

# MCA-Lite シリーズ用ライブラリ

# mcalitedll.dll

## 解説書

C++Builder 用の inport library は mcalitedll.lib です  
プロジェクトに追加してください。  
ヘッダーファイル mcalitedll.h を #include します。  
( 26-JUN-2018 版 )



株式会社TNB工房

<http://www.nabe-e.com>

## 目次

### 内容

はじめに.....	3
● TNBUSB_Open()      USB回線一つをオープンする.....	4
● TNBUSB_Write()      USB回線へデータを出力する（使わない）.....	5
● TNBUSB_Read()      USB回線からデータを読み出す.....	5
● TNBUSB_RW_Qsize()      USB回線(入出力)に残っているデータ量の確認.....	6
● TNBUSB_ReadQsize()      USB回線の入力バッファから指定バイトを空読み.....	6
● TNBUSB_Close()      USB回線一つを CLOSE する.....	7
● MCALite_ReadVersion()      MCA-Lite シリーズのバージョンを読み出します.....	8
● MCALite_Preset()      MCA 測定の測定条件を指定します.....	9
● MCALite_StartCollect()      MCA 測定を開始します.....	10
● MCALite_StopCollect()      MCA 測定を終了します.....	10
● MCALite_ClearSPC()      MCA-Lite シリーズ本体スペクトルをクリアします.....	10
● MCALite_ReadStatus()      MCA-Lite の測定中停止中の状態を読み込みます.....	11
● MCALite_ReadSPC()      MCA-Lite 内部のスペクトルを、読み出します.....	12

## はじめに

MCA-Lite シリーズは、共通の USB 命令で動作します。

命令は、USBの基本命令 6命令と、 7 個のMCA専用の制御命令があります。

MCA 命令では、MCA-Lite シリーズのハードウェアや通信命令形式を意識しないで、簡単にプログラムできます。

### USB基本命令

- |                      |                         |
|----------------------|-------------------------|
| ● TNBUSB_Open()      | USB回線一つをオープンする          |
| ● TNBUSB_Write()     | USB回線へデータを出力する          |
| ● TNBUSB_Read()      | USB回線からデータを読み出す         |
| ● TNBUSB_RW_Qsize()  | USB回線(入出力)に残っているデータ量の確認 |
| ● TNBUSB_ReadQsize() | USB回線の入力バッファから指定バイトを空読み |
| ● TNBUSB_Close()     | USB回線一つを CLOSE する       |

### MCA命令

- |                          |                              |
|--------------------------|------------------------------|
| ● MCALite_ReadVersion()  | MCA-Lite シリーズのバージョンを読み出します   |
| ● MCALite_Preset()       | MCA 測定の測定条件を指定します。           |
| ● MCALite_StartCollect() | MCA 測定を開始します。                |
| ● MCALite_StopCollect()  | MCA 測定を終了します。                |
| ● MCALite_ClearSPC()     | MCA-Lite シリーズ本体スペクトルをクリアします。 |
| ● MCALite_ReadStatus()   | MCA-Lite の測定中停止中の状態を読み込みます。  |
| ● MCALite_ReadSPC()      | MCA-Lite 内部のスペクトルを、読み出します    |

解説書は、C++Builder で記述しています。

使用期間制限の無い最新の C++Builder で開発する場合は、そのまま利用できます。

C++Builder の様に開発言語専用のIMPORT-Library .lib を必要とする言語の場合はそれぞれの言語の生成方法で、mcalitedll.dll → mcalitedll.lib の様に生成して

リンクしてください。 技術的な方法は、それぞれの DLL の使い方を調べて対応してください。

## ● TNBUSB\_Open()                    U S B 回線一つをオープンする

---

記述方法

```
int TNBUSB_Open(char USB番号, int 装置番号,
                 int W-timeout, int R-timeout)
```

