

ユニラインインターフェースボード用

Windows95/98/NT4.0/2000

ドライバソフト 取扱説明書

Version 2.1

N K E 株式会社

Copyright© NKE Corporation 2004

EWINDRV-800D

ご注意

本書の内容に関しましては将来予告なしに変更することがあります。

本書の一部または全部を無断で転載することは禁止されています。

本書および本ソフトウェアを使用した結果につきましては責任を負いかねますので、あらかじめご了承ください。

リソースおよびレジストリの操作につきましては責任を負いかねますので、十分気をつけて行ってください。

MS-DOS、Windows、Microsoft C/C++、Visual Basic、Visual C++ は、米国マイクロソフト社の登録商標です。
Borland C/C++ は、米国ボーランド社の登録商標です。

はじめに

このたびは本システム機器をお買い上げ頂きまして誠にありがとうございます。

本ソフトウェアは、ユニラインインターフェースボードを Windows 95 / 98 / NT4.0 / 2000 上で 使用するための
デバイスドライバです。下記の表に示す仕様になります。

項目	仕様 (確認済)
適応 OS	Windows 95 / Windows 98 Windows NT4.0 (サービスパック 4 以上 推奨) Windows 2000
適応言語	Visual C++ Ver 5.0 Visual Basic (32bit) Ver 5.0
適応機種	IBM PC/AT 互換機、DOS/V 機 [動作確認機種] NEC PC98-NX シリーズ (Windows NT Ver4.0 サービスパック 3 以上必要 差分モジュール 推奨) DELL DIMENSION 4200 (Windows NT Ver4.0 サービスパック 3 以上必要)
最大ボード枚数	16 枚
対象ボード	AT-UW / AT-HUW AT-H500 AT-H250 (AT-H500 として登録) PCI-HUW

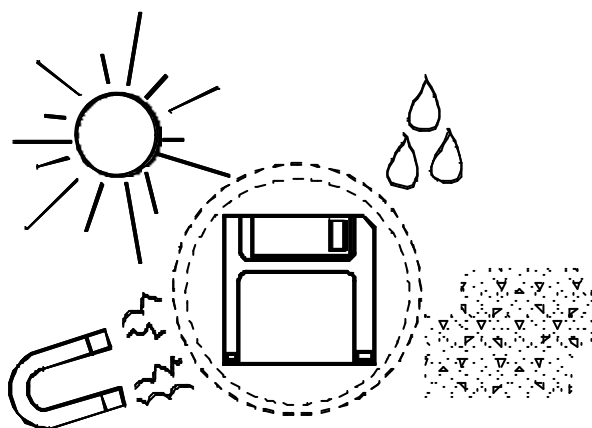
注意

添付フロッピーディスクの README ファイルに、フロッピーディスクの内容、および最新情報が載せてありますので、
必ずお読みください。

本ソフトウェアが収められているフロッピーディスクの取り扱いには十分注意してください。

直射日光や高温、高湿、水やホコリ、磁気 には近づけないでください。

または、これらから十分保護して、使用、保管してください。



目次

はじめに	3
1 . Windows 95/98 へのインストールと削除	4
- 1 . PCI-Bus ボードの登録	5
- 2 . ISA-Bus ボードの登録	6
2 . Windows NT Ver 4.0 へのインストールと削除	9
3 . Windows 2000 へのインストールと削除	11
- 1 . PCI-Bus ボードの登録	12
- 2 . ISA-Bus ボードの登録	13
4 . インターフェース関数	15
- 1 . ボードの有無を見る	16
- 2 . 個別ボードの詳細情報を見る	17
- 3 . ボードに対してデータを読み込む	18
- 4 . ボードに対してデータを書き込む	19
- 5 . ISA-Bus ボードの初期値を読み込む	20
- 6 . ISA-Bus ボードの初期値を書き込む	21
5 . 構造体	
- 1 . 個別ボード詳細情報構造体	22
- 2 . ISA-Bus レジストリ情報構造体	23
6 . 注意事項	24
7 . サンプルプログラム	25
8 . 変更履歴	26

1 .Windows 95/98 へのインストールと削除

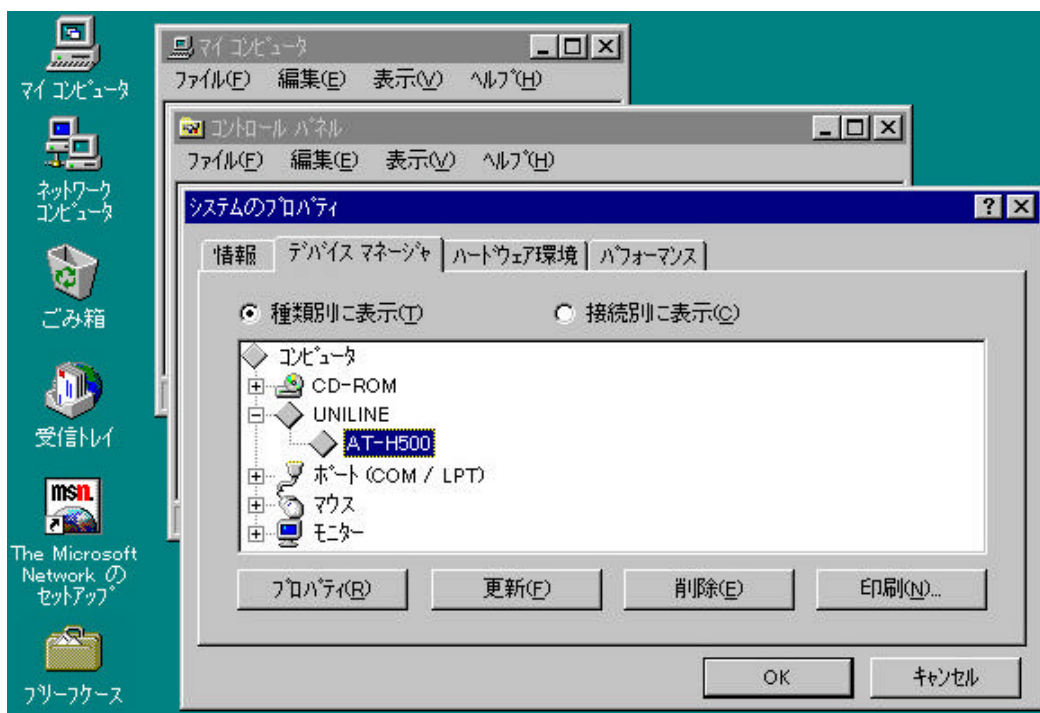
Windows 95 には、バージョン 4.00.950、4.00.950 a、4.00.950B があり、インストールの方法が異なります。
Windows 95、98 とともに、バージョンとは、[マイコンピュータ] の [コントロールパネル] にある [システム] の [システムのプロパティ] のシステム番号です。



PCI-Bus ボード、ISA-Bus ボード を一番はじめに登録する時に、添付のフロッピーディスクの ディレクトリ [Win95] にある [Uwi95.vxd] が [Windows] の [System] にコピーされます。

本ソフトウェアをアプリケーションで使用するには、添付のフロッピーディスクの ルートディレクトリ にある [UwiDll.dll] を [Windows] の中か、[アプリケーションがある同じディレクトリ] にコピーしてください。

削除するには、[コントロールパネル] の [システム] 内の [デバイスマネージャ] の [UNILINE] の削除したいボードを選んで削除してください。



1-1 .PCI-Bus ボードの登録

ボード No (ボードにある SW1) を設定した PCI-Bus 版ユニラインインターフェースボードをスロットに挿入し、パソコンの電源を ON します。同時に複数枚のボードを実装して、Windows 95/98 を起動することも出来ます。

