

[ZPL Windows SDK]

[打印机 ZPL 指令集开发帮助文档 v2.0]

1.手册信息.....	4
2.运行平台.....	4
3.备注.....	4
4.方法.....	5
4.1 PrinterCreator.....	5
4.2 PrinterCreatorS.....	6
4.3 PrinterDestroy.....	7
4.4 PortOpen.....	8
4.5 PortClose.....	10
4.6 WriteData.....	11
4.7 ReadData.....	12
4.8 DirectIO.....	13
4.9 ZPL_StartFormat.....	15
4.10 ZPL_EndFormat.....	16
4.11 ZPL_ScalableFontText.....	17
4.12 ZPL_Text.....	19
4.13 ZPL_BarCode39.....	22
4.14 ZPL_Pdf417.....	24
4.15 ZPL_CodeEan8.....	26
4.16 ZPL_UpceCode.....	28
4.17 ZPL_BarCode93.....	30
4.18 ZPL_BarCode128.....	32
4.19 ZPL_CodeEan13.....	34
4.20 ZPL_MicroPdf417.....	36
4.21 ZPL_QRCode.....	38
4.22 ZPL_UpcExtensions.....	40
4.23 ZPL_UpcaBarcode.....	42
4.24 ZPL_SetChangeFontEncoding.....	44
4.25 ZPL_SetChangeCaret.....	46
4.26 ZPL_SetChangeDelimiter.....	47
4.27 ZPL_SetChangeTilde.....	48
4.28 ZPL_GraphicBox.....	49
4.29 ZPL_GraphicCircle.....	51
4.30 ZPL_GraphicDiagonalLine.....	52
4.31 ZPL_GraphicEllipse.....	54
4.32 ZPL_PrintImage.....	56
4.33 ZPL_GraphicSymbol.....	57
4.34 ZPL_SetDiagnosticsMode.....	59
4.35 ZPL_SetLabelHome.....	60
4.36 ZPL_SetLabelLength.....	61
4.37 ZPL_SetLabelShift.....	62
4.38 ZPL_SetLabelTop.....	63
4.39 ZPL_SetPrintMode.....	64
4.40 ZPL_SetMediaType.....	66
4.41 ZPL_SetPrintingMirrorImage.....	67
4.42 ZPL_SetPrintOrientation.....	68
4.43 ZPL_SetPrintRate.....	69
4.44 ZPL_SetPrintWidth.....	70
4.45 ZPL_SetSerialCommunications.....	71
4.46 ZPL_SetPrintDarkness.....	73
4.47 ZPL_SetTearOffAdjustPosition.....	74
4.48 ZPL_PrintConfigurationLabel.....	75

4.49 ZPL_GetPrinterIpAddress.....	76
4.50 ZPL_GetPrinterStatus.....	77
4.51 ZPL_GetLabelLength.....	79
4.52 ZPL_GetLabelWidth.....	80
4.53 ZPL_GetPrinterSeriesNumber.....	81
4.54 ZPL_GetPrinterMacAddress.....	82
4.55 ZPL_GetPrinterName.....	83
4.56 ZPL_GetPrinterFirmwareVersion.....	84
4.57 ZPL_GetPrinterDpi.....	85
4.58 ZPL_LearnLabel.....	86
4.59 ZPL_SetReprintAfterError.....	87
4.60 ZPL_SetNetworkSetting.....	88
4.61 ZPL_SetMediaTracking.....	90
4.62 ZPL_SetUserFontName.....	91
4.63 ZPL_SetVietMode.....	92
4.64 ZPL_SetVietFontEncoding.....	93
4.65 ZPL_Text_Block.....	94
4.66 ZPL_RfidWrite.....	97
4.67 ZPL_RfidRead.....	99
4.68 ZPL_RfidCalibration.....	101
4.69 ZPL_SetPrintQuantity.....	102
4.70 ZPL_DataMatrixBarcode.....	104
4.71 ZPL_GetPrinterName.....	106
4.72 ZPL_GetPrinterSeriesNumber.....	107
4.73 ZPL_GetPrinterOdometer.....	108
4.74 ZPL_GetPrinterFonts.....	109
4.75 ZPL_SetPrinterInstruction.....	110
4.76 ZPL_SetPrinterNetMode.....	111
4.76 ZPL_SetPrinterNetSSID.....	112
4.77 ZPL_SetPrinterNetPwdSwitch.....	113
4.78 ZPL_SetPrinterNetPwd.....	114
4.79 ZPL_SetPrinterNetDHCP.....	115
4.80 ZPL_SetPrintIpAddress.....	116
4.81 ZPL_SetPrintSubnetMask.....	117
4.82 ZPL_SetPrintDefaultGateway.....	118
4.83 ZPL_SetPrinterBluetoothSSID.....	119
4.84 ZPL_SetPrinterBluetoothPIN.....	120
4.85 ZPL_SetPrinterSleepTime.....	121
4.86 ZPL_SetPrinterShutdownTime.....	122
4.87 ZPL_FirmwareUpgrade.....	123
4.88 ZPL_FontDownload.....	125
4.89 ZPL_VectorFontDownload.....	127

1.手册信息

本 SDK 手册提供了 Windows 应用程序开发所需的*.dll 文件信息。

我们在不断地努力提高和升级我们所有产品的功能与质量。

之后，产品规格和用户手册的内容可能会更改，将不再另行通知。

2.运行平台

- Windows 2003/XP/7/8/10

3.备注

- 错误代码返回值大于0时，属于 Windows 系统内部错误，请查阅相关帮助文档。

4.方法

4.1 PrinterCreator

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

```
int* PrinterCreator(  
  
    void* handle,  
  
    const TCHAR* model  
  
);
```

参数：

```
void* handle  
    [in,out] 创建目标打印机对象。  
const TCHAR* model  
    [in] 指定目标打印机型号。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_INVALID_MODEL	-8	机型名称无效

4.2 PrinterCreatorS

此函数功能与 PrinterCreator 相同，即创建指定机型的目标打印机（使用任何打印机操作之前必须先创建打印机对象）。

```
void* PrinterCreatorS(  
  
    const char* model  
  
);
```

参数：

const char model*
[in] 指定目标打印机型号。

返回值：

成功:返回打印机对象的句柄。

失败:返回 NULL, 无效句柄。

4.3 PrinterDestroy

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

int PrinterDestroy(

void* handle

);

参数：

void handle*

[in] 需要释放的目标打印机对象。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_BAD_HANDLE	-6	句柄无效

4.4 PortOpen

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

