WPL2

Wasp Technologies
BAR CODE PRINTER SERIES

PROGRAMMING
MANUAL

Revise History

Date	Description	Remark
2003/09/19	1. BASIC functions are added to WPL2 2. Supported code page update	
2004/5/21	QR code is supported	

TABLE OF CONTENTS

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 , Tab	(ASCII 32),(ASCII 9) characters will be ignored in the command line
“	(ASCII 34), beginning and ending of expression
CR,LF	(ASCII 13),(ASCII 10) or (ASCII 10) is placed at the end of command line.
NULL	(ASCII 0) is not supported in the expression, except the 2D bar code commands.
¥nnn	(ASCII 92), nnn is a number. Available range is from 000 to 255 to express the character in decimal.
FF , EOF	(ASCII 12),(ASCII 26) can be used at the end of command line only.

Note: ***200 DPI: 1 mm = 8 dots*** Arial font in bold and italic type is used for note.

Setup and System Commands

● SIZE

Description

This command defines the label width and length.

Syntax

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

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

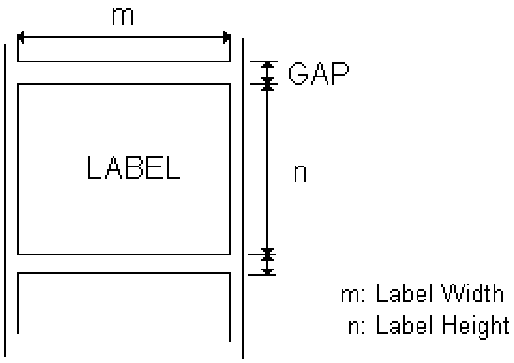
Note:

200 DPI: 1 mm = 8 dots
300 DPI: 1mm = 12 dots
For metric system, there must be a space between parameter and "mm".

Max. width	102mm	104mm	106mm	108mm
WPL606				**x**
WPL305				**x**

Example

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



See Also

GAP, BLINE

● GAP

Description

Define the gap distance between two labels

Syntax

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

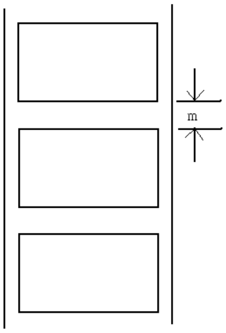
Parameter	Description
m	The gap distance between two labels 0 ≤ m ≤ 1 (inch), 0 ≤ m ≤ 25.4 (mm)
n	The offset distance of the gap n ≤ label length (inch or mm)

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

Example

Normal gap

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



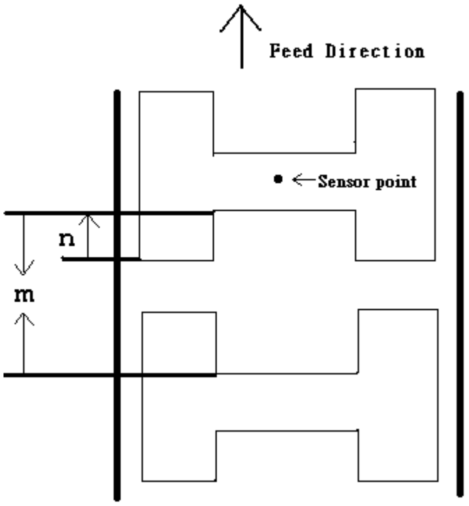
Special gap

(1) English system (inch)

GAP 0.30, 0.10

(2) Metric system (mm)

GAP 7.62 mm, 2.54 mm



See Also

SIZE, BLINE

● BLINE

Description

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

Syntax

(1) English system (inch)

    BLINE m, n

(2) Metric system (mm)

    BLINE m mm, n mm

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

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

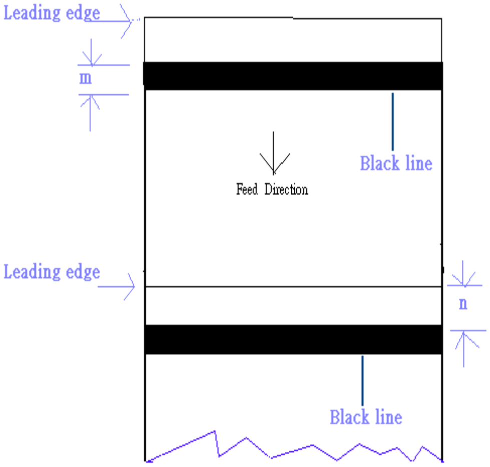
Example

(1) English system (inch)

    BLINE 0.20,0.50

(2) Metric system (mm)

    BLINE 5.08 mm,12.7 mm



See Also

SIZE, GAP

● OFFSET

Description

This command defines the selective, extra label feeding length each form feed takes, which, especially in peel-off mode and cutter mode, is used to adjust label stop position, so as for label to register at proper places for the intended purposes. The printer 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≤ m ≤1(inch)

CAUTION: Impropriety offset value may cause paper jam.

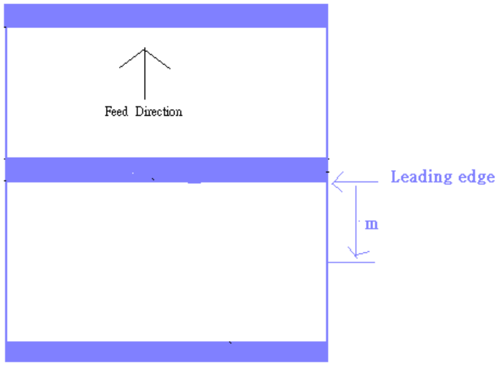
Example

(1) English system (inch)

```
OFFSET 0.5
```

(2) Metric system (mm)

```
OFFSET 12.7 mm
```



See Also

SIZE, GAP, SET PEEL, SET CUTTER

- ## SPEED

Description

This command defines the print speed.

Syntax

SPEED n

Parameter	**Description**
n	printing speed in inch per second

N	1	1.5	2	3	4	5	6	8	10	12
WPL606			**x**	**x**	**x**	**x**	**x**			
WPL305			**x**	**x**	**x**	**x**				

Example

SPEED 10

See Also

DENSITY

- ## DENSITY

Description

This command designates the level of darkness of printing.

