
THERMAL LABEL PRINTER SERIES

PROGRAMMING MANUAL

Thermal Barcode Printer Series

Version	Date	Content
V1.0.0	2016.01.19	Standard
V1.0.1	2016.04.25	Add SET CUTTER and SET RESPONSE command
V1.0.2	2016.04.26	Add ESC command syntax description

CONTENTS

Document Conventions.....	1
1. SIZE.....	2
2. CASHDRAWER.....	2
3. GAP.....	2
4. OFFSET.....	3
5. SPEED.....	4
6. DENSITY.....	4
7. DIRECTION.....	5
8. REFERENCE.....	5
9. COUNTRY.....	5
10. CODEPAGE.....	6
11. CLS.....	7
12. FEED.....	7
13. BACKFEED & BACKUP.....	7
14. FORMFEED.....	7
15. HOME.....	8
16. PRINT.....	8
17. SOUND.....	9
18. LIMITFEED.....	9
Label Formatting Commands.....	9
19. BAR.....	9
20. BARCODE.....	10
21. BOX.....	12
22. BITMAP.....	12
23. PUTBMP.....	14
24. PUTPCX.....	14
24. ERASE.....	15
25. REVERSE.....	15
26. TEXT.....	16
27. QRCODE.....	17
Inquiry for printer status commands.....	17
28. <ESC>!?.....	17
29. <ESC>!R.....	18
30. ~!@.....	18
31. ~!A.....	18
32. ~!D.....	19
33. ~!F.....	19
34. ~!I.....	19
35. ~!T.....	19
File Management Commands.....	19
37. DOWNLOAD.....	19
38. EOP.....	20
39. FILES.....	20
40. KILL.....	21
41. MOVE.....	21
42. RUN.....	21
BASIC Commands and Functions.....	21

43. ABS().....	21
44. ASC().....	22
45. CHR\$().....	22
46. END.....	23
47. EOF().....	23
48. OPEN.....	24
49. READ.....	26
50. SEEK.....	27
51. LOF().....	28
52. FREAD\$().....	28
53. FOR...NEXT LOOP.....	29
54. IF...THEN...ELSE.....	30
55. GOSUB...RETURN.....	31
56. GOTO.....	31
57. REM.....	32
58. INT().....	32
59. LEFT\$().....	33
60. LEN().....	33
61. MID\$().....	34
62. RIGHT\$().....	34
63. STR\$().....	35
64. VAL().....	35
65. BEEP.....	36
66. INPUTFILE.....	36
67. ADJUST.....	37
Device Reconfiguration Commands.....	38
68. SET COUNTER.....	38
69. SET KEY1, SET KEY2.....	38
70. SET PEEL.....	39
71. SET TEAR & SET STRIPPER.....	40
72. SET HEAD.....	40
73. SET COM1.....	40
74. SET PRINTKEY.....	41
75. SET REPRINT.....	42
76. PEEL.....	42
77. KEY1,KEY2.....	42
78. SET CUTTER.....	43
79. SET RESPONSE.....	44
Printer Global Variables.....	44
80. @LABEL.....	44
81. YEAR.....	45
82. MONTH.....	46
83. DATE.....	46
84. WEEK.....	47
85. HOUR.....	48
86. MINUTE.....	49
87. SECOND.....	49
88. @YEAR.....	50
89. @MONTH.....	51

90. @DATE..... 51

91. @DAY.....52

92. @HOUR.....52

93. @MINUTE..... 53

94. @SECOND.....53

Document Conventions

This manual uses the following typographic conventions.

Convention	Description
[expression list]	Items inside square brackets are optional.
<ESC>	ESCAPE (ASCII 27), control code of status polling command, which returns the printer status immediately, no matter the printer is ready or not.
~	(ASCII 126), control code of status polling command, which returns the printer status only when the printer is ready.

Note: 200 DPI: 1 mm = 8 dots

Times New Roman font in bold and italic type is used for note.

Setup and System Commands

1. SIZE

This command defines the label width and length.

Syntax

(1) English system (inch)

SIZE m,n

(2) Metric system (mm)

SIZE m mm,n mm

Parameter	Description
m	Label width (inch or mm, not including back sheet)
n	Label length (inch or mm, not including back sheet)

Note: 200 DPI: 1mm=8dots

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

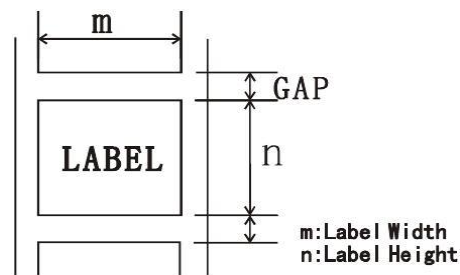
Example

(1) English system (inch)

SIZE 3.5,3.00

(2) Metric system (mm)

SIZE 100mm,100mm



2. CASHDRAWER

Generate pulse

Syntax

CASHDRAWER m,t1,t2

ESC p m,t1,t2

Parameter	Description
m	0, 48 Drawer kick-out connector pin2. 1, 49 Drawer kick-out connector pin5.
t1, t2	$0 \leq t1 \leq 255$, $0 \leq t2 \leq 255$ Outputs the pulse specified by t1 and t2 to connector pin m

Note:

1) The pulse ON time is $[t1 \times 2 \text{ ms}]$ and the OFF time is $[t2 \times 2 \text{ ms}]$.

2) If $t2 < t1$, the OFF time is $[t1 \times 2 \text{ ms}]$.

3. GAP

Define the gap distance between two labels.

Syntax

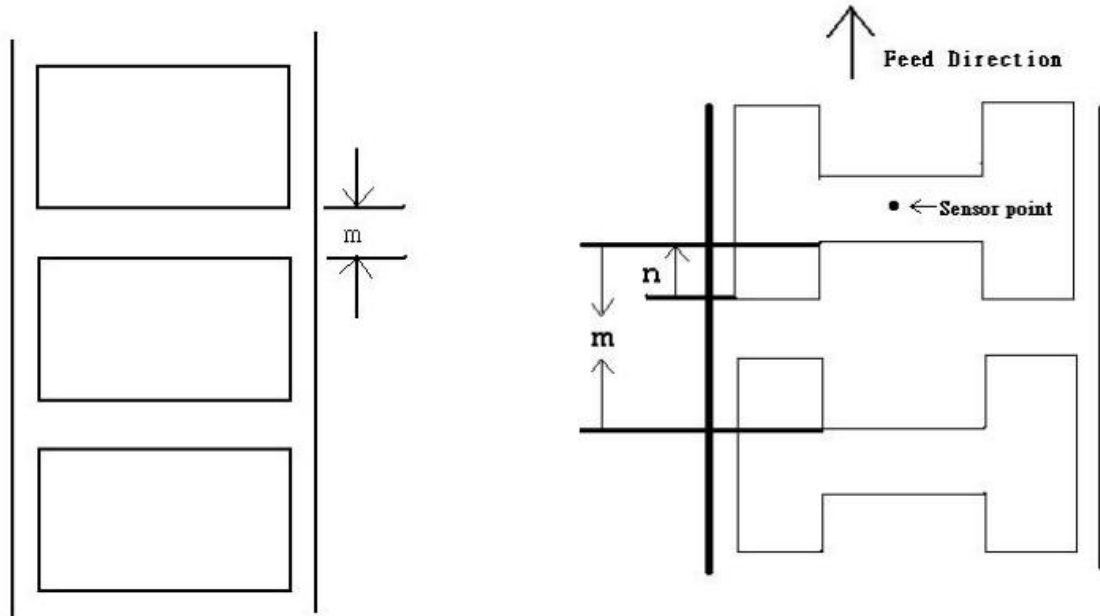
(1) English system (inch)

