POSTEK PPLE

API 函数手册

条码标签打印机

Version 3.0.0.6

深圳市博思得科技发展有限公司

二〇一七年

API 函数库文件说明

名称：CDFPSK.dll
版本编号：3.0.0.6170522

用途

本 API 函数库为深圳市博思得科技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, Windows 7, Windows 8 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

PPLE：深圳市博思得科技发展有限公司的第一套打印机编程语言（Printer Program Language E）。

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, 1dot = 25.4mm/203 ≈ 0.125mm(1dot = 1000 / 203 ≈ 5mil)；
- 对于 300DPI 的打印机来说, 1dot = 25.4mm/300 ≈ 0.085mm(1dot = 1000 / 300 ≈ 3mil)；
- 对于 600DPI 的打印机来说, 1dot = 25.4mm/600 ≈ 0.042mm(1dot = 1000 / 600 ≈ 2mil)。

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

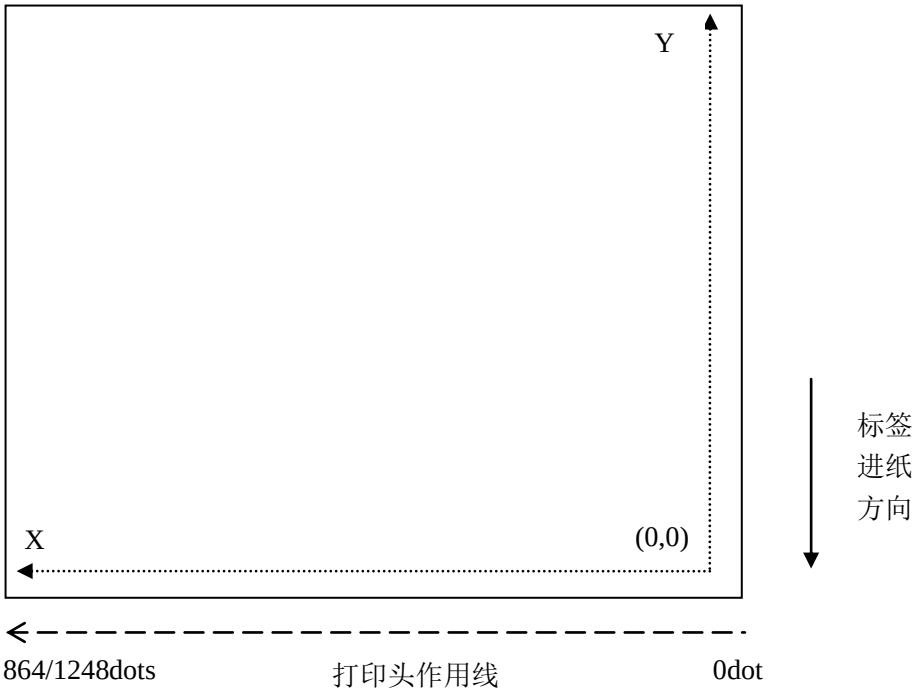
字符串

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

字符	输入
“	\ “
\	\\
0x00 - 0x7F	\x00 - \x7F

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
OpenPort	打开通讯端口。
ClosePort	关闭使用 OpenPort 函数打开的通讯端口。
SetPCComPort	设置 PC 机上串口的传输波特率。
PTK_GetErrState	检测使用 CDFPSK.DLL 里的其它函数后是否有错误产生;
PTK_GetInfo	得到本 API 函数库的版本信息。
PTK_ClearBuffer	清除打印机缓冲内存的内容。
PTK_SetDarkness	设置打印头发热温度。
PTK_SetPrintSpeed	设置打印速度。
PTK_SetLabelHeight	设置标签的高度和定位间隙\黑线\穿孔的高度。
PTK_SetLabelWidth	设置标签的宽度。
PTK_SetDirection	设置标签打印方向。
PTK_SetCoordinateOrigin	设置/改变坐标原点。
PTK_PrintLabel	命令打印机执行打印工作。
PTK_DrawText	打印一行文本文字。
PTK_DrawTextEx	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。
PTK_DrawTextTrueTypeW	打印一行 TrueType Font 文字，并且文字宽度和高度可以微调。
PTK_DrawBarcode	打印一个条码。
PTK_DrawBarcodeEx	打印一个条码，内容可以是常量、序列号、变量或组合字符串。
PTK_DrawBar2D_DATAMATRIX	打印一个 DataMatrix 二维条码。
PTK_DrawBar2D_QR	打印一个 QR 条码（指令方式）。
PTK_DrawBar2D_QREx	打印一个 QR 条码（图形方式）。
PTK_DrawBar2D_MaxiCode	打印一个 MaxiCode 条码。
PTK_DrawBar2D_Pdf417	打印一个 PDF417 二维条码。
PTK_DrawBar2D_HANXIN	打印一个汉信码二维条码。
PTK_PcxGraphicsList	打印已存储在打印机 RAM 或 FLASH 存储器里的图形名称清单。
PTK_PcxGraphicsDel	删除存储在打印机 RAM 或 FLASH 存储器里的里的一个或所有图形。
PTK_PcxGraphicsDownload	存储一个 PCX 格式的图形到打印机。
PTK_DrawPcxGraphics	打印指定的图形。
PTK_PrintPCX	打印一个 PCX 格式的图形。
PTK_BmpGraphicsDownload	转换 BMP 到 PCX 格式，然后将 PCX 格式的图形到打印机。
PTK_BinGraphicsList	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)。
PTK_BinGraphicsDel	删除已存储在打印机 RAM 或 FLASH 存储器里的一个或所有图形(此图形可是 Bin 格式或 PCX 格式的)。

PTK_BinGraphicsDownload	存储一个 Bin 格式的图形到打印机。
PTK_RecallBinGraphics	打印一个已保存在打印机里的 Bin 格式图形。
PTK_DrawBinGraphics	打印二进制格式的图形。
PTK_DrawRectangle	画距形。
PTK_DrawLineXor	画直线(两直线相交处作"异或"处理)。
PTK_DrawLineOr	画直线(两直线相交处作"或"处理)。
PTK_DrawDiagonal	画斜线。
PTK_DrawWhiteLine	画白色直线。
PTK_SoftFontList	打印存储在 RAM 或 FLASH 存储器里的软字体的名称清单。
PTK_SoftFontDel	删除存储在 RAM 或 FLASH 存储器里的一个或所有的软字体。
PTK_FormList	打印存储在打印机里的表格名称清单。
PTK_FormDel	删除存储在打印机里的一个或所有的表格。
PTK_FormDownload	存储一个表格到打印机；此命令与 PTK_FormEnd 函数配对使用。
PTK_FormEnd	结束存储表格(Form)，此函数与 PTK_FormDownload 配对使用。
PTK_ExecForm	运行指定的表格。
PTK_DefineCounter	定义一个序列号变量。
PTK_DefineVariable	定义变量。
PTK_Download	下载变量或系列号变量。
PTK_DownloadInitVar	初始化变量或系列号变量。
PTK_PrintLabelAuto	自动执行打印工作。
PTK_SendFile	发送指令文件到打印机。
PTK_GetUSBID	获取打印机 USBID 号。
PTK_DisableBackFeed	取消打印回转功能。
PTK_EnableBackFeed	设置打印回转功能。
PTK_PrintConfiguration	打印机器当前的设置/工作状态。
PTK_SetPrinterState	设置打印机的工作状态。
PTK_DisableErrorReport	取消错误反馈。
PTK_EnableErrorReport	设置错误反馈 。
PTK_EnableFLASH	选择 FLASH 存储器。
PTK_DisableFLASH	取消选择 FLASH 存储器。
PTK_FeedMedia	命令打印机走一行标签。
PTK_MediaDetect	校准纸张探测器。
PTK_CutPage	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）。
PTK_CutPageEx	实时调整切刀切纸张数。
PTK_FeedBack	要求打印机立刻反馈错误报告。
PTK_Reset	将打印机复位。

PTK_ErrorReport	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。
PTK_ErrorReportUSB	发送错误查询指令到打印机并且指定 USB 端口接收和分析打印机当前错误代码。
PTK RWRFIDLabel	写 RFID 标签。
PTK SetRFLabelPWAndLockRFLabel	设置 RFID 标签密码和锁定 RFID 标签。
PTK SetRFID	RFID 设置指令。
PTK_SetFontGap	调整打印文字字间距。
PTK_SetBarCodeFontName	设置可打印条码文字的下载字体。 *该函数无效
PTK_RenameDownloadFont	将下载到打印机的字体进行重命名字体 ID。
PTK_SetCharSets	设置字符集。
PTK_ReadRFTagData	发送读取 RFID 标签数据指令到打印机并且从指定串口接收打印机当前读取的 RFID 标签信息。
PTK_ReadRFTagDataUSB	从指定 USB 端口发送读取 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签信息。
PTK_ReadRFTagDataNet	从指定 TCP/IP 端口发送读取 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签信息。
PTK_Connect	建立打印机网络连接。
PTK_CloseConnect	断开使用 PTK_Connect 函数建立的打印机网络连接。
PTK_ErrorReportNet	指定 TCP/IP 端口发送错误查询指令到打印机并且接收和分析打印机当前错误代码
PTK_ReadPrintConifUSB	通过 USB 端口将打印机信息反馈给主机
PTK_ReadPrintConifNet	通过网口将打印机信息反馈给主机
PTK_RFIDCalibrate	校准 RFID 芯片读写位置(支持 RFID 打印机固件版本 V1.73 及更高)
PTK_RWHFLabel	读写高频 RFID 标签
PTK_SetHFRFID	设置高频RFID 标签。
PTK_ReadHFTagData	从指定串口接收打印机当前读取的高频 RFID 标签信息。
PTK_ReadHFTagDataUSB	从指定 USB 端口发送读取高频 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签信息
PTK_ReadHFTagUID	从指定串口接收打印机当前读取取的高频 RFID 标签 UID 信息。
PTK_ReadHFTagUIDUSB	从指定 USB 端口发送读取高频 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签 UID 信息.
PTK_ReadHFTagDataPrintAuto	打印过程中先读取高频标签指定位置块的数据内容,然后打印在标签上面。
PTK_ReadHFTagUIDPrintAuto	打印过程中先读取高频标签标签 UID 号，然后打印在标签上面。

