JPM112

(JPM112I JPM112B)

便携微型针式打印机

开发手册(V1.0)

目　　录

修订历史

一、 概述

JPM112 系列便携针式打印机是新一代性能优异的采用针式（撞击式）打印技术的，支持红外、蓝牙无线通讯技术的便携式票据打印机。JPM112 采用了通用的 ESC/POS 指令集，便于开发者开发。

JPM112 便携针式打印机包含两个型号，分别为 JPM112I，JPM112B，JPM112I 采用红外数据通讯（IrDA）技术，JPM112B 采用蓝牙（BLUETOOTH）数据通讯技术。

其性能指标如下：

型号	JPM112I	JPM112B
打印方式	梭式点阵打印	
打印纸宽	58mm	
外形尺寸	99×97×36（L×W×H）	
重　　量	258g(不含纸卷)	
纸卷直径	≤30mm	
打印行宽	16 汉字/行 32 英文字符/行	
汉字编码方式	GBK，BIG5，Unicode	
内置字库	ASCII 字库(12×6、16×8)、GBK(12×12、16×16)	
打印速度	5.5mm/s 1.2lines/s(12 点阵字库)	
打印机芯寿命	1,500,000 行	
可打印内容	英文、数字、各种符号、汉字、图形、曲线、预存储图标	
分 辨 率	5 点/毫米(横向)	
数据通讯接口	RS232/IrDA/Vir/SIR	RS232/蓝牙 V1.1 CLASS 2
电　　源	700mAh/7.4V 可充电锂电池	
充电方式	带机充电	
一次充电可打印长度	12.5%打印密度下可打印 18m，25%打印密度 15m	
异常检测	缺纸检测、电量不足检测，检测异常后蜂鸣器报警	
黑标定位功能	有	
拷贝能力	1 份原件+1 份拷贝	
切纸方式	手动撕纸	
指 令 集	ESC/POS 兼容指令集	
使用环境	温度 -10°C~50°C 湿度 20%~85%	
充电环境	温度 5°C~40°C 湿度 20%~85%	
储存环境	温度 -20°C~70°C 湿度 5%~95%	
标配附件[注]	充电器、电池	
可选附件[注]	专用串口数据线、充电座	

注：有关附件的详细信息请参阅《JPM112 微型打印机使用手册》

二、 通讯接口

JPM112 系列便携式微型打印机可用五种接口方式与主机进行数据传输，分别为 IrDA（IrCOMM 协议）、RAW-IR/SIR（原始红外）、VIR、蓝牙、RS232 异步串行口。其中蓝牙是无线数据通讯接口；IrDA 为符合国际红外数据通讯协会制定的 IrCOMM 协议的数据传输方式；RAW-IR/SIR 为符合国际红外数据通讯协会物理层规范的数据通讯方式；VIR 为在 RAW-IR/SIR 基础上集成了 VIR 协议的数据通讯方式；RS232 为有线异步串口。JPM112 便携式微型打印机只支持能使用其中一种或几种数据通讯方式的主机设备，在开发前请先确认主机设备至少支持以上五种通讯方式中的一种。

JPM112I 与 JPM112B 打印机的打印接口不同，JPM112I 打印机可以使用串口(RS232)与红外(IrDA)接口打印，JPM112B 可以使用串口(RS232)与蓝牙(Bluetooth)接口打印。

2.1 蓝牙（Bluetooth）接口

蓝牙是一种支持设备短距离通信的无线电技术。能在包括移动电话、PDA、打印机、笔记本电脑、无线耳机等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。如需蓝牙更详细的信息，请访问蓝牙官方网站：http://www.bluetooth.org。

JPM112B 支持蓝牙无线数据传输接口，符合 Bluetooth 1.1 规范，功率级别为 CLASS 2。JPM112B 提供蓝牙虚拟串口服务。

JPM112B 是一个蓝牙从设备，只能由蓝牙主设备（如 PDA、手机、笔记本电脑）发起配对请求并连接打印机，其他如蓝牙耳机等蓝牙从设备无法通过蓝牙驱动打印机打印。

JPM112B 缺省的设备名为 JPM112B，开发者可根据自己的需要更改设备名。更改设备名的方法详见【三、JPM112 打印机工具软件】。

JPM112B 缺省的蓝牙连接密码为 0000，开发者可根据自己的需要更改连接密码。更改设备名的方法详见【三、JPM112 打印机工具软件】。

2.1.1 配对

JPM112B 便携式微型打印机工作前需与驱动 JPM112B 便携式微型打印机的主设备配对，配对过程由主设备发起。

通常的配对方法如下：

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 JPM112B 打印机

4、输入密码“ 0000 ”，需要注意的是，打印机的密码和蓝牙设备名可以通过工具软件 JPM112_CONFIG 修改(参见【三、JPM112 打印机工具软件】)，如果设备名和密码已经修改，需输入修改后的密码。

5、完成配对。

具体的配对方法请参阅主设备蓝牙功能说明。

配对时，JPM112B 便携式微型打印机必须处于开机状态。

注意：配对时，请不要将多台打印机同时开机，否则可能无法判断配对成功的是哪一台打印机。

配对成功后，其他上位机仍然可以与该打印机配对，每台打印机最多可以与 8 台上位机配对，如果更多的上位机与打印机配对的话，那么最早与打印机的上位机会被打印机从配对列表中自动清除，此时如果这台上位机需要驱动打印机打印的话，需要重新配对。

2.1.2 使用蓝牙接口驱动打印

对于采用 WINCE（包括 SMARTPHONE、POCKET PC、WNDOWS MOBILE 等）操作系统的上位机，配对成功后，就可以通过虚拟蓝牙串口向 JPM112B 便携式微型打印机发送打印数据进行打印了。如果上位机没有虚拟蓝牙串口，如要驱动 JPM112B 便携式微型打印机打印，请咨询上位机供应商。

2.2 红外（IR）端口

红外数据通讯技术是红外数据协会(IrDA)开发并发展起来的一项用红外光作为通讯数据载体的一种无线数据通讯技术。

红外(IR)端口作为无线数据传输接口，因其功耗低、技术成熟、使用方便等等诸多原因现在为大多数便携式设备作为数据通讯的主要手段。大部分的便携式设备都有红外(IR)端口，比如所有的 WINCE 掌上电脑、PALM 掌上电脑、笔记本电脑，一部分手机、大部分便携式数据采集器，以及一小部分用于野外作业的测量仪器。

JPM112I 打印机在硬件上符合 IrDA1.1 物理层的规范。

由于红外(IR)端口是以红外光线作为数据载体，所以在数据发送设备和接收设备之间不能有障碍物，双方的红外端口要互相对准，并且距离不能太远。所以 JPM112I 打印机与其他符合标准 IrDA 物理层规范的主机设备通过红外(IR)端口连接时，要注意主机和 JPM112I 打印机之间不能有障碍物，红外端口之间的夹角不能大于 30°，距离不能超过 0.5M。

JPM112I 可以使用原始红外(RAW-IR/SIR)与主机进行通讯，也可以选择采用 VIR 协议或 IrCOMM 协议(IrDA 标准)与主机进行通讯。

当采用原始红外与主机进行通讯时，红外收发器只是简单地按照一定地编码规则将串口数据信号转为红外光信号或将光信号转为串口数据信号。在这种用法下，红外端口被称为“ 原始红外(RAW-IR/SIR) ”，当您使用原始红外(RAW –IR/SIR)时，无法跟 IrDA 兼容，因为在收发数据的过程中，软件并没有使用 IrDA 协议栈。

VIR 是一种更安全可靠的红外数据通讯方式，即在原始红外的基础上增加了 VIR 协议。如果用户主机不支持 IrCOMM 协议，推荐使用 VIR 协议。

相较于 VIR 协议，IrCOMM 虽然也是基于原始红外硬件基础上的协议，但它是一种更通用的有国际红外数据通讯协会制定的红外数据传输协议。IrCOMM 是 IrDA 协议的一个子集，IrDA 协议是在

RAW-IR/SIR (原始红外)的硬件基础之上为了确保数据传输的稳定性、可靠性、易用性而由 IrDA 协会开制定的无线数据通讯协议，几乎所有支持 IrDA 协议栈的便携式设备(如 WINCE、POCKET PC、PALM 各种红外手机等等)都支持 IrCOMM。在这种模式下，如果驱动打印机的主机设备使用的是支持 IrDA 协议栈的操作系统(比如 WINCE、PALM OS)，那么对于开发者来说，IrCOMM 端口就是一个由软件虚拟的串行端口，如果主机设备的操作系统没有 IrDA 协议栈，如果要实现 IrCOMM 模式数据传输，那么需要开发者自己编写 IrDA 协议栈。主机设备是否支持 IrCOMM，请参阅主机设备的开发资料或向主机设备制造商咨询。

由于 JPM112I 打印机中 RAW-IR/SIR (原始红外)端口和 VIR 端口和 IrCOMM 端口使用同样的硬件资源，所以 RAW-IR/SIR (原始红外)、VIR 和 IrCOMM 不能同时使用，JPM112I 默认的红外模式为 IrCOMM，如要更改当前红外端口的使用模式，需要专用的工具 JPM112CONFIG 修改设置，详见【三、JPM112 打印机工具软件】。

如要了解详细的IrDA协议请参见IrDA协会官方网站（http://www.irda.org/）公布的技术资料。

2.2.1 原始红外（RAW-IR）

原始红外(IR)端口由于是直接把红外(IR)收发器附加在 RS232 异步串口的输入输出端，所以红外收发器只是简单地将 RS232 异步串口数据信号按照一定的编码规则转为红外光信号或将光信号转为串口数据信号。因此对于原始红外端口来说，除了细微的差别，主机应用开发者只需要象操作 RS232 异步串口那样操作主机的 RAW-IR/SIR(原始红外)就可以了。

在操作主机 RAW-IR/SIR(原始红外)端口之前，必须先知道 RAW-IR/SIR(原始红外)端口的端口号，RAW-IR/SIR(原始红外)端口的端口号可以从主机设备的开发资料或主机设备制造商处获知。对于部分可以使用 RAW-IR/SIR(原始红外)端口的主机设备，打开 RAW-IR/SIR(原始红外)端口的方法可能和打开串口的方法不一样，请仔细参阅主机设备的开发资料。

