

AP-Navigator SDK-API 函数编程手册

V1.0

2022. 11. 25

HOURS

www.hours-shop.com

目录

第一章 函数使用介绍	3
1.1. API 编程系统软件架构	3
1.2. 基于 visual studio 2019 C++ 软件开发的简介	3
第二章 光谱共焦位移传感器功能描述	3
2.1. 连接控制器	3
2.2. 使用 SDK 进行测量的步骤	4
2.3. 感测头设置	6
2.4. 应用场景（测量方法设置）	7
第三章 函数列表	23
3.1. 连接控制器	23
3.2. 配置控制器参数	26
3.3. 配置探头	33
3.4. 配置测量方法	36
3.5. 测量	39

第一章 函数使用介绍

1.1. API 编程系统软件架构

AP-Navigator 提供了动态链接库，用户可选择调用动态链接库提供的 API 函数来完成所需的各种功能。通过在 PC 机中编写应用程序代码，应用程序调用动态链接库提供的 API 函数来执行相关功能。动态链接库包含下面三个文件 AP-SDK.dll, AP-SDK.h, AP-SDK.lib, 动态库提供标准 C 接口，采用 VS 2017 进行编译。并为 x64 环境提供了动态链接库：

文件夹“x64”中的动态链接库适用于在 64 位环境下编程使用。

在 Windows 系统下，用户可以使用任何能够支持动态链接库的开发工具来开发应用程序。

1.2. 基于 visual studio 2019 C++ 软件开发的简介

略

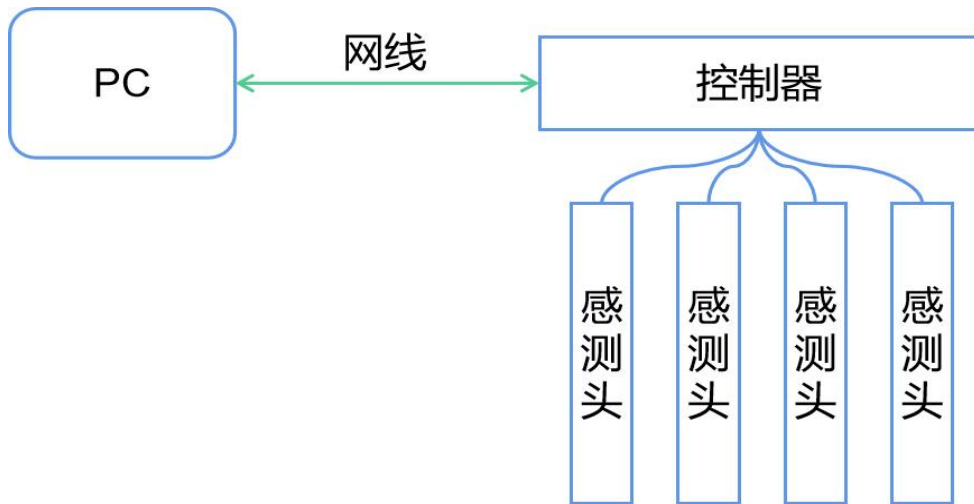
第二章 光谱共焦位移传感器功能描述

2.1. 连接控制器

设备组网说明：

1、PC 通过网口与控制器连接

2、SDK 在同一时间只能连接一个控制器，当需要连接其它控制器时需要关闭当前控制器连接，再连接其它控制器。



相关 API

1、设置所使用的控制器的类型值，该函数必须在调用其他接口之前调用

`int BS_SetDeviceSelectType`

2、初始化 SDK，函数调用前确认插入匹配的加密狗；

`int BS_InitSDK`

3、获取当前 pc 连接的控制器数量

`int BS_GetControllerNum();`

4、获取所有控制的序列号 SN

`int BS_GetControllerSN(pDevSNList, devNum);`

5、连接控制器

`int BS_CreateControllerObj(pDevSN);`

2.2. 使用 SDK 进行测量的步骤

STEP	说明
STEP1 连接控制器	
<code>BS_SetDeviceSelectType</code>	设置所使用的控制器的类型值，该函数必须在调用其他接口之前调用.
<code>BS_InitSDK</code>	初始化 SDK，函数调用前确认插入匹配的加密狗；初始化后需要在关闭程序前调用 <code>BS_UnInitSDK</code> 函数释放相关资源。
<code>BS_GetControllerNum</code>	获取控制器数量
<code>BS_GetControllerSN</code>	获取所有控制器 SN

BS_CreateControllerObj	创建控制器实例
STEP2 配置控制器参数	
BS_LoadControllerConfFromFile	通过加载配置文件，配置并启动控制器的所有感测头
BS_LoadLWCalibrationDataFromFile	通过加载校准文件，指定控制器的所有感测头的高度校准数据。
BS_SetSampleFrequency	设置控制器的采样频率
BS_SetExposureTimeValue	设置控制器的曝光时间
BS_SetGainValue	设置控制器的增益值
BS_DarkController	对控制器已启用感测头进行光亮校准
BS_GetDarkRate	获取控制器光亮校准进度
STEP3 配置探头	
BS_GetHeadNum	获取控制器的感测头数量
BS_SetHeadConfig	设置感测头的参数
BS_GetHeadDistRange	获取感测头的量程
STEP4 配置测量方法	
BS_ClearOutConfig	清除测量方法参数
BS_SetOutConfig	配置测量方法参数
STEP5 开始测量	
BS_StartMeasure	开始测量后 SDK 启动测量线程，周期性采集数据
STEP6 读取测量结果	
BS_GetHeadData	获取指定感测头的波形数据
BS_GetOutData	获取指定 Out 项的测量结果
STEP7 停止测量	
BS_StopMeasure	停止测量后 SDK 停止测量线程
STEP8 断开控制器连接	
BS_RemoveControllerObj	删除控制器对象
STEP9 释放 SDK	
BS_UnInitSDK	释放 SDK 资源

2.3. 感测头设置

探头配置的参数项：

- 1、中位数过滤器
- 2、强度阈值
- 3、屏蔽开关
- 4、波峰数量检查
- 5、折射率补正

/材质对应的折射率系数

```
typedef struct BS_RefractiveCoef
```

```
{
```

```
    float nc;
```

```
    float nd;
```

```
    float nf;
```

```
}BS_RefractiveCoef;
```

