

# RD-FK12

## 系列热敏打印机



版权所有：北京荣达创新科技有限公司

日期	版本	备注
2015-10-24	V1.0	



## 目 录

一、概述 .....	4
1.1 外形及安装尺寸（单位：mm） .....	4
1.2 电源连接器 .....	5
1.3 主要性能指标 .....	5
1.4 操作（控制键为单按键） .....	6
1.4.1 指示灯 .....	6
1.4.2 操作键 .....	6
1.4.3 换 纸 .....	6
二、通讯接口 .....	7
2.1 串行接口 .....	7
2.1.1 接口定义 .....	7
2.1.2 波特率选择位 .....	7
2.1.3 握手方式选择位 .....	7
2.1.4 缺纸控制选择位 .....	8
2.1.5 串口数据发送方法 .....	8
2.2 并行接口 .....	9
2.2.1 数据接口 .....	9
2.2.2 并口数据发送方法 .....	10
三、指令系统 .....	11
3.1 指令表 .....	11
3.2 基本控制指令 .....	13
ESC @ .....	13
LF .....	13
CR .....	13
ESC J .....	13
ESC d .....	14
ESC c .....	14
HT .....	15
ESC ! .....	15
ESC D n1 n2 ... nk NULL .....	16
ESC - n .....	16
ESC + n .....	17
GS B n .....	18
FS 2 n .....	18
ESC \$ nL nH .....	19
ESC l n .....	19
ESC Q n .....	19
ESC 1 n .....	20
ESC SP n .....	20
ESC a n .....	20



FS r n .....	21
ESC U n .....	21
ESC V n .....	21
ESC X .....	22
ESC N .....	22
ESC K nL nH d1 d2 .....dk .....	23
ESC * m nL nH d1...dk .....	23
GS v 0 m xL xH yL yH d1....dk .....	28
GS h n .....	28
GS w n .....	29
GS H n .....	29
GS Q n .....	30
GS k .....	30
ESC ' .....	34
GS ' .....	错误!未定义书签。
ESC v .....	36
FS & .....	36
FS .....	36
ESC 6 .....	38
ESC 7 .....	39
附 录 .....	47
A 打印字符集 .....	47
B 条码 .....	48
B.1 条码编码规则 .....	48
B.2 条码长度字符集表 .....	48
C. 字符集 1、2 .....	48
E. 国际标准 ASCII .....	49



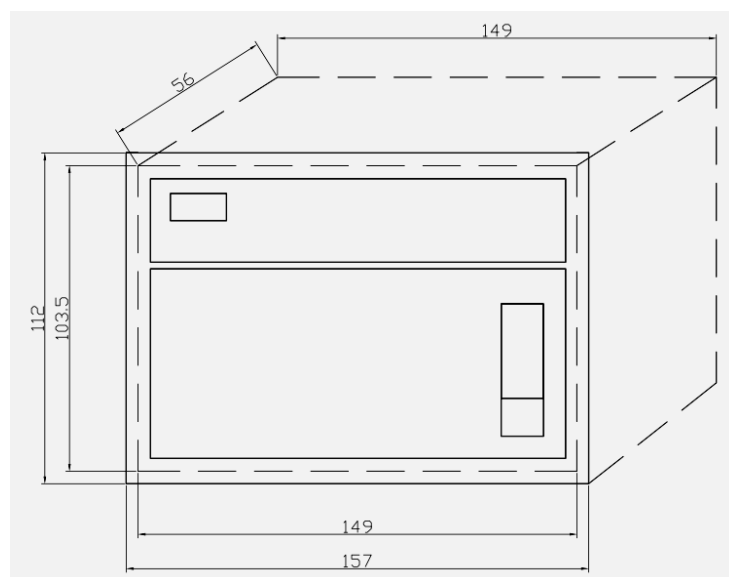
## 一、概述

RD-FK12 系列打印机是打印纸宽为 110mm 的热敏打印机。该系列打印机有两种型号：RD- FK12-P 与 RD-FK12-S。前者为 26 芯标准并口打印机；后者为 5 芯 RS232 串行接口打印机。该系列打印机具有打印纸宽、速度快。

主要特点：

- ◇配置原装打印机芯，热敏打印、无需色带、打印速度快、打印质量高、噪音低。
- ◇黑标定位、缺纸检测、过热检测、等功能。
- ◇产品适用于医疗设备、电力仪器、票据打印等需要宽幅清晰打印的场合。

### 1.1 外形及安装尺寸（单位：mm）

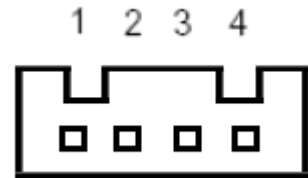




## 1.2 电源连接器

电源接口-----2.54mm 间距垂直，4PIN

引脚	名称	说明
1	GND	地
2	+8.5V	电源正， 8.5V， 电源平均电流为 2.3A， 峰值 5A。
3	+8.5V	
4	GND	地



## 1.3 主要性能指标

打印方式	行式热敏打印	
打印纸宽	110mm	
纸卷直径	小于50mm	
打印行宽	34汉字/行 69英文字符/行（标准24点阵高字体）	
汉字编码方式	GBK	
内置字库	ASCII字库(16x8、24x12、8x6)、GBK(24x24)	
打印速度	50mm/s	
打印机芯寿命	50KM	
可打印内容	英文、数字、各种符号、汉字、图形、曲线、 条码(CODE39、EAN13、EAN8、CODABAR、CODE128、CODE93、ITF)	
分辨率	203 dpi x203 dpi（8点/毫米）	
数据通讯接口	RD-FK12-SN	RS232串口（TTL串口）
	RD-FK12-8+	并口
电源	8.5V/5A	
异常检测	缺纸检测	
切纸方式	无	
指令集	ESC/POS兼容指令集	
使用环境	温度 -10°C~50°C 湿度 20%~85%	



## 1.4 操作（控制键为单按键）

本机为键灯一体设计，即按键与指示为一个。

### 1.4.1 指示灯

当接通电源时，指示灯亮。

当打印机上无纸时，指示灯闪亮。

### 1.4.2 操作键

1. 走纸：按下打印机面板上的按键使打印机连续走纸。

2. 自检：打印机断电，按住前面板按键，通电，约 2 秒钟，打印机进行自检。

### 1.4.3 换 纸

当打印机的指示灯连续闪烁时，说明打印机里的纸已经用完，需重新换纸才可工作。换纸方法为：

a. 打印机面板右侧的扳手扳起，此时纸仓门会自动弹开。

b. 打印机里的纸轴取出，穿到待上的纸卷上。

c. 将待上纸卷打开，拽出约 5CM 长的纸头，将打印机向上放到打印机的纸仓里，卡好纸轴，将纸头露出打印机出纸口。

d. 合上纸仓盖，使仓盖上的胶轮压住露出的纸头。





## 二、通讯接口

### 2.1 串行接口

#### 2.1.1 接口定义

数据传送：串行

同步方式：异步

握手信号：CTS 或者XON/XOFF

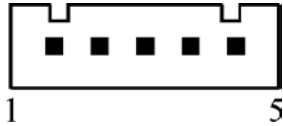
波特率：1200，2400，4800，9600，19200，38400，57600，115200 bps，可通过拨码开关选择变更。

数据长度：8Bit

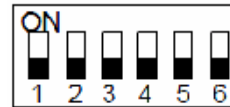
奇偶校验：None

停止位：1位；

接口：板侧为针型 5 针



接口示意图



拨码示意图

5 芯串口线	信号	信号来源	方向	说明
2	TXD	打印机	输出	控制板从主机接收数据。
3	RXD	主机	输入	控制板向主机发送数据。（当使用 X-ON/X-OFF 握手协议时，打印机向计算机发送控制码 X-ON/X-OFF。）
4	CTS	打印机	输出	该信号为“MARK”状态时，表示打印机正“忙”不能接受数据，而当该信号为“SPACE”状态时，表示打印机“准备好”，可以接受数据。
5	GND	—	—	信号地。

#### 2.1.2 波特率选择位

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

#### 2.1.3 握手方式选择位

[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)



握手方式有两种可供选择，一种是标志控制方式，另一种是 X-ON/X-OFF 协议方式。它可以通过机内的 DIP 开关 SW4 来选择。出厂时为 K4=OFF。两种握手方式如下：

SW4	握手方式	数据方向	RS-232 接口信号
ON	标志控制	数据可以进入	信号线 4 为 Space 状态
		数据不可进入	信号线 4 为 Mark 状态
OFF	X-ON/X-OFF 控制	数据可以进入	在信号线 2 上发 X-ON 码 11H
		数据不可进入	在信号线 2 上发 X-OFF 码 13H

### 2.1.4 缺纸控制选择位

SW5	说明
ON	当为 ON 时,打印机在缺纸时将停止打印,在线指示处于闪烁状态,以提示缺纸
OFF	当为 OFF 时,打印机将对缺纸状态不进行响应

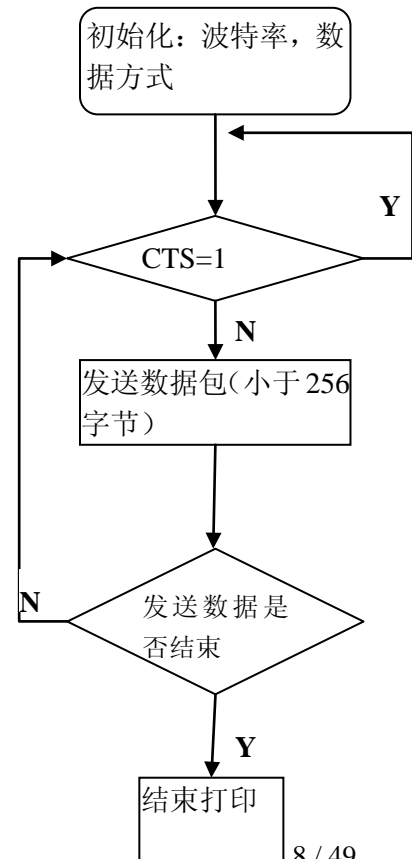
注：拨码的 6 位为无效位，待留以后版本打印机扩展应用。

### 2.1.5 串口数据发送方法

由于打印机上配有 8K 字节的缓存,当一次发送数据少于 8K 时,可直接发送数,发送方法为:



如发送的数据量很大，则在发送数据时需判断一下 CTS 标志，当此标志为 1 时，不能发送数据，为 0 时，发送数据。数据可以以包的形式发送，也可以以字节形式发送。当以包的形式时，每个数据包不得超过 256 个字节，发送流程图如下图：







## 2.2 并行接口

RD-W32-8+接口采用 26P 双排针座做为并口的通讯接口，该接口针与针之间的间距为 2.54mm。该型号的机器通过短接控制板上的 W1 可以实现 TTL 的串行通信。图 3-3 接并口示意图。

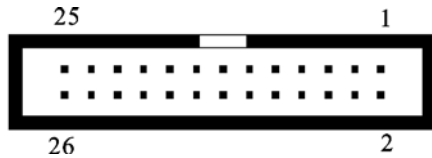


图 3-3

### 2.2.1 数据接口

26 芯排座	信 号	方 向	说 明	PC 机 DB25 并口线
1	STB/	入	数据选通触发脉冲，上升沿时读入数据	1
3	DATA1	入	8 位数据脚，逻辑 0 为低，1 为高	2
5	DATA2	入		3
7	DATA3	入		4
9	DATA4	入		5
11	DATA5	入		6
13	DATA6	入		7
15	DATA7	入		8
17	DATA8	入		9
19	ACK/	出	回答脉冲，“低”电平表示数据已被接受	10
21	BUSY	出	“高”电平表示打印机忙，不能接收数据	11
23	PE	— —	接地	— —
25	SEL	出	经电阻上拉“高”表示打印机在线	13
4	ERR/	出	经电阻上拉“高”电平表示无故障	15
2, 6, 8	NC	— —		— —
10~24	GND	— —	信号地。	12、25

