RD-V2

Portable Thermal Printer
Development Manual

Table of Contents

一、Overview

RD-V2 Series Portable Printer Is A Specially Designed Thermal Micro Printer For Mobile Receipts. With Built-In Rechargeable Lithium Battery, The Printer Can Load 40mm Diameter Paper Roll, It Is With Character Of Compact Size, Long Stand-by Time, Fast Printing Speed, Clear, etc.

The Printer Adopts Clamshell Paper Loading Way, Supports Auto Sleep, Auto Wake-Up Functions Meanwhile Possessing The Ability Of Voice And Light Alert.

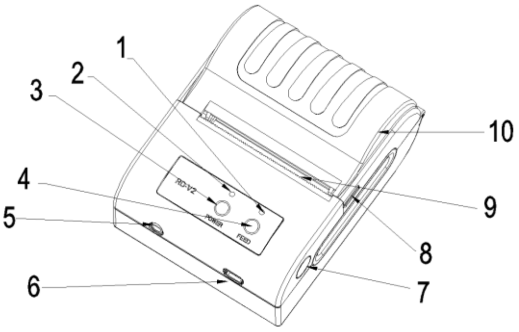
Technical Data：

Print Functions	Print Type	Thermal Line Printing
	Print Speed	60mm/Sec（Full Power Charging）
	Resolution	203dpi（8Dot/mm)，384dot/Line
	Effective Print Width	48mm
	Paper Feeding Space	0.125mm
	Western Character	ANK、ASCII 12x24,8X16,8X12、International Character(12x24)
	Chinese Character	Standard 24×24Dot Matrix GB18030Character Set
	Barcode Print	1D UPC-A、UPC-E、EAN-13、EAN-8、CODE39、ITF25、CODABAR、CODE93、CODE128、2D QRCODE、PDF417,Etc Various Barcode Printing。
	Graphic Print	Support
	Curve Print	Support
	Dynamical Table	Support
	Bitmap Download	Support
Testing Functions	Anomaly Detection	Out Of Paper Detection、Power Insufficient Detection、Overheating Protection By Buzz Warning.
	Black Line Location	Selectable
	Auto Sleep	Yes
Interface	Wired Interface	Serial Interface:5PIN MiNi USB（Standard RS232 Or TTL），485Interface: 5PIN MiNi USB
	Wireless Interface	Bluetooth Interface
Control System	Buffer	2K/64K
	Command System	ESC/POS，Compatible To IBM/EPSON ESC/P
	Print Driver	WIN2000/NT/XP/WIN7
Power Supply Parma	Way Of Supply	2000mAh、7.4V Rechargeable Lithium Battery，By 12.5%Print Density， 100m～200m Can be Printed Consecutively With Full Charging
	Way Of Charging	Stand by Charging，Charging Time：About 3 Hours，Stand by Time≧24Hour。
Reliability	Print head Life	50km

Paper	Paper Type	Common Thermal Paper，Width 58±0.5mm/OD≤Φ40mm/Thickness 0.06mm～0.07mm。
	Loading	Clamshell、Easy Paper Loading。
	Paper Cutting	Manual Tearing
Physical Property	Operating Temperature Range/Humidity	0～50℃/10～80%RH
	Storage Temperature Range / Humidity	–20～60℃/10～90%RH
	Weight(Not Including Paper)	240g(Include Battery)
	Outlook Size	99mm Length×76mmWidth×45.5mmHeight

二、Printer Status And Operating Instructions

2.1 Outlook &Size



Size(Length× Width× Height):	99mm×76mm×45.5mm
Printer Parts Name	
1. Charging Indicator	2. Function Indicator
3. Power Button(P)	4. Paper Feeding Button(F)
5. Charging Jack	6. Data Interface
7.Extended Port	8. Uncover
9.Paper Tearing Mouth	10 Paper Cover

2.2 Printer Status And Operating Instructions

1. When the printer is turned on, the buzzer will sound 3 beeps to alert the boot;

2. When the charging adapter is connected to the printer, the printer's buzzer will issue a short musical sound;

3. When the printer does not receive data, the printer automatically sleep, and wake up automatically after receiving the data.

4. In the boot state, if the printer does not print data within 10 minutes, the printer automatically shut down.

Specific status is as follows:：

power-up status	battery	
	Indicator	Buzzer
Stand by	The green (blue) indicator on the right flashes	--

Print status	The green (blue) indicator on the right brighten for long	--
Lack of paper status	The red indicator on the right flashes	2 beeps /2s
Lack of electricity	The red indicator on the right flashes	1 beep /2s
Charging	The red indicator on the left brighten for long	--
Charging complete	The green (blue) indicator on the left brighten for long	--

2.3 Booting method

The P key is the power button. Hold down the key and after hearing the tone the printer switched on, and then press this button again, the printer shuts down. When the printer does not print data in the 10 minutes, the printer will automatically shut down (the time can be adjusted according to customer's requirements), when used, the printer must be re-boot to print.

2.4 Loading roll paper

The Printer Is With The Easy Paper Loading Structure

Step 1: Open the paper storehouse door

Step 2: Directly put the thermal roll paper into the paper store house in the proper direction and the smooth side down

Step 3: Place the paper to the extent that it can be exposed from the printer and close the paper storehouse cover and press the paper's exposed end.

The F key is the FEED button. In the boot state, hold down the F key, printer starts feeding paper, and loosen the F key, the printer stops feeding paper.

2.5 Self-test Method

In the shutdown state, hold down the F key, then press the P key for about three seconds, and then loosen the button and the printer starts the self-test printing (print out the model of the printer, communication methods, manufacturers and other information).

2.6 Charging

Whether the printer is in the boot or shutdown state, the printer automatically enters the charging mode as long as charging adapter is inserted.

