



Thermal printer command manual book

Copy right: Beijing Rongda science & technology Co., Ltd

DATA	VERSION	NOTE
2015-08-19	V1.0	
2016-06-20	V1.12	Increase Unicode Command Set
2017-02-08		Amend 1B 76 Description
2018-04-12	V1.13	Increase 1B 3D Description

Note: This command book applied to DP/DG/DE/EM/EHV2/EF/ET/V2/FH8C7/FH5C7



Table of content

1.1 Command set	4
1.2 Command details.....	6
1.2.1 Control command.....	错误！未定义书签。
ESC @	错误！未定义书签。
FF	6
LF	7
CR	7
ESC J	7
ESC d	错误！未定义书签。
ESC c	8
HT	错误！未定义书签。
ESCD n1 n2 ... nk NULL	错误！未定义书签。
ESC - n	错误！未定义书签。
ESC . n	错误！未定义书签。
GS B n	10
FS 2 n	11
ESC \$ nL nh	11
ESC l n	12
ESC Q n	12
ESC 1 n	错误！未定义书签。
ESC SP n	13
ESC a n	错误！未定义书签。
FS r n	14
1.2.2 Zooming command.....	错误！未定义书签。
ESC U n	错误！未定义书签。
ESC V n	14
ESCX	15
1.2.3 Graphic command.....	16
ESCKnL nH d1 d2dk	错误！未定义书签。
ESC* m nL nH d1...dk	16
GS v 0 m xL xH yL yH d1...dk	23
1.2.4 Barcode command.....	24
EGS h n	24
GS w n	24
GS H n	25
GS Q n	26
GS k	26



GS W n	27
GS k m v r d1.....dk	30
1.2.5 Curve command.....	30
ESC ‘	31
1.2.6 Printing status command.....	33
ESC v	33
1.2.7 Kanji Character set up command.....	33
FS &	34
FS.	34
ESC 6	34
ESC 7	34
ESC ! n	35
FS Y	35
1.2.8 Vertical tabulation and printing command.....	36
FS V	36
1.2.9 Bit image download and printing command.....	36
GS * x y d1...dk	错误! 未定义书签。
GS / n	39
FS q n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n	39
FS p nm	40
1.2.10 Printing depth set up command.....	错误! 未定义书签。
ESC s	40
1.2.11 Paper Cutting Command (For printer with cutter only) 错误!未定义书签。	
ESC I	40
ESC m	41
1.2.12 International character set.....	42
ESC R n	42
ESC t n	43
1.2.13 Consecutive curve printing command.....	44
ESC) n	44
ESC # n	45
ESC (n	46
1.2.14 Mode Selection Command.....	错误! 未定义书签。
ESC= n	46



RD Series thermal printers use the ESC / POS compatible command, and increase some functions such as the Chinese characters printing, Character and Chinese characters rotation, word spacing adjustment, printing barcodes and other functions.

1.1 Command List

Command	Function
ESC @	To initialize the printer
FF	Print and feed paper to the top of the next page (only the models with the black mark detection function)
LF	To print and line feed
CR	To print and carriage return
ESC J	To print and feed paper
ESC d	To print and feed paper n lines
ESC c	To allow/ban reverse printing
HT	To execute horizontal tab
ESC D	To set the position of horizontal tab
ESC -	To allow/ban the underline printing (to set/clear the underline)
ESC .	To set and cancel underline
ESC !	Select printing modes
GS B	To allow/ban white reverse printing mode
FS 2	To set character rotation Printing
ESC \$	To set printing absolute position
ESC I	To set the printing position
ESC Q	To set the right margin width
ESC 1	To set the line spacing

ESC SP	To set the character spacing
ESC a	Select alignment methods
FS r	To select the superscript and subscript
ESC U	Horizontally magnify character
ESC V	Vertically magnify character
ESC X	Magnify characters
ESC K	Printing graphics command ①
ESC *	Printing graphics command ②
GS v	Print raster bit image
GS h	Select bar code's height
GS w	Select bar code's width
GS H	Select the printing position for bar code character
GS Q	Set horizontal printing position for the bar code
GS k	To print the bar code
GS W	To magnify bar code
GS k	To Print QRCODE (Only to support QRCODE Printer)
ESC ‘	To print Curve
ESC v	To Send the printer's status to the Mainframe
FS &	To Select the Kanji mode
FS .	To cancel Kanji mode
ESC 6	To select Character Set 1 (6X8)
ESC 7	To select Character Set 2 (6X8)
ESC !	To select printing modes for characters
FS V	To do Vertical tabulation and printing
GS *	To Define and download bitmap
GS /	To Print downloaded bitmap
FS q	To Define NV bit image
FS p	To Print the NV bit image
ESC s	To adjust print depth
ESC i	Full Paper Cut(For printer with cutter only)

ESC m	Semi paper cutting(For printer with cutter only)
ESC R	To select international character set
ESC t	To select character code table
ESC)	To print Horizontal n line segment
ESC #	To set printing curve printing mode
ESC (To print N lines Curves command
ESC =	Mode selection command

This chapter describes the commands of controlling the printer to print. Format specification is as follows:

【COMMAND】 + **【parameter】**

【COMMAND】 is the command, and consists of the escape character and command characters. But a small number of single-byte commands don't have the escape character.

【parameter】 is the parameter, which is in italics. And the parameters are not numeric characters, but the value of the character.。

All the examples in this chapter are compiled in C language. The 'Send Data To Printer' function is a virtual functions. And require developers to write according to the actual situation of the mainframe.:

```
Send Data To Printer(unsigned char *buffer, unsigned int len)
```

This function is defined as follows :

Unsigned char *buf: Print data;

Unsigned int len: Data length, Unit: Byte.

1.2 Command

1.2.1 control command

ESC @

[Name] To initialize the printer

[Type] ASCII:	ESC	@
Decimal:	27	64
Hex:	1B	40

[Explanation] Clear the data in the print buffer, and reset the printing parameters to the current printer default parameters.

[Note]

- The data in the receive buffer is not cleared.

[Example] unsigned char str [2];

```
Str [0] = 0x1B;
```

```
Str [1] = 0x40;
```

```
Send Data To Printer(str,2);
```

FF

[Name] Print and feed paper to the top of the next page

[Type] ASCII: FF

Decimal: 12

Hex: 0C

[Explanation] Print all data in the printing buffer and return to the standard mode

[Note] If the paper has pre-printed black mark, take the paper to the black mark after printing the data in the data buffer. if the paper does not have black mark, feed paper 30cm. Pre-printed black mark specifications, see Appendix C. Pre-black Label printing instructions.

[Example] unsigned char str[2];

```
str[0] = 0x0C;
```

```
Send Data To Printer(str,1);
```

LF

[Name] print and feed line

[Type] ASCII: LF

Decimal: 10

Hex: 0A

[Explanation] Print the data in the print buffer and feed one line

[Note] The command sets the print position to the beginning of the line

[Example] unsigned char str[2];

```
str[0] = 0x0A;//或str [0] = '\n'
```

```
Send Data To Printer(str,1);
```

CR

[Name] print and carriage return

[Type] ASCII: CR

Decimal: 13

Hex: 0D

[Explanation] Print the data in the print buffer and carriage return

[Reference] LF

[Example] unsigned char str[2];

```
str[0] = 0x0D;//或str[0] = '\r'
```

```
SendDataToPrinter(str,1);
```

ESC J

[Name] print and feed paper

[Type] ASCII:	ESC	J	n
Decimal:	27	74	n
Hex:	1B	4A	n

[Explanation] Print the data in the print buffer and feeds forward paper [$n \times 0.125\text{mm}(0.0049\text{'})$].

[Comment]

- After printing is finished, the command sets the print starting position to the beginning of the line.

[Scope] $0 \leq n \leq 255$

[Example] unsigned char str[3];

str[0] = 0x1B;

str[1] = 0x4A;

str[2] = 0x4;

Send Data To Printer(str,3);// feeds forward paper 0.5mm

ESC d

[Name] print and feed paper n lines

[Type] ASCII:	ESC d	n
Decimal:	27 100	n
Hex:	1B 64	n

[Scope] $0 \leq n \leq 255$

[Explanation] Print the data in the print buffer and feed paper n lines

[Comment]

- After finishing the printing, this command sets the print starting position to the beginning of the line.
- One line distance is 24 vertical pitch (0.125mm)

[Example] unsigned char str[3];

str[0] = 0x1B;

str[1] = 0x64;

str[2] = 0x4;

Send Data To Printer(str,3);// feed forward paper 4 lines

ESC c

[Name] To allow/ban reverse printing

[Type] ASCII:	ESC	C	n
Decimal:	27	99	n
Hex:	1B	63	n

[Scope] $0 \leq n \leq 1$

[Explanation]

When $n=1$, allow the reverse printing and the printing direction is from left to right.

When $n=0$, ban the reverse printing and the printing direction is from right to left.

[Comment]

When the printer is vertically installed, the printer uses the reverse printing way.

