

目 录

一、概 述	3
二、打印机状态与操作说明	4
2.1 外观与尺寸	4
2.2 打印机状态说明	4
2.3 开关机方法	4
2.4 安装纸卷	4
2.5 自检测方法	5
2.6 打印机充电	5
三、打印机接口	5
3.1 串行接口	5
3.2 蓝牙接口	6
3.3 USB 接口	6
四、指令系统	7
4.1 指令表	7
4.2 指令详解	8
ESC @	8
LF	9
CR	9
FF	9
ESC J	10
ESC d	10
ESC c	11
HT	11
ESC D n1 n2 ... nk NULL	12
ESC - n	13
ESC + n	13
GS B n	14
ESC !	14
FS U nL nH	15
FS 2 n	16
ESC \$ nL nH	17
ESC I n	17
ESC Q n	18
ESC 1 n	18
ESC SP n	19
ESC a n	19
FS r n	20

ESC U n	20
ESC V n	21
ESC X	21
ESC K nL nH d1 d2dk	22
ESC * m nL nH d1...dk	22
GS * x y d1...dk	29
GS / n	30
FS q n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n	30
FS p n m	31
EGS h n	32
GS w n	32
GS H n	33
GS Q n	34
GS k	34
GS k m v r d1.....dk	36
ESC ‘	37
ESC v	39
FS &	40
FS.	40
ESC 6	40
ESC 7	41
附录 1: 国际标准 ASCII 码及 ANK 字符集 1、2	42
1.1 国际标准 ASCII 码	42
1.2 ANK 字符集 1、2	43
附录 2: 条码	43
2.1 条码编码规则	43
2.2 条码长度字符集表	44

一、概述

RD-V80 型便携式热敏打印机是一款专为移动票据打印而设计的热敏便携式微型打印机。本机内置可充锂电池，具有体积小、待机时间长、打印速度快、字迹清晰等特点。

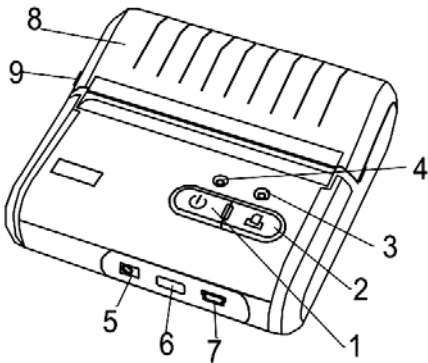
本机标准接口为串行接口，可配置蓝牙无线数据接口/USB 接口，可与各种手持设备如数据采集器、PDA、手机等连接打印。采用蛤壳式装纸方式，支持自动休眠、自动唤醒同时具有声光提示功能。

其技术指标如下：

型号	RD-V80-SN	RD-V80-USB	RD-V80-BL
打印方式	行式热敏打印		
纸张规格	80mm 宽热敏打印纸		
纸卷直径	纸卷直径 $\leq\Phi 33\text{mm}$		
分辨率	(72mm) 576docs/line, 分辨率 203DPI (8dots/mm)		
打印行宽	24 汉字/行、48 英文字符/行		
汉字编码	GBK		
字库使用	字符集 1,2(6x8) ASCII 字库 (12x24)、GBK (24x24) ASCII 字库 (8x16)、GBK (16x16)		
打印速度	50mm/s		
打印内容	英文、数字、各种符号、汉字、图形、曲线、条码等		
电源	700mAh、7.4V 可充电锂电池		
充电方式	带机充电		
工作时间	12.5%打印密度下，可打印 30~100m		
机芯寿命	50km 或 1 亿脉冲/doc (25%打印密度或以下)		
接口提供	RS-232 标准接口	USB 接口	蓝牙接口
指令集	ESC/POS 兼容指令集，并可根据客户需求增加其他专用命令。		
充电要求	快速充电 3 小时		
异常检测	缺纸检测、电量不足检测、过热保护		
切纸方式	手动撕纸		
装纸方式	蛤壳式装纸		
工作环境	温度：-20℃~50℃，湿度：20%~80%		
软件支持	WINXP、WIN2000、CE、windows mobile		
体积	101mmx100mmx39mm		
重量	约 220g (含一块电池、不含打印纸)		

