PROTYPE

PRINT ENHANCER

Amstrad CPC 6128
and CPC 6128 Plus

PROTYPE

PRINT ENHANCER

Amstrad CPC6128 and 6128+
Amstrad CPC464/664 with 64K expansion and disc drive

 Issue 1, 1991 (for v1.0)

Program written by Mark Davies.

The Protype program was developed using the Maxam assembler ROM.
This manual was produced using Protype and the Protext word processor ROM.
Maxam and Protext are both available from Arnor.

This manual was written using Protext and Protype and printed from camera-ready copy produced by Protype on a Star LC-10 dot matrix printer.

Arnor Ltd., 611 Lincoln Road, Peterborough PE1 3HA, England.

CONTENTS 3

1. About Protype

1.1 What is Protype?

Protype is a versatile typesetting program designed to squeeze the absolute maximum print quality from low cost 9-pin and 24-pin Epson compatible dot matrix printers. It does this by using up to twelve passes of the print head to produce each line of text. Protype characters are of varying width (proportionally spaced) as are the characters of many other programs, but Protype incorporates additional sophisticated spacing techniques normally only found in phototypesetting or laser printer applications (see §A5).

You cannot use Protype for editing text; it has been developed to work together with other programs, particularly word processing software such as Arnor's Protext. When installed in its simplest form, Protype takes up less than 350 bytes of the memory available for text and programs in the main 64K, so you can use it with almost any program which produces printer output, provided the program doesn't need the second bank of memory.

A starter library of seven basic Roman alphabet fonts (character sets) is supplied, each of which also contains a wide range of non-standard (non-ASCII) characters, symbols and diacritics (accents etc.). You can combine any number of these diacritics with any character creating, should the need arise, combinations such as a grave accent and macron over and an ogonek under the letter e (as in Old Lettish bę̄rns = child). Protype can cope with over 40 European languages.

There are many other features; standard ones such as $^{\text{super}}$scripts, $_{\text{sub}}$scripts or underlining and other less usual ones like boxing [x], encircling (x) and .pnipsmi ɹoɹɹim

1.2 Using the manual

If you will be using Protype with the Protext word processor then start reading at part 2 of this manual. Some familiarity with Protext is assumed. (NB. if you have Arnor's Promerge Plus ROM you will find that installing Protype disables certain features of Promerge (see §4.2).)

If you intend to use Protype mainly with BASIC or machine code programs you should start reading at part 3 of the manual. The text assumes some familiarity with BASIC programming.

If you wish to use Protype from other commercial programs or from machine code with a purely ASCII printer output then it is possible to start reading at part 4. However, if printer control codes or special characters are required it is advisable to work through part 3 first.

2. Using Protype from Protext

This part of the manual assumes you have some experience of using Arnor's Protext word processor. You are advised to familiarise yourself with the basics of Protext before reading on.

2.1 Getting started

Before using Protype you should make a working copy of both sides of the disc supplied onto a new disc. With a CPC6128 this is done using the DISCKIT3 program on the CP/M+ system disc supplied or Utopia's !DISCCOPY. Follow the procedure in the manual. Work using the copy and keep the original safe as a backup. Never close the write protect holes on the original disc. The original and copy should be for your personal use only.

The method of installing Protype depends on whether you have the disc version or ROM version of Protext.

2.1.1 Disc version

Reset the computer by pressing CONTROL, SHIFT and ESC together.
Insert the Protype disc into drive A with side B uppermost.
Enter the command RUN "DISC" RETURN
(Protype will now be loaded with the default fonts.)
Remove the Protype disc.
Insert the Protext disc into drive A.
Enter the command RUN "DISC" RETURN
(Protext will now be loaded.)
Remove the Protext disc.
Insert the Protype disc into drive A with side A uppermost.
Enter the command RUN "P" RETURN
(Protype will now configure and enter Protext.)
Read §2.1.3.

2.1.2 ROM version

Reset the computer by pressing CONTROL, SHIFT and ESC together.
Look at the screen message to check that the Protext ROM is initialised.
Insert the Protype disc into drive A with side A uppermost.
Enter the command RUN "DISC" RETURN
(Protype will now be loaded with the default fonts.)
(Protext is configured automatically and entered.)

2.1.3 First steps

On entering **Protext** the cursor appears beside the command prompt >. Type INFO RETURN in response to this prompt and a table will appear. This table contains information about the current (in this case default) status of **Protype**. Looking at the left of the table you will see that **Protype** is set to print in Near Letter Quality (NLQ) and that the names of the three fonts in memory are "KlassiK.626", "Lucca.636", and "Mikron.427". The meaning of the information to the right of the table will be covered in §2.5 and §2.6.

When the command prompt appears again type PROTYPE 0 RETURN. This switches **Protype** out (off). If you try to view the status table now using INFO RETURN the message "PROTYPE is off" will appear. Switch **Protype** in (on) again by typing PROTYPE RETURN in response to the prompt.

You are probably familiar with the printer driver which **Protext** uses to determine how text and control codes are sent to the printer port. When the **Protext** version of **Protype** is initialised it creates a second printer driver in memory for use when **Protype** is switched in. This second printer driver contains values and codes more appropriate for the **Protype** fonts. Whilst **Protype** is switched in the driver may be altered, saved or replaced in precisely the same way as the first printer driver. With **Protype** switched out **Protext** behaves exactly as it would if **Protype** were not installed. The **Protype** printer driver is swapped for the original printer driver and the codes stored in this driver are sent directly to the printer port.

2.2 Printing

To produce an example of **Protype** printout press ESC to go into **Protext** edit mode and type in a line of text (up to 255 characters). Press ESC again to return to command mode. Check that the printer is switched on, connected to the computer, on line, and loaded with paper. If the printer has a DIP switch to set the buffer mode to graphics this should be ON. If the printer allows you to select between friction or traction drive then set it to friction for the best results. Print by typing P RETURN at the command prompt as you would do normally when printing from **Protext**. The line of text will be printed with the default top and side margins in high quality (NLQ) characters. Be patient. If the resultant typeset line is over 20 cm long then the surplus will be ignored. The font used is font K, in this instance "KlassiK.626".

The printout will have been slower than you are probably used to; normally this quality of text would only be used for a finished document. To print the same text six times faster but with poorer quality (lower resolution), wait till the command prompt reappears and type **DRAFT** RETURN. You can check that **Protype** has switched to draft printing mode by typing **INFO** RETURN to view the **Protype** status table. Now print the line again. Typing **NLQ** RETURN will switch **Protype** back to high quality mode.

You can interrupt printing at any time by pressing the ESC key. After a second or two the printer should stop (N.B. if the printer has a large buffer it may be more convenient to switch off the printer first). Pressing ESC again will cause the cursor to appear on the screen, and pressing ESC a third time will return you to **Protext** command mode. Once a printout has been interrupted using the ESC key, printing may only be resumed by sending another print command. This has the effect of resetting **Protype**.

Printing a block of text (PB), printing to the screen (PS) and printing to a file (PF) all work normally from **Protext** command mode with **Protype** installed.

2.3 **Fonts**

2.3.1 Selecting fonts

You will have seen from the status table that the fonts in the memory at any one time are identified by the letters K, L and M. These fonts are known as the *resident fonts*. The initials of the three default fonts match these letters but the correspondence is merely an aid to memory; any font loaded into memory may by identified by any of the three letters. A particular resident font is selected within the text by inserting the relevant **Protext** printer control code K, L or M. These are the letters which appear on the screen in inverse video. Since you will probably be using printer control codes extensively, the function key **f0** has been defined as a printer control key. In edit mode, pressing **f0** followed by a letter will produce the corresponding printer control code. Until a resident font is specifically selected using an embedded code, the text will be printed using font K.

Go back into edit mode and pick a word or phrase that you want to emphasise in the line you have just printed. Insert the printer control code T before and the code F after your chosen section of text like this:

```
The Tresident fontsF are the fonts in the computer memory.
```

Now return to command mode and print the line again. Font L, "Lucca.636" is an italic cursive font so your chosen text will be printed italicised:

The *resident fonts* are the fonts in the computer memory.

2.3.2 Loading fonts

Any of the resident fonts may be changed by loading a new font in its place. Make sure the Protype disc is in drive A (any side up). Now from Protext command mode type FONT L SKYBOLD.646 RETURN. The font should be loaded from disc into memory. When the command prompt appears type INFO RETURN to confirm that the italic font "Lucca.636" has been replaced by the bold sanserif font.

