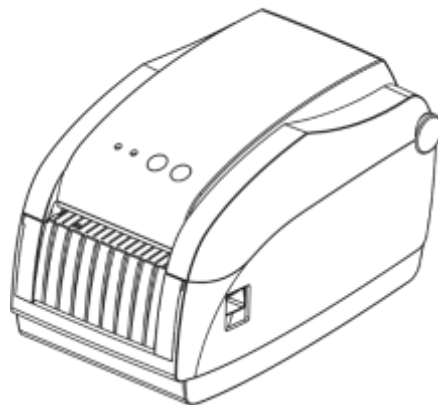


# **EC-3150**

## **Barcode Thermal Printer**

### **Programming Manual**



# CONTENTS

Setup and System Commands.....	1
1. SIZE .....	1
2. GAP.....	2
3. BLINE.....	2
4. OFFSET .....	4
5. SPEED .....	5
6. DENSITY.....	5
7. DIRECTION and Mirror Image.....	6
8. REFERENCE.....	6
9. SHIFT.....	8
10. CODEPAGE.....	9
11. CLS .....	10
12. FEED.....	10
13. BACKFEED & BACKUP .....	11
14. FORMFEED .....	12
15. HOME.....	13
16. PRINT .....	14
17. SOUND.....	15
18. LIMITFEED.....	16
19. SELFTTEST.....	16
Label Formatting Commands .....	17
20. BAR .....	17
21. BARCODE.....	18
22. BITMAP.....	22
23. BOX .....	24
24. ERASE.....	25
25. PUTBMP.....	26
26. PUTPCX .....	27
27. QRCODE .....	28
28. REVERSE.....	31
29. TEXT .....	32
Status Polling Commands (RS-232).....	33
30. <ESC>!?.....	33
<ESC>!R.....	34
31. ~!@ .....	35
32. ~!A .....	35
~!C .....	35
33. ~!D .....	36
34. ~!F .....	36
35. ~!I.....	37
36. ~!T.....	37
File Management Commands .....	38
37. DOWNLOAD .....	38
38. EOP .....	40
39. FILES .....	41
40. KILL.....	41
41. MOVE.....	43

42. RUN .....	43
<b>BASIC Commands and Functions .....</b>	<b>44</b>
43. ABS() .....	44
44. ASC() .....	45
45. CHR\$( ).....	46
46. END .....	47
47. EOF() .....	48
48. OPEN .....	49
49. READ.....	51
50. SEEK.....	52
51. LOF() .....	54
52. FREAD\$( ).....	55
53. FOR...NEXT LOOP .....	56
54. IF...THEN...ELSE...ENDIF LOOP .....	57
55. GOSUB...RETURN .....	60
56. GOTO.....	61
57. REM .....	62
58. INT() .....	63
59. LEFT\$( ) .....	64
60. LEN() .....	65
61. MID\$( ) .....	66
62. RIGHT\$( ).....	67
63. STR\$( ).....	68
64. VAL() .....	69
65. BEEP .....	70
<b>Device Reconfiguration Commands.....</b>	<b>71</b>
66. SET COUNTER.....	71
67. SET KEY1, SET KEY2 .....	72
68. SET PEEL .....	73
69. SET TEAR & SET STRIPPER .....	74
70. SET HEAD.....	75
71. SET RIBBON.....	75
72. SET COM1.....	76
73. SET PRINTKEY .....	77
74. SET REPRINT .....	79
75. PEEL .....	79
76. KEY1, KEY2 .....	80
<b>Printer Global Variables .....</b>	<b>81</b>
77. @LABEL .....	81

# Document Conventions

This manual uses the following typographic conventions.

Convention	Description
[expression list]	Items inside square brackets are optional, expression maximum length 2*1024 bytes;
<ESC>	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.
Space	(ASCII 32) characters will be ignored in the command line.
“	(ASCII 34), beginning and ending of expression
CR, LF	(ASCII 13), (ASCII 10) is placed at the end of command line.
<b>Note: 200 DPI: 1 mm = 8 dots</b>	<i>Times New Roman font in bold and italic type is used for note.</i>

# Setup and System Commands

## 1. 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

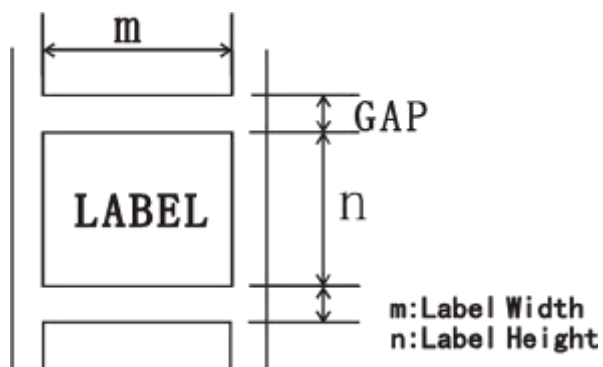
<u>Parameter</u>	<u>Description</u>
m	Label width (inch or mm)
n	Label length (inch or mm)

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

*For metric system, there must be a space between parameter and “mm”.*

### Example

- (1) English system (inch)  
SIZE 3.5, 3.00
- (2) Metric system (mm)  
SIZE 100 mm, 100 mm



### See Also

GAP, BLINE

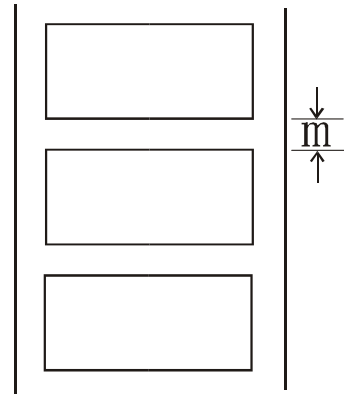
## 2. GAP

### 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

m

### Description

The gap distance between two labels  
 $0 \leq m \leq 1$  (inch),  $0 \leq m \leq 25.4$  (mm)

n

The offset distance of the gap  
 $n \leq \text{label length}$  (inch or mm)

0,0

Continuous label.

