



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 I/O ユニット編)

CB-23/USB

CB-34/IO

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の使用上の注意	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 I/Oユニット対応DLL	19
6-1. 汎用I/O PORTの制御	19
6-2. 構造体と関数	20
RESULT構造体	21
汎用I/O PORTオープン関数	23
汎用I/O PORTクローズ関数	24
汎用I/O PORT一括読み出し関数	25
汎用I/O PORT指定信号読み出し関数	26
汎用I/O PORT一括書き込み関数	28
汎用I/O PORT AND書き込み関数	29
汎用I/O PORT OR書き込み関数	30
汎用I/O PORT指定信号LOWレベル書き込み関数	31
汎用I/O PORT指定信号HIGHレベル書き込み関数	32
ラッチ機能選択書き込み関数	33
ラッチエッジ選択書き込み関数	34
ラッチ機能選択読み出し関数	35
ラッチエッジ選択読み出し関数	36
ラッチクリア書き込み関数	37
7. ソフト開発に必要なファイル	38
8. サンプルプログラム	39
8-1. 概要	39
8-2. ご使用になる前に	39
8-3. 仕様	40

9. プログラムツール	41
9-1. 概要	41
9-2. ご使用になる前に	41
9-3. 使用方法	41
10. トラブルシューティング	42
10-1. 環境設定ツールについて	42
10-2. ユーザーアプリケーション	44
本版で改訂された主な箇所	45

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 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>
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> (当取扱説明書)

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"> Microsoft Windows 8 (x86) *1 Microsoft Windows 7 (x86) Microsoft Windows Vista (x86) Microsoft Windows XP (x86) Microsoft Windows 2000 Professional SP4 	<ul style="list-style-type: none"> Microsoft Windows 8 (x64) *1 Microsoft Windows 7 (x64) Microsoft Windows Vista (x64) Microsoft Windows XP Professional x64 Edition
サポート言語	<ul style="list-style-type: none"> Visual Basic .NET 2002, 2003, 2005, 2008, 2010, 2012 Visual C# .NET 2002, 2003, 2005, 2008, 2010, 2012 Visual C++ .NET 2002, 2003, 2005, 2008, 2010, 2012 *2 Visual C++ 6.0 Visual Basic 6.0 C++ Builder 5.0 	<ul style="list-style-type: none"> Visual Basic .NET 2005, 2008, 2010, 2012 (x64) Visual C# .NET 2005, 2008, 2010, 2012 (x64) Visual C++ .NET 2005, 2008, 2010, 2012 (x64)
サポート製品	<p>《マスター》</p> <ul style="list-style-type: none"> CB-23/USB (USB マスター) <p>《MCC05スレーブユニット》</p> <ul style="list-style-type: none"> 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) CD-773A/ADB5F30 (コントローラドライバ:2 軸 5 相 1.4A/相) <p>《MCC06スレーブユニット》</p> <ul style="list-style-type: none"> 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) <p>《I/Oスレーブユニット》</p> <ul style="list-style-type: none"> CB-34/IO (汎用 I/O 32/32 点) 	
サポート機種	<ul style="list-style-type: none"> IBM PC/AT互換機 DOS/V機 	
その他	<ul style="list-style-type: none"> マスター同時使用可能数 : 1台 マルチスレッド対応 割り込み : 未使用 	

*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
        Dim ALK_Ch1 As Short
        Dim ALK_Ch2 As Short
        Dim ALK_Ch3 As Short
        Dim ALK_Retry As Short
        Dim ALK_RsBaudRate As Short
        Dim 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 I/Oユニット対応DLL

AL I/Oユニット対応DLLは、I/Oコントロール専用のスレーブユニット上の汎用I/OPORT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 I/Oユニット対応DLLを、単にI/O DLLと称します。

6-1. 汎用I/O PORTの制御

スレーブユニット上の汎用I/O PORTを制御するために説明をします。

(1) 汎用I/O PORTの説明

I/O 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
CB-34/10	あり（一部の信号でラッチ機能、ラッチエッジの選択が可能）

(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。

6-2. 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

● 関数一覧

関数名	機能
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の指定信号の状態を読み出し
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 AND書き込み関数	汎用I/O PORTに指定データとの論理積の書き込み
汎用I/O PORT OR書き込み関数	汎用I/O PORTに指定データとの論理和を書き込み
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
ラッチ機能選択書き込み関数	汎用I/O PORTのラッチ機能を設定
ラッチエッジ選択書き込み関数	汎用I/O PORTのラッチエッジを設定
ラッチ機能選択読み出し関数	汎用I/O PORTのラッチ機能の設定の読み出し
ラッチエッジ選択読み出し関数	汎用I/O PORTのラッチエッジの設定の読み出し
ラッチクリア書き込み関数	汎用I/O PORTのラッチデータをクリア

RESULT構造体

AL I/Oユニット対応DLL

CB-34/10

説明

I/O 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 ACIO_TAG_S_RESULT {
 WORD *ACIO_Result*[4];
 } ACIO_S_RESULT;