If you need more than three fonts for a particular document it is possible to load fonts automatically whilst printing using a stored command within the text. To assign "skyboLd.646" to resident font L the command would be:

```
>EX FONT L SKYBOLD.646
```

This should be entered as a separate line of text. If you print using stored commands to load fonts it is always wise to send the print output to the screen first (using PS). This allows bad parameters or spelling errors in the font name to be reported without wasting paper. Make sure the disc in drive A contains the fonts.

If a "file not found" or "bad parameter(s)" error occurs whilst printing to paper, Protype stops, bleeps and waits for you to press the ESC key.

The starter fonts included with Protype are:

KlassiK	.626	Medium upright serif
Lucca	*.636*	*Medium italic cursive*
Mikron	.427	Small medium upright serif
skyLite	.606	Light upright sanserif
skyKite	**.626**	**Medium upright sanserif**
skyMite	.407	Small light upright sanserif
skyboLd	**.646**	**Bold upright sanserif**

2.4 Toggled features

Protype recognises three special embedded printer codes which are used for superscripts (raised text), subscripts (lowered text) and underlining. These three codes are X, Y and Z respectively. In each case you use the same code to turn the feature on and off. This is called toggling so superscripting, subscripting and underlining are known as toggled features.

2.4.1 Underlining

Returning to edit mode replace each of the codes T and K (on either side of your italicised text) with the code Z:

The Zresident fontsZ are the fonts in the computer memory.

When the line is printed the relevant text will be underlined:

The resident fonts are the fonts in the computer memory.

If you are printing in adjust mode (see §2.6) it sometimes advisable to link the words of underlined phrases using non-break spaces (keypress CONTROL N) to avoid small breaks in the underlining.

2.4.2 Superscripts and subscripts

Any section of text may be raised or lowered with respect to the baseline using the printer control codes X or Y. In practice, however, the larger fonts will tend to lose part of their ascenders (b,d,f,h,k,l and upper case letters) or descenders (g,j,p,q,y) off the top or bottom of the line. As a consequence, superscripts and subscripts are best printed using one of the smaller fonts (Mikron.427 or skyMite.407). If you have loaded new fonts check that resident font M contains one of these two files. Now experiment with superscripts and subscripts remembering to select fonts as well as toggling the raising or lowering of the text eg.:

The density of ethanol (CHmyZyXHmy5yXOH) at STP is 789 kg mmX-3xX.

This will print as:

The density of ethanol (CH_2H_5OH) at STP is 789 kg m^{-3}.

If you are accustomed to using a single printer code to toggle one of these functions on and off, the above procedure will seem rather involved. The Protext printer control codes can of course be redefined to do this (see §2.8) but some other aspect of Protype will probably be lost unless codes are redefined very carefully. It will usually only be worth your while if you use a particular feature extensively.

2.5 Further Protype commands

You have already met several of Protype's commands INFO, DRAFT, NLQ and FONT. Protype itself is switched in and out using the command PROTYPE [0]. The advantage of this type of command is that it can be used from many very different programs. You have seen how a font can be loaded directly from Protext command mode using FONT K KLASSIK.626 RETURN or automatically whilst printing using >EX FONT K KLASSIK.626 as a line within the text. You could load the same font during the execution of a line of BASIC program using (for example) 10 !FONT,"K","KLASSIK.626".

RSX commands then, are the key to the versatility of Protype. This section introduces the remaining Protype commands.

ADJUST
: Switches the formatting mode of Protype, stretching spaces to adjust the position of the right hand end of lines. This mode is used in conjunction with Protext right-justification (see §2.6).

FREE
: Switches the formatting mode of Protype so that lines occupy the minimum width consistent with aesthetic spacing of the text (see §2.6).

TABLE
: Switches the formatting mode of Protype to facilitate tabulation (the vertical alignment of columns of text or figures (see §2.6)).

GAP *n*
: Sets the microscopic spacing between characters; the standard spacing is selected by GAP 0. The decrease or increase in spacing is given by *n*/120 inches where *n* is a number in the range -2 to +2. In BASIC negative values may be included directly in the command eg. !GAP,-1. In Protext the value must be added to 256 so !GAP,-1 as a Protext stored command would be >EX GAP 255 i.e. (256-1). The command GAP 1 is sometimes convenient if text is to be photoreduced.

LFEED *n*
: Sets the linefeed spacing (the vertical distance occupied by a line of text). The line spacing is set to *n*/216 inches for 9-pin printers (*n*/180 inches for 24-pins) where *n* is a number in the range -72 to +72. The default value is 39 (13 point). In BASIC negative values (i.e. backfeeds) may be included directly in the command eg. !LFEED,-39. In Protext the value must be added to 256 so !LFEED,-39 written as a Protext stored command would be >EX LFEED 217 i.e. (256-39). (NB. some printers will not allow backfeeds if friction drive is being used.)

CPI *n*

Sets the nominal pitch in characters per inch for justification and tabulation purposes. Since fonts are proportionally spaced the figure is a notional average value which serves to calculate the absolute horizontal position of columns or margins but has no meaning with respect to the actual number of characters in any given inch. Permitted values for *n* are 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 15, 17 and 20. Most of these values are included to permit future development of Protype. The main values of any practical use with the fonts supplied are CPI 12, CPI 15 and CPI 17.

CPCM *n*

As for CPI *n* except that values are given in characters per centimetre. The default value of *n* is 6 which is suitable for all the fonts supplied (6 characters per cm is a little over 15 characters per inch). CPCM 7 may be used if text only contains the fonts Mikron.427 and skyMite.407.

MIRROR [0]

Issuing the command MIRROR causes all subsequent text to be flipped so the line begins at the right of the printer carriage and the text image is reflected right to left. This will be of use in future developments where, for instance, an Arabic or Hebrew script output is required from a Latin script screen display. When mirror imaging is selected this is indicated in the status table. This feature is cancelled using MIRROR 0.

2.6 Formats

2.6.1 Formatting modes

Protype's three formatting modes, Adjust, Free, and Table are selected using the RSX command of the same name. Adjust mode does not in itself right-justify text; what it does is adjust the length of each line so that it is proportional to the number of characters in the line. Thus if the Protext text has a 'ragged' right margin, so will the printout even if Adjust mode is selected. This paragraph is printed in this way.

If the Protext text is right-justified so is the Protype output even though Protext works in columns and the Protype output is proportionally spaced. Because it can work with both text formats, Adjust mode is the Protype default mode. The position in the line of the 'soft spaces' which Protext uses to justify is unimportant. Protype doesn't print the soft spaces it merely counts them and then stretches each of the hard spaces to compensate producing a well-balanced appearance. To calculate the printed width of a block of right-justified text (in centimetres or inches), divide the number of occupied screen columns by the current Protype pitch (in characters/cm or characters/inch). The default pitch is 6 characters per centimetre.

If you wish to print mainly ragged right text then Free mode may be more appropriate. This paragraph is printed using Free mode. The advantage is that the spaces between words are equal throughout giving the body of the text a more uniform appearance. Additionally if the number of characters in consecutive lines happens to coincide by chance whilst using Adjust mode, this section of the text will appear justified and therefore slightly odd. In Free mode, the right margin is usually slightly more irregular than when ragged right text is printed in Adjust mode. For occasional text such as headings Free mode can be simulated by using non-break spaces between words.

Table mode is included to enable information to be easily presented in a tabular form. An absolute horizontal position is calculated for any character preceded by a space. This position is based on the number of previous characters in the line regardless of their widths. So any words vertically aligned on the Protext screen display will remain vertically aligned in the printout even though they are proportially spaced. Any tabulated phrases should be held together with non-break spaces. If this is not done a separate horizontal position will be calculated for each word producing uneven results. Table mode is unsuitable for bulk text.

2.6.1 Line graphics characters

Ruled tables may be produced in any formatting mode by using a set of line graphics characters whose horizontal positions are always calculated. The Protext configuration redefines the function keypad so that these characters may be typed using the shifted keys f0 to f9 and the unshifted and shifted decimal point key. A set of key stickers is provided with the program as a memory aid for the new definitions. The expansion strings 151 to 159 are also available to the user (or to Arnor's Utopia ROM if fitted) by pressing CONTROL together with the appropriate key f1 to f9. All these key definitions are summarised in §A2.