**Note:** 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.

**Ex:** In DOS mode,

`C:\>copy con lpt1 <Enter>`

`GAP 2 mm,0 <Enter>`

`<Ctrl>+<Z> <Enter>`

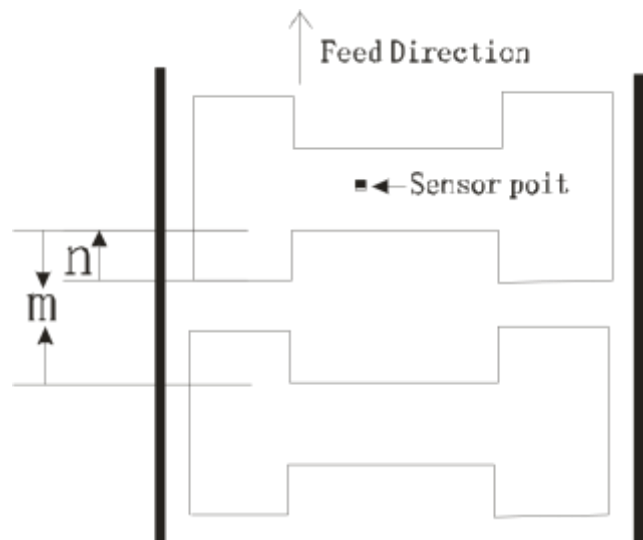
### Example

#### Normal gap

- (1) English system (inch)  
GAP 0.12,0
- (2) Metric system (mm)  
GAP 3 mm,0
- (3) Continuous label  
GAP 0,0

#### Special gap

- (1) English system (inch)  
GAP 0.30, 0.10
- (2) Metric system (mm)  
GAP 7.62 mm, 2.54 mm



### See Also

SIZE, BLINE

### 3. BLINE

#### Description

Set the height of the black mark and the length of the user-defined label

#### Syntax

- (1) English system (inch)  
BLINE m, n
- (2) Metric system (mm)  
BLINE m mm, n mm

<u>Parameter</u>	<u>Description</u>
m	Black mark height $0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm)
n	Extra feed paper length $0 \leq \text{label length}$ (inch or mm)
0,0	Continuous label.

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

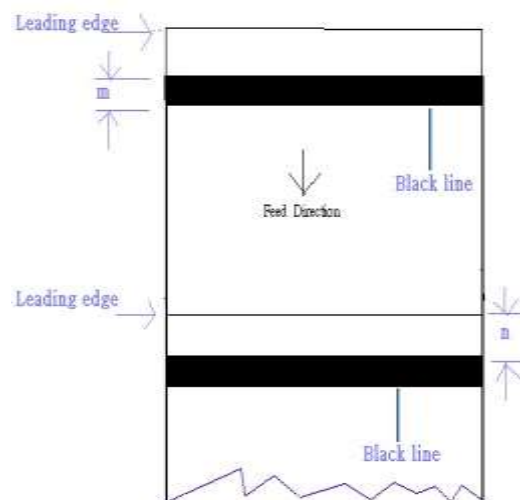
When the sensor type is changed from “GAP” to “Black Mark”, please send the “BLINE” command to the printer first.

Ex: 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)  
Bline5.08mm, 127mm
- (3) Continuous label  
GAP 0,0



#### See Also

SIZE, GAP

## 4. OFFSET

### Description

This command defines the selective, extra label feeding length each form feed takes, which, especially in peel-off 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

<u>Parameter</u>	<u>Description</u>
m	The offset distance (inch or mm) $-1 \leq m \leq 1(\text{inch})$

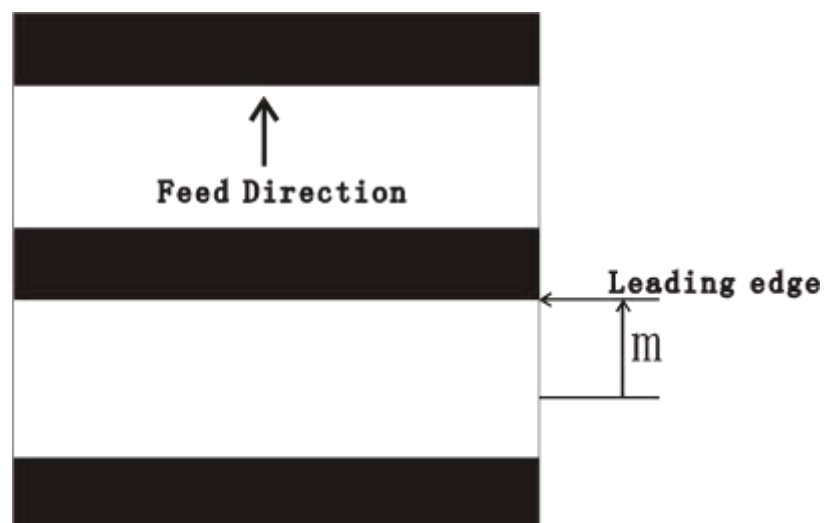
CAUTION: Improperly offset value may cause paper jam.

### Example

- (1) English system (inch)  
OFFSET 0.5
- (2) Metric system (mm)  
OFFSET 12.7 mm

### See Also

SIZE, GAP, SET PEEL





## 5. SPEED

### Description

This command defines the print speed.

### Syntax

SPEED n

<u>Parameter</u>	<u>Description</u>
n	printing speed in inch per second

### Example

SPEED 10

### See Also

DENSITY

## 6. DENSITY

### Description

This command designates the level of darkness of printing.

### Syntax

DENSITY n

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

### Example

DENSITY 7

### See Also

DENSITY

## 7. DIRECTION and Mirror Image

### Description

This command defines the printout direction and mirror image. And this will be memorized in EEPROM.

### Syntax

DIRECTION n[,m]

#### Parameter

n  
m

#### Description

0 or 1. Please refer to the illustrations below:  
0: Print normal image. 1: Print mirror image.

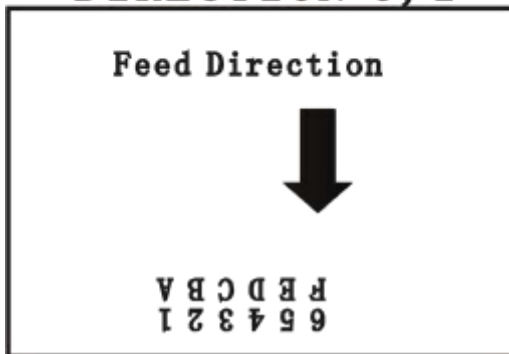
#### DIRECTION 0, 0



#### DIRECTION 1, 0



#### DIRECTION 0, 1



#### DIRECTION 1, 1



### Example

DIRECTION 0[,0]

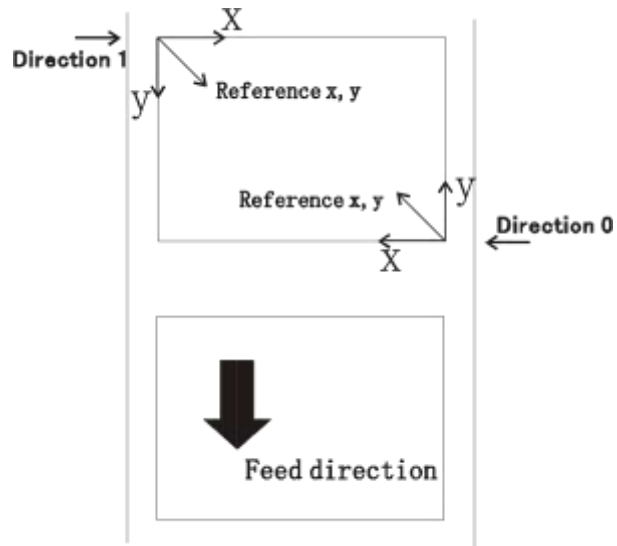
### See Also

REFERENCE

## 8. 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

x

y

#### Description

Horizontal coordinate, with “dot” as the unit.

