



*Barcode Printer*

# TSC BARCODE PRINTER Series

*Thermal Transfer • Direct Thermal*



Programming Manual

# Trademark and Copyright Notice

©2025 TSC Auto ID Technology Co., Ltd.

All trademarks mentioned in this document are the property of their respective owners. TSC is a trademark of TSC Auto ID Technology Co., Ltd., registered in many jurisdictions worldwide. Unauthorized reproduction or use of these trademarks, or any [part](#) of this document, is strictly prohibited.

## Product Improvements and Updates

TSC Auto ID continuously strives to improve our products. All specifications, features, and designs are subject to change without notice as part of our ongoing product enhancement initiatives. It is recommended to regularly consult the latest documentation to ensure the most up-to-date information is being used. Product users should validate that any new specifications or feature updates for compatibility with their existing applications before implementation.

## Proprietary Information and Confidentiality

This manual contains proprietary information of TSC Auto ID Technology Co., Ltd. (TSC), which is intended solely for the use of parties operating and maintaining the equipment described herein. Such proprietary information must not be used, reproduced, or disclosed to any third party for any purpose without the express written permission of TSC.

## Disclaimer

While TSC Auto ID makes every effort to ensure the accuracy of the information contained in our specifications and manuals, errors may still occur. TSC Auto ID reserves the right to correct any errors, and disclaims any liability caused by such errors. The information provided in this document is for reference only and does not constitute a guarantee of performance or suitability for any particular application.

## Limitation of Liability

TSC Auto ID is not responsible for any direct, indirect, incidental, or consequential damages arising from the use, inability to use, or performance of our products. This includes, but is not limited to, business losses, interruptions, or the loss of business data, even if TSC was advised of the possibility of such damages.

Some jurisdictions may not permit the exclusion of incidental or consequential damages, so the limitations and exclusions outlined here may not apply to you.

## User Responsibility

It is the responsibility of the user to comply with all relevant laws and licensing agreements when using this document and the associated products. TSC Auto ID disclaims responsibility for any consequences arising from improper usage or unauthorized modifications to the products.

## Security and System Integrity

TSC Auto ID is not responsible for security vulnerabilities introduced through third-party software, unauthorized file uploads, or improper system configurations via any access path. Users are responsible for implementing appropriate security measures to prevent potential risks. TSC Auto ID is not liable for any malfunctions, disruptions, or security issues resulting from such actions.



# TABLE OF CONTENTS

<b>How to Read</b> .....	<b>IX</b>
<b>Document Conventions</b> .....	<b>X</b>
<b>Object Position Calculation</b> .....	<b>XI</b>
<b>Printer Model List</b> .....	<b>XIII</b>
<b>Setup and System Commands</b> .....	<b>1</b>
SIZE.....	1
GAP.....	3
GAPDETECT .....	6
BLINEDETECT .....	7
AUTODETECT.....	8
BLINE .....	9
OFFSET.....	11
SPEED .....	13
DENSITY .....	16
DIRECTION AND MIRROR IMAGE .....	17
REFERENCE.....	18
SHIFT .....	20
COUNTRY.....	22
CODEPAGE.....	24
CLS.....	26
FEED .....	27
BACKFEED & BACKUP .....	28
FORMFEED .....	29
HOME .....	30
PRINT.....	31
SOUND .....	33
CUT.....	34
LIMITFEED .....	35
SELFTTEST.....	36
EOJ.....	39
DELAY .....	41
DISPLAY .....	43

INITIALPRINTER .....	45
MENU .....	46
VERTICAL .....	49
EXPORT .....	50
<b>Label Formatting Commands .....</b>	<b>52</b>
BAR .....	52
BARCODE .....	54
TLC39 .....	65
BITMAP .....	67
BOX .....	70
CIRCLE .....	72
ELLIPSE .....	73
CODABLOCK F MODE .....	74
DMATRIX .....	76
ERASE .....	80
MAXICODE .....	81
PDF417 .....	83
AZTEC .....	88
MPDF417 .....	90
PUTBMP .....	92
PUTPCX .....	94
PUTPNG .....	95
QRCODE .....	96
RMQR .....	103
RSS .....	106
REVERSE .....	112
DIAGONAL .....	113
TEXT .....	114
BLOCK .....	118
<b>Status Polling and Immediate Commands .....</b>	<b>122</b>
<ESC>!? .....	122
<ESC>!C .....	123
<ESC>!D .....	124
<ESC>!O .....	125
<ESC>!P .....	126

<ESC>!Q.....	127
<ESC>!R.....	128
<ESC>!S.....	129
<ESC>!F.....	133
<ESC>!.....	134
~!@.....	135
~!A.....	136
~!C.....	137
~!D.....	138
~!E.....	139
~!F.....	140
~!I.....	141
~!T.....	142
<ESC>Y.....	143
<ESC>Z.....	144
<b>Message Translation Protocols.....</b>	<b>145</b>
~#.....	145
<b>Commands for Windows Driver.....</b>	<b>146</b>
!B.....	146
!J.....	147
!N.....	148
<b>File Management Commands.....</b>	<b>149</b>
DOWNLOAD.....	149
EOP.....	152
FILES.....	153
KILL.....	154
MOVE.....	156
RUN.....	157
<b>BASIC Commands and Functions.....</b>	<b>159</b>
ABS( ).....	159
ASC( ).....	160
CHR\$( ).....	161
XOR\$( ).....	162
END.....	163

EOF()	165
OPEN	167
CLOSE	169
WRITE	170
READ	171
SEEK	173
LOF()	175
LOC()	177
FREAD\$( )	179
PUT	181
GET	183
COPY	185
FOR...NEXT LOOP	187
WHILE...WEND	189
DO...LOOP	191
IF...THEN...ELSE...ENDIF LOOP	195
GOSUB...RETURN	199
GOTO	201
INP\$( )	203
INP( )	204
LOB( )	206
INPUT	208
PREINPUT	210
POSTINPUT	211
SET FILTER ON/OFF	212
REM	214
OUT	215
OUTR	217
GETKEY( )	219
INT( )	221
LEFT\$( )	223
LEN( )	224
MID\$( )	225
RIGHT\$( )	226
STR\$( )	227

STRCOMP( ) .....	229
INSTR( ) .....	231
TRIM\$( ) .....	233
LTRIM\$( ) .....	235
RTRIM\$( ) .....	237
TEXTPIXEL( ) .....	239
BARCODEPIXEL( ) .....	241
VAL( ) .....	243
BEEP .....	245
NOW\$( ) .....	246
NOW .....	247
FORMAT\$( ) .....	248
DATEADD( ) .....	254
FSEARCH( ) .....	257
TOUCHPRESS( ) .....	259
MULTIKEY( ) .....	261
RECORDSET\$( ) .....	264
LABELRATIO .....	267
REPLACE\$( ) .....	268
SET COUNTER .....	269
SET CUTTER .....	271
SET PARTIAL_CUTTER .....	273
SET BACK .....	275
SET KEYn .....	276
SET LEDn .....	279
SET PEEL .....	281
SET REWIND .....	282
SET TEAR & SET STRIPER .....	284
SET GAP .....	286
SET BLINE .....	288
SET BLINE PRINTSIDE & SET BLINE BACKSIDE .....	289
SET HEAD .....	290
SET RIBBON .....	291
SET ENCODER .....	293
SET RIBBONEND .....	294

SET COM1 .....	295
SET PRINTKEY .....	297
SET REPRINT .....	299
SET FEED_LEN .....	300
GETSENSOR() .....	301
GETSETTING\$( ) .....	305
SET USBHOST .....	310
SET AUTORUN .....	311
SET RESPONSE .....	313
SET DAYLIGHT_SAVE .....	316
SET REGISTRATION .....	318
SET APPLICATOR .....	320
SET MENULOCK .....	321
PEEL .....	322
LED1, LED2, LED3 .....	324
KEY1, KEY2, KEY3 .....	326
SET SENSOR_REF .....	328
SET RIBBON_SAVER .....	329
SET SBPLIMCMD .....	330
SET DPLIMCMD .....	331
SET ZPLIMCMD .....	332
SET COVER .....	333

**Printer Global Variables ..... 334**

@LABEL .....	334
YEAR .....	336
MONTH .....	338
DATE .....	340
WEEK .....	342
HOUR .....	343
MINUTE .....	345
SECOND .....	347
@YEAR .....	349
@MONTH .....	350
@DATE .....	351
@DAY .....	352

@HOUR .....	353
@MINUTE.....	354
@SECOND .....	355
_MODEL\$ .....	356
_SERIAL\$.....	357
_VERSION\$ .....	358
<b>Bluetooth Module Setting Commands .....</b>	<b>359</b>
BT NAME .....	359
BT PINCODE .....	360
BT PAIRMODE.....	361
BT MODE .....	362
<b>Wi-Fi Module Setting Commands .....</b>	<b>364</b>
WLAN SSID.....	364
WLAN WPA.....	365
WLAN WEP .....	366
WLAN DHCP.....	367
WLAN IP.....	368
WLAN PORT .....	369
<b>Internal Ethernet Setting Commands .....</b>	<b>370</b>
NET DHCP .....	370
NET IP .....	371
NET PORT.....	372
NET NAME .....	373
NET DNS .....	374
NET SNMP .....	375
<b>UHF RFID Setting Commands.....</b>	<b>376</b>
RFID ON/OFF .....	376
RFID ERROR .....	377
RFID RETRY .....	378
RFID RETRYERROR ON/OFF .....	379
RFID POSITION.....	380
RFID POWER .....	381
RFID COUNTRESET.....	382
RFID READ/WRITE .....	383

RFID RFLOCK .....	389
RFIDDETECT .....	392
<b>HF RFID Setting Commands .....</b>	<b>393</b>
RFID ON/OFF .....	393
RFID ERROR .....	394
RFID POSITION .....	395
RFID READ/WRITE .....	396
<b>NFC Setting Commands .....</b>	<b>398</b>
NFC FEATURE .....	398
NFC STATUS .....	399
NFC TIMEOUT .....	400
NFC READ .....	401
NFC WRITE .....	402
NFC MODE .....	403
<b>Alpha-2R/TDM Series Setting Commands .....</b>	<b>404</b>
SET PRINTQUALITY .....	404
SET STANDBYTIME .....	405
SET SLEEPTIME .....	406
<b>GPIO Setting Commands .....</b>	<b>407</b>
SET GPO .....	407
SET GPI .....	409
GPM .....	412
APPLICATOR I/O INTERFACE (DB15F) CIRCUIT DIAGRAM .....	413
APPLICATOR I/O INTERFACE (DB25F) CIRCUIT DIAGRAM .....	417
<b>Revision History .....</b>	<b>423</b>

# How to Read

**PRINT**

The command name

Description of this command

## Description

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

## Syntax

**PRINT m[,n]**

Syntax of this command

The detail description of each parameter

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

## Example

Sample code	Result
<pre> 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           </pre>	

## See Also

SET COUNTER, INPUT, DOWNLOAD

The example and printout for reference

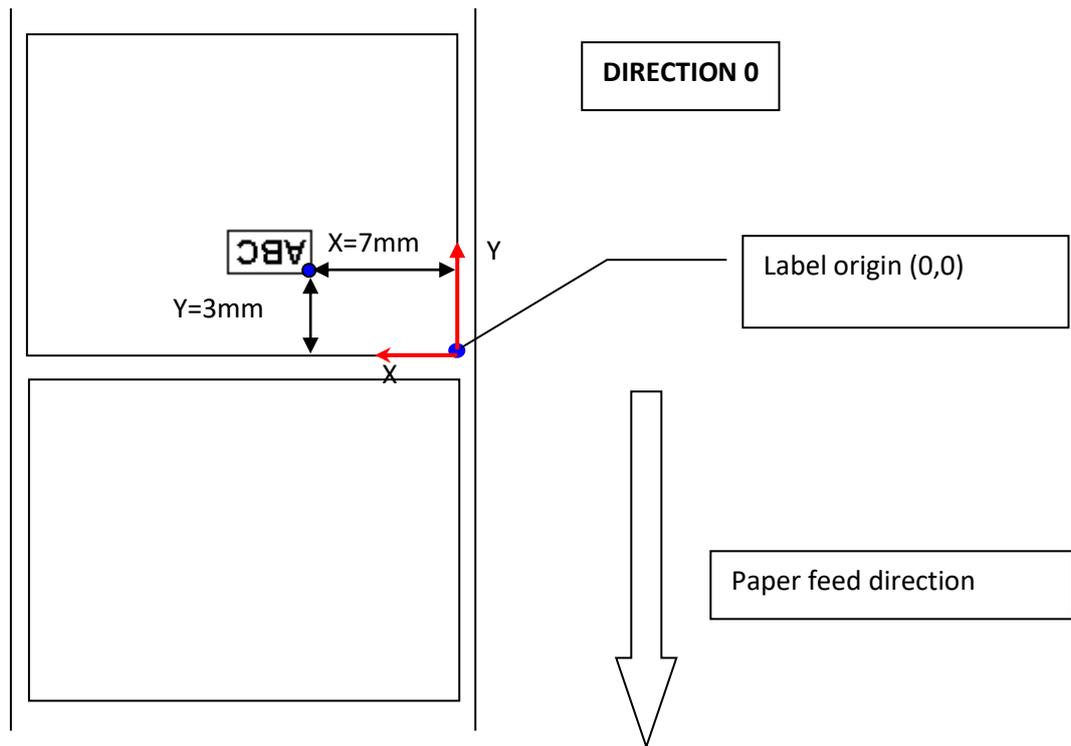
Other commands for reference

# Document Conventions

This manual uses the following typographic conventions.

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

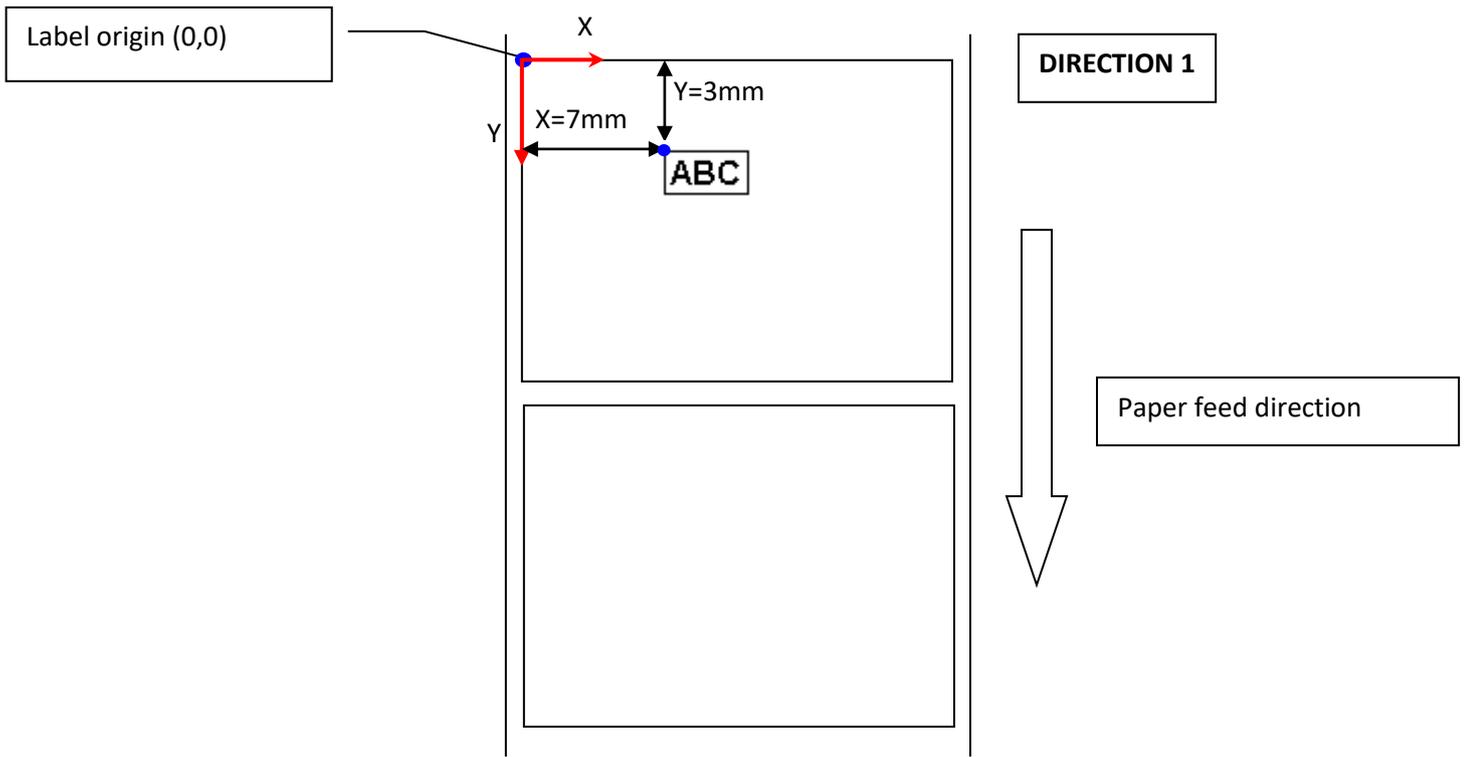
# Object Position Calculation



```
DIRECTION 0
CLS
TEXT 56,24,"3",0,1,1,"ABC"
```

**Note :**

1. 203 DPI, 1mm=8 dots ; 300 DPI, 1mm=12 dots ; 600 DPI, 1mm=24 dots
2. Only integer portion will be used. Ex. 2 mm = 23.6 dots then 23 dots will be used.



```
DIRECTION 1  
CLS  
TEXT 56,24,"3",0,1,1,"ABC"
```

## Printer Model List

Series	Models	Support Programming	F/W Version	F/W Maintainability
TDP-643 Plus	TDP-643 Plus	TSPL	V x.x	No
TTP-243 series	TTP-243, TTP-243E, TTP-342	TSPL	V x.x	No
TTP-244ME series	TTP-243M, TTP-244ME, TTP-342M	TSPL	V x.x	No
TDP-245 series	TDP-245, TDP-245G	TSPL2	V x.x	No
TTP-245 series	TTP-245, TTP-245G, TTP-343	TSPL2	V x.x	No
TTP-246M series	TTP-246M, TTP-246G, TTP-344M	TSPL2	V x.x	No
TTP-248M series	TTP-248M	TSPL2	V x.x	No
TDP-643R Plus	TDP-643R Plus	TSPL	V x.x	No
TTP-243 Plus series	TTP-243 Plus, TTP-243E Plus, TTP-342 Plus	TSPL	V x.x	No
TTP-244ME Plus	TTP-244ME Plus, TTP-342M Plus	TSPL	V x.x	No
TTP-2410M series	TTP-2410M, TTP-346M, TTP-644M	TSPL2	V x.x	No
TTP-246M Plus series	TTP-246M Plus, TTP-344M Plus			
TTP-244 series	TTP-244	TSPL2	V x.x	No
M23 series	M23	TSPL2	V x.x	No
TTP-244 Plus series	TTP-244 Plus	TSPL2	V x.x	No
TA200 series	TA200, TA300	TSPL2	V x.x	No
TTP-245C series	TTP-245C, TTP-343C	TSPL2	V x.x	No
TTP-2410M Pro series	TTP-2410M Pro, TTP-346M Pro, TTP-644M Pro	TSPL2	V x.x	No
TTP-268M series	TTP-268M, TTP-366M	TSPL2	V x.x	No
TTP-384M series	TTP-384M	TSPL2	V x.x	No
TTP-243 Pro series	TTP-243 Pro, TTP-243E Pro, TTP-342 Pro	TSPL	V x.x	Yes
TTP-244 Pro series	TTP-244 Pro	TSPL2	V x.x	Yes
TDP-247 series	TDP-245 Plus, TDP-244, TDP-247, TDP-345	TSPL2	V x.x	Yes
DA200 series	DA200, DA300	TSPL2	A x.x	Yes
TTP-247 series	TTP-245 Plus, TTP-343 Plus, TTP-247, TTP-345	TSPL2	V x.x	Yes

TE200 series	TE200, TE300	TSPL2	A x.x	Yes
TX200 series	TX200, TX300, TX600	TSPL2	A x.x	Yes
TX210 series	TX210, TX310, TX610	TSPL2	A x.x	Yes
TDP-225 series	TDP-225, TDP-324, TDP-225W, TDP-324W	TSPL2	V x.x	Yes
TTP-225 series	TTP-225, TTP-323	TSPL2	V x.x	Yes
TTP-244CE	TTP-244CE	TSPL2	V x.x	Yes
TC200 series	TC200, TC300, TC210, TC310	TSPL2	A x.x	Yes
TA210 series	TA210, TA310	TSPL2	V x.x	Yes
TTP-244M Pro series	TTP-244M Pro, TTP-244ME Pro, TTP-342M Pro, TTP-342ME Pro	TSPL2	V x.x	Yes
ME240 series	ME240, ME340	TSPL2	V x.x	Yes
MB240 series	MB240, MB340, MB240T, MB340T	TSPL2	A x.x	Yes
MB241 series	MB241, MB341, MB241T, MB341T	TSPL2	A x.x	Yes
ML240 series	ML240, ML340, ML240P, ML340P	TSPL2	A x.x	Yes
ML241 series	ML241P / ML341P	TSPL2	A x.x	Yes
TTP-246M Pro series	TTP-246M Pro, TTP-344M Pro	TSPL2	V x.x	Yes
TTP-2410MU series TTP-2410MT series	TTP-2410MU, TTP-346MU, TTP-644MU, TTP-2410MT, TTP-346MT, TTP-644MT	TSPL2	A x.x	Yes
MX240 series	MX240, MX340, MX640	TSPL2	A x.x	Yes
MX240P series	MX240P, MX340P, MX640P	TSPL2	A x.x	Yes
MX241P series	MX241P, MX341P, MX641P	TSPL2	A x.x	Yes
MH240 series	MH240, MH340, MH640, MH240T, MH340T, MH640T, MH240P, MH340P, MH640P	TSPL2	A x.x	Yes
MH241 series	MH241, MH341, MH641, MH241T, MH341T, MH641T, MH241P, MH341P, MH641P	TSPL2	A x.x	Yes
TTP-2610M series	TTP-2610MT, TTP-368MT	TSPL2	A x.x	Yes
MH261 series	MH261T, MH361T	TSPL2	A x.x	Yes
TTP-286MT series	TTP-286MT, TTP-384MT	TSPL2	A x.x	Yes
PEX-1001 series	PEX-1121, PEX-1131, PE-X1161, PEX-1221, PEX-1231, PEX-1261	TSPL2	A x.x	Yes
PEX-2000 series	PEX-2240L, PEX-2260L, PEX-2240R, PEX-2260R, PEX-2340L, PEX-2360L,	TSPL2	A x.x	Yes

	PEX-2340R, PEX-2360R, PEX-2640L , PEX-2640R			
Alpha-2R series	Alpha-2R	TSPL2	A x.x	Yes
Alpha-3R series	Alpha-3R	TSPL2	V x.x	Yes
Alpha-4L series	Alpha-4L	TSPL2	V x.x	Yes
TDM series	TDM-20, TDM-30	TSPL2	A x.x	Yes
Alpha-30R series	Alpha-30R	TSPL2	V x.x	Yes
Alpha-30L series	Alpha-30L	TSPL2	V x.x	Yes
Alpha-40L series	Alpha-40L	TSPL2	V x.x	Yes
DH220 series	DH220, DH320, DH220T, DH320T, DH220THC, DH320THC, DH220E, DH320E, DH220L, DH320L, DH220LT, DH320LT, DH220LHC, DH320LHC, DH220LTHC, DH320LTHC	TSPL2	V x.x	Yes
TH220 series	TH220, TH320, TH220T, TH320T, TH220THC, TH320THC	TSPL2	V x.x	Yes
DH240 series	DH240T, DH340, DH240T, DH340T, DH240THC, DH340THC	TSPL2	V x.x	Yes
TH240 series	TH240, TH340, TH240T, TH340T, TH240THC, TH340THC, TH240TRCHC, TH340TRCHC	TSPL2	V x.x	Yes

**The commands listed in the TSPL2 programming manual are included in all printer model firmware. The printer may not support the related commands if the function is not included in the printer specification.**

# Setup and System Commands

## SIZE

### Description

This command defines the label width and length.

### Syntax

<b>SIZE m[,n]</b>	English system (inch)
<b>SIZE m mm[,n mm]</b>	Metric system (mm)
<b>SIZE m dot[,n dot]</b>	Dot measurement <i>This command has been supported since V6.27 EZ and later firmware.</i>

<u>Parameter</u>	<u>Description</u>
M	Label width (inch/ mm/ dot)
[N]	Label length (inch/ mm/ dot); <i>This item can be optional since V8.13 &amp; A2.10 and later firmware.</i>
<b>Note :</b>	
<ul style="list-style-type: none"><li>203 DPI : 1mm = 8 dots</li><li>300 DPI : 1mm = 12 dots</li><li>600 DPI : 1mm = 24 dots</li></ul>	
<ul style="list-style-type: none"><li>For metric and dot systems, there must be a space between parameter and "mm" or "dot".</li></ul>	

### Example

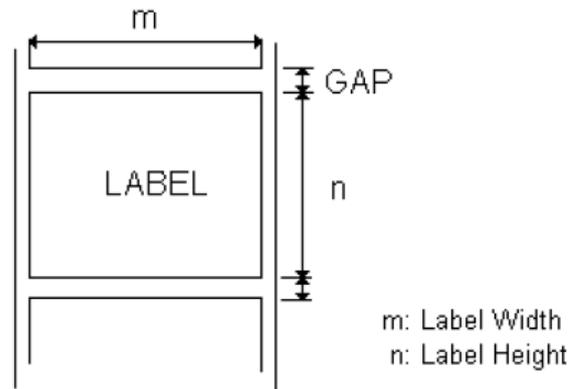
Sample Code	Result
-------------	--------

- English system (inch):

**SIZE 3.5,3.00**

- Metric system (mm):

**SIZE 100 mm,100 mm**



## See Also

GAP, BLINE

# GAP

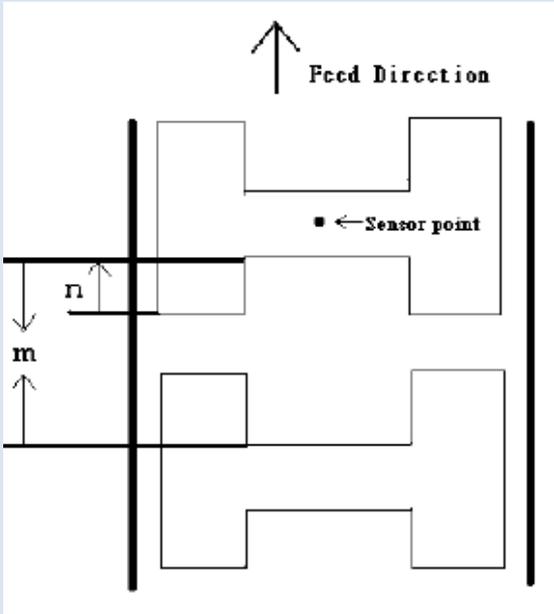
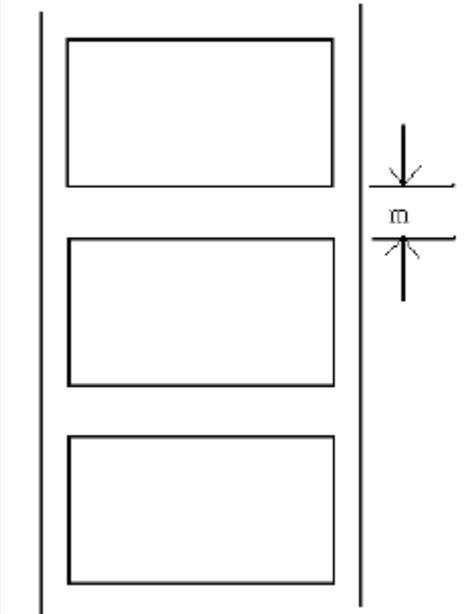
## Description

Defines the gap distance between two labels.

## Syntax

<b>GAP m,n</b>	English system (inch)
<b>GAP m mm,n mm</b>	Metric system (mm)
<b>GAP m dot,n dot</b>	Dot measurement <i>This command has been supported since V6.27 EZ and later firmware.</i>

<u>Parameter</u>	<u>Description</u>
M	The gap distance between two labels $0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch), $0 \leq m \leq 127$ (mm) / <i>since V6.21 EZ and later firmware</i>
N	The offset distance of the gap $n \leq$ label length (inch or mm)
0, 0	Continuous label



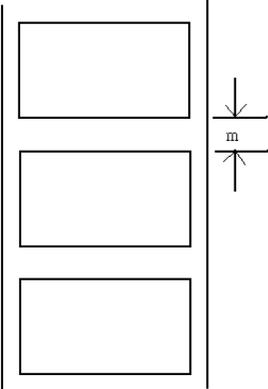
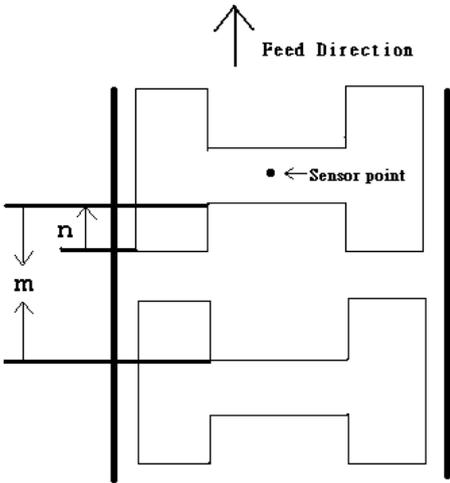
**Note :**

- 200 DPI : 1 mm = 8 dots
- 300 DPI : 1mm = 12 dots

**600 DPI : 1mm = 24 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
<p><b>Normal gap</b></p> <ul style="list-style-type: none"> <li>▪ English system (inch): <b>GAP 0.12,0</b></li> <li>▪ Metric system (mm): <b>GAP 3 mm,0 mm</b></li> <li>▪ Continuous label: <b>GAP 0,0</b></li> </ul>	<p><b>Normal gap</b></p>  <p>The diagram shows three rectangular labels stacked vertically. To the right of the labels, a vertical line indicates the gap between them. A dimension line labeled 'm' shows the distance between the bottom of one label and the top of the next, representing the gap.</p>
<p><b>Special gap</b></p> <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>	<p><b>Special gap</b></p>  <p>The diagram shows a complex label layout with a zig-zag path. A vertical line on the left and a vertical line on the right represent the edges of the label. A horizontal line with an arrow pointing left is labeled 'Sensor point'. An arrow pointing up is labeled 'Feed Direction'. Dimension lines 'n' and 'm' indicate specific distances: 'n' is the distance from the sensor point to the top edge of the label, and 'm' is the distance from the sensor point to the bottom edge of the label.</p>

## 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]**

<u>Parameter</u>	<u>Description</u>
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]**

<u>Parameter</u>	<u>Description</u>
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]**

<u>Parameter</u>	<u>Description</u>
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, BLINEDETECT

# BLINE

## Description

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

## Syntax

<b>BLINE m,n</b>	English system (inch)
<b>BLINE m mm,n mm</b>	Metric system (mm)
<b>BLINE m dot,n dot</b>	Dot measurement <i>This command has been supported since V6.27 EZ and later firmware.</i>

<u>Parameter</u>	<u>Description</u>
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) / <b>since V6.21 EZ and later firmware</b>
n	The extra label feeding length $0 \leq n \leq \text{label length}$
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 : 1mm = 12 dots  
600 DPI : 1mm = 24 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

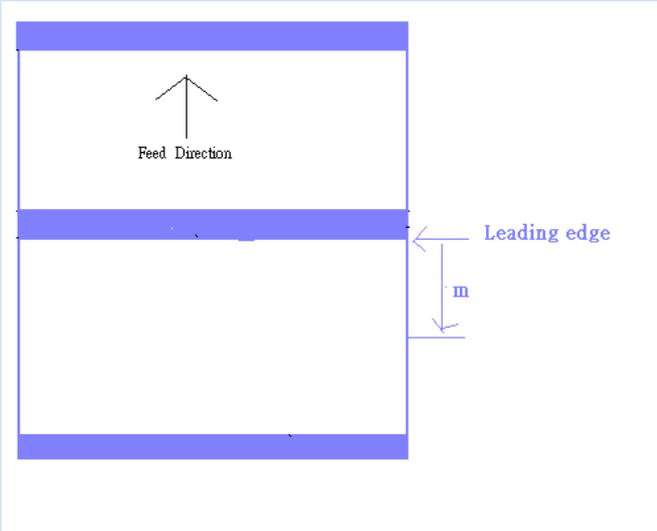
## Description

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

<b>OFFSET m</b>	English system (inch)
<b>OFFSET m mm</b>	Metric system (mm)
<b>OFFSET m dot</b>	Dot measurement <i>This command has been supported since V6.27 EZ and later firmware.</i>

<u>Parameter</u>	<u>Description</u>
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 : 1mm = 12 dots*

*600 DPI : 1mm = 24 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**

<u>Parameter</u>	<u>Description</u>																				
n	Printing speed in inch per second																				
Model / IPS	1	1.5	2	2.5	3	3.5	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
TDP-643 Plus/ TDP-643R Plus series		V	V		V																
TTP-243I/ TTP-243I Plus/ TTP-243I Pro series		V	V		V																
TTP-342/ TTP-342 Plus/ TTP-342I Pro series	V	V	V																		
TTP-244/ TTP-244 Plus series			V		V		V														
TTP-244 Pro series			V		V		V	V													
TDP-244 series			V		V		V														
TDP-245/ TDP-245 Plus/TTP-245/ TTP-245 Plus series			V		V		V	V													
TDP-247/ TTP-247 series			V		V		V	V	V	V											
TTP-343/ TTP-343 Plus series			V		V																
TDP-345/ TTP-345 series			V		V		V	V													
TTP-244CE/ TTP-343C series			V		V		V														
TTP-245C series/ TE200 series			V		V		V	V	V												
TA200/ DA300 series			V		V		V														
TA210/ DA200 series/ TE300 series			V		V		V	V													
TA300 series		V	V		V																
TA310 series		V	V		V		V														
TX200 series			V		V		V	V	V	V	V										
TX300 series		V	V		V		V	V	V												
TX600 series	V	V	V		V		V														
TDP-225/ TTP-225 series			V		V		V	V													
TDP-324/TDP-324W series			V		V		V														



Sample code

**SPEED 10**

## See Also

DENSITY

# DENSITY

## Description

This command sets the printing darkness.

