
POSTEK PPCS

API 函数手册

G Series

条码标签打印机

Version 2.05

深圳市博思得通信发展有限公司

二〇〇六年

API 函数库文件说明

名称: CDFPSK.dll

版本编号: 2.X.X.X

版权所有: ©2004 深圳市博思得通信发展有限公司。保留所有权利。

用途

本 API 函数库为深圳市博思得通信发展有限公司条码标签打印机的用户提供一组命令, 为他们编写基于 Windows9X, NT, 2000, XP 等操作系统的应用程序提供便利。

本 API 函数库仅支持本公司产品。

缩略语对照

PPL I: 深圳市博思得通信发展有限公司的第一套打印机编程语言 (Printer Program Language I)。

API: 应用程序编程接口 (Application Program Interface)。

Dots: 像素 (pixel) 是一种计算机科学技术尺寸单位, 原指电视图像成像的最小单位, 在打印机领域表示打印机的最小打印成像单位: 1dot 等于一英寸除以打印机的最大分辨率。

- 对于 203DPI 的打印机来说, $1\text{dot} = 25.4\text{mm}/203 = 0.125\text{mm}$ ($1\text{dot} = 1000 / 203 = 5\text{mil}$);
- 对于 300DPI 的打印机来说, $1\text{dot} = 25.4\text{mm}/300 = 0.085\text{mm}$ ($1\text{dot} = 1000 / 300 = 3\text{mil}$)。

TrueType Font: 是基于 Windows 操作系统使用, 可装卸的字体。

- 已经安装的 TrueType Font, 都可以被本函数使用。

使用前须知

字符串

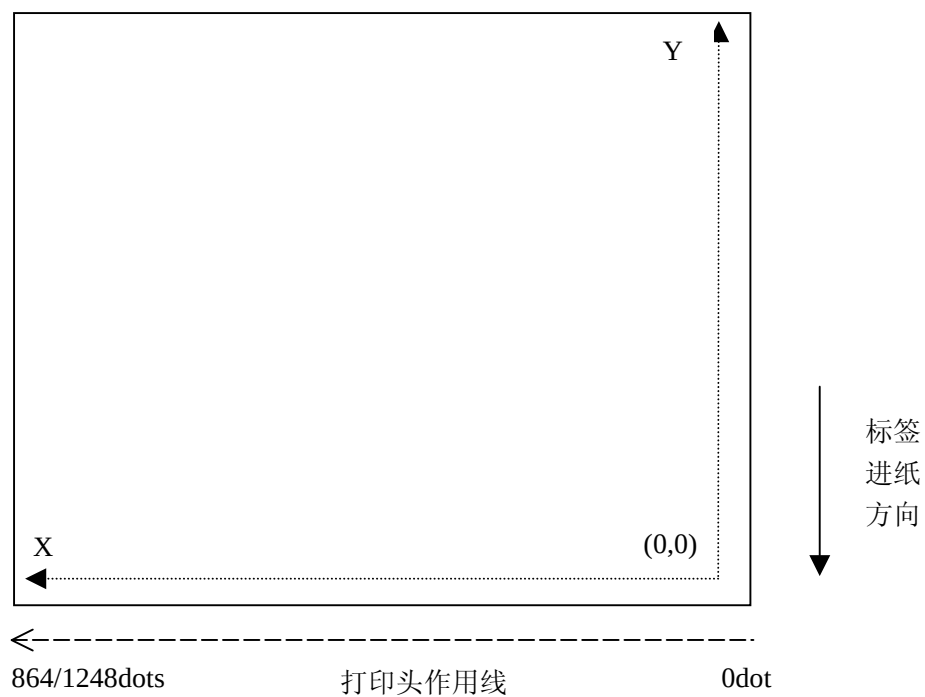
- * 字符串以双引号 (") 作为起始和结束标记;
- * 反斜杠 (\) 有以下作用:

字符	输入
"	\ "
\	\\
0x00 - 0x7F	\x00 - \x7F

-
- * 所有打印指令和名称均区分大小写。
 - * <CR>为 USASCII 码十进制的” 13” , 或十六进制的” 0DH” , 即” 回车” 符号。

条码标签打印机的坐标系统

如下图所示:



函数概述列表

函数名称	说明
<u>OpenPort</u>	打开通讯端口。
<u>ClosePort</u>	关闭使用 OpenPort 函数打开的通讯端口。
<u>SetPCComPort</u>	设置 PC 机上串口的传输波特率。
<u>GetErrState</u>	检测使用 CDFPSK. DLL 里的其它函数后是否有错误产生；
<u>PTK_GetInfo</u>	得到本 API 函数库的版本信息。
<u>PTK_DrawText</u>	打印一行文本文字，内容可以是常量、序列号、变量或组合字符串。
<u>PTK_DrawTextTrueTypeW</u>	打印一行 TrueType Font 文字，并且文字宽度和高度可以微调。
<u>PTK_DrawBarcode</u>	打印一个条码。
<u>PTK_DefineCounter</u>	定义一个序列号变量。
<u>PTK_SetDarkness</u>	设置打印头发热温度
<u>PTK_SoftFontList</u>	打印存储在 RAM 或 FLASH 存储器里的软字体的名称清单。
<u>PTK_SoftFontDel</u>	删除存储在 RAM 或 FLASH 存储器里的一个或所有的软字体。
<u>PTK_FormEnd</u>	结束存储表格(Form)，此函数与 PTK_FormDownload 配对使用。
<u>PTK_FormList</u>	打印存储在打印机里的表格名称清单。
<u>PTK_FormDel</u>	删除存储在打印机里的一个或所有的表格。
<u>PTK_ExecForm</u>	运行指定的表格。
<u>PTK_FormDownload</u>	存储一个表格到打印机；此命令与 PTK_FormEnd 函数配对使用。
<u>PTK_DrawPcxGraphics</u>	打印指定的图形。
<u>PTK_PcxGraphicsList</u>	打印已存储在打印机 RAM 或 FLASH 存储器里的图形名称清单。
<u>PTK_PcxGraphicsDel</u>	删除存储在打印机里的一个或所有图形。
<u>PTK_PcxGraphicsDownload</u>	存储一个 PCX 格式的图形到打印机。
<u>PTK_PrintPCX</u>	打印一个 PCX 格式的图形。
<u>PTK_DrawBinGraphics</u>	打印二进制格式的图形。
<u>PTK_DisableBackFeed</u>	取消打印回转功能。
<u>PTK_EnableBackFeed</u>	设置打印回转功能。
<u>PTK_DrawLineXor</u>	画直线(两直线相交处作"异或"处理)。
<u>PTK_DrawLineOr</u>	画直线(两直线相交处作"或"处理)。
<u>PTK_DrawDiagonal</u>	画斜线。
<u>PTK_DrawWhiteLine</u>	画白色直线。
<u>PTK_ClearBuffer</u>	清除打印机缓冲内存的内容。
<u>PTK_SetPrinterState</u>	设置打印机的工作状态。

<u>PTK_PrintLabel</u>	命令打印机执行打印工作。
<u>PTK_PrintLabelAuto</u>	自动执行打印工作。
<u>PTK_SetLabelHeight</u>	设置标签的高度和定位间隙\黑线\穿孔的高度。
<u>PTK_SetLabelWidth</u>	设置标签的宽度。
<u>PTK_SetCoordinateOrigin</u>	设置/改变坐标原点。
<u>PTK_SetPrintSpeed</u>	设置打印速度。
<u>PTK_PrintConfiguration</u>	打印机器当前的设置/工作状态。
<u>PTK_DisableErrorReport</u>	取消错误反馈。
<u>PTK_EnableErrorReport</u>	设置错误反馈。
<u>PTK_DefineVariable</u>	定义变量。
<u>PTK_DrawRectangle</u>	画矩形。
<u>PTK_SetDirection</u>	设置标签打印方向。
<u>PTK_EnableFLASH</u>	选择 FLASH 存储器。
<u>PTK_DisableFLASH</u>	取消选择 FLASH 存储器。
<u>PTK_Download</u>	下载变量或系列号变量。
<u>*PTK_Reset</u>	将打印机复位。
<u>*PTK_FeedBack</u>	要求打印机立刻反馈错误报告。
<u>PTK_CutPage</u>	设置切刀的工作周期（即每打印多少页标签后，切刀才切一次纸）。
<u>PTK_BinGraphicsList</u>	打印已存储在打印机 RAM 或 FLASH 存储器里的图形名称清单（包括 Bin 格式和 PCX 格式的图形）。
<u>PTK_BinGraphicsDel</u>	删除已存储在打印机里的一个或所有图形（此图形可是 Bin 格式或 PCX 格式的）。
<u>PTK_BinGraphicsDownload</u>	存储一个 Bin 格式的图形到打印机。
<u>PTK_RecallBinGraphics</u>	打印一个已保存在打印机里的 Bin 格式图形。
<u>PTK_ErrorReport</u>	发送错误查询指令到打印机并且指定串口接收和分析打印机当前错误代码。
<u>PTK_UserFeed</u>	令打印机进纸给定的长度。
<u>PTK_UserBackFeed</u>	令打印机退纸给定的长度。