Vertical coordinate, with “dot” as the unit.

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

### Example

REFERENCE 10,10

### See Also

DIRECTION

## 9. SHIFT

### Description

This command can be used to fine-tune the entire label up or down from its current position. The position relates to the top edge of the label. A negative value moves the entire label away from the top of the label; a positive value moves the entire label to the top of label.

### Syntax

SHIFT n

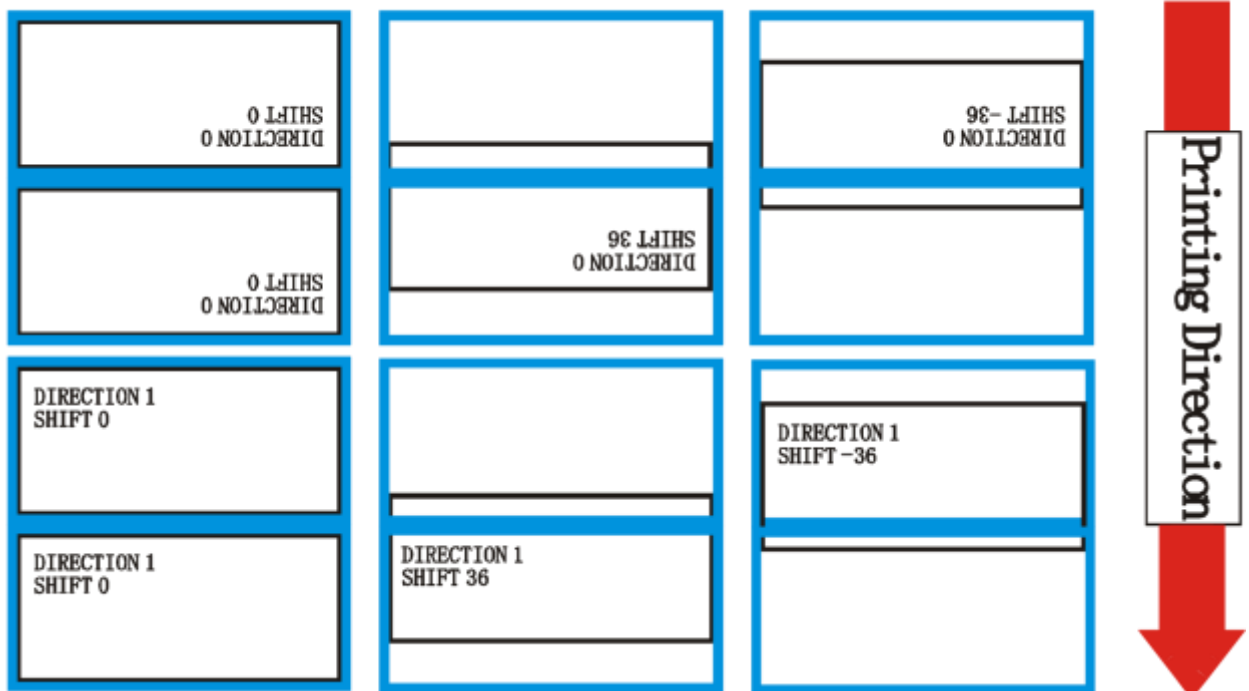
#### Parameter

n

#### Description

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

### Example



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

### See Also

OFFSET, REFERENCE

## 10. CODEPAGE

### Description

This command defines the code page of international character set.

### Syntax

CODEPAGE n

#### Parameter

n

#### Description

name or number of code page, which can be divided into 8-bit code page further.

#### 8-bit code page number

437: United States

850: Multilingual

852: Slavic

860: Portuguese

863: Canadian/French

865: Nordic

857: Turkish

#### Windows code page

1250: Central Europe

1252: Latin I

1253: Greek

1254: Turkish

*Note: 8-bit code page is determined by the communication parameter of DATA LENGTH*

### Example

CODEPAGE 437

### See Also

SET COM1, ~!I

## 11. CLS

### Description

This command clears the image buffer.

### Syntax

CLS

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

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

### Example

CLS

### See Also

SIZE, GAP

## 12. FEED

### Description

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

### Syntax

FEED n

<u>Parameter</u>	<u>Description</u>
n	unit: dot $1 \leq n \leq 9999$

### Example

FEED 40

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

### See Also

BACKFEED, SIZE, GAP, HOME, FORMFEED

## 13. BACKFEED & BACKUP

### Description

To back feed label with the specified length. The length is specified by dot.

### Syntax

BACKUP n

BACKFEED n

<u>Parameter</u>	<u>Description</u>
n	unit: dot $1 \leq n \leq 9999$

### Example

BACKUP 40

BACKFEED 40

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

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

### See Also

FEED, SIZE, GAP, HOME, FORMFEED

## 14. FORMFEED

### Description

This command feeds label to the beginning of next label.

### Syntax

FORMFEED

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

### Example

```
SIZE 3,2.5
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
CLS
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, , HOME, BACKFEED



## 15. 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 label to the beginning of next label. The size and gap of the label should be setup in advance.

### Syntax

HOME

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

### Example

```
SIZE 3,2.5
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, FORMFEED

## 16. PRINT

### Description

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

### Syntax

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

#### Parameter

m

#### Description

Specifies how many sets of labels will be printed.  
 $1 \leq m \leq 999999999$

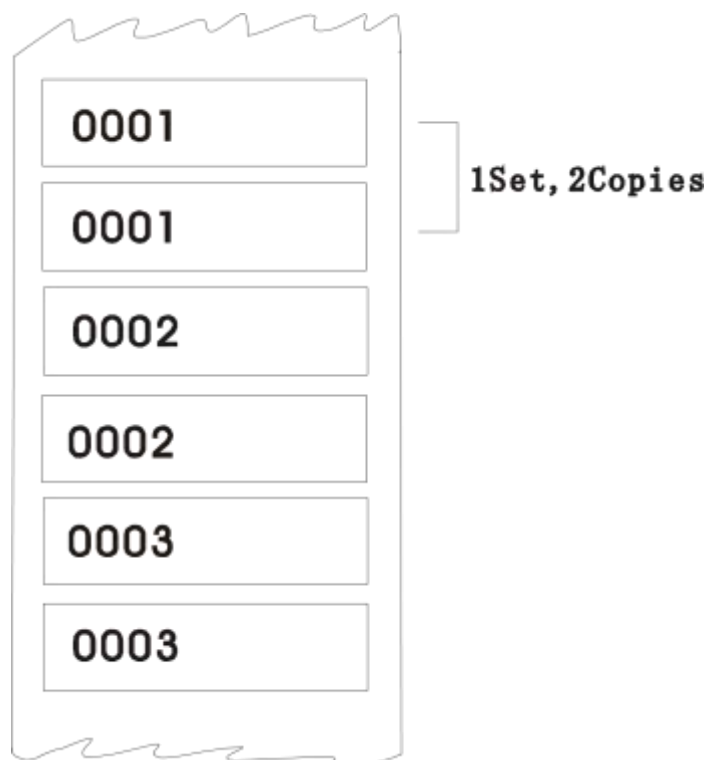
If m=-1, printer will print the last label content for n copies.