Reversely printing not only supports character mode, and also supports graphical mode. When reversely printing graphics, we should note the printing order of graphics unit. (See the ESC K command)

```
[Example] unsigned char str[3];
           str[0] = 0x1B;
           str[1] = 0x63;
           str[2] = 0x1
           Send DataTo Printer(str,3);// reverse printing
```

HT

[Name] horizontal tab

[Type] ASCII: HT
Decimal: 9
Hex: 09

[Explanation] Move the print position to the next horizontal tab position

[Note] • The command is ignored unless the next horizontal tab position has been set.
• Horizontal tab positions are set with the 'ESC D'.

[Reference] ESC D

ESC D n1 n2 ... nk NULL

[Name] To set the position of horizontal tab

[Type] ASCII: ESC D n1...nk NULL
Decimal: 27 68 n1...nk 0
Hex: 1B 44 n1...nk 00

[Scope] $1 \leq n \leq 255$ $0 \leq k \leq 20$

[Explanation] Set the position of horizontal tab

n specifies the column number for setting a horizontal tab position from the beginning of a line.

k indicates the total number of horizontal tab positions to be set.

[Note]

- The horizontal position is stored as a value of [character width × n]measured from the beginning of the line. The character width includes the default width of the characters' spacing.
- This command deletes the previously set level positioning location.
- When n = 8, the printing position is moved to the 9th column by sending HT.
- The command is not affected by the ESC X command.
- This command cancels the previous tabulator position settings.
- The character printing position ,which exceeds the positioning location, will be processed as normal data.
- Transmit [n] k in ascending order and place a NULL code 0 at the end.
- When nk is less than or equal to the preceding value n (k-1),tab setting is finished and the following data is processed as normal data.
- ESC D NULL cancels all horizontal tab position.

- Even if the character width changes, previously specified horizontal tab positions don't also change.

[Default] The default tab positions are Font A (12*24).

[Example] unsigned str[8];

```
unsigned char Order = 9;
str[0] = 0x1B;
str[1] = 0x44;
str[2] = 2; // one character spacing from the first column
str[3] = 9; // eight character spacing from the first column
str[4] = 14; // thirteen character spacing from the first column
str[5] = 0; // end
Send Data To Printer (str,6)
Send Data To Printer (&Order,1);
Send Data To Printer ("HT1",3);
Send Data To Printer (&Order,1);
Send Data To Printer ("HT2",3);
Send Data To Printer (&Order,1);
Send Data To Printer ("HT3",3);
Order = 0x0D;
Send Data To Printer (&Order,1);
Send Data To Printer ("1234567890123456\r",17)
```

```
HT1   HT2   HT3
1234567890123456
```

ESC – n

[Name] To select/cancel the underline mode

[Type] ASCII: ESC – *n*
Decimal: 27 45 *n*
Hex: 1B 2D *n*

[Explanation] *n* = 1, select the underline mode
n = 0, cancel the underline mode

[Note]

- Underline can't act in the rotation and reverse characters.
- This command only affects the English and Kanji characters.

[Default] *n* = 0.

[Example] unsigned char str[3];

```
str[0] = 0x1B;
str[1] = 0x2D;
str[2] = 0x1;
Send Data To Printer (str,3); // set the underline
```

ESC . n

[Name] Cancel/Set up underline

[Type] ASCII ESC . *n*
Decimal 27 46 *n*
Hex 1B 2E *n*

[Explanation] *n* =1, Allow underline printing; *n*=0, Cancel underline printing.

[Note] • The command is invalid to reverse and rotation character.
 • The command is only valid to English and Kanji character.

[Default] *n* = 0

[E. g] unsigned char str[3];

 str[0] = 0x1B;

 str[1] = 0x2E;

 str[2] = 0x1;

 Send Data To Printer (str,3); // Set up overline

GS B n

[Name] select/cancel white reverse printing mode

[Type] ASCII: GS B n
Decimal: 29 66 n
Hex: 1D 42 n

[Scope] $0 \leq n \leq 255$

[Explanation] select/cancel white reverse printing mode

- When the LSB of *n* is 0, cancel white/black reverse printing mode.
- When the LSB of *n* is 1, select white/black reverse printing mode.

[Comment]

- Only the lowest bit of *n* is valid.
- The command is valid for the built-in and user-defined characters.
- This command only affects the English and Kanji characters.

[Default] *n*=0

[Example] unsigned char str[3];

 str[0] = 0x1D;

 str[1] = 0x42;

 str[2] = 1; // set the white reverse printing mode

 Send Data To Printer(str, 3);

FS 2 n

[Name] set character rotation Printing

[Type] ASCII: FS 2 *n*
Decimal: 28 73 *n*
Hex: 1C 49 *n*

[Scope] $0 \leq n \leq 3$

[Explanation] The command can rotate the character. The value of *n* is as follows:

n (Decimal)	Counterclockwise rotation
0	Does not rotate
1	90 degrees (Counterclockwise rotation)
2	180 degrees (Counterclockwise rotation)
3	270 degrees (Counterclockwise rotation)

[Note] Under the 90 degrees or 270 degrees rotation mode, the character width and height magnification direction is opposite to the magnification direction of the general mode.

[Default] n=0

[Example] unsigned char str[3];

```
str[0] = 0x1C;
```

```
str[1] = 0x49;
```

```
str[2] = 1; // set 90 degrees rotation
```

```
SendDataToPrinter(str, 3);
```

ESC \$ nL nH

[Name] Set absolute print position

[Type] ASCII: ESC \$ nL nH

Decimal: 27 36 nL nH

Hex: 1B 24 nL nH

[Scope] $0 \leq nL + (nH \times 256) < 384$

[Explanation] Set the distance from the beginning of the line to the position at which subsequent characters are to be printed.

The distance from the beginning of the line to the printing position is N horizontal dot pitch

The nL and nH are the low and high bit of double-byte unsigned integer N. $N = nL + nH \times 256$

[Comment]

- Settings outside the specified printable area are ignored.
- In mode 1, $n \leq 372$; In mode 2, $n \leq 420$

[Example] unsigned char str[4];

```
str[0] = 0x1B;
```

```
str[1] = 0x24;
```

```
str[2] = 32; //
```

```
SendDataToPrinter (str, 3); // Set the absolute position to 32 horizontal dot pitch from  
the left margin
```

ESC I n

[Name] set the left margin

[Type] ASCII: ESC 1 n

Decimal: 27 108 n

Hex: 1B 6C n

[Scope] $0 \leq n \leq 32$

[Explanation]

The left margin is the number of characters, which isn't printed on the left side of the printing paper.

The distance from the beginning of the line to the printing position is the width of n English characters.

[Comment]

- If the printing position is outside the printable area, the command is ignored.
- The width of the character includes the default character width of the character spacing

[Example] unsigned char str[4];

```
str[0] = 0x1B;
```

```
str[1] = 0x6C;
```

```
str[2] = 3;//
```

Send Data To Printer (str, 3); // the left position is set to the width of 3 English characters from the left margin

ESC Q n

[Name] set the right margin

[Type] ASCII:	ESC	Q	n
Decimal:	27	81	n
Hex:	1B	51	n

[Scope] $0 \leq n \leq 32$

[Explanation] The right margin is the number of characters, which isn't printed on the right side of the printing paper.

[Comment]

- If the printing position is outside the printable area, the command is ignored.
- The width of the character includes the default character width of the character spacing

[Example] unsigned char str[4];

```
str[0] = 0x1B;
```

```
str[1] = 0x51;
```

```
str[2] = 3;//
```

Send Data To Printer (str, 3); // set the area of three characters' width to the unprintable area on the right side

ESC 1 n

[Name] set the line spacing

[Type] ASCII:	ESC	1	n
Decimal:	27	49	n
Hex:	1B	31	n

[Scope] $0 \leq n \leq 255$ (The default value of 'n' is 3)

[Default] n=3

[Explanation] Set the line spacing to n vertical dot pitch

[Example] unsigned char str[4];

```
str[0] = 0x1B;
```

```
str[1] = 0x31;  
str[2] = 8;  
Send DataTo Printer(str,3);// Set the line spacing to 8 vertical dot pitch
```

ESC SP n

[Name] set the character spacing

[Type] ASCII: ESC SP n
Decimal: 27 32 n
Hex: 1B 20 n

[Scope] $0 \leq n \leq 255$ (The default value of 'n' is 0)

[Explanation] Set the character spacing to n horizontal dot pitch

[Example] unsigned char str[4];

```
str[0] = 0x1B;  
str[1] = 0x20;  
str[2] = 8;  
Send Data To Printer(str,3);// Set the character spacing to 8 horizontal dot pitch
```

ESC a n

[Name] Select justification methods

[Type] ASCII: ESC a n
Decimal: 27 97 n
Hex: 1B 61 n

[Scope] $0 \leq n \leq 2$

[Explanation] Aligns all the data in one line to the specified position.

n selects the justification as follows:

n	justification methods
0	Left justification
1	Centering
2	Right justification

[Comment]

- This command is only valid at the beginning of the line.

