификатора.

## КЛЮЧЕВЫЕ СЛОВА, ВВОДИМЫЕ КАК ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, ИНТЕРПРЕТАТОРОМ НЕ РАСПОЗНАЮТСЯ!!!

## ГРАФИЧЕСКИЙ ЭКРАН:

Графический экран тождествен текстовому экрану и в графическом режиме разбивается на 256\*176 точек 2-х цветов (цвета символа и цвета окружения) из доступных 8-ми цветов. Координаты могут изменяться в пределах:

0...255 — по оси X,

0...175 — по оси Y.

Начало координат расположено в левом нижнем краю в 22-й строке экрана.

## ТЕКСТОВЫЙ ЭКРАН:

Текстовый экран тождествен графическому экрану и в текстовом режиме разбивается на 24 строки по 32 символа. Верхние 22 строки предназначены для отображения введенного текста. Позиция символа может находиться в пределах 0...21 для номера строки и 0...31 для номера столбца. Начальная позиция курсора вывода находится в левом верхнем углу экрана. Возможно программное позиционирование курсора вывода на экране с использованием оператора AT с номерами строки и столбца. Две нижние строки предназначены для отображения вводимого текста. Эта область может расширяться снизу вверх вплоть до занятия 22-х строк на экране. Каждый символ может иметь один из 8-ми цветов для знака и один из 8-ми цветов для знакоместа (окружения). Кроме того, каждый символ может мигать, иметь повышенную яркость и отображаться негативно.

## ИДЕНТИФИКАЦИЯ ФАЙЛА:

Файлы на магнитной ленте идентифицируются именами, состоящими не более чем из 10 произвольных символов. При загрузке содержимого файла пустое имя ("") относится к произвольному файлу, считываемому с магнитной ленты.



## КЛАВИАТУРА:

Клавиатура выполнена в стандарте QWERTY и дополнена управляющими клавишами, изменяющими функции основных клавиш. В ZX SPECTRUM+ сохраниены стандартная клавиатура и управляющие клавиши, а также клавиши (.), (.), ("), (:) — это ускоряет набор текста программы. О том, в каком режиме работают клавиши, информирует курсор на экране:

К — ввод цифр номера строки и ключевых слов, начинающих команду; этот курсор появляется в начале строки, после ввода слова THEN или символа (:);

L — ввод строчных букв и цифр;

C — ввод прописных букв и цифр;

G — ввод псевдографических символов, указанных на цифровых клавишах, а также знаков, определенных пользователем для клавиш "A"... "U";

E — ввод символов и слов, обозначенных вне клавиш.

Использование функций клавиш иллюстрирует рис. 1, а переключение режимов — рис. 2

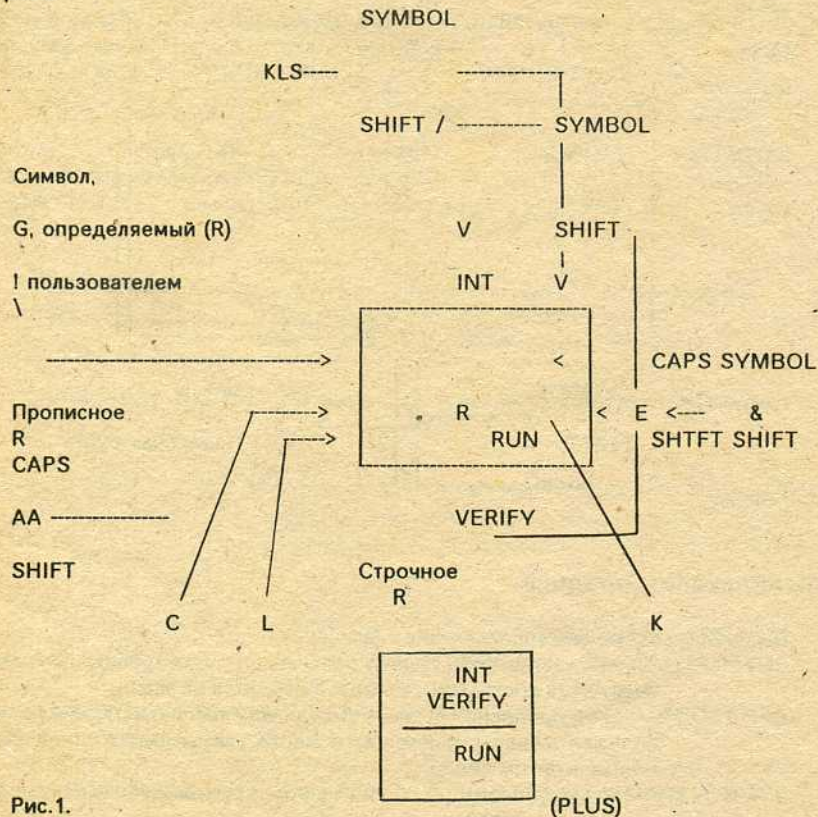


Рис. 1.

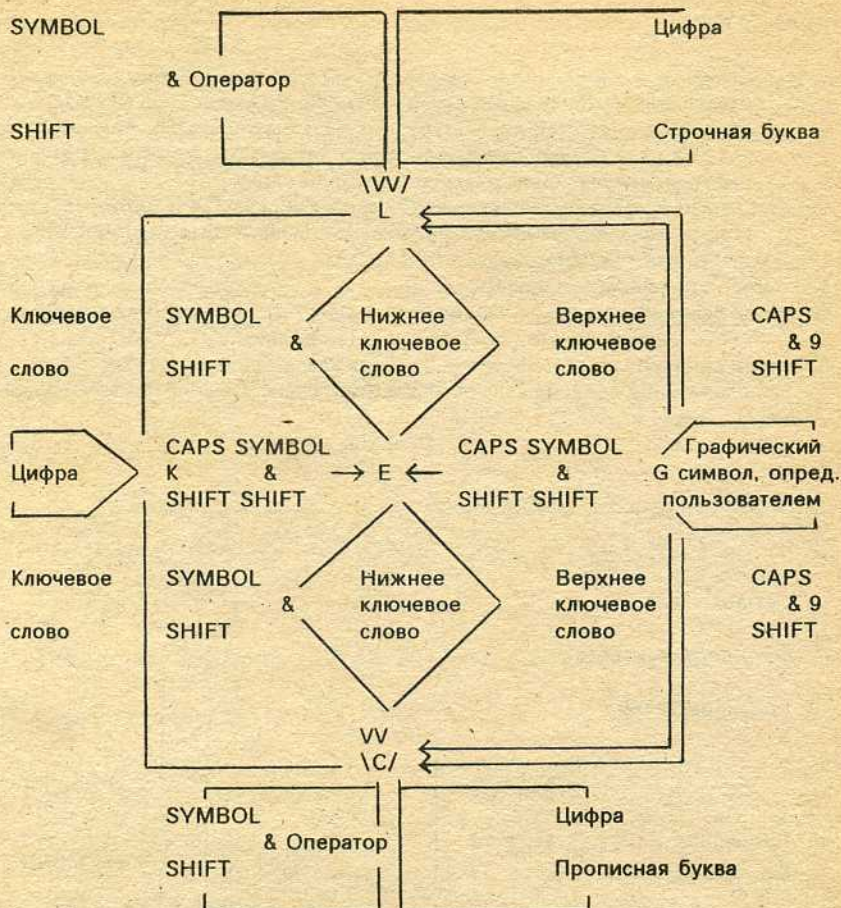


Рис. 2.

## УПРАВЛЯЮЩИЕ КЛАВИШИ.

Выделены следующие управляющие клавиши:

ENTER — окончание ввода либо редактирования команды; требование вывода фрагмента программы, которая выводится на экран.

CAPS SHIFT — ввод прописной буквы, команда управления, негативного символа или псевдографического символа, указываемого одновременно нажатой клавишей.

SYMBOL SHIFT — ввод оператора, обозначенного красным цветом на одновре-



Остальные команды управления вводятся путем одновременного нажатия управляющей и обычной клавиш.

CAPS SHIFT & 1 — вызов строки программы, отмеченной на экране EDIT (+) знаком "+", для редактирования;

CAPS SHIFT & SYMBOL SHIFT — переключение в режим E расширенного ввода дополнительных символов или возврата из этого режима в режим L или C.



В режиме E возможен ввод следующих команд управления и специальных символов:

- 0...7 — установить цвет вводимых символов, указанный над соответствующей клавишей;
- 8 — установить нормальную яркость вводимых символов;
- 9 — установить повышенную яркость для вводимых символов;
- ЛИТЕРА — ввод символа, обозначенного над клавишей литеры;
- SIMBOL SHIFT & ЛИТЕРА — ввод символа, обозначенного под клавишей литеры;
- SIMBOL SHIFT & ЦИФРА — ввод символа, обозначенного под клавишей цифры.

## СТРОКА ПРОГРАММЫ.

В строке теоретически может быть произвольное число символов, практически — до 255, поскольку ввод последующих символов замедляется. Номер строки может быть в пределах 1...9999.

## ОПЕРАТОРЫ.

- Арифметические: +, -, \*, /, \*\*, BIN, AND, OR (двоичное число, И, ИЛИ).
- Отношения: =, <, >, <=, >=, <>.
- Логические: NOT, OR, AND, (НЕ, И, ИЛИ).
- Строковые: + (номер позиции TO номер позиции) AND и отношения.

Одноаргументный оператор BIN с аргументом "двоичное число" осуществляет преобразование из двоичной записи в целое число. Оператор TO вырезает фрагмент текста из строковой переменной. Операторы AND и OR также применяются по-разному для числовых и строковых переменных:

- A AND B — если  $B \neq 0$ , то A, иначе 0;
- T\$ AND B — если  $B \neq 0$ , то T\$, иначе "";
- A OR B — если  $B \neq 0$ , то 1, иначе A.

## ОПИСАНИЕ ДИАПАЗОНА НОМЕРОВ СТРОК:

Диапазон номеров строк, задающий фрагмент программы, относится только к команде LIST (вывести на экран) и задается следующим образом:

ПУСТО — вся программа;

НОМЕР СТРОКИ — фрагмент, начиная с данной строки и до конца программы.

## ОПИСАНИЕ СИМВОЛА:

Для каждой из клавиш в пределах "A"... "U" для вывода в графическом режиме пользователем может быть определено новое начертание символа путем задания значений 8 последовательных значений байтов раstra. Байты должны быть размещены в области памяти, приписанной данному символу с помощью функции  $USR\ T\$+I$ , где  $T\$="A"... "U"$ ,  $I=0,1,...$



## ПАЛИТРА ЦВЕТОВ:

Для цветного телевизора (система PAL) доступны 8 цветов:

- 0 — черный, 2 — красный, 4 — зеленый, 6 — желтый,  
1 — синий, 3 — фиолетовый, 5 — голубой, 7 — белый.

Номера цветов используются при задании цвета рамки, фона, символа и его окружения.

## РЕДАКТИРОВАНИЕ СТРОКИ ПРОГРАММЫ:

Отдельная строка программы во время ее ввода или строка, вызванная на редактирование нажатием клавиши EDIT(+), (эта клавиша вызывает на редактирование строку, помеченную знаком ">"); этот указатель может перемещаться на другие строки с помощью клавиш "!!" (+) и "/" (+), может редактироваться следующим образом:

\/ !!

- передвинуть курсор ввода в требуемое место строки с помощью клавиш "——>" (+) и "<——" (+);
- удалить ошибочные символы клавишей DELETE(+);
- вставить новые символы на место, указываемое курсором;
- завершить редактирование строки нажатием клавиши ENTER.

## ВИДЫ ФАЙЛОВ:

На магнитной ленте могут быть записаны файлы следующих типов (различаются по атрибутам при вводе команды SAVE):

- ПУСТО — текст программы в символьном виде;
- CODE AND, LENGTH — двоичное отображение области памяти, начиная с адреса ADR и длиной LENGTH;
- SCREEN\$ — двоичное отображение изображения, выведенного в текущий момент времени на экран;
- DATA имя массива — содержимое строкового или числового массива.

## КЛЮЧЕВОЕ СЛОВО:

Все ключевые слова языка вводятся как отдельные коды символов — следовательно, они отличаются от любого идентификатора, вводимого посимвольно.

