

EN

# MetoPrint

## PROGRAMMING MANUAL TSPL-EZ™



**METO**®  
your retail label expert



## CONTENTS

How to Read .....	9
Document Conventions.....	10
Object Position Calculation .....	11
<b>1. Setup And System Commands.....</b>	<b>12</b>
SIZE.....	12
GAP .....	13
GAPDETECT .....	15
BLINEDECTECT .....	16
AUTODETECT .....	17
BLINE .....	18
OFFSET .....	20
SPEED .....	21
DENSITY .....	22
DIRECTION AND MIRROR IMAGE.....	23
REFERENCE.....	24
SHIFT .....	25
COUNTRY .....	26
CODEPAGE .....	27
CLS .....	29
FEED .....	30
BACKFEED & BACKUP.....	31
FORMFEED .....	32
HOME .....	33
PRINT .....	34
SOUND .....	35
CUT .....	36
LIMITFEED .....	37
SELFTEST .....	38
EOJ .....	40
DELAY .....	41
DISPLAY .....	42
INITIALPRINTER .....	44
MENU .....	45
<b>2. Label Formatting Commands .....</b>	<b>48</b>
BAR .....	48



## CONTENTS

BARCODE .....	49
TLC39 .....	58
BITMAP .....	60
BOX .....	62
CIRCLE .....	63
ELLIPSE.....	64
CODABLOCK F MODE .....	65
DMATRIX .....	66
ERASE.....	68
MAXICODE .....	69
PDF417 .....	71
AZTEC .....	74
MPDF417.....	76
PUTBMP .....	77
PUTPCX .....	79
QRCODE .....	81
RSS .....	88
REVERSE .....	93
DIAGONAL .....	94
TEXT.....	95
BLOCK .....	99
<b>3. Status Polling and Immediate Commands</b> .....	102
<ESC>!? .....	102
<ESC>!C .....	103
<ESC>!D .....	104
<ESC>!O .....	105
<ESC>!P.....	106
<ESC>!Q .....	107
<ESC>!R .....	108
<ESC>!S .....	109
<ESC>!F .....	112
<ESC>!I .....	113
~!@ .....	114
~!A .....	115
~!C .....	116



## CONTENTS

~!D .....	117
~!E .....	118
~!F .....	119
~!I .....	120
~!T .....	121
<ESC> Y .....	122
<ESC> Z .....	123
<b>4. Message Translation Protocols.....</b>	<b>124</b>
~# .....	124
<b>5. Commands for Windows Driver .....</b>	<b>125</b>
!B .....	125
!J .....	126
!N .....	127
<b>6. File Management Commands .....</b>	<b>128</b>
!DOWNLOAD.....	128
EOP .....	131
FILES.....	132
KILL .....	133
MOVE.....	135
RUN.....	136
<b>7. BASIC Commands and Functions.....</b>	<b>137</b>
ABS( ) .....	137
ASC( ) .....	138
CHR\$( ).....	139
XOR\$( ).....	140
END .....	141
EOF( ) .....	142
OPEN.....	143
CLOSE .....	145
WRITE .....	146
READ.....	147
SEEK.....	149
LOF( ).....	150
LOC( ).....	151
FREAD\$( ) .....	152



## CONTENTS

PUT .....	153
GET .....	154
COPY .....	156
FOR...NEXT LOOP .....	158
WHILE...WEND.....	160
DO...LOOP.....	162
IF...THEN...ELSE...ENDIF LOOP .....	166
GOSUB...RETURN.....	170
GOTO .....	171
INP\$( ) .....	172
INP( ) .....	173
LOB( ) .....	174
INPUT.....	175
PREINPUT .....	176
POSTINPUT .....	177
SET FILTER ON/OFF .....	178
REM .....	179
OUT .....	180
OUTR .....	181
GETKEY( ) .....	182
INT( ) .....	183
LEFT\$( ) .....	184
LEN( ) .....	185
MID\$( ).....	186
RIGHT\$( ).....	187
STR\$( ) .....	188
STRCOMP( ) .....	189
INSTR ( ).....	191
TRIM\$( ) .....	192
LTRIM\$( ) .....	194
RTRIM\$( ) .....	196
TEXTPIXEL( ) .....	198
BARCODEPIXEL( ) .....	199
VAL( ) .....	200
BEEP .....	201



## CONTENTS

NOW\$( ) .....	202
NOW .....	203
FORMAT\$( ) .....	204
DATEADD() .....	209
FSEARCH() .....	212
TOUCHPRESS() .....	213
RECORDSET\$ ( ) .....	214
LABELRATIO.....	216
<b>8. Device Reconfiguration Commands.....</b>	<b>217</b>
SET COUNTER .....	217
SET CUTTER.....	219
SET PARTIAL_CUTTER .....	220
SET BACK .....	222
SET KEYn.....	223
SET LEDn .....	225
SET PEEL .....	226
SET REWIND .....	227
SET TEAR & SET STRIPER .....	228
SET GAP .....	229
SET BLINE.....	231
SET HEAD .....	232
SET RIBBON .....	233
SET ENCODER .....	235
SET RIBBONEND .....	236
SET COM1 .....	237
SET PRINTKEY .....	238
SET REPRINT .....	240
SET FEED_LEN .....	241
GETSENSOR() .....	242
GETSETTING\$() .....	244
SET USBHOST .....	248
SET RS232_REWINDER .....	249
SET AUTORUN .....	250
SET VERIFIER .....	251
SET RESPONSE.....	252



## CONTENTS

SET DAYLIGHT_SAVE .....	254
PEEL .....	255
LED1, LED2, LED3 .....	256
KEY1, KEY2, KEY3 .....	257
<b>9. Printer Global Variables</b> .....	258
@LABEL .....	258
YEAR .....	259
MONTH .....	260
DATE .....	261
WEEK .....	262
HOUR .....	263
MINUTE .....	264
SECOND .....	265
@YEAR .....	266
@MONTH .....	267
@DATE .....	268
@DAY .....	269
@HOUR .....	270
@MINUTE .....	271
@SECOND .....	272
_MODEL\$ .....	273
_SERIAL\$ .....	274
_VERSION\$ .....	275
<b>10. Wi-Fi Module Setting Commands</b> .....	276
WLAN OFF .....	276
WLAN SSID .....	277
WLAN WPA .....	278
WLAN WEP .....	279
WLAN DHCP .....	280
WLAN IP .....	281
WLAN PORT .....	282
<b>11. Internal Ethernet Setting Commands</b> .....	283
NET DHCP .....	283
NET IP .....	284
NET PORT .....	285



## CONTENTS

NET NAME .....	286
<b>12. NFC Setting Commands.....</b>	<b>287</b>
NFC FEATURE.....	287
NFC STATUS.....	288
NFC TIMEOUT .....	289
NFC READ .....	290
NFC WRITE .....	291
NFC MODE .....	292
<b>13. Alpha-2R Setting Commands.....</b>	<b>293</b>
SET PRINTQUALITY .....	293
SET STANDBYTIME .....	294
GETSETTING\$() .....	295
<b>14. GPIO Setting Commands .....</b>	<b>296</b>
SET GPO .....	296
SET GPI .....	298
Copyright Information .....	305



# HOW TO READ

**MPDF417**
**Description**

This command defines a Micro PDF 417 bar code.

The command name

Description of this command

**Syntax**
**MPDF417 x,y,rotate,[Wn,][Hn,][Cn,] "content"**

Parameter	Description
x	Horizontal start position (in dots)
y	Vertical start position (in dots)
rotate	Rotation 0: No rotation 90: Rotate 90 degrees 180: Rotate 180 degrees 270: Rotate 270 degrees
Wn	Optional. Module width in dot. Default is 1.
Hn	Optional. Module height in dot. Default is 10.
Cn	Optional. Number of columns. Once the parameter is set, the printer will calculate the proper rows for the barcode base on the content automatically. 0: Auto mode. 1: Column is 1 and the calculated suitable rows will be 11, 14, 17, 20, 24, and 28. 2: Column is 2 and the calculated suitable rows will be 8, 11, 14, 17, 20, 23 and 26. 3: Column is 3 and the calculated suitable rows will be 6, 8, 10, 12, 15, 20, 26, 32, 38 and 44. 4: Column is 4 and the calculated suitable rows will be 4, 6, 8, 10, 12, 15, 20, 26, 32, 38 and 44.
Content	Content of Micro PDF 417 bar code
<b>Note:</b>	
This command has been supported since V6.61 EZ and later firmware.	

Syntax of this command

The detail description of each parameter

**Example**

Sample Code	Result
<pre>SIZE 4,1 GAP 0,0 CLS MPDF417 10,10,0, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " MPDF417 110,10,0,W2, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " MPDF417 210,10,0,W2,H3, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " MPDF417 310,10,0,W2,H3,C3, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " PRINT 1</pre>	

The example and printout for reference



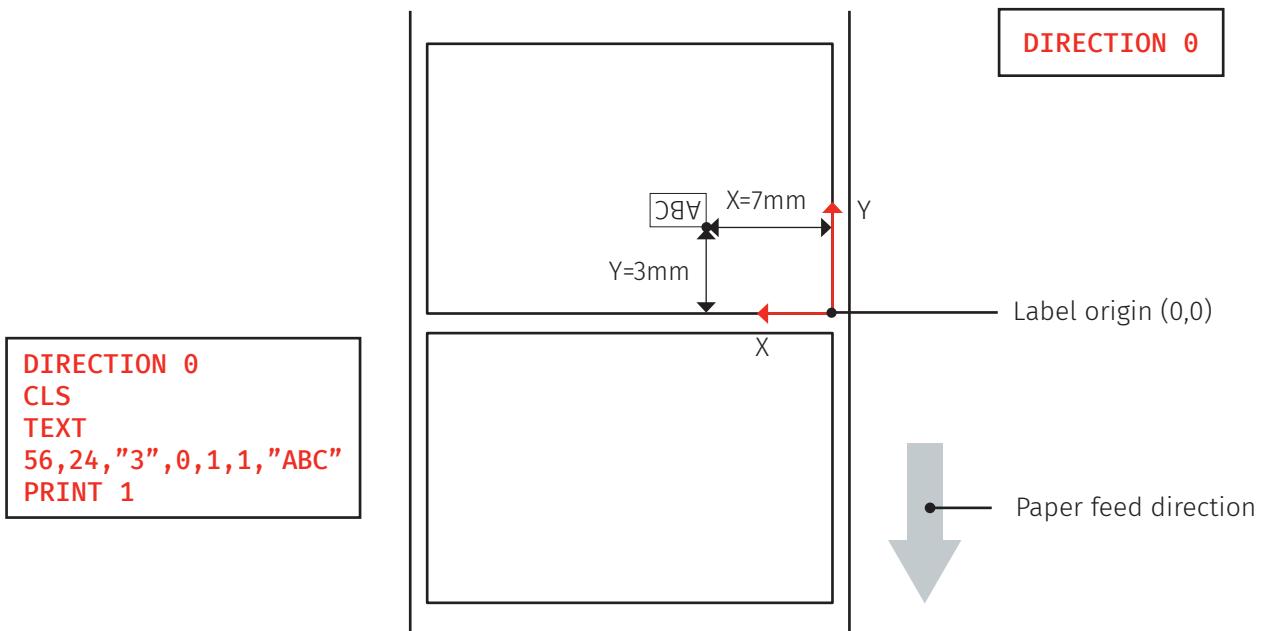
## DOCUMENT CONVENTIONS

This manual uses the following typographic conventions.

Convention	Description
<b>[expression list]</b>	Items inside square brackets are optional, expression maximum length 2*1024 bytes.
<b>&lt;ESC&gt;</b>	ASCII 27, control code of status polling command returns/runs the printer status immediately.
<b>~</b>	ASCII 126, control code of status polling command returns the printer status only when the printer is ready.
<b>Space</b>	ASCII 32, characters will be ignored in the command line.
<b>"</b>	ASCII 34, beginning and ending of expression.
<b>CR, LF</b>	ASCII 13, ASCII 10, denotes end of command line.
<b>NULL</b>	ASCII 0, supported in the expression.
<b>Note:</b>	The font in bold type is used for note.
<b>203 DPI: 1 mm = 8 dots</b>	

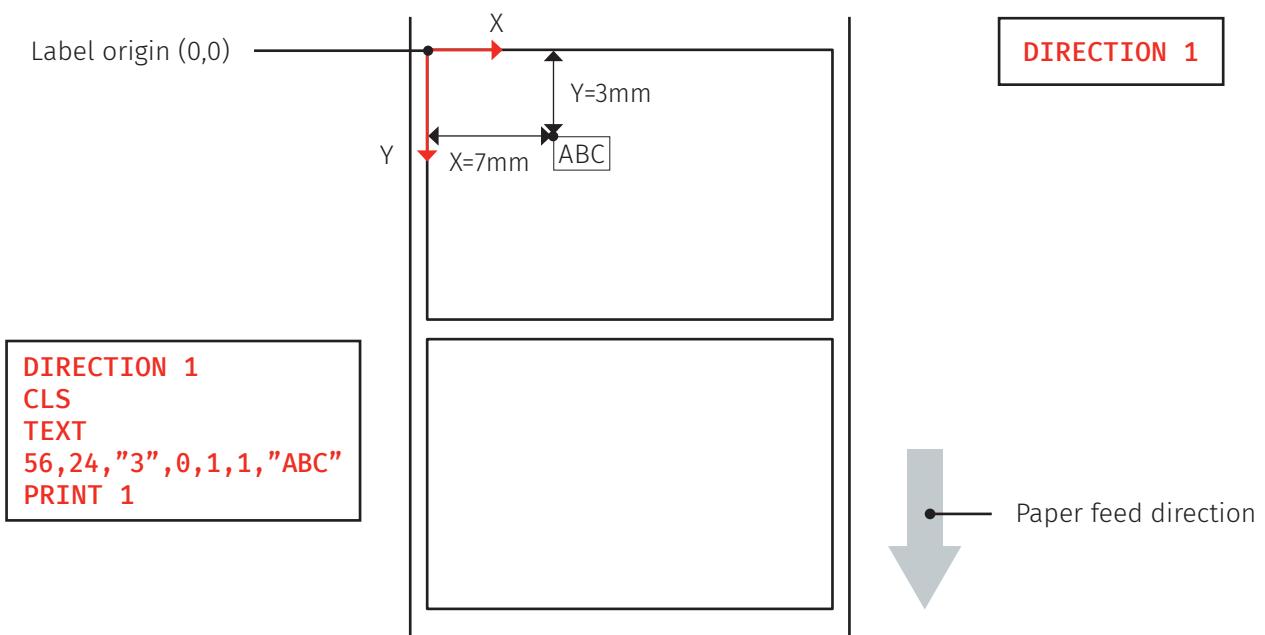


## OBJECT POSITION CALCULATION



**Note:**

- 203 DPI, 1mm=8 dots ; 300 DPI, 1mm=11.8 dots
- Only integer portion will be used. Ex. 2 mm = 23.6 dots then 23 dots will be used.





# 1. SETUP AND SYSTEM COMMANDS

## SIZE

### Description

This command defines the label width and length.

### Syntax

**SIZE m,n** English system (inch)

**SIZE m mm,n mm** Metric system (mm)

**SIZE m dot,n dot** Dot measurement

*This command has been supported since V6.27 EZ and later firmware.*

Parameter	Description
m	Label width (inch/mm/dot)
n	Label length (inch/mm/dot)

**Note :**

- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1 mm = 12 dots
- For metric and dot systems, there must be a space between parameter and “mm” or “dot”.

### Example

Sample Code	Result
<ul style="list-style-type: none"> <li>• English system (inch): <b>SIZE 3.5,3.00</b></li> <li>• Metric system (mm): <b>SIZE 100 mm,100 mm</b></li> </ul>	<p>The diagram illustrates the output of the SIZE command. It shows a single rectangular label labeled "LABEL". A horizontal double-headed arrow above the label indicates its width is "m". A vertical double-headed arrow to the right of the label indicates its height is "n". Above the label, there is a small gap labeled "GAP" between it and the next line of text. The text "m: Label Width" and "n: Label Height" is located at the bottom right of the diagram.</p>

### See Also

GAP, BLINE



## GAP

### Description

Defines the gap distance between two labels.

### Syntax

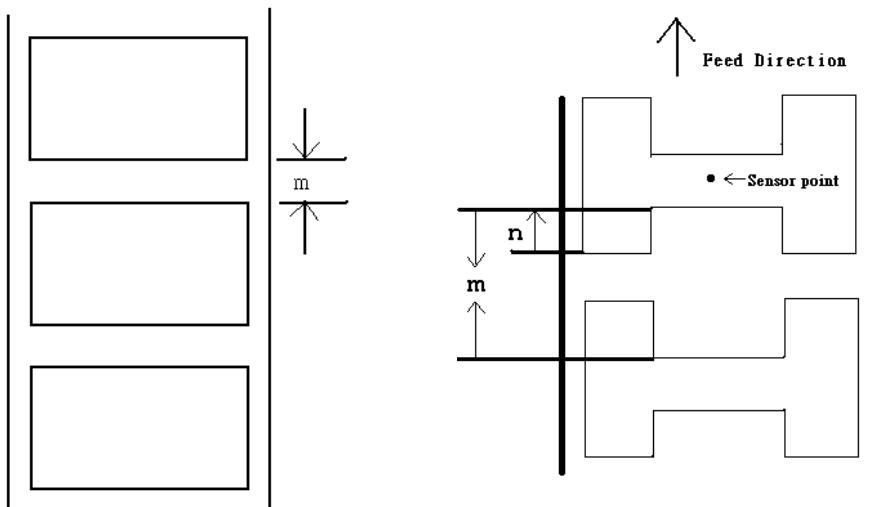
**GAP m,n** English system (inch)

**GAP m mm,n mm** Metric system (mm)

**GAP m dot,n dot** Dot measurement

*This command has been supported since V6.27 EZ and later firmware.*

Parameter	Description
m	Label width (inch/ mm/ dot)
n	Label length (inch/ mm/ dot)
0,0	Continuous label

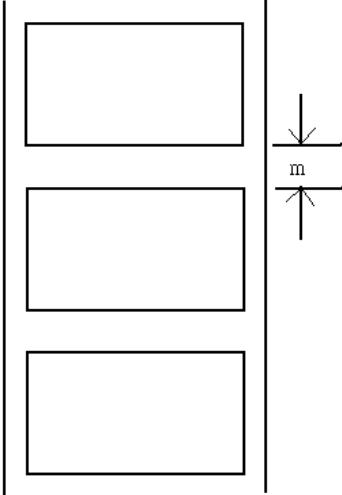
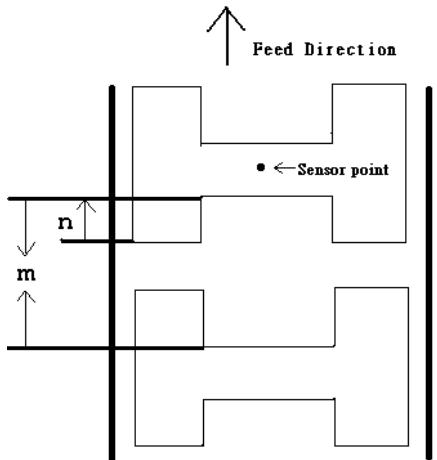


#### Note :

- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1 mm = 12 dots
- For metric and dot systems, there must be a space between parameter and mm.
- When the sensor type is changed from “Black Mark” to “GAP”, please send the “GAP” command to the printer first.



## Example

Sample Code	Result
<b>Normal gap</b> <ul style="list-style-type: none"><li>• English system (inch): <b>GAP 0.12,0</b></li><li>• Metric system (mm): <b>AP 3 mm,0 mm</b></li><li>• Continuous label: <b>GAP 0,0</b></li></ul>	<b>Normal gap</b> 
<b>Special gap</b> <ul style="list-style-type: none"><li>• English system (inch): <b>GAP 0.30,0.10</b></li><li>• Metric system (mm): <b>GAP 7.62 mm,2.54 mm</b></li></ul>	<b>Special gap</b> 

## See Also

[SIZE](#), [BLINE](#)



## GAPDETECT

### Description

This command feeds the paper through the gap sensor in an effort to determine the paper and gap sizes, respectively. This command references the user's approximate measurements. If the measurements conflict with the actual size, the GAPDETECT command will not work properly. This calibration method can be applied to the labels with pre-printed logos or texts.

### Syntax

**GAPDETECT [x,y]**

Parameter	Description
x	Paper length (in dots)
y	Gap length (in dots)

**Note:**

If the x, y parameters are ignored then the printer will calibrate and determine the paper length and gap size automatically.

### See Also

GAP, SIZE, BLINEDETECT, AUTODETECT



## BLINEDETECT

### Description

This command feeds the paper through the black mark sensor in an effort to determine the paper and black mark sizes, respectively. This command references the user's approximate measurements. If the measurements conflict with the actual size, the BLINEDETECT command will not work properly. This calibration method can be applied to the labels with pre-printed logos or texts.

### Syntax

**BLINEDETECT [x,y]**

Parameter	Description
x	Paper length (in dots)
y	Gap length (in dots)

**Note:**

If the x, y parameters are ignored then the printer will calibrate and determine the paper length and gap size automatically.

### See Also

GAP, SIZE, GAPDETECT, AUTODETECT



## AUTODETECT

### Description

This command feeds the paper through the gap/black mark sensor in an effort to determine the paper and gap/black mark sizes, respectively. This command references the user's approximate measurements. If the measurements conflict with the actual size, the AUTODETECT command will not work properly. This calibration method can be applied to the labels with pre-printed logos or texts.

### Syntax

**AUTODETECT [x,y]**

Parameter	Description
x	Paper length (in dots)
y	Gap length (in dots)

**Note:**

- If the x, y parameters are ignored then the printer will calibrate and determine the paper length and gap/black mark size automatically.
- When using this command, the printer will detect the label by the proper sensor type so please don't set the command GAP or BLINE in your program.
- It is supported in firmware V6.86 EZ or later.

### See Also

GAP, SIZE, GAPDETECT, BLINEDECTECT



## BLINE

### Description

This command sets the height of the black line and the user-defined extra label feeding length each form feed takes.

### Syntax

**BLINE m,n**

English system (inch)

**BLINE m mm,n mm**

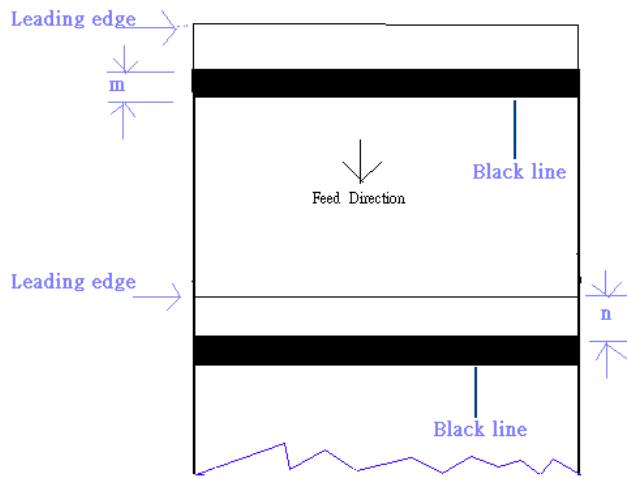
Metric system (mm)

**BLINE m dot,n dot**

Dot measurement

*This command has been supported since V6.27 EZ and later firmware.*

Parameter	Description
m	The height of black line either in inch or mm $0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch), $0 \leq m \leq 127$ (mm) / since V6.21 EZ and later firmware
n	The extra label feeding length $0 \leq n \leq$ label length
0,0	Continuous label



#### Note:

- For metric system, there must be a space between parameter and mm.
- When the sensor type is changed from “GAP” to “Black Mark”, please send the “BLINE” command to the printer first.
- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1 mm = 12 dots



## Example

### Sample Code

- English system (inch):

**BLINE 0.20,0.50**

- Metric system (mm):

**BLINE 5.08 mm,12.7 mm**

## See Also

**SIZE, GAP**



## OFFSET

### Description

This command defines the selective, extra label feeding length each form feed takes, which, especially in peel-off mode and cutter mode, is used to adjust label stop position, so as for label to register at proper places for the intended purposes. The printer back tracks the extra feeding length before the next run of printing.

### Syntax

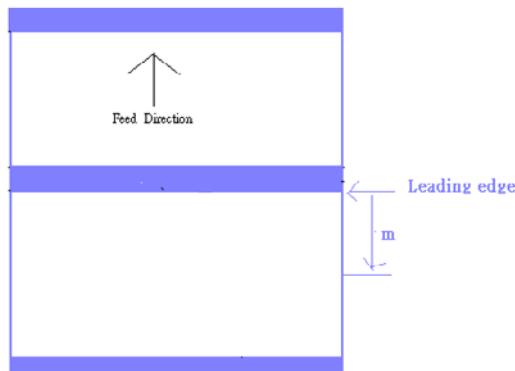
**OFFSET m** English system (inch)

**OFFSET m mm** Metric system (mm)

**OFFSET m dot** Dot measurement

*This command has been supported since V6.27 EZ and later firmware.*

Parameter	Description
m	The offset distance (inch or mm) $-1 \leq m \leq 1$ (inch)




---

#### CAUTION:

- Improperly offset value may cause paper jam.
- For metric system, there must be a space between parameter and mm.
- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1 mm = 12 dots

### Example

#### Sample Code

- English system (inch):  
**OFFSET 0.5**
- Metric system (mm):  
**OFFSET 12.7 mm**

### See Also

**SIZE, GAP, SET PEEL, SET CUTTER**



## SPEED

### Description

This command defines the print speed.

### Syntax

**SPEED n**

Parameter	Description
n	Printing speed in inch per second. Max: 152.4 mm/second (6 ips) Standard: 127 mm/second (5 ips)  PrintEasy: 3 / 3.5 / 4 / 4.5 / 5 / 5.5 / 6 PrintHybrid: 1.5 / 2 / 2.5 / 3 / 3.5 / 4 / 4.5 / 5 / 5.5 / 6

### Example

#### Sample Code

**SPEED 5**

### See Also

DENSITY



## DENSITY

### Description

This command sets the printing darkness.

### Syntax

**DENSITY n**

Parameter	Description
n	0~15 0: specifies the lightest level 15: specifies the darkest level

**Note:**

Default DENSITY setting is 8.

### Example

**Sample code**

**DENSITY 7**



## DIRECTION AND MIRROR IMAGE

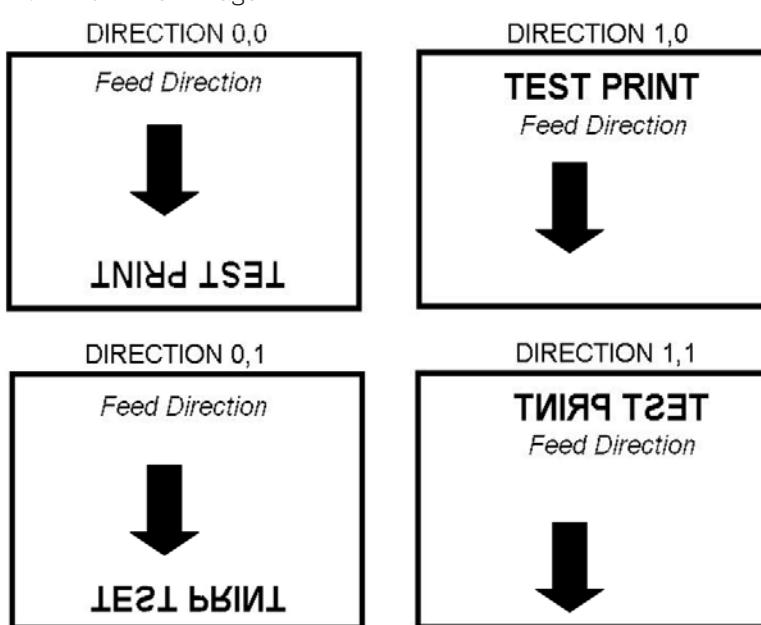
### Description

This command defines the printout direction and mirror image. This will be stored in the printer memory.

### Syntax

**DIRECTION n[,m]**

Parameter	Description
n	0 or 1. Please refer to the illustrations below
m	0: Print normal image 1: Print mirror image



### Example

#### Sample code

- DIRECTION 0
- DIRECTION 0,1

### See Also

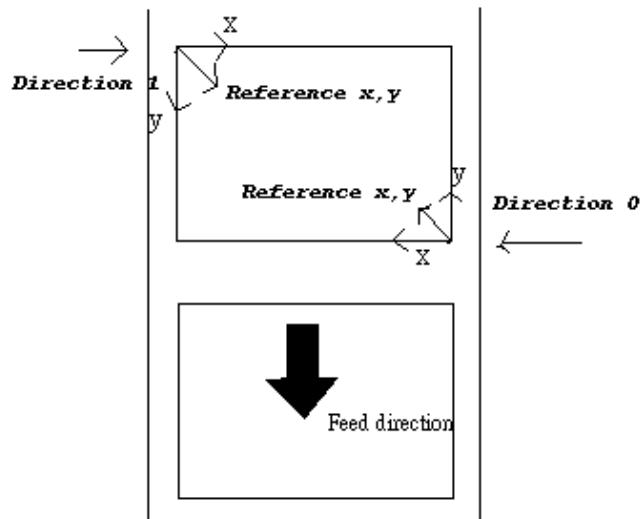
REFERENCE



## REFERENCE

### Description

This command defines the reference point of the label. The reference (origin) point varies with the print direction, as shown:



### Syntax

**REFERENCE x,y**

Parameter	Description
x	Horizontal coordinate (in dots)
y	Vertical coordinate (in dots)

**Note:**

200 DPI: 1 mm = 8 dots

300 DPI: 1 mm = 12 dots

### Example

**Sample code**

```
REFERENCE 10,10
```

### See Also

DIRECTION



## SHIFT

### Description

This command moves the label's horizontal and vertical position. A positive value moves the label further from the printing direction; a negative value moves the label towards the printing direction.

### Syntax

**SHIFT [x,] y**

Parameter	Description
x	Optional. The maximum value is 1 inch. For 200 dpi printers, the range is -203 to 203; The unit is dot.
y	The maximum value is 1 inch. For 200 dpi printers, the range is -203 to 203; The unit is dot.

### Example

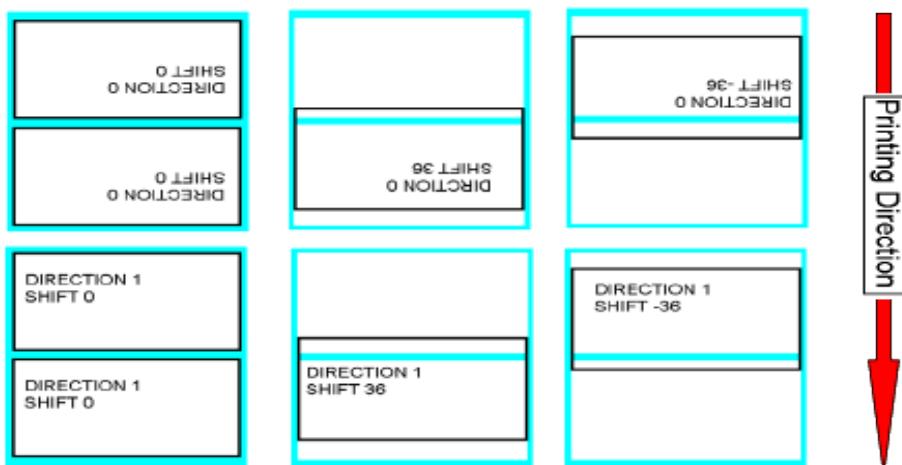
#### Sample Code

```

SIZE 4,2.5
GAP 2 mm,0
DIRECTION 0
SHIFT 36
OFFSET 0
CLS
TEXT 400,200, "3",0,1,1, "DIRECTION 0"
TEXT 400,250, "3",0,1,1, "SHIFT 36"
BOX 10,0,780,490,8
PRINT 3,1

```

#### Result



### See Also

OFFSET, REFERENCE



## COUNTRY

### Description

This command orients the keyboard for use in different countries via defining special characters on the KP-200 series portable LCD keyboard (option).

### Syntax

**COUNTRY n**

Parameter	Description
n	001: USA 002: Canadian-French 003: Spanish (Latin America) 031: Dutch 032: Belgian 033: French (France) 034: Spanish (Spain) 036: Hungarian 038: Yugoslavian 039: Italian 041: Switzerland 042: Slovak 044: United Kingdom 045: Danish 046: Swedish 047: Norwegian 048: Polish 049: German 055: Brazil 061: English (International) 351: Portuguese 358: Finnish

### Example

#### Sample Code

**COUNTRY 001**

### See Also

**CODEPAGE, ~!I**



## CODEPAGE

### Description

This command defines the code page of international character set.

### Syntax

**CODEPAGE n**

Parameter		Description	
		7-bit code page	8-bit code page
n	Name	n	Name
<b>USA</b>	USA	<b>437</b>	United States
<b>BRI</b>	British	<b>737</b>	Greek
<b>GER</b>	German	<b>850</b>	Multilingual
<b>FRE</b>	French	<b>851</b>	Greek 1
<b>DAN</b>	Danish	<b>852</b>	Slavic
<b>ITA</b>	Italian	<b>855</b>	Cyrillic
<b>SPA</b>	Spanish	<b>857</b>	Turkish
<b>SWE</b>	Swedish	<b>860</b>	Portuguese
<b>SWI</b>	Swiss	<b>861</b>	Icelandic
		<b>862</b>	Hebrew
		<b>863</b>	Canadian/French
		<b>864</b>	Arabic
		<b>865</b>	Nordic
		<b>866</b>	Russian
		<b>869</b>	Greek 2
		Windows code page	ISO code page
n	Name	n	Name
<b>1250</b>	Central Europe	<b>8859-1</b>	Latin 1
<b>1251</b>	Cyrillic	<b>8859-2</b>	Latin 2
<b>1252</b>	Latin I	<b>8859-3</b>	Latin 3
<b>1253</b>	Greek	<b>8859-4</b>	Baltic
<b>1254</b>	Turkish	<b>8859-5</b>	Cyrillic
<b>1255</b>	Hebrew	<b>8859-6</b>	Arabic
<b>1256</b>	Arabic	<b>8859-7</b>	Greek
<b>1257</b>	Baltic	<b>8859-8</b>	Hebrew
<b>1258</b>	Vietnam	<b>8859-9</b>	Turkish
<b>932</b>	Japanese Shift-JIS	<b>8859-10</b>	Latin 6
<b>936</b>	Simplified Chinese GBK	<b>8859-15</b>	Latin 9
<b>949</b>	Korean		
<b>950</b>	Traditional Chinese Big5		
<b>UTF-8</b>	UTF 8		

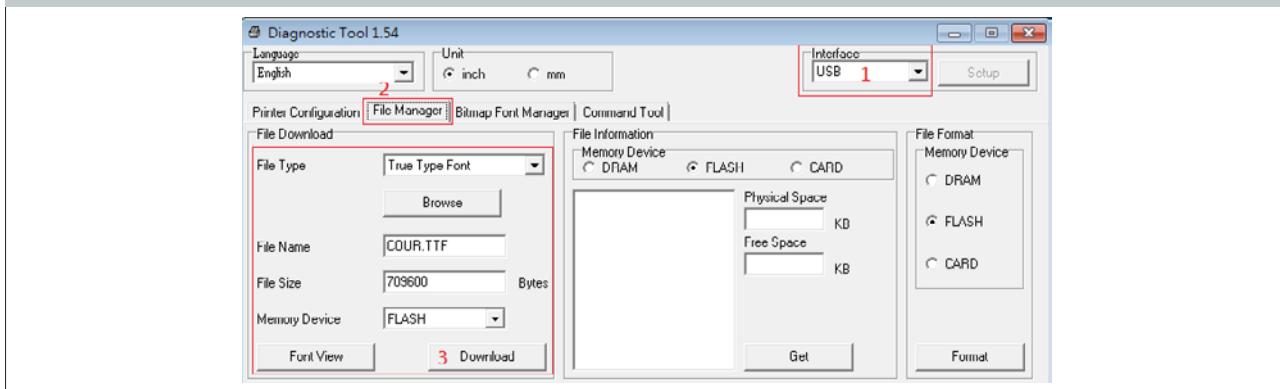
**Note:**

**DATA LENGTH** determines 7-bit or 8-bit communications parameter.



## Example

**Download the COUR.TTF into printer by DiagTool**



# Sample Code

```
DOWNLOAD "TEST.BAS"
str1$ = " "
J = 0
y = 50

CODEPAGE 1252
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10,"COUR.TTF",0,12,12,"CODEPAGE 1252"
FOR I=32 TO 255
str1$=str1$+CHR$(I) + " "
J=J+1
IF J=16 THEN GOSUB drawTEXT
NEXT

PRINT 1
END

drawTEXT:
TEXT 10,y,"COUR.TTF",0,12,12,str1$
str1$=" "
J=0
y=y+40
RETURN
EOP
TEST
```

## Result

CODEPAGE 1252  
 ! " # \$ % & ' ( ) \* + , - . /  
 0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
 @ A B C D E F G H I J K L M N O  
 P Q R S T U V W X Y Z [ \ ] ^ \_  
 ` a b c d e f g h i j k l m n o  
 p q r s t u v w x y z { | } ~  
 € , f „ … † ‡ ^ ‰ Š < © Ž  
 ‘ ’ “ ” • — ~ ™ š > œ ž Ÿ  
 ; ¢ £ ¤ ¥ ¡ § ¨ © ª « ¬ — ®  
 ° ± ² ³ ´ μ ¶ · ¸ º » ¼ ½ ¾ ¸  
 À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï  
 Đ Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß  
 à á â ã ä å æ ç è é ê ë ì í î ï  
 õ ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ

## See Also

COUNTRY, ~!I



## CLS

### Description

This command clears the image buffer.

### Syntax

**CLS**

Parameter	Description
none	N/A

**Note:**

This command must be placed after SIZE command.

### Example

Sample code
<b>CLS</b>

### See Also

SIZE, GAP, BLINE



## FEED

### Description

This command feeds label with the specified length. The length is specified by dot.

### Syntax

**FEED n**

Parameter	Description
n	unit: dot $1 \leq n \leq 9999$

**Note:**

200 DPI: 1 mm = 8 dots

300 DPI: 1 mm = 12 dots

### Example

Sample code

```
FEED 40
```

### See Also

BACKFEED, SIZE, GAP, BLINE, HOME, FORMFEED



## BACKFEED & BACKUP

### Description

This command feeds the label in reverse. The length is specified by dot.

### Syntax

<b>BACKUP n</b>	TSPL printers only
<b>BACKFEED n</b>	TSPL2 printers only

Parameter	Description
n	unit: dot 1 ≤ n ≤ 9999

**Note:**

**200 DPI: 1 mm = 8 dots**

**300 DPI: 1 mm = 12 dots**

**CAUTION:**

Improperly back feed value may cause paper jam or wrinkle.

### Example

**Sample code**

- TSPL printers  
**BACKUP 40**
- TSPL2 printers  
**BACKFEED 40**

### See Also

**FEED, SIZE, GAP, BLINE, HOME, FORMFEED**



## FORMFEED

### Description

This command feeds label to the beginning of next label.

### Syntax

#### FORMFEED

Parameter	Description
none	N/A

#### Note:

This command must be placed after SIZE command.

### Example

Sample code	Result
<pre>SIZE 4,2.5 GAP 2 mm,0 DIRECTION 1 FORMFEED CLS TEXT 25,25, "3",0,1,1, "FORMFEED COMMAND TEST" PRINT 1,1</pre>	<p>The diagram illustrates the effect of the FORMFEED command. On the left, the sample code is shown, with the 'FORMFEED' command highlighted by a red box. On the right, the resulting printout is shown. The printout consists of a blue rectangular frame containing the text 'FORMFEED COMMAND TEST'. Below this frame, the number '3' is printed. An arrow labeled 'Paper feed direction' points downwards, indicating that the label has been moved to the top of the page by the FORMFEED command.</p>

### See Also

FEED, SIZE, GAP, BLINE, HOME, BACKFEED



## HOME

### Description

This command will feed label until the internal sensor has determined the origin. Size and gap of the label should be defined before using this command.

### Syntax

#### HOME

Parameter	Description
none	N/A

For TSPL programming printer: Back label to origin position

For TSPL2 programming printer: Feed label to origin position

### Example

#### Sample code

```
SIZE 4,2.5
GAP 2 mm,0
SET COUNTER @0 +1
@0="000001"
HOME
CLS
BOX 1,1,360,65,12
TEXT 25,25, "3",0,1,1, "HOME COMMAND TEST"
TEXT 25,80, "3",0,1,1,@0
PRINT 3,1
```

### See Also

FEED, SIZE, GAP, BLINE, FORMFEED



## PRINT

### Description

This command prints the label format currently stored in the image buffer.

### Syntax

**PRINT m[,n]**

Parameter	Description
m	Specifies how many sets of labels will be printed. $1 \leq m \leq 999999999$
n	Specifies how many copies should be printed for each particular label set. $1 \leq n \leq 999999999$

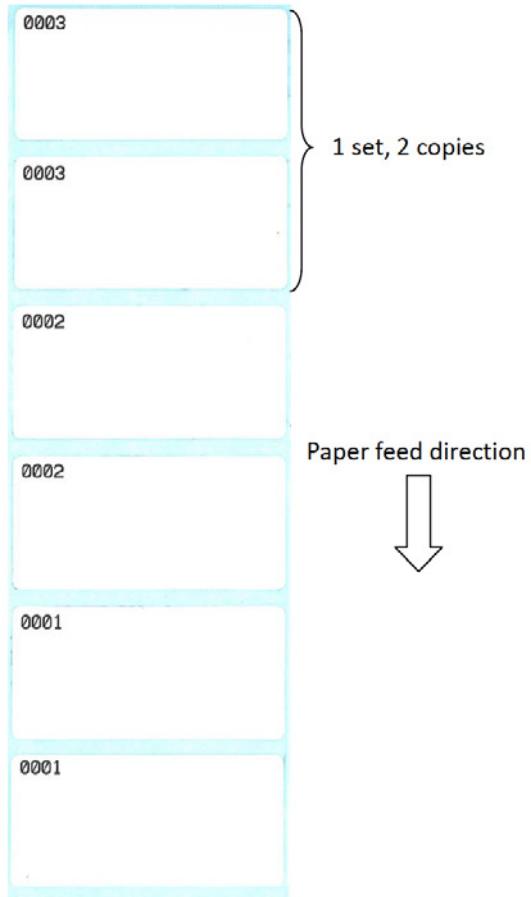
### Example

#### Sample code

```

SIZE 50 mm,25 mm
GAP 3 mm,0
DIRECTION 1
SET COUNTER @1 1
@1="0001"
CLS
TEXT 10,10, "3",0,1,1,@1
PRINT 3,2

```



### See Also

SET COUNTER, INPUT, DOWNLOAD



## SOUND

### Description

This command controls the sound frequency of the beeper. There are 10 levels of sounds. The timing control can be set by the “interval” parameter.

