

HAMAMATSU

C14465 ミニ分光器 DLL 関数仕様

仕 様 書

Doc Version 1.0

変更履歴

2017 年 10 月

初版

内容

変更履歴	2
1. 概要.....	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. 概要

【環境】

Microsoft(R) Windows 7 32bit, 64bit

Microsoft(R) Windows 8 32bit, 64bit

Microsoft(R) Windows 10 32bit, 64bit

【開発環境】

Microsoft Visual Studio 2008 Professional Edition + SP1

【ドライバ】

WinUSB.SYS を使用する。

【DLL】

HMSUSB2.dll

分光器と PC 間で USB2.0 を介し分光器の設定・制御・データの取得・データ演算抽出を行う。HMSUSB2.dll は、x86DLL と x64DLL になります。

2. *HMS_initialize*

【機能】

ライブラリのイニシャライズ

【書式】

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

【引数】

なし

【戻り値】

HMS_success	…………成功
HMS_unsuccess	…………失敗
HMS_initialized	…………イニシャライズ済み

【解説】

ライブラリの初期化を行います。初期化後、再び呼び出すと **HMS_initialized** が返ります。再初期化したい場合は、**HMS_uninitialize** 関数を使用してライブラリを開放した後再び初期化してください。

【参照】

HMS_uninitialize

3. HMS_uninitialize

【機能】

デバイスドライバとライブラリの終了

【書式】

```
long HMS_uninitalize( void );
```

【引数】

なし

【戻り値】

HMS_success	…………成功
HMS_unsuccess	…………失敗
HMS_existDevice	…………クローズしていないデバイスがある
HMS_nowCapturing	…………計測中

【解説】

ライブラリを開放します。クローズしていないデバイスが有る場合、HMS_existDevice が返ります。HMS_closeDevice 関数にて全てのデバイスをクローズした後にライブラリを開放してください。

【参照】

HMS_initialize

4. HMS_getDeviceList

【機能】

接続されているデバイスのリストを取得する

【書式】

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

【引数】

arysrList	…………シリアル番号のリスト格納先 シリアル番号(char[32])×8 個分のデータ
usNumber	…………デバイスの個数格納先

【戻り値】

HMS_success	…………成功
HMS_unsuccess	…………失敗
HMS_nowCapturing	…………計測中

【解説】

接続デバイスのリストを取得します。リストとは接続されている全てのデバイスのシリアル番号のことです。arysrList はリストを char[32]*8 の領域を確保してください。usNumber は検出したデバイスの個数を取得します。例えば 5 台のデバイスを検出した場合は 5 を取得します。

【参照】

5. HMS_openDevice

【機能】

デバイスのオープン

【書式】

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

【引数】

strSerial分光器シリアル番号

【戻り値】

HMS_success成功

HMS_unsuccess失敗

HMS_invalidSerial ……指定したシリアル番号が無効

HMS_opened既にオープン済み

【解説】

デバイスを初期化します。シリアル番号を指定してください。シリアル番号は `HMS_getDeviceList` 関数で取得できます。

初期化したデバイスはアプリケーション使用後に **HMS_closeDevice** 関数を使用して必ず開放してください。開放を行わず PC から分光器を外した場合、**Window** システムに支障を来す場合があります。

【参照】

HMS_closeDevice

【詳細】

Open 処理の後、蓄積時間をデバイスから取得し保持すること。このデータは HMS_getSensorData で使用する。

6. HMS_closeDevice

【機能】

デバイスのクローズ

【書式】

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

【引数】

strSerial分光器シリアル番号

【戻り値】

HMS_success成功

HMS_unsuccess失敗

HMS_invalidSerial ……指定したシリアル番号が無効

HMS_notAccess	…………指定したシリアル番号のアクセスに失敗
---------------	------------------------

HMS_nowCapturing計測中

【解説】

使用中のデバイスを開放します。アプリケーション終了後、必ず実行してください。開放をおこなわず PC からデバイスを取り外した場合、**Windows** システムに支障を来す場合があります。

【参照】

HMS_openDevice

10

8. HMS_getSeriesInformation

【機能】

デバイスのシリーズ製品情報を取得する

【書式】

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

【引数】

```
strSerial          .....分光器シリアル番号
public ref class CSeriesInformation {
    String^ m_szVender;           // 社名(char[32])
    String^ m_szProduct;          // 型番(char[32])
    String^ m_szFirmware;         // ファームウェア Ver(char[32])
    String^ m_szReserve;          // 予約(char[32])
    String^ m_szLibrary;          // DLL Ver(char[32])
};
```

【戻り値】

HMS_success	成功
HMS_unsuccess	失敗
HMS_invalidSerial	指定したシリアル番号が無効
HMS_notAccess	指定したシリアル番号のアクセスに失敗
HMS_nowCapturing	計測中

