

**ESC/P、ESC/P2、ESC/PK、ESC/PK2**

# 程序设计指南

**EPSON®**

# 目 录

## 前言

■ 本书内容介绍 .....	I
■ 如何使用本手册 .....	II
■ 资料处理的流程 .....	III

## 导 论

■ ESC/P 控制码的意义 .....	0-1
■ 按数字排列指令一览表 .....	0-3
■ 按功能分类指令一览表 .....	0-6
■ ESC/P 各版本指令功能总表 .....	0-10

## 第一章 打字机初始设定

■ 初始化打印机 .....	1-1
----------------	-----

## 第二章 纸张格式

■ 纸张长度设定 .....	2-1
■ 设定页缝空白功能 .....	2-6
■ 边界定位 .....	2-9
■ 行间距 .....	2-11

## 第三章 定位设定

■ 水平定位 .....	3-1
■ 垂直定位 .....	3-3
■ 垂直通道中的定位点设定 .....	3-6

## 第四章 打印中文字符

■ 中文字符码 .....	4-1
■ 汉字的位置安排 .....	4-2
■ 中文字符表的使用 .....	4-2

## 第五章 英文模式下的打印命令

■ 选择英文模式 .....	5-1
■ 选择打印品质 .....	5-2
■ 选择英文字体 .....	5-3
■ 字符间距及比例模式 .....	5-5
■ 设定字符间距 .....	5-7
■ 英文放大、缩小打印 .....	5-9
■ 特殊打印效果 .....	5-12
■ 英文打印模式组合 .....	5-15

## 第六章 中文模式下的打印命令

■ 中文模式的选择 .....	6-1
■ 设定汉字字间距 .....	6-3
■ 汉字放大、缩小打印 .....	6-6
■ 选择汉字字体 .....	6-10
■ 特殊打印效果 .....	6-11
■ 中文表格符号连线 .....	6-15
■ 选择中文 CC-DOS 打印方式 .....	6-16
■ 汉字打印模式组合 .....	6-17

## 第七章 中、英文模式下通用的打印功能

■ 选择字体形态 .....	7-1
----------------	-----

■ 特殊打印效果 .....	7-5
<b>第八章 图象打印</b>	
■ 何谓图象打印 .....	8-1
■ 图象打印模式的选择 .....	8-1
■ 图象资料的解释 .....	8-2
■ 各种图象打印的密度 .....	8-4
■ 图象打印指令序列 .....	8-4
<b>第九章 用户自定义字符</b>	
■ 用户自定义字符功能原理 .....	9-1
■ 英文用户自定义字符的设计及使用 .....	9-1
■ 中文用户自定义字符的设计及使用 .....	9-6
■ 用户自定义字符注意事项 .....	9-10
<b>第十章 使用自动送纸槽</b>	
<b>第十一章 对齐</b>	
<b>第十二章 打印机运作</b>	
■ 资料处理 .....	12-1
■ 打印处理 .....	12-3
■ 颜色选择 .....	12-4
■ 发声处理 .....	12-4
■ 打印位置 .....	12-5
■ 送纸方式 .....	12-11
<b>第十三章 字符表</b>	
■ 选择字符表 .....	13-1
■ 分配字符集 .....	13-3
■ 以字符方式打印数据 .....	13-4
■ 国际字符集 .....	13-4
<b>第十四章 光栅打印命令</b>	
■ 选择图形模式 .....	14-1
■ 微调整控制 .....	14-1
■ 打印光栅图形 .....	14-2
■ 控制点尺寸 .....	14-3
■ 选择光栅图象 (TIFF 压缩模式) .....	14-4
■ 选择光栅图像 (Delta row 压缩模式) .....	14-5
■ 图形数据传送 .....	14-6
■ 指定相对水平位置 .....	14-8
■ 指定相对垂直位置 .....	14-9
■ 指定颜色 .....	14-10
■ 解除 Seed row .....	14-10
■ 解除压缩模式 .....	14-10
■ 水平位置初始化 .....	14-11
■ 选择基准单位字节 .....	14-11
■ 选择基准单位点 .....	14-11
<b>第十五章 特殊功能打印命令</b>	
<b>附录 字符表</b> .....	
A-1	
<b>名词解释</b> .....	
A-3	

## 本书内容介绍

EPSON 爱普生公司积多年经验致力于中文打印机的开发,这次出版本书的目的,是提供给软件程序设计员设计打印机应用程序,或者在编打印机驱动程序时,使用 EPSONESC/P 系列控制码时参考。本书适用于 EPSON 公司生产的、使用 ESC/P → ESC/PK2 系列控制码的所有 24 针打印机、喷墨打印机及激光打印机。

此书包函了 EPSON 所开发出的 ESC/P 系列控制码中,所有应用在 24 针点阵打印机、喷墨及激光打印机上的控制码。文中叙述了打印机的工作原理,并以 BASIC 程序和 C 语言程序做打印实例的示范说明,再配合图表讲解,以期望对打印机所拥有的功能,做最完备的介绍。

全书在内容编排上,分有十三个章节及导论一篇,共计有以下几个重点:

1. [导论] 一篇先对 ESC/P 系列控制码做了全面的介绍。
2. 第四章 [打印中文字符] 中,叙述中文打字的基本原理;第八章 [图象打印模式] 和第九章 [用户自定义字符] 二章中,则讨论打印机绘图原理和用户自定义字符的工作步骤。
3. 第五、六、七三章的内容中,分别介绍了 EPSON 打印机在英文和中文模式下,及两种模式下都具有的共同的打印命令,都以 BASIC 程序和 C 语言程序做了最佳的范例。
4. 在第二、三、十一、十二章的 [纸张格式]、[定位设定]、[对齐] 和 [打印机操作] 的章节中,则详细的介绍了如何规划打印机以格式化打印的报表、文件。
5. 在第十三章 [字符表] 的叙述中,告诉您如何使用 EPSON 打印机中内置的字符表及国际字符集。

具体而言,本书是关于中文打印机的专业书籍,希望此书的出版,能提供电脑软件业者,一个遵循的标准,以共同提高资讯科技,方为所盼。

北京爱普生软件技术发展有限公司

1994 年 8 月

## 如何使用本手册

本书在导论之后附有三个指令一览表,您可由这些表格中迅速的找到您所需要的指令叙述及页次;另外,在本书的附录中,附列有字符表和名词解释,以提供您做需要时参考。

在每一个指令的叙述中,都分成格式和解释两部分。如下所示,便是格式的部分:

---

格式:	ASCII 码:	标准 ASCII 字符序列
	十进制码:	十进制数序列
	十六进制码:	十六进制数序列

---

这三个格式的作用是完全相同的,因此,用户只要选择最适合的一种格式来使用即可。譬如, BASIC 的程序设计员可能适合选用第一种格式和第二种格式;而文字处理程序的用户则适合选用第二种格式;至于组合语言的程序设计员则适合选用第三种格式。此外,在本书中为了区别十六进制数和十进制数,便在十六进制数的后面加上一个 H, 如 01H、1BH 和 17H。

目前,在中国国内使用的 EPSON 24 针点阵打印机,其命令体系主要有 ESC/P、ESC/P2、ESC/PK、ESC/PK2 四种。本书在每一条打印命令的格式中,都清楚的标明了该命令适用于哪些控制命令体系,标 " " 符号者表示该命令体系中有此命令。

至于在各章的编辑上,大致分成以下几项:

1. 各节描述:叙述本节内容重点。
2. 使用范围:介绍该指令适用于 EPSON 打印机的哪些控制命令体系。
3. 指令介绍:对指令的用途加以叙述。
4. 注意事项:对各指令或各段内容所要补充说明的叙述。
5. 程序范例:对各指令的功能,提供 BASIC 和 C 语言程序,加以详述。
6. 打印结果:由范例程序打印出的结果。

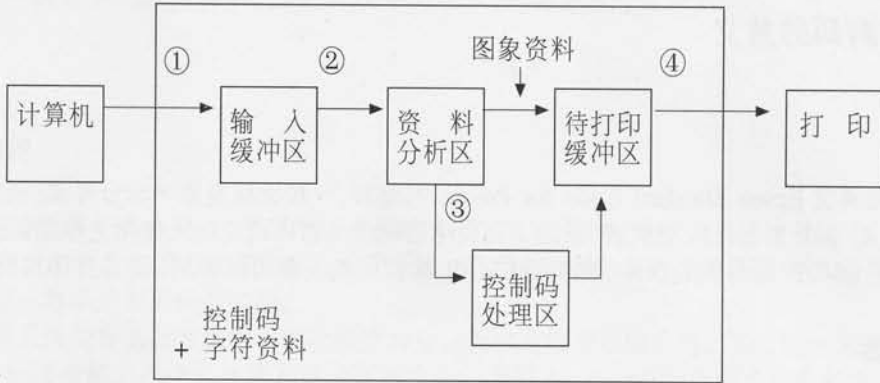
**注:**在每章的示例程序中都有一个头文件 `print.h`,此文件的内容列在了每一章的最后。

本书内容在编辑上,如有不周之处,望请各位同行,不吝指教,以期内容更加完善。

# 资料处理的流程

在说明各种不同的 ESC/P 打印控制码前,必须先了解资料的处理流程;电脑是如何将资料通过打印机的处理,而达到所要求的打印效果,其间的流程详述如下:

## 1. 资料打印处理的流程图



- ① 当电脑传送资料给打印机时,首先存入打印机内的输入缓冲区。
- ② 资料在经过输入缓冲区后送入资料分析区,资料在此被分成图象资料、控制码和字符码,图象资料则送入待打印缓冲区,而控制码和字符码,则送入控制码处理区。
- ③ 控制码处理区将字符码转换成字符形态,而控制码则执行其设定的功能,或将字符转换成设定功能下的字符模式打印出来。
- ④ 经过处理后,图象的字符则存入待打印缓冲区后,再行打印。

## 2. 打印机在遇到下列两种情况时,才会开始打印:

- ① 当待打印缓冲区资料已满时,再多接收一个资料,就会开始打印。
- ② 当送出打印控制码,如 LF、FF、CR、VT 和 ESC J 时,就会立即将缓冲区内资料打印出。

ESC/P 控制码的意义 .....	0-1
按数字顺序指令一览表 .....	0-3
按功能分类指令一览表 .....	0-6
ESC/P 各版本指令功能总表 .....	0-10

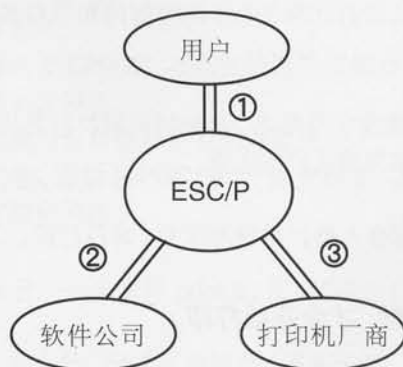
## ESC/P 控制码的意义

### 何谓 ESC/P

ESC/P 是英文 Epson Standard Code for Printer 的缩写,一般读法是取字母分开读;也就是读作 E [i:], S [es], C [si], P [pi:]. ESC/P 是为了实现电脑软件与打印机之间所使用之控制码的标准化而制定的。根据 ESC/P 所作成的软件,在遵循 ESC/P 的打印机上都可以使用,而与打印机的机种无关。

### ESC/P 的目的

ESC/P 是将打印机间的兼容性予以明确化,以便提高打印机驱动程序的开发与维护之效率。而且,在软件上可根据 ESC/P 的版本 (level) 来区分打印机控制码的种类。只要软件版本是高的,都可使用在版本低的打印机上,而不论是什么机种。所以用户都可以很容易地根据自己打印机的 ESC/P 版本,来判断哪些软件可用,哪些软件不能用了。换句话说,为了让用户、软件及打印机三者之间的关系更密切,ESC/P 是一个非常有效的工具。



- ①: 让打印机发挥软件的功能。
- ②: 提高软件开发,维护的效率。
- ③: 有一套可遵循的打印机控制码。

### ESC/P 的特征

ESC/P 含有丰富的控制码群,可让软件将打印特色无限地发挥,而且设计上也能模式化。其中主要特征有下列几个:

· 将单一功能的指令,在各打印机间,用单一控制码来统一对应。即,同样的打印效果,只存在一个控制码。

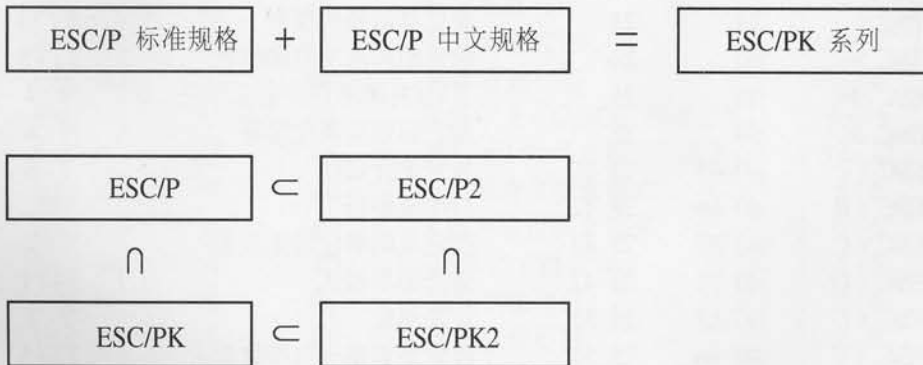
- . 整个结构的发展具有阶梯性。
- . 阶梯上的高版本之功能可完全包含低版本。
- . 从终端打印机（Terminal Printer 亦称桌上打印机）的基本功能开始，进而推广到系统型打印机（System Printer）的高度功能。
- . 软件上可根据 ESC/P 的版本来做打印机驱动程序，不必去考虑复杂多样的厂别与型号。

## ESC/P 的结构

ESC/P 在处理资料的分类上，有标准处理系统（拉丁文字语系）及非英文处理系统（中文、日文、韩文...）两种。后者在中国当然是指中文处理系统，是用来控制中文的各种打印变化，而前者是在处理标准的英文、数字及图形的控制。

各种处理系统均有高版本低版本的阶梯性构造。而标准处理系统有自己独立的一套 ESC/P 版本，称为 ESC/P 标准规格。而中文处理系统也有独立的一套版本（level）。将两者合并起来，就形成了 ESC/P 的中文规格了。

下图表示了，ESC/P 各版本间的互包含的关系：



## ESC/P 的世界性

ESC/P 目前已被欧、美、亚、澳等各洲的资讯先进国，视为一个标准。在这许多国家（如美、英、德、法、日、韩.....）内所开发的软件，均遵循 ESC/P 为打印机驱动程序的标准。相同地，在这些国家中所流行的打印机也都是具兼容性的机种，使得软件与打印机间的沟通能畅行无阻。更值得注意的是，日本 AX 计划（IBM-PC 兼容的日文版）中打印机的规格就是以 ESC/P 为控制码的标准。



# EPSON 按数字顺序指令一览表

指令	十进制数	十六进制数	功能	页数
BEL	7	07	蜂鸣器鸣叫	12-4
BS	8	08	退格	12-2
HT	9	09	执行水平定位	3-1
LF	10	0A	换行	12-11
VT	11	0B	执行垂直定位	3-4
FF	12	0C	换页	12-12
CR	13	0D	回车	12-11
SO	14	0E	设定单行倍宽	5-9
SI	15	0F	设定压缩打印	5-8
DC1	17	11	联机	12-1
DC2	18	12	解除压缩打印	5-9
DC3	19	13	脱机	12-1
DC4	20	14	解除单行倍宽	7-6
CAN	24	18	消除行缓冲区	12-2
DEL	127	7F	删除字符	12-2
ESC SO	14	0E	设定单行倍宽	7-6
ESC SI	15	0F	设定压缩打印	5-8
ESC EM	25	19	设定自动送纸装置	10-1
ESC SP	32	20	设定英文字符间距	5-7
ESC !	33	21	英文打印模式组合	5-14
ESC \$	36	24	设定绝对水平打印位置	12-5
ESC %	37	25	开启/关闭用户自定义字符	9-4
ESC &	38	26	用户自定义英文字符	9-4
ESC (-	40 45	28 2D	设定字符划线	7-8
ESC (B	40 66	28 42	打印条形码	15
ESC (C	40 67	28 43	按定义的单位设定页长	2-2
ESC (G	40 71	28 47	选择图形模式	14-1
ESC (U	40 85	28 55	定义单位	2-1
ESC (V	40 86	28 56	设定绝对垂直打印位置	12-5
ESC (X	40 88	28 58	设定/解除网点打印	7-9
ESC (^	40 94	28 5E	字符方式打印数据	13-4
ESC (c	40 99	28 63	设定页格式	2-3
ESC (e	40 101	28 65	控制点尺寸	14-3
ESC (i	10 105	28 69	微调控制	14-1
ESC (t	40 116	28 74	分配字符表	13-3
ESC (v	40 118	28 76	设定相对垂直打印位置	12-7
ESC *	42	2A	选择图象模式	8-4
ESC +	43	2B	设定 n/360 英寸行间距	2-12
ESC -	45	2D	设定/解除英文下划线	5-12
ESC .	46	2E	打印光栅图形	14-2
ESC .2	46 2	2E 2	选择TIFF压缩模式	14-4
ESC .3	46 3	2E 3	选择Delta row 压缩模式	14-5
ESC /	47	2F	选择垂直定位通道	3-6

指令	十进制数	十六进制数	功能	页数
ESC 0	48	30	设定 1/8 英寸行间距	2-11
ESC 2	50	32	设定 1/6 英寸行间距	2-11
ESC 3	51	33	设定 n/180 英寸行间距	2-12
ESC 4	52	34	设定斜体打印	7-3
ESC 5	53	35	解除斜体打印	7-3
ESC 6	54	36	取消后半部的控制码	13-2
ESC 7	55	37	选择后半部的控制码	13-3
ESC :	58	3A	将 ROM 的字符拷贝到 RAM	9-4
ESC <	60	3C	打印头归位	12-3
ESC ?	63	3F	重新指定图象模式	8-6
ESC @	64	40	初始化打印机	1-3
ESC A	65	41	设定 n/60 英寸行间距	2-12
ESC B	66	42	设定垂直定位点	3-4
ESC C	67	43	以行为单位设定页长	2-3
ESC C 0	67	43	以英寸为单位设定页长	2-2
ESC D	68	44	设定水平定位点	3-1
ESC E	69	45	设定粗体打印	7-5
ESC F	70	46	解除粗体打印	7-5
ESC G	71	47	设定重叠打印	7-5
ESC H	72	48	解除重叠打印	7-6
ESC I	73	49	选择中文 CC-DOS 打印方式	6-16
ESC J	74	4A	执行 n/180 英寸顺方向走纸	12-12
ESC K	75	4B	8 点单密度图象模式	8-6
ESC L	76	4C	8 点双密度图象模式	8-6
ESC M	77	4D	选择 12 CPI 字体	5-5
ESC N	78	4E	设定页缝空白	2-6
ESC O	79	4F	消除页缝空白	2-6
ESC P	80	50	选择 10 CPI 字体	5-5
ESC Q	81	51	设定右边界	2-9
ESC R	82	52	选择国际字符集	13-5
ESC S	83	53	设定英文上/下标打印	5-12
ESC T	84	54	解除英文上/下标打印	5-12
ESC U	85	55	设定单/双向打印	12-3
ESC W	87	57	设定/解除倍宽打印	7-6
ESC X	88	58	选择英文字符的宽度和高度	5-11
ESC Y	89	59	8 点高速双密度图象模式	8-5
ESC Z	90	5A	8 点四密度图象模式	8-5
ESC \	92	5C	设定相对水平打印位置	12-6

指令	十进制数	十六进制数	功能	页数
ESC a	97	61	设定对齐状态	11 - 1
ESC b	98	62	设定通道中的垂直定位点	3 - 6
ESC c	99	63	设定英文水平移动距离	12 - 10
ESC g	103	67	选择 15 CPI 字体	5 - 5
ESC j	106	6A	执行 n/180 英寸逆向走纸	12 - 12
ESC k	107	6B	选择英文字体	5 - 3
ESC l	108	6C	设定左边界	2 - 9
ESC p	112	70	设定/解除比例模式	5 - 5
ESC q	113	71	选择字体形态	7 - 1
ESC r	114	72	选择打印颜色	12 - 4
ESC s	115	73	设定/解除半速打印	12 - 3
ESC t	116	74	选择字符表	13 - 3
ESC w	119	77	设定/解除倍高打印	5 - 9
ESC x	120	78	选择打印品质	5 - 2
FS SO	14	0E	设定单行倍宽	6 - 6
FS SI	15	0F	设定半角汉字	6 - 8
FS DC2	18	12	解除半角汉字	6 - 8
FS DC4	20	14	解除单行倍宽	6 - 6
FS !	33	21	汉字打印模式组合	6 - 17
FS &	38	26	设定汉字模式	6 - 1
FS -	45	2D	设定/解除汉字下划线	6 - 13
FS .	46	2E	解除汉字模式	6 - 2
FS 2	50	32	用户自定义中文字符	9 - 7
FS D	68	44	纵向半角两字符并列打印	6 - 12
FS J	74	4A	设定纵向打印	6 - 11
FS K	75	4B	设定横向打印	6 - 12
FS S	83	53	设定全角汉字字间距	6 - 3
FS T	84	54	设定半角字符字间距	6 - 3
FS U	85	55	设定半角字符间距调整	6 - 3
FS V	86	56	取消半角字符间距调整	6 - 4
FS W	87	57	设定/解除四倍角打印	6 - 6
FS Y	89	59	选择汉字点阵和汉字高度	6 - 10
FS b	98	62	设定汉字基准线	6 - 15
FS c	99	63	设定汉字水平移动距离	12 - 10
FS k	89	59	选择汉字字体	6 - 11
FS r	114	72	设定汉字上/下标打印	6 - 8
FS v	118	76	设定/解除表格符连线	6 - 15
FS x	120	78	设定/解除高速打印	12 - 4

# EPSON ESC/P 按功能分类指令一览表

类别		指令功能		页号
打	打印区域	ESC (U	定义单位	2-1
		ESC (C	按定义的单位设定页长	2-2
		ESC (c	设定页格式	2-3
		ESC C 0	以英寸为单位设定页长	2-2
		ESC C	以行为单位设定页长	2-3
		ESC N	设定页缝空白	2-6
		ESC O	消除页缝空白	2-6
		ESC l	设定左边界	2-9
ESC Q	设定右边界	2-9		
印	行间距	ESC 0	设定 1/8 英寸行间距	2-11
		ESC 2	设定 1/6 英寸行间距	2-11
		ESC 3	设定 n/180 英寸行间距	2-12
		ESC A	设定 n/60 英寸行间距	2-12
		ESC +	设定 n/360 英寸行间距	2-12
格	定位设定	ESC D	设定水平定位点	3-1
		ESC B	设定垂直定位点	3-4
		ESC b	设定通道中的垂直定位点	3-6
		ESC /	选择垂直定位的通道	3-6
式	打印位置	HT	执行水平定位	3-1
		VT	执行垂直定位	3-4
		ESC (V	设定绝对垂直打印位置	12-5
		ESC (v	设定相对垂直打印位置	12-7
		ESC \$	设定绝对水平打印位置	12-5
		ESC \	设定相对水平打印位置	12-6
		ESC c	设定英文水平移动距离	12-10
		FS c	设定汉字水平移动距离	12-10
	对齐	ESC a	设定对齐状态	11-1
英文模式	字符设定	ESC (t	分配字符表	13-3
		ESC t	选择字符表	13-3
		ESC 6	取消后半部的控制码	13-2
		ESC 7	选择后半部的控制码	13-3

类别		指令功能		页号	
英	字符设定	ESC R	选择国际字符集	13-5	
		ESC (^	字符方式打印数据	13-4	
	字符定义	ESC &	用户自定义英文字符	9-4	
		ESC :	将 ROM 的字符拷贝到 RAM	9-4	
ESC %		开启/关闭用户自定义字符	9-4		
字符间距	ESC SP	设定英文字符间距	5-7		
文	打印形态	ESC x	选择打印品质	5-2	
		ESC k	选择英文字体	5-3	
		ESC P	选择 10CPI 字体	5-5	
		ESC M	选择 12CPI 字体	5-5	
		ESC g	选择 15CPI 字体	5-5	
		ESC p	设定/解除比例模式	5-5	
		ESC X	选择英文字符的宽度和高度	5-11	
		SI	设定压缩打印	5-8	
		ESC SI	设定压缩打印	5-8	
		DC2	解除压缩打印	5-9	
		SO	设定单行倍宽	5-9	
		ESC w	设定/解除倍高打印	5-9	
		ESC S	设定英文上/下标打印	5-12	
		ESC T	解除英文上/下标打印	5-12	
		ESC -	设定/解除英文下划线	5-12	
		ESC !	英文打印模式组合	5-14	
模	式	通	ESC q	选择字体形态	7-1
			ESC 4	设定斜体打印	7-3
			ESC 5	解除斜体打印	7-3
			ESC E	设定粗体打印	7-5
			ESC F	解除粗体打印	7-5
			ESC G	设定重叠打印	7-5
		用	ESC H	解除重叠打印	7-6
			ESC W	设定/解除倍宽打印	7-6
			ESC SO	设定单行倍宽	7-6
			DC4	解除单行倍宽	7-6
			ESC (-	设定字符划线	7-8
			ESC (X	设定/接除网点打印	7-9
中文模式					

类别		指令功能		页号
英文模式		ESC I	选择中文 CC-DOS 打印方式	6-16
		FS &	设定汉字模式	6-1
		FS .	解除汉字模式	6-2
		FS SO	设定单行倍宽	6-6
		FS DC4	解除单行倍宽	6-6
		FS W	设定/解除四倍角打印	6-6
		FS SI	设定半角汉字	6-8
		FS DC2	解除半角汉字	6-8
		FS b	设定汉字基准线	6-15
		FS r	设定汉字上/下标打印	6-8
		FS J	设定纵向打印	6-11
		FS K	设定横向打印	6-12
		FS D	纵向半角两字符并列打印	6-12
		FS -	设定/解除汉字下划线	6-13
		FS v	设定/解除表格符连线	6-15
		FS k	选择汉字字体	6-11
		FS Y	选择汉字点阵和汉字高度	6-10
	FS !	汉字打印模式组合	6-17	
	字符定义	FS 2	用户自定义中文字符	9-7
	字符间距	FS S	设定全角汉字字间距	6-3
		FS T	设定半角字符字间距	6-3
		FS U	设定半角字符间距调整	6-3
		FS V	取消半角字符间距调整	6-4
	图象打印	ESC *	选择图象模式	8-4
		ESC Y	8点高速双密度图象模式	8-5
		ESC Z	8点四密度图象模式	8-5
		ESC K	8点单密度图象模式	8-6
		ESC L	8点双密度图象模式	8-6
		ESC ?	重新指定图象模式	8-6
		ESC (G	选择图形模式	14-1
		ESC (i	微调控制	14-1
		ESC .	打印光栅图形	14-2
		ESC (e	控制点尺寸	14-3
		ESC .2	选择光栅图象(TIFF压缩模式)	14-4
	ESC .3	选择光栅图象(Delta row模式)	14-5	
	特殊功能打印	ESC (B	打印条形码	15

类别		指令功能		页号
打 印 机 运 作	打印机复位	ESC @	初始化打印机	1 - 3
	送纸方式	LF	换行	12 - 11
		FF	换页	12 - 12
		CR	回车	12 - 11
		ESC J	执行 n/180 英寸顺方向走纸	12 - 12
		ESC j	执行 n/180 英寸逆方向走纸	12 - 12
	打印处理	ESC <	打印头归位	12 - 3
		ESC U	设定单/双向打印	12 - 3
		ESC s	设定/解除半速打印	12 - 3
		FS x	设定/解除高速打印	12 - 4
资料处理	DC1	联机	12 - 1	
	DC3	脱机	12 - 1	
	CAN	消除行缓冲区	12 - 2	
	DEL	删除字符	12 - 2	
	BS	退格	12 - 2	
CSF处理	ESC EM	设定自动送纸装置	10 - 1	
颜色处理	ESC r	选择打印颜色	12 - 4	
发声处理	BEL	蜂鸣器鸣叫	12 - 4	

