

Scale SDK-Windows-V2.012 接口文档

文档修改记录

一.使用说明

1.1 工程配置说明

1.2 函数声明

1.3 调用流程

二.接口说明

2.1 重量信息回调函数

2.2 设置回调函数

2.3 打开串口

2.4 关闭串口

2.5 获取缓存重量信息

2.7 去皮

2.8退出去皮

2.9 强制去皮

2.10 归零

文档修改记录

序号	版本号	修改内容	修改者	修改日期
01	v2.012	文档建立	庄永兴	2025-12-16

一.使用说明

1.1 工程配置说明

1.本SDK接口需运行在Windows系统上

2.把Sdk目录下的ScalesSDK.dll，放到应用同目录下或指定位置

名称	修改日期
 ScalesSDK.dll	2020/11/19 14:4
 ScalesSDKDemo.exe	2020/11/14 16:0

1.2 函数声明

1.声明库函数

```
1  sScalesLibName = 'ScalesSDK.dll';
2
3  // 回调函数原型
4  Procedure OnOPOModule(sModuleInfo: PAnsiChar; sLength: Integer); Stdca
   ll;
5
6  // 回调设置接口
7  procedure OPO_SetModuleCallBack(OnCallBack: Pointer); stdcall; externa
   l sScalesLibName;
8
9  // 打开通讯
10 function OPO_Open(sConnect: PAnsiChar): Integer; stdcall; external sSc
   alesLibName;
11
12 // 关闭
13 function OPO_Close: Integer; stdcall; external sScalesLibName;
14
15 // 读取缓存区重量
16 function OPO_ReadResultCache(sWeight: PAnsiChar): Integer; stdcall; ex
   ternal sScalesLibName;
17
18 // 归零
19 function OPO_Zero: Integer; stdcall; external sScalesLibName;
20
21 // 去皮
22 function OPO_Tare: Integer; stdcall; external sScalesLibName;
23
24 // 清皮
25 function OPO_ExitTare: Integer; stdcall; external sScalesLibName;
26
27 // 预制去皮命令
28 function OPO_PreTare(Value: integer): integer; stdcall; external sSca
   lesLibName;
29
```

1.3 调用流程

1. 实现回调：实现回调函数OnOPOModule
2. 初始化：调用OPO_SetModuleCallBack传入OnOPOModule，设置回调函数
3. 打开端口：调用OPO_Open 开启串口
4. 获取信息：通过回调函数获取重量信息

5. 关闭端口：调用OPO_Close 关闭串口

二.接口说明

2.1 重量信息回调函数

- 描述

▼

Pascal |

1 Procedure OnOPOModule(sModuleInfo: PAnsiChar; sLength: Integer);

- 参数

参数	描述
----	----

sModuleInfo	<ol style="list-style-type: none"> 1. 参数类型：指向一个ANSI 字符串的指针，指针指向的内存需要提前申请 2. 调用函数后，结果通过指针返回： <ul style="list-style-type: none"> • 返回数组格式： "sMode,sStatus,sZero,sUnit,sNetWeight,sTareWeight,sGrossWeight" • 返回数组举例： "N (sMode),S (sStatus),->0<- (sZero),kg (sUnit),0.000 (sNetWeight),0.000 (sTareWeight),0.000 (sGrossWeight)" • 字段说明： <ul style="list-style-type: none"> ○ sMode: <ul style="list-style-type: none"> ▪ sMode = "N" (净重称量) ▪ sMode = "T" (去皮称量) ▪ sMode = "P" (预制去皮称量) ○ sStatus: <ul style="list-style-type: none"> ▪ sStatus = "F" (重量溢出或没有开机归零) ▪ sStatus = "S" (重量稳定) ▪ sStatus = "U" (重量不稳定) ○ sZero: <ul style="list-style-type: none"> ▪ sZero="->0<-" (零点) ▪ sZero="" (非零点) ○ sUnit: (单位) ○ sNetWeight: (净重) ○ sTareWeight: (皮重) ○ sGrossWeight: (毛重)
sLength	返回数组长度

- 例子

▼Pascal |

```
1 Procedure OnOPOModule(sModuleInfo: PAnsiChar; sLength: Integer); Stdcall;  
2 begin  
3     frmMain.lblOPWeight.Caption := string(sModuleInfo);  
4 end;
```