[Default] n=0

[Example]

```
unsigned char str[4];  
str[0] = 0x1B;  
str[1] = 0x61;  
str[2] = 1;  
SendDataToPrinter(str,3);// select the centering to print
```

FS r n

[Name] select the superscript and subscript

[Type] ASCII: FS r n
Decimal: 28 114 n
Hex: 1C 72 n

[Scope] $0 \leq n \leq 1$

[Explanation]

The value of n	Result
n=0	superscript
n=1	subscript

[Comment]

The command is effective for all characters (including English characters and Kanji)

The command is ignored if n is outside the defined scope

[Example] unsigned char str[3];

```
str[0] = 0x1C;
```

```
str[1] = 0x72;
```

```
str[2] = 0;
```

```
Send Data To Printer(str,3);//
```

1. 2. 2 Zoom command

ESC U n

[Name] Horizontally magnify characters

[Type] ASCII: ESC U n

Decimal: 27 85 n

Hex: 1B 55 n

[Scope] $0 \leq n \leq 8$

[Comment]

The command is effective for all characters (including English characters and Kanji)

The command is ignored if n is outside the defined scope

[Reference] ESC X

[Example] unsigned char str[4];

```
str[0] = 0x1B;
```

```
str[1] = 0x55;
```

```
str[2] = 2;
```

```
SendDataToPrinter(str,3);// Horizontally magnify 2 times
```

ESC V n

[Name] Vertically magnify characters

[Type] ASCII: ESC V n

Decimal: 27 86 n

Hex: 1B 56 n

[Scope] $0 \leq n \leq 8$

[Comment]

The command is effective for all characters (including English characters and Kanji)

The command is ignored if n is outside the defined scope

[Reference] ESC X

[Example] unsigned char str[4];

```
str[0] = 0x1B;  
str[1] = 0x56;  
str[2] = 2;  
Send Data To Printer(str,3);// Vertically magnify 2 times
```

ESCX

[Name] Magnify characters

[Type] ASCII:	ESC	X	n1	n2
Decimal:	27	88	n1	n2
Hex:	1B	58	n1	n2

[Scope] $0 \leq n \leq 8$ ($1 \leq n1$ horizontal times ≤ 8 , $1 \leq n2$ vertical times ≤ 8)

[Comment]

The command is effective for all characters (including English characters and Kanji), except barcode reading characters.

The command is ignored if n is outside the defined scope.

Vertical direction is the paper feeding direction, and horizontal direction is vertical with the paper feeding direction. When character clockwise rotate 90 °, the relationship between the vertical direction and horizontal direction is reversed, that is to say, this command's priority is lower than the FS 2. And when two commands is effective at the same time, the characters firstly rotate, then enlarge.

[Example] unsigned char str[4];

```
str[0] = 0x1B;  
str[1] = 0x58;  
str[2] = 2;  
str[3] = 2;  
SendDataToPrinter(str,4);// Vertically and horizontally magnify 2 times
```

1.2.3 Graphics command

ESCKnL nH d1 d2dk

[Name] Printing graphics command ①

[Type] ASCII	ESC	K	nL	nH	d1...dk
Decimal:	27	75	nL	nH	d1...dk
Hex:	1B	4B	nL	nH	d1...dk

[Scope] $0 \leq nL \leq 255$

$0 \leq nH \leq 1$

$0 \leq d \leq 255$

[Explanation]

This command can only print the black/white bit-image whose height is 8 dots and width does not exceed the printable area.

The nL and nH are the low and high bit of double-byte unsigned integer N. They express the number of the dots of the bit-image on the horizontal direction.

[Reference] ESC *

[Comment]

- The graphics command is influenced by the character enlargement command.

- When using reverse printing mode, successively print each graphics unit according to the order of the graphics from bottom to up.

[Example] unsigned char str[30];
unsigned char i=0;
str[i++] = 0x1B; str[i++] = 0x4B;
str[i++] = 15; //print the graphics whose width is 15 dots
str[i++] = 0x7C; str[i++] = 0x44; str[i++] = 0x44; str[i++] = 0xFF;
str[i++] = 0x44; str[i++] = 0x44; str[i++] = 0x7C; str[i++] = 0x00;
str[i++] = 0x41; str[i++] = 0x62; str[i++] = 0x54; str[i++] = 0xC8;
str[i++] = 0x54; str[i++] = 0x62; str[i++] = 0x41; str[i++] = 0x0D;
Send Data To Printer(str,i);//send the printing graphics command.

ESC* m nL nH d1...dk

[Name] Printing graphics command ②

[Type] ASCII ESC * m nL nH d1...dk
Decimal: 27 42 m nL nH d1...dk
Hex: 1B 2A m nL nH d1...dk

[Scope] m = 0, 1, 32, 33

0 ≤ nL ≤ 255

0 ≤ nH ≤ 1

0 ≤ d ≤ 255

[Explanation]

This command can only print the black/white bit-image whose height is 8 dots or 24 dots and width does not exceed the printable area.

The parameter meaning is as follows:

Using the m to select the bit image modes, and the dots of the bit image in the horizontal direction are specified by the nL and nH.:

m	The number of vertical dots (height)	Double-width mode
0	8	Twice as width
1	8	single-width
32	24	Twice as width
33	24	single-width

The nL and nH are the low and high bit of double-byte unsigned integer N. They express the number of the dots of the bit-image on the horizontal direction.

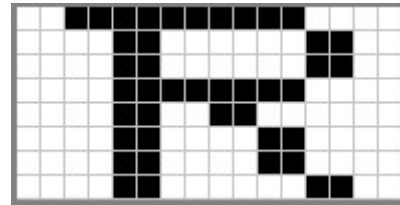
Mode 1: When the double-width mode is single-width, its maximum is 384. When the double-width mode is twice as width, its maximum is 192.

Mode 2: When the double-width mode is single-width, its maximum is 432. When the double-width mode is twice as width, its maximum is 216.

d1.....dk express the bit-image data. And the specific format is as follows:

[Example 1] m =0 (8 dots, twice as width), d1 represents the data to be printed in the first and second column. And dk represents the data to be printed in the $2k^{\text{th}}$ and $(2k-1)^{\text{th}}$ column. The bn represents, the n^{th} bit of the byte.

d1	d2	d3	d4	d5	d6	d7	d8	
0	1	1	1	1	1	0	0	b7
0	0	1	0	0	0	1	0	b6
0	0	1	0	0	0	1	0	b5
0	0	1	1	1	1	0	0	b4
0	0	1	0	1	0	0	0	b3
0	0	1	0	0	1	0	0	b2
0	0	1	0	0	1	0	0	b1
0	0	1	0	0	0	1	0	b0



Program code is as follows:

```

unsigned char str[100];

j=0;

str [j++] = 0x1B;

str r[j++] = 0x2A;

str [j++] = 0; //m=0(height is 8 dots, twice as width)

str [j++] = 8; // the width of the graphic is 8dots

str [j++] = 0;

// the bit image data

str [j++] = 0x00;str [j++] = 0x80;str [j++] = 0xFF;str [j++] = 0x90;str [j++] = 0x98;

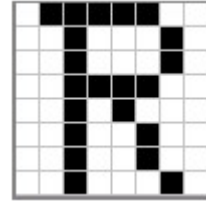
str [j++] = 0x96;str [j++] = 0x61;str [j++] = 0x00;str [j++] = 0x0D;//打印出图形

SendDataToPrinter(str,j);
    
```

[Example 2]: (8 dots, single-width), d1 represents the data to be printed in the first column.And dk represents the data to be printed in the k^{th} column. The bn represents the n^{th} bit of the byte.

d1	d2	d3	d4	d5	d6	d7	d8	

0	1	1	1	1	1	0	0	b7
0	0	1	0	0	0	1	0	b6
0	0	1	0	0	0	1	0	b5
0	0	1	1	1	1	0	0	b4
0	0	1	0	1	0	0	0	b3
0	0	1	0	0	1	0	0	b2
0	0	1	0	0	1	0	0	b1
0	0	1	0	0	0	1	0	b0



Program code is as follows:

```
unsigned char str[100];
```

```
j=0;
```

```
str[j++] = 0x1B;
```

```
str[j++] = 0x2A;
```

```
str[j++] = 1; //m=1(Height is 8 dots 、 No magnify)
```

```
str[j++] = 8; // the graphic width is 8dots str[j++] = 0;
```

```
// Bitmap data
```

```
str[j++] = 0x00;str[j++] = 0x80;str[j++] = 0xFF;str[j++] = 0x90;str[j++] = 0x98;
```

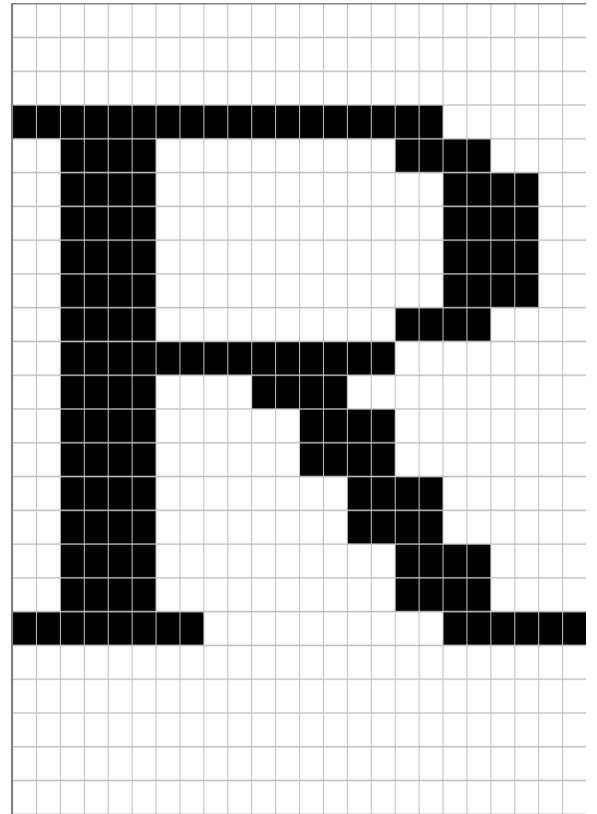
```
str[j++] = 0x96;str[j++] = 0x61;str[j++] = 0x00;str[j++] = 0x0D; ;//打印出图形
```

```
SendDataToPrinter(str,j);
```

