

Software Developer's Manual

FBPL Command Reference

TD-4420TN/4520TN

TD-4650TNWB/4750TNWB/4650TNWBR/4750TNWBR

RJ-2035B/2055WB/3035B/3055WB

TJ-4020TN/4021TN/4021TNR/4120TN/4121TN/4121TNR

TJ-4420TN/4520TN/4620TN/4422TN/4522TN

Version 07

TABLE OF CONTENTS

1	Example of Command Reference	1
2	Document Conventions	2
3	Object Position Calculation	3
4	Supported Printers.....	4
5	Setup and System Commands.....	5
5.1	SIZE	5
5.2	GAP	6
5.3	GAPDETECT	8
5.4	BLINEDETECT	9
5.5	AUTODETECT	10
5.6	BLINE	11
5.7	OFFSET	12
5.8	SPEED	13
5.9	DENSITY	14
5.10	DIRECTION AND MIRROR IMAGE	15
5.11	REFERENCE	16
5.12	SHIFT	17
5.13	COUNTRY	19
5.14	CODEPAGE	20
5.15	CLS	23
5.16	FEED	24
5.17	BACKFEED	25
5.18	FORMFEED	26
5.19	HOME	27
5.20	PRINT	28
5.21	SOUND	29
5.22	CUT	30
5.23	LIMITFEED	31
5.24	SELFTEST	32
5.25	EOJ	35
5.26	DELAY	36
5.27	DISPLAY	37
5.28	INITIALPRINTER	39
5.29	MENU	40
6	Label Formatting Commands.....	43
6.1	BAR	43
6.2	BARCODE	44
6.3	TLC39	50
6.4	BITMAP	51
6.5	BOX	53
6.6	CIRCLE	54
6.7	ELLIPSE	55
6.8	CODABLOCK F MODE	56
6.9	DMATRIX	57
6.10	ERASE	59
6.11	MAXICODE	60
6.12	PDF417	62
6.13	AZTEC	65
6.14	MPDF417	66
6.15	PUTBMP	67
6.16	PUTPCX	69
6.17	QRCODE	71
6.18	RSS	77
6.19	REVERSE	82
6.20	DIAGONAL	83
6.21	TEXT	84

6.22	BLOCK.....	88
7	Status Polling and Immediate Commands.....	91
7.1	<ESC>!?	91
7.2	<ESC>!C	92
7.3	<ESC>!D	93
7.4	<ESC>!O	94
7.5	<ESC>!P	95
7.6	<ESC>!Q	96
7.7	<ESC>!R	97
7.8	<ESC>!S	98
7.9	<ESC>!F	100
7.10	<ESC>!	101
7.11	~!@	102
7.12	~!A	103
7.13	~!C	104
7.14	~!D	105
7.15	~!E	106
7.16	~!F	107
7.17	~!I	108
7.18	~!T	109
7.19	<ESC>Y	110
7.20	<ESC>Z	111
8	Commands for Windows Driver	112
8.1	!B	112
8.2	!J	113
8.3	!N	114
9	File Management Commands.....	115
9.1	DOWNLOAD	115
9.2	EOP	118
9.3	FILES	119
9.4	KILL	120
9.5	MOVE	121
9.6	RUN	122
10	BASIC Commands and Functions.....	123
10.1	ABS().....	123
10.2	ASC()	124
10.3	CHR\$().....	125
10.4	XOR\$().....	126
10.5	END	127
10.6	EOF()	128
10.7	OPEN	129
10.8	CLOSE	131
10.9	WRITE	132
10.10	READ	133
10.11	SEEK.....	134
10.12	LOF().....	135
10.13	LOC()	136
10.14	FREAD\$()	137
10.15	PUT	138
10.16	GET	139
10.17	COPY	140
10.18	FOR...NEXT LOOP	141
10.19	WHILE...WEND	142
10.20	DO...LOOP	143
10.21	IF...THEN...ELSE...ENDIF LOOP	146
10.22	GOSUB...RETURN	149
10.23	GOTO	150
10.24	INP\$().....	151
10.25	INP()	152

10.26	LOB()	153
10.27	INPUT	154
10.28	PREINPUT	155
10.29	POSTINPUT	156
10.30	SET FILTER ON/OFF	157
10.31	REM	158
10.32	OUT	159
10.33	OUTR	160
10.34	GETKEY()	161
10.35	INT()	162
10.36	LEFT\$()	163
10.37	LEN()	164
10.38	MID\$()	165
10.39	RIGHT\$()	166
10.40	STR\$()	167
10.41	STRCOMP()	168
10.42	INSTR()	169
10.43	TRIM\$()	170
10.44	LTRIM\$()	171
10.45	RTRIM\$()	172
10.46	TEXTPIXEL()	173
10.47	BARCODEPIXEL()	174
10.48	VAL()	175
10.49	NOW\$()	176
10.50	NOW	177
10.51	FORMAT\$()	178
10.52	DATEADD()	182
10.53	FSEARCH()	184
10.54	TOUCHPRESS()	185
10.55	RECORDSET\$()	186
10.56	REPLACE\$()	188
11	Device Reconfiguration Commands	189
11.1	SET COUNTER	189
11.2	SET CUTTER	190
11.3	SET PARTIAL_CUTTER	191
11.4	SET BACK	192
11.5	SET KEY _N	193
11.6	SET LED _N	195
11.7	SET PEEL	196
11.8	SET REWIND	197
11.9	SET TEAR	198
11.10	SET GAP	199
11.11	SET BLINE	201
11.12	SET HEAD	202
11.13	SET RIBBON	203
11.14	SET ENCODER	204
11.15	SET RIBBONEND	205
11.16	SET COM1	206
11.17	SET PRINTKEY	207
11.18	SET REPRINT	209
11.19	SET FEED_LEN	210
11.20	GETSENSOR()	211
11.21	GETSETTING\$()	214
11.22	SET USBHOST	217
11.23	SET AUTORUN	218
11.24	SET RESPONSE	219
11.25	SET DAYLIGHT_SAVE	221
11.26	SET REGISTRATION	222
11.27	PEEL	223

11.28	LED1, LED2, LED3	224
11.29	KEY1, KEY2, KEY3.....	225
11.30	SET SENSOR_REF.....	226
12	Printer Global Variables	227
12.1	@LABEL.....	227
12.2	YEAR	228
12.3	MONTH.....	229
12.4	DATE.....	230
12.5	WEEK	231
12.6	HOURL	232
12.7	MINUTE	233
12.8	SECOND	234
12.9	@YEAR	235
12.10	@MONTH.....	236
12.11	@DATE	237
12.12	@DAY	238
12.13	@HOUR	239
12.14	@MINUTE	240
12.15	@SECOND	241
12.16	_MODEL\$.....	242
12.17	_SERIAL\$.....	243
12.18	_VERSION\$.....	244
13	Bluetooth Module Setting Commands.....	245
13.1	BT NAME.....	245
13.2	BT PINCODE	246
13.3	BT MODE	247
13.4	SET BTLINKBACK.....	249
14	Wi-Fi Module Setting Commands	250
14.1	WLAN OFF	250
14.2	WLAN SSID.....	251
14.3	WLAN WPA	252
14.4	WLAN WEP	253
14.5	WLAN DHCP.....	254
14.6	WLAN IP	255
14.7	WLAN PORT	256
15	Internal Ethernet Setting Commands.....	257
15.1	NET DHCP	257
15.2	NET IP.....	258
15.3	NET PORT	259
15.4	NET NAME	260
16	Setting Commands for RJ-2035B/2055WB/3035B/3055WB	261
16.1	SET PRINTQUALITY.....	261
16.2	SET STANDBYTIME.....	262
16.3	SET SLEEPTIME	263
17	RFID.....	264
17.1	RFID READ / RFID WRITE	264
17.2	RFIDDETECT	269
Appendix A: Command List by Model.....		270

1 Example of Command Reference

1.1 Print

Description

Print the label format currently stored in the image buffer.

Syntax

PRINT m[,n]

Parameter	Description
m	Number of label set 1 ≤ m ≤ 999999999
n	Number of label copies 1 ≤ n ≤ 999999999

Example

Sample code

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

Result

0003

0003

0002

0002

0001

0001

1 set, 2 copies

Paper feed direction

See Also

SET COUNTER, INPUT, DOWNLOAD

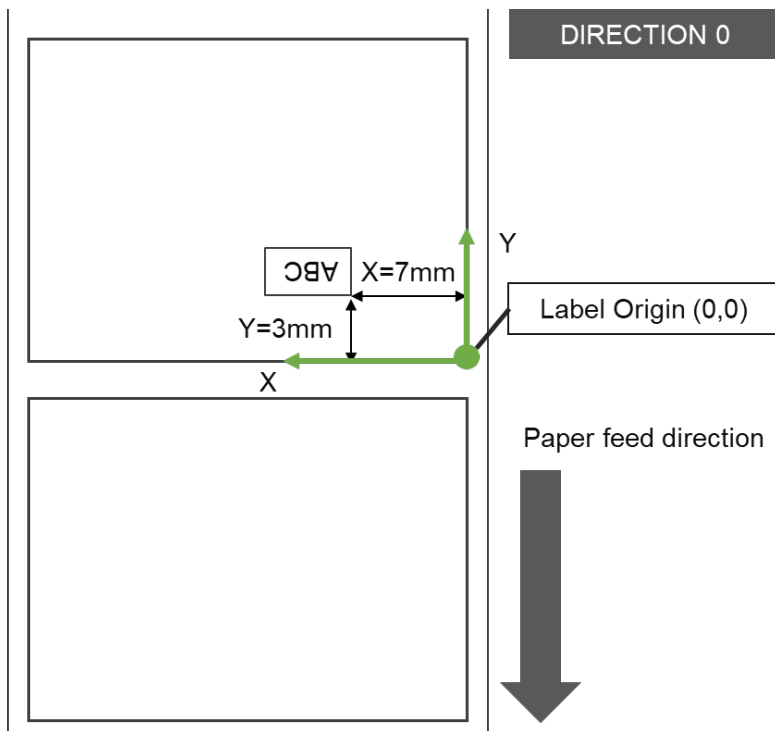
Related commands

2 Document Conventions

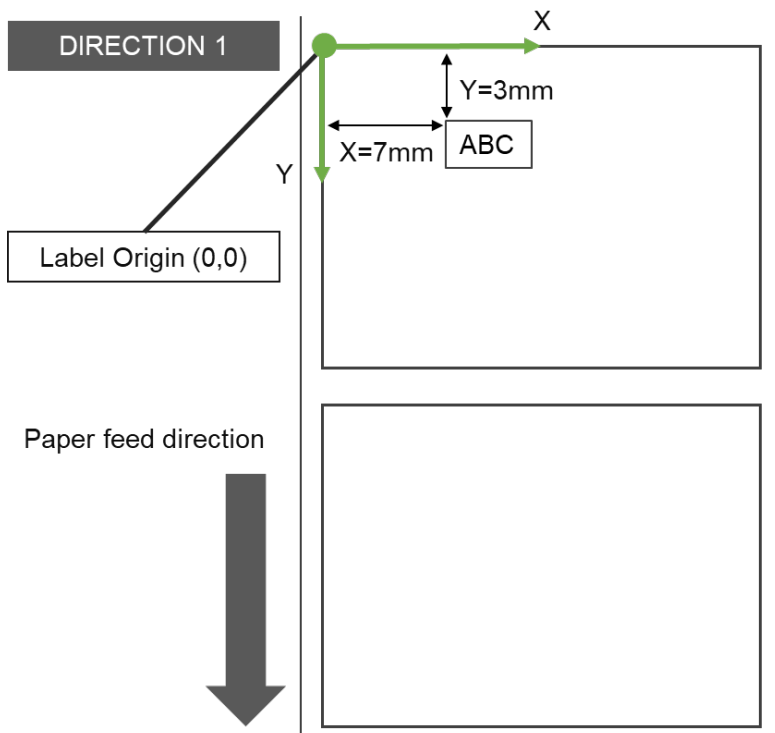
This manual uses the following typographic conventions.

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.
Note: 203 dpi: 1 mm = 8 dots	Note

3 Object Position Calculation



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



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

Note :

1. 203 dpi, 1 mm=8 dots ; 300 dpi, 1 mm=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.

4 Supported Printers

Series	Models
TD-4T series	TD-4420TN/4520TN
	TD-4650TNWB/4750TNWB/4650TNWBR/4750TNWBR
RJ series	RJ-2035B/2055WB/3035B/3055WB
TJ- series	TJ-4020TN/4021TN/4021TNR/4120TN/4121TN/4121TNR

For more information, see [Appendix A: Command List by Model](#).

5 Setup and System Commands

5.1 SIZE

Description

Define the label width and length.

Syntax

<code>SIZE m[,n]</code>	in inches
<code>SIZE m mm[,n mm]</code>	in mm
<code>SIZE m dot[,n dot]</code>	in dots

Parameter

Description

M Label width (inch/ mm/ dot)
[N] Label length (inch/ mm/ dot), Optional

Note :

- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 24 dots
- For metric and dot systems, there must be a space between parameter and “mm” or “dot”.

Example

Sample Code

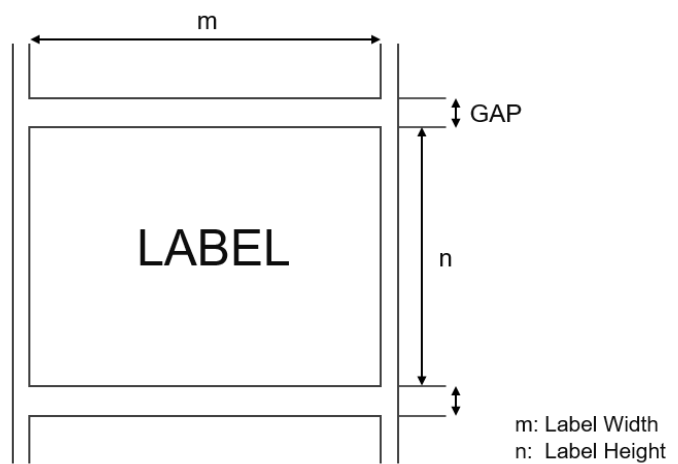
In inches:

```
SIZE 3.5,3.00
```

In mm:

```
SIZE 100 mm,100 mm
```

Result



See Also

GAP, BLINE

5.2 GAP

Description

Define the gap distance between two labels.

Syntax

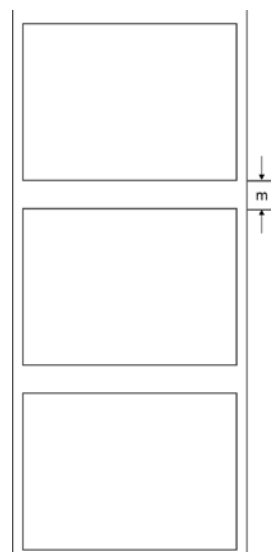
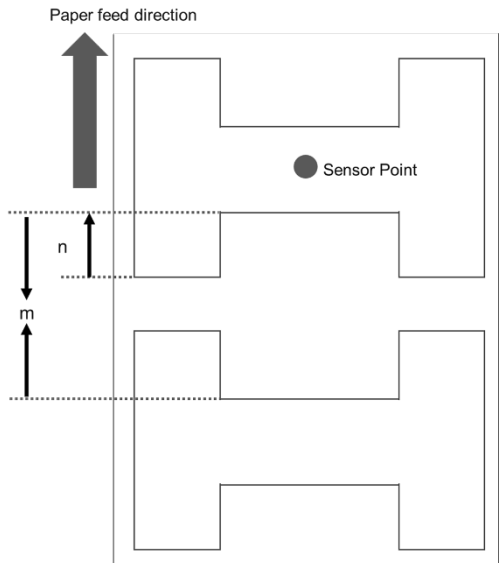
GAP m,n	in inches
GAP m mm,n mm	in mm
GAP m dot,n dot	in dots

Parameter	Description
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)
N	The offset distance of the gap $n \leq \text{label length}$ (inch or mm)
0, 0	Continuous label

Note :

- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 24 dots
- For metric and dot systems, make sure to add a space between parameter and mm.
- When the sensor type is changed from "Black Mark" to "GAP", make sure to send the **GAP** command to the printer first

Example

Sample Code	Result
<p>Normal gap</p> <p>In inches:</p> <p>GAP 0.12,0</p> <p>In mm:</p> <p>GAP 3 mm,0 mm</p> <p>Continuous label:</p> <p>GAP 0,0</p>	<p>Normal gap</p> 
<p>Special gap</p> <p>In inches</p> <p>GAP 0.30,0.10</p> <p>In mm</p> <p>GAP 7.62 mm,2.54 mm</p>	<p>Special gap</p> 

See Also
SIZE, BLINE

5.3 GAPDETECT

Description

Feed the paper through the gap sensor 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, BLINEDTECT, AUTODETECT

5.4 BLINEDTECT

Description

Feed the paper through the black mark sensor 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 **BLINEDTECT** command will not work properly. This calibration method can be applied to the labels with pre-printed logos or texts.

Syntax

BLINEDTECT [*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

5.5 AUTODETECT

Description

Feed the paper through the gap/black mark sensor 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>
<i>x</i>	Paper length (in dots)
<i>y</i>	Gap length (in dots)
Note: <ul style="list-style-type: none">▪ If the <i>x</i>, <i>y</i> parameters are ignored then the printer will calibrate and determine the paper length and gap/black mark size automatically.▪ The printer will detect the label by the proper sensor type. When using this command, make sure GAP or BLINE command are not used in your program.	

See Also

GAP, SIZE, GAPDETECT, BLINEDETECT

5.6 BLINE

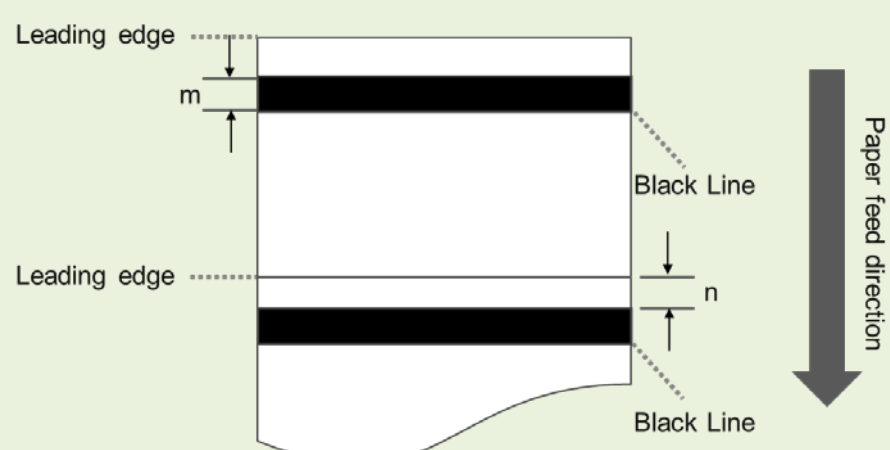
Description

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

Syntax

<code>BLINE m,n</code>	in inches
<code>BLINE m mm,n mm</code>	in mm
<code>BLINE m dot,n dot</code>	in dots

Parameter	Description
m	The height of black line either in inch or mm $0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch), $0 \leq m \leq 127$ (mm)
n	The extra label feeding length $0 \leq n \leq \text{label length}$
0,0	Continuous label



Note:

- For metric system, make sure to add a space between parameter and mm.
- When the sensor type is changed from "GAP" to "Black Mark", make sure to send the **GAP** command to the printer first.
- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 24 dots

Example

Sample Code

In inches:

```
BLINE 0.20,0.50
```

In mm:

```
BLINE 5.08 mm,12.7 mm
```

See Also

SIZE, GAP

5.7 OFFSET

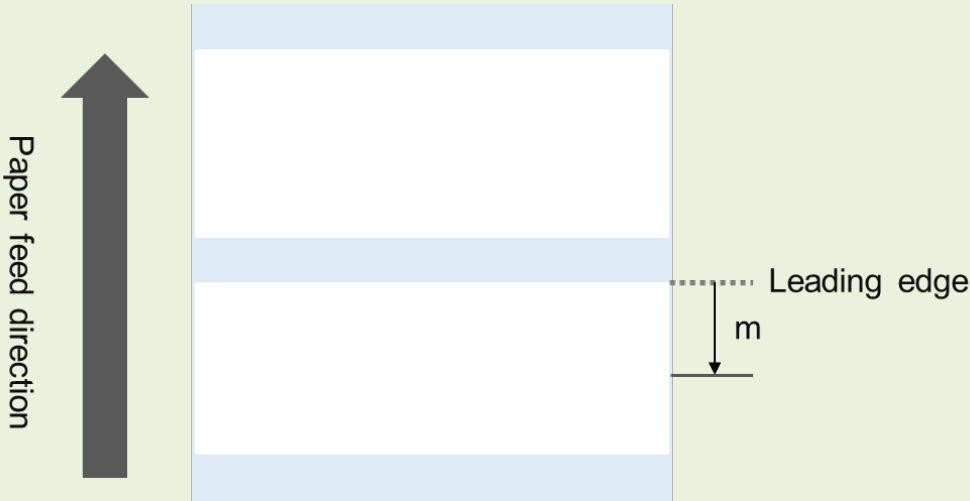
Description

Define 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

<code>OFFSET m</code>	In inches
<code>OFFSET m mm</code>	In mm
<code>OFFSET m dot</code>	In dots

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



Note:

- If offset value is incorrect, it may create a paper jam.
- For metric system, make sure to add a space between parameter and mm.
- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 24 dots

Example

Sample Code

In inches:

```
OFFSET 0.5
```

In mm:

```
OFFSET 12.7 mm
```

See Also

SIZE, GAP, SET PEEL, SET CUTTER

5.8 SPEED

Description

Define 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
TD-4420TN		V	V		V		V	V	V												
TD-4520TN		V	V		V		V	V													
TD-4650TNWB, TD-4650TNWBR		V	V		V		V	V	V	V	V										
TD-4750TNWB, TD-4750TNWBR		V	V		V		V	V	V												
RJ-2035B, RJ-2055WB	V	V	V		V		V														
RJ-3035B, RJ-3055WB	V	V	V		V		V														
TJ-4020TN, TJ-4021TN, TJ-4021TNR	V	V	V		V		V	V	V	V	V	V	V								
TJ-4120TN, TJ-4121TN, TJ-4121TNR	V	V	V		V		V	V	V	V											
TJ-4420TN, TJ-4422TN			V		V		V	V	V	V	V	V	V	V	V	V	V				
TJ-4520TN, TJ-4522TN			V		V		V	V	V	V	V	V	V	V	V						
TJ-4620TN		V	V		V		V	V	V												

Example

Sample code

```
SPEED 10
```

See Also

DENSITY

5.9 DENSITY

Description

Set the printing darkness.

Syntax

```
DENSITY n
```

<u>Parameter</u>	<u>Description</u>
n	0~15
	0 : the lightest level
	15 : the darkest level

Note:

Default **DENSITY** setting is 8.

Example

Sample code

```
DENSITY 7
```

5.10 DIRECTION and Mirror Image

Description

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

Syntax

DIRECTION n[,m]

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

DIRECTION 0,0

TEST PRINT

DIRECTION 1,0

TEST PRINT

DIRECTION 0,1

TEST PRINT

DIRECTION 1,1

TEST PRINT

Paper feed direction

Example

Sample code
■ DIRECTION 0
■ DIRECTION 0,1

See Also

REFERENCE

5.11 REFERENCE

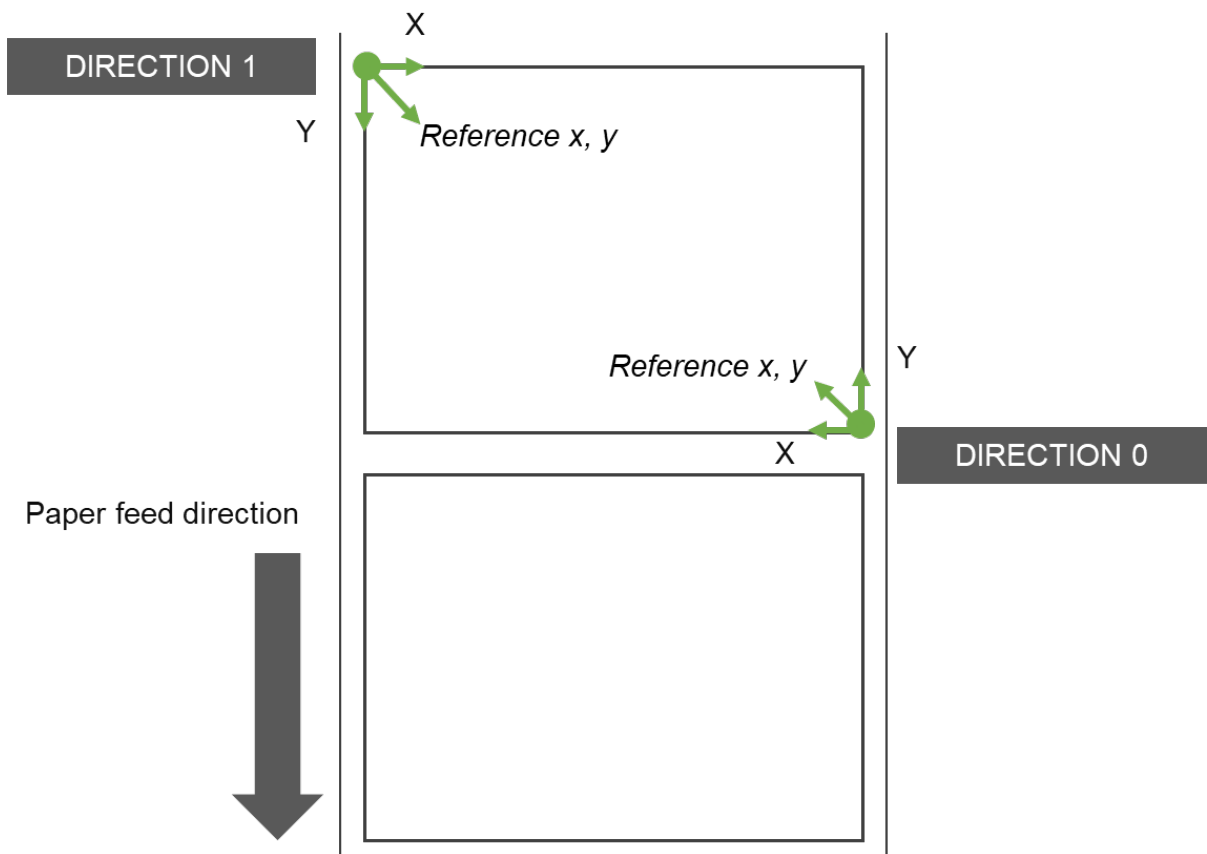
Description

Define the reference point of the label. The reference point varies depending on the print direction.

Syntax

REFERENCE x, y

Parameter	Description
x	Horizontal coordinate (in dots)
y	Vertical coordinate (in dots)
Note: <ul style="list-style-type: none">203 dpi : 1 mm = 8 dots300 dpi : 1 mm = 12 dots600 dpi : 1 mm = 24 dots	



Example

Sample code
REFERENCE 10,10

See Also

DIRECTION

5.12 SHIFT

Description

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

Syntax

`SHIFT [x,] y`

Parameter	Description
x	Optional. The maximum value is 1 inch. <ul style="list-style-type: none">• 200 dpi: -203 to 203 (in dots)• 300 dpi: -300 to 300 (in dots)
Y	The maximum value is 1 inch. <ul style="list-style-type: none">• 200 dpi: -203 to 203 (in dots)• 300 dpi: -300 to 300 (in dots)

Example

Sample Code

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



See Also
 OFFSET, REFERENCE

5.13 COUNTRY

Description

Specify a language for an external USB keyboard.

Syntax

```
PRINT m[,n]
```

<u>Parameter</u>	<u>Description</u>
n	001 USA
	003 Spanish (Latin America)
	007 Russia
	031 Dutch
	033 French (France)
	034 Spanish
	061 English (international)

Example

Sample code

```
COUNTRY 001
```

See Also

CODEPAGE, ~!!