//感测头参数结构体

```
typedef struct BS_HeadConfig
```

```
{
```

```
    //中位数过滤器参数[0, 7, 15, 31]
```

```
    int medianFilterParam;
```

```
    //强度阈值控制开关[0 为关闭, 1 为开启]
```

```
    int intensityThresholdEnable;
```

```
    //强度阈值
```

```
    double intensityValue;
```

```
    //屏蔽开关[0 为关闭, 1 为开启]
```

```
    int maskEnable;
```

```
    //近端边界[单位:um]
```

```
    double nearBoundary;
```

```
    //远端边界[单位:um]
```

```
    double farBoundary;
```

```
    //峰值形状判断开关[0 关闭, 1 为开启]
```

```
    int peakShapeCheckEnable;
```

```
    //峰值形状判断级别(1 - 5 级)
```

```
    int peakShapeCheckLevel;
```

```
    //波峰数量检查开关[0 关闭, 1 为开启]
```

```

int peakNumCheckEnable;
//需要检查的波峰数，波峰数检查开启时生效
int peakNum;

//折射率补正开关(透明方法时生效)，0 为关闭，1 为开启
int refractiveEnable;
//折射率补正开启的层数
int refractiveNum;
//折射率系数列表(最大 5 层材质)，折射率补正开启时生效
BS_RefractiveCoef refractiveCoefArray[5];
}BS_HeadConfig;

```

配置参数按功能分为三类

1、测试数据修正

在测试透明对象时，通过配置折射率补正值对测试结果进行修正

2、测试数据处理

通过配置中位数过滤器，对感测头测得的数据进行处理

3、波峰筛选

强度阈值：光亮强度超过阈值的波峰才有效。

屏蔽开关：nearBoundary < farBoundary 时，[n,f] 范围内的波峰有效。

nearBoundary > farBoundary 时，(f,n) 范围外的波峰有效

波峰数量检查：

<= peakNum 的波峰有效

2.4. 应用场景（测量方法设置）

SDK 支持的测量方法如下表

编号	测量方法	备注
1	位移	作用：测量非透明物体的高度、位移等值。 处理方法： 1、取得 Near 端最近的有效波峰索引值； 2、根据有效波峰索引值获取对应位移数组中的位移值；
2	位移（透明）	作用：测量透明物体的高度、位移等值。 处理方法：

		<p>1、根据选择的波峰索引值，换算成有效波峰的索引值；</p> <p>2、根据有效波峰索引值获取对应位移数组中的位移值；</p>
3	厚度（透明）	<p>作用：测量透明物体的厚度，或者透明物体之间的间隙厚度。</p> <p>处理方法：</p> <p>1、根据选择的波峰索引值 N 和 M，换算成有效波峰的索引值 N'和 M'；</p> <p>2、分别根据有效波峰索引值 N'和 M'，获取对应位移数组中的位移值 Sn 和 Sm；</p> <p>3、将 Sn 和 Sm 相减后求绝对值；</p>
4	厚度（两测头）	<p>作用：测量两个感测头对射中间的测量物的厚度。</p> <p>处理方法：</p> <p>1、取得两个感测头 Near 端最近的有效波峰索引值 N1 和 N2；</p> <p>2、根据有效波峰索引值 N1 和 N2 获取对应位移数组中的位移值 S1 和 S2；</p> <p>3、计算 S1 和 S2 的和；</p>
5	高度差（两测头）	<p>作用：测量两个感测头对应测量物的高度差</p> <p>处理方法：</p> <p>1、取得两个感测头 Near 端最近的有效波峰索引值 N1 和 N2；</p> <p>2、根据有效波峰索引值 N1 和 N2 获取对应位移数组中的位移值 S1 和 S2；</p> <p>3、计算 S1 减去 S2 的值；</p>
6	运算	<p>作用：对多个测量方法结果值进行计算得到得结果为运算方法的结果值；</p> <p>计算类型处理方法有如下几种：</p> <p>1、根据公式 $A * OutA + B * OutB + C$ 计算结果，其中 OutA 和 OutB 为测量结果值；</p> <p>2、Ave 计算，将所有选中的 Out 结果值相加，再除以选中 Out 的个数；</p> <p>3、Max 计算，将所有选中的 Out 结果值排序，选中最大的那个测量结果值输出；</p> <p>4、Min 计算，将所有选中的 Out 结果值排序，选中最小的那个测量结果值输出；</p> <p>5、Max - Min 计算，将所有选中的 Out 结果值排序，最大的减去最小的结果值输出；</p> <p>6、所有的 Out 测量结果值若出现无效值，则此次计算结果值为无效值；</p>

2.4.1. 测量方法设置参数

```
typedef struct BS_OutConfig
{
    //滤波类型: 0 为移动平均
    int filterType;
    //移动平均次数: 1, 2, 4, 8, 16, 64, 256, 1024
    int filterParam;

    //保持模式(内部时钟保持), 0 为常规(无保持), 1 为峰值保持, 2 为谷值保持, 3 为峰谷保持, 4 为平均值保持, 5 为采样保持
    int holdMode;
    //采样个数(1 - 9999)
    int sampleNum;

    //无效数据保持次数(遇到无效值时, 用最近的一次有效数据进行替换的次数), 默认值为 8
    int invalidNum;
    //有效数据恢复次数(数据从有效数据变成有效数据时, 必须满足设定的次数后才恢复成有效数据)
    int recoveryNum;

    //结果数据显示单位: 0 为 um, 1 为 mm
    int resultUnit;
    //结果数据小数位数
    int resultPrecision;

    //缩放系数 (直线方程  $y=kx+b$  中的  $k$ )
    double scaleFactor;
    //缩放值(单位:um) (直线方程  $y=kx+b$  中的  $b$ )
    double scaleOffset;

    //归零状态(0:不归零, 1: 归零)
    int zeroStatus;
    //归零值(单位:um)
    double zeroOffset;

    //偏移值(单位:um)
    double offset;

    //第一个波峰的索引值 (透明方法时生效)
    int targetPeak1;
    //第二个波峰的索引值 (透明方法时生效)
```

```

int targetPeak2;

//上公差(单位:um)
double upperTolerance;
//下公差(单位:um)
double lowerTolerance;

//当前测量方法中关联的感测头索引值(0 - 3), -1 为无效值
int headFirst;
int headSecond;

//公式计算类型中的系数值(A * OutA + B * OutB + C), 公式计算类型中生效
double calcFactorA;
double calcFactorB;
double calcFactorC;

//计算测量方法中, 关联的 Out 方法索引值(0 - 7)列表, -1 为无效值, 列表最大为 8 项;
//公式计算类型(A * OutA + B * OutB + C)时, 列表中的第一项为 OutA 索引, 第二项为 OutB
索引
int outCalcArray[8];

//计算测量方法中的计算类型, 测量方法中生效; 0 为公式类型, 1 为平均类型, 2 为谷
峰类型, 3 为最大值类型, 4 为最小值类型
int calcType;

//测量方法类型, 0 为位移, 1 为透明位移, 2 为透明厚度, 3 为双感测头厚度, 4 为双
感测头高度差, 5 为计算
int outType;
}BS_OutConfig;

