TSPL

TAIWAN SEMICONDUCTOR BAR CODE PRINTER SERIES

PROGRAMMING MANUAL

CLEVER • CODE

CONTENT

INDEX

Document Conventions

This manual uses the following typographic conventions.

Convention	Description
[expression list]	Items inside square brackets are optional.
<ESC>	ESCAPE (ASCII 27), control code of status polling command, which returns the printer status immediately, no matter the printer is ready or not.
~	(ASCII 126), control code of status polling command, which returns the printer status only when the printer is ready.
Note: 200 DPI: 1 mm = 8 dots ***300 DPI: 1 mm = 11.8 dots***	Arial font in bold and italic type is used for notes.

`DOWNLOAD “TEST.BAS”`
 `SET COUNTER @1 1`
 `@1=”0001”`
 `TEXT 10,10,”3”,0,1,1,@1`
 `PRINT 3,2`
 `EOP`
	The courier font is used for program code.

Setup and System Commands

● SIZE

Description

This command defines the label width and length.

Syntax

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

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

Parameter	Description
m	Label width (inch or mm)
n	Label length (inch or mm)

Note: ***200 DPI: 1 mm = 8 dots****
300 DPI: 1 mm = 11.8 dots

Example

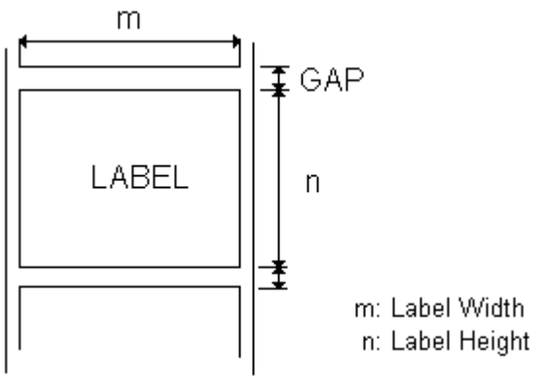
(1) English system (inch)

```
SIZE 3.5, 3.00
```

(2) Metric system (mm)

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

* As used in the printer jargon of dot matrix. Note that like inch or millimeter, it is almost always referred to in this manual as a measurement unit.



● GAP

Description

Define the gap distance between two labels

Syntax

(1) English system (inch)

GAP m, n

(2) Metric system (mm)

GAP m mm, n mm

Parameter	**Description**
m	The gap distance between two labels 0 ≤ m ≤ 1 (inch), 0 ≤ m ≤ 25.4 (mm)
n	The offset distance of the gap [-]n ≤ label length (inch or mm)

Note: 200 DPI : 1 mm = 8 dots

300 DPI : 1 mm = 11.8 dots

Example

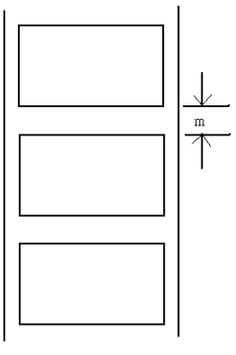
Normal gap

(1) English system (inch)

```
GAP 0.12,0
```

(2) Metric system (mm)

```
GAP 3 mm,0
```



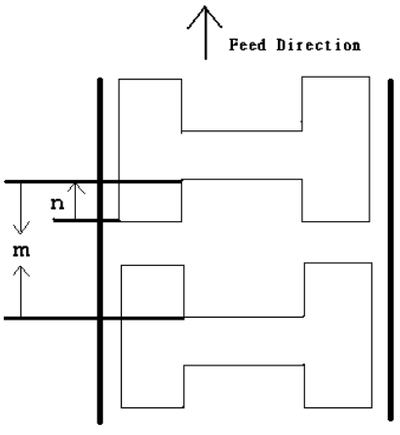
Special gap

(1) English system (inch)

GAP 0.30,-0.10

(2) Metric system (mm)

GAP 7.62 mm, -2.54 mm



● BLINE

Description

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

Syntax

(1) English system (inch)

BLINE m, n

(2) Metric system (mm)

BLINE m mm, n mm

Parameter	**Description**
m	The height of black line either in inch or mm. 0.1 ≤ m ≤ 1 (inch), 2.54 ≤ m ≤ 25.4 (mm)
n	The extra label feeding length. 0 ≤ n ≤ label length

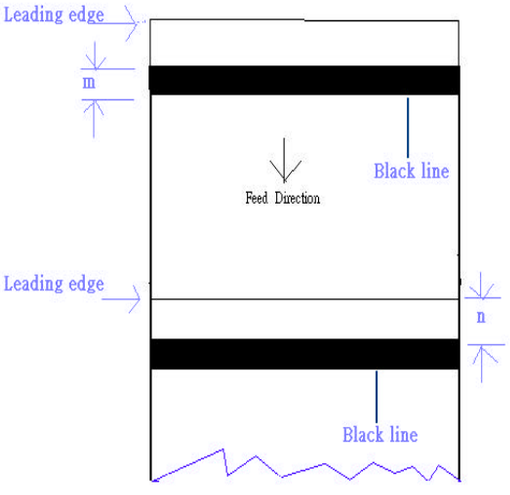
Example

(1) English system (inch)

```
BLINE 0.20,0.50
```

(2) Metric system (mm)

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



● OFFSET

Description

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

Syntax

(1) English system (inch)
OFFSET m

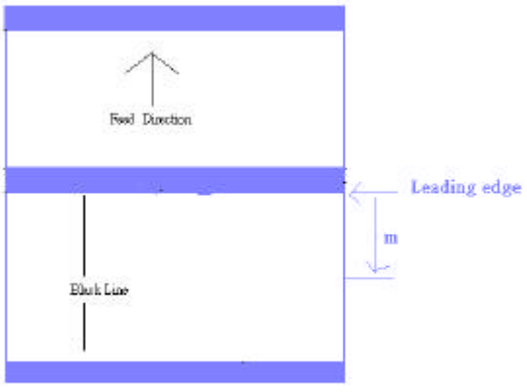
(2) Metric system (mm)
OFFSET m mm

Parameter	**Description**
m	The offset distance (inch or mm) $0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm)

Example

(1) English system (inch)
```
OFFSET 0.5
```

(2) Metric system (mm)
```
OFFSET 12.7 mm
```

● SPEED

Description

This command defines the print speed.

Syntax

SPEED n

Parameter	**Description**
n	1.0 Set print speed at 1 inch/sec (TTP-242) 1.5 Set print speed at 1.5 inch/sec (TTP-242, 243) 2.0 Set print speed at 2.0 inch/sec (TTP-242, 243) 3.0 Set print speed at 3.0 inch/sec (TTP-243)

Example

```
SPEED 2.0
```

● DENSITY

Description

This command designates the level of darkness of printing.

Syntax

DENSITY n

Parameter	Description
n	0, specifies the lightest level 15, specifies the darkest level

Example

```
DENSITY 7
```


● DIRECTION

Description

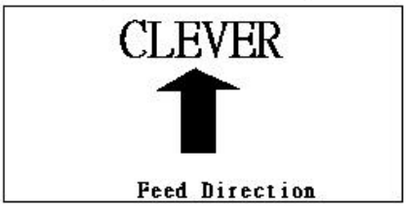
This command defines the print direction.

Syntax

DIRECTION n

Parameter	**Description**
n	0 or 1. Please refer to the illustrations below:

(DIRETCTION 0)



(DIRECTION 1)

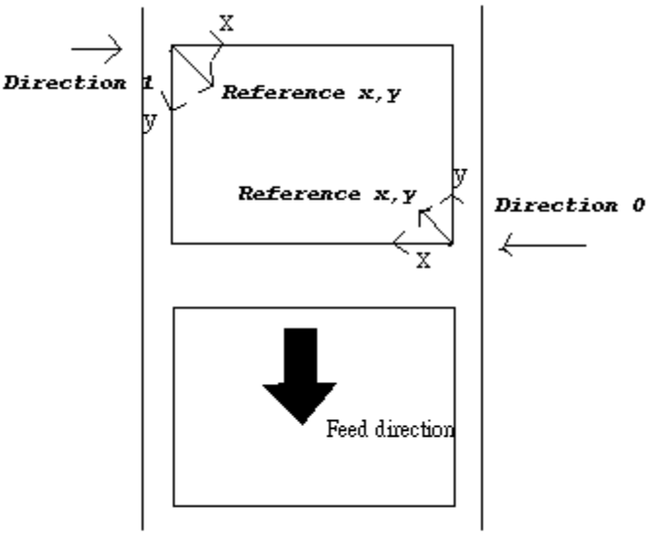
Example

```
DIRECTION 0
```


● REFERENCE

Description

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



Syntax

REFERENCE x, y

Parameter	**Description**
x	Horizontal coordinate, with “dot” as the unit.
y	Vertical coordinate, with “dot” as the unit.

Example

```
REFERENCE 10,10
```


● COUNTRY

Description

This command defines what special character to have on the portable LCD keyboard (KP-200, option) to orient the keyboard for use in different countries.

Syntax

COUNTRY n

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

Example

```
COUNTRY 001
```


● CODEPAGE

Description

This command defines the code page of international character set.

Syntax

CODEPAGE n

Parameter	**Description**
n	name or number of code page, which can be divided into 7-bit code page and 8-bit code page further. 7-bit code page name USA: USA BRI: British GER: German FRE: French DAN: Danish ITA: Italian SPA: Spanish SWE: Swedish SWI: Swiss 8-bit code page number 437: United States 850: Multilingual 852: Slavic 860: Portuguese 863: Canadian/French 865: Nordic

Note: Whether to use 7-bit or 8-bit code page is determined by the communication parameter of DATA LENGTH

.

Example

```
CODEPAGE 437
```


● CLS

Description

This command clears the image buffer.

