

THERMOCODE SERIES 2

ESCAPE SEQUENCES

Issue 3.1

Designed and manufactured by;

**OPEN DATE EQUIPMENT LIMITED
Units 8 & 9 Puma Trade Park,
145 Morden Road,
Mitcham
Surrey
CR4 4DG
ENGLAND**

TEL: - 0044 (0) 208 655-4999

FAX: - 0044 (0) 208 655 4990

Index

Description	Page
1. General operation of the printer	4
1.1 How to use the escape codes	4
1.2 Defining a label	4
1.2 Defining a label continued	5
1.3 Controlling the printer	6
1.4 Escape code listing	7
1.4.1 Escape code listing	8
1.4.2 Escape code listing	9
1.5 Communications	10
1.5.1 Serial Communications	10
1.5.2 USB Specification for Open Date Printers	10
1.5.3 Ethernet or TCP/IP Networking	10
1.6 Font management	11
1.6.1 Loading a font	11
1.6.2 Requesting printer font names	11
1.6.3 Requesting printer font information	11
1.6.4 Deleting all printer fonts	12
1.6.5 Deleting an individual font	12
Commands Used to Define a Format	
1.7 Commands used to define a format	12
1.7.1 Print format description	12
1.7.2 Print parameter variables	12
1.7.3 Standard parameter field names	13
1.7.4 Possible offset errors	13
1.7.5 Home position offset	13
1.8 Text lines	14
1.9 Line / Box fields	15
2.0 Barcode fields	15 - 16
2.0.1 Control codes for code 128	17
2.0.2 2D Code Datamatrix	17
2.0.3 2D Code Datamatrix continued	18
2.0.4 2D Code Datamatrix continued	19
2.0.5 2D Code PDF 417	20
2.0.6 2D Code QR	21
2.1 Graphics fields used in a format	22
2.1.1 Graphics fields loaded to a printer	22 - 23
2.1.2 Updating and overwriting a local graphic field	23
2.1.3 Request global graphic names	23
2.1.4 Request global graphic information	23
2.1.5 Delete all global graphic fields	24
2.1.6 Delete individual global graphic fields	24
2.1.7 Downloading Global Graphics to the printer	24
2.2 Variable Insertion fields	25
2.2.1 Variable Insertion text fields type "A"	25
2.2.2 Variable Insertion text fields type "0"	26
2.3 Updating existing variable fields	27
2.4 Requesting format variables	28
2.5 Global variable insertion text fields type "0"	28
2.5.1 Requesting global text variable names	28
2.6 Requesting global text variable information	29
2.7 Deleting global text fields & all global text fields	29
2.7.1 Delete	29
2.7.2 Delete individual global graphic fields	29
2.8 Variable insertion text fields type "1"	29
2.8.1 Variable insertion text fields type "4" counter fields	30

Index

Description	Page
2.8.2 Variable insertion text fields type "4" counter fields continued	31
2.8.3 Variable insertion text fields type "3" 24 hour time	31
2.8.4 Variable insertion text fields type "8" 12 hour time	31 - 32
2.8.5 Variable insertion text fields type "5" Day types	32
2.8.6 Variable insertion text fields type "6" Year types	33
2.8.7 Variable insertion text fields type "7) Month types	33 - 34
2.9 Shift Codes type "9"	34
2.9.1 Specifying rollover times of automatic date fields	34 - 35
2.9.2 Variable field type "W"	35 - 36
3.0 Select format from the file store	37
3.0.1 Deselect format from the image memory	37
3.0.2 Request format data	37
3.0.3 Delete all formats	38
3.0.4 Delete individual formats	38
3.0.5 Print quantity used within a format	38
3.0.6 Request print quantity	38
3.0.7 Request total prints	38
3.0.8 Request format name	38
3.0.9 Request all print parameters	39
3.1 Request individual print parameter	39
3.1.1 Specifying daylight saving dates	39
3.1.2 Request daylight saving	40
3.1.3 Special control variables used within automatic date codes	40
Commands to Control and Monitor the Printer	
3.2 Commands to control and monitor the printer	41
3.2.1 Auto print command	41
3.2.2 Quantity command	41
3.2.3 Change mechanical variables	41
3.2.4 Request system variables	41
3.2.5 Request individual system variables	41
3.2.6 Delete system variable changes	42
3.2.7 Delete individual system variable changes	42
3.2.8 Pause printing	42
3.2.9 Resume printing	42
3.3 Request printer status	42
3.3.1 Printer output to communications lines	42 - 43
3.3.2 Request memory	43
3.3.3 Request errors	43
3.3.4 Clear errors	43
3.3.5 Request a range of variables & parameters	43 - 44
3.3.6 Initialise part of the printer memory	44
3.3.7 Initialise all of the printer memory	44
3.3.8 Request all files	44
3.3.9 Request individual files information	44
3.4 Delete a file from the file store	45
3.4.1 Selecting a new burn file	45
3.4.2 Program the printer	45
3.4.3 Loading a file to the printer	45 - 46
3.4.4 Text line output to communication port	46
3.4.5 The text output line in defined in a format	46 - 47
3.4.6 Request printer type & software version	47
3.4.7 Request mechanical variable	47
3.4.8 Request individual mechanical variable	47
4.0 Passwords	48
4.1 Setting passwords continued	49
4.2 Requesting individual passwords	49

1. General Operation of The Printer

The Thermocode 2 printer can be programmed / controlled by a computer or other communicating device via a communications link.

All the commands used start with an escape character followed by command instructions and this manual describes how, by the use of escape codes to control and manage the printer.

The manual starts by describing the general use and layout of the escape codes and then gives details of all the available codes and control variables.

1.1 How to Use Escape Codes

There are two things that we can do with escape codes with the printer:

- Define the image for the label to be printed
- Control the operation of the printer

1.2 Defining a Label

The label to be printed is defined in a PRINT FORMAT. This consists of a number of lines of escape codes between a start line, containing the name of the format and an end line. This named format is saved within the file store of the printer and can be selected for printing either with an escape code or from the hand held terminal on the printer.

The escape code lines of a format will contain a range of information: -

- Print parameters that control the burn level, the speed, the head height etc. (Esc0P)
- Local text variable definitions that create text field, counters, dates and time (Esc0E)
- Local system variables to give precise printer control specific to a format (Esc0X)
- Local pieces of graphic stored within the format (Esc0V)
- Line/Box drawing commands (Esc0L)
- Text line for fonts (Esc0T)
- Barcode line definitions (Esc0B)
- Graphic file use lines (Esc0W)
- Split memory plane controls (Esc0E...W)
- Label count control (Esc0Q)
- Text output control lines (Esc0Y)

A typical format would look like this: -

Esc0F Example1		Format Name
Esc0P SPEED 0100	[]	Print Parameters
Esc0P BURN1 0270		
Esc0P BURN2 0270		
Esc0P PRESSURE 20		
Esc0P HOMOFF 03.0		
Esc0P ROTATE 0		
Esc0E Lot 0QWERTY	[]	Text Variable Fields
Esc0E BB 012/07/11		
Esc0E Counter0 40001,1,1,1,9999		Counter Variable
Esc0T Arial Bold 0372006512001Lot No: \0Lot0	[]	Actual Print text and Counter Field
Esc0T Arial Bold 0372012112001Best Before: \0BB\0		
Esc0T Arial Bold 0740006512001\0Counter0\0		
Esc0B 01039401380163040110507865443706		EAN13 Barcode
Esc0Q 000000		Print Qty field
Esc0K		Format end
Esc0S Example1		Enables the label to go on line ready for printing

The printed format:

Lot No: QWERTY0001
Best Before: 12/07/11



5 078654 437065 >

The number of formats stored within the printer is limited by the size of the formats and the available space within the printer. This could be as high as 150 depending on format size.

The use of text variables allows lines of text and barcode in a format to have any required level of complexity. Each variable defines a small bit of text, such as the current day number or a counter with 3 digits or a shift code. Once defined, these can be used in the format any number of times and combined with fixed bits of code.

Here is an example: -

```

Esc0Eday1    51
Esc0Emonth1  71
Esc0Eyear1   63
  
```

The above defines: -

- A variable day1 of type day with 2 numeric digits
- A variable month1 of type month with 2 digits
- A variable year1 of type year with 4 digits

These can then be built into a text line: -

```
Esc0TArial Bold0100010012001\0day1\0:\0month1\0:\0year1\0
```

Which will print the following: -

Date=16:07:2010

In Arial bold font, at position 100*100.

In the same label we could also have the following text lines: -

```

xxxxyyyy
Esc0TArial Bold0100020012001The current month=\0month1\0
Esc0TArial Bold0100030012001The current year=\0year1\0
  
```

Which will print: -

The current month = 01

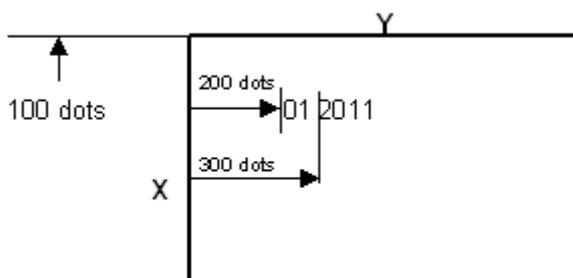
At 100 by 200

And

The current year = 2011

At 100 by 300

Most of the text variables automatically update when printing and give the format designer complete control over the text and barcodes on a label.



1.3 Controlling The Printer

Many of the escape codes help to control and monitor the operation of the printer.

Typical escape codes: -

- Initialise the printer
- Load fonts to the printer memory
- Load global graphics to the printer memory
- Stop and start printing
- Request one or many of the system settings
- Request information from the file store
- Return one or more system settings to their default value
- Delete one or more files from the file store

These are all one line commands that can be sent at any time, for example:-

```

Esc0CINew
Esc0AArial      0034578
Esc0GVGRAPH2   02000200000000
Esc0GECOUNTER0 40000
Esc0GP

```

The above defines the following functions: -

- Full initialisation of the printer
- Load the font Arial into the file store
- Load the named graphic GRAPH2 to the file store
- Create the global variable named Counter0 of type counter with 4 digits
- Do an immediate print

In addition to these direct commands there are a large number of system and mechanical variables within the printer that control every aspect of printer operation. Each of these variables are named and all have default values that allow the printer to be used in normal conditions but which can be modified to improve system performance or fit the printer to a particular application. For example:-

```

Esc0XSYSHEIGHT 2.0
Esc0XPACCVERT 4000
Esc0XSYSIPADDR 192.168.1.21

```

The above controls the following: -

- The head height is set to 2.0mm
- The vertical acceleration is defined set to 4000mm/s²
- The IP address is set to 192.168.1.21

Some functions require a number of system variables to control and you need to set all the relevant variables to achieve correct operation.

Use the escape code listings to help find the command you need. Look in the special sections that explain particular aspects of the printer operation. Look at the lists of system and mechanical variables for one that helps your application (see separate manuals for full system & mechanical listings).