ボード No で、ボードの区別をしますので特に注意してください。

Windows 95 4.00.950、4.00.950 a の場合

Windows 95 を起動すると、[新しいハードウェア] の検出の画面が表示されます。

添付のフロッピーディスクをディスクドライブに入れて、[ハードウェアの製造元が提供するドライバ(M)] を選択し、[フロッピーディスクからのインストール] でフロッピーディスクのドライブ名とディレクトリ名 [Win95] を入れ、指示に従い操作が終了すると、インストール完了です。

Windows 95 4.00.950 B の場合

Windows 95 を起動すると、[新しいハードウェア] の検出の画面が表示されます。

添付のフロッピーディスクをディスクドライブに入れて、[デバイスドライバウィザード] の ドライバの検索の [場所の指定 (O)] を選択し、[場所 (L)] にフロッピーディスクのドライブ名とディレクトリ名 [Win95] を入れ、指示に従い操作が終了すると、インストール完了です。

Windows 98 (4.10.1998) の場合

Windows 98 を起動すると、[新しいハードウェア] の検出の画面が表示されます。

[新しいハードウェアの追加ウィザード] の [使用中のデバイスに最適なドライバを検索する] を選び、添付のフロッピーディスクをディスクドライブに入れて、検索場所を [フロッピーディスクドライブ (F)] [検索場所の指定 (L)] をチェックし、フロッピーディスクのドライブ名とディレクトリ名 [Win95] を入れ、指示に従い操作が終了すると、インストール完了です。

2 枚目以降

それぞれの ボード No (ボードにある SW1) は他のボード No と重複しないように設定してください。

Windows 95/98 を起動すると、[新しいハードウェア] の検出の画面が表示され、OS が自動的にインストールを行い、次の [新しいハードウェア] の検出をし、自動的にインストールを行います。ボードの枚数これを繰り返します。

リソース

ボードに自動的に割り当てられたリソース (I/O アドレス) を見るには、[コントロールパネル] の [システム] 内の [デバイスマネージャ] の [UNILINE] で、目的のボードを選び、[プロパティ (R)] の [リソース] で見ます。設定を変更する場合は、[自動設定 (U)] を解除して、アドレスの競合がないように設定してください。

(リソースの画面は、「1-2 ISA-Bus ボードの登録」 リソース の画面を参考にしてください。)

1-2 .ISA-Bus ボードの登録

プラグアンドプレイに対応していないISA-Bus ボードは、実際のボードがあるなしにかかわらず登録できます。

リソースで割り当てられるアドレスと、実際に使用するユニラインインターフェースボードのアドレスを一致させないと正常に動作しません。ご注意ください。

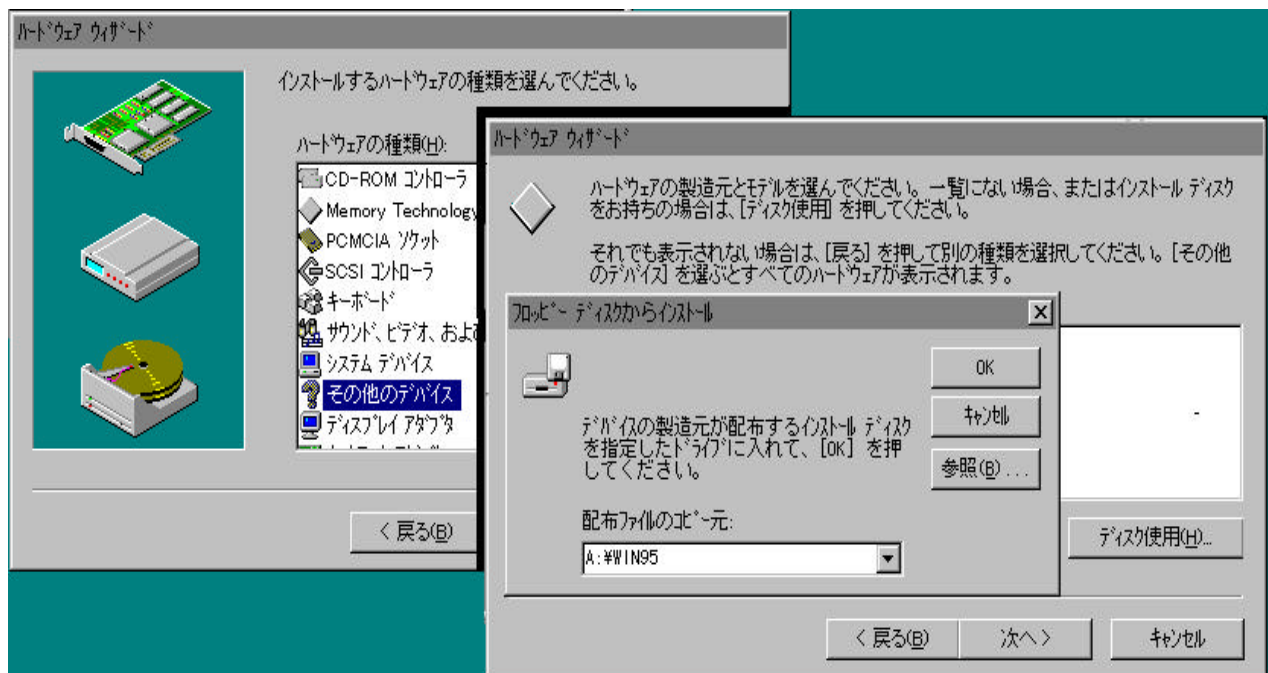
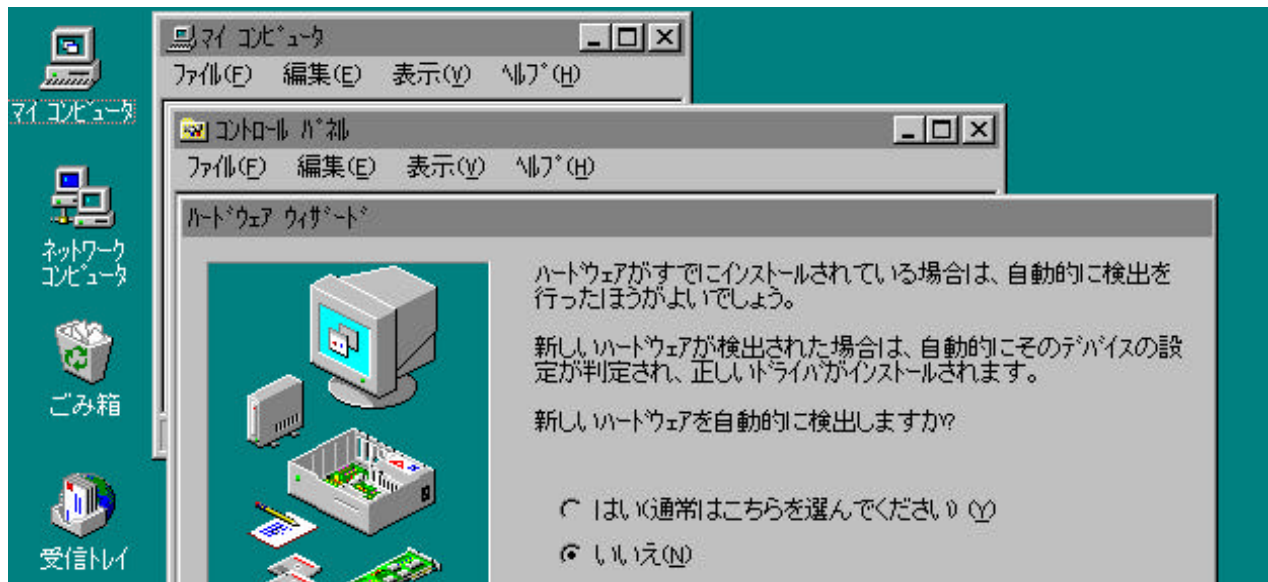
Windows 95 の場合

ISA-Bus 版ユニラインインターフェースボードをスロットに挿入し、パソコンの電源を ON します。

Windows 95 を起動し、[マイコンピュータ] の [コントロールパネル] にある [ハードウェア] をダブルクリックして、[ハードウェア ウィザード] を起動します。