ESC/P 各版本指令功能总表

打印功能分类		ESC/P	ESC/P2	ESC/PK	ESC/PK2
打印格式	打印区域	ESC C, ESC C 0 ESC N, ESC O ESC Q, ESC 1	ESC (U ESC (C ESC (c	<input type="checkbox"/> ESC (U <input type="checkbox"/> ESC (C <input type="checkbox"/> ESC (c	ESC (U ESC (C ESC (c
	行间距	ESC 0, ESC 2 ESC A, ESC 3 ESC +			
	定位设定	ESC D, ESC B ESC /, ESC b	<input type="checkbox"/> ESC / <input type="checkbox"/> ESC b	ESC / ESC b	<input type="checkbox"/> ESC / <input type="checkbox"/> ESC b
	打印位置	HT, VT, ESC \$ ESC \	ESC (V ESC (v ESC c	<input type="checkbox"/> ESC (V <input type="checkbox"/> ESC (v <input type="checkbox"/> ESC c	ESC (V, ESC c ESC (v, FS c
	字符对齐	ESC a	<input type="checkbox"/> ESC a	ESC a	<input type="checkbox"/> ESC a
英文模式	字符设定	ESC 6, ESC 7 ESC t, ESC R	ESC (t ESC (^	<input type="checkbox"/> ESC (t <input type="checkbox"/> ESC (^	ESC (t ESC (^
	字符定义	ESC %, ESC & ESC :			
	打印效果	ESC w, ESC - SI, DC2, SO ESC !, ESC SI ESC P, ESC M ESC g, ESC p ESC S, ESC T ESC k, ESC x ESC SP	ESC X	<input type="checkbox"/> ESC X	ESC X
通用	打印效果	ESC E, ESC F ESC G, ESC H ESC 4, ESC 5 ESC q, ESC w ESC (-, DC 4 ESC SO		<input type="checkbox"/> ESC (-	ESC (- ESC (X
中文模式	字符定义			FS 2	
	打印效果			FS SO, FS DC4 FS SI, FS DC2 FS J, FS K FS D, FS - FS v, FS r FS &, FS .. FS !, FS W FS S, FS T FS U, FS V	FS k, FS Y FS b, ESC I



打印功能分类		ESC/P	ESC/P2	ESC/PK	ESC/PK2
图象处理	图象打印	ESC K, ESC L ESC Y, ESC Z	ESC ( G ESC .		ESC ( G ESC .
其它功能	送纸方式	ESC *, ESC ? CR, LF, FF ESC J, ESC j			
	打印机复位	ESC @			
	打印处理	ESC <, ESC U ESC s,	□ESC s	ESC s, FS x	□ESC s
	资料处理	DC1, DC3, CAN DEL, BS			
	CSF 处理	ESC EM			
	颜色选择	ESC r			
	发声处理	BEL			
适用机型	标准机型	LQ-1000 LQ-1500	LQ-100 LQ-2500 Stylus-300 Stylus-800 Stylus-1000		
	中文机型			LQ-150K LQ-1600K DLQ-2000K	LQ-1800K LQ-1900K MJ-800K

注: ① 上表中位于后面的命令集包含前面的命令集, 后面的命令集所列命令为在前面命令集的基础上增加的命令。

② 在上表中, 前边标有“□”符号的命令, 表示该命令集中没有此命令。

### ■ 初始化打印机

打印机的初始设定值是当电源被打开、初始化或重新启动时,所生效的设定值。

一般说来,为了避免进行中的打印模式影响以后的打印形态,而造成混乱,所以在每一个文件结束之后,立即用 ESC @ 指令把打印机重置回预设的状态,是很必要的工作。而打印机除了关机以后再开机,可以返回初始设定值外,还有以下二种方法:

1. ESC/P 的 ESC @ 打印指令

2. INIT 的信号

下表一表示打印机的初始定值:

打印机设定	开机时	INIT信号	ESC @ 指令
是否回到纸张顶端?	是	是	是
输入缓冲区是否清除?	是	是	是
待打印缓冲区是否清除?	是	是	是
打字头是否归位?	是	是	是
图形密度是否如下: ESC K: 60点/英寸 ESC L: 120点/英寸 ESC Y: 120点/英寸 ESC Z: 240点/英寸	是 是 是 是	是 是 是 是	是 是 是 是
用户自定义字符是否清除?	是	是	是
其它设定: DIP切换开关和控制面板是否存在?	是	是	是

上表一所描述的"其它设定",是自行调整打印机出厂设定值之后的设定值,通常打印机在出厂时的初始设定值,如下表二所列:

表二：EPSON 打印机出厂设定值

打印结果	设定值
字符间距	每一英寸 10 个字符
特殊打印效果	取消
右边界	取消
换行间距	1/6 英寸
水平定位	每 8 格一定位点
垂直定位	无
VFU 通道 (请见第三章)	通道 0
对齐位置	左边对齐
打印方向	双向

除了上表二打印机的出厂设定值的项目外，每一种都可能不同预设值，其项目如下：

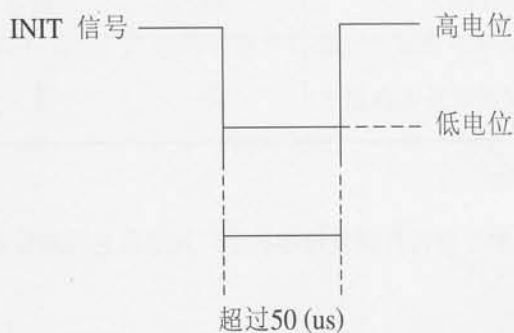
- . 选择的打印字体
- . 国际字符集
- . 页缝空白功能的设定
- . SLCT IN 信号设定
- . AUTO FEED XT 信号设定
- . 送纸槽

## 初始化打印机

初始化打印机的方法有两种：

### 1. INIT 信号

当 INIT 在并行接口是低电位，且超过 50 微秒 (us) 后，便会重新启动打印机，回到初始设定值，而且会清除缓冲区的资料。



## 2. ESC @

当软件输入 "ESC @"初始化打印机的指令时,会清除先前对打印机的控制指令,返回初始设定。

ESC @	初始化打印机	ESC/P	ESC/P2	ESC/PK	ESC/PK2
-------	--------	-------	--------	--------	---------

格式:	ASCII 码:	ESC	@
	十进制码:	27	64
	十六进制码:	1B	40

### 注释:

使用此指令时,必须注意的是,由于 ASCII 码 64 被用来表示各个不同国家的各个不同字符,因此,出现在用户键盘上的字符也许会异于格式中的 @ 字符。

### 注意事项

- 凡在下达 ESC@或 INIT 信号后,会清除先前对打印机改变的设定,使打印机返回初始设定;之后,若要重新设定打印机,则须重下控制码,也可由 DIP 切换开关和控制面板来改变 ESC @或 INIT 信号之后的打印机设定。
- 因为 ESC @会把待打印缓冲区中的资料清除,除非确定缓冲区中的资料不再需要,否则不要轻易的使用 ESC@初始化打印机指令。
- 而在 ESC@指令之前,要打印出缓冲区中的资料时,必须在 ESC @指令前,先下达回车 (CR), 换行 (LF) 指令,即可打出缓冲区内的资料。

### 程序范例

```
#include "stdio.h"
#define ESC 27
main ()
{
    fprintf (stdprn,"%c%c",ESC,'@');
}
```

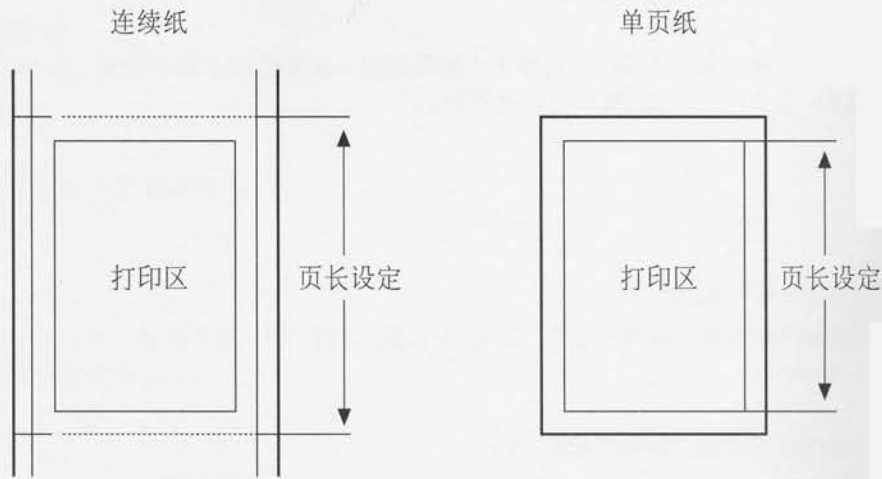
## 第二章

### 纸张格式

■ 纸张长度设定 .....	2-1
定义单位 .....	2-1
按定义的单位设定页长 .....	2-2
以英寸为单位设定页长 .....	2-2
以行为单位设定页长 .....	2-3
设定页格式 .....	2-3
■ 设定页缝空白功能 .....	2-6
■ 边界定位 .....	2-8
■ 行间距 .....	2-11

### 纸张长度设定

通常设定纸张的长度是为了要定义纸张可打印的区域;而用户可依据行数或英寸数来设定页面的长度。而打印区域的范围又随纸张规格不同而有差异(如单页纸或连续纸)。



### 定义单位

该指令用于定义移动打印位置的单位。

ESC ( U

定义单位

ESC/P2

ESC/PK2

格式:    ASCII 码:    ESC    (    U    n1    n2    m

十进制码：	27	40	85	n1	n2	m
十六进制码：	1B	28	55	n1	n2	m

注释：

设定单位为 m/3600 英寸。打印机根据此命令所设定的单位，应用如下命令移动打印位置，设定页长及页顶和页底的空白量：

ESC ( V, ESC ( v, ESC , ESC \$ ,  
ESC ( C, ESC ( c。

参数范围如下：

n1 = 1, n2 = 0。  
m 的有效值为：10, 20, 30, 40, 50, 60。

### 按定义的单位设定页长

按定义的单位设定页长的指令如下：

ESC ( C	按定义的单位设定页长						ESC/P2	ESC/PK2
格式：ASCII 码：	ESC	(	C	n1	n2	m1	m2	
十进制码：	27	40	67	n1	n2	m1	m2	
十六进制码：	1B	28	43	n1	n2	m1	m2	

注释：

根据以下公式，按已定义的单位设定页长。

页长 = (m1 + m2 \* 256) \* (单位)  
其中，(单位) 是用 ESC ( U 命令定义的。

参数范围如下：

n1 = 2, n2 = 0 ;

此外, 0 < 页长 < 22 英寸

### 以英寸为单位设定页长

以英寸为单位设定页长的指令如下：

ESC C NUL	以英寸为单位设定页长				ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式：ASCII 码：	ESC	C	NUL	n				

十进制码：	27	67	0	n
十六进制码：	1B	43	00	n

注释：

设定页面的长度为 n 英寸；至于 n 的值则介于 1 至 22 之间。

## 以行为单位设定页长

当要以行为单位设定页长时，要先设定行间距（见第 2-15 页），再乘上所设定的行数，得到的乘积，即为页面长度。

ESC C	以行为单位设定页长	ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式：	ASCII 码：	ESC	C		n
	十进制码：	27	67		n
	十六进制码：	1B	43		n

注释：

可根据目前的行间距，而把页面长度设定为 n 行；至于 n 的值则必须介于 1 至 127 之间。但页长设定不得超过 22 英寸。

## 设定页格式

设定页格式的指令如下：

ESC ( c	设定页格式	ESC/P2	ESC/PK2
格式：	ASCII 码：	ESC ( C n1 n2 t1 t2 b1 b2	
	十进制码：	27 40 67 n1 n2 t1 t2 b1 b2	
	十六进制码：	1B 28 43 n1 n2 t1 t2 b1 b2	

注释：

根据以下公式，按已定义的单位设定页顶和页底的空白量。

页顶空白量 = (t1 + t2 \* 256) \* (单位)

页底空白量 = (b1 + b2 \* 256) \* (单位)

其中，单位是用 ESC (U 命令定义的。

参数范围如下：

n1 = 4, n2 = 0 ;

并且，顶部空白量 < 底部空白量，max (顶部空白量) = 22 英寸

## 页长设定注意事项

- . 当下达页面长度设定值时, 则页面的顶端位置是指目前这一行。
- . 此指令设定后, 则先前的设定页缝空白 (ESC N) 指令无效。
- . 一旦页长设定后, 即使行间距改变, 而页长仍然不变。
- . 当纸张长度以毫米 (mm) 为计算单位, 而行数相对于行间距的值, 其计算公式如下:

$$\frac{X}{180} \times N = \frac{L}{25.4}$$

L = 页长以毫米计 (mm)

N = 页长设定的行数

X = 行间距设定值; 此值即为 n/180 英寸行间距的 n 值, 亦即 ESC 3 n 中的 n 值 (见第 2 - 20 页)。

## 程序范例一

下面是一个设定 2 英寸页面长度的程序范例。

```
#include "print2.h"

main ()
{

    unsigned char para [] [10] = {
        {1}, {1},
        {0,2}, {2},
        {0,11}, {2},
    };
    unsigned char *data [] = {
        "---Page top line---\n",
        "---Next page top line---\n"
    };

    PrintOut (LQPrintData [0], LQPrintData [1] [0]);
    PrintOut (para [0], para [1] [0]);
    PrintOut ( LQPrintData [2], LQPrintData [3] [0]);
    PrintOut (para [2], para [3] [0]);          /* 页长设定 2 英寸 */
    PrintOut (data [0], 0);
```



```

PrintOut (LQPrintData [18],LQPrintData [19] [0]); /* 换页      */
PrintOut (data [1],0);
PrintOut (LQPrintData [2],LQPrintData [3] [0]);
PrintOut (para [4],para [5] [0]);          /* 返回初始设定 */
}

```

打印结果

--- Page top line ---

--- Next page top line ---

## 程序范例二

这是一个设定页长后,计算出行数 and 行间距相对值的程序范例。若页长设定 38.1 毫米 (mm), 则计算 X 和 N 值如下:

$$\frac{X}{180} \times N = \frac{38.1}{25.4} \rightarrow X \times N = 270$$

X = 3 (表示 3/180 英寸行间距)

N = 90 (行数)

## C 语言

```

#include "print2.h"

main ()
{
    unsigned char para [] [10] = {
        {1}, {1},
        {3}, {1},
        {90}, {1},
        {0,11}, {2},
    };
    unsigned char *data [] = {
        "---Page top line---\n",
        "---Next page top line---\n"
    };
}

```

```

PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
PrintOut (para [0] ,para [1] [0]);
PrintOut (LQPrintData [4] ,LQPrintData [5] [0]);
PrintOut (para [2] ,para [3] [0]);          /* 行间距 3/180 英寸 */
PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
PrintOut (para [4] ,para [5] [0]);        /* 页长 90 行    */
PrintOut (data [0] ,0);
PrintOut (LQPrintData [18] ,LQPrintData [19] [0]); /* 换页      */
PrintOut (data [1] ,0);
PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
PrintOut (para [6] ,para [7] [0]);
}

```

打印结果

--- Page top line ---

--- Next page top line ---

## 设定页缝空白功能

设定页缝空白功能是在每一页的最后一行到次页第一行之间，加入几个空白行，以跳过指定的页缝空白行数。此功能仅在使用连续纸时才有效。

页缝空白的距离是由当前的行间距以及由 ESC N 指令所指定的行数（即 n 值）来决定，指令格式如下：

ESC N	设定页缝空白			
	ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式：	ASCII 码：	ESC	N	n
	十进制码：	27	78	n
	十六进制码：	1B	4E	n

注释：

把每一页最后一行和次页第一行之间的页缝空白行数设定为 n 行。例如：行间距为 1/6 英寸，页面长度为 66 行的话，输入 ESC N 6 指令，则打印机在打印 60 行之后，会跳过 6 行后再打印。n 的值介于 1 和 127 之间，但必须小于页面可打印行数，否则无效。

ESC O	消除页缝空白			
	ESC/P	ESC/P2	ESC/PK	ESC/PK2

格式：	ASCII 码：	ESC	O
	十进制码：	27	79
	十六进制码：	1B	4F

注释：

消除由 ESC N 指令设定的页缝空白。此指令可解除 DIP 切换开关或由 [选择模式] 所设定的 1 英寸页缝空白功能。

注意事项

- . 当在页缝空白功能执行之后重新设定页面长度时，则设定页缝空白功能被取消。
- . 页缝空白功能在 ESC C, ESC C 0 和 ESC O 之后时无效。
- . 当行间距设定改变，并不会改变已设定的页缝空白功能。

## 程序范例

下面是一个页长设定为 6 行，而设定页缝空白 2 行的程序。

C 语言

```
#include "print2.h"
main ()
{
    int i ;
    unsigned char para [] [10] = {
        {6}, {1},
        {2}, {1},
        {66}, {1},
        };
    unsigned char *data [] = {
        "--- Page top line ---\n",
        "2 行\n",
        "3 行\n",
        "4 行\n",
        "5 行\n",
        "6 行\n"
        };
    PrintOut (LQPrintData [6],LQPrintData [7] [0]);
    PrintOut (LQPrintData [8],LQPrintData [9] [0]); /*1/6 英寸行间距*/
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
```

```
PrintOut (para [0],para [1] [0]);          /* 页长 6 行    */
PrintOut (LQPrintData [10],LQPrintData [11] [0]);
PrintOut (para [2],para [3] [0]);          /* 页缝空白 2 行 */
for (i=0 ;i<6 ;i++)
PrintOut (data [i] ,0);
PrintOut (LQPrintData [12],LQPrintData [13] [0]); /* 取消页缝空白 */
PrintOut (LQPrintData [2],LQPrintData [3] [0]);
PrintOut (para [4],para [5] [0]);          /* 页长 66 行    */
}
```

### 打印结果

--- Page top line ---

2 行

3 行

4 行

5 行

6 行

## 边界定位

边界定位是设定打印区的左右边界。通常打印机的左右边界的初始设定值是打印机最大可打印列数。在 80 列的机种上,最大打印宽度是 8 英寸;136 列的机种最大宽度 13.6 英寸。

以一个 80 列打印机的左右边界设定为例:如左边界设定在第 10 列,右边界设定在第 70 列。则打印区域为从 11 列至 70 列;1 至 10 为左边界区域,71 至 80 为右边界区域。

\* 若没有设定左边界时,则左边界从第一列开始。

\* 若没有设定右边界时,则右边界即为打印机可打印的最大列数。

左右边界设定的控制指令如下:

ESC I	设定左边界	ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式:	ASCII 码 :	ESC	l	n	
	十进制码 :	27	108	n	
	十六进制码 :	1B	6C	n	

注释:

用户可根据目前的字符宽度,而把左边界设定为 n 列;其中, n 的值必须介于 0 至 255 之间。

ESC Q	设定右边界	ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式:	ASCII 码 :	ESC	Q	n	
	十进制码 :	27	81	n	
	十六进制码 :	1B	51	n	

注释:

用户可根据目前的字符宽度,而把右边界设定为 n 列;其中, n 的值必须介于 0 至 255 之间。

注意事项

- . 设定左边界时,在 80 列的机型上超过 4.5 英寸;或在 136 列的机型上超过 8 英寸时,则此设定值无效。
- . 在使用上,当打印出的字符到达右边界时,打印机会自动产生回车和换行。
- . 当使用 ESC SP 设定字符间的空格,要设定边界时,请注意包括加入的空格数。

. 该指令所设定的绝对边界在英文模式下是由目前字符宽度所决定的。

. 左右边界设定在中文模式下：

① 在 EPSON ESC/PK 或 ESC/PK2 机型上, 皆以目前的字符宽度 (中文模式下为一个全角中文字符, 西文模式下为一个 ASCII 字符) 为一个设定的基本单位。

② 在 EPSON ESC/P 或 ESC/P2 机型上, 皆以目前的字符宽度 (一个 ASCII 字符) 为一个设定的基本单位。

## 程序范例

下面是一个在 136 列打印机上, 左边界设定在第 10 列, 右边界设定在第 26 列的打印范例。

```
#include "print2.h"

main ()
{
    int i ;
    unsigned char para [][5] = {
        {10}, {1},
        {18}, {1},
        {0}, {1},
        {136}, {1},
    };
    unsigned char *data [][5] = {
        "1234567890123456789012345678901234567890\n",
        "中文之美中文之美\n"
    };

    PrintOut (LQPrintData [6], LQPrintData [7] [0]);
    PrintOut (data [0], 0);
    PrintOut (LQPrintData [14], LQPrintData [15] [0]);
    PrintOut (para [0], para [1] [0]);          /* 左边界在第 10 列 */
    PrintOut (LQPrintData [16], LQPrintData [17] [0]);
    PrintOut (para [2], para [3] [0]);        /* 右边界在第 18 列 */
    for (i=1 ; i<=5 ; i++)
        PrintOut (data [i], 0);
    PrintOut (LQPrintData [14], LQPrintData [15] [0]);
    PrintOut (para [4], para [5] [0]);       /* 返回初始设定 */
    PrintOut (LQPrintData [16], LQPrintData [17] [0]);
    PrintOut (para [6], para [7] [0]);
}

```

## 打印结果

1234567890123456789012345678901234567890

中文之美中文之美  
中文之美中文之美  
中文之美中文之美  
中文之美中文之美  
中文之美中文之美

## 行间距

所谓行间距是指当打印机接受换行 (LF) 指令后, 移动纸张的距离。页长设定、垂直定位点及页缝空白的距离, 均以此行间距为基本计算单位。

HHHHHHHHHHHHHHHH

行间距

HHHHHHHHHHHHHHHH

行间距

HHHHHHHHHHHHHHHH

一般打印机在开机时的行间距, 初始设定值是 1/6 英寸, 但用户可由软件改变为 1/8 英寸、n/60 英寸、n/180 英寸或 n/360 英寸的行间距。

ESC 0 设定 1/8 英寸行间距

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式: ASCII 码: ESC 0  
十进制码: 27 48  
十六进制码: 1B 30

注释:

将以后换行指令的换行距离设定为 1/8 英寸; 其中, 0 是阿拉伯数字 0, 而不是 ASCII 码 0。

ESC 2 设定 1/6 英寸行间距

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式: ASCII 码: ESC 2  
十进制码: 27 50  
十六进制码: 1B 32

注释:

将以后换行指令的换行距离设定为 1/6 英寸; 其中, 2 是阿拉伯数字 2, 而不是 ASCII 码 2。而这一个换行距离也是开机后的初始设定值。

ESC 3

设定 n/180 英寸行间距

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	3	n
	十进制码:	27	51	n
	十六进制码:	1B	33	n

注释:

将以后换行指令的换行距离设定为 n/180 英寸;其中,3 是阿拉伯数字 3,而不是 ASCII 码 3,而 n 的值必须介于 0 至 255 之间。

ESC A

设定 n/60 英寸行间距

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	A	n
	十进制码:	27	65	n
	十六进制码:	1B	41	n

注释:

将以后换行指令的换行距离设定为 n/60 英寸;其中,n 的值必须介于 0 至 85 之间。

ESC +

设定 n/360 英寸行间距

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	+	n
	十进制码:	27	43	n
	十六进制码:	1B	2B	n

注释:

将以后换行指令的换行距离设定为 n/360 英寸;而 n 的值则必须介于 0 至 255 之间。

## 程序范例

```
#include "print2.h"
main ()
{
    unsigned char para [] [1] = {3};
    PrintOut (LQPrintData [8],LQPrintData [9] [0]); /* 设定 1/6 英寸间距 */
    PrintOut (LQPrintData [4],LQPrintData [5] [0]); /* 设定 3/180 英寸距 */
    PrintOut (para,1);
}
```



```

/*      print2.h      */

#include "stdio.h"

/*****
/* 变量定义      */
*****/
#define ESC      0x1B /*模式设定  */
#define FF      0x0C /*换页    */

void PrintOut ( ) ;
/*****
/* 数据结构体设定      */
*****/
static unsigned char LQPrintData [] [10] = {
    {ESC,'x'}, {2}, /*选择打印品质  */
    {ESC,'C'}, {2}, /*以行为单位设定页长 */
    {ESC,'3'}, {2}, /*设定 n/180 英寸行间距 */
    {ESC,'@'}, {2}, /*打印机初始化  */
    {ESC,'2'}, {2}, /*设定 1/6 行间距  */
    {ESC,'N'}, {2}, /*设定页缝空白  */
    {ESC,'O'}, {2}, /*消除页缝空白  */
    {ESC,'I'}, {2}, /*设定左边界  */
    {ESC,'Q'}, {2}, /*设定右边界  */
    {FF}, {1} /*换页    */
};
/*****
void PrintOut (unsigned char* lpcbData,int Length) /*打印输出 */
{
    if (!Length) fprintf (stdprn,"%s",lpcbData) ;
    else do {
        fprintf (stdprn,"%c",*lpcbData++) ;
        Length-- ;
    } while (Length) ;
}

```

## 定位设定

- 水平定位 ..... 3-1
- 垂直定位 ..... 3-3
- 垂直通道中的定位点设定 ..... 3-6

### 水平定位

打印机开机时的水平定位点的缺省设定值是每隔 8 个字符一个定位点, 用户也可利用 ESC D 指令来改变水平定点的位置, 当打印机接收到执行水平定位动作 (HT) 的指令时, 便将打印头推进至下一水平定位点的位置。

ESC D	设定水平定位点		ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式:	ASCII 码:	ESC D	n1	n2 ... 0		
	十进制码:	27 68	n1	n2 ... 0		
	十六进制码:	1B 44	n1	n2 ... 00		

注释:

此指令以递增的顺序输入, 如格式中的 n1 和 n2 表示定位点的列数 (column number), 其值介于 1 至 255 之间, 可任意设定至 32 个水平定位点, 0 则表示设定的结束, 而 ESC D 0 也可以清除整个水平定位点的设定值。

定位点位置 = [设定的字符宽度] × [列数]  
(column number)

HT	执行水平定位		ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式:	ASCII 码:	HT				
	十进制码:	9				
	十六进制码:	09				

注释:

把打印的位置推进到下一个水平定位点的位置。

### 水平设定的注意事项

1. 在英文模式状态下:

- . 若要设定水平定位点时, 则不同的比例模式、设定粗体、压缩等特殊效果时, 定位点会依字符宽度的不同而有差异。
- . 左边界设定是水平定位的第 0 点; 一旦改变了左边界后, 所有水平定位点都会跟着依序移动。
- . 一旦定位点已设定, 则定位点位置将不会受字符间距改变的影响。
- . 使用 ESC D 水平定位点的设定时, 必须以 n1, n2, ... 递增的顺序输入, 一旦输入值小于前一数值, 则在该点结束设定, 把后面的设定值作为无关的数据处理。
- . 当使用 ESC SP 设定字符间空格时, 则设定定位点时要包括加入的空格。
- . 若在打印区域之外下达执行水平定位 HT 指令时无效。
- . 在比例模式下, 无法设定水平定位点; 即使必须设定, 则采用的字符宽度为 10 CPI。
- . 当重新设定定位点位置时, 先前设定的位置全部消除, 重设值即生效。

## 2. 中文模式状态下 :

- . 在 EPSON ESC/P-K 系列机型上 (包括 P-K2), 要设定中文的水平定位点时, 皆以目前字符的宽度 (一个全角中文字符) 为一个基本的计算单位。
- . 在中文倍宽、压缩模式下, 仍以其当前字符宽度为一个水平定位的基本单位。
- . 其它设定时的方法, 与英文模式状态下相同。

## 程序范例

下面是左边界设定在第 4 列, 而水平定位点分别在左边界后第 2, 11, 20, 25 列的例子。

```
#include "print3.h"

main ()
{
    int i ;
    unsigned char para [] [10] = {
        {4}, {1},
        {2,5,10,12,0}, {5},
        {0}, {1}
    };
    unsigned char *data [] = {
        "1234567890123456789012345678901234567890\n",
```

```

"水",
"平",
"定",
"位\n"
};

```

```

PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
PrintOut (LQPrintData [4] ,LQPrintData [5] [0]);
PrintOut (data [0] ,0);
PrintOut (LQPrintData [6] ,LQPrintData [7] [0]);
PrintOut (para [0] ,para [1] [0]);          /* 左边界设定在第4列 */
PrintOut (LQPrintData [8] ,LQPrintData [9] [0]);
PrintOut (para [2] ,para [3] [0]);        /* 水平定位的位置 */
for (i=1 ;i<=4 ;i++)
{
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (data [i] ,0);
}
PrintOut (LQPrintData [6] ,LQPrintData [7] [0]);
PrintOut (para [4] ,para [5] [0]);        /* 取消左边界设定 */

for (i=1 ;i<=4 ;i++)
{
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (data [i] ,0);
}
PrintOut (LQPrintData [8] ,LQPrintData [9] [0]);
PrintOut (para [4] ,para [5] [0]);
}

```

打印结果

1234567890123456789012345678901234567890

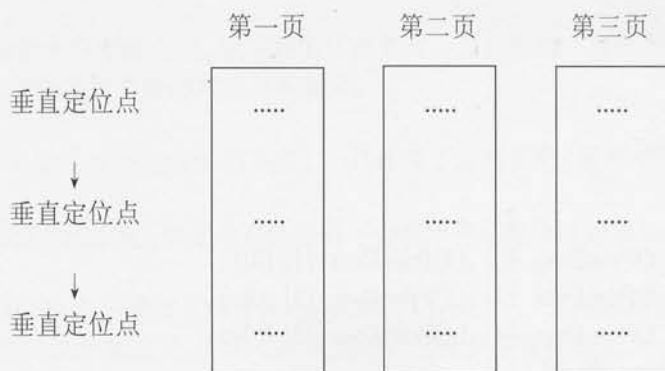
```

      水 平 定 位
水 平 定 位

```

## 垂直定位

一般在打印图表时，除了需要设定水平定位点外，还可能需设定垂直定位点，此时可先用 ESC B 来设定垂直定位点，然后再用 VT 指令可将打印头移到下一个垂直定位点，如下图：



即垂直定位有二个步骤：首先要以 ESC B 指令设定垂直位置，再以 VT 指令执行垂直定位运行。

ESC B	设定垂直定位点		ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式：	ASCII 码：	ESC B	n1	n2 .....	nk	0
	十进制码：	27 66	n1	n2 .....	nk	0
	十六进制码：	1B 42	n1	n2 .....	nk	00

