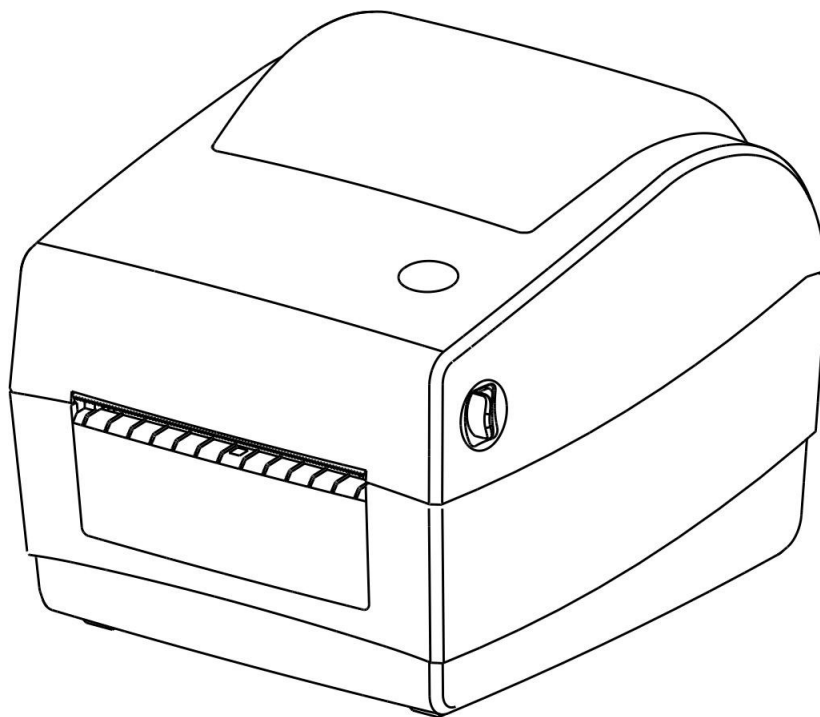




**SX-TLP-410**

# **Label Printer Command Set**



## Table of Contents

|  |           |
|--|-----------|
| <b>Document Conventions .....</b>            | <b>4</b>  |
| <b>Setup and System Commands .....</b>       | <b>4</b>  |
| SIZE .....                                   | 4         |
| GAP .....                                    | 5         |
| BLINE .....                                  | 6         |
| OFFSET .....                                 | 8         |
| SPEED .....                                  | 9         |
| DENSITY .....                                | 9         |
| DIRECTION .....                              | 10        |
| REFERENCE .....                              | 10        |
| CODEPAGE .....                               | 11        |
| CLS .....                                    | 12        |
| FEED .....                                   | 12        |
| BACKFEED .....                               | 13        |
| FORMFEED .....                               | 13        |
| HOME .....                                   | 14        |
| PRINT .....                                  | 14        |
| SOUND .....                                  | 15        |
| LIMITFEED .....                              | 16        |
| SELFTEST .....                               | 17        |
| <b>Label Formatting Commands .....</b>       | <b>17</b> |
| BAR .....                                    | 17        |
| BARCODE .....                                | 18        |
| BITMAP .....                                 | 22        |
| BOX .....                                    | 23        |
| ERASE .....                                  | 24        |
| PUTBMP .....                                 | 25        |
| PUTPCX .....                                 | 26        |
| QRCODE .....                                 | 27        |
| REVERSE .....                                | 30        |
| TEXT .....                                   | 30        |
| <b>Status Polling Commands(RS-232) .....</b> | <b>32</b> |
| <ESC>!? .....                                | 32        |
| <ESC>IR .....                                | 33        |

|   |           |
|---|-----------|
| ~!@ .....                                 | 33        |
| ~!A .....                                 | 33        |
| ~!C .....                                 | 34        |
| ~!D .....                                 | 34        |
| ~!F .....                                 | 34        |
| ~!I .....                                 | 35        |
| ~!T .....                                 | 35        |
| <b>FileManagement Commands.....</b>       | <b>35</b> |
| DOWNLOAD.....                             | 35        |
| EOP .....                                 | 38        |
| FILES.....                                | 39        |
| KILL .....                                | 39        |
| MOVE .....                                | 40        |
| RUN .....                                 | 40        |
| <b>BASIC Commands and Functions .....</b> | <b>41</b> |
| ABS( ) .....                              | 41        |
| ASC( ) .....                              | 41        |
| CHR\$( ) .....                            | 42        |
| END .....                                 | 42        |
| EOF( ) .....                              | 43        |
| OPEN.....                                 | 44        |
| READ.....                                 | 45        |
| SEEK .....                                | 46        |
| LOF( ) .....                              | 47        |
| FREAD\$( ) .....                          | 47        |
| FOR...NEXT .....                          | 48        |
| IF...THEN...ELSE...ENDIF .....            | 49        |
| GOSUB...RETURN .....                      | 52        |
| GOTO .....                                | 53        |
| REM.....                                  | 54        |
| INT( ) .....                              | 54        |
| LEFT\$( ) .....                           | 55        |
| IFN( ) .....                              | 55        |

|  |           |
|--|-----------|
| MID\$( ) .....                               | 56        |
| RIGHT\$( ) .....                             | 57        |
| STR\$( ) .....                               | 57        |
| VAL( ) .....                                 | 58        |
| BEEP .....                                   | 59        |
| <b>Device Reconfiguration Commands .....</b> | <b>59</b> |
| SET COUNTER .....                            | 59        |
| SET KEY1, SET KEY2 .....                     | 60        |
| SET PEEL .....                               | 61        |
| SET TEAR .....                               | 62        |
| SET HEAD .....                               | 63        |
| SET RIBBON .....                             | 63        |
| SET COM1 .....                               | 64        |
| SET PRINTKEY .....                           | 65        |
| SET REPRINT .....                            | 66        |
| PEEL .....                                   | 67        |
| KEY1, KEY2 .....                             | 68        |
| <b>Printer Global Variables .....</b>        | <b>68</b> |
| @LABEL .....                                 | 68        |

# Document Conventions

This manual uses the following typographic conventions.

| Convention        | Description  |
|-------------------|--|
| [expression list] | Items inside square brackets are optional, expression maximum length 2*1024bytes.                            |
| <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. |

Note: **200 DPI: 1 mm = 12 dots** The font in bold and italic type is used for note.

DOWNLOAD "TEST.BAS"

SET COUNTER @1 1

@1="0001"

TEXT 10,10,"3",0,1,1,@1

PRINT 3,2

EOP

## Setup and System Commands

### SIZE

|               |   |
|---------------|---|
| [Description] | This command defines the label width and length.                                  |
| [Syntax]      | (1)English system (inch)<br>SIZE m, n<br>(2)Metric system (mm)<br>SIZE m mm, n mm |

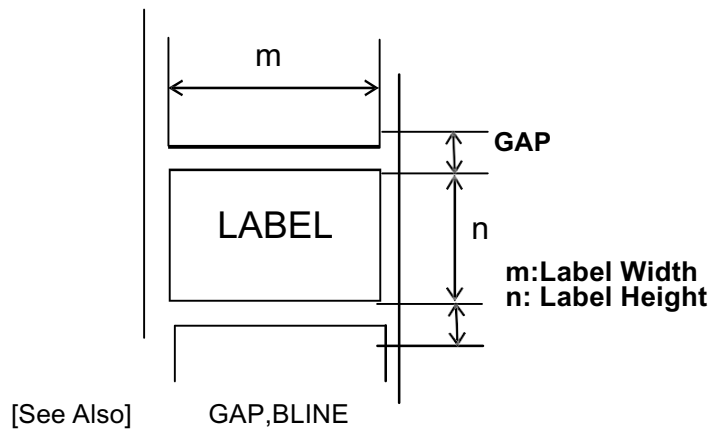
| Parameter | Description                   |
|-----------|-------------------------------|
| m         | Label width (inch or mm/dot)  |
| n         | Label length (inch or mm/dot) |

Notes: 200 DPI: 1 mm = 8 dots

300 DPI: 1mm = 12 dots

For metric and dot systems, there must be a space between parameter and "mm" or "dot".

|           |  |
|-----------|--|
| [Example] | (1) English system (inch)<br>SIZE 3.5, 3.00<br>(2) Metric system (mm)<br>SIZE 100 mm, 100 mm |
|-----------|--|



## 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<br>0 m 1(inch), 0 m 25.4 (mm) |
| n         | The offset distance of the gap<br>n label length (inch or mm)     |
| 0,0       | Continuous label  |

Notes:

For metric system, there must be a space between parameter and mm. When the sensor type is changed from "Black Mark" to "GAP", please send the "GAP" command to the printer first.