### Syntax

**SOUND level,interval**

Parameter	Description
level	Sound level: 0~9
interval	Sound interval: 1~4095

### Example

Sample code
<ul style="list-style-type: none"><li>• SOUND 5,200</li><li>• SOUND 3,200</li><li>• SOUND 3,200</li><li>• SOUND 4,200</li><li>• SOUND 2,200</li><li>• SOUND 2,200</li><li>• SOUND 1,200</li><li>• SOUND 2,200</li><li>• SOUND 3,200</li><li>• SOUND 4,200</li><li>• SOUND 5,200</li></ul>



## CUT

### Description

This command activates the cutter to immediately cut the labels without back feeding the label.

### Syntax

**CUT**

Parameter	Description
none	N/A

### Example

Sample code
<b>SIZE 3,3</b> <b>GAP 0,0</b> <b>CLS</b> <b>BOX 0,0,866,866,5</b> <b>TEXT 100,100, "5",0,1,1, "FEED &amp; CUT"</b> <b>TEXT 100,200, "5",0,1,1, "300 DPI"</b> <b>PRINT 1,1</b> <b>FEED 260</b> <b>CUT</b>

### See Also

[SET CUTTER](#), [SET BACK](#), [SET PARTITAL\\_CUTTER](#)



## LIMITFEED

### Description

If the gap sensor is not set to a suitable sensitivity while feeding labels, the printer will not be able to locate the correct position of the gap. This command stops label feeding and makes the red LED flash if the printer does not locate gap after feeding the length of one label plus one preset value.

### Syntax

<b>LIMITFEED n[,minpaper,maxgap]</b>	English system (inch)
<b>LIMITFEED n mm[,minpaper mm,maxgap mm]</b>	Metric system (mm)
<b>LIMITFEED n dot[,minpaper dot,maxgap dot]</b>	Dot measurement
<i>This command has been supported since V6.34 EZ.</i>	

Parameter	Description
n	The maximum length for sensor detecting
minpaper	The minimum length of paper
maxgap	The maximum length of gap

#### Note:

- The setting will remain resident in memory.
- For metric system, there must be a space between parameter n and mm.
- The default value is 10 inches when printer initializes.
- Since V6.76 EZ, the default value for TDP-225 series printer is 14 inches when printer initializes.
- The setting of parameters “minpaper” and “maxgap” are using for calibrating the preprinted label.  
This parameter has been supported since V6.98.7 EZ.

### Example

#### Sample code

- English system (inch)  
**LIMITFEED 12**



## SELFTEST

### Description

At this command, the printer will print out the printer information.

### Syntax

**SELFTEST** [page]

Parameter	Description
page	<b>omitted</b> : Print a self-test page with whole printer information. <b>PATTERN</b> : Print a pattern to check the status of print head heat line. <b>ETHERNET</b> : Print a self-test page with Ethernet settings. <b>WLAN</b> : Print a self-test page with Wi-Fi settings. <b>RS232</b> : Print a self-test page with RS-232 settings. <b>SYSTEM</b> : Print a self-test page with printer settings. <b>Z</b> : Print a self-test page with emulated language settings.

### Example

Sample code	Result
<b>SELFTEST</b>	<pre> ----- SYSTEM INFORMATION ----- MODEL: TDP247 FIRMWARE: 7.00 EZ CHECKSUM: 07CB0355 S/N: D452350388 TCF: NO DATE: 1970/01/01 TIME: 00:04:18 NON-RESET: 110 m {TPH} RESET: 110 m {TPH} NON-RESET: 0 {CUT} RESET: 0 {CUT}  ----- PRINTING SETTING ----- SPEED: 5 IPS DENSITY: 8.0 WIDTH: 4.00 INCH HEIGHT: 4.00 INCH GAP: 0.00 INCH INTENSION: 5 CODEPAGE: 850 COUNTRY: 001  ----- Z SETTING ----- DARKNESS: 16.0 SPEED: 4 IPS WIDTH: 4.00 INCH TILDE: 7EH (^) CARET: 5EH {^} DELIMITER: 2CH {{}} POWER UP: NO MOTION HEAD CLOSE: NO MOTION  ----- RS232 SETTING ----- BAUD: 9600 PARITY: NONE DATA BIT: 8 STOP BIT: 1  ----- DRAM FILE (0 FILES) ----- PHYSICAL 8192 KBYTES AVAILABLE 256 KBYTES  ----- FLASH FILE (0 FILES) ----- PHYSICAL 4096 KBYTES AVAILABLE 2560 KBYTES  ----- PATTERN ----- </pre>



Sample code	Result
SELFTEST PATTERN	
SELFTEST ETHERNET	<pre>----- ETHERNET SETTING ----- NAME: PS-FF02FD MAC ADDR: 001B82-FF02FD DHCP: ON IP ADDR: 10.0.10.115 SUBNET: 255.255.255.0 GATEWAY: 10.0.10.252 PORT: 9100 -----</pre>
SELFTEST WLAN	<pre>----- WLAN SETTING ----- MAC ADDR: 001DC9-908397 MODE: AD-HOC SSID: TEST-AP IP ADDR: 192.168.1.3 SUBNET: 255.255.255.0 GATEWAY: 192.168.1.1 PORT: 9100 -----</pre>
SELFTEST RS232	<pre>----- RS232 SETTING ----- BAUD: 9600 PARITY: NONE DATA BIT: 8 STOP BIT: 1 -----</pre>
SELFTEST SYSTEM	<pre>----- SYSTEM INFORMATION ----- MODEL: TDP247 FIRMWARE: 7.00 EZ CHECKSUM: 07CBD355 S/N: D452350388 TCF: NO  DATE: 2013/01/11 TIME: 14:57:55 NON-RESET: 145 m (TPH) RESET: 145 m (TPH) NON-RESET: 0 (CUT) RESET: 0 (CUT) -----</pre>
SELFTEST PRINTER	<pre>----- PRINTING SETTING ----- SPEED: 5 IPS DENSITY: 8.0 WIDTH: 4.00 INCH HEIGHT: 1.00 INCH GAP: 0.00 INCH  INTENSION: 5 CODEPAGE: 850 COUNTRY: 001 -----</pre>
SELFTEST Z	<pre>----- Z SETTING ----- DARKNESS: 16.0 SPEED: 4 IPS WIDTH: 4.00 INCH TILDE: 7EH (~) CARET: 5EH (^)  DELIMITER: 2CH (,) POWER UP: NO MOTION HEAD CLOSE: NO MOTION -----</pre>



## EOJ

### Description

Let the printer wait until process of commands (before EOJ) be finished then go on the next command.

### Syntax

**EOJ**

**Note:**

This command has been supported since V6.39 EZ and later firmware.

### Example

#### Sample code

```
SIZE 4,0.2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10,"3",0,1,1,"Two labels are printed without stop."
PRINT 1
PRINT 1

SIZE 4,0.2
GAP 0,0
CLS
TEXT 10,10,"3",0,1,1,"Printer stops before next printing."
PRINT 1
EOJ
PRINT 1
```

#### Result

Paper feed direction



Printer stops before next printing.  
Printer stops before next printing.  
Two labels are printed without stop.  
Two labels are printed without stop. } without stop



## DELAY

### Description

Let the printer wait specific period of time then go on next command.

### Syntax

**DELAY ms**

Parameter	Description
ms	The specific period of time. Unit is millisecond. 1000 ms = 1 second.

**Note:**

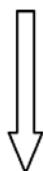
This command has been supported since V6.34 EZ and later firmware.

### Example

**Sample code**

```
SIZE 4,0.7
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10,"3",0,1,1,"The delay time between two labels is 3 seconds."
TEXT 10,60,"3",0,1,1,"Now second:" +@SECOND
PRINT 1
DELAY 3000
PRINT 1
```

**Result**



```
The delay time between two labels is 3 seconds.
Now second:9

The delay time between two labels is 3 seconds.
Now second:6
```



## DISPLAY

### Description

This command can show the image, which is in printer's image buffer, on LCD panel.

### Syntax

```
DISPLAY IMAGE/OFF/CLS/forecolor,backcolor/x,y,width,height/x,y,
"bmpfile"/x,y, "font","content"
```

Parameter	Description
IMAGE	Show the image in printer's image buffer on LCD panel. (since V6.39 EZ)
OFF	Disable this function. (since V6.39 EZ)
CLS	Show the background color and clear the items in printer's image buffer on LCD panel (since A1.90 EZ)
forecolor,backcolor	Set the color (decimal) for item and background in printer's image buffer on LCD panel (since A1.90 EZ)
x,y,width,height	Draw the bar in printer's image buffer on LCD panel (since A1.90 EZ)
x,y,"bmpfile"	Show the .bmp in printer's image buffer on LCD panel (since A1.90 EZ)
x,y,"font","content"	Show the text in printer's image buffer on LCD panel (since A1.90 EZ)

forecolor	RGB color code for text or bar (decimal)
backcolor	RGB color code for background (decimal)
x	Horizontal multiplication
y	Vertical multiplication
width	frame width
height	frame height
bmpfile	.bmp file name
font	Font name
content	Content of text string

#### Note:

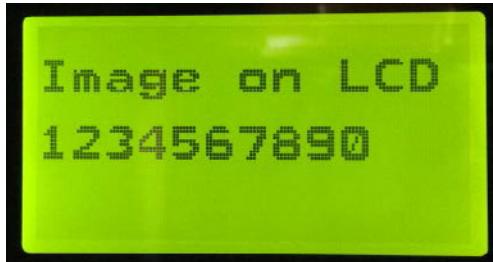
This command only can be performed on the printer with LCD display.



## Example

### Sample code

```
CLS  
TEXT 1,10, "1",0,1,1, "Image on LCD"  
TEXT 1,30, "1",0,1,1, "1234567890"  
DISPLAY IMAGE  
DELAY 5000  
DISPLAY OFF
```



```
CLS  
DISPLAY 15128749,16711680  
DISPLAY CLS  
DISPLAY 10,30, "1","1234567890"  
DELAY 5000  
DISPLAY OFF
```





## INITIALPRINTER

### Description

This command can restore printer settings to defaults.

### Syntax

**INITIALPRINTER**

Parameter	Description
none	N/A

### Example

#### Sample code

**INITIALPRINTER**



## MENU

### Description

This command can design user's own menu with a database resident on the printer.

### Syntax

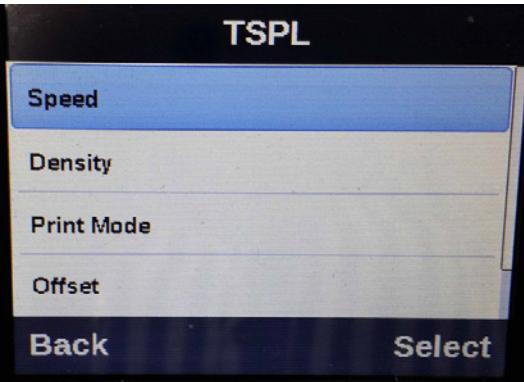
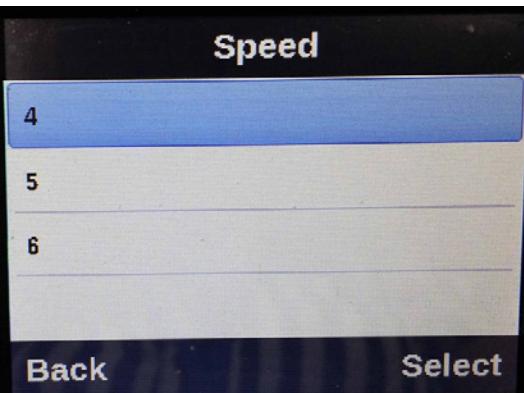
**MENU title\$, list\$, selected**

Parameter	Description
title\$	The title string is shown on LCD screen.
list\$	List of items, separated by CRLF.
selected	It must be a variable to get the result of selection. When selected is 0, the operator has hit ESC (USB keyboard) or MENU button.

**Note:**

- This command only can be performed on the printer with LCD display.
- This command has been supported since VA1.97 and later firmware.

### Example

Sample code	Result
<pre>DOWNLOAD F, "TSPL" Speed Density Print Mode Offset Country EOP  DOWNLOAD F, "Speed" 4 5 6 EOP  DOWNLOAD F, "Density" 6 7 8 9 10 11 12 EOP</pre>	 



Sample code	Result
<pre>DOWNLOAD F,"Print Mode" NONE TEAR OFF PEEL OFF CUT OFF EOP  DOWNLOAD F,"Country" 007 031 033 034 045 EOP  DOWNLOAD F,"DEMO.BAS"  DPI = VAL(GETSETTING\$("SYSTEM","INFORMATION","DPI"))  :MAINLOOP OPEN "TSPL",0 LIST\$ = FREAD\$(0, LOF("TSPL")) CLOSE 0 MENU "TSPL", LIST\$, OPTION\$  IF LEN(OPTION\$) = 0 THEN END  IF OPTION\$ = "Speed"      THEN SETTING\$ = GETSETTING\$("CONFIG","TSPL","SPEED") IF OPTION\$ = "Density"    THEN SETTING\$ = GETSETTING\$("CONFIG","TSPL","DENSITY") IF OPTION\$ = "Print Mode" THEN SETTING\$ = GETSETTING\$("CONFIG","TSPL","PRINT MODE") IF OPTION\$ = "Offset"     THEN SETTING\$ = GETSETTING\$("CONFIG","TSPL","OFFSET") IF OPTION\$ = "Country"    THEN SETTING\$ = GETSETTING\$("CONFIG","TSPL","COUNTRY CODE")</pre>	



Sample code	Result
<pre>IF LOF(OPTION\$) &lt;&gt; 0 THEN OPEN OPTION\$,0 LIST\$ = FREAD\$(0, LOF(OPTION\$)) CLOSE 0 MENU OPTION\$, LIST\$, SETTING\$ ELSE IF OPTION\$ = "Offset" THEN INPUT "Offset", SETTING\$ ENDIF  IF LEN(SETTING\$) &lt;&gt; 0 THEN     IF OPTION\$ = "Speed"      THEN SPEED VAL(SETTING\$)     IF OPTION\$ = "Density"    THEN DENSITY VAL(SETTING\$)     IF OPTION\$ = "Print Mode" THEN GOSUB SET_PRINT_MODE     IF OPTION\$ = "Offset"     THEN OFFSET VAL(SETTING\$) / DPI     IF OPTION\$ = "Country"   THEN GOSUB SET_COUNTRY ENDIF  GOTO MAINLOOP  :SET_PRINT_MODE IF SETTING\$ = "NONE"      THEN SET TEAR OFF IF SETTING\$ = "TEAR OFF"   THEN SET TEAR ON IF SETTING\$ = "PEEL OFF"   THEN SET PEEL ON IF SETTING\$ = "CUT OFF"    THEN SET CUTTER ON RETURN  :SET_COUNTRY IF SETTING\$ = "007" THEN COUNTRY 007 IF SETTING\$ = "031" THEN COUNTRY 031 IF SETTING\$ = "033" THEN COUNTRY 033 IF SETTING\$ = "034" THEN COUNTRY 034 IF SETTING\$ = "045" THEN COUNTRY 045 RETURN  EOP  RUN "DEMO.BAS"</pre>	



## 2. LABEL FORMATTING COMMANDS

### BAR

#### Description

This command draws a bar on the label format.

#### Syntax

**BAR x,y,width,height**

Parameter	Description
x	The upper left corner x-coordinate (in dots)
y	The upper left corner y-coordinate (in dots)
width	Bar width (in dots)
height	Bar height (in dots)

#### Note :

- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1 mm = 12 dots
- Recommended max. bar height is 12 mm at 4" width. Bar height over 12 mm may damage the power supply and affect the print quality.
- Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20 % and 30 % respectively.

#### Example

Sample Code	Result
<pre> SIZE 50 mm,25 mm GAP 3 mm,0 DIRECTION 1 CLS BAR 80,80,300,100 PRINT 1,1 </pre>	

#### See Also

BOX



## BARCODE

### Description

This command prints 1D barcodes. The available barcodes are listed below:

Code Type	Description	Narrow : Width					Max. data length
		1:1	1:2	1:3	2:5	3:7	
<b>128</b>	Code 128, switching code subset automatically.	V					
<b>128M</b>	Code 128, switching code subset manually.	V					
<b>EAN128</b>	EAN128, switching code subset automatically.	V					
<b>EAN128M</b>	EAN128M, switching code subset automatically.	V					
<b>25</b>	Interleaved 2 of 5.		V	V	V		Length is even
<b>25C</b>	Interleaved 2 of 5 with check digit.		V	V	V		Length is odd
<b>25S</b>	Standard 2 of 5.		V	V	V		
<b>25I</b>	Industrial 2 of 5.		V	V	V		
<b>39</b>	Code 39, switching standard and full ASCII mode automatically.	V	V	V			
<b>39C</b>	Code 39 with check digit.		V	V	V		
<b>93</b>	Code 93.			V			
<b>EAN13</b>	EAN 13.	V					12
<b>EAN13+2</b>	EAN 13 with 2 digits add-on.	V					14
<b>EAN13+5</b>	EAN 13 with 5 digits add-on.	V					17
<b>EAN8</b>	EAN 8.	V					7
<b>EAN8+2</b>	EAN 8 with 2 digits add-on.	V					9
<b>EAN8+5</b>	EAN 8 with 5 digits add-on.	V					12
<b>CODA</b>	Codabar.		V	V	V		
<b>POST</b>	Postnet.	V					5, 9, 11
<b>UPCA</b>	UPC-A.	V					11
<b>UPCA+2</b>	UPC-A with 2 digits add-on.	V					13
<b>UPA+5</b>	UPC-A with 5 digits add-on.	V					16
<b>UPCE</b>	UPC-E.	V					6
<b>UPCE+2</b>	UPC-E with 2 digits add-on.	V					8
<b>UPE+5</b>	UPC-E with 5 digits add-on.	V					11
<b>MSI</b>	MSI.		V	V	V		
<b>MSIC</b>	MSI with check digit.		V	V	V		
<b>PLESSEY</b>	PLESSEY.		V	V	V		
<b>CPOST</b>	China post.					V	
<b>ITF14</b>	ITF14.		V	V	V		13
<b>EAN14</b>	EAN14.	V					13
<b>11</b>	Code 11.		V	V	V		
<b>TELEPEN</b>	Telepen. *Since V6.89EZ.		V	V	V		
<b>TELEPENN</b>	Telepen number. *Since V6.89EZ.		V	V	V		
<b>PLANET</b>	Planet. *Since V6.89EZ.	V					
<b>CODE49</b>	Code 49. *Since V6.89EZ.	V					
<b>DPI</b>	Deutsche Post Identcode. *Since V6.91EZ.		V	V	V		11
<b>DPL</b>	Deutsche Post Leitcode. *Since V6.91EZ.		V	V	V		13
<b>LOGMARS</b>	A special use of Code 39. *Since V6.88EZ.		V	V	V		



## Syntax

**BARCODE X,Y, "code type",height,human readable,rotation,narrow,wide,[alignment,]  
"content"**

Parameter	Description																																																
x	Specify the x-coordinate bar code on the label																																																
y	Specify the y-coordinate bar code on the label																																																
code type	<table> <tr> <td>128</td><td>Code 128, switching code subset A, B, C automatically</td></tr> <tr> <td>128M</td><td>Code 128, switching code subset A, B, C manually</td></tr> </table> <table border="1"> <thead> <tr> <th>Control code</th><th>A</th><th>B</th><th>C</th></tr> </thead> <tbody> <tr> <td>096</td><td>FNC3</td><td>FNC3</td><td>NONE</td></tr> <tr> <td>097</td><td>FNC2</td><td>FNC2</td><td>NONE</td></tr> <tr> <td>098</td><td>SHIFT</td><td>SHIFT</td><td>NONE</td></tr> <tr> <td>099</td><td>CODE C</td><td>CODE C</td><td>NONE</td></tr> <tr> <td>100</td><td>CODE B</td><td>FNC4</td><td>CODE B</td></tr> <tr> <td>101</td><td>FNC4</td><td>CODE A</td><td>CODE A</td></tr> <tr> <td>102</td><td>FNC1</td><td>FNC1</td><td>FNC1</td></tr> <tr> <td>103</td><td>Start (CODE A)</td><td></td><td></td></tr> <tr> <td>104</td><td>Start (CODE B)</td><td></td><td></td></tr> <tr> <td>105</td><td>Start (CODE C)</td><td></td><td></td></tr> </tbody> </table>	128	Code 128, switching code subset A, B, C automatically	128M	Code 128, switching code subset A, B, C manually	Control code	A	B	C	096	FNC3	FNC3	NONE	097	FNC2	FNC2	NONE	098	SHIFT	SHIFT	NONE	099	CODE C	CODE C	NONE	100	CODE B	FNC4	CODE B	101	FNC4	CODE A	CODE A	102	FNC1	FNC1	FNC1	103	Start (CODE A)			104	Start (CODE B)			105	Start (CODE C)		
128	Code 128, switching code subset A, B, C automatically																																																
128M	Code 128, switching code subset A, B, C manually																																																
Control code	A	B	C																																														
096	FNC3	FNC3	NONE																																														
097	FNC2	FNC2	NONE																																														
098	SHIFT	SHIFT	NONE																																														
099	CODE C	CODE C	NONE																																														
100	CODE B	FNC4	CODE B																																														
101	FNC4	CODE A	CODE A																																														
102	FNC1	FNC1	FNC1																																														
103	Start (CODE A)																																																
104	Start (CODE B)																																																
105	Start (CODE C)																																																
<p>Use “!” as a starting character for the control code followed by three control codes. If the start subset is not set, the default starting subset is B.</p>																																																	
EAN128	Code 128, switching code subset A, B, C automatically																																																
EAN128M	Code 128, switching code subset A, B, C manually																																																
25	Interleaved 2 of 5																																																
25C	Interleaved 2 of 5 with check digits																																																
25S	Standard 2 of 5																																																
25I	Industrial 2 of 5																																																
39	Code 39 full ASCII for TSPL2 printers Code 39 standard for TSPL printers																																																
39C	Code 39 full ASCII with check digit for TSPL2 printers Code 39 standard with check digit for TSPL printers																																																
39S	Code 39 standard for TSPL2 printers																																																
93	Code 93																																																
EAN13	EAN 13																																																



Parameter	Description
code type	EAN13+2 EAN 13 with 2 digits add-on EAN13+5 EAN 13 with 5 digits add-on EAN8 EAN 8 EAN8+2 EAN 8 with 2 digits add-on EAN8+5 EAN 8 with 5 digits add-on CODA Codabar POST Postnet UPCA UPC-A UPCA+2 UPC-A with 2 digits add-on UPCA+5 UPC-A with 5 digits add-on UPCE UPC-E UPCE+2 UPC-E with 2 digits add-on UPCE+5 UPC-E with 5 digits add-on CPOST China post code MSI MSI code MSIC MSI with check digit PLESSEY PLESSEY code ITF14 ITF 14 code EAN14 EAN 14 code 11 Code 11 TELEPEN Telepen code TELEPENN Telepen code. Number only PLANET Planet code CODE49 Code 49 DPI Deutsche Post Identcode DPL Deutsche Post Leitcode
Height	Bar code height (in dots)
human readable	0: not readable 1: human readable aligns to left 2: human readable aligns to center 3: human readable aligns to right
rotation	0: No rotation 90: Rotate 90 degrees clockwise 180: Rotate 180 degrees clockwise 270: Rotate 270 degrees clockwise



Parameter	Description				
narrow	Width of narrow element (in dots)				
wide	Width of wide element (in dots)				
	narrow : wide 1:1	narrow : wide 1:2	narrow : wide 1:3	narrow : wide 2:5	narrow : wide 3:7
<b>128</b>	10x	-	-	-	-
<b>EAN128</b>	10x	-	-	-	-
<b>EAN128M</b>	10x				
<b>25</b>	-	10x	10x	5x	-
<b>25C</b>	-	10x	10x	5x	-
<b>25S</b>		10x	10x	5x	
<b>25I</b>		10x	10x	5x	
<b>39</b>	-	10x	10x	5x	-
<b>39C</b>	-	10x	10x	5x	-
<b>93</b>	-	-	10x	-	-
<b>EAN13</b>	8x	-	-	-	-
<b>EAN13+2</b>	8x	-	-	-	-
<b>EAN13+5</b>	8x	-	-	-	-
<b>EAN 8</b>	8x	-	-	-	-
<b>EAN 8+2</b>	8x	-	-	-	-
<b>EAN 8+5</b>	8x	-	-	-	-
<b>CODA</b>	-	10x	10x	5x	-
<b>POST</b>	1x	-	-	-	-
<b>UPCA</b>	8x	-	-	-	-
<b>UPCA+2</b>	8x	-	-	-	-
<b>UPCA+5</b>	8x	-	-	-	-
<b>UPCE</b>	8x	-	-	-	-
<b>UPCE+2</b>	8x	-	-	-	-
<b>UPCE+5</b>	8x	-	-	-	-
<b>CPOST</b>	-	-	-	-	1x
<b>MSI</b>	-	-	10x	-	-
<b>MSIC</b>			10x		
<b>PLESSY</b>	-	-	10x	-	-
<b>ITF14</b>	-	10x	10x	5x	-
<b>EAN14</b>	8x	-	-	-	-
<b>11</b>	-	10x	10x	5x	-



Parameter	Description	
alignment	Specify the alignment of barcode 0: default (Left) 1: Left 2: Center 3: Right	
content	Content of barcode <i>Please note that the maximum number of digits of bar code content.</i>	
Code Type	Character sets	
<b>128</b>	See Character set for CODE128.	
<b>128M</b>	See Character set for CODE128.	
<b>EAN128</b>	See Character set for CODE128.	
<b>EAN128M</b>	See Character set for CODE128.	
<b>25</b>	0123456789	Length is even.
<b>25C</b>	0123456789	Length is odd.
<b>25S</b>	0123456789	
<b>25I</b>	0123456789	
<b>39 I</b>	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ -.\$/+%	-
<b>39 I Full ASCII</b>	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ !#\$%&()'*,.-./;[]?@[\\]^_`abcdefghijklmnopqrstuvwxyz { }~	-
<b>93</b>	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ !#\$%&()'*,.-./;[]?@[\\]^_`abcdefghijklmnopqrstuvwxyz { }~	-
<b>EAN13</b>	0123456789	12
<b>EAN13+2</b>	0123456789	14
<b>EAN13+5</b>	0123456789	17
<b>EAN8</b>	0123456789	7
<b>EAN8+2</b>	0123456789	9
<b>EAN8+5</b>	0123456789	12
<b>CODA</b>	0123456789-\$/.+	-
<b>POST</b>	0123456789	5, 9, 11
<b>UPCA</b>	0123456789	11
<b>UPCA+2</b>	0123456789	13
<b>UPA+5</b>	0123456789	16
<b>UPCE</b>	0123456789	6
<b>UPCE+2</b>	0123456789	8



Parameter	Description		
	Code Type	Character sets	Max. data length
content	UPE+5	0123456789	11
	MSI	0123456789	-
	MSIC	0123456789	-
	PLESSEY	0123456789	-
	CPOST	0123456789	-
	ITF14	0123456789	13
	EAN14	0123456789	13
	11	0123456789-	-
	TELEPEN	ASCII 0 to 127	30
	TELEPENN	0123456789	60
	PLANET	0123456789	38
	CODE49	ASCII 0 to 127	81
	DPI	0123456789	11
	DPL	0123456789	13
	LOGMARS	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ -.\$/+%	-

**Note:**

Since V5.10EZ, \[R] means carriage return character 0x0D and \[L] means line feed character 0x0A.



## Character set for CODE 128

Value	128A	128B	128C	Value	128A	128B	128C	Value	128A	128B	128C
0	space	space	00	36	D	D	36	72	BS	h	72
1	!	!	01	37	E	E	37	73	HT	i	73
2	"	"	02	38	F	F	38	74	LF	j	74
3	#	#	03	39	G	G	39	75	VT	k	75
4	\$	\$	04	40	H	H	40	76	FF	l	76
5	%	%	05	41	I	I	41	77	CR	m	77
6	&	&	06	42	J	J	42	78	SO	n	78
7	'	'	07	43	K	K	43	79	SI	o	79
8	(	(	08	44	L	L	44	80	DLE	p	80
9	)	)	09	45	M	M	45	81	DC1	q	81
10	*	*	10	46	N	N	46	82	DC2	r	82
11	+	+	11	47	O	O	47	83	DC3	s	83
12	,	,	12	48	P	P	48	84	DC4	t	84
13	-	-	13	49	Q	Q	49	85	NAK	u	85
14	.	.	14	50	R	R	50	86	SYN	v	86
15	/	/	15	51	S	S	51	87	ETB	w	87
16	0	0	16	52	T	T	52	88	CAN	x	88
17	1	1	17	53	U	U	53	89	EM	y	89
18	2	2	18	54	V	V	54	90	SUB	z	90
19	3	3	19	55	W	W	55	91	ESC	{	91
20	4	4	20	56	X	X	56	92	FS		92
21	5	5	21	57	Y	Y	57	93	GS	}	93
22	6	6	22	58	Z	Z	58	94	RS	~	94
23	7	7	23	59	[	[	59	95	US	DEL	95
24	8	8	24	60	\	\	60	96	FNC 3	FNC 3	96
25	9	9	25	61	]	]	61	97	FNC 2	FNC 2	97
26	:	:	26	62	^	^	62	98	Shift B	Shift A	98
27	;	;	27	63	-	-	63	99	Code C	Code C	99
28	<	<	28	64	NUL	`	64	100	Code B	FNC4	Code B
29	=	=	29	65	SOH	a	65	101	FNC 4	Code A	Code A
30	>	>	30	66	STX	b	66	102	FNC 1	FNC 1	FNC 1
31	?	?	31	67	ETX	c	67	103	Start Code A		
32	@	@	32	68	EOT	d	68	104	Start Code B		
33	A	A	33	69	ENQ	e	69	105	Start Code C		
34	B	B	34	70	ACK	f	70				
35	C	C	35	71	BEL	g	71				



## Example



## Example

Sample Code	Result
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "2",0,1,1, "TELEPEN" BARCODE 10,50, "TELEPEN",100,1,0,2,6, "ab- cd1234ABCD" PRINT 1 </pre>	<p>TELEPEN</p>  <p>abcd1234ABCD</p>
<pre> SIZE 4,4 GAP 0,0 DIRECTION 1 CLS TEXT 400,26, "2",0,1,1,2, "TELEPEN Number" BARCODE 400,50, "TELEPENN",60,2,0,2,6,2, "1234567890" TEXT 400,136, "2",0,1,1,2, "Code 11" BARCODE 400,160, "11",60,2,0,2,6,2, "1234567890" TEXT 400,246, "2",0,1,1,2, "PLANET" BARCODE 400,270, "PLANET",60,2,0,2,2,2, "12345678901" TEXT 400,356, "2",0,1,1,2, "Deutsche Post Identcode." BARCODE 400,380, "DPI",60,2,0,2,6,2, "12345678901" TEXT 400,466, "2",0,1,1,2, "Deutsche Post Leitcode. " BARCODE 400,490, "DPL",60,2,0,2,6,2, "123456789012" TEXT 400,576, "2",0,1,1,2, "Code 49" BARCODE 400,600, "CODE49",60,2,0,2,2,2, "1234567890" PRINT 1 </pre>	<p>TELEPEN Number</p>  <p>1234567890 Code 11</p> <p>1234567890 PLANET</p>  <p>12345678901 Deutsche Post Identcode.</p>  <p>123456789016 Deutsche Post Leitcode.</p>  <p>01234567898128 Code 49</p>  <p>1234567890</p>





## TLC39

### Description

This command draws TLC39, TCIF Linked Bar Code 3 of 9, barcode.

### Syntax

**TLC39 x,y,rotation,[height,]narrow,]wide,]cellwidth,]cellheight,] “ECI number,  
Serial number & additional data”**

Parameter	Description
x	Specify the x-coordinate
y	Specify the y-coordinate
rotation	0 : No rotation 90 : Rotate 90 degrees clockwise 180 : Rotate 180 degrees clockwise 270 : Rotate 270 degrees clockwise
height	Height of Code39 in dots (Default is 40)
narrow	Width of narrow element of Code39 in dots (Default is 2)
wide	Width of wide element of Code39 in dots (Default is 4)
cellwidth	Width of cell of MicroPDF417 in dots (Default is 2)
cellheight	Height of cell of MicroPDF417 in dots (Default is 4)
ECI number	Must be 6 digits which is used to generate Code39
Serial number & additional data	Alphanumeric is for Micro-PDF417

#### Note :

- Comma (“”) is necessary between ECI number and Serial number & additional data.
- This command has been supported since V6.89 EZ and later firmware.



## Example

### Sample Code

```
SIZE 4,1.2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "TLC39 code"
TLC39 10,50,0, "123456,SN00000001,00601,01501"
TLC39 310,50,0,80,3,6,3,4, "123456,SN00000001,00601,01501"
PRINT 1
```

### Result

TLC39 code





## BITMAP

### Description

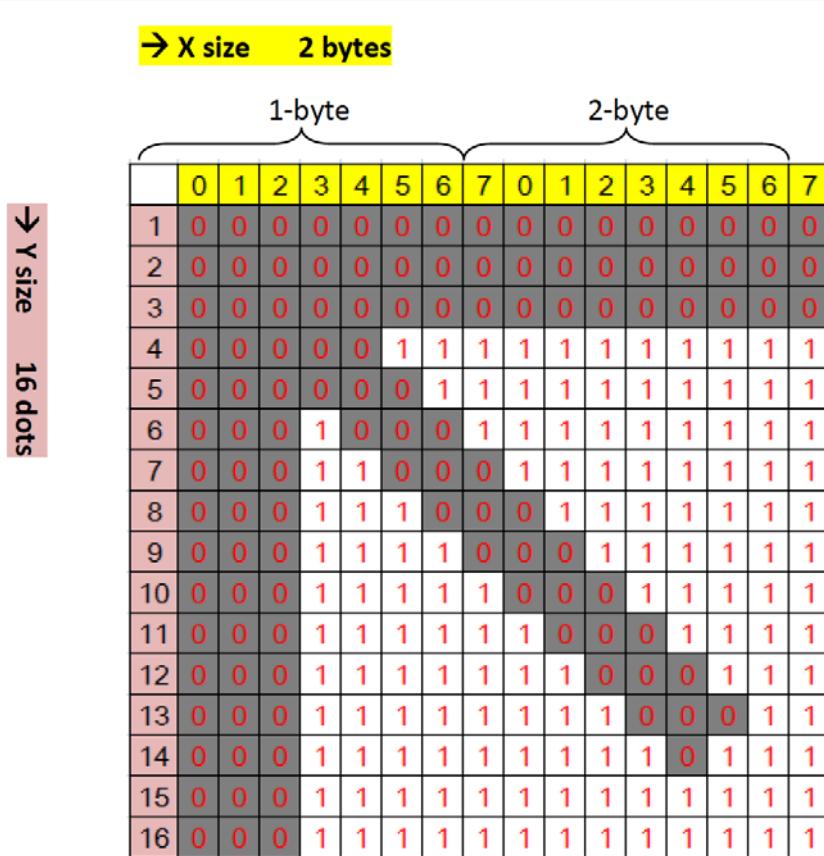
This command draws bitmap images (as opposed to BMP graphic files).

### Syntax

**BITMAP X,Y,width,height,mode,bitmap data...**

Parameter	Description
x	Specify the x-coordinate
y	Specify the y-coordinate
width	Image width (in bytes)
height	Image height (in dots)
mode	Graphic modes listed below: 0: OVERWRITE 1: OR 2: XOR
bitmap data	Bitmap data

### Example

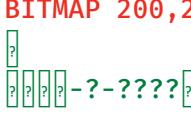




## Example

Y- axis	X – axis			
	1-byte		2-byte	
	Binary	Hexadecimal	Binary	Hexadecimal
1	00000000	00	00000000	00
2	00000000	00	00000000	00
3	00000000	00	00000000	00
4	00000111	07	11111111	FF
5	00000011	03	11111111	FF
6	00010001	11	11111111	FF
7	00011000	18	11111111	FF
8	00011100	1C	01111111	7F
9	00011110	1E	00111111	3F
10	00011111	1F	00011111	1F
11	00011111	1F	10001111	8F
12	00011111	1F	11000111	C7
13	00011111	1F	11100011	E3
14	00011111	1F	11110111	F7
15	00011111	1F	11111111	FF
16	00011111	1F	11111111	FF

Sample Code (ASCII)	Hexadecimal	Result
<b>SIZE 4,2</b> <b>GAP 0,0</b> <b>CLS</b> <b>BITMAP 200,200,2,16,0,</b>  <b>PRINT 1,1</b>	53 49 5A 45 20 34 2C 32 0D 0A 47 41 50 20 30 2C 30 0D 0A 43 4C 53 0D 0A 42 49 54 4D 41 50 20 32 30 30 2C 32 30 30 2C 32 2C 31 36 2C 30 2C 00 00 00 00 00 00 07 FF 03 FF 11 FF 18 FF 1C 7F 1E 3F 1F 1F 1F 8F 1F C7 1F E3 1F E7 1F FF 1F FF 0D 0A 50 52 49 4E 54 20 31 2C 31 0D 0A	

## See Also

[PUTBMP](#), [PUTPCX](#)



## BOX

### Description

This command draws rectangles on the label.

### Syntax

**BOX x,y,x\_end,y\_end, line thickness[,radius]**

Parameter	Description
x	Specify x-coordinate of upper left corner (in dots)
y	Specify y-coordinate of upper left corner (in dots)
x_end	Specify x-coordinate of lower right corner (in dots)
y_end	Specify y-coordinate of lower right corner (in dots)
line thickness	Line thickness (in dots)
radius	Optional. Specify the round corner. Default is 0. *Since V5.28 EZ

#### Note:

- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1 mm = 12 dots
- Recommended max. thickness of box is 12 mm at 4" width. Thickness of box larger than 12 mm may damage the power supply and affect the print quality. Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.

### Example

Sample code	Result
<pre>SIZE 4,1.1 CLS BOX 60,60,610,210,4 BOX 80,80,590,190,4 BOX 100,100,570,170,4,20 BOX 120,120,550,150,4,20 PRINT 1</pre>	

### See Also

BAR



## CIRCLE

### Description

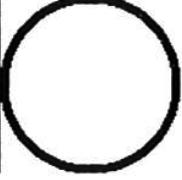
This command draws a circle on the label.

### Syntax

**CIRCLE x\_start,y\_start,diameter,thickness**

Parameter	Description
x_start	Specify x-coordinate of upper left corner (in dots)
y_start	Specify y-coordinate of upper left corner (in dots)
diameter	Specify the diameter of the circle (in dots)
thickness	Thickness of the circle (in dots)

### Example

Sample code	Result
<pre>SIZE 80 mm,30 mm GAP 0,0 DIRECTION 1 CLS BAR 250,20,100,1 BAR 250,20,1,100 CIRCLE 250,20,100,5 PRINT 1</pre>	



## ELLIPSE

### Description

This command draws an ellipse on the label.

### Syntax

**ELLIPSE x,y,width,height,thickness**

Parameter	Description
x	Specify x-coordinate of upper left corner (in dots)
y	Specify y-coordinate of upper left corner (in dots)
width	Specify the width of the ellipse (in dots)
height	Specify the height of the ellipse (in dots)
thickness	Thickness of the ellipse (in dots)

**Note:**

This command has been supported since V6.91 EZ and later firmware.

### Example

Sample code	Result
<pre>SIZE 4,3 GAP 0,0 DIRECTION 1 CLS BOX 10,10,410,110,1 ELLIPSE 10,10,400,100,2 BOX 10,120,110,520,1 ELLIPSE 10,120,100,400,5 PRINT 1</pre>	



## CODABLOCK F MODE

### Description

This command draws CODABLOCK F mode barcode.

### Syntax

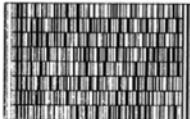
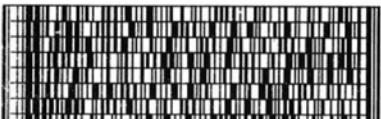
**CODABLOCK x,y,rotation,[row height,]module width,] “content”**

Parameter	Description
x	Specify the x-coordinate
y	Specify the y-coordinate
rotation	0: No rotation 90: Rotate 90 degrees clockwise 180: Rotate 180 degrees clockwise 270: Rotate 270 degrees clockwise
row height	The height of individual row equals to row height x module width (Default is 8)
module width	Width of narrow element of CODABLOCK in dots (Default is 2)
content	content of CODABLOCK bar code

**Note:**

This command has been supported since V6.89 EZ and later firmware.

### Example

Sample code	Result
<pre> SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "Codablock F" CODABLOCK 10,50,0, "We stand behind our products with one of the most comprehensive support programs in the Auto-ID industry." PRINT 1 CLS TEXT 10,10, "3",0,1,1, "Codablock F" CODABLOCK 10,50,0,16,1, "We stand be- hind our products with one of the most comprehensive support programs in the Auto-ID industry." PRINT 1 </pre>	<p style="text-align: center;">Codablock F</p>  <p style="text-align: center;">Codablock F</p> 



## DMATRIX

### Description

This command defines a DataMatrix 2D bar code. Currently, only ECC200 error correction is supported.

### Syntax

**DMATRIX x,y,width,height,[c#,x#,r#,a#,row,col,] "content"**

Parameter	Description
x	Horizontal start position (in dots)
y	Vertical start position (in dots)
width	The expected width of barcode area (in dots)
height	The expected height of barcode area (in dots)
c#	Escape sequence control character (decimal digit) Ex. C126 means ~

1. ~X is shift character for control characters.

-X	Hex	ASCII									
~@	00	NUL	~H	08	BS	~P	10	DLE	~X	18	CAN
~A	01	SOH	~I	09	HT	~Q	11	DC1	~Y	19	EM
~B	02	STX	~J	0A	LF	~R	12	DC2	~Z	1A	SUB
~C	03	ETX	~K	0B	VT	~S	13	DC3	~[	1B	ESC
~D	04	EOT	~L	0C	FF	~T	14	DC4	~\	1C	FS
~E	05	ENQ	~M	0D	CR	~U	15	NAK	~]	1D	GS
~F	06	ACK	~N	0E	SO	~V	16	SYN	~^	1E	RS
~G	07	BEL	~O	0F	SI	~W	17	ETB	~_	1F	US

2. ~1 means FNC1.
3. ~dNNN creates ASCII decimal value NNN for a codeword. Must be 3 digits. 000 ~ 255.
4. ~ in data is encoded by ~~.

x#	Module size (in dots)
r#	Rotation 0: No rotation 90: Rotate 90 degrees clockwise 180: Rotate 180 degrees clockwise 270: Rotate 270 degrees clockwise
a#	0: Square (default) 1: Rectangle
row	Symbol size of row: 10 to 144
col	Symbol size of col: 10 to 144
content	Content of DataMatrix 2D bar code



Parameter	Description
-----------	-------------

**Note:**

- This command has been supported since V6.89 EZ and later firmware. The parameter "a#" has been supported since V8.01 EZ and later firmware.
- For standard symbol sizes for DataMatrix 2D barcode, please refer to below list.