E.g:

m =32 (24 dots, twice as width), d1,d2 and d3 represent the data to be printed in the first, second and third column. And dk represents the data to be printed in the kth column. The bn represents the nth bit of the byte.

	d4	d7									d34		
d1	0	0	0	0	0	0	0	0	0	0	0	b7	
	0	0	0	0	0	0	0	0	0	0	0	b6	
	0	0	0	0	0	0	0	0	0	0	0	b5	
	1	1	1	1	1	1	1	1	1	0	0	b4	
	0	1	1	0	0	0	0	0	0	1	1	0	b3
	0	1	1	0	0	0	0	0	0	1	1	0	b2
	0	1	1	0	0	0	0	0	0	1	1	0	b1
d2	0	1	1	0	0	0	0	0	0	1	1	0	b7
	0	1	1	0	0	0	0	0	1	1	0	0	b6
	0	1	1	1	1	1	1	1	0	0	0	0	b5
	0	1	1	0	0	1	1	0	0	0	0	0	b4
	0	1	1	0	0	0	1	1	0	0	0	0	b3
	0	1	1	0	0	0	1	1	0	0	0	0	b2
	0	1	1	0	0	0	0	1	1	0	0	0	b1
d3	0	1	1	0	0	0	0	0	1	1	0	0	b7
	0	1	1	0	0	0	0	0	1	1	0	0	b6
	1	1	1	1	0	0	0	0	0	1	1	1	b5
	0	0	0	0	0	0	0	0	0	0	0	0	b4
	0	0	0	0	0	0	0	0	0	0	0	0	b3
	0	0	0	0	0	0	0	0	0	0	0	0	b2
	0	0	0	0	0	0	0	0	0	0	0	0	b1
	0	0	0	0	0	0	0	0	0	0	0	b0	



Program code is as follows:

```
unsigned char str[200];
```

```
j=0;
```

```
str[j++] = 0x1B;
```

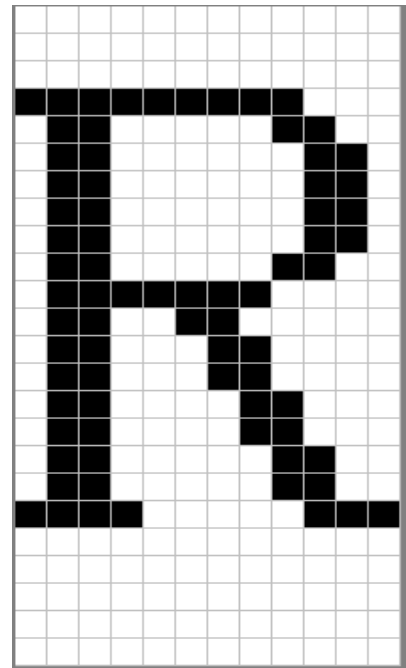
```
WWW.RD-CN.COM
```



```
str[j++] = 0x2A;
str[j++] = 32; //m=32(width 24 dots, double time width)
str[j++] = 12; // Graphic width is 12dots
str[j++] = 0;
// bitmap data str[j++] = 0x10;str[j++] = 0x00;str[j++] = 0x20;str[j++] = 0x1F;str[j++] =
0xFF;str[j++] = 0xE0;
str[j++] = 0x1F;str[j++] = 0xFF;str[j++] = 0xE0;str[j++] = 0x10;str[j++] = 0x20;str[j++] =
0x20;
str[j++] = 0x10;str[j++] = 0x20;str[j++] = 0x00;str[j++] = 0x10;str[j++] = 0x30;str[j++] =
0x00;
str[j++] = 0x10;str[j++] = 0x3C;str[j++] = 0x00;str[j++] = 0x10;str[j++] = 0x2f;str[j++] =
0x00;
str[j++] = 0x18;str[j++] = 0x43;str[j++] = 0xC0;str[j++] = 0x0F;str[j++] = 0xC0;str[j++] =
0xE0;
str[j++] = 0x07;str[j++] = 0x80;str[j++] = 0x20;str[j++] = 0x00;str[j++] = 0x00;str[j++] =
0x20;
str[j++] = 0x0D;// Print out present bitmap
SendDataToPrinter(str,j);
```

例 4: m=33(Height is 24dot,No magnify) d1,d2 and d3 represent the data to be printed in the first, second and third column, The bn represents the nth bit of the byte

	d4	d7									D	d49
d1	0	0	0	0	0	0	0	0	0	0	0	b7
	0	0	0	0	0	0	0	0	0	0	0	b6
	0	0	0	0	0	0	0	0	0	0	0	b5
	1	1	1	1	1	1	1	1	1	0	0	b4
	0	1	1	0	0	0	0	0	1	1	0	b3
	0	1	1	0	0	0	0	0	0	1	1	b2
	0	1	1	0	0	0	0	0	0	1	1	b1
	0	1	1	0	0	0	0	0	0	1	1	b0
d2	0	1	1	0	0	0	0	0	1	1	0	b7
	0	1	1	0	0	0	0	0	1	1	0	b6
	0	1	1	1	1	1	1	1	0	0	0	b5
	0	1	1	0	0	1	1	0	0	0	0	b4
	0	1	1	0	0	0	1	1	0	0	0	b3
	0	1	1	0	0	0	1	1	0	0	0	b2
	0	1	1	0	0	0	0	1	1	0	0	b1
	0	1	1	0	0	0	0	1	1	0	0	b0
d3	0	1	1	0	0	0	0	0	1	1	0	b7
	0	1	1	0	0	0	0	0	1	1	0	b6
	1	1	1	1	0	0	0	0	0	1	1	b5
	0	0	0	0	0	0	0	0	0	0	0	b4
	0	0	0	0	0	0	0	0	0	0	0	b3
	0	0	0	0	0	0	0	0	0	0	0	b2
	0	0	0	0	0	0	0	0	0	0	0	b1
	0	0	0	0	0	0	0	0	0	0	0	b0



Program code is as follows:

```
unsigned char str[200];
```

```
j=0;
```

```
str[j++] = 0x1B;
str[j++] = 0x2A;
str[j++] = 32; //m=33(Height is 24dot,No magnify)
str[j++] = 12; // Bitmap width is 12 dots
str[j++] = 0;
// bitmap data

str[j++] = 0x10;str[j++] = 0x00;str[j++] = 0x20;str[j++] = 0x1F;str[j++] = 0xFF;str[j++]
= 0xE0;
str[j++] = 0x1F;str[j++] = 0xFF;str[j++] = 0xE0;str[j++] = 0x10;str[j++] = 0x20;str[j++]
= 0x20;
str[j++] = 0x10;str[j++] = 0x20;str[j++] = 0x00;str[j++] = 0x10;str[j++] = 0x30;str[j++]
= 0x00;
str[j++] = 0x10;str[j++] = 0x3C;str[j++] = 0x00;str[j++] = 0x10;str[j++] = 0x2f;str[j++]
= 0x00;
str[j++] = 0x18;str[j++] = 0x43;str[j++] = 0xC0;str[j++] = 0x0F;str[j++] = 0xC0;str[j++]
= 0xE0;
str[j++] = 0x07;str[j++] = 0x80;str[j++] = 0x20;str[j++] = 0x00;str[j++] = 0x00;str[j++]
= 0x20;
str[j++] = 0x0D;// Print out present bitmap

SendDataToPrinter(str,j);
```

GS v 0 m xL xH yL yH d1...dk

[Name] Print raster bit image

[Type] ASCII: GS v 0 m xL xH yL yH d1...dk

Decimal: 29 118 48 m xL xH yL yH d1...dk

Hex: 1D 76 30 m xL xH yL yH d1...dk

[Scope] $0 \leq m \leq 3$, $48 \leq m \leq 51$

$0 \leq xL \leq 255$

$0 \leq xH \leq 255$ where $1 \leq (xL + xH \times 256) \leq 128$

$0 \leq yL \leq 255$

$0 \leq yH \leq 8$ where $1 \leq (yL + yH \times 256) \leq 4095$

$0 \leq d \leq 255$

$k = (xL + xH \times 256) \times (yL + yH \times 256)$ ($k \neq 0$):

m	mode	Vertical resolution (DPI)	Lateral resolution (DPI)
0, 48	Normal	200	200
1, 49	Double-width	200	100
2, 50	Double-height	100	200
3, 51	Double-width、Double-height	100	100

- The xL and xH indicates the number of bytes in the horizontal direction of the bit-image

- The yL and yH indicates the number of bytes in the vertical direction of the bit-image

(Note)

- By standard, only when there is no data in the buffer area, the command effective.

