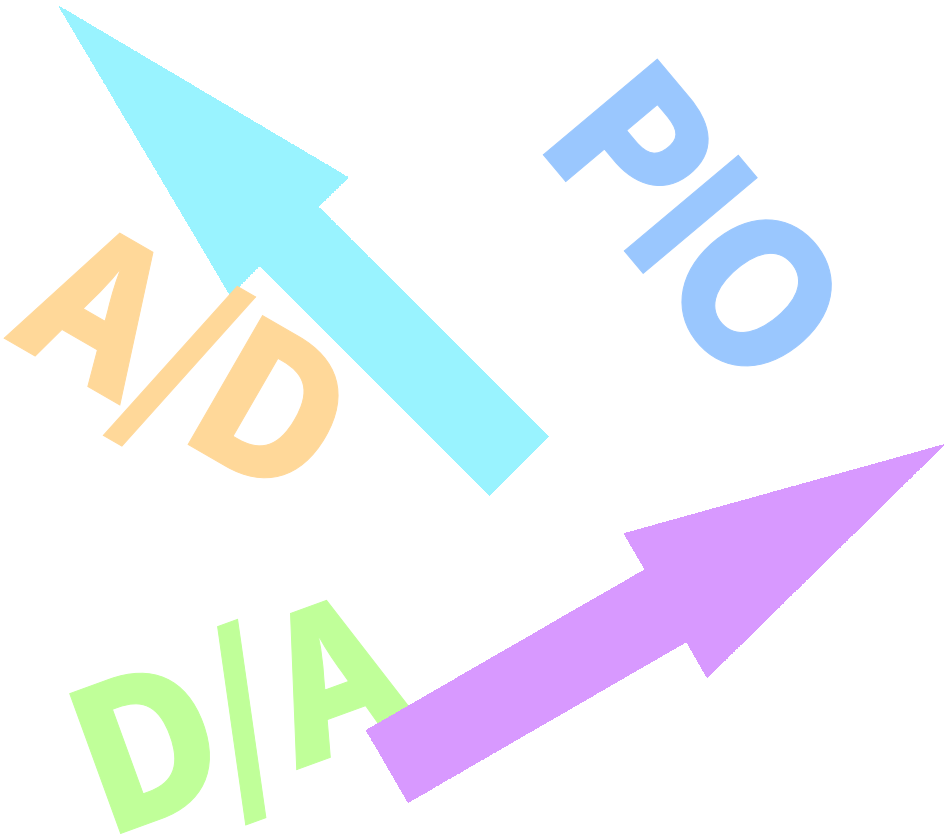
TUSB-ADAPIO

USBインタフェース付き計測制御ユニット

取扱い説明書



本文中のマークについて(必ず始めにお読み下さい)

この取扱説明書には、あなたや他の人々への危害や財産への損害を未然に防ぎ、本製品を安全にお使いいただくために、守っていただきたい事項を示しています。
その表示と図記号の意味は次のようになっています。内容をよみ理解してから本文をお読み下さい。

⚠ 警告

この表示を無視して、誤った取扱をすると、人が死亡または重傷を負う可能性がある内容を示しています。

⚠ 注意

この表示を無視して、誤った取扱をすると、人が損害を負う可能性が想定される内容および物的損害のみの発生が想定される内容を示しています。

① 製品の仕様および取扱説明書の内容は予告なく変更することがあります。
② 本製品および本取扱説明書の一部または全部を無断転載することは禁じられています。
③ 本取扱説明書の内容は万全を期して作成いたしましたが、万が一ご不審な事やお気づきの事がございましたら、（株）タートル工業　サービス課までご連絡下さい。
④ 当社では、本製品の運用を理由とする損失、逸失利益等の請求につきましては、上記に関わらずいかなる責任も負いかねますので、予めご了承下さい。
⑤ 本製品は、人命に関わる設備や機器、高度な信頼性を必要とする設備や機器などへの組込や制御などへの使用は意図されておりません。これら設備や機器などに本装置を使用され人身事故、財産損害などが生じても、当社はいかなる責任も負いかねます。
⑥ 本製品およびソフトウェアが外国為替及び外国貿易管理法の規定により戦略物資（又は役務）に該当する場合には日本国外へ輸出する際に日本国政府の輸出許可が必要です。

