



株式会社 エニイワイヤ

AnyWireBus - DBシリーズ

ISAバスインターフェースボード用

PCIバスインターフェースボード用

WindowsNT4.0 / Windows2000

ドライバソフトウェアユーザズマニュアル

Version2.0

ビット制御と情報伝送の統合

省配線システム

全 **4** 重 AnyWire DB シリーズ

- 目次 -

1 .	はじめに.....	3
2 .	I S Aバスインターフェースのインストール.....	4
2 - 1	windowsNT4.0 へのインストール.....	4
2-1-1	ドライバのインストール.....	4
2-1-2	I S Aボードの設定	5
2 - 2	windows2000 へのインストール.....	7
1)	ドライバのインストール	7
2)	I S Aボードの設定	9
3 .	P C Iバスインターフェースのインストール.....	10
4 .	インターフェース関数	11
4 - 1	個別ボードの詳細情報を見る場合は？	12
4 - 2	ボードに対してデータ読み込みを行う場合は？（バイト単位）.....	13
4 - 3	ボードに対してデータ書き込みを行う場合は？（バイト単位）.....	14
4 - 4	ボードに対してデータ読み込みを行う場合は？（ワード単位）.....	15
4 - 5	ボードに対してデータ書き込みを行う場合は？（ワード単位）.....	16
4 - 6	ボード情報の初期値を読み込む場合は？	17
4 - 7	ボードの初期値を書き込む場合は？	18
4 - 8	ボードの初期情報を消すには？	19
5 .	ボード情報構造体.....	20
6 .	注意事項.....	21
7 .	更新履歴.....	21

1. はじめに

本ソフトウェアは、AnyWireBus-DB シリーズのパソコンインターフェースボードを Windows NT4.0・Windows2000 で使用するためのデバイスドライバです。

適応OS	Windows NT4.0(サービスパック 4 推奨) Windows 2000
適応言語	Visual C++ Ver5.0 以上 Visual Basic(32bit) Ver5.0 以上
適応機種	IBM PC/AT 互換機、DOS/V 機 NEC PC98-NX シリーズ
最大ボード数	16 枚
対象ボード	AI48-01 AnyWireBus-DB ISA バス I/F AI48-01-Y07 AnyWireBus-DB ISA バス I/F AI28-01 AnyWireBus-HX ISA バス I/F AP48-01 AnyWireBus-DB PCI バス I/F

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。

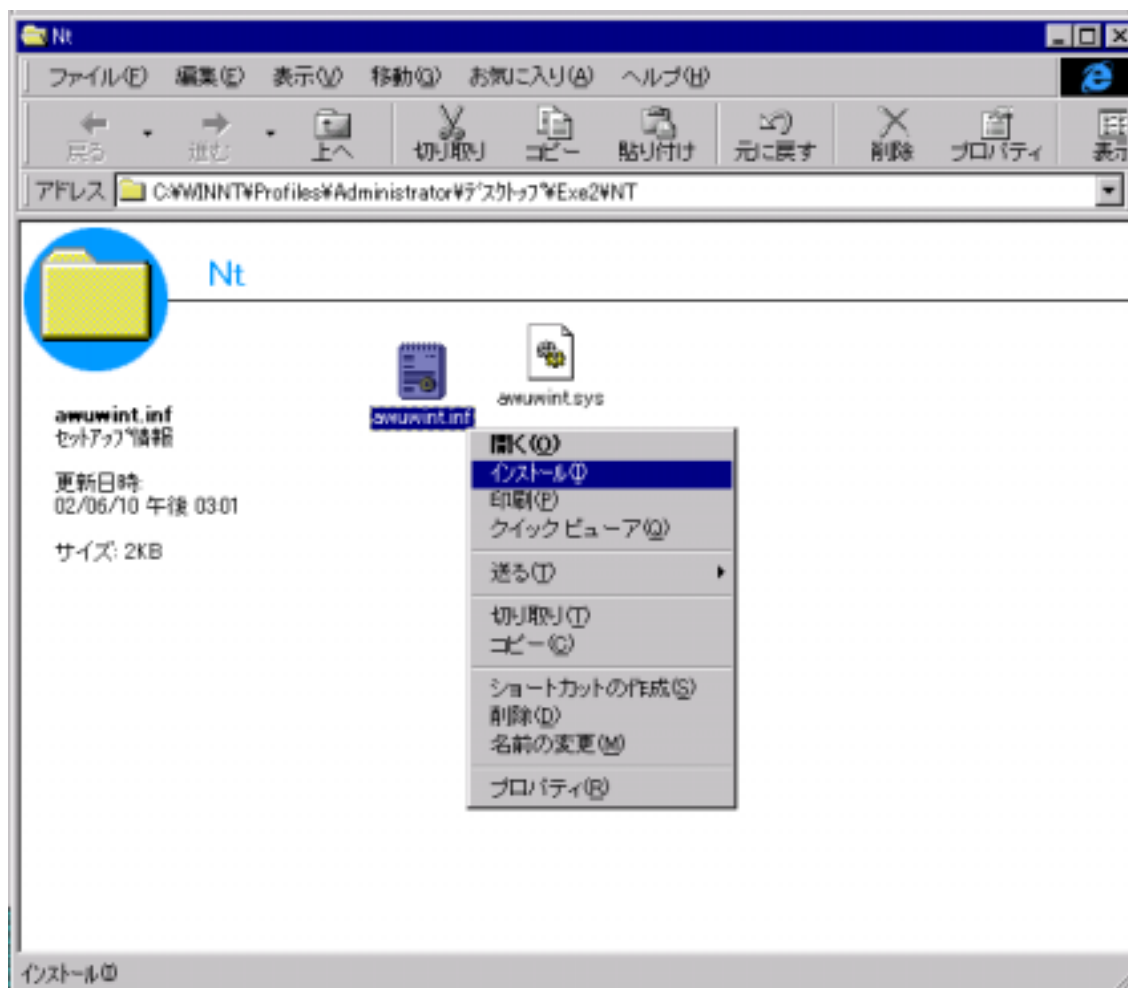
Awuwint.inf	NT 用インストール用セットアップ情報ファイル
Awuwi2k.inf	2000 用インストール用セットアップ情報ファイル
Awuwint.sys	NT 用インストールシステムファイル
Awuwi2k.sys	2000 用インストールシステムファイル
Awuwi.dll	ドライバ DLL ファイル
AwuwiAPI.lib	C 言語用 LIB ファイル
AwuwiAPI.h	C 言語用ヘッダーファイル
AwuwiAPI.txt	VB 用 API ファイル
AwuwSet.exe	ISA ボード設定アプリケーション