n

Specifies how many copies should be printed for each set of label.  
 $1 \leq n \leq 999999999$

### Example

```
SIZE 60 mm, 20 mm  
SET COUNTER @1 1  
@1="0001"  
CLS  
TEXT 10,10,"3",0,1,1,@1  
PRINT 3,2  
PRINT -1,2
```



### See Also

SET COUNTER, DOWNLOAD

## 17. 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

level

interval

#### Description

Sound level: 0~9

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

## 18. LIMITFEED

### Description

When feeding labels, if the gap sensor is not set to a suitable sensitivity, the printer will not be able 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.

### Syntax

LIMITFEED n (inch, the English system)

LIMITFEED n mm (mm, the metric system)

<u>Parameter</u>	<u>Description</u>
n	inch or mm

### Remark

The setting will remain resident in memory.

The default value is 10 inches when printer initializes.

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

## 19. SELFTEST

### Description

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

### Syntax

SELFTEST

### Example

SELFTEST

# Label Formatting Commands

## 20. BAR

### Description

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

### Syntax

BAR x, y, width, height

<u>Parameter</u>	<u>Description</u>
x	The upper left corner x-coordinate in 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

**Recommended max. bar height is 12mm at 3" width. Bar height over than 12 mm may damage the power supply and affect the print quality.**

**Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.**

### Example

```
SIZE 3,2.5  
GAP 0,0  
SPEED 6  
DENSITY 8  
DIRECTION 0  
CLS  
BAR 100, 100, 300, 200  
PRINT 1,1
```



### See Also

BOX

## 21. BARCODE

### Description

This command is used to print 1D barcodes on label form. The available bar codes 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 full ASCII
- Code 39 full ASCII 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

x

y

code type

128

128M

#### Description

Specify the x-coordinate of the bar code on label

Specify the y-coordinate of the bar code on label

Code 128, switching code subset A, B, C automatically

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 Auto switch full ASCII and standard code 39 for PLUS models.
- 39C Code 39 full ASCII with check digit  
Code 39 standard with check digit  
Auto switch full ASCII and standard code 39 for **PLUS** models.
- 39S Code 39 standard
- 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 Postnet
- UPCA UPC-A
- UPCA + 2 UPC-A with 2 digits add-on
- UPCA + 5 UPC-A with 5 digits add-on
- UPCE UPC-E
- UPCE + 2 UPC-E with 2 digits add-on
- UPCE + 5 UPC-E with 5 digits add-on
- CPOST China post code
- MSI MSI code
- MSIC
- 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	width of narrow element in dot
wide	width of wide element in dot

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



code number		the maximum number of digits of bar code content	
Barcode type	Maximum bar Code length	Barcode type	Maximum bar Code length
128	—	POST	5,9,11
EAN128	—	UPCA	11
25	—	UPCA + 2	13
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 128 function character FNC3 which is then followed by the ABCD characters and EFGH characters encoded as CODE A subset.

## 22. BITMAP

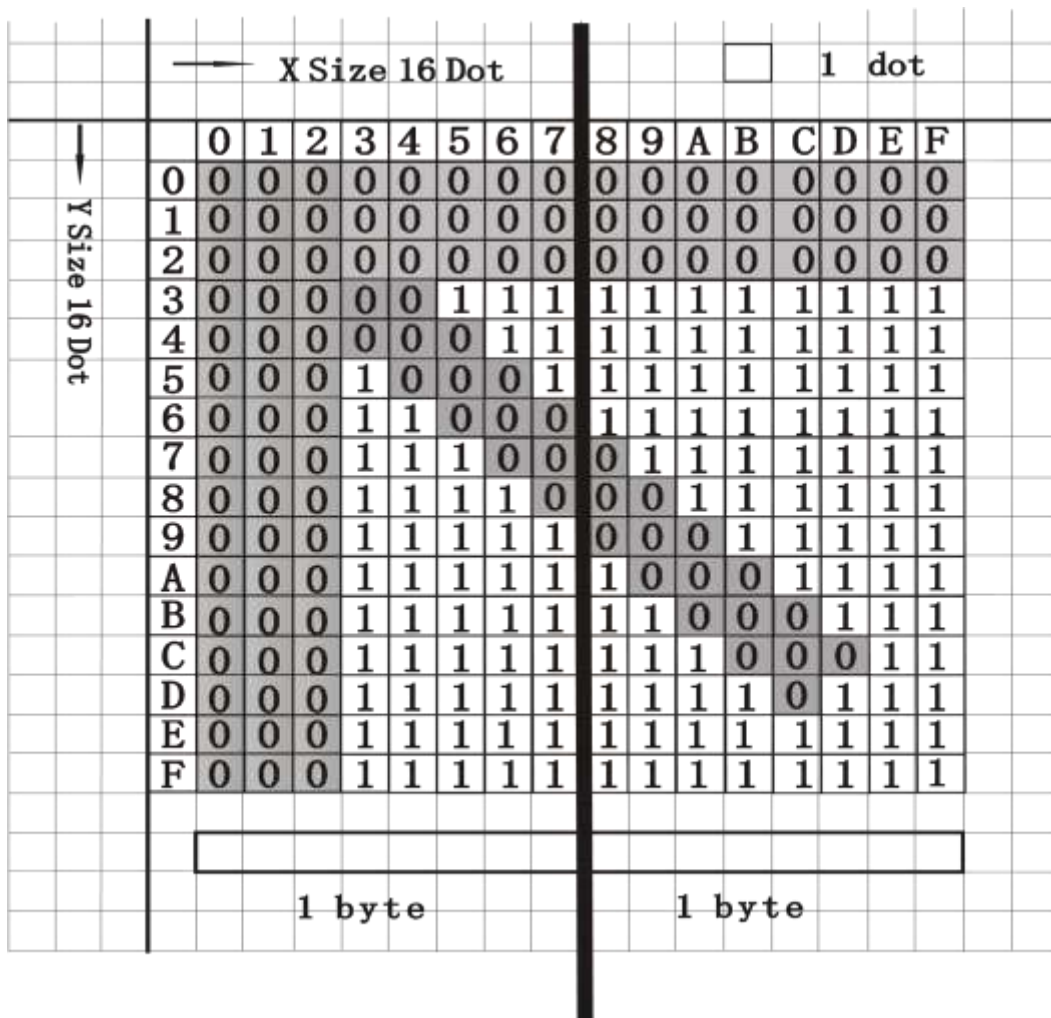
### Description

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

### Syntax

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

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



ROW (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

**Ex:**

SIZE 3,2  
 GAP 0,0  
 CLS  
 BITMAP 200,200,2,16,0,  
 PRINT 1,1

**See Also**

PUTBMP, PUTPCX

## 23. BOX

### Description

This command is used to draw rectangles on the label.

