

**Anywire DB A40/20 シリーズ PCI インターフェース用
Windows7, 8.1, 10 (64bit) ドライバライブラリ
使用説明書**

2016/11/21

株式会社エニワイヤ

目次

1. はじめに	1
2. ドライバのインストール	2
3. ドライバライブラリ	8
3-1. インターフェース関数	9
3-1-1. ボードの情報を取得する	10
3-1-2. ボードの枚数を取得する	11
3-1-3. 指定ボード1枚分のボードの情報を取得する	12
3-1-4. バイトデータを入力する	13
3-1-5. バイトデータを出力する	14
3-1-6. ワードデータを入力する	15
3-1-7. ワードデータを出力する	16

1. はじめに

本ソフトウェアは、AnyWire DB A40/A20 シリーズの PCI インターフェースボードを Windows7, 8.1, 10 の 64bit OS で使用するためのデバイスドライバです。

適応 OS	Windows7(64bit), Windows8.1(64bit), Windows10(64bit)
最大ボード数	16 枚
対象ボード	AP28-01A AP48-01 AP48-02

本ソフトウェアには以下のファイルが納められています。

PCI_1port2port_Win7-10_64_DRV_20161121			
	—dllLib	AW_Function.h	ドライバライブラリ ヘッダファイル
	—x86	AW_DB_A40_PCI.dll	ドライバライブラリ DLL(32bit ソフト用)
		AW_DB_A40_PCI.lib	ドライバライブラリ・リンク用ライブラリ
	—x64	AW_DB_A40_PCI.dll	ドライバライブラリ DLL(64bit ソフト用)
		AW_DB_A40_PCI.lib	ドライバライブラリ・リンク用ライブラリ
	—driver		
	—AW_DB_A40_PCI_Driver64		
	—Win7_81	AW_DB_A40_PCI.inf	セットアップ情報ファイル(Win7/8.1 用)
		AW_DB_A40_PCI.sys	デバイスドライバ (Win7/8.1 用)
	—Win10	AW_DB_A40_PCI.inf	セットアップ情報ファイル(Win10 用)
		AW_DB_A40_PCI.sys	デバイスドライバ (Win10 用)
	—smpApp		
	—smp1Cs	C#	サンプルソフト
	—smp1Vb	VB.net	サンプルソフト
	—smp1Vc	VC(MFC)	サンプルソフト

