RD-FH5C7

Embedded thermal printers specification of Development

目录

Chapter 1: Overview

The printing paper width of RD-FH5C7 THERMAL PRINTER is 57mm. Equipped with a cutter, and supports the full cutting or half cutting function. The series printer increases sheet metal casing on the basis of the W series, which is more convenient to install for customers. The printer uses fully enclosed large paper warehouse design, which can effectively prevent rain and dust from entering.

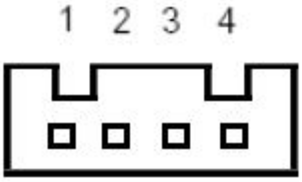
This series take into account unattended printing capabilities, and in order to prevent inadvertent phenomenon, the paper and key functions are placed inside the printer. Only when the panel is opened, the printer can change paper and do the button operation. Meanwhile, also equip with a safety lock on the printer panel, after the lock to the printer's front panel will not be opened, and at this time only the professional staff with the key can open the front panel.

The series printers have two models: RD-FH5C7-P and RD- FH5C7-S. The former is a 26-pin standard parallel printer; and the latter is 5-pin RS232 serial interface printer. The series printer has the printing paper width, high speed, small size and other characteristics. They are used in the Bank Queue, supermarkets lockers, game consoles, subway coupons and other equipments.

1.1Power connector

Power Interface is the 4PIN, which is 2.54mm at equal distance and perpendicular.

pin	name	description
1	GND	GND
2	+24V	Positive power supply, 24V ± 1V The average supply current : 2A, the MAX : 4A
3	+24V	
4	GND	GND



1.2Performance indicators

printing method	thermal line printing
Printing width	58mm
Roll paper diameter	Unlimited
Printing line width	36 English characters / line (standard 24 dot array high fonts)
Dots per line	432dots/54mm
Kanji Character Encoding	GBK
Built-in font	ASCII font (16×8、24×12、8×6)、GBK(24×24)
Printing speed	200mm/s

Data Interface	RD-FH5C7-S	Standard serial interface （S）（RS232 or TTL）
	RD-FH5C7-P	standard parallel interface（P）
Resolution	203 dpi ×203 dpi（8 dots / mm）	
Instruction system	ESC/PON (Compatible instruction system)	
Printable contents	English, numbers, symbols, characters, graphics, curves, bar code (CODE39、EAN13、EAN8、 CODABAR、CODE128、CODE93、 ITF)	
Power supply	24V/2A	
Print head life	100km	
Anomaly detection	lack of paper detection	
Black Label positioning function	Black Label detection	
Cut type	Full cut or half cut	
Operating temperature / humidity	-10～50℃/20～85%RH(RH)	

1.3operating

There is an indication button on the printer's panel, and this button is not only an indicator but also a function key.

1.3.1operating key

FEED button: Click the button and the printer feed paper one line. Hold down the button and the printer continuously feed paper.

1.3.2 Operation

1. self- test

Step 1: The power of the printer is turned off.

Step 2: Press the indicator button

Step 3: Energize the printer for about 2 seconds and the printer will start the self-test.

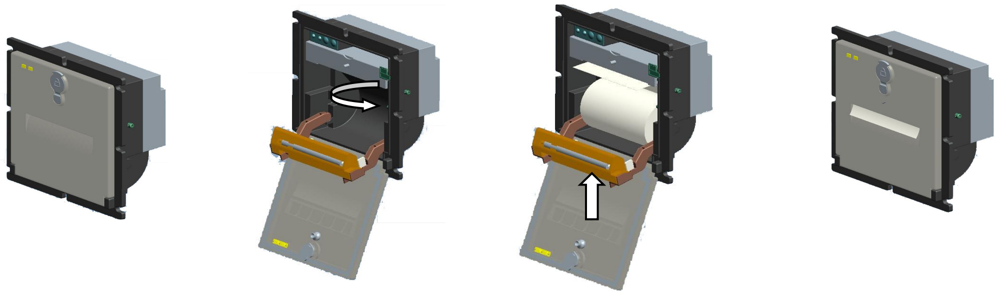
2. Load paper

Step 1: Open the security lock of the printer, and open the printer’s cover.

Step 2: Pull out the right wrench of the printer, and open the printer's creel.

Step 3: Remove the printer’s paper axis and insert the roll paper. Then put the paper into the paper storehouse, and get stuck the paper axis, and fasten the printer creel.

Step 4: Pull out the paper end, and fasten the cover. End the loading paper.



3 Paper feed

Under the printer charged state, hold down the buttons, and the printing paper starts feeding.

1.3.3Indicator light

Power Light:

When the printer is powered, the indicator light is bright with green light for long.

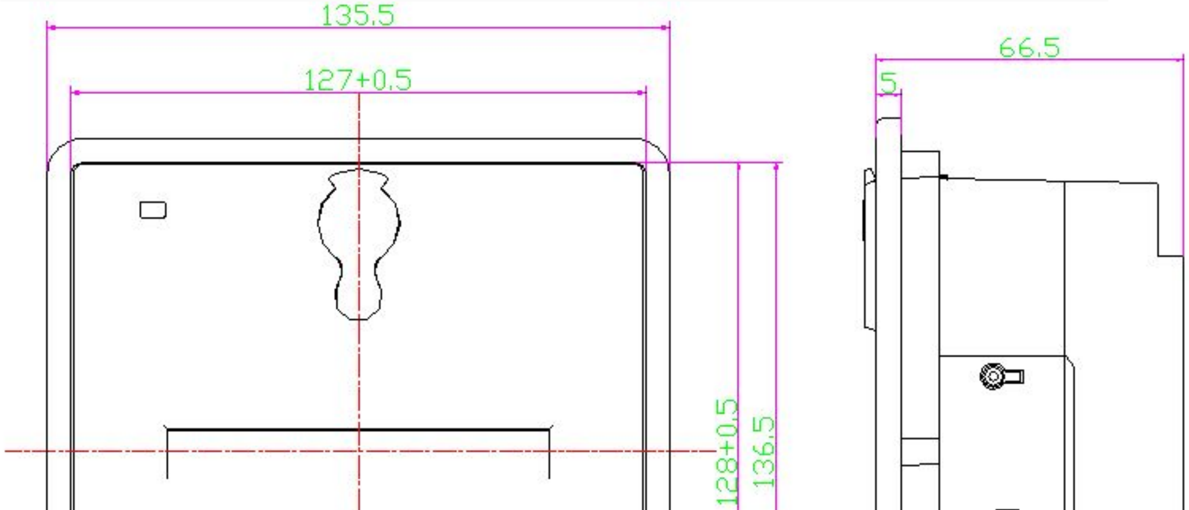
Status Light:

When the printer lacks the paper, the indicator light will be in a state of continuous glint with green light.

1.4 Appearance size and installation dimensions (Unit: mm)

Appearance size: feet inches length 135.5 mm * width 135.5 mm * depth 66.5 mm

Installation dimensions: length 127.5 mm * width 128.5 mm



Chapter 2: communication interface

2.1 serial interface

2.1.1Define of Interface

Data transfer: Serial

Synchronization way: Asynchronous

Handshake signals: CTS / RTS, DTR / DSR or XON / XOFF

Baud Rate: 1200,2400,4800,9600,19200,38400,57600,115200 bps. They can be changed by DIP switch selection.

Data Length: 8Bit Parity: None Stopping bits: 1

Interface: 5pin needle socket

(Figure 2-1 is the 5P communication interface, Figure 2-2 is the DIP switch figure)

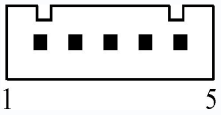


figure2-1

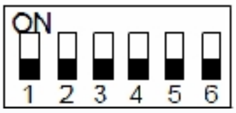


figure2-2

5 core single row socket (Pin No.)	Signal name	signal source	direction	Illustration
2	TXD	printer	Export(out)	The printer's dashboard receives the data from the host.
3	RXD	host	Import(in	The printer's dashboard

			)	sends data to the host. (When using the ‘X-ON/X-OFF’ Handshake Protocol，the printer sends control code ‘X-ON/X-OFF’ to the computer.)
4	CTS	printer	Export(out)	When the signal is in a state of ‘MARK’, it means that the printer is busy and can’t receive data. But when the signal is in a state of ‘SPACE’, it means that the printer is ready to receive data.
5	GND	——	——	Signal ground
1	NC	——	——	no connection

2.1.2 Baud rate select bit

SW NO.	1200	2400	4800	9600	19200	38400	57600	115200
1	off	on	off	on	off	on	off	on
2	off	off	on	on	off	off	on	on
3	off	off	off	off	on	on	on	on

2.1.3handshake method select bit

The handshake methods can be selected by the DIP switch SW4 in the machine. When the machines leave factory, the location of the sw4 is ‘OFF’ (Default).

SW4	Handshake method	Data direction	RS-232 interface signal
ON	Mark control	Data can enter	The signal line ‘4’ is in a state of ‘SPACE’
		Data can’t enter	The signal line ‘4’ is in a state of ‘mark’
OFF	X-ON/X-OFF	Data can enter	The signal line ‘2’send out the X-ON code ‘11H’

	control	Data can’t enter	The signal line ‘2’send out the X-OFF code ‘13H’

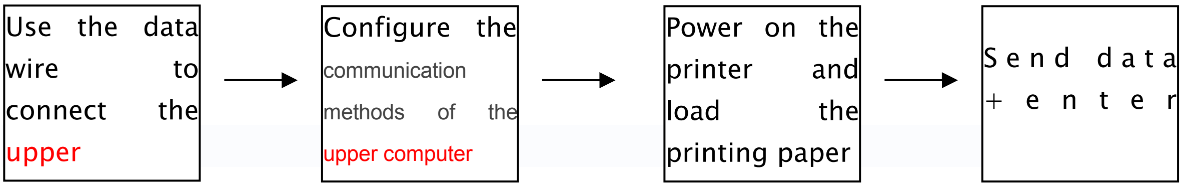
2.1.4Lack of paper selecting bit

SW5	说　明
ON	When the printer is lack of paper, it will stop printing, and online the indicator light glows green and will be continuous glint to alert the lack of paper.
OFF	When it is OFF, the printer will not respond for the lack of paper.

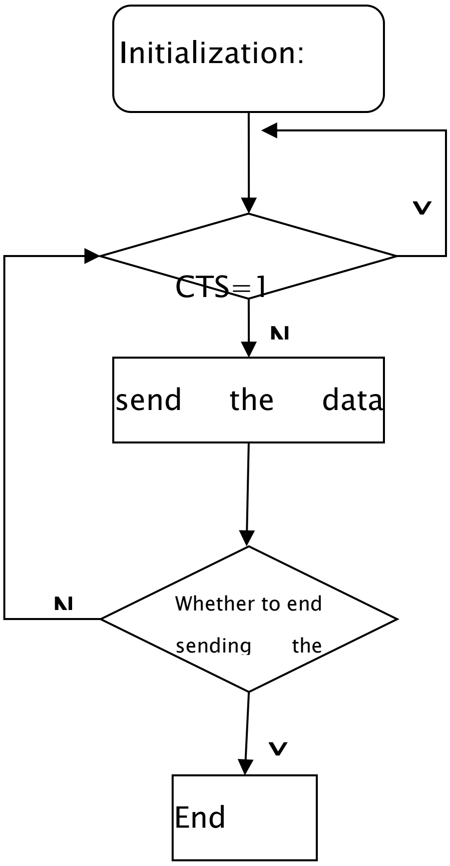
2.1.5Data transmission method of serial interface

The receiving buffer of the printer is 16K

(1) When the number of sending data once is less than 16K, the data can be directly sent. And the sending method is as follows:


Use the data wire to connect the upper
Configure the communication methods of the upper computer
Power on the printer and load the printing paper
Send data + enter

(2) If large amounts of data once is sent, need to judge the mark ‘CTS’ when sending the data. When the mark is ‘1’, the data can’t be sent. When the mark is ‘0’, the data can be sent. Data can be sent in the form of packets or byte. When the data is sent in the form of packets, each data packet can’t exceed 256 bytes, and the sending flowchart is as follows:



2.2Parallel interface

The RD-FH5C7 system thermal parallel interface printer uses 26P double-row needle socket as the communication interface, and the spacing between the needles is 2.54mm. Figure 2-3 is the parallel interface figure.