```

2.4.2. 位移

位移测量时 BS_OutConfig 参数使用说明如下:

名称	类型	说明	使用
滤波类型	Int	滤波类型值(0 未移动平均), 用于测量数据的滤波运算	是
移动平均次数	Int	移动平均次数(1,2,4,8,16,64,256,1024), 用于测量数据的滤波运算	是
保持模式	Int	保持模式值(0 为常规保持, 1 为峰值保持, 2 为谷值保持, 3 为峰谷保持, 4 为平均值保持, 5 为采样保持), 用于测量数	是

		据的保持处理运算	
保持采样个数	Int	保持处理时，一个保持周期内的采样个数，用于测量数据的保持处理运算	是
无效保持次数	Int	测量数据由有效数据转为无效数据时，将无效数据保持为最近一个有效数据的个数，用于测量数据的无效数据保持处理运算	是
有效恢复次数	Int	测量数据由无效数据转为有效数据时，将有效数据保持为无效数据的个数，用于测量书的有效恢复保持处理运算	是
测量数据单位值	Int	测量数据的单位(0 为 um, 1 为 mm)，用于公差判断时，要先将测量值按单位和精度进行四舍五入后，再与公差进行对比	是
测量数据精度值	Int	测量数据小数位数精度值，用于公差判断时，要先将测量值按单位和精度进行四舍五入后，再与公差进行对比	是
缩放系数	Double	测量数据的缩放系数 $k(y=kx+b)$ ，用于测量数据的缩放处理运算	是
缩放偏移值	Double	测量数据的缩放偏移值 $b(y=kx+b)$ ，用于测量数据的缩放处理运算	是
归零状态值	Int	测量数据的归零状态(0 为不归零, 1 为归零)，用于测量数据的归零处理运算	是
归零值	Double	测量数据的归零值，用于测量数据的归零处理运算	是
偏移值	Double	测量数据的偏移值，用于测量数据的偏移处理运算	是
第一个波峰索引值	Int	测量方法中选择的第一个波峰位置索引值(+1 ~ +4, -4 ~ -1)，用于透明方法时，选择波峰对应的测量位移值	是
第二个波峰索引值	Int	测量方法中选择的第二个波峰位置索引值(+1 ~ +4, -4 ~ -1)，用于透明方法时，选择波峰对应的测量位移值	否
上公差	Double	上公差值，用于测量数据的公差判断运算	是
下公差	Double	下公差值，用于测量数据的公差判断运算	是
第一个感测头索引	Int	双感测头方法中，选择的第一个感测头的索引(0 ~ 3)，用于获取对应感测头的位移值	是

第二个感测头索引	Int	双感测头方法中，选择的第二个感测头的索引(0~3)，用于获取对应感测头的位移值	否
公式运算系数 A	Double	公式计算类型中系数 A ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	否
公式运算系数 B	Double	公式计算类型中系数 B ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	否
公式运算系数 C	Double	公式计算类型中系数 C ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	否
计算类型依赖测量项	Array	计算测量方法中，关联的测量方法索引值(0~7)列表； 公式运算中，列表第一项为 OutA 的索引，第二项为 OutB 的索引； 多项式运算中，列表中为所有选中的 Out 依赖项； 用于运算测量方法的计算	否
计算方法类型	Int	计算测量方法中的计算类型(计算方法中生效)， 0 为公式类型；1 为平均类型；2 为孤峰类型； 3 为最大值类型；4 为最小值类型； 用于运算测量方法的计算	否
测量方法类型	Int	0 为普通位移，1 为透明位移，2 为透明厚度，3 为双感测头厚度， 4 为双感测头高度差，5 为运算	是
测量方法索引值	Int	当前测量方法的索引值 (0~7)，与 Out01 ~ Out08 对应，用于标记当前测量方法的序号	是

2.4.3. 位移（透明）

透明位移测量时 BS_OutConfig 参数使用说明如下：

名称	类型	说明	使用
滤波类型	Int	滤波类型值(0 未移动平均)，用于测量数据的滤波运算	是
移动平均次数	Int	移动平均次数(1,2,4,8,16,64,256,1024)，用于测量数据的滤波	是

		运算	
保持模式	Int	保持模式值(0 为常规保持, 1 为峰值保持, 2 为谷值保持, 3 为峰谷保持, 4 为平均值保持, 5 为采样保持), 用于测量数据的保持处理运算	是
保持采样个数	Int	保持处理时, 一个保持周期内的采样个数, 用于测量数据的保持处理运算	是
无效保持次数	Int	测量数据由有效数据转为无效数据时, 将无效数据保持为最近一个有效数据的个数, 用于测量数据的无效数据保持处理运算	是
有效恢复次数	Int	测量数据由无效数据转为有效数据时, 将有效数据保持为无效数据的个数, 用于测量书的有效恢复保持处理运算	是
测量数据单位值	Int	测量数据的单位(0 为 um, 1 为 mm), 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
测量数据精度值	Int	测量数据小数位数精度值, 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
缩放系数	Double	测量数据的缩放系数 $k(y=kx+b)$, 用于测量数据的缩放处理运算	是
缩放偏移值	Double	测量数据的缩放偏移值 $b(y=kx+b)$, 用于测量数据的缩放处理运算	是
归零状态值	Int	测量数据的归零状态(0 为不归零, 1 为归零), 用于测量数据的归零处理运算	是
归零值	Double	测量数据的归零值, 用于测量数据的归零处理运算	是
偏移值	Double	测量数据的偏移值, 用于测量数据的偏移处理运算	是
第一个波峰索引值	Int	测量方法中选择的第一个波峰位置索引值(+1 ~ +4, -4 ~ -1), 用于透明方法时, 选择波峰对应的测量位移值	是
第二个波峰索引值	Int	测量方法中选择的第二个波峰位置索引值(+1 ~ +4, -4 ~ -1), 用于透明方法时, 选择波峰对应的测量位移值	否
上公差	Double	上公差值, 用于测量数据的公差判断运算	是