二、打印机状态与操作说明

2.1 外观与尺寸



尺寸(长×宽×高):		101mm×100mm×39mm
打印机部件名称		
1.电源键(P)	1.进纸键(F)	3.功能指示灯
4.充电指示灯	5.充电插孔	6.红外窗口
7.数据接口	8.纸仓盖	9.开盖突起

2.2 打印机状态说明

打印机在开机时，蜂鸣器将会响 3 声，以提示开机；当充电适配器连接打印机后，打印机的蜂鸣器将会发出短促的音乐声以提示充电适配器已连机。

打印机在没有收到数据时，打印机自动休眠，收到数据后自动唤醒。

打印机在开机状态下 10 分钟内不打印数据，打印机自动关机。

具体状态如下表：

状态 供电	电池	
	指示灯	蜂鸣器
待机状态	右绿(蓝)色指示灯闪烁	--
打印状态	右绿(蓝)色指示灯长亮	--
缺纸状态	右红色指示灯闪烁	2 声/2s
电量不足	右红色指示灯闪烁	1 声/2s
充电中	左红色指示灯长亮	--
充电完成	左绿(蓝)色指示灯长亮	--

注:提示状态蜂鸣器状态指示提示 3 次,3 次以后不再提示.

2.3 开关机方法

P 键为电源按键，长按听到提示音后为开机，再按一下为关机。当打印机在 10 分钟不打印数据则自动关机（该时间可根据客户要求作调整），使用时须重新才能开机打印。

2.4 安装纸卷

打印机为易上纸结构，用手指捏住纸仓盖两侧开盖突起将纸仓盖打开，将热敏纸卷放入(光滑面朝下)，

露出纸头，合上纸仓盖，完成装纸。

F 键为走纸按键，在开机状态下，按住 F 键，打印机进行走纸，松开停止走纸。

2.5 自检测方法

在关机状态下，按住 F 键，再按开机 P 键，约 3 秒，然后松开按键，开始自检测打印，此时会打印出打印机的型号，通讯方式，生产厂商等信息。

2.6 打印机充电

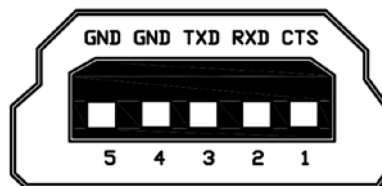
打印机无论处于开机或者关机状态，插入充电适配器时，打印机自动进入充电模式。

三、打印机接口

3.1 串行接口

RD-V80 型便携式热敏打印机采用异步串口方式，接口电平为 RS232 电平，通讯的波特率为 9600；8 位数据位，无奇偶效验位；握手方式：CTS。

插座为 MINI_USB 接口座，引脚定义如下图：



打印机的数据缓存为 8K 字节，当发送数据小于 8K 字节时，可不用流控方式，引脚的具体功能如下表：

Mini usb 孔座	信号	信号来源	方向	说明
1	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。
2	RXD	打印机	输出	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。
3	TXD	主机	输入	打印机从主计算机接收数据
4	GND	——	——	信号地。
5	GND	——	——	信号地。

RS232 接口打印机的编程操作如下：

连接打印机接口——打印机开机——初始化上位机串口——向串口发送数据

具体步骤为：

- 1) 将打印机的接口与上位机的串口连接好，注意串口的电平，应为 **RS232** 电平。
- 2) 确定打印纸已装好，按下 P 键，打开打印机。
- 3) 将上位机的串口打开，通迅速率设置成与打印机的通迅方式相同。正常为 **9600,8,N**，如不能确定通迅的具体参数，通过自检测打印机，在自检条上有详细的通迅参数说明，按此参数设置。
- 4) 向串口发送数据，如打印 **RONGDA**，则向串口发送 **R O N G D A** 的 **A S C I I** 码，十六进制表示则为：**52H 4FH 4EH 47H 44H 41H 0D**。