Syntax

DENSITY n

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

Example

DENSITY 7

See Also

DENSITY

● DIRECTION

Description

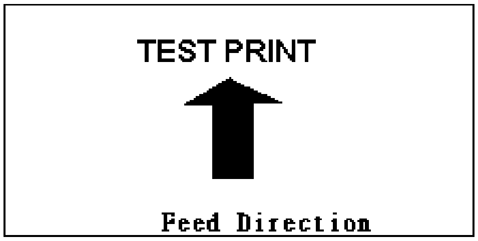
This command defines the printout direction.

Syntax

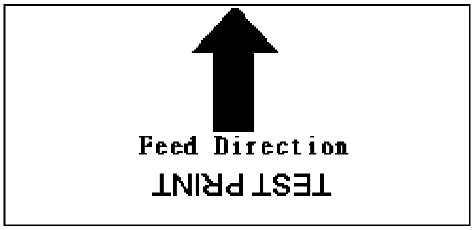
DIRECTION n

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

(DIRETCTION 0)



(DIRECTION 1)



Example

DIRECTION 0

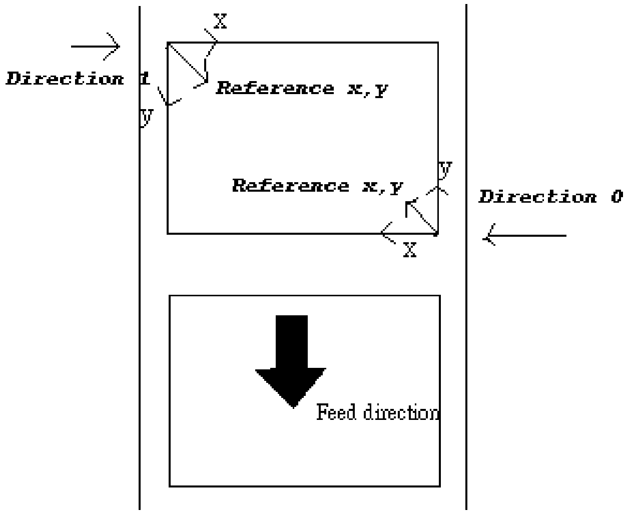
See Also

REFERENCE

● REFERENCE

Description

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



Syntax

REFERENCE x, y

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

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

Example

REFERENCE 10,10

See Also

DIRECTION

- # COUNTRY

Description

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

Syntax

COUNTRY n

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

Example

COUNTRY 001

See Also

CODEPAGE

● CODEPAGE

Description

This command defines the code page of international character set.

Syntax

CODEPAGE n

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

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

.

Example

CODEPAGE 437

See Also

COUNTRY, SET COM1

- # FEED

Description

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

Syntax

FEED n

Parameter	**Description**
n	unit: dot 1 ≤ n ≤ 9999

Example

FEED 40

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

See Also

BACKFEED, SIZE, GAP, BLINE

- ## BACKFEED

Description

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

Syntax

BACKFEED n

Parameter	**Description**
n	unit: dot 1 ≤ n ≤ 9999

Example

BACKFEED 40

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

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

See Also

FEED, SIZE, GAP, BLINE

● FORMFEED

Description

This command feeds label to the beginning of next label.

Syntax

FORMFEED

Parameter	Description
None	N/A

Example

FORMFEED

See Also

FEED, SIZE, GAP, BLINE

- # 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 of the label should be setup in advance.

Syntax

HOME

Parameter	Description
None	N/A

Example

HOME

● PRINT

Description

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

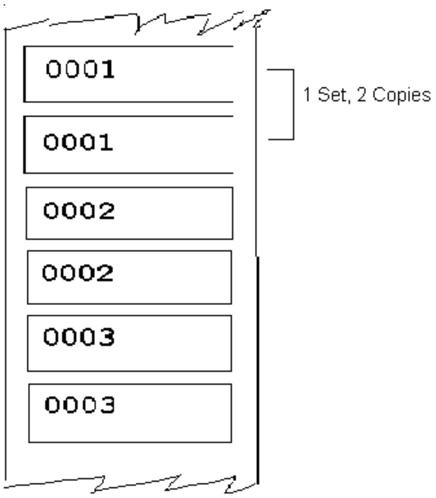
Syntax

PRINT m [,n]

