



STEPPING & SERVO MOTOR CONTROLLER'S OPTION

**MPL-28-02v1.00/USBW32**

**MPL-28-03v1.00/USBW64**

# 取扱説明書 (設計者用)

(デバイスドライバ AL MCC06 ユニット編)

CB-23/USB

C-772A

**USER'S MANUAL**

本製品を使用する前に、この取扱説明書を良く読んで十分に理解してください。  
この取扱説明書は、いつでも取り出して読めるように保管してください。

# 目次

---

1. 概要	4
2. 取扱説明書の構成	4
3. サポート環境	5
4. ご使用になる前に	6
4-1. DLLについて	6
4-2. この取扱説明書の見方	7
4-3. マルチプロセス対応	8
4-4. パソコンとI/Fユニット間の通信エラー	8
4-5. AL通信上の通信エラー	8
5. AL I/Fユニット対応DLL	9
5-1. 環境設定	9
5-2. 構造体と関数	9
RESULT構造体	10
環境設定ツール情報構造体	12
スレーブ情報構造体	13
環境設定ツール関数	14
環境設定直接関数	14
環境設定ツール情報読み出し関数	15
環境設定問い合わせ関数	15
AL通信エラー累計回数読み出し関数	16
AL通信エラー累計回数クリア関数	16
USB通信エラー累計回数読み出し関数	17
USB通信エラー累計回数クリア関数	17
スレーブ情報読み出し関数	18
6. AL MCC06ユニット対応DLL	19
6-1. デバイスの制御	19
6-2. 汎用I/O PORTの制御	19
6-3. モータ制御用I/Oの制御	20
6-4. 構造体と関数	21
RESULT構造体	22
データ構造体	24
デバイスオープン関数	25
デバイスクローズ関数	26
スレーブパラメータPORT書き込み関数	26
スレーブパラメータPORT読み出し関数	27
DRIVE COMMAND一括書き込み関数	27
DRIVE DATA PORT一括書き込み関数	28
DRIVE COMMAND PORT書き込み関数	28
DRIVE DATA1 PORT書き込み関数	29
DRIVE DATA2 PORT書き込み関数	29
DRIVE DATA3 PORT書き込み関数	30
STATUS1 PORT読み出し関数	30
STATUS2 PORT読み出し関数	31
STATUS3 PORT読み出し関数	31
STATUS4 PORT読み出し関数	32
STATUS5 PORT読み出し関数	32
DRIVE DATA PORT一括読み出し関数	33
DRIVE DATA1 PORT読み出し関数	33
DRIVE DATA2 PORT読み出し関数	34
DRIVE DATA3 PORT読み出し関数	34
READY WAIT関数	35
READY WAIT状態読み出し関数	35

READY WAIT中止関数 .....	36
COUNTER COMMAND一括書き込み関数 .....	36
COUNTER COMMAND PORT書き込み関数 .....	37
COUNTER DATA1 PORT書き込み関数 .....	37
COUNTER DATA2 PORT書き込み関数 .....	38
COUNTER DATA3 PORT書き込み関数 .....	38
汎用I/O PORTオープン関数 .....	39
汎用I/O PORTクローズ関数 .....	39
汎用I/O PORT一括読み出し関数 .....	40
汎用I/O PORT指定信号読み出し関数 .....	41
汎用I/O PORT一括書き込み関数 .....	42
汎用I/O PORT指定信号LOWレベル書き込み関数 .....	43
汎用I/O PORT指定信号HIGHレベル書き込み関数 .....	44
制御信号オープン関数 .....	45
制御信号クローズ関数 .....	45
制御信号読み出し関数 .....	46
制御信号指定ビット読み出し関数 .....	47
制御信号書き込み関数 .....	48
制御信号指定ビットON書き込み関数 .....	49
制御信号指定ビットOFF書き込み関数 .....	49
データセット1関数 .....	50
データセット2関数 .....	51
データセット3関数 .....	52
データゲット関数 .....	52
7. ソフト開発に必要なファイル .....	53
8. サンプルプログラム .....	54
8-1. 概要 .....	54
8-2. ご使用になる前に .....	54
8-3. 仕様 .....	55
9. プログラムツール .....	56
9-1. 概要 .....	56
9-2. ご使用になる前に .....	56
9-3. 使用方法 .....	56
10. トラブルシューティング .....	57
10-1. 環境設定ツールについて .....	57
10-2. ユーザーアプリケーション .....	59
本版で改訂された主な箇所 .....	60

## 1. 概要

MPL-28-02v1.00/USBW32 および MPL-28-03v1.00/USBW64 は、DOS/VパソコンのWindows上でCB-23/USB（ALシリーズ対応USB I/Fユニット）を使用し、ALシリーズ（弊社オリジナルステッピング&サーボモータコントローラシステム）上のスレーブユニットを動作させるためのDLLベースの関数ライブラリです。

MPL-28-02v1.00/USBW32 および MPL-28-03v1.00/USBW64 は、MPL-28-01v1.00/USBW32のバージョンアップ品です。

- ・ MPL-28-02v1.00/USBW32 は、Windows 32 ビット対応版です。
- ・ MPL-28-03v1.00/USBW64 は、Windows 64 ビット対応版です。
- ・ MPL-28-02v1.00/USBW32 と MPL-28-03v1.00/USBW64 の各関数は互換性があります。

各関数は、次に示すスレーブユニット上のPORTのアクセス（読み出し/書き込み）を行う為のものです。

スレーブユニット上のPORTについては、各スレーブユニットの取扱説明書を御覧ください。

◎C-771

◎CD-773A/ADB5F30

◎C-772A

◎CB-34/10

以降、MPL-28-02v1.00/USBW32およびMPL-28-03v1.00/USBW64は、MPL-28と称します。

## 2. 取扱説明書の構成

MPL-28は、次に示すようにユニットタイプで分類された取扱説明書が用意されていますので、ご使用になるユニットに該当する取扱説明書をご覧ください。

ユニット	ユニット	取扱説明書
I/Fユニット	パソコン-スレーブユニット間のインタフェースユニット <i>CB-23/USB</i>	注.
MCC05ユニット	パルスジェネレータMCC05を搭載したスレーブユニット <i>C-771</i> <i>CD-773A/ADB5F30</i>	<i>AL MCC05ユニット編</i>
MCC06ユニット	パルスジェネレータMCC06を搭載したスレーブユニット <i>C-772A</i>	<i>AL MCC06ユニット編</i> (当取扱説明書)
I/Oユニット	汎用I/Oポートを持つスレーブユニット <i>CB-34/10</i>	<i>AL I/Oユニット編</i>

注. AL MCC05ユニット編、AL MCC06ユニット編、AL I/Oユニット編いずれでも示しています。

## 3. サポート環境

R1

項目	MPL-28-02v1.00/USBW32	MPL-28-03v1.00/USBW64
サポートOS	<ul style="list-style-type: none"> <li>Microsoft Windows 8 (x86) *1</li> <li>Microsoft Windows 7 (x86)</li> <li>Microsoft Windows Vista (x86)</li> <li>Microsoft Windows XP (x86)</li> <li>Microsoft Windows 2000 Professional SP4</li> </ul>	<ul style="list-style-type: none"> <li>Microsoft Windows 8 (x64) *1</li> <li>Microsoft Windows 7 (x64)</li> <li>Microsoft Windows Vista (x64)</li> <li>Microsoft Windows XP Professional x64 Edition</li> </ul>
サポート言語	<ul style="list-style-type: none"> <li>Visual Basic .NET 2002, 2003, 2005, 2008, 2010, 2012</li> <li>Visual C# .NET 2002, 2003, 2005, 2008, 2010, 2012</li> <li>Visual C++ .NET 2002, 2003, 2005, 2008, 2010, 2012 *2</li> <li>Visual C++ 6.0</li> <li>Visual Basic 6.0</li> <li>C++ Builder 5.0</li> </ul>	<ul style="list-style-type: none"> <li>Visual Basic .NET 2005, 2008, 2010, 2012 (x64)</li> <li>Visual C# .NET 2005, 2008, 2010, 2012 (x64)</li> <li>Visual C++ .NET 2005, 2008, 2010, 2012 (x64)</li> </ul>
サポート製品	<p>《マスター》</p> <ul style="list-style-type: none"> <li>CB-23/USB (USB マスター)</li> </ul> <p>《MCC05スレーブユニット》</p> <ul style="list-style-type: none"> <li>C-771 (4 軸ステッピング/サーボ対応コントローラ:エンコーダ入力なし)</li> <li>CD-773A/ADB5F30 (コントローラドライバ:2 軸 5 相 1.4A/相)</li> </ul> <p>《MCC06スレーブユニット》</p> <ul style="list-style-type: none"> <li>C-772A (2 軸ステッピング/サーボ対応コントローラ:エンコーダ入力あり、汎用 I/O 8/8 点)</li> </ul> <p>《I/Oスレーブユニット》</p> <ul style="list-style-type: none"> <li>CB-34/IO (汎用 I/O 32/32 点)</li> </ul>	
サポート機種	<ul style="list-style-type: none"> <li>IBM PC/AT互換機</li> <li>DOS/V機</li> </ul>	
その他	<ul style="list-style-type: none"> <li>マスター同時使用可能数 : 1台</li> <li>マルチスレッド対応</li> <li>割り込み : 未使用</li> </ul>	

\*1 : Windows 8環境では、デスクトップアプリにのみ対応しています。

ストアアプリには対応していません。

\*2 : アンマネージコード対応です。

※ MPL-28-02v1.00/USBW32 と MPL-28-03v1.00/USBW64 を同一パソコンに同時にインストールすることはできません。

#### ■ 旧バージョンのデバイスドライバとの互換性について

MPL-28-02v1.00/USBW32とMPL-28-03v1.00/USBW64は、旧バージョンのデバイスドライバに対してソースレベルで互換ですが、バイナリレベルでは非互換です。

#### ● 旧バージョンのデバイスドライバ

- MPL-28/AL2USBWXP
- MPL-28-01v1.00/USBW32
- DH318/MPL-28/AL2USBWXP
- DH318-01/MPL28/USBW32

旧バージョンのデバイスドライバをMPL-28-02v1.00/USBW32またはMPL-28-03v1.00/USBW64に更新する際には、旧バージョンのデバイスドライバの関数定義ファイルをMPL-28-02v1.00/USBW32またはMPL-28-03v1.00/USBW64の関数定義ファイルに差し替え、ユーザアプリケーションを再構築する必要があります。

## 4. ご使用になる前に

### 4-1. DLLについて

MPL-28は、次に示すようにユニットタイプで分類された4つのDLLで構成されています。

ユーザアプリケーションは、使用するユニットに該当するDLLの関数を呼び出して、モータコントロール及び、汎用I/O PORTへのアクセスを行います。

よって、ユーザアプリケーションは、使用ユニットに対応したモジュール定義ファイルが必要になります。

詳細は、7. ソフト開発に必要なファイル を参照して下さい。

ユニット	ユニット	DLL名称
I/Fユニット	パソコン-スレーブユニット間のインタフェースユニット <i>CB-23/USB</i>	AL I/Fユニット対応DLL
MCC05ユニット	パルスジェネレータMCC05を搭載したスレーブユニット <i>C-771 CD-773A/ADB5F30</i>	AL MCC05ユニット対応DLL
MCC06ユニット	パルスジェネレータMCC06を搭載したスレーブユニット <i>C-772A</i>	AL MCC06ユニット対応DLL
I/Oユニット	汎用I/Oポートを持つスレーブユニット <i>CB-34/10</i>	AL I/Oユニット対応DLL



### 4-3. マルチプロセス対応

複数のユーザアプリケーションで同時に当DLLを使用することはできません。

### 4-4. パソコンとI/Fユニット間の通信エラー

USB通信I/Oエラーが発生した場合、環境設定ツール関数または環境設定直接関数を実行しない限り、エラーはクリアされず、USB通信は行われません。

これらのエラーが発生した場合、再度、環境設定ツール関数または環境設定直接関数を実行してください。

#### ● USB通信I/Oエラー

- ・ リクエストの最終バイト送信後、200ms経過しても、アンサーバックの先頭バイトを受信できない
- ・ アンサーバックの受信中に、受信データが得られないまま200ms以上経過
- ・ I/Fユニットから受信したデータに異常がある。

#### 【注意】

リトライ回数が0以外に設定されている場合、USB通信エラー、USBタイムアウトエラー発生後、リトライを行いますが、Windowsに割り込まれた場合、リトライ処理に遅れが生じます。

### 4-5. AL通信上の通信エラー

AL通信上の通信エラーが発生した場合、環境設定ツール関数または環境設定直接関数を実行しない限り、エラーはクリアされません。

AL通信上の通信エラーが発生した場合、再度、環境設定ツール関数または環境設定直接関数を実行してください。

※ AL通信エラーについては、CB-23/USBの取扱説明書をご覧ください。



## 5. AL I/Fユニット対応DLL

### 5-1. 環境設定

AL I/Fユニット対応DLLでは、リトライ回数、AL通信ポーレートを内部に記憶しています。

これらの情報のことを環境設定情報と称します。

MPL-28の各種ライブラリ関数を実行することによるUSB通信、ALシリーズ通信は、環境設定情報をもとに行われるため、ユーザアプリケーションは、次のいずれかの方法により、環境設定情報を設定しなければなりません。

