



热敏打印机 指令手册

版权所有：北京荣达创新科技有限公司

日期	版本	备注
2020-03-02	V2.1	增加QR码混打指令
2022-06-29	V2.2	增加一维码混打指令
2024-11-20	V2.2	修改加深及下载指令

注：本指令手册适用于带有标签功能打印机



目录

1.1 指令表.....	4
1.2 指令详解.....	6
1.2.1 控制指令.....	6
ESC @.....	6
FF.....	7
LF.....	7
CR.....	7
ESC J.....	8
ESC d.....	8
ESC =.....	9
ESC c.....	9
HT.....	9
ESCD n1 n2 ... nk NULL.....	10
ESC - n.....	11
ESC . n.....	12
GS B n.....	12
FS 2 n.....	13
ESC \$ nL nh.....	13
ESC l n.....	14
ESC Q n.....	14
ESC 1 n.....	15
ESC SP n.....	15
ESC a n.....	16
FS r n.....	16
1.2.2 缩放指令.....	17
ESC U n.....	17
ESC V n.....	17
ESC X.....	18
1.2.3 图形指令.....	18
ESCKnL nH d1 d2dk.....	18
ESC* m nL nH d1...dk.....	19
GS v 0 m xL xH yL yH d1...dk.....	25
1.2.4 条码指令.....	26
EGS h n.....	26
GS w n.....	27
GS H n.....	27
GS Q n.....	28
GS k.....	28
DC2 k xL xH yL yH.....	31
DC2 j n.....	31
DC2 n.....	31



GS k m v r d1.....dk.....	33
GS l xL xH yL yH.....	34
GS n m v r d1.....dk.....	34
1.2.5 曲线指令.....	35
ESC '.....	35
1.2.6 状态传送指令.....	37
ESC v.....	37
1.2.7 汉字字符设置指令.....	38
FS &.....	38
FS.....	38
ESC 6.....	38
ESC 7.....	39
ESC ! n.....	39
1.2.8 垂直制表并打印指令.....	40
FS V.....	40
1.2.9 位图下载及打印命令.....	41
GS * x y d1...dk.....	41
GS / n.....	42
FS q n [xL xH yL yH d1...dk]1...[xL xH yL yH d1...dk]n.....	42
FS p nm.....	43
1.2.10 打印深度设置指令.....	44
ESC s.....	44
1.2.11 标签指令（限标签机型）.....	44
ESC l.....	44
FF ! nl nh.....	44
1.2.12 国际字符集指令.....	45
ESC R n.....	45
GS t n.....	46
1.2.13 连续曲线打印指令.....	47
ESC) n.....	47
ESC # n.....	48
ESC (n.....	48



RD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、条型码打印等功能。

1.1 指令表

命令	功能
ESC @	初始化打印机
FF	打印并走纸到下页首(仅限黑标检测功能机型)
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 =	模式选择命令
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/取消上划线
ESC .	设置/取消下划线
ESC !	设置字符打印模式
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	光栅图形打印命令
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
DC2 k	设置字符混打一维码位置
DC2 j	设置字符混打一维码方向
DC2 n	字符混打一维码
GS W	二维码放大命令
GS k	打印QRCODE二维条码（紧限支持二维码机型）
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
ESC !	设置字符打印模式
FS V	垂直制表及打印
GS *	定义下载位图
GS /	打印下载位图
FS q	下传多个NV位图
FS p	打印下载NV位图
ESC s	打印深度调整命令
ESC i	走纸到下一标签起始位置
FF ! nl nh	退纸命令
ESC R	选择国际字符集



GS t	选择字符代码表
ESC)	打印一水平行上n个线段
ESC #	设置打印曲线打印方式
ESC (	打印n条曲线命令
DC2 EM	查询打印机版本号命令

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。

1.2 指令详解

1.2.1 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] unsigned char str[2];

str[0] = 0x1B;



```
str[1] = 0x40;  
  
SendDataToPrinter(str, 2);
```

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30cm后停止，预印刷黑标的规范请见附录C.预印刷黑标说明。

[例子] `unsigned char str[2];`

`str[0] = 0x0C;`

`SendDataToPrinter(str, 1);`

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);`

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 $[n \times 0.125\text{mm}(0.0049")]$ 。

[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	<i>n</i>
十进制		27	100	<i>n</i>
十六进制		1B	64	<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*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 =**