### Syntax

BOX X\_start, Y\_start, X\_end, Y\_end, line thickness

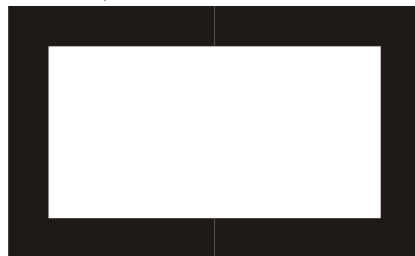
<u>Parameter</u>	<u>Description</u>
X_start	Specify x-coordinate of upper left corner in dot
Y_start	Specify y-coordinate of upper left corner in dot
X_end	Specify x-coordinate of lower right corner in dot
Y_end	Specify y-coordinate of lower right corner in dot
line thickness	Line thickness of the box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

**Note:** *200 DPI: 1 mm = 8 dots*  
*Recommended max. thickness of box is 12mm at 3" width.*  
*Thickness of box that is larger than 12 mm may damage the power supply and affect the print quality.*  
*Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.*

### Example

```
SIZE 3,2.5  
GAP 0,0  
SPEED 6  
DENSITY 8  
DIRECTION 0  
CLS  
BOX 100,100,200,200,5  
PRINT 1,1
```

(100, 100)



(200, 200)

### See Also

BAR

## 24. ERASE

### Description

This command is used to clear a specified region in image buffer.

### Syntax

ERASE X\_start, Y\_start, X\_width, Y\_height

#### Parameter

X\_start

Y\_start

X\_width

Y\_height

#### Description

The x-coordinate of the starting point in dot

The y-coordinate of the starting point in dot

The region width in x-axis direction in dot

The region height in y-axis direction in dot

### Example

SIZE 3,2.5

GAP 0,0

SPEED 6

DENSITY 8

DIRECTION 0

CLS

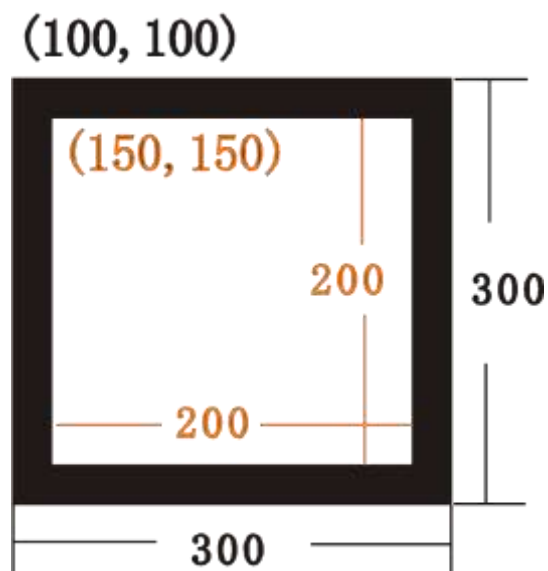
BAR 100, 100, 300, 300

ERASE 150,150,200,200

PRINT 1,1

### See Also

CLS



## 25. PUTBMP

### Description

This command is used to print BMP format image.

### Syntax

PUTBMP X, Y, "filename"

#### Parameter

x

y

filename

#### Description

The x-coordinate of the BMP format image

The y-coordinate of the BMP format image

The downloaded BMP filename.

### Example

```
G:\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

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

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

G:\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.
G:\BMP-PCX>_
```

### See Also

DOWNLOAD, BITMAP, PUTPCX

## 26. PUTPCX

### Description

This command is used to print PCX format image.

### Syntax

PUTPCX X, Y, "filename"

<u>Parameter</u>	<u>Description</u>
X	The x-coordinate of the PCX image
y	The y-coordinate of the PCX 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 G:\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

## 27. QR CODE

### Description

Create QR CODE 2D barcode

### Syntax

QR CODE X, Y, ECC Level, cell width, mode, rotation, [model, mask,]"Data string"

<u>Parameter</u>	<u>Description</u>
x	QR CODE Barcode upper left corner X coordinate
y	QR CODE Barcode upper left corner Y coordinate
ECC level	Error correction capability level
L	7%
M	15%
Q	25%
H	30%
cell width	1~10
mode	Automatically generate code / manually generate code
A	Auto
M	Manual
rotation	Turn the angle clockwise
0	No rotate
90	Rotate 90 degrees clockwise
180	Rotate 180 degrees clockwise
270	Rotate 270 degrees clockwise
model	Barcode generation style
1	(Default), original version
2	Expanded version
mask	Range: 0 ~ 8, default 7
Data string	Barcode information content

Available encoded character set:

- (1) Numeric data: number 0 ~ 9
- (2) Alphanumeric data: number 0 ~ 9; capital letters A-z; other: space, \$% \* + - /:
- (3) 8-bit binary data (JIS 8-bit character table (Latin and kana) matches JIS X 0201)
- (4) Japanese characters (Shift JIS values 8140<sub>HEX</sub> -9FFC<sub>HEX</sub> and E040<sub>HEX</sub> -EAA4<sub>HEX</sub>. These are values shifted from those of JIS X 0208. Refer to JIS X 0208 Annex 1 Shift Coded Representation for detail.).

The maximum length of the bar code:

	<u>Model 1 (Version 14-L)</u>	<u>Model 2 (Version 40-L)</u>
(1) Numerical data:	1,167 characters	7,089 characters
(2) Digital data:	707 characters	4,296 characters
(3) 8-bit binary data:	486 characters	2,953 characters
(4) Japanese Chinese characters:	299 characters	1,817 characters

In the manually generated code:

1. If the first character of the barcode content is "A", the subsequent data is of the "text" type.
2. If the first character of the barcode content is "N", the subsequent data is of the "numeric" type.
3. If the first character of the barcode content is "B"  
The subsequent four digits represent the length of the binary data (In bytes) followed by the "binary data" type.
4. If the first character of the barcode content is "K", the subsequent data is "Japanese Kanji".
5. "!" Is used to convert data "N", "A", "B", "K" and so on  
When you convert with "!", You can convert the data type to a set of barcode content.



