MARTEL Instruments

PRINTER PROGRAMMING MANUAL

CONTENTS

This manual summarises the control codes and escape sequences used by the Martel range of thermal and impact printers. Where a control code only applies to one family of printer, it is highlighted in the description of that control code.

The MCP9800 family or serial head thermal printers comprises the MCP9800, MCP9801 and MCP9830. The MCP9830 has a resolution of 144 dots per line, compared to the MCP9800/MCP9801's 166 dots per line, so the ESC K graphics sequence needs to be handled differently.

The MCP6701 and MCP6702 desktop impact printers have similar functionality to the MCP9800, in addition to the MPP5400 and MPP5370 series Panel mount printers.

The MCP9810 family of fixed head thermal printers comprises the MCP9810, MCP9811, MCP9814, MCP9815, MCP9818, MCP9820, MCP9821 and MCP9840.

The MCP6810, MCP6820 and MCP6880 desktop printers have similar functionality to the MCP9810, in addition to the MOP1200, MPP5310 and MPP5410.

Control codes and escape sequences are specified as characters, hexadecimal and decimal values.

This manual summaries the functionality provided by the latest version of firmware for each model of printer. Printers with earlier versions of firmware may not support all of the features described.

HT

09_{16}

9_{10}

This control code moves the print position to the next horizontal tab position (set by **ESC D**).

If the tab takes the print position beyond the end of the current line, the line will be printed automatically on receipt of the next character.

The default tab setting is 8.

Care must be taken when mixing single and double width printing, as tab stops are based on numbers of characters regardless of character width.

LINE FEED

LF

0A$_{16}$

10$_{10}$

This control code causes any data in the line buffer to be printed and printing advances to the next line. If the line buffer is empty, printing advances to the next line regardless.

FORM FEED

FF

0C$_{16}$

12$_{10}$

This control code prints the current line and advances printing to the top of the next page (set by **ESC C**).

The default page length is 5.

CARRIAGE RETURN

CR

0D$_{16}$

13$_{10}$

This control code prints the current line and advances to the next line only if the line buffer contains data. This code is ignored if the line buffer is empty.

If data is printed by **CR**, then a **LF** immediately following the **CR** will be ignored (to avoid double line spacing).

Single spaced printing is achieved by placing **CR LF** or **LF** at the end of each line. Using **CR** only will result in blank lines not being printed.

Double line spacing is achieved by sending an additional **LF** or **CR LF** at the end of each line.

DOUBLE SPACING ON

SO

0E$_{16}$

14$_{10}$

This control code enables double width printing for the current line only. Double width is disabled at the end of the current line (or on reception of **SI** or **ESC W 0**)

ESC W should be used to permanently enable double width printing.

DOUBLE SPACING OFF

SI

0F$_{16}$

15$_{10}$

This control code cancels double width printing and is equivalent to **ESC W 0**.

CANCEL CURRENT LINE

CAN

18$_{16}$

24$_{10}$

This control code cancels the current print line. The next character taken from the serial buffer becomes the first character on the line.

DEL

7F$_{16}$

127$_{10}$

This control code deletes the last character on the current line.

This code cannot delete data already printed, only data stored in the line buffer.

Care must be taken when mixing double and single width characters, as **DEL** will delete the current character width only and could result in multiple or partial character deletions.

ESC ! *n*

1B 21 *n*$_{16}$

27 33 *n*$_{10}$

This escape sequence sets the print mode according to the value of *n*. *n* is defined as follows:

Bit	Function	Value 0	1
0	} Character font	see below	
1			
2	} Print density	see below	
3			
4	Double height	Disabled	Enabled
5	Double width	Disabled	Enabled
6	Undefined		
7	Underline	Disabled	Enabled

Character Font	Bit 1	Bit 0
24 characters per line	0	0
48 characters per line	0	1
32 characters per line	1	0
Undefined	1	1

Print Density	Bit 3	Bit 2
Light 1 (Default)	0	0
2	0	1
3 (Label Default)	1	0
Dark 4	1	1

MCP6880 ONLY

Character Font	Bit 1	Bit 0
36 characters per line	0	0
72 characters per line	0	1
48 characters per line	1	0
Undefined	1	1

Double and single height cannot be mixed on the same line. Changing Bit 4 part way through a line will cause that line to be printed before the change in height occurs.

Undefined bits should be set to zero to ensure compatibility with future software upgrades.