この章では、AL I/Fユニット対応DLLを、単にI/F DLLと称します。

#### (1) 環境設定ツールによる設定

環境設定ツールは、画面上から手動操作により環境設定情報を設定するためのツールです。

環境設定ツールを使用した場合、環境設定情報に変更があっても、ユーザアプリケーションの修正と再コンパイルが不要になります。

環境設定ツールで設定した環境設定情報は、環境設定ツール関数でI/F DLL内部に取り込まれます。

この情報をもとに、USB通信のオープン、I/Fユニットの初期化が実行されるため、環境設定ツール関数は、ユーザアプリケーションの先頭で実行してください。

#### (2) ユーザアプリケーションによる直接設定

ユーザアプリケーションから環境設定直接関数を実行して、環境設定情報を指定します。

環境設定直接関数が実行されると、指定された環境設定情報は、I/F DLL内部に取り込まれます。

この情報をもとに、USB通信のオープン、I/Fユニットの初期化が実行されるため、環境設定直接関数は、ユーザアプリケーションの先頭で実行してください。

### 5-2. 構造体と関数

#### ● 構造体一覧

構造体名	説明
RESULT構造体	関数を実行した結果を格納
環境設定ツール情報構造体	環境設定情報を格納
スレーブ情報構造体	全スレーブのスレーブタイプを格納

#### ● 関数一覧

関数名	機能
環境設定ツール関数	環境設定ツールで設定した情報で環境設定
環境設定直接関数	設定値を直接指定して環境設定
環境設定ツール情報読み出し関数	環境設定ツールで設定された情報の読み出し
環境設定問い合わせ関数	すでに環境設定が行われているかの問い合わせ
AL通信エラー累計回数読み出し関数	AL通信上で発生したエラーの累計回数の読み出し
AL通信エラー累計回数クリア関数	AL通信エラー累計回数のクリア
USB通信エラー累計回数読み出し関数	USB通信エラーの累計回数の読み出し
USB通信エラー累計回数クリア関数	USB通信エラー累計回数のクリア
スレーブ情報読み出し関数	全スレーブのスレーブタイプの読み出し

## RESULT構造体

AL I/Fユニット対応DLL

CB-23/USB

## 説明

I/F DLLの関数を実行した結果が格納されます。

## 書式

**C言語**    `typedef struct ALK_TAG_S_RESULT {  
                  WORD ALK_Result[4];  
          } ALK_S_RESULT;`

**VB**        `Type ALK_S_RESULT  
              ALK_Result(1 To 4) As Integer  
          End Type`

**VB.NET**    `Structure ALK_S_RESULT  
              <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public ALK_Result() As Short  
              Public Sub Initialize()  
                  ReDim ALK_Result(3)  
              End Sub  
          End Structure`

**C#.NET**    `struct ALK_S_RESULT  
{  
    [MarshalAs( UnmanagedType.ByValArray, SizeConst=4 )] public ushort[] ALK_Result;  
    public ALK_S_RESULT( ushort dummy )  
    {  
        ALK_Result = new ushort[4];  
    }  
}`

## メンバ

次に示すメンバは、C言語で表記しています。C言語の`ALK_Result[0]~ALK_Result[3]`は、Visual Basicでは`ALK_Result(1)~ALK_Result(4)`、Visual Basic.NETでは`ALK_Result(0)~ALK_Result(3)`、Visual C#.NETでは`ALK_Result[0]~ALK_Result[3]`に対応します。

`ALK_Result[0]`    …… 実行された関数を示します。このメンバに格納される値は、次のいずれかです。

値	実行された関数	値	実行された関数
1	環境設定ツール関数	102	AL通信エラー累計回数読み出し関数
2	環境設定直接関数	103	AL通信エラー累計回数クリア関数
100	環境設定ツール情報読み出し関数	104	USB通信エラー累計回数読み出し関数
101	スレーブ情報読み出し関数	105	USB通信エラー累計回数クリア関数

値は10進表記です。

`ALK_Result[1]`    …… 実行結果を示します。このメンバに格納される値は、次のいずれかです。

値	実行結果
0	関数の実行が正常に終了しました
1	DLL内部でAPIエラーが発生しました
2	NULLポインタが指定されています
4	指定したパラメータが設定範囲を超えています
8	環境設定されていません
19	I/Fユニットが検出できません
20	USB通信時にI/Oエラーが発生しました
31	DLLが複数のプロセスから同時に使用されました

値は10進表記です。

*ALK\_Result[2]* …… I/Fユニット、スレーブユニット上で発生したエラーを示します。  
このメンバに格納される値は、次のいずれかです。

値	実行結果
0 (H' 00)	エラーは発生しませんでした
1~127 (H' 01~H' 7F)	スレーブユニット固有のエラーです。各スレーブの取扱説明書をご覧ください
128 (H' 80h)	初期化エラー（スレーブユニットが不正に電源OFFまたはRESETされました）
129 (H' 81h)	シリアルエラー（スレーブユニットからの受信時にエラーが発生しました）
130 (H' 82h)	タイムアウトエラー（スレーブユニットへの送信時にエラーが発生しました）

値は10進表記です。()内は、16進表記です。

*ALK\_Result[3]* …… 将来の拡張用です。

## 環境設定ツール情報構造体

AL I/Fユニット対応DLL

CB-23/USB

## 説明

環境設定情報が格納されます。

## 書式

```
C言語  typedef struct ALK_TAG_S_ENVIRONMENT_INFO {
        WORD  ALK_Ch1;
        WORD  ALK_Ch2;
        WORD  ALK_Ch3;
        WORD  ALK_Retry;
        WORD  ALK_RsBaudRate;
        WORD  ALK_AIBaudRate;
    } ALK_S_ENVIRONMENT_INFO;
```

```
VB      Type ALK_S_ENVIRONMENT_INFO
        ALK_Ch1 As Integer
        ALK_Ch2 As Integer
        ALK_Ch3 As Integer
        ALK_Retry As Integer
        ALK_RsBaudRate As Integer
        ALK_AIBaudRate As Integer
    End Type
```

```
VB.NET  Structure ALK_S_ENVIRONMENT_INFO
        Public ALK_Ch1 As Short
        Public ALK_Ch2 As Short
        Public ALK_Ch3 As Short
        Public ALK_Retry As Short
        Public ALK_RsBaudRate As Short
        Public ALK_AIBaudRate As Short
    End Structure
```

```
C#.NET  struct ALK_S_ENVIRONMENT_INFO
    {
        public ushort ALK_Ch1;
        public ushort ALK_Ch2;
        public ushort ALK_Ch3;
        public ushort ALK_Retry;
        public ushort ALK_RsBaudRate;
        public ushort ALK_AIBaudRate;
    }
```

## メンバ

*ALK\_Ch1* … ALK\_USBが格納されます。

*ALK\_Ch2* … 0が格納されます。

*ALK\_Ch3* … 0が格納されます。

*Retry* … 環境設定ツールで設定したリトライ回数が格納されます。0～3のいずれかになります。

*RsBaudRate* … このメンバに格納される値は、不定となります。

*AIBaudRate* … 環境設定ツールで設定したAL通信ボーレートが格納されます。  
このメンバに格納される値は、次のいずれかです。

値（識別子）	値（数値）	AL通信ボーレート
ALK_RATE_9765	0	9765bps
ALK_RATE_39062	1	39062bps
ALK_RATE_156250	2	156250bps
ALK_RATE_625000	3	625000bps

値（数値）は10進表記です。

## スレーブ情報構造体

AL I/Fユニット対応DLL

CB-23/USB

## 説明

全スレーブのスレーブタイプを格納します。

## 書式

```
C言語  typedef struct ALK_TAG_S_SLAVE_INFO {
        WORD  ALK_SlaveType[31];
    } ALK_S_SLAVE_INFO;
```

```
VB      Type ALK_S_SLAVE_INFO
        ALK_SlaveType(1 To 31) As Integer
    End Type
```

```
VB.NET  Structure ALK_S_SLAVE_INFO
        <MarshalAs(UnmanagedType.ByValArray, SizeConst:=31)> Public ALK_SlaveType() As Short
        Public Sub Initialize()
            ReDim ALK_SlaveType(30)
        End Sub
    End Structure
```

```
C#.NET  struct ALK_S_SLAVE_INFO
    {
        [MarshalAs(UnmanagedType.ByValArray, SizeConst = 31)] public ushort[] ALK_SlaveType;
        public MC07_S_SLAVE_INFO(ushort dummy)
        {
            ALK_SlaveType = new ushort[31];
        }
    }
}
```

## メンバ

次に示すメンバは、C言語で表記しています。C言語の *ALK\_SlaveType[0]~ALK\_SlaveType[30]* は、Visual Basicでは *ALK\_SlaveType(1)~ALK\_SlaveType(31)*、Visual Basic.NETでは *ALK\_SlaveType(0)~ALK\_SlaveType(30)*、Visual C#.NETでは *ALK\_SlaveType[0]~ALK\_SlaveType[30]* に対応します。

*ALK\_SlaveType[0]* …… スレーブアドレスH' 01に接続されているスレーブのスレーブタイプが格納されます。

…

*ALK\_SlaveType[30]* …… スレーブアドレスH' 1Fに接続されているスレーブのスレーブタイプが格納されます。スレーブが接続されていない場合、これらのメンバにはH' FFが格納されます。スレーブが接続されている場合、これらのメンバに格納される値は、次のいずれかです。

値（識別子）	値（数値）	スレーブタイプ
ALK_SLAVE_C771	H' 01	C-771
ALK_SLAVE_C772	H' 02	C-772A
ALK_SLAVE_CD773	H' 22	CD-773A/ADB5F30
ALK_SLAVE_CB34	H' 11	CB-34

## 環境設定ツール関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

環境設定ツールで設定した情報で環境設定を行います。

## 書式

**C言語** BOOL ALK\_EnvironmentInfo\_Tool (ALK\_S\_RESULT \**psResult*);

**VB** Function ALK\_EnvironmentInfo\_Tool (*psResult* As ALK\_S\_RESULT) As Boolean

**VB.NET** Function ALK\_EnvironmentInfo\_Tool (ByRef *psResult* As ALK\_S\_RESULT) As Boolean

**C#.NET** bool ALK.EnvironmentInfo\_Tool (ref ALK\_S\_RESULT *psResult*);

## 引数

*psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0)を返します。

## 環境設定直接関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

リトライ回数、USB通信ボーレート、AL通信ボーレートを直接指定して環境設定を行います。

## 書式

**C言語** BOOL ALK\_Environment\_Direct (WORD *Retry*, WORD *RsBaudRate*, WORD *AlBaudRate*, ALK\_S\_RESULT \**psResult*);

**VB** Function ALK\_Environment\_Direct (ByVal *Retry* As Integer, ByVal *RsBaudRate* As Integer, ByVal *AlBaudRate* As Integer, *psResult* As ALK\_S\_RESULT) As Boolean

**VB.NET** Function ALK\_Environment\_Direct (ByVal *Retry* As Short, ByVal *RsBaudRate* As Short, ByVal *AlBaudRate* As Short, ByRef *psResult* As ALK\_S\_RESULT) As Boolean

**C#.NET** bool ALK.Environment\_Direct (ushort *Retry*, ushort *RsBaudRate*, ushort *AlBaudRate*, ref ALK\_S\_RESULT *psResult*);

## 引数

*Retry* …… リトライ回数を指定します。0～3のいずれかになります。

*RsBaudRate* …… 必ず、NULLまたは0を指定します。

*AlBaudRate* …… AL通信ボーレートを指定します。

引数 <i>AlBaudRate</i> の値	AL通信ボーレート
ALK_RATE_9765	9765bps
ALK_RATE_39062	39062bps
ALK_RATE_156250	156250bps
ALK_RATE_625000	625000bps

*psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0)を返します。

## 環境設定ツール情報読み出し関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

環境設定ツールで設定された情報を読み出します。

## 書式

**C言語** BOOL ALK\_EnvironmentInfo\_Read(ALK\_S\_ENVIRONMENT\_INFO \*psEnvInfo, ALK\_S\_RESULT \*psResult);

**VB** Function ALK\_EnvironmentInfo\_Read(psEnvInfo As ALK\_S\_ENVIRONMENT\_INFO, psResult As ALK\_S\_RESULT) As Boolean

**VB.NET** Function ALK\_EnvironmentInfo\_Read(ByRef psEnvInfo As ALK\_S\_ENVIRONMENT\_INFO, ByRef psResult As ALK\_S\_RESULT) As Boolean

**C#.NET** bool ALK.EnvironmentInfo\_Read(ref ALK\_S\_ENVIRONMENT\_INFO psEnvInfo, ref ALK\_S\_RESULT psResult);

## 引数

- psEnvInfo* …… 環境設定ツールで設定された情報が格納される環境設定ツール情報構造体のポインタを指定します。
- psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## 関数環境設定問い合わせ関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

すでに環境設定が行われているかを問い合わせます。

## 書式

**C言語** BOOL ALK\_Environment\_Inquiry(VOID);

**VB** Function ALK\_Environment\_Inquiry() As Boolean

**VB.NET** Function ALK\_Environment\_Inquiry() As Boolean

**C#.NET** bool ALK.Environment\_Inquiry();

## 引数

この関数に、引数はありません。

## 戻り値

すでに環境設定が行われている場合はTRUE(1)、環境設定が行われていない場合はFALSE(0)を返します。

## AL通信エラー累計回数読み出し関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

AL通信上で発生したエラーの累計回数を読み出します。

## 書式

C言語 `BOOL ALK_Err_Count(WORD IfNo, WORD *pCount, ALK_S_RESULT *psResult);`

VB `Function ALK_Err_Count(ByVal IfNo As Integer, pCount As Integer, psResult As ALK_S_RESULT) As Boolean`

VB.NET `Function ALK_Err_Count(ByVal IfNo As Short, ByRef pCount As Short, ByRef psResult As ALK_S_RESULT) As Boolean`