One special line graphics character is produced by unshifted f6. This is a tabulation character which is visible on the screen but invisible in the printout. If placed before a character it fixes the horizontal position of that character enabling unruled tables to be produced even whilst in Adjust or Free modes.

The unshifted f9 character is a 'dummy' character which occupies a screen column but which occupies no width in the final printout. It can be regarded as a 'soft' space under the control of the user rather than Protext. Used together with Adjust mode it allows any line to be padded out to a given length whilst maintaining even spacing.

2.7 **Diacritics and modifiers**

You have already seen that Protype uses the Protext printer control codes K, T and m for font selection (§2.3) and the codes X, Y and Z to toggle features common to all fonts (§2.4). The remaining printer control codes are in theory free for each font to interpret as suits it. In practice, similar fonts follow the same conventions.

The conventions for Latin fonts are as follows:

a b c d f g h I J n o q t u	are diacritics wc → ŵ
e p r s	are modifiers ss → ß
v w	may vary from font to font

2.7.1 Diacritics

A diacritic is a mark (for example an accent) added to a character to show a change in meaning or phonetic value. The Protext configuration allows for a small number of common European diacritic/character combinations to be displayed on the screen by pressing one of the unshifted function keys f1, f2, f4, f5 or f7 followed by the character. While in Protext edit mode press f1 followed by a. The diacritic/character combination á appears on the screen. A full table of these combinations is given in §A2. You can fix the supplied self-adhesive key stickers to the function keys so that you don't need to remember their new definitions.

The above procedure is fine if you want to put an acute accent over the letter a but what if you want to do something less common like placing an acute accent over the letter s to transcribe a sound which occurs in many languages of the Indian sub-continent? The printer control code a is used to represent an acute accent so the screen sequence sa will be printed as ś.

The printer codes to represent certain diacritics in Latin script applications have also been chosen to be meaningful on-screen. The háček ˇ is used over the c and s in many East European languages to represent the sounds which in English are written ch and sh. Since the printer control code h is used for the háček, Czechoslovakian č and š appear on-screen as ch and sh. The full list of diacritic printer codes is given in §A3.

2.7.2 Modifiers

The Protext configuration also enables several common non-ASCII symbols to be displayed on the screen. While in Protext edit mode press f8 followed by the s key. The German ß symbol (sometimes written ss) will appear on the screen. Now type f3 followed by s to produce the section symbol §. A full list of the characters and symbols available using these two function keys (called 'local' and 'extra') is given in §A2.

There are many more symbols available in print than on-screen. These are generally represented in the text by a character or pair of characters followed by one of the modifier control codes ¯, ˘, ˚ or ¸. The digraph Æ, for example is represented by the sequence AE¸. Some of the more language specific character/diacritic pairs such as Latvian ġ, Hungarian ő, Maltese ħ, or Romanian ţ are also represented using modifiers rather than the more general diacritics. Occasionally a character pair may combine to form a single character; the on-screen sequence -> prints as →. A full list of these combinations is given in §A3.

2.8 Configurations

It may be that the default configurations of Protype and Protext are not suitable for your own purposes. You can configure Protype yourself either before or after Protype is initialised by chaining short BASIC programs. The examples in this section assume that your configuration programs are on separate disc from your working copy of Protype but with a little modification they may be included on the same disc.

2.8.1 Configuring before Protype is initialised

This is the best option if you want to change the default fonts. Configuring beforehand avoids loading the default fonts and then immediately substituting them. Here is an example program:

```
10 fontK$="skyKite.626": fontM$="skyMite.407"
20 PRINT "Insert Protype disc and press any key"
30 i$=INKEY$: IF i$="" THEN 30 'wait for keypress
40 CHAIN "DISC"
```

If the program is on a separate disc from Protype then you can call it "DISC".
If the program is on the same disc as Protype you don't need lines 20 and 30.
To configure keys etc. after installing Protype (§2.8.2) use 40 CHAIN "PROTYPE"

2.8.2 Configuring after Protype is initialised

In order to do this you need to initialise Protype with the command RUN "PROTYPE" rather than RUN "DISC". The program will then stop at some point and ask for a disc containing CONFIG.BAS to be inserted. Here is an example of what CONFIG.BAS might look like:

```
10 INK 0,0: INK 1,26                'set colours to maximum contrast
20 name$="letter":!PRINTER,@name$ 'load a printer driver
30 KEY 150,"!INFO"+chr$(13)         'expansion string to CONTROL f0
40 !DRAFT                           'switch to draft printing mode
50 !P                               'enter Protext
```

The program CONFIG.BAS already present on side A of the Protype disc loads a printer driver which redefines the printer control codes to allow you to print files you may have written previously using the more conventional meanings of the codes. Should you wish to redefine the Protext codes yourself the numeric (hexadecimal) values corresponding to the Protype printer control codes can be found in §A4.

NB. The program which sets up the new key and matrix definitions for use from Protext, automatically disables the Protext language options so that selecting one of these produces the message 'Cannot extend symbol table'. This has been done to avoid the confusion of duplicate matrices being defined by Protext. Since the symbol table is 'locked up' it cannot be collapsed using, for example, SYMBOL AFTER 256. By including the instruction MEMORY HIMEM+1 in CONFIG.BAS the symbol table may be subsequently extended or collapsed and the Protext language options used if required.

2.9 Tips

If you wish to print a diacritic on its own this is best done by typing a non-break space followed by the diacritic's printer control code.

If you need to place diacritics both above and below a lower case character (eg. a háček and a jot) type the printer control code for the top diacritic first. Otherwise Protype won't recognise the character as lower case and will place the háček too high. o<^ prints as ǒ. o<^. prints as ǫ.

If you need to print two consecutive characters that are normally interpreted as one you can separate them with an unused control code. <- prints as ←. <v- prints as <-.

You can perform a spelling and parameter check on all your stored Protype commands without having to print on paper or to the screen (as described in §2.3.2). At the end of your text define a block to contain only a blank line. Now, making sure that a disc containing any necessary fonts is in the relevant drive, print using the Protext command PB (Print Block). In order to ensure that the (dummy) block is printed correctly, Protext executes all the preceeding stored commands and so allows Protype to report any errors.

When printing right-justified text in Adjust mode it is just possible that a line containing many wider-than-average characters such as m and w will not fit into the space available and will protrude beyond the right margin in the printout. If this ever happens re-edit the line by joining the last word in the offending line to the first word in the following line using a non-break space. Then reformat the paragraph.

The page length of continuous stationery may not be exactly divisible by your chosen line feed so that the number of lines per page cannot be easily set. Alternatively you may regularly change line spacings or use backfeeds so that the exact vertical position of the next page is difficult to keep track of. In both these cases the easiest solution is to enable form feeds so that your printer keeps track of where you are. This is done from Protext using the stored command >FF ON.

If you don't anticipate printing anything during a particular session, you may wish to use the Protext editing facilities together with Protype's key and matrix definitions but without having to install Protype itself or any of the fonts. From BASIC (and with Protext installed) insert the Protype disc in drive A with side A uppermost and type RUN "K" RETURN. The keys and matrices will be configured and Protext entered. (NB. This procedure also allows you to use Promerge Plus features such as box mode and 2-file editing.)

3. Using Protype from BASIC

This part of the manual assumes you have some experience of programming in BASIC and that you understand hexadecimal notation. If you are not familiar with hexadecimal numbers you are advised to read the relevant part of §A5 before continuing further.

3.1 Getting started

Before using Protype you should make a working copy of both sides of the disc supplied using a new disc. With a CPC6128 this is done using the DISCKIT3 program on the CP/M+ system disc supplied or Utopia's !DISCCOPY. Follow the procedure in the manual. Work using the copy and keep the original safe as a backup. Never close the write protect holes on the original disc. The original and copy should be for your personal use only.

3.1.1 Installing Protype

Reset the computer by pressing CONTROL, SHIFT and ESC together.
Insert the Protype disc into drive A with side B uppermost.
Enter the command RUN "DISC" RETURN
(Protype will now be loaded with the default fonts.)