5.14 CODEPAGE

Description

Define the code page of international character set.

Syntax

`CODEPAGE n`

Parameter

`n`

Description

Name or number of code page, which can be divided into 7-bit code page and 8-bit code page.

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

Note:

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

Example

Download the COUR.TTF into printer by Brother Printer Management Tool (BPM)

File Manager

File Download

File Type

True Type Font

▼

Browse

File Name

cour.ttf

File Size

818,480

Bytes

Memory Device

FLASH

▼

Http download file

☐

Save to file

☐

Download

File Information

Printer

HL-4070CN

▼

Memory Device

☐ DRAM

☒ FLASH

☐ CARD

Physical Space

131072

KB

Free Space

80768

KB

Remove

Get

Format

Sample Code	Result
<pre> DOWNLOAD "TEST.BAS" str1\$ = " " J = 0 y = 50 CODEPAGE 1252 SIZE 4,3 GAP 0,0 DIRECTION 1 CLS TEXT 10,10,"COUR.TTF",0,12,12,"CODEPAGE 1252" FOR I=32 TO 255 str1\$=str1\$+CHR\$(I) + " " J=J+1 IF J=16 THEN GOSUB drawTEXT NEXT PRINT 1 END drawTEXT: TEXT 10,y,"COUR.TTF",0,12,12,str1\$ str1\$=" " J=0 y=y+40 RETURN EOP TEST </pre>	<pre> CODEPAGE 1252 ! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w x y z { } ~ € , f " " ... † ‡ ^ % Š < Œ Ž \ / " " • - — ~ ™ š > œ ž Ÿ ı ċ Ł ₣ ¥ ¦ § ¨ © 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 õ ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ </pre>

See Also

COUNTRY, ~!|

5.15 CLS

Description

Clear the image buffer.

Syntax

CLS

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

Note:

Make sure to use this command after **SIZE** command.

Example

Sample code

```
CLS
```

See Also

SIZE, GAP, BLINE

5.16 FEED

Description

Feed the label for the specified distance. The length is specified in dots.

Syntax

`FEED n`

<u>Parameter</u>	<u>Description</u>
n	unit: dot $1 \leq n \leq 9999$
Note: <ul style="list-style-type: none">203 dpi : 1 mm = 8 dots300 dpi : 1 mm = 12 dots600 dpi : 1 mm = 24 dots	

Example

Sample code

```
FEED 40
```

See Also

BACKFEED, SIZE, GAP, BLINE, HOME, FORMFEED

5.17 BACKFEED

Description

Backfeeds the label for the specified distance. The length is specified in dots.

Syntax

`BACKFEED n`

<u>Parameter</u>	<u>Description</u>
n	unit: dot $1 \leq n \leq 9999$
Note: <ul style="list-style-type: none">203 dpi: 1 mm = 8 dots 300 dpi : 1 mm = 12 dots 600 dpi : 1 mm = 24 dotsIf the parameter is incorrect, it may create paper jam or wrinkle.	

Example

Sample code

```
BACKFEED 40
```

See Also

FEED, SIZE, GAP, BLINE, HOME, FORMFEED

5.18 FORMFEED

Description

Feed the label to the beginning of the next label.

Syntax

FORMFEED

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

Note:

Make sure to use this command after **SIZE** command.

Example

Sample code

```
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
```

Result

FORMFEED COMMAND TEST

Paper feed direction



See Also

FEED, SIZE, GAP, BLINE, HOME, BACKFEED

5.19 HOME

Description

Feed the label until the internal sensor has determined the origin.

Syntax

HOME

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

Note:

Make sure to define the size and gap of the label before using this command.

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

5.20 PRINT

Description

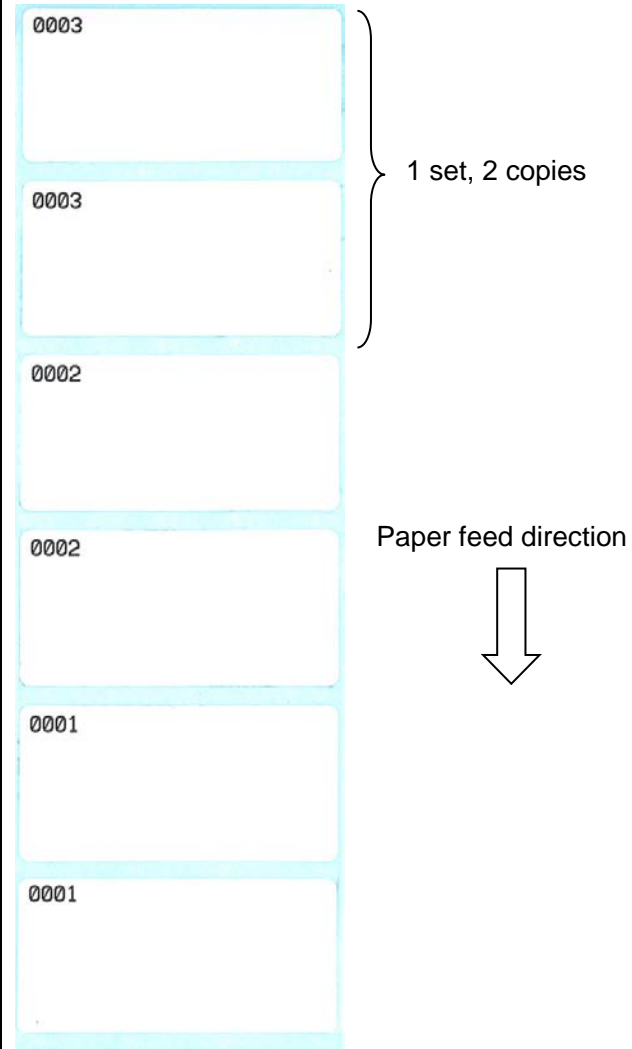
Print the label format currently stored in the image buffer.

Syntax

PRINT m[,n]

Parameter	Description
m	Number of label set $1 \leq m \leq 999999999$
n	Number of label copies $1 \leq n \leq 999999999$

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>	

See Also

SET COUNTER, INPUT, DOWNLOAD

5.21 SOUND

Description

Control 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 (in milliseconds)

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
```

5.22 CUT

Description

Activate the cutter to immediately cut the labels without backfeeding 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

5.23 LIMITFEED

Description

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

Syntax

<code>LIMITFEED n[,minpaper,maxgap]</code>	In inches
<code>LIMITFEED n mm[,minpaper mm,maxgap mm]</code>	In mm
<code>LIMITFEED n dot[,minpaper dot,maxgap dot]</code>	In dots

<u>Parameter</u>	<u>Description</u>
N	The maximum length for the sensor detection
Minpaper	The minimum length of paper
Maxgap	The maximum length of gap
Note:	
<ul style="list-style-type: none">▪ The setting will remain resident in memory.▪ For metric system, make sure to add a space between parameter n and mm.▪ The default value is 10 inches when the printer is initialized.	

Example

Sample code

```
In inches
LIMITFEED 12
```

5.24 SELFTEST

Description



Print the printer information.

Syntax

`SELFTEST [page]`

<u>Parameter</u>	<u>Description</u>
page	None: Print the whole printer information. PATTERN: Print a pattern to check the print head status. ETHERNET: Print the Ethernet configuration. WLAN: Print the Wi-Fi configuration. RS232: Print the RS-232 serial port configuration. SYSTEM: Print the printer configuration. Z: Print the emulated language settings. BT: Print the Bluetooth configuration.

Example

Sample code	Result
SELFTEST	<pre> ----- SYSTEM INFORMATION ----- MODEL: XXXXXX FIRMWARE: XXXXXX CHECKSUM: XXXXXX S/N: XXXXXX 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: XXXXXX MAC ADDR: XXXXXX DHCP: ON IP ADDR: XXXXXX SUBNET: XXXXXX GATEWAY: XXXXXX PORT: 9100 ----- </pre>

SELFTEST WLAN	<pre> ----- WLAN SETTING ----- MAC ADDR: XXXX-XXXX SSID: TEST-AP DHCP: OFF IP ADDR: 10.0.10.138 SUBNET: 255.255.255.0 GATEWAY: 10.0.10.252 PORT: 9100 ----- </pre>
SELFTEST RS232	<pre> ----- RS232 SETTING ----- BAUD: 9600 PARITY: NONE DATA BIT: 8 STOP BIT: 1 ----- </pre>
SELFTEST SYSTEM	<pre> ----- SYSTEM INFORMATION ----- MODEL: XXXXXX FIRMWARE: XXXXXX CHECKSUM: XXXXXX S/N: XXXXXX 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>
SELFTEST BT	<pre> ----- BT SETTING ----- MAC ADDR: XXXXXX111111 NAME: BROTHER01 PIN CODE: 0000 PRINTER NAME: PAIR MODE: LEGACY MODULE: XXXX XXX MFi SUPPORTED: YES ----- </pre>

5.25 EOJ

Description

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

Syntax

EOJ

Example

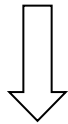
Sample Code

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

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

Result

Paper feed direction



Printer stops before next printing.

Printer stops before next printing.

Two labels are printed without stop.

Two labels are printed without stop.

} without stop

5.26 DELAY

Description

Define the period of time for the printer to wait before processing the next command.

Syntax

DELAY ms

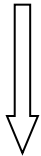
<u>Parameter</u>	<u>Description</u>
ms	Set the print delay time in milliseconds. 1000 ms = 1 second.

Example

Sample Code

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

Result



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

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

5.27 DISPLAY

Description

Display the image, which is in the printer's image buffer, on the LCD panel.

Syntax

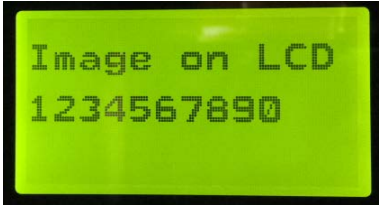

DISPLAY IMAGE/OFF/CLS/.....

<u>Parameter</u>	<u>Description</u>																										
IMAGE	Display the image in printer's image buffer on the LCD panel.																										
OFF	Disable this function.																										
CLS	Display the background color and clear the items in the printer's image buffer on the LCD panel.																										
forecolor,backcolor	Set the color (decimal) for item and background in the printer's image buffer on the LCD panel																										
x,y,width,height	Draw the bar in the printer's image buffer on the LCD panel																										
x,y,width,height, thick																											
x,y,width,height, thick, radius																											
x,y,"bmpfile"	Display the .bmp in the printer's image buffer on the LCD panel																										
x,y,"font","content"	Display the text in printer's image buffer on the LCD panel																										
x,y,"font",rotate,"content"																											
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><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><tr><td>radius</td><td>Frame radius</td></tr><tr><td>bmpfile</td><td>.bmp file name</td></tr><tr><td>font</td><td>Font name</td></tr><tr><td>rotate</td><td>Rotation (0, 90, 180, and 270 valid)</td></tr><tr><td>x-multi</td><td>Horizontal multiplication</td></tr><tr><td>y-multi</td><td>Vertical multiplication</td></tr></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	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
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.
For more information, see [Appendix A: Command List by Model](#).

Example

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

5.28 INITIALPRINTER

Description

Restore printer settings to defaults.

Syntax

`INITIALPRINTER`

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

Example

Sample code

```
INITIALPRINTER
```

5.29 MENU

Description

Design user's own menu with a database resident on the printer.

Syntax

```
MENU title$, list$, selected
```

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

Note:

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

For more information, see [Appendix A: Command List by Model](#).

Example

Sample code

```
DOWNLOAD F,"FBPL"  
Speed  
Density  
Print Mode  
Offset  
Country  
EOP
```

```
DOWNLOAD F,"Speed"  
4  
5  
6  
EOP
```

```
DOWNLOAD F,"Density"  
6  
7  
8  
9  
10  
11  
12  
EOP
```

```
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 "FBPL",0
LIST$ = FREAD$(0, LOF("FBPL"))
CLOSE 0
MENU "FBPL", LIST$, OPTION$

IF LEN(OPTION$) = 0 THEN END

IF OPTION$ = "Speed" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","SPEED")
IF OPTION$ = "Density" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","DENSITY")
IF OPTION$ = "Print Mode" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","PRINT MODE")
IF OPTION$ = "Offset" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","OFFSET")
IF OPTION$ = "Country" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","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

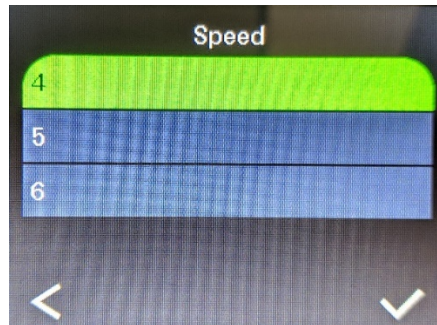
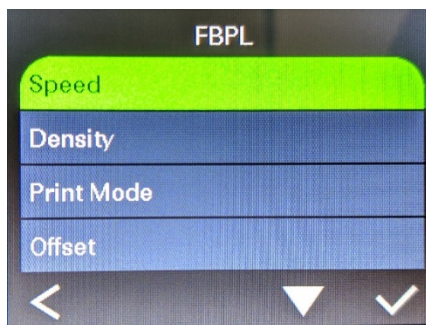
```

```
: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"

Result



6 Label Formatting Commands

6.1 BAR

Description

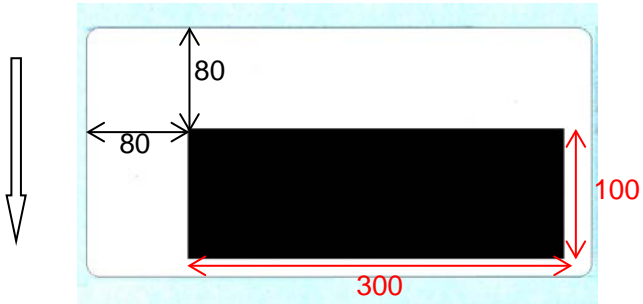
Draw a bar on the label format.

Syntax

`BAR x,y,width,height`

Parameter	Description
x	The upper left corner x-coordinate (in dots)
y	The upper left corner y-coordinate (in dots)
width	Bar width (in dots)
height	Bar height (in dots)
Note:	
<ul style="list-style-type: none">203 dpi : 1 mm = 8 dots 300 dpi : 1 mm = 12 dots 600 dpi : 1 mm = 24 dotsRecommended max. bar height is 12 mm at 4" width. Bar height over 12 mm may damage the power supply and affect the print quality.Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.	

Example

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

See Also

BOX

6.2 BARCODE

Description

Print 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	
128	Code 128, switching code subset automatically	V					
128M	Code 128, switching code subset manually.	V					
EAN128	EAN128, switching code subset automatically	V					
EAN128M	EAN128M, switching code subset manually	V					
25	Interleaved 2 of 5		V	V	V		Length is even
25C	Interleaved 2 of 5 with check digit		V	V	V		Length is odd
25S	Standard 2 of 5		V	V	V		
25I	Industrial 2 of 5		V	V	V		
39	Code 39, switching standard and full ASCII mode automatically		V	V	V		
39C	Code 39 with check digit		V	V	V		
93	Code 93			V			
EAN13	EAN 13	V					12
EAN13+2	EAN 13 with 2 digits add-on	V					14
EAN13+5	EAN 13 with 5 digits add-on	V					17
EAN8	EAN 8	V					7
EAN8+2	EAN 8 with 2 digits add-on	V					9
EAN8+5	EAN 8 with 5 digits add-on	V					12
CODA	Codabar		V	V	V		
POST	Postnet	V					5, 9, 11
UPCA	UPC-A	V					11
UPCA+2	UPC-A with 2 digits add-on	V					13
UPA+5	UPC-A with 5 digits add-on	V					16
UPCE	UPC-E	V					6
UPCE+2	UPC-E with 2 digits add-on	V					8
UPE+5	UPC-E with 5 digits add-on	V					11
MSI	MSI		V	V	V		
MSIC	MSI with check digit		V	V	V		
PLESSEY	PLESSEY		V	V	V		
CPOST	China post					V	
ITF14	ITF14		V	V	V		13
EAN14	EAN14	V					13
11	Code 11		V	V	V		
TELEPEN	Telepen		V	V	V		
TELEPENNN	Telepen number		V	V	V		
PLANET	Planet	V					
CODE49	Code 49	V					
DPI	Deutsche Post Identcode		V	V	V		11
DPL	Deutsche Post Leitcode		V	V	V		13
LOGMARS	A special use of Code 39		V	V	V		

Syntax

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

Parameter	Description																																																
X	Specify the x-coordinate barcode on the label																																																
Y	Specify the y-coordinate barcode on the label																																																
code type																																																	
128	Code 128, switching code subset A, B, C automatically																																																
128M	<table><tr><td colspan="4">Code 128, switching code subset A, B, C manually</td></tr><tr><td>Control code</td><td>A</td><td>B</td><td>C</td></tr><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></table> <p>Use "!" as a starting character for the control code followed by three control codes. If the start subset is not set, the default starting subset is B.</p>	Code 128, switching code subset A, B, C manually				Control code	A	B	C	096	FNC3	FNC3	NONE	097	FNC2	FNC2	NONE	098	SHIFT	SHIFT	NONE	099	CODE C	CODE C	NONE	100	CODE B	FNC4	CODE B	101	FNC4	CODE A	CODE A	102	FNC1	FNC1	FNC1	103	Start (CODE A)			104	Start (CODE B)			105	Start (CODE C)		
Code 128, switching code subset A, B, C manually																																																	
Control code	A	B	C																																														
096	FNC3	FNC3	NONE																																														
097	FNC2	FNC2	NONE																																														
098	SHIFT	SHIFT	NONE																																														
099	CODE C	CODE C	NONE																																														
100	CODE B	FNC4	CODE B																																														
101	FNC4	CODE A	CODE A																																														
102	FNC1	FNC1	FNC1																																														
103	Start (CODE A)																																																
104	Start (CODE B)																																																
105	Start (CODE C)																																																
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 FBPL printers Auto switch full ASCII and standard code 39 for PLUS models																																																
39C	Code 39 full ASCII with check digit for FBPL printers Auto switch full ASCII and standard code 39 for PLUS models																																																
39S	Code 39 standard for FBPL printers																																																
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				
	TELEPEN N	Telepen code. Number only				
	PLANET	Planet code				
	CODE49	Code 49				
	DPI	Deutsche Post Identcode				
	DPL	Deutsche Post Leitcode				
Height	Barcode 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	narrow : wide 1:2	narrow : wide 1:3	narrow : wide 2:5	narrow : wide 3:7
	128	10x	-	-	-	-
	EAN128	10x	-	-	-	-
	EAN128M	10x				
	25	-	10x	10x	5x	-
	25C	-	10x	10x	5x	-
	25S		10x	10x	5x	
	25I		10x	10x	5x	
	39	-	10x	10x	5x	-
	39C	-	10x	10x	5x	-
	93	-	-	10x	-	-
	EAN13	8x	-	-	-	-
	EAN13+2	8x	-	-	-	-
	EAN13+5	8x	-	-	-	-
	EAN 8	8x	-	-	-	-
	EAN 8+2	8x	-	-	-	-
	EAN 8+5	8x	-	-	-	-
	CODA	-	10x	10x	5x	-
	POST	1x	-	-	-	-
	UPCA	8x	-	-	-	-
	UPCA+2	8x	-	-	-	-
	UPCA+5	8x	-	-	-	-
	UPCE	8x	-	-	-	-
	UPCE+2	8x	-	-	-	-
	UPCE+5	8x	-	-	-	-
	CPOST	-	-	-	-	1x
	MSI	-	-	10x	-	-
	MSIC			10x		-
	PLESSY	-	-	10x	-	-
	ITF14	-	10x	10x	5x	-
	EAN14	8x	-	-	-	-
	11	-	10x	10x	5x	-
alignment	Specify the alignment of barcode 0 : default (Left) 1 : Left 2 : Center 3 : Right					

Note:



Check the maximum number of digits in the barcodes.

Code Type	Character sets	Max. data length
128	See Character set for CODE128.	-
128M	See Character set for CODE128.	-
EAN128	See Character set for CODE128.	-
EAN128M	See Character set for CODE128.	-
25	0123456789	Length is even.
25C	0123456789	Length is odd.
25S	0123456789	
25I	0123456789	
39 I	0123456789[Space]ABCDEFGHIJKLMNQRST UVWXYZ-.\$/+%	-
39 I Full ASCII	0123456789[Space]ABCDEFGHIJKLMNQRST UVWXYZ!#\$%&'()*+,-./:;=<?@[\\]^_` abcdefghijklmnopqrstuvwxyz{ }~	-
93	0123456789[Space]ABCDEFGHIJKLMNQRST UVWXYZ!#\$%&'()*+,-./:;=<?@[\\]^_` abcdefghijklmnopqrstuvwxyz{ }~	-
EAN13	0123456789	12
EAN13+2	0123456789	14
EAN13+5	0123456789	17
EAN8	0123456789	7
EAN8+2	0123456789	9
EAN8+5	0123456789	12
CODA	0123456789-.\$/+	-
POST	0123456789	5, 9, 11
UPCA	0123456789	11
UPCA+2	0123456789	13
UPA+5	0123456789	16
UPCE	0123456789	6
UPCE+2	0123456789	8
UPE+5	0123456789	11
MSI	0123456789	-
MSIC	0123456789	-
PLESSEY	0123456789	-
CPOST	0123456789	-
ITF14	0123456789	13
EAN14	0123456789	13
11	0123456789-	-
TELEPEN	ASCII 0 to 127	30
TELEPENN	0123456789	60
PLANET	0123456789	38
CODE49	ASCII 0 to 127	81
DPI	0123456789	11
DPL	0123456789	13
LOGMARS	0123456789[Space]ABCDEFGHIJKLMNQRST UVWXYZ-.\$/+%	-

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
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "2",0,1,1, "Human readable alignment" BARCODE 10,50, "128",100,1,0,2,2,"left" BARCODE 310,50, "128",100,2,0,2,2,"center" BARCODE 610,50, "128",100,3,0,2,2,"right" PRINT 1 </pre>	<p>Human readable alignment</p>  <p>left center right</p>
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "2",0,1,1, "Code 128, switch code subset automatically. " BARCODE 10,50, "128",100,1,0,2,2, "123456abcd123456" PRINT 1 </pre>	<p>Code 128, switch code subset automatically.</p>  <p>123456abcd123456</p>

<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "2",0,1,1, "Code 128, switch code subset manually." BARCODE 10,50, "128M",100,1,0,2,2, "!104!096ABCD!101EFGH" PRINT 1 </pre> <p>Note: 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 and EFGH characters encoded as CODE A subset.</p>	<p>Code 128, switch code subset manually.</p> 
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "2",0,1,1, "TELEPEN" BARCODE 10,50, "TELEPEN",100,1,0,2,6, "abcd1234ABCD" PRINT 1 </pre>	<p>TELEPEN</p> 
<pre> 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 </pre>	

6.3 TLC39

Description

Draw TLC39, TCIF Linked Barcode 3 of 9, barcode.

Syntax

`TLC39 x,y,rotation,[height],[narrow],[wide],[cellwidth],[cellheight,] "ECI number,Serial number & additional data"`

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

Example

Sample Code

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

Result

TLC39 code



6.4 BITMAP

Description

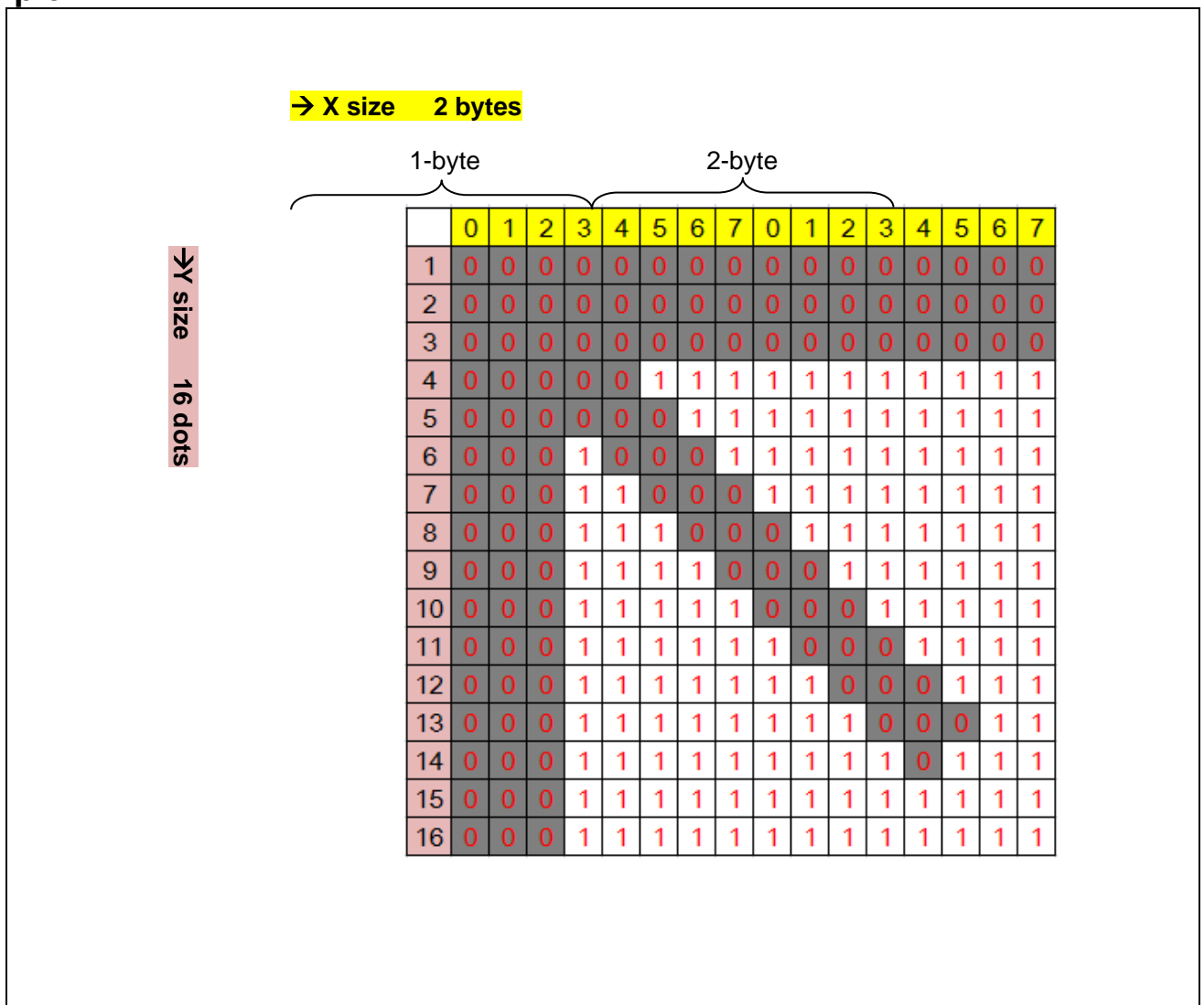
Draw a bitmap image (as opposed to BMP graphic files).

Syntax

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

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

Example



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

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

See Also

PUTBMP, PUTPCX

6.5 BOX

Description

Draw a rectangle on the label.

Syntax

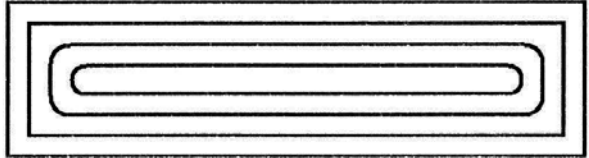
```
BOX x,y,x_end,y_end,line thickness[,radius]
```

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

Note:

- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 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 30% respectively.

Example

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

See Also

BAR

6.6 CIRCLE

Description

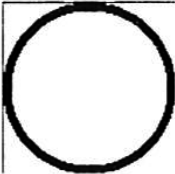
Draw a circle on the label.