使用上の警告と注意

⚠ 警告

入出力端子に仕様に規定された信号以上の高電圧をかけないで下さい。高電圧をかけると感電の危険性と装置破損の可能性があります。

電源アダプタは指定の物をご使用下さい。誤った電源を入力すると感電の危険性と装置破損の可能性があります。

水や薬品のかかる可能性のある場所でご使用ならさないでください。火災やその他の災害の原因となる可能性があります。

発火性ガスの存在するところでご使用なさらないでください。引火により火災、爆発の可能性があります。

煙や異臭の発生した時は直ちにご使用をおやめ下さい。ACアダプタおよびUSBケーブルを取り外し、当社サービス課までご相談下さい。

⚠ 注意

温度の高い場所では使用しないでください。故障や火災の原因となります。

不安定な所には設置しないでください。落下によりけがをする恐れがあります。

腐食性のあるガスの存在するところでは使用しないで下さい。故障や火災の原因となります。

目次

1 はじめに

この度は、（株）タートル工業製のUSBインターフェース付き計測制御ユニットTUSB-ADAPIOをお買い求めいただき、誠にありがとうございます。
本書は、本製品の特徴、使用方法、取扱における注意事項、その他本製品に関する情報など、本製品をご使用される上で必要な事項について記述されております。
本製品の使用には製品の性質上、若干の電子回路の知識を必要とします。誤った使用をすると本製品の破損だけでなく重大な事故が発生する事も考えられます。本書の内容をよくご理解の上、正しくご使用下さる様お願いします。

1.1 製品概要

本製品は、先進のインタフェースであるUSB(Universal Serial Bus)を使用した計測、制御ユニットです。ADコンバータ、DAコンバータ、ディジタル入出力を本製品一つに全て内蔵し、計測制御において幅広い応用が可能です。ドライバソフトウェアおよびVisual C++ 6.0 とVisual Basic 6.0のサンプルソフトウェアを付属しておりますので、これらの応用によって短時間に利用する事が可能です。

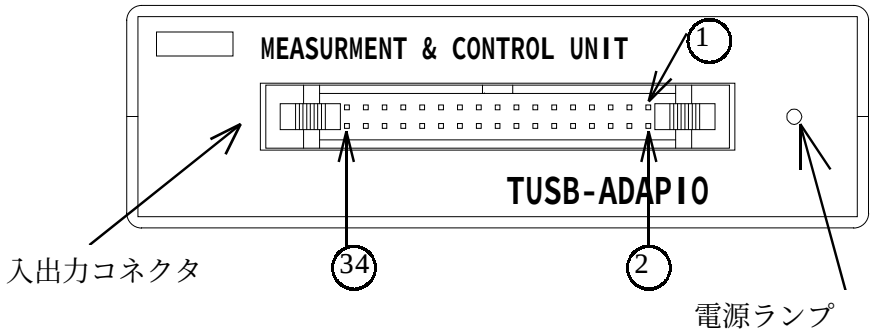
1.2 製品構成

本製品には以下の物が含まれます。
① TUSB-ADAPIO本体
② USBケーブル(1m)
③ 取扱説明書(本書)
④ 添付ソフトウェア(3.5インチ2HDフロッピーディスク1.44MB 1枚)
⑤ ユーザ登録書

不足品などがあれば、当社サービス課までご連絡下さい。(ご連絡先 30ページ)