C#.NET `bool ALK.Err_Count(ushort IfNo, ref ushort pCount, ref ALK_S_RESULT psResult);`

## 引数

*IfNo* … I/F番号を指定します。

マスターのボード／ユニット	引数 <i>IfNo</i> の値
CB-23/USB	ALK_USB

*pCount* … エラーの累計回数が格納される変数のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## AL通信エラー累計回数クリア関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

AL通信エラー累計回数を0にします。

## 書式

C言語 `BOOL ALK_Err_Clear(WORD IfNo, ALK_S_RESULT *psResult);`

VB `Function ALK_Err_Clear(ByVal IfNo As Integer, psResult As ALK_S_RESULT) As Boolean`

VB.NET `Function ALK_Err_Clear(ByVal IfNo As Short, ByRef psResult As ALK_S_RESULT) As Boolean`

C#.NET `bool ALK.Err_Clear(ushort IfNo, ref ALK_S_RESULT psResult);`

## 引数

*IfNo* … I/F番号を指定します。

マスターのボード／ユニット	引数 <i>IfNo</i> の値
CB-23/USB	ALK_USB

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。



## USB通信エラー累計回数読み出し関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

環境設定ツール関数、環境設定直接関数実行後、発生したUSB通信エラー累計回数（USB通信エラーとUSBタイムアウトエラーの累計回数）を読み出します。

カウントは最大で65535回までカウントし、その後エラーが発生してもカウントはストップします。

カウンタは環境設定ツール関数、環境設定直接関数、USB通信エラー累計回数クリア関数の実行によって0にクリアされます。

## 書式

**C言語** BOOL ALK\_Usb\_Err\_Count(WORD \*pCount, ALK\_S\_RESULT \*psResult);

**VB** Function ALK\_Usb\_Err\_Count(pCount As Integer, psResult As ALK\_S\_RESULT) As Boolean

**VB.NET** Function ALK\_Usb\_Err\_Count(ByRef pCount As Short, ByRef psResult As ALK\_S\_RESULT) As Boolean

**C#.NET** bool ALK.Usb\_Err\_Count(ref ushort pCount, ref ALK\_S\_RESULT psResult);

## 引数

*pCount* … エラーの累計回数が格納される変数のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## USB通信エラー累計回数クリア関数

AL I/Fユニット対応DLL

CB-23/USB

## 機能

USB通信エラー累計回数を0にします。

## 書式

**C言語** BOOL ALK\_Usb\_Err\_Clear(ALK\_S\_RESULT \*psResult);

**VB** Function ALK\_Usb\_Err\_Clear(psResult As ALK\_S\_RESULT) As Boolean

**VB.NET** Function ALK\_Usb\_Err\_Clear(ByRef psResult As ALK\_S\_RESULT) As Boolean

**C#.NET** bool ALK.Usb\_Err\_Clear(ref ALK\_S\_RESULT psResult);

## 引数

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## スレーブ情報読み出し関数

AL I/Fユニット対応DLL

CB-23/USB

## 機 能

スレーブアドレスH' 01～H' 1Fに接続されているスレーブのスレーブタイプを読み出します。

## 書 式

**C言語** BOOL ALK\_SlaveInfo\_Read(WORD *IfNo*, ALK\_S\_SLAVE\_INFO \**psSlaveInfo*, ALK\_S\_RESULT \**psResult*);

**VB** Function ALK\_SlaveInfo\_Read(ByVal *IfNo* As Integer, *psSlaveInfo* As ALK\_S\_SLAVE\_INFO, *psResult* As ALK\_S\_RESULT) As Boolean

**VB.NET** Function ALK\_SlaveInfo\_Read(ByVal *IfNo* As Short, ByRef *psSlaveInfo* As ALK\_S\_SLAVE\_INFO, ByRef *psResult* As ALK\_S\_RESULT) As Boolean

**C#.NET** bool ALK.SlaveInfo\_Read(ushort *IfNo*, ref ALK\_S\_SLAVE\_INFO *psSlaveInfo*, ref ALK\_S\_RESULT *psResult*);

## 引 数

*IfNo* … I/F番号を指定します。

マスターのボード／ユニット	引数 <i>IfNo</i> の値
CB-23/USB	ALK_USB

*psSlaveInfo* … スレーブ情報が格納されるスレーブ情報構造体のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 6. AL MCC06ユニット対応DLL

AL MCC06ユニット対応DLLは、パルスジェネレータMCC06を搭載したスレーブユニット上の、MCC06、ステッピング脱調検出部、汎用I/O PORT、モータ制御用I/Oを制御するための各種関数を備えています。

この章では、AL MCC06ユニット対応DLLを、単にMCC06 DLLと称します。

### 6-1. デバイスの制御

1軸分のMCC06とステッピング脱調検出部を組み合わせたものをデバイスと称します。  
デバイスを制御するための説明をします。

#### (1) デバイス上のPORTの説明

MCC06 DLLは、デバイス上のPORTを直接制御します。

スレーブ ユニット	デバイス上のPORT		
	スレーブ設定をするPORT	MCC06上のPORT	ステッピング脱調検出部のPORT
C-772A	スレーブパラメータPORT	DRIVE COMMAND PORT DRIVE DATA1 PORT DRIVE DATA2 PORT DRIVE DATA3 PORT STATUS1 PORT STATUS2 PORT STATUS3 PORT STATUS4 PORT STATUS5 PORT COUNTER COMMAND PORT COUNTER DATA1 PORT COUNTER DATA2 PORT COUNTER DATA3 PORT	なし

#### (2) デバイスのオープンとクローズ

デバイスを制御する前に、デバイスをオープンし、デバイスハンドルを取得します。  
以降、デバイスを制御する関数を実行する際に、このデバイスハンドルを指定します。  
デバイスハンドルは、デバイスをクローズするまで有効です。  
ユーザアプリケーション終了時は、必ずデバイスをクローズしてください。  
クローズが行われていないと、以降正常に動作しません。

### 6-2. 汎用I/O PORTの制御

スレーブユニット上の汎用I/O PORTを制御するための説明をします。

#### (1) 汎用I/O PORTの説明

MCC06 DLLは、汎用I/O PORTを直接制御します。

スレーブユニット	汎用I/O PORT
C-772A	あり

#### (2) 汎用I/O PORTのオープンとクローズ

汎用I/O PORTを制御する前に、汎用I/O PORTをオープンし、汎用I/O PORTハンドルを取得します。  
以降、汎用I/O PORTを制御する関数を実行する際に、この汎用I/O PORTハンドルを指定します。  
汎用I/O PORTハンドルは、汎用I/O PORTをクローズするまで有効です。  
ユーザアプリケーション終了時は、必ず汎用I/O PORTをクローズしてください。  
クローズが行われていないと、以降正常に動作しません。

### 6-3. モータ制御用I/Oの制御

スレーブユニット上のモータ制御用I/Oを制御するための説明をします。

#### (1) モータ制御用I/Oの説明

MCC06 DLLは、モータ制御用I/Oを直接制御します。

スレーブユニット	モータ制御用I/O
C-772A	制御I/O PORT

#### (2) 制御信号または制御I/O PORTのオープンとクローズ

モータ制御用I/Oを制御する前に、制御信号または制御I/O PORTをオープンし、制御信号ハンドルを取得します。

以降、制御信号または制御I/O PORTを制御する関数を実行する際に、この制御信号ハンドルを指定します。

制御信号ハンドルは、制御信号または制御I/O PORTをクローズするまで有効です。

ユーザアプリケーション終了時は、必ず制御信号または制御I/O PORTをクローズしてください。

クローズが行われていないと、以降正常に動作しません。

## 6-4. 構造体と関数

## ● 構造体一覧

構造体名	説明
RESULT構造体 データ構造体	関数を実行した結果を格納 デバイスに一括でアクセスするためのデータを格納

## ● 関数一覧

関数名	機能
デバイスオープン関数	デバイスのオープン
デバイスクローズ関数	デバイスのクローズ
スレーブパラメータPORT書き込み関数	スレーブパラメータPORTに書き込み
スレーブパラメータPORT読み出し関数	スレーブパラメータPORTの読み出し
DRIVE COMMAND一括書き込み関数	DRIVE COMMAND, DATA1~3 PORTに一括書き込み
DRIVE DATA PORT一括書き込み関数	DRIVE DATA1~3 PORTに一括書き込み
DRIVE COMMAND PORT書き込み関数	DRIVE COMMAND PORTに書き込み
DRIVE DATA1 PORT書き込み関数	DRIVE DATA1 PORTに書き込み
DRIVE DATA2 PORT書き込み関数	DRIVE DATA2 PORTに書き込み
DRIVE DATA3 PORT書き込み関数	DRIVE DATA3 PORTに書き込み
STATUS1 PORT読み出し関数	STATUS1 PORTの読み出し
STATUS2 PORT読み出し関数	STATUS2 PORTの読み出し
STATUS3 PORT読み出し関数	STATUS3 PORTの読み出し
STATUS4 PORT読み出し関数	STATUS4 PORTの読み出し
STATUS5 PORT読み出し関数	STATUS5 PORTの読み出し
DRIVE DATA PORT一括読み出し関数	DRIVE DATA1~3 PORTの一括読み出し
DRIVE DATA1 PORT読み出し関数	DRIVE DATA1 PORTの読み出し
DRIVE DATA2 PORT読み出し関数	DRIVE DATA2 PORTの読み出し
DRIVE DATA3 PORT読み出し関数	DRIVE DATA3 PORTの読み出し
READY WAIT関数	パルスジェネレータがREADYになるまで待機
READY WAIT状態読み出し関数	READY WAIT関数の状態の読み出し
READY WAIT中止関数	READY WAIT関数のREADY待ちを中止
COUNTER COMMAND一括書き込み関数	COUNTER COMMAND, DATA1~3 PORTに一括書き込み
COUNTER COMMAND PORT書き込み関数	COUNTER COMMAND PORTに書き込み
COUNTER DATA1 PORT書き込み関数	COUNTER DATA1 PORTに書き込み
COUNTER DATA2 PORT書き込み関数	COUNTER DATA2 PORTに書き込み
COUNTER DATA3 PORT書き込み関数	COUNTER DATA3 PORTに書き込み
汎用I/O PORTオープン関数	汎用I/O PORTのオープン
汎用I/O PORTクローズ関数	汎用I/O PORTのクローズ
汎用I/O PORT一括読み出し関数	汎用I/O PORTの状態を一括読み出し
汎用I/O PORT指定信号読み出し関数	汎用I/O PORTの指定信号の状態の読み出し
汎用I/O PORT一括書き込み関数	汎用I/O PORTに一括書き込み
汎用I/O PORT指定信号LOWレベル書き込み関数	汎用I/O PORTの指定信号にLOWレベルを書き込み
汎用I/O PORT指定信号HIGHレベル書き込み関数	汎用I/O PORTの指定信号にHIGHレベルを書き込み
制御信号オープン関数	制御信号のオープン
制御信号クローズ関数	制御信号のクローズ
制御信号読み出し関数	制御信号の状態を読み出し
制御信号指定ビット読み出し関数	制御信号の指定ビットの状態を読み出し
制御信号書き込み関数	制御信号に書き込み
制御信号指定ビットON書き込み関数	制御信号の指定ビットにONを書き込み
制御信号指定ビットOFF書き込み関数	制御信号の指定ビットにOFFを書き込み
データセット1関数	32ビットデータをデータ構造体に格納
データセット2関数	2つの24ビットデータをデータ構造体に格納
データセット3関数	32ビットデータを2つのデータ構造体に格納
データゲット関数	データ構造体のデータを32ビットデータに変換

## RESULT構造体

AL MCC06ユニット対応DLL

C-772A

## 説明

MCC06 DLLの関数を実行した結果が格納されます。

## 書式

```
C言語  typedef struct AC06_TAG_S_RESULT {
        WORD AC06_Result[4];
    } AC06_S_RESULT;
```

```
VB      Type AC06_S_RESULT
        AC06_Result(1 To 4) As Integer
    End Type
```

```
VB.NET Structure AC06_S_RESULT
        <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public AC06_Result() As Short
        Public Sub Initialize()
            ReDim AC06_Result(3)
        End Sub
    End Structure
```

```
C#.NET struct AC06_S_RESULT
{
    [MarshalAs( UnmanagedType.ByValArray, SizeConst=4 )] public ushort[] AC06_Result;
    public AC06_S_RESULT( ushort dummy )
    {
        AC06_Result = new ushort[4];
    }
}
```

## メンバ

次に示すメンバは、C言語で表記しています。C言語のAC06\_Result[0]~AC06\_Result[3]は、Visual BasicではAC06\_Result(1)~AC06\_Result(4)、Visual Basic.NETではAC06\_Result(0)~AC06\_Result(3)、Visual C#.NETではAC06\_Result[0]~AC06\_Result[3]に対応します。