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(int PortFlag);

参数：

xx：通讯端口代号；

0： 表示打印到文件 PBuffi.txt（在执行程序目录下建立文件）；

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。

7： 表示打开USB001。

8： 表示打开USB002。

9： 表示打开USB003。

……

255: 表示当且仅有一台Postek打印机时，默认打开这台Postek打印机。

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：

```
OpenPort(1);            //表示打开 lpt1 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：

Void ClosePort(void);

参数：无

返回值：无

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（若打印机配置 DIP 开关通过调整 DIP 开关的 7，8PIN 设置，请参阅用户手册；若无 DIP 开关可通过 Postek Utility 3.0 软件设置或联系客服设置方法）

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);

PTK_GetErrState

说明：
PTK_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 PTK_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 = PTK_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_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_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_SetPrintSpeed

说明：

PTK_SetPrintSpeed 函数的作用是设置打印速度。

此函数适用于 POSTEK 所有机型的打印速度设置。

注意：不同型号打印机最大打印速度不同（具体请查阅对应用户手册），若设置值大于打印机最大打印速度，设置无效。

原型：

int PTK_SetPrintSpeed (unsigned int px);

参数：

px：取值范围为0 – 10，或者10 – 100。

p1值	速度	速度
10	1.0 ips (25.40 mm/s)	0 or 1
15	1.5 ips (38.10 mm/s)	
20	2.0 ips (50.80 mm/s)	2
25	2.5 ips (63.50 mm/s)	
30	3.0 ips (76.20 mm/s)	3
35	3.5 ips (88.90 mm/s)	
40	4.0 ips (101.60 mm/s)	4
50	5.0 ips (127.00 mm/s)	5
60	6.0 ips (152.40 mm/s)	6
70	7.0 ips (177.80 mm/s)	7
80	8.0 ips (203.20 mm/s)	8
90	9.0 ips (228.60 mm/s)	9
100	10.0 ips (254.00 mm/s)	10

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_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,
int gapOffset, BOOL bFlag);

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。
gapOffset：标签间隙/黑线/穿孔定位偏移值，以点(dots)为单位。
bFlag：gapOffset 是否有效,TRUE-有效；FALSE-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：
PTK_SetLabelHeight (160, 24,0,FALSE);

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_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的方向，如文本、条码、直线、距形。

原型：

int PTK_SetDirection (TCHAR direct);

参数：

direct：方向，取值为 B 或 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 (_T('B'));

PTK_SetCoordinateOrigin

说明：

PTK_SetCoordinateOrigin 函数的作用是设置/改变坐标原点。通过改变坐标原点可以实现一行多列的打印。

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_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_DrawText

说明：

PTK_DrawText 函数作用是打印一行文本文字。

原型：

int PTK_DrawText (unsigned int px, unsigned int py,
unsigned int pdirec, unsigned int pFont,
unsigned int pHorizontal, unsigned int pVertical,
TCHAR 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: 为为打印机内置 5 种西文字体;

6：为打印机内置 1 种中文字体；
若打印机非 Postek V6 系列打印机时，6 表示打印机内置 24*24 简体汉字；
若打印机为 Postek V6 系列打印机时，6 表示为打印机内置的黑体。
‘A’—‘Z’：为下载的软字体.

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(字母大写有效)
6	中文字体
‘A’~’Z’	软字体

pHorizontal：当 pFont 设置为内置字体时（1~6），此时 pHorizontal 为设置点阵水平放大系数. 可选择:1—24.
当 pFont 选择 TrueType 字体时（A~Z），此时 pHorizontal 为设置字体的宽度，单位为像素点（不限大小）。
注意：若您使用 POSTEK V6 系列打印机时 ，当 pFont 设置为 6（内置中文字体）时，此时 pHorizontal 是设置字体宽度（同选择 TrueType 字体），单位为像素点（不限大小）。

pVertical：当 pFont 设置为内置字体时(1~6),此时 pVertical 为设置点阵垂直放大系数. 择:1—24.
当 pFont 选择 TrueType 字体时（A~Z），此时 pVertical 为设置字体的高度，单位为像素点（不限大小）。
注意：若您使用 POSTEK V6 系列打印机时 ，当 pFont 设置为 6（内置中文字体）时，此时 pVertical 是设置字体高度（同选择 TrueType 字体），单位为像素点（不限大小）。

ptext：选’N’ 对应 ASCII 值 78 则打印正常文本(如黑字白底文本),
选’R’ 对应 ASCII 值 82 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。

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:

```
PTKDrawText (50,30,0,2,1,1,78,”123456789”);
```

PTK_DrawTextEx

说明：

PTK_DrawTextEx 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_DrawTextEx(unsigned int   px,   unsigned int   py,
                   unsigned int   pdirec,   unsigned int   pFont,
                   unsigned int   pHorizontal, unsigned int   pVertical,
                   TCHAR   ptext, LPTSTR pstr, BOOL Varible);
```

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.

1—5: 为为打印机内置 5 种西文字体;

6：为打印机内置 1 种中文字体；

若打印机非 Postek V6 系列打印机时，6 表示打印机内置 24*24 简体汉字；

若打印机为 Postek V6 系列打印机时，6 表示为打印机内置的黑体。

‘A’—‘Z’: 为下载的软字体.

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
6	中文字体
‘A’~’Z’	软字体

pHorizontal: 当 pFont 设置为内置字体时（1~6），此时 pHorizontal 为设置点阵水平放大系数. 可选择:1—24.

当 pFont 选择 TrueType 字体时（A~Z），此时 pHorizontal 为设置字体的宽度，单位为像素点（不限大小）。

注意：若您使用 POSTEK V6 系列打印机时 ，当 pFont 设置为 6（内置中文字体）时，此时 pHorizontal 是设置字体宽度（同选择 TrueType 字体），单位为像素点（不限大小）。

pVertical: 当 pFont 设置为内置字体时(1~6),此时 pVertical 为设置点阵垂直放大系数. 择:1—24.

当 pFont 选择 TrueType 字体时（A~Z），此时 pVertical 为设置字体的高度，单位为像素点（不限大小）。

注意：若您使用 POSTEK V6 系列打印机时 ，当 pFont 设置为 6（内置中文字体）时，此时 pVertical 是设置字体

高度（同选择 TrueType 字体），单位为像素点（不限大小）。

ptext：选’N’对应 ASCII 值 78 则打印正常文本(如黑字白底文本),
选’R’对应 ASCII 值 82 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。
Vn：变量字符串，此变量字符串必须已经定义。
如:”data1”CnVn”data2”。
R： 表明此次打印内容为当前 UHF RFID 标签 TID(64bit)数据（注意此功能仅支持 RFID 打印机）。

Varible：TRUE 表示当前字符串当中包含有变量操作,需加‘\”’将打印常量字符串包含。
例如：PTK_DrawTextEx (50,30,0,2,1,1,’N’,”\”123456789\”C0”,TRUE);
FALSE 表示当前字符串当中不包含有变量操作，不需加‘\”’。
例如：PTK_DrawTextEx (50,30,0,2,1,1,’N’,” 123456789”,FALSE);和
PTK_DrawText(50,30,0,2,1,1,’N’,” 123456789”)效果相同；

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_DrawTextEx (50,30,0,2,1,1,’N’,”\”123456789\”C0”,TRUE);
PTK_DrawTextEx (50,30,0,2,1,1,’N’,” 123456789”,FALSE);
PTK_DrawTextEx (50,30,0,2,1,1,’N’,”C1”,TRUE);
PTK_DrawTextEx (50,30,0,2,1,1,’N’,”V3”,TRUE);
PTK_DrawTextEx (50,30,0,2,1,1,’N’,”  “\”Printer\” C2V1\”is ok.\” ”,TRUE);
```

打印序列号和变量字符串,（一般在 Form 中使用）

```
PTK_FormDel("LSFORM");                                  //删除 Form " LSFORM "
PTK_FormDownload("LSFORM");                             //存储 Form " LSFORM "
PTK_DefineCounter(0,10,78,"+1","InputC0");              //定义一个序列号变量 C0
PTK_DefineCounter(1,10,78,"+1"," InputC1");             //定义一个序列号变量 C1
PTK_DefineVariable(0,16,78," InputV0");                 //定义变量字符串 V0，命名为：'V'+第一个参数
PTK_DrawBarcodeEx(100,100,0,"3",2,6,30,66,"C0",true);   //打印序列号条码，
PTK_DrawTextEx(100,160,0,2,1,1,78,"\"条码内容为：\"C0",true);//打印一行文本文字(字符串中即有常量也有变量 C0)

PTK_DrawTextEx(100,220,0,2,1,1,78,"\"打印序列号：\"C1",true);//打印序列号数值 C1
PTK_DrawTextEx(100,280,0,2,1,1,78,"\"打印变量：\"V0",true);  //打印变量字符串 V0
PTK_DrawTextEx(100,340,0,2,1,1,78,"\"组合功能：序号:\"C1\"变量:\"V0",true);//组合打印
PTK_FormEnd();                                          //结束存储表格
```

```
PTK_ExecForm("LSFORM");          // 运行指定的表格:LSFORM
PTK_Download();                  // 下载变量或系列号变量

// 以下按序号，及变量已定义的顺序初始化（此例子定义顺序为 C0，C1，V0）
PTK_DownloadInitVar("12345678");   // 初始化系列号变量 C0
PTK_DownloadInitVar("123456");     // 初始化系列号变量 C1
PTK_DownloadInitVar("1111");       // 初始化变量 V0

