

aPCI-013-050323

aPCI シリーズ絶縁型パラレル I/O ボード

aPCI-P34

ソフトウェアマニュアル

株式会社 **アドテックシステムサイエンス**

— 目次 —

1. はじめに.....	2
2. 動作環境.....	2
3. ファイル一覧.....	3
4. ドライバ.....	4
4-1. インストール.....	4
4-1-1. Windows2000 へのインストール.....	4
4-1-2. WindowsXP へのインストール.....	8
4-2. アンインストール.....	10
4-2-1. Windows2000 からのアンインストール.....	10
4-2-2. WindowsXP からのアンインストール.....	13
5. 動作チェックソフト.....	16
5-1. 概要.....	16
5-2. 起動画面.....	16
5-3. 「ファイル」メニュー.....	16
5-4. 「ヘルプ」メニュー.....	16
5-5. デバイス情報表示.....	17
5-6. IRQ の設定.....	17
5-7. 日時表示.....	17
5-8. ポート入出力.....	17
6. サンプルソース.....	18
7. API 仕様.....	19
7-1. 概要.....	19
7-2. プログラム構成.....	19
7-3. API リファレンス.....	20
7-3-1. Apci34Create.....	20
7-3-2. Apci34Close.....	22
7-3-3. Apci34GetVersion.....	23
7-3-4. Apci34GetResource.....	24
7-3-5. Apci34GetSwitchValue.....	27
7-3-6. Apci34InPort.....	29
7-3-7. Apci34OutPort.....	31
7-3-8. Apci34IrqMode.....	32
7-3-9. Apci34EnableTimer.....	34
7-3-10. Apci34DisableTimer.....	36
7-3-11. Apci34GetLastError.....	37
7-4. 定義.....	38
7-4-1. エラーコード.....	38
7-4-2. 論理ポート番号定数.....	38
7-4-3. 定数.....	38
7-5. メッセージ.....	38
製品のお問い合わせについて.....	39
改訂履歴.....	40

1. はじめに

本マニュアルは、入力 32 点／出力 32 点 絶縁型パラレル I/O ボード「aPCI-P34」を Windows2000／WindowsXP でご利用いただくためのドライバのインストール手順および添付ソフトウェアの使用方法などについて記述しています。

ハードウェアに関する詳細は、ハードウェアマニュアルを参照してください。

2. 動作環境

API は、Windows(R) 2000 および Windows(R) XP 上で、最大 16 枚までの aPCI-P34 (以下「デバイス」) を制御します。

3. ファイル一覧

¥		
├	Win2000_XP	Windows2000/XP 用 32bit ドライバ
├	├ apci34.inf	インストールファイル
├	├ apci34.sys	ドライバ
├	├ apci34.dll	ドライバ制御 DLL
├	Sample	サンプルソース
├	├ VC	Visual C++サンプルソース Ver.4/5/6 対応
├	├ ├ apci34.h	DLL 定義ヘッダ
├	├ ├ apci34w.h	ラッパー関数ヘッダ
├	├ ├ apci34w.c	ラッパー関数
├	├ ├ apci34s.c	ラッパー関数使用例
├	├ ├ buildvc.txt	サンプルソース構築例
├	├ ├ Lib	
├	├ ├ ├ Win2000_XP	
├	├ ├ ├ ├ apci34.lib	インポートライブラリ
├	├ VB	Visual Basic サンプルソース Ver.4(32bit)/5/6 対応
├	├ ├ apci34.bas	DLL 定義
├	├ ├ apci34w.bas	ラッパー関数
├	├ ├ apci34s.bas	ラッパー関数使用例
├	├ ├ buildvb.txt	サンプルソース構築例
├	├ Delphi	Delphi サンプルソース Ver.2/3/4/5/6/7 対応
├	├ ├ apci34w.pas	DLL 定義/ラッパー関数
├	├ ├ apci34s.pas	ラッパー関数使用例
├	├ ├ apci34s.dfm	フォームファイル
├	├ ├ builddp.txt	サンプルソース構築例
├	apci34.hlp/cnt	ヘルプ
├	apci34.exe	動作チェックアプリケーション
├	apci34ap.txt	API リファレンス
├	readme.txt	リードミー

4. ドライバ

4-1. インストール

本製品をご使用になる前に、ソフトウェアの組み込み等の準備が必要です。

ソフトウェアは、サポートソフト（添付サポートディスクまたは弊社ホームページ <http://www.adtek.co.jp/> からダウンロード）に収められています。

ここでは、サポートソフトを、フロッピーディスク（以下「サポートディスク」）にコピーして使用する場合について示しています。CD-R 等他のメディアをご使用の場合は、適宜読み替えて作業を進めてください。