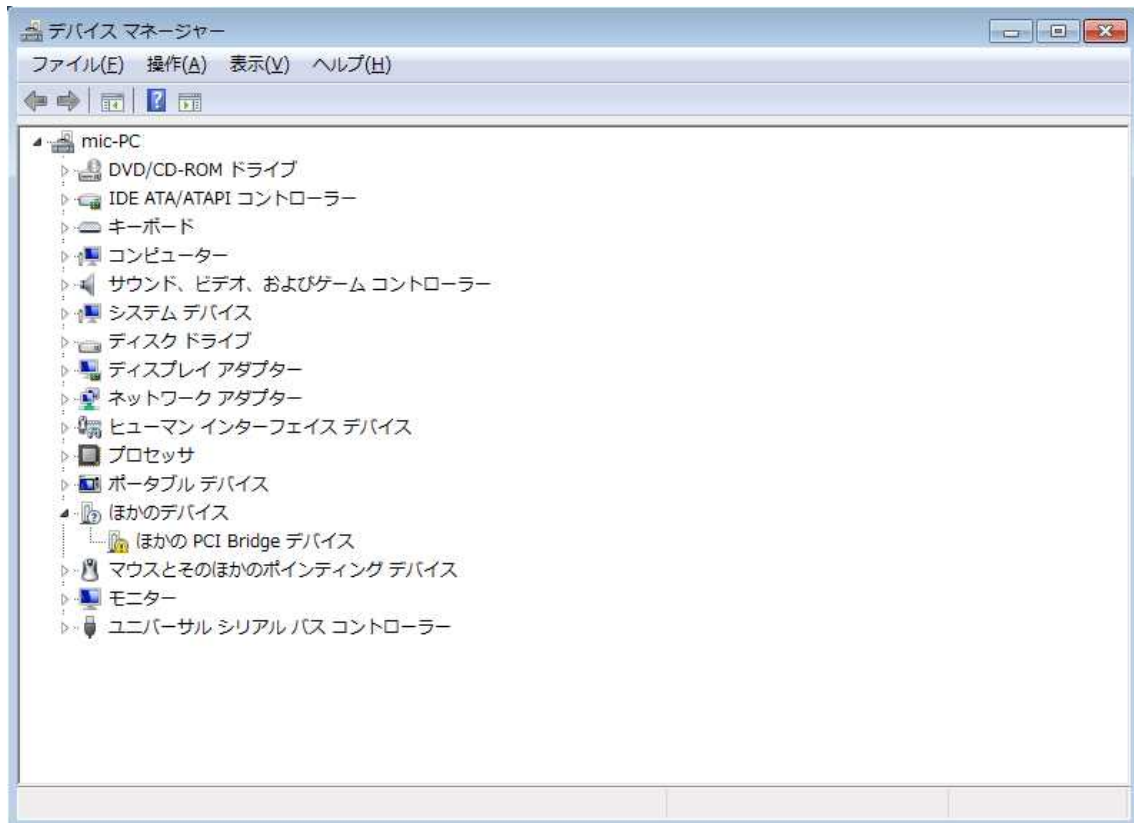
(サンプルソフトは参考用に同梱しており、動作を保証するものではありません。

お問い合わせ等のご遠慮ください。)

