

RD-EH

嵌入式热敏打印机 开发手册

版权所有：北京荣达创新科技有限公司

目 录

| | |
|------------------------|----|
| 一、概 述..... | 3 |
| 二、按键及操作..... | 4 |
| 2.1 按 键..... | 4 |
| 2.2 自检..... | 4 |
| 2.3 上纸..... | 4 |
| 2.4 走纸..... | 4 |
| 2.5 指示灯..... | 4 |
| 三、接口的连接及作用..... | 4 |
| 3.1 串行接口（SN）机型详解..... | 4 |
| 3.1.1 数据接口..... | 4 |
| 3.1.2 串口数据发送方法..... | 5 |
| 3.1.3 波特率选择位..... | 6 |
| 3.2 并行接口（-8+）机型详解..... | 6 |
| 3.2.1 数据接口..... | 6 |
| 3.2.2 并口数据发送方法..... | 7 |
| 3.3 电源接口..... | 7 |
| 四、打印命令详解..... | 9 |
| 4.1 指令表..... | 9 |
| 4.2 指令详解..... | 11 |
| 4.2.1 汉字打印命令..... | 11 |
| 4.2.2 纸进给命令..... | 11 |
| 4.2.3 格式设置命令..... | 13 |
| 4.2.4 字符设置命令..... | 15 |
| 4.2.5 用户定义字符设置命令..... | 21 |
| 4.2.6 初始化命令..... | 25 |
| 4.2.7 数据控制命令..... | 25 |
| 五、安装尺寸..... | 27 |
| 5.1 开口尺寸..... | 27 |
| 5.2 外形尺寸..... | 27 |
| 5.3 外观示图..... | 28 |
| 附 录..... | 29 |
| A ASCII 字符集..... | 29 |
| B 字符集 1..... | 29 |
| C 字符集 2..... | 30 |



一、概述

RD-EH 系列打印机是一款可嵌入到客户系统中的打印机，该机操作方便、结构精巧、尺寸超小。采用易上纸结构，上纸方便，简单；该系列打印机采用全封闭设计，内嵌控制板。是嵌入式打印的最佳选择机型。

该系列打印机有 4 种型号，分别为：EH32-SN_16；EH32-8+_16；EH32-SN_24；EH32-8+_16。SN 表示为串口，-8+表示为并口；16 表示该机采用 16 点阵字库；24 表示该机采用 24 点阵字库。

其具体技术参数如下：

1) 打印方式：热敏加热点阵打印

2) 打印速度：30~62 毫米/秒

3) 每行点数：384 点

4) 打印字符：

24 点阵机型：ASCII 字符集：12×24 点阵；字符集 1：6×8 点阵

16 点阵机型：ASCII 字符集：8×12，8×16 点阵；字符集 1,2：6×8 点阵

5) 打印汉字：

24 点阵机型：24×24 点阵国际一、二级汉字库

16 点阵机型：16×16，12×12 点阵国际一、二级汉字库

6) 纸张类型：热敏卷纸，57mm×30mm

7) 换纸方式：自动上纸

8) 缓存器：32K

9) 外 接 口：

串口机型：RS232 串行接口、可选 TTL 串口，485 接口

并口机型：标准并口，可通过 W1 短接实现简单的 TTL 串口

10) 工作电源：DC5V/3A

11) 工作温度：0~50℃

12) 工作湿度：0~80%

二、按键及操作

2.1 按键

RD-EH-SN 打印机上有一个按键，平时做为走纸键用，在打印机刚上电时可做为自检键来使用。

2.2 自检

在未加电时按住键盘按键，通电，约 2 秒钟，打印机进行自检。自检会打印出机器的型号，电话，公司名称，接口参数等基本信息。

2.3 上纸

慢慢提起打印机前盖的拉手，拉开纸仓盖，将打印纸卷放入纸仓，露出纸头，将纸光滑面朝内，关好纸仓门，使其压住打印纸露出端即可。

2.4 走纸

在打印机带电状态下，按住按键，则打印机开始走纸，松开按键，则打印机停止走纸。

2.5 指示灯

RD-EH-SN 打印机上有一个为绿色指示灯。该指示灯可指示缺纸及在线两种状态。当缺纸时，绿色指示灯处于长灭状态，在线状态下，处于长亮状态。当打印机的缓冲区满或打印机忙时，绿色的指示灯将熄灭(或为闪烁状态)，此时打印机将不能接收数据及指令。

三、接口的连接及作用

3.1 串行接口 (SN) 机型详解

EH 串行接口机型通讯接口采用 10 芯双排针 (图 3-1)，针与针之间的间距为 2mm。该机出厂默认通讯的方式为：9600，8，N，CTS。即通讯的速度为 9600BPS；8 位数据位，1 位停止位；无效验位；硬件流控方式。

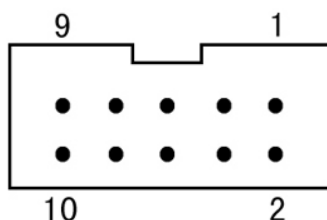


图 3-1

3.1.1 数据接口

| 5 芯排座 | 信号 | 信号来源 | 方向 | 说明 | DB-9 芯串口线 |
|-------|-----|------|----|---|-----------|
| 5 | TXD | 主机 | 输入 | 打印机从主计算机接收数据。 | 3 |
| 3 | 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。 | 2 |
| 7 | 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。 | 8 |
| 9 | GND | -- | -- | 电源地 | 5 |

注: ①信号来源一项中的“打印机”和“主机”表示信号发出的来源。

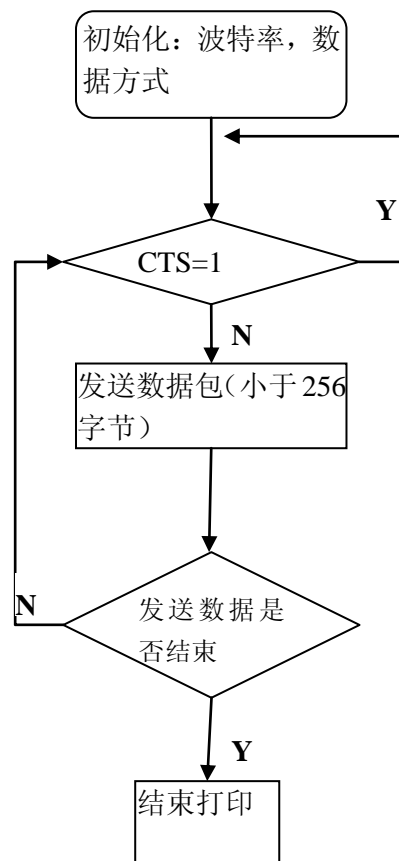
②信号逻辑电平为 RS232 电平。

3.1.2 串口数据发送方法

由于打印机上配有32K字节的缓存,当一次发送数据少于32K时,可直接发送数,发送方法为:



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



3.1.3 波特率选择位

| SW NO. | 19200 | 9600 | 4800 | 2400 | 1200 | 600 | 300 | 150 |
|--------|-------|------|------|------|------|-----|-----|-----|
| 1 | off | on | off | on | off | on | off | on |
| 2 | off | off | on | on | off | off | on | on |
| 3 | off | off | off | off | on | on | on | on |

3.2 并行接口 (-8+) 机型详解

EH 并行接口机型采用 20P 双排针做为并口的通讯接口，该接口针与针之间的间距为 2.0mm。该型号的机器通过短接控制板上的 W1 可以实现 TTL 的串行通信。图 3-3 接并口示意图。

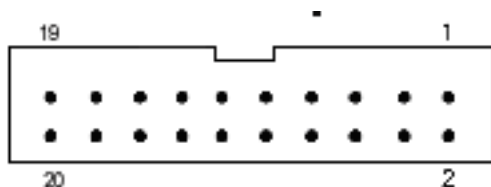


图 3-3

3.2.1 数据接口

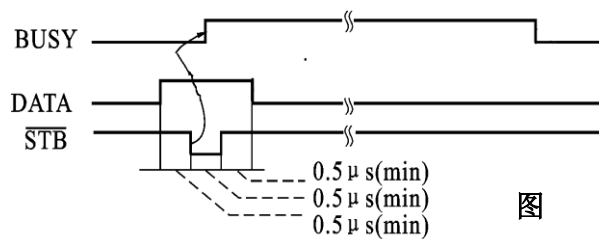
| 面板 20 线 并口引脚号 | 信号 | 方向 | 说 明 | PC 机 25 芯并口线 |
|-----------------------|-------|-----|---|--------------|
| 1 | —STB | 入 | 数据选通触发脉冲，上升沿时读入数据 | 1 |
| 3 | DATA1 | 入 | 这些信号分别代表并行数据的第一位至第八位信号。每个信号当其逻辑为“1”时为“高”电平，逻辑为“0”时为低电平。 | 2 |
| 5 | DATA2 | 入 | | 3 |
| 7 | DATA3 | 入 | | 4 |
| 9 | DATA4 | 入 | | 5 |
| 11 | DATA5 | 入 | | 6 |
| 13 | DATA6 | 入 | | 7 |
| 15 | DATA7 | 入 | | 8 |
| 17 | DATA8 | 入 | | 9 |
| 20 | —ACK | 出 | 回答脉冲，“低”电平表示数据已被接受 当 W1 短接时，做为 TTL 串口的数据引脚 | 10 |
| 18 | BUSY | 出 | “高”电平表示打印机正忙不接收数据 当 W1 短接时，做为 TTL 的 BUSY 引脚 | 11 |
| 19 | PE | — — | 接地 | — — |
| 4 | SEL | 出 | 经电阻上拉“高”表示打印机在线 | 13 |
| 2、6、8、10、 12、14、16 | GND | — — | 接地 | 25, 12 |

注：①信号来源一项中的“打印机”和“主机”表示信入发出的来源。

②信号逻辑电平为 EIA 电平。

3.2.2 并口数据发送方法

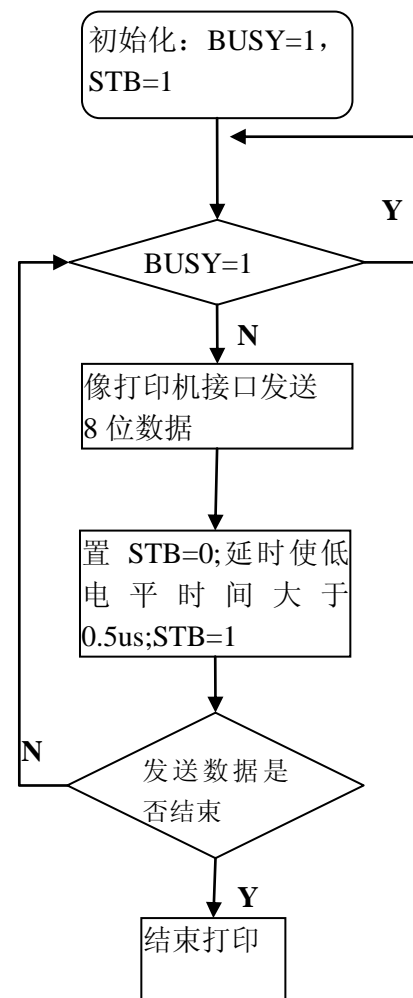
并口数据的发送相对来说比串口要麻烦些，需要 STB, BUSY 及 DATA 数据线之间的时序配合，才可以发送，图 3-4 为并口发送一字节的时序图。



