

Scale SDK-Windows-V2.012 Interface documentation

Document modification records

I. Instructions for Use

- 1.1 Description of Engineering Configuration
- 1.2 Function Declaration
- 1.3 Call Process

II. Interface Description

- 2.1 Weight Information Callback Function
- 2.2 Setting Callback Function
- 2.3 Open Serial Port
- 2.4 To Close The Serial Port
- 2.5 Retrieve The Cached Weight Information
- 2.6 Peeled
- 2.7 Quit Peel
- 2.8 Forced Peeling
- 2.9 Zeroing

Document modification records

serial Number	version Number	modify content	modifier	date modified
01	v2.012	document creation	zhuang Yongxing	2025-12-16

I. Instructions for Use

1.1 Description of Engineering Configuration

1. This SDK interface needs to run on Windows system
2. Put ScalesSDK.dll in the Sdk directory into the same directory or specified location of the application

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

1.2 Function Declaration

1. Declare library functions

```

1  sScalesLibName = 'ScalesSDK.dll';
2
3  // Callback function prototype
4  Procedure OnOPOModule(sModuleInfo: PAnsiChar; sLength: Integer); Stdca
   ll;
5
6  // Callback registration interface
7  procedure OPO_SetModuleCallBack(OnCallBack: Pointer); stdcall; externa
   l sScalesLibName;
8
9  // Open communication
10 function OPO_Open(sConnect: PAnsiChar): Integer; stdcall; external sSc
   alesLibName;
11
12 // Close communication
13 function OPO_Close: Integer; stdcall; external sScalesLibName;
14
15 // Read the weight from the buffer
16 function OPO_ReadResultCache(sWeight: PAnsiChar): Integer; stdcall; ex
   ternal sScalesLibName;
17
18 // Zero the scale
19 function OPO_Zero: Integer; stdcall; external sScalesLibName;
20
21 // Tare (subtract current weight as tare weight)
22 function OPO_Tare: Integer; stdcall; external sScalesLibName;
23
24 // Clear tare (exit tare mode)
25 function OPO_ExitTare: Integer; stdcall; external sScalesLibName;
26
27 // Preset tare command (set a specific tare value)
28 function OPO_PreTare(Value: integer): integer; stdcall; external sScal
   esLibName;

```

1.3 Call Process

1. Implement callback: Implement callback function OnOPOModule
2. Initialization: Call OPO_SetModuleCallBack to pass in OnOPOModule to set the callback function
3. Open Port: Call OPO_Open to open the serial port
4. Obtain information: Retrieve weight data via the callback function.

5. To close a port: call OPO_Close to close the serial port

II. Interface Description

2.1 Weight Information Callback Function

- description

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

- parameters

parameters	description

sModuleInfo	<ol style="list-style-type: none"> 1. parameter type: pointer to an ANSI string, the memory pointed to by The Pointer needs to be applied in advance. 2. After calling the function, the result is returned via a pointer : <ul style="list-style-type: none"> • returns array format: "sMode,sStatus,sZero, suit, snetwight, sTareWeight,sGrossWeight" • return an array example: "N (sMode),S (sStatus),->0<- (sZero),kg (unit),0.000 (snetwight),0.000 (sTareWeight),0.000 (sgrossight)" • field description: <ul style="list-style-type: none"> ○ sMode: <ul style="list-style-type: none"> ▪ sMode = "N" (net weighing) ▪ sMode = "T" (peeled and weighed) ▪ sMode = "P" (prefabricated peeling and weighing) ○ sStatus: <ul style="list-style-type: none"> ▪ sStatus = "F" (Weight overflow or no boot to zero) ▪ sStatus = "S" (stable weight) ▪ sStatus = "U" (unstable weight) ○ sZero: <ul style="list-style-type: none"> ▪ sZero = "->0<-" (zero point) ▪ sZero = "" (non-zero) ○ sUnit: (unit) ○ sNetWeight: (net weight) ○ sTareWeight: (tare weight) ○ sGrossWeight: (gross weight)
sLength	return Array length

