

# ZPLII条码打印机 中文编程手册

## THERMAL BARCODE PRINTER SERIES

## Revise History

Date	Description	Remark
2015-05-26	新版本修订	陶玮
2020-02-20	增加RFID相关指令	冯远明

^A 使用字体编号调用字体.....	6
^A@ 使用字体路径调用字体.....	7
^B0 Aztec 条码.....	8
^B1 Code 11 条码.....	9
^B2 交叉二五码.....	10
^B3 Code 39 码.....	11
^B4 Code 49 码.....	12
^B5 Planet 条码.....	14
^B7 PDF417 条码.....	15
^B8 EAN-8 条码.....	17
^B9 UPC-E 条码.....	18
^BA Code 93 条码.....	19
^BC Code 128 条码.....	21
^BE EAN-13 条码.....	25
^BF Micro PDF417 条码.....	26
^BI Industrial 2 of 5 条码.....	27
^BJ Standard 2 of 5 条码.....	28
^BK ANSI Codabar 条码.....	29
^BL LOGMARS 条码.....	30
^BM MSI 条码.....	31
^BO Aztec 条码.....	32
^BP Plessey 条码.....	33
^BQ QR 条码.....	34
^BS UPC / EAN 扩展.....	35
^BU UPC-A 条码.....	36
^BX Data Matrix 条码.....	37
^BY 条码字段默认参数设置.....	39
^BZ POSTNET 条码.....	40
^CC, ~CC 改变格式指令前缀.....	41
^CD, ~CD 改变 ZPL 分隔符.....	42
^CF 改变字符字体默认字体.....	43
^CI 国际字符设置.....	44
^CM 转换指定盘符名.....	45
^CT, ~CT 改变指令前缀.....	46
^CV 代码检验.....	47
^CW 字体标识.....	48
~DB 下载点阵字体.....	49
~DE 下载编码表.....	51
^DF 下载格式.....	52
~DG 下载图象.....	53
~DN 取消下载图象.....	55
~DU 下载 TrueType 字体.....	56
~EG 从存储中删除所有图象.....	57

^FB 字段块.....	58
^FD 字段数据.....	60
^FH 十六进制转义字符.....	61
^FN 字段编号指令.....	62
^FO 字段位置.....	63
^FP 字段参数.....	64
^FR 字段反相打印.....	65
^FS 字段分隔.....	66
^FT 字段排版.....	67
^FV 可变数据段.....	68
^FW 字段方向.....	69
^FX 注解.....	70
^GB 画框.....	71
^GC 画圆.....	72
^GD 画斜线.....	73
^GE 画椭圆.....	74
^GF 区域绘图.....	75
^GS 符号图像.....	76
^HF 返传标签文件.....	77
^HG 返传图像数据.....	78
^HH 返传配置信息.....	79
^HM 返传 RAM 信息.....	80
~HS 返传状态信息.....	81
^ID 删除对象.....	84
^IL 图像加载.....	85
^IM 图像移动.....	86
^IS 图像存储.....	87
^JB 初始化闪存.....	89
~JD 开启通信诊断功能.....	90
~JE 关闭通信诊断功能.....	91
^JM 设置分辨率.....	92
^JU 配置更新.....	93
^LH 坐标原点.....	94
^LL 标签长度.....	95
^LR 反相打印标签.....	96
^LS 标签左右偏移.....	97
^LT 标签上下偏移.....	98
^MC 清除上张标签内容.....	99
^MD 打印相对浓度.....	100
^MM 打印模式.....	101
^MN 纸张类型.....	102
^MT 介质类型.....	103
^MU 设置长度单位.....	104
^PH ~PH 跳至原点位置.....	105

^PM 打印镜像标签.....	106
^PO 打印方向.....	107
^PQ 打印数量.....	108
^PR 打印速度.....	109
~PS 打印重新开始.....	110
^PW 打印宽度.....	111
^RF 读写 RFID 标签内容.....	112
^RR 重读次数.....	113
^RS 设置 RFID 参数.....	114
^RZ 设置 RFID 标签密码.....	115
~SD 打印绝对浓度.....	116
^SE 选择编码表.....	117
^SF 序列字段.....	118
^SN 序列数据.....	120
~TA 撕离位置调整.....	121
^TB 文本框.....	122
^TO 转移对象.....	123
~WC 打印配置标签.....	124
^WD 打印目录标签.....	125
^XA 标签开始.....	126
^XF 调用标签.....	127
^XG 调用图像.....	128
^XZ 标签结束.....	129

## ^A 使用字体编号调用字体

^A(可缩放/点阵字体)命令用于内置的 True Type 字体。可缩放字体（也可以认为是平滑矢量字体）能够以点为单位来对横向、纵向进行扩展。点阵字体由点阵像素组成，通常高度大于宽度。

内置的可缩放字体(A0=CG Triumvirate Bold Condensed)默认为不旋转，15 点高，12 点宽。打印机将从^A 命令得到字体的旋转方向、宽度、高度等参数。

### ^A 命令的格式:

^Af, o, h, w

f = 字体名

默认值: 0

其他值: A-Z, 0-9 (打印机的任何字体, 包括下载字体, EPROM 中储存的, 当然这些字体必须用^CW 来定义为 A-Z, 0-9)

o = 方向

默认值: ^FW 默认值或上一个^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 字符高度

矢量字体:

默认值: 15 点或上一次^CF 的值。

可接受 10-1500 点的值:

点阵字体:

默认值: 指定点阵字体的标准高度

其他值: 标准高度的整数倍, 2-10。

w = 字符宽度

矢量字体:

默认值: 12 点或上一次^CF 的值, 也可以显示为 0

可接受 10-1500 点的值:

点阵字体:

默认值: 指定点阵字体的标准宽度

其他值: 标准高度的整数倍, 2-10。

## ^A@ 使用字体路径调用字体

^A@（使用字体名来调用字体）命令使用字体的全名，来调用字体

示例：

```
^XAA@N, 25, 25, R:CYRILLIC.FNT^F0100, 20^FS
^FDThis is a test.^FS
^A@N, 50, 50^F0200, 40^FS
^FDThis string uses the R:Cyrillic.FNT^FS
^XZ
```

第一行命令将查找 RAM 中的“Cyrillic.FNT”字体，当字体找到后，命令将继续定义字符的方向，大小，然后在标签上打印字段数据“This is a test.”

一旦^A@定义了“Cyrillic.FNT”，它的作用将一直保持到下一个^A@命令定义一个新字体名时。

在本例中第二个^A@命令中，字符的大小增加了，设置了新的旋转方向，打印机用同样的字体打印出字段数据“This string use the B:Cyrillic.FNT.”

^A@o, h, w, n

o = 方向

默认值：上一个^FW 值，如省略则为 N。

其他值：N = 正常（Normal）

R = 顺时针旋转 90 度（Roated）

I = 顺时针旋转 180 度（Inverted）

B = 顺时针旋转 270 度（Bottom）

h = 字符高度（以点为单位）

缩放字体：整个字符高度的点数，放大倍数不是必须的，因为字符是可缩放的。

点阵字体：数值将四舍五入为字体基本高度的整数倍，然后再除以基本高度，得到最接近的放大倍数。

w = 字符宽度（以点为单位）

缩放字体：整个字符宽度的点数，放大倍数不是必须的，因为字符是可缩放的。

点阵字体：数值将四舍五入为字体基本宽度的整数倍，然后再除以基本宽度，得到最接近的放大倍数。

n = 字体名（按照 ZPL 命名惯例）

如果没有指定设备号的字母，默认的设备是 RAM，或 R:。如果不指定字体名，上一个^A@指定的字体仍将发生作用。

## **^B0 Aztec 条码**

^B0 (Aztec) 条码是指令会创建一个围绕中心牛眼图样的正方形二维码图形。

**^B0 命令的格式:**

**^B0a, b, c, d, e, f, g**

a = 条码方向

默认值: ^FW 默认值或上一个 ^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

b = 放大倍数

默认值: 3

其他值: 1 到 10

c = 扩展码标示

默认值: N (如果数据包含 ECICs)

其他值: Y (如果数据不包含 ECICs)

d = 纠错等级和编码大小/生成样式

默认值: 0

其他值: 01-99 纠错等级百分比(最小值)

101-104 1 到 4 层压缩编码

201-232 1 到 32 层完整编码

300 一个简单的 Aztec 码” Rune”

e = 手动编码标示

默认值: N = 自动编码

其他值: Y = 手动编码(扫描器预设)

f = 附加编码数量

默认值: 1

其他值: 1 到 26

g = 可选参数 附加 ID

默认值: 没有 ID

其他值: ID 是一个最大长度 24 个字符的字符串



## ^B1 Code 11 条码

^B1 (Code 11)也就是 USD-8 码。在 Code 11 条码中，每个字符由三个条和两个空格组成，字符集为 10 个数字和连接符(-)。

打印比例调整：2.0 到 3.0

^FD(Field Data:数据字段)限制：100+字符。实际总数据由^BY 的比例与标签的宽度（如果旋转，则指长度）

^B1 命令的格式：

^B1o, e, h, f, g

o = 方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

e = 校验位

默认值：N(No) = 2 位校验位

其他值：Y(Yes) = 1 位校验位

h = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印(Yes)

其他值：N = 不打印(No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

## ^B2 交叉二五码

^B2 (交叉二五码) 是高密度的、自校验的、连续的、数字的符号。它的每个字符由五个元素组成：五个条或五个空。这五个元素中，两个是宽的三个是窄的。条码由条和空交叉组成。

条码中，每个字符由三个条的两个空组成，字符集为 10 个数字和连接符(-)。

打印比例调整：2.0 到 3.0

^FD(Field Data:数据字段)限制：100+字符。实际总数据由^BY 的比例与标签的宽度（如果旋转，则指长度）

由定义可知，交叉二五码的位数必须是偶数。如果是奇数，则打印机自动在接收到的数据前加上 0。

交叉二五码使用模 10 校验法。 字符集：数字 0-10

### ^B2 命令的格式：

^B2o, h, f, g, e

o = 方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

e = 校验位

默认值：N(No) = 2 位校验位

其他值：Y(Yes) = 1 位校验位

## ^B3 Code 39 码

^B3 (Code 39 码)是很多行业的标准，为美国国防部(DOD)所采用，是美国国家标准协会(ANSI)MH10.8-1983 的三种识别符号之一。也叫作 USD-3 码或 39 码。

39 码的每个字符由九个元素组成：五个条，四个空和字符间隙，这九个元素中，三个是宽的，六个是窄的。

打印比例调整：2.0 到 3.0

^FD(Field Data:数据字段)限制：100+字符。实际总数据由^BY 的比例与标签的宽度（如果旋转，则指长度）

Code 39 码可以兼容 128 位 ASCII 字符集，参看表 A 和表 B

字符集：数字 0-10，A-Z, space, -, . , \$, / , + , %

### ^B3 命令的格式：

^B3o, e, h, f, g

o = 方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

e = 模 43 校验位

默认值：N(No) = 不打印校验位

其他值：Y(Yes) = 打印校验位

h = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印(Yes)

其他值：N = 不打印(No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

Code 39 码的起始位和终止位是自动生成的。

## ^B4 Code 49 码

^B4 (Code 49 码) 是一种多行、连续、可变长的兼容 128 位全 ASCII 字符集的条码。它较完美的实现了在较小的空间存储了大量的数据。

128 码一般 2 到 8 层，每层由左空白区、层起始符、4 个符号字符、层终止符及右空白区组成。层与层之间由一个模块的层分隔条分隔。Code 49 码的每个符号字符可表示 2 个基本字符集中的字符。各层能以任意次序扫描。

打印比例是固定的。

### ^B4 命令的格式：

^B4o, h, f, m

o = 方向

默认值：^FW 默认值或上一个 ^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 每层高度

定义：

默认值：由 ^BY 值确认。

其他值：1 到标签高度

注意：1 并不是可取的数值

f = 打印识别码

默认值：N = 不打印 (No)

其他值：A = 打印在条码的上方 (Above)

B = 打印在条码的下方 (Below)

注意：当条码超过 2 层，识别码将越过右边缘。

m = 开始模式

默认值：A = 自动模式。打印机分析数据字段并决定起始模式。

0 = 规范字母数字模式

1 = 多种可读字母数字

2 = 规则数字模式

其他值：3 = 数字字母组

4 = 规则数字字母变化 1

5 = 规则数字字母变化 2

## Code 49 字段数据字符集

当采用起始模式为 0 到 5 时，^FD 送往打印机的数据是基于 Code49 码的内部字符集的。它们在下图的第一列。字符：<=>和? 是 Code 49 码的控制字符。

使用模式 0—5 时，有效的字段数据都将被支持。变化字符由一个转换字符和一个非转换字符组成一个两个字符的序列。例如，打印小写的字符 a，先送一个“SHIFT2”，接着再送“A”(>A)，如果打印识别码，识别码上将出现小写的 a。

**注意：Code49 码只使用大写字母。**

如果检测到一个无效的字串，Code49 格式化器将停止解释，并打印一个符号来代替这个无效的字串，下面就是一些无效的字串的例子。

用除了 0 到 9 或空格的任意字符结束数字模式

用了第四种模式但数据字段起始字符并不是 SHIFT 1 字符集。

用了第五种模式但数据字段起始字符并不是 SHIFT 2 字符集。

送了一个 SHIFT 1 字符但下一个字符不在 SHIFT 1 字符集中。

送了一个 SHIFT 2 字符但下一个字符不在 SHIFT 2 字符集中。

送了两个 SHIFT 1 或 SHIFT 2 控制字符。

## 使用 Code 49 自动模式的优势

使用自动模式（默认值）可完全不用人工选择起始模式和人工控制字符变化。自动模式分接收的 ASCII 字符串，选择适当的模式，处理字符变化，以最大密度压缩数据。

**注意：**当大于等于 5 位数字时，只能选择数字模式，数字模式在数字字符串长度小于 8 时在与字母方式相比在条码所占空间并没有多大优势。

## ^B5 Planet 条码

邮码是所有条码打印机支持的常用条码。

条码字符集是 0 - 9。

**^B5 命令的格式:**

**^B5o, h, f, g**

**o** = 方向

默认值: ^FW 默认值或上一个 ^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

**h** = 条码高度

默认值: 由 ^BY 设置

其他值: 1 到 9999 点

**f** = 打印识别码

默认值: Y = 打印 (Yes)

