

SynoAPIEx 函数接口库使用手册

版本 1.0, 2011 年 12 月

杭州晟元芯片技术有限公司

杭州市西湖区天目山路 176 号 17 幢 2 楼



声明

下列文件包涵晟元芯片技术有限公司（以下简称为晟元）的私有信息。这些信息是精确、可靠的，在没有本公司管理层许可的情况下，第三方不得使用或随意泄露；当然，任何在没有授权、特殊条件、限制或告知的情况下对此信息的复制和擅自修改都是侵权行为。

在任何时间,无需告知任何方的情况下，晟元有权对本公司产品和服务进行更改、添加、删除、改进以及其他任何变更。在对本公司产品的使用中，晟元不背负任何责任或义务；而第三方在使用中则不得侵害任何专利或其他知识产权。

所有产品的售出都受制于本公司在订购承认书里的销售条款和条件。本公司利用测试、工具、质量控制等技术手段来支持产品的相关性能符合所需规格的一定程度的保证。除了明确的政府书面要求外，没必要执行每款产品的所有参数测试。

除了晟元的logo设计，其他所有的商标或注册商标都是属于各自所有者所有。

晟元芯片技术有限公司2005–2015©版权。版权所有，侵权必究

版本历史

版本	日期	修改内容		
		章节	作者	内容
1.0	2011-12-12		BLUE	初始版本

目录

声明	I
版本历史	II
目录	III
1 概述	1
2 文件说明	2
3 开发环境	3
4 函数接口及宏定义	4
4.1 宏定义	4
4.2 相关函数	5
4.2.1 打开设备	5
4.2.2 关闭设备	6
4.2.3 检测手指并录取图像	6
4.2.4 生成指纹特征	6
4.2.5 比对指纹特征	7
4.2.6 搜索指纹库	7
4.2.7 合并特征	7
4.2.8 存储模版	7
4.2.9 读取模版	8
4.2.10 上传特征到PC	8
4.2.11 PC下载特征到设备	8
4.2.12 上传原始图像到PC	9
4.2.13 PC下载原始图像到设备	9
4.2.14 将图像数据存储为BMP图片	9
4.2.15 从BMP图片读取图像数据	9
4.2.16 删除模版	10
4.2.17 清空指纹库	10
4.2.18 读系统参数表	10
4.2.19 读Flash信息页	11
4.2.20 读模版个数	11
4.2.21 指纹图细化	11
4.2.22 设置设备握手口令	11
4.2.23 验证设备握手口令	12
4.2.24 读记事本	12
4.2.25 写记事本	12
4.2.26 自动注册	12
4.2.27 写模块寄存器	13
4.2.28 设置波特率	13
4.2.29 设置安全等级	13
4.2.30 设置数据包大小	13
4.2.31 上传缓冲区特征文件到PC	14

4.2.32	从PC下载一个指纹特征到设备缓冲区	14
4.2.33	从芯片获取一个随机数	14
4.2.34	设置芯片地址	15
4.2.35	自动验证指纹	15
4.2.36	设置GPIO值	15
4.2.37	读模版索引表	15
4.2.38	自定义发送包	16
4.2.39	根据错误码获取错误信息	16
4.2.40	设置模版特征大小	16
4.2.41	获取模版特征大小	17
4.2.42	烧写数据	17
4.2.43	通过 6xx擦除 5xx标志（专用函数）	17
5	其他	18
5.1	说明	18