```
int PortOpen(  
    void* handle,  
    const TCHAR* ioSettings  
);
```

参数：

- `void* handle`
[in,out] 创建的目标打印机对象。
- `const TCHAR* ioSettings`
[in] 设置连接目标打印机的通讯端口参数。具体内容查看下表：

配置列表：

类别	配置	描述	示例
USB	USB [,Position/Model/PortNum]	USB：连接任一本公司 USB 打印机。 USB[,Position] ：当同时连接本公司多台打印机时，可以通过 USB 位置信息(Position 参数)来指定连接某一特定 USB 端口的打印机。	USB USB,Port_#0004.Hub_#0003 USB,LPG4 USB,USB001
NET	NET , IP Addr (IPV4)[,Port]	指定网络打印机的 IP 地址和端口。如果不指定端口，默认端口是9100。	NET,192.168.0.36 NET,192.168.0.36,9100
COM	COMn ,BAUDRATE_rate	指定连接的串口端口号和波特率。	COM5,BAUDRATE_19200
LPT	LPTn	指定连接的并口端口号。	LPT1

注：[]表示可选参数。

如何查看 USB 打印机的位置信息（Position 参数）：

在“Windows 设备管理器”中，展开“通用串行总线控制器”，选中指定机型的“USB 打印支持”设备，右键菜单中选择“属性”，点击“详细信息”。

属性“总线关系”包含机型名称与虚拟 USB 端口号。

* 如果同时连接本公司多台不同型号的打印机，建议采用“USB, 机型”的方式连接。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_OPEN_FAILED	-311	端口打开失败

4.5 PortClose

此函数功能为关闭通讯端口。

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

参数:

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

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_BAD_HANDLE	-6	句柄无效

4.6 WriteData

此函数功能为向打印机发送数据。

```
int WriteData(  
  
    void* handle,  
  
    unsigned char* writeData,  
  
    unsigned int writeNum  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- unsigned char* writeData*
[in] 发给打印机的数据，数据是十六进制字符串。
- unsigned int writeNum*
[in] 发送数据的长度。

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	通信端口未打开
E_IO_WRITE_FAILED	-321	写入失败
E_IO_WRITE_TIMEOUT	-322	写入超时
E_IO_READ_FAILED	-331	读取失败
E_IO_READ_TIMEOUT	-332	读取超时

4.7 ReadData

此函数功能为读取打印机的数据。

```
int ReadData(  
  
    void* handle,  
  
    unsigned char* readData,  
  
    unsigned int readNum,  
  
    unsigned int* preadedNum  
  
);
```

参数:

void handle*

[in,out] 创建的目标打印机对象。

unsigned char readData*

[in] 需要读取的打印机数据。

unsigned int readNum

[in] 需要读取的数据长度。

unsigned int preadedNum*

[in] 实际读取到的数据长度。

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	通信端口未打开
E_IO_READ_FAILED	-331	读取失败
E_IO_READ_TIMEOUT	-332	读取超时

4.8 DirectIO

此函数功能为用户自定义发送和读取打印机的数据。

当某些功能未提供函数接口时，用户可以通过此接口向打印机发送指令数据。

```
int DirectIO(  
    void* handle,  
    unsigned char* writedata,  
    unsigned int writeNum,  
    unsigned char* readdata,  
    unsigned int readNum,  
    unsigned int* preadNum  
);
```

参数：

void* *handle*

[in,out] 创建的目标打印机对象。

unsigned char* *writedata*

[in] 写入打印机的数据。

unsigned int *writeNum*

[in] 写入打印机的数据长度。当 *writeNum*=0时，不实行写入数据操作。

unsigned char* *readdata*

[in,out] 获取打印机返回的数据。

unsigned int *readNum*

[in] 预设需要读取的数据长度。当 *readNum*=0时，不实行读取数据操作。

unsigned int* *preadNum*

[in,out] 实际读取的数据长度。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时
E_IO_READ_FAILED	-331	读取数据失败
E_IO_READ_TIMEOUT	-332	读取数据超时

4.9 ZPL_StartFormat

此函数功能为表示一个新的标签格式的开始。

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

参数:

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

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.10 ZPL_EndFormat

此函数功能为表示一个标签格式的结束。

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

参数：

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

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.11 ZPL_ScalableFontText

此函数功能为打印可缩放字体。

```
int ZPL_ScalableFontText(  
  
    void* handle,  
  
    int xPos,  
  
    int yPos,  
  
    char fontName,  
  
    int orientation,  
  
    int fontWidth,  
  
    int fontHeight,  
  
    char* text  
  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

char fontName

[in] 字体（取值：A-Z 和 0-9）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int *fontWidth*

[in] 字体宽度。

int *fontHeight*

[in] 字体高度。

*char** *text*

[in] 文本数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.12 ZPL_Text

此函数功能为打印文本。

```
int ZPL_Text(  
    void* handle,  
    int xPos,  
    int yPos,  
    int fontNum,  
    int orientation,  
    int fontWidth,  
    int fontHeight,  
    char* text  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int fontNum

[in] 字体。

- 0 : FONT 0 - 可缩放字体
- 1 : FONT A - 位图字体
- 2 : FONT B - 位图字体
- 3 : FONT D - 位图字体
- 4 : FONT E - 位图字体
- 5 : FONT F - 位图字体
- 6 : FONT G - 位图字体
- 7 : FONT H - 位图字体
- 8 : FONT GS - 位图字体
- 9 : FONT P - 位图字体
- 10 : FONT Q - 位图字体
- 11 : FONT R - 位图字体
- 12 : FONT S - 位图字体
- 13 : FONT T - 位图字体
- 14 : FONT U - 位图字体
- 15 : FONT V - 位图字体
- 16 : SIMSUN.TTF - 宋体
- 17 : FONT Z - 越南字体

int orientation

[in] 打印方向。

- 0 : 正常
- 90 : 顺时针旋转90度
- 180 : 顺时针旋转180度
- 270 : 顺时针旋转270度

int fontWidth

[in] 字体宽度。

int fontHeight

[in] 字体高度。

备注：当选择 FONT Z 时，宽高最小值为12*24，且只能倍增

char text*

[in] 文本数据。

FONT A -- ABCDxyz 12345
FONT B -- ABCDXYZ 12345 UPPER CASE ONLY
FONT D -- ABCDwxyz 12345
FONT E -- (OCR-B)ABCDwxyz 12345
FONT F -- ABCDwxyz 12345
FONT G -- **AByz 12**
FONT H -- (OCR-A) UPPER CASE ONLY
FONT O -- (Scaleable) ABCDwxyz 12345
FONT GS -- ® © ™ ®
FONT P -- ABCDwxyz 12345
FONT Q -- ABCDwxyz 12345
FONT R -- ABCDwxyz 12345
FONT S -- ABCDwxyz 12345
FONT T -- ABCDwxyz 12345
FONT U -- **ABCDwxyz 12345**
FONT V -- **ABCDwxyz 12345**

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.13 ZPL_BarCode39

此函数功能为打印 Barcode39条码。

int ZPL_BarCode39(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *moduleWidth*,

int *codeHeight*,

char *line*,

char *lineAboveCode*,

char *digit*,

char* *text*

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

‘N’：不打印

‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

‘N’：不打印在条码上方

‘Y’：打印在条码上方

char digit

[in] 校验位。

‘N’：不打印校验位

‘Y’：打印校验位

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.14 ZPL_Pdf417

此函数功能为打印 PDF417 二维码。

```
int ZPL_Pdf417(  
    void* handle,  
    int xPos,  
    int yPos,  
    int orientation,  
    int moduleWidth,  
    int codeHeight,  
    int securityLevel,  
    int column,  
    int rows,  
    char truncate,  
    char* text  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

- 0 : 正常
- 90 : 顺时针旋转90度
- 180 : 顺时针旋转180度
- 270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

int securityLevel

[in] 安全级别（范围：1-8）。

int column

[in] 要编码的列数。

int rows

[in] 要编码的行数。

char truncate

[in] 截断层指示和停止模式。

‘N’:不截断

‘Y’:执行截断

char text*

[in] 二维码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.15 ZPL_CodeEan8

此函数功能为打印 CodeEan8条码。