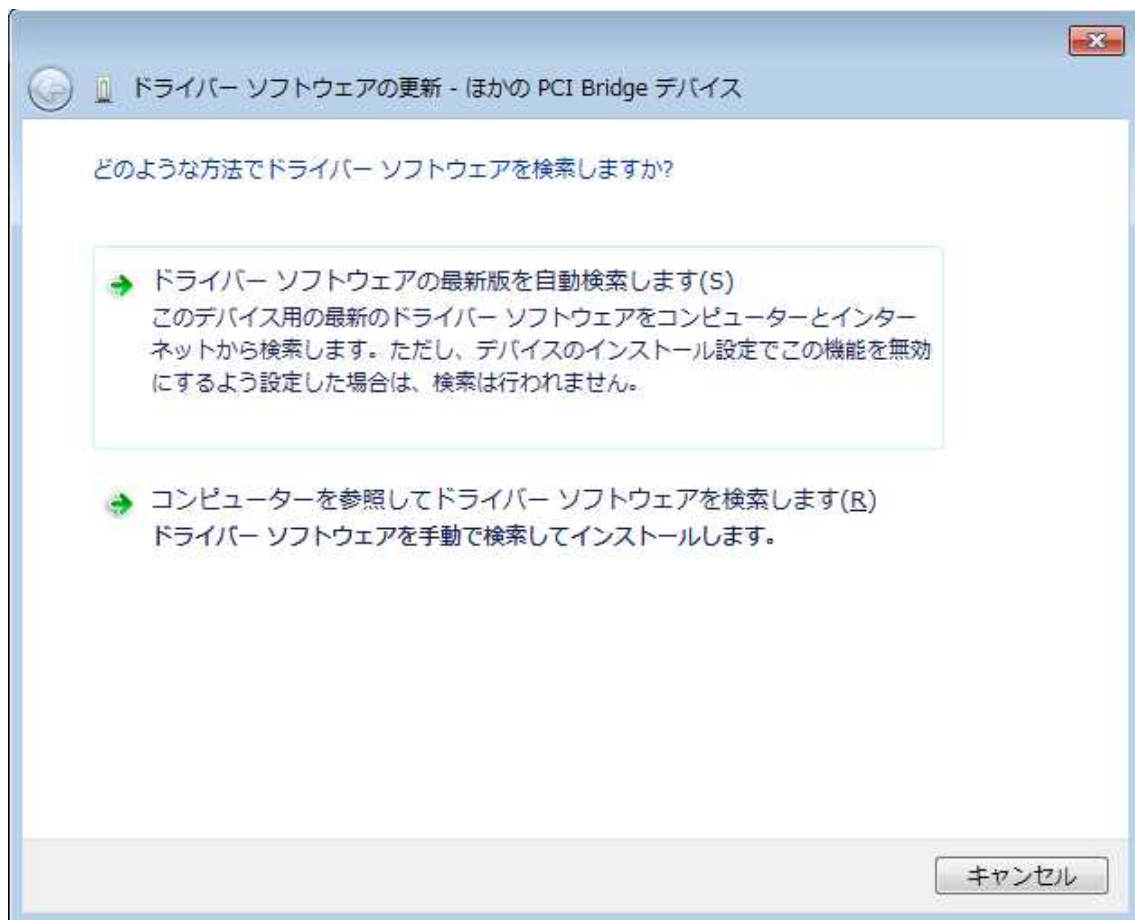
2. ドライバのインストール

本ボードをパソコンの P C I バスに挿入して、電源を入れると plug & play 機能が働き、ハードウェア追加ウィザードが起動されます。