Syntax

CLS

Parameter	Description
N/A	N/A

Example

CLS

● FEED

Description

This command feeds label with the specified length (in dot).

Syntax

FEED n

Parameter	**Description**
n	unit: dot 1 ≤ n ≤ 65535

Example

```
FEED 40
```

Note: ***200 DPI: 1 mm = 8 dots***
300 DPI: 1 mm = 11.8 dots

● FORMFEED

Description

This command feeds label to the beginning of next label.

Syntax

FORMFEED

Parameter	Description
N/A	N/A

Example

```
FORMFEED
```


● HOME

Description

It is not expected the first label will be printed on the right position when the printer power is turned on. This command will feed the label to find the gap first and then back feed to the beginning of label, on condition that the size of the label was set in advance.

Note: The label length must be longer than 30 mm when using this command.

Syntax

HOME

Parameter	**Description**
N/A	N/A

Example

```
HOME
```


● PRINT

Description

This command prints the label form stored in the image buffer.

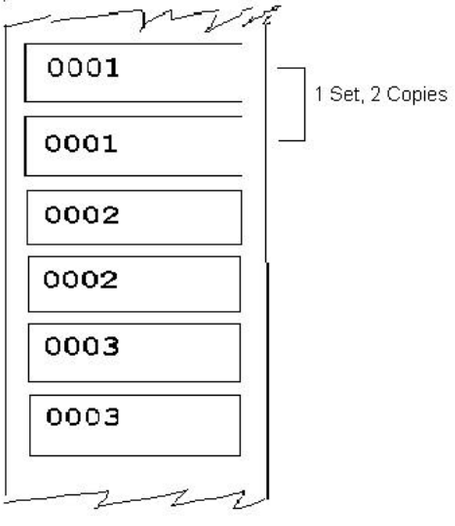
Syntax

PRINT m [,n]

Parameter	**Description**
m	Specifies how many sets of labels will be printed. 1 ≤ m ≤ 65535
n	Specifies how many copies should be printed for each set of label. 1 ≤ n ≤ 65535

Example

```
DOWNLOAD “TEST.BAS”
SET COUNTER @1 1
@1=”0001”
TEXT 10,10,”3”,0,1,1,@1
PRINT 3,2
EOP
```



● SOUND

Description

This command is used to control the sound frequency of the beeper. There are 10 levels of sounds. The timing control of the sound can be set by the "interval" parameter.

Syntax

SOUND level, interval

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

Example

```
SOUND 5,200
SOUND 3,200
SOUND 3,200
SOUND 4,200
SOUND 2,200
SOUND 2,200
SOUND 1,120
SOUND 2,200
SOUND 3,200
SOUND 4,200
SOUND 5,200
SOUND 5,200
SOUND 5,400
SOUND 5,200
SOUND 3,200
SOUND 3,200
SOUND 4,200
SOUND 2,200
```


● CUT

Description

At this command, the printer will activate the cutter to cut the labels. Applicable models: TTP/TDP 243

Syntax

Cut

Parameter	**Description**
None	N/A

● LIMITFEED

Description

When feeding labels, if the gap sensor is not set to a suitable strength, the printer will be unable to locate the correct position of the gap. This command is used stop label feeding and make the red LED flash if the printer does not locate gap after feeding the length of one label plus one preset value. Applicable models: TTP/TDP 243

Syntax

LIMITFEED n (inch, the English system)

LIMITFEED n mm (mm, the metric system)

Parameter	**Description**
n	inch or mm

Remark

The setting will remain resident in memery.

The default value is 4 inches when printer initializes.

For metric system, there must be a space between parameter n and mm.

Label Formatting Commands

● BAR

Description

This command is used to draw a line or a bar on the label form.

Syntax

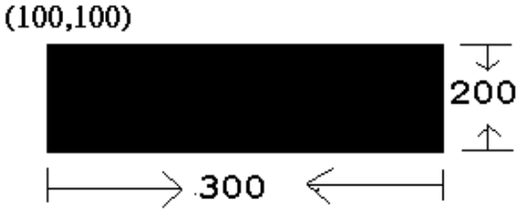
BAR x, y, width, height

Parameter	Description
x	The upper left corner x-coordinate in dot
y	The upper left corner y-coordinate in dot
width	The width of bar in dot
height	The height of bar in dot

Note: 200 DPI: 1 mm = 8 dots
300 DPI: 1 mm = 11.8 dots

Example

BAR 100, 100, 300, 200



● BARCODE

Description

This command is used to print 1D barcodes on label form.

The available bar codes in TTP-243 are listed below:

- Code 128 (switching code subset automatically)
- Code 128M (switching code subset manually)
- EAN 128 (switching code subset automatically)
- Interleaved 2 of 5
- Interleaved 2 of 5 with check digit
- Code 39
- Code 39 with check digit
- Code 93
- EAN 13
- EAN 13 with 2 digits add-on
- EAN 13 with 5 digits add-on
- EAN 8
- EAN 8 with 2 digits add-on
- EAN 8 with 5 digits add-on
- Codabar
- Postnet
- UPC-A
- UPC-A with 2 digits add-on
- UPC-A with 5 digits add-on
- UPCE
- UPCE with 2 digits add-on
- UPCE with 5 digits add-on

Syntax

BARCODE X, Y, ”code type”, height, human readable, rotation, narrow, wide, “code”

Parameter	**Description**
X	Specify the x-coordinate of the bar code on label
Y	Specify the y-coordinate of the bar code on label
code type	
128	Code 128, switching code subset A, B, C automatically
128M	Code 128, switching code subset A, B, C manually.

Control code	A	B	C
096	FNC3	FNC3	NONE
097	FNC2	FNC2	NONE
098	SHIFT	SHIFT	NONE
099	CODE C	CODE C	NONE
100	CODE B	FNC4	CODE B
101	FNC4	CODE A	CODE A
102	FNC1	FNC1	FNC1
103	Start (CODE A)		
104	Start (CODE B)		
105	Start (CODE C)		

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.

EAN128	Code 128, switching code subset A, B, C automatically
25	Interleaved 2 of 5
25C	Interleaved 2 of 5 with check digits
39	Code 39
39C	Code 39 with check digits
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

height	bar code height in dot
human readable	0: human not readable
	1: human readable
rotation	Rotate bar code clockwise
	in degrees
0	non rotation
90	rotate 90 degrees clockwise
180	rotate 180 degrees clockwise
270	rotate 270 degrees clockwise
narrow	width of narrow element in dot
wide	width of wide element in dot

	narrow : wide 1:1	Narrow : wide 1:2	narrow : wide 1:3	narrow : wide 2:5
128	10x	N/A	N/A	N/A
EAN128	4x	N/A	N/A	N/A
25	N/A	10x	10x	5x
25C	N/A	10x	10x	5x
39	N/A	10x	10x	5x
39C	N/A	10x	10x	5x
93	N/A	N/A	10x	N/A
EAN13	4x	N/A	N/A	N/A
EAN13+2	4x	N/A	N/A	N/A
EAN13+5	4x	N/A	N/A	N/A
EAN 8	4x	N/A	N/A	N/A

EAN 8+2	4x	N/A	N/A	N/A
EAN 8+5	4x	N/A	N/A	N/A
CODA	N/A	10x	10x	5x
POST	1x	N/A	N/A	N/A
UPCA	4x	N/A	N/A	N/A
UPCA+2	4x	N/A	N/A	N/A
UPCA+5	4x	N/A	N/A	N/A
UPCE	4x	N/A	N/A	N/A
UPCE+2	4x	N/A	N/A	N/A
UPCE+5	4x	N/A	N/A	N/A

code number the bar code content

Example

BARCODE 100,100,”39”,96,1,0,2,4,”1000”

BARCODE 10,10,”128M”,48,1,0,2,2,”!104!096ABCD!101EFGH”

(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.)

● BITMAP

Description

This command is used to draw bitmap images (Not BMP graphic file).

Syntax

BITMAP X, Y, width, height, mode, bitmap data…

Parameter	**Description**
X	Specify the x-coordinate of the bitmap image
Y	Specify the y-coordinate of the bitmap image
width	The width of the image in bytes
height	The height of the image in dot
mode	Graphic mode is listed below:
0	OVERWRITE
1	OR
2	XOR
bitmap data	The bitmap data

Example

BITMAP 100,100,10,1,2,1111111111

● BOX

Description

This command is used to draw rectangles on the label.

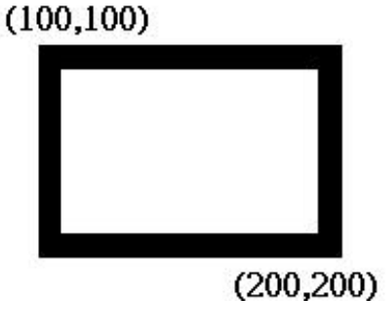
Syntax

BOX X_start, Y_start, X_end, Y_end, line thickness

Parameter	**Description**
X_start	Specify x-coordinate of upper left corner in dot
Y_start	Specify y-coordinate of upper left corner in dot
X_end	Specify x-coordinate of lower right corner in dot
Y_end	Specify y-coordinate of lower right corner in dot
line thickness	Line thickness of the box in dot

Example

BOX 100,100,200,200,5



● ERASE

Description

This command is used to blot out a specified region from image.

Syntax

ERASE X_start, Y_start, X_width, Y_height

Parameter	Description
X_start	The x-coordinate of the starting point in dot
Y_start	The y-coordinate of the starting point in dot
X_width	The region width in x-axis direction in dot
Y_height	The region height in y-axis direction in dot

Example

ERASE 100,100,200,200

● DMATRIX

Description

This command is used to define the DataMatrix 2D bar code.