VB Type ACIO_S_RESULT
 ACIO_Result(1 To 4) As Integer
 End Type

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

C#.NET struct ACIO_S_RESULT
 {
 [MarshalAs(UnmanagedType.ByValArray, SizeConst=4)] public ushort[] *ACIO_Result*;
 public ACIO_S_RESULT(ushort dummy)
 {
 ACIO_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*ACIO_Result*[0]~*ACIO_Result*[3]は、Visual Basicでは*ACIO_Result*(1)~*ACIO_Result*(4)、Visual Basic.NETでは*ACIO_Result*(0)~*ACIO_Result*(3)、Visual C#.NETでは*ACIO_Result*[0]~*ACIO_Result*[3]に対応します。

ACIO_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
60	汎用I/O PORTオープン関数	130	ラッチ機能選択書き込み関数
61	汎用I/O PORTクローズ関数	131	ラッチエッジ選択書き込み関数
62	汎用I/O PORT一括読み出し関数	132	ラッチ機能選択読み出し関数
63	汎用I/O PORT指定信号読み出し関数	133	ラッチエッジ選択読み出し関数
64	汎用I/O PORT一括書き込み関数	134	ラッチクリア書き込み関数
65	汎用I/O PORT AND書き込み関数		
66	汎用I/O PORT OR書き込み関数		
67	汎用I/O PORT指定信号LOW書き込み関数		
68	汎用I/O PORT指定信号HIGH書き込み関数		

値は10進表記です。

ACIO_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
11	接続されているスレーブでは、関数の実行ができません
12	指定された汎用I/O PORTは、オープンされていません
13	指定された汎用I/O PORTは、すでにオープンされています
18	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。

ACIO_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。

ACIO_Result[3] …… 将来の拡張用です。

汎用I/O PORTオープン関数

AL I/Oユニット対応DLL

CB-34/10

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 ACIO_BPortOpen(WORD IfNo, WORD SlaveAddr, WORD SlaveType, WORD Sel, DWORD *phPort, ACIO_S_RESULT *psResult);`

VB `Function ACIO_BPortOpen(ByVal IfNo As Integer, ByVal SlaveAddr As Integer, ByVal SlaveType As Integer, ByVal Sel As Integer, phPort As Long, psResult As ACIO_S_RESULT) As Boolean`

VB.NET `Function ACIO_BPortOpen(ByVal IfNo As Short, ByVal SlaveAddr As Short, ByVal SlaveType As Short, ByVal Sel As Short, ByRef phPort As Integer, ByRef psResult As ACIO_S_RESULT) As Boolean`

C#.NET `bool ACIO.BPortOpen(ushort IfNo, ushort SlaveAddr, ushort SlaveType, ushort Sel, ref uint phPort, ref ACIO_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 <code>IfNo</code> の値
CB-09	ボード番号 (0~9)
CB-23/USB	ACIO_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。この引数は、次の値のいずれかになります。

引数 <code>SlaveType</code> の値	スレーブタイプ
ACIO_SLAVE_CB34	CB-34/10

`Sel` … オープンするPORTを指定します。CB-34/10のみで有効です。

引数 <code>Select</code> の値	オープンするPORT
ACIO_CB34_10_20	$\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$
ACIO_CB34_30_40	$\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$

`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 I/Oユニット対応DLL

CB-34/10

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 ACIO_BPortClose(DWORD hPort, ACIO_S_RESULT *psResult);`

VB `Function ACIO_BPortClose(ByVal hPort As Long, psResult As ACIO_S_RESULT) As Boolean`

VB.NET `Function ACIO_BPortClose(ByVal hPort As Integer, ByRef psResult As ACIO_S_RESULT) As Boolean`

C#.NET `bool ACIO.BPortClose(uint hPort, ref ACIO_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 I/Oユニット対応DLL

CB-34/10

機能

指定された汎用I/O PORTの状態を一括読み出しします。

書式

C言語 `BOOL ACIO_BPortIn(DWORD hPort, DWORD *pData, ACIO_S_RESULT *psResult);`VB `Function ACIO_BPortIn(ByVal hPort As Long, pData As Long, psResult As ACIO_S_RESULT) As Boolean`VB.NET `Function ACIO_BPortIn(ByVal hPort As Integer, ByRef pData As Integer, ByRef psResult As ACIO_S_RESULT) As Boolean`C#.NET `bool ACIO.BPortIn(uint hPort, ref uint pData, ref ACIO_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。

- ・ $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンしている場合

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}
$\overline{IN27}$	$\overline{IN26}$	$\overline{IN25}$	$\overline{IN24}$	$\overline{IN23}$	$\overline{IN22}$	$\overline{IN21}$	$\overline{IN20}$	$\overline{IN17}$	$\overline{IN16}$	$\overline{IN15}$	$\overline{IN14}$	$\overline{IN13}$	$\overline{IN12}$	$\overline{IN11}$	$\overline{IN10}$
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
$\overline{OUT27}$	$\overline{OUT26}$	$\overline{OUT25}$	$\overline{OUT24}$	$\overline{OUT23}$	$\overline{OUT22}$	$\overline{OUT21}$	$\overline{OUT20}$	$\overline{OUT17}$	$\overline{OUT16}$	$\overline{OUT15}$	$\overline{OUT14}$	$\overline{OUT13}$	$\overline{OUT12}$	$\overline{OUT11}$	$\overline{OUT10}$