当然 RAW-IR/SIR(原始红外)端口和标准的 RS232 异步串行口仍然有细微的差别，原始红外端口由于采用无线红外连接方式，所以只有串口的 TXD 信号和 RXD 信号有效，对于主机 RS232 其他引脚的操作不会对 RAW-IR/SIR(原始红外)端口产生影响。

JPM112I 打印机在休眠状态下可以通过 RAW-IR/SIR(原始红外)端口唤醒。

JPM112I 打印机的 RAW-IR/SIR(原始红外)可以工作的波特率为：9600bps，19200bps，38400bps，57600bps，115200bps。出厂时打印机的 RAW-IR/SIR(原始红外)波特率被设置成 9600bps，如果用户要更改波特率，需要专用的工具 JPM112CONFIG 修改设置，详见【三、JPM112 打印机工具软件】。

注意：并不是所有的主机的 RAW-IR/SIR(原始红外)都能在这些波特率下工作，在修改波特率前，请先确认您所拥有主机设备的 RAW-IR/SIR(原始红外)是否能在此波特率下工作。

在主机设备编程时，RAW-IR/SIR(原始红外)端口的设置请遵照如下设置：

数据位：8 位； 停止位 1 位； 奇偶校验：无； 流控制 无。

在使用 RAW-IR/SIR(原始红外)端口时请注意，虽然对于一般的红外源，RAW-IR/SIR(原始红外)有足够的抗干扰能力，但是对于正在试图搜索其他红外设备的红外设备(比如笔记本电脑、打开红外功能的手机)所发出的红外光干扰，并不能有效的屏蔽，所以使用时切记不能靠近这些红外信号源。

2.2.2 VIR

为使便携式微型打印机能可靠地与不带 IrDA 协议栈的诸多红外手持终端进行红外数据通讯，开发了 VIr 协议，本协议是基于符合 IrDA 物理层规范的硬件的打印数据通讯协议。

任何可以使用原始红外收发数据的设备都可以使用此协议控制 JPM112I 打印。

打印机在休眠状态下可以通过 VIR 协议唤醒。

VIR 协议对于没有集成 IrDA 协议栈的红外数据通讯设备来说，是一个很好的替代 IrDA 协议的数据通讯协议。VIR 协议消耗资源少，并且可以很容易地实现，但同时可以有效地防止红外数据传输过程中出现误码。有关 VIR 协议的定义及如何实现 VIR 协议请见【附录 D】，如需技术支持，请浏览本公司网站 http://www.jqsh.com。

2.1.3 IrCOMM

IrDA 的协议有很多，IrCOMM 是 IrDA 协会推荐使用在打印机上的红外数据通讯协议。

在使用 IrCOMM 协议时，打印机可以完全杜绝其他红外源的干扰。

JPM112I 支持 IrCOMM 协议。

在使用 IrCOMM 时，虽然把它当作一个虚拟串行口，但是不需要设置波特率，数据传输的真实波特率是在 IrCOMM 协议工作的时候自适应的。并且因为 IrCOMM 协议负责数据校验、数据缓冲等等工作，所以对于 IrCOMM 串口的其他设置都是没有意义的。

对于开发者来讲，需要知道的是端口号以及如何将其打开。IrCOMM 端口的端口号以及打开方法可以从主机设备的开发资料或主机设备制造商处获知。

打印机在休眠状态下可以通过 IrCOMM 端口唤醒。

如果开发者需要自己开发IrCOMM协议，请参见IrDA协会官方网站（http://www.irda.org/）公布的技术资料。

2.3 RS232 端口

RS232 接口是最常用的数据通讯接口。JPM112 打印机都带 RS232 异步串行数据接口。

JPM112 打印机 RS232 端口规格：

数据传送： 串行

同步方式： 异步

握手信号： 无

流控制： 硬件流控制/软件流控制/无 可选

信号电平： RS232电平

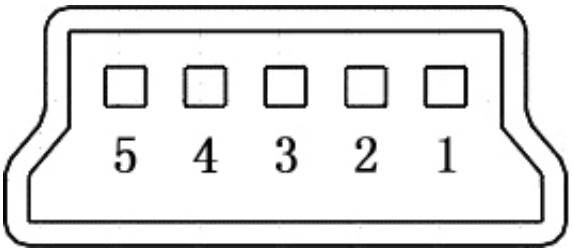
波特率： 9600bps、19200bps、38400bps、57600bps、115200bps可选。

数据字长度： 8位

校验方式： 无

停止位： 1位

插座管脚定义(打印机侧)：5PIN MINI-USB 下图为打印机RS232通讯插座的管脚定义：



1、 RXD
2、 TXD
3、 RTS
4、 保留
5、 GND

TXD：打印机数据发送

RTS：打印机请求数据（仅当打印机设为硬件流控制方式时，该管脚生效，否则该管脚无用，设置打印机为硬件流控制方式的方法见【三、JPM112 打印机工具软件】）

GND：地

RXD：打印机数据接收

注意：保留的管脚本公司有其他用途，开发者如要自行制作连接线，请不要使用保留的管脚，否则可能导致打印机不能正常工作甚至损坏打印。

打印机的数据接收缓冲区大小为 3K byte，当打印数据量小于 3K byte 时，无需使用流控制，接线方式如下：

打印机	主设备
RXD	TXD
GND	GND

当打印数据量大于 3K byte 时，需要使用流控制，当使用蓝牙、IrCOMM 与打印机连接时，开发者无需考虑流控制，当使用 VIR 协议时，开发者需根据 VIR 的流控制方式来避免打印机的数据缓冲区溢出，原始红外无法使用流控制。

打印机为主机提供了两种流控制方式，硬件流控制和软件流控制。设置打印机为流控制方式的方法见【三、JPM112 打印机工具软件】。

当打印机被设为硬件流控制时，应按如下方法连接主设备与打印机：

打印机	主设备
TXD	RXD
RTS	CTS

RXD	TXD
GND	GND

主设备在通讯过程中监测 CTS 的电平，当 CTS 电平为高时，主设备可以向打印机发送数据，当 CTS 电平为低时，表示打印机忙，需停止向打印机发送数据，直到再次监测到 CTS 电平为高时，继续发送打印数据。

软件流控制采用 XON/XOFF 方式，当采用软件流控制方式时，应按如下方式连接主设备与打印机：

打印机	主设备
TXD	RXD
RXD	TXD
GND	GND

当使用软件流控制时，主设备需检测自身 RXD 的数据以判断打印机数据缓冲区是否满。具体方法如下：开始打印时，主设备向打印机发送数据，同时监测串行口接收的数据，当接收到 XOFF(0x13)时，停止向打印机发送数据，当接收到 XON(0x11)时，重新开始发送数据，再次接收到 XOFF 时，再停止向打印机发送数据，等待再次接收 XON(0x11)时，再重新开始发送数据，如此循环，直到发送完打印数据。

三、 JPM112 打印机工具软件

JPM112 打印机有两个运行在 PC 上的工具软件 PRINTER_CONFIG，PRINTER_LOGO，这两个软件可以从本公司网站（http://www.jqsh.com）上下载。

PRINTER_CONFIG 为设置打印机用户参数的工具软件，可设置的参数如下：

- 打印机串口波特率，
- 红外通讯模式(仅限 JPM112I)，可选的红外通讯模式为 SIR(原始红外)、VIR、IrCOMM。
- SIR(原始红外)和 VIR 的波特率(仅限 JPM112I)，
- 打印机的蓝牙设备名称及密码（仅限 JPM112B)，
- 走纸模式，用于设置手动走纸的长度。

具体的使用方法请参见 PRINTER _CONFIG 帮助，该帮助随 PRINTER _CONFIG 一起发布。

PRINTER_LOGO 为 JPM112 打印机预存储图片上传下载的工具，用于上传下载 JPM112 中的预存储图片。

具体的使用方法请参见 PRINTER _LOGO 的帮助，该帮助随 PRINTER _LOGO 一起发布。

四、 打印控制指令

JPM112 打印机控制采用 ESC/POS 兼容指令。通过打印机的通讯接口向打印机发送打印机控制指令和打印数据以控制打印机打印。

下表中是打印指令的简表，按照功能分类：

章节	指令	简述
打印机控制指令		
4.1.1	ESC @	打印机初始化
4.1.2	FF	打印并走纸到下页首（仅用于黑标定位时）
4.1.3	LF	打印并换行
4.1.4	CR	打印并回车
4.1.5	ESC J n	打印并进纸 n 个垂直运动单位
4.1.6	ESC d n	打印并进纸 n 行
4.1.7	HT	移动打印位置到下一个水平制表位置
4.1.8	FS U nL nH…	打印 Unicode 编码字符
字符参数设置命令		
4.2.1	ESC ! n	设置字符打印模式
4.2.2	GS ! n	设置字符大小
4.2.3	ESC M n	设置打印字体
4.2.4	ESC - n	设置/取消下划线打印
4.2.5	ESC E n	设置/取消粗体打印
4.2.6	ESC G n	设置/取消重叠（粗体）打印、效果同 ESC E
4.2.7	GS B n	设置/取消反白打印
4.2.8	ESC V n	设置/取消字符旋转
打印排版参数设置命令		
4.3.1	ESC $ nL nH	设置绝对打印位置
4.3.2	ESC D n1 n2…nk NULL	设置水平制表位
4.3.3	ESC \ nL nH	设置相对打印位置
4.3.4	ESC 2	设置行间距为缺省行间距
4.3.5	ESC 3 n	设置行间距
4.3.6	ESC SP n	设置字间距
4.3.7	ESC a n	设置对齐方式
图形/图象打印命令		
4.4.1	ESC * m nL nH d1…dk	打印位图
4.4.2	GS * x y d1…dk	定义下传位图
4.4.3	GS / n	打印下传位图
4.4.4	FS P n	打印预存储位图
曲线打印命令		
4.5.1	GS `	打印曲线
4.5.2	GS "	打印曲线指示字符