2. ISAバスインターフェースのインストール

2 - 1 windows NT4.0 へのインストール

2-1-1 ドライバのインストール

Windows NT4.0 へインストールするには、[Awwint.inf]ファイルをクリックして表示を反転させ、[ファイル(F)]の[インストール(I)]を実行してください。



パソコンを再起動すればインストール完了です。

本ソフトウェアをアプリケーションで使用するには、[Awwi.dll]ファイルを[Winnt]の中か、[アプリケーションがあるディレクトリ]にコピーしてください。

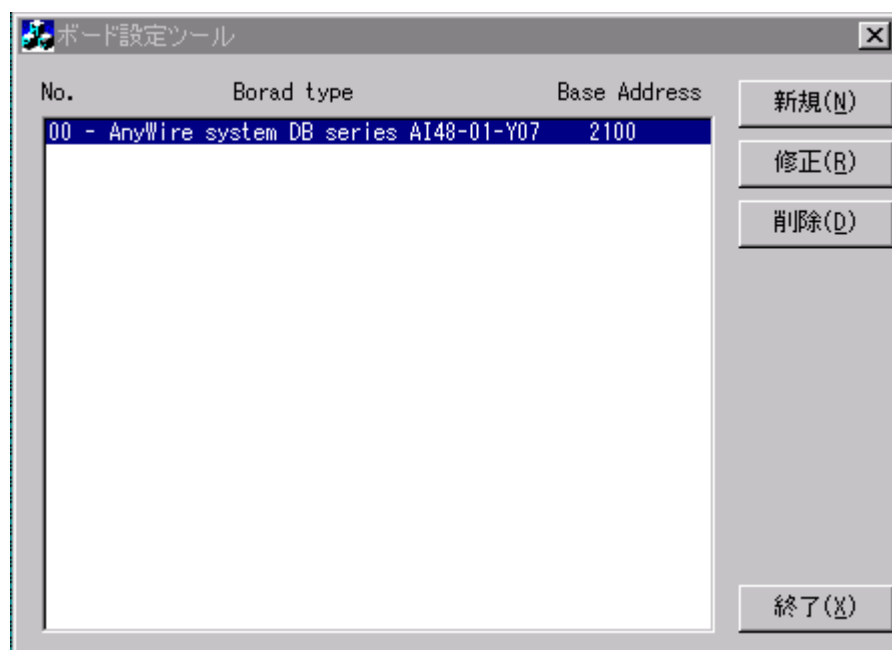
削除するには[マイコンピュータ]の[コントロールパネル][アプリケーションの追加と削除のプロパティ]上の[インストールと削除]で、[ドライバの削除]を行ってください。

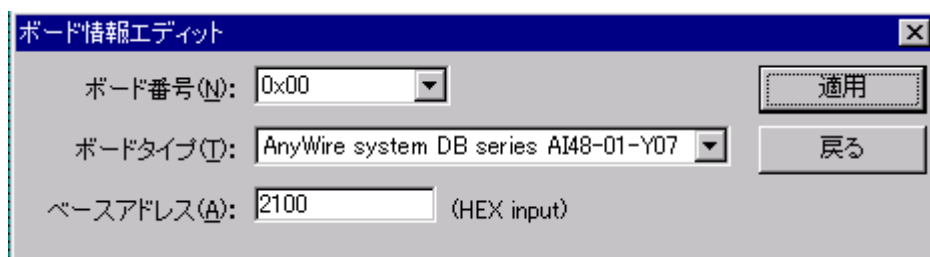


2-1-2 ISAボードの設定

プラグアンドプレイに対応していないISAバスボードは、ボードで設定されたベースのアドレスとボードナンバーを設定してレジストリに書き加える必要があります。その操作を行うには付属のアプリケーション[AwwiSet.exe]を実行してください。

[ボードの追加]で[ボードナンバー]、[ベースアドレス]、[ボードの種類]を設定することが出来、レジストリに書き加えられます。





ボード情報エディット

ボード番号(N): 0x00

ボードタイプ(T): AnyWire system DB series AI48-01-Y07

ベースアドレス(A): 2100 (HEX input)

