

Lifetron

UsbIODLL.dll 関数説明書

2014/10/16

(株) ライフترون

URL:<http://www.lifetron.jp>

UsbIODLL.dll は旧 VB6.0、VC6.0 用に提供してきたものです。
現在の VisualStudio ではラッパーの DLL (UsbIOWrap.dll) を使用してこの UsbIODLL.dll にアクセスすることが出来ます。

VisualStudio の各言語からラッパー-DLL を参照し、ラッパー-DLL が UsbIODLL.dll をアクセスします。このためラッパー-DLL を使用する場合も関数仕様は変わりません。

目次

1. 関数一覧表	2
2. ユニットデバイス番号一覧表	3
3. 関数説明	
• UsbIO_Initial	4
• UsbIO_BrowseDevice	5
• UsbIO_BrowseDeviceIndex	6
• UsbIO_Open	7
• UsbIO_Close	8
• UsbIO_ReadDIO	9
• UsbIO_MultiReadDIO	10
• UsbIO_WriteDIO	11
• UsbIO_MultiWriteDIO	12
• UsbIO_SetChDirDIO	13
• UsbIO_ReadAIO	14
• UsbIO_MultiReadAIO	15
• UsbIO_WriteAIO	16
• UsbIO_MultiWriteAIO	17
• UsbIO_SetGainAIO	18
• UsbIO_ReadCOU	19
• UsbIO_WriteCOU	20
• UsbIO_SetModeCOU	21

1. 関数一覧表

区分	関数名	概要
共通	Usbio_initial	関数使用時の初期化処理
	UsbIO_BrowseDevice	接続されている利用可能なユニットの情報取得
	UsbIO_BrowseDeviceIndex	指定したユニットの情報取得
	UsbIO_Open	ユニットのオープン処理
	UsbIO_Close	ユニットのクローズ処理
接点入出力	UsbIO_ReadDIO	指定チャンネルの接点読込
	UsbIO_MultiReadDIO	複数の指定チャンネル接点読込
	UsbIO_WriteDIO	指定チャンネルへの接点指令書込み
	UsbIO_MultiWriteDIO	複数の指定チャンネルへの接点書込み
	UsbIO_SetChDirDIO	入出力ポートの入力/出力切替
アナログ入出力	UsbIO_ReadAIO	指定チャンネルのアナログ電圧の読込
	UsbIO_MultiReadAIO	複数の指定チャンネルのアナログ電圧の読込
	UsbIO_WriteAIO	指定チャンネルへのアナログ電圧出力指令
	UsbIO_MultiWriteAIO	複数のチャンネルへのアナログ電圧出力指令
	UsbIO_SetGainAIO	アナログ出力のゲイン指定
カウンタ入出力	UsbIO_ReadCOU	カウンタの読込
	UsbIO_WriteCOU	カウンタへの値の設定
	UsbIO_SetModeCOU	カウンタ動作モードの指定

2. ユニットデバイス番号一覧表

PCに接続される USBIO シリーズの各ユニットは

- ・ユニット名 (USB I/O デバイス名) とそれに対応するユニット番号 (UnitDevice 番号)、
- ・ユニットのロータリスイッチに設定されたユニット ID 番号 (UnitID 番号)

で特定されます。

関数を使用する上では、ユニット名称 (USB I/O Device 名) の代わりに UnitDevice 番号を使います。

以下は unitDevice 番号と USB I/O Device 名との対応表です。

Unit Device 番号	USB I/O Device 名
1	DI-16 (U)
2	DI-16T (U)
3	DIO-8/8 (U)
4	DIO-8/8T (U)
5	RO-16 (U)
6	DO-16 (U)
7	DO-16T (U)
8	DIO-24D (U)
9	AI-16 (U)
10	AO-4 (U)
11	COU-2 (U)
12	DI-64 (U)
13	DI-32 (U)
14	DO-64 (U)
15	DO-32 (U)
16	DIO-32/32 (U)
17	DIO-16/16 (U)

3. 関数説明

Usbio_initial

Short `Usbio_initial (LPCTSTR pszDevName)`

■引数:

`pszDevName` Device 名を格納した Buffer への Pointer を指定します。
Device 名は**"UsbIO**" を指定してください。