图

发送的步骤为：

- 1) 将打印机与上位机接按“3.2.1 数据接口”所示用数据线连好。
- 2) 给打印机上电，并上好打印纸。
- 3) 开始发送数据，流程图如右图



3.3 电源接口

电源线插座为 3 芯插座。引脚定义为：1 脚 VCC (5V)；2 脚 空脚；3 脚 GND。电流 $\cong 3A$ 。
图 3-5 为电源接口示意图。

注：电源可根据客户的要求选择使用直流：8.5V 供电或 12V~36V 供电。

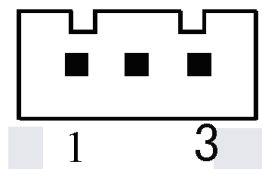


图 3-5

四、打印命令详解

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

4.1 指令表

| 命令 | 功能 |
|-------------|--------------|
| NUL | 结束标志 |
| HT | 执行水平造表 |
| LF | 换行 |
| CR | 回车 |
| ESC SP n | 设置字间距 |
| ESC % | 替换用户定义字符 |
| ESC & | 用户自定义字符 |
| ESC ' M | 打印 M 个曲线点 |
| ESC + n | 允许/标止上划线打印 |
| ESC - n | 允许/标止下划线打印 |
| ESC 1 n | 设定行间距为 n 点行 |
| ESC 6 | 选择字符集 1 |
| ESC 7 | 选择字符集 2 |
| ESC 8 n | 选择不同点阵汉字打印 |
| ESC : | 恢复字符集中的字符 |
| ESC @ | 初始化打印机 |
| ESC D | 设置水平造表值 |
| ESC E | 打印条型码 |
| ESC J n | 执行 n 点行走纸 |
| ESC K | 打印 n1×8 点阵图形 |
| ESC Q n | 设定右限宽度 |
| ESC U n | 横向放大 n 倍 |
| ESC V n | 纵向放大 n 倍 |
| ESC W n | 横向纵向放大 n 倍 |
| ESC X n1 n2 | 横向纵向放大不同倍数 |
| ESC C n | 允许/禁止反向打印 |

| | |
|----------------------|---------------|
| ESC f M n | 打印空格或换行 |
| ESC i n | 允许/禁止反白打印 |
| ESC l n | 设定左限宽度 |
| ESC r 2B/2D n | 热敏打印深度调整 |
| FS 2 n | 设置字符旋转打印 |
| FS J | 设置纵向打印 |
| FS K | 设置横向打印 |
| FS r n | 选择上下标 |
| FS 8 | 选择 12X24ASCII |

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 `print()`、`print_chs()` 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：

```
Void print(unsigned char ch) //单字节发数函数
```

```
Void print_chs(unsigned char ch[]) //多字节发送函数
```

4.2 指令详解

4.2.1 汉字打印命令

ESC 8 n

[名称] 选择不同点阵汉字打印

[格式] ASCII: ESC 8 n
 十进制: 27 56 n
 十六进制 1B 38 n

[范围] $0 \leq n \leq 7$

[说明]

打印机在接收该命令之后将根据 n 值选择不同点阵的汉字。在汉字打印方式中, 打印机接收的汉字代码是 2 字节对应一个汉字的标准机内码, 即打印机每接收 2 个字节的机内码可调出一个汉字。打印机先接收机内码的高位字节, 再接收低位字节。

当 n=0 选择 16×16 点阵汉字打印

当 n=1 选择 8×16 点阵汉字打印

当 n=2 选择 16×8 点阵汉字打印

当 n=3 选择 8×8 点阵汉字打印

当 n=4 选择 12×12 点阵汉字印

当 n=5 选择 6×12 点阵汉字打印

当 n=6 选择 8×16 点阵 ASCII 字符打印

当 n=7 选择 8×12 点阵 ASCII 字符打印

汉字代码为标准汉字内码, 可通过网上下载汉字内码表来查找汉字内码:

1) 高字节数值范围 A1H-F7H, 对应 1-87 区汉字, 计算方法: 区码+A0H

2) 低字节数值范围 A1H-FEH, 对应汉字位码 1-94, 计算方法: 位码+A0H

例: “荣”字的区位码是 4057, 即 40 区, 第 57 个字, 其机内码为 C8D9

将 40 (十进制) 转为 28 (十六进制), 28H+A0H=C8H

将 57 (十进制) 转为 39 (十六进制), 39H+A0H=D9H

当输入代码为 20H-A0H 时, 自动选择字符集 (见附录 1)

当输入代码大于 A0H 时, 如果下一字节小于 A1H, 则选择国际标准 ASCII 码, 否则打印汉字。

[注意]

1) 24 点阵汉字库的机型 n 无 1~7 选项, 字符为 12×24 点阵 ASCII 码, 汉字命令为: 1B 38 或 1B 38 0

2) 16 点阵汉字库的机型 n 的默认值为 0。

[例子]

如向打印机发送如下数据:

```
print(0x1b);print(0x38);print(0x00); //调用 16*16 字库  
print(0xc8);print(0xd9);           //打印“荣”内码  
print(0x0d);                        //回车
```

则会在打印机上打印“荣”字出来。

4.2.2 纸进给命令

LF



[名称] 换行

[格式] ASCII: LF

十进制: 10

十六进制: 0A

[说明]

打印机打印当前缓冲器里的数据并向前走纸一行，如果缓冲器里没有数据，则只执行向前走纸一行。针打一行为8+行间距，热敏一行为16+行间距。

ESC J

[名称] 执行 n 点行走纸

[格式] ASCII: ESC J n

十进制: 27 74 n

十六进制: 1B 4A n

[说明]

打印纸向前进给 n 点行，n 的值应在 1~255 点之间。这个命令不发生回车换行，它也不影响后面的换行命令。如果你需要产生立即进纸走纸而不要回车，就可以使用本命令。

如果打印机缓冲器内有未打印的数据，则在走纸前先打印这些数据。

ESC 1

[名称] 设置 n 点行间距

[格式] ASCII: ESC 1 n

十进制: 27 49 n

十六进制: 1B 31 n

[说明]

为换行命令设置 n 点行间距，n 的值应在 0~255 之间。默认值 n=3。

[例子]

观察这个命令效果的 C 程序如下：

```
for(i=1;i<11;i=i+2)
{
    print(0x1b);print(0x31);print(i);           //ESC 1, 设置行间距
    print_chs("LINE SPACING");                 //打印字符串
    print(0x0a);                               //换行
}
}
```

[打印结果]

```
LINE SPACING
LINE SPACING
LINE SPACING
LINE SPACING
LINE SPACING
LINE SPACING
```

ESC SP



- [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
- [说明] 设置字符之间的空白点数, 即打印完一字符打印机自动在字符右侧加入的空白点数。的值应在 0~128 之间。默认值为 n=0。

4.2.3 格式设置命令

ESC D

- [名称] 设置水平造表值
- [格式] ASCII: ESC D n1 n2 NUL
 十进制: 27 68 n1 n2 0
 十六进制: 1B 44 n1 n2 00
- [说明] 输入水平造表位置 n1, n2 等, 所有这些都应在该型号打印机的行宽之内。字符 NUL 加在最后, 表示该命令的结束。所有设置的水平造表值都可以用该命令的 ESC D NUL 的格式清除。HT 命令执行水平造表。