AC06\_Result[0] …… 実行された関数を示します。このメンバに格納される値は、次のいずれかです。

値	実行された関数	値	実行された関数
10	デバイスオープン関数	70	制御信号オープン関数
11	デバイスクローズ関数	71	制御信号クローズ関数
20	DRIVE COMMAND PORT書き込み関数	72	制御信号読み出し関数
21	DRIVE DATA1 PORT書き込み関数	73	制御信号書き込み関数
22	DRIVE DATA2 PORT書き込み関数	74	制御信号指定ビット読み出し関数
23	DRIVE DATA3 PORT書き込み関数	75	制御信号指定ビットON書き込み関数
30	COUNTER COMMAND PORT書き込み関数	76	制御信号指定ビットOFF書き込み関数
31	COUNTER DATA1 PORT書き込み関数	80	READY WAIT関数
32	COUNTER DATA2 PORT書き込み関数	81	READY WAIT状態読み出し関数
33	COUNTER DATA3 PORT書き込み関数	82	READY WAIT中止関数
41	STATUS1 PORT読み出し関数	90	DRIVE COMMAND一括書き込み関数
42	STATUS2 PORT読み出し関数	91	DRIVE DATA PORT一括読み出し関数
43	STATUS3 PORT読み出し関数	92	COUNTER COMMAND一括書き込み関数
44	STATUS4 PORT読み出し関数	93	DRIVE DATA PORT一括書き込み関数
45	STATUS5 PORT読み出し関数	110	スレーブパラメータPORT書き込み関数
51	DRIVE DATA1 PORT読み出し関数	111	スレーブパラメータPORT読み出し関数
52	DRIVE DATA2 PORT読み出し関数		
53	DRIVE DATA3 PORT読み出し関数		
60	汎用I/O PORTオープン関数		
61	汎用I/O PORTクローズ関数		
62	汎用I/O PORT一括読み出し関数		
63	汎用I/O PORT指定信号読み出し関数		
64	汎用I/O PORT一括書き込み関数		
67	汎用I/O PORT指定信号LOWレベル書き込み関数		
68	汎用I/O PORT指定信号HIGHレベル書き込み関数		

値は10進表記です。

*AC06\_Result[1]* …… 実行結果を示します。このメンバに格納される値は、次のいずれかです。

値	実行結果
0	関数の実行が正常に終了しました
1	DLL内部でAPIエラーが発生しました
2	NULLポインタが指定されています
4	指定したパラメータが設定範囲を超えています
6	ハンドルの値が範囲外です
8	環境設定されていません
9	指定されたアドレスにスレーブが接続されていません
10	指定したスレーブタイプと実際に接続されているスレーブのタイプが異なります
12	指定されたデバイス、汎用I/O PORT、制御信号（制御I/O PORT）は、オープンされていません
13	指定されたデバイス、汎用I/O PORT、制御信号（制御I/O PORT）は、すでにオープンされています
14	READY WAIT関数がTIME OVERで終了しています
15	WM_QUITメッセージを受信しました
16	READY WAIT中にREADY WAIT中止関数が実行されました
17	同一デバイスのREADY WAIT関数が複数同時に実行されました
18	デバイス又は、I/O PORT数が最大値を超えています
30	I/FユニットDLLが見つかりません

値は10進表記です。

*AC06\_Result[2]* …… I/Fユニット、スレーブユニット上で発生したエラーを示します。  
このメンバに格納される値は、次のいずれかです。

値	実行結果
0	(H' 00) エラーは発生しませんでした
1~127	(H' 01~H' 7F) スレーブユニット固有のエラーです。各スレーブの取扱説明書をご覧ください
128	(H' 80h) 初期化エラー（スレーブユニットが不正に電源OFFまたはRESETされました）
129	(H' 81h) シリアルエラー（スレーブユニットからの受信時にエラーが発生しました）
130	(H' 82h) タイムアウトエラー（スレーブユニットへの送信時にエラーが発生しました）

値は10進表記です。()内は、16進表記です。

*AC06\_Result[3]* …… 将来の拡張用です。

## データ構造体

AL MCC06ユニット対応DLL

C-772A

## 説明

データを一括で読み書きするときに使用します。

## ● データを一括で読み書きするとき

- ・ DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORTのデータを一括で書き込むとき
- ・ DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORTのデータを一括で読み出すとき
- ・ COUNTER DATA1 PORT、COUNTER DATA2 PORT、COUNTER DATA3 PORTのデータを一括で書き込むとき

## 書式

```
C言語  typedef struct AC06_TAG_S_DATA {
        WORD AC06_Data[4];
    } AC06_S_DATA;
```

```
VB      Type AC06_S_DATA
        AC06_Data(1 To 4) As Integer
    End Type
```

```
VB.NET Structure AC06_S_DATA
        <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public AC06_Data() As Short
        Public Sub Initialize()
            ReDim AC06_Data(3)
        End Sub
    End Structure
```

```
C#.NET struct AC06_S_DATA
    {
        [MarshalAs( UnmanagedType.ByValArray, SizeConst=4 )] public ushort[] AC06_Data;
        public AC06_S_DATA( ushort dummy )
        {
            AC06_Data = new ushort[4];
        }
    }
```

## メンバ

次に示すメンバは、C言語で表記しています。C言語のAC06\_Data[0]～AC06\_Data[3]は、Visual BasicではAC06\_Data(1)～AC06\_Data(4)、Visual Basic.NETではAC06\_Data(0)～AC06\_Data(3)、Visual C#.NETではAC06\_Data[0]～AC06\_Data[3]に対応します。

AC06\_Data[0] …… DRIVE DATA1 PORT、COUNTER DATA1 PORTのいずれかの内容を格納します。  
 AC06\_Data[1] …… DRIVE DATA2 PORT、COUNTER DATA2 PORTのいずれかの内容を格納します。  
 AC06\_Data[2] …… DRIVE DATA3 PORT、COUNTER DATA3 PORTのいずれかの内容を格納します。  
 AC06\_Data[3] …… 将来の拡張用です。



## デバイスオープン関数

AL MCC06ユニット対応DLL

C-772A

## 機能

I/F番号、スレーブアドレス、軸、スレーブタイプを指定して、デバイスをオープンし、引数 $phDev$ で示される変数にデバイスハンドルを格納します。

## 書式

**C言語** BOOL AC06\_BOpen(WORD  $IfNo$ , WORD  $SlaveAddr$ , WORD  $Axis$ , WORD  $SlaveType$ , DWORD  $*phDev$ , AC06\_S\_RESULT  $*psResult$ );

**VB** Function AC06\_BOpen(ByVal  $IfNo$  As Integer, ByVal  $SlaveAddr$  As Integer, ByVal  $Axis$  As Integer, ByVal  $SlaveType$  As Integer,  $phDev$  As Long,  $psResult$  As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BOpen(ByVal  $IfNo$  As Short, ByVal  $SlaveAddr$  As Short, ByVal  $Axis$  As Short, ByVal  $SlaveType$  As Short, ByRef  $phDev$  As Integer, ByRef  $psResult$  As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BOpen(ushort  $IfNo$ , ushort  $SlaveAddr$ , ushort  $Axis$ , ushort  $SlaveType$ , ref uint  $phDev$ , ref AC06\_S\_RESULT  $psResult$ );

## 引数

$IfNo$  ... I/F番号を指定します。

マスターのボード／ユニット	引数 $IfNo$ の値
CB-09	ボード番号 (0~9)
CB-23/USB	AC06_USB

$SlaveAddr$  ... スレーブアドレスを指定します。H' 01~H' 1Fのいずれかになります。

$Axis$  ... 軸を指定します。

引数 $Axis$ の値	軸
AC06_X	X軸
AC06_Y	Y軸

$SlaveType$  ... スレーブタイプを指定します。

値	スレーブタイプ
AC06_SLAVE_C772	C-772A

$phDev$  ... デバイスハンドルが格納される変数のポインタを指定します。

$psResult$  ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

---

**デバイスクローズ関数**

AL MCC06ユニット対応DLL

---

**C-772A**


---

**機能**

指定されたデバイスをクローズします。

**書式**

**C言語**    `BOOL AC06_BClose(DWORD hDev, AC06_S_RESULT *psResult);`

**VB**        `Function AC06_BClose(ByVal hDev As Long, psResult As AC06_S_RESULT) As Boolean`

**VB.NET**   `Function AC06_BClose(ByVal hDev As Integer, ByRef psResult As AC06_S_RESULT) As Boolean`

**C#.NET**   `bool AC06.BClose(uint hDev, ref AC06_S_RESULT psResult);`

**引数**

*hDev*        … デバイスハンドルを指定します。

*psResult*   … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

**戻り値**

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

---

**スレーブパラメータPORT書き込み関数**

AL MCC06ユニット対応DLL

---

**C-772A**


---

**機能**

指定されたデバイスのスレーブパラメータPORTにデータを書き込みます。

**書式**

**C言語**    `BOOL AC06_BWSlavePara(DWORD hDev, WORD *pData, AC06_S_RESULT *psResult);`

**VB**        `Function AC06_BWSlavePara(ByVal hDev As Long, pData As Integer, psResult As AC06_S_RESULT) As Boolean`

**VB.NET**   `Function AC06_BWSlavePara(ByVal hDev As Integer, ByRef pData As Short, ByRef psResult As AC06_S_RESULT) As Boolean`

**C#.NET**   `bool AC06.BWSlavePara(uint hDev, ref ushort pData, ref AC06_S_RESULT psResult);`

**引数**

*hDev*        … デバイスハンドルを指定します。

*pData*       … 書き込むデータが格納されている変数のポインタを指定します。  
書き込むデータが格納されている変数は、下位8ビットが有効です。

*psResult*   … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

**戻り値**

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

---

## スレーブパラメータPORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのスレーブパラメータPORTを読み出します。

## 書式

**C言語** BOOL AC06\_BRSlavePara(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRSlavePara(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRSlavePara(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRSlavePara(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。

*pData* … 読み出した内容が格納される変数のポインタを指定します。  
読み出した内容は、下位8ビットが有効です。上位8ビットには、0が読み出されます。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE COMMAND一括書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのDRIVE COMMAND PORT、DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORTにコマンドコード、データを一括書き込みします。

## 書式

**C言語** BOOL AC06\_IWDrive(DWORD *hDev*, WORD *Cmd*, AC06\_S\_DATA \**psData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_IWDrive(ByVal *hDev* As Long, ByVal *Cmd* As Integer, *psData* As AC06\_S\_DATA, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_IWDrive(ByVal *hDev* As Integer, ByVal *Cmd* As Short, ByRef *psData* As AC06\_S\_DATA, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.IWDrive(uint *hDev*, ushort *Cmd*, ref AC06\_S\_DATA *psData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。

*Cmd* … 書き込むコマンドコードを指定します。

*psData* … 書き込むデータが格納されているデータ構造体のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE DATA PORT一括書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORTにデータを一括書き込みします。

## 書 式

**C言語** BOOL AC06\_IWData(DWORD *hDev*, AC06\_S\_DATA \**psData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_IWData(ByVal *hDev* As Long, *psData* As AC06\_S\_DATA, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_IWData(ByVal *hDev* As Integer, ByRef *psData* As AC06\_S\_DATA, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.IWData(uint *hDev*, ref AC06\_S\_DATA *psData*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*psData* … 書き込むデータが格納されているデータ構造体のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE COMMAND PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのDRIVE COMMAND PORTにコマンドコードを書き込みます。

## 書 式

**C言語** BOOL AC06\_BWDriveCommand(DWORD *hDev*, WORD \**pCmd*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWDriveCommand(ByVal *hDev* As Long, *pCmd* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWDriveCommand(ByVal *hDev* As Integer, ByRef *pCmd* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWDriveCommand(uint *hDev*, ref ushort *pCmd*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pCmd* … 書き込むコマンドコードが格納されている変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE DATA1 PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのDRIVE DATA1 PORTにデータを書き込みます。

## 書式

**C言語** BOOL AC06\_BWDriveData1(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWDriveData1(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWDriveData1(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWDriveData1(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。  
*pData* … 書き込むデータが格納されている変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE DATA2 PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのDRIVE DATA2 PORTにデータを書き込みます。

## 書式

**C言語** BOOL AC06\_BWDriveData2(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWDriveData2(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWDriveData2(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWDriveData2(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。  
*pData* … 書き込むデータが格納されている変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE DATA3 PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのDRIVE DATA3 PORTにデータを書き込みます。

## 書 式

**C言語** BOOL AC06\_BWDriveData3(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWDriveData3(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWDriveData3(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWDriveData3(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pData* … 書き込むデータが格納されている変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## STATUS1 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのSTATUS1 PORTを読み出します。

## 書 式

**C言語** BOOL AC06\_BRStatus1(DWORD *hDev*, WORD \**pStatus*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRStatus1(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRStatus1(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRStatus1(uint *hDev*, ref ushort *pStatus*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pStatus* … 読み出した内容が格納される変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## STATUS2 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのSTATUS2 PORTを読み出します。

## 書式

**C言語** BOOL AC06\_BRStatus2(DWORD *hDev*, WORD \**pStatus*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRStatus2(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRStatus2(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRStatus2(uint *hDev*, ref ushort *pStatus*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。  
*pStatus* … 読み出した内容が格納される変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## STATUS3 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのSTATUS3 PORTを読み出します。

## 書式

**C言語** BOOL AC06\_BRStatus3(DWORD *hDev*, WORD \**pStatus*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRStatus3(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRStatus3(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRStatus3(uint *hDev*, ref ushort *pStatus*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。  
*pStatus* … 読み出した内容が格納される変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## STATUS4 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのSTATUS4 PORTを読み出します。

## 書 式

**C言語** BOOL AC06\_BRStatus4(DWORD *hDev*, WORD \**pStatus*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRStatus4(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRStatus4(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRStatus4(uint *hDev*, ref ushort *pStatus*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pStatus* … 読み出した内容が格納される変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## STATUS5 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのSTATUS5 PORTを読み出します。

## 書 式

**C言語** BOOL AC06\_BRStatus5(DWORD *hDev*, WORD \**pStatus*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRStatus5(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRStatus5(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRStatus5(uint *hDev*, ref ushort *pStatus*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pStatus* … 読み出した内容が格納される変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。



## DRIVE DATA PORT一括読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのDRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORTを一括読み出しします。