## ШИРИНА ЗОНЫ ВЫВОДА:

Стандартная ширина зоны вывода на печати и на экране составляет 10 знаков.

## ТИП ПЕРЕМЕННОЙ И ФУНКЦИИ:

Различаются два типа переменных и функций:

- числовой (без символа "\$");
- \$ — строковый.



## ВЕЛИЧИНА ЧИСЛА:

Допустимы действительные значения чисел в диапазоне —E38... -4E-39,0,4E-39...E38. Точность представления — 9 значащих цифр в виде целого числа, десятичной дроби без десятичного множителя или с десятичным множителем, идентифицируемым литерой E.

## ЛОГИЧЕСКИЕ ВЕЛИЧИНЫ:

Логические значения истина (1) или ложь (0) получаются в результате применения операторов отношения и сравнения.

## СТРОКОВЫЕ ПЕРЕМЕННЫЕ:

Тексты могут иметь произвольную длину или ограничиваться описанием строкового массива, в котором последнее значение размерности задает максимальную длину текста. Знаки (") могут располагаться в тексте с помощью их двукратного набора. Текстовая переменная без индекса обозначает строковую переменную с этим же идентификатором.

## ВЫЗОВ СИСТЕМЫ BASIC:

Система BASIC вызывается при каждом включении компьютера или при выполнении оператора NEW, при этом в последней строке экрана выводится текст:  
C 1982 SINCLAIR RESEARCH LTD.

В ZX SPECTRUM PLUS имеется клавиша RESET, находящаяся слева под клавиатурой. Нажатие на нее заменяет временное выключение-включение компьютера.

## ОКРУГЛЕНИЕ:

Действительное число округляется до ближайшего целого, не превышающего данное число.

# О П И С А Н И Е   К О М А Н Д

## РЕДАКТИРОВАНИЕ ПРОГРАММЫ:

номер строки — удаление строки;

LIST номер строки — вывести строки программы на экран;

LLIST номер строки — вывести строки программы на принтер.

## КОММЕНТАРИИ:

REM произвольный текст — комментарий.



## ЗАПУСК ПРОГРАММЫ:

RUN номер строки — исполнить программу;

CONTINUE — продолжить.

## ЗАГРУЗКА, ЗАПИСЬ ПРОГРАММЫ:

SAVE имя файла [LINE номер строки] — записать;  
LOAD имя файла — загрузить;  
VERIFY имя файла — проверить;  
MERGE имя файла — добавить.

## ОПРЕДЕЛЕНИЕ И ОПИСАНИЕ:

DIM имя массива (размер [, размер]...) — размер массива;  
DEF FN имя функции (параметр[, параметр]...) — определение функции.

## АРИФМЕТИЧЕСКИЕ ВЫЧИСЛЕНИЯ:

# FN имя арифметической функции (аргумент [, аргумент]...) — вызвать функцию;

# ABS арифметическое выражение — абсолютная величина;  
# ASC арифметическое выражение — код ASCII;  
# ASN арифметическое выражение — арксинус;  
# ATN арифметическое выражение — арктангенс;  
# COS арифметическое выражение — косинус;  
# EXP арифметическое выражение — экспонента;  
# INT арифметическое выражение — целая часть;  
# LN арифметическое выражение — натуральный логарифм;  
# PI арифметическое выражение — 3.1415926;  
# SGN арифметическое выражение — знак числа;  
# SIN арифметическое выражение — синус;  
# SQR арифметическое выражение — квадратный корень;  
# TAN арифметическое выражение — тангенс;  
# VAL строковое выражение — числовое значение записанного текста;  
# RND арифметическое выражение — случайное число;  
LET арифметическая переменная=арифметическое выражение — присвоить значение;  
RANDOMIZE [арифметическое выражение] — запуск генератора случайных чисел;

## ОБРАБОТКА ТЕКСТА:

# FN имя строковой функции (аргумент [, аргумент [аргумент]...]) — вызов функции;  
# CHR\$ код символа — знак ASCII с заданным кодом;



# CODE строковое выражение	— код знака ASCII;
# LEN строковое выражение	— длина строки;
# STR\$ арифметическое выражение	— строка, выводимая для данного аргумента;
# VAL\$ арифметическое выражение	— текстовое представление числа;
LET строковая переменная-строковое выражение	— присвоить значение.

#### УПРАВЛЕНИЕ ПРОГРАММОЙ:

STOP	— остановить выполнение программы;
GOTO номер строки	— перейти к строке номер строки;
IF логическое выражение THEN	
инструкция	— если... то... ;
FOR управляющая	
переменная=начальное	
значение TO конечное	
значение STEP шаг	— от... до ... с шагом ... ;
NEXT управляющая переменная	
	— следующий цикл;
GOSUB номер строки	— вызвать подпрограмму;
RETURN	— вернуться из подпрограммы.

#### ВВОД ДАННЫХ:

# INKEY\$	— ввести код нажатой клавиши;
DATA константа [константа]...	— данные;
INPUT [[текст приглашения]... переменная [переменная]...]	— ввести число;
INPUT LINE [текст приглашения]... строковая переменная	— ввести строку;
RESTORE [номер строки]	— перейти к началу данных;
READ переменная [,переменная]...	— прочитать данные.

#### ВЫВОД РЕЗУЛЬТАТОВ:

# ATTR (номер строки, номер столбца)	— атрибуты символов;
# SCREEN\$ (номер строки, номер столбца)	— параметр экрана;
CLS	— очистить экран;
PAUSE длительность	— пауза;
INK номер цвета	— цвет "чернил";
PAPER номер цвета	— цвет "бумаги";
BORDER номер цвета	— цвет "рамки";
FLASH параметр 018	— мигание;
BRIGHT параметр 018	— яркость;
OVER параметр 01	— наложение;
INVERSE параметр 01	— инверсия;
PRINT [список вывода]	— вывести на экран;



LPRINT[список вывода]

— вывести на принтер.

## СПИСОК ВЫВОДА:

Элемент списка вывода [разделитель элемент списка вывода]...

## РАЗДЕЛИТЕЛЬ:

- , — печать через пробелы;
- ; — печать подряд.

## ЭЛЕМЕНТ СПИСКА ВЫВОДА:

AT номер строки, номер столбца — печатать в строке, столбце;  
INK номер цвета  
PAPER номер цвета  
OVER параметр 01  
FLASH параметр 018  
BRIGHT параметр 018  
INVERSE параметр 01  
Арифметическое выражение  
Строковое выражение  
TAB номер столбца — номер позиции текста на экране.

## ГРАФИКА:

# POINT (координата X, координата Y)	— атрибуты точки;
# PLOT [OVER параметр 01:] [INVERSE параметр 01]	
координата X, координата Y	— установить точку;
DRAW координата X, координата Y, [угол]	— чертить отрезок;
CIRCLE координата X, координата Y, радиус	— чертить окружность;
COPY	— копировать экран на принтер;
SAVE имя файла SCREEN\$	— записать экран;
LOAD имя файла	— загрузить экран.

## РАБОТА С ФАЙЛОМ НА МАГНИТНОЙ ЛЕНТЕ:

SAVE имя файла DATA имя массива	— записать;
LOAD имя файла DATA имя массива	— загрузить;
VERIFY имя файла DATA имя массива	— проверить.

## СВЯЗЬ С ВНУТРЕННЕЙ ФОРМОЙ ПРЕДСТАВЛЕНИЯ ПРОГРАММЫ:

# PEEK адресное выражение	— взять байт по адресу;
# POKE адресное выражение, арифметическое выражение	
	— поместить байт по указанному адресу;



- # USR адресное выражение
- # USR строковое выражение — используется для адресации символов;
- # OUT номер порта, целочисленное выражение — вывести;
- GLEAR [ адресное выражение] — очистить;
- SAVE имя файла CODE адресное выражение, целочисленное выражение
- LOAD имя файла CODE адресное выражение, целочисленное выражение
- VERIFY имя файла CODE адресное выражение, целочисленное выражение

## ГЕНЕРАЦИЯ ЗВУКА:

BEEP длительность, высота тона — гудок.

## Примечание:

В тексте знаком "#" в начале строки отмечены стандартные функции языка BASIC.

## СООБЩЕНИЯ:

- 0 OK — безошибочное завершение команды или программы.
- 1 NEXT WITHOUT FOR — отсутствует установка начального значения управляющей переменной, указанной в инструкции NEXT.
- 2 VARIABLE NOT FOUND — использование переменной без присвоения ей ранее значения или отсутствие описания массива до использования индекса.
- 3 SUBSCRIPT WRONG — значение индекса элемента массива находится за пределами описанных размеров массива, либо несоответствие количества индексов и размерности массива.
- 4 OUT OF MEMORY — переполнение памяти.
- 5 OUT OF SCREEN — попытка вывести текст за пределы 22-й строки либо ввести более 23-х строк (в обоих случаях с использованием оператора AT).
- 6 NUMBER TOO BIG — результат вычисления превышает 1.0E38.
- 7 RETURN WITHOUT GOSUB — попытка возврата из подпрограммы без ее вызова.
- 8 END OF FILE — конец файла (для MIKRODRIVE).
- 9 STOP STATMENT — прерывание выполнения программы инструкцией STOP  
Продолжить работу можно командой CONTINUE.
- A INVALID ARGUMENT — недопустимый аргумент функции ASN, ACS, LN, SQR или USR ...
- B INTEGER OUT OF RANGE — невозможно преобразование действительного значения в целочисленное из-за превышения диапазона целых чисел.
- C NONSENSE IN BASIC — недопустимый аргумент функции VAL или VAL\$.
- D BREAK — CONT REPEATS — прервано выполнение инструкции ввода-вывода. Продолжить работу можно командой CONTINUE.
- E OUT OF DATA — не хватает данных для инструкции READ.



- F INVALID FILE NAME — в команде SAVE — недопустимое имя файла.
- G NO ROOM FOR LINE — переполнение памяти при вводе очередной строки программы.
- H STOP IN INPUT — в начале введенного текста обнаружена команда STOP — работа программы прерывается. Команда CONTINUE возобновляет работу программы с повторного выполнения инструкции INPUT.
- I FOR WITHOUT NEXT — отсутствует инструкция, завершающая цикл FOR.
- J INVALID I/O DEVICE — ошибка в операторе ввода-вывода.
- K INVALID COLOUR — недопустимый номер цвета.
- L BREAK INTO PROGRAM — выполнение программы прервано во время исполнения инструкции в строке с указанным номером. Продолжить работу можно командой CONTINUE.
- M RAMTOP NO GOOD — слишком велик или слишком мал адрес области памяти для системы BASIC.
- N STATEMENT LOST — переход к несуществующей инструкции.
- O INVALID STREAM — недопустимый поток для операции ввода-вывода (при наличии MIKRODRIVE).
- P FN WITHOUT DEF — вызов неопределенной функции.
- Q PARAMETR ERROR — неверное количество параметров или несоответствие типов при вызове функции.
- R TAPE LOADING ERROR — обнаружена ошибка при считывании или проверке содержимого программы или данных, записанных на магнитной ленте.

## КРАТКОЕ ОПИСАНИЕ ЯЗЫКА

### BETA BASIC

BETA BASIC дополняет BASIK ZX SPECTRUM 30-ю новыми командами и 21-ю новой функцией. Все старые команды и функции выполняются. Новые команды вызываются в режиме GRAPHICS.

### Описание команд BETA BASIC

- ALTER описание атрибутов TO описание атрибутов.[A] — позволяет быстро изменить атрибуты экрана (INK, FLASH, BRIGHT, PAPER) без необходимости очищения экрана. Например:  
 ALTER TO PAPER 1, INK 6 или  
 ALTER INK 7 TO PAPER 2, INK 0.
- AUTO номер строки, шаг [6] — включение автоматической нумерации строк, начиная с данной строки и с данным шагом (если задан), либо от текущей строки с шагом 10. Выключение — нажатие и удержание более 1-й секунды клавиши BREAK.
- BREAK [SHIFT-SPACE] — это не ключевое слово, но используется в режиме GRAPHICS. Оно может прерывать действие каждой программы, т. к. система



BETa BASIC работает в режиме INTERRUPT 2 микропроцессора Z80.  
CLOCK число или цепочка [C] — управляет 24-ми часами, которые выполняют различные функции, зависящие от поданного аргумента.