2 各部の名称

2.1 フロントパネル

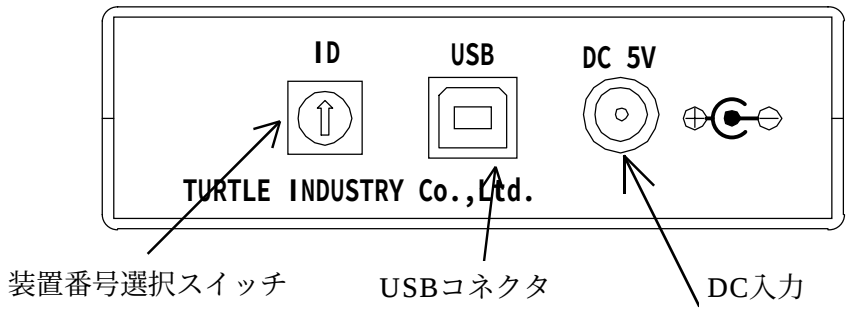


入出力コネクタ　：　フラットケーブル用34ピンコネクタ(ロック付き)です。
ADコンバータ、DAコンバータ、ディジタル入出力の入出力をここから行います。

電源ランプ　　：　電源ON時に点灯します。

※　○で囲まれた数字はコネクタのピン番号です。

2.2 リアパネル



USBコネクタ　：　コンピュータと付属のUSBケーブルで接続します

DC入力　　：　外部電源使用時に専用電源を接続します

ユニット番号選択(ID):　本ユニットのユニット番号を選択します

3 各端子説明

3.1 入出力コネクタ

ADコンバータ、DAコンバータ、ディジタル入出力の各入出力はフロントパネルの入出力コネクタを使用します。

ケーブル側コネクタ：ヒロセ電機（株）製 HIF3BA-34D-2.54Rまたは同等品

ピン番号	名称	機能
1	AD0	ADコンバータ入力チャンネル0
2	GND	信号グラウンド
3	AD1	ADコンバータ入力チャンネル1
4	GND	信号グラウンド
5	AD2	ADコンバータ入力チャンネル2
6	GND	信号グラウンド
7	AD3	ADコンバータ入力チャンネル3
8	GND	信号グラウンド
9	AD4	ADコンバータ入力チャンネル4
10	GND	信号グラウンド
11	AD5	ADコンバータ入力チャンネル5
12	GND	信号グラウンド
13	DA0	DAコンバータ出力チャンネル0
14	GND	信号グラウンド
15	DA1	DAコンバータ出力チャンネル1
16	GND	信号グラウンド
17	PIO0	ディジタル入出力チャンネル0
18	PIO1	ディジタル入出力チャンネル1
19	PIO2	ディジタル入出力チャンネル2
20	PIO3	ディジタル入出力チャンネル3
21	PIO4	ディジタル入出力チャンネル4
22	PIO5	ディジタル入出力チャンネル5
23	PIO6	ディジタル入出力チャンネル6
24	PIO7	ディジタル入出力チャンネル7
25	GND	信号グラウンド
26	PIO8	ディジタル入出力チャンネル8 / クロック出力
27	PIO9	ディジタル入出力チャンネル9
28	PIO10	ディジタル入出力チャンネル10
20	PIO11	ディジタル入出力チャンネル11
30	PIO12	ディジタル入出力チャンネル12
31	PIO13	ディジタル入出力チャンネル13
32	PIO14	ディジタル入出力チャンネル14
33	PIO15	ディジタル入出力チャンネル15 / ADサンプルトリガ入力
34	GND	信号グラウンド

※ DAコンバータの出力電流は仕様上10mA以上となっております。しかし、ユニット内部環境の温度が25℃で出力電圧が2.5Vの時には約50mAの出力が可能です。温度が高くなると出力可能電流は極端に小さくなりますのでご注意下さい。
※ ソフトウェアでクロック出力を選択した場合にはPIO8はクロック出力となります。
※ ADコンバータの連続取り込みの動作中はPIO15はADコンバータのトリガと併用されます。

3.2 電源入力コネクタ

本ユニットはUSBバスから供給されるDC5V電源で動作します。ただし、以下の様な場合があります。必要に応じて外部電源を使用してください。

1) コンピュータがサスペンド状態になるとUSBに供給される電源が遮断される可能性があります。
2) サスペンド状態で電源が遮断されなくとも、USB機器の使用出来る電源電流はサスペンド状態では500μAにまで制限されます。しかし、本ユニットは約70mAほど消費するため、この時には低消費電力状態で待機しなければなりません。低消費電力状態では入出力のデバイスは全てOFFになるため、構成によっては本ユニットまたは相手接続装置に動作異常や故障の発生する可能性があります。
3) ハブには自己電源をもつセルフパワードハブと自己電源をもたないバスパワードハブがあります。後者の場合は内部に電源を持たないためUSBラインから電源をとることになります。USBラインから供給される電源の電流は標準で100mAまでしか利用できないので、ハブの消費電流、本ユニットの消費電流、他の接続機器の消費電流の合計がこの値を超えない様にシステムを構築しなければなりません。本ユニットからIOを通して外部機器に電流を供給する場合にはさらに増加しますのでご注意下さい。