USB番号 8 文字	MCA-Lite の USB 番号で “A9004B6E” のような 8 文字の文字列
装置番号	USB 番号の MCA-Lite に機器番号を割り付けます。0 から 7 までの数値
W-timeout	データ出力命令の時のタイムアウトを msec の単位で指定する
R-timeout	データ入力命令の時のタイムアウトを msec の単位で指定する

return 値

-1	何らかのエラーが発生
0	正常終了

使用例

```
char  usbName[]="A9005BAZ";
int    rt;
      rt=TNBUSB_Open(usbSname ,1,200,200);
      if(rt<0)
      {
          MessageBeep(0);
          ShowMessage("USB SN name ERROR");
      }
```

通信のタイムアウト時間を 200msec にしています。

オープンした MCA-Lite は機器番号 1 番として割り付けました。

複数の MCA-Lite を接続する場合は、続けて OPEN 命令を発行しますが  
機器番号は、2,3,4...と異なる番号で OPEN します。

● TNUSB\_Write()                    U S B回線へデータを出力する (使わない)

---

int TNUSB\_Write(int 装置番号, char\* 送信データ, int データByte数)

装置番号	OPEN の時に指定した、MCA-Lite の識別番号
送信データ	MCA-Lite へ送り出すデータ列
データByte数	送信データのバイト数

return 値

-1	何らかのエラーが発生
0	正常終了

この命令は、後で説明する、MCA-Lite 専用命令を使うので、使うことは無い。

● TNUSB\_Read()                    U S B回線からデータを読み出す

---

int TNUSB\_Read(int 装置番号, char\* 受信データ領域, int データByte数)

装置番号	OPEN の時に指定した、MCA-Lite の識別番号
受信データ領域	MCA-Lite が送り出すデータ列を読んで、格納する領域を指定する
データByte数	送信データのバイト数

return 値

-1	何らかのエラーが発生
0	正常終了

この命令は、後で説明する、MCA-Lite 専用命令を使うので、使うことは無い。

---

● TNBUSB\_RW\_Qsize()                      U S B回線(入出力)に残っているデータ量の確認

---

**int TNBUSB\_Open(int 装置番号, int &受信データ数, &送信データ数)**

装置番号	OPEN の時に指定した、MCA-Lite の識別番号
受信データ量	USBドライバの中に残っている未読データ数(Byte)
送信データ量	USBドライバの中に残っている未読送信データ数(Byte)

return 値

-1	何らかのエラーが発生
0	正常終了

この命令は、使うことは無い。

専用命令で、データ受信数よりも少ない量を読み込んでいる場合にバッファにデータが残り次の命令の応答データとして取り出してしまい、トラブルを起こすことがある。

この様な時に、この命令で、内部のバッファデータを確認するための、DEBUG 用の命令である

---

● TNBUSB\_ReadQsize()                      U S B回線の入力バッファから指定バイトを空読み

---

**int TNBUSB\_ReadQsize(int 装置番号)**

装置番号	OPEN の時に指定した、MCA-Lite の識別番号
------	-----------------------------

return 値

-1	何らかのエラーが発生
0	正常終了

この命令は、使うことは無い。

専用命令で、データ受信数よりも少ない量を読み込んでいる場合にバッファにデータが残り次の命令の応答データとして取り出してしまい、トラブルを起こすことがある。

この様な時に、この命令で、内部のバッファデータを確認するための、DEBUG 用の命令である  
もし、読み残しがある場合は、読み残しが無いように、プログラムを修正する必要がある。

そうでなければ、USB エラーが多発するプログラムになる。

● TNUSB\_Close()                    U S B 回線一つを CLOSE する

---

int TNUSB\_Close(int 装置番号)

装置番号                    OPEN の時に指定した、MCA-Lite の識別番号

return 値

- 1      何らかのエラーが発生
- 0      正常終了

この命令は実行しないで、プログラムを終了すると、システムが USB を CLOSE してくれます。ただ、OPEN したものは、Close して終わるようにしましょう。

USB ハブで、8 個を超える MCA-Lite を使う場合は、この命令で、任意の MCA-Lite を Close して、他の MCA-Lite を OPEN することも考えられますが