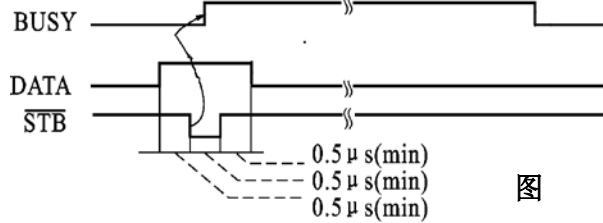
注：①信号来源一项中的“打印机”和“主机”表示信入发出的来源。

②信号逻辑电平为 EIA 电平。



### 2.2.2 并口数据发送方法

并口数据的发送相对来说比串口要麻烦些，需要 STB，BUSY 及 DATA 数据线之间的时序配合，才可以发送，图 3-4 为并口发送一字节的时序图。



图

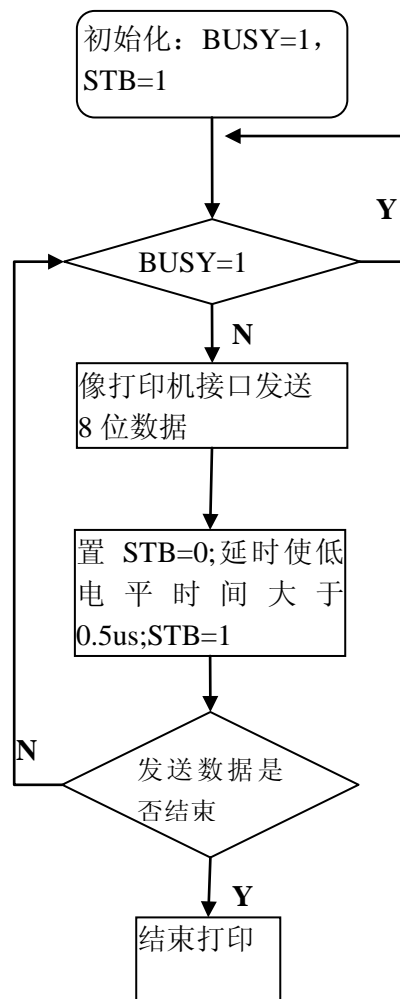
发送的步骤为：

1) 将打印机与上位机接按“2.2.1 数据接口”

所示用数据线连好。

2) 给打印机上电，并上好打印纸。

3) 开始发送数据，流程图如下图：





## 三、指令系统

### 3.1 指令表

- RD-FK12系列热敏打印机采用ESC/POS兼容指令。并增加了汉字打印、字符汉字旋转、字间距调整、条形码打印等功能。

下表中是打印指令的简表：

命令	功能
ESC @	初始化打印机
LF	打印并换行
CR	打印并回车
ESC J	打印并进纸
ESC d	打印并进纸 $n$ 行
ESC c	允许/禁止反向打印
HT	水平制表
ESC !	选择字符打印模式
ESC D	设置水平制表位
ESC -	设置/取消下划线
ESC +	设置/取消上划线
GS B	设置/取消反白打印
FS 2	设置字符旋转打印
ESC \$	设置打印绝对位置
ESC I	设置打印位置
ESC Q	设置右侧不打印字符区域
ESC 1	设置行间距
ESC SP	设置字间距
ESC a	选择对齐方式
FS r	选择上下标
ESC U	水平放大字符
ESC V	垂直放大字符
ESC X	放大字符
ESC K	图形打印命令①
ESC *	图形打印命令②
GS v	打印光栅位图
GS h	设置条形码高度
GS w	设置条形码宽度
GS H	选择可识读字符
GS Q	设置条码水平打印位置
GS k	打印条码
ESC ‘	打印曲线



<b>ESC N</b>	设置打印模式
<b>ESC v</b>	向主机传送打印机状态
<b>FS &amp;</b>	进入汉字模式
<b>FS .</b>	取消汉字模式
<b>ESC 6</b>	选择6X8字符集1
<b>ESC 7</b>	选择6X8字符集2
<b>ESC r</b>	打印深度调整指令

本章详细描述了控制打印机打印的指令，描述中的格式说明如下：

**【COMMAND】 + 【*parameter*】**

**【COMMAND】** 是命令部分，由转义字符和命令字符组成，有少量的单字节命令没有转义字符。

**【*parameter*】** 是参数部分，用斜体表示，参数并不是数字字符，而是字符的值。

本章所有例子都以 C 语言编写，其中 `SendDataToPrinter` 函数为虚拟函数，需要开发者根据主机实际情况编写，该函数定义如下：

`SendDataToPrinter(unsigned char *buffer, unsigned int len)`

说明：向打印机发送数据。

Unsigned char \*buf: 打印数据的指针；

Unsigned int len: 数据长度，单位：字节。



## 3.2 基本控制指令

### ESC @

---

[名称]	初始化打印机		
[格式]	ASCII	ESC	@
	十进制	27	64
	十六进制	1B	40
[说明]	清除打印缓冲区中的数据，复位打印参数到当前打印机缺省参数。		
[注意]	• 打印机接收缓冲区的数据并不被清除。		
[例子]	<pre>unsigned char str[2]; str[0] = 0x1B; str[1] = 0x40; SendDataToPrinter(str,2);</pre>		

### LF

---

[名称]	打印并换行	
[格式]	ASCII	LF
	十进制	10
	十六进制	0A
[说明]	把打印缓冲区中的数据打印出来，并换行。	
[注意]	• 该命令把打印位置设置为行的开始位置。	
[例子]	<pre>unsigned char str[2]; str[0] = 0x0A;//或str[0] = '\n' SendDataToPrinter(str,1);</pre>	

### CR

---

[名称]	打印并回车	
[格式]	ASCII	CR
	十进制	13
	十六进制	0D
[说明]	把打印缓冲区中的数据打印出来，并回车。	
[参考]	LF	
[例子]	<pre>unsigned char str[2]; str[0] = 0x0D;//或str[0] = '\r' SendDataToPrinter(str,1);</pre>	

### ESC J

---

[名称]	打印并进纸			
[格式]	ASCII	ESC	J	<i>n</i>
	十进制	27	74	<i>n</i>
	十六进制	1B	4A	<i>n</i>
[范围]	$0 \leq n \leq 255$			
[说明]	打印输出打印缓冲区中的数据，并进纸[ $n \times 0.125\text{m m}$ (0.0049")]。			
[注意]	• 打印结束后，将下一行的开始设定为打印起始位置。			
[例子]	<pre>unsigned char str[3]; str[0] = 0x1B;</pre>			



```
str[1] = 0x4A;  
str[2] = 0x4;  
SendDataToPrinter(str,3);//向前走纸0.5mm。
```

### ESC d

[名称] 打印并进纸 $n$ 行

[格式]

ASCII	ESC	d	$n$
十进制	27	100	$n$
十六进制	1B	64	$n$

[范围]  $0 \leq n \leq 255$

[说明] 打印输出打印缓冲区中的数据，并进纸 $n$ 行。

[注意]

- 一行的距离为24个垂直点距(0.125mm)。
- 打印结束后，该命令设置打印起始位置为行起点。

[例子]

```
unsigned char str[3];  
str[0] = 0x1B;  
str[1] = 0x64;  
str[2] = 0x4;  
SendDataToPrinter(str,3);//向前走纸4行。
```

### ESC j

[名称] 退纸 $n$ 点行

[格式]

ASCII	ESC	j	$n$
十进制	27	106	$n$
十六进制	1B	6A	$n$

[范围]  $0 \leq n \leq 255$

[说明] 倒走纸 $n$ 点行。

[注意]

- 一点行的距离为0.125mm。
- 打印结束后，该命令设置打印起始位置为行起点。

[例子]

```
unsigned char str[3];  
str[0] = 0x1B;  
str[1] = 0x6A;  
str[2] = 0x4;  
SendDataToPrinter(str,3);//向后退纸4点行。
```

### ESC c

[名称] 允许/禁止反向打印

[格式]

ASCII	ESC	c	$n$
十进制	27	99	$n$
十六进制	1B	63	$n$

[范围]  $0 \leq n \leq 1$

[描述] 当 $n=1$ 时，允许反向打印打印方向由左向右，当 $n=0$ 时，禁止反向打印，打印方向由右向左。

[注意]

- 通常在打印机垂直安装时，会使用反向打印方式。反向打印不但支持字符方式，也支持图形方式。在反向打印图形时，请注意图形单元的打印顺序，参考 ESC K。

[例子]

```
unsigned char str[3];
```



```
str[0] = 0x1B;
str[1] = 0x63;
str[2] = 0x1
SendDataToPrinter(str,3);//反向打印。
```

**HT**

- [名称] 水平制表
- [格式] ASCII        HT  
十进制        9  
十六进制     09
- [说明] 移动打印位置到下一个水平制表位置。
- [注意]
  - 通过ESC D命令设置水平制表位的位置。
  - 如果没有设置下一个水平制表位置，则该命令被忽略。
- [参考] ESC D。

**ESC !**

- [名称] 选择字符打印模式
- [格式] ASCII        ESC     !        *n*  
十进制        27     33        *n*  
十六进制     1B     21        *n*
- [说明] 命令只对字符有效，汉字无效。
- [范围]  $0 \leq n \leq 255$

位	值	意义
0	0	西文字符（半宽）字体A (12×24)
	1	西文字符（半宽）字体B (8×16)
1	-	-
2	-	-
3	0	取消加粗
	1	加粗
4	0	取消倍高模式
	1	设置倍高模式
5	0	取消倍宽模式
	1	设置倍宽模式
6	—	—
7	0	取消下划线模式
	1	设置下划线模式

- [例子] 

```
unsigned char str[3];
str[0] = 0x1B;
str[1] = 0x21;
str[2] = 0x31;
SendDataToPrinter(str,3);//在倍宽倍高模式下打印8X16字符。
```

**ESC D n1 n2 ... nk NULL**

[名称] 设置水平制表位

[格式] ASCII码 ESC D n1...nk NULL

十进制码 27 68 n1...nk 0

十六进制码 1B 44 n1...nk 00

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

[描述] 设置水平定位位置。

n 指定从一行开始的列号，用来设置水平定位位置。

k 表示将被设置水平定位点的总数。

水平制表的移动单位为12点

[注意] • 水平制表位置作为一个值储存，这个值为n个西文字符宽度，是从行的开始测量的。字符宽度包括字符间距的缺省字符宽。

• 该命令不受字符放大命令(ESC X)的影响。

• 该命令删除了之前设定的水平定位位置。

• 字符打印位置超过定位位置将被处理为普通数据。

• 按升序传输[n]k，并且在末尾放置一个NULL码0。

• 该命令中nk&gt;n(k-1),如果nk小于或等于前面的值n(k-1)，定位设定结束并且n(k-1)后面的数据按普通数据处理。

• ESC D NULL 取消所有水平定位位置。

• 即使字符宽度变化，以前指定的水平定位位置也不变。

[缺省值] 缺省定位位置为字体A (12 × 24)。

[例子] unsigned str[8];

unsigned char Order = 9;

str[0] = 0x1B;

str[1] = 0x44;

str[2] = 2;//距第1列1个字符间距

str[3] = 9;//距第1列8个字符间距

str[4] = 14;//距第1列13个字符间距

str[5] = 0; //结束

SendDataToPrinter (str,6)

