

CPCL 编程手册

Ver 1.01

厦门信远通科技有限公司

Xiamen XinYuanTong Technology Co., Ltd

地址：厦门市湖里区梧桐西路 1 号 411 室

电话：13063031816

邮箱：xyt.xm@aliyun.com

网址：<http://xmxyt.tech>

- CPCL编程手册
 - 前言
 - 标签打印命令
 - 标签起始命令
 - PRINT 命令
 - END 命令
 - ABORT 命令
 - ENCODING 命令
 - FORM命令
 - JOURNAL命令
 - UNITS 命令
 - 使用注释
 - 文本
 - TEXT 命令
 - 使用字体组
 - FONT-GROUP (FG) 命令
 - TEXT 串联命令 (CONCAT 和 VCONCAT)
 - MULTILINE (ML) 命令
 - COUNT命令
 - SETMAG 命令
 - 一维条码简介
 - UPC和EAN/JAN条码
 - UPC/EAN 规范
 - Code 39或Code 3 of 9条码
 - Code 39 (3 of 9)规范
 - Code93 或 Code 9 of 3 条码
 - Code93 规范
 - Interleaved 2 of 5 条码
 - Interleaved 2 of 5 规范
 - Code128 和 UCC-128 货运标准
 - Code 128/UCC-128 规范
 - Codabar 条码
 - Codabar 规范
 - MSI Plessey条码
 - MSI Plessey 规范
 - Postnet和信封识别标记
 - Postnet 和 FIM 规范
 - 一维条码命令

- [BARCODE命令](#)
 - [标准条码](#)
 - [表B1D-1](#)
 - [表B1D-2](#)
- [BARCODE-TEXT命令](#)
- [COUNT命令](#)
- [RSS符号](#)
 - [RSS Limited](#)
 - [RSS-14](#)
 - [RSS Expanded](#)
 - [RSS Stacked](#)
 - [RSS-Truncated](#)
 - [RSS-14 Stacked Omnidirectional](#)
 - [RSS/复合符号](#)
 - [Composite Code A atop RSS Limited:](#)
 - [Composite Code B atop Code 128](#)
 - [Composite Code-C atop Code 128](#)
- [RSS/复合 命令](#)
 - [表rss-1](#)
- [二维条码简介](#)
 - [PDF417码](#)
 - [QR 码](#)
- [二维条码命令](#)
 - [PDF417](#)
 - [QR Code](#)
- [图形命令](#)
 - [BOX 命令](#)
 - [LINE命令](#)
 - [INVERSE-LINE命令](#)
 - [PATTERN命令](#)
 - [位图命令](#)
 - [PCX命令](#)
- [设置命令](#)
 - [CONTRAST命令](#)
 - [TONE命令](#)
 - [JUSTIFICATION命令](#)
 - [PAGE-WIDTH命令](#)
 - [PACE命令](#)

- AUTO-PACE命令
- NO-PACE命令
- WAIT命令
- REWIND命令
- TENSION命令
- SPEED命令
- SETSP命令
- UNDERLINE命令
- ON-OUT-OF-PAPER命令
- ON-FEED命令
- PREFEED命令
- POSTFEED命令
- PRESENT-AT命令
- COUNTRY/CODEPAGE 命令
- 文件命令
 - 使用格式文件
- 机器控制命令
 - BEEP 命令
 - CUT 命令
 - PARTIAL-CUT 命令
 - CUT-AT 命令
- 行式打印模式
 - 简介
 - BEGIN-PAGE 命令
 - END-PAGE 命令
 - 使用实用工具函数的特殊命令
 - LP-ORIENT 命令
 - UNITS 命令
 - SETLP 命令
 - SETLF 命令
 - 使用X和Y坐标移动位置
 - LMARGIN 命令
 - SETBOLD 命令
 - SETSP 命令
 - PAGE-WIDTH / PAGE-HEIGHT 命令
 - 特殊的ASCII字符
 - SETFF 命令
 - SET-TOF 命令

- 撕纸或切纸
 - PRESENT-AT 命令
 - CUT-AT 命令
 - CUT 和 PARTIAL-CUT 命令
- 条码、图形和线条
- SETLP-TIMEOUT 命令
- 定义一张票据

前言

本手册详细介绍了CPCL语言中的各种命令，通过这些命令，编程人员可以利用打印机中的内置文本、图形、条码打印和通信等功能。文中通篇使用了一下符号约定：

{ }	必选项
[]	可选项
()	缩写命令
{ }	文字项

空格字符用于分隔命令行中的各个字段。

标签打印命令

标签起始命令

格式:

```
<!> {offset} <200> <200> {height} {qty}
```

说明:

<!>: 使用“!”作为控制会话的起始字符。
{offset}: 整个标签的横向偏移。此值可以使所有作用域以指定的单位数量进行横向偏移。
<200>: 横向分辨率（以点/英寸为单位）。
<200>: 纵向分辨率（以点/英寸为单位）。
{height}: 标签的最大高度。

标签最大高度的计算方法：先测出从第一个标签间隙（或黑标）底部到下一个标签间隙（或黑标）顶部之间的距离，然后从中间隙（或黑标）高度所得结果即最大高度。

{qty}: 要打印的标签数量。最大值=1024。

示例:

```
! 0 200 200 210 1  
TEXT 4 0 30 40 Hello World  
FORM  
PRINT
```

打印结果:



PRINT 命令

结束并打印一个标签文件。除行打印模式外，PRINT指令必须是最后一条指令。执行PRINT指令后，打印机退出控制会话。

格式:

```
{command}
```

说明:

```
{command}: PRINT
```

END 命令

结束并执行一条命令，不执行打印动作。

格式:

```
{command}
```

说明:

```
{command}: END
```

ABORT 命令

结束当前控制会话，不执行打印动作。

格式:

```
{command}
```

说明:

```
{command}: ABORT
```

ENCODING 命令

设置后续发送给打印机的数据的编码类型。

格式:

```
{command} {name}
```

说明:


```
{command}: ENCODING  
{name}: 编码类型 ASCII, UTF-8, GB18030
```

示例:

```
! 0 200 200 200 1  
ENCODING GB18030  
TEXT 24 0 20 30 Font: GBUNSG24, 中文  
ENCODING ASCII  
TEXT 7 0 20 80 Font 7, Size 0  
PRINT
```

打印结果:

Font: GBUNSG24, 中文

Font 7, Size 0

FORM命令

打印完成后定位到下一页的顶部。

格式:

```
{command}
```

说明:

```
{command}: FORM
```

示例:

```
! 0 200 200 3 1  
IN-CENTIMETERS  
CENTER  
TEXT 4 1 0 0.5 Form Command  
FORM  
PRINT
```

JOURNAL命令

默认情况下，如果在打印周期内（LABEL模式）发现明显标记（介质背面的黑色水平条，黑标，缝隙等），则打印机会检查介质对齐情况是否正确。必要时，可以使用JOURNAL命令禁用自动校正功能。用户程序负责在JOURNAL模式下进行检查并确保有纸。

格式：

```
{command}
```

说明：

```
{command}: JOURNAL
```

UNITS 命令

此命令可以针对控制会话中的所有后续命令字段指定度量单位。输入所有控制命令的坐标、宽度和高度时，均可精确到四位小数位。通过在控制会话的第一行后紧跟单位命令，还可将指定的度量单位应用于偏移和高度字段。在单位命令发出之前，打印机默认以点为单位。

格式：

```
{command}
```

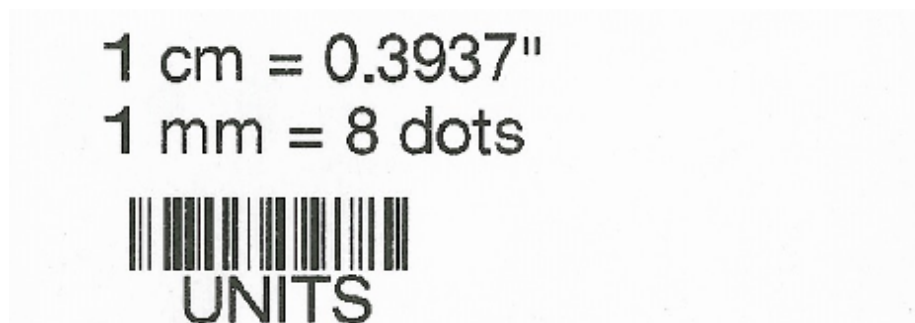
说明：

```
{command}: 从以下选项中选择一项  
            IN-INCHES - 以英寸为单位  
            IN-CENTIMETERS - 以厘米为单位  
            IN-MILLIMETERS - 以毫米为单位  
            IN-DOTS - 以点为单位
```

示例1：

```
! 0.3937 200 200 1 1  
IN-INCHES  
T 4 0 0 0 1 cm = 0.3937"  
IN-DOTS  
T 4 0 0 48 1 mm = 8 dots  
B 128 1 1 48 16 112 UNITS  
T 4 0 48 160 UNITS  
FORM  
PRINT
```

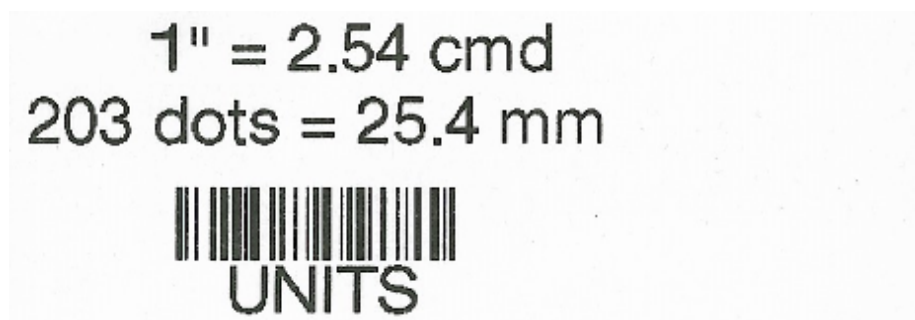
打印结果:



示例2:

```
! 0 200 200 2.54 1
IN-CENTIMETERS
T 4 0 1 0 1" = 2.54 cmd
IN-MILLIMETERS
T 4 0 0 6 203 dots = 25.4 mm
B 128 0.125 1 6 12 14 UNITS
T 4 0 16 20 UNITS
FORM
PRINT
```

打印结果:



使用注释

注释可以添加在命令会话第一行和“PRINT”命令之间。在文件中添加注释时，需要以“;”作为注释行的起始部分。“;”字符及其之后的整行文本都将被忽略。CONCAT与ENDCONCAT命令之间不可添加注释。

示例:

```
! 0 200 200 25 1
IN-MILLIMETERS
JOURNAL
; Center justify text
CENTER
; Print the words 'A COMMENT'
TEXT 5 1 0 5 A COMMENT
; Print the label and go to next form
FORM
PRINT
```

打印结果:



文本

内置字体示例

TEXT 命令

TEXT命令用于在标签上添加文本。这项命令及其各衍生命令可以控制使用的具体字体号和大小、标签上文本的位置以及文本的方向。标准内置字体能够以90度的增量旋转，如下例所示。

格式:

```
{command} {font} {size} {x} {y} {data}
```

说明:

{command}: 从以下选项中选择一项

TEXT (或T): 横向打印

VTEXT (或VT): 逆时针旋转90度, 纵向打印

TEXT90 (或T90): 同VTEXT

TEXT180 (或T180): 逆时针旋转180度, 反转打印

TEXT270 (或T270): 逆时针旋转270度, 纵向打印

{font}: 字体编号

{size}: 字体大小标识

{x}: 横向起始位置

{y}: 纵向起始位置

{data}: 打印文本数据

示例:

```
! 0 200 200 210 1
TEXT 4 0 200 100 TEXT
TEXT90 4 0 200 100 T90
TEXT180 4 0 200 100 T180
TEXT270 4 0 200 100 T270
FORM
PRINT
```