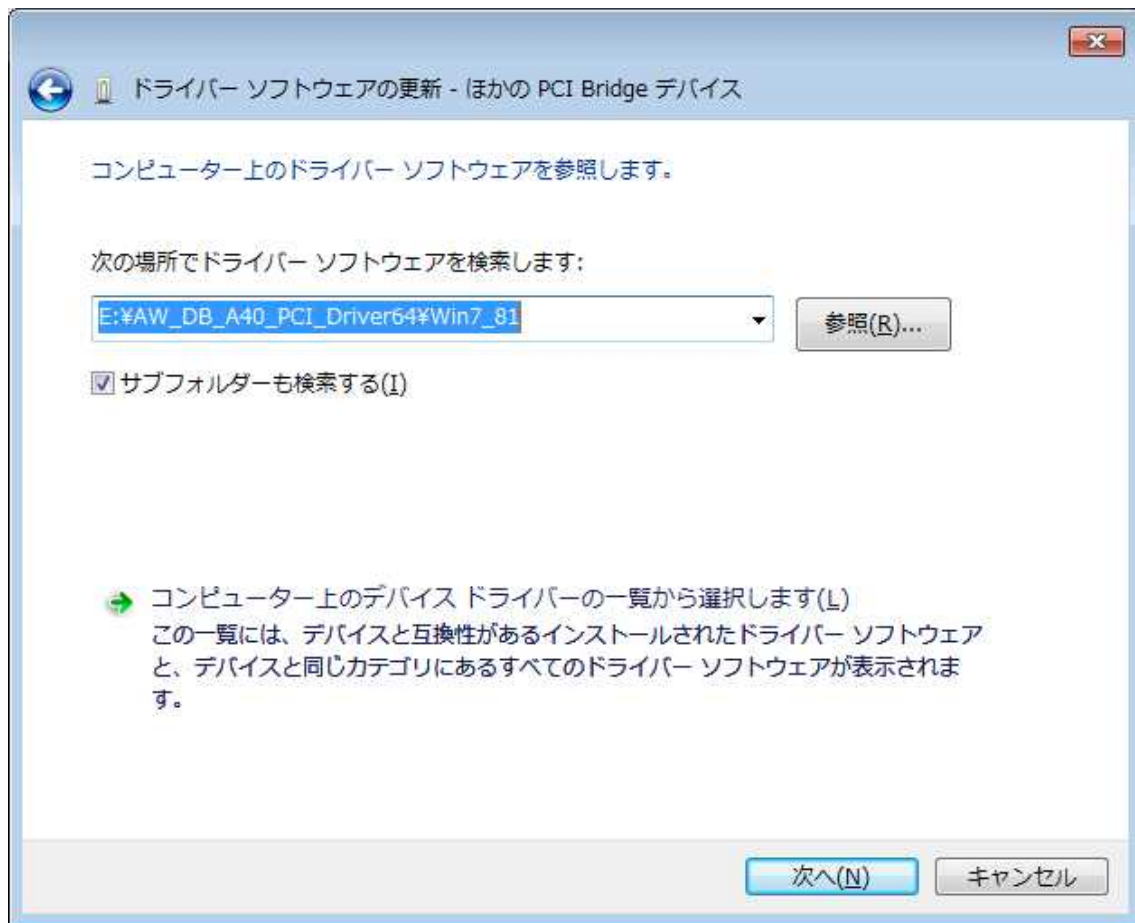
起動されない場合は、[設定] [コントロールパネル] [デバイスとプリンター] [デバイスマネージャ]をダブルクリックして起動してください。