下公差	Double	下公差值，用于测量数据的公差判断运算	是
第一个感测头索引	Int	双感测头方法中，选择的第一个感测头的索引(0~3)，用于获取对应感测头的位移值	是
第二个感测头索引	Int	双感测头方法中，选择的第二个感测头的索引(0~3)，用于获取对应感测头的位移值	否
公式运算系数 A	Double	公式计算类型中系数 A ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	否
公式运算系数 B	Double	公式计算类型中系数 B ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	否
公式运算系数 C	Double	公式计算类型中系数 C ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	否
计算类型依赖测量项	Array	计算测量方法中，关联的测量方法索引值(0~7)列表； 公式运算中，列表第一项为 OutA 的索引，第二项为 OutB 的索引； 多项式运算中，列表中为所有选中的 Out 依赖项； 用于运算测量方法的计算	否
计算方法类型	Int	计算测量方法中的计算类型(计算方法中生效)， 0 为公式类型；1 为平均类型；2 为孤峰类型； 3 为最大值类型；4 为最小值类型； 用于运算测量方法的计算	否
测量方法类型	Int	0 为普通位移，1 为透明位移，2 为透明厚度，3 为双感测头厚度， 4 为双感测头高度差，5 为运算	是
测量方法索引值	Int	当前测量方法的索引值 (0~7)，与 Out01 ~ Out08 对应，用于标记当前测量方法的序号	是

2.4.4. 厚度（透明）

透明厚度测量时 BS_OutConfig 参数使用说明如下：

名称	类型	说明	使用
滤波类型	Int	滤波类型值(0 未移动平均), 用于测量数据的滤波运算	是
移动平均次数	Int	移动平均次数(1,2,4,8,16,64,256,1024), 用于测量数据的滤波运算	是
保持模式	Int	保持模式值(0 为常规保持, 1 为峰值保持, 2 为谷值保持, 3 为峰谷保持, 4 为平均值保持, 5 为采样保持), 用于测量数据的保持处理运算	是
保持采样个数	Int	保持处理时, 一个保持周期内的采样个数, 用于测量数据的保持处理运算	是
无效保持次数	Int	测量数据由有效数据转为无效数据时, 将无效数据保持为最近一个有效数据的个数, 用于测量数据的无效数据保持处理运算	是
有效恢复次数	Int	测量数据由无效数据转为有效数据时, 将有效数据保持为无效数据的个数, 用于测量书的有效恢复保持处理运算	是
测量数据单位值	Int	测量数据的单位(0 为 um, 1 为 mm), 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
测量数据精度值	Int	测量数据小数位数精度值, 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
缩放系数	Double	测量数据的缩放系数 $k(y=kx+b)$, 用于测量数据的缩放处理运算	是
缩放偏移值	Double	测量数据的缩放偏移值 $b(y=kx+b)$, 用于测量数据的缩放处理运算	是
归零状态值	Int	测量数据的归零状态(0 为不归零, 1 为归零), 用于测量数据的归零处理运算	是
归零值	Double	测量数据的归零值, 用于测量数据的归零处理运算	是
偏移值	Double	测量数据的偏移值, 用于测量数据的偏移处理运算	是
第一个波峰索引值	Int	测量方法中选择的第一个波峰位置索引值(+1 ~ +4, -4 ~ -1), 用于透明方法时, 选择波峰对应的测量位移值	是

第二个波峰索引值	Int	测量方法中选择的第二个波峰位置索引值(+1 ~ +4, -4 ~ -1), 用于透明方法时, 选择波峰对应的测量位移值	是
上公差	Double	上公差值, 用于测量数据的公差判断运算	是
下公差	Double	下公差值, 用于测量数据的公差判断运算	是
第一个感测头索引	Int	双感测头方法中, 选择的第一个感测头的索引(0 ~ 3), 用于 获取对应感测头的位移值	是
第二个感测头索引	Int	双感测头方法中, 选择的第二个感测头的索引(0 ~ 3), 用于 获取对应感测头的位移值	否
公式运算系数 A	Double	公式计算类型中系数 A ($A * OutA + B * OutB + C$), 用于运算 方法中的公式类型计算	否
公式运算系数 B	Double	公式计算类型中系数 B ($A * OutA + B * OutB + C$), 用于运算 方法中的公式类型计算	否
公式运算系数 C	Double	公式计算类型中系数 C ($A * OutA + B * OutB + C$), 用于运算 方法中的公式类型计算	否
计算类型依赖测量 项	Array	计算测量方法中, 关联的测量方法索引值(0 ~ 7)列表; 公式运算中, 列表第一项为 OutA 的索引, 第二项为 OutB 的 索引; 多项式运算中, 列表中为所有选中的 Out 依赖项; 用于运算测量方法的计算	否
计算方法类型	Int	计算测量方法中的计算类型(计算方法中生效), 0 为公式类型; 1 为平均类型; 2 为孤峰类型; 3 为最大值类型; 4 为最小值类型; 用于运算测量方法的计算	否
测量方法类型	Int	0 为普通位移, 1 为透明位移, 2 为透明厚度, 3 为双感测头 厚度, 4 为双感测头高度差, 5 为运算	是
测量方法索引值	Int	当前测量方法的索引值 (0 ~ 7), 与 Out01 ~ Out08 对应, 用 于标记当前测量方法的序号	是

2.4.5. 厚度（两测头）

两测头厚度测量时各测头 BS_OutConfig 参数使用说明如下：

名称	类型	说明	使用
滤波类型	Int	滤波类型值(0 未移动平均)，用于测量数据的滤波运算	是
移动平均次数	Int	移动平均次数(1,2,4,8,16,64,256,1024)，用于测量数据的滤波运算	是
保持模式	Int	保持模式值(0 为常规保持，1 为峰值保持，2 为谷值保持，3 为峰谷保持，4 为平均值保持，5 为采样保持)，用于测量数据的保持处理运算	是
保持采样个数	Int	保持处理时，一个保持周期内的采样个数，用于测量数据的保持处理运算	是
无效保持次数	Int	测量数据由有效数据转为无效数据时，将无效数据保持为最近一个有效数据的个数，用于测量数据的无效数据保持处理运算	是
有效恢复次数	Int	测量数据由无效数据转为有效数据时，将有效数据保持为无效数据的个数，用于测量书的有效恢复保持处理运算	是
测量数据单位值	Int	测量数据的单位(0 为 um，1 为 mm)，用于公差判断时，要先将测量值按单位和精度进行四舍五入后，再与公差进行对比	是
测量数据精度值	Int	测量数据小数位数精度值，用于公差判断时，要先将测量值按单位和精度进行四舍五入后，再与公差进行对比	是
缩放系数	Double	测量数据的缩放系数 $k(y=kx+b)$ ，用于测量数据的缩放处理运算	是
缩放偏移值	Double	测量数据的缩放偏移值 $b(y=kx+b)$ ，用于测量数据的缩放处理运算	是
归零状态值	Int	测量数据的归零状态(0 为不归零，1 为归零)，用于测量数据的归零处理运算	是
归零值	Double	测量数据的归零值，用于测量数据的归零处理运算	是