```
int ZPL_CodeEan8(  
    void* handle,  
    int xPos,  
    int yPos,  
    int orientation,  
    int moduleWidth,  
    int codeHeight,  
    char line,  
    char lineAboveCode,  
    char* text  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

‘N’：不打印

‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

‘N’：不打印在条码上方

‘Y’：打印在条码上方

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.16 ZPL_UpceCode

此函数功能为打印 UPC-E 条码。

int ZPL_UpceCode(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *moduleWidth*,

int *codeHeight*,

char *line*,

char *lineAboveCode*,

char* *text*

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

‘N’：不打印

‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

‘N’：不打印在条码上方

‘Y’：打印在条码上方

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.17 ZPL_BarCode93

此函数功能为打印 Barcode93条码。

int ZPL_BarCode93(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *moduleWidth*,

int *codeHeight*,

char *line*,

char *lineAboveCode*,

char *digit*,

char* *text*

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

‘N’：不打印

‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

‘N’：不打印在条码上方

‘Y’：打印在条码上方

char digit

[in] 校验位。

‘N’：不打印校验位

‘Y’：打印校验位

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.18 ZPL_BarCode128

此函数功能为打印 Barcode128条码。

int ZPL_BarCode128(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *moduleWidth*,

int *codeHeight*,

char *line*,

char *lineAboveCode*,

char *checkDigit*,

char *mode*,

char* *text*

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

- 0 : 正常
- 90 : 顺时针旋转90度
- 180 : 顺时针旋转180度
- 270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

- ‘N’：不打印
- ‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

- ‘N’：不打印在条码上方
- ‘Y’：打印在条码上方

char checkDigit

[in] UCC 校验位。

- ‘N’：不打印校验位
- ‘Y’：打印校验位

char mode

[in] 模式。

- ‘N’：不选择模式
- ‘U’：UCC 匹配模式
- ‘A’：自动模式
- ‘D’：UCC/EAN 模式

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.19 ZPL_CodeEan13

此函数功能为打印 CodeEan13条码。

int ZPL_CodeEan13(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *moduleWidth*,

int *codeHeight*,

char *line*,

char *lineAboveCode*,

char* *text*

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

‘N’：不打印

‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

‘N’：不打印在条码上方

‘Y’：打印在条码上方

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.20 ZPL_MicroPdf417

此函数功能为打印 MicroPdf417码。

int ZPL_MicroPdf417(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *moduleWidth*,

int *codeHeight*,

int *mode*,

char* *text*

);

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

int mode

[in] 模式（范围：0-33）。

Mode (M)	Number of Data Columns	Number of Data Rows	% of Cws for EC	Max Alpha Characters	Max Digits
0	1	11	64	6	8
1	1	14	50	12	17
2	1	17	41	18	26
3	1	20	40	22	32
4	1	24	33	30	44
5	1	28	29	38	55
6	2	8	50	14	20
7	2	11	41	24	35
8	2	14	32	36	52
9	2	17	29	46	67
10	2	20	28	56	82
11	2	23	28	64	93
12	2	26	29	72	105
13	3	6	67	10	14
14	3	8	58	18	26
15	3	10	53	26	38
16	3	12	50	34	49
17	3	15	47	46	67
18	3	20	43	66	96
19	3	26	41	90	132
20	3	32	40	114	167
21	3	38	39	138	202
22	3	44	38	162	237
23	4	6	50	22	32
24	4	8	44	34	49
25	4	10	40	46	67
26	4	12	38	58	85
27	4	15	35	76	111
28	4	20	33	106	155
29	4	26	31	142	208
30	4	32	30	178	261
31	4	38	29	214	313
32	4	44	28	250	366
33	4	4	50	14	20

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败

E_IO_WRITE_TIMEOUT	-322	写入数据超时
--------------------	------	--------

4.21 ZPL_QRCode

此函数功能为打印二维码。

```
int ZPL_QRCode(  
  
    void* handle,  
  
    int xPos,  
  
    int yPos,  
  
    int orientation,  
  
    int model,  
  
    int dpi,  
  
    char eccLevel,  
  
    char input,  
  
    char charMode,  
  
    char* text  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int xPos  
    [in] 水平起始位置（范围：0-32000，单位：dot）。  
int yPos  
    [in] 垂直起始位置（范围：0-32000，单位：dot）。  
int orientation  
    [in] 打印方向。  
        0 ： 正常  
        90 ： 顺时针旋转90度  
        180 ： 顺时针旋转180度
```

270 : 顺时针旋转270度

int model

[in] 设置二维码版本（1:原始版,2:强化版）。

int dpi

[in] 放大系数（范围：1-10）。

char eccLevel

[in] 纠错级别。

H:超高可靠性

Q:高可靠性

M:标准水平

L:高密度水平

char input

[in] 输入模式。

A:自动输入

M:手动输入

char charMode

[in] 数据类型。

N:数字

A:字母数字

B:8位字节模式

K:Kanji — 仅将 Kanji 字符按照基于 JIS X 0208的 Shift JIS 系统处理。这意味着字符模式 K 之后的所有参数都应16位字符。如果出现8位字符（如 ASCII 代码），则会发生错误。

char text*

[in] 二维码数据。仅当 charMode 是 B 时，数据最前面四位应为数据大小，例如数据为 qrcode 时传0006qrcode。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.22 ZPL_UpcExtensions

此函数功能为打印 UPC 扩展条码。

```
int ZPL_UpcExtensions(  
    void* handle,  
    int xPos,  
    int yPos,  
    int orientation,  
    int moduleWidth,  
    int codeHeight,  
    char line,  
    char lineAboveCode,  
    char* text  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

‘N’：不打印

‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

‘N’：不打印在条码上方

‘Y’：打印在条码上方

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.23 ZPL_UpcaBarcode

此函数功能为打印 UPC-A 条码。

int ZPL_UpcaBarcode(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *moduleWidth*,

int *codeHeight*,

char *line*,

char *lineAboveCode*,

char *digit*,

char* *text*

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int moduleWidth

[in] 条码宽度（范围：0-10，单位：dot）。

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

char line

[in] 注释行。

‘N’：不打印

‘Y’：打印

char lineAboveCode

[in] 条码上方的注释行。

‘N’：不打印在条码上方

‘Y’：打印在条码上方

char digit

[in] 校验位

‘N’：不打印校验位

‘Y’：打印校验位

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.24 ZPL_SetChangeFontEncoding

此函数功能为选择国际字符集。

```
int ZPL_SetChangeFontEncoding(  
  
    void* handle,  
  
    int encodeType  
  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int encodeType