### Example: IN DOS mode

C:\>COPY CON LPT1 <Enter>

GAP 2 mm,0<Enter>

<Ctrl>+<Z><Enter>

[Example] **Normal gap**

(1).English system (inch)

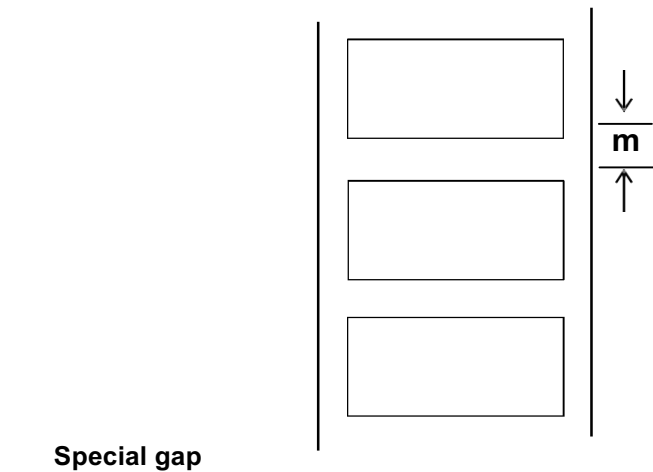
GAP 0.12,0

(2).Metric system (mm)

GAP 3 mm,0

(3).Continuous label

GAP 0,0

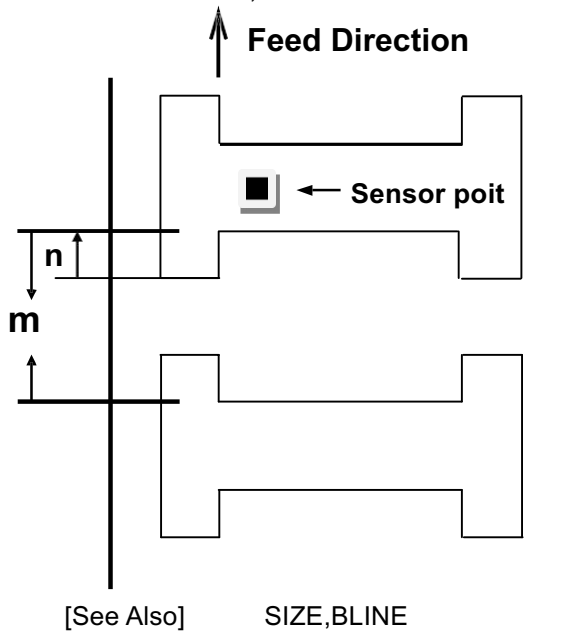


(1).English system (inch)

GAP 0.30,0.10

(2).Metric system (mm)

GAP 7.62 mm, 2.54 mm



## BLINE

[Description]      This command sets 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<br>$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) |
| n         | The extra label feeding length<br>$0 \leq n \leq \text{label length}$                                |
| 0,0       | Continuous label   |

**Note:** For metric system, there must be a space between parameter and mm.

When the sensor type is changed from "GAP" to "Black Mark", please send the "BLINE" command to the printer first.

### Example: IN DOS mode

C:\>COPY CON LPT1 <Enter>

BLINE 2 mm,0<Enter>

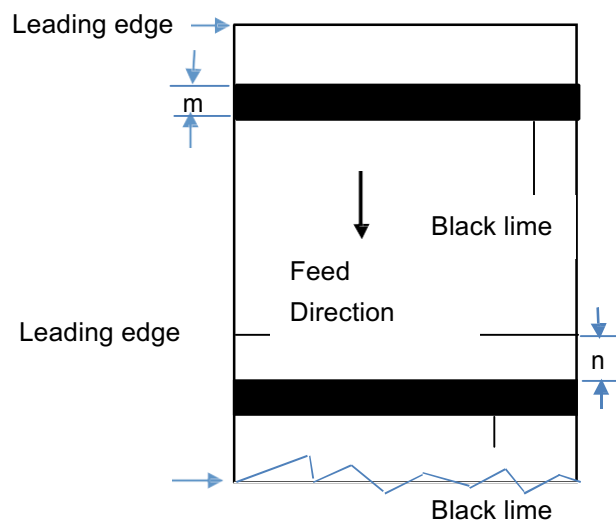
<Ctrl>+<Z><Enter>

[Example] (1) English system (inch)

BLINE 0.20,0.50

(2) Metric system (mm)

BLINE 5.08 mm,12.7 mm



[See Also] SIZE, GAP



# OFFSET

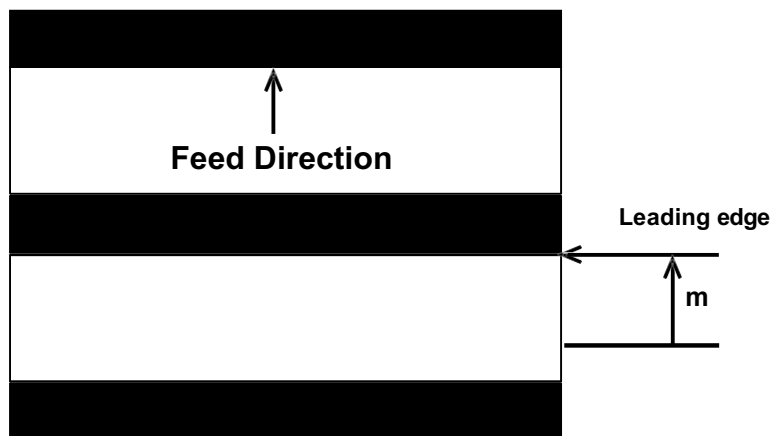
[Description] This command defines the selective, extra label feeding length each form feed takes, which, especially in peel-off mode and cutter mode, is used to adjust label stop position, so as for label to register at proper places for the intended purposes. The printer back tracks the extra feeding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

[Syntax] (1)English system (inch)  
 OFFSET m  
 (2)Metric system (mm)  
 OFFSET m mm

| Parameter | Description                                      |
|-----------|--|
| m         | The offset distance (inch or mm)<br>-1 m 1(inch) |

**CAUTION:** Improprity offset value may cause paperjam.

[Example] (1) English system (inch)  
 OFFSET 0.5  
 (2)Metric system (mm)  
 FFSET 12.7(mm)



[See Also] SIZE, GAP, SET PEEL

## SPEED

[Description] This command defines the printspeed.

[Syntax] SPEED n

| Parameter | Description                       |
|-----------|-----------------------------------|
| n         | printing speed in inch per second |

[Example] SPEED 4

[See Also] DENSITY

## DENSITY

[Description] This command sets the printing darkness.

[Syntax] DENSITY n

| Parameter | Description  |
|-----------|--|
| n         | 0~15<br>0, specifies the lightest level<br>15, specifies the darkest level |

[Example] DENSITY7

[See Also] SPEED

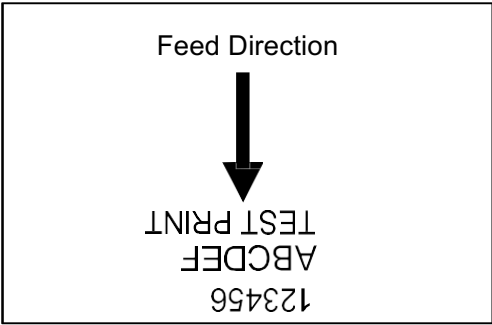
## DIRECTION

[Description] This command defines the printout direction. This will be stored in the printer memory.

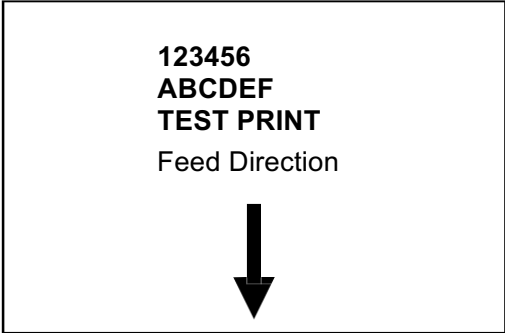
[Syntax] DIRECTION n

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

DIRECTION 0, 0

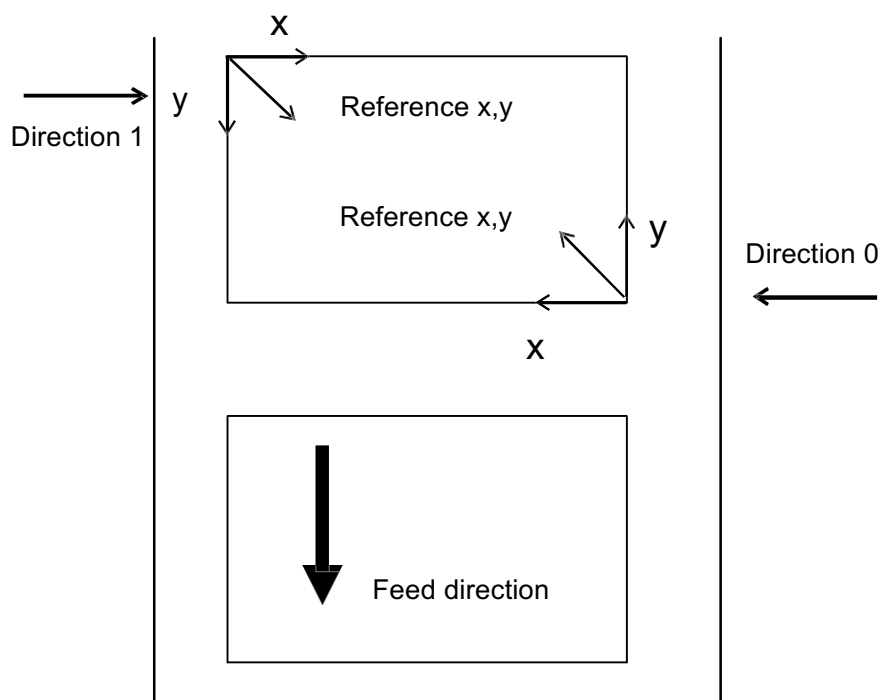


DIRECTION 1, 0



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 (in dots) |
| y         | Vertical coordinate (in dots).  |