1.4.0 Escape Code Listing

Esc Code	Qualifier	Command Global	Description	Section
A	None	Yes	Loading fonts to printer file store	1.6.0
B	None	No	Loading barcodes within a format	2.0
C	C	Yes	Pause printing	3.2.0
C	R	Yes	Resume printing	3.2.1
C	INEW	Yes	Deletes fonts, variables, graphics, and formats from file store and memory, restores all default settings	3.3.7
C	INIT	Yes	Deletes fonts, variables, graphics, and formats from file store and memory	3.3.6
C	IProg	Yes	Resets printer and allows new code to be downloaded	3.3.7
D	A	Yes	Delete all fonts from file store	1.6.4
D	A (font name)	Yes	Delete specific font from file store	1.6.5
D	E	Yes	Deletes all global variables from file store & memory	2.7
D	E (global name)	Yes	Deletes specific global variable from file store & memory	2.7
D	F	Yes	Deletes all formats from file store & memory	3.0.3
D	F (format name)	Yes	Deletes specific format from file store & memory	3.0.4
D	H (File name)	Yes	Deletes file from memory	3.4
D	V	Yes	Deletes all global graphics from file store & memory	2.1.5
D	V (graphic name)	Yes	Deletes specific global graphic from files tore & memory	2.1.5
D	X	Yes	Restores all system variables back to default	3.1.8
D	X (variable name)	Yes	Restores a specific system variable to default	3.1.9
D	Z	Yes	Clear all errors	3.2.6
E	none	No	Variable insertion fields	2.8
F	none	No	Format name (file name)	1.2
G	E	Yes	Global variable text fields loaded to file store	2.2
G	P	Yes	Automatic immediate print command	3.1.4
G	Q	Yes	Remote quantity command	3.0.5
G	V	Yes	Global graphic file loaded to file store	2.1.1
H	none	Yes	Loading files to printer	3.4.3

1.4.1 Escape Code Listing

Esc Code	Qualifier	Command Global	Description	<u>Section</u>
I	none	Yes	Overwriting of existing local variable insertion fields	2.3
K	none	No	Identifies end of format file	3.0.2
L	none	No	Line & box field	1.9.0
O	none	No	Password identification number (future use)	
P	none	No	Print parameters	1.7.1
Q	none	No	Quantity of prints command within a format	3.0.5
S	none	Yes	Deselect current format	3.0.1
S	S(name)	Yes	Select format from file store and load to printer memory	3.0.0
T	None	No	Text fields	1.8.0
U	V	Yes	Updates a local graphic	2.1.2
U	E	Yes	Updates a local variable insertion field	2.3
V	None	No	Local graphics fields loaded within a format	2.1.1
W	None	No	Graphic used in a format, including co-ordinates etc.	2.1.0
X	None	Yes/No	System variables	3.2.4
Y	None	No	Text line output to communications port	3.4.4
Z	None	Yes	Request printer type & software version	3.4.0
Z	A	Yes	Request font names from the file store	1.6.2
	A (font name)	Yes	Requests font file from file store	1.6.3
Z	E	Yes	Requests global text field names from the file store	2.6
Z	E (global name)	Yes	Requests global text field file from the file store	2.6
Z	F	Yes	Requests format names from file store	3.0.2
Z	F (format name)	Yes	Request format file from the file store	3.0.2
Z	H	Yes	Requests all files in memory, including identification	3.3.8
Z	H (file name & Identification)	Yes	Request actual file from memory	3.3.9
Z	I	Yes	Request all variables in selected format	2.4
Z	M	Yes	Request available memory of file store	3.3.2
Z	N	Yes	Request name of format being printed	3.0.8
Z	P	Yes	Requests all parameters of the format being printed	3.0.9

1.4.2 Escape Code Listing

Esc Code	Qualifier	Command Global	Description	Section
Z	P (name)	Yes	Requests individual print parameter	3.1.0
Z	Q	Yes	Request quantity that has been set and amount of prints that have been done (current image only)	3.0.6
Z	R	No	Reserved future use	
Z	S	Yes	Requests status line of printer	3.2.9
Z	T	Yes	Request total prints (cycles in life of printer)	3.0.7
Z	V	Yes	Request global graphic names from file store	2.1.3
Z	V (graphic name)	Yes	Requests global graphic file from the file store	2.1.4
Z	X	Yes	Request system variables settings from the memory	3.2.4
Z	X (system name)	Yes	Requests a specific system variable value	3.2.4
Z	Y	Yes	Requests mechanical variables settings from the memory	3.4.7
Z	Y (mechanical name)	Yes	Requests a mechanical system variable value	3.4.6
Z	Z	Yes	Requests all errors	3.3.2
Z	8	Yes	Reserved for future use	
Z	9	Yes	Requests a range of variables and parameters	3.3.4

General Notes

1. All lines prefixed by an escape code (character 27) and a zero (character 48).
2. All lines terminated by carriage-return (character 13) and a line feed (character 10).
3. Null lines allowed (carriage return and line feed with no other data).
4. The maximum number of characters on each line within a format is 150.
5. Names of formats can have up to 15 alphanumerical characters (no punctuation characters allowed) followed by a minimum of one space character.
6. The names of fonts, graphics & variable insertion fields, can have up to 10 alphanumerical characters (no punctuation characters allowed) followed by a minimum of one space character.
7. Within a dedicated name there must be no space characters. The only valid characters used within a name: -

Alphabetical (upper or lower case) A to Z
Numerical, 0 to 9
Underscore, _

1.5 Communications

1.5.1 Serial Communications

There is a selection 7-baud rates (Bit per second) ranging from 4800 to 230400. The default value is 115200.

Bits per second	115200 (Baud Rate)
Data bits	8
Parity	None
Stop Bits	1
Flow Control	Hardware

When using a USB to serial converter then the above settings apply and the serial lead supplied with the printer must be used (part number LEA 755021).

1.5.2 USB Specification for Open Date Printers

The USB system international standard has had a number of revisions; 1.0, 1.1 and 2.0.

At each revision, the maximum speed has been increased, but full compatibility has been maintained with the previous versions.

The current version is 2.0 and it defines 3 possible transfer speeds (the speeds are defined as bits per second):

Low speed	= 1.5Mb/sec
Full speed	= 12Mb/sec
High speed	= 480Mb/sec

The Open Date printers meet the USB 2.0 specification for full speed operation at 12Mb/sec (1200000 bits per second).

We do not support operation at 480Mb/sec, because we cannot manage the high data rate in the printer.

The way the specification works, our printers will plug into any USB enabled computer, regardless of the USB version that the computer runs (1.0, 1.1 or 2.0) as long as the correct driver is used.

Specifically, they will plug into a computer with a USB 2.0 specification port, and will run at a maximum 12Mb/sec (depends on computer, windows version and printer operations).

1.5.3 ETHERNET or TCP/IP Networking for Open Date Printers

Printers that include a USB & Ethernet card need the following information, either pre-set or programmed to the printer: -

Port Number	=	9100 (Fixed in printer)
MAC Address	=	Pre-programmed on board by manufacturer.
IPADDRESS	=	Customers own identification (separate number for each printer)
IPGATEWAY	=	Customers own identification (not required though listed)
IPMASK	=	Customers own identification (should match the Network)

The system has been designed to work at **10 Megabit / second specification**, but will connect to most 10/100 hubs to work with equipment running at 100 Megabit / second.

1.6 Font Management

1.6.1 Loading a Font

The Printer can store and utilise either True Type (*.ttf) or Bit Map (*.sfp) fonts, for the printing of text characters, automatic dates and counter fields.

The size of font is important. Large fonts will use too much memory and create errors such as font file problem, get character fail etc. If two fonts are being loaded to the printer the maximum size of 105Kb each. The smaller the font allows the user to load more than two fonts.

The format for loading a font into the printer is as follows: -

Esc 0A {name} {ss} {FFFFFF} {CRLF} Esc {font}

{name} = The name of the font, this field must be ten characters long with printable ASCII characters followed by trailing spaces

{ss} = 00 for scalable fonts; point size for bit mapped fonts

{FFFFFF} = Size in bytes of the font file (000000) always six digits

{CRLF} = Carriage Return, Line Feed

Esc = Escape

{font} = Font file contents {FFFFFF} bytes

1.6.2 Request Printer Font Names

Esc0ZA

Printer returns: -

{nnnnnnnnn1}

{nnnnnnnnn2}

{nnnnnnnnn3}

Etc

OK

Where: - nnnnnnnnn1 = First name of font in printer as specified

Where: - nnnnnnnnn2 = Second name of font in printer as specified

Names of fonts will always be padded to 10 characters long, spaces after name. The next font will automatically follow on the next line. After the last font name OK will indicate end of list.

1.6.3 Request Printer Font Information

Esc0ZA {name}

Where: -

{name} = Actual name of font in printer, this name must be correct and is case sensitive

Printer returns: -

Esc0A {name} {FFFFFF} Esc {data} OK

Where: -

{name} = Actual name of font in printer

{FFFFFF} = File size in bytes (six numerical characters)

{data} = Actual file contents of the font

Note: this exactly the same format in which the data was sent.

1.6.4 Delete All Printer Fonts

Esc0DA

Printer returns: - OK

This command deletes all fonts from the printer file store. The current print format will be de-selected, and the printer will be "Off Line".

1.6.5 Delete Individual Font

Esc0DA {name}

Where: -
{name} = Actual name of font in printer, this name must be correct and is case sensitive

Printer returns: - OK

This command deletes a specific font from the printer file store. If the name is not recognised the printer returns OK and the error "Delete file unknown" will be displayed on the terminal.

1.7 Commands Used to Define a Format

1.7.1 Print Format Description (Used Within a Format)

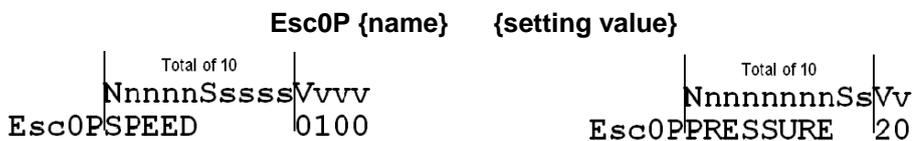
The format for sending a new print format is as follows: -

Esc0F {name}
{name} = The name of the format: this field can be up to 15 characters long, with printable ASCII characters, followed by trailing spaces (no spaces or punctuation are allowed within name only the underscore)

1.7.2 Print Parameter Variables (Used Within a Format)

The print parameters variables must be entered before any other data in a format. The parameters may be sent in any order.

These sequences will set the values of the print control parameters of the format. The variable types must be named and entered correctly (see list below).



Nnnnn = Name Ss = spaces dependent on name width Vvvv = Value dependent on field type

Where: -
P = Escape code for print parameter variables

{name} = Name of parameter to change. Max of

{setting value} = Actual value required

Note: Formats do not have to contain the print parameter variables, as the printer has default settings. You may however wish to include some of the parameter variables if required these will override the printer default settings. If you need to edit values using the hand held terminal, then the parameters should be included in the format because terminal editing is only allowed on parameters included in the format.