- Font magnify, bold, double print, inversion printing, underline, black and white reverse, etc printing mode is invalid to the command.

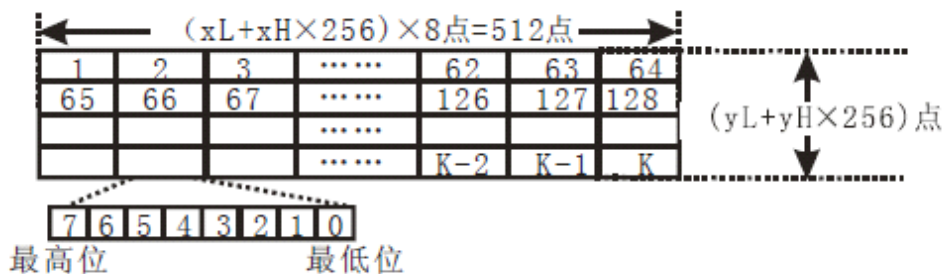
- Bitmap beyond the printing area can't be printed.

- ESC a (select align mode) Which is valid to raster and bitmap

- During Micro definition, The command will stop Micro definition and perform the command. this command is not part of micro definition.

d represent bitmap data. each byte's corresponding bit is 1 means print the point, 0 means not to print it.

[E.g] When $xL + xH \times 256 = 64$



1.2.4 Barcode command

EGS h n

[Name] Set up barcode height

[Format] ASCII GS h n

Decimal 29 104 n

Hex 1D 68 n

[Scope] $1 \leq n \leq 127$

[Description] Select barcode height

N is the vertical point counts

Default value: $n = 48$

```
[E.G] unsigned char str[4];  
      str[0] = 0x1D;  
      str[1] = 0x68;  
      str[2] = 30;  
  
      SendDataToPrinter(str,3); // Set the bar code height to 30 vertical  
dot heights
```

GS w n

[Name] 设置条形码宽度 Set up barcode Width

```
[Format] ASCII      GS w n  
          Decimal   29 119 n  
          Hex       1D 77 n
```

[Scope] $1 \leq n \leq 4$

[Description] Set the horizontal width of the bar code and n specifies the bar code width as follows:

N	Module width for multi-level bar code (mm)	Binary-level bar code	
		Thin element width (mm)	Thick element width (mm)
1	0.125	0.125	0.25
2	0.25	0.25	0.50
3	0.375	0.375	0.75
4	0.50	0.50	1.0

```
[Example] unsigned char str[4];  
          str[0] = 0x1D;  
          str[1] = 0x77;  
          str[2] = 3;  
  
          SendDataToPrinter(str,3); //Set up barcode width
```

GS H n

[Name] Select printing position for HRI characters

[Format] ASCII GS h n
 Decimal system 29 72 n
 Hexadecimal 1D 48 n

[Scope] $0 \leq n \leq 2$

[Description] Selects the printing position of HRI characters when printing a bar code.

n selects the printing position as follows



n selects the printing position as follows :

<i>n</i>	Printing position
0	Not printed
1	Above the bar code
2	Below the bar code

[Default]: $n = 0$

[E.g] `unsigned char str[4];`

`str[0] = 0x1D;`

`str[1] = 0x48;`

`str[2] = 2;`

`Send Data To Printer(str,3); // The HRI is printed below the bar code`

GS Q n

[Name] 设置条码水平打印位置 Set bar code absolute print position

[Format] ASCII GS Q nl nh
 Decimal systems 29 81 nl nh
 Hexadecimal 1D 51 nl nh

[Scope]: $0 \leq n \leq 255, 0 \leq nh \leq 2$

[Description]: Sets the distance from the beginning of the line to the position at which subsequent bar code are to be printed.

[Default]: $nl=nh= 0$

[E.g] `unsigned char str[4];`



```

str[0] = 0x1D;
str[1] = 0x51;
str[2] = 32;
str[3] = 0;

Send Data To Printer(str,4);//

```

GS k

[Name] Print barcode

[Format] The command has two format:

Format1: ($0 \leq m \leq 8$)

ASCII code: GS k m d1...dk NUL

Decimal: 29 107 m d1...dk 0

Hex: 1D 6B m d1...dk 00

Format 2: ($65 \leq m \leq 73$)

ASCII Code: GS k m n d1...dn

Decimal: 29 107 m n d1...dn

Hex: 1D 6B m n d1...dn

[Scope] $0 \leq m \leq 8$ (k and d Depends on used barcode systems) $65 \leq m \leq 73$ (n and d depends on used barcode systems)

The range of k and d are determined by the type of bar code used.

The range of n and d are determined by the type of bar code used.

The n is the data length of the bar code to be printed.

[Description] Select a bar code system and print the bar code

Use m to select a bar code system as follows:

m	Type of barcode	Length	Scope	
Format 1	0	UPC-A	$11 \leq k \leq 12$	$48 \leq d \leq 57$
	1	UPC-E	$K=8$	$48 \leq d \leq 57$
	2	JAN13 (EAN13)	$12 \leq k \leq 13$	$48 \leq d \leq 57$
	3	JAN 8 (EAN8)	$7 \leq k \leq 8$	$48 \leq d \leq 57$
	4	CODE39	$1 \leq k$	$48 \leq d \leq 57, 65 \leq d \leq 90, 32, 36, 37, 43, 45, 46, 47$
	5	ITF	$1 \leq k$ (even number)	$48 \leq d \leq 57$
	6	CODABAR	$1 \leq k$	$48 \leq d \leq 57, 65 \leq d \leq 68, 36, 43, 45, 46, 47, 58$



Format 2	65	UPC-A	$11 \leq n \leq 12$	$48 \leq d \leq 57$
	66	UPC-E	$n=8$	$48 \leq d \leq 57$
	67	JAN13 (EAN13)	$12 \leq n \leq 13$	$48 \leq d \leq 57$
	68	JAN 8 (EAN8)	$7 \leq n \leq 8$	$48 \leq d \leq 57$
	69	CODE39	$1 \leq n \leq 255$	$48 \leq d \leq 57, 65 \leq d \leq 90, 32, 36, 37, 43, 45, 46, 47$
	70	ITF	$1 \leq n \leq 255$ (even number)	$48 \leq d \leq 57$
	71	CODABAR	$1 \leq n \leq 255$	$48 \leq d \leq 57, 65 \leq d \leq 68, 36, 43, 45, 46, 47, 58$
	72	CODE93	$1 \leq n \leq 255$	$0 \leq d \leq 127$
	73	CODE128	$2 \leq n \leq 255$	$0 \leq d \leq 127$

*Except UPC-E, Other barcode's Parity bit all is auto calculated by printer, The user may not add Parity bit.

*CODE39 don't need add *

[Note] • When using the format 1 command, if the bar code type specifies the data length of the bar code, k (the barcode data length received by the printer) should be equal to the specified data length, and if not equal to the specified data length, the instruction is invalid. See the related barcode data bit length [Appendix B].

- The barcode data received by the printer should be included in the character set specified by the bar code, if some characters of the bar code data characters are outside the character set, the command is invalid. See the related barcode character set [Appendix B].
- When using the format 2 command, the value of n should be equal to the specified data length (if the kind of bar code specifies the data bit length). And if the value of n is not equal to the specified data bit length, the command is invalid. See the related barcode data bit length [Appendix B].
- The number of ITF code data length must be even numbers. If using the format 1 to print ITF bar code, the value of k should be even numbers, but if it is odd number, the last one bit data will be ignored. If using the format 2 to print ITF bar code, the value of n should be even numbers, but if it is odd number, the last one bit

data will be ignored.

- If the bar code on the horizontal direction exceeds the printable area, it is invalid
- The command is not affected by the print modes (Eg: emphasized, double-strike print, underline, character size, or white/black reverse printing, etc.)
- Printing barcode need obey the barcode specifications, or will cause that the bar code cannot be scanned.
- The printer does not calculate the checksum, but if barcode needs the checksum, the checksum should be included in the bar code data, and the printer is not responsible for checking whether the checksum is wrong or right. The user calculates the checksum, and if it is wrong, it will cause that the bar code cannot be scanned.
- CODE39 code does not include the extended CODE39 code (EXTERN CODE 93)
- CODE93 code does not include the extended CODE93 code (EXTERN CODE 93).
- When using the CODE128, must first select the character set (CODE A, CODE B or CODE C) before the barcode data. Select the character set through sending the character "{" and another character; the ASCII code characters " {" is defined by sending " {" twice consecutively. Details see following Chart:

ASCII	HEX	Function
{A	7B, 41	Select the code set A
{B	7B, 42	Select the code set B
{C	7B, 43	Select the code set C
{S	7B, 53	SHIFT
{1	7B, 31	FNC1
{2	7B, 32	FNC2
{3	7B, 33	FNC3
{4	7B, 34	FNC4



[Name] QR Magnify Command

[Format] ASCII GS W *n*

Decimal 29 87 *n*

Hex 1D 57 *n*

[Scope] $1 \leq n \leq 8$

[Description] Set up QR Code Magnification times

Default value: $n = 2$

[E.G] `unsigned char str[4];`

`str[0] = 0x1D;`

`str[1] = 0x57;`

`str[2] = 06;`

`Send Data To Printer(str,3); // Set up 6 times magnification`

GS k m v r d1.....dk

[Name] Print QR CODE

[Format] The command has two type of format:

Format 1 $m=32$

ASCII code GS k m v r d1...dk NUL

Decimal code 29 107 m v r d1...dk 0

Hexadecimal code 1D 6B m v r d1...dk 00

Format 2 $m=97$

ASCII Code GS k m v r nL nH d1...dn

Decimal code 29 107 m v r nL nH d1...dn

Hexadecimal code 1D 6B m v r nL nH d1...dn

[Scope] $m=32$ or 97

$1 \leq v \leq 20$ $1 \leq r \leq 4$

[Description] v is DQCODE version number

r=1 Error Correction level is L r=2 Error Correction level M r=3 Error Correction level

is Q r=4 Error Correction level is H

nL,nH

nL,nH is The Integer N's low order and high order, N is the barcode data length,

The unit is byte.