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

[See Also]      DIRECTION

## 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 8-bit code page. |

|                      |
|----------------------|
| 437: United States   |
| 850: Multilingual    |
| 852: Slavic          |
| 860: Portuguese      |
| 863: Canadian/French |
| 865: Nordic          |

|                          |
|--------------------------|
| <b>Windows code page</b> |
| 1250: Central Europe     |
| 1252: LatinI             |
| 1253: Greek              |
| 1254: Turkish            |

[Example]            CODEPAGE 437

## CLS

[Description]      This command clears the image buffer.

[Syntax]            CLS

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

[Example]            CLS

[See Also]            SIZE, GAP, BLINE

## FEED

[Description]      This command feeds label with the specified length.

[Syntax]            FEED n

| Parameter | Description |
|-----------|-------------|
|-----------|-------------|

|   |           |
|---|-----------|
| n | Unit: dot |
| 1 | n 9999    |

[Example] FEED 40

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

[See Also] BACKFEED, SIZE, GAP, BLINE, HOME, FORMFEED

## BACKFEED & BACKUP

[Description] This command feeds the label in reverse.

[Syntax] BACKFEED n

| Parameter | Description |
|-----------|-------------|
| n         | Unit: dot   |
| 1         | n 9999      |

**CAUTION:** Improperly back feed value may cause paper jam or wrinkle.

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

[Example] BACKFEED 40  
BACKFEED 40

[See Also] FEED, SIZE, GAP, BLINE, HOME, FORMFEED

## FORMFEED

[Description] This command feeds label to the beginning of next label.

[Syntax] FORMFEED

[Example]  
SIZE 4,2.5  
GAP 0 mm,0  
SPEED 4  
DENSITY 7  
DIRECTION 0  
OFFSET 0.00  
REFERENCE 0,0  
SET PEEL OFF  
SET COUNTER @0 +1  
@0="000001"  
FORMFEED  
...

```
BOX 1,1,360,65,12
TEXT 25,25,"3",0,1,1,"FORMFEED COMMAND TEST"
TEXT 25,80,"3",0,1,1,@0
PRINT 3,1
```

[See Also]      FEED, SIZE, GAP, BLINE, HOME, FORMFEED

## HOME

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

[Syntax]          HOME

[Example]          Sample code

```
SIZE 4,2.5
GAP 2 mm,0
SPEED 4
DENSITY 7
DIRECTION 0
OFFSET 0.00
REFERENCE 0,0
SET PEEL OFF
SET COUNTER @0 +1
@0="000001"
HOME
CLS
BOX 1,1,360,65,12
TEXT 25,25,"3",0,1,1,"HOME COMMAND TEST"
TEXT 25,80,"3",0,1,1,@0
PRINT 3,1
```

[See Also]      FEED, SIZE, GAP, BLINE, FORMFEED

## PRINT

[Description]    This command prints the label format currently stored in the image buffer

[Syntax]          PRINT m[,n]

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

[Example]  
 SIZE 60 mm, 20 mm  
 SET COUNTER @1 1

```
CLS
TEXT 10,10,"3",0,1,1,@1
PRINT 3,2
```

[See Also]      SET COUNTER, INPUT, DOWNLOAD

## SOUND

[Description]      This command is used to control the sound frequency of the beeper. There are 10 levels of sounds. The timing control 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 1,200  
                         SOUND 2,200  
                         SOUND 3,200  
                         SOUND 4,200  
                         SOUND 5,200

## 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.

[Syntax]            (the English system, inch)

LIMITFEED n

(the metric system, mm)

LIMITFEED n mm

| Parameter | Description                             |
|-----------|---|
| N         | The maximum length for sensor detecting |

[Note]              1、The setting will remain resident in memory.  
                         2、The default value is 4 inches when printer initializes

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

## SELFTEST

|               |  |
|---------------|--|
| [Description] | At this command, the printer will print out the printer information. |
| [Syntax]      | SELFTEST   |
| [Example]     | SELFTEST   |

## Label Formatting Commands

### BAR

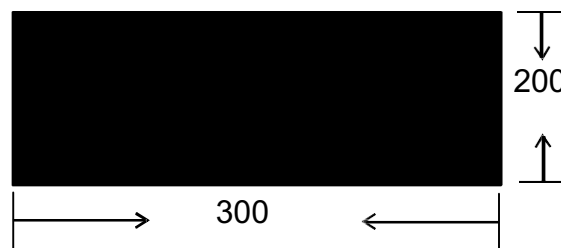
|               |   |
|---------------|---|
| [Description] | This command draws a bar on the label format. |
| [Syntax]      | BAR x, y, width, height                       |

| Parameter | Description                               |
|-----------|---|
| x         | The upper left corner x-coordinate in 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 = 12 dots

[Example]  
SIZE 4,2.5  
GAP 0,0  
SPEED 6  
DENSITY 8  
DIRECTION 0  
CLS  
BAR 100, 100, 300, 200  
PRINT 1,1

(100,100)



[See Also]      BOX



# BARCODE

[Description] This command prints 1D barcodes

The available barcodes are listed below:

- Code 128
- Code 128M
- EAN 128
- Interleaved 2 of 5
- Interleaved 2 of 5 with check digit
- Code 39 standard
- 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
- UPC-E
- UPC-E with 2 digits add-on
- UPC-E with 5 digits add-on
- MSI
- PLESSEY
- China POST
- ITF14
- EAN14

[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      |
| Codetype<br>128 | Code 128, switching code subset A, B, C automatically. |
| 128M            | Code 128, switching code subset A, B, C manually       |

|             |   |   |   |
|-------------|---|---|---|
| Controlcode | A | B | C |
|-------------|---|---|---|

|     |                |        |        |
|-----|----------------|--------|--------|
| 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 full ASCII with check digit
- 93 Code 93
- EAN 13 EAN 13
- EAN 13+2 EAN 13 with 2 digits add-on
- EAN 13+5 EAN 13 with 5 digits add-on
- EAN 8 EAN 8
- EAN 8+2 EAN 8 with 2 digits add-on
- EAN 8+5 EAN 8 with 5 digits add-on
- CODA Codabar
- POST Post net
- 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
- PLESSEY PLESSEY code
- ITF 14 ITF 14 code
- EAN 14 EAN 14 code
- Height bar code height expressed by 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 Wide