Syntax

DMATRIX x, y, width, height, [xm, row, col], expression

Parameter	Description
x	Horizontal start position in dot
y	Vertical start position in dot
width	The width of barcode area in dot
height	The height of barcode area in dot
xm	Module size in dot
row	Rows of the bar code
col	Columns of the bar code

Example

DMATRIX 10,10,400,400,”DMATRIX”

● MAXICODE

Description

This command is used to define a 2D Maxicode.

Syntax

MAXICODE x, y, ”class, country, post, message”

For mode 2 or 3:

MAXICODE x, y, "class, country, postalcode, low priority message"

if country is 840, the postalcode could be 999999999 or 99999,9999

(Both expressions are acceptable, with or without comma in between)

for all the other countries, the code is up to 6 alphanumeric characters.

For mode 4,5,6

MAXICODE x, y, "message"

SET MAXIMODE [2..6]

* if the SET MAXIMODE is not used, the printer will use mode 2 or 3 (mode 2 for country 840, mode 3 for all the other countries)
If the input command is neither regular format (service code, country code, postal code), nor AIM special format, then mode 4 will be used.

* Mode 4, 5, 6 do not have any syntax in the HPM and LPM. But the SET MAXICODE command has to be used to select mode.

* AIM special format is supported, see page 23 in the spec.
Example:

```
MAXICODE
24,24,"[)>-01 96152382802 840 001 1Z00004951 UPSN 06X61          0 15
9 1234567 1/1 Y 634 ALPHA DR PITTSBURGH PA           - "
```


Parameter	Description
x	X-coordinate of the staring point in dot
y	Y-coordinate of the starting point in dot
class	Class of service, 3-digit number
country	Country code, 3-digit number
post	Post code USA: 5-digit, 4-digit number Canada: 6-digit number
message	Barcode content

Example

For USA:
MAXICODE 100,100,”300,840,06810,7317,DEMO FOR MAXICODE”

For Canada:
MAXICODE 100,100,”300,840,107317,DEMO FOR MAXICODE”

EXAMPLES:

```
REM MODE 4
SPEED 2
CLS
SIZE 4.00,3.00
GAP 0.10,0
DENSITY 10
MAXICODE 24,24,"THIS IS A 93 CHARACTER CODE SET A MESSAGE
THAT FILLS A MODE 4, UNAPPENDED, MAXICODE SYMBOL..."
BOX 424,16,700,60,2
DIRECTION 0
PRINT 1

REM MODE 5
SPEED 2
CLS
SIZE 4.00,3.00
GAP 0.10,0
DENSITY 10
```



```
SET MAXIMODE 5
MAXICODE 24,24,"THIS IS A 93 CHARACTER CODE SET A MESSAGE
THAT FILLS A MODE 4, UNAPPENDED, MA"
BOX 424,16,700,60,2
DIRECTION 0
PRINT 1


REM this will use mode 2 automatically
REM please check the difference of the 3 MAXICODE commands
REM all 3 MAXIMODE produce same symbol

SPEED 2
CLS
SIZE 4.00,3.00
GAP 0.10,0
DENSITY 10
MAXICODE
24,24,"[)>-01 96152382802 840 001 1Z00004951 UPSN 06X610 15
9 1234567 1/1 Y 634 ALPHA DR PITTSBURGH PA            - "
MAXICODE
24,300,"001,840,152382802,[)>-01 961Z00004951 UPSN 06X610 159
 123456 1/1 Y 634 ALPHA DR PITTSBURGH PA              - "
MAXICODE
400,24,"001,840,15238,2802,[)>-01 961Z00004951 UPSN 06X610 159
 123456 1/1 Y 634 ALPHA DR PITTSBURGH PA              - "
DIRECTION 0
PRINT 1

REM this will use mode 3 automatically
SPEED 2
CLS
SIZE 4.00,3.00
GAP 0.16,0
DENSITY 10
MAXICODE
24,24,"[)>-01 96B1050 056 999 1Z00004951 UPSN 06X610 159 1
```


234567 1/1 Y 634 ALPHA DR PITTSBURGH PA - "
MAXICODE
24,300,"001,056,B1050,[)>-01 961Z00004951 UPSN 06X610 159 123
4567 1/1 Y 634 ALPHA DR PITTSBURGH PA - "
MAXICODE
400,24,"001,056,B1050,[)>-01 961Z00004951 UPSN 06X610 159 123
4567 1/1 Y 634 ALPHA DR PITTSBURGH PA - "
DIRECTION 0
PRINT 1

● PDF417

Description

This command is used to define a PDF417 2D barcode.

Syntax

PDF417 x, y, width, height, rotate, [option], expression

Parameter	**Description**
x	X-coordinate of the starting point in dot
y	Y-coordinate of the starting point in dot
width	The width of barcode in dot
height	The height of barcode in dot
rotate	Rotation counterclockwise. 0: 0 degree 90: 90 degrees 180: 180 degrees 270: 270 degrees
expression	Barcode text or string expression to be printed on label.
[option]	
P	Data compression method 0: Auto encoding 1: Binary mode
E	Error correction level Range: 0~8
M	Center pattern in barcode area. Range: 0~1 0: The pattern will print upper left justified in the area. 1: The pattern is printed middle of area.
U x,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

Example

PDF417 100,200,200,300,0,P0,E1,U100,400,10,”abcdef”

● PUTPCX

Description

This command is used to print PCX format image.

Syntax

PUTPCX X, Y, ”filename”

Parameter	Description
X	The x-coordinate of the PCX image
Y	The y-coordinate of the PCX image
filename	The downloaded PCX filename.

Note: Only two colors (black and white) are supported by this command.

Example

PUTPCX 100,100,”LOGO.PCX”

● REVERSE

Description

This command is used to reverse a region of image.

Syntax

REVERSE X_start, Y_start, X_width, Y_height

Parameter	Description
X_start	The x-coordinate of the starting point in dot
Y_start	The y-coordinate of the starting point in dot
X_width	The region width in x-axis direction in dot
Y_height	The region height in y-axis direction in dot

Example

REVERSE 100,100,200,200

● TEXT

Description

This command is used to print text on label

Syntax

TEXT X, Y, ”font”, rotation, x-multiplication, y-multiplication, “content”

Parameter	**Description**
X	The x-coordinate of the text
Y	The y-coordinate of the text
font:	font name
1	8 x 12 dot font
2	12 x 20 dot font
3	16 x 24 dot font
4	24 x 32 dot font
5	32 x 48 dot font
TST24.BF2	Traditional Chinese 24 x 24 font (Big-5 code)
TST16.BF2	Traditional Chinese 16 x 16 font (Big-5 code)
TTT24.BF2	Traditional Chinese 24 x 24 font (TEL code)
TSS24.BF2	Simplified Chinese 24 x 24 font (GB code)
TSS16.BF2	Simplified Chinese 16 x 16 font (GB code)
K	Japanese 24 x 24 font (Shift JIS code for Windows, JIS for DOS mode)
L	Japanese 16 x 16 font (Shift JIS code, JIS for DOS mode)
K	Korea 24 x 24 font (KS code)
rotation:	The rotation angle of text
0	0 degree
90	90 degrees, in clockwise direction
180	180 degrees, in clockwise direction
270	270 degrees, in clockwise direction
x-multiplication:	Horizontal multiplication, up to 8x
1~8	
y-multiplication:	Vertical multiplication, up to 8x
1~8	

NOTE: Font "5" supports capital characters only.

If there is any double quote (") within the text, please change it " to \["]

If there is any carriage return within the text, please change CR to \[R].

If there is any line feed within the text, please change LF to \[L]

Example

TEXT 100,100,"4",0,1,1,"DEMO FOR TEXT"

Status Polling Commands (RS-232)

● <ESC>!?

Description

This command is used to obtain the printer status. An inquiry request is solicited by sending an <ESC> (ASCII 27, escape character) as the beginning control character to the printer. It can be sent any time, even in the event of printer error. One byte character is returned, of which one bit is used to flag the printer's current readiness status. If 0 is returned, the printer is ready to print labels.

Bit	Status
0	Carriage opened (TTP-243, 243M)
1	Paper jam
2	Out of paper
3	Out of ribbon
4	Pause (TTP-243, 243M) Reserved (TTP-242)
5	Printing
6	Cover opened (TTP-243M), Reserved (TTP-242,243)
7	Error

Syntax

<ESC>!?

Parameter	Description
N/A	N/A

● <ESC>!R

Description

This command is used to reset the printer. It can be sent at any time as long as the printer is powered on and not in the dump mode. The beginning of the command is an ESCAPE character (ASCII 27). The files downloaded in memory will be deleted.

Syntax

<ESC>!R

Parameter	Description
N/A	N/A

- **~!A**

Description

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

Syntax

~!A

Parameter	Description
N/A	N/A

- **~!T**

Description

This command is used to inquire the model name and number of the printer. They are returned in ASCII characters.

Printer type	Returned string
TTP/TDP-243	TTP/TDP243
TTP/TDP-342	TTP/TDP342

Syntax

~!T

Parameter	Description
N/A	N/A

- **~!C**

Description

This command is used to inquire the presence of Real Time Clock. One byte is returned from the printer, indicating whether or not the RTC is installed.

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

Syntax

~!C

Parameter	Description
N/A	N/A

- **~!I**

Description

The command is used to inquire the code page and country setting of the printer.

The returned information is given in the following format

codepage, country code

ex: 8 bit: 437, 001

7 bit: USA, 001

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

Syntax

~!I

Parameter	**Description**
N/A	N/A

- **~!F**

Description

This command is used to inquire about 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 of the last filename is 0x0D or 0x1A.

Syntax

~!F

Parameter	Description
N/A	N/A

- **~!@**

Description

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

Syntax

~!@

Parameter	**Description**
N/A	N/A

Message Translation Protocols

● <ESC>!, <ESC>&

Description

The error message is returned by printer either to label or through RS-232 to PC by setting SET DEBUG LABEL or SET DEBUG RS232.