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

<u>Sample code</u>	<u>Result</u>
<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>	

6.7 ELLIPSE

Description

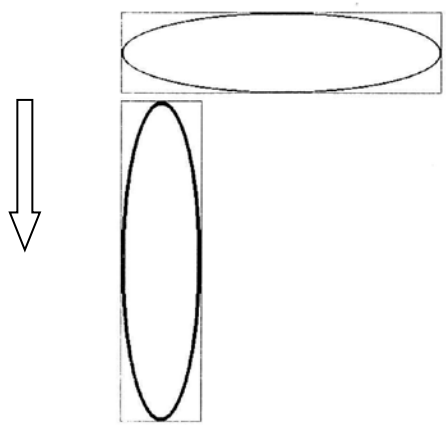
Draw an ellipse on the label.

Syntax

```
ELLIPSE x,y,width,height,thickness
```

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

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>	

6.8 CODABLOCK F mode

Description

Draw a CODABLOCK F mode barcode.

Syntax

```
CODABLOCK x,y,rotation,[row height],[module width,] "content"
```

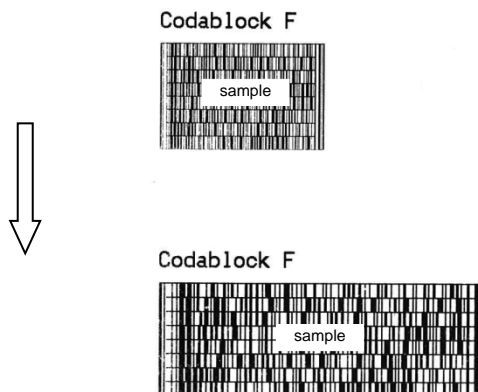
Parameter	Description
x	Specify the x-coordinate
y	Specify the y-coordinate
rotation	0 : No rotation 90 : Rotate 90 degrees clockwise 180 : Rotate 180 degrees clockwise 270 : Rotate 270 degrees clockwise
row height	The height of individual row equals to row height x module width (Default is 8)
module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

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, "With the "At your side." spirit in mind. the Brother
Group aims to continually create value."
PRINT 1
CLS
TEXT 10,10, "3",0,1,1, "Codablock F"
CODABLOCK 10,50,0,16,1, "With the "At your side." spirit in mind. the
Brother Group aims to continually create value."
PRINT 1
```

Result



6.9 DMATRIX

Description

Define a DataMatrix 2D barcode. Only ECC200 error correction is supported.

Syntax

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




Parameter	Description																																																																																																												
x	Horizontal start position (in dots)																																																																																																												
y	Vertical start position (in dots)																																																																																																												
width	The expected width of barcode area (in dots)																																																																																																												
height	The expected height of barcode area (in dots)																																																																																																												
c#	Escape sequence control character (decimal digit) Ex. C126 means ~ (1) ~x is shift character for control characters.																																																																																																												
	<table><tr><td>~X</td><td>Hex</td><td>ASCII</td><td>~X</td><td>HEX</td><td>ASCII</td><td>~X</td><td>HEX</td><td>ASCII</td><td>~X</td><td>HEX</td><td>ASCII</td></tr><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></table>	~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
~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#	0 : Square (default) 1 : Rectangle																																																																																																												
row	Symbol size of row: 10 to 144																																																																																																												
col	Symbol size of col: 10 to 144																																																																																																												
content	Content of DataMatrix 2D barcode																																																																																																												

Note:

For standard symbol sizes for DataMatrix 2D barcode, see the list below.

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

6.10 ERASE

Description

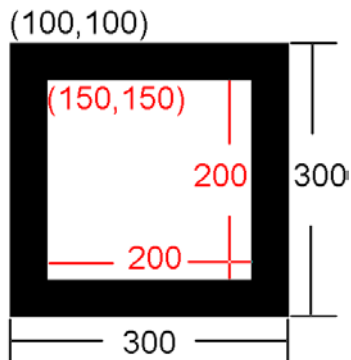
Clear a specified region in the image buffer.

Syntax

```
ERASE x,y,x_width,y_height
```

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

Example

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

See Also

CLS

6.11 MAXICODE

Description

Define a 2D Maxicode.

Syntax

<code>MAXICODE x,y,mode,[class,country,post,Lm,] "content"</code>	
<code>MAXICODE x,y,mode,class,country,postal code, "content"</code>	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.
<code>MAXICODE x,y,mode,[Lm,] "content"</code>	For mode 4 or 5, AIM special format is supported.

<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≤m≤138, (this parameter is just for mode 4 and 5)
content	Content of 2D Maxicode Note: If parameter Lm is used, double quotes (") are unnecessary.

Example

Sample code
<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</pre>

```

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

```

```

REM *****MODE 5*****

```

```

CLS

```

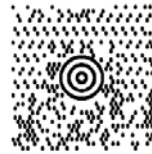
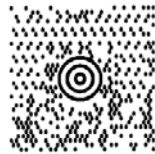
```

MAXICODE 110,100,5, "DEMO 5 FOR MAXICODE"
MAXICODE 600,100,5,L19,DEMO 5 FOR MAXICODE
TEXT 100,50, "3",0,1,1, "DEMO 5 FOR MAXICODE"
PRINT 1

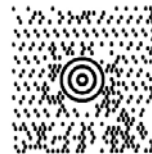
```

Result

DEMO 5 FOR MAXICODE



Mode 4 FOR MAXICODE



Mode 3 For CANADA



Mode 2 For USA



6.12 PDF417

Description

Define a PDF417 2D barcode.



Syntax

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

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

Example

Sample code	Result
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 REM *****WITHOUTR 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 PDF417 50,50,600,600,0,E4,W4,H4, "Error correction level:4 module width 4 dots bar height 4 dots" PRINT 1,1 </pre>	

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

6.13 AZTEC

Description

Define a AZTEC 2D barcode.

Syntax

```
AZTEC x,y,rotate,[size,]ecp,[flg,]menu,[multi,]rev,] "content"
```

```
AZTEC x,y,rotate,size,ecp,flg,menu,multi,rev,bytes,content
```

Parameter	Description
x	Horizontal start position (in dots)
y	Vertical start position (in dots)
rotate	Rotation 0 : No rotation 90 : Rotate 90 degrees 180 : Rotate 180 degrees 270 : Rotate 270 degrees
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 barcode
Note: If parameter bytes is used, double quotes (") are unnecessary.	

Example

Sample Code	Result
<pre>SIZE 4,2 GAP 0,0 CLS AZTEC 10,10,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" AZTEC 210,10,0,4,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" AZTEC 410,10,0,4,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" AZTEC 610,10,0,4,1,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" AZTEC 10,310,0,4,1,0,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" AZTEC 210,310,0,4,1,0,0,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" AZTEC 410,310,0,4,1,0,0,1,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" " AZTEC 610,310,0,4,1,0,0,1,1,10,1234567890 PRINT 1</pre>	

6.14 MPDF417

Description

Define a Micro PDF 417 barcode.

Syntax

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

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

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



6.15 PUTBMP

Description

Print BMP format images. The grayscale printing is supported in the direct thermal mode only. 1-bit (monochrome) and 8-bit (256-color) BMP graphic are only supported.

Syntax

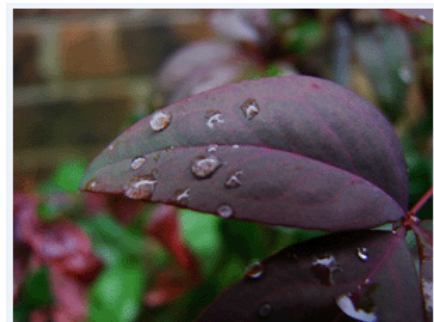
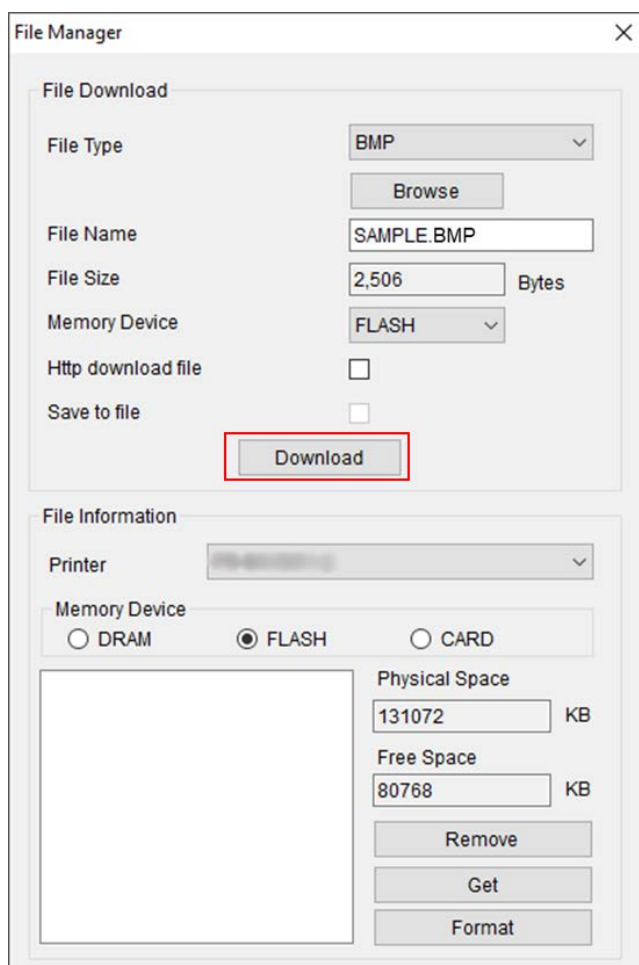
```
PUTBMP x,y, "filename" [, bpp][, contrast]
```

Parameter	Description
x	x-coordinate of the BMP format image
y	y-coordinate of the BMP format image
filename	Downloaded BMP filename
bpp	Optional. Bits per pixel of grayscale graphic. Default is 1. 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.

Example

Download BMP file to the printer by Brother Printer Management Tool (BPM).

Important: Make sure that the file name extension is in upper case letters (.PCX).



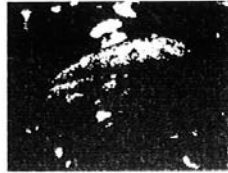
Sample Code

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

Result



bpp and contrast
are omitted.



bpp = 1
contrast = 80



bpp = 8
contrast = 80

Sample Code

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

See Also

DOWNLOAD, BITMAP, PUTPCX

6.16 PUTPCX

Description

Print PCX format images. FBPL language supports 256-color PCX format graphics.

Syntax

```
PUTPCX x,y, "filename"
```

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

Example

Download PCX file to the printer by Brother Printer Management Tool (BPM).

Important: Make sure that the file name extension is in upper case letters (.PCX).

File Manager

File Download

File Type

PCX

Browse

File Name

SAMPLE.PCX

File Size

2,506

Bytes

Memory Device

FLASH

Http download file

☐

Save to file

☐

Download

File Information

Printer

Memory Device

☐ DRAM ☒ FLASH ☐ CARD

Physical Space

131072

KB

Free Space

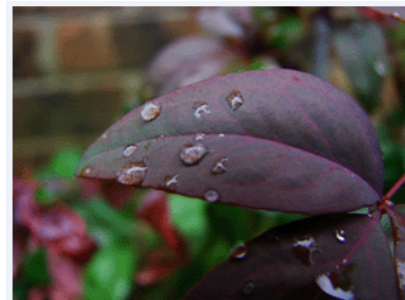
80768

KB

Remove

Get

Format



Sample Code

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

Result



See Also

DOWNLOAD, BITMAP, PUTPCX

6.17 QR CODE

Description

Print a QR code.

Syntax

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

Parameter	Description															
x	The upper left corner x-coordinate of the QR code															
y	The upper left corner y-coordinate of the QR code															
ECC level	Error correction recovery level L : 7% M : 15% Q : 25% H : 30%															
cell width	1~10															
mode	Auto / manual encode A : Auto M : Manual															
rotation	0 : 0 degree 90 : 90 degree 180 : 180 degree 270 : 270 degree															
[justification]	Barcode justification (J1 to J9 valid; See the Sample Code below)															
[model]	M1: (default), original version M2: enhanced version (Almost all smartphone support this version.)															
[mask]	S0~S8, default is S7															
[area]	Maximum size of barcode area (Xdots; e.g. X100)															
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 _{HEX} –9FFC _{HEX} and E040 _{HEX} –EAA4 _{HEX} . 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): <table><tr><th></th><th>Model 1 (Version 14-L)</th><th>Model 2 (Version 40-L)</th></tr><tr><td>Numeric data</td><td>1,167 characters</td><td>7,089 characters</td></tr><tr><td>Alphanumeric data</td><td>707 characters</td><td>4,296 characters</td></tr><tr><td>8-bit byte data</td><td>486 characters</td><td>2,953 characters</td></tr><tr><td>Kanji data</td><td>299 characters</td><td>1,817 characters</td></tr></table> *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		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
	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														

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:

QRCODE 100,10,L,7,M,0,M1,S1, "ATHE FIRMWARE HAS BEEN UPDATED"

(Where A: Alphanumeric data)

QRCODE 100,10,M,7,M,0,M1,S2, "N123456"

(Where N: Numeric data)

QRCODE 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"

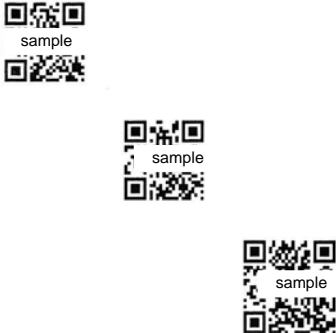
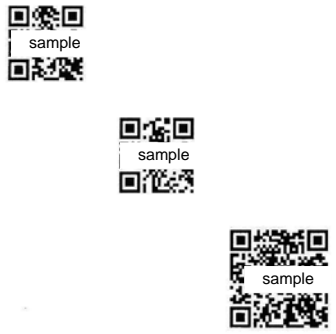

(Where K: Kanji data)

Auto mode example:

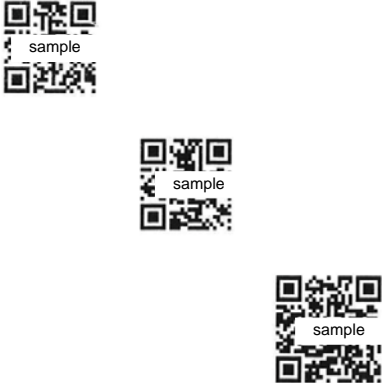


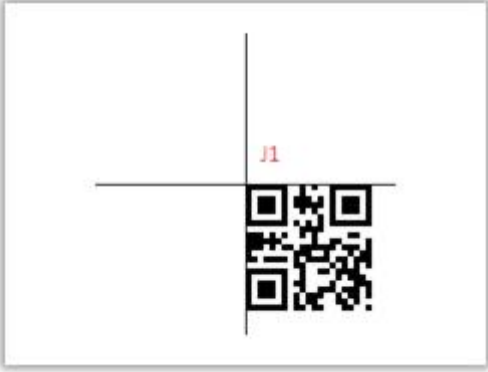
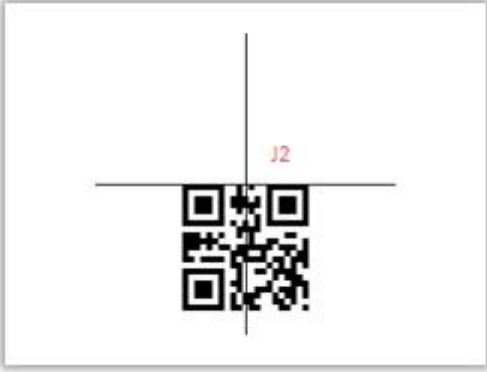
QRCODE 100,10,M,7,A,0, "THE FIRMWARE HAS BEEN UPDATED"

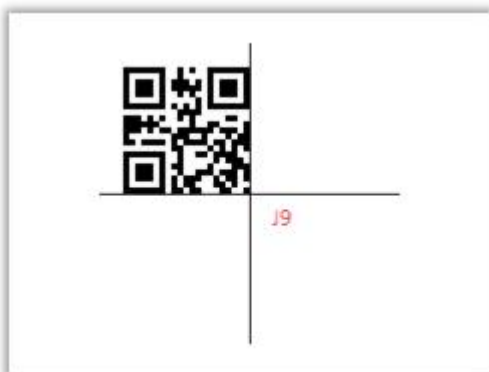
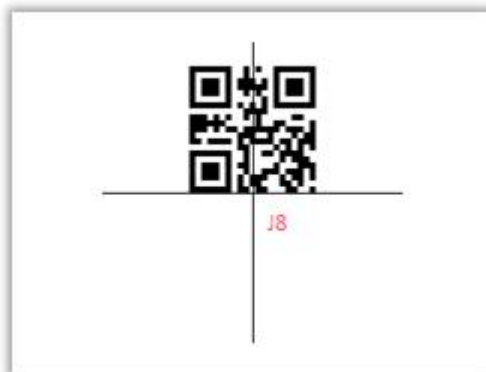
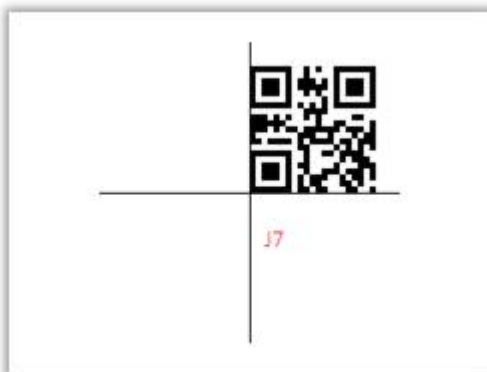
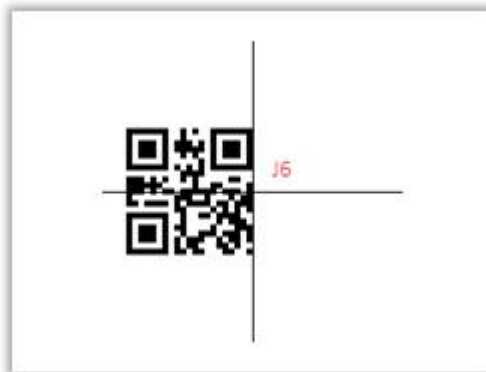
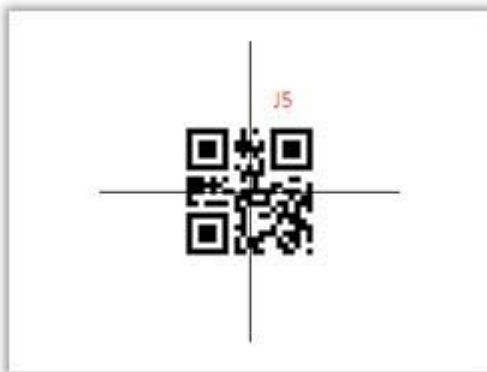
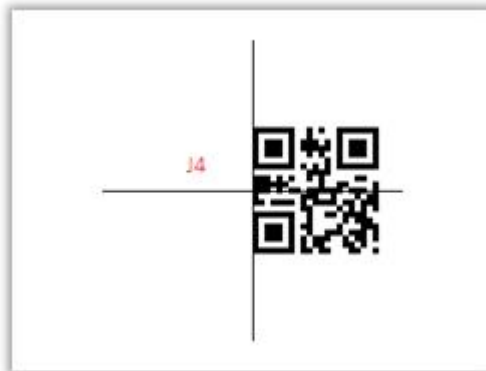
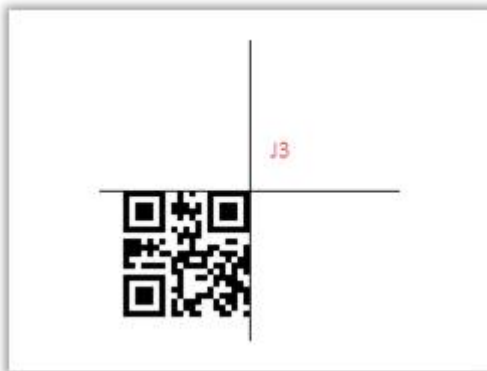
Example

Sample code	Result
Auto mode example	
<u>General data string</u>	
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, "Printer ABCabc123" PRINT 1,1	

<p><u>Data string including <Enter> character (0Dh, 0Ah)</u></p> <pre> SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,A,0, "ABC<Enter> abc<Enter> 123 " QRCODE 160,160,H,4,A,0, "123<Enter> ABC<Enter> abc" QRCODE 310,310,H,4,A,0, "Printer<Enter> ABC<Enter> abc<Enter> 123" PRINT 1,1 </pre>	
<p><u>Data string concatenation</u> (Must be used with DOWNLOAD ... EOP command)</p> <pre> DOWNLOAD "DEMO.BAS" SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,A,0, "ABCabc123" +STR\$(1234) QRCODE 160,160,H,4,A,0, "123ABCabc" +"1234" QRCODE 310,310,H,4,A,0, "PrinterABCabc123"+"1234"+"abcd" PRINT 1,1 EOP DEMO </pre>	
<p><u>Data string including double quote (") character</u> <u>Make sure to 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" QRCODE 160,160,H,4,A,0, "123\["]ABC\["]abc" QRCODE 310,310,H,4,A,0, "\["]Printer\["]ABCabc123" PRINT 1,1 </pre>	

Manual mode	
<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 Printer!AABC!B0006abc123" PRINT 1,1 </pre>	
<p><u>Data string including <Enter> character. <Enter> 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<Enter> abc<Enter> !N123" QRCODE 160,160,H,4,M,0, "N123!B0002<Enter> !AABC!B0005<Enter> abc" QRCODE 310,310,H,4,M,0, "K Printer!B0002<Enter> !AABC!B0010<Enter> abc<Enter> 123" PRINT 1,1 </pre>	
<p><u>Data string concatenation</u> (Must be used with DOWNLOAD ... EOP command)</p> <pre> DOWNLOAD "A.BAS" SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,M,0, "AABC!B0006abc123!N"+STR\$(1234) QRCODE 160,160,H,4,M,0, "N123!AABC!B0007abc"+"1234" QRCODE 310,310,H,4,M,0, "K Printer!AABC!B0014abc123"+" 1234"+"abcd" PRINT 1,1 EOP A </pre>	

<p><u>Data string including double quote (") character</u> <u>Make sure to use \" instead of (")</u></p> <pre> SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,M,0, "AABC!B0005\["]abc\["]!N123" QRCODE 160,160,H,4,M,0, "N123!B0001\["]!AABC!B0004\["]abc" QRCODE 310,310,H,4,M,0, "B0001\["]!K Printer!B0010\["]ABCabc123" PRINT 1,1 </pre>	
<p><u>Smart phone data string</u></p> <pre> 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 日本語" QRCODE 300,300, L, 8, M, 0, M2,"B0026http://www.brother.com" PRINT 1,1 EOP A </pre>	
<p><u>Data string for parameter [justification] & [area]</u></p> <pre> SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS BAR 60,120,200,1 BAR 160,20,1,200 QRCODE 160,120,H,10,A,0,X100,J5,"123456789" PRINT 1,1 </pre>	
<p><u>For other [justification] results (J1~J9)</u></p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div>	



6.18 RSS

Description

Draw a RSS barcode on the label format.

Syntax

```
RSS x,y, "sym",rotate,pixMult,sepHt, "content"
```

```
RSS x,y, "RSSEXP",rotate,pixMult,sepHt,segWidth, "content"
```

```
RSS x,y, "UCC128CCA",rotate,pixMult,sepHt,linHeight, "content"
```



```
RSS x,y, "UCC128CCC",rotate,pixMult,sepHt,linHeight, "content"
```




Parameter	Description																								
x	X-coordinate																								
y	Y-coordinate																								
sym	Symbology type: <table><tr><td>RSS14</td><td>RSS14</td></tr><tr><td>RSS14T</td><td>RSS14 Truncated</td></tr><tr><td>RSS14S</td><td>RSS14 Stacked</td></tr><tr><td>RSS14SO</td><td>RSS14 Stacked Omnidirectional</td></tr><tr><td>RSSLIM</td><td>RSS Limited</td></tr><tr><td>RSSEXP</td><td>RSS Expanded</td></tr><tr><td>UPCA</td><td>UPC-A</td></tr><tr><td>UPCE</td><td>UPC-E</td></tr><tr><td>EAN13</td><td>EAN-13</td></tr><tr><td>EAN8</td><td>EAN-8</td></tr><tr><td>UCC128CCA</td><td>UCC/EAN-128 & CC-A/B</td></tr><tr><td>UCC128CCC</td><td>UCC/EAN-128 & CC-C</td></tr></table>	RSS14	RSS14	RSS14T	RSS14 Truncated	RSS14S	RSS14 Stacked	RSS14SO	RSS14 Stacked Omnidirectional	RSSLIM	RSS Limited	RSSEXP	RSS Expanded	UPCA	UPC-A	UPCE	UPC-E	EAN13	EAN-13	EAN8	EAN-8	UCC128CCA	UCC/EAN-128 & CC-A/B	UCC128CCC	UCC/EAN-128 & CC-C
RSS14	RSS14																								
RSS14T	RSS14 Truncated																								
RSS14S	RSS14 Stacked																								
RSS14SO	RSS14 Stacked Omnidirectional																								
RSSLIM	RSS Limited																								
RSSEXP	RSS Expanded																								
UPCA	UPC-A																								
UPCE	UPC-E																								
EAN13	EAN-13																								
EAN8	EAN-8																								
UCC128CCA	UCC/EAN-128 & CC-A/B																								
UCC128CCC	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><tr><td>RSS14</td><td>33 x pixMult</td></tr><tr><td>RSS14T</td><td>13 x pixMult.</td></tr><tr><td>RSS14S</td><td>13 x pixMult.</td></tr><tr><td>RSS14SO</td><td>33 x pixMult.</td></tr><tr><td>RSSLIM</td><td>13 x pixMult.</td></tr><tr><td>RSSEXP</td><td>33 x pixMult.</td></tr><tr><td>EAN8</td><td>60 x pixMult.</td></tr><tr><td>EAN13</td><td>74 x pixMult.</td></tr><tr><td>UPCA</td><td>74 x pixMult.</td></tr><tr><td>UPCE</td><td>74 x pixMult.</td></tr></table>	RSS14	33 x pixMult	RSS14T	13 x pixMult.	RSS14S	13 x pixMult.	RSS14SO	33 x pixMult.	RSSLIM	13 x pixMult.	RSSEXP	33 x pixMult.	EAN8	60 x pixMult.	EAN13	74 x pixMult.	UPCA	74 x pixMult.	UPCE	74 x pixMult.				
RSS14	33 x pixMult																								
RSS14T	13 x pixMult.																								
RSS14S	13 x pixMult.																								
RSS14SO	33 x pixMult.																								
RSSLIM	13 x pixMult.																								
RSSEXP	33 x pixMult.																								
EAN8	60 x pixMult.																								
EAN13	74 x pixMult.																								
UPCA	74 x pixMult.																								
UPCE	74 x 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:


- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 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 30% respectively.

Example

Sample code	Result
<pre> SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSS14",0,2,2, "1234567890 ABCDEFGF" RSS 300,300,"RSS14T",90,2,2,"1234567890 ABCDE FG" RSS 300,300,"RSS14S",180,2,2,"1234567890 ABCD EFG" RSS 300,300, "RSS14SO",270,2,2, "1234567890 ABCDEFGF" PRINT 1,1 </pre>	
<pre> SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSSLIM",0,2,2, "1234567890 ABCDEFGF" RSS 300,300, "RSSEXP",90,2,2,22, "1234567890 ABCDEFGF" RSS 300,300, "UPCA",180,2,2, "1234567890 ABCDEFGF" RSS 300,300, "UPCE",270,2,2, "000 ABCDEFGF" PRINT 1,1 </pre>	

<pre> SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300,"EAN13",0,2,2,"123456789012 ABCDE FG" RSS 300,300,"EAN8",90,2,2,"1234567 ABCDEFG" RSS 300,300,"UCC128CCA",180,2,2,25,"123456789 0 ABCDEFG" RSS 300,300,"UCC128CCC",270,2,2,25,"123456789 0 ABCDEFG" PRINT 1,1 </pre>	
<pre> SIZE 100 mm, 100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,10, "RSSEXP",90,2,2,12, "8110106141411234562891101201212085010048 000214025610048000310123191000" PRINT 1 </pre>	
Example of UPCE mode	
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS REM UPCE Rule 1: 00abc0000hij = abhijc, where c = 0-2 RSS 10,10,"UPCE",0,2,2,"001200000456 ABCDEFG" RSS 210,10,"UPCE",0,2,2,"001210000456 ABCDEFG " RSS 410,10,"UPCE",0,2,2,"001220000456 ABCDEFG " PRINT 1 </pre>	

<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS REM UPCE Rule 2: 00abc00000ij = abciij3 RSS 10,10,"UPCE",0,2,2,"001230000045 ABCDEFGF" PRINT 1 SIZE 4,1 CLS REM UPCE Rule 3: 00abcd00000j = abcdj4 RSS 10,10,"UPCE",0,2,2,"001234000005 ABCDEFGF" PRINT 1 </pre>	
<pre> SIZE 4,1 GAP 0,0 DIRECTION 1 CLS REM UPCE Rule 4: 00abcde0000j = abcdej where j = 5-9 RSS 10,10,"UPCE",0,2,2,"001234500005 ABCDEFGF" RSS 160,10,"UPCE",0,2,2,"001234500006 ABCDEFGF" " RSS 310,10,"UPCE",0,2,2,"001234500007 ABCDEFGF" " RSS 460,10,"UPCE",0,2,2,"001234500008 ABCDEFGF" " RSS 610,10,"UPCE",0,2,2,"001234500009 ABCDEFGF" " PRINT 1 </pre>	
Example of barcode height of EAN8 EAN13 UPCA and UPCE	
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"EAN8",0,1,1,"1234567 ABCDEFGF" RSS 210,10, "EAN8",0,2,1,"1234567 ABCDEFGF" RSS 410,10, "EAN8",0,3,1,"1234567 ABCDEFGF" PRINT 1 </pre>	

<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"EAN13",0,1,1,"123456789012 ABCDEFGG" " RSS 210,10,"EAN13",0,2,1,"123456789012 ABCDEFGG" " RSS 410,10,"EAN13",0,3,1,"123456789012 ABCDEFGG" " PRINT 1 </pre>	
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"UPCA",0,1,1,"12345678901 ABCDEFGG" " RSS 210,10,"UPCA",0,2,1,"12345678901 ABCDEFGG" " RSS 410,10,"UPCA",0,3,1,"12345678901 ABCDEFGG" " PRINT 1 </pre>	
<pre> SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"UPCE",0,1,1,"001200000456 ABCDEFGG" " RSS 210,10,"UPCE",0,2,1,"001210000456 ABCDEFGG" " RSS 410,10,"UPCE",0,3,1,"001220000456 ABCDEFGG" " PRINT 1 </pre>	

6.19 REVERSE

Description

Reverse a region in image buffer.

Syntax