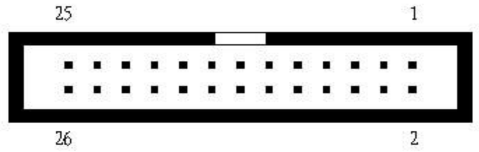


Figure 2-3

2.2.1data interface

26pin row socket (pin NO.)	**Signal name**	**Signal direction**	**Illustration**	**DB25** parallel interface wire on the PC (pin NO.)
1	STB/	in	数据选通触发脉冲，上升沿时读入数据	1
3	DATA1	In	8 data pins, the logic ‘0’ indicates low, ‘1’ indicates high	2
5	DATA2	In		3
7	DATA3	In		4
9	DATA4	In		5
11	DATA5	in		6
13	DATA6	In		7
15	DATA7	In		8
17	DATA8	In		9
19	ACK/	Out	Answer pulse, and "Low" level indicates that the data has been accepted	10
21	BUSY	Out	"High" level indicates that the printer is busy and can’t receive data	11
23	PE	——	Grounding (PERROR)	--
25	SEL	Out	经电阻上拉“高”表示打印机在线	13
4	ERR/	Out	经电阻上拉“高”电平表示无故障	15
2，6，8	NC	——	no connection	--
10～24	GND	——	Signal ground	12, 25

Matters need attention: ①The mainframe and printer in the item ‘signal source’ means the source of the signal sending out

②The signal logic level is EIA level.

2.2.2 Parallel interface data transmission method

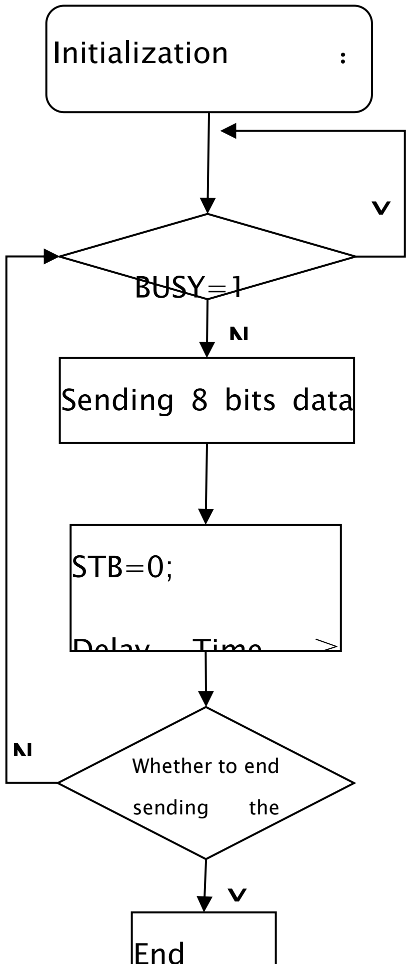
Using parallel interface to send data is relatively more trouble than using serial interface, because using parallel interface to send data need the cooperation of the 'STB','BUSY' AND 'DATA' data wires.

Sending step:

step1: using the appropriate data wire connects the printer to PC

step2:power on the printer and load the printing paper

step3:Begin to transmit data，and flow chart is as follows:



Chapter 3: Command system

3.1 Command list

RD-FH5C7 Series thermal printers use the ESC / POS compatible command, and add tosome functions such as the Chinese characters printing, Character and Chinese characters rotation, and word spacing adjustment.

Command	Function
ESC c	To allow/ban reverse printing
HT	To execute horizontal tab
LF	To print and line feed
CR	To print and carriage return
ESC SP	To set the character spacing
ESC !	To select printing modes for characters
ESC $	To set printing absolute position
ESC ‘	To print curve
ESC i	To cut the paper (only for printers with cutter)
ESC v	Send the printer’s status to the host
FS &	Select the Kanji mode
FS .	Cancel the Kanji mode
ESC +	To allow/ban the overline printing

ESC –	To allow/ban the underline printing (to set/clear the underline mode)
ESC 1	To set the line spacing
ESC 6	To select Character Set 1 （6X8）
ESC 7	To select Character Set 2 （6X8）
ESC a	Select alignment methods
GS v	Print raster bit image
ESC @	To initialize the printer
ESC D	To set the position of horizontal tab
GS k	To print the bar code
ESC J	To print and feed paper
ESC d	To print and feed paper n lines
ESC K	Printing graphics command ①
ESC *	Printing graphics command ②
ESC Q	To set the right margin width
ESC U	Horizontally magnify character
ESC V	Vertically magnify character
GS H	Select the printing position for bar code character
GS F	Select mode
GS Q	Set horizontal printing position for the bar code
GS k	Print the bar code(s)
ESC X	Magnify characters

GS h	Select bar code's height
GS w	Select bar code's width
GS B	To allow/ban white reverse printing mode
ESC I	To set the printing position
ESC r d n	To adjust the depth of thermal printing
FS 2	To set character rotation Printing
FS r	To select the superscript and subscript

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

【COMMAND】+【*parameter*】

1) 【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.

