

Приложение D. Сведения о клавиатуре и экране микрокомпьютера Пылдин

Вывод символов вызывает появления их изображения на экране микрокомпьютера. Если программа использует стандартный выходной файл, он может быть перенаправлен к дисковому файлу. Например:

```
A:\>срс myprog > lst
```

Если стандартный выходной файл перенаправлен, на экране не будут показываться выводимые процедурой WRITE в файл OUTPUT символы. Но файла MESSAGE нельзя перенаправить, поэтому выводимые ими символы всегда показываются на экране.

Независимо от того в котором из файлов выводятся символы, для их изображения на экране используются стандартные (для этого компьютера) функции базовой системы ввода/вывода (BIOS). Они не изображают директно посланные им символы, ASCII код которых меньше 32 (\$20). Часть этих кодов являются управляющими символами системы BIOS. Если необходимо, чтобы их изображения появились на экране, надо сделать так, что управляющий символ ESC с ASCII кодом 27 (\$1B) предшествовал им.

Так как управляющие символы имеют соответствующее действие (см D.2), только если они посылаются системе BIOS, рекомендуется не выводит их в стандартном выходном файле. В противном случае, если он будет перенаправленным, эти символы запишутся в файле, к которому был перенаправлен output. Если файл будет обрабатываться какой либо программой, ожидающей нормального текста, могут произойти неожиданные ситуации. А если этот файл перенаправлен к печатающему устройству, неприятности гарантированы. На этот факт надо обращать внимание еще во время составления программы. Проще всего будет, если всегда файл MESSAGE поставлен первым в списке внешних файлов в заголовке программы. Если будете использовать и оба файла (MESSAGE, OUTPUT) одновременно, хорошо было бы составить программу так, что показываемый на экране текст не потерял смысл и был подходящим образом оформлен независимо от того перенаправлен ли будет стандартный выходной файл или нет. Необходимо тоже, что выходной файл содержал обычный текст, если будет перенаправлен. Чтобы файл содержал нормальный (обычный) текст, он должен удовлетворять следующим условиям:

- в нем не должны попадать символы, чей ASCII код меньше 32;
- в нем не должен попадать символ с ASCII кодом 127;
- переход к новой строке должен делаться только стандартной процедурой WRITELN.

Единственным символом с ASCII кодом меньше 32, которого можно использовать, является символ <FormFeed> (ASCII код 12 (\$0C)). Но этого делать тоже не рекомендуется, потому что не каждая программа соображалась бы с ним, если получит его при чтении с текстового файла.

По отношению клавиатуры и стандартного входного файла относится то же самое. Если используется стандартная процедура READ (файлом MESSAGE или INPUT, если не был перенаправленным), часть кодов нажатых

клавишей преобразуется, чтобы прочтенные с текстового файла коды и принятые с клавиатуры коды клавишей могли быть одинаковыми. Преобразование происходит согласно следующей таблице (даются сведения о том, используется ли соответствующий код в текстовом файле; те коды, которые не используются, получают новый код для полноты):

клавиша	ASCII код символа		процедуры READ
RETURN	13	\$0D	используется в текстовом файле
стрелка налево (←)	8	\$08	редко используется
стрелка направо (→)	7	\$08	не используется
стрелка вниз (↓)	10	\$0A	используется в текстовом файле
стрелка вверх (↑)	11	\$0B	не используется
TAB	9	\$09	используется в текстовом файле

Если используется функция для чтения символа (ReadKey), экспортированная модулем UniCRT, то коды клавишей совпадают вполне с теми, выдаваемые системой BIOS. Они даются в следующей таблице.