GAP m,n

(2) Metric system (mm)

GAP m mm,n mm

Parameter	Description
m	The gap distance between two labels $0 \leq m \leq 1(\text{inch}), 0 \leq m \leq 25.4(\text{mm})$
n	The offset distance of the gap $n \leq \text{label length} (\text{inch or mm})$



Note: 200 DPI: 1mm=8dots

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

Example

Normal gap

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

Special gap

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

4. OFFSET

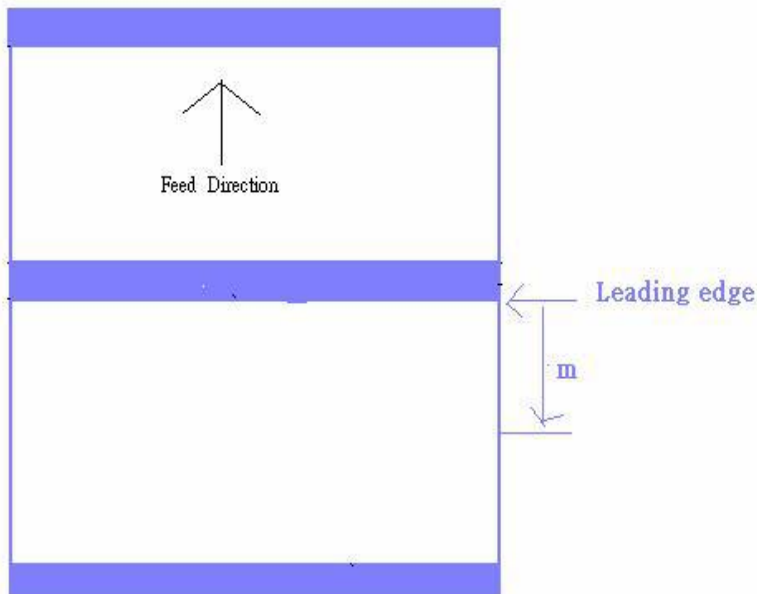
This command defines the selective, extra label feeding length each form feed takes, which, only in peel-off mode, is used to adjust label stop position.

Syntax

- (1) English system (inch)
OFFSET m
- (2) Metric system (mm)
OFFSET m mm

Parameter	Description
m	The offset distance (inch or mm) $0 \leq m \leq 1 (\text{inch}),$

$$0 \leq m \leq 25.4 \text{ (mm)}$$



Note:

Improprity offset value may cause paper jam.

200 DPI: 1mm=8dots

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

Example

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

5. SPEED

This command defines the print speed.

Syntax

SPEED n

Parameter	Description
n	1.5 printing speed in 1.5 inch per second 2.0 printing speed in 2.0 inch per second 3.0 printing speed in 3.0 inch per second 4.0 printing speed in 4.0 inch per second

Example

SPEED 2.0

6. DENSITY

This command designates the level of darkness of printing.

Syntax

DENSITY n

Parameter	Description
-----------	-------------

n	0 ~ 15 0: specifies the lightest level 15: specifies the darkest level
---	--

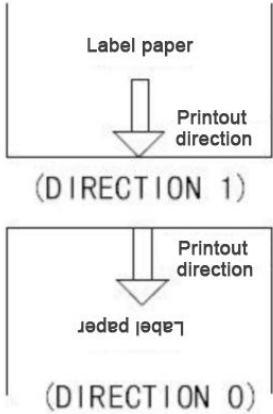
Example
 DENSITY 7

7. DIRECTION

This command defines the printout direction.

Syntax
 DIRECTION n

Parameter	Description
n	0 or 1. Please refer to the illustrations as showing.

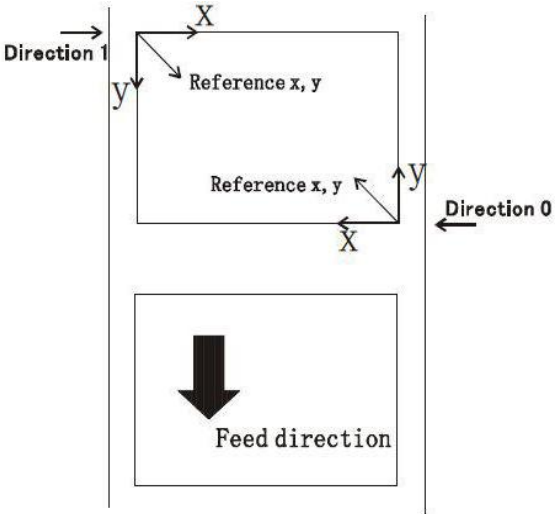


8. REFERENCE

This command defines the reference point of the label.

Syntax
 REFERENCE x,y

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



9. COUNTRY

This command defines the choice of international character set.

Syntax
 COUNTRY n

Parameter	Description
n	001:USA 002:French 003:Latin America 034:Spanish 039:Italian 044:United Kingdom 046:Swedish 047:Norwegian

	049:German
--	------------

Example

COUNTRY 001

10. CODEPAGE

This command defines the code page of international character set.

Syntax

CODEPAGE n

Parameter	Description
n	<p>Name or number of code page, which can be divided into 8-bit code page.</p> <p>437:United States 850:Multilingual 852:Slavic 860:Portuguese 863:Canadian/French 865:Nordic</p> <p>Windows code page</p> <p>1250:Central Europe 1252:Latin I 1253:Greek 1254:Turkish</p> <p><u>The following code pages are only for 12X24dot Alphanumeric</u></p> <p>WestEurope: WestEurope Greek: Greek Hebrew: Hebrew EastEurope: EastEurope Iran: Iran IranII: IranII Latvian: Latvian Arabic: Arabic Vietnam: Vietnam Uygur: Uygur Thai: Thai 1252:Latin I 1257:WPC1257 1251:WPC1251 866:Cyrillic 858:PC858 747:PC747 864:PC864 1001:PC1001</p>

Example

CODEPAGE 437

11. CLS

This command clears the image buffer.