```
REVERSE x_start,y_start,x_width,y_height
```

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

Note:

- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 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 ratios are limited to 20% and 30% respectively.

Example

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

6.20 DIAGONAL

Description

Draw a diagonal.

Syntax


```
DIAGONAL x1, y1, x2, y2, thickness
```

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

Note:

- 203 dpi : 1 mm = 8 dots
300 dpi : 1 mm = 12 dots
600 dpi : 1 mm = 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>	

6.21 TEXT

Description

Print a text on label.

Syntax

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

Parameter	Description																																														
x	The x-coordinate of the text																																														
y	The y-coordinate of the text																																														
font	Font name <table><tr><td>0</td><td>Monotype CG Triumvirate Bold Condensed, font width and height is stretchable</td></tr><tr><td>1</td><td>8 x 12 fixed pitch dot font</td></tr><tr><td>2</td><td>12 x 20 fixed pitch dot font</td></tr><tr><td>3</td><td>16 x 24 fixed pitch dot font</td></tr><tr><td>4</td><td>24 x 32 fixed pitch dot font</td></tr><tr><td>5</td><td>32 x 48 dot fixed pitch font</td></tr><tr><td>6</td><td>14 x 19 dot fixed pitch font OCR-B</td></tr><tr><td>7</td><td>21 x 27 dot fixed pitch font OCR-B</td></tr><tr><td>8</td><td>14 x25 dot fixed pitch font OCR-A</td></tr><tr><td>ROMAN.TTF</td><td>Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.</td></tr><tr><td>1.EFT</td><td>EPL2 font 1</td></tr><tr><td>2.EFT</td><td>EPL2 font 2</td></tr><tr><td>3.EFT</td><td>EPL2 font 3</td></tr><tr><td>4.EFT</td><td>EPL2 font 4</td></tr><tr><td>5.EFT</td><td>EPL2 font 5</td></tr><tr><td>A.FNT</td><td>ZPL2 font A</td></tr><tr><td>B.FNT</td><td>ZPL2 font B</td></tr><tr><td>D.FNT</td><td>ZPL2 font D</td></tr><tr><td>E8.FNT</td><td>ZPL2 font E8</td></tr><tr><td>F.FNT</td><td>ZPL2 font F</td></tr><tr><td>G.FNT</td><td>ZPL2 font G</td></tr><tr><td>H8.FNT</td><td>ZPL2 font H8</td></tr><tr><td>GS.FNT</td><td>ZPL2 font GS</td></tr></table>	0	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable	1	8 x 12 fixed pitch dot font	2	12 x 20 fixed pitch dot font	3	16 x 24 fixed pitch dot font	4	24 x 32 fixed pitch dot font	5	32 x 48 dot fixed pitch font	6	14 x 19 dot fixed pitch font OCR-B	7	21 x 27 dot fixed pitch font OCR-B	8	14 x25 dot fixed pitch font OCR-A	ROMAN.TTF	Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.	1.EFT	EPL2 font 1	2.EFT	EPL2 font 2	3.EFT	EPL2 font 3	4.EFT	EPL2 font 4	5.EFT	EPL2 font 5	A.FNT	ZPL2 font A	B.FNT	ZPL2 font B	D.FNT	ZPL2 font D	E8.FNT	ZPL2 font E8	F.FNT	ZPL2 font F	G.FNT	ZPL2 font G	H8.FNT	ZPL2 font H8	GS.FNT	ZPL2 font GS
0	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable																																														
1	8 x 12 fixed pitch dot font																																														
2	12 x 20 fixed pitch dot font																																														
3	16 x 24 fixed pitch dot font																																														
4	24 x 32 fixed pitch dot font																																														
5	32 x 48 dot fixed pitch font																																														
6	14 x 19 dot fixed pitch font OCR-B																																														
7	21 x 27 dot fixed pitch font OCR-B																																														
8	14 x25 dot fixed pitch font OCR-A																																														
ROMAN.TTF	Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.																																														
1.EFT	EPL2 font 1																																														
2.EFT	EPL2 font 2																																														
3.EFT	EPL2 font 3																																														
4.EFT	EPL2 font 4																																														
5.EFT	EPL2 font 5																																														
A.FNT	ZPL2 font A																																														
B.FNT	ZPL2 font B																																														
D.FNT	ZPL2 font D																																														
E8.FNT	ZPL2 font E8																																														
F.FNT	ZPL2 font F																																														
G.FNT	ZPL2 font G																																														
H8.FNT	ZPL2 font H8																																														
GS.FNT	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.

alignment

Optional. Specify the alignment of text.

0: Default (Left)

1: Left

2: Center

3: Right

content

Content of text string

Note:

- Font "0" and "ROMAN.TTF" internal True Type Fonts are available in FBPL language printers.
- If there is any double quote (") within the text, make sure to use \["] instead of (").
- If font "0" is used, the font width and font height is stretchable by x-multiplication and y-multiplication parameter. It is expressed by point. 1 point=1/72inch.
- EPL2 and ZPL2 fonts are also supported.

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

Example

Continued on next page

Sample Code

```

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

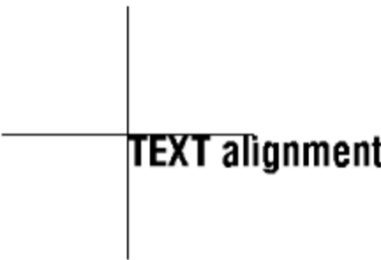
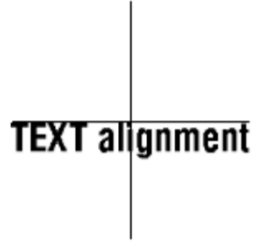
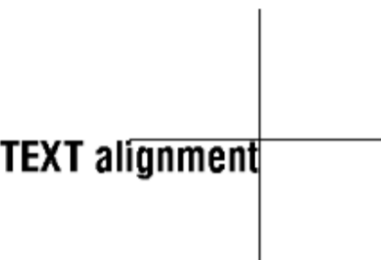
TEXT 400,10,"0",0,12,12,2,"EPL 2"
TEXT 400,40,"0",0,8,8,2,"align center"
TEXT 400,110,"1.EFT",0,1,1,2,"FONT 1"
TEXT 400,160,"2.EFT",0,1,1,2,"FONT 2"
TEXT 400,210,"3.EFT",0,1,1,2,"FONT 3"
TEXT 400,260,"4.EFT",0,1,1,2,"FONT 4"
TEXT 400,310,"5.EFT",0,1,1,2,"FONT 5"

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

```

Result

FBPL align left	EPL 2 align center	ZPL 2 align right
FONT 0	FONT 1	FONT 4
FONT 1	FONT 2	FONT 8
FONT 2	FONT 3	FONT D
FONT 3	FONT 4	FONT E8
FONT 4	FONT 5	FONT F
FONT 5	FONT 6	FONT G
FONT 6		FONT H8
FONT 7		© © TM ® ®
FONT 8		
FONT ROMAN.TTF		

<p>Sample Code</p> <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>	<p>Result</p> 
<p>Sample Code</p> <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>	<p>Result</p> 
<p>Sample Code</p> <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>	<p>Result</p> 

6.22 BLOCK

Description

Print a paragraph on the label.

Syntax

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

Parameter	Description																																														
x	The x-coordinate of the text																																														
y	The y-coordinate of the text																																														
width	The width of block for the paragraph in dots																																														
height	The height of block for the paragraph in dots																																														
font	<div><div>Font name</div><table><tr><td>0</td><td>Monotype CG Triumvirate Bold Condensed, font width and height is stretchable</td></tr><tr><td>1</td><td>8 x 12 fixed pitch dot font</td></tr><tr><td>2</td><td>12 x 20 fixed pitch dot font</td></tr><tr><td>3</td><td>16 x 24 fixed pitch dot font</td></tr><tr><td>4</td><td>24 x 32 fixed pitch dot font</td></tr><tr><td>5</td><td>32 x 48 dot fixed pitch font</td></tr><tr><td>6</td><td>14 x 19 dot fixed pitch font OCR-B</td></tr><tr><td>7</td><td>21 x 27 dot fixed pitch font OCR-B</td></tr><tr><td>8</td><td>14 x25 dot fixed pitch font OCR-A</td></tr><tr><td>ROMAN.TTF</td><td>Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.</td></tr><tr><td>1.EFT</td><td>EPL2 font 1</td></tr><tr><td>2.EFT</td><td>EPL2 font 2</td></tr><tr><td>3.EFT</td><td>EPL2 font 3</td></tr><tr><td>4.EFT</td><td>EPL2 font 4</td></tr><tr><td>5.EFT</td><td>EPL2 font 5</td></tr><tr><td>A.FNT</td><td>ZPL2 font A</td></tr><tr><td>B.FNT</td><td>ZPL2 font B</td></tr><tr><td>D.FNT</td><td>ZPL2 font D</td></tr><tr><td>E8.FNT</td><td>ZPL2 font E8</td></tr><tr><td>F.FNT</td><td>ZPL2 font F</td></tr><tr><td>G.FNT</td><td>ZPL2 font G</td></tr><tr><td>H8.FNT</td><td>ZPL2 font H8</td></tr><tr><td>GS.FNT</td><td>ZPL2 font GS</td></tr></table></div>	0	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable	1	8 x 12 fixed pitch dot font	2	12 x 20 fixed pitch dot font	3	16 x 24 fixed pitch dot font	4	24 x 32 fixed pitch dot font	5	32 x 48 dot fixed pitch font	6	14 x 19 dot fixed pitch font OCR-B	7	21 x 27 dot fixed pitch font OCR-B	8	14 x25 dot fixed pitch font OCR-A	ROMAN.TTF	Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.	1.EFT	EPL2 font 1	2.EFT	EPL2 font 2	3.EFT	EPL2 font 3	4.EFT	EPL2 font 4	5.EFT	EPL2 font 5	A.FNT	ZPL2 font A	B.FNT	ZPL2 font B	D.FNT	ZPL2 font D	E8.FNT	ZPL2 font E8	F.FNT	ZPL2 font F	G.FNT	ZPL2 font G	H8.FNT	ZPL2 font H8	GS.FNT	ZPL2 font GS
0	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable																																														
1	8 x 12 fixed pitch dot font																																														
2	12 x 20 fixed pitch dot font																																														
3	16 x 24 fixed pitch dot font																																														
4	24 x 32 fixed pitch dot font																																														
5	32 x 48 dot fixed pitch font																																														
6	14 x 19 dot fixed pitch font OCR-B																																														
7	21 x 27 dot fixed pitch font OCR-B																																														
8	14 x25 dot fixed pitch font OCR-A																																														
ROMAN.TTF	Monotype CG Triumvirate Bold Condensed, font width and height proportion is fixed.																																														
1.EFT	EPL2 font 1																																														
2.EFT	EPL2 font 2																																														
3.EFT	EPL2 font 3																																														
4.EFT	EPL2 font 4																																														
5.EFT	EPL2 font 5																																														
A.FNT	ZPL2 font A																																														
B.FNT	ZPL2 font B																																														
D.FNT	ZPL2 font D																																														
E8.FNT	ZPL2 font E8																																														
F.FNT	ZPL2 font F																																														
G.FNT	ZPL2 font G																																														
H8.FNT	ZPL2 font H8																																														
GS.FNT	ZPL2 font GS																																														
rotation	<div>The rotation angle of text 0 : No rotation 90 : degrees, in clockwise direction 180 : degrees, in clockwise direction 270 : degrees, in clockwise direction</div>																																														
x-multiplication	<div>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.</div>																																														

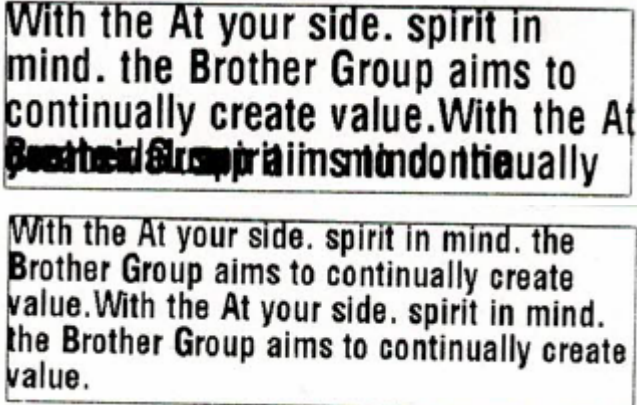
y-multiplication	Vertical multiplication, up to 10x Available factors: 1~10 For true type font, this parameter is used to specify the height (point) of true type font. 1 point=1/72 inch. For *.TTF font, x-multiplication and y-multiplication support floating value.
[space]	Add or delete the space between lines (in dots)
[align]	Text alignment. 0: Default (Left) 1: Left 2: Center 3: Right
[fit]	Shrink the text so that it fits in the block 0 : No shrink (default) 1: Shrink
content	Data in block. The maximum data length is 4092 bytes.

Note:

- Font 0 and ROMAN.TTF internal True Type Fonts are available in FBPL language printers.
- If there is any double quote (") within the text, make sure to use \["] instead of (").
- If font "0" is used, the font width and font height is stretchable by x-multiplication and y-multiplication parameter. It is expressed by point. 1 point=1/72inch.
- \[R] means carriage return character 0x0D.
- \[L] means line feed character 0x0A.
- EPL2 and ZPL2 fonts are also supported.

Example

Sample Code	Result
<pre> SIZE 4,0.5 GAP 0,0 DIRECTION 1 CLS BOX 10,10,800,100,2 BLOCK 15,15,790,90, "0",0,8,8," With the At your side. spirit in mind. the Brother Group aims to continually create value." PRINT 1 CLS BOX 10,10,800,100,2 BLOCK 15,15,790,90,"0",0,8,8,20,2," With the At your side. spirit in mind. the Brother Group aims to continually create value." PRINT 1 </pre>	<div>With the At your side. spirit in mind. the Brother Group aims to continually create value.</div> <div>With the At your side. spirit in mind. the Brother Group aims to continually create value.</div>

Sample Code for [fit] Parameter	Result
<pre> DATA\$ = "With the At your side. spirit in mind. the Brother Group aims to continually create value. With the At your side. spirit in mind. the Brother Group aims to continually create value." SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS BLOCK 20,20,500,170,"0",0,10,10,0,0,1 ,DATA\$ BOX 20,20,500+20,170+20,2 PRINT 1 SIZE 4,1.5 GAP 0,0 DIRECTION 1 CLS BLOCK 20,20,500,170,"0",0,10,10,0,0,0 ,DATA\$ BOX 20,20,500+20,170+20,2 PRINT 1 </pre>	

7 Status Polling and Immediate Commands

7.1 <ESC>!?

Description

Obtain 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

7.2 <ESC>!C

Description

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

Syntax

<ESC>!C

Note:

When the printer receives this command, the printer will restart itself no matter AUTO.BAS exists or not.

See Also

<ESC>!Q

7.3 <ESC>!D

Description

Disable an immediate command that starts with <ESC>!. (e.g. <ESC>!R <RSC>!? <ESC>!C)
The beginning of the command is an ESCAPE character (ASCII 27).

Syntax

<ESC>!D

See Also

~!E

7.4 <ESC>!O

Description

Cancel the PAUSE status of printer. The beginning of the command is an ESCAPE character (ASCII 27).

Syntax

<ESC>!O

See Also

<ESC>!P

7.5 <ESC>!P

Description

Pause the printer. The beginning of the command is an ESCAPE character (ASCII 27).

Syntax

<ESC>!P

See Also

<ESC>!O

7.6 <ESC>!Q

Description

Restart the printer and omit 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.

See Also

<ESC>!C

7.7 <ESC>!R

Description

Reset 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>!?

7.8 <ESC>!S

Description

Obtain 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

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	Status
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	Status
0	1	0	0	0	0	0	0	40	64	@	Normal
0	1	0	0	0	0	0	1	41	65	A	Paper low
0	1	0	0	0	0	1	0	42	66	B	Ribbon low
0	1	0	0	0	1	0	0	44	68	D	Reversed
0	1	0	0	1	0	0	0	48	72	H	Receive buffer full
0	1	1	0	0	0	0	0	60	96	`	Reversed

Status Byte #3: error											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	ASCII	Char	Status
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
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

Status Byte #4: error											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	ASCII	Char	Status
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

This example uses the Brother CommTool via RS-232 port.

The screenshot shows the Brother CommTool interface. The 'Transmis' tab is active, showing 'Transmissive way' set to RS232, 'Port' as 9100, and 'USB Device' as 'USB Input Device'. The 'Receive' tab shows 'Receive way' with RS232, USB, and Ethernet checked. The 'Status' section on the right shows COM, CTS, DSR, and RI status indicators. The 'Length' is set to 48 and 'Line' to 6. The 'RS232 Setup' section shows 'Length' as 3/3. The 'Hex Data' section at the bottom shows 'Hex Data' set to 1B 21 53. The 'Send Hex Data' button is highlighted with a red box. The 'Return values in Hex' section shows a list of hex values: 02 40 40 40 40 03 0D 0A, 02 46 40 40 40 03 0D 0A, 02 60 40 40 40 03 0D 0A, 02 42 40 40 40 03 0D 0A, 02 45 40 40 42 03 0D 0A, 02 45 40 40 62 03 0D 0A. The 'Return values in characters' section shows: @@@@, F@@@, \@@@, B@@@, E@@B, E@@b. A callout box explains: 'Hex Data 1B 21 53 means <ESC> ! S.' Another callout box says: 'Click to send Hex 1b 21 53 to request the printer status.'

Result

1: Start character, 2: 4-byte status in Hex, 3: End characters, 4: 4-byte status in characters

Item	1	2	3	4
Normal	0 2	4 0 4 0 4 0 4 0	0 3 0 D 0 A	@ @ @ @
Feed label	0 2	4 6 4 0 4 0 4 0	0 3 0 D 0 A	F @ @ @
Pause	0 2	6 0 4 0 4 0 4 0	0 3 0 D 0 A	\ @ @ @
Back feed label	0 2	4 2 4 0 4 0 4 0	0 3 0 D 0 A	B @ @ @
Error: Paper Jam	0 2	4 5 4 0 4 0 4 2	0 3 0 D 0 A	E @ @ B
Error: Paper jam and Head open	0 2	4 5 4 0 4 0 6 2	0 3 0 D 0 A	E @ @ b

Note:

Paper Jam <Hex 42>

Head Open <Hex 60>

0x42 | 0x60 = 62 <Hex b>

See Also

<ESC>!?

7.9 <ESC>!F

Description

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

7.10 <ESC>!.

Description

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

Syntax

<ESC>!.

7.11 ~!@

Description

Inquire 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

~!@

7.12 ~!A

Description

Inquire 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

7.13 ~!C

Description

Inquire the presence of Real Time Clock (RTC). One byte is return from the printer, indicating whether the RTC is installed.

Syntax

~!C

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

Example

~!C

7.14 ~!D

Description

Enter the printer into DUMP mode. In DUMP mode, the printer outputs code directly without interpretation.

Syntax

~!D

Example

~!D

7.15 ~!E

Description

Enable immediate command (e.g. <ESC>!R <RSC>! ? <ESC>!C), which is starting by <ESC>!.

Syntax

~!E

Example

~!E

See also

<ESC>!D

7.16 ~!F

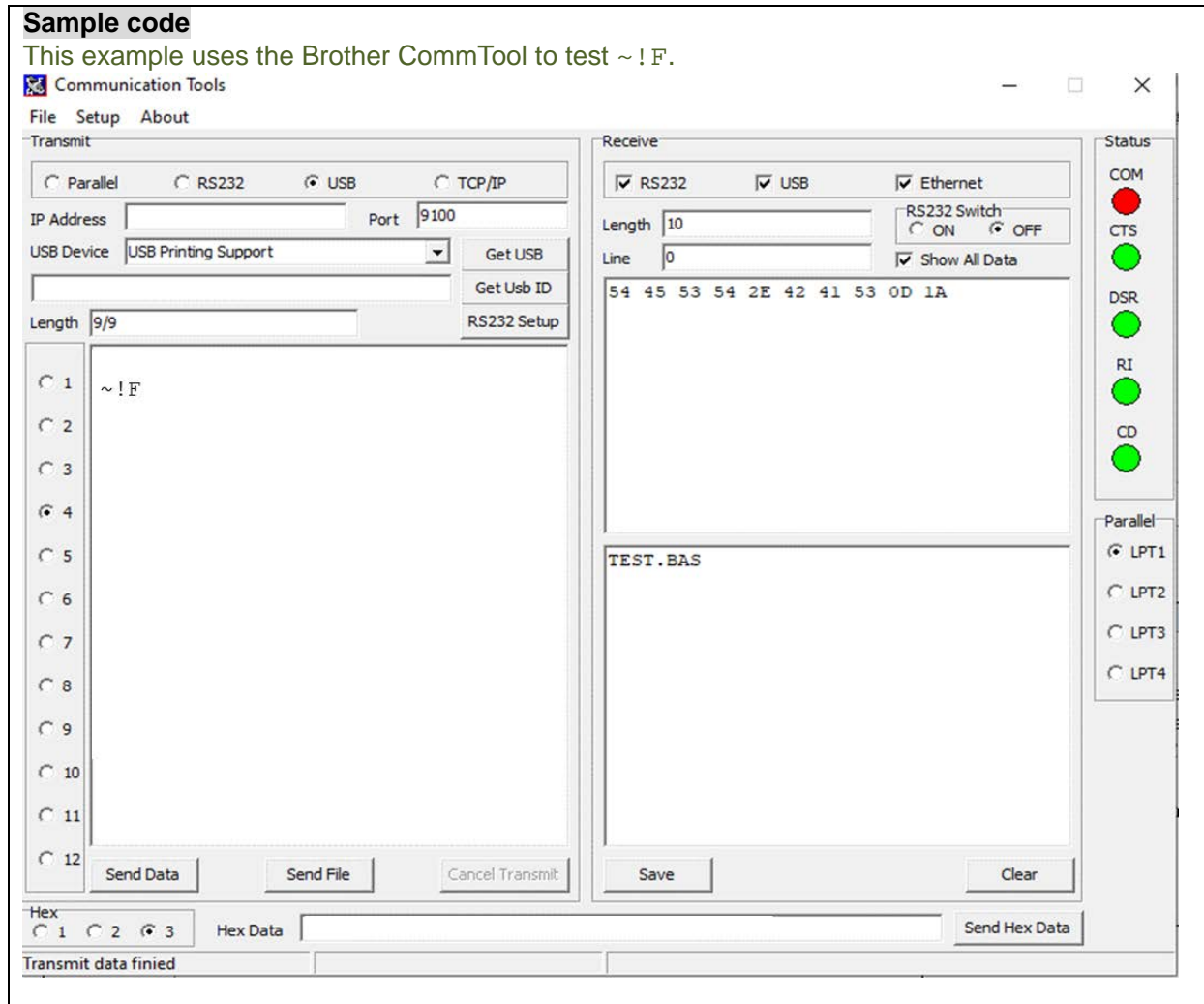
Description

Request all files resident in the printer memory and fonts installed in the memory module. The filename is 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



See Also

FILES

7.17 ~!I

Description

Inquire the code page and country setting of the printer.

Syntax

~!I

The returned information is given in the following format:

code page, country code

e.g. 8 bit: 437, 001

7 bit: USA, 001

For more information, see **CODEPAGE** command.

Example

~!I

See Also

CODEPAGE

7.18 ~!T

Description

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

Syntax

~ !T

Example

~ !T

7.19 <ESC>Y

Description

Enable Line Mode for EZC (CPCL) or EZP (ESC/POS) printer.

Syntax

<ESC>Y

Example

<ESC>Y

See Also

<ESC>Z

7.20 <ESC>Z

Description

Disable Line Mode for EZC (CPCL) or EZP (ESC/POS) printer.

Syntax

<ESC>Z

Example

<ESC>Z

See Also

<ESC>Y

8 Commands for Windows Driver

8.1 !B

Description

Store bitmap image data in the memory. Behind the `nnn` is the bitmap data.

Syntax

`!Bnnn`

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

Example

`!B100`

See Also

BITMAP

8.2 !J

Description

Print 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

8.3 !N

Description

Print a specified number of labels.

Syntax

!Nnnn

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

Example

!N001

9 File Management Commands

9.1 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 files that can be saved in the printer:

DRAM : 50 files
FLASH : 256 files

If "AUTO.BAS" exists in the printer memory, it runs automatically at startup. To disable the auto execution function, follow these steps:

1. Turn off the printer.
2. Press the following button, and then turn on the printer.
TD-4T, RJ, TJ-4020TN/TJ-4120TN Printers : **Feed/Pause** button
Other TJ series : Right **Selection** button
3. Release the button when the LED is lit in green.

Note:

The LED color will change in the following order:

TD-4420TN/4520TN

Amber (lit) → red (5 times) → amber (5 times) → green (5 times) → green/amber (5 times) → red/amber (5 times) → **green (lit)**

TD-4650TNWB/4750TNWB/4650TNWBR/4750TNWBR

Amber (lit) → red/amber (5 times) → amber (5 times) → green/red (5 times) → green (5 times) → red (5 times) → **green (lit)**

RJ-2035B/2055WB/3035B/3055WB

Amber (lit) → right/green (5 times) → center/green (5 times) → left/green (5 times) → **green (lit)**

TJ-4021TN/4021TNR/4121TN/4121TNR/4420TN/4520TN/4620TN/4422TN/4522TN

Amber (lit) → red (5 times) → amber (5 times) → green (5 times) → green/amber (5 times) → red/amber (5 times) → **green (lit)**

Syntax

1. Download a program file:

DOWNLOAD [n,] "FILENAME.BAS"

<u>Parameter</u>	<u>Description</u>
n	Specify memory used to save downloaded files. none: Download files to DRAM only. If you would like to save the files from DRAM to Flash memory before turning off power, issue MOVE command to printer. F: Download files to main board flash memory. E: Download files to expansion memory module.
FILENAME.BAS	The filename resident in the printer memory.
Note: <ul style="list-style-type: none">▪ Filenames are case sensitive.▪ File extensions must be ".BAS"▪ Filenames must be in 8.3 format.	

- Make sure to use with **EOP** command.
- If memory is not specified, all files will be downloaded to DRAM.
- The priority of AUTO.BAS in each memory device:
DRAM > CARD (Ext. FLASH) > FLASH
- No Battery is used to back up files in DRAM. Which will be lost in the event printer power is lost.

2. Download a data file:

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

Parameter	Description
n	Specify the memory location to save downloaded files. none: Download files to DRAM only. If you would like to save the files from DRAM to Flash memory before turning off power, issue MOVE command to printer. F: Download files to main board flash memory. E: Download files to expansion memory module.
FILENAME	The name of data file that will remain resident in the printer memory (case sensitive).
DATA SIZE	The actual size in bytes of the data file (without header)
DATA CONTENT	The data which will be downloaded into printer.
Note: <ul style="list-style-type: none"> ▪ 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. ▪ Call the main filename without BAS extension or use RUN command to start the download program. 	

Example

Sample code This is an example of how to 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 This is an example of how to 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

9.2 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

9.3 FILES

Description

Print 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
<code>FILES</code>	<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`

9.4 KILL

Description

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

Syntax

KILL [n], "FILENAME"

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

Note:

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

Model	Support		
	KILL "*"	KILL "*" MOVE	KILL F, "*"
FBPL programming printer	V		V

Example

Users can use printer **SELFTEST** utility to print the 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

9.5 MOVE

Description

Move the downloaded files from DRAM to FLASH memory.

Syntax

MOVE

See Also

DOWNLOAD, EOP

9.6 RUN

Description

Execute a program resident in the printer memory. It is available for FBPL language printers only.


Syntax

```
RUN "FILENAME.BAS"
```

Note:

This command can be replaced to filename that without typing ".BAS".

Example

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

See Also

DOWNLOAD, EOP

10 BASIC Commands and Functions

10.1 ABS()

Description

Return 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

10.2 ASC()

Description

Return 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\$()

10.3 CHR\$()

Description

Return 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\$()

10.4 XOR\$()

Description

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.

Example

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

10.5 END

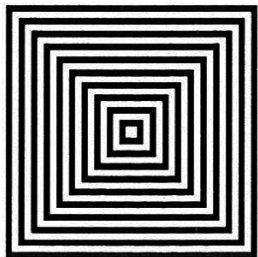
Description

State 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

10.6 EOF()

Description

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

```
DOWNLOAD "DATA",16,COMPUTER
2000

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

Result

*****EOF TEST*****

COMPUTER\$2000[EOF(0)=1]



PRICE-2000

See Also

DOWNLOAD, EOP, OPEN, READ, SEEK

10.7 OPEN

Description

Open a downloaded file and establishes the file handle. 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.