Square	Rectangle
10 x 10	26 x 26
12 x 12	32 x 32
14 x 14	36 x 36
16 x 16	40 x 40
18 x 18	44 x 44
20 x 20	48 x 48
22 x 22	52 x 52
24 x 24	64 x 64
	72 x 72
	80 x 80
	88 x 88
	96 x 96
	104 x 104
	120 x 120
	144 x 144
	8 x 18
	12 x 26
	16 x 36
	16 x 36
	16 x 48
	-
	-

## Example

Sample code	Result
<pre> SIZE 4,3 GAP 0,0 DIRECTION 1 CLS DMATRIX 10,110,400,400, "DMATRIX EXAMPLE 1" DMATRIX 310,110,400,400,x6, "DMATRIX EXAMPLE 2" DMATRIX 10,310,400,400,x8,18,18, "DMATRIX EXAMPLE 3" PRINT 1,1 </pre>	
Sample code for FNC	Result
<pre> SIZE 4,1 GAP 0,0 CLS DIRECTION 1 DMATRIX 100,50,100,100,c126,x6,18,18, "~1241sPn~110sLot~130sQty " PRINT 1 </pre>	
Sample code in rectangular shape	Result
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS DMATRIX 100,110,600,600,a1,"DMATRIX EXAMPLE 1" PRINT 1,1 </pre>	



## ERASE

### Description

This command clears a specified region in the image buffer.

### Syntax

**ERASE x,y,x\_width,y\_height**

Parameter	Description
x	The x-coordinate of the starting point (in dots)
y	The y-coordinate of the starting point (in dots)
x_width	The region width in x-axis direction (in dots)
y_height	The region height in y-axis direction (in dots)

### Example

Sample code	Result
<pre>SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS BAR 100,100,300,300 ERASE 150,150,200,200 PRINT 1,1</pre>	

### See Also

CLS



## MAXICODE

### Description

This command defines a 2D Maxicode.

### Syntax

**MAXICODE x,y,mode,[class,country,post,Lm,] “content”**

**MAXICODE x,y,mode,class,country,postal code, “content”** For mode 2 or 3, If country is 840, the postal code is in 99999,9999 format. For other countries, the code is up to 6 alphanumeric characters.

**MAXICODE x,y,mode,[Lm,] “content”**

For mode 4,5,6, AIM special format is supported, see page 23 in the spec.

*Mode 6 is not supported in TSPL2 printer firmware.*

Parameter	Description
x	X-coordinate of the starting point (in dot)
y	Y-coordinate of the starting point (in dot)
mode	2,3,4,5
class	Class of service, 3-digit number (for mode 2,3)
country	Country code, 3-digit number (for mode 2,3)
post	Post code (for mode 2,3) Mode 2(USA): 5-digit + 4-digit number Mode 3(Canada): 6 alphanumeric post code included by double quotes.
Lm	Expression length (double quote is ignored) , $1 \leq m \leq 138$ , (this parameter is just for mode 4 and 5)
content	Content of 2D Maxicode

#### Note:

If parameter Lm is used, double quotes (“”) are unnecessary.

### Example

#### Sample code

```
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
```



### Sample code

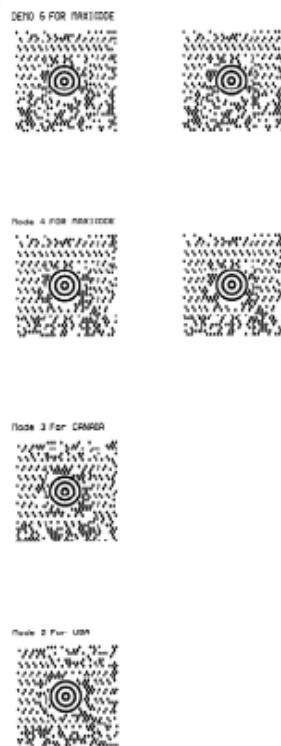
```
REM *****Mode 2 For USA*****
MAXICODE 110,100,2,300,840,06810,7317, "DEMO 2 FOR USA MAXICODE"
TEXT 100,50, "3",0,1,1, "Mode 2 For USA"
PRINT 1,1

REM *****Mode 3 For Canada*****
CLS
MAXICODE 110,100,3,300,863, "107317", "DEMO 3 FOR CANADA MAXICODE"
TEXT 100,50, "3",0,1,1, "Mode 3 For CANADA"
PRINT 1,1

REM *****MODE4*****
CLS
MAXICODE 110,100,4, "DEMO 4 FOR MAXICODE"
MAXICODE 600,100,4,L19,DEMO 4 FOR MAXICODE
TEXT 100,50, "3",0,1,1, "Mode 4 FOR MAXICODE"
PRINT 1,1

REM *****MODE 5*****
CLS
MAXICODE 110,100,5, "DEMO 5 FOR MAXICODE"
MAXICODE 600,100,5,L19,DEMO 5 FOR MAXICODE
TEXT 100,50, "3",0,1,1, "DEMO 5 FOR MAXICODE"
PRINT 1
```

### Result





## PDF417

### Description

This command defines a PDF417 2D bar code.

### Syntax

**PDF417 x,y,width,height,[option], “content”**

Parameter	Description	
x	X-coordinate of starting point (in dot)	
y	Y-coordinate of starting point (in dot)	
width	Expected width (in dots)	
height	Expected height (in dots)	
rotate	Rotation counterclockwise 0: No rotation 90: Rotate 90 degrees 180: Rotate 180 degrees 270: Rotate 270 degrees	
option	P Data compression method 0: Auto encoding 1: Binary mode	
	E Error correction level (Range: 0~8)	
M	Center pattern in barcode area 0: The pattern will print upper left justified the area 1: The pattern is printed middle of area	
Ux,y,c	Human readable x: Human readable characters in the specified x-coordinate y: Human readable characters in the specified y-coordinate c: Maximum characters of human readable character per line	
W	Module width in dot (Range: 2~9)	
H	Bar height in dot (Range: 4~99)	
R	Maximum number of rows	
C	Maximum number of columns	
T	Truncation 0: Not truncated 1: Truncated	
Lm	Expression length, $1 \leq m \leq 2048$ (without “ for content)	
content	Content of PDF417 2D bar code	

**Note:**

If parameter Lm is used, double quotes (“) are unnecessary for content.



Sample code	Result
<pre>SIZE 4,1 GAP 0,0 DIRECTION 1  REM *****WITHOUT OPTIONS***** CLS PDF417 50,50,400,200,0, "Without Options" PRINT 1,1</pre>	
<pre>SIZE 4,1.5 GAP 0,0 DIRECTION 1  REM *****OPTION:E3***** CLS PDF417 50,50,400,200,0,E3, "Error correction level:3" PRINT 1,1</pre>	
<pre>REM *****OPTION:E4***** CLS PDF417 50,50,400,200,0,E4, "Error correction level:4" PRINT 1,1</pre>	
<pre>SIZE 4,1.5 GAP 0,0 DIRECTION 1  REM *****OPTION:E4 W4***** CLS PDF417 50,50,600,600,0,E4,W4, "Error correction level:4 module width 4 dots" PRINT 1,1</pre>	
<pre>REM *****OPTION:E4 W4 H4***** CLS PDF417 50,50,600,600,0,E4,W4,H4, "Error correction lev- el:4 module width 4 dots bar height 4 dots" PRINT 1,1</pre>	



Sample code	Result
<pre>SIZE 4,1.5 GAP 0,0 DIRECTION 1  REM *****OPTION:E4 W4 H4 R40 C4 T1***** CLS PDF417 50,50,800,800,0,E4,W4,H4,R40,C4,T1, "Error correction level:4 Module Width 4 dots Bar Height 4 dots Maximum Number of Rows:5 Rows Maximum number of columns:90 Cols Truncation:1" PRINT 1,1</pre>	
<pre>SIZE 4,2.5 GAP 0,0 DIRECTION 1  REM *****OPTION:P1 E4 M1 U50,300,50,W4,H4,R60,C4,T0,L297***** CLS PDF417 50,50,900,600,0,P1,E4,M1,U50,300,50,W4,H4, R60,C4,T0,L297,Data compression method: P1 Error correction level: E4 Center pattern in barcode area: M1 Human Readable: Yes: U50,300,50 Module Width 4 dots: W4 Bar Height 4 dots: H4 Maximum Number of Rows: 60 Rows: R60 Maximum number of columns: 4 Cols: C4 Truncation:1: T0 Expression length:297: L297 PRINT 1,1</pre>	 <p>Data compression method: P1 Error correction leve l: E4 Center pattern in barcode area: M1 Human R eadable: Yes: U50,300,50 Module Width 4 dots: W4 Bar Height 4 dots: H4 Maximum Number of Rows: 60 Rows: R60 Maximum number of columns: 4 Cols: C4 Truncation:1: T0 Expression length:297: L297</p>



## AZTEC

### Description

This command defines a AZTEC 2D bar code.

### Syntax

**AZTEC x,y,rotate,[size,]ecp,]flg,]menu,]multi,]rev,] “content”** Since V6.60EZ  
**AZTEC x,y,rotate,size,ecp,flg,menu,multi,rev,bytes,content** Since V6.91EZ

Parameter	Description
x	Horizontal start position (in dots)
y	Vertical start position (in dots)
rotate	Rotation 0: No rotation 90: Rotate 90 degrees 180: Rotate 180 degrees 270: Rotate 270 degrees
size	Element module size (1 to 20), default is 6
ecp	Error control (& symbol size/type) parameter 0: default error correction level 1 to 99: minimum error correction percentage 101 to 104: 1 to 4-layer Compact symbol 201 to 232: 1 to 32-layer Full-Range symbol 300: a simple Aztec “Rune”
flg	0: input message is straight bytes 1: input uses “<Esc>n” for FLG(n), “<Esc><Esc>” for “<Esc>”
menu	Menu symbol (0 : no, 1 : yes), default is 0
multi	Number of symbols (1 to 26), default is 6
rev	Output to be reversed (0 : no, 1 : yes), default is 0
bytes	Length of content
content	Content of AZTEC 2D bar code

**Note:**

If parameter bytes is used, double quotes (“”) are unnecessary.

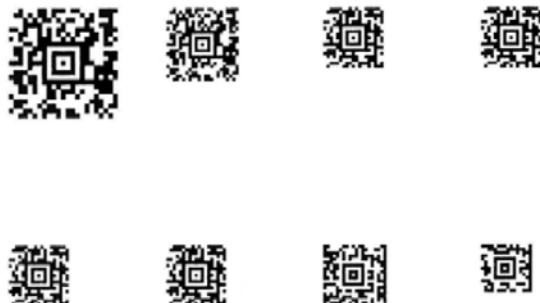


## Example

### Sample Code

```
SIZE 4,2  
GAP 0,0  
CLS  
AZTEC 10,10,0,"ABCDEFGHIJKLM NOPQRSTUVWXYZ0123456789"  
AZTEC 210,10,0,4,"ABCDEFGHIJKLM NOPQRSTUVWXYZ0123456789"  
AZTEC 410,10,0,4,1,"ABCDEFGHIJKLM NOPQRSTUVWXYZ0123456789"  
AZTEC 610,10,0,4,1,0,"ABCDEFGHIJKLM NOPQRSTUVWXYZ0123456789"  
AZTEC 10,310,0,4,1,0,0,"ABCDEFGHIJKLM NOPQRSTUVWXYZ0123456789"  
AZTEC 210,310,0,4,1,0,0,1,"ABCDEFGHIJKLM NOPQRSTUVWXYZ0123456789"  
AZTEC 410,310,0,4,1,0,0,1,1,"ABCDEFGHIJKLM NOPQRSTUVWXYZ0123456789"  
AZTEC 610,310,0,4,1,0,0,1,1,10,1234567890  
PRINT 1
```

### Result





## MPDF417

### Description

This command defines a Micro PDF 417 bar code.

### Syntax

**MPDF417 x,y,rotate,[Wn,][Hn,][Cn,] “content”**

Parameter	Description
x	Horizontal start position (in dots)
y	Vertical start position (in dots)
rotate	Rotation 0: No rotation 90: Rotate 90 degrees 180: Rotate 180 degrees 270: Rotate 270 degrees
Wn	Optional. Module width in dot. Default is 1.
Hn	Optional. Module height in dot. Default is 10.
Cn	Optional. Number of columns. Once the parameter is set, the printer will calculate the proper rows for the barcode base on the content automatically. 0: Auto mode. 1: Column is 1 and the calculated suitable rows will be 11, 14, 17, 20, 24, and 28. 2: Column is 2 and the calculated suitable rows will be 8, 11, 14, 17, 20, 23 and 26. 3: Column is 3 and the calculated suitable rows will be 6, 8, 10, 12, 15, 20, 26, 32, 38 and 44. 4: Column is 4 and the calculated suitable rows will be 4, 6, 8, 10, 12, 15, 20, 26, 32, 38 and 44.
Content	Content of Micro PDF 417 bar code

#### Note:

This command has been supported since V6.61 EZ and later firmware.

### Example

Sample Code	Result
<pre> SIZE 4,1 GAP 0,0 CLS MPDF417 10,10,0, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " MPDF417 110,10,0,W2, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " MPDF417 210,10,0,W2,H3, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " MPDF417 310,10,0,W2,H3,C3, "ABCDEFGHIJKLMNOPQRSTUVWXYZ- YZ0123456789 " PRINT 1 </pre>	



## PUTBMP

### Description

This command prints BMP format images. The grayscale printing is for direct thermal mode only. Support 1-bit (monochrome) and 8-bit (256-color) BMP graphic only.

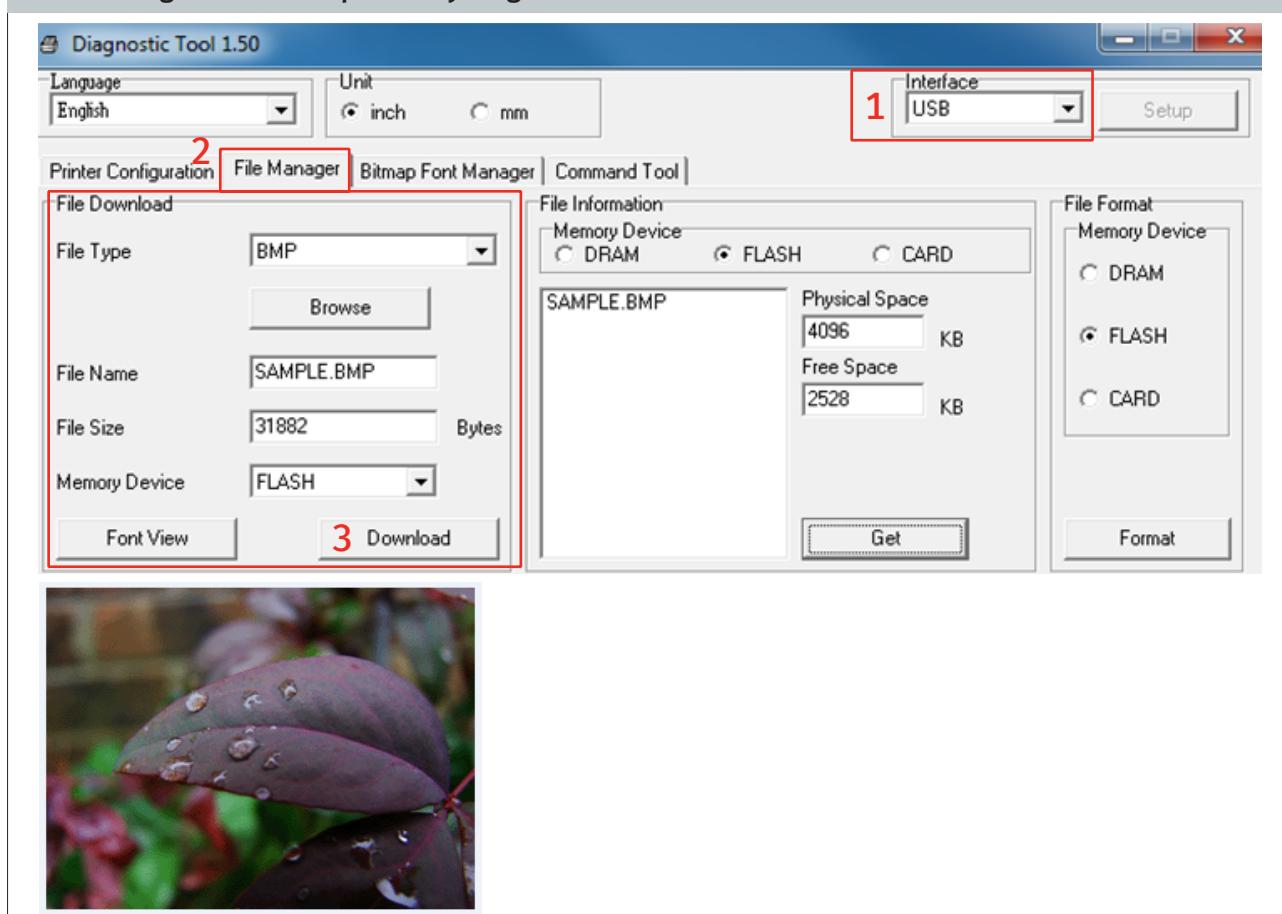
### Syntax

**PUTBMP x,y, "filename" [, bpp][, contrast]**

Parameter	Description
x	The x-coordinate of the BMP format image
y	The y-coordinate of the BMP format image
filename	The downloaded BMP filename (Support ZPL *.GRF)
bpp	Optional. Bits per pixel of grayscale graphic. Default is 1. *Since V6.91EZ. 1: 1-bit (monochrome) graphic 8: 8-bit (256-color) graphic
contrast	Optional. Contrast of grayscale graphic. Default is 80. Suggested range is from 60 to 100. *Since V6.91EZ.

### Example

#### Downloading BMP file into printer by DiagTool





## Example

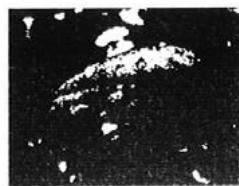
### Sample Code

```
SPEED 2
DENSITY 3
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
PUTBMP 10,10,"SAMPLE.BMP"
BLOCK 10,180,240,100,"2",0,1,1,"bpp and contrast are omitted."
PUTBMP 300,10, "SAMPLE.BMP",1,80
BLOCK 300,180,240,100,"2",0,1,1, "bpp = 1
contrast = 80"
PUTBMP 590,10, "SAMPLE.BMP",8,80
BLOCK 590,180,240,100,"2",0,1,1,"bpp = 8
contrast = 80"
PRINT 1
```

### Result



bpp and contrast  
are omitted.



bpp = 1  
contrast = 80



bpp = 8  
contrast = 80

### Sample Code

```
SIZE 2,2
GAP 0,0
CLS
PUTBMP 10,10, "SAMPLE.GRF"
PRINT 1
```

## See Also

DOWNLOAD, BITMAP, PUTPCX



## PUTPCX

### Description

This command prints PCX format images. TSPL language supports 2-color PCX format graphics. TSPL2 language supports 256-color PCX format graphics.

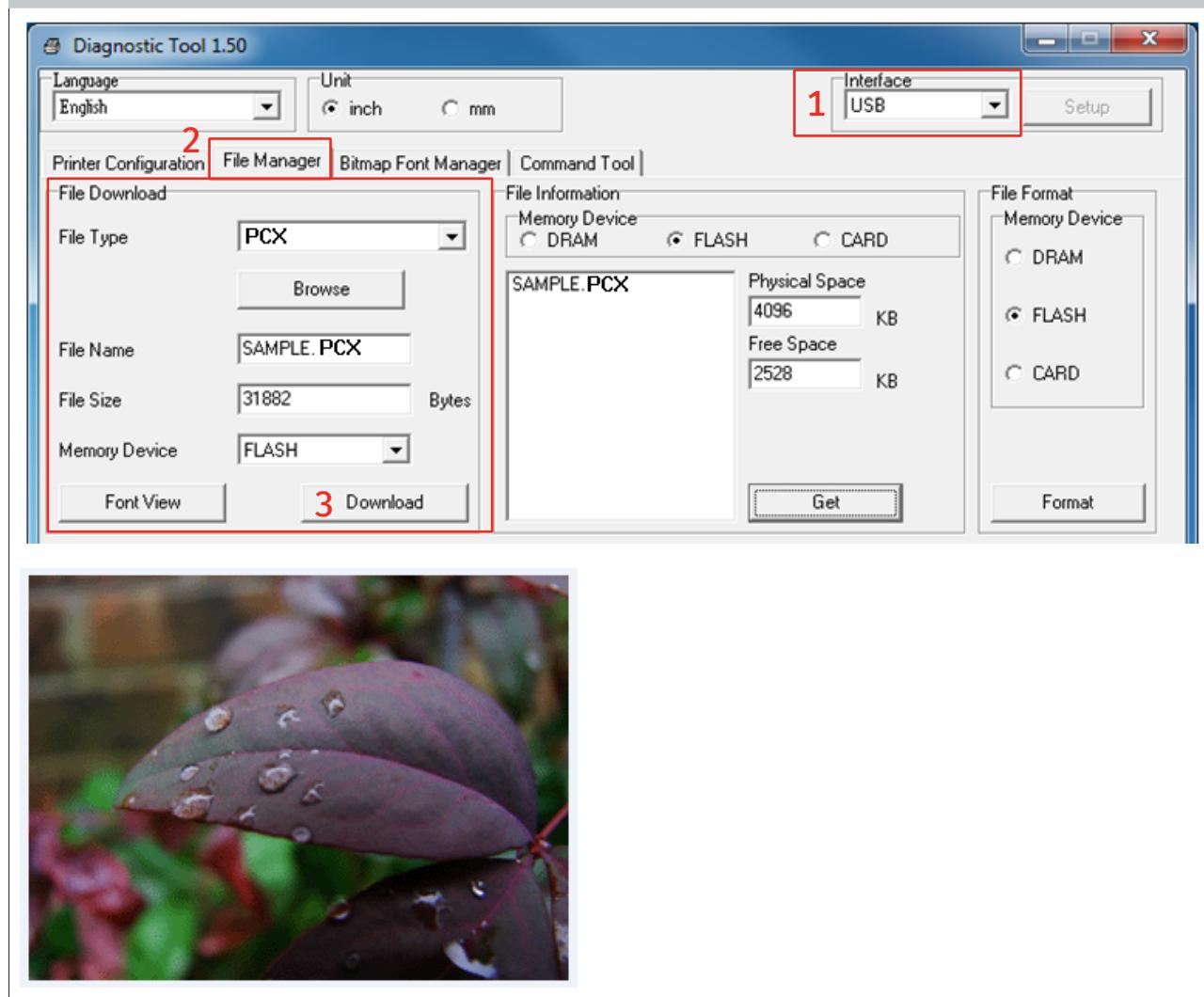
### Syntax

**PUTPCX x,y, "filename"**

Parameter	Description
x	The X-coordinate of the PCX format image
y	The Y-coordinate of the PCX format image
filename	The downloaded PCX file name (Case sensitive)

### Example

#### Downloading BMP file into printer by DiagTool



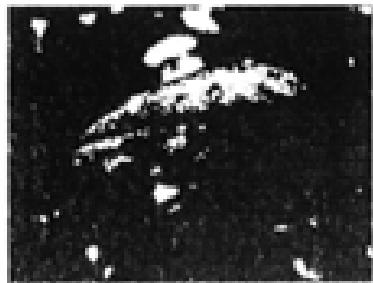


## Example

### Sample Code

```
SPEED 2  
DENSITY 3  
SIZE 4,1.5  
GAP 0,0  
DIRECTION 1  
CLS  
PUTBMP 10,10, "SAMPLE.PCX"  
PRINT 1
```

### Result



## See Also

DOWNLOAD, BITMAP, PUTPCX



## QRCODE

### Description

This command prints QR code.

### Syntax

```
QRCODE x,y,ECC Level,cell width,mode,rotation,[justification,]model,[mask,]area]
“content”
```

Parameter	Description
x	The upper left corner x-coordinate of the QR code
y	The upper left corner y-coordinate of the QR code
ECC level	Error correction recovery level L: 7% M: 15% Q: 25% H: 30%
cell width	1~10
mode	Auto / manual encode A: Auto M: Manual
rotation	0: 0 degree 90: 90 degree 180: 180 degree 270: 270 degree
[justification]	Barcode justification (J1 to J9 valid; refer to “Sample code” example below); since version A1.97 firmware.
[model]	M1: (default), original version M2: enhanced version (Almost smart phone is supported by this version.)
[mask]	S0~S8, default is S7
[area]	Maximum size of barcode area (Xdots; ex: X100); since version A1.97 firmware.



Parameter	Description
content	<p>The encodable character set is described as below,</p> <p>Encodable character set:</p> <ul style="list-style-type: none"> <li>1) Numeric data: (digits 0~9)</li> <li>2) Alphanumeric data           <ul style="list-style-type: none"> <li>Digits 0-9</li> <li>Upper case letters A-Z</li> <li>Nine other characters: space, \$ % * + - . / : )</li> </ul> </li> <li>3) 8-bit byte data           <ul style="list-style-type: none"> <li>JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201</li> </ul> </li> <li>4) Kanji characters           <ul style="list-style-type: none"> <li>Shift JIS values 8140HEX –9FFC HEX and E040HEX –EAA4 HEX. These are values shifted from those of JIS X 0208. Refer to JIS X 0208 Annex 1 Shift Coded Representation for detail.</li> </ul> </li> </ul> <p>Data characters per symbol (for maximum symbol size):</p>

	Model 1 (Version 14-L)	Model 2 (Version 40-L)
Numeric data	1,167 characters	7,089 characters
Alphanumeric data	707 characters	4,296 characters
8-bit byte data	486 characters	2,953 characters
Kanji data	299 characters	1,817 characters

- If “A” is the first character in the data string, then the following data after “A” is alphanumeric data.
- If “N” is the first character in the data string, then the following data after “N” is numeric data.
- If “B” is the first character in the data string, then the following 4 digits after “B” is used to specify numbers of data. After the 4 digits is the number of bytes of binary data to be encoded.
- If “K” is the first character in the data string, then the following data after “K” is Kanji data.
- If “!” is in the data string and follows by “N”, “A”, “B”, “K” then it will be switched to specified encodable character set.



Parameter	Description
content	<p><b>Manual mode example:</b></p> <p><b>QRCode 100,10,L,7,M,0,M1,S1, "ATHE FIRMWARE HAS BEEN UPDATED"</b> (Where A: Alphanumeric data)</p> <p><b>QRCode 100,10,M,7,M,0,M1,S2, "N123456"</b> (Where N: Numeric data)</p> <p><b>QRCode 100,10,Q,7,M,0,M1,S3, "N123456!ATHE FIRMWARE HAS BEEN UPDATED"</b> (Where N: Numeric data ; !:Transfer char ; A: Alphanumeric data)</p> <p><b>QRCode 100,10,H,7,M,0,M1,S3, "B0012Product name"</b> (where B: Binary data ; 0012: 12 bytes )</p> <p><b>QRCode 100,10,M,7,M,0,M1,S3, "K"</b> (Where K: Kanji data)</p> <p><b>Auto mode example:</b></p> <p><b>QRCode 100,10,M,7,A,0, "THE FIRMWARE HAS BEEN UPDATED"</b></p>

## Example

Sample code	Result
<p><b>Auto mode example</b></p> <p><b>General data string</b></p> <p><b>SIZE 4,2.5</b></p> <p><b>GAP 0,0</b></p> <p><b>DIRECTION 1</b></p> <p><b>CLS</b></p> <p><b>QRCode 10,10,H,4,A,0, "ABCabc123"</b></p> <p><b>QRCode 160,160,H,4,A,0, "123ABCabc"</b></p> <p><b>QRCode 310,310,M,4,A,0,M2, "印表機ABCabc123"</b></p> <p><b>PRINT 1,1</b></p>	  
<p><b>Data string including &lt;Enter&gt; character (0Dh, 0Ah)</b></p> <p><b>SIZE 4,2.5</b></p> <p><b>GAP 0,0</b></p> <p><b>DIRECTION 1</b></p> <p><b>CLS</b></p> <p><b>QRCode 10,10,H,4,A,0, "ABC&lt;Enter&gt;</b></p> <p><b>abc&lt;Enter&gt;</b></p> <p><b>123 "</b></p> <p><b>QRCode 160,160,H,4,A,0, "123&lt;Enter&gt;</b></p> <p><b>ABC&lt;Enter&gt;</b></p> <p><b>abc"</b></p> <p><b>QRCode 310,310,H,4,A,0, "印表機&lt;Enter&gt;</b></p> <p><b>ABC&lt;Enter&gt;</b></p> <p><b>abc&lt;Enter&gt;</b></p> <p><b>123"</b></p> <p><b>PRINT 1,1</b></p>	 

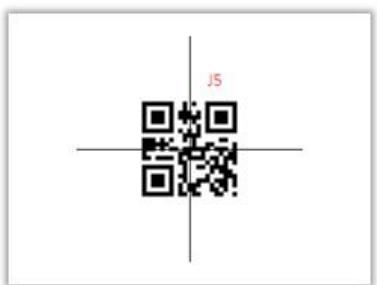


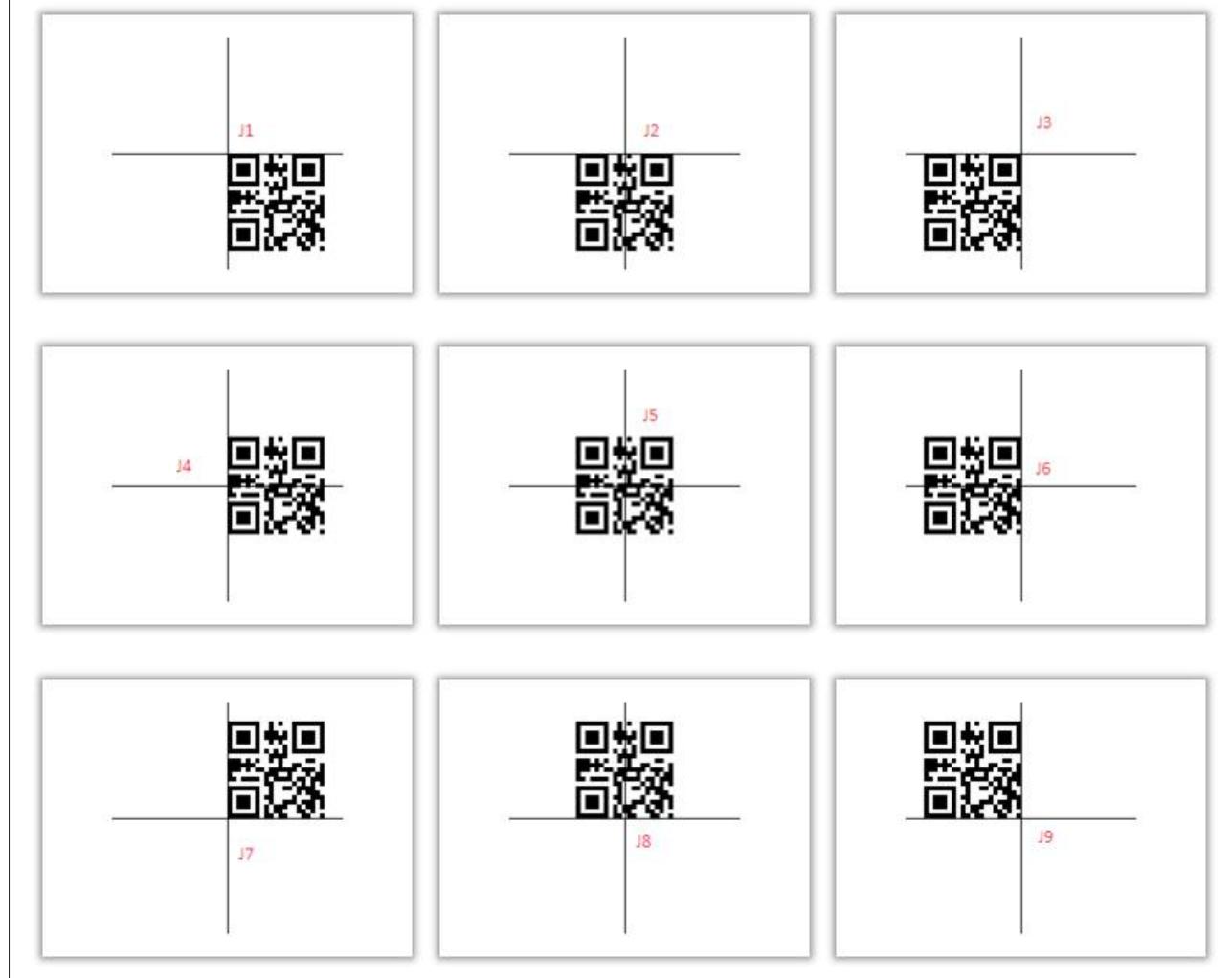
Sample code	Result
<p>Auto mode example</p> <p>Data string concatenation (Must be used with DOWNLOAD ... EOP command)</p> <pre>DOWNLOAD "DEMO.BAS" SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCode 10,10,H,4,A,0, "ABCabc123" +STR\$(1234) QRCode 160,160,H,4,A,0, "123ABCabc" +"1234" QRCode 310,310,H,4,A,0, "印表機 ABCabc123"+"1234"+abcd" PRINT 1,1 EOP DEMO</pre>	  
<p>Data string including double quote ("") character, please use \" instead of</p> <pre>SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCode 10,10,H,4,A,0, "ABC\"["]abc\"["]123" QRCode 160,160,H,4,A,0, "123\"["]ABC\"["]abc" QRCode 310,310,H,4,A,0, "\"["]印表機\"["]ABCabc123" PRINT 1,1</pre>	  



Sample code	Result
<b>Manual mode</b> <p>General data string</p> <pre>SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCode 10,10,H,4,M,0, "AABC!B0003abc!N123" QRCode 160,160,H,4,M,0, "N123!AABC!B0003abc" QRCode 310,310,H,4,M,0, "K印表機!AABC!B0006abc123" PRINT 1,1</pre>	  
<p>Data string including &lt;Enter&gt; character, &lt;Enter&gt; is an 8-bit byte data</p> <pre>SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCode 10,10,H,4,M,0, "AABC!B0007&lt;Enter&gt; abc&lt;Enter&gt; !N123" QRCode 160,160,H,4,M,0, "N123!B0002&lt;Enter&gt; !AABC!B0005&lt;Enter&gt; abc" QRCode 310,310,H,4,M,0, "K印表機!B0002&lt;Enter&gt; !AABC!B0010&lt;Enter&gt; abc&lt;Enter&gt; 123" PRINT 1,1</pre>	  
<p>Data string concatenation (Must be used with DOWNLOAD ... EOP command)</p> <pre>DOWNLOAD "A.BAS" SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCode 10,10,H,4,M,0, "AABC!B0006abc123!N"+STR\$(1234) QRCode 160,160,H,4,M,0, "N123!AABC!B0007abc"+"1234" QRCode 310,310,H,4,M,0, "K印表機!AABC!B0014abc123"+" 1234"+"abcd" PRINT 1,1 EOP A</pre>	  



Sample code	Result
<p><b>Manual mode</b></p> <p>Data string including double quote ("") character, please use \" instead of "</p> <pre data-bbox="160 507 981 893">SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCode 10,10,H,4,M,0, "AABC!B0005\"[\"]abc\[\"]!N123" QRCode 160,160,H,4,M,0, "N123!B0001\[\"]!AABC!B0004\[\"]abc" QRCode 310,310,H,4,M,0, "B0001\[\"]K印表機!B0010\[\"] ABCabc123" PRINT 1,1</pre>	  
<p><b>Smart phone data string</b></p> <pre data-bbox="160 968 933 1439">DOWNLOAD "A.BAS" SIZE 3,3 GAP 0,0 DIRECTION 1 CLS QRCode 10,10,H,7,M,0,M2,S7,"Aabcd" QRCode 170,170,H,4,M,0, M2,"B0008繁體中文" QRCode 300,300, L, 8, M, 0, M2,"B0026http://www. meto.com/metoprint" PRINT 1,1 EOP A</pre>	  
<p><b>Data string for parameter [justification] &amp; [area]</b></p> <pre data-bbox="160 1507 854 1821">SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS BAR 60,120,200,1 BAR 160,20,1,200 QRCode 160,120,H,10,A,0,X100,J5,"123456789" PRINT 1,1</pre>	

**Sample code****Manual mode****For other [justification] results (J1~J9)**



## RSS

### Description

This command is used to draw a RSS bar code on the label format.

### Syntax

```
RSS x,y, "sym",rotate,pixMult,sepHt, "content"  
RSS x,y, "RSSEXP",rotate,pixMult,sepHt,segWidth, "content"  
RSS x,y, "UCC128CCA",rotate,pixMult,sepHt,linHeight, "content"  
RSS x,y, "UCC128CCC",rotate,pixMult,sepHt,linHeight, "content"
```

Parameter	Description
x	X-coordinate
y	Y-coordinate
sym	Symbology type:  <b>RSS14</b> RSS14 <b>RSS14T</b> RSS14 Truncated <b>RSS14S</b> RSS14 Stacked <b>RSS14SO</b> RSS14 Stacked Omnidirectional <b>RSSLIM</b> RSS Limited <b>RSSEXP</b> RSS Expanded <b>UPCA</b> UPC-A <b>UPCE</b> UPC-E <b>EAN13</b> EAN-13 <b>EAN8</b> EAN-8 <b>UCC128CCA</b> UCC/EAN-128 & CC-A/B <b>UCC128CCC</b> UCC/EAN-128 & CC-C
rotate	Rotation (0, 90, 180, and 270 valid)



Parameter	Description
pixMult	Module width in dot (1 to 10 valid) The following barcode height is calculated by printer.
<b>RSS14</b>	$33 \times \text{pixMult}$ .
<b>RSS14T</b>	$13 \times \text{pixMult}$ .
<b>RSS14S</b>	$13 \times \text{pixMult}$ .
<b>RSS14SO</b>	$33 \times \text{pixMult}$ .
<b>RSSLIM</b>	$13 \times \text{pixMult}$ .
<b>RSSEXP</b>	$33 \times \text{pixMult}$ .
<b>EAN8</b>	$60 \times \text{pixMult}$ .
<b>EAN13</b>	$74 \times \text{pixMult}$ .
<b>UPCA</b>	$74 \times \text{pixMult}$ .
<b>UPCE</b>	$74 \times \text{pixMult}$ .
sepHt	Separator row height (1 and 2 valid) <i>pixMult times sepHt is the real separator row height. It is calculated by printer.</i>
segWidth	Segment width of RSS expanded (even 2 to 22 valid)
linHeight	UCC/EAN-128 height in dot (1 to 500 valid)
content	Barcode content or string expression Content of UPCE must be: <ul style="list-style-type: none"><li>• 00abc0000hij = abhijc, where c = 0-2</li><li>• 00abc00000ij = abcij3</li><li>• 00abcd00000j = abcdj4</li><li>• 00abcde0000j = abcdej where j = 5-9</li></ul>

**Note:**

- **200 DPI: 1 mm = 8 dots**
- 300 DPI: 1 mm = 12 dots**
- Recommended max. height of reversed black area is 12 mm at 4 " width. Height of reversed area that is larger than 12 mm may damage the power supply and affect the print quality.
- Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.
- This command has been supported since V6.56 EZ and later firmware.



## Example

Sample code	Result
<pre> SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSS14",0,2,2, "1234567890 ABCDEFG" RSS 300,300,"RSS14T",90,2,2,"1234567890 ABCDEFG" RSS 300,300,"RSS14S",180,2,2,"1234567890 ABCDEFG" RSS 300,300, "RSS14S0",270,2,2, "1234567890 ABCDEFG" PRINT 1,1 </pre>	
<pre> SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSSLIM",0,2,2, "1234567890 ABCDEFG" RSS 300,300, "RSSEXP",90,2,2,22, "1234567890 ABCDEFG" RSS 300,300, "UPCA",180,2,2, "1234567890 ABCDEFG" RSS 300,300, "UPCE",270,2,2, "000 ABCDEFG" PRINT 1,1 </pre>	
<pre> SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300,"EAN13",0,2,2,"123456789012 ABCDEFG" RSS 300,300,"EAN8",90,2,2,"1234567 ABCDEFG" RSS 300,300,"UCC128CCA",180,2,2,25,"1234567890 ABCDEFG" RSS 300,300,"UCC128CCC",270,2,2,25,"1234567890 ABCDEFG" PRINT 1,1 </pre>	
<pre> SIZE 100 mm, 100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,10, "RSSEXP",90,2,2,12, "8110106141411234562891 101201212085010048000214025610048000310123191000" PRINT 1 </pre>	



## Example

Sample code	Result
<p>Example of UPCE mode</p> <pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS REM UPCE Rule 1: 00abc0000hij = abhijc, where c = 0-2 RSS 10,10,"UPCE",0,2,2,"001200000456 ABCDEFG" RSS 210,10,"UPCE",0,2,2,"001210000456 ABCDEFG" RSS 410,10,"UPCE",0,2,2,"001220000456 ABCDEFG" PRINT 1 </pre>	
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS REM UPCE Rule 2: 00abc00000ij = abcij3 RSS 10,10,"UPCE",0,2,2,"001230000045 ABCDEFG" PRINT 1 SIZE 4,1 CLS REM UPCE Rule 3: 00abcd00000j = abcdj4 RSS 10,10,"UPCE",0,2,2,"001234000005 ABCDEFG" PRINT 1 </pre>	
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS REM UPCE Rule 4: 00abcde0000j = abcdej where j = 5-9 RSS 10,10,"UPCE",0,2,2,"001234500005 ABCDEFG" RSS 160,10,"UPCE",0,2,2,"001234500006 ABCDEFG" RSS 310,10,"UPCE",0,2,2,"001234500007 ABCDEFG" RSS 460,10,"UPCE",0,2,2,"001234500008 ABCDEFG" RSS 610,10,"UPCE",0,2,2,"001234500009 ABCDEFG" PRINT 1 </pre>	



Sample code	Result
<b>Example of barcode height of EAN8 EAN13 UPCA and UPCE.</b>	
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"EAN8",0,1,1,"1234567 ABCDEFG" RSS 210,10, "EAN8",0,2,1,"1234567 ABCDEFG" RSS 410,10, "EAN8",0,3,1,"1234567 ABCDEFG" PRINT 1 </pre>	
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"EAN13",0,1,1,"123456789012 ABCDEFG" RSS 210,10,"EAN13",0,2,1,"123456789012 ABCDEFG" RSS 410,10,"EAN13",0,3,1,"123456789012 ABCDEFG" PRINT 1 </pre>	
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"UPCA",0,1,1,"12345678901 ABCDEFG" RSS 210,10,"UPCA",0,2,1,"12345678901 ABCDEFG" RSS 410,10,"UPCA",0,3,1,"12345678901 ABCDEFG" PRINT 1 </pre>	
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"UPCE",0,1,1,"001200000456 ABCDEFG" RSS 210,10,"UPCE",0,2,1,"001210000456 ABCDEFG" RSS 410,10,"UPCE",0,3,1,"001220000456 ABCDEFG" PRINT 1 </pre>	



## REVERSE

### Description

This command reverses a region in image buffer.