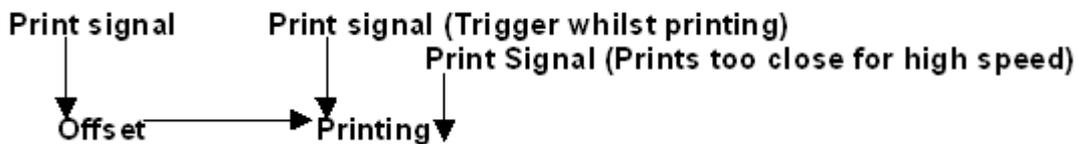
1.7.3 Standard Parameter Field Names

SPEED	=	Print speed in millimetres per second (0100 = 100 mm/sec)
BURN1	=	Burn value in microseconds, for intermittent A to B and continuous style printer (0250 = 250 microseconds)
BURN2	=	Burn value in microseconds, for intermittent B to A only (0250 = 250 microseconds)
PRESSURE	=	Pressure value in Newton's (20 = 20 Newton's or 2Kg)
OFFSET	=	Distance in millimetres of print position from actual print signal (050 = 50 millimetres) range 0 to 999 mm. Continuous models only
HOMOFF	=	Print position offset from home datum range 0 to 99 millimetres dependent on print area. Intermittent models only
ROTATE	=	Rotation of the format, either 0 = 0° or 2 = 180°

1.7.4 Possible Offsets Errors (OFFSET)

The user has to be careful with the **offset length** as this could give an error. Trigger whilst printing 123 and or "Prints too close for high speed" 122.

With the offset distance, time between prints and print signals is shorter.



1.7.5 Home Position Offset (HOMOFF)

This feature allows the user to specify within a format an offset in millimetres from the home sensor. This could save having to move the printer on the "Y" axis of a frame, enabling a new print position of the format to be achieved.

(See the standard printer default setting sheets for valid ranges of the parameter variables)

Example: print parameter variables within a format: -

```

Esc0FExapmle1
Esc0PSPEED      0100
Esc0PBURN1      0270
Esc0PBURN2      0270
Esc0PPRESSURE   20
Ecs0POFFSET     050 This line is entered in Continuous printers
Esc0PHOMOFF     03.0 This line is entered in Intermittant printers
Esc0PROTATE     0
Esc0TArial Bold 0740006512001Open Date Equipment
Esc0Q000000
Esc0K
Esc0SExample1

```

1.8 Text Lines

Maximum 120 text lines per format and a 100 characters maximum per text field. Also see file store variables text fields that can include large amounts of data (150 text fields per format).

Esc0T {ffffff} {xxxx} {yyyy} {ss} {o} {r} {P} (\1) {ppp} (\1) {a..a} (\0) {variable_name} (\0) {b...b}

Note: (\0) means binary all zeros or the NULL character.

- {ffffff} = The name of the font: this field must be ten characters long with printable ASCII characters followed by trailing spaces
- {xxxx} = Horizontal position, in dots
- {yyyy} = Vertical position, in dots
- {ss} = Scaling factor (point size nominal)
- {o} = Orientation (Clockwise rotation) 0 = 0°, 1 = 90°, 2 = 180°, 3 = 270°
- {r} = Reverse outline option (not implemented) set at 0
- {P} = Proportional flag
0 = fully proportional
1 = mono space variable (not implemented)
- (\1) = Binary Character 1 (start of percentage of font width)
- {ppp} = Percentage width of font, must be 3 digits (may be left out if not required)
- (\1) = Binary Character 1 (end of percentage of font width)
- {a...a} = Fixed ASCII text, prefix text may be left out if not required
- {variable_name} = Name of pre-defined variable, Must have a \0 before and after variable_name, If not required the Variable may be left out
- {b...b} = Fixed ASCII text, suffix text may be left out if not required

Note:

Fixed Text Only Fixed Text with a maximum of 120 lines
Esc0T**Arial Bold**0050005012000**100 characters maximum per text field**

Example: 1 Prefix Fixed Text
Prefix max 100
Esc0T**Arial Bold**0050005012000**5212QWERTY\0variable1\0**

Example 2: Suffix Fixed Text
Suffix max 100
Esc0T**ARIALBD** 0050005012000\0**variable1\05212QWERTY**

Example: 3 Prefix & Suffix Fixed Text Fixed Text
Prefix Max 100 Suffix Max 92
Esc0T**Arial Bold**0050005012000**5212QWERTY\0variable1\05212QWERTY**

Example: 4
Esc0T**ARIALBD** 0050005012000**5212QWERTY\0variable1\0\0variable2\05212QWERTY**

There can be fixed text before and after two or more variable named fields if required. When using multiple variable insertion fields within a format, you must still have a \0 before and after each variable_name. The variable text field has a maximum of 59 Characters but only 40 can be edited in the Mini Terminal.

1.9 Line / Box Field

No maximum limits on number of fields.

Esc0L {xxxx} {yyyy} {www} {hhh} {vvv} {rrr}

{xxxx}	=	Horizontal position, in dots (X axis)
{yyyy}	=	Vertical position, in dots (Y axis)
{www}	=	Horizontal width, in dots (X axis)
{hhh}	=	Vertical height, in dots (Y axis)
{vvv}	=	Width thickness, in dots (X axis)
{rrr}	=	Height thickness, in dots (Y axis)

Together, these parameters may describe vertical lines, horizontal lines, filled blocks or boxes, with sides of different thickness.

Lines or filled blocks are described by supplying width and height parameters as required and setting both thickness parameters (vvv) + (rrr) to all zero's. Boxes are described by supplying all parameters as required, where the thickness parameters set the thickness of the vertical and horizontal sides of the box respectively.

2.0 Barcode Field

Maximum 20 barcode fields per format.

Esc0B {ss} {xxxx} {yyyy} {hhh} {o} {b} {r} {h} {c} {s} {a..a} (\0) {variable_name} (\0) {b...b}

Note: (\0) means binary all zeros or the NULL character.

{ss}	=	Style	Symbology	Style	Symbology	
00	=		EAN 8	09	=	CODABAR
01	=		EAN 13	10	=	MSI
02	=		EAN 128	11	=	CODE 93
03	=		UPC-A	12	=	EXTENDED CODE 93
04	=		UPC-E	13	=	UCC - 128
05	=		CODE 39	14	=	HIBC
06	=		CODE 128	15	=	UPCE (6, SYSTEM 0)
07	=		ITF CODES	16	=	UPCE (6, SYSTEM 1) No longer supported
08	=		EXTENDED CODE 39	17	=	Not used
18	=		Not used	19	=	Not used
20	=		DATAMATRIX	21	=	PDF417
22	=		QR CODE			

{xxxx} = Horizontal position

{yyyy} = Vertical position

{hhh} = Height (not including human readable characters)

{o} = Orientation (clockwise rotation)

0 = 0°

1 = 90°

2 = 180°

3 = 270°

{b} = Narrow bar width in dots (between 1 to 9 dependent on the type of barcode)

{r} = Styles (element thickness & ratios)

0 = 3:1 ratio (Suitable for all types of barcode)

1 = 2.5:1 ratio (Dependent on the type of barcode)

2 = 2:1 ratio (Dependent on the type of barcode)

Continued on the next page

- {h} = Human readable option, (1 = ON, 0 = OFF)
- {c} = Checksum Digit Automatic, (1 = ON, 0 = OFF)
- {s} = Speed flag, 0 = normal, 1 = boundary move (future use)
- {a...a} = Fixed ASCII text, may be left out if not required
- {variable_name} = Name of pre-defined variable must have a \Ø before and after variable_name
- {b...b} = Fixed ASCII text, may be left out if not required

(See pages 16 to 20 for Datamatrix & PDF417)

Note: When using multiple variable insertion fields within a format, you must still have a \Ø before and after each variable_name.

Example 1:

```
Esc0B01005000200150040115212\Øvariable1\Ø
```

Fixed text (5212) before a variable named field(variable1)

Example 2:

```
Esc0B01005000200150040115212\Øvariable1\Ø\Øvariable2\Ø
```

Fixed text (5212) before a two variable named fields (variable1 & variable2)

Example 3:

```
Esc0B0100500020015004011\Øvariable1\Ø5212
```

Fixed text (5212) after a variable named field(variable1)

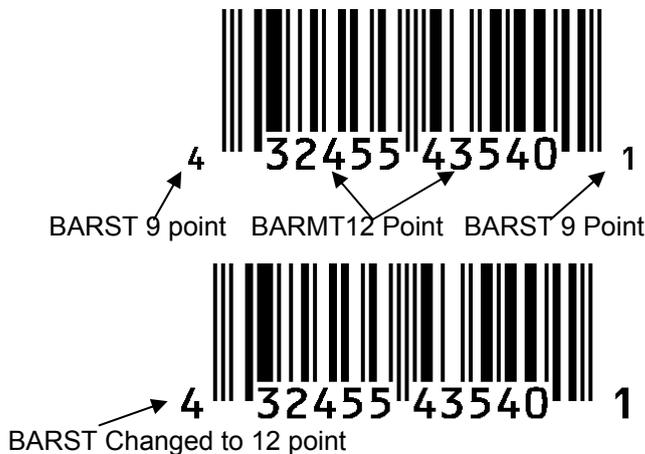
Example 4:

```
Esc0B0100500020015004011\Øvariable1\Ø5212\Øvariable2\Ø
```

Fixed text (5212) between two variable named fields (variable1 & variable2)

Note: For EAN type barcodes, the style parameter will be ignored.

Human readable font size can be changed with system variable XSYSBARMT and XSYSBARST are designed to be used with UPCA & UPCE barcodes. RBARMT and RBARST return the two font sizes used.



(For more information on Printer Variables see the manuals System Variable & the Mechanical Variable Fields)

2.0.1 Control Codes for Code 128

Code B and C are the most commonly used code sets. The structure is also smaller Code C is numeric only and has to be an even number of numeric characters.

CODE A

B06000700170120032100\CA01000506789012341712121410QWERTY123

CODE B

B06000700170120032100\CB01000506789012341712121410QWerTY123

CODE C

B06019800140120032100\CC010005067890123417121214101123

AUTO

B06016800190120032100\CC01000506789012341712121410\CBQWerTY123

2.0.2 2D Code Datamatrix

As the table below shows, rows and columns sizing 10 X 10 to 88 X 88 are supported

Symbol Size*		Data Region		Mapping		Total		Reed-Solomon		Inter-	Data Capacity			Error	Max. Correctable
				Matrix		Codewords		Block		leaved	Num.	Alphanum.	Byte	Correction	Codeword
Row	Col	Size	No.	Size	Data	Error	Data	Error	Blocks	Cap.	Cap.	Cap.	Overhead %	Error/Erasure	
10	10	8x8	1	8x8	3	5	3	5	1	6	3	1	62.5	2/0	
12	12	10x10	1	10x10	5	7	5	7	1	10	6	3	58.3	3/0	
14	14	12x12	1	12x12	8	10	8	10	1	16	10	6	55.6	5/7	
16	16	14x14	1	14x14	12	12	12	12	1	24	16	10	50	6/9	
18	18	16x16	1	16x16	18	14	18	14	1	36	25	16	43.8	7/11	
20	20	18x18	1	18x18	22	18	22	18	1	44	31	20	45	9/15	
22	22	20x20	1	20x20	30	20	30	20	1	60	43	28	40	10/17	
24	24	22x22	1	22x22	36	24	36	24	1	72	52	34	40	12/21	
26	26	24x24	1	24x24	44	28	44	28	1	88	64	42	38.9	14/25	
32	32	14x14	4	28x28	62	36	62	36	1	124	91	60	36.7	18/33	
36	36	16x16	4	32x32	86	42	86	42	1	172	127	84	32.8	21/39	
40	40	18x18	4	36x36	114	48	114	48	1	228	169	112	29.6	24/45	
44	44	20x20	4	40x40	144	56	144	56	1	288	214	142	28	28/53	
48	48	22x22	4	44x44	174	68	174	68	1	348	259	172	28.1	34/65	
52	52	24x24	4	48x48	204	84	204	84	2	408	304	202	29.2	42/78	
64	64	14x14	16	56x56	280	112	280	112	2	560	418	277	28.6	56/106	
72	72	16x16	16	64x64	368	144	368	144	4	736	550	365	28.1	72/132	
80	80	18x18	16	72x72	456	192	456	192	4	912	682	453	29.6	96/180	
88	88	20x20	16	80x80	576	224	576	224	4	1152	862	573	28	112/212	

2.0.3 2D Code Datamatrix

Esc 0B {ss} {xxxx} {yyyy} {qq} {i} {o} {m} {a} {cc} {rr} {bb} {dd} {a..a} (\0) {variable_name} (\0) {b...b}

Note: (\0) means binary all zeros or the NULL character.

