

## Приложение D. Сведения за клавиатурата и екрана на микрокомпютъра Пълдин

При извеждане на символи на екрана на микрокомпютъра се визуализират техните изображения. Ако програмата използва стандартния изходен файл, то възможно е той да бъде пренасочен към дисков файл. Например:

```
A:\>upc myprog > lst
```

Ако стандартният изходен файл е пренасочен, на екрана няма да се визуализират извежданите чрез процедурата WRITE във файла OUTPUT символи. Файлът MESSAGE не може да бъде пренасочен, затова извежданите в този файл символи винаги се визуализират.

Независимо в кой файл се извеждат символите, за визуализирането им се използват стандартните (за този компютър) функции на базовата/входно изходната система (BIOS). Те не визуализират директно изпратени символи с ASCII код, по-малък от 32 (\$20). Част от тези символи са управляващи символи за BIOS. Ако е необходимо извеждането им на екрана, всеки от тях трябва да бъде предхождан непосредствено от символ с ASCII код 27 (\$1B) (управляващ символ ESC).

Тъй като управляващите символи имат съответното действие (виж Таблица на управляващите символи за BIOS), само когато се изпращат към BIOS, не е желателно да се извеждат във стандартния изходен файл. Това е така, защото ако той бъде пренасочен, тези символи ще се запишат във файла, към който е пренасочен файлът OUTPUT. Ако той бъде пренасочен към дисков файл и след това се обработва с някаква програма, очакваща нормален текст, могат да се получат неприятни изненади. А ако това е печатащо устройство, неприятностите са гарантирани. Добре е още при съставянето на програмите това да се има предвид (най-просто е, ако винаги указвате файла MESSAGE като първи файл в списъка от външни файлове в заглавната част на програмата). Ако използвате и двата файла (MESSAGE и OUTPUT) едновременно, добре е програмата да бъде съставена така, че визуализираният на екрана текст да бъде смислен и подходящо оформен, независимо от това дали стандартният изходен файл е пренасочен или не. Необходимо е и изходният файл да съдържа нормален текст, ако се пренасочи. За да има един файл винаги нормален текст, в него трябва да се изпращат само символи с ASCII код по-голям от 31 (\$1F), да не се използва символът с ASCII код 127 (\$7F) и преминаването на нов ред да става само със стандартната процедура WRITELN. Единственият символ с ASCII код по-малък от 32, който може да си позволите да използвате, е <FormFeed> - ASCII код 12 (\$0C). Но и това не се препоръчва, защото не всяка програма ще се съобрази с него, ако го получи при четене от текстов файл. (Вашите програми съобразяват ли се?)

При клавиатурата и при стандартния входен файл нещата стоят по аналогичен начин. Ако се използва стандартната процедура read (независимо от кой файл - MESSAGE или INPUT, ако не е пренасочен), част от кодовете на натиснатите клавиши се преобразуват за да може прочетените от текстов файл и възприетите от клавиатурата кодове да са съответно еднакви. Преобразуването става по следната таблица (показано е използването на съответния код в текстов файл; тези кодове, които не се използват, се прекодират за пълнота):

клавиш	ASCII код на символа от процедурата READ	
RETURN	13 (\$0D)	използва се в текстов файл
стрелка наляво (←)	8 (\$08)	рядко се използва.
стрелка надясно (→)	7 (\$07)	не се използва
стрелка надолу (↓)	10 (\$0A)	използва се в текстов файл
стрелка нагоре (↑)	11 (\$0B)	не се използва
TAB	9 (\$09)	използва се в текстов файл

Ако се използва функцията за четене на символ (ReadKey), експортирана от модула UniCRT, то кодовете на клавишите съвпадат напълно с тези, които дава BIOS. Те са дадени в следната таблица.

### D.1. Кодове на клавишите за Пълдин 601 / 601A / 601M

Клавиш		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/Кирп	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)

Кодовете на последните четири клавиша (отбелязани със \*) не могат да бъдат получени чрез функцията ReadKey. Те се обработват незабавно от (BIOS) и не достигат до потребителските програми. Показани са само за пълнота.

## D.2. Таблица на управляващите символи за BIOS

Dec	Hex	Ctrl	Име	значение в ASCII	Действие на BIOS
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) използва се само в подпрограми на асемблер;

**BEL** (\$07) микрокомпютърът издава кратък звуков сигнал;

**BS** (\$08) премества курсора една позиция вляво;

**HT** (\$09) премества курсора на следващата кратна на 8 позиция;

**LF** (\$0A) курсорът преминава в началото на следващия ред;

**FF** (\$0C) изчиства екрана и поставя курсора в горния ляв ъгъл;

**CR** (\$0D) премества курсора в началото на реда;

**CAN** (\$18) изтрива всички символи от текущото положение на курсора до края на реда, без да премества самия курсор;

**ESC** (\$1B) първият символ, изпратен след него (escape), се визуализира на екрана. Използва се за визуализация на управляващите символи;

**RS** (\$1e) следващите два символа, изпратени към BIOS, не се визуализират, а се използват за определяне на новото местоположение на курсора на екрана;

**RUB** (\$7F) изтрива символа, намиращ се непосредствено пред курсора и премества курсора на това място.