Width of narrow element (in dots) Width of wide element

(indots)

|         | narrow:wide<br>1:1 | narrow: wide<br>1:2 | narrow: wide<br>1:3 |     | narrow: wide<br>3:7 |
|---------|--------------------|---------------------|---------------------|-----|---------------------|
| 128     | 10x                | N/A                 | N/A                 | N/A | N/A                 |
| EAN128  | 10x                | N/A                 | N/A                 | N/A | N/A                 |
| 25      | N/A                | 10x                 | 10x                 | 5x  | N/A                 |
| 25C     | N/A                | 10x                 | 10x                 | 5x  | N/A                 |
| 39      | N/A                | 10x                 | 10x                 | 5x  | N/A                 |
| 39C     | N/A                | 10x                 | 10x                 | 5x  | N/A                 |
| 93      | N/A                | N/A                 | 10x                 | N/A | N/A                 |
| EAN13   | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| EAN13+2 | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| EAN13+5 | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| EAN8    | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| EAN8+2  | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| EAN8+5  | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| CODA    | N/A                | 10x                 | 10x                 | 5x  | N/A                 |
| POST    | 1x                 | N/A                 | N/A                 | N/A | N/A                 |
| UPCA    | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| UPCA+2  | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| UPCA+5  | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| UPCE    | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| UPCE+2  | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| UPCE+5  | 8x                 | N/A                 | N/A                 | N/A | N/A                 |
| CPOST   | N/A                | N/A                 | N/A                 | N/A | 1x                  |
| MSI     | N/A                | N/A                 | 10x                 | N/A | N/A                 |
| MSIC    | N/A                | N/A                 | 10x                 | N/A | N/A                 |
| PLESSY  | N/A                | N/A                 | 10x                 | N/A | N/A                 |
| ITF14   | N/A                | 10x                 | 10x                 | 5x  | N/A                 |
| EAN14   | N/A                | N/A                 | N/A                 | 5x  | N/A                 |

| Barcode type | Maximum bar<br>Code length | Barcode type | Maximum bar<br>Code length |
|--------------|----------------------------|--------------|----------------------------|
| 128          | —                          | POST         | 5,9,11                     |
| EAN128       | —                          | UPCA         | 11                         |

|         |    |        |    |
|---------|----|--------|----|
| 25C     | —  | UPCA+5 | 16 |
| 39      | —  | UPCE   | 6  |
| 39C     | —  | UPCE+2 | 8  |
| 93      | —  | UPCE+5 | 11 |
| EAN13   | 12 | CPOST  | —  |
| EAN13+2 | 14 | MSI    | —  |
| EAN13+5 | 17 | MSIC   | —  |
| EAN8    | 7  | PLESSY | —  |
| EAN8+2  | 9  | ITF14  | 13 |
| EAN8+5  | 12 | EAN14  | 13 |
| CODA    | —  |        |    |

[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 128function character FNC3 which is then followed by the ABCD characters and EFGH characters encoded as CODE A subset.

## BITMAP

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

[Syntax]        BITMAP X, Y, width, height, mode, bitmap data...

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

[Example]

|   | X Size 16 dot |   |   |   |   |   |   |   | 1 dot  |   |   |   |   |   |   |   |
|---|---------------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
|   | 0             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8      | 9 | A | B | C | D | E | F |
| 0 | 0             | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0             | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0             | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0             | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 0             | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 0             | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 0             | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0             | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0      | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0      | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0      | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| A | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1      | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| B | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1      | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| C | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1      | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| D | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1      | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| E | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F | 0             | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|   | 1 byte        |   |   |   |   |   |   |   | 1 byte |   |   |   |   |   |   |   |

| ROW<br>(Y- axis) | L-Byte   |             | R-Byte   |             |
|------------------|----------|-------------|----------|-------------|
|                  | Binary   | Hexadecimal | Binary   | Hexadecimal |
| 0                | 00000000 | 00          | 00000000 | 00          |
| 1                | 00000000 | 00          | 00000000 | 00          |
| 2                | 00000000 | 00          | 00000000 | 00          |
| 3                | 00000111 | 07          | 11111111 | FF          |
| 4                | 00000011 | 03          | 11111111 | FF          |
| 5                | 00010001 | 11          | 11111111 | FF          |
| 6                | 00011000 | 18          | 11111111 | FF          |
| 7                | 00011100 | 1C          | 01111111 | 7F          |
| 8                | 00011110 | 1E          | 00111111 | 3F          |
| 9                | 00011111 | 1F          | 00011111 | 1F          |
| A                | 00011111 | 1F          | 10001111 | 8F          |
| B                | 00011111 | 1F          | 11000111 | C7          |
| C                | 00011111 | 1F          | 11100011 | E3          |
| D                | 00011111 | 1F          | 11110111 | F7          |
| E                | 00011111 | 1F          | 11111111 | FF          |
| F                | 00011111 | 1F          | 11111111 | FF          |

[Example]      SIZE 4,2  
                   GAP 0,0  
                   CLS  
                   BITMAP 200,200,2,16,0,

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

[See Also] PUTBMP, PUTPCX

## BOX

[Description] This command draws rectangles on the label.

[Syntax] BOX X\_start, Y\_start, X\_end, Y\_end, line thickness

| Parameter      | Description                                    |
|----------------|--|
| X_start        | Specify x-coordinate of upper left corner (in  |
| Y_start        | Specify y-coordinate of upper left corner (in  |
| X_end          | Specify x-coordinate of lower right corner (in |
| Y_end          | Specify y-coordinate of lower right corner (in |
| line thickness | Line thickness (in dots)                       |

Note: 200 DPI: 1 mm = 8 dots

300 DPI: 1 mm = 12 dots

[Example]

```
SIZE 4,2.5
GAP 0,0
SPEED 6
DENSITY 8
DIRECTION 0
CLS
BOX 100,100,200,200,5
PRINT 1,1
```

[See Also] BAR

## ERASE

[Description] This command clear a specified region in image buffer.

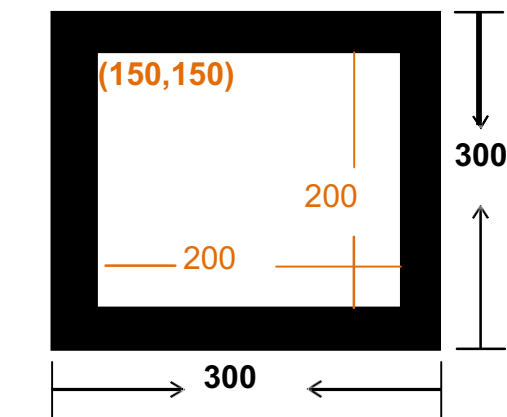
[Syntax] ERASE X\_start, Y\_start, X\_width, Y\_height

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

[Example]

```
SIZE 4,2.5
GAP 0,0
SPEED 6
DENSITY 8
DIRECTION 0
CLS
BAR 100, 100, 300, 300
ERASE 150,150,200,200
PRINT 1,1
```

(100,100)



[See Also] CLS

## PUTBMP

[Description] This command prints BMP format images. color) BMP graphic only.

[Syntax] PUTBMP X, Y, "filename "

| Parameter | Description                              |
|-----------|--|
| X         | The x-coordinate of the BMP format image |
| Y         | The y-coordinate of the BMP format image |
| filename  | The downloaded BMP filename.             |

|          |   |
|----------|---|
| bpp      | Optional. Bits per pixel of grayscale graphic. Default is 1. *Since V6.91EZ.<br>1: 1-bit (monochrome) graphic<br>8: 8-bit (256-color) graphic |
| contrast | Optional. Contrast of grayscale graphic. Default is 80. Suggested range is from 60 to 100. *Since V6.91EZ.                                    |

[Example]

```

C:\BMP-PCX>DIR
Volume in drive C is WIN98
Volume Serial Number is 4140-4735

Directory of C:\BMP-PCX

06/08/2008    03:06 PM    <DIR>
06/08/2008    03:06 PM    <DIR>
06/08/2008    03:58 PM                12,430 GP. bmp
06/08/2008    03:10 PM                1,181 GP. pcx
                2 File(s)                13,611 bytes
                2 Dir(s)  8,802,189,312 bytes free

C:\BMP-PCX>COPY CON LPT1
DOWNLOAD "GP. BMP", 12430, ^Z
                1 file(s) copied.

C:\BMP-PCX>COPY GP. BMP/B LPT1
                1 file(s) copied.

C:\BMP-PCX>COPY CON LPT1
SIZE 3, 2.5
GAP 0, 0
CLS
PUTBMP 100, 100, "GP. BMP"
PRINT 1, 1
^Z
                1 file(s) copied.
C:\BMP-PCX>_

```

[See Also]      DOWNLOAD, BITMAP, PUTPCX

## PUTPCX

[Description] This command prints PCX format images.

[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 filename. (Case sensitive) |



[Example]

```
C:\BMP-PCX>DIR
Volume in drive C is WIN98
Volume Serial Number is 4140-4735

Directory of C:\BMP-PCX

06/08/2008    03:06 PM    <DIR>
06/08/2008    03:06 PM    <DIR>
06/08/2008    03:56 PM                12,430 GP.bmp
06/08/2008    03:10 PM                1,181 GP.pcx
                2 File(s)                13,611 bytes
                2 Dir(s)  8,802,189,312 bytes free

C:\BMP-PCX>COPY CON LPT1
DOWNLOAD "GP.PCX",12430,^Z
                1 file(s) copied.

C:\BMP-PCX>COPY GP.PCX/B LPT1
                1 file(s) copied.

C:\BMP-PCX>COPY CON LPT1
SIZE 3,2.5
GAP 0,0
CLS
PUTBMP 100,100,"GP.PCX"
PRINT 1,1
^Z
                1 file(s) copied.
C:\BMP-PCX>_
```

[See Also]      DOWNLOAD, BITMAP, PUTPCX

## QRCODE

[Description] This command prints QR code.