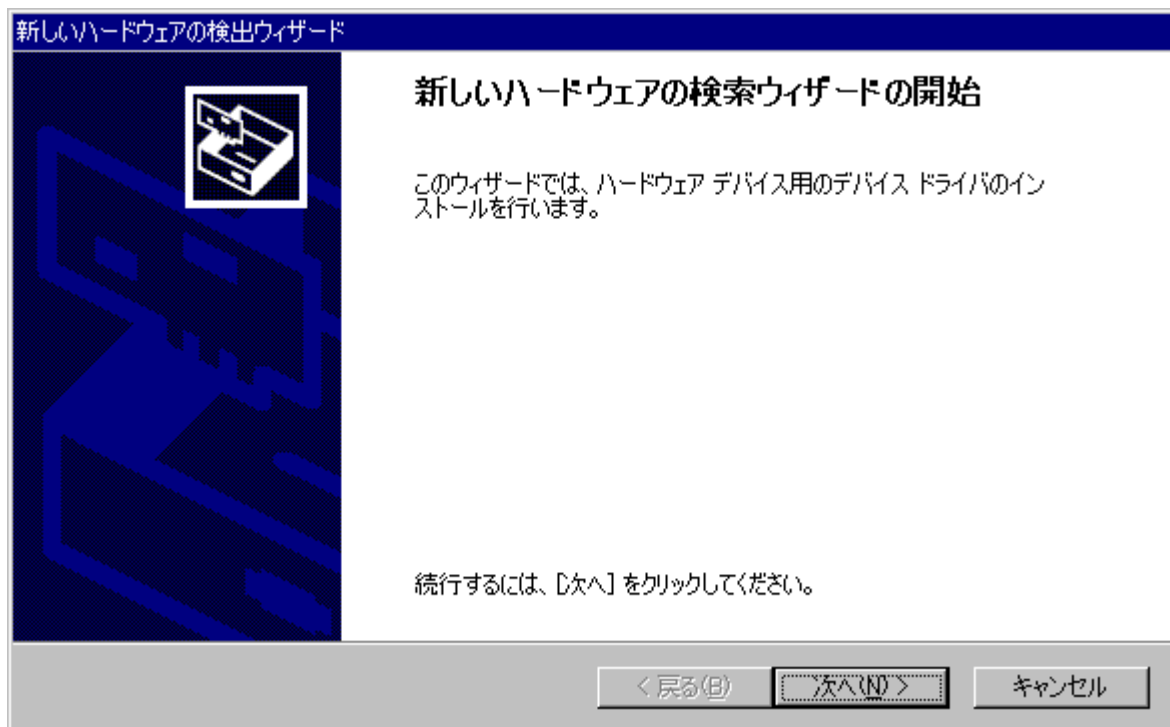
4-1-1. Windows2000 へのインストール

当ボードは、プラグアンドプレイに対応しておりますので、以下の手順に従って登録(インストール)を行って下さい。

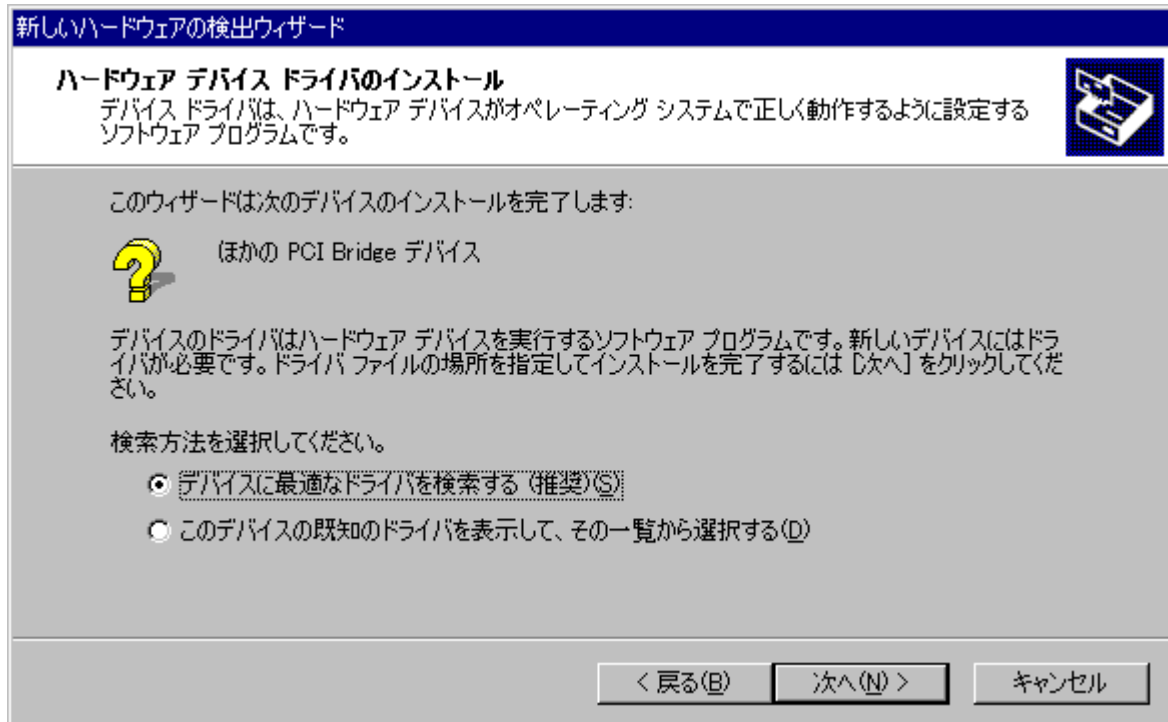
システムの電源が切れていることを確認し、ボードをスロットに挿入します。

システムの電源を入れ Windows2000 が起動しますと、[新しいハードウェアが見つかりました]とメッセージが出ます。その後"新しいハードウェアの検出ウィザード"が起動しますので、メッセージに従ってインストールを行います。

以下の画面が現れましたら、フロッピーディスクをドライブに挿入し、[次へ]ボタンをクリックします。



[デバイスに最適なドライバを検索する(推奨)]を選択し、[次へ]ボタンをクリックします。

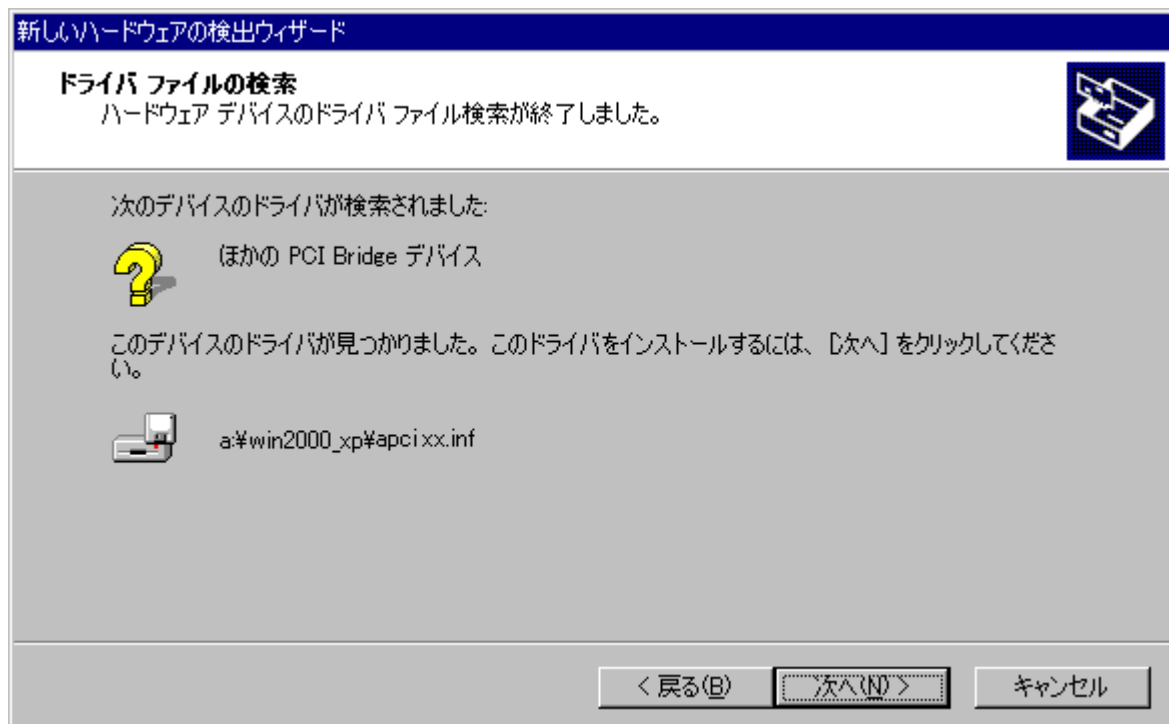


※ 画面中の「ほかの PCI Bridge デバイス」は「マルチメディア コントローラ」と表示される場合もあります。(以下の画面でも同様)

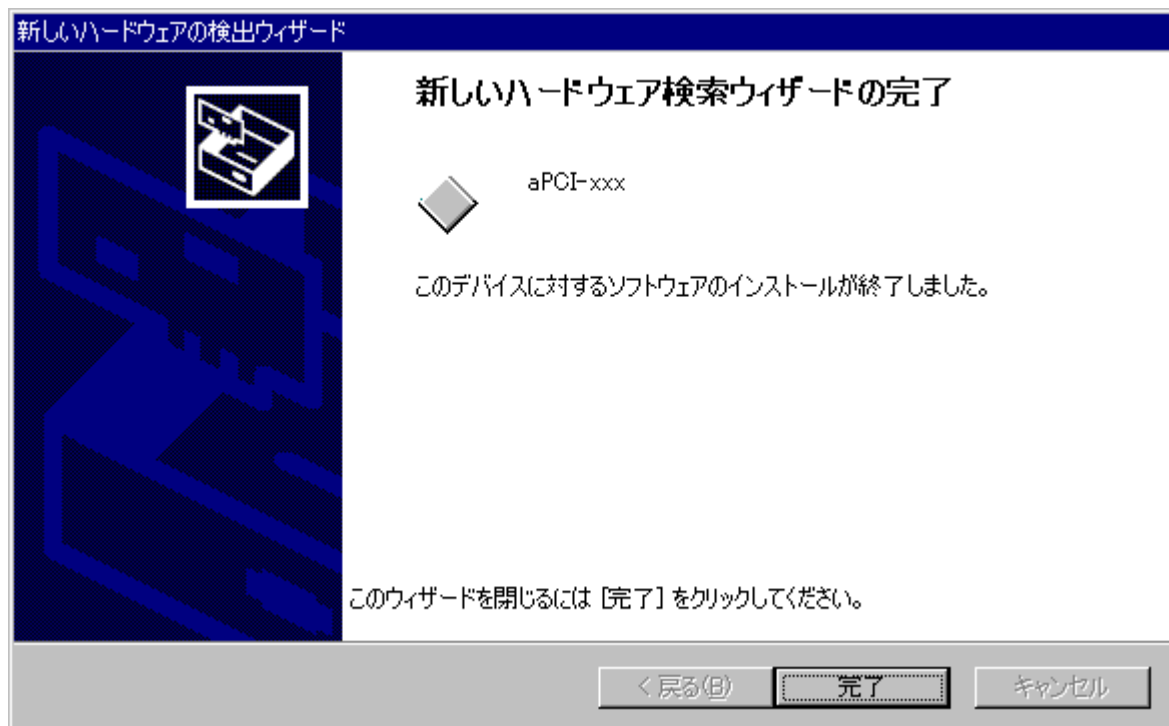
[フロッピーディスクドライブ]を選択し、[次へ]ボタンをクリックします。



[次へ]ボタンをクリックします。



[完了]ボタンをクリックしてインストールは終了です。



ドライバを制御する DLL ファイルは自動的にコピーされませんので、お使いの環境に合わせて、ファイルをコピーしてください。

一般的には「アプリケーションと同じディレクトリ」か、Windows2000 では「%WinDir%\System32 (例 C:\Winnt\System32)」となります。