打印结果:

使用字体组

FONT-GROUP (FG) 命令

使用FG命令,用户可以将最多10个预缩放字体文件分至一个组。然后用户可在TEXT命令中指定字体组。如果文本命令中使用了字体组,则打印机将在标签可打印范围内选择字体组中的最大字体。在TEXT命令中进行指定时,{font}参数将指定为FG,而{SIZE}参数则指定为{fg}。请注意,用户还可以在CONCAT/ENCONCAT命令中指定FG命令。

格式:

```
{command} {fg fn fs} [fn fs] ...
```

说明:

{command}: FG
{fg}: 字体组编号。最多可指定10个字体组,有效范围为0~9。
{fn}: 字体编号。
{fs}: 字体大小标识。

示例:

```
! 0 200 200 250 1
; Specify fonts 0-1, 7-0, 5-0, 4-0 as members of font group 3
FG 3 0 0 7 0 5 0 4 0
VT FG 3 10 250 Ketchup
VT FG 3 70 250 Fancy Ketchup
VT FG 3 120 250 Extra Fancy Ketchup
VT FG 3 180 250 Large Size Extra Fancy Ketchup
FORM
PRINT
```

打印结果:



TEXT 串联命令 (CONCAT 和 VCONCAT)

使用文本串联，在同一文本行上可以为字符串设置不同的字符样式，使用统一间距进行打印。这项命令可以与矢量字体组合使用。

格式:

```
{command} {x} {y} {font} {size} {offset} {data} ... {font} {size} {offset} {data}
```

说明:

{command}: 从以下选项中选择一项。

CONCAT: 横向串联

VCONCAT: 纵向串联

{x}: 横向起始位置

{y}: 纵向起始位置

{font}: 字体编号

{size}: 字体大小标识

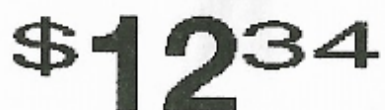
{offset}: 文本相对于起始位置的偏移值，用于对齐单个文本字符串或创建上标/下标字符

{data}: 打印的文本数据

<ENDCONCAT>: 终止串联

示例:

```
! 0 200 200 210 1
CONCAT 75 75
4 2 5 $
4 3 0 12
4 2 5 34
ENDCONCAT
FORM
PRINT
```

打印结果:The image shows the output of the CPCL code. It displays the text "\$1234" in a large, bold, monospaced font. The dollar sign is slightly smaller and positioned to the left of the numbers. The numbers are in a standard weight, while the dollar sign has a slightly different, more stylized appearance.

MULTILINE (ML) 命令

使用 MULTILINE (ML), 可以以相同字体和行高打印多行文本

格式:

```
{command} {height}
{text} {font} {size} {x} {y}
{data}
...
{data}
<ENDMULTILINE>
```

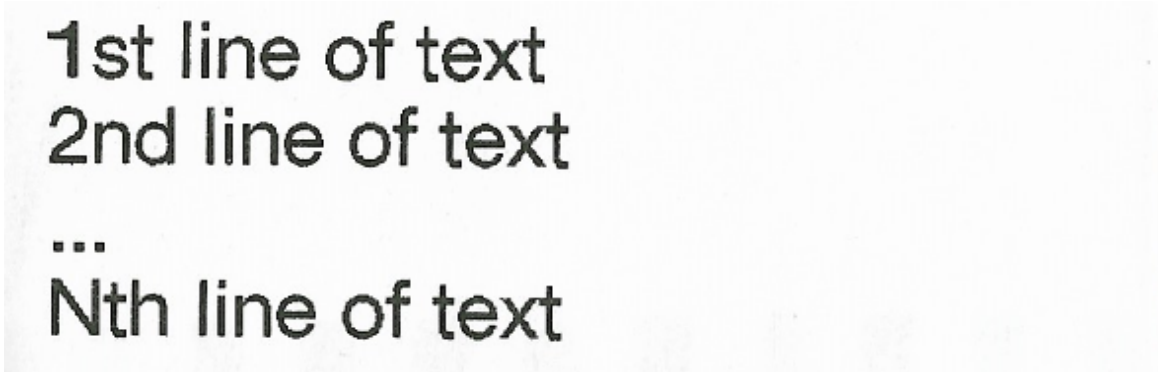
说明:

{command}: MULTILINE (或ML) -打印多行文本
{height}: 行高
{text}: 文本命令(TEXT、VTEXT等)
{font}: 字体名称
{size}: 字体大小标识
{x}: 横向起始位置
{y}: 纵向起始位置
{data}: 要打印的文本
<ENDMULTILINE>(或 ENDML): 终止MULTILINE

示例:

```
! 0 200 200 210 1
ML 47
TEXT 4 0 10 20
1st line of text
2nd line of text
...
Nth line of text
ENDML
FORM
PRINT
```

打印结果:



1st line of text
2nd line of text
...
Nth line of text

COUNT命令

COUNT命令可以用于打印多个标签，其中文本或条码中的数字针对每个标签一次递增或递减。只对TEXT/BARCODE命令字符串中作为结尾的数字字符串起作用，最多包含20个字符，且可以以‘-’符号作为前缀；不能以‘0’为增量或减量。字符串中的前导零将予以保留。一个标签文件中最多可使用三个COUNT命令。递增/递减的数字数据包含在TEXT或BARCODE命令中，后面紧跟COUNT命令。

格式:

```
{command} {numeric value}
```

说明:

{command}: COUNT

{numeric value}: 任何不超过20个字符的整数，可以是以‘-’开头的负数，输出结果中将保留前导零

示例:


```
! 0 200 200 210 3
; Print 3 labels
CENTER
TEXT 4 0 0 50 TESTING 001
COUNT 1
TEXT 7 0 0 100 Barcode Value is 123456789
COUNT -10
BARCODE 128 1 1 50 0 130 123456789
COUNT -10
FORM
PRINT
```

打印结果:



SETMAG 命令

SETMAG命令可将内置字体放大指定的放大倍数。

格式:

```
{command} {w} {h}
```

说明:

{command}: SETMAG


{w}: 字体的宽度放大倍数, 有效放大倍数为1到16。

{h}: 字体的高度放大倍数, 有效放大倍数为1到16。

示例:

```
! 0 200 200 210 1
CENTER
SETMAG 1 1
TEXT 0 0 0 10 Font 0-0 at SETMAG 1 1
SETMAG 1 2
TEXT 0 0 0 40 Font 0-0 at SETMAG 1 2
SETMAG 2 1
TEXT 0 0 0 80 Font 0-0 at SETMAG 2 1
SETMAG 2 2
TEXT 0 0 0 110 Font 0-0 at SETMAG 2 2
SETMAG 2 4
TEXT 0 0 0 145 Font 0-0 at SETMAG 2 4
; Restore default font sizes
SETMAG 0 0
FORM
PRINT
```

打印结果:



```
Font 0-0 at SETMAG 1 1
Font 0-0 at SETMAG 1 2
Font 0-0 at SETMAG 2 1
Font 0-0 at SETMAG 2 2
Font 0-0 at SETMAG 2 4
```

一维条码简介

UPC和EAN/JAN条码

UPC和EAN/JAN条码通常用于为产品提供唯一条码标识, 以便于查询价格和跟踪所售货物库存。此外, 它们还可用于商店优惠券、期刊和平装书。UPC和EAN/JAN条码一般为矩形形状, 包含固定长度的数据, 大部分情况下, 条码下面会印有肉眼可识别的文本。

UPC/EAN条码中的第一个数字为数制字符。规范中列出了字符 0 至 9 的用法, 如下所示:

- 0 常规 UPC 条码 (UPCA 和 UPCE)
- 1 预留
- 2 随机称重货品, 比如商店包装肉类。(仅 UPCA)
- 3 国家药物代码和国家卫生相关产品码, 采用现行的10位代码长度 (仅 UPCA) *
- 4 非食品类产品标记, 无条码格式限制且具有校验位保护 (仅 UPCA)
- 5 优惠券 (仅 UPCA)
- 6 常规UPC条码 (仅 UPCA)
- 7 常规UPC条码 (仅 UPCA)
- 8 预留
- 9 预留

校验和是条码中的最后一个数字, 可用于确保条码得到正确解码。打印机会自动计算这一数字。

创建UPCA和EAN13条码时可以或无需提供校验和。如果程序员提供了校验和数字, 则打印机将使用提供的校验位创建条码, 如果校验位不正确, 打印机将自动纠正。

UPCE条码适用于小型物品 (例如, 糖果和口香糖等), 这类条码通过“消零”方法创建。例如, 如果要编制01000000567, 则生成的条码将是仅包含数据、压缩方案和校验和的压缩条码, 没有任何多余的零。本例中, 条码将解码为1056707。有关消零的更多信息, 请参阅 Uniform Code Council (UCC) 发布的 UPC Symbol Specification Manual (UPC 符号规范手册)。

UPC-E和EAN8条码有一些限制。首先, 数制字符必须设置为0。如果您的应用程序需要, 则数制可能会设置为0之外的其他数字, 但数制1至9不支持UPCE和EAN8条码, 可能无法通过激光扫描设备解码。如果程序员提供了校验和数字, 则打印机将使用提供的校验位创建条码, 如果校验位不正确, 打印机将自动纠正。因此, 可以发送六位 (无数制, 无校验和)、七位 (有数制, 无校验和) 或八位 (有数制和校验和) 数字, 并创建条码。

Plus 2和Plus 5条码扩展仅适用于期刊和平装书。具体来说, 条码规范中指出, Plus 2扩展只能用于期刊期号。根据条码规范, Plus 2和Plus 5扩展不包含任何校验和。要创建扩展条码, 请在应编入UPC/EAN条码的数据与应编入扩展的数据之间添加一个空格。您也可以使用PLUS2和PLUS5条码单独创建扩展。请务必在UPC/EAN条码与扩展之间保留足够的空间 (大约是比率的 9倍)。

UPC/EAN 规范

条码符号	类型	数据长度	字符	理想宽/窄比	理想窄点宽	校验和计算
UPCA	UPCA	11或12	'0'~'9'	2:1	2	Mod10
UPCA plus 2	UPCA2	13	'0'~'9'	2:1	2	Mod10

条码符号	类型	数据长度	字符	理想宽/窄比	理想窄点宽	校验和计算
UPCA plus 5	UPCA5	16	'0'~'9'	2:1	2	Mod10
UPCE	UPCE	6,7或11	'0'~'9'	2:1	2	Mod10
UPCE plus 2	UPCE2	8或13	'0'~'9'	2:1	2	Mod10
UPCE plus 5	UPCE5	11或16	'0'~'9'	2:1	2	Mod10
EAN/JAN-13	EAN13	12或13	'0'~'9'	2:1	2	Mod10
EAN/JAN-13 plus 2	EAN132	14	'0'~'9'	2:1	2	EAN13
EAN/JAN-13 plus 5	EAN135	17	'0'~'9'	2:1	2	EAN13
EAN/JAN-8	EAN8	6,7或8	'0'~'9'	2:1	2	Mod10
EAN/JAN-8 plus 2	EAN82	9	'0'~'9'	2:1	2	Mod10
EAN/JAN-8 plus 5	EAN85	12	'0'~'9'	2:1	2	Mod10

Code 39或Code 3 of 9条码

Code 39条码适用于多种应用，包括库存盘点、医院应用或所扫描物品的条码长度不固定的情况。这类条码可使用字符0至9、A至Z、‘-’（破折号）、“.”（句点）、空格、“\$”（美元符号）、“/”（正斜杠）、“+”（加号）和“%”（百分）。还有一种名为“S/S”的特殊字符，用作起始/终止字符。F39和F39C类型允许使用回车符、换行符和空字符。要确保数据在应用中的完整性，请使用带校验和的条码。如果用户选择39C或F39C条码，则打印机会自动计算校验码。