3.2 蓝牙接口

蓝牙是一种支持设备短距离通信（一般是 10m 之内）的无线电技术。能在包括移动电话、PDA、无线耳机、笔记本电脑、相关外设等众多设备之间进行无线信息交换。蓝牙的标准是 **IEEE802.15**，工作在 **2.4GHz** 频带，带宽为 **1Mb/s**。

使用蓝牙接口打印前需与驱动 **RD-V80** 便携式打印机的主机配对，配对过程由主机发起。

设置法如下：

- 1) 打印机开机，
- 2) 主设备搜寻外部蓝牙设备，
- 3) 如果有多台外部蓝牙设备的话，选中 **RDV80BL** 打印机
- 4) 输入密码“**0000**”
- 5) 完成配对。

配对完成后，用户根据当前蓝牙设备在主机中映射的端口来操作打印机。

如使用笔记本电脑、**SMARTPHONE** 的手机、**POCKET PC**、**PALM** 等有虚拟蓝牙串口的宿主，可以通过虚拟蓝牙串口向 **RD-V80** 便携式打印机发送打印数据进行打印。如果宿主没有虚拟蓝牙串口，本公司可提供宿主蓝牙模块配件。

3.3 USB接口

USB 是一个外部总线标准，一般用于 **PC** 机上，用于规范电脑与外部设备的连接和通讯。**USB** 接口支持设备的即插即用和热插拔功能。

打印机的 **USB** 接口的实现是通过一个 **USB** 接口芯片来实现的，安装相应的驱动之后，则会在 **PC** 机上虚拟出一个串口来。操作 **USB** 接口打印机与操作 **RS232** 接口打印机是一样的，详见 **3.1** 节。

四、指令系统

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、条型码打印等功能。

4.1 指令表

命令	功能
ESC @	初始化打印机
LF	打印并换行
CR	打印并回车
FF	打印并走纸到下页首
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 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
ESC !	选择打印模式
FS U	按Unicode编码向打印发送数据
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 *	定义下载位图
GS /	打印下载位图
FS q	上传多个NV位图

FS p	打印下载NV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
GS k	打印QRCODE二维条码
ESC ‘	打印曲线
ESC v	向主机传送打印机状态
FS &	进入汉字模式
FS .	取消汉字模式
ESC 6	选择6X8字符集1
ESC 7	选择6X8字符集2

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。

4.2 指令详解

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] unsigned char str[2];

 str[0] = 0x1B;

 str[1] = 0x40;

 SendDataToPrinter(str,2);

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] unsigned char str[2];

 str[0] = 0x0A;//或str[0] = '\n'

 SendDataToPrinter(str,1);

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] unsigned char str[2];

 str[0] = 0x0D;//或str[0] = '\r'

 SendDataToPrinter(str,1);

FF

[名称] 打印并走纸到下页首

- [格式] ASCII FF
 十进制 12
 十六进制 0C
- [说明] 将打印缓冲区中的数据全部打印出来并返回标准模式。
- [注意] • 如果打印纸有预印刷黑标，则打印缓冲区中的数据后，走纸到黑标处，如果打印纸无黑标，则走纸25cm后停止，预印刷黑标的规范请见附录3.预印刷黑标说明。
- [例子] unsigned char str[2];
 str[0] = 0x0C;
 SendDataToPrinter(str,1);

ESC J

- [名称] 打印并进纸
- [格式] ASCII ESC J *n*
 十进制 27 74 *n*
 十六进制 1B 4A *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[$n \times 0.125\text{m m}$ (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] unsigned char str[3];
 str[0] = 0x1B;
 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 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);`//允许反向打印,打印方向由左向右。

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 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。

[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>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7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 (&Order,1);  
SendDataToPrinter ("HT1",3);  
SendDataToPrinter (&Order,1);  
SendDataToPrinter ("HT2",3);  
SendDataToPrinter (&Order,1);  
SendDataToPrinter ("HT3",3);  
Order = 0x0D;  
SendDataToPrinter (&Order,1);  
SendDataToPrinter ("1234567890123456\r",17)
```