Parameter	Description
m	Specifies how many sets of labels will be printed. 1 ≤ m ≤ 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 ≤ n ≤ 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
```



See Also

SET COUNTER, INPUT, DOWNLOAD

- # CUT

Description

At this command, the printer will activate the cutter to cut the labels immediately without back feeding the label.

Syntax

CUT

Parameter	Description
None	N/A

See Also

SET CUTTER

Label Formatting Commands

• CLS

Description

This command clears the image buffer.

Syntax

CLS

Parameter	**Description**
None	N/A

Note: This command must be placed after SIZE command.

Example

CLS

See Also

SIZE, GAP, BLINE

● BAR

Description

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

Syntax

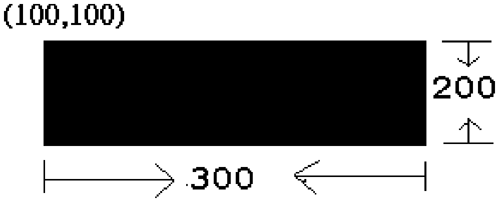
BAR x, y, width, height

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: ***200 DPI: 1 mm = 8 dots***
300 DPI: 1 mm = 12 dots
Recommended max. bar height is 12mm at 4" 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

BAR 100, 100, 300, 200



See Also

BOX

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

Example

ERASE 100,100,200,200

See Also

CLS

● REVERSE

Description

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 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***
300 DPI: 1 mm = 12 dots
Recommended max. height of reversed black area is 12mm at 4" width. Height of reversed area that is larger than 12 mm may damage the power supply and affect the print quality.
Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.

Example

REVERSE 100,100,200,200

- # BOX

Description

This command is used to draw rectangles on the label.

Syntax

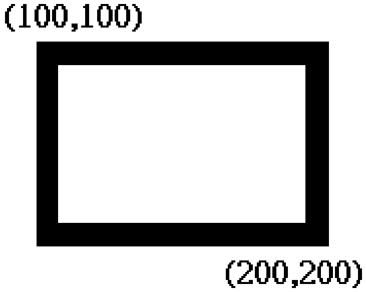
BOX X_start, Y_start, X_end, Y_end, line thickness

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

Note: ***200 DPI: 1 mm = 8 dots***
300 DPI: 1 mm = 12 dots
Recommended max. thickness of box is 12mm at 4" 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

BOX 100,100,200,200,5



See Also

BAR

- # BITMAP

Description

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

Syntax

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

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

Example

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

See Also

PUTBMP, PUTPCX

- ## PUTBMP

Description

This command is used to print BMP format image.

Syntax

PUTBMP X, Y, ”filename”

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

Note: ***Recommend two-color format (black and white). Maximum 256-color. This command is available for the following printer models.***
WPL606, WPL305

Example

PUTBMP 100,100,”LOGO.BMP”

- ## PUTPCX

Description

This command is used to print PCX format image.

Syntax

PUTPCX X, Y, ”filename”

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

Note: ***Recommend two-color format (black and white). Maximum 256-color.***
256-color supported models: WPL606, WPL305

Example

PUTPCX 100,100,”LOGO.PCX”

See Also

BITMAP, PUTPCX

● BARCODE

Description

This command is used to print 1D barcodes on label form.
The available bar codes are listed below:

- Code 128 (switching code subset automatically)
- Code 128M (switching code subset manually)
- EAN 128 (switching code subset automatically)
- Interleaved 2 of 5
- Interleaved 2 of 5 with check digit
- Code 39
- Code 39 with check digit
- Code 93
- EAN 13
- EAN 13 with 2 digits add-on
- EAN 13 with 5 digits add-on
- EAN 8
- EAN 8 with 2 digits add-on
- EAN 8 with 5 digits add-on
- Codabar
- Postnet
- UPC-A
- UPC-A with 2 digits add-on
- UPC-A with 5 digits add-on
- UPC-E
- UPC-E with 2 digits add-on
- UPC-E with 5 digits add-on
- MSI
- PLESSEY
- China POST
- ITF14
- EAN14

Syntax

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

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

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

Use “!” as a starting character for the control code followed by three control codes.
If the start subset is not set, the default starting subset is B.

EAN128	Code 128, switching code subset A, B, C automatically
25	Interleaved 2 of 5
25C	Interleaved 2 of 5 with check digits
39	Code 39
39C	Code 39 with check digits
93	Code 93
EAN13	EAN 13
EAN13+2	EAN 13 with 2 digits add-on
EAN13+5	EAN 13 with 5 digits add-on
EAN8	EAN 8
EAN8+2	EAN 8 with 2 digits add-on
EAN8+5	EAN 8 with 5 digits add-on
CODA	Codabar
POST	Postnet
UPCA	UPC-A
UPCA+2	UPC-A with 2 digits add-on

UPCA+5	UPC-A with 5 digits add-on
UPCE	UPC-E
UPCE+2	UPC-E with 2 digits add-on
UPCE+5	UPC-E with 5 digits add-on
CPOST	China post code
MSI	MSI code
PLESSEY	PLESSEY code
ITF14	ITF 14 code
EAN14	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 1:1	Narrow : wide 1:2	narrow : wide 1:3	narrow : wide 2:5
128	10x	-	-	-
EAN128	10x	-	-	-
25	-	10x	10x	5x
25C	-	10x	10x	5x
39	-	10x	10x	5x
39C	-	10x	10x	5x
93	-	-	10x	-
EAN13	8x	-	-	-
EAN13+2	8x	-	-	-
EAN13+5	8x	-	-	-
EAN 8	8x	-	-	-
EAN 8+2	8x	-	-	-
EAN 8+5	8x	-	-	-
CODA	-	10x	10x	5x
POST	1x	-	-	-
UPCA	8x	-	-	-
UPCA+2	8x	-	-	-
UPCA+5	8x	-	-	-
UPCE	8x	-	-	-
UPCE+2	8x	-	-	-
UPCE+5	8x	-	-	-
CPOST	-	10x	10x	5x
MSI	-	-	-	-

PLESSY	-	-	-	-
ITF14	-	10x	10x	5x
EAN14	-	-	-	-

code number the maximum number of digits of bar code content

Barcode type	Maximum bar code length
128	-
EAN128	-
25	-
25C	-
39	-
39C	-
93	-
EAN13	12
EAN13+2	14
EAN13+5	17
EAN 8	7
EAN 8+2	9
EAN 8+5	12
CODA	-
POST	-
UPCA	11
UPCA+2	13
UPCA+5	16
UPCE	6
UPCE+2	8
UPCE+5	11
CPOST	-
MSI	-
PLESSY	-
ITF14	13
EAN14	13

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.)

● DMATRIX

Description

This command is used to define the DataMatrix 2D bar code. Currently, it supports ECC200 error correction only.

Syntax

```
DMATRIX x, y, width, height, [Xm, Lm], expression
```

Parameter	Description
x	Horizontal start position in dot
y	Vertical start position in dot
width	The expected width of barcode area in dot
height	The expected height of barcode area in dot
Xm	Module size in dot
Lm	Expression length (without double quote), 1≤m≤2048

Example

```
DMATRIX 10,10,400,400,"DMATRIX"
DMATRIX 10,10,400,400,L7,DMATRIX
```

● MAXICODE

Description

This command is used to define a 2D Maxicode.

Syntax

```
MAXICODE x, y, mode, [class, country, post, Lm,] “message”
```

For mode 2 or 3:

```
    MAXICODE x, y, mode, class, country, postal code, ”low priority message"
```

if country is 840, the postal code is in 99999,9999 format.
For other countries, the code is up to 6 alphanumeric characters.

For mode 4,5,6

```
    MAXICODE x, y, mode, [Lm], "message"
```

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

Note: Mode 6 is not supported in WPL303, WPL305 Printers firmware.

Parameter	**Description**
x	X-coordinate of the staring point in dot
y	Y-coordinate of the starting point in dot
mode	2,3,4,5
class	Class of service, 3-digit number (for mode 2,3)
country	Country code, 3-digit number (for mode 2,3)
post	Post code (for mode 2,3) Mode 2: (USA) 5-digit+ 4-digit number Mode3: (Canada) 6 alphanumeric post code included by double quotes.
Lm	Expression length (double quote is ignored) , 1≤m≤138
message	Barcode content

Example

For USA:

```
MAXICODE 100,100,2,300,840,0681,07317,”DEMO FOR MAXICODE”
```

For Canada:

```
MAXICODE 100,100,3,300,840,”107317”,”DEMO FOR MAXICODE”
```

EXAMPLES:

```
REM MODE 4
SPEED 6
```

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

REM MODE 5
SPEED 6
CLS
SIZE 4.00,3.00
GAP 0.10,0
DENSITY 10
REM SET MAXIMODE 5
MAXICODE 24,24,5,"THIS IS A 93 CHARACTER CODE SET A MESSAGE
THAT FILLS A MODE 4, UNAPPENDED, MA"
BOX 424,16,700,60,2
DIRECTION 0
PRINT 1
```

● PDF417

Description

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

Syntax

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

Parameter	Description
x	X-coordinate of the starting point in dot
y	Y-coordinate of the starting point in dot
width	The expected width of barcode in dot
height	The expected height of barcode in dot
rotate	Rotation counterclockwise.
0:	0 degree
90:	90 degrees
180:	180 degrees
270:	270 degrees
expression	Barcode text or string expression to be printed on label.
[option]	
E	Error correction level Range: 0~8
W	Module width in dot Range: 2~9
H	Bar height in dot Range: 4~99
R	Maximum number of rows
C	Maximum number of columns
T	Truncation. 0: Not truncated 1: Truncated
Lm	Expression length (without double quote) , 1≤m≤2048

Example

```
PDF417 100,200,200,300,0,E1,"abcdef"
PDF417 100,200,200,300,0,E1,L6,abcdef
```

- **QRCODE**

Description

This command is used to print QR code

Syntax

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

Parameter	**Description**
X	The upper left corner x-coordinate of the QR code
Y	The upper left corner y-coordinate of the QR code
ECC level	Error correction recovery level
L	7%
M	15%
Q	25%
H	30%
cell width	1~10
mode	Auto / manual encode
A	Auto
M	Manual
rotation	
0	0 degree
90	90 degree
180	180 degree
270	270 degree
model	
1	(default), original version
2	enhanced version
mask	0~8, default is 7
Data string	The encodable character set is described as below

Encodable character set:

1). Numeric data: (digits 0~9)

2). Alphanumeric data (digits 0-9; upper case letters A-Z; nine other characters: space, $ % * + - . / :);

3). 8-bit byte data (JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201);

4). Kanji characters (Shift JIS values 8140_{HEX} –$9FFC_{HEX}$ and $E040_{HEX}$ –EAA4 $_{HEX}$. These are values shifted from those of JIS X 0208. Refer to JIS X 0208 Annex 1 Shift Coded Representation for detail.).

Data characters per symbol (for maximum symbol size):

	Model 1 (Version 14-L)	Model 2 (Version 40-L)
1). Numeric data:	1,167 characters	7,089 characters
2). Alphanumeric data:	707 characters	4,296 characters

3). 8-bit byte data:	486 characters	2,953 characters
4). Kanji data:	299 characters	1,817 characters

*If “A” is the first character in the data string, then the following data after “A” are Alphanumeric data.
*If “N” is the first character in the data string, then the following data after “N” are numeric data.
*If “B” is the first character in the data string, then the following 4 digits after “B” is used to specify numbers of data. After the 4 digits are the number of bytes of binary data to be encoded.
*If “K” is the first character in the data string , then the following data after “K” is Kanji data.
*If “!” is in the data string and follows by “N”, “A”, “B”, “K” then it will be switched to specified encodable character set.

Example

Manual mode example:

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

(Where A: Alphanumeric data)

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

(Where N: Numeric data)

```
QRCODE 100,10,Q,7,M,0,M1,S3,"N123456!ATHE FIRMWARE HAS BEEN UPDATED"
```

(Where N: Numeric data ; !:Transfer char ; A: Alphanumeric data)

```
QRCODE 100,10,H,7,M,0,M1,S3,"B0012Product name"
```

(where B: Binary data ; 0012: 12 bytes)

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

(Where K: Kanji data)

Auto mode example:

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

(1) Auto mode example

a. General data string

```
SIZE 4,2.5
GAP 0.12,0
CLS
QRCODE 10,10,H,4,A,0,"ABCabc123"
QRCODE 160,160,H,4,A,0,"123ABCabc"
QRCODE 310,310,H,4,A,0,"印表機 ABCabc123"
PRINT 1,1
```

b. Data string including <Enter> character (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,"印表機<Enter>
ABC<Enter>
abc<Enter>
123"
 PRINT 1,1
```

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

```
DOWNLOAD “DEMO.BAS”
SIZE 4,2.5
CAP 0.12,0
CLS
QRCODE 10,10,H,4,A,0,"ABCabc123"+STR$(1234)
QRCODE 160,160,H,4,A,0,"123ABCabc"+"1234"
QRCODE 310,310,H,4,A,0,"印表機 ABCabc123"+"1234"+"abcd"
PRINT 1,1
EOP
DEMO
```

d. Data string including double quote (“) character, please use ¥["] instead of

```
SIZE 4,2.5
CAP 0.12,0
CLS
QRCODE 10,10,H,4,A,0,"ABC¥["]abc¥["]123"
QRCODE 160,160,H,4,A,0,"123¥["]ABC¥["]abc"
QRCODE 310,310,H,4,A,0,"¥["]印表機¥["]ABCabc123"
PRINT 1,1
```

(3) Manual mode

a. General data string：

```
SIZE 4,2.5
CAP 0.12,0
CLS
QRCODE 10,10,H,4,M,0,"AABC!B0003abc!N123"
QRCODE 160,160,H,4,M,0,"N123!AABC!B0003abc"
QRCODE 310,310,H,4,M,0,"K 印表機!AABC!B0006abc123"
 PRINT 1,1
```

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

```
SIZE 4,2.5
CAP 0.12,0
CLS
```

```
QRCODE 10,10,H,4,M,0,"AABC!B0007<Enter>
abc<Enter>
!N123"
QRCODE 160,160,H,4,M,0,"N123!B0002<Enter>
!AABC!B0005<Enter>
abc"
QRCODE 310,310,H,4,M,0,"K 印表機!B0002<Enter>
!AABC!B0010<Enter>
abc<Enter>
123"
  PRINT 1,1
```

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