3、Interface

3.1 Serial interface

Data transfer: Serial

Synchronization way: Asynchronous

Interface level: RS232 level

Baud Rate: 9600

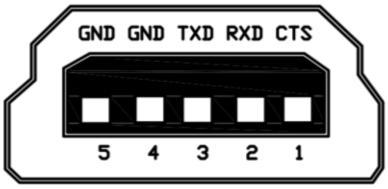
Data Length: 8Bit

Parity: None

Handshake way: CTS

Interface: MINI_USB socket

Socket Is The MINI_USB Interface, Pin Definition Language Is Showed As Followed：



Printer’s data buffer is 8K bytes. When the data which is sent is less than 8K byte, don’t use the ‘flow control’ way, the specific function of the pin is in the following table：

Mini USB Socket **(Pin No.)**	**Signal name**	**signal source**	**direction**	**Illustration**
3	TXD	mainframe	Import(in)	The printer receives the data from the main computer. (TRANSMIT DATA)
2	RXD	printer	Export(out)	When using the ‘X-ON/X-OFF’ Handshake Protocol，the printer sends control code ‘X-ON/X-OFF’ to the computer. (RECEIVE DATA)
1	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.
4	GND	——	——	Signal ground
5	GND	——	——	Signal ground

Programming operation of the printer with RS232 interface is as follows:
Connect the printer interface-- Printer is turned on—Initialize the PC serial port—Send the data
Specific steps:

1) Connect the printer's serial interface with the host’s serial interface, pay attention to the serial port level, should RS232 level.

2) Determine the paper has been installed, press the P key, turn on the printer.

3) Open the PC serial port, communication speed and communication mode is set to the same printer. Normal for 9600,8, N, if they cannot determine the specific parameters of communication by self-detect the printer on self-test strips communication parameters are detailed instructions, click here parameter settings.

4) Send data feedback to the serial Port, For example, if we print RONGDA, The ASCII code of RONGDA will be sent back to the port, whose Hexadecimal digits form is : 52H 4FH 4EH 47H 44H 41H 0D

3.2 Bluetooth Interface

Bluetooth interface of the RD-V2 printer is a radio technology to support for short distance (usually within 10m) communication between devices. Information can be exchanged between many devices such as the mobile phone, PDAs, wireless headsets, laptops, peripherals and other related external equipment. Bluetooth standard is IEEE802.15, and operates in the 2.4GHz frequency band, and the bandwidth is 1Mb / s.

Before using the Bluetooth interface to print, the printer needs to match with the mainframe. And the pairing process is initiated by the mainframe.

Setting method is as follows:

(1) RD Bluetooth printer can be found and searched when the printer is in the boot state. After 10 minutes the printer enters the standby state. Search again, and need to re- open the printer.

(2) When the mainframe is searching for external Bluetooth devices, the device is a Bluetooth printer if finding a ‘RD-V2’ Bluetooth device.

(3) Select the ‘RD-V2’ printer.

(4) Input the password "**0000**"

(5) Complete the pairing.

After completing the pairing, the user can operate the printer according to the port, which is mapped in the mainframe by the current Bluetooth devices.

If using a laptop, SMARTPHONE mobile phones, POCKET PC, PALM and other mainframe having virtual Bluetooth serial port, you can send the printing data to the RD-V80 printer through the virtual Bluetooth serial port. If the host does not have a virtual Bluetooth serial port, the company can provide the host Bluetooth module accessories.

3.3 USB Interface

USB is an external bus standard for the specification of computer and external device connectivity and communications. USB interface support equipment features plug-and-play and hot-swappable.

The RD-V2 printer does not need to install the interface driver. After connecting the printer’s USB port, generate an USB printer device on the "Universal Serial Bus" of the "Device Manager", and generate a "**USB001**" USB port in the system. Then, select the port in the program for printing control. Details see 3.1。

四、Print Command In Detail

RD-V2 Series thermal printers use the ESC / POS compatible command, Increased functions such as the Chinese characters printing, Character and Chinese characters rotation, and word spacing adjustment, Barcode Printing, etc.。

4.1 Command List

Command	**Functions**
ESC @	To initialize the printer
FF	Print and feed paper to Next Page Beginning**(Only Subject Black Mark Testing Machine)**
LF	print and line feed
CR	print and carriage return
ESC J	print and feed paper
ESC d	print and feed paper n lines
ESC c	allow/Forbid Reverse Printing
HT	execute horizontal tab
ESC D	set the position of horizontal tab
ESC -	allow/ban the underline printing (to set/clear the underline
ESC +	allow/ban the over line printing
GS B	allow/ban white reverse printing mode
FS 2	set character rotation Printing
ESC $	set printing absolute position
ESC l	set Printing Position
ESC Q	set the right margin width
ESC 1	set the line spacing
ESC SP	set the character spacing
ESC a	Select alignment methods
FS r	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 h	Select bar code height
GS w	Select bar code’s width
GS H	Select printing position for HRI characters
GS Q	Set bar code absolute print position
GS k	Print bar code
GS k	Print QRCODE 2D Code
ESC ‘	Print Curve
ESC v	Transmit status
FS &	Select Chinese character mode
FS .	Cancel Chinese character mode
ESC 6	Select ANK character 1
ESC 7	Select ANK character 2
FS V	Vertical Tab And Print
GS *	Define The Downloaded Bitmap
GS /	Print The Downloaded Bitmap
FS q	Download Multiple NV Bitmap
FS p	Print The Downloaded NV Bitmap

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 ‘Send Data To Printer’ function is virtual function. And require developers to write according to the actual situation of the mainframe.

This function is defined as follows:

Send Data To Printer(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

4.2 Command In Detail

4.2.1 Control Command

ESC @

[Name] Initialize printer

[Format]

ASCII	ESC	@
Decimal	27	64
Hex	1B	40

[Description] Clears the data in the print buffer and resets the printer parameter.

[Notes] •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 marked paper to print starting position

[Format]

ASCII	FF
Decimal	12
Hex	0C

[Description] Prints the data in the print buffer collectively and returns to standard mode.

[Notes] • Prints the data in the print buffer and feeds marked paper to the print starting position.

•This command is enabled only when the BM sensor is set to be effective using with DIP SW6.