PTK_PrintLabel(2,1);               // 命令打印机执行打印工作
```

打印 RFID 标签

```
PTK_DrawTextEx(80, 168, 0, 3, 1, 1, 'N', _T("R"), TRUE);
```

打印输出：

在坐标为 80(X 坐标),168(Y 坐标)位置打印当前 UHF RFID TID 十六进制数据:E28051052000555D（此为范例数据，请以实际 RFID 标签数据为准）。

PTK_DrawTextTrueTypeW

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,
                          unsigned int FHeight, unsigned int FWidth,
                          LPCTSTR FType, unsigned int Fspin,
                          unsigned int FWeight,BOOL FItalic,
                          BOOL FUnline, BOOL FStrikeOut,
                          LPTSTR id_name,LPCTSTR data);
```

参数：

x：设置 X 坐标，以点(dots)为单位；

y：设置 X 坐标，以点(dots)为单位；

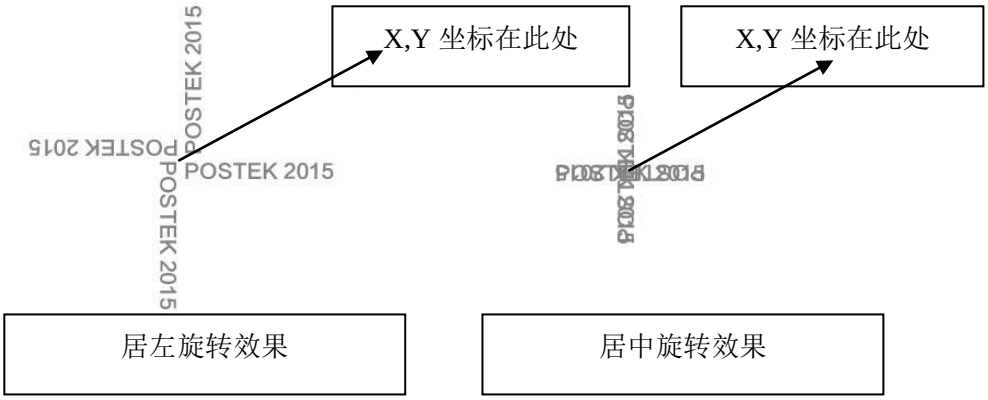
FHeight：字型高度，以点(dots)为单位；

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->居左 0 度，2->居左 90 度，3->居左 180 度，4->居左 270 度
5->居中 0 度，6->居中 90 度，7->居中 180 度，8->居中 270 度



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；(当 data 参数或其他参数不同时，请务必设定不同的 id_name 值)

data：字符串内容。

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：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_PcxGraphicsDel("A1")或者 PTK_PcxGraphicsDel("*");//建议打印 TrueType 字体前调用

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

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.(不同类型条码有字符限制或字符个数等限制，请参考具体标准)

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
2	Interleaved 2 of 5
2C	Interleaved 2 of 5 with check sum digit
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’ 对应 ASCII 值 78 则不打印条码下面的人可识别文字,
选’B’ 对应 ASCII 值 66 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。

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_DrawBarcodeEx

说明：

PTK_DrawBarcodeEx 函数作用是打印一个条码, 内容可以是常量、序列号、变量或组合字符串

原型：

```
int   PTK_DrawBarcodeEx ( unsigned int   px,unsigned int   py,
                          unsigned int   pdirec, LPTSTR pCode,
                          unsigned int   NarrowWidth,
                          unsigned int   pHorizontal,
                          unsigned int   pVertical,
```

char ptext, LPTSTR pstr, BOOL Varible);

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.（不同类型条码有字符限制或字符个数等限制，请参考具体标准）

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
2	Longerleaved 2 of 5
2C	Longerleaved 2 of 5 with check sum digit
2D	Longerleaved 2 of 5 with human readable check digit
2G	German Postcode
2M	Matrix 2 of 5
2U	UPC Longerleaved 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’ 对应 ASCII 值 78 则不打印条码下面的人可识别文字,
选’B’ 对应 ASCII 值 66 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 PTK_DrawTextEx 函数范例。
Vn：变量字符串，此变量字符串必须已经定义, 请参考 PTK_DrawTextEx 函数范例。。
如：”data1”CnVn”data2”。
Varible：true 表示当前字符串当中包含有变量操作,需加‘\”’将打印常量字符串包含。
例如：PTKDrawBarcodeEx (50,30,0,”1A”,1,1,10,’N’,”\”123456\””,true);
false 表示当前字符串当中不包含有变量操作，不需加‘\”’。
例如：PTKDrawBarcodeEx (50,30,0,”1A”,1,1,10,’N’,”123456”,false);和
PTKDrawBarcode(50,30,0,”1A”,1,1,10,’N’,”123456”)效果相同；

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

范例:

PTKDrawBarcodeEx (50,30,0,”1A”,1,1,10,’N’,”\”123456\””, true);
PTKDrawBarcodeEx (50,30,0,”1A”,1,1,10,’N’,”123456”, **false**);
PTKDrawBarcodeEx (50,30,0,”1A”,1,1,10,’N’,”C2”, true);
PTKDrawBarcodeEx (50,30,0,”1A”,1,1,10,’N’,”V1”, true);
PTKDrawBarcodeEx (50,30,0,”1A”,1,1,10,’N’,”C1\” is\ ”V2”, true);

PTK_DrawBar2D_DATAMATRIX

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

原型：
int 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：

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

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

PTK_DrawBar2D_DATAMATRIX(50, 30, 0, 0, 0, 5,"123456789");

PTK_DrawBar2D_QR

说明：

PTK_DrawBar2D_QR 函数作用是打印一个 QR 条码。

原型：

int 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, LPTSTR pstr);

参数：

x: ●X 座标。

y: ●Y 座标。备注：1 dot = 0.125 mm。

w: ●最大打印宽度，单位 dots。

v: ●最大打印高度，单位 dots。

o: ●设置旋转方向, 范围：0～3。
（0--0°，1--90°，2--180°，3--270°）

r: ●设置放大倍数，以点(dots)为单位,范围值：(1 - 99)。
（1--放大 1 倍， 2--放大 2 倍，3--放大 3 倍…… ）
若放大倍数在 1-99 范围外，默认设置为放大 1 倍

m: ●QR 码编码模式选择,范围值(0 - 4)。
0 是选择数字模式
1 是选择数字字母模式
2 是选择字节模式 0~256
3 是选择中国汉字模式
4 是选择混合模式

g: ●QR 码纠错等级选择,范围值(0 - 3)。
0 是'L'等级
1 是'M'等级
2 是'Q1'等级
3 是'H1'等级

s: ●QR 码掩模图形选择,范围值(0 - 8)。
0 - 是掩模图形 000
1 - 是掩模图形 001
2 - 是掩模图形 010
3 - 是掩模图形 011
4 - 是掩模图形 100
5 - 是掩模图形 101

6 - 是掩模图形 110
7 - 是掩模图形 111
8 - 是自动选择掩模图形

pstr: ●资料字串。

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

范例：

```
PTK_DrawBar2D_QR(50, 30, 180, 180, 0, 0, 1, 0, 8, "123456789");
```

PTK_DrawBar2D_QREx

说明：

PTK_DrawBar2D_QREx 函数作用是打印一个 QR 条码。

原型：

```
int PTK_DrawBar2D_QREx ( unsigned int x, unsigned int y,
                         unsigned int o, unsigned int r,
                         unsigned int g, unsigned int v,
                         unsigned int s, LPTSTR id_name,
                         LPTSTR pstr);
```

参数：

x: ●X 座标。

y: ●Y 座标。备注：1 dot = 0.125 mm。

o: ●设置旋转方向, 范围：0～3。
（0--0°，1--90°，2--180°，3--270°）

r: ●设置放大倍数，以点(dots)为单位,范围值：(1 - 99)。
（1--放大 1 倍， 2--放大 2 倍，3--放大 3 倍… … ）
若放大倍数在 1-99 范围外，默认设置为放大 1 倍

g: ●QR 码纠错等级选择,范围值(0 - 3)。
0 是'L'等级
1 是'M'等级
2 是'Q1'等级
3 是'H1'等级

v: ●QR 码版本（Version)对应 QR 码图形大小(size)，版本号从 1 到 40。
版本 1 就是一个 21*21 的矩阵，每增加一个版本号，矩阵的大小就增加 4 个模块(Module)，因此，版本 40 就是一个 177*177 的矩阵，如果选择对应。
0:为自动匹配(默认使用)
1: 21* 21
2: 25* 25
.
.
.
.
.
.
40: 177* 177

s: ●QR 码掩模图形选择,范围值(0 - 8)。
0 - 是掩模图形 000
1 - 是掩模图形 001
2 - 是掩模图形 010

3 - 是掩模图形 011
4 - 是掩模图形 100
5 - 是掩模图形 101
6 - 是掩模图形 110
7 - 是掩模图形 111
8 - 是自动选择掩模图形

id_name: ●识别名称，因为二维码将被转换图形以 id_name 作为 bin 格式图形的名称下载到打印机内进行打印。
当 pstr 参数或其他参数不同时，请务必设定不同的 id_name 值

pstr: ●资料字串。

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

范例：
PTK_DrawBar2D_QREx(50,30, 0, 5, 1, 0, 8,"A1", "1321421421421");

PTK_DrawBar2D_MaxiCode

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

原型：
int PTK_DrawBar2D_MaxiCode (unsigned int x, unsigned int y,
unsigned int m, unsigned int u,
LPTSTR pstr)

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

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

范例：
PTK_DrawBar2D_MaxiCode(50,30,4,0,"1Z000A7&dajc_iaj-3=+~#^$5");

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：

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

传回值： 0 -> OK.

