

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРО-ЭВМ**

**«ЭЛЕКТРОНИКА БК 0010»**

**«ЭЛЕКТРОНИКА БК 0010Ш»**

**ЯЗЫК «БЕЙСИК»**

**РУКОВОДСТВО ОПЕРАТОРА**

**00001 — 01.34.09**

**1991**

## АННОТАЦИЯ

Настоящий документ представляет собой руководство оператора МИКРО-ЭВМ «ЭЛЕКТРОНИКА БК 0010» для работы с БЕЙСИК—системой. В данной системе пользовательские программы перед выполнением компилируются. Это позволяет лучше использовать быстродействие МИКРО-ЭВМ, чем в БЕЙСИК—системах, основанных на интерпретаторах.

Система обладает большим количеством удобных средств по вводу и редактированию текстов программ, широким набором команд для загрузки и запуска программ, хранящихся на кассетной магнитной ленте. В документе описаны все эти средства, даны указания по пользованию клавиатурой МИКРО-ЭВМ, а также полный список сообщений об ошибках.

## 1. НАЗНАЧЕНИЕ ПРОГРАММЫ И УСЛОВИЯ ЕЁ ВЫПОЛНЕНИЯ

БЕЙСИК—система для бытовой МИКРО-ЭВМ «ЭЛЕКТРОНИКА БК 0010» представляет собой единый комплекс программ, позволяющий пользователю создавать и выполнить программы на языке программирования БЕЙСИК в режиме диалога человека с ЭВМ. Она предназначена как для непосредственного выполнения команд и операторов языка, так и для выполнения БЕЙСИК—программ с предыдущей их компиляцией. Система обладает широкими возможностями по редактированию и отладке программ.

Для работы с БЕЙСИК—системой необходима МИКРО-ЭВМ «ЭЛЕКТРОНИКА БК 0010», подключённый к ней бытовой телевизионный приёмник и бытовой кассетный магнитофон, предназначенный для хранения текстов пользовательских программ и данных. Предусмотрено также подключение печатающего устройства для распечатки текстов и/или результатов работы программ.

Настоящая версия системы хранится в постоянном запоминающем устройстве (ПЗУ) «ЭЛЕКТРОНИКИ БК 0010».

## 2. ПУСК БЕЙСИК-СИСТЕМЫ

### 2.1. Начальный пуск БЕЙСИК—системы

Так как БЕЙСИК—система находится в ПЗУ машины, то пуск системы осуществляется автоматически при включении питания МИКРО-ЭВМ. Признаком нормальной работы системы служит выполнение следующих действий: очистка экрана телевизора, формирование на телевизионном экране служебной строки и вывод начального сообщения:

CAGLRCCC . R

ЛАТ

БЕЙСИК


(ВИЛЬНЮС, 1986.07.24)

ОК

Дата указывает время изготовления системы, вывод «ОК» означает готовность системы к работе и ожидание ввода команд оператора. Служебная строка является подсказкой пользователю о действующем режиме работы клавиатуры и экрана.

## 2.2. Повторный пуск

В БЕЙСИК—системе предусмотрена команда передачи управления монитору драйверной системы МИКРО-ЭВМ (команда MONIT). Для повторного входа в БЕЙСИК—систему достаточно ввести команду:

? BASIC <  >

Следует иметь в виду, что при выходе в монитор созданная БЕЙСИК—программа теряется.

## 3. НАЗНАЧЕНИЕ КЛАВИШ МИКРО-ЭВМ

Клавиатура МИКРО-ЭВМ используется для ввода строк программы, команд и другой управляющей информации, в соответствии с которой БЕЙСИК—система выполняет определённые действия. Все клавиши по функциональному назначению подразделяются на четыре группы

- алфавитно-цифровые.
- регистровые;
- редактирующие;
- управляющие.

### 3.1. Алфавитно-цифровые и регистровые клавиши

Алфавитно-цифровые клавиши предназначены для набора текстов команд и операторов языка, а также данных для БЕЙСИК—программ. В эту группу также входят некоторые редактирующие и управляющие экраном клавиши.


Каждая клавиша в зависимости от нажатых регистровых клавиш обозначает один из трёх символов. Ввод заглавных и строчных букв русского и латинского алфавитов осуществляется при включении соответствующей комбинации регистров «ЗАГЛ», «СТР», «РУС», и «ЛАТ», которые действуют до нажатия альтернативной клавиши. О включении режимов «РУС» и «ЛАТ» свидетельствует соответствующая надпись в служебной строке.