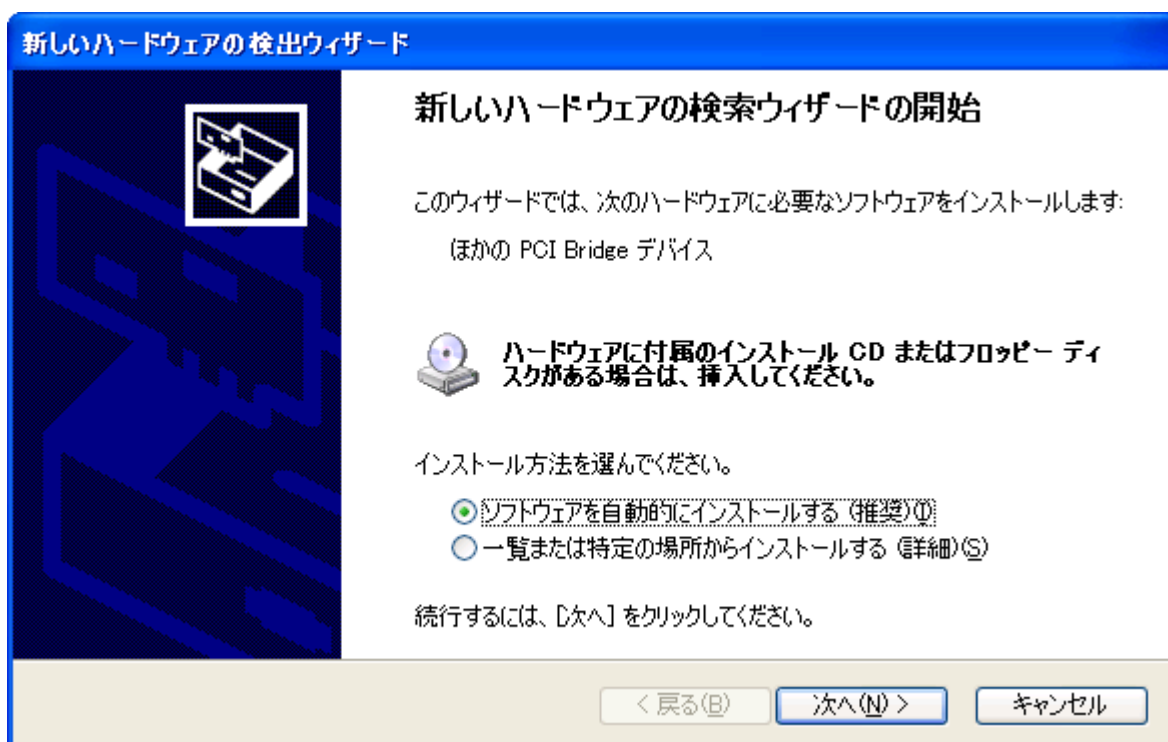
4-1-2. WindowsXP へのインストール

当ボードは、プラグアンドプレイに対応しておりますので、以下の手順に従って登録(インストール)を行って下さい。

システムの電源が切れていることを確認し、ボードをスロットに挿入します。

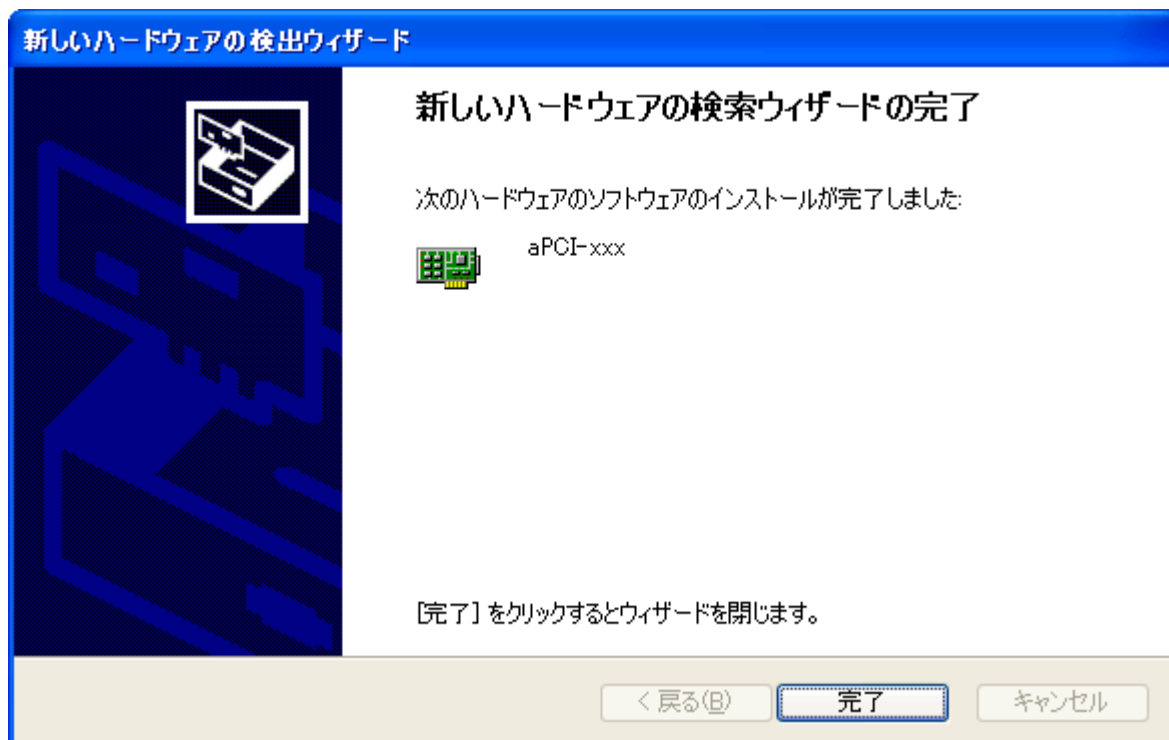
システムの電源を入れ WindowsXP が起動しますと、[新しいハードウェアが見つかりました]とメッセージが出ます。その後"新しいハードウェアの検出ウィザード"が起動しますので、メッセージに従ってインストールを行います。

以下の画面が現れましたら、フロッピーディスクをドライブに挿入し、[次へ]ボタンをクリックします。



※ 画面中の「ほかの PCI Bridge デバイス」は「マルチメディア コントローラ」と表示される場合もあります。

[完了]ボタンをクリックしてインストールは終了です。



ドライバを制御する DLL ファイルは自動的にコピーされませんので、お使いの環境に合わせて、ファイルをコピーしてください。
一般的には「アプリケーションと同じディレクトリ」か、WindowsXP では「%WinDir%\System32 (例 C:\Windows\System32)」となります。