- [注意] 水平造表所指的位置定义为在一行内所要打印的字符位置。而每个字符是: 针打 6+字间距, 24 点阵热敏 8+字间距, 16 点阵热敏 12+字间距。

- [例子] 在一行的第 2, 第 9 和第 14 字符的位置设置三个水平造表值, 你可以发送下列命令:
 关于这个例子的 C 程序如下:

```
print_chs("1234567890123456");// 标尺
print(0x0d);
print(0x1b);print(0x44);print(2);print(9);print(14);print(0);//ESC D
print(9); //HT 命令
print_chs("HT1"); //打印字符串
print(9); //HT 命令
print_chs("HT2"); //打印字符串
print(9); //HT 命令
print_chs("HT3"); //打印字符串
print(0x0d);
```

- [打印结果]

```
HT1    HT2   HT3
1234567890123456
```

HT

- [名称] 执行水平造表
- [格式] ASCII: HT
 十进制: 9
 十六进制: 09

- [说明]

打印位置进行到由 ESC D 命令设置的下一水平造表位置。

如果没有设置水平造表值或当前打印位置等于或超过最后一个水平造表, HT 命令都将不执行。

ESC f

[名称] 打印空格或空行

[格式] ASCII: ESC f M n
十进制: 27 102 M n
十六进制: 1B 66 M n

[说明]

如果 M=0, ESC f NUL n 命令将打印 n 个空格。每个空格的宽度按 6+字间距 (热敏按 12+字间距) 计算, n 的值应在该型号打印机的行宽之内。

如果 M=1, ESC f SOH n 命令将打印 n 行空行。每个空行的高度针打按 8+行间距, 热敏按 16+行间距计算, n 的值应在 0~255 之间。

[例子]

1) 在一行中打印 6 个空格字符, 命令如下:

```
ASCII: ESC  f  NUL  ACK
十进制: 27  102  0    6
十六进制: 1B  66  00  06
print_chs("TE");           //打印字符串
print(0x1b);print(0x66);print(0);print(0x06); //空六格
print_chs("ST");           //打印字符串
print(0x0d);               //回车
```

2) 要打印 6 行空行, 命令如下:

```
ASCII: ESC  f  SOH  ACK
十进制: 27  102  1    6
十六进制: 1B  66  01  06
print_chs("TE");           //打印字符串
print(0x1b);print(0x66);print(0x01);print(0x06); //6 个回车
print_chs("ST");           //打印字符串
print(0x0d);               //回车
```

ESC Q

[名称] 设置右限

[格式] ASCII: ESC Q n
十进制: 27 81 n
十六进制: 1B 51 n

[说明]

右限即打印纸右侧不打印的字符数, 每个字符的宽度按 6+字间距 (热敏按 12+字间距) 计算。n 的值应在 0 到该型号打印机的行宽之内。默认值 n=0, 即没有右限。

该命令设置的是绝对位置, 不受字符放大命令 ESC U 和 ESC W 的影响。该命令设置之后, 只要达到右限位置, 打印机会自动加入回车换行。

[例子] 要设置右限值为 6, 命令如下:

```
ASCII: ESC  Q  ACK
十进制: 27  81  6
```



十六进制: 1B 51 06

该例子的 C 程序如下:

```
print_chs("1234567890123456");           //打印字符串
print(0x1b);print(0x51);print(0x06);      // 设置右限为 6
print_chs("12345678901234567890");//打印字符串
print(0x0d);                               //回车
```

[打印结果]

```
1234567890
1234567890
1234567890123456
```

ESC I

[名称] 设置左限

[格式] ASCII: ESC 1 n
十进制: 27 108 n
十六进制: 1B 6C n

[说明]

左限即打印纸左侧不打印的字符数, 每个字符的宽度按 6+字间距(热敏按 12+字间距)计算。N 的数值应在 0 到该型号打印机的行宽之内。默认值 n=0, 即没有左限, 该命令设置的是绝对位置, 不受字符放大命令 ESC U 和 ESC W 的影响。

[例子]

要设置左限值为 6, 你可以发送下列命令:

```
ASCII: ESC 1 ACK
十进制: 27 108 6
十六进制: 1B 6C 06
```

该例子的 C 程序如下:

```
print_chs("1234567890123456");           //打印字符串
print(0x1b);print(0x6c);print(0x06);      //左限为 6
print_chs("12345678901234567890");//打印字符串
print(0x0d);                               //回车
```

[打印结果]

```
1234567890
1234567890
1234567890123456
```

4.2.4 字符设置命令

ESC 6

[名称] 选择字符集 1

[格式] ASCII: ESC 6
十进制: 27 54
十六进制: 1B 36

[说明]

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 1), 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)。

ESC 7

[名称] 选择字符集 2



[格式] ASCII: ESC 7
 十进制: 27 55
 十六进制: 1B 37

[说明]

在该命令输入之后的所有字符均使用字符集 2 中的字符打印（见符录 1），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)。

[注意]

24 点阵的热敏机型无字符集 2

FS 8

[名称] 选择 12*24 ASCII 集

[格式] ASCII: FS 8
 十进制: 28 56
 十六进制: 1C 38

[说明]

该命令仅适用于 24 点阵热敏机型，输入该命令之后将打印 12X24 点阵 ASCII 集字符（见符录 1），代码范围 20H~FFH(32~255)。

ESC U

[名称] 横向放大

[格式] ASCII: ESC U n
 十进制: 27 85 n
 十六进制: 1B 55 n

[说明]

在该命令之后的字符将以正常宽度的 n 倍打印。针打 n 的值在 1~8 之间，默认值：n=1。既正常宽度，无放大。热敏 n 的值在 0~8 之间，默认值：n=1。即横向放大 1 倍。

为了观察 ESC U 命令的放大效果，请看下面的 C 程序：

```
for(i=1;i<4;i++)                    //从 1 到 3 倍
{
    print (0x1b);print(0x55);print(i); // ESC U 命令
    print_chs("RD");                 //打印字符
}
print(0x0d);                         //回车命令
```

[打印结果]

RD RD RD

ESC V

[名称] 纵向放大

[格式] ASCII: ESC V n
 十进制: 27 86 n
 十六进制: 1B 56 n

[说明]

