

# [CPCL Linux SDK]

[打印机 CPCL 指令集开发帮助文档 v2.0.4]

## 目录

[CPCL Linux SDK] .....	1
1. 手册信息 .....	3
2. 运行平台 .....	3
3. 备注 .....	3
4. 方法 .....	3
4.1. InitPrinter .....	3
4.2. ReleasePrinter .....	4
4.3. OpenPort .....	4
4.4. ClosePort .....	5
4.5. WriteData .....	5
4.6. ReadData .....	6
4.7. CPCL_AddLabel .....	6
4.8. CPCL_SetAlign .....	7
4.9. CPCL_AddText .....	8
4.10. CPCL_AddBarCode .....	9
4.11. CPCL_AddBarCodeText .....	10
4.12. CPCL_AddQRCode .....	12
4.13. CPCL_AddPDF417 .....	13
4.14. CPCL_AddBox .....	15
4.15. CPCL_AddLine .....	15
4.16. CPCL_AddImage .....	16
4.17. CPCL_AddImageData .....	17
4.18. CPCL_SetFontSize .....	18
4.19. CPCL_SetDensity .....	18
4.20. CPCL_SetSpeed .....	19
4.21. CPCL_SetTextSpacing .....	19
4.22. CPCL_SetLeftMargin .....	20
4.23. CPCL_SetTextBold .....	20
4.24. CPCL_SetTextUnderline .....	21
4.25. CPCL_Abort .....	21
4.26. CPCL_Print .....	22
4.27. CPCL_NextLabelPos .....	22
4.28. CPCL_PreFeed .....	23
4.29. CPCL_PostFeed .....	23
4.30. CPCL_GetPrinterStatus .....	24

## 1. 手册信息

本 SDK 手册提供了 Linux 应用程序开发所需的\*.so 文件信息。  
我们在不断地努力提高和升级我们所有产品的功能与质量。  
之后，产品规格和用户手册的内容可能会更改，将不再另行通知。

## 2. 运行平台

Linux debian 5.10.0 及以上版本

## 3. 备注

1. 错误代码返回值大于0时，属于 Linux 系统内部错误，请查阅相关帮助文档。
2. 打印机分辨率为200 dpi时，1 mm=8 dot(点);打印机分辨率为300 dpi时，1 mm=12 dot(点)。
3. SDK中引用了第三方库:libserialport、libusb-1.0。请提前在操作系统安装。
4. 串口连接需要root权限。

## 4. 方法

### 4.1.InitPrinter

此函数功能为创建指定机型的打印机对象(在进行任何打印机操作之前必须先创建打印机对象)。

```
void* InitPrinter (  
    const TCHAR* model  
);
```

**参数：**

*const TCHAR\* model*  
[in] 指定目标打印机型号。

**返回值：**

成功：返回打印机对象的句柄  
失败：返回 NULL

## 4.2.ReleasePrinter

此函数功能为释放已创建指定机型打印机对象的资源(在操作结束后且不再进行打印机操作时必须释放创建的打印机对象)。

```
int ReleasePrinter (  
    void* hPrinter  
);
```

参数：

*void\* hPrinter*

[in] 需要释放的目标打印机对象的句柄。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败

## 4.3.OpenPort

此函数功能为打开通讯端口，与打印建立连接。连接成功后才能正常使用其它功能。连接失败时，请查看函数返回的错误信息。目前支持 USB、网络通信、串口通信。

```
int OpenPort (  
    void* hPrinter,  
    const TCHAR* setting  
);
```

参数：

*void\* hPrinter*

[in] 打印机对象句柄。

*const TCHAR\* setting*

[in] 设置连接目标打印机的通讯端口参数。具体内容查看下表：

配置列表：

类别	配置	描述	示例
USB	USB,path	USB,USB路径	USB,/001/007
NET	NET, IP 地址 (IPV4)[,端口]	指定网络打印机的IP地址和端口。如果不指定端，默认端口是9100。	NET,192.168.1.10 NET,192.168.1.10,9100
COM	COM,path,rate	指定连接的串口路径和波特率。	COM,/dev/ttyACM0,19200

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_USB_DEVICE_NOT_FOUND	-17	找不到USB设备
ERROR_IO_OPEN_FAILED	-8	打开IO失败
ERROR_CM_INVALID_PARAMETER	-1	参数无效

## 4.4. ClosePort

此函数功能为关闭通讯。当不使用端口通讯时，请关闭端口。

```
int ClosePort (
    void* hPrinter
);
```

**参数：**

**void\* hPrinter**

[in] 创建的目标打印机对象。

**返回值：**

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-3	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-2	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败

## 4.5. WriteData

此函数功能为向打印机发送数据。

```
int WriteData(
    void* handle,
    unsigned char* buffer,
    unsigned int size
);
```