2) 【*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 'SendDataToPrinter' function is virtual function. And require developers to write according to the actual situation of the mainframe.

This function is defined as follows:

SendDataToPrinter(unsigned char *buffer, unsigned int len)
Illustration: send the data to the printer

Unsigned char *buf // Print data command

Unsigned int len // Data length. Unit: byte.

3.2 Command Details

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;
SendDataToPrinter(str,2);
```

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'
SendDataToPrinter(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* x 0.125mm(0.0049")].

[Comment]

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

[Scope] 0 ≤ *n* ≤ 255

[Example] unsigned char str[3];

```
str[0] = 0x1B;
str[1] = 0x4A;
str[2] = 0x4;
SendDataToPrinter(str,3);// feeds forward paper 0.5mm
```

ESC d n

[Name] print and feed paper n lines

[Type] ASCII: ESC d n

Decimal: 27 100 n

Hex: 1B 64 n

[Scope] 0 ≤n ≤ 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;
SendDataToPrinter(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 ≤ *n* ≤ 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
SendDataToPrinter(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 ! n

[Name] To set printing mode(s) for characters

[Type] ASCII： ESC ! *n*

Decimal: 27 33 n

Hex: 1B 21 n

[Scope] 0 ≤ n ≤ 255

[Explanation]Set the printing mode(s) according to the value of n

bit	0/1	function
0	0	English characters (half-width) font A（12×24）
	1	English characters（half-width）font B (8×16)
1，2,3	——	--
4	0	Cancel the double-height mode
	1	Select the double-height mode
5	0	Cancel the double-width mode
	1	Select the double-width mode
6	——	--
7	0	Cancel the underline mode
	1	Select the underline mode

[Example] unsigned char str[3];

```
str[0] = 0x1B;
str[1] = 0x21;
str[2] = 0x31;
```

SendDataToPrinter(str,3);// print the character (8X16) under the double-width and double-height mode

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] $1 \leqslant n \leqslant 255$ $0 \leqslant k \leqslant 20$

[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).

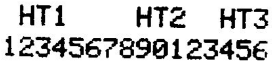
[Example] unsigned str[8];

```
        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
SendDataToPrinter (str,6)
SendDataToPrinter (&Order,1);
SendDataToPrinter (“HT1”,3);
SendDataToPrinter (&Order,1);
SendDataToPrinter (“HT2”,3);
SendDataToPrinter (&Order,1);
SendDataToPrinter (“HT3”,3);
Order = 0x0D;
SendDataToPrinter (&Order,1);
SendDataToPrinter (“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;
```

```
        SendDataToPrinter (str,3);// set the underline
```

ESC +

[Name] allow/ban the overline printing

[Type] ASCII: ESC + n

Decimal: 27 43 n

Hex: 1B 2B n

[Explanation]When n=1, allow the overline printing

When n=0, ban the overline printing

[Note]

- Overline 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] = 0x2B;
 str[2] = 0x1;
SendDataToPrinter (str,3);//    set the 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 ≤n ≤ 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
SendDataToPrinter(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 ≤n ≤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 ≤ *nL + (nH x 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×256

[Comment]

- Settings outside the specified printable area are ignored.
- In mode 1, n <= 372; In mode 2, n <= 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 ≤n ≤ 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;//
SendDataToPrinter (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 ≤n ≤ 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;//
```

SendDataToPrinter (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 ≤ n ≤ 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;
```

SendDataToPrinter(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 ≤ n ≤ 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;
```

SendDataToPrinter(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 ≤ n ≤ 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: 28114n

Hex: 1C 72 n

[Scope] 0 ≤ n ≤ 1

[Explanation]

The value of **n**	Result
n=o	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;
SendDataToPrinter(str,3);//
```

ESC U

[Name] Horizontally magnify characters

[Type] ASCII： ESC U n

Decimal: 27 85 n

Hex: 1B 55 n

[Scope] 0 ≤ n ≤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

[Name]Vertically magnify characters

[Type] ASCII： ESC V n

Decimal: 27 86 n

Hex: 1B   56 n

[Scope] 0 ≤ n ≤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;
SendDataToPrinter(str,3);// Vertically magnify 2 times
```

ESC X

[Name] Magnify characters

[Type] ASCII: ESC   X n1 n2

Decimal: 27   88 n1 n2

Hex: 1B   58 n1 n2

[Scope] 0 ≤ n ≤8 (1 ≤ *n1* horizontal times ≤ 8,1 ≤ *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
```

ESC i

[Name] Full cutting

[Type] ASCII： ESC i

Decimal: 27 105

Hex: 1B 69

[Explanation] cutter: full 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] = 0x69;
SendDataToPrinter(str,2);// Send the 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
```

ESC K nL nH   d1 d2 ……dk

[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 ≤nL ≤255

0 ≤nH ≤1
0 ≤d ≤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;
SendDataToPrinter(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.

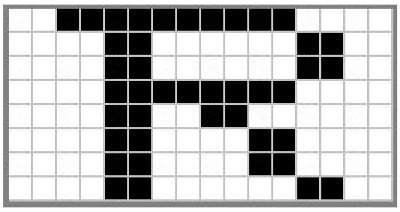
Mode 1: When the double-width mode is single-width, its maximum is 384. When the double-width mode is twice as width, its maximum is 192.