The error message is enclosed in between the beginning <ESC>! identifier and the ending <ESC>& identifier.

Note: <ESC> is the ESCAPE (ASCII 27) character

Syntax

<ESC>!Error Message<ESC>&

Parameter	**Description**
N/A	N/A

Example

<ESC>!Syntax Error<ESC>&

- **~#**

Description

The beginning identifier of the prompt message is sent from the printer to the portable keyboard (KP-200). The ending identifier is ~&.

Syntax

~#Prompt~&

Parameter	**Description**
N/A	N/A

Example

~#SELF TEST~&

Commands for Windows Driver

● !B

Description

This command is used to store bitmap image data in the memory. Behind the nnn is the bitmap data.

Syntax

!Bnnn

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

Example

!B100

- **!J**

Description

This command is used to print the bitmap data at the specified position (in y-direction).

Syntax

!Jnnnn

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

Example

!J0100

- **!N**

Description

This command is used to print specified number of labels.

Syntax

!Nnnn

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

Example

!N001

File Management Commands

● DOWNLOAD

Description

"DOWNLOAD" is a header of the file that is to be saved in the printer's memory.

The downloaded files can be programs, data files, PCX graphic files and bitmap font files

The detailed descriptions regarding the download syntax for different files are shown below:

Syntax

1. Download a program file

The program listed below will download to printer memory.

```
DOWNLOAD "EXAMPLE.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 100,100,"3",0,1,1,"EXAMPLE PROGRAM"
PRINT 1
EOP
```

Note: 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.

2. Download a data file

The syntax of earlier version format (TTP-242) is as shown:

DOWNLOAD “FILENAME”, DATA SIZE, DATA CONTENT…where

FILENAME is the name of data file that will remain resident in the printer memory.

DATA SIZE parameter is the actual size of the data file without header.

Note: CR (carriage return) 0x0D and LF (Line Feed) 0x0A are accepted by printer.

3. Download a PCX graphic file

This printer accepts the two color (black and white) PCX graphic format. To download a PCX file to printer, users have to add header in the beginning of PCX file. The format is as follows:

DOWNLOAD “FILENAME.PCX”, FILE SIZE, PCX CONTENT…

where

FILENAME.PCX is the name of the file resident in the printer memory.

FILESIZE is the original size of PCX graphic file in bytes without header.

PCX CONTENT is the original graphic data.

There are two methods to add header to PCX graphic file:

(1) Use HEX editor to add header to the beginning of PCX file.

(2) Under DOS environment, follow the steps below to add header file to printer.

A. C:\>MODE COM2 96,N,8,1

B. C:\>COPY CON COM2

DOWNLOAD “LOGO.PCX”,4910,^Z

C. C:\>COPY LOGO.PCX COM2

4. Download font file

Two categories of fonts are available in this printer, which are *fixed pitch font* and *variable pitch font* respectively. Double byte character set is also supported by the two categories of fonts. The detailed specifications of fonts are as describedbelow:

(1) **Fixed pitch font**

A. BF1 font file

Extension with BF1 font file is the format of ASCII sequential arrangement.

File format as follows:

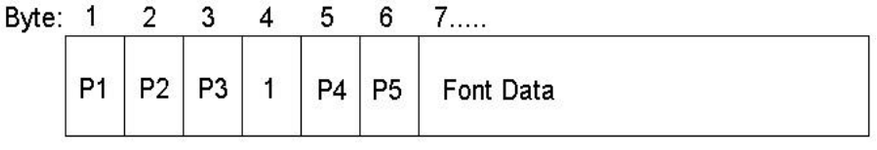
DOWNLOAD “FILENAME.BF1”, FILE SIZE, FONT DATA …

where

FILE SIZE: The original size of *.BF1 file; “DOWNLOAD “FILENAME.BF1”, FILE SIZE” are not included.

FONT DATA: The font arrangement is as shown below:

Single byte character set:



where

P1: 0 (constant)

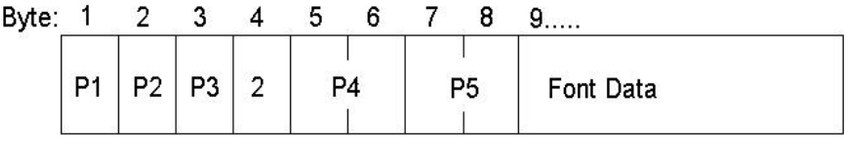
P2: Character height (in dot)

P3: Character width (in dot)

P4: ASCII code of first character

P5: ASCII code of last character

Double bytes character set



where

P1: 0 (constant)

P2: Character height (in dot)

P3: Character width (in dot)

P4: ASCII code of first character

P5: ASCII code of last character

B. BF2 font file

BF2 font file is characterized by the feature that its font is searched by a specified address formula. The semi-colon must be added at the end of formula, which serves as the separator between formula and font data.

File format is as follows:

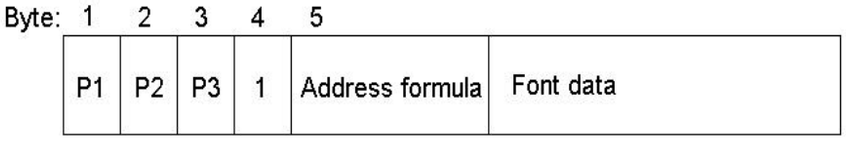
DOWNLOAD “FILENAME.BF2”, FILE SIZE, FONT DATA …

where

FILE SIZE: The original size of *.BF2 file; “DOWNLOAD “FILENAME.BF2”, FILE SIZE” are not included.

FONT DATA: The font arrangement is as shown:

Single byte character set



where

P1: 0 (constant)

P2: Character height (in dot)

P3: Character width (in dot)

Address formula:

Example: (LB-32)*24+18;

where

LB: A fixed variable for single byte character

32: The starting ASCII code

24: The size of character in bytes

18: The offset bytes which begin from P1 to ;

;: Semi-colon, the ending of formula

Double bytes character set

Byte: 1 2 3 4 5 .....

P1	P2	P3	2	Address formula	Font data

where

P1: 0 (constant)

P2: Character height (in dot)

P3: Character width (in dot)

Address formula:

Example: (94*(HB-163)+LB-161)*72+128;

where

HB: A fixed variable for double byte character, which means high byte

LB: A fixed variable which means low byte

163: The starting ASCII code of high byte

161: The staring ASCII code of low byte

72: The size of character in byte

128: The offset bytes which begin from P1 to ;

;: Semi-colon, the ending of formula

C. BF3 font file

BF3 font file is characterized by the feature that the font is searched by an address table; and BF3 character set, that it only encloses a few daily used characters, not the whole character set. For example, there are around 13,000 Chinese characters in the general font file, but maybe only 100 of them are actually used in your application. In this situation, you may choose BF3 font because it takes up less memory of the printer. BF3 font supports double-byte character only.

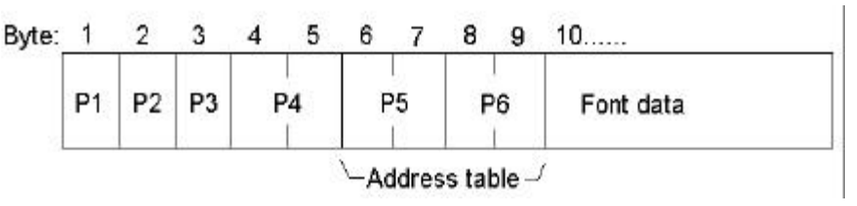
File format is as following:

DOWNLOAD “FILENAME.BF3”, FILE SIZE, FONT DATA…

where

FILE SIZE: The original size of *.BF3 file. “DOWNLOAD “FILENAME.BF3”, FILE SIZE” are not included.

FONT DATA: The font arrangement is as shown below:



where

P1: 0 (constant)

P2: Character height (in dot)

P3: Character width (in dot)

P4: Character counts

P5: Character code. Character codes are arranged in increasing order, not decreasing order.

P6: Character address

(2) **Variable pitch font**

A. VF1 font file

Font files with VF1 extension name use address table to index fonts.

File format is as follows:

DOWNLOAD “FILENAME.VF1”, FILE SIZE, FONT DATA…

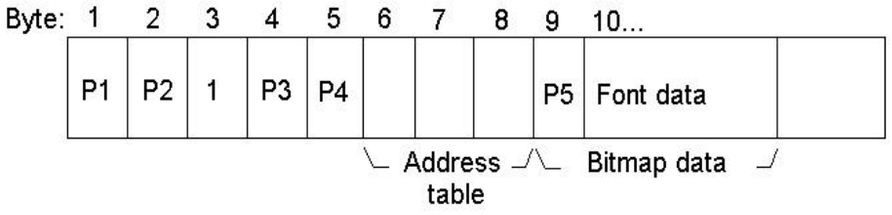
where

FILE SIZE: The original size of *.VF1 file, “DOWNLOAD “FILENAME.VF1”, FILE SIZE” are not included.