其他值: N = 不打印 (No)

**g** = 将识别码打印在条码上方

默认值: N = 不打印在条码上方

其他值: Y = 打印在条码上方

## ^B7 PDF417 条码

^B7 (PDF417) 条码命令是一个二维、多层、连续、堆栈的符号码。这种条码可以在每个标签上打印超过 1000 个字节。它完美地适用于那些需要在读条码时读入大量信息的应用。

代码由 3 到 90 个堆栈层组成，每一层由开始、停止和符号字符构成的码词组成，每个码词包含四个条和四个空。每层最少必须有 3 个码词。

打印比例固定。

**^B7 命令的格式：**

**^B7o, h, s, c, r, t**

**o** = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

**h** = 每层高度

默认值：由^BY 值确认。

其他值：1 到标签高度

注意：1 并不是可取的数值

**s** = 这个数字表示错误检查和纠错级别，默认值是只检测错误（不纠错），提高安全级别将增加纠错能力，当然也增加了符号的大小）

默认值：0 = 只检测错误。

其他值：1 到 8。纠错能力从小到大。

注意：当条码超过 2 层，识别码将越过右边缘。

**c** = 编码的数据的列数

用户可以指定每列的 codeword 来控制符号的宽度。

默认值：1: 2 层 / 列

其他值：1 到 30

**r** = 编码的层数

用户可以指定层数来控制符号的高度。

默认值：1: 2 层 / 列

其他值：3-90

示例：如果没有输入层数和列数，有 72 个码字，共 6 列 12 层（随着码字的不同，比例并不是一定的。）

**t** = 截去层指示和停止区

默认值：N = 不截去 (No)，打印层指示和终止符

其他值：Y = 截去 (No)，不打印层指示和终止符

**注意：**

1. 如果层和列都是用户自定义的，它们的乘积必须小于 928。
2. 如果 层 × 列 > 928 将什么都不打印。
3. 如果 码字总数 > 层 × 列 将不打印。

4. 条码不能连续变化。
5. 截短功能用在标签不会损坏的场合，右连的层指示和终止符可以减少一个单独的模块。

## 使用 PDF417 时特殊的<sup>^</sup>BY 命令

使用 PDF417 时<sup>^</sup>Byw, r, h 的参数意义如下：

w = 模块宽度（默认值=2），上限为 10。

r = 比例（默认值=3）固定。在 PDF417 中没作用。

h = 条码高度。整个条码的高度。如果 PDF417 在<sup>^</sup>B7 中没有指定高度时，将使用这个值作为整个条码的高度。

## 使用 PDF417 时特殊的<sup>^</sup>FD 字符集

除了对打印机有特殊意义的那些字符外，所有 ASCII 字符都可送往打印机。在所有<sup>^</sup>FD 语句中 CR / LF 已成为合法字符。用下列办法替换。

“\&” = 回车 / 换行

“\(\*)” = 软连字号（单词断开时使用），如果连字符放在一行的最后，它会被打印出来，如果它不是在在一行的最后，将会被忽略。（在<sup>^</sup>B7 中忽略）

“\\” = 字符 \，（注意必须选择<sup>^</sup>CI13，使它能打印 \）

(\*) = 任意字母 / 数字



## ^B8 EAN-8 条码

^B8(EAN-8)条码命令是 EAN-13 的缩短版，参见 EAN-13 可以得到更多的关于 EAN 的信息。EAN 是国际物品编码协会 (European Article Number Association) 的字母的首位。

每个 EAN-8 的条码符号由四个元素组成：两个条，两个空。

打印比例固定。

^FD(Field data, 字段数据)限制：必须 7 位，ZPLII 自动在大于 7 位时截去超过的部分，在小于 7 位于在前面加上 0。

**注意：**JAN-8 (日本物品编码) 系统是 EAN-8 中的特殊应用。在这种情况下，送给打印机的前两位数据永远是 49。

EAN-8 字符集：0 到 9。

**^B8 命令的格式：**

^B8o, h, f, g

o = 条码方向

默认值：^FW 默认值或上一个 ^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值：由 ^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

## ^B9 UPC-E 条码

^B9(UPC-E)条码命令是UPC系统的一种变化用于数字系统0，UPC是美国统一代码(Universal Product Code)的缩写。它是UPC-A带有压缩0的缩短版，这样打印空间就可以减少。通常用于印制小标签。

**注意：使用带压缩0的UPC码，用户必须输入满10位的字符串，ZPLII将计算并打印出缩短版UPC。**

每个UPC-E的条码符号由四个元素组成：两个条，两个空。

打印比例固定。

^FD(Field data, 字段数据)限制：必须10位字符，5位制造商代码，5位产品代码。

UPC-9字符集：0到9。

如想进一步了解，可参阅UPC码规范。

**^B9命令的格式：**

^B9o, h, f, g, e

o = 条码方向

默认值：^FW默认值或上一个^FW的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转90度 (Roated)

I = 顺时针旋转180度 (Inverted)

B = 顺时针旋转270度 (Bottom)

h = 条码高度

默认值：由^BY设置

其他值：1到9999点

f = 打印识别码

默认值：Y = 打印(Yes)

其他值：N = 不打印(No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

e = 打印校验位

默认值：Y = 打印(Yes)

其他值：N = 不打印(No)

**四种规则适合产品编码：**

1. 如果厂商编码的最后编码为000, 100, 或200, 有效的产品编码为00000—00999。
2. 如果厂商编码的最后三位为300, 400, 500, 600, 700, 800或900, 有效的产品编码为00000—00099。
3. 如果厂商编码的最后两位为10, 20, 30, 40, 50, 60, 70, 80或90, 有效的产品编码为00000—00009。
4. 如果厂商编码的最后一位不为0, 有效的产品编码为00005—00009。

# ^BA Code 93 条码

^BA(Code 93)条码命令是一种变长、连续型符号码。它可就用在许多同 39 码一样的场合。它使用 128 位全 ASCII 码，但由于 ZPLII 不支持 ASCII 的控制代码或 ESC 序列，所以它用以下字符来替代。该代码也叫 USS-93。

控制码	ZPL II 表示
Ctrl \$	&
Ctrl %	%
Ctrl /	(
Ctrl +	)

每个 Code 93 码的字符由六个元素组成：三个条，三个空。所不同的是，可阅读的注释上将把所用的控制代码了打印出来。

注意：控制码是成对使用的，如须知详情，参阅 93 码规范。

打印比例固定。

^FD(Field Data:数据字段)限制：100+字符。实际总数据由^BY 的比例与标签的宽度（如果旋转，则指长度）

Code93 字符集：0 到 9、A 到 Z，- . \$ / + % & ? ( ) space

如想进一步了解，可参阅 UPC 码规范。

## ^BA 命令的格式：

^BAo, h, f, g, e

o = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印(Yes)

其他值：N = 不打印(No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

e = 打印校验位

默认值：Y = 打印(Yes)

其他值：N = 不打印(No)

Code 39 码能够对前 128 位 ASCII 码字符集进行编码，请看下表。

## THERMAL BARCODE PRINTER SERIES

ASCII I	93 码	ASCII I	93 码	ASCII I	93 码	ASCII I	93 码	ASCII I	93 码	ASCII I	93 码	ASCII I	93 码	ASCII I	93 码
NUL	'U	DEL	&P	SP	Spac e	0	0	@	'V	P	P	'	'W	p	)P
SOH	&A	DC1	&Q	!	(A	1	1	A	A	Q	Q	a	)A	q	)Q
STX	&B	DC2	&R	“	(B	2	2	B	B	R	R	b	)B	r	)R
ETX	&C	DC3	&S	#	(C	3	3	C	C	S	S	c	)C	s	)S
EOT	&D	DC4	&T	\$	\$	4	4	D	D	T	T	d	)D	t	)T
ENQ	&E	NAK	&U	%	%	5	5	E	E	U	U	e	)E	u	)U
ACK	&F	SYN	&V	&	(F	6	6	F	F	V	V	f	)F	v	)V
BEL	&G	ETB	&W	'	(G	7	7	G	G	W	W	g	)G	w	)W
BS	&H	CAN	&X	(	(H	8	8	H	H	X	X	h	)H	x	)X
HT	&I	EM	&Y	)	(I	9	9	I	I	Y	Y	i	)I	y	)Y
LF	&J	SUB	&Z	*	(J	:	(Z	J	J	Z	Z	j	)J	z	)Z
VT	&K	ESC	'A	+	+	:	'F	K	K	[	'K	k	)K	{	'P
FF	&L	FS	'B	,	(L	<	'G	L	L	/	'L	l	)L		'Q
CR	&M	FS	'C	—	—	=	'H	M	M	]	'M	m	)M	}	'R
SO	&N	RS	'D	.	.	>	'I	N	N	^	'N	n	)N	~	'S
SI	&O	US	'E	/	/	?	'J	O	O		'O	o	)O	DEL	'T

## ^BC Code 128 条码

^BC(Code 128)条码是一种高密度、变长、连续型字母数字型符号码。它被设计用来对一些复杂的产品标识进行编码。

CODE128 码有三个字符子集。每个子集可对 106 个可打印字符进行编码。所以每个字符在不同的子集有不同的含义。

每个 Code 128 码的字符由六个元素组成：三个条，三个空。所不同的是，可阅读的注释上将把所用的控制代码了打印出来。

注意：控制码是成对使用的，如须知详情，参阅 93 码规范。

打印比例固定。

^FD(Field Data:数据字段)限制：100+字符。实际总数据由^BY 的比例与标签的宽度确定。（如果旋转，则指长度）

### ^BC 命令的格式：

^BCo, h, f, g, e, m

o = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

e = 打印 UCC 校验位

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

m = 模式

默认值：N = 不选择模式

其他值：U = UCC 匹配模式。

(^FD 或^SN 后必须有 19 位数字。)用 FNC1 值自动选择 C 子集。

A = 自动模式。

自动模式将分析收到的数据并自动决定选取最佳的压缩方式。^FD 可以用全部 ASCII 字符。打印机将确定是否变化子集。四个及以下的数字将自动变化为 C 子集。

D = UCC/EAN 模式。

这个模式会使用 UCC/EAN 方式来产生条码。条码会使用合适的字符子集接 FNC1 码来表示一个 UCC/EAN 128 条码。打印机会自动在编码

时把双括号和空格分开，并打印在可识别码上。打印机会自动判断是否使用校验位并计算和打印在可识别码上。可识别码的大小也会自动调整。

## 128 码字符集

值	A 代 码	B 代 码	C 代 码	值	A 代 码	B 代 码	C 代 码	值	A 代 码	B 代 码	C 代 码	值	A 代 码	B 代 码	C 代 码
0	SP	SP	00	28	<	<	28	56	X	X	56	84	DC4	t	84
1	!	!	01	29	=	=	29	57	Y	Y	57	85	NAK	u	85
2	“	“	02	30	>	>	30	58	Z	Z	58	86	SYN	v	86
3	#	#	03	31	?	?	31	59	[	[	59	87	ETB	w	87
4	¥	¥	04	32	@	@	32	60	/	/	60	88	CAN	x	88
5	%	%	05	33	A	A	33	61	]	]	61	89	EM	y	89
6	&	&	06	34	B	B	34	62			62	90	SUB	z	90
7	‘	‘	07	35	C	C	35	63			63	91	EM	{	91
8	(	(	08	36	D	D	36	64	NUL	—	64	92	FS		92
9	)	)	09	37	E	E	37	65	SOH	.	65	93	GS	}	93
10	*	*	10	38	F	F	38	66	STX	a	66	94	RS	~	94
11	+	+	11	39	G	G	39	67	ETX	b	67	95	US	DEL	95
12	,	,	12	40	H	H	40	68	EOT	c	68	96	FNC3		96
13	—	—	13	41	I	I	41	69	ENQ	d	69	97	FNC2		97
14	.	.	14	42	J	J	42	70	ACK	e	70	98	SHIFT		98
15	/	/	15	43	K	K	43	71	BEL	f	71	99	Code C		99
16	0	0	16	44	L	L	44	72	BS	g	72	100	Code B	FNC4	CODE B
17	1	1	17	45	M	M	45	73	HT	h	73	101	FNC4	Code A	
18	2	2	18	46	N	N	46	74	LF	i	74	102	FNC1		
19	3	3	19	47	O	O	47	75	VT	j	75	103	A 子集起始符		
20	4	4	20	48	P	P	48	76	FF	k	76	104	B 子集起始符		
21	5	5	21	49	Q	Q	49	77	CR	l	77	105	C 子集起始符		
22	6	6	22	50	R	R	50	78	SO	m	78				
23	7	7	23	51	S	S	51	79	SI	n	79				
24	8	8	24	52	T	T	52	80	DLE	o	80				
25	9	9	25	53	U	U	53	81	DC1	p	81				
26	:	:	26	54	V	V	54	82	DC2	q	82				
27	;	;	27	55	W	W	55	83	DC3	r	83				

## 选择 UCC 匹配模式的特殊情况

1. 在`^FD`和`^SN`语句中超过 19 位的数据将被截去。
2. 在`^FD`和`^SN`语句中小于 19 位的数据将被自动在前面加 0 到满 19 位。