新しいハードウェアの自動検出を [いいえ] にして、[ハードウェアの種類] を [その他のデバイス] にします。

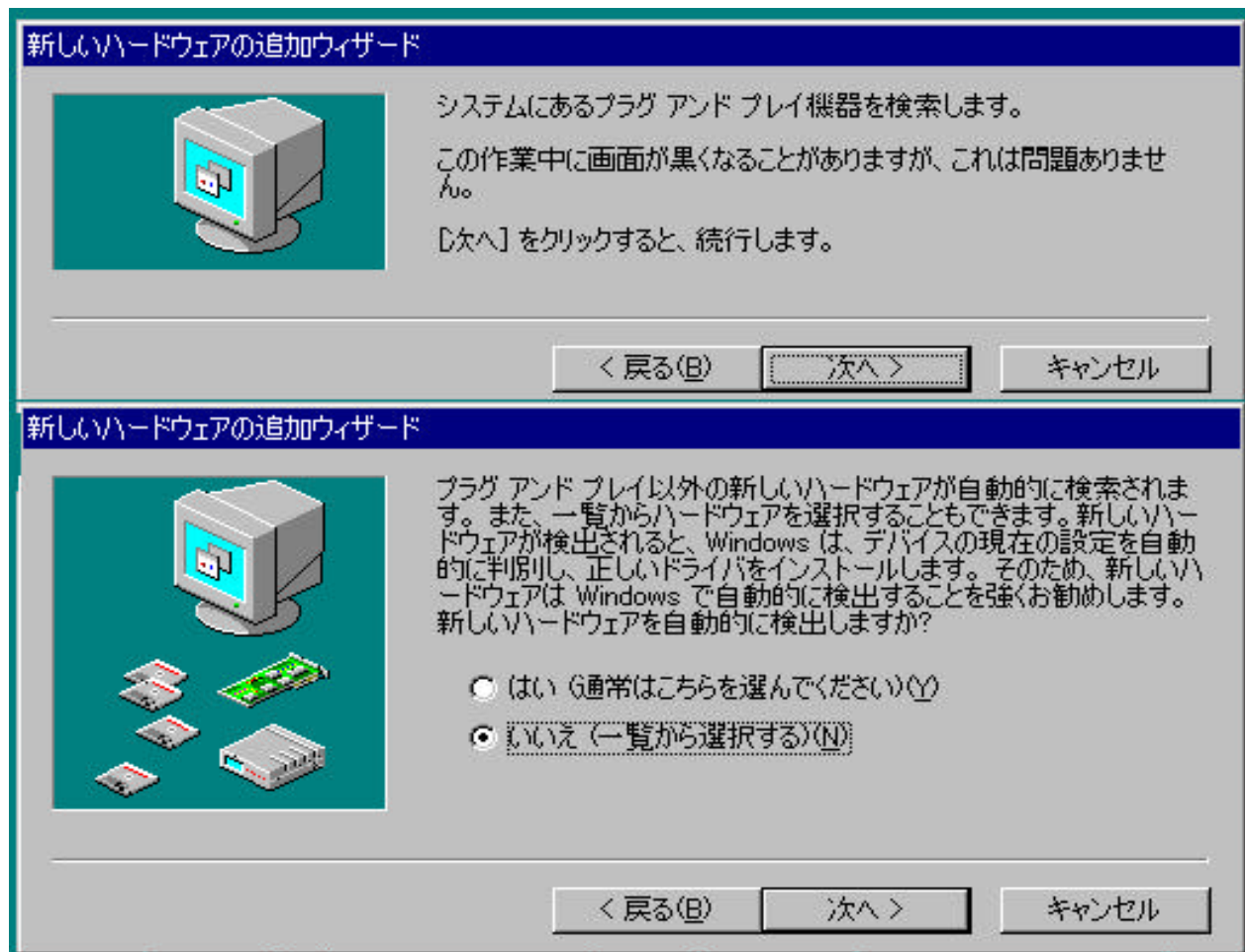
[ディスク使用(H)] を選択し、ドライブ名と、ディレクトリ名 [Win95] をいれて、現れる [モデル]の中から、設定するボードを選ぶと、リソースが割り当てられます。



Windows 98 の場合

ISA-Bus 版ユニラインインターフェイスボードをスロットに挿入し、パソコンの電源を ON します。

Windows 98 を起動し、[マイコンピュータ] の [コントロールパネル] にある [ハードウェア] をダブルクリックして、[ハードウェアの追加ウィザード] を起動します。プラグアンドプレイ機器の検索が終了した後、プラグアンドプレイ機器以外の新しいハードウェアの自動検出を [いいえ] にして、[ハードウェアの種類] を [その他のデバイス] にします。[ディスク使用(H)] を選択し、ドライブ名と、ディレクトリ名 [Win95] をいれて、現れる [モデル] の中から、設定するボードを選ぶと、リソースが割り当てられます。

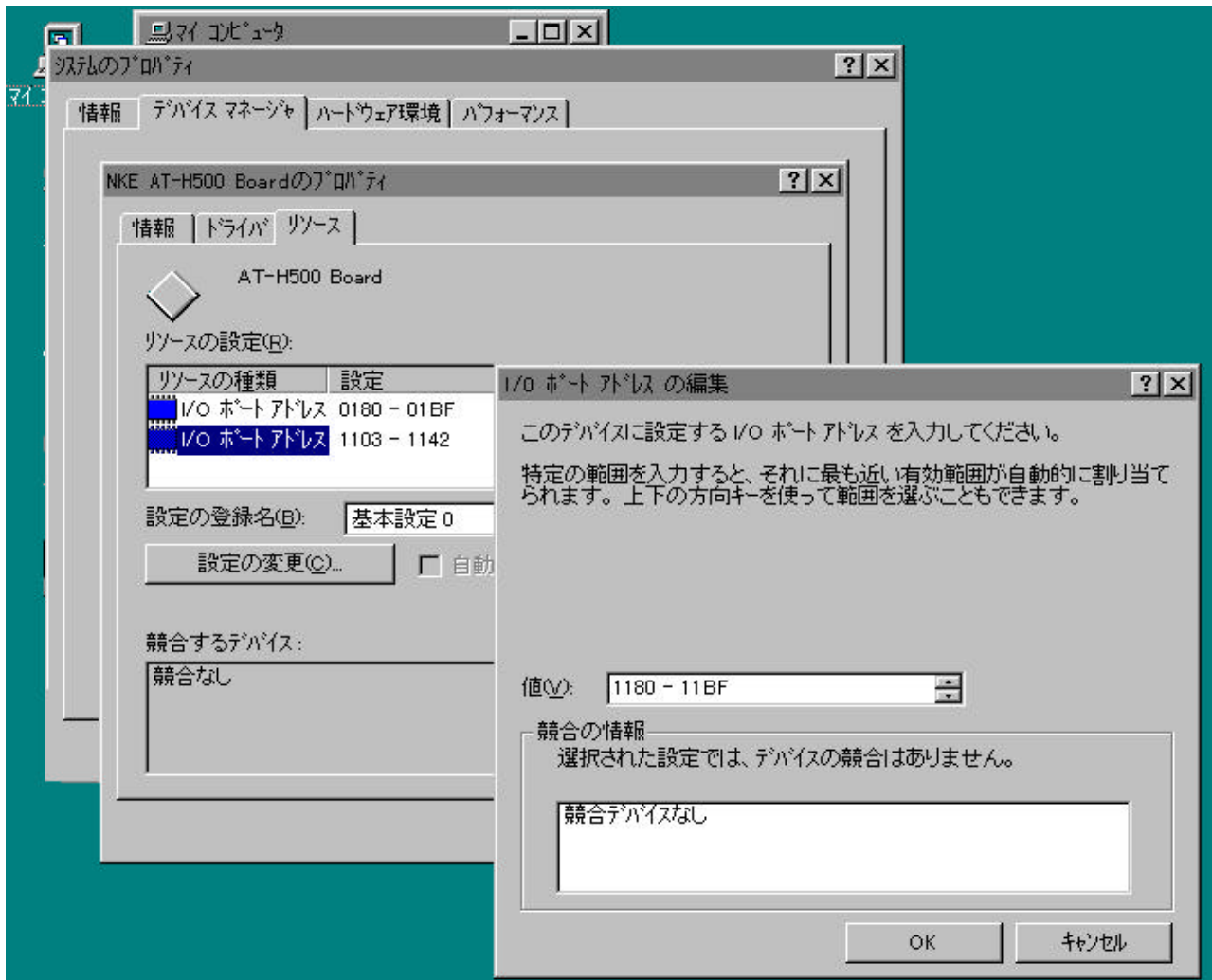


リソースの設定

自動的に割り当てられたリソース (I/O アドレス) を [コントロールパネル] の [システム] 内の [デバイスマネージャ] の [UNILINE] で、ボードを選び、[プロパティ (R)] の [リソース] で見ます。