ほかの PCI Bridge デバイスを右クリックし、ドライバの更新を選択してください

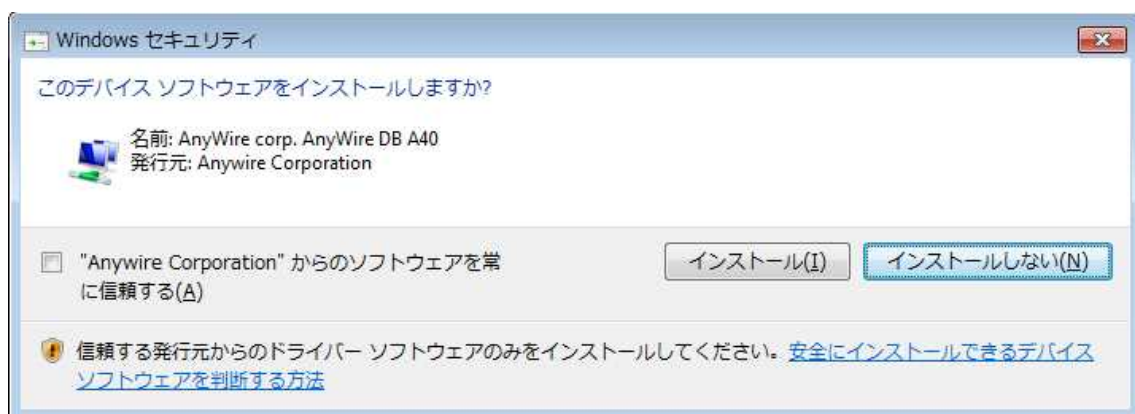


「ドライバーソフトウェアを手動で検索してインストールします」を選択してください。



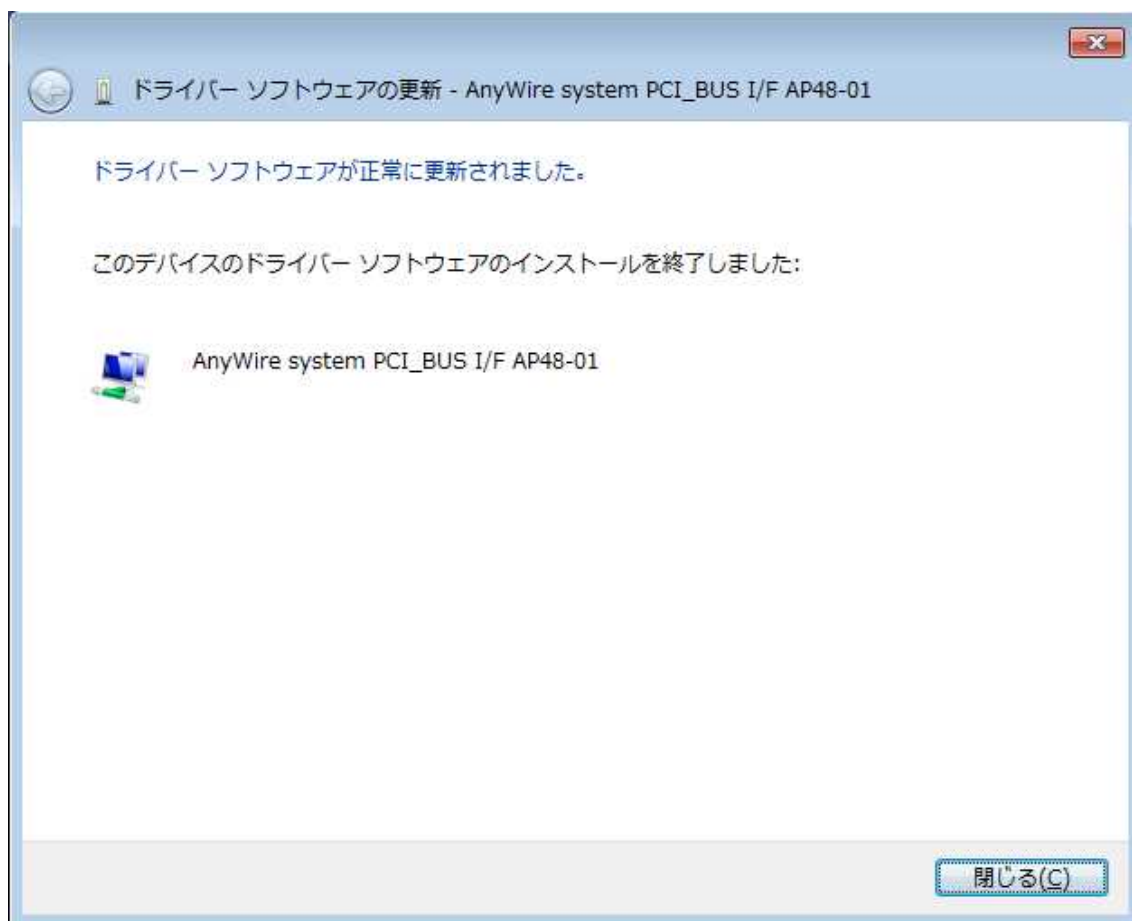
参照ボタンを押して、デバイスドライバが納められているフォルダを選択し、次へを押してください。