Code 39 (3 of 9)规范

条码符号	类型	数据长度	字符	理想宽/窄比	理想窄点宽	校验和计算
CODE39	39	可变	参见正文	2.5:1	2	无
	39C	可变	参见正文	2.5:1	2	Mod43
	F39	可变	参见正文	2.5:1	2	无
	F39C	可变	参见正文	2.5:1	2	Mod43

Code93 或 Code 9 of 3 条码

Code 93条码适用于需要严格纠错功能的应用。为了实现此目的，Code 93条码包含了可自动计算并加入条码的两个单独查错校验和。这类条码用于库存盘点、医院应用或所扫描物品的条码长度不固定的环境。（参见上文关于Code 39的说明。）这类条码可以使用整个ASCII 128字符集。在进行数据和短语（例如“Code 93”）编码时非常有用。这类条码中有两个校验和，在解码的条码中分别位于最后一位字符和倒数第二位字符。Code 93的校验和计算十分复杂。有关校验和创建和解码的信息，请参阅条码符号规范。另外，条码符号规范没有为窄条的比率和宽度设定任何理想值。

Code93 规范

条码符号	类型	输入长度	字符	理想宽/窄比	理想窄点宽	校验和计算
Code93	93	可变	128 ASCII	1.5:1	1	两次mod47

Interleaved 2 of 5 条码

Interleaved 2 of 5（或ITF）条码适用于所有要扫描物品具有固定数据长度的应用。日期、电话号码或者固定长度的SKU非常适合采用这类条码。符号规范中指出，ITF条码可能只能完成部分解码，而没有任何可识别的差异。因此为了避免这一问题，必须保持数据长度不变，并在解码过程中执行纠错程序以确定数据是否正确。只有数字0-9可以编码，且数据必须为偶数位。如果是奇数位，则打印机将自动在条码开头处插入零(0)。有两种包含校验和的衍生条码：Interleaved 2 of 5 “with checksum” and German Post Code。German Post Code长度是固定的，包含12或14个字符（包括校验和）。

Interleaved 2 of 5 规范

条码符号	类型	输入长度	字符	理想宽/窄比	理想窄点宽	校验和计算
Interleaved 2 of 5	I2OF5	可变	仅'0'~'9'	2.5:1	2	参见正文
Interleaved 2 of 5 (带校验)	I2OF5C	可变	仅'0'~'9'	2.5:1	2	参见正文
德国邮政码	I2OF5G	11,12,13或14	仅'0'~'9'	2.5:1	2	mod10, 加权4,9

Code128 和 UCC-128 货运标准

Code 128适用于需要包含大量数据的应用（例如，货运、献血标记），以及所扫描条码长度不固定的任何其他应用。这类条码也包含校验和，作为条码最后一个字符用来确保数据完整。Code 128可以使用整个ASCII 128字符集，以及通用符号规范中的其他可用子集。三个起始和终止字符决定了要使用的字符集。这类条码的校验和后紧跟终止字符。条码符号规范中包含计算校验和的所有信息。

UCC-128货运标准为Uniform Code Council发布的Application Standard for Shipping Container Codes（货运集装箱条码应用标准）文档的一部分。

Code 128/UCC-128 规范

条码符号	类型	输入长度	字符	理想宽/窄比	理想窄点宽	校验和计算
Code128/A/B/C/Auto	128	可变	参见正文	N/A	2	mod103
UCC-128Std.	UCCEAN16	参见正文	参见正文	N/A	2	mode103

Codabar 条码

对于涉及的符号大多是数字且不同条码的长度存在变化的应用，Codabar是理想之选。它支持0至9、字符‘-’（破折号）、‘\$’（美元符号）、‘:’（冒号）、‘/’（正斜杠）、‘.’（句点）和‘+’（加号），以及起始/终止字符A至D。一个可选校验和将自动附加到条码数据中的最后一位，其后紧跟终止字符。条码符号规范中包含计算校验和的所有信息。条码类型NW7仅用于实现逆兼容。不建议针对新系统使用此命令。CODABAR与NW7之间没有区别。

Codabar 规范

条码符号	类型	输入长度	字符	理想宽/窄比	理想窄点宽	校验和计算
Codabar	CODABAR	可变	'0'~'9','A'~'D', 符号	2.5:1	2	无
	CODABAR 16	可变	'0'~'9','A'~'D', 符号	2.5:1	2	mod16

MSI Plessey条码

MSI Plessey条码是固定长度码，仅使用数字字符。这类条码主要用于杂货店应用。目前有三种不同的编码类型，分别具备不同的数据保护级别。有关校验和计算的更多信息，请参阅条码符号规范。条码类型

“PLESSEY”仅用于实现逆兼容。不建议针对新系统使用此命令。PLESSEY强制使用2:1的宽条与窄条宽度比。

MSI Plessey 规范

条码符号	类型	输入长度	字符	理想宽/窄比	理想窄点宽	校验和计算
MSI Plessey	MSI	最多13位	'0'~'9'	2:1	2	无
	MSI10	最多13位	'0'~'9'	2:1	2	mod10
	MSI1010	最多13位	'0'~'9'	2:1	2	两次mod10
	MSI1110	最多13位	'0'~'9'	2:1	2	mod11 mod10

Postnet和信封识别标记

US Postnet条码仅用于实现自动的邮件递送。为了遵循邮政规则，请在 200 d.p.i.打印机上条码高度设置为30点，宽/窄条比为 3.5:1，窄条宽度为3点。发送给条码的数据的长度可以为 5、9或11位。例如，要将邮件发送至 RI Warwick 30 Plan Way, 02886-1234，则数据应为：

- 5位 - 仅邮政编码：02886
- 9位 - 邮政编码+4位码：028861234
- 11位 - 邮政编码+4位码和地址中的后两位：02886123430

Postnet条码还包含一个自动计算的校验和，作为解码条码中的最后一个字符。有关更多信息，请参阅 USPS Postal Business Center的Publication 25。如果要制作包含地址的标签，请尝试使用常驻字体7或字体4，这样可以保证邮局使用的光学字符识别软件更好地识别。

Postnet 和 FIM 规范

条码符号	类型	输入长度	字符	理想宽/窄比	理想窄点宽	校验和计算
Postnet	POSTNET	5,9,11位	'0'~'9'	3.5:1	3	mod10
信封识别标记	FIM	仅A,B或C	A,B或C	1.5:1	6	N/A

一维条码命令

以下命令用于在标签上创建条码和设置条码格式。

BARCODE命令

用于以指定的宽和高横向或纵向打印条码

标准条码

格式:

```
{command} {type} {width} {ratio} {height} {x} {y} {data}
```

说明:

{command}: 从以下选项选择一项:
 BARCODE (或B): 横向打印条码
 VBARCODE (或VB): 纵向打印条码
{type}: 从表B1D-1选择一项:
{width}: 窄条单位宽度
{ratio}: 宽条与窄条比例, 参考表B1D-2
{height}: 条码单位高度
{x}: 横向起始位置
{y}: 纵向起始位置
{data}: 条码数据

表B1D-1

条码	命令
UPC-A	UPCA, UPCA2, UPCA5
UPC-E	UPCE, UPCE2, UPCE5
EAN/JAN-13	EAN13, EAN132, EAN135
EAN/JAN-8	EAN8, EAN82, EAN85
Code 39	39, 39C, F39, F39C
Code 93/Ext.93	93
Interleaved 2 of 5	I2OF5
Interleaved 2 of 5(带校验)	I2OF5C
German Post Code	I2OF5G
Code 128(自动)	128

条码	命令
UCC EAN 128	UCCEAN128
Codabar	CODABAR, CODABAR16
MSI/Plessey	MSI, MSI10, MSI1010, MSI1110
Postnet	POSTNET
FIM	FIM

表B1D-2

0 = 1.5:1	20 = 2.0:1	26 = 2.6:1
1 = 2.0:1	21 = 2.1:1	27 = 2.7:1
2 = 2.5:1	22 = 2.2:1	28 = 2.8:1
3 = 3.0:1	23 = 2.3:1	29 = 2.9:1
4 = 3.5:1	24 = 2.4:1	30 = 3.0:1
	25 = 2.5:1	

示例：

```
! 0 200 200 210 1
BARCODE 128 1 1 50 150 10 HORIZ.
TEXT 7 0 210 60 HORIZ.
VBARCODE 128 1 1 50 10 200 VERT.
VTEXT 7 0 60 140 VERT.
FORM
PRINT
```

打印结果：



BARCODE-TEXT命令

BARCODE-TEXT命令用于指定在打印条码时同时打印文本。该命令避免了使用单独文本命令注释条码的必要。文本位于条码下方的中间位置。

使用BARCODE-TEXT OFF（或BT OFF）禁止打印文本功能。

格式:

```
{command} {font number} {font size} {offset}
```

说明:

```
{command}: BARCODE-TEXT（或BT）  
{font number}: 注释条码的文本使用的字体编号  
{font size}: 注释条码的文本使用的字体大小  
{offset}: 文本距离条码的单位偏移量
```

示例:

```
! 0 200 200 400 1  
JOURNAL  
CENTER  
; Annotate bar codes using font 7 size 0  
; and offset 5 dots from the bar code.  
BARCODE-TEXT 7 0 5  
BARCODE 128 1 1 50 0 20 123456789  
VBARCODE 128 1 1 50 40 400 112233445  
BARCODE-TEXT OFF  
FORM  
PRINT
```

打印结果:



COUNT命令

COUNT命令用于打印多个标签时，对其中条码或文本内容中的数字，并且是字符串结尾的若干数字字符，做递增或递减操作。递增量或递减量最多可以包含20个数字，且可以以‘-’符号作为前缀。9至0计数将循环到9，00至99计数将循环到00，保留前导0。一个标签文件最多可使用3个COUNT命令。

BARCODE或TEXT的递增量或递减量由紧跟其后的COUNT命令设置。

格式：

```
{command} {numeric value}
```

说明：

{command}: COUNT

{numeric value}: 任意小于等于20个数字（递减时包含‘-’）的整数

示例：

```
! 0 200 200 210 3
; Print 3 labels
CENTER
TEXT 4 0 0 50 TESTING 001
COUNT 1
TEXT 7 0 0 100 Barcode Value is 123456789
COUNT -10
BARCODE 128 1 1 50 0 130 123456789
COUNT -10
FORM
PRINT
```

打印结果:



RSS符号

缩减码型 (RSS) 涵盖一系列线性符号，旨在为用户解决特定空间限制和应用需求提供相应功能。RSS最多支持74个字符。EAN.UCC RSS条码用于对识别号以及识别信息的补充数据进行编码。由EAN和UCC管理编号系统可以确保分配给特定物品的标识码是全球独一无二的，而且标识码及其关联的补充数据采用统一方式定义。对于使用RSS符号的用户来说，主要优势在于，可以在贸易活动中使用唯一定义的标识码和补充数据格式。

RSS Limited

RSS Limited用于编码完整的14位全球贸易项目代码 (GTIN)。这是最小的RSS码格式。其指示符位必须为“0”或“1”。这种符号不支持全向读取，也不适合销售点应用。

RSS-14

RSS-14用于编码完整14位EAN。使用正确编程的销售点条码扫描仪，可对线性符号中的UCC物品标识进行全向扫描。适用于销售点应用和标准 EAN.UCC物品标识。

RSS Expanded

RSS Expanded可以编码EAN.UCC物品标识和补充信息。

RSS Expanded最多支持74个数字字符或41个非数字字符的编码。

RSS Stacked

RSS-14 Stacked是RSS-14符号的变体，它能够将RSS-14码分割为两部分并分别放在两行中。这种符号适用于常规符号过宽的情况。它有两个版本：用于小型物品标识应用的高度截断版本和可以通过全向扫描仪读取的较高全向版本。RSS Expanded也可以作为堆迭符号打印在多行上。