Для получения символов «[», «]», «\», «@» и знаков логического отрицания и подчёркивания необходим режим

ЛАТ—ЗАГЛ. При режиме ЛАТ—СТР эти клавиши выдают соответственно правые и левые фигурные скобки, вертикальную черту, знаки слабого ударения, надчеркивания и забоя.

Клавиша «Ъ» при русском регистре выдаёт заглавный или строчный твёрдый знак русского алфавита, в режиме ЛАТ—ЗАГЛ — символ подчёркивания, в режиме ЛАТ—СТР — редактирующий код «ЗАБОЙ» (см. ниже).

Пустая клавиша означает пробел.

Для цифр и остальных клавиш со специальными символами клавиши «ЛАТ», «РУС», «ЗАГЛ», «СТР» не действуют. При нажатии на эти клавиши выдаётся верхний символ. Для получения нижнего символа необходимо нажать на регистровую клавишу «» шифт и, не отпуская её, нажать на желаемую клавишу со спецсимволом.

Для русского алфавита: нажатие шифта, «» всегда обеспечивает ввод строчных букв, а для латинского — заглавных, независимо от режима «СТР» или «ЗАГЛ».

Для получения любого дополнительного символа необходимо аналогичным образом нажать на клавишу «АР2» и любую алфавитно-цифровую. Получаемые при этом символы полуграфики используются в основном для построения различных таблиц.

Регистры «» и «АР2» не фиксируются.

### 3.2. Управляющие клавиши

**ПРЕДНАЗНАЧЕНИЕ ОСНОВНЫХ УПРАВЛЯЮЩИХ КЛАВИШ СЛЕДУЮЩЕЕ:**


- «СУ» — при одновременном нажатии на эту клавишу и на любую буквенную получают управляющие или редактирующие коды (см. ниже);
- «СТОП» — прерывание выполнения программы с запоминанием номера строки останова, прекращение работы с внешними устройствами, вывод сообщения и перевод системы в непосредственный режим выполнения;
- «ШАГ» — при наличии скомпилированной

программы в памяти нажатие на эту клавишу в начале пустой строки приводит к выполнению очередного оператора программы;

- «ИНДСУ» — включает и сбрасывает режим индикации управляющих символов на экране (поддерживается при выводе кодов);
- «БЛОКРЕД» включает и сбрасывает режим индикации символов редактирующих клавиш на экране (поддерживается при выводе кодов);
- «AP2» / «СБР» — включение режима расширенной области памяти, отведённой для пользователя, с уменьшением экрана до четырёх информационных строк и обратно;
- «AP2» / «;» — переключение режима 32 символов в строке в 64 и обратно (одновременно 256 точек в графической строке в 512 и обратно);
- «AP2» / «-» — обмен текущего цвета символов с цветом фона экрана;
- «AP2» / «:» — гашение и восстановление курсора;
- «AP2» / «.» — формирование служебной строки в режиме формирования информации на основной части экрана (фон, цвет и т. п.);
- «AP2» / «,» — установка и сброс режима вывода на экран инверсных символов;
- «AP2» / «/» — установка и сброс режима вывода на экран подчёркнутых символов;
- «ПОВТ» — повтор последнего введённого символа неограниченное число раз до тех пор, пока эта клавиша нажата.

При нажатии на клавиши «ИНДСУ», «БЛОКРЕД» и некоторых других в служебной строке появляется или стирается соответствующая надпись.

Режимы блокировки редактирования, 64-х символов в строке и расширения памяти БЕЙСИК—системой не поддерживаются в том смысле, что информация на экране в этих режимах не всегда соответствует содержимому буфера ввода, что может привести к ошибкам.

При одновременном нажатии на клавиши «», «AP2» и на

одну из цифр «1», «2», «3», «4», можно переключить текущий цвет символов соответственно на красный, зелёный, синий, чёрный.


Редактирующие клавиши и программируемые ключи описываются в следующих разделах.

## **4. ВВОД ПРОГРАММ**

### **4.1. Основные инструкции**

Текст программы можно начать вводить сразу же после включения питания ЭВМ и появления на экране приглашения «ОК», которое означает, что ЭВМ ждёт ввода инструкций оператора. В таком режиме система находится с появления на экране букв «ОК» до ввода команды RU.