[Syntax] QRCODE X, Y, ECC Level, cell  
width, mode, rotation, \*model, mask, +"Data  
string"

| 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<br>L : 7%<br>M : 15%                   |
|                   | Q : 25%<br>H : 30%   |
| cell width        | 1~10   |
| mode              | Auto / manual encode<br>A : Auto<br>M : Manual                         |
| rotation          | 0 : 0 degree<br>90 : 90 degree<br>180 : 180 degree<br>270 : 270 degree |
| Model<br>1:<br>2: | (default), original version<br>enhanced version                        |
| mask              | S0~S8, default is S7   |

|                                  |   |                        |                        |                               |                  |                                  |                  |                                |                  |                           |                  |
|----------------------------------|---|------------------------|------------------------|-------------------------------|------------------|----------------------------------|------------------|--------------------------------|------------------|---------------------------|------------------|
| content                          | <p>The encodable character set is described as below,</p> <p>Encodable character set:</p> <p>1) Numeric data: (digits 0~9)</p> <p>2) Alphanumeric data</p> <p>Digits 0-9</p> <p>Upper case letters A-Z</p> <p>Nine other characters: space, \$ % * + - . / : )</p> <p>3) 8-bit byte data</p> <p>JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201</p> <p>4) Kanji characters</p> <p>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</p> <p>Shift Coded Representation for detail.</p> <p>Data characters per symbol (for maximum symbol size):</p> <table><tr><td>Model 1 (Version 14-L)</td><td>Model 2 (Version 40-L)</td></tr><tr><td>Numeric data 1,167 characters</td><td>7,089 characters</td></tr><tr><td>Alphanumeric data 707 characters</td><td>4,296 characters</td></tr><tr><td>8-bit byte data 486 characters</td><td>2,953 characters</td></tr><tr><td>Kanji data 299 characters</td><td>1,817 characters</td></tr></table> <p>* If "A" is the first character in the data string, then the following data after "A" is alphanumeric data.</p> | Model 1 (Version 14-L) | Model 2 (Version 40-L) | Numeric data 1,167 characters | 7,089 characters | Alphanumeric data 707 characters | 4,296 characters | 8-bit byte data 486 characters | 2,953 characters | Kanji data 299 characters | 1,817 characters |
| Model 1 (Version 14-L)           | Model 2 (Version 40-L)  |                        |                        |                               |                  |                                  |                  |                                |                  |                           |                  |
| Numeric data 1,167 characters    | 7,089 characters  |                        |                        |                               |                  |                                  |                  |                                |                  |                           |                  |
| Alphanumeric data 707 characters | 4,296 characters  |                        |                        |                               |                  |                                  |                  |                                |                  |                           |                  |
| 8-bit byte data 486 characters   | 2,953 characters  |                        |                        |                               |                  |                                  |                  |                                |                  |                           |                  |
| Kanji data 299 characters        | 1,817 characters  |                        |                        |                               |                  |                                  |                  |                                |                  |                           |                  |

\*If "N" is the first character in the data string, then the following data after "N" is numeric data.

\*If "B" is the first character in the data string, then the following 4 digits after "B" is used to specify numbers of data. After the 4 digits is the number of bytes of binary data to be encoded.

\*If "K" is the first character in the data string, then the following data after "K" is Kanji data.

\*If "!" is in the data string and follows by "N", "A", "B", "K" then it will be switched to specified encodable character set.

Manual mode example:

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]

a. General data string:

SIZE 4,2.5

GAP 0.12,0

CLS

QRCODE 10,10,H,4,A,0,"ABCabc123"

QRCODE 160,160,H,4,A,0,"123ABCabc"

QRCODE 310,310,H,4,A,0,"打印机ABCabc123"

PRINT 1,1

.Data string including <Enter> character (0Dh,0Ah)

-----

```

GAP 0.12,0
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,"打印机<Enter>
ABC<Enter>
abc<Enter>
123"
PRINT 1,1
c..Data string concatenation (Must be used with DOWNLOAD EOP command)
DOWNLOAD "DEMO.BAS"
SIZE 4,2.5
CAP 0.12,0
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,"打印机ABCabc123"+"1234"+"abcd"
PRINT 1,1
EOP

```

```

DEMO
d.Data string including double quote (") character, please use \" instead of
SIZE 4,2.5
CAP 0.12,0
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,\"打印机\"ABCabc123"
PRINT 1,1

```

(2) Manual mode:

a.General data string:

```

SIZE 4,2.5
CAP 0.12,0
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 打印机!AABC!B0006abc123"
PRINT 1,1

```

b.Data string including <Enter> character, <Enter> is an 8-bit byte data

```

SIZE 4,2.5
CAP 0.12,0
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 打印机!B0002<Enter>
!AABC!B0010<Enter>
abc<Enter>
123"
PRINT 1,1

```

c. Data string concatenation (Must be used with DOWNLOAD EOP command)

```

SIZE 4,2.5
CAP 0.12,0
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 打印机!AABC!B0014abc123"+"1234"+"abcd"
PRINT 1,1
EOP

    A

```

d. Data string including double quote (") character, please use \" instead of

```

SIZE 4,2.5
CAP 0.12,0
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 打印机!B0010\"ABCabc123"
PRINT 1,1

```

## REVERSE

[Description] This command reverses a region in image buffer.

[Syntax] REVERSE X\_start, Y\_start, X\_width, Y\_height

| Parameter | Description                                      |
|-----------|--|
| X_start   | The x-coordinate of the starting point (in dots) |
| Y_start   | The y-coordinate of the starting point (in dots) |
| X_width   | X-axis region width (in dots)                    |
| Y_height  | Y-axis region height (in dots)                   |

Note: 200 DPI: 1 mm = 8 dots

300 DPI: 1 mm = 12 dots

```

[Example]
SIZE 4,2.5
GAP 0,0
SPEED 6
DENSITY 8
DIRECTION 0
CLS
TEXT 100,100,"3",0,1,1,"REVERSE"
REVERSE 90,90,128,40
PRINT 1,1

```

## TEXT

[Description] This command 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                    |

font

Font name

|                   |   |
|-------------------|---|
| 1                 | 8 x 12fixed pitch dot font  |
| 2                 | 12 x 20fixed pitch dot font   |
| 3                 | 16 x 24 fixed pitch dot font  |
| 4                 | 24 x 32fixed 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   |
| TST24.BF2         | Traditional Chinese 24 x 24font   |
| TSS24.BF2         | Simplified Chinese 24 x 24font(GB)  |
| K                 | Korean 24 x 24font (KS)   |
| Rotation          | The rotation angleof 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, upto10x.<br>Available factors:1~10width(point)of true type font. 1 point=1/72 inch.  |
| Y-multiplication: | Vertical multiplication, up to 10x.<br>Available factors: 1~10<br>For true type font, this parameter is used to specify the height (point) of true type font.<br>1 point=1/72 inch. |

Note:

The internal font (font #1~#5) pitch between TSPL and TSPL2 is different.

Font "0" and "ROMAN.TTF" internal True Type Fonts are available in TSPL2 language printers, but not TSPL language printers.

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

If font "0" is used, the font width and font height is stretchable by x-multiplication and y-multiplication parameter. It is expressed by pt (point). 1 point=1/72inch.

EPL2 and ZPL2 are emulating for Eltron® and Zebra® languages

[Example] 1:

```

GAP 0,0
CLS
TEXT 100,100,"5",0,1,1,"["]DEMO FOR TEXT\["]
TEXT 100,200,"ROMAN.TTF",0,1,20,"["]True Type Font Test Print\["]
PRINT 1,1

[Example] 2:
  DOWNLOAD F,"test.BAS"
  SPEED 4
  DENSITY 8
  SET PEEL OFF
  DIRECTION 0
  SIZE 1.000,0.800
  GAP 0.12,0.00
  OFFSET 0.000
  REFERENCE 0,0
:START
CLS
INPUT
"Description",A$ Again:
F$=""
INPUT "Font:",1,F$
IF ASC(F$)>53 THEN GOTO
Again IF ASC(F$)<49 THEN
GOTO Again
X_Again:
INPUT "X_multiple:",1,XM
IF XM>8 THEN GOTO X_Again
IF XM<1 THEN GOTO X_Again
Y_Again:
INPUT "Y_multiple:",1,YM
IF YM>8 THEN GOTO Y_Again
IF YM<1 THEN GOTO Y_Again
Z=LEN(A$)
WWidth=Z*XM*8
IF F$="1" THEN
  REM *****When Font1, the space between every single word is XM*2 dots*****
  WWidth=Z*(XM*8)
  SWidth=Z*(XM*2)
  X=101-((WWidth+SWidth)/2)
ELSEIF F$="2" THEN
  REM *****When Font2, the space between every single word is XM*2 dots*****
  WWidth=Z*(XM*12)
  SWidth=Z*(XM*2)
  X=101-((WWidth+SWidth)/2)
ELSEIF F$="3" THEN
  REM *****When Font3, the space between every single word is XM-1 dots*****
  WWidth=Z*(XM*16)
  SWidth=Z*(XM-1)
  X=101-((WWidth+SWidth)/2)
ELSEIF F$="4" THEN
  REM *****When Font4, the space between every single word is XM-1 dots*****
  WWidth=Z*(XM*24)
  SWidth=Z*(XM-1)
  X=101-((WWidth+SWidth)/2)
ELSEIF F$="5" THEN

```



```
SWidth=Z*(XM*4)
X=101-((WWidth+SWidth)/2)
ENDIF
TEXT
X,116,F$,0,XM,YM,A$ PRINT
1,1
GOTO START
EOP
```

## Status Polling Commands

### <ESC>!?

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

| Bit | Status                                      |
|-----|---|
| 0   | Head opened                                 |
| 1   | Paper jam                                   |
| 2   | Out of paper                                |
| 3   | Out of ribbon                               |
| 4   | Pause                                       |
| 5   | Printing                                    |
| 6   | Cover opened (option)                       |
| 7   | Environment Temperature over range (option) |

[Syntax] <ESC>! ?