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。

- ・ $\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$ をオープンしている場合

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}
$\overline{IN47}$	$\overline{IN46}$	$\overline{IN45}$	$\overline{IN44}$	$\overline{IN43}$	$\overline{IN42}$	$\overline{IN41}$	$\overline{IN40}$	$\overline{IN37}$	$\overline{IN36}$	$\overline{IN35}$	$\overline{IN34}$	$\overline{IN33}$	$\overline{IN32}$	$\overline{IN31}$	$\overline{IN30}$
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
$\overline{OUT47}$	$\overline{OUT46}$	$\overline{OUT45}$	$\overline{OUT44}$	$\overline{OUT43}$	$\overline{OUT42}$	$\overline{OUT41}$	$\overline{OUT40}$	$\overline{OUT37}$	$\overline{OUT36}$	$\overline{OUT35}$	$\overline{OUT34}$	$\overline{OUT33}$	$\overline{OUT32}$	$\overline{OUT31}$	$\overline{OUT30}$

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 I/Oユニット対応DLL

CB-34/I0

機能

指定された汎用I/O PORTの指定信号の状態を読み出します。

書式

C言語 `BOOL ACIO_BSigIn(DWORD hPort, WORD SigNo, WORD *pStatus, ACIO_S_RESULT *psResult);`VB `Function ACIO_BSigIn(ByVal hPort As Long, ByVal SigNo As Integer, pStatus As Integer, psResult As ACIO_S_RESULT) As Boolean`VB.NET `Function ACIO_BSigIn(ByVal hPort As Integer, ByVal SigNo As Short, ByRef pStatus As Short, ByRef psResult As ACIO_S_RESULT) As Boolean`C#.NET `bool ACIO.BSigIn(uint hPort, ushort SigNo, ref ushort pStatus, ref ACIO_S_RESULT psResult);`

引数

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

- ・ $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンしている場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
ACIO_CB34_IN10	$\overline{IN10}$	ACIO_CB34_IN20	$\overline{IN20}$
ACIO_CB34_IN11	$\overline{IN11}$	ACIO_CB34_IN21	$\overline{IN21}$
ACIO_CB34_IN12	$\overline{IN12}$	ACIO_CB34_IN22	$\overline{IN22}$
ACIO_CB34_IN13	$\overline{IN13}$	ACIO_CB34_IN23	$\overline{IN23}$
ACIO_CB34_IN14	$\overline{IN14}$	ACIO_CB34_IN24	$\overline{IN24}$
ACIO_CB34_IN15	$\overline{IN15}$	ACIO_CB34_IN25	$\overline{IN25}$
ACIO_CB34_IN16	$\overline{IN16}$	ACIO_CB34_IN26	$\overline{IN26}$
ACIO_CB34_IN17	$\overline{IN17}$	ACIO_CB34_IN27	$\overline{IN27}$

- ・ $\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$ をオープンしている場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
ACIO_CB34_IN30	$\overline{IN30}$	ACIO_CB34_IN40	$\overline{IN40}$
ACIO_CB34_IN31	$\overline{IN31}$	ACIO_CB34_IN41	$\overline{IN41}$
ACIO_CB34_IN32	$\overline{IN32}$	ACIO_CB34_IN42	$\overline{IN42}$
ACIO_CB34_IN33	$\overline{IN33}$	ACIO_CB34_IN43	$\overline{IN43}$
ACIO_CB34_IN34	$\overline{IN34}$	ACIO_CB34_IN44	$\overline{IN44}$
ACIO_CB34_IN35	$\overline{IN35}$	ACIO_CB34_IN45	$\overline{IN45}$
ACIO_CB34_IN36	$\overline{IN36}$	ACIO_CB34_IN46	$\overline{IN46}$
ACIO_CB34_IN37	$\overline{IN37}$	ACIO_CB34_IN47	$\overline{IN47}$

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	指定信号のレベル
ACIO_LOW	LOWレベル
ACIO_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 I/Oユニット対応DLL

CB-34/10

機 能

指定された汎用I/O PORTにデータを一括書き込みします。

書 式

C言語 `BOOL ACIO_BPortOut(DWORD hPort, DWORD *pData, ACIO_S_RESULT *psResult);`VB `Function ACIO_BPortOut(ByVal hPort As Long, pData As Long, psResult As ACIO_S_RESULT) As Boolean`VB.NET `Function ACIO_BPortOut(ByVal hPort As Integer, ByRef pData As Integer, ByRef psResult As ACIO_S_RESULT) As Boolean`C#.NET `bool ACIO.BPortOut(uint hPort, ref uint pData, ref ACIO_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。

- ・ $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンしている場合

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
$\overline{OUT27}$	$\overline{OUT26}$	$\overline{OUT25}$	$\overline{OUT24}$	$\overline{OUT23}$	$\overline{OUT22}$	$\overline{OUT21}$	$\overline{OUT20}$	$\overline{OUT17}$	$\overline{OUT16}$	$\overline{OUT15}$	$\overline{OUT14}$	$\overline{OUT13}$	$\overline{OUT12}$	$\overline{OUT11}$	$\overline{OUT10}$

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。

- ・ $\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$ をオープンしている場合

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
$\overline{OUT47}$	$\overline{OUT46}$	$\overline{OUT45}$	$\overline{OUT44}$	$\overline{OUT43}$	$\overline{OUT42}$	$\overline{OUT41}$	$\overline{OUT40}$	$\overline{OUT37}$	$\overline{OUT36}$	$\overline{OUT35}$	$\overline{OUT34}$	$\overline{OUT33}$	$\overline{OUT32}$	$\overline{OUT31}$	$\overline{OUT30}$