設定をボードのディップスイッチと同じアドレスに変更するために、[自動設定 (U)] を解除して、アドレスを変えてください。この時、他との競合 (特に同じユニラインインターフェースボード) がないように設定してください。

また、AT-H500 (AT-H250) では、2 段目のアドレスが 1 段目のアドレス + 1000 になるように設定してください。



2 枚目以降

ボードのアドレス (ボードにあるディップスイッチ) を他のボードと重ならないように設定してください。

1 枚目と同じように、Windows 95 の [ハードウェア ウィザード] または、Windows 98 の

[ハードウェアの追加ウィザード] を起動し、インストールを行います。

ただし、[ハードウェアの種類] を [UNILINE] にすると、フロッピーディスクを使用する必要なく、[モデル]の中から、設定するボードを選べます。

アドレス変更時、同じ種類のボードを競合デバイスとして表示しませんので注意してください。

2 .Windows NT Ver 4.0 へのインストールと削除

Windows NT Ver4.0 へインストールするには、添付のフロッピーディスクの ディレクトリ [Nt40] にある [Uwint.inf] 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。

パソコンを再起動させれば、インストール完了です。

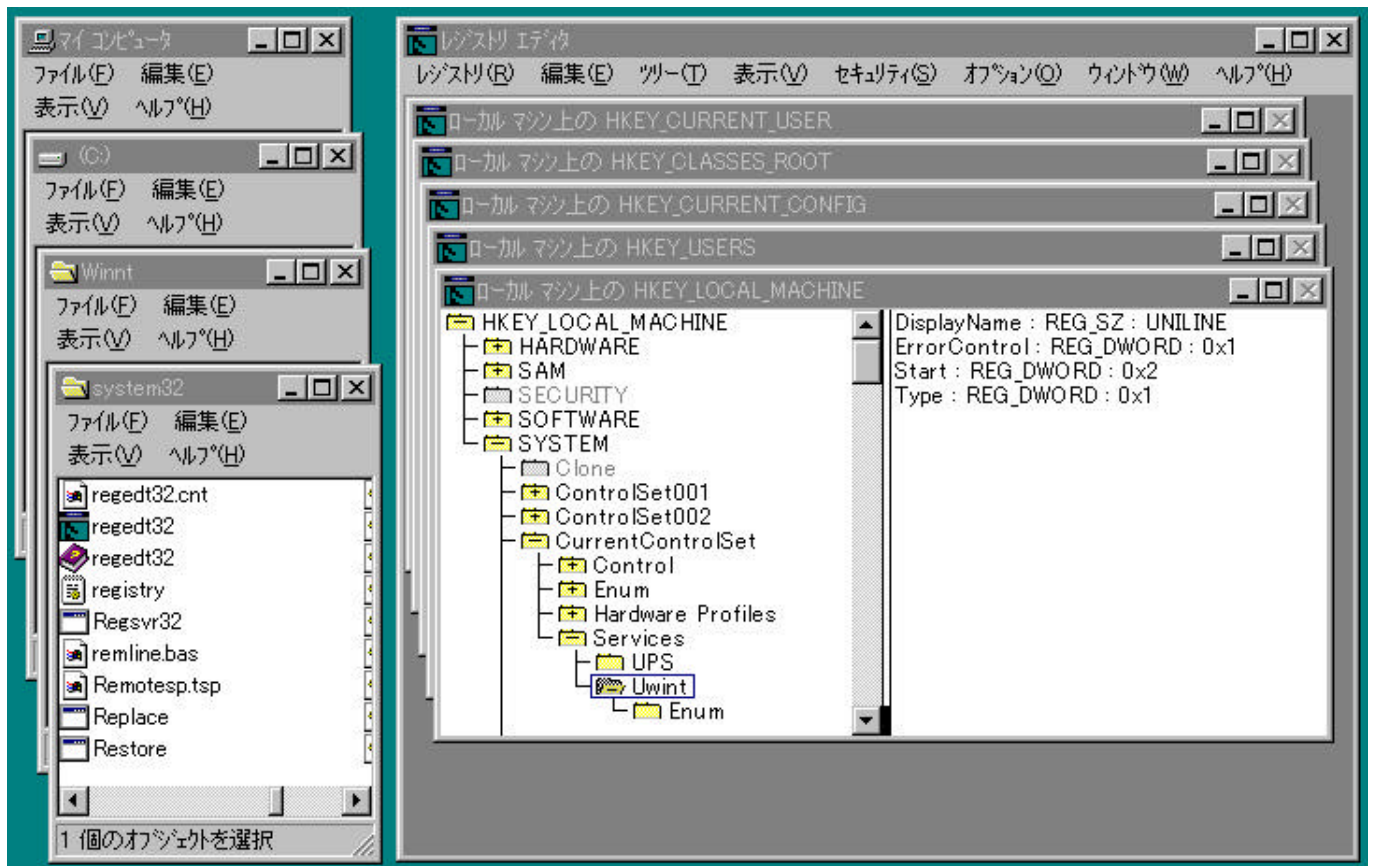
本ソフトウェアをアプリケーションで使用するには、添付のフロッピーディスクの ルートディレクトリ にある [UwiDll.dll] を [Winnt] の中か、[アプリケーションがある同じディレクトリ] にコピーしてください。



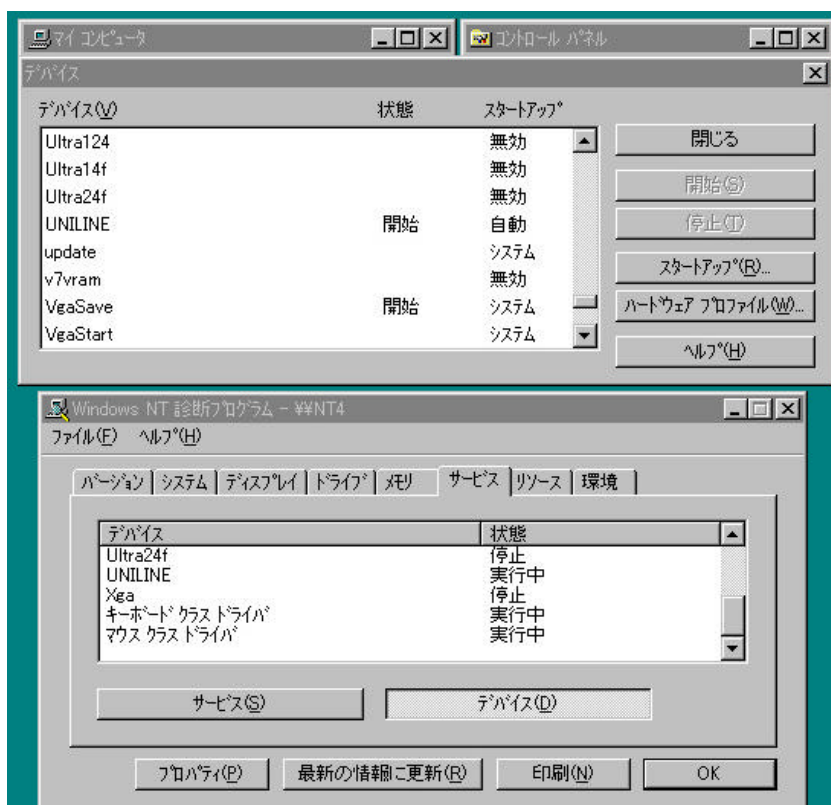
削除するには [マイコンピュータ] の [コントロールパネル] [アプリケーションの追加と削除のプロパティ] 上の [インストールと削除] で、[Remove UNILINE Interface Board Driver] を実行してください。