```
HT1   HT2  HT3  
1234567890123456
```

ESC - n

[名称] 取消/设置下划线

[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

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);

ESC !

[名称] 选择打印模式

[格式] ASCII ESC ! *n*

十进制	27	33	<i>n</i>
十六进制	1B	21	<i>n</i>

[范围] $0 \leq n \leq 255$

位	值	意义
0	0	西文字符（半宽）字体A(12×24)，汉字字符（全宽）字体A(24×24)
	1	西文字符（半宽）字体B(8×16)，汉字字符（全宽）字体B(16×16)
1	-	-
2	-	-
3	-	-
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。

FS U nL nH

[名称] 按Unicode编码向打印发送数据

[格式] ASCII FS U nL nH

十进制 28 85 nL nH

十六进制 1C 55 nL nH

[描述] 打印 $n(n=nL+nH*256)$ 个Unicode编码字符。

[注意] •Unicode是双字节编码，此命令后的 $2*n(n=nL+nH*256)$ 个字节被当作Unicode字符处理；

•该指令中输入汉字只支持GBK字库中包含汉字，没有包含Unicode中所包含的所有汉字；

•内码发送方式为高八位在前，低八位在后。

[例子]

`char str[64];`

`str[0]=0x1C; str[1]=0x55; str[2]=11; str[3]=0;`

`str[4]=0x00;str[5]= 0x55; //'U'`

`str[6]= 0x00;str[7]= 0x4E; //'N'`

```
str[8]= 0x00;str[9]= 0x49; //'I'  
str[10]= 0x00;str[11]= 0x43; //'C'  
str[12]= 0x00;str[13]= 0x4F; //'O'  
str[14]= 0x00;str[15]= 0x44; //'D'  
str[16]= 0x00;str[17]= 0x45; //'E'  
str[18]= 0x62;str[19]= 0x53; //'打'  
str[20]= 0x53;str[21]= 0x70; //'印'  
str[22]= 0x6D;str[23]= 0x4B; //'测'  
str[24]= 0x8B;str[25]= 0xD5; //'试'  
SendDataToPrinter (str,26);
```

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] $0 \leq n \leq 3$

[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度旋转  
  
          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， $N = nL + nH \square 256$

[注意] • 如果设定的打印位置超出了可打印区域（ $N > 384$ ），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 ESC I 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。

<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];

```
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] $0 \leq n \leq 1$
- [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[4];
 str[0] = 0x1C;
 str[1] = 0x72;
 str[2] = 0;
 SendDataToPrinter(str,4);//设置水平放大2倍.

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] $1 \leq n \leq 8$
- [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 ESC V *n*
 十进制 27 86 *n*
 十六进制 1B 56 *n*

[范围] $1 \leq n \leq 8$

[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] = 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 ESC X *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 K nL nH d1 d2dk

[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 ≤ nL ≤ 255

0 ≤ nH ≤ 1

0 ≤ d ≤ 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++] = 0x0f; //打印15个点宽图形nL
```

```
str[i++] = 0x00; //nH
```

```
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

$0 \leq nL \leq 255$

$0 \leq nH \leq 1$

$0 \leq d \leq 255$

[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 576，在双倍宽时其值最大为 288。

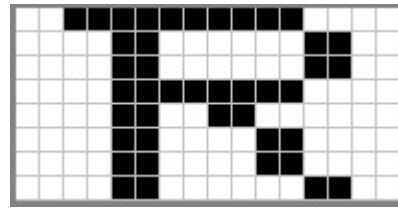
d1.....dk 表示位图数据：具体格式见下图：

[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
```

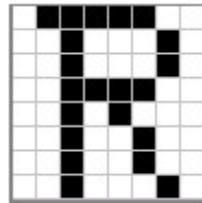
```
str [j++] = 0x00;str [j++] = 0x80;str [j++] = 0xFF;str [j++] = 0x90;str [j++] = 0x98;
```