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 AND書き込み関数

AL I/Oユニット対応DLL

CB-34/10

機能

指定された汎用I/O PORTの内容と、指定された変数の内容の論理積を、指定された汎用I/O PORTに書き込みます。

書式

C言語 BOOL ACIO_BPortAndOut(DWORD *hPort*, DWORD **pData*, ACIO_S_RESULT **psResult*);

VB Function ACIO_BPortAndOut(ByVal *hPort* As Long, *pData* As Long, *psResult* As ACIO_S_RESULT) As Boolean

VB.NET Function ACIO_BPortAndOut(ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As ACIO_S_RESULT) As Boolean

C#.NET bool ACIO.BPortAndOut(uint *hPort*, ref uint *pData*, ref ACIO_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。

- ・ $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンしている場合

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
$\overline{OUT27}$	$\overline{OUT26}$	$\overline{OUT25}$	$\overline{OUT24}$	$\overline{OUT23}$	$\overline{OUT22}$	$\overline{OUT21}$	$\overline{OUT20}$	$\overline{OUT17}$	$\overline{OUT16}$	$\overline{OUT15}$	$\overline{OUT14}$	$\overline{OUT13}$	$\overline{OUT12}$	$\overline{OUT11}$	$\overline{OUT10}$

HIGHレベルにするときに0、変更しないときに1を書き込みます。

- ・ $\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$ をオープンしている場合

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
$\overline{OUT47}$	$\overline{OUT46}$	$\overline{OUT45}$	$\overline{OUT44}$	$\overline{OUT43}$	$\overline{OUT42}$	$\overline{OUT41}$	$\overline{OUT40}$	$\overline{OUT37}$	$\overline{OUT36}$	$\overline{OUT35}$	$\overline{OUT34}$	$\overline{OUT33}$	$\overline{OUT32}$	$\overline{OUT31}$	$\overline{OUT30}$

HIGHレベルにするときに0、変更しないときに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 OR書き込み関数

AL I/Oユニット対応DLL

CB-34/10

機能

指定された汎用I/O PORTの内容と、指定された変数の内容の論理和を、指定された汎用I/O PORTに書き込みます。

書式

C言語 BOOL ACIO_BPortOrOut(DWORD *hPort*, DWORD **pData*, ACIO_S_RESULT **psResult*);

VB Function ACIO_BPortOrOut(ByVal *hPort* As Long, *pData* As Long, *psResult* As ACIO_S_RESULT) As Boolean

VB.NET Function ACIO_BPortOrOut(ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As ACIO_S_RESULT) As Boolean

C#.NET bool ACIO.BPortOrOut(uint *hPort*, ref uint *pData*, ref ACIO_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。

- ・ $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンしている場合

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
$\overline{OUT27}$	$\overline{OUT26}$	$\overline{OUT25}$	$\overline{OUT24}$	$\overline{OUT23}$	$\overline{OUT22}$	$\overline{OUT21}$	$\overline{OUT20}$	$\overline{OUT17}$	$\overline{OUT16}$	$\overline{OUT15}$	$\overline{OUT14}$	$\overline{OUT13}$	$\overline{OUT12}$	$\overline{OUT11}$	$\overline{OUT10}$

変更しないときに0、LOWレベルにするときに1を書き込みます。

- ・ $\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$ をオープンしている場合

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
$\overline{OUT47}$	$\overline{OUT46}$	$\overline{OUT45}$	$\overline{OUT44}$	$\overline{OUT43}$	$\overline{OUT42}$	$\overline{OUT41}$	$\overline{OUT40}$	$\overline{OUT37}$	$\overline{OUT36}$	$\overline{OUT35}$	$\overline{OUT34}$	$\overline{OUT33}$	$\overline{OUT32}$	$\overline{OUT31}$	$\overline{OUT30}$

変更しないときに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 I/Oユニット対応DLL

CB-34/I0

機能

指定された汎用I/O PORTの指定信号をLOWレベルにします。

書式

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

引数

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

- ・ $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンしている場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
ACIO_CB34_OUT10	$\overline{OUT10}$	ACIO_CB34_OUT20	$\overline{OUT20}$
ACIO_CB34_OUT11	$\overline{OUT11}$	ACIO_CB34_OUT21	$\overline{OUT21}$
ACIO_CB34_OUT12	$\overline{OUT12}$	ACIO_CB34_OUT22	$\overline{OUT22}$
ACIO_CB34_OUT13	$\overline{OUT13}$	ACIO_CB34_OUT23	$\overline{OUT23}$
ACIO_CB34_OUT14	$\overline{OUT14}$	ACIO_CB34_OUT24	$\overline{OUT24}$
ACIO_CB34_OUT15	$\overline{OUT15}$	ACIO_CB34_OUT25	$\overline{OUT25}$
ACIO_CB34_OUT16	$\overline{OUT16}$	ACIO_CB34_OUT26	$\overline{OUT26}$
ACIO_CB34_OUT17	$\overline{OUT17}$	ACIO_CB34_OUT27	$\overline{OUT27}$