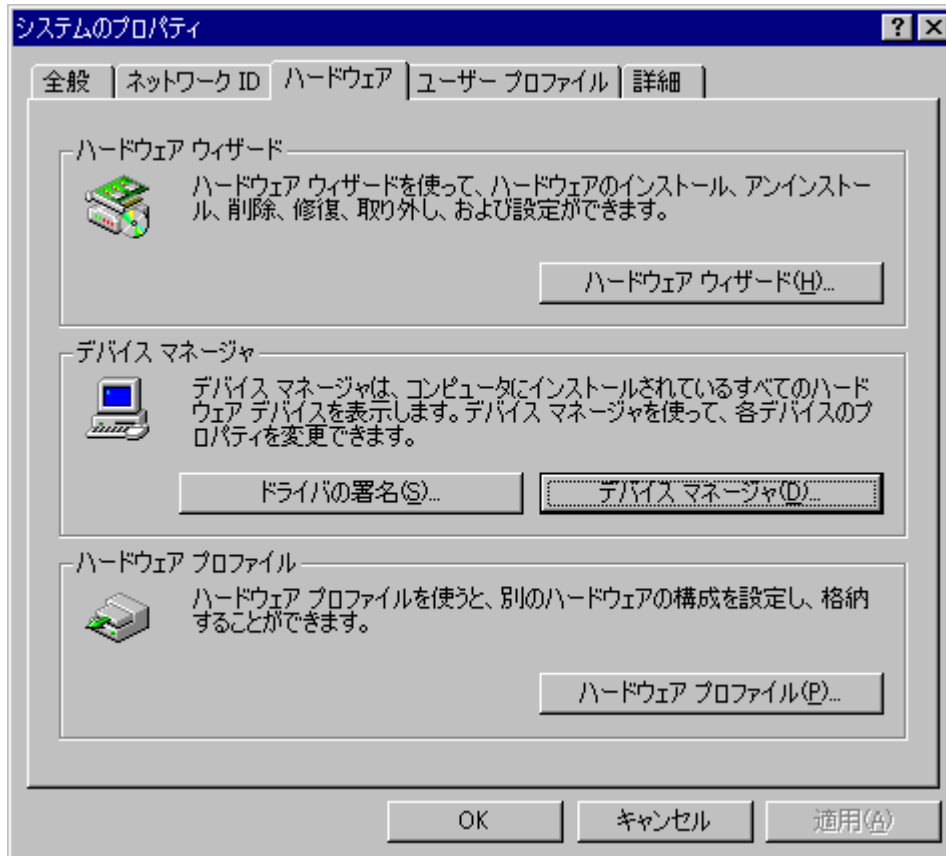
4-2. アンインストール

4-2-1. Windows2000 からのアンインストール

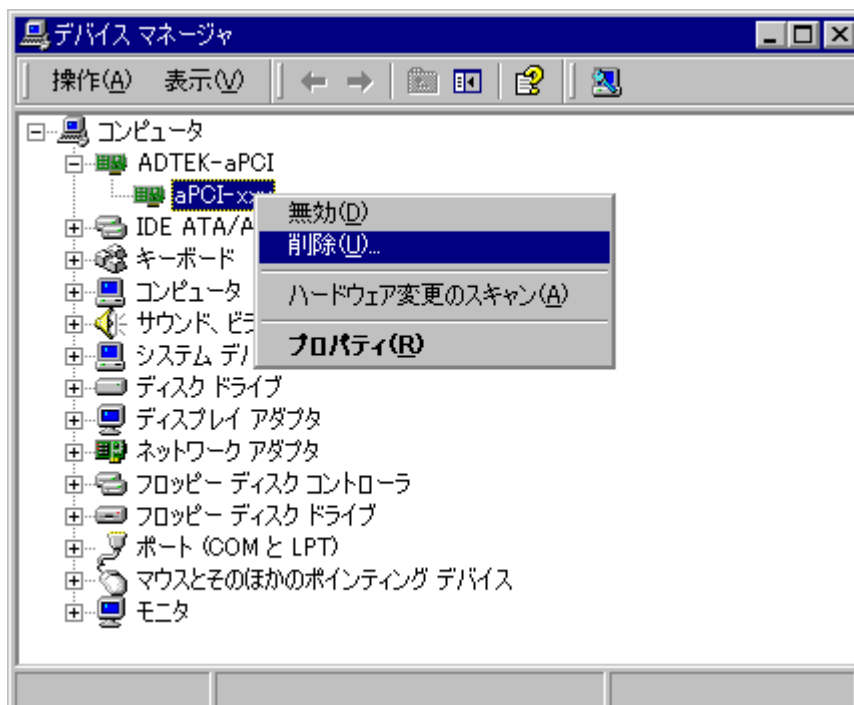
[スタート-設定-コントロールパネル]を実行して、コントロールパネルを開きます。開いたら[システム]アイコンをダブルクリックします。



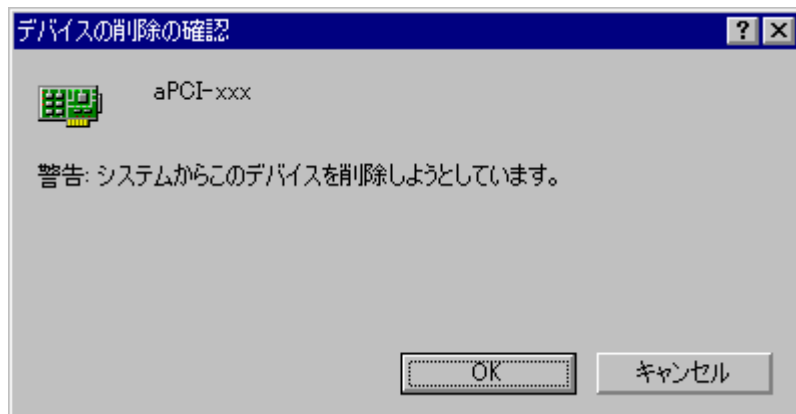
下図の欄で[デバイスマネージャ]ボタンをクリックします。



[aPCI-P34]上で右クリックして、[削除]を選択します。



[OK]ボタンをクリックします。



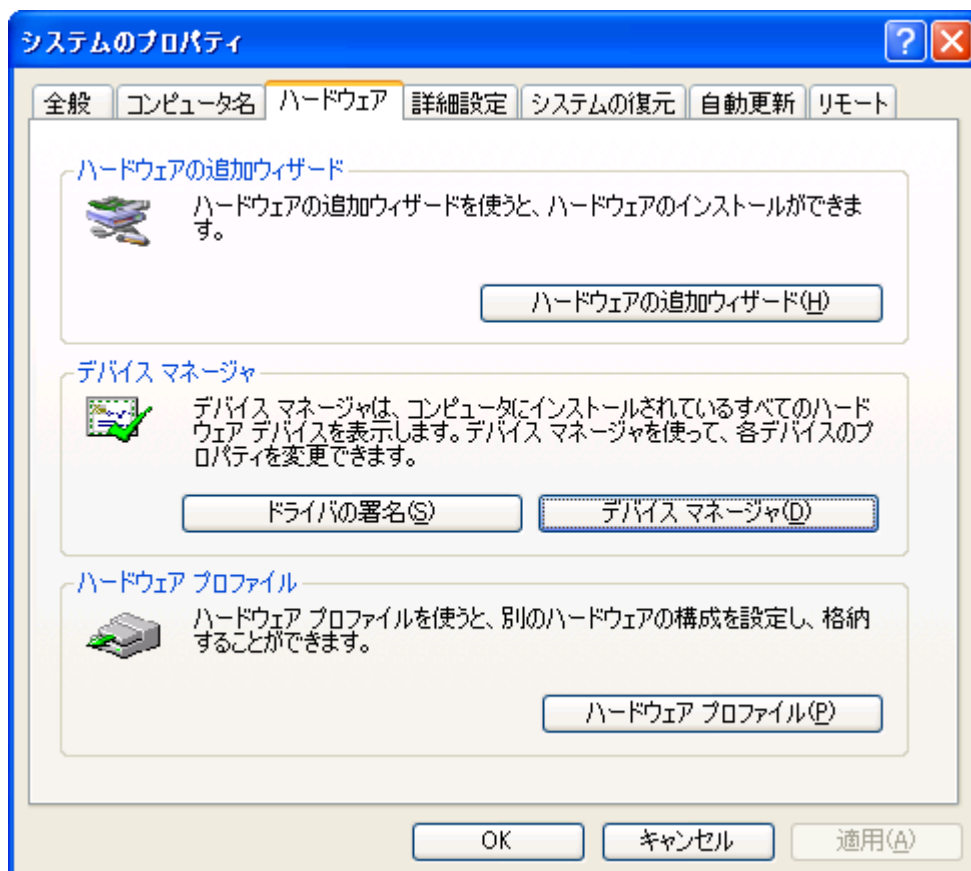
Windows2000 を終了してからパソコンの電源を切りボードを取り外します。