[See Also] <ESC>!R

## <ESC>!R

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

[Syntax] <ESC>! R

| Parameter | Description |
|-----------|-------------|
| N/A       | N/A         |

[See Also] <ESC>! ?

## ~!@

[Description] This command inquire the mileage of the printer. The integer part of mileage is returned (the decimal part of mileage is not return). It is returned to PC in ASCII characters. The ending character of mileage is 0x0D 0x0A

[Syntax] ~! @

[Example] ~! @

## ~!A

[Description] This command inquires the free memory of the printer. The number of bytes of free memory is returned in decimal digits, with 0x0D 0x0A as ending code of PC.

[Syntax] ~!A

[Example] ~!A

[See Also] FILES

## ~!C

[Description] This command inquires the presence of Real Time Clock. One byte is return from the printer, indicating whether or not the RTC is installed.

[Syntax] ~!C

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

[Example] ~!C

## ~!D

[Description] This command enters the printer into DUMP mode. I

[Syntax] ~!D

[Example] ~!D

## ~!F

[Description] This command inquires all about files resident in the printer memory, and fonts installed in the memory module. The filename are returned in ASCII characters. Each file name ends with 0x20. The ending character is 0x0D 0x1A. Entering this command multiple times will cycle through the files resident on memory.

[Syntax] ~!F

[Example]     ~!F  
 [See Also]    FILES

## ~!!

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

Data through RS-232

Example: 8 bit : 437,001

Regarding the code pages supported by the printer, please refer to the CODEPAGE command

[Syntax]        ~!!  
 [Example]       ~!!  
 [See Also]      CODEPAGE

## ~!T

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

[Syntax]        ~!T

[Example]       ~!T

[See Also]      ~!!, ~!F

# FileManagement Commands

## DOWNLOAD

[Description] "DOWNLOAD" is a header of the file that is to be saved in the printer's memory. The downloaded files can be divided into two categories: program files and data files (including text data files, PCX graphic files and bitmap font files)

[Syntax]        1. Download a program file  
 DOWNLOAD[n,]"FILENAME.BAS"

| Parameter | Description |
|-----------|-------------|
|-----------|-------------|

|              |   |
|--------------|---|
| n            | Specify the memory which is used to save the download files.                  |
| n is ignored | Download files to DRAM only.<br>F: Download files to main board flash memory. |
| FILENAME.BAS | The filename resident in printer memory.                                      |

**Note:**

- (1). *Filenames are case sensitive.*  
(2). *File extensions must be ".BAS"*  
(3). *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.*

## 2. Download a data file

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

| Parameter    | Description   |
|--------------|---|
| n            | Specify the memory location to save the download files.                         |
| n is ignored | Download files to DRAM only. I<br>F: Download files to main board flash memory. |
| FILENAME.    | The name of data file that will remain resident in the printer memory.          |
| DATA SIZE    | The actual size (numbers of byte) of the data file without header.              |

**Note:**

- (1). *For text data file, CR (carriage return) 0x0D and LF (Line Feed) 0x0A is the separator of data.*  
(2). *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.*

[Example] The example program listed below will download to printer SDRAM.

DOWNLOAD “EXAMPLE.BAS”  
SIZE 11

```

DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
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.***

***To run the program, call the main filename without BAS extension or use RUN command to start the download program.***

[Example]

1. Call the main filename:

```
C:\>COPY CON LPT1<ENTER>
```

```
EXAMPLE<ENTER>
```

```
<CTRL><Z>
```

```
C:\>
```

2. Use Run command to start the program:

```
C:\>COPY CON LPT1<ENTER>
```

```
RUN "EXAMPLE.BAS"<ENTER>
```

```
<CTRL><Z>
```

```
C:\>
```

Below is an example of downloading data file.

```
DOWNLOAD "DATA",20,COMPUTER<Enter>
```

```
2001<Enter>
```

```
21<Enter>
```

**Note:**

***<ENTER> stands for PC keyboard "ENTER" key.***

[See Also] EOP, RUN, PUTBMP, PUTPCX, INPUT, FILES, ~!F

## EOP

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

[Syntax] EOP

[Example] DOWNLOAD "DEMO.BAS"

SIZE 4,4

GAP 0,0

DENSITY 8

SPEED 6

DIRECTION 0

REFERENCE 0,0

```
CLS
TEXT 100,100,"3",0,1,1,"DEMO PROGRAM"
PRINT 1
EOP
[See Also]    DOWNLOAD, EOP
```

## FILES

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

[Syntax] FILES

### [Example]

That are saved in printer memory in DOS environment through serial port or parallel port connection.

Serial port:

```
C:\>MODE COM1 96,N,8,1<ENTER>
```

```
C:\>COPY CON COM1<ENTER>
```

```
FILES<ENTER>
```

```
<CTRL><Z><ENTER>
```

```
C:\>
```

Parallel port:

```
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, KILL

## KILL

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

[Syntax] KILL \*n+,"FILENAME"

| Parameter    | Description  |
|--------------|--|
| n            | Specify the memory location that files will be deleted.                  |
| n is ignored | Kill files saved in DRAM.<br>F: Kill files from main board flash memory. |

**Note:**

(1). If optional parameter *n* is not specified, firmware will delete the file in DRAM.

Syntax example

1. KILL "FILENAME"
2. KILL "\*.PCX"
3. KILL "\*\*"
4. KILL F,"FILENAME"

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

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

**Note:**

<ENTER> stands for PC keyboard "ENTER" key.

<CTRL><Z> means to hold PC keyboard "CTRL" key then press the PC keyboard <Z> key

[See Also] ~!F, FILES

## MOVE

[Description] This command moves downloaded files from DRAM to FLASH memory.

[Syntax] MOVE

| Parameter | Description |
|-----------|-------------|
| N/A       | N/A         |

[See Also] DOWNLOAD, EOP

## RUN



[Syntax]      RUN "FILENAME.BAS"  
 Example]

```
C:\>COPY CON LPT1<ENTER>
RUN "DEMO.BAS"<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

[See Also]      DOWNLOAD, EOP

## BASIC Commands and Functions

### ABS( )

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

[Syntax]  
 ABS (-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 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

[See Also]      DOWNLOAD, EOP

### ASC( )

[Description]      This function returns the ASCII code of the character.

[Syntax]      ASC("A")

[Example]

```

SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
CLS
CODE1=ASC("A")
TEXT 100,100,"3",0,1,1,STR$(CODE1)
PRINT 1
EOP

```

[See Also]      DOWNLOAD, EOP,STR\$()

## CHR\$( )

[Description]    This function returns the character with the specified ASCII code.

[Syntax]          CHR\$(n)

| Parameter | Description    |
|-----------|----------------|
| n         | The ASCII code |

[Example]

```

DOWNLOAD "TEST.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
CLS
A=65
WORD$=CHR$(A)
TEXT
100,100,"3",0,1,1,WORD$ PRIN
T 1
EOP

```

[See Also]      DOWNLOAD, EOP, STR\$(), ASC\$()

## END

[Description]    This command states the ending of program.

[Syntax]          END

[Example]

```

DOWNLOAD "DEMO.BAS"
SIZE 4,2
GAP 0,0

```

```

SPEED 6
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
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

```

[See Also]      DOWNLOAD, 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 "DATA",16,COMPUTER
2000
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.0,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
CLS
OPEN "DATA",0
SEEK 0,0
Y=110
TEXT 10,10,"3",0,1,1,"*****EOF TEST*****"
:A
Temp$=""
READ 0 ITEM$ P

```

```

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

```

[See Also]      DOWNLOAD, EOP, OPEN, READ, SEEK

## OPEN

[Description] This command opens a downloaded file and establishes the file handle.