[名称] 设置模式命令

[格式]	ASCII	ESC	=	<i>n</i>
	十进制	27	61	<i>n</i>
	十六进制	1B	3D	<i>n</i>

[范围] $0 \leq n \leq 1$

[描述] 当 $n=1$ 时，进入驱动模式，当 $n=0$ 时，退出驱动模式。

[注意] 此命令会影响1B 63，当进入驱动模式 $n=1$ 时，1B 63 命令将失效。

ESC c

[名称] 允许/禁止反向打印

[格式]	ASCII	ESC	c	<i>n</i>
	十进制	27	99	<i>n</i>
	十六进制	1B	63	<i>n</i>

[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，此命令仅在非驱动模式下有效，即1B 3D 00模式下有效。

如打印机默认在驱动模式下，可在其前面发送1B 3D 00 再发送1B 63 *n*命令。

[例子] unsigned char str[3];

str[0] = 0x1B;

str[1] = 0x63;

str[2] = 0x0;

SendDataToPrinter(str,3);//。

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] unsignedstr[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15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/设置上划线

[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] = 0x2D;  
str[2] = 0x1;  
  
SendDataToPrinter (str,3); //设置上划线
```



ESC . n

[名称] 取消/设置下划线

[格式] ASCII ESC . *n*

十进制 27 46 *n*

十六进制 1B 2E *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] = 0x2E;`

`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);
```

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°或270°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， $N = nL + nH \times 256$

[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; //`

 `str[2] = 00; //`

 `SendDataToPrinter (str, 4); //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 ESC Q *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 ESC 1 *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 ESC SP *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); // 设置行间距为8个水平点距。
```

ESC a n

[名称] 选择对齐方式

[格式] ASCII ESC a *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[3];`

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

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

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

```
SendDataToPrinter(str, 3); // 设置水平放大2倍.
```

1.2.2 缩放指令

ESC U n

[名称] 水平放大字符

[格式] ASCII ESC U 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.`

ESCX

[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] $1 \leq n \leq 8$ ($1 \leq n1$ 水平倍数 ≤ 8 , $1 \leq n2$ 垂直倍数 ≤ 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.`

1.2.3 图形指令

ESCKnL 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] $0 \leq nL \leq 255$

$0 \leq nH \leq 1$

$0 \leq d \leq 255$

[描述] 本命令只能打印高度为 8 点，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 的低位和高位字节，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] $m = 0, 1, 32, 33$

$0 \leq nL \leq 255$

$0 \leq nH \leq 1$

$0 \leq d \leq 255$

[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：

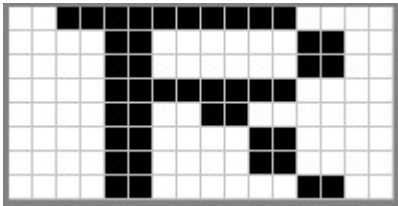
m	垂直点数（高度）	倍宽模式
0	8	两倍宽
1	8	单倍宽
32	24	两倍宽
33	24	单倍宽

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 384，在双倍宽时其值最大为 192。

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;
```

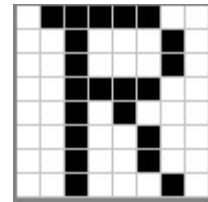
```
str[j++] = 0x96;str[j++] = 0x61;str[j++] = 0x00;str[j++] = 0x0D;//打印出图形
```

```
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;
```

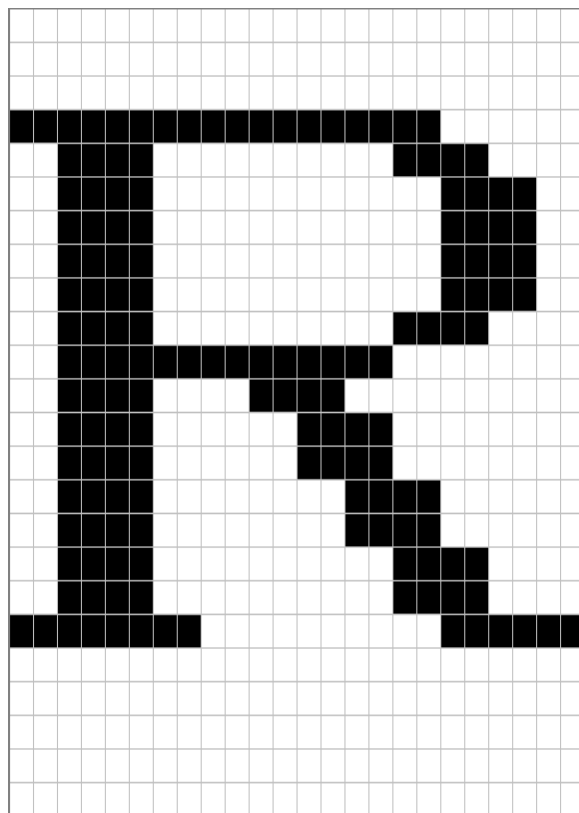
```
str[j++] = 0x96;str[j++] = 0x61;str[j++] = 0x00;str[j++] = 0x0D; ;//打印出图形
```

```
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;

bn 表示字节的第 n 位

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



程序代码如下:

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