## Example

Example of manually generating QRCODE:

```
QRCODE 100,10,L,7,M,0,1,1,"ATHE FIRMWARE HAS BEEN UPDATED"
```

(A: text digital mixed type of information)

```
QRCODE 100,10,M,7,M,0,1,2,"N123456"
```

(N: numerical data)

```
QRCODE 100,10,Q,7,M,0,1,3,"N123456!ATHE FIRMWARE HAS BEEN UPDATED"
```

(N: digital data; data type conversion characters; A; text mixed digital data) QRCODE

```
100,10,H,7,M,0,1,3,"B0012Product name"
```

(B: binary data; 0012: 12 bytes)

```
QRCODE 100,10,M,7,M,0,M1,S3,"K Printer "
```

(K: Japanese characters)

Example for generate QRCODE automatically:

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

(1) Example of automatic generation of QRCODE:

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,"barcode printer ABCabc123"
```

```
PRINT 1,1
```

b. The data string contains <Enter> (0Dh, 0Ah)

```
SIZE 4,2.5
```

```
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,"barcode printer<Enter>
```

```
ABC<Enter> abc<Enter>
```

```
123"
```

```
PRINT 1,1
```

c. Data concatenation (must be in DOENLOAD ... EOP mode)

```
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,"barcode printer ABCabc123"+"1234"+"abcd"
```

```
PRINT 1,1 EOP
```

```
DEMO
```

d. When the content contains double quotation marks ("), replace it with \["]

```
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,"\["]barcode
printer\["]ABCabc123" PRINT 1,1

```

(2) Example of manually generating QRCODE:

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          barcode
printer!AABC!B0006abc123" PRINT 1,1

```

b. The data string contains <Enter> (0Dh, 0Ah) and <Enter> is 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 barcode printer!B0002<Enter>
!AABC!B0010<Enter> abc<Enter>
123"
PRINT 1,1

```

c. Data concatenation (must be in DOENLOAD ... EOP mode)

```

DOWNLOAD "A.BAS"
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          barcode
printer!AABC!B0014abc123"+"1234"+"abcd"
PRINT 1,1
EOP
A

```

d. When the content contains double quotation marks ("), replace it with \["]

```

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 barcode printer!B0010\["]ABCabc123"
PRINT 1,1

```

## 28. REVERSE

### Description

This command is used to reverse a region in image buffer.

### Syntax

REVERSE X\_start, Y\_start, X\_width, Y\_height

<u>Parameter</u>	<u>Description</u>
X_start	The x-coordinate of the starting point in 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

**Note:** *200 DPI: 1 mm = 8 dots*  
*Recommended max. height of reversed black area is 12mm at 3" width. Height of reversed area that is larger than 12 mm may damage the power supply and affect the print quality.*  
*Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.*

### Example

```
SIZE 3,2.5  
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,
```



## 29. TEXT

### Description

This command is used to print text on label

### Syntax

TEXT X, Y, "font", rotation, x-multiplication, y-multiplication, "content"

<u>Parameter</u>	<u>Description</u>
X	The x-coordinate of the text
Y	The y-coordinate of the text
font	Font name
0	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable
1	8 x 12 fixed pitch dot font
2	12 x 20 fixed pitch dot font
3	16 x 24 fixed pitch dot font
4	24 x 32 fixed pitch dot font
5	32 x 48 dot fixed pitch font
6	14 x 19 dot fixed pitch font OCR-B
7	21 x 27 dot fixed pitch font OCR-B
8	14 x25 dot fixed pitch font OCR-A
TST24.BF2	Traditional Chinese 24 x 24 font
TSS24.BF2	Simplified Chinese 24 x 24 font (GB)
K	Korean 24 x 24 font (KS)
Rotation	The rotation angle of text
0	0 degree
0	90 degrees, in clockwise direction
180	180 degrees, in clockwise direction
270	270 degrees, in clockwise direction
X-multiplication:	Horizontal multiplication, up to 10x. Available factors: 1~10 width (point) of true type font. 1 point=1/72 inch.
Y-multiplication:	Vertical multiplication, up to 10x. Available factors: 1~10 For true type font, this parameter is used to specify the height (point) of true type font. 1 point=1/72 inch.

#### Note:

1. *If there is any double quote (") within the text, please change it to \["].*
2. *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.*

### Example

```
SIZE 3,2
GAP 0,0
CLS
TEXT 100,100,"5",0,1,1,"["]DEMO FOR TEXT\[["]
TEXT 100,200,"0,1,20,"["]True Type Font Test Print\[["]
PRINT 1,1
```

## Status Polling Commands (RS-232)

### 30. <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.

<u>Bit</u>	<u>Status</u>
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)

Hex Receive	Printer Status
00	Normal
01	Head opened
02	Paper Jam
03	Paper Jam and head opened
04	Out of paper
05	Out of paper and head opened
08	Out of ribbon
09	Out of ribbon and head opened
0A	Out of ribbon and paper jam

0B	Out of ribbon, paper jam and head opened
0C	Out of ribbon and out of paper
0D	Out of ribbon, out of paper and head opened
10	Pause
20	Printing

### Syntax

<ESC>!?

### See Also

<ESC>!R

## <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

N/A

#### Description

N/A

### See Also

<ESC>!?

## 31. ~!@

### 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 return). It is returned to PC in ASCII characters. The ending character of mileage is 0x0D.

### Syntax

~!@

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

N/A

N/A

### Example

~!@

## 32. ~!A

### Description

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

### Syntax

~!A

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

N/A

N/A

### Example

~!A

### See Also

FILES

## ~!C

### Description

Through the serial port back to the printer whether to install RTC

### Syntax

~!C

<u>Return value</u>	<u>Description</u>
---------------------	--------------------

0

1

RTC is not installed

RTC installed

### Example

~!C

### 33. ~!D

#### Description

This command is used to enter DUMP mode.

#### Syntax

~!D

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

#### Example

~!D

### 34. ~!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 is 0x1A.

#### Syntax

~!F

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

#### Example

~!F

#### See Also

FILES



## 35. ~!I

### Description

The command is used to inquire the code page setting of the printer. The returned information is given in the following format

codepage,code  
ex: 8 bit: 437, 001

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

### Syntax

~!I

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

### Example

~!I

### See Also

CODEPAGE

## 36. ~!T

### Description

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

### Syntax

~!T

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

### Example

~!T

### See Also

~!I, ~!F

# File Management Commands

## 37. 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 to two categories: program file and data file (including text data file, PCX graphic files and bitmap font file). The detailed descriptions regarding the download syntax for different files are shown below:

Maximum numbers of file saved in DRAM:

256 files.

Maximum numbers of file saved in Flash memory:

256 files

Printers will execute the program automatically when turning on power, if one of the downloaded programs is named “AUTO.BAS”. To disable the auto execution function, please follow the procedures below.

### Syntax

#### 1. Download a program file

DOWNLOAD [n,]“FILENAME.BAS”

<u>Parameter</u>	<u>Description</u>
n	Specify the memory which is used to save the download files.
n is ignored	Download files to DRAM only. If you would like to save the files from DRAM to Flash memory before turning off power, you can issue MOVE command to printer to move the files from DRAM to Flash memory. F: Download files to main board flash memory. E: Download files to expansion memory module.

FILENAME.BAS      The filename resident in printer memory.

#### *Note:*

*The filename is case sensitive.*

- (1) The extension of the program file must be “.BAS”*
- (2) Filename format must be in 8.3 format.*
- (3) If memory is not specified, all files will be downloaded to*
- (4) DRAM.*

*No Battery is used to backup DRAM. The downloaded files in DRAM will be lost in case turns off printer power.*

## 2. Download a data file

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

<u>Parameter</u>	<u>Description</u>
n	Specify the memory location to save the download files.
n is ignored	Download files to DRAM only. If you would like to save the files from DRAM to Flash memory before turning off power, you can issue MOVE command to printer to move the files from DRAM to Flash memory. F: Download files to main board flash memory. E: Download files to expansion memory module.

### *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 backup DRAM. The downloaded files in DRAM will be lost in case turns off printer power.*

### **Example**

The program listed below will download to printer SDRAM.