■戻り値:

`short` 利用可能な UNIT 数

■説明:

共通関数 USB I/O API 関数を使用するときに最初に実行する関数です。USB I/O API 関数を利用する場合は、必ず最初に、この関数を呼び出してください。

■書式

(C言語用)

```
LPCTSTR      pszDevName;  
Short        sRet;  
sRet=Usbio_initial (pszDevName);
```

(Visual Basic 用)

```
DIM            pszDevName            As        String  
DIM            wRet                    As        Integer  
wRet=Usbio_initial (pszDevName)
```

UsbIO_BrowseDevice

short `UsbIO_BrowseDevice(PUSBIUNIT pUsbIoUnit)`

■引数:

`pUsbIoUnit` 接続されている USB I/O 機器の情報を格納する Structure 配列への Pointer。
この引数に `**NULL**` を設定すると、利用可能な UNIT 数のみを返す。

■戻り値:

short 利用可能な UNIT 数

■説明:

共通関数
HOST に接続された利用可能な USB I/O 機器の一覧を取得することができます。この関数を利用する場合、最初に PUSBIUNIT に NULL を指定し関数を呼び出し、接続総数を取得します。その後、接続総数分の USBIUNIT 構造体を用意して再度、その Pointer を指定してこの関数を呼び出すことで、接続台数分の各情報を呼び出すことが可能です。

■書式

(C言語用)

```
PUSBIUNIT    pUsbIoUnit;  
int            sRet;  
sRet=UsbIO_BrowseDevice(pUsbIoUnit);
```

(Visual Basic用)

```
DIM            pszDevName            As        String  
DIM            wRet                    As        Integer  
wRet=UsbIO_BrowseDevice(pUsbIoUnit)
```

UsbIO_BrowseDeviceIndex

BOOL UsbIO_BrowseDeviceIndex(int iIndex, PUSBIOUNIT pUsbIoUnit)

■引数:

iIndex 取得したいUSB I/O 機器情報の格納されたListの登録Index
pstrUsbIO 接続されているUSB I/O 機器の情報を格納するStructure配列へのPointer。

■戻り値:

BOOL 関数が正常終了した場合は、TRUE。
異常終了した場合は、FALSEを返します。

■説明:

共通関数 指定Indexに従い、HOSTに接続された利用可能なUSB I/O 機器の各情報をPUSBIOUNITの示す構造体へのBuffer Pointerに格納します。
この関数は、VBからUSB I/Oの情報を取得する場合に使用できます。

■構造体:

pstrUsbIO で引き渡すUSBIOUNIT構造体の内容

```
BYTE      btDevNo ;           // USB I/O Device Unit Device 番号
BYTE      btUnitID ;         // USB I/O Device Unit ID 番号
BYTE      btSubID ;         // USB I/O Version No. ( Sub ID )
BYTE      btTotalCh ;       // USB I/O Device の総Channel 数
BYTE      btChSize ;        // USB I/O Device の1 Ch 当たり Bit 数
CHAR      szName[16] ;      // USB I/O Device Name
```

■書式

(C言語用)

```
int        iIndex;
PUSBIOUNIT pUsbIoUnit;
BOOL       bRet;
sRet=UsbIO_BrowseDeviceIndex(iIndex, pUsbIoUnit);
```

(Visual Basic用)

```
DIM        iIndex          As Integer
DIM        pUsbIoUnit      As PUSBIOUNIT
DIM        bRet            As Boolean
bRet=UsbIO_BrowseDeviceIndex(iIndex, pUsbIoUnit)
```

UsbIO_Open

HANDLE **UsbIO_Open**(**BYTE** btDevNo, **BYTE** btUnitID)

■引数 :

btDevNo 利用する USB I/O Device の UnitDevice 番号を指定します。
btUnitID 利用する USB I/O Device の UnitID 番号を指定します。

■戻り値 :

HANDLE 戻り値として **HANDLE** が返ります。
この関数の処理に失敗した場合は、**NULL** が返ります。
この **HANDLE** 値を、各 USB I/O Access 関数利用時に指定し、USB I/O への Access が終了したときには、必ず **UsbIO_Close()** 関数を使用して **HANDLE** を **Close** してください。

■説明 :