* 标志的此函数暂时不能使用。

函数详细说明

OpenPort

说明:

OpenPort 函数的作用是打开通讯端口。

使用本函数库其它函数之前，必须首先正确执行 **OpenPort** 函数。

原型:

```
int OpenPort(LPTSTR xxxx);
```

参数:

xxxx: 当前所使用的打印机在WINDOWS下的名称;

返回值:

0 -> OK;

其它返回值请参考章节: **WINPSK.dll 错误返回值解析**。

范例:

```
int return = OpenPort("POSTEK G-2108");           //表示打开 G-2108 当前所选用的打印端口。
```

ClosePort

说明:

ClosePort函数的作用是关闭使用**OpenPort**函数打开的通讯端口。

用户在对打印机操作完成之后，建议调用**ClosePort**关闭通讯端口;

否则用户的程序一直占用打开的通讯端口，直到程序被关闭。

原型:

```
int ClosePort(void);
```

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节：**CDFPSK.dll 错误返回值解析**。

范例：

```
ClosePort( );
```

SetPCComPort

说明：

SetPCComPort 函数的作用是设置 PC 机上串口的传输波特率。

这个函数只有在使用串口进行通讯时才有效。

注意： 必须对应打印机上所选择串口波特率（通过调整 DIP 开关的 7，8PIN，请参阅用户手册）

原型：

```
int SetPCComPort (DWORD BaudRate, BOOL HandShake);
```

参数：

BaudRate: 要设置的串口波特率，可取值：
9600, 19200, 38400, 57600;

HandShake: 是否使用硬件握手 (HandShaking);
TRUE: 硬件握手 (HandShaking) 有效,
FALSE: 硬件握手 (HandShaking) 无效。

返回值：

0 -> OK;

其它返回值请参考章节：**CDFPSK.dll 错误返回值解析**。

范例：

```
SetPCComPort ( 9600, TRUE);
```

GetErrState

说明：

GetErrState 函数的作用是检测使用 CDFPSK.DLL 里的其它函数后是否有错误产生；
错误代码请参阅 “CDFPSK.dll 错误返回值解析”

这个函数必须在 ClosePort () 函数前使用！

原型：

```
int GetErrState(void);
```

参数： 无

返回值：

0 -> OK;

其它返回值请参考章节： **CDFPSK.dll 错误返回值解析**。

示范：

```
int state = 0;
OpenPort(1);
...
state = GetErrState();
...
ClosePort();
```

PTK_GetInfo

说明：

PTK_GetInfo 函数作用是得到本 API 函数库的版本信息。

原型：

int PTK_GetInfo(void)

参数： 无

返回值： 0 -> OK;

其它返回值请参考章节： **CDFPSK.dll 错误返回值解析**。

范例：

```
PTK_GetInfo(void)
```

PTK_DrawText

说明：

PTK_DrawText 函数作用是打印一行文本文字，内容可以是常量、序列号、变量或组合字符串。

原型：

int PTK_DrawText (unsigned int px, unsigned int py,

```
unsigned int  pdirec, unsigned int  pFont,
unsigned int  pHorizontal, unsigned int  pVertical,
char ptext, LPTSTR pstr );
```

参数:

px: 设置 X 坐标, 以点 (dots) 为单位.
py: 设置 Y 坐标, 以点 (dots) 为单位.
pdirec: 选择文字的打印方向. 0—不旋转;1—旋转 90° ; 2—旋转 180° ; 3—旋转 270° .
pFont: 选择内置字体或软字体. 1—5: 为打印机内置字体; ‘A’ — ‘Z’ : 为下载的软字体.
[a 为打印机内置 24*24 简体汉字.](#)

取值	描述
1	西文字体1
2	西文字体2
3	西文字体3
4	西文字体4
5	西文字体5
‘a’	24点阵中文宋体
‘A’~‘Z’	软字体

pHorizontal: 设置点阵水平放大系数. 可选择:1—24.
pVertical: 设置点阵垂直放大系数. 可选择:1—24.
ptext: 选 ‘N’ 则打印正常文本 (如黑字白底文本),
选 ‘R’ 则打印文本反色文本 (如白字黑底文本).

pstr: 一个长度为 1-100 的字符串. 用户可以用 “DATA” , Cn, Vn 自由排列组合成一个组合字符串,
“DATA” : 常量字符串, 必须用 “” 作为起始和结束符号, 如 “POSTEK Printer”.
Cn: 序列号数值, 此序列号必须已经定义, 请参考 C 命令.
Vn: 变量字符串, 此变量字符串必须已经定义, 请参考 V 命令.
如: `“data1” CnVn “data2” .`

返回值: 0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DrawText (50,30,0,2,1,1,'N','\123456789\");
PTK_DrawText (50,30,0,2,1,1,'N',C1);
PTK_DrawText (50,30,0,2,1,1,'N',V3);
PTK_DrawText (50,30,0,2,1,1,'N','Printer" C2V1 "is ok. "');
```

PTK_DrawBarcode

说明:

PTK_DrawBarcode 函数作用是打印一个条码。

原型:

```
int PTK_DrawBarcode ( unsigned int px, unsigned int py,
                      unsigned int pdirec, LPTSTR pCode,
                      unsigned int NarrowWidth, unsigned int pHorizontal,
                      unsigned int pVertical, char ptext, LPTSTR pstr );
```

参数:

- px: 设置 X 坐标, 以点(dots)为单位.
- py: 设置 Y 坐标, 以点(dots)为单位.
- pdirec: 选择条码的打印方向. 0—不旋转; 1—旋转 90° ; 2—旋转 180° ; 3—旋转 270° .
- pCode: 选择要打印的条码类型.

P4 值	条码类型
0	Code 128 UCC (shipping container code)
1	Code 128 AUTO
1A	Code 128 subset A
1B	Code 128 subset B
1C	Code 128 subset C
1E	UCC/EAN
1F	EAN 128 subset A
1G	EAN 128 subset B
1H	EAN 128 subset C
2D	Interleaved 2 of 5 with human readable check digit
2G	German Postcode
2M	Matrix 2 of 5
2U	UPC Interleaved 2 of 5
3	Code 3 of 9
3C	Code 3 of 9 with check sum digit
3E	Extended Code 3 of 9
3F	Extended Code 3 of 9 with check sum digit
9	Code93
E30	EAN-13
E32	EAN-13 2 digit add-on
E35	EAN-13 5 digit add-on
E80	EAN-8
E82	EAN-8 2 digit add-on
E-85	EAN-8 5 digit add-on
K	Codabar
P	Postnet

UA0	UPC-A
UA2	UPC-A 2 digit add-on
UA5	UPC-A 5 digit add-on
UE0	UPC-E
UE2	UPC-E 2 digit add-on
UE5	UPC-E 5 digit add-on

NarrowWidth: 设置条码中窄单元的宽度, 以点(dots)为单位.

pHorizontal: 设置条码中宽单元的宽度, 以点(dots)为单位.

pVertical: 设置条码高度, 以点(dots)为单位.

ptext: 选' N' 则不打印条码下面的人可识别文字,
选' B' 则打印条码下面的人可识别文字.

pstr: 一个长度为 1-100 的字符串. 用户可以用" DATA", Cn, Vn 自由排列组合成一个组合字符串,
"DATA": 常量字符串, 必须用"'" 作为起始和结束符号, 如 "POSTEK Printer".
Cn: 序列号数值, 此序列号必须已经定义, 请参考 C 命令。
Vn: 变量字符串, 此变量字符串必须已经定义, 请参考 V 命令。
如: `"data1" CnVn "data2"`.

返回值: 0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DrawBarcode (50,30,0,'1A',1,1,10,'N',"123456");  
PTK_DrawBarcode (50,30,0,'1A',1,1,10,'N',C2);  
PTK_DrawBarcode (50,30,0,'1A',1,1,10,'N',V1);  
PTK_DrawBarcode (50,30,0,'1A',1,1,10,'N',C1" is "V2);
```