## Syntax

**DENSITY n**

<u>Parameter</u>	<u>Description</u>
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]**

<u>Parameter</u>	<u>Description</u>
n	0 or 1. Please refer to the illustrations below
m	0: Print normal image 1: Print mirror image  (Note: TDP-643 Plus , TTP-243, TTP-342, TTP-244ME, TTP-342M and TTP-248M series are not supported this mirror feature)

The diagrams show the following configurations:

- DIRECTION 0,0:** Feed Direction (down arrow), TEST PRINT (upright)
- DIRECTION 1,0:** Feed Direction (down arrow), TEST PRINT (upside down)
- DIRECTION 0,1:** Feed Direction (down arrow), TEST PRINT (upright)
- DIRECTION 1,1:** Feed Direction (down arrow), TEST PRINT (upside down)

## Example

### Sample code

- **DIRECTION 0**
- **DIRECTION 0,1**

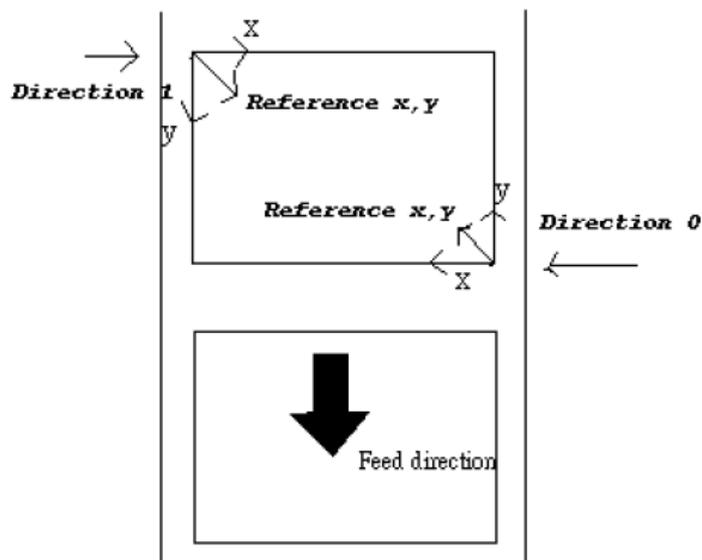
## See Also

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*

<u>Parameter</u>	<u>Description</u>
x	Horizontal coordinate (in dots)
y	Vertical coordinate (in dots)

*Note:*

*200 DPI: 1 mm = 8 dots*

*300 DPI: 1 mm = 12 dots*

*600 DPI : 1mm = 24 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

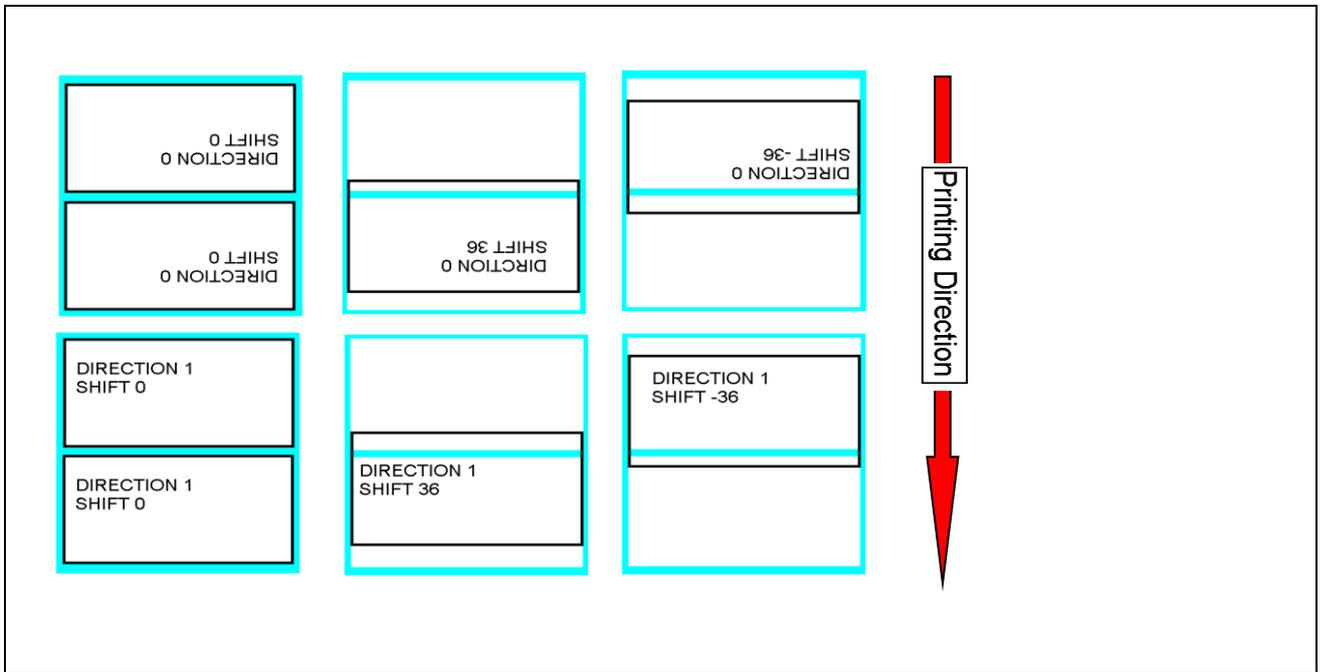
**SHIFT [x,] y**

<u>Parameter</u>	<u>Description</u>
x	Optional. The maximum value is 1 inch. For 200 dpi printers, the range is –203 to 203; for 300 dpi printers, the range is –300 to 300. The unit is dot.
Y	The maximum value is 1 inch. For 200 dpi printers, the range is –203 to 203; for 300 dpi printers, the range is –300 to 300. The unit is dot.

(Note: TDP-643 Plus , TTP-243, TTP-342, TTP-244ME, TTP-342M, TTP-248M and M23 series are not supported this feature)

## Example

Sample Code
<pre>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</pre>
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**

<u>Parameter</u>	<u>Description</u>
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, ~!!

# CODEPAGE

## Description

This command defines the code page of international character set.

## Syntax

**CODEPAGE n**

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

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

## Example

Sample Code (Download the COUR.TTF into printer)	Result
<pre> <b>DOWNLOAD "TEST.BAS"</b>  <b>str1\$ = " "</b>  <b>J = 0</b>  <b>y = 50</b>  <b>CODEPAGE 1252</b>  <b>SIZE 4,3</b>  <b>GAP 0,0</b>  <b>DIRECTION 1</b>  <b>CLS</b>  <b>TEXT 10,10,"COUR.TTF",0,12,12,"CODEPAGE 1252"</b>  <b>FOR I=32 TO 255</b>  <b>str1\$=str1\$+CHR\$(I) + " "</b>  <b>J=J+1</b>  <b>IF J=16 THEN GOSUB drawTEXT</b>  <b>NEXT</b>  <b>PRINT 1</b>  <b>END</b>  <b>drawTEXT:</b>  <b>TEXT 10,y,"COUR.TTF",0,12,12,str1\$</b>  <b>str1\$=" "</b>  <b>J=0</b>  <b>y=y+40</b>  <b>RETURN</b>  <b>EOP</b>  <b>TEST</b> </pre>	<pre> CODEPAGE 1252 ! " # \$ % &amp; ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; &lt; = &gt; ? @ 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 &lt; Œ Ž \ / " " • - - ~ ™ š &gt; œ ž Ÿ ı ċ £ ¨ ¥ ¦ § ¨ © ª « ¬ - ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð 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 ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ </pre>

## See Also

COUNTRY, ~!l

# CLS

## Description

This command clears the image buffer.

## Syntax

CLS

<u>Parameter</u>	<u>Description</u>
None	N/A

*Note:*  
*This command must be placed after SIZE command.*

## Example

```
Sample code

CLS
```

## See Also

SIZE, GAP, BLINE

# FEED

## Description

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

## Syntax

**FEED n**

<u>Parameter</u>	<u>Description</u>
n	unit: dot $1 \leq n \leq 9999$
<i>Note:</i>	
<i>200 DPI: 1 mm = 8 dots</i>	
<i>300 DPI: 1 mm = 12 dots</i>	
<i>600 DPI : 1mm = 24 dots</i>	

## Example

Sample code
<b>FEED 40</b>

## 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

**Note:** Please refer to [printer model list](#) for checking TSPL or TSPL2.

<u>Parameter</u>	<u>Description</u>
n	unit: dot $1 \leq n \leq 9999$
<i>Note:</i>	
<i>200 DPI: 1 mm = 8 dots</i>	
<i>300 DPI: 1 mm = 12 dots</i>	
<i>600 DPI : 1mm = 24 dots</i>	
<i>CAUTION:</i>	
<i>Impropriety back feed value may cause paper jam or wrinkle.</i>	

## Example

Sample code
<ul style="list-style-type: none"><li>TSPL printers <b>BACKUP 40</b></li><li>TSPL2 printers <b>BACKFEED 40</b></li></ul>

## See Also

FEED, SIZE, GAP, BLINE, HOME, FORMFEED

# FORMFEED

## Description

This command feeds label to the beginning of next label.

## Syntax

### FORMFEED

<u>Parameter</u>	<u>Description</u>
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>	

## 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

<u>Parameter</u>	<u>Description</u>
None	N/A
For TSPL programming printer: Back label to origin position	
For TSPL2 programming printer: Feed label to origin position	
<b>Note:</b> Please refer to <a href="#">printer model list</a> for checking TSPL or TSPL2.	

## 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]**

<u>Parameter</u>	<u>Description</u>
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	Result
<pre>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</pre>	<p>0003 0003 0002 0002 0001 0001</p> <p>1 set, 2 copies</p> <p>Paper feed direction ↓</p>

## 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**

<u>Parameter</u>	<u>Description</u>
level	Sound level: 0~9
interval	Sound interval: 1~4095

## Example

### Sample code

- **SOUND 5,200**
- **SOUND 3,200**
- **SOUND 3,200**
- **SOUND 4,200**
- **SOUND 2,200**
- **SOUND 2,200**
- **SOUND 1,200**
- **SOUND 2,200**
- **SOUND 3,200**
- **SOUND 4,200**
- **SOUND 5,200**

# CUT

## Description

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

## Syntax

**CUT**

<u>Parameter</u>	<u>Description</u>
None	N/A

## Example

```
Sample code

SIZE 3,3
GAP 0,0
CLS
BOX 0,0,866,866,5
TEXT 100,100, "5",0,1,1, "FEED & CUT"
TEXT 100,200, "5",0,1,1, "300 DPI"
PRINT 1,1
FEED 260
CUT
```

## See Also

SET CUTTER, SET BACK, SET PARTIAL\_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 a 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>

<u>Parameter</u>	<u>Description</u>
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
<ul style="list-style-type: none"><li>• English system (inch) <b>LIMITFEED 12</b> <b>LIMITFEED 10, 2.36, 0.12</b></li><li>• Metric system (mm) <b>LIMITFEED 250 mm, 60 mm, 3 mm</b></li><li>• Dot measurement <b>LIMITFEED 2000 dot, 480 dot, 24 dot</b></li></ul>

# SELFTEST

## Description

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

## Syntax

**SELFTEST** [page]

<u>Parameter</u>	<u>Description</u>
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. <b>BT</b> : Print a self-test page with Bluetooth settings.

## Example

Sample code	Result
SELFTEST	<pre> ----- SYSTEM INFORMATION ----- MODEL: TDP247 FIRMWARE: 7.00 EZ CHECKSUM: 07CBD355 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 -----  </pre> 
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>

Sample code	Result
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 EOI) be finished then go on the next command.

## Syntax

**EOJ**

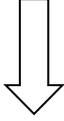
*Note:*

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

## Example

Sample Code
<pre>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</pre>
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

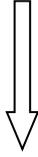
**DELAY** ms

<u>Parameter</u>	<u>Description</u>
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
<pre>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</pre>
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/.....**

<u>Parameter</u>	<u>Description</u>														
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,width,height,thick	Draw the bar in printer's image buffer on LCD panel (since A2.x EZ)														
x,y,width,height,thick,radius															
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)														
x,y, "font", rotate, "content"	Show the text in printer's image buffer on LCD panel (since A2.x EZ)														
x,y, "font", rotate, multi, "content"															
x,y, "font", rotate, x-multi, y-multi, "content"															
x,y, "font", rotate, x-multi, y-multi, align, "content"															
<table border="1"> <tbody> <tr> <td>forecolor</td> <td>RGB color code for text or bar (decimal)</td> </tr> <tr> <td>backcolor</td> <td>RGB color code for background (decimal)</td> </tr> <tr> <td>x</td> <td>Horizontal multiplication</td> </tr> <tr> <td>y</td> <td>Vertical multiplication</td> </tr> <tr> <td>width</td> <td>Frame width</td> </tr> <tr> <td>height</td> <td>Frame height</td> </tr> <tr> <td>thick</td> <td>Frame thickness</td> </tr> </tbody> </table>		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	thick	Frame thickness
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														
thick	Frame thickness														

radius	Frame radius
bmpfile	BMP file name
font	Font name
rotate	Rotation (0, 90, 180, and 270 valid)
x-multi	Horizontal multiplication
y-multi	Vertical multiplication
align	Text justification (1:left, 2:center, 3:right)
content	Content of text string

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

## Example

Sample code	Result
<pre>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</pre>	
<pre>CLS DISPLAY 15128749,16711680 DISPLAY CLS DISPLAY 10,30, "1", "1234567890" DELAY 5000 DISPLAY OFF</pre>	

# INITIALPRINTER

## Description

This command can restore printer settings to defaults.

## Syntax

**INITIALPRINTER**

<u>Parameter</u>	<u>Description</u>
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

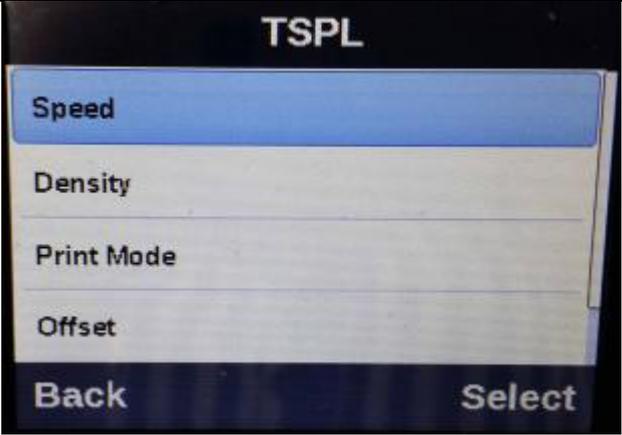
**MENU title\$, list\$, selected**

<u>Parameter</u>	<u>Description</u>
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.

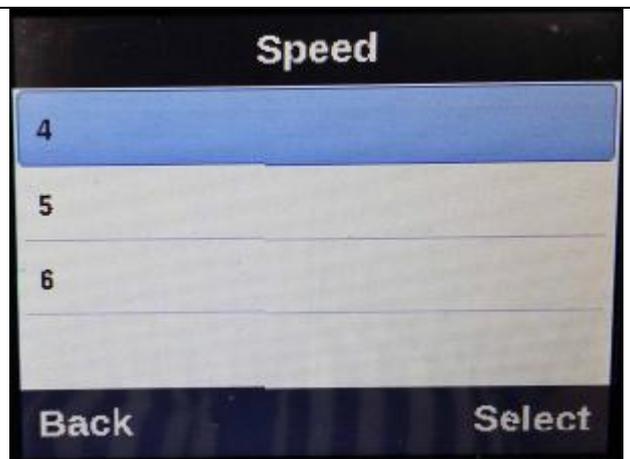
## Example

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</pre>	

```

10
11
12
EOP
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$("CONFIG","TSPL","COUNTRY CODE")
IF LOF(OPTION$) <> 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$) <> 0 THEN
    IF OPTION$ = "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"

```

# VERTICAL

## Description

This command can enable the vertical adjustment function and setup value of it. (Since A2.16)

## Syntax

**VERTICAL n**

<u>Parameter</u>	<u>Description</u>
n	n=OFF, disable vertical adjustment function n=90.0 ~ 105.0, setup percentage (%) to adjust vertical length

## Example

Sample code
<b>VERTICAL OFF</b>
<b>VERTICAL 95.0</b>

# EXPORT

## Description

This command is used to back up the current printer firmware or printer configuration to an external storage device, such as a USB flash drive or SD card. (Since A2.18)

The exported file (e.g., AUTO.NEW or AUTO.CFG) can be used to update other printers. This function is supported on printers with USB host or SD card slot.

## Syntax

1. Export the printer firmware:

**EXPORT FIRMWARE** [dev,]"filename"

2. Export the printer configuration:

**EXPORT CONFIG** [dev,]"filename"

<u>Parameter</u>	<u>Description</u>
dev	<p>Specify storage device used to save export file.</p> <p><b>E</b>: Expansion memory module (Auto select the installed storage device)</p> <p><b>U</b>: Export to USB flash drive</p> <p><b>C</b>: Export to SD memory card</p> <p><b>dev is ignored (= E)</b>: If <b>dev</b> is not specified, the system will use the default storage device (the system will automatically select the installed storage device, such as if both a USB drive and an SD card are connected, the SD card will be given priority).</p>
filename	<p>The filename resident in storage device.</p> <p>If you want to automatically load and update to other printers later, the file name must be <b>AUTO.XXX</b>.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"><li>• <b>Filenames are case sensitive.</b></li><li>• <b>File extensions must be ".NEW" for printer firmware.</b></li><li>• <b>File extensions must be ".CFG" for printer configuration.</b></li></ul>

## Example

### Sample code

(The following example program will download the printer firmware to an USB drive, and automatically update the firmware downloaded to the USB drive to another printer of the same model.

1. Install the USB drive to the printer.
2. Send the following command to load the printer firmware to the USB drive, the file name will be "AUTO.NEW".

**EXPORT FIRMWARE U,"AUTO.NEW"**

3. Then, install the USB flash drive to another printer. After the printer is turned on, it will automatically update the "AUTO.NEW" firmware to the printer.

# Label Formatting Commands

## BAR

### Description

This command draws a bar on the label format.

### Syntax

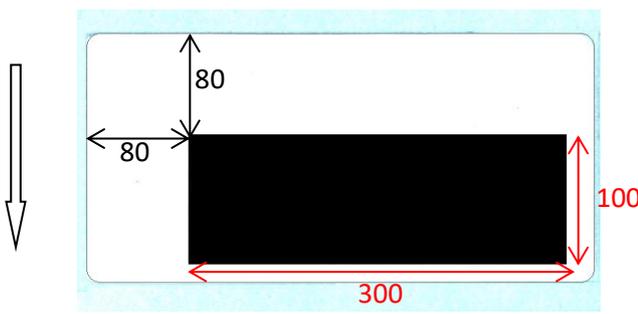
**BAR x,y,width,height**

<u>Parameter</u>	<u>Description</u>
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*
- *600 DPI : 1mm = 24 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 25% respectively.*

### Example

Sample code	Result
SIZE 50 mm,25 mm GAP 3 mm,0 DIRECTION 1 CLS BAR 80,80,300,100 PRINT 1,1	

--	--

## 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 manually.	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. <i>*Since V6.89EZ.</i>		V	V	V		
<b>TELEPENN</b>	Telepen number. <i>*Since V6.89EZ.</i>		V	V	V		
<b>PLANET</b>	Planet. <i>*Since V6.89EZ.</i>	V					
<b>CODE49</b>	Code 49. <i>*Since V6.89EZ.</i>	V					
<b>DPI</b>	Deutsche Post Identcode. <i>*Since V6.91EZ.</i>		V	V	V		11
<b>DPL</b>	Deutsche Post Leitcode. <i>*Since V6.91EZ.</i>		V	V	V		13
<b>LOGMARS</b>	A special use of Code 39. <i>*Since V6.88EZ.</i>		V	V	V		

## Syntax

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

<u>Parameter</u>	<u>Description</u>																																												
X	Specify the x-coordinate bar code on the label																																												
Y	Specify the y-coordinate bar code on the label																																												
code type																																													
128	Code 128, switching code subset A, B, C automatically																																												
128M	Code 128, switching code subset A, B, C manually <table border="1" data-bbox="486 622 1125 1205"> <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 colspan="3">Start (CODE A)</td> </tr> <tr> <td>104</td> <td colspan="3">Start (CODE B)</td> </tr> <tr> <td>105</td> <td colspan="3">Start (CODE C)</td> </tr> </tbody> </table> <p><i>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.</i></p>	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)		
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)																																												
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 Auto switch full ASCII and standard code 39 for <b>PLUS</b> models <i>Note: Please refer to <a href="#">printer model list</a> for detail.</i>																																												
39C	Code 39 full ASCII with check digit for TSPL2 printers Code 39 standard with check digit for TSPL printers Auto switch full ASCII and standard code 39 for <b>PLUS</b> models <i>Note: Please refer to <a href="#">printer model list</a> for detail.</i>																																												
39S	Code 39 standard for TSPL2 printers																																												

	<i>Note: Please refer to <a href="#">printer model list</a> for detail.</i>
93	Code 93
EAN13	EAN 13
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
<p>Note:</p> <p>* TDP-643 Plus , TTP-243, TTP-342, TTP-244ME and TTP-342M models are not supported MSI, MSIC, PLESSY, ITF14, EAN14 and 11.</p> <p>* TTP-248M model are not supported MSIC and 11.</p>	
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

narrow                 Width of narrow element (in dots)

wide                    Width of wide element (in dots)

	narrow : wide				
	1:1	1:2	1:3	2:5	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	-

alignment

Specify the alignment of barcode

0 : default (Left)

1 : Left

2 : Center

3 : Right

content

Content of barcode

**Please note that the maximum number of bar code content.**

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

**Note:**

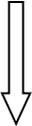
Since V5.10EZ, \[R] means carriage return character 0x0D and \[L] means line feed character 0x0A.

## 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

Sample Code	Result
<p>SIZE 4,1</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>TEXT 10,10, "2",0,1,1, "Human readable alignment"</p> <p>BARCODE 10,50, "128",100,1,0,2,2,"left"</p> <p>BARCODE 310,50, "128",100,2,0,2,2,"center"</p> <p>BARCODE 610,50, "128",100,3,0,2,2,"right"</p> <p>PRINT 1</p>	<p>Human readable alignment</p>  <p>left center right</p>
<p>SIZE 4,1</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>TEXT 10,10, "2",0,1,1, "Code 128, switch code subset automatically. "</p> <p>BARCODE 10,50, "128",100,1,0,2,2, "123456abcd123456"</p> <p>PRINT 1</p>	<p>Code 128, switch code subset automatically.</p>  <p>123456abcd123456</p>
<p>SIZE 4,1</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>TEXT 10,10, "2",0,1,1, "Code 128, switch code subset manually."</p> <p>BARCODE 10,50, "128M",100,1,0,2,2, "!104!096ABCD!101EFGH"</p> <p>PRINT 1</p> <p><b>Note:</b></p> <p><i>The above example of code 128M encoded with CODE B start character. The next character will be the code 128 function character FNC3 which is then followed by the ABCD characters</i></p>	<p>Code 128, switch code subset manually.</p>  <p>ABCDEFGH</p>

Sample Code	Result
<p><i>and EFGH characters encoded as CODE A subset.</i></p>	
<p>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, "abcd1234ABCD"  PRINT 1</p>	<p>TELEPEN</p>  <p>abcd1234ABCD</p>
<p>SIZE 4,4  GAP 0,0  DIRECTION 1  CLS  TEXT 400,26, "2",0,1,1,2, "TELEPEN Number"  BARCODE 400,50, "TELEPEN",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</p>	 <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>01234567890128 Code 49</p>  <p>1234567890</p>

## Example for GS1 Code128

### Ex1: GS1 format for GTIN and Serial Number

Data	Description
~1	FNC1
01	AI for GTIN
12345678901234	GTIN
21	AI for Serial number
12345	Serial number

**DIRECTION 1**

**CLS**

**BARCODE 150,50, "128M",100,1,0,2,2,"!105!10201123456789012312112345"**

**PRINT 1**

### Ex2: GS1 format for GTIN, Serial Number and Batch number

Data	Description
~1	FNC1
01	AI for GTIN
12345678901234	GTIN
21	AI for Serial number
12345	Serial number
10	AI for Batch number
ABCD1234	Batch number

**DIRECTION 1**

**CLS**

**BARCODE 150,50, "128M",100,1,0,2,2,"!105!102011234567890123421123456!10210!100ABCD1234"**

**PRINT 1**

# 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"**

<u>Parameter</u>	<u>Description</u>
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
<pre>SIZE 4,1.2 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "TLC39 code"</pre>

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

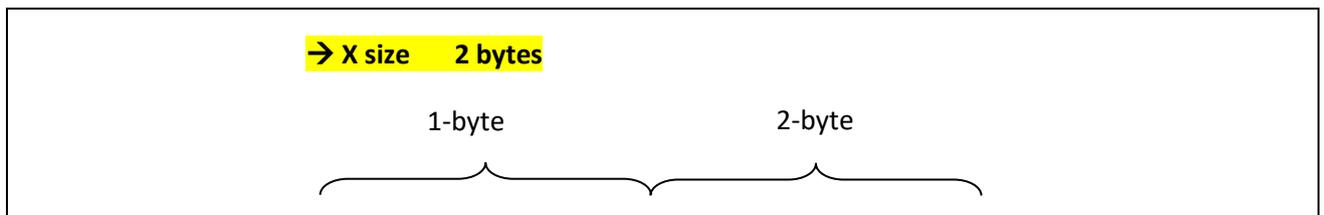
This command draws bitmap images (as opposed to BMP graphic files).

## Syntax

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

<u>Parameter</u>	<u>Description</u>
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



→ Y size 16 dots

	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
6	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1
7	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1
8	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1
9	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1	1
10	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
11	0	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1
12	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	1
13	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	1
14	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1
15	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
16	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

		X – axis			
Y- 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
<pre> SIZE 4,2 GAP 0,0 CLS BITMAP 200,200,2,16,0, _____ ???? PRINT 1,1 </pre>	<pre> 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 </pre>	<pre> ↖ </pre>

**See Also**

PUTBMP, PUTPCX

# BOX

## Description

This command draws rectangles on the label.

## Syntax

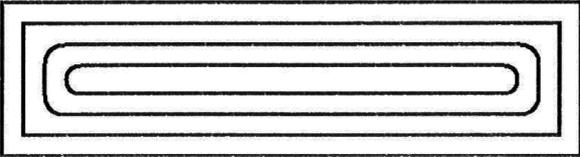
**BOX x,y,x\_end,y\_end,line thickness[,radius]**

<u>Parameter</u>	<u>Description</u>
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.  <i>*Since V5.28 EZ</i>

**Note:**

- *200 DPI : 1 mm = 8 dots*  
*300 DPI : 1 mm = 12 dots*  
*600 DPI : 1mm = 24 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 25% 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</pre>	

<b>PRINT 1</b>	
----------------	--

## See Also

BAR

# CIRCLE

## Description

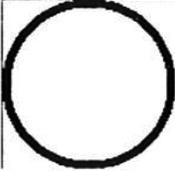
This command draws a circle on the label.

## Syntax

**CIRCLE X\_start,Y\_start,diameter,thickness**

<u>Parameter</u>	<u>Description</u>
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.

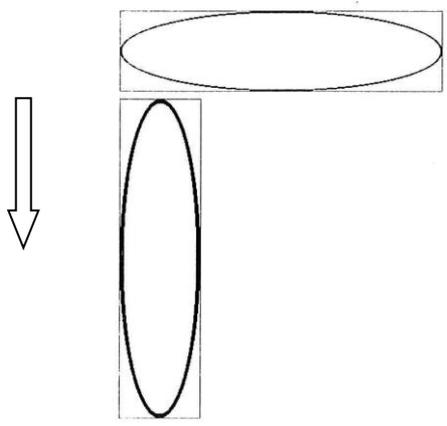
## Syntax

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

<u>Parameter</u>	<u>Description</u>
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"**

<u>Parameter</u>	<u>Description</u>
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.*

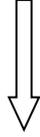
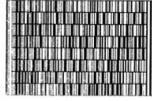
## Example

```
Sample Code

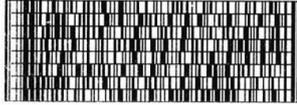
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Codablock F"
CODABLOCK 10,50,0, "We stand behind our products with one of the most comprehensive support programs in the Auto-ID industry."
PRINT 1
CLS
TEXT 10,10, "3",0,1,1, "Codablock F"
CODABLOCK 10,50,0,16,1, "We stand behind our products with one of the most comprehensive support programs in the Auto-ID industry."
```

Result

Codablock F



Codablock F



# 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"**

<u>Parameter</u>	<u>Description</u>																																																																																																												
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.																																																																																																												
	<table border="1"> <thead> <tr> <th>~X</th> <th>Hex</th> <th>ASCII</th> <th>~X</th> <th>HEX</th> <th>ASCII</th> <th>~X</th> <th>HEX</th> <th>ASCII</th> <th>~X</th> <th>HEX</th> <th>ASCII</th> </tr> </thead> <tbody> <tr> <td>~@</td> <td>00</td> <td>NUL</td> <td>~H</td> <td>08</td> <td>BS</td> <td>~P</td> <td>10</td> <td>DLE</td> <td>~X</td> <td>18</td> <td>CAN</td> </tr> <tr> <td>~A</td> <td>01</td> <td>SOH</td> <td>~I</td> <td>09</td> <td>HT</td> <td>~Q</td> <td>11</td> <td>DC1</td> <td>~Y</td> <td>19</td> <td>EM</td> </tr> <tr> <td>~B</td> <td>02</td> <td>STX</td> <td>~J</td> <td>0A</td> <td>LF</td> <td>~R</td> <td>12</td> <td>DC2</td> <td>~Z</td> <td>1A</td> <td>SUB</td> </tr> <tr> <td>~C</td> <td>03</td> <td>ETX</td> <td>~K</td> <td>0B</td> <td>VT</td> <td>~S</td> <td>13</td> <td>DC3</td> <td>~[</td> <td>1B</td> <td>ESC</td> </tr> <tr> <td>~D</td> <td>04</td> <td>EOT</td> <td>~L</td> <td>0C</td> <td>FF</td> <td>~T</td> <td>14</td> <td>DC4</td> <td>~\</td> <td>1C</td> <td>FS</td> </tr> <tr> <td>~E</td> <td>05</td> <td>ENQ</td> <td>~M</td> <td>0D</td> <td>CR</td> <td>~U</td> <td>15</td> <td>NAK</td> <td>~]</td> <td>1D</td> <td>GS</td> </tr> <tr> <td>~F</td> <td>06</td> <td>ACK</td> <td>~N</td> <td>0E</td> <td>SO</td> <td>~V</td> <td>16</td> <td>SYN</td> <td>~^</td> <td>1E</td> <td>RS</td> </tr> <tr> <td>~G</td> <td>07</td> <td>BEL</td> <td>~O</td> <td>0F</td> <td>SI</td> <td>~W</td> <td>17</td> <td>ETB</td> <td>~_</td> <td>1F</td> <td>US</td> </tr> </tbody> </table>	~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									
~X	Hex	ASCII	~X	HEX	ASCII	~X	HEX	ASCII	~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#	150 : 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

**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	72 x 72	8 x 18
12 x 12	32 x 32	80 x 80	8 x 32
14 x 14	36 x 36	88 x 88	12 x 26
16 x 16	40 x 40	96 x 96	12 x 36
18 x 18	44 x 44	104 x 104	16 x 36
20 x 20	48 x 48	120 x 120	16 x 48
22 x 22	52 x 52	132 x 132	
24 x 24	64 x 64	144 x 144	

## Example

Sample code	Result
<p>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</p>	
<p><b>Sample code for FNC</b></p> <p>SIZE 4,1            GAP 0,0            CLS            DIRECTION 1            DMATRIX            100,50,100,100,c126,x6,18,18, « ~1241sPn~110sLot~130sQty »            PRINT 1</p>	
<p><b>Sample code in rectangular shape</b></p> <p>SIZE 4,1            GAP 0,0            DIRECTION 1            CLS            DMATRIX 100,110,600,600,a1, »DMATRIX EXAMPLE 1 »            PRINT 1,1</p>	

## Example for GS1 DataMatrix

### Ex1: GS1 format for GTIN and Serial Number

Data	Description
~1	FNC1
01	AI for GTIN
12345678901234	GTIN
21	AI for Serial number
12345	Serial number

**DIRECTION 1**

**CLS**

**DMATRIX 150,50,100,100,c126,x6,18,18, "~101123456789012312112345"**

**PRINT 1**

### Ex2: GS1 format for GTIN, Serial Number and Batch number

Data	Description
~1	FNC1
01	AI for GTIN
12345678901234	GTIN
21	AI for Serial number
12345	Serial number
10	AI for Batch number
ABCD1234	Batch number

**DIRECTION 1**

**CLS**

**DMATRIX 150,50,100,100,c126,x6,18,18, "~101123456789012312112345~110ABCD1234"**

**PRINT 1**

# ERASE

## Description

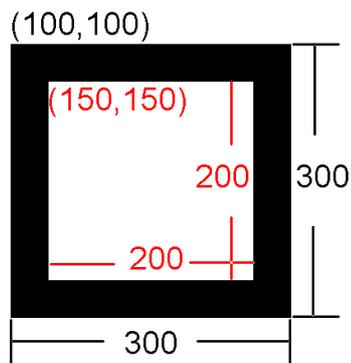
This command clears a specified region in the image buffer.

## Syntax

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

<u>Parameter</u>	<u>Description</u>
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

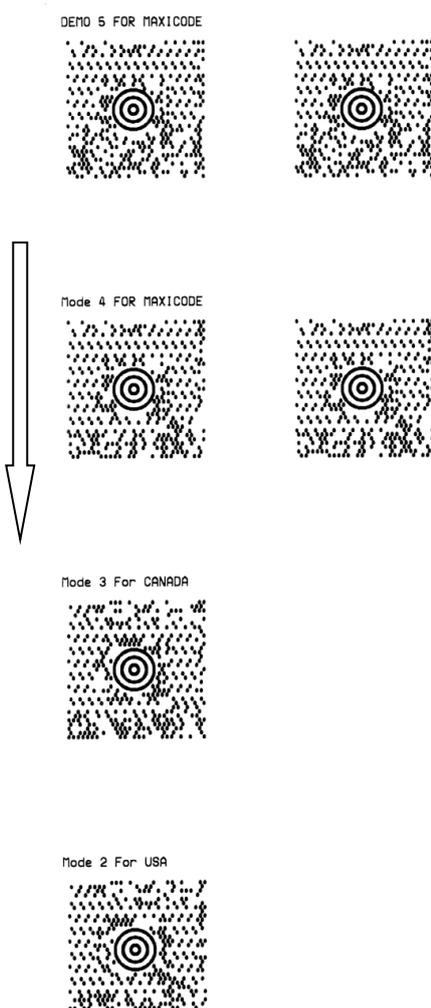
This command defines a 2D Maxicode.