4-2-2. WindowsXP からのアンインストール

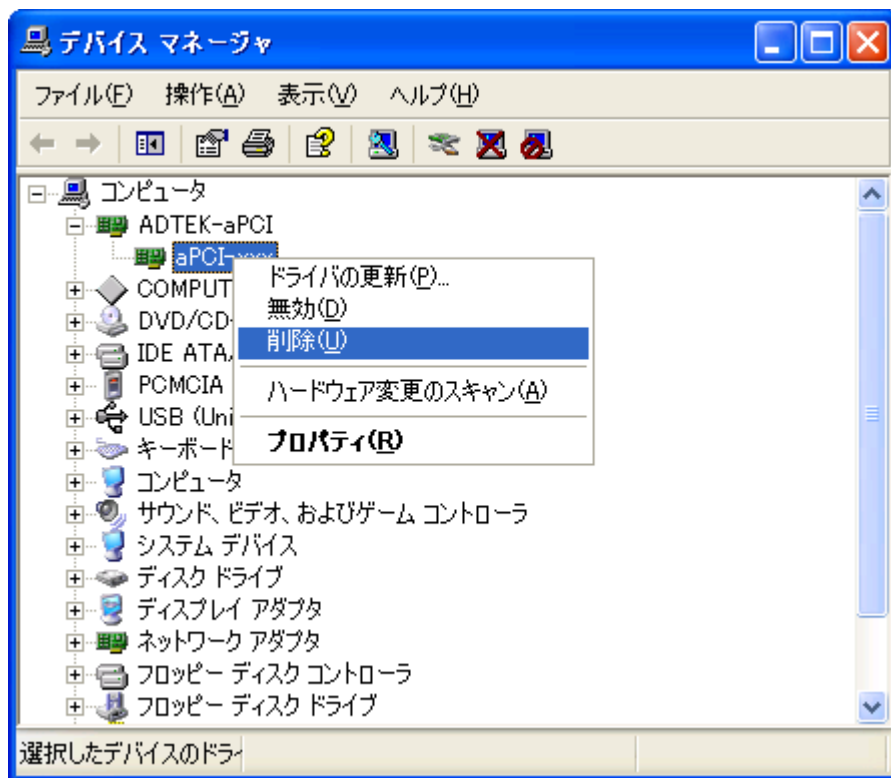
[スタート-コントロールパネル]を実行して、コントロールパネルを開きます。開いたら[システム]アイコンをダブルクリックします。



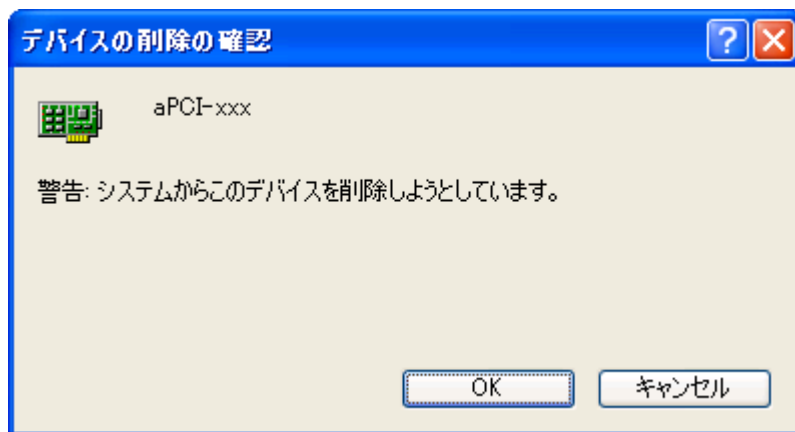
下図の欄で[デバイスマネージャ]ボタンをクリックします。



[aPCI-P34]上で右クリックして、[削除]を選択します。



[OK]ボタンをクリックします。



WindowsXP を終了してからパソコンの電源を切りボードを取り外します。

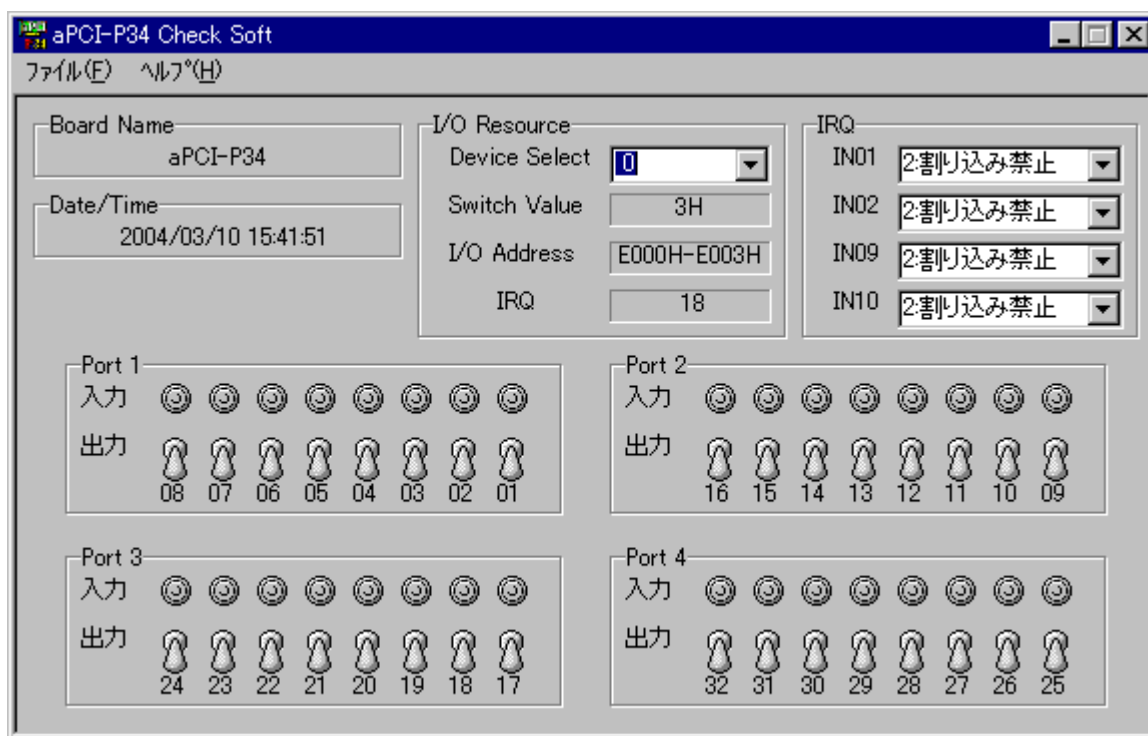
5. 動作チェックソフト

5-1. 概要

本アプリケーションソフト(apci34.exe)は、各機能の動作チェックを行うツールです。

5-2. 起動画面

本アプリケーションを起動すると、下記の様な画面を表示します。



5-3. 「ファイル」メニュー

[プログラムの終了] を選択することにより、本アプリケーションを終了します。

5-4. 「ヘルプ」メニュー

[トピックの検索] を選択することにより、ヘルプファイル(apci34.hlp)を表示します。

[このプログラムについて] を選択することにより、ドライバおよび DLL のバージョン情報を表示します。

5-5. デバイス情報表示

[Device Select]

複数のボードが同一のシステムにインストールされている場合、どのボードに対して制御を行うか選択します。

[I/O Address]

このデバイスが使用している I/O アドレスが表示されます。

[BSN]

BSN(Board Select Number)セレクトロータリースイッチで設定した値が表示されます。

[IRQ]

このデバイスが使用している IRQ が表示されます。

5-6. IRQ の設定

IN01/IN02/IN09/IN10 の割り込みを設定します。

割り込み要因となる信号が Low から High または High から Low に変化(割り込みが発生)した場合に、メッセージボックスが表示されます。

5-7. 日時表示

デバイスのタイマー割り込みを使用し、現在の日時が表示されます。

5-8. ポート入出力

「入力」の場合、ビットの H/L の状態が表示に反映されます。

「出力」の場合、クリックすると H-L が反転します。

6. サンプルソース

サンプルソースには、Visual C++ (Ver.4/5/6) 版、Visual Basic (Ver.4(32bit)/5/6) 版、Delphi (Ver.2/3/4/5/6/7) 版がございます。