1 概述

本文档所指 API 库函数，即为我公司 SYDemo 的程序开发函数接口库。

调用方可根据需要调用相关 API 函数来完成一些对下位机的命令式操作。

2 文件说明

下面的函数用到如下相关定义：

SynoAPIEx.dll : PS 协议 API 函数库。

protocol.h : API 库支持的函数的 C/C++ 方式声明文件。

3 开发环境

SynoAPIEx.dll 基于 Microsoft Visual Studio C++ 6.0 开发。

4 函数接口及宏定义

4.1 宏定义

```

#define PS_MAXWAITTIME 2000
#define DELAY_TIME      150
//////////错误返回码//////////
#define PS_OK           0x00
#define PS_COMM_ERR     0x01
#define PS_NO_FINGER    0x02
#define PS_GET_IMG_ERR  0x03
#define PS_FP_TOO_DRY   0x04
#define PS_FP_TOO_WET   0x05
#define PS_FP_DISORDER  0x06
#define PS_LITTLE_FEATURE 0x07
#define PS_NOT_MATCH     0x08
#define PS_NOT_SEARCHED  0x09
#define PS_MERGE_ERR     0x0a
#define PS_ADDRESS_OVER  0x0b
#define PS_READ_ERR      0x0c
#define PS_UP_TEMP_ERR   0x0d
#define PS_RECV_ERR      0x0e
#define PS_UP_IMG_ERR    0x0f
#define PS_DEL_TEMP_ERR  0x10
#define PS_CLEAR_TEMP_ERR 0x11
#define PS_SLEEP_ERR     0x12
#define PS_INVALID_PASSWORD 0x13
#define PS_RESET_ERR     0x14
#define PS_INVALID_IMAGE 0x15
#define PS_HANGOVER_UNREMOVE 0x17

//////////缓冲区//////////
#define CHAR_BUFFER_A    0x01
#define CHAR_BUFFER_B    0x02
#define MODEL_BUFFER     0x03

//////////串口号//////////
#define COM1              0x01
#define COM2              0x02
#define COM3              0x03

//////////波特率//////////

```

```

#define BAUD_RATE_9600      0x01
#define BAUD_RATE_19200    0x02
#define BAUD_RATE_38400    0x04
#define BAUD_RATE_57600    0x06 //default
#define BAUD_RATE_115200   0x0C

#define MAX_PACKAGE_SIZE_   350 // 数据包最大长度
#define CHAR_LEN_AES1711    1024 // 512->1024 [2009.11.12] AES1711使用大小模版
#define CHAR_LEN_NORMAL     512 // 512 通用版本使用大小的模版

#define DEVICE_USB          0
#define DEVICE_COM          1
#define DEVICE_UDisk       2

#define IMAGE_X 256
#define IMAGE_Y 288

```

4.2 相关函数

4.2.1 打开设备

1. 函数原型: `int WINAPI PSOpenDeviceEx(`

```

HANDLE* pHandle, // [OUT] 设备句柄

int nDeviceType, // [IN] 设备类型, DEVICE_USB、DEVICE_COM、DEVICE_UDisk

int iCom=1,      // [IN] 串口号 1~16

int iBaud=1,     // [IN] 波特率 1~n, 其值*9600=实际波特率值

int nPackageSize=2, // [IN] 数据包大小, 实际数据包大小= 32*(0x01<nPackageSize)

int iDevNum=0);  // [IN] 设备号, 0~n

```

函数说明: 打开 PS 设备。

返回值: 成功-PS_OK;

2. 函数原型: `int WINAPI PSAutoOpen(`

```

HANDLE* pHandle, // [OUT] 设备句柄

int *type,       // [OUT] 设备类型, DEVICE_USB、DEVICE_COM、DEVICE_UDisk

int nAddr=0xFFFFFFFF, // [IN] 设备地址

UINT uPwd=0x00,    // [IN] 验证密码

int bVfy=1);      // [IN] 是否校验密码 0-不校验 1-校验

```

函数说明：自动搜索打开 PS 设备。

返回值：成功-PS_OK；

4.2.2 关闭设备

函数原型：`int WINAPI PSCloseDeviceEx(
HANDLE hHandle);` // [IN] 设备句柄

函数说明：关闭 PS 设备。

返回值：成功-PS_OK；

4.2.3 检测手指并录取图像

1. 函数原型：`int WINAPI PSGetImage(
HANDLE hHandle, // [IN] 设备句柄
int nAddr);` // [IN] 设备地址

函数说明：检测手指并录取图像。 命令码 0x01

返回值：成功-PS_OK；

2. 函数原型：`int WINAPI PSGetImage_Enroll(
HANDLE hHandle, // [IN] 设备句柄
int nAddr);` // [IN] 设备地址

函数说明：检测手指并录取图像；光学特殊命令，注册时使用。 命令码 0x29

返回值：成功-PS_OK；

4.2.4 生成指纹特征

函数原型：`int WINAPI PSGenChar(
HANDLE hHandle, // [IN] 设备句柄
int nAddr, // [IN] 设备地址
int iBufferID);` // [IN] BufferID 号。1, 2, 3, 4

函数说明：根据原始图像生成指纹特征。

返回值：成功-PS_OK；

4.2.5 比对指纹特征

函数原型: **int** WINAPI PSMatch(
 HANDLE hHandle, //[IN] 设备句柄
 int nAddr, //[IN] 设备地址
 int* iScore); //[OUT] 比对分值, 值越大越相似, 一般阈值为 50