共通関数

利用したい Device の指定 (Unit Device 番号と Unit ID 番号を指定) を行い、**Open** 処理を行います。
ここに指定する Unit Device 番号や Unit ID 番号は、**UsbIO_Browse()** 関数や **UsbIO_BrowseDeviceIndex()** 関数で取得した Data を利用することができます。
以後、この関数の戻り値 (**HANDLE** 値) を使用して各種関数を利用します。
一連の処理が完了したときは、必ず **UsbIO_Close()** 関数により関数の後処理を行う必要があります。

■書式

(C言語用)

```
BYTE            btDevNo;  
BYTE            btUnitID;  
HANDLE          hHandle;  
hHandle=UsbIO_Open(btDevNo, btUnitID);
```

(Visual Basic用)

```
DIM            btDevNo            As    Byte  
DIM            btUnitID           As    Byte  
DIM            hHandle            As    Long  
hHandle=UsbIO_Open(btDevNo, btUnitID)
```


UsbIO_Close

BOOL UsbIO_Close(HANDLE hHandle)

■引数：

hHandle UsbIO_Open() 関数で取得した HANDLE を指定

■戻り値：

BOOL
関数が正常終了した場合は、TRUE。
異常終了した場合は、FALSE を返します。

■説明：

共通関数 UsbIO_Open() 関数で取得した HANDLE を利用して各種関数を利用後は必ず、この関数で、HANDLE を閉じてください。

■書式

(C言語用)

```
HANDLE hHandle;  
BOOL bRet;  
bRet=UsbIO_Close(hHandle);
```

(Visual Basic 用)

```
DIM hHandle As Long  
DIM bRet As Boolean  
bRet=UsbIO_Close(hHandle)
```

UsbIO_ReadDIO

WORD UsbIO_ReadDIO(HANDLE hHandle, BYTE btCH, PBYTE pbtData)

■引数:

hHandle UsbIO_Open() 関数の戻り値の HANDLE 番号
btCH Read したい DIO の Channel No.
pbtData 読み出した Data を格納する Byte Data Pointer

■戻り値:

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外の場合異常終了

■説明:

DIO 用処理 対象 DIO Device の指定 Channel から Byte Data を読み出します。
USB I/O series の DIO は、必ず 8 bit/Channel です。

■書式

(C言語用)

```
HANDLE hHandle;  
BYTE btCH;  
BYTE btData;  
WORD wRet;  
wRet=UsbIO_ReadDIO(hHandle, btCH, &btData);
```

(Visual Basic 用)

```
DIM hHandle As Long  
DIM btCH As Byte  
DIM btData As Byte  
DIM wRet As Integer  
wRet=UsbIO_ReadDIO(hHandle, btCH, btData)
```

UsbIO_MultiReadDIO

WORD UsbIO_MultiReadDIO(HANDLE hHandle, BYTE btLen, PBYTE pbtCH, PBYTE pbtBuff)

■引数 :

hHandle UsbIO_Open() 関数の戻り値の HANDLE 番号
btLen Channel の格納数
pbtCH 読み出す Channel の配列 Pointer
Channel は、必ず昇順で格納してください。
pbtBuff 読み出した Data を格納する Byte Data 配列への Buffer Pointer

■戻り値 :

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外の場合異常終了

■説明 :

DIO 用処理 対象 DIO Device から指定した複数 Channel から Byte Data を読み出します。このとき、pbtCH に格納する呼び出し対象 Channel は、昇順に格納してください。USB I/O series の DIO は、必ず 8 bit/Channel です。

■書式

(C言語用)

```
HANDLE    hHandle;  
BYTE      btLen;  
BYTE      btCH[n];  
BYTE      btBuff[n];  
WORD      wRet;  
wRet=UsbIO_MultiReadDIO(hHandle, btLen, &btCH[0], &btBuff[0])
```

(Visual Basic 用)

```
DIM      hHandle      As Long  
DIM      btLen        As Byte  
DIM      btCH(n)     As Byte  
DIM      btBuff(n)   As Byte  
DIM      wRet         As Integer  
wRet=UsbIO_MultiReadDIO(hHandle, btCH(0), btBuff(0))
```