### Syntax

**REVERSE x\_start,y\_start,x\_width,y\_height**

Parameter	Description
x_start	The x-coordinate of the starting point (in dots)
y_start	The y-coordinate of the starting point (in dots)
x_width	X-axis region width (in dots)
y_height	Y-axis region height (in dots)

**Note:**

- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1 mm = 12 dots
- Recommended max. height of reversed black area is 12mm at 4" width. Height of reversed area that is larger than 12 mm may damage the power supply and affect the print quality.
- Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.

### Example

Sample code	Result
<pre>SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS TEXT 100,100,"3",0,1,1,"REVERSE" REVERSE 90,90,128,40 PRINT 1,1</pre>	



## DIAGONAL

### Description

This command is used to draw a diagonal.

### Syntax

**DIAGONAL x1, y1, x2, y2, thickness**

Parameter	Description
x1	The x1-coordinate of the starting point (in dots)
y1	The y1-coordinate of the starting point (in dots)
x2	The x2-coordinate of the ending point (in dots)
y2	The y2-coordinate of the ending point (in dots)
thickness	Thickness of diagonal

**Note:**

200 DPI : 1 mm = 8 dots

300 DPI : 1 mm = 12 dots

### Example

Sample code	Result
<pre>SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS DIAGONAL 50, 200, 200, 50, 16 DIAGONAL 50, 500, 500, 50, 8 PRINT 1,1</pre>	



## TEXT

### Description

This command prints text on label.

### Syntax

```
TEXT x,y, " font ",rotation,x-multiplication,y-multiplication,[alignment,]
" content "
```

Parameter	Description
x	The x-coordinate of the text
y	The y-coordinate of the text
font	Font name <hr/> <b>0</b> Monotype CG Triumvirate Bold Condensed, font width and height is stretchable <hr/> <b>1</b> 8 x 12 fixed pitch dot font <hr/> <b>2</b> 12 x 20 fixed pitch dot font <hr/> <b>3</b> 16 x 24 fixed pitch dot font <hr/> <b>4</b> 24 x 32 fixed pitch dot font <hr/> <b>5</b> 32 x 48 dot fixed pitch font <hr/> <b>6</b> 14 x 19 dot fixed pitch font OCR-B <hr/> <b>7</b> 21 x 27 dot fixed pitch font OCR-B <hr/> <b>8</b> 14 x 25 dot fixed pitch font OCR-A <hr/> <b>ROMAN.TTF</b> Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.
<i>Following fonts were supported since V6.80 EZ.</i>	
<b>1.EFT</b>	EPL2 font 1
<b>2.EFT</b>	EPL2 font 2
<b>3.EFT</b>	EPL2 font 3
<b>4.EFT</b>	EPL2 font 4
<b>5.EFT</b>	EPL2 font 5
<b>A.FNT</b>	ZPL2 font A
<b>B.FNT</b>	ZPL2 font B
<b>D.FNT</b>	ZPL2 font D
<b>E8.FNT</b>	ZPL2 font E8
<b>F.FNT</b>	ZPL2 font F
<b>G.FNT</b>	ZPL2 font G
<b>H8.FNT</b>	ZPL2 font H8
<b>GS.FNT</b>	ZPL2 font GS



Parameter	Description
rotation	The rotation angle of text 0: No rotation 90: degrees, in clockwise direction 180: degrees, in clockwise direction 270: degrees, in clockwise direction
x-multiplication	Horizontal multiplication, up to 10x Available factors: 1~10 For "ROMAN.TTF" true type font, this parameter is ignored. For font "0", this parameter is used to specify the width (point) of true type font. 1 point=1/72 inch.
y-multiplication	Vertical multiplication, up to 10x Available factors: 1~10 For true type font, this parameter is used to specify the height (point) of true type font. 1 point=1/72 inch. For *.TTF font, x-multiplication and y-multiplication support floating value. (V6.91 EZ)
alignment	Optional. Specify the alignment of text. (V6.73 EZ) 0: Default (Left) 1: Left 2: Center 3: Right
content	Content of text string

**Note:**

- The internal font (font #1~#5) pitch between TSPL and TSPL2 is different.
- Font "0" and "ROMAN.TTF" internal True Type Fonts are available in TSPL2 language printers, but not TSPL language printers.
- If there is any double quote ("") within the text, please change it to \"[\"].
- If font "0" is used, the font width and font height is stretchable by x-multiplication and y-multiplication parameter. It is expressed by pt (point). 1 point=1/72inch.
- EPL2 and ZPL2 are emulating for Eltron® and Zebra® languages.

MODEL	MODEL Font Type									
	0	1	2	3	4	5	6	7	8	ROMAN.TTF
TSPL language printers	V	V	V	V	V	V				
TSPL2 language printers	V	V	V	V	V	V	V	V	V	



## Example

### Sample code

```
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10,"0",0,12,12,"TSPL 2"
TEXT 10,40,"0",0,8,8,"align left"
BAR 0,70,800,4
TEXT 10,110,"0",0,12,12,"FONT 0"
TEXT 10,160,"1",0,1,1,"FONT 1"
TEXT 10,210,"2",0,1,1,"FONT 2"
TEXT 10,260,"3",0,1,1,0,"FONT 3"
TEXT 10,310,"4",0,1,1,0,"FONT 4"
TEXT 10,360,"5",0,1,1,0,"FONT 5"
TEXT 10,410,"6",0,1,1,1,"FONT 6"
TEXT 10,460,"7",0,1,1,1,"FONT 7"
TEXT 10,510,"8",0,1,1,1,"FONT 8"
TEXT 10,560,"ROMAN.TTF",0,12,12,"FONT ROMAN.TTF"

TEXT 400,10,"0",0,12,12,2,"EPL 2"
TEXT 400,40,"0",0,8,8,2,"align center"
TEXT 400,110,"1.EFT",0,1,1,2,"FONT 1"
TEXT 400,160,"2.EFT",0,1,1,2,"FONT 2"
TEXT 400,210,"3.EFT",0,1,1,2,"FONT 3"
TEXT 400,260,"4.EFT",0,1,1,2,"FONT 4"
TEXT 400,310,"5.EFT",0,1,1,2,"FONT 5"

TEXT 800,10,"0",0,12,12,3,"ZPL 2"
TEXT 800,40,"0",0,8,8,3,"align right"
TEXT 800,110,"A.FNT",0,1,1,3,"FONT A"
TEXT 800,160,"B.FNT",0,1,1,3,"FONT B"
TEXT 800,210,"D.FNT",0,1,1,3,"FONT D"
TEXT 800,260,"E8.FNT",0,1,1,3,"FONT E8"
TEXT 800,310,"F.FNT",0,1,1,3,"FONT F"
TEXT 800,360,"G.FNT",0,1,1,3,"FONT G"
TEXT 800,410,"H8.FNT",0,1,1,3,"FONT H8"
TEXT 800,460,"GS.FNT",0,1,1,3,"ABCDEF"
PRINT 1
```



## Result

**TSPL 2**  
align left**EPL 2**  
align center**ZPL 2**  
align right**FONT 0**

FONT 1

FONT 2

FONT 3

FONT 4

**FONT 5**  
FONT 6

FONT 7

FONT 8

**FONT ROMAN.TTF**

FONT 1

FONT 2

FONT 3

FONT 4

**FONT 5**

FONT 6

FONT 7

FONT 8

FONT 9

FONT A

FONT B

FONT C

FONT D

FONT E

FONT F

FONT G

FONT H

FONT I

FONT J

FONT K

FONT L

FONT M

FONT N

FONT O

FONT P

FONT Q

FONT R

FONT S

FONT T

FONT U

FONT V

FONT W

FONT X

FONT Y

FONT Z

© ® ™ UL SB ®



## BLOCK

### Description

This command prints paragraph on label.

### Syntax

```
BLOCK x,y,width,height, "font",rotation,x-multiplication,y-multiplication,[space,]align,]fit,] "content"
```

Parameter	Description
x	The x-coordinate of the text
y	The y-coordinate of the text
width	The width of block for the paragraph in dots
height	The height of block for the paragraph in dots
font	Font name <ul style="list-style-type: none"> <li><b>0</b> Monotype CG Triumvirate Bold Condensed, font width and height is stretchable</li> <li><b>1</b> 8 x 12 fixed pitch dot font</li> <li><b>2</b> 12 x 20 fixed pitch dot font</li> <li><b>3</b> 16 x 24 fixed pitch dot font</li> <li><b>4</b> 24 x 32 fixed pitch dot font</li> <li><b>5</b> 32 x 48 dot fixed pitch font</li> <li><b>6</b> 14 x 19 dot fixed pitch font OCR-B</li> <li><b>7</b> 21 x 27 dot fixed pitch font OCR-B</li> <li><b>8</b> 14 x 25 dot fixed pitch font OCR-A</li> <li><b>ROMAN.TTF</b> Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.</li> </ul>
<i>Following fonts were supported since V6.80 EZ.</i>	
<b>1.EFT</b>	EPL2 font 1
<b>2.EFT</b>	EPL2 font 2
<b>3.EFT</b>	EPL2 font 3
<b>4.EFT</b>	EPL2 font 4
<b>5.EFT</b>	EPL2 font 5
<b>A.FNT</b>	ZPL2 font A
<b>B.FNT</b>	ZPL2 font B
<b>D.FNT</b>	ZPL2 font D
<b>E8.FNT</b>	ZPL2 font E8
<b>F.FNT</b>	ZPL2 font F
<b>G.FNT</b>	ZPL2 font G
<b>H8.FNT</b>	ZPL2 font H8
<b>GS.FNT</b>	ZPL2 font GS



Parameter	Description
rotation	The rotation angle of text 0: No rotation 90: degrees, in clockwise direction 180: degrees, in clockwise direction 270: degrees, in clockwise direction
x-multiplication	Horizontal multiplication, up to 10x Available factors: 1~10 For "ROMAN.TTF" true type font, this parameter is ignored. For font "0", this parameter is used to specify the width (point) of true type font. 1 point=1/72 inch.
y-multiplication	Vertical multiplication, up to 10x Available factors: 1~10 For true type font, this parameter is used to specify the height (point) of true type font. 1 point=1/72 inch. For *.TTF font, x-multiplication and y-multiplication support floating value. (V6.91 EZ)
[space]	Add or delete the space between lines (in dots)
[align]	Text alignment. (V6.73 EZ) 0: default (Left) 1: Left 2: Center 3: Right
[fit]	Shrink the text so that it fits in the block (VA1.97) 0: No shrink (default) 1: Shrink
content	Data in block. The maximum data length is 4092 bytes.

**Note:**

- The internal font (font #1~#5) pitch between TSPL and TSPL2 is different.
- Font "0" and "ROMAN.TTF" internal True Type Fonts are available in TSPL2 language printers, but not TSPL language printers.
- If there is any double quote ("") within the text, please change it to \"[\"].
- If font "0" is used, the font width and font height is stretchable by x-multiplication and y-multiplication parameter. It is expressed by pt (point). 1 point=1/72inch.
- \[R] means carriage return character 0x0D.
- \[L] means line feed character 0x0A.
- This command has been supported since V6.91 EZ and later firmware.
- EPL2 and ZPL2 are for emulating Eltron® and Zebra® languages.



## Example

Sample code	Result
<pre> SIZE 4,0.5 GAP 0,0 DIRECTION 1 CLS BOX 10,10,800,100,2 BLOCK 15,15,790,90, "0",0,8,8,"We stand behind our products with one of the most comprehensive support programs in the Auto-ID industry." PRINT 1 CLS BOX 10,10,800,100,2 BLOCK 15,15,790,90,"0",0,8,8,20,2,"We stand behind our products with one of the most comprehensive support programs in the Auto-ID industry." PRINT 1 </pre>	<p>We stand behind our products with one of the most comprehensive support programs in the Auto-ID industry.</p> <p>We stand behind our products with one of the most comprehensive support programs in the Auto-ID industry.</p>
Sample Code for [fit] Parameter	Result
<pre> DATA\$ = "By more than a 2-1 ratio, lawmakers in West Virginia's House of Delegates have approved a bill that would allow gun owners to carry concealed handguns without a permit. The only concealed-carry permits would be for people who are 18-21 years old. Urging her colleagues to approve the bill, its 19-year-old sponsor, Delegate Saira Blair, " SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS BLOCK 20,20,500,170,"0",0,10,10,0,0,1,DATA\$ BOX 20,20,500+20,170+20,2 PRINT 1  SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS BLOCK 20,20,500,170,"0",0,10,10,0,0,0,DATA\$ BOX 20,20,500+20,170+20,2 PRINT 1 </pre>	<p><b>By more than a 2-1 ratio, lawmakers in West Virginia's House of Delegates have approved a bill that would allow gun owners to carry concealed handguns without a permit. The only concealed-carry permits would be for people who are 18-21 years old. Urging her colleagues to approve the bill, its 19-year-old sponsor, Delegate Saira Blair,</b></p> <p>By more than a 2-1 ratio, lawmakers in West Virginia's House of Delegates have approved a bill that would allow gun owners to carry concealed handguns without a permit. The only concealed-carry permits would be for people who are 18-21 years old. Urging her colleagues to approve the bill, its 19-year-old sponsor, Delegate Saira Blair,</p>



### 3. STATUS POLLING AND IMMEDIATE COMMANDS

These commands support RS-232, USB and Ethernet.

**<ESC>!?**

#### Description

This command obtains the printer status at any time, even in the event of printer error. An inquiry request is solicited by sending an <ESC> (ASCII 27, escape character) as the beginning control character to the printer. A one byte character is returned, flagging the printer status. A 0 signifies the printer is ready to print labels.

#### Syntax

**<ESC>!?**

Hex Receive	Printer Status
00	Normal
01	Head opened
02	Paper Jam
03	Paper Jam and head opened
04	Out of paper
05	Out of paper and head opened
08	Out of ribbon
09	Out of ribbon and head opened
0A	Out of ribbon and paper jam
0B	Out of ribbon, paper jam and head opened
0C	Out of ribbon and out of paper
0D	Out of ribbon, out of paper and head opened
10	Pause
20	Printing
80	Other error

#### See Also

**<ESC>!S**



## <ESC>!C

### Description

This command restarts the printer and omits to run AUTO.BAS. The beginning of the command is an ESCAPE character (ASCII 27).

### Syntax

<ESC>!C

#### Note:

- When printer receives this command, printer will restart itself no matter AUTO.BAS exists or not.
- This command has been supported since V5.23 EZ and later firmware.

### See Also

<ESC>!Q



## <ESC>!D

### Description

This command is used to disable immediate command, ex. <ESC>!R <RSC>!? <ESC>!C and so on, which is starting by <ESC>!.. The beginning of the command is an ESCAPE character (ASCII 27).

### Syntax

<ESC>!D

**Note:**

This command has been supported since V6.61 EZ and later firmware.

### See Also

~!D



## <ESC>!0

### Description

This command is using to cancel the PAUSE status of printer. The beginning of the command is an ESCAPE character (ASCII 27).

### Syntax

<ESC>!0

#### Note:

This command has been supported since V6.93 EZ and later firmware.

### See Also

<ESC>!P



## <ESC>!P

### Description

This command is using to PAUSE the printer. The beginning of the command is an ESCAPE character (ASCII 27).

### Syntax

<ESC>!P

#### Note:

This command has been supported since V6.93 EZ and later firmware.

### See Also

<ESC>!O



## <ESC>!Q

### Description

This command restarts the printer and omits to run AUTO.BAS. The beginning of the command is an ESCAPE character (ASCII 27).

### Syntax

<ESC>!Q

#### Note:

- If there is no AUTO.BAS inside the printer, the printer will not restart itself.
- This command has been supported since V6.72 EZ and later firmware.

### See Also

<ESC>!C



**<ESC>!R**

## Description

This command resets the printer. The beginning of the command is an ESCAPE character (ASCII 27). The files downloaded in memory will be deleted. This command cannot be sent in dump mode.

## Syntax

**<ESC>!R**

## See Also

**<ESC>!?**



## <ESC>!S

### Description

This command obtains the printer status at any time, even in the event of printer error. An inquiry request is solicited by sending an <ESC> (ASCII 27, escape character) as the beginning control character to the printer. 8 bytes will be returned, flagging the printer status.

### Syntax

<ESC>!S

**Note:**

**This command has been supported since V6.29 EZ and later firmware.**

### Response Format

<STX>[4-byte status]<ETX><CR><LF>

Status Byte #1: message											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	ASCII	Char	Meaning
0	1	0	0	0	0	0	0	40	64	@	Normal
0	1	1	0	0	0	0	0	60	96	`	Pause
0	1	0	0	0	0	1	0	42	66	B	Backing label
0	1	0	0	0	0	1	1	43	67	C	Cutting
0	1	0	0	0	1	0	1	45	69	E	Printer error
0	1	0	0	0	1	1	0	46	70	F	Form feed
0	1	0	0	1	0	1	1	4B	75	K	Waiting to press print key
0	1	0	0	1	1	0	0	4C	76	L	Waiting to take label
0	1	0	1	0	0	0	0	50	80	P	Printing batch
0	1	0	1	0	1	1	1	57	87	W	Imaging

Status Byte #2: warning											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	ASCII	Char	Meaning
0	1	0	0	0	0	0	0	40	64	@	Normal
0	1	0	0	0	0	0	1	41	65	A	Paper low (since A2.08 EZD)
0	1	0	0	0	0	1	0	42	66	B	Ribbon low (since A2.08 EZD)
0	1	0	0	0	1	0	0	44	68	D	Reversed
0	1	0	0	1	0	0	0	48	72	H	Receive buffer full
0	1	1	0	0	0	0	0	60	96	`	Reversed



## Response Format

**Status Byte #3: error**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	ASCII	Char	Meaning
0	1	0	0	0	0	0	0	40	64	@	Normal
0	1	0	0	0	0	0	1	41	65	A	Print head overheat
0	1	0	0	0	0	1	0	42	66	B	Stepping motor overheat
0	1	0	0	0	1	0	0	44	68	D	Print head error (since V7.01 EZ)
0	1	0	0	1	0	0	0	48	72	H	Cutter jam
0	1	0	1	0	0	0	0	50	96	P	Insufficient memory

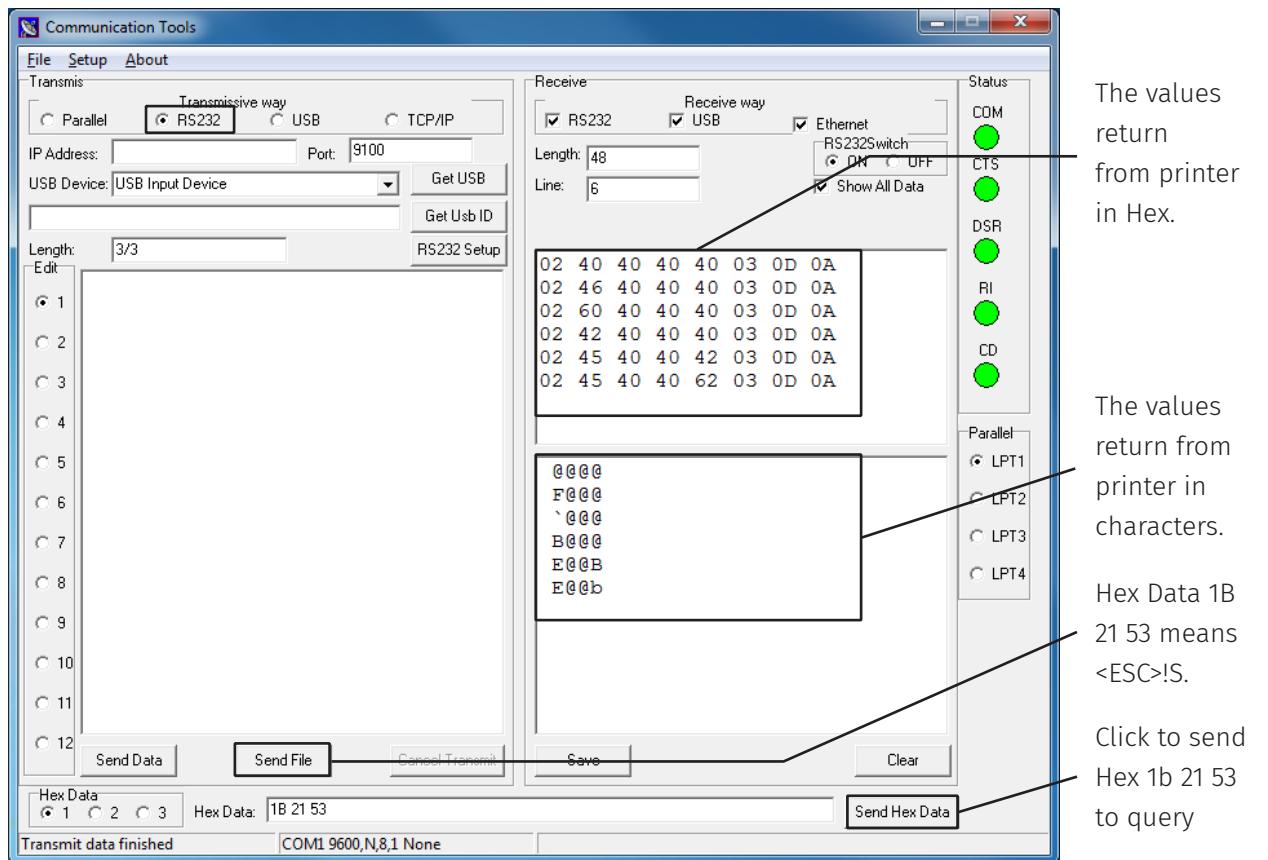
**Status Byte #4: error**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	ASCII	Char	Meaning
0	1	0	0	0	0	0	0	40	64	@	Normal
0	1	0	0	0	0	0	1	41	65	A	Paper empty
0	1	0	0	0	0	1	0	42	66	B	Paper jam
0	1	0	0	0	1	0	0	44	68	D	Ribbon empty
0	1	0	0	0	0	0	0	48	72	H	Ribbon jam
0	1	0	1	0	0	0	0	60	96	`	Print head open



## Example

Test <ESC>!S by CommTool via RS-232 port.



## Result

1	02 40 40 40 03 0D 0A
2	02 46 40 40 03 0D 0A
3	02 60 40 40 03 0D 0A
	02 42 40 40 03 0D 0A
	02 45 40 40 42 03 0D 0A
	02 45 40 40 62 03 0D 0A

4	@@@@
	F@@@
	`@@@
	B@@@
	E@@B
	E@@b

Item	Meaning
1	The start character of returned value.
2	The 4-byte status in Hex.
3	The end characters of returned value.
4	4-byte status in characters. @@@@: The printer is normal for use. F@@@: The printer is feeding label. `@@@: Printer is in PAUSE mode. B@@@: The printer is backing label. E@@B: Printer is in error "Paper Jam". E@@b: Printer is in error "Paper Jam" & "Head open".

## See Also

<ESC>!?



**<ESC>!F**

## Description

This command is using to feed a label. This function is the same as to press the FEED button. The beginning of the command is an ESCAPE character (ASCII 27).

## Syntax

**<ESC>!F**

**Note:**

This command has been supported since V7.00 EZ and later firmware.



**<ESC>!.**

## Description

This command can cancel all printing files. The beginning of the command is an ESCAPE character (ASCII 27).

## Syntax

**<ESC>!.**

### Note:

This command has been supported since V7.00 EZ and later firmware.

**~!@**

## Description

This command inquires the mileage of the printer. The integer part of mileage is returned (the decimal part of mileage is not return) to the PC in ASCII characters. The ending character of mileage is 0x0D.

## Syntax

**~!@**

## Example

**~!@**

**~!A**

## Description

This command inquires the free memory of the printer. The number of bytes of free memory is returned in decimal digits, with 0x0d as ending code of PC.

## Syntax

**~!A**

## Example

**~!A**

## See Also

**FILES**

**~!C**

## Description

This command inquires the presence of Real Time Clock. One byte is return from the printer, indicating whether or not the RTC is installed. This command is only for the firmware before V6.xx.

## Syntax

**~!C**

Return value	Description
0	RTC is not installed.
1	RTC is installed.

## Example

**~!C**

**~!D**

## Description

This command enters the printer into DUMP mode. In DUMP mode, the printer outputs code directly without interpretation.

## Syntax

**~!D**

## Example

**~!D**

**~!E**

## Description

This command is used to enable immediate command, ex. <ESC>!R <RSC>!? <ESC>!C and so on, which is starting by <ESC>!.

## Syntax

**~!E****Note:**

This command has been supported since V6.61 EZ and later firmware.

## Example

**~!E**

## See also

&lt;ESC&gt;!D

**~!F**

## Description

This command inquires all about files resident in the printer memory, and fonts installed in the memory module. The filename are returned in ASCII characters. Each file name ends with 0x0D. The ending character is 0x1A. Entering this command multiple times will cycle through the files resident on memory.

## Syntax

**~!F**

## Example

**~!F**

## See also

**FILES**

**~!I**

## Description

The command inquires the code page and country setting of the printer.

## Syntax

**~!I**

The returned information is given in the following format:

**code page, country code**

ex: 8 bit : 437, 001  
7 bit: USA, 001

Regarding the code pages and country codes supported by the printer, please refer to the **CODEPAGE** and **COUNTRY** command respectively.

## Example

**~!I**

## See Also

**COUNTRY, CODEPAGE**

**~!T**

## Description

This command inquires the model name and number of the printer. This information is returned in ASCII characters.

## Syntax

**~!T**

## Example

**~!T**



## <ESC> Y

### Description

This command is used to enable line mode (from EZPL to CPCL) for EZC printer.

### Syntax

<ESC> Y

### Example

<ESC> Y

### See Also

<ESC> Z



## <ESC> Z

### Description

This command is used to disable line mode (from CPCL to EZPL) for EZC printer.

### Syntax

<ESC> Z

### Example

<ESC> Z

### See Also

<ESC> Y



## 4. MESSAGE TRANSLATION PROTOCOLS

**~#**

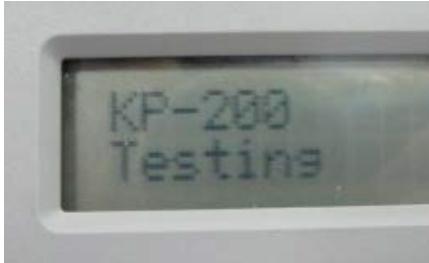
### Description

The beginning identifier (~#) of the prompt message is sent from the printer to the KP-200 portable keyboard. The ending identifier is ~&. @0 following the ending identifier ~& is used to instruct keyboard to display the prompt in the first line of LCD display. @1 following the ending identifier ~& is used to instruct keyboard to display the prompt in the first line of LCD display. If @0 or @1 are not present, prompt string will be displayed in first line of LCD and input data will be displayed in second line of LCD.

### Syntax

**~#Prompt~&[@0]  
~#Prompt~&[@1]**

### Example

Sample code	Result
<b>DOWNLOAD "A.BAS" OUT "~#KP-200~&amp;@0" OUT "~#Testing~&amp;@1" EOP A</b>	

### See Also

**INPUT, OUT**



## 5. COMMANDS FOR WINDOWS DRIVER

**!B**

### Description

This command stores bitmap image data in the memory. Behind the nnn is the bitmap data.

### Syntax

**!Bnnn**

Parameter	Description
nnn	The number of bytes of image data sent from PC to printer, expressed in 3 decimal digits.

### Example

**!B100**

### See Also

**BITMAP**



!J

## Description

This command prints bitmap data at the specified position (in y-direction).

## Syntax

**!Jnnnn**

Parameter	Description
nnn	Print image at the specified position in y-direction. The position is expressed in 4 decimal digits.

## Example

**!J0100**

## See Also

**FEED**

**!N**

## Description

This command prints a specified number of labels.

## Syntax

**!Nnnn**

Parameter	Description
nnn	Specifies the number of copies to be printed.

## Example

**!N001**



## 6. FILE MANAGEMENT COMMANDS

### !DOWNLOAD

#### Description

“DOWNLOAD” is a header of the file that is to be saved in the printer’s memory. The downloaded files can be divided into two categories: program files and data files (including text data files, PCX graphic files and bitmap font files) The detailed descriptions regarding the download syntax for different files are as follows:

#### Maximum numbers of file saved in DRAM:

50 files for TSPL/TSPL2 language printers

#### Maximum numbers of file saved in Flash memory:

50 files for TSPL language printers

256 files for TSPL2 language printers

If “AUTO.BAS” exists in the printer memory, it will be automatically executed upon printer startup. To disable the auto execution function, please follow the procedures below.

#### Ignore AUTO.BAS

##### For two buttons desktop printer series

Hold down the PAUSE and FEED buttons and turn on the printer power. Do not release the buttons until the three LEDs flash in turn. Printer will Ignore AUTO.BAS and initialize the printer.

##### For one button desktop printer series

Hold the FEED key and power on the switch. Release the FEED key while LED becomes solid green to prevent the printer from running “AUTO.BAS”.

The LED color will be changed as following pattern:

##### \*For firmware version before V3.37 printer:

Orange → red (5 blinks) → orange (5 blinks) → green (5 blinks) → solid green

##### \*For firmware version after V3.37:

Orange → red (5 blinks) → orange (5 blinks) → green (5 blinks) → green and orange (5 blinks) → red and orange (5 blinks) → solid green

##### For three buttons industrial printer series

Hold the FEED key and power on the switch. The ERROR LED will be on. Printer is now ready to use.

##### For six or two buttons industrial printer series

Hold the PAUSE and FEED keys and power on the switch. “AUTO.BAS” will not be executed after printer initialization, and will now be ready for use.

Alternatively, hold the PAUSE key and power on the switch. After sensor calibration, the “AUTO.BAS” will not be executed. Printer is now ready for use.



## Syntax

1. Download a program file:

**DOWNLOAD [n,] “FILENAME.BAS”**

Parameter	Description
n	Specify memory used to save downloaded files. <i>N is ignored:</i> Download files to DRAM only. If you would like to save the files from DRAM to Flash memory before turning off power, issue the MOVE command to printer. F: Download files to main board flash memory. E: Download files to expansion memory module.
FILENAME.BAS	The filename resident in printer memory.

**Note:**

- **Filenames are case sensitive.**
- **File extensions must be “.BAS”**
- **Filenames must be in 8.3 format.**
- **It should use with EOP command.**
- **If memory is not specified, all files will be downloaded to DRAM.**
- **The priority of AUTO.BAS in each memory device: A. DRAM > FLASH > CARD (Ext. FLASH) if firmware is before V6.80EZ. B. DRAM > CARD (Ext. FLASH) > FLASH if firmware is after V6.80EZ (include).**
- **No Battery is used to back up files in DRAM. Which will be lost in the event printer power is lost.**

2. Download a data file:

**DOWNLOAD [n,] “FILENAME”,DATA SIZE,DATA CONTENT...**

Parameter	Description
n	Specify the memory location to save downloaded files. <i>N is ignored:</i> Download files to DRAM only. If you would like to save the files from DRAM to Flash memory before turning off power, issue the MOVE command to printer. F: Download files to main board flash memory. E: Download files to expansion memory module.
FILENAME	The name of data file that will remain resident in the printer memory (case sensitive).
DATA SIZE	The actual size in bytes of the data file (without header)
DATA CONTENT	The data which will be downloaded into printer.

**Note:**

- **For text data files, CR (carriage return) 0x0D and LF (Line Feed) 0x0A is the separator of data.**
- **If memory is not specified, all files will be downloaded to DRAM.**
- **No Battery is used to back up files in DRAM. Which will be lost in the event printer power is lost.**
- **When writing a download program, “DOWNLOAD” header must be placed in the beginning of file, and “EOP” must be placed at the end of program.**
- **To run the program, call the main filename without BAS extension or use RUN command to start the download program.**



## Example

Sample code (The example program listed below will download to printer SDRAM.)

```
DOWNLOAD "EXAMPLE.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
SET TEAR ON
CLS
TEXT 100,100, "3",0,1,1, "EXAMPLE PROGRAM"
PRINT 1
EOP
```

## See Also

EOP, RUN, PUTBMP, PUTPCX, INPUT, FILES, ~!F



## EOP

### Description

End of program. To declare the start and end of BASIC language commands used in a program, DOWNLOAD “FILENAME.BAS” must be added in the first line of the program, and “EOP” statement at the last line of program.

### Syntax

**EOP**

### Example

**Sample code (The example program listed below will download to printer SDRAM.)**

```
DOWNLOAD "DEMO.BAS"  
SIZE 4,4  
GAP 0,0  
DIRECTION 1  
SET TEAR ON  
CLS  
TEXT 100,100, "3",0,1,1, "DEMO PROGRAM"  
PRINT 1  
EOP
```

### See Also

DOWNLOAD, INPUT, FILES, ~!F



## FILES

### Description

This command prints out the total memory size, available memory size and files lists (or lists the files through RS-232) in the printer memory (both FLASH memory and DRAM).

### Syntax

**FILES**

### Example

Sample code	Result
<b>FILES</b>	<pre>----- DRAM FILE (0 FILES) ----- PHYSICAL     8192 KBYTES AVAILABLE    256 KBYTES ----- FLASH FILE (0 FILES) ----- PHYSICAL     4096 KBYTES AVAILABLE    2560 KBYTES</pre>

### See Also

**~!F, KILL**



## KILL

### Description

This command deletes a file in the printer memory. The wild card (\*) will delete all files resident in specified DRAM or FLASH memory.

### Syntax

**KILL [n], “FILENAME”**

Parameter	Description
n	Specify the memory location that files will be deleted. <i>N</i> is ignored: Kill files saved in DRAM. F: Kill files from main board flash memory. E: Kill files from expansion memory module.
FILENAME	The name of data file that will delete in the printer memory (case sensitive)

#### Note:

- If optional parameter n is not specified, firmware will delete the file in DRAM.
- Syntax example
  - KILL “FILENAME”** : Delete the specify file in DRAM.
  - KILL “\*.PCX”** : Delete all PCX files in DRAM.
  - KILL “\* ”** : Delete all files in DRAM.
  - KILL F, “FILENAME”** : Delete the specify file in FLASH.
  - KILL E, “ \*.PCX ”** : Delete all PCX file in extension memory card.
- For TSPL printers, please send MOVE command to printer after sending KILL command.

Model	Support		
	KILL “*”	KILL “*” MOVE	KILL F, “*”
TSPL programming printer	V	V	
TSPL2 programming printer	V		V



## Example

Users can use printer SELFTEST utility to list printer configurations and files saved in the printer memory, or use the FILES command to print the downloaded file list in printer. Follow the steps below to delete files in the printer memory via parallel port connection.

```
C:\>COPY CON LPT1<ENTER>
FILES<ENTER>
<CTRL><Z><ENTER>
C:\>COPY CON LPT1<ENTER>
KILL "DEMO.BAS" <ENTER>
<CTRL><Z><ENTER>
C:\>COPY CON LPT1<ENTER>
FILES<ENTER>
<CTRL><Z><ENTER>
```

**Note:**

<ENTER> stands for PC keyboard “ENTER” key. <CTRL><Z> means to hold PC keyboard “CTRL” key then press the PC keyboard <Z> key

## See Also

~!F, FILES



## MOVE

### Description

This command moves downloaded files from DRAM to FLASH memory.

### Syntax

**MOVE**

### See Also

DOWNLOAD, EOP



## RUN

### Description

This command executes a program resident in the printer memory. It is available for TSPL2 language printers only.

### Syntax

**RUN “FILENAME.BAS”**

**Note:**

- This command can be replaced to filename that without typing “.BAS”.

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,4 GAP 0,0 DIRECTION 1 SET TEAR ON CLS TEXT 100,100, "3",0,1,1, "DEMO PROGRAM" PRINT 1 EOP RUN "DEMO.BAS"</pre>	
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,4 GAP 0,0 DIRECTION 1 SET TEAR ON CLS TEXT 100,100, "3",0,1,1, "DEMO PROGRAM" PRINT 1 EOP DEMO</pre>	<b>DEMO PROGRAM</b>

### See Also

DOWNLOAD, EOP



## 7. BASIC COMMANDS AND FUNCTIONS

### ABS()

#### Description

This function returns the absolute value of an integer, floating point or variable.

#### Syntax

**ABS (VARIABLE)**

#### Example

Sample code	Result
<code>DOWNLOAD "TEST.BAS" SIZE 4,4 GAP 0,0 DIRECTION 1 SET TEAR ON CLS A=ABS(-100) B=ABS(-50.98) C=-99.99 TEXT 100,100, "3",0,1,1,STR\$(A) TEXT 100,150, "3",0,1,1,STR\$(B) TEXT 100,200, "3",0,1,1,STR\$(ABSI) PRINT 1 EOP RUN "TEST.BAS"</code>	<b>100</b>
	<b>50.98</b>
	<b>99.99</b>

#### See Also

DOWNLOAD, EOP



## ASC()

### Description

This function returns the ASCII code of the character.

### Syntax

ASC (" A ")

### Example

Sample code	Result
<pre>DOWNLOAD "TEST.BAS" SIZE 4,4 GAP 0,0 DIRECTION 1 SET TEAR ON CLS CODE1=ASC(" A ") TEXT 100,100, " 3 ",0,1,1,STR\$(CODE1) PRINT 1 EOP RUN "TEST.BAS"</pre>	<b>65</b>

### See Also

DOWNLOAD, EOP, STR\$()



## CHR\$( )

### Description

This function returns the character with the specified ASCII code.

### Syntax

**CHR\$(n)**

Parameter	Description
n	The ASCII code

### Example

Sample code	Result
<pre>DOWNLOAD "TEST.BAS" SIZE 4,4 GAP 0,0 DIRECTION 1 SET TEAR ON CLS A=75 WORD\$=CHR\$(A) TEXT 100,100, "3",0,1,1,WORD\$ PRINT 1 EOP RUN "TEST.BAS"</pre>	K

### See Also

DOWNLOAD, EOP, STR\$( ), ASC\$( )



## XOR\$( )

### Description

This command can encode the original data to a new data by logic XOR.

### Syntax

**XOR\$(data\$,password\$)**

Parameter	Description
data\$	The original data needs to be encoded by password\$.
password\$	This parameter will be used to create the new data.

**Note:**

This command has been supported since V6.38 EZ and later firmware.

### Example

Sample code	Result
<pre>data\$="1234" password\$="ABCD" encoded\$=XOR\$(data\$,password\$) deconded\$=XOR\$(encoded\$,password\$) SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10,"3",0,1,1, "Encoded data: "+encoded\$ TEXT 10,60, "3",0,1,1, "Decoded data: "+deconded\$ PRINT 1</pre>	<p>Encoded data: pppp Decoded data: 1234</p>



## END

### Description

This command states the end of program.

### Syntax

**END**

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,2 GAP 0,0 DIRECTION 1 CLS TEXT 200,60, "4",0,1,1, "END COMMAND TEST" X=300 Y=200 X1=500 Y1=400 GOSUB DR_LINE PRINT 1 END  :DR_LINE FOR I=1 TO 100 STEP 10 BOX X+I,Y+I,X1-I,Y1-I,5 NEXT RETURN EOP DEMO</pre>	<p>END COMMAND TEST</p>

### See Also

DOWNLOAD, EOP, GOSUB



## EOF()

### Description

This function is used to detect an opened download file to see whether it has reached the end of file.

### Syntax

**EOF (File Handle)**

Parameter	Description
File handle	Either 0 or 1

Return value	Description
None-zero	End of file
0	Not end of file

### Example

Sample code	Result
<pre> DOWNLOAD "DATA",16,COMPUTER 2000  DOWNLOAD "DEMO.BAS" SIZE 3,3 GAP 0.0,0 DIRECTION 1 CLS OPEN "DATA",0 SEEK 0,0 Y=110 TEXT 10,10, "3",0,1,1, "*****EOF TEST*****" :A Temp\$="" READ 0,ITEM\$,P TEXT 10,Y,"2",0,1,1,ITEM\$+"\$" +STR\$(P)+"[EOF(0)="+ STR\$(EOF(0))+"]" BARCODE 10,Y+25,"39",40,1,0,2,4,"PRICE-"+STR\$(P) Y=Y+100 IF EOF(0)=0 THEN GOTO A PRINT 1 EOP DEMO </pre>	<p>*****EOF TEST*****</p> <p>COMPUTER\$2000[EOF(0)=1]</p>  <p>PRICE-2000</p>

### See Also

DOWNLOAD, EOP, OPEN, READ, SEEK



## OPEN

### Description

This command opens a downloaded file and establishes the file handle. Up to two file handles are supported, thus only up to two files can be opened simultaneously. The file to be opened should be downloaded prior to using this command. When opening a file, the firmware will search automatically to see if the file exists in the on board flash memory or extended memory card. \*Since V6.37 EZ, if the file doesn't exist, the printer will create this file in the onboard FLASH.

### Syntax

**OPEN [memory ID,] "filename",file handle**

Parameter	Description	
[memory ID]	Optional. Open the file in specific memory device. *Since V6.68 EZ.	
ID	Memory device	
Omitted	DRAM	
F	FLASH	
E	CARD	
filename	The file downloaded in the printer memory	
file handle	Either 0 or 1	



## Example

Sample code	Result
<pre>DOWNLOAD "DATA.DAT",18,Open file in DRAM. DOWNLOAD F, "DATA.DAT",19,Open file in FLASH. DOWNLOAD "TEST.BAS" data1\$="" data2\$="" data3\$="" OPEN "DATA.DAT",0 READ 0,data1\$ CLOSE 0 OPEN F, "DATA.DAT",0 READ 0,data2\$ CLOSE 0 KILL F, "*" OPEN "NEW.DAT",0 SEEK 0,0 WRITE 0, "Auto create a new file in FLASH." SEEK 0,0 READ 0,data3\$ CLOSE 0 SIZE 4,1 GAP 0,0 CLS TEXT 10,10,"3",0,1,1,data1\$ TEXT 10,60,"3",0,1,1,data2\$ TEXT 10,110,"3",0,1,1,data3\$ PRINT 1 EOP TEST</pre>	<pre>Open file in DRAM. Open file in FLASH. Auto create a new file in FLASH.</pre>

## See Also

DOWNLOAD, EOP, READ, WRITE, SEEK, CLOSE



## CLOSE

### Description

Close the file handle which is open by command OPEN.

### Syntax

**CLOSE file handle**

Parameter	Description
file handle	Either 0 or 1

### Example

See the example in command **OPEN**.



## WRITE

### Description

This command writes data to a downloaded data file. Two files can be open simultaneously, by virtue of printer support for two file handles.

### Syntax

**WRITE file handle,variables**

Parameter	Description
file handle	0 or 1
variables	string, integer or float point variable

### See Also

**READ, DOWNLOAD, EOP, OPEN, EOF, LOF, SEEK, FREAD\$()**



## READ

### Description

This command reads data from downloaded data file.