Note:

If the file doesn't exist, the printer will create this file in the onboard FLASH.

Syntax

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

Parameter	Description								
[memory ID]	Optional. Open the file in specific memory device. <table><tr><th>ID</th><th>Memory device</th></tr><tr><td>Omitted</td><td>DRAM</td></tr><tr><td>F</td><td>FLASH</td></tr><tr><td>E</td><td>CARD</td></tr></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." 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\$</pre>	<pre>Open file in DRAM. Open file in FLASH. Auto create a new file in FLASH.</pre>

PRINT 1 EOP TEST	
------------------------	--

See Also

DOWNLOAD, EOP, READ, WRITE, SEEK, CLOSE

10.8 CLOSE

Description

Close the file handle which is open by **OPEN** command.

Syntax

```
CLOSE file handle
```

<u>Parameter</u>	<u>Description</u>
file handle	Either 0 or 1

Example

See the example in **OPEN** command.

10.9 WRITE

Description

Write data to a downloaded data file. Two files can be open simultaneously.

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\$()

10.10 READ

Description




Read data from a downloaded data file.

Syntax

```
READ file handle,variables
```

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

Example

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

See Also

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

10.11 SEEK

Description

Shift the specified file pointer to a certain position.

Syntax

```
SEEK file handle,offset
```

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

Example

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

See Also

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

10.12 LOF()

Description

Return the size of the specified file.

Syntax

LOF ("FILENAME")

<u>Parameter</u>	<u>Description</u>
FILENAME	The file downloaded in the printer memory.

Example

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

See Also

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

10.13 LOC()

Description

Return the current read/write position within an open file.

Syntax

`LOC(file handle)`

<u>Parameter</u>	<u>Description</u>
file handle	0 or 1

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>

10.14 FREAD\$()

Description

Read 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

<u>Sample code</u>	<u>Result</u>
<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

10.15 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

Example

Sample code

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

Result

```
Original data in DATA1: 1234567890  
New data in Data1: 1234567890aB1
```

See Also

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

10.16 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

Example

Sample code

```
DOWNLOAD "DATA1",10,1234567890  
DOWNLOAD "TEST.BAS"  
a$=" "  
b$=" "  
c=0  
d$=" "  
e$=" "  
OPEN "DATA1",0  
SEEK 0,0  
GET 0,a$,b$,c  
SEEK 0,0  
FOR I=1 TO 5  
GET 0,d$  
e$=e$+d$  
NEXT  
  
SIZE 4,0.5  
GAP 0,0  
CLS  
TEXT 10,10,"3",0,1,1,"The first 3 characters in DATA1: "+ a$+b$+"  
("+STR$(c)+")"  
TEXT 10,60,"3",0,1,1,"The first 5 characters in DATA1: "+e$  
PRINT 1  
EOP  
TEST
```

Result

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

See Also

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

10.17 COPY

Description

Copy the existed file from CARD to FLASH.

Syntax

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

Parameter	Description								
memory ID of source	Optional. <table><tr><th>ID</th><th>Memory device</th></tr><tr><td>none</td><td>DRAM</td></tr><tr><td>F</td><td>FLASH</td></tr><tr><td>E</td><td>CARD</td></tr></table>	ID	Memory device	none	DRAM	F	FLASH	E	CARD
ID	Memory device								
none	DRAM								
F	FLASH								
E	CARD								
source filename	The file in CARD which you want to copy to on board FLASH.								
Memory ID of new file	Optional. <table><tr><th>ID</th><th>Memory device</th></tr><tr><td>none</td><td>DRAM</td></tr><tr><td>F</td><td>FLASH</td></tr></table>	ID	Memory device	none	DRAM	F	FLASH		
ID	Memory device								
none	DRAM								
F	FLASH								
new filename	The new filename you want to use in the on board FLASH.								

Example

Sample Code

```
DOWNLOAD "DATA_D.DAT",105, With the "At your side." spirit in mind. the Brother
Group aims to continually create value.
DOWNLOAD "TEST.BAS"
KILL F,"*"
COPY "DATA_D.DAT",F,"DATA_F.DAT"
OPEN "DATA_F.DAT",0
SEEK 0,0
data$=FREAD$(0,LOF("DATA_F.DAT"))
CLOSE 0
SIZE 4,0.5
GAP 0,0
CLS
BOX 10,10,800,100,2
BLOCK 15,15,790,90,"0",0,8,8,20,2,data$
PRINT 1
EOP
TEST
```

Result

203 dpi

With the "At your side." spirit in mind, the Brother Group aims to continually create value.

300 dpi

With the "At your side." spirit in mind, the Brother Group
aims to continually create value.

See Also

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

10.18 FOR...NEXT LOOP

Description

Repeat one or more lines of program a specified number of times. 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	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

<u>Sample code</u>	<u>Result</u>
<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) 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</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>

See Also

DOWNLOAD, EOP

10.19 WHILE...WEND

Description

Execute a series of statements if a given condition is True. Nested loops are allowed (up to 39 nested loops) in the printer.

Syntax

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

<u>Parameter</u>	<u>Description</u>
condition	Available relational operator: <, >, =, <=, >=, <> *Relational operator <>, not equal, was supported.
Statement	One or more statements executed while condition is True.

Example

Sample Code	Result
DOWNLOAD "TEST.BAS" I=0 TOTAL=0 WHILE I<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	1+2+3+ ... + 100 = 5050
DOWNLOAD "TEST.BAS" data\$ = "" SIZE 4,0.3 GAP 0,0 DIRECTION 1 INPUT "Data: ",data\$ WHILE data\$ <> "Quit" CLS TEXT 10,10, "3",0,1,1, "Data: "+data\$ PRINT 1 INPUT "Data: ",data\$ WEND CLS TEXT 10,10, "3",0,1,1, "Quit BAS" PRINT 1 EOP TEST 12345 67890 quit Quit	Quit BAS Data: quit Data: 67890 Data: 12345

10.20 DO...LOOP

Description

Repeat 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: <, >, =, <=, >=, <> *Relational operator <>, not equal.
Statement	One or more statements executed while condition is True.
EXITDO	Exit loop

Example

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

<pre> DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO TOTAL=TOTAL+I I=I+1 LOOP WHILE I<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>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>	$1+2+3+ \dots + 100 = 5050$

10.21 IF...THEN...ELSE...ENDIF LOOP

Description

Execute one or more statements conditionally. Either a single-line syntax or multiple-line “block” syntax can be used.

Syntax

IF condition THEN statement

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

Or

```
IF condition THEN
    Statements
ENDIF
```

Or

```
IF condition THEN
    Statements
ELSE
    Statements
ENDIF
```

Or

```
IF condition 1 THEN
    Statement block 1
ELSEIF condition 2 THEN
    Statement block 2
...
ELSEIF condition n THEN
    Statement block n
ENDIF
```

Note:

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

<u>Parameter</u>	<u>Description</u>
condition	Available relational operator: <, >, =, <=, >=, <> *Relational operator <>, not equal.
Statement	Only one statement is available in

Example

Sample Code	Result
<pre> DOWNLOAD "DEMO.BAS" SIZE 4,4 GAP 0,0 DIRECTION 1 CLS A=0 B=0 C=0 D=0 E=0 F=0 G=0 H=0 J=0 K=0 L=0 FOR I=1 TO 100 IF I-INT(I/1)*1=0 THEN A=A+I IF I-INT(I/2)*2=1 THEN B=B+I ELSE C=C+I IF I-INT(I/3)*3=0 THEN D=D+I ENDIF IF I-INT(I/5)*5=0 THEN E=E+I ELSE F=F+I ENDIF IF I-INT(I/7)*7=0 THEN G=G+I ELSEIF I-INT(I/17)*17=0 THEN H=H+I ELSEIF I-INT(I/27)*27=0 THEN J=J+I ELSEIF I-INT(I/37)*37=0 THEN K=K+I ELSE L=L+I ENDIF NEXT 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) </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> 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 </pre>	
<pre> DOWNLOAD F, "TEST.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 CLS A=85 B=10 :START IF A<100 THEN GOTO L1 ELSE GOTO L2 :L1 CLS TEXT 100,10,"3",0,1,1,STR\$(A) + " IS SMALLER THEN 100" PRINT 1 A=A+B GOTO START ENDIF :L2 CLS TEXT 100,10,"3",0,1,1,STR\$(A) + " IS LAGER THEN 100" PRINT 1 EOP TEST </pre>	<p>105 IS LAGER THEN 100</p> <p>95 IS SMALLER THEN 100</p> <p>85 IS SMALLER THEN 100</p>

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

10.22 GOSUB...RETURN

Description

Branch to a subroutine, executing statements until RETURN is reached.

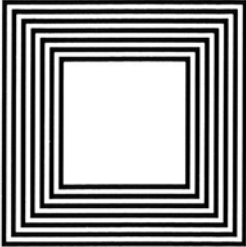
Syntax

```
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

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

See Also

DOWNLOAD, EOP, END, GOTO

10.23 GOTO

Description

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<100 THEN GOTO SUM ELSE GOTO PRTOUT ENDIF :SUM A=A+1 TOTAL=TOTAL+A GOTO START :PRTOUT B\$="THE SUMMATION OF 1..100 IS "+STR\$(TOTAL) TEXT 10,100, "3",0,1,1,B\$ PRINT 1 END EOP</pre>	<p>THE SUMMATION OF 1..100 IS 5050</p>

See Also

DOWNLOAD, EOP, END, GOSUB...RETURN

10.24 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

```
DOWNLOAD "TEST.BAS"

T$=" "
FOR I=1 TO 5
T$=T$+INP$(1)
NEXT

SIZE 4,0.5
GAP 0,0
CLS
TEXT 10,10, "3",0,1,1, "The received data is: "+T$
PRINT 1
EOP
TEST
12345
```

Result

The received data is: 12345

See Also

INP()

10.25 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

Example

Sample code

```
DOWNLOAD "TEST.BAS"
```

```
152sci=0
```

```
str$=" "
```

```
FOR I=1 TO 5
```

```
152sci=INP(1)
```

```
str$=str$+" " +STR$(152sci)
```

```
OUT 152sci
```

```
NEXT
```

```
SIZE 4,0.5
```

```
GAP 0,0
```

```
CLS
```

```
TEXT 10,10, "3",0,1,1, "The received data is: "+str$
```

```
PRINT 1
```

```
EOP
```

```
TEST
```

```
12345
```

Result

The received data is: 49 50 51 52 53

See Also

`INP$()`

10.26 LOB()

Description

Return the size of data in receiving buffer.

Syntax

LOB ()

Example

Sample Code

```
DOWNLOAD "TEST.BAS"

DATA$= " "

WHILE LOB() <> 0
DATA$=DATA$+INP$(1)
WEND

SIZE 4,0.5
GAP 0,0
CLS
BOX 10,10,800,100,2
BLOCK 15,15,790,90, "0",0,8,8,DATA$
PRINT 1
EOP
TEST
With the "At your side." spirit in mind. the Brother Group aims to
continually create value.
```

Result

203 dpi:

With the "At your side." spirit in mind, the Brother Group aims to continually create value.

300 dpi:

With the "At your side." spirit in mind, the Brother Group
aims to continually create value.

See Also

INP\$(), WHILE ... WEND

10.27 INPUT

Description

Receive data sent externally.

Syntax



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

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

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

Parameter	Description
Prompt string	The 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 as template	Result
<pre>DOWNLOAD F,"TEXT.BAS" SIZE 4,3 GAP 0,0 DIRECTION 1 :START INPUT "CODE 39 : ",C39\$ INPUT "EAN 13: ",12,E13\$ CLS TEXT 20,50, "3",0,1,1, PLC or Barcode Scanner 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> Sample code as PLC or Barcode Scanner <pre>TEXT 123456 123456789012</pre>	<p>PLC or Barcode Scanner Test</p>  <p>123456</p>  <p>1 234567 890128</p>

See Also

DOWNLOAD, EOP, END, GOTO

10.28 PREINPUT

Description

Define the start character for **INPUT** command.

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

Example

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

See also

POSTINPUT, INPUT, SET FILTER

10.29 POSTINPUT

Description

Define the end character for **INPUT** command.

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

Example

```
POSTINPUT ">"
```

```
POSTINPUT CHR$(3)
```

See also

PREINPUT, INPUT, SET FILTER

10.30 SET FILTER ON/OFF

Description

Enable/disable **PREINPUT** and **POSTINPUT** commands.

Syntax

```
SET FILTER ON/OFF
```

<u>Parameter</u>	<u>Description</u>
ON	Enable PREINPUT and POSTINPUT
OFF	Disable PREINPUT and POSTINPUT

Example

<u>Sample Code</u>	<u>Result</u>
<pre>DOWNLOAD "TEST.BAS" PREINPUT "<=" POSTINPUT "=>" 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 <=1234=><=5678=><=9012=></pre>	<pre>DATA = 9012 DATA = 5678 DATA = 1234</pre>

See also

PREINPUT, POSTINPUT, INPUT

10.31 REM

Description

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

Syntax

REM

Example

Sample code

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

Result

REMARK DEMO PROGRAM

See Also

DOWNLOAD, EOP, END

10.32 OUT

Description

Return data through the specific port.

Syntax

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

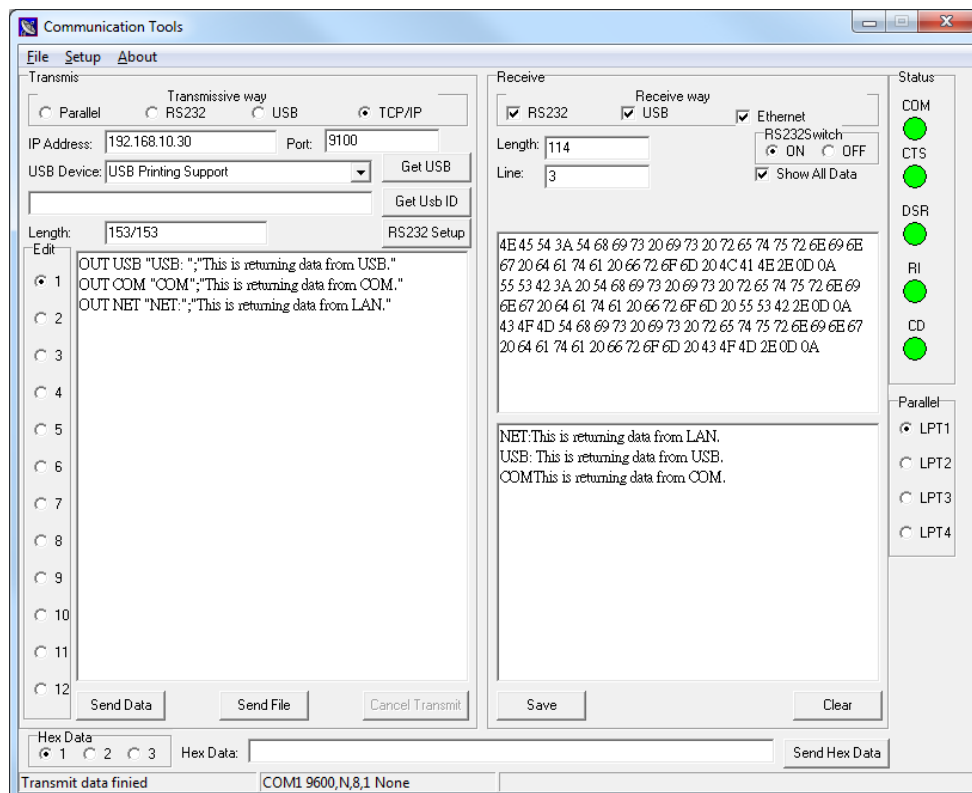
Parameter	Description
port	Optional. Specified the port for returning data/string. Default is returning the data/string from the port which is sending data to printer. COM: Returning data/string from COM port. USB: Returning data/string from USB port. NET: Returning data/string from LAN port.
Prompt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

Example

Sample Code

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

Result



10.33 OUTR

Description

Send data through RS-232 port only.

Syntax

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

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

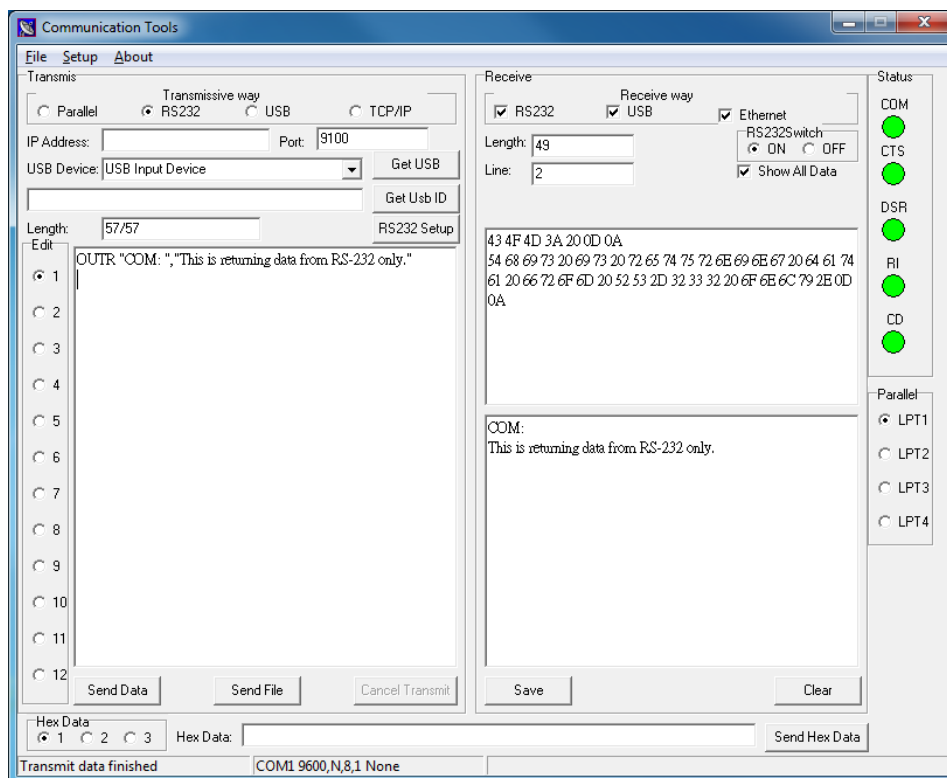
Parameter	Description
prompt	Prompt string.
Variable	The output message.
,	The "prompt" and "variable" are separated by <0x0D><0x0A>.
;	The "variable" comes behinds "prompt" directly.

Example

Sample Code

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

Result



10.34 GETKEY()

Description

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

Syntax

GETKEY ()

PAUSE	FEED
0	1

Example

Sample code

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

See Also

DOWNLOAD, EOP, END, GOTO

10.35 INT()

Description

Truncate 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

<u>Sample code</u>	<u>Result</u>
<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>Num THEN TEXT 50,100, "3",0,1,1, "To round up= " +STR\$(N) ELSE TEXT 50,100, "3",0,1,1, "To round down= " +STR\$(N) ENDIF PRINT 1 EOP 56.2</pre>	<p>To round down= 56</p>

See Also

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

10.36 LEFT\$()

Description

Return 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

<u>Sample code</u>	<u>Result</u>
<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\$()

10.37 LEN()

Description

Return 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()

10.38 MID\$()

Description

Retrieve 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>
<code>string</code>	The string to be processed
<code>m</code>	The beginning of m^{th} characters in the string $1 \leq m \leq \text{string length}$
<code>n</code>	The number of characters to return

Example

<u>Sample code</u>	<u>Result</u>
<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 10,200, "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()

10.39 RIGHT\$()

Description

Return a specified number of characters up from the end of a string.

Syntax

RIGHT\$(X\$,n)

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

Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="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()

10.40 STR\$()

Description

Convert a specified value or expression into corresponding string of characters.

Syntax

STR\$(n)

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

Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHIJKLMNOPQRSTUVWXYZ" 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>ABCDEFGHIJKLMNOPQRSTUVWXYZ F=100 G=500 F+G=600</pre>

See Also

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

10.41 STRCOMP()

Description

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

Syntax

`STRCOMP(str1$,str2$[,comp])`

Parameter	Description
str1\$	Required. Any valid string expression.
str2\$	Required. Any valid string expression.
Comp	Optional. Specifies the type of string comparison. 0: Binary comparison. Default. 1: Textual comparison. The comparison is case-insensitive .

Condition	Return value
str1\$ sorts ahead of str2\$	-1
str1\$ is equal to str2\$	0
str1\$ sorts after str2\$	1

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()

10.42 INSTR()

Description

Return 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. String expression being searched.
Str2\$	Required. String expression sought.

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()

10.43 TRIM\$()

Description

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

Syntax

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

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

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\$()

10.44 LTRIM\$()

Description

Remove leading blank space from a string.

Syntax

`LTRIM$(str$[,list$])`

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

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$()`, `RTRIM$()`

10.45 RTRIM\$()

Description

Remove trailing blank space from a string.

Syntax

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

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

Example

Sample Code

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

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

Result

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

See Also

TRIM\$(), LTRIM\$()

10.46 TEXTPIXEL()

Description


Return the width of the text string in dots.

Syntax

`TEXTPIXEL(cont$,font$,size)`

<u>Parameter</u>	<u>Description</u>
cont\$	Content of text string.
Font \$	Font type. See the font parameter in TEXT command.
Size	Font size. See the x-multiplication parameter in TEXT command.

Example

<u>Sample code</u>	<u>Result</u>
<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 TEST</pre>	

See Also

TEXT, BARCODEPIXEL()

10.47 BARCODEPIXEL()

Description

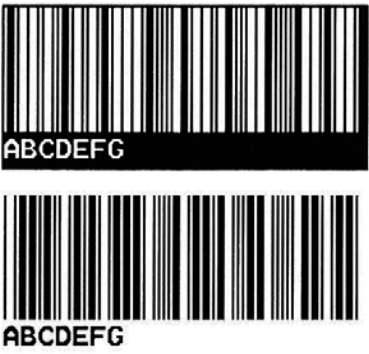
Return the width of the barcode in dots.

Syntax

```
BARCODEPIXEL(cont$, sym$, narrow, wide)
```

Parameter	Description
cont\$	The content of barcode.
Sym \$	Barcode type. See the <code>code type</code> parameter in BARCODE command.
Narrow	The width of narrow bar. See the <code>narrow</code> parameter in BARCODE command.
Wide	The width of wide bar. See the <code>wide</code> parameter in BARCODE command.

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\$ 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()

10.48 VAL()

Description

Convert numeric characters into corresponding integer or floating point number.

Syntax

