

POSTEK PSKPrn 控件 使用手册

POSTEK

条码标签打印机

Version 2.0.0.5

深圳市博思得科技发展有限公司

二〇一五年

API 函数库文件说明

名称: PSKPrn.ocx

版本编号: 2.0.0.5150722

版权所有: ©2015 深圳市博思得科技发展有限公司。保留所有权利。

用途

本 PSKPrn.ocx 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, Windows 7, Windows 8 等操作系统的应用程序提供便利。

本 ocx 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， $1\text{dot} = 25.4\text{mm}/203 \approx 0.125\text{mm}$ ($1\text{dot} = 1000 / 203 \approx 5\text{mil}$);
- 对于 300DPI 的打印机来说， $1\text{dot} = 25.4\text{mm}/300 \approx 0.085\text{mm}$ ($1\text{dot} = 1000 / 300 \approx 3\text{mil}$);
- 对于 600DPI 的打印机来说， $1\text{dot} = 25.4\text{mm}/600 \approx 0.042\text{mm}$ ($1\text{dot} = 1000 / 600 \approx 2\text{mil}$)。

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

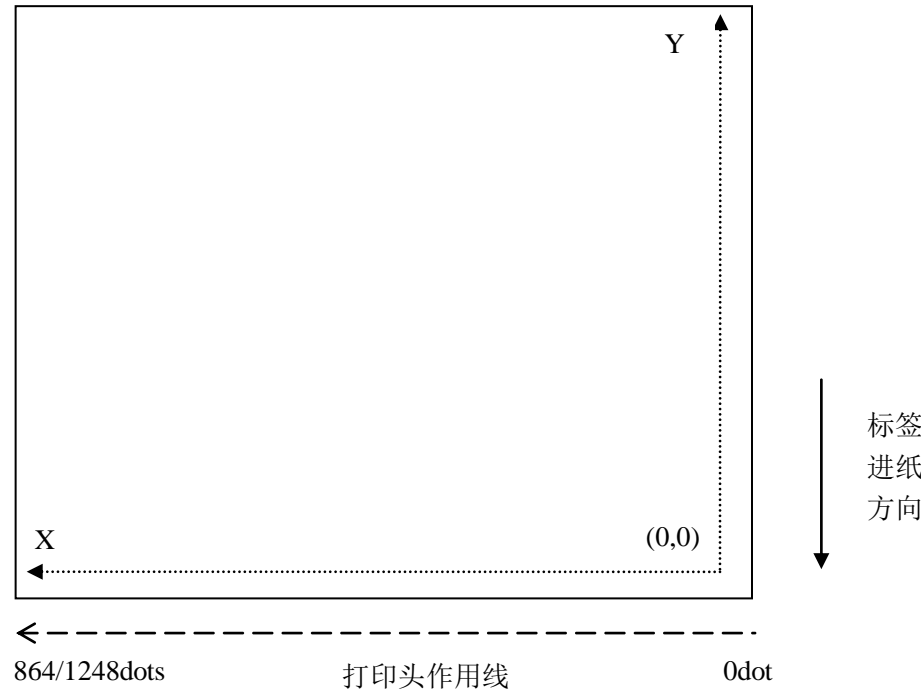
字符串

- * 字符串以双引号 (“) 作为起始和结束标记;
- * 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	打开指定名称的打印机
ClosePort	关闭使用 OpenPort 函数打开的打印机。
PTKGetErrState	检测使用 PSKPRN.OCX 里的其它函数后是否有错误产生；
PTKGetInfo	得到本 API 函数库的版本信息。
PTKClearBuffer	清除打印机缓冲内存的内容。
PTKSetDarkness	设置打印头发热温度。
PTKSetPrintSpeed	设置打印速度。
PTKSetLabelHeight	设置标签的高度和定位间隙\黑线\穿孔的高度。
PTKSetLabelWidth	设置标签的宽度。
PTKSetDirection	设置标签打印方向。
PTKSetCoordinateOrigin	设置/改变坐标原点。
PTKPrintLabel	命令打印机执行打印工作。
PTKDrawText	打印一行文本文字。
PTKDrawTextEx	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。
PTKDrawTextTrueTypeW	打印一行 TrueType Font 文字，并且文字宽度和高度可以微调。
PTKDrawBarcode	打印一个条码。
PTKDrawBarcodeEx	打印一个条码，内容可以是常量、序列号、变量或组合字符串。
PTKDrawBar2DDATAMATRIX	打印一个 DataMatrix 二维条码。
PTKDrawBar2DQR	打印一个 QR 条码(指令方式)。
PTKDrawBar2DQREx	打印一个 QR 条码(图形方式)。
PTKDrawBar2DMaxiCode	打印一个 MaxiCode 条码。
PTKDrawBar2DPdf417	打印一个 PDF417 二维条码。
PTKDrawBar2DHANXIN	打印一个汉信码二维条码。
PTKPcxGraphicsList	打印已存储在打印机 RAM 或 FLASH 存储器里的图形名称清单。
PTKPcxGraphicsDel	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。
PTKPcxGraphicsDownload	存储一个 PCX 格式的图形到打印机。
PTKDrawPcxGraphics	打印指定的图形。
PTKPrintPCX	打印一个 PCX 格式的图形。
PTKBmpGraphicsDownload	转换 BMP 到 PCX 格式，然后将 PCX 格式的图形到打印机。
PTKBinGraphicsList	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)。
PTKBinGraphicsDel	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)。