3.1.2 First steps

When the message Ready and the BASIC cursor appear type !INFO RETURN and a table will appear. This table contains information about the current (in this case default) status of Protype. Looking at the left of the table you will see that Protype is set to print in Near Letter Quality (NLQ) and that the names of the three fonts in memory are "KlassiK.626", "Lucca.636", and "Mikron.427". The meaning of the information to the right of the table will be covered in §3.5 and §3.6.

When the BASIC cursor appears again type !PROTYPE,0 RETURN. This switches Protype out (off). If you try to view the status table now using !INFO RETURN the message "PROTYPE is off" will appear. Switch Protype in (on) again by typing !PROTYPE RETURN.

3.2 Printing

To produce an example of Protype printout type in the following BASIC line enclosing whatever text you want (up to 255 characters) within the quotes:

```
10 PRINT #8,"Enclose any text you like."
```

Check that the printer is switched on, connected to the computer, on line, and loaded with paper. If the printer has a DIP switch to set the buffer mode to graphics this should be ON. If your printer allows you to select friction or tractor drive then you will get better results using friction. RUN your single line BASIC program. The line of text will be printed in high quality (NLQ) characters. Be patient. If the resultant typeset line is over 20 cm long then the surplus will be ignored. The font used is font K, in this instance "KlassiK.626".

The printout will have been slower than you are probably used to; normally this quality of text would only be used for a finished document. To print the same text six times faster but with poorer quality (lower resolution), wait till BASIC is Ready again and type !DRAFT RETURN. You can check that Protype has switched to draft printing mode by typing !INFO RETURN to view the Protype status table. Type RUN RETURN to print the line again. Typing !NLQ RETURN will switch Protype back to high quality mode.

You can interrupt printing at any time by pressing the ESC key. After a second or two the printer should stop (N.B. if the printer has a large buffer it may be more convenient to switch off the printer first). Pressing ESC again will cause the cursor to appear on the screen, and pressing ESC a third time will return you to BASIC command mode. Once a printout has been interrupted using the ESC key, printing may only be resumed by sending the reset code &E0 to Protype using PRINT #8,CHR$(&E0). Alternatively you can switch Protype out and in again using !PROTYPE,0: !PROTYPE. This also has the effect of resetting Protype.

It is advisable to send a reset code at the beginning of a printout anyway since it causes the slack in the printer's paper feed mechanism to be taken up before printing the first line.

3.3 Fonts

3.3.1 Selecting fonts

You will have seen from the status table that the fonts in the memory at any one time are identified by the letters K,L and M. These fonts are known as the *resident fonts*. The initials of the three default fonts match these letters but the correspondence is merely an aid to memory; any font loaded into memory may by identified by any of the three letters. A particular resident font is selected by printing the relevant control code to stream #8. The three font selection codes are &EB, &EC and &ED; the symbols used here to designate these codes are K, L and M respectively (There are 26 printer control codes &E1 to &FA which are represented by the printer control letters a to z). Until a resident font is specifically selected by 'printing' one of the three font select codes, the text will be printed using font K.

Pick a word or phrase that you want to emphasise in the line you have just printed. Editing the BASIC line insert the printer control code L before and the code K after your chosen section of text like this:

```
10 PRINT #8,"Enclose ";CHR$(&EC);"any";CHR$(&EB);" text you like."
```

Now RUN the BASIC line again. Font L, "Lucca.636" is an italic cursive font so your chosen text will be printed italicised:

Enclose *any* text you like.

3.3.2 Loading fonts

Any of the resident fonts may be changed by loading a new font in its place. Make sure the Protype disc is in drive A (any side up). Now from BASIC command mode type |FONT,"L","SKYBOLD.646" RETURN (if you are using a CPC464 see the note ‡ in §A1). The font should be loaded from disc into memory. When the cursor reappears type |INFO RETURN to confirm that the italic font "Lucca.636" has been replaced by the bold sanserif font.

The starter fonts included with Protype are:

KlassiK	.626	Medium upright serif
Lucca	*.636*	*Medium italic cursive*
Mikron	.427	Small medium upright serif
skyLite	.606	Light upright sanserif
skyKite	**.626**	**Medium upright sanserif**
skyMite	.407	Small light upright sanserif
skyboLd	**.646**	**Bold upright sanserif**

3.4 Toggled features

Protype recognises three special embedded printer codes which are used for superscripts (raised text), subscripts (lowered text) and underlining. These three codes are X, Y and Z respectively. In each case you use the same code to turn the feature on and off. This is called toggling so superscripting, subscripting and underlining are known as toggled features.

3.4.1 Underlining

Returning to your BASIC line replace each of the codes T and K (on either side of your italicised text) with the code Z (&FA):

```
10 PRINT #8,"Enclose ";CHR$(&FA);"any";CHR$(&FA);" text you like."
```

When the line is printed the relevant text will be underlined:

Enclose any text you like.

If you are printing in adjust mode (see §3.6) it sometimes advisable to link the words of underlined phrases using non-break spaces (code &91) to avoid small breaks in the underlining.

3.4.2 Superscripts and subscripts

Any section of text may be raised or lowered with respect to the baseline using the printer control codes X or Y. In practice, however, the larger fonts will tend to lose part of their ascenders (b,d,f,h,k,l and upper case letters) or descenders (g,j,p,q,y) off the top or bottom of the line. As a consequence, superscripts and subscripts are best printed using one of the smaller fonts (Mikron.427 or skyMite.407). If you have loaded new fonts check that resident font M contains one of these two files. When using superscripts and subscripts remember to select fonts as well as toggling the raising or lowering of the text eg.:

```
10 PRINT #8,"H";CHR$(&ED);CHR$(&F9);"2";CHR$(&F9);CHR$(&EB);"O"
```

This will print as: H_2O

These are, of course, very inelegant and clumsy examples intended only to demonstrate the effect of the printer control codes.

3.5 Further Protype RSX commands

You have already met several of Protype's commands !INFO, !DRAFT, !NLQ and !FONT. Protype itself is switched in and out using the command !PROTYPE[,0]. The advantage of this type of command is that it can be used from many very different programs. RSX commands are the key to the versatility of Protype. This section intruduces the remaining commands.

!ADJUST
: Switches the formatting mode of Protype, stretching spaces to adjust the position of the right hand end of lines. This mode is used in conjunction with right-justification (see §3.6).

!FREE
: Switches the formatting mode of Protype so that lines occupy the minimum width consistent with aesthetic spacing of the text (see §3.6).

!TABLE
: Switches the formatting mode of Protype to facilitate tabulation (the vertical alignment of columns of text or figures (see §3.6)).

!GAP, *n*
: Sets the microscopic spacing between characters; the standard spacing is selected by !GAP,0. The decrease or increase in spacing is given by *n*/120 inches where *n* is a number in the range -2 to +2. In BASIC negative values may be included directly in the command eg. !GAP,-1. The command !GAP,1 is sometimes convenient if text is to be photoreduced.

!LFEED, *n*
: Sets the linefeed spacing (the vertical distance occupied by a line of text). The line spacing is set to *n*/216 inches for 9-pin printers (*n*/180 inches for 24-pins) where *n* is a number in the range -72 to +72. The default value is 39 (13 point). In BASIC negative values (i.e. backfeeds) may be included directly in the command eg. !LFEED,-39. (NB. some printers will not allow backfeeds if friction drive is being used.)

!CPI, *n*
: Sets the nominal pitch in characters per inch for justification and tabulation purposes. Since fonts are proportionally spaced the figure is a notional average value which serves to calculate the absolute horizontal position of columns or margins but has no meaning with respect to the actual number of characters in any given inch. Permitted values for *n* are 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 15, 17 and 20. Most of these values are included to permit future development. The main values of any practical use with the fonts supplied are !CPI,12, !CPI,15 and !CPI,17.

!CPCM, *n*

As for !CPI, *n* except that values are given in characters per centimetre. The default value of *n* is 6 which is suitable for all the fonts supplied (6 characters per cm is a little over 15 characters per inch). !CPCM 7 may be used if text only contains the fonts Mikron.427 and skyMite.407.

!MIRROR [,0]

Issuing the command !MIRROR causes all subsequent text to be flipped so the beginning of the line is at the right hand side of the printer carriage and the text image is reflected right to left. This will be of use in future developments where, for instance, an Arabic or Hebrew script output is required. When mirror imaging is selected this is indicated at the centre top of the status table. This feature is cancelled using !MIRROR,0.