## CODE 128 子集

CODE128 有三种子集：A 子集，B 子集，C 子集。子集可以有两种方法选择。

1. 可以在数据字段(`^FD`)的条码前加上一个特殊的转换符号。
2. 在字段数据前加上起始字符，如果没有输入起始代码，将选用 B 子集。

在一个条码内部改变子集，要将转换符号置于字段数据的合适的位置。这样新的子集将一直作用到遇到下一个转换符号为止。（例如：在 C 子集中，数据字段中的“>7”将转换为 A 子集。）下表显示 CODE128 的转换代码与三个子集的起始代码。

替代码	十进制值	A 子集字符	B 子集字符	C 子集字符
><	62			
>0	30	>	>	
>=	94		~	
>1	95	USQ	DEL	
>2	96	FNC3	FNC3	
>3	97	FNC2	FNC2	
>4	98	SHIFT	SHIFT	
>5	99	CODE C	CODE C	
>6	100	CODE B	FNC4	CODE B
>7	101	FNC4	CODE A	CODE A
>8	102	FNC1	FNC1	FNC1
	起始符			
>9	103	以 Code A 开始（成对的字母 / 数字）		
>:	104	以 Code B 开始（一般的字母 / 数字）		
>;	105	以 Code C 开始（所有的数字 00—99）		

## Code 128 的示例——B 子集

由于 Code 128 的 B 子集是最常用的子集，在字符串没有指定起始符时 ZPLII 默认使用 B 子集。以下是两个示例。

这两个示例的条码是一模一样的。

前两条命令(`^XA^F0100,75`)开始标签格式并设置字段位置。(从左上角开始)到条码字段 x, y 座标为(100,75)。

第三条命令(`^BCN,100,Y,N,N`)打印不旋转的高度为 100 点的 Code128 条码。

第四条命令(左图中的`^FDCODE128`和右图中的`^FD>:CODE128`)指定了条码的内容。

第五条命令`^XZ`表明结束打印字段并结束标签格式。

当 UCC 校验位关闭后，识别码将打印在条码下方。

**注意：**左图并没有指定任何子集，所以使用 B 子集，在右图中，则指定使用 B 子集。由于 ZPLII 默认使用 CODE B，本示例对于转换符在命令中的运用是一个很好的练习。

Code 128 B 子集是直接 ASCII 文本进行编程的，除了几个数值大于 94 的几个特殊字符：

^ > ~

这些字符必须用转换字符来编程。

## Code 128 的示例——A 子集和 C 子集

Code 128 码 C 子集中对双数字进行编程。在 A 子集中，双数字中的每个数字都作为一个单独的字符来打印，在 C 子集中则作为一个码字来打印。下图右为 A 子集的示例。（“>9”是 A 子集的起始符）

下图中左图和中图是 C 子集。请注意这些条码是一样的，在中图中，字母 D 被忽略，2 和 4 是一对数字。



## ^BE EAN-13 条码

^BE(EAN-8)条码命令是类似 UPC-A 的条码，它在欧州和日本的零售商场得到非常广泛的应用。

EAN-13 条码有 12 位数据，比 UPC-A 多一个字符，EAN-13 兼容 UPC-A，但有 13 位数字，左手区为 6 位数字。第 13 位是校验位。

打印比例固定。

^FD(Field data, 字段数据)限制：必须 12 位，ZPLII 自动在大于 12 位时截去超过的部分，在小于 12 位时在前面加上 0。

EAN-13 使用模 10 法生成校验位进行错误检测。（参见附录 C）

注意：JAN-13（日本物品编码）系统是 EAN-8 中的特殊应用。在这种情况下，送给 打印机的前两位数据永远是 49。

EAN-13 字符集：0 到 9。

**^BE 命令的格式：**

**^BEo, h, f, g**

o = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

# ^BF Micro PDF417 条码

^BF(微型 PDF417)条码是二维、多层、连续、堆栈型和 PDF417 一样，不同之处在于 PDF417 使用 17 个模块宽的起始符终止符和左右层指示。而微型 PDF417 使用独特的 10 个模块宽的层地址符来缩短整个条码的宽度。并允许层高尽量低于 2X。

微型 PDF417 设计用于需要增加单位面积的效率，但又不需 PDF417 那样大的数据容量。它最多只打印 44 层 乘 4 列的数据。

^FD(Field data, 字段数据)和^FH(Field Hex)限制：250 个 7-比特的文本字符，150 个 8 比特的十六进制字符，或 366 个 4 比特的数字字符。

## ^BF 命令的格式：

^BFo, h, m

o = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值：由^BY 设置，如果^BY 没有设定，则为 10 点

其他值：1 到 9999 点，超出范围则设为最接近的值

m = 模式

默认值：0 (查表)

可选择：0—33 (看下表)

模式	数据列	数据层	模式	数据列	数据层	模式	数据列	数据层	模式	数据列	数据层	模式	数据列	数据层
0	1	11	7	2	11	14	3	8	21	3	38	28	4	20
1	1	14	8	2	14	15	3	10	22	3	44	29	4	26
2	1	17	9	2	17	16	3	12	23	4	6	30	4	32
3	1	20	10	2	20	17	3	15	24	4	8	31	4	38
4	1	24	11	2	23	18	3	20	25	4	10	32	4	44
5	1	28	12	2	26	19	3	26	26	4	12	33	4	4
6	2	8	13	3	6	20	3	32	27	4	15			

## ^BI Industrial 2 of 5 条码

^BI(工业二五码)条码是自校验, 连续的数字型条码。工业 25 码是 25 家族码中最长的条码。在 ZPLIII 中, 25 码家族还有交叉 25 码和标准 25 码。

在工业 25 码中, 所有信息由条组成。宽条是窄条的宽度的 3 倍。

打印比例: 2.0 到 3.0。

^FD(Field Data:数据字段)限制: 100+字符。实际总数据由^BY 的比例与标签的宽度确定 (如果旋转, 则指长度)  
字符集: 数字 0—9, Start(自动), stop(自动)。

### ^BI 命令的格式:

^BIo, h, f, g

o = 条码方向

默认值: ^FW 默认值或上一个^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值: 由^BY 设置

其他值: 1 到 9999 点

f = 打印识别码

默认值: Y = 打印 (Yes)

其他值: N = 不打印 (No)

g = 将识别码打印在条码上方

默认值: N = 不打印在条码上方

其他值: Y = 打印在条码上方

## ^BJ Standard 2 of 5 条码

^BJ(标准二五码)条码是自校验, 连续的数字型条码。

在标准 25 码中, 所有信息由条组成。每个条之间的宽度为 2 倍的条宽, 宽条是窄条的宽度的 3 倍。

打印比例: 2.0 到 3.0。

^FD(Field Data:数据字段)限制: 100+字符。实际总数据由^BY 的比例与标签的宽度确定 (如果旋转, 则指长度)

字符集: 数字 0—9, Start (内部使用), stop (内部使用)。

### ^BJ 命令的格式:

^BJo, h, f, g

o = 条码方向

默认值: ^FW 默认值或上一个^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值: 由^BY 设置

其他值: 1 到 9999 点

f = 打印识别码

默认值: Y = 打印 (Yes)

其他值: N = 不打印 (No)

g = 将识别码打印在条码上方

默认值: N = 不打印在条码上方

其他值: Y = 打印在条码上方

## ^BK ANSI Codabar 条码

^BK(ANSI 库德巴) 条码通常广泛地用于图书馆、药品工业和快递公司的应用中。这种条码也叫作 USD-4 码, NW-7 或 27 码。最初是开发来作为价格标牌用的。

每个字符由 7 个元素组成。四个条和三个空。它有两个字符集: (1) 数字, (2) 控制符, 起始符 (— : . \$ / +) 停止符 (A B C D E T N \* )。

打印比例: 2.0:1 到 3.0:1。

^FD(Field Data:数据字段)限制: 100+字符。实际总数据由^BY 的比例与标签的宽度 (如果旋转, 则指长度)

### ^BK 命令的格式:

^BKo, e, h, f, g, k, L

o = 条码方向

默认值: ^FW 默认值或上一个^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

e = 校验位

默认值: N(No) = 不打印校验位

h = 条码高度

默认值: 由^BY 设置

其他值: 1 到 9999 点

f = 打印识别码

默认值: Y = 打印 (Yes)

其他值: N = 不打印 (No)

g = 将识别码打印在条码上方

默认值: N = 不打印在条码上方

其他值: Y = 打印在条码上方

k = 起始符

默认值: A

其他值: B, C, D, E, N, T, \*

l = 停止符

默认值: A

其他值: B, C, D, E, N, T, \*

参数 k 和 l 确定起始符和终止符。

# ^BL LOGMARS 条码

^BL (LOGMARS) 条码在美国国防部中作为 39 码的一种特殊应用。LOGMARS 是后勤自动标识与识读符号应用 (Logistics Applications of Automated Marking and Reading Symbols) 的缩写。

打印比例: 2.0:1 到 3.0:1。

^FD (Field Data: 数据字段) 限制: 100+ 字符。实际总数据由 ^BY 的比例与标签的宽度确定。(如果旋转, 则指长度)

字符集: 同 39 码。

如想知道更多的信息, 请参阅 LOGMARS 条码规范。

## ^BL 命令的格式:

^BL o, h, g

o = 条码方向

默认值: ^FW 默认值或上一个 ^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值: 由 ^BY 设置

其他值: 1 到 9999 点

g = 将识别码打印在条码上方

默认值: N = 不打印在条码上方

其他值: Y = 打印在条码上方

## ^BM MSI 条码

^BM(MSI) 条码是一种脉冲宽度调制的，连续的，无自校验的条码。它是 Plessey 条码的一种变形。

每个字符由 8 个元素组成。四个条和四个相邻的空。

打印比例：2.0:1 到 3.0:1。

^FD(Field Data:数据字段)限制：参数 e 为 B 时 1 到 14 位数字，C 或 D 时 1 到 13 位数字，参数 e 为 A 时，加上静区。

**^BM 命令的格式：**

**^BMo, e, h, f, g, e2**

o = 条码方向

默认值：^FW 默认值或上一个 ^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

e = 选择校验位

默认值：B = 1 Mod 10

其他值：A = 无校验位

C = 2 Mod 10

D = 1 Mod 10 和 1 Mod 11

h = 条码高度

默认值：由 ^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

e2 = 将校验位打印在识别码内

默认值：N = 不打印在识别码内

其他值：Y = 打印在识别码内

## **^B0 Aztec 条码**

^B0 (Aztec) 条码是指令会创建一个围绕中心牛眼图样的正方形二维码图形。

**^B0 命令的格式:**

**^B0a, b, c, d, e, f, g**

a = 条码方向

默认值: ^FW 默认值或上一个 ^FW 的值。

其他值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

b = 放大倍数

默认值: 3

其他值: 1 到 10

c = 扩展码标示

默认值: N (如果数据包含 ECICs)

其他值: Y (如果数据不包含 ECICs)

d = 纠错等级和编码大小/生成样式

默认值: 0

其他值: 01-99 纠错等级百分比(最小值)

101-104 1 到 4 层压缩编码

201-232 1 到 32 层完整编码

300 一个简单的 Aztec 码” Rune”

e = 手动编码标示

默认值: N = 自动编码

其他值: Y = 手动编码(扫描器预设)

f = 附加编码数量

默认值: 1

其他值: 1 到 26

g = 可选参数 附加 ID

默认值: 没有 ID

其他值: ID 是一个最大长度 24 个字符的字符串



## ^BP Plessey 条码

^BP(Plessey) 条码是脉冲宽度调制的，连续的，无自校验的条码。它是 Plessey 条码的一种变形。

每个字符由 8 个元素组成。四个条和四个相邻的空。

打印比例：2.0:1 到 3.0:1。

字符集：0-9，A-F。

^FD(Field Data:数据字段)限制：100+字符。实际总数据由^BY 的比例与标签的宽度确定。（如果旋转，则指长度）

### ^BP 命令的格式：

^BPo, e, h, f, g

o = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

e = 打印校验位 (CRC8 2 位)

默认值：N = 不打印 (No)

其他值：Y = 打印 (Yes)

h = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

## ^BQ QR 条码

^BQ(Plessey) 是在一个正方形范围内用大量的小的正方形模块组成的矩阵图像，在四角的其中三个位置有特殊的正方块辅助判断二维码的大小，位置和旋转角度。

多种符号尺寸选择，四种纠错率可选，用户自定义模块尺寸，可生成多种图像形式。

QR code 模式 1 是原始的技术规格，而 QR CODE 模式 2 就是象征着一个高级的形式/加强版。

模式 2 有一些额外的特性并且能够自动区别于模式 1。

建议使用模式 2，并且通常应该也是要用模式 2

此条码内容由^FD(Field Data:数据字段)定义

字符集包括数字的数据，字母数字数据，8 位字节数据，以及日本字符

**^BQ 命令的格式：**

**^BQa, b, c, d, e**

- a = field 方向  
固定值：正向(^FW 没有旋转效果)
- b = 模式  
默认值：2(增强型)  
其他值：1(原始型)
- c = 放大因子  
默认值：2(200dpi 机器)/3(300dpi 机器)  
其他值：1 到 10
- d = 纠错率  
默认值：Q(参数为空)/M(参数非法)  
其他值：H = 超高纠错等级  
Q = 高纠错等级  
M = 普通纠错等级  
L = 高密度等级
- e = 掩码  
默认值：7  
其他值：0 到 7

## ^BS UPC / EAN 扩展

^BS(UPC / EAN 扩展)是出版商的 ISBN(国际标准书号)的 2 到 5 位的附加数字。它是 ISBN (国际标准书号)的扩展。这些扩展字符作为一个独立的条码来打印。

由于扫描器设计用来扫描单独的 UPC 码，尽管扩展符兼容 UPC 码，但还是不能作为 UPC 码的一部分被扫描器识别。因为 UPC/EAN 扩展码有一个警戒字符（左边，编码为 1011），这个字符用来表示分隔。