其它返回值请参考章节： **CDFPSK.dll 错误返回值解析**。

范例：

```
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_DrawBar2D_HANXIN

说明：

PTK_DrawBar2D_HANXIN 函数作用是打印一个汉信码二维条码。

原型：

int PTK_DrawBar2D_HANXIN (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, LPTSTR pstr);

参数：

x ： ●X 座标。

y ： ●Y 座标。

w ： 最大打印宽度, 以点(dots)为单位.

v ： 最大打印高度,以点(dots)为单位.

o ： 设置旋转方向. 范围值(0 到 3)
(0-0°,1--90°,2--180°, 3--270°)

r ： 设置放大倍数，以点(dots)为单位.范围值：(0 - 30)
(0-放大 1 倍， 1-放大 2 倍 2-放大 3 倍……依此类推)。

m ： 汉信码编码模式选择. 范围值(0 到 6)
0 是选择数字模式，
1 是选择 TEXT 模式，
2 是选择二进制模式，
3 是选择常用汉字 1 区模式编码
4 是选择常用汉字 2 区模式编码
5 是 GB 18030 双字节区模式
6 是 GB 18030 四字节模式编码

g ： 汉信码纠错等级选择. 范围值(0 到 3)
0 是'L1'等级
1 是'L2'等级
2 是'L3'等级
3 是'L4'等级

s ： 汉信码掩模图形选择.范围值(0 到 3)
0 是掩模图形 00
1 是掩模图形 01
2 是掩模图形 10
3 是掩模图形 11

pstr ： ●资料字串。

传回值： 0 -> OK.

其它返回值请参考章节： **CDFPSK.dll 错误返回值解析**。

范例：PTK_DrawBar2D_HANXIN(50, 30, 0, 0, 0, 5, 1, 3, 2,"POSTEK");

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);

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_PcxGraphicsList ( );

PTK_PcxGraphicsDel

说明：

PTK_PcxGraphicsDel 函数作用是删除存储在打印机 RAM 或 FLASH 存储器里的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_PcxGraphicsDel (“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 (LPTSTR pcxname, LPTSTR 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_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_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, LPTSTR 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_BmpGraphicsDownload

说明：
PTK_BmpGraphicsDownload 函数的作用是先转换 BMP 到 PCX 格式，然后将 PCX 格式的图形到打印机。

原型：
int PTK_BmpGraphicsDownload (LPTSTR pcxname, LPTSTR pcxpath, unsigned int iDire);

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: BMP 图形文件在 PC 机存储器里的路径;
iDire ：旋转方向 0->0° 1->90° 2->180° 3->270° 其他->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：以下为打印 BMP 图形流程
PTK_PcxGraphicsDel (“BMPA”);
PTK_BmpGraphicsDownload (“BMPA”, “c:\\test1111.bmp”, 0);
PTK_DrawPcxGraphics (100, 50, “BMPA”);
注意：BMP 图形仅支持单色位图形

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，该函数的作用与 PTK_PcxGraphicsList 方法相同。

原型：

int PTK_BinGraphicsList (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_BinGraphicsList ( );

PTK_BinGraphicsDel

说明：

PTK_BinGraphicsDel 函数作用是删除存储在打印机 RAM 或 FLASH 存储器里的一个或所有图形，此图形可以是 Bin 格式或 PCX 格式的，该函数的作用与 PTK_PcxGraphicsDel 相同。

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_BinGraphicsDel (“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 (LPTSTR 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；

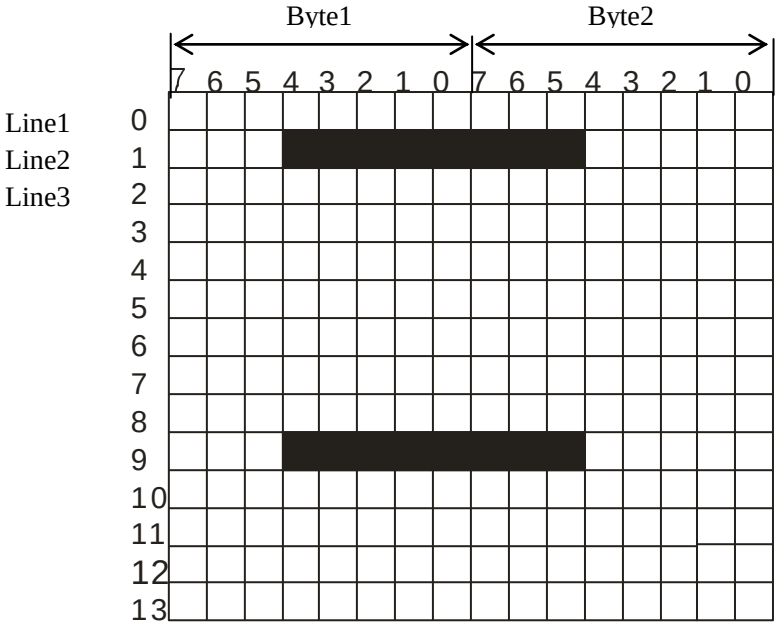
Gdata([…raster data…])：二进制图形数据,数据量大小= pbyte * pH (Bytes)。

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, LPTSTR 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_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；

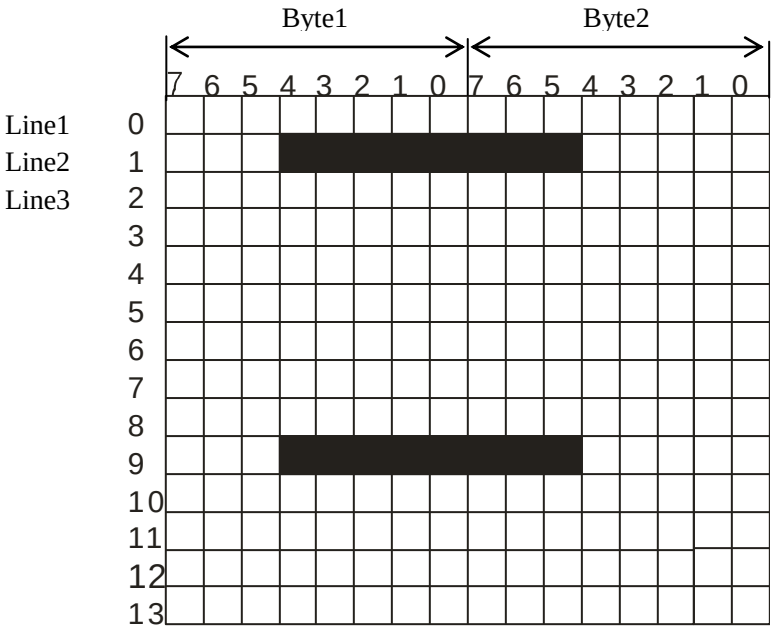
Ph：图形的高度，以点(dots)为单位；

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_DrawRectangle

说明：

PTK_DrawRectangle 函数的作用是画距形。

原型：

int PTK_DrawRectangle (unsigned int px, unsigned int py,
unsigned int thickness, unsigned 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_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_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 (TCHAR 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_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_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_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_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_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,
TCHAR 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：TE2DOKLUO46MNY37，起始值是”T062”，则 T062，T06K，T060，…；

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.dll 错误返回值解析**。

范例:

PTK_DefineCounter (0,6,’N’,”+1”,"\”Enter\” Code:");

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,
TCHAR 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_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_DownloadInitVar

说明：
PTK_DownloadInitVar 函数的作用是初始化变量或系列号变量。
需跟在 PTK_Download()函数后使用。

原型：
int PTK_DownloadInitVar(LPTSTR pstr);

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_DownloadInitVar("123456");

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_SendFile

说明：
PTK_SendFile 函数的作用是发送文件到打印机。

原型：
int PTK_SendFile(LPTSTR FilePath);

参数：
FilePath：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_SendFile("cmdfile.txt");

PTK_GetUSBID

说明：
PTK_GetUSBID 函数的作用是获取打印机 USBID 号。

原型：
int PTK_GetUSBID (LPTSTR USBDeviceSerial);

参数：
USBDeviceSerial：存储获取的 USBID 号。

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

注意： USBID 号仅通过 USB 端口获取。

范例：
```
TCHAR  USBIDString[10] = {_T('0')};
PTK_GetUSBID (USBIDString);
```

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_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_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 (TCHAR 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_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_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_FeedMedia

说明：
PTK_FeedMedia 函数作用命令打印机走一行标签。

原型：
int PTK_FeedMedia (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_ FeedMedia ( );

PTK_MediaDetect

说明：
PTK_ MediaDetect **函数的作用校准纸张探测器。**

原型：
int PTK_MediaDetect (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_ MediaDetect ( );

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 (unsigned int 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_CutPageEx

说明：

PTK_CutPageEx 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_CutPageEx (unsigned int 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_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_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, int TimeOut);

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。

TimeOut: 接收串口超时等待时间；单位为：100ms

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();
PTK_ ErrorReport (1, 1, 9600, TRUE, 10);

…….
```

2. 透过 COM1 发送指令，并从 COM1 读取当前错误状态代码：

```
OpenPort(4);
….
ClosePort();
PTK_ ErrorReport (4, 1, 9600, TRUE, 10 );
…….

…….
```

PTK_ErrorReportUSB

PTK_ErrorReportUSB 函数的作用是发送错误查询指令到打印机并且从指定 USB 端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！

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_ErrorReportUSB(int USBport);

参数：

USBport：发送数据的端口；

1： 表示打开 USB001。

2： 表示打开 USB002。

3： 表示打开 USB003。

……

255: 表示当且仅有一台 Postek 打印机时，默认打开这台 Postek 打印机。

返回值：

返回上述打印机状态

Error/Status code
00
01
82
83
86
87
88
99

其它返回值请参考章节： **CDFPSK.dll 错误返回值解析**。

范例：PTK_ErrorReportUSB(255);

OpenPort(255);

….