**参数：**

**void\* handle**

[in] 打印机对象句柄。

**unsigned char\* buffer**

[in] 发给打印机的数据，数据是十六进制字符串。

**unsigned int size**

[in] 发送数据的长度。

**返回值：**

错误代码	值	描述
------	---	----

ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

## 4.6.ReadData

此函数功能为读取打印机的数据。

```
int ReadData(
    void* handle,
    unsigned char* buffer,
    unsigned int size
);
```

**参数：**

*void\* handle*

[in] 打印机对象句柄。

*unsigned char\* buffer*

[in] 需要读取的打印机数据。

*unsigned int size*

[in] 所需读取的数据长度。

**返回值：**

错误代码	值	描述
	>0	成功，读取的数据长度
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_OPEN_FAILED	-8	打开IO失败

## 4.7.CPCL\_AddLabel

此函数功能为设置标签大小和打印的数量。

```
int CPCL_AddLabel(
    void* handle,
    int offSet,
    int height,
    int qty
);
```

**参数：**

*void\* handle*

[in,out] 创建的目标打印机对象。

*int offSet*

[in] 标签的起始偏移量 (单位: dot)。  
备注:使所有字段按设置值进行水平位移。

*int height*

[in] 打印的标签高度 (范围: 0-2400, 单位: dot)。

*int qty*

[in] 打印的标签数量。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.8.CPCL\_SetAlign

此函数功能为设置文本对齐方式。

```
int CPCL_SetAlign(
    void* handle,
    int align
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int align*

[in]设置文本对齐方式。

位置	值
左边	0
中间	1
右边	2

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.9.CPCL\_AddText

此函数功能为打印文本。

```
int CPCL_AddText(  
    void* handle,  
    int rotate,  
    const char* fontType,  
    int fontSize,  
    int xPos,  
    int yPos,  
    const TCHAR* data  
);
```

**参数：**

**void\* handle**

[in,out] 创建的目标打印机对象。

**int rotate**

[in] 设置打印方向。

旋转角度	值
不旋转	0
旋转90度	1
旋转180度	2
旋转270度	3

**const char\* fontType**

[in] 字体类型。

Font	Value
Font_12x24	"0"
Font_8x16	"1"

**int fontSize**

[in] 字体大小 (范围: 0-7)。

**int xPos**

[in] 水平起始位置 (范围: 0-32000, 单位: dot)。

**int yPos**

[in] 垂直起始位置 (范围: 0-32000, 单位: dot)。

**const TCHAR\* data**

[in] 文本数据。

**返回值：**

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码



# 4.10. CPCL\_AddBarCode

此函数功能为打印条码。

```
int CPCL_AddBarCode(  
    void* handle,  
    int rotate,  
    int type,  
    int width,  
    int ratio,  
    int height,  
    int xPos,  
    int yPos,  
    const TCHAR* data  
);
```

参数:

*void\* handle*  
[in,out] 创建的目标打印机对象。

*int rotate*  
[in] 设置旋转模式。  
0: 不旋转  
1: 旋转90度

*int type*  
[in] 设置条码类型。

条 码 类 型	值
UPCA	0
UPCA2	1
UPCA5	2
UPCE	3
UPCE2	4
UPCE5	5
EAN13	6
EAN132	7
EAN135	8
EAN8	9
EAN82	10
EAN85	11
39	12
39C	13
F39	14
F39C	15
93	16
I2OF5	17
I2OF5C	18
I2OF5G	19
128	20
UCCEAN128	21
CODABAR	22
CODABAR16	23

MSI	24
MSI10	25
MSI1010	26
MSI1110	27
POSTNET	28
FIM	29

*int width*

[in] 设置条码宽度 (单位: dot)。

*int ratio*

[in] 条码黑白块宽窄比率。

*int height*

[in] 设置条码高度 (单位: dot)。

*int xPos*

[in] 水平起始位置 (范围: 0-32000, 单位: dot)。

*int yPos*

[in] 垂直起始位置 (范围: 0-32000, 单位: dot)。

*const TCHAR\* data*

[in] 条码数据。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.11. CPCL\_AddBarCodeText

此函数功能为显示条码内容。

```
int CPCL_AddBarCodeText(
    void* handle,
    int enable,
    int fontType,
    int fontSize,
    int offset
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int enable*

[in] 是否显示条码内容。

0:不显示

1:显示

*int* *fontType*

[in] 字体类型 (取值参考下图)。

*int* *fontSize*

[in] 字体大小。(取值参考下图)。

*int* *offset*

[in] 位移距离

字体类型和大小:

Font	Size	Width	Height	Char. Height	Char Width
0	0	1	1	9	8
0	1	2	1	9	16
0	2	1	2	18	8
0	3	2	2	18	16
0	4	3	2	18	32
0	5	2	3	36	16
0	6	3	3	36	32
1	0	1	1	48	8-25 Variable
2	0	1	1	12	20
2	1	1	2	24	20
4 (A)	0	1	1	47	8-43 Variable
4 (A)	1	1	2	94	8-43 Variable
4 (B)	2	1	½	45	26-51 Variable
4 (B)	3	1	1	90	26-51 Variable
4 (B)	4	1	2	180	26-51 Variable
4 (B)	5	1	3	270	26-51 Variable
4 (B)	6	1	4	360	26-51 Variable
4 (B)	7	1	5	450	26-51 Variable
5	0	1	1	24	5-23 Variable
5	1	1	2	48	5-23 Variable
5	2	2	2	46	8-39 Variable
5	3	2	3	92	8-39 Variable
6	0	1	1	27	28
7	0	1	1	24	12
7	1	1	2	48	12

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.12. CPCL\_AddQRCode

此函数功能为打印二维条码。

```
int CPCL_AddQRCode(
    void* handle,
    int rotate,
    int xPos,
    int yPos,
    int model,
    int unitWidth,
    int eccLevel,
```

```
const TCHAR* data
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int rotate*

[in] 设置旋转模式。

0: 不旋转

1: 旋转90度

*int xPos*

[in] 水平起始位置 (范围: 0-32000, 单位: dot)。

*int yPos*

[in] 垂直起始位置 (范围: 0-32000, 单位: dot)。

*int model*

[in] 设置二维码版本 (1:基本版,2:强化版)。

*int unitWidth*

[in] 设置二维码宽度。(范围是1到32.默认值是6)

*int eccLevel*

[in] 纠错级别。

容错级别	值
7%	0
15%	1
25%	2
30%	3

*const TCHAR\* data*

[in] 二维码数据。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.13. CPCL\_AddPDF417

此函数功能为打印 PDF417二维条码。

```
int CPCL_AddPDF417(
```

```
void* handle,
```

```
int rotate,
```

```
int xPos,
```

```
int yPos,
```

```
int xDots,
```

```
int yDots,
```

```

    int columns,
    int eccLevel,
    const TCHAR* data
);

```

#### 参数:

**void\* handle**

[in,out] 创建的目标打印机对象。

**int rotate**

[in] 设置旋转模式。

0: 不旋转

1: 旋转90度

**int xPos**

[in] 水平起始位置 (范围: 0-32000 , 单位: dot) 。

**int yPos**

[in] 垂直起始位置 (范围: 0-32000 , 单位: dot) 。

**int xDots**

[in] 像素点宽度 (单位: dot, 范围是1到32, 默认值是2) 。

**int yDots**

[in] 像素点高度 (单位: dot 范围是1到32, 默认值是6) 。

**int columns**

[in] 条码列数。(范围是1到30; 默认值为3)

**int eccLevel**

[in] 纠错级别。(范围是0到8; 默认值为1)

容错程度	值
0	0
2	1
6	2
14	3
30	4
62	5
126	6
254	7
510	8

**const TCHAR\* data**

[in] PDF417二维条码数据。

#### 返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.14. CPCL\_AddBox

此函数功能为打印机画出矩形框。

```
int CPCL_AddBox(  
    void* handle,  
    int xPos,  
    int yPos,  
    int endXPos,  
    int endYPos,  
    int thickness  
);
```

**参数：**

*void\* handle*

[in,out] 创建的目标打印机对象。

*int xPos*

[in] 水平起始位置 (范围：0-32000，单位：dot)。

*int yPos*

[in] 垂直起始位置 (范围：0-32000，单位：dot)。

*int endXPos*

[in] 水平结束位置 (范围：0-32000，单位：dot)。

*int endYPos*

[in] 垂直结束位置 (范围：0-32000，单位：dot)。

*int thickness*

[in] 矩形边框宽度。

**返回值：**

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.15. CPCL\_AddLine

此函数功能为画线。

```
int CPCL_AddLine(  
    void* handle,  
    int xPos,  
    int yPos,  
    int endXPos,  
    int endYPos,  
    int thickness  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int xPos*

[in] 水平起始位置 (范围: 0-32000 , 单位: dot) 。

*int yPos*

[in] 垂直起始位置 (范围: 0-32000 , 单位: dot) 。

*int endXPos*

[in] 水平结束位置 (范围: 0-32000 , 单位: dot) 。

*int endYPos*

[in] 垂直结束位置 (范围: 0-32000 , 单位: dot) 。

*int thickness*

[in] 直线宽度。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.16. CPCL\_AddImage