外部電源は安定化されたDC5V電源が必要となります。外部電源を使用される場合には専用ACアダプタ(別売)をご利用下さい。

3.3 USBコネクタ

付属のUSBケーブルを使用して、ご利用されるコンピュータまたはハブに接続してください。
※ 初めて接続される時にはインストール作業が必要です。

4 インストールの方法

4.1 デバイスのインストール(Windows 98)

インストールを始める前に付属のフロッピーディスク、ご使用のコンピュータ、TUSB-ADAPIO(本ユニット)をご用意下さい。
以下の作業は初回のみ行います。2回目以降はTUSB-ADAPIOをご使用のコンピュータに接続すると自動的に認識します。

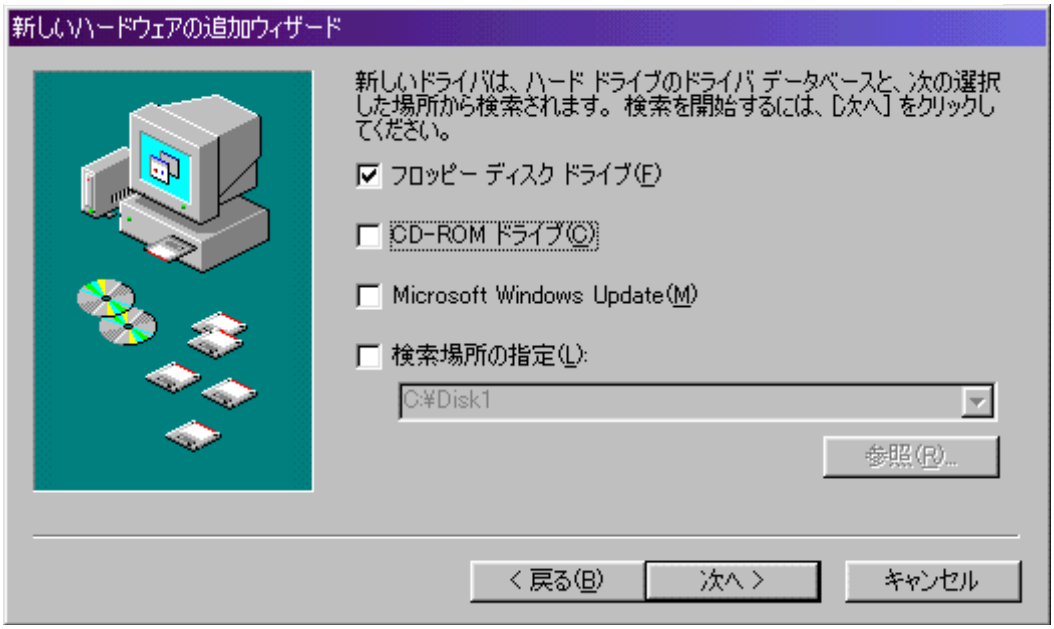
TUSB-ADAPIOとコンピュータを付属のケーブルで接続します。
(コンピュータは予め起動しておきます)