Каждая программная строка начинается с номера строки. Минимальный номер строки — 0, максимальный — 65535. За номером строки следуют ключевое слово оператора и его параметры.

Ввод строки осуществляется нажатием на клавишу «». Вводится вся логическая строка, независимо от того, что она может занимать несколько физических строк. Максимальная длина вводимой логической строки — 255 символов, включая номер строки.

Номер строки является единственным признаком отличия программных строк от строк, предназначенных для непосредственного выполнения. Это означает, что если номер строки во вводимой оператором инструкции не указан, то она выполняется сразу же после её ввода. Если номер строки указан, строка запоминается как одна из программных строк.

При вводе строк программы возможны следующие ситуации:

- к программе добавляется новая строка каждый раз, когда вводится непустая строка с допустимым номером и в программе пока нет строки с номером, идентичным вводимому;
- уже существующая программная строка заменяется новой, если номер непустой вводимой строки

совпадает с номером существующей в программе строки;

- существующая строка удаляется из программы, если введён лишь номер этой строки без единого отличного от пробела символа за ним;
- никакие действия системой не предпринимаются, когда вводится пустая строка без номера или с номером, не совпадающим ни с одним номером раньше введённых программных строк;
- выдаётся сообщение об ошибке 2, если указан номер строки, превышающий допустимый предел, сообщение об ошибке 25, если введено больше 255 символов или сообщение об ошибке 7, когда не достаёт места в памяти ЭВМ для запоминания вводимой строки.

В последнем случае введённая строка к программе не добавляется.

Программные строки можно вводить в любой последовательности, но выполняются они всегда в порядке возрастания номеров строк.

Рекомендуется номеровать строки с некоторым шагом, например, 10, 20, 30 и т.д. Это позволит в случае необходимости добавить во внутрь программы новые строки.

Если без выключения питания ЭВМ вводятся тексты нескольких программ, перед вводом каждой новой программы необходимо ввести команду NEW для стирания старой. Сразу после включения питания команду давать не обязательно.

## **4.2. Программируемые ключи**






Ввод как программных строк, так и команд, могут существенно облегчить и ускорить так называемые программируемые ключи или функциональные клавиши. Это такие клавиши, нажатие на которые может вызывать ввод сразу нескольких символов, например, ключевых слов операторов БЕЙСИКА. При этом, оператор ЭВМ может сам задавать произвольное значение этих ключей.


В МИКРО-ЭВМ «ЭЛЕКТРОНИКА БК 0010» существуют 10 программируемых ключей. Это клавиши «1», «2» и т.д. до «0».



Нажимать на них необходимо при нажатой клавише «AP2». Значение этих ключей, т.е. последовательности символов, вводимых при нажатии на эти клавиши, можно задавать специальным оператором KEY (см. «ОПИСАНИЕ ЯЗЫКА»).

Ключи имеют свои начальные значения, которые они получают при запуске БЕЙСИК—системы. Эти значения следующие:

- 1 COLOR <ПРОБЕЛ>
- 2 AUTO <ПРОБЕЛ>
- 3 GOTO <ПРОБЕЛ>
- 4 LIST <ПРОБЕЛ>
- 5 RUN <  >
- 6 COLOR 1.4 <  >
- 7 CLOAD"
- 8 CONT<  >
- 9 .<  >
- 0 <СБР> RUN <  >

<СБР> означает код очистки экрана. Как видно, нажатие на некоторые из клавиш вызывает даже выполнение определённых команд, поскольку они обеспечивают передачу в ЭВМ и кода клавиши «  ».

Из начальных печатных символов каждого ключа в левой стороне служебной строки формируется строка—подсказка. Её содержание соответственно меняется при каждом определении нового содержимого любого ключа. Если этой подсказки недостаточно, значения ключей можно просмотреть при помощи оператора KEYLIST (см. «ОПИСАНИЕ ЯЗЫКА»).

Рекомендуется при вводе программ присвоить ключам наиболее часто встречаемые в программе комбинации символов и затем при вводе текстов использовать эти ключи. Максимальная длина значения ключа — 15 символов.

## 5. РЕДАКТИРОВАНИЕ

Работа с ЭВМ в диалоговом режиме часто сопровождается многочисленными ошибками при вводе с клавиатуры. Каждый программист также редко избегает синтаксических ошибок при записи операторов и, тем более, логических ошибок в алгоритмах, реализованных в программах. Поэтому во всех системах программирования предусмотрены средства редактирования программ.