RSS-Truncated

编码完整14位GTIN。适用于化妆品和珠宝之类的物品。其截断格式不支持全向读取。

RSS-14 Stacked Omnidirectional

编码完整14位GTIN。适用于销售点受空间限制需要窄高型符号的物品。例如，零散类物品，例如苹果、土豆和桔子。这种符号格式适用于超市常用的固定位置全向扫描仪，可以编码完整14位GTIN。

RSS 系列的所有符号都可作为独立的线性符号打印，或作为复合符号打印（包含直接打印在RSS线性部分的附带2D复合符号）。

RSS/复合符号

复合符号系列可以提供额外的供应链数据，同时允许与已用的符号共存。由线性条码符号和2D符号组成的符号称为复合符号。它包括一个EAN/UCC 线性符号和一个2D符号。2D符号可以将补充性应用标识元素字符串数据添加到EAN.UCC系统线性符号中。它具有以下优势：

- 复合符号是唯一具有可轻松扫描物品标识的符号。
- 复合符号在大小方面与矩阵符号类似，但在支持的扫描器技术种类上，前者更为广泛。复合符号比其他激光扫描2D符号小。

Composite Code A atop RSS Limited:

基于Micro-PDF的一种衍生形式。Composite Code A可以用于高效编制补充数据。复合符号不能通过全向扫描仪读取，但线性符号可以单独读取。

Composite Code B atop Code 128

复合形式基于Micro-PDF，在首个数据代码字位置使用了代码字920作为连接标记，且表示EAN.UCC数据压缩。CC-B可以与多种符号组合使用，但不能单独使用。复合符号无法在POS读取，但线性符号可以（如果有）。它最多可以包含由应用程序标识分隔的338个补充数据字符。

Composite Code-C atop Code 128

复合形式基于PDF-417，在首个数据代码字位置使用了代码字920作为连接标记，且表示EAN.UCC数据压缩。CC-C可以与多种符号组合使用，但不能单独使用。复合符号无法在POS读取，但线性符号可以（如果有）。它最多可以包含由应用程序标识分隔的338个补充数据字符。

RSS/复合 命令

格式：

```
{command} {type} {x} {y} {width} {lin_height} {sep_height} {segments}  
{subtype} {linear_data|2D_data}
```

说明：

{command}: 从以下选项中选择一项：
 BARCODE（或B）：横向打印条码
 VBARCODE（或VB）：纵向打印条码
{type}: RSS
{x}: 横向起始位置
{y}: 纵向起始位置
{width}: 最窄条的单位宽度
{lin_height}: 条码线性部分高度
{sep_height}: 分隔符的高度
{segments}: 每行的段数
{subtype}: RSS/复合子类。从[表rss-1]子类中选择一项
{linear_data|2D_data}: 条码数据（其中 | 为一维二维数据的分隔符）

表rss-1

子类	符号
1	RSS-14

子类	符号
2	RSS-14 Truncated
3	RSS-14 Stacked
4	RSS-14 Stacked Omnidirectional
5	RSS Limited
6	RSS Expanded
7	UPCA Composite
8	UPCE Composite
9	EAN-13 Composite
10	EAN-8 Composite
11	UCC-128 Composite A/B
12	UCC-128 Composite C

示例1 RSS14 Composite:

```
! 0 200 200 300 1
T 5 0 10 40 RSS14 Composite
T 5 0 10 70 1234567890123|1234567890
BARCODE RSS 10 110 2 25 3 22 1 1234567890123|1234567890
PRINT
```

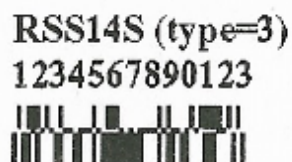
打印结果:



示例2 RSS14 Stacked:

```
! 0 200 200 300 1
T 5 0 10 40 RSS14S (type=3)
T 5 0 10 70 1234567890123
BARCODE RSS 10 100 3 25 3 22 3 1234567890123
PRINT
```

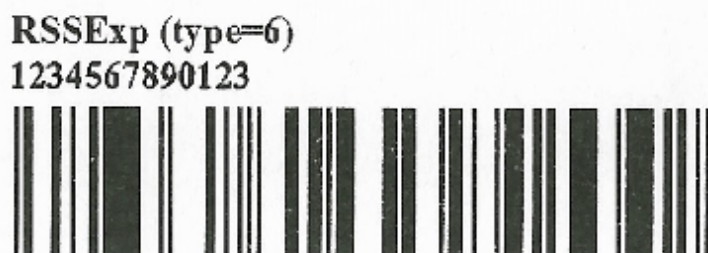
打印结果:



示例3 RSS Expanded:

```
! 0 200 200 300 1
T 5 0 10 40 RSSExp (type=6)
T 5 0 10 70 1234567890123
BARCODE RSS 10 100 3 25 3 22 6 1234567890123
PRINT
```

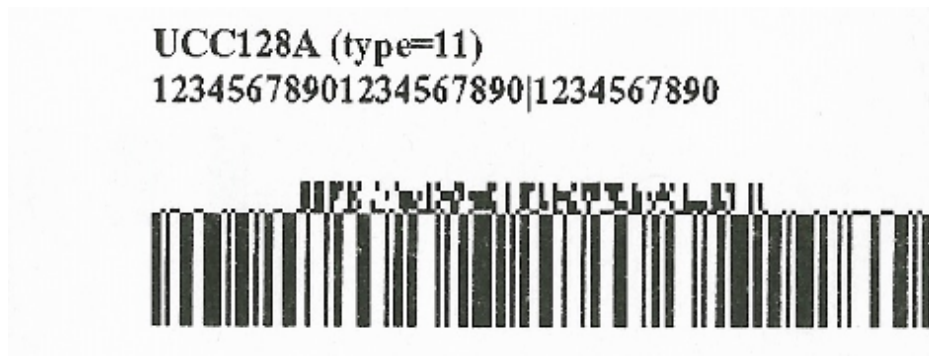
打印结果:



示例4 EAN Composite:

```
! 0 200 200 400 1
T 5 0 10 40 UCC128A (type=11)
T 5 0 10 70 12345678901234567890|1234567890
BARCODE RSS 10 140 3 25 3 22 11 12345678901234567890|1234567890
PRINT
```

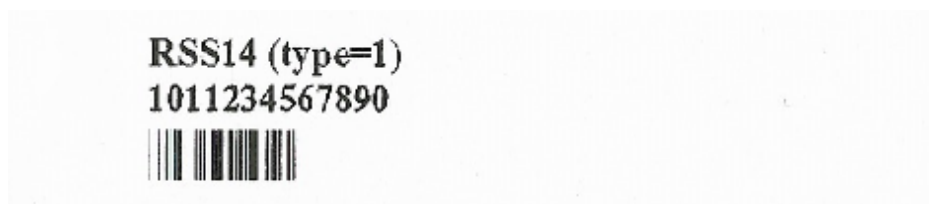
打印结果:



示例5 RSS 14:

```
! 0 200 200 300 1
T 5 0 10 40 RSS14 (type=1)
T 5 0 10 70 1011234567890
BARCODE RSS 10 100 1 25 3 22 1 1011234567890
PRINT
```

打印结果:



二维条码简介

PDF417码

PDF417码可以在狭小的空间里包含数量巨大的数据。仔细观察PDF417条码，您会发现其实它是由较小的条码堆叠而成。堆叠的数量和高度由用户控制。这类条码可以包含整个ASCII 255字符集，并能使用不同的编码方案和不同的纠错安全级别。最大数据编码量为2725个字符。

QR 码

QR 码能够处理各种各样的数据，例如数字和字母数字字符、Kanji、Kana、Hiragana、符号、二进制码和控制码等。一个符号中最多可以支持7089个字符的编码。因为QR 码在水平和垂直两个方向上都包含信息，所以在编码相同数据量的情况下，它只需使用传统条码约十分之一的空间。它的纠错功能能够恢复数据，即使符号有部分损坏或污染也无妨。QR 码利用位于符号三个角端处的位置检测图案完成全向(360°)读取。这三个位置检测图案能够避开背景干扰带来的负面影响，确保稳定的高速读取。

二维条码命令

PDF417

格式:

```
{command} {type} {x} {y} [XD n] [YD n] [C n] [S n]  
{data}  
<ENDPDF>
```

说明:

{command}: 从以下选项选择一项:

 BARCODE (或B): 横向打印条码

 VBARCODE (或VB): 纵向打印条码

{type}: PDF-417

{x}: 横向起始位置

{y}: 纵向起始位置

[XD n]: n为最窄元素的单位宽度, 范围1~32, 默认2。

[YD n]: n为最窄元素的单位高度, 范围1~32, 默认6。

[C n]: n为使用的列数, 数据列不包括起始/终止字符和左/右指示符。范围1~30, 默认3。

[S n]: n为安全级别, 指示要检测和/或纠正的最大错误量。范围0~8, 默认1。

{data}: 条码数据。

<ENDPDF>: 终止PDF417

示例:

```
! 0 200 200 210 1  
B PDF-417 10 20 XD 3 YD 12 C 3 S 2  
PDF Data  
ABCDE12345  
ENDPDF  
T 4 0 10 120 PDF Data  
T 4 0 10 170 ABCDE12345  
FORM  
PRINT
```

打印结果:



QR Code

格式:

```
{command} {type} {x} {y} [M n] [U n]  
{data}  
<ENDQR>
```

说明:

{command}: 从下面选项选择一项:
 BARCODE (或B): 横向打印条码
 VBARCODE (或VB): 纵向打印条码

{type}: QR
{x}: 横向起始位置
{y}: 纵向起始位置
[M n]: QR Code规范编号, 1或2。仅支持QR Code Model 2。
[U n]: n为模块的单位宽度/单位高度, 范围1~32, 默认6。
{data}: 条码数据
<ENDQR>: 终止QR Code

{data}除了包含实际的编码数据外, 还包含一些模式选择符号。输入数据类型可由打印机自动识别, 也可以手动设置。模式选择字符和编码数据间通过分隔符 (逗号) 区分。

自动识别的数据字段格式:

<纠错等级> <掩码号> <数据输入模式 (必须为“A”) >,<数据>

纠错等级:

- H - 极高可靠性级别 (H级)
- Q - 高可靠性级别 (Q级)
- M - 标准级别 (M级)
- L - 高密度级别 (L级)

掩码号可以省略或介于0-8之间:

- 无 – 软件自动选择掩码
- 0到7 – 使用编号0到7的掩码
- 8 – 无掩码

手动设置的数据字段格式:

<纠错等级> <掩码号> <数据输入模式 (必须为“A”) > <字符模式1> <数据1>,<字符模式2> <数据2>,<...> <...>,<字符模式n> <数据n>

字符模式:

- N - 数字
- A - 字母数字
- Bxxxx - 二进制, 二进制模式包含2字节BCD代码标识的数据长度 (xxxx)
- K - Kanji

不同的数据字段右都好分隔。

如果输入模式设置为“自动”, 则无法输入0x80_{0x9F}, 0xE0_{0xFF}的二进制代码。

示例1:

```
纠错等级: M
掩码号: 自动选择
数据输入模式: A
数据: QR Code ABC123
以上设定数据字段为: MA,QR Code ABC123
```

```
! 0 200 200 500 1
B QR 10 100 M 2 U 10
MA,QR Code ABC123
ENDQR
T 4 0 10 400 QR code ABC123
FORM
PRINT
```

打印结果:



QR code ABC123

示例2:

纠错等级: H
掩码号: 0
数据输入模式: M
字符模式: N
数据: 0123456789012345
以上设定的数据字段为: H0M,N0123456789012345

```
! 0 200 200 500 1
B QR 10 100 M 2 U 10
H0M,N0123456789012345
ENDQR
T 4 0 10 400 QR code ABC123
FORM
PRINT
```

打印结果:



QR code ABC123

示例3:

纠错等级: M
掩码: 自动选择
数据输入模式: M
字符模式: A
数据: AC-42
以上设定的数据字段为: MM,AAC-42

```
! 0 200 200 500 1
B QR 10 100 M 2 U 10
MM,AAC-42
ENDQR
T 4 0 10 400 QR code ABC123
FORM
PRINT
```

打印结果:



示例4:

纠错等级: L
掩码: 自动选择
数据输入模式: M
字符模式: A
数据: QR Code
字符模式: N
数据: 0123456789012345
字符模式: B
数据: qrcode
以上设定的数据字段为: LM,AQR Code,N0123456789012345,B0006qrcode

```
! 0 200 200 500 1
B QR 10 100 M 2 U 10
LM,AQR Code,N0123456789012345,B0006qrcode
ENDQR
T 4 0 10 400 QR code ABC123
FORM
PRINT
```

打印结果:



图形命令

BOX 命令

BOX命令用于生成指定线条宽度的矩形

格式:

```
{command} {x0} {y0} {x1} {y1} {width}
```

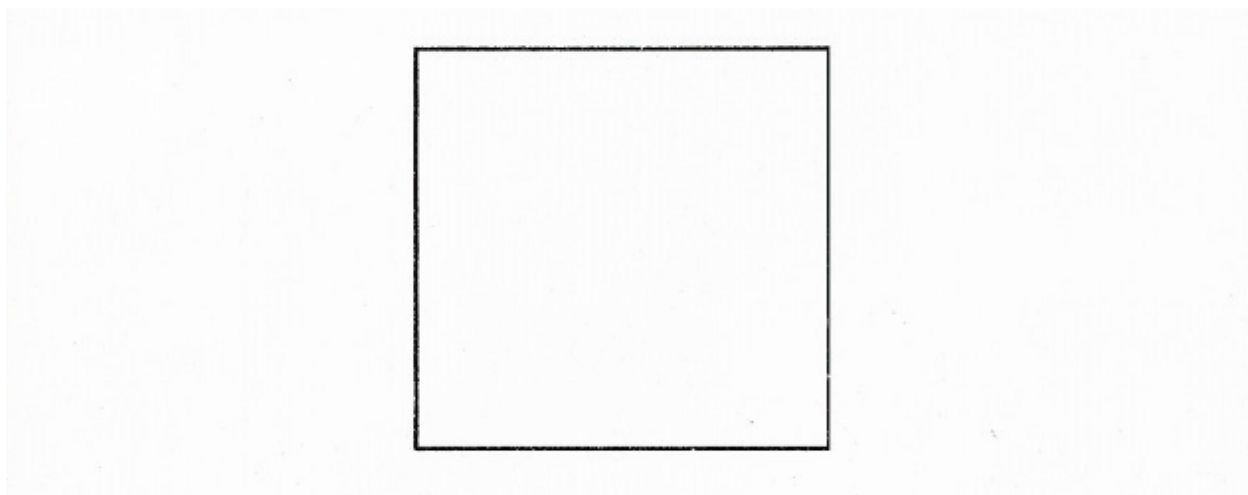
说明:

```
{command}: BOX  
{x0}: 左上角的X坐标  
{y0}: 左上角的Y坐标  
{x1}: 右下角的X坐标  
{y1}: 右下角的Y坐标  
{width}: 线条单位宽度
```

示例:

```
! 0 200 200 210 1  
BOX 0 0 200 200 1  
FORM  
PRINT
```

打印结果:



LINE命令

LINE命令用于绘制任何长度、宽度和方向的线条

格式:

```
{command} {x0} {y0} {x1} {y1} {width}
```

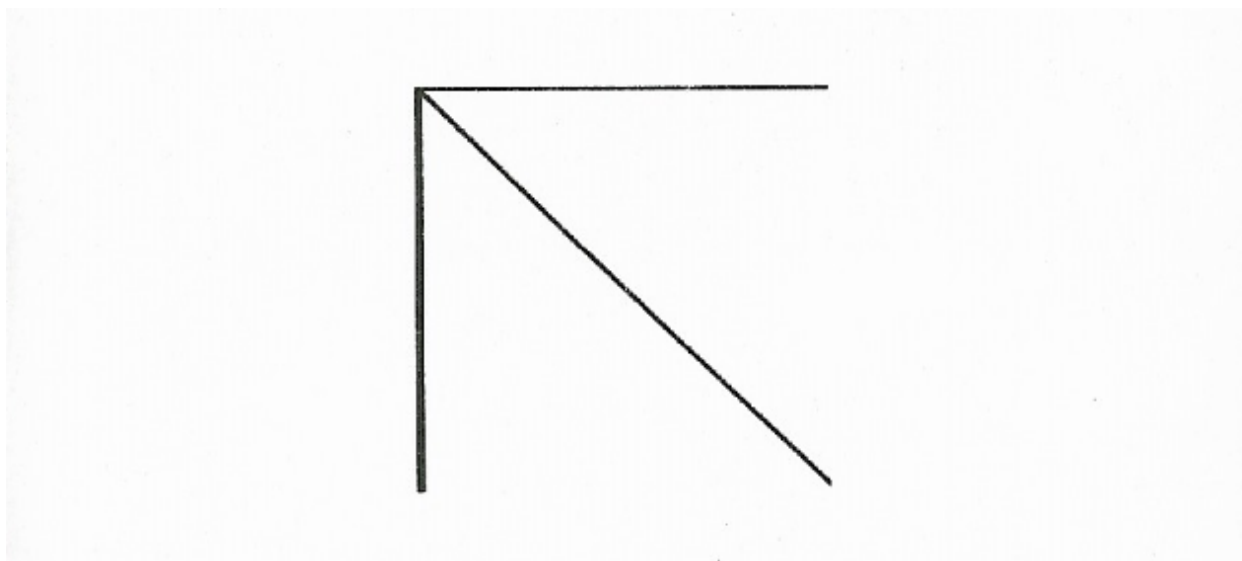
说明:

{command}: LINE (或L)
{x0}: 起始点的X坐标
{y0}: 起始点的Y坐标
{x1}: 终止点的X坐标
{y1}: 终止点的Y坐标
{width}: 线条单位宽度

示例:

```
! 0 200 200 210 1
LINE 0 0 200 0 1
LINE 0 0 200 200 2
LINE 0 0 0 200 3
FORM
PRINT
```

打印结果:



INVERSE-LINE命令

INVERSE-LINE命令语法格式与LINE命令相同。位于INVERSE-LINE命令所定义区域内的在其之前创建的对象黑色区域将重绘为白色，白色区域将重绘为黑色。这些对象可以包括文本、条码和/或图形（包括下载的.pcx文件）。INVERSE-LINE对在其之后创建的对象不起作用，即使这些对象位于该命令的覆盖区域内也是如此。在示例INVERSE2.LBL中，在INVERSE-LINE命令之后创建的文本字段部分仍然为黑色，因此不可见，即使被放置在INVERSE-LINE区域内也是如此。

格式:

{command} {x0} {y0} {x1} {y1} {width}

说明:

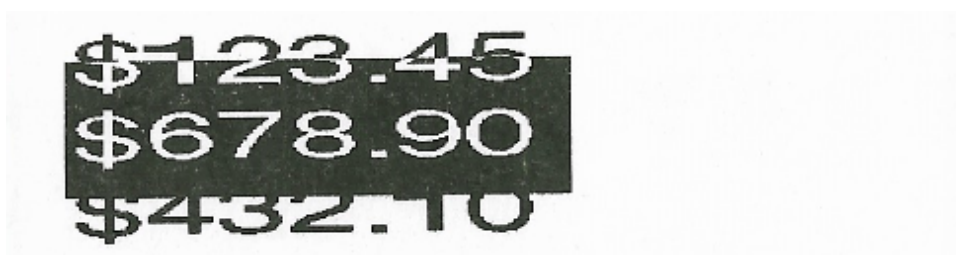
{command}: INVERSE-LINE (或IL)
{x0}: 起始点的X坐标
{y0}: 起始点的Y坐标
{x1}: 终止点的X坐标
{y1}: 终止点的Y坐标
{width}: 线条单位宽度

示例1:

```
! 0 200 200 210 1
CENTER
TEXT 4 0 0 45 SAVE
TEXT 4 0 0 95 MORE
INVERSE-LINE 0 45 145 45 45
INVERSE-LINE 0 95 145 95 45
FORM
PRINT
```

打印结果:**示例2:**

```
! 0 200 200 210 1
T 4 2 30 20 $123.45
T 4 2 30 70 $678.90
IL 25 40 350 40 90
T 4 2 30 120 $432.10
FORM
PRINT
```

打印结果:

PATTERN命令

PATTERN命令与LINE和SCALE-TEXT命令结合使用，可以更改用于填充这些形状的模式。下面列出了有效的图案值。

格式:

```
{command} {pattern number}
```

说明:

{command}: PATTERN

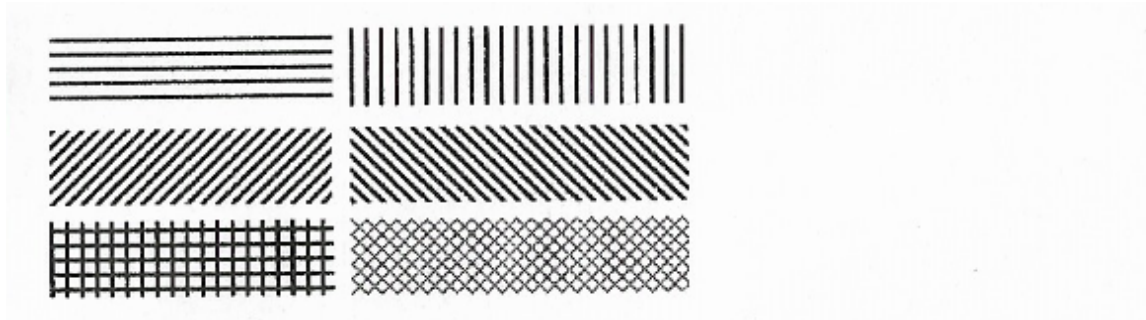
{pattern number}: 从以下选项中选择一项

- 100 填充（实心黑块/默认选项）
- 101 横线
- 102 竖线
- 103 右上升对角线
- 104 左上升对角线
- 105 方格
- 106 网格

示例:

```
! 0 200 200 700 1
; Draw horizontal and vertical patterns
PATTERN 101
LINE 10 10 160 10 42
PATTERN 102
LINE 170 10 350 10 42
; Draw left and right diagonal patterns
PATTERN 103
LINE 10 65 160 65 40
PATTERN 104
LINE 170 65 350 65 40
; Draw square and cross hatch patterns
PATTERN 105
LINE 10 115 160 115 40
PATTERN 106
LINE 170 115 350 115 40
FORM
PRINT
```

打印结果:



位图命令

图形命令用于打印位图，扩展图形数据使用十六进制ASCII字符（参见示例）。COMPRESSED-GRAPHICS命令使用十六进制数据，可以将数据大小减半。如果使用CG，对于每8位图形数据，将会发送一个8位字符。如果使用EG，将使用两个字符（16位）来传输8位图形数据，因此EG的效率会减半。但是由于该数据是字符数据，因此比二进制数据更容易处理和传输。