注释：

十进数  $0 \leq nk \leq 255, 1 \leq k \leq 16$

$n1, n2, \dots$  的值表示垂直定位点的行数，可设定最多 16 个垂直定位点。而 ESC B 0 这一个指令则可清除垂直定位点的设定值。在这一个指令下，所有的设定值都储存在通道 0 中。

VT	执行垂直定位		ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式：	ASCII 码：	VT				
	十进制码：	11				
	十六进制码：	0B				

注释：

根据 ESC/ 指令所设定的垂直定位通道，把打印头移到下一个定位点。如果用户并未使用 ESC/ 指令来选择通道的话，那么，使用 ESC B 指令（即通道 0）的垂直定位设定值。当然，如果通道中并未设定垂直定位点的话，纸张便向前推进一行。

## 程序范例

下面是一个设定 15 行的页长，而三个垂直定位分别第 4 行、第 8 行和第 12 行。该程序下达了四次 VT 指令。其中的垂直定位点与英文模式下皆相同。

```

#include "print3.h"
main ()
{
    int i ;
    unsigned char para [] [10] = {
        {15}, {1},
        {4,8,12,0}, {4},
        {66}, {1},
        {0}, {1}
    };
    unsigned char *data [] = {
        "--- Page top line ---\n",
        "中文之美\n",
        "---Next page top line---\n"
    };

    PrintOut (LQPrintData [12] ,LQPrintData [13] [0]);
    PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
    PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
    PrintOut (para [0] ,para [1] [0]);
    PrintOut (LQPrintData [16] ,LQPrintData [17] [0]);
    PrintOut (para [2] ,para [3] [0]);          /* 垂直定位在 4,8,12 行 */
    PrintOut (data [0] ,0);
    PrintOut (LQPrintData [4] ,LQPrintData [5] [0]);
    for (i=1 ;i<4 ;i++)
    {PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
    PrintOut (data [1] ,0);
    }
    PrintOut (LQPrintData [18] ,LQPrintData [19] [0]); /* 执行垂直位 */
    PrintOut (LQPrintData [12] ,LQPrintData [13] [0]);
    PrintOut (data [2] ,0);
    PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
    PrintOut (para [4] ,para [5] [0]);          /* 返回初始设定 */
    PrintOut (LQPrintData [16] ,LQPrintData [17] [0]);
    PrintOut (para [6] ,para [7] [0]);
}

```

打印结果

--- Page top line ---

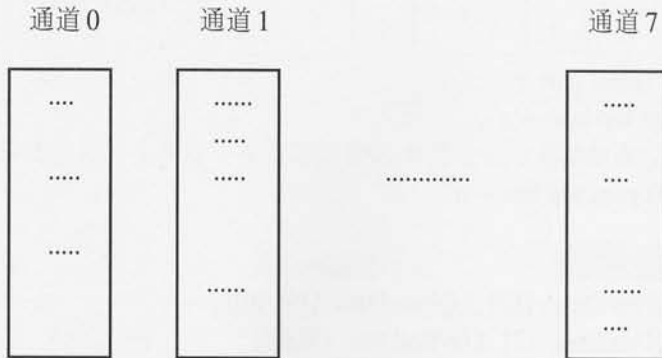
中文之美

中文之美

中文之美

### 垂直通道中的定位点 (VFU) 设定

在打印机中有 8 个通道可以储存 8 组以内的垂直定位点, 可用 ESC / 来选择垂直定位点的通道。



选择 VFU (Vertical Formatting Unit) 通道和设定定位点的指令格式如下:

ESC b	设定通道中的垂直定位点						ESC/P	ESC/PK
格式:	ASCII 码:	ESC	b	c	n1	n2 ... 0		
	十进制码:	27	98	c	n1	n2 ... 0		
	十六进制码:	1B	42	c	n1	n2 ... 00		

注释:

$$0 \leq c \leq 7, 0 \leq nk \leq 255, 1 \leq k \leq 16$$

这一个指令的功能和 ESC B 相同, 只不过这一个指令又多了一个参数 c, 可用以选择一个通道来储存垂直定位点, 而其值则必须介于 0 至 7 之间。因此, 总共可以设定 8 组垂直定位点。其中, 选择通道 0 时的垂直定位点即 ESC B 指令的垂直定位点, 二者可互相改变设定的垂直定位点。选择通道的指令则为 ESC /; 而 ESC b c 0 则可以清除通道 c 的垂直定位点。

ESC /	选择垂直定位通道			ESC/P	ESC/PK
格式:	ASCII 码:	ESC	/	c	
	十进制码:	27	47	c	
	十六进制码:	1B	2F	c	

注释:

这一个指令可用以选择垂直定位点的通道, 而 c 的值则必须介于 0 至 7 之间。如此, 其后的 VT 指令便将使用此一指令所选定的通道中的垂直定位点。

## 垂直定位注意事项

- . 当垂直定位点超过页长时, 超过页长的定位点被记忆着, 直到页长设定加长后才生效。
- . 若垂直定位点设定在页缝空白的范围内, 则垂直定位点直到取消页缝空白功能后才有效。
- . 当下达 VT 指令, 而下一个定位点超过页长时, 则打印机执行换页动作, 将打印头移至下页的顶端。
- . 使用 ESC B 或 ESC b 设定垂直定位点时, n1, n2 ... 必须以递增的顺序输入。一旦输入值小于前一数值, 则结束此指令, 并将后续的数值当作一般资料处理。
- . 当垂直定位点位置重新设定时, 则先前的定位点位置被消除, 而新设定的位置则生效。
- . 在中文与英文模式下, 功能没有差别。

## 程序范例

下面是二个通道的垂直定位例子。二者页长设定皆为 10 行; 一个设定在通道 2 中的第 3、7 行, 一个设定在通道 4 中的第 2、5 行。

```
#include "print3.h"

main ()
{
    unsigned char para [] [10] = {
        {1}, {1},
        {10}, {1},
        {2,3,7,0}, {4},
        {4,2,5,0}, {4},
        {2}, {1},
        {4}, {1},
        {0}, {1},
        {66}, {1},
        {2,0}, {2},
        {4,0}, {2}
    };

    unsigned char *data [] = {
        "---Page top line channel 2---\n",
        "CH2 中文之美\n",
        "---Page top line channel 4---\n",
        "CH4 中文之美\n",
        "---Next page top line---\n"
    }
}
```



```

PrintOut (LQPrintData [20] ,LQPrintData [21] [0]);
PrintOut (para [0] ,para [1] [0]);
PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
PrintOut (para [2] ,para [3] [0]);          /* 页长 10 行 */
PrintOut (LQPrintData [22] ,LQPrintData [23] [0]);
PrintOut (para [4] ,para [5] [0]);          /* 设定通道 2 的 3,7 行 */
PrintOut (LQPrintData [22] ,LQPrintData [23] [0]);
PrintOut (para [6] ,para [7] [0]);          /* 设定通道 4 的 2,5 行 */
PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
PrintOut (para [8] ,para [9] [0]);          /* 选择通道 2 */
PrintOut (data [0] ,0);
PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
PrintOut (data [1] ,0);                    /* 中文之美 */
PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
PrintOut (data [1] ,0);
PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
PrintOut (para [10] ,para [11] [0]);        /* 选择通道 4 */
PrintOut (data [2] ,0);
PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
PrintOut (data [3] ,0);
PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
PrintOut (data [3] ,0);
PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
PrintOut (data [4] ,0);
PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
PrintOut (para [12] ,para [13] [0]);        /* 返回通道 0 */
PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
PrintOut (para [14] ,para [15] [0]);
PrintOut (LQPrintData [22] ,LQPrintData [23] [0]);
PrintOut (para [16] ,para [17] [0]);        /* 清除垂直通道 2 */
PrintOut (LQPrintData [22] ,LQPrintData [23] [0]);
PrintOut (para [18] ,para [19] [0]);        /* 清除垂直通道 4 */
}

```

打印结果

---- Page top line Channel 2 ----

CH2 中文之美

CH2 中文之美

---- Page top line Channel 4 ----

CH4 中文之美

CH4 中文之美

---- Next page top line ----

```
/*      print3.h      */

#include "stdio.h"
/*****
/* 变量定义          */
/*****
#define ESC      0x1B /* 模式设定   */
#define FS      0x1C /* 汉字模式设定 */
#define HT      0x09 /* 执行水平跳格 */
#define VT      0x0B /* 执行垂直跳格 */
void PrintOut ( );
/*****
/* 数据结构体设定          */
/*****
static unsigned char LQPrintData [] [10] = {
{ESC,'@'}, {2}, /* 初始化打印机   */
{ESC,'2'}, {2}, /* 设定 1/6 英寸行间距 */
{FS,'&'}, {2}, /* 设定汉字模式   */
{ESC,'I'}, {2}, /* 设定左边界     */
{ESC,'D'}, {2}, /* 设定水平定位点 */
{HT}, {1}, /* 执行水平定位   */
{FS,'.'}, {2}, /* 设定英文模式   */
{ESC,'C'}, {2}, /* 以行为单位设定页长 */
{ESC,'B'}, {2}, /* 设定垂直定位点 */
{VT}, {1}, /* 执行垂直定位   */
```

```
{ESC,'x'}, {2}, /* 选择打印品质 */
{ESC,'b'}, {2}, /* 设定通道中的垂直定位点 */
{ESC,'/'}, {2} /* 选择垂直定位通道 */
};
```

```
/***/
void PrintOut (unsigned char* lpcbData,int Length) /* 打印输出 */
{
    if (!Length) fprintf (stdprn,"%s",lpcbData);
    else do {
        fprintf (stdprn,"%c",*lpcbData++);
        Length--;
    } while (Length);
}
```

### 打印中文字符

■ 中文字符码 .....	4-1
编码单位 .....	4-1
字符的种类及设定 .....	4-1
■ 汉字的位置安排 .....	4-2
国标码汉字的位置安排 .....	4-2
■ 中文字符表的使用 .....	4-2

EPSON 中文打印机在其内置的 ROM 存储器中, 拥有 7144 个以上的中文字符, 以便快速、正确的打印出中文。本章将描述打印机在中文模式下, 是如何处理中文字符的。

### 中文字符码

#### 编码单位

由于在中文系统下, 有许多中文字符要识别, 而 ASCII 码表中的 256 个码已不够使用; 因此, 中文字符码的编码单位为 2 个字节。在本书中, 中文字符码是以十六进制数字来表示的, 如下所示:

D8C7H, BAC3H

其中, 前两个数字表示第一个字节, 而后两个数字则表示第二个字节。H 表示十六进制数。至于在传送上, 当然是先送出第一个字节, 紧接着再送出第二个字节。

#### 字码的种类及设定

在 EPSON 的中文打印机中都已内置有国家标准汉字编码的字库。其编范围如表一所示 (表一):

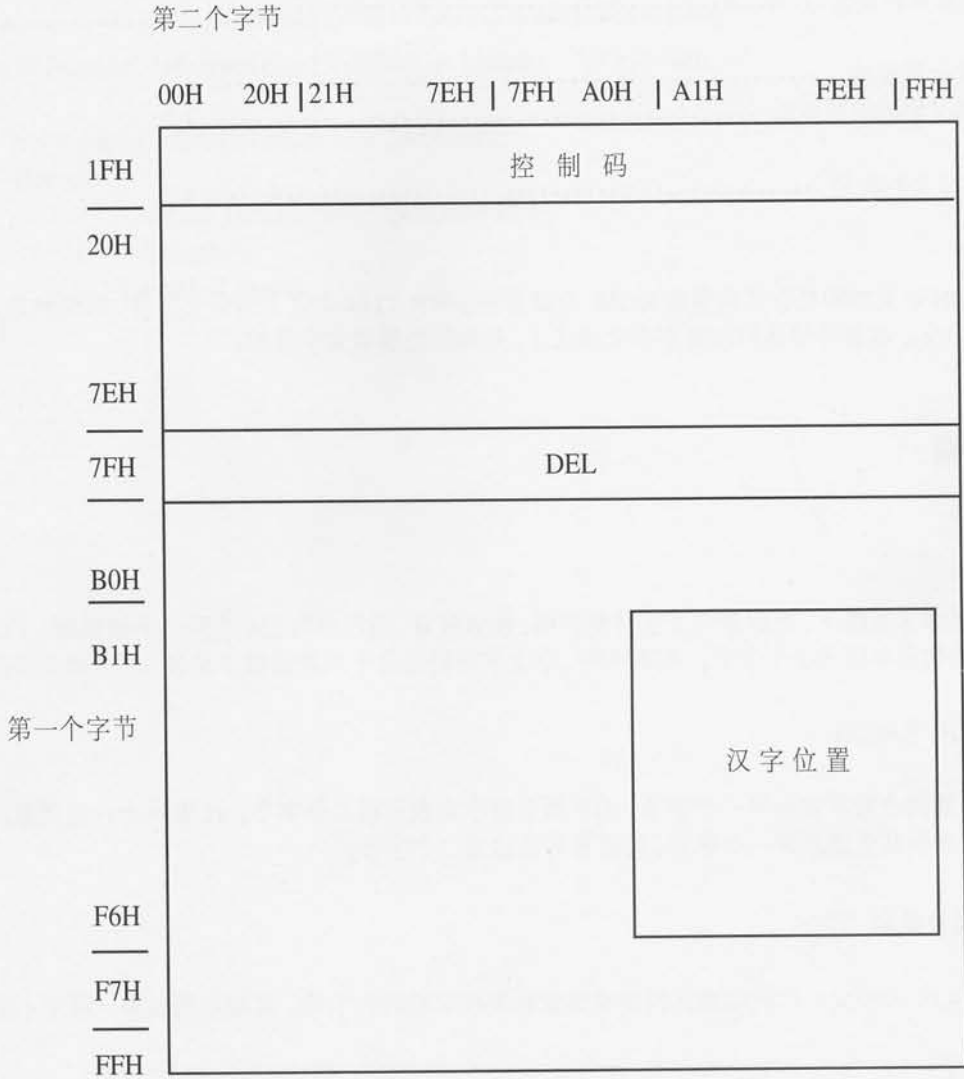
字码种类	编 码 范 围		字 码 数
	第一个字节	第二个字节	
国家标准码	B0H ~ F7H	A1H ~ FEH	7144

(表一)

## 汉字的位置安排

### 国标码汉字的位置安排

EPSON 中文打印机中,其内置的汉字库的位置如下所示:



### 中文字符表的使用

在 EPSON 打印机手册中都附有中文字符参考表,从表中可以找出您所需要的汉字的编码。在表中,行编号表示汉字的第一个字节;列编码表示汉字的第二个字节。

例如,要打印出字符"利";怎样利用此参考表?

先查出"利"的第一个字节为 C0H, 然后, 再查出"利"的第二个字节为 FBH; 因此, 将双字节 C0FBH 送往打印机即打印出汉字"利"。

写成 C 程序是 :

```
fprintf (stdprn,"%c%c",0xc0,0xfb);
```

注意事项

- . 中文字符表中空格部份意味着没有适合那个码的字符。如您送出那个码, 将无字符打印出。
- . 并非所有机种都内置有汉字库, 有关内置汉字库的种类, 请查阅打印机操作手册。

## 英文模式下的打印命令

■ 选择英文模式 .....	5-1
■ 选择打印品质 .....	5-2
■ 选择英文字体 .....	5-3
■ 字符间距及比例模式 .....	5-5
■ 设定字符间距 .....	5-7
■ 英文放大、缩小打印 .....	5-9
英文压缩打印 .....	5-9
英文倍宽打印 .....	5-9
英文倍高打印 .....	5-10
选择英文字符的宽度和高度 .....	5-11
■ 特殊打印效果 .....	5-12
英文上标下标打印 .....	5-12
英文下划线模式 .....	5-13
■ 英文打印模式组合 .....	5-15

打印机的打印模式按所处理资料的种类,可分为英文模式和中文模式。EPSON 中文打印机(内置汉字库)的缺省状态即为中文模式;在中文模式下,本章节对英文字符的打印控制命令无效,可通过软件将中文模式变为英文模式。而英文打印机(无内置汉字库)则只有英文模式。这两种模式都各有一套独立完整的打印指令来控制各种打印形式。而所谓英文模式,就是当您在处理纯英文的资料时,所必须要设定的打印机状态。

所以在您做资料处理时,要先考虑打印机应设定在何种模式下,这样打印机才能正确地打印所处理的资料。

### 选择英文模式

在 EPSON 中文打印机上,要打印英文字符时,必须要确定打印机是在英文模式下,否则在中文模式下,将打印出 13.3 CPI 的英文字符。此时用户除了可调整 DIP 切换开关,或者在 [选择模式] 中设定英文模式 (ASCII MODE) 外,也可以在中文模式下直接用 FS . (请参考 6-2 页) 指令来设定打印机成英文模式。

## 选择打印品质

在 ESC x 指令下,可选择文书体 (Letter Quality, LQ) 或草体 (Draft) 两种打印品质。

LQ : 在较低的速度下,用较多的点数打印出较密实的字体。

Draft : 在较高的速度下,用较少的点数打印出较稀疏的字体。

ESC x	选择打印品质	ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式: ASCII 码:	ESC	x	n		
十进制码:	27	120	n		
十六进制码:	1B	78	n		

注释:

n = 0 时,选择草体;

n = 1 时,选择文书体。

注意事项

. 草体无法在比例模式下打印出来。

## 程序范例

```
#include "print5.h"
main ()
{

    unsigned char para [] [10] = {
        {0}, {1},
        {1}, {1},
        };
    unsigned char *data [] = {
        "Draft Quality\n",
        "Letter Quality\n"
    };
    PrintOut (LQPrintData [0],LQPrintData [1] [0]); /* 选择英文模式 */
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (para [0],para [1] [0]); /* 选择草体 */
    PrintOut (data [0],0);
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (para [2],para [3] [0]); /* 选择文书体 */
    PrintOut (data [1],0);
}
```



## 打印结果

Draft 0 space  
Draft 5 space  
Letter 0 space  
Letter 5 space

## 选择英文字体

在选择文书体打印时,用 ESC k 最多可选择七种文书字体。

ESC k	选择英文字体	ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式: ASCII 码:	ESC	k	n		
十进制码:	27	107	n		
十六进制码:	1B	6B	n		

注释:

此指令只在文书体状态下才生效,在草体状态下无效。

下列为 n 值及所选的字型:

- 0 = Roman
- 1 = Sans Serif
- 2 = Courier
- 3 = Prestige
- 4 = Script
- 5 = OCR-B
- 6 = OCR-A

注意事项

- . 当打印机没有内置多种字体时,一般都是 ROMAN 为开机设定。
- . 有些中文打印机没有全部内置 7 种字体,如 LQ - 1800K 只内置前两种字体,对于这些打印机,选择没有内置的字体时,本命令无效。
- . 当变量值超出规定的范围时,本命令无效。
- . ESC k 指令只在文书体 (Letter Quality) 下才有效,在草体 (Draft) 下无效。

## 程序范例

```
#include "print5.h"
```

```

main ()
{
    unsigned char para [] [10] = {
        {0}, {1},
        {1}, {1},
        };
    unsigned char *data [] = {
        "ROMAN",
        "--Seeing is believing\n",
        "SANS SERIF"
        };

    PrintOut (LQPrintData [4],LQPrintData [5] [0]); /* 初始化打印机 */
    PrintOut (LQPrintData [0],LQPrintData [1] [0]); /* 选择英文模式 */
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (para [2],para [3] [0]); /* 选择文书体 */
    PrintOut (LQPrintData [6],LQPrintData [7] [0]);
    PrintOut (para [0],para [1] [0]); /* 选择字体打印 */
    PrintOut (data [0],0);
    PrintOut (data [1],0);
    PrintOut (LQPrintData [6],LQPrintData [7] [0]);
    PrintOut (para [2],para [3] [0]);
    PrintOut (data [2],0);
    PrintOut (data [1],0);
}

```

打印结果

```

ROMAN--Seeing is believing
SANS SERIF--Seeing is believing

```

## 字符间距及比例模式

一般字符间距有三种基本的选择，即 10 CPI、12 CPI、和 15 CPI。而所谓 CPI (Character Per Inch) 即表示每英寸可打印的字符数。

打印机在出厂时的初始设定，都是 10 CPI。若您重新改变了字符间距之后，可取消先前的字符间距。

比例模式的打印形式是将各个字符容许的空格和字符的大小成比例。例如，较窄的字符（如 i）所拥有的空间便小于较宽的字符（如 W）所占的空间，而使整段文字更为美观。

ESC P                    选择 10 CPI 字体

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式： ASCII 码：        ESC        P  
          十进制码：        27        80  
          十六进制码：      1B        50

注释：

选择 10 CPI 的字符间距。

ESC M                    选择 12 CPI 字体

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式： ASCII 码：        ESC        M  
          十进制码：        27        77  
          十六进制码：      1B        4D

注释：

选择 12 CPI 的字符间距。

ESC g                    选择 15 CPI 字体

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式： ASCII 码：        ESC        g  
          十进制码：        27        103  
          十六进制码：      1B        67

注释：

选择 15 CPI 的字符间距。15 CPI 的字符不能应用在压缩模式上。

ESC p                    设定 / 解除比例模式

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式： ASCII 码：        ESC        p        n  
          十进制码：        27        112      n  
          十六进制码：      1B        70        n

注释：

n = 1 时，设定比例模式

n = 0 时，解除比例模式

比例模式的字符大小和 10 CPI 的字符大小相等。

注意事项

比例字符间距设定的优先权高于其他字符间距的设定，意即要设定 10 CPI, 12 CPI 或 15 CPI 的字符间距时，需先关闭比例模式。

比例模式只在文书体模式下有效,草体无效。

## 程序范例

```
#include "print5.h"

main ()
{
    unsigned char para [][5] = {
        {1}, {1},
        {0}, {1},
        };
    unsigned char *data [] = {
        "15 character/inch\n",
        "12 character/inch\n",
        "10 charatcer/inch\n",
        "Proportional spacing\n"
    };

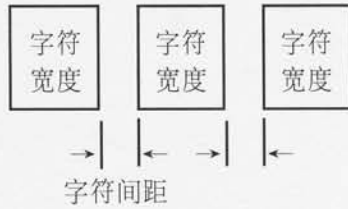
    PrintOut (LQPrintData [0],LQPrintData [1] [0]); /* 设定英文模式 */
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (para [0],para [1] [0]); /* 选择文书模式 */
    PrintOut (LQPrintData [8],LQPrintData [9] [0]); /* 选择 15 CPI 字体 */
    PrintOut (data [0],0);
    PrintOut (LQPrintData [10],LQPrintData [11] [0]); /* 选择 12 CPI 字体 */
    PrintOut (data [1],0);
    PrintOut (LQPrintData [12],LQPrintData [13] [0]); /* 选择 10 CPI 字体 */
    PrintOut (data [2],0);
    PrintOut (LQPrintData [32],LQPrintData [33] [0]);
    PrintOut (para [0],para [1] [0]); /* 设定比例模式 */
    PrintOut (data [3],0);
    PrintOut (LQPrintData [32],LQPrintData [33] [0]);
    PrintOut (para [2],para [3] [0]); /* 解除比例模式 */
}
```

## 打印结果

```
15 character/inch
12 character/inch
10 character/inch
Proportional spacing
```

## 设定字符间距

字符间距,就是在已有的字符间空格外,再加入指定的空格于每一个字符的右边,如下图:



ESC SP

设定英文字符间距

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	SP	n
	十进制码:	27	32	n
	十六进制码:	1B	20	n

注释:

$0 \leq n \leq 127$ ,  $n$  是字符间距的点数,  $0$  是初始设定值。在草体下点间距单位为  $1/120$  英寸; 而文书字体和比例模式下则为  $1/180$  英寸。

## 程序范例

```
#include "print5.h"
main ()
{
    unsigned char para [] [5] = {
        {0}, {1},
        {5}, {1},
        {1}, {1},
        };
    unsigned char *data [] = {
        "Draft 0 space\n",
        "Draft 5 space\n",
        "Letter 0 space\n",
        "Letter 5 space\n"
        };

    PrintOut (LQPrintData [0], LQPrintData [1] [0]); /* 设定英文模式 */
    PrintOut (LQPrintData [2], LQPrintData [3] [0]);
    PrintOut (para [0], para [1] [0]); /* 选择草体 */
    PrintOut (data [0], 0);
    PrintOut (LQPrintData [34], LQPrintData [35] [0]);
}
```

```

PrintOut (para [2] ,para [3] [0]);          /* 设定字符间距 5/120 英寸 */
PrintOut (data [1] ,0);
PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
PrintOut (para [0] ,para [1] [0]);          /* 返回初始设定      */
PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
PrintOut (para [4] ,para [5] [0]);          /* 选择文书体      */
PrintOut (data [2] ,0);
PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
PrintOut (para [2] ,para [3] [0]);          /* 设定字符间距 5/180 英寸 */
PrintOut (data [3] ,0);
PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
PrintOut (para [0] ,para [1] [0]);          /* 返回初始设定      */
}

```

## 打印结果

```

Draft 0 space
Draft 5 space
Letter 0 space
Letter 5 space

```

## 英文放大、缩小打印

在 EPSON 打印机的文字处理效果上,英文压缩模式可将 10 CPI 和 12 CPI 字符宽度压缩为 60% 来打印。在倍高或倍宽模式中,则可以将任一字符放大为 2 倍的高度或宽度,也可以一起使用放大成四倍的英文字符。

## 英文压缩打印

SI	设定压缩打印	ESC/P	ESC/P2	ESC/PK	ESC/PK2
----	--------	-------	--------	--------	---------

格式:	ASCII 码:	SI
	十进制码:	15
	十六进制码:	0F

注释:  
压缩打印不能应用在 15 CPI 的字符上。

ESC SI	设定压缩打印	ESC/P	ESC/P2	ESC/PK	ESC/PK2
--------	--------	-------	--------	--------	---------

格式:	ASCII 码:	ESC	SI
	十进制码:	27	15
	十六进制码:	1B	0F



## 注意事项

. 当设英文倍高模式时, 行间距不变。

## 程序范例

```
#include "print5.h"
main ()
{
    unsigned char para [] [5] = {
        {0}, {1},
        {1}, {1},
        };
    unsigned char *data [] = {
        "Double-Height Font\n",
        "Double-Width Font\n",
        "Condensed Font\n"
    };
    PrintOut (LQPrintData [0],LQPrintData [1] [0]); /* 设定英文模式 */
    PrintOut (LQPrintData [14],LQPrintData [15] [0]); /* 1/6 英寸行间距 */
    PrintOut (LQPrintData [12],LQPrintData [13] [0]);
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (para [2],para [3] [0]);
    PrintOut (LQPrintData [16],LQPrintData [17] [0]);
    PrintOut (para [2],para [3] [0]); /* 英文倍高 */
    PrintOut (data [0],0);
    PrintOut (LQPrintData [16],LQPrintData [17] [0]);
    PrintOut (para [0],para [1] [0]);
    PrintOut (LQPrintData [18],LQPrintData [19] [0]); /* 单行倍宽 */
    PrintOut (data [1],0);
    PrintOut (LQPrintData [20],LQPrintData [21] [0]); /* 英文压缩打印 */
    PrintOut (data [2],0);
    PrintOut (LQPrintData [22],LQPrintData [23] [0]);
}
```

打印结果

Double - Height Font  
**Double - Width Font**  
Condensed Font



## 选择英文字符的宽度和高度

该命令用给定的数值选择 ASCII 字符宽度和字符高度。

ESC X

选择英文字符的宽度和高度

ESC/P2

ESC/PK2

格式:	ASCII 码:	ESC	X	m	n1	n2
	十进制码:	27	88	m	n1	n2
	十六进制码:	1B	58	m	n1	n2

注释:

此命令根据下列公式选择 ASCII 字符的宽度和高度。

m = 0 : 不改变字符宽度

m = 1 : 选择比例打印

m > 5 : 选择字符宽度

字符宽度 =  $360/m$  cpi (每英寸字符数)

字符高度 =  $(n1 + n2 * 256) / 2$  point

1 point = 1/72 英寸

参数范围如下:

$$5 \leq m \leq 127, 0 \leq n1 \leq 255, 0 \leq n2 \leq 127$$

缺省值如下:

字符宽度 = 10 cpi (m = 36)

字符高度 = 10.5 point (n2 = 0, n1 = 21)

此命令不可与下列命令同时组合使用:

SO, ESC SO, ESC !, ESC SP, ESC W, FS SO, FS !, FS W

## 特殊打印效果

### 英文上标、下标打印

在英文模式中的上标和下标字体,一般都应用在注解、数学公式或化学注标等,用途相当广泛。

在此模式下的字符会以正常高度的 2/3 打印出,而且由指令来决定打印的位置是上方或下方(即上标或是下标)。

ESC S                    设定英文上 / 下标打印

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式：	ASCII 码：	ESC	S	n
	十进制码：	27	83	n
	十六进制码：	1B	53	n

注释：

n = 0 时，选择上标打印

n = 1 时，选择下标打印

ESC T                    解除英文上 / 下标打印

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式：	ASCII 码：	ESC	T
	十进制码：	27	84
	十六进制码：	1B	54

注释：

取消上标打印或下标打印。

注意事项

在 EPSON 扩充图形字符表中的 B0H 至 DFH 及 F0H 至 FEH 的范围中，设定上标 / 下标模式无效。(有关于 "EPSON 扩充图形字符表" 的说明，请参考第十三章字符表)

## 英文下划线模式

在 [下划线] 模式下，打印机可自动划出字符、空白和上标、下标的下划线。但在上 / 下标字符加下划线时，下划线便划在正常字符下方的位置。

ESC -                    设定 / 解除英文下划线

ESC/P    ESC/P2    ESC/PK    ESC/PK2

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

注释：

n = 1 时，设定英文下划线模式

n = 0 时，解除英文下划线模式

注意事项

. 此指令不可与水平定位 (HT)、设定绝对位置 (ESC \$) 和设定相对位置 (ESC 等指令连用, 否则无效。

. 在 EPSON 扩充图形字符中的 B0H 至 DFH、F4H、F5H 的范围中, 也无法使用下划线模式。

## 程序范例

```
#include "print5.h"
main ()
{
    unsigned char para [] [4] = {
        {0}, {1},
        {1}, {1}
    };
    unsigned char *data [] = {
        "Underline Font\n",
        "H",
        "2",
        "O",
        " Subscript\n",
        "X",
        " Superscript\n"
    };
    PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
    PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
    PrintOut (LQPrintData [12] ,LQPrintData [13] [0]);
    PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
    PrintOut (para [2] ,para [3] [0]);
    PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
    PrintOut (para [2] ,para [3] [0]);          /* 下划线模式 */
    PrintOut (data [0] ,0);
    PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
    PrintOut (para [0] ,para [1] [0]);
    PrintOut (data [1] ,0);
    PrintOut (LQPrintData [26] ,LQPrintData [27] [0]);
    PrintOut (para [2] ,para [3] [0]);          /* 设定下标打印 */
    PrintOut (data [2] ,0);
    PrintOut (LQPrintData [28] ,LQPrintData [29] [0]);
    PrintOut (data [3] ,0);
    PrintOut (LQPrintData [26] ,LQPrintData [27] [0]);
    PrintOut (para [2] ,para [3] [0]);
    PrintOut (data [4] ,0);
    PrintOut (data [5] ,0);
    PrintOut (LQPrintData [26] ,LQPrintData [27] [0]);
```

```

PrintOut (para [0] ,para [1] [0]);          /* 设定上标打印 */
PrintOut (data [2] ,0);
PrintOut (data [6] ,0);
PrintOut (LQPrintData [28] ,LQPrintData [29] [0]); /* 取消上下标 */
}

```

打印结果

Underline Font

H<sub>2</sub>O Subscript

x<sup>2</sup> Superscript

## 英文打印模式组合

英文打印模式组合是将 8 种打印形式，加以选择组合起来成一种打印模式表现出来。选择组合的打印形式计有：10 CPI 或 12 CPI 字间距、比例模式、压缩打印、粗体、重叠、倍宽、斜体字和下划线模式。

ESC !