UsbIO_WriteDIO

WORD `UsbIO_WriteDIO(HANDLE hHandle, BYTE btCH, BYTE btData, BYTE btMask)`

■引数 :

hHandle `UsbIO_Open()` 関数の戻り値の HANDLE 番号
btCH Write したい DIO の Channel No.
btData 書き込みたい Byte Data
btMask 書き込みたい Byte Data にかける Mask Byte Data
Mask Data は、対象 Channel のうち書き込みたい Bit だけを Mask するために使
用します。例えば、現在の Data に対し Bit 0 と Bit5 にのみ Handling したい場
合は、btMask には、0x21 を設定します。

■戻り値 :

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外のとき異常終了

■説明 :

DIO 用処理 対象 DIO Device の指定 Channel に Byte Data を書き込みます。
USB I/O series の DIO は、必ず 8 byte/Channel です。

■書式

(C言語用)

```
HANDLE      hHandle;  
BYTE        btCH;  
BYTE        btData;  
BYTE        btMask;  
WORD        wRet;  
wRet=UsbIO_WriteDIO(hHandle, btCH, btData, btMask);
```

(Visual Basic 用)

```
DIM          hHandle          As      Long  
DIM          btCH             As      Byte  
DIM          btData          As      Byte  
DIM          btMsk            As      Byte  
DIM          wRet             As      Integer  
wRet=UsbIO_WriteDIO(hHandle, btCH, btData, btMask)
```

UsbIO_MultiWriteDIO

WORD UsbIO_MultiWriteDIO (HANDLE hHandle, BYTE btLen, PBYTE pbtCH, PBYTE pbtBuff, PBYTE pbtMask)

■引数:

hHandle	UsbIO_Open() 関数の戻り値の HANDLE 番号
btLen	Channel の格納数
pbtCH	書き込む対象 Channel の配列 Pointer Channel は、必ず昇順で格納してください。
pbtBuff	書き込みたい Data を格納する Byte Data 配列への Buffer Pointer
pbtMask	書き込み Data にかける Mask を格納する Byte Data 配列への Buffer Pointer

■戻り値:

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外のとき異常終了

■説明:

DIO 用処理 対象 DIO Device に指定した複数 Channel に対し Mask Data をかけた Byte Data を書き込みます。

このとき、pbtCH に格納する呼び出し対象 Channel は、昇順に格納してください。
USB I/O series の DIO は、必ず 8 bit/Channel です。

■書式

(C言語用)

```
HANDLE    hHandle;
BYTE      btLen;
BYTE      btCH[n];
BYTE      btBuff[n];
BYTE      btMask[n];
WORD      wRet;
wRet=UsbIO_MultiWriteDIO(hHandle, &btCH, &btData, &btMask)
```

(Visual Basic 用)

```
DIM      hHandle As Long
DIM      btCH(n) As Byte
DIM      btData(n) As Byte
DIM      btMsk(n) As Byte
DIM      wRet As Integer
wRet=UsbIO_MultiWriteDIO(hHandle, btCH(0), btData(0), btMask(0))
```

UsbIO_SetChDirDIO

WORD UsbIO_SetChDirDIO(HANDLE hHandle, BYTE btLen, PBYTE pbtDir)

■引数:

hHandle UsbIO_Open()関数の戻り値のHANDLE番号

btLen 入出力を設定したいChannel数

pbtDir 各Channelの入出力方向を格納する配列Pointer

ここでは、Byte単位で入出力を設定し、btLenで示すChannel分のData配列を指定します。

DIOの入出力方向	設定値
IN (Read)方向に設定する時	USBIO_DIO_DIR_IS_READ
OUT (Write)方向に設定する時	USBIO_DIO_DIR_IS_WRITE

■戻り値:

WORD 関数 Result Status

0x0000の場合、正常終了、0x0000以外のとき異常終了

■説明:

DIO用処理 対象DIO Deviceの入出力方向を設定します。

現在のところこの関数の対象となるDIO機器は、DIO-24Dのみとなります。

入出力の指定は、下記のとおり

■書式

(C言語用)

HANDLE hHandle;

BYTE btLen;

BYTE btDir[n];

WORD wRet;

wRet=UsbIO_SetChDirDIO(hHandle, btLen, &btDir[0])

(Visual Basic用)