偏移值	Double	测量数据的偏移值，用于测量数据的偏移处理运算	是
第一个波峰索引值	Int	测量方法中选择的第一个波峰位置索引值(+1 ~ +4, -4 ~ -1)， 用于透明方法时，选择波峰对应的测量位移值	是(值为 0)
第二个波峰索引值	Int	测量方法中选择的第二个波峰位置索引值(+1 ~ +4, -4 ~ -1)， 用于透明方法时，选择波峰对应的测量位移值	是(值为 0)
上公差	Double	上公差值，用于测量数据的公差判断运算	是
下公差	Double	下公差值，用于测量数据的公差判断运算	是
第一个感测头索引	Int	双感测头方法中，选择的第一个感测头的索引(0 ~ 3)，用于 获取对应感测头的位移值	是
第二个感测头索引	Int	双感测头方法中，选择的第二个感测头的索引(0 ~ 3)，用于 获取对应感测头的位移值	否
公式运算系数 A	Double	公式计算类型中系数 A ($A * OutA + B * OutB + C$)，用于运算 方法中的公式类型计算	是
公式运算系数 B	Double	公式计算类型中系数 B ($A * OutA + B * OutB + C$)，用于运算 方法中的公式类型计算	否
公式运算系数 C	Double	公式计算类型中系数 C ($A * OutA + B * OutB + C$)，用于运算 方法中的公式类型计算	否
计算类型依赖测量项	Array	计算测量方法中，关联的测量方法索引值(0 ~ 7)列表； 公式运算中，列表第一项为 OutA 的索引，第二项为 OutB 的 索引； 多项式运算中，列表中为所有选中的 Out 依赖项； 用于运算测量方法的计算	否
计算方法类型	Int	计算测量方法中的计算类型(计算方法中生效)， 0 为公式类型；1 为平均类型；2 为孤峰类型； 3 为最大值类型；4 为最小值类型； 用于运算测量方法的计算	否
测量方法类型	Int	0 为普通位移，1 为透明位移，2 为透明厚度，3 为双感测头 厚度， 4 为双感测头高度差，5 为运算	是

测量方法索引值	Int	当前测量方法的索引值(0~7), 与 Out01 ~ Out08 对应, 用于标记当前测量方法的序号	是
---------	-----	--	---

2.4.6. 高度差（两测头）

两测头高度差测量时各测头 BS_OutConfig 参数使用说明如下：

名称	类型	说明	使用
滤波类型	Int	滤波类型值(0 未移动平均), 用于测量数据的滤波运算	是
移动平均次数	Int	移动平均次数(1,2,4,8,16,64,256,1024), 用于测量数据的滤波运算	是
保持模式	Int	保持模式值(0 为常规保持, 1 为峰值保持, 2 为谷值保持, 3 为峰谷保持, 4 为平均值保持, 5 为采样保持), 用于测量数据的保持处理运算	是
保持采样个数	Int	保持处理时, 一个保持周期内的采样个数, 用于测量数据的保持处理运算	是
无效保持次数	Int	测量数据由有效数据转为无效数据时, 将无效数据保持为最近一个有效数据的个数, 用于测量数据的无效数据保持处理运算	是
有效恢复次数	Int	测量数据由无效数据转为有效数据时, 将有效数据保持为无效数据的个数, 用于测量书的有效恢复保持处理运算	是
测量数据单位值	Int	测量数据的单位(0 为 um, 1 为 mm), 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
测量数据精度值	Int	测量数据小数位数精度值, 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
缩放系数	Double	测量数据的缩放系数 $k(y=kx+b)$, 用于测量数据的缩放处理运算	是
缩放偏移值	Double	测量数据的缩放偏移值 $b(y=kx+b)$, 用于测量数据的缩放处理运算	是

归零状态值	Int	测量数据的归零状态(0 为不归零, 1 为归零), 用于测量数据的归零处理运算	是
归零值	Double	测量数据的归零值, 用于测量数据的归零处理运算	是
偏移值	Double	测量数据的偏移值, 用于测量数据的偏移处理运算	是
第一个波峰索引值	Int	测量方法中选择的第一个波峰位置索引值(+1 ~ +4, -4 ~ -1), 用于透明方法时, 选择波峰对应的测量位移值	是(值为 0)
第二个波峰索引值	Int	测量方法中选择的第二个波峰位置索引值(+1 ~ +4, -4 ~ -1), 用于透明方法时, 选择波峰对应的测量位移值	是(值为 0)
上公差	Double	上公差值, 用于测量数据的公差判断运算	是
下公差	Double	下公差值, 用于测量数据的公差判断运算	是
第一个感测头索引	Int	双感测头方法中, 选择的第一个感测头的索引(0 ~ 3), 用于获取对应感测头的位移值	是
第二个感测头索引	Int	双感测头方法中, 选择的第二个感测头的索引(0 ~ 3), 用于获取对应感测头的位移值	是
公式运算系数 A	Double	公式计算类型中系数 A ($A * OutA + B * OutB + C$), 用于运算方法中的公式类型计算	否
公式运算系数 B	Double	公式计算类型中系数 B ($A * OutA + B * OutB + C$), 用于运算方法中的公式类型计算	否
公式运算系数 C	Double	公式计算类型中系数 C ($A * OutA + B * OutB + C$), 用于运算方法中的公式类型计算	否
计算类型依赖测量项	Array	计算测量方法中, 关联的测量方法索引值(0 ~ 7)列表; 公式运算中, 列表第一项为 OutA 的索引, 第二项为 OutB 的索引; 多项式运算中, 列表中为所有选中的 Out 依赖项; 用于运算测量方法的计算	否
计算方法类型	Int	计算测量方法中的计算类型(计算方法中生效), 0 为公式类型; 1 为平均类型; 2 为孤峰类型; 3 为最大值类型; 4 为最小值类型; 用于运算测量方法的计算	否

测量方法类型	Int	0 为普通位移, 1 为透明位移, 2 为透明厚度, 3 为双感测头厚度, 4 为双感测头高度差, 5 为运算	是
测量方法索引值	Int	当前测量方法的索引值 (0~7), 与 Out01 ~ Out08 对应, 用于标记当前测量方法的序号	是

2.4.7. 运算

运算方法 BS_OutConfig 参数使用说明如下:

名称	类型	说明	使用
滤波类型	Int	滤波类型值(0 未移动平均), 用于测量数据的滤波运算	是
移动平均次数	Int	移动平均次数(1,2,4,8,16,64,256,1024), 用于测量数据的滤波运算	是
保持模式	Int	保持模式值(0 为常规保持, 1 为峰值保持, 2 为谷值保持, 3 为峰谷保持, 4 为平均值保持, 5 为采样保持), 用于测量数据的保持处理运算	是
保持采样个数	Int	保持处理时, 一个保持周期内的采样个数, 用于测量数据的保持处理运算	是
无效保持次数	Int	测量数据由有效数据转为无效数据时, 将无效数据保持为最近一个有效数据的个数, 用于测量数据的无效数据保持处理运算	是
有效恢复次数	Int	测量数据由无效数据转为有效数据时, 将有效数据保持为无效数据的个数, 用于测量书的有效恢复保持处理运算	是
测量数据单位值	Int	测量数据的单位(0 为 um, 1 为 mm), 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
测量数据精度值	Int	测量数据小数位数精度值, 用于公差判断时, 要先将测量值按单位和精度进行四舍五入后, 再与公差进行对比	是
缩放系数	Double	测量数据的缩放系数 $k(y=kx+b)$, 用于测量数据的缩放处理运	是

		算	
缩放偏移值	Double	测量数据的缩放偏移值 $b(y=kx+b)$ ，用于测量数据的缩放处理运算	是
归零状态值	Int	测量数据的归零状态(0 为不归零，1 为归零)，用于测量数据的归零处理运算	是
归零值	Double	测量数据的归零值，用于测量数据的归零处理运算	是
偏移值	Double	测量数据的偏移值，用于测量数据的偏移处理运算	是
第一个波峰索引值	Int	测量方法中选择的第一个波峰位置索引值(+1 ~ +4, -4 ~ -1)，用于透明方法时，选择波峰对应的测量位移值	否
第二个波峰索引值	Int	测量方法中选择的第二个波峰位置索引值(+1 ~ +4, -4 ~ -1)，用于透明方法时，选择波峰对应的测量位移值	否
上公差	Double	上公差值，用于测量数据的公差判断运算	是
下公差	Double	下公差值，用于测量数据的公差判断运算	是
第一个感测头索引	Int	双感测头方法中，选择的第一个感测头的索引(0 ~ 3)，用于获取对应感测头的位移值	否
第二个感测头索引	Int	双感测头方法中，选择的第二个感测头的索引(0 ~ 3)，用于获取对应感测头的位移值	否
公式运算系数 A	Double	公式计算类型中系数 A ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	是
公式运算系数 B	Double	公式计算类型中系数 B ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	是
公式运算系数 C	Double	公式计算类型中系数 C ($A * OutA + B * OutB + C$)，用于运算方法中的公式类型计算	是
计算类型依赖测量项	Array	计算测量方法中，关联的测量方法索引值(0 ~ 7)列表； 公式运算中，列表第一项为 OutA 的索引，第二项为 OutB 的索引； 多项式运算中，列表中为所有选中的 Out 依赖项； 用于运算测量方法的计算	是
计算方法类型	Int	计算测量方法中的计算类型(计算方法中生效)，	是

		0 为公式类型；1 为平均类型；2 为孤峰类型； 3 为最大值类型；4 为最小值类型； 用于运算测量方法的计算	
测量方法类型	Int	0 为普通位移，1 为透明位移，2 为透明厚度，3 为双感测头厚度， 4 为双感测头高度差，5 为运算	是
测量方法索引值	Int	当前测量方法的索引值（0~7），与 Out01 ~ Out08 对应，用于标记当前测量方法的序号	是

第三章 函数列表

3.1. 连接控制器

3.1.1. void GetSDKVersion(wchar_t* pSDKVersion, int len)

函数功能: 获取当前 SDK 的版本号

参数:

[out]pSDKVersion: 返回获取到的 SDK 的版本号

[in]len: pSDKVersion 数组的长度，长度不小于 32

返回值:无

3.1.2. int BS_SetDeviceSelectType(int devType)

函数功能: 设置所使用的控制器的类型值，该函数必须在调用其他接口之前调用。

参数:

[in]devType: 选择控制器设备的类型, 0-USB 3.0 接口 1 代, 1-以太网接口, 2-USB 3.0 接口 2 代; 3-嵌入式控制器

返回值: 成功返回 1，其他返回值参照错误代码定义表。

3. 1. 3. **int BS_GetDeviceSelectType()**

函数功能: 获取当前所使用的控制器类型, 本函数可以在任意位置调用。

参数:

返回值: 0-USB 3.0 接口 1 代, 1-以太网接口, 2-USB 3.0 接口 2 代; 3-嵌入式控制器

3. 1. 4. **void BS_SetSDKLogType(int type = 0)**

函数功能: 设置 SDK 日志的打印模式。

参数: 0 为关闭, 1 为打印错误日志, 2 为打印所有日志

返回值: 无

3. 1. 5. **int BS_InitSDK()**

函数功能: 初始化 SDK, 函数调用前确认插入匹配的加密狗; 初始化后需要在关闭程序前调用 BS_UnInitSDK 函数释放相关资源。

参数: 无

返回值:

-27: SDK 已经完成了初始化;

-29: 未找到匹配的加密狗;

-30: USB 一代控制器初始化失败, 请检查是否已安装 USB 一代最新版驱动;

-31: 以太网控制器初始化失败, 请检查是否已安装以太网版本驱动;

-32: USB 二代控制器初始化失败, 请检查是否已安装 USB 二代最新版驱动;

其他返回值请参考返回值定义表。

3. 1. 6. **int BS_UnInitSDK()**

函数功能: 反初始化 SDK。

参数: 无

返回值:

1: 成功

0: 失败

3.1.7. int BS_GetControllerNum()

函数功能：获取可连接的控制器个数，调用时确认没有控制器处于开启状态。

参数：无

返回值：

获取到的控制器个数；

失败则返回-10

3.1.8. int BS_GetControllerSN(wchar_t* pSNArray, int num)

函数功能：获取可连接的控制器的序列号列表。

参数：

[out]pSNArray: 保存所有控制器序列号的数据，每个序列号固定占用 32 个字符，数组长度需要大于等于 $32 * num$

[in]num: 需要获取控制器的个数，最大值为 10

返回值：

成功返回 1

失败返回 0

3.1.9. int BS_CreateControllerObj(wchar_t* pStrSN)