Syntax

CLS

Parameter	Description
N/A	N/A

Note:

This command must be placed after SIZE command.

Example

CLS

12. FEED

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

Syntax

FEED n

Parameter	Description
n	$1 \leq n \leq 9999$ unit: dot

Note:

Impropriety value may cause paper jam or fold.

200 DPI: 1 mm = 8 dots

Example

FEED 40

13. BACKFEED & BACKUP

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

Syntax

BACKFEED n

BACKUP n

Parameter	Description
n	$1 \leq n \leq 9999$ unit: dot

Note:

Impropriety value may cause paper jam or fold.

200 DPI: 1 mm = 8 dots

Example

BACKFEED 40

BACKUP 40

14. FORMFEED

This command feeds label to the beginning of next label.

Parameter	Description
N/A	N/A

Example

FORMFEED

15. HOME

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.

Parameter	Description
N/A	N/A

Note: Label length must be greater than or equal to 30mm when using this command.

Example

HOME

16. PRINT

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

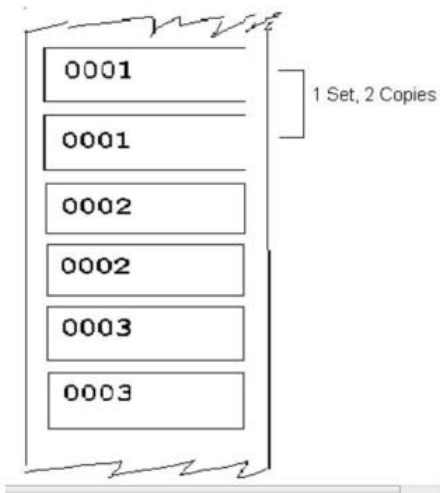
Syntax

PRINT m [,n]

Parameter	Description
m	Specifies how many sets of labels will be printed (set) $1 \leq m \leq 65535$
n	Specifies how many copies should be printed for each particular label set. $1 \leq n \leq 65535$

Example

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



17. SOUND

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 second parameter.

Syntax

SOUND Level interval

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

Example

SOUND 1,300

SOUND 2,400

SOUND 3,600

18. LIMITFEED

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 if the printer does not locate gap after feeding the length of one label plus one preset value.

Syntax

(1) English system (inch)

LIMITFEED n

(2) Metric system (mm)

LIMITFEED n mm

Parameter	Description
n	inch or mm

Note:

(1) . The setting will remain resident in memory.

(2) . The default value is 4 inches when printer initializes.

Example

LIMITFEED 12

LIMITFEED 0.5 mm

Label Formatting Commands

19. BAR

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

Syntax

BAR x,y,width,height

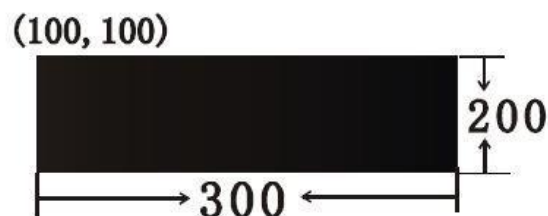
Parameter	Description
x	The upper left corner x-coordinate in dot
y	The upper left corner y-coordinate in dot

width	The width of bar in dot
height	The height of bar in dot

NOTE: 200DPI: 1mm=8dots

Example

BAR 100,100,300,200



20. BARCODE

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

Syntax

BARCODE x,y,"code type",height,human readable,rotation,narrow,wide,"content"

Parameter	Description
x	Specify the x-coordinate of the bar code on label
y	Specify the y-coordinate of the bar code on label
height	Bar code height (in dot)
human readable	0 for human not readable, 1 for 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
code type	Refer to the table1
content	

Table1:

Parameter	Description
128	Code 128, switching code subset A, B, C automatically
128M	Code 128M, switching code subset A, B, C manually, refer to table2
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 full ASCII for TSPL2 printers Code 39 standard for TSPL printers
39C	Code 39 full ASCII with check digit for TSPL2 printers Code 39 standard with check digit for TSPL printers
39S	Code 39 standard for TSPL2 printers
93	Code 93
EAN13	EAN 13
EAN13+2	EAN 13 with 2 digits add-on
EAN13+5	EAN 13 with 5 digits add-on
EAN8	EAN 8
EAN8+2	EAN 8 with 2 digits add-on
EAN8+5	EAN 8 with 5 digits add-on
CODA	Codabar
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

Table2 CODE128M

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	CODE 4	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 as above. If the start subset is not set, the default starting subset is B.

Table3 Narrow wide ratio supported or not

narrow/wide	narrow : wide 1:1	narrow : wide 1:2	narrow : wide 1:3	narrow : wide 2:5	narrow : wide 3:7
128	Y	N	N	N	N
EAN128	Y	N	N	N	N
25	N	Y	Y	Y	N
25C	N	Y	Y	Y	N
39	N	Y	Y	Y	N
39C	N	Y	Y	Y	N
93	N	N	Y	N	N
EAN13	Y	N	N	N	N
EAN13+2	Y	N	N	N	N
EAN13+5	Y	N	N	N	N
EAN8	Y	N	N	N	N
EAN8+2	Y	N	N	N	N
EAN8+5	Y	N	N	N	N
CODA	N	Y	Y	Y	N
UPCA	Y	N	N	N	N
UPCA+2	Y	N	N	N	N
UPCA+5	Y	N	N	N	N
UPCE	Y	N	N	N	N
UPCE+2	Y	N	N	N	N
UPCE+5	Y	N	N	N	N

Table4 code numbers the maximum number of digits of bar code content:

Barcode type	Maximum bar Code length	Barcode type	Maximum bar Code length
--------------	----------------------------	--------------	----------------------------

128	—	CODA	—
EAN128	—	UPCA	11
25	—	UPCA+2	13
25C	—	UPCA+5	16
39	—	UPCE	6
39C	—	UPCE+2	8
93	—	UPCE+5	11
EAN13	12		
EAN13+2	14		
EAN13+5	17		
EAN8	7		
EAN8+2	9		
EAN8+5	12		

Example

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

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

21. BOX

This command is used to draw rectangles on the label.