В БЕЙСИК—системе существует встроенный редактор текстов, который по сути дела организует весь диалог человека с ЭВМ: ввод и редактирование программ, организацию выполнения команд и операторов в непосредственном режиме, загрузку и запуск программ.

Условно можно выделить следующие случаи редактирования:


- исправление ошибок во вводимой строке;
- редактирование содержимого буфера ввода;
- редактирование строк программы;
- редактирование при помощи команд.

### 5.1. Исправление ошибок во вводимой строке

Редактор БЕЙСИК—системы обеспечивает полное экранное редактирование любой логической строки, но не выходя за её пределы. При обнаружении опечатки во вводимой строке можно клавишами управления курсором навести его на нужный символ строки, а потом применять любую из следующих мер:

- напечатать новый символ на месте введённого раньше;
- удалить символ, находящийся слева от курсора;
- удалить символ, находящийся на позиции курсора;
- удалить все символы, начиная от позиции курсора и до конца логической строки;
- раздвинуть строку для вставки новых символов.

Кроме того, можно вообще отказаться от вводимой строки, нажав на клавишу «СБР» или «КТ» (см. следующий пункт).

Все указанные действия можно выполнять клавишами редактирования, но лишь до ввода строки (нажатия на клавишу «»). Положение курсора во вводимой строке при вводе не имеет значения.

Переход символов из одной физической строки в другую при вставке или удалении символов производится автоматически.



## 5.2. Клавиши редактирования

Ниже представлена таблица всех редактирующих клавиш с указанием восьмеричных кодов, вырабатываемых при нажатии на эти клавиши, и с краткой характеристикой предпринимаемых по ним действий редактора. В таблице указаны и некоторые другие клавиши, применяемые как при вводе, так и при редактировании программ, а также некоторые управляющие клавиши.








Коды, соответствующие большинству этих клавиш, можно получить, нажав на клавишу «СУ» и на некоторую алфавитно-цифровую. В таблице указаны и эти клавиши.

За таблицей поясняются действия некоторых клавиш.

Таблица 1.

Восьм. код	Клавиша СУ	Редакт. клавиша	Назначение
000	@		Задержка вывода на экран
001	A		Игнорируется
002	B		Игнорируется
003	C	КТ	Отказ от вводимой строки
004	D		Игнорируется
005	E		Игнорируется
006	F		Игнорируется
007	G		Звонок
010	H		Курсор влево
011	I		Игнорируется
012	J		Ввод строк

## Продолжение таблицы 1

Восьм. код	Клавиша СУ	Редакт. клавиша	Назначение
013	К		Игнорируется
014	L	СБР	Очистка экрана
015	М		Установка позиции табуляции
016	N	РУС	Регистр русского алфавита
017	O	ЛАТ	Регистр латинского алфавита
020	P		Сброс позиции табуляции
021	Q		Игнорируется
022	R		Игнорируется
023	S	ВС	Курсор в начало логической строки
024	T		Горизонтальная табуляция
025	U		Игнорируется
026	V		Сдвигка строки
027	W		Раздвигка строки
030	X		Аннулирование символа слева
031	Y		Курсор вправо
032	Z		Курсор вверх
033	[		Курсор вниз
034	\		Курсор влево вверх
035	]		Курсор вправо вверх
036	^		Курсор вправо вниз
037	Ъ		Курсор влево вниз
177			Забой текущего символа
211		ТАБ	Ввод пробелов
231			Сброс конца строки

ПРИМЕЧАНИЕ. В таблице указаны основные действия редактора, вызванные нажатием на эти клавиши. При выводе кодов этих клавиш действия могут быть иными.

*Задержка вывода на экран*

- При нажатии на клавиши СУ/@ организуется цикл ожидания ввода следующего символа. Это приводит к приостановлению любых других действий ЭВМ (вывода на экран, вычислений и т.д.). Продолжение работы — нажатие любой клавиши. Есть особые случаи, когда клавиатура блокируется (прерывания от неё не обрабатывается) и ввести СУ/@ невозможно. Так случается, например, при работе с магнитофоном. При вводе код игнорируется.