## Syntax

<b>MAXICODE x,y,mode,[class,country,post,Lm,] "content"</b>	
<b>MAXICODE x,y,mode,class,country,postal code, "content"</b>	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.
<b>MAXICODE x,y,mode,[Lm,] "content"</b>	For mode 4,5,6,  AIM special format is supported, see page 23 in the spec.  <b>Mode 6 is not supported in TSPL2 printer firmware.</b>

<u>Parameter</u>	<u>Description</u>
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
	<b>Note:</b>  <i>If parameter Lm is used, double quotes (") are unnecessary.</i>

# Example

Sample code	Result
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS  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 </pre>	 <p>DEMO 5 FOR MAXICODE</p> <p>Mode 4 FOR MAXICODE</p> <p>Mode 3 For CANADA</p> <p>Mode 2 For USA</p>

## Description

This command defines a PDF417 2D bar code.

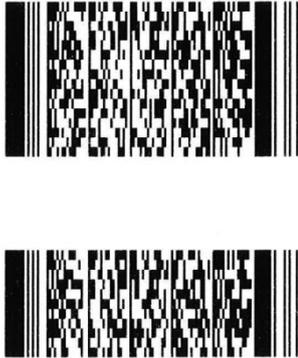
## Syntax

**PDF417 x,y,width,height,rotate,[option], "content"**

<u>Parameter</u>	<u>Description</u>																
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	<table border="1"> <tbody> <tr> <td>P</td> <td>Data compression method  0: Auto encoding  1: Binary mode</td> </tr> <tr> <td>E</td> <td>Error correction level (Range: 0~8)</td> </tr> <tr> <td>M</td> <td>Center pattern in barcode area  0: The pattern will print upper left justified the area  1: The pattern is printed middle of area</td> </tr> <tr> <td>Ux,y,c</td> <td>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</td> </tr> <tr> <td>W</td> <td>Module width in dot (Range: 2~9)</td> </tr> <tr> <td>H</td> <td>Bar height in dot (Range: 4~99)</td> </tr> <tr> <td>R</td> <td>Maximum number of rows</td> </tr> <tr> <td>C</td> <td>Maximum number of columns</td> </tr> </tbody> </table>	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
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  <i>Note:</i>  <i>If parameter Lm is used, double quotes (") are unnecessary for content.</i>

## Example

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  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  REM *****OPTION:E4 W4 H4***** CLS           </pre>	

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

Sample code	Result
<p>Module Width 4 dots: W4</p> <p>Bar Height 4 dots: H4</p> <p>Maximum Number of Rows: 60 Rows: R60</p> <p>Maximum number of columns: 4 Cols: C4</p> <p>Truncation:1: T0</p> <p>Expression length:297: L297</p> <p>PRINT 1,1</p>	

# AZTEC

## Description

This command defines a AZTEC 2D bar code.

## Syntax

<code>AZTEC x,y,rotate,[size,]ecp,[flg,]menu,[multi,]rev,] "content"</code>	Since V6.60EZ
<code>AZTEC x,y,rotate,size,ecp,flg,menu,multi,rev,bytes,content</code>	Since V6.91EZ

<u>Parameter</u>	<u>Description</u>
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
	<b>Note:</b>  <i>If parameter bytes is used, double quotes (") are unnecessary.</i>

## Example

Sample Code	Result
SIZE 4,2 GAP 0,0 CLS	
AZTEC 10,10,0,"ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789" AZTEC 210,10,0,4,"ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789" AZTEC 410,10,0,4,1,"ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789" AZTEC 610,10,0,4,1,0,"ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789" AZTEC 10,310,0,4,1,0,0,"ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789" AZTEC 210,310,0,4,1,0,0,1,"ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789" AZTEC 410,310,0,4,1,0,0,1,1,"ABCDEFGHJKLMNOPQRSTUVWXYZ0123456789" AZTEC 610,310,0,4,1,0,0,1,1,10,1234567890	
PRINT 1	

# MPDF417

## Description

This command defines a Micro PDF 417 bar code.

## Syntax

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

<u>Parameter</u>	<u>Description</u>
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

SIZE 4,1

GAP 0,0

CLS

MPDF417 10,10,0, « ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 «

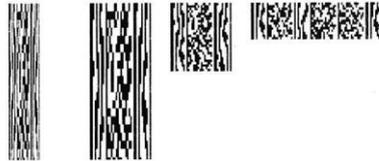
MPDF417 110,10,0,W2, « ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 «

MPDF417 210,10,0,W2,H3, « ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 «

MPDF417 310,10,0,W2,H3,C3, « ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 «

PRINT 1

### Result



# 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]**

<u>Parameter</u>	<u>Description</u>
x	The x-coordinate of the BMP format image
y	The y-coordinate of the BMP format image
filename	The downloaded BMP filename
bpp	Optional. Bits per pixel of grayscale graphic. Default is 1. <b><i>*Since V6.91EZ.</i></b> 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. <b><i>*Since V6.91EZ.</i></b>

Note: TDP-643 Plus, TTP-243, TTP-342, TTP-244ME, TTP-342M, TTP-248M and **mobile barcode printer** series are not supported this PUTBMP command.

## Example

Sample Code	Result
<p><b>Downloading BMP file into printer</b></p>  <pre>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</pre>	 <p>bpp and contrast are omitted.</p> <p>bpp = 1 contrast = 80</p> <p>bpp = 8 contrast = 80</p>

```
contrast = 80"
PUTBMP 590,10, "SAMPLE.BMP",8,80
BLOCK 590,180,240,100,"2",0,1,1,"bpp = 8
contrast = 80"
PRINT 1
```

**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.

*Note: Please refer to [printer model list](#) for checking TSPL or TSPL2.*

## Syntax

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

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

## Example

Sample Code	Result
<p><b>Downloading PCX file into printer</b></p>  <pre>SPEED 2 DENSITY 3 SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS PUTBMP 10,10, "SAMPLE.PCX" PRINT 1</pre>	

## See Also

DOWNLOAD, BITMAP, PUTPCX

# PUTPNG

## Description

This command prints PNG format images. (since A2.12)

## Syntax

**PUTPNG x,y, "filename"**

<u>Parameter</u>	<u>Description</u>
x	The X-coordinate of the PNG format image
y	The Y-coordinate of the PNG format image
filename	The downloaded PNG file name ( <i>Case sensitive</i> )

## Example

### Sample Code (Downloading PNG file into printer)

```
SPEED 2  
DENSITY 3  
SIZE 4,1  
GAP 0,0  
CLS  
PUTPNG 100,100, "test.png"  
PRINT 1
```

## See Also

DOWNLOAD, BITMAP, PUTPCX

# QR CODE

## Description

This command prints QR code.

## Syntax

**QR CODE x,y,ECC Level,cell width,mode,rotation,[justification,]model,[mask,]area],[length]"content"**

<u>Parameter</u>	<u>Description</u>
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 (dot)	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.
[length]	Content length; since version A2.13 firmware.

Content

The encodable character set is described as below,

**Encodable character set:**

1) Numeric data: (digits 0~9)

2) Alphanumeric data

Digits 0-9

Upper case letters A-Z

Nine other characters: space, \$ % \* + - . / : )

3) 8-bit byte data

JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201

4) Kanji characters

Shift JIS values 8140<sub>HEX</sub> –9FFC<sub>HEX</sub> and E040<sub>HEX</sub> –EAA4<sub>HEX</sub>. These are values shifted from those of JIS X 0208. Refer to JIS X 0208 Annex 1 Shift Coded Representation for detail.

**Data characters per symbol (for maximum symbol size):**

	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.

Manual mode example:

**QR CODE 100,10,L,7,M,0,M1,S1, "ATHE FIRMWARE HAS BEEN UPDATED"**

(Where A: Alphanumeric data)

**QR CODE 100,10,M,7,M,0,M1,S2, "N123456"**

(Where N: Numeric data)

**QR CODE 100,10,Q,7,M,0,M1,S3, "N123456!ATHE FIRMWARE HAS BEEN UPDATED"**

(Where N: Numeric data ; !:Transfer char ; A: Alphanumeric data)

**QRCODE 100,10,H,7,M,0,M1,S3, "B0012Product name"**

(where B: Binary data ; 0012: 12 bytes )

**QRCODE 100,10,M,7,M,0,M1,S3, "K"**

(Where K: Kanji data)

Auto mode example:

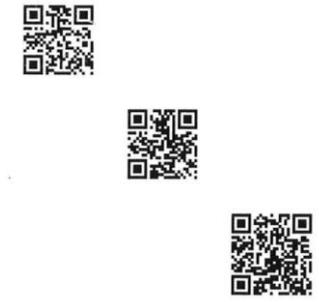
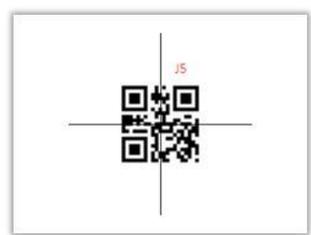
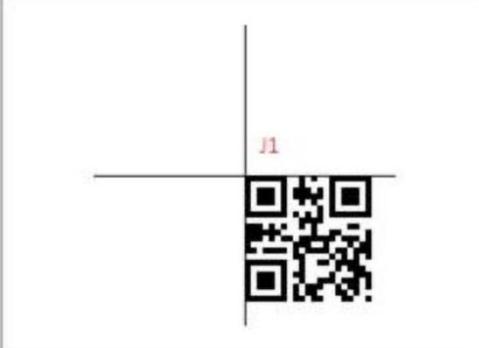
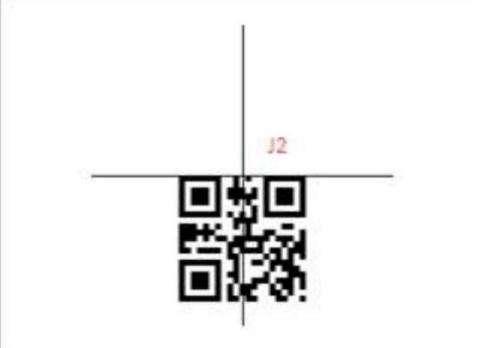
**QRCODE 100,10,M,7,A,0, "THE FIRMWARE HAS BEEN UPDATED"**

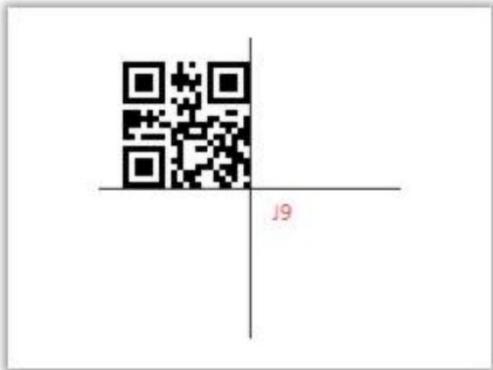
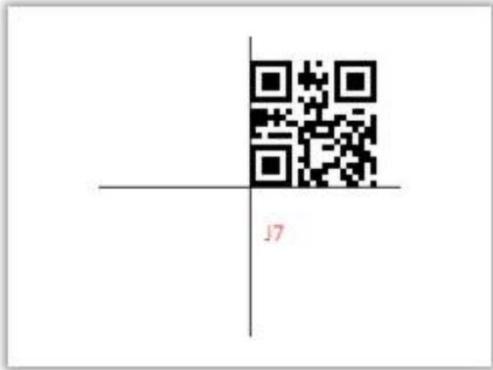
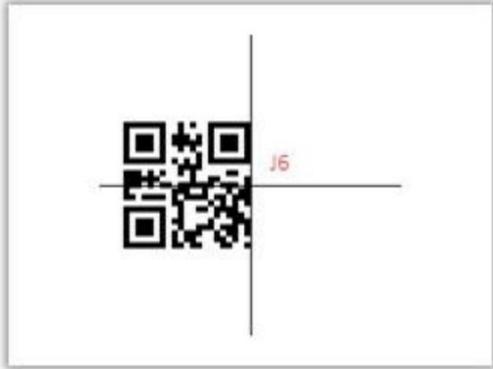
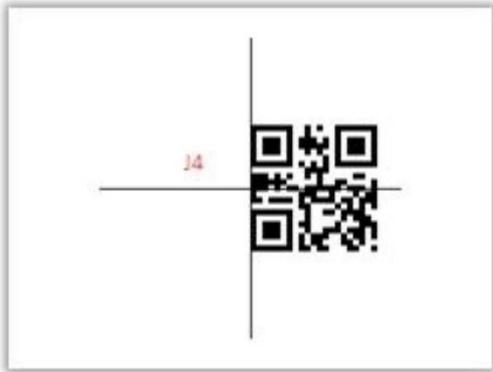
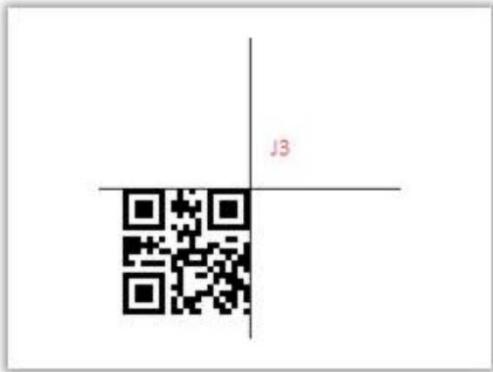
Note: TDP-643 Plus, TTP-243, TTP-342, TTP-244ME, TTP-342M and TTP-248M series are not supported this QRCODE command.

## Example

Sample code	Result
<p><b>Auto mode example</b></p> <p><u>General data string</u></p> <pre> SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,A,0, "ABCabc123" QRCODE 160,160,H,4,A,0, "123ABCabc" QRCODE 310,310,M,4,A,0,M2, "印表機 ABCabc123" PRINT 1,1 </pre>	
<p><u>Data string including &lt;Enter&gt; character (0Dh, 0Ah)</u></p> <pre> SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,A,0, « ABC&lt;Enter&gt; abc&lt;Enter&gt; 123 » QRCODE 160,160,H,4,A,0, « 123&lt;Enter&gt; ABC&lt;Enter&gt; abc » QRCODE 310,310,H,4,A,0, « 印表機&lt;Enter&gt; ABC&lt;Enter&gt; abc&lt;Enter&gt; 123 » PRINT 1,1 </pre>	
<p><u>Data string concatenation (Must be used with DOWNLOAD ... EOP command)</u></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><u>Data string including double quote (") character, please use \"[\" instead of</u></p> <pre> SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,A,0, "ABC\[\"abc\[\"123" </pre>	

<pre> 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>	
<b>Manual mode</b>	
<p><u>General data string</u></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><u>Data string including &lt;Enter&gt; character, &lt;Enter&gt; is an 8-bit byte data</u></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><u>Data string concatenation (Must be used with DOWNLOAD ... EOP command)</u></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>	

<p>Data string including double quote (") character, please use \" instead of</p> <p>SIZE 4,2.5  GAP 0,0  DIRECTION 1  CLS  QRCODE 10,10,H,4,M,0, "ABC!B0005\"abc\"!N123"  QRCODE 160,160,H,4,M,0, "N123!B0001\"!ABC!B0004\"abc"  QRCODE 310,310,H,4,M,0, "B0001\"!K 印表機!B0010\"ABCabc123"  PRINT 1,1</p>	
<p>Smart phone data string</p> <p>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.tscprinters.com"  PRINT 1,1  EOP  A</p>	
<p>Data string for parameter [justification] &amp; [area]</p> <p>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</p>	
<p>For other [justification] results (J1~J9)</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="175 1545 670 1915">  </div> <div data-bbox="710 1545 1204 1915">  </div> </div>	



Data string for parameter [length]

CLS  
QRCODE 50,462,H,5,M,0,M2,S5,L21,B00161234567890ABCDEF  
PRINT 1

# RMQR

## Description

This command prints rMQR code. This command has been supported since A2.15.111 and later firmware.

## Syntax

**RMQR x,y,ecc,size,type,rotate,"content"**

<u>Parameter</u>	<u>Description</u>
x	The upper left corner x-coordinate of the rMQR code
y	The upper left corner y-coordinate of the rMQR code
ecc	Error correction recovery level M: ~ 37% H: ~ 65%
size	Element module size
type	Barcode size. 0 = Automatic; 1-38 type  1: R7x43      11: R11x27      21: R13x99      31: R17x99 2: R7x59      12: R11x43      22: R13x139     32: R17x139 3: R7x77      13: R11x59      23: R15x43      33: R7xAutomatic 4: R7x99      14: R11x77      24: R15x59      34: R9xAutomatic 5: R7x139     15: R11x99      25: R15x77      35: R11xAutomatic 6: R9x43      16: R11x139     26: R15x99      36: R13xAutomatic 7: R9x59      17: R13x27      27: R15x139     37: R15xAutomatic 8: R9x77      18: R13x43      28: R17x43      38: R17xAutomatic 9: R9x99      19: R13x59      29: R17x59 10: R9x139     20: R13x77      30: R17x77
rotation	1 : 0 degree 90 : 90 degree 180 : 180 degree 270 : 270 degree
Content	The encodable character set is described as below, <u>Encodable character set:</u> 1) Numeric data: (digits 0~9) 2) Capital letters: (A-Z) 3) Small letters: (a-z) 4) Symbol: (~!@#\$%^&*()_+{}:<>?*/[,...etc)

## Example

Sample code	Result
<pre> CLS SIZE 4,3 GAP 0,0 DIRECTION 1 RMQR 250,50,H,6,0,0,"hi this is the rmqr" PRINT 1           </pre>	
<pre> CLS SIZE 4,3 GAP 0,0 DIRECTION 1 RMQR 250,50,H,6,0,0,"HI THIS IS THE RMQR" PRINT 1           </pre>	
<pre> SIZE 4,3 GAP 0,0 DIRECTION 1 CLS RMQR 400,400,H,9,0,0,"123456789123456789123456789" RMQR 400,400,H,9,0,90,"123456789123456789123456789" RMQR 400,400,H,9,0,180,"123456789123456789123456789" RMQR 400,400,H,9,0,270,"123456789123456789123456789" PRINT 1           </pre>	
<pre> CLS SIZE 4,3 GAP 0,0 DIRECTION 1 RMQR 250,50,H,6,0,0,"~!@#%\$%^&amp;*()_+{}:&lt;&gt;?*/[],." PRINT 1           </pre>	
<pre> CLS SIZE 4,3 GAP 0,0 DIRECTION 1 RMQR 250,50,H,6,0,0,"~Hi! This is the qMRQ!~" PRINT 1           </pre>	

CLS  
SIZE 4,3  
GAP 0,0  
DIRECTION 1  
RMQR  
250,50,H,6,0,0,"12345abcdeABCDE~!@#\$%12345ab  
cdeABCDE~!@#\$%"  
PRINT 1



# 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"`

<u>Parameter</u>	<u>Description</u>																								
x	X-coordinate																								
y	Y-coordinate																								
sym	Symbology type: <table border="1"><tbody><tr><td><b>RSS14</b></td><td>RSS14</td></tr><tr><td><b>RSS14T</b></td><td>RSS14 Truncated</td></tr><tr><td><b>RSS14S</b></td><td>RSS14 Stacked</td></tr><tr><td><b>RSS14SO</b></td><td>RSS14 Stacked Omnidirectional</td></tr><tr><td><b>RSSLIM</b></td><td>RSS Limited</td></tr><tr><td><b>RSSEXP</b></td><td>RSS Expanded</td></tr><tr><td><b>UPCA</b></td><td>UPC-A</td></tr><tr><td><b>UPCE</b></td><td>UPC-E</td></tr><tr><td><b>EAN13</b></td><td>EAN-13</td></tr><tr><td><b>EAN8</b></td><td>EAN-8</td></tr><tr><td><b>UCC128CCA</b></td><td>UCC/EAN-128 &amp; CC-A/B</td></tr><tr><td><b>UCC128CCC</b></td><td>UCC/EAN-128 &amp; CC-C</td></tr></tbody></table>	<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
<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)																								
pixMult	Module width in dot (1 to 10 valid)  The following barcode height is calculated by printer. <table border="1"><tbody><tr><td>RSS14</td><td>33 × pixMult</td></tr><tr><td>RSS14T</td><td>13 × pixMult.</td></tr><tr><td>RSS14S</td><td>13 × pixMult.</td></tr><tr><td>RSS14SO</td><td>33 × pixMult.</td></tr><tr><td>RSSLIM</td><td>13 × pixMult.</td></tr></tbody></table>	RSS14	33 × pixMult	RSS14T	13 × pixMult.	RSS14S	13 × pixMult.	RSS14SO	33 × pixMult.	RSSLIM	13 × pixMult.														
RSS14	33 × pixMult																								
RSS14T	13 × pixMult.																								
RSS14S	13 × pixMult.																								
RSS14SO	33 × pixMult.																								
RSSLIM	13 × pixMult.																								

RSSEXP	33 × pixMult.
EAN8	60 × pixMult.
EAN13	74 × pixMult.
UPCA	74 × pixMult.
UPCE	74 × pixMult.

sepHt                      Separator row height (1 and 2 valid)  
pixMult times sepHt is the real separator row height. It is calculated by printer.

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:

\*00abc0000hij = abhijc, where c = 0-2

\*00abc00000ij = abcij3

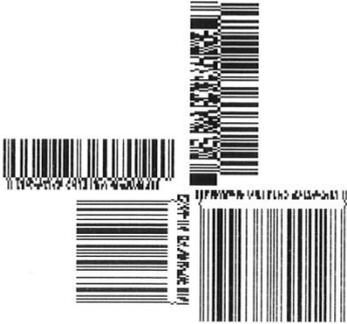
\*00abcd00000j = abcdj4

\*00abcde0000j = abcdej where j = 5-9

**Note:**

- **200 DPI: 1 mm = 8 dots**
- **300 DPI: 1 mm = 12 dots**
- **600 DPI : 1mm = 24 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 25% respectively.**
- **This command has been supported since V6.56 EZ and later firmware.**

## Example

Sample code	Result
<p>SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSS14",0,2,2, "1234567890 ABCDEFGH" RSS 300,300, "RSS14T",90,2,2, "1234567890 ABCDEFGH" RSS 300,300, "RSS14S",180,2,2, "1234567890 ABCDEFGH" RSS 300,300, "RSS14SO",270,2,2, "1234567890 ABCDEFGH" PRINT 1,1</p>	
<p>SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSSLIM",0,2,2, "1234567890 ABCDEFGH" RSS 300,300, "RSSEXP",90,2,2,22, "1234567890 ABCDEFGH" RSS 300,300, "UPCA",180,2,2, "1234567890 ABCDEFGH" RSS 300,300, "UPCE",270,2,2, "000 ABCDEFGH" PRINT 1,1</p>	
<p>SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "EAN13",0,2,2, "123456789012 ABCDEFGH" RSS 300,300, "EAN8",90,2,2, "1234567 ABCDEFGH" RSS 300,300, "UCC128CCA",180,2,2,25, "1234567890 ABCDEFGH" RSS 300,300, "UCC128CCC",270,2,2,25, "1234567890 ABCDEFGH" PRINT 1,1</p>	
<p>SIZE 100 mm, 100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,10, "RSSEXP",90,2,2,12, "81101061414112345628911012012120850100480002140256100"</p>	

<p>48000310123191000"</p> <p>PRINT 1</p>	
<p>Example of UPCE mode</p>	
<p>SIZE 4,1</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>REM UPCE Rule 1: <b>00abc0000hij</b> = abhijc, where c = 0-2</p> <p>RSS 10,10,"UPCE",0,2,2,"001200000456 ABCDEFGF"</p> <p>RSS 210,10,"UPCE",0,2,2,"001210000456 ABCDEFGF"</p> <p>RSS 410,10,"UPCE",0,2,2,"001220000456 ABCDEFGF"</p> <p>PRINT 1</p>	
<p>SIZE 4,1</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>REM UPCE Rule 2: <b>00abc00000ij</b> = abcij3</p> <p>RSS 10,10,"UPCE",0,2,2,"001230000045 ABCDEFGF"</p> <p>PRINT 1</p> <p>SIZE 4,1</p> <p>CLS</p> <p>REM UPCE Rule 3: <b>00abcd00000j</b> = abcdj4</p> <p>RSS 10,10,"UPCE",0,2,2,"001234000005 ABCDEFGF"</p> <p>PRINT 1</p>	
<p>SIZE 4,1</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p>	

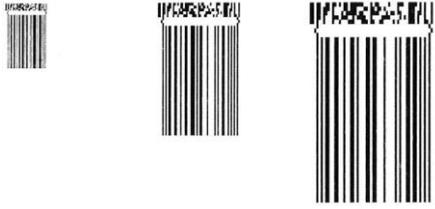
<p>REM UPCE Rule 4: <i>00abcde000j</i> = abcdej where j = 5-9</p> <p>RSS 10,10,"UPCE",0,2,2,"001234500005 ABCDEFGF"</p> <p>RSS 160,10,"UPCE",0,2,2,"001234500006 ABCDEFGF"</p> <p>RSS 310,10,"UPCE",0,2,2,"001234500007 ABCDEFGF"</p> <p>RSS 460,10,"UPCE",0,2,2,"001234500008 ABCDEFGF"</p> <p>RSS 610,10,"UPCE",0,2,2,"001234500009 ABCDEFGF"</p> <p>PRINT 1</p>	
--	--

**Example of barcode height of EAN8 EAN13 UPCA and UPCE.**

<p>SIZE 4,2</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>RSS 10,10,"EAN8",0,1,1,"1234567 ABCDEFGF"</p> <p>RSS 210,10,"EAN8",0,2,1,"1234567 ABCDEFGF"</p> <p>RSS 410,10,"EAN8",0,3,1,"1234567 ABCDEFGF"</p> <p>PRINT 1</p>	
--	---

<p>SIZE 4,2</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>RSS 10,10,"EAN13",0,1,1,"123456789012 ABCDEFGF"</p> <p>RSS 210,10,"EAN13",0,2,1,"123456789012 ABCDEFGF"</p> <p>RSS 410,10,"EAN13",0,3,1,"123456789012 ABCDEFGF"</p> <p>PRINT 1</p>	
--	--

<p>SIZE 4,2</p> <p>GAP 0,0</p> <p>DIRECTION 1</p>	
---	--

<p>CLS</p> <p>RSS 10,10,"UPCA",0,1,1,"12345678901 ABCDEFGH"</p> <p>RSS 210,10,"UPCA",0,2,1,"12345678901 ABCDEFGH"</p> <p>RSS 410,10,"UPCA",0,3,1,"12345678901 ABCDEFGH"</p> <p>PRINT 1</p>	
<p>SIZE 4,2</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>RSS 10,10,"UPCE",0,1,1,"001200000456 ABCDEFGH"</p> <p>RSS 210,10,"UPCE",0,2,1,"001210000456 ABCDEFGH"</p> <p>RSS 410,10,"UPCE",0,3,1,"001220000456 ABCDEFGH"</p> <p>PRINT 1</p>	
<p><b>Example of RSS GS1</b></p>	
<p>DIRECTION 1</p> <p>CLS</p> <p>CODEPAGE 850</p> <p>TEXT 62,240,"0",0,9,9,0,"(21)1234567891234(11)001225"</p> <p>RSS 62,170,"RSSEXP",0,2,1,22,"211234567891234#11001225"</p> <p>PRINT 1</p>	

# REVERSE

## Description

This command reverses a region in image buffer.

## Syntax

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

<u>Parameter</u>	<u>Description</u>
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*
- *600 DPI : 1mm = 24 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 25% 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

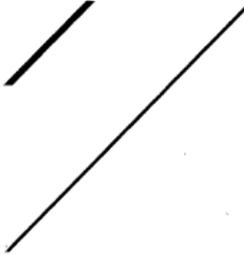
**DIAGONAL** x1, y1, x2, y2, thickness

<u>Parameter</u>	<u>Description</u>
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*
- *600 DPI : 1mm = 24 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 «

<u>Parameter</u>	<u>Description</u>																														
x	The x-coordinate of the text																														
y	The y-coordinate of the text																														
font	Font name <table border="1"><tbody><tr><td><b>0</b></td><td>Monotype CG Triumvirate Bold Condensed, font width and height is stretchable</td></tr><tr><td><b>1</b></td><td>8 x 12 fixed pitch dot font</td></tr><tr><td><b>2</b></td><td>12 x 20 fixed pitch dot font</td></tr><tr><td><b>3</b></td><td>16 x 24 fixed pitch dot font</td></tr><tr><td><b>4</b></td><td>24 x 32 fixed pitch dot font</td></tr><tr><td><b>5</b></td><td>32 x 48 dot fixed pitch font</td></tr><tr><td><b>6</b></td><td>14 x 19 dot fixed pitch font OCR-B</td></tr><tr><td><b>7</b></td><td>21 x 27 dot fixed pitch font OCR-B</td></tr><tr><td><b>8</b></td><td>14 x25 dot fixed pitch font OCR-A</td></tr><tr><td><b>ROMAN.TTF</b></td><td>Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.</td></tr><tr><td colspan="2"><b><i>Following fonts were supported since V6.80 EZ.</i></b></td></tr><tr><td><b>1.EFT</b></td><td>EPL2 font 1</td></tr><tr><td><b>2.EFT</b></td><td>EPL2 font 2</td></tr><tr><td><b>3.EFT</b></td><td>EPL2 font 3</td></tr><tr><td><b>4.EFT</b></td><td>EPL2 font 4</td></tr></tbody></table>	<b>0</b>	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable	<b>1</b>	8 x 12 fixed pitch dot font	<b>2</b>	12 x 20 fixed pitch dot font	<b>3</b>	16 x 24 fixed pitch dot font	<b>4</b>	24 x 32 fixed pitch dot font	<b>5</b>	32 x 48 dot fixed pitch font	<b>6</b>	14 x 19 dot fixed pitch font OCR-B	<b>7</b>	21 x 27 dot fixed pitch font OCR-B	<b>8</b>	14 x25 dot fixed pitch font OCR-A	<b>ROMAN.TTF</b>	Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.	<b><i>Following fonts were supported since V6.80 EZ.</i></b>		<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>0</b>	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable																														
<b>1</b>	8 x 12 fixed pitch dot font																														
<b>2</b>	12 x 20 fixed pitch dot font																														
<b>3</b>	16 x 24 fixed pitch dot font																														
<b>4</b>	24 x 32 fixed pitch dot font																														
<b>5</b>	32 x 48 dot fixed pitch font																														
<b>6</b>	14 x 19 dot fixed pitch font OCR-B																														
<b>7</b>	21 x 27 dot fixed pitch font OCR-B																														
<b>8</b>	14 x25 dot fixed pitch font OCR-A																														
<b>ROMAN.TTF</b>	Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.																														
<b><i>Following fonts were supported since V6.80 EZ.</i></b>																															
<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																														
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.
alignment	For *.TTF font, x-multiplication and y-multiplication support floating value. (V6.91 EZ)  Optional. Specify the alignment of text. (V6.73 EZ)  0 : Default (Left)  1 : Left  2 : Center  3 : Right
content	Content of text string

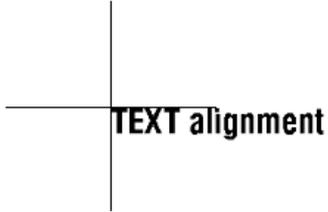
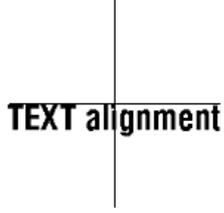
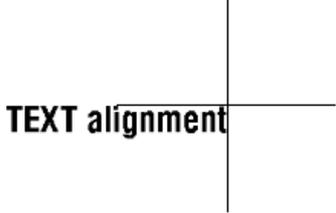
**Note:**

- *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.*
- *Please refer to printer model list for checking TSPL or TSPL2.*
- *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	Font Type									
	0	1	2	3	4	5	6	7	8	ROMAN.TTF
TSPL language printers		V	V	V	V	V				
TSPL2 language printers	V	V	V	V	V	V	V	V	V	V
TTP-248M printer		V	V	V	V	V	V	V		V

## Example

Sample Code	Result
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"	<div style="display: flex; justify-content: space-between;"> <div style="text-align: left;"> <p><b>TSPL 2</b> align left</p> <p>FONT 0 FONT 1 FONT 2 FONT 3 FONT 4 <b>FONT 5</b> FONT 6 FONT 7 FONT 8 FONT ROMAN.TTF</p> </div> <div style="text-align: center;"> <p><b>EPL 2</b> align center</p> <p>FONT 1 FONT 2 FONT 3 FONT 4 <b>FONT 5</b> FONT 6 FONT 7 FONT 8</p> </div> <div style="text-align: right;"> <p><b>ZPL 2</b> align right</p> <p>FONT A FONT B FONT D FONT E8 FONT F <b>FONT G</b> FONT H8 © © TM UL SR ©</p> </div> </div>

Sample Code	Result
<pre>TEXT 800,410,"H8.FNT",0,1,1,3,"FONT H8" TEXT 800,460,"GS.FNT",0,1,1,3,"ABCDEF" PRINT 1</pre>	
<pre>SIZE 4,2 GAP 0,0 DIRECTION 1 CLS BAR 60,120,200,1 BAR 160,20,1,200 TEXT 160,120,"0",0,12,12,1,"TEXT alignment" PRINT 1,1</pre>	
<pre>SIZE 4,2 GAP 0,0 DIRECTION 1 CLS BAR 60,120,200,1 BAR 160,20,1,200 TEXT 160,120,"0",0,12,12,2,"TEXT alignment" PRINT 1,1</pre>	
<pre>SIZE 4,2 GAP 0,0 DIRECTION 1 CLS BAR 160,120,200,1 BAR 260,20,1,200 TEXT 260,120,"0",0,12,12,3,"TEXT alignment" PRINT 1,1</pre>	

# BLOCK

## Description

This command prints paragraph on label.