```
SIZE 4,2.5
CAP 0.12,0
CLS
QRCODE 10,10,H,4,M,0,"AABC!B0006abc123!N"+STR$(1234)
QRCODE 160,160,H,4,M,0,"N123!AABC!B0007abc"+"1234"
QRCODE 310,310,H,4,M,0,"K 印表機!AABC!B0014abc123"+"1234"+"abcd"
 PRINT 1,1
```

e. Data string including double quote (“) character, please use ¥["] instead of

```
SIZE 4,2.5
CAP 0.12,0
CLS
QRCODE 10,10,H,4,M,0,"AABC!B0005¥["]abc¥["]!N123"
QRCODE 160,160,H,4,M,0,"N123!B0001¥["]!AABC!B0004¥["]abc"
QRCODE 310,310,H,4,M,0,"B0001¥["]!K 印表機!B0010¥["]ABCabc123"
PRINT 1,1
```

● TEXT

Description

This command is used to print text on label

Syntax

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

Parameter	**Description**
X	The x-coordinate of the text
Y	The y-coordinate of the text
font	Font name
1	8 x 12 fixed pitch dot font
2	12 x 20 fixed pitch dot font
3	16 x 24 fixed pitch dot font
4	24 x 32 fixed pitch dot font
5	32 x 48 dot fixed pitch font
6	14 x 19 dot fixed pitch font OCR-B
7	21 x 27 dot fixed pitch font OCR-B
8	14 x25 dot fixed pitch font OCR-A
ROMAN.TTF	ROMAN True type font
Rotation	The rotation angle of text
0	0 degree
90	90 degrees, in clockwise direction
180	180 degrees, in clockwise direction
270	270 degrees, in clockwise direction
X-multiplication:	Horizontal multiplication, up to 10x. Available factors: 1~10 For true type font, this parameter is ignored.
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.

Example

```
TEXT 100,100,”3”,0,1,1,”DEMO FOR TEXT”
TEXT 100,400,”ROMAN.TTF”,0,1,20,”True Type Font Test Print”
```

Status Polling Commands (RS-232)

● <ESC>!?

Description

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

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

Syntax

<ESC>!?

Parameter	Description
None	N/A

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	**Description**
N/A	N/A

See Also

<ESC>!?

- **~!I**

Description

The command is used to inquire the code page and country setting of the printer.
The returned information is given in the following format
code page, country code
ex: 8 bit: 437, 001
    7 bit: USA, 001
Regarding the code pages and country codes supported by the printer, please refer to the **CODEPAGE** and **COUNTRY** command respectively.

Syntax

~!I

Parameter	Description
None	N/A

See Also

COUNTRY, CODEPAGE, SET COM1

- **~!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 0x0A.

The ending character is 0x1A.

Syntax

~!F

Parameter	Description
None	N/A

See Also

FILES, KILL

Message Translation Protocols

- **~#**

Description

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

@0 following the ending identifier ~& is used to instruct keyboard to display the prompt in the first line of LCD display.

@1 following the ending identifier ~& is used to instruct keyboard to display the prompt in the first line of LCD display.

If @0 or @1 is ignored, prompt string will be displayed in first line of LCD and input data will be displayed in second line of LCD.

Syntax

~#Prompt~&[@0]
~#Prompt~&[@1]

Parameter	Description
None	N/A

Example

```
OUT "~#KP-200 V1.0~&@0"
OUT "~#Initialization~&@1"
```

See Also

INPUT, OUT

● BEEP

Description

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

Syntax

BEEP

Parameter	**Description**
None	N/A

Example

```
DOWNLOAD “DEMO.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
BEEP
INPUT “Text1 ?”,TEXT1$
TEXT 100,100,”3”,0,1,1,TEXT1$
PRINT 1
EOP
```

Commands for Windows Driver

- ## !B

Description

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

Syntax

!Bnnn

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

Example

!B100

See Also

BITMAP

- **!J**

Description

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

Syntax

!Jnnnn

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

Example

!J0100

See Also

FEED

File Management Commands

● DOWNLOAD

Description

"DOWNLOAD" is a header of the file that is to be saved in the printer's memory. The downloaded files can be 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: 50
Maximum numbers of file saved in Flash Files: 100

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. F: Download files to main board flash memory. E: Download files to expansion memory module.
FILENAME.BAS	The filename resident in printer memory.

Note:

(1). The filename is case sensitive.
(2). The extension of of the program file must be ".BAS"
(3). Filename format must be in 8.3 format.
(4). 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.

2. Download a data file

DOWNLOAD [F,]"FILENAME", DATA SIZE, DATA CONTENT…where

<u>Parameter</u>	<u>Description</u>
n	Specify the memory which is used to save the download files. F: Download files to main board flash memory. E: Download files to expansion memory module.

FILENAME: The name of data file that will remain resident in the printer memory. It is case sensitive.

DATA SIZE: The actual size (numbers of byte) of the data file without header.

Note:

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

(2). If memory is not specified, all files will be downloaded to DRAM. No Battery is used to 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 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 100,100,"3",0,1,1,"EXAMPLE PROGRAM"
PRINT 1
EOP
```

Note: When writing a download program, "DOWNLOAD" header must be placed in the beginning of file, and "EOP" must be placed at the end of program. To run the program, you can call the main filename without BAS extension or use RUN command to start the download program.

Example:

1. ***Call the main filename***

```
C:¥>COPY CON LPT1<ENTER>
EXAMPLE<ENTER>
<CTRL><Z>
C:¥>
```

2. ***Use Run command to start the program***

```
C:¥>COPY CON LPT1<ENTER>
RUN "EXAMPLE.BAS"<ENTER>
<CTRL><Z>
C:¥>
```

Below is an example of downloading data file.

```
DOWNLOAD “DATA”,20,COMPUTER<ENTER>
2001<ENTER>
21<ENTER>
```

Note: <ENTER> stands for 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, INPUT

● 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 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 100,100,”3”,0,1,1,”DEMO PROGRAM”
PRINT 1
EOP
```

See Also

DOWNLOAD, EOP, INPUT

FILES

Description

This command lists the files that remained in the printer memory (both FLASH memory and DRAM). The total memory size and available memory size are listed as well.

Syntax

FILES

Example

Follow the steps below to list the files that are saved in printer memory in DOS environment through serial port connection

.