適用

戻る

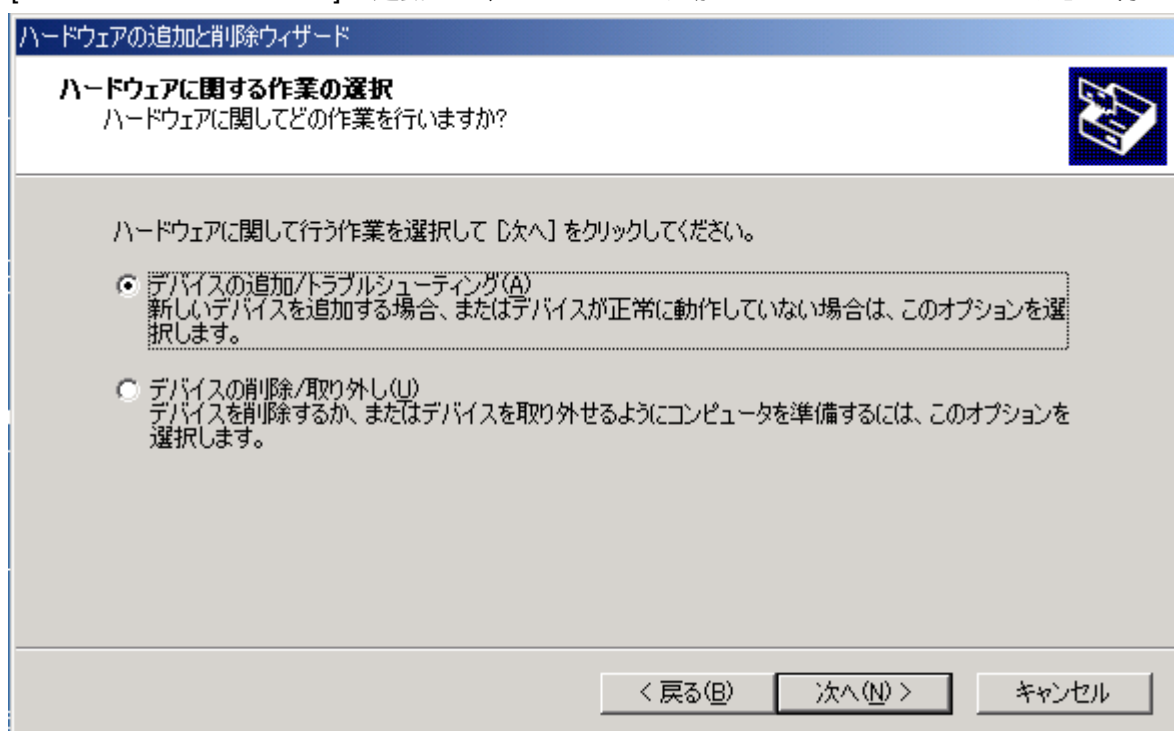
[ボードの変更]で設定した情報を変更できます。これらの変更は再起動することで有効になります。

