

# 更改记录

版本	日期	说明	拟制	审核	批准	日期
V1.0	2020/7/11	初稿	周海立			
V1.1	2021/8/28	修改 SetCallback 返回说明	周海立			

## HidPosApi 简介

以动态库的形式提供开发 HID（POS）扫码设备的接口，方便用户进行二次开发。包括：打开设备、关闭设备、获取设备连接状态、获取设备 SN 号、获取设备扫码数据、发送数据以及设置接收超时时间接口。

### 1 接口列表

序号	接口名称	功能说明	备注
1	OpenDevice (int p_nVid, int p_nPid)	打开设备	
2	CloseDevice ()	关闭设备	
3	ConnectStaus ()	获取设备连接状态	
4	GetDeviceSn (char *pSn, int *pSnLength)	获取设备 SN 号	
5	SetReceiveTimeOut (int nTimeOutMillisecond)	设置接收超时时间	
6	WriteUsb (LPCVOID lpBuffer, DWORD dwSize)	发送数据	
7	SetCallback (ReceivePosData pf)	设置回调函数	
8	typedef int(ReceivePosData)(const char *pData, int pDataLength, unsigned char cResult);	回调函数	

## 2 HidPosApi 接口定义

### 2.1 打开设备

```
bool OpenDevice(int p_nVid, int p_nPid);
```

参数：

p\_nVid: 设备的VID

p\_nPid: 设备的PID

返回:

true:打开设备成功, false: 打开设备失败。

## 2.2 关闭设备

`bool CloseDevice();`

参数: 无。

返回:

true: 关闭成功, false: 关闭失败。

## 2.3 获取设备连接状态

`bool ConnectStaus()`

参数: 无。

返回:

true: 设备已连接, false: 设备未连接。

## 2.4 获取设备 SN 号

`bool GetDeviceSn(char *pSn, int *pSnLength)`

参数:

pSn: 传入接收 SN BUF 的指针 (最少开辟 16 字节长度),

pSnLength: SN 长度。

返回:

true: 获取成功, false: 获取失败。

## 2.5 设置接收超时时间

`bool SetReceiveTimeOut(int nTimeOutMillisecond)`

参数:

nTimeOutMillisecond: 超时时间数值 (50-5000ms), 默认设置 100ms

返回:

true: 设置成功, false: 设置失败。

## 2.6 发送数据

`bool WriteUsb(LPCVOID lpBuffer, DWORD dwSize)`

参数:

lpBuffer: 待发送数据的指针,

dwSize: 待发送数据长度。

返回:

true: 发送成功, false: 发送失败。

## 2.7 设置回调函数

```
int SetCallback(ReceivePosData pf);
```

参数:

pf: 回调函数指针。

返回: 0

## 2.8 回调函数说明

```
typedef int(*ReceivePosData)(const char *pData, int pDataLength, unsigned char cResult);
```

参数:

pData: 接收到的数据指针,

pDataLength: 接收到的数据长度,

cResult: 接收数据状态, 0: 数据包接收正常 1: 数据包接收错误 (数据可能不完整)