•This command sets the print position to the beginning of the line.

• If BM sensor detection is OFF then Prints the data in the print buffer and feeds the paper 12.5cm.

[Example]

```
unsigned char str[2];
str[0] = 0x0C;
Send Data To Printer (str,1);
```

LF

[Name] Print and line feed

[Format]

ASCII	LF
Decimal	10
Hex	0A

[Description] Prints the data in the print buffer and feeds one line, based on the current line spacing.

[Notes] •This command sets the print position to the beginning of the line.

[Example]

```
unsigned char str[2];
str[0] = 0x0A;//or str[0] = ‘\n’
Send Data To Printer (str,1);
```

CR

[Name] Print and carriage return

[Format]

	ASCII	CR
	Decimal	13
	Hex	0D

[Description] Prints the data in the print buffer and feeds one line, based on the current line spacing.

[Reference] LF

[Example]

```
unsigned char str[2];
str[0] = 0x0D;//or str[0] = ‘\r’
Send Data To Printer (str,1);
```

ESC J

[Name] Print and feed paper

[Format]

	ASCII	ESC	J	*n*
	Decimal	27	74	*n*
	Hex	1B	4A	*n*

[Range] 0 ≤ *n* ≤ 255

[Description] Prints the data in the print buffer and feeds the paper [*n* x 0.125mm(0.0049”)]。

[Notes] •After printing is completed, this command sets the print starting position to the beginning of the line

[Example]

```
unsigned char str[3];
str[0] = 0x1B;
str[1] = 0x4A;
str[2] = 0x4;
Send Data To Printer (str,3);// feeds the paper 0.5mm。
```

ESC d

[Name] Print and feed n lines

[Format]

	ASCII	ESC	d	*n*
	Decimal	27	100	*n*
	Hex	1B	64	*n*

[Range] 0 ≤ *n* ≤ 255

[Description] Prints the data in the print buffer and feeds n lines

[Notes] • The distance of a line is 24 of vertical dots (0.125mm).

• This command sets the print starting position to the beginning of the line.

[Example]

```
unsigned char str[3];
str[0] = 0x1B;
str[1] = 0x64;
str[2] = 0x4;
Send Data To Printer (str,3);// feeds 4 lines。
```

ESC c

[Name] Select/cancel Reverse mode

[Format]	ASCII	ESC	c	*n*
	Decimal	27	99	*n*
	Hex	1B	63	*n*

[Range] 0 ≤ *n* ≤ 1

[Description] *n* = 1 Reverse mode selected, *n* = 0 Reverse mode not selected.

[Notes] • Print direction is from left to right. Usually reverse print is adopted when printers are installed vertically, so as to observe the print result.

• Reverse print not only supports character mode but also supports graphics mode. When print the graphics in reverse direction, pay attention to the print order of graphic units, please see ESC K command.

[Example]

```
unsigned char str[3];
str[0] = 0x1B;
str[1] = 0x63;
str[2] = 0x1;
SendDataToPrinter(str,3);// Reverse mode selected
```

HT

[Name] Horizontal tab

[Format]	ASCII	HT
	Decimal	9
	Hex	09

[Description] Moves the print position to the next horizontal tab position.

[Notes] • This command is ignored unless the next horizontal tab position has been set.

• If the next horizontal tab position exceeds the printing area, the printer sets the printing position to [printing area width + 1].

• Horizontal tab positions are set with **ESC D**.

• If this command is received when the printing position is at [printing area width + 1], the printer executes print buffer-full printing of the current line and horizontal tab processing from the beginning of the next line.

[Reference] ESC D.

ESC D n1 n2 … nk NULL

[Name] Set horizontal tab positions

[Format]	ASCII	ESC	D	*n1…nk*	*NULL*
	Decimal	27	68	*n1…nk*	*0*
	Hex	1B	44	*n1…nk*	*00*

[Range] 1 ≤ *n* ≤ 255 0 ≤ *k* ≤ 20

[Description] Sets horizontal tab positions.

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

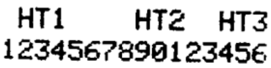
k indicates the total number of horizontal tab positions to be set.

[Notes]

- The horizontal tab position is stored as a value of [character width x n] measured from the beginning of the line. The character width includes the right-side character spacing.
- This command cancels the previous horizontal tab settings.
- When setting n = 8, the print position is moved to column 9 by sending **HT**.
- Up to 32 tab positions (k = 32) can be set. Data exceeding 32 tab positions is processed as normal data.
- Transmit [n]k in ascending order and place a NUL code 0 at the end. When [n]k 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 NUL** cancels all horizontal tab positions.
- The previously specified horizontal tab positions do not change, even if the character width changes.

[Example]

```
unsigned str[8];
unsigned char Order = 9;
str[0] = 0x1B;
str[1] = 0x44;
str[2] = 2;
str[3] = 9;
str[4] = 14;
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] cancel/set underline mode

[Format]

ASCII	ESC	-	*n*
Decimal	27	45	*n*
Hex	1B	2D	*n*

[Description] *n* = 1, Underline mode selected; n=0,Underline mode not selected.

[Notes] • This command is effective for all characters.

[Default] *n* = 0

[Example]

```
unsigned char str[3];
str[0] = 0x1B;
str[1] = 0x2D;
str[2] = 0x1;
SendDataToPrinter (str,3);// Underline mode selected
```

ESC + n

[Name] cancel/set dash mode

[Format]

ASCII	ESC	+	*n*
Decimal	27	43	*n*
Hex	1B	2B	*n*

[Description] *n* = 1, dash mode selected; *n* =0, dash mode not selected.

[Notes] • This command is effective for all characters.

[Default] *n* = 0

[Example]

```
unsigned char str[3];
str[0] = 0x1B;
str[1] = 0x2B;
str[2] = 0x1;
Send Data To Printer (str,3);// dash mode selected
```

GS B n

[Name] cancel/set inverse mode

[Format]

ASCII	GS	B	*n*
Decimal	29	66	*n*
Hex	1D	42	*n*

[Description] *n* = 1, inverse mode selected; *n* =0, inverse mode not selected.

[Notes] • This command is effective for all characters.

[Default] *n* = 0

[Example]