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 2kth and (2k-1)th column. The bn represents the nth 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

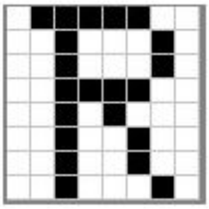


Program code is as follows:

```
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;//print the graphic
SendDataToPrinter(str,j);
```

[Example 2] m =1 (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 kth column. The bn represents the nth 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, don't enlarge)
str [j++] = 8; //the graphic width is 8dots
str [j++] = 0;//bit image data
str[j++] = 0x00;str[j++] = 0x80;str [j++] = 0xFF;str[j++] = 0x90;str[j++] = 0x98;
strr[j++] = 0x96;str[j++] = 0x61;str[j++] = 0x00;str[j++] = 0x0D; ;//print the graphic
SendDataToPrinter(str,j);
```

[Example 3] 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**								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**	**0**	b4
	0	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**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
	0	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	b0
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
	0	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	b0
d3	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	b7
	0	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**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**	**0**	b0

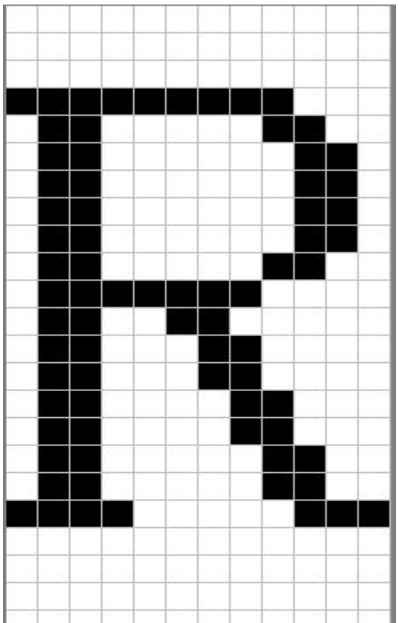
Program code is as follows:

```
unsigned char str[200];
j=0;
str[j++] = 0x1B;
str[j++] = 0x2A;
str[j++] = 32; //m=32(height is 24 dots, double-width)
str[j++] = 12; //graphic width is 12dots
str[j++] = 0;//bit image 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 the current graphics
SendDataToPrinter(str,j);
```

[Example 4] m =33 (24 dots, don't enlarge), 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.

d1

	d4	d7									D	d49
0	0	0	0	0	0	0	0	0	0	0	0	b7
0	0	0	0	0	0	0	0	0	0	0	0	b6
0	0	0	0	0	0	0	0	0	0	0	0	b5
1	1	1	1	1	1	1	1	1	0	0	0	b4
0	1	1	0	0	0	0	0	1	1	0	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
0	1	1	0	0	0	0	0	0	1	1	0	b0
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



d2	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
	0	1	1	0	0	0	0	1	1	0	0	0	b0
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	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 24 dots, don't enlarge)
    str[j++] = 12; // graphic width is 12dots
    str[j++] = 0;
    // bit image 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 the current graphics
    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

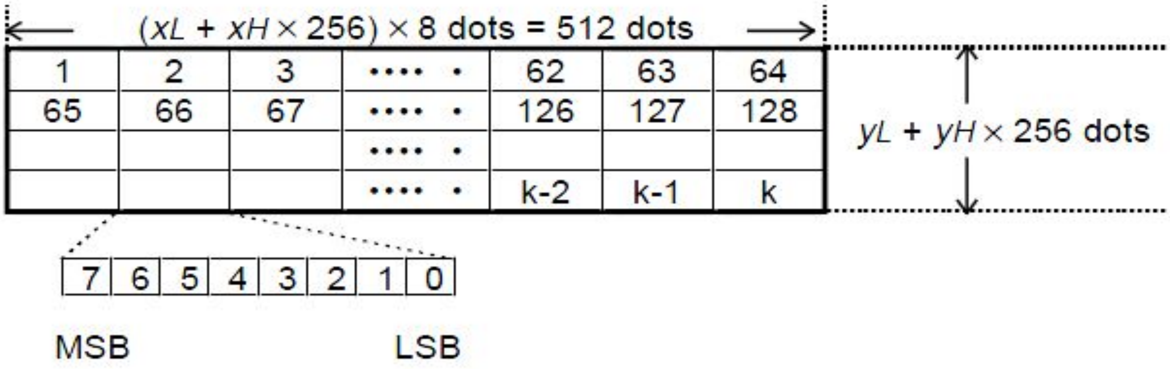
[Scope] 0 ≤m ≤ 3, 48 ≤m ≤ 51

0 ≤xL ≤255
0 ≤ xH ≤255 where 1 ≤ (xL + xH × 256) ≤128
0 ≤yL ≤255
0 ≤yH ≤8 where 1 ≤ (yL + yH × 256) ≤ 4095
0 ≤d ≤255
k = (xL + xH × 256)×(yL + yH × 256) (k≠0)

m	mode	Vertical resolution（DPI）	Lateral resolution（DPI）
0，48	Normal	203.2dpi	203.2dpi
1，49	Double-width	203.2dpi	101.6dpi
2，50	Double-height	101.6dpi	203.2dpi
3，51	Double-width、Double-height	101.6dpi	101.6dpi

• 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



GS h n

[Name] Select bar code height

[Type] ASCII: GS h   n

Decimal: 29 104 n

Hex: 1D 68   n

[Scope] 1 ≤ n ≤ 255

[Explanation] Select bar code height. And N is the number of dots on the vertical direction.

[Default] n=48

[Example] 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 pitch
```

GS w n

[Name] Select bar code width

[Type] ASCII: GS w n

Decimal: 29 119 n

Hex: 1D 77 n

[Scope] 1≤ n ≤ 4

[Explanation] 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 the bar code width
```

GS H n

[Name] Select the printing position for bar code character

[Type] ASCII: GS H n

Decimal: 29 72 n

Hex: 1D 48 n

[Scope] 0 ≤ n ≤ 2

[Explanation]Selects a font for the HRI characters used when printing a bar code.

Use n to specify the printing position of HRI:

n	printing position
0	Do not print
1	Above the bar code
2	Below the bar code

[Default] n=0

[Example] unsigned char str[4];

```
 str[0] = 0x1D;
 str[1] = 0x48;
 str[2] = 2;
SendDataToPrinter(str,3);// The HRI is printed below the bar code
```

GS Q n

[Name] Set the printing position of the bar code on the horizontal direction

[Type] ASCII: GS Q *n*

Decimal: 29 81 *n*

Hex: 1D 51 *n*

[Scope] 0≤ n ≤ 255

[Explanation]Set the distance from the beginning of one line to the position of printing bar code to N horizontal dot pitch.

[Default] n=0

[Example] unsigned char str[4];