```
VAL("numeric character")
```

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

Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 A\$="ABCDEFGHIJKLMNOPQRSTUVWXYZ" 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> ABCDEFGHIJKLMNOPQRSTUVWXYZ F=100 G=500 F+G=600</pre>

See Also

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

10.49 NOW\$()

Description

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

Syntax

NOW\$()

Example

Sample code

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

Result

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

See Also

FORMAT\$()

10.50 NOW

Description

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

Syntax

NOW

Example

Sample Code

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

Result

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

See Also

FORMAT\$(), DATEADD(), NOW

10.51 FORMAT\$()

Description

Return the current date, time, number and number value according to the printer setting.

Syntax

`FORMAT$(expression[,style$])`

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

Predefined date/time	Description
General Date	Date and time
Long Date	Long Date format
Medium Date	dd-mmm-yy format
Short Date	Short Date format
Long Time	Hour, minute, second, and "AM" or "PM" (h:mm:ss format)
Medium Time	Hour, minute, and "AM" or "PM" (hh:mm AM/PM format)
Short Time	Hour and minute (hh:mm format)

User-defined date/time	Description
c	Date (dddddd) and time as (tttt)
d	Day as a number without a leading zero (1 – 31).
dd	Day as a number with a leading zero (01 – 31).
ddd	Day as an abbreviation (Sun – Sat).
dddd	Day as a full name (Sunday – Saturday).
dddddd	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	Day 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	Day of the week as a number (1 for Sunday through 7 for Saturday).
ww	Week of the year as a number (1 – 53).
m	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	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	Month as an abbreviation (Jan – Dec).
mmmm	Month as a full month name (January – December).
q	Quarter of the year as a number (1 – 4).
y	Day of the year as a number (1 – 366).
yy	Year as a 2-digit number (00 – 99).
yyyy	Year as a 4-digit number (100 – 9999).
h	Hour as a number without leading zeros (0 – 23).
hh	Hour as a number with leading zeros (00 – 23).
n	Minute as a number without leading zeros (0 – 59).
nn	Minute as a number with leading zeros (00 – 59).
s	Second as a number without leading zeros (0 – 59).
ss	Second as a number with leading zeros (00 – 59).

ttttt	Time as a complete time (including hour, minute, and second). The default time format is h:mm:ss AM/PM.
AM/PM	Uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.
am/pm	Lowercase AM with any hour before noon; display a lowercase PM with any hour between noon and 11:59 P.M.
A/P	Uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.
a/p	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.
\	Next character in the format string.
"string"	String inside the double quotation marks.

Number	Description
General Number	Number as entered, with no rounding and no commas.
Currency	Number with a dollar sign, comma (if appropriate), and two digits to the right of the decimal point. Shows negative numbers inside parentheses.
Fixed	Number with at least one digit to the left of the decimal separator and two digits to the right. Does not show comma.
Standard	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	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	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.
\	Next character in the format string.
"ABC"	String inside the double quotation marks.

Different formats for different number values	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.

Example

Sample Code	Result
<pre> SIZE 800 dot,1900 dot GAP 0,0 DIRECTION 1 CLS TEXT 15,10, "3",0,1,1, "General Date: "+FORMAT\$(NOW,"General Date") TEXT 15,60, "3",0,1,1, "Long Date: "+FORMAT\$(NOW,"Long Date") TEXT 15,110, "3",0,1,1, "Medium Date: "+FORMAT\$(NOW,"Medium Date") TEXT 15,160, "3",0,1,1, "Short Date: "+FORMAT\$(NOW,"Short Date") TEXT 15,210, "3",0,1,1, "Long Time: "+FORMAT\$(NOW,"Long Time") TEXT 15,260, "3",0,1,1, "Medium Time: "+FORMAT\$(NOW,"Medium Time") TEXT 15,310, "3",0,1,1, "Short Time: "+FORMAT\$(NOW,"Short Time") TEXT 15,360, "3",0,1,1, "c: "+FORMAT\$(NOW,"c") TEXT 15,410, "3",0,1,1, "d: "+FORMAT\$(NOW,"d") TEXT 15,460, "3",0,1,1, "dd: " +FORMAT\$(NOW,"dd") TEXT 15,510, "3",0,1,1, "ddd: " +FORMAT\$(NOW,"ddd") TEXT 15,560, "3",0,1,1, "dddd: " +FORMAT\$(NOW,"dddd") TEXT 15,610, "3",0,1,1, "dddddd: " +FORMAT\$(NOW,"dddddd") TEXT 15,660, "3",0,1,1, "ddddddd: " +FORMAT\$(NOW,"ddddddd") TEXT 15,710, "3",0,1,1, "w: " +FORMAT\$(NOW,"w") TEXT 15,760, "3",0,1,1, "ww: " +FORMAT\$(NOW,"ww") TEXT 15,810, "3",0,1,1, "m: " +FORMAT\$(NOW,"m") TEXT 15,860, "3",0,1,1, "mm: " +FORMAT\$(NOW,"mm") TEXT 15,910, "3",0,1,1, "mmm: " +FORMAT\$(NOW,"mmm") TEXT 15,960, "3",0,1,1, "mmmm: " +FORMAT\$(NOW,"mmmm") TEXT 15,1010, "3",0,1,1, "q: " +FORMAT\$(NOW,"q") TEXT 15,1060, "3",0,1,1, "y: " +FORMAT\$(NOW,"y") TEXT 15,1110, "3",0,1,1, "yy: " +FORMAT\$(NOW,"yy") TEXT 15,1160, "3",0,1,1, "yyyy: " +FORMAT\$(NOW,"yyyy") TEXT 15,1210, "3",0,1,1, "h: " +FORMAT\$(NOW,"h") TEXT 15,1260, "3",0,1,1, "hh: " +FORMAT\$(NOW,"hh") TEXT 15,1310, "3",0,1,1, "n: " +FORMAT\$(NOW,"n") TEXT 15,1360, "3",0,1,1, "nn: " +FORMAT\$(NOW,"nn") TEXT 15,1410, "3",0,1,1, "s: " +FORMAT\$(NOW,"s") TEXT 15,1460, "3",0,1,1, "ss: " +FORMAT\$(NOW,"ss") TEXT 15,1510, "3",0,1,1, "ttttt: " +FORMAT\$(NOW,"ttttt") TEXT 15,1560, "3",0,1,1, "AM/PM: " +FORMAT\$(NOW,"AM/PM") TEXT 15,1610, "3",0,1,1, "am/pm: " +FORMAT\$(NOW,"am/pm") TEXT 15,1660, "3",0,1,1, "A/P: " +FORMAT\$(NOW,"A/P") TEXT 15,1710, "3",0,1,1, "a/p: " +FORMAT\$(NOW,"a/p") TEXT 15,1760, "3",0,1,1, "AMPM: " +FORMAT\$(NOW,"AMPM") TEXT 15,1810, "3",0,1,1, "\:" +FORMAT\$(NOW,"To\da\y i\s ddddd") TEXT 15,1860, "3",0,1,1, "string: " +FORMAT\$(NOW,"To\da\y i\s dddddd") PRINT 1 </pre>	<pre> General Date:1/9/2013 2:46:18 PM Long Date:Tuesday, January 09 2013 Medium Date:09-Jan-13 Short Date:1/9/2013 Long Time:2:46:18 PM Medium Time:02:46 PM Short Time:14:46 c:1/9/2013 2:46:18 PM d:9 dd:09 ddd:Tue dddd:Tuesday dddddd:1/9/2013 ddddddd:Tuesday, January 09 2013 u:3 uu:2 m:1 mm:01 mmm:Jan mmmm:January q:1 y:9 yy:13 yyyy:2013 h:14 hh:14 n:46 nn:46 s:18 ss:18 ttttt:2:46:18 PM AM/PM:PM am/pm:pm A/P:P a/p:p AMPM:PM \:Today is 1/9/2013 string:Today is 1/9/2013 </pre>

Sample Code	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>

See Also

NOW\$(), DATEADD(), NOW

10.52 DATEADD()

Description

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

Syntax

`DATEADD(interval$,number,date)`

Parameter	Description																						
interval\$,	The time/date interval for adding. It can be one of following values. <table><tr><th>Interval\$</th><th>The interval unit of parameter interval\$</th></tr><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></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																						

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

10.53 FSEARCH()

Description

Return 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.

Example

<u>Sample Code</u>	<u>Result</u>
<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>

10.54 TOUCHPRESS()

Description

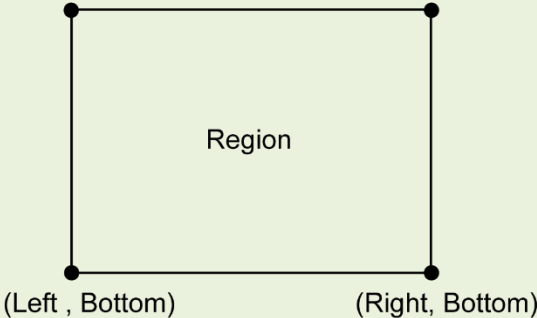
Detect the status of the touch screen. Return 1 if the touch screen for the specified region is pressed, otherwise returns 0.

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)

(Left , Top) (Right, Top)



(Left , Bottom) (Right, Bottom)

Region

Note: TJ-4021TN/TJ-4021TNR/TJ-4121TN/TJ-4121TNR/TJ-4422TN/TJ-4522TN only.

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
```

10.55 RECORDSET\$()

Description

Return a value from a table. Table is represented in a grid format, tabular form in rows and columns. For more information, see the following table format in the **example**.

Syntax

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

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

Note:

ROW is always a number. COLUMN can be a number or name.

Example

Sample Code 1:

```
DOWNLOAD F,"TEST.CSV",75,3
Name,Age,Height,Weight
John,18,180,80
Mary,30,150,50
Mark,65,170,65

DOWNLOAD F,"TEST.BAS"
CLOSE 0
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 100,50,"3",0,1,1,"Row 1 and Column 1 = "
+ RECORDSET$("TEST.CSV", 1, 1, ASC(", "))
TEXT 100,100,"3",0,1,1,"Row 2 and Column 1 = "
+ RECORDSET$("TEST.CSV", 2, 1, ASC(", "))
TEXT 100,150,"3",0,1,1,"John Age = " +
RECORDSET$("TEST.CSV", 1, 2, ASC(", "))
TEXT 100,200,"3",0,1,1,"Mary Age = " +
RECORDSET$("TEST.CSV", 2, 2, ASC(", "))
TEXT 100,250,"3",0,1,1,"John Height = " +
RECORDSET$("TEST.CSV",1,"Height", ASC(", "))
TEXT 100,300,"3",0,1,1,"Mary Height = " +
RECORDSET$("TEST.CSV",2,"Height", ASC(", "))
PRINT 1

EOP
TEST
```

Result

Row 1 and Column 1 = John

Row 2 and Column 1 = Mary

John Age = 18

Mary Age = 30

John Height = 180

Mary Height = 150

Table format (TEST.CSV)

Number of rows

3

Name of column

	Name	Age	Height	Weight
Row 1	John	18	180	80
Row 2	Mary	30	150	50
Row 3	Mark	65	170	65

Column 1 Column 2 Column 3 Column 4

Sample Code 2:

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

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

OUT ""

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

Table format (TEST.CSV)

Number of rows	6	
Name of column	Number	String
Row 1	1234	ABCD
Row 2	12,34	AB,CD
Row 3	12 34	AB 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"
```


10.56 REPLACE\$()

Description


Return 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.

Example

<u>Sample Code</u>	<u>Result</u>
<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 ABCDEFG</pre>	

11 Device Reconfiguration Commands

11.1 SET COUNTER

Description

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

Syntax

```
SET COUNTER @n step
@n= "Expression "
```

Parameter	Description
@n	n: counter number. There are 61 counters available (@0 ~ @60) in the printer. @0 to @50 will be cleared while restarting the printer. @51 to @60 will be stored in the printer until the printer is restored to factory default.
step	The increment of the counter can be positive or negative. -999999999 <= step <= 999999999 If the counter is used as a fixed variable, set the increment to 0.
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

11.2 SET CUTTER

Description

Enable/disable the cutter and define how many printed labels is to be cut at one time. This setting will be saved in the 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. $0 \leq \text{pieces} \leq 65535$

Example

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

See Also

OFFSET, PRINT, SET PARTIAL_CUTTER

11.3 SET PARTIAL_CUTTER

Description

Enable/disable the cutter and define how many printed labels is to be cut at one time. This setting will be saved in the 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

11.4 SET BACK

Description

Disable/Enable back feed the labels after a cut. Make sure to use this command after **SET CUTTER** command.

Syntax

SET BACK OFF/ON

<u>Parameter</u>	<u>Description</u>
OFF	Disable back function.
ON	Enable back function.

Example

Sample code

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

See Also

OFFSET, PRINT, SET CUTTER

11.5 SET KEYn

Description

Enable/disable the **KEYn** function. Before setting **KEYn** function, make sure to disable the default function of **KEYn** first. The setting will remain resident in the printer even when the printer is powered off.

Syntax

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

Parameter	Description
n	0, 1, 2, 3, 4, 5, 6
ON	Enable KEYn function
OFF	Disable KEYn function
DEFAULT	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 can manually control the feeding distance by pressing and holding the key.
BACKFEED	Set to BACKFEED key that can manually control the backfeed distance by pressing and holding the key.
FORMFEED	Set to FORMFEED key that will feed the label under the format. (e.g. If format is "size 4,6, it will feed 6")
CUT	Set to CUT key
INPUT "string "	Send the command by PRESS key (e.g. SET KEY1 INPUT "CONFIG " + CHR\$(13) + CHR\$(10))

The default function of KEYn is as listed below:

Model	KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6
TD-4T series		FEED					
RJ series		FEED					
TJ series		FEED	MENU	UP	RIGHT	LEFT	DOWN

Example

Sample code

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

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

See Also

OFFEST, PRINT

11.6 SET LEDn

Description

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>
LED1	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	LED2 & LED3
TD-4T series Note: For this series, the LED1=LED2	GREEN	GREEN	RED					ORANGE
RJ series	GREEN	GREEN	RED	BAT1	BAT2	BAT3	WIFI/BT	ORANGE
TJ series	GREEN	GREEN	RED					ORANGE

Example

Sample code

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


11.7 SET PEEL

Description

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 the 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

11.8 SET REWIND

Description

Enable/disable the internal rewind function. The default setting is OFF. When this function is ON, the printer's Media Rewind Spindle will rewind the printed labels. This setting will be saved in the 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)

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

11.9 SET TEAR

Description

Send the label to the tear position up to the gap/black mark. This setting will be saved in the printer memory when turning off the power.

Syntax

SET TEAR ON/OFF (FBPL language printers only)

<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

11.10 SET GAP

Description

Adjust the gap sensor emission sensitivity. The printer initiates automatic gap sensor calibration when the **PAUSE** key is held down while powering up. Use this setting when auto calibration does not work because the thickness of the backing paper and that of the label with backing paper are not within the sensor's detection range or there are pre-printed marks or patterns on the label. This setting will be saved in the printer memory when turning off the power.

Syntax

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

Parameter	Description
N	Gap sensor light emission strength. Available range is listed as below. 0 is the lowest sensitivity.
AUTO	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, the separation between labels, can't be detected by the black mark sensor. The parts of the media which can be passed through by the 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
TD-4420TN/4520TN	0~15	0~3	V
TD-4650TNWB / TD-4650TNWBR, TD-4750TNWB / TD-4750TNWBR	0~15	0~3	V
RJ-2035B / RJ-2055WB RJ-3035B / RJ-3055WB	0~7	0~7	V
TJ-4020TN / TJ-4021TN / TJ-4021TNR/ TJ-4120TN / TJ-4121TN / TJ-4121TNR	0~15	0~3	V
TJ-4420TN / TJ-4520TN / TJ-4620TN / TJ-4422TN / TJ-4522TN	0~15	0~3	V

Note:

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

Example

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

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

Note:

<ENTER> stands for keyboard **ENTER** key. In the above example, 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 the 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

11.11 SET BLINE

Description

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.

11.12 SET HEAD

Description

Enable/disable the 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 the printer memory.

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

11.13 SET RIBBON

Description

Enable/disable ribbon sensor detection. The 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 the printer memory.

Syntax

SET RIBBON ON/OFF/INSIDE/OUTSIDE

<u>Parameter</u>	<u>Description</u>
ON	Thermal transfer printing
OFF	Direct Thermal Printing

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
```


11.14 SET ENCODER

Description

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

11.15 SET RIBBONEND

Description

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.

Example

SET RIBBONEND ON
SET RIBBONEND OFF

11.16 SET COM1

Description

Define communication parameters for printer serial port.

Syntax

```
SET COM1 baud,parity,data,stop
```

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

Example

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

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

Note:

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

11.17 SET PRINTKEY

Description

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

Syntax

SET PRINTKEY OFF/ON/AUTO/<num>

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

Example

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

Execute:

Syntax	Receive "PRINT m"	Print Out
SET PRINTKEY ON or SET PRINTKEY AUTO	1.) PRINT 2	Label 1~2
	2.) Press FEED key	Label 3~4

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

11.18 SET REPRINT

Description

Enable/disable 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

Example

SET REPRINT ON

11.19 SET FEED_LEN

Description

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	Feeding length (in dots)

Example

Sample code

```
SET FEED_LEN 100
```

Result

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

11.20 GETSENSOR()

Description

Get the sensor status/AD value. You can use it to check the sensor function.

Syntax

```
GETSENSOR(sensor$[,intension])
```

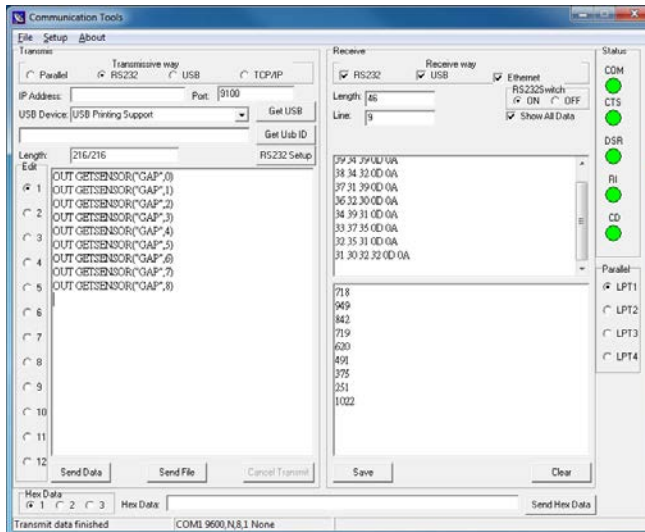
Parameter	Description	
sensor\$	Sensor type	
	GAP	Gap sensor
	BLINE	Black mark sensor
	RIBBON	Ribbon-end sensor
	PEEL	Peeler sensor
	HEAD UP	Thermal print head open sensor
	HEAD TEMP	Temperature of print head
	HEAD VOLT	Voltage of print head
	BATTERY VOLT	Voltage of battery (V)
	BATTERY CAP	Capacity of battery (%)
intension	Sensor intension	
	Gap	Refer to SET GAP for gap sensor range of different model.
	BLINE	Refer to SET GAP for black mark sensor range of different model.
	RIBBON	0 ~ 3
	PEEL sensor	Ignored
	HEAD UP sensor	Ignored
	HEAD TEMP	Ignored
	HEAD VOLT	Ignored
Returned value	Gap	Return the AD value of gap sensor
	BLINE	Return the AD value of black mark sensor
	RIBBON	Return the AD value of ribbon sensor
	PEEL	The return value will be either 0 or 1 0: Paper is not on the sensor 1: Paper is on the sensor
	HEAD UP	The return value will be either 0 or 1 0: Print head module is close 1: Print head module is open
	HEAD TEMP	Return the temperature of thermal print head
	HEAD VOLT	Return the voltage of thermal print head

Example (This example uses the Brother CommTool to get the 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 )
```

Result

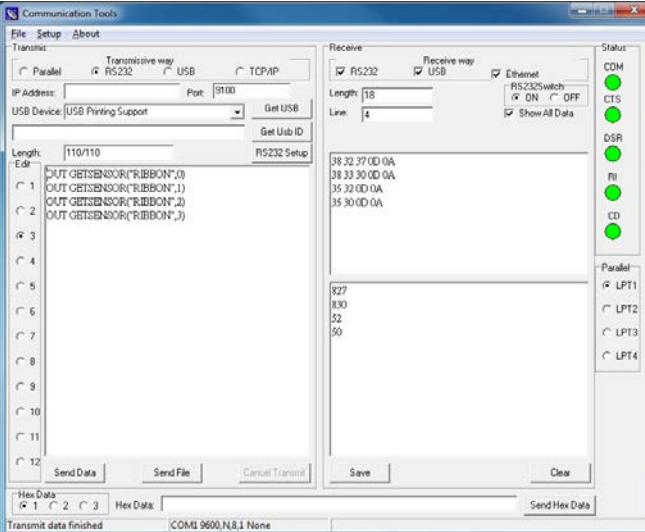


If the returned valued is changed in different sensor intension, the sensor is functional.

Sample code

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

Result

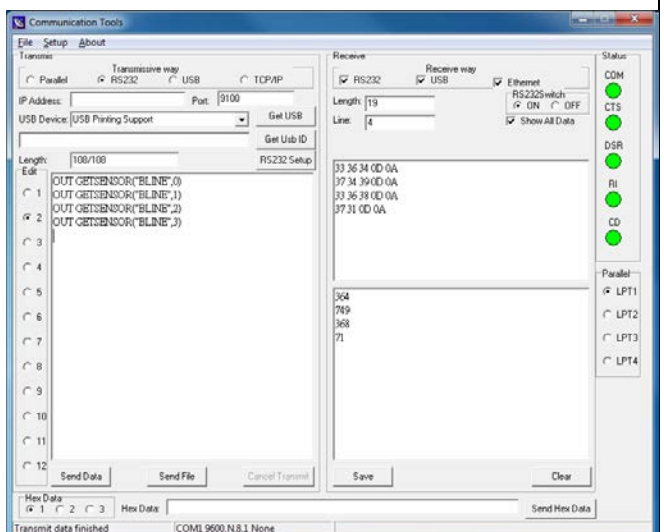


If the returned valued is changed in different sensor

Sample code

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

Result

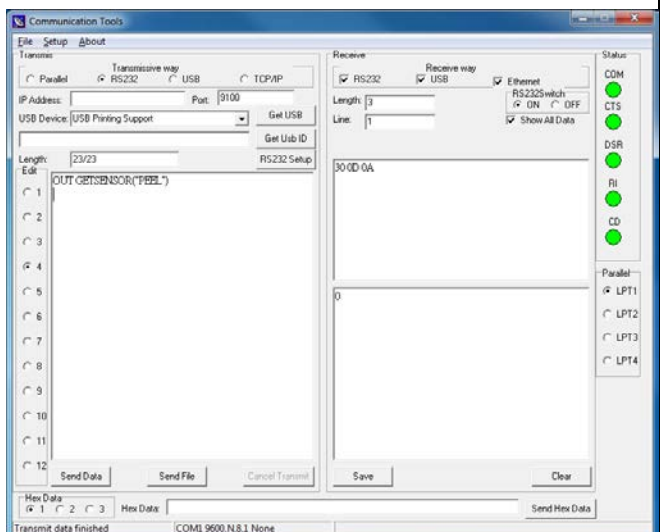


If the returned valued is changed in different sensor intension, the sensor is functional.

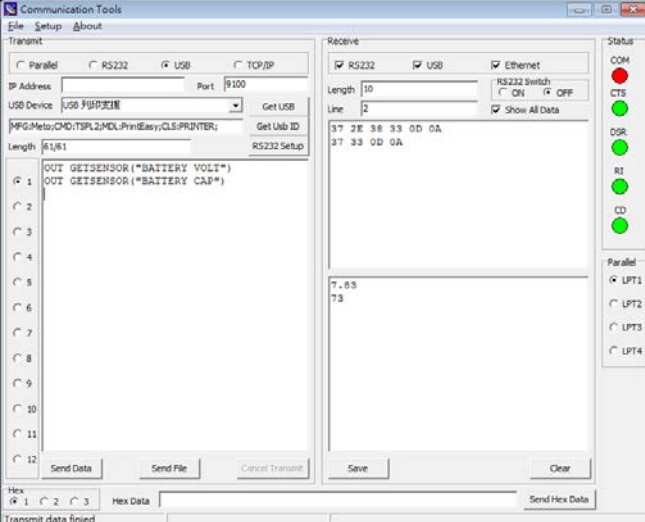
Sample code

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

Result



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

intension, the sensor is functional.	sensor.
<p>Sample code</p> <pre>OUT GETSENSOR("BATTERY VOLT") OUT GETSENSOR("BATTERY CAP")</pre> <p>Result</p>  <p>This code is used to detect the battery volatage and battery capacity for RJ printer.</p>	

11.21 GETSETTING\$()

Description

Obtain the printer configurations.

Syntax

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

app\$	sec\$	key\$	Comment
SYSTEM	INFORMATION	DPI	Resolution
		MODEL	Model name
		SERIAL	Serial number
		VERSION	Firmware version
		CHECKSUM	Firmware checksum
		PRINTQUALITY	Print mode (DRAFT, STANDARD or OPTIMUM; see SET PRINTQUALITY) For RJ-2035B/2055WB/3035B/3055WB only
	RECORD	STANDBYTIME	Printer standby time (OFF or number) For RJ-2035B/2055WB/3035B/3055WB only
		MILAGE	Printed mileage (in dots)
		CUT COUNTER	Cutting counter
FILE	DRAM	CAPACITY	Total capacity of DRAM
		AVAILABLE	Available capacity of DRAM
	FLASH	CAPACITY	Total capacity of FLASH
		AVAILABLE	Available capacity of FLASH
	CARD	CAPACITY	Total capacity of CARD
		INSTALLED	Status of card. 1: installed; 0: none installed.
CONFIG	NET	MAC ADDRESS	MAC address
		IP ADDRESS	IP address
		SUBNET MASK	Subnet Mask
		DEFAULT GATEWAY	Default gateway
		RAW PORT	Raw port
		NAME	Printer name
		PRIMARY DNS	Primary DNS
	WLAN	SECONDARY DNS	Secondary DNS
		MAC ADDRESS	MAC address
		IP ADDRESS	IP address
		SUBNET MASK	Subnet Mask
		DEFAULT GATEWAY	Default gateway
	COM1	RAW PORT	Raw port
		BAUD RATE	Baud rate of COM port
		DATA BIT	Data bit of COM port
		PARITY	Parity of COM port
	SENSOR	STOP BIT	Stop bit of COM port
		SENSOR TYPE	Current sensor type
		CARRIAGE	Status of head open sensor
		GAP INTENSION	Intension of gap sensor
	FBPL	BLINE INTENSION	Intension of black mark sensor
		CONTINUOUS INTENSION	Intension of continuous sensor
		PRINT MODE	Post-print action
		DENSITY	Print density
		PAPER SIZE	Paper size
		GAP SIZE	Gap size
		BLINE SIZE	Black mark size
		DIRECTION	Printing direction
		MIRROR	Mirror status
		RIBBON	Ribbon status
		REPRINT	Reprint status
		PAPER WIDTH	Paper width
		LIMIT FEED	Maximum length for sensor calibration.

		OFFSET	OFFSET value.
		REFERENCE X	REFERENCE X value.
		REFERENCE Y	REFERENCE Y value.
		SHIFT X	SHIFT X value.
		SHIFT Y	SHIFT Y value.
		SPEED	Print speed.
		COUNTRY CODE	COUNTRY code.
		CODEPAGE	CODEPAGE.
<u>Parameter</u> default\$		<u>Description</u> 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 ("").	

Example

Sample code (This example uses the Brother CommTool to get the printer configurations via RS-232.)

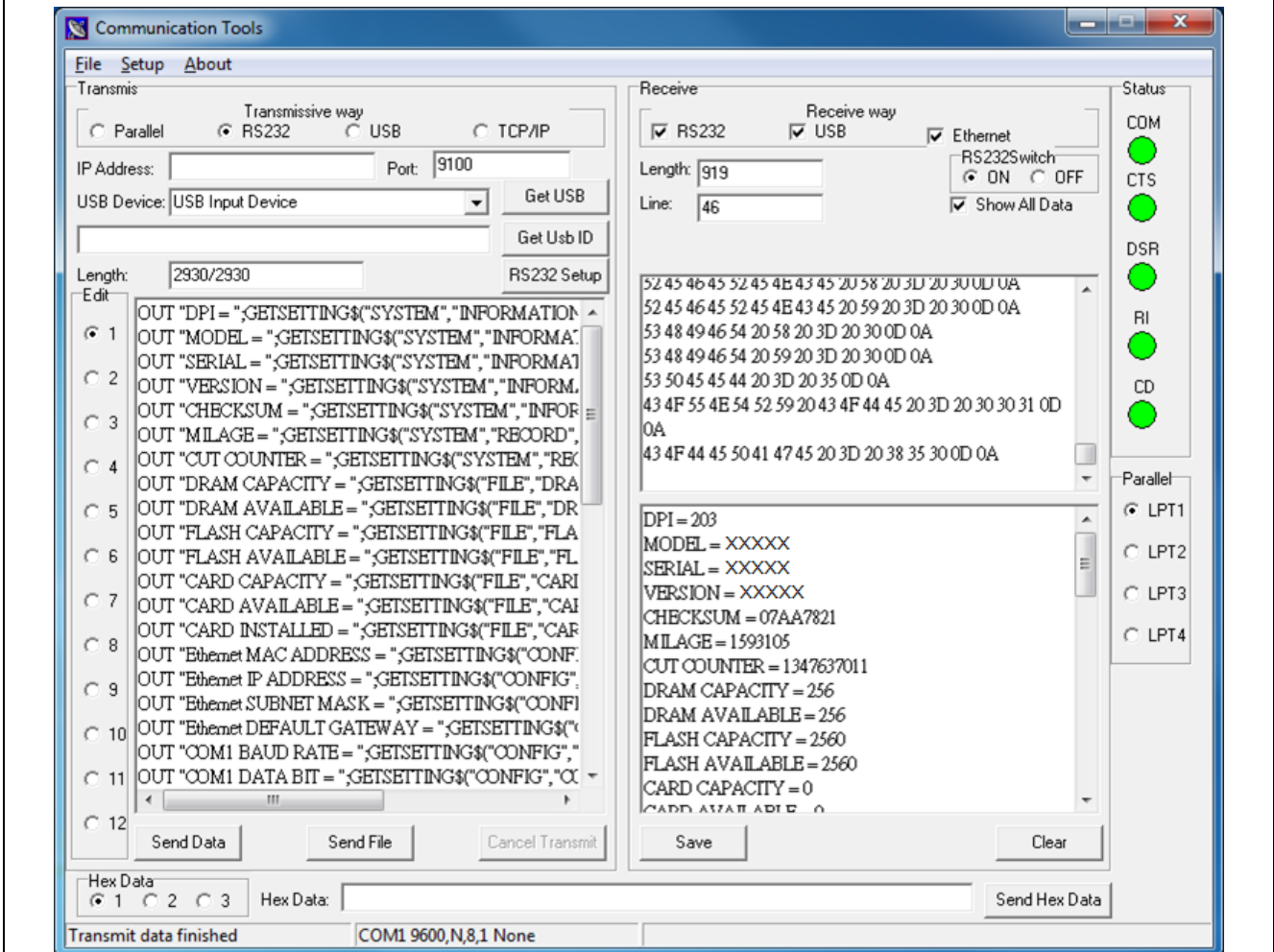
```

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", "FBPL", "PRINT MODE")
OUT "DENSITY = ";GETSETTING$("CONFIG", "FBPL", "DENSITY")
OUT "PAPER SIZE = ";GETSETTING$("CONFIG", "FBPL", "PAPER SIZE")
OUT "GAP SIZE = ";GETSETTING$("CONFIG", "FBPL", "GAP SIZE")
OUT "BLINE SIZE = ";GETSETTING$("CONFIG", "FBPL", "BLINE SIZE")
OUT "DIRECTION = ";GETSETTING$("CONFIG", "FBPL", "DIRECTION")
OUT "MIRROR = ";GETSETTING$("CONFIG", "FBPL", "MIRROR")
OUT "RIBBON = ";GETSETTING$("CONFIG", "FBPL", "RIBBON")
OUT "REPRINT = ";GETSETTING$("CONFIG", "FBPL", "REPRINT")
OUT "PAPER WIDTH = ";GETSETTING$("CONFIG", "FBPL", "PAPER WIDTH")
OUT "LIMIT FEED = ";GETSETTING$("CONFIG", "FBPL", "LIMIT FEED")
OUT "OFFSET = ";GETSETTING$("CONFIG", "FBPL", "OFFSET")
OUT "REFERENCE X = ";GETSETTING$("CONFIG", "FBPL", "REFERENCE X")
OUT "REFERENCE Y = ";GETSETTING$("CONFIG", "FBPL", "REFERENCE Y")
OUT "SHIFT X = ";GETSETTING$("CONFIG", "FBPL", "SHIFT X")

```

```
OUT "SHIFT Y = ";GETSETTING$("CONFIG","FBPL","SHIFT Y")
OUT "SPEED = ";GETSETTING$("CONFIG","FBPL","SPEED")
OUT "COUNTRY CODE = ";GETSETTING$("CONFIG","FBPL","COUNTRY CODE")
OUT "CODEPAGE = ";GETSETTING$("CONFIG","FBPL","CODEPAGE")
```

Result



Sample code(NET, WLAN)

```
OUT "Ethernet DEFAULT RAW PORT = ";GETSETTINGS$ ("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")
```

11.22 SET USBHOST

Description

Set the USB host for the usage of PC USB keyboard and USB HID interface 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: <ul style="list-style-type: none">▪ 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
```

11.23 SET AUTORUN

Description

Redefine 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.