ボード情報を個別に削除するには、[マイコンピュータ] の [Winnt] の [system32] にある [regedt32] をダブルクリックして、レジストリエディタを起動し、[ローカルマシン上の HKEY_LOCAL_MACHINE] の [SYSTEM] 内の [CurrentControlSet] の [Services] の [Uwint] を、または その下のファイルの中で削除したいボードを、削除してください。レジストリの操作により Windows が起動、動作しなくなる場合がありますので十分気をつけてください。



ソフトウェアの状態は、[コントロールパネル] の [デバイス] か、[NT 診断プログラム] の [サービス] の [デバイス] で知ることが出来ます。



3 .Windows 2000 へのインストールと削除

PCI-Bus ボード、ISA-Bus ボード を 1 番はじめに登録する時に、添付のフロッピーディスクの ディレクトリ [Win2k] にある [Uwi2k.sys] が、[WINNT] の [System32] の [drivers] にコピーされます。

本ソフトウェアをアプリケーションで使用するには、添付のフロッピーディスクの ルートディレクトリ にある [UwiDll.dll]を [WINNT] の中か、[アプリケーションがある同じディレクトリ] にコピーしてください。

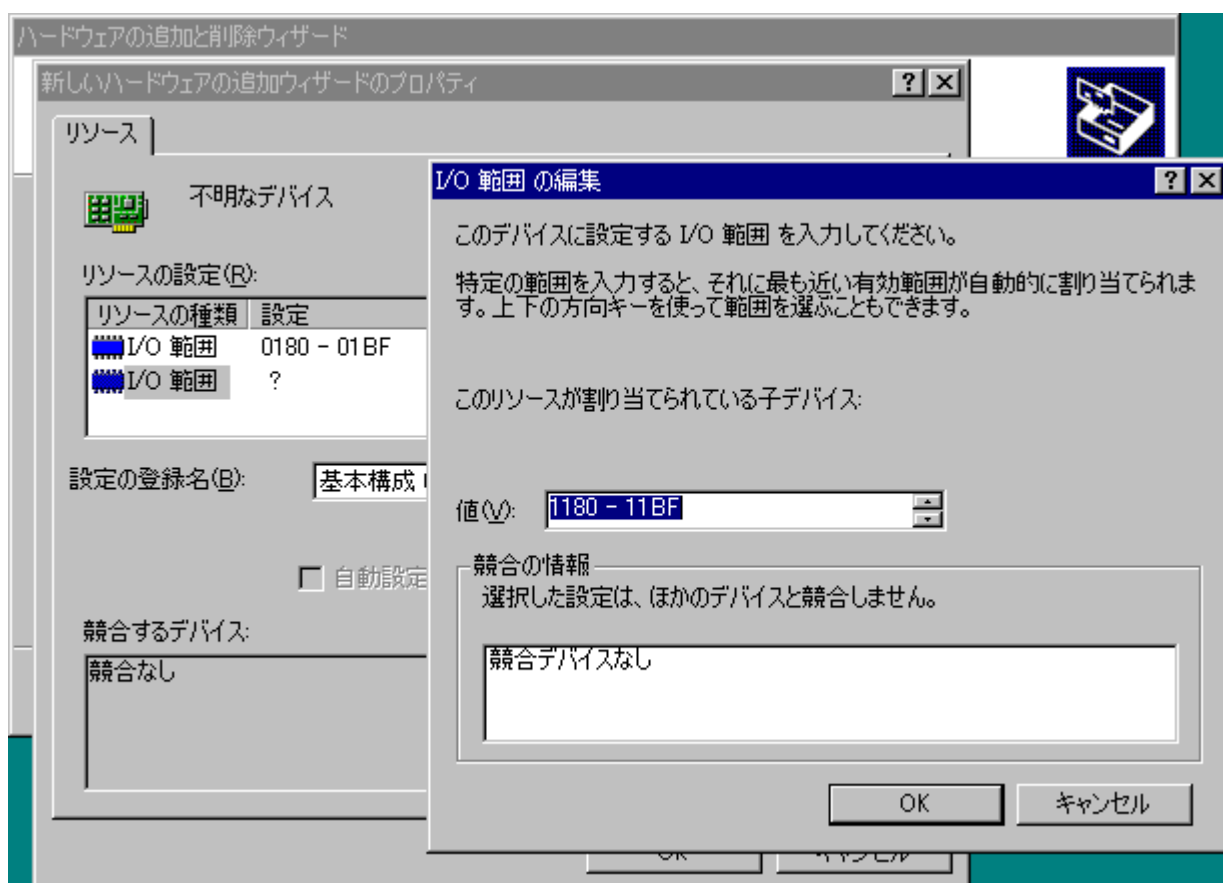
削除するには、[コントロールパネル] の [ハードウェアの追加と削除] で [デバイスの削除(U)] を選択し、削除したいボードを選んで削除してください。

[コントロールパネル] の [システム] 内の [ハードウェア] の [デバイスマネージャ(D)] の [UNILINE] の削除したいボードを選んで削除することが可能です。

リソース

ボードに自動的に割り当てられたリソース (I/O アドレス) を見るには、[コントロールパネル] の [システム] 内の [ハードウェア] の [デバイスマネージャ(D)] の [UNILINE] で目的のボードを選び、[プロパティ (R)] の [リソース] で見ます。設定を変更する場合は、[自動設定 (U)] を解除して、アドレスの競合がないように設定してください。

また、AT-H500 (AT-H250) では、2 段目のアドレスが 1 段目のアドレス + 1000 になるように設定してください。



3-1 .PCI-Bus ボードの登録

ボード No (ボードにある SW1) を設定した PCI-Bus 版ユニラインインターフェースボードをスロットに挿入し、パソコンの電源を ON します。同時に複数枚のボードを実装して、Windows 2000 を起動することも出来ます。

ボード No で、ボードの区別をしますので特に注意してください。

Windows 2000 を起動すると、[新しいハードウェアの検出ウィザード]画面が表示されます。

最適なドライバの検索場所を [フロッピーディスクドライブ (D)] [場所を指定 (S)] をチェックし、フロッピーディスクのドライブ名とディレクトリ名 [Win2k] を入れ、指示に従い操作が終了すると、インストール完了です。



2 枚目以降

それぞれの ボード No (ボードにある SW1) は他のボード No と重複しないように設定してください。

Windows 2000 を起動すると、[新しいハードウェアが見つかりました] の画面が表示され、OS が自動的にインストールを行います。

このとき、Windows 2000 では添付のフロッピーディスクが毎回必要になりますのでご注意ください。

フロッピーディスクのドライブ名とディレクトリ名 [Win2k] を入れ、指示に従い操作が終了すると、

次の [新しいハードウェア] の検出をし、自動的にインストールを行います。ボードの枚数これを繰り返します。

Windows 2000 の新しいハードウェアの検出ウィザードでは、添付フロッピーディスクを、

[‘UNILINE Interface Board Install Disk’のラベルの付いたフロッピーディスク] と呼んでいますのでご注意ください。

3-2 .ISA-Bus ボードの登録

プラグアンドプレイに対応していないISA-Bus ボードは、実際のボードがあるなしにかかわらず登録できます。リソースで割り当てられるアドレスと、実際に使用するユニラインインターフェースボードのアドレスを一致させないと正常に動作しません。ご注意ください。



