

株式会社エニイワイヤ

エニイワイヤインターフェイスボード(D2.DB)用

Windows95・NT・2000・XPドライバインターフェイス

取扱説明書

0. 目次

0. 目次	2
1. インターフェイス関数	3
-1. 個別ボードの詳細情報を見る場合は？	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）	5
-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）	6
-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）	7
-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）	8
-6. ボードの初期値を読み込む場合は？	9
-7. ボードの初期値を書き込む場合は？	10
-8. ボードの初期情報を消すには？	11
2. 構造体	
-1. ボード情報構造体	12
3. 注意事項	13
4. 修正履歴	14

1. インターフェイス関数

本インターフェイスは、エニイワイヤ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
ボード情報会得関数	awuwi_boardinfo	指定したボード No の詳細情報を会得します。
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。
ボード情報会得関数	awuwi_briget	ドライバの起動に必要なボードの情報を会得します。
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。

1-1. 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エニイワイ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

```
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

    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
```

1-2. 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: 0x0000~0x003F (16進数) 0x0400~0x043F 0x0800~0x083F 0x0C00~0x0C3F DB: 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) AP48-02: 0x0000~0xFFFF (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 ' 変数宣言
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。"
```

1-3. 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: 0x0000~0x003F (16進数) 0x0400~0x043F 0x0800~0x083F 0x0C00~0x0C3F DB: 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) AP48-02: 0x0000~0xFFFF (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
wsprintf(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 ulPmBno As Long, ByVal ulPmPno 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。"
```

1-4. 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 (16進数)
int	NPmPno	ポートNo HX: 0x0000~0x003F (16進数) 0x0400~0x043F 0x0800~0x083F 0x0C00~0x0C3F DB: 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) AP48-02: 0x0000~0xFFFF (16進数)
リターン値		説明
USHORT		読み込みデータ 0x0000~0xFFFF (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

```
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 uPmBno As long, ByVal lPmPno As long) 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。"
```

1-5. 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 (16進数)
int	NPmPno	ポートNo HX: 0x0000~0x003F (16進数) 0x0400~0x043F 0x0800~0x083F 0x0C00~0x0C3F DB: 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) AP48-02: 0x0000~0xFFFF (16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No: 0x05 (16進数) ポートNo: 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Noが0x05・ポートNo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 ulPmBno As Long, ByVal ulPmPno As Long, 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Noが0x05・ポートNoが0x16にデータ値0x" + Hex (bValue) + "を出力しました。"
```

1-6. 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エニワイ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
```

1-7. 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エニワイ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 (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		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。

1-8. 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エニワイヤ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

```
Declare Sub awuwi_bridel Lib "awuwi.dll" (ByRef nPmBno As long) ' 外部関数参照宣言
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からのデバイス削除によりボード情報を消去します。

2-1. 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

構造体名	SPMAWUWI_BIF	
ポインタ名	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 : DPMAWUWI_BTP_HX___ (0x10) DB : 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) AP48-02: DPMAWUWI_BTP_AP4802 (0x21)

・VisualBasicの場合

構造体名	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 : DPMAWUWI_BTP_HX___ (0x10) DB : 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) AP48-02: DPMAWUWI_BTP_AP4802 (0x21)

3. 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下さい。
- ・ 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。



株式会社 エニイワイヤ

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

■ 本 社

〒617-8550 京都府長岡京市馬場図所 1
TEL 075-956-1611(代) FAX 075-956-1613

■ 西日本営業所

〒617-8550 京都府長岡京市馬場図所 1
TEL 075-956-4911 FAX 075-956-1613

■ 東日本営業所

〒101-0035 東京都千代田区神田紺屋町 47 番地
新広栄ビル6F
TEL 03-5209-5711 FAX 03-5209-5713

■ 中部営業所

〒461-0048 愛知県名古屋市東区矢田南 5-1-14
TEL 052-452-8711 FAX 052-452-8713

■ 九州営業所

〒810-0001 福岡県福岡市中央区天神 1-15-2
第 6 明星ビル 7F
TEL 0942-46-9811 FAX 0942-46-9813