### Syntax

**READ file handle,variables**

Parameter	Description
File handle	0 or 1
variables	string, integer or float point variable

### Example

Sample code	Result
<pre> DOWNLOAD "DATA1",20,COMPUTER 2000 12 DOWNLOAD "DATA2",16,Mouse 900 93 DOWNLOAD "DEMO.BAS" SIZE 3,1 GAP 0,0 DIRECTION 1 I=0 Y=100 OPEN "DATA1",0 OPEN "DATA2",1 SEEK 0,0 SEEK 1,0 :Start CLS TEXT 10,10,"3",0,1,1,"*****READ COMMAND TEST*****" TEXT 10,50,"3",0,1,1,"OPEN-READ DATA"+STR\$(I+1) ITEM\$=""" READ I,ITEM\$,P,Q TEXT 10,Y, "2",0,1,1,ITEM\$+"\$" +STR\$(P) BARCODE 10,Y+25, "39 ",40,1,0,2,4, "PRICE* "+STR\$(Q)+ "=" +STR\$(P*Q) Y=Y+100 </pre>	<pre> *****READ COMMAND TEST***** OPEN-READ DATA3 \$900  PRICE*93=83700  *****READ COMMAND TEST***** OPEN-READ DATA2 Mouse\$900  PRICE*93=83700  *****READ COMMAND TEST***** OPEN-READ DATA1 COMPUTER\$2000  PRICE*12=24000 </pre>



```
PRINT 1
Y=100
IF I<=1 THEN
IF EOF(I)=1 THEN
I=I+1
GOTO Start
ELSE
GOTO Start
ENDIF
ELSE
END
ENDIF
EOP
DEMO
```

## See Also

DOWNLOAD, EOP, OPEN, EOF, LOF, SEEK, FREAD\$()



## SEEK

### Description

This command shifts the specified file pointer to a certain position.

### Syntax

**SEEK file handle,offset**

Parameter	Description
File handle	0 or 1
offset	the offset characters which are shifted to a new position

### Example

Sample code	Result
<pre> DOWNLOAD "DATA",12,1234567890 DOWNLOAD "TEST.BAS" SIZE 4,1.5 GAP 0,0 DIRECTION 1 REFERENCE 0,0 CLS OPEN "DATA",0 SEEK 0,4 READ 0,Num\$ TEXT 100,10,"3",0,1,1,"SEEK COMMAND TEST" BAR 100,40,300,4 TEXT 100,60,"3",0,1,1,"SHIFT 4 CHARACTERS" TEXT 100,110,"3",0,1,1,Num\$ BAR 100,140,300,4 SEEK 0,0 READ 0,Num\$ TEXT 100,160,"3",0,1,1,"SHIFT 0 CHARACTERS" TEXT 100,210,"3",0,1,1,Num\$ PRINT 1 EOP TEST </pre>	<p><b>SEEK COMMAND TEST</b></p> <hr/> <p><b>SHIFT 4 CHARACTERS</b></p> <hr/> <p><b>567890</b></p> <hr/> <p><b>SHIFT 0 CHARACTERS</b></p> <hr/> <p><b>1234567890</b></p>

### See Also

DOWNLOAD, EOP, OPEN, READ, EOF, LOF, FREAD\$()



## LOF()

### Description

This function returns the size of the specified file.

### Syntax

**LOF("FILENAME")**

Parameter	Description
FILENAME	The file downloaded in the printer memory.

### Example

Sample code	Result
<pre>DOWNLOAD "DATA1",10,1234567890 DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO DOWNLOAD "LofTest.BAS" SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS OPEN "DATA1",0 OPEN "DATA2",1 TEXT 10,20,"4",0,1,1,"LOF() FUNCTION TEST" J=LOF("DATA1") K=LOF("DATA2") TEXT 10,140,"3",0,1,1,"DATA1 IS: "+STR\$(J)+"Bytes" TEXT 10,200,"3",0,1,1,"DATA2 IS: "+STR\$(K)+"Bytes" PRINT 1 EOP LofTest</pre>	<p>LOF() FUNCTION TEST</p> <p>DATA1 IS: 10 Bytes</p> <p>DATA2 IS: 15 Bytes</p>

### See Also

DOWNLOAD, EOP, OPEN, READ, EOF, SEEK, FREAD\$()



## LOC()

### Description

This function returns the current read/write position within an open file.

### Syntax

LOF("FILENAME")

Parameter	Description
file handle	0 or 1

**Note:**

This command has been supported since V6.86 EZ and later firmware.

### Example

Sample code	Result
<pre>DOWNLOAD "DATA.DAT",30,12345678 12345678 12345678  DOWNLOAD "TEST.BAS" str1\$ = "" location = 0 OPEN "DATA.DAT",0 READ 0,str1\$ location = LOC(0) CLOSE 0 SIZE 4,1 GAP 0,0 CLS TEXT 10,10,"3",0,1,1,"str1\$: "+str1\$ TEXT 10,60,"3",0,1,1,"Location:"+STR\$(location) PRINT 1 EOP TEST</pre>	<pre>str1\$: 12345678 Location:10</pre>



## FREAD\$( )

### Description

This function reads a specified number of bytes of data from a file.

### Syntax

**FREAD\$ (file handle,byte)**

Parameter	Description
file handle	0 or 1
byte	Number of bytes to be read

### Example

Sample code	Result
<pre> DOWNLOAD "DATA1",10,1234567890 DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO DOWNLOAD "OPEN2.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 CLS OPEN "DATA1",0 OPEN "DATA2",1 SEEK 0,0 SEEK 1,0 Y\$=FREAD\$(0,6) Z\$=FREAD\$(1,6) TEXT 10,100,"3",0,1,1,"FREAD\$(0,6) IS: " +Y\$ TEXT 10,150,"3",0,1,1,"FREAD\$(1,6) IS: " +Z\$ PRINT 1 EOP OPEN2 </pre>	<pre> FREAD\$(0,6) IS: 123456 FREAD\$(1,6) IS: ABCDEF </pre>

### See Also

DOWNLOAD, EOP, OPEN, READ, EOF, LOF(), SEEK



## PUT

### Description

One byte is appended into file.

### Syntax

```
PUT file handle,var1$[, var2$][,var3$][, ...]
PUT file handle,var1[, var2][,var3][, ...]
PUT file handle,var1$[, var2$][,var3][, ...]
```

Parameter	Description
file handle	0 or 1
var\$	Data is a character
var	Data is ASCII value

**Note:**

This command has been supported since V6.91 EZ and later firmware.

### Example

Sample code	Result
<pre>DOWNLOAD "DATA1",10,1234567890 DOWNLOAD "TEST.BAS" str1\$ = "" str2\$ = "" OPEN "DATA1",0 SEEK 0,0 READ 0,str1\$ PUT 0,"a","B",49 SEEK 0,0 READ 0,str2\$ CLOSE 0  SIZE 4,0.5 GAP 0,0 CLS TEXT 10, 10,"3",0,1,1,"Original data in DATA1: "+str1\$ TEXT 10, 60,"3",0,1,1,"New data in Data1: "+str2\$ PRINT 1 EOP TEST</pre>	<pre>Original data in DATA1: 1234567890 New data in Data1: 1234567890aB1</pre>

### See Also

DOWNLOAD, EOP, OPEN, READ, EOF, LOF(), SEEK



## GET

### Description

Get one byte from file.

### Syntax

```
GET file handle, var1$[, var2$][, var3$][, ...]  
GET file handle, var1[, var2][, var3][, ...]  
GET file handle, var1$[, var2$][, var3][, ...]
```

Parameter	Description
file handle	0 or 1
var\$	Get a character
var	Get ASCII value

**Note:**

This command has been supported since V6.91 EZ and later firmware.

### Example

#### Sample code

```
DOWNLOAD "DATA1",10,1234567890  
DOWNLOAD "TEST.BAS"  
a$=""  
b$=""  
c=0  
d$=""  
e$=""  
OPEN "DATA1",0  
SEEK 0,0  
GET 0,a$,b$,c  
SEEK 0,0  
FOR I=1 TO 5  
GET 0,d$  
e$=e$+d$  
NEXT
```



```
SIZE 4,0.5
GAP 0,0
CLS
TEXT 10,10,"3",0,1,1,"The first 3 characters in DATA1: "+ a$+b$+
("+"+STR$(c)+"")
TEXT 10,60,"3",0,1,1,"The first 5 characters in DATA1: "+e$
PRINT 1
EOP
TEST
```

#### Result

```
The first 3 characters in DATA1: 12 (51)
The first 5 characters in DATA1: 12345
```

#### See Also

DOWNLOAD, EOP, OPEN, READ, EOF, LOF(), SEEK, PUT



## COPY

### Description

Copy the existed file from CARD to FLASH.

### Syntax

```
COPY [memory ID of source,] "filename of source",[memory ID of new file,] "new filename"
```

Parameter	Description	
memory ID of source	Optional.	
	<b>ID</b>	<b>Memory device</b>
	Omitted	DRAM
	F	FLASH
	E	CARD
source filename	The file in CARD which you want to copy to on board FLASH.	
memory ID of new file	Optional.	
	<b>ID</b>	<b>Memory device</b>
	Omitted	DRAM
	F	FLASH
new filename	The new filename you want to use in the on board FLASH.	

**Note:**

This command has been supported since V6.78 EZ and later firmware.

### Example

#### Sample code

```
DOWNLOAD "DATA_D.DAT",105,We stand behind our products with one of the most
comprehensive support programs in the Auto-ID industry.
DOWNLOAD "TEST.BAS"
KILL F,"*"
COPY "DATA_D.DAT",F,"DATA_F.DAT"
OPEN "DATA_F.DAT",0
SEEK 0,0
data$=FREAD$(0,LOF("DATA_F.DAT"))
CLOSE 0
```



```
SIZE 4,0.5
GAP 0,0
CLS
BOX 10,10,800,100,2
BLOCK 15,15,790,90,"0",0,8,8,20,2,data$
PRINT 1
EOP
TEST
```

#### Result

- 203 dpi

We stand behind our products with one of the most comprehensive support programs in  
the Auto-ID industry.

- 300 dpi

We stand behind our products with one of the most  
comprehensive support programs in the Auto-ID industry.

#### See Also

DOWNLOAD, EOP, OPEN, FREAD\$(), EOF, LOF(), SEEK, CLOSE



## FOR...NEXT LOOP

### Description

Loop is used to execute one or more lines of program repetitively. A loop counter value specifies the number of executions. Nested loops are allowed (up to 39 nested loops) in this printer. Jumping out in the middle of the FOR...NEXT loop is prohibited.

### Syntax

```
FOR variable = start TO end STEP increment  
    statement; start < end  
    [EXITFOR]
```

NEXT

Parameter	Description
variable	The variable name (up to 8 characters)
start	Integer or floating point numbers
end	Integer or floating point numbers
increment	Integer or floating point, positive or negative
EXITFOR	Exit for loop

### Example

#### Sample code

```
DOWNLOAD "TEST.BAS"  
SIZE 4,2.5  
GAP 0,0  
CLS  
FOR I=1 TO 10 STEP 1  
TEXT 100,10+30*(I-1),"3",0,1,1,STR$(I)  
NEXT  
  
FOR I=1 TO 1000 STEP 100  
TEXT 200,10+((I-1)/10)*3,"3",0,1,1,STR$(I)  
NEXT  
  
FOR I=110 TO 10 STEP -10  
TEXT 300,10+(ABS(I-110))*3,"3",0,1,1,STR$(I)  
NEXT
```

**Sample code**

```
FOR I=1 TO 5 STEP 0.5
IF I-INT(I)=0 THEN Y=10+60*(I-1) ELSE Y=Y+30
TEXT 400,Y,"3",0,1,1,STR$(I)
NEXT

PRINT 1
EOP
TEST
```

**Result**

1	1	110	1
2	101	100	1.5
3	201	90	2
4	301	80	2.5
5	401	70	3
6	501	60	3.5
7	601	50	4
8	701	40	4.5
9	801	30	5
10	901	20	10

**See Also**

DOWNLOAD, EOP



## WHILE...WEND

### Description

Executes a series of statements as long as a given condition is True. Nested loops are allowed (up to 39 nested loops) in this printer.

### Syntax

```
WHILE condition  
[statement]  
WEND
```

Parameter	Description
condition	Available relational operator: <, >, =, <=, >=, <> <i>*Relational operator &lt;&gt;, not equal, was supported since V5.10 EZ.</i>
Statement	One or more statements executed while condition is True.

#### Note:

This command has been supported since V5.10 EZ and later firmware.

### Example

Sample code	
<pre>DOWNLOAD "TEST.BAS" I=0 TOTAL=0 WHILE I&lt;100 I=I+1 TOTAL=TOTAL+I WEND SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " +STR\$(TOTAL) PRINT 1 EOP TEST</pre>	1+2+3+ ... + 100 = 5050



## Example

### Sample code

```
DOWNLOAD "TEST.BAS"
data$ = ""
SIZE 4,0.3
GAP 0,0
DIRECTION 1
INPUT "Data: ",data$
WHILE data$ <> "Quit"
CLS
TEXT 10,10, "3",0,1,1, "Data: "+data$
PRINT 1
INPUT "Data: ",data$
WEND
CLS
TEXT 10,10, "3",0,1,1, "Quit BAS"
PRINT 1
EOP
TEST
12345
67890
quit
Quit
```

Quit BAS  
Data: quit  
Data: 67890  
Data: 12345



## DO...LOOP

### Description

Repeats a block of statement while a condition is True.

### Syntax

DO

```
[statement]  
[EXITDO]  
[statement]
```

LOOP

DO WHILE condition

```
[statement]  
[EXITDO]  
[statement]
```

LOOP

DO UNTIL condition

```
[statement]  
[EXITDO]  
[statement]
```

LOOP

DO

```
[statement]  
[EXITDO]  
[statement]
```

LOOP WHILE condition

DO

```
[statement]  
[EXITDO]  
[statement]
```

LOOP UNTIL condition

Parameter	Description
condition	Available relational operator: <, >, =, <=, >=, <> <i>*Relational operator &lt;&gt;, not equal, was supported since V5.10 EZ.</i>
Statement	One or more statements executed while condition is True.
EXITDO	Exit loop

**Note:**

This command has been supported since V5.10 EZ and later firmware.



## Example

Sample code	Result
<pre>DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO I=I+1 TOTAL=TOTAL+I IF I=100 THEN EXITDO LOOP SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " + STR\$(TOTAL) PRINT 1 EOP TEST</pre>	$1+2+3+ \dots + 100 = 5050$
<pre>DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO WHILE I&lt;=100 TOTAL=TOTAL+I I=I+1 LOOP SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " + STR\$(TOTAL) PRINT 1 EOP TEST</pre>	$1+2+3+ \dots + 100 = 5050$



Sample code	Result
<pre>DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO UNTIL I&gt;100 TOTAL=TOTAL+I I=I+1 LOOP SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " + STR\$(TOTAL) PRINT 1 EOP TEST</pre>	$1+2+3+ \dots + 100 = 5050$
<pre>DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO TOTAL=TOTAL+I I=I+1 LOOP WHILE I&lt;101 SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 =" + STR\$(TOTAL) PRINT 1 EOP TEST</pre>	$1+2+3+ \dots + 100 = 5050$



Sample code	Result
<pre>DOWNLOAD "TEST.BAS" I=0 TOTAL = 0 DO TOTAL = TOTAL + I I=I+1 LOOP UNTIL I&gt;100 SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " + STR\$(TOTAL) PRINT 1 EOP TEST</pre>	1+2+3+ ... + 100 = 5050



## IF...THEN...ELSE...ENDIF LOOP

### Description

Use IF...THEN block to execute one or more statements conditionally. Either a single-line syntax or multiple-line “block” syntax can be used.

### Syntax

**IF condition THEN statement**

Note the single-line form of IF ...THEN does not use an ENDIF statement.

Or

```
IF condition THEN      (TSPL2 printers only)
  Statements
ENDIF
```

Or

```
IF condition THEN      (TSPL2 printers only)
  Statements
ELSE
  Statements
ENDIF
```

Or

```
IF condition 1 THEN      (TSPL2 printers only)
  Statement block 1
ELSEIF condition 2 THEN
  Statement block 2
  ...
ELSEIF condition n THEN
  Statement block n
ENDIF
```

#### Note:

The syntax of IF...THEN...ELSE requires that the command be typed in one single line in less than 255 characters.

Parameter	Description
condition	Available relational operator: <, >, =, <=, >=, <> *Relational operator <>, not equal, was supported since V5.10 EZ.
Statement	Only one statement is available in



## Example

Sample code	Result
<pre> DOWNLOAD "DEMO.BAS" SIZE 4,4 GAP 0,0 DIRECTION 1 CLS A=0 B=0 C=0 D=0 E=0 F=0 G=0 H=0 J=0 K=0 L=0 FOR I=1 TO 100 IF I-INT(I/1)*1=0 THEN A=A+I IF I-INT(I/2)*2=1 THEN B=B+I ELSE C=C+I IF I-INT(I/3)*3=0 THEN D=D+I ENDIF IF I-INT(I/5)*5=0 THEN E=E+I ELSE F=F+I ENDIF IF I-INT(I/7)*7=0 THEN G=G+I ELSEIF I-INT(I/17)*17=0 THEN H=H+I ELSEIF I-INT(I/27)*27=0 THEN J=J+I ELSEIF I-INT(I/37)*37=0 THEN K=K+I ELSE L=L+I ENDIF NEXT </pre>	$(1) 1+2+3+\dots+100=5050$ $(2) 1+3+5+\dots+99=2500$ $(3) 2+4+6+\dots+100=2550$ $(4) 3+6+9+\dots+99=1683$ $(5) 5+10+15+\dots+100=1050$ $(1)-(5)=4000$ $(6) 7+14+21+\dots+98=735$ $(7) 17+34+51+\dots+85=255$ $(8) 27+54+\dots+81=162$ $(9) 37+74=111$ $(1)-(6)-(7)-(8)-(9)=3787$



```

TEXT 100,110,"3",0,1,1,"(1) 1+2+3+...+100="+STR$(A)
TEXT 100,160,"3",0,1,1,"(2) 1+3+5+...+99="+STR$(B)
TEXT 100,210,"3",0,1,1,"(3) 2+4+6+...+100="+STR$(C)
TEXT 100,260,"3",0,1,1,"(4) 3+6+9+...+99="+STR$(D)
TEXT 100,310,"3",0,1,1,"(5) 5+10+15+...+100="+STR$(E)
TEXT 100,360,"3",0,1,1, " (1)-(5)= "+STR$(F)
TEXT 100,410,"3",0,1,1,"(6) 7+14+21+...+98="+STR$(G)
TEXT 100,460,"3",0,1,1,"(7) 17+34+51+...+85=" +STR$(H)
TEXT 100,510,"3",0,1,1,"(8) 27+54+...+81="+STR$(J)
TEXT 100,560,"3",0,1,1,"(9) 37+74="+STR$(K)
TEXT 100,610,"3",0,1,1," (1)-(6)-(7)-(8)-(9)="+STR$(L)
PRINT 1,1
EOP

```

DOWNLOAD F, "TEST.BAS"

SIZE 4,1

GAP 0,0

DIRECTION 1

CLS

A=85

B=10

:START

IF A<100 THEN GOTO L1 ELSE GOTO L2

:L1

CLS

TEXT 100,10,"3",0,1,1,STR\$(A) + " IS SMALLER THEN  
100"

PRINT 1

A=A+B

GOTO START

ENDIF

:L2

CLS

TEXT 100,10,"3",0,1,1,STR\$(A) + "IS LAGER THEN 100"

PRINT 1

EOP

TEST

105 IS LAGER THEN 100

95 IS SMALLER THEN 100

85 IS SMALLER THEN 100

**Note:**

- If the result of the expression is nonzero, the statement following THEN will be executed. If the result of the expression is zero, and the statement following the ELSE is present, it will be executed. Otherwise the next line of statement is executed.
- If there are block of statements in IF...THEN ...ELSE, ENDIF must be used at the end of the IF...THEN ...ELSE statement.

**Limitations:**

- The total numbers of nested IF ...THEN ...ELSE statement in a program cannot exceed 40.
- The total numbers of nested IF ...THEN ...ELSE, FOR...NEXT, GOSUB RETURN in a program cannot exceed 40 loops.

**See Also**

DOWNLOAD , EOP



## GOSUB...RETURN

### Description

This command will branch to a subroutine, executing statements until "RETURN" is reached.

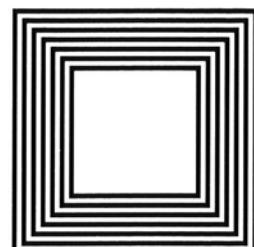
### Syntax

```
GOSUB LABEL
    statement
END

:LABEL
    statement
RETURN
```

Parameter	Description
LABEL	Beginning of the subroutine. The maximum length of the label is 8 characters.

### Example

Sample code	Result
<pre>DOWNLOAD "GOSUB1.BAS" SIZE 4,3 GAP 0,0 DIRECTION 1 CLS TEXT 10,10,"3",0,1,1,"GOSUB &amp; RETURN COMMAND TEST" GOSUB DR_BOX PRINT 1 END :DR_BOX FOR I=21 TO 81 STEP 10 BOX 80+I,80+I,80+300-I,80+300-I,5 NEXT RETURN EOP GOSUB1</pre>	<p>GOSUB &amp; RETURN COMMAND TEST</p> 

### See Also

DOWNLOAD, EOP, END, GOTO



## GOTO

### Description

This command is used to branch to a specified label. The label cannot exceed 8 characters in length.

### Syntax

**GOTO LABEL**

**:LABEL**

Parameter	Description
LABEL	Beginning of the point. The maximum length of the label is 8 characters.

### Example

Sample code	Result
<pre> DOWNLOAD "GOT01.BAS" SIZE 4,3 GAP 0,0 DIRECTION 1 CLS A=0 TOTAL=0 :START IF A&lt;100 THEN GOTO SUM ELSE GOTO PRTOUT ENDIF :SUM A=A+1 TOTAL=TOTAL+A GOTO START :PRTOUT B\$="THE SUMMATION OF 1..100 IS "+STR\$(TOTAL) TEXT 10,100, "3",0,1,1,B\$ PRINT 1 END EOP </pre>	THE SUMMATION OF 1..100 IS 5050

### See Also

DOWNLOAD, EOP, END, GOSUB...RETURN



## INP\$( )

### Description

One byte is received from communication port.

### Syntax

**INP\$(n)**

Parameter	Description
n	1: com1 port in printer

### Example

#### Sample code

```
DOWNLOAD "TEST.BAS"  
T$=""  
FOR I=1 TO 5  
T$=T$+INP$(1)  
NEXT  
  
SIZE 4,0.5  
GAP 0,0  
CLS  
TEXT 10,10, "3",0,1,1, "The received data is: "+T$  
PRINT 1  
EOP  
TEST  
12345
```

#### Result

The received data is: 12345

### See Also

**INP()**



## INP()

### Description

One byte (ASCII value) is received from communication port.

### Syntax

**INP\$(n)**

Parameter	Description
n	1 : com1 port in printer

**Note:**

This command has been supported since V6.91 EZ and later firmware.

### Example

**Sample code**

```
DOWNLOAD "TEST.BAS"
146sci=0
str$=""
FOR I=1 TO 5
146sci=INP(1)
str$=str$+" "+STR$(146sci)
OUT 146sci
NEXT
SIZE 4,0.5
GAP 0,0
CLS
TEXT 10,10, "3",0,1,1, "The received data is: "+str$
PRINT 1
EOP
TEST
12345
```

**Result**

```
The received data is: 49 50 51 52 53
```

### See Also

**INP\$( )**



## LOB()

### Description

This function returns the size of data in receiving buffer.

### Syntax

LOB()

#### Note:

This command has been supported since V6.78 EZ and later firmware.

### Example

#### Sample code

```
DOWNLOAD "TEST.BAS"  
  
DATA$=""  
  
WHILE LOB()<>0  
DATA$=DATA$+INP$(1)  
WEND  
  
SIZE 4,0.5  
GAP 0,0  
CLS  
BOX 10,10,800,100,2  
BLOCK 15,15,790,90, "0",0,8,8,DATA$  
PRINT 1  
EOP  
TEST  
We stand behind our products with one of the most comprehensive support  
programs in the Auto-ID industry.
```

#### Result

- 203 dpi:

We stand behind our products with one of the most comprehensive support programs in  
the Auto-ID industry.

- 300 dpi:

We stand behind our products with one of the most  
comprehensive support programs in the Auto-ID industry.

### See Also

INP\$(), WHILE ... WEND



## INPUT

### Description

This command receives data through specific port. This command is used with portable keyboard KP-200.

### Syntax

```
INPUT ["Prompt string", number of digits], variables
```

The comma also can be replaced by semicolon, such as:

```
INPUT ["Prompt string"; number of digits]; variables
```

Parameter	Description
Prompt string	The prompt string is shown on keyboard LCD screen. The maximum length of prompt string is 20 characters
Number of digits	Maximum number of characters is 255
Variables	The variable to receive input data

### Example

Sample code	Result
<pre>DOWNLOAD "TEXT.BAS" SIZE 4,3 GAP 0,0 DIRECTION 1  :START INPUT "CODE 39 : ",C39\$ INPUT "EAN 13: ",12,E13\$ CLS TEXT 20,50, "3",0,1,1, "INPUT and KP-200 Test" BARCODE 20,100, "39",48,1,0,2,5,C39\$ BARCODE 20,200, "EAN13",48,1,0,4,4,E13\$ PRINT 1 GOTO START EOP TEXT 123456 1234567890128</pre>	<p>INPUT and KP-200 Test</p>  <p>123456</p>  <p>1234567890128</p>

### See Also

DOWNLOAD, EOP, END, GOTO



## PREINPUT

### Description

This command can define the start character for command INPUT.

### Syntax

```
PREINPUT var$  
PREINPUT CHR$(n)
```

Parameter	Description
var\$	The specific character or string in front of data.
N	n = 1 ~ 255

**Note:**

This command has been supported since V6.81 EZ and later firmware.

### Example

```
PREINPUT "<"  
PREINPUT CHR$(2)
```

### See Also

POSTINPUT, INPUT, SET FILTER



## POSTINPUT

### Description

This command can define the end character for command INPUT.

### Syntax

```
POSTINPUT var$  
POSTINPUT CHR$(n)
```

Parameter	Description
var\$	The specific character or string in end of data.
N	n = 1 ~ 255

#### Note:

This command has been supported since V6.81 EZ and later firmware.

### Example

```
POSTINPUT ">"  
POSTINPUT CHR$(3)
```

### See Also

PREINPUT, INPUT, SET FILTER



## SET FILTER ON/OFF

### Description

This command is using to enable/disable commands PREINPUT and POSTINPUT.

### Syntax

**SET FILTER ON/OFF**

Parameter	Description
ON	Enable PREINPUT and POSTINPUT
OFF	Disable PREINPUT and POSTINPUT

**Note:**

This command has been supported since V6.81 EZ and later firmware.

### Example

Sample code	Result
<pre> DOWNLOAD "TEST.BAS" PREINPUT "&lt;=" POSTINPUT "&gt;=" SET FILTER ON  START: INPUT "DATA",data1\$ SIZE 4,0.25 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "DATA = "+data1\$ PRINT 1 GOTO START EOP TEST &lt;=1234=&gt;&lt;=5678=&gt;&lt;=9012=&gt; </pre>	<pre> DATA = 9012 DATA = 5678 DATA = 1234 </pre>

### See Also

PREINPUT, POSTINPUT, INPUT



## REM

### Description

Comment. Prefix is “REM”, which will be ignored by the printer.

### Syntax

REM

### Example

#### Sample code

```
REM ****
REM This is a demonstration program*
REM ****
DOWNLOAD "REMARK.BAS"
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT 50,50, "3",0,1,1, "REMARK DEMO PROGRAM"
REM TEXT 50,100, "3",0,1,1, "REMARK DEMO PROGRAM"
PRINT 1,1
EOP
REMARK
```

#### Result

REMARK DEMO PROGRAM

### See Also

DOWNLOAD, EOP, END



## OUT

### Description

This command returns data through the specific port.

### Syntax

```
OUT [port] "prompt",variable
OUT [port] "prompt";variable
```

Parameter	Description
port	Optional. Specified the port for returning data/string. Default is returning the data/string from the port which is sending data to printer. <b>COM:</b> Returning data/string from COM port. <b>USB:</b> Returning data/string from USB port. <b>NET:</b> Returning data/string from LAN port.
Prompt	Prompt string.
Variable	The output message.
,	The “prompt” and “variable” are separated by <0x0D><0x0A>.
;	The “variable” comes behind “prompt” directly.

#### Note:

This command has been supported since V6.93 EZ and later firmware.

### Example

#### Sample code

```
OUT USB "USB: ";"This is returning data from USB. "
OUT COM "COM"; "This is returning data from COM. "
OUT NET "NET: ";"This is returning data from LAN. "
```

#### Result



## OUTR

### Description

This command sends data through RS-232 port only.

### Syntax

```
OUTR "prompt",variable
OUTR "prompt";variable
```

Parameter	Description
prompt	Prompt string.
variable	The output message.
,	The “prompt” and “variable” are separated by <0x0D><0x0A>.
;	The “variable” comes behinds “prompt” directly.

#### Note:

This command has been supported since V6.68 EZ and later firmware.

### Example

#### Sample code

```
OUTR "COM: "," This is returning data from RS-232 only."
```

#### Result

The screenshot shows the Communication Tools application interface. In the 'Transmit' tab, the 'RS232' option is selected, and the message 'OUTR "COM: "," This is returning data from RS-232 only."' is entered. In the 'Receive' tab, the message is received in both hex and ASCII formats. The status bar at the bottom indicates 'Transmit data finished' and 'COM1 9600,N,8,1 None'.



## GETKEY( )

### Description

This command is used to get the status of the PAUSE and FEED keys. This command waits until either key is pressed, whereupon 0 is returned if PAUSE key is pressed and 1 is returned if FEED key is pressed.

### Syntax

**GETKEY()**

PAUSE	FEED
0	1

### Example

#### Sample code

```
DOWNLOAD "DEMO4.BAS"
SIZE 4,3
GAP 0,0
CLS
:START
A=GETKEY()
IF A=0 THEN GOTO PAUSEB
IF A=1 THEN GOTO FEEDB
:PAUSEB
CLS
TEXT 50,10, "4",0,1,1, "PAUSE key is pressed !"
PRINT 1
GOTO START
:FEEDB
CLS
TEXT 50,10, "4",0,1,1, "FEED key is pressed !"
PRINT 1
EOP
```

### See Also

DOWNLOAD, EOP, END, GOTO



## INT( )

### Description

This function truncates a floating point number.

### Syntax

**INT (n)**

Parameter	Description
n	Positive or negative integer, floating point number or mathematical expression

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 INPUT "Number: ",Num CLS REM **** To round up or down**** N=INT(Num+0.5) IF N&gt;Num THEN TEXT 50,100, "3",0,1,1, "To round up= " +STR\$(N) ELSE TEXT 50,100, "3",0,1,1, "To round down= " +STR\$(N) ENDIF PRINT 1 EOP 56.2</pre>	To round down= 56

### See Also

DOWNLOAD, EOP, END, ABS(), ASC(), STR\$()



## LEFT\$( )

### Description

This function returns the specified number of characters down from the initial character of a string.

### Syntax

**LEFT\$ (X\$, n)**

Parameter	Description
X\$	The string to be processed
n	The number of characters to be returned

### Example

Sample code	Result
<pre>DOWNLOAD "TEST.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="BARCODE PRINTER DEMO PRINTING" C\$=LEFT\$(A\$,10) CLS TEXT 10,10,"3",0,1,1,A\$ TEXT 10,100,"3",0,1,1, "10 LEFT 10 CHARS: "+C\$ PRINT 1 EOP TEST</pre>	<pre>BARCODE PRINTER DEMO PRINTING 10 LEFT 10 CHARS: BARCODE PR</pre>

### See Also

[DOWNLOAD](#), [EOP](#), [END](#), [RIGHT\\$\( \)](#), [MID\\$\( \)](#), [LEN\( \)](#), [STR\\$\( \)](#)



## LEN()

### Description

This function returns the length of a string.

### Syntax

**LEN (string)**

Parameter	Description
string	The string whose length is to be measured.

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHIJKLMNPQRSTUVWXYZ" B=LEN(A\$) CLS TEXT 10,10, "3",0,1,1,A\$ TEXT 10,50, "3",0,1,1,"STRING LENGTH=" +STR\$(B) PRINT 1 EOP DEMO</pre>	<pre>ABCDEFGHIJKLMNPQRSTUVWXYZ STRING LENGTH=26</pre>

### See Also

DOWNLOAD, EOP, END, LEFT\$(), LEN(), RIGHT\$(), MID\$(), STR\$(), VAL()



## MID\$( )

### Description

This function retrieves the specified number of characters down from the mth character of a string.

### Syntax

**MID\$(string,m,n)**

Parameter	Description
string	The string to be processed
m	The beginning of mth characters in the string 1 <= m <= string length
n	The number of characters to return

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHIJKLMNPQRSTUVWXYZ" E\$=MID\$(A\$,11,10) CLS TEXT 10,10, "3",0,1,1,A\$ TEXT 10,200, "3",0,1,1,"10 MIDDLE CHARS: "+E\$ PRINT 1 EOP DEMO</pre>	<pre>ABCDEFGHIJKLMNPQRSTUVWXYZ 10 MIDDLE CHARS: KLMNOPQRST</pre>

### See Also

DOWNLOAD, EOP, END, LEFT\$( ), LEN( ), RIGHT\$( ), STR\$( ), VAL( )



## RIGHT\$( )

### Description

This function returns a specified number of characters up from the end of a string.

### Syntax

**RIGHT\$ (X\$,n)**

Parameter	Description
X\$	The string to be processed
n	The number of characters to be returned from the right side (end) of the string

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHIJKLMNPQRSTUVWXYZ" D\$=RIGHT\$(A\$,10) CLS TEXT 10,10,"3",0,1,1,A\$ TEXT 10,150,"3",0,1,1, "10 RIGHT CHARS: "+D\$ PRINT 1 EOP DEMO</pre>	<pre>ABCDEFGHIJKLMNPQRSTUVWXYZ 10 RIGHT CHARS: QRSTUVWXYZ</pre>

### See Also

DOWNLOAD, EOP, END, LEFT\$( ), LEN( ), MID\$( ), STR\$( ), VAL( )



## STR\$( )

### Description

This function converts a specified value or expression into corresponding string of characters.

### Syntax

**STR\$ (n)**

Parameter	Description
n	An integer, floating point number or mathematical expression

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHIJKLMNPQRSTUVWXYZ" F=100 G=500 H\$=STR\$(F+G) CLS TEXT 10,10, "3",0,1,1,A\$ TEXT 10,60, "3",0,1,1, "F=" +STR\$(F) TEXT 10,110, "3",0,1,1, "G=" +STR\$(G) TEXT 10,160, "3",0,1,1, "F+G=" +H\$ PRINT 1 EOP DEMO</pre>	<pre>ABCDEFGHIJKLMNPQRSTUVWXYZ F=100 G=500 F+G=600</pre>

### See Also

[DOWNLOAD](#), [EOP](#), [END](#), [LEFT\\$\(\)](#), [LEN\(\)](#), [RIGHT\\$\(\)](#), [MID\\$\(\)](#), [VAL\(\)](#)



## STRCOMP( )

### Description

Returns -1, 0, or 1, based on the result of a string comparison.

### Syntax

**STRCOMP(str1\$,str2\$[,comp])**

Parameter	Description
str1\$	Required. Any valid string expression.
str2\$	Required. Any valid string expression.
comp	Optional. Specifies the type of string comparison. 0: Binary comparison. Default. 1: Textual comparison. The comparison is case-insensitive.
Condition	Return value
str1\$ sorts ahead of str2\$	-1
str1\$ is equal to str2\$	0
str1\$ sorts after str2\$	1

**Note:**

This command has been supported since V6.81 EZ and later firmware.

### Example

#### Sample code

```

DOWNLOAD "TEST.BAS"
STR1$ = "ABCD"
STR2$ = "abcd"

result1 = STRCOMP(STR1$,STR2$)
result2 = STRCOMP(STR1$,STR2$,1)
result3 = STRCOMP(STR2$,STR1$)

SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 100,10,"3",0,1,1,STR$(result1)+": \[" +STR1$+"\" sorts ahead of \[" +STR2$+" \""
TEXT 100,60,"3",0,1,1," "+STR$(result2)+": \[" +STR1$+"\" is equal to \[" +STR2$+" \""
TEXT 100,110,"3",0,1,1," "+STR$(result3)+": \[" +STR2$+"\" sorts after \[" +STR1$+" \""
PRINT 1
EOP
TEST

```

**Result**

```
-1: "ABCD" sorts ahead of "abcd"  
0: "ABCD" is equal to "abcd"  
1: "abcd" sorts after "ABCD"
```

**See Also**[INSTR\(\)](#)



## INSTR()

### Description

Returns an integer specifying the start position of the first occurrence of one string within another.

### Syntax

**INSTR ([start,]str1\$,str2\$)**

Parameter	Description
start	Optional. Numeric expression that sets the starting position for each search. If omitted, search begins at the first character position. The stat index is 1 – based.
str1\$	Required. String expression being searched.
str2\$	Required. String expression sought.

**Note:**

This command has been supported since V6.59 EZ and later firmware.

### Example

**Sample code**

```

DOWNLOAD "DEMO.BAS"
string$="ABC123ABC123"
searchfor$="123"
starpos=8

temp1=INSTR(string$,searchfor$)
temp2=INSTR(starpos,string$,searchfor$)

str1$=searchfor$+"in "+string$+" is "+STR$(temp1)
str2$=searchfor$+"in "+string$+"after"+STR$(starpos)+" is "+STR$(temp2)

SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1,str1$
TEXT 10,60, "3",0,1,1,str2$
PRINT 1
EOP
DEMO

```

**Result**

```

123 in ABC123ABC123 is 4
123 in ABC123ABC123 after 8 is 10

```

### See Also

[STRCOMP\(\)](#)



## TRIM\$( )

### Description

Removes both leading and trailing blank spaces or specific characters from a string.

### Syntax

**TRIM\$ (str\$,list\$)**

Parameter	Description
str\$	The string that will be trimmed.
list\$	Optional. The specific characters in list\$ will be removed.

**Note:**

This command has been supported since V6.59 EZ and later firmware.

### Example

#### Sample code

```

DOWNLOAD "DEMO.BAS"
data1$="1234567"
data2$="a1234567a"
data3$="[<12345>]"

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 50,020,"3",0,1,1,"LTRIM$([[" +data1$+" \[[")
TEXT 50,050,"3",0,1,1,"TRIM$ ([[" +data1$+" \[[") = " +TRIM$(data1$)
TEXT 50,080,"3",0,1,1,"RTRIM$([[" +data1$+" \[[") = " +RTRIM$(data1$)
TEXT 50,110,"3",0,1,1,"LTRIM$([[" +data2$+ "\[["], \[["]a\[["]) = " +LTRIM$(data2$,"a")
TEXT 50,140,"3",0,1,1,"TRIM$ ([[" +data2$+ "\[["], \[["]a\[["]) = " +TRIM$(data2$,"a")
TEXT 50,170,"3",0,1,1,"RTRIM$([[" +data2$+ "\[["], \[["]a\[["]) = " +RTRIM$(data2$,"a")
TEXT 50,200,"3",0,1,1,"LTRIM$([[" +data3$+ "\[["], \[["][<>]\[["]) = " +LTRIM$(data3$,"[<>]")
TEXT 50,230,"3",0,1,1,"TRIM$ ([[" +data3$+ "\[["], \[["][<>]\[["]) = " +TRIM$(data3$,"[<>]")
TEXT 50,260,"3",0,1,1,"RTRIM$([[" +data3$+ "\[["], \[["][<>]\[["]) = " +RTRIM$(data3$,"[<>]")
PRINT 1
EOP
DEMO

```



## Result

```
LTRIM$(" 1234567 ")      = 1234567
TRIM$(" 1234567 ")       = 1234567
RTRIM$(" 1234567 ")      = 1234567
LTRIM$("a1234567a", "a") = 1234567a
TRIM$("a1234567a", "a") = 1234567
RTRIM$("a1234567a", "a") = a1234567
LTRIM$("<12345>", "<>") = 12345>
TRIM$("<12345>", "<>") = 12345
RTRIM$("<12345>", "<>") = [<12345
```

## See Also

[LTRIM\\$\(\)](#), [RTRIM\\$\(\)](#)



## LTRIM\$( )

### Description

Removes leading blank space from a string.

### Syntax

**LTRIM\$ (str\$,list\$)**

Parameter	Description
str\$	The string that will be trimmed.
list\$	Optional. The specific characters in list\$ will be removed.

**Note:**

This command has been supported since V6.59 EZ and later firmware.

### Example

#### Sample code

```

DOWNLOAD "DEMO.BAS"
data1$="1234567"
data2$="a1234567a"
data3$="[<12345>]"

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 50,020,"3",0,1,1,"LTRIM$([[" +data1$+" \[[")
TEXT 50,050,"3",0,1,1,"TRIM$ ([[" +data1$+" \[[")      =" +LTrim$(data1$)
TEXT 50,080,"3",0,1,1,"RTRIM$([[" +data1$+" \[[")      =" +RTrim$(data1$)
TEXT 50,110,"3",0,1,1,"LTRIM$([[" +data2$+" \[["],\[["]a\[["])    =" +LTrim$(data2$,"a")
TEXT 50,140,"3",0,1,1,"TRIM$ ([[" +data2$+" \[["],\[["]a\[["])    =" +Trim$(data2$,"a")
TEXT 50,170,"3",0,1,1,"RTRIM$([[" +data2$+" \[["],\[["]a\[["])    =" +RTrim$(data2$,"a")
TEXT 50,200,"3",0,1,1,"LTRIM$([[" +data3$+" \[["],\[["][<>]\[["])  =" +LTrim$(data3$,"[<>]")
TEXT 50,230,"3",0,1,1,"TRIM$ ([[" " +data3$+" \[["],\[["][<>]\[["])  =" +Trim$(data3$,"[<>"])
TEXT 50,260,"3",0,1,1,"RTRIM$([[" +data3$+" \[["],\[["][<>]\[["])  =" +RTrim$(data3$,"[<>"])

PRINT 1
EOP
DEMO

```



## Result

```
LTRIM$(" 1234567 ")      = 1234567
TRIM$(" 1234567 ")       = 1234567
RTRIM$(" 1234567 ")      = 1234567
LTRIM$("a1234567a", "a") = 1234567a
TRIM$("a1234567a", "a") = 1234567
RTRIM$("a1234567a", "a") = a1234567
LTRIM$("[<12345>]", "[<>]") = 12345>
TRIM$("[<12345>]", "[<>]") = 12345
RTRIM$("[<12345>]", "[<>]") = [<12345
```

## See Also

[TRIM\\$\(\)](#), [RTRIM\\$\(\)](#)



## RTRIM\$( )

### Description

Removes trailing blank space from a string.

### Syntax

**RTRIM\$ (str\$ [, list\$])**

Parameter	Description
str\$	The string that will be trimmed.
list\$	Optional. The specific characters in list\$ will be removed.

**Note:**

This command has been supported since V6.59 EZ and later firmware.