PTK_DrawBar2D_DATAMATRIX

说明:
PTK_DrawBar2D_DATAMATRIX 函数作用是打印一个 **DataMatrix** 二维条码。

原型:

```
PTK_DrawBar2D_DATAMATRIX( unsigned int x, unsigned int y,  
                           unsigned int w, unsigned int v,  
                           unsigned int o, unsigned int m,  
                           LPTSTR pstr )
```

***参数:**

int	x;	● X 座标。
int	y;	● Y 座标。备注: 1 dot = 0.125 mm。
int	w;	● 最大列印宽度, 单位 dots。
int	v;	● 最大列印高度, 单位 dots。
int	o;	● 设置旋转方向, 范围: 0~3。
int	m;	● 设置放大倍数, 以点(dots)为单位, 范围值: (1 - 9)。

LPCTSTR pstr; ●资料字串。

传回值: 0 -> OK.
 Reference Error.txt file.

PTK_DrawBar2D_QR

功能: 打印一个 QR 条码

```
PTK_DrawBar2D_QR( unsigned int x, unsigned int y,
                  unsigned int w, unsigned int v,
                  unsigned int o, unsigned int r,
                  unsigned int m, unsigned int g,
                  unsigned int s, LPCTSTR pstr )
```

*参数: int x; ●X 座标。
 int y; ●Y 座标。备注: 1 dot = 0.125 mm。
 int w; ●最大列印宽度, 单位 dots。
 int v; ●最大列印高度, 单位 dots。
 int o; ●设置旋转方向, 范围: 0~3。
 int r; ●设置放大倍数, 以点(dots)为单位, 范围值: (1 - 9)。
 int m; ●QR 码编码模式选择, 范围值(0 - 4)。
 int g; ●QR 码纠错等级选择, 范围值(0 - 3)。
 int s; ●QR 码掩模图形选择, 范围值(0 - 8)。
 LPCTSTR pstr; ●资料字串。

传回值: 0 -> OK.
 Reference Error.txt file.

PTK_DrawBar2D_MaxiCode

说明:

PTK_DrawBar2D_MaxiCode 函数作用是打印一个 MaxiCode 条码。

```
PTK_DrawBar2D_MaxiCode( unsigned int x, unsigned int y,
                        unsigned int m, unsigned int u,
                        LPCTSTR pstr )
```

*参数: int x; ●X 座标。
 int y; ●Y 座标。备注: 1 dot = 0.125 mm。
 int m; ●Mode [2 - 4];
 int u; ●是否是 UPS 格式
 LPCTSTR pstr; ●资料字串。

传回值: 0 -> OK.
Reference Error.txt file.

PTK_DrawBar2D_Pdf417

说明:

PTK_DrawBar2D_Pdf417 函数作用是打印一个 **PDF417** 二维条码。

原型:

```
int PTK_DrawBar2D_Pdf417(unsigned int x, unsigned int y,
                          unsigned int w, unsigned int v,
                          unsigned int s, unsigned int c,
                          unsigned int px, unsigned int py,
                          unsigned int r, unsigned int l,
                          unsigned int t, unsigned int o,
                          LPTSTR pstr)
```

参数:

unsigned int	x;	X 座标。
unsigned int	y;	Y 座标。备注: 1 dot = 0.125 mm。
unsigned int	w;	最大列印宽度, 单位 dots。
unsigned int	v;	最大列印高度, 单位 dots。
unsigned int	s;	错误校正等级, 范围: 0~8。
unsigned int	c;	资料压缩等级, 范围: 0 或 1。
unsigned int	px;	模组宽度, 范围: 2~9 dots。
unsigned int	py;	模组高度, 范围: 4~99 dots。
unsigned int	r;	最大 row count。
unsigned int	l;	最大 column count。
unsigned int	t;	Truncation flag, '0' 是 normal 和 '1' 是 truncated.
unsigned int	o;	列印方向定位, '0' 是 0°, '1' 是 90°、 '2' 是 180°, '3' 是 270°
LPCTSTR	pstr;	资料字串。

传回值: 0 -> OK.
Reference Error.txt file.

范例: unsigned int x, y, w, v, s, c, px, py, r, l, t, o;
LPCTSTR pstr = "POSTEKINFO";
x=10;y=10;w=400;v=300;s=0;c=0;px=3;py=7;r=10;l=2;t=0;o=0;
PTK_DrawBar2D_Pdf417 (x, y, w, v, s, c, px, py, r, l, t, o, pstr);

PTK_DrawTextTrueTypeW

注意:

必须先正确安装 **POSTEK G-2108** 或 **POSTEK G-3106** 的 **Windows** 打印机驱动程序, 才能使用这个 **PTK_DrawTextTrueTypeW ()** 函数。

说明:

PTK_DrawTextTrueTypeW 作用是打印一行 TrueType Font 文字, 并且文字宽度和高度可以微调。

原型:

```
int PTK_DrawTextTrueTypeW( int x, int y,
                           int FHeight, int FWidth,
                           LPCTSTR FType, int Fspin,
                           int FWeight, BOOL FItalic,
                           BOOL FUnline, BOOL FStrikeOut,
                           LPCTSTR id_name, LPCTSTR data )
```

参数:

x: 设置 X 坐标, 以点(dots)为单位;

y: 设置 Y 坐标, 以点(dots)为单位;

FHeight: 字型高度, 以点(dots)为单位;

FWidth: 字型宽度, 以点(dots)为单位;

*** 如果想打印正常比例的字体, 需将 FWidth 设置为 0;**

FType: 字型名称;

Fspin: 字体旋转角度;

1 -> 0, 2 -> 90, 3 -> 180, 4 -> 270

Fweight: 字体粗细。

0 and 400 -> 400 标准、

100 -> 非常细、200 -> 极细、

300 -> 细 、500 -> 中等、

600 -> 半粗 、700 -> 粗 、

800 -> 特粗 、900 -> 黑体。

Fitalic: 斜体, 0 -> FALSE、1 -> TRUE;

Funline: 文字加底线, 0 -> FALSE、1 -> TRUE;

FstrikeOut: 文字加删除线, 0 -> FALSE、1 -> TRUE;

id_name: 识别名称, 因为一行 TrueType 文字将被转换成 PCX 格式数据以 id_name 作为 PCX 格式图形的名称存放到打印机内, 在关机前都可以多次通过 PTK_DrawPcxGraphics()调用 id_name 打印这行文字;

data: 字符串内容。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.d11 错误返回值解析**。

范例：打印 3mm 高度的汉字：

203DPI 打印机需将 FHeight 设置为 $3 / 0.125 = 24$ 个点；

300DPI 打印机需将 FHeight 设置为 $3 / 0.08 = 38$ 个点(四舍五入)。

```
PTK_DrawTextTrueTypeW (30, 35, 24, 0, "宋体", 4, 400, 0, 0, 0, "A1", "机要绝密");
```

PTK_DefineCounter

说明：

PTK_DefineCounter 函数作用是定义一个序列号变量。

原型：

```
int PTK_DefineCounter ( unsigned int  id, unsigned int  maxNum,  
                        char  ptext, LPTSTR  pstr, LPTSTR  pMsg );
```