DIM hHandle As Long

DIM btCH(n) As Byte

DIM btData(n) As Byte

DIM btMsk(n) As Byte

DIM wRet As Integer

wRet=UsbIO_SetChDirDIO(hHandle, btLen, btDir(0))

UsbIO_ReadAIO

WORD UsbIO_ReadAIO(HANDLE hHandle, BYTE btCH, PLONG pIData)

■引数 :

hHandle UsbIO_Open() 関数の戻り値の HANDLE 番号
btCH Read したい AIO の Channel No.
pIData 読み出した Data を格納する Long Data Pointer

■戻り値 :

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外るとき異常終了

■説明 :

AI (A/D) 用処理 指定した Channel から読み出した Data を指定した Buffer に格納します。
AI (A/D) からの読み出し値は、符号付 LONG Data に拡張された値となります。

■書式

(C言語用)

```
HANDLE      hHandle;  
BYTE        btCH;  
LONG        IData;  
WORD        wRet;  
wRet=UsbIO_ReadAIO(hHandle, btCH, &IData);
```

(Visual Basic 用)

```
DIM          hHandle      As      Long  
DIM          btCH        As      Byte  
DIM          IData       As      Long  
DIM          wRet        As      Integer  
wRet=UsbIO_ReadAIO(hHandle, btCH, IData)
```

UsbIO_MultiReadAIO

WORD UsbIO_MultiReadAIO(HANDLE hHandle, BYTE btLen, PBYTE pbtCH, PLONG pData)

■引数:

hHandle UsbIO_Open() 関数の戻り値の HANDLE 番号
btLen Channel の格納数
pbtCH 読み出す Channel の配列 Pointer
Channel は、必ず昇順で格納してください。
pIData 読み出した Data を格納する Long Data 配列への Buffer Pointer

■戻り値:

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外の場合異常終了

■説明:

AI (A/D) 用処理 指定した複数 Channel から読み出した Data を指定した Long Data 配列 Buffer に btLen で示した Data 分格納します。
AI (A/D) からの読み出し値は、符号付 LONG Data に拡張された値となります。

■書式

(C言語用)

```
HANDLE    hHandle
BYTE      btLen
BYTE      btCH[n]
LONG      lData[n]
WORD      wRet
wRet=UsbIO_MultiReadDIO(hHandle, btLen, &btCH[0], &lData[0])
```

(Visual Basic 用)

```
DIM      hHandle      As      Long
DIM      btLen        As      Byte
DIM      btCH(n)      As      Byte
DIM      lData(n)     As      Long
DIM      wRet         As      Integer
wRet=UsbIO_MultiReadDIO(hHandle, btCH(0), lData(0))
```


UsbIO_WriteAIO

WORD UsbIO_WriteAIO(HANDLE hHandle, BYTE btCH, LONG lData)

■引数:

hHandle UsbIO_Open() 関数の戻り値のHANDLE 番号
btCH Write したいAIO のChannel No.
lData Write したいLong Data

■戻り値:

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外るとき異常終了

■説明:

AO (D/A) 用処理 指定したChannel に書き出したいLong Data を書き込みます。
AO (D/A) への書き出し値は、符号付 LONG Data に拡張された値を設定してください。

■書式

(C言語用)

HANDLE hHandle
BYTE btCH
LONG lData
WORD wRet
wRet=UsbIO_WriteAIO(hHandle, btCH, lData)

(Visual Basic用)

DIM hHandle As Long
DIM btCH As Byte
DIM lData As Long
DIM wRet As Integer

wRet=UsbIO_WriteAIO(hHandle, btCH, lData)

UsbIO_SetGainAIO

WORD UsbIO_SetGainAIO (HANDLE hHandle, BYTE btCH, BYTE btGain)

■引数 :

hHandle UsbIO_Open() の戻り値
btCH 設定したい対象 Channel
btGain 設定したい Gain Data

AIO に設定する GAIN 値	設定値
±1.25V	USBIO_AIO_GAIN_IS_1250
±2.50V	USBIO_AIO_GAIN_IS_2500
±5.00V	USBIO_AIO_GAIN_IS_5000
±10.00V	USBIO_AIO_GAIN_IS_10000

■戻り値 :