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.

11.24 SET RESPONSE

Description

Respond to an error 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	Respond at the end of the print job.

Response Syntax

```
{Status,#####,ID}
```

Status

[Hex Receive]

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

#####: 00001 ~ 99999

Example

Sample Code

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

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

Sample Code

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

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

Sample Code

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

```
{00,00006,CCCC}
```

11.25 SET DAYLIGHT_SAVE

Description

Set daylight saving time.

Syntax

```
SET DAYLIGHT_SAVE ON/OFF
```

```
SET DAYLIGHT_SAVE "Start", "End"
```

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

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"
```

11.26 SET REGISTRATION

Description

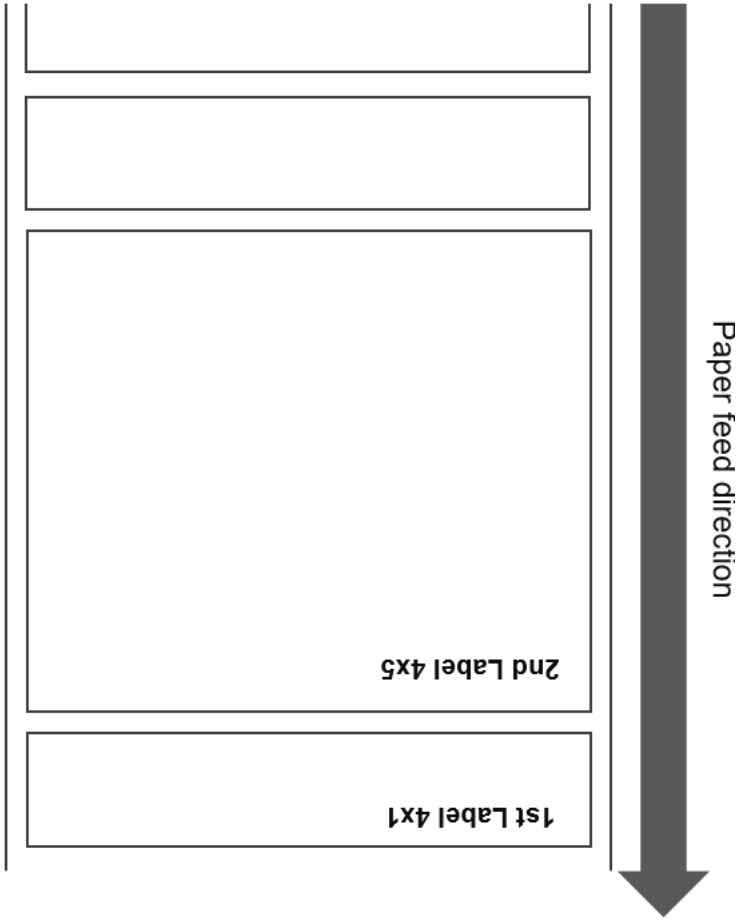
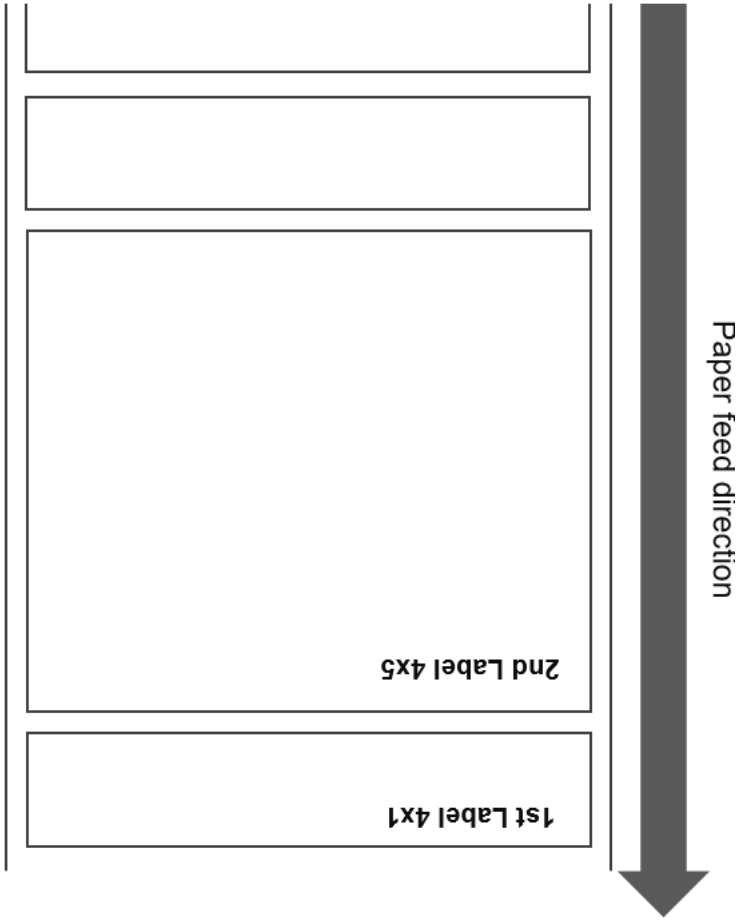
Set the label position mode for the label roll with a different size of labels.

Syntax

SET REGISTRATION mode

Parameter	Description
mode	BYSIZE: Default ACTUAL: For the label roll with different sizes of labels on the same roll

Example

<p>Sample Code</p> <p>2 jobs on 2 labels:</p> <pre>SET REGISTRATION ACTUAL SIZE 4,1 GAP 0.12,0 CLS TEXT 40,40,"0",0,10,10,"1st Label 4x1" PRINT 1 SIZE 4,5 GAP 0.12,0 CLS TEXT 40,40,"0",0,10,10,"2nd Label 4x5" PRINT 1</pre>	 <p>The diagram illustrates the paper feed direction, indicated by a large downward arrow on the right labeled "Paper feed direction". It shows two labels being processed. The first label is labeled "1st Label 4x1" and the second label is labeled "2nd Label 4x5". The labels are positioned on a roll of paper, with the first label being smaller than the second label.</p>
<p>1 job on 2 labels:</p> <pre>SET REGISTRATION ACTUAL SIZE 4,6.12 GAP 0.12,0 CLS TEXT 40,40,"0",0,10,10,"1st Label 4x1" TEXT 40,267,"0",0,10,10,"2nd Label 4x5" PRINT 1</pre>	 <p>The diagram illustrates the paper feed direction, indicated by a large downward arrow on the right labeled "Paper feed direction". It shows two labels being processed. The first label is labeled "1st Label 4x1" and the second label is labeled "2nd Label 4x5". The labels are positioned on a roll of paper, with the first label being smaller than the second label.</p>

11.27 PEEL

Description

Obtain 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
```

11.28 LED1, LED2, LED3

Description

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, make sure to cancel the default LED functions. See 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

Model	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED2 & LED3
TD-4T series Note: For this series, the LED1=LED2	GREEN	GREEN	RED					ORANGE
RJ series	GREEN	GREEN	RED	BAT1	BAT2	BAT3	WIFI/BT	ORANGE
TJ series	GREEN	GREEN	RED					ORANGE

Example

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

11.29 KEY1, KEY2, KEY3

Description

Read the status of KEY1, KEY2 and KEY3.

Model	KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6
TD-4T series		FEED					
RJ series		FEED					
TJ series		FEED	MENU	UP	RIGHT	LEFT	DOWN

Syntax

KEYm = n

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

Example

Sample code

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

11.30 SET SENSOR_REF

Description

Set the threshold detection of the sensor.

Syntax

```
SET SENSOR_REF AUTO
SET SENSOR_REF MANUAL
```

<u>Key</u>	<u>Return Value</u>
AUTO	When feeding the media, the media positioning threshold is automatically fine-tuned according to the sensor values (high/low peak); Default
MANUAL	When feeding the media, the media positioning threshold is NOT automatically fine-tuned according to the sensor values (high/low peak), the paper positioning threshold is fixed.

Example

Sample code

```
SET SENSOR_REF AUTO

SET SENSOR_REF MANUAL
```

12 Printer Global Variables

12.1 @LABEL

Description

Count 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 is turned off.

Syntax

Write attribute: @LABEL=n or @LABEL= "n"

Read attribute: A=LABEL or A\$=STR\$(LABEL)

Parameter	Description
n	Number of labels printed. $0 \leq n \leq 999999999$

Example

Sample code	Result
<pre>DOWNLOAD "DEMO.BAS" SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS TEXT 10,50, "3",0,1,1,@LABEL TEXT 10,100, "3",0,1,1, "@LABEL="+STR\$(LABEL) TEXT 10,150, "3",0,1,1, "*****Statement 1*****" IF LABEL>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</pre>	<pre>1661 @LABEL=1661 *****Statement 1***** LABEL>1000 *****Statement 1***** A>1000 *****Statement 3***** VAL(A\$)>1000</pre>

12.2 YEAR

Description

Write/read the year data via the Real Time Clock (RTC). Four-digit year format is supported by RTC.

Syntax

Write attribute: YEAR = 02

Read attribute: A = YEAR

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

Example

Sample code

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

Sample code

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

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

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

Result

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

See Also

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

12.3 MONTH

Description

Write/read the month data via the Real Time Clock (RTC). Two-digit (01~12) month format is supported by RTC.

Syntax

Write attribute: MONTH = 01

Read attribute: A = MONTH

Range: 01~12

Example

Sample code

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

Sample code

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

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

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

Result

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

See Also

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

12.4 DATE

Description

Write/read the date data via the Real Time Clock (RTC). Two-digit (01~31) date format is supported by RTC.

Syntax

Write attribute: DATE = 12

Read attribute: A = DATE

Range: 01~31

Example

Sample code

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

Sample code

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

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

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

Result

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

See Also

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

12.5 WEEK

Description

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

Syntax

Read attribute: A = WEEK

Range: 1(Sunday)~7(Saturday)

Example

Sample code

```
DOWNLOAD "DEMO.BAS "  
SIZE 4,1  
GAP 0,0  
DIRECTION 1  
CLS  
  
REM *****Read Week parameter form RTC*****  
WEEK$=STR$(WEEK)  
W=WEEK  
  
REM *****Print*****  
TEXT 10,10, "3",0,1,1, "WEEK1="+WEEK$  
TEXT 10,50, "3",0,1,1, "WEEK2="+STR$(W)  
TEXT 10,90, "3",0,1,1, "WEEK3="+STR$(WEEK)  
PRINT 1  
EOP  
DEMO
```

Result

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

See Also

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

12.6 HOUR

Description

Write/read the hour data via the Real Time Clock (RTC). The 24-hour-day system (00~23) is supported by RTC.

Syntax

Write attribute: HOUR = 12

Read attribute: A = HOUR

Range: 00~23

Example

Sample code

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

Sample code

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

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

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

Result

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

See Also

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

12.7 MINUTE

Description

Write/read the minute data via the Real Time Clock (RTC). Two-digit (00~59) minute format is supported by RTC.

Syntax

Write attribute: MINUTE = 12

Read attribute: A = MINUTE

Range: 00~59

Example

Sample code

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

Sample code

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

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

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

Result

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

See Also

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

12.8 SECOND

Description

Write/read the second data via the Real Time Clock (RTC). Two-digit (00~59) second format is supported by RTC.

Syntax

Write attribute: SECOND = 12

Read attribute: A = SECOND

Range: 00~59

Example

Sample code

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

Sample code

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

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

REM *****Print*****
TEXT 10,10, "3",0,1,1, "SECOND1="+SECOND$
TEXT 10,50, "3",0,1,1, "SECOND2="+STR$(SEC)
TEXT 10,90, "3",0,1,1, "SECOND3="+STR$(SECOND)
PRINT 1
EOP
DEMO
```

Result

```
SECOND1=59
SECOND2=59
SECOND3=59
```

See Also

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

12.9 @YEAR

Description

Write/read the year data via the Real Time Clock (RTC). Two-digit year format is supported by RTC. @YEAR global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @YEAR = "01"

Read attribute: @YEAR

Range: 00~99

Example

Sample code	Result
<pre>REM *****Set @YEAR***** @YEAR="05" REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@YEAR" TEXT 210,10, "3",0,1,1, @YEAR PRINT 1</pre>	<div>@YEAR2005</div>

See Also

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

12.10 @MONTH

Description

Write/read the month data via the Real Time Clock (RTC). Two-digit (01~12) month format is supported by RTC. **@MONTH** global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @MONTH = "01"

Read attribute: @MONTH

Range: 01~12

Example

Sample code	Result
<pre>REM *****Set @MONTH***** @MONTH="12" REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@MONTH" TEXT 210,10, "3",0,1,1,@MONTH PRINT 1</pre>	<div>@MONTH12</div>

See Also

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

12.11 @DATE

Description

Write/read the date data via the Real Time Clock (RTC). Two-digit (01~31) date format is supported by RTC.
@DATE global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @DATE = "12"

Read attribute: @DATE

Range: 01~31

Example

Sample code	Result
<pre>REM *****Set @DATE***** @DATE="31" REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@DATE" TEXT 210,10, "3",0,1,1,@DATE PRINT 1</pre>	<div>@DATE31</div>

See Also

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

12.12 @DAY

Description

Write/read the day of the week data via the Real Time Clock (RTC), which is represented by one single digit (1~7). **@DAY** global variable can be accessed directly without using BASIC language functions.

Syntax

Read attribute: @DAY

Range: 1(Sunday)~7(Saturday)

Example

Sample code	Result
<pre>REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@DAY" TEXT 210,10, "3",0,1,1,@DAY PRINT 1</pre>	<div>@DAY7</div>

See Also

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

12.13 @HOUR

Description

Write/read the hour data via the Real Time Clock (RTC). The 24-hour-day system (00~23) is supported by RTC. **@HOUR** global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @HOUR = "12"

Read attribute: @HOUR

Range: 00~23

Example

Sample code	Result
<pre>REM *****Set @HOUR***** @HOUR="23" REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@HOUR" TEXT 210,10, "3",0,1,1,@HOUR PRINT 1</pre>	<div>@HOUR23</div>

See Also

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

12.14 @MINUTE

Description

Write/read the minute data via the Real Time Clock (RTC). The two-digit (00~59) minute format is supported by RTC. **@MINUTE** global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @MINUTE = "12"

Read attribute: @MINUTE

Range: 00~59

Example

Sample code	Result
<pre>REM *****Set @MINUTE***** @MINUTE="59" REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@MINUTE" TEXT 210,10, "3",0,1,1,@MINUTE PRINT 1</pre>	<p>@MINUTE 59</p>

See Also

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

12.15 @SECOND

Description

Write/read the second data via the Real Time Clock (RTC). The two-digit (00~59) second format is supported by RTC. **@SECOND** global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @SECOND = "12"

Read attribute: @SECOND

Range: 00~59

Example

Sample code	Result
<pre>REM *****Set @SECOND***** @SECOND = "59" REM *****Print***** SIZE 4,1 GAP 0,0 DIRECTION 1 CLS TEXT 10,10, "3",0,1,1, "@SECOND" TEXT 210,10, "3",0,1,1,@SECOND PRINT 1</pre>	<p>@SECOND 59</p>

See Also

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

12.16 _MODEL\$

Description

This variable can be read only. It includes the information of printer's model name.

Syntax

_MODEL\$

Example

Sample code

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

Result

```
Model:*****
Serial No.:*****
F/W Version:*.**
```

See Also

_SERIAL\$, _VERSION\$

12.17 _SERIAL\$

Description

This variable can be read only. It includes the information of printer's serial number.

The printer's serial number must be programmed into printer at factory.

Syntax

_SERIAL\$

Example

Sample code

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

Result

```
Model:*****
Serial No.:*****
F/W Version:*.**
```

See Also

_MODEL\$, _VERSION\$

12.18 _VERSION\$

Description

This variable can be read only. It includes the information of the printer's firmware version.

Syntax

_VERSION\$

Example

Sample code

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

Result

```
Model:*****
Serial No.:*****
F/W Version:*.**
```

See Also

_MODEL\$, _SERIAL\$

13 Bluetooth Module Setting Commands

13.1 BT NAME

Description

Set the 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 "BROTHER01" SELFTEST BT</pre>	<pre>----- BT SETTING ----- MAC ADDR: XXXXXX111111 NAME: BROTHER01 PIN CODE: 0000 PRINTER NAME: PAIR MODE: LEGACY MODULE: XXXX XXX MFI SUPPORTED: YES -----</pre>

13.2 BT PINCODE

Description

Set the Bluetooth module PIN code (Max.15 byte).

Syntax

```
BT PINCODE "pincode"
```

Note:

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

Example

Sample code	Result
<pre>BT PINCODE "1234" SELFTEST BT</pre>	<pre>----- BT SETTING ----- MAC ADDR: XXXXXX111111 NAME: BROTHER01 PIN CODE: 1234 PRINTER NAME: PAIR MODE: LEGACY MODULE: XXXX XXX MFI SUPPORTED: YES -----</pre>

13.3 BT MODE

Description

Enable or disable the Bluetooth Low Energy (BLE) mode.

Syntax

- **Enable Bluetooth Low Energy (BLE)**

[TD-4650TNWB/TD-4750TNWB/TD-4650TNWBR/TD-4750TNWBR/RJ-2055WB/RJ-3055WB/TJ-4420TN/TJ-4520TN/TJ-4620TN/TJ-4422TN/TJ-4522TN]

Send the following commands.

```
BT MODE "BT4.0"  
WLAN MODULE SAVECFG  
DELAY 20000  
INITIALPRINTER
```

Wait for about 15 seconds for the printer to restart automatically.

[RJ-2035B/RJ-3035B]

Send the command below.

```
BT MODE BT4.0  
SET BTLINKBACK OFF
```

When BLE is enabled, the automatic Bluetooth reconnection between your iOS/iPadOS device and the printer will be disabled.

- **Disable Bluetooth Low Energy (BLE)**

[TD-4650TNWB/TD-4750TNWB/TD-4650TNWBR/TD-4750TNWBR/RJ-2055WB/RJ-3055WB/TJ-4420TN/TJ-4520TN/TJ-4620TN/TJ-4422TN/TJ-4522TN]

Send the following commands.

```
BT MODE "BT2.1"  
WLAN MODULE SAVECFG  
DELAY 20000  
INITIALPRINTER
```

Wait for about 15 seconds for the printer to restart automatically.

[RJ-2035B/RJ-3035B]

To enable the automatic Bluetooth reconnection between your iOS/iPadOS device and the printer:

```
BT MODE BT2.1  
SET BTLINKBACK ON
```

Wait for about 15 seconds for the printer to restart automatically.

To disable the automatic Bluetooth reconnection between your iOS/iPadOS device and the printer:

BT MODE BT2.1

Restart the printer.

Note:

[TD-4650TNWB/TD-4750TNWB/TD-4650TNWBR/TD-4750TNWBR/RJ-2055WB/RJ-3055WB]

- Make sure you use the correct firmware version to enable Bluetooth Low Energy (BLE).

Model	Firmware Version
TD-4650TNWB / 4750TNWB / 4650TNWBR / 4750TNWBR	V1.04.S18 or later
RJ-2055WB / 3055WB	B1.00.Q28 or later

- Check whether the Bluetooth/Wi-Fi modules current firmware supports BLE. For more information, see FAQ: *Enable or disable Bluetooth Low Energy (BLE)* on your model's **FAQs & Troubleshooting** page at support.brother.com.

[RJ-2035B/RJ-3035B]

- When BLE is enabled, the automatic Bluetooth reconnection between your iOS/iPadOS device and the printer will be disabled.
- Update the firmware to B1.00.Q33 or later to use the automatic Bluetooth reconnection feature.

13.4 SET BTLINKBACK

Description

Enable or disable the automatic Bluetooth reconnection feature (MFi model only).

Syntax

- **Enable the automatic Bluetooth reconnection feature (Default)**

```
SET BTLINKBACK ON
```

- **Disable the automatic Bluetooth reconnection feature**

Disable the automatic Bluetooth reconnection feature to prevent any previously paired iOS/iPadOS devices from automatically connecting to the printer:

```
SET BTLINKBACK OFF
```

Note:

- If the printer firmware version is B1.00.Q33 or later, the printer can use the automatic Bluetooth reconnection feature.
- When BLE is enabled, the automatic Bluetooth reconnection between your iOS/iPadOS device and the printer will be disabled (For more information, see **BT MODE** command).

14 Wi-Fi Module Setting Commands

14.1 WLAN OFF

Description

Disable the Wi-Fi module. Make sure to restart the printer for the command to be executed.

Syntax

WLAN OFF

Note:

You can use command **SELFTEST WLAN** to check the status of WLAN. The SSID is empty.

Example

Sample code	Result
WLAN OFF SELFTEST WLAN	<pre>----- WLAN SETTING ----- MAC ADDR: XXXX-XXXX MODE: INFRASTRUCTURE SSID: DHCP: OFF IP ADDR: 0.0.0.0 SUBNET: 0.0.0.0 GATEWAY: 0.0.0.0 PORT: 9100 -----</pre>

See Also

WLAN SSID

14.2 WLAN SSID

Description

Set the SSID of your wireless network into Wi-Fi module. Make sure to restart the printer for the command to be executed.

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

<u>Sample code</u>	<u>Result</u>
<pre>WLAN SSID "TEST-AP" SELFTEST WLAN</pre>	<pre>----- WLAN SETTING ----- MAC ADDR: XXXX-XXXX 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

14.3 WLAN WPA

Description

Disable the WPA security mode or set the password (Network key). Make sure to restart the printer for the command to be executed.

Syntax

```
WLAN WPA OFF
WLAN WPA "key"
```

<u>Parameter</u>	<u>Description</u>
OFF	Disable WPA security mode.
Key	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"
```

14.4 WLAN WEP

Description

Disable the WEP security mode or set the password (Encryption key). Make sure to restart the printer for the command to be executed.