参数：

id： 系列号 ID，取值范围：0—9；

maxNum： 序列号最大数字位数；取值范围：1—40；

ptext： 对齐方式；L—左对齐，R—右对齐，C—居中，N—不对齐；

Pstr： 序列号的变化规律；由“+”或“-”加上一个数字，再加上一个变化标志（D - 十进制，B - 二进制，O - 八进制，H - 十六进制，X-自定义模式，允许用户设置最多64个字符）组成：

“+1”=每次增加 1，默认按照十进制计算：如 1234, 1235, 1236, ...；

“+3D”=每次增加 3，按照十进制计算，同上；

“-1B”=每次减少 1，按照二进制计算：如 1111, 1110, 1101, ...；

“-4O”=每次减少 4，按照八进制计算：如 1234, 1230, 1224, ...；

“-6H”=每次减少 6，按照十六进制计算：如 1234, 122E, 1228, ...；

“+3X”=如变化规律表内容为：TE2DOKLU046MNY37，起始值是“T062”，
则 T062, T06K, T06O, ...；

pMsg：提示信息字符串；可在打印机 LCD 上或可编程键盘(KDU)的显示屏上显示。

返回值： 0 -> OK；

其它返回值请参考章节：**CDFPSK.d11 错误返回值解析**。

范例：

```
PTK_DefineCounter (0, 6, ' N' , "+1", "\ Enter\" Code:");
```

PTK_SetDarkness

说明:

PTK_SetDarkness 函数的作用是设置打印头发热温度。

原型:

```
int PTK_SetDarkness (unsigned int id);
```

参数:

id: 取值范围: 0—20, 缺省为 10;

此值并不是真正意义的温度数值, 而是相对数值, 0 表示打印头工作在最低发热状态, 20 表示打印工作在最高发热状态.

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_SetDarkness (10);
```

PTK_SoftFontList

说明:

PTK_SoftFontList 函数的作用是打印存储在 RAM 或 FLASH 存储器里的软字体的名称清单。

原型:

```
int PTK_SoftFontList (void);
```

参数: 无

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_SoftFontList ();
```

PTK_SoftFontDel

说明:

PTK_SoftFontDel 函数作用是删除存储在 RAM 或 FLASH 存储器里的一个或所有的软字体。

原型:

```
int PTK_SoftFontDel (char pid);
```

参数:

pid: 软字体 ID, 取值范围: A—Z 或 * ;

如果 pid = ‘*’, 打印机将删除存储在 RAM 或 FLASH 存储器里所有的软字体.

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_SoftFontDel ('A');
```

PTK_FormEnd

说明:

PTK_FormEnd 函数:作用是结束存储表格(Form), 此函数与 PTK_FormDownload 配对使用。

原型:

```
int PTK_FormEnd (void );
```

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_FormDownload ("Form1") ;  
...  
PTK_FormEnd ( );
```

PTK_FormList

说明:

PTK_FormList 函数作用是打印存储在打印机里的表格名称清单。

原型:

```
int PTK_FormList (void );
```

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_FormList ( );
```

PTK_FormDel

说明:

PTK_FormDel 函数是删除存储在打印机里的一个或所有的表格。

原型:

```
int PTK_FormDel (LPTSTR pid);
```

参数:

pid: 即将删除的软字体的名称, 最大长度为 16 个字符;

如果 pid = “*”, 打印机将删除存储在 RAM 或 FLASH 存储器里所有的表格。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_FormDel (“FORMNAME”);
```

PTK_ExecForm

说明:

PTK_ExecForm 函数的作用是运行指定的表格。

原型:

int PTK_ExecForm (LPTSTR pid);

参数:

pid: 即将运行的表格的名称, 最大长度为 16 个字符。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_ExecForm ("FORM1" );
```

PTK_FormDownload

说明:

PTK_FormDownload 函数的作用是存储一个表格到打印机; 此命令与 **PTK_FormEnd** 函数配对使用; 如果在 **EnableFLASH** () 函数后使用, 表格的内容则存储到 FLASH 存储器; 如果在默认状态下或在 **DisableFLASH** () 函数后使用, 表格的内容则存储到 RAM 存储器。

原型:

int PTK_FormDownload (LPTSTR pid);

参数:

pid: 自定义的表格名称, 最大长度为 16 个字符; 此表格内容存储到打印机后, 用户必须使用才能运行它。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_FormDownload ("FORMNAME" );
```

PTK_DrawPcxGraphics

说明:

PTK_DrawPcxGraphics 函数作用是打印指定的图形。

注: 被打印的图形必须预先使用 PTK_PcxGraphicsDownload () 存储到打印机里。

原型:

int PTK_DrawPcxGraphics (unsigned int px, unsigned int py, LPTSTR gname);

参数:

px: 设置 X 坐标;以点(dots)为单位;

py: 设置 Y 坐标;以点(dots)为单位;

game: 即将打印的图形名称, 最大长度为 16 个字符, 必须是在 PTK_PcxGraphicsDownload () 中自定义的图形名称。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

PTK_DrawPcxGraphics (100, 50, "PCX1");

PTK_PcxGraphicsList

说明:

PTK_PcxGraphicsList 函数的作用是打印已存储在打印机 RAM 或 FLASH 存储器里的图形名称清单。

原型:

int PTK_PcxGraphicsList (void);

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

PTK_PcxGraphicsList ();

PTK_PcxGraphicsDel

说明:

PTK_PcxGraphicsDel 函数作用是删除存储在打印机里的一个或所有图形。

原型:

int PTK_PcxGraphicsDel (LPTSTR pid);

参数:

pid: 即将删除的图形名称, 最大长度为 16 个字符;
如果 pid = “*”, 则将删除所有存储在 RAM 或 FLASH 存储器里的图形。

返回值: 0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_PcxGraphics Del ( “PCX2” );
```

PTK_PcxGraphicsDownload

说明:

PTK_PcxGraphicsDownload 函数的作用是存储一个 PCX 格式的图形到打印机。

原型:

```
int PTK_PcxGraphicsDownload (char* pcxname, char* pcxpath);
```

参数:

pcxname: 自定义图形的名称, 最大长度为 16 个字符; 当图形存储到打印机后, 用户在
PTK_DrawPcxGraphics () 中使用此名称才能将图形读取出来打印。
pcxpath: PCX 图形文件在 PC 机存储器里的路径;

返回值: 0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_PcxGraphicsDownload ( “PCXA”, “c:\\test1111.pcx” );
```

PTK_PrintPCX

说明:

PTK_PrintPCX 函数是打印一个 PCX 格式的图形。

这个函数将 **PTK_PcxGraphicsDownload ()** 和 **PTK_DrawPcxGraphics ()** 组合封装到一起使用。

原型:

```
int PTK_PrintPCX (unsigned int px, unsigned int py, char* filename);
```

参数:

px: 设置 X 坐标;以点(dots)为单位;
py: 设置 Y 坐标;以点(dots)为单位;
filename: PCX 图形文件名称, 可包含文件路径。
格式如: “XXXXXXXX.XXX” 或 “X:\\XXX\\XXX.PCX”。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_PrintPCX(10, 100, "c:\\phone.pcx");
```

PTK_DrawBinGraphics

说明:

PTK_DrawBinGraphics 函数的作用是打印二进制格式的图形。

二进制格式图形是不压缩的图形数据;每一个比特(bit)表示一个点;比特值为 0 时此点将打印, 为 1 时此点不打印。

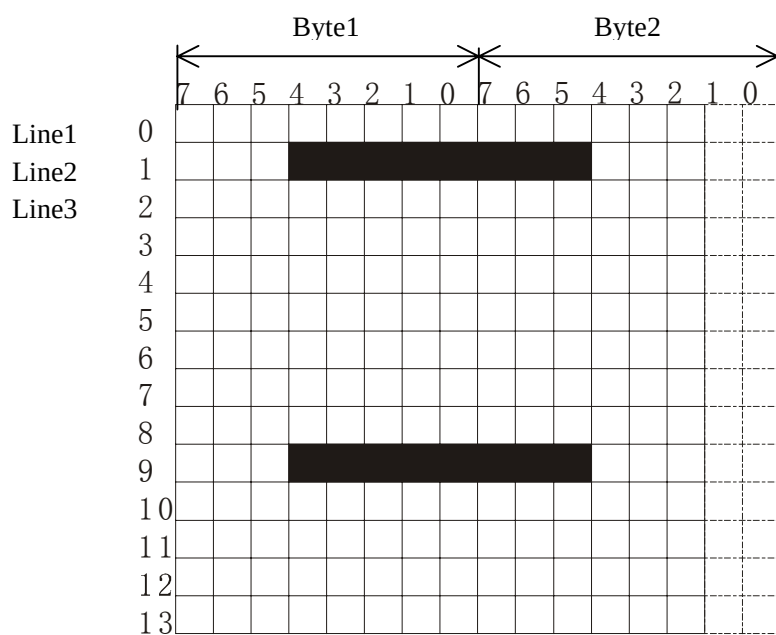
原型:

```
int PTK_DrawBinGraphics ( unsigned int px, unsigned int py,  
                          unsigned int pbyte, unsigned int pH,  
                          UCHAR* Gdata );
```

参数:

px: 设置 X 坐标, 以点(dots)为单位;
py: 设置 Y 坐标, 以点(dots)为单位;
pbyte: 一行数据的字节数(1Byte = 8bits); 如果一行数据的点数不能整除 8, 则其字节数应该等于商加上 1; 如: 一行是 14bit 的数据的字节数是 2;
Ph: 图形的高度, 以点(dots)为单位;
Gdata([...raster data...]): 二进制图形数据, 数据量大小= pbyte * pH (Bytes)。**Bit 值为 0 是打印内容, 为 1 是空白内容。**
二进制数据传输顺序是从左到右, 从上到下, 以下图为例:
数据传输顺序为: Line1 的 Byte1 (0xff), Line1 的 Byte2 (0xff), Line2 的 Byte1 (0xe0), Line2 的 Byte2 (0x1f), Line3 的 Byte1 (0xff), Line3 的 Byte2 (0xff), ...

其中虚线部分是非图形区域，对应它们的 bit 值为 1。



返回值: 0 → OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
char buf[] = {0xff, 0xff, 0xe0, 0x1f, 0xff, 0xff...};
PTK_DrawBinGraphics (20, 30, 4, 14, buf);
```

PTK_DisableBackFeed

说明:

PTK_DisableBackFeed 函数作用是取消打印回转功能。

原型:

```
int PTK_DisableBackFeed(void);
```

参数: 无

返回值: 0 → OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DisableBackFeed ( );
```

PTK_EnableBackFeed

说明:

PTK_EnableBackFeed 函数作用是设置打印回转功能。

原型:

```
int PTK_EnableBackFeed (unsigned int distance);
```

参数:

distance: 回转距离,以点(dots)为单位。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_EnableBackFeed (140);
```

PTK_DrawLineXor

说明:

PTK_DrawLineXor 函数作用是画直线(两直线相交处作”异或”处理)。

原型:

```
int PTK_DrawLineXor (unsigned int px, unsigned int py,  
                     unsigned int pbyte, unsigned int pH);
```

参数:

px: X 坐标, 以点(dots)为单位;

py: Y 坐标, 以点(dots)为单位;

pbyte: 设置直线的水平长度, 以点(dots)为单位;

pH: 设置直线的垂直高度, 以点(dots)为单位。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DrawLineXor (100,20,5,110);
```

PTK_DrawLineOr

说明:

PTK_DrawLineOr 函数作用是画直线(两直线相交处作”或”处理)。

原型:

```
int PTK_DrawLineOr (unsigned int  px, unsigned int  py,  
                    unsigned int  plength, unsigned int  pH);
```

参数:

px: 设置 X 坐标, 以点(dots)为单位;
py: 设置 Y 坐标, 以点(dots)为单位;
plength: 设置直线的水平长度, 以点(dots)为单位;
pH: 设置直线的垂直高度, 以点(dots)为单位。

返回值: 0 → OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DrawLineOr (100,20,5,110);
```

PTK_DrawDiagonal

说明:

PTK_DrawDiagonal 函数的作用是画斜线。

原型:

```
int PTK_DrawDiagonal (unsigned int  px, unsigned int  py,  
                      unsigned int  thickness, unsigned int  pEx,  
                      unsigned int  pEy);
```

参数:

px: 设置斜线起始 X 坐标, 以点(dots)为单位;
py: 设置斜线起始 Y 坐标, 以点(dots)为单位;
thickness: 设置斜线粗细, 以点(dots)为单位;

pEx: 设置斜线终止 X 坐标, 以点(dots)为单位;
pEy: 设置斜线终止 Y 坐标, 以点(dots)为单位。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DrawDiagonal (50, 30, 10, 100, 80);
```

PTK_DrawWhiteLine

说明:

PTK_DrawWhiteLine函数的作用是画白色直线。

原型:

```
int PTK_DrawWhiteLine (unsigned int px, unsigned int py,  
                        unsigned int plength, unsigned int pH);
```

参数:

px: 设置 X 坐标, 以点(dots)为单位;
py: 设置 Y 坐标, 以点(dots)为单位;
plength: 设置直线的水平长度, 以点(dots)为单位;
pH: 设置直线的垂直高度, 以点(dots)为单位。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DrawWhiteLine (100, 20, 5, 110);
```

PTK_ClearBuffer

说明:

PTK_ClearBuffer 函数的作用是清除打印机缓冲内存的内容。

当发送新的一张标签内容到打印机前, 建议使用此命令先清空打印机图形缓存里已有的数据内容。
请不要在 FORM 的编排过程中使用此函数。

原型:

```
int PTK_ClearBuffer (void);
```

参数：无

返回值：0 -> OK;

其它返回值请参考章节：**CDFPSK.dll 错误返回值解析**。

范例：

```
PTK_ClearBuffer ( );
```

PTK_SetPrinterState

说明：

PTK_SetPrinterState函数的作用是设置打印机的工作状态。

原型：

```
int PTK_SetPrinterState (char state);
```

参数：

state 为以下几种字符：

D：设置打印机为热感印(热传导)状态；

P：设置打印机为连续送纸状态(缺省)；

L：设置打印机为打印一张标签后，暂停等待用户确定再打印下一张标签；
(确定方式：1. 按” FEED” 键；2. 在安装剥纸器情况下，当用户取走标签后自动打印下一张标签)

C：设置打印机为安装切纸刀状态；

N：设置打印机为安装剥纸器状态。

注意：

1. 切纸刀与剥纸器不能同时安装；
2. 如果打印机状态设置不正确时，打印机前面板的 READY 指示灯将闪烁，请参考打印机用户手册的故障排除章节。

返回值：0 -> OK;

其它返回值请参考章节：**CDFPSK.dll 错误返回值解析**。

范例：

```
PTK_SetPrinterState ('D');
```

PTK_PrintLabel

说明:

PTK_PrintLabel 函数的作用是命令打印机执行打印工作。

注: 此函数不能在 FORM 的编排过程中, 而用 **PTK_PrintLabelAuto ()** 函数代替。

原型:

```
int PTK_PrintLabel (unsigned int number, unsigned int cpnumber);
```

参数:

number: 打印标签的数量, 取值范围: 1—65535;

cpnumber: 每张标签的复制份数, 取值范围: 1—65535;

如果 cpnumber 没有设置, 那么默认为 1。

返回值: 0 → OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_PrintLabel (2,3);
```

PTK_PrintLabelAuto

说明:

PTK_PrintLabelAuto 函数的作用是自动执行打印工作。

当 FORM 中存在变量或者序列号时, 建议使用此函数, 当用户输入全部的变量内容, 打印机将立刻开始打印标签。

注: 只能在 **FORM** 里使用。

原型:

```
int PTK_PrintLabelAuto (unsigned int number, unsigned int cpnumber);
```

参数:

number: 打印标签的数量, 取值范围: 1—65535;

cpnumber: 每张标签的复制份数, 取值范围: 1—65535;

如果 cpnumber 没有设置, 那么默认为 1。

返回值: 0 → OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_PrintLabelAuto (2,3);
```