Windows7(64bit), Windows8.1(64bit)用ドライバは、
AW_DB_A40_PCI_Driver64¥Win7_81 フォルダに格納されています。
Windows10(64bit)用ドライバは、AW_DB_A40_PCI_Driver64¥Win10 フォルダに格納されています。

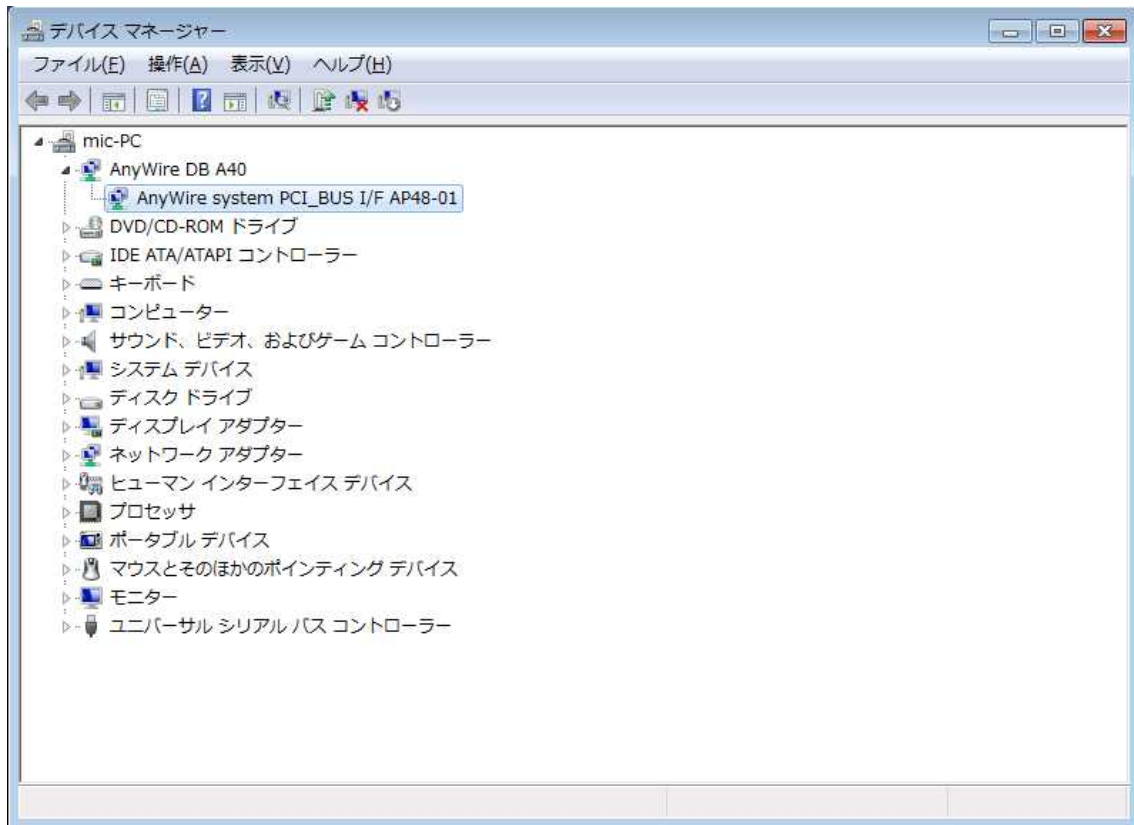


インストールを押してください。

以下の表示になれば、インストールの完了です。



デバイスマネージャで、デバイスが認識されていることを確認してください。



※ボードが『AP28-01A』または『AP48-01』の時、上記のように『AP48-01』と表示されます。ボードが『AP48-02』の時、『AP48-02』と表示されます。

3. ドライバライブラリ

ドライバライブラリは、以下の3つのファイルで構成されます

ファイル名	内容
AW_DB_A40_PCI.dll	ドライバライブラリ DLL
AW_DB_A40_PCI.lib	ドライバライブラリ・リンク用ライブラリ
AW_Function.h	ドライバライブラリ ヘッダファイル

dll, lib ファイルは、作成されるアプリケーションに合わせて 32bit 用(X86 フォルダ), 64bit 用(X64 フォルダ) から必要なものをご使用下さい。

3－1．インターフェース関数

関数名	内容
AW_GetDeviceInfo	ボードの情報を取得する
AW_GetBoardCount	ボードの枚数を取得する（.net 用）
AW_Get1DeviceInfo	指定ボード 1 枚分のボード情報を取得する（.net 用）
AW_ReadByte	バイトデータを入力する
AW_WriteByte	バイトデータを出力する
AW_ReadWord	ワードデータを入力する
AW_WriteWord	ワードデータを出力する

3-1-1. ボードの情報を取得する

int AW_GetDeviceInfo(AW_DEVICE_LIST *AwDeviceList)	
引数	内容
AW_DEVICE_LIST *AwDeviceList	ボード情報を格納する構造体へのポインタ
戻り値	
int	認識されているボード数

使用方法：

```
int ReturnValue;
AW_DEVICE_LIST AwDeviceList;

ReturnValue = AW_GetDeviceInfo(&AwDeviceList);
```

AW_DEVICE_LIST 構造体	デバイス情報リスト構造体
unsigned int BoardCount;	認識ボード数
AW_DEVICE_INFO DeviceInfo[16];	デバイス情報

AW_DEVICE_INFO 構造体	デバイス情報構造体
unsigned short BoardNo;	ボード番号
unsigned short BoardType;	ボードタイプ 0x0000: デバイス無し 0x0020: AP28-01A／AP48-01 0x0021: AP48-02
unsigned long MemSize;	メモリサイズ(アドレス範囲)

※ドライバで『AP28-01A』と『AP48-01』とを区別することは出来ません。

同一 PC に混載される場合は、ボード上のロータリ SW で指定する『ボード番号』で区別して下さい。

3 - 1 - 2 . ボードの枚数を取得する

int AW_GetBoardCount(unsigned int *BoardCount)	
引数	内容
unsigned int *BoardCount	ボード枚数
戻り値	
int	0

使用方法 :

```
int ReturnValue;
```

```
unsigned int BoardCount;
```

```
ReturnValue = AW_GetBoardCount(&BoardCount);
```

※ 本関数は VB.net, C# 用の関数です。

使用方法はサンプルソフトもご参照下さい。

3-1-3. 指定ボード 1 枚分のボードの情報を取得する

int AW_Get1DeviceInfo(unsigned short BoardNo, unsigned short *BoardType, unsigned long *MemSize)	
引数	内容
unsigned short BoardNo	指定ボード番号(0-15)
unsigned short *BoardType	ボードタイプ 0x0000：デバイス無し 0x0020：AP28-01A／AP48-01 0x0021：AP48-02
unsigned long *MemSize	メモリサイズ（アドレス範囲）
戻り値	
int	0：正常終了 -1：ボード番号指定範囲外