Example

To set an MCP9810 printer to 48cpl, double height, normal print density, normal width, no underlining, send:

ESC ! 00010001$_2$

1B 21 11$_{16}$

27 33 17$_{10}$

ESC * *m n1 n2 [d]*

1B 2A *m n1 n2 [d]*$_{16}$

27 42 *m n1 n2 [d]*$_{10}$

This escape sequence specifies and prints bit image graphics. *m* specifies the graphics format, *n1* and *n2* specify the graphics width and *[d]* specifies the dot pattern.

m = 0, 1, 32, 33$_{10}$
n1, n2 = graphics width *n1* + (256 x *n2*)
   [d] = graphics data
      n1 + (256 x n2*) bytes when m* = 0, 1
      (*n1* + (256 x *n2*)) x 3 bytes when *m* = 32, 33

If the bit image width exceeds one line, that line will be printed automatically and the graphics will continue on the next line.

If *m* is out of range, bytes *n1* onwards will be treated as normal data.

If the graphics width is not a multiple of eight dots, white space padding will be added to pad the graphics to the next multiple of 4 dots. This ensures that different character widths will line up vertically if interspersed with graphics.

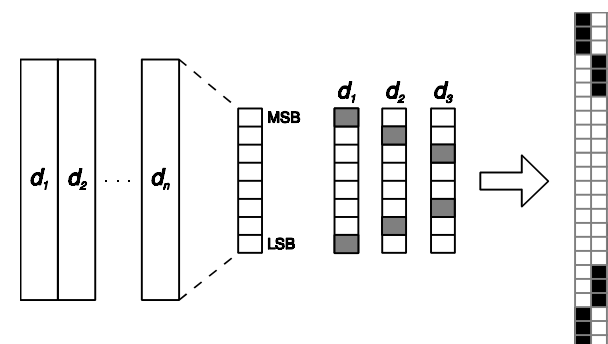| m_{16}	m_{10}	Mode	Vertical dots Per line	Horizontal dots Per line (max)	MCP6880 Horiz. dpl
00	0	8 dot single density	8	192	288
01	1	8 dot double density	8	384	576
20	32	24 dot single density	24	192	288
21	33	24 dot double density	24	384	576

Example

$m = 0$ 8 dot single density

ESC * 00 03 00 81 42 24$_{16}$
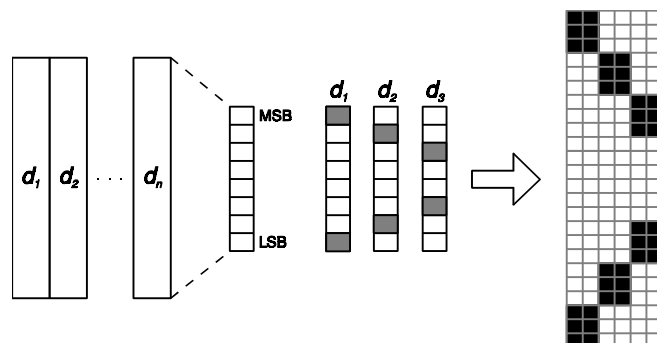
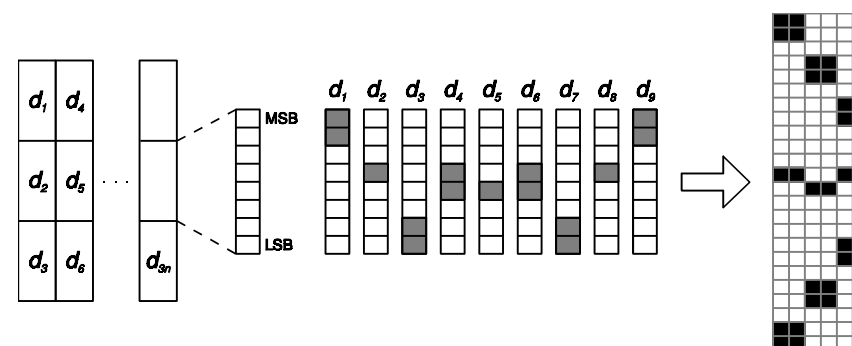$m = 1$ 8 dot double density