ISA-Bus 版ユニラインインターフェースボードをスロットに挿入し、パソコンの電源を ON します。

Windows 2000 を起動し、[マイコンピュータ] の [コントロールパネル] にある [ハードウェアの追加と削除] をダブルクリックして、[ハードウェアの追加と削除ウィザード] を起動します。

[デバイスの追加(A)] を選択すると、プラグアンドプレイハードウェアの検索をし、その後ハードウェアデバイスの選択画面になりますので、[新しいデバイスの追加] を選択します。

新しいハードウェアの検索画面で、[いいえ、一覧からハードウェアを選択します。(O)] を選択し、

ハードウェアの種類選択画面で [その他のデバイス] を選択し、[ディスク使用(H)] ボタンを押し、添付のフロッピーディスクをディスクドライブに入れ、ドライブ名とディレクトリ名 [Win2k] を入力し、現れる [モデル]の中から、設定するボードを選ぶと、リソース設定画面になります。

ISA-Bus ボードはプラグアンドプレイに対応していませんので、警告ダイアログが現れます。

リソースについては、「3 Windows 2000 へのインストールと削除」のリソースの項をご参照ください。

リソースの設定の後、ハードウェアのインストールが始まりますので添付のフロッピーディスクはディスクドライブに入れたままにしておいてください。フロッピーディスクが無い場合以下のダイアログが表示されます。



添付のフロッピーディスクをいれれば、インストールが完了します。

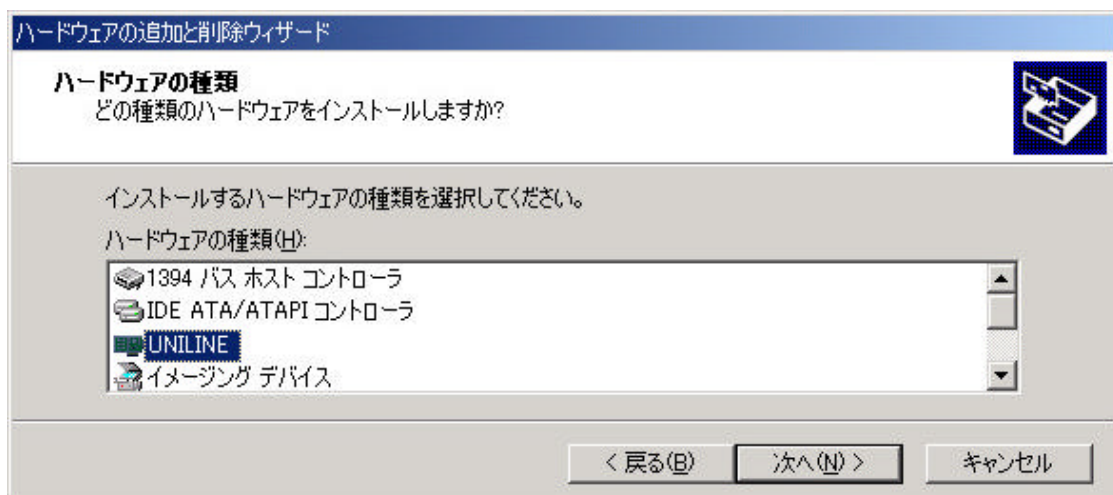
2 枚目以降

ボードのアドレス（ボードにあるディップスイッチ）を他のボードと重ならないように設定してください。

1 枚目と同じように、Windows 2000 の [ハードウェアの追加と削除ウィザード] を起動し、インストールを行います。

ただし、ハードウェアの種類選択画面で [UNILINE] を選択すると、フロッピーディスクを使用する必要なく、[モデル]の中から設定するボードを選べます。

ただし、リソース設定後のハードウェアのインストールでは添付のフロッピーディスクが必要になりますのでご注意ください。



4 . インターフェース関数

本インターフェースは、ユニラインインターフェー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	uw_board	ボードの有無を取得します。
個別ボード詳細情報取得関数	uw_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	uw_din	指定したボード No のポートからデータを入力します。
ボード別ポート出力関数	uw_dout	指定したボード No のポートからデータを出力します。
個別 isa-Bus ボードレジストリ情報取得関数	uw_lsaboardregread	ISA-Bus 用の初期値データを取得します。
個別 isa-Bus ボードレジストリ情報設定関数	uw_lsaboardregwrite	ISA-Bus 用の初期値データを設定します。

4-1 . ボードの有無を見る

使用中のパソコンに増設されているユニラインインターフェースボードの有無を確認する場合は、下記の表に示す関数を使用します。

DWORD uw_board(void)		
パラメータ	変数名	説 明
リターン値		説 明
DWORD		ボード有無情報 (ビット ON : ボード有 ビット OFF : ボード無) 0x00000001 : ボード No 0x00 0x00000002 : ボード No 0x01 0x00000004 : ボード No 0x02 0x00000008 : ボード No 0x03 0x00000010 : ボード No 0x04 0x00000020 : ボード No 0x05 0x00000040 : ボード No 0x06 0x00000080 : ボード No 0x07 0x00000100 : ボード No 0x08 0x00000200 : ボード No 0x09 0x00000400 : ボード No 0x0A 0x00000800 : ボード No 0x0B 0x00001000 : ボード No 0x0C 0x00002000 : ボード No 0x0D 0x00004000 : ボード No 0x0E 0x00008000 : ボード No 0x0F

C 言語記述

上記の関数を使用して増設されているボードの有無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

```

DWORD dwBoard;           // 変数宣言
char szMsg[256];
int i,nCount;
nCount = 0;              // 変数初期化
dwBoard = uw_board();    // ボード有無読み込み
for(i=0;i<16;i++) {      // ボード数計算
    if(dwBoard&(1L<<i)) nCount++;
}
wsprintf(szMsg,"ボードが%d 増設されています。",nCount); // メッセージ出力
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 uw_board Lib "uwidll.dll" () As Long
Dim iCount As Integer    '変数宣言
Dim lBoard As Long
iCount = 0
lBoard = uw_board        'ボード有無読み込み
for i=0 to 15
    if lBoard And (2^i) then iCount = iCount + 1
Next i
MsgBox "ボードが" + Format(iCount) + "増設されています。" 'メッセージ出力

```

4-2 . 個別ボードの詳細情報を見る

使用中のパソコンに増設されているユニラインインターフェー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 uw_boardinfo(int nBoardno,PSUW_BOARDINFO psBoardinfo)		
パラメータ	変数名	説 明
int	nBoardno	ボード No 0x00 ~ 0x0F (16進数)
PSUW_BOARDINFO	psBoardinfo	個別ボード詳細情報構造体格納先アドレス 構造体の内容は、個別ボード詳細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増設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

```

DWORD dwBoard; // 変数宣言
SUW_BOARDINFO sBoardinfo;
char szMsg[256];
int i;
dwBoard = uw_board(); // ボード有無読み込み
for(i=0; i<16; i++) { // ボード数計算
    if(!(dwBoard & (1L<<i))) continue;
    uw_boardinfo(i, &sBoardinfo);
    sprintf(szMsg, "ボード No. %d は、アドレスが 0x%04X になります。",
        i, sBoardinfo.dwAddress);
    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 uw_boardinfo Lib "uwidll.dll" (ByVal lBoardno As Long, ByRef psBoardinfo As PSUW_BOARDINFO)
Dim lBoard As Long
Dim sBoardinfo As PSUW_BOARDINFO
lBoard = uw_board 'ボード有無読み込み
for i=0 to 15
    uw_boardinfo i, sBoardinfo
    'メッセージ出力
    if lBoard And (2^i) then MsgBox "ボード No." + Format(i) + "は、アドレスが 0x" + Hex(sBoardinfo.lAddress) +
        "になります。"
Next i