```
unsigned char str[3];
str[0] = 0x1D;
str[1] = 0x42;
str[2] = 1;// inverse mode selected;
Send Data To Printer(str, 3);
```

FS 2 n

[Name] Set Character Rotational mode

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

[Range] 0 ≤ *n* ≤ 3

[Description] Set Character Rotational mode

***n*(Decimal)**	**characters anticlockwise rotated**
0	Turns off anticlockwise rotation mode
1	Turns on 90°anticlockwise rotation mode
2	Turns on 180°anticlockwise rotation mode
3	Turns on 270°anticlockwise rotation mode

[Default] *n* = 0

[Example]

```
unsigned char str[3];
str[0] = 0x1C;
str[1] = 0x49;
str[2] = 1;//
Send Data To Printer(str, 3);
```

ESC $ nL nH

[Name] Set absolute print position

[Format] ASCII ESC $ *nL* *nH*
Decimal 27 36 *nL* *nH*
Hex 1B 24 *nL* *nH*

[Range] 0 ≤ *nL + (nH x 256)* < 384

[Description] Sets 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 print position is [(nL + nH *x* 256) *x*0.125 mm].

[Notes] • Settings outside the specified printable area are ignored.

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x24;
str[2] = 32;//
Send Data To Printer (str, 3); //
```

ESC l n

[Name] Set Left Margin

[Format] ASCII ESC l *n*
Decimal 27 108 *n*
Hex 1B 6C *n*

[Range] 0 ≤ *n* ≤ 32

[Description] Left margin is character number that the left-hand print paper doesn’t print; the width of each character is calculated by 12+ character line spacing.
The value of n should be in the range from 0 to the line width of this model printer.

[Default] *n* = 0, that means no left margin.

[Notes]
- Settings outside the specified printable area are ignored.
- This command sets absolute position, and won’t be influenced by character enlarging commands ESC U and ESC W

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x6C;
str[2] = 3;//
Send Data To Printer (str, 3); //
```

ESC Q n

[Name] Set Right Margin

[Format]	ASCII	ESC	Q	*n*
	Decimal	27	81	*n*
	Hex	1B	51	*n*

[Range] 0 ≤ *n* ≤ 32

[Description] Right margin is character number that the right-hand print paper doesn’t print; the width of each character is calculated by 12 + character line spacing.
The value of n should be in the range from 0 to the line width of this model printer.

[Notes]
- Settings outside the specified printable area are ignored.

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x51;
str[2] = 3;//
Send Data To Printer (str, 3); //
```

ESC 1 n

[Name] Set line spacing

[Format]	ASCII	ESC	1	*n*
	Decimal	27	49	*n*
	Hex	1B	31	*n*

[Range] 0 ≤ *n* ≤ 255

[Description] Sets the line spacing to [n *x* 0.125 mm].

[Default] *n* = 3

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x31;
str[2] = 8;
Send Data To Printer(str,3);
```

ESC SP n

[Name] Set right-side character spacing

[Format]

ASCII	ESC	SP	*n*
Decimal	27	32	*n*
Hex	1B	20	*n*

[Range] 0 ≤ *n* ≤ 255

[Description] Sets the character spacing for the right side of the character to [n *x* 0.125 mm (n *x* 0.0049”)].

[Default] *n* = 0

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x20;
str[2] = 8;
Send Data To Printer(str,3);/。
```

ESC a n

[Name] Select justification

[Format]

ASCII	ESC a		*n*
Decimal	27	97	*n*
Hex	1B	61	*n*

[Range] 0 ≤ *n* ≤ 2

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

n selects the justification as follows:

n	**Justification**
0	Left justification
1	Centering
2	Right justification

[Notes] • The command is enabled only when processed 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 Centering mode
```

FS r n

[Name] Select Up and under Superscript and Subscript

[Format]

ASCII	FS	r	*n*
Decimal	28	114	*n*
Hex	1C	72	*n*

[Range] 0 ≤ *n* ≤ 1

[Description] *n* =0 superscript mode selected.
n =1 subscript mode selected.

[Default] *n* = 1

[Notes] • This command is effective for all characters.

[Example]

```
unsigned char str[3];
str[0] = 0x1C;
str[1] = 0x72;
str[2] = 0;
SendDataToPrinter(str,3);//
```

4.2.2 Zoom Command

ESC U n

[Name] Enlarge Width

[Format] ASCII ESC U *n*
Decimal 27 85 *n*
Hex 1B 55 *n*

[Ranges] 1 ≤ *n* ≤ 8
(1 □. horizontal number of times □. 8)

[Description] The characters are enlarged with horizontal number of times.

[Notes] • This command is effective for all characters and graphics of ESC K .
.•If n is outside the defined range, this command is ignored.

[Default] *n* = 1

[Reference] ESC X

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x55;
str[2] = 2;
SendDataToPrinter(str,3);//
```

ESC V n

[Name] Enlarge Height

[Format] ASCII ESC V *n*
Decimal 27 86 *n*
Hex 1B 56 *n*

[Ranges] 1 ≤ *n* ≤ 8
(1 □. vertical number of times □. 8)

[Description] The characters are enlarged with vertical number of times.

[Notes] • This command is effective for all characters and graphics of ESC K .
.•If n is outside the defined range, this command is ignored.。

[Reference] ESC X

[Example]

```
unsigned char str[4];
```

```
str[0] = 0x1B;
str[1] = 0x56;
str[2] = 2;
SendDataToPrinter(str,3);//
```

ESC X

[Name] Select character size

[Format]	ASCII	ESC X *n1* *n2*
	Decimal	27 88 *n1* *n2*
	Hex	1B 58 *n1* *n2*

[Ranges] 1 ≤ *n* ≤ 8

(1 □. horizontal*n1* of times □. 8,1 □. vertica*n2* of times □. 8)

[Description] The characters are enlarged with vertical and horizontal number of times.