FONT DATA: The font arrangement is as shown below:

Single byte character set

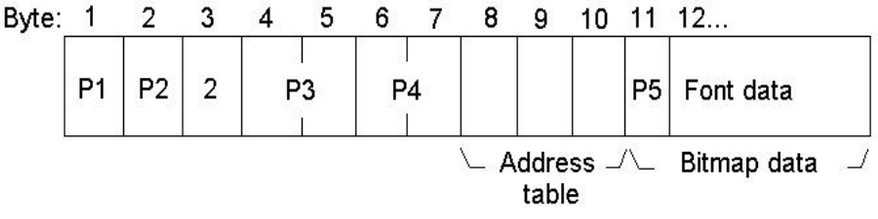
.



where

P1: 0 (constant)
P2: Character height (in dot)
P3: ASCII code of first character
P4: ASCII code of last character
P5: Character width (in dot)

Double byte character set



where

P1: 0 (constant)
P2: Character height (in dot)
P3: ASCII code of first character
P4: ASCII code of last character
P5: Character width (in dot)

B. VF2 font

Font files with VF2 extension name use address formula to index fonts

File format is as follows:

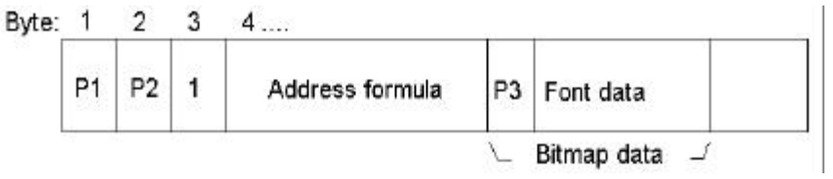
DOWNLOAD “FILENAME.VF2”, FILE SIZE, FONT DATA…

where

FILE SIZE: The original size of *.VF2 file, “DOWNLOAD “FILENAME.VF2”, FILE SIZE” are not included.

FONT DATA: The font arrangement is as shown below:

Single byte character set



where

P1: 0 (constant)

P2: Character height (in dot)

P3: Character width (in dot)

Address formula:

Example: (LB-32)*24+18;

where

LB: A fixed variable for single byte character

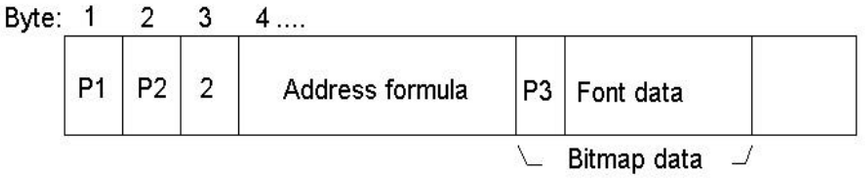
32: The starting ASCII code

24: The size of character in bytes

18: The offset bytes beginning from P1 to ;

;: Semi-colon, the ending of formula

Double byte character set



where

P1: 0 (constant)

P2: Character height (in dot)

P3: Character width (in dot)

Address formula:

Example: (94*(HB-163)+LB-161)*72+128;

where

HB: A fixed variable for double byte

character, meaning high byte
LB: A fixed variable, meaning low byte
163: The starting ASCII code of high byte
161: The staring ASCII code of low byte
72: The size of character in byte
128: The offset bytes beginning from P1 to ;
;: Semi-colon, the ending of formula

● REDRAW

Description

This command is used to copy font file to the flash memory of the font cartridge

Syntax

REDRAW n, size, font data…

Parameter	**Description**
n	The Number of flash memory. Available numbers are 1~4. The size of each flash memory is 1MB.
size	The file size of font file

Note: If the size of font file is larger that 1MB, you have to split it into two files and redraw to the next flash memory respectively.

● EOP

Description

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

Syntax

EOP

Example

```
DOWNLOAD “DEMO.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 100,100,”3”,0,1,1,”DEMO PROGRAM”
FOR I=1 TO 10
I=I+1
NEXT
PRINT 1
EOP
```


● FILES

Description

This command lists the files that are resident in the printer memory. The total memory size as well as available memory size are also listed.

Syntax

FILES

Example

Follow the steps below to list the files that are saved in printer memory in DOS environment.

C:\>MODE COM2 96,N,8,1

C:\>COPY CON COM2

FILES

^Z

C:\>

● KILL

Description

This command deletes a file in the printer memory. The wild card can be called into use in this command.

Syntax

KILL “FILENAME”
KILL “*.PCX”
KILL “*”

Example

Users can use printer SELF TEST utility to list printer configurations and files saved in the printer memory, or use the FILES command to inquire the files saved in printer.
Follow the steps below to delete files in the printer memory.

```
C:\>MODE COM2 96,N,8,1
C:\>COPY CON COM2
C:\>FILES
C:\>COPY CON COM2
    KILL “DEMO.BAS”
    ^Z
C:\>FILES
```


● MOVE

Description

This command is used to move downloaded files to the FLASH memory.

Syntax

MOVE

Parameter	Description
N/A	N/A

● UpdatBios

Description

This command is used to upgrade the printer firmware. To upgrade, the printer must be in Ready status before the command can be issued. When updating the firmware, the checksum will be calculated and compared with the value of the pre-calculated checksum stored at address FE00h. If the two values are different, the upgrade process is a failure and the SYNTAX ERROR message is printed. The checksum is the summation of firmware data of 2000h~EFFFH and 10000h~2FFFFh. After upgrading the firmware, the checksum of the flash memory will be calculated. If it is not equal to the checksum of address FE00h, the flash memory will be upgraded continuously until the two values are equivalent.

Syntax

UpdatBios (space) (Firmware Total Page) (Firmware Data).

Parameter	**Description**
(Space)	Space character (hex: 20h)
Firmware Total Page	Indicated in hexademical system digits; one page is 64K bytes (For example, total firmware data are 320KB, or 05h)
Firmware Data	Binary code of the firmware program

Example

UpdatBios ¡ 05 (Firmware Data)

BASIC Commands and Functions

● ABS( )

Description

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

Syntax

ABS (-100)
ABS (-99.99)
ABS (VARIABLE)

Example

```
DOWNLOAD “TEST.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
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$(ABS(C))
PRINT 1
EOP
```


● ASC( )

Description

This function returns the ASCII code of the character.

Syntax

ASC (“A”)

Example

```
DOWNLOAD “TEST.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
CODE1=ASC(“A”)
TEXT 100,100,”3”,0,1,1,STR$(CODE1)
PRINT 1
EOP
```


● CHR$( )

Description

This function returns the character that has the specified ASCII code.

Syntax

CHR$(n)

Parameter	Description
n	The ASCII code

Example

```
DOWNLOAD “TEST.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=65
WORD$=CHR$(A)
TEXT 100,100,”3”,0,1,1,WORD$
PRINT 1
EOP
```


● END

Description

This command stops the execution of program.

Syntax

END

Example

```
DOWNLOAD “TEST.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=1
:START
IF A>10 THEN END ELSE A=A+1
GOTO START
TEXT 100,100,”3”,0,1,1,STR$(A)
PRINT 1
EOP
```


● EOF( )

Description

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

Syntax

EOF (File Handle)

Parameter	**Description**
File handle	Either 0 or 1.

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

Example

```
DOWNLOAD “DEMO.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA",0
SEEK 0,4
:A
ITEM$=""
READ 0,ITEM$,PRICE,QTY
TEXT  10,10,"3",0,1,1,"SEEK  COMMAND  TEST  (SHIFT  4
CHARACTERS)"
BOX 98,48,502,514,2
A$="ITEMS:"+ITEM$
```



```
B$="PRICE:"+STR$(PRICE)
C$="QTY:"+STR$(QTY)
TEXT 128,114,"2",0,1,1,A$
TEXT 130,198,"2",0,1,1,B$
TEXT 132,268,"2",0,1,1,C$
BARCODE 132,365,"39",96,1,0,2,4,"PRICE-2000"
PRINT 1
I=EOF(0)
IF I=0 THEN GOTO A
PRINT 1
EOP
```


● OPEN

Description

This command is used to open a downloaded file and establish the file handle. Up to 2 files can be opened at the same time. The file to be opened should be downloaded prior to using this command.

Syntax

OPEN "Filename", File handle

Parameter	**Description**
Filename	The file downloaded in the printer memory
File handle	Either 0 or 1.

Example

If a file by the name of "DATA ¡ ¡s to be downloaded,
The file format contains:

```
DOWNLOAD  ¡ DATA",20,Computer<CR>
20000<CR>
15<CR>
```

Saving the above contents of data under the file name of "DATA". Follow the steps below to download data to the printer

<under MS-DOS mode>:
C:\>MODE COM2:96,N,8,1 ↵
C:\>COPY DATA /B COM2 ↵

The above example sets the following: baud rate at 9600 bps, no parity, 8 bits data, 1 top bit. If a file by name of "DEMO.BAS ¡ ¡s to be downloaded, the file format contains:

DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
CLS


```
SPEED 2
DENSITY 8
SET CUTTER OFF
SET PEEL OFF
DIRECTIO 0
REFERENCE 0,0
OPEN “DATA”,1
SEEK 1,0
READ 1,ITEM$,PRICE,QTY
I=EOF(1)
IF I>0 THEN END
BOX 98,48,502,514,2
A$=”ITEMS:”+ITEM$
B$=”PRICE:”+STR$(PRICE)
C$=”QTY:”+STR$(QTY)
TEXT 128,114,”2”,0,1,1,A$
TEXT 130,198,”2”,0,1,1,B$
TEXT 132,268,”2”,0,1,1,C$
BARCODE 132,365,”39”,96,1,0,2,4,”PRICE-2000”
PRINT 1
CLOSE 1
EOP
```

Saving the above contents of data under the file name of **“**DEMO**”.**
Follow the steps below to download data to the printer
<under MS-DOS mode>:
C:\>MODE COM1:96,N,8,1 ↵
C:\>COPY DEMO COM1 ↵

The above example sets the following: baud rate at 9600 bps, no parity, 8 bits data, 1 stop bit. Saving the following command of program under the file name of “EXECUTE”: DEMO
C:\>COPY EXECUTE COM1↵
The above example instructs the printer to open the file “DATA” with a file handle of 1 and read items from the file.