[in] 字符集类型（范围：0-31, 33-36）。

- 0 : 单字节编码 - 美国1字符集
- 1 : 单字节编码 - 美国2字符集
- 2 : 单字节编码 - 英国字符集
- 3 : 单字节编码 - 荷兰字符集
- 4 : 单字节编码 - 丹麦/挪威字符集
- 5 : 单字节编码 - 瑞典/芬兰字符集
- 6 : 单字节编码 - 德国字符集
- 7 : 单字节编码 - 法国1字符集
- 8 : 单字节编码 - 法国2字符集
- 9 : 单字节编码 - 意大利字符集
- 10 : 单字节编码 - 西班牙字符集
- 11 : 单字节编码 - 杂项字符集
- 12 : 单字节编码 - 日本字符组
- 13 : 代码页850
- 14 : 双字节亚洲编码
- 15 : Shift-JIS
- 16 : EUC-JP 和 EUC-CN
- 17 : 不推荐使用 - UCS-2 Big Endian
- 18-23 : 保留
- 24 : 单字节亚洲编码
- 25 : 保留
- 26 : 多字节亚洲编码

- 27 : 代码页1252
- 28 : Unicode (UTF-8编码) - Unicode 字符集
- 29 : Unicode (UTF-16 Big-Endian 编码) - Unicode 字符集
- 30 : Unicode (UTF-16 Little-Endian 编码) - Unicode 字符集
- 31 : 代码页1250
- 32 : 越南字符集
- 33 : 代码页1251
- 34 : 代码页1253
- 35 : 代码页1254
- 36 : 代码页1255

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.25 ZPL_SetChangeCaret

此函数功能为更改格式命令前缀。

```
int ZPL_SetChangeCaret(  
  
    void* handle,  
  
    char character  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char character  
    [in] 格式命令前缀。
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.26 ZPL_SetChangeDelimiter

此函数功能为更改分隔符。

```
int ZPL_SetChangeDelimiter(  
  
    void* handle,  
  
    char character  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char character  
    [in] 分隔符。
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.27 ZPL_SetChangeTilde

此函数功能为更改控制命令前缀。

```
int ZPL_SetChangeTilde(  
  
    void* handle,  
  
    char charactor  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char charactor  
    [in] 控制命令前缀。
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.28 ZPL_GraphicBox

此函数功能为绘制图形框。

int ZPL_GraphicBox(

void* *handle*,

int *xPos*,

int *yPos*,

int *width*,

int *height*,

int *thickness*,

int *rounding*,

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int width

[in] 框的宽度（范围：1-32000，单位：dot）。

int height

[in] 框的高度（范围：1-32000，单位：dot）。

int thickness

[in] 边界厚度（范围：1-32000，单位：dot）。

int rounding

[in] 转角程度（范围：0-8）。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.29 ZPL_GraphicCircle

此函数功能为绘制图形圆圈。

```
int ZPL_GraphicCircle(  
  
    void* handle,  
  
    int xPos,  
  
    int yPos,  
  
    int diameter,  
  
    int thickness,  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int xPos  
    [in] 水平起始位置（范围：0-32000，单位：dot）。  
int yPos  
    [in] 垂直起始位置（范围：0-32000，单位：dot）。  
int diameter  
    [in] 圆的直径（范围：3-4095，单位：dot）。  
int thickness  
    [in] 边界厚度（范围：1-4095，单位：dot）。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.30 ZPL_GraphicDiagonalLine

此函数功能为绘制对角线。

int ZPL_GraphicDiagonalLine(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *width*,

int *height*,

int *thickness*

);

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 对角线的方向。

R（或/）：右倾斜的对角线

L（或\）：左倾斜的对角线

int width

[in] 框的宽度（范围：1-32000，单位：dot）。

int height

[in] 框的高度（范围：1-32000，单位：dot）。

int thickness

[in] 边界厚度（范围：1-32000，单位：dot）。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.31 ZPL_GraphicEllipse

此函数功能为绘制图形椭圆。

int ZPL_GraphicEllipse(

void* *handle*,

int *xPos*,

int *yPos*,

int *width*,

int *height*,

int *thickness*

);

参数：

void* handle

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int width

[in] 椭圆宽度（范围：3-4095，单位：dot）。

int height

[in] 椭圆高度（范围：3-4095，单位：dot）。

int thickness

[in] 边界厚度（范围：2-4095，单位：dot）。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.32 ZPL_PrintImage

此函数功能为打印图片。

```
int ZPL_PrintImage(  
  
    void* handle,  
  
    int xPos,  
  
    int yPos,  
  
    char* imgName  
  
);
```

参数：

- void* handle*
[in,out] 创建的目标打印机对象。
- int xPos*
[in] 水平起始位置（范围：0-32000，单位：dot）。
- int yPos*
[in] 垂直起始位置（范围：0-32000，单位：dot）。
- char* imgName*
[in] 图片的路径。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.33 ZPL_GraphicSymbol

此函数功能为生成注册商标，版权符号和其他符号。

int ZPL_GraphicSymbol(

void* *handle*,

int *xPos*,

int *yPos*,

int *orientation*,

int *width*,

int *height*,

char* *type*

);

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

0 : 正常

90 : 顺时针旋转90度

180 : 顺时针旋转180度

270 : 顺时针旋转270度

int width

[in] 符号宽度。

int height

[in] 符号高度。

char type*
[in] 数据字符串。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.34 ZPL_SetDiagnosticsMode

此函数功能为启动诊断模式。

```
int ZPL_SetDiagnosticsMode(  
  
    void* handle,  
  
    int isEnabled  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- int isEnabled*
[in] 是否开启诊断模式。
1: 开启诊断模式
0: 取消诊断模式

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.35 ZPL_SetLabelHome

此函数功能为设置标签首页位置。

```
int ZPL_SetLabelHome(  
  
    void* handle  
  
    int xPos,  
  
    int yPos  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- int xPos*
[in] 横坐标起始位置（范围：0-32000，单位：dot）。
- int yPos*
[in] 纵坐标起始位置（范围：0-32000，单位：dot）。

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.36 ZPL_SetLabelLength

此函数功能为设置标签长度。

```
int ZPL_SetLabelLength(  
  
    void* handle,  
  
    int length  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int length  
    [in]  标签长度（范围：1-32000，单位：dot）。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.37 ZPL_SetLabelShift

此函数功能为将标签内容向左移动。

```
int ZPL_SetLabelShift(  
  
    void* handle,  
  
    int shift  
  
);
```

参数：

- void* handle*
[in,out] 创建的目标打印机对象。
- int shift*
[in] 向左移动的值（范围：-9999–9999，单位：dot）。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.38 ZPL_SetLabelTop

此函数功能为相对于标签的上边缘，将标签的位置向上或向下短距离移动。

```
int ZPL_SetLabelTop(  
  
    void* handle,  
  
    int top  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- int top*
[in] 最大化程度（范围：-120~120，单位：dot）。

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.39 ZPL_SetPrintMode

此函数功能为设置打印标签或标签组之后打印机执行的操作。

int ZPL_SetPrintMode(

void* *handle*,

char *mode*,

char *prePeelSelect*

);