When using the format 1, Command ends as 00, d1...dk is the barcode data. when using format 2, The printer will put N Characters(d1...dn) behind nH as the barcode data.

- [Notice] • As the printing paper width limits, The max version number of QR CODE is 20.
- ISO/IEC 18004:2000. For the detailed QR CODE Standard, pls see Chinese National Standard GB/T 18284-2000 Or ISO Standard ISO/IEC 18004:2000

[E.G]

```
unsigned char str[16];  
  
str[0] = 0x1D; str[1] = 0x6B; str[2] = 32;  
  
str[3] = 1; // Version No.is 1  
  
str[4] = 2; // Error Correction Level is M  
  
str[5] = '1'; str[6] = '2'; str[7] = '3'; str[8] = '4'; str[9] = '5'; str[10] = '6'; str[11] = '7';  
  
str[12] = '8'; str[13] = '9'; str[14] = '0';  
  
SendDataToPrinter ( str, 5);
```

1.2.5 Curve command

ESC ‘

[Name] 打印一水平行上n个点. Print n dots on one horizontal line

[Format] ASCII 码 GS ‘ nL nH x1L x1H x21L x21H xkL xkH CR
Decimal 29 39 nL nH x1L x1H x21L x21H xkL xkH 13
Hexadecimal 1B 27 nL nH x1L x1H x21L x21H xkL xkH 0D

[Scope] $0 \leq nL \leq 255$

$0 \leq nHL \leq 1$

Number of Curve Dots $N = nH \times 256 + nL$

曲线点在水平行上的位置 $X = xkH \times 256 + xkL$ 。

[描述] 每条曲线都是由很多点组成。本指令为打印一水平行上 n 个点，连续使用该指令可以打印出用户所需要的曲线。

[例子] 曲线关系函数为下面 5 个函数

$Y1=50+40*\text{abs}(-0.01*X)*\sin(X/10)$

$$Y2=50-40*\text{abs}(-0.01*X)*\sin(X/10)$$
$$Y3=50$$
$$Y4=50+40*\text{abs}(-0.1*X)$$
$$Y5=50-40*\text{abs}(-0.01*X)$$

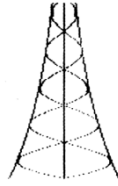
[打印例程]:

```
unsigned char str[50];
float X;
unsigned int m_cur1,m_cur2,i;
for(X=0;X<150;X++) //打印150点行
{
    m_cur1= 40*exp(-0.01*X);
    YY= Y*sin(X/10);
    str[i++] = 0x1b;
    str[i++] = 0x27;
    str[i++] = 0x5; //打印5条曲线
    str[i++] = 0x0;
    str[i++] = 50+m_cur2;
    str[i++] = 0;
    str[i++] = 50-m_cur2;
    str[i++] = 0;
    str[i++] = 50;
    str[i++] = 0;
    str[i++] = 50+m_cur1;
    str[i++] = 0;
    str[i++] = 50-m_cur1;
    str[i++] = 0;
    str[i++] = 0x0D;

    Send Data To Printer(str,i); //设置可识读字符在条码的下方打印。
}
```

[Result]:

WWW.RD-CN.COM



1.2.6 Status transmission command_

ESC v

[Name] Forward printer status to mainframe

[Format] ASCII ESC v

Decimal 27 118

Hexadecimal 1B 76

[Description] Forward Printer status to mainframe

[Note]• Only valid to serial port printer

Bit	Function	Value	
		0	1
0	Paper detection	Paper	No paper
1	Work Status	Free	Printing
2	Buffer	Not full	Full
3	Printer status	Normal	Error
4	Undefined	---	---
5	Undefined	---	---
6	Undefined	---	---
7	Undefined	---	---

[E.G] `unsigned char str[4];`

`str[0] = 0x1B;`

`str[1] = 0x76;`

`SendDataToPrinter(str,2); //Send inquiry command to printer`

1.2.7 Kanji Character set up command

FS &

[Name] Enter Kanji mode

[Format] ASCII FS &
Decimal 28 38
Hexadecimal 1C 26

[Description] Printer enter into Kanji printing mode

[Note]• Printer default is Kanji printing model.

[E.G] `unsigned char str[4];`
`str[0] = 0x1C;`
`str[1] = 0x26; Send Data To Printer(str,2); // Enter printer Kanji`
`model`

FS.

[Name] Cancel Kanji model

[Format] ASCII FS .
Decimal 28 46
Hexadecimal 1C 2E

[Description]: When Cancel Kanji mode, The code beyond 0x80 still shall be taken as ASCII Character to deal with, But it will not print Kanji, Unless recover FS& Command and select Kanji model, Printer will enter Kanji printing model.

[E.G] `unsigned char str[4];`
`str[0] = 0x1C;`
`str[1] = 0x2E;`
`Send Data To Printer(str,2); // Enter ASCII Character and printing`
`model.`

ESC 6

[Name] Select 6x8 character set 1

[Format] ASCII ESC 6
Decimal 27 54
Hexadecimal 1B 36

[Description]: 20H~FFH(32~255)。 After the command input, all the character will be selected from Character set 1(see appendix D),224 Character of 6x8 dot are covered, Including ASCII and various Graphic symbols, etc. code range: 20H~FFH(32~255)

[E.G] `unsigned char str[4];`
`str[0] = 0x1B;`
`str[1] = 0x36;`
`Send Data To Printer(str,2); // Print 6x8 character set 1`

ESC 7

[Name] **Select 6x8 character 2**

[Format] ASCII ESC 7
 Decimal 27 55
 Hexadecimal 1B 37

[Description] 20H~FFH(32~255)。 After the command input, all the character will be selected from Character set 1(see appendix D),224 Character of 6x8 dot are covered, Including German, French, Russian, Japanese Katakana ,etc. code range: 20H~FFH(32~255)

[E.G] `unsigned char str[4];`
`str[0] = 0x1B;`
`str[1] = 0x37;`
`SendDataToPrinter(str,2); // Print 6x8 character set2 fonts`

ESC ! n

[Name] **Set up Character Printing Mode**

[Format] ASCII码 ESC ! n
 Decimal 27 33 n
 Hexadecimal 1B 21 n

[Scope] $0 \leq n \leq 255$

[Statement] n decide the character printing model set up.

Bits	0/1	Decimal	Hexadecimal	Function
0, 1, 2	0	0	0	Default 0
3	0	0	00	Cancel bold pattern
	1	8	08	Select bold pattern

4	0	0	00	Cancel double height pattern
	1	16	10	Select double height pattern
5	0	0	0	Cancel double width pattern
	1	32	20	Select double width pattern
6 , 7	0	0	0	Default 0

[Note] • When double-width and double-height modes are selected, the characters will be horizontally and vertically enlarged two times.

• When some characters in a row is double-width or double-height, all the characters are aligned with the bottom.

• The bold pattern are valid for English and Chinese characters. In addition to the bold pattern, all printing modes are only valid for the English and figure characters.

[Default] $n = 0$

FS Y

[Name] Print Kanji and Character as per UNICODE

[Format] ASCII FS Y NL NH
 Decimal 28 89 NL NH
 Hexadecimal 1C 59 NL NH

[Description] When Kanji is printed as per UNICODE, NL NH indicate the number of Kanji and characters (NL means low eight number, NH high eight number)

[E.G] Send the following hexadecimal data

1C 59 0B 00 00 55 00 4E 00 49 00 43 00 4F 00 44 00 45 62 53 53 70 6D 4B 8B D5
0D 0D 0D 0D

Printer will print " UNICODE Printing Test".