{ss}	=	Style symbology	
20	=	Data matrix	
{xxxx}	=	Horizontal position	
{yyyy}	=	Vertical position	
{qq}	=	ECC Quality Printer (represents)	ECC Code Standard
		00	0
		01	10
		02	40
		03	50
		04	60
		05	70
		06	80
		07	90
		08	100
		09	110
		10	120
		11	130
		12	140
		13	200
			EAN UCC Standard

{i}	=	Character set [1] to [6]
		1 1..9' ' (500c)
		2 A..Z ' ' (500c)
		3 A..Z 1..9' ' , . - / (500c)
		4 A..Z 1..9 (500c)
		5 ASCII 128 Keyboard (500c)
		6 ISO 8 – Bit International (500c)

{o}	=	Orientation (clockwise rotation)
		0 = 0°
		1 = 90°
		2 = 180°
		3 = 270°

{m}	=	Mirror
		0 = Normal
		1 = Mirror

{a}	=	Automatic size
		1 = Automatic size; SizeCol and SizeRow are disabled

{cc}	=	SizeCol	[10 – 88]
{rr}	=	SizeRow	[10 – 88]
{bb}	=	Border size	[1 – 15]
{dd}	=	Dots multiplier	[1 – 26]

Fixed and variable data fields: -

- {a...a} = Fixed ASCII text, may be left out if not required
- {variable_name} = Name of pre-defined variable, must have a \0 before and after variable_name
- {b...b} = Fixed ASCII text, may be left out if not required

Esc 0B {ss} {xxxx} {yyyy} {qq} {i} {o} {m} {a} {cc} {rr} {bb} {dd} {a a} (\0) {variable_name} (\0) {b...b}
20 0165 0009 13 6 0 0 1 00 00 01 04 QWER \0 VAR0 \0 12q34

2.0.5 2D Code PDF417

Esc 0B {ss} {xxxx} {yyyy} {qq} {o} {m} {a} {eh} {ew} {cc} {rr} {a....a} (\0) {variable_name} (\0) {b....b}

Note: (\0) means binary all zeros or NULL character.

{ss} = Style Symbology 21 = PDF417

{xxxx} = Horizontal position in dots

{yyyy} = Vertical position in dots

{qq}	Characters			
	ECC Quality Printer	ECL Code Standard	Numeric	Alphanumeric
	00	0		
	01	1		
	02	2	1 to 80	120
	03	3	81 to 480	320
	04	4	481 to 960	640
	05	5	961 to 2580	1720
	06	6	2581 to 2710	1721 to 1850
	07	7		
	08	8		

The above shows the recommended minimum ECL

{o} = Orientation (clockwise rotation)
 0 = 0°
 1 = 90°
 2 = 180°
 3 = 270°

{m} = Mirror
 0 = Normal
 1 = Mirror Not implemented

{a} = Aspect Ratio Not implemented

{eh} = Element Height in dots
 01 to 16

{ew} = Element Width in dots
 01 to 09

{cc} = Columns 00 = Automatic sizing
 03 to 10

{rr} = Rows 00 = Automatic sizing
 00 to 09

Fixed and variable data fields: -

{a....a} = Fixed ASCII text, may be left out if not required

{Variable_name} = Name of pre-defined variable, must have a \0 before and after the variable_name

{b....b} = Fixed ASCII text, may be left out if not required

PDF417

0BssxxxxxyyyqqomaehewccrrText and or variable fields
 0B21010000202000007040000Your text field etc

Note: ECC & ECL = Error Correction Level.

2.0.6 2D Code QR Code

Esc 0B {ss} {xxxx} {yyyy} {q} {o} {m} {a} {eh} {ew} {cc} {rr} {a....a} (\0) {variable_name} (\0) {b....b}

Note: (\0) means binary all zeros or NULL character.

{ss} = Style Symbology 22 = QR Code

{xxxx}= Horizontal position in dots

{yyyy}= Vertical position in dots

{qq} =	Characters			
	ECC Quality Printer	ECL Code Standard	Numeric	Alphanumeric
	00	0		
	01	1		
	02	2	1 to 80	120
	03	3	81 to 480	320
	04	4	481 to 960	640
	05	5	961 to 2580	1720
	06	6	2581 to 2710	1721 to 1850
	07	7		
	08	8		

The above shows the recommended minimum ECL

{o} = Orientation (clockwise rotation)

- 0 = 0°
- 1 = 90°
- 2 = 180°
- 3 = 270°

{m} = mirror

- 0 = Normal
- 1 = Mirror Not implemented

{a} = Aspect Ratio Not implemented

{eh} = Element Height in dots
01 to 16

{ew} = Element Width in dots
01 to 09

{cc} = Columns 00 = Automatic sizing
03 to 10

{rr} = Rows 00 = Automatic sizing
00 to 09

Fixed and variable data fields: -

{a....a} = Fixed ASCII text, may be left out if not required

{Variable_name} = Name of pre-defined variable, must have a \0 before and after the variable_name

{b....b} = Fixed ASCII text, may be left out if not required

PDF417

0BsxxxxxyyyqqomaehewccrrText and or variable fields
0B21010000202000007040000Your text field etc

Note: ECC & ECL = Error Correction Level.

2.1 Graphics Field Used in a Format (Max 100 graphic Fields Per Format)

Graphics data is saved in the file store either globally in named files or locally as part of a format. Once they have been saved they can be used within the format. The 'W' command places a graphic onto the label specifying its position, orientation and scale factor. A single graphic can be used many times in a format.

Esc0W {nnnnnnnnn} {xxxx} {yyyy} {s} {o}

{nnnnnnnnnn}	=	Name of global graphic file stored within printer file store
{xxxx}	=	Horizontal position, in dots
{yyyy}	=	Vertical position, in dots
{s}	=	Scaling factor 0 = 1:1 1 = 2:1 2 = 4:1
{o}	=	Orientation (clockwise rotation) 0 = 0° 1 = 90° 2 = 180° 3 = 270°

Where a scaling factor of N:1 applies, each dot and each row are copied N times into the image memory, so that the final image size is N* width x N* height.

2.1.1 Graphics Field Loaded to a Printer

There are two commands for loading graphics into the printer:

Esc0GV {nnnnnnnnnn} {www} {hhh} {S} {ffff} {CR} Esc {d...d}

Esc0V {nnnnnnnnnn} {www} {hhh} {S} {ffff} {CR} Esc {d...d}

They are almost identical, but, the first loads a global graphic and the second loads a local graphic. Apart from the extra 'G' they are the same.

Global graphic is loaded into the printer before the formats are loaded and can be used by any of the formats in the file store. This is ideal for company logo's etc. You can load as many global graphics as the file store will take.

Local graphic is loaded as part of a format and can only be used from within that particular format. This is good for something specific to a format, such as a picture of a particular product.

{nnnnnnnnnn}	=	Name of graphic image (10 alphanumerical characters)
{www}	=	width (bits)
{hhh}	=	height (bits)
{S}	=	Style of graphic image 0 = *. BMP files 1 = *. PCX files 2 = *. BMP files (special application) 3 = *. LZ Compressed files (special applications)
{ffff}	=	Size in bytes of the graphic, not just width x height, unless graphic is rectangular
{CR}	=	Carriage return
Esc	=	Esc.
{d...d}	=	Content (binary data)

No terminator character is used, as the data is in binary format.

Continued on the next page

Note: The data formats are NOT what you might expect.

Type 0 is a non-compressed format that is derived from a windows .bmp BUT our own implementation. Each line of graphic has a 2 byte count followed by the data. This is repeated for each line.

Type 1 is a compressed graphic, using the .pcx method BUT our own implementation. It is NOT the normal windows .pcx. See the example code provided by Open Date.

Type 3 is a compressed graphic, using the .lz method BUT our own implementation. It is NOT the normal windows .lz.

In all cases the width and height of the data is calculated from the picture before sending the data to the printer.

2.1.2 Updating and Overwriting a Local Graphic Field

This command allows a local graphic embedded within a format to be updated without reloading the whole format. The format will be edited in the file store and the image will be updated.

Esc0UV {nnnnnnnnnn} {www} {hhh} {S} {ffff} {CR} Esc {d...d}

2.1.3 Request Global Graphic Names

Esc0ZV

Printer returns: -

```
{nnnnnnnnn1}
{nnnnnnnnn2}
{nnnnnnnnn3}
Etc
OK
```

Where: - nnnnnnnnn1 = First name of Graphic in file store as specified

Where: - nnnnnnnnn2 = Second name of Graphic in file store as specified

Names of graphics will always be padded to 10 characters long, spaces after name. Next graphics will automatically follow on next line. After the last graphic name OK indicates end of list.

2.1.4 Request Global Graphic Information

Esc0ZV {name}

Where: -

{name} = Actual name of graphic in file store

Printer returns: -

```
Esc0V{name} {FFFFFF} Esc {data}
OK
```

Where: -

{name} = Actual name of graphic in file store

{FFFFFF} = File size in bytes (six numerical characters)

{data} = Actual graphic file data

Note: This exactly the same format in which the data was sent.

2.1.5 Delete all Global Graphics Fields

All Commands are followed by: - <CR><LF>

Esc0DV

This command deletes all Global graphics from the printer file store. The current print format will be de-selected, and the printer will be "Off Line".

2.1.6 Delete Individual Global Graphic Fields

This command deletes a specific Global Graphic field from printer memory.

Esc0DV {name}

Where: -

{name} = Actual name of global graphic in printer, this name must be correct and is case sensitive

This command deletes specific global graphic fields from the printer file store. The current print format will be de-selected, and the printer will be "Off Line".