打印比例：固定。

字符集：0—9

^FD(Field Data:数据字段)限制：2 个或 5 个字符，ZPLII 自动截短或在左边添加 0，直到满足要求。

### ^BS 命令的格式：

^BS<sub>o, h, f, g</sub>

<sub>o</sub> = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

<sub>h</sub> = 条码高度

默认值：由^BY 设置

其他值：1 到 9999 点

<sub>f</sub> = 打印识别码

默认值：Y = 打印(Yes)

其他值：N = 不打印(No)

<sub>g</sub> = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

UPC 码，选择宽度为 2 个模式（默认值），打印位置为

	UPC-A		UPC-E	
	X 轴偏移	Y 轴偏移	X 轴偏移	Y 轴偏移
正常	209 点	21 点	122 点	21 点
旋转	0	209 点	0	122 点

还有，条码必须比主条码小 27 点（0.135 英寸），当 UPC 码高度为 183 点时（0.9 英寸）扩展条码高度为 155 点。

下例是如何建立一个值为 7000002198，扩展码为 04414 的 UPC-A 码。

^XA

^FO100,100^BY3

^BUN,137

^FD07000002198^FS

^FO400,121

^BSN,117

^FD04414^FS

^XZ

## ^BU UPC-A 条码

^BU (UPC-A) 条码是固定长度、数字、连续型的条码，它主要用于零售业的包装标签。UPC-A 条码有 11 个数据字符。  
一个 8 点的打印头能够按 77% 的比例打印条码。

要想知道更多，可参考 UPC-A 条码规范。

打印比例：固定。

字符集：0-9

^FD(Field Data:数据字段)限制：11 个字符，ZPLII 自动截短或在左边添加 0，直到满足要求。

UPC-A 码使用模 10 法来计算生成校验位。

### ^BU 命令的格式：

^BUo, h, f, g, e

o = 条码方向

默认值：^FW 默认值或上一个 ^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值：由 ^BY 设置

其他值：1 到 9999 点

f = 打印识别码

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

g = 将识别码打印在条码上方

默认值：N = 不打印在条码上方

其他值：Y = 打印在条码上方

e = 打印 UCC 校验位

默认值：Y = 打印 (Yes)

其他值：N = 不打印 (No)

识别码风格由 ^BY 确定的窄条的宽度决定。

8 点打印头：3 点或以上用 OCR-B 字体来打印识别码，1 点或 1 点用 A 字体打印。

12 点打印头：4 点或以上用 OCR-B 字体来打印识别码，1，2，3 点用 A 字体打印。

## ^BX Data Matrix 条码

^BX（数据矩阵）条码是一个二维的矩阵条码。

^BX 命令的格式：

^BXo, h, s, c, r, f, g, a

o = 条码方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常（Normal）

R = 顺时针旋转 90 度（Roated）

I = 顺时针旋转 180 度（Inverted）

B = 顺时针旋转 270 度（Bottom）

h = 分层高度

默认值：由^BY 设置

其他值：1 到条码高度

s = 质量等级

默认值：0

其他值：0, 50, 80, 100, 140, 200

质量等级指的是纠错率增加的图像符号的总量。以 ECC 值作为基准。ECC 为 50-140 使用回卷编码，ECC 200 则使用罗门码编码。

c = 编码列数

默认值：自动

其他值：9 到 49

奇数只能对质量等级 0 到 140 生效，偶数只能对质量等级 200 生效。矩阵二维码的行数和列数是自动调整的。有时可能需要强制固定高或者宽达到条码统一的效果，如果尝试强制定义一个过小的图像，则不会有任何图像输出，如果值大于 49，则行列数调整成 0，默认使用自动行列调整输出正常图像。如果输入了一个偶数值，则输出无效参数。如果 a 值小于 9 但不等于 0 或者数据量超过了行列设定大小，不会打印任何图像。如果设置了^CV，输出无效参数。

r = 编码行数

默认值：自动

其他值：9 到 49

f = 格式 ID

默认值：6

其他值：1 = 数据字段为数字+空格

2 = 数据字段为大写字母+空格

3 = 数据字段为大写字母+空格+句号，逗号，连接符和左斜杠

4 = 数据字段为数字+大写字母+空格

5 = 数据字段为 128 个 ASCII 码

6 = 数据字段为 256 个 ISO 码

g = 转义控制符

默认值：~

其他值：其他字符

这个参数仅在质量等级为 200 时使用。是嵌入在数据字段中的转义控制字符。  
注意，使用这个参数时必须指定 a 值，不指定则忽略此参数。

a = 高宽比  
默认值：正方  
其他值：矩形

## ^BY 条码字段默认参数设置

^BY 命令用来改变窄元素（窄条或窄空）的宽度模块。宽条与窄条的比例与条码的高度。它在一个标签格式中通常是必要的。

比例	模块宽度（点）									
	1	2	3	4	5	6	7	8	9	10
2.0	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1
2.1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2:1	2.1:1
2.2	2:1	2:1	2:1	2:1	2.2:1	2.16:1	2.1:1	2.12:1	2.1:1	2.2:1
2.3	2:1	2:1	2.3:1	2.25:1	2.2:1	2.16:1	2.28:1	2.25:1	2.2:1	2.3:1
2.4	2:1	2:1	2.3:1	2.25:1	2.4:1	2.3:1	2.28:1	2.37:1	2.3:1	2.4:1
2.5	2:1	2.5:1	2.3:1	2.5:1	2.4:1	2.5:1	2.4:1	2.5:1	2.4:1	2.5:1
2.6	2:1	2.5:1	2.3:1	2.5:1	2.6:1	2.5:1	2.57:1	2.5:1	2.5:1	2.6:1
2.7	2:1	2.5:1	2.6:1	2.5:1	2.6:1	2.6:1	2.57:1	2.65:1	2.6:1	2.7:1
2.8	2:1	2.5:1	2.6:1	2.75:1	2.8:1	2.6:1	2.7:1	2.75:1	2.7:1	2.8:1
2.9	2:1	2.5:1	2.6:1	2.75:1	2.8:1	2.8:1	2.87:1	2.87:1	2.8:1	2.9:1
3.0	3:1	3:1	3:1	3:1	3:1	3:1	3:1	3:1	3:1	3:1

^BY 命令的格式:

^BYw,r,h

w = 模块（窄条）宽

开机初始化值：2 点

可接受的数值：1-10 点

r = 宽条与窄条的比例

开机初始化值：3.0

可接受的数值：2.0 到 3.0，0.1 的增量（对固定比例的条码无效）

h = 条码高度

开机初始化值：1 0 点

可接受的数值：1 点到标签高度。

参数 r, 是真实的比例是和点数 w 有关的，关系可见上表。

例如，选择 w 为 9, r 为 2.4 时，窄条宽度为 9 点，宽条为  $9 \times 2.4 = 21.6$  点，然而，打印机将自动四舍五入到最近的点，也就是说，宽条的宽度为 22 点。

这样，条码的比例是 2.44 (22 除以 9)，它的比例接近于 2.4。

条码的模块宽和高（参数 w 和 h）在任何时候可以用 ^BY 修改来对后面的条码产生作用。

注意：

1. 在一个标签格式中一旦使用了 ^BY 命令后，它将一直作用到下一个 ^BY 命令时。
2. 参数 h 是其他条码命令忽略高度参数时，从 ^BY 所给的 h 中获得高度值。

## ^BZ POSTNET 条码

^BZ (POSTNET, 邮政网) 条码用于自动分检信件。POSTNET 条码一系列 5 个条, 2 高 3 短, 对应数字 0 到 9。

打印比例: 固定。

字符集: 0—9

^FD(Field Data:数据字段)限制: 100+字符。实际总数据由^BY 的比例与标签的宽度 (如果旋转, 则指长度)

**注意:** 如果^CV (代码有效性) 被激活, 数据字段长度必须为 5, 9, 或 11 位。如果是 6 位长的话, 也可以用空格和 “—”

**^BZ 命令的格式:**

^BZo, h, f, g

o = 条码方向

默认值: ^FW 默认值或上一个^FW 的值。

其他

值: N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 条码高度

默认值: 由^BY 设置

其他值: 1 到 9999 点

f = 打印识别码

默认值: N = 不打印 (No)

其他值: Y = 打印 (Yes)

g = 将识别码打印在条码上方

默认值: N = 不打印在条码上方

其他值: Y = 打印在条码上方



## **^CC, ~CC 改变格式指令前缀**

^CC, ~CC (改变脱字符) 指令是用于改变指令前缀。缺省前缀是脱字符 (^)。

**^CC, ~CC 指令格式**

**^CCx, ~CCx**

x = 任何 ASCII 字符

缺省值：要求有参数。如不用参数，下一字符接收后作为新的前缀字符。

**注意：**不要设任何和另一个前缀相同的值。

## **^CD, ~CD 改变 ZPL 分隔符**

^CD, ~CD (改变分隔符) 指令是用于改变 ZPLII 分隔符。这符号是用来分离输入 ZPL 指令中几个参数。缺省的分隔符 ( , ) 。

**^CD, ~CD 指令格式**

**^CDx, ~CDx**

x = 任何 ASCII 字符

缺省值：要求有参数。如不用参数，下一字符接收后作为新的前缀字符。

## **^CF 改变字符字体默认字体**

你可用<sup>^</sup>CF（改变字母缺省字体）指令，使你程序变得更简单。

### **指令格式**

<sup>^</sup>CFf, h, w

f = 指定默认字体

A = 字体 A （开机初始化值）

其它值：B 到 H，和 0~9

（任何在打印机的字体，包括下载字体，EPROM 贮存字体和字体 A-Z 和 1-9 可通过<sup>^</sup>CW 选择）

h = 单个字符点高度

可接受的数值：0~9999 开机初始化值 =9

w = 单个字符点宽度

可接受的数值：0~9999 开机初始化值 =5

参数 f 指定所有字符字段的缺省字体。参数 h 指定对每一字符字段的缺省高度，参数指定每一字符字段的缺省宽度。

缺省字体字母是 A，如果不改变缺省字体字母（<sup>^</sup>CF 指令），也不用任何段指令字母（<sup>^</sup>Af）或输入一个非法字体值，将用字体 A 打印数据。

定义仅仅高或宽，放大比例由参数定义。如值被定义，上次所给高和宽是<sup>^</sup>CF 设定值或缺省<sup>^</sup>CF 值。如果两个都没有定义，使用最后一个<sup>^</sup>CF 值给出的值或默认的<sup>^</sup>CF 值给出高度和宽度。

## ^CI 国际字符设置

斑马打印机可打印各种国际字符字体：美国 1，美国 2，英国，荷兰，丹麦，挪威，瑞典，芬兰，德国，法国 1，法国 2，意大利，西班牙等。ZPLII 遵循国际字符的 ISO 标准。

^CI（改变国际字体）指令允许你调用国际字符设置给你打印。你可混合表中字符设置。以下说明可用国际字符设置。调用一个国际字符设置指令是：

**^CI x**

x = 希望的字符设置  
 0 = USA1 （开机初始化设置）  
 其它可接受的值  
 1=美国 2，  
 2=英国，  
 3=荷兰，  
 4=丹麦/挪威，  
 5=瑞典/芬兰，  
 6=德国，  
 7=法国 1，  
 8=法国 2，  
 9=意大利，  
 10= 西班牙，  
 11=其它，  
 12=日本，  
 13=IBM 850 码  
 18-23 = 保留。  
 24= 8 位存取的 Unicode 编码字体。  
 开机初始化值= 上次永久保存的值。

\*编码由转换表(\*.DAT)控制。该表可由 ZTOOLS 对 TrueType 字库进行编码时生成。

## **^CM 转换指定盘符名**

^CM 会重新指定一个新盘符名给打印机已经存在的存储盘符名。如果一个字体文件已经存在，可以通过此方法避免强制性变更或者重写已经存在的字体文件。

使用此指令会影响到后续所有用到指定盘符的指令

**^CM a, b, c, d**

a = 盘符 B 的别名

b = 盘符 E 的别名

c = 盘符 R 的别名

d = 盘符 A 的别名

## **^CT, ~CT 改变指令前缀**

^CT, ~CT (改变分隔符) 指令是用于改变指令前缀。缺省前缀是 (^)。

**^CT, ~CT 指令格式**

**^CTx, ~CTx**

x = 任何 ASCII 字符

缺省值：要求有参数。如不用参数，下一字符接收后作为新的前缀字符。

## ^CV 代码检验

^CV(代码检验)是一个开关，它打开或关闭代码检验功能。当代码检验打开时，所有的条码数据都将检查以下几种错误情况：

- 字符不在字符集中
- 校验位不正确
- 数据字段太长（字符太多）
- 数据字段太短（字符太少）
- 参数串包含了不正确的参数或丢失参数

检测到非法字符时，打印机将打印一个错误信息和代码来替代代码图像。信息以“INVALID-X”的方式显示，X 是以下的错误代码：

C = 字符不在字符集中

E = 校验位不正确

L = 数据字段太长（字符太多）

S = 数据字段太短（字符太少）

P = 参数串包含了不正确的参数或丢失参数

一旦打开了条码检验功能，^CV 将一直保持作用到下一个^CV 关闭条码检验功能。本命令不永久保存。

**注：如果存在不止一个错误，显示第一个错误。**

^CV 命令的格式是：

^CVa

a = 代码检验

默认值：N=不检测

其他值：Y= 检测

## ^CW 字体标识

所有机内字体有一字符标识如 A、B、C 等。^CW（字体标识）指令分配单个字母字符（A 到 Z 和 0 到 9）用于下载到 DRAM R: 的字体，内存卡 B: ， EPROM E: ， 或内建的 Z: 。