函数说明: 精确比对 CharBufferA 与 CharBufferB 中的特征文件。

返回值: 成功-PS_OK;

4.2.6 搜索指纹库

1. 函数原型: **int** WINAPI PSSearch(
 HANDLE hHandle, //[IN] 设备句柄
 int nAddr, //[IN] 设备地址
 int iBufferID, //[IN] BufferID 号。1, 2, 3, 4
 int iStartPage, //[IN] 搜索起始页
 int iPageNum, //[IN] 搜索页数
 int *iMbAddress, //[OUT] 如果搜索成功, 返回搜索到的 ID 地址值
 int *iscore=NULL); // [OUT] 搜索到 ID 的分值

函数说明: 以 CharBufferA 或 CharBufferB 中的特征文件搜索整个或部分指纹库。

返回值: 成功-PS_OK;

4.2.7 合并特征

函数原型: **int** WINAPI PSRegModule(
 HANDLE hHandle, //[IN] 设备句柄
 int nAddr); //[IN] 设备地址

函数说明: 将 CharBufferA 与 CharBufferB 中的特征文件合并生成模板存于 ModelBuffer。

返回值: 成功-PS_OK;

4.2.8 存储模版

函数原型: **int** WINAPI PSStoreChar(
 HANDLE hHandle, //[IN] 设备句柄
 int nAddr, //[IN] 设备地址

```
int iBufferID,      //[IN] BufferID 号。1, 2, 3, 4
int iPageID);      //[IN] 指纹库位置号
```

函数说明：将 ModelBuffer 中的文件储存到 flash 指纹库中。

返回值： 成功-PS_OK;

4.2.9 读取模版

函数原型： `int WINAPI PSLoadChar(`

```
HANDLE hHandle,      //[IN] 设备句柄
int nAddr,           //[IN] 设备地址
int iBufferID,       //[IN] BufferID 号。1, 2, 3, 4
int iPageID);        //[IN] 指纹库位置号
```

函数说明：从 flash 指纹库中读取一个模板到 ModelBuffer。

返回值： 成功-PS_OK;

4.2.10 上传特征到 PC

函数原型： `int WINAPI PSUpChar(`

```
HANDLE hHandle,      //[IN] 设备句柄
int nAddr,           //[IN] 设备地址
int iBufferID,       //[IN] BufferID 号。1, 2, 3, 4
unsigned char* pTemplet,  //[OUT]PC 数据缓冲区，用于存放特征。需分配大小建议为 1024Bytes
int* iTempletLength);  //[OUT]特征大小。（AES1711-1024 通用-512）
```

函数说明：将特征缓冲区中的文件上传给上位机。

返回值： 成功-PS_OK;

4.2.11 PC 下载特征到设备

函数原型： `int WINAPI PSDownChar(`

```
HANDLE hHandle,      //[IN] 设备句柄
int nAddr,           //[IN] 设备地址
int iBufferID,       //[IN] BufferID 号。1, 2, 3, 4
unsigned char* pTemplet,  //[IN] 特征数据流
int iTempletLength);  //[IN] 特征大小
```

函数说明：从上位机下载一个特征文件到特征缓冲区。

返回值： 成功-PS_OK;

4.2.12 上传原始图像到 PC

函数原型: `int` WINAPI PSUpImage(
 HANDLE hHandle, `//[IN]` 设备句柄
 `int` nAddr, `//[IN]` 设备地址
 `unsigned char*` pImageData, `//[OUT]`PC 数据缓存区, 用于存储原始图像数据流。默认图像数据流为 256*288
 `int*` iImageLength); `//[OUT]`原始图像数据大小。(256*288)

函数说明: 上传原始图像到 PC。

返回值： 成功-PS_OK;

4.2.13 PC 下载原始图像到设备

函数原型: `int` WINAPI PSDownImage(
 HANDLE hHandle, `//[IN]` 设备句柄
 `int` nAddr, `//[IN]` 设备地址
 `unsigned char *`pImageData, `//[IN]` 原始图像数据流
 `int` iLength); `//[IN]` 数据流大小

函数说明: PC 下载原始图像到设备

返回值： 成功-PS_OK;

4.2.14 将图像数据存储为 BMP 图片

函数原型: `int` WINAPI PSImgData2BMP(
 `unsigned char*` pImgData, `//[IN]` 图像数据流
 `const char*` pImageFile); `//[IN]` BMP 图片保存路径

函数说明: 将图像数据存储为 BMP 图片

返回值： 成功-PS_OK;