此函数功能为打印图片 (仅支持单色bmp格式) 。

```
int CPCL_AddImage(  
    void* handle,  
    int rotate,  
    int xPos,  
    int yPos,  
    const TCHAR* filePath  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int rotate*

[in] 设置图片旋转模式。

0: 不旋转

1: 旋转90度

*int xPos*

[in] 水平起始位置 (范围: 0-32000 , 单位: dot) 。

*int yPos*

[in] 垂直起始位置 (范围: 0-32000 , 单位: dot) 。

*const TCHAR\* filePath*

[in] 图片的正确路径。

返回值:



错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.17. CPCL\_AddImageData

此函数功能为打印图片 (直接传入图片像素数据)。

```
int CPCL_AddImageData(
    void* handle,
    int rotate,
    int widthBytes,
    int height,
    int xPos,
    int yPos,
    const char* data
);
```

**参数：**

*void\* handle*

[in,out] 创建的目标打印机对象。

*int rotate*

[in] 设置图片旋转模式。

0: 不旋转

1: 旋转90度

*int widthBytes*

[in] 图像数据宽度。(单位:dpi)

*int height*

[in] 图像高度。(单位:dpi)

*int xPos*

[in] 水平起始位置 (范围：0-32000，单位：dot)。

*int yPos*

[in] 垂直起始位置 (范围：0-32000，单位：dot)。

*const char\* data*

[in] 图像数据。

**返回值：**

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.18. CPCL\_SetFontSize

此函数功能为设置字体大小。

```
int CPCL_SetFontSize(  
    void* handle,  
    int width,  
    int height  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int width*

[in] 设置字体宽度 (宽度放大倍数: 0-16)。

*int height*

[in] 设置字体高度 (高度放大倍数: 0-127)。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.19. CPCL\_SetDensity

此函数功能为设置打印浓度。

```
int CPCL_SetDensity(  
    void* handle,  
    int density  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int density*

[in] 打印浓度 (范围: 0-3)。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效

ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.20. CPCL\_SetSpeed

此函数功能为设置打印速度。

```
int CPCL_SetSpeed(
    void* handle,
    int speed
);
```

**参数:**

*void\* handle*

[in,out] 创建的目标打印机对象。

*int speed*

[in] 打印速度 (范围: 0-5)。

**返回值:**

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.21. CPCL\_SetTextSpacing

此函数功能为设置字符间距。

```
int CPCL_SetTextSpacing(
    void* handle,
    int spacing
);
```

**参数:**

*void\* handle*

[in,out] 创建的目标打印机对象。

*int spacing*

[in] 字符间距 (范围: 0-255)。

**返回值:**

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.22. CPCL\_SetLeftMargin

此函数功能为设置起始打印时左边距的值。(行模式下的接口)

```
int CPCL_SetLeftMargin(
    void* handle,
    int margin
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int margin*

[in] 左边距的值 (范围: 0-999)。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.23. CPCL\_SetTextBold

此函数功能为设置字体加粗程度。

```
int CPCL_SetTextBold(
    void* handle,
    int bold
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int bold*

[in] 字体加粗程度 (范围: 0-5)。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.24. CPCL\_SetTextUnderline

此函数功能为设置文本下划线。

```
int CPCL_SetTextUnderline(  
    void* handle,  
    int underline  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int underline*

[in] 下划线。

0: 关闭下划线,

1: 激活下划线

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.25. CPCL\_Abort

此函数功能为终止当前控制会话而不打印。

```
int CPCL_Abort(  
    void* handle  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.26. CPCL\_Print

此函数功能为打印标签。

```
int CPCL_Print(  
    void* handle  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.27. CPCL\_NextLabelPos

此函数功能为走纸到下一张标签。

```
int CPCL_NextLabelPos(  
    void* handle  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.28. CPCL\_PreFeed

此函数功能为在打印标签前走纸指定的距离。

```
int CPCL_PreFeed(  
    void* handle,  
    int distance  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int distance*

[in] 走纸距离 (范围: -4000-4000, 单位: dot)。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.29. CPCL\_PostFeed

此函数功能为在打印标签后走纸指定的距离。

```
int CPCL_PostFeed(  
    void* handle,  
    int distance  
);
```

参数:

*void\* handle*

[in,out] 创建的目标打印机对象。

*int distance*

[in] 走纸距离 (范围: -4000-4000, 单位: dot)。

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
其他值	其他值	Linux系统返回的错误码

## 4.30. CPCL\_GetPrinterStatus

此函数功能为获取打印机当前状态。

```
int CPCL_GetPrinterStatus (  
    void* handle,  
    int* status  
);
```

参数:

*void\* handle*

[in] 创建的目标打印机对象。

*int \* status*

[in,out]打印机当前状态。

状态	十进制值	位
正常	0	-
打印头被打开	1	0
卡纸	2	1
缺纸	4	2
缺碳带	8	3
打印暂停	16	4
打印中	32	5
上盖打开	64	6
其它错误	128	7

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_READ_FAILED	-11	读取失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时



其他值	其他值	Linux系统返回的错误码
-----	-----	---------------