D.1. Коды клавишей микрокомпьютера Пылдин 601/601A/601M

Клавиша		shift	control
F1	201 (\$c9)	213 (\$d5)	225 (\$e1)
F2	202 (\$ca)	214 (\$d6)	226 (\$e2)
F3	203 (\$cb)	215 (\$d7)	227 (\$e3)
F4	204 (\$cc)	216 (\$d8)	228 (\$e4)
F5	205 (\$cd)	217 (\$d9)	229 (\$e5)
F6	206 (\$ce)	218 (\$da)	230 (\$e6)
F7	207 (\$cf)	219 (\$db)	231 (\$e7)
F8	208 (\$d0)	220 (\$dc)	232 (\$e8)
F9	209 (\$d1)	221 (\$dd)	233 (\$e9)
F10	210 (\$d2)	222 (\$de)	234 (\$ea)
F11	211 (\$d3)	223 (\$df)	235 (\$eb)
F12	212 (\$d4)	224 (\$e0)	236 (\$ec)
ESC	27 (\$1b)	27 (\$1b)	27 (\$1b)
TAB	237 (\$ed)	238 (\$ee)	237 (\$ed)
BackSpace	127 (\$7f)	127 (\$7f)	239 (\$ef)
RETURN	192 (\$c0)	192 (\$c0)	240 (\$f0)
left (←)	193 (\$c1)	197 (\$c5)	241 (\$f1)
right (→)	194 (\$c2)	198 (\$c6)	242 (\$f2)
down (↓)	195 (\$c3)	199 (\$c7)	243 (\$f3)
up (↑)	196 (\$c4)	200 (\$c8)	244 (\$f4)
F13 Home (s-left)	197 (\$c5)	197 (\$c5)	245 (\$f5)
F14 End (s-right)	198 (\$c6)	198 (\$c6)	246 (\$f6)
PageDown (s-down)	199 (\$c7)	199 (\$c7)	247 (\$f7)
PageUp (s-up)	200 (\$c8)	200 (\$c8)	248 (\$f8)
Delete	249 (\$f9)	249 (\$f9)	249 (\$f9)
F15 Insert	250 (\$fa)	250 (\$fa)	250 (\$fa)
* Lat/Кип	251 (\$fb)	251 (\$fb)	251 (\$fb)
* CapsLock	252 (\$fc)	252 (\$fc)	252 (\$fc)
* Pause (s-space)	253 (\$fd)	253 (\$fd)	253 (\$fd)
* Break (c-space)	254 (\$fe)	254 (\$fe)	254 (\$fe)

Последние четыре клавиши (помеченные звездочкой) не могут быть получены при помощи функции `ReadKey`. Они немедленно обрабатываются системой BIOS и не достигают пользовательской программы. Их коды даны для полноты.

D.2. Таблица управляющих символов системы BIOS

Dec	Hex	Ctrl	Char	Имя	Смысл в ASCII
0	00	^@	NUL	null	end of string
7	07	^G	BEL	bell	bell
8	08	^H	BS	backspace	backspace
9	09	^I	HT	horizontal tab	tab
10	0a	^J	LF	line feed	new line
12	0c	^L	FF	form feed	clear screen
13	0d	^M	CR	carriage return	begin of line
22	16	^V	SYN	synchronous idle	set video mode
24	18	^X	CAN	cancel	clear to end of line
27	1b	^[ESC	escape	escape
127	7f		RUB	rubout	rubout

NUL (**\$00**) используется только ассемблерскими подпрограммами;

BEL (**\$07**) микрокомпьютер воспроизводит короткий звуковой сигнал;

BS (**\$08**) передвигает курсор на предыдущую позицию;

HT (**\$09**) передвигает курсор на следующую позицию, кратную 8;

LF (**\$0A**) курсор переходит в начало следующей строки;

FF (**\$0C**) очищает экран и передвигает курсор в верхний левый угол;

CR (**\$0D**) передвигает курсор в начало строки;

CAN (**\$18**) очищает все символы с текущего положения курсора до конца строки, не передвигая курсор;

ESC (**\$1B**) первый символ, посылаемый после **escape**, визуализируется на экране. Используется для визуализации управляющих символов;

RS (**\$1e**) следующие два символа, посылаемые к системе BIOS, не визуализируются, а используются для определения нового местоположения курсора;

RUB (**\$7F**) вытирает предыдущий символ (по отношению местоположения курсора).