*Отказ от вводимой строки*

- Нажатие на клавишу «КТ» приводит к прерыванию ввода текущей строки, выводу приглашения «ОК» и переводу ЭВМ к ожиданию ввода новой строки. Вводимая строка не запоминается и не выполняется (если это была команда). При выводе игнорируется.

*Звонок*

- Звуковой сигнал, аналогичный получаемому при нажатии на любую клавишу.

*Ввод строк*

При выводе кода этой клавиши осуществляется перевод курсора в начало следующей строки.

*Установка позиции табуляции*

- Ниже линии, отделяющей служебную строку от основной части экрана, на позиции, соответствующей позиции курсора, появляется чёрточка. После этого, по нажатию на клавишу «ТАБ», вводятся пробелы до тех пор, пока курсор не окажется на следующей затабулированной позиции.

*РУС и ЛАТ*

- При выводе игнорируются.

*Сброс позиции табуляции*

- Отказ от затабулированной позиции. Чёрточка под линией служебной строки исчезает. При выводе игнорируется.

*Курсор в начало строки*

- Курсор переводится в начало логической строки, если

только строка не пустая. При выводе нижние от курсора физические строки экрана сдвигаются на одну строку вверх. Строка, на которой находился курсор, удаляется.

#### *Горизонтальная табуляция*

- Перевод курсора на следующую позицию, кратную восьми (0, 8, 16, 24). При передвижении курсора по уже введённой части логической строки, символы не стираются. В конце строки добавляются пробелы. При выводе производится сдвигение физических строк вниз, начиная от строки с курсором. Данная физическая строка становится пустой.

#### *Сдвигка строк*

- Осуществляется удаление символа на позиции курсора со сдвижкой всей оставшейся части логической строки на один символ влево. Символы с крайней левой позиции переходят в крайнюю правую позицию предыдущей строки. При выводе производится аналогичное смыкание в пределах физической строки. В конце строки появляется пробел.

#### *Раздвижка строки*

- Сдвиг всех символов логической строки на одну позицию вправо, начиная от позиции курсора. Позиция курсора освобождается. Крайние правые символы переходят в крайнюю левую позицию следующей физической строки. При выводе сдвиг производится в пределах физической строки. Крайний правый символ теряется.

#### *Аннулирование символа слева*

- Курсор переходит на одну позицию влево. Одновременно передвигаются и все символы оставшейся части строки. Символ, раньше находившийся слева от курсора, удаляется. Если курсор находится в первой позиции логической строки, действия аналогичны сдвигке строки. Используется для аннулирования последнего введённого символа. При выводе удаляется левый символ без сдвижки остальных символов.

#### *Забой символов*

- Аналогичен сдвигке строки. При выводе на экран

рисуется закрашенный прямоугольник величиной с заглавную букву.

#### *Ввод пробелов*

- Вводятся пробелы, пока курсор не достигает затабулированной позиции. При отсутствии затабулированных позиций пробелы вводятся до конца текущей физической строки. При выводе игнорируется.

#### *Сброс конца строки*

- Удаление всех символов, начиная от позиции курсора и до конца логической строки. При выводе действует в пределах физической строки.


Все клавиши со стрелками продвигают курсор на одну позицию в соответствующем направлении, но при вводе действуют лишь в пределах логической строки. При выводе кода 22 курсор переводится в начало экрана. Код 25 при выводе аналогичен коду 12.

### **5.3. Вызов последней введённой строки**

Все редактирующие клавиши действуют в пределах вводимой строки и лишь до её ввода. Для случаев, когда ошибки обнаруживаются уже после ввода, в БЕЙСИК—системе существует возможность вызвать последнюю введённую строку на редактирование и повторный ввод. Это достигается при помощи клавиши «ВС», на которую в системе наложены всего три функции (об одной из них см. «Описание языка», команда LIST)

Если строка уже начата, т.е. набрано несколько символов (даже если это пробелы), то «ВС» переводит курсор в начало логической строки (см. п. 5.2.). Но если ни один символ ещё не набран, то при нажатии на «ВС» на экран выводится содержимое буфера ввода, т.е. строка, которая была введена только что. Теперь её можно отредактировать обычным способом и ввести заново.