3.6 Formats

3.6.1 Formatting modes

Protype's three formatting modes are Adjust, Free, and Table and are selected using the RSX command of the same name. Adjust mode does not in itself right-justify text; what it does is fix the length of each line to be proportional to the number of characters in the line. The character codes counted are all those in the range &01 to &DF with the exception of the line feed (&0A), form feed (&0C) and carriage return (&0D) codes. Thus if the number of characters sent in each line is variable the printout will have a 'ragged' right margin even if Adjust mode is selected. This paragraph is printed in this way.

If the number of characters in each line is the same and Protype is in Adjust mode the text will be right-justified even though the Protype output is proportionally spaced. Because it can work with both text formats Adjust mode is the Protype default mode. 'Soft spaces' (code &90) may be used to pad out a line to give a fixed number of characters. The position of these soft spaces in the line is unimportant. Protype doesn't print the soft spaces it merely counts them and then stretches each of the hard spaces to compensate producing a well-balanced appearance. To calculate the printed width of a block of right-justified text (in centimetres or inches), divide the number of characters in a line by the current Protype pitch (in characters/cm or characters/inch). The default pitch is 6 characters per centimetre.

If you wish to print mainly ragged right text then Free mode may be more appropriate. This paragraph is printed using Free mode. The advantage is that the spaces between words are equal throughout giving the body of the text a more uniform appearance. Additionally if the number of characters in consecutive lines happens to coincide by chance whilst using Adjust mode, this section of the text will appear justified and therefore slightly odd. In Free mode, the right margin is usually slightly more irregular than when ragged right text is printed in Adjust mode. For occasional text such as headings Free mode can simulated by using non-break spaces (code &91) between words.

Table mode is included to enable information to be easily presented in a tabular form. An absolute horizontal position is calculated for any character preceded by a space. This position is based on the number of previous characters in the line regardless of their widths. So any words preceded by the same number of characters in consecutive print lines will be vertically aligned in the printout even though their letters are proportially spaced. Any tabulated phrases should be held together with non-break spaces (code &91). If this is not done a separate horizontal position will be calculated for each word producing uneven results. Table mode is therefore unsuitable for bulk text.

3.6.2 Line graphics characters

Ruled tables may be produced in any formatting mode by using a set of line graphics characters whose horizontal positions are always calculated. These characters (codes &93 to &9F) are shown in §A4.

One special line graphics character is the tab character (code &94). This character is visible on the screen but invisible in the printout. If placed before a character it fixes the horizontal position of that character enabling unruled tables to be produced whilst in Adjust or Free modes.

3.6.3 Line feeds and form feeds

Each line sent should be terminated by the code sequence &0D, &0A (carriage return, line feed) before sending the next line. The PRINT #8 instruction does this automatically if a final semi-colon (;) is not present.

If you use continuous stationery the page length may not be exactly divisible by your chosen line spacing so that an integer number of lines per page cannot be calculated. Or you may regularly change line spacings or use backfeeds so that the exact vertical position of the next page is difficult to keep track of. In both these cases the easiest solution is to send a form feed code so that your printer moves automatically. to the start of the next page. The form feed code is &0C and should be sent immediately after a carriage return in place of a line feed.

3.7 **Diacritics and modifiers**

You have already seen that Protype selects fonts using the printer control codes K (&EB), T (&EC) and M (&ED) for font selection (§3.3) and the codes X (&F8), Y (&F9) and Z (&FA) to toggle features common to all fonts (§3.4). The remaining printer control codes are in theory free for each font to interpret as suits it. In practice though, similar fonts follow the same conventions.

The conventions for Latin fonts are as follows:

a b c d f g h I J n o q t u	are diacritics wc → ŵ
e p r s	are modifiers ss → ß
v w	may vary from font to font

3.7.1 Diacritics

A diacritic is a mark (for example an accent) added to a character to show a change in meaning or phonetic value. The printer control code A (&E1) is used to represent an acute accent so that sending the sequence aa i.e. ["a"+CHR$(&E1)] to stream #8 will result in á being printed. The full list of diacritic printer codes is given in §A3.

3.7.2 Modifiers

There are many more characters and symbols available in print than the standard ASCII characters. These extra symbols are generally represented in a stream of text by an ASCII character or pair of ASCII characters followed by one of the modifier control codes E (&E5), P (&f0), R (&F2) or S (&F3). The digraph Æ, for example is represented by the sequence AES (&41, &45, &F3). Some of the more language specific character/diacritic combinations such as Latvian ģ, Hungarian ő, Maltese ħ, or Romanian ţ are also represented using modifiers rather than the more general diacritics. Occasionally a character may combine with the previous character; the two code sequence -> (&2D, &3E) prints as →. A full list of these combinations is given in §A3. You will see from the table at §A4 that many special characters may also be printed using single codes in the ranges (&80 to &8F) and (&A0 to &CF). When initialised Protype cancels the default firmware printer translation table so that codes &A0 to &AF may be sent untranslated.

3.8 **Configurations**

It may be that you wish to customise Protype for your own special application. You can configure Protype yourself either before or after Protype is initialised by chaining short BASIC programs. The examples in this section assume that your configuration programs are on separate disc from your working copy of Protype but with a little modification they may be included on the same disc.

3.8.1 Configuring before Protype is initialised

This is the best option if you want to change the default fonts. Configuring beforehand avoids loading the default fonts and then immediately substituting them. Here is an example program:

```
10 fontK$="skyKite.626": fontM$="skyMite.407"
20 PRINT "Insert Protype disc and press any key"
30 i$=INKEY$: IF i$="" THEN 30 'wait for keypress
40 CHAIN "DISC"
```

If the program is on a separate disc from Protype then you can call it "DISC". If the program is on the same disc as Protype you don't need lines 20 or 30. If you also intend to configure after Protype is initialised (§3.8.2) the last line should be 40 CHAIN "PROTYPE".

3.8.2 Configuring after Protype is initialised

In order to do this you need to initialise Protype with the command RUN "PROTYPE" rather than RUN "DISC". The program will then stop at some point and ask for a disc containing CONFIG.BAS to be inserted. Here is an example of what CONFIG.BAS might look like:

```
10 INK 0,0: INK 1,26        'set colours to maximum contrast
20 KEY 0,"|INFO"+chr$(&0D) 'expansion string to key f0
30 |FREE                    'switch to Free format
40 END                      'to BASIC command mode
```

The program CONFIG.BAS already present on side B of the Protype disc simply switches to screen mode 2 and deletes itself.

3.9 Tips

If you wish to print a diacritic on its own this is best done by sending to stream #8 a non-break space (code &91) followed by the diacritic's printer control code.

If you need to place diacritics both above and below a lower case character (eg. a háček and a jot) send the printer control code for the top diacritic first. Otherwise Protype won't recognise the character as lower case and will place the háček too high. o[hj] prints as ǒ. o[jh] prints as o̬.

If you need to print two consecutive characters that are normally interpreted as one you can separate them with an unused control code. <- prints as ←. <[v]- prints as <-.

4. Using Protype from other programs

Programs producing non-graphics printer output (word processors, spreadsheets, databases etc.) should work directly with Protype if their output is pure ASCII; those which use printer control codes will usually allow these to be redefined. If you need to redefine these codes read §3 to learn about Protype's printer codes. (Machine coders - Programs using Protype must print using the jumpblock entry MC PRINT CHAR at &BD2B.)

Protype should be installed before the main program is run using the instructions in §3.1. It makes extensive use of the second 64K bank of CPC6128 memory and cannot, therefore, be used with programs which use this memory. This excludes use with CP/M Plus programs.

Since Protype is relocatable it is not strictly necessary to reset the computer before installing Protype provided certain conditions are valid:

Prior to installation, HIMEM must be &8200 or above.

Since Protype collapses the user-defined matrix table this must either be non-existent or else at HIMEM (i.e. if HIMEM has been lowered the table must have been collapsed previously using SYMBOL AFTER 256).

It may be possible to configure some programs with keyboard text input so that they accept sequential keypresses for accented characters etc. in the same way as for the Protext configuration (§2.7, §2.8 and §A2). To this RUN"K" on side B of the Protype disc. Bear in mind, though, that many programs have their own ideas as to the interpretation of keypress codes &80 to &FC and will not always produce a screen character. (Machine coders - program input must be via a call to the jumpblock KM WAIT CHAR at &BB06.)