用户自定义字符		
4.6.1	ESC % n	允许/禁止用户自定义字符
4.6.2	ESC & y c1 c2 …	定义用户自定义字符
4.6.3	ESC ?	取消用户自定义字符
汉字命令		
4.7.1	FS &	设定汉字字符打印模式
4.7.2	FS 2 c1 c2 d1…dk	定义用户自定义汉字
4.7.3	FS .	解除汉字字符打印模式

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 `PrtSendData` 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：

```
PrtSendData(char *buf, int len)
```

描述：向打印机发送数据；

`char *buf`：打印数据的指针；

`int len`：数据长度，单位：字节。

4.1 基本控制指令

4.1.1 ESC @

[名称] 初始化打印机

[格式] ASCII码 `ESC @`

十六进制码 `1B 40`

十进制码 `27 64`

[描述] 清除打印缓冲区中的数据，复位打印机打印参数到当打印机缺省参数。

[注意] • 不是完全恢复到出厂设置，用户参数设置（见三、JPM112打印机工具软件）不会被更改。

[例子]
```
char SendStr[3];
SendStr[0] = 0x1B
SendStr[1] = 0x40;
PrtSendData(SendStr, 2);
```

4.1.2 FF

[名称] 打印并走纸到下页首

[格式] ASCII码 `FF`

十六进制码 0C
十进制码 12

[描述] 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] • 打印后，删除打印缓冲区中的数据。
• 该命令设置打印位置为行的起始点。
• 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0.5m后停止，预印刷黑标的规范请见**附录B.预印刷黑标说明**。

[例子]
```
char SendStr[2];
SendStr[0] = 0x0C;
PrtSendData( SendStr, 1);
```

4.1.3 LF

[名称] 打印并换行

[格式] ASCII码 LF
十六进制码 0A
十进制码 10

[描述] 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。

[参考] **CR**

[例子]
```
char SendStr[2];
SendStr[0]='\n';  //C语言中'\n'即为换行
PrtSendData( SendStr, 1);
```

4.1.4 CR

[名称] 打印并回车

[格式] ASCII码 CR
十六进制码 0D
十进制码 13

[描述] 打印但不进纸。

[注意] • 打印结束后，将下一行的开始设定为打印起始位置。

[参考] **LF**

[例子]
```
char SendStr[2];
SendStr[0]='\r';  //C语言中'\r'即为回车
PrtSendData(SendStr,1);
```

4.1.5 ESC J *n*

[名称] 打印并进纸
[格式] ASCII码 ESC J *n*
十六进制码 1B 4A *n*
十进制码 27 74 *n*
[范围] 0 ≤ *n* ≤ 255
[描述] 打印输出打印缓冲区中的数据，并进纸*n*个 垂直点距。
[注意] • 打印结束后，将下一行的开始设定为打印起始位置。
• 一个垂直点距为0.33mm,以下同。
[参考] **ESC d**
[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = 'J';
SendStr[2] = 3;
PrtSendData( SendStr, 3);//走纸1mm
```

4.1.6 ESC d *n*

[名称] 打印并进纸*n* 行
[格式] ASCII码 ESC d *n*
十六进制码 1B 64 *n*
十进制码 27 100 *n*
[范围] 0 ≤ *n* ≤ 255
[描述] 打印打印缓冲区中的数据并进纸*n*字符行。
[注意] • 该命令设置打印起始位置为行起点。
[参考] **ESC J**
[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = 'd';
SendStr[2] = 2;
PrtSendData( SendStr, 3);//走纸2行
```

4.1.7 HT

[名称] 移动打印位置到下一个水平制表位置
[格式] ASCII码 HT
十六进制码 09
十进制码 9
[描述] 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。
• 水平定位位置的缺省设定为字符A(6×12)的8个字符宽度(第9，17，25， … 列)。

[参照] **ESC D**

[例子]
```
char NextPos = 9;
PrtSendData("商品名",6);
PrtSendData(&NextPos,1);
PrtSendData("单价",4);
PrtSendData(&NextPos,1);
PrtSendData("数量",4);
PrtSendData(&NextPos,1);
PrtSendData("金额",4);
```

4.1.8 FS U *nL nH*

[名称] 按Unicode编码向打印发送数据

[格式]
ASCII码 FS U *nL nH*
十六进制码 1C 55 *nL nH*
十进制码 28 85 *nL nH*

[描述] 打印n(n=nL+nH*256)个Unicode编码字符。

[注意] • 因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编码字符处理；
• 该指令中输入的汉字只支持GBK字库中包含的汉字，没有包含Unicode中所包含的所有汉字；
• 有关GBK的详细信息请参阅全国信息技术标准化技术委员会制定的“汉字内码扩展规范”；
• 有关Unicode的详细信息可以参见Unicode的官方网站http://www.unicode.org；
• 本命令不受汉字命令的影响，也不受自定义字符命令的影响；
• 本命令可以嵌入其他ESC/GS/FS指令，但要采用Unicode编码。

[例子]
```
char SendStr[64];
SendStr[0]=0x1C; SendStr[1]='U'; SendStr[2]=11; SendStr[3]=0;
SendStr[4]=0x55; SendStr[5]=0x00; //U+0055:'U'
SendStr[6]=0x4E; SendStr[7]=0x00; //U+004E:'N'
SendStr[8]=0x49; SendStr[9]=0x00; //U+0049:'I'
SendStr[10]=0x43; SendStr[11]=0x00; //U+0043:'C'
SendStr[12]=0x4F; SendStr[13]=0x00; //U+004F:'O'
SendStr[14]=0x44; SendStr[15]=0x00; //U+0044:'D'
SendStr[16]=0x45; SendStr[17]=0x00; //U+0045:'E'
SendStr[18]=0x53; SendStr[19]=0x62; //U+6253:'打'
SendStr[20]=0x70; SendStr[21]=0x53; //U+5370:'印'
SendStr[22]=0x4B; SendStr[23]=0x6D; //U+6D4B:'测'
SendStr[24]=0xD5; SendStr[25]=0x8B; //U+8BD5:'试'
PrtSendData(SendStr,26);
PrtSendData("\n",1);
```

4.2 字符参数设置命令

4.2.1 ESC ! *n*

[命令] 选择打印模式

[格式] ASCII码 ESC ! *n*

十六进制码 1B 21 *n*

十进制码 27 33 *n*

[范围] $0 \le n \le 255$

[描述] 通过指定参数*n* 的值选择打印模式。参数*n* 的定义如下：

位	值	意义
0	0	西文字符（半宽）字体A (6×12)，汉字字符（全宽）字体A（12×12）
	1	西文字符（半宽）字体B (8×16)，汉字字符（全宽）字体B（16×16）
1	—	未定义
2	—	未定义
3	0	取消粗体模式
	1	设置粗体模式
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	—	未定义
7	0	取消下划线模式
	1	设置下划线模式

[注意]
- 当同时选择倍高及倍宽模式时，则打印出四倍大小字符。
- 打印机可以为所有字符加下划线，但不能为由HT命令产生的空白或顺时针旋转90°的字符加下划线。
- 当一行中有一些倍高或更高字符时，行中所有字符都沿基线对齐。
- ESC M也可设定字体。最后接收到的命令的设定有效。
- ESC E也可设定或取消粗体模式。最后接收到的命令的设定有效。
- ESC – 也可设定或取消下划线模式，最后接收到的命令的设定有效。
- GS !也可设定字符大小。最后接收到的命令的设定有效。
- 本命令对英数字符和汉字都有效。