DLL 内の関数のインターフェースは、各関数のヘッダ、注釈および「7. API 仕様」を参照してください。

<ラッパー関数>

apci34w.* のように「w」が付いているファイルには、DLL 内の関数を簡単にコールするためのラッパー (Wrapper) 関数が定義されています (DLL のロード/アンロード関数も含まれています)。デバイスの制御を開始する前には DLL ロード関数を、デバイスの制御を終了する際には DLL アンロード関数をコールしてください。

サンプルソースは各開発環境にて実行してお試しいただけます。
詳しくは各ディレクトリ内の buildxx.txt の例をご覧ください。

- ※ サンプルソースのご利用については、各開発環境および OS・言語に対する十分な理解を前提としております。よって、これらそのものの使用方法に関するお問い合わせには一切お答えいたしかねますので、あらかじめご了承ください。

7. API仕様

7-1. 概要

本章では、aPCI-P34 用ユーザ公開 API を定義します。
デバイスドライバの詳細については触れていません。

7-2. プログラム構成

本章で定義される API は、`apci34.sys` (デバイスドライバ) および `apci34.dll` (ダイナミックリンクライブラリ) により実現されます。

7-3. API リファレンス

7-3-1. Apci34Create

■機能 デバイスの使用を宣言

■形式 Visual C++

```
BOOL Apci34Create  
(  
    LPWORD pwLogSlot,  
    HWND hwnd,  
    UINT uWMBase,  
    LPVOID lpvInitArg  
);
```

■形式 Visual Basic

```
Function Apci34Create  
(  
    pwLogSlot As Integer,  
    ByVal hwnd As Long,  
    ByVal uWMBase As Long,  
    LpvInitArg As Any  
) As Long
```

■形式 Delphi

```
function Apci34Create  
(  
    var pwLogSlot: WORD;  
    hwnd: THandle;  
    uWMBase: UINT;  
    var lpvInitArg  
): BOOL;
```

■入力 pwLogSlot

使用したいデバイスが挿入されている論理スロットを指定します。

論理スロット番号は 0 から始まります。

APCI34_SLOT_AUTO を指定した場合、使用可能な論理スロットを探します。

論理スロット 0 から検索し、見つかった時点でその論理スロット番号を返します。

すでにアプリケーションにより使用されているデバイスはスキップします。

以後、アプリケーションはこの値でデバイスを識別します。

NULL 不可。

hwnd

ドライバが送出するメッセージを受け取るウィンドウのハンドルを指定します。

メッセージを使用しない場合は NULL を指定します。

uWMBase

送出するメッセージ ID の値を指定します。
通常 WM_USER 以上の値を指定します。

lpvInitArg

使用しません (将来の為の予約です)。
Visual Basic の場合はダミーの変数を渡してください。

■出力 *pwLogSlot

使用可能なデバイスが見つかった場合は、その論理スロット番号を格納します。
見つからなかった場合、この値は未定義です。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。
TRUE 正常終了。

■解説 論理スロットに存在しているであろうデバイスを、アプリケーションが使用することをドライバに通知します。論理スロットにデバイスが存在していない場合は、エラーとなります。論理スロット内のデバイスがすでに他のアプリケーションで使用されている場合もエラーとなります。これにより、1 つのデバイスは単一のアプリケーションから排他的に使用されます。他の API を呼び出す前に必ずこの API を呼び出してください。

■エラー APCI34_ERR_SYSTEM

Windows の GetLastError() をコールしてください。

APCI34_ERR_NO_DEVICE

使用可能なデバイスがありません (APCI34_SLOT_AUTO を指定した場合)。

APCI34_ERR_IN_USE

指定のデバイスは使用中です。

APCI34_ERR_INVALID_ARGUMENT

pwLogSlot が NULL です。

この API が失敗した場合、Apci34GetLastError() の wLogSlot には APCI34_SLOT_AUTO を指定してください。

7-3-2. Apci34Close

■機能 デバイスの解放

■形式 Visual C++

```
BOOL Apci34Close  
(  
    WORD wLogSlot  
);
```

■形式 Visual Basic

```
Function Apci34Close  
(  
    ByVal wLogSlot As Integer  
) As Long
```

■形式 Delphi

```
function Apci34Close  
(  
    wLogSlot: WORD  
): BOOL;
```

■入力 wLogSlot

このデバイスを解放します。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解説 アプリケーションがデバイスの使用を終了し、デバイスを他のアプリケーションに解放することをドライバに通知します。アプリケーションを終了する前に必ずこの API を呼び出してください。

■エラー APCI34_ERR_SYSTEM

Windows の GetLastError() をコールしてください。

APCI34_ERR_INVALID_SLOT

無効な論理スロット番号です。

APCI34_ERR_NO_CREATE

デバイスがクリエイトされていません。

7-3-3. Apci34GetVersion

■機能 バージョン情報取得

■形式 Visual C++

```
BOOL Apci34GetVersion  
(  
    LPDWORD pdwDllVer,  
    LPDWORD pdwDrvVer  
);
```

■形式 Visual Basic

```
Function Apci34GetVersion  
(  
    pdwDllVer As Long,  
    pdwDrvVer As Long  
) As Long
```

■形式 Delphi

```
function Apci34GetVersion  
(  
    var pdwDllVer: DWORD;  
    var pdwDrvVer: DWORD  
): BOOL;
```

■入力 pdwDllVer

DLL のバージョン情報を格納する領域へのポインタ。
NULL 可。

pdwDrvVer

DRIVER のバージョン情報を格納する領域へのポインタ。
NULL 可。

■出力 *pdwDllVer

DLL のバージョン情報。

*pdwDrvVer

DRIVER のバージョン情報。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解説 DLL と DRIVER のバージョン情報を取得します。

それぞれのバージョンは 4 桁で表現されます。

初回リリースは 0.1.0.0 とし、0x00010000 となります。

最大値は 255.255.255.255 となり、0xffffffff となります。

7-3-4. Apci34GetResource

■機能 リソース情報の取得

■形式 Visual C++

```
#define APCI34_MAX_MEM 9
#define APCI34_MAX_IO 20
#define APCI34_MAX_IRQ 7
#define APCI34_MAX_DMA 7

typedef struct _APCI34RESOURCE
{
    DWORD dwNumMemWindows; /* Not used */
    DWORD dwMemBase [APCI34_MAX_MEM]; /* Not used */
    DWORD dwMemLength [APCI34_MAX_MEM]; /* Not used */
    DWORD dwMemAttrib [APCI34_MAX_MEM]; /* Not used */
    DWORD dwNumIOPorts; /* Num IO ports */
    DWORD dwIOPortBase [APCI34_MAX_IO]; /* I/O port base */
    DWORD dwIOPortLength [APCI34_MAX_IO]; /* I/O port length */
    DWORD dwNumIRQs; /* Num IRQ info */
    DWORD dwIRQRegisters [APCI34_MAX_IRQ]; /* IRQ list */
    DWORD dwIRQAttrib [APCI34_MAX_IRQ]; /* IRQ Attrib list */
    DWORD dwNumDMAs; /* Not used */
    DWORD dwDMALst [APCI34_MAX_DMA]; /* Not used */
    DWORD dwDMAAttrib [APCI34_MAX_DMA]; /* Not used */
    DWORD dwReserved1 [3]; /* Not used */
} APCI34RESOURCE;
typedef APCI34RESOURCE * PAPCI34R;

BOOL Apci34GetResource
(
    WORD wLogSlot,
    PAPCI34R pres
);
```