[Notes] • This command is effective for all characters and graphics of ESC K.

.•If n is outside the defined range, this command is ignored.

• The vertical direction is the paper feed direction, and the horizontal direction is perpendicular to the paper feed direction. However,when character orientation changes in 90° or 270° anticlockwise rotation mode, the relationship between vertical and horizontal directions is reversed.

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x58;
str[2] = 2;
str[3] = 2;
SendDataToPrinter(str,4);
```

4.2.3 Graphics Command

ESC K nL nH d1 d2 ……dk

[Name] Printing graphics command ①

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

[Range] 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

[Format] ASCII ESC * m nL nH d1…dk

Decimal: 27 42 m nL nH d1…dk

Hex: 1B 2A m nL nH d1…dk

[Range] 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

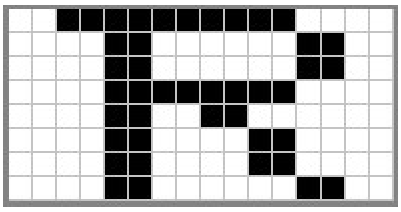
The nL and nH are the low and high bit of double-byte unsigned integer N. They express the number of the dots of the bit-image on the horizontal direction.

Mode 1: When the double-width mode is single-width, its maximum is 576 When the double-width mode is twice as width, its maximum is 288

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

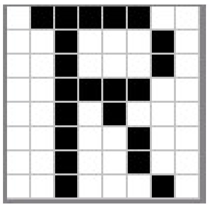
str [j++] = 0x00;str [j++] = 0x80;str [j++] = 0xFF;str [j++] = 0x90;str [j++] = 0x98;

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

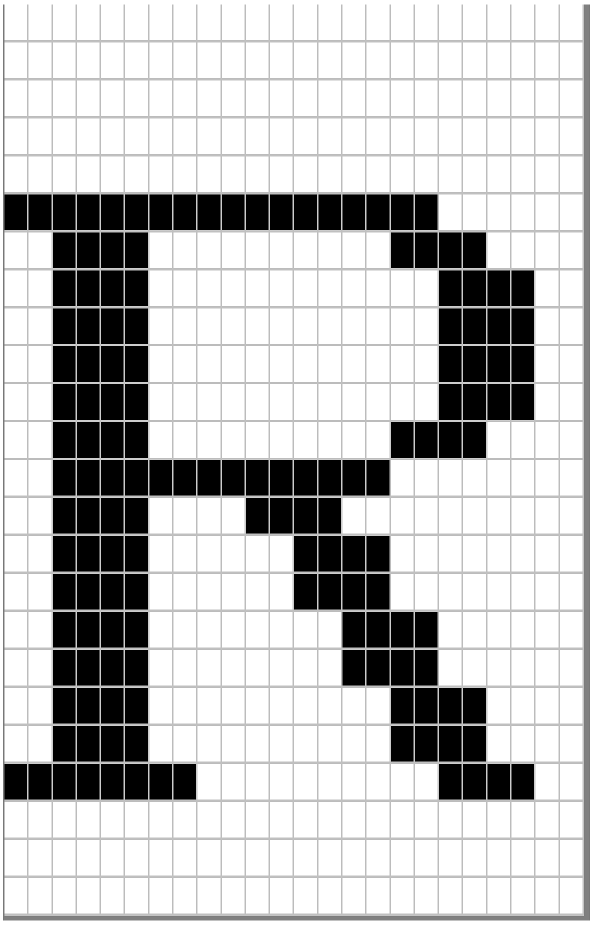
str[j++] = 0x00;str[j++] = 0x80;str [j++] = 0xFF;str[j++] = 0x90;str[j++] = 0x98;

strr[j++] = 0x96;str[j++] = 0x61;str[j++] = 0x00;str[j++] = 0x0D; ;//print the graphic

Send Data To Printer(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										d49
d1	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
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	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

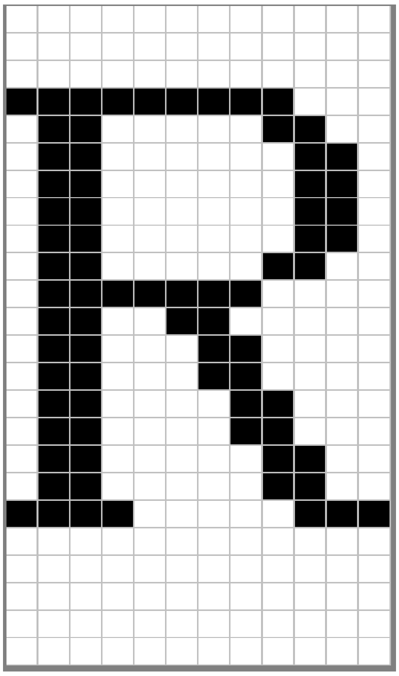


Program code is as follows:

```
unsigned char str[200];
j=0;
str[j++] = 0x1B;
str[j++] = 0x2A;
str[j++] = 32; //m=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.

		d4	d7										d49
d1	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
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
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
    Send Data To Printer(str,j);
```

4.2.5 Barcode Command

EGS h n

[Name] Select bar code height

[Format] ASCII: GS h n

Decimal: 29 104 n

Hex: 1D 68 n

[Range] 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;

Send Data To Printer(str,3);//Set the bar code height to 30 vertical dot pitch

GS w n

[Name] Select bar code width

[Format] ASCII: GS w n

Decimal: 29 119 n

Hex: 1D 77 n

[Range] 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 printing position for HRI characters

[Format] ASCII GS h *n*

Decimal 29 72 *n*

HEX 1D 48 *n*

[Range] 0 ≤ *n* ≤ 2

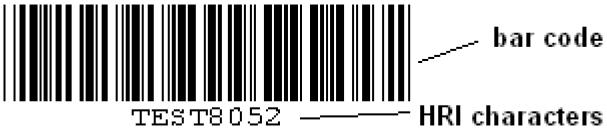
[Description] Selects the printing position of HRI characters when printing a bar code.

n selects the printing position as follows:

n	**Printing position**
0	Not printed
1	Above the bar code
2	Below the bar code

[Notes] • HRI characters are printed using the current font.

[Default]] *n* = 0



[Example]