### Example

#### Sample code

```

DOWNLOAD "DEMO.BAS"
data1$="1234567"
data2$="a1234567a"
data3$="[<12345>]"

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 50,020,"3",0,1,1, "LTRIM$([“]"+data1$+"[“])" = "+LTrim$(data1$)
TEXT 50,050,"3",0,1,1, "TRIM$ ([“]"+data1$+"[“])" = "+Trim$(data1$)
TEXT 50,080,"3",0,1,1, "RTRIM$([“]"+data1$+"[“])" = "+RTrim$(data1$)
TEXT 50,110,"3",0,1,1, "LTRIM$([“]"+data2$+"[“], [“]a\[“])" = "+LTrim$(data2$,"a")
TEXT 50,140,"3",0,1,1, "TRIM$ ([“]"+data2$+"[“], [“]a\[“])" = "+Trim$(data2$,"a")
TEXT 50,170,"3",0,1,1, "RTRIM$([“]"+data2$+"[“], [“]a\[“])" = "+RTrim$(data2$,"a")
TEXT 50,200,"3",0,1,1, "LTRIM$([“]"+data3$+"[“], [“][<>]\[“])" = "+LTrim$(data3$,"[<>]")
TEXT 50,230,"3",0,1,1, "TRIM$ ([“]"+data3$+"[“], [“][<>]\[“])" = "+Trim$(data3$,"[<>]")
TEXT 50,260,"3",0,1,1, "RTRIM$([“]"+data3$+"[“], [“][<>]\[“])" = "+RTrim$(data3$,"[<>]")
PRINT 1
EOP
DEMO

```



## Result

```
LTRIM$(" 1234567 ")      = 1234567
TRIM$(" 1234567 ")      = 1234567
RTRIM$(" 1234567 ")      = 1234567
LTRIM$("a1234567a", "a") = 1234567a
TRIM$("a1234567a", "a") = 1234567
RTRIM$("a1234567a", "a") = a1234567
LTRIM$("[<12345>]", "[<>]") = 12345>]
TRIM$("[<12345>]", "[<>]") = 12345
RTRIM$("[<12345>]", "[<>]") = [<12345
```

## See Also

[TRIM\\$\(\)](#), [LTRIM\\$\(\)](#)



## TEXTPIXEL()

### Description

Returns the width of the text string in dot.

### Syntax

**TEXTPIXEL (cont\$,font\$,size)**

Parameter	Description
cont\$	The content of text string.
font\$	The font type. Please refer to the parameter <i>font</i> in command TEXT.
size	The font size. Please refer to the parameter x-multiplication in command TEXT.

#### Note:

This command has been supported since V6.61 EZ and later firmware.

### Example

Sample code	Result
<pre>DOWNLOAD "TEST.BAS" str\$="ABCDEFG" font\$="3" fontsize=3 strwidth=TEXTPIXEL(str\$,font\$,fontsize)  SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10,font\$,0,fontsize,fontsize,str\$ REVERSE 8,8,strwidth,72 PRINT 1 EOP TEST</pre>	

### See Also

TEXT, BARCODEPIXEL()



## BARCODEPIXEL( )

### Description

Returns the width of barcode in dot.

### Syntax

**BARCODEPIXEL (cont\$, sym\$, narrow, wide)**

Parameter	Description
cont\$	The content of barcode.
sym\$	Barcode type. Please refer to the parameter <i>code type</i> in command BARCODE.
narrow	The width of narrow bar. Please refer to the parameter <i>narrow</i> in command BARCODE.
wide	The width of wide bar. Please refer to the parameter <i>wide</i> in command BARCODE.

**Note:**

This command has been supported since V6.72 EZ and later firmware.

### Example

Sample code	Result
<pre> DOWNLOAD "TEST.BAS"  cont\$="ABCDEFG" sym\$="39" narrow=2 wide=6 codewidth=BARCODEPIXEL(cont\$,sym\$,narrow,wide)  SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS BARCODE 10,10,sym\$,100,1,0,narrow,wide,cont\$ REVERSE 8,8,codewidth+8,132 BARCODE 10,160,sym\$,100,1,0,narrow,wide,cont\$ PRINT 1 EOP TEST </pre>	 

### See Also

BARCODE, TEXTPIXEL()



## VAL()

### Description

This function converts numeric characters into corresponding integer or floating point number.

### Syntax

**VAL ("numeric character")**

Parameter	Description
numeric character	" 0~9","."

### Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHIJKLMNPQRSTUVWXYZ" F\$="100" G\$="500" CLS H=VAL(F\$)+VAL(G\$) I\$=STR\$(H) TEXT 10,10, "3",0,1,1,A\$ TEXT 10,60, "3",0,1,1, "F=" +F\$ TEXT 10,110, "3",0,1,1, "G=" +G\$ TEXT 10,160, "3",0,1,1, "F+G=" +I\$ PRINT 1 EOP DEMO</pre>	<pre>ABCDEFGHIJKLMNPQRSTUVWXYZ F=100 G=500 F+G=600</pre>

### See Also

DOWNLOAD, EOP, END, LEFT\$(), LEN(), RIGHT\$(), MID\$(), STR\$()



## BEEP

### Description

This command issues a beep sound on portable keyboard. Printer sends the string 0x07 to KP-200 portable keyboard.

### Syntax

**BEEP**

### Example

#### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
BEEP
INPUT "Text1 =",TEXT1$
CLS
TEXT 100,100, "3",0,1,1,TEXT1$
PRINT 1
EOP
```



## NOW\$()

### Description

Returns the current date and time according to the setting of your printer. The returned value always uses with commands FORMAT\$().

### Syntax

**NOW\$()**

**Note:**

This command has been supported since V6.81 EZ and later firmware.

### Example

Sample code

```
SIZE 4,1  
GAP 0,0  
DIRECTION 1  
CLS  
TEXT 10,10, "3",0,1,1, "Now is " +NOW$()  
TEXT 10,60, "3",0,1,1,FORMAT$(NOW$(),"Long Date")  
PRINT 1
```

Result

```
Now is 1/9/2013 2:19:27 PM  
Tuesday, January 09 2013
```

### See Also

FORMAT\$()



## NOW

### Description

Returns the total days since A.D. 1900. This global variable always uses with commands `FORMAT$()` and `DATEADD()`.

### Syntax

`NOW`

**Note:**

This command has been supported since V6.87 EZ and later firmware.

### Example

#### Sample code

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Total days since a.d. 1900: " +STR$(NOW)+ " days"
TEXT 10,50, "3",0,1,1, "Date Info in RTC: " +FORMAT$(NOW, "General Date")
TEXT 10,90, "3",0,1,1, "Date after a year: " +FORMAT$(DATEADD("yyyy",1,NOW), "General Date")
PRINT 1
```

#### Result

```
Total days since a.d. 1900: 41283.597176 days
Date Info in RTC: 1/9/2013 2:19:56 PM
Date after a year: 1/9/2014 2:19:56 PM
```

### See Also

`FORMAT$()`, `DATEADD()`, `NOW`



## FORMAT\$( )

### Description

Returns the current date, time, number and number value according to the setting of your printer.

### Syntax

**FORMAT\$(expression[, style\$])**

Parameter	Description
expression	Required. Any valid expression.
style\$	Optional. A valid named or user-defined format string expression.

Predefined date/time formats	Description
General Date	Shows date and time.
Long Date	Uses the Long Date format.
Medium Date	Uses the dd-mmm-yy format.
Short Date	Uses the Short Date format.
Long Time	Shows the hour, minute, second, and “AM” or “PM” using the h:mm:ss format.
Medium Time	Shows the hour, minute, and “AM” or “PM” using the “hh:mm AM/PM” format.
Short Time	Shows the hour and minute using the hh:mm format.

User-defined date/time formats	Description
c	Display the date as dddd and display the time as tttt, in that order.
d	Display the day as a number without a leading zero (1 – 31).
dd	Display the day as a number with a leading zero (01 – 31).
ddd	Display the day as an abbreviation (Sun – Sat).
dddd	Display the day as a full name (Sunday – Saturday).
ddddd	Display a date serial number as a complete date (including day, month, and year), formatted according to your system’s short date format setting. The default short date format is m/d/yyyy.
ddddd	Display the date as a complete date (including day, month, and year), formatted according to the long date setting recognized by your system. The default long date format is dddd, mmmm dd, yyyy.
w	Display the day of the week as a number (1 for Sunday through 7 for Saturday).
ww	Display the week of the year as a number (1 – 53).
m	Display the month as a number without a leading zero (1 – 12). If m immediately follows h or hh, the minute rather than the month is displayed.



User-defined date/time formats	Description
mm	Display the month as a number with a leading zero (01 – 12). If mm immediately follows h or hh, the minute rather than the month is displayed.
mmm	Display the month as an abbreviation (Jan – Dec).
mmmm	Display the month as a full month name (January – December).
q	Display the quarter of the year as a number (1 – 4).
y	Display the day of the year as a number (1 – 366).
yy	Display the year as a 2-digit number (00 – 99).
yyyy	Display the year as a 4-digit number (100 – 9999).
h	Display the hour as a number without leading zeros (0 – 23).
hh	Display the hour as a number with leading zeros (00 – 23).
n	Display the minute as a number without leading zeros (0 – 59).
nn	Display the minute as a number with leading zeros (00 – 59).
s	Display the second as a number without leading zeros (0 – 59).
ss	Display the second as a number with leading zeros (00 – 59).
tttt	Display a time as a complete time (including hour, minute, and second). The default time format is h:mm:ss AM/PM.
AM/PM	Display an uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.
am/pm	Display a lowercase AM with any hour before noon; display a lowercase PM with any hour between noon and 11:59 P.M.
A/P	Display an uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.
a/p	Display a lowercase A with any hour before noon; display a lowercase P with any hour between noon and 11:59 P.M.
AMPM	AMPM can be either uppercase or lowercase, but the case of the string displayed matches the string as defined by your system settings.
\	Display the next character in the format string.
"string"	Display the string inside the double quotation marks.



<b>Number formats (since A1.97)</b>	<b>Description</b>
General Number	Displays the number as entered, with no rounding and no commas.
Currency	Displays the number with a dollar sign, comma (if appropriate), and two digits to the right of the decimal point. Shows negative numbers inside parentheses.
Fixed	Displays the number with at least one digit to the left of the decimal separator and two digits to the right. Does not show comma.
Standard	Displays the number with at least one digit to the left of the decimal separator and two digits to the right and commas (if appropriate).
Percent	Multiplies the value by 100 and displays the result with two digits to the right of the decimal point and a percent sign at the end.
Scientific	Uses standard scientific notation.
Yes/No	Any nonzero numeric value is Yes. Zero is No.
True/False	Any nonzero numeric value is True. Zero is False.
On/Off	Any nonzero numeric value is On. Zero is Off.

<b>User-defined number formats (since A1.97)</b>	<b>Description</b>
0	Digit placeholder. Displays a digit or a zero.
#	Digit placeholder. Displays a digit or nothing.
.	Decimal placeholder.
%	Percent placeholder. Multiplies the expression by 100.
,	Thousand separator.
E- E+ e- e+	Scientific format.
\	Display the next character in the format string.
"ABC"	Display the string inside the double quotation marks.



Different formats for different number values (since A1.97)	Description
One section only	The format expression applies to all values.
Two section	The first section applies to positive values and zeros; the second applies to negative values.
Three sections	The first section applies to positive values, the second applies to negative values, and the third applies to zeros.

**Note:**

This command has been supported since V6.81 EZ and later firmware.

**See Also**

NOW\$(), DATEADD(), NOW

**Example**

Sample Code	Result
<pre> SIZE 800 dot,1900 dot GAP 0,0 DIRECTION 1 CLS TEXT 15,10, "3",0,1,1, "General Date: "+FORMAT\$(NOW,"General Date") TEXT 15,60, "3",0,1,1, "Long Date: "+FORMAT\$(NOW,"Long Date") TEXT 15,110, "3",0,1,1, "Medium Date: "+FORMAT\$(NOW,"Medium Date") TEXT 15,160, "3",0,1,1, "Short Date: "+FORMAT\$(NOW,"Short Date") TEXT 15,210, "3",0,1,1, "Long Time: "+FORMAT\$(NOW,"Long Time") TEXT 15,260, "3",0,1,1, "Medium Time: "+FORMAT\$(NOW,"Medium Time") TEXT 15,310, "3",0,1,1, "Short Time: "+FORMAT\$(NOW,"Short Time") TEXT 15,360, "3",0,1,1, "c: "+FORMAT\$(NOW,"c") TEXT 15,410, "3",0,1,1, "d: "+FORMAT\$(NOW,"d") TEXT 15,460, "3",0,1,1, "dd: "+FORMAT\$(NOW,"dd") TEXT 15,510, "3",0,1,1, "ddd: "+FORMAT\$(NOW,"ddd") TEXT 15,560, "3",0,1,1, "dddd: "+FORMAT\$(NOW,"dddd") TEXT 15,610, "3",0,1,1, "ddddd: "+FORMAT\$(NOW,"ddddd") TEXT 15,660, "3",0,1,1, "ddddddd: "+FORMAT\$(NOW,"ddddddd") TEXT 15,710, "3",0,1,1, "w: "+FORMAT\$(NOW,"w") TEXT 15,760, "3",0,1,1, "ww: "+FORMAT\$(NOW,"ww") TEXT 15,810, "3",0,1,1, "m: "+FORMAT\$(NOW,"m") TEXT 15,860, "3",0,1,1, "mm: "+FORMAT\$(NOW,"mm") TEXT 15,910, "3",0,1,1, "mmm: "+FORMAT\$(NOW,"mmm") TEXT 15,960, "3",0,1,1, "mmmm: "+FORMAT\$(NOW,"mmmm") TEXT 15,1010, "3",0,1,1, "q: "+FORMAT\$(NOW,"q") TEXT 15,1060, "3",0,1,1, "y: "+FORMAT\$(NOW,"y") TEXT 15,1110, "3",0,1,1, "yy: "+FORMAT\$(NOW,"yy") TEXT 15,1160, "3",0,1,1, "yyyy: "+FORMAT\$(NOW,"yyyy") </pre>	<pre> General Date:1/9/2013 2:46:18 PM Long Date:Tuesday, January 09 2013 Medium Date:09-Jan-13 Short Date:1/9/2013 Long Time:2:46:18 PM Medium Time:02:46 PM Short Time:14:46 c:1/9/2013 2:46:18 PM d:9 dd:09 ddd:Tue dddd:Tuesday ddddd:1/9/2013 ddddddd:Tuesday, January 09 2013 u:3 uu:2 n:1 nn:01 nnn:Jan nnnn:January q:1 y:9 yy:13 yyyy:2013 h:14 hh:14 n:48 nn:48 s:18 ss:18 ttttt:2:46:18 PM AM/PM:PM am/pm:pm A/P:P a/p:p AMPM:PM \:Today is 1/9/2013 string:Today is 1/9/2013 </pre>



```

TEXT 15,1210, "3",0,1,1, "h: " +FORMAT$(NOW,"h")
TEXT 15,1260, "3",0,1,1, "hh: " +FORMAT$(NOW,"hh")
TEXT 15,1310, "3",0,1,1, "n: " +FORMAT$(NOW,"n")
TEXT 15,1360, "3",0,1,1, "nn: " +FORMAT$(NOW,"nn")
TEXT 15,1410, "3",0,1,1, "s: " +FORMAT$(NOW,"s")
TEXT 15,1460, "3",0,1,1, "ss: " +FORMAT$(NOW,"ss")
TEXT 15,1510, "3",0,1,1, "tttt: " +FORMAT$(NOW,"tttt")
TEXT 15,1560, "3",0,1,1, "AM/PM: " +FORMAT$(NOW,"AM/PM")
TEXT 15,1610, "3",0,1,1, "am/pm: " +FORMAT$(NOW,"am/pm")
TEXT 15,1660, "3",0,1,1, "A/P: " +FORMAT$(NOW,"A/P")
TEXT 15,1710, "3",0,1,1, "a/p: " +FORMAT$(NOW,"a/p")
TEXT 15,1760, "3",0,1,1, "AMPM: " +FORMAT$(NOW,"AMPM")
TEXT 15,1810, "3",0,1,1, "\: " +FORMAT$(NOW,"To\day i\s dddd")
TEXT 15,1860, "3",0,1,1, "string: " +FORMAT$(NOW,"To\day i\s
dddd")
PRINT 1

```

**Sample Code (Since A1.97)**

```

SIZE 800 dot,850 dot
GAP 0,0
DIRECTION 1
CLS
TEXT 15,10, "3",0,1,1, "General Number: "+FORMAT$(1234.5,"General
Number")
TEXT 15,60, "3",0,1,1, "Currency: "+FORMAT$(1234.5,"Currency")
TEXT 15,110, "3",0,1,1, "Fixed: "+FORMAT$(1234.5,"Fixed")
TEXT 15,160, "3",0,1,1, "Standard: "+FORMAT$(1234.5,"Standard")
TEXT 15,210, "3",0,1,1, "Percent: "+FORMAT$(1234.5,"Percent")
TEXT 15,260, "3",0,1,1, "Scientific: "+FORMAT$(1234.5,"Scientific")
TEXT 15,310, "3",0,1,1, "Yes/No: "+FORMAT$(1234.5,"Yes/No")
TEXT 15,360, "3",0,1,1, "Yes/No: "+FORMAT$(0,"Yes/No")
TEXT 15,410, "3",0,1,1, "True/False: "+FORMAT$(0,"True/False")
TEXT 15,460, "3",0,1,1, "On/Off: "+FORMAT$(0,"On/Off")
TEXT 15,510, "3",0,1,1, "00000.00: "+FORMAT$(1234.5,"00000.00")
TEXT 15,560, "3",0,1,1, "#####.##: "+FORMAT$(1234.5,"#####.##")
TEXT 15,610, "3",0,1,1, "##.##0.00: "+FORMAT$(1234.5,"##.##0.00")
TEXT 15,660, "3",0,1,1, "$##0.00: "+FORMAT$(1234.5,"$##0.00")
TEXT 15,710, "3",0,1,1, "$0.00%: "+FORMAT$(1234.5,"0.00%")
TEXT 15,760, "3",0,1,1, "Yes/No: "+FORMAT$(-12.3,"Yes/No")
TEXT 15,810, "3",0,1,1, "0.00;(0.00): "+FOR-
MAT$(-12.3,"0.00;(0.00)")
PRINT 1

```

**Result**

General Number: 1234.5  
 Currency: \$1,234.50  
 Fixed: 1234.50  
 Standard: 1,234.50  
 Percent: 123450.00%  
 Scientific: 1.23E+03  
 Yes/No: Yes  
 Yes/No: No  
 True/False: False  
 On/Off: Off  
 00000.00: 01234.50  
 #####.##: 1234.5  
 ##.##0.00: 1.234.50  
 \$##0.00: \$1234.50  
 \$0.00%: 123450.00%  
 Yes/No: Yes  
 0.00;(0.00): (12.30)



## DATEADD()

### Description

Returns a date after which a specified time/date interval has been added. The returned value always uses with commands FORMAT\$().

### Syntax

**DATEADD(interval\$,number,date)**

Parameter	Description																							
interval\$,	The time/date interval for adding. It can be one of following values.																							
	<table border="1"><thead><tr><th>Interval\$</th><th>The interval unit of parameter interval\$</th></tr></thead><tbody><tr><td>“yyyy”</td><td>Year.</td></tr><tr><td>“q”</td><td>Quarter.</td></tr><tr><td>“m”</td><td>Month.</td></tr><tr><td>“y”</td><td>Day of year.</td></tr><tr><td>“d”</td><td>Day.</td></tr><tr><td>“w”</td><td>Weekday.</td></tr><tr><td>“ww”</td><td>Week of year.</td></tr><tr><td>“h”</td><td>Hour.</td></tr><tr><td>“n”</td><td>Minute.</td></tr><tr><td>“s”</td><td>Second.</td></tr></tbody></table>		Interval\$	The interval unit of parameter interval\$	“yyyy”	Year.	“q”	Quarter.	“m”	Month.	“y”	Day of year.	“d”	Day.	“w”	Weekday.	“ww”	Week of year.	“h”	Hour.	“n”	Minute.	“s”	Second.
Interval\$	The interval unit of parameter interval\$																							
“yyyy”	Year.																							
“q”	Quarter.																							
“m”	Month.																							
“y”	Day of year.																							
“d”	Day.																							
“w”	Weekday.																							
“ww”	Week of year.																							
“h”	Hour.																							
“n”	Minute.																							
“s”	Second.																							
number	The number of interval\$ for adding.																							
date	The date which is used to add the interval\$. Date format: “yyyy/mm/dd” Time format: “hh:nn:ss”																							

#### Note:

This command has been supported since V6.87 EZ and later firmware.



## Example

### Sample code 1

```
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Current RTC info: " +NOW()
TEXT 10,60, "3",0,1,1, "-1 year: " +FORMAT$(DATEADD("yyyy",-1, " 11/26/2012 10:08:00"), "yyyy/mm/dd hh:mm:ss")
TEXT 10,110, "3",0,1,1, "+9 months: " +FORMAT$(DATEADD("m",9,NOW), "Short Date")
TEXT 10,160, "3",0,1,1, "-8 hours: " +FORMAT$(DATEADD("h",-8,NOW), "Short Time")
TEXT 10,210, "3",0,1,1, "+5 mins: " +FORMAT$(DATEADD("n",5,NOW), "Short Time")
TEXT 10,260, "3",0,1,1, "+00 day: " +FORMAT$(NOW, "Short Date")
TEXT 10,310, "3",0,1,1, "+20 days: " +FORMAT$(DATEADD("d",20,NOW), "Short Date")
TEXT 10,360, "3",0,1,1, "-20 day: " +FORMAT$(DATEADD("d",-20,NOW), "Short Date")
PRINT 1
```

### Result 1

```
Current RTC info: 1/9/2013 3:20:06 PM
-1 year: 2011/11/26 10:08:00
+9 months: 10/9/2013
-8 hours: 07:20
+5 mins: 15:25
+00 day: 1/9/2013
+20 days: 1/29/2013
-20 day: 12/20/2012
```

**Sample code 2**

```
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,60, "3",0,1,1, "-1 year: "+FORMAT$(DATEADD("yyyy", -1, "11/26/2012 10:08"),"yyyy/mm/dd hh:nn AM/PM")
TEXT 10,110,"3",0,1,1,"+9 months: "+FORMAT$(DATEADD("m",9,"11/26/2012 10:08"),"yyyy/mm/dd hh:nn AM/PM")
TEXT 10,160,"3",0,1,1,"+8 hours: "+FORMAT$(DATEADD("h", +8,"11/26/2012 10:08"),"yyyy/mm/dd hh:nn AM/PM")
TEXT 10,210,"3",0,1,1,"+00 day: "+FORMAT$("11/26/2012 10:08:00","yyyy/mm/dd hh:nn AM/PM")
TEXT 10,260,"3",0,1,1,"+20 days: "+FORMAT$(DATEADD("d",20,"11/26/2012 10:08"),"yyyy/mm/dd hh:nn AM/PM")
TEXT 10,310,"3",0,1,1,"-20 days: "+FORMAT$(DATEADD("d", -20,"11/26/2012 10:08"),"yyyy/mm/dd hh:nn AM/PM")
PRINT 1
```

**Result 2**

-1 year: 2011/11/26 10:08 AM  
+9 months: 2013/08/26 10:08 AM  
+8 hours: 2012/11/26 06:08 PM  
+00 day: 2012/11/26 10:08 AM  
+20 days: 2012/12/16 10:08 AM  
-20 days: 2012/11/06 10:08 AM



## FSEARCH()

### Description

This function returns the position of a string.

### Syntax

**FSEARCH(file handle, STR\$)**

Parameter	Description
file handle	0 or 1
STR\$	Required. Any valid string expression.

**Note:**

This command has been supported since A1.88 EZ and later firmware.

### Example

Sample code	Result
<pre>DOWNLOAD "DATA1",10,1234567890 DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO DOWNLOAD "Test.BAS" SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS OPEN "DATA1",0 OPEN "DATA2",1 TEXT 10,90,"4",0,1,1,"FSEARCH() FUNCTION TEST" A=FSEARCH(0,"8") B=FSEARCH(1,"J") TEXT 10,140,"3",0,1,1,"8 position is:"+STR\$(A) TEXT 10,180,"3",0,1,1,"J position is:"+STR\$(B) PRINT 1 EOP Test</pre>	<pre>FSEARCH() FUNCTION TEST 8 position is: 7 J position is: 9</pre>



# TOUCHPRESS()

## Description

This command is used to detect the status of touch screen. Returns 1 if the touch screen for the specified region is pressed, otherwise returns 0.

## Syntax

**TOUCHPRESS (left, top, right, bottom)**

## Note:

- This command has been supported since A1.76 EZ and later firmware
  - This command only can be performed on the printer with touch screen. 272(W) x 480(H) pixels for MT & MX series

## Example

## Sample code

```
DOWNLOAD "DEMO.BAS"
:START
IF TOUCHPRESS(0,90,272,120) <> 0 THEN GOTO A
GOTO START
ENDIF
:A
CLS
SIZE 4,1
GAP 0,0
DIRECTION 1
TEXT 30,30,"3",0,1,1,"TOUCH TEST!!"
PRINT 1,1
EOP
DEMO
```



## RECORDSET\$()

### Description

This function returns a value from a table. Table is represented in a grid format, tabular form in rows and columns. Please refer to following table format on example.

### Syntax

**RECORDSET\$(TABLE\$, ROW, COLUMN [ , DELIMITER])**

Parameter	Description
TABLE\$	Table name
ROW	Number of row
COLUMN	Number (or name) of column
DELIMITER	Optional. Set the delimiter of table. The default is 09H <Tab>

#### Note:

The Row is always a number. But the column can be a number or name

### Example

Sample code	Result
<pre> DOWNLOAD F,"TEST.CSV",75,3 Name,Age,Height,Weight John,18,180,80 Mary,30,150,50 Mark,65,170,65  DOWNLOAD F,"TEST.BAS" CLOSE 0 SIZE 4,2 GAP 0,0 DIRECTION 1 CLS  TEXT 100,50,"3",0,1,1,"Row 1 and Column 1 = " + RECORDSET\$("TEST.CSV", 1, 1, ASC(",")) TEXT 100,100,"3",0,1,1,"Row 2 and Column 1 = " + RECORDSET\$("TEST.CSV", 2, 1, ASC(",")) TEXT 100,150,"3",0,1,1,"John Age = " + RECORDSET\$("TEST.CSV", 1, 2, ASC(",")) TEXT 100,200,"3",0,1,1,"Mary Age = " + RECORDSET\$("TEST.CSV", 2, 2, ASC(",")) TEXT 100,250,"3",0,1,1,"John Height = " + RECORDSET\$("TEST.CSV", 1, "Height", ASC(",")) TEXT 100,300,"3",0,1,1,"Mary Height = " + RECORDSET\$("TEST.CSV", 2, "Height", ASC(",")) PRINT 1  EOP TEST </pre>	<pre> Row 1 and Column 1 = John Row 2 and Column 1 = Mary John Age = 18 Mary Age = 30 John Height = 180 Mary Height = 150 </pre>

Table format (TEST.CSV)				
Number of rows	3			
Name of column	Name	Age	Height	Weight
Row 1	John	18	180	80
Row 2	Mary	30	150	50
Row 3	Mark	65	170	65
	Column 1	Column 2	Column 3	Column 4



## Example

### Sample Code 2: (since VA1.97)

```

DOWNLOAD "TEST.CSV",121,6,
Number,String
1234,ABCD
"12,34","AB,CD"
"12
34","AB
CD"
"12""34","AB""CD"
""1234","","ABCD"
"1234","", "ABCD""

OUT RECORDSET$("TEST.CSV", 1, "Number", ASC(","))
OUT RECORDSET$("TEST.CSV", 2, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 3, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 4, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 5, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 6, 1, ASC(","))

OUT ""

OUT RECORDSET$("TEST.CSV", 1, "String", ASC(","))
OUT RECORDSET$("TEST.CSV", 2, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 3, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 4, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 5, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 6, 2, ASC(","))

```

### Table format (TEST.CSV)

Number of rows	<b>6</b>	
Name of column	Number	String
Row 1	1234	ABCD
Row 2	12,34	AB,CD
Row 3	12	AB
	34	CD
Row 4	12"34	AB"CD
Row 5	"1234	"ABCD
Row 6	1234"	ABCD"
	Column 1	Column 2

### Return

```

1234
12,34
12
34
12"34
"1234
1234"

ABCD
AB,CD
AB
CD
AB"CD
"ABCD
ABCD"

```



## LABELRATIO

### Description

This command returns label print ratio.

### Syntax

**LABELRATIO**

**Note:**

This command has been supported since V8.00 EZ and later firmware

### Example

Sample code	Result
<b>LABELRATIO</b>	<p>Width:4.25*203=864 High:8*203=1624</p> <p>width=864, high=1624 TotalPrintRate = 10880/1403136 bits (0.78%)</p>



## 8. DEVICE RECONFIGURATION COMMANDS

### SET COUNTER

#### Description

Counters can be a real counter or a variable. This setting sets the counter number in the program and its increments. There are three different types of counters: digit (0~9~0), lower case letter (a~z~a) or upper case letter (A~Z~A).

#### Syntax

**SET COUNTER @n step**  
@n= “Expression”

Parameter	Description
@n	n: counter number. There are 61 counters available (@0 ~ @60) in the printer. @0 to @50 will be cleared while restarting the printer. @51 to @60 will be stored in printer until the printer is restored to factory default. <i>@51~@55 were supported since V6.37 EZ.</i> <i>@56~@60 were supported since V6.74 EZ.</i>
step	The increment of the counter, can be positive or negative. -999999999 <= step <= 999999999 <i>If the counter is used as a fixed variable, please set the increment to 0.</i>
expression	Initial string. String length is 101 bytes



## Example

Sample code	Result
<pre>SET COUNTER @0 +1 SET COUNTER @1 +0 SET COUNTER @2 -1 SET COUNTER @3 1  @0=" 0001" @1=" 0101" @2=" 000A" @3=" 1"  SIZE 4,0.5 GAP 0,0 DIRECTION 1 CLS TEXT 600,10," 3",0,1,1,3," @0      @1      @2" TEXT 600,30," 3",0,1,1,3, « Label" +@3+ ----- TEXT 600,50,"3",0,1,1,3,@0+"      " +@1+      " +@2 PRINT 5</pre>	<pre>Label 5 -- @0      @1      @2           0005    0101    999W  Label 4 -- @0      @1      @2           0004    0101    999X  Label 3 -- @0      @1      @2           0003    0101    999Y  Label 2 -- @0      @1      @2           0002    0101    999Z  Label 1 -- @0      @1      @2           0001    0101    000A</pre>

## See Also

PRINT, TEXT, BARCODE



## SET CUTTER

### Description

This setting activates or deactivates the cutter and defines how many printed labels is to be cut at one time. This setting will be saved in printer memory after turning off the power.

### Syntax

**SET CUTTER OFF/BATCH/pieces**

Parameter	Description
OFF	Disable cutter function.
BATCH	Set printer to cut label at the end of printing job.
pieces	Set number of printing labels per cut. 0<= pieces <=65535

**Note:**

- Care label cutter module was supported since V6.86 EZ in industrial printer TTP-2410M series.
- Since V6.86 EZ, if cutter is not installed, the cutter error doesn't happen even SET CUTTER ON is set.

### Example

Sample code	Result
<b>SIZE 3,3</b> <b>GAP 0,0</b> <b>SET CUTTER OFF</b> <b>SET PEEL OFF</b> <b>CLS</b> <b>TEXT 50,50, "3",0,1,1, "SET CUTTER OFF"</b> <b>PRINT 3</b>	The cutter function is disabling.
<b>SET CUTTER BATCH</b> <b>CLS</b> <b>TEXT 50,50, "3",0,1,1, "SET CUTTER BATCH"</b> <b>PRINT 3,2</b>	The cutter cuts once after 6 labels are printed.
<b>SET CUTTER 1</b> <b>CLS</b> <b>TEXT 50,50, "3",0,1,1, "SET CUTTER 1"</b> <b>PRINT 3,2</b>	The cutter cuts every label.
<b>CLS</b> <b>TEXT 50,50, "3",0,1,1, "SET CUTTER 2"</b> <b>PRINT 3,2</b>	The cutter cuts every 2 labels.

### See Also

**OFFSET, PRINT, SET PARTIAL\_CUTTER**



## SET PARTIAL\_CUTTER

### Description

This setting activates or deactivates the cutter and defines how many printed labels is to be cut at one time. This setting will be saved in printer memory after turning off the power. This function prevents label back feeding after a cut.

### Syntax

**SET PARTIAL\_CUTTER OFF/BATCH/Pieces**

Parameter	Description
OFF	Disable cutter function.
BATCH	Set printer to cut label at the end of printing job.
pieces	Set number of printing labels per cut. 0<= pieces <=65535

**Note:**

This command is supported for the printer that have cutter module.

### Example

**Sample code**

```
REM **SET PARTIAL_CUTTER FUNCTION OFF EXAMPLE PROGRAM**
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET PARTIAL_CUTTER OFF
CLS
TEXT 50,50, "3",0,1,1, "SET PARTIAL_CUTTER OFF"
PRINT 3
REM ***This program cuts once at the batch***
SET PARTIAL_CUTTER BATCH
CLS
TEXT 50,50, "3",0,1,1, "SET PARTIAL_CUTTER BATCH"
PRINT 3,2
REM ***This program cuts every label***
SET PARTIAL_CUTTER 1
CLS
```



```
TEXT 50,50, "3",0,1,1, " SET PARTIAL_CUTTER 1"
PRINT 3,2
REM ***This program cuts 2 label ***
SET PARTIAL_CUTTER 2
CLS
TEXT 50,50, "3",0,1,1, "SET PARTIAL_CUTTER 2"
PRINT 3,2
```

## See Also

OFFSET, PRINT, SET CUTTER



## SET BACK

### Description

This setting is used after SET CUTTER function. This function prevents label backfeeding after a cut.

### Syntax

**SET BACK OFF/ON**

Parameter	Description
OFF	Disable back function.
ON	Enable back function.

### Example

#### Sample code

```
REM **SET BACK FUNCTION OFF EXAMPLE PROGRAM**
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 1
REFERENCE 0,0
SET CUTTER 1
SET BACK OFF
CLS
TEXT 50,50, "3",0,1,1, "SET BACK OFF "
PRINT 3
CLS
SET CUTTER 1
SET BACK ON
TEXT 50,50, "3",0,1,1, "SET BACK ON "
PRINT 3
```

### See Also

OFFSET, PRINT, SET CUTTER



## SET KEYn

### Description

This setting is used to enable/disable the KEYn function. Before setting KEYn function, please disable the default function of KEYn first. The setting will remain resident in the printer even when the printer is power off.

### Syntax

**SET KEYn ON/OFF/DEFAULT/MENU/PAUSE/PRINT m/FEED/BACKFEED/FORMFEED/CUT/INPUT "string "**

Parameter	Description
n	0, 1, 2, 3, 4, 5, 6
ON	Enable KEYn function
OFF	Disable KEYn function
DEFAULT	Resume KEYn default function
MENU	Set to "MENU " key
PAUSE	Set to "PAUSE " key
PRINT m	Set to "PRINT " key m: Set number of printing labels per print. (0 < m < 32000)
FEED	Set to "FEED " key that can manual control the feeding distance
BACKFEED	Set to "BACKFEED " key that can manual control the backfeeding distance
FORMFEED	Set to "FORMFEED " key that will feed the label under the format. Ex: If format is "size 4,6, it will feed 6 ".
CUT	Set to "CUT " key
INPUT "string "	Send the command by press key (ex: <b>SET KEY1 INPUT "CONFIG " + CHR\$(13) + CHR\$(10)</b> )

The default function of KEYn id as listed below:

Model	KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6
PrintEasy		FEED					
PrintHybrid		FEED	MENU	UP	RIGHT	LEFT	DOWN



## Example

### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET KEY1 OFF
SET KEY2 OFF
SET KEY3 OFF
KEY1=0
KEY2=0
KEY3=0

:START
IF KEY1=1 THEN
CLS
TEXT 100,10, "3",0,1,1, "KEY1 (MENU key) is pressed!! "
PRINT 1,1
ELSEIF KEY2=1 THEN
CLS
TEXT 100,10, "3",0,1,1, "KEY2 (PAUSE key) is pressed!! "
PRINT 1,1
ELSEIF KEY3=1 THEN
CLS
TEXT 100,10, "3",0,1,1, "KEY3 (FEED key) is pressed!! "
TEXT 100,60, "3",0,1,1, "End of test"
PRINT 1,1
SET KEY1 ON
SET KEY2 ON
SET KEY3 ON
END
ENDIF
GOTO START
EOP
DEMO
```

## See Also

OFFEST, PRINT



## SET LEDn

### Description

This setting is used to control LED on/off function.

### Syntax

**SET LED1 ON/OFF**

**SET LED2 ON/OFF**

**SET LED3 ON/OFF**

LED no.	Default Function
LED1	Power on/off
LED2	Printer on-line/off-line
LED3	Error/normal
Parameter	Description
ON	Enable LEDn function
OFF	Disable LEDn function

The default function of LED1, LED2 and LED3 id as listed below:

Model	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED2 & LED3
PrintEasy, PrintHybrid	GREEN	GREEN	RED					ORANGE

### Example

#### Sample code

```

DOWNLOAD "DEMO4.BAS "
SET LED1 OFF
SET LED2 OFF
SET LED3 OFF
FOR I=1 TO 100
LED1=0
LED2=0
LED3=0
IF I-INT(I/2)*2=0 THEN
LED1=1
ELSEIF I-INT(I/3)*3=0 THEN
LED2=1
ELSE
LED3=1
ENDIF
NEXT
LED1=1
LED2=1
LED3=0
SET LED1 ON
SET LED2 ON
SET LED3 ON
EOP
DEMO4

```



## SET PEEL

### Description

This setting is used to enable/disable the self-peeling function. The default setting for this function is off. When this function is set on, the printer stops after each label printing, and does not print the next label until the peeled label is taken away. This setting will be saved in printer memory when turning off the power.

### Syntax

**SET PEEL ON/OFF**

Parameter	Description
ON	Enable the self-peeling function
OFF	Disable the self-peeling function

### Example

#### Sample code

```
REM ***SELF-PEELING FUNCTION ON***  
SIZE 4,4  
GAP 0,0  
DENSITY 8  
SPEED 6  
DIRECTION 0  
REFERENCE 0,0  
SET CUTTER OFF  
SET PEEL ON  
CLS  
TEXT 50,100, "3 ",0,1,1, "SELF-PEELING FUNCTION TEST "  
PRINT 5
```

### See Also

OFFSET, PRINT



## SET REWIND

### Description

This setting is used to enable/disable the internal rewind function for MX240/TTP-2610MT series & external rewind module (via RS-232 port). The default setting for this function is off. When this function is set on, the printer rewind spindle will rewind the printed labels. This setting will be saved in printer memory when turning off the power.

### Syntax

**SET REWIND ON/OFF/RS232**

Parameter	Description
ON	Enable the internal rewind function
OFF	Disable the internal rewind or external rewind module function
RS232	Enable the external rewind module function (via RS-232 port / pull high signal)

**Note:**

The function of external rewind module has been supported since A1.92 and later firmware.

### Example

**Sample code**

```
REM ***REWIND FUNCTION ON***  
SIZE 4,4  
GAP 0.12,0  
DENSITY 8  
SPEED 6  
DIRECTION 0  
REFERENCE 0,0  
SET CUTTER OFF  
SET REWIND ON  
CLS  
TEXT 50,100, "3 ",0,1,1, "REWIND FUNCTION TEST "  
PRINT 500
```

### See Also

OFFSET, PRINT



## SET TEAR & SET STRIPER

### Description

This command is used to enable/disable feeding of labels to gap/black mark position for tearing off. This setting will be saved in printer memory when turning off the power.

### Syntax

SET TEAR ON/OFF        (TSPL2 language printers only)  
SET STRIPER ON/OFF    (TSPL language printers only)

Parameter	Description
ON	The label gap will stop at the tear off position after print.
OFF	The label gap will NOT stop at the tear off position after print. The beginning of label will be aligned to print head.

### Example

#### Sample code

```
REM ***TEAR FUNCTION ON***  
SIZE 3,3  
GAP 0.08,0  
DENSITY 8  
SPEED 4  
DIRECTION 0  
REFERENCE 0,0  
SET CUTTER OFF  
SET PEEL OFF  
SET TEAR ON  
CLS  
TEXT 50,100, "3 ",0,1,1, "TEAR FUNCTION TEST "  
PRINT 1
```

### See Also

SET PEEL, SET CUTTER



## SET GAP

### Description

This setting sets the gap sensor emission sensitivity. The printer initiates automatic gap sensor calibration when the PAUSE key is held down while powering up. This function may cease to work if the thickness of the backing paper and that of label with backing paper are not of appreciable difference to the sensor, or when there are pre-printed marks or patterns on the label. In such case, users must calibrate the gap sensor manually by this command through trial-and-error method to attain the proper setting. This setting will be saved in printer memory when turning off the power.

### Syntax

**SET GAP n/AUTO/OFF/0,/REVERSE/OBVERSE**

Parameter	Description
n	Gap sensor light emission strength. Available range is listed as below. 0 is the lowest sensitivity
AUTO	The printer will feed 2 or 3 labels to calibrate the gap. If the label is continuous, the printer will feed label to limit 10~20 inches to confirm if the label is continuous.
OFF	Disable the SET GAP AUTO function.
0,	Automatically calibrate the gap size.
REVERSE	This function is used when the Black Mark is the separation in the front of the label and which can't be detected by the Black Mark sensor. The parts of the media which can be passed through by GAP sensor are defined to be the printable area, otherwise it will be defined to the GAP of the media.
OBVERSE	Disable the "SET GAP REVERSE " function.

**Note:**

**When in "SET HEAD OFF " mode, the function "SET GAP AUTO " doesn't work even the printer head is opened and closed, but it can work when power on the printer.**



## Example

The example below is operated in DOS environment via the parallel port connection to setup the label size, gap distance and sensor sensitivity.