Syntax

BOX x_start,y_start,x_end,y_end,line thickness

Parameter	Description
x_start	Specify x-coordinate of upper left corner in dot
y_start	Specify y-coordinate of upper left corner in dot
x_end	Specify x-coordinate of lower right corner in dot
y_end	Specify y-coordinate of lower right corner in dot
thickness	Line thickness (in dots)

Example

BOX 100,100,200,200,5

(100, 100)



(200, 200)

22. BITMAP

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

Syntax

BITMAP x,y,width,height,mode,bitmap data

Parameter	Description
x	Specify x-coordinate
y	Specify 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	Hex Bitmap data

	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	1	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

Bitmap data

ROW (Y-axis)	L-Byte		R-Byte	
	Binary	Hexadecimal	Binary	Hexadecimal
0	0000 0000	00	0000 0000	00
1	0000 0000	00	0000 0000	00
2	0000 0000	00	0000 0000	00
3	0000 0111	07	1111 1111	00
4	0000 0011	03	1111 1111	FF
5	0001 0001	11	1111 1111	FF
6	0001 1000	18	1111 1111	FF
7	0001 1100	1C	0111 1111	7F
8	0001 1110	1E	0011 1111	3F
9	0001 1111	1F	0001 1111	1F
A	0001 1111	1F	1000 1111	8F
B	0001 1111	1F	1100 0111	C7
C	0001 1111	1F	1110 0011	E3
D	0001 1111	1F	1111 0111	F7
E	0001 1111	1F	1111 1111	FF

F	0001 1111	1F	1111 1111	FF
---	-----------	----	-----------	----

Example

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

23. PUTBMP

This command is used to print monochrome BMP format image.

Syntax

PUTBMP x,y,"filename"

Parameter	Description
x	The x-coordinate of the BMP format image in dot
y	The y-coordinate of the BMP format image in dot
filename	The BMP filename wanted to print (downloaded into printer buffer already)

Note:

This command supports monochrome BMP graphic only.

```

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.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>_

```

Example

PUTBMP 100,100,"LOGO.BMP"

Note: ^Z means <Ctrl>+<Z> or <F6>

24. PUTPCX

This command is used to print monochrome PCX format image.

Syntax

PUTPCX x,y,"filename"

Parameter	Description
x	The x-coordinate of the BMP format image in dot
y	The y-coordinate of the BMP format image in dot
filename	The BMP filename wanted to print (downloaded into printer buffer already)

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

Directory of C:\BMP-PCX

01/03/2005  01:06 PM    <DIR>          .
01/03/2005  01:06 PM    <DIR>          ..
01/03/2005  01:52 PM                12,430 TSC.bmp
01/03/2005  01:10 PM                1,181 TSC.pcx
                2 File(s)                13,611 bytes
                2 Dir(s)      8,802,189,312 bytes free

C:\BMP-PCX>COPY CON LPT1
DOWNLOAD "TSC.PCX",1181,^Z
        1 file(s) copied.

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

C:\BMP-PCX>COPY CON LPT1
SIZE 4,2.5
GAP 0,0
CLS
PUTPCX 100,100,"TSC.PCX"
PRINT 1,1
^Z
        1 file(s) copied.

C:\BMP-PCX>_
```

Example

PUTPCX 10,10,"TSC.PCX"

Note: ^Z means "<Ctrl>+<Z>" or <F6>

24. ERASE

This command is used to remove partial area image buffer

Syntax

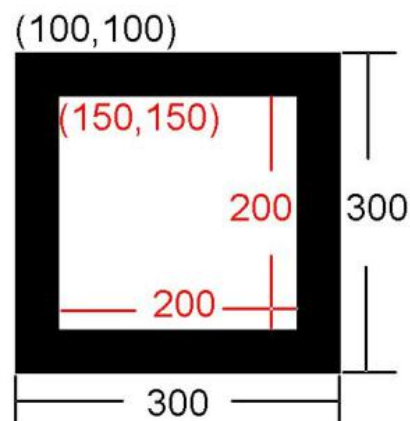
ERASE x_start,y_start,x_width,y_height

Parameter	Description
x_start	The upper left corner x-coordinate in dot
y_start	The upper left corner y-coordinate in dot
x_width	The region width in x-axis direction (in dots)
y_height	The region height in y-axis direction (in dots)

Example

BAR 100,100,300,300

ERASE 150,150,200,200



25. REVERSE

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

Syntax

REVERSE x_start,y_start,x_width,y_height

Parameter	Description
x_start	The upper left corner x-coordinate in dot
y_start	The upper left corner y-coordinate in dot
x_width	The region width in x-axis direction (in dots)
y_height	The region height in y-axis direction (in dots)

Example

REVERSE 100,100,200,200



26. TEXT

This command is used to print text on label

Syntax

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

Parameter	Description
x	The x-coordinate of the text
y	The y-coordinate of the text
font	Font name
1	8×12 dot Alphanumeric
2	12×20 dot Alphanumeric
3	16×24 dot Alphanumeric
4	24×32 dot Alphanumeric
5	32×48 dot Alphanumeric
6	14×19 dot Alphanumeric OCR-B
7	21×27 dot Alphanumeric OCR-B
8	14×25 dot Alphanumeric OCR-A
9	9×17 dot Alphanumeric
10	12×24 dot Alphanumeric
TST24.BF2	Traditional Chinese 24×24 Font (BIG5)
TSS24.BF2	Simplified Chinese 24×24Font (GB)
K	Korean 24×24Font (KS)
rotation	The rotation angle of text (clockwise)
0	No rotation
90	90 degree
180	180 degree
270	270 degree
x-multiplication	Horizontal multiplication, available factors: 1~10
y-multiplication	Vertical multiplication, available factors: 1~10

Note:

Capital letters is printed by the 5th font only.

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

If there is any 0D (hex) character within the text, please change it to \[R] to print CR

If there is any 0A (hex) character within the text, please change it to \[A] to print LF

Example

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

27. QRCODE

This command is used to print QR code

Syntax

QRCODE x,y,ECC level,cell width,mode,rotation,"data string"

Parameter	Description
x	The x-coordinate of the QR code
y	The y-coordinate of the QR code
ECC level	Selects the error correction recovery level for QR Code
L	7%
M	15%
Q	25%
H	30%
cell width	the width of QR code 1~10
mode	Auto/Manual encode
A	Auto
M	Manual
rotation	The rotation angle (clockwise)
0	No rotation
90	90 degree
180	180 degree
270	270 degree
data string	Encoded string

Example

SIZE 60 mm,30 mm

GAP 2 mm

CLS

QRCODE 20,20,L,4,A,0,"www.Gainscha.cn"

PRINT 1,1

Inquiry for printer status commands

28. <ESC>!?

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 returned a byte data to flag the printer's current readiness status through RS-232 even in the event of printer error. If 0x00 is returned, the printer is ready to print labels.

Syntax

<ESC>!?

Hex: 1B 21 3F

Decimal: 27 33 63

Hex receive	Printer status
00	Normal

01	top cover opened
02	Paper jam
03	Paper jam, top cover opened
04	Out of paper,
05	Out of paper, top cover opened
08	Out of ribbon
09	Out of ribbon, top cover opened
0A	Out of ribbon , Paper jam
0B	Out of ribbon , Paper jam, top cover opened
0C	Out of ribbon, out of paper
0D	Out of ribbon, out of paper, top cover opened
10	Pause
20	Printing
80	Other error

Received value can be converted to binary data based on bit, 0 for false, 1 for true.

Bit	Printer status (0 or 1)
0	Printer is not powered off.
1	Paper jam
2	Out of paper
3	Out of ribbon
4	Pause
5	Printing

29. <ESC>!R

This command used to be force printer restart.

Syntax

<ESC>!R

Hex: 1B 21 52

Decimal: 27 33 82

30. ~!@

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) to PC in ASCII characters. The ending character of mileage is 0x0D.

Syntax

~!@

31. ~!A

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

32. ~!D

This command enters the printer into DUMP mode. In DUMP mode, the printer outputs code directly without interpretation.

Syntax

~!D

33. ~!F

This command is used to inquire about files resident in the printer memory and fonts installed in the memory module. The file name is returned in ASCII characters. Each file name ends with 0x0D. The last file name ends with character 0x0D, 0x1A.

Syntax

~!F

34. ~!I

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

Code page, country code

ex: 8 bit: 437, 001

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

Syntax

~!I

35. ~!T

This command is used to inquire the model name.

Syntax

~!T

File Management Commands

37. DOWNLOAD

“DOWNLOAD” is a header of the file that is to be saved in the printer's memory. When execute this command, the data will be saved in the printer's memory, and just call the file-name when need to call. 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) .

Syntax

1. Download program file

Program file as below, after receiving this program file, it will be named EXAMPLE.BAS and stored in DRAM, EOP means the end point of program .

DOWNLOAD "EXAMPLE.BAS"