ESC * 01 03 00 81 42 24$_{16}$

$m = 32$ 24 dot single density
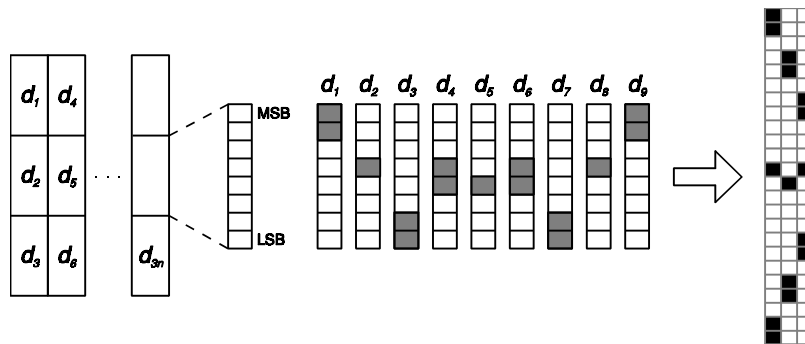
ESC * 20 03 00 C0 10 03 18 08 18 03 10 C0$_{16}$

m = 33   24 dot double density

ESC * 21 03 00 C0 10 03 18 08 18 03 10 C0$_{16}$

ESC – *n*

1B 2D *n*$_{16}$

27 45 *n*$_{10}$

This escape sequence enables or disables underlining.

$n = 00_{16}$ Underline off
$n = 01_{16}$ Underline on

Underlining is applied to all characters, including graphics, when enabled.

ESC @

1B 40$_{16}$

27 64$_{10}$

This escape sequence resets the printer to its power on state. All buffer data will be lost and all settings will be returned to their default. The DIP switches will also be read.

No data should be sent for one second after the **ESC @** to allow the printer to reset.

ESC 4

1B 34$_{16}$

27 52$_{10}$

This escape sequence enables italic mode printing. Characters are printed as italics, although a character printed in the last print column may not appear fully italicised if some of its pixels would exceed the maximum printing width of the mechanism.

Note that italic printing is only available on newer revisions of fixed-head printers. Please contact Martel Instruments Limited if you wish to use this feature.

DISABLE ITALIC CHARACTERS

ESC 5

1B 35$_{16}$

27 53$_{10}$

This escape sequence disables italic printing.

PRINT CHARACTER

ESC 6 *n*

1B 36$_{16}$ *n*

27 54$_{10}$ *n*

This escape sequence inserts one single character (with ASCII code *n*) into the print buffer. The sequence can be used to print characters whose ASCII codes are normally used for print formatting (such as ASCII 0A$_{16}$, which is normally interpreted as LF).

Note that this sequence is only available on newer revisions of fixed-head printers. Please contact Martel Instruments Limited if you wish to use this feature.

SET PAGE LENGTH

ESC C *n*

1B 43 n_{16}

27 67 n_{10}

This escape sequence sets the page length in lines.

$1 \leqslant n \leqslant 255_{10}$

Use **FF** to advance to the top of the next page.

The default page length is 5.

SET TAB WIDTH

ESC D *n*

1B 44 n_{16}

27 68 n_{10}

This escape sequence sets the width of the tab stops use by the printer.

Use **HT** to advance to the next tab stop.

The default tab width is 8.

Care must be taken when mixing single and double width printing, as tab stops are based on numbers of characters regardless of character width.

ENABLE BOLD PRINTING

ESC G

1B 47$_{16}$

27 71$_{10}$

This escape sequence enables bold printing.

The MCP9800 family cannot print bold and normal characters simultaneously. Using **ESC G** part way through a line will cause that line to print before bold is enabled.

The MCP9810 only supports bold printing when the 24 characters per line font is selected. The 32 and 48 characters per line fonts will be printed as normal if bold is enabled.

DISABLE BOLD PRINTING

ESC H

1B 48$_{16}$

27 72 n_{10}

This escape sequence disables bold printing.

The MCP9800 family cannot print bold and normal characters simultaneously. Using **ESC H** part way through a line will cause that line to print before bold is disabled.

ESC I *n*

1B 49$_{16}$ *n*

27 73$_{10}$ *n*

This sequence sets the maximum current of the printer. It can be used to reduce the peak current of externally-powered printers so that the power supply demands are reduced.
n must be between 0 (highest current, default) and 3 (lowest).
The print speed will be reduced if the current is reduced.

Note that this sequence is only available on newer revisions of fixed-head printers. Please contact Martel Instruments Limited if you wish to use this feature.