● READ

Description

This command is used to read data from downloaded data file

Syntax

READ file handle, variables

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

Example

```
DOWNLOAD “OPEN1.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA",0
OPEN "DATA1",1
SEEK 0,4
:A
ITEM$=""
READ 0,ITEM$,PRICE,QTY
TEXT  10,10,"3",0,1,1,"SEEK  COMMAND  TEST  (SHIFT  4
CHARACTERS)"
BOX 98,48,502,514,2
A$="ITEMS:"+ITEM$
B$="PRICE:"+STR$(PRICE)
C$="QTY:"+STR$(QTY)
TEXT 128,114,"2",0,1,1,A$
```



```
TEXT 130,198,"2",0,1,1,B$
TEXT 132,268,"2",0,1,1,C$
BARCODE 132,365,"39",96,1,0,2,4,"PRICE-2000"
PRINT 1
I=EOF(0)
IF I=0 THEN GOTO A
SEEK 1,0
:B
READ 1,ITEM$,PRICE,QTY
TEXT 10,10,"4",0,1,1,"OPEN, READ, EOF() COMMAND TEST"
BOX 98,48,502,514,2
A$="ITEMS:"+ITEM$
B$="PRICE:"+STR$(PRICE)
C$="QTY:"+STR$(QTY)
TEXT 128,114,"2",0,1,1,A$
TEXT 130,198,"2",0,1,1,B$
TEXT 132,268,"2",0,1,1,C$
BARCODE 132,365,"39",96,1,0,2,4,"PRICE-2000"
PRINT 1
I=EOF(1)
IF I=0 THEN GOTO B
OPEN "DATA2",0
CLS
Z$=""
Z$=FREAD$(0,6)
TEXT 10,20,"4",0,1,1,"FREAD$() FUNCTION TEST"
TEXT 10,70,"4",0,1,1,"ITEM3$= "+Z$
J=LOF("DATA2")
TEXT 10,140,"3",0,1,1,"THE FILE SIZE OF DATA2 IS:
"+STR$(J)+" Bytes"
PRINT 1
EOP
```


● SEEK

Description

This command is used to 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 the beginning of a new position

Example

```
DOWNLOAD “TEST.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA",0
SEEK 0,4
ITEM$=""
READ 0,ITEM$,PRICE,QTY
TEXT 10,10,"3",0,1,1,"SEEK COMMAND TEST (SHIFT 4
CHARACTERS)"
BOX 98,48,502,514,2
A$="ITEMS:"+ITEM$
B$="PRICE:"+STR$(PRICE)
C$="QTY:"+STR$(QTY)
TEXT 128,114,"2",0,1,1,A$
TEXT 130,198,"2",0,1,1,B$
```



```
TEXT 132,268,"2",0,1,1,C$
BARCODE 132,365,"39",96,1,0,2,4,"PRICE-2000"
PRINT 1
EOP
```


● LOF( )

Description

This function returns the size of the specified file.

Syntax

LOF (“FILENAME”)

Parameter	Description
FILENAME	The file downloaded in the printer memory.

Example

```
DOWNLOAD “OPEN2.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA2",0
CLS
Z$=""
Z$=FREAD$(0,6)
TEXT 10,20,"4",0,1,1,"FREAD$() FUNCTION TEST"
TEXT 10,70,"4",0,1,1,"ITEM3$= "+Z$
J=LOF("DATA2")
TEXT 10,140,"3",0,1,1,"THE FILE SIZE OF DATA2 IS:
"+STR$(J)+" Bytes"
PRINT 1
EOP
```


● FREAD$( )

Description

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

Syntax

FREAD$ (file handle, byte)

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

Example

```
DOWNLOAD “FREAD.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA2",0
CLS
Z$=""
Z$=FREAD$(0,6)
TEXT 10,20,"4",0,1,1,"FREAD$() FUNCTION TEST"
TEXT 10,70,"4",0,1,1,"ITEM3$= "+Z$
J=LOF("DATA2")
TEXT 10,140,"3",0,1,1,"THE FILE SIZE OF DATA2 IS:
"+STR$(J)+" Bytes"
PRINT 1
EOP
```


● FOR..NEXT LOOP

Description

Loop is used to execute one or more lines of program repetitively. Before anything, a value should be assigned the loop counter to specify the execution times. Nested loop is allowed (up to 10 nested loops) in this printer. Jumping out in the middle of the FOR..NEXT loop is prohibited because it is not a good programming skill..

Syntax

For variable = start TO end STEP increment
    statement
NEXT

Parameter	**Description**
variable	The variable name is up to 8 characters
start	Can be integer or floating point numbers
end	Can be integer of floating point numbers
increment	Integer or floating point, positive or negative.

Example

```
DOWNLOAD “FREAD.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A$=""
B$=""
C$=""
H$=""
FOR I=1 TO 10 STEP 1
```



```
A$=A$+STR$(I)+" "
TEXT 10,10,"3",0,1,1,A$
NEXT

FOR I=1 TO 1000 STEP 100
B$=B$+STR$(I)+" "
TEXT 10,50,"3",0,1,1,B$
NEXT

FOR I=50 TO 10 STEP -10
C$=C$+STR$(I)+" "
TEXT 10,100,"3",0,1,1,C$
NEXT

FOR I=1 TO 5 STEP 0.5
H$=H$+STR$(I)+" "
TEXT 10,150,"3",0,1,1,H$
NEXT

PRINT 1
EOP
```


● IF..THEN..ELSE

Description

Use the IF..THEN..ELSE to execute programs conditionally.

Syntax

IF condition THEN statement [ELSE statement]

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

Parameter	**Description**
condition	Available relational operator: <, >, =, <=, >=
statement	Only one statement is available in IF..THEN..ELSE

Example

```
DOWNLOAD “IFTHEN.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=50
B=5
C$=""
D$=""

:L1
IF A>100 THEN GOTO L1 ELSE A=A+10
C$=STR$(A)+" IS SMALLER THAN 100"
TEXT 100,10,"4",0,1,1,C$
PRINT 1
```



```
END

:L2
A=A+B
D$=STR$(A)+" IS LARGER THAN 100"
TEXT 100,100,"4",0,1,1,D$
PRINT 1
GOTO L1
EOP
```


● GOSUB…RETURN

Description

Branch to and return from a subroutine. Branch to the specified label and execute subroutines until “RETURN” is reached and then go back to the statement following the GOSUB statement.

Syntax

GOSUB LABEL
    statement

END

:LABEL
    statement

RETURN

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

Example

```
DOWNLOAD “GOSUB1.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,600,"5",0,1,1,"GOSUB & RETURN COMMAND TEST"
X=300
Y=300
```



```
GOSUB DASH
GOSUB DR_LINE
PRINT 1
END

:DR_LINE
BOX X,Y,X+200,Y+200,5
RETURN

:DASH
SET PEEL OFF
DENSITY 13
FOR I=21 TO 50
TEXT 10,I,100,"3",0,1,1,"============================="
NEXT
RETURN

EOP
```


● GOTO

Description

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

Syntax

GOTO LABEL

:LABEL

Parameter	Description
N/A	N/A

Example

```
DOWNLOAD “GOSUB1.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=0
TOTAL=0

:START
IF A<100 THEN GOTO SUM ELSE GOTO PRTOUT
: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
```


● INP$( )

Description

One byte is received from a serial port through this function.

Syntax

INP$(n)

Parameter	Description
n	1: com1 port in printer

Example

```
DOWNLOAD “GOSUB1.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
PRICE$=”123456”
T$=INP$(1)
TEXT 100,100,”4”,0,1,1,T$
PRINT1
EOP
```


● INPUT

Description

This command is used to receive data through serial port. This command is used with TSC portable keyboard KP-200.

Syntax

INPUT [“Prompt string”], variables

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

Example

```
DOWNLOAD “INPUT1.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
:START
CLS
A$=""
TEXT 20,50,"3",0,1,1,"Please connect LCD keyboard for
testing"
INPUT "Enter the code number"; A$
BARCODE 20,100,"39",48,1,0,2,5,A$
PRINT 1
```



```
GOTO START
EOP
```


● REM

Description

Comment. Anything beginning with “REM” is ignored by the printer.

Syntax

REM

Example

```
REM ********************************
REM This is a demonstration program*
REM ********************************
DOWNLOAD “REMARK.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”REMARK DEMO PROGRAM”
EOP
```


● OUT

Description

This command is used to send data through printer serial port.

Syntax

OUT “prompt”, variable

Parameter	Description
prompt	Prompt which is shown on LCD screen.
Variable	The output message

Example

```
DOWNLOAD “INPUT1.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
PRICE$=”123456”
OUT “PRICE:”,PRICE$
EOP
```


● GETKEY( )

Description

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

Syntax

GETKEY()

Example

```
DOWNLOAD “DEMO4.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
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
```


● INT( )

Description

This function is used to truncate a floating point number.

Syntax

INT (n)

Parameter	Description
n	n can be positive or negative integer, floating point number or mathematical expression.

Example

```
DOWNLOAD “DEMO5.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=INT(99.99)
B=INT(-199.89)
C=INT(80)
TEXT 50,100,”3”,0,1,1,”INT(99.99)”+STR$(A)
TEXT 50,150,”3”,0,1,1,”INT(-199.89)”+STR$(B)
TEXT 50,200,”3”,0,1,1,”INT(80)”+STR$(C)
PRINT 1
EOP
```


● LEFT$( )

Description

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

Syntax

LEFT$ (X$, n)

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

Example

```
DOWNLOAD "STR1.BAS"
SIZE 4.00,4.00
GAP 0.12,0.00
SPEED 2.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
SET DEBUG LABEL
CLS
A$="TAIWAN SEMICONDUCTOR CO., LTD"
C$=LEFT$(A$,10)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,100,"3",0,1,1,"10 LEFT CHARS "+C$
PRINT 1
EOP
```


● LEN( )

Description

This function returns the length of a string.

Syntax

LEN (string)

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

Example

```
DOWNLOAD "DEMO6.BAS"
SIZE 4.00,4.00
GAP 0.12,0.00
SPEED 2.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
SET DEBUG LABEL
CLS
A$="TAIWAN SEMICONDUCTOR CO., LTD"
B=LEN(A$)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,50,"3",0,1,1,"STRING LENGTH="+STR$(B)
PRINT 1
EOP
```


● MID$( )

Description

This function is used to get the specified number of characters down from the mth character of a string.