2.1.7 Downloading Global Graphics to the Printer

```
REMnnnnnnnnnnwwwwhhhhSFFFF Print0.bmp
<Esc>0GVPrint0 00000000000000<CR><LF>
```

```
REM Print1.bmp
<Esc>0GVPrint1 00000000000000<CR><LF>
```

The BMP format is a simplified form of the Microsoft BMP format as follows:

The data is single bit per pixel monochrome.

The binary data is arranged as horizontal lines of the image, each line encoded separately.

Each horizontal line of the bit-mapped image is encoded as a word count of the number of data bytes in the line, followed by the binary data starting with the left most pixel. The word count is two bytes MSB, LSB.

Each line is output with subsequent lines running top to bottom until the full height of the image is completed.

The use of the byte count in each horizontal line allows lines with no data at the end to be shortened, i.e. the last bytes of data do not need to be stored.

The PCX format is the standard PCX format as supported by many graphics programs.

The LZ format uses a special form of compression to reduce the download size of a graphic file.

2.2 Variable Insertion Fields

Types of Variable Insertion fields

Variable Insertion Field types	Description	Variable Insertion Field types	Description
0	Fixed text	6	Year types
1	Text using limits	7	Month types
2	Reserved future use	8	12 hour times
3	24 Hour times	9	Shift codes
4	Counter fields	A	Specialist text lines
5	Day types	W	Image memory partitions

A maximum of 59 characters per field is allowed, but only 40 characters can be edited in the Mini Terminal.

If a variable field is more than 40 characters we therefore allow a special kind of variable that gets its contents from a file within the file store. This can be useful when you need to print fixed data with a mixture of variable data. There are two parts to this function.

2.2.1 Variable Insertion Text Field type “A”

Storing the data in the file store:

This Variable is specifically designed for Datamatrix & PDF417 2D Codes.

Use the ‘H’ command that allows a file to be written directly to the file store. The following example shows all alphanumeric characters written into a file called file THT00001. The field type ‘9’ is the type of the file, and should always be “9” (not to be confused with shift code type “9”)

Esc0HfileTHT00001 9000045

Esc1234567890QWERTYUIOP1234567890QWERTYUIOP12345 (max of 100 characters)

Using the file data on a format:

To use the file data you must first define a variable of type ‘A’. In this example the variable is called ‘Var0’ is of type ‘A’ and its value is the name of the file in the file store “fileTHT00001” You can then use the variable in a barcode definition or as a text field.

This type of variable cannot be edited in the Mini Terminal or Touch Screen.

Example 1:

```
Esc0FDocument1
(Speed, Burn(s), Pressure, Homeoff and Rotate are removed for clarity)
Esc0EVar0 AfileTHT00001 (This field cannot be edited)
Esc0B200165000913600100000104\0Var0\0
Esc0Q000000
Esc0K
```

Example 2:

```
Esc0FDocument1
(Speed, Burn(s), Pressure, Homeoff and Rotate are removed for clarity)
Esc0EVar0 AfileTHT00001 (This field cannot be edited)
Esc0EVar1 0123456789A123456789B123456789C12345MAX40 (This field can be edited)
Esc0B200165000913600100000104\0Var0\0Var1\0
Esc0Q000000
Esc0K
```

2.2.2 Variable Insertion Text Field type "0"

Maximum 100 Variable insertion fields Global + Format, 59 characters per field generated, 100 characters line length.

Named variable insertion fields create text or numbers that become part of text or barcode lines. Variables are of different types and each type creates its own sort of information. Many types create data that changes with time, such as counters or dates. By combining a range of variables and fixed text it is possible to produce any kind of text or barcode line.

Variables can be global, defined outside a format, or local, defined inside a format. Global variables can be used by any format in the printer. Local variables can only be used within the format in which they are defined.

Each variable can be used as many times as required (or not at all). This would allow a barcode and a text line to contain the same variable information.

Local variables can be edited using the terminal.

Some of the later parameters to a variable may be optional. This is indicated in the descriptions.

Local variables use the form: -

Esc0EName

and are part of a format.

Global variables use the form: -

Esc0GName

and are defined outside a format.

This section shows the local format (see page 29 for a global example).

Variable Insertion Field (fixed text) type 0

Esc0E {nnnnnnnnn} {0} {ttttttt...}

{nnnnnnnnn} = Name of Variable Insertion Field is a Maximum of 9 Alphanumerical characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name)

{0} = 0 = unchecked variable

{ttttttt} = Text of Variable Insertion

(Field = 59 characters max. 40 of the characters can be edited using the mini terminal)

This type provides simple text that can be easily edited or changed in the printer.

```

0FDocument1
0PSPEED      0100
0PBURN1      0270
0PBURN2      0270
0PPRESSURE   20
0PHOMOFF     03.0
0PROTATE     0
0Evariable1a 0QWERTYUIOP1234567890QWERTYUIOP1234567890
0TArial Bold0252007213001\0Variable1a\0
0Q000000
OK

```

2.3 Updating Existing Variable Fields (Within Currently Selected Format)

It is possible to update an existing variable field within a format already loaded to the printer, this will allow a much higher speed updating of a format, as you do not have to send again any fixed text fields, lines or graphics. Updated fields should be made the same length as the original loaded within the format, but they can be shorter (longer fields may overlap other printing fields).

Example: typical format including a variable batch & counter field

```

0FDocument1
0PSPEED      0100
0PBURN1      0250
0PBURN2      0250
0PPRESSURE   22
0PHOMOFF     03.0
0PROTATE     0
0EBatch      0QWERTY123456
0ECounter0   40001,1,1,1,9999
0TArial Bold0300005413001Batch \0Batch\0
0TArial Bold0300011413001Count \0Counter0\0
0B06029801300120042000\0Batch\0\0Counter0\0
0TArial Bold0411029010001\1102\1\0Batch\0
0TArial Bold0746029010001\1102\1\0Counter0\0
0Q000000
OK

```

Esc0I{Name} {aa.aa}

Also escape code **Esc0UE**{Name} {aa.aa} can also be used.

Where: -

{Name} = Actual name of the variable text field
Max of 7 characters with no punctuation (under score is allowed) followed by space

{aa.aa} = The required information to updated

To update the batch & bounter field you send: -

```

<Esc>0IBatch      0QWETTYR334477<CR>
<Esc>0ICounter0   040029<CR>

```

or

```

<Esc>UEBatch      0QWETTYR334477<CR>
<Esc>UECounter0   040029<CR>

```

This would then change the alphanumeric value of the batch variable to QWTTTYR334477 instead of QWERTY123456. With the Counter0, the next printed number would be 0029.

Printed before updating

Batch QWERTY123456
Count 0006



QWERTY1234560006

Printed after updating the two fields

Batch QWETTY334477
Count 0029



QWETTY3344770029

2.4 Request Format Variables

EscOZI

Printer returns: -

```
{nameL_1} {GenTextL_1}\09{nameL_2} {GenTextL_2}
{nameG_1} {GenTextG_1}\09{nameG_2} {GenTextG_2}
OK
```

Where: -

```
{nameL_1} = Name of first local variable used
{GenTextL_1} = Current generated text from variable
{nameG_1} = Name of first global variable used
{GenTextG_1} = Current generated text from variable
```

2.5 Global Variable Insertion Text Field type “0”

The escape code is a similar command to the variable text field stored with the format. “G” added being a global variable field and is held in memory and not part of the format design, as shown on the previous page. This variable can only be changed, by using a Computer or PLC.

Esc0GE {nnnnnnnnn} {0}{ttttttt...}

{nnnnnnnnn} = Name of variable insertion field is a Maximum of 9 alphanumerical characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name only an underscore)

{0} = 0 = unchecked variable

{ttttttt} = Text of variable Insertion

Example: Esc0GE**VAR1Global** 0QWWERTYU1237<CR><LF>

```
0FGlobalVar1
0PSPEED 0100
0PBURN1 0200
0PBURN2 0200
0PRESSURE 20
0PHOMOFF 03.0
0PROTATE 0
0TArial Bold0110005110001Gabal Variable Text Fields
0TArial Bold0647005110001\0VAR1Global\0
0Q000000
OK
0SGlobalVar1
```

The same command is also used to update/ change the global variable field. Being global this will update the field if it's on or off line.

2.5.1 Request Global Text Variable Names

Esc0ZE

Printer returns:-

```
{nnnnnnnnn1}
{nnnnnnnnn2}
{nnnnnnnnn3}
Etc.....
OK
```

Where: - nnnnnnnn1 = First name of global text in file store as specified

Where: - nnnnnnnn2 = Second name of global text in file store as specified

Names of texts will always be padded to 10 characters long, spaces after name. Next text will automatically follow on next line. After last text name OK will indicate end of list.

2.6 Request Global Text Variable Information

Esc0ZE {name}

Where: -

{name} = Actual name of text in file store

Printer returns: -

EscGE{name} {T} {S} {O} {aa...aa}
OK

Where: -

{name} = Actual name of global text variable in file store
{T} = Type of global variable field
{S} = Style of the type of global variable field
{O} = Offset characters
{aa..aa} = User defined variable used text fields

2.7 Deleting Global Text Fields & All Global Text Fields

2.7.1 Delete

All Commands are followed by: - <CR><LF>

Esc0DE

This command deletes all global graphics from the printer file store. The current print format will be de-selected, and the printer will be "Off Line".

2.7.2 Delete Individual Global Graphic Fields

This command deletes a specific global text field from printer memory.

Esc0DE {name}

Where: -

{name} = Actual name of global graphic in printer, this name must be correct and is case sensitive

This command deletes a specific global graphic fields from the printer file store. The current print format will be de-selected, and the printer will be "Off Line".

2.8 Variable Insertion Text Field type "1"(Using Limits)

Esc0E {nnnnnnnn} {1} {ttttttt...} {,} {LLLLL} {,} {HHHHH}

{nnnnnnnn} = Name of variable insertion field, max 9 alphanumerical characters (no punctuation characters allowed other than an underscore) followed by a minimum of one space character

{1} = Variable set up with high and low limits (numerical text only)

{ttttttt} = Text of variable insertion field

{,} = Separator character (comma)

{LLLLL} = Lower limit of variable text

{,} = Separator character (comma)

{HHHHH} = Higher limit of variable field

The text must be number between the lower limit and the upper limit or an error will be generated. The number range is -2147483646 to 2147483647.

2.8.1 Variable Insertion Fields Type “4” Counter Fields

This function is a special variable insertion counter field that allows the user to specify: -

- Start Number**
- Justification**
- Increment**
- Pallet Count**
- Counter rollover or Batch Quantity**

Maximum 20 fields per format design.

Esc0E {nnnnnnnnnn} {4} {sssssss.....} {,} {J} {,} {iiiiiii...} {,} {pppp..} {,} {rrrrrrr...}

Where: -

- Esc E = Escape code to tell printer variable insertion field
- {nnnnnnnnnn} = Name of variable insertion field, max 10 alphanumeric characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name)
- {4} = Specifies counter type variable insertion field
- {sssssss} = Counter start number, leading zero's specify the maximum counter capacity and printability. Leading spaces before the start number indicate the number will be printed with leading spaces (see justification parameter)
- {,} = Separator character (comma)
- {J} = Justification character (0 = left), (1 = right)
- {,} = Separator character (comma)
- {iiiiiii...} = Increment amount, specifies the amount each print has to advance
- {,} = Separator character (comma)
- {pppp..} = Pallet count/Repeating sequence number, specifies the number of the same prints required before incrementing to the next count
- {,} = Separator character (comma)
- {rrrrrr} = Roll over number, specifies the last printed number before the counter resets to the original start number

Specifying a Stop Number

To specify the number of prints required, or stop number simply “edit” the quantity line of the format.

