

Утверждён  
00008-01.34.04-ЛУ

**ПЕРСОНАЛЬНАЯ МИКРО-ЭВМ  
«ЭЛЕКТРОНИКА МС0513»**

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
БЕЙСИК  
РУКОВОДСТВО ОПЕРАТОРА**

**00008 — 01.34.04**

## АННОТАЦИЯ

Настоящий документ представляет собой руководство оператора микро-ЭВМ «Электроника МС0513» (Электроника БК0011) для работы с Бейсик-системой. В данной системе пользовательские программы перед выполнением компилируются.

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.

## СОДЕРЖАНИЕ

1.	НАЗНАЧЕНИЕ ПРОГРАММЫ И УСЛОВИЯ ЕЁ ВЫПОЛНЕНИЯ ...	4
2.	ПУСК БЕЙСИК-СИСТЕМЫ .....	4
2.1.	Начальный пуск Бейсик-системы .....	4
2.2.	Повторный пуск .....	5
3.	НАЗНАЧЕНИЕ КЛАВИШ МИКРО-ЭВМ.....	5
3.1.	Алфавитно-цифровые и регистровые клавиши .....	5
3.2.	Управляющие клавиши .....	6
4.	ВВОД ПРОГРАММ .....	6
4.1.	Основные инструкции .....	6
4.2.	Программируемые ключи.....	8
5.	РЕДАКТИРОВАНИЕ .....	8
5.1.	Исправление ошибок во вводимой строке.....	9
5.2.	Вызов последней введённой строки. ....	11
5.3.	Редактирование строк программы .....	11
5.4.	Команды редактирования .....	12
5.5.	Пример редактирования программы .....	13
6.	ХРАНЕНИЕ ПРОГРАММ НА МАГНИТНОЙ ЛЕНТЕ.....	13
6.1.	Форматы файлов.....	13
6.2.	Основные команды системы .....	14
7.	СООБЩЕНИЯ ОПЕРАТОРУ ОБ ОШИБКАХ.....	15
7.1.	Формат сообщений.....	15
7.2.	Ошибки .....	15
	СООБЩЕНИЯ ОБ ОШИБКАХ .....	16

# 1. НАЗНАЧЕНИЕ ПРОГРАММЫ И УСЛОВИЯ ЕЁ ВЫПОЛНЕНИЯ

Бейсик-система для бытовой микро-ЭВМ «Электроника БК0011» 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 БК0011», 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 БК0011».

## 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 (см. рис. 1.) и вывод начального сообщения (см. рис. 2.):

---

Lat

co au go li ru co cl co .L Lr

Рис. 1. Вид служебной строки при запуске Бейсик-системы.

**Примечание.** Если при включении питания не происходит автоматического запуска Бейсик-системы, то следует воспользоваться командами повторного пуска (см. п .2.2).

**БЕЙСИК БК0011**  
**ВИЛЬЮС, 1987.12.25**  
**ОК**

Рис. 2. Начальное сообщение Бейсик-системы.

Дата в сообщении указывает время изготовления системы (версию), 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 команд оператора драйверной системы микро-ЭВМ (команда SYSTEM). Для повторного входа в Бейсик-систему достаточно ввести последовательность команд:

```
@11;1C
```

```
@100000G
```

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. Это основная часть клавиатуры – все белые клавиши, в основном они все вызывают печать на экране какого-то символа. В особых случаях они могут вызвать некоторые управляющие или редактирующие действия (при одновременном нажатии на определённые регистровые или управляющие клавиши).

Каждая клавиша в зависимости от нажатых регистровых клавиш обозначает один из двух на ней нарисованных символов. Ввод заглавных и строчных букв русского и латинского алфавитов осуществляется при включении соответствующей комбинации специальных клавиш «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 «^» необходим режим ЛАТ—ЗАГЛ. При режиме ЛАТ—СТР эти клавиши выдают соответственно правые и левые фигурные скобки, вертикальную черту, знаки слабого ударения и надчеркивания.

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 нижнего регистра и, не отпуская её, нажать на желаемую клавишу со спецсимволом. Клавишей

нижнего регистра считается чёрная клавиша в левой стороне клавиатуры с нарисованной толстой стрелкой вниз.

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

Нажатием на клавишу «AP2» и клавиши букв можно получить символы полуграфики.

Нижний регистр и регистр «AP2» не фиксируются.