Аргумент	Переход на строку по ALARM	Звучание ALARMa	(высвечивание)
0	нет	выкл.	нет
1	нет	выкл.	есть
2	нет	установ.	нет
3	нет	установ.	есть
4	да	выкл.	нет
5	да	выкл.	есть
6	да	установ.	нет
7	да	установ.	есть

Пример:

CLOCK "09:29:04" — установка времени;  
CLOCK "A006:20" — установка будильника;  
CLOCK "A9000" — переход на строку.

CHR\$ 8 — курсор влево;  
CHR\$ 9 — курсор влево;  
CHR\$ 10 — курсор влево;  
CHR\$ 11 — курсор влево.

Эти знаки, примененные в цепи печати с помощью PRINT, изменяют позицию печати в соответствии с описанием.

DEF KEY — цепь однозначная; цепь SHIFT 1;

DEF KEY — цепь однозначная; инструкция: инструкция: ... подстановка под клавишу с данным знаком данной цепочки либо цепи инструкций.

DEF PROC название процедуры [1] — ключевое слово, начинающее определение процедуры, вызванной через название.

DELETE номер строки TO номер строки [7] — удаление строки программы в заданной области. Допускается использование различных вариантов, например:

DELETE TO 50;  
DELETE 10 TO 50;  
DELETE TO

и т. д.

DO

DO WHILE условие

DO UNTIL условие [D] — ключевые слова, начинающие определение петли. Петля заканчивается словом LOOP.

DROKE адрес, число [P] — загрузка 2-х очередей байтов памяти с данным адресом, числом из области 0...65535.

EDIT номер строки [0] — нормальный ввод. Позволяет редактировать произ-



вольную строку. Появляется после нажатия клавиши 0, если до этого была нажата клавиша ENTER.

ELSE инструкция [E] — это часть инструкции IF-THEN как альтернатива начальной инструкции. Пример использования:

IF условие THEN инструкция ELSE инструкция.

ENDPROC [3] — слово, оканчивающее определение процедуры, вызываемой через название.

EXIT IF условие [I] — условный выход из петли DO-LOOP.

FILL X, Y [F]

FILL цвет; X, Y [F]

FILL APER цвет; X, Y [F] — заполнение области фона цветом знака (если используется FILL или FILL INK) или области знака цветом фона (если используется FILL или FILL PAPER). Допускается также использование сложных конструкций, заполнение при этом начинается с текущих координат X, Y:

FILL INK 2; PAPER 1; FLASH 1; X, Y.

GET цифровая переменная или рядовая переменная [G] — присваивает переменной номер клавиши (с цифровой клавишей или 11 — для A, 12 — для B, 13 — для C и т. д.) или знак при нажатой клавише.

JOIN номер строки [SHIFT 6] — связывает строку с данным номером, либо текущую строку, если номера нет, со строкой, находящейся в нижней части экрана, придавая новой строке заданный номер (либо номер текущей строки).

KEYIN цепь-ряд [SHIFT 4] — допускается использовать только как инструкцию в программе. Вызывает вставку в программу поданной цепи (к примеру, это может быть строка программы).

KEYWORDS 1 [8]

KEYWORDS 0 [8] — переключатели ключевых слов BETA BASIC на графические знаки ZX SPECTRUM, получаемые в режиме G. Это слово никогда не выключается.

LIST номер строки TO номер строки (LLIST — также, но для принтера). Расширение синтаксиса нормального ключевого слова SINCLAIR BASICa.

LOOP

LOOP UNTIL условие [L]

LOOP WHILE условие [L] — заканчивающая часть структуры DO-LOOP.

ON [O] — применяется в структурах вида:

GOTO ON переменная; номер строки, номер строки, ... или

GOSUB ON переменная; номер строки, номер строки, ...

Производит переход к соответствующей строке в зависимости от переменной.

ON ERROR номер строки [N] — включение обслуживания ошибок через строку с заданным номером (к ней переходит программа при появлении ошибки при ее выполнении). Дополнительно переменной ERROR присваивается значение, равное коду ошибки. Не выполняется при кодах 0 (O. K.) и 9 (STOP). Выключение обслуживания ошибок наступает после перехода к программе обработки ошибок либо по команде ON ERROR 0.

PLOT X, Y; цепь — нормальный ввод. Позволяет рисовать в любом месте экрана.



Координаты относятся к левому верхнему краю первого знака цепи. В цепи можно использовать знаки управления курсором.

POKE адрес, цепь — ключевое слово, расширяющее синтаксис. Производит запись в память заданной цепи знаков, начиная с заданного адреса.

POP цифровая переменная [Q] — записывает адрес со стека GOSUB, DO-LOOP, PROC. Указанный этим адресом номер строки подставляется как значение переменной, если ее название было задано.

PROC название [2] — выполнение процедуры с заданным названием.

RENUM <начало TO конец> <новое начало> <STEP шаг> [4] — перенумерация строк в заданной области в соответствии с заданными параметрами. Если не был задан ни один параметр, то начальный номер строки принимает значение 10 и шаг равен 10.

ROLL код направления <, пиксель> <X, Y; ширина, высота> [R] — "переворот" определенного окна экрана. Образ, исчезающий с одной стороны, появляется с другой. Заданная ширина окна касается количества позиций знака, а не элементов образа. Координаты X и Y показывают левый верхний угол выбранного окна. Команда вызывает перемещение на один элемент. Если задан только код управления, то смещается целый экран. Можно переместить целый образ вместе с атрибутами, только атрибуты либо только часть образа. При повороте атрибутов заданное число должно быть равно 8.

Коды направления имеют следующие значения:

Коды направления	Направление	Охватывает
1	влево	атрибуты
2	вниз	атрибуты
3	вверх	атрибуты
4	вправо	атрибуты
5	влево	содержимое образа
6	вниз	содержимое образа
7	вверх	содержимое образа
8	вправо	содержимое образа
9	влево	атрибуты
10	вниз	атрибуты
11	вверх	атрибуты
12	вправо	атрибуты

SCROLL код направления, пиксель ;X, Y; ширина, высота [S] — перемещение содержимого определенного окна на один элемент. Синтаксис такой же, как и у ROLL. Здесь также можно не задавать ни одного параметра, и тогда наступит переход всего экрана на одну строку вверх (как в ZX-81). Содержание экрана пропадает на краю окна.

SORT таблица или цепь [M]



**SORT INVERSE** таблица или цепь [M] — упорядочивает знаковую таблицу или цепь в очередности кодов знаков или наоборот, а цифровую таблицу — от наибольшей цифры до наименьшей или наоборот. Числовая таблица может иметь не более 2-х измерений.

**SPLIT** — это не ключевое слово, вместо него применяется нормальный знак "<>" (SS/W). Он указывает, в каком месте программы строка может быть разделена на две части. Часть перед знаком присоединяется к программе; часть после знака остается в программе с тем же самым номером.

**TRACE** <номер строки> [T] — команда, приводящая к пошаговому выполнению программы. Ее подача вызывает переход GOSUB к заданной строке перед выполнением каждой инструкции программы. Выключение через RUN, CLEAR или TRACE 0.

**UNTIL** условие [K] — часть петли DO-LOOP, используемая в DO UNTIL или в LOOP UNTIL, позволяющая ее условное выполнение.

**USING** цепь-образец; число [U] — применяемое в инструкции PRINT USING, позволяет задавать соответствующий формат печатаемой цифры. Знак "\*" в цепи-образце означает предварительное разрежение.

**WHILE** условие [J] — часть петли DO LOOP, используемая в DO WHILE или в LOOP WHILE. Позволяет условное выполнение цикла.

## Описание функций BETA BASIC

Ниже дано описание функций, дополненных в системе BETA BASIC, вместе с описанием их аргументов и способов вызова.

**AND** (число, число) [FN A( )] — логическое произведение 2-х чисел, выполненное в двоичном представлении.

**BIN\$** (число) [FN B\$ ( )] — двоичное представление десятичного числа.

**CHAR\$** (число) [FN C& ( )] — преобразование целого числа без знака из области

0...65535 в соответствующее двузначное число.

**COSE** (число) [FN C( )] — косинус числа (4 значащие цифры, работает быстрее, чем в оригинале системы SPECTRUM).

**DEC** (число) [FN D( )] — значение 2-байтового числа, находящегося по заданному адресу.

**FILLED** () (число) [FN F( )] — количество элементов образа, заполненных перед последней командой FILL.

**HEX\$** (число) [FN H\$( )] — переводит шестнадцатичное число в десятичное.

**INSTRING** (старт, последовательность 1, последовательность 2) [FN I( )] — позиция первого знака последовательности 2 в последовательности 1 при просмотре последовательности от заданной стартовой позиции, либо 0, если элементы последовательности 2 не содержатся в последовательности 1.

**MEM** () [FN M( )] — количество свободных байтов памяти.

**MEMORY\$** () [FN M\$( )] — содержимое всей памяти (от 1 до 65532). Трактуются как цепь-последовательность.



MOD (число 1, число 2) [FN V ( )] — результат деления числа 1 по модулю числа 2.  
NUMBER (последовательность) [FN N ( )] — целое число без знака в пределах 0...65535, которое является соответствием двузначной последовательности.

OR (число 1, число 2) [FN O ( )] — логическая сумма 2-х чисел, представленных в двоичном виде.

RNDM (число) [FN R ( )] — псевдослучайное число от 0 до заданного числа включительно.

SCRN\$ (строка, столбец) [FN K\$ ( )] — знак, находящийся на заданной позиции (а также знак, определенный пользователем).

SINE (число) [FN S ( )] — синус данного числа (4 значащих цифры).

STRING\$ (число, цепь) [FN S\$ ( )] — повторение последовательности заданное число раз.

TIMES ( ) [FN T ( )] — прошедшее время, задаваемое через CLOCK.

USING\$ (цепь-образец, число) [FN U\$ ( )] — знаковая запись числа в заданном формате (как USING).

XOR (число 1, число 2) [FN X ( )] — логическая операция ИСКЛЮЧАЮЩЕЕ ИЛИ (EXCLUSIVE OR) двух двоичных чисел.

### Специальные переменные BETA BASIC

В системе BETA BASIC существуют определенные переменные, генерируемые через систему и доступные через названия. Ниже приведено их описание:

XOS, YOS — координаты середины системы расположения (в начале установлены на 0.0). Их можно изменять:

X=0...255

Y=0...175.

Обнуление через команды CLEAR и RUN.

XRG, YRG — диапазон координат на экране (XRG=256, YRG=176), которые можно свободно изменять. CLEAR и RUN задают начальные значения.

Во время исполнения команды ON ERROR или TRACE создаются специальные переменные:

ERROR — выдает код последней обнаруженной ошибки.

LINE — номер остановленной выполняемой строки (при TRACE) или строки, в которой была обнаружена ошибка (при ON ERROR).

STAT — номер остановленной выполняемой инструкции (при TRACE), либо инструкции, в которой была найдена ошибка (при ON ERROR).

Названия специальных переменных можно набирать как большими, так и маленькими буквами.

### RAM DOS

RAM DOS — квазиэлектронный диск, который позволяет быстро менять программы без помощи дополнительной аппаратуры. Она использует RAM как электронный диск при помощи новых BASIC-команд доступа к диску. Новые команды просты и похожи на стандартные команды доступа к магнитофону. RAM DOS не ограничивает количество записанных файлов, но ограничивает количество занятой



памяти (примерно 33 К), определяемой установкой RAMTOP. Каждый файл записывается на диск с заголовком, занимающим 17 байтов.

RAM DOS включается командой:

RAMD USR 63600

С этого момента все новые команды — активны, но не мешают выполнению старых команд. Вы можете использовать команду NEW без риска стереть файлы, т. к. они находятся под RAMTOP, который перед вводом программы с магнитофона устанавливается командой:

CLEAR 3200 (или другое, более подходящее значение).