```
C:¥>MODE COM2 96,N,8,1<ENTER>
C:¥>COPY CON COM2<ENTER>
      FILES<ENTER>
      <CTRL><Z><ENTER>
C:¥>
```

Note: <ENTER> stands for keyboard "ENTER" key. In the above example, please press "ENTER" key instead of typing <ENTER> in the above example.
<CTRL> stands for keyboard "Ctrl" key.

See Also

~!F, KILL

● KILL

Description

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

Syntax

KILL [n],"FILENAME"

Parameter	Description
n	Specify the memory location that files will be deleted. F: Download files to main board flash memory. E: Download files to 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 SELF TEST 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 serial port connection

.

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

Note: <ENTER> stands for keyboard "ENTER" key. In the above example, please press "ENTER" key instead of typing <ENTER> in the above example.
<CTRL> stands for keyboard "Ctrl" key.

See Also

~!F, FILES

● 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 keyboard “ENTER” key. In the above example, please press “ENTER” key instead of typing <ENTER> in the above example.
<CTRL> stands for keyboard “Ctrl” key.

See Also

DOWNLOAD, EOP

BASIC Commands and Functions

● ABS( )

Description

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

Syntax

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

Example

```
DOWNLOAD "TEST.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=ABS(-100)
B=ABS(-50.98)
C=-99.99
TEXT 100,100,"3",0,1,1,STR$(A)
TEXT 100,150,"3",0,1,1,STR$(B)
TEXT 100,200,"3",0,1,1,STR$(ABS(C))
PRINT 1
EOP
```

See Also

DOWNLOAD, EOP

- # ASC( )

Description

This function returns the ASCII code of the character.

Syntax

```
ASC (“A”)
```

Example

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

See Also

DOWNLOAD, EOP, STR$()

● CHR$( )

Description

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

Syntax

CHR$(n)

Parameter	Description
n	The ASCII code

Example

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

See Also

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

- # END

Description

This command states the ending of program.

Syntax

```
END
```

Example

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

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

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

EOP
```

See Also

DOWNLOAD, EOP

- # EOF( )

Description

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

Syntax

EOF (File Handle)

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

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

Example

```
DOWNLOAD “DEMO.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA",0
SEEK 0,4
:A
ITEM$=""
READ 0,ITEM$,PRICE,QTY
TEXT 10,10,"3",0,1,1,"SEEK COMMAND TEST (SHIFT 4 CHARACTERS)"
BOX 98,48,502,514,2
A$="ITEMS:"+ITEM$
B$="PRICE:"+STR$(PRICE)
C$="QTY:"+STR$(QTY)
TEXT 128,114,"2",0,1,1,A$
TEXT 130,198,"2",0,1,1,B$
TEXT 132,268,"2",0,1,1,C$
BARCODE 132,365,"39",96,1,0,2,4,"PRICE-2000"
PRINT 1
I=EOF(0)
IF I=0 THEN GOTO A
PRINT 1
EOP
```

See Also

DOWNLOAD, EOP, OPEN, READ, SEEK

● OPEN

Description

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

Syntax

OPEN “Filename”, File handle

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

Example

If a file by the name of “DATA is to be downloaded,
The file format contains:

```
DOWNLOAD DATA”,20,Computer<CR>
20000<CR>
15<CR>
```

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

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

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

```
DOWNLOAD “DEMO.BAS”
SIZE 3.00,3.00
CLS
SPEED 2
DENSITY 8
SET CUTTER OFF
SET PEEL OFF
DIRECTIO 0
REFERENCE 0,0
OPEN “DATA”,1
SEEK 1,0
READ 1,ITEM$,PRICE,QTY
I=EOF(1)
```

```
IF I>0 THEN END
BOX 98,48,502,514,2
A$=”ITEMS:”+ITEM$
B$=”PRICE:”+STR$(PRICE)
C$=”QTY:”+STR$(QTY)
TEXT 128,114,”2”,0,1,1,A$
TEXT 130,198,”2”,0,1,1,B$
TEXT 132,268,”2”,0,1,1,C$
BARCODE 132,365,”39”,96,1,0,2,4,”PRICE-2000”
PRINT 1
CLOSE 1
EOP
```

Saving the above contents of data under the file name of **“**DEMO**”.**
Follow the steps below to download data to the printer
<under MS-DOS mode>:

```
C:¥>MODE COM1:96,N,8,1 ↵
C:¥>COPY DEMO COM1 ↵
```

The above example sets the following: baud rate at 9600 bps, no parity, 8 bits data, 1 stop bit. Saving the following command of program under the file name of “EXECUTE”: DEMO

```
C:¥>COPY EXECUTE COM1↵
```

The above example instructs the printer to open the file “DATA” with a file handle of 1 and read items from the file.

See Also

DOWNLOAD, EOP, READ, EOF, LOF, SEEK, FREAD$()

● READ

Description

This command is used to read data from downloaded data file

Syntax

READ file handle, variables

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

Example

```
DOWNLOAD “OPEN1.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA",0
OPEN "DATA1",1
SEEK 0,4
:A
ITEM$=""
READ 0,ITEM$,PRICE,QTY
TEXT 10,10,"3",0,1,1,"SEEK COMMAND TEST (SHIFT 4 CHARACTERS)"
BOX 98,48,502,514,2
A$="ITEMS:"+ITEM$
B$="PRICE:"+STR$(PRICE)
C$="QTY:"+STR$(QTY)
TEXT 128,114,"2",0,1,1,A$
TEXT 130,198,"2",0,1,1,B$
TEXT 132,268,"2",0,1,1,C$
BARCODE 132,365,"39",96,1,0,2,4,"PRICE-2000"
PRINT 1
I=EOF(0)
IF I=0 THEN GOTO A
SEEK 1,0
:B
READ 1,ITEM$,PRICE,QTY
TEXT 10,10,"4",0,1,1,"OPEN, READ, EOF() COMMAND TEST"
BOX 98,48,502,514,2
```

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

See Also

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

SEEK

Description

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

Syntax

SEEK file handle, offset

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

Example

```
DOWNLOAD “TEST.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA",0
SEEK 0,4
ITEM$=""
READ 0,ITEM$,PRICE,QTY
TEXT 10,10,"3",0,1,1,"SEEK COMMAND TEST (SHIFT 4 CHARACTERS)"
BOX 98,48,502,514,2
A$="ITEMS:"+ITEM$
B$="PRICE:"+STR$(PRICE)
C$="QTY:"+STR$(QTY)
TEXT 128,114,"2",0,1,1,A$
TEXT 130,198,"2",0,1,1,B$
TEXT 132,268,"2",0,1,1,C$
BARCODE 132,365,"39",96,1,0,2,4,"PRICE-2000"
PRINT 1
EOP
```

See Also

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

- # LOF( )

Description

This function returns the size of the specified file.