その場合は、一度に OPEN できるハンドル数を増やす必要がありますが、そこまでは、このライブラリでは考えていません。

● MCALite\_ReadVersion() MCA-Lite シリーズのバージョンを読み出します

---

## int MCALite\_ReadVersion(int 装置番号, char &受取領域)

装置番号	OPEN の時に指定した、MCA-Lite の識別番号
受取領域	7 文字のレビジョン番号と CR コードの 8 文字が戻ります。 “REV1.20”+CR など

### RETURN

-1	命令送信の時にエラーが発生
-2	データ受信の時にエラーが発生
0	正常終了

### 記述例

```
int rt = MCALite_ReadVersion(1,cbuff);
if(rt<0)
{
    MessageBeep(0);
    ShowMessage("MCA-Lite ReadVersion ERROR");
    hajime=1;
    return;
}
StatusBar1->SimpleText = cbuff;
```



● MCALite\_Preset() MCA 測定の測定条件を指定します。

---

**int MCALite\_Preset(int 装置番号, char Mode, int spcSize,  
int 測定時間)**

装置番号	OPEN の時に指定した、MCA-Lite の識別番号
Mode	‘L’= livetime 設定 ‘R’=realtime 設定
spcsize	スペクトルサイズ 256,512,1024,2048,4096 が指定できる
測定時間	秒の単位で、測定時間を指定します

RETURN

-1	命令送信の時にエラーが発生
-2	データ受信の時にエラーが発生
0	正常終了

記述例

```
int rt;  
int ptime = 1000; // 1000 秒測定  
int fullsize = 4096;  
char presetMode='L';  
  
rt = MCALite_Preset(1,presetMode, fullsize, ptime);  
if(rt<0)  
{  
    ShowMessage("Preset ERROR");  
}
```

- MCALite\_StartCollect() MCA 測定を開始します。
- 

## int MCALite\_StartCollect(int 装置番号)

装置番号                    OPEN の時に指定した、MCA-Lite の識別番号

### RETURN

- 1            命令送信の時にエラーが発生
- 2            データ受信の時にエラーが発生
- 0            正常終了

- MCALite\_StopCollect() MCA 測定を終了します。
- 

## int MCALite\_StopCollect(int 装置番号)

装置番号                    OPEN の時に指定した、MCA-Lite の識別番号

### RETURN

- 1            命令送信の時にエラーが発生
- 2            データ受信の時にエラーが発生
- 0            正常終了

- MCALite\_ClearSPC()                    MCA-Lite シリーズ本体スペクトルをクリアします。
- 

## int MCALite\_ClearSPC(int 装置番号)

装置番号                    OPEN の時に指定した、MCA-Lite の識別番号

### RETURN

- 1            命令送信の時にエラーが発生
- 2            データ受信の時にエラーが発生
- 0            正常終了

- MCALite\_ReadStatus() MCA-Lite の測定中停止中の状態を読み込みます。
- 

## int MCALite\_ReadStatus(int 装置番号)

装置番号                    OPEN の時に指定した、MCA-Lite の識別番号

### RETURN

- 1            命令送信の時にエラーが発生
- 2            データ受信の時にエラーが発生
- 0            測定は停止中
- 1            測定中

## ● MCALite\_ReadSPC() MCA-Lite 内部のスペクトルを、読み出します

**int MCALite\_ReadSPC(int 装置番号, int 読み開始 ch, char \* 読み領域)**

装置番号            OPEN の時に指定した、MCA-Lite の識別番号  
 読み開始ch        スペクトルの読み出し開始ch番号を指定する  
 装置番号            読み出したスペクトルの格納場所  
                     普通は、32ビットの整数の配列    int spc[4096] 等

## RETURN

-1            命令送信の時にエラーが発生  
 -2            データ受信の時にエラーが発生  
 0            測定は停止中  
 1            測定中

## 機能

一回の読み込みは、読み込み開始 ch から 256ch 分です。

## 記述例

```
int  l, rt;
int  spc[4096];            // Spectrum area
int  sch;

sch=0;
for(i=0; i<16; i++)
{
    rt=MCALite_ReadSPC(1, sch, (char*)&spc[sch]);
    if(rt<0)
    {
        ShowMessage("Spectrum read error");
        return;
    }
    sch += 256;
}
```

一挙に 4096ch のスペクトルを読み込んでいます。