函数功能：创建控制器对象

参数：

[in]pStrPN: 控制器的序列号，字符串长度为 32

返回值：

成功返回 1

其他返回值请参考返回值定义表。

3. 1. 10. **int BS_RemoveControllerObj(wchar_t* pStrSN)**

函数功能：删除控制器对象

参数：

[in]pStrPN：控制器的序列号，字符串长度为 32

返回值：

成功返回 1

其他返回值请参考返回值定义表。

3. 2. 配置控制器参数

3. 2. 1. **int BS_LoadControllerConfFromFile(wchar_t* pStrSN, wchar_t* filePath)**

函数功能：通过加载配置文件，配置并启动控制器的所有感测头。

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]filePath：配置文件的路径

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 2. **int BS_LoadLWCalibrationDataFromFile(wchar_t* pStrSN, wchar_t* filePath)**

函数功能：通过加载配置文件，指定控制器的所有感测头的高度校准数据。

参数：

[in]pStrSN：校准器的序列号，字符串长度为 32

[in]filePath：叫转文件的路径

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 3. **int BS_SetSampleFrequency(wchar_t* pStrSN, int sample)**

函数功能：设置指定控制器的采样频率

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]sample：采样频率值

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 4. **int BS_GetSampleFrequency(wchar_t* pStrSN, int* pSample)**

函数功能：获取指定控制器的采样频率

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]sample：采样频率值

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 5. **int BS_GetCurrSampleFrequency(wchar_t* pStrSN, int* pCurrSample)**

函数功能：获取指定控制器当前实际的采样频率

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pCurrSample：实际的采样频率值

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 6. int BS_GetMaxSampleFrequency(wchar_t* pStrSN, int* pMaxSample)

函数功能：获取指定控制器最大能达到的采样频率

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pMaxSample：最大的采样频率值

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 7. int BS_GetGainRange(wchar_t* pStrSN, float* pMinGain, float* pMaxGain)

函数功能：获取指定控制器的增益范围

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pMinGain：最小的增益值

[out]pMaxGain：最大的增益值

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 8. int BS_SetGainValue(wchar_t* pStrSN, float gainValue)

函数功能：设置指定控制器的增益值

参数：

[in]pStrSN: 控制器的序列号, 字符串长度为 32

[int]gainValue: 增益值

返回值:

成功返回 1

其他返回值请参考返回值定义表

3. 2. 9. int BS_GetGainValue(wchar_t* pStrSN, float* pGainValue)

函数功能: 取得指定控制器的增益值

参数:

[in]pStrSN: 控制器的序列号, 字符串长度为 32

[out]pGainValue: 增益值

返回值:

成功返回 1

其他返回值请参考返回值定义表

3. 2. 10. int BS_GetExposureTimeRange(wchar_t* pStrSN, float* pMinTime, float* pMaxTime)

函数功能: 获取指定控制器的曝光时间范围

参数:

[in]pStrSN: 控制器的序列号, 字符串长度为 32

[out]pMinTime: 最小的曝光时间

[out]pMaxTime: 最大的曝光时间

返回值:

成功返回 1

其他返回值请参考返回值定义表

3. 2. 11. **int BS_SetExposureTimeValue(wchar_t* pStrSN, float timeValue)**

函数功能：设置指定控制器的曝光时间范围

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[int]timeValue：曝光时间

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 12. **int BS_GetExposureTimeValue(wchar_t* pStrSN, float* pTimeValue)**

函数功能：取得指定控制器的曝光时间

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pTimeValue：曝光时间

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 13. **int BS_SetTriggerMode(wchar_t* pStrSN, int triggerMode)**

函数功能：设置控制器触发模式

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]triggerMode：触发模式，0 表示连续获取模式，2 表示硬件触发模式

返回值：

成功返回 1，

其他返回值请参考返回值定义表

3. 2. 14. **int BS_GetTriggerMode(wchar_t* pStrSN, int* pTriggerMode)**

函数功能：获取控制器触发模式。

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pTriggerMode：触发模式，0 表示连续获取模式，2 表示硬件触发模式

返回值：

成功返回 1，

其他返回值请参考返回值定义表

3. 2. 15. **int BS_SetTriggerSourceType(wchar_t* pStrSN, int triggerType)**

函数功能：设置外触发源类型

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]triggerType：外触发源类型，0-编码器触发；1-外部单端 24V 信号

返回值：

设置成功返回 1，

其他返回值请参考返回值定义表。

3. 2. 16. **int BS_GetTriggerSourceType(wchar_t* pStrSN, int* pTriggerType)**

函数功能：获取外触发源类型

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pTriggerType：外触发源类型，0-编码器触发；1-外部单端 24V 信号

返回值:

设置成功返回 1,

其他返回值请参考返回值定义表。

3. 2. 17. int BS_SetEncoderFrequencyParam(wchar_t* pStrSN, unsigned char sta, unsigned char axis, unsigned char dir, int step)

函数功能: 设置编码器分频触发参数

参数:

[in]pStrSN: 控制器的序列号, 字符串长度为 32

[in]sta: 0-禁用触发信号输出, 1-启用触发信号输出

[in]axis: 轴号, 可设范围 0~5

[in]dir: 触发方向, 0: 正向; 1: 负向; 2-双向

[in]step: 分频间隔, 单位 pulse

返回值: 1-操作成功, 其他返回值请参考返回值定义表。

3. 2. 18. int BS_GetEncoderFrequencyParam(wchar_t* pStrSN, unsigned char* pSta, unsigned char* pAxis, unsigned char* pDir, int* pStep)

函数功能: 获取编码器分频触发参数

参数:

[in]pStrSN: 控制器的序列号, 字符串长度为 32

[out]pSta: 0-禁用触发信号输出, 1-启用触发信号输出

[out]pAxis: 轴号, 可设范围 0~5

[out]pDir: 触发方向, 0: 正向; 1: 负向; 2-双向

[out]pStep: 分频间隔, 单位 pulse

返回值:

1-操作成功,

其他返回值请参考返回值定义表。

3. 2. 19. int BS_DarkController(wchar_t* pStrSN)

函数功能：采集控制器已启用感测头的 dark 数据(采集前请先用黑色盖子盖住对应通道探头，直到采集完成)

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 2. 20. int BS_GetDarkRate(wchar_t* pStrSN, int* pDarkRate)

函数功能：获取控制器通道当前 Dark 完成进度，单位为%

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pDarkRate：Dark 进度比例，单位为%

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 3. 配置探头

3. 3. 1. int BS_GetHeadNum(wchar_t* pStrSN, int* pHeadNum)

函数功能：获取指定控制器的感测头数量

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pHeadNum：感测头的数量

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 3. 2. **int BS_EnableHead(wchar_t* pStrSN, int iHead, bool bEnable)**