### **3.2. Управляющие клавиши**

Предназначение основных управляющих клавиш следующее:

— «СУ» — при одновременном нажатии на эту клавишу и на любую буквенную получаются управляющие или редактирующие коды (см. ниже);

— «ПОВТ» — повтор последнего введённого символа неограниченное число раз до тех пор, пока эта клавиша нажата;

— «ИНД СУ» — включает и сбрасывает индикацию служебной строки;

— «БЛОК РЕД» включает и сбрасывает режим индикации управляющих символов и символов редактирующих клавиш на экране в виде инверсных алфавитно-цифровых символов;

— «СТОП» — прерывание выполнения программы с запоминанием номера строки останова, прекращение работы с внешними устройствами, вывод сообщения и перевод системы в непосредственный режим выполнения, так же эта клавиша используется для отказа от ввода строки;

При нажатии на клавишу «БЛОК РЕД» в служебной строке появляется или стирается надпись "BLK", соответственно включается или выключается режим «БЛОК РЕД».

В этом режиме управляющие и редактирующие коды на экране отображаются в виде инверсных изображений соответствующих им букв или стрелок. Например, вместо кода очистки экрана печатается инверсная «L».

Управляющие клавиши «КТ» и «ШАГ» Бейсик-системой игнорируются.

Редактирующие клавиши и некоторые остальные возможности клавиатуры описываются в следующих разделах.

## **4. ВВОД ПРОГРАММ**

### **4.1. Основные инструкции**

Текст программы можно начать вводить после 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 RUN.

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 «ВВОД» («»)). Вводится вся логическая строка, независимо от того, что она может занимать несколько физических строк. Положение курсора в вводимой строке при вводе не имеет значения. 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 6 (см. приложение), если указан номер строки, превышающий допустимый предел, сообщение об ошибке 25, если введено больше 255 символов или сообщение об ошибке 7, когда в памяти ЭВМ нет достаточно места для запоминания вводимой строки. В этом случае введённая строка к программе не добавляется, а старая под таким же номером стирается.

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... Эта позволит в случае необходимости добавить во внутрь программы новые строки. Если всё-таки возникает ситуация, когда строки вставлять больше некуда, необходимо воспользоваться командой RENUM (см. 5.4).

Если не выключая питания ЭВМ вводятся тексты нескольких программ, перед вводом каждой новой программы необходимо ввести команду NEW для стирания старой. Сразу после включения питания эту 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 БК0011» существуют 10 программируемых ключей. Их можно задействовать нажимая клавиши цифр при одновременно нажатой клавише «AP2». Значение этих ключей 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
- 2 AUTO
- 3 GOTO
- 4 LIST
- 5 RUN <ПС>
- 6 COLOR 4,0 <ПС>
- 7 CLOAD
- 8 CONT<ПС>
- 9 .<ПС>
- 10 <СБР> RUN <ПС>

<СБР> означает код очистки экрана, код <ПС> получается при нажатии клавиши «ВВОД».

Как видно, нажатие на некоторые клавиши вызывает даже выполнение определённых команд, так как они обеспечивают передачу в ЭВМ и кода клавиши «ВВОД», обозначающей ввод строки.

Из начальных двух символов каждого ключа в служебной строке формируется строка-подсказка. Её содержание соответственно меняется при каждом определении нового содержимого любого ключа. Управляющие и редактирующие коды отображаются в режиме «БЛОК РЕД» (см. [3.2](#)).

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 — 63 символа.

## 5. РЕДАКТИРОВАНИЕ

Работе с ЭВМ в диалоговом режиме часто сопутствуют многочисленные ошибки при вводе знаков с клавиатуры. Каждый программист также не 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 микро-ЭВМ «Электроника БК0011» существует встроенный редактор текстов, который по сути дела организует весь диалог человека с ЭВМ в Бейсик-системе: 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 микро-ЭВМ обеспечивает редактирование любой логической строки, не выходя за её пределы. При обнаружении опечатки во вводимой строке можно клавишами управления курсором навести его на нужный символ строки, а потом применять любое из следующих действий:

- 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 начала логической строки;
- удалить все символы, начиная от позиции курсора и до конца логической строки;
- раздвинуть строку для вставки новых символов.

Все указанные действия можно выполнять клавишами редактирования, но лишь до ввода строки (нажатия на клавишу «ВВОД»).

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

Управляющие символы, не используемые для редактирования строки (см. ниже), вводятся в строку и отображаются в режиме «БЛОК РЕД» (см. [3.2](#)). Их можно набрать, нажимая на алфавитно-цифровые клавиши при одновременно нажатой клавише «СУ».

Управляющие символы, используемые при редактировании строки, обрабатываются в зависимости от установленного режима:

- в режиме «БЛОК РЕД» все символы вводятся в строку;
- при выключенном режиме «БЛОК РЕД» выполняется редактирование строки.

Редактирование строки заканчивается при нажатии клавиши «ВВОД».

Ниже перечислены функции управляющих символов, используемых для редактирования строки.