PTK_SetLabelHeight

说明:

PTK_SetLabelHeight 函数的作用是设置标签的高度和定位间隙\黑线\穿孔的高度。

原型:

```
int PTK_SetLabelHeight (unsigned int lheight, unsigned int gapH);
```

参数:

lheight: 标签的高度, 以点(dots)为单位, 取值范围: 0-65535;

gapH: 标签间的定位间隙/黑线/穿孔的高度, 以点(dots)为单位,
取值范围: 0-65535;

gapH 的取值与标签定位方式相关:

间隙模式 (GAP MODE): 缺省模式, gapH 设置为间隙的高度,

穿孔定位属于间隙模式的特例;

黑线模式 (BLACK LINE MODE): gapH 设置为黑线的高度;

连续纸模式 (CONTINUOUS MODE): gapH 设置为 0. 这时候, 纸张探测器只检测纸张是否用尽。

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_SetLabelHeight (160, 24);
```

PTK_SetLabelWidth

说明:

PTK_SetLabelWidth 函数的作用是设置标签的宽度。

原型:

```
int PTK_SetLabelWidth (unsigned int lwidth);
```

参数:

lwidth: 标签的宽度, 以点(dots)为单位。

返回值: 0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:
PTK_SetLabelWidth (250);

PTK_SetCoordinateOrigin

说明:
PTK_SetCoordinateOrigin 函数的作用是设置/改变坐标原点。

原型:
int PTK_SetCoordinateOrigin (unsigned int px, unsigned int py);

参数:
px: X 坐标移动的距离, 确以点(dots)为单位;
Py: Y 坐标移动的距离, 确以点(dots)为单位。

返回值:
0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:
PTK_SetCoordinateOrigin (12,23);

PTK_SetPrintSpeed

说明:
PTK_SetPrintSpeed 函数的作用是设置打印速度。

原型:
int PTK_SetPrintSpeed (unsigned int px);

参数:
px: 取值范围为0 - 6, 或者10 - 80。

p1值	速度	PPLB(compatible)
10	1.0 ips (25 mm/s)	0 or 1

15	1.5 ips (37 mm/s)	
20	2.0 ips (50 mm/s)	2
25	2.5 ips (63 mm/s)	
30	3.0 ips (75 mm/s)	3
35	3.5 ips (83 mm/s)	
40	4.0 ips (100 mm/s)	4
50	5.0 ips (125 mm/s)	5
60	6.0 ips (150 mm/s)	6
70	7.0 ips (175 mm/s)	
80	8.0 ips (200 mm/s)	

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例: PTK_SetPrintSpeed (5);

PTK_PrintConfiguration

说明:

PTK_PrintConfiguration 函数的作用是打印机器当前的设置/工作状态。

原型:

```
int PTK_PrintConfiguration ( );
```

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_PrintConfiguration ( );
```

PTK_DisableErrorReport

说明:

PTK_DisableErrorReport 函数的作用是取消错误反馈。

原型:

int PTK_DisableErrorReport(void);

参数: 无

返回值:

0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

PTK_DisableErrorReport();

PTK_EnableErrorReport

说明:

PTK_EnableErrorReport函数的作用是设置错误反馈 。

打印机的反馈数据从 RS232 串口返回电脑.

如果打印中发生错误, 打印机将先发送一个 NACK (15H) 字符回电脑, 跟着发送出错编号.
如果没有错误发生, 打印机将在接收到 P 命令后发送 ACK (06H) 字符.

错误代码	说明
0x00	No Error
0x01	Object Exceeded Label Border
0x02	Bar Code Data Length Error
0x03	Insufficient Memory to Store Data
0x04	Memory Configuration Error
0x05	RS-232 Interface Error
0x06	Paper or Ribbon Empty
0x07	Duplicate Name: Form, Graphic or Soft Font
0x08	Name Not Found: Form, Graphic or Soft Font
0x09	Not in Data Entry Mode
0x0a	Print Head Up (Open)
0x0b	Pause Mode or Paused in Peel mode
0x0c	Does not fit in area specified
0x0d	Data length to long
0x0c	PDF-417 coded data to large to fit in bar code
0x0d	
0x0e	

原型:

int PTK_EnableErrorReport (void);

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_EnableErrorReport ( );
```

PTK_DefineVariable

说明:

PTK_DefineVariable 函数的作用是定义变量。

在 FORM 里使用此函数来定义一个变量。

原型:

```
int PTK_DefineVariable (unsigned int  pid, unsigned int  pmax,  
                        char  porder, LPTSTR  pmsg);
```

参数:

pid: 变量 ID 号码, 取值范围: 00—99;

pmax: 最大字符个数, 取值范围: 1—99;

如果使用 KDU, 只能在 16 以内;

porder: 对齐方式, L—左对齐, R—右对齐, C—居中, N—不对齐;

pmsg: 提示内容, 将会在 KDU 或打印机的 LCD 显示。

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DefineVariable (0, 16, L, "\\ Enter Title:\\ ");
```

PTK_DrawRectangle

说明:

PTK_DrawRectangle 函数的作用是画矩形。

原型:

```
int PTK_DrawRectangle (unsigned int  px, unsigned int  py,
                      unsigned int  thickness, nsigned int  pEx,
                      unsigned int  pEy);
```

参数:

- px: 起始点的 X 坐标，以点(dots)为单位;
- py: 起始点的 Y 坐标，以点(dots)为单位;
- thickness: 边框的粗细，以点(dots)为单位;
- pEx: 终止点的 X 坐标，以点(dots)为单位;
- pEy: 终止点的 Y 坐标，以点(dots)为单位。

返回值:

- 0 -> OK;
- 其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DrawRectangle (50, 120, 5, 250, 150);
```

PTK_Y(作废)

说明:

PTK_Y 函数的作用是设置 RS232 串口参数。

原型:

```
int PTK_Y(unsigned int  pbaudrate, char  paritybit,
          unsigned int  pdata, unsigned int  pstop);
```

参数:

pbaudrate: 波特率;

pbaudrate 值	传送速率
38	38,400 baud
19	19,200 baud
96	9,600 baud
48	4,800 baud

24	2,400 baud
----	------------

paritybit: 奇偶校验, 0—奇校验, E—偶校验, N—不校验;

pdata: 数据位, 7 或 8;

pstop: 停止位, 1 或 2。

注:

G-2108 和 G-3106 系列打印机的串口参数由背板上的 DIP 开关设置。

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_Y(50, 120, 5, 250, 150);
```

PTK_SetDirection

说明:

PTK_SetDirection 函数的作用是设置标签打印方向。

注: 此函数将改变整张标签上所有内容方向, 如文本、条码、直线、矩形。

原型:

```
int PTK_SetDirection (char direct);
```

参数:

direct: 方向, 取值为 B 或 T, 缺省值为 T。

B: 将从标签右下角开始打印;

T: 从标签左上角开始正常打印。

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_SetDirection ( 'B' );
```

PTK_EnableFLASH

说明:

PTK_EnableFLASH 函数的作用是选择 FLASH 存储器。

当使用此函数后，发送到打印机的数据将被存储到 **FLASH** 里。

原型:

```
int PTK_EnableFLASH ( void);
```

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_EnableFLASH ( );
```

PTK_DisableFLASH

说明:

PTK_DisableFLASH 函数的作用是取消选择 FLASH 存储器;

当使用此函数后，发送到打印机的数据将被存储到 **SDRAM** 里。

原型:

```
int PTK_DisableFLASH (void);
```

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_DisableFLASH ( );
```

PTK_Download

说明:

PTK_Download 函数的作用是下载变量或系列号变量。

请参阅 PPL I 的 “?” 命令。

原型:

```
int PTK_Download(void);
```

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_Download( );
```

PTK_Reset

说明:

PTK_Reset 函数的作用是将打印机复位。

这个命令相当于关闭打印机电源, 然后重新打开打印机电源, 其系统重新初始化工作。

这个命令在以下情况下无效:

当下载软字体、PCX 图形或打印机在 DUMP 状态时;

在 FORM 里使用此命令;

当打印机正在执行打印任务中;