PTKBinGraphicsDownload	存储一个 Bin 格式的图形到打印机。
PTKRecallBinGraphics	打印一个已保存在打印机里的 Bin 格式图形。
PTKDrawBinGraphics	打印二进制格式的图形。
PTKDrawRectangle	画距形。
PTKDrawLineXor	画直线(两直线相交处作”异或”处理)。
PTKDrawLineOr	画直线(两直线相交处作”或”处理)。
PTKDrawDiagonal	画斜线。
PTKDrawWhiteLine	画白色直线。
PTKSoftFontList	打印存储在 RAM 或 FLASH 存储器里的软字体的名称清单。
PTKSoftFontDel	删除存储在 RAM 或 FLASH 存储器里的一个或所有的软字体。
PTKFormList	打印存储在打印机里的表格名称清单。
PTKFormDel	删除存储在打印机里的一个或所有的表格。
PTKFormDownload	存储一个表格到打印机；此命令与 PTKFormEnd 函数配对使用。
PTKFormEnd	结束存储表格(Form)，此函数与 PTKFormDownload 配对使用。
PTKExecForm	运行指定的表格。
PTKDefineCounter	定义一个序列号变量。
PTKDefineVariable	定义变量。
PTKDownload	下载变量或序列号变量。
PTKDownloadInitVar	初始化变量或序列号变量。
PTKPrintLabelAuto	自动执行打印工作。
PTKSendFile	发送指令文件到打印机
PTK_GetUSBID	获取打印机 USBID 号
PTKDisableBackFeed	取消打印回转功能。
PTKEnableBackFeed	设置打印回转功能。
PTKPrintConfiguration	打印机器当前的设置/工作状态。
PTKSetPrinterState	设置打印机的工作状态。
PTKDisableErrorReport	取消错误反馈。
PTKEnableErrorReport	设置错误反馈。
PTKEnableFLASH	选择 FLASH 存储器。
PTKDisableFLASH	取消选择 FLASH 存储器。
PTKFeedMedia	命令打印机走一行标签。
PTKMediaDetect	校准纸张探测器
PTKCutPage	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)。
PTKCutPageEx	实时调整切刀切纸张数
PTKFeedBack	要求打印机立刻反馈错误报告。
PTKReset	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_RWRFLDLabel	读写 RFID 标签。
PTKSetRFLabelPWAndLockRFLabel	设置 RFID 标签密码和锁定 RFID 标签。
* PTKSetRFIDLabelRetryCount	设置每个 RFID 标签重试次数。 *该函数无效
PTK_SetRFID	RFID 设置指令。
PTKSetFontGap	调整打印文字字间距
* PTKSetBarCodeFontName	设置可打印条码文字的下载字体 *该函数无效
PTKRenameDownloadFont	将下载到打印机的字体进行重命名字体 ID
PTKSetCharSets	设置字符集

函数详细说明

OpenPort

说明:

OpenPort 函数的作用是打开指定名称的打印机。

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:

```
LONG OpenPort(BSTR szPrinterName);
```

参数:

szPrinterName: 当前所使用的打印机在WINDOWS下的名称（打印机驱动名称）;

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:

```
OpenPort("POSTEK G-2108"); //表示打开 POSTEK G-2108 打印机。
```

ClosePort

说明:

ClosePort函数的作用是关闭使用**OpenPort**函数打开的打印机。

用户在对打印机操作完成之后，调用ClosePort关闭打印机；

原型:

```
LONG ClosePort(void);
```

参数: 无

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:

```
ClosePort();
```

PTKGetErrState

说明:

PTKGetErrState 函数的作用是检测使用 PSKPRN.OCX 里的其它函数后是否有错误产生;
错误代码请参阅“PSKPrn.ocx 错误返回值解析” **这个函数必须在 ClosePort() 函数前使用!**

原型:

```
LONG PTKGetErrState(void);
```

参数: 无

返回值:

0 -> OK;

其它返回值请参考章节: **PSKPrn.ocx 错误返回值解析**。

示范:

```
int state = 0;
OpenPort( "POSTEK C168/200s" );
...
state = PTKGetErrState();
...
ClosePort();
```

PTKGetInfo

说明:

PTKGetInfo 函数作用是得到本 API 函数库的版本信息。

原型:

```
LONG PTKGetInfo(void);
```

参数: 无

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:

```
OpenPort( "POSTEK G-3106" );
PTKClearBuffer();
PTKGetInfo();
PTKPrintLabel(1,1);
ClosePort();
```


PTKClearBuffer

说明:

PTKClearBuffer 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, 务必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:

LONG PTKClearBuffer(void);

参数: 无

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:

```
PTKClearBuffer ( );
```

PTKSetDarkness

说明:

PTKSetDarkness 函数的作用是设置打印头发热温度。

原型:

LONG PTKSetDarkness(LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKSetDarkness (10);
```

PTKSetPrintSpeed

说明:

PTKSetPrintSpeed 函数的作用是设置打印速度。

不同型号打印机最大打印速度不同（具体请查看您使用机型用户手册），当设置值（**px**）超过打印机最大打印速度将以打印机最大值为准，设置无效。

原型:

LONG PTKSetPrintSpeed(LONG px);

参数:

px: 取值范围为0 - 10, 或者10 - 100。

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)	7
80	8.0 ips (200 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：**PSKPrn.ocx** 错误返回值解析。

范例: PTKSetPrintSpeed (5);

PTKSetLabelHeight

说明:

PTKSetLabelHeight 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:

```
LONG PTKSetLabelHeight(LONG lheight, LONG gapH, LONG gapOffset, VARIANT_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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKSetLabelHeight (160, 24,0,FALSE);
```

PTKSetLabelWidth

说明:

PTKSetLabelWidth 函数的作用是设置标签的宽度。

原型:

```
LONG PTKSetLabelWidth(LONG 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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKSetLabelWidth (250);
```

PTKSetDirection

说明:

PTKSetDirection 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、矩形。

原型:

```
LONG PTKSetDirection(SHORT direct);
```

参数:

direct: 方向，取值为 B 或 T，缺省值为 T。

B: (对应 ASCII 值 66) 将从标签右下角开始打印;

T: (对应 ASCII 值 84) 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：**PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKSetDirection ( 'B' );
```

PTKSetCoordinateOrigin

说明:

PTKSetCoordinateOrigin 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:

```
LONG PTKSetCoordinateOrigin(LONG px, LONG 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：**PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKSetCoordinateOrigin (12, 23);
```

PTKPrintLabel

说明:

PTKPrintLabel 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 **PTKPrintLabelAuto（）函数代替。**

原型:

LONG PTKPrintLabel(LONG number, LONG 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: **PSKPrn.ocx 错误返回值解析**。

范例:

PTKPrintLabel (2, 3);

PTKDrawText

说明:

PTKDrawText 函数作用是打印一行文本文字。

原型:

LONG PTKDrawText(LONG px, LONG py,
LONG pdirec, LONG pFont,
LONG pHorizontal, LONG pVertical,
SHORT ptext, BSTR 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.
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。

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:
PTKDrawText (50, 30, 0, 2, 1, 1, 'N', "123456789");

PTKDrawTextEx

说明:
PTKDrawTextEx 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:

```
LONG PTKDrawTextEx( LONG px, LONG py, LONG pdirec,
                    LONG pFont, LONG pHorizontal,
                    LONG pVertical, SHORT ptext,
                    BSTR pstr, VARIANT_BOOL Variable);
```

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
‘a’	24点阵中文宋体
‘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"**。

Variable: 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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKDrawTextEx (50, 30, 0, 2, 1, 1, 'N', "\"123456789\"C0\", true);
PTKDrawTextEx (50, 30, 0, 2, 1, 1, 'N', "\"123456789\", false);
PTKDrawTextEx (50, 30, 0, 2, 1, 1, 'N', "\"C1\", true);
PTKDrawTextEx (50, 30, 0, 2, 1, 1, 'N', "\"V3\", true);
PTKDrawTextEx (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)

```
PTKFormDel("LSFORM"); //删除 Form "TEST"
PTKFormDownload("LSFORM"); //存储 Form "TEST"
PTKDefineCounter(0, 10, 78, "+1", "InputC0"); //定义一个序列号变量 C0
PTKDefineCounter(1, 10, 78, "+1", "InputC1"); //定义一个序列号变量 C1
PTKDefineVariable(0, 16, 78, "InputV0"); //定义变量字符串 V0, 命名为: 'V'+第一个参数
PTKDrawBarcodeEx(100, 100, 0, "3", 2, 6, 30, 66, "C0", true); //打印序列号条码,
PTKDrawTextEx(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)
```

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

```
PTKExecForm("LSFORM"); // 运行指定的表格:LSFORM
PTKDownload(); // 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**)

```
PTKDownloadInitVar("12345678"); // 初始化系列号变量 C0
PTKDownloadInitVar("123456"); // 初始化系列号变量 C1
PTKDownloadInitVar("1111"); // 初始化变量 V0
```

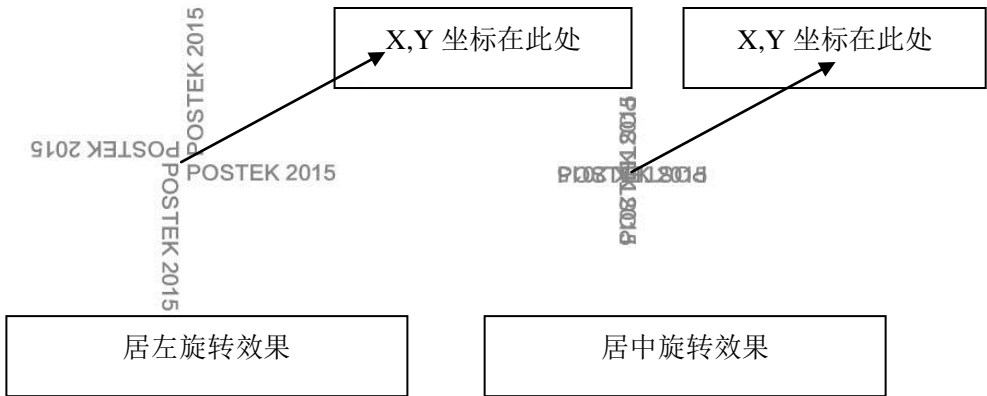
```
PTKPrintLabel(2, 1); // 命令打印机执行打印工作
```


PTKDrawTextTrueTypeW

说明:
PTKDrawTextTrueTypeW 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:
LONG PTKDrawTextTrueTypeW(LONG x, LONG y, LONG FHeight, LONG FWidth,
BSTR FType, LONG Fspin, LONG FWeight, LONG FItalic,
LONG FUnline, LONG FStrikeOut, BSTR id_name, BSTR 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: 字体旋转角度及文字相对坐标对齐方式;
 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;
Funderline: 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: PSKPrn.ocx 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)。

```
PTKPcxGraphicsDel("A1")或者 PTKPcxGraphicsDel("*");//建议打印 TrueType 字体前调用  
PTKDrawTextTrueTypeW (30, 35, 24, 0, "宋体", 4, 400, 0, 0, 0, "A1", "机要绝密");
```

PTKDrawBarcode

说明:
PTKDrawBarcode 函数作用是打印一个条码。

原型:
LONG PTKDrawBarcode(LONG px, LONG py, LONG pdirec, BSTR pCode,
LONG NarrowWidth, LONG pHorizontal,
LONG pVertical, SHORT ptext, BSTR 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
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' 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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKDrawBarcode (50, 30, 0,"1A", 1, 1, 10, 'N', "123456");
```

PTKDrawBarcodeEx

说明:

PTKDrawBarcodeEx 函数作用是打印一个条码。

原型:

```
LONG PTKDrawBarcodeEx (LONG px, LONG py, LONG pdirec,
                        BSTR pCode, LONG NarrowWidth, LONG pHorizontal,
                        LONG pVertical, SHORT ptext, BSTR pstr, VARIANT_BOOL Variable);
```

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
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' 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。

Vn: 变量字符串, 此变量字符串必须已经定义。

如: "data1"CnVn"data2".

Variable: 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);

PTKDrawBar2DDATAMATRIX

说明:

PTKDrawBar2DDATAMATRIX 函数作用是打印一个 **DataMatrix** 二维条码。

原型:

```
LONG PTKDrawBar2DDATAMATRIX( LONG x, LONG y,  
                                LONG w, LONG v,  
                                LONG o, LONG m, BSTR pstr);
```

*参数:

x:	●X 座标。
y:	●Y 座标。备注: 1 dot = 0.125 mm。
w:	●最大列印宽度, 单位 dots。
v:	●最大列印高度, 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:	●资料字符串。

传回值: 0 -> OK.