```
unsigned char str[4];
str[0] = 0x1D;
str[1] = 0x48;
str[2] = 2;
SendDataToPrinter(str,3);//
```

GS Q n

[Name] Set bar code absolute print position

[Format]

ASCII	GS	Q	*n*
Decimal	29	81	*n*
Hex	1D	51	*n*

[Range] 0 ≤ *n* ≤ 255

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

The distance from the beginning of the line to the print position is [n x0.125 mm].

[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

[Format]

Format 1：

(0 ≤ *m* ≤ 8)

ASCII	GS	k	*m*	*d1…dk*	*NUL*
Decimal	29	107	*m*	*d1…dk*	*0*
Hex	1D	6B	*m*	*d1…dk*	*00*

Format 2：

(65 ≤ *m* ≤ 73)

ASCII	GS	k	*m*	*n*	*d1…dn*
Decimal	29	107	*m*	*n*	*d1…dn*
Hex	1D	6B	*m*	*n*	*d1…dn*

[Range] 0 ≤ *m* ≤ 8 (k and d depend on the bar code system used)

65 ≤ *m* ≤ 73 (n and d depend on the bar code system used)

[Description] Selects a bar code system and prints the bar code.

m selects a bar code system as follows:

m		Bar Code System	Number of	Remarks
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

[Notes for 1] □. This command ends with a NUL code.

□. When the bar code system used is UPC-A or UPC-E, the printer prints the bar code data after receiving 12 bytes of bar code data and processes the following data as normal data.

□. When the bar code system used is JAN13 (EAN13), the printer prints the bar code after receiving 13 bytes of bar code data and processes the following data as normal data.

□. When the bar code system used is JAN8 (EAN8), the printer prints the bar code after receiving 8 bytes of bar code data and processes the following data as normal data.

□. The numberof data for the ITF bar code must be even numbers. When an odd number of bytes of data is input, the printer ignores the last received data.

[Notes for 2]

□. n indicates the number of bar code data bytes, and the printer processes n bytes from the next character data as bar code data.

□. If n is outside the specified range, the printer stops

When CODE128 (m = 73) is used:

□. Refer to Appendix F for the information for the CODE128 bar code and its code table.

□. When using CODE128 in this printer, take the folowing points into account for data transmission:

1.The top of the bar code data string must be the code set selection character (CODE A, CODE B, or CODE C), which selects the first code set.

2.Special characters are defined by combining two characters "{" and one character. The ASCII character "{" is defined by transmitting "{" twice consecutively.

ASCII	HEX	**Specific character**
{A	7B, 41	CODE A
{B	7B, 42	CODE B
{C	7B, 43	CODE C
{S	7B, 53	SHIFT
{1	7B, 31	FNC1
{2	7B, 32	FNC2
{3	7B, 33	FNC3
{4	7B, 34	FNC4

[Example]

Example data for printing "No. 123456"

In this example, the printer first prints "No." using CODE B, then prints the following numbers using CODE C.

GS k 73 10 123 66 78 111 46 123 67 12 34 56

GS k m v r d1……dk

[Name] Print QRCODE 2D Barcode

[Format] The Command Include Two Formats：

Format 1　m=32

ASCII GS k *m v r d1…dk NUL*

Decimal 29 107 m v r d1…dk 0

Hex 1D 6B *m v r d1…dk 00*

Format 2　m=97

ASCII GS k *m v r nL nH d1…dn*

Decimal 29 107 m v r nL nH d1…dn

Hex 1D 6B *m v r nL nH d1…dn*

[Range] m=32 Or 97

1 ≤ v ≤ 20 1 ≤ r ≤ 4

[Description] v is DQCODE Version Number

r=1 Error Correction Level L

r=2 Error Correction Level M

r=3 Error Correction Level Q

r=4 Error Correction Level H

nL,nH is Integer N’s Low order and High order, N is barcode data by byte

Format1 Command Ends as00, d1...dk as barcode data.For Formal2,Printer will take N Characters behind nH as barcode data.

[Notice] • As the limit of printer paper, QRCODE max version number is 20.

•ISO/IEC 18004:2000 Ref Detailed Encode Standard of QRCODE,Please see GB/T 18284-2000 or ISO/IEC

[Example]

unsigned char str[16];

str[0] = 0x1D; str[1] = 0x6B; str[2] = 32;

str[3] = 1;//versions1

str[4] = 2;//Error Correction Level 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’;

Send Data To Printer (str, 5);

4.2.6 Curve Command

ESC ‘

[Name] Print Curve

[Format] ASCII GS ‘ 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

[[Range] 0 ≤ nL ≤255

0 ≤ nHL ≤1

dots number N = nH x 256 + nL

dots position X = xkH x 256 + xkL 。

[Description] This command is designed to print curving graphics along with the paper feeding direction. The value of N is the line number of the printing curve. It should be within the range of the maximal dots number of each line of this model printer.

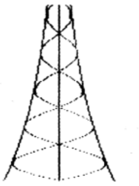
[Example] If you want to print the following 5 equational curving graphics:

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)

[CODE]

```
unsigned char str[50];
float X;
unsigned int  m_cur1,m_cur2,i;
for(X=0;X<150;X++)                              //Print 150 dots
{
    m_cur1= 40*exp(-0.01*X);
    YY= Y*sin(X/10);
    str[i++] = 0x1b;
    str[i++] = 0x27;
    str[i++] = 0x5;//print 5 dots
    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);//。
```

[Result]



4.2.7 Status Transmit

ESC v

[Name] Transmit status

[Format] ASCII ESC v
Decimal 27 118
HEX 1B 76

[Description] Transmits the status of printer.

[Notes] •When using a serial interface。

•This command is executed when the data in the receive buffer is developed. Therefore, there may be a time lag between receiving this command and transmitting the status, depending on the receive buffer status.

Bit	Function	status	
		0	1
0	Paper roll near-end sensor	paper near end	paper adequate
1	Work status	Idle	printing
2	Receive buffer	not full	full
3	Printer status	normal	error
4	NC	——	——
5	NC	——	——
6	NC	——	——
7	NC	——	——

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x76;
Send Data To Printer(str,2);//.
```

4.2.8 Character Set Command

FS &

[Name] Select Chinese character mode

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

[Description] Selects Chinese character mode.