SendDataToPrinter (&amp;Order,1);

SendDataToPrinter ("HT1",3);

SendDataToPrinter (&amp;Order,1);

SendDataToPrinter ("HT2",3);

SendDataToPrinter (&amp;Order,1);

SendDataToPrinter ("HT3",3);

Order = 0x0D;

SendDataToPrinter (&amp;Order,1);

SendDataToPrinter ("1234567890123456\r",17)

```

HT1    HT2    HT3
1234567890123456

```

**ESC - n**

[名称] 取消/设置下划线

[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)





[格式]    ASCII            ESC-    *n*  
          十进制            27 45 *n*  
          十六进制        1B 2D *n*

[说明]    *n* = 1,允许下划线打印; *n*=0, 取消下划线打印。

[注意]    • 反白及旋转字符此命令无效。  
          • 该命令仅对英文和汉字字符有效。

[缺省值] *n* = 0

[例子]    unsigned char str[3];  
          str[0] = 0x1B;  
          str[1] = 0x2D;  
          str[2] = 0x1;  
          SendDataToPrinter (str,3);//设置下划线

### ESC + *n*

[名称]    取消/设置上划线

[格式]    ASCII            ESC+    *n*  
          十进制            27 43 *n*  
          十六进制        1B 2B *n*

[说明]    *n* = 1,允许上划线打印; *n*=0, 取消上划线打印。

[注意]    • 反白及旋转字符此命令无效。  
          • 该命令仅对英文和汉字字符有效。

[缺省值] *n* = 0

[例子]    unsigned char str[3];  
          str[0] = 0x1B;  
          str[1] = 0x2B;  
          str[2] = 0x1;  
          SendDataToPrinter (str,3);//设置上划线

### ESC G *n*

[名称]    选择/取消双重打印模式

[格式]    ASCII            ESC    G        *n*  
          十进制            27        71        *n*  
          十六进制            1B        47        *n*

[范围]     $0 \leq n \leq 255$

[描述]    选择/取消双重打印模式。  
          • 当*n*的最低位为0时, 取消双重打印模式。  
          • 当*n*的最低位为1时, 选择双重打印模式。

[注释]    • *n*只有最低位有效。  
          • 该命令与加粗打印效果相同。

[默认值]    *n* = 0

[参考]    ESC E

[例子]    unsigned char str[3];  
[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)



```

str[0] = 0x1B;

str[1] = 0x71;

str[2] = 0x01;

SendDataToPrinter(str, 3); //打印双重打印。

```

## GS B n

[名称] 设置/取消反白打印

[格式]    ASCII            GS B    n  
          十进制            29 66    n  
          十六进制        1D 42    n

[描述]    设置或取消反白打印。  
          当n 的最低有效位为0时，取消反白模式。  
          当n 的最低有效位为1时，设置反白模式。

[注意]    • 仅n 的最低位有效。  
          • 该命令对内置字符和用户自定义字符均有效。  
          • 该命令仅对英文和汉字字符有效。

[缺省值] n = 0

[例子]    unsigned char str[3];  
          str[0] = 0x1D;  
          str[1] = 0x42;  
          str[2] = 1; //设置反白模式  
          SendDataToPrinter(str, 3);

## FS 2 n

[名称] 设置字符旋转打印

[格式]    ASCII            FS 2    n  
          十进制            28 73    n  
          十六进制        1C 49    n

[范围]    0 ≤ n ≤ 3

[描述]    设置字符旋转模式

N (十进制)	意义
0	不旋转
1	设置90° 逆时针旋转
2	设置180° 逆时针旋转
3	设置270° 逆时针旋转

[注意]    • 在90度或270度旋转模式下，字符放大命令的宽高放大方向与一般模式下的放大方向相反。

[缺省值] n = 0

[例子]    unsigned char str[3];  
          str[0] = 0x1C;  
          str[1] = 0x49;  
          str[2] = 1; //设置90° 旋转  
          SendDataToPrinter(str, 3);



## ESC \$ nL nH

- [名称] 设置打印绝对位置
- [格式]   ASCII       ESC\$    *nL nH*  
          十进制       27 36   *nL nH*  
          十六进制     1B 24   *nL nH*
- [范围]    $0 \leq nL + (nH \times 256) < 384$
- [描述]   设定从一行的开始到将要打印字符的位置之间的距离。  
          从一行的开始到打印位置的距离为N个水平点距。  
          *nL nH*是双字节无符号整数N的低位和高位,  $N = nL + nH \square 256$   
          指令对曲线、图形、字符、汉字、条码全部有效
- [注意]   • 如果设定的打印位置超出了可打印区域 ( $N > 832$ ) , 命令将被忽略。
- [例子]   

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x24;
str[2] = 32;//
SendDataToPrinter (str, 3); //绝对位置设为距左边界32水平点距。
```

## ESC I n

- [名称]   设置左侧不打印区域
- [格式]   ASCII       ESCI    *n*  
          十进制       27 108 *n*  
          十六进制     1B 6C   *n*
- [范围]    $0 \leq n \leq 32$
- [描述]   设定左侧不打印的字符数。  
          从一行的开始到打印位置的距离为*n*个西文字符宽度。
- [注意]   • 如果设定的打印位置超出了可打印区域, 命令将被忽略。  
          • 字符宽度包括字符间距的缺省字符宽。
- [例子]   

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x6C;
str[2] = 3;//
SendDataToPrinter (str, 3); //左侧位置设为距左边界3个西文字符宽度。
```

## ESC Q n

- [名称]   设置右侧不打印区域
- [格式]   ASCII       ESCQ    *n*  
          十进制       27 81   *n*  
          十六进制     1B 51   *n*
- [范围]    $0 \leq n \leq 32$
- [描述]   设定右侧不打印的字符数。
- [注意]   • 如果设定的打印位置超出了可打印区域, 命令将被忽略。  
          • 字符宽度包括字符间距的缺省字符宽。
- [例子]   

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x51;
str[2] = 3;//
```



SendDataToPrinter (str, 3); //右侧设置3个西文字符宽区域不打印。

## ESC 1 n

[名称] 设置行间距

[格式]   ASCII           ESC1    *n*  
           十进制        27  49   *n*  
           十六进制     1B  31   *n*

[范围]    $0 \leq n \leq 255$

[描述]   设置字符行间距为*n*个垂直点距。

[缺省值] *n* = 3

[例子]   unsigned char str[4];  
           str[0] = 0x1B;  
           str[1] = 0x31;  
           str[2] = 8;  
           SendDataToPrinter(str,3);//设置行间距为8个垂直点距。

## ESC SP n

[名称] 设置字间距

[格式]   ASCII           ESCSP   *n*  
           十进制        27  32   *n*  
           十六进制     1B  20   *n*

[范围]    $0 \leq n \leq 255$

[描述]   设置字符字间距为*n*个水平点距。

[缺省值] *n* = 0

[例子]   unsigned char str[4];  
           str[0] = 0x1B;  
           str[1] = 0x20;  
           str[2] = 8;  
           SendDataToPrinter(str,3);//设置字间距为8个水平点距。

## ESC a n

[名称] 选择对齐方式

[格式]   ASCII           ESCa    *n*  
           十进制        27  97   *n*  
           十六进制     1B  61   *n*

[范围]    $0 \leq n \leq 2$

[描述]   将一行数据按照*n*指定的位置对齐。  
           对齐方式对图形、字符、汉字、条码有效。曲线无效。

<i>N</i>	意义
0	左对齐
1	居中
2	右对齐

[注意]   •仅在一行的开始处理时，该命令才有效。

[缺省值] *n* = 0

[例子]   unsigned char str[4];

[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)



```
str[0] = 0x1B;  
str[1] = 0x61;  
str[2] = 1;  
SendDataToPrinter(str,3);//设置为居中方式打印
```

### FS r n

---

[名称] 选择上下标

[格式]   ASCII        FS r n  
          十进制       28 114 n  
          十六进制     1C 72 n

[范围] 0 ≤ n ≤ 1

[描述] n=0 选择上标，一行字符图形顶部对齐。  
      n=1 选择下标，一行字符图形低部对齐

[注意] 

- 该命令对所有字符(英数字符和汉字) 有效。
- 如果n 在定义范围之外，忽略该命令。

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

### ESC U n

---

[名称] 水平放大字符

[格式]   ASCII        ESCU n  
          十进制       27 85 n  
          十六进制     1B 55 n

[范围] 1 ≤ n ≤ 8

[注意] 

- 该命令对所有字符(英数字符和汉字) 有效。
- 如果n 在定义范围之外，忽略该命令。

[参考] ESC X

[例子] unsigned char str[4];  
      str[0] = 0x1B;  
      str[1] = 0x55;  
      str[2] = 2;  
      SendDataToPrinter(str,3);//设置水平放大2倍.

### ESC V n

---

[名称] 垂直放大字符

[格式]   ASCII        ESCV n  
          十进制       27 86 n  
          十六进制     1B 56 n

[范围] 1 ≤ n ≤ 8

[注意] 

- 该命令对所有字符(英数字符和汉字) 有效。
- 如果n 在定义范围之外，忽略该命令。

[参考] ESC X

[例子] unsigned char str[4];  
[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)



```
str[0] = 0x1B;
str[1] = 0x56;
str[2] = 2;
SendDataToPrinter(str,3);//设置垂直放大2倍.
```

## ESC X

[名称] 放大字符

[格式]	ASCII	ESCX	n1	n2
	十进制	27 88	n1	n2
	十六进制	1B 58	n1	n2

[范围]  $1 \leq n \leq 8 (1 \leq n1 \text{水平倍数} \leq 8, 1 \leq n2 \text{垂直倍数} \leq 8)$

- [注意]
- 该命令对除条码识读字符外的所有字符(英数字符和汉字) 有效。
  - 如果n 在定义范围之外，忽略该命令。
  - 垂直方向是指进纸方向，水平方向与进纸方向垂直。然而，当字符方向顺时针旋转90°后，垂直方向与水平方向之间的关系颠倒，也就是说本命令优先级低于于FS 2，当两个命令同时有效时，字符显示是先旋转，再放大。

[例子]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x58;
str[2] = 2;
str[3] = 2;
SendDataToPrinter(str,4);//设置横向纵向放大2.
```

## ESC N

[名称] 设置打印机模式

[格式]	ASCII	ESC	N	n
	十进制	27	78	n
	十六进制	1B	4E	n

[范围]:  $0 \leq n \leq 1$

- 缺省值:  $n = 0$ 。  
 $n = 0$  打印机为正常打印模式。  
 $n = 1$  打印机为精细打印模式 (32点/行)。

[描述]: 设置打印机的打印模式

- [注意]
- ESC @ 恢复为缺省值 0。
  - 行首有效。
  - 在收到设置为精细打印模式命令后，打印机一直处于精细打印模式，直到接收到 ESC @命令或者使用本命令设置为正常打印模式。

[例子]

```
unsigned char str[4];
str[0] = 0x1B;
str[1] = 0x4E;
str[2] = 0x01;
SendDataToPrinter(str,3);//设置精细模式.
```