在该命令输入之后的字符将以正常高度的 n 倍打印。针打 n 的值在 1~8 之间,默认值：n=1，既

无纵向放大。热敏 n 的值在 0~8 之间,默认值: $n=1$, 既纵向放大 1 倍。这个命令应当在一行的开始发出。

[例子]

```
for(i=1;i<4;i++)          //从 1 到 3 倍
{
    print (0x1b);print(0x56);print(i);    //ESC V 命令
    print_chs("RD");                      //打印字符
    print(0x0d);                          //回车
}
```

[打印结果]

```
RD
RD
RD
```

ESC W

[名称] 横向纵向放大

[格式] ASCII: ESC W n
十进制: 27 87 n
十六进制: 1B 57 n

[说明]

在该命令之后的字符将以正常宽度和高度的 n 倍打印。针打 n 的值在 1~8 之间,默认值: $n=1$, 即无横向纵向放大。热敏 n 的值在 0~8 之间,默认值: $n=1$, 即横向纵向放大 1 倍。

[例子]

```
for(i=1;i<4;i++)          //从 1 到 3 倍
{
    print (0x1b);print(0x57);print(i);    //ESC W 命令
    print_chs("RD");                      //打印字符
    print(0x0d);                          //回车
}
```

[打印结果]

```
RD
RD
RD
```

ESC X

[名称] 横向纵向放大不同倍数

[格式] ASCII: ESC X n1 n2
十进制: 27 88 n1 n2
十六进制: 1B 58 n1 n2

[说明]

在该命令之后的字符将以正常宽度的 $n1$ 倍和正常高度的 $n2$ 倍进行打印。 $n1, n2$ 的值在 1 到 8

之间。默认值 $n_1=n_2=1$ ，即无放大。

ESC r

[名称] 热敏打印深度调整

[格式] ASCII: ESC r 2B/2D n
十进制: 27 114 43/45 n
十六进制: 1B 72 2B/2D n

[说明]

n 的值在 0 ~15 之间;

当使用 1B 72 2B n 为加强深度调节, n 值越大打印深度越深。

当使用 1B 72 2D n 为减少深度调节, n 值越大打印深度越浅。

恢复指令为 1B 72 2D 0 或 1B 72 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。

ESC —

[名称] 允许/禁止下划线打印

[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, 禁止下划线打印。上电或初始化后 n=0。允许下划线打印之后的所有字符、汉字包括空格都将打印出下划线。

[例子]

```
print(0x1b);print(0x57);print(2);        //放大两倍
print_chs("RD");                         //打印字符串
print(0x1b);print(0x2d);print(1);        //允许下划线打印
print_chs("UPT");                         //UPT 打印下划线
print(0x1b);print(0x2d);print(0);        //禁止下划线打印
print_chs("RD");print(0x0d);             //打印字符串
print(0x0d);                             //回车
```

[打印结果]

RDUPTRD

ESC +

[名称] 允许/禁止上划线打印

[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, 禁止上划线打印。上电或初始化后 n=0。允许上划线打印后的所有字符、汉字包括空格都将打印出上划线。

[例子]

```
print(0x1b);print(0x57);print(2);
print_chs("RD");
print(0x1b);print(0x2b);print(1); //允许上打印线打印
print_chs("UPT"); //UPT 上打印上划线
print(0x1b);print(0x2b);print(0); //禁止上打印线打印
print_chs("RD");print(0x0d);
```

[打印结果]

RD UPT RD

FS r

[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

[说明]

该命令可以选择上下标的位置；n=0 是上标，n=1 时是下标。所谓上(下)标是指当某字符的高度小于本行最大字符的高度时，该字符与最大字符是上(下)对齐的。

[例子]

```
print(0x1b);print(0x57);print(0x02); //放大两倍
print('M'); //打印字符'M'
print(0x1b);print(0x57);print(0x01); //原倍
print(0x1c);print(0x72);print(0); //上标命令
print('2'); //打印字符'2'
print(0x0d); //回车
print(0x1b);print(0x57);print(0x01); //原倍
print(0x1b);print(0x57);print(0x02); //放大两倍
print('M'); //打印字符'M'
print(0x1c);print(0x72);print(1); //上标命令
print(0x1b);print(0x57);print(0x01); //原倍
print('2'); //打印字符'2'
print(0x0d); //回车
```

[打印结果]

M₂
M²

ESC i

[名称] 允许/禁止反白打印

[格式] ASCII: ESC i n
十进制: 27 105 n
十六进制: 1B 69 n

[说明]

n=1 允许反白打印，n=0 禁止反白打印。反白打印是在黑色背景下打印白字。正常打印是在白色背景下打印黑字。

当某行最后一个字符是反白字符时，本行与下一行之间的空白由黑线填充。上电或初始化后 n=0。

[例子]

```
print(0x1b);print(0x69);print(1);           //允许反白
print_chs("ABCDEFGHJKLMNO"); //打印字符串
print(0x0d);                               //回车
```

[打印效果]



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

[说明]

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。通常在打印机垂直安装时，会使用反向打印方式，以便观察打印结果。面板式上电或初始化后 n=1。台式打印方向正好相反。

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。（详见 ESC K 命令）

[例子]

```
print(0x1b);print(0x63);print(1);           //允许反向打印
print_chs("RONGDA");                       //打印字符
print(0x0d);                               //回车
```

[打印结果]



```
print(0x1b);print(0x63);print(0);           //禁止反向打印
print_chs("RONGDA");                       //打印字符
print(0x0d);                               //回车
打印结果:
```



FS J

[名称] 设置纵向打印

[格式] ASCII: FS J
十进制: 28 74
十六进制: 1C 4A

[说明]

该命令设置字符为左旋字符，即纵向打印。打印方向是由左向右。

[例子]

```
print(0x1c);print(0x4a);                   //设置纵向打印
print_chs("RONGDA");                       //打印字符
print(0x0d);                               //回车
```

[打印结果]

RONGDA

FS K

[名称] 设置横向打印

[格式] ASCII: FS K
十进制: 28 75
十六进制: 1C 4B

[说明]

该命令设置字符为正向字符，即横向打印。打印方向是由左向右。

[例子]

```
print(0x1C);print(0x4B);           //设置横向打印
print_chs("RONGDA");              //打印字符
print(0x0d);                       //回车
```

[打印结果]

RONGDA

FS 2

[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

[说明]