其它返回值请参考章节: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKDrawBar2DDATAMATRIX(50, 30, 0, 0, 0, 5, "123456789");
```

PTKDrawBar2DQR

说明:

PTKDrawBar2DQR 函数作用是打印一个 QR 条码(指令方式)。

原型:

```
LONG PTKDrawBar2DQR( LONG x, LONG y,  
                      LONG w, LONG v,  
                      LONG o, LONG r,  
                      LONG m, LONG g,  
                      LONG s, BSTR pstr);
```

*参数:

x:	●X 座标。
y:	●Y 座标。备注: 1 dot = 0.125 mm。
w:	●最大列印宽度, 单位 dots。

- v: ●最大列印高度, 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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKDrawBar2D_QR(50, 30, 180, 180, 0, 0, 1, 0, 8, "123456789");
```

PTKDrawBar2DQREx

说明:

PTKDrawBar2DQREx 函数作用是打印一个 QR 条码(图形方式)。

原型:

```
LONG PTKDrawBar2DQREx( LONG x, LONG y,
```

```

LONG o, LONG r,
LONG g, LONG v,
LONG s, BSTR id_name, BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKDrawBar2DQREx(50, 30, 0, 5, 1, 0, 8, "QR1", "123456789");
```

PTKDrawBar2DMaxiCode

说明:

PTKDrawBar2DMaxiCode 函数作用是打印一个 MaxiCode 条码。

原型:

```
LONG PTKDrawBar2DMaxiCode( LONG x, LONG y,  
                           LONG m, LONG u, BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKDrawBar2DMaxiCode(50, 30, 4, 0, "1Z000A7&dajc_iaj-3=+~#^$5");
```

PTKDrawBar2DPdf417

说明:

PTKDrawBar2DPdf417 函数作用是打印一个 **PDF417** 二维条码。

原型:

```
LONG PTKDrawBar2DPdf417( LONG x, LONG y, LONG w, LONG v,  
                          LONG s, LONG c, LONG px, LONG py,  
                          LONG r, LONG l, LONG t, LONG o, BSTR pstr);
```

参数:

x:	X 座标。
y:	Y 座标。备注: 1 dot = 0.125 mm。
w:	最大列印宽度, 单位 dots。
v:	最大列印高度, 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:	列印方向定位, '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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKDrawBar2DPdf417 (10, 10, 400, 300, 0, 0, 3, 7, 10, 2, 0, 0, "POSTEKINFO");
```

PTKDrawBar2DHANXIN

说明:

PTKDrawBar2DHANXIN 函数作用是打印一个汉信码二维条码。

```
LONG PTKDrawBar2DHANXIN( LONG x, LONG y, LONG w,
                          LONG v, LONG o, LONG r,
                          LONG m, LONG g, LONG s, BSTR 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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKDrawBar2DHANXIN(50, 30, 0, 0, 0, 5, 1, 3, 2, "POSTEK");
```

PTKPcxGraphicsList

说明:

PTKPcxGraphicsList 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:

```
LONG PTKPcxGraphicsList(void);
```

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:

```
PTKPcxGraphicsList ();
```

PTKPcxGraphicsDel

说明:

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

原型:

```
LONG PTKPcxGraphicsDel(BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKPcxGraphicsDel (“PCX”);
```

PTKPcxGraphicsDownload

说明:

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

原型:

```
LONG PTKPcxGraphicsDownload(BSTR pcxname, BSTR 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 **PTKDrawPcxGraphics** () 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKPcxGraphicsDownload ("PCXA", "c:\\test1111.pcx");
```

PTKDrawPcxGraphics

说明:

PTKDrawPcxGraphics 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 PTKPcxGraphicsDownload () 存储到打印机里。

原型:

```
LONG PTKDrawPcxGraphics (LONG px, LONG py, BSTR 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 **PTKPcxGraphicsDownload** () 中自定义的图形名称。

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:

```
PTKDrawPcxGraphics (100, 50, "PCX");
```

PTKPrintPCX

说明:

PTKPrintPCX 函数是打印一个 **PCX** 格式的图形。

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

原型:

```
LONG PTKPrintPCX(LONG px, LONG py, BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

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

PTKBmpGraphicsDownload

说明:

PTKBmpGraphicsDownload 函数的作用是先转换 **BMP** 到 **PCX** 格式, 然手将 **PCX** 格式的图形到打印机。

原型:

```
LONG PTKBmpGraphicsDownload(BSTR pcxname, BSTR pcxpath, LONG 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 **PTKDrawPcxGraphics** () 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKBmpGraphicsDownload("PCXA", "c:\\test1111.bmp", 0);
```

PTKBinGraphicsList

说明:

PTKBinGraphicsList 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 **PTKPcxGraphicsList** 方法相同

原型:

LONG PTKBinGraphicsList(void);

参数:

无

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:

```
PTKBinGraphicsList ( );
```

PTKBinGraphicsDel

说明:

PTKBinGraphicsDel 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 **PTKPcxGraphicsDel** 相同。

原型:

LONG PTKBinGraphicsDel(BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKBinGraphics Del (“Bin2”);
```

PTKBinGraphicsDownload

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

原型：
LONG PTKBinGraphicsDownload(BSTR name, LONG pbyte, LONG pH, BYTE* Gdata);

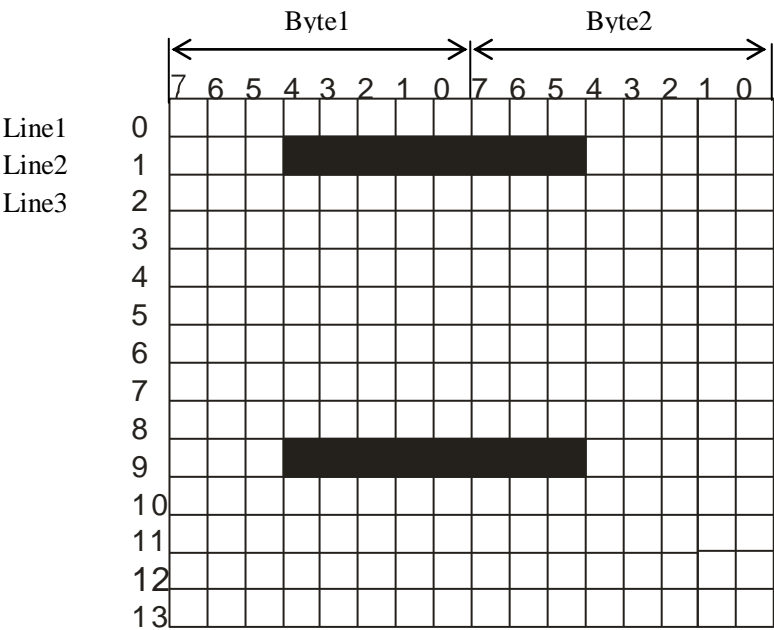
参数：

- 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 PTKRecallBinGraphics（）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： PSKPrn.ocx 错误返回值解析。