```

4-3 . ボードに対してデータを読み込む

ボードに対してデータを読み込む場合は、下記の表に示す関数を使用します。

UCHAR uw_din(int nBoardno,int nPortno)		
パラメータ	変数名	説 明
Int	nBoardno	ボード No 0x00 ~ 0x0F (16 進数)
Int	nPortno	ポート No そのボードの先頭アドレスからのアドレス 0x0000 ~ 0x0040 (16 進数) 0x1000 ~ 0x1040 (16 進数) AT-H500 ボードのみ可能
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 ucValue;           // 変数宣言
char szMsg[256];;
ucValue = uw_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。",ucValue);
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 uw_din Lib "uwidll.dll" (ByVal IBoardno As long,ByVal IPortno As long) As Byte
Dim bValue As Byte      '変数宣言
bValue = uw_din(&h05,&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(bValue) + "です。"
```

4-4 . ボードに対してデータを書き込む

ボードに対してデータを書き込む場合は、下記の表に示す関数を使用します。

void uw_dout(int nBoardno,int nPortno, UCHAR ucValue)		
パラメータ	変数名	説 明
Int	Nboardno	ボード No そのボードの先頭アドレスからのアドレス 0x00 ~ 0x0F (16 進数)
Int	Nportno	ポート No 0x0000 ~ 0x0040 (16 進数) 0x1000 ~ 0x1040 (16 進数) AT-H500 ボードのみ可能
UCHAR	UcValue	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 : 0xAC を書き込む場合)

・ C 言語の場合

```

UCHAR ucValue;           // 変数宣言
ucValue = 0xAC;           // 出力データ格納
uw_dout(0x05,0x16,0xAC); // 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。",ucValue);
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 uw_dout Lib "uwidll.dll" (ByVal IBoardno As Long,ByVal IPortno As Long,ByVal bValue As Byte)
Dim bValue As Byte      '変数宣言
bValue = &hAC           '出力データ格納
uw_dout &h05, &h16, bValue '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。"

```

4-5 .ISA-Bus ボードの初期値を読み込む

使用中のパソコンに増設されている ISA-Bus 版のユニラインインターフェー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。

void uw_Isaboardregread (int nBoardno, PSUW_ISAREGSET psIsaRegistry)		
パラメータ	変数名	説 明
int	NBoardno	ボード No 0x00 ~ 0x0F (16 進数)
PSUW_ISAREGSET	PsIsaRegistry	ISA-Bus レジストリ情報構造体格納先アドレス 構造体の内容は、ISA-Bus レジストリ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 ISA-Bus 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

```

SUW_ISAREGSET sIsaregset;                                // 変数宣言
char szMsg[256];
int i;
for(i=0;i16;i++) {                                       // ボード数計算
    uw_Isaboardregread(i, &sIsaregset);                  // 設定値読み込み
    if(!sIsaregset.dwEnable) continue;                   // ?ボード無効
    sprintf(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。",
            i, sIsaregset.dwBaseAddress);
    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 uw_Isaboardregread Lib "uwidll.dll" (ByVal lBoardno as Long,ByRef psIsaregset As SUW_ISAREGSET)
Dim sIsaregset As SUW_ISAREGSET
for i=0 to 15
    uw_Isaboardregread i , sIsaregset
    if sIsaregset.lEnable then
        'メッセージ出力
        MsgBox "ボード No." + Format(i)+ "は、ベースアドレスが 0x" + Hex(sIsaregset.lBaseAddress) + _
            "に設定されています。"
    Next i

```

4-6 .ISA-Bus ボードの初期値を書き込む

使用中のパソコンに増設されている ISA-Bus 版のユニラインインターフェー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 uw_Isaboardregwrite (int nBoardno, PSUW_ISAREGSET psIsaRegistry)		
パラメータ	変数名	説 明
Int	Nboardno	ボード No 0x00 ~ 0x0F (16 進数)
PSUW_ISAREGSET	PsIsaRegistry	ISA-Bus レジストリ情報構造体格納元アドレス 構造体の内容は、ISA-Bus レジストリ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 ISA-Bus 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 0x01 にベースアドレス 0x0300 の AT-H500 ボードを割り当てる場合)

・C 言語の場合

```

SUW_ISAREGSET sIsaregset;                                // 変数宣言
                                                            // 変数初期化
sIsaregset.dwEnable = 1;                                    // 使用許可設定
sIsaregset.dwBaseAddress = 0x0300;                          // ベースアドレス設定
sIsaregset.dwType= 1;                                       // ボードタイプ設定

uw_Isaboardregwrite (i, & sIsaregset);                      // 設定値書き込み

```

・Visual Basic の場合

```

'外部関数参照宣言
Declare Sub uw_Isaboardregwrite Lib "uwidll.dll" (ByVal IBoardno as Long,ByRef psIsaregset As SUW_ISAREGSET)
Dim sIsaregset As SUW_ISAREGSET

sIsaregset.lEnable = 1                                     '変数初期化
sIsaregset.lBaseAddress = &h300                           '使用許可
sIsaregset.lType = 1                                       'ベースアドレス設定
                                                            'ボードタイプ設定