- ・ $\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$ をオープンしている場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
ACIO_CB34_OUT30	$\overline{OUT30}$	ACIO_CB34_OUT40	$\overline{OUT40}$
ACIO_CB34_OUT31	$\overline{OUT31}$	ACIO_CB34_OUT41	$\overline{OUT41}$
ACIO_CB34_OUT32	$\overline{OUT32}$	ACIO_CB34_OUT42	$\overline{OUT42}$
ACIO_CB34_OUT33	$\overline{OUT33}$	ACIO_CB34_OUT43	$\overline{OUT43}$
ACIO_CB34_OUT34	$\overline{OUT34}$	ACIO_CB34_OUT44	$\overline{OUT44}$
ACIO_CB34_OUT35	$\overline{OUT35}$	ACIO_CB34_OUT45	$\overline{OUT45}$
ACIO_CB34_OUT36	$\overline{OUT36}$	ACIO_CB34_OUT46	$\overline{OUT46}$
ACIO_CB34_OUT37	$\overline{OUT37}$	ACIO_CB34_OUT47	$\overline{OUT47}$

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 I/Oユニット対応DLL

CB-34/10

機 能

指定された汎用I/O PORTの指定信号をHIGHレベルにします。

書 式

C言語 `BOOL ACIO_BSigHighOut(DWORD hPort, WORD SigNo, ACIO_S_RESULT *psResult);`VB `Function ACIO_BSigHighOut(ByVal hPort As Long, ByVal SigNo As Integer, psResult As ACIO_S_RESULT) As Boolean`VB.NET `Function ACIO_BSigHighOut(ByVal hPort As Integer, ByVal SigNo As Short, ByRef psResult As ACIO_S_RESULT) As Boolean`C#.NET `bool ACIO.BSigHighOut(uint hPort, ushort SigNo, ref ACIO_S_RESULT psResult);`

引 数

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

- ・ $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンしている場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
ACIO_CB34_OUT10	$\overline{OUT10}$	ACIO_CB34_OUT20	$\overline{OUT20}$
ACIO_CB34_OUT11	$\overline{OUT11}$	ACIO_CB34_OUT21	$\overline{OUT21}$
ACIO_CB34_OUT12	$\overline{OUT12}$	ACIO_CB34_OUT22	$\overline{OUT22}$
ACIO_CB34_OUT13	$\overline{OUT13}$	ACIO_CB34_OUT23	$\overline{OUT23}$
ACIO_CB34_OUT14	$\overline{OUT14}$	ACIO_CB34_OUT24	$\overline{OUT24}$
ACIO_CB34_OUT15	$\overline{OUT15}$	ACIO_CB34_OUT25	$\overline{OUT25}$
ACIO_CB34_OUT16	$\overline{OUT16}$	ACIO_CB34_OUT26	$\overline{OUT26}$
ACIO_CB34_OUT17	$\overline{OUT17}$	ACIO_CB34_OUT27	$\overline{OUT27}$

- ・ $\overline{IN30} \sim \overline{IN37}$ 、 $\overline{IN40} \sim \overline{IN47}$ 、 $\overline{OUT30} \sim \overline{OUT37}$ 、 $\overline{OUT40} \sim \overline{OUT47}$ をオープンしている場合

引数 <i>SigNo</i> の値	指定信号	引数 <i>SigNo</i> の値	指定信号
ACIO_CB34_OUT30	$\overline{OUT30}$	ACIO_CB34_OUT40	$\overline{OUT40}$
ACIO_CB34_OUT31	$\overline{OUT31}$	ACIO_CB34_OUT41	$\overline{OUT41}$
ACIO_CB34_OUT32	$\overline{OUT32}$	ACIO_CB34_OUT42	$\overline{OUT42}$
ACIO_CB34_OUT33	$\overline{OUT33}$	ACIO_CB34_OUT43	$\overline{OUT43}$
ACIO_CB34_OUT34	$\overline{OUT34}$	ACIO_CB34_OUT44	$\overline{OUT44}$
ACIO_CB34_OUT35	$\overline{OUT35}$	ACIO_CB34_OUT45	$\overline{OUT45}$
ACIO_CB34_OUT36	$\overline{OUT36}$	ACIO_CB34_OUT46	$\overline{OUT46}$
ACIO_CB34_OUT37	$\overline{OUT37}$	ACIO_CB34_OUT47	$\overline{OUT47}$

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。

ラッチ機能選択書き込み関数

AL I/Oユニット対応DLL

CB-34/10

機 能

指定された汎用I/O PORTのラッチ機能を設定します。

書 式

C言語 BOOL ACIO_BWLatchSel (DWORD *hPort*, WORD **pData*, ACIO_S_RESULT **psResult*);

VB Function ACIO_BWLatchSel (ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT) As Boolean

VB.NET Function ACIO_BWLatchSel (ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT) As Boolean

C#.NET bool ACIO.BWLatchSel (uint *hPort*, ref ushort *pData*, ref ACIO_S_RESULT *psResult*);

引 数

hPort ... $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンした汎用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。

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	0	0	0	0	IN13L	IN12L	IN11L	IN10L

IN10L～IN13Lで、 $\overline{IN10} \sim \overline{IN13}$ のラッチ機能を選択します。

ラッチ機能を無効にするときに0、有効にするときに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。

ラッチエッジ選択書き込み関数

AL I/Oユニット対応DLL

CB-34/10

機能

指定された汎用I/O PORTのラッチのエッジを設定します。

ラッチ機能選択書き込み関数で、ラッチ機能が有効にされているビットのみで有効です。

書式

C言語 BOOL ACIO_BWLatchEdge(DWORD *hPort*, WORD **pData*, ACIO_S_RESULT **psResult*);