```
C:\>COPY CON LPT1<ENTER>
SIZE 4,2.5<ENTER>
GAP 0.12,0<ENTER>
SET GAP 1<ENTER>
<CTRL><Z><ENTER>
C:\>
```

**Note:**

<ENTER> stands for keyboard “ENTER” key. In the above example, please press “ENTER” key instead of typing <ENTER> in the above example. <CTRL> stands for keyboard “Ctrl” key.

**Troubleshooting:**

Press the FEED key to test. Does printer stop at the same position on each label without the error light blinking? If not, adjust the setting to a larger number. When adjusting this setting, begin from 0 and then on to higher values incrementally.

## See Also

SIZE, GAP, BLINE



## SET BLINE

### Description

This setting is used to reverse/obverse the sensor function.

### Syntax

**SET BLINE REVERSE/OBVERSE**

Parameter	Description
REVERSE	Reverse the sensor function. Redefine the reflective area is black line and non-reflective part is paper. (Normally, reflective part is paper and non-reflective part is black line.)
OBVERSE	Disable the "SET BLINE REVERSE" function.



## SET HEAD

### Description

This setting is used to enable/disable head open sensor. If the head open sensor is turned off, an open printer head will not return an error message. This setting will be saved in printer memory. This command is only available for TSPL2 printers.

### Syntax

**SET HEAD ON /OFF**

Parameter	Description
ON	Turn on the “HEAD OPEN” sensor
OFF	Turn off the “HEAD OPEN” sensor

### Example

**SET HEAD ON**

**SET HEAD OFF**



## SET RIBBON

### Description

This setting is used to enable/disable ribbon sensor detection. (Thermal Transfer Printing/Thermal Direct Printing) Printer will detect the presence of a ribbon to determine using either direct thermal or thermal transfer printing upon printer startup. This setting will NOT be saved in printer memory.

### Syntax

**SET RIBBON ON/OFF/INSIDE/OUTSIDE**

Parameter	Description
ON	Thermal transfer printing
OFF	Thermal direct printing
INSIDE	The ribbon is inside wound. For TTP-384M only. *Since V6.80EZ.
OUTSIDE	The ribbon is outside wound. For TTP-384M only. *Since V6.80EZ.

### Example

#### Sample code

```
REM *****Disable ribbon detection sensor for direct thermal printing.  
SET RIBBON OFF  
SIZE 4,1  
GAP 0,0  
CLS  
TEXT 10,10, " 3 " ,0,1,1, " Direct thermal printing. "  
PRINT 1  
  
REM *****Enable ribbon detection sensor for thermal transfer printing.  
SET RIBBON ON  
SIZE 4,1  
GAP 0,0  
CLS  
TEXT 10,10, " 3 " ,0,1,1, " Thermal transfer printing. "  
PRINT 1
```



```
REM *****For using ink-in ribbon in TTP-384M.  
SET RIBBON INSIDE  
SIZE 4,1  
GAP 0,0  
CLS  
TEXT 10,10, " 3 " ,0,1,1, " TTP-384M is using ink-in ribbon. "  
PRINT 1  
  
REM *****For using ink-out ribbon in TTP-384M.  
SET RIBBON OUTSIDE  
SIZE 4,1  
GAP 0,0  
CLS  
TEXT 10,10, " 3 " ,0,1,1, " TTP-384M is using ink-out ribbon. "  
PRINT 1
```



## SET ENCODER

### Description

This setting is used to enable/disable ribbon encoder sensor detection.

### Syntax

**SET ENCODER ON/OFF**

Parameter	Description
ON	Enable ribbon encoder sensor.
OFF	Disable ribbon encoder sensor.

### Example

**SET ENCODER ON**

**SET ENCODER OFF**



## SET RIBBONEND

### Description

This setting is used to enable/disable ribbon-end sensor detection.

### Syntax

**SET RIBBONEND ON/OFF**

Parameter	Description
ON	Enable ribbon-end sensor.
OFF	Disable ribbon-end sensor.

**Note:**

This command has been supported since V6.91 EZ and later firmware.

### Example

**SET RIBBONEND ON**  
**SET RIBBONEND OFF**



## SET COM1

### Description

This setting defines communication parameters for printer serial port.

### Syntax

**SET COM1 baud,parity,data,stop**

Parameter	Description
baud	Baud rate, available baud rates are as listed: 24: 2400 bps 48: 4800 bps 96: 9600 bps 19: 19200 bps 38: 38400 bps 57: 57600 bps 115: 115200 bps
parity	Parity check N: No parity check E: Even parity check O: Odd parity check
Data	Data bit 8: 8 bits data 7: 7 bits data
stop	Stop bit 1: 1 stop bit 2: 2 stop bits

### Example

The parallel port is used to setup the printer serial port in this example via MS-DOS mode.

```
C:\>COPY CON LPT1<ENTER>
SET COM1 19,N,8,1<ENTER>
<CTRL><Z><ENTER>
C:\>
```

**Note:**

<ENTER> stands for PC keyboard “ENTER” key. <CTRL><Z> means to hold PC keyboard “CTRL” key then press the PC keyboard <Z> key.



## SET PRINTKEY

### Description

This command will print one label and feed label gap to tear bar position for tearing away. Press FEED button to print the next label or batch of labels. If label content includes serial text or barcode, it will change the serial number accordingly. This setting will be saved in printer memory.

### Syntax

**SET PRINTEKY OFF/ON/AUTO/<num>**

Parameter	Description
OFF	Disable this function
ON	Enable this function
AUTO	Enable this function
<num>	Numbers of labels will be printed if FEED button is pressed.

### Example

#### Sample code

```
SIZE 4,2.5
GAP 0.12,0
SET PRINTKEY ON
SET COUNTER @0 1
@0= "0001"
CLS
TEXT 10,10, "5",0,1,1,@0
PRINT 1
```



## Example

Execute:

Syntax	Receive "PRINT m"	Print Out
SET PRINTKEY ON or	1.) PRINT 2	Label 1~2
SET PRINTKEY AUTO	2.) Press FEED key	Label 3~4
Syntax	Receive "PRINT m,n"	Print Out
SET PRINTKEY ON or	1.) PRINT 1,2	Label 1, Label 1
SET PRINTKEY AUTO	2.) Press FEED key	Label 2, Label 2
Syntax	Receive "PRINT -1,n"	Print Out
SET PRINTKEY ON or	1.) PRINT -1,2	Label 1, Label 1
SET PRINTKEY AUTO	2.) Press FEED key	Label 1, Label 1
Syntax	Receive "PRINT m"	Print Out
SET PRINTKEY 5	1.) PRINT 2	Label 1~2
	2.) Press FEED key	Label 3~7
Syntax	Receive "PRINT m,n"	Print Out
SET PRINTKEY 5	1.) PRINT 1,2	Label 1, Label 1
	2.) Press FEED key	Label 2~6
Syntax	Receive "PRINT -1,n"	Print Out
SET PRINTKEY 5	1.) PRINT -1,2	Label 1, Label 1
	2.) Press FEED key	Label 1, Label 1



## SET REPRINT

### Description

This command will disable/enable a reprinting attempt subsequent to a “no paper”, “no ribbon” or “carriage open” error.

### Syntax

**SET REPRINT OFF/ON**

Parameter	Description
OFF	Disable this function
ON	Enable this function

### Example

**SET REPRINT ON**



## SET FEED\_LEN

### Description

This command can set the feeding length when FEED key is pressed. This setting will be memorized by printer. The initialized value is the label length.

### Syntax

**SET FEED\_LEN n**

Parameter	Description
n	The feeding length in dot.

**Note:**

This command has been supported since V5.10 EZ and later firmware.

### Example

**Sample code**

**SET FEED\_LEN 100**

**Result**

The feeding length is 100 dots when you press the FEED button after this setting.



## GETSENSOR()

### Description

This command is used to get the sensor status/AD value. We can use it to check the sensor function.

### Syntax

**GETSENSOR(sensor\$[, intension])**

Parameter	Description
sensor\$	Sensor type.  <b>GAP</b> Gap sensor <b>BLINE</b> Black mark sensor <b>RIBBON</b> Ribbon-end sensor <b>PEEL</b> Peeler sensor <b>HEAD UP</b> Thermal print head open sensor <b>HEAD TEMP</b> The temperature of print head <b>HEAD VOLT</b> The voltage of print head
intension	Sensor intension.  <b>GAP</b> intension Please refer to SET GAP for gap sensor range of different model. <b>BLINE</b> intension Please refer to SET GAP for black mark sensor range of different model. <b>RIBBON</b> intension 0 ~ 3 <b>PEEL</b> sensor intension Ignored <b>HEAD UP</b> sensor intension Ignored <b>HEAD TEMP</b> Ignored <b>HEAD VOLT</b> Ignored
returned value	<b>GAP</b> Return the AD value of gap sensor <b>BLINE</b> Return the AD value of black mark sensor <b>RIBBON</b> Return the AD value of ribbon sensor <b>PEEL</b> The return value will be either 0 or 1 0: Paper is not on the sensor 1: Paper is on the sensor <b>HEAD UP</b> The return value will be either 0 or 1 0: print head module is close 1: print head module is open <b>HEAD TEMP</b> Return the temperature of thermal print head <b>HEAD VOLT</b> Return the voltage of thermal print head

#### Note:

This command has been supported since V6.75 EZ and later firmware.

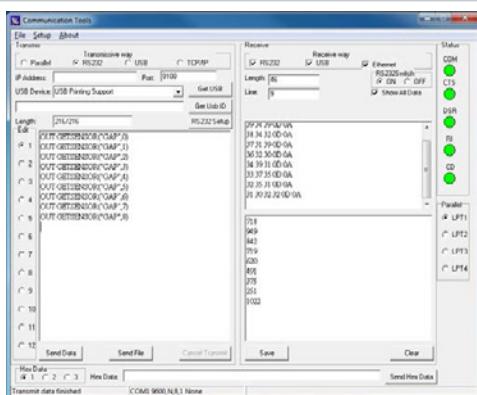


## Example

### Sample code

```
OUT GETSENSOR("GAP",0)
OUT GETSENSOR("GAP",1)
OUT GETSENSOR("GAP",2)
OUT GETSENSOR("GAP",3)
OUT GETSENSOR("GAP",4)
OUT GETSENSOR("GAP",5)
OUT GETSENSOR("GAP",6)
OUT GETSENSOR("GAP",7)
OUT GETSENSOR("GAP",8)
```

### Result

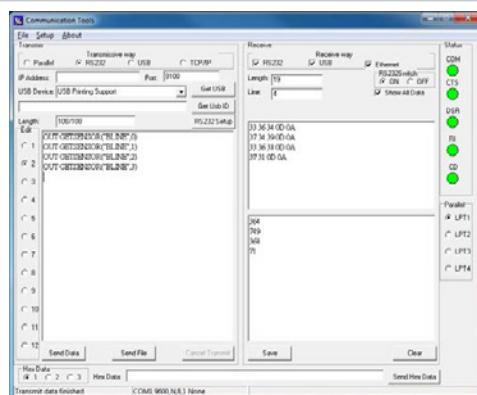


\*If the returned valued is changed in different sensor intension, we can say the sensor is functional.

### Sample code

```
OUT GETSENSOR("BLINE",0)
OUT GETSENSOR("BLINE",1)
OUT GETSENSOR("BLINE",2)
OUT GETSENSOR("BLINE",3)
```

### Result

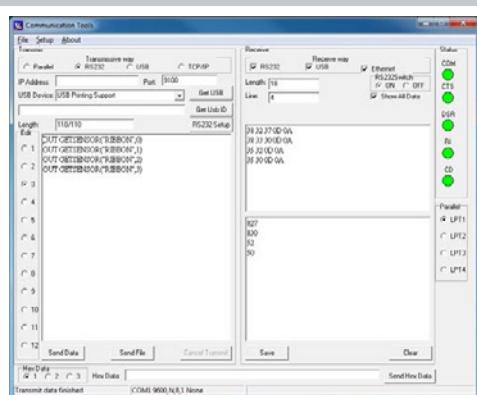


\*If the returned valued is changed in different sensor intension, we can say the sensor is functional.

### Sample code

```
OUT GETSENSOR("RIBBON",0)
OUT GETSENSOR("RIBBON",1)
OUT GETSENSOR("RIBBON",2)
OUT GETSENSOR("RIBBON",3)
```

### Result

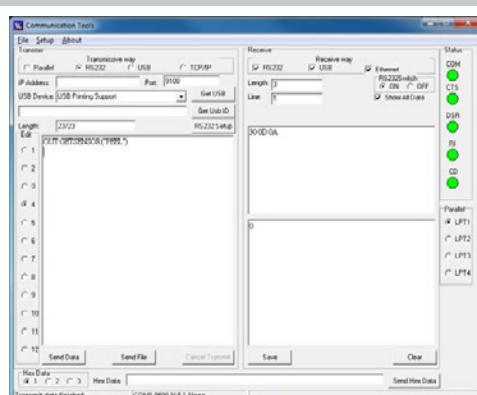


\*If the returned valued is changed in different sensor intension, we can say the sensor is functional.

### Sample code

```
OUT GETSENSOR("PEEL")
```

### Result



\*0: Paper is not on the sensor. 1: Paper is on the sensor.



## GETSETTING\$()

### Description

This command is used to get printer settings.

### Syntax

**GETSETTING\$ (app\$, sec\$, key\$[, default\$])**

app\$	sec\$	key\$	Comment
SYSTEM	INFORMATION	DPI	Return printer resolution
		MODEL	Return printer model name
		SERIAL	Return Printer serial number
		VERSION	Return Printer firmware version
		CHECKSUM	Return Printer firmware checksum
	RECORD	MILAGE	Return printed mileage
		CUT COUNTER	Return cutter cuts
	DRAM	CAPACITY	Return the total capacity of DRAM
		AVAILABLE	Return the available capacity of DRAM
FILE	FLASH	CAPACITY	Return the total capacity of FLASH
		AVAILABLE	Return the available capacity of FLASH
	CARD	CAPACITY	Return the total capacity of CARD
		AVAILABLE	Return the available capacity of CARD
		INSTALLED	Return the status of card. 1: installed; 0: none installed.
CONFIG	NET	MAC ADDRESS	Return MAC address
		IP ADDRESS	Return IP address
		SUBNET MASK	Return Subnet Mask
		DEFAULT GATEWAY	Return default gateway
		RAW PORT	Return raw port
		NAME	Return printer name
	WLAN	MAC ADDRESS	Return MAC address
		IP ADDRESS	Return IP address
		SUBNET MASK	Return Subnet Mask
		DEFAULT GATEWAY	Return default gateway
	COM1	RAW PORT	Return raw port
		BAUD RATE	Return baud rate of COM port
		DATA BIT	Return data bit of COM port
		PARITY	Return parity of COM port
		STOP BIT	Return stop bit of COM port



## Syntax

app\$	sec\$	key\$	Comment
CONFIG	SENSOR	SENSOR TYPE	Return the current sensor type
		CARRIAGE	Return the status of head open sensor.
	TSPL	GAP INTENSION	Return intension of gap sensor.
		BLINE INTENSION	Return intension of black mark sensor.
		CONTINUOUS INTENSION	Return intension of continuous sensor.
	TSPL	PRINT MODE	Return pos-print action.
		DENSITY	Return print density
		PAPER SIZE	Return paper size
		GAP SIZE	Return gap size
		BLINE SIZE	Return black mark size
		DIRECTION	Return printing direction
		MIRROR	Return mirror status.
		RIBBON	Return ribbon status.
		REPRINT	Return reprint status.
		PAPER WIDTH	Return paper width
TSPL	TSPL	LIMIT FEED	Return maximum length for sensor calibration.
		OFFSET	Return OFFSET value.
		REFERENCE X	Return REFERENCE X value.
		REFERENCE Y	Return REFERENCE Y value.
		SHIFT X	Return SHIFT X value.
		SHIFT Y	Return SHIFT Y value.
		SPEED	Return print speed.
		COUNTRY CODE	Return COUNTRY code.
		CODEPAGE	Return CODEPAGE.

Parameter	Description
default\$	Optional. Expression containing the value to return if no value is set in the key\$ setting. If omitted, default is assumed to be a zero-length string ("").

**Note:**

This command has been supported since V6.72 EZ and later firmware.



## Example

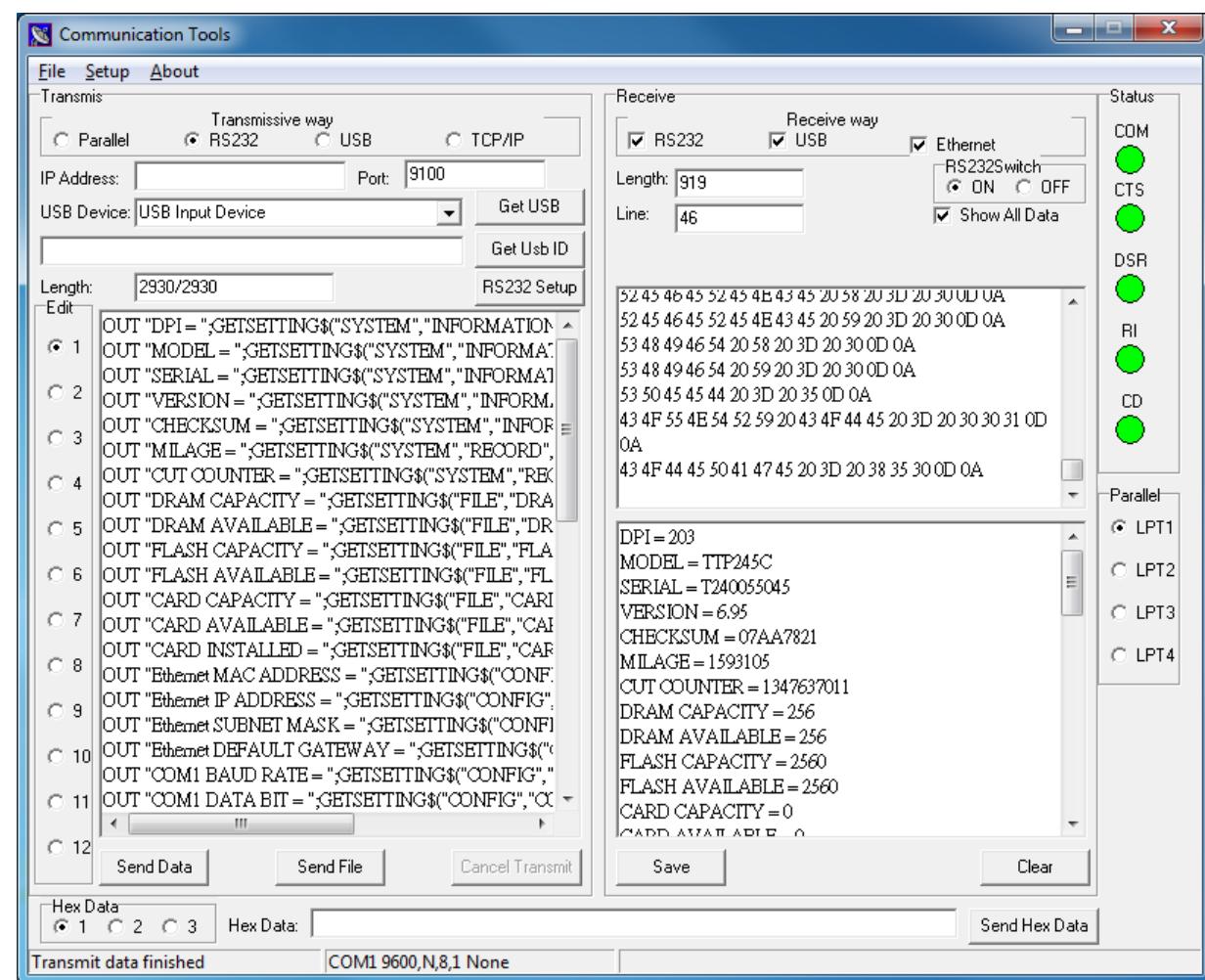
Sample code (Use CommTool to get printer settings via RS-232.)

```
OUT "DPI = ";GETSETTING$("SYSTEM", "INFORMATION", "DPI")
OUT " MODEL =";GETSETTING$("SYSTEM ", " INFORMATION ", " MODEL")
OUT "SERIAL = ";GETSETTING$("SYSTEM", "INFORMATION", "SERIAL")
OUT "VERSION = ";GETSETTING$("SYSTEM", "INFORMATION", "VERSION")
OUT "CHECKSUM = ";GETSETTING$("SYSTEM", "INFORMATION", "CHECKSUM")
OUT "MILAGE = ";GETSETTING$("SYSTEM", "RECORD", "MILAGE")
OUT "CUT COUNTER = ";GETSETTING$("SYSTEM", "RECORD", "CUT COUNTER")
OUT "DRAM CAPACITY = ";GETSETTING$("FILE", "DRAM", "CAPACITY")
OUT "DRAM AVAILABLE = ";GETSETTING$("FILE", "DRAM", "AVAILABLE")
OUT "FLASH CAPACITY = ";GETSETTING$("FILE", "FLASH", "CAPACITY")
OUT "FLASH AVAILABLE = ";GETSETTING$("FILE", "FLASH", "AVAILABLE")
OUT "CARD CAPACITY = ";GETSETTING$("FILE", "CARD", "CAPACITY")
OUT "CARD AVAILABLE = ";GETSETTING$("FILE", "CARD", "AVAILABLE")
OUT "CARD INSTALLED = ";GETSETTING$("FILE", "CARD", "INSTALLED")
OUT "Ethernet MAC ADDRESS = ";GETSETTING$("CONFIG", "NET", "MAC ADDRESS")
OUT "Ethernet IP ADDRESS = ";GETSETTING$("CONFIG", "NET", "IP ADDRESS")
OUT "Ethernet SUBNET MASK = ";GETSETTING$("CONFIG", "NET", "SUBNET MASK")
OUT "Ethernet DEFAULT GATEWAY = ";GETSETTING$("CONFIG", "NET", "DEFAULT GATEWAY")
OUT "COM1 BAUD RATE = ";GETSETTING$("CONFIG", "COM1", "BAUD RATE")
OUT "COM1 DATA BIT = ";GETSETTING$("CONFIG", "COM1", "DATA BIT")
OUT "COM1 PARITY = ";GETSETTING$("CONFIG", "COM1", "PARITY")
OUT "COM1 STOP BIT = ";GETSETTING$("CONFIG", "COM1", "STOP BIT")
OUT "SENSOR TYPE = ";GETSETTING$("CONFIG", "SENSOR", "SENSOR TYPE")
OUT "CARRIAGE = ";GETSETTING$("CONFIG", "SENSOR", "CARRIAGE")
OUT "GAP INTENSION = ";GETSETTING$("CONFIG", "SENSOR", "GAP INTENSION")
OUT "BLINE INTENSION = ";GETSETTING$("CONFIG", "SENSOR", "BLINE INTENSION")
OUT "CONTINUOUS INTENSION = ";GETSETTING$("CONFIG", "SENSOR", "CONTINUOUS INTENSION")
OUT "PRINT MODE = ";GETSETTING$("CONFIG", "TSPL", "PRINT MODE")
OUT "DENSITY = ";GETSETTING$("CONFIG", "TSPL", "DENSITY")
OUT "PAPER SIZE = ";GETSETTING$("CONFIG", "TSPL", "PAPER SIZE")
OUT "GAP SIZE = ";GETSETTING$("CONFIG", "TSPL", "GAP SIZE")
OUT "BLINE SIZE = ";GETSETTING$("CONFIG", "TSPL", "BLINE SIZE")
OUT "DIRECTION = ";GETSETTING$("CONFIG", "TSPL", "DIRECTION")
OUT "MIRROR = ";GETSETTING$("CONFIG", "TSPL", "MIRROR")
OUT "RIBBON = ";GETSETTING$("CONFIG", "TSPL", "RIBBON")
OUT "REPRINT = ";GETSETTING$("CONFIG", "TSPL", "REPRINT")
OUT "PAPER WIDTH = ";GETSETTING$("CONFIG", "TSPL", "PAPER WIDTH")
OUT "LIMIT FEED = ";GETSETTING$("CONFIG", "TSPL", "LIMIT FEED")
OUT "OFFSET = ";GETSETTING$("CONFIG", "TSPL", "OFFSET")
OUT "REFERENCE X = ";GETSETTING$("CONFIG", "TSPL", "REFERENCE X")
OUT "REFERENCE Y = ";GETSETTING$("CONFIG", "TSPL", "REFERENCE Y")
OUT "SHIFT X = ";GETSETTING$("CONFIG", "TSPL", "SHIFT X")
OUT "SHIFT Y = ";GETSETTING$("CONFIG", "TSPL", "SHIFT Y")
OUT "SPEED = ";GETSETTING$("CONFIG", "TSPL", "SPEED")
OUT "COUNTRY CODE = ";GETSETTING$("CONFIG", "TSPL", "COUNTRY CODE")
OUT "CODEPAGE = ";GETSETTING$("CONFIG", "TSPL", "CODEPAGE")
```



## Example

### Result



### Sample code(NET, WLAN)

```

OUT "Ethernet DEFAULT RAW PORT = ";GETSETTING$("CONFIG","NET","RAW PORT")

OUT "WLAN MAC ADDRESS = ";GETSETTING$("CONFIG", "WLAN", "MAC ADDRESS")
OUT "WLAN IP ADDRESS = ";GETSETTING$("CONFIG", "WLAN", "IP ADDRESS")
OUT "WLAN SUBNET MASK = ";GETSETTING$("CONFIG", "WLAN", "SUBNET MASK")
OUT "WLAN DEFAULT GATEWAY = ";GETSETTING$("CONFIG", "WLAN", "DEFAULT GATEWAY")
OUT "WLAN DEFAULT RAW PORT = ";GETSETTING$("CONFIG", "WLAN", "RAW PORT")
OUT "NET Name = ";GETSETTING$("CONFIG", "NET", "NAME")

```



## SET USBHOST

### Description

This command can set the USB host for the usage of USB keyboard or scanner.

### Syntax

**SET USBHOST KEYBOARD/SCANNER**

Parameter	Description
KEYBOARD	USB keyboard (Enable the prompt shown on LCD)
SCANNER	USB scanner (Disable the prompt shown on LCD)

**Note:**

- This command has been supported since V6.95 EZ and later firmware.
- This command is for the model which has USB HOST connector.

### Example

**Sample code**

```
SET USBHOST KEYBOARD
DOWNLOAD "A.BAS "
LOOP:
SIZE 4,2
GAP 0,0
CLS
INPUT A$
TEXT 50,50, "0 ",0,20,20,A$
PRINT 1
GOTO LOOP
EOP
A.BAS
```



## SET RS232\_REWINDER

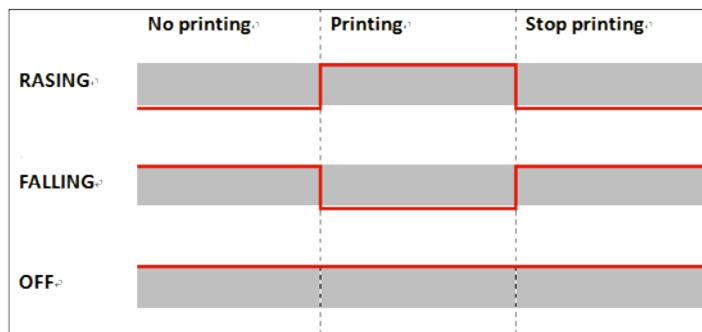
### Description

This command is used to set the CTS (PIN8) control signal to High or Low status via RS-232 port for external rewind module application.

### Syntax

**SET RS232\_REWINDER “RASING”/ “FALLING”/ “OFF”**

Parameter	Description
“RASING”	Set high level signal status while printer is printing
“FALLING”	Set low level signal status while printer is printing
“OFF”	Disable this function (Default)



#### Note:

- This command has been supported since V8.01 EZ & A1.85 EZ and later firmware.
- This command is for the model which has RS-232 connector.

### Example

#### Sample code

```
SET RS232_REWINDER “RASING”
```



## SET AUTORUN

### Description

This command redefines the BAS file which can be run automatically while switching on the printer. Default is AUTO.BAS.

### Syntax

**SET AUTORUN “filename”**

Parameter	Description
filename	The file will be defined to AUTO-RUN file. Default is AUTO.BAS.

**Note:**

This command has been supported since V6.86 EZ and later firmware.

### Example

**Sample code**

```
REM *****Step1: Send the following command to redefine the auto-run file from  
“AUTO.BAS” to “TEST.BAS”  
SET AUTORUN “TEST.BAS”  
  
REM *****Step2: Send the following commands to download “TEST.BAS” file into  
printer.  
DOWNLOAD F, “TEST.BAS”  
SIZE 4,1  
GAP 0,0  
DIRECTION 1  
CLS  
BLOCK 10,10,600,200, “3”,0,1,1,12, “[ ]TEST.BAS\[ ] is running automatically  
while turning on the printer.”  
PRINT 1  
EOP  
  
REM *****Step3: Turn off and on the printer to run “TEST.BAS” automatically.
```

**Result**

“TEST.BAS” is running automatically  
while turning on the printer.



## SET VERIFIER

### Description

This command is used to add a data at start printing or end printing then return it via RS-232 port for verifier (scanner) application.

### Syntax

```
SET VERIFIER ON/OFF
VERIFYTRIGGER CHR$( )+CHR$( )...
VERIFYEND CHR$( )+CHR$( )...
```

Parameter	Description
ON	Enable verifier function
OFF	Disable verifier function
VERIFYTRIGGER	Add data at start printing
VERIFYEND	Add data at end printing

#### Note:

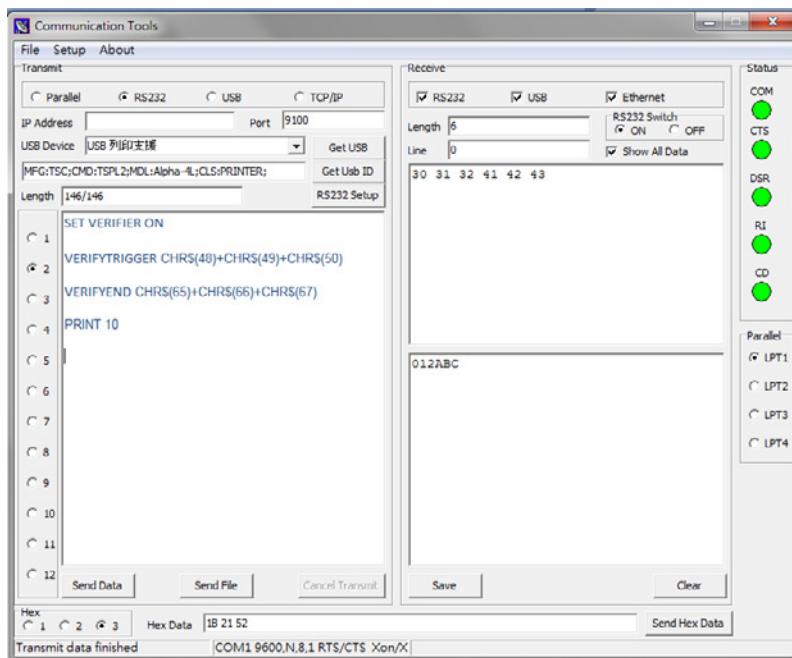
This command has been supported since V6.92 EZ and later firmware.

### Example

#### Sample code

```
SET VERIFIER ON
VERIFYTRIGGER CHR$(48)+CHR$(49)+CHR$(50)
VERIFYEND CHR$(65)+CHR$(66)+CHR$(67)
PRINT 10
```

#### Result





## SET RESPONSE

### Description

This command can response issue automatically.

### Syntax

**SET RESPONSE [“Job ID”,] ON/OFF/BATCH**

Parameter	Description
[“Job ID”]	Optional. Set job ID. Default is Null
ON	Enable this function
OFF	Disable this function. Default is OFF
BATCH	Response at the end of printing job

**Note:**

This command has been supported since V7.09 EZ and later firmware.

### Response Syntax

**{Status,#####,ID}**

Status
[Hex Receive]
00 Normal
01 Head opened
02 Paper Jam
03 Paper Jam and head opened
04 Out of paper
05 Out of paper and head opened
08 Out of ribbon
09 Out of ribbon and head opened
0A Out of ribbon and paper jam
0B Out of ribbon, paper jam and head opened
0C Out of ribbon and out of paper
0D Out of ribbon, out of paper and head opened
10 Pause
20 Printing
80 Other error
#####: 00001 ~ 99999



## Example

### Sample code

```
SET RESPONSE ON
SIZE 4,2
GAP 0,0
PRINT 3

{00,00001}{00,00002}{00,00003}
```

### Sample code

```
SET RESPONSE "ID1",ON
SIZE 4,2
GAP 0,0
PRINT 3,2

{00,00001,ID1}{00,00002,ID1}{00,00003,ID1}{00,00004,ID1}{00,00005,ID1}
{00,00006,ID1}
```

### Sample code

```
SET RESPONSE "CCCC ",BATCH
SIZE 4,2
GAP 0,0
PRINT 3,2

{00,00006,CCCC}
```



## SET DAYLIGHT\_SAVE

### Description

This command is used to set daylight saving time.

### Syntax

```
SET DAYLIGHT_SAVE ON/OFF
SET DAYLIGHT_SAVE "Start", "End"
```

Parameter	Description
ON	Enable function
OFF	Disable function (Default)
"Start"	The time will be increased 1 hour from "Start time"
"End"	The time will be reduced 1 hour (return) from "End time"
	Month "JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT", "NOV", "DEC" "JANUARY", "FEBRUARY", "MARCH", "APRIL", "MAY", "JUNE", "JULY", "AUGUST", "SEPTEMBER", "OCTOBER", "NOVEMBER", "DECEMBER"
	Week "SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT" "SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY"
	Which Week "FIRST", "SECOND", "THIRD", "FOURTH", "LAST" "1ST", "2ND", "3RD", "4TH", "LAST"
	Date 1~31
	Time 0:00~23:00

#### Note:

This command has been supported since V8.03 EZ and later firmware.

### Example

#### Sample code

```
SET DAYLIGHT_SAVE ON
SET DAYLIGHT_SAVE OFF
SET DAYLIGHT_SAVE "MAR 1 4:00", "NOV 1 5:00"
SET DAYLIGHT_SAVE "MAR FIRST SUN 2:00", "NOV LAST SUN 3:00"
```



## PEEL

### Description

This command obtains the status of the peel-off sensor. This attribute is read only.

### Syntax

**PEEL**

Parameter	Description
0	Paper is not on top of peel sensor
1	Paper is on top of peel sensor

### Example

#### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
SET PEEL OFF
SET KEY1 OFF
SET LED1 OFF
SET LED3 OFF
:START
LED1=0
LED3=0
IF KEY1=1 THEN GOTO A
GOTO START
:A
LED1=1
CLS
TEXT 10,10, "3",0,1,1, "PEEL Function Test!! "
PRINT 1,1
:B
LED1=0
IF PEEL=1 THEN
LED3=1
GOTO B
ELSE
CLS
TEXT 10,10, "3",0,1,1, "The label is removed from the PEEL sensor!! "
PRINT 1,1
GOTO START
ENDIF
EOP
DEMO
```



## LED1, LED2, LED3

### Description

This command is used to control LED on/off. This attribute is write-only. Specify 1 to light on LED and 0 to turn off LED. Before using this command, be sure to cancel the default LED functions. Please refer to the SET LED command.

### Syntax

**LEDm = n**

Parameter	Description
m	m=1, LED1 m=2, LED2 m=3, LED3
n	0: turn off LED 1: light on LED

Model	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED2 & LED3
PrintEasy, PrintHybrid	GREEN	GREEN	RED					ORANGE

### Example

#### Sample code

```

DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.12,0
SPEED 4
DENSITY 8
DIRECTION 1
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET LED1 OFF
SET LED2 OFF
SET LED3 OFF
LED1=0
LED2=1
LED3=0
EOP
DEMO

```



## KEY1, KEY2, KEY3

### Description

This command reads the status of KEY1, KEY2 and KEY3.

Model	KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6
PrintEasy		FEED					
PrintHybrid		FEED	MENU	UP	RIGHT	LEFT	DOWN

### Syntax

KEYm = n

Key	Return Value
KEY1 (MENU)	0: released 1: pressed
KEY2 (PAUSE)	0: released 1: pressed
KEY3 (FEED)	0: released 1: pressed

### Example

#### Sample code

```

DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
SPEED 4
DENSITY 8
DIRECTION 1
REFERENCE 0,0
SET LED1 OFF
SET KEY1 OFF
LED1=0
:START
IF KEY1=1 THEN
LED1=1
CLS
TEXT 100,10, "3",0,1,1, "KEY FUNCTION TEST"
PRINT 1,1
ELSE
LED1=0
ENDIF
GOTO START
EOP
DEMO

```



## 9. PRINTER GLOBAL VARIABLES

### @LABEL

#### Description

This variable counts how many pieces of labels have been printed. This attribute cannot be initialized if the printer is reset, but will be retained if the printer power is turned off.

#### Syntax

**Write attribute:** @LABEL=n or @LABEL= "n"  
**Read attribute:** A=LABEL or A\$=STR\$(LABEL)

Parameter	Description
n	Number of labels printed. 0<=n<=9999999999

#### Example

Sample code	Result
<pre> DOWNLOAD "DEMO.BAS" SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS TEXT 10,50, "3",0,1,1,@LABEL TEXT 10,100, "3",0,1,1, "@LABEL="+STR\$(LABEL) TEXT 10,150, "3",0,1,1, "*****Statement 1*****" IF LABEL&gt;1000 THEN TEXT 10,200, "3",0,1,1, "LABEL&gt;1000" ELSE TEXT 10,200, "3",0,1,1, "LABEL&lt;1000" ENDIF TEXT 10,250, "3",0,1,1, "*****Statement 1*****" A=LABEL IF A&gt;1000 THEN TEXT 10,300, "3",0,1,1, "A&gt;1000" ELSE TEXT 10,300, "3",0,1,1, "A&lt;1000" ENDIF TEXT 10,350, "3",0,1,1, "*****Statement 3*****" A\$=STR\$(LABEL) IF VAL(A\$)&gt;1000 THEN TEXT 10,400, "3",0,1,1, "VAL(A\$)&gt;1000" ELSE TEXT 10,400, "3",0,1,1, "VAL(A\$)&lt;1000" ENDIF PRINT 1,1 EOP DEMO </pre>	<pre> 1661 @LABEL=1661 *****Statement 1***** LABEL&gt;1000 *****Statement 1***** A&gt;1000 *****Statement 3***** VAL(A\$)&gt;1000 </pre>



## YEAR

### Description

This variable reads/writes the year data via the Real Time Clock (RTC). Four-digit year formats are supported by RTC.

### Syntax

**Write attribute:** YEAR = 02

**Read attribute:** A = YEAR

Range: 00~50 = 2000~2050; 51~99 = 1951~1999

### Example

#### Sample code

```
DOWNLOAD "SetYear.BAS"
REM *****Set Year Parameter to RTC*****
YEAR=13
EOP
SetYear
```

#### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read YEAR parameter from RTC*****
YEAR$=STR$(YEAR)
Y=YEAR

REM *****Print*****
TEXT 10,10, "3",0,1,1, "YEAR1="+YEAR$
TEXT 10,50, "3",0,1,1, "YEAR2="+STR$(Y)
TEXT 10,90, "3",0,1,1, "YEAR3="+STR$(YEAR)
PRINT 1
EOP
DEMO
```

#### Result

```
YEAR1=2013
YEAR2=2013
YEAR3=2013
```

### See Also

~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND



## MONTH

### Description

This variable reads/writes the month data via the Real Time Clock (RTC). Two-digit (01~12) month formats are supported by RTC.

### Syntax

**Write attribute:** MONTH = 01

**Read attribute:** A = MONTH

Range: 01~12

### Example

#### Sample code

```
DOWNLOAD "SetMonth.BAS"
REM *****Set Month Parameter to RTC*****
MONTH=01
EOP
SetMonth
```

#### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Month parameter form RTC*****
MONTH$=STR$(MONTH)
M=MONTH

REM *****Print*****
TEXT 10,10, "3",0,1,1, "MONTH1="+MONTH$
TEXT 10,50, "3",0,1,1, "MONTH2="+STR$(M)
TEXT 10,90, "3",0,1,1, "MONTH3="+STR$(MONTH)
PRINT 1
EOP
DEMO
```

#### Result

```
MONTH1=1
MONTH2=1
MONTH3=1
```

### See Also

`~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND`



## DATE

### Description

This variable reads/writes the date data via the Real Time Clock (RTC). Two-digit (01~31) date formats are supported by RTC.

### Syntax

**Write attribute:** DATE = 12

**Read attribute:** A = DATE

Range: 01~31

### Example

#### Sample code

```
DOWNLOAD "SetDate.BAS"
REM *****Set Date Parameter to RTC*****
DATE=10
EOP
SetDate
```

#### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Date parameter form RTC*****
DATE$=STR$(DATE)
D=DATE

REM *****Print*****
TEXT 10,10, "3",0,1,1, "DATE1="+DATE$
TEXT 10,50, "3",0,1,1, "DATE2="+STR$(D)
TEXT 10,90, "3",0,1,1, "DATE3="+STR$(DATE)
PRINT 1
EOP
DEMO
```

#### Result

```
DATE1=10
DATE2=10
DATE3=10
```

### See Also

**~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND**



## WEEK

### Description

This variable reads/writes the day of the week data via the Real Time Clock (RTC), which is represented by one single digit (1~7).

### Syntax

**Read attribute:** A = WEEK

Range: 1(Sunday)~7(Saturday)

### Example

#### Sample code

```
DOWNLOAD "DEMO.BAS "
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Week parameter form RTC*****
WEEK$=STR$(WEEK)
W=WEEK

REM *****Print*****
TEXT 10,10, "3",0,1,1, "WEEK1="+WEEK$
TEXT 10,50, "3",0,1,1, "WEEK2="+STR$(W)
TEXT 10,90, "3",0,1,1, "WEEK3="+STR$(WEEK)
PRINT 1
EOP
DEMO
```

#### Result

```
WEEK1=5
WEEK2=5
WEEK3=5
```

### See Also

~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND



## HOUR

### Description

This variable reads/writes the hour data via the Real Time Clock (RTC). The 24-hour-day system (00~23) is supported by RTC.

### Syntax

**Write attribute:** HOUR = 12

**Read attribute:** A = HOUR

Range: 00~23

### Example

#### Sample code

```
DOWNLOAD "SetHour.BAS"
REM *****Set Hour Parameter to RTC*****
HOUR=10
EOP
SetHour
```

#### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Hour parameter form RTC*****
HOUR$=STR$(HOUR)
H=HOUR

REM *****Print*****
TEXT 10,10, "3",0,1,1, "HOUR1="+HOUR$
TEXT 10,50, "3",0,1,1, "HOUR2="+STR$(H)
TEXT 10,90, "3",0,1,1, "HOUR3="+STR$(HOUR)
PRINT 1
EOP
DEMO
```

#### Result

```
HOUR1=10
HOUR2=10
HOUR3=10
```

### See Also

`~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND`



## MINUTE

### Description

This variable reads/writes the minute data via the Real Time Clock (RTC). Two-digits (00~59) minute format is supported by RTC.