[Example]

```
unsigned char str[4];
str[0] = 0x1C;
str[1] = 0x26;
Send Data To Printer(str,2);//
```

FS.

[Name] Cancel Chinese character mode

[Format]

ASCII	FS	.
Decimal	28	46
HEX	1C	2E

[Description] Cancel Chinese character mode, When the Chinese character mode is not selected, all character codes are processed one byte at a time as ASCII(12X24dots) code.
(see appendix A)

[Example]

```
unsigned char str[4];
str[0] = 0x1C;
str[1] = 0x2E;
Send Data To Printer(str,2);//
```

ESC 6

[Name] Select ANK character 1

[Format]

ASCII	ESC	6
Decimal	27	54
HEX	1B	36

[Description] All characters following this command are printed using the character set I (see appendix D) that are total 224 6×8 dots character, including ASCII character, all kinds of graphic characters and etc. And the code range is 20H~FFH (32~255).

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x36;
Send Data To Printer(str,2);//
```

ESC 7

[Name] Select ANK character 2

[Format]

ASCII	ESC	7
Decimal	27	55
HEX	1B	37

[Description] All characters following this command are printed using the character set I (see appendix D) that are total 224 6×8 dots character, including ASCII character, all kinds of graphic characters and etc. And the code range is 20H~FFH (32~255).

[Example]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x37;
SendDataToPrinter(str,2);//
```

4.2.9 Vertical Tab And Print

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 ≤ m ≤ 17；

LP1...LPm Vertical Line Coordinate：0 ≤ LPm ≤ 48；

n Table Body Numbers：0 ≤ n ≤ 16；

IP1...IPn Table body Coordinate：0 ≤ IPn ≤ 45；

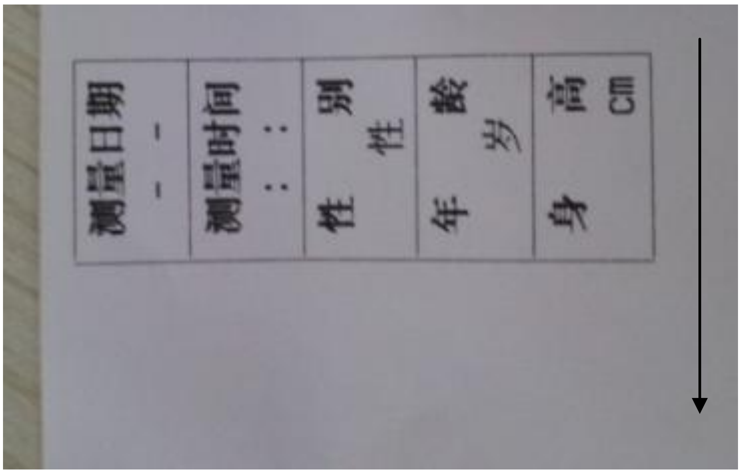
FT1 First Table Body Font Type：

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]:



测量日期 - -

测量时间 : :

性别 性

年龄 岁

身高 cm

4.2.10 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 ≤ *x* ≤ 72 1 ≤ *y* ≤ 20 *x* × *y* ≤ 1024 k=x*y*8

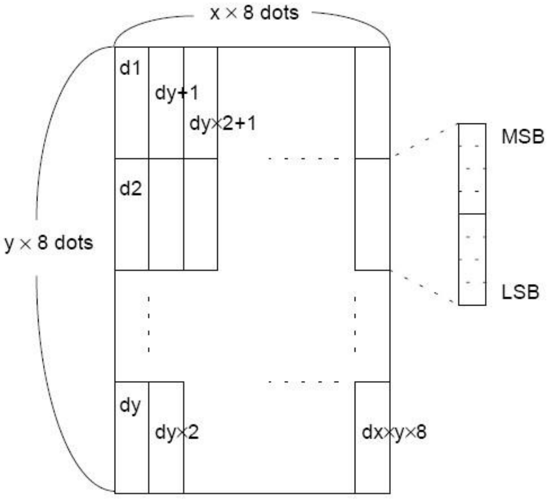
[Explanation] Use x and y appointed count to define the bitmap

• *x*8* is Horizontal Direction Count。 • *y*8* is vertical direction Count。

[Notice] • As the buffer area limit, if *x* × *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 ≤ *n* ≤ 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 □ yL □ 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 (xL xH 256)8 dots

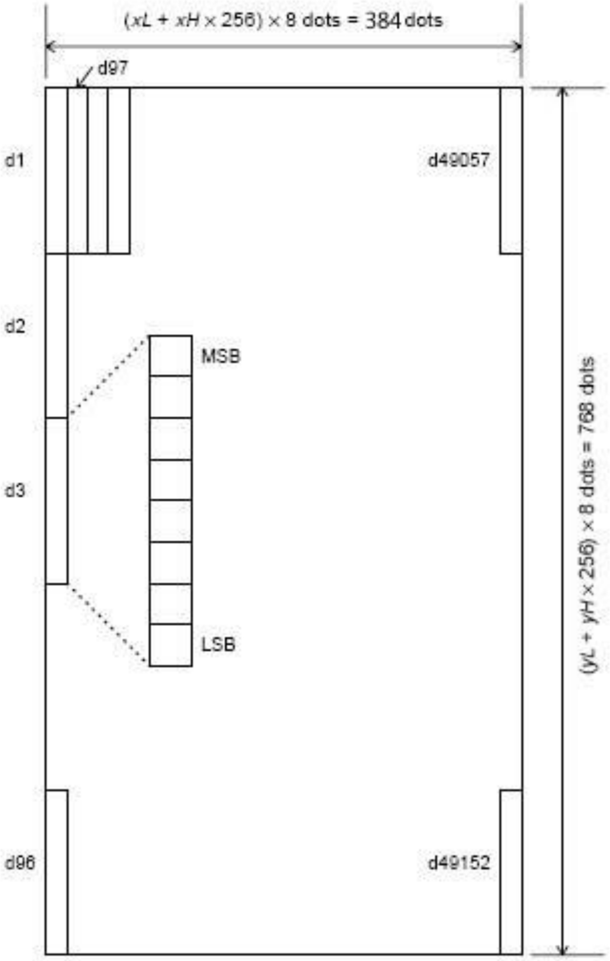
• d is the graphic data.