## Syntax

**BLOCK** *x,y,width,height, « font »,rotation,x-multiplication,y-multiplication,[space,]align,[fit,] »content »*

<u>Parameter</u>	<u>Description</u>
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
<b>0</b>	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable
<b>1</b>	8 x 12 fixed pitch dot font
<b>2</b>	12 x 20 fixed pitch dot font
<b>3</b>	16 x 24 fixed pitch dot font
<b>4</b>	24 x 32 fixed pitch dot font
<b>5</b>	32 x 48 dot fixed pitch font
<b>6</b>	14 x 19 dot fixed pitch font OCR-B
<b>7</b>	21 x 27 dot fixed pitch font OCR-B
<b>8</b>	14 x25 dot fixed pitch font OCR-A
<b>ROMAN.TTF</b>	Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.
<b><i>Following fonts were supported since V6.80 EZ.</i></b>	
<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

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)

150 : Left

150 : Center

3 : Right

[fit]                         Shrink the text so that it fits in the block (VA1.97)

0 : No shrink (default)

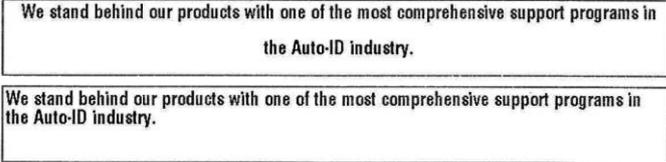
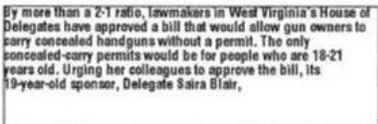
1 : Shrink

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
<p>SIZE 4,0.5</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p> <p>BOX 10,10,800,100,2</p> <p>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."</p> <p>PRINT 1</p> <p>CLS</p> <p>BOX 10,10,800,100,2</p> <p>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."</p> <p>PRINT 1</p>	
Sample Code for [fit] Parameter	Result
<p>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, "</p> <p>SIZE 4,1.5</p> <p>GAP 0,0</p> <p>DIRECTION 1</p> <p>CLS</p>	<p>0 : No shrink (default)</p>  <p>1 : Shrink</p> 

**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**

# 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

~!E

## <ESC>!O

### Description

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

### Syntax

<ESC>!O

**Note:**

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

### See Also

<ESC>!P

## <ESC>!P

### Description

This command is used 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	Reserved

0	1	0	0	1	0	0	0	48	72	H	Receive buffer full (RS-232)
0	1	1	0	0	0	0	0	60	96	`	Unhealthy Dots
<b>Status Byte #3: error</b>											
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	80	P	Insufficient memory
<b>Status Byte #4: error</b>											
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	1	0	0	0	48	72	H	Ribbon jam
0	1	1	0	0	0	0	0	60	96	`	Print head open

# Example

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

The values return from printer in Hex.

The values return from printer in Character.

Hex Data 1B 21 53 means <ESC>!S.

Click to send Hex 1b 21 53 to query printer status.

## Result

Item	Meaning								
1	The start character of returned value.								
2	The 4-byte status in Hex.								
3	The end characters of returned value.								
4	<p>4-byte status in characters.</p> <p>@@@@: The printer is normal for use.</p> <p>F@@@: The printer is feeding label.</p> <p>\@@@: Printer is in PAUSE mode.</p> <p>B@@@: The printer is backing label.</p> <p>E@@B: Printer is in error "Paper Jam".</p> <p>E@@b: Printer is in error "Paper Jam" &amp; "Head open".</p> <p>Note: Paper Jam &lt;Hex 42&gt;   Head open &lt;Hex 60&gt;</p> <p>0x42   0x60 = 62 &lt;Hex b&gt;</p> <table border="1"> <tr> <td>E@@a</td> <td>Paper empty + Print head open</td> </tr> <tr> <td>E@@b</td> <td>Paper jam + Print head open</td> </tr> <tr> <td>E@@d</td> <td>Ribbon empty + Print head open</td> </tr> <tr> <td>E@@h</td> <td>Ribbon jam+ Print head open</td> </tr> </table>	E@@a	Paper empty + Print head open	E@@b	Paper jam + Print head open	E@@d	Ribbon empty + Print head open	E@@h	Ribbon jam+ Print head open
E@@a	Paper empty + Print head open								
E@@b	Paper jam + Print head open								
E@@d	Ribbon empty + Print head open								
E@@h	Ribbon jam+ Print head open								

1 2 3

4

## See Also

<ESC>!?

## <ESC>!F

### Description

This command is used 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

<ESC>!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

~!!

## Description

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

## Syntax

~!!

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

~!!

## 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

# Message Translation Protocols

~#

## Description

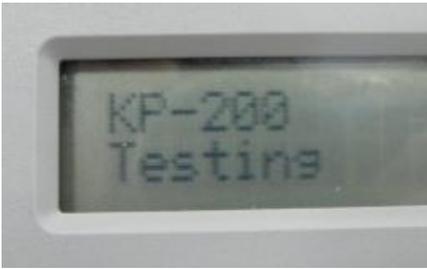
The beginning identifier (~#) of the prompt message is sent from the printer to the KP-200 portable keyboard. The ending identifier is ~&. @0 following the ending identifier ~& is used to instruct keyboard to display the prompt in the first line of LCD display. @1 following the ending identifier ~& is 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
<pre>DOWNLOAD "A.BAS" OUT "~#KP-200~&amp;@0" OUT "~#Testing~&amp;@1" EOP A</pre>	

## See Also

INPUT, OUT

# Commands for Windows Driver

## !B

### Description

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

### Syntax

**!Bnnn**

<u>Parameter</u>	<u>Description</u>
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**

<u>Parameter</u>	<u>Description</u>
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**

<u>Parameter</u>	<u>Description</u>
nnn	Specifies the number of copies to be printed.

## Example

**!N001**

# 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

Please refer to [printer model list](#) for checking TSPL or TSPL2.

***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
<b><i><u>For one button desktop printer series</u></i></b>  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:  <b>Orange → red (5 blinks) → orange (5 blinks) → green (5 blinks) → green and orange (5 blinks) → red and orange (5 blinks) → solid green</b>
<b><i><u>For three buttons industrial printer series</u></i></b>  Hold the FEED key and power on the switch. The ERROR LED will be on. Printer is now ready to use.
<b><i><u>For six or two buttons industrial printer series</u></i></b>  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

### 3. Download a program file:

**DOWNLOAD [n,] "FILENAME.BAS"**

<u>Parameter</u>	<u>Description</u>
n	Specify memory used to save downloaded files.  <b><i>N is ignored:</i></b> 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.  <b><i>F:</i></b> Download files to main board flash memory.  <b><i>E:</i></b> Download files to expansion memory module.
FILENAME.BAS	The filename resident in printer memory.
<b>Note:</b>	
<ul style="list-style-type: none"><li>• <i>Filenames are case sensitive.</i></li><li>• <i>File extensions must be ".BAS"</i></li><li>• <i>Filenames must be in 8.3 format.</i></li><li>• <i>It should use with EOP command.</i></li><li>• <i>If memory is not specified, all files will be downloaded to DRAM.</i></li><li>• <i>The priority of AUTO.BAS in each memory device:</i><ul style="list-style-type: none"><li><i>A. DRAM &gt; FLASH &gt; CARD (Ext. FLASH) if firmware is before V6.80EZ.</i></li><li><i>B. DRAM &gt; CARD (Ext. FLASH) &gt; FLASH if firmware is after V6.80EZ (include).</i></li></ul></li><li>• <i>No Battery is used to back up files in DRAM. Which will be lost in the event printer power is lost.</i></li></ul>	

### 4. Download a data file:

**DOWNLOAD [n,] "FILENAME",DATA SIZE,DATA CONTENT...**

<u>Parameter</u>	<u>Description</u>
n	Specify the memory location to save downloaded files.  <b><i>N is ignored:</i></b> 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.  <b><i>F:</i></b> Download files to main board flash memory.  <b><i>E:</i></b> 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
```

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

```
DOWNLOAD F, "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"**

<u>Parameter</u>	<u>Description</u>
n	Specify the memory location that files will be deleted.  <b>N is ignored:</b> Kill files saved in DRAM.  <b>F:</b> Kill files from main board flash memory.  <b>E:</b> 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:*

<b>KILL "FILENAME"</b>	Delete the specify file in DRAM.
<b>KILL "*.PCX"</b>	Delete all PCX files in DRAM.
<b>KILL "*"</b>	Delete all files in DRAM.
<b>KILL F, "FILENAME"</b>	Delete the specify file in FLASH.
<b>KILL E, "*.PCX "</b>	Delete all PCX file in extension memory card.

- *For TSPL printers, please send MOVE command to printer after sending KILL command.*
- *Please refer to [printer model list](#) for checking TSPL or TSPL2.*

Model	Support		
	<b>KILL "*"</b>	<b>KILL "*"   MOVE</b>	<b>KILL F, "*"</b>
TSPL programming printer	<b>V</b>	<b>V</b>	
TSPL2 programming printer	<b>V</b>		<b>V</b>

## 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".*

*\* TDP-643 Plus, TTP-243, TTP-342, TTP-244ME, TTP-342M and TTP-248M series are not supported this feature*

## 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>DEMO PROGRAM</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</pre>	

DEMO	
------	--

## See Also

DOWNLOAD, EOP

# 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
<pre>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"</pre>	<pre>100 50.98 99.99</pre>

### 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>	65

## See Also

DOWNLOAD, EOP, STR\$( )

# CHR\$( )

## Description

This function returns the character with the specified ASCII code.

## Syntax

**CHR\$(n)**

<u>Parameter</u>	<u>Description</u>
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$)`

<u>Parameter</u>	<u>Description</u>
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>	<pre>Encoded data: pppp Decoded data: 1234</pre>

# END

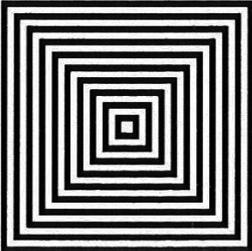
## Description

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)

<u>Parameter</u>	<u>Description</u>
File handle	Either 0 or 1
<u>Return value</u>	<u>Description</u>
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</pre>	<pre>*****EOF TEST*****  COMPUTER\$2000[EOF(0)=1]  PRICE-2000</pre>

<pre>IF EOF(0)=0 THEN GOTO A</pre>	
------------------------------------	--

```
PRINT 1
```

```
EOP
```

```
DEMO
```

## 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**

<u>Parameter</u>	<u>Description</u>								
[memory ID]	Optional. Open the file in specific memory device. <i>*Since V6.68 EZ.</i> <table border="1"><thead><tr><th>ID</th><th>Memory device</th></tr></thead><tbody><tr><td>Omitted</td><td>DRAM</td></tr><tr><td>F</td><td>FLASH</td></tr><tr><td>E</td><td>CARD</td></tr></tbody></table>	ID	Memory device	Omitted	DRAM	F	FLASH	E	CARD
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."</pre>	<pre>Open file in DRAM. Open file in FLASH. Auto create a new file in FLASH.</pre>

<pre>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>	
--	--

## See Also

DOWNLOAD, EOP, READ, WRITE, SEEK, CLOSE

# CLOSE

## Description

Close the file handle which is open by command OPEN.

## Syntax

**CLOSE file handle**

<u>Parameter</u>	<u>Description</u>
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

<u>Parameter</u>	<u>Description</u>
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

<u>Parameter</u>	<u>Description</u>
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)</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>

<pre>Y=Y+100 PRINT 1 Y=100 IF I&lt;=1 THEN IF EOF(I)=1 THEN I=I+1 GOTO Start ELSE GOTO Start ENDIF ELSE END ENDIF EOP DEMO</pre>	
--	--

## 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**

<u>Parameter</u>	<u>Description</u>
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</pre>	<pre>SEEK COMMAND TEST SHIFT 4 CHARACTERS 567890 SHIFT 0 CHARACTERS 1234567890</pre>

<b>EOP</b> <b>TEST</b>	
---------------------------	--

## See Also

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

# LOF( )

## Description

This function returns the size of the specified file.

## Syntax

LOF("FILENAME")

<u>Parameter</u>	<u>Description</u>
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>	<pre>LOF( ) FUNCTION TEST DATA1 IS: 10 Bytes DATA2 IS: 15 Bytes</pre>

## See Also

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

# LOC( )

## Description

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

## Syntax

LOC(file handle)

<u>Parameter</u>	<u>Description</u>
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)**

<u>Parameter</u>	<u>Description</u>
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$][, ...]
```

<u>Parameter</u>	<u>Description</u>
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</pre>	<pre>Original data in DATA1: 1234567890 New data in Data1: 1234567890aB1</pre>

<pre>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>	
--	--

## See Also

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

# 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$][, ...]
```

<u>Parameter</u>	<u>Description</u>
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	Result
<pre>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</pre>	<pre>The first 3 characters in DATA1: 12 (51) The first 5 characters in DATA1: 12345</pre>

<pre>GAP 0,0 CLS TEXT 10,10,"3",0,1,1,"The first 3 characters in DATA1: "+a\$b\$b\$+" ("+STR\$(c)+")" TEXT 10,60,"3",0,1,1,"The first 5 characters in DATA1: "+e\$ PRINT 1 EOP TEST</pre>	
---	--

## See Also

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

# COPY

## Description

Copy the existed file from CARD/ USB drive to FLASH.

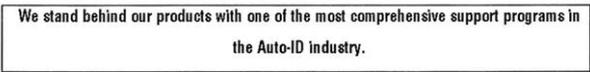
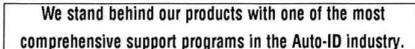
## Syntax

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

<u>Parameter</u>	<u>Description</u>										
memory ID of source	Optional. <table border="1"><thead><tr><th>ID</th><th>Memory device</th></tr></thead><tbody><tr><td>Omitted</td><td>DRAM</td></tr><tr><td>F</td><td>FLASH</td></tr><tr><td>E</td><td>CARD</td></tr><tr><td>U</td><td>USB drive</td></tr></tbody></table>	ID	Memory device	Omitted	DRAM	F	FLASH	E	CARD	U	USB drive
ID	Memory device										
Omitted	DRAM										
F	FLASH										
E	CARD										
U	USB drive										
source filename	The file in CARD which you want to copy to on board FLASH.										
Memory ID of new file	Optional. <table border="1"><thead><tr><th>ID</th><th>Memory device</th></tr></thead><tbody><tr><td>Omitted</td><td>DRAM</td></tr><tr><td>F</td><td>FLASH</td></tr></tbody></table>	ID	Memory device	Omitted	DRAM	F	FLASH				
ID	Memory device										
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	Result
<pre>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</pre>	203 dpi  300 dpi 

<code>GAP 0,0</code> <code>CLS</code> <code>BOX 10,10,800,100,2</code> <code>BLOCK 15,15,790,90,"0",0,8,8,20,2,data\$</code> <code>PRINT 1</code> <code>EOP</code> <code>TEST</code>	
--	--

## 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**

<u>Parameter</u>	<u>Description</u>
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	Result
<pre>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)</pre>	<pre>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</pre>

```
NEXT
```

```
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
```

## 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**

<u>Parameter</u>	<u>Description</u>
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.
<i>Note:</i>  <i>This command has been supported since V5.10 EZ and later firmware.</i>	

## Example

Sample Code	Result
<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>	$1+2+3+ \dots + 100 = 5050$
<pre>DOWNLOAD "TEST.BAS"</pre>	

```
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
```

<u>Parameter</u>	<u>Description</u>
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
<i>Note:</i>	
<i>This command has been supported since V5.10 EZ and later firmware.</i>	

## 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 </pre>	$1+2+3+ \dots + 100 = 5050$

Sample Code	Result
<pre> 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>	
<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" </pre>	

Sample Code	Result
<pre> 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>	$1+2+3+ \dots + 100 = 5050$
<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>	$1+2+3+ \dots + 100 = 5050$

## 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.

Note: TDP-643 Plus, TTP-243, TTP-342, TTP-244ME and TTP-342M series are not supported multiple-line form.

### 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

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

<u>Parameter</u>	<u>Description</u>
condition	Available relational operator: <, >, =, <=, >=, <>  <i>*Relational operator &lt;&gt;, not equal, was supported since V5.10 EZ.</i>
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</pre>	<pre>(1) 1+2+3+...+100=5050 (2) 1+3+5+...+99=2500 (3) 2+4+6+...+100=2550 (4) 3+6+9+...+99=1683 (5) 5+10+15+...+100=1050 (1)-(5)=4000 (6) 7+14+21+...+98=735 (7) 17+34+51+...+85=255 (8) 27+54+...+81=162 (9) 37+74=111 (1)-(6)-(7)-(8)-(9)=3787</pre>

<pre> 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 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 DEMO </pre>	
<pre> DOWNLOAD F, "TEST.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 CLS A=85 B=10 :START IF A&lt;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 </pre>	<pre> 105 IS LAGER THEN 100  95 IS SMALLER THEN 100  85 IS SMALLER THEN 100 </pre>

<pre>:L2 CLS TEXT 100,10,"3",0,1,1,STR\$(A) + "IS LAGER THEN 100" PRINT 1 EOP TEST</pre>	
--	--

**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.

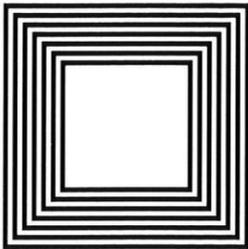
## Syntax

```
GOSUB LABEL
    statement
END

:LABEL
    statement
RETURN
```

<u>Parameter</u>	<u>Description</u>
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</pre>	<p>GOSUB &amp; RETURN COMMAND TEST</p> 

<b>RETURN</b>	
---------------	--

<b>EOP</b>	
------------	--

<b>GOSUB1</b>	
---------------	--

## 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**

<u>Parameter</u>	<u>Description</u>
LABEL	Beginning of the point. The maximum length of the label is 8 characters.

## Example

Sample code	Result
<pre>DOWNLOAD "GOTO1.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)</pre>	<pre>THE SUMMATION OF 1..100 IS 5050</pre>

```
TEXT 10,100, "3",0,1,1,B$
```

```
PRINT 1
```

```
END
```

```
EOP
```

```
GOTO1
```

## See Also

DOWNLOAD, EOP, END, GOSUB...RETURN

# INP\$( )

## Description

One byte is received from communication port.

## Syntax

**INP\$(n)**

<u>Parameter</u>	<u>Description</u>
N	1 : com1 port in printer

## Example

Sample code	Result
<pre>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</pre>	<p>The received data is: 12345</p>

## See Also

INP()

# INP( )

## Description

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

## Syntax

INP(n)

<u>Parameter</u>	<u>Description</u>
n	1 : com1 port in printer

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

## Example

Sample code	Result
<pre>DOWNLOAD "TEST.BAS"  sci=0 str\$=""  FOR I=1 TO 5 sci=INP(1) str\$=str\$+" " +STR\$(sci) OUT sci 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</pre>	<pre>The received data is: 49 50 51 52 53</pre>

--	--

## 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
<pre>DOWNLOAD "TEST.BAS" DATA\$="" WHILE LOB(&lt;&gt;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.</pre>
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**

<u>Parameter</u>	<u>Description</u>
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 F, "TEST.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</pre>	<pre>INPUT and KP-200 Test 123456 1 2 3 4 5 6 7 8 9 0 1 2 8</pre>

<b>TEST</b> <b>123456</b> <b>123456789012</b>	
---	--

## See Also

DOWNLOAD, EOP, END, GOTO

# PREINPUT

## Description

This command can define the start character for command INPUT.

## Syntax

`PREINPUT var$`

`PREINPUT CHR$(n)`

<u>Parameter</u>	<u>Description</u>
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)`

<u>Parameter</u>	<u>Description</u>
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

<u>Parameter</u>	<u>Description</u>
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</pre>	<pre>DATA = 9012 DATA = 5678 DATA = 1234</pre>

<=1234=><=5678=><=9012=>	
--------------------------	--

## See also

PREINPUT, POSTINPUT, INPUT

# REM

## Description

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

## Syntax

REM

## Example

Sample code	Result
<pre>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</pre>	<pre>REMARK DEMO PROGRAM</pre>

## See Also

DOWNLOAD, EOP, END

# OUT

## Description

This command returns data through the specific port.

## Syntax

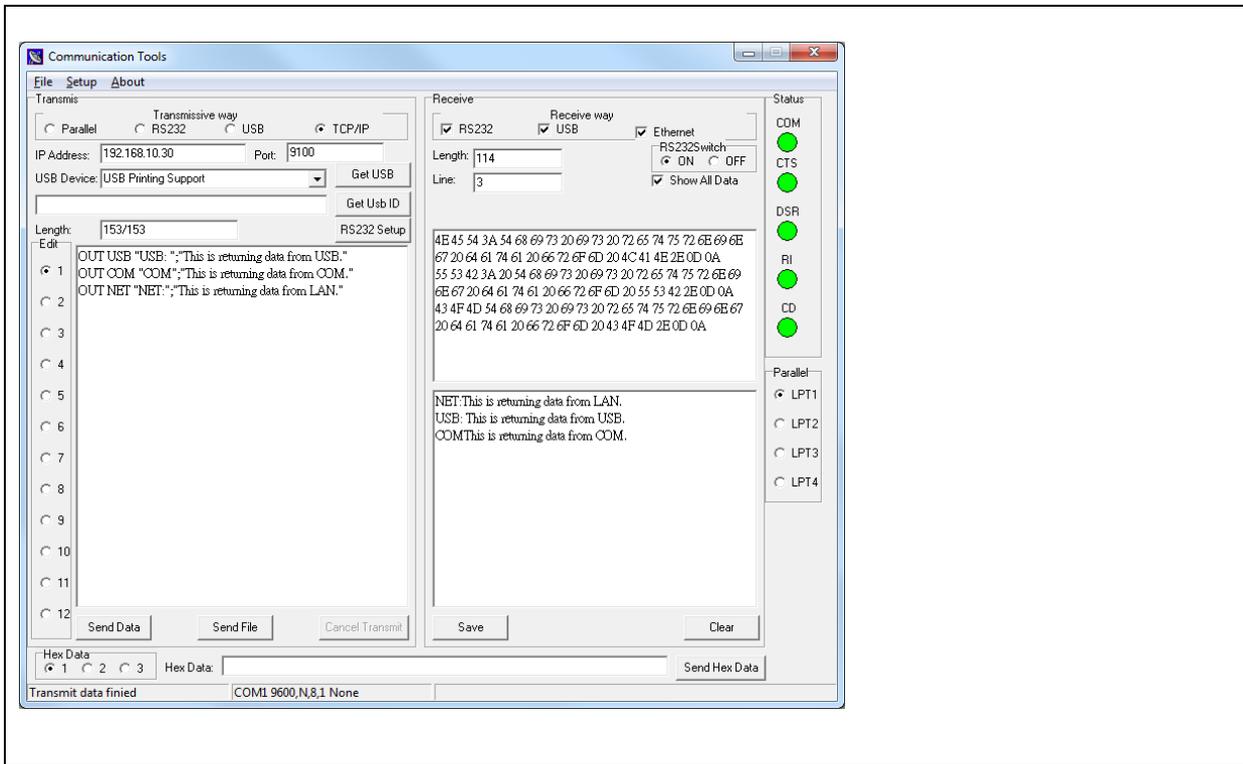
**OUT [port] "prompt",variable**

**OUT [port] "prompt";variable**

<u>Parameter</u>	<u>Description</u>
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 " <i>prompt</i> " and " <i>variable</i> " are separated by <0x0D><0x0A>.
;	The " <i>variable</i> " comes behind " <i>prompt</i> " directly.
<b>Note:</b> <i>This command has been supported since V6.93 EZ and later firmware.</i>	

## Example

Sample Code
<pre>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. "</pre>
Result



# OUTR

## Description

This command sends data through RS-232 port only.

## Syntax

**OUTR "prompt",variable**

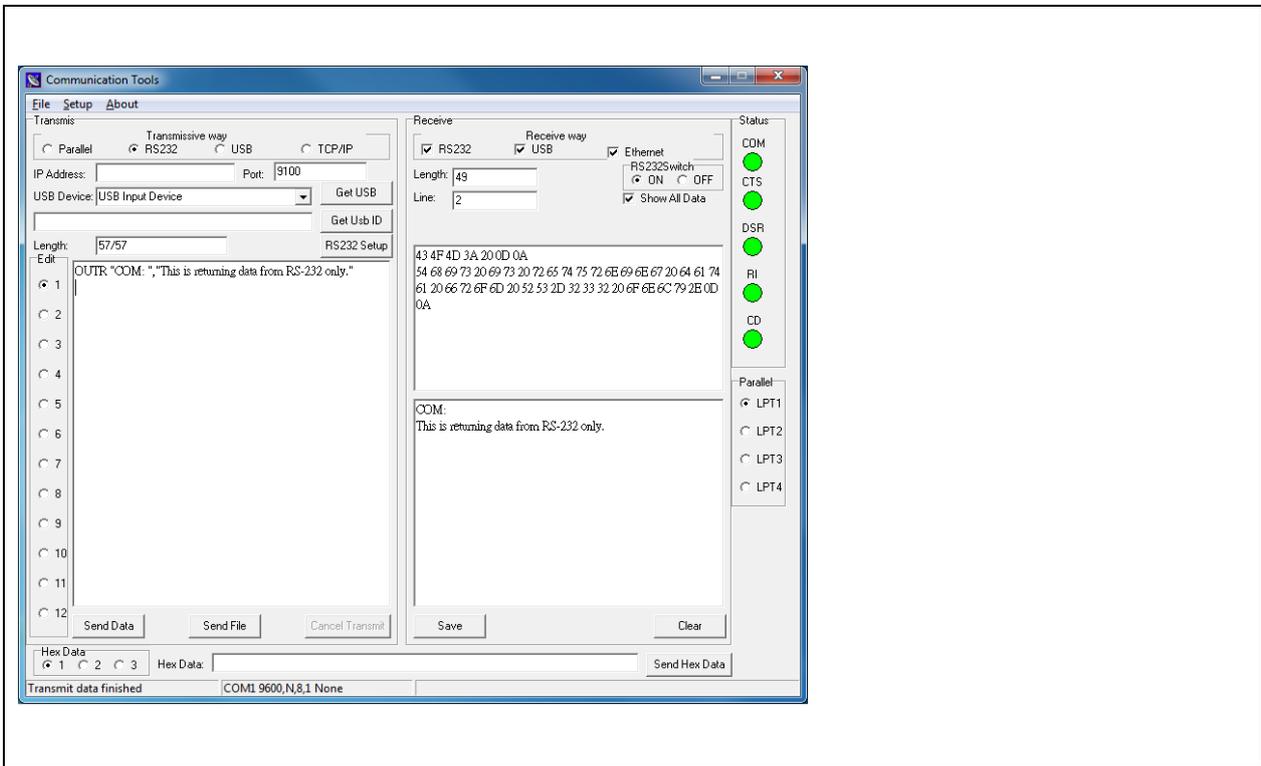
**OUTR "prompt";variable**

<u>Parameter</u>	<u>Description</u>
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
<pre>OUTR "COM: "," This is returning data from RS-232 only."</pre>
Result



# 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

Note: Desktop printers do not have the PAUSE key except TTP-243/244 series printers.

## 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)

<u>Parameter</u>	<u>Description</u>
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 DEMO 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)**

<u>Parameter</u>	<u>Description</u>
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)

<u>Parameter</u>	<u>Description</u>
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\$="ABCDEFGHIJKLMNOPQRSTUVWXYZ" 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>ABCDEFGHIJKLMNOPQRSTUVWXYZ 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 *m*th character of a string.

### Syntax

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

<u>Parameter</u>	<u>Description</u>
string	The string to be processed
m	The beginning of <i>m</i> <sup>th</sup> 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\$="ABCDEFGHIJKLMNOPQRSTUVWXYZ" E\$=MID\$(A\$,11,10) CLS TEXT 10,10, "3",0,1,1,A\$ TEXT 10,40, "3",0,1,1,"10 MIDDLE CHARS: "+E\$ PRINT 1 EOP DEMO</pre>	<pre>ABCDEFGHIJKLMNOPQRSTUVWXYZ 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)**

<u>Parameter</u>	<u>Description</u>
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\$="ABCDEFGHIJKLMNOPQRSTUVWXYZ" 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>ABCDEFGHIJKLMNOPQRSTUVWXYZ 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)

<u>Parameter</u>	<u>Description</u>
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\$="ABCDEFGHIJKLMN OPQRSTUVWXYZ" 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>ABCDEFGHIJKLMN OPQRSTUVWXYZ 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])**

<u>Parameter</u>	<u>Description</u>
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 <b>case-insensitive</b> .

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\$)**

<u>Parameter</u>	<u>Description</u>
start	Optional. Numeric expression that sets the starting position for each search. If omitted, search begins at the first character position. The start index is 1 – based.
Str1\$	Required. The string expression to be searched.
Str2\$	Required. The string expression to search for.

*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\$)**

<u>Parameter</u>	<u>Description</u>
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

LTRIM\$, RTRIM\$()

# LTRIM\$( )

## Description

Removes leading blank space from a string.

## Syntax

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

<u>Parameter</u>	<u>Description</u>
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\$(), RTRIM\$()

# RTRIM\$( )

## Description

Removes trailing blank space from a string.

## Syntax

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

<u>Parameter</u>	<u>Description</u>
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$(\"[\" +data3$+\" \", \")a\"[\"])  = \" +LTRIM$(data3$,\"a\")
TEXT 50,230,\"3\",0,1,1, \"TRIM$(\"[\" +data3$+\" \", \")a\"[\"])  = \" +TRIM$(data3$,\"a\")
TEXT 50,260,\"3\",0,1,1, \"RTRIM$(\"[\" +data3$+\" \", \")a\"[\"])  = \" +RTRIM$(data3$,\"a\")

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)**

<u>Parameter</u>	<u>Description</u>
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\$="ABCDEFGG" 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</pre>	

TEST	
------	--

## See Also

TEXT, BARCODEPIXEL()

# BARCODEPIXEL( )

## Description

Returns the width of barcode in dot.

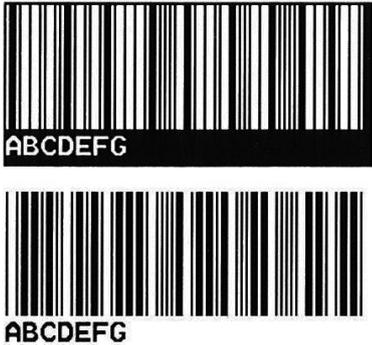
## Syntax

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

<u>Parameter</u>	<u>Description</u>
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\$="ABCDEFGG" 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\$</pre>	

<pre>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")

<u>Parameter</u>	<u>Description</u>
numeric character	" 0~9", "."

## Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHJKLMNOPQRSTUVWXYZ" 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> ABCDEFGHI JKLMNOPQRSTUVWXYZ 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	Result
<pre>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</pre>	<pre>Now is 1/9/2013 2:19:27 PM Tuesday, January 09 2013</pre>

## 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
<pre>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</pre>
Result
<pre>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</pre>

## 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\$])**

<u>Parameter</u>	<u>Description</u>
expression	Required. Any valid expression.
Style\$	Optional. A valid named or user-defined format string expression.

<b>Predefined date/time formats</b>	<b>Description</b>
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.

<b>User-defined date/time formats</b>	<b>Description</b>
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.
Dddddd	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.

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.

Number formats (since A1.97)	Description
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.

User-defined number formats (since A1.97)	Description
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
SIZE 800 dot,1900 dot	General Date:1/9/2013 2:46:18 PM
GAP 0,0	Long Date:Tuesday, January 09 2013
DIRECTION 1	Medium Date:09-Jan-13
CLS	Short Date:1/9/2013
TEXT 15,10, "3",0,1,1, "General Date: "+FORMAT\$(NOW,"General Date")	Long Time:2:46:18 PM
TEXT 15,60, "3",0,1,1, "Long Date: "+FORMAT\$(NOW,"Long Date")	Medium Time:02:46 PM
TEXT 15,110, "3",0,1,1, "Medium Date: "+FORMAT\$(NOW,"Medium Date")	Short Time:14:46
TEXT 15,160, "3",0,1,1, "Short Date: "+FORMAT\$(NOW,"Short Date")	c:1/9/2013 2:46:18 PM
TEXT 15,210, "3",0,1,1, "Long Time: "+FORMAT\$(NOW,"Long Time")	d:9
TEXT 15,260, "3",0,1,1, "Medium Time: "+FORMAT\$(NOW,"Medium Time")	dd:09
TEXT 15,310, "3",0,1,1, "Short Time: "+FORMAT\$(NOW,"Short Time")	ddd:Tue
TEXT 15,360, "3",0,1,1, "c: "+FORMAT\$(NOW,"c")	dddd:Tuesday
TEXT 15,410, "3",0,1,1, "d: "+FORMAT\$(NOW,"d")	dddd:1/9/2013
TEXT 15,460, "3",0,1,1, "dd: "+FORMAT\$(NOW,"dd")	dddddd:Tuesday, January 09 2013
TEXT 15,510, "3",0,1,1, "ddd: "+FORMAT\$(NOW,"ddd")	w:3
TEXT 15,560, "3",0,1,1, "dddd: "+FORMAT\$(NOW,"dddd")	ww:2
TEXT 15,610, "3",0,1,1, "dddd: "+FORMAT\$(NOW,"dddd")	m:1
TEXT 15,660, "3",0,1,1, "dddddd: "+FORMAT\$(NOW,"dddddd")	mm:01
TEXT 15,710, "3",0,1,1, "w: "+FORMAT\$(NOW,"w")	mmm:Jan
TEXT 15,760, "3",0,1,1, "ww: "+FORMAT\$(NOW,"ww")	mmmm:January
TEXT 15,810, "3",0,1,1, "m: "+FORMAT\$(NOW,"m")	q:1
TEXT 15,860, "3",0,1,1, "mm: "+FORMAT\$(NOW,"mm")	y:9
TEXT 15,910, "3",0,1,1, "mmm: "+FORMAT\$(NOW,"mmm")	yy:13
TEXT 15,960, "3",0,1,1, "mmmm: "+FORMAT\$(NOW,"mmmm")	yyyy:2013
TEXT 15,1010, "3",0,1,1, "q: "+FORMAT\$(NOW,"q")	h:14
TEXT 15,1060, "3",0,1,1, "y: "+FORMAT\$(NOW,"y")	hh:14
TEXT 15,1110, "3",0,1,1, "yy: "+FORMAT\$(NOW,"yy")	n:46
TEXT 15,1160, "3",0,1,1, "yyyy: "+FORMAT\$(NOW,"yyyy")	nn:46
TEXT 15,1210, "3",0,1,1, "h: "+FORMAT\$(NOW,"h")	s:18
TEXT 15,1260, "3",0,1,1, "hh: "+FORMAT\$(NOW,"hh")	ss:18
TEXT 15,1310, "3",0,1,1, "n: "+FORMAT\$(NOW,"n")	tttt:2:46:18 PM
TEXT 15,1360, "3",0,1,1, "nn: "+FORMAT\$(NOW,"nn")	AM/PM:PM
TEXT 15,1410, "3",0,1,1, "s: "+FORMAT\$(NOW,"s")	am/pm:pm
TEXT 15,1460, "3",0,1,1, "ss: "+FORMAT\$(NOW,"ss")	A/P:P
TEXT 15,1510, "3",0,1,1, "tttt: "+FORMAT\$(NOW,"tttt")	a/p:p
TEXT 15,1560, "3",0,1,1, "AM/PM: "+FORMAT\$(NOW,"AM/PM")	AMPM:PM
	\:Today is 1/9/2013
	string:Today is 1/9/2013

```
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$(NOW,"To\da\y i\s dddd")
TEXT 15,1860, "3",0,1,1, "string: " +FORMAT$(NOW,"To\da\y i\s dddd")
PRINT 1
```

Sample Code (Since A1.97)	Result
<pre> 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): "+FORMAT\$(-12.3,"0.00;(0.00)") PRINT 1 </pre>	<pre> 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) </pre>

# 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)**

<u>Parameter</u>	<u>Description</u>																						
interval\$,	The time/date interval for adding. It can be one of following values. <table border="1" data-bbox="571 734 1136 1305"><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"																						
<i>Note:</i>	<i>This command has been supported since V6.87 EZ and later firmware.</i>																						

## 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:nn: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\$)**

<u>Parameter</u>	<u>Description</u>
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()

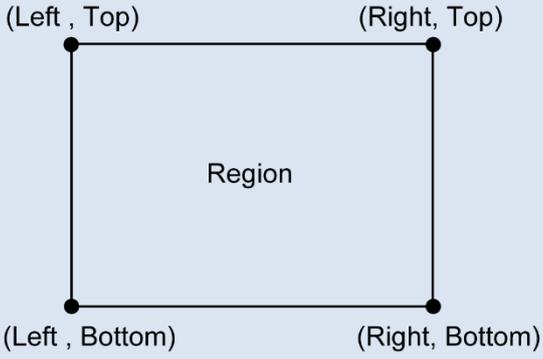
## Description

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)**

<u>Parameter</u>	<u>Description</u>
left	Left side position of region (pixel)
top	Top side position of region (pixel)
right	Right side position of region (pixel)
bottom	Bottom side position of region (pixel)

The diagram shows a rectangle with four vertices marked by black dots. The top-left vertex is labeled "(Left , Top)", the top-right vertex is labeled "(Right, Top)", the bottom-left vertex is labeled "(Left , Bottom)", and the bottom-right vertex is labeled "(Right, Bottom)". The word "Region" is centered inside the rectangle.