[缺省值] *n* = 0

[参照] **ESC -, ESC E, GS !, ESC M**

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '!';
SendStr[2] = 0x28;// 00101000 倍宽粗体
PrtSendData( SendStr, 3);
```

4.2.2 GS ! *n*

[名称] 放大字符

[格式] ASCII码 GS ! *n*

十六进制码 1D 21 *n*

十进制码 29 33 *n*

[范围] 0 ≤ *n* ≤ 255 (1 ≤垂直倍数≤ 2, 1 ≤ 水平倍数≤ 2)

[描述] 用位0~3位选择字符高度，用位4~7位选择字符宽度，如下所示：

0	1	2	3	高度
0	0	0	0	1倍
1	0	0	0	2倍

4	5	6	7	宽度
0	0	0	0	1倍
1	0	0	0	2倍

[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。
- 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ESC V，当两个命令同时有效时，字符显示是先旋转，再放大。
- 当字符以不同的尺寸在一行中放大时，一行中所有的字符沿基线对齐。
- 用**ESC !** 命令也可以设置字符大小。以最后接收到的命令设置当前模式。

[缺省值] *n* = 0

[参考] **ESC !**

[例子]
```
char SendStr[4];
SendStr[0] = 0x1D;
SendStr[1] = '!';
SendStr[2] = 0x01;// 00000001 倍高
PrtSendData( SendStr, 3);
```

4.2.3 ESC M *n*

[名称] 选择字符字体

[格式] ASCII码 ESC M *n*

十六进制码 1B 4D *n*

十进制码 27 77 *n*

[范围] *n* = 0, 1, 16, 17,18,19

[描述] 选择字符字体。

N（十进制）	意义
0	西文字符（半宽）字体(6×12)
1	西文字符（半宽）字体(8×16)
16	简体汉字字符字体12×12

17	简体汉字字符字体16×16
18	BIG5汉字字符字体12×12
19	BIG5汉字字符字体16×16

[注意] • ESC ！也可设定字体。最后接收到的命令的设定有效。
• 当使用该命令设定字体时。可以分别设定西文字体和汉字字体，而且互不影响。

[参考] **ESC !**

[例子]
```
char SendStr[8];
SendStr[0]=0x1B;
SendStr[1]='M';
SendStr[2]=0;// 西文6×12
SendStr[0]=0x1B;
SendStr[1]='M';
SendStr[2]=0x11;// 简体中文16×16
PrtSendData( SendStr, 6);//以后打印的中文字体为16×16，西文为6×12
```

4.2.4 ESC – *n*

[名称] 设置／取消下划线

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

[范围] $0 \le n \le 2$

[描述] 基于以下的*n*值，设定／解除下划线模式：

n（十进制）	意义
0	解除下划线模式
1	设定下划线模式(1 点粗)
2	设定下划线模式(2 点粗)

[注意] • 打印机不能给旋转字符以及反白字符打印下划线。
• 改变字符大小不影响当前下划线的粗细。
• 使用ESC ！也可以设定或解除下划线模式。最后接收的命令设定有效。
• 该命令对英文和汉字字符都有效。

[缺省值] *n* = 0

[参照] **ESC !**

[例子]
```
char SendStr[3];
SendStr[0] = 0x1B;
SendStr[1] = '-';
SendStr[2] = 1;//单行下划线
PrtSendData( SendStr, 3);
```

4.2.5 ESC E *n*

[名称] 设定/解除粗体打印

[格式] ASCII码 ESC E *n*

十六进制码 1B 45 *n*

十进制码 27 69 *n*

[范围] 0 ≤ *n* ≤ 255

[描述] 设定或解除粗体打印模式。

当*n* 的最低位(LSB)为0时，解除粗体打印模式。

当*n* 的最低位(LSB)为1时，设定粗体打印模式。

[注意] • 仅*n* 的最低有效位允许使用。

• 使用**ESC !**也可以设置或取消粗体模式。最后接收的命令设定有效。

[缺省值] *n* = 0

[参照] **ESC !, ESC G**

[例子]
```
char SendStr[3];
SendStr[0] = 0x1B;
SendStr[1] = 'E';
SendStr[2] = 1;//粗体
PrtSendData(SendStr,3);
```

4.2.6 ESC G *n*

[名称] 设定/解除重叠打印

[格式] ASCII码 ESC G *n*

十六进制码 1B 47 *n*

十进制码 27 71 *n*

[范围] 0 ≤ *n* ≤ 255

[描述] 设定或解除重叠打印模式。

当*n* 的最低有效位(LSB)为0时，解除重叠打印模式。

当*n* 的最低有效位(LSB)为1时，设定重叠打印模式。

[注意] • 仅*n* 的最低有效位允许使用。

• 在重叠模式和粗体模式中打印机输出是相同的。

[缺省值] *n* = 0

[参照] **ESC E, ESC !**

[例子]
```
char SendStr[3];
SendStr[0] = 0x1B;
SendStr[1] = 'G';
SendStr[2] = 1;//重叠
PrtSendData( SendStr, 3);
```

4.2.7 GS B *n*

[名称] 设定/解除反白打印模式

[格式] ASCII码 GS B *n*

十六进制码 1D 42 *n*

十进制码 29 66 *n*

[范围] 0 ≤ *n* ≤ 255

[描述] 设定或解除反白打印模式。

当*n* 的最低有效位为0时，关闭反白模式。

当*n* 的最低有效位为1时，打开反白模式。

[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。

• 反白模式打开时，它对**ESC SP**设定的空白也有效。

• 该命令不影响位图，用户自定义位图，条形码，条码显示字符和由**HT**, **ESC $**, 及**ESC ** 跳过的间距。

• 反白模式优先于下划线模式。选择反白模式时，即使下划线模式打开也被禁止(但不取消)。

[缺省值] *n* = 0

[例子]
```
char SendStr[3];
SendStr[0] = 0x1D;
SendStr[1] = 'B';
SendStr[2] = 1;//反白
PrtSendData( SendStr, 3);
```

4.2.8 ESC V *n OK*

[名称] 设置/解除字符旋转模式

[格式] ASCII码 ESC V *n*

十六进制码 1B 56 *n*

十进制码 27 86 *n*

[范围] 0 ≤ *n* ≤ 3

[描述] 设置/解除字符旋转模式

N（十进制）	意义
0	解除旋转模式
1	设置90°顺时针旋转模式
2	设置180°顺时针旋转模式
3	设置270°顺时针旋转模式

[注意] • 当设置了下划线模式时，对于顺时针90°旋转的字符，打印机不加下划线。

• 在 旋转模式下，倍宽和倍高命令放大字符的方向与一般模式下倍高倍宽命令放大字符的方向相反。

[缺省值] *n* = 0

[参照] **ESC !, ESC –**

[例子]
```
char SendStr[3];
SendStr[0] = 0x1B;
SendStr[1] = 'V';
SendStr[2] = 2;//旋转180度
PrtSendData( SendStr, 3);
```

4.3 打印排版参数设置命令

4.3.1 ESC $ *nL nH*

[名称] 设置绝对打印位置

[格式] ASCII码 ESC $ *nL nH*

十六进制码 1B 24 *nL nH*

十进制码 27 36 *nL nH*

[范围] $0 \le nL \le 255$

$0 \le nH \le 255$

[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>244），则被设置为可打印区域的最大值（N=244）。

[参照] **ESC **

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '$';
SendStr[2] = 15;//0.2×15＝3
PrtSendData( SendStr, 3);    //绝对位置设为距左边界3毫米（15水平点距）
PrtSendData( "从左侧3毫米处开始打印\n", 22);
```

4.3.2 ESC D *n1 n2…nk NULL*

[名称] 设置水平制表位

[格式] ASCII码 ESC D *n1…nk NULL*

十六进制码 1B 44 *n1…nk 00*

十进制码 27 68 *n1…nk 0*

[范围] $1 \le n \le 255$ $0 \le k \le 8$

[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 ! GS !**)的影响。
- 该命令删除了之前设定的水平定位位置。
- 当设置*n* = 8时，通过发送**HT**，打印位置被移到第九列。
- 可以设置8个定位位置(*k* = 8)。超过8定位位置的数据被处理为普通数据。
- 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（6 × 12)的8个字符间隔(列9，17，25，…)。

[参照] **HT**

[例子]
```
char SendStr[16];
char NextPos = 9;
SendStr[0] = 0x1B;
SendStr[1] = 'D';
SendStr[2] = 11;// 距第一列10个字符间距
SendStr[3] = 17;// 距第一列16个字符间距
SendStr[4] = 23;// 距第一列22个字符间距
SendStr[5] = 0;// 结束
PrtSendData(SendStr,6)
PrtSendData("姓名",4);
PrtSendData(&NextPos,1);
PrtSendData("语文",4);
PrtSendData(&NextPos,1);
PrtSendData("数学",4);
PrtSendData(&NextPos,1);
PrtSendData("外语",4);
```

4.3.3 ESC 2

[名称] 选择缺省行间距

[格式] ASCII码 ESC 2

十六进制码 1B 32

十进制码 27 50

[描述] 将当前字符行间距设置为缺省行间距：1mm（3个垂直点距）。

[注意] • 该命令将影响图片与字符之间的行间距。

[参照] **ESC 3**

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '2';
PrtSendData(SendStr,2);
```

4.3.4 ESC 3 *n*

[名称] 设置行间距
[格式] ASCII码 ESC 3 *n*
十六进制码 1B 33 *n*
十进制码 27 51 *n*
[范围] 0 ≤ *n* ≤ 255
[描述] 设置字符行间距为*n*个垂直点距。
[注意] • 该命令将影响图片与字符之间的行间距。
[缺省值] *n* = 8
[参照] **ESC 2**
[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '3';
SendStr[2] = 10;
PrtSendData(SendStr,3);//设置行间距为10个垂直点距(3mm)
```

4.3.5 ESC SP *n*

[名称] 设置字符间距
[格式] ASCII码 ESC SP *n*
十六进制码 1B 20 *n*
十进制码 27 32 *n*
[范围] 0 ≤ *n* ≤ 255
[描述] 设置字符右侧的间距为*n*个水平点距。
[注意] • 在倍宽模式下，字符右侧间距是正常值的两倍。当字符被放大时，字符右侧间距被放大同样的倍数。
• 该命令同时影响英文和汉字字符的设定。
[缺省值] *n* = 0
[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = 0x20;
SendStr[2] = 10;
PrtSendData(SendStr,3);//设置字符间距为10个水平点距(2mm)
```

4.3.6 ESC a *n*

[名称] 选择对齐方式
[格式] ASCII码 ESC a *n*
十六进制码 1B 61 *n*

十进制码 27 97 *n*

[范围] 0 ≤ *n* ≤ 2

[描述] 将一行数据按照n指定的位置对齐。

*n*的可选值及意义：

n	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。

• 该命令在打印区域执行对齐。

• 该命令根据**HT, ESC $**或**ESC **对齐空白区域。

[缺省值] *n* = 0

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = 'a';
SendStr[2] = 1;
PrtSendData(SendStr,3);//设置水平对齐方式为居中
```

4.3.7 GS L *nL nH*

[名称] 设置左边距

[格式] ASCII码 GS L *nL nH*

十六进制码 1D 4C *nL nH*

十进制码 29 76 *nL nH*

[范围] 0 ≤ *nL* ≤ 255 0 ≤ *nH* ≤ 255

[描述] 左边距设置为N个 水平点距。nL nH 分别为无符号双字节整数的低位字节和高位字节，N=nL+nH*256，左边距为可打印区域左边距离打印区域宽度。

[注意] •该命令仅在一行的起始位置处理时有效。

• 左边距最大可设为200，如果超过200，则被当作200。

[缺省值] *nL* = 0, *nH* = 0

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = 'L';
SendStr[2] = 16;
SendStr[3] = 0;
PrtSendData(SendStr,4);//设置左边距为16水平点距(2mm)
```

4.4 图形/图象打印命令

4.4.1 ESC * *m nL nH d1…dk*

[名称] 打印黑白位图

[格式] ASCII码 ESC * *m nL nH d1…dk*

十六进制码 1B 2A *m nL nH d1…dk*

十进制码 27 42 *m nL nH d1…dk*

[范围] *m* = 0, 1, 32, 33

$0 \le nL \le 255$

$0 \le nH \le 1$

$0 \le d \le 255$

[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：

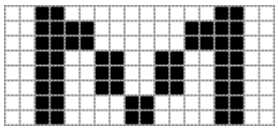
m	垂直点数（高度）	倍宽模式
0	8	两倍宽
1	8	单倍宽
32	24	两倍宽
33	24	单倍宽

nL nH分别为无符号型双字节整数N的高位和低位字节，表示水平方向上位图中的点数。N在单倍宽时最大值为244，在双倍宽时其值最大为122。

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	d9	
0	1	0	0	0	0	0	1	0	b7
0	1	1	0	0	0	1	1	0	b6
0	1	1	0	0	0	1	1	0	b5
0	1	0	1	0	1	0	1	0	b4
0	1	0	1	0	1	0	1	0	b3
0	1	0	1	0	1	0	1	0	b2
0	1	0	0	1	0	0	1	0	b1
0	1	0	0	1	0	0	1	0	b0



打印放大图

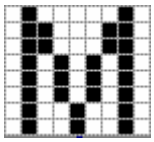
如要打印以上图象，程序代码如下