格式：

```
{command} {width} {height} {x} {y} {data}
```

说明：

{command}：从以下选项中选择一项
 EXPANDED-GRAPHIC（EG）：横向打印扩展图形
 VEXPANDED-GRAPHIC（VEG）：纵向打印扩展图形
 COMPRESSED-GRAPHIC（CG）：横向打印压缩图形
 VCOMPRESSED-GRAPHIC（CG）：纵向打印压缩图形
 {width}：图片宽度（字节）
 {height}：图片高度（点）
 {x}：横向起始位置
 {y}：纵向起始位置
 {data}：图形数据

示例：

```
! 0 200 200 210 1
EG 2 16 90 45 F0F0F0F0F0F0F0F0F0F0F0F0F0F0FF0F0F0F0F0F0F00F0F0F0F0F0F0F0F
FORM
PRINT
```

打印结果：



PCX命令

通过PCX命令，用户可以给打印机发送“.PCX”格式的图形数据，该图形必须是黑白二值图片。

格式：

```
{command} {x} {y}  
{data}
```

说明：

```
{command}: PCX  
{x}: 左上角的X坐标  
{y}: 左上角的Y坐标  
{data}: PCX图形数据
```

示例1：

Input 2为IMAGE.PCX图形数据

```
! 0 200 200 500 1  
PCX 0 30  
Input 2 (IMAGE.PCX)  
ENDPCX  
FORM  
PRINT
```

打印结果：



示例2:

IMAGE.PCX已预先下载并存储到打印机的flash的文件系统中，“!<”指示打印机取出图形数据。

```
! 0 200 200 500 1  
PCX 0 30 !<IMAGE.PCX  
FORM  
PRINT
```

打印结果:



设置命令

CONTRAST命令

CONTRAST命令用于设置整个标签的打印浓度。最淡的打印输出为等级0，最浓的打印输出为等级3。

格式:

```
{command} {level}
```

说明:

{command}: CONTRAST

{level}: 浓度等级:

0 = 默认

1 = 中等

2 = 浓

3 = 非常浓

TONE命令

TONE命令可代替CONTRAST命令设置打印浓度。最亮的打印输出为色调级别-99。最暗的色调级别为200。打印机在开机时的默认色调级别为0。色调级别设置在更改前对所有打印任务保持有效。TONE和CONTRAST命令不能组合使用。

格式:

```
{command} {level}
```

说明:

{command}: TONE

{level}: 等级 -99~200

CONTRAST与TONE的映射值:

Contrast 0 = Tone 0 Contrast 1 = Tone 100

Contrast 2 = Tone 200 Contrast 2 = 没有对应值

JUSTIFICATION命令

使用对齐命令可以控制字段的对齐方式。默认情况下，打印机将左对齐所有字段。对齐命令将对所有后续字段保持有效，直至指定了其他对齐方式。

格式:

```
{command} [end]
```

说明:

{command}: 从以下选项中选择一项

CENTER: 所有后续字段居中

LEFT: 所有后续字段左对齐

RIGHT: 所有后续字段右对齐

[end]: 对齐的结束点。如果没有设置结束点, 则横向打印的结束点为最大打印宽度, 纵向打印的结束点为0。

示例:

```
! 0 200 200 210 1
CENTER 383
TEXT 4 0 0 75 C
LEFT
TEXT 4 0 0 75 L
RIGHT 383
TEXT 4 0 0 75 R
FORM
PRINT
```

打印结果:



PAGE-WIDTH命令

打印机假定页面宽度为打印机的最大宽度。打印会话的最大高度由页面宽度和可用打印内存决定。如果页面宽度小于打印机的最大宽度, 则用户可以通过指定页面宽度来增加最大页面高度。

格式:

```
{command} {width}
```

说明:

{command}: PAGE-WIDTH (或PW)

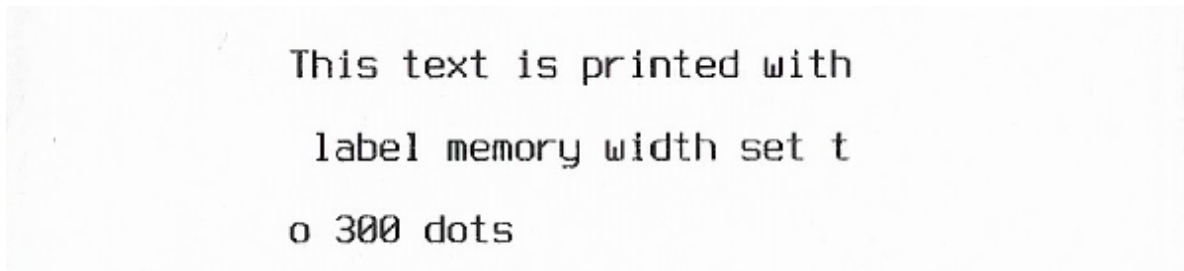
{width}: 页面单位宽度

示例1:


```
! UTILITIES
SETLP 7 0 15
PW 300
PRINT
```

This text is printed with label memory width set to 300 dots

打印结果:



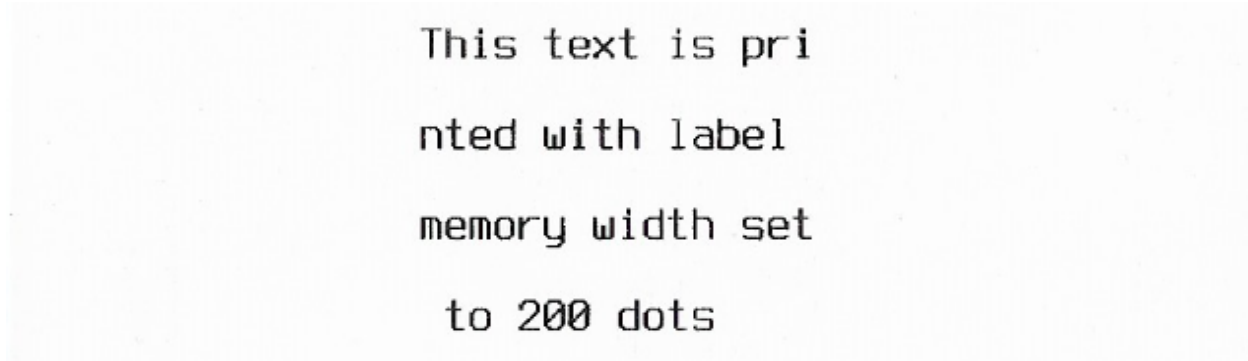
This text is printed with
label memory width set t
o 300 dots

示例2:

```
! UTILITIES
SETLP 7 0 15
PW 200
PRINT
```

This text is printed with label memory width set to 200 dots

打印结果:



This text is pri
nted with label
memory width set
to 200 dots

PACE命令

此命令用于批量打印。在激活PACE后，用户必须按下打印机的FEED(送纸)键才能打印剩余标签，直至完成批次数。默认情况下，开机时PACE功能处于禁用状态。

格式:

```
{command}
```

说明:

```
{command}: PACE
```

示例:

以下示例中，所示数据一次性发送给打印机，打印机打印第一张标签，一次按下FEED键可打印两外两张标签。

```
! 0 200 200 210 3
; Tell printer to print a label
; after each 'FEED' key press
; until all 3 labels are printed
PACE
; Printer holds journal stock
JOURNAL
; Center the text
CENTER
TEXT 4 1 0 10 Print 3 labels
TEXT 4 1 0 90 Using PACE
PRINT
```

AUTO-PACE命令

此命令可用于指示配备了标签取纸传感器的打印机延迟打印，直至取走了之前打印的标签。

格式:

```
{command}
```

说明:

```
{command}: AUTO-PACE
```

示例:

此示例指示打印机打印10个标签。打印机打印一个标签，然后等待该标签被取走，接着打印下一标签。

```
! 0 200 200 250 10
CENTER
TEXT 7 0 0 10 AUTO-PACE EXAMPLE
AUTO-PACE
FORM
PRINT
```

NO-PACE命令

此命令用于取消PACE和AUTO-PACE模式。打印机开机默认为NO-PACE模式。

格式:

```
{command}
```

说明:

```
{command}: NO-PACE
```

示例:

此示例指示打印机打印10个标签。打印机打印一个标签，然后等待该标签被取走，接着打印下一标签。第二组10个标签将以批量模式打印，且打印机将不等待操作人员取走标签。

```
! 0 200 200 250 10
TEXT 7 0 0 10 AUTO-PACE EXAMPLE
AUTO-PACE
FORM
PRINT
! 0 200 200 250 10
TEXT 7 0 0 10 NO-PACE EXAMPLE
NO-PACE
FORM
PRINT
```

WAIT命令

此命令用于在打印一个标签后引入一段延时。

格式:

```
{command} {delay-time}
```

说明:

```
{command}: WAIT  
{delay-time}: 延时实际, 以1/8秒为单位
```

示例:

以下示例中, 打印机每打印一张标签后延时10秒 ($10 \times 8 = 80$)

```
! 0 200 200 150 5  
WAIT 80  
TEXT 5 0 0 20 DELAY 10 SECONDS  
FORM  
PRINT
```

REWIND命令

此命令用于打开或关闭回卷马达。打印机开机回卷马达默认为打开状态。未配置回卷马达的设备忽略该指令。

格式:

```
{command}
```

说明:

```
{command}: REWIND-OFF 或 REWIND-ON
```

示例:

```
! 0 200 200 150 1  
REWIND-OFF  
TEXT 5 0 0 20 TURNS REWIND OFF  
PRINT
```

TENSION命令

此命令用于在打印标签之前和/或之后, 通过按预先指定的长度运行回卷电机来调整衬纸张力。对于配备了电机驱动的回卷机制的打印机, 此调整操作可以改善剥离器的性能。如果打印机没有配备电机驱动的回卷机制, 则忽略此命令。

格式:

```
{command} {length}
```

说明:

{command}: 从以下选项中选择一项

PRE-TENSION: 在打印标签之前执行张力调整

POST-TENSION: 在打印标签之后执行张力调整

{length}: 回卷电机用于收紧衬纸张力的单位长度。

示例:

在以下示例中，打印机将运行回卷电机移动30点行，在打印标签之前绷紧衬纸上的所有松弛部分以调整衬纸张力

```
! 0 200 200 150 1
PRE-TENSION 30
TEXT 5 0 0 20 ADJUSTS TENSION
PRINT
```

SPEED命令

此命令用于设置马达最高速度等级。不同机型的最小和最大速度等级不同。设定的等级越低，速度越小。此命令设置的速度仅在理想情况下可达到，实际速度受电池或供电电压，纸张厚度，打印浓度，环境温度，标签长度等等因素的影响。

格式:

```
{command} {speed level}
```

说明:

{command}: SPEED

{speed level}: 数值0~5，0为最慢速度

示例:

```
! 0 200 200 150 1
SPEED 4
TEXT 5 0 0 20 PRINTS AT SPEED 4
FORM
PRINT
```

SETSP命令

此命令用于改变文本字符间距。

格式:

```
{command} {spacing}
```

说明:

{command}: SETSP
{spacing}: 字符间距的单位尺寸, 默认为0。该设定值受UNITS命令影响

示例:

```
! 0 200 200 210 1
T 4 0 0 10 Normal Spacing
SETSP 5
T 4 0 0 50 Spread Spacing
SETSP 0
T 4 0 0 90 Normal Spacing
FORM
PRINT
```

打印结果:



Normal Spacing
Spread Spacing
Normal Spacing

UNDERLINE命令

此命令用于为文本字符添加下划线。该命令仅对支持下滑的字体有效。

格式:

```
{command} {mode}
```

说明:

```
{command}: UNDERLINE  
{mode}: ON或者OFF
```

示例:

```
! 0 200 200 200 1  
ENCODING GB18030  
UNDERLINE ON  
TEXT GBUNSG24.CPF 0 20 30 Underlined 下划线  
UNDERLINE OFF  
TEXT GBUNSG24.CPF 0 20 80 Normal 无下划线  
ENCODING ASCII  
UNDERLINE OFF  
PRINT
```