Example 1:

0ECounter0 4000001,1,1,1,999999999 Incrementing after every print (prints the leading zeros)
000001, 000002, 000003, etc to max of 999999999

Example 2:

0ECounter0 4000001,**0**,1,1,999999999 Incrementing after every print (doesn't print the leading zeros)
1,2,3,4,5,6,7,8,9,10,11,12, etc to max of 999999999

Example 3:

0ECounter0 4000002,1,**2**,1,999999999 Increments as 2,4,6,8,10,12 etc

Example 4:

0ECounter0 4000001,1,1,**2**,999999999 Pallet counter (prints two labels with the same number)

2.8.2 Variable Insertion Fields Type “4” Counter Fields

The counter field can also be change using <Esc>0ICounter0 4000027 the next printed count would be 000027 (for more information on updating variable text field see page 26 and also see the System Variable Description Manual regarding SYSCOUNTWB for storing the last printed count).

ECounter0 4000001,1,1,1,999999,990002

On reselection of the format from the storage memory would print 990003.

2.8.3 Variable Insertion field type “3” (24 Hour Time)

Esc0E {nnnnnnnnnn} {3} {T} {oooo}

{nnnnnnnnnn} = Name of variable insertion field, max 10 alphanumerical characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name)

{3} = Variable time indication

[Optional parameters from here]

{T} = Type of time insertion field (see below listing)

{oooo} = Amount of minutes to offset (valid range -1439 to +1439)

Types of 24 Hour Time Insertion Fields: -

1 = hh:mm:ss (24 hour) 2 = hh:mm (24 hour)

3 = hhmm (24 hour) 4 = hh (24 hour)

5 = mm 6 = ss

Each field is always 2 digits with '0' padding on the left.

The times and dates are all taken from the real time clock which should be set correctly. The offset field allows the time to be shifted backwards and forwards by up to 1 day. Each variable field has its own offset and will not affect other time fields.

All 24 hour time types update automatically.

2.8.4 Variable Insertion Field Type 8 (12 Hour Time)

Esc0E {nnnnnnnnnn} {8} {T} {offset} {,} {am} {,} {pm}

{nnnnnnnnnn} = Name of Variable Insertion Field, Max 10 Alphanumerical characters. (No punctuation characters allowed) followed by a minimum of one space character. (No spaces allowed within name)

{8} = Variable time indication

{T} = Type of time insertion field (see listing in next section)

[Optional parameters from here]

{offset}	=	Amount of minutes to offset (valid range -1439 to +1439)
{,}	=	Separator character (comma)
{am}	=	Optional first 12 hour indication, change as required
{,}	=	Separator character (comma)
{pm}	=	Optional second 12 hour indication, change as required

Types of 12 Hour Time Insertion Fields: -

1 = hh:mm:ss	(12 hour)	2 = hh:mm	(12 hour)
3 = hhmm	(12 hour)	4 = hh	(12 hour)
5 = mm		6 = ss	

Each field is always 2 digits with '0' padding on the left. The optional am and pm fields are added to the end of the numeric output.

The times and dates are all taken from the real time clock which should be set correctly. The offset field allows the time to be shifted backwards and forwards by up to 1 day. Each variable field has its own offset and will not affect other time fields.

All 12 hour time types update automatically.

2.8.5 Variable Insertion field type 5 (Day Types)

Esc0E {nnnnnnnnnn} {5} {T} {oooo} {,} {rrrr} {,} {mmmm}

{nnnnnnnnnn} = Name of variable insertion field, max 10 alphanumerical characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name)

{5} = Variable day indication

{T} = Type of time insertion field (see below listing)

[Optional parameters from here]

{oooo} = Number of days to offset (valid range -9,999 to 9,999)

{,} = Separator character (comma)

{rrrr} = Offset from midnight in minutes to rollover date (valid range -1439 to +1439)

{,} = Separator character (comma)

{mmmm} = Number of months to offset, from printer date (valid range -9,999 to 9,999)

All Day fields update automatically. You can't have month and day offset at the same time, one at least must be zero.

Types of Day Insertion Fields: -

1 = dd	(Day of month, 2 numerical digits)
2 = d	(Day of week as a numerical number, uses SYSDAY1 global variable)
3 = ddd	(Day of week, 3 alphabetical, uses SYSDAY3 global variable)
4 = ddddddd....	(Day of week, alphabetical spelling, uses SYSDAY global variable)
5 = JJJ	(Julian day of year, day 366 = 29 Feb)
6 = JJJ	(Julian day of year, day 366 = 31 December)

2.8.6 Variable Insertion Field Type 6 (Year Types)

Esc0E {nnnnnnnnn} {6} {T} {oooo} {,} {rrrr} {,} {mmmm}

{nnnnnnnnn} = Name of variable insertion field, max 10 alphanumeric characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name)

{6} = Variable year indication

{T} = Type of time insertion field (see below listing)

[Optional parameters from here]

{oooo} = Number of days to offset (valid range -9,999 to 9.999)

{,} = Separator character (comma)

{rrrr} = Offset from midnight in minutes to rollover date (valid range -1439 to +1439)

{,} = Separator character (comma)

{mmmm} = Number of months to offset, from printer date (valid range -9,999 to 9.999)

All year fields update automatically. Month and day cannot be offset together, one must be zero.

Types of Year Insertion Fields:-

1 = Y (Year, 1 numerical digits 1998 = 8)

2 = YY (Year, 2 numerical digits 1998 = 98)

3 = YYYY (Year, 4 numerical digits 1998 = 1998)

4 = WW (Week number of year, 2 numerical digits)

2.8.7 Variable Insertion Field (Month Types) Type 7

Esc0E {nnnnnnnnn} {7} {T} {oooo} {,} {rrrr} {,} {mmmm}

{nnnnnnnnn} = Name of variable insertion field, max 10 alphanumeric characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name)

{7} = Variable month indication

{T} = Type of time insertion field (see listing in next section)

{oooo} = Number of days to offset (valid range -9,999 to 9.999)

{,} = Separator character (comma)

{rrrr} = Offset from midnight in minutes to rollover date (valid range -1439 to +1439)

{,} = Separator character (comma)

{mmmm} = Number of months to offset, from printer date (valid range -9,999 to 9.999)

All year fields update automatically. Month and day cannot be offset together, one must be zero.

Continued on the next page

Types of Month Insertion Fields: -

- 1 = MM (Month, 2 numerical digits, JAN = 01)
- 2 = MMM (Month, 3 alphabetical characters, JAN = JAN) uses **SYSMON3** global variable
- 3 = MMMMM... (Month, alphabetical characters, APRIL = APRIL) uses **SYSMONTH** global variable
- 4 = M (Month, single alphabetical character, JAN = A) uses **SYSMON1** global variable

2.9 Shift Codes Type 9 (Hours of Shift)

Max 24 shift periods in one day, max 4 shift variables.

Inserts the “**SYSSHIFT**” text into the relevant fields. **See section 2.2.1**

Esc0E {nnnnnnnnn} {9} {TTT1} {,} {TTT2} {,} {TTT3} {,} {TTT4} {,} {TTT5} {,} {TTT6}

{nnnnnnnnn} = Name of variable insertion field, max 10 alphanumeric characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name)

{TTT1} = Start time of 1st shift period (24 hour clock)

{,} = Separator character (comma)

{TTT2} = Start time of 2nd shift period (24 hour clock)

[Optional parameters from here]

{,} = Separator character (comma)

{TTT3} = Start time of 3rd shift period (24 hour clock)

{,} = Separator character (comma)

{TTT4} = Start time of 4th shift period (24 hour clock)

{,} = Separator character (comma)

{TTT5} = Start time of 5th shift period (24 hour clock)

{,} = Separator character (comma)

{TTT6} = Start time of 6th shift period (24 hour clock)

If only 3 shifts are required per day, then only specify those values {TTT1} through to {TTT3}.

2.9.1 Specifying Rollover Times of Automatic Date Fields

Normally automatic date fields always rollover (increment to next date) at midnight. It is possible to change the time that the specified format that is loaded to the printer will update using offsets.

Rollover times are always set from midnight, either plus or minus in minutes (-1439 to + 1439) range.

This does not change the actual time of the printer.

Example: You want to print a normal day field type 5 with no day offsets but rollover the day at 6am 0600 hours the next day, instead of midnight 2400.

Esc0EDate 51000,360 also can be sent as: - Esc0EDate 51+0,360,0

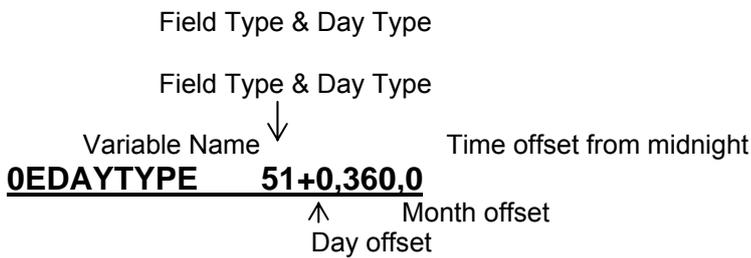
Where: - DAYTYPE = Name of variable

- 5 = Day type variable (field type)
- 1 = Style of day type
- 000 = Offset in days (not used) can be +0
- ,
- 360 = Offset time in minutes from midnight
- ,
- 0 = Month offset

```

0FHOUROFF
0PSPEED      0100
0PBURN1      0170
0PBURN2      0170
0PPRESSURE   20
0PHOMOFF     03.0
0PROTATE     0
0EDAYTYPE    51+0,360,0
0MONTHTYPE   71+0,360,0
0YEARTYPE    62+0,360,0
0TArial Bold0353008713001\0DAYTYPE\0\0MONTHTYPE\0\0YEARTYPE\0
0Q000000
OK
    
```

All of above presided by Esc



2.9.2 Variable Type W

This variable allows multiple separate images to be created within a single format. Using SYSILEN the image memory that is normally used as a single image is split into a number of smaller sections. Each section can then be written with a different image all from the same format. When printing, a different image can be selected instantaneously for each print.

Used in conjunction with SYSCOMTRIG for continuous printers and SYSISEL for Intermittent printers.

Make a Multi-Image printer by splitting up the image memory into several sections and then store each image into a section.

This variable allows multi images in a format to be loaded direct to the image sections, which have previously been created by using the variable SYSILEN, and seeing the result in SYSNUM.

SYSNUM shows the number of images allowed in a single image plane.