Все новые команды вводятся одной из малых латинских букв, следующих за звездочкой:

- \*L — LOAD — вводится перед именем программы, которое не должно превышать 10 знаков. Выбор SCREEN, CODE — обычен. Если имя файла нуль (""), то вызывается последний введенный файл. Данные не доступны.
- \*S — SAVE — используется так же, но имя не может быть нулем. Остально данные не записываются (только вместе с программой). Если на диске уже имеется файл с таким именем, то он будет стерт.
- \*M — MERGE — работает только с программными файлами, как и с магнитофоном.
- \*D — DELETE — должно следовать за именем файла. Если оно пустое, то будет стерт последний файл. За именем указывается тип файла.
- \*C — CATALOG — без параметров. Печатает все имена файлов, а также количество оставшейся свободной памяти на RAM DOS.
- \*E — ERASE — очищает RAM-диск.
- \*T — TAPE — необходимо указывать имя файла. Записывает все содержимое RAM-диска вместе с программой RAM DOS как стандартный код. Перед вводом файла с магнитофона не забывайте ввести CLEAR 30000 или другое значение, затем введите LOAD "" CODE и затем для включения новых команд введите RAMD USR 63600.
- \*F — FREE — печатает число байтов в памяти BASIC ниже RAMTOP. Общее значение свободной памяти определяется сложением с величиной из \*C.
- \*X — DELETE — (BLOCK DELETE) стирает все линии BASIC от указанного номера до указанного. Эта команда не относится к RAM DOS, но она коротка и очень полезна.

#### СООБЩЕНИЯ:

##### OUT OF MEMORY

— нет места на RAM DOS для новых файлов. Если стирание старого файла не дает эффекта, попробуйте опустить ниже RAMTOP. Если такое сообщение встретится при вводе программы с магнитофона,



значит, не хватает места в памяти BASIC. Попробуйте поднять RAMTOP выше.

END OF FILE

— файл не найден.

Если вы хотите изменить атрибуты, введите их новое значение в ячейку 63714 (POKE 63714, XX).

## Стандартные программы работы с файлами

Для работы с файлами из программ, написанных на языке Ассемблера, рекомендуется пользоваться стандартными программами, находящимися в ПЗУ компьютера. Их десятичные адреса следующие:

1366 — подпрограмма чтения с ленты;

1218 — подпрограмма записи на ленту.

При вызове этих подпрограмм необходимо задать:

Регистр "A" — тип файла (0...255);

Регистр "DE" — длина файла;

Регистр "IX" — адрес начала загрузки файла.

Кроме того, для подпрограммы чтения с ленты необходимо задать тип работы (проверка или чтение) флажком переноса. Содержимое флажка переноса означает:

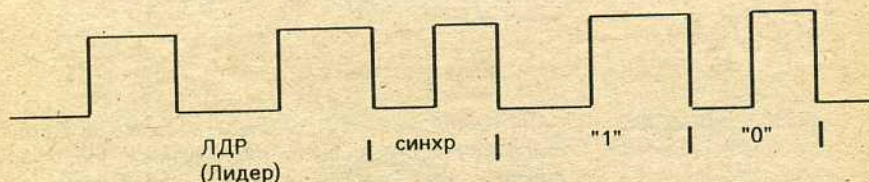
0 — режим ЧТЕНИЯ;

1 — режим ПРОВЕРКИ (т. е. сравнения содержимого памяти с информацией, считываемой с ленты).

Для работы с кассетами, записанными в формате ZX SPECTRUM на других компьютерах, необходимо помнить:

- 1) Для определения начала файла, сигнал ЛДР детектируется не менее 0,5 секунд для предупреждения сбоев.
- 2) Если длительность очередного полупериода ЛДР стала значительно меньше заданного, то весь цикл надо начать сначала.
- 3) Если длительность очередного полупериода ЛДР стала значительно больше заданного, то весь цикл надо начать сначала.
- 4) Проверку на окончание данных надо производить (даже если имеется счетчик считанных байтов) сравнением длины полного периода бита с указанными значениями (0.978 или 0.489 мс). Если счетчик слишком мал или слишком велик, то считается, что данные кончились.

## Временные диаграммы записи-считывания





Каждый блок информации (файл) на магнитной ленте начинается с синхронизирующего сигнала (ЛДР) длительностью 2 или 5 секунд. Частота этого сигнала — около 810 Гц (период 1.25 мс). После этого сигнала идет один период специального синхросигнала — длительность "нуля" — около 0.19 мс, "единицы" — 0.21 мс. Затем следуют байты данных, передаваемые последовательно, начиная со старшего бита. "Нулевой" бит передается одним периодом сигнала частотой около 2047 Гц (период — 0.489 мс). "Единичный" — одним периодом сигнала частотой около 1023 Гц (период — 0.978 мс).

Каждый файл состоит из байта типа (значение от 0 до 255), собственно данных и байта контрольной суммы. Байт типа и байт контрольной суммы не входят в длину файла. В контрольную сумму входит значение байта типа.

Байт типа обычно принимает значения 0 (для заголовков) и 255 (для данных). Если есть необходимость, пользователь может использовать и любой другой тип. От значения байта типа зависит длительность сигнала ЛДР. При значении байта типа от 0 до 7 длительность сигнала равна 5 секундам. Если байт типа больше 7, то длительность сигнала — 2 секунды.

Стандартный файл, формируемый компьютером по команде разгрузки, состоит из двух файлов: файла заголовка длиной 17 байт и файла данных.

## ФОРМАТ ФАЙЛА ЗАГОЛОВКА

Номера  
байтов

Назначение

1	<p>Тип информации, описываемый заголовком:</p> <p>0 — программа на Бейсике</p> <p>1 — числовой массив</p> <p>2 — символьный массив</p> <p>3 — "байтовый" файл: программа в машинных кодах или образ экрана</p>
2-11	Имя файла в кодах КОИ-7. Если имя отсутствует, то первый байт равен 255.
12-13	Длина файла данных.
14-15	Для программы — номер строки для автозапуска программы. Если первый байт равен 80 Н, то автозапуск не был задан. Для "байтового" файла — адрес загрузки для массивов — второй байт содержит имя массива (один символ КОИ-7).
16-17	Только для программы — длина программы на Бейсике.



# SINKLAIR RESEARCH LTD ZX LASER BASIC COMPILER CASIO SOFTWARE

Автор: Крис Неллинг.

## ВВЕДЕНИЕ

Компилятор языка LASER-BASIC предназначен для создания автономной программы, независимой от интерпретатора ЛАЗЕР-БЕЙСИКа.

Он может выполнять и обработку программ, написанных на стандартном БЕЙСИКе, но реального выигрыша во времени работы откомпилированных программ не представляет (в 2 раза и менее).

Компилированный код занимает меньший объем памяти, чем исходная программа, но при запуске он должен сопровождаться пакетом рабочих процедур. Любая программа, созданная в расширенном интерпретаторе ЛАЗЕР-БЕЙСИКа, поместится в компиляторе.

## 2. РАСПОЛОЖЕНИЕ ФАЙЛОВ НА ЛЕНТЕ.

A: COMPCODE ..... LOAD " " CODE  
LOADER ..... LOAD " " CODE  
RT CODE ..... LOAD " " RT CODE " CODE  
B: DEWO ..... LOAD " "

## 3. КОМПИЛЯЦИЯ.

При компиляции программы в памяти должны располагаться только текст исходной программы и компилятор. Следовательно, после ввода программы с помощью ЛАЗЕР-БЕЙСИКа и ее запуска программу следует записать на ленту, сбросить компьютер и загрузить в компилятор только свою программу без интерпретатора.

Программа, написанная на языке ЛАЗЕР-БЕЙСИК, не может редактироваться средствами СИНКЛЕР-БЕЙСИКа.

Перед загрузкой выполнить: CLEAR 59799. Загрузить компилятор: LOAD " " CODE или LOAD "COMPCODE" CODE.

Исходная программа может загружаться как до, так и после компилятора. Начало компиляции: RANDOMIZE USR 59800.

При компиляции экран отображает случайную информацию, что не является сбоем. После компиляции экран очищается и появляется сообщение: "OK", а также номер последней откомпилированной строки.

## 4. СТРУКТУРА ОТКОМПИЛИРОВАННОЙ ПРОГРАММЫ.

Скомпилированный код стирает исходный текст. Тем не менее компьютер воспринимает его как обычную БЕЙСИК-программу. Так, например, процедуры записи



на ленту и загрузки в память откомпилированного кода ни чем не отличаются от соответствующих операций для обычных БЕЙСИК-программ. Откомпилированный текст дальнейшим изменениям не подлежит. Нельзя также выполнять команду "MERGE". Если надо скомпилировать программу, сначала надо создать один исходный модуль (MERGE), а затем его откомпилировать.

После команды LIST на экране появляется несколько строк текста — первые две устанавливаются компилятором независимо от содержания исходной программы.

1-я строка (не выполняется): O PRINT O; содержит недоступную пользователю информацию для пакета рабочих процедур.

2-я строка (всегда): RANDOMIZE USR 59800. Это вызов откомпилированной программы. Если в исходной программе имеются функции, определяемые пользователем, то их список (подряд) помещается в 3-й строке с номером 0. Это не относится к назначениям процедур ЛАЗЕР-БЕЙСИКа, которые компилируются отдельно.

Если в исходной программе есть операторы DATA, то они все собираются в один оператор DATA и размещаются в 4-й строке, которая также нумеруется 0. Несмотря на такое объединение, оператор RESTORE в откомпилированной программе будет выполняться правильно.

## 5. ВЫПОЛНЕНИЕ ОТКОМПИЛИРОВАННОЙ ПРОГРАММЫ.

Для выполнения откомпилированной программы необходимо снабдить ее дополнительно пакетом рабочих процедур. RAMTOP следует установить в 59799 (о чем не следует забывать при загрузке откомпилированных модулей). Пакет рабочих процедур загружается командой SPRITES.

Если программа имеет файл SPRITES, его следует загрузить точно так же, как и в программе, выполненной на языке ЛАЗЕР-БЕЙСИК. Загрузка откомпилированной программы выполняется, как и для обычной программы, на БЕЙСИКе.

Инициализация выполнения программы: RANDOMIZE USR 59800 или GOTO 1.

### ПРИМЕЧАНИЕ:

Нажатие BREAK не прерывает выполнения программы, кроме режимов загрузки и выгрузки.

## 6. КОМПОНОВКА АВТОНОМНОЙ ПРОГРАММЫ.

Программу, написанную на ЛАЗЕР-БЕЙСИКе, необходимо соответственно настроить, чтобы для работы было достаточно набрать LOAD " ". Для этого программу необходимо скомпоновать и снабдить загрузчиком, который входит в набор, имеющийся на исходной ленте. Вот его текст:

```
10 CLEAR 59799
20 LET SPST=
30 LOAD "RT CODE" CODE
40 LOAD "SPRITES CODE" CODE SPST
50 LET T=INT (SPST/256): POKE 62464, SPST-256*T: POKE 62465, T
```



60 POKE 62466,255: POKE 62467, 220

70 POKE 56575,0

80 LOAD " "

Здесь:

Строка 10:	Выделение пространства для пакета рабочих процедур.
Строка 20:	Установка начала областей размещения файла SPRITES (см. описание генератора спрайтов).
Строка 30:	Загрузка пакетов рабочих процедур.
Строка 40:	Загрузка файла спрайтов.
Строка 50:	Занесение в программу адреса начала размещения области спрайтов.
Строка 60:	Занесение в программу адреса конца размещения области спрайтов.
Строка 70:	Занесение конца области размещения знакогенератора спрайтов.
Строка 80:	Загрузка текста скомпилированной программы.

#### ПРИМЕЧАНИЕ:

Если спрайты в программе не используются, то строки 20-70 можно исключить.

Вам остается ввести требуемое значение SPST и подготовить набор файлов на чистой ленте. Помните, что при выполнении команды SAVE для загрузчика и для откомпилированной программы необходимо указать строку автостарта.