## 書 式

**C言語** BOOL AC06\_IRDrive(DWORD *hDev*, AC06\_S\_DATA \**psData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_IRDrive(ByVal *hDev* As Long, *psData* As AC06\_S\_DATA, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_IRDrive(ByVal *hDev* As Integer, ByRef *psData* As AC06\_S\_DATA, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06\_IRDrive(uint *hDev*, ref AC06\_S\_DATA *psData*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*psData* … 読み出したデータが格納されるデータ構造体のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE DATA1 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのDRIVE DATA1 PORTを読み出します。

## 書 式

**C言語** BOOL AC06\_BRDriveData1(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRDriveData1(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRDriveData1(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06\_BRDriveData1(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pData* … 読み出した内容が格納される変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE DATA2 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのDRIVE DATA2 PORTを読み出します。

## 書式

**C言語** BOOL AC06\_BRDriveData2(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRDriveData2(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRDriveData2(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRDriveData2(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。

*pData* … 読み出した内容が格納される変数のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## DRIVE DATA3 PORT読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのDRIVE DATA3 PORTを読み出します。

## 書式

**C言語** BOOL AC06\_BRDriveData3(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BRDriveData3(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BRDriveData3(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BRDriveData3(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。

*pData* … 読み出した内容が格納される変数のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## READY WAIT関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスがREADY（STATUS1 PORTのBUSY BIT = 0）になるまで待機します。最大待ち時間を越えるとエラー終了します。

## 書式

**C言語** BOOL AC06\_BWaitDriveCommand(DWORD *hDev*, WORD *WaitTime*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWaitDriveCommand(ByVal *hDev* As Long, ByVal *WaitTime* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWaitDriveCommand(ByVal *hDev* As Integer, ByVal *WaitTime* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWaitDriveCommand(uint *hDev*, ushort *WaitTime*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* ... デバイスハンドルを指定します。  
*WaitTime* ... 最大待ち時間を1ms単位で設定します。0を指定するとREADYになるまで無限に待機します。  
*psResult* ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## READY WAIT状態読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのREADY WAIT関数の状態を返します。

## 書式

**C言語** BOOL AC06\_BIsWait(DWORD *hDev*, WORD \**pWaitSts*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BIsWait(ByVal *hDev* As Long, *pWaitSts* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BIsWait(ByVal *hDev* As Integer, ByRef *pWaitSts* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BIsWait(uint *hDev*, ref ushort *pWaitSts*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* ... デバイスハンドルを指定します。  
*pWaitSts* ... READY WAIT関数の状態が格納される変数のポインタを指定します。

格納される値	READY WAIT関数の状態
0	READYになるまで待機していない状態
1	READYになるまで待機している状態

*psResult* ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## READY WAIT中止関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのREADY WAIT関数のREADY待ちを中止します。

## 書式

**C言語** BOOL AC06\_BBreakWait(DWORD *hDev*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BBreakWait(ByVal *hDev* As Long, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BBreakWait(ByVal *hDev* As Integer, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BBreakWait(uint *hDev*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* ... デバイスハンドルを指定します。

*psResult* ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## COUNTER COMMAND一括書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのCOUNTER COMMAND PORT、COUNTER DATA1 PORT、COUNTER DATA2 PORT、COUNTER DATA3 PORTにコマンドコード、データを一括書き込みします。

## 書式

**C言語** BOOL AC06\_IWCounter(DWORD *hDev*, WORD *Cmd*, AC06\_S\_DATA \**psData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_IWCounter(ByVal *hDev* As Long, ByVal *Cmd* As Integer, *psData* As AC06\_S\_DATA, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_IWCounter(ByVal *hDev* As Integer, ByVal *Cmd* As Short, ByRef *psData* As AC06\_S\_DATA, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.IWCounter(uint *hDev*, ushort *Cmd*, ref AC06\_S\_DATA *psData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* ... デバイスハンドルを指定します。

*Cmd* ... 書き込むコマンドコードを指定します。

*psData* ... 書き込むデータが格納されているデータ構造体のポインタを指定します。

*psResult* ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## COUNTER COMMAND PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのCOUNTER COMMAND PORTにコマンドを書き込みます。

## 書式

**C言語** BOOL AC06\_BWCounterCommand(DWORD *hDev*, WORD \**pCmd*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWCounterCommand(ByVal *hDev* As Long, *pCmd* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWCounterCommand(ByVal *hDev* As Integer, ByRef *pCmd* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWCounterCommand(uint *hDev*, ref ushort *pCmd*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。

*pCmd* … 書き込むコマンドコードが格納されている変数のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## COUNTER DATA1 PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定されたデバイスのCOUNTER DATA1 PORTにデータを書き込みます。

## 書式

**C言語** BOOL AC06\_BWCounterData1(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWCounterData1(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWCounterData1(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWCounterData1(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hDev* … デバイスハンドルを指定します。

*pData* … 書き込むデータが格納されている変数のポインタを指定します。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## COUNTER DATA2 PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのCOUNTER DATA2 PORTにデータを書き込みます。

## 書 式

**C言語** BOOL AC06\_BWCounterData2(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWCounterData2(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWCounterData2(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWCounterData2(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pData* … 書き込むデータが格納されている変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## COUNTER DATA3 PORT書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

指定されたデバイスのCOUNTER DATA3 PORTにデータを書き込みます。

## 書 式

**C言語** BOOL AC06\_BWCounterData3(DWORD *hDev*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BWCounterData3(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BWCounterData3(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BWCounterData3(uint *hDev*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hDev* … デバイスハンドルを指定します。  
*pData* … 書き込むデータが格納されている変数のポインタを指定します。  
*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## 汎用I/O PORTオープン関数

AL MCC06ユニット対応DLL

C-772A

## 機能

I/F番号、スレーブアドレス、スレーブタイプを指定して、汎用I/O PORTをオープンし、引数 $phPort$ で示される変数に汎用I/O PORTハンドルを格納します。

## 書式

**C言語** BOOL AC06\_BPortOpen(WORD  $IfNo$ , WORD  $SlaveAddr$ , WORD  $SlaveType$ , DWORD  $*phPort$ , AC06\_S\_RESULT  $*psResult$ );

**VB** Function AC06\_BPortOpen(ByVal  $IfNo$  As Integer, ByVal  $SlaveAddr$  As Integer, ByVal  $SlaveType$  As Integer,  $phPort$  As Long,  $psResult$  As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BPortOpen(ByVal  $IfNo$  As Short, ByVal  $SlaveAddr$  As Short, ByVal  $SlaveType$  As Short, ByRef  $phPort$  As Integer, ByRef  $psResult$  As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BPortOpen(ushort  $IfNo$ , ushort  $SlaveAddr$ , ushort  $SlaveType$ , ref uint  $phPort$ , ref AC06\_S\_RESULT  $psResult$ );

## 引数

$IfNo$  ... I/F番号を指定します。

マスターのボード／ユニット	引数 $IfNo$ の値
CB-09	ボード番号 (0~9)
CB-23/USB	AC06_USB

$SlaveAddr$  ... スレーブアドレスを指定します。H' 01~H' 1Fのいずれかになります。

$SlaveType$  ... スレーブタイプを指定します。

引数 $SlaveType$ の値	スレーブタイプ
AC06_SLAVE_C772	C-772A

$phPort$  ... 汎用I/O PORTハンドルが格納される変数のポインタを指定します。

$psResult$  ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 汎用I/O PORTクローズ関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定された汎用I/O PORTをクローズします。

## 書式

**C言語** BOOL AC06\_BPortClose(DWORD  $hPort$ , AC06\_S\_RESULT  $*psResult$ );

**VB** Function AC06\_BPortClose(ByVal  $hPort$  As Long,  $psResult$  As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BPortClose(ByVal  $hPort$  As Integer, ByRef  $psResult$  As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BPortClose(uint  $hPort$ , ref AC06\_S\_RESULT  $psResult$ );

## 引数

$hPort$  ... 汎用I/O PORTハンドルを指定します。

$psResult$  ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 汎用I/O PORT一括読み出し関数

AL MCC06ユニット対応DLL

## G-772A

## 機 能

汎用I/O PORTの状態を一括読み出しします。

## 書 式

C言語 `BOOL AC06_BPortIn(DWORD hPort, DWORD *pData, AC06_S_RESULT *psResult);`VB `Function AC06_BPortIn(ByVal hPort As Long, pData As Long, psResult As AC06_S_RESULT) As Boolean`VB.NET `Function AC06_BPortIn(ByVal hPort As Integer, ByRef pData As Integer, ByRef psResult As AC06_S_RESULT) As Boolean`C#.NET `bool AC06.BPortIn(uint hPort, ref uint pData, ref AC06_S_RESULT psResult);`

## 引 数

- hPort* …… 汎用I/O PORTハンドルを指定します。  
*pData* …… 読み出した内容が格納される変数のポインタを指定します。  
 変数の内容は、次のようになります。

## ● G-772Aの汎用I/O PORTを読み出す場合

$2^{31}$	$2^{30}$	$2^{29}$	$2^{28}$	$2^{27}$	$2^{26}$	$2^{25}$	$2^{24}$	$2^{23}$	$2^{22}$	$2^{21}$	$2^{20}$	$2^{19}$	$2^{18}$	$2^{17}$	$2^{16}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
$\overline{\text{IN17}}$	$\overline{\text{IN16}}$	$\overline{\text{IN15}}$	$\overline{\text{IN14}}$	$\overline{\text{IN13}}$	$\overline{\text{IN12}}$	$\overline{\text{IN11}}$	$\overline{\text{IN10}}$	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

HIGHレベルのときに0、LOWレベルのときに1が読み出されます。

- psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0)を返します。



## 汎用I/O PORT指定信号読み出し関数

AL MCC06ユニット対応DLL

## C-772A

## 機能

汎用I/O PORTの指定信号の状態を読み出します。

## 書式

**C言語** BOOL AC06\_BSigIn(DWORD *hPort*, WORD *SigNo*, WORD \**pStatus*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BSigIn(ByVal *hPort* As Long, ByVal *SigNo* As Integer, *pStatus* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSigIn(ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *pStatus* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSigIn(uint *hPort*, ushort *SigNo*, ref ushort *pStatus*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hPort* …… 汎用I/O PORTハンドルを指定します。

*SigNo* …… 入力信号を指定します。

- C-772Aの汎用I/O PORTを読み出す場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
AC06_C772_IN10	$\overline{\text{IN10}}$	AC06_C772_OUT10	$\overline{\text{OUT10}}$
AC06_C772_IN11	$\overline{\text{IN11}}$	AC06_C772_OUT11	$\overline{\text{OUT11}}$
AC06_C772_IN12	$\overline{\text{IN12}}$	AC06_C772_OUT12	$\overline{\text{OUT12}}$
AC06_C772_IN13	$\overline{\text{IN13}}$	AC06_C772_OUT13	$\overline{\text{OUT13}}$
AC06_C772_IN14	$\overline{\text{IN14}}$	AC06_C772_OUT14	$\overline{\text{OUT14}}$
AC06_C772_IN15	$\overline{\text{IN15}}$	AC06_C772_OUT15	$\overline{\text{OUT15}}$
AC06_C772_IN16	$\overline{\text{IN16}}$	AC06_C772_OUT16	$\overline{\text{OUT16}}$
AC06_C772_IN17	$\overline{\text{IN17}}$	AC06_C772_OUT17	$\overline{\text{OUT17}}$

*pStatus* …… 指定信号の状態が格納される変数のポインタを指定します。

格納される値	指定信号の状態
AC06_LOW	LOWレベル
AC06_HIGH	HIGHレベル

*psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 汎用I/O PORT一括書き込み関数

AL MCC06ユニット対応DLL

G-772A

## 機 能

汎用I/O PORTにデータを一括書き込みします。

## 書 式

C言語 `BOOL AC06_BPortOut(DWORD hPort, DWORD *pData, AC06_S_RESULT *psResult);`VB `Function AC06_BPortOut(ByVal hPort As Long, pData As Long, psResult As AC06_S_RESULT) As Boolean`VB.NET `Function AC06_BPortOut(ByVal hPort As Integer, ByRef pData As Integer, ByRef psResult As AC06_S_RESULT) As Boolean`C#.NET `bool AC06.BPortOut(uint hPort, ref uint pData, ref AC06_S_RESULT psResult);`

## 引 数

*hPort* …… 汎用I/O PORTハンドルを指定します。  
*pData* …… 書き込むデータが格納されている変数のポインタを指定します。  
 変数の内容は、次のようになります。

## ● G-772Aの汎用I/O PORTに書き込む場合

$2^{31}$	$2^{30}$	$2^{29}$	$2^{28}$	$2^{27}$	$2^{26}$	$2^{25}$	$2^{24}$	$2^{23}$	$2^{22}$	$2^{21}$	$2^{20}$	$2^{19}$	$2^{18}$	$2^{17}$	$2^{16}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	0	0	0	0	0	0	0	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

HIGHレベルにするときに0、LOWレベルにするときに1を書き込みます。

*psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
 NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0)を返します。

## 汎用I/O PORT指定信号LOWレベル書き込み関数

AL MCC06ユニット対応DLL

## C-772A

## 機 能

汎用I/O PORTの指定信号をLOWレベルにします。

## 書 式

C言語 `BOOL AC06_BSigLowOut(DWORD hPort, WORD SigNo, AC06_S_RESULT *psResult);`VB `Function AC06_BSigLowOut(ByVal hPort As Long, ByVal SigNo As Integer, psResult As AC06_S_RESULT) As Boolean`VB.NET `Function AC06_BSigLowOut(ByVal hPort As Integer, ByVal SigNo As Short, ByRef psResult As AC06_S_RESULT) As Boolean`C#.NET `bool AC06.BSigLowOut(uint hPort, ushort SigNo, ref AC06_S_RESULT psResult);`