ClosePort();

PTK_RWRFIDLabel

说明：
PTK_RWRFIDLabel 函数的作用写 RFID 标签。

原型：
int PTK _RWRFIDLabel (int nRWMode, int nWForm,
int nStartBlock, int nWDataNum,
int nWArea,LPTSTR pstr)

参数：
nRWMode: RFID 操作方式. 0—(预留：暂无功能)；1—写 RFID；
nWForm: RFID 写入格式. 0—HEX（十六进制)；1—ASCII；
nStartBlock: 写入起始块.
备注:如针对 EPC 区进行写入操作， 单位:字(2 个字节)；
nWDataNum: 写入字节数.
nWArea: 写入区域. 0—Reserved（保留区)；1—EPC；2—TID；3—USER；
pstr: 一个常量字符串。(格式由参数 P2 限制）
备注:如果 nWForm 为 ASCII 格式，写入数据长度必须以字(2 个字节)为单位，有效数据长度为 2 个字节的整数倍；
如果 nWForm 为 16 进制格式，写入数据长度以 4 个字节为单位，有效数据长度为 4 个字节的整数倍。

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：
例 1:
int return = PTK_RWRFIDLabel(1,0,2,6,1,"313233343536");
输出结果：

读取 EPC 区（Start=2，size=3word）
313233343536

例 2：
int return = PTK_RWRFIDLabel(1，1，0，6，3，“POSTEK”);

输出结果：

读取 USER 区（Start=0，size=3word）

504F5354454B

PTK_SetRFLabelPWAndLockRFLabel

说明：

PTK_SetRFLabelPWAndLockRFLabel 函数的作用设置 RFID 标签密码和锁定 RFID 标签。

原型：

int PTK_SetRFLabelPWAndLockRFLabel(int nOperationMode, int OperationnArea,LPTSTR pstr)

参数：

nOperationMode：操作方式，0—解锁；1—锁定；2—完全解锁；3—完全锁定；4—密码写入

OperationnArea：操作区域，0—销毁密码区；1—访问密码区；2—EPC；3—TID；4—USER

pstr： 一个常量字符串。（格式限制为 8 位 HEX 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：

例 1：

int return = PTK_SetRFLabelPWAndLockRFLabel(1,1,"73BE115B");

输出结果：

读取访问密码区（password=“00000000”）

Cannot Read

读取访问密码区（password=“73BE115B”）

73BE115B

例 2：

int return = PTK_SetRFLabelPWAndLockRFLabel(4，0，“5462EF21”)；

输出结果：

读取销毁密码区

5462EF21

PTK_SetRFID

说明：

PTK_SetRFID 函数的作用是 RFID 设置指令。

原型：

int PTK_SetRFID(int nReservationParameters, int nReadWriteLocation,
int ReadWriteArea, int nMaxErrNum, int nErrProcessingMethod)

参数：

nReservationParameters：预留参数，默认输入 0

nReadWriteLocation：RFID 读写位置，范围：0-999，默认为 0. 单位 mm.

ReadWriteArea：RFID 读写区域 ，范围：0-99，默认为 0. 单位 mm.

nMaxErrNum：最大错误数 0~9 默认 1

nErrProcessingMethod：RFID 出错处理方式 0-继续 1-暂停 3-停止 默认 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：

例 1：

int return = PTK_SetRFID(0,0,0,3,0);

PTK_SetFontGap

说明：

PTK_SetFontGap 函数的作用是调整打印文字字间距。

原型：

int PTK_SetFontGap(int gap);

参数：

gap：字间距调节值，以点(dots)为单位.取值范围为-99 — 99.

注：打印机内置字体（包括下载字体）有初始间距，通过设置 g 指令可以调节字间距大小，实际字间距 = 初始字间距 + 可调节字间距. 该指令仅对具备调整打印机文字间距功能机型有效.

提示：以下是不同分辨率的单位转换关系

203DPI: 1mm = 8 dot;

300DPI: 1mm = 11.8dot;

600DPI: 1mm = 23.6 dot;

例如：需要将字体间距设置为 1mm ,则在 203 DPI 下设置为 8，300DPI 下设置为 12，600DPI 下设置为 24

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：

例 1：

int return = PTK_SetFontGap (10);

PTK_SetBarCodeFontName

说明：

PTK_SetBarCodeFontName 函数的作用是**设置可打印条码文字的下载字体**。

原型：

int PTK_SetBarCodeFontName (UCHAR Name,unsigned int FontW,unsigned int FontH);

参数：

Name：设置条码内置打印字体名称说明，取值范围：A-Z

该参数和 PTKRenameDownloadFont 函数中的参数二必须一致。

FontW：设置字体宽度，取值范围 0~65535。

FontH：设置字体高度，取值范围 0~65535。

注意：该函数用于将下载 TrueType 字体作为条码内置字体进行打印，字体大小将随条码的宽度进行自动调整，**该函数仅对具备调用下载字体打印条码文字的功能的机型有效**。

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：

例 1：

int return = PTK_SetBarCodeFontName (‘A’ ,32,32);

PTK_RenameDownloadFont

说明：

PTK_RenameDownloadFont 函数的作用是将下载到打印机的字体进行重命名字体 ID

原型：

int PTK_RenameDownloadFont (unsigned int StoreType,UCHAR Fontname,LPTSTR DownloadFontName);

参数：

StoreType：下载字体在打印机中的存储位置，0：SDRAM, 1：FLASH.

提示：下载到打印机 SDRAM 中的字体在打印机断电后被擦除，下载到 FLASH 中的字体在打印机断电后仍保存.

Fontname ：重命名下载字体 ID，取值范围：A-Z。

DownloadFontName：下载字体在打印机中的名称。

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：

int return = PTK_RenameDownloadFont (1,'A',"arial");

PTK_SetCharSets

说明：

PTK_SetCharSets 函数的作用是设置字符集

原型：

int PTK_SetCharSets (unsigned int BitValue,UCHAR CharSets,LPTSTR CountryCode);

参数：

BitValue：数据比特值;8 表示 8 位码,7 表示 7 位码.

CharSets： 字符集.

CountryCode：可编程键盘(KDU)的国家编码.

8 位码 (p1=8)	字符集 (Code page)	7 位码 (p1=7)	字符集
0	English (437)	0	USASCII
1	Latin 1 (850)	1	British
2	Slavic (852)	2	German
3	Portugal (860)	3	French
4	Canadian/French (863)	4	Danish
5	Nordic (865)	5	Italian
		6	Spanish
		7	Swedish
		8	Swiss

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：

Int return = PTK_SetCharSets (8,'N',"001");

PTK_ReadRFTagData

说明：

PTK_ReadRFTagData 函数的作用是发送读取 RFID 标签数据指令到打印机并且从指定串口接收打印机当前读取的 RFID 标签信息。

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！

原型：

int PTK_ReadRFTagData(unsigned int wPort, unsigned int rPort, DWORD BaudRate,BOOL HandShake, unsigned int TimeOut, unsigned int nDataBlock, unsigned int nRFPower,BOOL bFeed,LPTSTR strRFData);

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。

TimeOut： 接收串口超时等待时间；单位为：ms

nDataBlock：选择数据区域；0：TID ， 1：EPC ， 2：TID+EPC.

nRFPower： 设置读功率;范围：0 ~ 30dBm；设置为 0 时，采用系统默认 23dBm 读取功率;

bFeed： 读取后是否向前走一张标签;

TRUE：向前走一张标签有效，

FALSE：向前走一张标签无效。

strRFData： 用于存储获取 RFID 标签数据信息。

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 发送打印指令，从 COM1 读取当前 RFID 标签 EPC 信息：

```
LPTSTR strRFData;
OpenPort(1);
….
ClosePort();
PTK_ReadRFTagData    (1, 1, 38400, TRUE, 300,1,0,TRUE,strRFData);

…….
```

2. 透过 COM1 发送指令，并从 COM1 读取当前 RFID 标签 EPC 信息：

```
LPTSTR strRFData;
OpenPort(4);
….
ClosePort();
PTK_ReadRFTagData (4, 1, 38400, TRUE, 300,1,0, TRUE,strRFData);
…….
```

PTK_ReadRFTagDataUSB

说明：

PTK_ ReadRFTagDataUSB 函数的作用是从指定 USB 端口发送读取 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签信息。

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！

原型：

```
int PTK_ReadRFTagDataUSB(unsigned int usbPort, unsigned int nDataBlock, unsigned int nRFPower,
                         BOOL bFeed, LPTSTR strRFData);
```

参数：

usbPort：发送数据的端口；

1： 表示打开 USB001。

2： 表示打开 USB002。

3： 表示打开 USB003。

…….

255: 表示当且仅有一台 Postek 打印机时，默认打开这台 Postek 打印机。

nDataBlock：选择数据区域；0：TID ， 1：EPC ， 2：TID+EPC.

nRFPower： 设置读功率;范围：0 ~ 30dBm；设置为 0 时，采用系统默认 23dBm 读取功率;

bFeed： 读取后是否向前走一张标签;

TRUE：向前走一张标签有效，

FALSE：向前走一张标签无效。

strRFData： 用于存储获取 RFID 标签数据信息。

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：

```
LPTSTR strRFData;
OpenPort(255);
….
ClosePort();
PTK_ReadRFTagDataUSB(255, 1,0,TRUE,strRFData);
```

PTK_ReadRFTagDataNet

说明：

PTK_ ReadRFTagDataNet 函数的作用是从指定 TCP/IP 端口发送读取 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签信息。

原型：

```
int PTK_ReadRFTagDataNet(LPTSTR IPAddress, unsigned int Port, unsigned int nFeedbackPort,
                         unsigned int nRFPower,BOOL bFeed, LPTSTR strRFData);
```

参数：

IPAddress： 打印机 IP 地址.