#### 7. ОГРАНИЧЕНИЯ ДЛЯ СТАНДАРТНОГО БЕЙСИКА.

Ниже перечисленные значения были заложены в компилятор уже на этапе его проектирования:

- Нельзя использовать вычисляемые выражения в операторах: RUN, GOTO, GOSUB, RESTORE, так как на этапе компиляции все переменные будут установлены в 0, как и длина стринговых переменных.
- В операторе DIM допустимы вычисляемые границы, но только такие, которые в процессе компиляции будут установлены в 0. Следовательно, нельзя объявлять переменные вида DIM(N+12) — здесь значение выражения в скобках после компиляции будет равно 12, а в ходе выполнения программы перевычисляться не будет.
- Ограничений на число переопределений массива компилятор не накладывает, но надо учесть, что 1-е определение выполняется в ходе компиляции и оно не переносится в откомпилированную программу. Следовательно, если в программе есть переход к началу программы в точку, расположенную выше 1-го оператора DIM, то при 2-м проходе этой же части программы объявление массива может оказаться недействительным. Эту сложность можно обойти, поместив все описания массивов в самом начале программы до той точки, где начинаются ветвления.
- Использование оператора CLEAR не допускается.



- Команда RUN выполняется так же, как и в БЕЙСИКе, но сброса переменных нет.
- Недопустимы операторы: MERGE, CONTINUE, LIST, LLIST.
- Командой CONTINUE нельзя запускать сбившуюся программу или программу с оператором STOP.
- Команды OPEN и CLOSE работают только с потоками S и P. Подкачка и сброс файлов в режиме работы с микродрайвом не поддерживаются.
- Команды LOAD, SAVE, VERIFY являются допустимыми. Если загружается БЕЙСИК-программа, рабочие процедуры рассматривают ее как откомпилированную программу на ЛАЗЕР-БЕЙСИКе и попытаются выполнить ее, передав управление на первый оператор.

## 8. ИСПОЛЬЗОВАНИЕ КОМПИЛЯТОРА С МИКРОДРАЙВОМ.

Компилятор и пакет рабочих процедур можно использовать с микродрайвом.

Исключения:

В процессе выполнения компиляции программы обмен файлами с микродрайвом не поддерживается.

Все примеры, приведенные в данной инструкции, можно перенести на ZX-MICRODRIVE, соответствующим образом изменив команды ввода-вывода.

## 9. СООБЩЕНИЯ ОБ ОШИБКАХ.

- ILLEGAL STATMENT FOUND — встретился один из недопустимых операторов: CLEAR, MERGE, LIST, LLIST, CONTINUE.
- PROCEDURE DEFINITION NESTING ERROR — найдено два описания процедуры без промежуточного оператора RETURN. В описаниях процедур вложенность не допускается.
- RETURN WITHOUT DEF FN ERROR — найден оператор RETURN без соответствующего описания процедуры.
- PROCEDURE NOT FOUND — обращение к неописанной процедуре. Описание процедуры обязательно должно предшествовать ее использованию.
- PROGRAMM NOT COMPILED — это сообщение пакета рабочих процедур. При старте ее начало не было идентифицировано как начало откомпилированной программы.

В ходе выполнения программы могут также выполняться стандартные сообщения об ошибках, однако пакет рабочих процедур не в состоянии указать точку возникновения ошибки. Кроме того, следует игнорировать цифровые коды, сопровождающие сообщения об ошибках стандартного БЕЙСИКа.

Для того, чтобы быть полностью уверенным в работе откомпилированной программы, предварительно ее следует тщательно отладить и оттестировать перед компиляцией.

При возникновении ошибок в ходе выполнения программа или ее часть должна быть перекомпилирована, для чего необходимо повторно загрузить компилятор и исходную программу (предварительно исправленную).



## 10. КАРТА ПАМЯТИ.

Компилятор и пакет управляющих процедур загружается, начиная с адреса 59800. Пакет рабочих процедур занимает всю верхнюю часть памяти. Конец компилятора — по адресу 62586, т. е. его длина 3057 байтов. Следовательно, перед загрузкой этих модулей необходимо зарезервировать для них области памяти, соответствующие их начальным и конечным адресам.

Откомпилированная программа размещается в памяти следующим образом:

ПРОГРАММА-#	ПЕРЕМЕННЫЕ-#	СПРАЙТЫ-#	СТЕК Z80-#	RT CODE-#
PROG	VAR	ELINE	56575	59800
				65535

## 11. РЕКОМЕНДАЦИИ ПО ПРОГРАММИРОВАНИЮ В КОДАХ.

Между областью спрайтов и пакетом рабочих процедур имеется область памяти более 3 К. Обычно для стека процессора требуется менее 256 байтов, поэтому здесь можно размещать процедуры в машинных кодах, установив соответствующие значения в операторе CLEAR на конец файла спрайтов. Кроме того, можно в принципе отдать под процедуры пользователя всю область, начиная с ELINE и до адреса 59800.

# К Р А Т К О Е   О П И С А Н И Е   Я З Ы К А P A S K A L

## Описание редактора HP 80

### УПРАВЛЕНИЕ РЕДАКТОРОМ.

CAPS SHIFT 1 EDIT

— редактировать строку, выделенную экранным курсором.

CAPS SHIFT 2 CAPS LOCK

— фиксация регистра больших и малых символов.

CAPS SHIFT 3 TRUE VIDEO

— страница вперед.

CAPS SHIFT 4 INV VIDEO

— курсор влево/строка влево.

CAPS SHIFT 6 стрелка вниз

— экранный курсор вниз.

CAPS SHIFT 7 стрелка вверх

— экранный курсор вверх.

CAPS SHIFT 8 стрелка вправо

— курсор вправо/строка вправо.

CAPS SHIFT 9 GRAPHIC

— включить графический регистр.

CAPS SHIFT 0 DELETE

— удалить символ.

CAPS SHIFT Q

— перейти в начало текста.

CAPS SHIFT W

— перейти в конец текста.

CAPS SHIFT E

— установить режим редактирования.

CAPS SHIFT I

— установить режим ввода символов.

CAPS SHIFT A

— установить режим добавления символов.

CAPS SHIFT S

— найти образец.



CAPS SHIFT D

— удалить строку, выделенную экраным курсором.

## ВНИМАНИЕ!

Звуковой сигнал сообщает о нажатии недопустимой клавиши.

SIMBOL SHIFT CAPS SHIFT — запрос "COMMAND" (команда).

### ДОПУСТИМЫЕ ОТВЕТЫ:

4 OPEN#	— открыть блок текста.
5 CLOSE#	— закрыть блок текста.
6 MOVE	— перенести блок текста.
7 ERASE	— удалить блок текста.
0 FORMAT	— переключить формат экрана 64/32 и наоборот.
Z COPY	— копировать блок текста.
X CLEAR	— очистить текстовый буфер.
Y RETURN	— перейти к HI SOFT формату и вводить текст средствами БЕЙСИКа.
F	— задать литерал и установить режим поиска.
R	— задать литерал для замены.

## ВНИМАНИЕ!

Для выполнения прочих команд используйте стандартный редактор HP 80.

### КОМАНДА РЕЖИМА ПОИСКА.

S	— найти образец.
R	— заменить найденный образец на заданный литерал.
G	— перейти к замене остальных образцов.

## ВНИМАНИЕ!

Звуковой сигнал сообщает о нажатии недопустимой клавиши.

Не используйте графический регистр вне текста программы!

PASCAL не имеет встроенных графических процедур и функций!

Далее приводятся сведения, в основном касающиеся отличий от стандартного языка PASCAL.

### ОГРАНИЧЕНИЯ.

- тип FILE не реализован;
- тип RECORD не может иметь вариантыные поля;
- процедуры и функции не могут являться формальными аргументами;
- недопустимо использовать указатели на ранее не определенный тип.



## КОМПИЛЯЦИЯ И ИСПОЛНЕНИЕ:

Компиляция — C LINE NUMBER;

Остановка листинга — BREAK.

В случае успешной компиляции программы на запрос <RUN ?> ответить <Y>, если требуется исполнение программы. Возврат в редактор осуществляется нажатием любого другого символа.

При контроле правильности использования типов данная версия ПАСКАЛЯ использует эквивалентность имен, а не структур, поэтому переменные

VAR A: ARRAY [1...3] OF T и

B: ARRAY [1...3] OF T

не могут совместно применяться в выражении.

Присваивание A:=B также недопустимо.

## Синтаксис и семантика

Число без знака ::=

# шестнадцатиричное число;

Строковая константа записывается в кавычках " " и содержит не более 255 символов.

Символьная константа ::=

"CHR" ("константа") для управляющих символов CR, LF, NULL и т. д.

Ключевое слово <ASKED> игнорируется.

Тип множества может иметь не более 256 символов.

Пустой список в операторе <CASE> приводит к ошибке. Если подходящая ветвь в списке <CASE> не найдена, то управление передается оператору, следующему за <END>.

Управляющая переменная оператора <FOR> должна быть простой и не может являться параметром.

Оператор <GOTO> должен передавать управление внутри того блока, где он находится, и быть на одном уровне вложения с меткой.

Программа не имеет параметров.

## ПРЕДОПРЕДЕЛЕННЫЕ ИДЕНТИФИКАТОРЫ:

Константы: MAXINT [=32768],

TRUE,

FALSE.

Типы:

INTEGER — 2 байта;

REAL — 4 байта;

CHAR — расширенный ASCII (256) код;

BOOLEAN.

## ПРОЦЕДУРЫ И ФУНКЦИИ:

Ввод и вывод:

WRITE;

WRITELN.



— на дисплей  
или принтер:

CHR(8) — BS на дисплее (шаг назад)ж  
CHR (12) — очистка экрана (CLEAR или NEWPAGE);  
CHR (13) — CR и LF ("возврат каретки" и "перевод строки");  
CHR (16) — переключить вывод с дисплея на принтер и наоборот;  
READ;  
READLN.

— с клавиатуры: INCH.

Функция читает символ с клавиатуры и, если ни одна клавиша не нажата, возвращает CHR(0).

### АРИФМЕТИЧЕСКИЕ ФУНКЦИИ:

FRAS — дробная часть.

### ДРУГИЕ ПРОЦЕДУРЫ И ФУНКЦИИ:

---

Процедура исключена, вместо нее имеются процедуры:

- MARK;
- REALEASE.
- INLINE (C1, C2, C3...) — C1, C2, C3... шестнадцатиричные числа, размещаются как последовательность байтов;
- USER (V) — V — целое число. Функция USER (V) вызывает подпрограмму с адресом V, она должна завершаться инструкцией RET;
- HALT — завершает программу;
- POKE (X, Y) — помещает значение S в ячейку с адресом X;
- TOUT (NAME, START, SIZE) — записывает данные на магнитную ленту;
- TIN (NAME, START) — читает данные с магнитной ленты;
- OUT (P, C) — равносильно: BC:=P; A:=C; OUT (C), A;
- RANDOM — возвращает случайное число в диапазоне 0..255;
- ADDR — возвращает адрес аргумента произвольного типа;
- PEEK (X, R) — R — аргумент произвольного типа, значение которого заносится в ячейку с адресом X;
- SIZE (W) — возвращает размер памяти, занимаемой переменной W;
- INP (P) — аналогично OUT.



## КОММЕНТАРИИ:

"(\*)" и "{" "}"

Ключи располагаются внутри комментариев вслед за символом "\$" (знак денежной единицы). Состоят из знака "+" или "-", за которым следует один из символов:

- L — управление листингом при компиляции (по умолчанию +L);
- O — контроль переполнения (по умолчанию +O);
- S — контроль (по умолчанию +S);
- A — контроль границ массива (по умолчанию +A);
- I
- P

F — прочитать часть программы с магнитной ленты;  
позволяет экономить память.

## Стандартный редактор

- I M,N — режим автоматической нумерации строк программы;
- L N,K — распечатать на дисплее текст от строки N до строки K;
- D N,K — удалить часть текста от строки N до строки K;
- K N — при выдаче листинга распечатать без остановки N строк;
- M N,K — заменить строку с номером K на строку с номером N;
- F N,K,V,W — найти в диапазоне строк с номерами N,K образец V и заменить его на W;
- E N — редактировать строку с номером N.

## ПОДКОМАНДЫ:

- <пробел> — инкремент текстового указателя;
- DELETE — декремент текстового указателя;
- CAPS SHIFT/T — табуляция вправо;
- ENTER — конец редактирования;
- Q — отменить редактирование;
- R — начать редактирование заново;
- F — найти ранее заданный образец;
- S — заменить образец;
- I — вводить символы после курсора с автораздвижкой (завершается ENTER);
- X — курсор перемещается в конец строки;
- C — вводить символы с позиции курсора с уничтожением исходного текста;
- K — удалить символы под курсором;
- Z — удалить символы от курсора до конца строки.