4.2.15 从 BMP 图片读取图像数据

函数原型: `int` WINAPI PSGetImgDataFromBMP(
 HANDLE hHandle, `//[IN]` 设备句柄
 `const char *`pImageFile, `//[IN]` BMP 图片路径

```

unsigned char *pImageData,    //[OUT]PC 数据缓存区,用于存放解析出的图像数据流。(默认流大小 256*288)

int *pnImageLen);            //[OUT]图像数据流大小。(256*288)

```

函数说明: 从 BMP 图片读取图像数据

返回值: 成功-PS_OK;

4.2.16 删除模版

```

函数原型:      int WINAPI    PSDelChar(

HANDLE hHandle,    //[IN] 设备句柄

int nAddr,          //[IN] 设备地址

int iStartPageID,  //[IN] 起始指纹模版 ID 号

int nDelPageNum);  //[IN] 要删除的模版个数

```

函数说明: 删除 flash 指纹库中的一个或多个特征文件

返回值: 成功-PS_OK;

4.2.17 清空指纹库

```

函数原型:      int WINAPI    PSEmpty(

HANDLE hHandle,    //[IN] 设备句柄

int nAddr);        //[IN] 设备地址

```

函数说明: 清空 flash 指纹库

返回值: 成功-PS_OK;

4.2.18 读系统参数表

```

函数原型:      int WINAPI    PSReadParTable(

HANDLE hHandle,          //[IN] 设备句柄

int nAddr,                //[IN] 设备地址

unsigned char* pParTable);  //[OUT]基本参数信息。需分配空间大小 16Bytes

```

函数说明: 读取模块的基本参数(波特率,包大小等)

参数表前 16 个字节存放了模块的基本通信和配置信息,称为模块的基本参数

返回值: 成功-PS_OK;

4.2.19 读 Flash 信息页

函数原型: `int WINAPI PSReadInfPage(`
 `HANDLE hHandle, // [IN] 设备句柄`
 `int nAddr, // [IN] 设备地址`
 `unsigned char* pInf); // [OUT] 输出 Flash 信息页数据流。(512Bytes)`

函数说明: 读取 FLASH Information Page 所在的信息页(512bytes)

返回值: 成功-PS_OK;

4.2.20 读模版个数

函数原型: `int WINAPI PSTemplateNum(`
 `HANDLE hHandle, // [IN] 设备句柄`
 `int nAddr, // [IN] 设备地址`
 `int *iMbNum); // [OUT] 模版个数`

函数说明: 读有效模版个数

返回值: 成功-PS_OK;

4.2.21 指纹图细化

函数原型: `int WINAPI PSGenBinImage(`
 `HANDLE hHandle, // [IN] 设备句柄`
 `int nAddr, // [IN] 设备地址`
 `int nImgType); // [IN] 细化类型 0-二值图 1-细化图 2-特征点细化图`

函数说明: 对图像缓冲区中的指纹图像进行处理并生成细化指纹图像。

返回值: 成功-PS_OK;

4.2.22 设置设备握手口令

函数原型: `int WINAPI PSSetPwd(`
 `HANDLE hHandle, // [IN] 设备句柄`
 `int nAddr, // [IN] 设备地址`
 `unsigned char* pPassword); // [IN] 握手口令(4 Bytes)`

函数说明: 设置模块握手口令。

返回值: 成功-PS_OK;

4.2.23 验证设备握手口令

函数原型: `int WINAPI PSVfyPwd(`
 `HANDLE hHandle, //[[IN] 设备句柄`
 `int nAddr, //[[IN] 设备地址`
 `unsigned char* pPassword); //[[IN] 握手口令(4 Bytes)`

函数说明: 验证模块握手口令

返回值: 成功-PS_OK;

4.2.24 读记事本

函数原型: `int WINAPI PSReadInfo(`
 `HANDLE hHandle, //[[IN] 设备句柄`
 `int nAddr, //[[IN] 设备地址`
 `int nPage, //[[IN] 页码`
 `unsigned char* UserContent); //[[OUT] 对应页码的记事本信息(32 Bytes)`

函数说明: 模块内部为用户开辟了 512bytes 的 FLASH 空间用于存放用户数据, 该存储空间称为用户记事本, 该记事本逻辑上被分成 16 个页, 读记事本命令用于读取指定的记事本页的 32bytes 数据

返回值: 成功-PS_OK;