uw_Isaboardregwrite i , sIsaregset                        '設定値書き込み

```

注意

Windows 9 5 / 9 8 / 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「ISA-Bus レジストリ情報構造体」内の「ポートのベースアドレス」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。

5-1 . 個別ボード詳細情報構造体

この個別ボード詳細情報構造体は、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、「uw_boardinfo」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	SUW_BOARDINFO	
ポインタ型名	PSUW_BOARDINFO	
メンバ		
変数型	変数名	説明
DWORD	dwEnable	ボードの有無が設定されます。 ボード有 : DUW_BOARD_ENABLE (1) ボード無 : DUW_BOARD_DISABLED (0)
DWORD	dwAddress	ボードのベースアドレスが設定されます。 0x0000 ~ 0xFFFF (16 進数)
DWORD	dwBustype	バスのタイプが設定されます。 ISA-Bus : DUW_BUS_ISA (0) PCI-Bus : DUW_BUS_PCI (1)
DWORD	dwType	ボードの種類が設定されます。 AT-UW , AT-HUW : DUW_BOARD_ATHUW (0) AT-H500 , AT-H250 : DUW_BOARD_ATH500 (1) PCI-HUW : DUW_BOARD_PCIHUW (9)

・VisualBasic の場合

構造体型名	SUW_BOARDINFO	
メンバ		
変数型	変数名	説明
Long	lEnable	ボードの有無が設定されます。 ボード有 : DUW_BOARD_ENABLE (1) ボード無 : DUW_BOARD_DISABLED (0)
Long	lAddress	ボードのベースアドレスが設定されます。 &h0000 ~ &hFFFF (16 進数)
Long	lBustype	バスのタイプが設定されます。 ISA-Bus : DUW_BUS_ISA (0) PCI-Bus : DUW_BUS_PCI (1)
Long	lType	ボードの種類が設定されます。 AT-UW , AT-HUW : DUW_BOARD_ATHUW (0) AT-H500 , AT-H250 : DUW_BOARD_ATH500 (1) PCI-HUW : DUW_BOARD_PCIHUW (9)

5-2 .ISA-Bus レジストリ情報構造体

この ISA-Bus レジストリ情報情報構造体は、プラグアンドプレイに対応していない ISA-Bus のボードのベースアドレス等の設定を行うための構造体になります。使用関数は、「uw_Isaboardregread」と「uw_Isaboardregwrite」で使します。

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	SUW_ISAREGSET	
ポインタ型名	PSUW_ISAREGSET	
メンバ		
変数型	変数名	説明
DWORD	dwEnable	ボードの有無が設定されます。 ボード有 : DUW_BOARD_ENABLE (1) ボード無 : DUW_BOARD_DISABLED (0)
DWORD	dwBaseAddress	ボードのベースアドレスが設定されます。 0x0000 ~ 0xFFFF (16 進数) Windows95/98/2000 の場合は、デバイスのリソース (コントロールパネル・システム・デバイスマネジャー) で 設定された値と同じ必要があります。
DWORD	dwType	ボードの種類が設定されます。 AT-UW , AT-HUW : DUW_BOARD_ATHUW (0) AT-H500 , AT-H250 : DUW_BOARD_ATH500 (1) PCI-HUW : DUW_BOARD_PCIHUW (9)

・VisualBasic の場合

構造体名	SUW_ISAREGSET	
メンバ		
変数型	変数名	説明
Long	lEnable	ボードの有無が設定されます。 ボード有 : DUW_BOARD_ENABLE (1) ボード無 : DUW_BOARD_DISABLED (0)
Long	lBaseAddress	ボードのベースアドレスが設定されます。 &h0000 ~ &hFFFF (16 進数) Windows95/98/2000 の場合は、デバイスのリソース (コントロールパネル・システム・デバイスマネジャー) で 設定された値と同じ必要があります。
Long	lType	ボードの種類が設定されます。 AT-UW , AT-HUW : DUW_BOARD_ATHUW (0) AT-H500 , AT-H250 : DUW_BOARD_ATH500 (1) PCI-HUW : DUW_BOARD_PCIHUW (9)

6 . 注意事項

本インターフェースを使用する場合、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「UwiDll.dll」が、Windows または WINNT の、ディレクトリかアプリケーションと同じディレクトリに必要となります。
- ・ C 言語でご使用になる場合には、「UwiApi.h」と「UwiDll.lib」ファイルが必要となります。
「UwiApi.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ください。
「UwiDll.lib」ファイルは、使用するプロジェクトにリンクしてください。
- ・ 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「UwiApi.TXT」ファイルを API ビューアでご使用ください。

7 . サンプルプログラム

添付フロッピーディスクにサンプルプログラムがおさめられています。

ルートディレクトリにデバッグ用として使用可能な

UwiDebug.exe

ディレクトリ [sample] の [95vc++5] にソースプログラム

UwiDebug.cpp

ディレクトリ [sample] の [95vb5] に取扱説明書の VisuaiBasic 例をまとめた

VbSample.exe

ディレクトリ [sample] の [95vb5] に VisuaiBasic 例のソースプログラム

VbSample.vbp があります。

UwiDEbug.exe では、はじめに制御するボードをボード No で選択します。

入力が、[in] の欄に 16 進のワードデータとして表示され、出力は、16 進のワードデータを入力して、[out] をクリックすると出力されます。[Enter]キーは押さないでください。

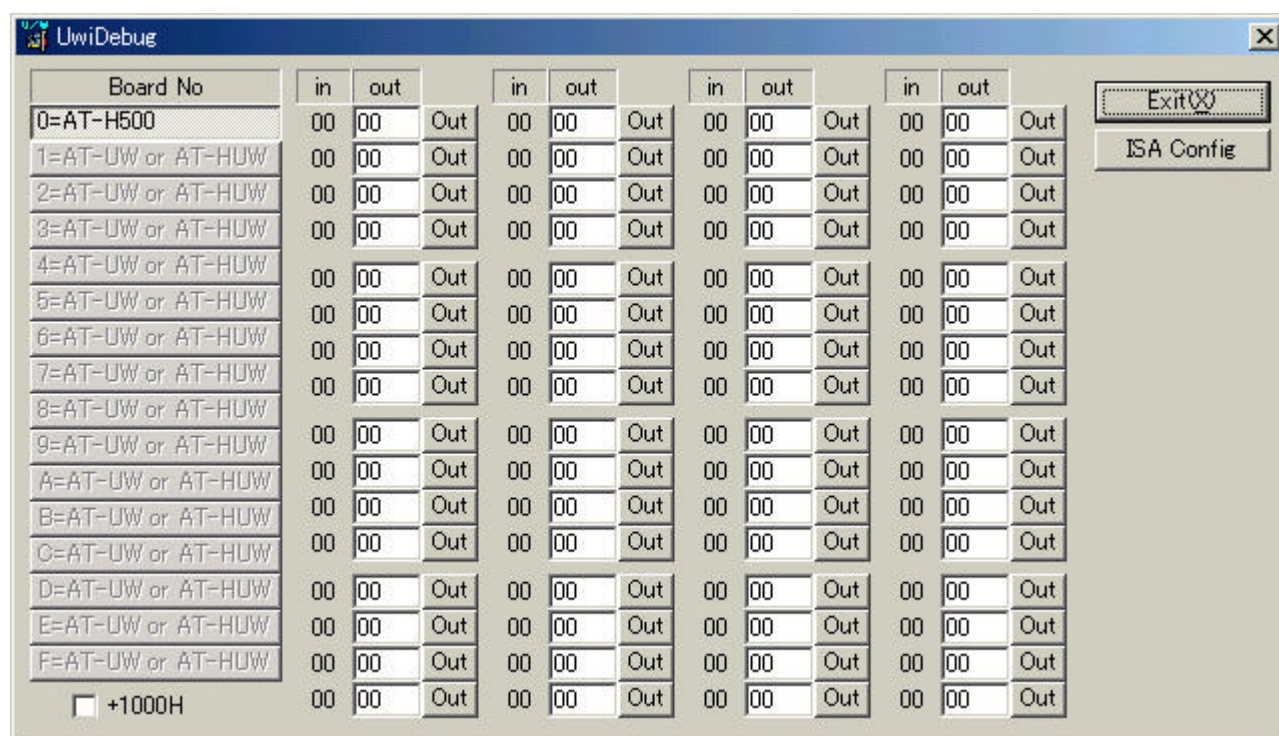
AT-H500 (AT-H250) の場合は、+ 1 0 0 0 H のボタンを押すと、表示が、全て + 1 0 0 0 H されたものになります。

[ISA Config] ボタンを押すと、新たに、設定用ウインドウが開きます。

ここで、ボード No を選び、ボードの種類と先頭アドレス (ボードのディップスイッチの値) を入れます。

PCIバスボードと No が重ならないようにしてください。

次に Windows を起動させると、登録されて使用できるようになります。



ディレクトリ [PCI-HUW] には、PCI-HUW を DOS 上で使用するためのリソースチェックプログラム [CHK_HUW] と、入出力テストサンプルプログラム [SMP_HUW] がおさめられています。

[CHK_HUW]は、PCI-HUW の枚数分 ボード No、先頭アドレス、リソース を表示します。

[SMP_HUW]は、選択したボード No のボードの I/O0 ~ 7 を入力、I/O64 ~ 71 に 55 を出力します。

8 . 変更履歴

取扱説明書

バージョン	日付	変更内容
EWINDRV-800C (V2.0)	2002.3.4	Windows2000 対応版 リリース
EWINDRV-800D (V2.1)	2004.6.29	名称変更

フロッピーディスク

バージョン	日付	変更内容
Windows 95/98/NT4.0/2000 (V2.0)	2002.3.4	Windows2000 対応版 リリース
Windows 95/98/NT4.0/2000 (V2.1)	2004.6.29	名称変更

N K E 株式会社

本 社 工 場 〒617-0828 京都府長岡京市馬場図所27	TEL 075-955-0071 (代) FAX 075-955-1063
東 京 営 業 所 〒110-0016 東京都台東区台東2丁目12-2(不二DICビル)	TEL 03-3833-5330 (代) FAX 03-3833-5350
名 古 屋 営 業 所 〒460-0026 名古屋市中区伊勢山2丁目13-22(ITOHビル)	TEL 052-322-3481 (代) FAX 052-322-3483
大 阪 営 業 所 〒550-0013 大阪市西区新町1丁目2-13(新町ビル)	TEL 06-6538-7136 (代) FAX 06-6538-7138
京 都 営 業 所 〒612-8487 京都市伏見区羽束師菱川町336-1	TEL 075-924-3293 (代) FAX 075-924-3290
伏 見 工 場 〒612-8487 京都市伏見区羽束師菱川町336-1	TEL 075-931-2731 (代) FAX 075-934-8746