英文打印模式组合

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式：	ASCII 码：	ESC	!	n
	十进制码：	27	33	n
	十六进制码：	1B	21	n

注释：

参数：

十进制：0 ≤ n ≤ 255

n 值是所选择的打印形式数值的总和（请参考下列打印模式组合表）

## 打印模式组合表

选择打印模式表				
设定值	数值 (十进制)	设定选择	设定打印模式	与设定值相等的 控制码
20	1	1	12 CPI	ESC M
		0	10 CPI	ESC P
21	2	1	设定比例模式	ESC p 1
		0	解除比例模式	ESC p 0
22	4	1	设定压缩模式	SI, ESC SI
		0	解除压缩模式	DC2
23	8	1	设定粗体模式	ESC E
		0	解除粗体模式	ESC F
24	16	1	设定重叠模式	ESC G
		0	解除重叠模式	ESC H
25	32	1	设定倍宽模式	ESC W 1
		0	解除倍宽模式	ESC W 0
26	64	1	设定斜体模式	ESC 4
		0	解除斜体模式	ESC 5
27	128	1	设定下划线模式	ESC - 1
		0	解除下划线模式	ESC - 0

### 打印模式组合范例

下面是一个将 12 CPI、倍宽、斜体字符和下划线模式四种打印形式组合成一种打印模式表现出来的计算实例。此 n 值是根据 [打印模式组合表] 中的模式设定值加以计算而得。

下式中, 设定值乘以 0 表示取消比例模式、压缩模式、粗体模式和重叠模式, 如下所示:

$$n = 1 + 0 + 0 + 0 + 0 + 32 + 64 + 128 = 225$$

$$n = 1 \times 1 + 2 \times 0 + 4 \times 0 + 8 \times 0 + 16 \times 0 + 32 \times 1$$

↑	↑	↑	↑	↑	↑
设定 12 CPI	解除比 例模式	解除压 缩模式	解除粗 体模式	解除重 叠模式	设定倍 宽模式

$$+ 64 \times 1 + 128 \times 1 = 255$$

↑            ↑  
设定斜     设定下划  
体模式     线模式

## 注意事项

.打印模式组合时无法配合使用 15 CPI 字符,只能在 10 CPI、12 CPI 或比例模式三项中选择一项。

## 程序范例

下面是一个组合 12CPI、下划线和重叠模式的打印范例：

```
#include "print5.h"
main ()
{
    unsigned char para [] [1] = {
        {0}, {1},
        {1}, {1},
        {145}, {1}
    };
    unsigned char *data="Underline+Double-strike+12pitch\n" ;

    PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
    PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
    PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
    PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
    PrintOut (para [2] ,para [3] [0]);          /* 选择文书体 */
    PrintOut (LQPrintData [30] ,LQPrintData [31] [0]);
    PrintOut ` (para [4] ,para [5] [0]);      /* 组合打印令 */
    PrintOut (data,0) ;
    PrintOut (LQPrintData [30] ,LQPrintData [31] [0]);
    PrintOut (para [0] ,para [1] [0]);
}
```

打印结果

Underline+Double-strike+12pitch

```

/*          print5.h          */

#include "stdio.h"

/*****
/* 变量定义          */
*****/
#define ESC          0x1B /* 模式设定 */
#define FS          0x1C /* 汉字模式设定 */
#define SI          0x0F /* 选择压缩打印 */
#define DC2          0x12 /* 解除压缩模式 */
#define SO          0x0E /* 设定倍宽打印 */
#define SP          0x20

void PrintOut () ;

/*****
/* 数据结构体设定          */
*****/
static unsigned char LQPrintData [] [10] = {
    {FS,'.'}, {2}, /* 设定英文模式 */
    {ESC,'x'}, {2}, /* 选择打印品质 */
    {ESC,'@'}, {2}, /* 初始化打印机 */
    {ESC,'k'}, {2}, /* 选择英文字体 */
    {ESC,'g'}, {2}, /* 选择 15CPI 字体 */
    {ESC,'M'}, {2}, /* 选择 12CPI 字体 */
    {ESC,'P'}, {2}, /* 选择 10CPI 字体 */
    {ESC,'2'}, {2}, /* 设定 1/6 行间距 */
    {ESC,'w'}, {2}, /* 设定 / 解除倍高打印 */
    {SO}, {1}, /* 设定单行倍宽 */
    {SI}, {1}, /* 设定压缩打印 */
    {DC2}, {1}, /* 解除压缩打印 */
    {ESC,'-'}, {2}, /* 设定 / 解除英文下划线 */
    {ESC,'S'}, {2}, /* 设定英文上 / 下标打印 */
    {ESC,'T'}, {2}, /* 解除英文上 / 下标打印 */
    {ESC,'!'}, {2}, /* 英文打印模式组合 */
    {ESC,'p'}, {2}, /* 设定 / 解除比例模式 */
    {ESC,SP}, {2} /* */
};

/*****
void PrintOut (unsigned char* lpcbData,int Length) /* 打印输出 */
{
    if (! Length) fprintf (stdprn,"%s",lpcbData) ;

```

```
else do
{
fprintf (stdprn,"%c",*lpcbData++);
Length--;
} while (Length);
}
```



## 第六章

### 中文模式下的打印命令

■ 中文模式的选择 .....	6-1
■ 设定汉字字间距 .....	6-3
■ 汉字放大、缩小打印 .....	6-6
汉字倍宽打印 .....	6-6
汉字四倍角打印 .....	6-6
汉字倍高打印 .....	6-7
汉字半角打印 .....	6-8
汉字上标、下标打印 .....	6-8
选择汉字点阵和汉字高度 .....	6-10
■ 选择汉字字体 .....	6-11
■ 特殊打印效果 .....	6-11
汉字纵向打印 .....	6-11
汉字下划线打印 .....	6-13
设定汉字基准线 .....	6-15
■ 中文表格符号连线 .....	6-15
■ 选择中文 CC-DOS 打印方式 .....	6-16
■ 汉字打印模式组合 .....	6-17

在中文模式下打印中文资料时,打印指令大多与英文模式下不同,只有少部分打印效果指令与英文模式下通用。因为 EPSON 中文打印机的缺省状态即为中文模式;所以,本章的打印命令在开机后可直接使用,无需再重新设定汉字模式。EPSON 英文打印机(无内置汉字库)不能使用本章的命令。有关汉字打印效果的指令其前引码必为 FS (ASCII 28);本章所述均为中文模式下的 ESC/PK 及 ESC/PK2 的指令。

### 中文模式的选择

用户打印中文资料时,如果打印机当前是在英文模式下,必须将打印机设定为中文模式;可直接使用 FS & 命令切换打印机为中文模式,如此打印机才可打印出汉字。

FS &

设定汉字模式

ESC/PK

ESC/PK2

格式: ASCII 码:

FS

&

十进制码：	28	38
十六进制码：	1C	26

FS . 解除汉字模式

ESC/PK

ESC/PK2

格式：	ASCII 码：	FS	.
	十进制码：	28	46
	十六进制码：	1C	2E

注释：

解除由 FS & 指令所设定的汉字模式，使打印机进入英文模式。

注意事项

.FS & 和 FS . 这两个指令可盖过 DIP 切换开关或 [选择模式] 的设定值。

## 程序范例

下面是中、英文字符在中文模式和英文模式下不同打印结果的比较。

```
#include "print6.h"
main ()
{
    unsigned char *data [] = {
        "---***--- Chinese mode ---***---\n",
        "中文之美",
        "---***--- English mode ---***---\n",
        "--'Seeing is Believing'\n"
    };

    PrintOut (LQPrintData [0],LQPrintData [1] [0]); /* 初始化打印机 */
    PrintOut (LQPrintData [2],LQPrintData [3] [0]); /* 设定汉字模式 */
    PrintOut (data [0],0);
    PrintOut (data [1],0); /* 中,文,之,美 */
    PrintOut (data [3],0);
    PrintOut (LQPrintData [4],LQPrintData [5] [0]); /* 解除汉字模式 */
    PrintOut (data [2],0);
    PrintOut (data [1],0);
    PrintOut (data [3],0);
}
```



注释：

调整半角字符间的间隔以使它们两个的宽度与一个中文字符的宽度相匹配。

FS V

取消半角字符间距调整

ESC/PK

ESC/PK2

格式：	ASCII 码：	FS	V
	十进制码：	28	86
	十六进制码：	1C	56

注释：

消除由 FS U 指令设定的半角字符的间距调整。

## 程序范例

以下是一个以 FS U 配合 FS V 使用比较的程序范例。若以 FS V 解除 FS U 指令后，半角字符便会无法对齐中文字符而差距也会加大。

```
#include "print6.h"

void fon1 ();
main ()
{
    unsigned char para [] [2] = {
        {0,3}, {2},
        {0,2}, {2},
    };
    unsigned char *data [] = {
        "---***---FS U Command Result---***---\n\n",
        "---***---FS V Command Result---***---\n\n",
        "中文之美",
        "ABCDEFGHJKLMNOPQRSTUVWXYZ\n\n"
    };
    PrintOut (LQPrintData [0],LQPrintData [1] [0]); /* 初始化打印机 */
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (data [0],0);
    PrintOut (LQPrintData [6],LQPrintData [7] [0]); /* 设定半角字符间距调整 */
    fon1 (data,para);

    PrintOut (data [1],0);
    PrintOut (LQPrintData [8],LQPrintData [9] [0]); /* 取消半角字符间距调整 */
    fon1 (data,para);
}
```

```

void fon1 (unsigned char * data [],unsigned char para [] [2])
{
    unsigned char i,j ;
    for (i=1 ;i<=10 ;i=i+4)
    {
        PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
        PrintOut (&i,1) ;
        j=i+1 ;
        PrintOut (&j,1) ;
        PrintOut (LQPrintData [12] ,LQPrintData [13] [0]);
        j=i/2 ;
        PrintOut (&j,1) ;
        j= (i+1) /2 ;
        PrintOut (&j,1) ;
        for (j=1 ;j<=3 ;j++)
        {
            PrintOut (data [2] ,0) ;
            PrintOut ("\n",0) ;
            PrintOut (data [3] ,0) ;
        }
        PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
        PrintOut (para [0] ,para [1] [0]);          /* 返回初始设定 */
        PrintOut (LQPrintData [12] ,LQPrintData [13] [0]);
        PrintOut (para [2] ,para [3] [0]);
    }
}

```

打印结果

---\*\*\* --- F S U Command Result ---\*\*\* ---

中文之美中文之美中文之美  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ

中文之美中文之美中文之美  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ

中文之美中文之美中文之美  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ

---\*\*\* --- F S V Command Result ---\*\*\* ---

中文之美中文之美中文之美  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ

中文之美中文之美中文之美  
ABCDEFGHIJKLMN OPQRSTUVWXYZ

中文之美中文之美中文之美  
ABCDEFGHIJKLMN OPQRSTUVWXYZ

## 汉字放大、缩小打印

### 汉字倍宽打印

在中文模式下，设定汉字的单行倍宽模式，可使用 **FS SO** 指令，亦可使用在中文和英文模式皆可行的 **ESC W 1** 倍宽打印指令。设定汉字倍宽模式后，中文字符以正常宽度的两倍打印。

FS SO	设定单行倍宽		ESC/PK	ESC/PK2
格式：	ASCII 码：	FS	SO	
	十进制码：	28	14	
	十六进制码：	1C	0E	

注释：

此指令被 DC4, LF, VT 或 FF 码中的任一个所消除，也会被 FS\_W 0 和 ESC W 0 指令消除。

FS DC4	解除单行倍宽		ESC/PK	ESC/PK2
格式：	ASCII 码：	FS	DC4	
	十进制码：	28	20	
	十六进制码：	1C	14	

注释：

解除由 **FS SO** 指令设定的汉字倍宽打印模式。此指令不能消除由 **FS W 1** 和 **ESC W 1** 指令选择的倍宽打印模式。

### 汉字四倍角打印

在英文模式下，可使用倍高加倍宽指令，产生四倍角的英文字形；而在中文模式下，可由 **FS W** 的设定，直接产生四倍放大的中文字形。

FS W	设定 / 解除四倍角打印			ESC/PK	ESC/PK2
格式：	ASCII 码：	FS	W	n	

十进制码：	28	87	n
十六进制码：	1C	57	n

注释：

设定/解除四倍角（双倍宽度，双倍高度）的中文模式。

n = 1，设定此模式；

n = 0，解除此模式。

依靠打印头的两次打印产生四倍角中文字符。

## 汉字倍高打印

因为这项功能，在中文模式下，并没有直接的指令设定，所以设定的方法比较特殊。

在设定的方法中，必须先以 **FS W 1** 设定四倍角中文，再以 **ESC W 0** 解除中文倍宽模式，才能得到双倍高度的中文打印字形。之后，只要再以 **FS W 0** 即可解除倍高模式。

## 程式范例

```
#include "print6.h"

main ()
{
    unsigned char para [] [3] = {
        {0}, {1},
        {1}, {1}
    };
    unsigned char *data=" 中文之美 ";

    PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
    PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
    PrintOut (LQPrintData [36] ,LQPrintData [37] [0]); /* 汉字单行倍宽 */
    PrintOut (data,0);
    PrintOut ("\n",0);
    PrintOut (LQPrintData [16] ,LQPrintData [17] [0]);
    PrintOut (para [2] ,para [3] [0]); /* 汉字四倍角 */
    PrintOut (data,0);
    PrintOut ("\n",0);
    PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
    PrintOut (para [0] ,para [1] [0]); /* 解除倍宽即为倍高模式 */
    PrintOut (data,0);
}
```

```

PrintOut (data,0) ;
PrintOut ("\n",0) ;
PrintOut (LQPrintData [16] ,LQPrintData [17] [0]);
PrintOut (para [0] ,para [1] [0]);
}

```

打印结果

中文之美  
中文之美  
中文之美中文之美

## 汉字半角打印

汉字半角模式是将全角汉字以半角字大小打印，亦即汉字宽度缩小为一半，而高度不变。

FS SI ESC/PK ESC/PK2

---

设定半角汉字

格式：	ASCII 码：	FS	SI
	十进制码：	28	15
	十六进制码：	1C	0F

注释：

设定汉字以半角字大小打印的模式。半角字符不受此指令影响。

FS DC2 ESC/PK ESC/PK2

---

解除半角汉字

格式：	ASCII 码：	FS	DC2
	十进制码：	28	18
	十六进制码：	1C	12

注释：

解除汉字半角（即 FS SI）及 1/4 角（即 FS r）的设定。

## 汉字上标、下标打印

用 FS r 指令，可将全角中文字符压缩成 1/4 角大小的中文字符。

FS r ESC/PK ESC/PK2

---

设定汉字上 / 下标打印



格式:	ASCII 码:	FS	r	n
	十进制码:	28	114	n
	十六进制码:	1C	72	n

注释:

选择 1/4 角字大小打印模式。

如 n = 1, 选择下标位置打印

n = 0, 选择上标位置打印

此指令可由 FS DC2 来解除。

## 程式范例

```
#include "print6.h"

main ()
{
    unsigned char para [] [3] = {
        {0}, {1},
        {1}, {1},
        };
    unsigned char *data=" 中文之美 ";

    PrintOut (LQPrintData [2] ,LQPrintData [3] [0]); /* 设定汉字模式 */
    PrintOut (data,0) ;
    PrintOut (LQPrintData [20] ,LQPrintData [21] [0]); /* 设定汉字半角 */
    PrintOut (data,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [22] ,LQPrintData [23] [0]); /* 解除汉字半角 */
    PrintOut (data,0) ;
    PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
    PrintOut (para [2] ,para [3] [0]); /* 设定汉字下标 */
    PrintOut (data,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [22] ,LQPrintData [23] [0]); /* 解除汉字下标 */
    PrintOut (data,0) ;
    PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
    PrintOut (para [0] ,para [1] [0]); /* 设定汉字上标 */
    PrintOut (data,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [22] ,LQPrintData [23] [0]); /* 解除汉字上标 */
}
```

## 打印结果

中文之美中文之美  
中文之美 中文之美  
中文之美 中文之美

## 选择汉字点阵和汉字高度

FS Y

选择汉字点阵和汉字高度

ESC/PK

ESC/PK2

格式： ASCII 码： FS Y m nl nh  
十进制码： 28 89 m nl nh  
十六进制码： 1C 59 m nl nh

### 注释：

此命令根据下列公式计算汉字宽度。

汉字宽度 =  $180/m$  CPI (每英寸字符数)

上述公式仅对汉字有效。

对于参数  $m$ , 缺省值为 27, 即：汉字宽度 =  $180/27 = 6.66$  CPI

可选择的汉字点阵及其相应参数如下：

nH	nL	字号	全角点阵	半角点阵
00H	20H	6	16 × 16	16 × 8
00H	28H	小 5	20 × 20	20 × 10
00H	30H	5	24 × 24	24 × 12
00H	38H	小 4	28 × 28	28 × 14
00H	40H	4	32 × 32	32 × 16
00H	48H	3	36 × 36	36 × 18
00H	50H	小 2	40 × 40	40 × 20
00H	60H	2	48 × 48	48 × 24
00H	70H	小 1	56 × 56	56 × 28
00H	80H	1	64 × 64	64 × 32

该命令不可与下列命令同时组合使用（当选 5 号字时除外）

SO, ESC SO, ESC, ESC !, ESC, W, FS SO, FS !, FS S,  
FS T, FS W, FS r

## 选择汉字字体

FS k

选择汉字字体

ESC/PK2

格式：	ASCII 码：	FS	k	n
	十进制码：	28	107	n
	十六进制码：	1C	6B	n

注释：

此命令用于选择汉字字体，有效字体则取决于打印机型号。

参数范围如下：

- n = 0 : 宋体
- n = 1 : 黑体
- n = 2 : 楷体
- n = 3 : 仿宋体

此命令解除控制面板的字体选择。

对于没有内置楷体和仿宋体汉字的打印机，参数 2 和 3 无效。参数的有效范围取决于打印机的内置字库。请参阅打印机手册。

## 特殊打印效果

### 汉字纵向打印

在中文的文字或公文资料处理上，时常需要使用汉字纵向打印，使用 FS J 和 FS K 就可以纵向打印出汉字或返回横向打印。

此外，常需要两个半角字符配合一个中文字一起垂直打印，如「中华人民共和国 88 年 12 月」，要设定垂直打印时，88 与 12 就必须在一个垂直中文字间隔中打印，而 FS D 就可达到这种效果。

FS J

设定纵向打印

ESC/PK

ESC/PK2

格式：	ASCII 码：	FS	J
	十进制码：	28	74
	十六进制码：	1C	4A

注释：

将跟在此指令后的所有字符（除了国标字符码表中 A9A4H ~ A9EFH 范围内的表格符号以外），皆以纵向打印模式打印出。

FS K

设定横向打印

ESC/PK

ESC/PK2

格式: ASCII 码: FS K  
 十进制码: 28 75  
 十六进制码: 1C 4B

注释:

取消 FS J 指令而恢复为水平汉字打印。

FS D

纵向半角两字符并列打印

ESC/PK

ESC/PK2

格式: ASCII 码: FS D  
 十进制码: 28 68  
 十六进制码: 1C 44

注释:

此指令在 FS J 设定为纵向打印时有效。在输入此命令后，一对相继输入的半角字符便以一个中文字的间隔纵向打印出。

## 程式范例

```
#include "print6.h"

main ()
{
    unsigned char *data [] = {
        "电话: 3824894\n",
        "94年",
        "12月\n"
    };

    PrintOut (LQPrintData [0],LQPrintData [1] [0]);
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (data [0],0);
    PrintOut (LQPrintData [26],LQPrintData [27] [0]);
    PrintOut (data [0],0);
    PrintOut (LQPrintData [28],LQPrintData [29] [0]);
    PrintOut (data [1],0);
    PrintOut (data [2],0);
    PrintOut (LQPrintData [26],LQPrintData [27] [0]);
    PrintOut (data [1],0);
}
```

```

PrintOut (data [2] ,0);
PrintOut (LQPrintData [30] ,LQPrintData [31] [0]);
PrintOut (data [1] ,0);
PrintOut (LQPrintData [30] ,LQPrintData [31] [0]);
PrintOut (data [2] ,0);
PrintOut (LQPrintData [28] ,LQPrintData [29] [0]);
}

```

## 打印结果

电话 : 3824894

电 话 : 3 8 2 4 8 9 4  
 9 4 年 1 2 月  
 9 4 年 1 2 月  
 94 年 12 月

## 汉字下划线打印

汉字下划线打印可分单下划线及双下划线打印,除了可加强打印效果外,还可配合纵向打印、压缩及上标、下标等效果一起使用。

当设定为下划线模式时,则经打印头的两次通过打印出中文字符,而底线是在第二次通过时打印出。在两种下划线设定中,纸张在第一次通过和第二次通过之间被送进了 4/180 英寸。此指令设定划出的底线是连续的。

当由 FS J 指令设定为纵向打印模式时,则下划线被打印在中文字符的右侧。

FS -	设定 / 解除汉字下划线			ESC/PK	ESC/PK2
格式:	ASCII 码:	FS	-	n	
	十进制码:	28	45	n	
	十六进制码:	1C	2D	n	

### 注释:

为中文字符设定下划线。

如 n=0, 解除下划线,

如 n=1, 设定 1 个点的下划线,

如 n=2, 设定 2 个点的下划线。

### 注意事项

- 在选择纵向打印时 (FS J), 国标字符码表中 A9A4H ~ A9EFH 范围内的表格符号不会旋转 90°。

- 选择下划线模式的指令 FS - 不可与 HT , ESC \$ , ESC 等指令连用, 否则无效。
- 设定表格符号连线时 (FS v), 国标字符码表中 A9A4H ~ A9EFH 范围内的表格符号, 无法使用下划线的功能。

## 程式范例

```
#include "print6.h"

main ()
{
    unsigned char para [] [3] = {
        {0}, {1},
        {1}, {1},
        {2}, {1},
    };
    unsigned char *data=" 中文之美 ";

    PrintOut (LQPrintData [0] ,LQPrintData [1] [0]); /* 初始化打印机 */
    PrintOut (LQPrintData [2] ,LQPrintData [3] [0]); /* 设定汉字模式 */
    PrintOut (LQPrintData [32] ,LQPrintData [33] [0]);
    PrintOut (para [2] ,para [3] [0]); /* 1 个点下划线 */
    PrintOut (data,0) ;
    PrintOut (LQPrintData [26] ,LQPrintData [27] [0]); /* 纵向打印 */
    PrintOut (data,0) ;
    PrintOut (LQPrintData [28] ,LQPrintData [29] [0]);
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [32] ,LQPrintData [33] [0]);
    PrintOut (para [4] ,para [5] [0]); /* 2 个点下划线 */
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
    PrintOut (para [2] ,para [3] [0]); /* 倍宽模式 */
    PrintOut (data,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [32] ,LQPrintData [33] [0]);
    PrintOut (para [0] ,para [1] [0]); /* 解除下划线 */
    PrintOut (LQPrintData [16] ,LQPrintData [17] [0]);
    PrintOut (para [0] ,para [1] [0]); /* 解除倍宽 */
}

```

打印结果

## 中文之美

### 设定汉字基准线

该命令用于设定汉字打印时的基准线。

FS b	设定汉字基准线			ESC/PK2
格式:	ASCII 码:	FS	b	n
	十进制码:	28	98	n
	十六进制码:	1C	62	n

注释:

为汉字设定基准线。

n = 0, "0": 基准线为要打印的最大字符的基准线。打印后根据字符和下划线, 基准线向下移动。

n = 1, "1": 基准线为当前位置下面 28/180 英寸处。打印后基准线不移动。

### 中文表格符连线

在 EPSON ESC/PK 系列打印指令中, 加强了中文表格符连线的效果, 使用 FS v 指令, 在汉字模式下, 便可划出连续不断的直线。

在本指令后的表格符号, 会自动延长而达到连接的效果。此表格符是指国标字符码表中 A9A4H ~ A9EFH 范围内的表格符号。当设定表格符连线时, 原本所做的字距、行距设定在 1/2 英寸内, 仍然有效。

FS v	设定 / 解除表格符连线			ESC/PK	ESC/PK2
格式:	ASCII 码:	FS	v	n	
	十进制码:	28	118	n	
	十六进制码:	1C	76	n	

注释:

n = 0 时, 解除表格符号连线,

n = 1 时, 设定表格符号连线。

## 注意事项

- 本指令效果下,纵向打印时,表格符号连线不会一起旋转 90° 角。
- 本指令不可与 ESC J 指令连用,否则无效。

## 打印范例

1. 未使用表格符号连线指令前的打印结果：

部门代号	员工编号	姓名
A - 001	1001	李大安
C - 003	1007	陈晓玲
B - 005	2001	方小英

2. 使用表格符号连线指令后的打印结果：

部门代号	员工编号	姓名
A - 001	1001	李大安
C - 003	1007	陈晓玲
B - 005	2001	方小英

## 选择中文 CC-DOS 打印方式

使用 CC-DOS 的控制代码,也能以同样的效果在 EPSON 打印机上打印出来。

ESC I

选择中文 CC-DOS 打印方式

ESC/PK2

格式：	ASCII 码：	ESC	I	n
	十进制码：	27	73	n
	十六进制码：	1B	49	n

注释：

对于 CC-DOS 用户来说,使用原来 CC-DOS 的控制代码,不用加驱动程序,也能以同样的效果在 EPSON 打印机上打印出来。



CC-DOS 控制代码：

横向放大、纵向放大、重复打印,以及组合方式和解除可根据以下 CC-DOS 控制代码实现。

	倍 宽	倍 高
ESC I A		
ESC I B	○	
ESC I C		○
ESC I D	○	○

### 汉字打印模式组合

此打印模式,可将 7 种中文打印形式,如纵向打印、半角打印、倍宽、四倍角、上/下标、下划线和倍高等模式,加以选择组合起来,以一种打印模式表示出来。

FS !