4.2.25 写记事本

函数原型: `int WINAPI PSWriteInfo(`
 `HANDLE hHandle, //[[IN] 设备句柄`
 `int nAddr, //[[IN] 设备地址`
 `int nPage, //[[IN] 页码`
 `unsigned char* UserContent); //[[IN] 记事本信息流(32 Bytes)`

函数说明: 写入用户的 32bytes 数据到指定的记事本页。

返回值: 成功-PS_OK;

4.2.26 自动注册

函数原型: `int WINAPI PSEnroll(`
 `HANDLE hHandle, //[[IN] 设备句柄`
 `int nAddr, //[[IN] 设备地址`

```
int* nID);          // [OUT] 页码（相配指纹模版）
```

函数说明：采集一次指纹注册模板，在指纹库中搜索空位并存储，返回存储 ID。

返回值：成功-PS_OK;

4.2.27 写模块寄存器

函数原型： `int WINAPI PSWriteReg(`

```
HANDLE hHandle,    // [IN] 设备句柄
```

```
int nAddr,          // [IN] 设备地址
```

```
int iRegAddr,       // [IN] 寄存器序号 4/5/6
```

```
int iRegValue);     // [IN] 内容
```

函数说明：写模块寄存器。（本函数具体封装见 4.2.28~30）

返回值：成功-PS_OK;

4.2.28 设置波特率

函数原型： `int WINAPI PSSetBaud(`

```
HANDLE hHandle,    // [IN] 设备句柄
```

```
int nAddr,          // [IN] 设备地址
```

```
int nBaudNum);      // [IN] 波特率。9600 的倍数 N
```

函数说明：写模块寄存器—波特率设置

返回值：成功-PS_OK;

4.2.29 设置安全等级

函数原型： `int WINAPI PSSetSecurLevel(`

```
HANDLE hHandle,    // [IN] 设备句柄
```

```
int nAddr,          // [IN] 设备地址
```

```
int nLevel);        // [IN] 安全等级 1~5 值越大，等级越高
```

函数说明：写模块寄存器—安全等级设置

返回值：成功-PS_OK;

4.2.30 设置数据包大小

函数原型： `int WINAPI PSSetPacketSize(`

```

HANDLE hHandle,    //[IN] 设备句柄

int nAddr,         //[IN] 设备地址

int nSize);        //[IN] 包大小 0-32Bytes 1-64Bytes 2-128Bytes 3-256Bytes

```

函数说明：写模块寄存器—数据包大小设置

返回值： 成功-PS_OK;

4.2.31 上传缓冲区特征文件到 PC

```

函数原型:      int WINAPI    PSUpChar2File(

HANDLE hHandle,    //[IN] 设备句柄

int nAddr,         //[IN] 设备地址

int iBufferID,     //[IN] 缓冲区号。1, 2, 3, 4

const char* pFileName);//[IN] 文件路径, 用于存储特征数据

```

函数说明：这是一个封装的组合函数（PSUpChar），将指定特征缓冲区中的特征文件上传给上位机，并保存为文件。

返回值： 成功-PS_OK;

4.2.32 从 PC 下载一个指纹特征到设备缓冲区

```

函数原型:      int WINAPI    PSDownCharFromFile(

HANDLE hHandle,    //[IN] 设备句柄

int nAddr,         //[IN] 设备地址

int iBufferID,     //[IN] 缓冲区号。1, 2, 3, 4

const char* pFileName);//[IN] 特征数据文件路径

```

函数说明：从 PC 特征数据文件下载特征到设备缓冲区。

返回值： 成功-PS_OK;

4.2.33 从芯片获取一个随机数

```

函数原型:      int WINAPI    PSGetRandomData(

HANDLE hHandle,    //[IN] 设备句柄

int nAddr,         //[IN] 设备地址

unsigned char* pRandom);    //[OUT] 随机数流 (4Bytes)

```

函数说明：令芯片生成一个随机数并返回给上位机。

返回值： 成功-PS_OK;

4.2.34 设置芯片地址

函数原型: `int WINAPI PSetChipAddr(`
 `HANDLE hHandle, // [IN] 设备句柄`
 `int nAddr, // [IN] 设备地址`
 `unsigned char* pChipAddr); // [IN] 芯片地址 (4Bytes)`

函数说明: 设置芯片地址。

返回值: 成功-PS_OK;

4.2.35 自动验证指纹

函数原型: `int WINAPI PSIdentify(`
 `HANDLE hHandle, , // [IN] 设备句柄`
 `int nAddr, // [IN] 设备地址`
 `int *iMbAddress); // [OUT] 页码 (相匹配指纹模版)`