```
SIZE 3,4
GAP 0,0
DENSITY 8
SPEED 3
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 filename3.BAS**” must be placed in the beginning of file, and “**EOP**” must be placed at the end of program.

2. Download -data file

DOWNLOAD "FILENAME", DATA SIZE, DATA CONTENT...

FILENAME: The name of data file that will save in the printer memory (case sensitive)

DATA SIZE: Actual size of data file, not include the header of the file.

38. EOP

End of program, place at the last line of program.

Syntax

EOP

Example

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

39. FILES

This command prints out the file name resident in printer memory, after setting “SET DEBUG LABEL”.

Syntax

FILES

Example

Please refer to the following steps to print out the file name resident in printer memory.

C:\>COPY CON LPT1 <ENTER>


```
SET DEBUG LABEL <ENTER>
FILE<ENTER>
<CTRL>Z
C:\>
```

Note: <ENTER>, <CTRL>, <CTRL> means the keyboard keys on PC

40. KILL

This command deletes a file in the printer memory.

Syntax

```
KILL "FILENAME"
```

Parameter	Description
FILENAME	The name of data file that will delete in the printer memory (case sensitive)
*	Delete all files resident in printer memory.

Example

```
KILL "DEMO.BAS"
KILL "*.PCX"
KILL "*"
```

41. MOVE

This command moves downloaded files from DRAM to FLASH memory.

Syntax

```
MOVE
```

42. RUN

This command executes a program resident in the printer memory.

Syntax

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

Example

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

BASIC Commands and Functions

43. ABS()

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
```

44. ASC()

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
```

45. CHR\$()

This function returns the character with the specified ASCII code.

Syntax

CHR\$(n)

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
WORD$=CHR$(A)
TEXT 100,100, "3",0,1,1,WORD$
PRINT 1
EOP

```

46. END

This command states the end of program.

Syntax

```
END
```

Example

```

DOWNLOAD "DEMO.BAS"
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 200,60,"4",0,1,1,"END COMMAND TEST"
X=300
Y=200
X1=500
Y1=400
GOSUB DR_LINE
PRINT 1
END
:DR_LINE
FOR I=1 TO 100 STEP 10
BOX X+I,Y+I,X1-I,Y1-I,5
NEXT
RETURN
EOP
DEMO

```

47. EOF()

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

```

48. OPEN

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.

Syntax

OPEN "filename", file handle

Parameter	Description
filename	The file downloaded in the printer memory
file handle	0 or 1

Example

```

DOWNLOAD "DATA1",56,COMPUTER
2000
12
MOUSE

```

500
13
KEYBOARD
300
100

DOWNLOAD "DATA2",56,Computer
3000
32
Mouse
900
93
Keyboard
700
700

DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 4
DIRECTION 1
REFERENCE 0,0
SET CUTTER OFF
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
ENDIF
IF EOF(0)=0 THEN
GOTO Start
ENDIF
END
:OpenData
IF I=1 THEN

```

OPEN "DATA1",0
ENDIF
IF I=2 THEN
OPEN "DATA2",0
ENDIF
SEEK 0,0
IF I>2 THEN
END
ENDIF
I=I+1
RETURN
EOP
DEMO

```

49. READ

This command reads data from downloaded data file.

Syntax

READ file handle, variables

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

Example

```

DOWNLOAD "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 1
REFERENCE 0,0
SET CUTTER OFF
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
I=I+1
GOTO Start
ELSE
END
ENDIF
EOP
DEMO

```

50. SEEK

This command shifts the specified file pointer to a certain position.

Syntax

SEEK file handle,offset

Parameter	Description
file handle	The file downloaded in the printer memory.

Example

```

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

```

51. LOF()

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,ABCDEFGHJKLMNO
DOWNLOAD "LofTest.BAS"
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
OPEN "DATA1",0
OPEN "DATA2",1
TEXT 10,20,"4",0,1,1,"LOF() FUNCTION TEST"
J=LOF("DATA1")
K=LOF("DATA2")
TEXT 10,140,"3",0,1,1,"DATA1 IS: "+STR$(J)+" Bytes"
TEXT 10,200,"3",0,1,1,"DATA2 IS: "+STR$(K)+" Bytes"
PRINT 1
EOP
LofTest
```

52. FREAD\$()

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,ABCDEFGHJKLMNO
DOWNLOAD "OPEN2.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
OPEN "DATA1",0
OPEN "DATA2",1
SEEK 0,0
SEEK 1,0
Y$=FREAD$(0,6)
```



```

Z$=FREAD$(1,6)
TEXT 10,100,"3",0,1,1,"FREAD$(0,6) IS: "+Y$
TEXT 10,150,"3",0,1,1,"FREAD$(1,6) IS: "+Z$
PRINT 1
EOP
OPEN2

```

53. FOR...NEXT LOOP

Loop is used to execute a program automatically until the condition is met. Please do not execute from external jumping to internal directly, otherwise the unexpected error occurred.