```
j=0;
```

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

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

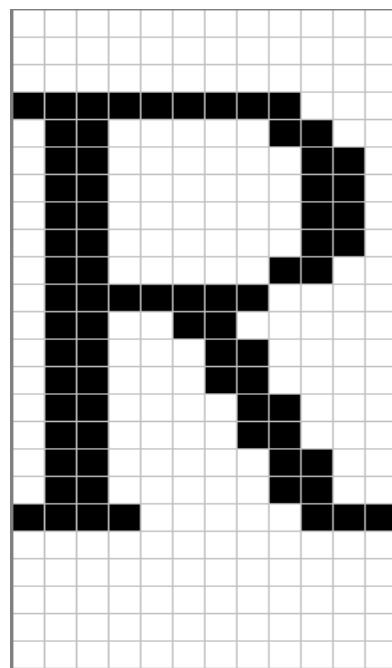


```
str[j++] = 0x2A;  
str[j++] = 32; //m=32(高度 24 点、倍宽)  
str[j++] = 12; //图象宽度为 12dots  
str[j++] = 0;  
//位图数据  
str[j++] = 0x10;str[j++] = 0x00;str[j++] = 0x20;str[j++] = 0x1F;str[j++] = 0xFF;str[j++] =  
0xE0;  
str[j++] = 0x1F;str[j++] = 0xFF;str[j++] = 0xE0;str[j++] = 0x10;str[j++] = 0x20;str[j++] =  
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；
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	0	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	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
	0	1	1	0	0	0	0	1	1	0	0	b0
d3	0	1	1	0	0	0	0	0	1	1	0	b7
	0	1	1	0	0	0	0	0	1	1	0	b6
	1	1	1	1	0	0	0	0	0	1	1	b5
	0	0	0	0	0	0	0	0	0	0	0	b4
	0	0	0	0	0	0	0	0	0	0	0	b3
	0	0	0	0	0	0	0	0	0	0	0	b2
	0	0	0	0	0	0	0	0	0	0	0	b1
	0	0	0	0	0	0	0	0	0	0	0	b0



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码 GS v 0 m xL xH yL yH d1...dk
十进制码 29 118 48 m xL xH yL yH d1...dk
十六进制码 1D 76 30 m xL xH yL yH d1...dk

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

$0 \leq xL \leq 255$

$0 \leq xH \leq 255$

$0 \leq yL \leq 255$

$0 \leq d \leq 255$

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

[描述] 打印光栅位图，由m 值选择光栅位图模式：

m	模式	纵向分辨率 (DPI)	横向分辨率 (DPI)
0, 48	正常	200	200



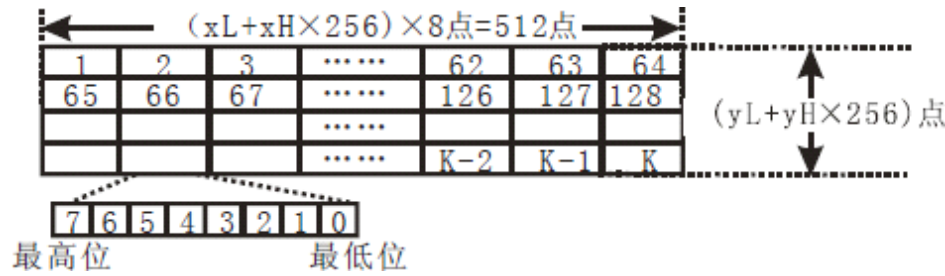
1, 49	倍宽	200	100
2, 50	倍高	100	200
3, 51	倍宽、倍高	100	100

