

C14465Mini Spectrometer Function Specification

Specification Document

Doc Version 1.0

Modification History

Oct.2017

First Edition

Content

<i>Modification History</i>	2
1. <i>Overview</i>	4
2. <i>HMS_initialize</i>	5
3. <i>HMS_uninitialize</i>	6
4. <i>HMS_getDeviceList</i>	7
5. <i>HMS_openDevice</i>	8
6. <i>HMS_closeDevice</i>	9
7. <i>HMS_checkDevice</i>	10
8. <i>HMS_getSeriesInformation</i>	11
9. <i>HMS_getDeviceInformation</i>	12
10. <i>HMS_getCalibrationCoefficient</i>	13
11. <i>HMS_setCalibrationCoefficient</i>	14
12. <i>HMS_readEepromUserArea</i>	15
13. <i>HMS_writeEepromUserArea</i>	16
14. <i>HMS_getIntegrationTime</i>	17
15. <i>HMS_getSensorInfo</i>	18
16. <i>HMS_setSensorInfo</i>	19
17. <i>HMS_getSensorData</i>	20
18. <i>HMS_stopSensorData</i>	21
19. <i>HMS_getGain</i>	22
20. <i>HMS_setGain</i>	23
21. <i>Error Code</i>	24

1. Overview

[Environment]

Microsoft(R) Windows 7 32bit, 64bit

Microsoft(R) Windows 8 32bit, 64bit

Microsoft(R) Windows 10 32bit, 64bit

[Development Environment]

Microsoft Visual Studio 2008 Professional Edition + SP1

[Driver]

Use WinUSB.SYS

[DLL]

HMSUSB2.dll

Spectrometer setting, control, data acquisition, data calculations are done by connecting USB2.0 between spectrometer and PC.

HMSUSB2.dll is x86DLL and x64DLL.

2. *HMS_initialize*

[Function]

Initialization of library.

[Format]

```
long HMS_initialize ( void );
```

[Arguments]

Nothing

[Return]

HMS_success	...Successful initialization of the library.
HMS_unsuccess	...Initialization of library is failed.
HMS_initialized	...Already initialized.

[Comment]

The library is initialized. After initialization, if HMS_initialized is called again, then it will return back. If you want to reinitialize, please reinitialize by using HMS_uninitialize function after opening a library.

[References]

HMS_uninitialize

3. *HMS_uninitialize*

[Function]

Exit device driver and library.

[Format]

long HMS_uninitialize (void);

[Arguments]

Nothing

[Return]

HMS_success	... Successful in exiting library
HMS_unsuccess	...Initialization of library is failed.
HMS_existDevice	... Exist the device which is not closed.
HMS_nowCapturing	... Capturing

[Comment]

Device driver and library is released. If the device is not closed, HMS_existDevice is returned. Please release the library after all devices are closed by HMS_close function.

[References]

HMS_initialize

4. *HMS_getDeviceList*

[Function]

Get the list of connected devices.

[Format]

```
long HMS_getDeviceList (
    array<String^>^ arylstrList ,
    unsigned short% usNumber
);
```

[Arguments]

arylstrList	...List of serial number is stored. Data of Serial number (char[32]) x 8 usNumber
usNumber	...Stores the number of connected devices.

[Return]

HMS_Success	...Successful in acquiring the list.
HMS_unsuccess	...Initialization of library is failed.
HMS_nowCapturing	...Capturing.

[Comment]

Get the list of connected devices. The list is a Device serial number of all connected modules. Please allocate space to arylstrList which is a list of char [32] * 8. usNumber is number of detected modules. For example if 5 devices is detected then 5 is acquired.

[References]

5. *HMS_openDevice*

[Function]

Opens device.

[Format]

long HMS_openDevice (String^ strSerial);

[Arguments]

strSerial ...Mini spectrometer serial number.

[Return]

HMS_success	...Successful in opening the device.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_opened	...Already opened.

[Comment]

Device is initialized. Please specify the serial number. Serial number can be acquired from HMS_getDeviceList function.

After using application, please close initialized device by using HMS_close function. If the spectrometer is removed from PC without closing, Windows system may get affected.