```
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

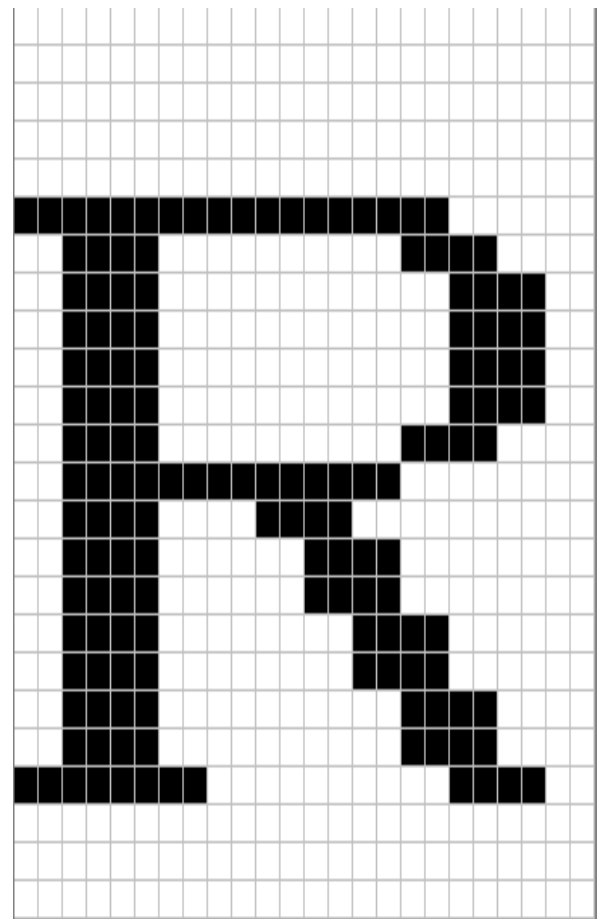
str[j++] = 0x00;str[j++] = 0x80;str[j++] = 0xFF;str[j++] = 0x90;str[j++] = 0x98;

str[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; b_n 表示字节的第 n 位

	d4	d7										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
d2	0	1	1	0	0	0	0	0	0	1	1	b7
	0	1	1	0	0	0	0	0	1	1	0	b6
	0	1	1	1	1	1	1	0	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
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=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;

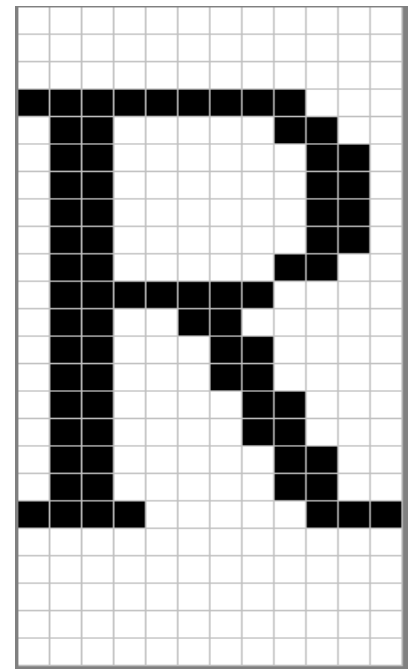
str[j++] = 0x07;str[j++] = 0x80;str[j++] = 0x20;str[j++] = 0x00;str[j++] = 0x00;str[j++] = 0x20;