ESC K *n1 n2 [d]*

1B 4B *n1 n2 [d]*$_{16}$

27 75 *n1 n2 [d]*$_{10}$

This escape sequence specifies and prints bit image graphics and is equivalent to **ESC * 1** on the MCP9810 family.

n1,n2 = graphics width *n1 + (256 x n2)*
    [d] = graphics data *n1 + (256 x n2)* bytes

If the graphics width is not a multiple of 8 dots, white space padding will be added by the MCP9810 family to pad to the next multiple of 4 dots. No padding is performed by the MCP9800 family.

If the bit image exceeds one line, that line will be printed automatically and the graphics will continue on the next line.

Printer	Horizontal dots Per line (max)
MCP9800/MCP9801	166
MCP9810/MCP9811	384
MCP9820/MCP9821	384
MCP9830	144
MCP9840	384
MCP6810/MCP6820	384
MCP6880	576
MCP6701	144
MCP6702	240

Example

ESC K 03 00 81 42 24$_{16}$

MCP9810 family   MCP9800 family
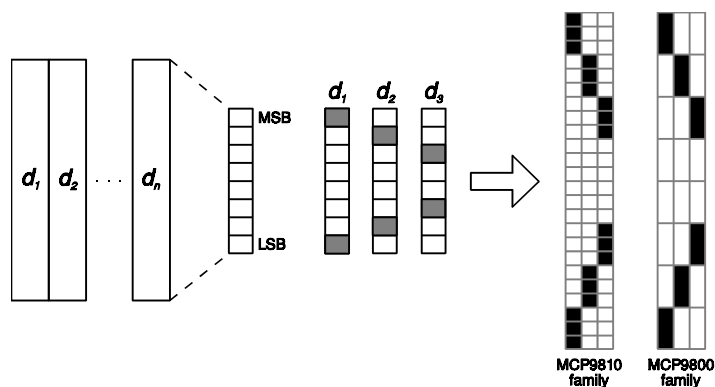
ESC L

1B 35$_{16}$

27 53$_{10}$

This escape sequence enables label mode. Following the receipt of this sequence the printer will use the paper-out sensor of the printer mechanism to align thermal labels (in conjunction with the ESC f sequence).

There is no escape sequence to cancel label mode.

SELECT INTERNATIONAL CHARACTER SET

ESC R *n*

1B 52 n_{16}

27 82 n_{10}

This escape sequence selects the international character set as specified by *n*.

$0 \leqslant n \leqslant 10_{10}$

The default character set is 0 (USA).

n	Hex Character set	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
0	U.S.A	#	$	@	[	\	]	^	`	{	|	}	~
1	France	#	$	à	°	ç	§	^	`	é	ù	è	"
2	Germany	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
3	U.K	£	$	@	[	\	]	^	`	{	|	}	~
4	Denmark I	#	$	@	Æ	Ø	Å	^	`	æ		å	~
5	Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
6	Italy	#	$	@	°	\	é	^	ù	à	ò	è	Ì
7	Spain I	Pt	$	@	¡	Ñ	¿	^	`	"	ñ	}	~
8	Japan	#	$	@	[	¥	]	^	`	{	|	}	~
9	Norway	#	¤	É	Æ	Ø	Å	Ü	è	æ		å	ü
10	Denmark II	#	$	É	Æ	Ø	Å	Ü	è	æ		å	ü

DOUBLE WIDTH ON/OFF

ESC W *n*

1B 57 n_{16}

27 87 n_{10}

This escape sequence enables or disables double width printing.

n = 00_{16} Double width off
n = 01_{16} Double width on

ESC Z *n1 [d1] n2 [d2] …. n24 [d24]*

1B 5A *n1 [d1] n2 [d2] …. n24 [d24]*$_{16}$

27 90 *n1 [d1] n2 [d2] …. n24 [d24]*$_{10}$

This escape sequence specifies and prints Run Length Encoded bit image graphics. The sequence allows high resolution graphics to be downloaded to the printer far quicker than using the conventional **ESC * 33** sequence.