**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
```

# MULTIKEY()

## Description

This command is used to detect the status of keys by keyboard scancode. Returns which key is currently pressed using keyboard scancodes.

## Syntax

**MULTIKEY(scancode)**

<u>Parameter</u>	<u>Description</u>
<b>scancode</b>	<p>Here follows a list of hardware keyboard scancodes accepted by the MULTIKEY function. These are equal to DOS scancodes, and are guaranteed to be always recognized on all platforms.</p> <p>SC_ESCAPE = 1 SC_1 = 2 SC_2 = 3 SC_3 = 4 SC_4 = 5 SC_5 = 6 SC_6 = 7 SC_7 = 8 SC_8 = 9 SC_9 = 10 SC_0 = 11 SC_MINUS = 12 SC_EQUALS = 13 SC_BACKSPACE = 14 SC_TAB = 15 SC_Q = 16 SC_W = 17 SC_E = 18 SC_R = 19 SC_T = 20 SC_Y = 21 SC_U = 22 SC_I = 23 SC_O = 24 SC_P = 25 SC_LEFTBRACKET = 26 SC_RIGHTBRACKET = 27 SC_ENTER = 28 SC_CONTROL = 29 SC_A = 30 SC_S = 31 SC_D = 32 SC_F = 33 SC_G = 34 SC_H = 35 SC_J = 36 SC_K = 37</p>

SC\_L = 38  
SC\_SEMICOLON = 39  
SC\_QUOTE = 40  
SC\_TILDE = 41  
SC\_LSHIFT = 42  
SC\_BACKSLASH = 43  
SC\_Z = 44  
SC\_X = 45  
SC\_C = 46  
SC\_V = 47  
SC\_B = 48  
SC\_N = 49  
SC\_M = 50  
SC\_COMMA = 51  
SC\_PERIOD = 52  
SC\_SLASH = 53  
SC\_RSHIFT = 54  
SC\_MULTIPLY = 55  
SC\_ALT = 56  
SC\_SPACE = 57  
SC\_CAPSLOCK = 58  
SC\_F1 = 59  
SC\_F2 = 60  
SC\_F3 = 61  
SC\_F4 = 62  
SC\_F5 = 63  
SC\_F6 = 64  
SC\_F7 = 65  
SC\_F8 = 66  
SC\_F9 = 67  
SC\_F10 = 68  
SC\_NUMLOCK = 69  
SC\_SCROLLLOCK = 70  
SC\_HOME = 71  
SC\_UP = 72  
SC\_PAGEUP = 73  
SC\_LEFT = 75  
SC\_RIGHT = 77  
SC\_PLUS = 78  
SC\_END = 79  
SC\_DOWN = 80  
SC\_PAGEDOWN = 81  
SC\_INSERT = 82  
SC\_DELETE = 83  
SC\_F11 = 87  
SC\_F12 = 88

**Note:**

*It will return -1 if the key is pressed, otherwise it will return 0.*

## Example

### Sample Code

```
DOWNLOAD "DEMO.BAS"

SC_ESCAPE = 1
SC_ENTER  = 28
SC_UP     = 72
SC_DOWN   = 80

SET KEY2 OFF
SET KEY4 OFF
SET KEY5 OFF
SET KEY6 OFF

;; Main Loop
:MAINLOOP
GOSUB WAITKEY
IF KEY = 2 THEN OUT "Exit"
IF KEY = 5 THEN OUT "Enter"
IF KEY = 4 THEN OUT "Up"
IF KEY = 6 THEN OUT "Down"
GOTO MAINLOOP

;; Wait Button or Keyboard
:WAITKEY
KEY = 0
IF KEY2 <> 0 THEN KEY = 2
IF KEY5 <> 0 THEN KEY = 5
IF KEY4 <> 0 THEN KEY = 4
IF KEY6 <> 0 THEN KEY = 6
IF MULTIKEY(SC_ESCAPE) <> 0 THEN KEY = 2
IF MULTIKEY(SC_ENTER)  <> 0 THEN KEY = 5
IF MULTIKEY(SC_UP)     <> 0 THEN KEY = 4
IF MULTIKEY(SC_DOWN)   <> 0 THEN KEY = 6
IF KEY = 0 THEN GOTO WAITKEY
SOUND 26,100
RETURN

EOP
RUN "DEMO.BAS"
```

# 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])**

<u>Parameter</u>	<u>Description</u>
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 1:	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(", ")) </pre>	<p>Row 1 and Column 1 = John            Row 2 and Column 1 = Mary            John Age = 18            Mary Age = 30            John Height = 180            Mary Height = 150</p> <p><b>Table format (TEST.CSV)</b></p> <table border="1"> <tr> <td>Number of rows</td> <td colspan="4">3</td> </tr> <tr> <td>Name of column</td> <td>Name</td> <td>Age</td> <td>Height</td> <td>Weight</td> </tr> <tr> <td>Row 1</td> <td>John</td> <td>18</td> <td>180</td> <td>80</td> </tr> <tr> <td>Row 2</td> <td>Mary</td> <td>30</td> <td>150</td> <td>50</td> </tr> <tr> <td>Row 3</td> <td>Mark</td> <td>65</td> <td>170</td> <td>65</td> </tr> <tr> <td></td> <td>Column 1</td> <td>Column 2</td> <td>Column 3</td> <td>Column 4</td> </tr> </table>	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
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																											

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

**Sample Code 2: (since VA1.97)****Result**

```
DOWNLOAD "TEST.CSV",123,6
```

```
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

6

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

# REPLACE\$( )

## Description

This command returns a string in which a specified substring has been replaced with another substring.

## Syntax

**REPLACE\$( "str1\$", " sub1\$", " sub2\$")**

<u>Parameter</u>	<u>Description</u>
str1\$	Required. The string that will be searched for replacing.
Sub1\$	Required. The specified substring that will be replaced.
Sub2\$	Required. Replacement substring.

**Note:**  
*This command has been supported since A1.92 EZD and later firmware.*

## Example

Sample Code	Result
<pre>DOWNLOAD F,"TEST.BAS" SIZE 3,2 GAP 0,0 DIRECTION 1 INPUT A\$ DATA\$ = REPLACE\$(A\$,"ABC","123") CLS TEXT 100,100,"3",0,1,1,DATA\$ PRINT 1 EOP TEST ABCDEF</pre>	

# 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"**

<u>Parameter</u>	<u>Description</u>
@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.  <b><i>@51~@55 were supported since V6.37 EZ.</i></b> <b><i>@56~@60 were supported since V6.74 EZ.</i></b>
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      999U 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**

<u>Parameter</u>	<u>Description</u>
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
<pre>SIZE 3,3 GAP 0,0 SET CUTTER OFF SET PEEL OFF CLS TEXT 50,50, "3",0,1,1, "SET CUTTER OFF" PRINT 3</pre>	The cutter function is disabling.
<pre>SET CUTTER BATCH CLS TEXT 50,50, "3",0,1,1, "SET CUTTER BATCH" PRINT 3,2</pre>	The cutter cuts once after 6 labels are printed.

<pre>SET CUTTER 1 CLS TEXT 50,50, "3",0,1,1, "SET CUTTER 1" PRINT 3,2</pre>	<p>The cutter cuts every label.</p>
<pre>CLS TEXT 50,50, "3",0,1,1, "SET CUTTER 2" PRINT 3,2</pre>	<p>The cutter cuts every 2 labels.</p>

## 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**

<u>Parameter</u>	<u>Description</u>
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/ SUPRESS

<u>Parameter</u>	<u>Description</u>
OFF	Disable back function.
ON	Enable back function. Additional length of the label will be printed to ensure the content is complete.
SUPRESS	Disable back function. To adhere to the current SIZE setting for label printing, some parts of the content may be missing. (Since A2.17)

Note: TDP-643 Plus, TTP-243, TTP-342, TTP-244ME, TTP-342M and TTP-248M series are not supported this feature.

## 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"**

<u>Parameter</u>	<u>Description</u>
n	0, 1, 2, 3, 4, 5, 6
ON	Enable KEYn function
OFF	Disable KEYn function
DEFAULT	Set KEYn back to 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 that can manual control the feeding distance by keep pressing the key.
BACKFEED	Set to "BACKFEED" key that that can manual control the back feeding distance by keep pressing the key.
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
TDP-643 Plus/ 643R Plus		PAUSE					
TTP-243/243 Plus/243 Pro series, TTP-244ME/244 ME Plus/244M Pro series, TTP-244/ 244 Plus series		PAUSE	FEED				
TDP-245/247 series, TTP-245/247 series, TTP-245C series, TDP-225 series, TTP-225 series, TA200 series, Alpha-3R, DA series, TE series, Alpha-2R, TDM-20, TDM-30, Alpha-30R		FEED					

TX200/210 series (with LCD), TC210 series (with LCD), MX240P series, MX241 series, MH series, ML240P series, ML241 series, MB240T/241T series, PEX series, MH261 series, Alpha-30L/40L series,		FEED	MENU	UP	RIGHT	LEFT	DOWN
TTP-246M series		MENU	PAUSE	FEED	(UP)	(DOWN)	(SELECT)
TTP-248M series		MENU	PAUSE	FEED			
TTP-2410M/2410M Pro series, TTP-246M Plus/246M Pro series, TTP-268M series, TTP-384M series, ME240(with LCD) series		MENU	PAUSE	FEED	UP	DOWN	SELECT
ME240 (Non-LCD) series, ML240 series, MB240 series		FEED	PAUSE				
M23 series	FEED	LEFT	MID	RIGHT			
Alpha-4L		FEED	INFO	MENU			
MX240 series, TTP-2410MT/MU series		PAUSE	MENU	FEED	UP	SELECT	DOWN
DH/TH series		FEED	PAUSE	REPRINT			

**Note:** Please refer to [printer model list](#) for more detail

## 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

<u>LED no.</u>	<u>Default Function</u>								
LDE1	Power on/off								
LED2	Printer on-line/off-line								
LED3	Error/normal								
<u>Parameter</u>	<u>Description</u>								
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	LED8	LED2 & LED3
TDP-643 Plus/ 643R Plus series	ONLINE	ERROR	ERROR						
TTP-243/243 Plus/243 Pro series, TTP-244ME/244 ME Plus/244M Pro series, TTP-244/ 244 Plus series, TTP-2410M/2410M Pro series, TTP-246M Plus/246M Pro series, TTP-268M series, TTP-384M series, ME240 series, MX240 series, MX240P series	POWER	ONLINE	ERROR						
MB240 series	GREEN	GREEN	RED	CARRIAGE	RIBBON	PAPER	WIRELESS		ORANGE
TDP-245/247 series, TTP-245/247 series, TTP-245C series, TDP-225 series, TTP-225 series, DA200 series, TA200 series, TC210series, TE200 series, MH series <b>Note: For this series, the LED1=LED2</b>	GREEN	GREEN	RED						ORANGE
Alpha-2R series, Alpha-3R series, TDM-20 series, TDM-30 series	GREEN	GREEN	RED	BAT1	BAT2	BAT3	BT/WIFI		ORANGE
Alpha-4L series	GREEN	RED	BAT1	BAT2	BAT3	BT	WIFI		ORANGE
PEX series, TX210 series, MB241 series, ML241 series, MX241 series, MH241 series, MH261 series, Alpha-30L/40L series, Alpha-30R	GREEN	GREEN	RED						ORANGE
DH/TH series (LED version only)	GREEN	GREEN	RED		RIBBON	PAPER	CARRIAGE	WIRELESS	ORANGE

**Note: Please refer to [printer model list](#) for checking series printers.**

## 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

<u>Parameter</u>	<u>Description</u>
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

OFFEST, 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

<u>Parameter</u>	<u>Description</u>
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

OFFEST, 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)

Note: Please refer to [printer model list](#) for checking TSPL or TSPL2.

<u>Parameter</u>	<u>Description</u>
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**

<u>Parameter</u>	<u>Description</u>		
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.		

Printer model	Gap Sensor Range	Black Mark Sensor Range	SET GAP REVERSE SET GAP OBVERSE SET GAP AUTO
TTP-243 series, TTP-244ME series, TDP-643 Plus series, TTP-342 series, TTP-342M series	0~15	ON/OFF	<b>V</b>
TTP-243 Plus series, TTP-244 series, TTP-244ME Plus series, TDP-643R Plus series, TTP-342 Plus series	0~255	ON/OFF	<b>V</b>
TTP-243 Pro series, TTP-244 Plus series	0~63	ON/OFF	
TTP-245C series, TTP-225 series, TDP-225 series	0~31	0~3	<b>V</b>
TTP-245/343 series, TDP-245 series, TTP-246M/344M series (non usb)	0~63	0~63	<b>V</b>
TTP-245 Plus/343 Plus series, TDP-245 Plus series, TTP-247 series, TDP-247 series	0~15	0~15	<b>V</b>
TTP-246M/344M series (usb)	0~31	0~31	<b>V</b>

TTP-246M Plus, TTP-2410M series, TTP-344M Plus series, TTP-346M series, TTP-384M series, TTP-644M series, ME240 series	0~7	0~3	V
TTP-2410M Pro series, TDM-20, TDM-30	0~7	0~7	V
TTP-248M series, M23 series	0~255	0~255	V
TA200 series, MB240 series, MH240 series	0~15	0~3	V
Alpha-4L series	0~15	0~7	V

**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.*

*\* Please refer to [printer model list](#) for checking series printers.*

## 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 using to reverse/obverse the sensor function.

### Syntax

#### SET BLINE REVERSE/OBVERSE

<u>Parameter</u>	<u>Description</u>
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 BLINE PRINTSIDE & SET BLINE BACKSIDE

### Description

This setting is used to set the top or bottom black mark sensor as the main transmitter.

### Syntax

**SET BLINE PRINTSIDE**

**SET BLINE BACKSIDE**

<u>Parameter</u>	<u>Description</u>
PRINTSIDE	The light of the sensor will face toward the print side of the label.
BACKSIDE	The light of the sensor will face toward the back side of the label.

Supported printer model	Default
TDM-30	Print side
Alpha-30L	Back side
Alpha-40L	Back side
MH241 Series	Back side
MX241P Series	Back side
PEX-1001 Series	Back side

## 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.

Note: Please refer to [printer model list](#) for checking TSPL2 printers.

### Syntax

**SET HEAD ON /OFF**

<u>Parameter</u>	<u>Description</u>
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

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

## 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**

<u>Parameter</u>	<u>Description</u>
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

<u>Parameter</u>	<u>Description</u>
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**

<u>Parameter</u>	<u>Description</u>
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.*



Syntax	Receive "PRINT m,n"	Print Out
SET PRINTKEY ON or SET PRINTKEY AUTO	1.) PRINT 1,2	Label 1, Label 1
	2.) Press FEED key	Label 2, Label 2

Syntax	Receive "PRINT -1,n"	Print Out
SET PRINTKEY ON or SET PRINTKEY AUTO	1.) PRINT -1,2	Label 1, Label 1
	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**

<u>Parameter</u>	<u>Description</u>
OFF	Disable this function
ON	Enable this function

**Note:** This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

## 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**

<u>Parameter</u>	<u>Description</u>
n	The feeding length in dot.

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

### Example

Sample code	Result
<b>SET FEED_LEN 100</b>	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 (Analogue-to-Digital value). We can use it to check the sensor function.

## Syntax

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

<u>Parameter</u>	<u>Description</u>	
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
	<b>BATTERY VOLT</b>	The voltage of battery (since A2.05 EZC)
	<b>BATTERY CAP</b>	The capacity of battery (since A2.05 EZC)
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:**

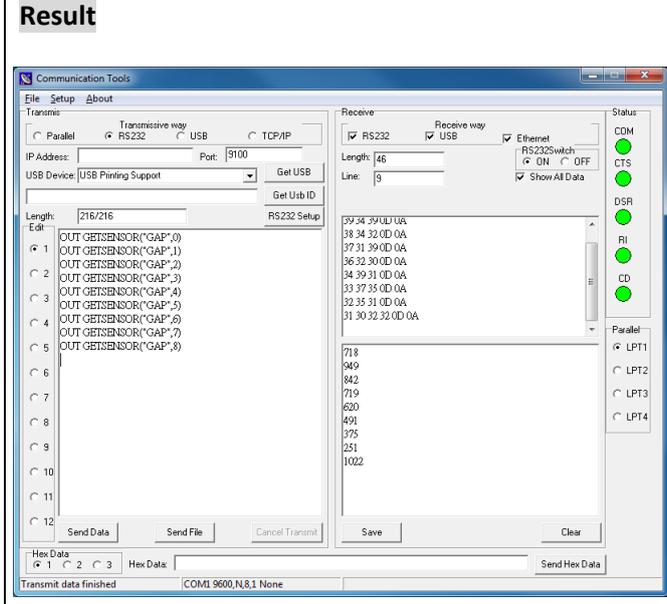
*This command has been supported since V6.75 EZ and later firmware.*

**Example (Use CommTool to get sensor status via RS-232.)**

```

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)
    
```

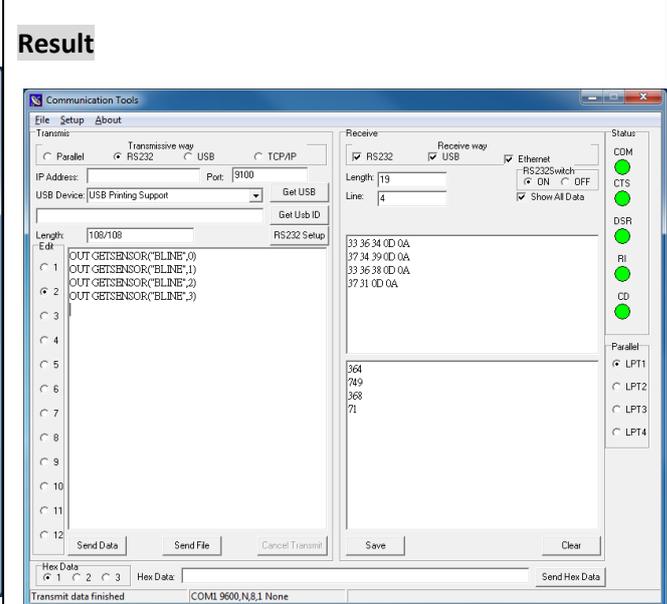


***\*If the returned valued is changed in different sensor intension, we can say the sensor is functional.***

```

Sample code

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

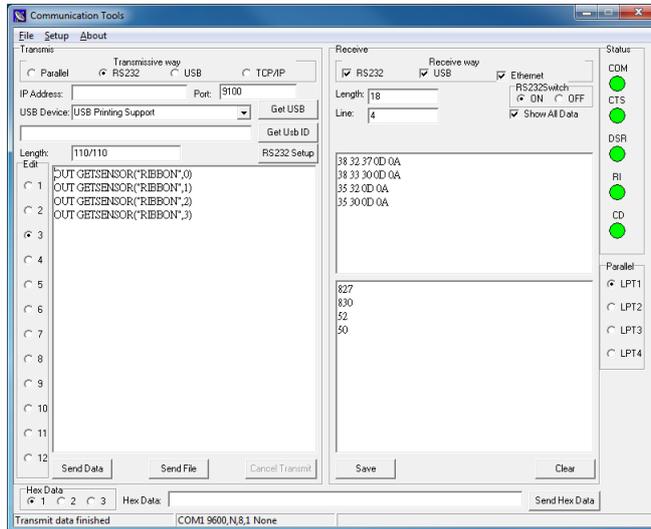


***\*If the returned valued is changed in different sensor intension, we can say the sensor is functional.***

### Sample code

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

### Result

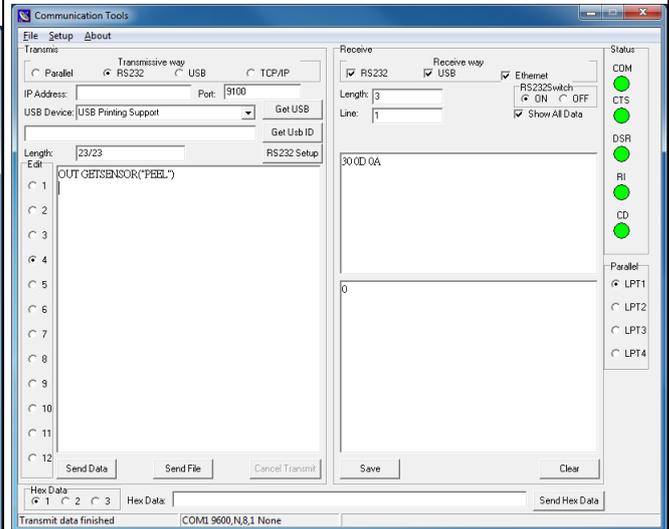


**\*If the returned valued is changed in different sensor intension, we can say the sensor is functional.**

### Sample code

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

### Result

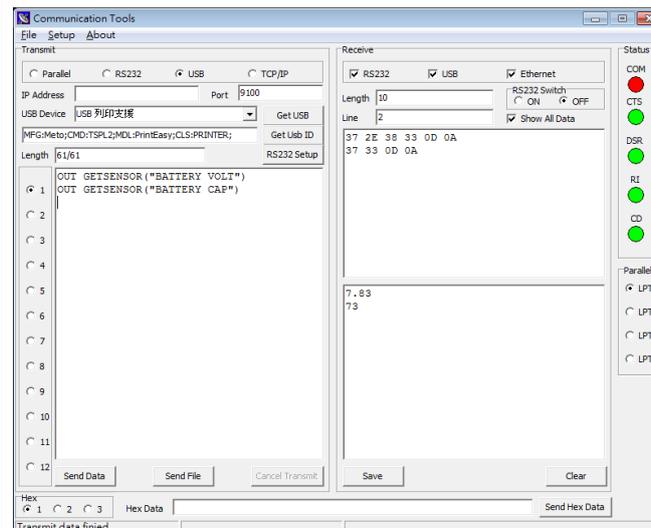


**\*0: Paper is not on the sensor. 1: Paper is on the sensor.**

### Sample code

```
OUT GETSENSOR("BATTERY VOLT")
OUT GETSENSOR("BATTERY CAP")
```

### Result



**\*This code used to detect the battery voltage and battery capacity for mobile Printer since A2.05 EZC and later firmware.**

# GETSETTING\$()

## Description

This command is used to get printer settings.

## Syntax

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

Parameter			Description	
app\$	sec\$	key\$		
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	
		PRINTQUALITY	Return Printer print mode (DRAFT, STANDARD or OPTIMUM; see SET PRINTQUALITY)	For Alpha-2R series printer
		STANDBYTIME	Return Printer standby time (OFF or number)	
		IMAGE LENGTH	Return image length value by dots. (Since A2.17)	
		IMAGE WIDTH	Return image width value by dots. (Since A2.17)	
	RECORD	MILAGE	Return printed mileage (in dots)	
		LABEL COUNTER	Return the total number of prints	
		CUT COUNTER	Return cutter cuts	
	WLAN	RSSI	Return WIFI RSSI	Since A2.13
	DOTSCAN	BADDOT	Return bad dot	
	INFORMATION	PRINTER STATUS	Return printer status	
	TPHID	PRINT QUALITY	Return Print Quality	
	RTC	YEAR	Return Year	Since A2.13
	RTC	MONTH	Return Month	
	RTC	DATE	Return Date	
	RTC	HOUR	Return Hour	
	RTC	MINUTE	Return Minute	
	RTC	SECOND	Return Second	
	FILE	DRAM	CAPACITY	Return the total capacity of DRAM
			AVAILABLE	Return the available capacity of DRAM
PHYSICAL			Return the Dram Phy space	

CONFIG	FLASH	CAPACITY	Return the total capacity of FLASH		
		AVAILABLE	Return the available capacity of FLASH		
		PHYSICAL	Return the Flash Phy space		
		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.	
			PHYSICAL	Return the Card Phy space	
		USB	PHYSICAL	Return the USB Phy space	
		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				
PRIMARY DNS	Return primary DNS			since V8.12 & A2.09	
SECONDARY DNS	Return secondary DNS				
WLAN	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		
	SSID		Return SSID	Since A2.13	
	REGION		Return WiFi region		
	PRINT SERVER NAME	Return print server name			
FREQUENCY	Return WiFi frequency				
CONFIG	BT	PIN CODE	Return BT Pin Code	Since A2.13	
		NAME	Return BT Name		
		MODE	Return BT Mode		
		SW VERSION	Return BT Version		
		MAC ADDRESS	Return BT Local MAC		
		PAIR MODE	Return BT Pair Mode		
	COM1	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		
	SENSOR	SENSOR TYPE	Return the current sensor type		
		CARRIAGE	Return the status of head open sensor.		
GAP INTENSION		Return intension of gap sensor.			
BLINE INTENSION		Return intension of black mark sensor.			

TSPL	CONTINUOUS INTENSION	Return intension of continuous sensor.	
	THRESHOLD	Return threshold	Since A2.13
	AUTO THRESHOLD	Return auto Threshold	
	LABEL CAPACITY	Return label capacity	
	RIBBON CAPACITY	Return ribbon capacity	
	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	
	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	
	GAP OFFSET	Return gap offset value	Since A2.13
RIBBON SENSOR	Return ribbon sensor value		
RIBBON ENCODER	Return ribbon encoder value		
CUT PIECE	Return cutter piece		

**Parameter**

[default\$]

**Description**

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 "Ethernet PRIMARY DNS = ";GETSETTING$("CONFIG","NET","PRIMARY DNS")
OUT "Ethernet SECONDARY DNS = ";GETSETTING$("CONFIG","NET","SECONDARY DNS")
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")

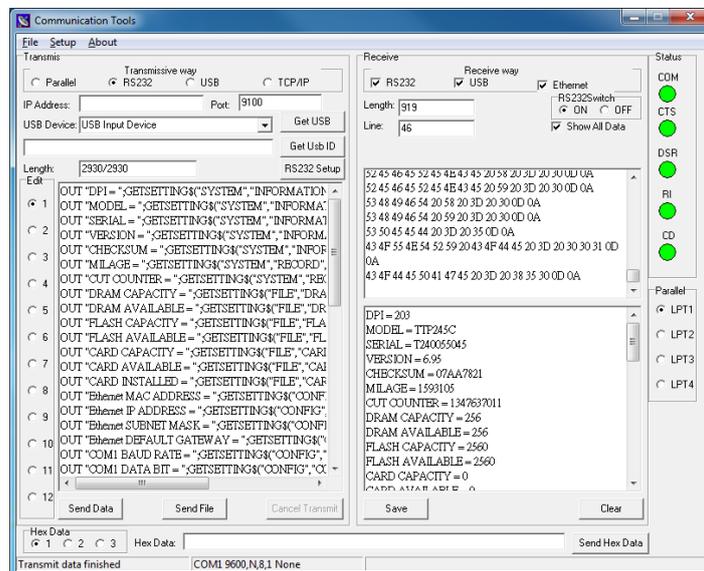
OUT "SPEED = ";GETSETTING$("CONFIG", "TSPL", "SPEED")

OUT "COUNTRY CODE = ";GETSETTING$("CONFIG", "TSPL", "COUNTRY CODE")

OUT "CODEPAGE = ";GETSETTING$("CONFIG", "TSPL", "CODEPAGE")

```

## 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

<u>Parameter</u>	<u>Description</u>
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 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"**

<u>Parameter</u>	<u>Description</u>
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 RESPONSE

## Description

This command can response issue automatically.

## Syntax

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

<u>Parameter</u>	<u>Description</u>
["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}**

<u>Status</u>
[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}

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}

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"**

<u>Parameter</u>	<u>Description</u>
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"
	<b>Month</b>
	"JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"
	"JANUARY", "FEBRUARY", "MARCH", "APRIL", "MAY", "JUNE", "JULY", "AUGUST", "SEPTEMBER", "OCTOBER", "NOVEMBER", "DECEMBER"
	<b>Week</b>
	"SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT"
	"SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY"
	<b>Which Week</b>
	"FIRST", "SECOND", "THIRD", "FOURTH", "LAST"
	"1 <sup>ST</sup> ", "2 <sup>ND</sup> ", "3 <sup>RD</sup> ", "4 <sup>TH</sup> ", "LAST"
	<b>Date</b>
	1~31
	<b>Time</b>
	0:00~23:00
<b>Note:</b>	
	<i>This command has been supported since V8.03 EZ and later firmware.</i>

## 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"
```

# SET REGISTRATION

## Description

This command is used to set the mode of label positioning for the label roll with multiple sizes labels.

## Syntax

### SET REGISTRATION mode

<u>Parameter</u>	<u>Description</u>
mode	BYSIZE: Original registration (Default)  ACTUAL: For handle label rolls that have different label sizes on the same roll.
<i>Note:</i>  <i>This command has been supported since A2.12 EZD and later firmware.</i>	

## Example

Sample Code	Result
150 <u>jobs on 2 labels:</u>  SET REGISTRATION ACTUAL SIZE 4,1 GAP 0.12,0 CLS TEXT 40,40,"0",0,10,10,"1 <sup>st</sup> Label 4x1" PRINT 1 SIZE 4,5 GAP 0.12,0 CLS TEXT 40,40,"0",0,10,10,"2 <sup>nd</sup> Label 4x5" PRINT 1	

150 job with 2 labels:

**SET REGISTRATION ACTUAL**

**SIZE 4,6.12**

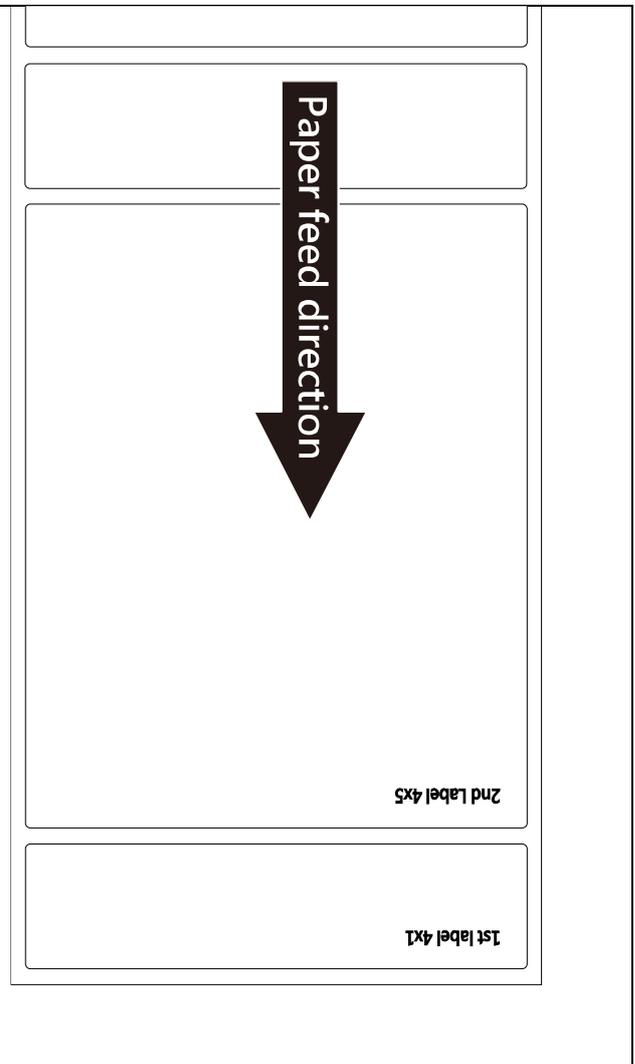
**GAP 0.12,0**

**CLS**

**TEXT 40,40,"0",0,10,10,"1<sup>st</sup> Label 4x1"**

**TEXT 40,267,"0",0,10,10,"2<sup>nd</sup> Label 4x5"**

**PRINT 1**



## SET APPLICATOR

### Description

Used for print & apply application, the label is moved forward to be removed by applicator, and applied to an item. Only printer with GPIO interface supported this mode.

### Syntax

#### SET APPLICATOR ON/OFF

<u>Return Value</u>	<u>Description</u>
ON	Enable the applicator function
OFF	Disable the applicator function

*Note:*  
*This command has been supported since A2.15 EZD and later firmware.*

### Example

Sample code
SET APPLICATOR ON

# SET MENULOCK

## Description

This command can enable or disable menu lock function, or setup password of it.

## Syntax

**SET MENULOCK n**

<u>Return Value</u>	<u>Description</u>
ON	Enable the menu lock function
OFF	Disable the menu lock function
n	0000 <= "n" <= 9999, 4 digits number to setup password

*Note:*  
*This command has been supported since A2.17 EZD and later firmware.*

## Example

```
Sample code

REM *****Enable function*****
SET MENULOCK ON

REM *****Setup password as 1234*****
SET MENULOCK "1234"
```

# PEEL

## Description

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

## Syntax

### PEEL

<u>Return Value</u>	<u>Description</u>
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**

<u>Return Value</u>	<u>Description</u>								
m	m=1, LED1 m=2, LED2 m=3, LED3								
n	0: turn off LED 1: light on LED								
<b>Model</b>	<b>LED1</b>	<b>LED2</b>	<b>LED3</b>	<b>LED4</b>	<b>LED5</b>	<b>LED6</b>	<b>LED7</b>	<b>LED8</b>	<b>LED2 &amp; LED3</b>
TDP-643 Plus/ 643R Plus series	ONLINE	ERROR	ERROR						
TTP-243/243 Plus/243 Pro series, TTP-244ME/244 ME Plus/244M Pro series, TTP-244/ 244 Plus series  TTP-2410M/2410M Pro series, TTP-246M Plus/246M Pro series, TTP-268M series, TTP-384M series, ME240 series, MX240 series, MX240P series	POWER	ONLINE	ERROR						
MB240 series	GREEN	GREEN	RED	CARRIAGE	RIBBON	PAPER	WIRELESS		ORANGE
TDP-245/247 series, TTP-245/247 series, TTP-245C series, TDP-225 series, TTP-225 series, DA200 series, TA200 series, TC210series, TE200 series, MH series  <b>Note: For this series, the LED1=LED2</b>	GREEN	GREEN	RED						ORANGE
Alpha-2R series , Alpha-3R series, TDM-20 series, TDM-30 series	GREEN	GREEN	RED	BAT1	BAT2	BAT3	BT/WIFI		ORANGE
Alpha-4L series	GREEN	RED	BAT1	BAT2	BAT3	BT	WIFI		ORANGE
PEX series, TX210 series, MB241 series, ML241 series, MX241 series, MH241 series, MH261 series, Alpha-30L/40L series, Alpha-30R	GREEN	GREEN	RED						ORANGE
DH/TH series (LED version only)	GREEN	GREEN	RED		RIBBON	PAPER	CARRIAGE	WIRELESS	ORANGE

**Note: Please refer to printer model list for checking series printers.**

## 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
TDP-643 Plus/ 643R Plus		PAUSE					
TTP-243/243 Plus/243 Pro series, TTP-244ME/244 ME Plus/244M Pro series, TTP-244/244 Plus series		PAUSE	FEED				
TDP-245/247 series, TTP-245/247 series, TTP-245C series, TDP-225 series, TTP-225 series, TA200 series, Alpha-3R, DA series, TE series, Alpha-2R, TDM-20, TDM-30		FEED					
TX200 series (with LCD), TC210 series (with LCD), TX600 series, MX240P series, MH series		FEED	MENU	UP	RIGHT	LEFT	DOWN
TTP-246M series		MENU	PAUSE	FEED	(UP)	(DOWN)	(SELECT)
TTP-248M series		MENU	PAUSE	FEED			
TTP-2410M/2410M Pro series, TTP-246M Plus/246M Pro series, TTP-268M series, TTP-384M series, ME240(LCD control panel) series		MENU	PAUSE	FEED	UP	DOWN	SELECT
ME240 (Non-LCD control panel) series		FEED	PAUSE				
M23 series	FEED	LEFT	MID	RIGHT			
Alpha-4L		FEED	INFO	MENU			
MX240 series, TTP-2410MT/MU series		PAUSE	MENU	FEED	UP	SELECT	DOWN
DH/TH series		FEED	PAUSE	REPRINT			

**Note:** Please refer to [printer model list](#) for checking series printers.

### Syntax

**KEYm = n**

Key	Return Value
KEY1 (MENU)	0: released 1: pressed
KEY2 (PAUDE)	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
```

## SET SENSOR\_REF

### Description

This command can set the threshold detection of sensor.