2 - 2 windows2000 へのインストール

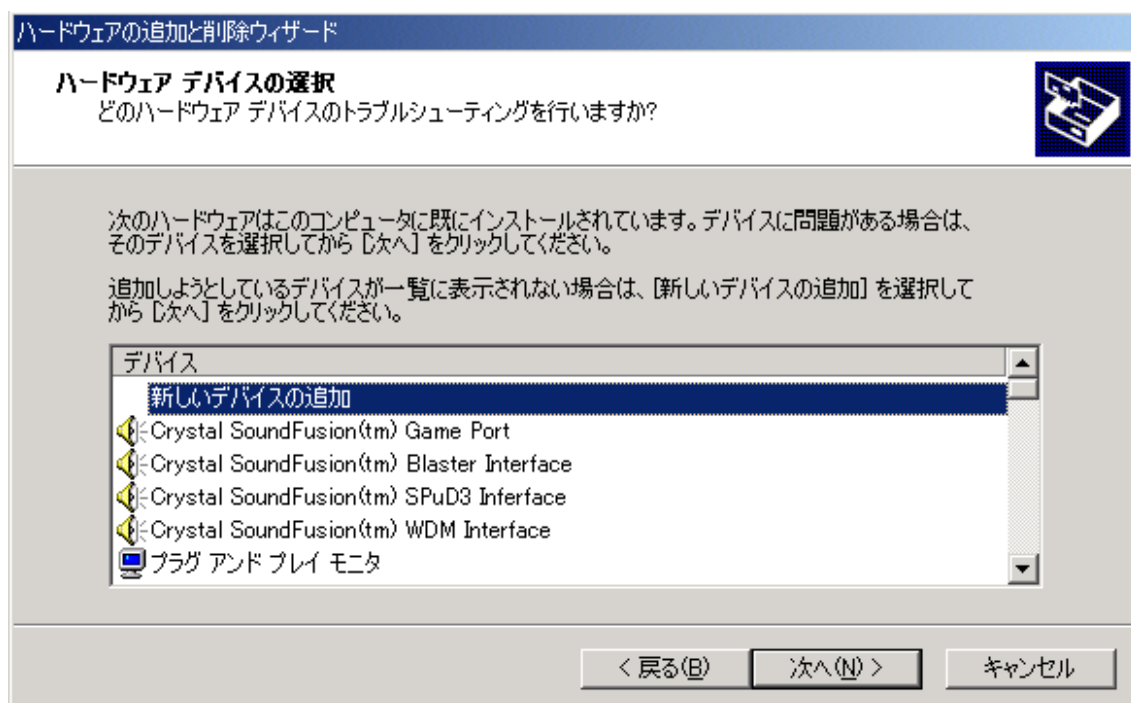
1) ドライバのインストール

Windows2000 へインストールするには、[コントロールパネル]の[システム]の中の[ハードウェアウィザード]を使用します。

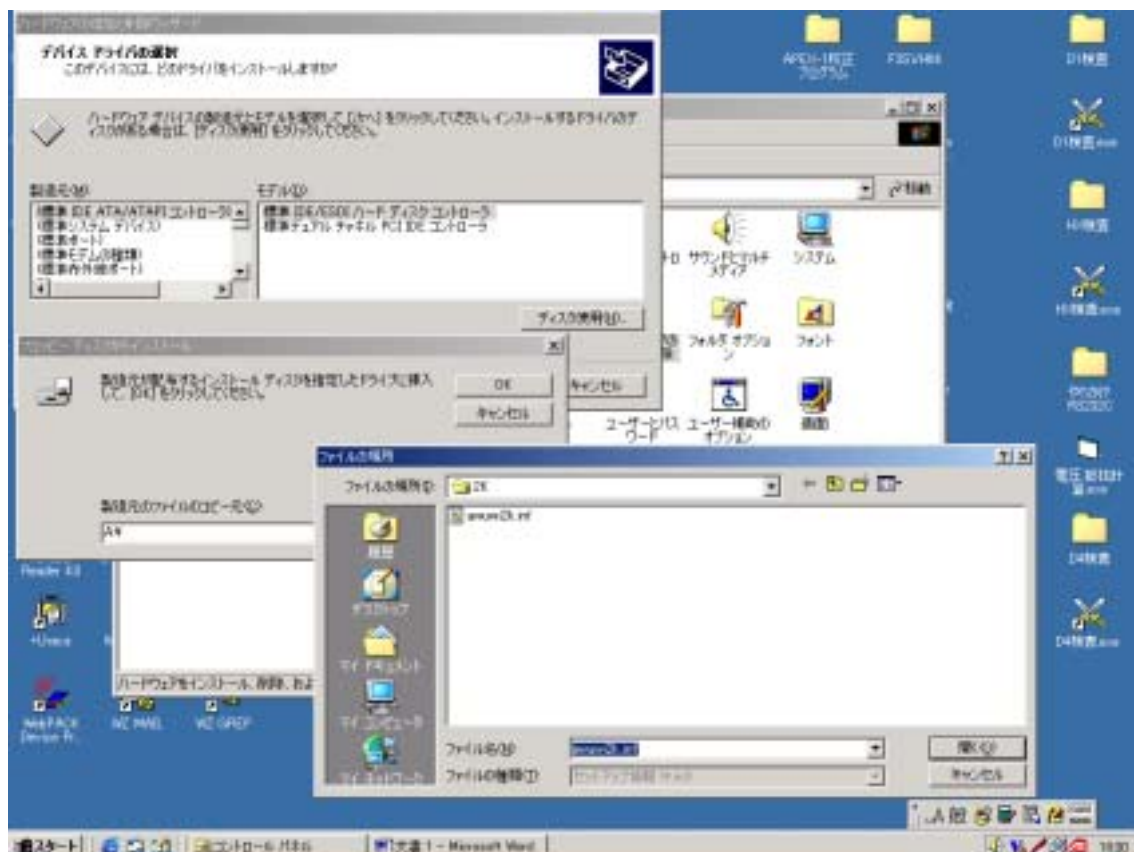
[ハードウェアウィザード]を起動して、「デバイスの追加/トラブルシューティング」を行います。



新しいデバイスの追加を選んで、「いいえ、一覧からハードウェアを選択します」を選択し、[その他のデバイス]を選択し次に進みます。



「デバイスドライバの選択」で[ディスク使用]をクリックし[参照 (B)]でドライバソフトの[¥2k¥awuwi2k.inf]を参照します。[OK]を押し、AnyWireSystem DB series AI48-01 を選択します。



ハードウェアの追加と削除ウィザード

デバイス ドライバの選択

このデバイスには、どのドライバをインストールしますか？



ハードウェア デバイスの製造元とモデルを選択して [次へ] をクリックしてください。インストールするドライバのディスクがある場合は、[ディスク使用] をクリックしてください。

モデル(D):

AnyWire system DB series AI48-01
AnyWire system HX series AI28-01

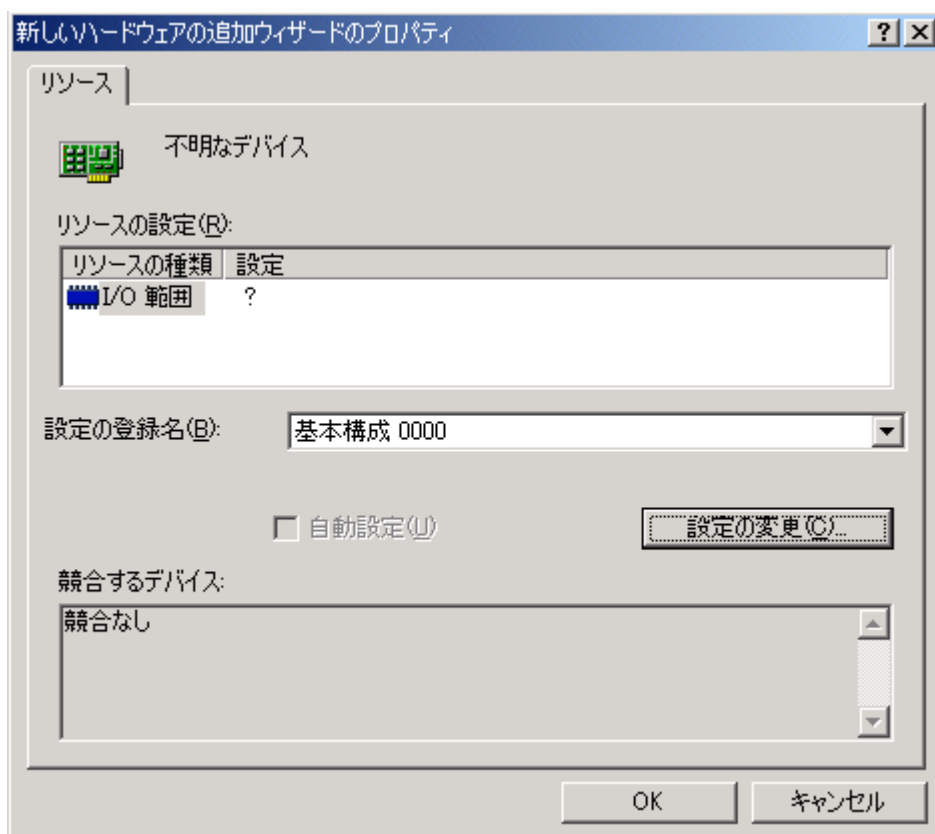
ディスク使用(H)...

< 戻る(B)

次へ(N) >

キャンセル

[リソースの設定]で他のデバイスで使用していないところに設定します。競合するデバイスがなければ[インストールを開始]するを選びますとインストールが開始します。



パソコンを再起動すればインストール完了です。

本ソフトウェアをアプリケーションで使用するには、[Awuwi.dll] ファイルを [Winnt] の中か、[アプリケーションがあるディレクトリ] にコピーしてください。

削除するには同じくハードウェアウィザードを使ってデバイスの削除を行ってください。

2) ISAボードの設定

プラグアンドプレイに対応していないISAバスボードは、ボードで設定されたベースのアドレスとボードナンバーを設定してレジストリに書き加える必要があります。その操作を行うには付属のアプリケーション [AwuwiSet.exe] を実行してください。