■形式 Visual Basic

```
Global Const APCI34_MAX_MEM = 9
Global Const APCI34_MAX_IO = 20
Global Const APCI34_MAX_IRQ = 7
Global Const APCI34_MAX_DMA = 7
```

```
Type APCI34RESOURCE
    dwNumMemWindows As Long ' Not used
    dwMemBase(1 To APCI34_MAX_MEM) As Long ' Not used
    dwMemLength(1 To APCI34_MAX_MEM) As Long ' Not used
    dwMemAttrib(1 To APCI34_MAX_MEM) As Long ' Not used
```

```

dwNumIOPorts As Long           ' Num IO ports
dwIOPortBase(1 To APCI34_MAX_IO) As Long ' I/O port base
dwIOPortLength(1 To APCI34_MAX_IO) As Long ' I/O port length
dwNumIRQs As Long              ' Num IRQ info
dwIRQRegisters(1 To APCI34_MAX_IRQ) As Long ' IRQ list
dwIRQAttrib(1 To APCI34_MAX_IRQ) As Long   ' IRQ Attrib list
dwNumDMAs As Long              ' Not used
dwDMALst(1 To APCI34_MAX_DMA) As Long     ' Not used
dwDMAAttrib(1 To APCI34_MAX_DMA) As Long   ' Not used
dwReserved1(1 To 3) As Long             ' Not used
End Type

```

```

Function Apci34GetResource
(
  ByVal wLogSlot As Integer,
  pres As APCI34RESOURCE
) As Long

```

■形式 Delphi

```

const
  APCI34_MAX_MEM = 9;
  APCI34_MAX_IO = 20;
  APCI34_MAX_IRQ = 7;
  APCI34_MAX_DMA = 7;

type
  TAPCI34RESOURCE = record
    dwNumMemWindows: DWORD;           { Not used }
    dwMemBase:      array [1 .. APCI34_MAX_MEM] of DWORD; { Not used }
    dwMemLength:   array [1 .. APCI34_MAX_MEM] of DWORD; { Not used }
    dwMemAttrib:   array [1 .. APCI34_MAX_MEM] of DWORD; { Not used }
    dwNumIOPorts:  DWORD;              { Num IO ports }
    dwIOPortBase:  array [1 .. APCI34_MAX_IO] of DWORD;  { I/O port base }
    dwIOPortLength: array [1 .. APCI34_MAX_IO] of DWORD; { I/O port length }
    dwNumIRQs:     DWORD;              { Num IRQ info }
    dwIRQRegisters: array [1 .. APCI34_MAX_IRQ] of DWORD; { IRQ list }
    dwIRQAttrib:   array [1 .. APCI34_MAX_IRQ] of DWORD; { IRQ Attrib list }
    dwNumDMAs:     DWORD;              { Not used }
    dwDMALst:      array [1 .. APCI34_MAX_DMA] of DWORD; { Not used }
    dwDMAAttrib:   array [1 .. APCI34_MAX_DMA] of DWORD; { Not used }
    dwReserved1:   array [1 .. 3] of DWORD;             { Not used }
  end;
  PAPCI34RESOURCE = ^TAPCI34RESOURCE;

```

```
function Apci34GetResource  
(  
    wLogSlot: WORD;  
    pres: PAPCI34RESOURCE  
): BOOL;
```

■入 力 wLogSlot

このデバイスのリソース情報を取得します。

pres

リソース情報を格納する領域へのポインタ。

NULL 不可。

■出 力 *pres

リソース情報。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解 説 wLogSlot で指定されたデバイスに割り当てられているリソースを表示用に取得します。

■エラー APCI34_ERR_INVALID_SLOT

無効な論理スロット番号です。

APCI34_ERR_NO_CREATE

デバイスがクリエイトされていません。

APCI34_ERR_INVALID_ARGUMENT

pres が NULL です。

APCI34_ERR_SYSTEM

Windows の GetLastError() をコールしてください。

7-3-5. Apci34GetSwitchValue

■機能 ロータリスイッチの値を取得

■形式 Visual C++

```
BOOL Apci34GetSwitchValue  
(  
    WORD wLogSlot,  
    LPDWORD pdwSwitchValue  
);
```

■形式 Visual Basic

```
Function Apci34GetSwitchValue  
(  
    ByVal wLogSlot As Integer,  
    pdwSwitchValue As Long  
) As Long
```

■形式 Delphi

```
function Apci34GetSwitchValue  
(  
    wLogSlot: WORD;  
    var pdwSwitchValue: DWORD  
): BOOL;
```

■入力 wLogSlot

デバイスの論理スロット番号を指定。

pdwSwitchValue

ロータリスイッチの値を格納する領域へのポインタ。
NULL は不可。

■出力 *pdwSwitchValue

ロータリスイッチの値。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解説 wLogSlot で指定されたデバイスに割り当てられているロータリスイッチの値を取得します。

- エラー APCI34_ERR_INVALID_SLOT
無効な論理スロット番号です。
- APCI34_ERR_NO_CREATE
デバイスがクリエイトされていません。
- APCI34_ERR_INVALID_ARGUMENT
pdwSwitchValue が NULL です。
- APCI34_ERR_SYSTEM
Windows の GetLastError() をコールしてください。

7-3-6. Apci34InPort

■機能 入力実行

■形式 Visual C++

```
BOOL Apci34InPort  
(  
    WORD wLogSlot,  
    DWORD dwLogPort,  
    PBYTE pbInValue  
);
```

■形式 Visual Basic

```
Function Apci34InPort  
(  
    ByVal wLogSlot As Integer,  
    ByVal dwLogPort As Long,  
    pbInValue As Byte  
    ) As Long
```

■形式 Delphi

```
function Apci34InPort  
(  
    wLogSlot: WORD;  
    wLogPort: DWORD;  
    var pbInValue: BYTE  
    ): BOOL;
```

■入力 wLogSlot

このデバイスのモードを設定します。

dwLogPort

この論理ポート番号の入力を実行します。

pbInValue

入力データを格納する領域へのポインタ。
NULL は不可。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解説 wLogSlot で指定されたデバイスの論理ポート番号 dwLogPort から入力します。

- エラー APCI34_ERR_INVALID_SLOT
無効な論理スロット番号です。
- APCI34_ERR_NO_CREATE
デバイスがクリエイトされていません。
- APCI34_ERR_INVALID_PORT
無効な論理ポート番号です。
- APCI34_ERR_INVALID_ARGUMENT
pbInValue が NULL です。

7-3-7. Apci34OutPort

■機能 出力実行。

■形式 Visual C++

```
BOOL Apci34OutPort
(
    WORD wLogSlot,
    DWORD dwLogPort,
    BYTE bOutValue
);
```

■形式 Visual Basic

```
Function Apci34OutPort
(
    ByVal wLogSlot As Integer
    ByVal dwLogPort As Long,
    ByVal bOutValue As Byte
) As Long
```

■形式 Delphi

```
function Apci34OutPort
(
    wLogSlot: WORD;
    dwLogPort: DWORD;
    bOutValue: BYTE
): BOOL;
```

■入力 wLogSlot

デバイスの論理スロット番号を指定。

dwLogPort

この論理ポート番号への出力を実行します。

bOutValue

出力データ。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解説 wLogSlot で指定されたデバイスの論理ポート番号 dwLogPort に出力します。

■エラー APCI34_ERR_INVALID_SLOT

無効な論理スロット番号です。

APCI34_ERR_NO_CREATE

デバイスがクリエイトされていません。

APCI34_ERR_INVALID_PORT

無効な論理ポート番号です。

7-3-8. Apci34IrqMode

■機能 IN01/IN02/IN09/IN10 の割り込み設定。

■形式 Visual C++

```
typedef struct _APCI34MODE
{
    DWORD    dwIn01; // IN01 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
    DWORD    dwIn02; // IN02 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
    DWORD    dwIn09; // IN09 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
    DWORD    dwIn10; // IN10 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
} APCI34MODE, *PCAPCI34M;

BOOL Apci34IrqMode
(
    WORD wLogSlot,
    PCAPCI34M pcmode
);
```

■形式 Visual Basic

```
Type APCI34MODE
    dwIn01 As Long ' IN01 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
    dwIn02 As Long ' IN02 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
    dwIn09 As Long ' IN09 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
    dwIn10 As Long ' IN10 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止)
End Type

Function Apci34IrqMode
(
    ByVal wLogSlot As Integer,
    ByRef pcmode As APCI34MODE
) As Long
```