Syntax

LOF (“FILENAME”)

Parameter	**Description**
FILENAME	The file downloaded in the printer memory.

Example

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

See Also

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

- ## FREAD$( )

Description

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

Syntax

FREAD$ (file handle, byte)

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

Example

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

See Also

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

- ## FOR…NEXT LOOP

Description

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

Syntax

```
For variable = start TO end statement; start < end

NEXT
```

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

Example

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

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

FOR I=50 TO 10 STEP -10
C$=C$+STR$(I)+" "
```

```
TEXT 10,100,"3",0,1,1,C$
NEXT

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

PRINT 1
EOP
```

See Also

DOWNLOAD, EOP

● IF…THEN…ELSE

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 condition1 THEN
        Statement block 1
```

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

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

Example

```
DOWNLOAD “IFTHEN.BAS”
SIZE 4,4
GAP 0,0
```

```
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=50
B=5
C$=""
D$=""

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

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

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 IF …THEN …ELSE statement in a program can not exceed than 40.

The total numbers of IF …THEN …ELSE, FOR…NEXT, WHILE…WEND, and DO …LOOP in a program can not exceed than 40 loops.

See Also

DOWNLOAD, EOP

- ## GOSUB…RETURN

Description

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

Syntax

```
GOSUB LABEL
      statement

END

:LABEL
      statement

RETURN
```

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

Example

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

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

:DASH
```

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

EOP
```

See Also

DOWNLOAD, EOP, END, GOTO

- ## GOTO

Description

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

Syntax

```
GOTO LABEL

:LABEL
```

Parameter	**Description**
None	N/A

Example

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

:START
IF A<100 THEN GOTO SUM ELSE GOTO PRTOUT
:SUM
A=A+1
TOTAL=TOTAL+A
GOTO START
:PRTOUT
B$="THE SUMMATION OF 1..100 IS "+STR$(TOTAL)
TEXT 10,100,"3",0,1,1,B$
PRINT 1
END
EOP
```

See Also

DOWNLOAD, EOP, END, GOSUB…RETURN

- ## INP$( )

Description

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

Syntax

INP$(n)

Parameter	**Description**
n	1: com1 port in printer

Example

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

See Also

DOWNLOAD, EOP, END, INPUT, GOSUB…RETURN, GOTO

- ## INPUT

Description

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

Syntax

INPUT [“Prompt string”], variables

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

Example

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

See Also

DOWNLOAD, EOP, END, GOTO

- ## 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 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”REMARK DEMO PROGRAM”
EOP
```

See Also

DOWNLOAD, EOP, END

- ## OUT

Description

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

Syntax

OUT “prompt”, variable

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

Example

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

See Also

DOWNLOAD, EOP, END, ~#…~&

- ## GETKEY( )

Description

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

Syntax

```
GETKEY()
```

Example

```
DOWNLOAD “DEMO4.BAS”
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
:START
A=GETKEY()
IF A=0 THEN GOTO PAUSEB
IF A=1 THEN GOTO FEEDB
:PAUSEB
CLS
TEXT 50,10,"4",0,1,1,"PAUSE key is pressed !"
PRINT 1
GOTO START
:FEEDB
CLS
TEXT 50,10,"4",0,1,1,"FEED key is pressed !"
PRINT 1
EOP
```

See Also

DOWNLOAD, EOP, END, GOTO

- ## INT( )

Description

This function is used to truncate a floating point number.

Syntax

```
INT (n)
```

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

Example

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

See Also

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

- ## LEFT$( )

Description

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

Syntax

```
LEFT$ (X$, n)
```

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

Example

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

See Also

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

● LEN( )

Description

This function returns the length of a string.

Syntax

LEN (string)

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

Example

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

See Also

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

- ## MID$( )

Description

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

Syntax

MID$(string, m, n)

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

Example

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

See Also

DOWNLOAD, EOP, END, LEFT$(), LEN(), RIGHT$(), STR$(), VAL()

- ## RIGHT$( )

Description

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

Syntax

RIGHT$ (X$, n)

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

Example

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

See Also

DOWNLOAD, EOP, END, LEFT$(), LEN(), MID$(), STR$(), VAL()

- ## STR$( )

Description

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

Syntax

STR$ (n)

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

Example

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

See Also

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

- ## VAL( )

Description

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

Syntax

VAL (“numeric character”)

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

Example

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

See Also

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

Device Reconfiguration Commands

● 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	**Description**
@n	n: counter number. There are 50 counters available (0~49) in the printer.
step	The increment of the counter, can be positive or negative. -999999999<= step <=999999999 If the counter is used as a fixed variable, please set the increment to 0.
Expression	Initial string. String length is 2K bytes

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
COUNTER @1 1
@1=”00001”
COUNTER @2 5
@2=”AB000001”
CLS
TEXT 50,50,”3”,0,1,1,@1
BARCODE 50,500,”39”,48,1,0,2,4,@2
PRINT 100
```

See Also

PRINT, TEXT, BARCODE

- # SET CUTTER

Description

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

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

Syntax

SET CUTTER OFF/BATCH/pieces

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

Example

```
REM SET CUTTER FUNCTION OFF EXAMPLE PROGRAM
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”SET CUTTER 0”
BARCODE 50,500,”39”,48,1,0,2,4,”DEMO14”
PRINT 1

REM SET CUTTER BATCH EXAMPLE PROGRAM
REM This program cuts 3 times (3 sets)
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER BATCH
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”SET CUTTER BATCH”
BARCODE 50,500,”39”,48,1,0,2,4,”DEMO14”
PRINT 3,2
```

```
REM SET CUTTER PIECE EXAMPLE PROGRAM
REM This program cuts each printed label
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER 1
SET PEEL OFF
CLS
TEXT 50,50,”3”,0,1,1,”SET CUTTER PIECE”
BARCODE 50,500,”39”,48,1,0,2,4,”DEMO16”
PRINT 3,2
```