```
char SendStr[16];
SendStr[0] = 0x1B;
SendStr[1] = '*';
SendStr[2] = 0;//m=0(高度8点、倍宽)
```

```
SendStr[3] = 9;//图象宽度为9dots
SendStr[4] = 0;
SendStr[5] = 0;//图象点阵数据
SendStr[6] = 0xFF;
SendStr[7] = 0x60;
SendStr[8] = 0x1C;
SendStr[9] = 0x03;
SendStr[10] = 0x1C;
SendStr[11] = 0x60;
SendStr[12] = 0xFF;
SendStr[13] = 0;
PrtSendData(SendStr,14);//打印图象
```

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	d9	
0	1	0	0	0	0	0	1	0	b7
0	1	1	0	0	0	1	1	0	b6
0	1	1	0	0	0	1	1	0	b5
0	1	0	1	0	1	0	1	0	b4
0	1	0	1	0	1	0	1	0	b3
0	1	0	1	0	1	0	1	0	b2
0	1	0	0	1	0	0	1	0	b1
0	1	0	0	1	0	0	1	0	b0



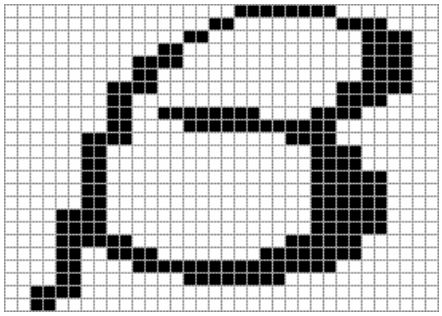
打印放大图

如要打印以上图象，程序代码如下

```
char SendStr[16];
SendStr[0] = 0x1B;
SendStr[1] = '*';
SendStr[2] = 1;//m=1(高度8点、不放大)
SendStr[3] = 9;//图象宽度为9dots
SendStr[4] = 0;
SendStr[5] = 0;//图象点阵数据
SendStr[6] = 0xFF;
SendStr[7] = 0x60;
SendStr[8] = 0x1C;
SendStr[9] = 0x03;
SendStr[10] = 0x1C;
SendStr[11] = 0x60;
SendStr[12] = 0xFF;
SendStr[13] = 0;
PrtSendData(SendStr,14);//打印图象
```

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

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



打印放大图

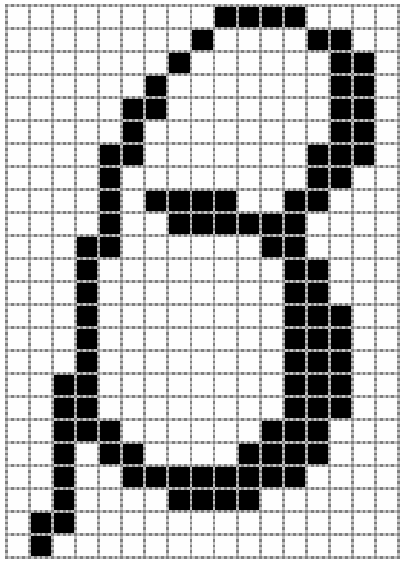
如要打印以上图象，程序代码如下

```
char SendStr[64];
SendStr[0] = 0x1B;
SendStr[1] = '*';
SendStr[2] = 32;//m=32(高度24点、倍宽)
SendStr[3] = 17;//图象宽度为17dots
SendStr[4] = 0;
//图象数据
SendStr[5] = 0x00;   SendStr[6] = 0x00;   SendStr[7] = 0x00;//1
SendStr[8] = 0x00;   SendStr[9] = 0x00;   SendStr[10] = 0x03;//2
SendStr[11] = 0x00;  SendStr[12] = 0x00;  SendStr[13] = 0xFE;//3
SendStr[14] = 0x00;  SendStr[15] = 0x3F;  SendStr[16] = 0xE0;//4
SendStr[17] = 0x03;  SendStr[18] = 0xE0;  SendStr[19] = 0x30;//5
SendStr[20] = 0x0E;  SendStr[21] = 0x00;  SendStr[22] = 0x18;//6
SendStr[23] = 0x11;  SendStr[24] = 0x00;  SendStr[25] = 0x08;//7
SendStr[26] = 0x20;  SendStr[27] = 0xC0;  SendStr[28] = 0x0C;//8
SendStr[29] = 0x40;  SendStr[30] = 0xC0;  SendStr[31] = 0x0C;//9
SendStr[32] = 0x80;  SendStr[33] = 0xC0;  SendStr[34] = 0x0C;//10
SendStr[35] = 0x80;  SendStr[36] = 0x40;  SendStr[37] = 0x1C;//11
```

```
SendStr[38] = 0x80;  SendStr[39] = 0x60;  SendStr[40] = 0x1C;//12
SendStr[41] = 0x80;  SendStr[42] = 0xFF;  SendStr[43] = 0xF8;//13
SendStr[44] = 0x43;  SendStr[45] = 0x9F;  SendStr[46] = 0xF0;//14
SendStr[47] = 0x7F;  SendStr[48] = 0x07;  SendStr[49] = 0xC0;//15
SendStr[50] = 0x3E;  SendStr[51] = 0x00;  SendStr[52] = 0x00;//16
SendStr[53] = 0x00;  SendStr[54] = 0x00;  SendStr[55] = 0x00;//17
PrtSendData(SendStr,56);//打印图象
```

m=33(24点、单倍宽)d1、d2、d3表示打印的第1列点的数据，依此类推；bn表示字节的第n位，

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



打印放大图

如要打印以上图象，程序代码如下

```
char SendStr[64];
SendStr[0] = 0x1B;
SendStr[1] = '*';
SendStr[2] = 33;//m=33(高度24点、不放大)
SendStr[3] = 17;//图象宽度为17dots
SendStr[4] = 0;
//图象数据
SendStr[5] = 0x00;   SendStr[6] = 0x00;   SendStr[7] = 0x00;//1
SendStr[8] = 0x00;   SendStr[9] = 0x00;   SendStr[10] = 0x03;//2
SendStr[11] = 0x00;  SendStr[12] = 0x00;  SendStr[13] = 0xFE;//3
SendStr[14] = 0x00;  SendStr[15] = 0x3F;  SendStr[16] = 0xE0;//4
SendStr[17] = 0x03;  SendStr[18] = 0xE0;  SendStr[19] = 0x30;//5
```

```
SendStr[20] = 0x0E;  SendStr[21] = 0x00;  SendStr[22] = 0x18;//6
SendStr[23] = 0x11;  SendStr[24] = 0x00;  SendStr[25] = 0x08;//7
SendStr[26] = 0x20;  SendStr[27] = 0xC0;  SendStr[28] = 0x0C;//8
SendStr[29] = 0x40;  SendStr[30] = 0xC0;  SendStr[31] = 0x0C;//9
SendStr[32] = 0x80;  SendStr[33] = 0xC0;  SendStr[34] = 0x0C;//10
SendStr[35] = 0x80;  SendStr[36] = 0x40;  SendStr[37] = 0x1C;//11
SendStr[38] = 0x80;  SendStr[39] = 0x60;  SendStr[40] = 0x1C;//12
SendStr[41] = 0x80;  SendStr[42] = 0xFF;  SendStr[43] = 0xF8;//13
SendStr[44] = 0x43;  SendStr[45] = 0x9F;  SendStr[46] = 0xF0;//14
SendStr[47] = 0x7F;  SendStr[48] = 0x07;  SendStr[49] = 0xC0;//15
SendStr[50] = 0x3E;  SendStr[51] = 0x00;  SendStr[52] = 0x00;//16
SendStr[53] = 0x00;  SendStr[54] = 0x00;  SendStr[55] = 0x00;//17
PrtSendData(SendStr,56);//打印图象
```

[注意]
- 如果m的值超出了指定的范围，可能出现不可预料的结果。
- 如果位图数据输入超出了一行上能被打印的点数，那么超出的数据被忽略。
- 在打印一个位图之后，打印机返回常规数据处理模式。
- 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。

4.4.2 GS * *x y d1…dk*

[名称] 定义下传位图

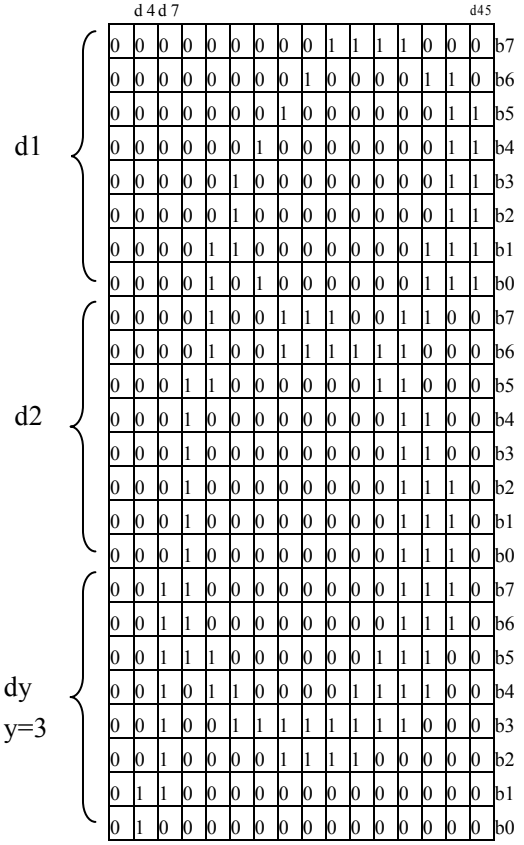
[格式]
ASCII码 `GS * x y d1…dk`
十六进制码 `1D 2A x y d1…dk`
十进制码 `29 42 x y d1…dk`

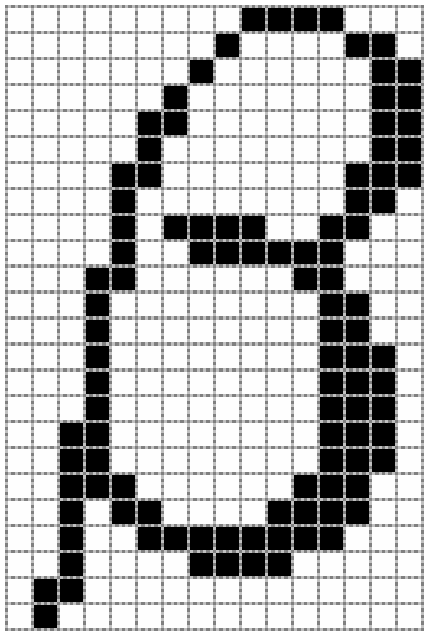
[范围]
$1 \le x \le 255$
$1 \le y \le 8$
$x \times y \le 1536$
$0 \le d \le 255$
k=x*y*8

[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]
- 如果$x * y$ 超出了指定范围，则禁止该命令。
- *d* 表示位图数据。数据(*d*) 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
- 在下列情况下，清除下传位图定义：
  1. 执行**ESC @** 。
  2. 打印机复位或关闭电源。

[例子] 下传位图与打印数据之间关系如下图例子所示。
如果x=2,y=3,d1...dk 的值如下图，则位图如右图所示。





如要定义以上图象，程序代码如下