### Syntax

**SET SENSOR\_REF AUTO/MANUAL**

<u>Parameter</u>	<u>Description</u>
AUTO	When feeding paper, the paper positioning threshold is automatically fine-tuned according to the paper picker (high/low peak); Default
MANUAL	When feeding paper, the paper positioning threshold is NOT automatically fine-tuned according to the paper picker (high/low peak), the paper positioning threshold is fixed.

### Example

Sample code
<b>SET SENSOR_REF MANUAL</b>
<b>SET SENSOR_REF AUTO</b>

# SET RIBBON\_SAVER

## Description

This command can enable or disable ribbon saver function. (Since A2.16)

## Syntax

**SET RIBBON\_SAVER ON/OFF**

<u>Parameter</u>	<u>Description</u>
ON	Enable Ribbon Saver function
OFF	Disable Ribbon Saver function

## Example

### Sample code

```
SET RIBBON_SAVER ON
```

# SET SBPLIMCMD

## Description

This command can allow printer to recognize immediate command for SBPL. (Since A2.16)

## Syntax

**SET SBPLIMCMD ON/OFF**

<u>Parameter</u>	<u>Description</u>
ON	Enable Immediate command for SBPL
OFF	Disable Immediate command for SBPL

## Example

Sample code
<b>SET SBPLIMCMD ON</b>

# SET DPLIMCMD

## Description

This command can allow printer to recognize immediate command for DPL. (Since A2.16)

## Syntax

**SET DPLIMCMD ON/OFF**

<u>Parameter</u>	<u>Description</u>
ON	Enable Immediate command for DPL
OFF	Disable Immediate command for DPL

## Example

Sample code
<b>SET DPLIMCMD ON</b>

# SET ZPLIMCMD

## Description

This command can allow printer to recognize immediate command for ZPL. (Since A2.16)

## Syntax

**SET ZPLIMCMD ON/OFF**

<u>Parameter</u>	<u>Description</u>
ON	Enable Immediate command for ZPL
OFF	Disable Immediate command for ZPL

## Example

Sample code
<b>SET ZPLIMCMD ON</b>

# SET COVER

## Description

This command is used to enable or disable the warning message of PEX-2000 once cover is opened.

## Syntax

**SET COVER ON/OFF**

<u>Parameter</u>	<u>Description</u>
ON	Enable warning once cover is opened
OFF	Disable warning once cover is opened

## Example

Sample code
<b>SET COVER ON</b>

# 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)

<u>Parameter</u>	<u>Description</u>
n	Number of labels printed. 0<=n<=999999999

### Example

Sample code	Result
-------------	--------

```
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$(LABEL)
TEXT 10,150, "3",0,1,1, "*****Statement 1*****"
IF LABEL>1000 THEN
TEXT 10,200, "3",0,1,1, "LABEL>1000"
ELSE
TEXT 10,200, "3",0,1,1, "LABEL<1000"
ENDIF
TEXT 10,250, "3",0,1,1, "*****Statement 1*****"
A=LABEL
IF A>1000 THEN
TEXT 10,300, "3",0,1,1, "A>1000"
ELSE
TEXT 10,300, "3",0,1,1, "A<1000"
ENDIF
TEXT 10,350, "3",0,1,1, "*****Statement 3*****"
A$=STR$(LABEL)
IF VAL(A$)>1000 THEN
TEXT 10,400, "3",0,1,1, "VAL(A$)>1000"
ELSE
TEXT 10,400, "3",0,1,1, "VAL(A$)<1000"
ENDIF
PRINT 1,1
EOP
DEMO
```

```
1661
@LABEL=1661
*****Statement 1*****
LABEL>1000
*****Statement 1*****
A>1000
*****Statement 3*****
VAL(A$)>1000
```

# 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	
<pre>DOWNLOAD "SetYear.BAS" REM *****Set Year Parameter to RTC***** YEAR=13 EOP SetYear</pre>	
Sample code	Result
<pre>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)</pre>	<pre>YEAR1=2013 YEAR2=2013 YEAR3=2013</pre>

<pre>TEXT 10,90, "3",0,1,1, "YEAR3="+STR\$(YEAR)</pre>	
--	--

```
PRINT 1
```

```
EOP
```

```
DEMO
```

## 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	
<pre>DOWNLOAD "SetMonth.BAS" REM *****Set Month Parameter to RTC***** MONTH=01 EOP SetMonth</pre>	
Sample code	Result
<pre>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)</pre>	<pre>MONTH1=1 MONTH2=1 MONTH3=1</pre>

<b>PRINT 1</b> <b>EOP</b> <b>DEMO</b>	
---	--

## 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	
<pre>DOWNLOAD "SetDate.BAS" REM *****Set Date Parameter to RTC***** DATE=10 EOP SetDate</pre>	
Sample code	Result
<pre>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)</pre>	<pre>DATE1=10 DATE2=10 DATE3=10</pre>

<code>TEXT 10,90, "3",0,1,1, "DATE3="+STR\$(DATE)</code>	
--	--

`PRINT 1`

`EOP`

`DEMO`

## See Also

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

# WEEK

## Description

This variable reads 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	Result
<pre>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</pre>	<pre>WEEK1=5 WEEK2=5 WEEK3=5</pre>

## 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	
<pre>DOWNLOAD "SetHour.BAS" REM *****Set Hour Parameter to RTC***** HOUR=10 EOP SetHour</pre>	
Sample code	Result
<pre>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)</pre>	<pre>HOUR1=10 HOUR2=10 HOUR3=10</pre>

<b>PRINT 1</b> <b>EOP</b> <b>DEMO</b>	
---	--

## 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	
<pre>DOWNLOAD "SetMinute.BAS" REM *****Set Minute Parameter to RTC***** MINUTE=27 EOP SetMinute</pre>	
Sample code	Result
<pre>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)</pre>	<pre>MINUTE1=27 MINUTE2=27 MINUTE3=27</pre>

<pre>TEXT 10,90, "3",0,1,1, "MINUTE3="+STR\$(MINUTE)</pre>	
--	--

```
PRINT 1
```

```
EOP
```

```
DEMO
```

## 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	
<pre>DOWNLOAD "SetSecond.BAS" REM *****Set Second Parameter to RTC***** SECOND=59 EOP SetSecond</pre>	
Sample code	Result
<pre>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)</pre>	<pre>SECOND1=59 SECOND2=59 SECOND3=59</pre>

<b>PRINT 1</b> <b>EOP</b> <b>DEMO</b>	
---	--

## 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

Note: This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

### 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>	<pre>@YEAR      2005</pre>

### 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

Note: This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

## 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>	<pre>@MONTH      12</pre>

## 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

Note: This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

## 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>	<pre>@DATE      31</pre>

## See Also

~!C, @YEAR, @MONTH, @DAY, @HOUR, @MINUTE, @SECOND

## @DAY

### Description

This variable reads 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)

Note: This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

### 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>	<pre>@DAY      7</pre>

### 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

Note: This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

## 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>	<pre>@HOUR      23</pre>

## See Also

~!C, @YEAR, @MONTH, @DATE, @DAY, @MINUTE, @SECOND

## 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

Note: This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

## 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>	<pre>@MINUTE      59</pre>

## 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

Note: This command is only available for TSPL2 printers. Please refer to [printer model list](#) for checking TSPL2 printers.

## 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>	<pre>@SECOND      59</pre>

## 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	Result
<pre>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</pre>	<pre>Model: TDP247 Serial No.: D452350388 F/W Version: 7.00</pre>

### **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	Result
<pre>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</pre>	<pre>Model: TDP247 Serial No.: D452350388 F/W Version: 7.00</pre>

### **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	Result
<pre>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</pre>	<pre>Model: TDP247 Serial No.: D452350388 F/W Version: 7.00</pre>

### **See Also**

**\_MODEL\$, \_SERIAL\$**

# Bluetooth Module Setting Commands

## BT NAME

### Description

This command is used to set Bluetooth module name. (Max. 15 byte)

### Syntax

**BT NAME "name"**

#### Note:

You can use command **SELFTEST BT** to check the updated name.

### Example

Sample code	Result
<pre>BT NAME "TSC001" SELFTEST BT</pre>	<pre>-----           BT SETTING ----- MAC ADDR: 000CBF1213C0 NAME: TSC001 PIN CODE: 0000 PRINTER NAME: PAIR MODE: LEGACY MODULE: BM78 MFI MFI SUPPORTED: YES -----</pre>

# BT PINCODE

## Description

This command is used to set Bluetooth module PIN code. (Max. 15 byte)

## Syntax

**BT PINCODE "pincode"**

**Note:**

You can use command **SELFTEST BT** to check the updated PIN code.

## Example

Sample code	Result
<pre>BT PINCODE "1234" SELFTEST BT</pre>	<pre>-----                 BT SETTING ----- MAC ADDR: 000CBF1213C0 NAME: TSC001 PIN CODE: 1234 PRINTER NAME: PAIR MODE: LEGACY MODULE: BM78 MFI MFI SUPPORTED: YES -----</pre>

# BT PAIRMODE

## Description

This command is used to set Bluetooth pair mode. (since A2.12)

**Note: This function supports Mfi BM78 Bluetooth module only.**

## Syntax

**BT PAIRMODE "mode"**

<u>Parameter</u>	<u>Description</u>
mode	<b>LEGACY:</b> Legacy pairing mode (need to key-in pairing code) <b>SSP_JUSTWORK:</b> Just work pairing mode (default) <b>SSP_USERCONFIRM:</b> User configuration pairing mode (will ask if it be agreed to pair)

## Example

Sample code
<b>BT PAIRMODE LEGACY</b>
<b>BT PAIRMODE SSP_USERCONFIRM</b>
<b>BT PAIRMODE SSP_JUSTWORK</b>

## BT MODE

### Description

This command is used to set Bluetooth mode.

**Note:** This function supports Mfi Bluetooth module and Wi-Fi module.

### Syntax

**BT MODE mode (or "mode")**

<u>Parameter</u>	<u>Description</u>
mode (for Mfi Bluetooth module)	<b>Classic</b> BT3.0 (Default)
	<b>BLE</b> BT4.0
	<b>Dual</b> DUAL
"mode" (for Wi-Fi module)	<b>Classic</b> "BT2.1" (Default)
	<b>BLE</b> "BT4.0"

### Example

#### Sample code (For Mfi Bluetooth module)

```
BT MODE BT4.0
```

```
<Reboot printer>
```

#### Sample code (For Mfi Bluetooth module)

```
BT MODE DUAL
```

```
<Reboot printer>
```

## Sample code (For Wi-Fi module)

**BT MODE "BT4.0"**

**WLAN MODULE SAVECFG**

**DELAY 35000**

<Reboot printer>

# Wi-Fi Module Setting Commands

## 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"

<u>Parameter</u>	<u>Description</u>
ssid	It is the SSID of your wireless network.

*Note:*  
SSID is case-sensitive. The maximum length is 32 bytes.

### Example

Sample code	Result
WLAN SSID "TEST-AP" SELFTEST WLAN	<pre>-----           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 -----</pre>

### 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"**

<u>Parameter</u>	<u>Description</u>
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"**

<u>Parameter</u>	<u>Description</u>
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"**

<u>Parameter</u>	<u>Description</u>
ip	IP address.
Mask	Subnet mask.
Gateway	Default gateway.

## 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 9100 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: 9100 -----</pre>

## 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**

<u>Parameter</u>	<u>Description</u>
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>

# 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
<pre>NET DHCP SELFTEST ETHERNET</pre>	<pre>-----           ETHERNET SETTING -----           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 -----</pre>

### 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"**

<u>Parameter</u>	<u>Description</u>
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**

<u>Parameter</u>	<u>Description</u>
number	Base raw port number. Default is 9100.

## Example

Sample code	Result
<pre>NET PORT 9100 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>

# NET NAME

## Description

This command is used to set the printer server name.

## Syntax

**NET NAME "printerserver"**

<u>Parameter</u>	<u>Description</u>
printerserver	The specific name of printer server.

## Example

Sample code	Result
<pre>NET NAME "TEST" SELFTEST ETHERNET</pre>	<pre>-----           ETHERNET SETTING ----- 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 -----</pre>

## NET DNS

### Description

This command is used to set the printer to DNS. It supports Static IP only. (since A2.09)

### Syntax

**NET DNS "primary dns", "secondary dns"**

<u>Parameter</u>	<u>Description</u>
primary dns	Primary DNS IP address
secondary dns	Secondary DNS IP address

### Example

#### Sample code

```
NET DNS "10.0.1.102", "10.0.5.11"
```

# NET SNMP

## Description

This command can select the version of SNMP protocol or disable SNMP function. (since A2.16)

## Syntax

**NET SNMP n**

<u>Parameter</u>	<u>Description</u>
n	n = OFF, Disable SNMP function n = V1/V2C/V3, Enable SNMP function over different version

**Note: This command can only be sent by USB interface.**

## Example

Sample code
<b>NET SNMP OFF</b>
<b>NET SNMP V2C</b>

# UHF RFID Setting Commands

Incorporate UHF RFID commands into new or existing printer programs.

## IMPORTANT:

With all examples make sure the label length matches the physical length of the installed media.

## RFID ON/OFF

### Description

This command is used to enable/disable the RFID encoder module.

### Syntax

#### RFID ON/OFF

<u>Parameter</u>	<u>Description</u>
ON	Enable the RFID encoder module
OFF	Disable the RFID encoder module

### Example

#### RFID ON

## RFID ERROR

### Description

If an error persists after the specified number of labels are tried, perform this error handling action.

### Syntax

#### RFID ERROR OFF/STOP/OVERSTRIKE

<u>Parameter</u>	<u>Description</u>
OFF	No specific action is taken when a tag fails to be programmed.
STOP	Place printer in Pause mode. The label is discarded and reprinting of the label (if desired) must be initiated from the host. When the error is cleared, the label with the failed tag moves forward until the next label is in position to be printed.
OVERSTRIKE	Each failed label prints with the Overstrike pattern and the form retries on a new label until the Label Retry count is exhausted.

### Example

#### RFID ERROR OVERSTRIKE

# RFID RETRY

## Description

This command is used to set the number of label retries that the RFID encoder will attempt before declaring a fault.

## Syntax

**RFID RETRY #**

<u>Parameter</u>	<u>Description</u>
#	Number of retries (1 – 10)

## Example

**RFID RETRY 2**

## RFID RETRYERROR ON/OFF

### Description

This command is used to set if errors are declared when the Label Retry count is exceeded.

### Syntax

**RFID RETRYERROR ON/OFF**

<u>Parameter</u>	<u>Description</u>
ON	Enable the RFID retry error function
OFF	Disable the RFID retry error function

### Example

**RFID RETRYERROR OFF**

# RFID POSITION

## Description

This command is used to set the how far the RFID tag encoding position of the currently installed tag should be offset from Top of Form. Normally, this value is set automatically by the RFID calibration process and should not be changed.

## Syntax

**RFID POSITION #**                    English system (inch)

**RFID POSITION # mm**                Metric system (mm)

**RFID POSITION # dot**                Dot measurement

<u>Parameter</u>	<u>Description</u>
#	Position of the antenna (inch/ mm/ dot)

## Example

**RFID POSITION 60 dot**

# RFID POWER

## Description

This command is used to set the for optimal tag encoding. Sets the read/write power level to be used in the RFID encoder. Normally, this value is set automatically by the RFID calibration process and should not be changed.

## Syntax

**RFID POWER read,write**

<u>Parameter</u>	<u>Description</u>	
Read	Custom tag read power level setting.	
Write	Custom tag write power level setting.	
Accepted Values:		
	read	write
Alpha-40L	1 – 27	1 – 27
PEX-2000	1 – 30	1 – 30
TH240/TH340	1 – 27	1 – 27
MB241/MB341	1 – 27	1 – 27

## Example

**RFID POWER 12,16**

# RFID COUNTRESET

## Description

This command is used to clear the total/failed tag statistics counters.

## Syntax

**RFID COUNTRESET**

<u>Parameter</u>	<u>Description</u>
N/A	

## Example

**RFID COUNTRESET**

# RFID READ/WRITE

## Description

This command allows you to write or read to a UHF RFID tag.

## Syntax

**RFID a,b,format,offset,size,memory bank,data**

<u>Parameter</u>	<u>Description</u>	
a	<b>WRITE</b> = write to the tag  <b>READ</b> = read the tag	
b	<b>WRITE</b> only	lock password  0 = write without lock. 1 to FFFFFFFF in hex = <u>write and lock</u> the data block to prevent it from being overwritten.
	<b>READ</b> only	unlock password  0 = read without unlock. 1 to FFFFFFFF in hex = <u>read and unlock</u> the data block so it can be overwritten later.
Format	A letter specifying the representation format of the field data.  <b>A</b> = ASCII  <b>H</b> = Hex	
offset	This optional parameter of starting position to do the write/read relative to the start of the mem bank. The position is a word value (16 bits).	
Size	A decimal number specifying the overall bit length of the RFID tag memory bank that will be read starting at the offset position (not necessarily the total bank size).  <b>Note:</b> <ul style="list-style-type: none"> <li>- When using <b>WRITE</b>, if the "size" is larger than the "data", it will be padded with 0 in front of the data to read.</li> <li>- When using <b>READ</b>, if the "size" is larger than the <b>WRITE</b> "data", it will be padded with 0 in back of the data to read.</li> <li>- Refer to the example 3 of sample code.</li> </ul>	
memory bank	<b>EPC</b>	EPC 12 bytes data area
	<b>TID</b>	Tag identification 8 bytes area (currently not applicable for RFID WRITE)
	<b>USR</b>	User 32 bytes area
	<b>ACS</b>	4 bytes access code area

	<b>KIL</b>	4 bytes kill code area
	<b>PC</b>	2 bytes PC code area (Gen 2 tags only)
data	<b>WRITE</b>	= content of data string
	<b>READ</b>	= [prompt of data]
	<b>Note:</b>	
	-	RFID WRITE supported "string" or basic variable (e.g. VAR\$)
	-	[ ] = Optional parameter

### Example

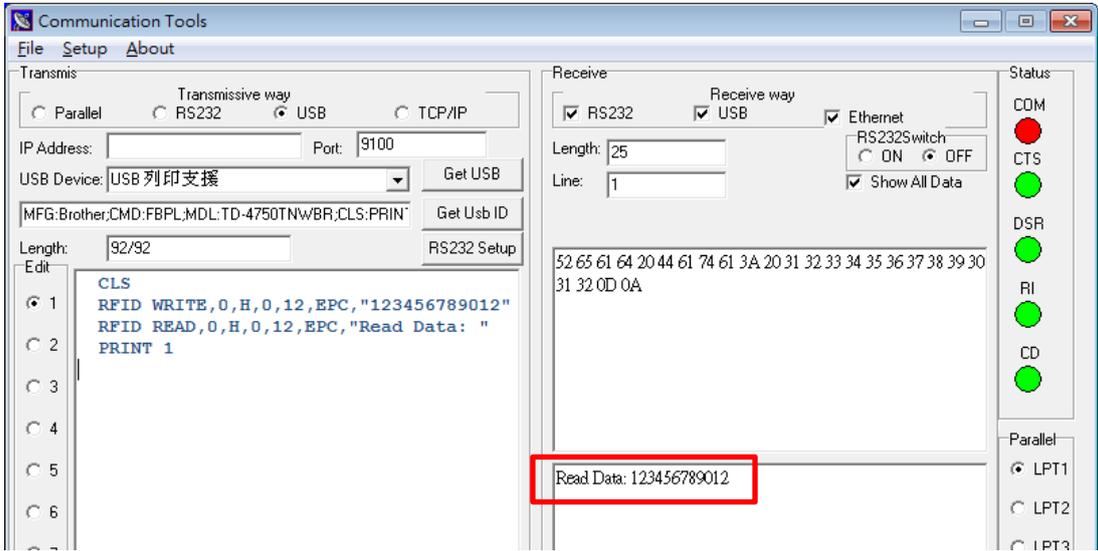
#### Example 1

This programming example writes a data into an RFID tag and reads the written data with a prompt.

```

CLS
RFID WRITE,0,H,0,12,EPC,"123456789012"
RFID READ,0,H,0,12,EPC,"Read Data: "
PRINT 1

```



## Example 2

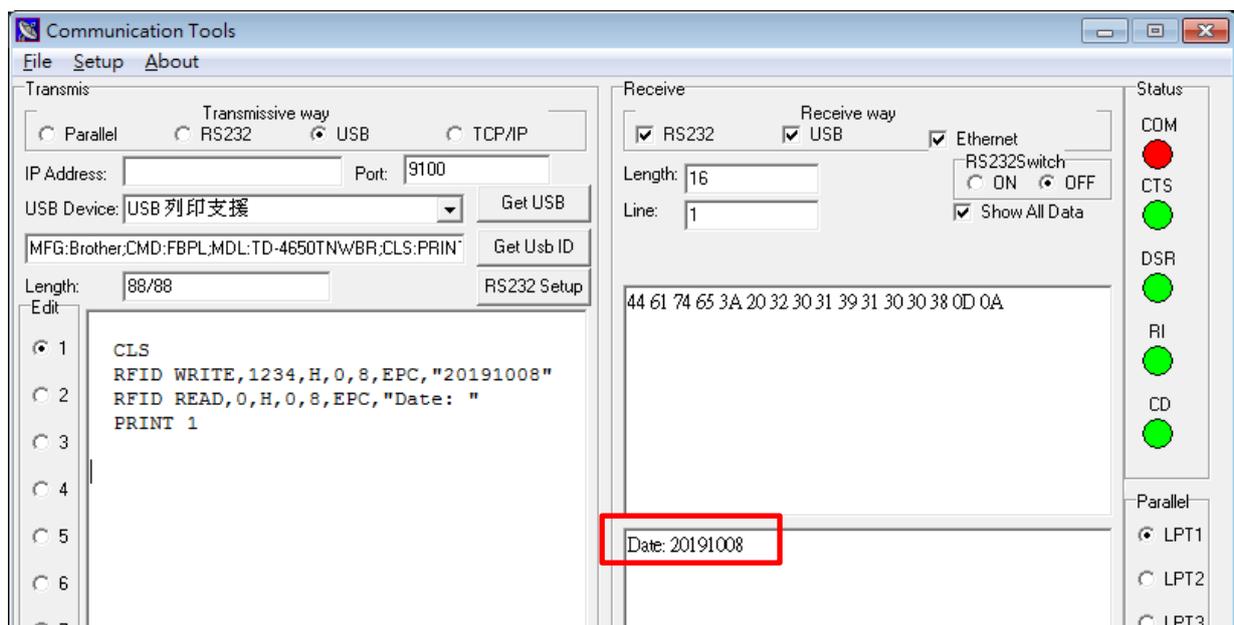
This programming example writes a data with lock password into an RFID tag and reads the written data with a prompt.

**CLS**

**RFID WRITE,1234,H,0,8,EPC,"20191008"**

**RFID READ,0,H,0,8,EPC,"Date: "**

**PRINT 1**



For this locked RFID tag, it cannot be overwritten data without using RFID READ unlock password command. If you re-send the RFID WRITE command, the printer LCD will be shown as below,



If you need to overwrite this locked tag, please use RFID READ unlock command as following programming example, to unlock password for the RFID tag so it can be overwritten later.

**CLS**

**RFID READ,1234,H,0,8,EPC,"Date: "**

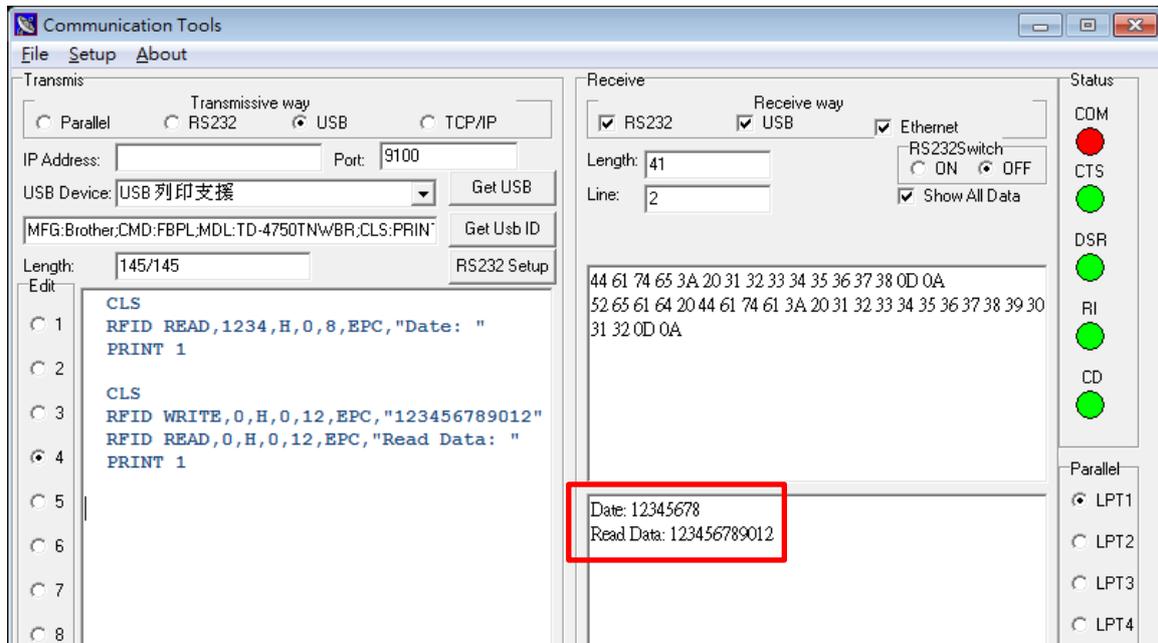
PRINT 1

CLS

RFID WRITE,0,H,0,12,EPC,"123456789012"

RFID READ,0,H,0,12,EPC,"Read Data: "

PRINT 1



### Example 3

When using WRITE, if the "size" is larger than the "data", it will be padded with 0 in front of the data to read. When using READ, if the "size" is larger than the WRITE "data", it will be padded with 0 in back of the data to read.

CLS

RFID WRITE,0,H,0,8,EPC,"1234"

RFID READ,0,H,0,8,EPC,"Read Data: "

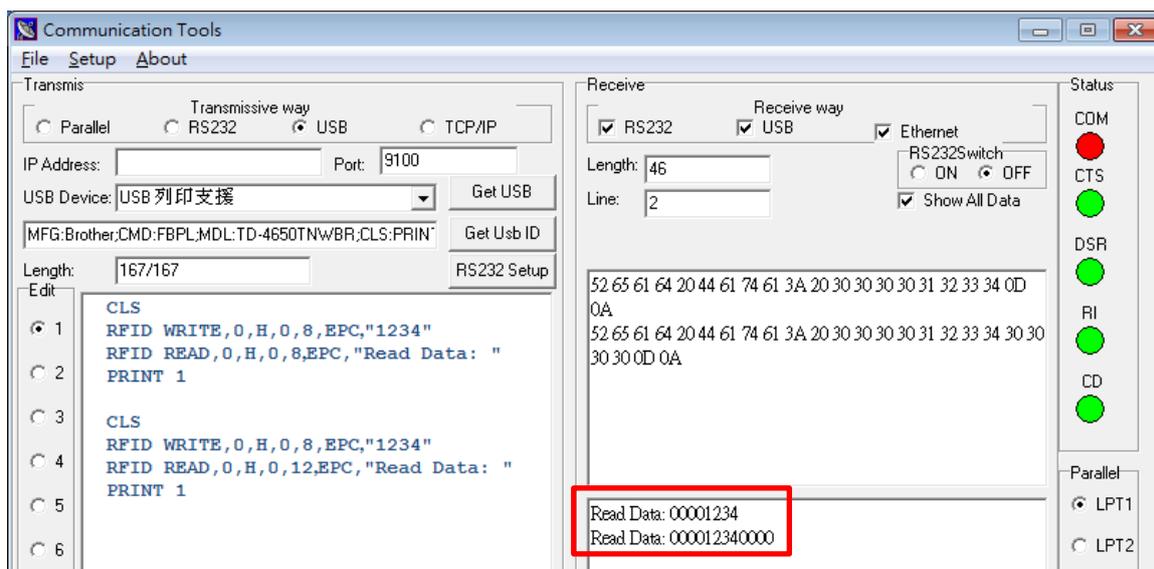
PRINT 1

CLS

RFID WRITE,0,H,0,8,EPC,"1234"

RFID READ,0,H,0,12,EPC,"Read Data: "

PRINT 1



### Example 4 (EPC & USR with Lock)

CLS

RFID WRITE,12345678,H,0,12,EPC,"123456789012"

RFID WRITE,12345678,H,0,12,USR,"987654321012"

RFID READ,12345678,H,0,12,EPC,"EPC : "

RFID READ,12345678,H,0,12,USR,"USR : "

PRINT 1

#### Example 5 (EPC & USR & ACS with Lock)

```
CLS
RFID WRITE,12345678,H,0,12,EPC,"123456789012"
RFID WRITE,12345678,H,0,12,USR,"987654321012"
RFID WRITE,12345678,H,0,8,ACS,"12345678"
RFID READ,12345678,H,0,8,ACS,"ACS : "
RFID READ,12345678,H,0,12,EPC,"EPC : "
RFID READ,12345678,H,0,12,USR,"USR : "
PRINT 1
```

#### Example 6 (EPC & USR & ACS & KIL with Lock)

```
CLS
RFID WRITE,12345678,H,0,12,EPC,"123456789012"
RFID WRITE,12345678,H,0,12,USR,"987654321012"
RFID WRITE,12345678,H,0,8,ACS,"12345678"
RFID WRITE,12345678,H,0,8,KIL,"12345678"
RFID READ,12345678,H,0,8,ACS,"ACS : "
RFID READ,12345678,H,0,8,KIL,"KIL : "
RFID READ,12345678,H,0,12,EPC,"EPC : "
RFID READ,12345678,H,0,12,USR,"USR : "
PRINT 1
```

#### Example 7 (PC+EPC)

```
CLS
RFID WRITE,0,H,0,4,PC,"3400"
RFID WRITE,0,H,0,24,EPC,"123456789012345678901234"
RFID READ,0,H,0,24,EPC,"EPC: "
RFID READ,0,H,0,4,PC,"PC: "
PRINT 1
```

# RFID RFLOCK

## Description

Perform different types of RFID lock operations on available RFID data fields.