NB. The program which sets up the new key and matrix definitions 'locks up' the symbol table so that it cannot be expanded or collapsed using, for example, SYMBOL AFTER 256. The BASIC instruction MEMORY HIMEM+1 used directly after the program "K" will 'unlock' the symbol table so it may be subsequently extended or collapsed.

If for some reason your main program cannot be installed alongside Protype, it is still possible to print any ASCII files produced by the program using a short BASIC program together with Protype. ASCPRINT.BAS on side B of the Protype disc is an elementary example of such a program.

4.1 Protype and Utopia

Your CPC may be fitted with Arnor's Utopia utilities ROM. When the Protype key configurations for Protext are set up there is not enough room in the expansion buffer for the new expansion strings as well as Utopia's strings. The Utopia keypress CONTROLf0 (INK 0,13: INK 1,0: BORDER 10) is disabled. Keypresses CONTROLf1-f9 are unchanged.

The Protype command !INFO has the same name as a Utopia external command. The Utopia command is still available using !U,"INFO".

4.2 Protype and Promerge Plus

When used with Protext, both Protype and Arnor's Promerge mail merge program work by intercepting Protext vectors. This, combined with the fact that Promerge Plus uses background printing (and the second bank of 64K), makes the two programs largely incompatible with each other. When Protype is installed with the Promerge Plus ROM initialised, it therefore disables all the additional stored commands, edit mode commands and command mode commands provided by Promerge Plus. However, the Extended Command Entry features of Promerge Plus (copy cursor editing, last command recall etc.) are still available from Protext.

You may wish to use the extended Promerge Plus text editing facilities, (box mode, two file editing etc.) together with Protype's key and matrix definitions but without installing Protype itself or any fonts. To do this, reset the computer by pressing CONTROL, SHIFT and ESC simultaneously, insert the Protype disc in drive A with side A uppermost and type RUN "K" RETURN. The keys and matrices will be configured and Protext entered.

Any pure ASCII files created by the Promerge Plus command PF may be printed using Protype together with the BASIC program ASCPRINT.BAS which is on side B of the Protype disc.

A1. **Protype commands (summary)**

These commands are given using their BASIC syntax:

!PROTYPE [,0]	Switches Protype in [or out].
!INFO	Displays Protype status table.
!DRAFT	Selects draft quality printing.
!NLQ	Selects near letter quality printing.
!FONT,"*font*","*fontname*"	Loads a font from disc. (*font* is one of K,L,M)‡
!ADJUST	Selects Adjust formatting mode.
!FREE	Selects Free formatting mode.
!TABLE	Selects tabulation (Table mode)
!GAP, *n*	Adjust inter-character spacing by *n*/120 inch. (-2 < *n* < +2) [Protext 254→255,0→2]
!LFEED, *n*	Set line feed in *n*/216 inch (9-pin printers). *n*/180 inch (24-pin printers) (-72 < *n* < +72) [Protext 184→255,0→72]
!CPI, *n*	Set nominal pitch to *n* characters per inch. (*n* is one of 1,2,3,4,5,6,7,8,10,11,12,15,17,20)
!CPCM, *n*	Set nominal pitch to *n* characters per centimetre. (*n* is one of 1,2,3,4,5,6,7,8,10,11,12,15,17,20)
!MIRROR [,0]	Switches mirror imaging on [or off].

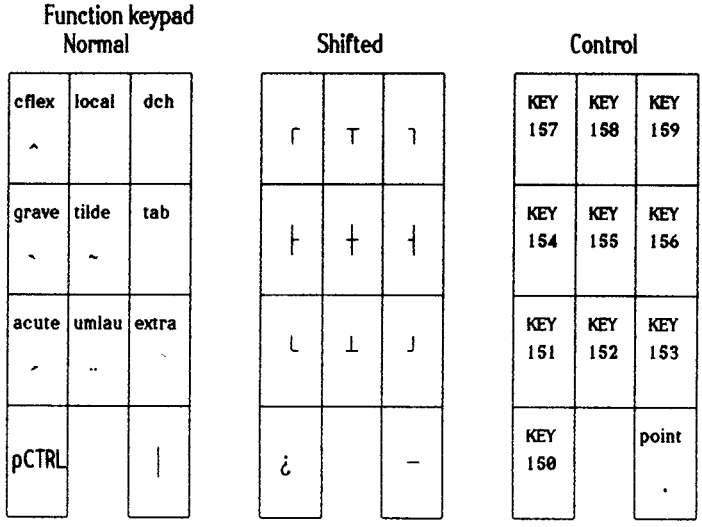
‡ Note for CPC464 users. The BASIC syntax:

```
!FONT,"K","SKYKITE.626"
```

is not possible on the CPC464. You need to use something like:

```
font$="K": fontname$="SKYKITE.626": !FONT,@font$,@fontname$
```

A2. **Keypress configurations**



Main keyboard CONTROL1 → ¡ , CONTROL4 → ° , CONTROL7 → ‘

Sequencial keypresses (keypad press first)

f1=´	f4=`	f7=^	f2=¨	f5=~	f8=local
´+a → á	`+a → à	^+a → â	¨+a → ä	~+a → ã	f8+a → å
´+e → é	`+e → è	^+e → ê	¨+e → ë	~+n → ñ	f8+A → Å
´+i → í	`+i → ì	^+i → î	¨+i → ï	~+N → Ñ	f8+c → ç
´+o → ó	`+o → ò	^+o → ô	¨+o → ö		f8+C → Ç
´+u → ú	`+u → ù	^+u → û	¨+u → ü		f8+o → ø
					f8+O → Ø

f3=extra

f3+a → ª	f3+/ → ÷	f8+s → ß	
f3+c → ©	f3+2 → ½	f8+! → ¡	
f3+f → ç	f3+3 → ♯	f8+? → ¿	
f3+o → º	f3+4 → ♭	f8+, → «	
f3+m → ♂	f3+: → ●	f8+. → »	
f3+p → ¶	f3+; → ±	f8+: → †	
f3+s → §	f3+@ → ○	f8+@ → ‡	

All the above characters are available on-screen simultaneously

A3. Diacritics and modifiers (Latin fonts)

(ā to ž represent printer control codes &E1 to &FA)

Diacritics (in the examples below 'x' represents any character)

xā →	x́ acute	xī →	ẋ iot (overdot)
xō →	x̄ bar (macron)	xū →	x̣ jot (underdot)
xē →	x̂ circumflex	xē →	ẍ strikeout
xā →	ẍ diaeresis (umlaut)	xō →	x̊ ring
xī →	x̧ cedilla (misto)	xā →	x̨ queue (ogonek)
xā →	x̀ grave	xī →	x̃ tilde
xē →	x̌ háček	xū →	x̆ breve

Here is a list of the lower case characters and diacritics required for a selection of European languages:

Albanian	ç	ë														
Anglo-Saxon	þ	ð	æ	œ	ā	ē	ī	ō	ū	ǣ	œ̄					
Czech	á	č	ď	é	ě	í	ň	ó	ř	š	ť	ú	ů	ý	ž	
Danish	æ	ø	å	ö												
Estonian	ë	ö	õ	ü												
Finnish	å	ä	ö													
French	œ	à	â	ç	é	è	ê	ë	î	ï	ô	ù	û	ü		
German	ß	ä	ö	ü												
Hungarian	á	é	í	ó	ö	ő	ú	ü	ű							
Icelandic	þ	ð	æ	œ	á	é	í	ó	ö	ú	ý					
Irish Gaelic	á	é	í	ó	ú											
Italian	à	è	ì	ò	ù											
Latvian	ā	č	ē	ģ	ī	ķ	ļ	ņ	š	ū	ž					
Lithuanian	ą	č	ę	ė	į	š	ų	ū	ž							
Norwegian	æ	ø	å													
Polish	ą	ć	ę	ł	ń	ó	ś	ź	ż							
Portuguese	á	à	â	ã	ç	é	è	ê	í	ì	ó	ò	ô	õ	ú	ù
Romanian	á	â	ă	è	ì	î	ș	ț	ù							
Scots Gaelic	à	é	è	í	ó	ò	ú									
Serbo-Croat	ć	č	đ	š	ž											
Slovak	ä	á	č	ď	é	í	ľ	ň	ó	ô	ŕ	š	ť	ú	ý	ž
Slovene	č	š	ž													
Spanish	á	é	í	ñ	ó	ú	ü									
Swedish	å	ä	ö													
Turkish	á	â	ç	ğ	ı	î	ö	ş (ş)		ü	û					
Welsh	á	â	ê	ë	î	ï	ô	ö	û	ŵ	ŷ					