Esc0E {nnnnnnnnnn} {W} {0} {1}

Where: -

{nnnnnnnnnn}	=	Name of variable insertion field, max 10 alphanumeric characters (no punctuation characters allowed) followed by a minimum of one space character (no spaces allowed within name). This field is a dummy and any name can be used
{W}	=	Image memory store type variable
{0}	=	Image plane (0 = Write to both image planes) (1 = Write to image plane 1) (2 = Write to image plane 2)
{1}	=	Number of the image section being stored

Note: Image plane 0 has to be used for updating fields such as Counter fields

If you require any dates or counters you must use both image planes (allowing for updates etc.).

If the images are just graphics or text that do not need updating, you can specify individual planes for storing the images. This allows you to use both image planes individually, allowing double the SYSNUM number of images.

To use this variable, place it before one or more image generating lines in the format. All those images will appear in that section. To put data into another section have another variable, i.e.:

```
Esc0Edummy1 W11
Esc0B20016500091300100000104A212\Øvariable1\Ø
Esc0TARIALBD 00500050120005212\Øvariable1\Ø\Øvariable2\Ø
Esc0Edummy1 W12
Esc0B20016500091300100000104A212\Øvariable1\Ø
Esc0TARIALBD 00500050120005212\Øvariable1\Ø\Øvariable2\Ø
Esc0W {nnnnnnnnnn} {xxxx} {yyyy} {s} {o}
Esc0Edummy1 W21
Esc0B20016500091300100000104A212\Øvariable1\Ø
Esc0TARIALBD 00500050120005212\Øvariable1\Ø\Øvariable2\Ø
```

This puts:

- barcode and text into plane1/section1
- barcode and text and graphics into plane1/section2
- barcode and text into plane2/section1

There is a manual for multi printing available on request.

3.0 Select Format

This command permits the selection of a format from the file store, which is then loaded to the printer memory ready for printing.

Esc0S{Name}

Where: -

{name} = Actual name of format in file store, this name must be correct and is case sensitive

3.0.1 De-Select Format

This command de-selects the currently selected format being printed, and parks the printhead.

Esc0S

Request Acknowledge

Esc0Z

Printer returns: - Esc Z Type mmmm,xxxxx
OK

Where (mmm,xxxxx) = Model no of printer & Software Version number

Z Type 107M, 5.012957IA
OK

3.0.2 Request Format Data

Esc0ZF

Printer returns: - All format names stored within the printer.

F {name 1}
F {name 2}
F {name 3}
Etc
OK

You can also upload a format file by specifying: -

Esc0ZF {name1}

Printer replies with: -

Esc0FName
Esc0PSPEED 0100
Esc0PBURN1 0220
Esc0PBURN2 0220
Esc0PPRESSURE 23
Esc0PROTATE 0
Esc0OFFSET 050.0
Esc0TARI ALBD 00500120120000pen Date Equipment
Esc0Q000000
Esc0K
OK

Note: If the format name is not recognised the printer will just return OK and the error "List file unknown" will be displayed on the terminal.

Esc0K

This Command identifies the end of a format file.

3.0.3 Delete All Formats

This command deletes all formats from file store & both printer memories.

Esc0DF

This command deletes all fonts from the printer File store. The current print format will be de-selected, and the printer will be "Off Line".

3.0.4 Delete Individual Formats

This command deletes a specific format from the printer file store. If this is the currently selected/printing format then it will be de-selected, and the printer will be "Off Line".

Esc0DF {name}

Where: -
{name} = Actual name of format in printer, this name must be correct and is case sensitive

Printer returns:- OK

3.0.5 Print Quantity (Used Within a Format Only)

Esc0Q {qqqqqq}

Where: -
{qqqqqq} = Quantity of prints required, 000001 - 999999 sets a batch of that quantity (000000 = continuously printing until stopped by operator)

3.0.6 Request Print Quantity

Esc0ZQ

Printer returns: - Esc0Q {000000} {,} {PPPPPP}
OK

Where: -
{000000} = Quantity Value set in format design
{,} = Comma (field separator)
{PPPPPP} = Quantity of prints done, within selected format

3.0.7 Request Total Prints

Esc0ZT

Printer returns: - Esc0T {0000000000}
OK

Where: -
{0000000000} = Total number of print cycles of printer

3.0.8 Request Format Name

Esc0ZN

Printer returns: - Esc0N {nnnnnnnnnnn}
OK

Where: -
{nnnnnnnnn} = The name of the format being printed

3.0.9 Request All Print Parameters

Within the format being printed.

Esc0ZP

Printer replies with all format parameters, and the associated values.

EscXSPEED	0050	
EscXBURN1	0450	
EscXBURN2	0450	
EscXPPRESSURE	20	
EscXPEEL	07.0	(Not normally used)
EscXROTATE	0	
EscXOFFSET	050.0	(Continuous Machines only)
EscXHOMOFF	26.0	(Intermittent machines only)
OK		

3.1 Request an Individual Print Parameter

Within the format being printed.

Esc0P{name}

Where: - P = Escape code for print parameter variables
 {name} = Description of parameter field to change, this field must be ten characters long with printable ASCII characters followed by trailing spaces (names of fields are pre-defined as list below)

Printer returns: - EscX{NNNN}{VVVV}
 OK

Where: - {NNNN} = Name of parameter requested
 {VVVV} = Value of parameter requested

3.1.1 Specifying Daylight Saving Dates

You can program in the exact dates the printer needs to adjust itself to allow for daylight savings. The customer must specify both dates at which the time clock should adjust automatically, and whether the first date specified has the clock going forward or backwards by 1 hour.

It is assumed that any clock going forward will happen at 1am (0100 hours).

It is assumed that any clock going backward will happen at 2am (0200 hours).

The settings are made in the form of a variable as shown below. This can be set at any time but will only be relevant for the particular year that you set the date at.

Esc0 {XSYSDSAVE} {ddd1} {,} {ddd2} {,} {h}

{XSYSDSAVE}	=	System variable name
{ddd1}	=	The first date when a change is to take place (4 digits MMDD)
{,}	=	Field separator (comma)
{ddd2}	=	The second date when a change is to take place (4 digits MMDD)
{,}	=	Field separator (comma)
{h}	=	Hour advance/retard flag 0 = Forwards 1 hour at date 1 = Backwards 1 hour at first date

Note: You must reset the time in the printer after sending the day saving variable.

3.1.2 Requesting Daylight Saving

Esc0 {XSYSDSAVE}

Printer returns:-

- {XSYSDSAVE} = System variable name
- {0416} = The first date when a change is to take place (4 digits MMDD)
- {,} = Field separator (comma)
- {1121} = The second date when a change is to take place (4 digits MMDD)
- {,} = Field separator (comma)
- {h} = Hour advance/retard flag
 0 = Forwards 1 hour
 1 = Backwards 1 hour

3.1.3 Special Control Variables Used Within Automatic Date Codes

Within the printer are some dedicated system variables that control the way that the date variable fields work. When a date variable is used the text created comes from these system variables. By changing the system variable you can change the text output. The system variables involved are:-

- Esc0SYSDAY** = **Monday etc (alphabetical characters)**
- Esc0SYSDAY3** = **MON (3 alphanumeric characters)**
- Esc0SYSDAY1** = **1 (1 alphanumeric character)**
- Esc0SYSMONTH** = **JANUARY (alphabetical characters)**
- Esc0SYSMON3** = **JAN (3 alphabetical characters)**
- Esc0SYSMON1** = **1 (1 alphanumeric character)**
- Esc0SYSSHIFT** = **S1 (max 5 characters)**

The above variables default to the English language. If customisation is required, please see the examples below that show how each type is created and sent to the printer.

Change the required text in the text string. Do not forget to separate each field with a coma.

Esc Code	Global Text Name	Variable type	Text string information
Esc0X	SYSDAY	5	Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday
Esc0X	SYSDAY3	5	MON,TUE,WED,THU,FRI,SAT,SUN
Esc0X	SYSDAY1	5	1,2,3,4,5,6,7
Esc0X	SYSMONTH	7	January, February, March, April, May, June, July, August Etc
Esc0X	SYSMON3	7	JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
Esc0X	SYSMON1	7	A,B,C,D,E,F,G,H,J,K,L,M
Esc0X	SYSSHIFT	9	S1,S2,S3,S4,S5,S6

Note: The printer always assumes that Monday is the start of the week, and January is the first month of any year.

The above fields cannot be deleted from the printer, only overwritten. An initialisation (Esc0CINew) of the printer will change all the variables back to the default settings & English.

3.2 Commands to Control and Monitor the Printer

All commands are followed by <CR><LF>

3.2.1 Auto Print Command From Remote Computer or Similar

Esc0GP

This command allows a user to command a print signal from a computer or similar.

3.2.2 Quantity Command

Esc0GQ {NNNNNN}

Where: -

GQ = Escape codes for sending new quantity command

{NNNNNN} = Actual new quantity value (up to six digits)

This command allows the operator to reset the quantity counter to zero and to specify a new quantity. The printer will then print up to that number of labels and then stop.

3.2.3 Change Mechanical Variables

Esc0XP {name} {data}

{name} = The actual specific system variable you wish to change

{data} = Content of data field, relevant to field description (see limits shown below)

Example: 0XPACCRIBBON

Esc0ZY Will return all the mechanical variable fields

Esc0ZY {name} {data} Will return a mechanical variable field and its value

(See the separate manuals giving full descriptions of system & mechanical variables).

System variables can be changed as global variables or as local. If they are not contained in a format they are global as they affect all the formats stored in the printer and the functionality of the printer.

If they are added to a format they become localized, as every time a different format is select the variable field would also be changed (see the example on the following page).

3.2.4 Request System Variables

Esc0ZX

This command returns a list of the system variable names and their current values

3.2.5 Request Individual System Variable

Esc 0ZX {Name}

Where {name} is the actual system variable name.

This will return the value variable name and the current value. If the name is not recognised the error "list parameter unknown" will be displayed on the terminal.

Printer returns:-

```
name ???
OK
```

3.2.6 Delete System Variables Changes

Esc0DX

This command restores all system variables back to the original default settings.

3.2.7 Delete Individual System Variable Changes

Esc0DX {name}

This command restores a system specific variable back to the original setting.

Where {name} is the actual system variable name.

Printer returns: - OK

If the name is not recognised the error "Sys param unknown" will be displayed on the terminal.

3.2.8 Pause Printing

Esc0CC

This command requires no additional parameters. It causes the printer to pause printing after the operation on the current format is complete.

3.2.9 Resume Print

Esc0CR

This command requires no additional parameters. It causes the printer to resume printing from the point at which it was stopped by the pause print command.

3.3 Request Printer Status

Esc0ZS

Where: - {nnnn} = Status text Identifier

 { , } = Comma (field separator)

 {ttttttttt...t} = The status of the printer (Cassette off etc.)

Printer returns: - Esc0S {nnnn} { , } {ttttttttt...t}

3.3.1 Printer Output to Communications Line

After a print cycle, the printer replies on the communications port with data to help synchronise downloads or to check that printing has happened correctly. The reply data can be:-

**EscDONE or
EscREADY or
Specialist Text lines (Created with 'Y')**

This is very useful for automatically downloading the next print in a queue etc.

Because the printer has dual image memories, you can decide how to utilise the memories for the best configuration for your application. Using the Esc0SYSUPMODE feature you can determine the kind of output you require: -

Esc0SYSUPMOD = 0 no output to communications port.