前項で設定されたデバイスが表示されます。

[ボードの修正] で [ボードナンバー] を仕様にあわせて設定してください。これらの変更は再起動することで有効になります。

ボードのベースアドレスの設定はこのツールではできません。[コントロールパネル] の [システム] [ハードウェア] の デバイスマネージャー で 目的のボードを右クリックし、リソースの再設定を行ってください。これらの変更は再起動することで有効になります。

3. PCIバスインターフェースのインストール

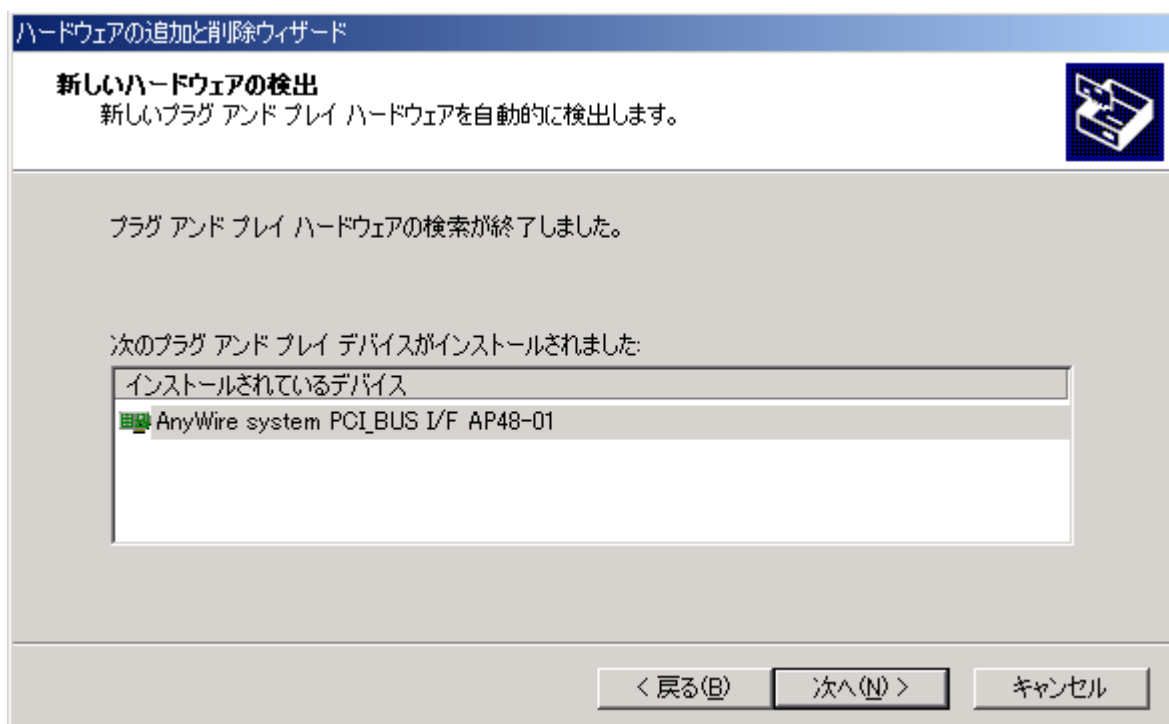
本ボードをパソコンのP C Iバスに挿入して、電源を入れると plug & play 機能が働き、ハードウェア追加ウィザードが起動されます。

起動されない場合は、[設定] [コントロールパネル] [ハードウェアの追加と削除]をダブルクリックして起動してください。



本ボードのドライバの所在を確認してきますので、windows2000 の場合、¥DEV¥EXE¥2k フォルダにある、【awuwi2k.inf】ファイルを、windowsNT の場合は¥DEV¥EXE¥NT フォルダの【awuwint.inf】を指定してください。

インストールが正常に終了すると、以下のように表示されインストールの終了です。



4. インターフェース関数

本インターフェイスは、AnyWireBus インターフェイスボードを操作する上で必要な関数を提供します。関数の種類は、下記の表に示す 6 つになります。

関数名	定義名称	説明
ボード情報取得関数	awuwi_boardinfo	指定したボード No の詳細情報を取得します。
ボード別ポート入力関数 (バイト)	awuwi_din	指定したボード No のポートからバイト単位データを入力します。
ボード別ポート出力関数 (バイト)	awuwi_dout	指定したボード No のポートからバイト単位データを出力します。
ボード別ポート入力関数 (ワード)	awuwi_dinw	指定したボード No のポートからワード単位データを入力します。
ボード別ポート出力関数 (ワード)	awuwi_doutw	指定したボード No のポートからワード単位データを入力します。
ボード情報取得関数	awuwi_briget	ドライバの起動に必要なボードの情報を取得します。
ボード情報設定関数	awuwi_briset	ドライバの起動に必要なボードの情報を設定します。
ボード情報削除関数	awuwi_bridel	ドライバの起動に必要なボードの情報を削除します。

4 - 1 個別ボードの詳細情報を見る場合は？

使用中のパソコンに増設されている AnyWireBus インターフェイスボードの詳細な設定情報を見る場合は、下記の表に示す関数を使用します。

void awuwi_boardinfo(int nPmBno,PSPMAWUW_BIFpsPmBif)		
パラメータ	変数名	説 明
Int	nPmBno	ボード No 0x00 ~ 0x0F (16 進数)
PSPMAWUW_BIF	psPmBif	ボード情報構造体格納先アドレス 構造体の内容は、ボード情報構造体を参照してください。
リターン値		説 明

C 言語記述

上記の関数を使用してインストールされているボードの詳細情報を調べるサンプル例を C 言語と VisualBasic で下記に示します。

・C 言語の場合

```
SPMAWUW_BIF sBif;
char szMsg[256];
int i;
for(i=0;i<DPMWUW_B___V_MAX___;i++) {
    awuwi_boardinfo(i,&sBif);                // ボード情報読込
    if(!sBif.ulBtp) continue;                // ?ボード無
    sprintf(szMsg,"ボード No. %d は、アドレスが 0x%04X になります。",
            sBif.ulBno,sBif.ulBad);
    MessageBox(hWnd,szMsg,"メッセージボックス",MB_OK);
}
```