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

[Syntax]          OPEN "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" is to be downloaded, The file format contains:

```

DOWNLOAD "DATA1",56,COMPUTER
2000
12
MOUSE
500
13
KEYBOARD
300
100
DOWNLOAD "DATA2",56,Computer
3000
32
Mouse
900
93
Keyboard
700
700

```

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

```

OPEN COPY DATA1.DAT

```

If a file by name of "DEMO.BAS is to be downloaded, the file format contains:

```

DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
I=1
Y=100
GOSUB OpenData
:Start
CLS
TEXT 10,10,"3",0,1,1,"*****OPEN COMMAND TEST*****"
ITEM$=""
READ 0,ITEM$,P,Q
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$(Q)+"="+STR$(P*Q)
Y=Y+100
PRINT 1
Y=100
IF EOF(0)=1 THEN GOSUB OpenData
IF EOF(0)=0 THEN GOTO Start
END
:OpenData
IF I=1 THEN OPEN "DATA1",0
IF I=2 THEN OPEN "DATA2",0
SEEK 0,0
IF I>2 THEN END
I=I+1
RETURN
EOP
DEMO

```

Saving the above contents of data under the file name of "DEMO".

Follow the steps below to download data to the printer

< In DOS mode>:

C:\>COPY DEMO/B LPT1

Execute DEMO.BAS in printer:

C:\>COPY CON LPT1

DEMO

<Ctrl><Z>

The above example instructs the printer to open the file "DATA1" and "DATA2" with same file handle of 0, and read items from the file.

[See Also] DOWNLOAD, EOP, READ, EOF, LOF, SEEK, FREAD\$()

## READ

[Description] This command reads data from downloaded data

file. [Syntax] READ file handle, variables

| Parameter   | Description |
|-------------|-------------|
| File handle | 0 or 1.     |

```

[Example]
DOWNLOAD "DATA1",20,COMPUTER
2000
12
DOWNLOAD "DATA2",16,Mouse
900
93
DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
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

```

[See Also]      DOWNLOAD, EOP, OPEN, EOF, LOF, SEEK, FREAD\$()

## SEEK

[Description]    This command shifts the specified file pointer to a certain position

[Syntax]      SEEK file handle, offset

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

[Example]

```

DOWNLOAD "DATA",12,1234567890
DOWNLOAD "TEST.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 3
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 GAP 0,0

```

[See Also]      DOWNLOAD, EOP, OPEN, READ, EOF, LOF, FREAD\$()

## LOF( )

[Description]      This function returns the size of the specified file.

[Syntax]      LOF("FILENAME")

| Parameter | Description                                |
|-----------|--|
| FILENAME  | The file downloaded in the printer memory. |

[Example]

```

DOWNLOAD "DATA1",10,1234567890
DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO
DOWNLOAD "LoFTest.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 3
DIRECTION 0

```

```

SET PEEL OFF
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

```

[See Also]      DOWNLOAD, EOP, OPEN, READ, EOF, SEEK,  
FREAD\$()

## FREAD\$( )

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

[Syntax]            FREAD\$ (file handle, byte)

| Parameter   | Description                |
|-------------|----------------------------|
| File handle | 0 or 1                     |
| byte        | Number of bytes to be read |

[Example]

```

DOWNLOAD "DATA1",10,1234567890
DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO
DOWNLOAD "OPEN2.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
CLS
Y$=""
Z$=""
OPEN "DATA1",0
OPEN "DATA2",1
SEEK 0,0
SEEK 1,0
TEXT 10,260,"3",0,1,1,"FREAD$(0,6) IS: "+Y$
TEXT 10,320,"3",0,1,1,"FREAD$(1,6) IS:
"+Z$ PRINT 1
EOP

```

[See Also]      DOWNLOAD, EOP, OPEN, READ, EOF, LOF(),  
SEEK



## FOR...NEXT LOOP

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

[Syntax]      FOR variable = start TO end STEP increment  
                  statement; start < end  
                  NEXT

| Parameter | Description                                      |
|-----------|--|
| variable  | The variable name is 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]

```

DOWNLOAD"LOOP.BA
S"
    SIZE 3,3
    GAP 0.08,0

DENSITY 8
SPEED 4
DIRECTION 1
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
LOOP

```

[See Also]      DOWNLOAD, EOP

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

[Description]      Use IF...THEN block to execute one or more statements conditionally. Either a single-line syntax or multiple-line "block" syntax can be used:

[Syntax]           IF condition THEN statement

Notice that 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

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             |

[Example]

DOWNLOAD "DEMO.BAS"

SIZE 3,3

GAP 0.12,0

SPEED 4

DENSITY 8

DIRECTION 1

REFERENCE 0,0

OFFSET 0.00

SET PEEL OFF

CLS

A=0

B=0

C=0

D=0

E=0

F=0

G=0

H=0

J=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$(G)
TEXT 100,460,"3",0,1,1,"(7) 17+34+51+...+85="+STR$(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,"
PRINT 1,1
(1)-(6)-(7)-(8)-(9)="+STR$(L)
EOP
DOWNLOAD "IFTHEN.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
CLS
A=50
B=5
C$=""
D$=""
:L1
IF A>100 THEN GOTO L1 ELSE A=A+10
C$=STR$(A)+" IS SMALLER THAN 100"
TEXT 100,10,"4",0,1,1,C$

```

```

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

**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.*

[See Also]      DOWNLOAD, EOP

## GOSUB...RETURN

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

[Syntax]            GOSUB LABEL  
                       Statement  
                       END  
                       :LABEL  
                       statement  
                       RETURN

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

[Example]

```

DOWNLOAD "GOSUB1.BAS"
SIZE 3,3
GAP 0,0
DENSITY 8
SPEED 4
DIRECTION 0
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

```

EOP

GOSUB1

[See Also]     DOWNLOAD, EOP, END, GOTO

## GOTO

[Description]     This command is used to branch to a specified label. The label can not exceed 8 characters in length.

[Syntax]           GOTO LABEL  
                      :LABEL  
                      Statement

[Example]

```

DOWNLOAD "GOTO1.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 1
REFERENCE 0,0
SET PEEL OFF
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

```

[See Also]        DOWNLOAD, EOP, END, GOSUB...RETURN

## REM

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

[Syntax]           REM string

```

REM *****
REM This is a demonstration program*
REM *****
DOWNLOAD "REMARK.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 1
REFERENCE 0,0
SET PEEL OFF
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

```

[See Also]      DOWNLOAD, EOP, END

## INT( )

[Description]    This function truncates a floating point number.

[Syntax]        INT (n)

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

```

[Example]
DOWNLOAD "DEMO.BAS"
SIZE 4,2
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
CLS
REM **** To round up or down****
INPUT "Number:",Num
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

```

[See Also]      DOWNLOAD, EOP, END, ABS(), ASC(), STR\$()

## LEFT\$( )

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

[Syntax] LEFT\$(X\$, n)

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

[Example]  
 DOWNLOAD "STR1.BAS"  
 SIZE 3.00,3.00  
 GAP 0.08,0.00  
 SPEED 4.0  
 DENSITY 8  
 DIRECTION 0  
 REFERENCE 0,0  
 CLS  
 A\$="BARCODE PRINTER DEMO PRINTING"  
 C\$=LEFT\$(A\$,10)  
 TEXT 10,10,"3",0,1,1,A\$  
 TEXT 10,100,"3",0,1,1,"10 LEFT 10 CHARS: "+C\$  
 PRINT 1  
 EOP

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

## LEN( )

[Description] This function returns the length of a string.

[Syntax] LEN (string)

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

[Example]  
 DOWNLOAD "DEMO.BAS"  
 SIZE 3.00,3.00  
 GAP 0.08,0.00  
 SPEED 4.0  
 DENSITY 8  
 DIRECTION 0  
 REFERENCE 0,0  
 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
```

[See Also]      DOWNLOAD, EOP, END, LEFT\$(), LEN(), RIGHT\$(), MID\$(), STR\$(), VAL()

## MID\$( )

[Description]    This function retrieves the specified number of characters down from the *m*th character of a string.

[Syntax]          MID\$(string, m, n)

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

```
[Example]
DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0.08,0.00
SPEED 4.0
DENSITY 8
DIRECTION 0
REFERENCE 0,0
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
```

[See Also]      DOWNLOAD, EOP, END, LEFT\$(), LEN(), RIGHT\$(), STR\$(), VAL()

## 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 "DEMO.BAS"
SIZE 3.00,3.00
GAP 0.08,0.00
SPEED 4.0
DENSITY 8
DIRECTION 0
REFERENCE 0,0
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

```

[See Also]      DOWNLOAD, EOP, END, LEFT\$(), LEN(), MID\$(), STR\$(), VAL()

**STR\$( )**

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

[Syntax]            STR\$(n)

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

**[Example]**

```

DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0,0.00
SPEED 4.0
DENSITY 8
DIRECTION 0
REFERENCE 0,0
CLS
A$="TAIWAN SEMICONDUCTOR CO., LTD"
F=100
G=500
H$=STR$(F+G)
TEXT 10,10,"3",0,1,1,A$
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$ PRIN
T 1
EOP

```

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

## VAL( )

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

[Syntax]          VAL("numeric character")

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

[Example]

```

DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0.00,0.00
SPEED 4.0
DENSITY 8
DIRECTION 0
REFERENCE 0,0
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 10,60,"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$ PRIN
T 1
EOP
DEMO

```

[See Also]      DOWNLOAD, EOP, END, LEFT\$(), LEN(), RIGHT\$(), MID\$(), STR\$()

## BEEP

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

[Syntax]          BEEP

| Parameter | Description |
|-----------|-------------|
| None      | N/A         |

[Example]

```

SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
CLS
BEEP
INPUT "Text1 =",TEXT1$
TEXT
100,100,"3",0,1,1,TEXT1$ PRIN
T 1
EOP

```

## Device Reconfiguration Commands

### SET COUNTER

[Description]                      Counters can be a real counter or a variable. This setting sets the counter number in the program and its increments.