1. 2. 8 垂直制表并打印指令

FS V

[Name] Vertical Tab And Print

[Format] ASCII FS V
 Decimal 28 86 m LP1...LPm n IP1...IPn FT1 D11...D1k 0...FTn Dn1...Dnk 0
 Hex 1C 56 m LP1...LPm n IP1...IPn FT1 D11...D1k 0...FTn Dn1...Dnk 0

[Statement] m Vertical Lines: $0 \leq m \leq 17$;

LP1...LPm Vertical Line Coordinate: $0 \leq LPm \leq 48$;

n Table Body Numbers: $0 \leq n \leq 16$;

IP1...IPn Table body Coordinate: $0 \leq IPn \leq 45$;

FT1 First Table Body Font Type:

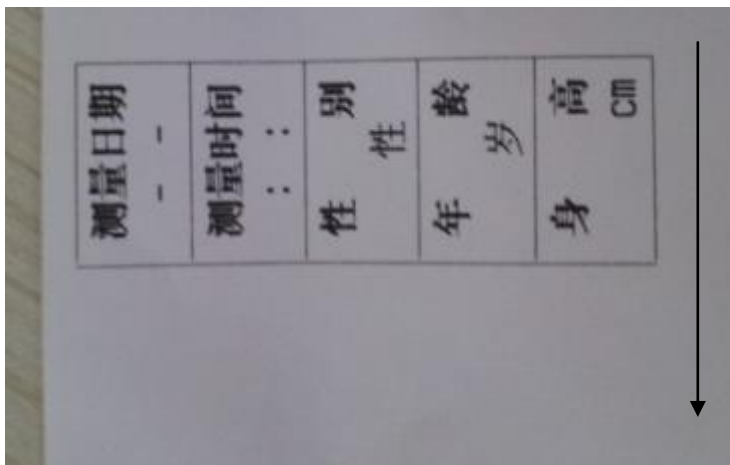
Position	Function	Value	
		0	1
0	Reserve		
1	Thicker	Cancel	Set
2	Underline	Cancel	Set
3	Reverse	Cancel	Set
4-7	Reserve		

[Example]

```

1C 56 06 00 09 12 1B 24 2D
0A 01 05 0A 0E 13 17 1C 20 25 29
02 20 B2 E2 C1 BF C8 D5 C6 DA 20 00
01 20 20 20 2D 20 20 2D 20 20 00
02 20 B2 E2 C1 BF CA B1 BC E4 00
01 20 20 20 3A 20 20 3A 20 20 00
02 20 D0 D4 20 20 20 20 B1 F0 00
01 20 20 20 20 20 D0 D4 00
02 20 C4 EA 20 20 20 20 C1 E4 00
01 20 20 20 20 20 CB EA 00
02 20 C9 ED 20 20 20 20 B8 DF 00
01 20 20 20 20 20 20 20 63 6D 00
    
```

[Result]:



[Note]:

This command only support panel printer, the desktop printer need to send 1B 63 01

Command before using this command.

1. 2. 9 Bitmap Download And Print Command

GS * x y d1...dk

[Name] Define The Bitmap

[Format] ASCII GS * x y d1...dk

Decimal 29 42 x y d1...dk

Hex 1D 2A x y d1...dk

[Range] $1 \leq x \leq 72$ $1 \leq y \leq 20$ $x \times y \leq 1024$ $k=x*y*8$

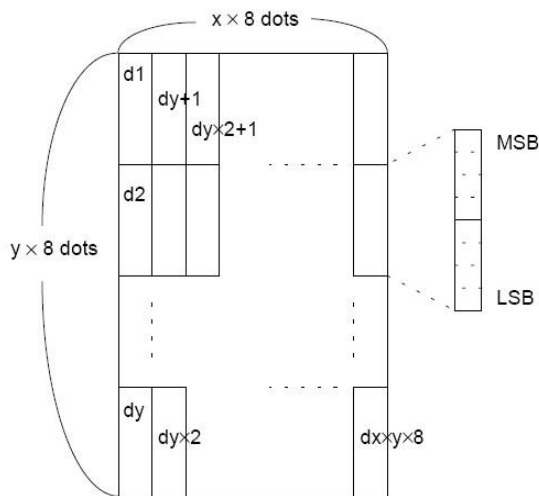
[Explanation] Use x and y appointed count to define the bitmap

- $x*8$ is Horizontal Direction Count.
- $y*8$ is vertical direction Count.

[Notice] • As the buffer area limit, if $x \times y$ is *beyond the appointed range*, The command may occur unexpected result.

• d is expressed as bitmap data. d1,d2...dn appointed printing equals 1,Non printing equals 0.

- the defined bitmap by this command is printed by **GS / n Order**.



**GS / n**

[Name] Print the bitmap

[Format] ASCII GS / n

Decimal 29 47 n

Hex 1D 2F n

[Range] $0 \leq n \leq 3$

[Description] Print the bitmap by designed Mode defined GS command.

n Mode is selected from following list:

n	Amplifying Mode
0	Normal
1	Double Width
2	Double Height
3	Double Width And Height

FS q n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n

[Name] Download Multiple NV Bitmap

[Format] ASCII FS q n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n

Decimal 28 113 n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n

Hex 1C 71 n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n

[Range]: 1 . n . 255

0 . xL . 255

0 . xH . 3 (when 1 . (xL xH*256) . 1023

0 . yL . 255

0 . yH . 1 (when 1 . (yL yH*256) . 288

0 . d . 255

 $k = (xL xH * 256) * (yL yH * 256) * 8$

The overall defined Graphic Data is 150K bytes

[Description] : • n is the designed Graphic downloaded Numbers.

• xL,xH indicates the bitmap horizontal width is $(xL xH / 256) * 8$ dots• yL,yH indicates the bitmap Vertical Height is $(yL yH / 256) * 8$ dots

• d is the graphic data.

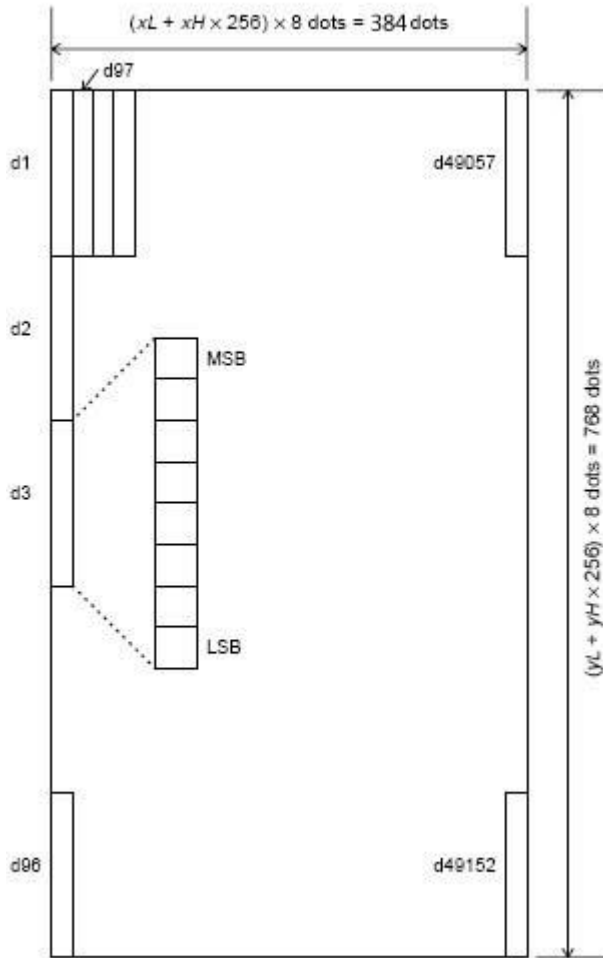
[Notice] : • Graphic Horizontal And Vertical Direction dots number is the time of 8

• The downloaded bitmap defined by this command is printed by **FS p n m**

[Example]:

: 当xL = 48, xH = 0, yL = 96, yH = 0

当xL = 48, xH = 0, yL = 96, yH = 0



FS p nm

[Name] Print The Downloaded NV bitmap

[Format] ASCII FS p n m

Decimal 28 112 n m

Hex 1C 70 n m

[Range] $1 \leq n \leq 255$

[Description] N is the number of Downloaded Bitmap Defined By FS q Command

m Is The Choice Model Selected From Following List:

n	Zoom Model
0,48	Normal
1,49	Double Width
2,50	Double Height
3,51	Double Width and Height

1.2.10 Printing Darkness Set up command

ESC s

[Name] Printing darkness adjustment

WWW.RD-CN.COM

[Format] ASCII: ESC s 2B/2D n
 Decimal: 27 114 43/45 n
 Hexadecimal: 1B 73 2B/2D n

[Explanation]

$0 < N \leq 255$

When put into use 1B 73 2B n means increasing darkness adjustment, more high digit n is, the printing is more darker

When put into use 1B 73 2D n means reducing darkness adjustment, more high digit n is, the printing is more lighter (ESC <0X25)

Restore command is 1B 73 2D 0 or 1B 73 2B 0

[Note] • **The user can adjust the darkness as per needs, when lighter, it will reduce the power consumption of printer, and darker will increase printer consumption.**

• **The user need to adjust printing darkness as per the (motor and model) when use this command, the overadjustment will make printer work unnormally.**

1.2.11 Label Command(for label command only)

ESC i

[Name] Go to next label position

[Format] ASCII ESC i
 Decimal 27 105
 Hexadecimal 1B 69

[Description]: Feeding to next label head

[Note]• The Command only apply to label printer

[Example] unsigned char str[4];
 str[0] = 0x1B;
 str[1] = 0x69;
 SendDataToPrinter(str,2);// Send full cutting command.

ESC m

[Name] Half cutting

[Type] ASCII: ESC i
 Decimal: 27 109
 Hex: 1B 6D

[Explanation] cutter: half cutting

[Comment]

- This command does not cause feeding line.
- This command is only used in the printers with cutter.