VB Function ACIO_BWLatchEdge(ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT) As Boolean

VB.NET Function ACIO_BWLatchEdge(ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT) As Boolean

C#.NET bool ACIO.BWLatchEdge(uint *hPort*, ref ushort *pData*, ref ACIO_S_RESULT *psResult*);

引数

hPort ... $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンした汎用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。

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	0	0	0	0	IN13E	IN12E	IN11E	IN10E

IN10E～IN13Eで、 $\overline{IN10} \sim \overline{IN13}$ のラッチのエッジを選択します。

立ち下がりエッジにするとときに0、立ち上がりエッジにするとときに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。

ラッチ機能選択読み出し関数

AL I/Oユニット対応DLL

CB-34/10

機 能

指定された汎用I/O PORTのラッチ機能の設定を読み出します。

書 式

C言語 BOOL ACIO_BRLatchSel (DWORD *hPort*, WORD **pData*, ACIO_S_RESULT **psResult*);

VB Function ACIO_BRLatchSel (ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT) As Boolean

VB.NET Function ACIO_BRLatchSel (ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT) As Boolean

C#.NET bool ACIO.BRLatchSel (uint *hPort*, ref ushort *pData*, ref ACIO_S_RESULT *psResult*);

引 数

hPort ... $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンした汎用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。

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	0	0	0	0	IN13L	IN12L	IN11L	IN10L

IN10L～IN13Lに、 $\overline{IN10} \sim \overline{IN13}$ のラッチ機能が読み出されます。
ラッチ機能が無効なときに0、有効なときに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。

ラッチエッジ選択読み出し関数

AL I/Oユニット対応DLL

CB-34/10

機 能

指定された汎用I/O PORTのラッチのエッジの設定状態を読み出します。

書 式

C言語 BOOL ACIO_BRLatchEdge(DWORD *hPort*, WORD **pData*, ACIO_S_RESULT **psResult*);

VB Function ACIO_BRLatchEdge(ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT) As Boolean

VB.NET Function ACIO_BRLatchEdge(ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT) As Boolean

C#.NET bool ACIO.BRLatchEdge(uint *hPort*, ref ushort *pData*, ref ACIO_S_RESULT *psResult*);

引 数

hPort ... $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンした汎用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。

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	0	0	0	0	IN13E	IN12E	IN11E	IN10E

IN10E～IN13Eに、 $\overline{IN10} \sim \overline{IN13}$ のラッチのエッジが読み出されます。
立ち下がりエッジのときに0、立ち上がりエッジのときに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。

ラッチクリア書き込み関数

AL I/Oユニット対応DLL

CB-34/10

機 能

指定された汎用I/O PORTのラッチデータをクリアします。

書 式

C言語 BOOL ACIO_BWLatchClr(DWORD *hPort*, WORD **pData*, ACIO_S_RESULT **psResult*);

VB Function ACIO_BWLatchClr(ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT) As Boolean

VB.NET Function ACIO_BWLatchClr(ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT) As Boolean

C#.NET bool ACIO.BWLatchClr(uint *hPort*, ref ushort *pData*, ref ACIO_S_RESULT *psResult*);

引 数

hPort ... $\overline{IN10} \sim \overline{IN17}$ 、 $\overline{IN20} \sim \overline{IN27}$ 、 $\overline{OUT10} \sim \overline{OUT17}$ 、 $\overline{OUT20} \sim \overline{OUT27}$ をオープンした汎用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。

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	0	0	0	0	IN13C	IN12C	IN11C	IN10C

IN10C～IN13Cが、 $\overline{IN10} \sim \overline{IN13}$ のラッチデータのクリアビットになります。
クリアしないときに0、クリアするときに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。

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¥AIUsbA.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vc¥VcAIUsbA.lib
C++ Builderヘッダファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥AIUsbA.h
C++ Builderライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥BcAIUsbA.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb.NET¥AIUsbA.vb
Visual Basic関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb¥AIUsbA.bas
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥C#.Net¥AIUsbA.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¥AIUsbA.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vc¥VcAIUsbA.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vb.NET¥AIUsbA.vb
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥C#.Net¥AIUsbA.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

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¥AcloA.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vc¥VcAcloA.lib
C++ Builderヘッダファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥AcloA.h
C++ Builderライブラリファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Builder¥BcAcloA.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb.NET¥AcloA.vb
Visual Basic関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥Vb¥AcloA.bas
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_02v1.00¥Bin¥x86¥C#.Net¥AcloA.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¥AcloA.h
Visual C++ライブラリファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vc¥VcAcloA.lib
Visual Basic .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥Vb.NET¥AcloA.vb
Visual C# .NET関数定義ファイル	¥Program Files¥Mpl28_03v1.00¥Bin¥x64¥C#.Net¥AcloA.cs

● 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

AL MCC06ユニット取扱説明書をご覧ください

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 ¥Program Files¥Mpl28_02v1.00¥Sample¥x86¥Builder
Visual Basic .NET Visual Basic	¥Program Files¥Mpl28_02v1.00¥Sample¥x86¥Vb.NET ¥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