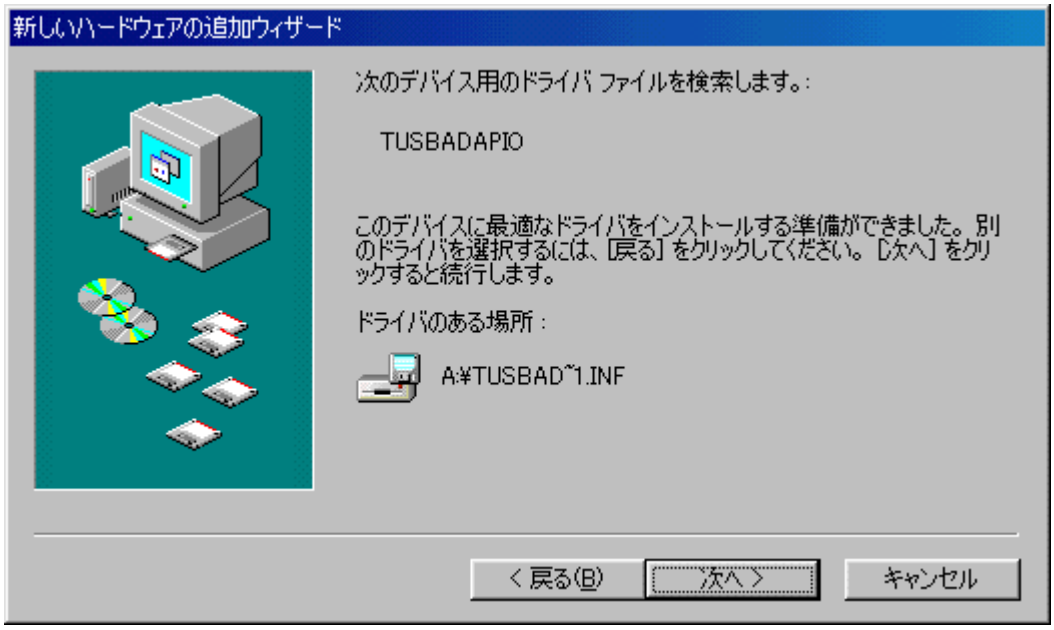
接続すると自動的にデバイスの情報を取得し、しばらくすると以下の様な画面が表示されます。

次へを押します。

下の画面が表示されたら「使用中のデバイスに最適なドライバを検索する (推奨)」を選択して次へを押してください。

下の画面が表示されたら「フロッピーディスクドライブ」をチェックして、次へを押してください。





上記の画面を確認して次へを押します。
最後に下記の画面が表示されますので、確認の上、完了を押してください。

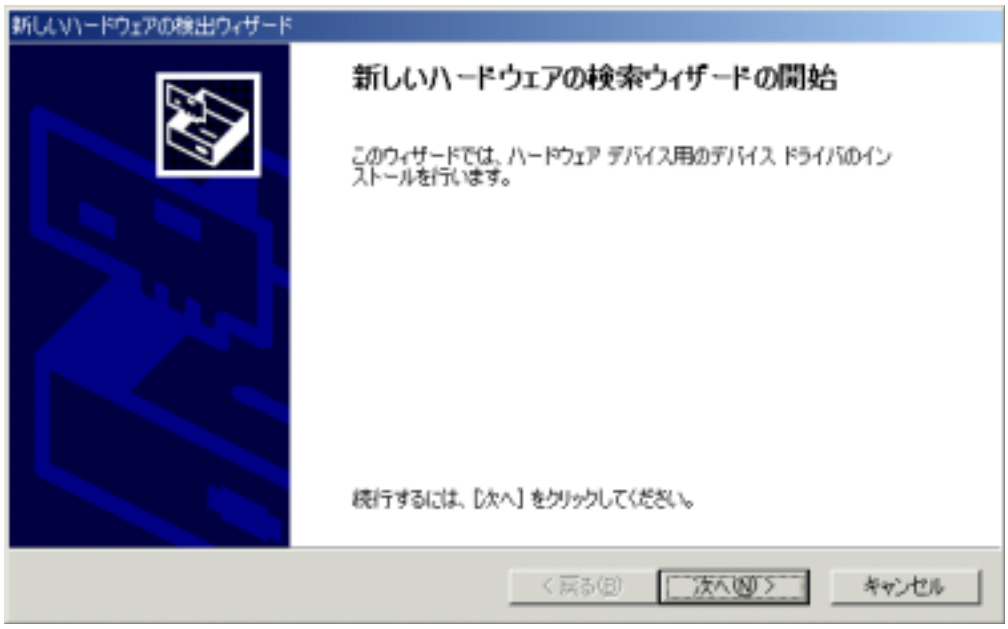
全て終了したら、システムを再起動してください。これでドライバのインストール作業は完了です。

4.2　デバイスのインストール(Windows 2000)

インストールを始める前に付属のフロッピーディスク、ご使用のコンピュータ、TUSB-ADAPIO(本ユニット)をご用意下さい。
設定されている権限によってはドライバのインストールが出来ない事があります。その場合適切な権限のユーザでインストールするか、ご使用システムの管理者にお問い合わせ下さい。
以下の作業は初回のみ行います。2回目以降はTUSB-ADAPIOをご使用のコンピュータに接続すると自動的に認識します。