函数功能：开启或关闭指定控制器的感测头

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iHead：感测头索引值（1 - 4）

[in]bEnable：true 表示启用通道，false 表示禁用

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3. 3. 3. **int BS_InitHeadConfig(wchar_t* pStrSN, int iHead)**

函数功能：初始化指定感测头的配置参数

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iHead：感测头索引值（1 - 4）

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3. 3. 4. **int BS_GetHeadDistRange(wchar_t* pStrSN, int iHead, float* pTopRange, float* pBottomRange)**

函数功能：获取指定感测头的量程，单位 um

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iHead: 感测头索引值 (0 - 3)

[out]pTopRange: 量程上边界

[out]pBottomRange: 量程下边界

返回值:

成功返回 1

其他返回值请参考返回值定义表

3. 3. 5. int BS_SetControllerMode(wchar_t* pStrSN, int iMode)

函数功能: 设置控制器模式, 包括测量模式和设定模式

参数:

[in]pStrSN: 控制器的序列号, 字符串长度为 32

[in]iMode: 对应的模式值, 0 为设定模式, 1 为测量模式

返回值:

成功返回 1

其他返回值请参考返回值定义表

3. 3. 6. int BS_SetHeadConfig(wchar_t* pStrSN, int iHead, const BS_HeadConfig* pHeadConfig)

函数功能: 设置感测头的参数

参数:

[in]pStrSN: 控制器的序列号, 字符串长度为 32

[in]iHead: 感测头索引值 (0 - 3)

[in]pHeadConfig: 感测头参数

返回值:

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3.3.7. int BS_GetHeadConfig(wchar_t* pStrSN, int iHead, BS_HeadConfig* pHeadConfig)

函数功能：获取感测头的参数

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iHead：感测头索引值（0-3）

[out]pHeadConfig：感测头参数对象

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3.4. 配置测量方法

3.4.1. int BS_SetOutConfig(wchar_t* pStrSN, int iOut, const BS_OutConfig* pOutConfig)

函数功能：设置测量方法参数

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iOut：Out 项索引值（0-7）

[in]pOutConfig：测量方法参数

返回值：

成功返回 1

返回 2 第一个感测头不存在；

返回 3 第二个感测头不存在

返回 4 第一个感测头共用于透明和非透明方法；

返回 5 第二个感测头共用于透明和非透明方法；

返回 6 第一个感测头共用于透明位移和透明厚度；

返回 12 计算参数依赖了自己；
返回 13 计算参数依赖了不存在的 Out 项；
返回 14 当前计算 Out 与其他计算 Out 相互依赖；
返回 15 计算参数依赖深度超过了两层；
其他返回值请参考返回值定义表

3. 4. 2. int BS_GetOutConfig(wchar_t* pStrSN, int iOut, BS_OutConfig* pOutConfig)

函数功能：获取测量方法参数

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iOut：Out 项索引值（0 - 7）

[out]pOutConfig：测量方法参数

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3. 4. 3. int BS_ClearOutConfig(wchar_t* pStrSN, int iOut)

函数功能：清除测量方法参数

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iOut：Out 项索引值（0 - 7），为-1 时清除所有方法的参数

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3. 4. 4. **int BS_SetOutZeroStatus(wchar_t* pStrSN, int iOut, int iStatus, double fValue)**

函数功能：设置 Out 的归零状态

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iOut：Out 项索引值（0 - 7）

[in]iStatus：归零状态值（0:不归零， 1:归零）

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3. 4. 5. **int BS_GetOutZeroStatus(wchar_t* pStrSN, int iOut, int* iStatus, double* fValue)**

函数功能：获取 Out 的归零状态

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iOut：Out 项索引值（0 - 7）

[out]iStatus：归零状态值（0:不归零， 1:归零）

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3.5. 测量

3.5.1. int BS_StartMeasure(wchar_t* pStrSN)

函数功能：开始测量，可以读取测量方法的测量值和感测头对应的数据

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

返回值：

成功返回 1

其他返回值请参考返回值定义表

3.5.2. int BS_StopMeasure(wchar_t* pStrSN)

函数功能：停止测量，不可以读取测量方法的测量值和感测头对应的数据

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

返回值：

成功返回 1

其他返回值请参考返回值定义表

3.5.3. int BS_GetMeasureStatus(wchar_t* pStrSN, int* pStatus)

函数功能：返回当前测量状态值

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[out]pStatus：1 表示正在测量，0 表示未开始测量

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 5. 4. **int BS_GetHeadData(wchar_t* pStrSN, int iHead, float* pIntensityArray, int* pWaveArray, float* pPeakArray, int* pPeakNum)**

函数功能：获取指定感测头的波形数据

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iHead：感测头索引值（0 - 3）

[out]pIntensityArray：感光度数组值(需要外部分配内存，长度为 5)

[out]pWaveArray：波形图数组(需要外部分配内存，长度要用 BS_GetHeadIntensityMsg 函数获取波形数据长度)

[out]pPeakArray：波峰位置数组(需要外部分配内存，长度为 10)

[out]pPeakNum：实际返回感光度和波峰数组的长度

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3. 5. 5. **int BS_GetHeadIntensityMsg(wchar_t* pStrSN, int iHead, int* pLength, int* pMaxValue)**

函数功能：获取指定感测头的单帧波形数据长度和最大的能量值。

参数：

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iHead：感测头索引值（0 - 3）

[out]pLength：单帧波形曲线数据数组的长度

[out]pMaxValue：波形曲线数据中波峰能达到的最大能量值

返回值：

成功返回 1

其他返回值请参考返回值定义表

3. 5. 6. int BS_GetOutData(wchar_t* pStrSN, int iOut, double* pDataArray, int* pJudgeArray, int* pDataNum)

函数功能：获取指定 Out 项的测量结果

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iOut：Out 项索引值（0 - 7）

[out]pDataArray：测量结果数组

[out]pJudgeArray：判定结果数组(0:LOW,1:GO,2:HI,3:无效)

[in,out]pDataNum：入参时为想要测量结果数组的长度，返回时为实际获取到的测量值数量

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表

3. 5. 7. int BS_GetOutOriginalData(wchar_t* pStrSN, int iOut, double* pDataArray, int* pDataNum)

函数功能：获取指定 Out 项的原始测量结果(只进行中值滤波后的测量结果)

[in]pStrSN：控制器的序列号，字符串长度为 32

[in]iOut：Out 项索引值（0 - 7）

[out]pDataArray：测量结果数组

[in,out]pDataNum：入参时为想要测量结果数组的长度，返回时为实际获取到的测量值数量

返回值：

成功返回 1

失败返回 0

其他返回值请参考返回值定义表