如果字符和机内字体符相同，则下载字体代替机内字体。当格式里调用机内字体时，下载字体将被打印在标签上。如替换了机内字体，这改变起作用到关机。

如给的字符是不同的，下载字体是作为一种增加的字体。所给的字体将保存直到新的指令输入或打印机关机。

^CW 指令格式是

^Cwa, d, f

a = 替代机内字体字母或加入新字体。（要求输入一个字符）

d = 贮存字体源设备（附加。缺省是 R: ）。

f = 下载字体名替换机内字体或增加字体。 .TTF .TTE .FNT

缺省: Unknown

以下是用 ^CW 指令一些例子

将 MYFONT.FNT 贮存在 DRAM 中，格式称为字体 A:

^XA^CWA, R:MYFONT.FNT^XZ

将 MYFONT.FNT 贮存在 DRAM 中，增加字体 Q:

^XA^CWQ, R:MYFONT.FNT^XZ

将 NEWFONT.FNT 贮存在 DRAM 中，格式称为 F:

^XA^CWF, R:MYFONT.FNT^XZ



## ~DB 下载点阵字体

~DB（下载点阵字体）指令设置打印机接收下载点阵字体，定义原始单元尺寸，基准线，空间大小和版权。

这指令由两部份组成，一个 ZPLII 指令定义字体和数据结构段内每个字符字体。

以下是一个使用~DB 命令的例子。它将头两个字符下载到 DRAM 中。

```
~DBR:TIMES.FNT,N,5,24,3,10,2,PRINTER 1992,
```

```
#0025.5.16.2.5.18.
```

```
00FF
```

```
00FF
```

```
FF00
```

```
FF00
```

```
FFFF
```

```
#0037.4.24.3.6.26.
```

```
00FF00
```

```
0F00F0
```

```
0F00F0
```

```
00FF00
```

### ~DB 指令格式

```
~DBd,o,x,a,h,w,base,space,#char,©,DATA
```

d = 贮存字体目标设备（附加。缺省是 R:）

o = 字体名, 1-8 字符,

默认值: 缺省 = Unknown

x = 扩展名, 3 字符（固定的, 始终是: FNT）

a = 字体旋转

缺省

值: N = 不旋转

其他 R = 90,

I = 180,

值: B = 270

目前仅支持 N。

h = 最大单元的高度（点）

w = 最大单元的宽度（点）

base = 单元项的点到字符基准线

space = 空的宽度或字间距

#char = 字体字符数（这必须符合下载的字符数）

© = 版权者（最大文本串长度是 63 字符）

DATA = ASCII 数据结构定义字体每个字符。# 符号表示字符码参数被分隔。字符码是 1 到 4 允许国际字符集下载到打印机。

### 数据结构是

```
# × × × × . h . w . x . y . i . data
```

# = 字符代码

× × × ×

h = 点阵高度（点）

w = 点阵宽度（点）

x = x 方向（点）

y = y 方向（点）

i = 排版移动替换（宽度包括字体字符间的间隙）

data = 十六进制点阵描述

## ~DE 下载编码表

标准的 Windows TrueType 字体的编码是 Unicode。因此，ZPL 字段数据必须将其他编码转换成 Unicode。它需要用 ~DE(下载编码)命令下载一个对照表。这些表由 Ztools for Windows 提供。

~DE 的格式：

~DEd:o, x, s, DATA

d = 贮存字体目标设备 （附加。缺省是 R: ）

o = 字体名, 1-8 字符,

默认值： 缺省 = Unknown

x = 扩展名, 3 字符 （ 固定的，始终是：DAT）

n = 表名

默认值：没有名字 =本命令被忽略

其他值：输入一个带标识符的目标设备的表名（最大为 8 位长度）

目标设备的标识符：

R: = RAM 内存

B: = 内存卡

s = 表的大小

默认值：没有数值 = 本命令被忽略

其他值：输入所占内存的字节数

DATA = 数据串

默认值：没有数值 = 本命令被忽略

其他值：用 ASCII 码表达的 HEX 值（2HEX/字节）

2 位的数值必须匹配表的大小。（一个字节数不足的数值会使整个命令被忽略）

## ^DF 下载格式

^DF (下载格式)命令以文本格式保存 ZPLII 格式指令，以后可以用^XF 加上变量来调用。保存的格式中可以包含字段数 (^FN)，以便将来参考调用。

将格式保存节省了传输时间，但并没有节省格式化时间，因为 ZPL II 是作为文本串保存的，在打印时仍须进行格式化。

如果忽略映象名，默认的名称和扩展名为 UNKNOW.ZPL。在^XA 后马上输入^DF 指令，输入的格式指令就会被保存起来。

**注：一个包含^DF 的格式不会被打印出来。**

### ^DF 命令的格式

^DFd:o.x

d = 保存映象的目标设备名（附加。缺省是 R:）

o = 映象名，1 到 8 个字母或数字（默认的名字为 UNKNOW）

x = 扩展名，3 位字母或数字（永远是.ZPL）

以下有一个使用^DF 的示范，它将 ZPLII 文本串保存到 DRAM 中，保存的文本串叫 STOREFMT.ZPL。

^XA

^DFR:STOREFMT.ZPL^FS

^F025,25

^AD,36,20^FN1^FS

^F0165,25

^AD,36,20^FN2^FS

^F025,75

^AB,22,14^FDBUILT BY^FS ^F025,125

^AE,28,15^FN1

^XZ

^XA

^XFR:STOREFMT.ZPL^FS

^FN1^FDPRINTER^FS

^XZ

## ~DG 下载图象

~DG（下载图象）执行以下功能。

1. 设置打印机为图象模式。
2. 命名图形。（这个名字将用来在标签中调用）
3. 定义图象尺寸
4. 下载十六进制字符串到打印机

**注意：**

1. 为使打印机明白，使用图形名时将以空格、句点、或扩展名结束
2. 避免偶然被空格替换，不要在图象名中用空格。不同图象用不同名字。
3. 如二个图象用了同一名送到打印机。第一个图象被删除而由第二个图象代替。

~DG 指令格式

~DGd:o, x, t, w, DATA

d = 贮存图象的目标设备。

缺省值：R: (DRAM)

其它值：E:

o = 图象名, 1-8 字符,  
(缺省时, 用 UNKNOWN 作字体名)

x = 扩展名, 3 字符 ( 固定的, 始终是: GRF)

t = 图象总的字节数

w = 每行字节数

DATA = ASCII 十六进制串图象定义

如目标名省略, 就用 UNKNOWN.GRF 作为字图象名。数据串使用 ASCII 十六进制串图象定义, 每个字符表示水平方向的四个点。

以下是一个用~DG 指令加载图象到 DRAM 的例子。贮存图象名叫 SAMPLE.GRF。

~DGR: SAMPLE.GRF, 00080, 010,

FFFFFFFFFFFFFFFFFFFF

8000FFFF0000FFFF0001

8000FFFF0000FFFF0001

8000FFFF0000FFFF0001

FFFF0000FFFF0000FFFF

FFFF0000FFFF0000FFFF

FFFF0000FFFF0000FFFF

FFFFFFFFFFFFFFFFFFFF

参数 t (图形总字节数) 用以下公式计算:

$$X(\text{毫米}) \times \text{打印机分辨率}(\text{点/毫米}) \times Y(\text{毫米}) \times \text{打印机分辨}(\text{点/毫米}) / 8(\text{点/字节}) = \text{总字节}$$

X 是单位毫米的图象宽度。Y 是单位毫米的图象高度。点/毫米打印机编程的打印分辨率。

例如, 确定图象 8 毫米宽, 16 毫米高, 打印分辨率 8 点/毫米的正确 t 参数其公式是:

$$8 \times 8 \times 16 \times 8 / 8 = 1024 \text{ 字节}$$

参数 w（每行字节数）用以下公式计算：

$$X(\text{毫米}) \times \text{打印机分辨率}(\text{点/毫米}) / 8(\text{点/字节}) = \text{每行字节数}$$

x 是单位毫米的图象宽充，点/毫米是打印机偏移打印分辨率。

例如，确定图象 8 毫米宽，打印分辨率 8 点/毫米的正确 w 参数，其公式是：

$$8 \times 8 / 8 = 8 \text{ 字节}$$

注意：

1. 所有字节中一行的字节
2. w 是 t 参数计算的第一个值

参数<DATA>是一串十六进制数作为图象表示送打印机。每一十六进制字符代表水平方向四个点。如图象前四个点是白的，后四个点是黑的。二进制码的点 00001111。十六进制表示二进制值将是 0F。完整的图象码就是这样。完整图象被送打印机是一长连续十六制值。

对~DG 和~DB 指令选择数据压缩方案。

斑马打印机有一数魏数据压缩方案。在~DG 和~DB 指令中这方案减少下载图象和点阵字体实际数据字节数和总的时间。

下面数 1, 2, 3, 4, 5, ……19 代表在后面十六进制值（注：值从 G 开始 0 到 9 和 A 到 F 已经在十六进制值是使用）

G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

例如，送 M6 到打印机表示如下十六进制数据。

6666666

M 的值是 7，所以传送 7 个十六进制的 6

g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400

例如：送~hB 到打印机表示送十六进制数据

BB  
40

h 值是 40，所以传送 40 个十六进制的 B。

## 重复值

几个重复值一起使用得到任何需要值，“vMB”和“MvB”一样，都是传送 327 个十六进制的 B 到打印机。

把逗号（，）填进行，表示该行右边，都是 0，

感叹号（！）填进行，表示该行右边都是 1，

冒号（：）表示重复上一行。

解码和参数 t 字节数打印后，打印机返回原来的打印模式。

## 减少下载图象时间

当用~DG 指令时，这方法减少实际送打印机的数据字节数。

如果十六进制串都是到零（0）结束。一个逗号（，）可以替代所有的零。如果十六进制串是奇数个 0 结束，需加一个零和逗号。感叹号（！）和冒号（：）在用法如前。

**警告：在你编辑器文本行将不同于 ZPLII 点的行。编辑器将字把点行包裹起来。ZPLII 跳过文本行结束（如：回车和换行字符）。**

## ~DN 取消下载图象

^DG 在解码和打印完参数 t 的字节后，打印机返回一般模式。用~DN（取消下载图象）指令。退出图像模式，恢复一般模式。

**~DN 指令格式**

~DN

**注意：**任何^或~指令将结束下载。

## ~DU 下载 TrueType 字体

有一些国际字体超过了 256 个可打印字符，这些字体是用~DU 来下载到打印机，作为 TrueType 字体来使用的。

字段块（^FB）命令不支持大型 TrueType 字体。

### ~DU 命令的格式

~DUf, s, DATA

d = 贮存图象的目标设备。

缺省值：R: (DRAM)

其它值：E:

s = 字体长度的字节数，

默认值： 没有 = 如果没有，本命令将被忽略

其他值：输入字体的斑马可下载格式所需的内存的字节数。

DATA = ASCII 十六进制字串的字体定义。

默认值： 没有，本命令将被忽略。

其他值：用 ASCII 表示的十六进制数。（2 位字符/字节）

必须匹配参数 S 所给的值，否则，本命令将被忽略。

例子：

~DTR:KANJI, 86753, 60c0ca017b0ce7..... (86753 个 2 位的 HEX 值)



## ~EG 从存储中删除所有图象

~EG（删除下载图象）指令是用于从存储中删除所有的图象（标签格式图象和十六进制图象）

~EG 指令格式是

~EG

## ^FB 字段块

^FB（段块）指令允许你在一个定义格式中打印文本。这指令格式^FD 文本串放在一指定带原点，字体，和旋转文本块中。这指令也包含一个自动的字回绕功能。

### ^FB 指令格式

^FB a, b, c, d, e

a = 文本块行点的宽度。

有效 最小 = 一个字符的宽度。  
数据： 最大 = 标签宽度。

缺省值：=0

许可值：0~9999

**注：值如小于字体宽度或不对，文本块将不打印出来。**

b = 文本块最大行数

缺省值：1 行

许可值：1 到 9999

**注：文本最大行数将复盖上次行数。改变字体尺寸将自动增加或减少块尺寸。**

c = 增加或删除行间点的空区

缺省值：=0

许可值：-9999 到+9999

**注：如前面没有负号，数字认为是正数。正值加空区；负值删空区。**

d = 块内文本对齐

缺省值 = L（左）

许可值：L（左），C（中心），J（边到边）和 R（右）

**注：如用“J”，最后一行为左对齐。**

e = 第二左边。点的宽度，第二行和以后所有有行文本块将被缩进。

许可值：0-9999

注意^FB 指令的一些注意事项

以下组合有特殊意义：

“\&” = 回车 / 换行

“\(\*)” = 软连字号（单词断开时使用），如果连字符放在一行的最后，它会被打印出来，  
如果它不是在在一行的最后，将会被忽略。（在^B7 中忽略）

“\\” = 字符\，（注意必须选择^CI13，使它能打印 \）

第一条：要打印一个\必须使用^CI13 进行国家代码选择。

第二条：如软连字符接近一行的最后，该连字符将被打印。如不放在最后一行附近，它将被跳过。

(\*) 表示任何字母/数字字符。

- 如果一个字在一行太长（没有软连字符），一个连字符将自动放在块的左字上。剩下字将放在下行。（连字符的位置取决于字长度而不是音节边界，放置一连字符表示连字符就会出现该单元）
- 最大数据串长度是 3K 包括控制字符和回车/换行
- 通常回车/换行和词间隙在行断时被放弃

## THERMAL BARCODE PRINTER SERIES

- 当用`^FT`（段排版）——`^FT`是用文本上行基准线增加字体尺寸将使文本块从底到顶尺寸增加（可使打印超过标签顶）。
- 当用`^FO`（字段原点）增加字体大小将使文本块从顶到底尺寸增加
- 如果`^SN`替代`^FD`，字段将不打印。
- `^FS`结束`^FB`语句，每个块要求有自己的`^FB`指令