The sequence comprises 24 lines of graphics bytes which are overlaid on the current text line. It is possible to combine text and graphics on a line provided the text information is sent first, then the **ESC Z** graphics and then the **LF** command to print the line.

  n1 = graphics width for line 1 (top)
[d1] = graphics data for line 1 *n1 bytes*

Each graphics byte can contain Run Length Encoded white or black pixels or non-encoded bit image pixels. The most significant bit of each byte denotes the graphics type.

d$_7$	d$_6$	d$_5$	d$_4$	d$_3$	d$_2$	d$_1$	d$_0$	
0	0	x	x	x	x	x	x	RLE white pixels (0 to 63)
0	1	x	x	x	x	x	x	RLE black pixels (0 to 63)
1	x	x	x	x	x	x	x	Seven bit image pixels (0=white, 1=black)

Example

ESC Z	01 40 01 40 01 40 01 43 01 43 01 43
	01 46 01 46 01 46 01 4C 01 4C 01 4C
	01 58 01 58 01 58 01 70 01 70 01 70
	02 70 70 02 70 70 02 70 70 03 7F 7F
	7F 03 7F 7F 7F 03 7F 7F 7F$_{16}$

Prints

PRINT AND FEED PAPER

ESC d *n*

1B 64 *n*$_{16}$

27 100 *n*$_{10}$

This escape sequence prints the current line and then feeds the paper by *n* lines.

LABEL ADVANCE

ESC f

1B 66$_{16}$

27 102$_{10}$

This escape sequence advances the print position to the top of the next label. This sequence is only active if label printing has been selected by the appropriate DIP switch.

ESC i *n*

1B 69 n_{16}

27 105 n_{10}

This escape sequence enables or disables reversed printing (white on black).

n = 00_{16} Reverse printing off
n = 01_{16} Reversed printing on

ESC t *n*…

1B 74_{16} *n*…

27 116_{10} *n*…

Where *n* can be…

p	Print the time/date in the current format
f *n1*	Set time/date printing format to *n1*
s hh:mm:ss,dd/oo/yy	Set time/date note oo = month

ESC t f Codes:

bit	= 0 Meaning	= 1 Meaning
0	No seconds shown	Seconds are shown
1	12 hour clock with "AM" or "PM"	24 hour clock
2	YY	YYYY
3	Year is printed	Year is not printed
4	Month is a number	Month is 3 letters (e.g. "JAN")
5	MM/DD	DD/MM
6	Time / Date Options… 00 = Time Date, 01 = Date Time, 10 = Time Only, 11 = Date Only	
7		

ESC t f Examples:

00_{16} 1:15PM 05/01/02   (note: no leading 0 for hour in AM/PM mode)
$D4_{16}$ MAY 01 2002   (note: days always show a leading zero)
67_{16} 01/05/2002 13:15:00

The date format will be retained between printer power-downs, provided that the battery has enough charge to retain the real-time clock hardware.

ESC t s Example:

ESC t s 13:15:00,01/05/02
(codes 1B 74 73 31 33 3A 31 35 3A 30 30 2C 30 31 2F 30 35 2F 30 32$_{16}$)

Sets the time to 1:15PM on 1st May 2002.

ESC t p Notes:

ESC t p   (codes 1B 74 70$_{16}$)

When this sequence is received the printer will print the time and date as specified by the last ESC t f *n* sequence.

The current print format is used (e.g. "double height" would be applied to the time/date printing if that mode were enabled).

Apart from the spaces which are part of the time/data print format, no other white-space is added (e.g. no leading or trailing CR/LF is added by the printer).

Note that this sequence is only available on request. Please contact Martel Instruments Limited if you wish to use this feature.

ESC w *n*

1B 77 *n*$_{16}$

27 119 *n*$_{10}$

This escape sequence enables or disables double height printing.

n = 00$_{16}$ Double height off
n = 01$_{16}$ Double height on

Double and single height printing cannot be mixed on the same line. If the height is changed part way through a line, the line will be printed before the change takes place.

ESC { *n*

1B 7B *n*$_{16}$

27 123 *n*$_{10}$

This escape sequence enables or disables inverse printing (upside down, right to left).

n = 00$_{16}$ Inverse printing off
n = 01$_{16}$ Inverse printing on

Normal and inverted printing cannot be mixed on the same line. If print orientation is changed part way through a line, the line will be printed before the change takes place.

GS h *n*

1D 68 *n*$_{16}$

29 104 *n*$_{10}$

This control sequence sets the vertical height of the barcode in dots.

The MPC9810 family has a vertical resolution of 8 dots/mm.

The default barcode height is 96$_{10}$ (12mm, equivalent to four text lines).