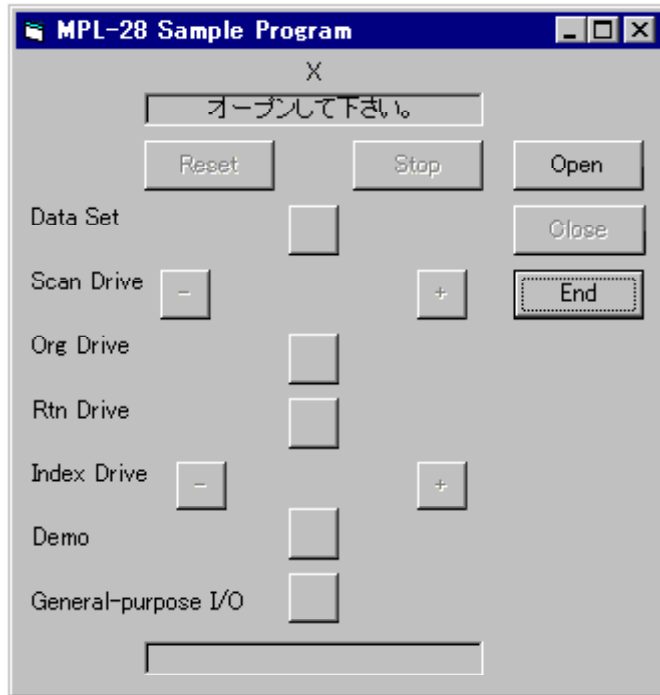
8-2. ご使用になる前に

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。

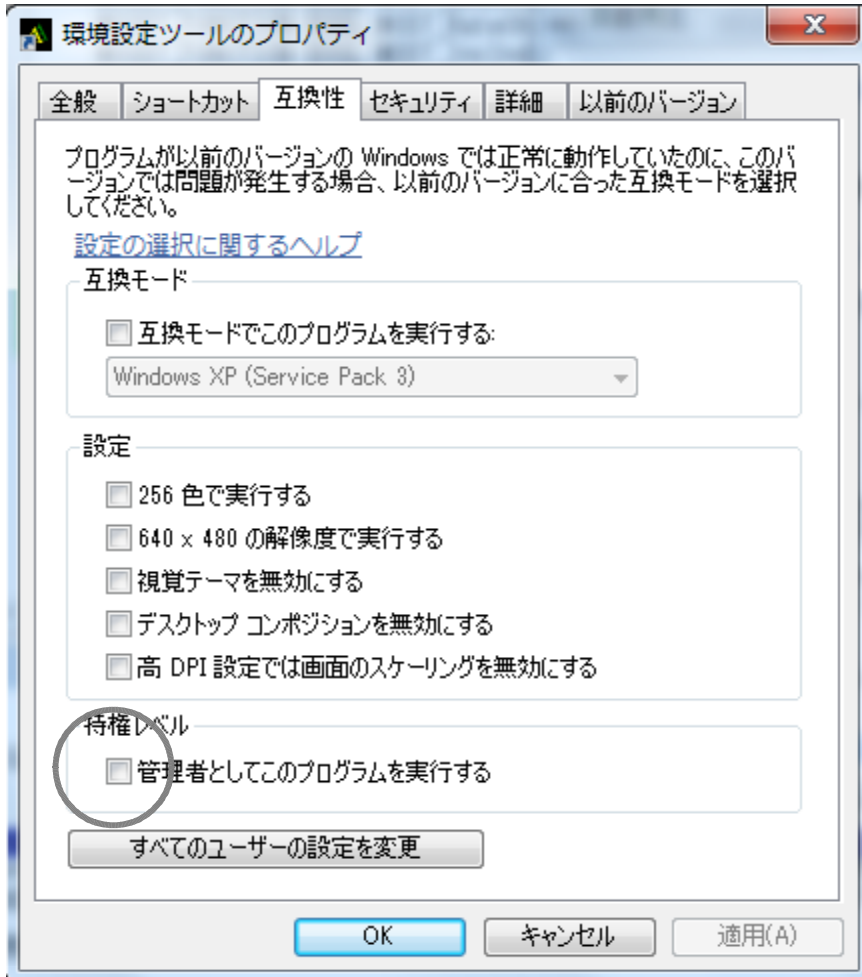
10. トラブルシューティング

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。