Syntax

MID$(string, m, n)

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

Example

```
DOWNLOAD "DEMO7.BAS"
SIZE 4.00,4.00
GAP 0.12,0.00
SPEED 2.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
SET DEBUG LABEL
CLS
A$="TAIWAN SEMICONDUCTOR CO., LTD"
E$=MID$(A$,11,10)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,200,"3",0,1,1,"10 MIDDLE CHARS "+E$
PRINT 1
EOP
```


● RIGHT$( )

Description

This function returns the 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

```
DOWNLOAD "DEMO8.BAS"
SIZE 4.00,4.00
GAP 0.12,0.00
SPEED 2.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
SET DEBUG LABEL
CLS
A$="TAIWAN SEMICONDUCTOR CO., LTD"
D$=RIGHT$(A$,10)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,150,"3",0,1,1,"10 RIGHT CHARS "+D$
PRINT 1
EOP
```


● STR$( )

Description

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

Syntax

STR$ (n)

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

Example

```
DOWNLOAD "DEMO9.BAS"
SIZE 4.00,4.00
GAP 0.12,0.00
SPEED 2.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
SET DEBUG LABEL
CLS
A$="TAIWAN SEMICONDUCTOR CO., LTD"
F$="100"
G$="500"
H=VAL(F$)+VAL(G$)
I$=STR$(H)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,250,"3",0,1,1,"STR$() FUNCTION TEST "+I$
PRINT 1
EOP
```


● VAL( )

Description

This function is used to convert numeric character into corresponding integer or floating point number.

Syntax

VAL (“numeric character”)

Parameter	**Description**
numeric character	“0~9”, “.”

Example

```
DOWNLOAD "DEMO10.BAS"
SIZE 4.00,4.00
GAP 0.12,0.00
SPEED 2.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
SET DEBUG LABEL
CLS
A$="TAIWAN SEMICONDUCTOR CO., LTD"
F$="100"
G$="500"
H=VAL(F$)+VAL(G$)
I$=STR$(H)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,250,"3",0,1,1,"VAL FUNCTION TEST= "+I$
PRINT 1
EOP
```


● BEEP

Description

This command is used to issue a beep sound on portable keyboard (KP-200). Printer sends 0x07 to portable keyboard.

Syntax

BEEP

Parameter	Description
N/A	N/A

Example

```
DOWNLOAD “DEMO11.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
:START
CLS
A$=""
TEXT 20,50,"3",0,1,1,"Please connect LCD keyboard for
testing"
BEEP
INPUT "Enter the code number"; A$
BARCODE 20,100,"39",48,1,0,2,5,A$
PRINT 1
GOTO START
EOP
```


Device Reconfiguration Commands

● SET COUNTER

Description

This setting sets the counter number in program and their increments. Counter does not support mathematical operation.

Syntax

SET COUNTER @n step

Parameter	Description
@n	n: counter number. There are 50 counters available (0~49) in the printer for text and barcode.
step	The increment of the counter, can be positive or negative. -999999999<= step <=999999999

Example

```
DOWNLOAD “DEMO13.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET COUNTER @0 1
SET COUNTER @1 5
CLS
@1=”00001”
@2=”TSC00001”
TEXT 50,50,”3”,0,1,1,@1
BARCODE 50,500,”39”,48,1,0,2,4,@2
PRINT 1
EOP
```


● SET CUTTER

Description

This setting is used to activate/deactivate the cutter and define how many printed labels to be cut at one time.

Syntax

SET CUTTER OFF/BATCH/pieces

Parameter	**Description**
OFF	Disable cutter function.
BATCH	Set printer to cut label per set of printing. Please refer to PRINT command
pieces	Set number of printing labels per cut. 0<= pieces <=127

Example

```
REM SET CUTTER FUNCTION OFF EXAMPLE PROGRAM
DOWNLOAD “DEMO14.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”DEMO14”
BARCODE 50,500,”39”,48,1,0,2,4,”DEMO14”
PRINT 1
EOP

REM SET CUTTER BATCH EXAMPLE PROGRAM
REM This program cuts 3 times (3 set)
```



```
DOWNLOAD “DEMO15.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER BATCH
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”DEMO14”
BARCODE 50,500,”39”,48,1,0,2,4,”DEMO14”
PRINT 3,2
EOP

REM SET CUTTER BATCH EXAMPLE PROGRAM
REM This program cuts each printed label
DOWNLOAD “DEMO16.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”DEMO14”
BARCODE 50,500,”39”,48,1,0,2,4,”DEMO14”
PRINT 3,2
EOP
```


● SET KEY1, SET KEY2

Description

This setting is used to enable/disable the KEY1/ KEY2 function. The default function of KEY1 is pause and KEY2 feed. Before setting KEY1/ KEY2 function otherwise, please disable KEY1/ KEY2 first. The settings will remain resident in the printer even when the printer is power off.

Syntax

SET KYE1 ON /OFF
SET KEY2 ON /OFF

Parameter	**Description**
ON	Enable KEY1 as PAUSE function
	Enable KEY2 as FEED function
OFF	Disable KEY1 as PAUSE function
	Disable KEY2 as FEED function

Note: The settings will remain in the printer even if the printer is power off.

Example

```
DOWNLOAD “DEMO17.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL OFF
SET KEY1 OFF
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
GOTO START
EOP
```


● SET LED1, LED2, LED3

Description

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

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

LED no.	**Default Function**
LED1	Power on/off
LED2	Printer on-line/off-line
LED3	Error/normal

Syntax

SET LED1 ON/OFF

SET LED2 ON/OFF

SET LED3 ON/OFF

Note: The setting will remain in the printer even if the printer is power off.

Example

The example below is operated under DOS environment.

C:\>MODE COM2 96,N,8,1

C:\>COPY CON COM2

SET LED3 OFF

^Z

Turn off the printer power and then on, without installing ribbon or label. Normally, the ERROR LED will flash. But it won't after this function is disabled.

● SET PEEL

Description

This setting is used to enable/disable the self-peeling function.

The default setting for this function is off. When this function is set on, the printer stops after each label printing, and does not print the next label until the peeled label is taken away.

Syntax

SET PEEL ON/OFF

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

Example

```
REM SELF-PEELING FUNCTION ON
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL ON
SET KEY1 OFF
CLS
TEXT 50,100,"3",0,1,1,"SELF-PEELING FUNCTION TEST"
PRINT 5
```


● SET DEBUG

Description

This setting is used to set the output of the error message. The default error message output is by printing on the label.

Syntax

SET DEBUG OFF/ LABEL/ RS232

Parameter	**Description**
OFF	Turn off error message output
LABEL	Output error message to label
RS232	Output error message through RS-232 port

Example

The example below tells printer to printthe error message on the label.

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL ON
SET KEY1 OFF
SET DEBUG LABEL
CLS
BARCODE 100,100,"39",48,1,0,2,5,"CODE 39”
PRINT 1
```


● SET GAP

Description

This setting is used to set the gap sensor light emission strength. The printer initiates automatic gap sensor calibration as you hold down the PAUSE key and then turn on the printer power. But this function may cease to work if the thickness of the backing paper and that of label with backing paper are not of appreciable difference to the sensor, or when there are printed marks or patterns on the label. In such case, users have to calibrate the gap sensor manually by this setting. This is a trial-and-error method to attain the proper setting.

Syntax

SET GAP n

Parameter	**Description**
n	The gap sensor light emission strength. Available values are 0 to15 for TTP-243, and 0 to 31 for TTP-242.

Example

The example below is operated in DOS environment.

```
C:\>MODE COM2 96,N,8,1
C:\>COPY CON COM2
    SET GAP 1
    ^Z
```

Press the FEED key to test. Does printer stop at the same position on each label without the error light turned on? If not, please adjust the setting again.

When trying with this setting, please begin from 0 and then on to higher values gradually.

● SET RIBBON

Description

This setting is used to enable/disable DC motor of the ribbon mechanism (Thermal Transfer Printing/Thermal Direct Printing)

.

Syntax

SET RIBBON ON /OFF

Parameter	**Description**
ON	Thermal transfer printing
OFF	Thermal direct printing

Example

```
REM Thermal direct printing
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL ON
SET KEY1 OFF
SET DEBUG LABEL
SET RIBBON OFF
CLS
BARCODE 100,100,"39",48,1,0,2,5,"CODE 39”
PRINT 1
```


● SET COM1

Description

This setting defines communication parameters for printer serial port.

Syntax

SET COM1 baud, parity, data, stop

Parameter	**Description**
baud	Baud rate, available baud rates are as listed : 24: 2400 bps 48: 4800 bps 96: 9600 bps 19: 19200 bps
parity	Parity check N: None 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

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL ON
SET KEY1 OFF
```



```
SET DEBUG LABEL
SET RIBBON OFF
SET COM1 96,N,8,1
CLS
BARCODE 100,100,"39",48,1,0,2,5,"CODE 39”
PRINT 1
```


Printer Global Variables

● @LABEL

Description

This variable is used to count how many pieces of labels have been printed. It won't be initialized if the printer is reset. It will be memorized if the printer power is turned off.

Syntax

Write attribute: @LABEL=n

Read attribute: A=@LABEL

Parameter	**Description**
n	number of labels printed 0<n<65535

Example

```
DOWNLOAD "DEMO20.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL ON
SET KEY1 OFF
SET DEBUG LABEL
SET RIBBON OFF
SET COM1 96,N,8,1
CLS
IF @LABEL=100 THEN @LABEL=0 ELSE TEXT
100 ,100 , "3",0,1,1,STR$(@LABEL)
PRINT 1
EOP
```


● PEEL

Description

This command is used to obtain the status of the peel-off sensor. Its attribute is read only.