```
 str[0] = 0x1D;
 str[1] = 0x51;
 str[2] = 32;
SendDataToPrinter(str,3);//
```

GS k

[Name] Print bar code

[Type] ASCII: ① GS k m d1...dk NUL ② GS k m n d1... dn

Decimal: 29 107 m d1...dk 0 29 107 m n d1... dn

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

[Scope] ① 0 ⩽ m ⩽ 6 ② 65 ⩽ m ⩽ 73

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.

[Explanation] Select a bar code system and print the bar code.

Use m to select a bar code system as follows:

m		Bar code system	length	scope
Format 1	0	UPC-A	11 k 12	48 d 57
	1	UPC-E	K	48 d 57
	2	JAN13 (EAN13)	12 k 13	48 d 57
	3	JAN 8 (EAN8)	7 k 8	48 d 57
	4	CODE39	1 k	48 d 57, 65 d 90, 32, 36, 37, 43, 45, 46, 47
	5	ITF	1 k (even number)	48 d 57
	6	CODABAR	1 k	48 d 57, 65 d 68 , 36, 43, 45, 46, 47, 58
Format 2	65	UPC-A	11 n 12	48 d 57
	66	UPC-E	n=8	48 d 57
	67	JAN13 (EAN13)	12 n 13	48 d 57
	68	JAN 8 (EAN8)	7 n 8	48 d 57
	69	CODE39	1 n 255	48 d 57, 65 d 90, 32, 36, 37, 43, 45, 46, 47
	70	ITF	1 n 255 (even number)	48 d 57
	71	CODABAR	1 n 255	48 d 57, 65 d 68, 36, 43, 45, 46, 47, 58
	72	CODE93	1 n 255	0 d 127
	73	CODE128	2 n 255	0 d 127

[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.

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

ESC ‘

[Name] Print curve

[Type] ASCII： ESC ‘ nL nH x1L x1H x21L x21H ……. xkL xkH CR

Decimal: 29 39 nL nH x1L x1H x21L x21H ……. xkL xkH 13

Hex: 1B 27 nL nH x1L x1H x21L x21H ……. xkL xkH 0D

[Scope] 0 ≤ nL ≤255

0 ≤ nHL ≤1

N is the number of the curve’s dots and N = nH x 256 + nL

The position of the curve’s dots in one horizontal line: X = xkH x 256 + xkL

[Explanation] Each curve consists of many dots. The command indicates that the printer prints n dots in one horizontal line, and continuously using the command can print out the curve which the user needs.

[Note] This command is only applicable to impact dot matrix printer and some thermal models.

[Example]

Print curve graphic of below five equations:

Y1=50+40*abs（-0.01*X）*sin（X/10）

Y2=50-40*abs（-0.01*X）*sin（X/10）

Y3=50

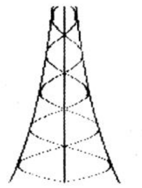
Y4=50+40*abs(-0.1*X)

Y5=50-40*abs(-0.01*X)

C program is as follows:

```
unsigned char str[50];
float X;
unsigned int    m_cur1,m_cur2,i;
for(X=0;X<150;X++)                                    //print one line of 150 dots
{
      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;
      SendDataToPrinter(str,i);//
}
```

[Print results]



ESC v

[Name] Send the printer’s status to the host

[Type] ASCII： ESCv

Decimal: 27 118

Hex: 1B 76

[Explanation] Send the printer’s status to the host

[Note] It is only effective for the printer with serial interface

Bit	Function	Value	
		0	1
0	Paper detector	No paper	Have paper
1	Work status	Idle	Printing
2	Receiving buffer	No full	Full
3	printer’s status	Normal	Error
4	Undefined	——	——
5	Undefined	——	——
6	Undefined	——	——
7	Undefined	——	——

[Example]

```
unsigned char str[4];
 str[0] = 0x1B;
 str[1] = 0x76;
SendDataToPrinter(str,2);//Send status query command to the print
```

FS &

[Name] Select the Kanji mode

[Type] ASCII： FS &

Decimal: 28 38

Hex: 1C 26

[Explanation] The printer enters Kanji printing mode

[Note] After powering on the printer, the printer defaults the Kanji printing mode

[Example]

```
unsigned char str[4];
 str[0] = 0x1C;
 str[1] = 0x26;
SendDataToPrinter(str,2);// Enter Kanji printing mode
```

FS.

[Name] Cancel the Kanji mode

[Type] ASCII： FS .

Decimal: 28 46

Hex: 1C 2E

[Explanation] Cancel the Kanji characters mode

[Example] unsigned char str[4];

```
 str[0] = 0x1C;
 str[1] = 0x2E;
SendDataToPrinter(str,2);//Enter ASCII characters printing mode
```

ESC 6

[Name] To select Character Set 1 （6X8）

[Type] ASCII： ESC6

Decimal: 27 54

Hex: 1B 36

[Explanation] After inputting the command, all of printing characters use the characters in the character set 1 (see appendix D). The character set 1 has 224 ‘6 x 8 dot matrix’ characters, including ASCII characters and all kinds of graphic marks, etc. The range of code is 20H~FFH(32~255).

[Example] unsigned char str[4];

```
 str[0] = 0x1B;
 str[1] = 0x36;
SendDataToPrinter(str,2);//Print ‘6X8’ characters in the Character Set 1
```

ESC 7

[Name] To select Character Set 2（6X8）

[Type] ASCII： ESC 7

Decimal: 27 55

Hex: 1B 37

[Explanation] After inputting the command, all of printing characters use the characters in the character set 2 (see appendix D). The character set 2 has 224 '6 x 8 dot matrix' characters, Including German, French, Russian, Japanese Katakana, etc. The range of code is 20H~FFH(32~255).

[Example] unsigned char str[4];