参数:

void* *handle*

[in,out] 创建的目标打印机对象。

char *mode*

[in] 操作模式。

‘T’ : 撕开

‘P’ : 剥离（取决于打印机型号）

‘R’ : 倒带（取决于打印机型号）

‘A’ : 涂抹器（取决于打印机型号）

‘C’ : 切刀（取决于打印机型号）

‘D’ : 切刀延迟

‘F’ : RFID

‘L’ : 保留

‘U’ : 保留

‘K’ : Kiosk

char *prePeelSelect*

[in] 选择。

‘N’ : 不执行

‘Y’ : 执行

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.40 ZPL_SetMediaType

此函数功能为选择在打印机中使用的媒介类型。

```
int ZPL_SetMediaType(  
  
    void* handle,  
  
    char type  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char type  
    [in] 媒介类型。  
    'T':碳带  
    'D':热敏
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.41 ZPL_SetPrintingMirrorImage

此函数功能为将标签的整个可打印区域打印为镜像图片。

```
int ZPL_SetPrintingMirrorImage(  
  
    void* handle,  
  
    char enable  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- char enable*
[in] 是否开启。
‘N’:不开启
‘Y’:开启

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.42 ZPL_SetPrintOrientation

此函数功能为将标签格式180度翻转打印。

```
int ZPL_SetPrintOrientation(  
  
    void* handle,  
  
    int orientation  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- Int orientation*
[in] 是否翻转。
0: 不翻转
180: 执行翻转

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.43 ZPL_SetPrintRate

此函数功能为设置打印速度。

```
int ZPL_SetPrintRate(  
  
    void* handle,  
  
    int printSpeed,  
  
    int slewSpeed,  
  
    int backfeedSpeed  
  
);
```

参数：

- void* handle*
[in,out] 创建的目标打印机对象。
- int printSpeed*
[in] 打印速度。单位为 inches/sec
- int slewSpeed*
[in] 回转速度。单位为 inches/sec
- int backfeedSpeed*
[in] 反馈速度。单位为 inches/sec

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.44 ZPL_SetPrintWidth

此函数功能为设置打印宽度。

```
int ZPL_SetPrintWidth(  
  
    void* handle,  
  
    int width  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int width  
    [in] 设置打印宽度（范围：2-944，单位：dot）。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.45 ZPL_SetSerialCommunications

此函数功能为更改串行通信参数。

```
int ZPL_SetSerialCommunications(  
  
    void* handle,  
  
    int baudRate,  
  
    int wordLength,  
  
    char parity,  
  
    int stopBits,  
  
    char protocolModo,  
  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int baudRate

[in] 带宽频率。范围如下：

110	300	600	1200	2400
4800	9600	14400	19200	28800
38400	57600	115200		

int wordLength

[in] 字长（范围： 7-8， 单位： data bits）。

char parity

[in] 如下：

‘N’：表示：无。

‘E’：表示：偶校验。

‘O’：表示：奇校验。

int stopBits

[in] 范围： 1-2。

char protocolMode

[in] 如下:

‘X’: 表示: XON/XOFF。

‘D’: 表示: DTR/DSR。

‘R’: 表示: RTS。

‘M’: 表示: DTR/DSR XON/XOFF *r*。

备注: 1、XON/XOFF (继续传输/停止传输) 是一种流量控制协议

2、DTR (数据终端准备好)

3、DSR (数据准备好)

4、RTS (请求发送)

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.46 ZPL_SetPrintDarkness

此函数功能为设置打印的浓度。

```
int ZPL_SetPrintDarkness (  
  
    void* handle,  
  
    int darkness  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int darkness  
    [in] 打印浓度(范围: 0-30, 单位: dot)
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.47 ZPL_SetTearOffAdjustPosition

此函数功能为设置标签撕离的位置。

```
int ZPL_SetTearOffAdjustPosition (  
  
    void* handle,  
  
    int position  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int position  
    [in] 撕离位置(范围: -120~+120)
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.48 ZPL_PrintConfigurationLabel

此函数功能为生成打印机配置标签。

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

参数:

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

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.49 ZPL_GetPrinterIpAddress

此函数功能为获取打印机 IP 地址。

```
int ZPL_GetPrinterIpAddress(  
  
    void* handle  
  
    char* ipAddress  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* ipAddress  
    [in] 打印机的 IP 地址。
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.50 ZPL_GetPrinterStatus

此函数功能为获取打印机当前状态。

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

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- int* status*
[in] 打印机的状态。

HT/HD/XT/XD 系列机型以及 HM-T300 PRO:

状态	值
高温	1
待机	2
打印中	4
标签定位异常	8
纸用尽	16
碳带用尽	32
卡纸	64
跳标	128
标签学习中	256
切刀异常	512
标签格式异常	1024
记忆体写入异常	2048
非法指令	4096
盒盖未到位	8192
碳带将尽	16384

其他机型：

状态	值
待机	0
缺纸	1
开盖	2
暂停	4
过温	8
碳带用尽	16

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.51 ZPL_GetLabelLength

此函数功能为获取标签的长度。

int ZPL_GetLabelLength (

void* *handle*,

char* *length*

);

参数：

void handle*

[in,out] 创建的目标打印机对象。

char length*

[in] 标签的长度。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.52 ZPL_GetLabelWidth

此函数功能为获取标签的宽度。

```
int ZPL_GetLabelWidth(  
  
    void* handle,  
  
    char* width  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* width  
    [in] 标签的宽度。
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.53 ZPL_GetPrinterSeriesNumber

此函数功能为获取打印机序列号。

```
int ZPL_GetPrinterSeriesNumber(  
  
    void* handle,  
  
    char* sn  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* sn  
    [in] 打印机序列号。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.54 ZPL_GetPrinterMacAddress

此函数功能为获取打印机的 MAC 地址。

```
int ZPL_GetPrinterMacAddress(  
  
    void* handle,  
  
    char* macAddress  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* macAddress  
    [in] 打印机 MAC 地址。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.55 ZPL_GetPrinterName

此函数功能为获取打印机的名称。

```
int ZPL_GetPrinterName(  
  
    void* handle,  
  
    char* name  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* name  
    [in] 打印机的名称。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.56 ZPL_GetPrinterFirmwareVersion

此函数功能为获取打印机的固件版本号。

```
int ZPL_GetPrinterFirmwareVersion(  
  
    void* handle,  
  
    char* version  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* version  
    [in] 打印机的固件版本号。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.57 ZPL_GetPrinterDpi

此函数功能为获取打印机的分辨率。

```
int ZPL_GetPrinterDpi(  
  
    void* handle,  
  
    char* dpi  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* dpi  
    [in] 打印机的分辨率。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.58 ZPL_LearnLabel

此函数功能为标签自动学习。

```
int ZPL_LearnLabel(  
  
    void* handle,  
  
);
```

参数：

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

(此函数需要在 ZPL_StartFormat 之前调用)

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.59 ZPL_SetReprintAfterError