## ^FD 字段数据

^FD（字段数据）指令定义字段的数据串。字段数据可以任何可打印字符除了那些指令前缀（如^和~）。

^FD 指令格式

^FDa

a = 打印数据

注意：

1. 字段数据串最多 3072 字符
2. ^和~字符可以改变前缀字符而打印出来。见 CC 和 CT 指令（注：新的前缀字符不能被打印）
3. 字符 ASCII 码超过 127 时或者用^FH 和^FD 打印^和~字符。

用以下办法完成指定的功能。

“\&” = 回车 / 换行

“\(\*)” = 软连字号（单词断开时使用），如果连字符放在一行的最后，它会被打印出来，如果它不是在在一行的最后，将会被忽略。（在^B7 中忽略）

“\\” = 字符\，（注意必须选择^CI13，使它能打印 \）

(\*) = 任意字母 / 数字

注意：

1. 要打印 \, 必须选择^CI13
2. 如果软连字符置于一行的最后，连字符会被打印出来，但如果没有放在一行的最后，它将会被忽略。

## ^FH 十六进制转义字符

^FH（十六进制段）指令允许你在^FD 语句里直接输入任何十六进制值。^FH 指令必须在每个^FD 指令前才能用。

在^FD 语句里，十六进制标识必须在每个十六进制值前。缺省的十六进制标识是下划线（\_）。

这指令可以用在任何有数据段的指令里，（如^FD，^FV（段变量），和^SN（串行数据））。

有效的十六进制字符是

0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f

### ^FH 指令格式

^FH a

a = 十六进制转义符

开机初始化值 = \_（下划线）

其他许可值：除了当前格式前缀或控制前缀的任何字符。

例：

^F0100, 100^AD^FH^FD~\_7e Used for HEX^FX

^F0100, 100^AD^FH/^FD~\_7e Used for HEX^FX

以上二行将产生以下结果

Tilde ~ used for HEX

## ^FN 字段编号指令

^FN（数字段）指令用于数据段数字。这指令用于贮存格式和调用格式操作。

在贮存格式，^FN 指令用在正常的^FD（数据段）指令的地方。在调用贮存格式时，用^FN 结合^FD（数据段）指令。

### ^FN 指令格式

^FNx

a = 字段所赋的编号值

缺省值：0

其它值：最小=1，最大=9999

注意：

1. 同一^FNx 值可被贮存在几个不同段里。
2. 如一个标签格式包含一个^FNx 和^FD，段的数据将用任何其它同^FN 的值打印。

## ^FO 字段位置

^FO 命令设置字段的位置，相对于由^LH 命令指定的标签原点。^FO 命令设置字段的左上角的位置。

^FO 命令格式：

^FOx, y

x = X 轴的点数

缺省值：0

许可值：最小=1，最大=9999

y= Y 轴的点数

缺省值：0

许可值：最小=1，最大=9999

注意：如果 x, y 的值太大，它将置打印位置于标签之外。

## ^FP 字段参数

^FP（字段参数）命令使字体格式可以纵向或者反相排列。比较常用于亚洲字体。

^FP 指令格式

^FPd,g

d = 方向

H = 水平打印

V = 垂直打印

R = 反相打印（从右向左）

g = 另加的字间距

范围：0—9999

**注意：**使用反相打印时，^FT 所指定的位置是最右边的字符的左下角。

示例：

^XA^FPV,10^AG^FDtest^XZ

^XA^FPR,10^AG^FDtest^XZ



## ^FR 字段反相打印

^FR（颠倒打印段）指令允许出现的段白变黑或黑变白。打印字段时，如打印点是黑则变白，如点是白则变黑。

^FR 指令格式

^FR

以下是怎样使用 FR 指令的例子。

^XA

^F0100, 60^GB100, 203, 203^FS

^F0180, 100

^CFG^FR^FDFIELD^FS

^F0130, 170

^FR^FDREVERSE^FS

^XZ

注意：

1. 这些字段位图是放在格式指令指定段点阵里，在标签格式中用^FR 指令超过一次时须当心。
2. ^FR 指令的结果 不会在另一个字段出现（如，文本后跟^FR^GB）如以上例子。

## ^FS 字段分隔

^FS（段括号）指令表示字段定义结束。段括号指令可以用一个简单 ASCII 控制码 SI（Ctrl—0，十六进制 0F）代替。

^FS 指令格式

^FS

## ^FT 字段排版

^FT(段排版)指令也是设段位置，和标签起始^LH 命令关系。字段位置在旋转时不改变，段排版原点是被固定。

^FT 指令格式

^FT x, y

x = x 方向点数

缺省值：=上次格式段位置

许可值：0~9999

y = y 方向点数

缺省值：=上次文本格式段位置

许可值：0~9999

以下定义^FT 怎样在文本条码，图象，框，镜像时工作。

- 文本——原点是字符串起始，压字体的基准线。一般基准线在大多数字符底部，除了那些下行字符如“g”，“y”等。

**注：当座标缺失时，位置被跟着上次格式字段。这种记忆功能将使字段和其它字段位置简单化。当第一个段被定位，其它段跟着自动定位。**

- 条码——原点是条码的基准点，如有条码下面解释行，或条码有防护条时也是。
- 图框——原点是在框的左下角。
- 映象——原点是在矩形镜象左下角。

重要注意

有几个例子，当用^FT 指令时，最好使用 x 和 y 参数。

1. 标签格式第一个段定位
2. 任何^FN（段数字）指令
3. 顺序^SN（串行数据）指令

## ^FV 可变数据段

^FV（可变数据段）指令替换在标签格式中可变的数据段^FD（数据段）指令：

**^FV 指令格式**

^FV<data>

data = 被打印的可变数据段。0-255 个字符[如无数据输入指令跳过]  
最大字符串长度=3072

以下是怎样用^MC 和^FV 指令的例子。

^XA

^F055,60^FVARIABLE DATA #1 ^FS

VARIABLE DATA #1

F080,150^FDFIXED DATA ^FS

FIXED DATA

^MCN^XZ

^XA

^F055,60^FVARIABLE DATA #2 ^FS

VARIABLE DATA #2

^MCY^XZ

FIXED DATA

**注意：**^FV 段在标签打印后始终被清除

## ^FW 字段方向

^FW 指令设定所有可以设定方向的内容的默认方向和对齐方式。字段可以顺时针旋转 0，90，180，270 度，对齐方式可选左对齐，右对齐和自动对齐。

^FW 指令作用或持续到重启机器或者下一个^FW 指令。

### ^FW 指令格式

^FWr, z

r = 字段方向

默认值：N = 正常

其他值：R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

z = 对齐方式

默认值：^TB 自动，其他指令左对齐

其他值：0 = 左对齐

1 = 右对齐

2 = 自动对齐

下面是一个对齐的例子

^XA

^FWR

^F0150, 90^A0N, 25, 20^FDPRINTER Technologies^FS

^F0115, 75^A0, 25, 20^FD0123456789^FS

^F0150, 115^A0N, 25, 20^FD333 Corporate Woods

Parkway^FS

^F0400, 75^A0, 25, 20^FDXXXXXXXX^FS

^XZ

## ^FX 注解

^FX（注释）指令是当你要在标签格式中加入信息注解时有效的方式。任何^FX 指令后的数据到下一脱字号（^）或波浪号（~）指令间在标签格式中无任何结果。

### ^FX 指令格式

**^FX < data >**

< data > =指令解释

以下怎样用^FX 指令的例子：

^XA

^LH100,100^FS

^FXSHIPPING LABEL^FS

^F010,10^GB470,280,4^FS

^F010,190^GB470,4,4^FS

^F010,80^GB240,2,2^FS

^F0250,10^GB2,100,2^FS

^F0250,110^GB226,2,2^FS

^F0250,60^GB226,2,2^FS

^F0156,190^GB2,95,2^FS

^F0312,190^GB2,95,2^FS

^XZ

注：通常做法是在数据后跟^FS 指令。

## ^GB 画框

^GB 命令用于在标签版面中画线，画矩形框。用于标注重要信息，分隔内容区域或者让版式设计更好看。画线和画框都用的是同一个指令。

^GB 指令格式

^GBw, h, t, c, r

- w = 框宽度(单位：点)  
缺省值：=t 的值或 1  
许可值：t~9999
- h = 框高度(单位：点)  
缺省值：=t 的值或 1  
许可值：t~9999
- t = 边框厚度 (单位：点)  
缺省值：=1  
许可值：1~9999
- c = 框颜色  
缺省值：B = 黑色  
许可值：W = 白色
- r = 边框圆角值  
缺省值：=0  
许可值：0~8

对于 w 和 h 值来说，必须根据机器分辨率来确定值的大小。

如果没有指定高度和宽度值，则指令会生成一个高宽都为 t 的一个实心矩形。

r 值是用于表示每个矩形框的圆角半径，计算公式如下：

$$\text{圆角半径} = (\text{r 值}/8) * (\text{短边}/2)$$

短边表示矩形框设置好或者应用默认值之后长宽较短的那条边的长度。

## ^GC 画圆

^GC 命令用于在标签版面中画圆。命令的参数定义圆的直径，轮廓的粗细和颜色。轮廓粗细是向内延伸的。

^GC 指令格式

^GCd, t, c

d = 圆直径(单位：点)

缺省值：=3

许可值：3~4095

t = 轮廓粗细(单位：点)

缺省值：=1

许可值：2~4095

c = 框颜色

缺省值：B = 黑色

许可值：W = 白色

一个画圆的例子：

^XA

^F050, 50

^GC250, 10, B^FS

^XZ



## ^GD 画斜线

^GD 命令在页面版式中生成一个斜线。这个指令可以综合其他画图指令在版面设计中更加多元化。

^GD 指令格式

^GDw, h, t, c, o

- w = 框宽度(单位: 点)  
缺省值: =t 的值或 3  
许可值: 3~9999
- h = 框高度(单位: 点)  
缺省值: =t 的值或 3  
许可值: 2~9999
- t = 轮廓粗细(单位: 点)  
缺省值: =1  
许可值: 2~4095
- c = 框颜色  
缺省值: B = 黑色  
许可值: W = 白色
- o = 方向  
缺省值: R = 右倾线  
许可值: L = 左倾线

一个斜线的例子:

^XA

^F0150, 100

^GB350, 203, 10^FS

^F0155, 110

^GD330, 183, 10, , R^FS

^XZ

## ^GE 画椭圆

^GE 命令在页面版式中生成一个椭圆。

^GE 指令格式

^GEw, h, t, c

- w = 框宽度(单位: 点)  
缺省值: =t 的值或 3  
许可值: 3~9999
- h = 框高度(单位: 点)  
缺省值: =t 的值或 3  
许可值: 2~9999
- t = 轮廓粗细(单位: 点)  
缺省值: =1  
许可值: 2~4095
- c = 框颜色  
缺省值: B = 黑色  
许可值: W = 白色

一个椭圆的例子:

^XA

^F0100, 100

^GE300, 100, 10, B^FS

^XZ

## ^GF 区域绘图

^GF 命令可以往打印机的图像数据缓冲区直接写入图像数据。这个指令的参数遵循其他指令的同样约定。图像可以放在任何位图区域内。

^GF 指令格式

^GFa, b, c, d, <data>

a = 压缩类型

缺省值：= A ASCII 码表示的十六进制数(与其他下载指令数据一致)

许可值：= B 二进制数据(直接二进制数据)

= C 压缩二进制(数据是主机通过使用斑马压缩算法计算出来的，传入打印机后再解压并放置在数据缓存)

b = 二进制字节总数

缺省值：必须有值，无值指令无效

许可值：1~99999

c = 图像字段总数

缺省值：必须有值，无值指令无效

许可值：1~99999

d = 每行字节数

缺省值：必须有值，无值指令无效

许可值：1~99999

data = 图像数据

许可值：ASCII 码表示的十六进制值

一个字节用两个 ASCII 码表示，可以插入回车使图像数据易读。ASCII 码的对数总和必须等于图像字段总数，如果有多余的数据则忽略。数据里出现逗号表示当前行填充 00(白点)，减少图像数据量。遇到~DN 或者任意一个^号或者~号图像下载过程均退出。

二进制数据下载的时候所有的控制字符均无效直至二进制字节数到达指定的字节总数。

## ^GS 符号图像

^GS 命令可以生成一系列注册商标，版权图像和其他符号。

^GS 指令格式

^GS<sub>o</sub>, h, w

<sub>o</sub> = 符号方向

默认值：N 或上一个 ^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

h = 字符高度

缺省值：= ^CF 设置的值

许可值：0~9999

w = 字符宽度

缺省值：= ^CF 设置的值

许可值：0~9999

一个符号图像的例子：

^XA ^CFD

^F050, 50

^FDPRINTER PROGRAMMING ^FS ^F050, 75

^FDLANGUAGE II (ZPL II ) ^FS ^F0280, 75

^GS ^FDC ^FS

^XZ

# ^HF 返传标签文件

^HF 命令向主机返传存储的文件。

^HF 指令格式

^HFd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 无指定时使用 UNKNOWN

许可值: 1~8 字母数字字符

x = 后缀

固定值: .ZPL

一个返传文件的例子:

首先向打印机写一个简单文件:

^XA

^DFB:FILE1.ZPL

^F0100,100^A0,100

^FDTEST^FS

^XZ

然后用一个终端向打印机发送:

## ^HG 返传图像数据

^HG 命令向主机返传存储的图像数据。

^HG 指令格式

^HGd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 无指定时使用 UNKNOWN

许可值: 1~8 字母数字字符

x = 后缀

固定值: .GRF

## ^HH 返传配置信息

^HH 命令向主机返传机器配置信息。

^HG 指令格式

格式: ^HH

## ^HM 返传 RAM 信息

^HM 命令向主机返传 RAM 信息。

格式：^HM

^HM 返传的数据分三段，用逗号隔开，第一串数表示机器装入的 RAM 总空间，单位是 KB，第二串表示机器开放给用户的最大空间容量，单位是 KB，第三串表示机器现在剩余可用的空间容量，单位是 KB