・Visual Basic の場合

```
'外部関数参照宣言
Declare Sub awuwi_boardinfo Lib "awuwi.dll" (ByVal ulPmBno as Long,ByRef sBif As SPMAWUW_BIF)
Dim sBif As SPMAWUW_BIF
for i=0 to DPMWUW_B___V_MAX___-1
    awuwi_boardinfo i, sBif
    if sBif.ulBtp==0 then Goto Jmp1
    'メッセージ出力
    MsgBox "ボード No." + Format(i) + "は、アドレスが 0x" + Hex(sBif.ulBad) + "になります。"
Jmp1:
Next i
```

4 - 2 ボードに対してデータ読み込みを行う場合は？（バイト単位）

ボードに対してデータのバイト単位の読み込みを行う場合は、下記の表に示す関数を使用します。

UCHAR awuwi_din(int nPmBno,int nPmPno)		
パラメータ	変数名	説 明
Int	nPmBno	ボード No 0x00 ~ 0x0F (16 進数)
Int	NPmPno	ポート No HX(AI28-01) : 0x0000 ~ 0x003F (16 進数) 0x0400 ~ 0x043F 0x0800 ~ 0x083F 0x0C00 ~ 0x0C3F DB (AI48-01): 0x0000 ~ 0x0003 (16 進数) DB (AI 48-01-Y07) : 0x0000 ~ 0x008F (16 進数) DB (AI 48-01-Y09) : 0x0000 ~ 0x01FF (16 進数) AP48-01 : 0x0000 ~ 0x01FF (16 進数)
リターン値		説 明
UCHAR		読み込みデータ 0x00 ~ 0xFF (16 進数)

C 言語記述

上記の関数を使用したボードに対するデータの読み込みサンプル例を C 言語と VisualBasic で下記に示します。

（ボード No : 0x05 (16 進数) ポート No : 0x16 (16 進数) のデータを読み込む場合)

・ C 言語の場合

```
UCHAR ucVal;           // 変数宣言
char szMsg[256];
ucVal = awuwi_din (0x05,0x16);    // データの読み込み
                                   // メッセージ出力
wsprintf(szMsg,"ボード No が 0x05 ・ ポート No が 0x16 のデータの読み込み結果は、0x%02X です。",ucVal);
MessageBox(hWnd,szMsg,"メッセージボックス",MB_OK);
```

・ Visual Basic の場合

```
'外部関数参照宣言
Declare Function awuwi_din Lib "awuwi.dll" (ByVal ulPmBno As Long,ByVal ulPmPno As Long) As Byte
Dim bVal As Byte
bVal = awuwi_din(CLng(&h05), CLng(&h16))    'データの読み込み
                                   'メッセージ出力
MsgBox "ボード No が 0x05 ・ ポート No が 0x16 のデータの読み込み結果は、0x" + Hex(bVal) + "です。"
```

4 - 3 ボードに対してデータ書き込みを行う場合は？（バイト単位）

ボードに対してデータのバイト単位の書き込みを行う場合は、下記の表に示す関数を使用します。

void awuwi_dout(int nPmBno,int nPmPno,UCHAR ucPmVal)		
パラメータ	変数名	説 明
Int	nPmBno	ボード No 0x00 ~ 0x0F (16 進数)
Int	NpmPno	ポート No HX(AI28-01): 0x0000 ~ 0x003F (16 進数) 0x0400 ~ 0x043F 0x0800 ~ 0x083F 0x0C00 ~ 0x0C3F DB (AI48-01): 0x0000 ~ 0x0003 (16 進数) DB (AI 48-01-Y07): 0x0000 ~ 0x008F (16 進数) DB (AI 48-01-Y09): 0x0000 ~ 0x01FF (16 進数) AP48-01: 0x0000 ~ 0x01FF (16 進数)
UCHAR	ucPmVal	出力データ 0x00 ~ 0xFF (16 進数)
リターン値		説 明

C 言語記述

上記の関数を使用したボードに対するデータの書き込みサンプル例を C 言語と VisualBasic で下記に示します。

（ボード No : 0x05 (16 進数) ポート No : 0x16 (16 進数) に対してデータ : 0xAB を書き込む場合)

・ C 言語の場合

```

    UCHAR ucVal;                // 変数宣言
    ucVal = 0xAB;                // 出力データ格納
    awuwi_dout(0x05,0x16,ucVal); // データの出力
                                // メッセージ出力
    sprintf(szMsg,"ボード No が 0x05・ポート No が 0x16 にデータ値 0x%02X を出力しました。",ucVal);
    MessageBox(hWnd,szMsg,"メッセージボックス",MB_OK);

```

・ Visual Basic の場合

```

'外部関数参照宣言
Declare Sub awuwi_dout Lib "awuwi.dll" (ByVal uIPmBno As Long,ByVal uIPmPno As Long,ByVal bPmVal As Byte)
Dim bVal As Byte                '変数宣言
bVal = &hAB                    '出力データ格納
awuwi_dout CLng(&h05), CLng(&h16), bVal 'データの書き込み
'メッセージ出力
MsgBox "ボード No が 0x05・ポート No が 0x16 にデータ値 0x" + Hex(bVal) + "を出力しました。"

```

4 - 4 ボードに対してデータ読み込みを行う場合は？（ワード単位）

ボードに対してデータのワード単位の読み込みを行う場合は、下記の表に示す関数を使用します。

USHORT awuwi_dinw(int nPmBno,int nPmPno)		
パラメータ	変数名	説 明
Int	nPmBno	ボード No 0x00 ~ 0x0F (1 6 進数)
Int	NPmPno	ポート No HX(AI28-01): 0x0000 ~ 0x003F (1 6 進数) 0x0400 ~ 0x043F 0x0800 ~ 0x083F 0x0C00 ~ 0x0C3F DB (AI48-01): 0x0000 ~ 0x0003 (1 6 進数) DB (AI 48-01-Y07): 0x0000 ~ 0x008F (1 6 進数) DB (AI 48-01-Y09): 0x0000 ~ 0x01FF (1 6 進数) AP48-01: 0x0000 ~ 0x01FF (1 6 進数)
リターン値		説 明
USHORT		読み込みデータ 0x0000 ~ 0xFFFF (1 6 進数)