## ESC O

[名称] 设置打印机打印功率

[格式]	ASCII	ESC	O	n
------	-------	-----	---	---



十进制	27	79	n
十六进制	1B	4F	n

[范围]:  $0 \leq n \leq 3$

缺省值:  $n = 0$ 。

$n = 0$  打印机为正常打印模式。

$n = 1$  打印机为低功耗1打印模式2区加热

$n = 2$  打印机为低功耗1打印模式4区加热

$n = 3$  打印机为低功耗1打印模式7区加热

[描述]: 设置打印机的打印模式

[注意] •ESC @ 恢复为缺省值 0。

•行首有效。

•在收到设置为低功耗打印模式命令后, 打印机一直处于低功耗打印模式, 直到接收到 ESC @ 命令或者使用本命令设置为正常打印模式。

[例子] unsigned char str[4];

str[0] = 0x1B;

str[1] = 0x4F;

str[2] = 0x01;

SendDataToPrinter(str,3);//设置低功耗打印

## ESC K nL nH d1 d2 .....dk

[名称] 图形打印命令①

[格式] ASCII ESC K nL nH d1...dk

十进制 27 75 nL nH d1...dk

十六进制 1B 4B nL nH d1...dk

[范围]:  $0 \leq nL \leq 255$

$0 \leq nH \leq 3$

$0 \leq d \leq 255$

[描述]: 本命令只能打印高度为 8 点, 宽度不超过可打印区域的黑白位图。

nL nH 分别为无符号型双字节整数 N 的低位和低位字节, 表示水平方向上位图中的点数。

[参考] ESC \*

[注意] •该图形命令受字符放大命令影响。

•当采用反向打印方式时, 要按图形从下到上的顺序依次打印每个图形单元。

[例子] unsigned char str[30];

unsigned char i=0;

str[i++] = 0x1B;

str[i++] = 0x4B;

str[i++] = 15; //打印15个点宽图形

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);//发送图形打印命令。

## ESC \* m nL nH d1...dk

[名称] 图形打印命令②



[格式] ASCII            ESC \*    m nL nH d1...dk  
 十进制            27 42 m nL nH d1...dk  
 十六进制          1B 2A m nL nH d1...dk

[范围]: m = 0, 1, 32, 33  
 0 ≤ nL ≤ 255  
 0 ≤ nH ≤ 3  
 0 ≤ d ≤ 255

[描述]: 本命令只能打印高度为 8 点或 24 点，宽度不超过可打印区域的黑白位图。  
 各参数含义如下：

用 m 选择位图的模式，位图的水平方向点数由 nL 和 nH 指定，如下所示：

m	垂直点数（高度）	倍宽模式
0	8	两倍宽
1	8	单倍宽
32	24	两倍宽
33	24	单倍宽

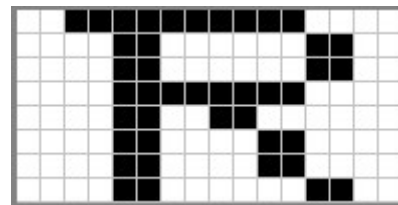
nL nH 分别为无符号型双字节整数 N 的低位和高位字节，表示水平方向上位图中的点数。N 在单倍宽时最大值为 832，在双倍宽时其值最大为 416。

d1.....dk 表示位图数据：具体格式见下图：

[例子]

例 1： m=0(8 点、两倍宽)d1 表示打印的第 1、2 列点的数据，dk 表示打印的第 2k-1 和 2k 列点的数据,bn 表示字节的第 n 位

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



程序代码如下：

```
unsigned char str[100];
j=0;
str [j++] = 0x1B;
str r[j++] = 0x2A;
str [j++] = 0; //m=0(高度 8 点、倍宽)
str [j++] = 8; //图象宽度为 8dots
str [j++] = 0;
//位图数据
```

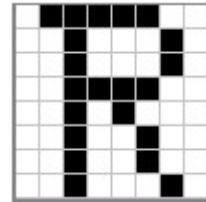




```
str [j++] = 0x00;str [j++] = 0x80;str [j++] = 0xFF;str [j++] = 0x90;str [j++] = 0x98;
str [j++] = 0x96;str [j++] = 0x61;str [j++] = 0x00;str [j++] = 0x0D;//打印出图形
SendDataToPrinter(str,j);
```

例 2: m=1(8 点、单倍宽)d1 表示打印的第 1 列点的数 据， dk 表示打印的第 k 列点的数 据， bn 表示字节的第 n 位

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



程序代码如下:

```
unsigned char str[100];
j=0;
str[j++] = 0x1B;
str[j++] = 0x2A;
str[j++] = 1; //m=1(高度 8 点、不放大)
str [j++] = 8; //图象宽度为 8dots
str [j++] = 0;
//位图数据
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; ;//打印出图形
SendDataToPrinter(str,j);
```

例 3: m=32(24 点、两倍宽)d1、d2、d3 表示打印的第 1、2、3 列点的数 据， 依此类推； bn 表示字节的第 n 位

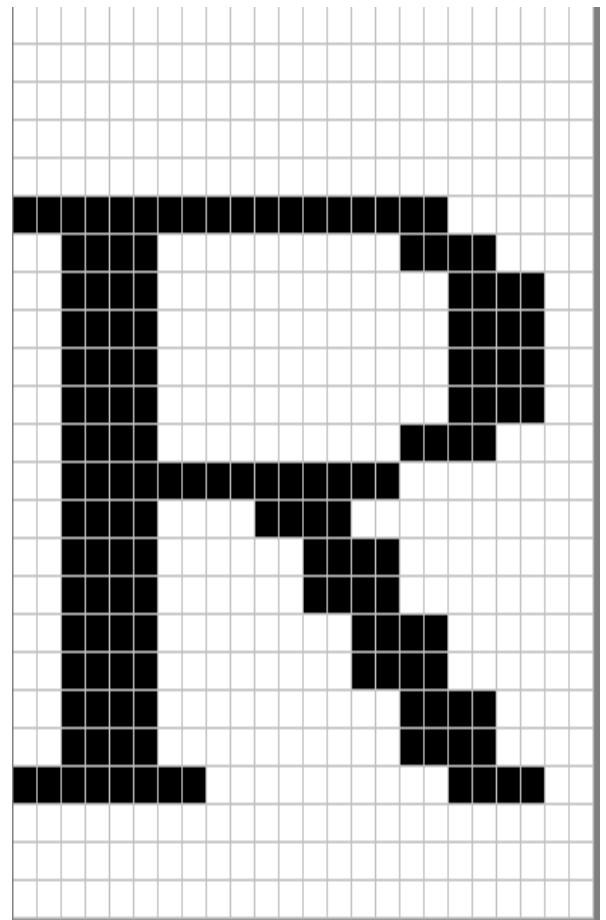
	d4	d7								D	d49	
d1	0	0	0	0	0	0	0	0	0	0	b7	
	0	0	0	0	0	0	0	0	0	0	b6	
	0	0	0	0	0	0	0	0	0	0	b5	
	1	1	1	1	1	1	1	1	0	0	b4	
	0	1	1	0	0	0	0	0	1	1	0	b3
	0	1	1	0	0	0	0	0	1	1	0	b2
	0	1	1	0	0	0	0	0	1	1	0	b1
	0	1	1	0	0	0	0	0	1	1	0	b0
	0	1	1	0	0	0	0	0	1	1	0	b7



d2

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	0	1	1	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

d3



程序代码如下:

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

```
j=0;
```

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

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

```
str[j++] = 32; //m=32(高度 24 点、倍宽)
```

```
str[j++] = 12; //图象宽度为 12dots
```

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

```
//位图数据
```

```
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;//打印出当前的图形
```

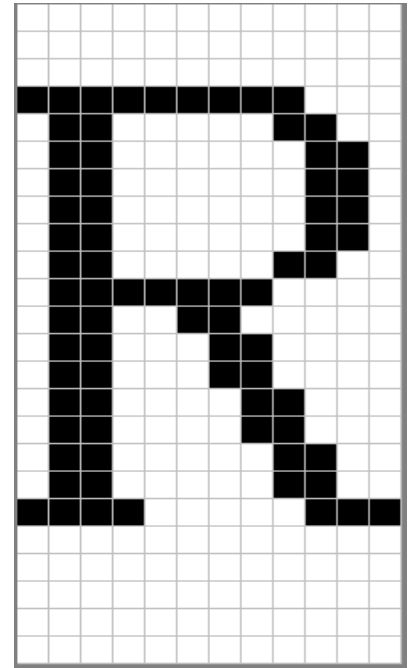
```
SendDataToPrinter(str,j);
```

[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)



例 4: m=33(24 点、无放大)d1、d2、d3 表示打印的第 1、2、3 列点的数据，依此类推；bn 表示字节的第 n 位

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



程序代码如下:

```

unsigned char str[200];
j=0;
str[j++] = 0x1B;
str[j++] = 0x2A;
str[j++] = 32; //m=33(高度 24 点、无放大)
str[j++] = 12; //图象宽度为 12dots
str[j++] = 0;
//位图数据
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;//打印出当前的图形
SendDataToPrinter(str,j);
```

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

[名称]打印光栅位图

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

十进制 29 118 48 m xL xH yL yH d1...dk

十六进制 1D 76 30 m xL xH yL yH d1...dk

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

$0 \leq xL \leq 255$

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

$0 \leq yL \leq 255$

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

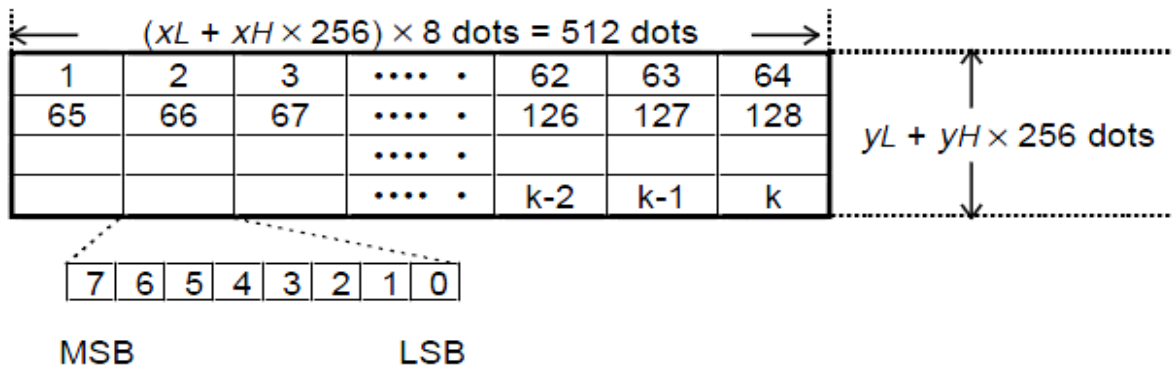
$0 \leq d \leq 255$

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

m	模式	垂直点密度	水平点密度
0,48	正常	203.2dpi	203.2dpi
1,49	倍宽	203.2dpi	101.6dpi
2,50	倍高	101.6dpi	203.2dpi
3,51	倍宽倍高	101.6dpi	101.6dpi

□ xL, xH位图水平宽度所占字节数。

□ yL, yH,位图垂直点长度



## GS h n

[名称] 设置条形码高度

[格式] ASCII GS h n

十进制 29 104 n

十六进制 1D 68 n

[范围]:  $1 \leq n \leq 255$

[描述]: 设置选择条形码高度。

N为垂直方向的点数。

缺省值:  $n = 48$

[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)



```
[例子] unsigned char str[4];
        str[0] = 0x1D;
        str[1] = 0x68;
        str[2] = 30;
        SendDataToPrinter(str,3);//设置条码高度为30个垂直点距高
```

## GS w n

[名称] 设置条形码宽度

[格式] ASCII        GS w n  
          十进制        29 119 n  
          十六进制      1D 77 n

[范围]:  $1 \leq n \leq 4$

[描述]: 设置条形码水平尺寸。 n 取值定义如下:

N	多级条形码单位宽度 (mm)	二进制条码宽度	
		窄条线宽度	宽条线宽度
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

```
[例子] unsigned char str[4];
        str[0] = 0x1D;
        str[1] = 0x77;
        str[2] = 3;
        SendDataToPrinter(str,3);//设置条码宽度
```

## GS f n

[名称] **选择条码字符字体**

[格式] ASCII码        GS f n  
          十进制码      29 102 n  
          十六进制码    1D 66 n

[范围] n = 0, 1, 48, 49

[描述] 打印条码时, 为HRI 字符选择一种字体  
用n 来选择字体如下:

n	字体
0, 48	标准ASCII码字符 (12×24)
1, 49	标准ASCII码字符 (8×16)

[注释]

- HRI 字符是对条码内容注释的字符。
- HRI 字符的打印位置由GS H 命令指定。

[默认值] n = 1

[参考] GS H, GS k

## GS H n

[名称] 选择可识读字符

[格式] ASCII        GS h n



十进制      29 72 *n*  
 十六进制    1D 48 *n*

[范围]:  $0 \leq n \leq 2$

[描述]: 打印条形码时, 选择可识读字符的打印。



*n* 选择打印, 如下所示:

<i>n</i>	识读字符位置
0	不打印
1	条形码上方
2	条形码下方

[缺省值]:  $n = 0$

```
[例子] unsigned char str[4];
        str[0] = 0x1D;
        str[1] = 0x48;
        str[2] = 2;
        SendDataToPrinter(str,3);//设置可识读字符在条码的下方打印。
```

## GS Q n

[名称] 设置条码水平打印位置  
 [格式] ASCII      GS Q *n*  
          十进制      29 81 *n*  
          十六进制    1D 51 *n*

[范围]:  $0 \leq n \leq 255$

[描述]: 设置条码从一行开始到打印位置的距离为N个水平点距。

[缺省值]:  $n = 0$

```
[例子] unsigned char str[4];
        str[0] = 0x1D;
        str[1] = 0x51;
        str[2] = 32;
        SendDataToPrinter(str,3);//
```

## GS k

[名称]: 打印条码。  
 [格式]: 该命令有两种格式:  
         格式1: ( $0 \leq m \leq 8$ )  
           ASCII码: GS k *m* *d1*...*dk* NUL  
           十进制码: 29 107 *m* *d1*...*dk* 0  
           十六进制码: 1D 6B *m* *d1*...*dk* 00  
         格式2: ( $65 \leq m \leq 73$ )  
           ASCII码: GS k *m* *n* *d1*...*dn*  
           十进制码: 29 107 *m* *n* *d1*...*dn*  
           十六进制码: 1D 6B *m* *n* *d1*...*dn*

[范围]:  $0 \leq m \leq 8$  (*k* 和 *d* 取决于使用的条码系统)



$65 \leq m \leq 73$  (n 和d取决于使用的条码系统)

n为打印条码的数据长度

[描述]: 选定条码系统并打印条码。

M定义所使用的条码系统, 如下表:

m	条码类型	长度	范围	
格式 1	0	UPC-A	$11 \square k \square 12$	$48 \square d \square 57$
	1	UPC-E	$K \square \square$	$48 \square d \square 57$
	2	JAN13 (EAN13)	$12 \square k \square 13$	$48 \square d \square 57$
	3	JAN 8 (EAN8)	$7 \square k \square 8$	$48 \square d \square 57$
	4	CODE39	$1 \square k$	$48 \square d \square 57, 65 \square d \square 90, 32, 36, 37, 43, 45, 46, 47$
	5	ITF	$1 \square k$ (even number)	$48 \square d \square 57$
	6	CODABAR	$1 \square k$	$48 \square d \square 57, 65 \square d \square 68, 36, 43, 45, 46, 47, 58$
格式 2	65	UPC-A	$11 \square n \square 12$	$48 \square d \square 57$
	66	UPC-E	$n=8$	$48 \square d \square 57$
	67	JAN13 (EAN13)	$12 \square n \square 13$	$48 \square d \square 57$
	68	JAN 8 (EAN8)	$7 \square n \square 8$	$48 \square d \square 57$
	69	CODE39	$1 \square n \square 255$	$48 \square d \square 57, 65 \square d \square 90, 32, 36, 37, 43, 45, 46, 47$
	70	ITF	$1 \square n \square 255$ (even number)	$48 \square d \square 57$
	71	CODABAR	$1 \square n \square 255$	$48 \square d \square 57, 65 \square d \square 68, 36, 43, 45, 46, 47, 58$
	72	CODE93	$1 \square n \square 255$	$0 \square d \square 127$
	73	CODE128	$2 \square n \square 255$	$0 \square d \square 127$

\*除UPC-E外, 其它条码的效验位都有打印机自动算出, 用户可不加效验位。

\*CODE39无需加\*

[注意]: • 当使用格式 1 的命令时, 如果条码类型中规定了条码的数据长度, 则 k (打印机接收到的条码数据长度) 应当等于规定的的数据长度, 如果不等于规定的的数据长度, 则该指令无效。有关条码的数据位长度见【附录 B】。

• 打印机接收到的条码数据字符应该包含在条码类型规定的字符集中, 如果条码数据字符中有字符超出了字符集, 该命令无效。有关条码的字符集见【附录 B】。

• 当使用格式 2 的命令时, n 的值要等于条码的规定数据长度 (如果该类型的条码规定了数据位长度的话), 如果 n 值不等于条码的规定数据位长度, 那么该命令无效, 有关条码的数据位长度见【附录 B】。

• INTERLEAVED 25(ITF)条码的数据长度为偶数, 如果使用格式 1 打印 ITF 条码, 那么 k (打印机接收到的条码数据长度) 的值要为偶数, 如果为奇数, 最后一位数据将被忽略。如果使用格式 2



打印 ITF 条码，那么 n 值要为偶数，如果 n 为奇数，那么最后一位数据将被忽略。

- 如果水平方向尺寸超出了打印区域，超出的部分将被忽略。
- 该命令不受打印模式(粗体、重叠、下划线、字符大小、或反白打印)影响。
- 打印条码时要遵守条码的编码规范，否则将会导致条码无法扫描。
- 打印机不计算校验码，如果条码需要校验码，需将校验码包含在条码数据当中，打印机不负责核对校验码的正误，用户计算校验码错误将导致条码无法扫描。
- CODE39 码不包括扩展 CODE39 码 (EXTERN CODE 39)。
- CODE93 码不包括扩展 CODE93 码 (EXTERN CODE 93)。
- CODE128 条码数据串的头部必须是编码集选择字符(CODE A, CODE B, 或 CODE C),在一个条码内部也可切换编码集。用字符 '{' 和一个字符组合用以定义特殊功能。通过连续传送两次 '{' 定义 ASCII 字符'{'。如下图

ASCII	HEX	功能
{A	7B, 41	选择编码集A
{B	7B, 42	选择编码集B
{C	7B, 43	选择编码集C
{S	7B, 53	SHIFT
{1	7B, 31	FNC1
{2	7B, 32	FNC2
{3	7B, 33	FNC3
{4	7B, 34	FNC4

## ESC ”

[名称] 设置打印曲线打印方式

[格式]    ASCII        ESC    “        *n*

          十进制        27     34        *n*

          十六进制     1B     22        *n*

[范围]     $0 \leq n \leq 1$

[描述]    此指令影响1B 23 曲线命令打印方式

*n* = 0    1B 23曲线命令打印曲线不自动连点。

*n* = 1    1B 23曲线命令打印曲线自动连点。

[注意]    打印机在接受到此命令后，将会把后续收到的第一条曲线命令作为打印曲线的初始位置。

## ESC #

[名称] 打印n条曲线命令

[格式]    ASCII码 ESC # *n* *x1L* *x1H* ... *xnL* *xnH*

          十进制码 27 35 *n* *x1L* *x1H* ... *xnL* *xnH*

          十六进制码 1B 23 *n* *x1L* *x1H* ... *xnL* *xnH*





[范围]  $0 \leq n \leq 8$

[描述]  $n$ 为所需要打印的曲线条数。

*xnL* 曲线点横向坐标的低8位。

*xnH* 曲线点横向坐标的高8位。

[描述] 打印机根据 $n$ 值的设置打印 $n$ 条曲线。

每条曲线自动连点。

打印一条曲线时设置 $n=1$ ;

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

```
int i;
```

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

```
SendStr[1] = 0x23;
```

```
SendStr[2] = 2; //打印2条曲线
```

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

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

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

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

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

---

### **ESC (**

---

[名称] 打印一水平上 $n$ 个线段

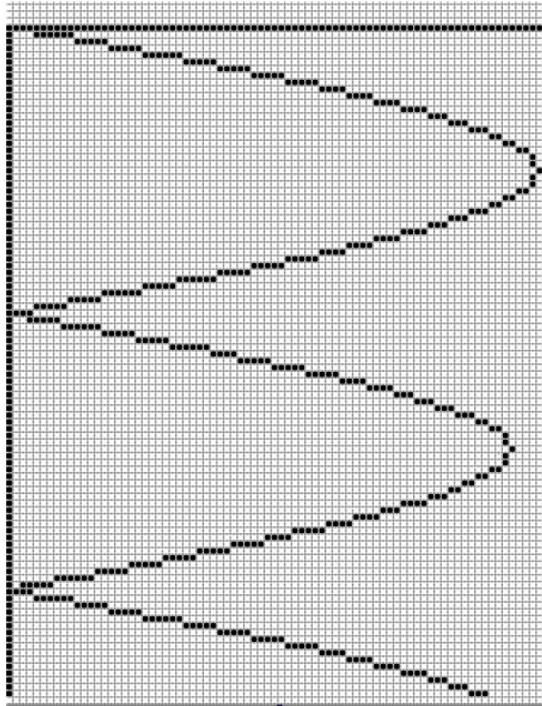
[格式] ASCII码 ESC ( *n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH*

十进制码 27 40 *n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH*

十六进制码 1B 28 *n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH*

[范围]  $0 \leq n \leq 8$

[描述] 如下打印放大图所示：每条曲线都是由很多水平线段(点可视为长度为1的线段)组成。本指令为打印一水平上 $n$ 个线段，连续使用该指令可以打印出用户所需要的线段。



n 线段数量;

xksL 第k条线段起始点横向坐标的低位;

xksH 第k条线段起始点横向坐标的高位;

xkeL 第k条线段结束点横向坐标的低位;

xkeH 第k条线段结束点横向坐标的高位;

坐标从打印区域最左侧开始计算, 最小值为1, 最大值为384, 也就是说xkeL+xkeH\*256最大值为384。

线段的数据不必按照顺序排列;

[注意] • 当打印一个点时, xkeL=xksL, xkeH=xksH。

```

char SendStr[8];
int i;
short y1,y2,y1s,y2s;
//打印y轴轴线(一条线)
SendStr[0] = 0x1B;
SendStr[1] = 0x28;
SendStr[2] = 1; //一条线段
SendStr[3] = 30; //起始点为30
SendStr[4] = 0;
SendStr[5] = 104; //结束点360
SendStr[6] = 1;
SendDataToPrinter (SendStr,7);