Syntax

```

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

Note: the nested loop is not supported.

Example

```

DOWNLOAD "TEST.BAS"
SIZE 4,2.5
GAP 0,0
CLS
FOR I=1 TO 10 STEP 1
TEXT 100,10+30*(I-1),"3",0,1,1,STR$(I)
NEXT
FOR I=1 TO 1000 STEP 100
TEXT 200,10+((I-1)/10)*3,"3",0,1,1,STR$(I)
NEXT
FOR I=110 TO 10 STEP -10
TEXT 300,10+(ABS(I-110))*3,"3",0,1,1,STR$(I)
NEXT
FOR I=1 TO 5 STEP 0.5
IF I-INT(I)=0 THEN
Y=10+60*(I-1)
ELSE
Y=Y+30
ENDIF
TEXT 400,Y,"3",0,1,1,STR$(I)
NEXT
PRINT 1
EOP
TEST

```

54. IF...THEN...ELSE

Conditional type command

Syntax

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

Parameter	Description
condition statement	Available relational operator: <, >, =, Only one line is available in

Example

```
DOWNLOAD "TEST.BAS"
SIZE 3,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 1
REFERENCE 0,0
SET PEEL OFF
CLS
A=90
B=5
C$=""
D$=""
:L1
IF A>100 THEN
GOTO L2
ELSE
A=A+10
ENDIF
CLS
C$=STR$(A)+" IS SMALLER THAN 100"
TEXT 10,10,"2",0,1,1,C$
PRINT 1
GOTO L1
:L2
A=A+B
D$=STR$(A)+" IS LARGER THAN 100"
CLS
TEXT 10,100,"2",0,1,1,D$
PRINT 1
END
EOP
```

55. GOSUB...RETURN

This command is used to execute from main program to subroutine and then return to main program.

Syntax

```
GOSUB LABEL
statement
END
```

```
:LABEL
statement
RETURN
```

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

Example

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

56. GOTO

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

Syntax

```
GOTO LABEL
:LABEL
```

Example

```
DOWNLOAD "GOTO1.BAS"
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
A=0
```

```

TOTAL=0
:START
IF A<100 THEN
GOTO SUM
ELSE
GOTO PRTOUT
ENDIF
:SUM
A=A+1
TOTAL=TOTAL+A
GOTO START
:PRTOUT
B$="THE SUMMATION OF 1..100 IS "+STR$(TOTAL)
TEXT 10,100,"2",0,1,1,B$
PRINT 1
END
EOP

```

57. REM

Comment.

Syntax

REM

Example

```

REM *****
REM This is a demonstration program*
REM *****
DOWNLOAD "REMARK.BAS"
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT 50,50,"3",0,1,1,"REMARK DEMO PROGRAM"
REM TEXT 50,100,"3",0,1,1,"REMARK DEMO PROGRAM2"
PRINT 1,1
EOP
REMARK

```

58. INT()

This function will return the integer part of the floating point number.

Syntax

INT (n)

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

Example

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

59. LEFT\$()

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

Syntax

LEFT\$ (X\$, n)

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

Example

```
DOWNLOAD "TEST.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="BARCODE PRINTER DEMO PRINTING"
C$=LEFT$(A$,10)
CLS
TEXT 10,10,"2",0,1,1,A$
TEXT 10,100,"2",0,1,1,"10 LEFT 10 CHARS: "+C$
PRINT 1
EOP
TEST
```

60. LEN()

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 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
B=LEN(A$)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,50,"3",0,1,1,"STRING LENGTH="+STR$(B)
PRINT 1
EOP
DEMO
```

61. MID\$()

This function retrieves the specified number of characters down from the mth character of a string.

Syntax

MID\$(string, m, n)

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

Example

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
E$=MID$(A$,11,10)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,40,"3",0,1,1,"10 MIDDLE CHARS: "+E$
PRINT 1
EOP
DEMO
```

62. RIGHT\$()

This function returns a specified number of characters up from the rightmost 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 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHJKLMNOPQRSTUVWXYZ"
D$=RIGHT$(A$,10)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,80,"3",0,1,1,"10 RIGHT CHARS: "+D$
PRINT 1
EOP
DEMO
```

63. STR\$()

This function converts numeral into corresponding string of characters.

Syntax

STR\$(n)

Parameter	Description
n	The numeral to be processed

Example

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHJKLMNOPQRSTUVWXYZ"
F=100
G=500
H$=STR$(F+G)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,60,"3",0,1,1,"F="+STR$(F)
TEXT 10,110,"3",0,1,1,"G="+STR$(G)
TEXT 10,160,"3",0,1,1,"F+G="+H$
PRINT 1
EOP
DEMO
```

64. VAL()

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 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHJKLMNOPQRSTUVWXYZ"
F$="100"
G$="500"
CLS
H=VAL(F$)+VAL(G$)
I$=STR$(H)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,60,"3",0,1,1,"F="+F$
TEXT 10,110,"3",0,1,1,"G="+G$
TEXT 10,160,"3",0,1,1,"F+G="+I$
PRINT 1
EOP
DEMO
```

65. BEEP

This command is used to control the beeper to issue a beep sound.

Syntax

BEEP

Example

```
DOWNLOAD "DEMO.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
BEEP
INPUT "Text1 =",TEXT1$
CLS
TEXT 100,100,"3",0,1,1,TEXT1$
PRINT 1
EOP
```

66. INPUTFILE

This command is used to receive the data from serial interface, mainly for receiving electronic balance data.

Syntax

INPUTFILE File handle

Parameter	Description
file handle	0 or 1

Example

```
DOWNLOAD F,"INPUTFILE.BAS"
SIZE 60 mm,40 mm
GAP 2 mm,0 mm
:START
CLS
INPUTFILE 0
TEXT 10,20,"3",0,1,1,"20"+STR$(YEAR)+"-"+STR$(MONTH)+"
"+STR$(DATE)+"
"+STR$(HOUR)+":"+STR$(MINUTE)+":"+STR$(SECOND)
READ 0,T1$
A$=RIGHT$(T1$,7)
TEXT 20,60,"3",0,1,1,"T1= "+A$
READ 0,G$
B$=RIGHT$(G$,7)
TEXT 20,100,"3",0,1,1,"G= "+B$
READ 0,T$
C$=RIGHT$(T$,7)
TEXT 20,140,"3",0,1,1,"T= "+C$
READ 0,N$
D$=RIGHT$(N$,7)
TEXT 20,180,"3",0,1,1,"N= "+D$
PRINT 1,1
ADJUST
GOTO START
EOP
```

67. ADJUST

this command is used to align the label and tear-off paper position when execute LOOP command, for tearing off paper easily.

Syntax

ADJUST

Example

```
DOWNLOAD F,"ADJUST.BAS"
SIZE 60 mm,40 mm
GAP 2 mm,0 mm
:START
CLS
INPUTFILE 0
TEXT 20,20,"3",0,1,1,"Hello"
PRINT 1,1
ADJUST
GOTO START
EOP
```

Device Reconfiguration Commands

68. SET COUNTER

This setting sets the counter number in the program and its increments, mathematical expressions are not supported.

Syntax

SET COUNTER @n step

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

Example

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

69. SET KEY1, SET KEY2

This setting is used to enable/disable the KEY1, KEY2 presupposed function.