该命令可以旋转字符，n 的数据如下：

| n | 逆时针旋转 | 打印方向 |
|---|-----------|------|
| 0 | 0 度(正向) | 由左向右 |
| 1 | 90 度(左旋) | 由左向右 |
| 2 | 180 度(反向) | 由右向左 |
| 3 | 270 度(左旋) | 由右向左 |

[注意]

一行打印中只能选择其中一种

4.2.5 用户定义字符设置命令

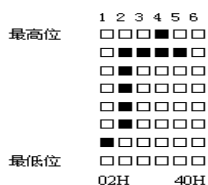
ESC &

[名称] 定义用户自定义字符

[格式] ASCII: ESC & m n1 n2...n6
十进制: 27 38 m n1 n2...n6
十六进制: 1B 26 m n1 n2...n6

[说明]

该命令允许用户定义一个字符。m 是该用户定义字符码，应在 32 到 255 之间。n1, n2, ...n6 是定义这个字符的结构码。字符是由 6×8 点阵组成。共 6 列每列 8 点，每一列由一个字节的 数据表示，最高位在上。



n1=02H, n2=7CH, n3=40H, n4=COH, n5=40H, n6=0H

用户定义的字符储存在 RAM 内, 断电后将会丢失。

如果许多 ESC U 命令使用同一 m 值, 只有最后一个有效。最多可以定义 32 个用户定义字符

ESC %

[名称] 替换用户定义字符

[格式] ASCII: ESC % m1 n1 m2 n2...mk nk NUL
 十进制: 27 37 m1 n1 m2 n2...mk nk 0
 十六进制: 1B 25 m1 n1 m2 n2...mk nk 0

[说明]

该命令可以将当前字符集中的字符 n 替换为用户定义字符 m。在该命令以后的用户定义字符 m 将会代替当前字符集中的字符 n 打印出来。

m1, m2...mk 是用户定义的字符码

n1, n2...nk 是当前字符集中要被替换的字符码

m 和 n 的数值都应在 32 到 255 之间。

下标 K 的值应在 1 到 32 之间。最多可替换的字符数是 32。

字符 NUL 加在最后表示该命令的结束。

ESC :

[名称] 恢复字符集中的字符

[格式] ASCII: ESC :
 十进制: 27 58
 十六进制: 1B 3A

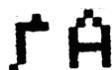
[说明]

该命令恢复字符集中的原字符, 该字符在此之前已被用 ESC % 命令替换为用户定义字符。但是用户定义字符并不从 RAM 中删除, 它还可以用 ESC% 命令调出使用。

[例子]

```
print(0x1b);print(0x57);print(2); //横向纵向放大 2 倍
print(0x1b);print(0x26);print(65); //ESC & 命令
print(0x02);print(0x7C);print(0x40); //字模数据
print(0xC0);print(0x40);print(0x00); //字模数据
print(0x1b);print(0x25);print(65);print(65);print(0); //ESC %
print(65); //打印定义字符
print(0x1b);print(0x3a); //ESC: 命令
print(65); //复字符集中的字符
print(0x0d);
```

[打印结果]



4.4.6 图形打印命令

ESC K

[名称] 打印点阵图形

[格式] ASCII: ESC K n1 n2 ...data...
 十进制: 27 75 n1 n2 ...data...
 十六进制: 1B 4B n1 n2 ...data...

[说明]

该命令打印 $n1 \times 8$ 点阵图形。该图形的宽度为 $n1$ 点，高度为 8 点。每一列的 8 个点可以由一个 8 位的字节来表示，最高位在上。

$n1, n2$ 的数值代表一个 16 位的二进制数， $n1$ 为低 8 位字节， $n2$ 为高 8 位字节，表示 ESC K 命令要打印的图形宽度为 $n2 \times 256 + n1$ 。 $n2$ 应是 0 或 1，点数之和小于等于 384，若超出则自动清除。 $n1$ 应在 1 到该机型的每行最大点数之间。

Data 是该点阵图形从左到右每列的字节内容，字节数应等于 $n1$ 。

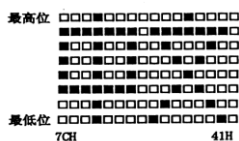
当图形的高度大于 8 点时，可按每 8 点行一个图形单元划分成多个单元。不足 8 点的用空点补齐。然后按顺序每个图形单元分别用 ESC K 命令打印出来，最后组成一个完整的图形。

[注意]

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] 用 ESC K 命令打印两个汉字“中文”这两个汉字的点阵如下图所示。每个字符由 7×8 点阵组成有 7 列。两字符之间有一空格，因此共有 15 列，于是 $n1=15, n2=0$ ，15 个字节的的数据以十六进制表示如下：

7CH, 44H, 44H, FFH, 44H, 44H, 7CH, 00H, 41H, 62H, 54H, C8H, 54H, 62H, 41H



程序如下：

```
unsigned char code ch[15]={0x7c,0x44,0x44,0xff,0x44,0x44,
0x7c,0x00m,0x41,0x62,0x54,0xc8,0x54,0x62,0x41};
print(0x1b);print(0x57);print(2);           //横向纵向放大 2 倍
print(0x1b);print(0x4b);print(15);print(0); //ESC K 命令
for(i=0;i<15;i++)                          //输入 15 个字节数据
    print(tch[i]);
print(0x0d);                                //输入回车
```

[打印结果]

中文

ESC ' ①

[名称] 打印曲线 I

[格式] ASCII: ESC ' m n1 n2...nk CR
 十进制: 27 39 m n1 n2 ...nk 13
 十六进制: 1B 27 m n1 n2 ...nk 0D

[说明]

该命令用于沿走纸方向打印曲线图形。m 的数值是要打印的曲线条数。它应当在 1 到该机型每行最大点数之间。

在一水平点行内,有 m 个曲线点。 n_1, n_2, \dots, n_k 代表这 m 个曲线的位置。 n_k 的数量应等于 m 。每一 n_k 都应小于该微打型号每行最大点数。最后的 CR (回车) 是让打印机打印出这一点行。整个 m 条曲线的图形就是由每一点行用 n_1, n_2, \dots, n_k 这些数据打印出来的。连续使用本命令可打印出任意长度的曲线。

[注意]

本命令只适用于针打及部分热敏机型。

[例子]

要打印下面 5 个方程的曲线图形:

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

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

$$Y3=50$$

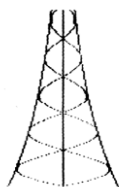
$$Y4=50+40*\text{abs}(-0.1*X)$$

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

你可以用 C 语言编写如下的程序:

```
#include <math.h>
float X;
unsigned int Y,YY;
for(X=0;X<150;X++) //打印 150 点行
{
    Y= 40*exp(-0.01*X);
    YY= Y*sin(X/10);
    print(0x1b);print(39);print(5); //ESC 命令,M=5,0
    print(50+YY);print(50-YY);print(50); print(50+Y);print(50-Y);
    print(13);
}
```

[打印结果]

**ESC** , @

[名称] 打印曲线 II

[格式] ASCII: ESC ' m 低 m 高 n1 低 n1 高.....nk 低 nk 高 CR

十进制: 27 39 m 低 m 高 n1 低 n1 高.....nk 低 nk 高 13

十六进制: 1B 27 m 低 m 高 n1 低 n1 高.....nk 低 nk 高 0D

[说明]

n_1 低 n_1 高代表这 m 个曲线的位置。 n_1 低表示 n_1 点的低位, n_1 高表示 n_1 点的高位。 m 的数值是要打印的曲线条数。它应当在 1 到该机型每行最大点数之间。

在一水平点行内,有 m 个曲线点。 n_1, n_2, \dots, n_k 代表这 m 个曲线的位置。 n_k 的数量应等于 m 。每一 n_k 都应小于该微打型号每行最大点数。最后的 CR (回车) 是让打印机打印出这一点行。整个 m 条曲线的图形就是由每一点行用 n_1, n_2, \dots, n_k 这些数据打印出来的。

连续使用本命令可打印出任意长度的曲线。

[例子]

要打印下面 5 个方程的曲线图形:

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

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

$$Y3=50$$

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

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

你可以用 C 语言编写如下的程序:

```
#include <math.h>
float X;
unsigned int Y,YY;
for(X=0;X<150;X++) //打印 150 点行
{
    Y= 40*exp(-0.01*X);
    YY= Y*sin(X/10);
    print(0x1b);print(39);print(5);print(0); //ESC 命令,M=5,0
    print(50+YY);print(0);print(50-YY);print(0);print(50);print(0);
    print(50+Y);print(0);print(50-Y);print(0);print(13);
}
```

ESC E

[名称] 打印条形码

[格式] ASCII: ESC E nq nc n1 n2 n3...nk NUL
十进制: 27 69 nq nc n1 n2 n3...nk 0
十六进制: 1B 45 nq nc n1 n2 n3...nk 00

[说明]

nq: 条码第 1 条线离打印纸端的距离 (点)。N 的值应在 0 到 64 之间。

nc: 条码线纵向高度 (点)。n 的值应在 1 到 255 之间。

nk: 第 k 个条码线的参数:

高 4 位 (D7~D4) 与前一个条码线之间的空白点数。n 的值应在 1 到 15 之间。

低 4 位 (D3~D0) 该条码线的宽度 (点)。K 为条码线数量, k 的值应在 1 到 15 之间。

[注意]

本命令只适用热敏系列。

4.2.6 初始化命令**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]

该命令后打印机恢复默认值。

4.2.7 数据控制命令**CR**

[名称] 回车



[格式] ASCII: CR
十进制: 13
十六进制: 0D

NUL

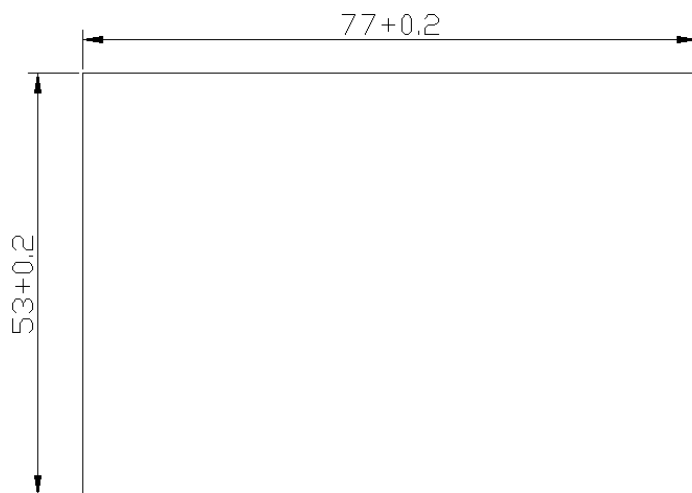
[格式] ASCII: NUL
十进制: 0
十六进制: 00

[说明]

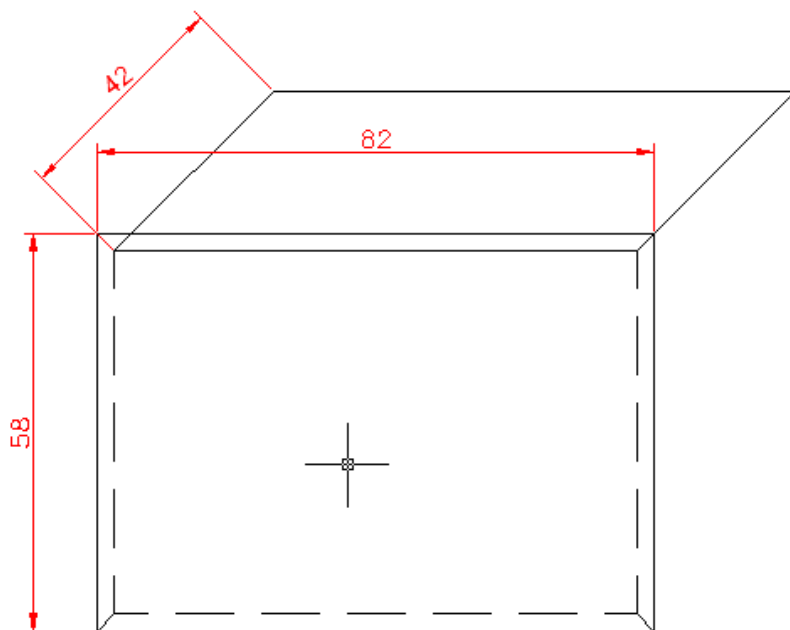
NUL命令和某些命令，如ESC B，ESC D，ESC %，ESC ' 等一起使用，用于表示这些命令的结束。NUL命令单独使用时不起作用。