另外, 此函数执行后, 打印机进行初始化工作可能持续 2 秒钟以上, 这个期间打印机不接受任何控制指令。

原型:

```
int PTK_Reset(void);
```

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_Reset( );
```

PTK_FeedBack

说明:

PTK_FeedBack 函数的作用是要求打印机立刻反馈错误报告。

用户可以使用此命令立刻确定打印机的当前错误状态，打印机将传回 4 个字节到主机：

0xXX XX 0x0d 0x0a : Error/Status code <CR><LF>

Error/Status code	解释
00	无错误
01	语法错误
82	碳带探测出错
83	标签探测出错
86	切刀检测出错
87	打印头未关闭
88	暂停状态
99	其它错误

原型:

int PTK_FeedBack (void);

参数: 无

返回值:

0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

PTK_FeedBack ();

PTK_CutPage

说明:

PTK_CutPage 函数的作用是设置切刀的工作周期（即每打印多少页标签后，切刀才切一次纸）。

原型:

int PTK_CutPage (UINT page);

参数:

page: 页数,取值范围: 1-999; 默认是 1。

返回值:

0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_CutPage(1);
```

PTK_BinGraphicsList

说明:

PTK_BinGraphicsList 函数的作用是打印已存储在打印机 RAM 或 FLASH 存储器里的图形名称清单, **包含 Bin 格式和 PCX 格式的图形名称**。

原型:

```
int PTK_BinGraphicsList (void );
```

返回值: 0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
PTK_BinGraphicsList ( );
```

PTK_BinGraphicsDel

说明:

PTK_BinGraphicsDel 函数作用是删除存储在打印机里的一个或所有图形, **此图形可以是 Bin 格式或 PCX 格式的**。

原型:

```
int PTK_BinGraphicsDel (LPTSTR pid);
```

参数:

pid: 即将删除的图形名称, 最大长度为 16 个字符;
如果 pid = “*”, 则将删除所有存储在 RAM 或 FLASH 存储器里的图形。

返回值: 0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:
PTK_BinGraphics Del (“Bin2”);

PTK_BinGraphicsDownload

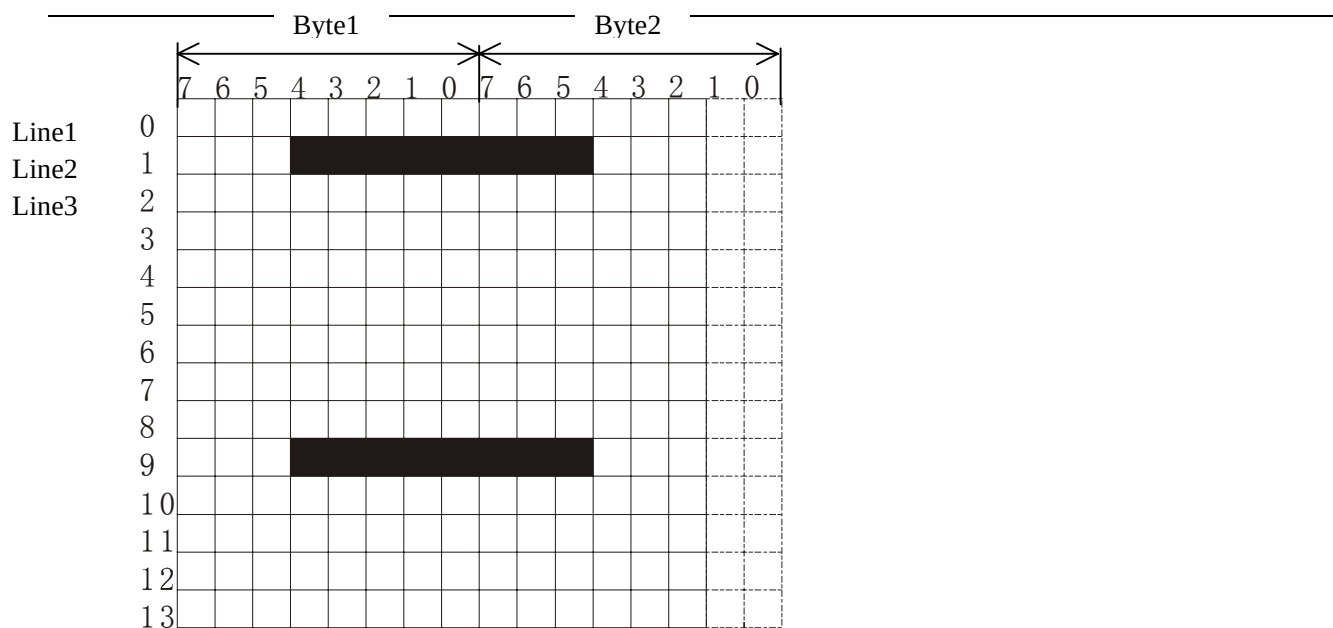
说明:
PTK_BinGraphicsDownload 函数的作用是存储一个 Bin 格式的图形到打印机。

原型:
int PTK_BinGraphicsDownload (char* name,unsigned int pbyte,unsigned int pH,UCHAR * Gdata);

参数:

- name: 自定义图形的名称, 最大长度为 16 个字符; 当图形存储到打印机后, 用户在 PTK_RecallBinGraphics () 中使用此名称才能将图形读取出来打印。
- pbyte: 一行数据的字节数(1Byte = 8bits); 如果一行数据的点数不能整除 8, 则其字节数应该等于商加上 1; 如: 一行是 14bit 的数据的字节数是 2;
- pH: 图形的高度, 以点(dots)为单位;
- Gdata([...raster data...]): 二进制图形数据, 数据量大小= pbyte * pH (Bytes)。
Bit 值为 1 是打印内容, 为 0 是空白内容。

二进制数据传输顺序是从左到右, 从上到下, 以下图为例:
数据传输顺序为: Line1 的 Byte1(0x00), Line1 的 Byte2(0x00), Line2 的 Byte1(0x1f), Line2 的 Byte2(0xe0), Line3 的 Byte1(0x00), Line3 的 Byte2(0x00), ...
其中虚线部分是非图形区域, 对应它们的 bit 值为 0。



返回值: 0 -> OK;
其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

```
char buf[] = {0xff, 0xff, 0xe0, 0x1f, 0xff, 0xff...};
PTK_BinGraphicsDownload ("BinA", 3, 24, buf);
```

PTK_RecallBinGraphics

说明:
PTK_RecallBinGraphics 函数是打印一个 Bin 格式的图形。

原型:

```
int PTK_RecallBinGraphics (unsigned int px, unsigned int py, char* name);
```

参数:
 px: 设置 X 坐标;以点(dots)为单位;
 py: 设置 Y 坐标;以点(dots)为单位;
 name: Bin 图形文件名称;

返回值: 0 -> OK;

其它返回值请参考章节：**CDFPSK.dll 错误返回值解析**。

范例：

```
PTK_RecallBinGraphics(10,100,"BinA");
```

PTK_ErrorReport

说明：

PTK_ErrorReport 函数的作用是发送错误查询指令到打印机并且从指定串口接收和分析打印机当前错误代码。

在执行此函数之前，如果程序正在打开一个发送端口，必须首先执行 **ClosePort()关闭已打开的端口！**

用户可以使用此命令立刻确定打印机的当前错误状态，打印机将传回 4 个字节到主机：

0xXX XX 0x0d 0x0a : Error/Status code <CR><LF>

Error/Status code	解释	PTK_ErrorReport()返回值	ErrorCode
00	无错误	0	"00"
01	语法错误	1	"01"
82	碳带探测出错	82	"82"
83	标签探测出错	83	"83"
86	切刀检测出错	86	"86"
87	打印头未关闭	87	"87"
88	暂停状态	88	"88"
99	其它错误	99	"99"

原型：