Syntax

```
WLAN WEP OFF
```

```
WLAN WEP n, "key"
```

<u>Parameter</u>	<u>Description</u>
OFF	Disable WEP security mode.
N	Index of key. 1 to 4.
Key	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"
```

14.5 WLAN DHCP

Description

Set the printer to get the IP address from DHCP server. Make sure to restart the printer for the command to be executed.

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: XXXX-XXXX 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 IP

14.6 WLAN IP

Description

Set the specific static IP address to printer. Make sure to restart the printer for the command to be executed.

Syntax

```
WLAN IP "ip", "mask", "gateway"
```

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

Example

<u>Sample code</u>	<u>Result</u>
<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: XXXX-XXXXX 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

14.7 WLAN PORT

Description

Specify the PORT number of Wi-Fi module. Make sure to restart the printer for the command to be executed.

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: XXXX-XXXXX SSID: TEST-AP DHCP: OFF IP ADDR: 10.0.10.138 SUBNET: 255.255.255.0 GATEWAY: 10.0.10.252 PORT: 9100 -----</pre>

15 Internal Ethernet Setting Commands

15.1 NET DHCP

Description

Set the printer to get the IP address from DHCP server. The printer will restart itself while setting this command.

Syntax

NET DHCP

Example

Sample code	Result
NET DHCP SELFTEST ETHERNET	----- ETHERNET SETTING ----- NAME: XXXXXX MAC ADDR: XXXXXX DHCP: ON IP ADDR: 192.168.0.107 SUBNET: 255.255.255.0 GATEWAY: 192.168.0.1 PORT: 9100 -----

See Also

NET IP

15.2 NET IP

Description

Set the specific IP address to the printer. The 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

```
NET IP "192.168.10.40", "255.255.255.0", "192.168.10.252"  
SELFTEST ETHERNET
```

Result

```
-----  
      ETHERNET SETTING  
-----  
      NAME : XXXXXX  
MAC ADDR : XXXXXX  
      DHCP : OFF  
      IP ADDR : 192.168.10.40  
      SUBNET : 255.255.255.0  
      GATEWAY : 192.168.10.252  
      PORT : 9100  
-----
```

See Also

NET DHCP

15.3 NET PORT

Description

Specify the PORT number of Ethernet. The 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 : XXXXXX MAC ADDR : XXXXXX DHCP : OFF IP ADDR : 192.168.10.40 SUBNET : 255.255.255.0 GATEWAY : 192.168.10.252 PORT : 9100 -----</pre>

15.4 NET NAME

Description

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

```
NET NAME "TEST"  
SELFTEST ETHERNET
```

Result

```
-----  
                ETHERNET SETTING  
-----  
                NAME: TEST  
MAC ADDR: XXXXXX  
DHCP: OFF  
IP ADDR: 192.168.10.40  
SUBNET: 255.255.255.0  
GATEWAY: 192.168.10.252  
PORT: 9100  
-----
```

16 Setting Commands for RJ-2035B/2055WB/3035B/3055WB

16.1 SET PRINTQUALITY

Description

Set the print mode (print quality) for RJ-2035B/2055WB/3035B/3055WB.

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

```
SET PRINTQUALITY DRAFT
SET PRINTQUALITY STANDARD
SET PRINTQUALITY OPTIMUM
```

16.2 SET STANDBYTIME

Description

Set the standby time for RJ-2035B/2055WB/3035B/3055WB.

Syntax

```
SET STANDBYTIME OFF/XXXXX
```

<u>Parameter</u>	<u>Description</u>
OFF	Disable
XXXXX	0 ~ 625534 (seconds)

Note:

The default value is 120.

Example

Sample code

```
SET STANDBYTIME OFF  
SET STANDBYTIME 480
```

16.3 SET SLEEPTIME

Description

Set the sleeping time for RJ-2035B/2055WB/3035B/3055WB.

Syntax

```
SET SLEEPTIME OFF/XXXXX
```

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

Note:

The default value is 120.

Example

Sample code

```
SET SLEEPTIME OFF  
SET SLEEPTIME 20
```

17 RFID

17.1 RFID READ / RFID WRITE

Description

Read/Write to an RFID tag.

IMPORTANT:

- Always test RFID media on your printer before purchasing large quantities of media.
- Perform Media Calibration before RFID Calibration. Make sure you correctly install the ribbon (Thermal transfer only) and the label roll.
- For more information, see *User's Guide* on your model's **Manual** page at support.brother.com.

Syntax

RFID READ,A,B,C,D,E,"Read Data:"

RFID WRITE,A,B,C,D,E,data

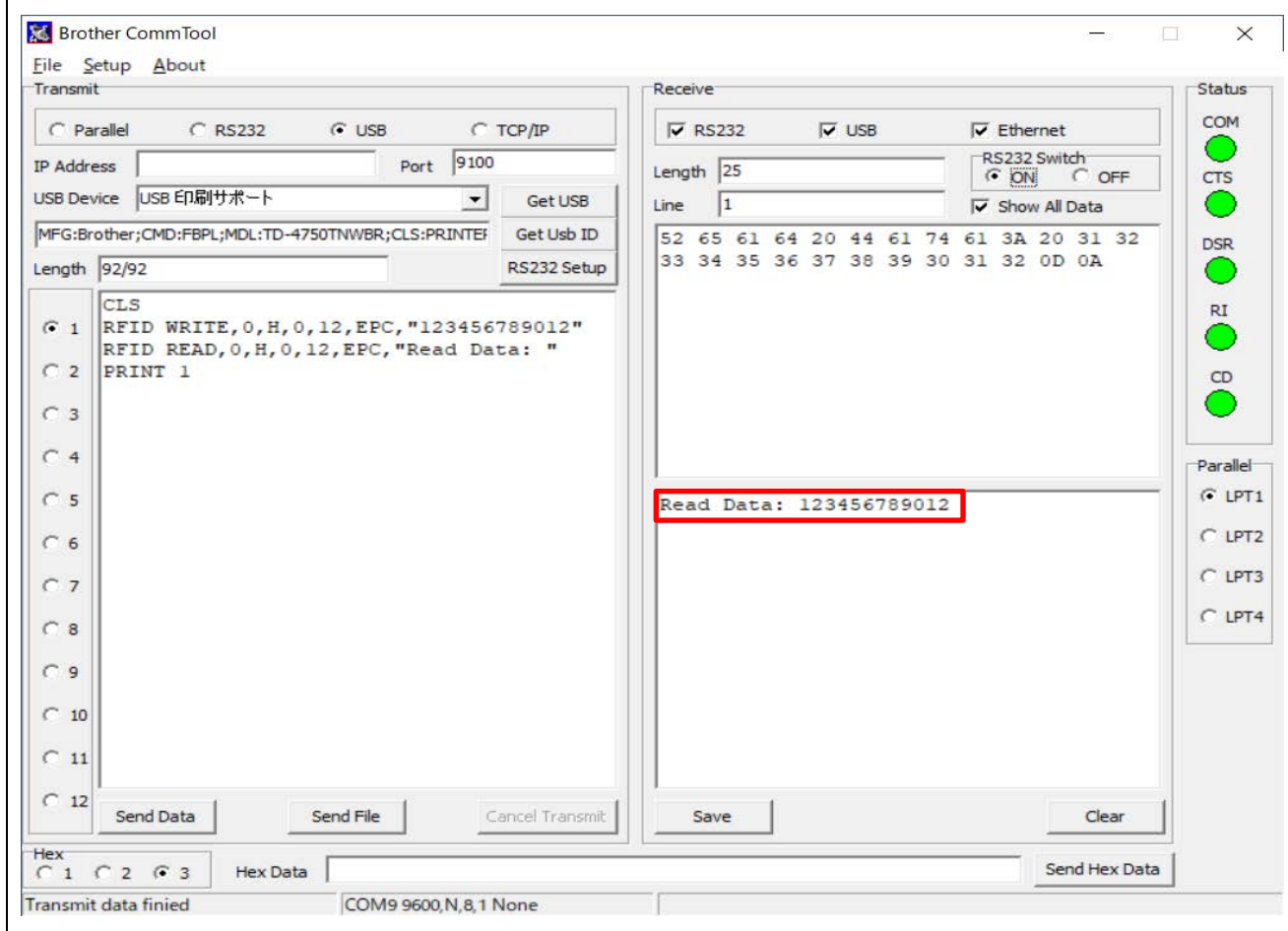
Parameter	Description		
RFID READ RFID WRITE	READ = read the tag WRITE = write to the tag		
A	READ only	unlock password	0 = read without unlock. 1 to FFFFFFFF in hex = read and unlock the data block so it can be overwritten later.
	WRITE only	lock password	0 = write without lock. 1 to FFFFFFFF in hex = write and lock the data block to prevent it from being overwritten.
B	A letter specifying the representation format of the field data. A = ASCII H = Hex		
C	Specifies 0.		
D Size	Read /Write data size from 1 to n in decimal number. Note: <ul style="list-style-type: none">- 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.- Refer to the Sample code (3).		
E Memory bank	EPC - EPC area (Up to 496 bits) TID - Tag Identification area (RFID READ only) USR - User area (Up to 8k bits) ACS - Access code area KIL - Kill code area PC - Protocol Control (PC) area		

<p>"Read Data:" or data</p>	<p>READ = [prompt of data] WRITE = content of data string</p> <p>Note:</p> <ul style="list-style-type: none"> - RFID WRITE supported "string" or basic variable (e.g. VAR\$) - [] = Optional parameter
-------------------------------------	--

Example

Sample code (1)

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



Sample code (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
```

Note:

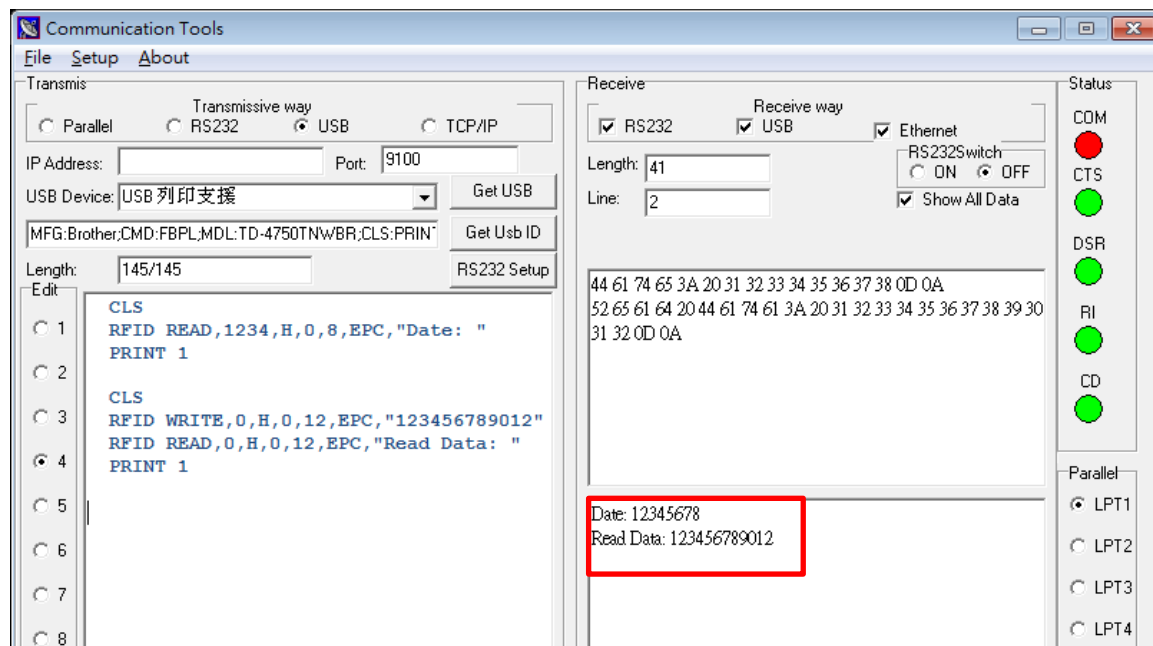
For this locked RFID tag, it cannot be overwritten data without using **RFID READ** unlock password command. If you re-send the **RFID WRITE** command, the printer LCD will be shown as below,



To overwrite this locked tag, 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
```

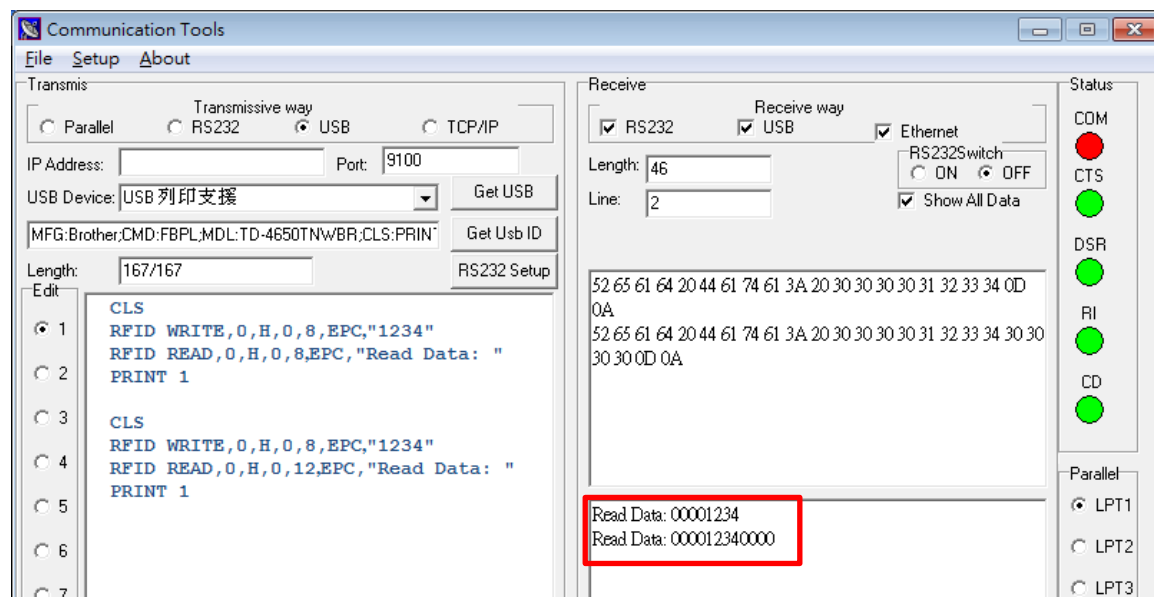


Sample code (3)

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
```



Sample code (4)

This programming example changes PC (Protocol Control) bits.

Note:

Make sure you use the correct firmware version to set the PC bits.

Model	Firmware Version
TD-4650TNWBR / TD-4750TNWBR	V1.04.S21 or later
TJ-4021TNR / TJ-4121TNR	B2.12.S27 or later

CLS

RFID WRITE,0,H,0,4,PC,"15A3"

RFID WRITE,0,H,0,8,EPC,"12345678"

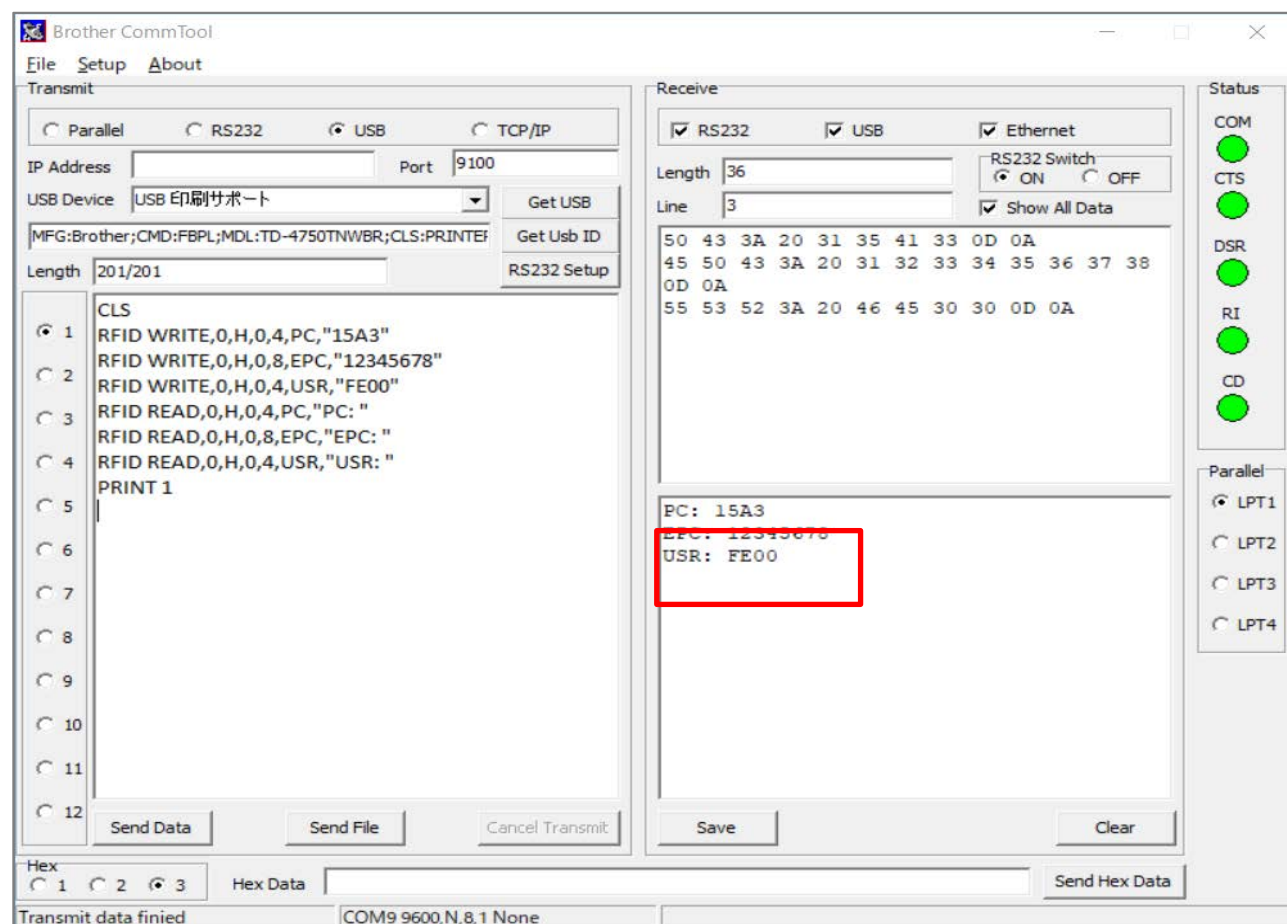
RFID WRITE,0,H,0,4,USR,"FE00"

RFID READ,0,H,0,4,PC,"PC: "

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

RFID READ,0,H,0,4,USR,"USR: "

PRINT 1



17.2 RFIDDETECT

Description

Perform RFID calibration.

Appendix A: Command List by Model

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
Setup and System Commands								
SIZE	○	○	○	○	○	○	○	○
GAP	○	○	○	○	○	○	○	○
GAPDETECT	○	○	○	○	○	○	○	○
BLINEDETECT	○	○	○	○	○	○	○	○
AUTODETECT	○	○	○	○	○	○	○	○
BLINE	○	○	○	○	○	○	○	○
OFFSET	○	○	○	○	○	○	○	○
SPEED	○	○	○	○	○	○	○	○
DENSITY	○	○	○	○	○	○	○	○
DIRECTION and Mirror Image	○	○	○	○	○	○	○	○
REFERENCE	○	○	○	○	○	○	○	○
SHIFT	○	○	○	○	○	○	○	○
COUNTRY	○	○	○			○	○	○
CODEPAGE	○	○	○	○	○	○	○	○
CLS	○	○	○	○	○	○	○	○
FEED	○	○	○	○	○	○	○	○
BACKFEED	○	○	○	○	○	○	○	○
FORMFEED	○	○	○	○	○	○	○	○
HOME	○	○	○	○	○	○	○	○
PRINT	○	○	○	○	○	○	○	○
SOUND	○	○	○	○	○	○	○	○

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
CUT	○	○	○ Except RFID tags			○	○ Except RFID tags	○
LIMITFEED	○	○	○	○	○	○	○	○
SELFTEST	○	○	○	○	○	○	○	○
EOJ	○	○	○	○	○	○	○	○
DELAY	○	○	○	○	○	○	○	○
DISPLAY		○	○			○	○	○
INITIALPRINTER	○	○	○	○	○	○	○	○
MENU		○	○			○	○	○
Label Formatting Commands								
BAR	○	○	○	○	○	○	○	○
BARCODE	○	○	○	○	○	○	○	○
TLC39	○	○	○	○	○	○	○	○
BITMAP	○	○	○	○	○	○	○	○
BOX	○	○	○	○	○	○	○	○
CIRCLE	○	○	○	○	○	○	○	○
ELLIPSE	○	○	○	○	○	○	○	○
CODABLOCK F mode	○	○	○	○	○	○	○	○
DMATRIX	○	○	○	○	○	○	○	○
ERASE	○	○	○	○	○	○	○	○
MAXICODE	○	○	○	○	○	○	○	○
PDF417	○	○	○	○	○	○	○	○
AZTEC	○	○	○	○	○	○	○	○
MPDF417	○	○	○	○	○	○	○	○
PUTBMP	○	○	○	○	○	○	○	○

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
PUTPCX	○	○	○	○	○	○	○	○
QRCODE	○	○	○	○	○	○	○	○
RSS	○	○	○	○	○	○	○	○
REVERSE	○	○	○	○	○	○	○	○
DIAGONAL	○	○	○	○	○	○	○	○
TEXT	○	○	○	○	○	○	○	○
BLOCK	○	○	○	○	○	○	○	○
Status Polling and Immediate Commands								
<ESC>!?	○	○	○	○	○	○	○	○
<ESC>!C	○	○	○	○	○	○	○	○
<ESC>!D	○	○	○	○	○	○	○	○
<ESC>!O	○	○	○	○	○	○	○	○
<ESC>!P	○	○	○	○	○	○	○	○
<ESC>!Q	○	○	○	○	○	○	○	○
<ESC>!R	○	○	○	○	○	○	○	○
<ESC>!S	○	○	○	○	○	○	○	○
<ESC>!F	○	○	○	○	○	○	○	○
<ESC>!.	○	○	○	○	○	○	○	○
~!@	○	○	○	○	○	○	○	○
~!A	○	○	○	○	○	○	○	○
~!C	○	○	○	○	○	○	○	○
~!D	○	○	○	○	○	○	○	○
~!E	○	○	○	○	○	○	○	○
~!F	○	○	○	○	○	○	○	○

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
~!l	○	○	○	○	○	○	○	○
~!T	○	○	○	○	○	○	○	○
<ESC>Y				○	○			
<ESC>Z				○	○			
Commands for Windows Driver								
!B	○	○	○	○	○	○	○	○
!J	○	○	○	○	○	○	○	○
!N	○	○	○	○	○	○	○	○
File Management Commands								
DOWNLOAD	○	○	○	○	○	○	○	○
EOP	○	○	○	○	○	○	○	○
FILES	○	○	○	○	○	○	○	○
KILL	○	○	○	○	○	○	○	○
MOVE	○	○	○	○	○	○	○	○
RUN	○	○	○	○	○	○	○	○
BASIC Commands and Functions								
ABS()	○	○	○	○	○	○	○	○
ASC()	○	○	○	○	○	○	○	○
CHR\$()	○	○	○	○	○	○	○	○
XOR\$()	○	○	○	○	○	○	○	○
END	○	○	○	○	○	○	○	○
EOF()	○	○	○	○	○	○	○	○
OPEN	○	○	○	○	○	○	○	○
CLOSE	○	○	○	○	○	○	○	○

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
WRITE	○	○	○	○	○	○	○	○
READ	○	○	○	○	○	○	○	○
SEEK	○	○	○	○	○	○	○	○
LOF()	○	○	○	○	○	○	○	○
LOC()	○	○	○	○	○	○	○	○
FREAD\$()	○	○	○	○	○	○	○	○
PUT	○	○	○	○	○	○	○	○
GET	○	○	○	○	○	○	○	○
COPY	○	○	○	○	○	○	○	○
FOR...NEXT LOOP	○	○	○	○	○	○	○	○
WHILE...WEND	○	○	○	○	○	○	○	○
DO...LOOP	○	○	○	○	○	○	○	○
IF...THEN...ELSE...ENDIF LOOP	○	○	○	○	○	○	○	○
GOSUB...RETURN	○	○	○	○	○	○	○	○
GOTO	○	○	○	○	○	○	○	○
INP\$()	○	○	○	○	○	○	○	○
INP()	○	○	○	○	○	○	○	○
LOB()	○	○	○	○	○	○	○	○
PREINPUT	○	○	○	○	○	○	○	○
POSTINPUT	○	○	○	○	○	○	○	○
SET FILTER ON/OFF	○	○	○	○	○	○	○	○
REM	○	○	○	○	○	○	○	○
OUT	○	○	○	○	○	○	○	○
OUTR	○	○	○	○	○	○	○	○

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
GETKEY()	○	○	○	○	○	○	○	○
INT()	○	○	○	○	○	○	○	○
LEFT\$()	○	○	○	○	○	○	○	○
LEN()	○	○	○	○	○	○	○	○
MID\$()	○	○	○	○	○	○	○	○
RIGHT\$()	○	○	○	○	○	○	○	○
STR\$()	○	○	○	○	○	○	○	○
STRCOMP()	○	○	○	○	○	○	○	○
INSTR ()	○	○	○	○	○	○	○	○
TRIM\$()	○	○	○	○	○	○	○	○
LTRIM\$()	○	○	○	○	○	○	○	○
RTRIM\$()	○	○	○	○	○	○	○	○
TEXTPIXEL()	○	○	○	○	○	○	○	○
BARCODEPIXEL()	○	○	○	○	○	○	○	○
VAL()	○	○	○	○	○	○	○	○
NOW\$()	○	○	○	○	○	○	○	○
NOW	○	○	○	○	○	○	○	○
FORMAT\$()	○	○	○	○	○	○	○	○
DATEADD()	○	○	○	○	○	○	○	○
FSEARCH()	○	○	○	○	○	○	○	○
TOUCHPRESS()						○* ₁	○	○* ₂
RECORDSET\$()	○	○	○			○	○	○
REPLACE\$()	○	○	○			○	○	○

*₁ TJ-4021TN/TJ-4121TN only, *₂ TJ-4422TN/TJ-4522TN only.

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
Device Reconfiguration Commands								
SET COUNTER	○	○	○	○	○	○	○	○
SET CUTTER	○	○	○ Except RFID tags			○	○ Except RFID tags	○
SET PARTIAL_CUTTER	○	○	○ Except RFID tags			○	○ Except RFID tags	○
SET BACK	○	○	○			○	○	○
SET KEYn	○	○	○	○	○	○	○	○
SET LEDn	○	○	○	○	○	○	○	○
SET PEEL	○	○	○ Except RFID tags			○	○ Except RFID tags	○
SET REWIND	○	○	○			○	○	○
SET TEAR & SET STRIPER	○	○	○	○	○	○	○	○
SET GAP	○	○	○	○	○	○	○	○
SET BLINE	○	○	○	○	○	○	○	○
SET HEAD	○	○	○	○	○	○	○	○
SET RIBBON	○	○	○			○	○	○
SET ENCODER	○	○	○			○	○	○
SET RIBBONEND	○	○	○			○	○	○
SET COM1	○	○	○	○	○	○	○	○
SET PRINTKEY	○	○	○	○	○	○	○	○
SET REPRINT	○	○	○	○	○	○	○	○
SET FEED_LEN	○	○	○	○	○	○	○	○
GETSENSOR()	○	○	○	○	○	○	○	○
GETSETTING\$()	○	○	○	○	○	○	○	○

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
SET USBHOST	○	○	○			○	○	○
SET AUTORUN	○	○	○	○	○	○	○	○
SET RESPONSE	○	○	○			○	○	○
SET DAYLIGHT_SAVE	○	○	○	○	○	○	○	○
SET REGISTRATION	○	○	○	○	○	○	○	○
PEEL	○	○	○ Except RFID tags			○	○ Except RFID tags	○
LED1, LED2, LED3	○	○	○	○	○	○	○	○
KEY1, KEY2, KEY3	○	○	○	○	○	○	○	○
SET SENSOR_REF	○	○	○	○	○	○	○	○
Printer Global Variables								
@LABEL	○	○	○	○	○	○	○	○
YEAR	○	○	○	○	○	○	○	○
MONTH	○	○	○	○	○	○	○	○
DATE	○	○	○	○	○	○	○	○
WEEK	○	○	○	○	○	○	○	○
HOUR	○	○	○	○	○	○	○	○
MINUTE	○	○	○	○	○	○	○	○
SECOND	○	○	○	○	○	○	○	○
@YEAR	○	○	○	○	○	○	○	○
@MONTH	○	○	○	○	○	○	○	○
@DATE	○	○	○	○	○	○	○	○
@DAY	○	○	○	○	○	○	○	○
@HOUR	○	○	○	○	○	○	○	○
@MINUTE	○	○	○	○	○	○	○	○

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
@SECOND	○	○	○	○	○	○	○	○
_MODEL\$	○	○	○	○	○	○	○	○
_SERIAL\$	○	○	○	○	○	○	○	○
_VERSION\$	○	○	○	○	○	○	○	○
Bluetooth Module Setting Commands								
BTNAME		○	○	○	○			○
BTPINCODE				○				
BT MODE		○	○	○	○			○
SET BTLINKBACK				○				
Wi-Fi Module Setting Commands								
WLAN OFF		○	○		○	○	○	○
WLAN SSID		○	○		○	○	○	○
WLAN WPA		○	○		○	○	○	○
WLAN WEP		○	○		○	○	○	○
WLAN DHCP		○	○		○	○	○	○
WLAN IP		○	○		○	○	○	○
WLAN PORT		○	○		○	○	○	○
Internal Ethernet Setting Commands								
NET DHCP	○	○	○			○	○	○
NET IP	○	○	○			○	○	○
NET PORT	○	○	○			○	○	○
NET NAME	○	○	○			○	○	○
Mobile Printer Setting Commands								
SET PRINTQUALITY				○	○			

Command	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4650TNWBR TD-4750TNWBR	RJ-2035B RJ-3035B	RJ-2055WB RJ-3055WB	TJ-4020TN TJ-4021TN TJ-4120TN TJ-4121TN	TJ-4021TNR TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN TJ-4422TN TJ-4522TN
SET STANDBYTIME				○	○			
SET SLEEPTIME				○	○			
RFID Commands								
RFID READ			○				○	
RFID WRITE			○				○	
RFIDDETECT			○				○	

brother