汉字打印模式组合

ESC/PK

ESC/PK2

格式：	ASCII 码：	FS	!	n
	十进制码：	28	33	n
	十六进制码：	1C	21	n

注释：

十进数： $0 \leq n \leq 255$

n 值是所选择之打印形式数值的总和。(请参考下列打印模式组合表)

## 中文打印模式组合表

打印模式组合				
设定值	十进制数值	设定值选择	设定打印模式	与设定值功能相等的控制码
2 的 0次幂	1	1	设定纵向打印	FS J
		0	设定横向打印	FS K
2 的 1次幂	2	1	设定汉字半角	FS SI
		0	解除汉字半角	FS DC2
2 的 2次幂	4	1	设定倍宽打印	ESC W 1
		0	解除倍宽打印	ESC W 0
2 的 3次幂	8	1	设定倍高打印	(FS W 1)+(ESC W 0)
		0	解除倍高打印	FS W 0
2 的 4次幂	16	1	设定上下标打印	FS r
		0	解除上下标打印	FS DC2
2 的 5次幂	32	1	设定上标打印	FS r 0
		0	设定下标打印	FS r 1
2 的 7次幂	128	1	设定下划线打印	FS - 1
		0	解除下划线打印	FS - 0

### 中文打印模式组合范例

下面是一个将倍宽、倍高组合成四倍角中文，再加上下划线模式，组合成一种打印模式表现出来的计算实例。此处 n 值的计算方法，和 ESC ! 指令的计算方法相同。

$$n = 4 \times 1 + 8 \times 1 + 128 \times 1 = 140$$

↑            ↑            ↑  
 设定倍宽 设定倍高 设定下划线

## 程式范例

```
#include "print6.h"

main ()
{
    unsigned char para [] [5] = {
        {0}, {1},
        {10}, {1},
        {137}, {1},
        {140}, {1},
        };
    unsigned char *data=" 中文之美\n" ;

    PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
    PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
    PrintOut (para [2] ,para [3] [0]);          /* 半角加倍高      */
    PrintOut (data,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
    PrintOut (para [0] ,para [1] [0]);
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
    PrintOut (para [4] ,para [5] [0]);        /* 倍高纵向打印加下划线 */
    PrintOut (data,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
    PrintOut (para [0] ,para [1] [0]);
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
    PrintOut (para [6] ,para [7] [0]);        /* 四倍角汉字加下划线  */
    PrintOut (data,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [34] ,LQPrintData [35] [0]);
    PrintOut (para [0] ,para [1] [0]);        /* 解除打印模式组合    */
    PrintOut ("\n",0) ;
}
```

中文之美

中文之美

中文之美

print.h 文件

```

/*          print6.h          */

#include "stdio.h"

/*****
/* 变量定义          */
/*****
#define ESC          0x1B /* 模式设定 */
#define FS          0x1C /* 汉字模式设定 */
#define SI          0x0F /* 选择压缩打印 */
#define DC2         0x12 /* 解除压缩模式 */
#define SO          0x0E /* 设定倍宽打印 */

void PrintOut ();
/*****
/* 数据结构体设定          */
/*****
static unsigned char LQPrintData [] [10] = {
    {ESC,'@'}, {2}, /* 打印机初始化 */
    {FS,'&'}, {2}, /* 设中文模式 */
    {FS,'.'}, {2}, /* 设英文模式 */
    {FS,'U'}, {2}, /* 设定半角字符间距调整 */
    {FS,'V'}, {2}, /* 取消半角字符间距调整 */
    {FS,'S'}, {2}, /* 设定全角汉字字间距 */
    {FS,'T'}, {2}, /* 设定半角字附间距 */
    {FS,'SO'}, {2}, /* 设定单行倍宽 */
    {FS,'W'}, {2}, /* 设定 / 解除四倍角打印 */
    {ESC,'W'}, {2}, /* 设定 / 解除倍宽打印 */
    {FS,'SI'}, {2}, /* 设定汉字半角 */
    {FS,'DC2'}, {2}, /* 解除汉字半角 */
    {FS,'r'}, {2}, /* 设定汉字上 / 下标打印 */

```

```

{FS,'J'}, {2}, /* 设定纵向打印 */
{FS,'K'}, {2}, /* 设定横向打印 */
{FS,'D'}, {2}, /* 纵向半角两字符并列打印 */
{FS,'-'}, {2}, /* 设定解除汉字下划线 */
{FS,'!'}, {2} /* 汉字打印模式组合 */
};

/*****
void PrintOut (unsigned char* lpcbData,int Length) /* 打印输出 */
{

    if (!Length) fprintf (stdprn,"%s",lpcbData);
    else
        do
        {
            fprintf (stdprn,"%c",*lpcbData++);
            Length--;
        } while (Length);
}

```

## 中、英文模式下通用的打印功能

■ 选择字体形态 .....	7-1
打印中空、阴影、立体字形 .....	7-1
打印斜体字形 .....	7-3
■ 特殊打印效果 .....	7-5
粗体和重叠打印 .....	7-5
设定倍宽打印 .....	7-6
选择字符划线 .....	7-8
背景网点及字内网点 .....	7-8

在 ESC/P 控制码中,有些控制打印指令,突破了中、英文模式的限制,都可应用在中、英文打印模式下,因而提供了更多样变化的字体效果和更丰富的版面设计;除了可打印出放大、缩小等一般效果,更加强了中空、阴影、立体、背景网点及字内网点、纵向打印、彩色等变化,这将使您打印出来的文字、报表,更加生动活泼。

### 选择字体形态

#### 打印中空、阴影、立体字形

在 EPSON 24 针点阵打印机中内置的中文字体,在中文模式下,用户可做多种形式上的变化。而在英文模式下,内置的所有的英文文书字体,也都可做此三种字形的变化。

ESC q	选择字体形态	ESC/P	ESC/P2	ESC/PK	ESC/PK2
-------	--------	-------	--------	--------	---------

格式:	ASCII 码:	ESC	q	n
	十进制码:	27	113	n
	十六进制码:	1B	71	n

注释:

- n = 0 时, 选择正常字形
- n = 1 时, 选择中空字形
- n = 2 时, 选择阴影字形
- n = 3 时, 选择立体字形

#### 程式范例

```
#include "print7.h"

main ()
```

```

int i ;
unsigned char para [] [5] = {
    {0}, {1},
    {1}, {1},
    {2}, {1},
    {3}, {1},
    };
unsigned char *data [] = {
    "Outline\n",
    "Shadow\n",
    "Outline with Shadow\n",
    " 中文之美\n"
};

```

```

PrintOut (LQPrintData [0] ,LQPrintData [1] [0]); /* 初始化打印机 */
PrintOut (LQPrintData [2] ,LQPrintData [3] [0]); /* 解除汉字模式 */
PrintOut (LQPrintData [4] ,LQPrintData [5] [0]);
PrintOut (LQPrintData [6] ,LQPrintData [7] [0]);
PrintOut (LQPrintData [8] ,LQPrintData [9] [0]);
PrintOut (para [2] ,para [3] [0]); /* 设定文书体 (LQ) */
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (para [2] ,para [3] [0]); /* 设定中空字形 */
PrintOut (data [0] ,0);
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (para [4] ,para [5] [0]); /* 设定阴影字形 */
PrintOut (data [1] ,0);
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (para [6] ,para [7] [0]); /* 设定立体字形 */
PrintOut (data [2] ,0);
PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
PrintOut (LQPrintData [12] ,LQPrintData [13] [0]); /* 设定汉字模式 */
PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
PrintOut (para [2] ,para [3] [0]);
for (i=1 ;i<=3 ;i++)
{
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (para [i*2] ,para [i*2+1] [0]); /* 选择字体形态 */
PrintOut (data [3] ,0); /* 中文之美 */
}
PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
PrintOut (para [0] ,para [1] [0]);
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (para [0] ,para [1] [0]);
}

```

## 打印结果

Outline  
Shadow  
Outline with Shadow  
中文之美  
中文之美  
中文之美

## 打印斜体字形

ESC 4

设定斜体打印

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式： ASCII 码：       ESC       4  
          十进制码：       27       52  
          十六进制码：     1B       34

注释：

以斜体字的形态来打印出字符。

ESC 5

解除斜体打印

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式： ASCII 码：       ESC       5  
          十进制码：       27       53  
          十六进制码：     1B       35

注释：

取消 ESC 4 所选择的斜体字形。

## 注意事项

1. 打印下列范围内的字符时, 选择字体形态的指令无效:

- 在英文模式下, EPSON 扩充图形字符表中 B0H 至 DFH, 及 F4H, F5H 范围内的图形字符。
- 在汉字模式下, 汉字国标字符码表中 A9A4H ~ A9EFH 范围内的表格符号。

2. 汉字纵向打印模式下, 设定斜体打印的指令无效。

## 程式范例

```
#include "print7.h"
```



```

main ()
{
    unsigned char para [5] = {
        {0}, {1},
        {1}, {1},
        };
    unsigned char *data [] = {
        "Italic font\n\n",
        "中文之美中文之美\n\n"
    };

    PrintOut (LQPrintData [0],LQPrintData [1] [0]);
    PrintOut (LQPrintData [4],LQPrintData [5] [0]);
    PrintOut (LQPrintData [2],LQPrintData [3] [0]);
    PrintOut (LQPrintData [6],LQPrintData [7] [0]);
    PrintOut (LQPrintData [16],LQPrintData [17] [0]); /* 设定斜体打印 */
    PrintOut (data [0],0);
    PrintOut (LQPrintData [32],LQPrintData [33] [0]);
    PrintOut (para [2],para [3] [0]); /* 设定倍高打印 */
    PrintOut (data [0],0);
    PrintOut (LQPrintData [32],LQPrintData [33] [0]);
    PrintOut (para [0],para [1] [0]);
    PrintOut (LQPrintData [12],LQPrintData [13] [0]);
    PrintOut (data [1],0); /* 中文之美 */
    PrintOut (LQPrintData [14],LQPrintData [15] [0]);
    PrintOut (para [2],para [3] [0]); /* 设定四倍角打印 */
    PrintOut (LQPrintData [18],LQPrintData [19] [0]);
    PrintOut (para [0],para [1] [0]);
    PrintOut (data [1],0); /* 中文之美 */
    PrintOut (LQPrintData [20],LQPrintData [21] [0]); /* 解除斜体打印 */
    PrintOut (LQPrintData [14],LQPrintData [15] [0]);
    PrintOut (para [0],para [1] [0]);
}

```

打印结果

*Italic font*

***Italic font***

中文之美中文之美

## 特殊打印效果

## 粗体和重叠打印

当您需要加强打印文字的颜色时,可利用软件指令来控制打印头的两次打印而使文字的颜色加深,因而产生粗体和重叠字体。另外,如果用户还想要获得线条更粗,颜色更深的字体,还可结合粗体模式和重叠模式两种效果,而使字符显得更粗黑明显。

ESC E                      设定粗体打印

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	E
	十进制码:	27	69
	十六进制码:	1B	45

注释:

在粗体模式下,每个点都被打印两次;其中,第二次是稍微靠第一次的右边打印出。由于每个点都要打印两次。因此,打印头的速度便变慢;另外,粗体模式也可以和重叠模式联用。

ESC F                      解除粗体打印

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	F
	十进制码:	27	70
	十六进制码:	1B	46

注释:

取消 ESC E 所设定的粗体模式。

ESC G                      设定重叠打印

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	G
	十进制码:	27	71
	十六进制码:	1B	47

注释:

在重叠模式下,每一个点也是被打印两次;其中,第二次是纸张向前推进 1/180 英寸,而稍微在第一次的下方印出。同样的,重叠模式也可以和粗体模式联用。

ESC H                      解除重叠打印                      ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式：  ASCII 码：      ESC      H  
           十进制码：      27      72  
           十六进制码：     1B      48

注释：  
       解除 ESC G 所设定的重叠模式。

## 设定倍宽打印

在 ESC/P 打印指令中, 设有多项的倍宽打印指令, 必须注意的是: SO 指令使用在英文模式下, 才可以设定单行倍宽; FS SO 和 FS DC4 则在中文模式下, 才可设定和取消倍宽。而以下的 ESC W, ESC SO 和 DC4 等指令, 不论在中文或英文模式下, 都可以设定字符的倍宽。

ESC W                      设定 / 解除倍宽打印                      ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式：  ASCII 码：      ESC      W      n  
           十进制码：      27      87      n  
           十六进制码：     1B      57      n

注释：  
       n = 1 时, 设定倍宽打印  
       n = 0 时, 解除倍宽打印

不论在中文或英文模式下, 皆可将所有的字符以双倍的宽度打印出。

ESC SO                      设定单行倍宽                      ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式：  ASCII 码：      ESC      SO  
           十进制码：      27      14  
           十六进制码：     1B      0E

注释：  
       可将所有的字符以双倍的宽度打印出; 此一倍宽模式可被回车 (CR) 或 DC4 所取消。

DC4                      解除单行倍宽                      ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式：  ASCII 码：      DC4  
           十进制码：      20  
           十六进制码：     14

注释：

解除 SO 或 ESC SO 所设定的单行倍宽模式；但不能解除 ESC W 或 ESC ! 所设定的倍宽模式。

## 程式范例

```
#include "print7.h"

main ()
{
    unsigned char para [] [5] = {
        {0}, {1},
        {1}, {1},
        };
    unsigned char *data [] = {
        "Emphasized Font\n",
        "Double-strike\n",
        "Double-Width Font\n",
        "中文之美\n",
        "中文之美中文之美\n"
    };
    PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
    PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
    PrintOut (LQPrintData [4] ,LQPrintData [5] [0]);
    PrintOut (LQPrintData [6] ,LQPrintData [7] [0]);
    PrintOut (LQPrintData [8] ,LQPrintData [9] [0]);
    PrintOut (para [2] ,para [3] [0]);
    PrintOut (LQPrintData [22] ,LQPrintData [23] [0]); /* 粗体打印 */
    PrintOut (data [0] ,0);
    PrintOut (LQPrintData [24] ,LQPrintData [25] [0]);
    PrintOut (LQPrintData [26] ,LQPrintData [27] [0]); /* 重叠打印 */
    PrintOut (data [1] ,0);
    PrintOut (LQPrintData [28] ,LQPrintData [29] [0]);
    PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
    PrintOut (para [2] ,para [3] [0]); /* 倍宽打印 */
    PrintOut (data [2] ,0);
    PrintOut (LQPrintData [12] ,LQPrintData [13] [0]); /* 设定汉字模式 */
    PrintOut (data [3] ,0);
    PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
    PrintOut (para [0] ,para [1] [0]);
    PrintOut (data [4] ,0);
}
```

Emphasized Font  
Double-strike  
**Double - Width Font**  
中文之美  
中文之美中文之美

### 选择字符划线

此指令可根据设定值在所有字符及空格上划出上线、中线及下线。

ESC (-	选择字符划线	ESC/P	ESC/P2	ESC/PK2					
格式:	ASCII 码:	ESC	(	-	n1	n2	m	d1	d2
	十进制码:	27	40	45	n1	n2	m	d1	d2
	十六进制码:	1B	28	2D	n1	n2	m	d1	d2

注释:

此一指令根据下列参数,在所有字符和空格上划线。

参数范围如下:

n1 = 3

n2 = 0

m = 1

d1 = 1 : 下线

d1 = 2 : 中线

d1 = 3 : 上线

d2 = 0 : 解除划线

d2 = 1 : 一条实线

d2 = 2 : 二条实线

d2 = 5 : 一条虚线

d2 = 6 : 二条虚线

### 背景网点及字内网点

此指令可设定 7 种颜色的 4 种背景网点和多种字底内部套网。自行选择搭配的结果,将使报表文字更加醒目生动。

此指令在指定网点颜色值时,必须注意在具备有彩色配备的机型上,才可做色彩的选择,在一般

单色的机型上，都指定黑色设定值。

ESC ( X

设定 / 解除网点打印

ESC/PK2

格式： ASCII 码： ESC ( X n1 n2 a1 a2 a3  
十进制码： 27 40 88 n1 n2 a1 a2 a3  
十六进制码： 1B 28 58 n1 n2 a1 a2 a3

注释：

此一指令以 ESC, ( 和 X 这三个字符为指令序列的开始。其余参数功能如下：

n1 = 3, n2 = 0 固定值

a1 : 指定网点区域

a1 = 0 : 选择背景网点。

a1 = 1 : 选择字体内部网点。

当 a1 = 1 时，须配合字体形态指令使用；亦即只在 (ESC q 1) ,  
(ESC q 2) 及 (ESC q 3) 等指令后 a1 = 1 才有效。

a2 : 选择网点的样式

0 : 解除网点            1 : 实心网点

2 : 点状网点            3 : 斜线网点

4 : 交叉线网点

a3 : 选择套网的颜色

0 : 黑色    1 : 红色    2 : 蓝色

3 : 紫色    4 : 黄色    5 : 橙色

6 : 绿色

注意事项

- 斜体字不能配合此打印指令一起使用。
- 以 HT, ESC \$, ESC 这三个指令来调整字符间距时, 此打印功能无效。
- 在汉字模式下, 此打印指令不能和“表格符号连线”指令连用, 否则无效。

程序范例

以下是以四倍角中空字形加斜线字底, 所做的程序范例。

```
#include "print7.h"

main ()
{
    unsigned char para [] [15] = {
        {1}, {1},
        {3,0,1,3,0}, {5},
```

```

        {3,0,1,0,0}, {5}
};
unsigned char *data=" 四倍中空字形加斜线字底\n" ;

PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
PrintOut (LQPrintData [12] ,LQPrintData [13] [0]);
PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
PrintOut (para [0] ,para [1] [0]);          /* 四倍角汉字 */
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (para [0] ,para [1] [0]);          /* 中空字形 */
PrintOut (LQPrintData [30] ,LQPrintData [31] [0]);
PrintOut (para [2] ,para [3] [0]);
PrintOut (data,0);                          /* 字内斜线网点 */
PrintOut (LQPrintData [30] ,LQPrintData [31] [0]);
PrintOut (para [4] ,para [5] [0]);          /* 解除字内网点 */
}

```

打印结果

## 四倍中空字形加斜线字底

```

/*      print7.h      */

#include "stdio.h"

/*****
/* 变量定义          */
*****/
#define ESC      0x1B /* 模式设定 */
#define FS      0x1C /* 汉字模式设定 */

void PrintOut () ;
/*****
/* 数据结构体设定          */
*****/
static unsigned char LQPrintData [] [20] = {
    {ESC,'@'}, {2}, /* 打印机初始化 */
    {FS,'.'}, {2}, /* 设定英文模式 */
    {ESC,'P'}, {2}, /* 选择 10CPI 字体 */

```

```

{ESC,'2'}, {2}, /* 设定 1/6 英寸行间距 */
{ESC,'x'}, {2}, /* 选择打印品质 */
{ESC,'q'}, {2}, /* 选择字形态 */
{FS,'&'}, {2}, /* 设定汉字模式 */
{FS,'W'}, {2}, /* 设定 / 解除四倍角打印 */
{ESC,'4'}, {2}, /* 设定斜体打印 */
{ESC,'W'}, {2}, /* 设定解除倍宽打印 */
{ESC,'5'}, {2}, /* 解除斜体打印 */
{ESC,'E'}, {2}, /* 设定粗体打印 */
{ESC,'F'}, {2}, /* 解除粗体打印 */
{ESC,'G'}, {2}, /* 设定重叠打印 */
{ESC,'H'}, {2}, /* 解除重叠打印 */
{ESC,'(','X'), {3}, /* 设定 / 解除网点打印 */
{ESC,'w'}, {2} /* 设定英文倍高打印 */
};

/*****/
void PrintOut (unsigned char* lpcbData,int Length) /* 打印输出 */
{
    if (!Length) fprintf (stdprn,"%s",lpcbData);
    else do
    {
        fprintf (stdprn,"%c",*lpcbData++);
        Length--;
    } while (Length);
}

```



## 图象打印

■ 何谓图象打印 .....	8-1
■ 图象打印模式的选择 .....	8-1
8 针图象打印模式 .....	8-1
24 针图象打印模式 .....	8-2
■ 图象资料的解释 .....	8-2
图案的对应值 .....	8-2
点图案的设计 .....	8-3
■ 各种图象打印的密度 .....	8-4
■ 图象打印指令序列 .....	8-4
图象打印模式的设定 .....	8-4
图象密度的改变 .....	8-5
图象打印模式的相互转换 .....	8-6

### 何谓图象打印

在图象模式下,打印机可以产生图形、影像、条状图等电脑的屏幕上所能产生的各种显示图形。由于 ROM 中的标准字符,或是预先定义好的图形字符都不足以产生复杂的图形影像。因此,若要产生复杂的图形时,便必须使用图象模式来控制各个要打印出的点,以建立复杂的图形。

一般 24 针的打印机能在图象模式下绘出各种点图案,但是绘出图案最快和最简单的方法还是利用绘图软件,先在屏幕上设计出一些图案;然后,再通过绘图指令以类似“屏幕打印”的方式来打印出结果。值得一提的是,上述这种绘图方式便是在图象模式下工作的。假如您是使用绘图软件的话,您要了解的只是如何使用该软件;但是您若想自己设计图形程序或想进一步了解打印机处理图形的过程,请参考下面几节。

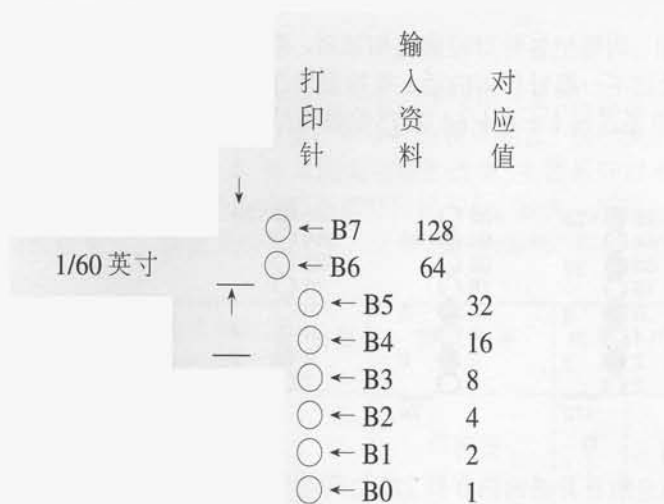
### 图象打印模式的选择

在图象打印模式下,打印机并不是一次接收一个字符资料来打印。而是一次接收一行的点阵资料。虽然说,在这模式下您必须输入更多的资料,但您可以直接控制“点”的打印位置以形成图案。意即您可以完全掌握打印头每一根打印针的动作情形。

### 8 针图象打印模式

一般 24 针的打印机能以 8 根针来绘出图象,意即打印机是用每第三根针来打印的。打印头的 8 根打印针都分别对应图象数据一个字节 (Byte) 的一个位 (Bit)。八个位构成图象数据的一个字节资料,因此这些针可打印出最多 8 个垂直点的线,而点直线中每一根打印针的位置,都有一个对应位

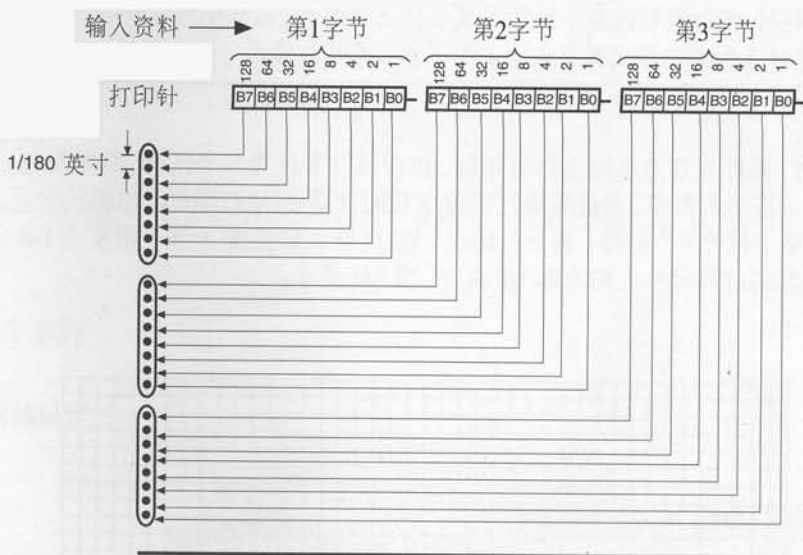
值,如图一所示。



(图一) 60点/英寸的垂直点密度

## 24针图象打印模式

打印机以全部的24根针来绘图,而这些针分为3组,由3个8位字节的数据来控制,以形成最多24个垂直点的线。而每一根打印针的位置,都有一对应位值。其关系如图二所示:



(图二) 180点/英寸的垂直点密度

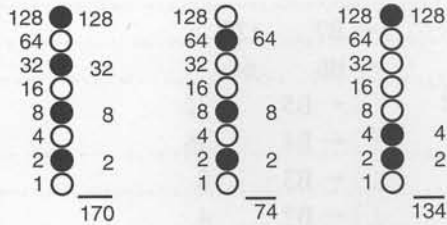
## 图形资料的解释

### 点图案的对应值

一旦明白了打印针和其对应值间的关系后,下一个步骤便是如何告诉打印机该激发那几根针。事

实上要使哪一根针撞击的话，只需利用 ESC \* 指令送出它的对应值即可。

如果同时要使几根针有作用，则需把各针对应数值相加后，把总和送给打印机。这些数值便是点图案所对应的值，例如：要使最底下一根针作用的话，电脑便要送出 1 的数值给打印机。另外，要使 1、8、64 这三根针作用的话，便要送出  $1 + 8 + 64 = 73$  的数值给打印机。下面这三个例子将告诉您如何计算出正确的数值。



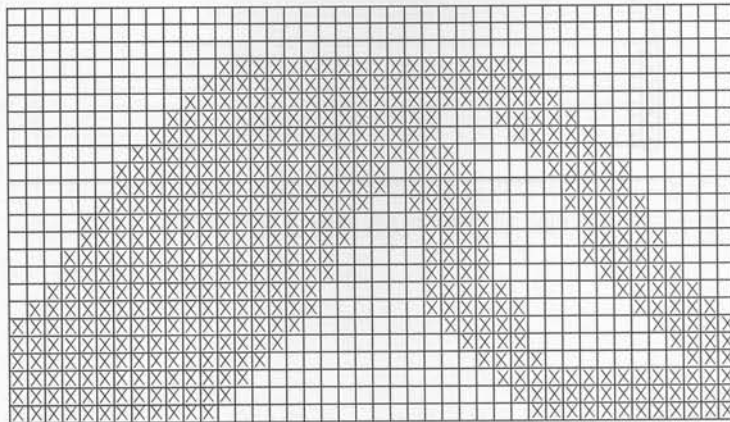
以这个数字关系，您可以任意组合并送出由 0 到 225 之间的十进数，且没有数值会有重复的现象。

又譬如说在 24 针图象打印模式时，您必须仔细计算每一个字节的数值，而每一列共有 3 个字节。另外，在三倍密度（180 DPI）的图象打印模式下每英寸有 180 列，要打印出一英寸长的图形则需送出 540 个字节的图象资料。所以说在绘制点图案的过程中，需要处理相当多的计算。所幸的是，绘图软件能帮您计算这些数值。

## 点图案的设计

根据以上的解说，相信您可以着手设计些属于自己的图形，虽然图形资料的计算非常繁琐。但是它的确可以帮助您对点图案的形成有更进一步的了解，尤其在设计一些常用的小图形元件时，更是有用。

在设计图形时，最好先在方格纸上打下草稿，接着再以 8 行为一个区段，将图形划分成适当的区段，然后再翻译成一连串的数值，通过图象打印指令 ESC \* 传递给打印机。如图三所示，便是一个 24 \* 42 点的图案，方格中打“×”记号，表示打印点，在 24 针三倍密度（水平密度为 180 点/英寸）图象模式下，图形实际的打印尺寸大约为高 1/8 英寸，宽 1/4 英寸。



(图三) 24 \* 42 点图案

## 各种图象打印的密度

图象输出的分辨率是由水平点密度和垂直点密度所决定的。而点与点之间的垂直距离是由打印头上打印针的距离来决定，因此无法改变。对 8 针的模式而言，这一固定的垂直距离是每英寸 60 点，而对 24 针的模式则是每英寸 180 点，所以图案密度的改变，主要是针对水平点密度而言，在 8 针与 24 针的模式下，分别可以选择多种的点间距离来打印出图象，如表一所示。不同的点间距离意味着输出的图形将有不同的纵横比。因此某些模式下，呈现圆形的图形，在另一些模式下，可能会成为椭圆形。

垂直点密度	水平点密度 (点/英寸)	针数	模式名称	m 值	代用指令	每列 字节数
60点/英寸	60	8	单密度	0	ESC K	*1
	120	8	双密度	1	ESC L	1
	120	8	高速双密度*	2	ESC Y	1
	240	8	四倍密度*	3	ESC Z	1
	80	8	CRT 图形	4	无	1
	90	8	CRT 图形II	6	无	1
180点/英寸	60	24	单密度	32	无	3
	120	24	双密度	33	无	3
	90	24	图形II	38	无	3
	180	24	CRT三倍密度	39	无	3
	360	24	六倍密度*	40	无	3

(表一)

注：打“\*”记号的模式不能打印出邻接点，意即每一个被打印出的点，它的左右必须是空白点。

## 图象打印指令序列

### 图象打印模式的设定

ESC \*                      选择图象模式

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式：    ASCII 码：    ESC    \*    m    n1    n2    d1    d2 ... di  
           十进制码：    27    42    m    n1    n2    d1    d2 ... di  
           十六进制码：    1B    2A    m    n1    n2    d1    d2 ... di