【解説】

デバイスのシリーズ製品情報を取得します。

【参照】

HMS_setSeriesInformation

9. HMS_getDeviceInformation

【機能】

デバイスの製品個別情報を取得

【書式】

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

【引数】

```
strSerial          ……分光器シリアル番号
public ref class CDeviceInformation {
    String^ m_szUnit;           // ユニット ID(char[8])
    String^ m_szSensor;        // センサ名(char[16])
    String^ m_szSerial;        // シリアル番号(char[32])
    long m_lUpperWavelength;   // 波長感度領域の上限
    long m_lLowerWavelength;   // 波長感度領域の下限
};
```

【戻り値】

HMS_success	……成功
HMS_unsuccess	……失敗
HMS_invalidSerial	……指定したシリアル番号が無効
HMS_notAccess	……指定したシリアル番号のアクセスに失敗
HMS_nowCapturing	……計測中

【解説】

デバイスの製品個別情報を取得します。

【参照】

HMS_setDeviceInformation

10. HMS_getCalibrationCoefficient

【機能】

デバイスの校正係数を取得

【書式】

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

【引数】

strSerial	分光器シリアル番号
lIndex	係数種類
	0 波長
	1 以降予約
arydblCoefficient	係数

【戻り値】

HMS_success	成功
HMS_unsuccess	失敗
HMS_invalidSerial	指定したシリアル番号が無効
HMS_notAccess	指定したシリアル番号のアクセスに失敗
HMS_nowCapturing	計測中
HMS_invalidValue	不適な数値があった

【解説】

デバイスの校正係数を取得します。実装するのは波長校正係数(1 式 6 種類)のみですが、感度校正を想定したものをリザーブします。

【参照】

HMS_setCalibrationCoefficient

11. HMS_setCalibrationCoefficient

【機能】

デバイスの校正係数を設定

【書式】

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

【引数】

strSerial ……分光器シリアル番号

lIndex ……係数種類

0 波長

1 以降予約

arydblCoefficient ……係数

【戻り値】

HMS_success ……成功

HMS_unsuccess ……失敗

HMS_invalidSerial ……指定したシリアル番号が無効

HMS_notAccess ……指定したシリアル番号のアクセスに失敗

HMS_nowCapturing ……計測中

HMS_invalidValue ……不適な数値があった

【解説】

デバイスの校正係数を設定します。実装するのは波長校正係数(1 式 6 種類)のみですが、感度校正を想定したものをリザーブします。

【参照】

HMS_getCalibrationCoefficient

12. HMS_readEepromUserArea

【機能】

ユーザー領域のデータを読み込み

【書式】

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

【引数】

strSerial	…………分光器シリアル番号
usAddress	…………データを読み込む先頭のアドレス 指定可能なアドレスは 0～67。アドレスはワード単位による番地とする。ユーザー領域外のアドレスを指定した場合、関数はエラーを返す。
aryucData	…………読み出すデータを格納する領域へのハンドル
usLength	…………読み出すデータの長さ 一度に読込可能なデータ長はワード単位で指定可能アドレス範囲によって変わる。今回は最大 68 データとする。1～68 を指定すること。

【戻り値】

HMS_success	…………成功
HMS_unsuccess	…………失敗
HMS_invalidSerial	…………指定したシリアル番号が無効
HMS_notAccess	…………指定したデバイスのアクセス失敗
HMS_invalidAddress	…………指定した先頭アドレスが無効
HMS_invalidLength	…………指定したデータの長さが無効
HMS_overArea	…………指定しデータがユーザー領域の範囲外 (例:ユーザー領域が 0～99 で、先頭アドレスが 95、データの長さが 10 の場合など)
HMS_nowCapturing	…………計測中

【解説】

ユーザー領域のデータを読み込みます。

【参照】

HMS_writeEepromUserArea

13. HMS_writeEepromUserArea

【機能】

ユーザー領域のデータを書き込み

【書式】

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

【引数】

strSerial	………分光器シリアル番号
usAddress	………データを読み込む先頭のアドレス 指定可能なアドレスは 0～67。アドレスはワード単位による番地とする。ユーザー領域外のアドレスを指定した場合、関数はエラーを返す。
aryucData	………読み出すデータを格納する領域へのハンドル
usLength	………読み出すデータの長さ 一度に読込可能なデータ長は指定可能アドレス範囲によって変わる。今回は最大 68 データとする。1～68 を指定すること。

【戻り値】

HMS_success	………成功
HMS_unsuccess	………失敗
HMS_invalidSerial	………指定したシリアル番号が無効
HMS_notAccess	………指定したデバイスのアクセス失敗
HMS_invalidAddress	………指定した先頭アドレスが無効
HMS_invalidLength	………指定したデータの長さが無効
HMS_overArea	………指定しデータがユーザー領域の範囲外 (例:ユーザー領域が 0～99 で、先頭アドレスが 95、データの長さが 10 の場合など)
HMS_nowCapturing	………計測中

【解説】

ユーザー領域のデータを書き込みます。

【参照】

HMS_readEepromUserArea

14. HMS_getIntegrationTime

【機能】

デバイス蓄積時間の取得

【書式】

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

【引数】

strSerial	…………分光器シリアル番号
ulIntegrationTime	…………蓄積時間を格納する領域(単位は msec) [msec] 5~10000