函数说明:

1. 自动采集指纹, 在指纹库中搜索目标模板并返回搜索结果。
2. 如果目标模板同当前采集的指纹比对得分大于最高阈值, 并且目标模板为不完整特征则以采集的特征更新目标模板的空白区域。

返回值: 成功-PS_OK;

4.2.36 设置 GPIO 值

函数原型: `int WINAPI PSDoUserDefine(`
 `HANDLE hHandle, // [IN] 设备句柄`
 `int nAddr, // [IN] 设备地址`
 `int GPIO, // [IN] GPIO 号 0~7`
 `int STATE); // [IN] 状态值 0-输出低电平 1-输出高电平`

函数说明: 用于实现用户 GPIO 输出控制命令。

返回值: 成功-PS_OK;

4.2.37 读模版索引表

函数原型: `int WINAPI PSReadIndexTable(`
 `HANDLE hHandle, // [IN] 设备句柄`

```

int nAddr,           //[IN] 设备地址

int nPage,           //[IN] 页码 0, 1, 2, 3 分别对应模版从 0-255, 256-511, 512-767, 768-1023 的索引

unsigned char* UserContent);//[OUT]模版信息流, 32Bytes。每 1 位代表一个模版, 1 表示录入, 0 未录入

```

函数说明：读取录入模版的索引表。

返回值： 成功-PS_OK;

4.2.38 自定义发送包

函数原型： `int WINAPI PS_SB(`

```

HANDLE hHandle,           //[IN] 设备句柄

int nAddr,               //[IN] 设备地址

unsigned char *pSendData,  //[IN] 发送数据流

int iLen,                //[IN] 发送数据流长度

unsigned char *pRecvData,  //[OUT] 输出数据流区。当值为 NULL 时，不输出值

int *pRecvLen,           //[OUT] 输出数据的长度

int flag=1);             //[IN] 0-不接收应答 1-接收处理应答

```

函数说明：自定义命令包流，可模拟所有命令，**慎用**。

返回值： 成功-PS_OK;

4.2.39 根据错误码获取错误信息

函数原型： `char* WINAPI PSErr2Str(`

```

int nErrCode);           //[IN] 错误码

```

函数说明：根据错误码获取错误信息。

返回值： 成功-信息数据流

4.2.40 设置模版特征大小

函数原型： `int WINAPI PSSetCharLen(`

```

int nLen = CHAR_LEN_NORMAL);           //[IN] 模版特征大小。通用-512 AES1711-1024

```

函数说明：设置模版特征大小。

返回值： 成功-PS_OK;

4.2.41 获取模版特征大小

函数原型: `int` WINAPI `PSGetCharLen`(
 `int` *pnLen); `//[OUT]` 模版大小

函数说明: 读取当前模版特征大小。

返回值: 成功-`PS_OK`;

4.2.42 烧写数据

函数原型: `int` WINAPI `PSBurnCode`(
 `HANDLE` hHandle, `//[IN]` 设备句柄
 `int` nAddr, `//[IN]` 设备地址
 `int` nType, `//[IN]` 升级类型 1-完整升级, 0-只升级信息页
 `unsigned char` *pImageData, `//[IN]` 烧写数据流
 `int` iLength); `//[IN]` 数据长度

函数说明: PS 烧写数据。

返回值: 成功-`PS_OK`

4.2.43 通过 6xx 擦除 5xx 标志 (专用函数)

函数原型: `int` WINAPI `PSClear5xxFLAG`(
 `HANDLE` hHandle, `//[IN]` 设备句柄
 `int` nAddr); `//[IN]` 设备地址

函数说明: 通过 6xx 擦除 5xx 设备标志位。

返回值: 成功-`PS_OK`

5 其他

5.1 说明

1. AES1711 的特征模版大小是 1024，但是默认的通用版本特征模版大小为 512.所以，如果是 AES1711 的设备，使用前，请先调用函数 `PSSetCharLen` 将特征模版大小设置为 1024。
你可以使用函数 `PSGetCharLen` 查看当前的特征模版大小
2. 扩增了对串口 10 及以上的串口号设备的支持
3. 关于波特率参数 `iBaud`，其值表示的波特率是 `iBaud` 乘以 9600
比如，`iBaud = 6`，那么波特率为 $9600 * 6 = 57600$
4. 去除了对 `BufferID` 的值范围限制。`BufferID` 起始值为 1