参数：

十进制数    : m = 0, 1, 2, 3, 4, 6, 32, 33, 38, 39, 40  
                    $0 \leq n1 \leq 255, 0 \leq n2 \leq 31$

$$0 \leq d_i \leq 255$$

十六进制数：m = 00H, 01H, 02H, 03H, 04H, 06H, 20H, 21H, 26H, 27H, 28H

$$00H \leq n_1 \leq FFH, 00H \leq n_2 \leq 1FH,$$

$$00H \leq d_i \leq FFH$$

注释：

- 用户可根据所选择的图象输出密度（请参考表一）来设定 m 值。
- 图形的总列数  $N = n_1 + n_2 * 256$ ，在 BASIC 中可用下列的计算公式来表示：

$$n_1 = N \text{ MOD } 256$$

$$n_2 = \text{INT} (N/256)$$

- $d_i$  表示输入的图象数据， $n_1, n_2$  决定要送出多少个图象数据，对 24 点模式而言，每一列需送出 3 个字节的资料，因此列数的 3 倍便是所需的图象数据字节数。

### 图象密度的改变

在 8 针的图象状态下，除了使用 ESC \* m n1 n2 指令来选择不同密度的绘图模式外，还可单独使用 ESC K、ESC L、ESC Y、ESC Z 这几个指令来选择绘图模式。表二所列为这些指令和 ESC \* 指令之间的关系。

指 令	绘图模式	ESC * 格式
ESC K	单密度	ESC * 0
ESC L	双密度	ESC * 1
ESC Y	高速双密度	ESC * 2
ESC Z	四密度	ESC * 3

ESC Y                      8 点高速双密度图象模式

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式： ASCII 码：        ESC        Y        n1        n2  
           十进制码：        27        89        n1        n2  
           十六进制码：        1B        59        n1        n2

注释：

此一指令可选择高速双密度的图象模式。

其中，所需的总列数 =  $n_1 + n_2 \times 256$

ESC Z                      8 点四密度图象模式

ESC/P    ESC/P2    ESC/PK    ESC/PK2

格式： ASCII 码：        ESC        Z        n1        n2  
           十进制码：        27        90        n1        n2  
           十六进制码：        1B        5A        n1        n2

注释：

此一指令可选择四密度的图象模式。

其中，所需的总列数 =  $n1 + n2 \times 256$

ESC K                    8点单密度图象模式

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式：	ASCII 码：	ESC	K	n1	n2
	十进制码：	27	75	n1	n2
	十六进制码：	1B	4B	n1	n2

注释：

此一指令可选择单密度的图象模式。

其中，所需的总列数 =  $n1 + n2 \times 256$

ESC L                    8点双密度的图象模式

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式：	ASCII 码：	ESC	L	n1	n2
	十进制码：	27	76	n1	n2
	十六进制码：	1B	4C	n1	n2

注释：

此一指令可选择双密度的图象模式。

其中，所需的总列数 =  $n1 + n2 \times 256$

## 图象打印模式间的相互转换

还有一个指令能让您直接由 ESC K、ESC L、ESC Y、ESC Z 指令转换为 ESC \* 指令。这个指令就是

ESC ?                    重新指定图象模式

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式：	ASCII 码：	ESC	?	s	m
	十进制码：	27	63	s	m
	十六进制码：	1B	3F	s	m

注释：

可将 ESC K、ESC L、ESC Y 和 ESC Z 中的任何一种图象模式重新指定为 ESC \* 指令下的任何一种图象打印模式。其中，变量 s 是 K、L、Y 或 Z 的 ASCII 码，以指出被重新定义的指令为何种模式；而变量 m 则表示 ESC \* 指令下的模式 m。

例如：想把由 ESC K 所选择的单密度图象模式转换成 CRT 图形 I 的绘图模式，则 C 指令如下：

```
fprintf (stdprn, "%c%c%c%c", 27, '?', 'K', 4);
```

这是改变图形比例的最快方法。改变绘图模式只会改变图形的宽度；图形高度并不会改变。因此，做此转换时要格外小心。

如果您没有改变程序中的绘图资料就把 8 针的绘图模式转换成 24 针的话，您就打印不出任何图形了。请记住 24 针绘图模式所需的数据量为 8 针绘图模式的三倍。

### 注意事项

- 在 8 针的图象模式下，您可以使用为 EPSON FX 和 RX 系列 9 针打印机所写的绘图软件。
- 通常，用户需要多行的图象数据资料来形成一个较大的输出。如果用户使用的是正常的行间距（1/6 英寸）的话，那么，图形便不会连接在一起，而会留下空格。此时，便要使用 ESC 3 n 的指令来改变行间距，以克服上述的问题；其中，n 的值要为 24。
- 在某些情况下，本来要打印图形的地方，可能会打印出文字；而在另一些情况下，本来要打印文字的地方，可能会打印出图形。如下所示，便是一些常见的原因：
  - ① 图象数据字节数大于规定的数量。
  - ② 图象数据已经在该行的结束之处，停止打印；多余的文字资料则接在下一行打印出。
  - ③ 数据的列数并无错误；但所送的资料字节数则为列数的 3 倍；这是因为数据字节数是在 24 点的模式下计算的，而所用的指令则是 8 点模式的指令。
- 某些电脑在送出固定数目的字符后，便会送出一个回车字符和一个换行字符。在这种情况下，打印机会将这些字符视为图象资料的一部份，而造成输出的错误。此时，利用 WIDTH LPRINT 255 这样一个宽度的叙述，便可将其消除。
- 如果在图象资料尚未完全送完之前，电脑程序便被中断的话；或者是 n1 和 n2 的值不正确的话，打印机都会继续等待资料的输入。这种情况下就好像打印机暂停执行一样。

### 程式范例及说明

下面是一个 C 程序，图案的设计如前面图三所示，您将能了解在 C 语言下如何定义绘图指令和图案的数值。

```
#include "print8.h"

main ()
{
    unsigned char para [] [4] = {{39,42,0}, {3} };
    unsigned char data [126] = {
        0,0,63,0,0,127,0,0,      /* 图象数据 */
        255,0,3,255,0,15,255,0,
```

```

31,255,0,127,255,0,255,255,
1,255,255,3,255,255,7,255,
255,15,255,255,31,255,254,31,
255,252,31,255,248,31,255,240,
31,255,224,31,255,192,31,255,
0,31,252,0,31,240,0,31,
224,0,31,128,0,31,240,0,
31,255,192,28,255,224,28,127,
240,28,15,248,30,0,252,31,
0,126,15,128,15,7,192,7,
3,240,7,1,254,7,0,255,
7,0,127,135,0,31,199,0,
7,231,0,1,247,0,0,255,
0,0,127,0,0,63};

```

```


PrintOut (LQPrintData [0] , LQPrintData [1] [0]);
PrintOut (LQPrintData [2] , LQPrintData [3] [0]);
PrintOut (para [0] ,para [1] [0]);
PrintOut (data,126) ;

```

```

}

```

打印结果 

```

/*          print8.h          */

#include "stdio.h"

/*****/
/* 变量定义          */
/*****/
#define ESC          0x1B /* 模式设定          */
void PrintOut () ;
/*****/
/* 数据结构体设定          */
/*****/
static unsigned char LQPrintData [] [3] = {

    {ESC,'@'}, {2}, /* 打印机初始化 */
    {ESC,'*'}, {2} /* 选择图象模式 */
};
/*****/
void PrintOut (unsigned char * lpcbData,int Length) /* 打印输出 */
{
    if (! Length) fprintf (stdprn,"%s",lpcbData) ;

```



```
    else do
    {
        fprintf (stdprn,"%c",*lpcbData++);
        Length--;
    } while (Length);
}
```

## 用户自定义字符

■ 用户自定义字符功能原理 .....	9-1
■ 英文用户自定义字符的设计及使用 .....	9-1
■ 中文用户自定义字符的设计及使用 .....	9-6
■ 用户自定义字符注意事项 .....	9-10

打印机提供一些指令，能将用户自行设计的字符，储存到打印机内，以供打印。这项功能便叫做「用户自定义字符」(Down - Load Character)。

### 用户自定义字符功能原理

打印机所打印出的字型都是由“点”图案所构成的，其中内置的标准字库资料是储存在 ROM (只读存储器) 内，而用户自行定义的字型则是储存在 RAM (随机存储器) 中。如此，当打印机接到电脑传送过来的字符代码时，便会从 RAM 或 ROM 中读取字型资料，来打印出该字符。由于打印机在英文模式和中文模式下的用户自定义字符处理方式不同。现分别作详细介绍如下：

### 英文用户自定义字符的设计及使用

#### 一 功能原理

在英文模式下，须先将 ROM 中的字符完全复制到 RAM 中；再选择一些字符，以自行定义的字符取代之。然后将打印机切换至用户自定义字符区，即指定 RAM 中的字符集为作用的字符集，便可打印出这些字符。

#### 二 字型规格

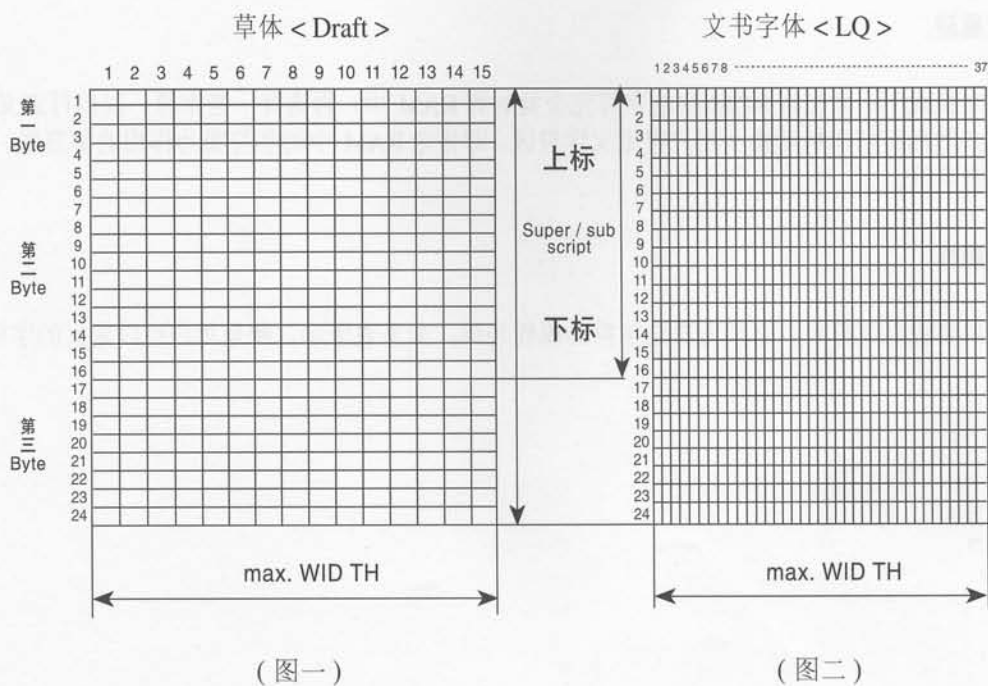
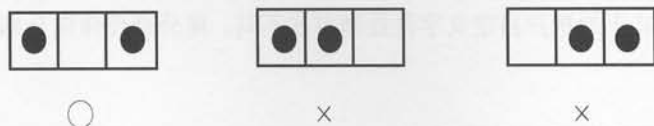
当用户要定义字符时，必须先决定字符的属性为何。如下表所示，便是用户可以定义的字符集。

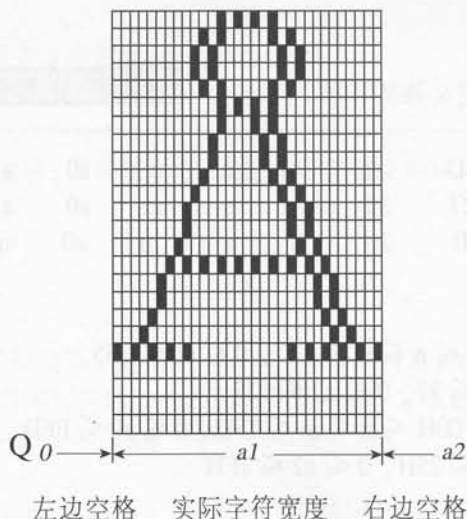
## 标准英文字型规格一览表：

	字符属性	矩阵规格	字符规格	字符宽度 (点)		点间距	每列字节数
				a1	a0+a1+a2		
标准字	草体	图一	9*24	9	12	1/120 英寸	3
	文书字体	图二	29*24	29	36	1/360 英寸	3
	比例字体	图二	37*24	37	42	1/360 英寸	3
上标 / 下标	草体	2/3(图一)	7*16	7	12	1/120 英寸	2
	文书字体	2/3(图二)	23*16	23	36	1/360 英寸	2
	比例字体	2/3(图二)	23*16	23	42	1/360 英寸	2

注 1：上表中字符间距的意义及关系，请参考图三。

注 2：在设计文书字体或比例字体时，(点间距为 1/360 英寸)，邻接点无法打印出，即每一个被打印出的点，它的左右位置都必须是空白，因此在设计字型时，下图第一种情况是正确的，而其余两种情况则是不允许的。





(图三)

### 三 范围及字符数

用户自定义字符的范围：0 ~ 127（即 ASCII 00H 至 7FH）

用户自定义字符数：128 个

### 四 载入程序

1. 按字型规格表，设计字型，并将其转换成数值。（转换的方法，请参阅第八章「图形资料解释」一节）
2. 利用 ESC : 指令，将内置于 ROM 中的所有字符组，拷贝到 RAM。
3. 选择定义用户自定义字符的代码。
4. 利用 ESC & 指令，载入字型至打印机的存储器（RAM）。

### 五 打印自定义字符

1. 利用 ESC % 1 指令切换至自定义字符区。
2. 设定打印机的打印模式。例如欲打印上/下标字时，需先利用 ESC S 指令，来改变打印模式。其它的字型，依此类推。
3. 送出用户自定义字符的代码。
4. 再利用 ESC % 0 指令切换回原先的字符集。

## 指令序列

ESC &	用户自定义英文字符				ESC/P	ESC/P2	ESC/PK	ESC/PK2		
格式:	ASCII 码:	ESC	&	s	n	m	a0	a1	a2	数据
	十进制码:	27	38	s	n	m	a0	a1	a2	数据
	十六进制码:	1B	26	s	n	m	a0	a1	a2	数据

### 参数值:

十进制数:  $S=0, 0 \leq n \leq m \leq 127, 0 \leq a0 \leq 255,$   
 $0 \leq a1 \leq 37, 0 \leq a2 \leq 127$

十六进制数:  $S=0H, 00H \leq n \leq m \leq 7FH, 0 \leq a0 \leq FFH,$   
 $0 \leq a1 \leq 25H, 0 \leq a2 \leq 7FH$

### 注释:

- n 是用户所要定义的第一个字符的 ASCII 码;当然,也可以使用字符本身来取代 ASCII 码。
- m 是用户所要定义的最后—个字符的 ASCII 码,如果用户只定义一个字符的话,n 和 m 便是相同的。
- a0、a1 和 a2 这三个字节值则规定了字符的宽度。其中,a0 所规定的是字符左边的空格数;a1 所规定的是可打印字符的宽度,而 a2 所规定的则是字符右边的空格数。
- 至于"数据"指的便是各个要被定义的字符资料。事实上,数据的正确数目,则由要被定义的字符型式所决定。如果用户所要定义的字符超过一个的话,那么,便连续的把所有的数据送给打印机。

ESC :	将 ROM 的字符拷贝到 RAM				ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式:	ASCII 码:	ESC	:	0	0	0		
	十进制码:	27	58	0	0	0		
	十六进制码:	1B	3A	00	00	00		

### 注释:

此一指令可把 ROM 中的字符集拷贝至用户自定义字符的区域,以便重新定义某些字符。

ESC %	开启/关闭用户自定义字符			ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式:	ASCII 码:	ESC	%	n			
	十进制码:	27	37	n			
	十六进制码:	1B	25	n			

注释：

n = 1 时，选择用户自定义字符集；

n = 0 时，选择内置字符集。

当然，在使用此指令之前，要先使用 ESC & 指令来定义用户自定义字符集。

## 程式范例

如下所示的 C 例子，简单的说明了定义用户自定义字符的过程。在这一例子中，定义了一个  $\Sigma$  字符来取代大写的 A 字母，并打印出其他一些大写的英文字母。定义这一个  $\Sigma$  字符是由 36 列  $\times$  3 个字节的的数据所组成的，因此，总共用了 108 个字节的的数据资料。

```
# include "print9.h"

main ()
{

unsigned char para [] [6] = { { 0x40,0x40,4,25,7}, {5} };
unsigned char data [] [75] = {
    { /* 英文用户自定义字符资料 */
        0x20,0x00,0x10,0x10,0x00,0x20,0x28,
        0x00,0x50,0x14,0x00,0xA0,0x2A,0x01,
        0x50,0x15,0x02,0xA0,0x2A,0x85,0x50,
        0x05,0x4A,0x80,0x22,0xB5,0x10,0x01,
        0x4A,0x00,0x20,0xB4,0x10,0x00,0x48,
        0x00,0x20,0x30,0x10,0x00,0x00,0x00,
        0x20,0x00,0x10,0x00,0x00,0x00,0x20,
        0x00,0x10,0x00,0x00,0x00,0x20,0x00,
        0x10,0x00,0x00,0x00,0x2A,0x01,0x50,
        0x14,0x00,0xA0,0x28,0x00,0x50,0x10,
        0x00,0x20,0x20,0x00,0x10 },
    "@ABC\n\n"
};

PrintOut (LQPrintData [0], LQPrintData [1] [0]);
PrintOut (LQPrintData [6], LQPrintData [7] [0]);
PrintOut (LQPrintData [4], LQPrintData [5] [0]);
PrintOut (LQPrintData [8], LQPrintData [9] [0]); /* 拷贝 ROM 至 RAM 自定义字符区 */
PrintOut (LQPrintData [10], LQPrintData [11] [0]);
PrintOut (para [0], para [1] [0]);
PrintOut (data [0], 75); /* 发送数据资料 */
PrintOut (LQPrintData [14], LQPrintData [15] [0]);
PrintOut (LQPrintData [16], LQPrintData [17] [0]); /* 四倍角放大 */
```

```

PrintOut (LQPrintData [18] ,LQPrintData [19] [0]);
PrintOut (data [1] ,0);
PrintOut (LQPrintData [20] ,LQPrintData [21] [0]); /* 打印用户自定义字符 */
PrintOut (data [1] ,0);
}

```

打印结果

```

@A B C
Σ A B C

```

## 中文用户自定义字符的设计及使用

### 一 功能原理

当用户要自行定义中文字符时，直接设定该字符在 RAM 中的地址代码（2 bytes），然后将字型资料载入至打印机存储器（RAM）内。以后用户只要由电脑送出该字符的地址代码给打印机，即可打印出此字。

### 二 中文字型规格一览表

字符属性	字符规格	字符宽度	点间距	每列字节数
全角字	24*24	24 点	1/180 英寸	3
半角字	11*24	11 点	1/180 英寸	3
1/4 角字	12*1	12 点	1/180 英寸	3

注：在定义半角字和 1/4 角字时，字型矩阵中未定义区域的值为 00H。

### 三 范围及字符数

用户自定义字符的范围：

中文字码	用户自定义字符的范围
国际标准区位码	F8A1 ~ F8FE

用户自定义字符数：94个

#### 四 载入程序

1. 按字型规格表,设计字型,并将其转换成数值。(转换的方法,请参考第八章「图形资料的解释」一节)
2. 选择用户自定义字符的代码。
3. 载入字型至打印机的存储器 (RAM) 。

#### 五 打印用户自定义字符

1. 送出用户自定义字符的代码,即可打印出此字符。
2. 要想打印半角字符或 1/4 角字符时,除了在字型设计时,需依照对应的字型规格设计外,在送出该字符代码前,必须先改变打印机的打印模式。(半角字符用 FS SI, 1/4 角字符用 FS r 来设定)

FS 2

用户自定义中文字符

ESC/PK

ESC/PK2

格式： ASCII 码： FS 2 a1 a2 ... 数据 ...  
 十进制码： 28 50 a1 a2 ... 数据 ...  
 十六进制码： 1C 32 a1 a2 ... 数据 ...

注释：

a1 及 a2 为用户自定义字符的地址。

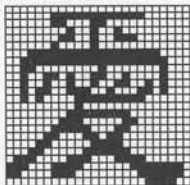
- a1 是用户自定义字符地址的第一个字节。其值为 F8H (十进制数为 248)。
- a2 是用户自定义字符地址的第二个字节。其值为 A1H 至 FEH (十进制数为 161 至 254 之间)。
- 至于"数据"部份,则是用户自定义的中文字符;由于用户自定义的中文字符是由垂直 24 点乘以水平的 24 列所组成,因此,定义一个中文字符便需要 3 × 24 个字节。

#### 程式范例及说明

用户自定义隶书字型 (爱), (普), (生) 三个字符。

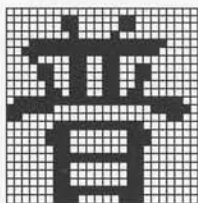
用户自定义字符设计图样：

第一个字 (对应数值资料如范例程序 450 行~ 520 行)

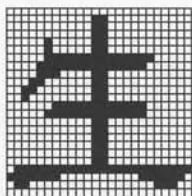




第二个字（对应数值资料如范例程序 540 行 ~ 610 行）



第三个字（对应数值资料如范例程序 630 行 ~ 700 行）



以上用户自定义的三个中文字符，程序说明如下：

```
#include "print9.h"

main ()
{

unsigned char para [] [2] = { {0xa1}, {1},
                               {0xa2}, {1},
                               {0xa3}, {1}
                               };
unsigned char data [] [72] = {
{ /* 定义“爱”字 */
  0x00,0x00,0x06,0x00,0x00,0x0E,0x01,
  0xF8,0x0C,0x03,0x80,0x18,0x03,0x18,
  0x30,0x03,0x70,0x60,0x67,0x40,0xC6,
  0x6B,0x01,0x86,0x5B,0x73,0x8E,0x43,
  0x1F,0x8C,0x43,0x2A,0xD8,0x7F,0x6A,
  0x70,0x63,0xCA,0x30,0x43,0x2A,0x70,
  0x5B,0x2A,0xF8,0x4B,0x6B,0x98,0x4B,
  0x09,0x0C,0x43,0x18,0x0E,0x03,0x30,
  0x07,0x03,0x00,0x07,0x01,0xF0,0x07,
  0x00,0x00,0x07,0x00,0x00,0x06,0x00,
  0x00,0x00 },
{ /* 定义“普”字 */
  0x00,0x0C,0x00,0x00,0x1C,0x00,0x04,
  0x18,0x00,0x0E,0x18,0x00,0x0D,0x98,
  0x00,0x0D,0xF9,0x80,0x0C,0x33,0xFF,
```

```

0x6C,0x13,0xFF,0x7F,0xF2,0x23,0x0F,
0xF2,0x22,0x0C,0x12,0x22,0x0C,0x12,
0x22,0x0F,0xF2,0x22,0x1F,0xF2,0x22,
0x6C,0x12,0x23,0x6C,0x33,0xFF,0x0D,
0xF3,0xFF,0x0D,0x91,0x80,0x0C,0x18,
0x00,0x08,0x18,0x00,0x00,0x1C,0x00,
0x00,0x1C,0x00,0x00,0x08,0x00,0x00,
0x00,0x00 } ,
{ /* 定义“生”字 */
0x00,0x00,0x04,0x00,0x00,0x0E,0x00,
0x30,0x0E,0x00,0x60,0x0C,0x01,0xC0,
0x0C,0x0F,0x84,0x0C,0x03,0x8C,0x0C,
0x03,0x0C,0x08,0x03,0x0C,0x08,0x03,
0x0C,0x08,0x03,0x0C,0x08,0x7F,0xFF,
0xF8,0x7F,0xFF,0xF8,0x03,0x0C,0x08,
0x03,0x0C,0x08,0x03,0x0C,0x08,0x03,
0x0C,0x08,0x03,0x0C,0x0C,0x02,0x00,
0x0C,0x00,0x00,0x0E,0x00,0x00,0x0E,
0x00,0x00,0x0E,0x00,0x00,0x0C,0x00,
0x00,0x00 } ,
{ 0xf8,0xa1,0xf8,0xa2,0xf8,0xa3 }
};

```

```

PrintOut (LQPrintData [0] , LQPrintData [1] [0]);
PrintOut (LQPrintData [2] , LQPrintData [3] [0]); /* 设定汉字模式 */
PrintOut (LQPrintData [22] , LQPrintData [23] [0]);
PrintOut (para [0] , para [1] [0]); /* 用户自定义：“爱”字 */
PrintOut (data [0] , 72); /* 中文字符字型 */

```

```

PrintOut (LQPrintData [22] , LQPrintData [23] [0]);
PrintOut (para [2] , para [3] [0]); /* 用户自定义：“普”字 */
PrintOut (data [1] , 72); /* 中文字符字型 */

```

```

PrintOut (LQPrintData [22] , LQPrintData [23] [0]);
PrintOut (para [4] , para [5] [0]); /* 用户自定义：“生”字 */
PrintOut (data [2] , 72); /* 中文字符字型 */

```

```

PrintOut (data [3] , 6);
PrintOut (LQPrintData [14] , LQPrintData [15] [0]);
PrintOut ("\n",0) ;
PrintOut (data [3] , 6);
PrintOut (LQPrintData [24] , LQPrintData [25] [0]);
PrintOut (LQPrintData [12] , LQPrintData [13] [0]);
PrintOut ("\n",0) ;
PrintOut (data [3] , 6);
PrintOut (LQPrintData [24] , LQPrintData [25] [0]);

```

```

PrintOut ("\n",0) ;
PrintOut (data [3] , 6) ;
PrintOut (LQPrintData [26] , LQPrintData [27] [0]) ;
PrintOut ("\n",0) ;
}

```

打印结果

### 用户自定义字符注意事项

- 用户自定义字符是储存在打印机的存储器中，因此，当打印机的电源被关闭后，或者被 INIT 讯号重新启动后，这些字符资料便会遗失。如果用户是利用程序来定义字符的话，一定要记得储存程序，以便稍后使用。
- 在设计上标字和下标字时，可采用相同的字型设计，然后利用选择上标、下标模式的指令，来调整打印的位置。
- 用户自定义中文字符可以通过各种特殊打印的方式来表现，例如：粗体、斜体、中空、阴影、立体 … 等等的打印效果。
- 在英文模式下，用户基本上可以定义 128 个字符，并赋予代码 00H 至 7FH（十进制数为 0 至 127），然而代码 0 至 32 和 127 已被定义为控制码，无法打印出。因此，实际上只有 95 个用户自定义字符可使用。如果用户想拥有 128 个用户自定义字符，便需利用 ESC t 指令。（详情请参考第十三章中 ESC t 的指令说明。）
- 用户所要定义的字符一定要在这些字符能被打印出的模式下定义；同样的，在某一模式下定义的字符也不能在别的模式下打印出。若用户想要打印出用户自定义的字符时，便一定要把打印机的打印模式切换到原先定义这些字符的模式。

```

/*      print9.h      */

#include "stdio.h"

/*****
/* 变量定义      */
*****/

```

```

#define ESC      0x1B /* 模式设定 */
#define FS      0x1C /* 汉字模式设定 */

void PrintOut () ;
/*****
/* 数据结构体设定 */
*****/
static unsigned char LQPrintData [] [6] = {
    {ESC,'@'}, {2}, /* 打印机初始化 */
    {FS,'&'}, {2}, /* 设定汉字模式 */
    {FS,'!'}, {2}, /* 解除汉字模式 */
    {ESC,'x',1}, {3}, /* 选择信函打印方式 */
    {ESC,':',0,0,0}, {5}, /* ROM 字符拷贝至 RAM */
    {ESC,'&',0}, {3}, /* 用户自定义英文字符 */
    {ESC,'W',0}, {3}, /* 解除倍宽打印 */
    {ESC,'W',1}, {3}, /* 设定倍宽打印 */
    {ESC,'w',1}, {3}, /* 设定倍高打印 */
    {ESC,'% ',0}, {3}, /* 选择内置字符集 */
    {ESC,'% ',1}, {3}, /* 选择用户自定义字符集 */
    {FS,'2',0xf8}, {3}, /* 用户自定义汉字 */
    {FS,'W',1}, {3}, /* 设定四倍角打印 */
    {FS,'W',0}, {3} /* 解除四倍角打印 */
};
/*****

void PrintOut (unsigned char * lpcbData,int Length) /* 打印输出 */
{
if (! Length) fprintf (stdprn,"%s",lpcbData) ;
else do {
    fprintf (stdprn,"%c",*lpcbData++) ;
    Length-- ;
} while (Length) ;
}

```

# 第十章

## 使用自动送纸器

单页纸自动送纸器的功能在于能连续的在每页打印完之后自动供纸,使单页纸的处理更为快速方便。

在使用自动送纸器之前,必须先将打印机设定在送纸器模式下,并设定好页面长度,这样送纸器就能在每页打印完后,顺利地自动装纸。

当您在送纸器模式下执行自我测试时,打印机会计算每一页所打印的行数,并且在最后一行打印出这一页的总行数,而打印机也以这一行做为页面长度的标准。至于其他纸张页面长度的设定,您可以利用自我测试时打印出的总行数,然后再由软件指令来设定页长。