Клавишу «ВС» удобно также использовать для выполнения одного оператора или команды несколько раз подряд. Но следует запомнить, что на экране всегда показывается содержимое буфера ввода, поэтому есть случаи, которые при

незнании сбивают с толку. При нажатии на клавишу «» без единого символа в логической строке вводится пустая строка. Клавиша «ВС» после этого не позволит возвратиться раньше введённые символы, поскольку последней введённой оказывается пустая строка. Другой случай, когда начат набор строки, потом все символы аннулированы при помощи редактирующих клавиш и нажата «ВС». На экране появится не последняя введённая строка, а содержимое буфера ввода, т.е. набранные и аннулированные символы, возможно, с конечными символами раньше введённых строк.

Кроме этих случаев, существуют команды, которые используют буфер ввода для своих целей (LIST, SAVE, LOAD). После корректного выполнения этих команд «ВС» даёт пустую строку.

#### 5.4. Редактирование строк программы


Особые случаи замены и удаления строк программы указаны в п. 4.1. Для редактирования текста какой-либо программной строки необходимо ввести следующую строку:

.<НОМЕР СТРОКИ> < >

После точки необходимо указать номер строки, подлежащей редактированию. После ввода указанной команды на экране появляется текст строки, и можно приступить к его редактированию, (включая номер строки). При указании номера несуществующей строки выдаётся сообщение об ошибке 8.


В БЕЙСИК—системе символ «.» во многих командах применяется для обозначения текущей строки. В описываемом случае точка означает, что строку с указанным номером необходимо сделать текущей. Текущей строкой в системе считается:

- последняя введённая;
- последняя редактированная;
- последняя выведенная командой LIST;
- последняя ошибочная строка.

Удаление строки путём набора её номера и нажатия на «» (см. п. 4.1.) также делает удалённую строку текущей.



Но после такого удаления всякие обращения к текущей строке посредством точки приведут к ошибке 8.

Для редактирования текущей строки достаточно набрать точку и «» (см. п. 4.2., ключ 9).

Термин «ТЕКУЩАЯ СТРОКА» относится исключительно к программным строкам. Со строками непосредственного выполнения он не имеет ничего общего.

### 5.5. Команды редактирования

В БЕЙСИК—системе существует ряд специальных команд, облегчающих ввод и редактирование программ. Их список приведён в таблице 2.

Таблица 2

Команды редактирования	
Команда	Действие
LIST [N][—[M]]	Вывод строк программы на экран, начиная с номера N, до номера M
DELETE [N][—[M]]	Удаление строки, строк или всей программы
NEW	Удаление всей программы
AUTO [N][,M]	Установка режима автонумерации, начиная с N, с шагом M
RENUM [N[,M][,L]]	Перенумерация строк: номер N заменяется на M и далее с шагом L

В командах LIST, DELETE, AUTO вместо номеров строк можно указывать текущую строку. Команда LIST выводит строки страницами в прямом и обратном направлении.

Команду RENUM удобно использовать в случаях, когда строки плотно пронумерованы и некуда вставить новые строки.

Команда AUTO при вычислении очередного номера, совпадающего с номером существующей строки, показывает её на экране и позволяет редактировать (включая номер строки). Поэтому команду удобно применять как для ввода, так и для

редактирования программ.

Полное описание команд редактирования можно найти в документе «ОПИСАНИЕ ЯЗЫКА».

## 5.6. Пример редактирования программы

Для замены строки

```
40 PRINT A
```

Можно ввести строку заново, например:

```
40 PRINT B
```

Но удобнее воспользоваться командой редактирования:

```
. 40 < ←↵ >
```

При выполнении этой команды на экран выводится строка с указанным номером, и теперь её можно исправить при помощи клавиш редактирования (напр., нажать клавиши « ←↵ », «B» и « ←↵ »).

Для удаления строки можно ввести директиву:

```
20 < ←↵ >
```

Для этой же цели можно воспользоваться командой:

```
DELETE 20 < ←↵ >
```

Для удаления фрагмента программы вводится команда:

```
DELETE 50-100 < ←↵ >
```

Вся программа удаляется при вводе следующих команд:

```
DELETE или NEW
```

**ПРИМЕЧАНИЕ.** Перед тем, как приступить к созданию программ с БЕЙСИК—система для МИКРО-ЭВМ «ЭЛЕКТРОНИКА БК-0010», необходимо изучить практически средства редактирования текста программы. Это существенно ускорит работу с текстом программы.