Modifiers (many are also available as single codes in the range &A0 to &CF)

<s	→	«	left guillemet	k,	→	ķ	Latvian [1]
>s	→	»	right guillemet	l,	→	ļ	Latvian [1]
{s	→	“	left double quote	n,	→	ņ	Latvian [1]
}s	→	”	right double quote	s,	→	ș	Romanian [1]
hs	→	ħ	Maltese [1]	t,	→	ț	Romanian [1]
Hs	→	Ħ	Maltese [1]	K,	→	Ķ	Latvian [1]
ds	→	ď	Czech [1]	L,	→	Ļ	Latvian [1]
ls	→	ľ	Slovak [1]	N,	→	Ņ	Latvian [1]
ts	→	ť	Czech [1]	S,	→	Ș	Romanian [1]
is	→	ı	dotless i	T,	→	Ț	Romanian [1]
Ls	→	£	pound sterling	0/	→	Ø	slashed zero
's	→	‘	left single quote	1/	→	ⅼ	narrow one
2s	→	½	half	a/	→	ɑ	round ɑ
+s	→	±	plus or minus	cr	→	©	copyright [1]
/s	→	÷	division sign	pr	→	¶	paragraph [1]
xs	→	×	times	gr	→	g	alternative g [1]
cs	→	¢	cents	dr	→	°	degree
.s	→	·	raised point	o/	→	ø	l. case Scand. ø
ss	→	ß	scharfes ss	O/	→	Ø	u. case Scand. Ø
#s	→	♯	sharp [1]	''	→	“	left double quote
os	→	º	Spanish m. ordinal 1º	,,	→	”	right double quote
as	→	ª	Spanish f, ordinal 2ª	,,	→	„	double comma
fs	→	♀	Venus (fem.) [1]	!!	→	↓	down arrow
ms	→	♂	Mars (masc.) [1]				
:s	→	†	dagger (deceased) [1]	de	→	ð	l. case eth
;s	→	‡	double dagger [1]	te	→	þ	l. case thorn
*s	→	●	black blob (bullet) [1]	De	→	Ð	u. case ETH [1]
@s	→	○	white blob [1]	Te	→	Þ	u. case THORN [1]
aes	→	æ	l. case ash	se	→	§	section
AEs	→	Æ	u. case ASH	?e	→	¿	inverted ?
oes	→	œ	l. case oe digraph	!e	→	¡	inverted !
OEs	→	Œ	u. case OE digraph	ge	→	ģ	Latvian ģ [1]
ks	→	¤	krona [1]	be	→	♭	flat (bemol) [1]
x]s	→	⊠	box [2]				
x)s	→	⊗	encircle [2]	up	→	ű	Hungarian [1]
				op	→	ő	Hungarian [1]
<-	→	←	left arrow	dp	→	đ	Serbo-Croat [1]
->	→	→	right arrow	-p	→	ł	Polish [1]
				Lp	→	Ł	Polish [1]
zr	→	ȝ	alternative z [3]				

Notes: [1] not italics, [2] small fonts only, [3] italics only

A4. Latin alphabet code conventions (table)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
&0-											LF		FF	CR		
&1-																
&2-	spa	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
&3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
&4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
&5-	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	↑	_
&6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
&7-	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	↓
&8-	←	→	«	»	ð	þ		Þ		ı						
&9-	ssp	nbs	dch	└	tab	┘	┌	┤		┴	─	┴	┐	┤	┬	┼
&A-	Ø	ł	ɑ	£	©	¶	§	‘	g	½		±	÷	×	¿	¡
&B-	ø	¢	°	å	·	ß	♯	♭	º	ª	♀	♂	†	‡	●	○
&C-	Ø	Ñ	Æ	Å	Œ	æ	œ	¤								
&D-																
&E-	rst	x́ a	x̄ b	x̂ c	ẍ d	e	x̧ f	x̀ g	x̌ h	ẋ i	x̣ j	ftK k	ftL l	ftM m	x̃ n	x̊ o
&F-	p	x̲ q	r	s	x̃ t	x̆ u	v	w	x^x x	x_x y	x̲ z					

LF line feed
FF form feed
CR carriage return

spa space

ssp soft space
nbs non-break space
dch dummy character
tab force tabulation
rst reset

s printer control s
ft font
x^x superscript
x_x subscript
x̲ underline

A5. Technical notes

The key ENTER has the same effect as the key RETURN.

A5.1 Memory use

Protype takes up very little of the first 64K of memory; the main program is in block 4 and the three resident fonts sit in blocks 5, 6 and 7. When Protype is installed it lowers HIMEM by varying amounts depending on the configuration used.

Protext-configured Protype (side A) lowers HIMEM by about &7a0 bytes.

Unconfigured Protype (side B) lowers HIMEM by about &150 bytes.

A5.2 Quality of print

The high quality appearance of text is achieved using micro-justification of spaces, kerning (altering character spacing to suit each character pair combination) and optical line-end compensation (vertically aligning characters according to where they appear to start or end rather than where they actually do).

Protype will work on all Epson compatible dot-matrix printers whether 9 or 24-pin but in particular it has been written to push low cost 9-pin printers to their limits. It is not necessary to fit an 8-bit printer port to your computer; Protype works using 7 bits only. Many print enhancers claim to produce high quality print on 9-pin printers with a second pass of the print head. In practice, a second pass just isn't enough. The vertical step resolution of a typical 9-pin printer is 1/216 inch. Since the pin separation is 1/72 inch it is evident that three passes are needed to exploit the vertical stepper motor to its maximum.

On 24-pin printers the vertical step resolution is usually 1/180 inch; so printout is 20% taller. However it is often possible to set the line feed unit to 1/216 inch by setting a DIP switch - sometimes setting IBM mode will work, or there may be an extra setting such as 'Alternate Graphics Method'. If this is possible the printed results may be preferable.

Horizontally, 9-pin printers generally manage 120 contiguous dots per inch. However, 240 dot positions are available; the print head just doesn't have the speed of response to print them contiguously. A second horizontal pass allows the alternate dots to be printed. Multiply this by two again to enable well-proportioned ascenders and descenders and room for diacritics and you get a figure of twelve passes per line of print. Higher quality is technically impossible. The dot resolution is now 1/216 inch vertically and 1/240 inch horizontally. A typical laser printer has a resolution of 1/300 inch in both directions.

A5.3 This manual

The text for this manual has been edited and typeset using the Amstrad CPC6128 computer, Arnor's Protext ROM, the Protype program and the Star LC-10 printer. The printout has generally been photoreduced to 80%.

A5.4 Hexadecimal notation

Rather than use commonplace decimal (base 10) numbers, computer programmers often prefer to use a counting system which corresponds more closely to the binary system of the computer. This system uses hexadecimal (base 16) numbers. To represent the sixteen different digits, the ten symbols 0 to 9 are borrowed from the decimal system and the letters A to F are used to represent the remaining six. The character codes and printer control codes in this manual are often represented in hexadecimal form. This makes it easier to look codes up in the table §A4 (for example). To show that they are not decimal numbers, hexadecimal (or hex) numbers are preceded by the symbol &. The table below will help you convert from one system to the other. By combining a hexadecimal digit in the left column with another in the top row you can see that hexadecimal &C6 is decimal 198 and that decimal 93 is hexadecimal &5D.