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外るとき異常終了

■説明 :

AIO 用処理 機器への Gain 設定を行います。
2002/07/07 現在、この関数は、A0 (D/A) Device にのみ設定可能です。指定 Channel No. は、0xff 固定です。

■書式

(C言語用)

HANDLE hHandle
BYTE btCH
BYTE btGain
WORD wRet
wRet=UsbIO_SetGainAIO (hHandle, btCH, btGain)

(Visual Basic 用)

DIM hHandle As Long
DIM btCH As Byte
DIM btGain As Byte
DIM wRet As Integer
wRet=UsbIO_SetGainAIO (hHandle, btCH, btGain)

UsbIO_ReadCOU

WORD **UsbIO_ReadCOU**(HANDLE hHandle, BYTE btCH, PLONG pData)

■引数：

hHandle UsbIO_Open() 関数の戻り値の HANDLE 番号
btCH Read したい Counter 値の対象 Channel No.
pIData Read した Data を格納する Long Data Pointer

■戻り値：

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外るとき異常終了

■説明：

COU 用処理 指定した Channel から読み出した Data を指定した Buffer に格納します。
COU からの読み出し値は、符号付 LONG Data に拡張された値となります。

(C言語用)

HANDLE hHandle
BYTE btCH
LONG dwData
WORD wRet
wRet=UsbIO_ReadCOU(hHandle, btCH, &IData)

(Visual Basic 用)

DIM hHandle As Long
DIM btCH As Byte
DIM dwData As Long
DIM wRet As Integer
wRet=UsbIO_ReadCOU(hHandle, btCH, IData)

UsbIO_WriteCOU

WORD UsbIO_WriteCOU(HANDLE hHandle, BYTE btCH, LONG lData)

■引数:

hHandle UsbIO_Open() 関数の戻り値の HANDLE 番号
btCH Write したい Counter 値の対象 Channel No.
lData 書き出したい Long Data

■戻り値:

WORD 関数 Result Status
0x0000 の場合、正常終了、0x0000 以外るとき異常終了

■説明:

COU 用処理 指定した Channel に書き出したい Long Data を書き込みます。
COU への書き出し値は、符号付 LONG Data に拡張された値を設定してください。

■書式

(C言語用)

HANDLE hHandle
BYTE btCH
LONG lData
WORD wRet
wRet=UsbIO_WriteCOU(hHandle, btCH, lData)

(Visual Basic 用)

DIM hHandle As Long
DIM btCH As Byte
DIM lData As Long
DIM wRet As Integer
wRet=UsbIO_WriteCOU(hHandle, btCH, lData)

UsbIO_SetModeCOU

WORD UsbIO_SetModeCOU(HANDLE hHandle, BYTE btCH, BYTE btMode, BYTE btFilter)

■引数：

hHandle UsbIO_Open() 関数の戻り値の HANDLE 番号
 btCH Mode 設定のの対象 Channel No.
 btMode Mode の設定値

Counter の動作 Mode	設定値
単純加減算	USBIO_COU_MODE_IS_0
A/B 相加減算	USBIO_COU_MODE_IS_1
A/B 相加減算 2 倍	USBIO_COU_MODE_IS_2
A/B 相加減算 4 倍	USBIO_COU_MODE_IS_3

btFilter Filter の設定値

Digital Filter の有効/無効	設定値
Filter を無効にする	USBIO_COU_FILTER_OFF
Filter を有効にする	USBIO_COU_FILTER_ON

■戻り値：

WORD 関数 Result Status
 0x0000 の場合、正常終了、0x0000 以外るとき異常終了

■説明：

COU 用処理 Counter 機器の Mode 設定、及び、Digital Filter の有効・無効の設定を行います。

■書式

(C言語用)

HANDLE hHandle
 BYTE btCH
 BYTE btMode
 BYTE btFilter
 WORD wRet
 wRet=UsbIO_SetModeCOU(hHandle, btCH, btMode, btFilter)

(Visual Basic 用)

DIM hHandle As Long
 DIM btCH As Byte
 DIM btMode As Byte
 DIM btFilter As Byte
 DIM wRet As Integer
 wRet=UsbIO_SetModeCOU(hHandle, btCH, btMode, btFilter)

以上