TUSB-ADAPIOとコンピュータを付属のケーブルで接続します。
　　(コンピュータは予め起動しておきます)

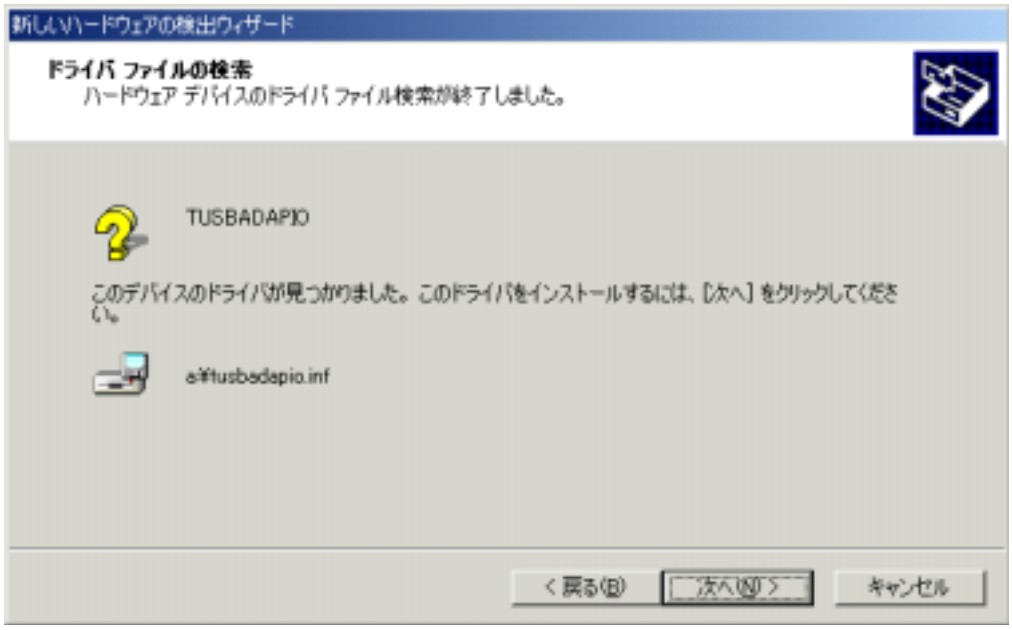
接続すると自動的にデバイスの情報を取得し、しばらくすると以下の様な画面が表示されます。