### 指令格式

ESC EM	设定自动送纸装置			ESC/P	ESC/P2	ESC/PK	ESC/PK2
格式:	ASCII 码:	ESC	EM	n			
	十进制码:	27	25	n			
	十六进制码:	1B	19	n			

注释:

n 的值有下列几种:

ASCII	十进制	十六进制	功能
0	48	30	关闭自动送纸模式
1	49	31	选择 1 号槽送纸
2	50	32	选择 2 号槽送纸
4	52	34	开启自动送纸模式
R	82	52	退出纸张不再装入

除非用户的打印机装有单页纸的送纸器;否则,不要使用此一指令;另外,若 n 的值不是规定的数值时,则此一指令被忽略。用这个指令开启自动送纸功能后,就不必再管 DIP 切换开关或由面板的「选择模式」的设定了。

当单页纸送纸器的模式被开启后,打印机接到换页的指令后,便会自动的装纸;而且,如果最后的打印位置超出打印区的话,换行的指令、垂直的定位指令,以及,ESC J 指令也会迫使打印机自动地装纸。因此,在这一种模式下,资料便可以连续的送到打印机,仿佛所使用的是连续打印纸。

## 对 齐

对齐的功能是指文字打印时采取左边界或右边界对齐的格式,在软件指令的控制下,对齐的格式包括了下列四种:

(1) 左边界对齐 (初始设定值)

┌ This is a sample of  
├ left just.  
└

左边界                  右边界

(2) 中间对齐

┌ This is a sample of  
├ centering  
└

左边界                  右边界

(3) 右边界对齐

┌ This is a sample of  
├ right just.  
└

左边界                  右边界

(4) 左右对齐

┌ This is a sample of  
├ full justification.  
└

左边界                  右边界

ESC a

设定对齐状态

ESC/P

ESC/PK

格式:	ASCII 码:	ESC	a	n
	十进制码:	27	97	n
	十六进制码:	1B	61	n

注释:

此一指令可根据下列的关系,来选择对齐的位置。

n=0 时,选择左边对齐 (初始设定的情况)

n=1 时,选择中间对齐

n=2 时,选择右边对齐

n=3 时,选择左右边界 (full) 对齐

n=0 是初始设定的情况;而左右对齐 (n=3) 则在缓冲区已"满"时,才执行;另外,除了在 n=0 的模式外,水平定位 (HT) 和退格 (BS) 都是无效的;而在 n=3 的模式下,文字段中不能有回车字符。

左右对齐注意事项

- 在左右对齐的格式下,设定点位置的指令 (如 ESC \$, ESC \, 和 HT) 在每一行开始时下达才有效。

- 左右对齐指令在每行资料输入回车 (CR)、换行 (LF) 指令后才执行,但打印资料必须在打印区的 75% 到 125% 之间。
- 当输入一行资料不足打印区的 75% 时,则无法执行左右对齐。
- 当输入一行资料超过打印区的 25% 时,则左右对齐指令将会把超过打印区最大打印量的资料,在下一行打印出。

## 程式范例

```
#include "print11.h"
main ()
{
    unsigned char para [] [2] = {{3}, {1},
                                {20}, {1},
                                {136}, {1},
                                {0}, {1}
                                };
    unsigned char *data [] = {
        "123456789*123456789*123456789*1234567890\n",
        "  中间对齐\n",
        "                右边对齐\n",
        "左右边界对齐\n",
        "  左边对齐\n"
    };
    PrintOut (LQPrintData [0], LQPrintData [1] [0]);
    PrintOut (LQPrintData [2], LQPrintData [3] [0]);
    PrintOut (data [0], 0);
    PrintOut (LQPrintData [4], LQPrintData [5] [0]);
    PrintOut (para [0], para [1] [0]);          /* 左边界第 3 列 */
    PrintOut (LQPrintData [6], LQPrintData [7] [0]);
    PrintOut (para [2], para [3] [0]);          /* 右边界第 20 列 */
    PrintOut (LQPrintData [10], LQPrintData [11] [0]); /* 中间对齐 */
    PrintOut (data [1], 0);
    PrintOut (LQPrintData [12], LQPrintData [13] [0]); /* 右边对齐 */
    PrintOut (data [2], 0);
    PrintOut (LQPrintData [14], LQPrintData [15] [0]); /* 左右边对齐 */
    PrintOut (data [3], 0);
    PrintOut (LQPrintData [8], LQPrintData [9] [0]); /* 左边对齐 */
    PrintOut (data [4], 0);
    PrintOut (LQPrintData [6], LQPrintData [7] [0]); /* 返回初始设定 */
    PrintOut (para [4], para [5] [0]);
    PrintOut (LQPrintData [4], LQPrintData [5] [0]);
}
```

```
PrintOut (para [6] ,para [7] [0]);
}
```

打印结果

```
123456789*123456789*123456789*1234567890
```

中间对齐

右边对齐

左右边界对齐

左边对齐

```
/*      print11.h      */

#include "stdio.h"

/*****
/* 变量定义      */
*****/
#define ESC      0x1B /* 模式设定 */
#define FS      0x1C /* 汉字模式设定 */
void PrintOut () ;
/*****
/* 数据结构体设定      */
*****/
static unsigned char LQPrintData [] [4] = {
    {ESC,'@'}, {2}, /* 打印机初始化 */
    {FS,'&'}, {2}, /* 设定汉字模式 */
    {ESC,'I'}, {2}, /* 设定左边界 */
    {ESC,'Q'}, {2}, /* 设定右边界 */
    {ESC,'a',0}, {3}, /* 选择左对齐 */
    {ESC,'a',1}, {3}, /* 选择中间对齐 */
    {ESC,'a',2}, {3}, /* 选择右对齐 */
    {ESC,'a',3}, {3} /* 选择左右边界对齐 */
};
/*****
void PrintOut (unsigned char * lpcbData,int Length) /* 打印输出 */
{
if (! Length) fprintf (stdprn,"%s",lpcbData) ;
else do {
fprintf (stdprn,"%c",*lpcbData++) ;
        Length-- ;
    } while (Length) ;
}
```





十进制码 : 19  
十六进制码 : 13

注释:

在接到控制码 DC1 以前,让打印机处于脱机的状态。必须注意的是,如果打印机是因为 DC3 指令而造成脱机的话,便不能使用 ON - LINE 键来使打印机恢复联机。

## 删除资料

当资料在待打印缓冲区中时,可用软件指令清除缓冲区,或删除缓冲区中的最后一个字符。

CAN                      消除行缓冲区                      **ESC/P**   **ESC/P2**   **ESC/PK**   **ESC/PK2**

---

格式:    ASCII 码 :        CAN  
          十进制码 :        24  
          十六进制码 :      18

注释:

消除此指令前的该行缓冲区中所有文字;但不影响控制码。

DEL                      删除字符                      **ESC/P**   **ESC/P2**   **ESC/PK**   **ESC/PK2**

---

格式:    ASCII 码 :        DEL  
          十进制码 :        127  
          十六进制码 :      7F

注释:

除非该字符已被打印出;否则,便删除缓冲区中的前一个文字字符;而对于控制码,则没有影响。此指令在绘图指令及定位指令(如 HT, ESC \$, ESC \) 后无效。

BS                        退格                              **ESC/P**   **ESC/P2**   **ESC/PK**   **ESC/PK2**

---

格式:    ASCII 码 :        BS  
          十进制码 :        8  
          十六进制码 :      08

注释:

打印出储存在缓冲区中的资料,同时,打印位置也会向左退一格。但退格的左限不会超过左边界的设定。如果送出 ESC a, ESC 2 或 ESC 3 指令的话,此一指令会被忽略。必须注意的是如果最后一个打印出的资料是在图象模式下的话,那么,打印头便会移回图象开始打印之处。

## 打印处理

### 打印头归位

ESC <

打印头归位

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式： ASCII 码： ESC <  
十进制码： 27 60  
十六进制码： 1B 3C

注释：

不管左边界的设定值为何,打印头都会回到原位 (home 的位置),并由左向右打印;要注意的是,这一个指令可为回车字符所取消。

### 打印方向

在正常的情况下,打印机是双向打印。但如果在打印文字或图形时,需要更准确的定位的话,便可利用此一指令来选择单向的打印模式。

ESC U

设定单 / 双向打印

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式： ASCII 码： ESC U n  
十进制码： 27 85 n  
十六进制码： 1B 55 n

注释：

n = 1 时,设定单向打印模式;  
n = 0 时,设定双向打印模式。

### 打印速度

ESC s

设定 / 解除半速打印

ESC/P

格式： ASCII 码： ESC s n  
十进制码： 27 115 n  
十六进制码： 1B 73 n

注释：

n = 1 时,选择半速打印模式。  
n = 0 时,选择全速打印模式。

FS x

设定 / 解除高速打印

ESC/PK

ESC/PK2

格式:	ASCII 码:	FS	x	n
	十进制码:	28	120	n
	十六进制码:	1C	78	n

注释:

设定 / 解除高速的打印模式。

如  $n = 1$ , 设定高速打印模式,

如  $n = 0$ , 解除高速打印模式。

## 颜色选择

在 EPSON 打印机中的 LQ-2500, DLG-2000K 及 LQ-2550 等具有彩色功能的机型上, 可选择打印的颜色, 其选择控制的指令如下:

ESC r

选择打印颜色

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	ESC	r	n
	十进制码:	27	114	n
	十六进制码:	1B	72	n

注释:

不具有彩色功能的机型, 该指令无效。

此一指令可根据下列的关系, 来选择打印的颜色:

n	颜色	n	颜色
0	黑	4	黄
1	红	5	橙
2	蓝	6	绿
3	紫		

## 发声装置

所有 EPSON 打印机都具有发声的能力, 而且由用户自行控制。

BEL

蜂鸣器鸣叫

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式:	ASCII 码:	BEL
	十进制码:	7

十六进制码： 07

注释：使打印机的发声装置发出声音。

## 打印位置

打印机有两种水平移动的方式，是用来控制打印头的移动。

## 绝对位置

ESC \$ 设定绝对水平打印位置

ESC/P ESC/P2 ESC/PK ESC/PK2

格式：	ASCII 码：	ESC	\$	n1	n2
	十进制码：	27	36	n1	n2
	十六进制码：	1B	24	n1	n2

注释：

不管目前的字符宽度为何，打印头都会移至指定的绝对位置。

其中， $n = (n1 + 256 * n2) * (\text{单位})$

也就是左边界右方第  $n$  个点的位置。对于使用 ESC/P 和 ESC/P-K 控制命令的打印机，各点间距单位为 1/60 英寸。而对于使用 ESC/P2 和 ESC/P-K2 控制命令的打印机，各点的间距单位为 1/60 英寸或由 ESC (U 命令定义。

其中  $n1$  的值必须介于 0 至 255 之间，而  $n2$  的值则必须介于 0 至 3 之间。至于用户所能规定的最大点位置则为 816。当然，若该指令使打印头移出右边界的话，则此指令被忽略。此绝对位置的例图如下：



ESC ( V 设定绝对垂直打印位置

ESC/P2 ESC/PK2

格式：	ASCII 码：	ESC	(	V	m1	m2	n1	n2
	十进制码：	27	40	86	m1	m2	n1	n2
	十六进制码：	1B	28	56	m1	m2	n1	n2

注释：

将打印头垂直移动到由以下公式指定的绝对位置。

其中， $n = (n1 + n2 * 256) * (\text{单位})$

也就是上边界下方第  $n$  个点的位置。对于使用 ESC/P 和 ESC/P-K 控制命令的打印机，该命令无效。而对于使用 ESC/P2 和 ESC/P-K2 控制命令的打印机，各点的间距单位为 1/60 英寸或由 ESC (U 命令定义)。

其中  $n1$ 、 $n2$  的值必须介于 0 至 255 之间。

这一个指令能够立即走纸，但不影响随后的行间距离。

此指令若遇页缝空白时，则自行移动至下页顶端开始打印。

其它参数： $m1 = 2$ ， $m2 = 0$ 。

## 相对位置

ESC \                      设定相对水平打印位置

ESC/P

ESC/P2

ESC/PK

ESC/PK2

格式：    ASCII 码：    ESC    \    n1    n2  
          十进制码：        27    92    n1    n2  
          十六进制码：    1B    5C    n1    n2

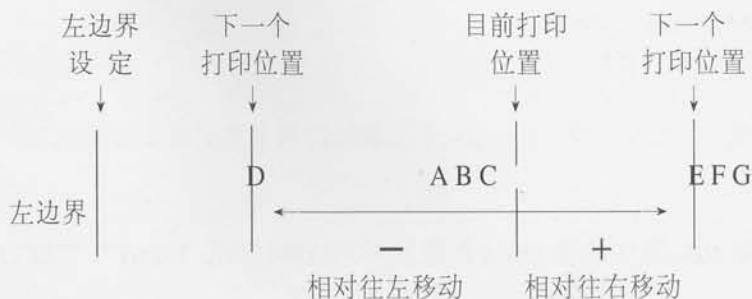
注释：

该指令将打印头水平方向移到距当前位置  $n$  个点距离的位置。

其中， $n = (256 * n2 + n1) * (\text{单位})$

也就是在距当前位置  $n$  个点的位置。和绝对移动指令 ESC \$ 不同的是，这一个指令的移动单位由目前的字符大小，以及所选用的打印模式（文书体或草体）决定。其中，在文书体模式下，对于使用 ESC/P 和 ESC/P-K 控制命令的打印机，移动的单位是 1/180 英寸；而对于使用 ESC/P2 和 ESC/P-K2 控制命令的打印机，各点的间距单位为 1/180 英寸或由 ESC (U 命令设定)。而在草体模式下，对于使用 ESC/P 和 ESC/P-K 控制命令的打印机，移动的单位是 1/120 英寸；而对于使用 ESC/P2 和 ESC/P-K2 控制命令的打印机，各点的间距单位为 1/120 英寸或由 ESC (U 命令设定)。

其中  $n1$ 、 $n2$  的值必须介于 0 至 255 之间。如果设定的相对位置超出当前边界的设定时，此指令会被忽略。





```

#include "print12.h"

main ()
{
    unsigned char para [][3] = {{4}, {1},
                                {100,0}, {2},
                                {50,0}, {2},
                                {150,0}, {2}
                                };
    unsigned char *data [] = {
        "1234567890123456789012345678901234567890\n",
        "First",
        "Second",
        "Third\n"    };
    PrintOut (LQPrintData [0] , LQPrintData [1] [0]);
    PrintOut (LQPrintData [4] , LQPrintData [5] [0]);
    PrintOut (data [0] ,0);
    PrintOut (LQPrintData [6] , LQPrintData [7] [0]);
    PrintOut (para [0] ,para [1] [0]);
    PrintOut (LQPrintData [8] , LQPrintData [9] [0]);
    Printout (para [2] ,para [3] [0]);          /* 100 点定位 */
    PrintOut (data [1] ,0);
    PrintOut (LQPrintData [8] , LQPrintData [9] [0]);
    Printout (para [4] ,para [5] [0]);          /* 50 点定位 */
    PrintOut (data [2] ,0);
    PrintOut (LQPrintData [8] , LQPrintData [9] [0]);
    Printout (para [6] ,para [7] [0]);          /* 150 点定位 */
    PrintOut (data [3] ,0);
}

```

打印结果

```

1234567890123456789012345678901234567890
      Second   First   Third

```

## 程式范例二

例二是设定绝对位置在第 120/60 英寸点位置上, 和设定相对位置在 + 60/180 英寸及 - 300/180 英寸。



```

#include "print12.h"

main ()
{
    unsigned char para [][3] = { {4}, {1},
        {120,0}, {2},
        {60,0}, {2},
        {0}, {1}
    };
    unsigned char *data [] = {
        "12345678901234567890123456789012345678901234567890\n",
        "绝对位置",
        "start n/60 inch\n\n",
        "相对位置",
        "+60/180 inch\n\n",
        "      ",
        "-300/180 inch"
    };
    unsigned char n1,n2 ;
    PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
    PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
    PrintOut (data [0] ,0) ;
    PrintOut (LQPrintData [6] ,LQPrintData [7] [0]);
    PrintOut (para [0] ,para [1] [0]);
    PrintOut (data [1] ,0) ;
    PrintOut (LQPrintData [8] ,LQPrintData [9] [0]);
    PrintOut (para [2] ,para [3] [0]);          /* 设定绝对位置 */
    PrintOut (data [2] ,0) ;
    PrintOut (data [3] ,0) ;
    PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
    PrintOut (para [4] ,para [5] [0]);          /* 设定相对位置 */
    PrintOut (data [4] ,0) ;                    /* 相对右移 60/180 英寸 */
    PrintOut (data [5] ,0) ;
    PrintOut (data [3] ,0) ;
    n2 = (65536-300) /256 ;
    n1 = (65535-300) -n2*256 ;                 /* 相对左移 300/180 英寸 */
    PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
    PrintOut (&n1,1) ; PrintOut (&n2,1) ;
    PrintOut (data [6] ,0) ;
    PrintOut ("\n",0) ;
    PrintOut (LQPrintData [6] ,LQPrintData [7] [0]);
    PrintOut (para [6] ,para [7] [0]);
}

```

## 打印结果

12345678901234567890123456789012345678901234567890

绝对位置

Start n/60 inch

相对位置 +60/180 inch

-300/180inch 相对位置

## 设定英文水平移动距离

用本命令设定 ASCII 字符的水平移动距离。

ESC c 设定英文水平移动距离

ESC/P2

ESC/PK2

格式： ASCII 码： ESC c n1 n2  
十进制码： 27 99 n1 n2  
十六进制码： 1B 63 n1 n2

注释：

该指令根据下列公式，设定 ASCII 字符的水平移动距离。

水平移动距离 (HMI) = (n1 + 256 \* n2) / 360 英寸。

缺省值为 1/10 英寸

参数范围如下：

$0 < n2 < 4$   
 $0 < n1 < 255$   
 $0 < HMI < 3.00$

## 设定汉字水平移动距离

用本命令设定汉字的水平移动距离 (HMI)。

FS c 设定汉字水平移动距离

ESC/PK2

格式： ASCII 码： FS c n1 n2  
十进制码： 28 99 n1 n2  
十六进制码： 1C 63 n1 n2

注释：





```

#define FS      0x1C /* 汉字模式设定 */

void PrintOut () ;
/*****
/* 数据结构体设定 */
/*****
static unsigned char LQPrintData [] [3] = {
    {ESC,'@'}, {2}, /* 打印机初始化 */
    {FS,'&'}, {2}, /* 设定汉字模式 */
    {FS,'.'}, {2}, /* 解除汉字模式 */
    {ESC,'l'}, {2}, /* 设定左边界 */
    {ESC,'$'}, {2}, /* 设定绝对水平打印位置 */
    {ESC,92}, {2} /* 设定相对水平打印位置 */
};

/*****
void PrintOut (unsigned char * lpcbData,int Length) /* 打印输出 */
{
    if (!Length) fprintf (stdprn,"%s",lpcbData);
    else do {
        fprintf (stdprn,"%c",*lpcbData++);
        Length--;
    } while (Length);
}

```

## 字符表

- 选择字符表 ..... 13-1
- 分配字符表 ..... 13-3
- 以字符方式打印数据 ..... 13-4
- 国际字符集 ..... 13-4

在打印机的内置存储器 ROM 中,除了已装有字符组外,并且依 ASCII 标准码的编码,已定义了由 0 至 127,共 128 个字符,而在后半部的字符表 128 至 255 (ASCII 码 80H 至 FFH) 中,则是字符集的设定存放区域,所以当您要选择斜体字符、EPSON 扩充图形字符和用户自定义字符时,就必须下达正确的软件指令。

由于这些字符集的字符或符号,只影响后半部的字符表,所以您仍然可以将标准文字与这些扩充字符同时使用。

### 选择字符表

(1) 在斜体字符表中是由 80H ~ 9FH 的控制码和后半部 A0H ~ FEH 的斜体字符来组成的字符表。

00H	20H	80H	A0H
			斜体字符 20H—7EH
1FH	7FH	9FH	FFH

#### 斜体字

```
!"#$%&' () *+,-./0123456789 : ;<=>? @ABCDEFGHIJ
KLMNOPQRSTUVWXYZ [\] ^_`abcdefghijklmnopqrs
tuvwxyz { | } ~
```

(2) EPSON 扩充图形字符表是由 IBM 图形字符和国际性字符来组成后半部 (80H ~ FFH) 的字符表。其中 80H 至 9FH 仍是控制码的定义区。

00H	20H	80H	A0H
			EPSON 扩充图形字符
1FH	7FH	9FH	FFH



格式： ASCII 码： ESC 7  
 十进制码： 27 55  
 十六进制码： 1B 37

#### 注释：

这一指令可影响储存在 ASCII 代码 128 至 159 的字符，由于该区域可以被定义为用户自定义字符、图形字符或重复 ASCII 码 0 到 31 之间的控制码。因此，便必须加以选择，而此指令便是选择控制码。

#### 注意事项

- 当用户选择斜体字符表时，ASCII 码 128 至 159 即被视为控制码无法打印，意即 ESC 6 和 ESC 7 指令无效。
- 当用户在定义字符时，所用的代码仍然要介于 0 至 127 的范围内，然而，当用户要在 ESC t2 指令之后，使用这些字符的话，便必须再把每个代码加上 128。

## 分配字符表

将指定字符表的内容分配给另一字符表。

格式： ASCII： ESC ( t n1 n2 d1 d2 d3  
 十进制数： 27 40 116 n1 n2 d1 d2 d3  
 十六进制数： 1B 28 74 n1 n2 d1 d2 d3

#### 注释：

根据下列数值，将 d2 和 d3 指定的字符表分配给 d1 所表示的字符表。

d2 = 0 d3 = 0：斜体  
 d2 = 1 d3 = 0：PC437 (美国)  
 d2 = 2 d3 = 0：PC850 (多国语言)

d1 字符表是由 ESC t 命令所选定的三种字符表之一。

d1 = 0, 1, 2 或 "0", "1", "2"  
 n1 = 3, n2 = 0





格式:    ASCII :        ESC        R        n  
           十进数:        27        82        n  
           十六进数:       1B        52        n

## 注释:

所选择的国际字符为何,由 n 的值来决定,如下所示:

n	国家	n	国家
0	美国	7	西班牙 I
1	法国	8	日本
2	德国	9	挪威
3	英国	10	丹麦II
4	丹麦 I	11	西班牙II
5	瑞典	12	拉丁美洲
6	意大利		

## 程式范例一

```
#include "print13.h"

main ()
{
unsigned char i;
unsigned char para [] [15] = {{1,12,15,18,21,24,27,30,33,36,39,42,45,0}, {14}}
unsigned char data [] [38] = {
"美国",
"法国",
"德国",
"英国",
"丹麦 I ",
"瑞典",
"意大利",
"西班牙 I ",
"日本",
"挪威",
"丹麦 II",
"西班牙 II",
"拉丁美洲",
"&h:",
"23","24","40","5B","5C","5D","5E","60","7B","7C","7D","7E",
```

```
{HT, 0x23,HT,0x24,HT,0x40,HT,0x5b,HT,0x5c,HT,0x5d,HT,
0x5e,HT,0x60,HT,0x7b,HT,0x7c,HT,0x7d,HT,0x7e,HT }
};
```

```
PrintOut (LQPrintData [0] ,LQPrintData [1] [0]);
PrintOut (LQPrintData [4] ,LQPrintData [5] [0]);
PrintOut (LQPrintData [6] ,LQPrintData [7] [0]);
PrintOut (LQPrintData [8] ,LQPrintData [9] [0]);
PrintOut (LQPrintData [10] ,LQPrintData [11] [0]);
PrintOut (para [0] ,para [1] [0]); /* 水平定位点设定 */
PrintOut (data [13] ,0);
for (i=14 ;i<=25 ;i++)
{ PrintOut (&data [26] [0] ,1);
PrintOut (data [i] ,0);
}
PrintOut (&data [26] [0] ,1);
PrintOut ("\n",0) ;
for (i=0 ;i<=12 ;i++)
{
PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
PrintOut (&i,1) ; /* 打印 13 种国际字符 */
PrintOut (LQPrintData [2] ,LQPrintData [3] [0]);
PrintOut (data [i] ,0);
PrintOut (LQPrintData [4] ,LQPrintData [5] [0]);
PrintOut (data [26] ,25);
PrintOut ("\n",0) ;
}
PrintOut (LQPrintData [12] ,LQPrintData [13] [0]);
PrintOut (LQPrintData [14] ,LQPrintData [15] [0]);
i=0; PrintOut (&i,1) ; /* 返回美国字符 */
}
```

打印结果

&H :	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
#	\$	@	[	\	]	^	`	{		}	~	
#	\$	à	°	ç	§	^	`	é	ù	è	~	
#	\$	§	Ä	Ö	Ü	^	`	ä	ö	ü	β	
£	\$	@	[	\	]	^	`	{		}	~	
#	\$	@	Æ	Ø	Å	^	`	æ	ø	å	~	
#	⊠	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü	ì
#	\$	@	°	\	é	^	ü	ä	ö	è	ì	
Pt	\$	@	ı	Ñ	ı	^	`	ñ	ñ	}	~	
#	\$	@	[	¥	]	^	`	{		}	~	
#	⊠	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü	
#	\$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü	
#	\$	á	ı	Ñ	ı	é	`	í	ñ	ó	ú	
#	\$	á	ı	Ñ	ı	é	ü	í	ñ	ó	ú	

美国  
法国  
德国  
英国  
丹麦 I  
瑞典  
意大利  
西班牙 I  
日本  
挪威  
丹麦 II  
西班牙 II  
拉丁美洲

```
/*      print13.h      */

#include "stdio.h"

/*****
/* 变量定义      */
*****/

#define ESC      0x1B /* 模式设定 */
#define FS      0x1C /* 汉字模式设定 */
#define HT      0x09 /* 执行水平跳格 */

void PrintOut ( ) ;
/*****
/* 数据结构体设定      */
*****/
static unsigned char LQPrintData [] [100] = {
    {ESC,'@'}, {2}, /* 打印机初始化 */
    {FS,'&'}, {2}, /* 设定汉字打印模式 */
    {FS,'.'}, {2}, /* 解除汉字打印模式 */
    {ESC,'2'}, {2}, /* 设定 1/6 英寸换行量 */
    {ESC,'x',1}, {3}, /* 选择信函打印方式 */
    {ESC,'D'}, {2}, /* 设定水平跳格位置 */
    {ESC,'D',0}, {3}, /* 清除所有跳格 */
    {ESC,'R'}, {2} /* 选择国际字符集 */
};
/*****
void PrintOut (unsigned char * lpcbData,int Length) /* 打印输出 */
{
    if (! Length) fprintf (stdprn,"%s",lpcbData) ;
```

```
else do {  
    fprintf (stdprn,"%c",*lpcbData++) ;  
    Length-- ;  
} while (Length) ;  
}
```

## 光栅打印命令

- 选择图形模式 ..... 14-1
- 微调控制 ..... 14-1
- 打印光栅图形 ..... 14-2
- 控制点 (dot) 尺寸 ..... 14-3
- 选择光栅图象 (TIFF 压缩模式) ..... 14-4
- 选择光栅图象 (Delta row 压缩模式) ..... 14-5
- 图形数据传送 ..... 14-6
- 指定相对水平位置 ..... 14-8
- 指定相对垂直位置 ..... 14-9
- 指定颜色 ..... 14-10
- 解除 Seed row ..... 14-10
- 解除压缩模式 ..... 14-10
- 水平位置初始化 ..... 14-11
- 选择基准单位字节 ..... 14-11
- 选择基准单位点 ..... 14-11

### ESC ( G

### 选择图形模式

ESC/P2

ESC/PK2

代码]	ASCII	ESC	(	G	nL	nH	m
	16 进制	1B	28	47	nL	nH	m
	10 进制	27	40	71	nL	nH	m

[有效范围] nL=1  
nH=0  
m=1,49

[功能] 选择图形模式 (允许打印光栅图象)

#### [注意]

- (1) 发送 ESC @ 命令 (初始化打印机) 可以退出光栅图形模式。
- (2) 本命令清除所有用户定义的字符和 TAB 的设置。
- (3) 在图形模式下不能打印文本。
- (4) 在同一页上文本和图形不能混合打印。

### ESC ( i

### 微调 (MicroWeave) 控制

ESC/P2

[代码]	ASCII	ESC	(	i	l	0	n
	16 进制	1B	28	69	1	0	n
	10 进制	27	40	105	1	0	n

[有效范围] n=0: 解除微调整控制  
n=1: 选择微调整控制

[功能] 选择/解除微调整控制

[初值] 解除微调整控制

[注意]

- (1) 在本命令前要先执行“换页”动作。在换页后的 TOF 位置以外设定时, 不能保证以后的动作。
- (2) 本代码只在 360x360dpi 模式、720 × 720dpi 模式下有效。
- (3) 微调整方式下, 由 ESC \、ESC \$ 等命令进行水平位置指定时, 请指定以 8 点为单位。此外, 非 ESC.2 而在 ESC.3 中进行水平位置指定的场合下, 请指定以 8 点为单位。
- (4) 用 ESC. 命令传送光栅图象数据时, 请将 n 置为 1。
- (5) 用 ESC@、ESC (G 命令初始化时会解除微调整模式。
- (6) 本模式只在图象模式下有效。在图象模式以外的场合下指定时, 本模式无效。(只将代码舍弃)

ESC .