五、安装尺寸

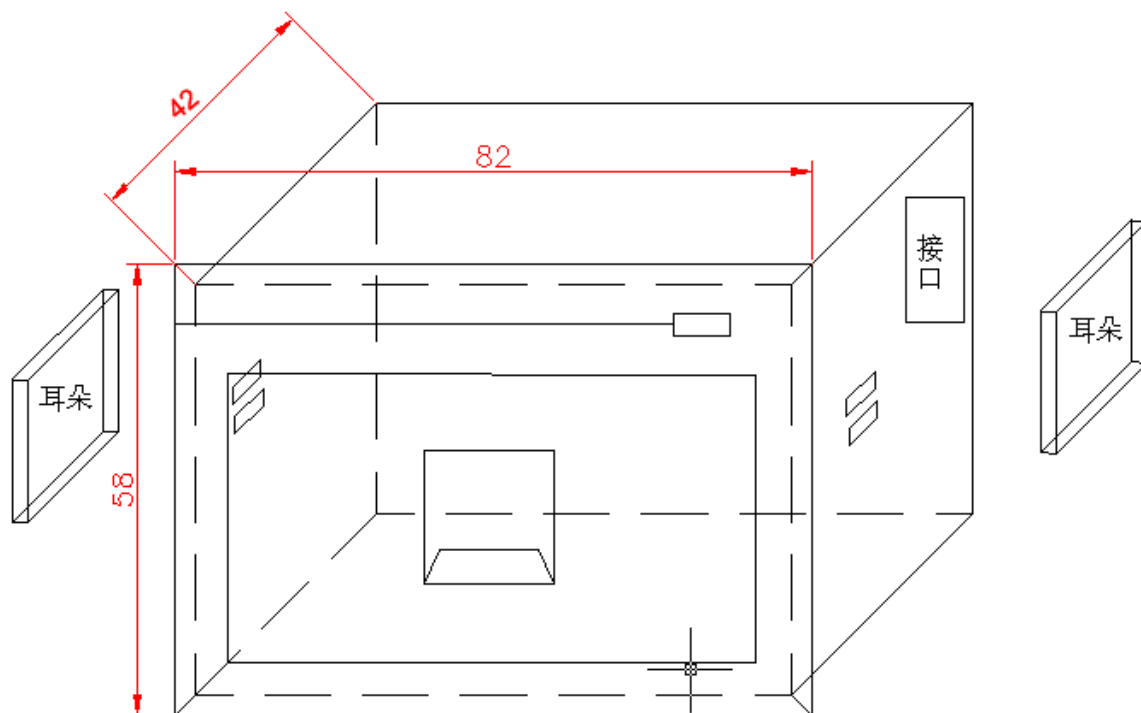
5.1 开口尺寸



5.2 外形尺寸



5.3 外观示图



附录

a ASCII字符集

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|
| 2 | | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / |
| 3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4 | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5 | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |
| 6 | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7 | p | q | r | s | t | u | v | w | x | y | z | { | | } | ~ | Δ |
| 8 | Ç | ü | é | â | ä | à | ã | ç | ê | ë | è | ï | î | ÿ | Ä | Å |
| 9 | É | æ | Æ | ô | ö | ò | û | ù | ÿ | ö | ü | ç | £ | ¥ | ℞ | ƒ |
| A | á | í | ó | ú | ñ | ñ | º | º | ¿ | ¿ | ½ | ¼ | ¼ | ¼ | « | » |
| B | ⋮ | ⋮ | ⋮ | | | | | | | | | | | | | |
| C | L | L | T | | | | | | | | | | | | | |
| D | ll | T | π | | | | | | | | | | | | | |
| E | α | β | Γ | π | Σ | μ | γ | γ | θ | Ω | δ | θ | θ | θ | € | ∩ |
| F | ≡ | ± | ≥ | ≤ | ∫ | ∫ | ≈ | ≈ | ° | . | . | √ | n | z | ■ | ■ |

b 字符集1

| | 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 | 年 | 月 | 日 | ¥ |
| 9 | £ | § | ↓ | → | ^ | ± | ÷ | ∞ | ≈ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| A | α | β | γ | δ | ε | ζ | η | θ | ι | κ | λ | μ | ν | ξ | π | ρ |
| B | τ | φ | ψ | ω | Γ | Δ | Π | Σ | Ψ | Ω | ⊗ | ⊗ | ∩ | Φ | ↑ | ∠ |
| C | □ | □ | □ | □ | — | — | | | / | \ | Γ | L | L | Γ | X | X |
| D | □ | □ | □ | □ | — | — | | | / | \ | Γ | L | L | Γ | — | |
| E | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ | ┘ |
| F | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | + |

c 字符集2

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 百 | 千 | 万 | Ⅱ | ℃ | ° | ′ | ″ | ¼ | ½ | ¾ | ℥ | × | √ | ∞ | ∩ |
| 3 | ∕ | ∥ | ∪ | ∩ | ⊕ | ⊖ | ⊗ | ⊘ | ⊙ | ⊚ | ⊛ | ⊜ | ⊝ | ⊞ | ⊟ | ⊠ |
| 4 | ∠ | ≡ | ≅ | ∴ | ≠ | ≈ | ≪ | ≫ | ≠ | ≠ | ≠ | ≠ | ≠ | ≠ | ≠ | ≠ |
| 5 | ※ | ※ | () | ※ | ※ | ⌒ | ⌒ | ⌒ | ⌒ | ⌒ | ⌒ | ⌒ | ⌒ | ⌒ | ⌒ | ⌒ |
| 6 | 壹 | ア | イ | ウ | エ | オ | カ | キ | ク | ケ | コ | サ | シ | ス | セ | ソ |
| 7 | タ | チ | ツ | テ | ト | ナ | ニ | ヌ | ネ | ノ | ハ | ヒ | フ | ヘ | ホ | マ |
| 8 | ※ | ㄥ | × | モ | ヤ | ユ | ヨ | ラ | リ | ル | レ | ロ | ワ | ヰ | ヱ | ヲ |
| 9 | ン | マ | ウ | エ | オ | ヤ | ユ | ヨ | ツ | ン | ° | ⊕ | ⊖ | ⊗ | ⊘ | ⊙ |
| A | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ |
| B | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ |
| C | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ |
| D | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ |
| E | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ |
| F | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ | ㄨ |