## 引 数

*hPort* …… 汎用I/O PORTハンドルを指定します。*SigNo* …… 出力信号を指定します。

- C-772Aの汎用I/O PORTに書き込む場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
AC06_C772_OUT10	OUT10	AC06_C772_OUT14	OUT14
AC06_C772_OUT11	OUT11	AC06_C772_OUT15	OUT15
AC06_C772_OUT12	OUT12	AC06_C772_OUT16	OUT16
AC06_C772_OUT13	OUT13	AC06_C772_OUT17	OUT17

*psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 汎用I/O PORT指定信号HIGHレベル書き込み関数

AL MCC06ユニット対応DLL

## C-772A

## 機 能

汎用I/O PORTの指定信号をHIGHレベルにします。

## 書 式

**C言語** BOOL AC06\_BSigHighOut(DWORD *hPort*, WORD *SigNo*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BSigHighOut(ByVal *hPort* As Long, ByVal *SigNo* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSigHighOut(ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSigHighOut(uint *hPort*, ushort *SigNo*, ref AC06\_S\_RESULT *psResult*);

## 引 数

*hPort* …… 汎用I/O PORTハンドルを指定します。

*SigNo* …… 出力信号を指定します。

- C-772Aの汎用I/O PORTに書き込む場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
AC06_C772_OUT10	OUT10	AC06_C772_OUT14	OUT14
AC06_C772_OUT11	OUT11	AC06_C772_OUT15	OUT15
AC06_C772_OUT12	OUT12	AC06_C772_OUT16	OUT16
AC06_C772_OUT13	OUT13	AC06_C772_OUT17	OUT17

*psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## 制御信号オープン関数

AL MCC06ユニット対応DLL

C-772A

## 機能

I/F番号、スレーブアドレス、スレーブタイプを指定して、制御信号または制御I/O PORTをオープンし、引数 $phSignal$ で示される変数に制御信号ハンドルを格納します。

## 書式

**C言語** BOOL AC06\_BSignalOpen(WORD  $IfNo$ , WORD  $SlaveAddr$ , WORD  $SlaveType$ , DWORD  $*phSignal$ , AC06\_S\_RESULT  $*psResult$ );

**VB** Function AC06\_BSignalOpen(ByVal  $IfNo$  As Integer, ByVal  $SlaveAddr$  As Integer, ByVal  $SlaveType$  As Integer,  $phSignal$  As Long,  $psResult$  As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSignalOpen(ByVal  $IfNo$  As Short, ByVal  $SlaveAddr$  As Short, ByVal  $SlaveType$  As Short, ByRef  $phSignal$  As Integer, ByRef  $psResult$  As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSignalOpen(ushort  $IfNo$ , ushort  $SlaveAddr$ , ushort  $SlaveType$ , ref uint  $phSignal$ , ref AC06\_S\_RESULT  $psResult$ );

## 引数

$IfNo$  ... I/F番号を指定します。

マスターのボード／ユニット	引数 $IfNo$ の値
CB-09	ボード番号 (0~9)
CB-23/USB	AC06_USB

$SlaveAddr$  ... スレーブアドレスを指定します。H' 01~H' 1Fのいずれかになります。

$SlaveType$  ... スレーブタイプを指定します。

引数 $SlaveType$ の値	スレーブタイプ
AC06_SLAVE_C772	C-772A

$phSignal$  ... 制御信号ハンドルが格納される変数のポインタを指定します。

$psResult$  ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 制御信号クローズ関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定された制御信号または制御I/O PORTをクローズします。

## 書式

**C言語** BOOL AC06\_BSignalClose(DWORD  $hSignal$ , AC06\_S\_RESULT  $*psResult$ );

**VB** Function AC06\_BSignalClose(ByVal  $hSignal$  As Long,  $psResult$  As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSignalClose(ByVal  $hSignal$  As Integer, ByRef  $psResult$  As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSignalClose(uint  $hSignal$ , ref AC06\_S\_RESULT  $psResult$ );

## 引数

$hSignal$  ... 制御信号ハンドルを指定します。

$psResult$  ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 制御信号読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定された制御信号または制御I/O PORTの状態を読み出します。

## 書式

C言語 `BOOL AC06_BSignalIn(DWORD hSignal, DWORD *pData, AC06_S_RESULT *psResult);`

VB `Function AC06_BSignalIn(ByVal hSignal As Long, pData As Long, psResult As AC06_S_RESULT) As Boolean`

VB.NET `Function AC06_BSignalIn(ByVal hSignal As Integer, ByRef pData As Integer, ByRef psResult As AC06_S_RESULT) As Boolean`

C#.NET `bool AC06.BSignalIn(uint hSignal, ref int pData, ref AC06_S_RESULT psResult);`

## 引数

*hSignal* …… 制御信号ハンドルを指定します。

*pData* …… 読み出した内容が格納される変数のポインタを指定します。  
変数の内容は、次のようになります。

## ● C-772Aの制御I/O PORTを読み出す場合

※スレーブパラメータPORT書き込み関数により、モータタイプをサーボモータに指定時は、 $\overline{\text{DRST}}/\overline{\text{OUT0}}$ 信号はDRST信号として割り付けられる為、OUT0は無効になります。

2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
0	0	0	0	0	0	0	0	0	0	YDALM	XDALM	$\overline{\text{YOUT1}}$	$\overline{\text{XOUT1}}$	$\overline{\text{YOUT0}}$	$\overline{\text{XOUT0}}$
														0	0

アクティブのときに1、ノットアクティブのときに0が読み出されます。

*psResult* …… この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0)を返します。

## 制御信号指定ビット読み出し関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定された制御信号または制御I/O PORTの指定ビットの状態を読み出します。

## 書式

**C言語** BOOL AC06\_BSignalBitIn(DWORD *hSignal*, WORD *SigNo*, WORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BSignalBitIn(ByVal *hSignal* As Long, ByVal *SigNo* As Integer, *pData* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSignalBitIn(ByVal *hSignal* As Integer, ByVal *SigNo* As Short, ByRef *pData* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSignalBitIn(uint *hSignal*, ushort *SigNo*, ref ushort *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hSignal* ... 制御信号ハンドルを指定します。

*SigNo* ... 制御信号のビットを指定します。

- C-772Aの制御I/O PORTを読み出す場合

引数 <i>SigNo</i> の値	ビット	引数 <i>SigNo</i> の値	ビット
AC06_C772_XOUT0	XOUT0	AC06_C772_YOUT0	YOUT0
AC06_C772_XOUT1	XOUT1	AC06_C772_YOUT1	YOUT1
AC06_C772_XDALM	XDALM	AC06_C772_YDALM	YDALM

*pData* ... 読み出した内容が格納される変数のポインタを指定します。  
変数には、次の値のいずれかが格納されます。

値	ビットの状態
AC06_ON	ON
AC06_OFF	OFF

*psResult* ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE (1)、エラーが発生したときはFALSE (0) を返します。

## 制御信号書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

指定された制御信号または制御I/O PORTにデータを書き込みます。

## 書式

**C言語** BOOL AC06\_BSignalOut(DWORD *hSignal*, DWORD \**pData*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BSignalOut(ByVal *hSignal* As Long, *pData* As Long, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSignalOut(ByVal *hSignal* As Integer, ByRef *pData* As Integer, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSignalOut(uint *hSignal*, ref uint *pData*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hSignal* … 制御信号ハンドルを指定します。

*pData* … 書き込むデータが格納される変数のポインタを指定します。  
変数の内容は、次のようになります。

## ● C-772Aの制御I/O PORTにデータを書き込む場合

※スレーブパラメータPORT書き込み関数により、モータタイプをサーボモータに指定時は、 $\overline{\text{DRST}}$ / $\overline{\text{OUT0}}$ 信号はDRST信号として割り付けられる為、OUT0は無効になります。

2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
0	0	0	0	0	0	0	0	0	0	0	0	$\overline{\text{YOUT1}}$	$\overline{\text{XOUT1}}$	$\overline{\text{YOUT0}}$	$\overline{\text{XOUT0}}$
														0	0

1を書き込むとアクティブ、0を書き込むとノットアクティブになります。

*psResult* … この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。



## 制御信号指定ビットON書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

制御信号または制御I/O PORTの指定されたビットをONにします。

## 書式

**C言語** BOOL AC06\_BSignalBitOnOut(DWORD *hSignal*, WORD *SigNo*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BSignalBitOnOut(ByVal *hSignal* As Long, ByVal *SigNo* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSignalBitOnOut(ByVal *hSignal* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSignalBitOnOut(uint *hSignal*, ushort *SigNo*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hSignal* ... 制御信号ハンドルを指定します。

*SigNo* ... 制御信号のビットを指定します。

- C-772Aの制御I/O PORTの指定ビットをONにする場合

引数 <i>SigNo</i> の値	ビット	引数 <i>SigNo</i> の値	ビット
AC06_C772_XOUT0	XOUT0	AC06_C772_YOUT0	YOUT0
AC06_C772_XOUT1	XOUT1	AC06_C772_YOUT1	YOUT1

*psResult* ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## 制御信号指定ビットOFF書き込み関数

AL MCC06ユニット対応DLL

C-772A

## 機能

制御信号または制御I/O PORTの指定されたビットをOFFにします。

## 書式

**C言語** BOOL AC06\_BSignalBitOffOut(DWORD *hSignal*, WORD *SigNo*, AC06\_S\_RESULT \**psResult*);

**VB** Function AC06\_BSignalBitOffOut(ByVal *hSignal* As Long, ByVal *SigNo* As Integer, *psResult* As AC06\_S\_RESULT) As Boolean

**VB.NET** Function AC06\_BSignalBitOffOut(ByVal *hSignal* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As AC06\_S\_RESULT) As Boolean

**C#.NET** bool AC06.BSignalBitOffOut(uint *hSignal*, ushort *SigNo*, ref AC06\_S\_RESULT *psResult*);

## 引数

*hSignal* ... 制御信号ハンドルを指定します。

*SigNo* ... 制御信号のビットを指定します。

- C-772Aの制御I/O PORTの指定ビットをOFFにする場合

引数 <i>SigNo</i> の値	ビット	引数 <i>SigNo</i> の値	ビット
AC06_C772_XOUT0	XOUT0	AC06_C772_YOUT0	YOUT0
AC06_C772_XOUT1	XOUT1	AC06_C772_YOUT1	YOUT1

*psResult* ... この関数を実行した結果が格納されるRESULT構造体のポインタを指定します。  
NULLポインタまたは0が指定されると、実行結果が格納されません。

## 戻り値

この関数を実行した結果、正常終了したときはTRUE(1)、エラーが発生したときはFALSE(0)を返します。

## データセット1関数

AL MCC06ユニット対応DLL

C-772A

## 機 能

32ビットデータを、次の形式で、データ構造体に格納します。

引数 $psData$ で示されるデータ構造体のメンバ $AC06\_Data[1]$ （C言語表記） [各種DATA2 PORTに対応]

2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
2 <sup>31</sup>	← 引数 $Data$ の上位16ビット (2 <sup>31</sup> ~2 <sup>16</sup> ) →														2 <sup>16</sup>

引数 $psData$ で示されるデータ構造体のメンバ $AC06\_Data[2]$ （C言語表記） [各種DATA3 PORTに対応]

2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
2 <sup>15</sup>	← 引数 $Data$ の下位16ビット (2 <sup>15</sup> ~2 <sup>0</sup> ) →														2 <sup>0</sup>

## 書 式

C言語 VOID AC06\_SetData1(DWORD  $Data$ , AC06\_S\_DATA \* $psData$ );

VB Sub AC06\_SetData1(ByVal  $Data$  As Long,  $psData$  As AC06\_S\_DATA)

VB.NET Sub AC06\_SetData1(ByVal  $Data$  As Integer, ByRef  $psData$  As AC06\_S\_DATA)

C#.NET void AC06.SetData1(int  $Data$ , ref AC06\_S\_DATA  $psData$ );

## 引 数

$Data$  … 32ビットのデータを指定します。

$psData$  … データ構造体のポインタを指定します。

## 戻り値

この関数に、戻り値はありません。

## データセット2関数

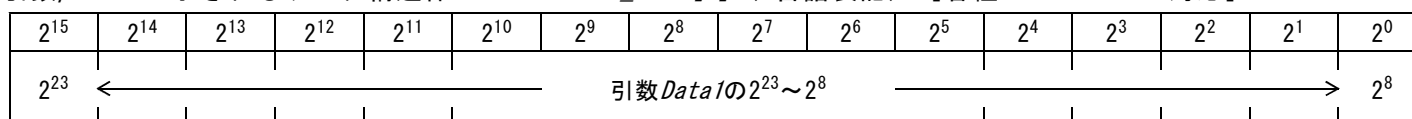
AL MCC06ユニット対応DLL

C-772A

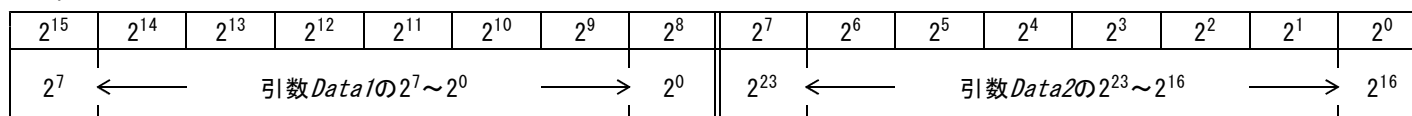
## 機能

2つの24ビットデータを、次の形式で、データ構造体に格納します。

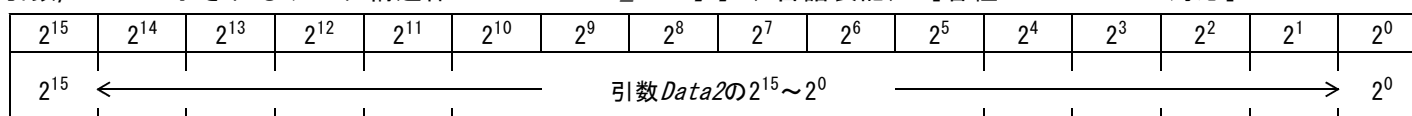
引数 $psData$ で示されるデータ構造体のメンバ $AC06\_Data[0]$ （C言語表記） [各種DATA1 PORTに対応]