```

**ESC ‘**

[名称] 打印一水平线上n个点

[格式] ASCII码 ESC ‘ nL nH x1L x1H x21L x21H ..... xkL xkH CR

十进制码 27 39 nL nH x1L x1H x21L x21H ..... xkL xkH 13



十六进制码 1B 27 nL nH x1L x1H x21L x21H ..... xkL xkH 0D

[范围] :  $0 \leq nL \leq 255$

$0 \leq nHL \leq 1$

曲线点数  $N = nH \times 256 + nL$

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

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

[例子]: 曲线关系函数为下面 5 个函数

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

$$Y2=50-40*\text{abs}(-0.01*X)*\sin(X/10)$$

$$Y3=50$$

$$Y4=50+40*\text{abs}(-0.1*X)$$

$$Y5=50-40*\text{abs}(-0.01*X)$$

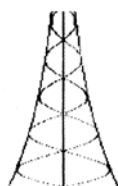
[打印例程]:

```

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

```

[结果]:



**ESC v**

[名称] 向主机传送打印机状态

[格式] ASCII        ESCv  
 十进制        27 118  
 十六进制     1B 76

[描述]: 向主机传送打印机状态。

[注意]: • 仅串口型打印机有效。

位	功能	值	
		0	1
0	纸检测器	无纸	有纸
1	工作状态	空闲	打印中
2	接收缓冲区	未满	满
3	打印机状态	正常	错误
4	未定义	---	---
5	未定义	---	---
6	未定义	---	---
7	未定义	---	---

[例子] unsigned char str[4];  
 str[0] = 0x1B;  
 str[1] = 0x76;  
 SendDataToPrinter(str,2);//向打印机发送状态查询命令。

**FS &**

[名称] 进入汉字模式

[格式] ASCII        FS &  
 十进制        28 38  
 十六进制     1C 26

[描述]: 打印机进入汉字打印模式。

[注意]: • 上电后打印机默认为汉字打印模式。

[例子] unsigned char str[4];  
 str[0] = 0x1C;  
 str[1] = 0x26;  
 SendDataToPrinter(str,2);//进入汉字打印模式。

**FS.**

[名称] 取消汉字模式

[格式] ASCII        FS .  
 十进制        28 46  
 十六进制     1C 2E

[描述]: 取消汉字字符模式,当取消汉字字符模式后,超过0x80的编码仍然当作12X24 ASCII字符处理,将不再打印汉字,除非再用FS &amp;命令选择汉字模式。打印机进入汉字打印模式。

[例子] unsigned char str[4];  
 str[0] = 0x1C;



```
str[1] = 0x2E;
SendDataToPrinter(str,2);//进入ASCII字符打印模式。
```

## FS 2 c1 c2 d1...dk

[名称] 定义用户自定义汉字(24\*24)

[格式] ASCII码 FS 2 c1 c2 d1...dk  
 十六进制码 1C 32 c1 c2 d1...dk  
 十进制码 28 50 c1 c2 d1...dk

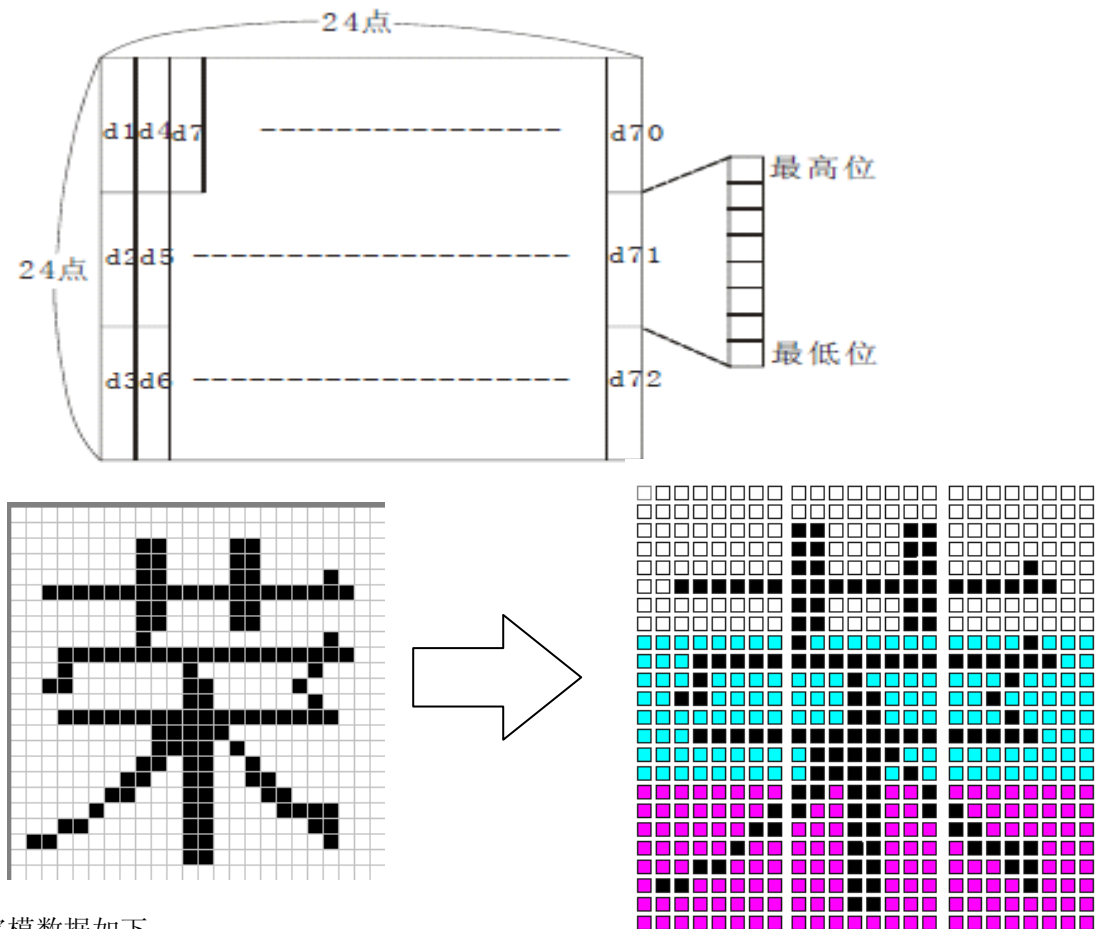
[范围] c1 ,c2代表定义字符的字符编码  
 c1 = 0XFE  
 0XA1 ≤ c2 ≤ 0XFE  
 0 ≤ d ≤ 255  
 k = 72

[描述] 定义由c1, c2指定的汉字

[注释]

- c1 , c2代表用户自定义汉字的编码, c1指定第一个字节, c2指定第二个字节。
- d代表数据。1表示打印一个点, 0表示不打印点。

[默认值] 没有自定义汉字  
 自定义汉字字型与数据之间关系见下图:



字模数据如下:

d1=0x00 , d2=0x00 , d3=0x00 , d4=0x00 , d5=0x00 , d6=0x04 , d7=0x04 , d8=0x10 , d9=0x04 , d10=0x04  
 d11=0x74, d12=0x08, d13=0x04, d14=0x44, d15=0x08, d16=0x04, d17=0x44, d18=0x10, d19=0x04, d20=0x44



## RD-K110 系列用户开发手册(V2.20)

d21=0x20, d22=0x04, d23=0x44, d24=0x60, d25=0x3F, d26=0xC4, d27=0xC0, d28=0x3F, d29=0x47, d30=0x80  
 d31=0x04, d32=0x47, d33=0x00, d34=0x04, d35=0x7F, d36=0xFE, d37=0x04, d38=0x5F, d39=0xFE, d40=0x04  
 d41=0x46, d42=0x00, d43=0x3F, d44=0x45, d45=0x00, d46=0x3F, d47=0x44, d48=0xC0, d49=0x04, d50=0x44  
 d51=0x60, d52=0x04, d53=0x44, d54=0x30, d55=0x04, d56=0x54, d57=0x10, d58=0x04, d59=0x6C, d60=0x18  
 d61=0x0C, d62=0xC4, d63=0x1C, d64=0x04, d65=0x40, d66=0x00, d67=0x00, d68=0x00, d69=0x00, d70=0x00  
 d71=0x00, d72=0x00

### FS 3 c1 c2 d1...dk

[名称] 定义用户自定义汉字(16\*16)

[格式] ASCII码 FS 2 c1 c2 d1...dk  
 十六进制码 1C 32 c1 c2 d1...dk  
 十进制码 28 50 c1 c2 d1...dk

[范围] c1 , c2代表定义字符的字符编码

c1 = 0XFE  
 0XA1 ≤ c2 ≤ 0XFE  
 0 ≤ d ≤ 255  
 k = 32

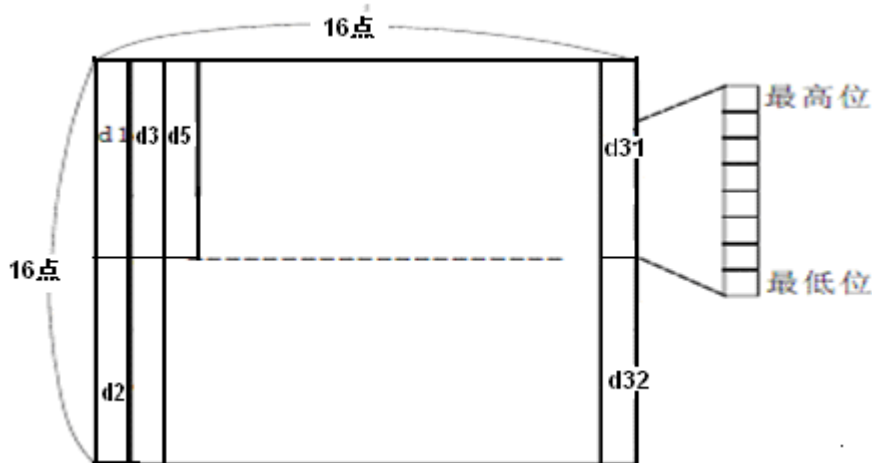
[描述] 定义由c1, c2指定的汉字

[注释]

- c1 , c2代表用户自定义汉字的编码, c1指定第一个字节, c2指定第二个字节。
- d代表数据。1表示打印一个点, 0表示不打印点。

[默认值] 没有自定义汉字

自定义汉字字型与数据之间关系见下图:



### ESC 6

[名称] 选择6X8字符集1

[格式] ASCII ESC6  
 十进制 27 54  
 十六进制 1B 36

[描述]: 在该命令输入之后的所有字符均使用字符集1中的字符打印(见附录D)字符集1中有6x8点阵字符224个, 包括ASCII字符及各种图形符号等。代码范围20H~FFH(32~255)。



[例子] unsigned char str[4];  
 str[0] = 0x1B;  
 str[1] = 0x36;  
 SendDataToPrinter(str,2);//打印6X8字符集1中的字符。

## ESC 7

[名称] 选择6X8字符集2

[格式] ASCII        ESC7  
 十进制        27 55  
 十六进制     1B 37

[描述]: 在该命令输入之后的所有字符均使用字符集2中的字符打印（见符录D），字符集2中有6x8点阵字符224个，包括德、法、俄文、日语片假名等。代码范围20H~FFH(32~255)。

[例子] unsigned char str[4];  
 str[0] = 0x1B;  
 str[1] = 0x37;  
 SendDataToPrinter(str,2);// 打印6X8字符集2中的字符。

## ESC r

[名称] 打印深度调整指令

[格式] ASCII        ESC7  
 十进制        27 114 nl nh  
 十六进制     1B 72 nl nh  
 范围]  $100 \leq n < 10000$

[描述]: 打印机默认深度为800（nl=20,nh=03）。在该命令输入之后将会改变打印机的打印深度，同时会改变打印机的打印速度，n值越大，打印速度越低，n值超出最大值，设置无效。

[注意]: •请根据打印情况来调整，不要将深度调整到太大。N值太大，将会影响打印效果同时会对打印机的使用寿命有一定的影响。

[例子] unsigned char str[4];  
 str[0] = 0x1B;  
 str[1] = 0x72;  
 str[2] = 0xDC;            //nl=1500mod256  
 str[3] = 0x05;            //nh=1500/256  
 SendDataToPrinter(str,4);// 设置打印深度为1500。

## ESC % n

[名称] 允许/禁止用户自定义字符

[格式] ASCII        ESC    %        *n*  
 十进制        27     37        *n*  
 十六进制     1B     25        *n*



[范围]  $0 \leq n \leq 255$

[描述] 允许或禁止用户自定义字符。

- 当n的最低位为0时，禁止用户自定义字符。
- 当n 的最低位为1 时，使用用户自定义字符。

[注释]

- 当禁止使用用户自定义字符的时候，自动使用内部字库。
- n 只有最低位有效。

[默认值]  $n = 0$

[参考] ESC &, ESC ?