Port： 打印机网络端口.

nDataBlock：选择数据区域；0：TID ， 1：EPC ， 2：TID+EPC.

nRFPower： 设置读功率;范围：0 ~ 30dBm；设置为 0 时，采用系统默认 23dBm 读取功率;

bFeed： 读取后是否向前走一张标签;
TRUE：向前走一张标签有效，
FALSE：向前走一张标签无效。

strRFData： 用于存储获取 RFID 标签数据信息。

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：

```
LPTSTR strRFData;
PTK_ReadRFTagDataNet("199.9.10.230",9100,1, 0,TRUE,strRFData);
```

PTK_Connect

说明：
PTK_Connect 函数的作用是建立打印机网络连接。

原型：
int PTK_Connect(LPTSTR IPAddress, unsigned int Port);

参数：
IPAddress：打印机 IP；
Port：打印机网络端口

返回值：
0 -> OK;
其它返回值最下错误返回值 ：**CDFPSK.dll 错误返回值解析**。

范例：

```
nErrorcode = PTK_Connect (“199.9.10.2”,9100); //表示打印机网络连接
// 标签及打印机设置部分方法如下，以下为例子，根据您的实际需求调用。
nErrorcode = PTK_SetLabelHeight (376,20,0, FALSE);
nErrorcode = PTK_SetLabelWidth (800);
nErrorcode = PTK_SetDarkness( 10 );
nErrorcode = PTK_SetPrintSpeed( 5 );
…….
// 打印内容 以下为例子，根据您的实际需求调用。
nErrorcode = PTK_DrawBarcode(300, 23, 0, _T("1"), 2, 2, 50, 'B', _T("123456789")); // 打印条码
nErrorcode = PTK_DrawText(80, 168, 0, 3, 1, 1, 'N', _T("Internal Soft Font")); // 打印文字
……
// 命令打印机执行打印工作
  nErrorcode=PTK_PrintLabel(1,1);
  PTK_CloseConnect();
```

PTK_CloseConnect

说明：
PTK_CloseConnect 函数的作用是断开使用 PTK_Connect 函数建立的打印机网络连接。
用户在对打印机操作完成之后，建议调用 PTK_CloseConnect 关闭通讯端口；

原型：
void PTK_CloseConnect();

参数：无

返回值： 无

范例： PTK_CloseConnect( );

PTK_ErrorReportNet

PTK_ErrorReportNet 函数的作用是指定 TCP/IP 端口发送错误查询指令到打印机并且接收和分析打印机当前错误代码。

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_ErrorReportNet(LPTSTR PrintIPAddress, unsigned int PrinterNetPort);

参数：
PrintIPAddress： 打印机 IP 地址.
PrinterNetPort： 打印机网络端口.

返回值：

返回上述打印机状态

Error/Status code
00
01
82
83
86
87
88
99

其它返回值请参考章节： **CDFPSK.dll 错误返回值解析**。

范例：int nPrintStatus;

nPrintStatus = PTK_ReadSerialNumberNet ("199.9.10.230",9100);

PTK_ReadPrintConifUSB

说明：

PTK_ReadPrintConifUSB函数的作用是通过USB端口将打印机信息反馈给主机（反馈信息见参数2说明）

原型：

int PTK_ReadPrintConifUSB(unsigned int usbPort,unsigned int nInfoType,LPTSTR strData)

参数说明:

usbPort: 设置反馈 USB 端口号;

nInfoType: 设置反馈信息类型，如下表：

p2	反馈信息格式说明
1	打印机型号，软件版本，固件编号，打印机分辨率,
2	工作方式(0 热敏/1 热转印)，纸张探测方式（0 穿透/1 下反射/2 上反射），碳带探测器(1 有效/0 无效),撕纸模式(1 开启/0 关闭),切纸模式(1 开启/0 关闭),打印黑度（1～20）
3	IP ADDRESS，SUBNET MASK,GATEWAY,BASE RAW PORT,MAC ADDRESS
4	打印线到探测器距离，切刀到打印线距离，撕纸口到打印线距离，标签起始位置到下一张标签起始位置的距离
5	RFID（1 有效/0 无效），RFID POWER ，RFID 探测偏移距离(单位:毫米)，当前 RFID 标签高度(单位:dot)

strData： 用于接收获取的打印机信息。

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：LPTSTR strData;

PTK_ReadPrintConifUSB（255,1, strData）;

PTK_ReadPrintConifNet

说明：

PTK_ReadPrintConifNet函数的作用是通过网口将打印机信息反馈给主机（反馈信息见参数2说明）

原型：

int PTK_ReadPrintConifNet(LPTSTR PrintIPAddress, unsigned int PrinterNetPort,
unsigned int nInfoType,LPTSTR strData)

参数说明:

PrintIPAddress： 打印机IP地址.

PrinterNetPort： 打印机网络端口.

nInfoType：设置反馈信息类型，如下表：

p2	反馈信息格式说明
1	打印机型号，软件版本，固件编号，打印机分辨率,
2	工作方式(0 热敏/1 热转印)，纸张探测方式（0 穿透/1 下反射/2 上反射），碳带探测器(1 有效/0 无效),撕纸模式(1 开启/0 关闭),切纸模式(1 开启/0 关闭)，打印黑度（1～20）
3	IP ADDRESS，SUBNET MASK,GATEWAY,BASE RAW PORT,MAC ADDRESS
4	打印线到探测器距离，切刀到打印线距离，撕纸口到打印线距离，标签起始位置到下一张标签起始位置的距离
5	RFID（1 有效/0 无效），RFID POWER ，RFID 探测偏移距离(单位:毫米)，当前 RFID 标签高度(单位:dot)

注意：如采用网络端口反馈功能，默认情况下反馈的 HOST 主机为当前连接 HOST IP 地址，如需更改反馈 IP 地址需配合 NF 一起使用，如例一。

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：LPTSTR strData;

PTK_ ReadPrintConifNet（”199.9.10.196”,1, strData）;

PTK_RFIDCalibrate

说明：

PTK_RFIDCalibrate函数的作用是校准RFID芯片读写位置(支持RFID打印机固件版本V1.73及更高)

注意：打印机固件版本可通过 LCD 屏或打印自检页查看。

原型：

int PTK_RFIDCalibrate();

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：

支持并口，USB，串口	**网口**
OpenPort(255); PTK_RFIDCalibrate(); ClosePort();	PTK_Connect (“199.9.10.2”,9100) PTK_RFIDCalibrate(); PTK_CloseConnect();

PTK_RWHFLabel

说明：

PTK_RWHFLabel函数的作用是读写高频RFID标签

原型：

int PTK_RWHFLabel(TCHAR ptext, unsigned int nStartBlock, unsigned int nBlockNum,LPTSTR pstr,BOOL Varible)

参数：

ptext: 高频标签读写设置 R:保留 W:写操作

R:对应ASCII值 82

W:对应ASCII值 87

nStartBlock：操作起始块(first block number).

nBlockNum：操作块的个数(Number of blocks).

pstr：一个常量字符串， 用户可以用”DATA”,Cn,Vn 自由排列组合字符串,

“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。

Vn: 变量字符串，此变量字符串必须已经定义。

如:”data1”CnVn”data2”。

注意：此参数仅当 ptext 参数为写操作时生效。

Varible： TRUE 表示 pstr 参数的字符串组合中包含有变量操作,需加‘\”’将打印常量字符串包含。

例如：PTK_RWHFLabel(‘W’,5,1,”\”123456789\”C0”,TRUE);

PTK_RWHFLabel('W',5,1,"C0",TRUE);
C0 并非字符串常量, 解释如下
字母 C 表示设置一个序列号变量。
数字 0 表示对应的序列号编号。
FALSE 表示字符串当中不包含有变量操作，不需加'\"'。
例如：PTK_RWHFLabel('W',5,1,"1234",FALSE); 1234 为一个常量字符串
此参数仅当 ptext 参数为写操作时生效。

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:

写高频标签数据：USB 端口

```
OpenPort(255)
PTK_ClearBuffer();
PTK_RWHFLabel('W',5,1,"1234", , FALSE);
PTK_PrintLabel(1,1);
```

以序列号方式写入 高频 RFID 电子标签

```
OpenPort(255)
PTK_ClearBuffer();
  此处省略打印机参数配置
PTK_FormDel("LSFORM");                          //删除 Form " LSFORM "
PTK_FormDownload("LSFORM");                     //存储 Form " LSFORM "
PTK_DefineCounter(0,10,78,"+1","InputC0");   //定义一个序列号变量 C0
PTK_DefineCounter(1,10,78,"+1"," InputC1");   //定义一个序列号变量 C1
PTK_RWHFLabel('W',4,1,"C0", TRUE);              //写入序列号变量写入高频 RFID 第 4 块
PTK_RWHFLabel('W',5,1,"C1", TRUE);              //写入序列号变量写入高频 RFID 第 5 块

PTK_FormEnd();                                  //结束存储表格
PTK_ExecForm("LSFORM");                         // 运行指定的表格:LSFORM
PTK_Download();                                 // 下载变量或系列号变量
// 以下按序号变量已定义的顺序初始化（此例子定义顺序为 C0，C1）

PTK_DownloadInitVar("1111");                    // 初始化系列号变量 C0
PTK_DownloadInitVar("2222");                    // 初始化系列号变量 C1
PTK_PrintLabel(1,1);
```

PTK_SetHFRFID

说明

PTK_SetHFRFID 函数的作用设置高频 RFID 标签。

原型：

int PTK_SetHFRFID(TCHAR pWForm, int nProtocolType,int nMaxErrNumd);

参数:

nWForm：读写数据类型。

A: ASCII；

H:十六进制方式；

默认为ASCII 格式。

nProtocol：协议类型。

1: ISO 15693 协议；

2: ISO 14443 协议；

3: NFC 协议（暂保留）。

nMaxErrNumd：读写操作失败尝试次数，最大次数失败后打印“Void”标识

默认值：3。

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:

```
OpenPort(255);
PTK_ClearBuffer();
…
PTK_SetHFRFID(_T('A'),1,3);
PTK_RWHFLabel('W',5,1,"1234", , FALSE);

ClosePort();
```

范例说明：向标签第五块写入 ASCII 值“1234”

```
OpenPort(255);
PTK_ClearBuffer();
…
PTK_SetHFRFID(_T('H'),1,3);
PTK_RWHFLabel('W',5,1,"31323334", , FALSE);

ClosePort();
```

范例说明：向标签第五块写入 16 进制数据“31323334”

PTK_ReadHFTagData

说明

PTK_ReadHFTagData 函数的作用是从指定串口接收打印机当前读取的高频 RFID 标签信息。

原型：

int PTK_ReadHFTagData(unsigned int wPort, unsigned int rPort, DWORD BaudRate,
BOOL HandShake, unsigned int TimeOut,unsigned int nStartBlock,
unsigned int nBlockNum,TCHAR pFeed,LPTSTR strRFData);

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。

TimeOu： 接收串口超时等待时间；单位为：ms

nStartBlock：电子标签读取起始块(first block number)

nBlockNum：电子标签读取块的个数(从 nStartBlock 开始，往后 nBlockNum 块)

bFeed：读取数据后是否向前走一张标签;

Y：表示读取信息后进行 feed 一张标签。

N：读取信息后不进行 feed 动作

strRFData：用于存储获取 RFID 标签数据信息。

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:

3. 通过 LPT1 发送打印指令，从 COM1 读取当前高频 RFID 标签数据：

LPTSTR strRFData;

PTK_ReadHFTagData (1, 1, 38400, TRUE, 300,5, 1, _T('Y') , strRFData);

范例说明：从第五块开始，读取 1 块（即第五块）数据通过串口返回,返回后前进一张标签。

…….