打印结果:

Underlined 下划线

Normal 无下划线

ON-OUT-OF-PAPER命令

此命令用于指示打印机在打印标签期间遇到错误（例如，纸张用完）时要采取的操作。

格式:

```
{command} {action} {number of retries}
```

说明:

```
{command}: ON-OUT-OF-PAPER
{action}: 从以下选项中选择一项
    PURGE: 在尝试指定次数后丢弃该标签
    WAIT: 不丢弃标签，打印机将一直等待直到错误恢复
{number of retries}: 指定的尝试次数
```

示例:

以下示例指示打印机尝试打印标签两次。

```
! 0 200 200 150 1
ON-OUT-OF-PAPER WAIT 2
TEXT 5 0 0 20 MAKE TWO ATTEMPT
FORM
PRINT
```

ON-FEED命令

打印机可将FEED按键或接收到换页字符（0x0c）功能配置为忽略，换页或重复打印最后一张标签。

格式:

```
{command} {action}
```

说明:

```
{command}: ON-FEED
{action}: 从以下选项中选择一项
    IGNORE: 无动作
    FEED: 进纸到下一页
    REPRINT: 重复打印最后一个标签
```

示例:

以下示例中，数据仅发送一次，打印机打印一张标签，随后按两次FEED键依次打印两张标签。


```
! 0 200 200 300 1
ON-FEED REPRINT
CENTER
JOURNAL
TEXT 4 1 0 20 PRESS FEED KEY
TEXT 4 1 0 100 TO REPRINT
TEXT 4 1 0 180 THIS TEXT
PRINT
```

PREFEED命令

此命令指示打印机在打印之前走纸指定的长度。

格式:

```
{command} {length}
```

说明:

```
{command}: PREFEED
{length}: 预先走纸的单位长度
```

示例:

以下示例设置打印机在打印之前走纸40点行

```
! 0 200 200 210 1
PREFEED 40
TEXT 7 0 0 20 PREFEED EXAMPLE
FORM
PRINT
```

POSTFEED命令

此命令指示打印机在打印完成后走纸指定长度。

格式:

```
{command} {length}
```

说明:

{command}: POSTFEED
{length}: 打印后走纸的单位长度

示例:

以下示例设置打印机在打印完成后走纸40点行。

```
! 0 200 200 210 1
TEXT 7 0 0 20 POSTFEED EXAMPLE
FORM
POSTFEED 40
PRINT
```

PRESENT-AT命令

此命令可用于将介质定位到打印机的撕纸刀处或者操作人员可以轻松取走打印后的标签的位置。发出 PRESENT-AT命令后，打印机将打印标签，然后在延迟一段时间后，将介质向前移动指定的距离。在开始新的打印作业之前，打印机会将介质回退相同的距离。

“delay”参数用于在执行批量打印作业时，避免不必要的向前/回退操作。PRESENT-AT命令可在标签文件中发出，也可在实用工具命令会话 (!UTILITIES...PRINT) 中发出。

格式:

{command} {length} {delay}

说明:

{command}: PRESENT-AT
{length}: 打印之后介质向前移动以及打印下一标签之前介质回退的单位长度，以点行为单位
{delay}: 打印标签之后打印机在向前移动介质前等待的时间间隔。

延时时间以 1/8 秒为单位递增。延迟“1”等价于1/8 秒。延迟“4”等价于 1/2 秒，以此类推。

示例:

以下示例指示打印机等待 1/4 秒，如果在此时间间隔内没有任何打印机活动，则将介质向前移动 80 点行。打印机将在打印下一标签之前将介质回退相同距离。

```
!0 200 200 250 1
TEXT 7 0 0 10 PRESENT-AT EXAMPLE
PRESENT-AT 80 2
FORM
PRINT
```

COUNTRY/CODEPAGE 命令

此命令用于对指定的国家设置适合的字符集。

格式:

```
{command} {name}
```

说明:

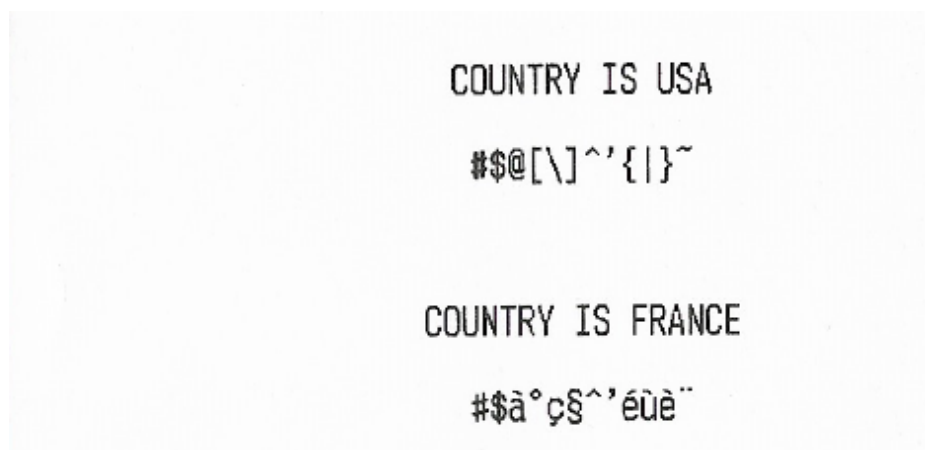
{command}: COUNTRY
{name}: 从下表中选择一项

USA	GERMANY	FRANCE
SWEDEN	SPAIN	NORWAY
ITALY	CP850	UK
LATIN9	CP974(泰语)	CHINA(简体中文, 繁体中文)
KOREA(韩语)	BIG5(繁体中文)	JAPAN-S(日语)

示例:

```
! 0 200 200 80 1
IN-MILLIMETERS
JOURNAL
CENTER
;Set the country as USA
COUNTRY USA
;Now print text from ISO substitution table
TEXT 24 0 0 8 COUNTRY IS USA
TEXT 24 0 0 15 #${[\]^'{}~}
; Set country for France and print the same text
COUNTRY FRANCE
TEXT 24 0 0 28 COUNTRY IS FRANCE
TEXT 24 0 0 35 #${[\]^'{}~}
PRINT
```

打印输出:



文件命令

使用格式文件

DEFINE-FORMAT和USE-FORMAT分别用于标识格式和数据。

格式文件避免了为每个标签重新发送格式相同的信息。通过预先下载格式，只需向打印机发送变量数据（如说明、价格等等）。

示例1:

不使用FORMAT命令的标签文件

```
! 0 200 200 210 1
CENTER
TEXT 4 3 0 15 $22.99
TEXT 4 0 0 95 SWEATSHIRT
BARCODE UPCA 1 1 40 0 145 40123456784
TEXT 7 0 0 185 40123456784
FORM
PRINT
```

打印输出:



以下示例描述了将示例1拆分为格式文件和数据的情形。

DEFINE-FORMAT

定义一个标签格式文件：使用DEFINE-FORMAT（或DF）命令标记格式文件的开头，PRINT命令标记文件结尾，“\”（双反斜杠）作为数据的占位符。

示例:

```
! DF SHELF.FMT
! 0 200 200 210 1
CENTER
TEXT 4 3 0 15 \
TEXT 4 0 0 95 \
BARCODE UPCA 1 1 40 0 145 \
TEXT 7 0 0 185 \
FORM
PRINT
```

USE-FORMAT

指示打印机使用指定的格式文件。使用该格式文件以及USE-FORMAT命令之后的数据创建一个标签，在获取格式文件后，用数据代替占位符“\”。

示例:

```
! UF SHELF.FMT
$22.99
SWEATSHIRT
40123456784
40123456784
```

同所有的打印命令一样，格式文件的每一行及其对应的变量必须以回车换行符结束。

定义一个格式文件后，该文件将存储在打印机内部并永久保存。一个已存在的文件可以通过重定义来改写，或者通过DEL命令删除。

格式文件名最多不超过8个字母或数字，扩展名不超过3个字母或数字。存储时，文件名及扩展名中的小写字母将自动被转换成大写字母。

机器控制命令

BEEP 命令

用于指示打印机控制蜂鸣器发出指定时间长度的声音。未配置蜂鸣器的打印机将忽略此命令。

格式:

```
{command} {beep_length}
```

说明:

```
{command}: BEEP
{beep_length}: 持续时间，以1/8秒为递增单位
```

示例:

此示例指示打印机控制蜂鸣器鸣叫2秒 ($16 \times 0.125s = 2s$)

```
! 0 200 200 210 1
CENTER
TEXT 5 0 0 10 beeps for two seconds
BEEP 16
FORM
PRINT
```

CUT 命令

打印标签完成后裁切标签。未配置切到的打印机将忽略此命令。

格式:

```
{command}
```

说明:

```
{command}: CUT
```

示例:

```
! 0 200 200 1.5 1
IN-INCHES
;Journal label 300 dots long
CENTER
;Print some text
TEXT 4 0 0 .15 CUT COMMAND
TEXT 4 0 0 .5 EXAMPLE
;After we print the label, cut it
CUT
PRINT
```

PARTIAL-CUT 命令

打印标签完成后裁切标签，保留部分标签不切断以便可以轻松撕下标签。

格式:

```
{command}
```

说明:

```
{command}: PARTIAL-CUT
```

示例:

```
! 0 200 200 1.5 1
IN-INCHES
;Journal label 300 dots long
JOURNAL
CENTER
;Print some text
TEXT 4 0 0 .15 PARTIAL CUT
TEXT 4 0 0 .5 EXAMPLE
;After we print the label, partially cut the label
PARTIAL-CUT
PRINT
```

CUT-AT 命令

此命令与CUT或PARTIAL-CUT命令配合使用，在打印机裁纸后回退指定的长度。未配置切到的打印机忽略此命令。

格式：

```
{command} {length}
```

说明：

```
{command}: CUT-AT
{length}: 指定裁纸后回退的长度
```

示例：

以下示例指示打印机打印一张标签，换页，裁纸后回退100点行。

```
! 0 200 200 250 1
CENTER
TEXT 7 0 0 20 CUT-AT EXAMPLE
CUT
CUT-AT 100
FORM
PRINT
```

行式打印模式

简介

除打印标签之外，打印机还可以使用行式打印模式生成收据、明细表或其它长度可变的文档。在行模式下，打印机接收原始ASCII文本，并以原始文本形式打印文档。

处于行模式的打印机可以通过解析特殊命令更改字体、字符间距，或者打印条码图片。收据可以像复杂的标签那样华丽和详细。

在行模式中，打印机将使用原始ASCII文本中的间距、字体和换页命令。打印机可以将回车换行符解析成换页符，它不打印tab字符。在标签模式，用户必须为每个项目提供x坐标和y坐标。在行模式，打印机可以自动换算坐标或者使用用户提供的坐标。

本章将介绍如何充分利用行式打印模式。其中包括有关如何使用实用工具命令的基础指示以及最常用于创建收据命令。本章末尾包含了一些示例文件及其打印结果，以及设计独特专业的收据的方法。

BEGIN-PAGE 命令

此命令用于清除行式打印内存和相关信息用以初始化行打印会话。

格式：

```
! U1 BEGIN-PAGE<cr><lf>
```

END-PAGE 命令

此命令用于关闭行式打印模式并打印行式打印内存里的内容。

格式：

```
! U1 END-PAGE<cr><lf>
```

使用实用工具函数的特殊命令

打印机可以一次执行一个或多个使用工具命令。

```
! U
SETLP 7 0 24
PAGE-WIDTH 576
PRINT
This line is printed as raw text.
These lines are in font 7 size 0.

! U1 SETLP 7 0 24
! U1 PAGE-WIDTH 576
This line is printed as raw text.
These lines are in font 7 size 0.
```