打印光栅图形

ESC/P2

ESC/PK2

[代码]

ASCII	ESC	.	c	v	h	m	nL	nH	d1	d2...dk
16 进制	1B	2E	c	v	h	m	nL	nH	d1	d1...dk
10 进制	27	46	c	v	h	m	nL	nH	d1	d2...dk

[有效范围]

c=0,1  
v=5,10,20  
h=5,10,20  
m=1,8,24  
 $0 \leq nL \leq 255$   
 $0 \leq nH \leq 127$   
 $0 \leq d \leq 255$

[功能]

- (1) 以光栅格式打印点阵图形。(行到行, 左到右)
- (2) 在打印光栅图形时, 可以对图形数据进行压缩; 计数器中包含一个数据, 该数据指定对某一特定字节的重复次数。
- (3) 参数描述如下:

c=0 全图形模式 (非压缩)  
1 压缩光栅图形模式 (RLE 模式)  
v: 垂直分辨率 (3600/v dpi)

h : 水平分辨率 (3600/h dpi)

v,h 只能取根据下表组合的值

v	h	v (dpi)	h (dpi)	m
20	20	180	180	1,8,24
20	10	180	360	1,8,24
10	10	360	360	1,8,24
5	5	720	720	1 (尽对 Stylus Color 使用专用纸时有效)

m : 垂直点的数目 (点阵图形的行数)

nL,nH 根据下面公式计算 nL,nH 值,从而得到水平点数目 (点阵图形的列数) 值

$$nH = \text{INT} \frac{\text{水平点数目}}{256} \quad nL = \text{MOD} \frac{\text{水平点数目}}{256}$$

$$k \quad \text{数据字节总数} \quad k = m \times \text{INT} \frac{(nH \times 256) + nL + 7}{8}$$

d 全图形模式时的图形数据

### [注意]

- (1) 在光栅图象模式中请不要变更打印分辨率。
- (2) 当选中 MicroWeave 时,m 必须设为 1。
- (3) 以 720 dpi 打印光栅图象时应使用 EPSON 提供的专用纸。
- (4) 本命令只在用 ESC (G 命令指定的光栅图象模式下才有效。
- (5) 最终的打印位置位于打印图象顶行最右边点后的一个点。
- (6) 超过右边界的打印数据将被忽略。
- (7) 请不要指定比打印分辨率更小单位的垂直移动。

ESC (e

控制点尺寸

ESC/P2

[代码]	ASCII	ESC	(	e	2	0	n1	n2
16 进制		1B	28	65	02	00	n1	n2
10 进制		27	40	105	2	0	n1	n2

[有效范围] n1=0  
n2=0,1,2

[功能]

根据 n2 的值,如下所示来控制点 (dot) 的尺寸。

n2=0: 缺省值 (缺省值是点尺寸 2)

n2=1: 点尺寸 1 (微小尺寸)

n2=2: 点尺寸 2 (标准尺寸)



[初值] 打印机初始化时选择缺省值

[注意]

- (1) 在本命令前要先执行“换页”动作。在换页后的 TOF 位置以外设定时，不能保证以后的动作。
- (2) 本代码只在一部文件中只限使用一次。
- (3) 本模式只在图象模式下有效。

ESC.2

选择光栅图象 (TIFF 压缩模式)

ESC/P2

[代码]	ASCII	ESC	.	2	v	h	1	0	0
	16 进制	1B	2E	2	v	h	1	0	0
	10 进制	27	46	2	v	h	1	0	0

[有效范围] v=5,10,20  
h=5,10,20

[功能]

- (1) 由本代码开始到 <EXIT> 为止的命令将作为 TIFF 压缩模式来处理。此后只有下面的 8 种代码有效，除此之外的代码将全被忽略。

<XFER> : 图形图象数据传送  
<MOVX> : 指定相对水平位置  
<MOVY> : 指定相对垂直位置  
<COLR> : 指定颜色  
<CR> : 水平位置返回左边界  
<EXIT> : 解除压缩模式  
<MOVXBYTE> : 将水平方向位置制定设为 8 个单位  
<MOVXDOT> : 将水平方向位置制定设为 1 个单位

- (2) 根据下面的参数来制定打印密度。

v: 垂直方向的打印密度 (3600/v DPI)

h: 水平方向的打印密度 (3600/h DPI)

v,h 只能取根据下表组合的值

v	h	垂直密度 x 水平密度
20	20	180 × 180 (DPI)
20	10	180 × 360 (DPI)
10	10	360 × 360 (DPI)
5	5	720 × 720 (DPI)

- (3) 根据本代码的执行，水平位置移到左边界位置，水平方向位置制定以点 (dot) 为单位。本代码的执行并不移动垂直位置。

[注意]

- (1) 本代码只在用 ESC (G 命令制定的图象模式下才有效。

- (2) 图象模式中请不要变更打印密度。
- (3) 请不要指定比打印密度更小单位的垂直移动。

[参照]

代码: ESC (G, ESC, 0, ESC, 1, ESC, 3, <XFER>, <MOVX>, <MOVY>, <COLR.>, <CR>, <EXIT>, <MOVXBYTE>, <MOVCDOT>。

解说: 《图象数据的种类》,《光栅图象》

ESC .3

光栅图象选择 (Delta row 压缩模式)

ESC/P2

[代码]	ASCII	ESC	.	3	v	h	1	0	0
	16 进制	1B	2E	3	v	h	1	0	0
	10 进制	27	46	3	v	h	1	0	0

[定义范围] v=5, 10, 20  
h=5, 10, 20

[功能]

(1)

由本代码开始到<EXIT>为止的命令将作为Delta row +TIFF压缩模式来处理。此后只有下面的8种代码有效,除此之外的代码将全被忽略。

- <XFER> : 图形图象数据传送
- <MOVX> : 指定相对水平位置
- <MOVY> : 指定相对垂直位置
- <COLR> : 指定颜色
- <CR> : 水平位置返回左边界
- <EXIT> : 解除压缩模式
- <MOVXBYTE> : 将水平方向位置指定设为8个单位
- <MOVXDOT> : 将水平方向位置指定设为1个单位
- <CLR> : 清除 Seed row

(2)

根据下面的参数来指定打印密度。

v : 垂直方向的打印密度 (3600/v DPI)

h : 水平方向的打印密度 (3600/h DPI)

v, h 只能根据下表组合的值

v	h	垂直密度 × 水平密度
20	20	180 × 180 (DPI)
20	10	180 × 360 (DPI)
10	10	360 × 360 (DPI)
5	5	720 × 720 (DPI)

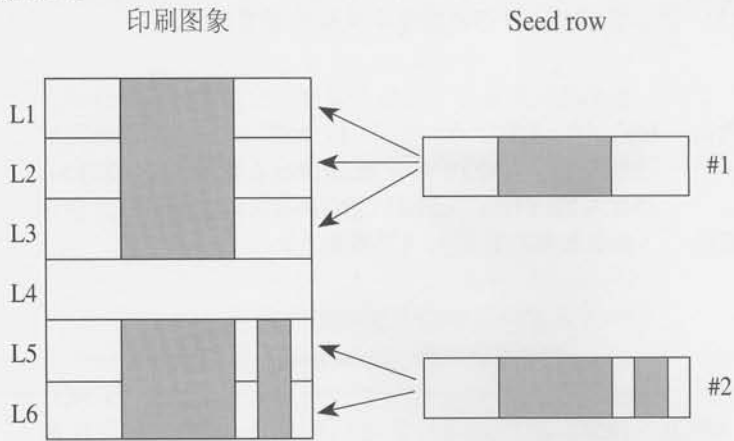
(3)

根据本代码的执行,全色的Seed row被清除。

(4)

根据本代码的执行,水平位置移到左边界位置,水平方向位置指定以点(dot)为单位。本代码的执行并不移动垂直位置。

<Delta row 的例子>



```
#1  L1 <MOVY>1/<COLR>Magenta/<MOVX>/<XFER>data/<CR>
    L2 <MOVY>2
    L3 <MOVEY>3
#2  L5 <MOVY>1/<COLR>Magenta/<CLR>/<MOVX>/<XFER> data/<CR>
    L6 <MOVY>6
```

[注意]

- (1) 本代码只在用 ESC (G 命令指定的图象模式下才有效。
- (2) 图象模式中, 请不要变更打印密度。
- (3) 请不要指定比打印密度更小单位的垂直移动。

[参照]

代码: ESC (G, ESC, 0, ESC, 1, ESC, 3, <XFER>, <MOVX>, <MOVY>, <COLR.>, <CR>, <EXIT>, <MOVXBYTE>, <MOVXDOT>, <CLR>

解说: 《图象数据的种类》, 《光栅图象》

在 <> 以后表示的代码, 只对由 ESC, 2 或 ESC, 3 来指定的光栅图象模式才有效。

<XFER>

图形数据传送

ESC/P2

[代码] ASCII <XFER> (nL nH) d1...dk  
二进制 001Fxxxx (nL nH) d1...dk

[有效范围] F=0, 1  
0< xxxx <15 (F=0 的时候)  
xxxx=1, 2 (F=1 的时候)  
0<nL<255  
0<nH<127  
0<d<255

[功能]

打印以下参数确定的压缩模式的光栅图形数据。

F: 低四位数的属性

F=0: 以低四位为计数器，确定图形数据的字节数 (k)。

如果 k 值超过 15 (超过 4 位二进制数所能表示的最大值) 请使 F=1。

F=1: 图形数据的字节数超过 15 (k) 时, 用  $k=nL+nH \times 256$  表示前四位 (xxxx) 用来表示计数器 nL,nH 的字节数。

k: 计数器的值 (传送图形数据的数量: 字节)

d1...dk: 光栅图形数据。

F的值	xxxx的值	计数器长	数据量(k)	被传送的代码
F=0	xxxx=k	4位	xxxx(0~15)	<XFER>d1...dk
F=1	xxxx=1	1字节	nL(15~255)	<XFER>nLd1...dk
	xxxx=2	2字节	nL+nH × 256 (256~65535)	<XFER>nLnHd1...dk

执行本命令代码后，水平方向移动到下一个位置，垂直方向不移动。



[注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。
- 在图形模式中，请不要改变打印密度。
- 图形数据不足时，紧随的代码被当作数据处理。数据过多时，多余的数据被当作代码处理。所以，数据量必须与指定的长度相吻合。
- 请不要送同一种颜色的图形数据。
- 请不要设置比打印密度还要小的垂直移动量。
- 忽略超过右边界的数据。
- 打印数据以 (计数器) + (数据) 的格式进行压缩传送。连续的不重复数据的字节数以正的计数表示，重复数据的重复次数以负的计数表示。各计数器的值是实际值 -1。
- 在 TIFF 压缩模式下 (ESC.)，以数据格式打印时，指定颜色的数据打印在指定行中。不影响别的行的打印。
- 在 Delta row 压缩模式下 (ESC.)，以本数据格式打印时，指定颜色的数据打印在指定行中的同时，数据被当作 Seed row 数据存储于记忆 (各种颜色)。

这个 Seed row 数据直到 <CLR> 命令代码被清除为止一直被当作打印数据存储已记忆，即使垂直移动到别的行时这个数据列仍有效，打印相同数据。

<MOVX>

指定相对水平位置

ESC/P2

[代码]            ASCII   <MOVX>   (nL nH)  
                   二进制   010Fxxxx   (nL nH)

[有效范围]    F=0, 1  
                   -8 < xxxx < 7 (F=0 的时候)  
                   xxxx=1, 2 (F=1 的时候)  
                   0 < nL < 255  
                   0 < nH < 127

[功能]

根下面的参数移动水平位置。

F: 低四位数的属性

F=0: 低四位 (xxxx) 作为位置指定参数指定位置。位置指定参数在 -8~7 的范围以外时 (四位数所能表示的范围以外)，请使 F=1。

F=1: 位置指定参数在 -8~7 的范围以外时 (四位数所能表示的范围以外)，用  $nL+nH \times 256$  表示，低四位 (xxxx) 用来表示位置指定参数的字节数。

k: 位置指定参数。

F的值	xxxx的值	位置指定参数(k)	被传送的代码
F=0	xxxx=k	xxxx(-8~7)	<MOVX>
F=1	xxxx=1	nL(-128~-9或8~127)	<MOVX>nL
	xxxx=2	nL+nH×256(-16384~-129或128~16384)	<MOVX>nLnH

- 移动后的水平位置 = 现在位置 + (位置指定参数) × 基准单位  
 (基准单位是由 <MOVXDOT>, <MOVXBYTE> 指定的单位)

[注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。
- 左移时指定值用 2 的补码表示，具体由下式可求得  
 $k \times \text{基准单位} = 16348 - \text{实际的移动量}$

[代码]        ASCII    <MOVY>    (nL nH)  
               二进制    011Fxxxx    (nL nH)

[有效范围]    F=0, 1  
                   0 < xxxx < 15 (F=0 的时候)  
                   xxxx=1, 2 (F=1 的时候)  
                   0 < nL < 255  
                   0 < nH < 127

## [功能]

根下面的参数移动垂直位置。

F: 低四位数的属性

F=0: 低四位 (xxxx) 作为位置指定参数指定位置。位置指定参数在大于 15 的情况下 (四位数所能表示的范围以外), 请使 F=1。

F=1: 位置指定参数在大于 15 的情况下 (四位数所能表示的范围以外), 用  $nL+nH \times 256$  表示, 低四位 (xxxx) 用来表示位置指定参数的字节数。

k: 位置指定参数 (点单位; 1 点 = 1/360 英寸)。

F 的值	xxxx 的值	位置指定参数(k)	被传送的代码
F=0	xxxx=k	xxxx(0~15)	<MOVY>
F=1	xxxx=1	nL(16~255)	<MOVY>nL
	xxxx=2	$nL+nH \times 256$ (256~65535)	<MOVY>nLnH

- 移动后的垂直位置 = 现在位置 + (位置指定参数)
- 本命令代码执行后, 水平位置移动到左边界。
- 在 Delta row 压缩模式下, 根据本命令代码 Seed row 数据被复制到行缓冲区 (Band Buffer)。

## [注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。
- 忽略超过 22 英寸的设置。
- 参数被当作正数处理, 不能设置向上 (逆) 的垂直位置移动。

## <COLR>

指定颜色

ESC/P2

[代码]        ASCII <COLR>  
              2 进制  1000xxxx

[有效范围]    xxxx=0000 : 黑  
              xxxx=0001 : 品红  
              xxxx=0010 : 蓝  
              xxxx=0100 : 黄

### [功能]

- 根据以上参数指定打印颜色。通过 <XFER> 送来的数据根据选择的颜色打印。
- 本命令代码执行后，水平位置移动到左边界。
- 在 Delta row 压缩模式下，根据本命令代码选择存取数据的 Seed row.

### [注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。
- 请不要指定混合色。

## <CLR>

解除 Seed row

ESC/P2

[代码]        ASCII <CLR>  
              2 进制  11100001

### [功能]

- 清除当前选择的颜色的 Seed row
- 水平方向移动到左边界位置，垂直位置不百变。

### [注意]

- 本命令代码只在由 ESC.3 所设定的 Delta row 压缩模式下才有效。

## <EXIT>

解除压缩模式

ESC/P2

[代码]        ASCII <EXIT>  
              2 进制  11100011

### [功能]

- 解除由 ESC.2 或 ESC.3 命令设置的压缩光栅图形模式。
- 打印当前行残余的图形数目据。
- 水平方向移动到左边界位置，垂直位置不变。

### [注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。

<CR>

水平位置初始化

ESC/P2

[代码]      ASCII <CR>  
             2 进制    11100010

[功能]

- 水平方向移动到左边界位置，垂直位置不变。

[注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。

<MOVXBYTE>

选择基准单位字节

ESC/P2

[代码]      ASCII <MOVXBYTE>  
             2 进制    11100100

[功能]

- 设置 <MOVX> 的移动单位为 8 个指定单位。指定单位是由 ESC (U 设置的基准单位)。
- 水平方向移动到左边界位置，垂直位置不变。

[注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。
- 请再进入由 ESC.2 或 ESC.3 设定的光栅图形模式之后立刻执行本命令代码。

<MOVXDOT>

选择基准单位点

ESC/P2

[代码]      ASCII <MOVXDOT>  
             2 进制    11100101

[功能]

- 设置 <MOVX> 的移动单位为 1 个指定单位。指定单位是由 ESC (U 设置的基准单位)。
- 水平方向移动到左边界位置，垂直位置不变。

[注意]

- 本命令代码只有在由 ESC.2 或 ESC.3 设定的光栅图形模式下才有效。
- 请再进入由 ESC.2 或 ESC.3 设定的光栅图形模式之后立刻执行本命令代码。



## 特殊功能打印命令

ESC ( B

打印条形码

格式	ASCII 码 :	ESC	(	B	n1	n2	k	m	s	v1	v2	c	数据
	十进制码 :	27	40	66	n1	n2	k	m	s	v1	v2	c	数据
	十六进制码 :	1b	28	42	n1	n2	k	m	s	v1	v2	c	数据

注释 :

n1+256 × n2 : 数据字节的个数

k : 选择条形码

k=0 : EAN -13

1 : EAN -8

2 : Interleaved 2 of 5

3 : UPC - A

4 : UPC - E

5 : Code 39

6 : Code 128

7 : POSTNET

m : 模数宽度 (180 dpi)

m=2 : 2 点

3 : 3 点

4 : 4 点

5 : 5 点

s : 空格调整值

$-3 \leq s \leq 3$  (1/360 英寸单位)

v1, v2 : 条长

$v1+v2 \times 256$  (1/180 英寸单位)

<+> 当选择 POSTNET 时, v1 和 v2 值被忽略。

<+> POSTNET 的长条形码长度总是 0.125 英寸; 短条形码长度总是 0.050 英寸。

# 附录

## 字符表

## 名词解释

### 附录一 字符表

以下这两个字符表可由 ESC t 指令选择,或由 DIP 切换开关和面板上的「选择 模式」来设定。  
(请参考第十三章)

#### 斜体字符表

此字符表可由 ESC t0 指令来设定。

CODE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL		0	@	P	'	p	NUL		0	@	P	'	p		
1	DC1	!	1	A	Q	a	q	DC1	!	1	A	Q	a	q		
2	DC2	"	2	B	R	b	r	DC2	"	2	B	R	b	r		
3	DC3	#	3	C	S	c	s	DC3	#	3	C	S	c	s		
4	DC4	\$	4	D	T	d	t	DC4	\$	4	D	T	d	t		
5		%	5	E	U	e	u		%	5	E	U	e	u		
6		&	6	F	V	f	v		&	6	F	V	f	v		
7	BEL	'	7	G	W	g	w	BEL	'	7	G	W	g	w		
8	BS CAN	(	8	H	X	h	x	BS CAN	(	8	H	X	h	x		
9	HT EM	)	9	I	Y	i	y	HT EM	)	9	I	Y	i	y		
A	LF	*	:	J	Z	j	z	LF	*	:	J	Z	j	z		
B	VT ESC	+	;	K	[	k	{	VT ESC	+	;	K	[	k	{		
C	FF	,	<	L	\	l		FF	,	<	L	\	l			
D	CR	-	=	M	]	m	}	CR	-	=	M	]	m	}		
E	SO	'	>	N	^	n	~	SO	'	>	N	^	n	~		
F	SI	/	?	O	_	o	DEL	SI	/	?	O	_	o			

## EPSON 扩充图形字符表

此字符表可由 ESC t1 指令来设定。在此表中,从 16 进制数 80H 到 9FH 之间的码可由 ESC 6 定义为字符,或是如同 00H 到 1FH 功能的控制码 (ESC 7)。

CODE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL		0	@	P	'	p	Ç	É	á	:	L	⊥	α	≡	
1	DC1	!	1	A	Q	a	q	ü	æ	Í	:	⊥	⊥	β	±	
2	DC2	"	2	B	R	b	r	é	Æ	Ó	:	⊥	⊥	Γ	≥	
3	DC3	#	3	C	S	c	s	â	Ô	ú		†	⊥	π	≤	
4	DC4	\$	4	D	T	d	t	ä	Ö	ñ	†	—	⊥	Σ		
5	§	%	5	E	U	e	u	à	Ò	Ñ	†	+	⊥	σ	J	
6		&	6	F	V	f	v	ã	û	ä	†	†	⊥	μ	÷	
7	BEL	'	7	G	W	g	w	ç	ù	ö	⊥	⊥	†	τ	≈	
8	BS CAN	(	8	H	X	h	x	ê	ÿ	¿	⊥	⊥	+	Φ	°	
9	HT EM	)	9	I	Y	i	y	ë	Ö	⊥	⊥	⊥	⊥	θ	•	
A	LF	*	:	J	Z	j	z	è	Ü	⊥	⊥	⊥	⊥	Ω	'	
B	VT ESC	+	;	K	[	k	{	ï	ç	½	⊥	⊥	⊥	δ	√	
C	FF	,	<	L	\	l		î	£	¼	⊥	⊥	⊥	∞	n	
D	CR	-	=	M	]	m	}	ì	¥	¡	⊥	=	⊥	∅	2	
E	SO	'	>	N	^	n	~	Ä	Pt	«	⊥	⊥	⊥	ε	■	
F	SI	/	?	O	_	o	DEL	Å	f	»	⊥	⊥	⊥	∩		

## 附录二 名词解释

### ■ 初始设定值 (Initialization)

当设备的电源被打开,重设或重新启动时,所生效的设定值。譬如,10 CPI 字符模式便是初始设定的字符模式,因此,除非用户要求打印机以别的模式打印;否则,打印机便会一直以 10 CPI 字符宽度来打印。

### ■ 回车 (Carriage Return)

可以把打印位置送回左边界控制码;但是双向打印时,打印头并不真正返回左边界。

### ■ DIP 切换开关 (DIP Switch)

在打印机上,用以控制各种不同功能的小开关;至于 DIP 意思则是“双排包装”(Dual In-line Package)。而这些开关可用以改变打印机的初始设定值。

### ■ 缓冲区 (Buffer)

打印机和电脑一样,也有存储器。当用户要打印出电脑中的文件资料时,这些资料会先快速的送到打印机的存储器;然后,再以较慢的速度打印出。而且,当打印机在打印资料时,电脑还可以处理别的事情。

### ■ 初始化 (Initiate)

利用 ESC @ 指令、INIT 讯号或重新打开打印机的电源, 来使打印机返回初始设定状况的操作。

### ■ 页缝空白功能 (Skip Over Peforation)

此项功能会在上一页最后一行和次页第一行之间产生一英寸的空白。

### ■ 行间距 (Line Spacing)

即行与行之间的距离。标准的行间距是 1/6 英寸; 也就是在 11 英寸的打印纸上, 可打印出 66 行的文字。

### ■ 字符宽度 (Character Width)

字符的宽度包括字符实际的宽度和字符左右边的空格。

### ■ 水平定位 (Horizontal tab setting)

水平定位点是设定在左边界右方的定位点, 当打印机接收到执行水平定位动作 (HT) 的指令时, 便将打印头推进至下一水平定位点的位置。

### ■ 垂直定位点 (Vertical tab setting)

垂直定位点位置是以距页面顶端的距离来设定的定位点, 当下达执行垂直定位 (VT) 指令时, 便会将打印头推进至下一个垂直定位点的位置。

### ■ 垂直通道中定位点设定 (VFU, Vertical Format Unit)

在打印机中有八个通道可储存八组不同的垂直定位点, 在需要时可选择用任一通道中所设定的垂直定位点。

### ■ 只读存储器 (ROM, Read Only Memory)

打印机内存放固定资料的存储器。虽然打印机可以使用 ROM 资料, 但却不能加以改变。

### ■ 随机存储器 (RAM, Random Access Memory)

打印机内用作缓冲区以存放用户定义字符的存储器。当电源关掉时, 存入 RAM 中的所有资料也便遗失。

### ■ 二进制数 (Decimal)

见“数系”的说明。

### ■ 十六进制数 (Hexadecimal)

参见“数系”的说明。

### ■ 数系 (Number System)

在打印机中, 常用的数系有下列三种:

- 基底为 10, 所用的数字为 0、1、2、3、4、5、6、7、8 和 9 的十进制数系。  
(此一数系就是我们所最熟悉的数系。)
- 基底为 16, 所用的数字为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E 和 F 的十六进制数系。这一个数系是程序员最常用的数系; 而且, 只需要两个十六进制数便可以表示 0 到 255 的十进制数。

• 基底为 2, 所用的数字为 0 和 1 的数系。电脑系统中的所有资料都是以二进制数的形式来表示, 而且, 以 ON 或 OFF 的电气讯号来表示。其中, 二进制数字通常也称作位。而 0 到 255 的十进制数则必须以 8 位的二进制数来表示。

#### ■ 中文字码 (Chinese Character Code)

中文字码可分为三类:

- 「输入码」(Input Code), 也叫「外码」。这就是每个中文字符输入电脑时所要敲入的键序。因为中文输入法种类繁多, 所以输入码也就各有差异。
- 「内码」(Internal Code), 就是每个字符依电脑系统的内部作业规则所设定的代表号码。
- 「交换码」(Interchange Code), 就是在不同的电脑系统资料互相交换时所使用的标准。

#### ■ 地址 (Address)

每一内置字型在打印机内存放的实际位置编号。

#### ■ CPS (Character Per Second)

即打印机每秒可打印的字符数, 为计算打印机打印速度的单位。

#### ■ CPI (Character Per Inch)

即每英寸可打印的字符数, 为计算打印密度的单位。

#### ■ DPI (Dot Per Inch)

即每英寸可打印的点数, 为评估分辨率的单位。

#### ■ 打印品质 (Print Quality)

是指打印字符的两种方式。若选择文书字体打印时, 会以较多的点来打印出字符, 以提高打印的品质, 但打印的速度较慢; 草体打印时, 会用较少的点来打印出字符, 以求得较快的打印速度, 但打印的品质较差。

#### ■ 比例模式打印 (Proportional Printing)

在这一种打印模式下, 各个字符所给定的空间和字符的宽度成比例。譬如, 窄字符 (如 i) 的空间便比宽字符 (如 W) 的空间窄。

#### ■ 压缩 (Condensed)

打印宽度约为标准字符宽度的 60%, 譬如, 压缩的 10 CPI 字符每一英寸便能打印出 17 个。至于在应用上, 压缩模式可应用于表格, 以便把宽广的表格缩小而打印在打印纸上。

#### ■ 重叠 (Double - strike)

每一个字符都被打印两次的一种打印模式, 其中, 第二次稍微在第一次的下方打印出。

#### ■ 粗体 (Emphasized)

每一个字符都被打印两次的一种打印模式, 其中, 第二次稍微靠第一次的右方打印出。

#### ■ 上标模式 (Superscript)

在字符空间的上半部, 英文字符以正常字符的 2/3 大小来打印字符。中文字符则以全角字符的 1/4 来打印出。

### ■ 下标模式 (Subscript)

在字符空间的下半部,英文字符以正常字符的 2/3 大小来打印字符。中文字符则是全角字符的 1/4 大小。

### ■ 位图 (Bit image graphics)

由点的图案所组成的图形;其中,每一个点都由一个位的资料所控制。

### ■ 用户自定义字符 (Down-load Characters)

由用户定义而存放在打印机的 RAM 存储器中的字符。有时候,也称作使用者定义的字符 (user-defined characters)。

### ■ 资料倾印 (Data Dump)

一种查错的模式;当打印机处在十六进制数的倾印模式时,每个接收到的代码都以十六进制数的形式表现出来。而有时候也叫作十六进数倾印 (hexadecimal dump)。

### ■ 双向打印 (Bidirectional printing)

在打印时,打印头每隔一行由左到右;而在其他的行时,则由右打印到左。由于打印头可以双向的打印,因此,可提高打印的速度。

### ■ 单向打印 (Unidirectional printing)

打印头只以一个方向来打印;因此,可提高垂直对齐的准确度。

### ■ 联机 (On-Line)

当打印机处在联机的状态时,打印机便可以和连接的电脑通信;反之,当打印机处在脱机的状态时,即使打印机和电脑之间已连接了电缆,仍然不能通信。

### ■ EPSON 扩充图形字符 (EPSON Expanded Graphics)

EPSON 扩充绘图字符组包括国际重音字符,希腊字母和一些绘图所需的字符。

### ■ 传输率 (Baud rate)

测量资料传输速度的一种方法;大约相当于每秒钟的位数,一个字节至少需要 11 个位来表示。

### ■ 接口 (Interface)

打印机和电脑之间的连接电缆的插口。其中,串行接口 (Serial interface) 是每一次送一个位的资料;并行接口 (Parallel interface) 则每一次送一个字节的资料。

### ■ 位 (Bit)

二进制数字 0 或 1;这也就是打印机或电脑所使用的最小单位。

### ■ 字节 (Byte)

用来代表一个字符或一个控制码的一序列二进制数;其中,一个字节包含 8 个位。

**EPSON®**