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

PTKRecallBinGraphics

说明:

PTKRecallBinGraphics 函数是打印一个 **Bin** 格式的图形。

原型:

```
LONG PTKRecallBinGraphics(LONG px, LONG py, BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

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

PTKDrawBinGraphics

说明:

PTKDrawBinGraphics 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:

```
LONG PTKDrawBinGraphics(LONG px, LONG py, LONG pbyte, LONG pH, BSTR 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;

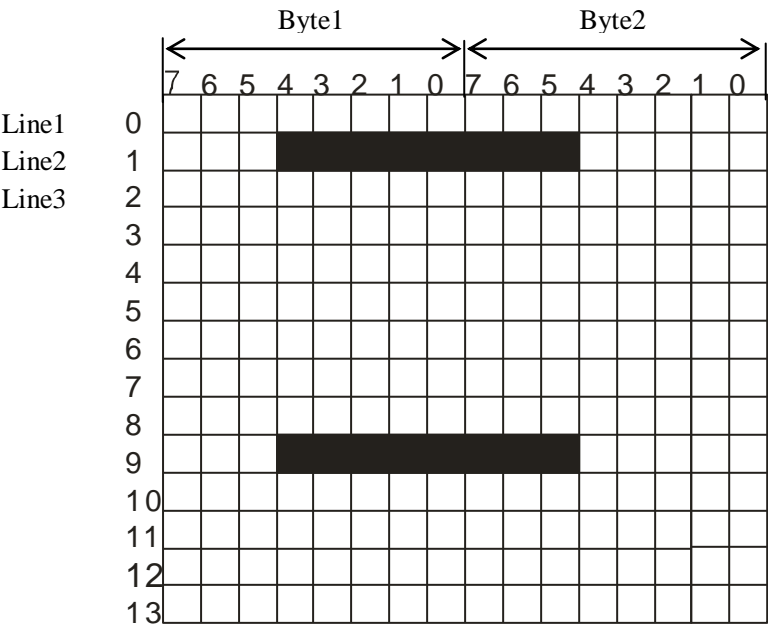
Ph: 图形的高度, 以点(dots)为单位;

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: **PSKPrn.ocx 错误返回值解析**。

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

PTKDrawRectangle

说明:

PTKDrawRectangle 函数的作用是画矩形。

原型:

LONG PTKDrawRectangle(LONG px, LONG py, LONG thickness, LONG pEx, LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

PTKDrawRectangle (50, 120, 5, 250, 150);

PTKDrawLineXor

说明:

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

原型:

LONG PTKDrawLineXor(LONG px, LONG py, LONG pbyte, LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

PTKDrawLineXor (100, 20, 5, 110);

PTKDrawLineOr

说明:

PTKDrawLineOr 函数作用是画直线(两直线相交处作”或”处理)。

原型:

LONG PTKDrawLineOr(LONG px, LONG py, LONG plength, LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

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

PTKDrawDiagonal

说明:

PTKDrawDiagonal 函数的作用是画斜线。

原型:

LONG PTKDrawDiagonal(LONG px, LONG py, LONG thickness, LONG pEx, LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

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

PTKDrawWhiteLine

说明:

PTKDrawWhiteLine函数的作用是画白色直线。

原型:

LONG PTKDrawWhiteLine(LONG px, LONG py, LONG plength, LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

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

PTKSoftFontList

说明:

PTKSoftFontList 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:

LONG PTKSoftFontList(void);

参数: 无

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:

```
PTKSoftFontList ();
```

PTKSoftFontDel

说明:

PTKSoftFontDel 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:

```
LONG PTKSoftFontDel (SHORT 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKSoftFontDel ('A');
```

PTKFormList

说明:

PTKFormList 函数作用是打印存储在打印机里的表格名称清单。

原型:

```
LONG PTKFormList (void);
```

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:

```
PTKFormList ();
```

PTKFormDel

说明:

PTKFormDel 函数是删除存储在打印机里的一个或所有的表格。

原型:

LONG PTKFormDel(BSTR 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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKFormDel (“FORMNAME”);
```

PTKFormDownload

说明:

PTKFormDownload 函数的作用是存储一个表格到打印机; 此命令与 **PTKFormEnd** 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:

LONG PTKFormDownload(BSTR 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: **PSKPrn.ocx 错误返回值解析**。

范例:

```
PTKFormDownload (“FORMNAME”);
```

PTKFormEnd

说明:

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

原型:

```
LONG PTKFormEnd(void);
```

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:

```
PTKFormDownload ("Form1");  
...  
PTKFormEnd ();
```

PTKExecForm

说明:

PTKExecForm 函数的作用是运行指定的表格。

原型:

```
LONG PTKExecForm(BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKExecForm ("FORM1");
```


PTKDefineCounter

说明:

PTKDefineCounter 函数作用是定义一个序列号变量。

原型:

```
LONG PTKDefineCounter(LONG id, LONG maxNum, SHORT ptext, BSTR pstr, BSTR pMsg);
```

参数:

id: 序列号 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, 0 - 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, ...;

“-40”=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, 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: **PSKPrn.ocx 错误返回值解析**。

范例:

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

PTKDefineVariable

说明:

PTKDefineVariable 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:

```
LONG PTKDefineVariable(LONG pid, LONG pmax, SHORT porder, BSTR 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKDefineVariable (0, 16, L, "\nEnter Title:\' \'');
```

PTKDownload

说明:

PTKDownload 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:

```
LONG PTKDownload(void);
```

参数: 无

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:

```
PTKDownload( );
```

PTKDownloadInitVar

说明:

PTKDownloadInitVar 函数的作用是初始化变量或序列号变量。

需跟在 PTKDownload() 函数后使用。

原型:

```
LONG PTKDownloadInitVar(BSTR pstr);
```

参数: 无

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:

```
PTKDownloadInitVar("123456");
```

PTKPrintLabelAuto

说明:

PTKPrintLabelAuto 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:

```
LONG PTKPrintLabelAuto(LONG number, LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKPrintLabelAuto (2, 3);
```

PTKSendFile

说明:

PTKSendFile 函数的作用是发送文件到打印机。

原型:

```
LONG PTKSendFile(BSTR szFilePath);
```

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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKSendFile("cmdfile.txt");
```

PTKGetUSBID

说明:

PTKGetUSBID 函数的作用是获取打印机 USBID 号。

原型:

```
BSTR PTKGetUSBID(LONG nUSBID);
```

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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKGetUSBID ();
```

PTKDisableBackFeed

说明:

PTKDisableBackFeed 函数作用是取消打印回转功能。

原型:

```
LONG PTKDisableBackFeed(void);
```

参数: 无

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:

```
PTKDisableBackFeed ( );
```

PTKEnableBackFeed

说明:

PTKEnableBackFeed 函数作用是设置打印回转功能。

原型:

```
LONG PTKEnableBackFeed(LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKEnableBackFeed (140);
```

PTKPrintConfiguration

说明:

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

原型:

```
LONG PTKPrintConfiguration(void);
```

参数: 无

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:

```
PTKPrintConfiguration ();
```

PTKSetPrinterState

说明:

PTKSetPrinterState函数的作用是设置打印机的工作状态。

原型:

```
LONG PTKSetPrinterState(SHORT 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKSetPrinterState ('D');
```

PTKDisableErrorReport

说明：
PTKDisableErrorReport 函数的作用是取消错误反馈。

原型：
LONG PTKDisableErrorReport(void);

参数：无

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：
PTKDisableErrorReport();

PTKEnableErrorReport

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 PTKEnableErrorReport (void);

参数: 无

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:

PTKEnableErrorReport ();

PTKEnableFLASH

说明:

PTKEnableFLASH 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:

LONG PTKEnableFLASH(void);

参数: 无

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:

PTKEnableFLASH ();

PTKDisableFLASH

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:

```
LONG PTKDisableFLASH(void);
```

参数: 无

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:

```
PTK DisableFLASH ( );
```

PTKFeedMedia

说明:

PTK FeedMedia 函数的作用是命令打印机走一行标签。

原型:

```
LONG PTKFeedMedia(void);
```

参数: 无

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:

```
PTK FeedMedia ( );
```

PTKMediaDetect

说明:

PTK MediaDetect 函数的作用校准纸张探测器。

原型:

```
LONG PTKMediaDetect(void);
```

参数: 无

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:

```
PTK MediaDetect ();
```

PTKCutPage

说明:

PTKCutPage 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:

```
LONG PTKCutPage(LONG 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: **PSKPrn.ocx** 错误返回值解析。

范例:

```
PTKCutPage(1);
```

PTKCutPageEx

说明：
PTKCutPageEx 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：
LONG PTKCutPageEx (LONG 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： PSKPrn.ocx 错误返回值解析。

范例：
PTKCutPage(1);

PTKFeedBack

说明：
PTKFeedBack 函数的作用是要求打印机立刻反馈错误报告。
此函数不返回错误代码,如需接收错误代码请参考 PTKErrorReport 方法。
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：
LONG PTKFeedBack (void);

参数：无

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:

```
PTKFeedBack ( );
```

PTKReset

说明:

PTKReset 函数的作用是将打印机复位。
这个命令对打印机复位，会将打印机设置恢复到出厂状态
不能在打印函数序列的开头或者中间使用此指令，否则会将该指令后面的内容清空，导致打印机不执行该函数后面的打印函数。

原型:

```
LONG PTKReset(void);
```

参数: 无

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:

```
PTKReset( );
```

PTKErrorReport

说明:

PTKErrorReport 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。。

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>
```

表<一>

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; (此参数主要为编程扩展预留，默认 0 即可，无作用。)
- 1:

表示打开 COM1;
- 2:

表示打开 COM2;
- 3:

表示打开 COM3。
- rPort: 接收打印机当前错误状态代码的端口; (此参数可以支持 COM1 到 COM255)

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, 请参考表<一>中的 Error/Status code 值
其它返回值请参考章节： PSKPrn.ocx 错误返回值解析。

范例:

```
1. 通过设置 C168 200DPI 打印机驱动中的打印端口，从 COM1 读取当前错误状态代码：
char buff[5] = {0};
PTK_ErrorReportEx (1, 1, 38400, FALSE, 20);

OpenPort ("POSTEK C168/200s ");
PTK_ClearBuffer ();
.....

.....
ClosePort ();
```

PTKErrorReportUSB

说明:

PTKErrorReportUSB 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。必须在 PTK_ClearBuffer()函数调用之后使用。

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>

表<一>

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:

LONG PTK_ErrorReportUSB();

返回值:

>=0, 请参考表<一>中的 Error/Status code 值
其它返回值请参考章节： PSKPrn.ocx 错误返回值解析。

范例:

```
1. 通过 USB 端口读取当前错误状态代码:
OpenUSBPort (255);
....
PTK_ClearBuffer ();
int nerrorStatusCode;
nerrorStatusCode = PTK_ErrorReportUSB() ;

if (nerrorStatusCode ==0)
{
.....
    打印
.....
}
ClosePort();
```

PTKRWRFIDLabel

说明:

PTKRWRFIDLabel 函数的作用读写 RFID 标签。

原型:

```
LONG PTKRWRFIDLabel(LONG nRWMode, LONG nWForm,  
                     LONG nStartBlock, LONG nWDataNum,  
                     LONG nWArea, BSTR pstr);
```

参数:

nRWMode: RFID 操作方式. 0—读 RFID; 1—写 RFID;
nWForm: RFID 写入格式. 0—HEX (十六进制); 1—ASCII;
nStartBlock: 写入起始块.
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)

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:

例 1:

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

读取 EPC 区 (Start=2, size=3word)

313233343536

例 2:

```
PTKRWRFIDLabel(1, 1, 0, 6, 3, "POSTEK");
```

输出结果:

读取 USER 区 (Start=0, size=3word)

504F5354454B

PTKSetRFLabelPWAndLockRFLabel

说明:

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

原型:

LONG PTKSetRFLabelPWAndLockRFLabel (LONG nOperationMode, LONG OperationnArea, BSTR 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: PSKPrn.ocx 错误返回值解析。

范例:

例 1:

PTKSetRFLabelPWAndLockRFLabel(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:

```
PTKSetRFLabelPWAndLockRFLabel(4, 0, “5462EF21”);
```

输出结果:

读取销毁密码区

5462EF21

PTKSetRFIDLabelRetryCount

说明:

PTKSetRFIDLabelRetryCount 函数的作用设置每个 RFID 标签重试次数。

原型:

```
LONG TKSetRFIDLabelRetryCount (LONG nRetryCount);
```

参数:

nRetryCount: 重试次数 0~9

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:

例 1:

```
PTKSetRFIDLabelRetryCount(7); 重试 7 次
```

PTKSetRFID

说明:

PTKSetRFID 函数的作用是 RFID 设置指令。

原型:

```
LONG PTKSetRFID(LONG nReservationParameters, LONG nReadWriteLocation,  
                LONG ReadWriteArea, LONG nMaxErrNum, LONG 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: PSKPrn.ocx 错误返回值解析。

范例:

例 1:

```
PTKSetRFID(0, 0, 0, 3, 0);
```

PTKSetFontGap

说明:

PTKSetFontGap 函数的作用是调整打印文字字间距。

原型:

```
LONG PTKSetFontGap(LONG 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：PSKPrn.ocx 错误返回值解析。

范例：

例 1：

```
int return = PTKSetFontGap (10);
```

PTKSetBarCodeFontName

说明：

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

原型：

```
LONG PTKSetBarCodeFontName (SHORT Name, LONG FontW, LONG 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：PSKPrn.ocx 错误返回值解析。

范例:

例 1:

```
PTKSetBarCodeFontName ( 'A' , 32, 32);
```

PTKRenameDownloadFont

说明:

PTKRenameDownloadFont 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:

```
LONG PTKRenameDownloadFont (LONG StoreType, SHORT Fontname, BSTR 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: PSKPrn.ocx 错误返回值解析。

范例:

```
PTKRenameDownloadFont (1, 'A', "arial");
```

PTKSetCharSets

说明:

PTKSetCharSets 函数的作用是设置字符集

原型:

```
LONG PTKSetCharSets (LONG BitValue, SHORT CharSets, BSTR 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： PSKPrn.ocx 错误返回值解析。

范例:

```
PTKSetCharSets (8, 'N', "001");
```

PSKPrn.ocx 错误返回值解析

- 1000 to -1003 : OpenPort 操作出错;
- 1010 to -1023 : 串口读取操作出错 (PTKErrorReport);
- 1026 to -1028 : 设置串口错误 (PTKErrorReport);
- 1030 to -1035 : 串口写操作出错 (PTKErrorReport);
- 1040 to -1049 : 写打印机出错;
- 1060 : 打开 USB 端口出错;

- 2000 : PTKGetInfo 执行出错;
- 2001 : PTKClearBuffer 创建文件失败;
- 2002 : PTKClearBuffer 执行出错;
- 2003 : PTKSetDarkness 执行出错;
- 2004 : PTKSetPrintSpeed 执行出错;
- 2005 : PTKSetPrintSpeed 参数错误;
- 2006 : PTKSetLabelHeight 执行出错;
- 2007 : PTKSetLabelWidth 执行出错;
- 2008 : PTKSetDirection 执行出错;
- 2009 : PTKSetDirection 参数错误;
- 2010 : PTKSetCoordinateOrigin 执行出错;
- 2011 : PTKPrintLabel 执行出错;
- 2012 : PTKPrintLabel 参数错误;
- 2013 : PTKPrintLabelAuto 执行出错;
- 2014 : PTKPrintLabelAuto 参数错误;
- 2015 : PTKDrawText 执行出错;
- 2016 : PTKDrawText 参数出错;
- 2017 : PTKDrawTextEx 执行出错;
- 2018 : PTKDrawTextEx 参数出错;
- 2019 : PTKDrawTextTrueTypeW 创建 PrinterDC 失败, 进行出错处理;;
- 2020 : PTKDrawTextTrueTypeW 分配保存 bitmap 内存出错;
- 2021 : 分配保存当前程序运行的文件路径内存出错;
- 2022 : 分配保存 PCX HEAD 文件结构内存出错;
- 2023 : 创建 PCX 文件出错;
- 2024 : 分配保存 PCX data 内存出错;
- 2025 : 保存 PCX data 出错;
- 2026 : PTKDrawBarcode 执行出错;
- 2027 : PTKDrawBarcode 参数错误;
- 2028 : PTKDrawBarcodeEx 执行出错;
- 2029 : PTKDrawBarcodeEx 参数错误;
- 2030 : PTKDrawBar2D_DATAMATRIX 分配内存失败;
- 2031 : PTKDrawBar2D_DATAMATRIX 执行出错;
- 2032 : PTKDrawBar2D_DATAMATRIX 执行出错;
- 2033 : PTKDrawBar2D_QR 执行出错;
- 2034 : PTKDrawBar2D_QR 执行出错;
- 2035 : PTKDrawBar2D_QR 执行出错;