```
DOWNLOAD "EXAMPLE.BAS"
```

```
SIZE 3,4
```

```
GAP 0,0
```

```
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, you can call the main filename without BAS extension or use RUN command to start the download program.*

### *Example:*

- 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 keyboard "ENTER" key. In the above example, please press "ENTER" key instead of typing <ENTER> in the above example.

### See Also

EOP, RUN, PUTBMP, PUTPCX,

## 38. 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" statement at the last line of program.

### Syntax

EOP

### Example

```
DOWNLOAD "DEMO.BAS"  
SIZE 3,4  
GAP 0,0  
DENSITY 8  
SPEED 6  
DIRECTION 0  
REFERENCE 0,0  
  
SET PEEL OFF  
CLS  
TEXT 100,100,"3",0,1,1,"DEMO PROGRAM"  
PRINT 1  
EOP
```

### See Also

DOWNLOAD, EOP,

## 39. FILES

### Description

This command prints out the filenames (or lists the files through RS-232) that remained in the printer memory (both FLASH memory and DRAM). The total memory size and available memory size are printed out or lists as well.

### Syntax

FILES

### Example

Follow the steps below to print out (or lists the files through RS-232) the files that are saved in printer memory in DOS environment through serial port or parallel port connection.

```
C:\>MODE COM1 96,N,8,1<ENTER>
C:\>COPY CON COM1<ENTER>
    FILES<ENTER>
    <CTRL><Z><ENTER>
C:\>
```

Or

```
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

## 40. KILL

### Description

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

### Syntax

KILL [n], "FILENAME"

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

**Note:** *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"
5. KILL E, "\* .PCX"

### 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

## 41. MOVE

### Description

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

### Syntax

MOVE

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

### See Also

DOWNLOAD, EOP

## 42. RUN

### Description

This command is used to execute a program that resident in printer memory

### 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

## 43. 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 3,4  
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



## 44. ASC()

### Description

This function returns the ASCII code of the character.

### Syntax

ASC ("A")

### Example

```
DOWNLOAD "TEST.BAS"  
SIZE 3,4  
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\$()

## 45. CHR\$( )

### Description

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

### Syntax

CHR\$(n)

<u>Parameter</u>	<u>Description</u>
n	The ASCII code

### Example

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

### See Also

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

## 46. END

### Description

This command states the ending of program.

### Syntax

END

### Example

```
DOWNLOAD "DEMO.BAS"  
SIZE 3,2  
GAP 0,0  
DENSITY 8  
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

## 47. EOF()

### Description

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

### Syntax

EOF (File Handle)

<u>Parameter</u>	<u>Description</u>
File handle	Either 0 or 1
<u>Parameter</u>	<u>Description</u>
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
TEXT 10,Y,"2",0,1,1,ITEM$+"$"+STR$(P)+"[EOF(0)="+STR$(EOF(0))+"]"
BARCODE 10,Y+25,"39",40,1,0,2,4,"PRICE-"+STR$(P)
Y=Y+100
IF EOF(0)=0 THEN GOTO A
PRINT 1
EOP
DEMO
```

### See Also

DOWNLOAD, EOP, OPEN, READ, SEEK

## 48. 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

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

```
C:\>COPY DATA/B LPT1
```

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 <under MS-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, FREADS()

## 49. READ

### Description

This command is used to read data from downloaded data file

### Syntax

READ file handle, variables

<u>Parameter</u>	<u>Description</u>
File handle	0 or 1.
variables	string, integer or float point variable

### Example

```
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$(I+1)
ITEMS$=""
READ I,ITEMS$,P,Q
TEXT 10,Y,"2",0,1,1,ITEMS$+"$"+STR$(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, FREADS()

## 50. SEEK

### Description

This command is used to shift the specified file pointer to a certain position.

### Syntax

SEEK file handle, offset

#### Parameter

File handle  
offset

#### Description

0 or 1.  
the offset characters which are shifted  
to the beginning of 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
```

## See Also

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

## 51. LOF()

### Description

This function returns the size of the specified file.

### Syntax

LOF ("FILENAME")

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

### Example

```
DOWNLOAD "DATA1",10,1234567890
```

```
DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO
```

```
DOWNLOAD "LofTest.BAS"
```

```
SIZE 3,3
```

```
GAP 0.08,0
```

```
DENSITY 8
```

```
SPEED 3
```

```
DIRECTION 0
```

```
REFERENCE 0,0
```

```
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\$()

## 52. FREAD\$( )

### Description

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

### Syntax

FREAD\$( file handle, byte)

<u>Parameter</u>	<u>Description</u>
file handle	Either
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

## 53. 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 39 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; start < end
NEXT
```

<u>Parameter</u>	<u>Description</u>
Variable	The variable name is up to 8 characters
start	Can be integer or floating point numbers
end	Can be integer or floating point numbers
increment	integer or floating point, positive or negative.

### Example

```
DOWNLOAD "LOOP.BAS"
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

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

### Description

Use IF...THEN block to execute one or more statements conditionally.  
You can use either a single-line syntax or multiple-line "block" syntax:

### 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

condition

statement

### Description

Available relational operator: <, >, =, <=, >=

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

```

DOWNLOAD "IFTHEN.BAS"

SIZE 3,4  
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\$  
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

**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 present, it will be executed. Otherwise the next line of statement is executed.*

*If there are block of statements in IF...THEN ...ELSE, ENDIF must be used at the end of the IF...THEN ...ELSE statement. Limitations:*

*The total numbers of nested IF ...THEN ...ELSE statement in a program cannot exceed than 40.*

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

**See Also**

DOWNLOAD, EOP

## 55. 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

LABEL

#### Description

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  
RETURN EOP  
GOSUB1
```

### See Also

DOWNLOAD, EOP, END, GOTO



## 56. GOTO

### Description

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

### Syntax

GOTO LABEL

:LABEL

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

### 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

## 57.REM

### Description

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

### Syntax

REM

### Example

```
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

## 58. INT()

### Description

This function is used to truncate a floating point number.