Syntax

Return Value	Description
0	Set peel-off sensor off
1	Set peel-off sensor on

Example

```
DOWNLOAD “DEMO19.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL OFF
SET LED1 OFF
CLS
IF PEEL=1 THEN LED1=1
EOP
```


● LED1, LED2, LED3

Description

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

Syntax

LED m=n

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

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET LED3 OFF
CLS
LED3=1
```


● KEY1, KEY2

Description

This command is used to read the status of KEY1 and KEY2.

Syntax

Key	Return value
KEY1	0: released
	1: pressed
KEY2	0: released
	1: pressed

Note: This command is read only.

Example

The example below instructs printer to light on LED1 if the Pause key is pressed.

```
DOWNLOAD “DEMO18.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET LED1 OFF
CLS
IF KEY1=1 THEN LED1=1
EOP
```


● YEAR

Description

This variable is used to read from/write to RTC the year data. Two-digit (00~99) year format is supported by RTC. Also the leap year and year of 2000.

Syntax

Write attribute: YEAR=98

Read attribute: A=YEAR

Range: 00~99

Example

```
DOWNLOAD "DEMO21.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
IF YEAR<10 THEN YEAR$="200"+STR$(YEAR) ELSE
YEAR$="19"+STR$(YEAR)
IF MONTH<10 THEN MONTH$="0"+STR$(MONTH) ELSE
MONTH$=STR$(MONTH)
IF DATE<10 THEN DATE$="0"+STR$(DATE) ELSE
DATE$=STR$(DATE)
IF HOUR<10 THEN HOUR$="0"+STR$(HOUR) ELSE
HOUR$=STR$(HOUR)
IF MINUTE<10 THEN MINUTE$="0"+STR$(MINUTE) ELSE
MINUTE$=STR$(MINUTE)
WEEK$=STR$(WEEK)
A$=YEAR$+"/"+MONTH$+"/"+DATE$+"/"+WEEK$+"
```



```
"+HOUR$+":"+MINUTE$
TEXT 10,10,"5",0,1,1,A$
PRINT 1
EOP
```


● MONTH

Description

This variable is used to read from/write to RTC the month data. Two-digits (01~12) month format is supported by RTC.

Syntax

Write attribute: MONTH=01

Read attribute: A=MONTH

Range: 01~12

Example

```
DOWNLOAD "DEMO21.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
IF YEAR<10 THEN YEAR$="200"+STR$(YEAR) ELSE
YEAR$="19"+STR$(YEAR)
IF MONTH<10 THEN MONTH$="0"+STR$(MONTH) ELSE
MONTH$=STR$(MONTH)
IF DATE<10 THEN DATE$="0"+STR$(DATE) ELSE
DATE$=STR$(DATE)
IF HOUR<10 THEN HOUR$="0"+STR$(HOUR) ELSE
HOUR$=STR$(HOUR)
IF MINUTE<10 THEN MINUTE$="0"+STR$(MINUTE) ELSE
MINUTE$=STR$(MINUTE)
WEEK$=STR$(WEEK)
A$=YEAR$+"/"+MONTH$+"/"+DATE$+"/"+WEEK$+"
"+HOUR$+":"+MINUTE$
TEXT 10,10,"5",0,1,1,A$
```



```
PRINT 1
EOP
```


● DATE

Description

This variable is used to read from/write to RTC the date data. Two-digits (01~31) date format is supported by RTC.

Syntax

Write attribute: DATE=12

Read attribute: A=DATE

Range: 01~31

Example

```
DOWNLOAD "DEMO21.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
IF YEAR<10 THEN YEAR$="200"+STR$(YEAR) ELSE
YEAR$="19"+STR$(YEAR)
IF MONTH<10 THEN MONTH$="0"+STR$(MONTH) ELSE
MONTH$=STR$(MONTH)
IF DATE<10 THEN DATE$="0"+STR$(DATE) ELSE
DATE$=STR$(DATE)
IF HOUR<10 THEN HOUR$="0"+STR$(HOUR) ELSE
HOUR$=STR$(HOUR)
IF MINUTE<10 THEN MINUTE$="0"+STR$(MINUTE) ELSE
MINUTE$=STR$(MINUTE)
WEEK$=STR$(WEEK)
A$=YEAR$+"/"+MONTH$+"/"+DATE$+"/"+WEEK$+"
"+HOUR$+":"+MINUTE$
TEXT 10,10,"5",0,1,1,A$
```



```
PRINT 1
EOP
```


● WEEK

Description

This variable is used to read from/write to RTC the week data, which is represented by one single digit (1~7).. .

Syntax

Write attribute: WEEK=3

Read attribute: A=WEEK

Range:

For TTP-242: 0 (Sunday)~6 (Saturday)

For TTP-243: 1(Sunday)~7(Saturday)

Example

```
DOWNLOAD "DEMO21.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
IF YEAR<10 THEN YEAR$="200"+STR$(YEAR) ELSE
YEAR$="19"+STR$(YEAR)
IF MONTH<10 THEN MONTH$="0"+STR$(MONTH) ELSE
MONTH$=STR$(MONTH)
IF DATE<10 THEN DATE$="0"+STR$(DATE) ELSE
DATE$=STR$(DATE)
IF HOUR<10 THEN HOUR$="0"+STR$(HOUR) ELSE
HOUR$=STR$(HOUR)
IF MINUTE<10 THEN MINUTE$="0"+STR$(MINUTE) ELSE
MINUTE$=STR$(MINUTE)
WEEK$=STR$(WEEK)
```



```
A$=YEAR$+"/"+MONTH$+"/"+DATE$+"/"+WEEK$+"
"+HOUR$+":"+MINUTE$
TEXT 10,10,"5",0,1,1,A$
PRINT 1
EOP
```


● HOUR

Description

This variable is used to read from/write to RTC the hour data. The 24-hour-day system (00~23) is supported by RTC.

Syntax

Write attribute: HOUR=12

Read attribute: A=HOUR

Range: 00~23

Example

```
DOWNLOAD "DEMO21.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
IF YEAR<10 THEN YEAR$="200"+STR$(YEAR) ELSE
YEAR$="19"+STR$(YEAR)
IF MONTH<10 THEN MONTH$="0"+STR$(MONTH) ELSE
MONTH$=STR$(MONTH)
IF DATE<10 THEN DATE$="0"+STR$(DATE) ELSE
DATE$=STR$(DATE)
IF HOUR<10 THEN HOUR$="0"+STR$(HOUR) ELSE
HOUR$=STR$(HOUR)
IF MINUTE<10 THEN MINUTE$="0"+STR$(MINUTE) ELSE
MINUTE$=STR$(MINUTE)
WEEK$=STR$(WEEK)
A$=YEAR$+"/"+MONTH$+"/"+DATE$+"/"+WEEK$+"
"+HOUR$+":"+MINUTE$
TEXT 10,10,"5",0,1,1,A$
PRINT 1
EOP
```


● MINUTE

Description

This variable is used to read from/write to RTC the minute data. Two-digits (00~59) minute format is supported by RTC.

Syntax

Write attribute: MINUTE=12

Read attribute: A=MINUTE

Range: 00~59

Example

```
DOWNLOAD "DEMO21.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
IF YEAR<10 THEN YEAR$="200"+STR$(YEAR) ELSE
YEAR$="19"+STR$(YEAR)
IF MONTH<10 THEN MONTH$="0"+STR$(MONTH) ELSE
MONTH$=STR$(MONTH)
IF DATE<10 THEN DATE$="0"+STR$(DATE) ELSE
DATE$=STR$(DATE)
IF HOUR<10 THEN HOUR$="0"+STR$(HOUR) ELSE
HOUR$=STR$(HOUR)
IF MINUTE<10 THEN MINUTE$="0"+STR$(MINUTE) ELSE
MINUTE$=STR$(MINUTE)
WEEK$=STR$(WEEK)
A$=YEAR$+"/"+MONTH$+"/"+DATE$+"/"+WEEK$+"
"+HOUR$+":"+MINUTE$
```



```
TEXT 10,10,"5",0,1,1,A$
PRINT 1
EOP
```


SECOND

Description

This variable is used to read from/write to RTC the second data. Two-digits (00~59) second format is supported by RTC.

Syntax

Write attribute: SECOND=12

Read attribute: A=SECOND

Range: 00~59

Example

```
DOWNLOAD "DEMO21.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
IF YEAR<10 THEN YEAR$="200"+STR$(YEAR) ELSE
YEAR$="19"+STR$(YEAR)
IF MONTH<10 THEN MONTH$="0"+STR$(MONTH) ELSE
MONTH$=STR$(MONTH)
IF DATE<10 THEN DATE$="0"+STR$(DATE) ELSE
DATE$=STR$(DATE)
IF HOUR<10 THEN HOUR$="0"+STR$(HOUR) ELSE
HOUR$=STR$(HOUR)
IF MINUTE<10 THEN MINUTE$="0"+STR$(MINUTE) ELSE
MINUTE$=STR$(MINUTE)
IF SECOND<10 THEN SECOND$="0"+STR$(SECOND) ELSE
SECOND$=STR$(SECOND)
WEEK$=STR$(WEEK)
```



```
A$=YEAR$+"/"+MONTH$+"/"+DATE$+"/"+WEEK$+"
"+HOUR$+":"+MINUTE$+":"+SECOND$
TEXT 10,10,"5",0,1,1,A$
PRINT 1
EOP
```


TAIWAN SEMICONDUCTOR CO., LTD.

2F, No.8, Alley 16, Lane 235, Pao, Chiao Rd., Hsin Tein, Taipei Hsien, Taiwan

TEL: +886-2-29174145 FAX: +886-2-29159741

E-mail: printer@mail2.ts.com.tw Web Site: http://www.ts.com.tw