[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]:

When xL = 64, xH = 0, yL = 96, yH = 0


(xL + xH × 256) × 8 dots = 384 dots
d97
d1
d49057
d2
MSB
d3
LSB
d96
d49152
(yL + yH × 256) × 8 dots = 768 dots

FS p n m

[Name] Print The Downloaded NV bitmap

[Format] ASCII FS p *n* *m*

Decimal 28 112 *n* *m*

Hex 1C 70 *n* *m*

[Range] 1 ≤ *n* ≤ 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

Appendix

A Printer Character Set

The Printer Character Set 0x80 And Following Code Is Printed Out Under The Model Of Cancelling Chinese Character. Related Chinese Character Set, Please See National Standard GB-2312 And Microsoft Code Page CP936.

HE		HE		HE		HE		HE		HE		HE		HE	
20	(空	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	₧	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 Barcode

B.1 Barcode Encoding Standard

UPC-A: Coding confirmed to UCC Standard (http://www.uccnet.org)。

UPC-E: Coding confirmed to UCC Standard (http://www.uccnet.org)。

ENA8: Coding confirmed to EAN Standard (http://www.ean-int.org)。

ENA13: Coding Confirmed to EAN Standard (http://www.ean-int.org)。

CODE39: Also Called 39 Code, Whose Start Bit and End Bit Must Be '*',There mustn't be '*'Between ,The Bit '*'Auto offered by printer, Not from programming, The data bit can or Can't Contain Check Code, Which has fixed algorithm。

ITF: Also Named INTERLEAVED 25 ,Crossed 25 Code, INTERLEAVED 2 of 5,Data Bit Length Only Can Be Even Number, The data bit can or Can't Contain Check Code, Which has fixed algorithm。

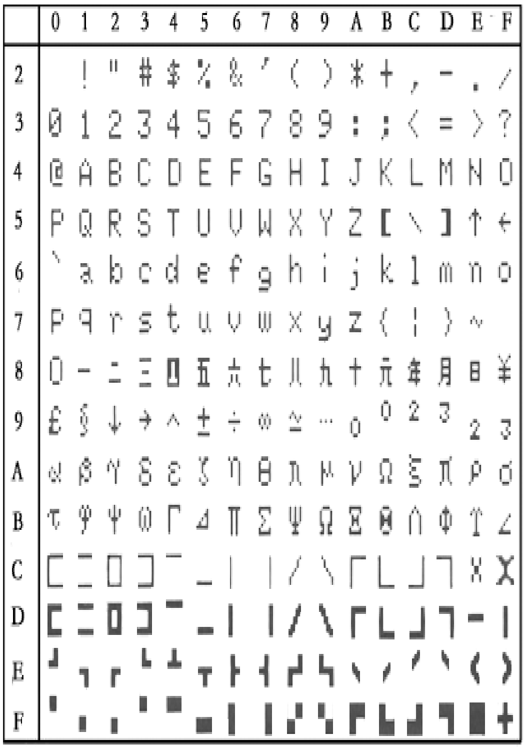
CODABAR: The Start Bit And Stop Bit Must Be One Of A、B、C、D Character, The two bits are not necessary alike, The data bit can or Can't Contain Check Code, Which is self-defined by program man。

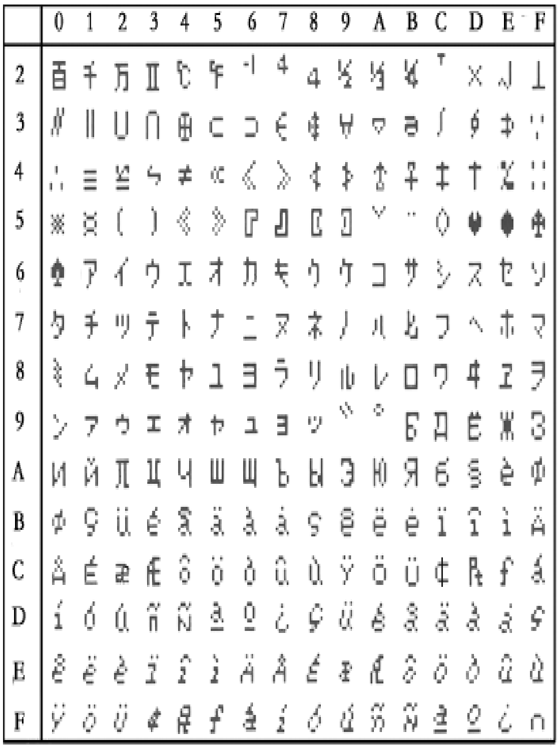
CODE93: CODE93 Start Bit and End Bit Must Be '*',There mustn't be '*'Between ,The Bit '*'Auto offered by printer, Not from programming,The data bit can or Can't Contain Check Code, Which has fixed algorithm

B.2 Barcode Length Character Set

Bar Code	Length	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 52	0~9
CODABAR	32	0~9 - : / % . A~D
CODE93	ALL	0~9 A~Z - . SP $ / + % *
CODE128	33	

C ANK 1、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
E
F


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
E
F