Syntax

SET KYE1 ON /OFF

SET KEY2 ON /OFF

Parameter	Description
ON	Enable KEY1 to be set "PAUSE" function Enable KEY2 to be set "FEED" function
OFF	Disable KEY1 to be set "PAUSE" function Disable KEY2 to be set "FEED" function

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

Example

```
DOWNLOAD "DEMO17.BAS"
```

```

SIZE 3,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET PEEL OFF
SET KEY1 OFF
CLS
:START
A=GETKEY()
IF A=0 THEN GOTO PAUSEB
IF A=1 THEN GOTO FEEDB
:PAUSEB
CLS
TEXT 50,10,"4",0,1,1,"PAUSE key is pressed!"
PRINT 1
GOTO START
:FEEDB
CLS
TEXT 50,10,"4",0,1,1,"FEED key is pressed!"
PRINT 1
GOTO START
EOP

```

70. SET PEEL

This setting is used to enable/disable the self-peeling function. The default setting for this function is off.

Syntax

SET PEEL ON/OFF

Parameter	Description
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 3
DIRECTION 0REFERENCE 0,0
SET PEEL ON
SET KEY1 OFF
CLS
TEXT 50,100,"3",0,1,1,"SELF-PEELING FUNCTION TEST"
PRINT

```

71. SET TEAR & SET STRIPPER

This command is used to enable/disable feeding of labels to gap position for tearing off. This setting will be saved in printer memory when turning off the power.

Syntax

```
SET TEAR ON/OFF
SET STRIPPER ON/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 ***TEAE 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
```

72. SET HEAD

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
```

Parameter	Description
ON	Turn on the "HEAD OPEN" sensor
OFF	Turn off the "HEAD OPEN" sensor

Example

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

73. SET COM1

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:

Parity	24: 2400 bps 48: 4800 bps 96: 9600 bps 19: 19200 bps Parity check N: None parity check E: Even parity check O: Odd parity check
data	Data bit 8: 8 bits data 7: 7 bits data
stop	Stop bit 1: 1 stop bit 2: 2 stop bits

Example

```

SIZE 3,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET PEEL ON
SET KEY1 OFF
SET DEBUG LABEL
SET RIBBON OFF
SET COM1 96,N,8,1
CLS
BARCODE 100,100, "39",48,1,0,2,5,"CODE 39"
PRINT 1

```

74. SET PRINTKEY

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 PRINTKEY OFF/ON/AUTO/<num>

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

Example

```

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
```

75. SET REPRINT

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

Parameter	Description
OFF	Disable this function
ON	Enable this function

Example

```
SET REPRINT ON
```

76. PEEL

This command is used to obtain status of the peel-off sensor. Its 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 "DEMO19.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL OFF
SET LED1 OFF
CLS
IF PEEL=1 THEN LED1=1
EOP
```

77. KEY1,KEY2

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

Syntax

KEYm = n

Parameter	Description
KEY1(PAUSE)	0:released

KEY2(FEED)	1:pressed 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

```

78. SET CUTTER

This setting activates or deactivates the cutter. This setting will be saved in printer memory after turning off the power.

Syntax

SET CUTTER OFF/BATCH/pieces

Parameter	Description
OFF	Disable cutter function.
BATCH	Set printer to cut label at the end of printing job.
pieces	Set number of printing labels per cut. 0<= pieces <=65535

Example

```

SIZE 3,3
GAP 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 50,50,"3",0,1,1,"SET CUTTER OFF"
PRINT 3

```

Result: *The cutter function is disabling.*

```

SET CUTTER BATCH
CLS
TEXT 50,50,"3",0,1,1,"SET CUTTER BATCH"
PRINT 3,2

```

Result: *The cutter cuts once after 6 labels are printed.*

```

SET CUTTER 1

```

```
CLS
TEXT 50,50,"3",0,1,1,"SET CUTTER 1"
PRINT 3,2
```

Result: *The cutter cuts every label.*

79. SET RESPONSE

This command can response issue automatically.

Syntax

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

Parameter	Description
["Job ID"]	Optional. Set job ID. Default is Null
ON	Enable this function, every response per every print
OFF	Disable this function.
BATCH	Enable this function, response one time when print job completed.

Returned value

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

Parameter	Returned value
Status	Refer to <ESC>! Command (HEX)
#####	00001-99999

Example

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

Result: {00,00001}{00,00002}{00,00003}

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

Result:

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

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

Result: {00,00006,CCCC}

Printer Global Variables

80. @LABEL

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

Syntax

Write attribute: @LABEL=n

Read attribute: A=@LABEL

Parameter	Returned value
n	Number of labels printed. 0<n<65535

Example

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

81. YEAR

This variable reads/writes the year data via the Real Time Clock (RTC). Four-digit year formats are supported by RTC.

Syntax

Write attribute: YEAR=02

Read attribute: A=YEAR

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

Example

```
DOWNLOAD "Set Year.BAS"
REM *****SetYear Parameter to RTC*****
YEAR=05
EOP
SetYear
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
```

```

CLS
REM *****Read YEAR parameter form RTC*****
YEAR$=STR$(YEAR)
Y=YEAR
REM *****Print*****
TEXT 10,10,"5",0,1,1,"YEAR1="+YEAR$
TEXT 10,110,"5",0,1,1,"YEAR2="+STR$(Y)
TEXT 10,210,"5",0,1,1,"YEAR3="+STR$(YEAR)
PRINT 1
EOP
DEMO

```

82. MONTH

This variable reads/writes the month data via the Real Time Clock (RTC). Two-digit (01~12) month formats are supported by RTC.

Syntax

Write attribute: MONTH=01
 Read attribute: A=MONTH
 Range: 01~12

Example

```

DOWNLOAD "SetMonth.BAS"
MONTH=05
EOP
SetMonth
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
MONTH$=STR$(MONTH)
M=MONTH
REM *****Print*****
TEXT 10,10,"5",0,1,1,"MONTH1="+MONTH$
TEXT 10,110,"5",0,1,1,"MONTH2="+STR$(M)
TEXT 10,210,"5",0,1,1,"MONTH3="+STR$(MONTH)
PRINT 1
EOP
DEMO

```

83. DATE

This variable reads/writes the date data via the Real Time Clock (RTC). Two-digit (01~31) date

formats are supported by RTC.