[例子] unsigned char str[3];

str[0] = 0x1B;

str[1] = 0x25;

str[2] = 0x01;

SendDataToPrinter(str, 3); //选择用户自定义字符。

## ESC ? n

[名称] **取消用户自定义字符**

[格式] ASCII码      ESC ? n

十进制码      27 63 n

十六进制码    1B 3F n

[范围]  $32 \leq n \leq 127$

[描述] 取消用户自定义字符。

[注释]

- 取消用户自定义字符中代码为n的字符。取消后，此字符使用内部字库。
- 如果自定义字符中没有该字符，该命令被忽略。

[参考] ESC &, ESC %

## ESC & y c1 c2 [x1 d1...d(y × x1)]...[xk d1...d(y × xk)]

[名称] **定义用户自定义字符**

[格式] ASCII 码      ESC & y c1 c2 [x1 d1...d(y × x1)]...[xk d1...d(y × xk)]

十进制码      27 38 y c1 c2 [x1 d1...d(y × x1)]...[xk d1...d(y × xk)]

十六进制码    1B 26 y c1 c2 [x1 d1...d(y × x1)]...[xk d1...d(y × xk)]

[范围]  $y = 3$

$32 \leq c1 \leq c2 \leq 127$

$0 \leq x \leq 12$  标准ASCII码字体A (12 × 24)

$0 \leq x \leq 9$  压缩ASCII码字体B (9 × 17)

$0 \leq d1 \dots d(y \times xk) \leq 255$

[描述] 定义用户自定义字符

- y 指定纵向字节数。
- c1 是起始字符代码，c2是终止字符代码。
- x 指定横向点数。





## [注释]

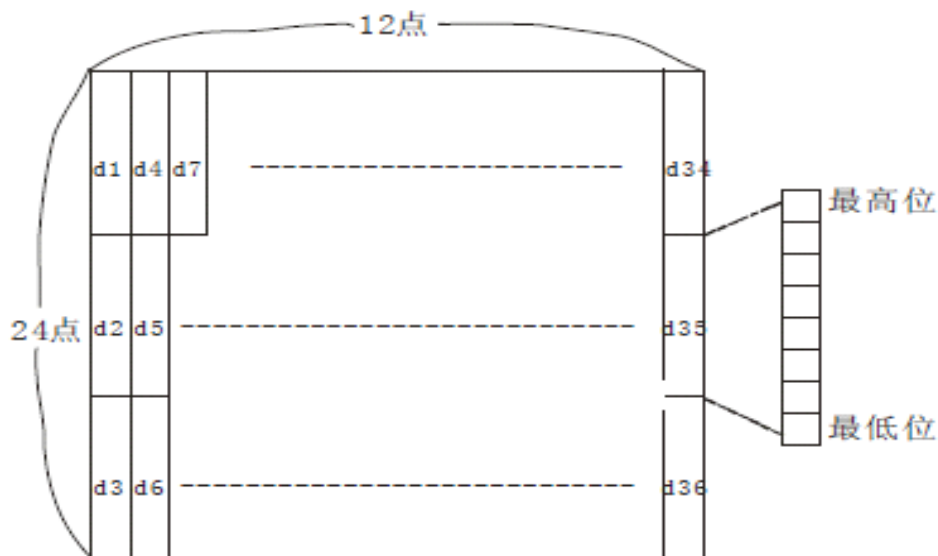
- 允许的字符代码范围是ASCII码的<20>H 到<7F>H (96个字符).
- 可以连续定义多个字符, 如果只定义一个字符 $c1 = c2$ .
- d是下载字符的数据. 各点的数据从左边开始.
- 自定义字符的大小是 $(y \times x)$  字节.
- 数据的各个位为1表示打印这个点, 为0表示不打印.
- 当下列情况, 用户自定义字符被清除:
  - a. ESC @ 被执行.
  - b. ESC ? 被执行.
  - c. FS q 被执行.
  - d. GS \* 被执行.
  - e. 打印机复位或电源关闭.
- 当用户自定义字符在字体B ( $9 \times 17$ )中定义时, 仅垂直方向数据的第三字节的最高有效位有效.

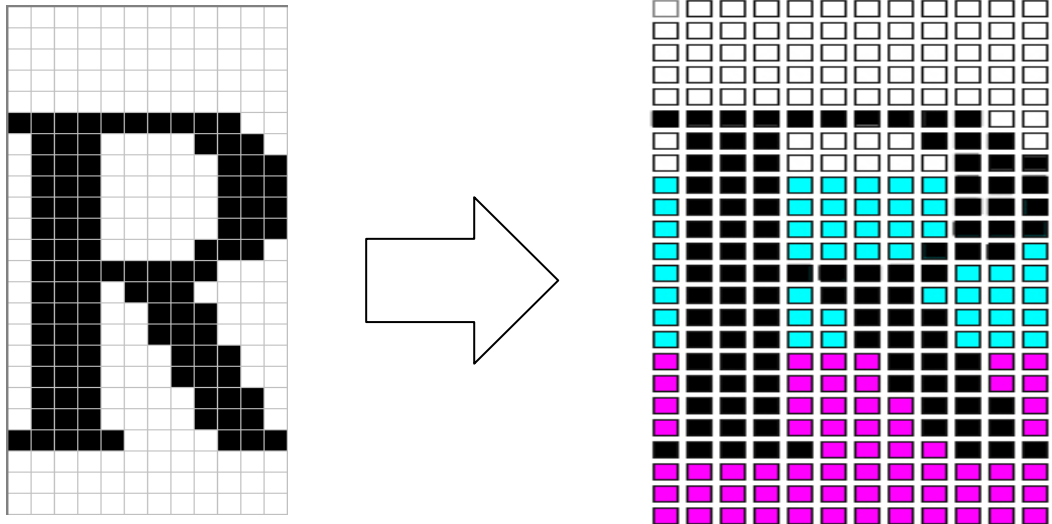
[默认值] 内部字库设置

[参考] ESC %, ESC ?

## [例子]

- 当选择标准ASCII 码字体( $12 \times 24$ ) 时

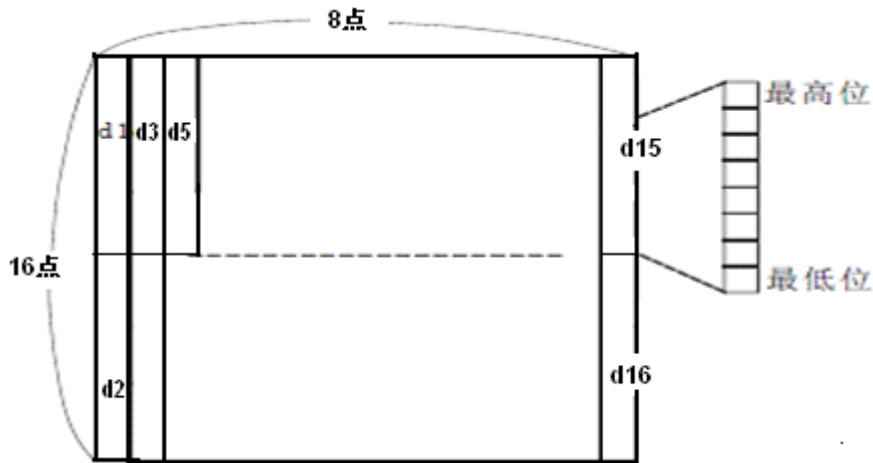




字模数据如下:

d1=0x04, d4=0x07, d7=0x07, d10=0x07, d13=0x04, d16=0x04, d19=0x04, d22=0x04, d25=0x06, d28=0x07, d31=0x03, d34=0x01  
 d2=0x00, d5=0xFF, d8=0xFF, d11=0xFF, d14=0x08, d17=0x0C, d20=0x0F, d23=0x0F, d26=0x1B, d29=0xF0, d32=0xF0, d35=0xE0  
 d3=0x08, d6=0xF8, d9=0xF8, d12=0xF8, d15=0x08, d18=0x00, d21=0x00, d24=0xC0, d27=0xF0, d30=0xF8, d33=0x38, d36=0x08

•当选择压缩 ASCII 码字体(9×17) 时



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

[名称] 定义Flash 位图

[格式] ASCII码 FS q n [ xL xH yL yH d1...dk]... [ xL xH yL yH d1... dk]

[HTTP://WWW.RD-CN.COM](http://www.rd-cn.com)



十进制码 28 113 n [xL xH yL yH d1...dk]... [ xL xH yL yH d1...dk]

十六进制码 1C 71 n [xL xH yL yH d1...dk]... [ xL xH yL yH d1...dk]

[范围]  $1 \leq n \leq 255$

$0 \leq xL \leq 255$

$1 \leq (xL + xH \times 256) \leq 1023$

$1 \leq (yL + yH \times 256) \leq 288$

$0 \leq d \leq 255$

$k = (xL + xH \times 256) \times (yL + yH \times 256) \times 8$

Flash下载容量最大为8096字节

[描述] 定义Flash 位图:

- n 指定所要定义的Flash位图的数目。
- xL、xH指定Flash 位图的横向点数  $(xL + xH \times 256) \times 8$ 。
- yL、yH指定Flash 位图的纵向点数  $(yL + yH \times 256) \times 8$ 。