GS k *n [d]*

1D 6B *n [d]*$_{16}$

29 107 *n [d]*$_{10}$

This control sequence prints a barcode. The type of barcode is chosen by the value of *n*. See below for a full description of each barcode type.

n$_{16}$	Barcode
00	UPC – A
01	UPC –E
02	EAN 13
03	EAN 8
04	CODE 39
05	Interleaved 2 of 5
06	CODABAR
07	CODE 128

If *n* is out of range *[d]* will be treated as normal data.

Once all barcode data has been received, the barcode is printed automatically.

The user must select the appropriate barcode magnification (**GS w *n***) to maximise barcode readability, and allow the barcode to fit in the resolution of the print head. See following table.

Maximum barcode character counts						
	Magnification			MCP6880		
Type	2	3	4	2	3	4
UPC-A[1]	11	11	11	11	11	11
UPC-E[1]	6	6	6	6	6	6
EAN 13[1]	12	12	12	12	12	12
EAN 8[1]	7	7	7	7	7	7
CODE 39[2]	12	7	5	20	12	9
Interleaved 2 of 5[2]	26	16	12	40	26	18
CODABAR[3]	13(15)	9(10)	6(7)	20(23)	13(15)	10(11)
CODE 128 (Code A/B)[4]	14	8	5	23	14	9
CODE 128 (Code C)[5]	28	16	10	46	28	18

(1) Excludes check digit – number represents actual user digits.

(2) Excludes start & stop – number represents actual user characters.

(3) Includes mandatory start & stop characters provided by the user. The number in brackets is the maximum for barcodes containing '0' to '9', '-' and $ only.

(4) Represents maximum for barcodes using only Code A or Code B character sets. Barcodes that use both Code A and Code B simultaneously will have lower maximum character counts. Excludes start, stop & check digits – number represents actual user characters.

(5) Represents maximum for barcodes using even numbers of numeric digits only (Code C). Odd numbers of digits or barcodes that mix digits and characters will deliver lower maximum character counts. Excludes start, stop & check digits – number represents actual user digits.

PRINT A UPC-A BARCODE

GS k 0 *[d]*

1D 6B 00 *[d]*$_{16}$

29 107 0 *[d]*$_{10}$

This control sequence prints a UPC-A barcode. Eleven numeric digits must follow the **GS k 0** sequence. The barcode is printed automatically after reception of the eleventh digit. The check digit is calculated automatically.

[d] = 11 Bytes: '0' to '9'
$\qquad\qquad$ 30 to 39$_{16}$
$\qquad\qquad$ 48 to 57$_{10}$

PRINT A UPC-E BARCODE

GS k 1 *[d]*

1D 6B 01 *[d]*$_{16}$

29 107 1 *[d]*$_{10}$

This control sequence prints a UPC-E barcode. Six numeric digits must follow the **GS k 1** sequence. The barcode is printed automatically after reception of the sixth digit. The check digit is calculated automatically.

[d] = 6 Bytes: '0' to '9'
$\qquad\qquad$ 30 to 39$_{16}$
$\qquad\qquad$ 48 to 57$_{10}$

PRINT AN EAN13 BARCODE

GS k 2 *[d]*

1D 6B 02 *[d]*$_{16}$

29 107 2 *[d]*$_{10}$

This control sequence prints an EAN13 barcode. Twelve numeric digits must follow the **GS k 2** sequence. The barcode is printed automatically after reception of the twelfth digit. The check digit is calculated automatically.

[d] = 12 Bytes: '0' to '9'
30 to 39$_{16}$
48 to 57$_{10}$

PRINT AN EAN8 BARCODE

GS k 3 *[d]*

1D 6B 03 *[d]*$_{16}$

29 107 3 *[d]*$_{10}$

This control sequence prints an EAN8 barcode. Seven numeric digits must follow the **GS k 3** sequence. The barcode is printed automatically after reception of the seventh digit. The check digit is calculated automatically.

[d] = 7 Bytes: '0' to '9'
30 to 39$_{16}$
48 to 57$_{10}$

PRINT A CODE39 BARCODE

GS k 4 *[d] (terminator)*

1D 6B 04 *[d] (terminator)*$_{16}$

29 107 4 *[d] (terminator)*$_{10}$

This control sequence prints a CODE39 barcode. The number of characters is variable, but must be within the maximum for the current magnification.