## Syntax

**RFID RFLOCK,Type[,Field][,BlockStart,BlockQuantity][,Format,Passcode]**

<u>Parameter</u>	<u>Description</u>								
Type	LOCK, UNLOCK, PERMALOCK, PERMAUNLOCK, PERMABLOCK, PERMACHIP								
Field	<table border="1"><tr><td>EPC</td><td>Electronic Product Code memory bank</td></tr><tr><td>USR</td><td>User memory bank</td></tr><tr><td>ACS</td><td>Access Password</td></tr><tr><td>KIL</td><td>Kill Password</td></tr></table>	EPC	Electronic Product Code memory bank	USR	User memory bank	ACS	Access Password	KIL	Kill Password
	EPC	Electronic Product Code memory bank							
	USR	User memory bank							
	ACS	Access Password							
KIL	Kill Password								
BlockStart	USR Block Permalock Start address								
BlockQuantity	USR Block Permalock Quantity								
Format	A, H								
Passcode	The value of the passcode for the lock operations. The size for the passcode is 32 bits. <b>Note: The passcode must be non-zero when "Type" is LOCK or UNLOCK.</b>								

## Example

<b>Sample code 1</b>
Write the password as 12345678 to lock the EPC memory bank.  <pre>CLS RFID RFLOCK,LOCK,EPC,,,,H,12345678 PRINT 1</pre>
<b>Sample code 2</b>
Write the password as 12345678 to unlock the EPC memory bank.  <pre>CLS RFID RFLOCK,UNLOCK,EPC,,,,H,12345678 PRINT 1</pre>

### Sample code 3

Write the password as 12345678 to permanently lock the EPC memory bank.

```
CLS  
RFID RFLOCK,PERMALOCK,EPC,,,,H,12345678  
PRINT 1
```

### Sample code 4

Write the password as 12345678 to permanently unlock the EPC memory bank.

```
CLS  
RFID RFLOCK,PERMAUNLOCK,EPC,,,,H,12345678  
PRINT 1
```

### Sample code 5

Write the password as 12345678 to lock the USR memory bank.

```
CLS  
RFID RFLOCK,LOCK,,USR,,,H,12345678  
PRINT 1
```

### Sample code 6

Write the password as 12345678 to lock the ACS memory bank.

```
CLS  
RFID RFLOCK,LOCK,,ACS,,H,12345678  
PRINT 1
```

### Sample code 7

Write the password as 12345678 to lock the EPC and USR memory bank.

```
CLS  
RFID RFLOCK,LOCK,EPC,USR,,,H,12345678  
PRINT 1
```

### Sample code 8

Write the password as 12345678 to lock the EPC, USR, and ACS memory bank.

```
CLS  
RFID RFLOCK,LOCK,EPC,USR,ACS,,H,12345678  
PRINT 1
```

### Sample code 9 (PERMACHIP)

```
CLS  
RFID RFLOCK,PERMACHIP  
PRINT 1
```

### Sample code 10 (USR PERMABLOCK)

```
CLS  
RFID RFLOCK,PERMABLOCK,1,2,H,12345678  
PRINT 1
```

CLS  
RFID RFLOCK,PERMABLOCK,0,3,H,12345678  
PRINT 1

**Sample code 11 (EPC)**

CLS  
RFID RFLOCK,LOCK,EPC,,,,H,12345678  
PRINT 1

CLS  
RFID RFLOCK,UNLOCK,EPC,,,,H,12345678  
PRINT 1

CLS  
RFID RFLOCK,PERMALOCK,EPC,,,,H,12345678  
PRINT 1

CLS  
RFID RFLOCK,PERMAUNLOCK,EPC,,,,H,12345678  
PRINT 1

**Sample code 12 (USR)**

CLS  
RFID RFLOCK,LOCK,,USR,,,H,12345678  
PRINT 1

CLS  
RFID RFLOCK,LOCK,EPC,USR,,,H,12345678  
PRINT 1

CLS  
RFID RFLOCK,LOCK,EPC,USR,ACS,,H,12345678  
PRINT 1

**Sample code 13 (ACS)**

CLS  
RFID RFLOCK,LOCK,,,ACS,,H,12345678  
PRINT 1

# RFIDDETECT

## Description

This command will calibrate the RFID and determine the position of the RFID chip. (Since A2.16)

## Syntax

RFIDDETECT

<u>Parameter</u>	<u>Description</u>
N/A	

## Example

RFIDDETECT

# HF RFID Setting Commands

Incorporate HF RFID commands into new or existing printer programs.

## IMPORTANT:

With all examples make sure the label length matches the physical length of the installed media.

## RFID ON/OFF

### Description

This command is used to enable/disable the RFID encoder module.

### Syntax

#### RFID ON/OFF

<u>Parameter</u>	<u>Description</u>
ON	Enable the RFID encoder module
OFF	Disable the RFID encoder module

### Example

#### RFID ON

## RFID ERROR

### Description

If an error persists after the specified number of labels are tried, perform this error handling action.

### Syntax

#### RFID ERROR OFF/STOP/OVERSTRIKE

<u>Parameter</u>	<u>Description</u>
OFF	No specific action is taken when a tag fails to be programmed.
STOP	Place printer in Pause mode. The label is discarded and reprinting of the label (if desired) must be initiated from the host. When the error is cleared, the label with the failed tag moves forward until the next label is in position to be printed.
OVERSTRIKE	Each failed label prints with the Overstrike pattern and the form retries on a new label until the Label Retry count is exhausted.

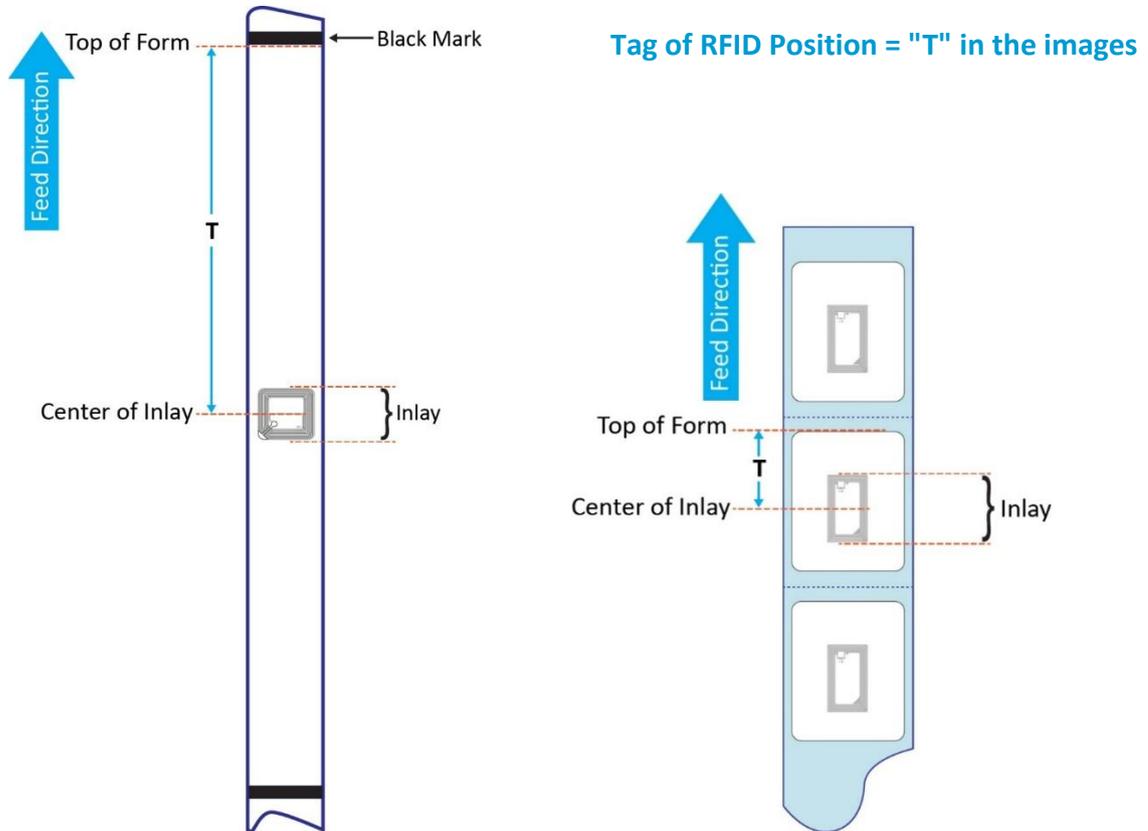
### Example

#### RFID ERROR OVERSTRIKE

# RFID POSITION

## Description

This command is used to set the how far the RFID tag encoding position of the currently installed tag should be offset from Top of Form. Use a ruler to manually measure the distance from the Top of Form (TOF) to the center of the inlay. This value is the RFID Position.



## Syntax

- RFID POSITION #** English system (inch)
- RFID POSITION # mm** Metric system (mm)
- RFID POSITION # dot** Dot measurement

<u>Parameter</u>	<u>Description</u>
#	Position of the antenna (inch/ mm/ dot)

## Example

**RFID POSITION 60 dot**

# RFID READ/WRITE

## Description

This command allows you to write or read to a HF RFID (13.56 MHz) tag. Please note that the RFID POSITION must be set before performing any read/write operations.

## Syntax

**RFID a,b,c,d,format,offset,size,memory bank,data**

<u>Parameter</u>	<u>Description</u>										
a	<b>WRITE</b> = write to the tag <b>READ</b> = read the tag										
b	HF RFID tag types (13.56 MHz) <table border="1"><tr><td><b>15693</b></td><td>Vicinity</td></tr><tr><td><b>14443A</b></td><td>Proximity</td></tr><tr><td><b>FELICA</b></td><td>Proximity</td></tr></table>	<b>15693</b>	Vicinity	<b>14443A</b>	Proximity	<b>FELICA</b>	Proximity				
<b>15693</b>	Vicinity										
<b>14443A</b>	Proximity										
<b>FELICA</b>	Proximity										
c	IC Chip, Set to 0 for now, reserved for future use.										
d	<table border="1"><tr><td>WRITE (only)</td><td><b>0</b> = write without lock. <b>1</b> = write and lock</td></tr><tr><td>READ</td><td>no effect</td></tr></table>	WRITE (only)	<b>0</b> = write without lock. <b>1</b> = write and lock	READ	no effect						
WRITE (only)	<b>0</b> = write without lock. <b>1</b> = write and lock										
READ	no effect										
Format	A letter specifying the representation format of the field data. <b>A</b> = ASCII <b>H</b> = Hex										
offset	This optional parameter of starting position to do the write/read relative to the start of the mem bank. The position is a word value (16 bits). (UID always set 0)										
Size	A decimal number specifying the overall bit length of the RFID tag memory bank that will be read starting at the offset position (UID always set 0).										
memory bank	<table border="1"><tr><td><b>UID</b></td><td>Unique Identifier</td></tr><tr><td><b>USR</b></td><td>User Memory (Variable size)</td></tr><tr><td><b>AFI</b></td><td>1 byte AFI code (15693 only)</td></tr><tr><td><b>DSFID</b></td><td>1 byte DSFID code (15693 only)</td></tr><tr><td><b>EAS</b></td><td>1 bit EAS operation (15693 only)</td></tr></table>	<b>UID</b>	Unique Identifier	<b>USR</b>	User Memory (Variable size)	<b>AFI</b>	1 byte AFI code (15693 only)	<b>DSFID</b>	1 byte DSFID code (15693 only)	<b>EAS</b>	1 bit EAS operation (15693 only)
<b>UID</b>	Unique Identifier										
<b>USR</b>	User Memory (Variable size)										
<b>AFI</b>	1 byte AFI code (15693 only)										
<b>DSFID</b>	1 byte DSFID code (15693 only)										
<b>EAS</b>	1 bit EAS operation (15693 only)										
data	<b>WRITE</b> = content of data string <b>READ</b> = [prompt of data]  <b>Note:</b> - RFID WRITE supported "string" or basic variable (e.g. VAR\$) - [ ] = Optional parameter										

## Example

### Example

This programming example writes a data into an RFID tag and reads the written data with a prompt.

CLS

TEXT 50,10,"0",0,12,12,"15693 USR HEX 64 ASCII 32"

RFID

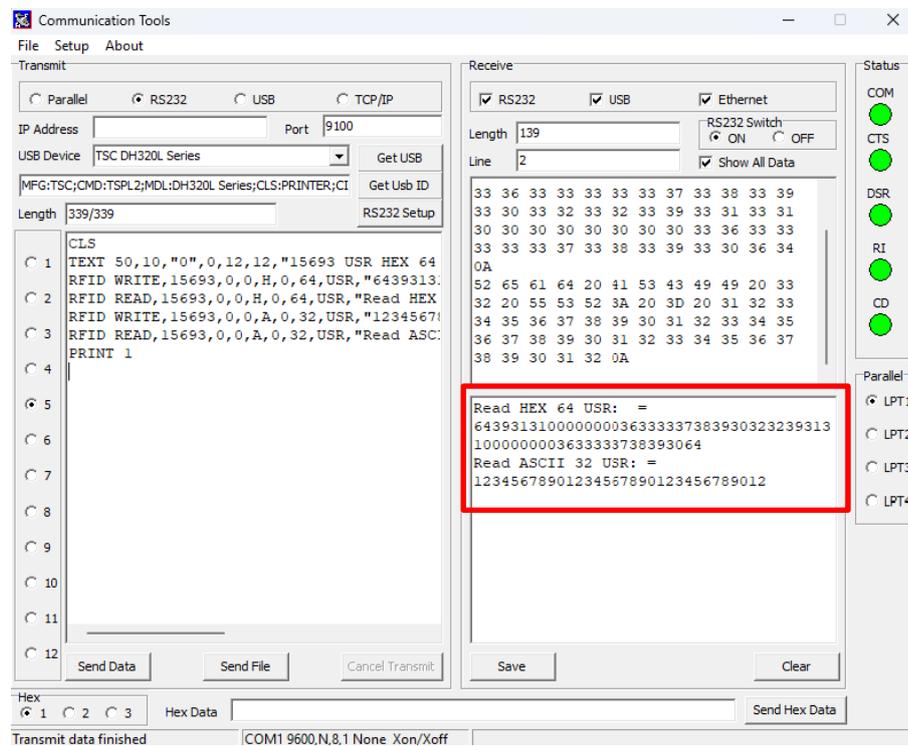
WRITE,15693,0,0,H,0,64,USR,"64393131000000036333337383930323239313100000003633333738393064"

RFID READ,15693,0,0,H,0,64,USR,"Read HEX 64 USR: "

RFID WRITE,15693,0,0,A,0,32,USR,"12345678901234567890123456789012"

RFID READ,15693,0,0,A,0,32,USR,"Read ASCII 32 USR:"

PRINT 1



# 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.

### Syntax

#### NFC FEATURE

<u>Parameter</u>	<u>Description</u>
None	N/A

<u>Return Information</u>	
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
<code>NFC FEATURE</code>	Example by CommTool: <pre>present</pre>

## NFC STATUS

### Description

This command is used to return current status of the NFC reader or status of last operation completed.

### Syntax

#### NFC STATUS

<u>Parameter</u>	<u>Description</u>
None	N/A

<u>Return Information</u>	
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>	Example by CommTool: <pre>in progress</pre>

# 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**

<u>Parameter</u>	<u>Description</u>
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

<u>Parameter</u>	<u>Description</u>
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"**

<u>Parameter</u>	<u>Description</u>
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

<u>Parameter</u>	<u>Description</u>
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	
<u>Write data to tag once</u>  NFC MODE OFF NFC TIMEOUT 3 NFC WRITE "Test"	<u>Continue to write data to tag</u>  NFC MODE WRITE NFC WRITE "123456789"
<u>Read data from tag once</u>  NFC MODE OFF NFC TIMEOUT 3 NFC READ	<u>Continue to read data from Tag</u>  NFC MODE READ NFC READ

# Alpha-2R/TDM Series Setting Commands

## SET PRINTQUALITY

### Description

This command is used to set the print mode (print quality) for Alpha-2R and TDM series printer.

(Supported device: Alpha-2R and TDM series only)

### Syntax

**SET PRINTQUALITY DRAFT/STANDARD/OPTIMUM**

<u>Parameter</u>	<u>Description</u>
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
<pre>SET PRINTQUALITY DRAFT SET PRINTQUALITY STANDARD SET PRINTQUALITY OPTIMUM</pre>

# SET STANDBYTIME

## Description

This command is used to set the standby time for Alpha-2R and TDM series printer.

(Supported device: Alpha-2R and TDM series only)

## Syntax

**SET STANDBYTIME OFF/XXXXX**

<u>Parameter</u>	<u>Description</u>
OFF	Disable
XXXXX	0 ~ 65534 (second)

*Note:*  
The default value is "SET STANDBYTIME 120".

## Example

Sample code
<pre>SET STANDBYTIME OFF SET STANDBYTIME 480</pre>

# SET SLEEPTIME

## Description

This command is used to set the sleeping time for Alpha-2R and TDM series printer.

(Supported device: Alpha-2R and TDM series only)

## Syntax

**SET SLEEPTIME OFF/XXXXX**

<u>Parameter</u>	<u>Description</u>
OFF	Disable
XXXXX	0 ~ 65534 (minute)

*Note:*  
The default value is "SET SLEEPTIME 30".

## Example

```
Sample code

SET SLEEPTIME OFF
SET SLEEPTIME 20
```

# GPIO Setting Commands

## SET GPO

### Description

Use this command to send out the GPIO signals by the printer.

### Syntax

**SET GPO***n* signal state, delay0, pulse0, delay1, pulse1, function condition

<u>Parameter</u>	<u>Description</u>								
n	n = 1 ~ 7  Seven dedicated outputs are available for the desired function conditions.								
Signal state	<table border="1"><tr><td><b>HIGH</b></td><td>Goes the high level signal when the following function condition is detected.</td></tr><tr><td><b>LOW</b></td><td>Goes the low level signal when the following function condition is detected.</td></tr><tr><td><b>POS</b></td><td>Goes the positive pulse signal when the following function condition is detected.</td></tr><tr><td><b>NEG</b></td><td>Goes the negative pulse signal when the following function condition is detected.</td></tr></table>	<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.
<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.								
Function condition (warning, error,	<table border="1"><tr><td><b>FAULT</b></td><td>Printer fault.</td></tr></table>	<b>FAULT</b>	Printer fault.						
<b>FAULT</b>	Printer fault.								

control...)	<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.	
	<b>FORWARD</b>	Paper is moving forward.	
	<b>BACKWARD</b>	Paper is moving backward.	
	<b>LOWLABEL</b>	Low label is occurred.	Since A2.09
	<b>LOWRIBBON</b>	Low ribbon is occurred.	

## Example

Sample code
<b>SET GPO1 HIGH,100,0,100,0,FAULT RIBBON</b>
<b>SET GPO2 LOW,100,0,100,0,FAULT PAPER</b>
<b>SET GPO3 POS,100,100,100,100,PAUSE</b>
<b>SET GPO4 NEG,100,50,100,50,IDLE</b>
<b>SET GPO1 LOW,0,0,0,0,FORWARD</b>
<b>SET GPO2 LOW,0,0,0,0,BACKWARD</b>

# SET GPI

## Description

Use this command to receive the GPIO signals from external controlling devices.

## Syntax

SET GPI n signal, pulse, function

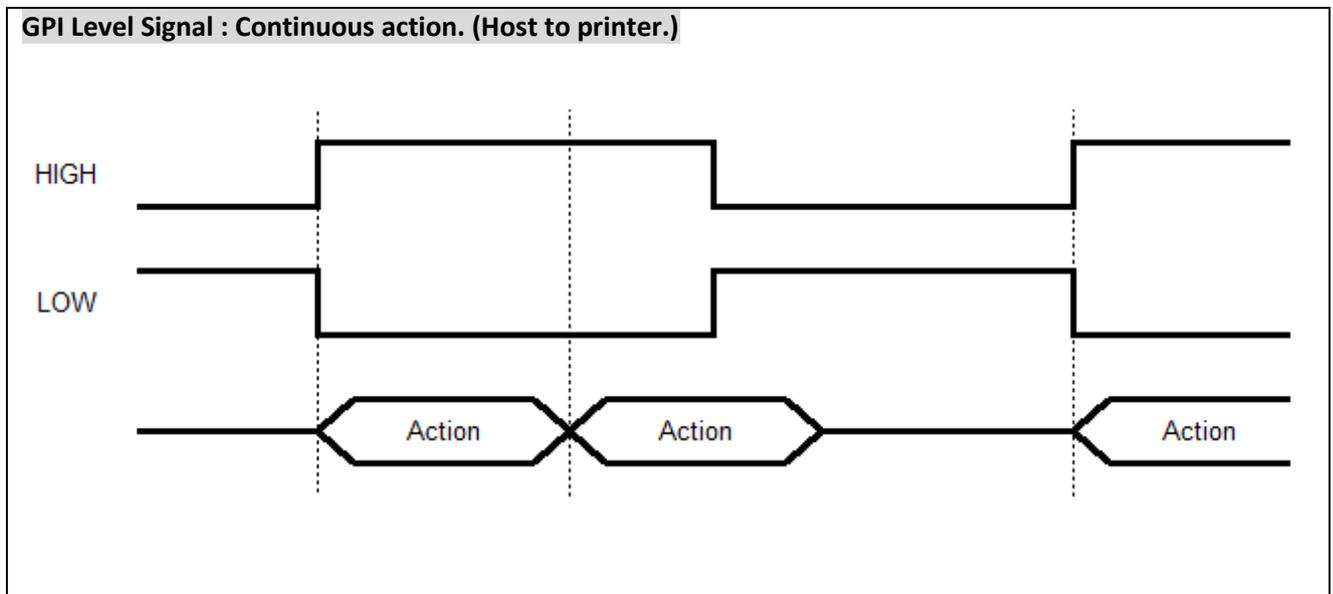
<u>Parameter</u>	<u>Description</u>																		
n	n = 1 ~ 4  Four dedicated inputs are available for the desired control functions.																		
Signal state	<table border="1"><tr><td><b>HIGH</b></td><td>When a high level signal received, will activate the following printer control functions.</td></tr><tr><td><b>LOW</b></td><td>When a low level signal received, will activate the following printer control functions.</td></tr><tr><td><b>POS</b></td><td>When a positive pulse signal received, will activate the following printer control functions.</td></tr><tr><td><b>NEG</b></td><td>When a negative pulse signal received, will activate the following printer control functions.</td></tr></table>	<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.										
<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)	<table border="1"><tr><td><b>PAUSE</b></td><td>Toggle pause status.</td></tr><tr><td><b>PAUSE ON</b></td><td>Enter pause status.</td></tr><tr><td><b>PAUSE OFF</b></td><td>Cancel pause status.</td></tr><tr><td><b>PRINT</b></td><td>Print batch of labels.</td></tr><tr><td><b>PRINT n</b></td><td>n is numerical. Specify how many labels to print. Maximum: 32000.</td></tr><tr><td><b>CUT</b></td><td>Cut immediately.</td></tr><tr><td><b>FEED n</b></td><td>n is numerical and the unit is dot. Specify the feeding length. Maximum: 32000.</td></tr><tr><td><b>BACKFEED n</b></td><td>n is numerical and the unit is dot. Specify the backfeeding length. Maximum: 32000.</td></tr><tr><td><b>FORMFEED</b></td><td>Feeding an empty label.</td></tr></table>	<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>	n is numerical. Specify how many labels to print. Maximum: 32000.	<b>CUT</b>	Cut immediately.	<b>FEED n</b>	n is numerical and the unit is dot. Specify the feeding length. Maximum: 32000.	<b>BACKFEED n</b>	n is numerical and the unit is dot. Specify the backfeeding length. Maximum: 32000.	<b>FORMFEED</b>	Feeding an empty label.
<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>	n is numerical. Specify how many labels to print. Maximum: 32000.																		
<b>CUT</b>	Cut immediately.																		
<b>FEED n</b>	n is numerical and the unit is dot. Specify the feeding length. Maximum: 32000.																		
<b>BACKFEED n</b>	n is numerical and the unit is dot. Specify the backfeeding length. Maximum: 32000.																		
<b>FORMFEED</b>	Feeding an empty label.																		

<b>INPUT n</b>	n is text or command. The n will be triggered to printer.
<b>REBOOT</b>	Reboot the printer

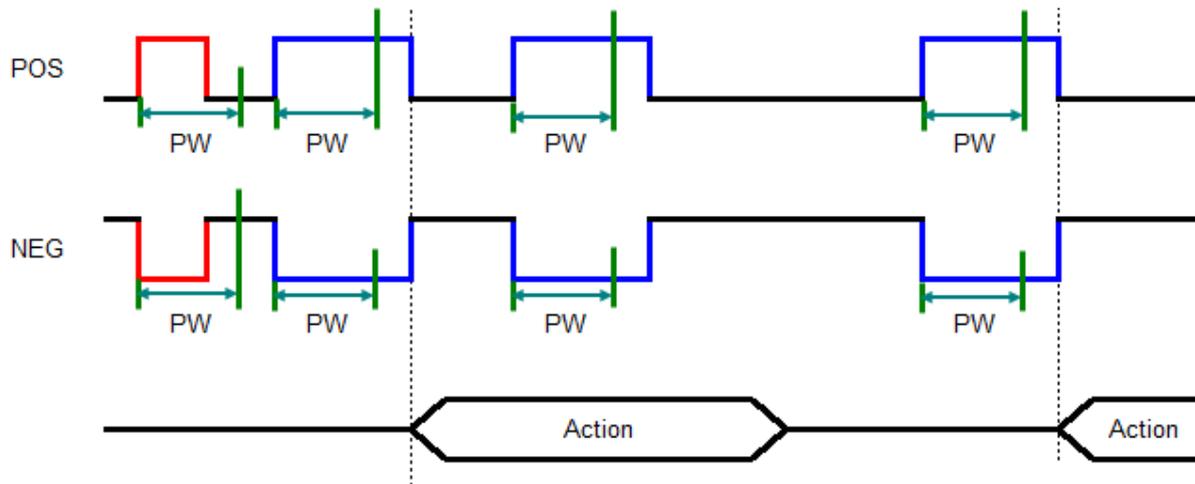
## Example

Sample code
<pre>SET GPI1 HIGH,0,PAUSE SET GPI2 LOW,0,PAUSE ON SET GPI3 POS,100,PAUSE OFF SET GPI4 NEG,100,CUT</pre>
<pre>SET GPI1 NEG,100,INPUT "TEST.BAS"+CHR\$(13)+CHR\$(10)</pre>
<pre>SET GPI1 NEG,100,REBOOT</pre>

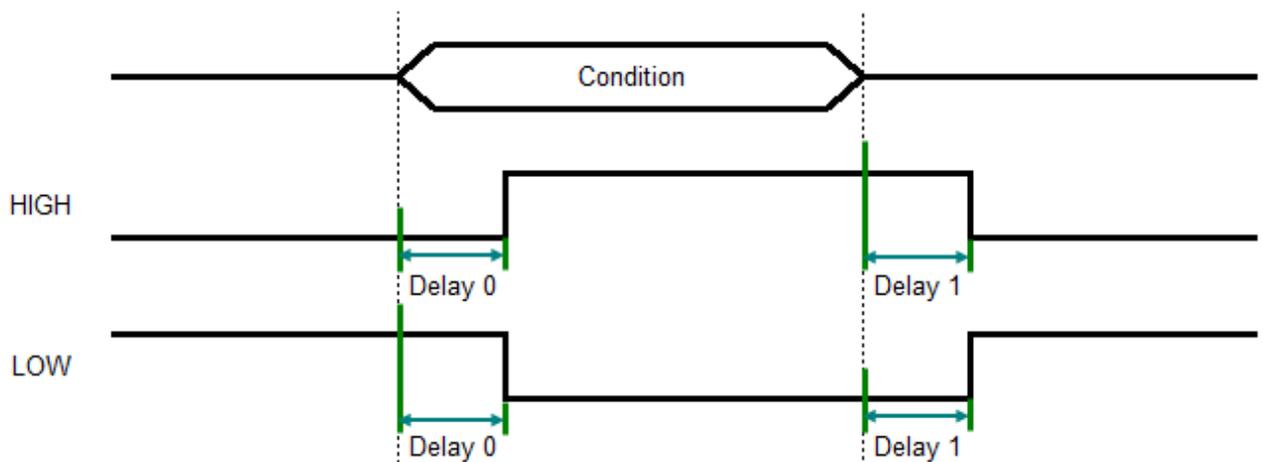
## GPIO Waveform



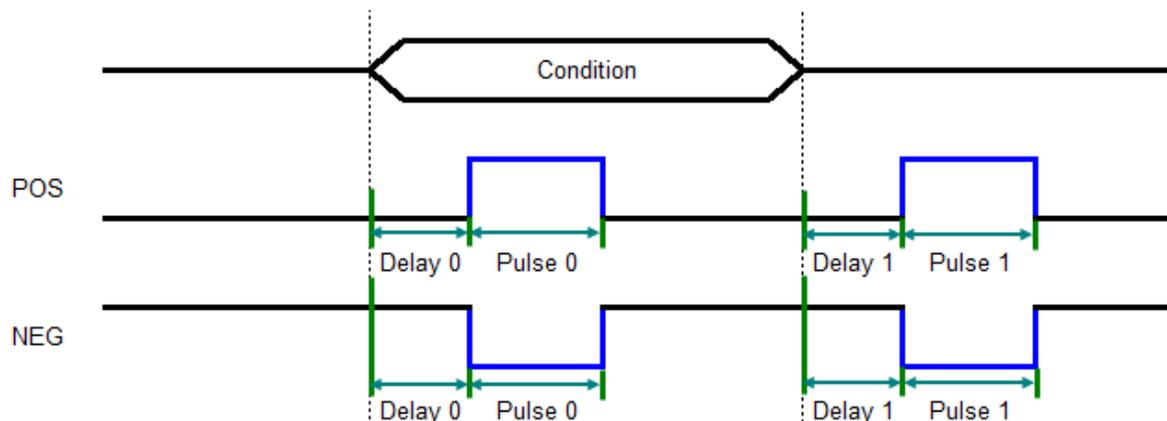
**GPI Pulse Signal : A pulse is an action. (Host to printer.)**



**GPO Level Signal : Continuous condition. (Printer to host.)**



**GPO Pulse Signal : A pulse is a condition. (Printer to host.)**



# GPm

## Description

This command is used to get status of GPI, set status of GPO. (since A2.15.111)

## Syntax

**GPm = n**

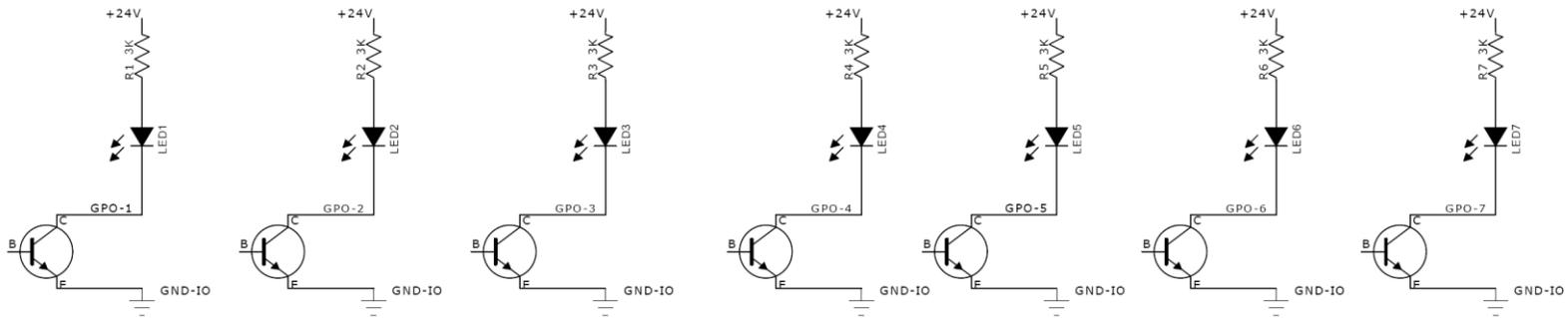
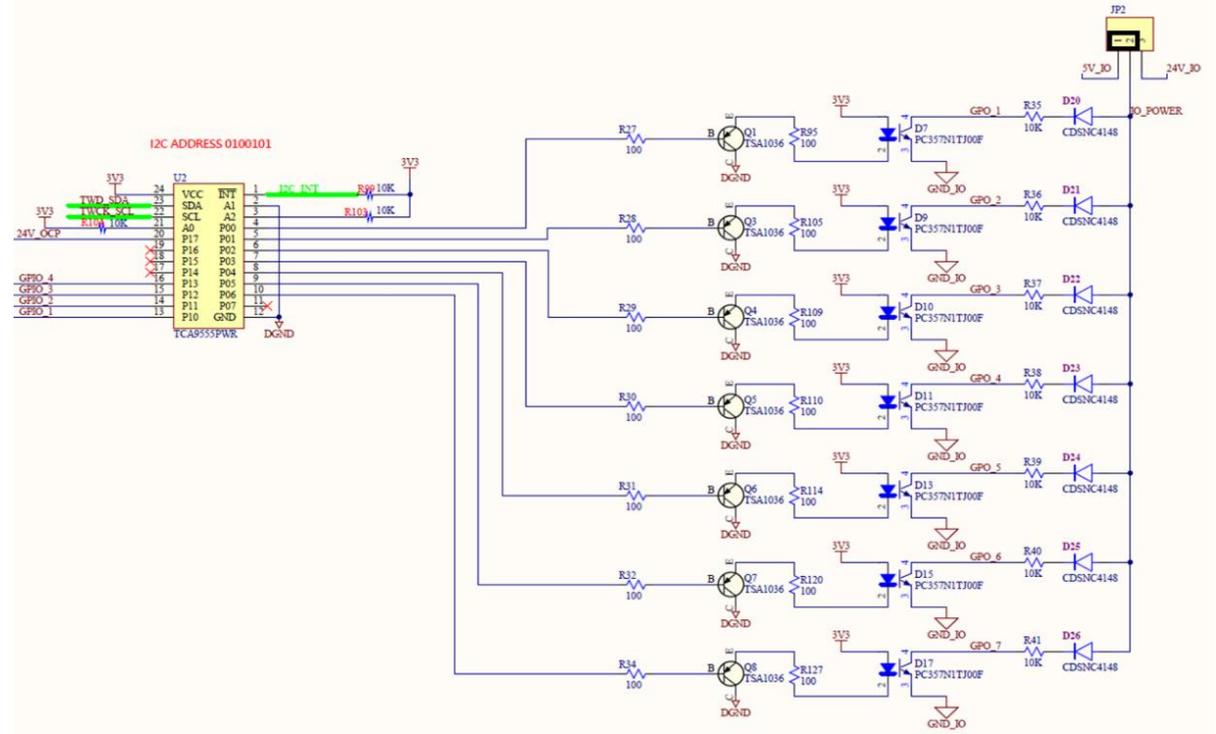
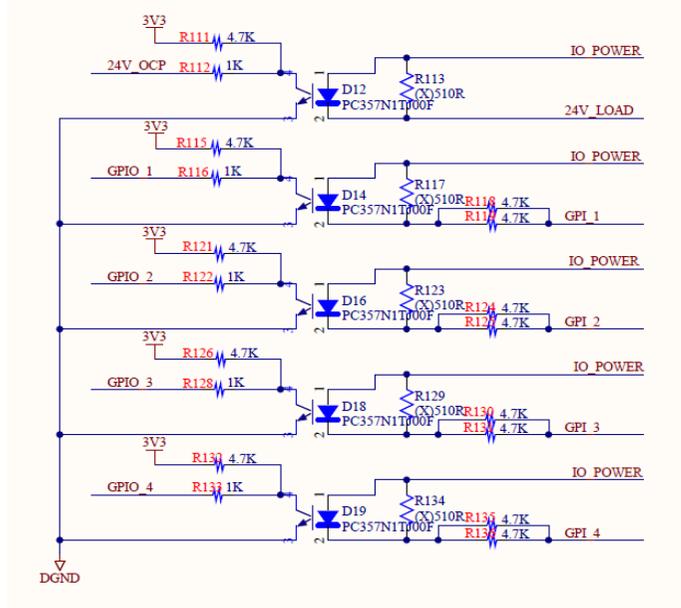
<u>Parameter</u>	<u>Description</u>
m	I, GPI signal O, GPO signal
n	0, represents low lever of signal 1, represents high lever of signal

## Example

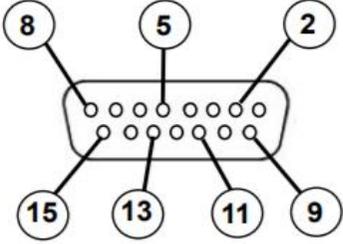
### Sample code

```
IF GPI2 = 1 THEN
    GPO1 = 0
    GPO2 = 1
ELSE
    GPO1 = 1
    GPO2 = 0
END IF
```

# Applicator I/O Interface (DB15F) Circuit Diagram



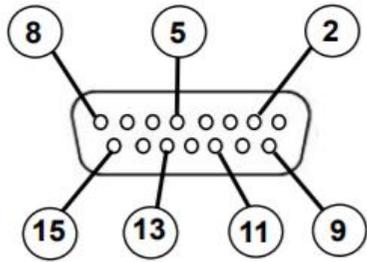
## GPIO Interface Pin Configuration (DB15F)

	Female Connector Front View	
	PIN	CONFIGURATION
	1	GND
	2	5V
	3	GPI_1
	4	GPI_2
	5	GPI_3
	6	GPI_4
	7	24V
	8	GND
	9	GPO_1
	10	GPO_2
	11	GPO_3
	12	GPO_4
	13	GPO_5
	14	GPO_6
	15	GPO_7

Below table's emulation will only be applied when users are using GPIO-DB15 with the **Applicator Port** function **turned on Mode 1/2/3/4**.