此函数功能为重新打印因发生错误而未能打印的标签（错误情况包括 Ribbon Out, Media Out, Head Open 三种）。

int ZPL_SetReprintAfterError(

void* handle,

char *enable

);

参数:

void handle*

[in,out] 创建的目标打印机对象。

Char enable*

[in] 是否开启重印。

“on ”：开启重印

“off ”：不开启重印

（接口需在 ZPL_StartFormat 之前调用）

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.60 ZPL_SetNetworkSetting

此函数功能为更改打印机上的网络设置。

int ZPL_SetNetworkSetting(

void* handle,

char* network

);

参数:

void* handle

[in,out] 创建的目标打印机对象。

char* network

[in] **Format “a,b,c,d,e,f,g,h,i,j”**

a 位置(正在修改的设备):

- 1 表示: 外部有线;
- 2 表示: 内部有线;
- 3 表示: 无线;

b 位置(IP 解析):

- A 表示: All;
- B 表示: BOOTP;
- C 表示: DHCP and BOOTP;
- D 表示: DHCP;

G 表示: 仅捡拾 (如果安装了无线打印服务器或 Wireless Plus 打印服务器, 建议不要使用该值。);

R 表示: RARP;

P 表示: 永久;

c 位置(IP 地址)

:格式为: xxx.xxx.xxx.xxx

d 位置(子网掩码)

:格式为: xxx.xxx.xxx.xxx

e 位置(默认网关)

:格式为: xxx.xxx.xxx.xxx

f 位置(WIN 服务器地址)

:格式为: xxx.xxx.xxx.xxx

g 位置(连接超时检查)

:是否超时检测: Y=yes, N=no

h 位置(超时值)

:范围: 0-9999。

i 位置(ARP 广播间隔)

:范围: 0-30。

j 位置(ARP 广播间隔)

:范围: 1-65535。

参数设置例子: “1, A, 192.168.1.1, 255.255.255.0, 192.168.1.1, 192.168.1.1, Y, 300, 0, 9100”

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.61 ZPL_SetMediaTracking

此函数功能为指定使用的介质类型和黑标偏移量。

```
int ZPL_SetMediaTracking(  
  
    void* handle,  
  
    char mediaType,  
  
    int offset  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- char mediaType*
[in] 介质类型。
‘N’: 连续介质（连续纸）
‘Y’: 非连续介质网眼感应（标签纸）
‘W’: 非连续介质网眼感应（标签纸）
‘M’: 非连续介质标记感应（黑标纸）
‘A’: 在校准过程中自动检测介质类型
‘V’: 连续介质，可变长度（与连续介质相同，但如果打印标签的部分超出定义的
 标签长度，标签尺寸将自动扩展以包含它们）
- int offset*
[in] 黑标偏移量（未使用到，设为0）。

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开

E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.62 ZPL_SetUserFontName

此函数功能为用户自己设置字体，用于文本打印

```
int ZPL_SetPrintDefaultGateway (

    void* handle

    const TCHAR* text

);
```

参数：

void handle*
[in,out] 创建的目标打印机对象.

const TCHAR text*
[in] 字体名称

返回值：

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	This model does not support this feature.
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

4.63 ZPL_SetVietMode

此函数功能为设置越南语模式

```
int ZPL_SetVietMode(  
  
    void* handle  
  
    int vietmode  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int vietmode  
    [in] 模式  
        1: ASCII  
        2: UTF-8
```

备注: 此接口需在 **ZPL_StartFormat** 之前调用

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.64 ZPL_SetVietFontEncoding

此函数功能为设置越南字符集

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

参数:

void handle*
[in,out] 创建的目标打印机对象。

备注: 此接口需在 **ZPL_StartFormat** 之前调用

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.65 ZPL_Text_Block

此函数功能为打印文本块, 可自动换行。

```
int ZPL_Text_Block(  
    void* handle,  
    int xPos,  
    int yPos,  
    int fontNum,  
    int orientation,  
    int fontWidth,  
    int fontHeight,  
    int textBlockWidth,  
    int textBlockHeight,  
    char* text  
);
```

参数:

void* *handle*

[in,out] 创建的目标打印机对象。

int *xPos*

[in] 水平起始位置（范围：0-32000，单位：dot）。

int *yPos*

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int fontNum

[in] 字体。

- 0 : FONT 0 - 可缩放字体
- 1 : FONT A - 位图字体
- 2 : FONT B - 位图字体
- 3 : FONT D - 位图字体
- 4 : FONT E - 位图字体
- 5 : FONT F - 位图字体
- 6 : FONT G - 位图字体
- 7 : FONT H - 位图字体
- 8 : FONT GS - 位图字体
- 9 : FONT P - 位图字体
- 10 : FONT Q - 位图字体
- 11 : FONT R - 位图字体
- 12 : FONT S - 位图字体
- 13 : FONT T - 位图字体
- 14 : FONT U - 位图字体
- 15 : FONT V - 位图字体

int orientation

[in] 打印方向。

- 0 : 正常
- 90 : 顺时针旋转90度
- 180 : 顺时针旋转180度
- 270 : 顺时针旋转270度

int fontWidth

[in] 字体宽度。

int fontHeight

[in] 字体高度。

int textBlockWidth

[in] 文本块宽度。

int textBlockHeight

[in] 文本块高度。

FONT A -- ABCDefgh 12345
FONT B -- ABCDHIJK 12345 UPPER CASE ONLY
FONT D -- ABCDwxyz 12345
FONT E -- **(OCR-B) ABCDwxyz 12345**
FONT F -- ABCDwxyz 12345
FONT G -- **AByz 12**
FONT H -- **(OCR-A) UPPER CASE ONLY**
FONT O -- **(Scaleable) ABCDwxyz 12345**
FONT GS -- ® © ™ ®
FONT P -- ABCDwxyz 12345
FONT Q -- ABCDwxyz 12345
FONT R -- ABCDwxyz 12345
FONT S -- ABCDwxyz 12345
FONT T -- ABCDwxyz 12345
FONT U -- **ABCDwxyz 12345**
FONT V -- **ABCDwxyz 12345**

char text*

[in] 文本数据。

备注：数据暂不支持中文

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.66 ZPL_RfidWrite

此函数功能为写入 RFID 数据

```
int ZPL_RfidWrite(  
  
    void* handle,  
  
    char format,  
  
    int begin,  
  
    int size,  
  
    int memoryBlock,  
  
    const TCHAR* text  
  
    );
```

参数:

void handle*

[in,out] 创建的目标打印机对象。

char format

[in] 格式。

A = ASCII

H = Hexadecimal

E = EPC

int begin

[in] 起始块编号

int size

[in] 要写入的字节数

int memoryBlock

[in] 内存分段

1:EPC

2:TID

const TCHAR text*

[in] 要写入的数据

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.67 ZPL_RfidRead

此函数功能为使 RFID 数据可读, (读取 RFID 数据需要调用本接口并且在 ZPL_EndFormat 后面调用 ReadData 读取)