C 言語記述

上記の関数を使用したボードに対するデータの読み込みサンプル例を C 言語と VisualBasic で下記に示します。

(ボード No : 0x05 (16 進数) ポート No : 0x16 (16 進数) のデータを読み込む場合)

・ C 言語の場合

```
USHORT usVal;           // 変数宣言
char szMsg[256];
usVal = awuwi_dinw(0x05,0x16);    // データの読み込み
// メッセージ出力
wsprintf(szMsg,"ボード No が 0x05 ・ ポート No が 0x16 のデータの読み込み結果は、0x%04X です。",usVal);
MessageBox(hWnd,szMsg,"メッセージボックス",MB_OK);
```

・ Visual Basic の場合

```
'外部関数参照宣言
Declare Function awuwi_dinw Lib "awuwi.dll" (ByVal uIPmBno As Integer,ByVal IPmPno As Integer) As Integer
Dim usVal As Integer      '変数宣言
usVal = awuwi_dinw(CLng(&h05), CLng(&h16)) 'データの読み込み
'メッセージ出力
MsgBox "ボード No が 0x05 ・ ポート No が 0x16 のデータの読み込み結果は、0x" + Hex(usVal) + "です。"
```

4 - 5 ボードに対してデータ書き込みを行う場合は？（ワード単位）

ボードに対してデータのワード単位で書き込みを行う場合は、下記の表に示す関数を使用します。

void awuwi_doutw(int nPmBno,int nPmPno,USHORT usPmVal)		
パラメータ	変数名	説 明
Int	nPmBno	ボード No 0x00 ~ 0x0F (1 6 進数)
Int	NPmPno	ポート N o HX(AI28-01) : 0x0000 ~ 0x003F (1 6 進数) 0x0400 ~ 0x043F 0x0800 ~ 0x083F 0x0C00 ~ 0x0C3F DB (AI48-01): 0x0000 ~ 0x0003 (16 進数) DB (AI 48-01-Y07) : 0x0000 ~ 0x008F (1 6 進数) DB (AI 48-01-Y09) : 0x0000 ~ 0x01FF (1 6 進数) AP48-01 : 0x0000 ~ 0x01FF (1 6 進数)
USHORT	usPmVal	出力データ 0x0000 ~ 0xFFFF (16 進数)
リターン値		説 明

C 言語記述

上記の関数を使用したボードに対するデータの書き込みサンプル例を C 言語と VisualBasic で下記に示します。

（ボード N o : 0x05 (16 進数) ポート N o : 0x16 (16 進数) に対してデータ : 0xABCD を書き込む場合）

・ C 言語の場合

```
USHORT usValue;           // 変数宣言
usVal = 0xABCD;           // 出力データ格納
awuwi_doutw (0x05,0x16,usVal); // データの出力
                           // メッセージ出力
wsprintf(szMsg,"ボード N o が 0x05 ・ ポート N o が 0x16 にデータ値 0x%04X を出力しました。",usVal);
MessageBox(hWnd,szMsg,"メッセージボックス",MB_OK);
```

・ Visual Basic の場合

```
'外部関数参照宣言
Declare Sub awuwi_doutw Lib "awuwi.dll" (ByVal uIPmBno As Integer,ByVal uIPmPno As Integer,ByVal bPmVal As Integer)
Dim usVal As Integer      '変数宣言
usValue = &hABCD          '出力データ格納
awuwi_doutw CLng(&h05), CLng(&h16), usPmVal 'データの書き込み
                           'メッセージ出力
MsgBox "ボード N o が 0x05 ・ ポート N o が 0x16 にデータ値 0x" + Hex(bValue) + "を出力しました。"
```


4 - 6 ボード情報の初期値を読み込む場合は？

使用中のパソコンにインストールされている AnyWireBus インターフェイスボードのドライバ起動時に使用する設定情報を読み込む場合は、下記の表に示す関数を使用します。

ULONG awuwi_briget (PSPMAWUWI_BIF psPmBif)		
パラメータ	変数名	説 明
PSPMAWUWI_BIF	psPmBif	ボード No 0x00 ~ 0x0F (16 進数 ボード情報構造体格納先アドレス 構造体の内容は、ボード情報構造体を参照してください。
リターン値		
ULONG		ボード情報取得数(MAX:16)

C 言語記述

上記の関数を使用してボード初期化用に設定されている情報を読み込むサンプル例を C 言語と VisualBasic で下記に示します。

・C 言語の場合

```
SPMAWUWI_BIF sBif[DPMWUWI_B___V_MAX___];           // 変数宣言
char szMsg[256];
int i;
int nCnt = awuwi_briget(&sBif);                       // 設定値読み込み
for(i=0; i<nCnt; i++) {                               // ボード数計算
    wsprintf(szMsg, "ボード No. %d は、ベースアドレスが 0x%04X に設定されています。",
        sBif.ulBno, sBif.ulBad);
    MessageBox(hWnd, szMsg, "メッセージボックス", MB_OK);
}
```

・Visual Basic の場合

```
'外部関数参照宣言
Declare Sub awuwi_briget Lib "awuwi.dll" (ByRef psBif As SPMAWUWI_BIF)
Dim sBif(DPMWUWI_B___V_MAX___-1) As SPMAWUWI_BIF
nCnt = awuwi_briget(sBif(0))
for i=0 to nCnt-1
    'メッセージ出力
    MsgBox "ボード No." + Format(sBif.ulBno) + "は、ベースアドレスが 0x" + Hex(sBif.ulBad) + _
        "に設定されています。"
Next i
```

4 - 7 ボードの初期値を書き込む場合は？

使用中のパソコンにインストールされている AnyWireBus インターフェイスボードのドライバ起動時に使用する設定情報を書き込む場合は、下記の表に示す関数を使用します。

void awuwi_briset (PSPMAWUWI_BIF psPmBif)		
パラメータ	変数名	説 明
PSPMAWUWI_BIF	psPmBif	ボード No 0x00 ~ 0x0F (1 6 進数) ボード情報構造体格納先アドレス 構造体の内容は、ボード情報構造体を参照してください。
リターン値		説 明