← — перемещает курсор на позицию влево; в первой позиции логической строки функция не выполняется;

«ВВОД» — завершает редактирование строки; в момент подачи команды курсор может находиться в любом месте строки;

 — стирание всех символов от курсора до конца логической строки;

СБР — очистка экрана и вывод редактируемой строки в текущем состоянии, начиная с первой позиции первой строки экрана; курсор после выполнения операции помещается в конце строки;

ВС — нажатие этой клавиши в случае, если курсор находится в первой позиции редактируемой строки, вызывает перемещение его в конец логической строки, в противном случае курсор переместится в начало редактируемой строки; также эта клавиша используется для возобновления последней введённой строки (см. ниже);

 — стирание символа (сдвигка строки); символ, на котором находится курсор, будет стёрт; символы от курсора до конца строки сдвинутся влево, заполняя освободившееся место;

 — раздвижка строки, символы от курсора до конца строки сдвинутся вправо и в строку вставится символ «ПРОБЕЛ»;

 («ЗАБОЙ») — символ перед курсором будет стёрт; символы от курсора до конца строки сдвинутся, заполняя освободившееся место; курсор переместится на позицию влево;

«ЗАБОЙ» в регистре «AP2»

— стирание всех символов от начала строки до курсора; оставшиеся символы (от курсора до конца строки) сдвинутся влево для заполнения освободившегося места.

→ — перемещение курсора на символ вперёд;

↑ — перемещение курсора вверх на соответствующую позицию предыдущей физической строки; если эта позиция не принадлежит редактируемой логической строке, функция игнорируется;

↓ — перемещение вниз на следующую физическую строку; работает аналогично предыдущей команде;

← в регистре «AP2» — переключение режима вставка/замена; в режиме замены вводимые символы помещаются на место, указанное курсором, замещая находящиеся там символы; в режиме вставки

символы от курсора до конца строки смещаются вправо, освобождая место для вводимого символа.

## 5.2. 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 БК0011 существует возможность вызвать последнюю введённую строку на редактирование и повторный ввод. Это достигается при помощи клавиши «ВС», на которую наложены две функции — перевод курсора в начало строки и возврат последней введённой строки.

Если строка уже начата, т.е. набрано несколько символов (даже если это пробелы), то «ВС» переводит курсор в начало или конец логической строки (см. подр. [5.1.](#)). 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 последней. Теперь её можно отредактировать обычным способом и ввести заново.

Клавишу «ВС» удобно также использовать для выполнения одного оператора или команды несколько раз подряд. При нажатии на клавишу «ВВОД» без единого символа в логической строке вводится пустая строка. Клавиша «ВС» после этого не позволит вернуть раньше введённых символов, поскольку последней введённой оказывается пустая строка.

## 5.3. Редактирование строк программы.

Особые случаи замены и удаления строк программы указаны в подр. [4.1.](#) Для редактирования текста какой-либо программной строки необходимо ввести следующую строку:

. <НОМЕР СТРОКИ>

Как и всегда, выполнение команды начинается после нажатия клавиши «ВВОД». 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 отредактированная;
- последняя выведенная командой LIST;

— последняя строка, в которой транслятор обнаружил ошибку.

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

Удаление строки путём набора её номера и нажатия на «ВВОД» (см. подр. 4.1.) также делает удалённую строку текущей. Но после такого удаления всякие обращения к текущей строке посредством точки приведут к ошибке 8.

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

#### 5.4. Команды редактирования

В Бейсик-системе существует ряд специальных команд, облегчающих ввод и редактирование программ. Их список приведён в таблице 1.

Таблица 1

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	удаляет всю программу на языке Бейсик
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, т. е. точку («.»).

Команду RENUM удобно использовать в случаях, когда строки плотно пронумерованы и для вставляемых строк нет свободных номеров.

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».

## 5.5. 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 «ВВОД»).

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 БК0011», необходимо практически изучить средства редактирования текста программы. Это существенно ускорит предстоящую работу.

## 6. ХРАНЕНИЕ ПРОГРАММ НА МАГНИТНОЙ ЛЕНТЕ

### 6.1. Форматы файлов

Бейсик-система допускает работу с тремя типами файлов:

- информация в кодах КОИ-8 (не более 255 символов в строке), (ASC);
- Бейсик-программы во внутреннем коде системы, (COD);
- файлы с двоичной информацией, (BIN).

Файлами первого типа являются Бейсик-программы, записанные на ленту при помощи команды SAVE (см. подр. [6.2.](#)) и файлы данных, созданных пользователем средствами Бейсик-системы. Программы можно прочитывать и операторами работы с файлами данных. Наоборот, команда LOAD требует, чтобы в файле были программные строки (в общем случае строки с допустимыми номерами), иначе загрузка такого файла приведёт к ошибке 57 (см. приложение). 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.