## КОМАНДЫ ДЛЯ РАБОТЫ С МАГНИТОФОНОМ:

- P N,K NAME — текст со строки N до строки K записать в файл с именем NAME;



G NAME  
W N,M,S

- прочитать текст из файла с именем NAME;
- записывает текст в форме, которая потом может использоваться с ключом \$F N,M,S.

#### КОМПИЛЯЦИЯ И ВЫПОЛНЕНИЕ ПРОГРАММЫ:

C N  
R  
T

- компилировать текст, начиная со строки N;
  - исполнить скомпилированную программу;
  - отложенная программа компилируется таким образом, что перекрывает в памяти компилятор. В дальнейшем может быть записана на магнитофон и исполнена автономно;
- B — возврат в БЕЙСИК.

#### Список рекомендуемой литературы:

1. Йенсен К., Вирт Н. ПАСКАЛЬ. РУКОВОДСТВО ДЛЯ ПОЛЬЗОВАТЕЛЯ И ОПИСАНИЕ ЯЗЫКА. М. Финансы и статистика, 1982.
2. Грогоно П. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПАСКАЛЬ. М. Мир, 1982.
3. Вирт Н. АЛГОРИТМЫ + СТРУКТУРЫ ДАННЫХ = ПРОГРАММЫ. М. Мир, 1985.
4. Грэхэм П. ПРАКТИЧЕСКИЙ КУРС ЯЗЫКА ПАСКАЛЬ ДЛЯ МИКРОЭВМ. М. Радио и связь, 1986.
5. Findlay W., Watt D. A. PASCAL. AN INTRODAKTION TO METHODICAL PROGRAMMING. Thlrd edition. London: pltman, 1985.
6. Абрамов Б. Г., Трифонов Н. П., Трифонова Г. Н. ВВЕДЕНИЕ В ЯЗЫК ПАСКАЛЬ. М. Наука, 1988.\*

Графический редактор

ART STUDIO.

ВОЗМОЖНОСТИ РЕДАКТОРА.

1. Система работы с перекрывающимися меню (аналогично "Макинтошу").
2. Использование следующих инструментов:
  - карандаша с произвольно назначаемым сечением грифеля;



- кисти с произвольными размером и формой;
- краскораспылителя с регулируемым факелом краски;
- валика для заливки с различной текстурой (кирпич, рельеф и т. д.);
- лупы для рассматривания мелких деталей 2-х, 4-х и 8-кратной.

3. Работа с окнами: размножение, переносы, повороты и т. д.
4. Работа с изображением: растяжение, сплющивание, наложение и т. д.
5. Работа с примитивами: точки, линии, окружности, треугольники и т. д.
6. Режим резиновых нитей (все примитивы — резиновые).
7. Печать надписей любым шрифтом в любом направлении.
8. Операции с файлами на кассетах (дискетах).
9. Работа с цветом и атрибутами.
10. Отмена любого неправильного действия.

## SHAPES

POINT

ВЫБРАНО  
КУРСОР  
МЕНЮ

LINES

CONT. LINE

RECTANGLES

TRIANGLES

CIRCLES

RAYS

ELASTIK V — ВКЛ.

SHAP HRZ X — ВЫКЛ.

SHAP VRZ Y

## ПРИМИТИВЫ

ТОЧКА

1

ЛИНИЯ

1

ЛОМАНАЯ

1 2 3 4

ПРЯМОУГОЛЬНИКИ

1 2 3 4

ТРЕУГОЛЬНИКИ

1 2 3

ОКРУЖНОСТЬ

1 2

ЛУЧИ

1 2 3 4

РЕЖИМ

РЕЗИНОВЫХ НИТЕЙ

ДИСКРЕТНОСТЬ

КООРДИНАТ ПО "X"

ДИСКРЕТНОСТЬ

КООРДИНАТ ПО "Y"



— курсор примитивов (на схемах обозначен кружком). Цифры около кружков указывают последовательность ввода точек.

## UNDO

Отменяет любое последнее (!) действие, выполненное над экраном (в т. ч. операции в окнах). Для отмены подведите курсор меню к надписи "UNDO" и нажмите кнопку "FIRE" ("ВЗЯТЬ").

## WINDOWS

DIFINE WINDOW

LAST WINDOW

WHOLE SCREEN  
CLEAR WINDOW

CAT & PASTE  
WINDOW

CAT,  
CLEAR & PASTE

INVERT WINDOW

RE-SCALE WINDOW

CLEAR & RE-SCALE

FLIP HORIZONTAL


FLIP VERTICAL

ROTATE 1/4  
ROTATE 1/2  
ROTATE 1/4  
MERGE X

MULTIPLE X

## ОТМЕНА

## ОКНА

ОПРЕДЕЛИТЬ 1  2  
ОКНО 3

ПОСЛЕДНЕЕ (ТО ЖЕ)  
ОКНО

ВСЬ ЭКРАН ОКНА  
ОЧИСТКА ОКНА

ПЕРЕНЕСТИ и  
РАЗМНОЖИТЬ

ПЕРЕНЕСТИ и  
СТЕРЕТЬ СТАРОЕ

ИНВЕРТИРОВАТЬ ОКНО

ПЕРЕНЕСТИ и  
МАСШТАБИРОВАТЬ

МАСШТАБИРОВАТЬ и  
СТЕРЕТЬ СТАРОЕ

ГОРИЗОНТАЛЬНЫЙ  
ПЕРЕВОРОТ (ЗЕРКАЛО)

ВЕРТИКАЛЬНЫЙ  
ПЕРЕВОРОТ (ЗЕРКАЛО)

ПОВОРОТ на 90  
ПОВОРОТ на 180  
ПОВОРОТ на -90  
РЕЖИМ НАЛОЖЕНИЯ

РЕЖИМ  
МНОГОКРАТНЫХ ДЕЙСТВИЙ



Окна имеют прямоугольную форму. При масштабировании с уменьшением тонкие линии и мелкие детали исчезают, а при увеличении линии увеличивают свою толщину. Многократные действия выполняются от последнего окна.

## MAGNIFY

MAG. X 2

MAG. X 4

MAG. X 8

GRID X

## ЛУПА

УВЕЛИЧЕНИЕ X 2

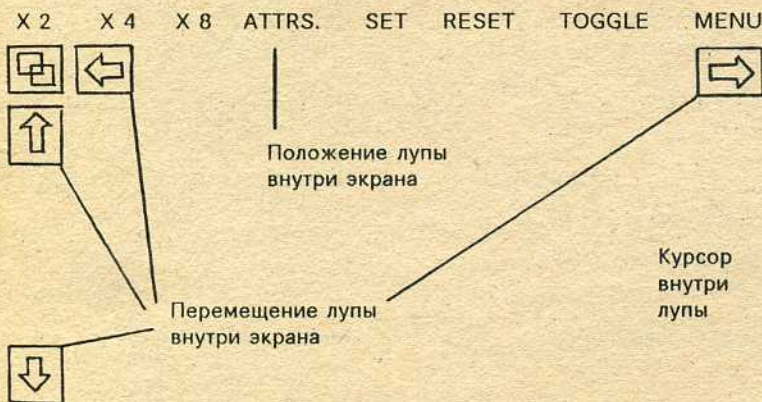
УВЕЛИЧЕНИЕ X 4

УВЕЛИЧЕНИЕ X 8

РЕЖИМ СЕТКИ



Курсор лупы



Поле лупы. В нем находится увеличенное изображение фрагмента экрана.

В режиме триггера 1-е нажатие "F" зажигает точку, 2-е — гасит (или наоборот, если точка уже была зажжена).

## MISCELLANEOUS

VIEW SCREEN

CLEAR SCREEN

BRIGHT GRID 1

BRIGHT GRID 2

REMOVE GRID

CHANGE COLOUR

VERSION NUMBER

## СЕРВИС — ОПЕРАЦИИ

ВСЁ ЭКРАН — НА ЭКРАН

ОЧИСТИТЬ ЭКРАН

ШАХМАТНОЕ ПОЛЕ 1X1 АТ

ШАХМАТНОЕ ПОЛЕ 2X2 АТ

СТЕРЕТЬ ШАХМАТНОЕ ПОЛЕ

НАЗНАЧИТЬ ЦВЕТ

НОМЕР ВЕРСИИ РЕДАКТОРА



Шахматное поле получается битами яркости (BRIGHT) в атрибутах и используется для совмещения точек и цветов на экране.

## АТТРИБУТЫ

SET INK

SET PAPER

SET BORDER

BRIGHT

FLASH

OVER X

INVERSE X

TRANSPARENT

STANDARD

Установка атрибутов действует на все функции (FILL, SHAPES и т. д.).

## А Т Р И Б У Т Ы

ЦВЕТ КАРАНДАША

ЦВЕТ БУМАГИ

ЦВЕТ БОРДЮРА

ЦВЕТ МАХ/MIN

МИГАЛКА ВКЛ./ВЫКЛ.

РЕЖИМ НАДПЕЧАТКИ

РЕЖИМ ИНВЕРСИИ

ВСЕ ПАРАМЕТРЫ — ТРАНСП.

ВСЕ ПАРАМЕТРЫ — СТАНДАРТ.

## PAINT

PEN

SPAY CAN

BRASH

EDIT BRASH

INVERSE

EDIT

BRASH

Выходят  
в режим  
инструментов

## КИСТИ

КАРАНДАШ

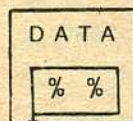
ПУЛЬВЕРИЗАТОР

КИСТЬ

РЕДАКТОР КИСТИ

РЕЖИМ ИНВЕРСИИ

## РЕДАКТИРОВАНИЕ КИСТИ



То, чем кисть  
красит.



То, чем кисть  
стирает.

1. Подведите курсор к нужной точке DATA или MASK.
2. Нажатием клавиши "ВЗЯТЬ" зажечь (погасить) точку.



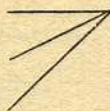
## FILL

SOLOD FILL

TEXTURED

WASH TEXTURED

EDIT TEXTURED



Выходит  
в меню  
текстуры.



"EDIT TEXTURED" похож на "EDIT BRUSH".

## ЗАЛИВКА

СПЛОШНАЯ ЗАЛИВКА

ТЕКСТУРНАЯ ЗАЛИВКА

ТЕКСТУРНАЯ РАЗМЫВКА

РЕДАКТОР ТЕКСТУРЫ

— Курсор заливки (валик)

## TEXT

LEFT TO RIGHT

DOWNWARDS

NORMAL HEIGHT

DOUBLE HEIGHT V/

TREBLE HEIGHT

NORMAL WIDTH

DOUBLE WIDTH V/

TREBLE WIDTH

SIDEWAYS X

BOLD V/

CAPS LOCK X

SHAP HRZ. X

SHAP VRZ. X

FOND EDITOR

## ТЕКСТ

ПЕЧАТАТЬ СЛЕВА НАПРАВО

ПЕЧАТАТЬ СВЕРХУ ВНИЗ

НОРМАЛЬНАЯ ВЫСОТА СИМВОЛОВ

ДВОЙНАЯ ВЫСОТА СИМВОЛОВ

ТРОЙНАЯ ВЫСОТА СИМВОЛОВ

НОРМАЛЬНАЯ ШИРИНА СИМВОЛОВ

ДВОЙНАЯ ШИРИНА СИМВОЛОВ

ТРОЙНАЯ ШИРИНА СИМВОЛОВ

БУКВЫ ПОВЕРНУТЬ НА 90

ЖИРНАЯ ПЕЧАТЬ

ТОЛЬКО ЗАГЛАВНЫЕ БУКВЫ

ДИСКРЕТНОСТЬ ПО ГОРИЗОНТАЛИ

ДИСКРЕТНОСТЬ ПО ВЕРТИКАЛИ

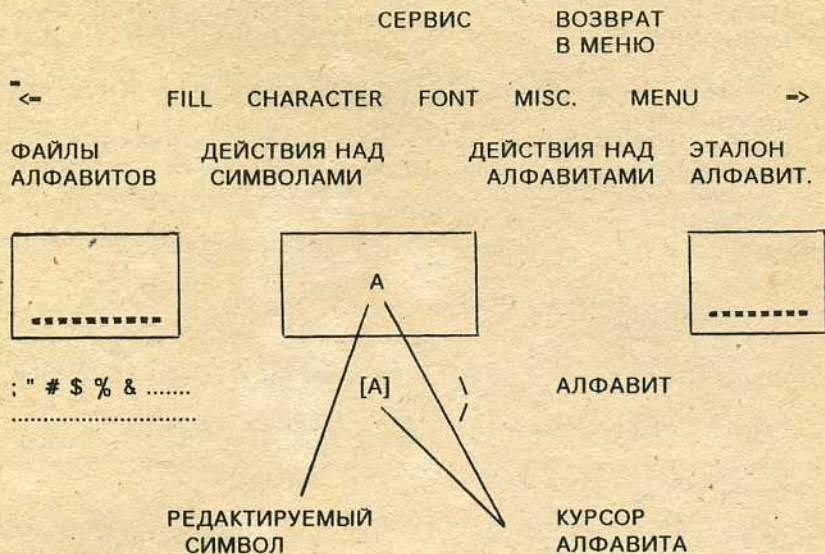
РЕДАКТОР СИМВОЛОВ

][ — курсор начала строки

— курсор символов



Курсор начала строки перемещается как любой другой курсор, но после нажатия клавиши "ВЗЯТЬ" исчезает, а на его месте появляется курсор ввода символов, после чего можно набивать любой текст с клавиатуры компьютера. После нажатия клавиши "ENT" ввод символов заканчивается, и на экране появляется курсор начала строки. "SHAP" совмещает символы с атрибутами.