- xL、xH表示水平方向位图字节数 ($xL + xH \times 256$)
- yL、yH表示垂直方向位图点数 ($yL + yH \times 256$)

[注释]

- 在标准模式下，只有打印机缓冲区无数据时该命令才有效。
- 字符放大、加粗、双重打印、倒置打印、下划线、黑白反显等打印模式对该命令无效。
- 位图超出打印区域的部分不打印。
- ESC a (选择对齐模式) 对光栅位图有效。
- 宏定义的过程中，该命令将停止宏定义而执行该命令。该命令不作为宏定义的一部分。
- d 代表位图数据。每个字节的相应位为1表示打印该点，为0不打印该点。

[例子] 当 $xL + xH \times 256 = 64$



1.2.4 条码指令

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] $1 \leq n \leq 127$

[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：

<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 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 $d1\dots dk$ 0

十六进制码: 1D 6B m $d1\dots dk$ 00

格式2: ($65 \leq m \leq 73$)

ASCII码: GS k m n $d1\dots dn$

十进制码: 29 107 m n $d1\dots dn$

十六进制码: 1D 6B m n $d1\dots dn$

[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		$48 \leq d \leq 57$
	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	$1 \leq k(\text{even number})$	$48 \leq d \leq 57$
	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	$1 \leq n \leq 255(\text{even number})$	$48 \leq d \leq 57$
	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

DC2 k xL xH yL yH

[名称]设置字符混打一维码打印位置

[格式] ASCII 码 DC2 k xL xH yL yH

十进制码 18 107 xL xH yL yH

十六进制码 12 6B xL xH yL yH

[例子]

12 6B 00 00 50 00 //设置一维码打印位置为X=0, Y=10mm(0.125*0X50)位置

DC2 j n

[名称]设置字符混打一维码方向

[格式] ASCII 码 DC2 j n

十进制码 18 106 n

十六进制码 12 6A n

[范围] $0 \leq n \leq 3$

[例子]

12 6a 00 //设置一维码方向为0度

DC2 n

[名称]与字符混打一维码

格式参考 GS k

[例子]

1B 40 //初始化

12 21 80 00 //退纸

1B 3D 00 1B 63 00 //设置台序打印

1D 68 50 //设置条码高度



1D 48 02 //设置条码字符显示位置

12 6A 00 //设置条码旋转角度

12 6B 20 00 00 00 //设置条码打印位置 X=4MM,Y=0

12 6E 49 07 7B 43 3D 5E 4D 5E 2E //以CODE128C 混打条码 “6194779446”

1B 4A 10 //设置字符从开始位走纸2MM(0x10*0.125MM)

1B 24 E0 00 //设置字符打印绝对位置X=28mm(0xe0*0.125MM)

32 32 2D 30 36 2D 32 39 0D //打印文字 “22-06-29” 并换行

31 32 3A 33 30 3A 32 39 0D //打印文字 “12: 30: 29” 并换行

1B 4A 20 //以上一行文字为起始位置处走纸2MM

1B 24 00 00 //设置字符打印绝对位置X=0

D1 AA C7 E5 BC EC B2 E2 0D 0A B1 E0 BA C5 A3 BA 30 30 31 31 31 30 33 33 30 38

0D 0A D0 D5 C3 FB A3 BA D5 C5 C8 FD 20 20 20 D0 D5 B1 F0 A3 BA C4 D0 20 20 20

C4 EA C1 E4 A3 BA 36 33 CB EA 0D 0A C3 DA C4 F2 CD E2 BF C6 C3 C5 D5 EF 0D

0A 2A 2A B2 E2 CA D4 BD E1 B9 FB A3 BA BA CF B8 F1 2A 2A 0D 0A

//打印 “血清检测

编号: 0011103308

姓名: 张三 性别: 男 年龄: 63岁

泌尿外科门诊

测试结果: 合格”

1B 69 //走纸到标签缝



GS W n

[名称] 二维码放大命令

[格式] ASCII GS W n



十进制 29 87 n

十六进制 1D 57 n

[范围] $1 \leq n \leq 8$

[描述] 设置二维码放大倍数

缺省值: $n = 2$

[例子] `unsigned char str[4];`

`str[0] = 0x1D;`

`str[1] = 0x57;`

`str[2] = 06;`

`SendDataToPrinter(str, 3); // 设置二维码放大倍数为6`

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

[名称] 打印QR CODE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, QR CODE的版本号最大为20.