## 6. ХРАНЕНИЕ ПРОГРАММ НА МАГНИТНОЙ ЛЕНТЕ

### 6.1. Форматы файлов

БЕЙСИК—система допускает работу с тремя типами файлов:

- информация в текстовом виде;

- БЕЙСИК—программы во внутреннем коде системы;
- файлы с двоичной информацией.

Файлами первого типа являются БЕЙСИК-программы, записанные на ленту при помощи команды SAVE (см. п. 6.2.) и файлы данных, созданных пользователем средствами БЕЙСИК-системы. Программы можно загружать и командой LOAD и операторами работы с файлами данных. Файлы данных можно загружать только операторами работы с файлами данных, т.к. команда LOAD требует, чтобы в файле были программные строки (в общем случае строки с допустимыми номерами). Иначе загрузка такого файла приведёт к ошибке. По умолчанию команды SAVE и LOAD работают с файлами с присвоенным им типом ASC. Файлы данных по умолчанию имеют тип DAT. Эти три буквы показываются рядом с именами файлов при чтении информации с ленты (см. команду FIND).

Для работы с файлами других типов также имеются специальные команды (см. п. 6.2.). Их типы по умолчанию соответственно COD и BIN. Программы типа COD загружаются быстрее, но внутренний формат программ может быть различным в различных версиях системы. Файлами типа BIN обычно являются программы в машинных кодах или копии экрана.

## 6.2. Основные команды системы

Список команд, осуществляющих загрузку программ с магнитофона, их запуск или запись приведён в таблице 3. В ней указаны и специальные команды определения конфигурации системы, передачи управления системным или пользовательским программам в машинных кодах.

Таблица 3

Команда	Действия
LOAD" <ИМЯ> [, R]	Загружает программу с указанным именем с магнитофона. При наличии буквы R производится пуск загруженной программы

Команды	Действия
C LOAD" <ИМЯ> [, R]	Загружает программу с магнитофона во внутреннем формате БЕЙСИК-системы
B LOAD" <ИМЯ> [, R]	Загружает в память машины двоичные коды
RUN [N]	Запускает БЕЙСИК-программу, находящуюся в текущий момент в памяти, со строки с указанным номером
FIND [" <ИМЯ> ]	Поиск программы с указанным именем на магнитной ленте
SAVE " <ИМЯ>	Записывает программу, находящуюся в памяти, на магнитную ленту с указанным именем
CSAVE" <ИМЯ>	Записывает программу на кассетную магнитную ленту во внутреннем формате БЕЙСИК-системы
BSAVE" <ИМЯ>, A1, A2	Записывает копию памяти на магнитную ленту
MONIT	Выход в монитор драйверной системы
CLEAR [L, A]]	Определение конфигурации системы
CALL	Вызов внешних модулей ПЗУ

Командами работы с магнитофоном можно осуществлять обмен информацией по линии связи (напр. в локальной учебной сети).

Полное описание формата команд и вызываемых ими действий можно найти в документе «ОПИСАНИЕ ЯЗЫКА».

## 7. СООБЩЕНИЯ ОПЕРАТОРУ ОБ ОШИБКАХ

### 7.1. Формат сообщений

БЕЙСИК осуществляет проверку операторов программы и вводимых данных и о каждой обнаруженной ошибке выводит соответствующее сообщение. Сообщения об ошибках имеют следующий формат:

Ошибка ХХ в строке УУУУУ,

где ХХ — код ошибки,

УУУУУ — номер строки, в которой обнаружена ошибка.

В случае синтаксических ошибок в командной строке, а также в случае ошибок, во время её выполнения, номер строки не выводится.

### 7.2. Ошибки

Коды ошибок и пояснения к ним приводятся в приложении.

Ошибки можно разделить на синтаксические и возникающие во время выполнения программы или команды. При обнаружении хотя бы одной синтаксической ошибки программа на выполнение не запускается. Если возникает ошибочная ситуация во время выполнения программы, процесс выполнения прерывается и БЕЙСИК-система переводится в режим непосредственного выполнения. В последнем случае есть возможность продолжить выполнение программы при помощи команд CONT или GOTO. Ошибочная строка становится текущей.

При обнаружении, любой ошибки в строке непосредственного выполнения её обработка также немедленно прекращается, и система переводится в режим ожидания ввода следующей инструкции.

## СООБЩЕНИЯ ОБ ОШИБКАХ

В сообщениях об ошибках указывается код ошибки в виде: ошибка XX в строке YYYYYY, где XX — код ошибки, YYYYYY — номер строки, содержащей ошибку.