```
int PTK_ErrorReport (int wPort, int rPort, DWORD BaudRate, BOOL HandShake, LPSTR errorCode);
```

参数：

wPort: 发送数据的端口；

- 0: 表示打印到文件 PBuffi.txt (在执行程序目录下建立文件)；**(此端口请勿使用！)**
- 1: 表示打开 LPT1；
- 2: 表示打开 LPT2；
- 3: 表示打开 LPT3；
- 4: 表示打开 COM1；
- 5: 表示打开 COM2；
- 6: 表示打开 COM3。

rPort: 接收打印机当前错误状态代码的端口；

- 1: 表示打开 COM1 作为接收端口；

-
- 2: 表示打开 COM2 作为接收端口;
 - 3: 表示打开 COM3 作为接收端口;

BaudRate: 要设置的串口波特率, 可取值:
9600, 19200, 38400, 57600;

HandShake: 是否使用硬件握手 (HandShaking);
TRUE: 硬件握手 (HandShaking) 有效,
FALSE: 硬件握手 (HandShaking) 无效。

errorCode: 用来返回错误状态代码字符串 (共四个字节) 给用户作为参考, 可为 NULL。

返回值:

>0 -> OK;

其它返回值请参考章节: **CDFPSK.dll 错误返回值解析**。

范例:

1. 透过 LPT1 发送指令, 从 COM1 读取当前错误状态代码:

```
OpenPort(1);
```

```
....
```

```
ClosePort();
```

```
char buff[5] = {0};
```

```
PTK_ErrorReport (1, 1, 9600, TRUE, buff );
```

```
(或 PTK_ErrorReport (1, 1, 9600, TRUE, NULL );)
```

```
.....
```

2. 透过 COM1 发送指令, 并从 COM1 读取当前错误状态代码:

```
OpenPort(4);
```

```
....
```

```
ClosePort();
```

```
char buff[5] = {0};
```

```
PTK_ErrorReport (4, 1, 9600, TRUE, buff );
```

```
(或 PTK_ErrorReport (4, 1, 9600, TRUE, NULL );)
```

```
.....
```

PTK_UserFeed

说明:

PTK_UserFeed 函数的作用是令打印机进纸给定的长度。

原型:

int PTK_UserFeed (UINT feedLen);

返回值: 0 -> OK;

其它返回值请参考章节: **RDFPSK.dll 错误返回值解析**。

范例:

PTK_UserFeed (100);

PTK_UserBackFeed

说明:

PTK_UserBackFeed 函数的作用是令打印机退纸给定的长度。

原型:

int PTK_UserBackFeed (UINT feedLen);

返回值: 0 -> OK;

其它返回值请参考章节: **RDFPSK.dll 错误返回值解析**。

范例:

PTK_UserBackFeed (100);

CDFPSK.dll 错误返回值解析

-1000
to
-1030 : 打开端口操作出错;

-1031
to
-1037 : 串口读取操作出错;

-1040 : 从串口读取数据出错;
-1041 : 从串口读取数据出错;
-1042 : 串口波特率值有误;

-3001 : PTK_GetInfo 执行出错;
-3002 : PTK_DrawText 执行出错;
-3003 : PTK_DrawText 的参数错误;
-3004 : PTK_DrawText 或者 PTK_DrawBarcode 的 pdirec 参数错误;
-3005 : PTK_DrawText 的 pFont 参数错误;
-3006 : PTK_DrawText 的 pHorizontal 或者 pVertical 参数错误;
-3007 : NULL;
-3008 : PTK_DrawBarcode 执行出错;
-3009 : PTK_DrawBarcode 参数错误;
-3010 : PTK_DefineCounter 执行出错;
-3011 : PTK_DefineCounter 参数错误;
-3012 : PTK_SetDarkness 执行出错;
-3013 : PTK_DS 执行出错;
-3014 : PTK_SoftFontList 执行出错;
-3015 : PTK_SoftFontDel 参数错误;
-3016 : PTK_SoftFontDel 执行出错;
-3017 : PTK_FormEnd 执行出错;
-3018 : PTK_FormList 执行出错;
-3019 : PTK_FormDel 分配内存出错;
-3020 : PTK_FormDel 参数错误;
-3021 : PTK_FormDel 执行出错;
-3022 : PTK_ExecForm 分配内存出错;
-3023 : PTK_ExecForm 参数错误;
-3024 : PTK_ExecForm 执行出错;
-3025 : PTK_FormDownload 分配内存出错;
-3026 : PTK_FormDownload 参数错误;
-3027 : PTK_FormDownload 执行出错;

-3028 : PTK_DrawPcxGraphics 分配内存出错;
-3029 : PTK_DrawPcxGraphics 参数错误;
-3030 : PTK_DrawPcxGraphics 执行出错;
-3031 : PTK_PcxGraphicsList 执行出错;
-3032 : PTK_PcxGraphicsDel 分配内存出错;
-3033 : PTK_PcxGraphicsDel 参数错误;
-3034 : PTK_PcxGraphicsDel 执行出错;
-3035 : PTK_PcxGraphicsDownload 内存分配错误;
-3036 : PTK_PcxGraphicsDownload 打开文件错误;
-3037 : PTK_PcxGraphicsDownload 执行出错;
-3038 : PTK_DrawBinGraphics 执行出错;
-3039 : (同上);
-3040 : NULL;
-3041 : PTK_DisableCircumgyrate 执行出错;
-3042 : PTK_EnableCircumgyrate 执行出错;
-3043 : PTK_DrawLineXor 执行出错;
-3044 : PTK_DrawLineOr 执行出错;
-3045 : PTK_DrawDiagonal 执行出错;
-3046 : PTK_DrawWhiteLine 执行出错;
-3047 : PTK_ClearBuffer 执行出错;
-3048 : PTK_SetPrinterState 执行出错;
-3049 : PTK_SetPrinterState 参数错误;
-3050 : PTK_PrintLabel 执行出错;
-3051 : PTK_PrintLabel 参数错误;
-3052 : PTK_PrintLabelAuto 执行出错;
-3053 : PTK_PrintLabelAuto 参数错误;
-3054 : PTK_SetLabelHeight 执行出错;
-3055 : PTK_SetLabelWidth 执行出错;
-3056 : PTK_SetCoordinateOrigin 执行出错;
-3057 : PTK_SetPrintSpeed 执行出错;
-3058 : PTK_SetPrintSpeed 参数错误;
-3059 : PTK_PrintConfigunation 执行出错;
-3060 : PTK_DisableErrorReport 执行出错;
-3061 : PTK_EnableErrorReport 执行出错;
-3062 : PTK_DefineVariable 执行出错;
-3063 : PTK_DefineVariable 参数错误;
-3064 : PTK_DefineVariable 参数错误;
-3065 : PTK_DrawRectangle 执行出错;
-3066 : PTK_Y 执行出错;
-3067 : PTK_Y 参数错误;
-3068 : PTK_SetDirection 执行出错;
-3069 : PTK_SetDirection 参数错误;
-3070 : PTK_EnableFLASH 执行出错;
-3071 : PTK_DisableFLASH 执行出错;

-3072 : PTK_Download 执行出错;
-3073 : PTK_Reset 执行出错;
-3074 : PTK_BackFeed 执行出错;
-3075 : 分配保存 PCX HEAD 文件结构内存出错;
-3076 : 分配保存 PCX data 内存出错;
-3077 : 分配保存当前程序运行的文件路径内存出错;
-3078 : 创建 PCX 文件出错;
-3079 : 保存 PCX data 出错;
-3080 : PTK_PrintPCX 执行出错;
-3081 : 创建 PrinterDC 失败, 进行出错处理;
-3082 : 分配保存 bitmap 内存出错;

-3083 : PTK_BinGraphicsDownload(执行出错;
-3084 : PTK_BinGraphicsDel 执行出错;
-3085 : PTK_BinGraphicsList 执行出错;
-3087 : PTK_RecallBinGraphics 执行出错;
-3088 : PTK_RecallBinGraphics 的" name"参数字符长度=0 或>16;

-3089 : PTK_UserFeed 执行出错;
-3090 : PTK_UserBackFeed 执行出错;

-3100 : SetCommPort 中更改串口出错;

-3101 : PTK_CutPage 执行出错;

-3200
to
-3400 : 端口未打开或已经关闭;

-3250 : 请检查 PTK_DrawBar2D_Pdf417()函数输入参数;

-3251
to
-3257 : 端口未打开或已经关闭;