引数 $psData$ で示されるデータ構造体のメンバ $AC06\_Data[1]$ （C言語表記） [各種DATA2 PORTに対応]



引数 $psData$ で示されるデータ構造体のメンバ $AC06\_Data[2]$ （C言語表記） [各種DATA3 PORTに対応]



## 書式

**C言語** VOID AC06\_SetData2(DWORD  $Data1$ , DWORD  $Data2$ , AC06\_S\_DATA \* $psData$ );

**VB** Sub AC06\_SetData2(ByVal  $Data1$  As Long, ByVal  $Data2$  As Long,  $psData$  As AC06\_S\_DATA)

**VB.NET** Sub AC06\_SetData2(ByVal  $Data1$  As Integer, ByVal  $Data2$  As Integer, ByRef  $psData$  As AC06\_S\_DATA)

**C#.NET** void AC06.SetData2(int  $Data1$ , int  $Data2$ , ref AC06\_S\_DATA  $psData$ );

## 引数

$Data1$  … 24ビットのデータを指定します。下位24ビットが有効です。

$Data2$  … 24ビットのデータを指定します。下位24ビットが有効です。

$psData$  … データ構造体のポインタを指定します。

## 戻り値

この関数に、戻り値はありません。

## データセット3関数

AL MCC06ユニット対応DLL

C-772A

## 機能

32ビットデータを、次の形式で、2つのデータ構造体に格納します。

引数 $psData1$ で示されるデータ構造体のメンバ $AC06\_Data[0]$ （C言語表記） [各種DATA1 PORTに対応]

2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
2 <sup>31</sup>	← 引数 $Data$ の上位16ビット (2 <sup>31</sup> ~2 <sup>16</sup> ) →														2 <sup>16</sup>

引数 $psData2$ で示されるデータ構造体のメンバ $AC06\_Data[0]$ （C言語表記） [各種DATA1 PORTに対応]

2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
2 <sup>15</sup>	← 引数 $Data$ の下位16ビット (2 <sup>15</sup> ~2 <sup>0</sup> ) →														2 <sup>0</sup>

## 書式

C言語 `VOID AC06_SetData3(DWORD Data, AC06_S_DATA *psData1, AC06_S_DATA *psData2);`

VB `Sub AC06_SetData3(ByVal Data As Long, psData1 As AC06_S_DATA, psData2 As AC06_S_DATA)`

VB.NET `Sub AC06_SetData3(ByVal Data As Integer, ByRef psData1 As AC06_S_DATA, ByRef psData2 As AC06_S_DATA)`

C#.NET `void AC06.SetData3(int Data, ref AC06_S_DATA psData1, ref AC06_S_DATA psData2);`

## 引数

$Data$  ... 32ビットのデータを指定します。  
 $psData1$  ... データ構造体のポインタを指定します。  
 $psData2$  ... データ構造体のポインタを指定します。

## 戻り値

この関数に、戻り値はありません。

## データゲット関数

AL MCC06ユニット対応DLL

C-772A

## 機能

データ構造体の内容を、次の形式で、32ビットデータに変換します。

変換後の32ビットデータ

2 <sup>31</sup>	2 <sup>30</sup>	2 <sup>29</sup>	2 <sup>28</sup>	2 <sup>27</sup>	2 <sup>26</sup>	2 <sup>25</sup>	2 <sup>24</sup>	2 <sup>23</sup>	2 <sup>22</sup>	2 <sup>21</sup>	2 <sup>20</sup>	2 <sup>19</sup>	2 <sup>18</sup>	2 <sup>17</sup>	2 <sup>16</sup>
2 <sup>15</sup>	← 引数 $psData$ で示されるデータ構造体のメンバ $AC06\_Data[1]$ （C言語表記） [各種DATA2 PORTに対応] →														2 <sup>0</sup>
2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
2 <sup>15</sup>	← 引数 $psData$ で示されるデータ構造体のメンバ $AC06\_Data[2]$ （C言語表記） [各種DATA3 PORTに対応] →														2 <sup>0</sup>

## 書式

C言語 `DWORD AC06_GetData(AC06_S_DATA *psData);`

VB `Function AC06_GetData(psData As AC06_S_DATA) As Long`

VB.NET `Function AC06_GetData(ByRef psData As AC06_S_DATA) As Integer`

C#.NET `int AC06.GetData(ref AC06_S_DATA psData);`

## 引数

$psData$  ... データ構造体のポインタを指定します。

## 戻り値

32ビットに変換されたデータを返します。

## 7. ソフト開発に必要なファイル

ユーザアプリケーション開発に必要なファイルは、インストール時に指定する次のフォルダに格納されています。  
（インストール時にパスを¥Program Files指定した場合）

- AL I/Fユニット対応DLLを用いたソフト開発に必要なファイル

MPL-28-02v1.00/USBW32

ファイル	ファイルの場所
Visual C++ヘッダファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vc¥AlUsbA.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vc¥Vc¥AlUsbA.lib
C++ Builderヘッダファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥AlUsbA.h
C++ Builderライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥Bc¥AlUsbA.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb.NET¥AlUsbA.vb
Visual Basic関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb¥AlUsbA.bas
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥C#.Net¥AlUsbA.cs

MPL-28-03v1.00/USBW64

ファイル	ファイルの場所
Visual C++ヘッダファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vc¥AlUsbA.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vc¥Vc¥AlUsbA.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vb.NET¥AlUsbA.vb
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥C#.Net¥AlUsbA.cs

- AL I/Oユニット対応DLLを用いたソフト開発に必要なファイル

AL I/Oユニット取扱説明書をご覧ください

- AL MCC05ユニット対応DLLを用いたソフト開発に必要なファイル

AL MCC05ユニット取扱説明書をご覧ください

- AL MCC06ユニット対応DLLを用いたソフト開発に必要なファイル

MPL-28-02v1.00/USBW32

ファイル	ファイルの場所
Visual C++ヘッダファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vc¥Ac06A.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vc¥Vc¥Ac06A.lib
C++ Builderヘッダファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥Ac06A.h
C++ Builderライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥Bc¥Ac06A.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb.NET¥Ac06A.vb
Visual Basic関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb¥Ac06A.bas
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥C#.Net¥Ac06A.cs

MPL-28-03v1.00/USBW64

ファイル	ファイルの場所
Visual C++ヘッダファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vc¥Ac06A.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vc¥Vc¥Ac06A.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vb.NET¥Ac06A.vb
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥C#.Net¥Ac06A.cs

## 8. サンプルプログラム

### 8-1. 概要

MPL-28-02v1.00/USBW32には、Visual C++ .NET, Visual C++, C++ Builder, Visual Basic .NET, Visual Basic, Visual C# .NETのサンプルプログラムが用意されています。

MPL-28-03v1.00/USBW64には、Visual C++ .NET, Visual Basic .NET, Visual C# .NETのサンプルプログラムが用意されています。

サンプルプログラムのファイルは、インストール時に指定する次のフォルダに格納されています。  
（インストール時にパスをProgram Filesと指定した場合）

#### MPL-28-02v1.00/USBW32

言語	ファイルの場所
Visual C++ .NET, Visual C++ C++ Builder	¥Program Files¥Mpl28_02v1.00¥Sample¥x86¥Vc
Visual Basic .NET	¥Program Files¥Mpl28_02v1.00¥Sample¥x86¥Vb.NET
Visual Basic	¥Program Files¥Mpl28_02v1.00¥Sample¥x86¥Vb
Visual C# .NET	¥Program Files¥Mpl28_02v1.00¥Sample¥x86¥C#.Net

#### MPL-28-03v1.00/USBW64

言語	ファイルの場所
Visual C++ .NET	¥Program Files¥Mpl28_03v1.00¥Sample¥x64¥Vc
Visual Basic .NET	¥Program Files¥Mpl28_03v1.00¥Sample¥x64¥Vb.NET
Visual C# .NET	¥Program Files¥Mpl28_03v1.00¥Sample¥x64¥C#.Net

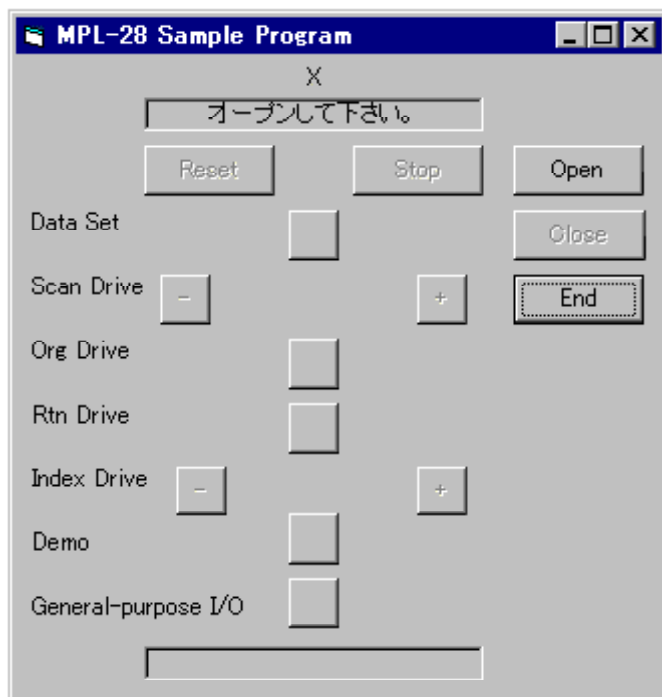
### 8-2. ご使用になる前に

環境設定ツールでAL通信ポーレート、リトライ回数を設定して下さい。

パソコン、CB-23/USB、C-772A(SLAVE ADDRESS = H'01)を接続して下さい。

## 8-3. 仕様

サンプルプログラムは、Visual C++ 6.0、C++ Builder 5.0、Visual Basic .NET 2002、Visual Basic 6.0、Visual C# .NET 2002で作成したものを用意してあります。これらは、同じ仕様で作られています。サンプルプログラムを参照する場合には、それぞれの言語の開発環境からプロジェクトを開いて下さい。



Openボタン	デバイス (C-772A X軸) をオープンします。
Closeボタン	デバイス (C-772A X軸) をクローズします。
Endボタン	サンプルプログラムを終了します。
Resetボタン	ADDRESS COUNTERを0にPRESETします。
Stopボタン	DRIVEを即時停止します。
Data Setボタン	次の設定にします。 RATE TYPE : L1-TYPE URATE : 10ms/1000Hz      LSPD : 1000Hz DRATE : 10ms/1000Hz      HSPD : 5000Hz
Scan Drive +ボタン	+(CW)方向へSCAN DRIVEします。
Scan Drive -ボタン	-(CCW)方向へSCAN DRIVEします。
Org Driveボタン	機械原点検出形式ORG-3でORIGIN DRIVEを行います。
Rtn Driveボタン	絶対ADDRESS 0へ移動するABSOLUTE INDEX DRIVEを行います。
Index Drive +ボタン	+(CW)方向へ3000パルス移動するINCREMENTAL INDEX DRIVEを行います。
Index Drive -ボタン	-(CCW)方向へ3000パルス移動するINCREMENTAL INDEX DRIVEを行います。
Demoボタン	次の動作を連続して行います。 機械原点の検出 (ORIGIN DRIVE) 電気原点の設定 (ADDRESS COUNTERを0にPRESET) +(CW)方向へ4000パルス移動を4回繰り返す (INCREMENTAL INDEX DRIVE) 絶対ADDRESS 30000へ移動 (ABSOLUTE INDEX DRIVE) 電気原点へ移動 (ABSOLUTE INDEX DRIVE)

## 9. プログラムツール

### 9-1. 概要

MPL-28-02v1.00/USBW32及びMPL-28-03v1.00/USBW64には、スレーブ情報確認プログラム、バージョン情報、環境設定ツール、信号チェックプログラムのプログラムツールが用意されています。  
プログラムツールは、インストール時に指定する次のフォルダに格納されています。  
(インストール時にパスをProgram Filesと指定した場合)

#### MPL-28-02v1.00/USBW32

ツール	ツールの場所
スレーブ情報確認プログラム	¥Program Files¥Mpl28_02v1.00¥SlaveCheck.exe ¥Program Files¥Mpl28_02v1.00¥SlaveCheck.chm
バージョン情報	¥Program Files¥Mpl28_02v1.00¥VerInfo.exe
環境設定ツール	¥Program Files¥Mpl28_02v1.00¥EnvTool.exe ¥Program Files¥Mpl28_02v1.00¥EnvTool.chm
信号チェックプログラム	¥Program Files¥Mpl28_02v1.00¥SignalCheck.exe ¥Program Files¥Mpl28_02v1.00¥SignalCheck.chm