### Syntax

**Write attribute:** MINUTE = 12

**Read attribute:** A = MINUTE

Range: 00~59

### Example

#### Sample code

```
DOWNLOAD "SetMinute.BAS"
REM *****Set Minute Parameter to RTC*****
MINUTE=27
EOP
SetMinute
```

#### Sample code

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Minute parameter form RTC*****
MINUTE$=STR$(MINUTE)
MIN=MINUTE

REM *****Print*****
TEXT 10,10, "3",0,1,1, "MINUTE1="+MINUTE$
TEXT 10,50, "3",0,1,1, "MINUTE2="+STR$(MIN)
TEXT 10,90, "3",0,1,1, "MINUTE3="+STR$(MINUTE)
PRINT 1
EOP
DEMO
```

#### Result

```
MINUTE1=27
MINUTE2=27
MINUTE3=27
```

### See Also

~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND



## SECOND

### Description

This variable reads/writes the second data via the Real Time Clock (RTC). Two-digits (00~59) second format is supported by RTC.

### Syntax

**Write attribute:** SECOND = 12

**Read attribute:** A = SECOND

Range: 00~59

### Example

#### Sample code

```
DOWNLOAD "SetSecond.BAS"
REM *****Set Second Parameter to RTC*****
SECOND=59
EOP
SetSecond
```

#### Sample code

```
DOWNLOAD "DEMO.BAS "
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Second parameter form RTC*****
SECOND$=STR$(SECOND)
SEC=SECOND

REM *****Print*****
TEXT 10,10, "3",0,1,1, "SECOND1="+SECOND$
TEXT 10,50, "3",0,1,1, "SECOND2="+STR$(SEC)
TEXT 10,90, "3",0,1,1, "SECOND3="+STR$(SECOND)
PRINT 1
EOP
DEMO
```

#### Result

```
SECOND1=59
SECOND2=59
SECOND3=59
```

### See Also

~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND



## @YEAR

### Description

This variable reads/writes the year data via the Real Time Clock (RTC). Two-digit year formats are supported by RTC. @YEAR global variable can be accessed directly without using BASIC language functions.

### Syntax

**Write attribute:** @YEAR = "01"

**Read attribute:** @YEAR

Range: 00~99

### Example

Sample code	Result	
<pre>REM *****Set @YEAR***** @YEAR="05"  REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@YEAR" TEXT 210,10, "3",0,1,1, @YEAR PRINT 1</pre>	@YEAR	2005

### See Also

~!C, @MONTH, @DATE, @DAY, @HOUR, @MINUTE, @SECOND



## @MONTH

### Description

This variable reads/writes the month data via the Real Time Clock (RTC). Two-digits (01~12) month formats are supported by RTC. @MONTH global variable can be accessed directly without using BASIC language functions.

### Syntax

**Write attribute:** @MONTH = "01"

**Read attribute:** @MONTH

Range: 01~12

### Example

Sample code	Result
<pre>REM *****Set @MONTH***** @MONTH="12"  REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@MONTH" TEXT 210,10, "3",0,1,1,@MONTH PRINT 1</pre>	<p style="text-align: center;">@MONTH 12</p>

### See Also

~!C, @YEAR, @DATE, @DAY, @HOUR, @MINUTE, @SECOND



## @DATE

### Description

This variable reads/writes the date data via the Real Time Clock (RTC). Two-digits (01~31) date formats are supported by RTC. @DATE global variable can be accessed directly without using BASIC language functions.

### Syntax

**Write attribute:** @DATE = "12"

**Read attribute:** @DATE

Range: 01~31

### Example

Sample code	Result
<pre>REM *****Set @DATE***** @DATE="31"  REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@DATE" TEXT 210,10, "3",0,1,1,@DATE PRINT 1</pre>	<p style="text-align: center;">@DATE            31</p>

### See Also

~!C, @YEAR, @MONTH, @DAY, @HOUR, @MINUTE, @SECOND



## @DAY

### Description

This variable reads/writes the day of the week data via the Real Time Clock (RTC), which is represented by one single digit (1~7). @DAY global variable can be accessed directly without using BASIC language functions.

### Syntax

**Read attribute:** @DAY

Range: 1(Sunday)~7(Saturday)

### Example

Sample code	Result
<pre>REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@DAY" TEXT 210,10, "3",0,1,1,@DAY PRINT 1</pre>	<p>@DAY 7</p>

### See Also

~!C, @YEAR, @MONTH, @DATE, @HOUR, @MINUTE, @SECOND



## @HOUR

### Description

This variable reads/writes the hour data via the Real Time Clock (RTC). The 24-hour-day system (00~23) is supported by RTC. @HOUR global variable can be accessed directly without using BASIC language functions.

### Syntax

**Write attribute:** @HOUR = "12"

**Read attribute:** @HOUR

Range: 00~23

### Example

Sample code	Result
<pre>REM *****Set @HOUR***** @HOUR="23"  REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@HOUR" TEXT 210,10, "3",0,1,1,@HOUR PRINT 1</pre>	<p style="text-align: center;">@HOUR      23</p>

### See Also

~!C, @YEAR, @MONTH, @DATE, @DAY, @MINUTE, @SECOND



## @MINUTE

### Description

This variable reads/writes the minute data via the Real Time Clock (RTC). The two-digits (00~59) minute format is supported by RTC. @MINUTE global variable can be accessed directly without using BASIC language functions.

### Syntax

**Write attribute:** @MINUTE = "12"

**Read attribute:** @MINUTE

Range: 00~59

### Example

Sample code	Result
<pre>REM *****Set @MINUTE***** @MINUTE="59"  REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@MINUTE" TEXT 210,10, "3",0,1,1,@MINUTE PRINT 1</pre>	<p style="text-align: center;">@MINUTE      59</p>

### See Also

~!C, @YEAR, @MONTH, @DATE, @DAY, @HOUR, @SECOND



## @SECOND

### Description

This variable reads/writes the second data via the Real Time Clock (RTC). The Two-digit (00~59) second format is supported by RTC. @SECOND global variable can be accessed directly without using BASIC language functions.

### Syntax

**Write attribute:** @SECOND = "12"

**Read attribute:** @SECOND

Range: 00~59

### Example

Sample code	Result
<pre>REM *****Set @SECOND***** @SECOND = "59"  REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@SECOND" TEXT 210,10, "3",0,1,1,@SECOND PRINT 1</pre>	<p style="text-align: center;">@SECOND      59</p>

### See Also

~!C, @YEAR, @MONTH, @DATE, @DAY, @HOUR, @MINUTE



## \_MODEL\$

### Description

This variable can be read only. It includes the information of printer's model name.

### Syntax

\_MODEL\$

### Example

#### Sample code

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

#### Result

```
Model: TDP247
Serial No.: D452350388
F/W Version: 7.00
```

### See Also

\_SERIAL\$, \_VERSION\$



## \_SERIAL\$

### Description

This variable can be read only. It includes the information of printer's serial number.

\*The printer's serial number must be programmed into printer at factory.

### Syntax

\_SERIAL\$

### Example

#### Sample code

```
SIZE 4,1  
GAP 0,0  
DIRECTION 1  
CLS  
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$  
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$  
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$  
PRINT 1
```

#### Result

```
Model: TDP247  
Serial No.: D452350388  
F/W Version: 7.00
```

### See Also

\_MODEL\$, \_VERSION\$



## \_VERSION\$

### Description

This variable can be read only. It includes the information of printer's firmware version.

### Syntax

\_VERSION\$

### Example

#### Sample code

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

#### Result

```
Model: TDP247
Serial No.: D452350388
F/W Version: 7.00
```

### See Also

\_MODEL\$, \_SERIAL\$



## 10. WI-FI MODULE SETTING COMMANDS

### WLAN OFF

#### Description

This command is used to disable Wi-Fi module. Restart the printer is necessary.

#### Syntax

**WLAN OFF**

#### Note:

We can use command SELFTEST WLAN to check the status of WLAN. The SSID is empty.

#### Example

Sample code	Result
<b>WLAN OFF</b> <b>SELFTEST WLAN</b>	<pre>-----           WLAN SETTING -----       MAC ADDR: 001DC9-908397       MODE: INFRASTRUCTURE       SSID:       DHCP: OFF       IP ADDR: 0.0.0.0        SUBNET: 0.0.0.0       GATEWAY: 0.0.0.0       PORT: 9100 -----</pre>

#### See Also

[WLAN SSID](#)



## WLAN SSID

### Description

This command is used to set the SSID of your wireless network into Wi-Fi module. Restart the printer is necessary.

### Syntax

**WLAN SSID "ssid"**

Parameter	Description
ssid	It is the SSID of your wireless network.

**Note:**

**SSID is case-sensitive. The maximum length is 32 bytes.**

### Example

Sample code	Result
<b>WLAN SSID "TEST-AP"</b> <b>SELFTEST WLAN</b>	----- WLAN SETTING ----- MAC ADDR: 001DC9-908397 SSID: TEST-AP DHCP: OFF IP ADDR: 0.0.0.0  SUBNET: 0.0.0.0 GATEWAY: 0.0.0.0 PORT: 9100 -----

### See Also

**WLAN OFF**



## WLAN WPA

### Description

This command is used to set WPA security mode. This command only can be set but not be checked. Restart the printer is necessary.

### Syntax

**WLAN WPA OFF**

**WLAN WPA “key”**

Parameter	Description
OFF	Disable WPA security mode.
Key	The network security key. 8 to 63 characters. Key = Passphrase or Pre-Shared Key (Passphrase is a string containing between 8 and 63 characters) (Pre-Shared Key is a 32-byte key, formatted as hexadecimal number)

### Example

#### Sample code

**WLAN WPA OFF**

**WLAN WPA “123456789”**



## WLAN WEP

### Description

This command is used to set WEP security mode. This command only can be set but not be checked. Restart the printer is necessary.

### Syntax

**WLAN WEP OFF**

**WLAN WEP n, "key"**

Parameter	Description
OFF	Disable WEP security mode.
n	The index of key. 1 to 4.
key	The encryption key. 5 or 13 characters or 10 or 26 hexadecimal digits.

### Example

#### Sample code

```
WLAN WEP OFF
WLAN WEP 1, "ABCDE"
WLAN WEP 2, "ABCDE"
WLAN WEP 3, "ABCDE"
WLAN WEP 4, "4142434445"
```



## WLAN DHCP

### Description

This command is used to set the printer to get the IP address from DHCP server. Restart the printer is necessary.

### Syntax

**WLAN DHCP**

### Example

Sample code	Result
<pre>WLAN SSID "TEST-AP" WLAN WPA "123456789" WLAN DHCP WLAN PORT 9100 SELFTEST WLAN</pre>	<pre>----- WLAN SETTING ----- MAC ADDR: 001DC9-908397 SSID: TEST-AP DHCP: ON IP ADDR: 10.0.10.138 SUBNET: 255.255.255.0 GATEWAY: 10.0.10.252 PORT: 9100 -----</pre>

### See Also

[WLAN IP](#)



## WLAN IP

### Description

This command is used to set the specific static IP address to printer. Restart the printer is necessary.

### Syntax

**WLAN IP “ip”, “mask”, “gateway”**

Parameter	Description
ip	IP address.
mask	Subnet mask.
gateway	Default gateway.

### Example

Sample code	Result
<b>WLAN SSID “TEST-AP”</b> <b>WLAN WPA “123456789”</b> <b>WLAN IP “10.0.10.138”, “255.255.255.0”,</b> <b>“10.0.10.252”</b> <b>WLAN PORT 9100</b> <b>SELFTEST WLAN</b>	----- ----- <b>WLAN SETTING</b> ----- MAC ADDR: 001DC9-908397 SSID: TEST-AP DHCP: OFF IP ADDR: 10.0.10.138 SUBNET: 255.255.255.0 GATEWAY: 10.0.10.252 PORT: 9100 -----

### See Also

[WLAN DHCP](#)



## WLAN PORT

### Description

This command is used to specify the PORT number of Wi-Fi module. Restart the printer is necessary.

### Syntax

**WLAN PORT number**

Parameter	Description
number	Base raw port number. Default is 9100.

### Example

Sample code	Result
<pre>WLAN SSID "TEST-AP" WLAN WPA "123456789" WLAN IP "10.0.10.138", "255.255.255.0","10.0.10.252" WLAN PORT 8000 SELFTEST WLAN</pre>	<pre>----- ----- WLAN SETTING ----- MAC ADDR: 001DC9-908397 SSID: TEST-AP DHCP: OFF IP ADDR: 10.0.10.138 SUBNET: 255.255.255.0 GATEWAY: 10.0.10.252 PORT: 8000 -----</pre>



## 11. INTERNAL ETHERNET SETTING COMMANDS

### NET DHCP

#### Description

This command is used to set the printer to get the IP address from DHCP server. Printer will restart itself while setting this command.

#### Syntax

**NET DHCP**

#### Example

Sample code	Result
<b>NET DHCP</b> <b>SELFTEST ETHERNET</b>	<b>ETHERNET SETTING</b> ----- NAME : PS-600002 MAC ADDR : 001B82-600002 DHCP : ON IP ADDR : 192.168.0.107 SUBNET : 255.255.255.0 GATEWAY : 192.168.0.1 PORT : 9100 -----

#### See Also

**NET IP**



## NET IP

### Description

This command is used to set the specific IP address to printer. Printer will restart itself while setting this command.

### Syntax

**NET IP "ip","mask","gateway"**

Parameter	Description
ip	IP address.
mask	Subnet mask.
gateway	Default gateway.

### Example

Sample code	Result
<pre>NET IP "192.168.10.40","255.255.255.0","192. 168.10.252" SELFTEST ETHERNET</pre>	<pre>----- ETHERNET SETTING ----- NAME: PS-600002 MAC ADDR: 001B82-600002 DHCP: OFF IP ADDR: 192.168.10.40 SUBNET: 255.255.255.0 GATEWAY: 192.168.10.252 PORT: 9100 -----</pre>

### See Also

[NET DHCP](#)



## NET PORT

### Description

This command is used to specify the PORT number of Ethernet. Printer will restart itself while setting this command.

### Syntax

**NET PORT number**

Parameter	Description
number	Base raw port number. Default is 9100.

### Example

Sample code	Result
<b>NET PORT 9100</b> <b>SELFTEST ETHERNET</b>	<pre>----- ETHERNET SETTING ----- NAME: PS-600002 MAC ADDR: 001B82-600002 DHCP: OFF IP ADDR: 192.168.10.40 SUBNET: 255.255.255.0 GATEWAY: 192.168.10.252 PORT: 9100 -----</pre>



## NET NAME

### Description

This command is used to set the printer server name.

### Syntax

**NET NAME “printerserver”**

Parameter	Description
printerserver	The specific name of printer server.

### Example

Sample code	Result
<b>NET NAME “TEST”</b> <b>SELFTEST ETHERNET</b>	<b>-----</b> <b>ETHERNET SETTING</b> <b>-----</b> NAME : TEST MAC ADDR : 001B82-600002 DHCP : OFF IP ADDR : 192.168.10.40 SUBNET : 255.255.255.0 GATEWAY : 192.168.10.252 PORT : 9100 <b>-----</b>



## 12. NFC SETTING COMMANDS

### NFC FEATURE

#### Description

This command is used to return information if the printer supports the optional NFC feature, and if it is currently installed. (Supported device: Alpha-2R series)

#### Syntax

##### NFC FEATURE

Parameter	Description
None	N/A
<b>Return Information</b>	
not available	NFC is not supported
not present	The feature is unavailable. NFC is supported, but no reader is installed
present	The feature is available. NFC is supported with a reader is installed

#### Example

Sample code	Result
<b>NFC FEATURE</b>	<b>Example by CommTool:</b>  present



## NFC STATUS

### Description

This command is used to return current status of the NFC reader or status of last operation completed.

### Syntax

**NFC STATUS**

Parameter	Description
None	N/A
<b>Return Information</b>	
Idle	The reader is inactive or hasn't been used
in progress	The operation is pending
timed out	The operation has timed out
successful	The operation has been completed successfully

### Example

Sample code	Result
<b>NFC STATUS</b>	<p>Example by CommTool:</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">in progress</div>



## NFC TIMEOUT

### Description

This command is used to set the timeout for the current read/write operation (in seconds) 0 to 3600, setting to 0 disables the timeout feature.

### Syntax

**NFC TIMEOUT m**

Parameter	Description
m	0 to 3600

**Note:**

- The default value is 10 seconds when printer initializes.
- The printer will beep for notice when it's timeout.

### Example

Sample code

**NFC TIMEOUT 20**



## NFC READ

### Description

This command is used to return content stored in the last NFC read event. (Max. of 2048 characters)

### Syntax

**NFC READ**

Parameter	Description
None	N/A

### Example

**Sample code**

**NFC READ**



## NFC WRITE

### Description

This command is used to set the content to be transmitted by the NFC system. (Max. of 2048 characters)

### Syntax

**NFC WRITE “content”**

Parameter	Description
content	Content of text string

### Example

#### Sample code

**NFC WRITE “Test”**



## NFC MODE

### Description

This command is used to set the NFC reader mode. This command can start or stop a read or write operation. (Max. of 2048 characters)

### Syntax

**NFC MODE OFF/READ/WRITE**

Parameter	Description
OFF	Disable
READ	Read tag mode
WRITE	Write tag mode

**Note:**

- The default value is “OFF”. It returns to “OFF” after a read or write operation completes, fails or times out.
- For continue to write or read data to tag, set this value to the desired “READ” or “WRITE”.

### Example

Sample code	
Write data to tag once	Continue to write data to tag
<b>NFC MODE OFF</b>	
<b>NFC TIMEOUT 3</b>	
<b>NFC WRITE “Test”</b>	
Read data from tag once	Continue to read data from Tag
<b>NFC MODE OFF</b>	
<b>NFC TIMEOUT 3</b>	
<b>NFC READ</b>	



## 13. ALPHA-2R SETTING COMMANDS

### SET PRINTQUALITY

#### Description

This command is used to set the print mode (print quality) for Alpha-2R series printer.

#### Syntax

**SET PRINTQUALITY DRAFT/STANDARD/OPTIMUM**

Parameter	Description
DRAFT	High print speed with lower density
STANDARD	Standard print speed and quality
OPTIMUM	According to the label content such as barcode, text, and graphic to lower the print speed for getting higher print quality

**Note:**

The default value is “**STANDARD**”.

#### Example

**Sample code**

```
SET PRINTQUALITY DRAFT  
SET PRINTQUALITY STANDARD  
SET PRINTQUALITY OPTIMUM
```



## SET STANDBYTIME

### Description

This command is used to set the standby time for Alpha-2R series printer.

### Syntax

**SET STANDBYTIME OFF/XXXXX**

Parameter	Description
OFF	Disable
XXXXX	0 ~ 625534 (sec.)

**Note:**

The default value is “**SET STANDBYTIME 120**”.

### Example

**Sample code**

```
SET STANDBYTIME OFF  
SET STANDBYTIME 480
```



## GETSETTING\$()

### Description

This command is used to get the print mode (print quality) or standby time for Alpha-2R series printer.

### Syntax

```
GETSETTING$("SYSTEM", "INFORMATION", "PRINTQUALITY")
GETSETTING$("SYSTEM", "INFORMATION", "STANDBYTIME")
```

Parameter	Description
None	N/A

### Return Information

**PRINTQUALITY** DRAFT, STANDARD or OPTIMUM

**STANDBYTIME** OFF or number



## 14. GPIO SETTING COMMANDS

### SET GPO

#### Description

Use this command to send out the GPIO signals by the printer.

#### Syntax

**SET GPOn signal state, delay0, pulse0, delay1, pulse1, function condition**

Parameter	Description	
n	n = 1 ~ 7	Seven dedicated outputs are available for the desired function conditions.
signal state	<b>HIGH</b>	Goes the high level signal when the following function condition is detected.
	<b>LOW</b>	Goes the low level signal when the following function condition is detected.
	<b>POS</b>	Goes the positive pulse signal when the following function condition is detected.
	<b>NEG</b>	Goes the negative pulse signal when the following function condition is detected.
delay0	After detecting the following function condition, the printer will wait this period of time before sending out the “true” output signal. Unit: millisecond. Maximum: 32000.	
pulse0	Pulse width corresponding to the function condition becoming “true”. (Ignored for level-type signals.) Unit: millisecond. Maximum: 32000.	
delay1	After detecting the following function condition, the printer will wait this period of time before sending out the “false” output signal. Unit: millisecond. Maximum: 32000.	
pulse1	Pulse width corresponding to the function condition becoming “false”. (Ignored for level-type signals.) Unit: millisecond. Maximum: 32000.	



## Syntax

Parameter	Description
Function condition (warning, error, control...)	<b>FAULT</b> Printer fault.
	<b>FAULT RIBBON</b> Ribbon error is occurred.
	<b>FAULT PAPER</b> Paper empty or paper jam is occurred.
	<b>FAULT CARRIAGE</b> Carriage is open.
	<b>FAULT MEMORY</b> Out of memory is occurred.
	<b>FAULT CUTTER</b> Cutter error is occurred.
	<b>FAULT OVERHEAT</b> Stepping motor or print head is over heat.
	<b>PAUSE</b> Pause status is occurred.
	<b>TAKELABEL</b> Take label is occurred.
	<b>IDLE</b> Printer is idle.
	<b>PRINT</b> Printer is printing.

## Example

### Sample code

```
SET GP01 HIGH,100,0,100,0,FAULT RIBBON
SET GP02 LOW,100,0,100,0,FAULT PAPER
SET GP03 POS,100,100,100,100,PAUSE
SET GP04 NEG,100,50,100,50,IDLE
```



## SET GPI

### Description

Use this command to receive the GPIO signals from external controlling devices.

### Syntax

**SET GPI** *n*, *signal*, *pulse*, *function*

Parameter	Description	
<i>n</i>	<i>n</i> = 1 ~ 4	Four dedicated inputs are available for the desired control functions.
signal state	<b>HIGH</b>	When a high level signal received, will activate the following printer control functions.
	<b>LOW</b>	When a low level signal received, will activate the following printer control functions.
	<b>POS</b>	When a positive pulse signal received, will activate the following printer control functions.
	<b>NEG</b>	When a negative pulse signal received, will activate the following printer control functions.
pulse	Filter pulse width. Ignored for level-type signals. Unit: millisecond. Maximum: 32000.	
function (control)	<b>PAUSE</b>	Toggle pause status.
	<b>PAUSE ON</b>	Enter pause status.
	<b>PAUSE OFF</b>	Cancel pause status.
	<b>PRINT</b>	Print batch of labels.
	<b>PRINT n</b>	<i>n</i> is numerical. Specify how many labels to print. Maximum: 32000.
	<b>CUT</b>	Cut immediately.
	<b>FEED n</b>	<i>n</i> is numerical and the unit is dot. Specify the feeding length. Maximum: 32000.
	<b>BACKFEED n</b>	<i>n</i> is numerical and the unit is dot. Specify the backfeeding length. Maximum: 32000.
	<b>FORMFEED</b>	Feeding an empty label.
	<b>INPUT n</b>	<i>n</i> is text or command. The <i>n</i> will be triggered to printer.



## Example

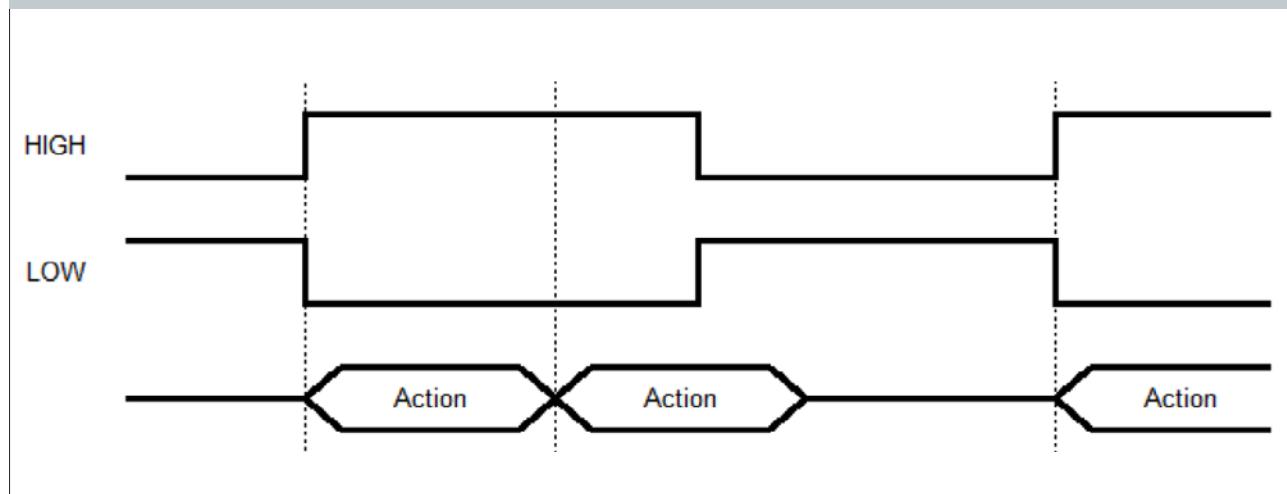
### Sample code

```
SET GPI1 HIGH,0,PAUSE  
SET GPI2 LOW,0,PAUSE ON  
SET GPI3 POS,100,PAUSE OFF  
SET GPI4 NEG,100,CUT  
SET GPI5 NEG,100,INPUT "TEST.BAS"+CHR$(13)+CHR$(10)
```

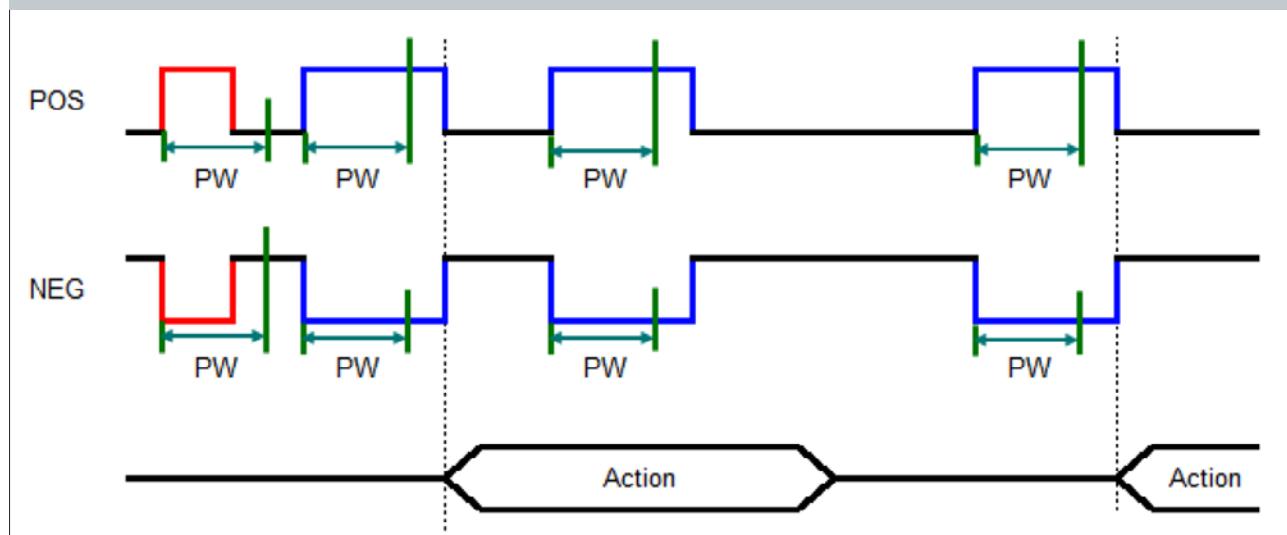


## GPIO Waveform

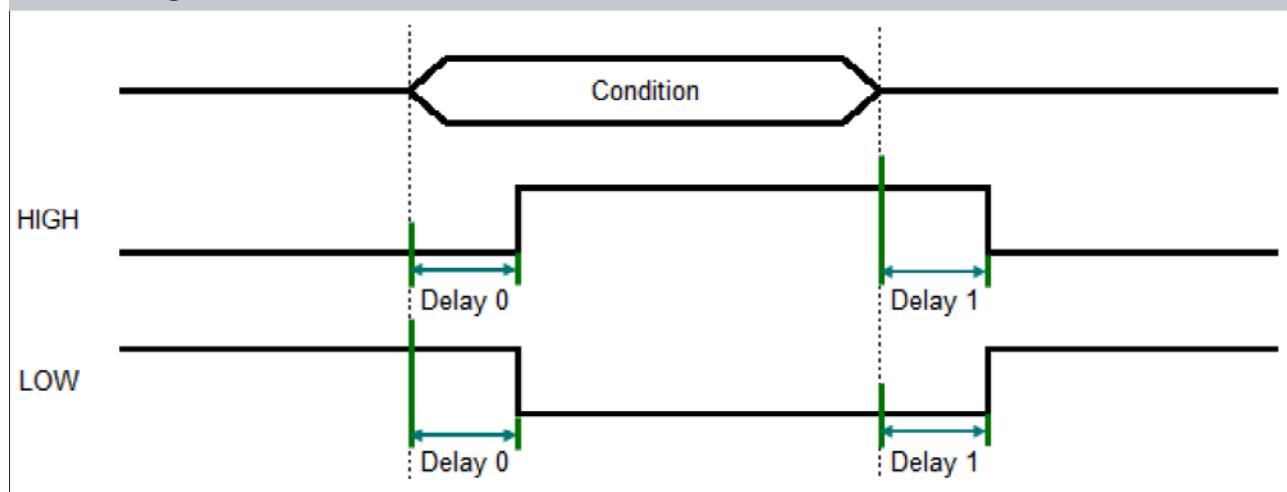
GPI Level Signal : Continuous action. (Host to printer.)



GPI Pulse Signal : A pulse is an action. (Host to printer.)

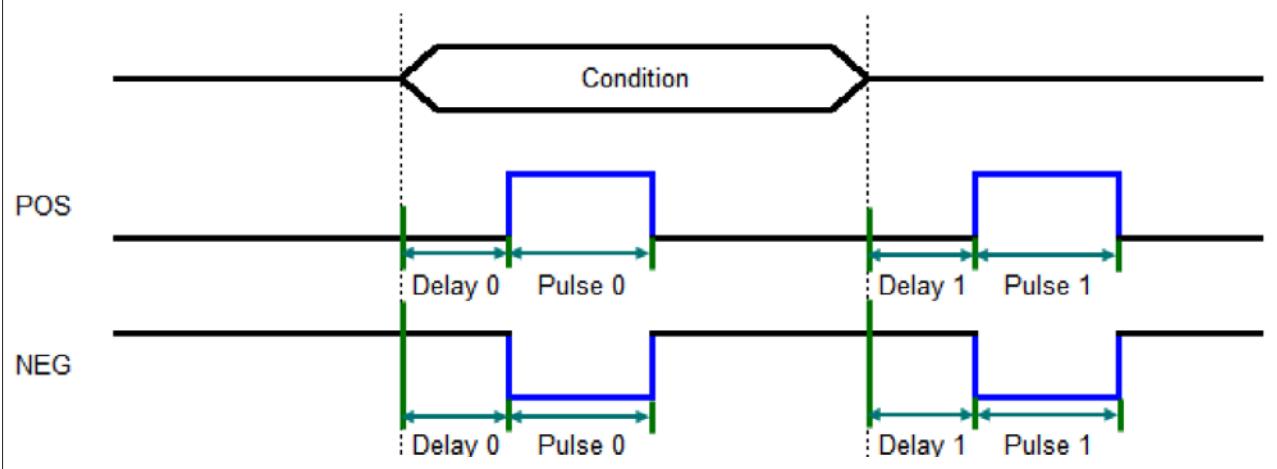


GPO Level Signal : Continuous condition. (Printer to host.)



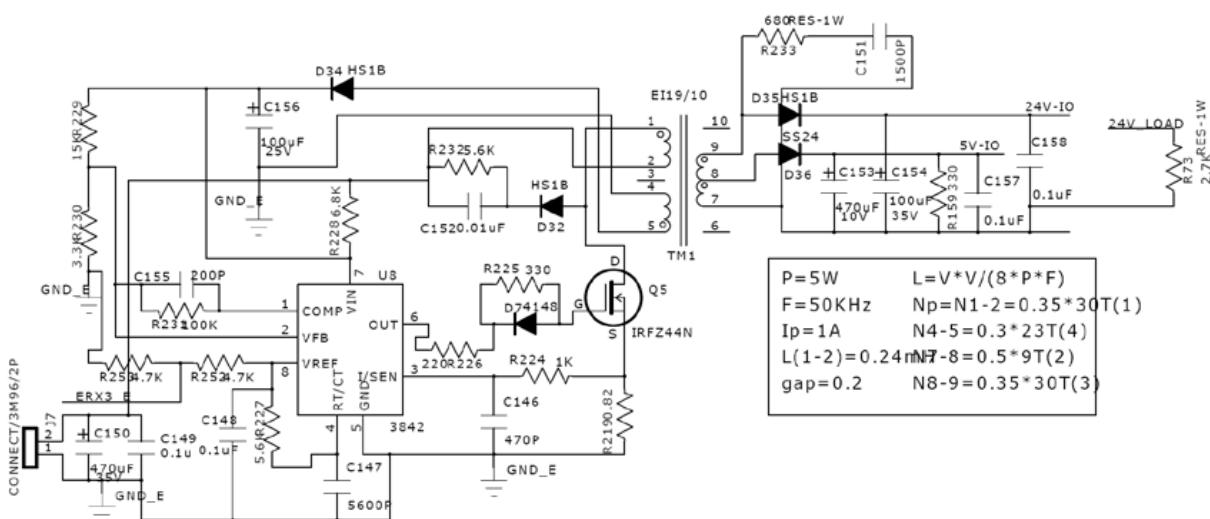
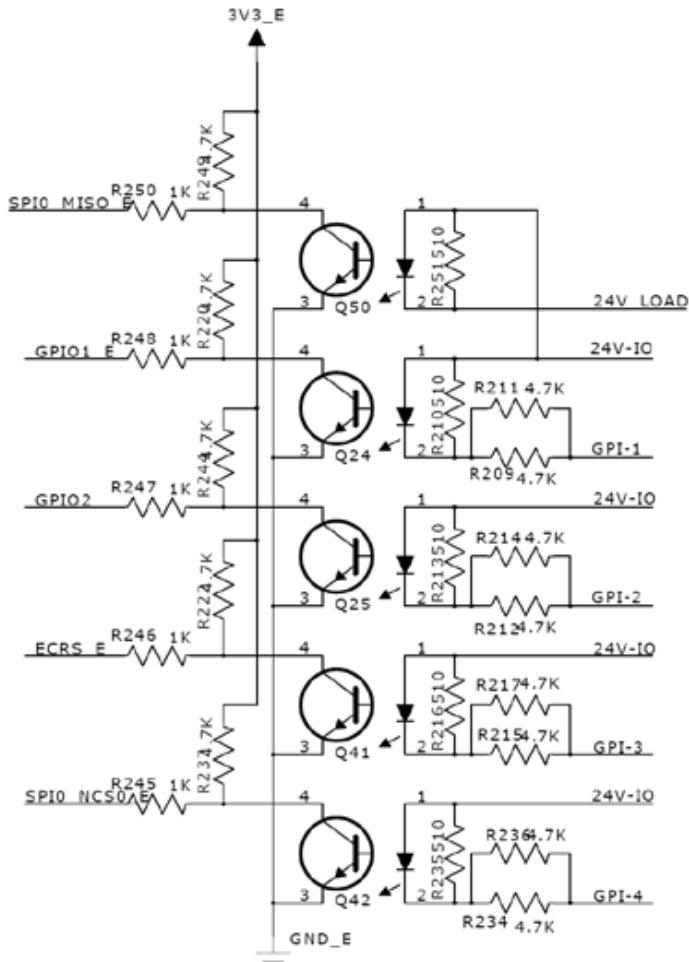


GPO Pulse Signal : A pulse is a condition. (Printer to host.)



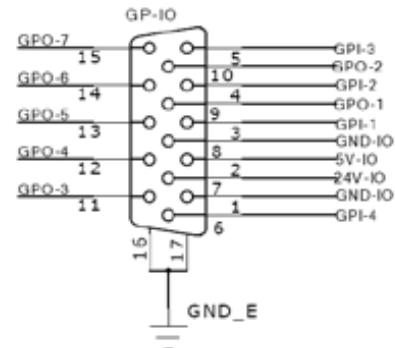
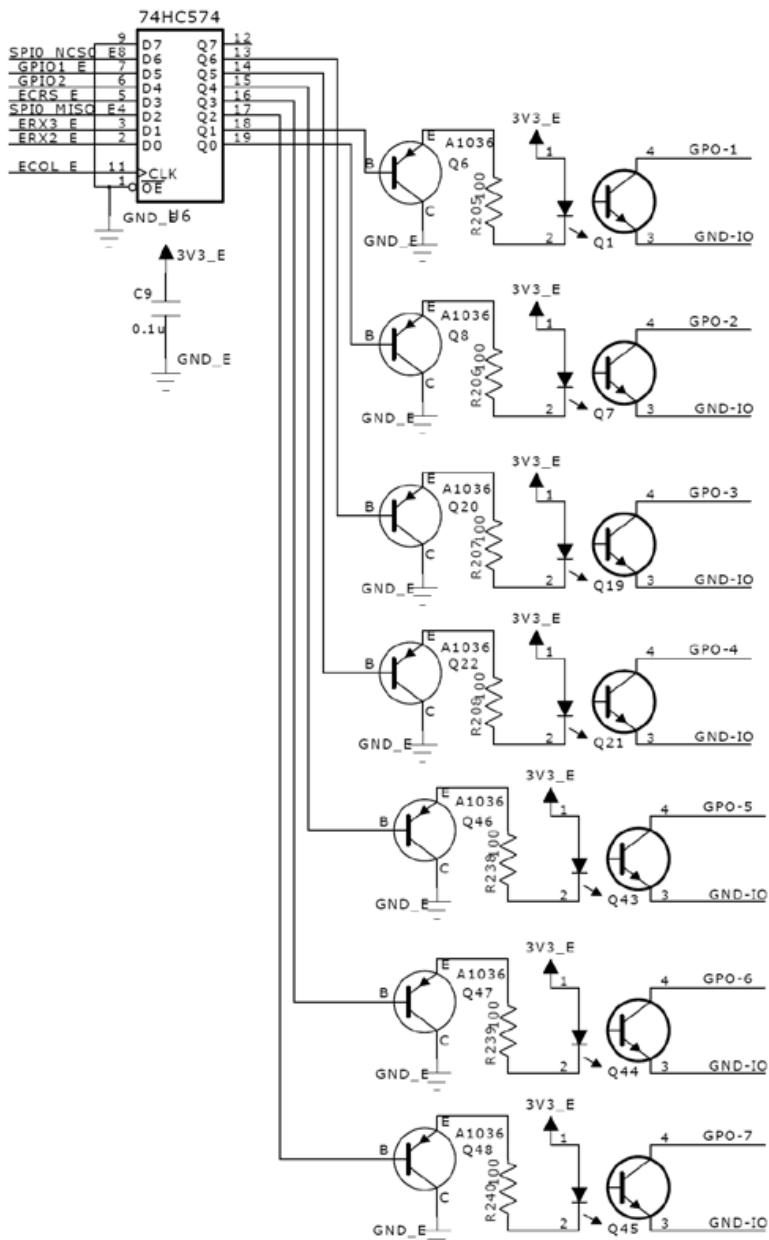


## GPIO Circuit Diagram

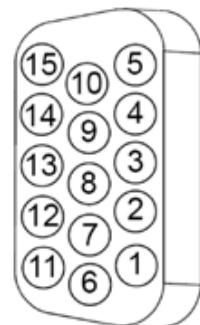




## GPIO Circuit Diagram

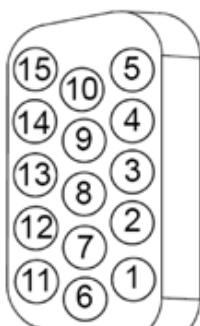


TTP-2410MT/ MXP/ MH series



PIN No.	Signal Name
1	GND-IO
2	5V-IO
3	GPI-1
4	GPI-2
5	GPI-3
6	GPI-4
7	24V-IO
8	GND-IO
9	GPO-1
10	GPO-2
11	GPO-3
12	GPO-4
13	GPO-5
14	GPO-6
15	GPO-7

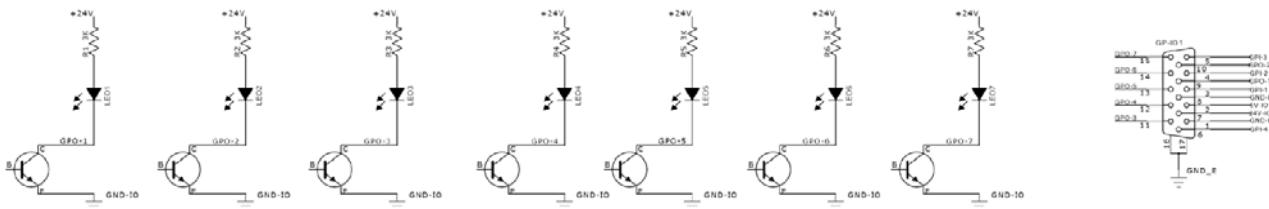
TTP-384M series



PIN No.	Signal Name
1	GND-IO
2	5V-IO
3	GPI-1
4	GPI-2
5	N/A
6	N/A
7	24V-IO
8	GND-IO
9	N/A
10	N/A
11	N/A
12	GPO-4
13	GPO-5
14	N/A
15	GPO-7



## GPIO Circuit Diagram

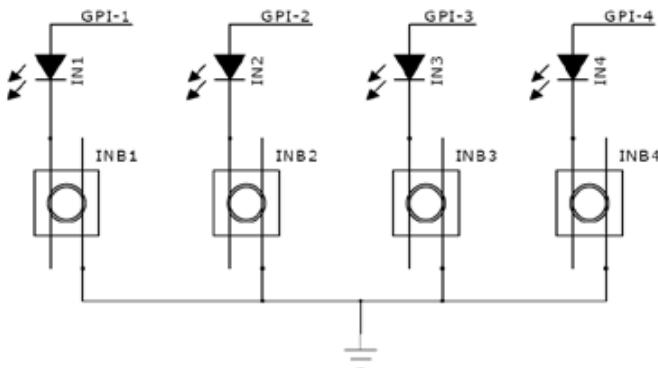


### GPO pin no. 1~7 application example:

Since we connect GPO pin no. 1~7 with seven individual LED, the output signal from GPO will light the individual LED on or off.

\*NPN output specification.

Collector-emitter voltage	$V_{CEO}$	35 V
Emitter-collector voltage	$V_{CEO}$	6 V
Collector current	$I_C$	Max. 50 mA
*1 Collector power dissipation	$P_C$	150 mW



### GPI pin no. 1~4 application example:

Since we connect GPI pin no. 1~4 with four individual button keys to control the desired printer functions. The input signal current suggests 20 mA.



## COPYRIGHT INFORMATION

©2019 Meto International GmbH

Information in this document is subject to change without notice and do not represent a commitment on the part of Meto International GmbH. No part of this manual may be reproduced or transmitted in any form or by any means, for any purpose other than the purchaser's personal use, without the expressed written permission by Meto International GmbH.

The firmware, software and TSPL-EZ™ printer language described in this manual are developments of TSC Auto ID Technology Co., Ltd. and are copyrighted by TSC Auto ID Technology Co., Ltd.



## METO PRINT SERVICE:

### Contact and technical support:

You can find all valid contact data concerning service, hotline and technical support under [www.meto.com/metoprint](http://www.meto.com/metoprint)

Or you may send an e-mail to  
[metoprint@meto.com](mailto:metoprint@meto.com)



**Meto International GmbH**  
69434 Hirschhorn, Germany  
[www.meto.com](http://www.meto.com)

**METO**<sup>®</sup>  
your retail label expert