```
 str[0] = 0x1B;
 str[1] = 0x37;
SendDataToPrinter(str,2);// Print '6X8' characters in the Character Set 2
```

GS F n

[Name] Select mode

[Type] ASCII: GS F *n*

Decimal: 29 70 *n*

Hex: 1D 46 *n*

[Scope] 0 ≤ *n* ≤ 1

[Explanation]

When n=0, the number of the dots printed in each line is 384.

When n=1, the number of the dots printed in each line is 432.

[Default] n=0

[Example]unsigned char str[4];

```
 str[0] = 0x1D;
 str[1] = 0x46;
 str[2] = 0;
SendDataToPrinter(str,3);//
```

ESC r d n

[Name] To adjust the depth of thermal printing

[Type] ASCII: ESC r d *n*

Decimal: 27 114 d *n*

Hex: 1B   72   d *n*

[Scope] d=2B or 2D, 0 ≤ *n* ≤ 255

[Explanation]

When d=2B, indicates the printing depth is adjusted deeper on the basis of the current printing depth.

When d=2D, indicates the printing depth is adjusted more shallowly on the basis of the current printing depth.

The n represents the value of the depth adjusting degree. And the larger the value, deeper the adjustment depth, and the maximum value does not exceed 255.

When n=0, using 'B 72 2B 00' or '1B 72 2D 00' will lead to revert to the default value

[Default] n=0

[Note] When using the deepening adjustment, the value of n is adjusted to the maximum value, or it not only affects the results but also reduces the printing life.

With increasing depth, the printer will increase power consumption.

[Example] unsigned char str[4];