В приложении указаны коды ошибок, их сокращённые названия и описания.

- 1 NEXT без FOR  
NEXT не предшествовал FOR или переменная, использованная в FOR не соответствует переменной, использованной NEXT
- 2 Синтаксическая ошибка  
Неверное использование символов, например, число открывающих скобок не соответствует числу закрывающих, неправильная запись операторов или их составных частей, неправильно использована запятая и т. п.
- 3 RETURN без GOSUB  
При выполнении RETURN обнаружено, что не был выполнен GOSUB.
- 4\* Исчерпан список оператора DATA
- 5 Недопустимое использование функции  
При выполнении оператора READ обнаружено, что список оператора DATA исчерпан.
- 6 Переполнение  
Результат арифметической операции не может быть записан в формате, принятом для чисел в БЕЙСИК-системе. В случае потери порядка результат приравнивается нулю.
- 7 Переполнение памяти  
Программа не уместается в памяти, или использовано слишком много операторов FOR или переменных.

- 8 Не определён номер строки  
В GOTO, GOSUB, IF... THEN... ELSE, RETURN, READ, RENUM, AUTO или DELETE использован неопределённый номер строки.
- 9 Недопустимый индекс, (за пределами)  
Ссылка на элемент массива с индексом, который выходит за пределы размерности массива, либо указано неправильное число индексов.
- 10 Повторное определение массива  
Массив определён двумя операторами DIM; или массив определён оператором DIM после того, как по умолчанию для этого массива была установлена размерность 10.
- 11 Деление на нуль  
В выражении встретилось деление на нуль; или нуль был возведён в отрицательную степень.
- 12 Недопустимый оператор в режиме непосредственного выполнения.
- 13 Ошибка типов  
Попытка присвоить символьной переменной числовое значение или наоборот; функции, использующей числовой аргумент, передаётся символьный аргумент или наоборот.
- 14 Не хватает места символьным переменным  
Превышено количество оставшейся памяти, которая была отведена символьным переменным по умолчанию или оператором CLEAR.
- 15 Слишком длинная строка  
Была сделана попытка создать строку длиной более чем. 255 символов.
- 16 Не определена
- 17 Продолжение выполнения программы невозможно  
Была сделана попытка продолжить выполнение программы, которая:

1. Была прервана из-за возникновения ошибки.
2. Была изменена.
3. Не существует

- 18\* Не определена функция пользователя  
Попытка обратиться к функции `USR` до её определения оператором `DEF`.
- 19 Ошибка устройства ввода/вывода.  
Встречается при работе устройства В/В.
- 20 Не определена
- 21 Не определена
- 22 Не определена
- 23 Не определена
- 24 Отсутствует операнд  
Выражение содержит оператор без операнда, команда/оператор задана без обязательных параметров.
- 25 Переполнение буфера ввода  
Была сделана попытка ввести строку длиной более чем 255 символов.
- 26-51 Не определены
- 52\* Ошибочный номер файла  
Оператор или команда ссылается на файл, который не открыт, или номер файла выходит за первоначально определённый предел номеров файлов.
- 53\* Не определена
- 54\* Файл уже открыт  
Применён оператор `OPEN` для файла, который уже открыт, или оператор `KILL` применён для файла, который открыт.
- 55\* Попытка прочитать запись окончания файла.  
Оператор `INPUT` был выполнен после того, как все данные из файла были введены, или применён для



пустого файла. Чтобы избежать этой ошибки, для обнаружения окончания файла применяйте функцию EOF.

- 56      Ошибочное имя файла  
В операторах LOAD, SAVE или OPEN использовано неправильное имя файла (имя файла состоит из слишком большого количества символов).
- 57      Команда непосредственного выполнения в программе во время загрузки текстового файла. Загрузка заканчивается.
- 58      Не определена
- 59\*     Файл не открыт  
Команда/оператор ввода/вывода была применена к файлу, который не был открыт.
- 60      Не определена
- 61      Не определена
- 62\*     Ошибочное имя устройства  
Было использовано ошибочное имя устройства.
- 63-255   Не определены.

**ПРИМЕЧАНИЕ.** Звёздочкой (\*) помечены ошибки, которые в данной версии могут не встречаться, так как не реализованы соответствующие операторы.