str[j++] = 0x0D; //打印出当前的图形

SendDataToPrinter(str,j);
```

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； b_n 表示字节的第 n 位

	d4	d7										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
d2	0	1	1	0	0	0	0	0	0	1	1	b0
	0	1	1	0	0	0	0	0	0	1	1	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
d3	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
	0	1	1	0	0	0	0	0	1	1	0	b6
	1	1	1	1	0	0	0	0	0	1	1	b5
	0	0	0	0	0	0	0	0	0	0	0	b4
	0	0	0	0	0	0	0	0	0	0	0	b3
	0	0	0	0	0	0	0	0	0	0	b2	
	0	0	0	0	0	0	0	0	0	0	b1	
	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 * x y d1...dk

[名称] 定义下传位图

[格式] ASCII码 GS * x y d1...dk

十进制码 29 42 x y d1...dk

十六进制码 1D 2A x y d1...dk

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

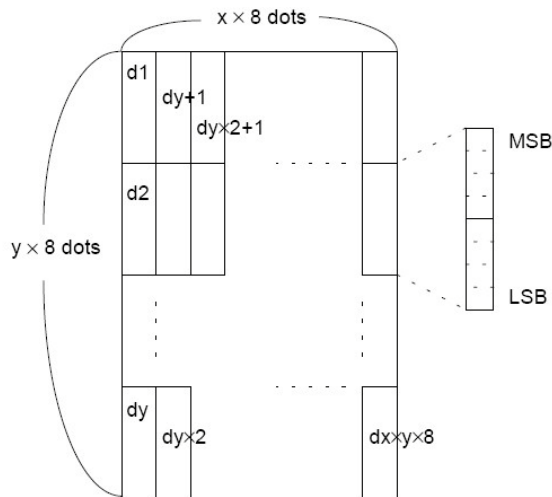
[描述] 用x 和y 指定的点数定义下传位图。

• $x*8$ 为水平方向点数。• $y*8$ 为垂直方向点数。

[注意] • 由于缓冲区的限制，如果 $x \times y$ 超出了指定范围，则该命令可能出现无法预料的结果。

• d 表示位图数据。数据d1,d2...dn 指定打印位为1，不打印位为0。

• 用该命令定义的下传位图，通过 **GS / n** 命令打印



GS / n

[名称] 打印下传位图

[格式] ASCII码 GS / n

十进制码 29 47 n

十六进制码 1D 2F n

[范围] $0 \leq n \leq 3$

[描述] 用n 指定的模式打印由GS *命令定义的下传位图。

n从下表选择模式:

n	放大模式
0	普通
1	倍宽
2	倍高
3	倍宽倍高

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

[名称] 下传多个NV位图

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

十进制码 28 113 n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n

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

[范围]: $1 \leq n \leq 255$

$0 \leq xL \leq 255$

$0 \leq xH \leq 3$ (when $1 \leq (xL + xH * 256) \leq 1023$)

$0 \leq yL \leq 255$

$0 \leq yH \leq 1$ (when $1 \leq (yL + yH * 256) \leq 288$)

$0 \leq d \leq 255$

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

总共定义图形数据150K 字节

[描述]: • n指定图形下载个数。

• xL, xH指示位图水平宽度为 $(xL + xH * 256) * 8$ 点。

• yL, yH指示位图垂直高度为 $(yL + yH * 256) * 8$ 点。

- d 为图形数据。

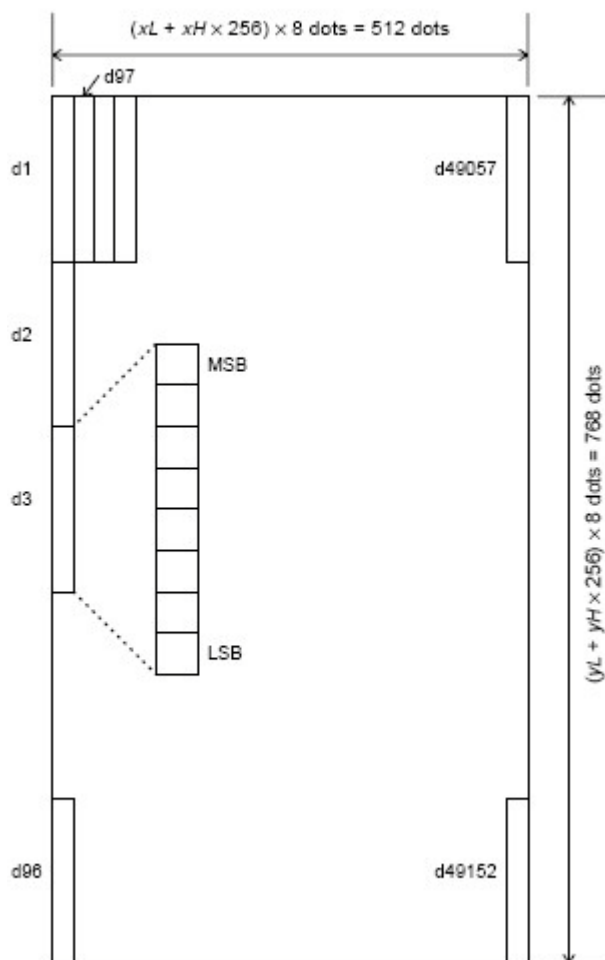
[注意]:

- 图形水平垂直方向点数为8的倍数。

- 用该命令定义的下传位图, 通过 **FS p n m** 命令打印

[例子]:

当 $xL = 64, xH = 0, yL = 96, yH = 0$



FS p n m

[名称] 打印下传NV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] $1 \leq n \leq 255$

[描述] n 为FS q命令定义的下传位图编号。

m 从下表选择模式:

n	放大模式
0,48	普通
1,49	倍宽
2,50	倍高
3,51	倍宽倍高

EGS 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: $n = 48$

[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 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:

n	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 nl nh
 十进制 29 81 nl nh
 十六进制 1D 51 nl nh

[范围]: $0 \leq nl \leq 255, 0 \leq nh \leq 2$

[描述]: 设置条码（包括二维条码）从一行开始到打印位置的距离为N个水平点距。

[缺省值]: $nl=nh = 0$

[例子] unsigned char str[4];

 str[0] = 0x1D;

 str[1] = 0x51;

 str[2] = 32;

 str[3] = 0;

 SendDataToPrinter(str,4);//

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

*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）应当等于规定的数据长度，如果不等于规定的数据长度，则该指令无效。有关条码的数据位长度见【备注 1】。

• 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【备注 1】。

- 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【备注 1】。
- 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

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

[名称] 打印QRCODE二维条码

[格式] 该命令有两种格式：

格式1 m=32
ASCII码 GS k m v r d1...dk NUL
十进制码 29 107 m v r d1...dk 0
十六进制码 1D 6B m v r d1...dk 00

格式2 m=97
ASCII码 GS k m v r nL nH d1...dn
十进制码 29 107 m v r nL nH d1...dn
十六进制码 1D 6B m v r nL nH d1...dn

[范围] m=32或97
 $1 \leq v \leq 20$ $1 \leq r \leq 4$

[描述] v为DQCODE版本号

r=1 纠错等级为L r=2 纠错等级为M r=3 纠错等级为Q r=4 纠错等级为H

nL,nH为整数N的低位和高位，N为打印条码的数据长度，单位为字节

当使用第一种格式时，指令以00结尾，d1...dk为条码数据。当使用第二种格式时，打印机将把nH后的N个字符(d1...dn)作为条码数据。

[注意] • 因打印纸张宽度有限，QRCODE的版本号最大为20.

- 有关QRCODE的编码标准的详细信息，请参见中国国家标准GB/T 18284-2000或ISO标准ISO/IEC 18004:2000

[例子]