#### MPL-28-03v1.00/USBW64

ツール	ツールの場所
スレーブ情報確認プログラム	¥Program Files¥Mpl28_03v1.00¥SlaveCheck.exe ¥Program Files¥Mpl28_03v1.00¥SlaveCheck.chm
バージョン情報	¥Program Files¥Mpl28_03v1.00¥VerInfo.exe
環境設定ツール	¥Program Files¥Mpl28_03v1.00¥EnvTool.exe ¥Program Files¥Mpl28_03v1.00¥EnvTool.chm
信号チェックプログラム	¥Program Files¥Mpl28_03v1.00¥SignalCheck.exe ¥Program Files¥Mpl28_03v1.00¥SignalCheck.chm

### 9-2. ご使用になる前に

詳細は、各ツールのヘルプボタンによるヘルプを参照して下さい。

### 9-3. 使用方法

詳細は、各ツールのヘルプボタンによるヘルプを参照して下さい。



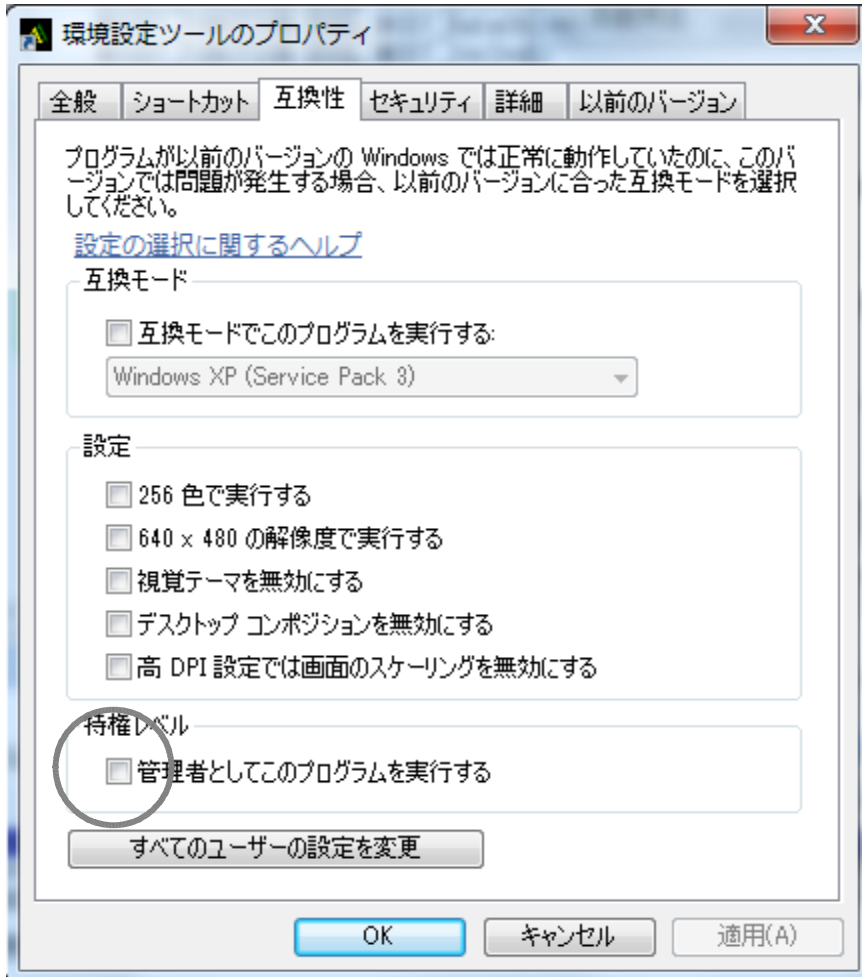
## 10. トラブルシューティング

### 10-1. 環境設定ツールについて

Windows Vista / 7 / 8において、環境設定ツールでレジストリオープンエラーが発生して正常に動作しない場合、次のことを行って下さい。

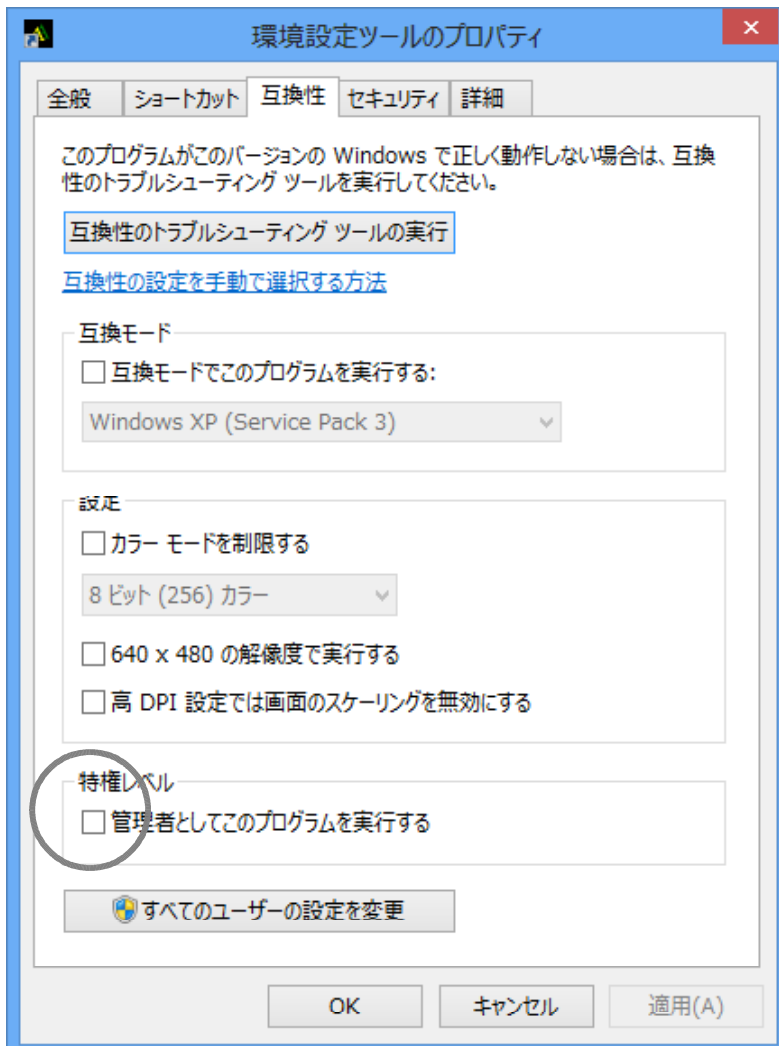
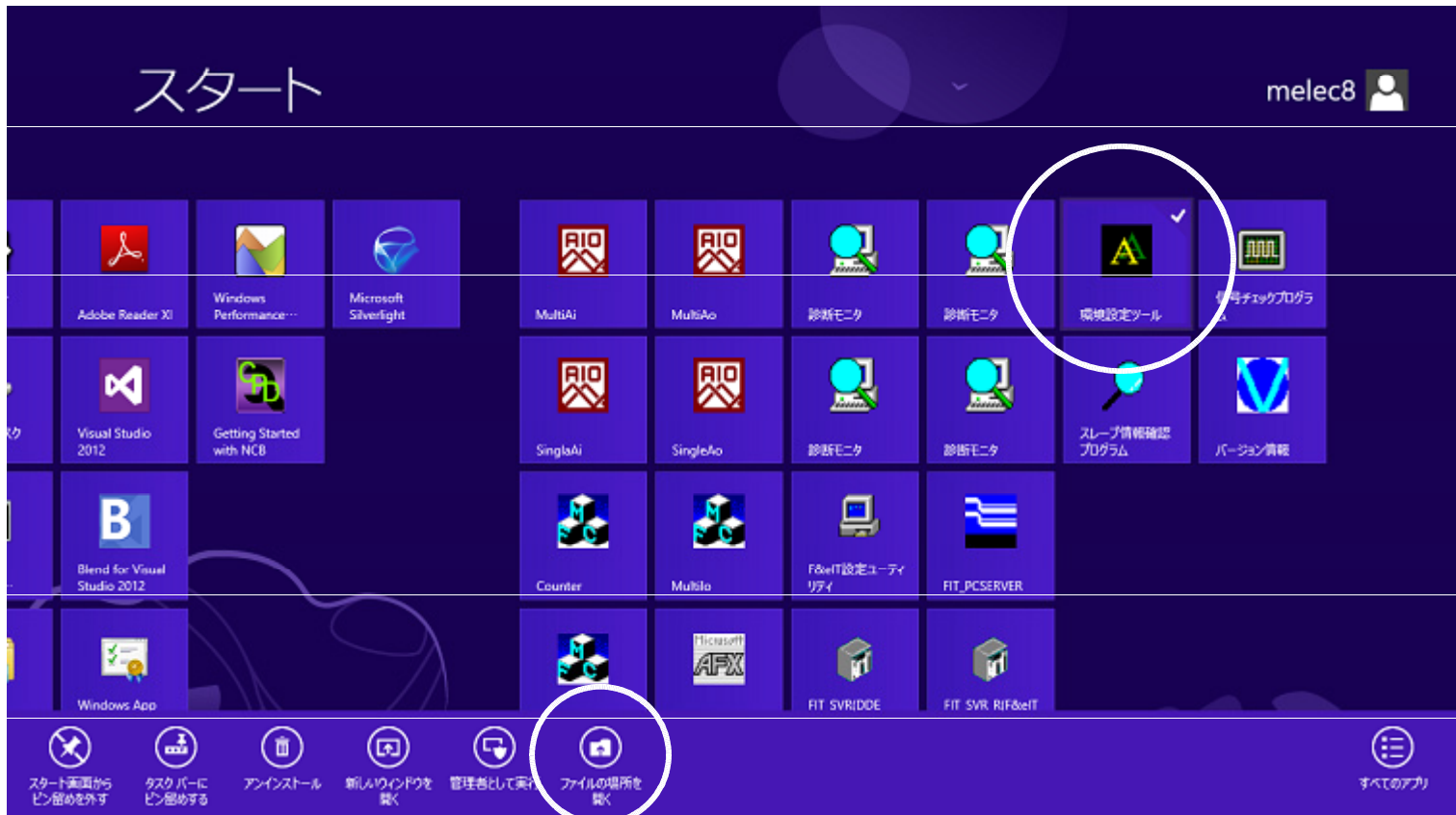
(1) Windows Vista / 7

- ① [スタートメニュー] から、[すべてのプログラム] → [MPL-28-02v1.00] または [MPL-28-03v1.00] → [環境設定ツール] を右クリックし、プロパティを開きます。
- ② [互換性] のタブを選択し、[特権レベル] 項目内の [管理者としてこのプログラムを実行する] のチェックボックスをチェックします。
- ③ [OK] をクリックして終了します。



## (2) Windows 8

- ① スタート画面にある[環境設定ツール]アイコン上で右クリックし、画面下側にサブメニューを表示します。サブメニューの中から[ファイルの場所を開く]を選択します。
- ② 各ツールのショートカットが格納されているフォルダが開かれますので、その中から[環境設定ツール]を右クリックし、プロパティを開きます。
- ③ 互換性のタブを選択し、[特権レベル]項目内の[管理者としてこのプログラムを実行する]のチェックボックスをチェックします。
- ④ [OK]をクリックして終了します。



## 10-2. ユーザーアプリケーション

作成したアプリケーションプログラムが正常に動作しない場合、次のことを行って下さい。

(1) I/Fユニットの認識

スレーブ情報確認プログラムを起動して下さい。

スレーブ情報確認プログラムは、起動時にI/Fユニットの認識を確認しています。

(2) スレーブの認識

スレーブ情報確認プログラムは、実際にUSB通信、AL通信を行い、各スレーブアドレスに接続されているスレーブを検出し、検出結果を表示しています。

メッセージ	スレーブの検出結果
Unconnection	スレーブが接続されていないことを示しています。
C-771	C-771が接続されていることを示しています。
CD-773A/ADB5F30	CD-773A/ADB5F30が接続されていることを示しています。
C-772A	C-772Aが接続されていることを示しています。
CB-34	CB-34が接続されていることを示しています。

(3) RESULT構造体の確認

各関数は、アプリケーションプログラムによって与えられた引数の内容をチェックし、エラーがある場合は、FALSE (0) を返し、正常である場合はTRUE (1) を返します。

関数が正常に動作していないと思われるステップの後にブレークポイントを設定し、関数が返した値がTRUE (1) であることを確認して下さい。

TRUE (1) でない場合は、エラー原因を特定する為にRESULT構造体の内容を参照して下さい。

## 本版で改訂された主な箇所

箇所	内容
P5	【R1】 旧バージョンのデバイスドライバとの互換性について 追加

---

## ■ 製品保証

### 保証期間と保証範囲について

- 納入品の保証期間は、納入後1ヶ年と致します。
- 上記保証期間中に当社の責により故障を生じた場合は、その修理を当社の責任において行います。  
(日本国内のみ)

ただし、次に該当する場合は、この保証対象範囲から除外させていただきます。

- (1) お客様の不適切な取り扱い、ならびに使用による場合。
- (2) 故障の原因が、当製品以外からの事由による場合。
- (3) お客様の改造、修理による場合。
- (4) 製品出荷当時の科学・技術水準では予見が不可能だった事由による場合。
- (5) その他、天災、災害等、当社の責にない場合。

(注1) ここでいう保証は、納入品単体の保証を意味するもので、納入品の故障により誘発される損害はご容赦頂きます。

(注2) 当社において修理済みの製品に関しましては、保証外とさせていただきます。

---

## 技術相談のお問い合わせ

TEL. (042) 664-5382 FAX. (042) 666-5664  
E-mail [s-support@melec-inc.com](mailto:s-support@melec-inc.com)

---

## 販売に関するお問い合わせ

TEL. (042) 664-5384 FAX. (042) 666-2031

株式会社 **メレック** 制御機器営業部  
〒193-0834 東京都八王子市東浅川町516-10

URL:<http://www.melec-inc.com>