- examples

▼

Pascal |

```

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

2.2 Setting Callback Function

- description

▼

Pascal |

```

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

- parameters

parameters	description
OnCallBack	onopomodule callback function pointer

- examples

▼

Pascal |

```

1 OP0_SetModuleCallBack(@OnOPOModule);
```

2.3 Open Serial Port

- description

▼

Pascal |

```

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

- parameters

parameters	description
sConnet	Serial port name + ":" + baud rate or serial port name (default baud rate 115200)

- return Value

return Value	description
0	failed
1	success

- examples

▼	Pascal
<pre> 1 if OPO_Open(PAnsiChar(AnsiString('COM1:115200')) = Integer(true) then 2 ShowMessage('Open Com OK!') 3 else 4 ShowMessage('Open Com Fail!');</pre>	

2.4 To Close The Serial Port

- description

▼	Pascal
<pre> 1 function OPO_Close: Integer;</pre>	

- return Value

return Value	description
0	failed
1	success

- examples

▼	Pascal
<pre> 1 if OPO_Close = Integer(true) then 2 ShowMessage('Close Com OK!') 3 else 4 ShowMessage('Close Com Fail!');</pre>	

2.5 Retrieve The Cached Weight Information

- description



Pascal

```
1  function OPO_ReadResultCache(sWeight: PAnsiChar): Integer;
```

- parameters

parameters	description
sWeight	<ol style="list-style-type: none"> 1. parameter type: pointer to an ANSI string, the memory pointed to by The Pointer needs to be applied in advance. 2. After calling the function, the result is returned via a pointer : <ul style="list-style-type: none"> • returns array format: "sMode,sStatus,sZero, suit, snetweight, sTareWeight,sGrossWeight" • Return an array example: "N (sMode),S (sStatus),->0<- (sZero),kg (unit),0.000 (snetweight),0.000 (sTareWeight),0.000 (sgrossweight)" • field description: <ul style="list-style-type: none"> ○ sMode: <ul style="list-style-type: none"> ▪ sMode = "N" (net weighing) ▪ sMode = "T" (peeled and weighed) ▪ sMode = "P" (prefabricated peeling and weighing) ○ sStatus: <ul style="list-style-type: none"> ▪ sStatus = "F" (Weight overflow or no boot to zero) ▪ sStatus = "S" (stable weight) ▪ sStatus = "U" (unstable weight) ○ sZero: <ul style="list-style-type: none"> ▪ sZero = "->0<-" (zero point) ▪ sZero = "" (non-zero) ○ sUnit: (unit) ○ sNetWeight: (net weight) ○ sTareWeight: (tare weight) ○ sGrossWeight: (gross weight)

- return Value

return Value	description
0	failed
1	success

- examples

▼ Pascal |

```

1
2 var s: array [0 .. 100] of Char;
3
4 // Before calling this function, you need to allocate a sufficiently large
  // block of memory (typically a character array) and pass in its starting address.
5 // After the call, the function will write the result as a formatted string
  // into this memory block.
6
7 if OPO_ReadResultCache(@s[0]) = Integer(true) then
8     ShowMessage('ReadResultCache: ' + string(PAnsiChar(@s[0])))
9 else
10     ShowMessage('ReadResultCache: Fail!');
11

```

2.6 Peeled

- description

▼ Pascal |

```

1 function OPO_Tare: Integer;

```

- examples

▼ Pascal |

```

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

```

2.7 Quit Peel

- description

▼ Java |

```
1  function OP0_ExitTare: Integer;
```

- examples

▼ Java |

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

2.8 Forced Peeling

- description

▼ Java |

```
1  function OP0_PreTare(Value: integer): integer;
```

- parameters

parameters	description
value	peeling weight in g.

- Examples

▼ Java |

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

2.9 Zeroing

- description

```
1  function OP0_Zero: Integer;
```

- examples

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