HEXADECIMAL ⟷ DECIMAL conversion table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
&0-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
&1-	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
&2-	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
&3-	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
&4-	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
&5-	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
&6-	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
&7-	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
&8-	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
&9-	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
&A-	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
&B-	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
&C-	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
&D-	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
&E-	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
&F-	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

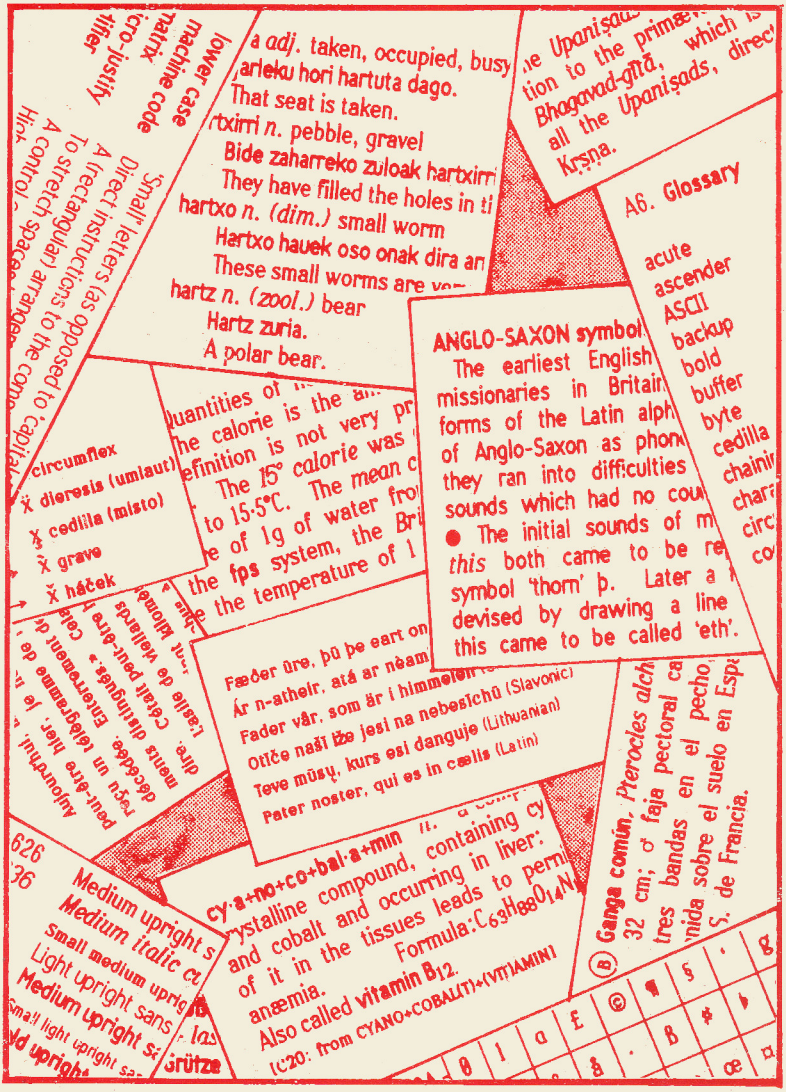
Looking at the table of codes (§A4.) you can see that the letter A is represented by hexadecimal &41 which is the same as decimal 65.

A6. Glossary

acute	an accent which rises from left to right (´)
ascender	Part of a character which is above the height of a lower case 'x'
ASCII	American Standard Code for Information Interchange
BASIC	Beginners All-purpose Symbolic Instruction Code
backup	A copy made to guard against losing information
bold	A heavy thick-lined typeface
breve	The diacritic (˘) often used for short vowels
buffer	The printer's temporary store for the data it receives
byte	The amount of memory used by one character or control code
cedilla	A diacritic (¸) often placed below **c** to change it's sound
chaining	The technique of running two or more BASIC programs in sequence
character	A letter, number, punctation mark, or symbol
circumflex	A diacritic (ˆ) eg. used over vowels in French and Welsh.
code	A number which represents a character or instruction
command	An instruction to do something
control code	A number which represents an instruction (eg. to the printer)
configure	To tailor a program to suit your needs
cursive	Having flowing character strokes
cursor	A marker on the screen to show where you are in the text
default	Refers to a number or a font used if no other is specified
decimal	A number written using base ten
descender	Part of a character which is below the base line
diacritic	A mark placed above, below or through a character
diaeresis	The diacritic (¨) (see Umlaut)
digraph	Two characters joined together (with a new meaning or sound)
DISCKIT3	A program to format or copy discs
draft	Printing 'in rough'
embedded	Included within the text
font	A set of characters with the same general design
format	The way text is laid out
form feed code	A control code sent to start a new sheet of continuous paper
friction drive	A paper positioning system usually used with non-perforated paper
grave	An accent which falls from left to right (`)
guillemets	French quotation marks « »
háček	A diacritic (ˇ) used in many East European languages
hexadecimal	A number (prèceded by '&' or '#') written using base sixteen
inverse video	The swapping of the text colour and background colour
italic	A right-sloping character or font
kerning	Fitting characters together according to their shapes
Latin script	The writing system used by most West European languages
ligature	Two characters joined for aesthetic reasons (see digraph)

linefeed	The vertical space occupied by a line of print
load	To transfer information from disc to the computer memory
lower case	'Small' letters (as opposed to 'capitals')
machine code	Direct instructions to the computer's microprocessor
macron	The diacritic (¯) often used for long vowels
matrix	A (rectangular) arrangement (eg. of dots to make a shape)
micro-justify	To stretch spaces by equal amounts to achieve a given line length
modifier	A control code which changes the previous character
NLQ	High quality printing
ogonek	The diacritic (˛) used with Polish and Lithuanian vowels
pitch	The (nominal) number of characters per unit of width
photoreduce	To make smaller by photocopying
point	A unit of typographic measurement (1/72 inch in Pica system)
port	A computer connection for other equipment (eg. a printer)
prompt	A noise or symbol (often '>') asking for some input
Promerge	A mail merge program produced by Arnor
Protext	A word processor produced by Arnor
Protype	Arnor's print enhancing, typesetting RSX
ragged-right	Having an irregular right margin
reset	To restore something to a standard state
resident font	A font in the computer memory
resolution	(for printers) The number of dots per inch
right-justify	To vertically align the right margin of a text
ROM	Read-Only Memory
Roman	1.(see Latin): 2.(see upright)
ruled table	A table divided by lines
RSX	Resident System eXtension: a program providing extra commands
sanserif	A typeface (font) without serifs
save	To transfer information from the computer memory to disc
serif	(A font having) a small line at the end of character strokes
soft space	A space character used to right justify text on the screen
status table	A table with information about current fonts and settings
symbol table	An area of memory where the screen character shapes are stored
tabulation	The arranging of text in columns
tilde	A diacritic (˜) (used eg. in Spanish and Portuguese)
toggle	To switch a function both on and off with the same operation
tractor drive	A paper positioning system using the holes of perforated paper
Umlaut	A diaeresis used in German **ä ö ü** to change the vowel quality
Utopia	A utilities ROM produced by Arnor
upper case	'Capital' letters
upright	Having vertical strokes (ie. not italic)

A7. Index


a adj. taken, occupied, busy
arleku hori hartuta dago.
That seat is taken.
rtxirri n. pebble, gravel
Bide zaharreko zuloak hartxirri
They have filled the holes in th
hartxo n. (dim.) small worm
Hartxo hauek oso onak dira an
These small worms are ve
hartz n. (zool.) bear
Hartz zuria.
A polar bear.
A6. Glossary
acute
ascender
ASCII
backup
bold
buffer
byte
cedilla
ANGLO-SAXON symbol
The earliest English
missionaries in Britain
forms of the Latin alph
of Anglo-Saxon as phon
they ran into difficulties
sounds which had no cou
● The initial sounds of m
this both came to be re
symbol 'thorn' þ. Later a
devised by drawing a line
this came to be called 'eth'.
circumflex
diæresis (umlaut)
cedilla (místo)
grave
hâček
Fæder ûre, þû þe eart on
Ár n-athair, atá ar nèam
Fader vår, som är i himmelen (Slavonic)
Otče naši tže jesi na nebesichû (Lithuanian)
Tevo mûsų, kurs esi danguje (Latin)
Pater noster, qui es in cælis
cy·a+no+co+bal·a+min
rystalline compound, containing cy
and cobalt and occurring in liver:
of it in the tissues leads to perni
anæmia.
Formula: C63H88O14N
Also called vitamin B12.
(C20: from CYANO+COBALT)+(VIT)AMIN)
Ganga común. Pterocles alch
32 cm; ♂ faja pectoral ca
tres bandas en el pecho
nida sobre el suelo en Espa
S. de Francia.
626
36
Medium upright s
Medium italic ca
Small medium upri
Light upright sans
Medium upright s
Small light upright sa
Bold upright
Grütze