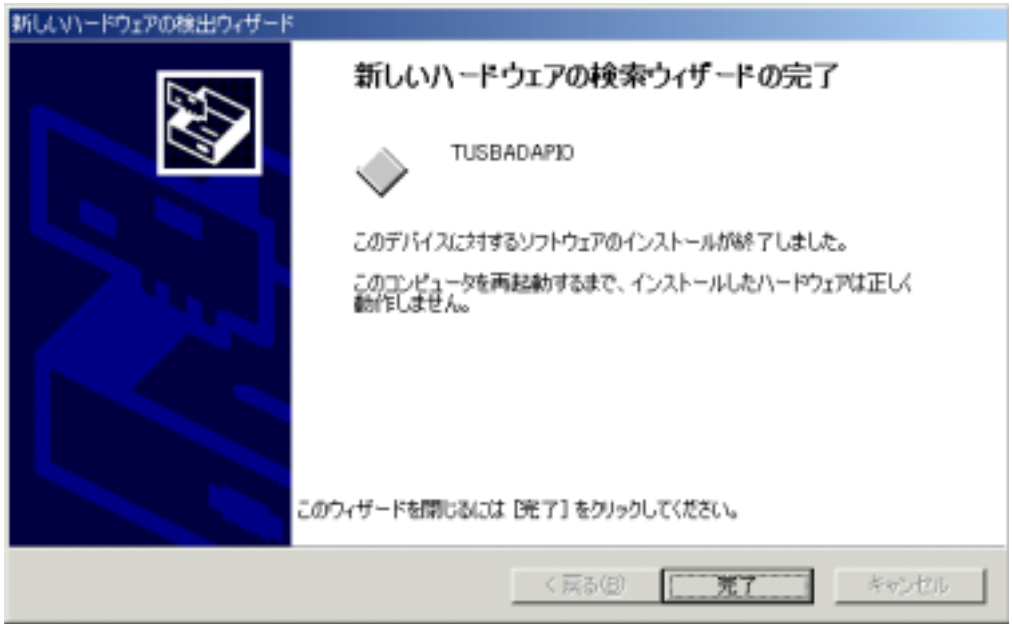
次へを押します。

下の画面が表示されたら「デバイスに最適なドライバを検索する (推奨)」を選択して次へを押してください。

下の画面が表示されたら「フロッピーディスクドライブ」をチェックして、次へを押してください。



上記の画面を確認して次へを押します。
最後に下記の画面が表示されますので、確認の上、完了を押してください。



全て終了したら、システムを再起動してください。これでドライバのインストール作業は完了です。

4.3　サンプルソフトのインストールと使い方

サンプルソフトはプロジェクトソースと共に以下の場所に格納されておりますので、適切な場所にコピーしてご使用下さい。

```
<付属フロッピーディスク>
  |-[DEV]
      |-[VB]          : Visual Basic用サンプル
      |               : プロジェクト
      |-[VC]          : Visual C++ 用サンプル
      |               : プロジェクト
      |-[TOOLS]       :DLL,LIB,H,BASファイル
```

Visual C++用のサンプルソフトとVisual Basic用のサンプルソフトの動作は同一です。Visual Basicのプロジェクトフォルダの中には構築後の実行ファイルが入っており、そのまま実行する事が出来ます。Visual C++のプロジェクトフォルダの中には実行ファイルはございませんので使用する場合には開発ツールで構築してください。

各アプリケーションとも起動後、本体装置のID番号を選択し、Openボタンを押下するとかくコマンドが実行できます。ボタン名にはドライバ関数の名前と同様の名前が付いております。各ボタンを押下する事により各関数の実行を確認する事が出来ます。

詳しくは各プロジェクトのソースコードをご覧下さい。

5 機能説明

5.1 ADコンバータの機能

シングル取込み
ADコンバータ6チャンネルの内、指定したチャンネルから1サンプルデータを取り込む事が出来ます。連続取込み中にこの機能を使用する事は出来ません。

変換範囲
-2.5V～2.5Vの範囲の電圧をディジタル値に変換します。変換値はオフセットストレートバイナリとなります。-2.5Vの時の変換値は0(Hex)、0Vの時の変換値は200(Hex)、2.5Vの時の変換値は3FF(Hex)となります。

外部トリガ連続取込み
外部トリガ (PIOのADコンバータトリガ入力端子を使用) 信号の立ち下がりに連動してADコンバータのデータを取得し、自動的に内部メモリに保存する事ができます。
○ 連続取込みが出来るのは0～3チャンネルのみです。
○ 1トリガでサンプルするチャンネル数を0～3の間で指定できます。
○ 1トリガでサンプルする数は各チャンネル1データです。
○ メモリは全チャンネル合計で512データです。
○ 10KHzのサンプルトリガで4チャンネルサンプルする事が出来ます。

アナログレベル検出トリガ連続取込み
基本的な動作は「外部トリガ連続取込み」と同様です。加えて次の機能を持っています。
○ 外部トリガが入力する毎に指定チャンネルをAD変換し、その値を評価します。
○ 変換値が前回の変換値に対して閾値を超えて大きい(立ち上がり)か小さい(立ち下がり)かを判定し、トリガ条件を判断するとその後のAD変換値をメモリに保存します。
○ 条件検出後は変換値に関わりなく外部トリガが入力する毎にAD変換し、メモリに保存し、保存指定数を越えるとメモリへの保存は行いません。
○ 判定を行うチャンネルを0～3の中から選択する事が出来ます。

5.2 DAコンバータの機能

出力範囲
0V～2.5Vを出力します。8ビットストレートバイナリです。
DAコンバータの変換時間は最大10μ秒となっておりますが、ソフトウェアによる繰り返し時のサイクルはさらに大きくなります。