• 有关QR CODE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);
```

GS | xL xH yL yH

[名称]设置字符混打QRCODE二维码打印位置

[格式] ASCII 码 GS l xL xH yL yH

十进制码 29 108 xL xH yL yH

十六进制码 1D 6C xL xH yL yH

[例子]

1D 6C 00 00 50 00 //设置二维码打印位置为X=0, Y=10mm位置

GS n m v r d1.....dk

[名称]与字符混打QRCODE二维条码

功能见GS k m v r d1.....dk

[例子]

1B 40 //初始化打印机

1B 73 2B 40 //加深

1D 6C 00 00 50 00 //设置二维码位置为X=0, Y=10mm位置

1D 57 03 //设置二维码放大倍数

1D 6E 20 06 01 //打印混打二维码

35 4B 65 54 77 56 65 4A //二维码数据

34 35 48 55 4E 4D 34 68

47 43 6B 48 6B 78 6E 42

56 4D 34 54 78 67 73 73

33 6F 65 48 7A 55 34 45

52 56 74 6D 44 67 71 76

67 36 76 00



```

1B 24 00 00    //设置文字绝对位置打印0
1B 57 02       //放大两倍
B2 E2 CA D4 0D //打印“测试”
1B 24 80 00    //绝对位置 128点位开始
1B 57 01 0D    //不放大
BF C6 CA D2 3A 43 0D 0A D2 BD B7 CF CA D5 BC AF //文字内容
C8 CB 3A B3 AC BC B6 B9 DC C0 ED D4 B1 0D 0A
D2 BD B7 CF C0 E0 D0 CD 3A D2 A9 CE EF D0 D4 0D 0A
D6 D8 C1 BF 3A 36 2E 33 34 4B 47 0D 0A
B5 C7 BC C7 C8 D5 C6 DA 3A 32 30 32 31 2D 31 32 2D 33 31 0D 0A
B5 C7 BC C7 CA B1 BC E4 3A 31 33 3A 34 35 3A 34 35 0D 0A
1B 69         //走标签到纸缝

```



1.2.5 曲线指令

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 $N = nH \times 256 + nL$



曲线点在水平行上的位置 $X = x_{kH} \times 256 + x_{kL}$ 。

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

[例子] 曲线关系函数为下面 5 个函数

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

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

$$Y3=50$$

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

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

[打印例程]:

```
unsigned char str[50];

float X;

unsigned int  m_cur1,m_cur2,i;

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



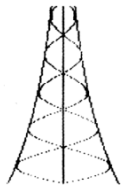
```
str[i++] = 0;

str[i++] = 0x0D;

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

}
```

[结果]:



1. 2. 6 状态传送指令

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	有纸	无纸
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。
```

1.2.7 汉字字符设置指令

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。
```

ESC ! n

[名称] 设置字符打印模式

[格式] ASCII码 ESC ! n

十进制 27 33 n

十六进制 1B 21 n

[范围] $0 \leq n \leq 255$

[说明] 根据n 的值设置字符打印模式

位	0/1	十进 制	十六进制	功能



0, 1, 2	0	0	0	默认0
3	0	0	00	取消加粗模式
	1	8	08	选择加粗模式
4	0	0	00	取消倍高模式
	1	16	10	选择倍高模式
5	0	0	0	取消倍宽模式
	1	32	20	选择倍宽模式
6, 7	0	0	0	默认0

- [注意]
- 当倍宽和倍高模式同时选择时，字符同时在横向和纵向放大两倍。
 - 当一行中部分字符为倍高或更高，所有字符以底端对齐。
 - 粗体模式对英数字符和汉字都有效。

[默认值] n = 0

1.2.8 垂直制表并打印指令

FS V

[名称] 垂直制表并打印

[格式] ASCII FS V

十进制 28 86 m LP1...LPm n IP1...IPn FT1 D11...D1k 0...FTn Dn1...Dnk 0

十六进制 1C56 m LP1...LPm n IP1...IPn FT1 D11...D1k 0...FTn Dn1...Dnk 0

[描述] m 表格竖线数: $0 \leq m \leq 17$;
 LP1...LPm 表格竖线坐标: $0 \leq LPm \leq 48$;
 n 表格项数: $0 \leq n \leq 16$;
 IP1...IPn 表项坐标: $0 \leq IPn \leq 45$;
 FT1 第一个表项的字体类型:

位	功能	值	
		0	1
0	保留		
1	加粗	取消	设定
2	下划线	取消	设定
3	反白	取消	设定
4—7	保留		

[例子]

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

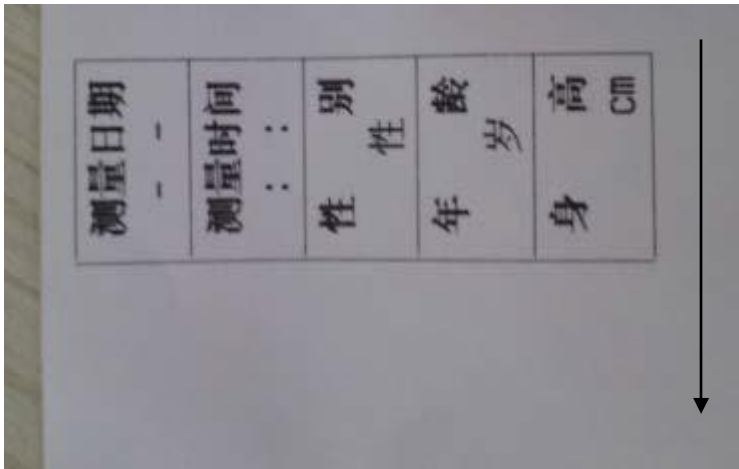
02 20 C4 EA 20 20 20 20 C1 E4 00

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 63 6D 00

[结果]:



[注意]:

此命令目前仅支持面板序打印,台序在使用进发送**1B 63 01**命令后再使用此命令。

1. 2. 9 位图下载及打印命令

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

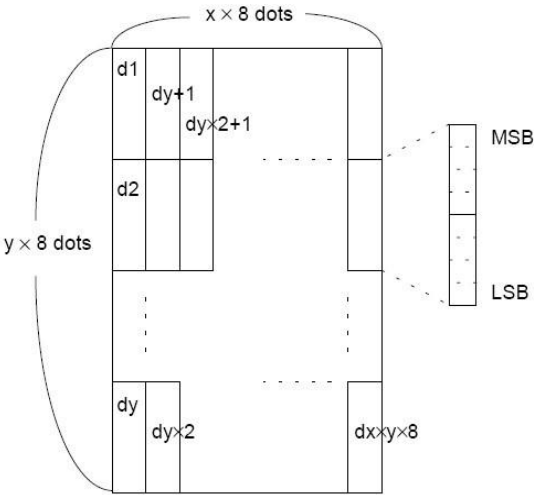
[描述] 用x 和y 指定的点数定义下传位图。

• x*8 为水平方向点数。• y*8 为垂直方向点数。

[注意] • 图形最大数据为12K, 如果x × 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 $d_1, d_2 \dots d_n$ 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 2Fn

[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]: $1 \leq n \leq 30$

$0 \leq xL \leq 255$

$0 \leq xH \leq 3$ (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68K 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。

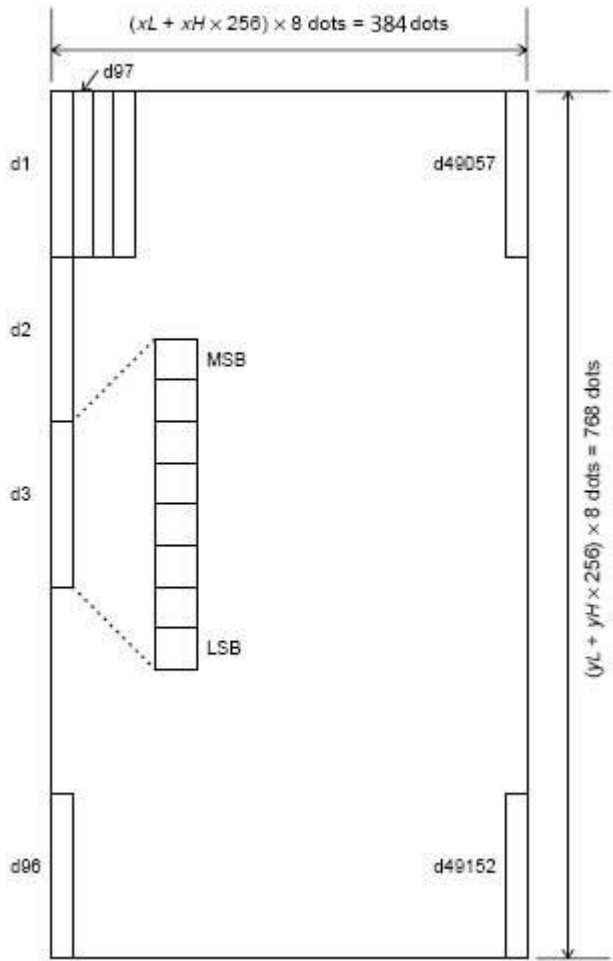
• yL, yH指示位图垂直高度为 $(yL + yH * 256)* 8$ 点。