```
char SendStr[64];
SendStr[0] = 0x1D;
SendStr[1] = '*';
SendStr[2] = 2;//x=2(宽度16点)
SendStr[3] = 3;//y=3(高度24点)
//图象数据
SendStr[4] = 0x00;   SendStr[5] = 0x00;   SendStr[6] = 0x00;//1
SendStr[7] = 0x00;   SendStr[8] = 0x00;   SendStr[9] = 0x03;//2
SendStr[10] = 0x00;  SendStr[11] = 0x00;  SendStr[12] = 0xFE;//3
SendStr[13] = 0x00;  SendStr[14] = 0x3F;  SendStr[15] = 0xE0;//4
SendStr[16] = 0x03;  SendStr[17] = 0xE0;  SendStr[18] = 0x30;//5
SendStr[19] = 0x0E;  SendStr[20] = 0x00;  SendStr[21] = 0x18;//6
SendStr[22] = 0x11;  SendStr[23] = 0x00;  SendStr[24] = 0x08;//7
SendStr[25] = 0x20;  SendStr[26] = 0xC0;  SendStr[27] = 0x0C;//8
SendStr[28] = 0x40;  SendStr[29] = 0xC0;  SendStr[30] = 0x0C;//9
SendStr[31] = 0x80;  SendStr[32] = 0xC0;  SendStr[33] = 0x0C;//10
SendStr[34] = 0x80;  SendStr[35] = 0x40;  SendStr[36] = 0x1C;//11
SendStr[37] = 0x80;  SendStr[38] = 0x60;  SendStr[39] = 0x1C;//12
SendStr[40] = 0x80;  SendStr[41] = 0xFF;  SendStr[42] = 0xF8;//13
SendStr[43] = 0x43;  SendStr[44] = 0x9F;  SendStr[45] = 0xF0;//14
SendStr[46] = 0x7F;  SendStr[47] = 0x07;  SendStr[48] = 0xC0;//15
```

```
SendStr[49] = 0x3E;  SendStr[50] = 0x00;  SendStr[51] = 0x00;//16
PrtSendData(SendStr,52);//定义图象
SendStr[0] = 0x1D;
SendStr[1] = 0x2F;
SendStr[2] = 0x00;
PrtSendData(SendStr,3);//打印图象
```

4.4.3 GS / *n*

[名称] 打印下传位图

[格式] ASCII码 GS / *n*

十六进制码 1D 2F *n*

十进制码 29 47 *n*

[范围] 0 ≤ *n* ≤ 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

[注意]
- 如果位图数据未定义，则忽略该命令。
- 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。
- 如果将要打印的下传位图超出可打印区域，则超出的数据不打印。
- 如果由**GS L**设定的打印区域宽度小于图象宽度，减少左边缘空白量以打印图形。
- 如果位图高度超出64点，则忽略该命令，如果用倍高模式打印，则位图高度不能超过32点。

[参照] **GS ***

[例子] 参见 4.4.2 GS * 例子

4.4.4 FS P *n*

[名称] 打印预存储位图

[格式] ASCII码 FS P *n*

十六进制码 1C 50 *n*

十进制码 28 80 *n*

[范围] 0 ≤ *n* ≤ 7

[描述] 本命令打印预先存储在打印机非易失存储器中的2值位图。打印机非易失存储器中的位图可通过PC机上的专用工具软件生成并写入，位图宽度最大为128，最大高度为64。

*n*为指定的位图编号 。

[注意]
- 指定编号的位图还未定义时，该命令无效。

- 位图必须是2值位图。
- 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。
- 如果要打印的位图宽度超过一行，则超出的部分不打印。
- 需用专用的工具上传打印位图，请参见【三、JPM112打印机工具软件】。通过这种方式上传的位图不会丢失，除非重新上传其他位图将其覆盖。

[参照] **ESC *, GS /**

[例子]
```
char SendStr[4];
SendStr[0] = 0x1C;
SendStr[1] = 'P';
SendStr[2] = 1;
PrtSendData(SendStr,3);//打印编号为1的预存储位图;
```

4.5 曲线打印命令

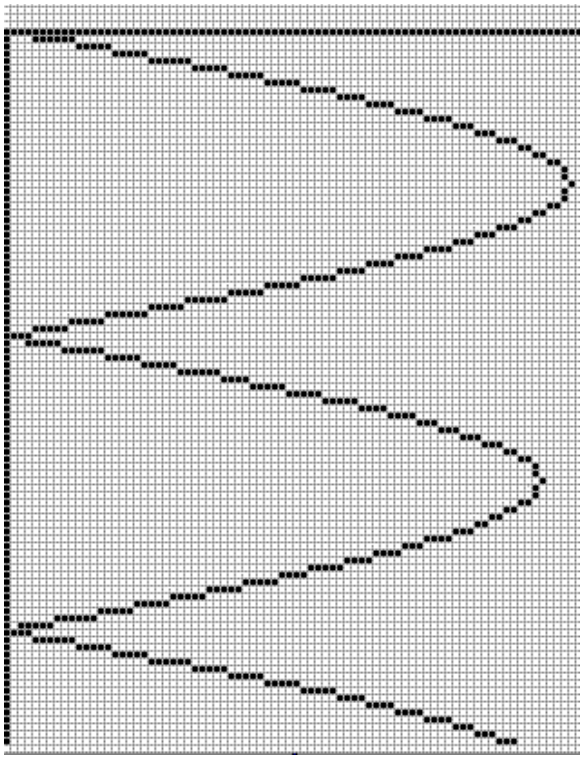
4.5.1 GS ‘

[名称] 打印一水平行上n个线段

[格式]
ASCII码 GS ‘ *n x1sL x1sH x1eL x1eH … xnsL xnsH xneL xneH*
十六进制码 1D 27 *n x1sL x1sH x1eL x1eH … xnsL xnsH xneL xneH*
十进制码 29 39 *n x1sL x1sH x1eL x1eH … xnsL xnsH xneL xneH*

[范围] $0 \le n \le 8$

[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0，最大值为243，也就是说xkeL+xkeH*256最大值为243。

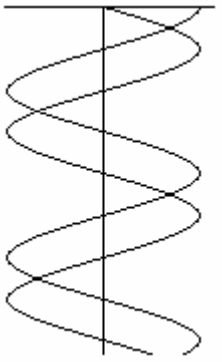
线段的数据不必按照顺序排列；

[注意] • 当打印一个点时，xkeL=xksL，xkeH=xksH。

[参照] **GS “**

[例子] 打印sin,cos函数的连续曲线，打印结果如右图：

```
char SendStr[8];
int i;
short y1,y2,y1s,y2s;
//打印y轴轴线(一条线)
SendStr[0] = 0x1D;
SendStr[1] = 0x27;
SendStr[2] = 1;   //一条线段
SendStr[3] = 30;  //起始点为30
SendStr[4] = 0;
SendStr[5] = 230; //结束点230
SendStr[6] = 1;
PrtSendData(SendStr,7);
//打印曲线
SendStr[0] = 0x1D;
```



打印图例

```
SendStr[1] = 0x27;
SendStr[2] = 3;   //三条线段，分别为x坐标轴，sin函数和cos函数曲线
SendStr[3] = 130; //x坐标轴位置为180
SendStr[4] = 0;
SendStr[5] = 130;
SendStr[6] = 0;
for(i=1;i<1200;i++){
    y1 = sin(i/180*3.1416)*(230-30)/2+130; //计算sin函数坐标点
    y2 = cos(i/180*3.1416)*(230-30)/2+130; //计算cos函数坐标点
    if(i==1) { y1s = y1; y2s = y2; }
    PrtSendData( SendStr, 7 );
    PrtSendData( &y1s, 2 );  //sin函数曲线位于该行的线段的起始点
    PrtSendData( &y1, 2 );   //sin函数曲线位于该行的线段的结束点
    PrtSendData( &y2s, 2 );  //cos函数曲线位于该行的线段的起始点
    PrtSendData( &y2, 2 );   //cos函数曲线位于该行的线段的结束点
    y1s = y1;          //设置下一行打印时sin函数曲线位于该行的起始点坐标
    y2s = y2;          //设置下一行打印时cos函数曲线位于该行的起始点坐标
}
```

4.5.2 GS “

[名称] 打印曲线上的文字

[格式] ASCII码 GS “ *n xL xH c1 c2 … NULL*

十六进制码 1D 22 *n xL xH c1 c2 … 00*

十进制码 29 34 *n xL xH c2 … 0*

[范围] $0 \le n \le 1$

[描述] 该命令按当前字体打印曲线上文字，打印文字时，本命令自动将文字旋转了90度(字符串整体顺时针旋转)；

n 文字编号；

xL xH为字符横向坐标的；

c1 c2 … NULL 为以0结尾的字符串。

[注意] • 只有当该命令出现在两个ESC ‘ 命令之间时，该命令才有效；

• 打印机接收到该命令后，会随后从当前行开始打印旋转90度后的文字；

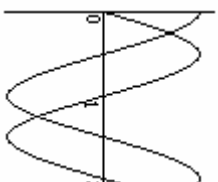
• 当该水平点行上已经有字符时，如果要打印另外的字符，需将文字编号设为另外的值，但只限于0和1；

• 每一水平点行上最多只能出现两个字符。

[参照] **GS ‘**

[例子] 打印sin,cos函数的连续曲线，打印结果如图：