```
int ZPL_RfidRead(  
  
    void* handle,  
  
    char format,  
  
    int begin,  
  
    int size,  
  
    int memoryBlock,  
  
    const TCHAR* headtext,  
  
    const TCHAR* tailText  
  
);
```

参数:

void handle*

[in,out] 创建的目标打印机对象。

char format

[in] 格式。

A = ASCII

H = Hexadecimal

E = EPC

int begin

[in] 起始块编号

int size

[in] 要读取的字节数

int memoryBlock

[in] 内存分段

1:EPC

2:TID

const TCHAR headtext*

[in] 数据头

const TCHAR tailtext*
[in] 数据尾

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.68 ZPL_RfidCalibration

此函数功能为 RFID 标签校准

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

参数:

void* handle
[in,out] 创建的目标打印机对象。

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.69 ZPL_SetPrintQuantity

此函数功能为控制要打印的标签数量，打印机暂停前打印的标签数量以及每个序列号的复制次数。

```
int ZPL_SetPrintQuantity(  
    void* handle,  
    int totalQuantity,  
    int pauseAndCutValue,  
    int replicatesOfEachSerialNumber,  
    char overridePauseCount  
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int totalQuantity

[in] 要打印的标签总量(范围：大于等于 1)

int pauseAndCutValue

[in] 暂停和切纸值(范围:大于等于 0,0 表示不暂停)

int replicatesOfEachSerialNumber

[in] 每个序列号的副本数(范围：大于等于 0)

char overridePauseCount

[in] 切纸或暂停(N = 暂停,Y = 切纸)

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.70 ZPL_DataMatrixBarcode

此函数功能为打印 Data Matrix 条码。

```
int ZPL_DataMatrixBarcode(
```

```
    void* handle,
```

```
    int xPos,
```

```
    int yPos,
```

```
    int orientation,
```

```
    int codeHeight,
```

```
    int level,
```

```
    int columns,
```

```
    int rows,
```

```
    int formatId,
```

```
    int aspectRatio,
```

```
    char* text
```

```
);
```

参数：

void handle*

[in,out] 创建的目标打印机对象。

int xPos

[in] 水平起始位置（范围：0-32000，单位：dot）。

int yPos

[in] 垂直起始位置（范围：0-32000，单位：dot）。

int orientation

[in] 打印方向。

- 0 : 正常
- 90 : 顺时针旋转90度
- 180 : 顺时针旋转180度
- 270 : 顺时针旋转270度

int codeHeight

[in] 条码高度（范围：1-32000，单位：dot）。

int level

[in] 安全级别（0、50、80、100、140、200）。

int column

[in] 要编码的列数。

int rows

[in] 要编码的行数。

Int formatId

[in] 格式 id（0-6）。

- 1 = 字段数据为数字 空格（ 、 ） 无
- 2 = 字段数据为大写字母数字 空格（ 、 ） 无
- 3 = 字段数据为大写字母数字 空格、句号、逗号、虚线和斜线（ 、 、 ）
- 4 = 字段数据为大写字母数字 空格（ 、 、 ） 无
- 5 = 字段数据为完整 位字符集
- 6 = 字段数据为完整 位字符集

int aspectRatio

[in] 长宽比。

- 1 = 正方形
- 2 = 矩形

char text*

[in] 条码数据。

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.71 ZPL_GetPrinterName

此函数功能为获取打印机型号。

```
int ZPL_GetPrinterName(  
  
    void* handle,  
  
    char* name  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* name  
    [in] 打印机型号。
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.72 ZPL_GetPrinterSeriesNumber

此函数功能为获取打印机序列号。

```
int ZPL_GetPrinterSeriesNumber(  
  
    void* handle,  
  
    char* sn  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* sn  
    [in] 打印机序列号。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.73 ZPL_GetPrinterOdometer

此函数功能为获取打印里程数。

```
int ZPL_GetPrinterOdometer(  
  
    void* handle,  
  
    char* meters  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* meters  
    [in] 打印里程数。
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.74 ZPL_GetPrinterFonts

此函数功能为获取打印机内置字体。

```
int ZPL_GetPrinterFonts(  
  
    void* handle,  
  
    char* fonts  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
char* fonts  
    [in] 打印机内置字体，格式为 E_xxx。
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.75 ZPL_SetPrinterInstruction

此函数功能为设置打印指令集。

```
int ZPL_SetPrinterInstruction(  
  
    void* handle,  
  
    int type  
  
);
```

参数:

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int type  
    [in] 指令集类型。0: ZPL, 1: cpcl
```

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.76 ZPL_SetPrinterNetMode

此函数功能为设置 wifi 模式。

```
int ZPL_SetPrinterNetMode(  
  
    void* handle,  
  
    int mode  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int mode  
    [in] wifi 模式。(0:close, 1:sta, 2: ap)
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.76 ZPL_SetPrinterNetSSID

此函数功能为设置 wifi SSID。

```
int ZPL_SetPrinterNetSSID(  
  
    void* handle,  
  
    int mode,  
  
    const TCHAR* ssid  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int mode  
    [in] wifi 模式。(1:sta, 2: ap)  
const TCHAR* ssid  
    [in] ssid 数据（范围：1-32）
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.77 ZPL_SetPrinterNetPwdSwitch

此函数功能为设置 wifi 密码开关。

```
int ZPL_SetPrinterNetPwdSwitch(  
  
    void* handle,  
  
    int mode,  
  
);
```

参数:

void handle*

[in,out] 创建的目标打印机对象。

int mode

[in] wifi 密码开关。(0:关闭, 1: 开启)

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.78 ZPL_SetPrinterNetPwd

此函数功能为设置 wifi 密码。

```
int ZPL_SetPrinterNetPwd(  
  
    void* handle,  
  
    int mode,  
  
    const TCHAR*  pwd  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int mode  
    [in] wifi 模式。(1:sta, 2: ap)  
const TCHAR*  pwd  
    [in] 密码（范围：1-64）
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.79 ZPL_SetPrinterNetDHCP

此函数功能为设置 wifi DHCP。

```
int ZPL_SetPrinterNetDHCP(  
  
    void* handle,  
  
    int mode,  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int mode  
    [in] 是否开启 (0:关闭, 1: 开启)
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.80 ZPL_SetPrintIpAddress

此函数功能为设置 wifi IP 地址。

```
int ZPL_SetPrintIpAddress(  
  
    void* handle,  
  
    int mode,  
  
    const TCHAR*  ipaddress  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
int mode  
    [in] wifi 模式。(0:关闭, 1: 开启)  
const TCHAR*  ipaddress  
    [in] ip 地址。格式为：xxx.xxx.xxx.xxx
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.81 ZPL_SetPrintSubnetMask

此函数功能为设置 wifi 子网掩码。

```
int ZPL_SetPrintSubnetMask(  
  
    void* handle,  
  
    const TCHAR* mask  
  
);
```

参数：

void* handle
[in,out] 创建的目标打印机对象。
const TCHAR* mask
[in] 子网掩码。格式为：xxx.xxx.xxx.xxx

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.82 ZPL_SetPrintDefaultGateway

此函数功能为设置 wifi 默认网关。

```
int ZPL_SetPrintDefaultGateway(  
  
    void* handle,  
  
    const TCHAR* gateway  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
const TCHAR* gateway  
    [in] 默认网关。格式为：xxx.xxx.xxx.xxx
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.83 ZPL_SetPrinterBluetoothSSID

此函数功能为设置蓝牙 SSID。