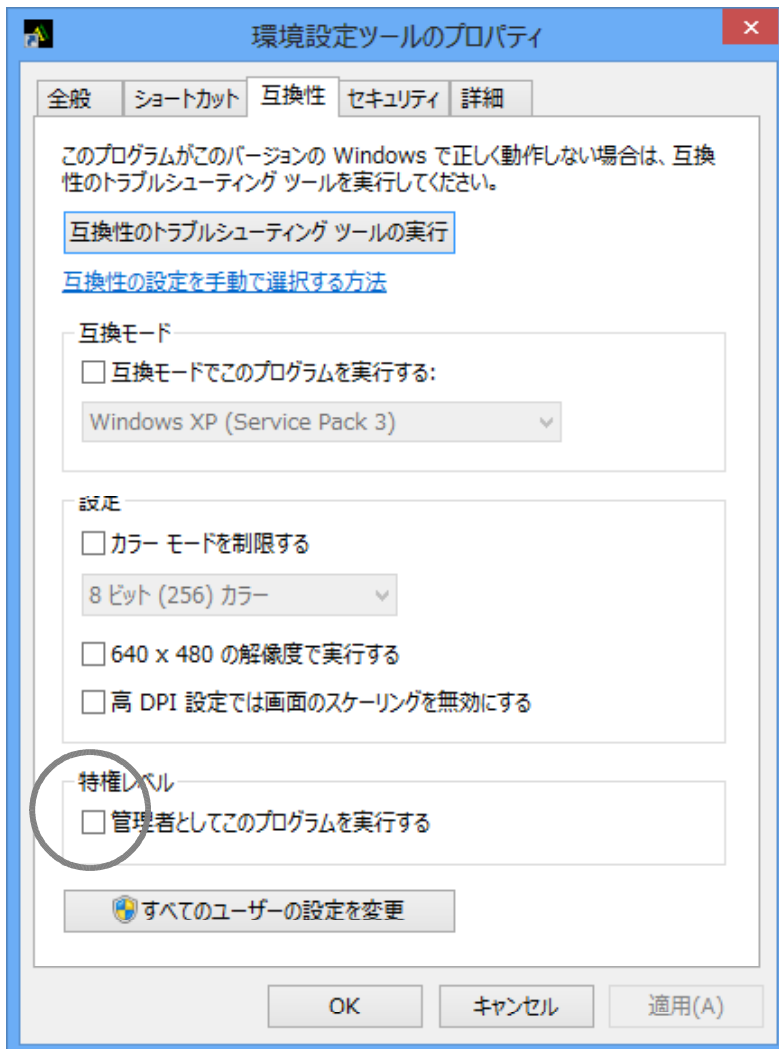
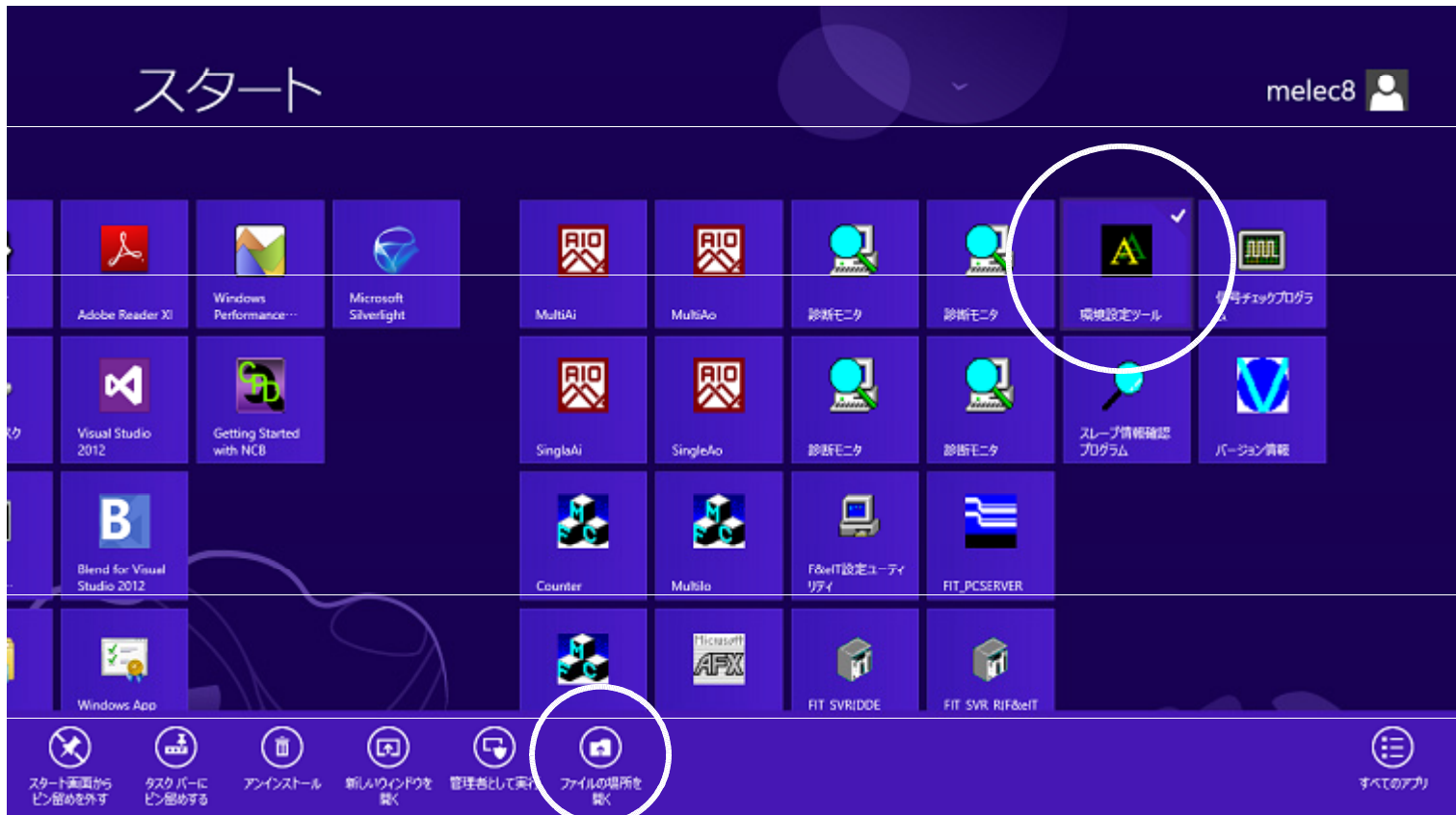
(1) Windows Vista / 7

- 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

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>