Please follow the procedures to turn on the function: **Menu > Interface > GPIO > Applicator Port (Default: Off) > Mode 1/2/3/4**.

PIN	CONFIGURATION	SIGNAL NAME	SIGNAL TYPE
1	GND	I/O SIGNAL GROUND	I/O Signal Ground
2	5V (JP2 short)	I/O SIGNAL POWER	Power
3	GPI_1	PRINT START	Input



**Female Connector Front View**

4	GPI_2	FORMFEED	Input
5	GPI_3	PAUSE	Input
6	GPI_4	REPRINT	Input
7	24V	(+) 24V	Power
8	GND	GROUND	Power Ground
9	GPO_1	LOWRIBBON	Output
10	GPO_2	FAULT	Output
11	GPO_3	PRINT END	Output
12	GPO_4	FAULT PAPER	Output
13	GPO_5	FAULT RIBBON	Output
14	GPO_6	DATA READY	Output
15	GPO_7	Dummy	Output

### GPO pin no. 1~7 application example:

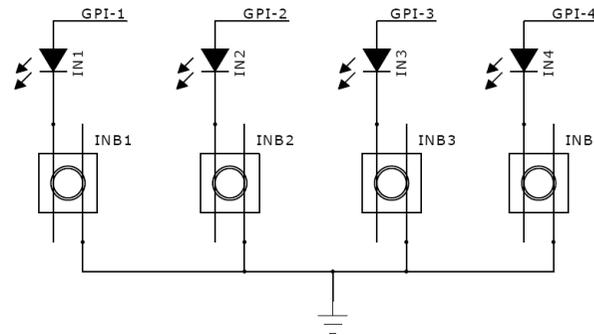
Since we connect GPO pin no. 1~7 with seven individual LED, the output signal from GPO will light the individual LED on or off.

\*NPN output specification.

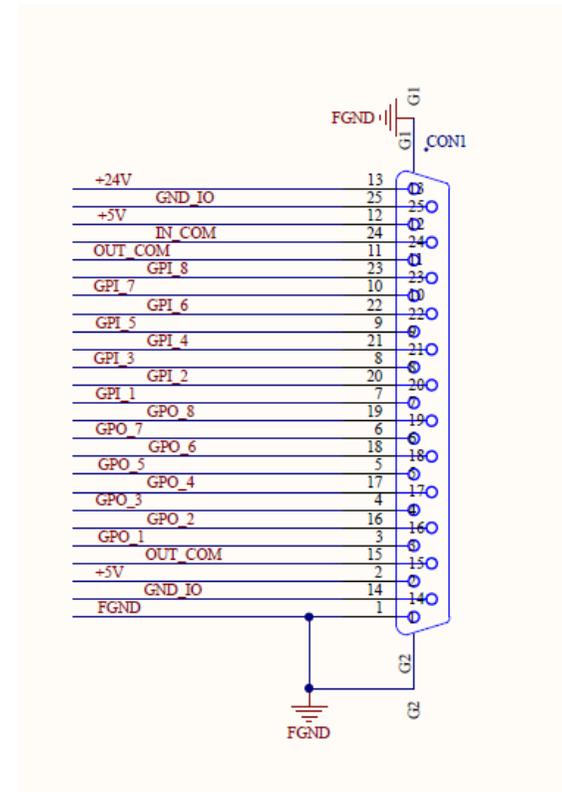
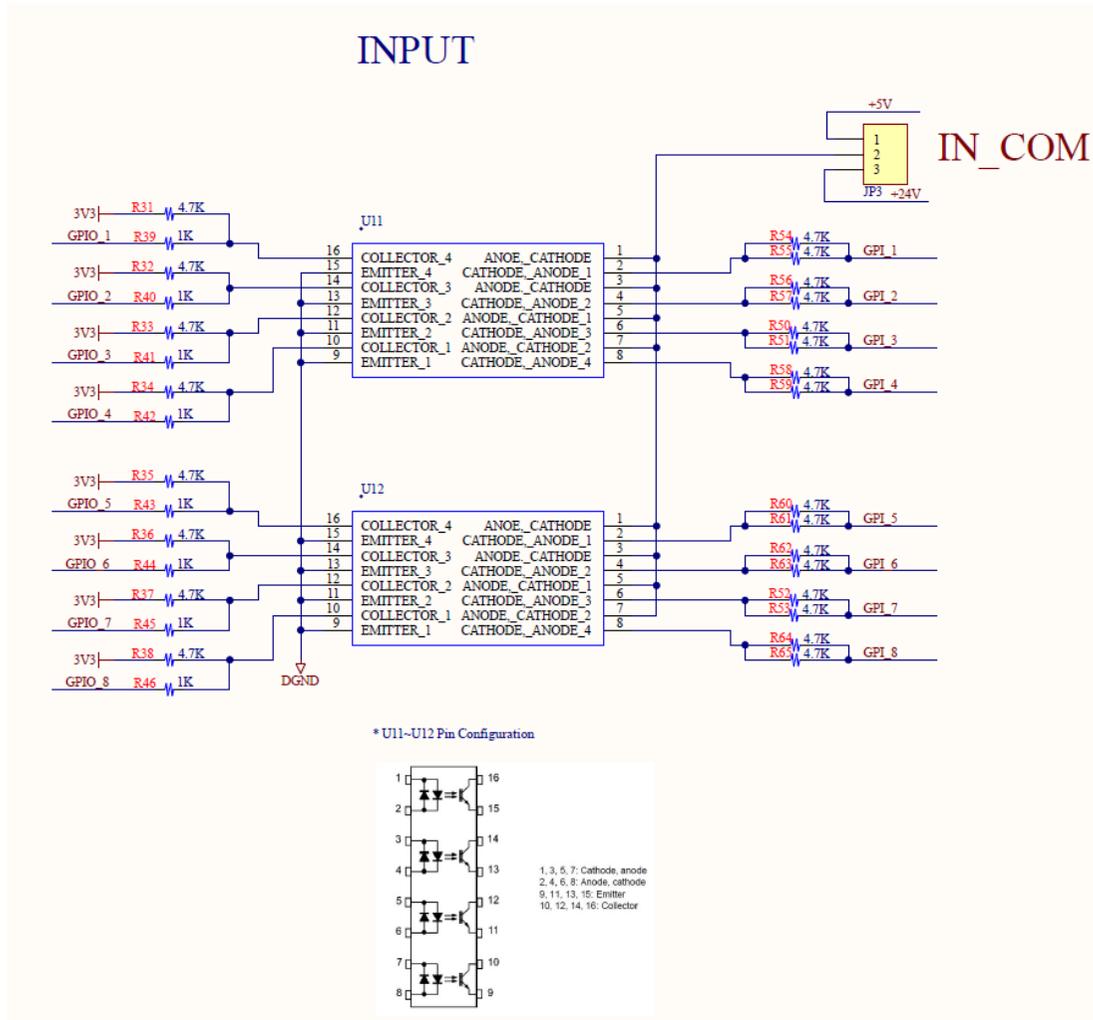
Collector-emitter voltage	$V_{CEO}$	35 V
Emitter-collector voltage	$V_{CEO}$	6 V
Collector current	$I_C$	Max. 50 mA
*1 Collector power dissipation	$P_C$	150 mW

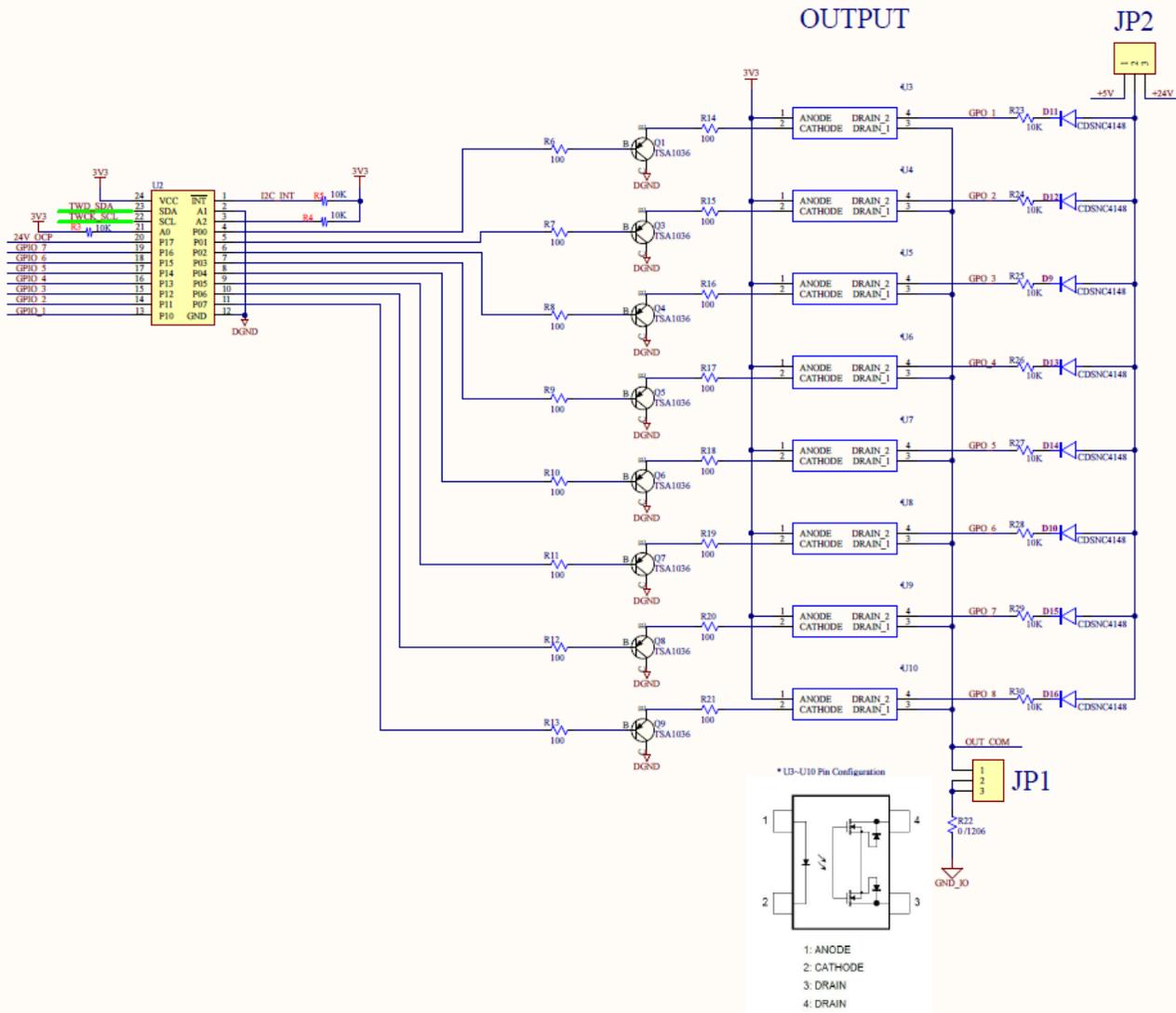
### GPI pin no. 1~4 application example:

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.

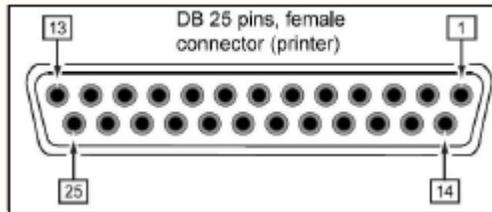


# Applicator I/O Interface (DB25F) Circuit Diagram





## GPIO Interface Pin Configuration (DB25F)

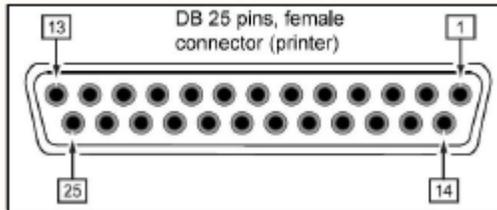


PIN	CONFIGURATION
1	FGND
2	+5V
3	GPO_1
4	GPO_3
5	GPO_5
6	GPO_7
7	GPI_1
8	GPI_3
9	GPI_5
10	GPI_7
11	OUT_COM
12	+5V
13	+24V
14	GND
15	OUT_COM
16	GPO_2
17	GPO_4
18	GPO_6
19	GPO_8
20	GPI_2
21	GPI_4
22	GPI_6

	23	GPI_8
	24	IN_COM
	25	GND

Below table's emulation will only be applied when users are using GPIO-DB25 with the **External Signal** function **turned on Type 1/2/3/4**.

Please follow the procedures to turn on the function: **Menu > Interface > GPIO > External Signal (Default: Off) > Type1/2/3/4**.



PIN	CONFIGURATION	SIGNAL NAME	SIGNAL TYPE	LEVEL
1	FGND	FG ( Frame Ground)	-	-
2	+5V	+5V	-	-
3	GPO_1	Start Print	Output	Low
4	GPO_3	Machine Error	Output	Low
5	GPO_5	Print Completed	Output	Low
6	GPO_7	Online	Output	Low
7	GPI_1	Label Near End	Input	Low
8	GPI_3	Reprint	Input	Low
9	GPI_5	Backfeed Order	Input	Low
10	GPI_7	Reverse	Input	-
11	OUT_COM	OUT_COM	-	-
12	+5V	+5V	-	-
13	+24V	+24V	-	-
14	GND	GND(Signal Ground)	-	-
15	OUT_COM	OUT_COM	-	-
16	GPO_2	Fault Ribbon	Output	Low
17	GPO_4	Fault Paper	Output	Low
18	GPO_6	Low Ribbon	Output	High
19	GPO_8	Reverse	Output	-
20	GPI_2	Start Print	Input	Low

21	GPI_4	Formfeed	Input	Low
22	GPI_6	Reverse	Input	-
23	GPI_8	Reverse	Input	-
24	IN_COM	IN_COM	Input	-
25	GND	GND(SIGNAL GROND)	-	-

## Revision History

Date	Content	Editor
2007/07/13	Revise some typos	Phil
2007/12/25	Revise FREAD\$( ) example	Camille
2008/04/10	Add update history list	Camille
2009/01/17	Add GAPDETECT command	Ken
2009/05/18	Add CIRCLE command	Phil
2009/06/24	Add RSS command	Phil
2010/07/06	Revise bar command section	Camille
2010/10/25	Revise some sections	Camille
2011/01/06	Add CODE 11 barcode	Ernest
2011/01/06	Add AZTEC barcode	Ernest
2011/01/06	Revise sensor intension table in SET GAP command	Ernest
2011/01/10	Add BLINEDETECT command	Ernest
2011/01/10	Add AUTODETECT command	Ernest
2011/01/10	Add BASIC function FORMAT\$( )	Ernest
2011/01/10	Add BASIC function NOW\$( )	Ernest
2011/01/10	Add BASIC function TRIM\$( )	Ernest
2011/01/10	Add BASIC function LTRIM\$( )	Ernest
2011/01/10	Add BASIC function RTRIM\$( )	Ernest
2010/01/10	Add BASIC function STRCOMP( )	Ernest
2010/01/10	Add BASIC function INSTR( )	Ernest
2011/01/25	Modify TSC address	Camille
2011/03/04	Revise, TTP-248M doesn't support mirror function	Ernest
2011/03/04	Add sensor range of TTP-225/ TDP-225 in command SET GAP	Ernest
2011/12/09	Add some command spec in RSS barcode.	Ernest
2012/11/20	Add command SET USBHOST KEYBOARD/SCANNER	Ernest
2012/11/20	Revise STRCOMP( ) example	Ernest
2012/11/20	Revise TRIM\$( ), LTRIM\$( ), RTRIM\$( ) example.	Ernest
2012/11/20	Add <ESC>!P command.	Ernest
2012/11/20	Add <ESC>!O command.	Ernest
2012/11/20	Revise OUT command.	Ernest
2012/11/20	Add SET BLINE command.	Ernest
2012/11/20	Add ELLIPSE command.	Ernest
2012/11/20	Add SET RIBBONEND command.	Ernest
2012/11/20	Add SET ENCODER command.	Ernest
2012/11/21	Revise TEXT command.	Ernest
2012/11/21	Revise speed table of SPEED command.	Ernest
2012/11/21	Revise AZTEC command.	Ernest
2012/11/21	Add BLOCK command.	Ernest
2012/11/21	Add PUT command.	Ernest
2012/11/21	Add GET command.	Ernest
2012/11/21	Add INP( ) command.	Ernest
2012/11/22	Revise PUTBMP command. Support grayscale printing in direct thermal printing.	Ernest
2012/11/22	Revise BARCODE command. New support barcode TELEPEN, TELEPENN, PLANET, CODE49, DPI, DPL.	Ernest
2012/11/23	Add TLC39 barcode.	Ernest
2012/11/23	Add CODABLOCK command (F mode only).	Ernest
2012/11/23	Add SELFTEST PATTERN command.	Ernest
2012/11/23	Revise the supported CODEPAGE table and example.	Ernest
2012/11/23	Add global variable NOW.	Ernest
2012/11/26	Add DATEADD( ) command.	Ernest
2012/11/26	Add SET AUTORUN command.	Ernest
2012/11/26	Add LOC( ) command.	Ernest
2012/11/26	Add NOW\$( ) command.	Ernest
2012/11/26	Revise SET RIBBON command.	Ernest
2012/11/26	Revise SET COUNTER command.	Ernest
2012/11/26	Add <ESC>!C command.	Ernest
2012/11/26	Add <ESC>!Q command.	Ernest
2012/11/26	Add <ESC>!S command.	Ernest
2012/11/26	Add OUTF command.	Ernest
2012/11/26	Add <ESC>!D command.	Ernest
2012/11/26	Add ~!E command.	Ernest
2012/11/27	Add LOB( ) command.	Ernest
2012/11/27	Add WHILE ... WEND command.	Ernest
2012/11/27	Add DO ... LOOP command.	Ernest

2012/11/27	Add TEXTPIXEL() command.	Ernest
2012/11/27	Add BARCODEPIXEL() command.	Ernest
2012/11/27	Add GETSENSOR() command.	Ernest
2012/11/27	Add GETSETTING() command.	Ernest
2012/11/28	Revise SET CUTTER command.	Ernest
2012/11/28	Revise OPEN command.	Ernest
2012/11/28	Revise FOR ... NEXT LOOP command.	Ernest
2012/11/28	Add CLOSE command.	Ernest
2012/11/28	Add COPY command.	Ernest
2012/11/28	Add MPDF417 command for Micro PDF 417 barcode.	Ernest
2012/11/30	Add EOJ command.	Ernest
2012/11/30	Add DELAY command.	Ernest
2012/11/30	Add DISPLAY command.	Ernest
2012/11/30	Add XOR\$() command.	Ernest
2012/11/30	Add _MODEL\$ variable.	Ernest
2012/11/30	Add _SERIAL\$ variable.	Ernest
2012/11/30	Add _VERSION\$ variable.	Ernest
2012/11/30	Revise LIMITFEED command.	Ernest
2012/11/30	Revise BOX command.	Ernest
2012/11/30	Add SET FEED_LEN command.	Ernest
2012/12/20	Add external Wi-Fi module setting commands.	Ernest
2012/12/20	Add Ethernet setting commands.	Ernest
2012/12/24	Revise DMATRIX command.	Ernest
2012/12/24	Revise LIMITFEED command.	Ernest
2012/12/24	Revise SELFTEST command	Camille
2013/2/5	Add sample result for each section	Camille
2013/2/6	Add CODEPAGE 864 (Arabic) ; since F/W V7.0	Camille
2013/2/26	Add <ESC>!F command.	Camille
2013/2/26	Add <ESC>!. Command.	Camille
2013/6/25	Modify sample code for PUTPCX command	Camille
2013/12/13	Modify GAP and BLINE command	Camille
2014/1/22	Add INITIALPRINTER command	Camille
2014/3/28	Modify sample code for SET COUNTER command	Samuel
2014/4/15	Add GPIO setting commands	Camille
2014/6/11	Modify SPEED section	Camille
2014/6/12	Modify BACKFEED & BACKUP section	Camille
2014/6/12	Modify DIRECTION section	Camille
2014/6/12	Modify SHIFT section	Camille
2014/6/12	Modify HOME section	Camille
2014/6/12	Modify BARCODE section	Camille
2014/6/12	Modify PUTBMP section	Camille
2014/6/12	Modify PUTPCX section	Camille
2014/6/12	Modify QRCODE section	Camille
2014/6/12	Modify TEXT section	Camille
2014/6/12	Modify ~!T section	Camille
2014/6/12	Modify DOWNLOAD section	Camille
2014/6/12	Modify KILL section	Camille
2014/6/12	Modify RUN section	Camille
2014/6/13	Add <ESC> Y command	Camille
2014/6/13	Add <ESC> Z command	Camille
2014/6/13	Modify IF...THEN...ELSE...ENDIF LOOP section	Camille
2014/6/13	Modify GETKEY() section	Camille
2014/6/13	Modify SET PARTIAL_CUTTER section	Camille
2014/6/17	Modify SET BACK section	Camille
2014/6/18	Modify SET KEY1, SET KEY2, SET KEY3 section	Camille
2014/6/18	Modify TEAR & SETSTRIPER section	Camille
2014/6/18	Modify SET HEAD section	Camille
2014/6/18	Modify SET PRINTKEY section	Camille
2014/6/18	Modify SET REPRINT section	Camille
2014/6/18	Modify KEY1, KEY2, KEY3 section	Camille
2014/6/18	Modify @YEAR, @MONTH, @DATE, @DAY, @HOUR, @MINUTE and @SECOND sections	Camille
2014/6/19	Modify SET LED1, SET LED2, SET LED3 section	Camille
2014/6/19	Modify LED1, LED2, LED3 section	Camille
2014/6/19	Modify SET GAP section	Camille
2014/6/20	Modify printer model list	Camille
2014/8/1	Modify GPO example	Camille

2014/10/14	Modify GPO function (Add PRINT)	Camille
2014/11/28	Modify printer model list	Camille
2014/11/28	Add SET REWIND section	Camille
2015/3/11	Modify printer model list	Camille
2015/4/10	Modify SPEED section	Camille
2015/5/11	Modify <ESC>!S command section (Add Print head error)	Camille
2015/5/15	Revise OFFSET command section	Camille
2015/9/11	Revise GETSETTING\$( ) section	Camille
2015/10/29	Modify SHIFT section Modify SET KEY section Modify PUTBMO section Add SET RESPONSE section	Camille
2015/10/30	Modify GPIO section	Camille
2015/11/18	Add DIAGONAL command Modify SET USBHOST section	Camille
2015/11/19	Modify DISPLAY section	Camille
2015/11/24	Add FSEARCH() command Add SET VERIFIER command	Camille
2015/11/25	Add TOUCHPRESS() command	Camille
2015/12/8	Modify DMATRIX section (add a# parameter)	Camille
2015/12/17	Add SET RS232_REWINDER command	Camille
2016/2/4	Add RECORDSET\$( ) command	Camille
2016/4/11	Add FNC sample code on DMATRIX section	Camille
2016/7/11	Modify SET KEYn section	Camille
2016/7/11	Update printer model list	Camille
2016/9/26	Update GETSETTING\$( ) section	Camille
2017/1/18	Add SET DAYLIGHT_SAVE command	Camille
2017/1/18	Add rectangular shape sample code on DMATRIX section	Camille
2017/1/18	Add LABELRATIO command	Camille
2017/2/15	Add NFC setting Command section	Camille
2017/3/8	Modify BLOCK section	Camille
2017/4/5	Modify SET KEYn section	Camille
2017/4/5	Modify KEY1, SET KEY2, SET KEY3 section	Camille
2017/4/5	Update printer model list	Camille
2017/4/5	Modify SPEED section	Camille
2017/4/6	Modify SET LEDn section	Camille
2017/4/6	Modify LED1, LED2, LED3 section	Camille
2017/4/11	Modify KEY1, SET KEY2, SET KEY3 section	Camille
2017/4/14	Add smart phone data string on QRCODE section	Camille
2017/4/17	Modify FORMAT\$( ) section (sample code)	Camille
2017/5/16	Add setting command section for Alpha-2R	Camille
2017/6/7	Add new parameters for QRCODE command	Camille
2017/6/8	Add MENU command	Camille
2017/6/8	Add sample code for [fit] parameter on BLOCK section	Camille
2017/6/8	Add sample code for (") on RECORDSET\$( ) section	Camille
2017/7/21	Add EAN128M to BARCODE section	Camille
2017/8/17	Add new parameters for SET REWIND command	Camille
2017/9/15	Add new parameters & examples for FORMAT\$( ) command	Camille
2017/10/16	Add the standard symbol sizes for DataMatrix 2D barcode on DMATRIX section	Camille
2017/10/23	- Modify the <ESC>!S section (#2: warning) - Add a parameter for SET GPI command	Camille
2017/11/22	Remove WLAN MODE (Ad-hoc)	Camille
2018/1/19	Update GPIO info.	Camille
2018/2/6	Add a sample for RSS command	Camille
2018/2/7	Add the new parameter for GETSETTING command	Camille
2018/2/12	Add the Qrcode sample code for smart phone data string	Camille
2018/2/13	Modify a parameter for SET GPI command	Camille
2018/3/1	Update FORMAT\$( ) section	Camille
2018/5/17	Update the sample Code 1 for RECORDSET\$( ) section	Camille
2018/7/13	Add parameters(25S/25I) for BARCODE command	Camille
2018/8/6	Modify the SIZE section ("n" can be an optional item)	Camille
2018/9/14	Add DNS parameter for GETSETTING\$( ) command	Camille
2018/9/25	Modify <ESC>!D section	Camille
2018/10/9	Add battery parameters and sample code on GETSENSOR() command section	Camille
2018/10/30	Add the note for example on <ESC>!S section	Camille
2018/11/20	Modify SET RS232_REWINDER section	Camille
2018/12/20	Modify sample for SET GPI section	Camille

2018/12/20	Add the parameter "BT" for SELFTEST section	Camille
2018/12/20	Add the Bluetooth module setting commands	Camille
2019/1/9	Add applicator I/O interface (DB15F) circuit diagram information	Camille
2019/3/12	Add REPLACE\$( ) command	Camille
2019/3/26	Add ML/ MB series models on SET KEYn section	Camille
2019/3/26	Modify printer model list	Camille
2019/3/29	Modify GPIO info for PEX	Camille
2019/4/17	Add SET SLEEPTIME command	Camille
2019/6/14	Move GETSETTING\$( ) of Alpha-2R to GETSETTING\$( ) section	Camille
2019/6/14	Add MB GPIO information	Camille
2019/6/17	Add parameters for DISPLAY command	Camille
2019/6/17	Add parameter for GETSETTING\$ command	Camille
2019/7/11	Modify REPLACE\$( ) section	Camille
2019/7/29	Modify the EAN128M info on BARCODE section	Camille
2019/10/3	Add SET SENSOR_REF command	Camille
2019/10/4	Modify the PUTBMP section	Camille
2020/1/16	Add TDM series for SET PRINTQUALITY, SET STANDBYTIME and SET SLEEPTIME sections	Camille
2020/2/11	Add sample code for alignment on TEXT section	Camille
2020/2/15	Add a note on PUTBMP section	Camille
2020/2/20	- Add a sample code on DOWNLOAD section - Modify the sample code on INPUT section - Add the info for RECORD MILAGE on GETSETTING\$( ) section	Camille
2020/3/10	Modify SET KEYn section	Camille
2020/3/18	Modify the sample code for SET AUTORUN section	Camille
2020/3/27	- Add a note on <ESC>!S section - Remove the SET VERIFIER section	Camille
2020/4/9	Modify SET COUNTER section	Camille
2020/6/1	- Modify SET LEDn section - Modify LED1, LED2, LED3 section - Modify SET KEYn section - Modify KEY1, KEY2, KEY3 section - Modify Printer Model List section - Modify _VERSION\$ section - Modify SET GAP section	Camille
2020/6/18	Modify sample code on SET USBHOST section	Camille
2020/9/3	Add SET REGISTRATION command	Camille
2021/1/7	-Modify Power-on utility's contents (blink pattern will be same even in different patch of firmware.)	Linda
2021/3/2	-Add DB25 Circuit diagram	Linda
2021/3/19	-Add DB25 Signal Name/Type in SATO simulation	Linda
2021/3/30	-Delete 2 button desktop printer's power-on contents. (page 109)	Linda
2021/5/7	-Add SET BLINE PRINTSIDE and SET BLINE BACKSIDE command	Linda
2021/6/4	-Modify the chart contents on <ESC>!S chapter.	Linda
2021/7/2	-Add example on LIMITFEED section	Linda
2021/7/2	-Add remark on BLOCK section	Linda
2021/9/27	-Add remark on QRCODE section	Linda
2022/3/14	-Remove WLAN OFF command	Linda
2022/4/1	-Modify RECORDSET\$ command contents	Linda
2022/7/4	-Modify SET BLINE default value for Alpha-30L/40L from printside to backside	Linda
2023/1/4	- Add the reboot parameter for SET GPI command	Camille
2023/1/9	- Add the [length] optional parameter and sample code for QRCODE command - Add the parameters for GETSETTING\$( ) command (since A2.13)	Camille
2023/5/18	Add conditions for SET GPO	Camille
2023/6/8	- Remove GPIO Interface (HD15F) section - Modify GPIO Interface (DB15F) & GPIO Interface (DB25F) sections - Add NET DNS command - Add BT PAIRMODE command - Add PUTPNG command - Add LABEL CAPACITY/RIBBON CAPACITY parameters on GETSETTING\$( ) command	Camille
2023/8/10	- Modify sample code for ABS( ) & ASC( ) - Modify the "" to "" for all sample code	Camille
2023/8/28	Modify the PUTPN sample code	Camille
2023/9/6	Add BT MODE command	Camille
2023/11/16	Add SET APPLICATOR command Update document template	Camille
2023/12/1	- Add RFID setting commands - Update RFID READ/WRITE command	Camille
2024/1/09	- Add 600 DPI : 1mm = 24 dots info.	Camille

	- Update COPY command	
2024/3/18	Update the <ESC>IS section (Status Byte #2: warning table/Unhealthy Dots)	Camille
2024/5/8	Update the RFLOCK command	Camille
2024/6/26	Update the RFID POSITION command	Camille
2024/7/1	Update the RFID POWER section	Camille
2024/9/4	<ul style="list-style-type: none"> <li>- Update the Printer Model List</li> <li>- Update the SET KEYn section</li> <li>- Update the SET LEDn section</li> <li>- Add the GPm command</li> <li>- Add the rMQR command</li> <li>- Add the sample code for GS1 Code128 on BARCODE section</li> <li>- Add the sample code for GS1 DataMatrix on DMATRIX section</li> <li>- Add the sample code for RSS GS1 on RSS section</li> <li>- Add the function conditions on SET GPO section (FORWARD &amp; BACKWARD)</li> </ul>	Camille
2024/12/26	<ul style="list-style-type: none"> <li>- Add the SET MENULOCK command</li> <li>- Update the SET BACK section (add SUPPRESS parameter)</li> <li>- Update the GETSETTING\$() section (add IMAGE LENGTH/ IMAGE WIDTH parameters)</li> </ul>	Camille
2024/12/27	Update the RFID RFLOCK command	Camille
2024/12/30	<ul style="list-style-type: none"> <li>- Add the RFIDDETECT command</li> <li>- Add the VERTICAL command</li> <li>- Add the NET SNMP command</li> <li>- Add the SET RIBBON_SAVER command</li> <li>- Add the SET SBPLIMCMD command</li> <li>- Add the SET DPLIMCMD command</li> <li>- Add the SET ZPLIMCMD command</li> <li>- Remove the SET RS232_REWINDER command</li> </ul>	Camille
2025/5/28	<ul style="list-style-type: none"> <li>- Update the "Trademark and Copyright Notice" info.</li> <li>- Add the EXPORT command</li> </ul>	Camille
2025/6/10	Update the RMQR command section	Camille
2025/7/24	Update the RFID RFLOCK command section	Camille
2025/12/22	<ul style="list-style-type: none"> <li>- Add the SET COVER command</li> <li>- Modify the print ratio limit % on BAR, BOX, RSS and REVERSE section</li> </ul>	Camille
2025/12/23	Add the HF RFID commands	Camille
2026/1/5	Add the MULTIKEY command	Camille
2026/2/5	<ul style="list-style-type: none"> <li>- Modify KEY1, KEY2, KEY3 section</li> <li>- Modify Printer Model List section</li> <li>- Modify SET LEDn section</li> <li>- Modify LED1, LED2, LED3 section</li> </ul>	Camille
2026/3/23	Correct typos and descriptions across multiple sections.	Camille