```
int ZPL_SetPrinterBluetoothSSID(  
  
    void* handle,  
  
    const TCHAR*  ssid  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
const TCHAR*  ssid  
    [in] ssid 数据（范围：1-32）
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.84 ZPL_SetPrinterBluetoothPIN

此函数功能为设置蓝牙 pin 码。

```
int ZPL_SetPrinterBluetoothPIN(  
  
    void* handle,  
  
    const TCHAR* pin  
  
);
```

参数：

```
void* handle  
    [in,out] 创建的目标打印机对象。  
const TCHAR* pin  
    [in] pin 数据（范围：1-32）
```

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.85 ZPL_SetPrinterSleepTime

此函数功能为设置休眠时间。

```
int ZPL_SetPrinterSleepTime(  
  
    void* handle,  
  
    int time,  
  
);
```

参数:

- void* handle*
[in,out] 创建的目标打印机对象。
- int time*
[in] 休眠时间 (范围: 0-999, 单位: 分钟)

返回值:

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.86 ZPL_SetPrinterShutdownTime

此函数功能为设置自动关机时间。

```
int ZPL_SetPrinterShutdownTime(  
  
    void* handle,  
  
    int time,  
  
);
```

参数：

- void* handle*
[in,out] 创建的目标打印机对象。
- int time*
[in] 自动关机时间 (范围：0-999, 单位：分钟)

返回值：

错误代码	值	描述
E_SUCCESS	0	正常
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.87 ZPL_FirmwareUpgrade

此函数功能为升级打印机固件，仅适用于 HM-T300 PRO。此接口需在 PrinterCreator 之前或者 PrinterDestroy 之后调用。

```
int ZPL_FirmwareUpgrade(  
  
    void* handle,  
  
    const TCHAR* cFileName,  
  
    const TCHAR* model,  
  
    const TCHAR* ioSettings  
  
);
```

参数：

- void* handle*
[in,out] 创建的目标打印机对象。
- const TCHAR* cFileName*
[in] 固件文件地址
- const TCHAR* model*
[in] 打印机机型名称
- const TCHAR* ioSettings*
[in] 设置连接目标打印机的通讯端口参数。具体内容查看下表：

配置列表：

类别	配置	描述	示例
USB	USB [,Position/Model/PortNum]	USB：连接任一本公司 USB 打印机。 USB[,Position] ：当同时连接本公司多台打印机时，可以通过 USB 位置信息(Position 参数)来指定连接某一特定 USB 端口的打印机。	USB USB,Port_#0004.Hub_#0003 USB,LPG4 USB,USB001

NET	NET , <i>IP Add (IPv4)[,Port]</i>	指定网络打印机的 IP 地址和端口。如果不指定端口，默认端口是9100。	NET,192.168.0.36 NET,192.168.0.36,9100
COM	COMn ,BAUDRATE_ <i>E_rate</i>	指定连接的串口端口号和波特率。	COM5,BAUDRATE_19200
LPT	LPTn	指定连接的并口端口号。	LPT1

注： []表示可选参数。

返回值：

错误代码	值	描述
E_SUCCESS	1	升级成功
E_FAILED	0	升级失败
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.88 ZPL_FontDownload

此函数功能为字库下载，仅适用于 HM-T300 PRO。此接口需在 PrinterCreator 之前或者 PrinterDestroy 之后调用。

```
int ZPL_FontDownload(  
  
    void* handle,  
  
    const TCHAR* cFileName,  
  
    const TCHAR* model,  
  
    const TCHAR* ioSettings  
  
);
```

参数：

- void* handle*
[in,out] 创建的目标打印机对象。
- const TCHAR* cFileName*
[in] 字库文件地址
- const TCHAR* model*
[in] 打印机机型名称
- const TCHAR* ioSettings*
[in] 设置连接目标打印机的通讯端口参数。具体内容查看下表：

配置列表：

类别	配置	描述	示例
USB	USB [,Position/Model/PortNum]	USB：连接任一本公司 USB 打印机。 USB[,Position] ：当同时连接本公司多台打印机时，可以通过 USB 位置信息(Position 参数)来指定连接某一特定 USB 端口的打印机。	USB USB,Port_#0004.Hub_#0003 USB,LPG4 USB,USB001

NET	NET , <i>IP Add (IPv4)[,Port]</i>	指定网络打印机的 IP 地址和端口。如果不指定端口，默认端口是9100。	NET,192.168.0.36 NET,192.168.0.36,9100
COM	COMn ,BAUDRATE_ <i>E_rate</i>	指定连接的串口端口号和波特率。	COM5,BAUDRATE_19200
LPT	LPTn	指定连接的并口端口号。	LPT1

注： []表示可选参数。

返回值：

错误代码	值	描述
E_SUCCESS	1	下载成功
E_FAILED	0	下载失败
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时

4.89 ZPL_VectorFontDownload

此函数功能为矢量字体下载，仅适用于 HM-T300 PR0。此接口需在 PrinterCreator 之前或者 PrinterDestroy 之后调用。

```
int ZPL_VectorFontDownload(  
  
    void* handle,  
  
    const TCHAR* cFileName,  
  
    const TCHAR* model,  
  
    const TCHAR* ioSettings  
  
);
```

参数：

- void* handle*
[in,out] 创建的目标打印机对象。
- const TCHAR* cFileName*
[in] 矢量字体文件地址
- const TCHAR* model*
[in] 打印机机型名称
- const TCHAR* ioSettings*
[in] 设置连接目标打印机的通讯端口参数。具体内容查看下表：

配置列表：

类别	配置	描述	示例
USB	USB [,Position/Model/PortNum]	USB：连接任一本公司 USB 打印机。 USB[,Position] ：当同时连接本公司多台打印机时，可以通过 USB 位置信息(Position 参数)来指定连接某一特定 USB 端口的打印机。	USB USB,Port_#0004.Hub_#0003 USB,LPG4 USB,USB001

NET	NET , <i>IP Add (IPv4)[,Port]</i>	指定网络打印机的 IP 地址和端口。如果不指定端口，默认端口是9100。	NET,192.168.0.36 NET,192.168.0.36,9100
COM	COMn ,BAUDRATE_ <i>E_rate</i>	指定连接的串口端口号和波特率。	COM5,BAUDRATE_19200
LPT	LPTn	指定连接的并口端口号。	LPT1

注： []表示可选参数。

返回值：

错误代码	值	描述
E_SUCCESS	1	下载成功
E_FAILED	0	下载失败
E_INVALID_PARAMETER	-1	无效的参数
E_NOT_ENOUGH_BUFFER	-2	内存不足
E_INVALID_MODEL_TYPE	-3	该机型不支持此功能
E_BAD_HANDLE	-6	句柄无效
E_IO_PORT_NOT_OPEN	-309	端口未打开
E_IO_WRITE_FAILED	-321	写入数据失败
E_IO_WRITE_TIMEOUT	-322	写入数据超时