■形式 Delphi

```
type
    TAPCI34MODE = record
        dwIn01: DWORD; { IN01 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止) }
        dwIn02: DWORD; { IN02 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止) }
        dwIn09: DWORD; { IN09 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止) }
        dwIn10: DWORD; { IN10 の設定 (0:Active Low, 1:Active High, 2:割り込み禁止) }
    end;
    PAPCI34MODE = ^TAPCI34MODE;

function Apci34IrqMode
(
    wLogSlot: WORD;
    pcmode: PAPCI34MODE
): BOOL;
```

■入 力 wLogSlot

デバイスの論理スロット番号を指定。

pcmode

割り込みデータ。

0: Active Low

割り込み要因となる信号が High から Low に変化した場合に、割り込みが発生します。

1: Active High

割り込み要因となる信号が Low から High に変化した場合に、割り込みが発生します。

2: 割り込み禁止

割り込み要因となる信号が Low から High または High から Low に変化しても、割り込みは発生しません。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解 説 wLogSlot で指定されたデバイスの IN01/IN02/IN09/IN10 の割り込み(割り込みの入力極性[Active Low/Active High] および 割り込み禁止)を設定します。

割り込みタイプは、エッジ入力のみ対応しております。

■エラー APCI34_ERR_INVALID_SLOT

無効な論理スロット番号です。

APCI34_ERR_NO_CREATE

デバイスがクリエイトされていません。

APCI34_ERR_INVALID_ARGUMENT

無効な引数を要求しました。

7-3-9. Apci34EnableTimer

■機能 タイマー割り込みを有効にする。

■形式 Visual C++

```
BOOL Apci34EnableTimer  
(  
    WORD wLogSlot,  
    DWORD dwTimerCount,  
    DWORD dwPlescaler  
);
```

■形式 Visual Basic

```
Function Apci34EnableTimer  
(  
    ByVal wLogSlot As Integer,  
    ByVal dwTimerCount As Long,  
    ByVal dwPlescaler As Long  
) As Long
```

■形式 Delphi

```
function Apci34EnableTimer  
(  
    wLogSlot: WORD;  
    dwTimerCount: DWORD;  
    dwPlescaler: DWORD  
): BOOL;
```

■入力 wLogSlot

デバイスの論理スロット番号を指定。

dwTimerCount

タイマーのカウンタを指定(24bit データ)。

dwPlescaler

タイマーのクロック周波数を指定(0 ~ 7)。

0: 30nsec

1: 60nsec

2: 120nsec

3: 240nsec

4: 480nsec

5: 960nsec

6: 1920nsec

7: 3840nsec

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解説 wLogSlot で指定されたデバイスのタイマー割り込みを有効にします。

- エラー APCI34_ERR_INVALID_SLOT
無効な論理スロット番号です。
- APCI34_ERR_NO_CREATE
デバイスがクリエイトされていません。
- APCI34_ERR_INVALID_ARGUMENT
無効な引数を要求しました。

7-3-10. Apci34DisableTimer

■機能 タイマー割り込みを無効にする。

■形式 Visual C++

```
BOOL Apci34DisableTimer  
(  
    WORD wLogSlot  
);
```

■形式 Visual Basic

```
Function Apci34DisableTimer  
(  
    ByVal wLogSlot As Integer  
) As Long
```

■形式 Delphi

```
function Apci34DisableTimer  
(  
    wLogSlot: WORD  
): BOOL;
```

■入力 wLogSlot

デバイスの論理スロット番号を指定。

■戻り値 API が正常終了したか、失敗したかを返します。

FALSE 失敗。

TRUE 正常終了。

■解説 wLogSlot で指定されたデバイスのタイマー割り込みを無効にします。

■エラー APCI34_ERR_INVALID_SLOT

無効な論理スロット番号です。

APCI34_ERR_NO_CREATE

デバイスがクリエイトされていません。

7-3-11. Apci34GetLastError

■機能 エラーコード取得

■形式 Visual C++

```
DWORD Apci34GetLastError  
(  
    WORD wLogSlot  
);
```

■形式 Visual Basic

```
Function Apci34GetLastError  
(  
    ByVal wLogSlot As Integer  
) As Long
```

■形式 Delphi

```
function Apci34GetLastError  
(  
    wLogSlot: WORD  
): DWORD;
```

■入力 wLogSlot

このデバイスのもっとも最近起こったエラーのコードを取得します。
デバイスに依存しないエラーは wLogSlot の番号に関わらず取得されます。

7-4. 定義

7-4-1. エラーコード

```
#define APCI34_SUCCESS          0 // 正常終了
#define APCI34_ERR_SYSTEM      1 // GetLastError を実行してください
#define APCI34_ERR_NO_DEVICE   2 // 使用可能なデバイスがない
#define APCI34_ERR_IN_USE      3 // 指定されたデバイスは使用中です
#define APCI34_ERR_INVALID_SLOT 4 // 無効な論理スロット番号を指定しました
#define APCI34_ERR_RESOURCE    5 // リソースエラー
#define APCI34_ERR_INVALID_PORT 6 // 無効なポートを要求しました
#define APCI34_ERR_INVALID_ARGUMENT 7 // 無効な引数を要求しました
#define APCI34_ERR_NO_CREATE   8 // デバイスがクリエイトされていません
```

APCI34_SUCCESS は、初期時に初期化される値です。

以降エラーが起こった場合はエラー内容が更新されるまでそのエラー値を保持します。

7-4-2. 論理ポート番号定数

```
#define APCI34_PORT_1          0
#define APCI34_PORT_2          1
#define APCI34_PORT_3          2
#define APCI34_PORT_4          3
```

7-4-3. 定数

```
#define APCI34_MAX_SLOTS      16 // サポートするボード枚数
#define APCI34_MAX_PORTS     4 // アクセス可能なポート数
#define APCI34_SLOT_AUTO     ((WORD)~0U) // デフォルト サーチ
```

7-5. メッセージ

wParam には論理スロット番号が入ります。

lParam にはデバイス固有の ID が入ります。

WM_APCI34_IN01 外部割り込み(IN01)で発生します。

WM_APCI34_IN02 外部割り込み(IN02)で発生します。

WM_APCI34_IN09 外部割り込み(IN09)で発生します。

WM_APCI34_IN10 外部割り込み(IN10)で発生します。

WM_APCI34_TIMER タイマー割り込みで発生します。

製品のお問い合わせについて

- ◆ お買い求めいただいた製品に対する次のようなお問い合わせは、お求めの販売店又は株式会社アドテックシステムサイエンスの各営業所にご連絡ください。
 - ・ お求めの製品にご不審な点や万一欠品があったとき
 - ・ 製品の修理
 - ・ 製品の補充品や関連商品について
 - ・ 本製品を使用した特注製品についてのご相談

- ◆ 技術サポート —— 技術的な内容のお問い合わせは、「ファックス」「郵送」「E-mail」のいずれかにて、下記までお問い合わせください。また、お問い合わせの際は、内容をできるだけ詳しく具体的にお書きくださるようお願いいたします。

————— 技術的な内容のお問い合わせ先 —————

株式会社 アドテック システム サイエンス テクニカルサポート
〒240-0005

神奈川県横浜市保土ヶ谷区神戸町 134 YBP ウェストタワー 8F

E-mail support@adtek.co.jp

Fax 045-331-7770

改訂履歴

発行年月日 2004年03月31日 第1版

発行年月日 2005年03月23日 第2版
本社住所を変更

aPCI-P34
ソフトウェアマニュアル

第2版発行 2005年03月23日
発行所 株式会社 アドテック システム サイエンス
〒240-0005 神奈川県横浜市保土ヶ谷区神戸町134
YBP ウエストタワー 8F
Tel 045-331-7345 (代) Fax 045-331-7770

不許複製

aPCI-013-050323
© 2004-2005 ADTEK SYSTEM SCIENCE Co.,Ltd.