```
char SendStr1[8], SendStr2[16];
int i;
short y1,y2,y1s,y2s;
//打印y轴轴线(一条线)
SendStr1[0] = 0x1D;
```



```
SendStr1[1] = 0x27;
SendStr1[2] = 1;  //一条线段
SendStr1[3] = 30; //起始点为30
SendStr1[4] = 0;
SendStr1[5] = 230;    //结束点230
SendStr1[6] = 1;
//打印坐标原点文字“0”
PrtSendData(SendStr,7);
SendStr2[0] = 0x1D;
SendStr2[1] = 0x22;
SendStr2[2] = 0;  //文字编号0
SendStr2[3] = 110;    //横向坐标为110
SendStr2[4] = 0;
SendStr2[5] = '0';    //字符‘0’
SendStr2[6] = 0;
PrtSendData(SendStr2,7);
//打印曲线
SendStr1[0] = 0x1D;
SendStr1[1] = 0x27;
SendStr1[2] = 3;  //三条线段，分别为x坐标轴，sin函数和cos函数曲线
SendStr1[3] = 130;    //x坐标轴位置为180
SendStr1[4] = 0;
SendStr1[5] = 130;
SendStr1[6] = 0;
//文字
SendStr2[0] = 0x1D;
SendStr2[1] = 0x22;
SendStr2[2] = 0;
SendStr2[3] = 110;
SendStr2[4] = 0;
SendStr2[5] = 0;
SendStr2[6] = 0;

for(i=1;i<1200;i++){
    y1 = sin(i/180*3.1416)*(230-30)/2+130; //计算sin函数坐标点
    y2 = cos(i/180*3.1416)*(230-30)/2+130; //计算cos函数坐标点
    if(i==1) { y1s = y1; y2s = y2; }
    PrtSendData( SendStr1, 7 );
    PrtSendData( &y1s, 2 );  //sin函数曲线位于该行的线段的起始点
    PrtSendData( &y1, 2 );   //sin函数曲线位于该行的线段的结束点
    PrtSendData( &y2s, 2 );  //cos函数曲线位于该行的线段的起始点
    PrtSendData( &y2, 2 );   //cos函数曲线位于该行的线段的结束点
    y1s = y1;          //设置下一行打印时sin函数曲线位于该行的起始点坐标
    y2s = y2;          //设置下一行打印时cos函数曲线位于该行的起始点坐标
```

```
        if(i%180==0) SendStr2[5] = '0'+i/180; //打印x轴标尺
        PrtSendData( SendStr1, 7 );
    }
```

4.6 自定义字符命令

4.6.1 ESC % *n*

[名称] 选择/取消用户自定义字符集

[格式] ASCII码 ESC % *n*

十六进制码 1B 25 *n*

十进制码 27 37 *n*

[范围] 0 ≤ *n* ≤ 255

[描述] 选择或取消用户自定义字符集。

当*n*的最低位(LSB)为0时，取消用户自定义字符集。

当*n*的最低位(LSB)为1时，选择用户自定义字符集。

[注意] • 当取消用户自定义字符集时，自动选择内部字符集。

• *n* 仅最低有效位可用。

[缺省值] *n* = 0

[参照] **ESC &**, **ESC ?**

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '%';
SendStr[2] = 1;
PrtSendData(SendStr,3);//选择用户自定义字符集
```

4.6.2 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)]*

十六进制码 1B 26 *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)]*

[范围] *y* = 2

32 ≤ *c1* ≤ *c2* ≤ 126

x = 8 字体A (8 * 16)

x = 6 字体B (6 * 12)

0 ≤ *d1 ... d(y * xk)* ≤ 255

[描述] 定义用户自定义字符。

• *y* 指定垂直方向字节数，

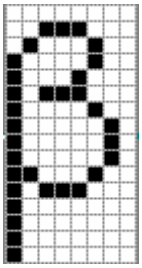
• *c1* 指定起始字符编码，*c2* 指定结束字符编码。

- *x* 指定水平方向点数，
  当定义字体A的字符时x=8，
  当定义字体B的字符时y=6，
- *k=c2-c1+1*

[注意]
- 可定义字符编码的范围：从0x20到0x7E的ASCII 码(95 字符)。
- 可定义多个字符的连续字符编码。当仅需要一个字符时，令*c1 = c2*。
- *d* 是字符的点阵数据。
- 定义用户自定义字符的数据是(*y* * *x*) 字节。
- 设定打印点的相应位为1或不打印点的相应位为0。
- 该命令可对每一种字体定义不同的用户自定义字符模式。用**ESC ！** 设定字体。
- 用户自定义字符和下传位图总数据量不超过32k。
- 在下列情况下，用户自定义字符被清除：
  1．执行**ESC @**。
  2．执行**ESC ?**。
  3．打印机复位或关闭电源。

[例子] y=2,x=8 将字符'B'(编码为0x42)定义为如下12*24点阵字符

d1	0	0	0	0	0	0	0	b7
	0	0	1	1	1	0	0	b6
	0	1	0	0	0	1	0	b5
	1	0	0	0	0	1	0	b4
	1	0	0	0	1	0	0	b3
	1	0	1	1	1	0	0	b2
	1	0	0	0	0	1	0	b1
	1	0	0	0	0	0	1	b0
d2	1	0	0	0	0	0	1	0 b7
	1	0	0	0	0	0	1	0 b6
	1	1	0	0	0	1	0	0 b5
	1	0	1	1	1	0	0	0 b4
	1	0	0	0	0	0	0	0 b3
	1	0	0	0	0	0	0	0 b2
	1	0	0	0	0	0	0	0 b1
	1	0	0	0	0	0	0	0 b0



打印放大图

```
char SendStr[16];
SendStr[0] = 0x1B;
SendStr[1] = '&';
SendStr[2] = 2;   //y=2 字符高度为16
SendStr[3] = 0x42;   // 字符'B'（编码0x42）被重新定义
SendStr[4] = 0x42;
SendStr[5] = 8;   //x=8 字符宽度为8
```

```
SendStr[6] = 0x1F;   SendStr[7] = 0xFF;   //第1列
SendStr[8] = 0x20;   SendStr[9] = 0x20;   //第2列
SendStr[10] = 0x44;  SendStr[11] = 0x10;  //第3列
SendStr[12] = 0x44;  SendStr[13] = 0x10;  //第4列
SendStr[14] = 0x4C;  SendStr[15] = 0x10;  //第5列
SendStr[16] = 0x32;  SendStr[17] = 0x20;  //第6列
SendStr[18] = 0x01;  SendStr[19] = 0xC0;  //第7列
SendStr[20] = 0x00;  SendStr[21] = 0x00;  //第8列
PrtSendData(SendStr,22);//选择用户自定义字符集
```

[缺省值] 内部字符集
[参照] **ESC %, ESC ?, FS 2**

4.6.3 ESC ?

[命令] 清除用户自定义字符
[格式] ASCII码 `ESC ?` *n*
十六进制码 `1B 3F` *n*
十进制码 `27 63` *n*
[范围] $32 \le n \le 126$
[描述] 清除用户自定义字符。
[注意] • 该命令清除指定的用户自定义字符，字符编码由*n* 指定。在用户自定义字符被取消后，以内部字符相应模式打印。
• 如果没有为指定字符代码定义一个用户自定义字符，则打印机忽略该命令。
[参照] **ESC &**, **ESC %**
[例子]

```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '?';
SendStr[2] = 0x42;
PrtSendData(SendStr,3);//清除为0x42定义的字符
```