See Also

OFFSET, PRINT

- ## SET PEEL

Description

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

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

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

Syntax

```
SET PEEL ON/OFF
```

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

Example

```
REM SELF-PEELING FUNCTION ON
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL ON
CLS
TEXT 50,100,”3”,0,1,1,”SELF-PEELING FUNCTION TEST”
PRINT 5
```

See Also

OFFEST, PRINT

● SET TEAR

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
```

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 SELF-PEELING FUNCTION ON
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET TEAR ON
CLS
TEXT 50,100,”3”,0,1,1,”SELF-PEELING FUNCTION TEST”
PRINT 5
```

See Also

SET PEEL, SET CUTTER

- ## SET GAP

Description

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

Syntax

SET GAP n

Parameter	**Description**
n	The gap sensor light emission strength. Available range is listed as below. 0 is the lowest sensitivity

Printer model	Range
WPL606	0~31
WPL305	0~31

Example

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

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

Note: <ENTER> stands for keyboard "ENTER" key. In the above example, please press "ENTER" key instead of typing <ENTER> in the above example.
<CTRL> stands for keyboard "Ctrl" key.

Press the FEED key to test. Does printer stop at the same position on each label without the error light blinking? If not, please adjust the setting to a larger number again.
When trying with this setting, please begin from 0 and then on to higher values gradually.

See Also

SIZE, GAP, BLINE

● SET RIBBON

Description

This setting is used to enable/disable ribbon sensor detection. (Thermal Transfer Printing/Thermal Direct Printing)
Printer will detect if there is ribbon or without ribbon installed in ribbon mechanism at the instance of turning on printer power to determine direct thermal or thermal transfer printing.
This setting will not be saved in printer memory.

Syntax

```
SET RIBBON ON /OFF
```

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

Example

```
REM Thermal direct printing
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET RIBBON OFF
CLS
BARCODE 100,100,"39",48,1,0,2,5,"CODE 39”
PRINT 1
```

● SET COM1

Description

This setting defines communication parameters for printer serial port.

Syntax

SET COM1 baud, parity, data, stop

Parameter	**Description**
baud	Baud rate, available baud rates are as listed : 24: 2400 bps 48: 4800 bps 96: 9600 bps 19: 19200 bps 38: 38400 bps 56: 56000 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 keyboard "ENTER" key. In the above example, please press "ENTER" key instead of typing <ENTER> in the above example.
<CTRL> stands for keyboard "Ctrl" key.

- ## SET PRINTKEY

Description

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

Syntax

SET PRINTEKY n

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

Example

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

Printer Global Variables

- ## @LABEL

Description

This variable (read only) 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

```
@LABEL                    0<=@LABEL<=999999999
```

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET RIBBON OFF
CLS
TEXT 100,100 , ”3”,0,1,1,@LABEL
PRINT 1
```

● YEAR

Description

This variable is used to read from/write to RTC the year data. Four-digit year format is supported by RTC.

Syntax

Write attribute: YEAR=02
Read attribute: A=YEAR
Range: 00~99

Example

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

See Also

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

● MONTH

Description

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

Syntax

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

Example

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

See Also

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

● DATE

Description

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

Syntax

Write attribute: DATE=12
Read attribute: A=DATE
Range: 01~31

Example

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

See Also

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

- **WEEK**

Description

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

Syntax

Write attribute: WEEK=3
Read attribute: A=WEEK
Range:
1(Sunday)~7(Saturday)

Example

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

See Also

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

- ## HOUR

Description

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

Syntax

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

Example

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

See Also

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

MINUTE

Description

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

Syntax

Write attribute: MINUTE=12
Read attribute: A=MINUTE
Range: 00~59

Example

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

See Also

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

● SECOND

Description

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

Syntax

Write attribute: SECOND=12
Read attribute: A=SECOND
Range: 00~59

Example

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

See Also

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

- ## @YEAR

Description

This variable is used to read from/write to RTC the year data. Four-digit year format is supported by RTC.

@YEAR global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: YEAR=”2001”
Read attribute: YEAR
Range: 2000~9999

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"3",0,1,1,@YEAR
PRINT 1
```

See Also

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

- ## @MONTH

Description

This variable is used to read from/write to RTC the month data. Two-digits (01~12) month format is supported by RTC.
@MONTH global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @MONTH=”01”
Read attribute: @MONTH
Range: 01~12

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"3",0,1,1,@MONTH
PRINT 1
```

See Also

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

- ## @DATE

Description

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

@DATE global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @DATE=”12”
Read attribute: @DATE
Range: 01~31

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"3",0,1,1,@DATE
PRINT 1
```

See Also

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

- ## @DAY

Description

This variable is used to read from/write to RTC the week data, 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:
  0(Sunday)~6(Saturday)

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"3",0,1,1,@DAY
PRINT 1
```

See Also

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

● @HOUR

Description

This variable is used to read from/write to RTC the hour data. The 24-hour-day system (00~23) is supported by RTC.
@HOUR global variable can be accessed directly without using BASIC language functions.

Syntax

Write attribute: @HOUR =”12”
Read attribute: @HOUR
Range: 00~23

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"3",0,1,1,@HOUR
PRINT 1
```

See Also

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

- ## @MINUTE

Description

This variable is used to read from/write to RTC the minute data. Two-digits (00~59) minute format is supported by RTC.
@MINUTE global variable can be accessed directly without using BASIC language functions.

Syntax

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

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"3",0,1,1,@MINUTE
PRINT 1
```

See Also

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

- ## @SECOND

Description

This variable is used to read from/write to RTC the second data. Two-digits (00~59) second format is supported by RTC.
@SECOND global variable can be accessed directly without using BASIC language functions.

Syntax

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

Example

```
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"3",0,1,1,@SECOND
PRINT 1
```

See Also

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

Last update: May. 21, 2004