[Example] unsigned char str[4];
 str[0] = 0x1B;
 str[1] = 0x6D;
 SendDataToPrinter(str,2);// Send the half-cutting command
 str[0] = 0x1B;
 str[1] = 0x6D;
 SendDataToPrinter(str,2);//Send half cut command

1.2.12 International Character Set

ESC R n

[Name] Select International Character Set

[Format] ASCII Code ESC R n

Hexdecimal Code 1B 52 n

Decimal Code 27 82 n

[Range] 0 . n . 13

[Description] Select Following Value n to set up International character set

[Default-Value] n = 0

[Note] •The command is only for 24 dot printer

n	Character Set
0	USA
1	France
2	German
3	UK
4	Denmark I
5	Sweden
6	Italy
7	Spain
8	Japan
9	Norway
10	Demark II
11	Spain II
12	Latin America
13	Korea
14	Slovenia
15	China

[Character Position Difference Table]

County	ASCII Code(Hex)											
	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
U.S.A.	#	\$	@	[\]	^	`	{		}	~



France	#	\$	à	°	ç	§	^	`	é	ù	è	¨
Germany	#	\$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
U.K.	£	\$	@	[\]	^	`	{		}	~
Denmark I	#	\$	@	Æ	Ø	Å	^	`	æ	ø	å	~
Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Italy	#	\$	@	°	\	é	^	ù	à	ò	è	ì
Spain I	Pt	\$	@	í	Ñ	¿	^	`	¨	ñ	}	~
Japan	#	\$	@	[¥]	^	`	{		}	~
Norway	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
Denmark II	#	\$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
Spain II	#	\$	á	í	Ñ	¿	é	`	í	ñ	ó	ú
Latin	#	\$	á	í	Ñ	¿	é	ü	í	ñ	ó	ú
Korea	#	\$	@	[₩]	^	`	{		}	~
Slovenia/Croatia	#	\$	Ž	Š	Đ	Ć	Č	ž	š	đ	ć	č
China	#	¥	@	[\]	^	`	{		}	~

ESC t n

[Name] Select Character code table

[Format] ASCII码ESC t n

Hexadecimal Code 1D 74 n

Decimal Code 29116n

[Range] 0 ≤ n ≤ 21, 32 ≤ n ≤ 34, 64 ≤ n ≤ 79,

[Description] Select n from Character Code table

[Note] • The command is only for 24 dot printer.

N(Decima)	Code table	N(Decima)	Code table
0	Normal*	32	Codepage 1252 (Windows Latin-1)
1	CodePage437 (USA, Std. Europe)	33	Codepage 1250 (Windows Latin-2)
2	Katakana	34	Codepage 1251 (Windows Cyrillic)
3	CodePage437 (USA, Std. Europe)	64	Codepage 3840 (IBM-Russian)
4	Codepage 858 (Multilingual)	65	Codepage 3841 (Gost)
5	Codepage 852 (Latin-2)	66	Codepage 3843 (Polish)
6	Codepage 860 (Portuguese)	67	Codepage 3844 (CS2)
7	Codepage 861 (Icelandic)	68	Codepage 3845 (Hungarian)
8	Codepage 863 (Canadian French)	69	Codepage 3846 (Turkish)
9	Codepage 865 (Nordic)	70	Codepage 3847 (Brazil-ABNT)
10	Codepage 866 (Cyrillic Russian)	71	Codepage 3848 (Brazil-ABICOMP)
11	Codepage 855 (Cyrillic Bulgarian)	72	Codepage 1001 (Arabic)

12	Codepage 857 (Turkey)	73	Codepage 2001 (Lithuanian-KBL)
13	Codepage 862 (Israel (Hebrew))	74	Codepage 3001 (Estonian-1)
14	Codepage 864 (Arabic)	75	Codepage 3002 (Estonian-2)
15	Codepage 737 (Greek)	76	Codepage 3011 (Latvian-1)
16	Codepage 851 (Greek)	77	Codepage 3012 (Latvian-2)
17	Codepage 869 (Greek)	78	Codepage 3021 (Bulgarian)
18	Codepage 928 (Greek)	79	Codepage 3041 (Maltese)
19	Codepage 772 (Lithuanian)		
20	Codepage 774 (Lithuanian)		
21	Codepage 874 (Thai)		

1.2.13 Consecutive curve printing command

ESC) n

[Name] Print N segments of same Horizontal line

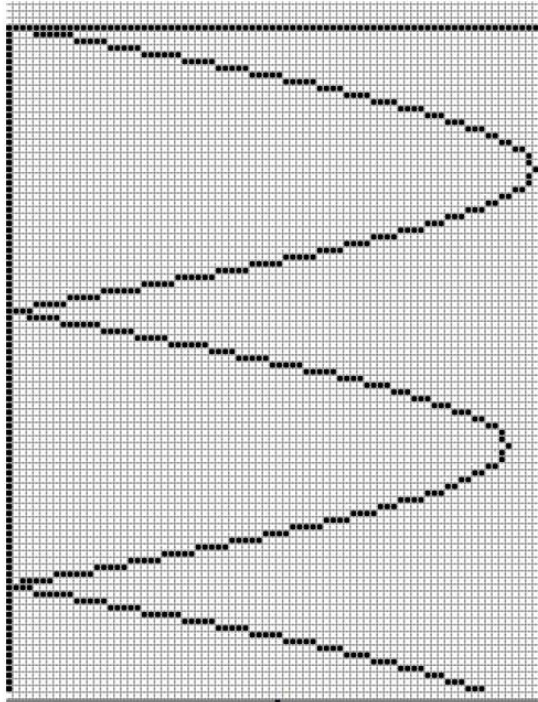
[Format] ASCII Code ESC) n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH 00

Decimal 2741 n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH 00

Hex 1B 29 n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH 00

[Range] $0 \leq n \leq 255$

[Description] As following Enlarged Bitmap showing: Each Curve is consisted of many horizontal line segment(dots seen as length as 1). The command is to Print N segments of same Horizontal line, Consecutive command use can print the demand segment.



n Segment lines number;

xksL The low point of the horizontal coordinate at the starting point of the k line segment

xksH The High point of the horizontal coordinate at the starting point of the k line segment

xkeL The low point of the horizontal coordinate at the close point of the k line segment

xkeH The High point of the horizontal coordinate at the close point of the k line segment

;

Counting from the left of printing area, The coordinate minimum value is 1, maximum value is 384, so $xkeL+xkeH*256$ max value can be 384.

Segment line data doesn't need count as order.

[Note] • When printing a dot , $xkeL=xksL$, $xkeH=xksH$.

```
char SendStr[8];
```

```
int i;
```

```
short y1,y2,y1s,y2s;
```

```
// Print y axis(One line)
```

```
SendStr[0] = 0x1B;
```

```
SendStr[1] = 0x29;
```

```
SendStr[2] = 1; //one line
```

```
SendStr[3] = 30; //starting point is 30
```

```
SendStr[4] = 0;
```

```
SendStr[5] = 104; //Closing point is 360
```

```
SendStr[6] = 1;
```

```
SendDataToPrinter(SendStr,7);
```

ESC # n

[Name] Set up curve printing mode

[Format]	ASCII	ESC	#	<i>n</i>
	Decimal	27	35	<i>n</i>
	Hex	1B	23	<i>n</i>

[Range] $0 \leq n \leq 1$

[Description] The command affects 1B 28 Printing mode.

n = 0 1B 28 Curve command connects point non automatically.

n = 1 1B 28 Curve command connects point automatically

[Note] After printer accepts the command, it will put the first curve command which will be received as its initial point.

ESC (n

[Name] Print *n* curve command

[Format]	ASCII code	ESC (<i>n</i> <i>x1L</i> <i>x1H</i> ... <i>xnL</i> <i>xnH</i> 00
	Decimal	2740 <i>n</i> <i>x1L</i> <i>x1H</i> ... <i>xnL</i> <i>xnH</i> 00
	Hex	1B 28 <i>n</i> <i>x1L</i> <i>x1H</i> ... <i>xnL</i> <i>xnH</i> 00

[Range] $0 \leq n \leq 20$

[Description] *n* is the demanded curve printing numbers.

xnL The lower 8 bits of the horizontal coordinate of curve point

xnH The Higher 8 bits of the horizontal coordinate of curve point

[Description] Print *n* curves according to *n* value set up.

Auto points connection of each curve.(Controlled by ESC# command)

Set *n*=1 when printing one curve.

```
char SendStr[8];
```

```
int i;
```

```
SendStr[0] = 0x1B;
```

```
SendStr[1] = 0x28;
```

```
SendStr[2] = 2; // Print out 2 curves.
```

```
SendStr[3] = 10;
```

```
SendStr[4] = 0;
```

```
SendStr[5] = 50;
```

```
SendStr[6] = 0;
```

```
SendDataToPrinter(SendStr,7);
```

ESC= n

[Name] Set up curve printing mode

[Format]	ASCII	ESC	#	<i>n</i>
	Decimal	27	61	<i>n</i>
	Hex	1B	3D	<i>n</i>



[scope] $0 \leq n \leq 1$

[description] The command affects 1B 63

$n = 0$ Quit Driver printing mode 1B 63 Command will be ok

$n = 1$ Enter the driver mode,the 1B 63 parameter is two bytes. At the same time, this command cant be used as the forward and reverse function. [Note] This command is only valid for models that default to the drive mode.