```
unsigned char str[16];  
str[0] = 0x1D; str[1] = 0x6B; str[2] = 32;  
str[3] = 1;//版本为1  
str[4] = 2;//纠错等级为M  
str[5] = '1'; str[6] = '2'; str[7] = '3'; str[8] = '4'; str[9] = '5'; str[10] = '6'; str[11] = '7';  
str[12] = '8'; str[13] = '9'; str[14] = '0';  
SendDataToPrinter ( str, 5);
```

ESC ‘

[名称] 打印一水平行上n个点

[格式] ASCII码 GS ‘ nL nH x1L x1H x21L x21H xkL xkH CR
十进制码 29 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

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

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

$Y3 = 50$

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

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

[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;

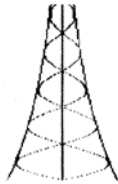
str[i++] = 0;

str[i++] = 0x0D;

SendDataToPrinter(str,i);//设置可识读字符在条码的下方打印。

}
    
```

[结果]:



ESC v

[名称] 向主机传送打印机状态

[格式] ASCII ESC v

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ASCII字符处理，将不再打印汉字，除非再用FS &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。

ESC 6

[名称] 选择6X8字符集1

[格式] ASCII ESC 6
 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6×8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 ESC 7
 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(见附录D), 字符集2中有6×8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。

附录1：国际标准ASCII码及ANK字符集1、2

1.1 国际标准ASCII码

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。

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	ø	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	π
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	π	D3	⊥	D4	⊥	D5	⊥	D6	π	D7	‡
D8	‡	D9	⊥	DA	⊥	DB	■	DC	■	DD	■	DE	■	DF	■
E0	α	E1	β	E2	γ	E3	Π	E4	Σ	E5	σ	E6	μ	E7	τ
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	ⁿ	FD	²	FE	■	FF	■