#### CHARACTER

CLEAR

INVERT

FLIP HRZ.

FLIP VRT.

ROTATE 1/4

SCROLL RIGHT

SCROLL DOWN

#### ДЕЙСТВИЯ НАД СИМВОЛАМИ

СТИРАНИЕ СИМВОЛА

ИНВЕРСИЯ СИМВОЛА

ГОРИЗ. ЗЕРКАЛЬНЫЙ ПОВОРОТ

ВЕРТ. ЗЕРКАЛЬНЫЙ ПОВОРОТ

ПОВОРОТ НА 90 ГРАДУСОВ

РОЛЛИНГ ВПРАВО

РОЛЛИНГ ВНИЗ

Такое же меню для действий над всем алфавитом. Алфавит можно загружать с магнитной ленты.



## CASSETTE

## КАССЕТА

SAVE FILE  
LOAD FILE  
LOAD NEXT FILE  
VERIFY FILE...  
VERIFY NEXT FILE  
MERGE FILE ...  
MERGE NEXT FILE

ЗАПИСАТЬ ФАЙЛ...  
ЗАГРУЗИТЬ ФАЙЛ...  
ЗАГРУЗИТЬ СЛЕДУЮЩИЙ ФАЙЛ  
ПРОВЕРИТЬ ФАЙЛ...  
ПРОВЕРИТЬ СЛЕДУЮЩИЙ ФАЙЛ  
НАЛОЖИТЬ ФАЙЛ  
НАЛОЖИТЬ СЛЕДУЮЩИЙ ФАЙЛ

- Для выбора режима нажмите "FIRE" ("ВЗЯТЬ"), указав курсором меню на необ ходимый режим.
- На запрос "FILE NAME ?" введите с клавиатуры имя файла и нажмите "ENTER".
- Выше изображено меню операций над файлами изображений. Полученные файлы имеют формат "SCREEN" и могут загружаться в экран командой "LOAD" "SCREEN" из программ на языке "BASIC".
- Операции над файлами есть также в режиме "TEXT" (через меню "FONT EDITOR").

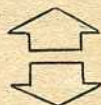
## ОСНОВНОЕ МЕНЮ.

ПЕЧАТЬ ФАЙЛЫ АТРИБУТЫ КИСТИ ЭКРАН ОТМЕНА РОЛЛИНГА

PRINT FILE ATTRS. PAINT MISC. UNDO

WINDOWS FILL MAGNIFY TEXT SHAPES

ОКНА ЗАКРАСКА ЛУПА ТЕКСТ ПРИМИТИВЫ РОЛЛИНГ  
ВАЛИКОМ ЭКРАНА



Это меню всегда (кроме особых случаев) находится вверху экрана. Под ним две строки "невидимы", и при попадании курсора в них его не видно. Если это произош- ло (курсор "пропал"), продолжайте перемещать курсор вверх (вниз), и он появится.

## ВСЕ УПРАВЛЕНИЕ КУРСОРОМ.

РУССКИЙ АНГЛ. KEYBOARDS SINCLAIR

ВВЕРХ UP Q

ВНИЗ DOWN A SINCLAIR  
JOYSTICK

ВЛЕВО LEFT O N 1

ВПРАВО RIGHT P

ДЕЙСТВИЕ FIRE ЛЮБАЯ КЛАВИША  
НИЖНЕГО РЯДА

РЕЖИМ  
ТЕКСТ  
КЛАВИША "ENT"

ЗАВЕРШАЕТ  
ПЕЧАТЬ СТРОКИ



# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ PC ZX SPECTRUM

Программа проектирования печатных плат "PROJ.PLYT".

Автор Б. Тобианских 1985.03.09.

## 1. Введение.

Программа позволяет практически полностью автоматизировать процесс проектирования рисунка проводников печатных плат (ПП) — как односторонних, так и двусторонних.

Максимальные размеры проектируемых ПП не превышают 16. 5\*16. 5 см, что вполне достаточно для большинства применений.

Программа значительно облегчает процесс проектирования, сокращая время разработки и делая работу более приятной. Несмотря на то, что данная программа предназначена для ZX SPECTRUM, большинство фаз проектирования ПП автор рассмотрел в больших компьютерных системах.

Как работает PROJ.PLYT.

Программа состоит из 2-х основных взаимозаменяемых блоков. 1-й блок, обозначенный WERSIA-1, обладает всеми возможностями программы, приведенной в ее каталоге под номером 1-15. 2-й блок WERSIA-2 не содержит подпрограмм для корректировки рисунка проводников с помощью курсора. Основным достоинством этого блока является высокое быстродействие.

В случае, когда изменения незначительны, рекомендуется использовать программу WERSIA-1, поскольку это не требует применения по защите данных на касете.

Ниже приведен перечень подпрограмм и описаны способы их применения.

- |                                  |                              |
|----------------------------------|------------------------------|
| 1) BIBLIOTEKA ELEMENTOW          | БИБЛИОТЕКА ЭЛЕМЕНТОВ (БЭ)    |
| 2) LIASTA ELEMETTOW              | ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ (ПЭ)      |
| 3) LIASTA POLASZEN               | ТАБЛИЦА СОЕДИНЕНИЙ (ТС)      |
| 4) ULOZENIE ELEMENTOW            | РАСПОЛОЖЕНИЕ ЭЛЕМЕНТОВ (РЭ)  |
| 5) USTALANIE ELEMENTOW OBWODOW   | ПРОВЕРКА СОЕДИНЕНИЙ (ПС)     |
| 6) SOTOWANIE DLUGOSCI POLAZEN    | СОТИРОВКА СОЕД. ПО ДЛИНЕ     |
| 7) TRASOWANIE                    | ТРАССИРОВКА                  |
| 8) WYPROWADZENIE LISTY POLAZEN   | ВЫВОД ТАБЛИЦЫ СОЕДИНЕНИЙ     |
| 9) SAVE                          |                              |
| 10) LOAD                         |                              |
| 11) WYPROWADZENIE LISTY POLASZEN | ВЫВОД ТАБЛИЦЫ<br>СОЕДИНЕНИЙ  |
| 12) WDRUK RYSUNKU PLYTKI         | ВЫВОД НА ПЕЧАТЬ РИСУНКА (ПП) |
| 13)                              |                              |
| 14) WYJSCIE DO BASICa            | ВЫХОД В BASIC                |

Рисунок проводников ПП на экране изображается на фоне точек, образующих растр с шагом 1/20 (0.125 мм). Шаг 1/20 является базовым для системы, поэтому все размеры элементов, а также их расположение должно быть кратны ему.

При создании новых баз данных работу надо начинать с инициализации, это вызовет удаление старых баз и занесение в счетчик номера 1.



При работе с отдельными подпрограммами на вопрос, появляющийся в последней строке, следует отвечать N (нет) или вводить данные в очередности, объявленной программой.

**ВНИМАНИЕ.** Следует безоговорочно придерживаться принципа, что возврат в систему возможен только через GOTO0. Иначе ранее введенные данные окажутся потерянными и потребуются их заново вводить в программу.

## 2.1. Библиотека элементов.

С помощью этой подпрограммы создается библиотека элементов, которые будут использованы в проектируемой плате. Пользователь может создать собственную библиотеку элементов, которую будет вводить к имеющейся при проектировании своих плат.

БЭ содержит информацию о конструкции каждого из элементов, т. е. его размеры и количество выводов. Эти данные кодируются приведенными ниже символами и связаны с ними соответствующими параметрами элементов.

HG — расстояние до следующего ряда контактных площадок (КП) выводов.

PN — количество между КП.

DZ — расстояние между КП.

XX — расстояние между выводом и краем корпуса.

YY — расстояние между выводом и другим краем корпуса.

XL — длина корпуса.

Следует помнить, что размеры должны быть кратны шагу 1/20. Ниже пример кодировки изменений, а размер элементов с расположением выводов, отличающимся от рядового, следует кодировать по расположению их КП в соответствии с программой 4.

Ниже приведена таблица кодировки размеров наиболее часто используемых элементов.

	1	2	3	4	5	6	7	8	9	10
HG	6	6	12	0	0	6	6	12	6	0
PN	8	7	20	32	1	9	6	12	4	2
DZ	2	2	2	2	0	2	2	2	2	6
XX	0	0	0	0	0	0	0	0	0	0
YY	0	0	0	0	0	0	0	0	0	0
XL	14	12	38	62	0	16	10	22	6	8
YL	6	6	12	0	0	6	6	12	6	0



## ХАРАКТЕРИСТИКА ЭЛЕМЕНТОВ, ПРИВЕДЕННЫХ В ТАБЛИЦЕ.

- |                            |  |
|----------------------------|--|
| 1. Микросхема 16-выв.      | 6. Микросхема 18-выв.  |
| 2. Микросхема 14-выв.      | 7. Микросхема 12-выв.  |
| 3. Микросхема 40-выв.      | 8. Микросхема 24-выв.  |
| 4. Один ряд разъема CANON. | 9. Микросхема 8-выв.   |
| 5. Одиночная КП.           | 10. Резистор с расстоянием между выводами $80 \times 1/20$ ( $80 \times 0.125-10\text{мм}$ ) |

**ВНИМАНИЕ.** В качестве 1-й позиции данного перечня всегда следует давать параметры, связанные с КП (отверстием). Любая позиция данного перечня может быть автоматически вызвана подпрограммой "8" при вводе дополнительных КП (отверстий). Перечень может содержать до 20 различных позиций.

### 2.2. Перечень элементов.

Перечень элементов содержит позиционные обозначения элементов в соответствии с принципиальной схемой; например: U21, R3 и т. д. и их номера из библиотеки элементов; характеризующие их размеры и расположение выводов. Вводимые обозначения должны содержать не более 100 позиций.

ПЭ автоматически дополняется координатами X и Y данного элемента во время работы подпрограммы "4" и подпрограммы "8" при вводе новых КП (отверстий).

### 2.3. Таблица соединений.

ТС содержит наименование (позиционное обозначение) элемента и номер соединяемого вывода, а также наименование и номер элемента, с которым осуществляется соединение.

ТС может содержать до 500 позиций. При необходимости содержание любой можно изменить с помощью процедуры "ROR". После ее вызова следует ввести номер изменяемой строки, а затем полное ее содержание. В ТС могут быть внесены дополнительные соединения с помощью подпрограммы "S". Эта подпрограмма позволяет также автоматически создавать новую ТС, содержащую сведения о невыполненных соединениях (отмеченных в исходной таблице знаком \*) для ее использования при проектировании второй стороны платы.