[References]

HMS_closeDevice

[Detail]

After Open process, exposure time is acquired and stored. This data is used in HMS_getSensorData.

6. *HMS_closeDevice*

[Function]

Closes device.

[Format]

```
long HMS_Close ( String^ strSerial);
```

[Arguments]

strSerial ...Spectrometer serial number.

[Return]

HMS_success ...Successful in closing the device.

HMS_unsuccess ...Failed in opening the device.

HMS_invalidSerial ...Invalid specified serial number.

HMS_notAccess ...Failed to access specified device.

HMS_nowCapturing ...Capturing.

[Comment]

Device in use is closed. Please use it when the application ends. If the USB device is removed from PC without releasing, Windows system may get affected.

[References]

HMS_openDevice

7. *HMS_checkDevice*

[Function]

Check device.

[Format]

long HMS_checkDevice(String^ strSerial);

[Arguments]

strSerial ...Spectrometer serial number.

[Return]

HMS_success	...Successful in closing the device.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_notAccess	...Failed to access specified device.
HMS_nowCapturing	...Capturing.
HMS_checkRemove	...The device has been removed

[Comment]

Connection of device is checked.

[References]

HMS_openDevice
HMS_closeDevice

8. *HMS_getSeriesInformation*

[Function]

Get device series product information.

[Format]

```
long HMS_getSeriesInformation(String^ strSerial,
                             CSeriesInformation^ hSeriesInformation );
```

[Arguments]

strSerial ...Spectrometer serial number.

```
public ref class CSeriesInformation {
    String^ m_szVender ;           // Company name (char[32])
    String^ m_szProduct ;         // Product(char[32])
    String^ m_szFirmware;         // Firmware Ver (char[32])
    String^ m_szReserve;          // Reserve (char[32])
    String^ m_szLibrary;          // DLL Ver (char[32])
};
```

[Return]

HMS_success	...Successful in closing the device.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_notAccess	...Failed to access specified device.
HMS_nowCapturing	...Capturing.

[Comment]

Device series product information is acquired.

[References]

HMS_setSeriesInformation

9. *HMS_getDeviceInformation*

[Function]

Get information of each device.

[Format]

```
long HMS_getDeviceInformation(String^ strSerial,
                             CDeviceInformation ^ hDeviceInformation);
```

[Arguments]

strSerial ...Spectrometer serial number.

```
public ref class CDeviceInformation {
    String^ m_szUnit;           // Unit ID(char[8])
    String^ m_szSensor;         // Sensor name (char[16])
    String^ m_szSerial;         // Serial number (char[32])
    long    m_lUpperWavelength; // Upper sensitivity wavelength
    long    m_lLowerWavelength; // Lower sensitivity wavelength
};
```

[Return]

HMS_success	...Successful in closing the device.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_notAccess	...Failed to access specified device.
HMS_nowCapturing	...Capturing.

[Comment]

Get information of the each individual device.

[References]

HMS_setDeviceInformation

10. *HMS_getCalibrationCoefficient*

[Function]

Get calibration coefficient of device.

[Format]

```
long HMS_getCalibrationCoefficient(String^ strSerial, long lIndex,
    array<double>^ arydblCoefficient );
```

[Arguments]

strSerial	...Spectrometer serial number.
lIndex	...Coefficient type
0	Wavelength
1	after reservation
coefficient ...	calibration coefficient

[Return]

HMS_success	...Successful in closing the device.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_notAccess	...Failed to access specified device.
HMS_nowCapturing	...Capturing.
HMS_invalidValue	...Invalid value

[Comment]

Currently set calibration coefficient is acquired. There are only (1- 6 types) of calibration coefficients, but calibration coefficient for sensitivity is reserved.

[References]

HMS_setCalibrationCoefficient

11. HMS_setCalibrationCoefficient

[Function]

Set calibration coefficient of device.