## ~HS 返传状态信息

~HS 命令向主机返传机器状态信息。信息是三串字符串，字符串的排列是<STX>控制字符起头，<ETX><CR><LF>结束的字符串序列。为了避免混乱，主机上的字符串要分三行显示。

格式：~HS

**注意：**~HS 指令在以下几种情况中不会向主机反馈信息：

- 纸尽
- 碳带用尽
- 打印头打开
- 热敏片过热

**字符串 1：** <STX>aaa, b, c, dddd, eee, f, g, h, iii, j, k, l<ETX><CR><LF>

aaa	= 通信接口设定
b	= 纸尽标志 (1=纸尽)
c	= 暂停标志 (1=暂停)
dddd	= 纸张长度 (点长度)
eee	= 接收缓冲区里的格式数量
f	= 缓冲区标志 (1=接收区满)
g	= 通信诊断模式标志 (1=诊断模式开启)
h	= 局部格式标志 (1=局部格式)
iii	= 不使用 (永远为 000)
j	= 缓存失效标志 (1=配置数据丢失)
k	= 温度范围 (1=正常温度)
l	= 温度范围 (1=过热)

通信接口设定：

<u>aaa</u> = a <sub>8</sub> a <sub>7</sub> a <sub>6</sub> a <sub>5</sub> a <sub>4</sub> a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub>	
<sup>7</sup> a = Handshake 0 = <del>Xon/Xoff</del> 1 = DTR	<sup>0..1</sup> a <sub>8</sub> a <sub>2</sub> a <sub>1</sub> a = Baud  0000 = 110 0001 = 300 0010 = 600 0011 = 1200 0100 = 2400 0101 = 4800 0110 = 9600 0111 = 19200 1000 = 28800 (available only on certain printer models) 1001 = 38400 (available only on certain printer models) 1010 = 57600 (available only on certain printer models) 1011 = 14400
<sup>6..1</sup> a = Parity Odd/Even 0 = Odd 1 = Even	
<sup>5</sup> a = Disable/Enable 0 = Disable 1 = Enable	
<sup>4..1</sup> a = Stop Bits 0 = 2 Bits 1 = 1 Bit	
<sup>3</sup> a = Data Bits 0 = 7 Bits 1 = 8 Bits	

字符串 2: <STX>mmm, n, o, p, q, r, s, t, uuuuuuuu, v, www<ETX><CR><LF>

mmm = 机器设定  
n = 不使用  
o = 打印头打开标志(1=打开)  
p = 碳带用尽标志(1=用尽)  
q = 热转模式标志(1=选择热转模式)  
r = 打印模式  
s = 打印宽度模式  
t = 打印机等待标志(1=在剥离模式中等待状态)  
uuuuuuuu = 未处理标签  
v = 打印标志(一直是 1)  
www = 存储设备中的图像数

机器设定:

<u>mmm</u> = m <sub>7</sub> m <sub>6</sub> m <sub>5</sub> m <sub>4</sub> m <sub>3</sub> m <sub>2</sub> m <sub>1</sub> m <sub>0</sub>	
m <sub>7</sub> = Media Type 0 = Die-Cut 1 = Continuous	m <sub>4</sub> m <sub>3</sub> m <sub>2</sub> m <sub>1</sub> = Unused 0 = Off 1 = On
m <sub>6</sub> = Sensor Profile 0 = Off	m <sub>0</sub> = Print Mode 0 = Direct Thermal 1 = Thermal Transfer
m <sub>5</sub> = Communications Diagnostics 0 = Off 1 = On	

**字串 3:** <STX>xxxx, y<ETX><CR><LF>

xxxx = 密码

y = 0(没有安装静态 RAM)

1(安装了静态 RAM)

## ^ID 删除对象

^ID 命令从存储设备中删除对象，图片，字体和存储的格式。删除对象时可以指定其中一个或者成组删除。

对象的名称和扩展名都可以用\*代替，可以很方便实现成组删除的目的。

^ID 指令格式

^IDd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 无指定时使用 UNKNOWN

许可值: 1~8 字母数字字符

x = 后缀

默认值: .GRF

其他值: 其他 ZPL 语言支持的后缀

几个删除对象的例子:

例一 删除 DRAM 存储的所有后缀名为 ZPL 的文件

^XA

^IDR:\*.ZPL^FS

^XZ

例二 删除名称为 SAMPLE 的文件，不管它的后缀是什么

^XA

^IDR:SAMPLE.\*^FS

^XZ

例三 删除图像 SAMPLE1.GRF 然后存储 SAMPLE2.GRF

^XA

^F025,25^AD,18,10

^FDDelete^FS

^F025,45^AD,18,10

^FDthen Save^FS

^IDR:SAMPLE1.GRF^FS

^ISR:SAMPLE2.GRF^FS

^XZ

注释: RAM 内的格式一旦删除，则释放的空间可以继续使用。其他盘符内格式删除后空间并不能马上使用，除非做了碎片处理或者存储初始化操作才可以。

如果指定的名称在存储内找不到，指令被忽略。

## ^IL 图像加载

^IL 命令一般写在标签的开头，读取已经存入存储中的图像并且和后面的内容结合在一起输出。^IL 输出的图像坐标起始永远是 0,0

用此方式把变量和固定格式结合起来可以提高标签的制作效率。

^IL 指令格式

^ILd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 无指定时使用 UNKNOWN

许可值: 1~8 字母数字字符

x = 后缀

固定值: .GRF .PNG

## ^IM 图像移动

^IM 命令可以在^IL 的基础上直接移动图像位置。此命令和^XG 的功能一致，除了没有放大参数。

^IM 指令格式

^IMd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 无指定时使用 UNKNOWN

许可值: 1~8 字母数字字符

x = 后缀

固定值: .GRF .PNG

例子 下面是一个使用^IM 把 SAMPLE1.GRF 原样放置在标签的几个不同位置输出

^XA

^F0100,100^IMR:SAMPLE.GRF^FS

^F0100,200^IMR:SAMPLE.GRF^FS

^F0100,300^IMR:SAMPLE.GRF^FS

^F0100,400^IMR:SAMPLE.GRF^FS

^F0100,500^IMR:SAMPLE.GRF^FS

^XZ

注释: 使用^FO 指令可以把图像放置在标签的任何位置。^IM 和^XG 的不同点在于^IM 没有放大倍率, 处理起来也快一点。

## ^IS 图像存储

^IS 命令在一个标签格式内使用把当前的标签内容存成图像。一般都会写在一个标签的最后。存储的图像可以给后续命令快速调用并结合变量来设计一个完整的标签版面。

用此方式把变量和固定格式结合起来可以提高标签的制作效率。

^IS 指令格式

```
^ISd:o,x,p
d = 盘符
    默认值: R:
    其他值: E:
o = 文件名
    缺省值: 无指定时使用 UNKNOWN
    许可值: 1~8 字母数字字符
x = 后缀
    固定值: .GRF .PNG
p = 存储后打印
    缺省值: Y = yes
    许可值: N = no
```

例子 下面是一个使用^IS 把样式保存为 SAMPLE2.GRF

```
^XA
^LH10,15^FWN^BY3,3,85^CFD,36 ^GB430,750,4^FS
^F010,170^GB200,144,2^FS
^F010,318^GB410,174,2^FS
^F0212,170^GB206,144,2^FS
^F010,498^GB200,120,2^FSR
^F0212,498^GB209,120,2^FS
^F04,150^GB422,10,10^FS
^F0135,20^A0,70,60
^FDPRINTER^FS
^F080,100^A0,40,30
^FDTECHNOLOGIES CORP^FS
^F015,180 ^CFD,18,10^FS
^FDARTICLE#^FS
^F0218,180
^FDLOCATION^FS
^F015,328
^FDINFORMATION^FS
^F015,508
^FDREQ.NO.^FS
^F0220,508
^FDWORK NUMBER^FS
```

## THERMAL BARCODE PRINTER SERIES

^F015,630^AD,36,20

^FDCOMMENTS^FS

^ISR:SAMPLE2.GRF,Y

^XZ



## ^JB 初始化闪存

^JB 命令初始化打印机内的闪存。

^JB 指令格式

^JBa

a = 盘符

默认值：盘符必须指定

其他值：E:

**注意：**初始化闪存操作可能持续几分钟，必须等待足够长的时间让打印机初始化闪存，再去做其他操作。

## ~JD 开启通信诊断功能

~JD 命令启动诊断模式。诊断模式下所有接收到的字符都不会作处理而直接打印在纸上。打印内容包括了 ASCII 码，十六进制值和通信错误。

指令格式 ~JD

## ~JE 关闭通信诊断功能

~JE 命令关闭诊断模式并使打印机回到正常打印模式。

指令格式 ~JE

## ^JM 设置分辨率

^JM 命令通过成倍放大点数来降低分辨率。^JM 会影响^F0 坐标放置在标签上的距离。

发送^JM 指令会把标签的大小放大一倍。

^JM 命令必须放置在一个^FS 前，后续才会生效。

^JM 指令格式

^JMn

n = 设置分辨率

默认值：A = 默认分辨率

其他值：B = 增大一倍

例子：

原始分辨率

^XA

^JMA^FS

^F0100,100

^B2N,50,Y,N,N

^FD1234567890^FS

^XZ

增大分辨率

^XA

^JMB^FS

^F0100,100

^B2N,50,Y,N,N

^FD1234567890^FS

^XZ

## ^JU 配置更新

^JU 命令设置打印机的配置参数。

^JU 指令格式

^JUa

a = 配置参数

默认值：必须设置一个值

F = 恢复出厂默认

N = 恢复出厂默认(网络参数)

这些值如果不设置^JUS 则会在掉电后消失

其他值：

R = 调用保存配置参数

S = 保存当前配置参数

这些值在开机时设置

# ^LH 坐标原点

^LH 命令指定标签内容的坐标原点。

默认的标签坐标原点均在标签内容的左上角。^LH 定义的是标签的坐标参考点，任何在这个坐标右侧和下方的区域是可打印区域。举个例子，如果使用的标签是有预先印制内容，可以使用这个指令把坐标的起始点放在印制内容以外，这样打印的时候就不会跑到印制内容上了。

这个指令影响所有在其后的指令，所以必须放在标签的最开始。

^LH 指令格式

^LHx, y

x = X 坐标(单位点)

默认值：0 或者上次设置的值

其他值：0 到 9999

y = Y 坐标(单位点)

默认值：0 或者上次设置的值

其他值：0 到 9999

注释：如果要适应当前标签，则^LH 后必须有一个^FS 结束符。一旦定义了^LH 后，设置一直保留到重启机器或者下一个^LH 的定义。

## ^LL 标签长度

^LL 命令定义标签的长度。这个指令在打印连续纸时是必须定义的。

如果要适应当前标签，则^LL 后必须有一个^FS 结束符。一旦定义了^LL 后，设置一直保留到重启机器或者下一个^LL 的定义。

^LL 指令格式

^LLy

y = 长度(单位点)

默认值：不要大于打印机能够支持的最大值

其他值：1 到 9999

## ^LR 反相打印标签

^LR（反相打印标签）指令在标签格式中印有打印段的内容黑白反色。它允许一个段由白变黑或由黑变白。当打印一个段，如果打印点是黑的，它变白；如果点是白的，它变黑。

用^LR 是等同于用^FR。

^LR 指令格式

^LRa

a = 反相打印

缺省值：N=不反相打印标签

其他值：Y=是

开机初始值=N

（如无参数指令跳过）

以下是怎样用^LR 指令的例子。

注意

1. ^LR 将保留到^LRN 指令转换或打印机关机
2. ^LR 指令必须跟^GB 一起使用。
3. 仅仅在这指令后的段被影响。



## ^LS 标签左右偏移

^LS 指令可以使标签内容根据需要左右偏移。

^LS 指令格式

^LSa

a = 左移值

缺省值：0

其他值：-9999 到 9999

注释：如果要适应当前标签，则^LS 后必须有一个^FS 结束符。一旦定义了^LS 后，设置一直保留到重启机器或者下一个^LS 的定义。

## ^LT 标签上下偏移

^LT 指令可以使标签内容根据自己需要上下偏移。负值把内容向标签上沿移动，正值把内容远离上沿方向移动。

^LT 指令格式

^LTx

x = 下移值

缺省值：必须指定值

其他值：-120 到 120

## ^MC 清除上张标签内容

正常打印过程中，图像打印后都会清除。^MC 指令可以保存当前图像供当前标签或者后续标签快速重用直到用^MCY 清除图像

^MC 指令格式

^MCa

a = 是否清除

缺省值：Y(清除格式)

其他值：N(不清除格式)

注释：^MC 保持当前编辑的图像格式，会出现在后续的图像的背景上。

## ^MD 打印相对浓度

^MD 是相对调整当前的浓度等级。

^MD 指令格式

^MDa

a = 相对浓度等级

缺省值：0

如果没有设置值指令忽略

其他值：-30 到 30

注释：^MD 是设置相对当前设置的绝对浓度增加或减少等级。计算公式是绝对浓度+相对浓度=浓度。如果数字小于 0 或者大于 30，则浓度回绕，从头或者从尾再计算。如当前 15，^MD-17，则浓度为 28

## ^MM 打印模式

^MM 指令决定打印机在打印一批标签后的操作。

^MM 指令格式

^MMa

a = 选择模式

缺省值：T(撕纸模式)

其他值：P, R, A, C, D, F, L, U, K, V, S(无动作模式)

注释：^MM 调整的打印模式仅有两种，除撕纸外其余全部都为无操作。

## ^MN 纸张类型

^MN 指令决定打印机使用何种纸张打印并且设置黑标的打印起始偏移。

^MN 指令格式

^MN<sub>a, b</sub>

a = 使用的纸张类型

缺省值：必须设置值否则指令无效

其他值：N = 连续纸张

Y = 非连续间隙纸张

W = 非连续间隙纸张

M = 非连续黑标纸张

A = 自动选择

b = 黑标起始偏移

缺省值：0(在 a=M 时才有效)