[d] = '0' to '9', 'A' to 'Z', '-', '.', ' ', '$', '/', '+', '%'
    30 to 39, 41 to 5A, 2D, 2E, 20, 24, 2F, 2B, 25$_{16}$
    48 to 57, 65 to 90, 45, 46, 32, 36, 47, 43, 37$_{10}$

The terminator character must be provided before the barcode will print. Any non-valid character will cause the barcode to terminate, so any of the following characters may be used:

(terminator) = '*', CR, LF or NUL
        2A, 0D, 0A or 00$_{16}$
         42, 13, 10 or 0$_{10}$

PRINT AN INTERLEAVED 2 OF 5 BARCODE

GS k 5 *[d] (terminator)*

1D 6B 05 *[d] (terminator)*$_{16}$

29 107 5 *[d] (terminator)*$_{10}$

This control sequence prints an Interleaved 2 of 5 barcode. The number of digits is variable, but must be within the maximum for the current magnification.

[d] = '0' to '9'
    30 to 39$_{16}$
    48 to 57$_{10}$

If an odd number of digits is provided, a leading zero will be added automatically.

The terminator character must be provided before the barcode will print. Any non-digit character will cause the barcode to terminate, so any of the following characters may be used:

(terminator) = CR, LF or NUL
        2A, 0D or 0A$_{16}$
        42, 13 or 10$_{10}$

PRINT A CODABAR BARCODE

GS k 6 *(start) [d] (stop) (terminator)*

1D 6B 06 *(start) [d] (stop) (terminator)*$_{16}$

29 107 6 *(start) [d] (stop) (terminator)*$_{10}$

This control sequence prints a CODABAR barcode. The user must provide valid start and stop characters. The number of characters is variable, but must be within the maximum for the current magnification.

(start) / (stop) = 'A' to 'D'
        41 to 44$_{16}$
        65 to 68$_{10}$

[d] = '0' to '9', '-', '\$', ':', '/', '.', '+'
   30 to 39, 2D, 24, 3A, 2F, 2E, 2B$_{16}$
   48 to 57, 45, 36, 58, 47, 46, 43$_{10}$

The terminator character must be provided before the barcode will print. Any non-valid character will cause the barcode to terminate, so any of the following characters may be used:

(terminator) = CR, LF or NUL
        0D, 0A or 00$_{16}$
        13, 10 or 0$_{10}$

PRINT A CODE128 BARCODE

GS k 7 *n [d]*

1D 6B 07 *n [d]*$_{16}$

29 107 7 *n [d]*$_{10}$

This control sequence prints a CODE128 barcode. The number of characters in the barcode must be provided by the user as all possible terminator characters are valid characters that could be built into the barcode.

n = number of characters built into the barcode
[d] = *n* bytes:
        Code A: NUL to '-'
                00 to 5F$_{16}$
                0 to 95$_{10}$
        Code B: ' ' to DEL
                20 to 7F$_{16}$
                32 to 127$_{10}$
        Code C: '00' to '99' (Pairs of digits '0' to '9'
                30 to 39$_{16}$
                48 to 57$_{10}$)

The check digit is calculated automatically. Printing of the barcode commences on reception of the *n*th byte. The user must pick the appropriate magnification for a given barcode.

The implementation of this barcode is not complete at present, as the SHIFT and FUNC codes are not available currently. The printer automatically selects the most appropriate code set and handles changes between code sets.

GS w *n*

1D 77 *n*$_{16}$

29 119 *n*$_{10}$

This control sequence sets the barcode magnification. *n* represents the number of dots used to make up the thinnest element of a given barcode.

$2 \leqslant n \leqslant 4$

The user must select the largest possible value for a given application and barcode type.

The default magnification is 3.

ESC $ *n1 n2*

1B 24 *n1 n2*$_{16}$

27 36 *n1 n2*$_{10}$

This escape sequence sets the barcode start position in dots from the left margin. The barcode starts at dot:

$$\textbf{dot} = n1 + (256 \times n2), \text{ where } 0 \leqslant \textbf{dot} \leqslant 383.$$

$$(\text{MCP6880} \leqslant 575)$$

The default value for *n1* and *n2* is zero, positioning the barcode at the left margin.

This manual covers programming of the following printer models:

MCP9800	MCP8810	MCP6701
MCP9801	MCP8830	MCP6702
MCP9810	MCP8840	MCP6810
MCP9811	MCP8880	MCP6820
MCP9814		MCP6880
MCP9815		MOP1200
MCP9818		MPP5310
MCP9820		MPP5370
MCP9821		MPP5400
		MPP5410

MP/PPM/E