Для работы с файлами других типов также имеются специальные команды (см. подр. 6.2.). Их типы по умолчанию COD и BIN. Программы типа COD загружаются быстрее, чем типа ASC, 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 2. В ней указаны и специальные операторы и команды определения конфигурации системы, передачи управления системным или пользовательским программам в машинных кодах.

Таблица 2

оператор	действия
LOAD <ИМЯ> [,R]	загружает программу с указанным именем с магнитофона. При наличии буквы R производится пуск загруженной программы
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
FILES [<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 (с адреса A1 по адрес A2).
SYSTEM	выход в монитор команд оператора драйверной системы
CLEAR [L, A]	определение конфигурации системы
CALL	вызов дополнительного программного обеспечения (например, в кассете ПЗУ)

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> в строке NNNNN,  
где <ОШИБКА> — диагностическое сообщение о виде ошибки,  
NNNNN — 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

Пояснения к ошибкам 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 ожидания ввода команд. Продолжить выполнение программы операторами CONT или GOTO в этом случае нельзя.

Ошибочная программная строка всегда становится текущей.

При обнаружении любой ошибки в команде её обработка также немедленно прекращается, и система переводится в режим ожидания ввода текущей инструкции. Исключение составляют ошибки при выполнении INPUT и RENUM (см. описание языка).

## СООБЩЕНИЯ ОБ ОШИБКАХ

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

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. Кончились DATA

При выполнении оператора READ обнаружено, что список операторов DATA исчерпан.

### 5. Неправильный вызов функции.

Ошибка возникает в следующих случаях:

- отрицательный индекс;
- неправильный аргумент для LOG, USR;
- неправильные значения аргументов для графических и других операторов;
- неопределённая с DEFUSR функция USR и др.

### 6. Переполнение

Результат арифметической операции слишком велик по абсолютному значению и не может быть записан в формате, принятом для чисел в Бейсик-системе. (В случае слишком малых величин результат приравнивается нулю и ошибки не выдаётся).

### 7. Нет больше памяти.

Ошибка возникает в следующих случаях:

- программа не помещается в памяти;
- использовано слишком много вложенных операторов FOR или GOSUB;
- слишком много переменных;
- слишком много затребовано памяти для массива или области строк;
- слишком мала свободная область для генерирования кода непосредственно выполняемых операторов;
- в операторах CLOAD или BLOAD файл не помещается в

отведённое для него место в памяти;

— в операторе PAINT контур для закрашивания слишком сложный.

#### **8. Не определённый номер строки**

В GOTO, GOSUB, IF, RESTORE, RENUM, AUTO, ON или DELETE использован номер несуществующей строки.

#### **9. Несоответствие индекса.**

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 ноль**

В выражении встретилось деление на ноль; или ноль был возведён в отрицательную степень.

#### **12. Невозможно выполнить.**

Недопустимый оператор в режиме непосредственного выполнения или команда в косвенном режиме.

#### **13. Ошибка типов**

Попытка присвоить символьной переменной или функции, использующей символьный аргумент, числовое значение или наоборот.

#### **14. Кончилась память строк.**

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. Слишком сложное выражение.**

Слишком большое число вложенных скобок или функций MID\$ в выражении.

#### **17. Продолжение невозможно**

Была сделана попытка продолжить выполнение программы, которая:

- была прервана из-за возникновения ошибки;
- была изменена;
- не существует.

#### **18. Не определена функция пользователя**

Попытка обратиться к функции FN до её определения оператором DEFFN.

#### **19. Ошибка ввода/вывода.**

Встречается при работе устройства ввода/вывода.

#### **20. Не определена**

**21. Не определена**

**22. Не определена**

**23. Не определена**

**24. Отсутствует операнд**

Выражение содержит оператор без операнда, или в операторе (команде) нет обязательных параметров.

**25. Переполнение буфера ввода**

Была сделана попытка ввести строку длиной более чем 255 символов.

**26-51. Не определены**

**52. Недопустимое обращение к файлу.**

Оператор или команда ссылается на файл, который открыт с другой целью (не для записи или чтения).

**53. Файл не найден.**

В дисковых операциях не обнаружен файл с заданным именем

**54. Файл уже открыт**

Применён оператор OPEN для файла, который уже открыт.

**55. Конец 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. Неправильное имя файла**

В операторах LOAD, SAVE или OPEN использовано неправильное имя файла (имя файла пустое или состоит из слишком большого количества символов).

**57. Строка без номера.**

Команда непосредственного выполнения обнаружена в загружаемой программе во время выполнения оператора LOAD, загрузка прерывается.

**58. Не определена**

**59. Файл не открыт**

Оператор ввода/вывода применён к файлу, который не был открыт.

**60. Не определена**

**61. Не определена**

**62. Ошибочное имя устройства**

Было использовано ошибочное имя устройства.

.