其他值：-120 到 283

## ^MT 介质类型

^MT 指令选择使用的介质类型。

^MT 指令格式

^MTa

a = 类型

缺省值：必须设置值否则指令无效

其他值：D = 热敏模式

T = 热转模式

# ^MU 设置长度单位

^MU 指令设置打印机使用何种度量单位。^MU 在使用时后续所有数据的坐标和长度定义均使用^MU 设置的单位直到设置了新的单位。

^MU 指令格式

^MUa

a = 单位

缺省值: D = 点

其他值: I = 英寸

M = 毫米

这是一个设置长度单位的例子:

用在分辨率为 203dpi 的打印机上

使用点为单位:

^MUD^F0100,100^GB1024,128,128^FS

使用毫米为单位:

^MUM^F012.5,12.5^GB128,16,16^FS

使用英寸为单位:

^MUI^F0.493,.493^GB5.044,.631,.631^FS



## **^PH ~PH 跳至原点位置**

^PH ~PH 指令让打印机走一张设定长度的标签。

指令会在完成打印当前的标签后或者打印机处于暂停状态后走一张空白标签。

指令格式 ^PH ~PH

## ^PM 打印镜像标签

^PM 指令将整体的标签内容镜像打印出来。指令将图像左右颠倒过来。

^PM 指令格式

^PMa

a = 镜像变化

缺省值：N = 不变化

其他值：Y = 变化

这是一个打印镜像标签的例子：

^XA^PMY

^F0100,100

^CFG

^FDMIRROR^FS

^F0100,160

^FDIMAGE^FS

^XZ

# ^PO 打印方向

^PO 指令将整体的标签内容转过 180 度。如果标签格式中包含了^LL, ^LS, ^LT 等指令，输出内容就会有不同。

^PO 指令格式

^POa

a = 转 180 度

缺省值：N = 不旋转

其他值：Y = 旋转

这是一个打印方向旋转的例子：

^XA^CFD

^POI

^LH330, 10

^F050, 50

^FDPRINTER TECHNOLOGIES^FS ^F050, 75

^FDVernon Hills, IL^FS

^XZ

指令旋转坐标系，所有对象的放置都依照新的坐标系。^LH 指令可以使图像回到标签内。

注释：如果使用了多个^PO 指令，则最后一个^PO 生效。一旦设定了打印方向，则持续到下一个打印方向设定或者关机重开。

## ^PQ 打印数量

^PQ 指令控制批量打印中的一些操作。它可以控制标签的数量，多少张后暂停，同一张有多少份拷贝。

^PQ 指令格式

^PQq, p, r, o

q = 打印总数

缺省值：1

其他值：1 到 99999999

p = 多少张后暂停

缺省值：0(不暂停)

其他值：1 到 99999999

r = 每张拷贝数

缺省值：0(不复制)

其他值：0 到 99999999

o = 暂停

缺省值：N = 不暂停

其他值：Y = 暂停

如果参数设定为 Y，打印机打印不会有暂停操作，如果设定为 N，打印机每打印一组标签就会暂停，直到用户按下 FEED。

## ^PR 打印速度

^PR 指令决定打印速度。

^PR 指令格式

^PRp

p = 打印速度

缺省值：3

其他值：2 到 5，A 到 E

A 对应的速度为 2，B 对应为 3，C 对应为 4，D 和 E 均是对应 4。

## ~PS 打印重新开始

~PS 指令让进入暂停模式的打印机恢复打印过程。这个功能跟在打印机上按 PAUSE 键动作是一致的。  
指令格式 ~PS

## ^PW 打印宽度

^PW 指令可以定义打印宽度。

^PW 指令格式

^PWa

a = 打印宽度

缺省值：上次设置的值

其他值：2 到标签的最大宽度

## ^RF 读写 RFID 标签内容

^RF 指令可以读取和写入 RFID 标签内容

^RF 指令格式

^RFo,f,b,n,m

o = 执行操作

缺省值: W = 写入标签内容

其他值: L = 写入并锁定标签内容 (Gen2 标签类型不使用此锁定功能)

R = 读取标签

P = 读取密码 (仅限 Gen2 标签类型)

f = 格式

缺省值: H = Hexadecimal

A = ASCII

b = 起始块编号

缺省值: 0

其他值: 0-n, n 为标签最大区块数

n = 读取或写入的数据长度 (bytes)

缺省值: 1

其他值: 1-n, n 为标签最大字节数

m = 内存分段

缺省值: E = EPC 96 位

其他值: 0 = 保留

1 = EPC

2 = 标签标识 (TID)

3 = 用户



## **^RR 重试次数**

^RR 指令可以设定读写 RFID 标签时的重试次数

^RR 指令格式

^RRn

n = 重试次数

缺省值：6

其他值：2-10

## ^RS 设置 RFID 参数

^RS 指令可以设定 RFID 的参数，包括标签类型、读写区块位置以及错误处理方式

^RS 指令格式

^RS,p,v,n,e

- p = 标签读写位置  
缺省值：标签高度减 1 mm  
其他值：0-标签高度(dots)
- v = 无效打印输出长度  
缺省值：标签高度  
其他值：0-标签高度(dots)
- n = 读取 / 编码失败时尝试的标签数量  
缺省值：3  
其他值：1-10
- e = 错误处理  
默认值：N = 不处理  
其他值：P = 将打印机置于暂停状态  
E = 将打印机置于错误状态

## ^RZ 设置 RFID 标签密码及锁定标签

^RZ 指令可以用于设置或删除 RFID 标签的密码及锁定标签

^RZ 指令格式

^RZp, m, l

p = 密码

Gen 2 以外的标签类型

默认值: 00

其他值: 00-FF(hex)

Gen2 标签类型

默认值: 无

其他值: 00000000-FFFFFFFF(hex)

m = 内存分段 (仅对 Gen2 类型标签生效)

默认值: 无

其他值:

K = 破解密码

A = 访问密码

E = EPC

T = 标签标识 (TID)

U = 用户

l = 锁定样式 (仅对 Gen2 类型标签生效)

默认值: 无

其他值:

U = 未锁定

L = 已锁定

O = 永久解锁

P = 永久锁定

W = 写入参数 (仅用于破解密码内存分段)

## ~SD 打印绝对浓度

~SD 指令设定打印的浓度值。

~SD 指令格式

~SD##

## = 浓度(两位数字)

缺省值：上次设置的值

其他值：00 到 30

请参照~MD

## ^SE 选择编码表

^SE 指令选择解析使用的编码表。

^SE 指令格式

^SEd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 必须指定值

许可值: 1~8 字母数字字符

x = 后缀

固定值: .DAT

选择编码表最好就把所有的存储文件信息打印出来再选择, 因为名称必须指定, 不指定则忽略指令。

## ^SF 序列字段

^SF 指令能够序列化一个标准的^FD 定义的字符串。最大能够容纳格式化字符串和增量字符串达 3K 个。

字符串序列化均从最后一位数字或字符开始。如果字符串是固定的不参与增量的内容，需要在增量字符串里填上%

^SF 指令格式

^SFa, b

a = 格式化字符串

格式化字符串确定序列方案。格式化字符串的字符数等于需要格式化的^FD 字段字符数。计数从最右开始

格式化字符串代码：

D 或 d - 十进制数 0-9

H 或 h - 十六进制数 0-9 和 A-F 或 a-f

O 或 o - 八进制数 0-7

A 或 a - 英文字母 A-Z 或 a-z

N 或 n - 字母加数字 0-9 和 A-Z 或 a-z

% - 无用字符或跳过当前字符

b = 增量字符

增量字符是每张标签增加的量。默认值是十进制的 1。这个字符可以是任何可以定义为序列字段的字符，非法字符会被默认为输入 0 这个值。

例一 这是一个生成三张变化序列的标签内容：

^XA

^F0100, 100

^CF0, 100

^FD12A^SFnnA, F^FS

^PQ3

^XZ

类似的例子：

^FDBL0000^SFAAdddd, 1

这个指令的序列打印结果如下：

BL0000, BL0001, ... BL0009, BL0010, ...

BL0099, BL0100, ... BL9999, BM0000...

符合每一位的进位规则的时候均向前进位

^FDBL00 - 0^SFAAdd%d, 1%1

结果如下：

BL00 - 0, BL01 - 1, BL02 - 2, ... BL09 - 9,

BL11 - 0, BL12 - 1...

-作为无用字符会跳过进位，再往前作进位

例二 这个例子中每两个标签会重复 H 和 Z

## THERMAL BARCODE PRINTER SERIES

^XA

^F0100,50^A0N,50,50^FDzzZ^SFnnN,I^FS ^PQ10

^XZ

例三 这个字符串增加一个小写字母 i，但是 zzZ 不能够再往前进位了。

^XA

^F0100,50^A0N,50,50^FDzzZ^SFnnN,i^FS ^PQ10

^XZ

## ^SN 序列数据

^SN 指令能够让打印机用增量或者减量序列化一个数据字段。让打印机在打印每张标签的时候都增加值或减少值。可以用于给定格式和条码字段区域。最大可以序列化最右开始计数的 12 位数据。

^SN 指令格式

^SNv, n, z

v = 起始值

默认值: 1

其他值: 最多 12 位序列数据

n = 增量或减量值

默认值: 1

其他值: 最多 12 位序列数据, 如果是减量值在值前加负号

z = 数据前面加 0

默认值: N

其他值: Y

一个指定了增量的打印例子:

^XA

^F0260, 110

^CFG

^SN001, 1, Y^FS

^PQ3

^XZ



## ~TA 撕离位置调整

~TA 指令让用户自行调整打印耗材打印完成后的停止位置，方便用户撕开或者切断。

~TA 指令格式

~TA###

### = 停止位置(一定要三个字符)

默认值：上次设置的值

其他值：-120 到 120

如果没有参数或者参数有误，指令被忽略

## ^TB 文本框

^TB 指令定义一个有长宽的文本框。文本框可以自动换行，如果文本超出了定义的高度，文本会被截掉。这个指令

注意：^TB 会优先处理后续的文本内容，而不是^FB

^TB 指令格式

^TBa, b, c

a = 块方向

默认值：^FW 默认值或上一个^FW 的值。

其他值：N = 正常 (Normal)

R = 顺时针旋转 90 度 (Roated)

I = 顺时针旋转 180 度 (Inverted)

B = 顺时针旋转 270 度 (Bottom)

b = 块宽度(点)

默认值：1

其他值：1 到块宽度

c = 块高度(点)

默认值：1

其他值：1 到块高度

## ^TO 转移对象

^TO 指令用于复制对象或一组对象到另一个存储区，这个跟电脑系统的拷贝有点类似。

^TO 指令格式

^TOs:o. x, d:o. x

s = 源盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 如果没有指定值, 则使用\*通配符

许可值: 存在的名称

x = 后缀

缺省值: 如果没有指定值, 则使用\*通配符

许可值: 存在的名称

d = 目标盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 如果没有指定值, 则使用\*通配符

许可值: 存在的名称

x = 后缀

缺省值: 如果没有指定值, 则使用\*通配符

许可值: 存在的名称

## ~WC 打印配置标签

~WC 指令用于打印一张机器配置的标签。机器的配置信息包括了纸张设置，版本信息，接口信息，控制符信息和盘符内容等。

~WC 指令格式

~WC

使用此指令的时候打印机必须为空闲状态。

# ^WD 打印目录标签

^WD 指令用于打印

^WD 指令格式

^WDd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: \*, 也可以使用?来表示不确定字符

许可值: 1-8 位字母数字字符

x = 后缀

缺省值: \*, 也可以使用?来表示不确定字符

许可值: ZPL 许可的后缀名

例一 打印 DRAM 中的所有文件

^XA

^WDR:\*. \*

^XZ

例二 打印所有条码文件

^XA

^WDZ:\*. BAR

^XZ

例三 打印所有内置字体文件

^XA

^WDZ:\*. FNT

^XZ

## **^XA 标签开始**

`^XA` 指令用于打印标签起始

`^XA` 指令格式

`^XA`

注释 有效的 ZPL 格式中标签必须以此指令作为起始，`^XZ` 作为结束。

## ^XF 调用标签

^XF 指令用于调用

^XF 指令格式

^XFd:o,x

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 如果没有指定值, 使用 UNKNOWN

许可值: 1-8 位字母数字字符

x = 后缀

固定值: .ZPL

可以参照^DF 指令内容

## ^XG 调用图像

^XG 指令用于调用一个或多个存储的图像并输出。这个指令用于在一张标签内合并图像内容，比如公司标志，零散图像和字段数据共同完成一个完整的标签。

^XG 指令格式

^XGd:o, x, mx, my

d = 盘符

默认值: R:

其他值: E:

o = 文件名

缺省值: 如果没有指定值, 使用 UNKNOWN

许可值: 1-8 位字母数字字符

x = 后缀

固定值: .GRF

mx = x 轴放大因子

默认值: 1

许可值: 1 到 10

my = y 轴放大因子

默认值: 1

许可值: 1 到 10

例子 这是一个使用^XG 调用 R 盘里存有的 SAMPLE.GRF 打印在不同位置不同大小的例子

^XA

^F0100,100^XGR:SAMPLE.GRF,1,1^FS

^F0100,200^XGR:SAMPLE.GRF,2,2^FS

^F0100,300^XGR:SAMPLE.GRF,3,3^FS

^F0100,400^XGR:SAMPLE.GRF,4,4^FS

^F0100,500^XGR:SAMPLE.GRF,5,5^FS

^XZ



## **^XZ 标签结束**

^XZ 指令用于标签内容的结束。一旦打印机接收到此指令，打印一张标签。此指令同样可下发一个 ASCII 控制符 ETX (0x03)。

^XZ 指令格式

^XZ

注释 有效的 ZPL 格式中标签必须以 ^XA 作为起始，此指令作为结束。