1.2 ANK字符集1、2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	↑	←
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8	0	-	二	三	四	五	六	七	八	九	十	元	角	分	月	日
9	€	¢	↓	→	^	±	÷	∞	∑	∏	∫	∂	∇	∞	∞	∞
A	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	π	ρ
B	σ	φ	ψ	ω	Γ	Δ	Π	Σ	Ψ	Ω	Θ	Φ	Ψ	Ω	Θ	Φ
C	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
D	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
E	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'
F	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	百	千	万	元	角	分	厘	毫	分	厘	毫	分	厘	毫	分	厘
3	#		U	Π	⊕	⊖	⊗	⊘	⊙	⊚	⊛	⊜	⊝	⊞	⊟	⊠
4	∴	≡	≅	≈	≠	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
5	*	⊗	()	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
6	⊗	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
7	夕	チ	ツ	テ	ト	ナ	ニ	ノ	ネ	ノ	ル	ロ	フ	ハ	ホ	マ
8	々	々	々	々	々	々	々	々	々	々	々	々	々	々	々	々
9	ン	ア	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ	タ
A	И	Й	П	И	У	Ш	Щ	Ъ	Ы	Э	Ю	Я	Б	В	Г	Д
B	Ф	Г	У	Е	Ж	З	А	С	Е	Е	И	И	И	И	И	И
C	А	Е	Ж	З	А	С	Е	Е	И	И	И	И	И	И	И	И
D	і	ó	ú	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ
E	ē	ē	ē	ī	ī	ī	ī	ī	ī	ī	ī	ī	ī	ī	ī	ī
F	ÿ	ö	ü	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø	ø

附录2：条码

2.1 条码编码规则

UPC-A: UPC-A 编码要符合 UCC 组织(<http://www.ucinet.org>)的规范。

UPC-E: UPC-E 编码要符合 UCC 组织(<http://www.ucinet.org>)的规范。

ENA8: ENA8 编码要符合 EAN 组织(<http://www.ean-int.org>)的规范。

ENA13: ENA13 编码要符合 EAN(<http://www.ean-int.org>)组织的规范。

CODE39: 又称 39 码, CODE39 的起始位字符和终止位字符必须为**，且起始位和终止位之间不能包含字符**，本打印机*由打印机自动给出，编程时不用给出，数据中可包含校验码也可不包含校验码，校验码有固定算法。

ITF: 又称INTERLEAVED 25, 交叉 25 码, INTERLEAVED 2 of 5, 数据位长度只能为偶数，数据中可包含校验码也可不包含校验码，校验码有固定算法。

CODABAR: 又称库德巴码, 起始位和终止位必须为 A、B、C、D 四个字符中的一个，起始位字符终止位字符不必相同，数据中可包含校验码也可不包含校验码，校验码由编码人自定义。

CODE93: CODE93 的起始位字符和终止位字符必须为**，且起始位和终止位

之间不能包含字符**，本打印机*由打印机自动给出，编程时不用给出，CODE93数据最后必须包含两个字符的校验码，校验码有固定算法。

2.2 条码长度字符集表

条码类型	长度	字符集(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	偶数52	0~9
CODABAR	32	0~9 - : / % . A~D
CODE93	无限制	0~9 A~Z - . SP \$ / + % *
CODE128	33	

附录3：黑标

3.1 预印刷黑标说明

如果用户使用预印刷黑标进行票据定位，印刷黑标时务必遵守以下黑标预印刷规范，否则可能导致打印机无法识别黑标。

黑标预印刷规范：

印刷位置：如左图所示，黑标应印刷于文字面的右侧边缘。

宽度范围：宽度 $\geq 7\text{mm}$

高度范围： $4\text{mm} \leq \text{度} \leq 6\text{mm}$

对红外光的反射率： $< 10\%$ （纸张黑标宽度其他部分对于红外光的反射率 $> 65\%$ ）

Hps: Hps 为打印机黑标下边缘距打印起始上边缘的距离。

$$0\text{mm} \leq \text{Hps} \leq 1\text{mm}$$