[Format]

```
long HMS_setCalibrationCoefficient(String^ strSerial, long lIndex,
    array<double>^ arydblCoefficient ) ;
```

[Arguments]

strSerial	...Spectrometer serial number.
lIndex	...Coefficient type
0	Wavelength
1	after reservation
coefficient ...	calibration coefficient

[Return]

HMS_success	...Successful in closing the device.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_notAccess	...Failed to access specified device.
HMS_nowCapturing	...Capturing.
HMS_invalidValue	...Invalid value

[Comment]

Calibration coefficient is set. There are only (1- 6 types) of calibration coefficients, but calibration coefficient for sensitivity is reserved.

[References]

HMS_getCalibrationCoefficient

12. *HMS_readEepromUserArea*

[Function]

It reads data of EEPROM user area.

[Format]

```
long HMS_readEepromUserArea(String^ strSerial, unsigned short usAddress,
                             array<unsigned char>^ aryucData,
                             unsigned short usLength );
```

[Arguments]

strSerial ...Spectrometer serial number.

usAddress ...The initial address where data is read. The address that can be specified is 0-67. Address will be specified in word. If an address is specified outside the region, the function returns an error.

aryucData...Handle to store read data.

usLength... Specifies length of read data. The data length that can be read at a time is changed by specified addressable range. This data is up to 128. Please specify from 1 to 128.

[Return]

HMS_Success ...Data is read successfully.

HMS_unsuccess ...Failed in opening the device.

HMS_invalidSerial ...Invalid specified serial number.

HMS_notAccess ...Failed to access specified device.

HMS_invalidAddress ...Invalid specified initial address.

HMS_invalidLength ...Invalid specified data length.

HMS_overArea ...Specified data is out of range of user area.

(Example: If user area is 0-99, an initial address is 95, and data length is 10)

HMS_nowCapturing ...Capturing.

[Comment]

It Reads data of EEPROM user area.

[References]

HMS_writeEepromUserArea

13. *HMS_writeEepromUserArea*

[Function]

It writes data of EEPROM user area.

[Format]

```
long HMS_writeEepromUserArea(String^ strSerial, unsigned short usAddress,
                             array<unsigned char>^ aryucData,
                             unsigned short usLength );
```

[Arguments]

strSerial ...Spectrometer serial number.

usAddress ...The initial address where data is read. The address that can be specified is 0-67. Address will be specified in word. If an address is specified outside the region, the function returns an error.

aryucData...Handle to store read data.

usLength... Specifies length of read data. The data length that can be read at a time is changed by specified addressable range. This data is up to 128. Please specify from 1 to 128.

[Return]

HMS_Success ...Data is read successfully.

HMS_unsuccess ...Failed in opening the device.

HMS_invalidSerial ...Invalid specified serial number.

HMS_notAccess ...Failed to access specified device.

HMS_invalidAddress ...Invalid specified initial address.

HMS_invalidLength ...Invalid specified data length.

HMS_overArea ...Specified data is out of range of user area.

(Example: If user area is 0-99, an initial address is 95, and data length is 10)

HMS_nowCapturing ...Capturing.

[Comment]

It writes data of EEPROM user area.

[References]

HMS_readEepromUserArea

14. *HMS_getIntegrationTime*

[Function]

Get device integration time.

[Format]

```
long HMS_getIntegrationTime(String^ strSerial, unsigned long% ulIntegrationTime);
```

[Arguments]

strSerial	...Spectrometer serial number
ulIntegrationTime	...Region to store integration time(unit is msec) [msec] 5~10000

[Return]

HMS_Success	...Data is read successfully.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_nowCapturing	...Capturing.

[Comment]

Get device integration time.

[References]

HMS_getSensorData

15. *HMS_getSensorInfo*

[Function]

Get device sensor information.

[Format]

```
long HMS_getSensorInfo(String^ strSerial, unsigned long ulFlag, unsigned long%
ulData );
```

[Arguments]

strSerial	...Spectrometer serial number
ulFlag	...Type
	0 Data size
	1 Unused
ulData	...Stores data

[Return]

HMS_Success	...Data is read successfully.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_nowCapturing	...Capturing.
HMS_invalidValue	...Invalid value

[Comment]

Get device sensor information.

[References]

16. *HMS_setSensorInfo*

[Function]

Set device sensor information.