【戻り値】

HMS_success	…………成功
HMS_unsuccess	…………失敗
HMS_invalidSerial	…………指定したシリアル番号が無効
HMS_nowCapturing	…………計測中

【解説】

デバイスの蓄積時間の値を取得します。

【参照】

HMS_getSensorData

15. HMS_getSensorInfo

【機能】

デバイスのセンサー情報を取得

【書式】

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

【引数】

strSerial	………分光器シリアル番号
ulFlag	………種類
	0 データサイズ
	1 以降は未使用
ulData	………データ格納領域

【戻り値】

HMS_success	………成功
HMS_unsuccess	………失敗
HMS_invalidSerial	………指定したシリアル番号が無効
HMS_nowCapturing	………計測中
HMS_invalidValue	………不適な数値があった

【解説】

デバイスのセンサー情報を取得します。

【参照】

16. HMS_setSensorInfo

【機能】

デバイスのセンサー情報を設定

【書式】

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

【引数】

strSerial	………分光器シリアル番号
ulFlag	………種類
	0 データサイズ
	1 以降は未使用
ulData	………データ

【戻り値】

HMS_success	………成功
HMS_unsuccess	………失敗
HMS_invalidSerial	………指定したシリアル番号が無効
HMS_nowCapturing	………計測中
HMS_invalidValue	………不適な数値があった

【解説】

デバイスのセンサー情報を設定します。

【参照】

17. HMS_getSensorData

【機能】

デバイスから分光データを取得

【書式】

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

【引数】

strSerial	………分光器シリアル番号
ulIntegrationTime	………蓄積時間 [msec] 5～10000
aryusIntegrationData	………蓄積データを格納する領域 (256)

【戻り値】

HMS_success	………成功
HMS_unsuccess	………失敗
HMS_invalidSerial	………指定したシリアル番号が無効
HMS_checkRemove	………デバイスが取り外されている
HMS_nowCapturing	………計測中
HMS_timeOutErr	………計測中タイムアウト発生

【解説】

デバイスから分光データを取得します。

【参照】

HMS_openDevice
HMS_checkDevice

【詳細】

- ①HMS_checkDevice でデバイス接続の確認
相当デバイスがない／外された → 該当戻り値にて終了
- ②HMS_openDevice で保持した蓄積時間の変更があった場合には、蓄積時間の設定し、
蓄積時間が設定できなかった → 該当戻り値にて終了
- ③データ取得要求コマンド発行
- ④データ取得（ループで待つ）

18. HMS_stopSensorData

【機能】

デバイスから分光データの取得を中止

【書式】

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

【引数】

strSerial分光器シリアル番号

【戻り値】

HMS_success成功

HMS_unsuccess失敗

HMS_invalidSerial	…………指定したシリアル番号が無効
-------------------	-------------------

HMS_checkRemoveデバイスが取り外されている

【解説】

デバイスから分光データの取得を中止します。

【参照】

HMS_getSensorData

19. HMS_getGain

【機能】

デバイスのゲイン値を取得

【書式】

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

【引数】

strSerial	…………分光器シリアル番号
ulMode	…………モード
	0 センサーゲインモード
	1 回路ゲインモード
ulGain	…………ゲイン値格納
	0 LOW ゲイン
	1 HIGH ゲイン

【戻り値】

HMS_success	…………成功
HMS_unsuccess	…………失敗
HMS_invalidSerial	…………指定したシリアル番号が無効
HMS_checkRemove	…………デバイスが取り外されている
HMS_nowCapturing	…………計測中
HMS_invalidValue	…………不適な数値があった

【解説】

デバイスのゲイン値を取得します。

【参照】

20. HMS_setGain

【機能】

デバイスのゲイン値を設定

【書式】

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

【引数】

strSerial	…………分光器シリアル番号
ulMode	…………モード
	0 センサーゲインモード
	1 回路ゲインモード
ulGain	…………ゲイン値格納
	0 LOW ゲイン
	1 HIGH ゲイン

【戻り値】

HMS_success	…………成功
HMS_unsuccess	…………失敗
HMS_invalidSerial	…………指定したシリアル番号が無効
HMS_checkRemove	…………デバイスが取り外されている
HMS_nowCapturing	…………計測中
HMS_invalidValue	…………不適な数値があった

【解説】

デバイスのゲイン値を設定します。

【参照】

21. Error Code

Symbol	Value	解説
HMS_success	0	成功
HMS_unsuccess	1	失敗
HMS_initialized	2	イニシャライズ済み
HMS_existDevice	3	クローズしていないデバイスが有る
HMS_nowCapturing	4	計測中
HMS_invalidSerial	5	指定したシリアル番号が無効
HMS_opened	6	既にオープン済み
HMS_notAccess	7	指定したシリアル番号のアクセスに失敗
HMS_invalidAddress	8	指定した先頭アドレスが無効
HMS_invalidLength	9	指定したデータ長が無効
HMS_overArea	10	指定したデータがユーザー領域の範囲外
HMS_invalidMode	11	指定した動作モードが無効
HMS_checkRemove	12	デバイスが取り外されている
HMS_invalidValue	13	不適な数値があった
HMS_timeOutErr	14	計測中タイムアウト発生