### Syntax

INT (n)

#### Parameter

n

#### Description

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

### Example

```
DOWNLOAD "DEMO.BAS"
SIZE 3,2
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\$()

## 59. LEFT\$()

### Description

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

### Syntax

LEFT\$ (X\$, n)

<u>Parameter</u>	<u>Description</u>
X\$	The string to be processed
n	The number of characters to be returned

### Example

```
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\$()

## 60. LEN()

### Description

This function returns the length of a string.

### Syntax

LEN (string)

<u>Parameter</u>	<u>Description</u>
string	The string whose length is to be measured.

### Example

```
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()

## 61. 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)

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

## 62. RIGHTS()

### Description

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

### Syntax

RIGHT\$(X\$, n)

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

### Example

```
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()

## 63. STR\$( )

### Description

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

### Syntax

STR\$(n)

<u>Parameter</u>	<u>Description</u>
n	An integer, floating point number or mathematical expression

### Example

```
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$  
PRINT 1  
EOP  
DEMO
```

### See Also

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



## 64. VAL()

### Description

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

### Syntax

VAL ("numeric character")

<u>Parameter</u>	<u>Description</u>
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$
PRINT 1
EOP
DEMO
```

### See Also

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

## 65. BEEP

### Description

This command is used to issue a beep sound on portable keyboard.

### Syntax

BEEP

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

### Example

```
DOWNLOAD "DEMO.BAS"  
SIZE 3,4  
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$  
PRINT 1  
EOP
```

# Device Reconfiguration Commands

## 66. SET COUNTER

### Description

Counter can be a real counter or a variable.

This setting sets the counter number in program and their increments. There are three different kind of counters: digit (0~9~0), lower case letter (a~z~a) and upper-case letter (A~Z~A).

### Syntax

```
SET COUNTER @n step
```

@n = "Expression"

#### Parameter

@n

step

Expression

#### Description

n: counter number. There are 51 counters available (@0~@50) in the printer.

The increment of the counter, can be positive or negative.

-999999999<= step <=999999999

If the counter is used as a fixed variable, please set the increment to 0.

Initial string. String length is 101 bytes

### Example

```
SIZE 3,3
```

```
GAP 0,0
```

```
DENSITY 8
```

```
SPEED 6
```

```
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

## 67. 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
```

<u>Parameter</u>	<u>Description</u>
ON	Enable KEY1 as FEED function Enable KEY2 as PAUSE function
OFF	Disable KEY1 as FEED function Disable KEY2 as PAUSE 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 (FEED key) is pressed!!"  
  PRINT 1,1  
ELSEIF KEY2=1 THEN  
  CLS  
  TEXT 100,10,"3",0,1,1,"KEY2 (PAUSE key) is pressed!!"  
  TEXT 100,60,"3",0,1,1,"End of test"  
  PRINT 1,1  
  SET KEY1 ON  
  SET KEY2 ON
```

END  
ENDIF  
GOTO START  
EOP  
DEMO

**See Also**

OFFEST, PRINT

## 68. SET PEEL

### Description

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

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

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

### Syntax

SET PEEL ON/OFF

<u>Parameter</u>	<u>Description</u>
ON	Enable the self-peeling function
OFF	Disable the self-peeling function

### Example

```
REM ***SELF-PEELING FUNCTION ON***  
SIZE 3,4  
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

## 69. SET TEAR & SET STRIPPER

### Description

This command is used to enable/disable feeding label to gap/black mark position for tearing off.

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

### Syntax

SET TEAR ON/OFF

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

### Example

```
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

## 70. SET HEAD

### Description

This setting is used to enable/disable head open sensor. If head open sensor is closed then when printer head is opened there isn't any message returned. This setting will be saved in printer memory.

### Syntax

SET HEAD ON /OFF

<u>Parameter</u>	<u>Description</u>
ON	Turn on the "HEAD OPEN" sensor
OFF	Turn off the "HEAD OPEN" sensor

### Example

```
SET HEAD ON
SET HEAD OFF
```

## 71. SET RIBBON

### Description

Set the on / off RIBBON sensor, ie to switch the Thermal transfer / Thermal Direct  
Usually when the printer is powered on,  
Ribbon sensor will automatically detect whether the printer has been installed on the ribbon, and to decide to use the thermal or thermal Printed.

This setting is not stored in the printer. This method is only applicable to heat transfer machines

### Syntax

SET RIBBON ON /OFF

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

### 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
```

## 72. SET COM1

### Description

This setting defines communication parameters for printer serial port.

### Syntax

SET COM1 baud, parity, data, stop

<u>Parameter</u>	<u>Description</u>
baud	Baud rate, available baud rates are as listed: 24: 2400 bps 48: 4800 bps 96: 9600 bps 19: 19200 bps
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

The parallel port is used to setup the printer serial port in this example by 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.



## 73. SET PRINTKEY

### Description

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

### Syntax

SET PRINTEKY OFF/ON/AUTO/<num>

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

### Example

#### Execute:

```
SIZE 3,2.5  
GAP 0.12,0  
SET PRINTKEY ON  
SET COUNTER @0 1  
@0="0001"  
CLS  
TEXT 10,10,"5",0,1,1,@0  
PRINT 1
```

**Execute:**

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

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

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

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

Syntax	Receive "PRINT m,n"	Print Out
SET PRINTKEY 5	1.) PRINT 1,2	Label 1, Label 1
	2.) Press FEED key	Label 2~6

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

## 74. SET REPRINT

### Description

This command will disable/enable reprint the label when the “no paper” or “no ribbon” or “carriage open” error is occurred.

### Syntax

SET REPRINT OFF/ON

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

### Example

SET REPRINT ON

## 75. PEEL

### Description

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

### Syntax

PEEL

<u>Parameter</u>	<u>Description</u>
0	Paper is not on top of peel sensor
1	Paper is on top of peel sensor

### Example

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

## 76. KEY1, KEY2

### Description

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

### Syntax

KEYm=n

<u>Key</u>	<u>Return Value</u>
KEY1 (FEED)	0: released 1: pressed
KEY2 (PAUSE)	0: released 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  
GOTO START EOP  
DEMO
```

# Printer Global Variables

## 77. @LABEL

### Description

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

### Syntax

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

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

### Example

```
DOWNLOAD "DEMO.BAS"
SIZE 3,2.5
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
```