Esc0SYSUPMOD = 1 Output to communications port, sends "DONE" after finished print

Esc0SYSUPMOD = 2 Output to communications port, sends "READY" as printing starts The quicker option.

Esc0SYSUPMOD = 3 Output to communications line, sends "READY" whilst printing but gives the error "Print before image update" if an update has not happened since the last print.

Esc0SYSUPMOD = 4 Output 'Y' line to communications port after finished printing

Esc0SYSUPMOD = 5 Output 'Y' line to communications port, as printing starts

3.3.2 Request Memory

Esc0ZM

Where:- {aa....aa} = The maximum file store memory (bytes)
 { , } = comma, field separator only
 {bb....bb} = The amount of available memory in file store (bytes)

Printer returns:- EscM {aa....aa} { , } {bb....bb}
 OK

3.3.3 Request Errors

Esc0ZZ

Where: - {nnnn1} = Error text Identifier
 { , } = Comma (field separator)

Printer returns:- EscZ {nnnn1} { , } {nnnn2}
 OK

3.3.4 Clear Errors

Esc0DZ

This command allows user to clear all errors from a computer.

Printer returns: - OK

3.3.5 Request a Range of Variables and Parameters

Esc0Z9

Printer returns: -

```
EscZ{n1}\01 {Id} \02\01{Vn}\01{R1}\02\01{R2}\02\01{R3}\02\01{Pm}\02
\01{M1}\02\01{{M2}\02\01{Pc}\02\01{S1}\02\01{S2}\02
\01\03{E1}\04\03{E2}\04\02
\01{Fn}\02
\01\03{L1}\04\03{I1}\04\03{L2}\04\03{I2}\04\02
```

Where: n1} = Network number
 {Id} = Printer Id
 {Vn} = Version number
 {R1} = Ribbon left
 {R2} = Total ribbon
 {R3} = Ribbon warning length
 {Pm} = Prints per minute
 {M1} = Total memory

{M2} = Memory available
 {Pc} = Prints of current label
 {S1} = Status message number
 {S2} = Status message text
 {E1} = First error message
 {E2} = Second error message
 All error messages are output.
 {Fn} = Format name
 {L1} = Name of first format variable
 {I1} = Value of first format variable
 {L2} = Name of second format variable
 {I2} = Value of second format variable
 All variables used in a format, global or local will be listed.

3.3.6 Initialise Part Printer

Esc0CINIT

This command erases all stored global graphics, global variables, formats and fonts.

The system variables & mechanical variables are left in the printer with their previously programmed values.

3.3.7 Initialise All Printer

Esc0CINEW

This command resets the printer firmware to the default settings (system variables & mechanical variables). Also erases all stored global graphics, global variables, formats and fonts and basically any programmed settings you have made will be totally cleared.

3.3.8 Request All Files

Esc0ZH{T}

This command allows you to view all files in the printer file store, with their type identification numbers.

Example of files received: -

```

ARIAL _____ 3
TETLEY ONE _____ 4
TETLEY TWO _____ 4
LOT _____ 8
BEST BEFORE _____ 8
OK
  
```

The {T} type parameter is optional. If it is included only files of that type will be listed.

3.3.9 Request Individual File Information

Esc0ZH {name}{T}

This requests that a particular named file is uploaded from the printer. The name must be made up to 15 character with additional spaces. The type number must be included.

Printer returns:

```

{All file data}
OK
  
```

3.4 Delete File From File Store

Esc0DH {name} {n}

Where: -

{name} = Name of file being sent to printer (15 characters including spaces)

{n} = Application Identifier (see list section 2.8.5)

Printer returns: - OK

3.4.1 Selecting a New Burn File

Esc0X {SYSBFILE} {name of file}

Where: -

{SYSBFILE} = Name of the Variable being used

{name of file} = Name of the file being loaded to the file store.

Note: The printer should then be switched off and then re-started to ensure the format has been updated.

3.4.2 Program Printer

Esc0CIPROG

This command generates a printer reset. A new control program can then be downloaded from the computer as the printer restarts (must be in capital letters).

3.4.3 Loading a File to the Printer

**Esc0H {name} {n} {XXXXXX} {NNNNNN}
Esc {file of data}**

Where: -

{name} = Name of file being sent to printer (15 characters including spaces)

{n} = Application Identifier (see list below)

{XXXXXX} = Size of file being loaded to the printer

{NNNNNN} = Length of file transferred

Identifier types: -

Terminal files	=	1
Printer files	=	2
Font files	=	3 (True type or SFP fonts)
Format Files	=	4
System files	=	5 (Internal generated files)
Graphic files	=	6
Burn Files	=	7
Configuration files	=	8
Variable files	=	9 (used with the "Y" command)

This command is used for restoring printer status and downloading burn and configuration files. The identifier type is the equivalent of the file extension for files in windows.

3.4.4 Text Line Output to Communications Port

This feature allows a used defined text line to be generated and sent via a communications port to an attached computer. The text is regenerated and sent after every print. The text can be fixed or contain a mixture of fixed and variable data exactly like a text field on a label. A standard printer status text field can also be included after the user-defined data.

This command only works if the variable SYSUPMOD is activated. See below for the options:-

- Esc0SYSUPMOD = 0 (no output to communications port)
- Esc0SYSUPMOD = 1 (Output to communications port, sends "DONE" after finished print)
- Esc0SYSUPMOD = 2 (Output to communications port, sends "READY" as printing starts. The quicker option).
- Esc0SYSUPMOD = 3 (Output to communications line, sends "READY" whilst printing but gives error "Print before image update" if an update has not happened since the last print).
- Esc0SYSUPMOD = 4 (Output 'Y' line to communications port after finished printing)
- Esc0SYSUPMOD = 5 (Output 'Y' line to communications port as printing starts)

3.4.5 The Text Output Line is Defined in a Format as Follows

Esc0Y { P } { S } {Fixed Text} {Variable Text} {Fixed Text} ...

Where: -

- {P} = Communication port number
- {S} = Status of Printer
0 = not included
1 = included
- {Fixed Text} = Fixed information
- {Variable Text} = Variable Name, actual value of the variable information output to communications port
- {Fixed Text} = Fixed information
- ... = More fixed and variable sections

You would normally want the text output to the communications port that from which you send the format (set value of 0.) However, you may wish to send the data to a different computer via a different port. This is controlled with the communication port number: -

- 0 = Active communications port
- 1 = Serial Port
- 2 = USB Port
- 3 = Ethernet (TCP/IP) Port
- 4 = Secondary Ethernet (TCP/IP) Port, we might add this later.

Example Format: -

```

0FCounter
0PSPEED 0100
0PBURN1 0240
0PBURN2 0240
0PPRESSURE20
0PHOMOFF 03.0
0PROTATION 0
0ECounter1 40001,1,1,1,1,9999
0ESeconds 36
0TArial 0010010016001Box No.\0Counter1\0

```

```
0Y01COUNTER1, \0Counter1\0;TIMES,\0Seconds\0:  
0Q00000  
OK
```

Example of Output from Printer: - <ESC>Y01COUNTER1,0001;TIMES,33:
6108:Print on Line
OK

3.4.6 Request Printer Type & Software Version

Esc0Z

Printer returns: - Z Type 107M,4.00 IA

3.4.7 Request Mechanical Variables

Esc0ZY

Where {M1} {M2} {M3} are list of the mechanical variable names and there current values.

Printer returns: - Esc0X {M1}
Esc0X {M2}
Esc0X {M3}
OK

3.4.8 Request Individual Mechanical variable

Esc 0ZY {Name}

Where {name} is the Actual system variable name. This will return the variable name and the current value.

(See the separate manuals for mechanical variables & system variables that have a full description of the function and ranges available)

4.0 Passwords

Up to 9 password levels with individual descriptions and access codes may be programmed within the printer, the levels allow the user to specify which staff can access or program specific types of data. Normally the higher level of password will mean stricter security requirements. Utilising the levels correctly will allow a Manager to restrict his personnel to dedicated functions only. Passwords are numeric values between 1 and 8 characters long.

Default Password Settings

Password	Default Code	Default Name	Functions
Level 1 SYSPASS1	1111	OPERATOR	Selecting formats Editing home offset Editing quantity of format Can be disable
Level 2 SYSPASS2	2222	CHARGEHAND	Access to level 1 Editing text fields (fixed & variable) Saving format as another name Can be disabled
Level 3 SYSPASS3	3333	ENGINEER	Access to levels 1 & 2 Editing format parameters Editing service functions Editing format "X" & "Y" positions Can be disabled
Level 4 SYSPASS4	4444	MANAGER	Access to levels 1, 2 & 3 Editing barcode variables Editing counter variables Can be disabled
Level 5 SYSPASS5	5555	SUPERVISOR	Access to levels 1, 2, 3 & 4 Access to supervisor menu Can be disabled.
Level 6 SYSPASS6	XXXX	OPENDATE SERVICE	Access to levels 1, 2, 3, 4 & 5 Allows service engineer ccess to customer machines
Level 7 SYSPASS8	XXXX	PROGRAMMING	Access to levels 1, 2, 3, 4, 5 & 6 Access to password set up menus
Level 8 SYSPASS8	XXXX	OPEN DATE TECHNICIAN	Allows access to level 8 only Programming of printer set up menu, building & testing of machines only
Level 9 SYSPASS9	XXXX	SECURITY	Allows access to all Levels

The above security levels are used by the mini-terminal, to allow or restrict access to specific functions of the printer. Editing of the formats or changing the printer settings, can then be restricted to prevent unauthorised access.

Once the required password has been entered correctly, access is then allowed, to only those specific functions of the program that are preceded with your security level number or less. Whichever level of password that is entered, will depend upon the type of access allowed. Entering a level 4 password will allow access to level 4 security codes and all functions of the lower security levels (1 to 3).

4.1 Setting Passwords (Activation, Descriptions & Password Codes)

Using these system variable codes, allows users to specify the name and pass-code of a specific system password. The passwords are system variables and only used by the mini-terminal display.

Esc0 {XSYSPASS1} {1} {,} {DDDDDDDD} {,} {NNNN..}

where:-

{XSYSPASS1}	=	System Password name
{1}	=	enable/disable flag
{,}	=	field separator (comma)
{DDDDDD...}	=	password description (18 alphanumerical characters max)
{,}	=	field separator (comma)
{NNNN..}	=	Password (1 to 8 numeric characters)

The whole string (excluding the XSYSPASSN) can be up to 34 characters long allowing a name up to 23 characters long.

4.2 Requesting Individual Passwords

Esc0 {ZSYSPASS1}

Where:-

{ZSYSPASS1}	=	System variable name for password for level 1
-------------	---	---

Printer returns:-

Esc0 {XSYSPASS1} {1} {,} {DDDDDDDD} {,} {NNNN..}

Where:-

{ZSYSPASS1}	=	Escape code to request password for level
{1}	=	Enable/disable flag
{,}	=	Field separator (comma)
{DDDDDD...}	=	Password description (18 alphanumerical characters max)
{,}	=	Field separator (comma)
{NNNN..}	=	Password (1 to 8 numeric characters)