```
str[0] = 0x1B;
str[1] = 0x72;
str[2] = 0x2B;
str[3]= 0x30;
SendDataToPrinter(str,4);// deepen '0X30'
```

Chapter 4: Maintenance and Troubleshooting

To ensure the printer to work normally, particularly note that we don't optionally remove the print head and do not make changes to the printer through ourself. For users not using the printer shell, more particularly note protecting the printing head.

1. If the printer is not used for a long time, we do not turn on the printer power.
2. If the printer is not working properly, please turn off the printer's power.
3. Power supply must meet the requirements, or it is unfavorable for the printing head, and even damages the printing head.
4. When replacing the paper roll, please note whether there are the paper scraps and dust on the printing head. If having paper scraps and dust, please gently remove. Note the thermal paper's obverse and reverse side, and if the reverse side is uncoated, the printer can‘t print out the handwriting.
5. When the printer is printing or paper feeding, we can't tear the paper, and can't more reversely drag the paper
6. Keep the printer control panel clean
7. When thermal printer prints unclearly, we can use the clean cotton ball soaked some alcohol to gently wipe the surface dirt on the print head chip heating element.
8. When we connect the printer to the host, we should connect the printer data cable, and then power on the printer.
9. To choose a good quality paper when we select the paper for the thermal printer can not only improve the printing quality, but also reduce the abrasion for thermal film.

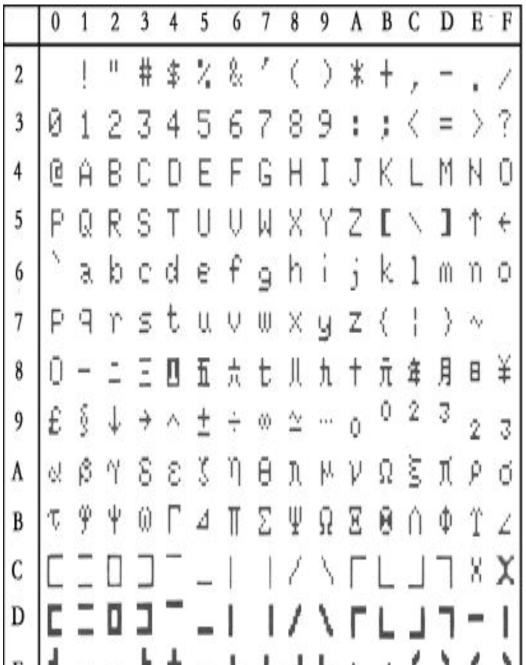
APPENDIX

A : printing character set

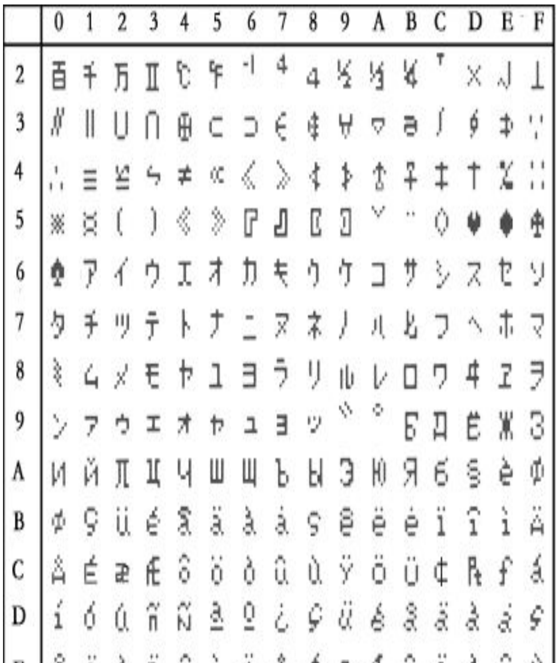
HEX		HEX		HEX		HEX		HEX		HEX		HEX		HEX	

20	(空格)	21	!	22	“	23	#	24	$	25	%	26	&	27	‘
28	(	29	)	2A	*	2B	+	2C	,	2D	-	2E	.	2F	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3A	:	3B	;	3C	<	3D	=	3E	>	3F	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4A	J	4B	K	4C	L	4D	M	4E	N	4F	O
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5A	Z	5B	[	5C	\	5D	]	5E	^	5F	_
60	`	61	a	62	b	63	c	64	c	65	e	66	f	67	g
68	h	69	i	6A	j	6B	k	6C	l	6D	m	6E	n	6F	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7A	z	7B	{	7C	|	7D	}	7E	~	7F	
80	Ç	81	ü	82	é	83	â	84	ä	85	à!	86	å	87	ç
88	ê	89	ë	8A	è	8B	ï	8C	î	8D	ì	8E	Ä	8F	Å
90	É	91	æ	92	Æ	93	ô	94	ö	95	ò	96	û	97	ù
98	ÿ	99	Ö	9A	Ü	9B	¢	9C	£	9D	¥	9E	Pts	9F	ƒ
A0	á	A1	í	A2	ó	A3	ú	A4	ñ	A5	Ñ	A6	ª	A7	º
A8	¿	A9	⌐	AA	¬	AB	½	AC	¼	AD	¡	AE	«	AF	»
B0	░	B1	▒	B2	▓	B3	│	B4	┤	B5	╡	B6	╢	B7	╖
B8	╕	B9	╣	BA	║	BB	╗	BC	╝	BD	╜	BE	╛	BF	┐
C0	└	C1	┴	C2	┬	C3	├	C4	─	C5	┼	C6	╞	C7	╟
C8	╚	C9	╔	CA	╩	CB	╦	CC	╠	CD	═	CE	╬	CF	╧
D0	╨	D1	╤	D2	╥	D3	╙	D4	╘	D5	╒	D6	╓	D7	╫
D8	╪	D9	┘	DA	┌	DB	█	DC	▄	DD	▌	DE	▐	DF	▀
E0	α	E1	β	E2	Γ	E3	Π	E4	Σ	E5	σ	E6	µ	E7	γ
E8	Φ	E9	θ	EA	Ω	EB	δ	EC	∞	ED	φ	EE	Є	EF	∩
F0	≡	F1	±	F2	≥	F3	≤	F4	⌠	F5	⌡	F6	÷	F7	≈
F8	°	F9	•	FA	·	FB	√	FC	ⁿ	FD	²	FE	■	FF	

B: character set 1 and 2

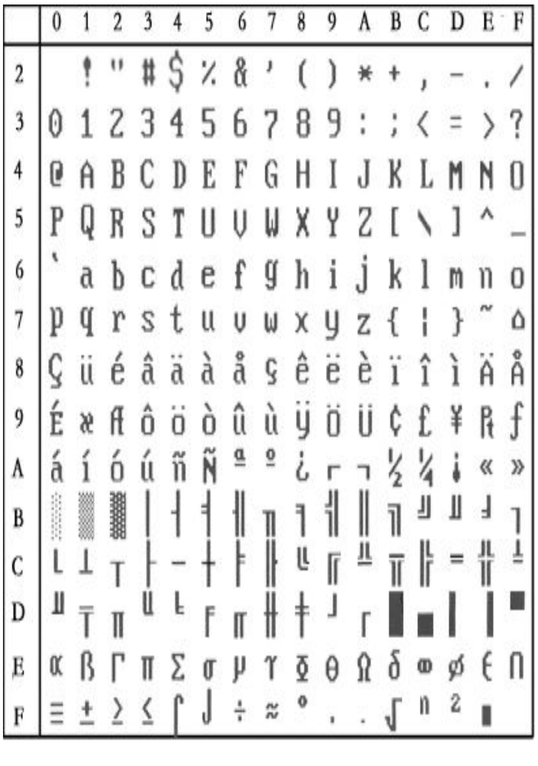


0 1 2 3 4 5 6 7 8 9 A B C D E F
2 3 4 5 6 7 8 9 A B C D



0 1 2 3 4 5 6 7 8 9 A B C D E F
2 3 4 5 6 7 8 9 A B C D

C: International standard ASCII



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	⌂
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

D: bar code

D.1 bar code coding rules

1. UPC-A: UPC-A coding should comply with the UCC Organization (http://www.uccnet.org) specification.

2. UPC-E: UPC-E coding should comply with the UCC Organization (http://www.uccnet.org) specification.

3. ENA8: ENA8 coding should comply with the ENA Organization (http://www.uccnet.org) specification.

4. ENA13: ENA13 coding should comply with the ENA Organization (http://www.uccnet.org) specification

5. CODE39: Also known as 39 codes, CODE39's starting bit and stopping bit characters must be the '*'character, and it cannot contain the characters '*' between the starting and stopping bits. And the printer's '*' is automatically given

by the printer, and when programming the data need not be given, and the data can contain or cannot contain check code, and the check code have fixed algorithm.

6. ITF: Also known as INTERLEAVED 25, intersect 25 code, INTERLEAVED 2 of 5. The length for the data bit must be even number, and the data can contain or cannot contain check code, and the check code have fixed algorithm.

7. CODABAR: The starting bit and stopping bit must be anyone of A、B、C and D, and The starting bit character and stopping bit character can be different. The data can contain or cannot contain check code. The check code can be defined by the code man.

8. CODE93: CODE93's starting bit and stopping bit characters must be the '*'character, and it cannot contain the characters '*' between the starting and stopping bits. And the printer's '*' is automatically given by the printer, and when programming the data need not be given, and the CODE93 data must contain two characters' check code, and the check code have fixed algorithm.

D.2 barcode length character set

Barcode Type	Length	character set (ASCII)
UPC-A	12	0~9
UPC-E	8	0~9
EAN8	8	0~9
EAN13	13	0~9
CODE39	27	0~9 A~Z - . SP $ / + % *
INTERLEAVED 25	even number 52	0~9
CODABAR	32	0~9 - : / % . A~D
CODE93	Unlimited	0~9 A~Z - . SP $ / + % *
CODE128	33	