4. 通过 COM1 发送指令，并从 COM1 读取当前高频 RFID 标签数据：

LPTSTR strRFData；

PTK_ReadHFTagData (4, 1, 38400, TRUE, 300, 5, 5, _T('N') , strRFData);

范例说明：从第五块开始，读取 5 块（即第五， 六，七，八，九块）数据通过串口返回。返回后不进纸一张标签。

PTK_ReadHFTagDataUSB

说明

PTK_ReadHFTagDataUSB 函数的作用是从指定 USB 端口发送读取高频 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签信息。

原型：

int PTK_ReadHFTagDataUSB(unsigned int usbPort,unsigned int nStartBlock,
unsigned int nBlockNum,TCHAR pFeed,LPTSTR strRFData);

参数

usbPort：发送数据的端口；

1： 表示打开 USB001。

2： 表示打开 USB002。

3： 表示打开 USB003。

……

255: 表示当且仅有一台 Postek 打印机时，默认打开这台 Postek 打印机。

nStartBlock：电子标签读取起始块(first block number)

nBlockNum：电子标签读取块的个数(从 nStartBlock 开始，往后 nBlockNum 块)

bFeed：读取数据后是否向前走一张标签；

Y：表示读取信息后进行 feed 一张标签。

N：读取信息后不进行 feed 动作

strRFData： 用于存储获取 RFID 标签数据信息。

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:

LPTSTR strRFData；

PTK_ReadHFTagDataUSB(255,5,1,_T('Y'),strRFData);

范例说明：从第五块开始，读取 1 块（即第五块）数据通过 USB 端口返回,返回后前进一张标签。

LPTSTR strRFData；

PTK_ReadHFTagDataUSB(255,5,5,_T('N'),strRFData);

范例说明：从第五块开始，读取 5 块（即第五， 六，七，八，九块）数据通过 USB 端口返回，返回后

不进纸一张标签。

PTK_ReadHFTagUID

说明

PTK_ReadHFTagUID 函数的作用是从指定串口接收打印机当前读取取的高频 RFID 标签 UID 信息。

原型：

int PTK_ReadHFTagUID(unsigned int wPort, unsigned int rPort, DWORD BaudRate,
BOOL HandShake, unsigned int TimeOut,TCHAR pFeed,LPTSTR strRFData);

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。

TimeOut: 接收串口超时等待时间；单位为：ms

bFeed：读取数据后是否向前走一张标签;

Y：表示读取信息后进行 feed 一张标签。

N：读取信息后不进行 feed 动作

strRFData： 用于存储获取 RFID 标签数据信息。

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. 通过 LPT1 发送打印指令，从 COM1 读取当前高频 RFID 标签 UID 数据：

```
LPTSTR strRFData;
PTK_ReadHFTagUID (1, 1, 38400, TRUE, 300, _T('Y') , strRFData);
```

范例说明：读取当前高频 RFID 标签 UID 数据通过串口返回,返回后前进一张标签。

…….

2. 通过 COM1 发送指令，并从 COM1 读取当前高频 RFID 标签数据：

```
LPTSTR strRFData;
PTK_ReadHFTagUID (4, 1, 38400, TRUE, 300, _T('N') , strRFData);
```

范例说明：读取当前高频 RFID 标签 UID 数据通过串口返回,返回后不进纸一张标签。

PTK_ReadHFTagUIDUSB

说明

PTK_ReadHFTagUIDUSB 函数的作用是从指定 USB 端口发送读取高频 RFID 标签数据指令到打印机并接收打印机当前读取的 RFID 标签 UID 信息.

原型：

```
int PTK_ReadHFTagUIDUSB(unsigned int usbPort,TCHAR pFeed,LPTSTR strRFData);
```

参数：

usbPort：发送数据的端口；

1： 表示打开 USB001。

2： 表示打开 USB002。

3： 表示打开 USB003。

……

255： 表示当且仅有一台 Postek 打印机时，默认打开这台 Postek 打印机。

bFeed：读取数据后是否向前走一张标签;

Y：表示读取信息后进行 feed 一张标签。

N：读取信息后不进行 feed 动作

strRFData： 用于存储获取 RFID 标签数据信息。

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:

```
LPTSTR strRFData;
PTK_ReadHFTagDataUSB(255, _T('Y'),strRFData);
```

范例说明：读取当前高频 RFID 标签 UID 数据通过 USB 端口返回,返回后前进一张标签。

```
LPTSTR strRFData;
```

PTK_ReadHFTagDataUSB(255, _T('N),strRFData);

范例说明：读取当前高频 RFID 标签 UID 数据通过 USB 端口返回，返回后不进纸一张标签。

PTK_ReadHFTagDataPrintAuto

说明

PTK_ReadHFTagDataPrintAuto 函数的作用打印过程中先读取高频标签指定位置块的数据内容，然后打印在标签上面。

注意： 此方法需配合 PTK_DrawTextEx 一起使用。

原型：

int PTK_ReadHFTagDataPrintAuto(unsigned int nStartBlock,unsigned int nBlockNum);

参数:

nStartBlock：电子标签读取起始块(first block number)

nBlockNum：读取块的个数(从 nStartBlock 开始，往后 nBlockNum)

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:

```
OpenPort(255);
PTK_ClearBuffer();
…此处省略打印机设置信息
PTK_ReadHFTagDataPrintAuto (5,1);
PTK_DrawTextEx (50,30,0,2,1,1,_T('N'),_T("R"),TRUE);
PTK_PrintLabel (1,1);
ClosePort();
```

PTK_ReadHFTagUIDPrintAuto

说明

PTK_ReadHFTagUIDPrintAuto 函数的作用打印过程中先读取高频标签标签 UID 号，然后打印在标签上面。

注意： 此方法需配合 PTK_DrawTextEx 一起使用。

原型：

int _stdcall PTK_ReadHFTagUIDPrintAuto();

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:

```
OpenPort(255);
PTK_ClearBuffer();
…此处省略打印机设置信息
PTK_ReadHFTagUIDPrintAuto ();
PTK_DrawTextEx (50,30,0,2,1,1,_T('N'),_T("R"),TRUE);
PTK_PrintLabel (1,1);
ClosePort();
```

CDFPSK.dll 错误返回值解析

-1000 to -1011 : OpenPort 函数操作出错;
-1012 to -1025 : 串口读取操作出错;
-1026 to -1028 : SetPCComPort 设置串口错误;
-1029 to -1030 : 写数据错误;

-2000 : PTK_GetInfo 执行出错;
-2001 : PTK_ClearBuffer 执行出错;
-2002 : PTK_SetDarkness 执行出错;
-2003 : PTK_SetPrintSpeed 执行出错;
-2004 : PTK_SetPrintSpeed 参数错误;
-2005 : PTK_SetLabelHeight 执行出错;
-2006 : PTK_SetLabelWidth 执行出错;
-2007 : PTK_SetDirection 执行出错;
-2008 : PTK_SetDirection 参数错误;
-2009 : PTK_SetCoordinateOrigin 执行出错;
-2010 : PTK_PrintLabel 执行出错;
-2011 : PTK_PrintLabel 参数错误;
-2012 : PTK_PrintLabelAuto 执行出错;
-2013 : PTK_PrintLabelAuto 参数错误;
-2014 : PTK_DrawText 执行出错;
-2015 : PTK_DrawText 参数出错;
-2016 : PTK_DrawTextEx 执行出错;
-2017 : PTK_DrawTextEx 参数出错;
-2018 : PTK_DrawTextTrueTypeW 创建 PrinterDC 失败,进行出错处理;;
-2019 : PTK_DrawTextTrueTypeW 分配保存 bitmap 内存出错;;
-2020 : 分配保存当前程序运行的文件路径内存出错;
-2021 : 分配保存 PCX HEAD 文件结构内存出错;
-2022 : 创建 PCX 文件出错;
-2023 : 分配保存 PCX data 内存出错;
-2024 : 保存 PCX data 出错;
-2025 : PTK_DrawBarcode 执行出错;
-2026 : PTK_DrawBarcode 参数错误;
-2027 : PTK_DrawBarcodeEx 执行出错;
-2028 : PTK_DrawBarcodeEx 参数错误;
-2029 : PTK_DrawBar2D_DATAMATRIX 执行出错;
-2030 : PTK_DrawBar2D_DATAMATRIX 执行出错;
-2031 : PTK_DrawBar2D_QR 执行出错;
-2032 : PTK_DrawBar2D_QR 执行出错;
-2033 : PTK_DrawBar2D_QREx 执行出错;
-2034 : PTK_DrawBar2D_MaxiCode 执行出错;
-2035 : PTK_DrawBar2D_MaxiCode 执行出错;
-2036 : PTK_DrawBar2D_Pdf417 执行出错;