- 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 = 48, xH = 0, yL = 96, yH = 0



FS p nm

[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] $1 \leq n \leq 30$

[描述] n为FSq命令定义的下传位图编号。

m从下表选择模式:

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



1.2.10 打印深度设置指令

ESC s

[名称] 打印深度调整

[格式] ASCII: ESC s 2B/2D n

 十进制: 27 115 43/45 n

 十六进制: 1B 73 2B/2D n

[说明]

n 的值在 0 ~255 之间;

当使用 1B 73 2B n 为加强深度调节, n 值越大打印深度越深。

当使用 1B 73 2D n 为减少深度调节, n 值越大打印深度越浅 (E2C <0X25)。

恢复指令为 1B 73 2D 0 或 1B 73 2B 0

[注意] • 用户可根据实际情况进行打印深度的调节, 当调浅时, 会降低打印机的功耗, 增加深度时, 打印机功耗会增加。

• 用户在使用该命令时请根据实际情况 (电压及机型) 进行调整, 调整过量将会导致打印机工作不正常。

1.2.11 标签指令 (限标签机型)

ESC I

[名称] 走到下一标签位置

[格式] ASCII ESC i

 十进制 27 105

 十六进制 1B 69

[描述]: 走纸到下一标签头。

[注意]•该命令只用于带标签打印功能机型。

[例子] unsigned char str[4];

 str[0] = 0x1B;

 str[1] = 0x69;

 SendDataToPrinter(str,2);//发送全切纸命令.

FF ! nl nh

[名称] 退纸命令

[格式] ASCII FF ! nl nh i

 十进制 18 33 nl nh

 十六进制 12 21 nl nh

[描述]: 退纸N点命令 (32点1MM) ,nl=N%256,nh=N/256。

[注意]•该命令只用于带标签打印功能机型。



```
[例子] unsigned char str[4];
        str[0] = 0x12;
        str[1] = 0x21;
        str[3] = 0x40;
        str[4] = 0x00;

        SendDataToPrinter(str,4);//退纸2MM
```

1. 2. 12 国际字符集指令

ESC R n

[名称] 选择国际字符集
[格式] ASCII码ESC R n
 十六进制码1B 52 n
 十进制码27 82 n
[范围] 0 n 13
[描述] 按照下表选择n的值设置国际字符集
[缺省值] n = 0
[注意] •该命令仅24点阵机型有效。

n	字符集
0	美国
1	法国
2	德国
3	英国
4	丹麦I
5	瑞典
6	意大利
7	西班牙I
8	日本
9	挪威
10	丹麦II
11	西班牙II
12	拉丁美洲
13	韩国
14	斯洛文尼亚
15	中国

[字符位置差异表]

County	ASCII Code(Hex)											
	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
U.S.A.	#	\$	@	[	\	]	^	`	{		}	~
France	#	\$	à	°	ç	§	^	`	é	ù	è	¨
Germany	#	\$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß



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

GS t n

[名称] 选择字符代码表

[格式] ASCII码 GS t n

十六进制码 1D 74 n

十进制码 29 116 n

[范围] $0 \leq n \leq 21$, $32 \leq n \leq 34$, $64 \leq n \leq 79$,

[描述] 从字符代码表中选择页n。

[注意] •该命令仅24点阵机型有效。

N(十进制)	代码页	N(十进制)	代码页
0	Normal*	32	Codepage 1252 (Windows Latin-1)
1	CodePage437 (USA, Std. Europe)	33	Codepage 1250 (Windows Latin-2)
2	Katakana	34	Codepage 1251 (Windows Cyrillic)
3	CodePage437 (USA, Std. Europe)	64	Codepage 3840 (IBM-Russian)
4	Codepage 858 (Multilingual)	65	Codepage 3841 (Gost)
5	Codepage 852 (Latin-2)	66	Codepage 3843 (Polish)
6	Codepage 860 (Portuguese)	67	Codepage 3844 (CS2)
7	Codepage 861 (Icelandic)	68	Codepage 3845 (Hungarian)
8	Codepage 863 (Canadian French)	69	Codepage 3846 (Turkish)
9	Codepage 865 (Nordic)	70	Codepage 3847 (Brazil-ABNT)
10	Codepage 866 (Cyrillic Russian)	71	Codepage 3848 (Brazil-ABICOMP)
11	Codepage 855 (Cyrillic Bulgarian)	72	Codepage 1001 (Arabic)
12	Codepage 857 (Turkey)	73	Codepage 2001 (Lithuanian-KBL)
13	Codepage 862 (Israel)	74	Codepage 3001 (Estonian-1)



	(Hebrew))		
14	Codepage 864 (Arabic)	75	Codepage 3002 (Estonian-2)
15	Codepage 737 (Greek)	76	Codepage 3011 (Latvian-1)
16	Codepage 851 (Greek)	77	Codepage 3012 (Latvian-2)
17	Codepage 869 (Greek)	78	Codepage 3021 (Bulgarian)
18	Codepage 928 (Greek)	79	Codepage 3041 (Maltese)
19	Codepage 772 (Lithuanian)		
20	Codepage 774 (Lithuanian)		
21	Codepage 874 (Thai)		

1. 2. 13 连续曲线打印指令

ESC) n

[名称] 打印一水平行上n个线段

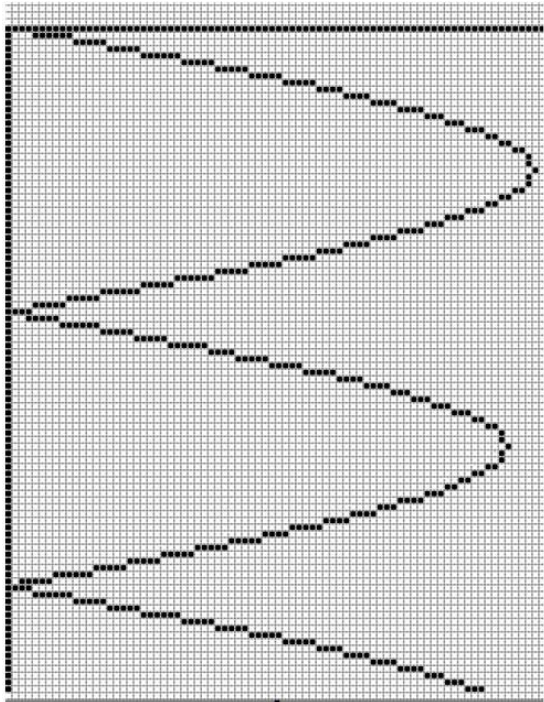
[格式] ASCII 码 ESC) n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH 00

十进制码 2741 n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH 00

十六进制码 1B 29 n x1sL x1sH x1eL x1eH ... xnsL xnsH xneL xneH 00

[范围] $0 \leq n \leq 255$

[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行上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] = 0x29;
```

```
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 # n

[名称] 设置打印曲线打印方式

[格式]	ASCII	ESC	#	n
	十进制	27	35	n
	十六进制	1B	23	n

[范围] $0 \leq n \leq 1$

[描述] 此指令影响1B 28 曲线命令打印方式

$n = 0$ 1B 28曲线命令打印曲线不自动连点。

$n = 1$ 1B 28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 (n

[名称] 打印 n 条曲线命令

[格式] ASCII码ESC (n $x1L$ $x1H$... xnL xnH 00

十进制码2740 n $x1L$ $x1H$... xnL xnH 00

十六进制码1B 28 n $x1L$ $x1H$... xnL xnH 00

[范围] $0 \leq n \leq 20$

[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（由ESC #命令控制）。

打印一条曲线时设置n=1；

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

```
int i;
```

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

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

```
SendStr[2] = 2; //打印2条曲线
```

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

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

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

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

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