- 2036 : PTKDrawBar2D_QREx 分配内存失败;
- 2037 : PTKDrawBar2D_QREx 执行出错;
- 2038 : PTKDrawBar2D_MaxiCode 分配内存失败;
- 2039 : PTKDrawBar2D_MaxiCode 执行出错;
- 2040 : PTKDrawBar2D_MaxiCode 执行出错;
- 2041 : PTKDrawBar2D_Pdf417 分配内存失败;
- 2042 : PTKDrawBar2D_Pdf417 执行出错;
- 2043 : PTKDrawBar2D_Pdf417 执行出错;
- 2044 : PTKDrawBar2D_HANXIN 分配内存失败;
- 2045 : PTKDrawBar2D_HANXIN 执行出错;
- 2046 : PTKDrawBar2D_HANXIN 执行出错;
- 2047 : PTKPcxGraphicsList 执行出错;
- 2048 : PTKPcxGraphicsDel 分配内存出错;
- 2049 : PTKPcxGraphicsDel 参数错误;
- 2050 : PTKPcxGraphicsDel 执行出错;
- 2051 : PTKPcxGraphicsDownload 分配内存出错;
- 2052 : PTKPcxGraphicsDownload 分配内存出错;
- 2053 : PTKPcxGraphicsDownload 打开文件错误;
- 2054 : PTKPcxGraphicsDownload 参数错误;
- 2055 : PTKPcxGraphicsDownload 执行发送文件信息出错;
- 2056 : PTKPcxGraphicsDownload 执行发送文件内容出错;
- 2057 : PTKDrawPcxGraphics 分配内存出错;
- 2058 : PTKDrawPcxGraphics 参数错误;
- 2059 : PTKDrawPcxGraphics 执行出错;
- 2060 : PTKPrintPCX 执行出错;
- 2061 --2062: PTKBmpGraphicsDownload 分配内存出错;
- 2063 : PTKBmpGraphicsDownload 下载 BMP 位深度错误;
- 2064 : PTKBmpGraphicsDownload 打开文件错误;
- 2065 : PTKBmpGraphicsDownload 参数错误;
- 2066 : PTKBmpGraphicsDownload 执行发送文件信息出错;
- 2067 : PTKBmpGraphicsDownload 执行发送文件内容出错;
- 2068 : PTKBinGraphicsList 执行出错;
- 2069 : PTKBinGraphicsDel 分配内存出错;
- 2070 : PTKBinGraphicsDel 参数错误;
- 2071 : PTKBinGraphicsDel 执行出错;
- 2072 : PTKBinGraphicsDownload 执行发送二进制格式信息出错;
- 2073 : PTKBinGraphicsDownload 执行发送二进制图形内容出错;
- 2074 : PTKRecallBinGraphics 分配内存出错;
- 2075 : PTKRecallBinGraphics 执行出错;
- 2076 : PTKRecallBinGraphics 参数错误;
- 2077 : PTKDrawBinGraphics 发送二进制格式信息出错;
- 2078 : PTKDrawBinGraphics 发送二进制图形内容出错;
- 2079 : PTKDrawRectangle 执行出错;
- 2080 : PTKDrawLineXor 执行出错;

- 2081 : PTKDrawLineOr 执行出错;
- 2082 : PTKDrawDiagonal 执行出错;
- 2083 : PTKDrawWhiteLine 执行出错;
- 2084 : PTKSoftFontList 执行出错;
- 2085 : PTKSoftFontDel 参数错误;
- 2086 : PTKSoftFontDel 执行出错;
- 2087 : PTKFormList 执行出错;
- 2088 : PTKFormDel 分配内存出错;
- 2089 : PTKFormDel 参数错误;
- 2090 : PTKFormDel 执行出错;
- 2091 : PTKFormDownload 分配内存出错;
- 2092 : PTKFormDownload 参数错误;
- 2093 : PTKFormDownload 执行出错;
- 2094 : PTKFormEnd 执行出错;
- 2095 : PTKExecForm 分配内存出错;
- 2096 : PTKExecForm 参数错误;
- 2097 : PTKExecForm 执行出错;
- 2098 : PTKDefineCounter 分配内存出错;
- 2099 : PTKDefineCounter 分配内存出错;
- 2100 : PTKDefineCounter 执行出错;
- 2101 : PTKDefineCounter 参数错误;
- 2102 : PTKDefineVariable 执行出错;
- 2103 : PTKDefineVariable 提示内容参数错误;
- 2104 : PTKDefineVariable 其他参数错误, 请查看函数说明;
- 2105 : PTKDownload 执行出错;
- 2106 : PTKDownloadInitVar 分配内存出错;
- 2107 : PTKDownloadInitVar 执行出错;
- 2108 : PTKSendFile 分配内存失败;
- 2109 : PTKSendFile 打开文件失败;
- 2110 : PTKSendFile 执行写数据出错;
- 2111 : PTKGetUSBID 执行出错;
- 2112 : PTKDisableBackFeed 执行出错;
- 2113 : PTKEnableBackFeed 执行出错;
- 2114 : PTKPrintConfiguration 执行出错;
- 2115 : PTKSetPrinterState 执行出错;
- 2116 : PTKSetPrinterState 参数错误;
- 2117 : PTKDisableErrorReport 执行出错;
- 2118 : PTKEnableErrorReport 执行出错;
- 2119 : PTKEnableFLASH 执行出错;
- 2120 : PTKDisableFLASH 执行出错;
- 2121 : PTKFeedMedia 执行出错;
- 2122 : PTKMediaDetect 执行出错;
- 2123 : PTKCutPage 执行出错;
- 2124 : PTKCutPageEx 执行出错;

- 2125 : PTKReset 执行出错;
- 2126 : PTKFeedBack 执行出错;
- 2127 : PTKErrorReport 打开写端口失败;
- 2128 : PTKErrorReport 获取反馈超时;
- 2129 : PTKErrorReportUSB 打开 USB 端口失败;
- 2130 : PTKErrorReportUSB 获取打印机反馈失败;
- 2131 : PTKRWRFIDLabel 分配内存失败;
- 2132 : PTKRWRFIDLabel 执行出错;
- 2133 : PTKSetRFLabelPWAndLockRFLabel 分配内存失败;
- 2134 : PTKSetRFLabelPWAndLockRFLabel 执行出错;
- 2135 : PTKSetRFIDLabelRetryCount 执行出错;
- 2136 : PTKSetRFID 执行出错;
- 2137 : PTKSetFontGap 执行出错;
- 2138 : PTKSetBarCodeFontName 执行出错;
- 2139 : PTKSetCharSets 执行出错;
- 2140 : PTKRenameDownloadFont 执行出错;
- 2141 to -2142 : PTKErrorReport 写入反馈命令失败;
- 2143 : PTKBmpGraphicsDownload 打开文件失败;
- 2144 : PTKBinGraphicsDownload 分配内存出错;

- 3000 to -3070 : 打印机未打开或已经关闭;
- 3071 : PTK_SendFile 读文件数据出错;