Syntax

Write attribute: DATE=12

Read attribute: A=DATE

Range: 01~31

Example

```
DOWNLOAD "SetDate.BAS"
REM *****Set Date Parameter to RTC*****
DATE=30
EOP
SetDate
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
REM *****Read Date parameter form RTC*****
DATE$=STR$(DATE)
D=DATE
REM *****Print*****
TEXT 10,10,"5",0,1,1,"DATE1="+DATE$
TEXT 10,110,"5",0,1,1,"DATE2="+STR$(D)
TEXT 10,210,"5",0,1,1,"DATE3="+STR$(DATE)
PRINT 1
EOP
DEMO
```

84. WEEK

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

Syntax

Write attribute: WEEK=3

Read attribute: A=WEEK

Range: 1(Sunday)~7(Saturday)

Example

```
DOWNLOAD "SetWeek.BAS"
REM *****Set Week Parameter to RTC*****
WEEK=6
EOP
SetWeek
DOWNLOAD "DEMO.BAS"
SIZE 3,3
```

```

GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
REM *****Read Week parameter form RTC*****
WEEK$=STR$(WEEK)
W=WEEK
REM *****Print*****
TEXT 10,10,"5",0,1,1,"WEEK1="+WEEK$
TEXT 10,110,"5",0,1,1,"WEEK2="+STR$(W)
TEXT 10,210,"5",0,1,1,"WEEK3="+STR$(WEEK)
PRINT 1
EOP
DEMO

```

85. HOUR

This variable reads/writes the hour data via the Real Time Clock (RTC). The 24-hour-day system (00~23) is supported by RTC.

Syntax

Write attribute: HOUR=12
 Read attribute: A=HOUR
 Range: 00~23

Example

```

DOWNLOAD "SetHour.BAS"
REM *****Set Hour Parameter to RTC*****
HOUR=11
EOP
SetHour
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
REM *****Read Hour parameter form RTC*****
HOUR$=STR$(HOUR)
H=HOUR
REM *****Print*****
TEXT 10,10,"5",0,1,1,"HOUR1="+HOUR$

```

```
TEXT 10,110,"5",0,1,1,"HOUR2="+STR$(H)
TEXT 10,210,"5",0,1,1,"HOUR3="+STR$(HOUR)
PRINT 1
EOP
DEMO
```

86. MINUTE

This variable reads/writes the minute data via the Real Time Clock (RTC). Two-digits (00~59) minute format is supported by RTC.

Syntax

Write attribute: MINUTE=12

Read attribute: A=MINUTE

Range: 00~59

Example

```
DOWNLOAD "SetMinute.BAS"
REM *****Set Minute Parameter to RTC*****
MINUTE=59
EOP
SetMinute
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
REM *****Read Minute parameter form RTC*****
MINUTE$=STR$(MINUTE)
MIN=MINUTE
REM *****Print*****
TEXT 10,10,"5",0,1,1,"MINUTE1="+MINUTE$
TEXT 10,110,"5",0,1,1,"MINUTE2="+STR$(MIN)
TEXT 10,210,"5",0,1,1,"MINUTE3="+STR$(MINUTE)
PRINT 1
EOP
DEMO
```

87. SECOND

This variable reads/writes the second data via the Real Time Clock (RTC). Two-digits (00~59) second format is supported by RTC.

Syntax

Write attribute: SECOND=12

Read attribute: A=SECOND

Range: 00~59

Example

```
DOWNLOAD "SetSecond.BAS"
REM *****Set Second Parameter to RTC*****
SECOND=59
EOP
SetSecond
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
REM *****Read Second parameter form RTC*****
SECOND$=STR$(SECOND)
SEC=SECOND
REM *****Print*****
TEXT 10,10,"5",0,1,1,"SECOND1="+SECOND$
TEXT 10,110,"5",0,1,1,"SECOND2="+STR$(SEC)
TEXT 10,210,"5",0,1,1,"SECOND3="+STR$(SECOND)
PRINT 1
EOP
DEMO
```

88. @YEAR

This variable reads/writes the year data via the Real Time Clock (RTC). Two-digit year formats are supported by RTC. @YEAR global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @YEAR="01"

Read attribute: @YEAR

Range: 00~99

Example:

```
REM *****Set @YEAR*****
@YEAR="05"
REM *****Print*****
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
```

```
TEXT 10,10, "5",0,1,1, "@YEAR"  
TEXT 310,10, "5",0,1,1,@YEAR  
PRINT 1
```

89. @MONTH

This variable reads/writes the month data via the Real Time Clock (RTC). Two-digits (01~12) month formats are supported by RTC. @MONTH global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @MONTH="01"

Read attribute: @MONTH

Range: 01~12

Example:

```
REM *****Set @MONTH*****  
@MONTH="12"  
REM *****Print*****  
SIZE 3,3  
GAP 0.08,0  
DENSITY 8  
SPEED 6  
DIRECTION 0  
REFERENCE 0,0  
SET CUTTER OFF  
SET PEEL OFF  
CLS  
TEXT 10,10,"5",0,1,1,"@MONTH"  
TEXT 310,10,"5",0,1,1,@MONTH  
PRINT 1
```

90. @DATE

This variable reads/writes the date data via the Real Time Clock (RTC). Two-digits (01~31) date formats are supported by RTC. @DATE global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @DATE="12"

Read attribute: @DATE

Range: 01~31

Example:

```
REM *****Set @DATE*****  
@DATE="31"  
REM *****Print*****  
SIZE 3,3  
GAP 0.08,0  
DENSITY 8  
SPEED 6  
DIRECTION 0  
REFERENCE 0,0
```

```
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"5",0,1,1,"@DATE"
TEXT 310,10,"5",0,1,1,@DATE
PRINT 1
```

91. @DAY

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

Syntax

Write attribute: @DAY="3"

Read attribute: @DAY

Range: 1(Sunday) ~7(Saturday)

Example:

```
REM *****Set @DAY*****
@DAY="5"
REM *****Print*****
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"5",0,1,1,"@DAY"
TEXT 310,10,"5",0,1,1,@DAY
PRINT 1
```

92. @HOUR

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

Syntax

Write attribute: @HOUR ="12"

Read attribute: @HOUR

Range: 00~23

Example:

```
REM *****Set @HOUR*****
@HOUR= "23"
REM *****Print*****
SIZE 3,3
GAP 0.08,0
DENSITY 8
```



```

SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10, "5",0,1,1,"@HOUR"
TEXT 310,10, "5",0,1,1,@HOUR
PRINT 1

```

93. @MINUTE

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

Syntax

```

Write attribute: @MINUTE ="12"
Read attribute: @MINUTE
Range: 00~59

```

Example:

```

REM *****Set @MINUTE*****
@MINUTE="59"
REM *****Print*****
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"5",0,1,1,"@MINUTE"
TEXT 310,10,"5",0,1,1,@MINUTE
PRINT 1

```

94. @SECOND

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

Syntax

```

Write attribute: @SECOND ="12"
Read attribute: @SECOND
Range: 00~59

```

Example:

```

REM *****Set @SECOND*****
@SECOND="59"

```

```
REM *****Print*****  
SIZE 3,3  
GAP 0,0  
DENSITY 8  
SPEED 6  
DIRECTION 0  
REFERENCE 0,0  
SET CUTTER OFF  
SET PEEL OFF  
CLS  
TEXT 10,10,"5",0,1,1,"@SECOND"  
TEXT 310,10,"5",0,1,1,@SECOND  
PRINT 1
```