-2037 : PTK_DrawBar2D_Pdf417 执行出错;
-2038 : PTK_DrawBar2D_HANXIN 执行出错;
-2039 : PTK_DrawBar2D_HANXIN 执行出错;
-2040 : PTK_PcxGraphicsList 执行出错;
-2041 : PTK_PcxGraphicsDel 分配内存出错;
-2042 : PTK_PcxGraphicsDel 参数错误;
-2043 : PTK_PcxGraphicsDel 执行出错;
-2044 : PTK_PcxGraphicsDownload 分配内存出错;
-2045 : PTK_PcxGraphicsDownload 分配内存出错;
-2046 : PTK_PcxGraphicsDownload 打开文件错误;
-2047 : PTK_PcxGraphicsDownload 参数错误;
-2048 : PTK_PcxGraphicsDownload 执行发送文件信息出错;
-2049 : PTK_PcxGraphicsDownload 执行发送文件内容出错;
-2050 : PTK_DrawPcxGraphics 分配内存出错;
-2051 : PTK_DrawPcxGraphics 参数错误;
-2052 : PTK_DrawPcxGraphics 执行出错;
-2053 : PTK_PrintPCX 执行出错;
-2054 : PTK_BmpGraphicsDownload 分配内存出错;
-2055 : PTK_BmpGraphicsDownload 下载 BMP 位深度错误;
-2056 : PTK_BmpGraphicsDownload 打开文件错误;
-2057 : PTK_BmpGraphicsDownload 参数错误;
-2058 : PTK_BmpGraphicsDownload 执行发送文件信息出错;
-2059 : PTK_BmpGraphicsDownload 执行发送文件内容出错;
-2060 : PTK_BinGraphicsList 执行出错;
-2061 : PTK_BinGraphicsDel 分配内存出错;
-2062 : PTK_BinGraphicsDel 参数错误;
-2063 : PTK_BinGraphicsDel 执行出错;
-2064 : PTK_BinGraphicsDownload 执行发送二进制格式信息出错;
-2065 : PTK_BinGraphicsDownload 执行发送二进制图形内容出错;
-2066 : PTK_RecallBinGraphics 分配内存出错;
-2067 : PTK_RecallBinGraphics 执行出错;
-2068 : PTK_RecallBinGraphics 参数错误;
-2069 : PTK_DrawBinGraphics 发送二进制格式信息出错;
-2070 : PTK_DrawBinGraphics 发送二进制图形内容出错;
-2071 : PTK_DrawRectangle 执行出错;
-2072 : PTK_DrawLineXor 执行出错;
-2073 : PTK_DrawLineOr 执行出错;
-2074 : PTK_DrawDiagonal 执行出错;
-2075 : PTK_DrawWhiteLine 执行出错;
-2076 : PTK_SoftFontList 执行出错;
-2077 : PTK_SoftFontDel 参数错误;
-2078 : PTK_SoftFontDel 执行出错;
-2079 : PTK_FormList 执行出错;
-2080 : PTK_FormDel 分配内存出错;

-2081 : PTK_FormDel 参数错误;
-2082 : PTK_FormDel 执行出错;
-2083 : PTK_FormDownload 分配内存出错;
-2084 : PTK_FormDownload 参数错误;
-2085 : PTK_FormDownload 执行出错;
-2086 : PTK_FormEnd 执行出错;
-2087 : PTK_ExecForm 分配内存出错;
-2088 : PTK_ExecForm 参数错误;
-2089 : PTK_ExecForm 执行出错;
-2090 : PTK_DefineCounter 分配内存出错;
-2091 : PTK_DefineCounter 分配内存出错;
-2092 : PTK_DefineCounter 执行出错;
-2093 : PTK_DefineCounter 参数错误;
-2094 : PTK_DefineVariable 执行出错;
-2095 : PTK_DefineVariable 提示内容参数错误;
-2096 : PTK_DefineVariable 其他参数错误，请查看函数说明;
-2097 : PTK_Download 执行出错;
-2098 : PTK_DownloadInitVar 分配内存出错;
-2099 : PTK_DownloadInitVar 执行出错;
-2100 : PTK_SendFile 分配内存失败;
-2101 : PTK_SendFile 打开文件失败;
-2102 : PTK_SendFile 执行写数据出错;
-2103 : PTK_GetUSBID 执行出错;
-2104 : PTK_DisableBackFeed 执行出错;
-2105 : PTK_EnableBackFeed 执行出错;
-2106 : PTK_PrintConfigunation 执行出错;
-2107 : PTK_SetPrinterState 执行出错;
-2108 : PTK_SetPrinterState 参数错误;
-2109 : PTK_DisableErrorReport 执行出错;
-2110 : PTK_EnableErrorReport 执行出错;
-2111 : PTK_EnableFlASH 执行出错;
-2112 : PTK_DisableFLASH 执行出错;
-2113 : PTK_FeedMedia 执行出错;
-2114 : PTK_MediaDetect 执行出错;
-2115 : PTK_CutPage 执行出错;
-2116 : PTK_CutPageEx 执行出错;
-2117 : PTK_Reset 执行出错;
-2118 : PTK_FeedBack 执行出错;
-2119 : PTK_ErrorReport 获取反馈超时;
-2120 : PTK_ErrorReportUSB 打开 USB 端口失败;
-2121 : PTK_ErrorReportUSB 获取打印机反馈失败;
-2122 : PTK_StringDownload 分配内存失败;
-2123 : PTK_StringDownload 执行出错;
-2124 : PTK_RWRFIDLabel 分配内存失败;

-2125 : PTK_RWRFIDLabel 执行出错;
-2126 : PTK_SetRFLabelPWAndLockRFLabel 分配内存失败;
-2127 : PTK_SetRFLabelPWAndLockRFLabel 执行出错;
-2129 : PTK_SetRFIDk 执行出错;
-2130 : PTK_SetFontGap 执行出错;
-2131 : PTK_SetBarCodeFontName 执行出错;
-2132 : PTK_SetCharSets 执行出错;
-2133 : PTK_RenameDownloadFont 执行出错;
-2134 : PTK_DrawBar2D_DATAMATRIX 分配内存失败;
-2135 : PTK_DrawBar2D_QR 分配内存失败;
-2136 : PTK_DrawBar2D_QREx 分配内存失败;
-2137 : PTK_DrawBar2D_MaxiCode 分配内存失败;
-2138 : PTK_DrawBar2D_Pdf417 分配内存失败;
-2139 : PTK_DrawBar2D_HANXIN 分配内存失败;
-2140 : PTK_BmpGraphicsDownload 打开文件失败;
-2141 : PTK_ErrorReport 打开写端口失败;
-2142 : PTK_BinGraphicsDownload 分配内存出错;
// 新增加
-2143 : PTK_Connect 初始化网络连接错误;
-2144 : PTK_Connect 打印机网络端口参数错误;
-2145 : PTK_Connect 打印机 IP 参数错误;
-2146 : PTK_Connect 连接打印机错误;
-2147 to -2148: PTK_Connect 分配空间失败;
-2149 to -2151: 网口方式写数据出错;
-2152 : PTK_ReadRFTagDataNet 连接网络打印机出错;
-2153 : PTK_ReadRFTagDataNet 写数据出错;
-2154 : PTK_ReadRFTagData 打开写发送获取 RFID 信息指令端口错误;
-2155 : PTK_ReadRFTagData 发送“获取 RFID 信息指令”错误;
-2156 : PTK_ReadRFTagData 读取 RFID 超时;
-2157 : PTK_ReadRFTagDataUSB 打开端口失败;
-2158 : PTK_ReadRFTagDataUSB 发送“获取 RFID 信息指令”错误;
-2159 : PTK_ReadDateUsb 分配读取空间失败;
-2160 : PTK_ReadDateUsb 读取数据信息超时;
-2161 : PTK_SendFile 读文件数据出错;
-2162 : PTK_ReadSerialNumberNet 连接网络打印机出错;
-2163 : PTK_ReadSerialNumberNet 写数据出错;
-2164 : PTK_ErrorReportNet 连接网络打印机出错;
-2165 : PTK_ErrorReportNet 写数据出错;
-2166 : PTK_RFIDCalibrate 写数据出错;
-2169 : PTK_ReadPrintConifUSB 打开端口失败;
-2170 : PTK_ReadPrintConifUSB 写数据出错;
-2173 : PTK_ReadPrintConifNet 连接网络打印机出错;
-2174 : PTK_ReadPrintConifNet 写数据出错;
-2177 : PTK_ErrorReportNet 未取到打印机状态;

-2178 ： PTK_RWHFLabel 执行出错
-2179 ： PTK_RWHFLabel 读写参数错误
-2180 ： PTK_SetHFRFID 执行出错
-2181 ： PTK_SetHFRFID 数据类型或协议类型出错
-2182 ： PTK_ReadHFTagData读取后是否向前走一张标签参数出错
-2183 ： PTK_ReadHFTagData打开打印机端口出错
-2184 ： PTK_ReadHFTagData发送“获取高频RFID数据指令”错误；
-2185 ： PTK_ReadHFTagData读取高频RFID数据超时；
-2186 ： PTK_ReadHFTagDataUSB读取后是否向前走一张标签参数出错
-2187 ： PTK_ReadHFTagDataUSB打开端口失败
-2188 ： PTK_ReadHFTagDataUSB发送“获取高频RFID数据指令”错误；
-2189 ： PTK_ReadHFTagUID读取后是否向前走一张标签参数出错；
-2190 ： PTK_ReadHFTagUID打开端口失败
-2191 ： PTK_ReadHFTagUID发送“获取高频RFID UID数据指令”错误；
-2192 ： PTK_ReadHFTagUID读取高频RFID数据超时；
-2193 ： PTK_ReadHFTagUIDUSB读取后是否向前走一张标签参数出错；
-2194 ： PTK_ReadHFTagUID打开端口失败
-2195 ： PTK_ReadHFTagUIDUSB发送“获取高频RFID UID数据指令”错误；
-2196 ： PTK_ReadHFTagDataPrintAuto执行出错
-2197 ： PTK_ReadHFTagUIDPrintAuto 执行出错

-3000 to -3084 ： 端口未打开或已经关闭；