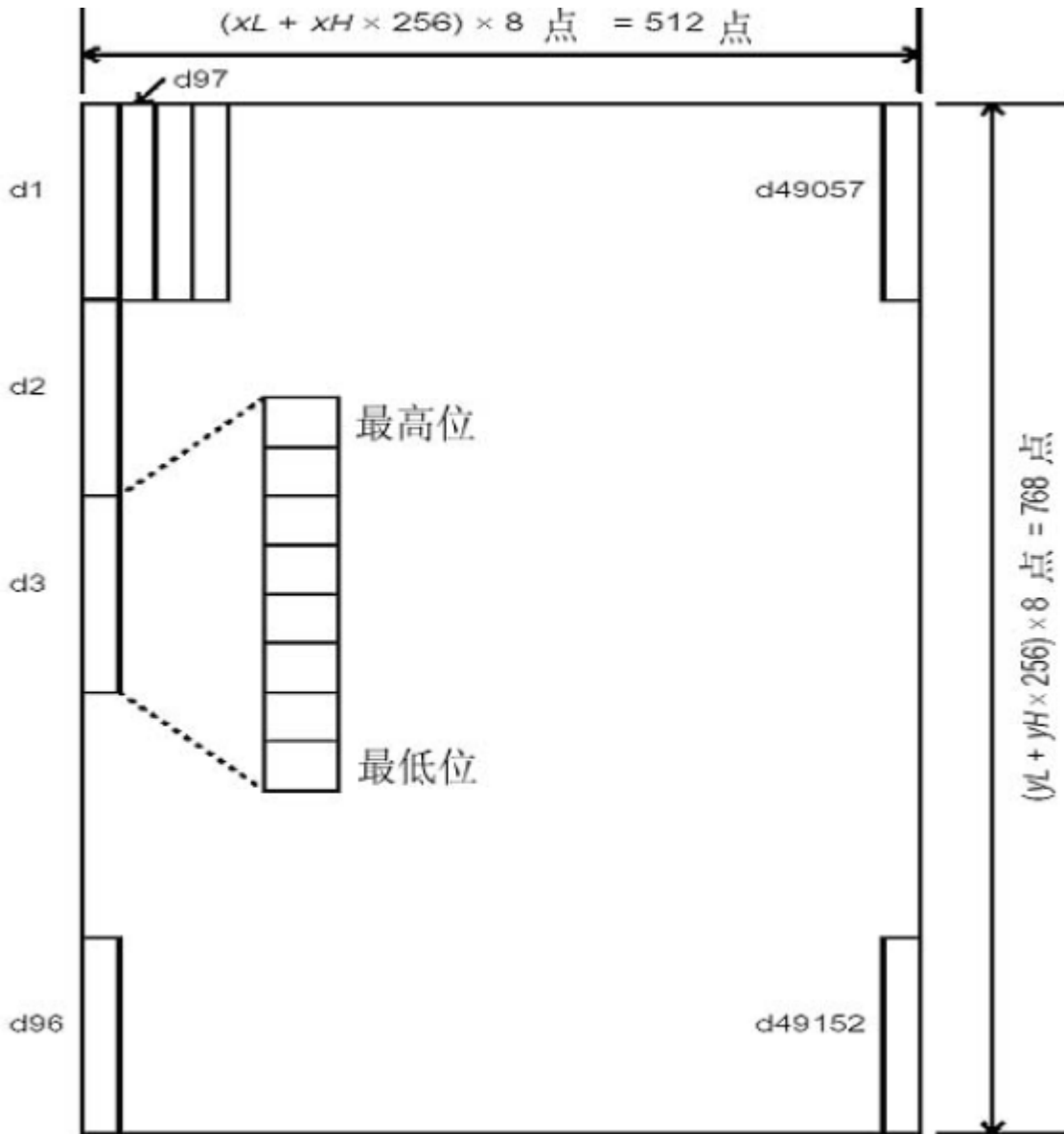
[注释]

- 频繁执行该命令能损坏Flash 存储器, 推荐每天最多写Flash 10次。
- 该命令删除所有的以前由该命令定义的Flash 位图, 打印机不能对上一次定义的多幅位图中的一个进行重新定义, 在这种情况下, 必须重新发送所有的数据。
- 由于在处理该命令的过程中, 打印机处于忙状态, 它向Flash 中写数据并且停止接收其他命令, 因此, 在该命令的执行过程中, 禁止向打印机发送其他命令, 包括实时命令。
- Flash位图是由命令FS q定义存储在Flash存储器中并用命令FS p打印的位图。
- 在标准模式下, 该命令仅在行首时有效。
- 从FS 到yH 的七个字节数据作为命令数据处理, 不是图形数据的一部分。
- 当位图数据字节数超出由它左边的xL、xH、yL、yH定义的范围, 则打印机只处理xL、xH、yL、yH定义的范围的数据。
- 在第一组Flash位图里, 当xL, xH, yL, yH中任何一个参数超出定义的范围时, 该命令无效。
- 在下载多幅位图时, 如果打印机处理xL、xH、yL、yH超出定义的范围, 则打印机停止执行这条命令。命令中在此之后的位图无效, 在此之前的位图有效。
- d是定义的位图数据, 在数据中相应位为1表示打印该点, 为0表示不打印。
- 这条命令定义了n 幅Flash位图。每个位图的序列号从1 依次增加, 因此, 第一个数据组[xL xH yL yH d1...dk] 是Flash位图1 的数据, 最后一个数据组[xL xH yL yH d1...dk] 是Flash位图n 的数据。在用FS p 命令打印位图时, 也是如此。
- 定义一幅Flash位图的数据由[xL xH yL yH d1...dk] 组成。因此, 当只有一幅位图时,  $n = 1$  打印机占用Flash存储器字节数如下: [位图数据字节数:  $(xL + xH \times 256) \times (yL + yH \times 256) \times 8$ ] + [头信息: 4]
- 在打印机里Flash下载空间最大为64K bits ( 8K bytes), 该命令可以定义多幅Flash位图, 但不能定义一幅大小超过64K bits的位图(不同的打印机下载空间也不相同, 请参照打印机配置信息)。
- 在写Flash 之前, 打印机立即处于忙状态。
- 在处理该命令的过程中, 打印机不传送状态也不执行状态查询。
- 在宏定义的过程中接受到该命令, 打印机将结束宏定义, 开始执行该命令。
- 如果一幅Flash位图被定义, 执行ESC @命令、复位和关闭电源不能将其擦除。
- 该命令只是定义Flash位图, 不执行打印, 打印Flash位图由FS p命令执行。



[参考] FS p<sub>\_\_</sub>

[例子] 当xL = 64, xH = 0, yL = 96, yH = 0



**FS p n m**

[名称] 打印下载到FLASH 中的位图

[格式] ASCII码 FS p n m  
 十进制码 28 112 n m  
 十六进制码 1C 70 n m

[范围]  $1 \leq n \leq 255$   $0 \leq m \leq 3$ ,  $48 \leq m \leq 51$

[描述] 以m指定的模式打印下载到FLASH中的位图。

m 模式纵向分辨率(DPI) 横向分辨率(DPI)  
 0, 48 正常200 200  
 1, 49 倍宽200 100



- 2, 50 倍高100 200
- 3, 51 倍宽、倍高100 100
  - n表示位图（由命令FS q定义）的图号。
- m指定打印位图的模式。

[注释]

- Flash 位图是由命令FS q 定义存储在Flash 存储器中并用命令FS p 打印的位图。
- 当Flash 位图没有被定义时，该命令无效。
- 该命令只有在打印缓冲区里没有数据时有效。
- 该命令除了受倒置打印模式的影响外，不受其他打印模式的影响（如：加粗打印、重叠打印、倍高、倍宽、下划线、字符放大、反白打印、顺时针旋转90度等）。
- 如果所要打印的下载位图超出当前打印区域，则不打印超出的部分。
- 在普通和倍宽模式下，该命令进纸n 点，n为NV 位图高度，在倍高和四倍大小模式下，该命令进纸  $n \times 2$  点，n为NV 位图高度，与ESC 2 或ESC 3设定的行间距无关。
- 打印完位图后，打印机换行，按普通模式处理其后的数据。

[参考] ESC \*, FS q, GS /, GS v 0

### GS \* x y d1...d(x × y × 8)

[名称] 定义下载位图

[格式] ASCII码      GS \* x y d1...d(x × y × 8)  
 十进制码      29 42 x y d1...d(x × y × 8)  
 十六进制码    1D 2A x y d1...d(x × y × 8)

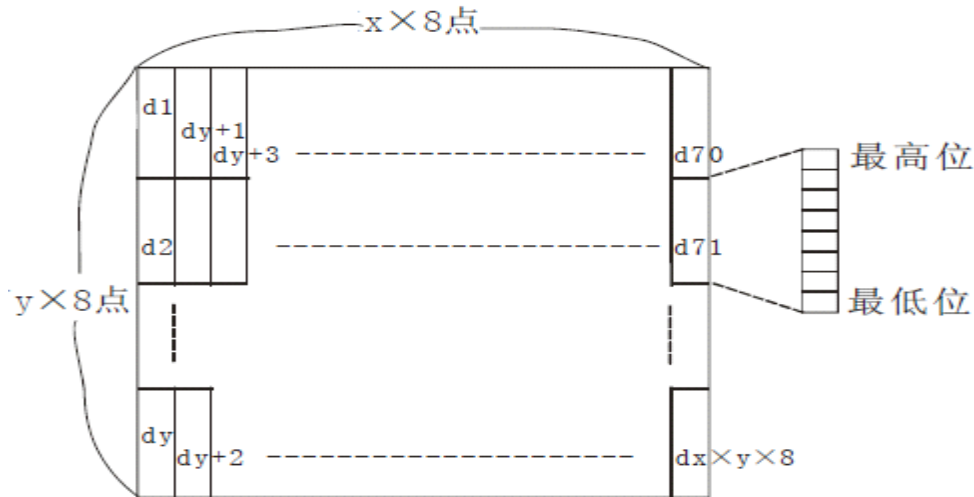
[范围]  $1 \leq x \leq 255$ ,  $1 \leq y \leq 48$   
 $x \times y \leq 912$   
 $0 \leq d \leq 255$

[描述] 定义一个下载位图，其点数由x 和y 指定

- x 指定位图的横向点数；
- y 指定位图的纵向点数。

[注释]

- 位图横向上的点数为  $x \times 8$ ；位图纵向上的点数为  $y \times 8$ 。
- 如果  $x \times y$  超出规定的范围，则这条命令无效。
- d为位图数据。数据对应位为1表示打印该点，为0表示不打印。
- 下载的位图在下列情况下被清除：
  - a. 执行ESC @。
  - b. 执行ESC &。
  - c. 执行FS q。
  - e. 打印机复位或关闭电源。
- 打印数据和下载位图的关系如下图所示：



[参考] GS /

GS / m

[名称] 打印下载位图

[格式] ASCII码 GS / m  
 十进制码 29 47 m  
 十六进制码 1D 2F m

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

[描述] 打印一幅下载位图，打印模式由m 指定

m 选择打印模式如下：

m	模式	纵向分辨率	横向分辨率
0, 48	正常	200	200
1, 49	倍宽	200	100
2, 50	倍高	100	200
3, 51	倍宽、倍高	100	100

[注释]

- 如果下载的位图没有被定义，这条命令被忽略。
- 只有打印缓冲区没有数据时，此命令才有效。；
- 除倒置打印模式外，其它打印模式对该命令无效（包括加粗、双重打印、下划线、字体放大以及反白打印等）。
- 如果下载的位图超出了打印区域，则超出的部分不打印。
- 该命令打印位图为下载到RAM中的位图，不是下载到FLASH中的位图，对应的图号为GS \* 命令设置的图号。

[参考] GS \*



## 附录

## A 打印字符集

本打印字符集 0x80 及之后的编码为取消汉字打印模式下打印出的字符。有关汉字字符，请参见国标 GB-2312

和微软代码页 CP936。

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	â	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	ç	9C	£	9D	¥	9E	Pts	9F	f
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	Υ
E8	Φ	E9	θ	EA	Ω	EB	δ	EC	∞	ED	φ	EE	€	EF	∩
F0	≡	F1	±	F2	≥	F3	≤	F4		F5		F6	÷	F7	≈
F8	°	F9	•	FA	.	FB	√	FC	∞	FD	²	FE	▪	FF	







### E. 国际标准 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	{		}	~	Δ
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	ƒ	ô	ö	ò	û	ü	ö	ü	ç	£	¥	℞	f	
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	¬	½	¼	↓	«	»	
B	⋮	⋮	⋮													
C	L	l	T	t	-	†	‡	£	¤	¥	¦	§	¨	©	ª	«
D	ª	¬	½	¼	↓	«	»	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
E	α	β	γ	π	Σ	σ	μ	γ	ϑ	θ	Ω	δ	ω	φ	ε	Π
F	≡	±	≥	≤	∫	J	÷	≈	°	.	.	√	n	2	■	