※ドライバで『AP28-01A』と『AP48-01』とを区別することは出来ません。

同一 PC に混載される場合は、ボード上のロータリ SW で指定する『ボード番号』で区別して下さい。

使用方法：

```
int ReturnValue;  
unsigned short BoardNo;  
unsigned short BoardType;  
unsigned long MemSize;  
  
BoardNo = 0;  
ReturnValue = AW_Get1DeviceInfo(BoardNo, &BoardType, &MemSize );
```

※ 本関数は VB.net, C# 用の関数です。

使用方法はサンプルソフトもご参照下さい。

3 - 1 - 4. バイトデータを入力する

int AW_ReadByte(unsigned int BoardNo,ULONG Address,UCHAR *ByteData)	
引数	内容
unsigned int BoardNo	対象ボード番号
ULONG Address	入力アドレス
UCHAR *ByteData	読み出したデータを格納する変数へのポインタ
戻り値	
int	0 : 正常終了 -1 : ボード番号が範囲外 -2 : 存在しないボード番号が指定された -3 : パラメータエラー(アドレス範囲外等)

使用方法 :

```
int ReturnValue;
```

```
unsigned int BoardNo;
```

```
ULONG Address;
```

```
UCHAR ByteData;
```

```
BoardNo = 1;
```

```
Address = 0x0000;
```

```
ReturnValue = AW_ReadByte(BoardNo,Address,&ByteData);
```

3 - 1 - 5 . バイトデータを出力する

int AW_WriteByte(unsigned int BoardNo,ULONG Address,UCHAR ByteData)	
引数	内容
unsigned int BoardNo	対象ボード番号
ULONG Address	出力アドレス
UCHAR ByteData	出力データ
戻り値	
int	0 : 正常終了 -1 : ボード番号が範囲外 -2 : 存在しないボード番号が指定された -3 : パラメータエラー(アドレス範囲外等)

使用方法 :

```
int ReturnValue;  
unsigned int BoardNo;  
ULONG Address;  
UCHAR ByteData;
```

```
BoardNo = 1;  
Address = 0x0000;  
ByteData = 0xFF;
```

```
ReturnValue = AW_WriteByte(BoardNo,Address,ByteData);
```


3-1-6. ワードデータを入力する

int AW_ReadWord(unsigned int BoardNo,ULONG Address,USHORT *WordData)	
引数	内容
unsigned int BoardNo	対象ボード番号
ULONG Address	入力アドレス
USHORT *WordData	読み出したデータを格納する変数へのポインタ
戻り値	
int	0 : 正常終了 -1 : ボード番号が範囲外 -2 : 存在しないボード番号が指定された -3 : パラメータエラー(アドレス範囲外等)

使用方法：

```
int ReturnValue;  
unsigned int BoardNo;  
ULONG Address;  
USHORT WordData;
```

```
BoardNo = 1;  
Address = 0x0000;
```

```
ReturnValue = AW_ReadWord(BoardNo,Address,&WordData);
```

3-1-7. ワードデータを出力する

int AW_WriteWord(unsigned int BoardNo,ULONG Address,USHORT WordData)	
引数	内容
unsigned int BoardNo	対象ボード番号
ULONG Address	出力アドレス
USHORT WordData	出力データ
戻り値	
int	0 : 正常終了 -1 : ボード番号が範囲外 -2 : 存在しないボード番号が指定された -3 : パラメータエラー(アドレス範囲外等)

使用方法：

```
int ReturnValue;
```

```
unsigned int BoardNo;
```

```
ULONG Address;
```

```
USHORT WordData;
```

```
BoardNo = 1;
```

```
Address = 0x0000;
```

```
WordData = 0xFFFF;
```

```
ReturnValue = AW_WriteWord(BoardNo,Address,WordData);
```