4.7 汉字命令

4.7.1 FS &

[名称] 设定汉字模式
[格式] ASCII码 `FS &`
十六进制码 `1C 26`
十进制码 `28 38`

[描述] 选择汉字字符模式。

[注意] • 打开电源时，打印机已选择汉字模式。

[参照] **FS .**, **FS C**

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '&';
PrtSendData(SendStr,2);//设定汉字模式
```

4.7.2 FS 2 *c1 c2 d1…dk*

[名称] 定义用户自定义汉字字符

[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* 和*c2* 表示所定义的字符的代码。

c1 = FEH A1H ≤ *c2* ≤ FEH

0 ≤ *d* ≤ 255

k = 32

[描述] 定义用户自定义汉字字符，由*c1* 和*c2* 指定字符代码。

[注意] • *c1* 和*c2* 表示所定义的字符的代码。*c1* 为第一个字节， *c2* 为第二字节。

• *d* 表示该字符的点阵数据。相应位置为1，打出该点，相应位为0，不打该点。

定义数据的格式与**ESC &**指令相同

• 当用户选择字体A时，定义的汉字应为24点阵汉字，选择字体B时，定义的汉字应为16点阵汉字

[缺省值] 全空白。

[参照] **ESC &**

4.7.3 FS .

[名称] 取消汉字字符模式

[格式] ASCII码 FS .

十六进制码 1C 2E

十进制码 28 46

[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。

[注意] • 打开电源时，打印机选择汉字模式。

[参照] **FS &**, **FS C**

[例子]
```
char SendStr[4];
SendStr[0] = 0x1B;
SendStr[1] = '.';
PrtSendData(SendStr,2);//取消汉字模式
```

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	å	87	ç
88	ê	89	ë	8A	è	8B	ï	8C	î	8D	ì	8E	Ä	8F	Å
90	É	91	æ	92	Æ	93	ô	94	ö	95	ò	96	û	97	ù
98	ÿ	99	Ö	9A	Ü	9B	¢	9C	£	9D	¥	9E	Pts	9F	ƒ
A0	á	A1	í	A2	ó	A3	ú	A4	ñ	A5	Ñ	A6	ª	A7	º
A8	¿	A9	⌐	AA	¬	AB	½	AC	¼	AD	¡	AE	«	AF	»
B0	░	B1	▒	B2	▓	B3	|	B4	┤	B5	╡	B6	╢	B7	╖
B8	╕	B9	╣	BA	║	BB	╗	BC	╝	BD	╜	BE	╛	BF	┐
C0	└	C1	┴	C2	┬	C3	├	C4	─	C5	┼	C6	╞	C7	╟
C8	╚	C9	╔	CA	╩	CB	╦	CC	╠	CD	═	CE	╬	CF	╧
D0	╨	D1	╤	D2	╥	D3	╙	D4	╘	D5	╒	D6	╓	D7	╫
D8	╪	D9	┘	DA	┌	DB	█	DC	▄	DD	▌	DE	▐	DF	▀
E0	α	E1	β	E2	Γ	E3	Π	E4	Σ	E5	σ	E6	µ	E7	γ
E8	Φ	E9	θ	EA	Ω	EB	δ	EC	∞	ED	φ	EE	∈	EF	∩
F0	≡	F1	±	F2	≥	F3	≤	F4	⌠	F5	⌡	F6	÷	F7	≈
F8	°	F9	•	FA	·	FB	√	FC	ⁿ	FD	²	FE	▪	FF	

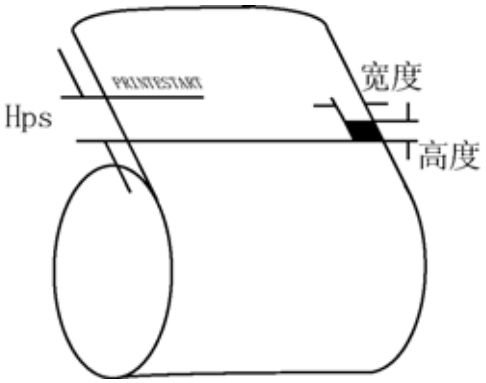
B.预印刷黑标说明

JPM112 打印机采用了卷纸智能定位技术，也就是黑标检测技术，使用该技术可以自动为连续的打印纸分页。

如果要使用该技术，需要在打印纸上合适的地方印刷黑标，以便打印机检测并判断是否到了每页的结尾或开头。

黑标印刷一般都是在打印纸上沿走纸方向距离固定的间隔印刷，所以印刷黑标的位置需要根据用户所打印的单据的长度来确定，除以上因素外，印刷的黑标必须能让 JPM112 能够识别，能让 JPM112 能够识别的黑标必须满足以下要求：

印刷位置：如下图所示，黑标应印刷于文字面的右侧边缘。



宽度范围：宽度≥5mm

高度范围：4mm≤高度≤6mm

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%）

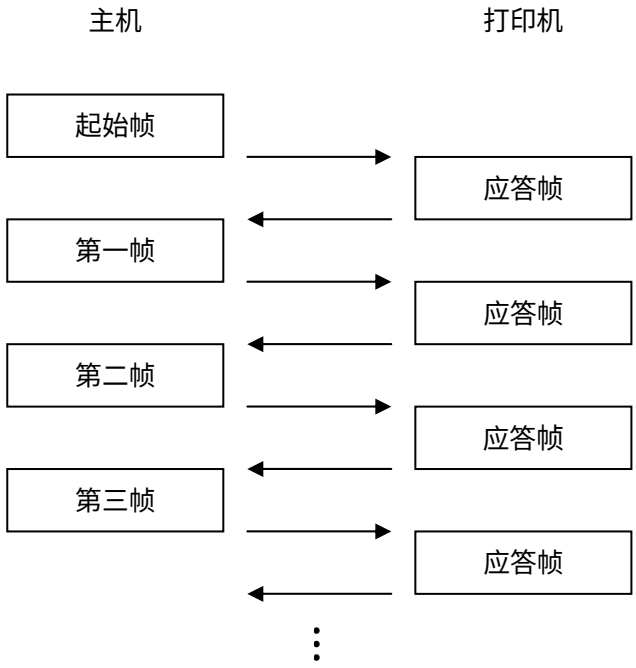
Hps：Hps 为打印机黑标下边缘距打印起始上边缘的距离。

8mm≤Hps≤9mm

C.VIr 协议

为使便携式微型打印机能可靠地与不带 IrDA 协议栈的诸多红外手持终端进行红外数据通讯，开发了 VIr 协议，该协议适用于 JPM112 的打印机。协议框架如下：

主机先向打印机发送起始帧，接收到正确的应答帧后表示连接成功，然后向打印机发送打印数据帧，如果打印机正确接收，会向主机发送表示接收成功的应答帧，主机接收到该帧后才能继续向打印机发送下一帧数据，如此循环，直到主机将打印数据发送完并且接收到最后的应答帧后，结束通讯。



D.1 帧格式

1、主机发送每一帧都由如下几部分组成：

帧起始（SOF）	帧 ID	数据长度	数据	校验和
2 字节(0xAA0x55)	1 字节	2 字节	N 字节(0≤N≤256)	2 字节

每部分定义如下：

帧起始：标识帧开始，总是由两个字节组成，第一字节为 0xAA，第二字节为 0x55；

帧 ID：表示帧的 ID 号，ID 号从 0 开始到 255，如果超过 255，ID 号重新从 0 开始。当帧 ID 为 0 并且帧长度为 0 时，表示该帧为起始帧，打印机接收到起始帧时，将清除打印缓冲区中的内容；

数据长度：由两字节的整数表示，低位在前，高位在后。数据长度不包含校验和的长度。起始帧的数据长度为 0，数据长度最大为 256 字节，超过 256 字节可能会引起不可预知的后果；

数据：打印数据，起始帧的该部分为空；

校验和：校验和用于校验帧传输的正确性，防止误码。校验和的计算方法：在整帧中，除帧起始和校验和外，其他部分的每个字节（unsigned char）相加所得为校验和；

2、当打印机接收到主机帧数据后，打印机会向主机发送回应帧，回应主机帧数据接收是否成功，回应帧格式如下：

帧起始（SOF）	帧 ID	数据长度	回应信息	校验和
2 字节(xAAx55)	1 字节	2 字节(x01x00)	1 字节(x00/xFF)	2 字节

每部分定义如下：

帧起始：标识帧开始，总是由 2 个字节组成，第一字节为 0xAA，第二字节为 0x55；

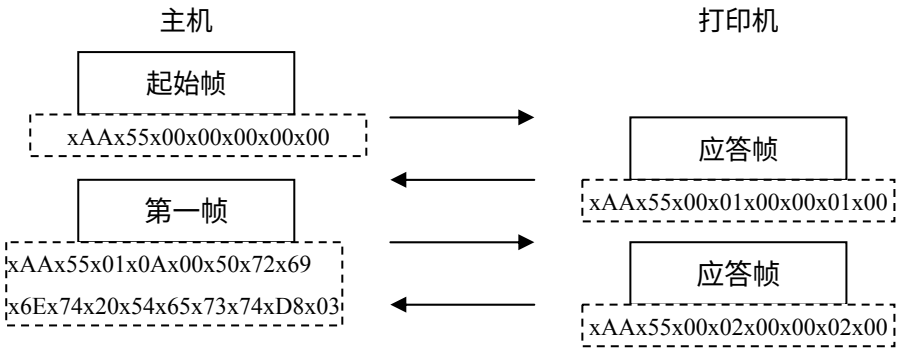
帧 ID：接收到的主机帧的 ID 号；

数据长度：由 2 字节的整数表示，低位在前，高位在后。在回应帧中，该部分总为 1（x01x00）；

回应信息：由 1 个字节表示，0xFF 表示接收失败，0x80 表示打印缓冲区满，0x00 表示接收成功；

校验和：校验和用于校验帧传输的正确性，防止误码。校验和的计算方法：在整帧中，除帧起始和校验和外，其他部分的每个字节（unsigned char）相加所得为校验和。

3、例如：如果主机要向打印机发送“ Print Test ”几个字符，应如下操作：



D.2 异常处理

当主机向打印机发送数据时，会出现误码、掉线等等情况，为避免这些状况对最终数据传输造成影响，主机和打印机需要处理这些异常现象以保证数据传输的正确性。

1、 打印机接收帧错误

当打印机接收到的帧出现错误时，打印机会向主机发送接收错误的应答帧或者不发送应答帧。

当出现以下几种情况时，打印机不发送应答帧：

没有正确的帧起始或帧 ID 号；

没有接收到完整的帧。

当出现以下情况时，打印机会向主机发送接收错误的应答帧：

接收到的校验和与打印机计算出的校验和不同。

2、 主机接收应答帧错误

当主机接收到的应答帧不完整或者无法按照回应帧的格式解析出应答帧的含义时，主机应向打印机重新发送最近一次发出的帧。

3、 主机接收应答帧超时

如果主机向打印机发送数据 500~1000 毫秒后仍然没有接收到打印机发送的应答帧，主机应向打

印机重新发送最近一次发出的帧。

4、 打印缓冲区满

如果打印机接收的主机发送的数据超出了打印缓冲区的大小，打印机会向主机发送表示打印缓冲区满的应答帧，此时打印机并没有接收到主机此前打印机发送的该帧数据。当主机接收到表示打印缓冲区满的应答帧时，应该等待 500～1000 毫秒后再次向打印机发送该帧数据。如果仍然收到表示打印缓冲区满的应答帧，则主机应继续等待 500～1000 毫秒后再向打印机发送该帧数据。

D.3 主机 VIr 协议软件流程