“! UTILITIES” 或者 “! U”（缩写）命令，必须以跟随CR/LF（回车/换行，或 "Enter"）的“PRINT”命令结束。“! U1”命令只执行一条使用工具命令，并且必须以CR/LF结尾。同时，“! U1”可以被放置在一行的任何位置。

示例：

```
Although this text is all on the same line, ! U1 SETLP 5 0 24 this font is new.
```

打印输出：

```
Although this text is all on the same line, this
font is new.
```

若要改变打印机的默认设置，可以将这些命令中的任一命令放置在AUTOEXEC.BAT文件中，具体请参考[章节13](#)。

LP-ORIENT 命令

此命令用于设置行式打印字符的旋转角度。

格式：

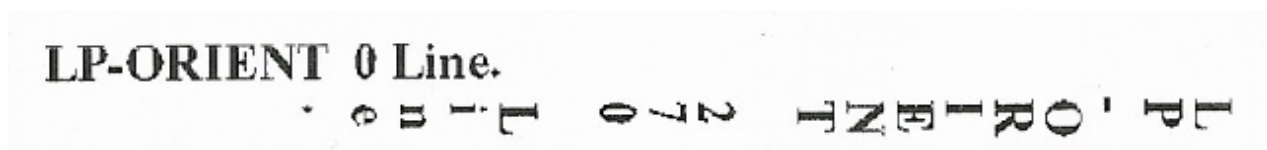
```
{command} {value}
```

说明：

```
{command}: LP-ORIENT
{value}: 旋转角度 0（默认）或 270
```

示例:

```
! U1 SETLP 5 0 30
! U1 LP-ORIENT 0
LP-ORIENT 0 Line.
! U1 LP-ORIENT 270
LP-ORIENT 270 Line.
```

打印结果:

UNITS 命令

此命令用于指定实用工具会话中所有或许命令字段的度量系统。所有坐标、宽度和高度可以精确到四位小数。在设置单位之前，系统默认单位为点。

格式:

```
<!> <UTILITIES>
{command}
<PRINT>
```

说明:

{command}: 从以下选项中选择一项

- IN-INCHES: 度量单位为英寸
- IN-CENTIMETERS: 度量单位为厘米
- IN-MILLIMETERS: 度量单位为毫米
- IN-DOTS: 度量单位为点（默认）

SETLP 命令

此命令可以同时设置行式打印模式的打印字体和行高。

格式:

```
! U1 SETLP {font name or number} {size} {unit height}
```

{unit height}: 行高，应该设置成当前使用字体的实际高度。

使用行式打印模式打印票据时，可以设置字体放大倍数。

示例：

```
! U1 SETLP 5 2 46
AURORA'S FABRIC SHOP
! U1 SETLP 7 0 24
123 Castle Drive, Kingston, RI 02881
(401)555-4CUT
```

打印结果：



AURORA'S FABRIC SHOP
123 Castle Drive, Kingston, RI 02881
(401)555-4CUT

SETLF 命令

此命令用于改变行高，但不设置字体。

格式：

```
! U1 SETLF {unit height}
```

示例：

命令 “! U1 SETLF 40” 将指示打印机在执行每个LF（十六进制0x0a）时每行进纸40点行。

```
! U
SETLP 4 0 10
SETLF 40
PRINT
Output 2
Text line
Text line
Text line
```

打印结果：

```
Output 2
Text line
Text line
Text line
```

使用X和Y坐标移动位置

即使在行式打印模式，也可以通过设置X和Y坐标向下移动纸张。

格式：

```
! U1 X {x坐标值}
! U1 Y {y坐标值}
! U1 XY {x坐标值} {y坐标值}
! U1 RX {相对x坐标值}
! U1 RY {相对y坐标值}
! U1 RXY {相对x坐标值} {相对y坐标值}
```

Y坐标值不能设置成负数。

LMARGIN 命令

此命令用于在行式打印模式设置左边距。LMARGIN可将所有内容整体向右偏移。

格式：

```
! U1 LMARGIN {offset dots}
```

说明：

{offset dots}：从左向右偏移的点数。

示例：

```
! U1 SETLP 5 0 24
! U1 LMARGIN 0
Left margin is 0
! U1 LMARGIN 100
Left margin is 100
```

打印结果:

```
Left margin is 0
Left margin is 100
```

SETBOLD 命令

此命令用于加粗文本。

格式:

```
! U1 SETBOLD {value}
```

说明:

{value}: 设定加粗值, 范围 0 到 5

示例:

```
! U1 SETBOLD 2
This text is in bold ! U1 SETBOLD 0
but this text is normal.
```

打印结果:

```
This text is in bold but this tex
t is normal.
```

SETSP 命令

此命令用于改变字符间距。

格式:

```
! U1 SETSP {unit value}
```

说明:

{unit value}: 字符间距宽度

示例:

```
Normal Text ! U1 SETSP 5  
SPREAD OUT TEXT
```

打印结果:

```
Normal Text SPREAD OUT TEXT
```

PAGE-WIDTH / PAGE-HEIGHT 命令

打印机允许用户设置页宽和页高。例如如果使用3寸打印机，纸张宽度为2寸宽，则用PAGE-WIDTH，或PW命令设置页宽。如果票据需要固定长度但是纸张没有黑标或其它位置标记，则可以使用PAGE-HEIGHT，或PH命令设置页高。

格式:

```
! U1 PW {unit width}  
! U1 PH {unit height}
```

说明:

{unit width}: 页面单位宽度
{unit height}: 页面单位高度

特殊的ASCII字符

换页

ASCII字符 (0x0c) 将进纸到下一个纸张标记 (黑标、缝隙等)，或者前进由PAGE-HEIGHT，SETFF或SET-TOP命令设定的固定长度。

回退

ASCII字符 (0x08) 为非破坏性退格符。退格后打印的字符将重叠打印在前一个字符之上。

SETFF 命令

此命令将介质的顶部与打印头对齐。执行该命令后，将在以下情况下进行对齐：

- 按下FEED键
- 执行换页命令0x0c

- 执行FORM命令

格式:

```
<!> <UTILITIES>
{command} {max-feed} {skip-feed}
<PRINT>
```

说明:

{command}: SETFF
{max-feed}: 打印机查找下一个标记的最大单位长度
{skip-length}: 打印机查找到标记后跳过顶部的单位长度

示例:

以下示例指示打印机查找下一个标记点，最大进纸长度为25mm，找到标记点后，打印机再进纸2.5mm。

```
! UTILITIES
IN-MILLIMETERS
SETFF 25 2.5
PRINT
```

SET-TOF 命令

此命令用于设置页面顶部与下一个（正值）或上一个（负值）可视标记或间隙末尾之间的距离。应使用临近页面顶部的可视标记或间隙来设置页面顶部。

格式:

```
{command} {d}
```

说明:

{command}: SET-TOF
{d}: 页面顶部与下一个或上一个可视标记或间隙末尾之间的距离，以二者之中更近的为准。
如果使用上一个可视标记作为基准，则指定的值应为负值；
如果使用下一个可视标记作为基准，则指定的值应为正值。

撕纸或切纸

PRESENT-AT 命令

请务必向收据的末尾添加几个额外的回车符/换行符（CR/LF 或 Enter）。这可以走纸足够多的空白区域，以便能够撕掉收据而不会破坏最后一行文字。打印机走完纸后，将纸撕掉。

如果您的打印机可以双向移动打印纸，当打印机开始打印下一项时，它会在开始打印前自动回收纸张以节省纸张。如果不带任何参数使用PRESENT-AT，则会将走纸单位设置为打印机的默认值。

对预印介质使用PRESENT-AT命令时应小心。如果在继续打印之前回收介质，介质可能无法准确地重新自行定位，并且这种误差量在各个标签之间并不完全相同。如果使用PRESENT-AT，建议在每个收据的始边和末边保留一定的缓冲区域。

格式：

```
! U1 PRESENT-AT {units}  
! U1 PRESENT-AT
```

说明：

{units}：打印后进纸单位长度

CUT-AT 命令

对于带有切纸器的打印机，CUT-AT命令将执行走纸、切纸然后回收以避免浪费纸张。如果不带任何参数使用CUT-AT，则会将走纸长度设置为打印机的默认值。

格式：

```
! U1 CUT-AT {units}  
! U1 CUT-AT
```

说明：

{units}：裁纸前进纸单位长度

CUT 和 PARTIAL-CUT 命令

打印机可以简单的用全切或半切命令来切纸。

格式：

```
! U1 CUT
! U1 PARTIAL-CUT
```

条码、图形和线条

条码：

打印机在行式打印模式可以创建任意一维条码。BARCODE命令与在标签模式下相同。BARCODE命令受对齐方式的影响。

图形：

打印机在行式打印模式可以打印PCX格式图形。但是不建议这么使用。

格式：

```
! U1 PCX {x坐标} {y坐标} !<{文件名.PCX}
```

SETLP-TIMEOUT 命令

此命令可设置打印机在指定时间内未收到数据，则启动打印。

格式：

```
! U1 SETLP-TIMEOUT {time}
```

说明：

{time}：延时时间，以1/8秒为单位

定义一张票据

示例1：

```
! U1 JOURNAL
! U1 SETLP 4 0 47
YOURCO RETAIL STORES

! U1 SETLP 7 0 24
14:40 PM   Thursdat,06/04/20
Quantity   Item           Unit      Total
1          Babelfish      $4.20     $4.20
          Tax:             5%      $0.21

! U1 SETSP 5
Total:! U1 SETSP 0
      $4.41
```

Thank you for shopping at YOURCO

打印结果:

YOURCO RETAIL STORES

```
14:40 PM   Thursdat,06/04/20
Quantity   Item           Unit      Total
1          Babelfish      $4.20     $4.20
          Tax:             5%      $0.21
```

Total: \$4.41

Thank you for shopping at YOURCO

示例2:

```
! U1 JOURNAL
! U1 SETLP 5 2 46
    AURORA'S FABRIC SHOP
! U1 SETLP 5 0 24
    123 Castle Drive,Kingston,RI 02881
    (401)555-4CUT
! U1 SETLP 7 0 24

4:20 PM Thursday,June 04,2020 Store: 142
Order Number: #59285691
Status: ! U1 SETSP 10
INCOMPLETE ! U1 SETSP 0

Item Description Quant. Price Subtotal Tax

1211 45" Buckram 5 yds @ $3.42/yd $17.10 Y
Z121 60" Blue Silk 10 yds@ $15.00/yd $150.00 N
Z829 60" Muslin 20 yds@ $1.00/yd $20.00 Y

    SUBTOTAL: $187.10
    RHODE ISLAND SALES TAX 7.00%:    $2.60
TOTAL:    $189.70

! U1 SETLP 7 1 48
    PLEASE BRING THIS RECEIPT TO THE CASHIER
    WIDTH THE REST OF YOUR PURCHASES.

! U1 CENTER
! U1 B 128 1 2 100 0 0 59285691 ST 187.10 T 2.60
```

打印结果:

AURORA'S FABRIC SHOP

123 Castle Drive, Kingston, RI 02881
(401)555-4CUT

4:20 PM Thursday, June 04, 2020 Store: 142

Order Number: #59285691

Status: I N C O M P L E T E

Item Description Quant. Price Subtotal Tax

1211 45" Buckram 5 yds @ \$3.42/yd \$17.10 Y

2121 60" Blue Silk 10 yds@ \$15.00/yd \$150.00 N

2829 60" Muslin 20 yds@ \$1.00/yd \$20.00 Y

SUBTOTAL: \$187.10

RHODE ISLAND SALES TAX 7.00%: \$2.60

TOTAL: \$189.70

PLEASE BRING THIS RECEIPT TO THE CASHIER
WITH THE REST OF YOUR PURCHASES.