### 2.4. Размещение элементов.

Эта подпрограмма позволяет пользователю разместить элементы на поверхности платы. Операцию размещения можно произвести путем задания координат X и Y. После перемещения курсора в нужное место или определения координат X и Y следует ввести данные относительно поворота, обозначенного буквой D и принимающего значения от 0 до 3. Доступ к манипуляции данным элементом достигается путем нажатия клавиш 6 или 7 в зависимости от принятого направления считывания элементов. В последнем случае на экране появляется очередное обозначение выбранного, а также его данные: X, Y, D. Одновременно изображение выбранного элемента на экране начинает пульсировать.



Масштаб изображения рисунка печатных проводников можно изменять путем нажатия на клавишу Z и ввода цифры 1 или 2. Масштаб увеличения чертежа платы Z=2 можно использовать только в случае, когда ее размеры не превышают  $127 \times 1/20$  по оси X и  $84 \times 1/20$  по оси Y. Если ее размеры больше, то следует пользоваться масштабом уменьшения Z=1. В случае использования масштаба 2 и попытки размещения элемента за пределами поля, видимого на экране, программа расценит это как ошибку и заблокирует дальнейшую работу. В этой ситуации нужно перейти на уровень BASIC, ввести LET Z=1 и вернуться в программу по GOTO 0.

При размещении элементов используются следующие команды:

C — вызов курсора (треугольник) и перемещение его с помощью клавиш 5-8 (медленно) или CAPS SHIFT и 5-8 (быстро).

E — заканчивает локализацию элемента, после которого следует ввести параметр поворота D.

W — размещение элемента в соответствии с координатами X и Y. В этой системе программа требует очередные данные от X, Y, Z. Если не задавать параметры, то будут введены текущие значения.

B — возврат в основную программу.

После первоначального расположения элементов следует выполнить подпрограмму 5 и вернуться в программу 4 для вычерчивания прямых соединений. Полученный рисунок расположения элементов с прямыми линиями соединений необходимо тщательно проанализировать. Дальнейшая корректировка расположения элементов должна быть направлена на получение нескрещивающихся соединений минимальной длины. Эффективная реализация этого принципа имеет существенное значение для дальнейшей работы программы.

## 2.5. Проверка соединений.

Целью этой программы являются анализ таблицы соединений и определение цепей, находящихся под одним потенциалом. Программа выполняется в автоматическом режиме и может автоматически скорректировать таблицу соединений. Работа программы 3 заканчивается выводом сообщения об ошибке. После чего нужно вернуться в основную программу через GOTO 0.

После выполнения этой программы рекомендуется вернуться к подпрограмме 4 (расположение элементов) с вычерчиванием прямолинейных соединений проводников. Следует повторить анализ расположения элементов и при необходимости откорректировать его. В случае коррекции следует повторить подпрограмму 7.

## 2.6. Сортировка соединений по длине.

После выполнения подпрограммы порядок записи в таблице соединений обычно изменяется. Программа работает в автоматическом режиме и создает таблицу соединений, в которой соединения расположены по нарастанию длины.

Эта подпрограмма позволяет также сортировать по нарастанию значения DELTA X или DELTA Y. Сортировка по этим параметрам особенно полезна при проектировании двусторонних плат.

В этом случае нужно:

— после выполнения программы 5 произвести сортировку по длине (проц. 1);



— повторить сортировку по DELTA Y (проц. 2).

В итоге получается чертеж печатной платы, у которой большинство соединений выполнено по оси X.

Для выполнения второй стороны следует:

— с помощью программы 3 получить новую таблицу (невыполненных соединений);

— повторить программу 3;

— отсортировать по длине (проц. 1).

Вторая сторона платы при этом содержит соединения преимущественно по оси Y.

## 2.7. Трассировка.

Режим работы — автоматический. В конечном итоге мы получим чертеж рисунка соединительных проводников, который можно увидеть, вызвав подпрограмму 8. При ее работе на экране монитора высвечиваются те позиции из таблицы соединений, выполнить которые программа не в состоянии. Эта информация автоматически вводится в таблицу соединений, где невыполненные соединения отмечаются.

После вызова программы нужно выбрать один из режимов ее работы:

N) — это режим, который трассирует соединения заново и требует дополнительной информации: о невыполненных соединениях, на экран монитора или на печать, следует также указать, какую сторону нужно трассировать.

R) — этот режим позволяет выполнить ранее не выполненные соединения. Эту программу следует вызывать после ручной корректировки рисунка в подпрограмме 8.

A) — эта программа производит повторный анализ таблицы соединений и поиск другой точки, связанной с данной цепью, к которой следует произвести соединения.

Во время работы всех режимов таблица соединений автоматически обновляется по мере поступления информации о выполненных соединениях, что инициируется удалением знака "\*".

Эту программу следует повторить многократно попеременно с программой 8.

## Программная стыковка SINCLAIR ZX SPECTRUM с принтером

### ОБЩЕЕ УКАЗАНИЕ.

Для написания драйвера, соответствующего вашему принтеру, необходимо знать следующее:

1. Адреса портов принтера.
2. Коды настройки принтера и интерфейса.
3. Необходимые управляющие коды принтера (перевод строки, перевод каретки и т. п.).
4. Назначение разрядов слова состояния.
5. Максимальное количество символов в строке принтера.
6. Другую специальную информацию.



Драйвер принтера обычно размещается в области, называемой буфером принтера (PRINTER BUFFER). Это 256 байт между видео памятью и системными переменными, расположенных начиная с адреса 23296.

Первое, что необходимо сделать для написания драйвера, — это предусмотреть настройку каналов БЕЙСИК-интерпретатора, принтера и интерфейса.

Настройка канала необходима для того, чтобы с принтером можно было работать из БЕЙСИКа без дополнительных ухищрений (т. е. с использованием операторов LPRINT, LLIST и других). Как настроить канал, будет сказано ниже.

Настройка интерфейса и принтера довольно специфична, поэтому конкретных указаний дать просто невозможно.

Сама программа, обслуживающая принтер, как правило, проста. В минимальном варианте необходимо дожидаться готовности принтера к приему информации и послать байт. Когда БЕЙСИК-интерпретатор передает управление вашей программе, аккумулятор (регистр А) содержит код выводимого символа.

Нужно учесть, что при распечатке листинга БЕЙСИК-программы вашему драйверу будут передаваться не только коды символов, но и коды токенов (т. е. операторов БЕЙСИКа и т. п.). В этом случае принтер будет работать неправильно. Как этого избежать, будет показано ниже.

В разных моделях печатающих устройств для печати с новой строки необходимо передать 1 либо 2 управляющих кода. Эту особенность надо учитывать. Не лишним будет организовать подсчет количества напечатанных в строке символов, т. к. не все принтеры продолжают печатать со следующей строки после заполнения текущей, а зная количество напечатанных в строке символов, можно организовать перевод строки в нужный момент. Кроме того, по ходу дела можно изменять количество символов в строке, что поможет красиво оформить печатаемый текст.

Также необходимо учитывать специфику принтера и интерфейса. Иногда, до или после печати символа, приходится засылать в порт контрольные коды или другую информацию.

### ПРИМЕР ДРАЙВЕРА.

Далее на примере принтера ROBOTRON 1152 с параллельным интерфейсом рассмотрим возможный вариант драйвера. Полная его распечатка с подобными комментариями поможет вам быстрее разобраться во многих тонкостях программной стыковки печатающего устройства и SINCLAIR ZX SPECTRUM.

Исходные условия следующие:

1. Адрес настройки параллельного порта — 243.
2. Адрес слова состояния — 242.
3. Код символа передается по адресу — 241.
4. Максимальное число символов в строке — 132.
5. Для печати с новой строки необходимо передать перевод каретки (CARRIAGE RETURN) и перевод строки (LINE FEED).

Как уже говорилось, первое, что необходимо сделать, — это настроить канал БЕЙСИК-интерпретатора, принтер и интерфейс. Для ROBOTRON 1152 это можно сделать следующим образом:



LD	A, 156	;настройка порта.
OUT	(243),A	
LD	A, 5	;разрешение прерываний от принтера
OUT	(243),A	
LD	A, (LENGTH)	;установка длины строки
LD	(COUNT), A	;в счетчик.
LD	DE, (MAIN)	;настройка канала.
LD	HL, (23631)	
LD	BC, 15	
ADD	HL, BC	
LD	(HL), E	
INC	HL	
LD	(HL), D	
RET		;возврат в БЕЙСИК-интерпретатор

Теперь интерфейс готов к приему информации и передаче ее принтеру, принтер настроен, а БЕЙСИК-интерпретатору сообщено, что вывод на печать организован по адресу "MAIN".

Далее идет текст самого драйвера. Сначала необходимо определить, является ли передаваемый байт симвом, токеном или управляющим кодом.

MAIN:	CP	13	;передаваемый байт — перевод строки.
	JP	NZ, FORW2	;если нет, то переход на метку FORW2.
ENTER:	CALL	PRINT	;передача байта возврата каретки.
	LD	A, 10	;передача байта перевода строки.
	CALL	PRINT	
	LD	A, (LENGTH)	;установка длины строки.
	LD	(COUNT), A	
	RET		;возврат в БЕЙСИК-интерпретатор.
FORW2:	CP	165	;передаваемый байт — токен.
	JP	C, FORW4	;если нет, то переход на метку FORW4.
	SUB	165	;передача байта БЕЙСИК-интерпретатору
	JP	3088	;для дальнейшего разбора.
FORW4:	CALL	PRINT	;печать символа.
	LD	HL, COUNT	;уменьшение счетчика символов
	DEC	(HL)	;в строке.
	LD	A, 13	;код перевода строки — в регистр A.
	JP	Z, ENTER	;в случае, если строка заполнена, то
			;переход на ENTER.
	RET		;возврат в БЕЙСИК-интерпретатор.

Далее следует подпрограмма передачи байта принтеру. Она включает в себя опрос слова состояния принтера и опрос клавиши "BREAK".

PRINT:	DI	;запрет прерываний.
WAIT:	CALL	8020
	JP	NC, BREAK
	IN	A, (242)
	BIT	7, A

;опрос клавиши "BREAK".  
;если нажата — переход на метку BREAK.  
;опрос слова состояния.  
;старший бит, установленный в "1",  
;сигнализирует о неисправности  
;(конец бумаги и т. п.).



	JP	NZ, WAIT	;в этом случае — ждать готовности или ;нажатия клавиши "BREAK".
	BIT	0,A	;младший бит, установленный в "0", ;сигнализирует о неготовности принтера к ;приему информации.
	JP	Z, WAIT	;в этом случае — ждать готовности.
	OUT	(241),A	;передача байта принтеру.
	EI		;разрешение прерываний.
	RET		;возврат из подпрограммы.
BREAK:	EI		;разрешение прерываний.
	RST	8	;вывод сообщения.
	DEFB	20	"L BREAK INTO PROGRAMM".
LENGTH:	DEFB	132	;ячейка памяти, содержащая длину строки.
COUNT:	DEFB	0	;эта ячейка используется для подсчета количества напечатанных символов.

Запуск драйвера осуществляется оператором БЕЙСИКа RANDOMIZE USR 23296, после выполнения которого БЕЙСИК-интерпретатор и другие системные программы (GENS, MONS, PASCAL) могут работать с принтером.

Драйвер может включать в себя различные сервисные дополнения. Можно, например, организовать подсчет строк, напечатанных на листе. Это позволило бы без дополнительных ухищрений нумеровать страницы, ожидать замены листа и т. п.

Нельзя, однако, забывать, что буфер принтера — не бездонная бочка, а всего лишь 256 байт, а уложиться в них часто бывает затруднительно. В крайнем случае, можно содержать драйвер в каком-либо другом месте памяти, но тогда он может помешать другим программам, так что для разных программ для содержания драйвера будет необходимо выбирать разные области памяти.



## СОДЕРЖАНИЕ

Введение .....	1
1. Краткое описание языка BASIC .....	3
2 Краткое описание языка BETA BASIC .....	15
3. RAM DOS .....	20
4. SINCLAIR RESEARCH LTD .....	24
5. Краткое описание языка PASCAL .....	28
6. Графический редактор ART STUDIO .....	33
7. PLATA V.3 .....	41
8. Программная стыковка SINCLAIR ZX SPECTRUM с принтером .....	45