2.2 设置回调函数

- 描述

▼Pascal |

```
1 procedure OP0_SetModuleCallBack(OnCallBack: Pointer);
```

- 参数

参数	描述
OnCallBack	OnOPOModule回调函数指针

- 例子

▼Pascal |

```
1 OP0_SetModuleCallBack(@OnOPOModule);
```

2.3 打开串口

- 描述

▼Pascal |

```
1 function OP0_Open(sConnect: PAnsiChar): Integer;
```

- 参数

参数	描述
sConnet	串口名称 + ":" + 波特率 或 串口名称（默认波特率115200）

- 返回值

返回值	描述
0	失败
1	成功

-

- 例子

▼ Pascal |

```
1  if OP0_Open(PAnsiChar(AnsiString('COM1:115200'))) = Integer(true) then
2      ShowMessage('Open Com OK!')
3  else
4      ShowMessage('Open Com Fail!');
```

2.4 关闭串口

- 描述

▼ Pascal |

```
1  function OP0_Close: Integer;
```

- 返回值

返回值	描述
0	失败
1	成功

- 例子

▼ Pascal |

```
1  if OP0_Close = Integer(true) then
2      ShowMessage('Close Com OK!')
3  else
4      ShowMessage('Close Com Fail!');
```

2.5 获取缓存重量信息

- 描述

▼

Pascal |

1 **function** OP0_ReadResultCache(sWeight: PAnsiChar): Integer;

- 参数

参数	描述
----	----

sWeight

1. 参数类型：指向一个ANSI 字符串的指针，指针指向的内存需要提前申请
2. 调用函数后，结果通过指针返回：
 - 返回数组格式：
"sMode,sStatus,sZero,sUnit,sNetWeight,sTareWeight,sGrossWeight"
 - 返回数组举例："N (sMode),S (sStatus),->0<- (sZero),kg (sUnit),0.000 (sNetWeight),0.000 (sTareWeight),0.000 (sGrossWeight)"
 - 字段说明：
 - sMode:
 - sMode = "N" (净重称量)
 - sMode = "T" (去皮称量)
 - sMode = "P" (预制去皮称量)
 - sStatus:
 - sStatus = "F" (重量溢出或没有开机归零)
 - sStatus = "S" (重量稳定)
 - sStatus = "U" (重量不稳定)
 - sZero:
 - sZero="->0<-" (零点)
 - sZero="" (非零点)
 - sUnit: (单位)
 - sNetWeight: (净重)
 - sTareWeight: (皮重)
 - sGrossWeight: (毛重)

- 返回值

返回值	描述
0	失败

1	成功
---	----

- 例子

▼

Pascal |

```

1
2 var s: array [0 .. 100] of Char;
3
4 //调用函数前, 你需要分配一块足够大的内存 (通常是一个字符数组), 并传入其首地址;
5 //调用后, 该函数会将结果以特定格式的字符串写入这块内存;
6
7 if OP0_ReadResultCache(@s[0]) = Integer(true) then
8     ShowMessage('ReadResultCache: ' + string(PAnsiChar(@s[0])))
9 else
10     ShowMessage('ReadResultCache: Fail!');
11

```

2.7 去皮

- 描述

▼

Pascal |

```

1 function OP0_Tare: Integer;

```

- 例子

▼

Pascal |

```

1 if OP0_Tare = Integer(true) then
2     ShowMessage('Tare: OK!')
3 else
4     ShowMessage('Tare: Fail!');

```

2.8退出去皮

- 描述

▼

Java |

```

1 function OP0_ExitTare: Integer;

```

- 例子

▼

Java |

```

1  if OP0_ExitTare = Integer(true) then
2      ShowMessage('ExitTare: OK!')
3  else
4      ShowMessage('ExitTare: Fail!');

```

2.9 强制去皮

- 描述

▼

Java |

```

1  function OP0_PreTare(Value: integer): integer;

```

- 参数

参数	描述
value	去皮重量，以 g 为单位

- 例子

▼

Java |

```

1  if OP0_PreTare(100) = Integer(true) then
2      ShowMessage('OP0_PreTare: OK!')
3  else
4      ShowMessage('OP0_PreTare: Fail!');

```

2.10 归零

- 描述

▼

Java |

```

1  function OP0_Zero: Integer;

```

- 例子

```
1  if OP0_Zero = Integer(true) then
2      ShowMessage('Zero: OK!')
3  else
4      ShowMessage('Zero: Fail!');
```