上記の関数を使用してボード初期化用に設定する情報を書き込むサンプル例を C 言語と VisualBasic で下記に示します。(ボード No が 0x01 にベースアドレス 0x0300 の DB ボードを割り当てる場合)

・ C 言語の場合

```
SPMAWUWI_BIF sBif; // 変数宣言
// 変数初期化
sBif.ulBno = 1; // ボード番号設定
sBif.ulBad = 0x0300; // ベースアドレス設定
sBif.ulBtp = DPMWUWI_BTP_V_DB____; // ボードタイプ設定

awuwi_briset(&sBif); // 設定値書き込み
```

・ Visual Basic の場合

```
'外部関数参照宣言
Declare Sub awuwi_briset Lib "awuwi.dll" (ByRef psBif As SPMAWUWI_BIF)
Dim sBif As SPMAWUWI_BIF
'変数初期化
sBif.ulBno = 1 'ボード番号設定
sBif.ulBad = &h300 'ベースアドレス設定
sBif.ulBtp = DPMWUWI_BTP_V_DB____ 'ボードタイプ設定

awuwi_briset sBif '設定値書き込み
```

注意)

Windows95,2000 でデバイスのボード No を設定する場合は、設定を行うデバイスのコントロールパネルで割り当てたリソースのベースアドレスと上記の引数にある「ボード情報構造体」内の「ベースアドレス」を一致させる必要があります。一致していない場合には、ボード No を設定することができません。

4 - 8 ボードの初期情報を消すには？

使用中のパソコンにインストールされている AnyWireBus インターフェイスボードのドライバ起動時に使用する設定情報を消去する場合は、下記の表に示す関数を使用します。

Void awuwi_bridel (nPmBno)		
パラメータ	変数名	説 明
Int	nPmBno	ボード No 0x00 ~ 0x0F (16進数)
リターン値		説 明

上記の関数を使用してボード初期化用に設定する情報を削除するサンプル例を C 言語と VisualBasic で下記に示します。(ボード No が 0x01 を削除する場合)

・ C 言語の場合

```
awuwi_bridel(1); // 設定値書き込み
```

・ Visual Basic の場合

```
'外部関数参照宣言
Declare Sub awuwi_bridel Lib "awuwi.dll" (ByRef nPmBno As long)

awuwi_bridel CLng(1) '設定値書き込み
```

注意)

本関数は、WindowsNT でのみ動作します。Windows95,2000 では、デバイスマネージャーからのデバイス削除によりボード情報を消去します。

5. ボード情報構造体

このボード詳細情報構造体は、起動時に必要な情報を設定する場合や起動中のドライバにより個別に割り当てたボードのリソースの情報を格納する構造体になります。使用関数は、「awuwi_boardinfo」・「awuwi_briget」・「awuwi_briset」のみの使用となります。

構造体の内容については、下記の表に示します。

・C 言語の場合

構造体型名	SPMAWUWI_BIF	
ポインタ型名	PSPMAWUWI_BIF	
メンバ		
変数型	変数名	説明
ULONG	ulBno	ボードの番号が設定されます。 0x00 ~ 0x0F (16 進数)
ULONG	ulBad	ボードのベースアドレスが設定されます。 0x00000000 ~ 0xFFFFFFFF (16 進数)
ULONG	UIBtp	ボードの種類が設定されます。 HX AI28-01 : DPMWUWI_BTP_HX____ (0x10) DB AI48-01 : DPMWUWI_BTP_DB____ (0x11) DB AI 48-01-Y07: DPMWUWI_BTP_DBY07_ (0x12) DB AI 48-01-Y09: DPMWUWI_BTP_DBY09_ (0x13) AP48-01: DPMWUWI_BTP_AP4801 (0x20)

・VisualBasic の場合

構造体型名	SPMAWUWI_BIF	
メンバ		
変数型	変数名	説明
Long	ulBno	ボードの番号が設定されます。 0x00 ~ 0x0F (16 進数)
Long	ulBad	ボードのベースアドレスが設定されます。 0x00000000 ~ 0xFFFFFFFF (16 進数)
Long	ulBtp	ボードの種類が設定されます。 HX AI28-01 : DPMAWUWI_BTP_HX____ (0x10) DB AI48-01 : DPMAWUWI_BTP_DB____ (0x11) DB AI 48-01-Y07: DPMAWUWI_BTP_DBY07_ (0x12) DB AI 48-01-Y09: DPMAWUWI_BTP_DBY09_ (0x13) AP48-01: DPMAWUWI_BTP_AP4801 (0x20)

6. 注意事項

本インターフェイスを使用する場合、下記に示す点にご注意ください。

- ・ご使用になる場合には、必ず「Awuwi.dll」が必要となります。
- ・C言語でご使用になる場合には、「AwuwiApi.h」と「Awuwi.lib」ファイルが必要となります。
「AwuwiApi.h」ファイルは、使用するプロジェクトにインクルードして下さい。
「Awuwi.lib」ファイルは、使用するプロジェクトにリンクして下さい。
- ・C++言語でご使用する場合には、上記のC言語で使用する場合の項目とあわせて「CPLUSPLUS」をプリプロセッサ定義してください。
- ・VisualBasicをご使用になる場合は、提供ファイルの「AwuwiApi.TXT」ファイルをAPIビューアでご使用下さい。

7. 更新履歴

日付	バージョン	内容
2003.3.26	ver2.0	P C Iバス I / F 追加初版



株式会社 エニワイヤ

URL <http://www.anywire.jp>
e-mail info@anywire.jp

本社
〒617-0813
京都府長岡京市井ノ内下印田8番1
TEL 075-956-1611
FAX 075-956-1613
東京営業所
〒101-0035
東京都千代田区神田紺屋町47番
新広栄ビル7F
TEL 03-5209-5711
FAX 03-5209-5713

■ご用命は