[Syntax]                            SET COUNTER @n step  
@n = "Expression"

| Parameter  | Description   |
|------------|---|
| @          | n: counter number. There are 50 counters available (@0 ~ @49) in the printer.   |
| step       | The increment of the counter, can be positive or negative. -999999999<= step <=999999999<br>If the counter is used as a fixed variable, |
| Expression | Initial string. String length is 101 bytes  |

[Example]  
SIZE 3,3  
GAP 0,0  
DENSITY 8

```

DIRECTION 0
REFERENCE 0,0
SET COUNTER @1 1
@1="00001"
SET COUNTER @2 5
@2="AB000001"
CLS
TEXT 50,50,"3",0,1,1,@1
BARCODE 50,100,"39",48,1,0,2,4,@2
PRINT 2,1
[See Also] PRINT, TEXT, BARCODE

```

## SET KEY1, SET KEY2

[Description] This setting is used to enable/disable the KEY1/KEY2 function. The default function of KEY1 is "FEED" key, KEY2 is "PAUSE" key. Before setting KEY1/KEY2 function otherwise, please disable KEY1/KEY2 first. The setting will remain resident in the printer even when the printer is power off.

[Syntax] SET KEY1 ON/OFF  
SET KEY2 ON/OFF

| Parameter | Description           |
|-----------|-----------------------|
| ON        | Enable KEYn function  |
| OFF       | Disable KEYn function |

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

```

[Example]
DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET KEY1 OFF
SET KEY2 OFF
KEY1=0
KEY2=0
CLS
:START
IF KEY1=1 THEN
CLS
TEXT 100,10,"3",0,1,1,"KEY1 (PAUSE key) is pressed!!"
PRINT 1,1
ELSEIF KEY2=1 THEN

```

```
TEXT 100,10,"3",0,1,1,"KEY2 (FEED key) is pressed!!"
PRINT 1,1
SET KEY1 ON
SET KEY2 ON
END
ENDIF
GOTO START
EOP
DEMO
[See Also]      OFFEST, PRINT
```

## SET PEEL

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

[Syntax] SET PEELON/OFF

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

```
[Example]
REM ***SELF-PEELING FUNCTION ON***
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET PEEL ON
CLS
TEXT 50,100,"3",0,1,1,"SELF-PEELING
FUNCTION TEST"
PRINT 5
[See Also]      OFFEST, PRINT
```

## SET TEAR & SETSTRIPPER

[Description] This command is used to enable/disable feeding of labels to gap/black mark position for tearing off.

This setting will be saved in printer memory when turning off the power.

[Syntax] SET TEARON/OFF

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

[Example]

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

[See Also] SET PEEL

## SET HEAD

[Description] This setting is used to enable/disable head open sensor. If the head open sensor is turned off, an open printer head will not return an error message. This setting will be saved in printer memory.

[Syntax] SET TEARON/OFF

| Parameter | Description                    |
|-----------|--------------------------------|
| ON        | Turn on the "HEADOPEN" sensor  |
| OFF       | Turn off the "HEADOPEN" sensor |

[Example] SET HEAD ON  
SET HEAD OFF

## SET RIBBON

[Description] This setting is used to enable/disable ribbon sensor detection. (Thermal Transfer Printing/Thermal Direct Printing) Printer will detect the presence of a ribbon to determine using either direct thermal or thermal

printer memory.

[Syntax] SET RIBBON ON/OFF

| Parameter | Description               |
|-----------|---------------------------|
| ON        | Thermal transfer printing |
| OFF       | Thermal direct printing   |

[Example]

REM \*\*\*Direct printing\*\*\*\*

SIZE 4,4

GAP 0,0

DENSITY 8

SPEED 6

DIRECTION 0

REFERENCE 0,0

SET PEEL OFF

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 :<br>24: 2400 bps<br>48: 4800 bps<br>96: 9600 bps<br>19: 19200 bps |
| Parity    | Parity check<br>N: None parity check<br>E: Even parity check<br>O: Odd parity check                              |
| data      | Data bit<br>8: 8 bits data<br>7: 7 bits data   |
| stop      | Stop bit<br>1: 1 stop bit<br>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 PC keyboard <Z> key.**



## SET PRINTKEY

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

[Syntax] SET PRINTEKYOFF/ON/AUTO/<num>

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

[Example]  
 SIZE 4,2.5  
 GAP 0.12,0  
 SET FEEDPRINT ON  
 SET COUNTER @0 1  
 @0="0001"  
 CLS  
 TEXT 10,10,"5",0,1,1,@0  
 PRINT 1

Execute:

| Syntax                                 | Receive "PRINTm"   | Print Out |
|--|--------------------|-----------|
| SET PRINTKEYON or<br>SET PRINTKEY AUTO | 1.) PRINT 2        | Label 1~2 |
|  | 2.) Press FEED key | Label 3~4 |

| Syntax                                 | Receive "PRINT m,n" | Print Out        |
|--|---------------------|------------------|
| SET PRINTKEYON or<br>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 PRINTKEYON or<br>SET PRINTKEY AUTO | 1.) PRINT -1,2       | Label 1, Label 1 |
|  | 2.) Press FEED key   | Label 1, Label 1 |

| Syntax         | Receive "PRINTm"   | Print Out |
|----------------|--------------------|-----------|
| SET PRINTKEY 5 | 1.) PRINT 2        | Label 1~2 |
|                | 2.) Press FEED key | Label 3~7 |

|                |                    |                  |
|----------------|--------------------|------------------|
| SET PRINTKEY 5 | 1.) PRINT 1,2      | Label 1, Label 1 |
|                | 2.) Press FEED key | Label 2~6        |

|                |                      |                  |
|----------------|----------------------|------------------|
| Syntax         | Receive "PRINT -1,n" | Print Out        |
| SET PRINTKEY 5 | 1.) PRINT -1,2       | Label 1, Label 1 |
|                | 2.) Press FEED key   | Label 1, Label 1 |

## SET REPRINT

[Description] This command will disable/enable a reprinting attempt subsequent to a "no paper", "no ribbon" or "carriage open" error.

[Syntax] SET REPRINTOFF/ON

| Parameter | Description           |
|-----------|-----------------------|
| OFF       | Disable this function |
| ON        | Enable this function  |

[Example] SET REPRINTON

## PEEL

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

[Syntax] PEEL

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

```
[Example]
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
SPEED 4
DENSITY 8
SET PEEL OFF
SET KEY1 OFF
:START
IF KEY1=1 THEN GOTO A
GOTO START
:A
CLS
TEXT 10,10,"3",0,1,1,"PEEL Function Test!!"
PRINT 1,1
~
```

```
IF PEEL=1 THEN
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
```

## KEY1, KEY2

[Description] This command read the status of KEY1,KEY2.

[Syntax] KEY m=n

| Key         | Return Value              |
|-------------|---------------------------|
| KEY1 (FEED) | 0: released<br>1: pressed |
| KEY2(PAUSE) | 0: released<br>1: pressed |

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

# Printer Global Variables

## @LABEL

[Description] This variable counts how many pieces of labels have been printed.

This attribute cannot be initialized if the printer is reset, but will be retained if the printer power is turned off.

[Syntax] Write attribute: @LABEL=n or @LABEL="n"

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

| Parameter | Description                             |
|-----------|---|
| n         | Number of labels printed. 0 n 999999999 |

[Example]

```
DOWNLOAD "DEMO.BAS"
```

```
SIZE 4,2.5
```

```
GAP 2 mm,0
```

```
SPEED 6
```

```
DENSITY 12
```

```
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
```