[Format]

```
long HMS_setSensorInfo(String^ strSerial, unsigned long ulFlag, unsigned long%
    ulData );
```

[Arguments]

strSerial	...Spectrometer serial number
ulFlag	...Type
	1 Data size
	1 Unused
ulData	...Stores data

[Return]

HMS_Success	...Data is read successfully.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_nowCapturing	...Capturing.
HMS_invalidValue	...Invalid value

[Comment]

Set device sensor information.

[References]

17. *HMS_getSensorData*

[Function]

Get Spectroscopic data from the device.

[Format]

```
long HMS_getSensorData(String^ strSerial, unsigned long ulIntegrationTime,
                        array<unsigned short>^ aryusIntegrationData );
```

[Arguments]

strSerial	...Spectrometer serial number
ulIntegrationTime	...Integration time [msec] 5~10000
aryusIntegrationData	...Region to store Integration time (256)

[Return]

HMS_Success	...Data is read successfully.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_checkRemove	...Device is removed.
HMS_nowCapturing	...Capturing.
HMS_timeOutErr	...Timeout occurred during measurement.

[Comment]

Get Spectroscopic data from the device.

[References]

HMS_openDevice
HMS_checkDevice

[Details]

- ①Check the device connection using HMS_checkDevice
No corresponding device / Removed → Exit at the appropriate return value
- ②If there is change in the integration time stored in HMS_openDevice, integration time is set, integration time is not set →Exit at the appropriate return value
- ③Data acquisition request command issue
- ④Get the data. (Loop data)

18. *HMS_stopSensorData*

[Function]

Stop Spectroscopic data from the device.

[Format]

```
long HMS_stopSensorData(String^ strSerial ) ;
```

[Arguments]

strSerial	...Spectrometer serial number
-----------	-------------------------------

[Return]

HMS_Success	...Data is read successfully.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_checkRemove	...Device is removed.

[Comment]

Stop Spectroscopic data from the device.

[References]

HMS_getSensorData

19. *HMS_getGain*

[Function]

Get device gain value.

[Format]

```
long HMS_getGain(String^ strSerial, unsigned long ulMode, unsigned long% ulGain);
```

[Arguments]

strSerial	...Spectrometer serial number
ulMode	...Mode
	0 Sensor gain mode
	1 Circuit gain mode
ulGain	...Store gain
	0 LOW gain
	1 HIGH gain

[Return]

HMS_Success	...Data is read successfully.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_checkRemove	...Device is removed.
HMS_nowCapturing	...Capturing.
HMS_invalidValue	...Invalid value

[Comment]

Get device gain value.

[References]

20. *HMS_setGain*

[Function]

Set device gain value.

[Format]

```
long HMS_setGain(String^ strSerial, unsigned long ulMode, unsigned long% ulGain);
```

[Arguments]

strSerial	...Spectrometer serial number
ulMode	...Mode
	0 Sensor gain mode
	1 Circuit gain mode
ulGain	...Store gain
	0 LOW gain
	1 HIGH gain

[Return]

HMS_Success	...Data is read successfully.
HMS_unsuccess	...Failed in opening the device.
HMS_invalidSerial	...Invalid specified serial number.
HMS_checkRemove	...Device is removed.
HMS_nowCapturing	...Capturing.
HMS_invalidValue	...Invalid value

[Comment]

Set device gain value.

[References]

21. Error Code

Symbol	Value	Comment
HMS_success	0	Success
HMS_unsuccess	1	Failed
HMS_initialized	2	Already initialized
HMS_existDevice	3	Exists the device which is not closed.
HMS_nowCapturing	4	Capturing
HMS_invalidSerial	5	Invalid specified serial number.
HMS_opened	6	Already opened.
HMS_notAccess	7	Failed to access specified device.
HMS_invalidAddress	8	Invalid specified initial address.
HMS_invalidLength	9	Invalid specified data length.
HMS_overArea	10	Specified data is out of range of user area.
HMS_invalidMode	11	Invalid specified operating mode.
HMS_checkRemove	12	The device has been removed
HMS_invalidValue	13	Invalid value
HMS_timeOutErr	14	Timeout occurred during measurement.