5.3　ディジタルIOの機能

入出力選択
TUSB-ADAPIOのディジタル入出力は16ビット全てビット単位で入出力を決定する事ができます。起動時には全て入力となっております。

ロジックレベル
ディジタル入出力の論理レベルは全てTTLレベルです。

出力時のシンク、ソース電流
ソース電流は全て2mAです。シンク電流はPIO0～PIO7が2mA、PIO8～PIO15が10mAとなっております。

クロック出力
PIO8からはデューティー50%のクロックを出力する事ができます。

ADコンバータトリガ入力
PIO15はADコンバータのトリガ入力と併用されます。トリガ使用時には入力にしておかなければなりません。立ち下がりエッジでトリガされます。

5.4　ADコンバータ連続取込みの方法

本ユニットのADコンバータ部にて定周期連続取り込みを行うには外部よりディジタルトリガを一定周期で入力してください。本ユニット単独で連続取り込みを行うにはディジタル入出力のクロック出力とADコンバータトリガ入力を接続する事によって実現できます。
ADコンバータのトリガ入力は最高10KHzです。

6 ドライバソフトウェアの使用

6.1 開発環境の設定

Visual C++の場合

1 付属のフロッピーディスクより(TOOLSディレクトリの中)
TUSBADAPIO.LIB
TUSBADAPIO.H
を適当な場所にコピーします。

2 TUSBADAPIO.LIBファイルをプロジェクトに追加します。

3 使用するソースファイルにTUSBADAPIO.Hファイルをインクルードします。

設定は以上です。
本サンプルプログラムはVisual C++ 6.0で作成されました。

Visual Basicの場合

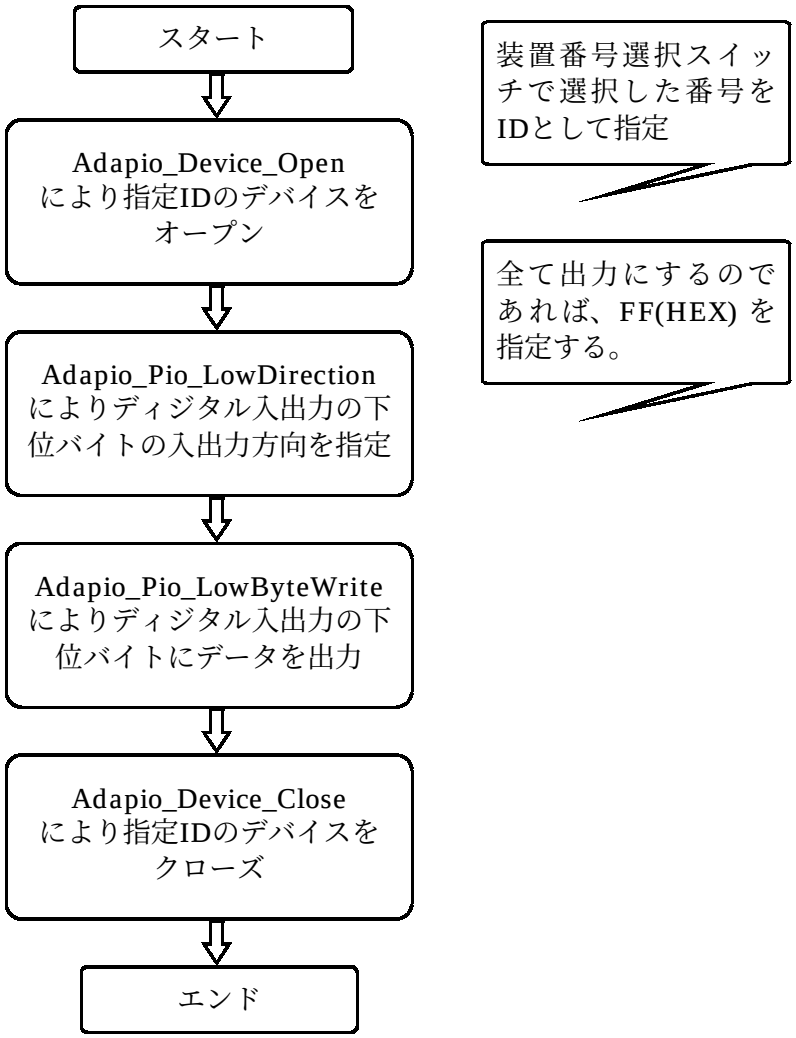
1 付属のフロッピーディスクより(TOOLSディレクトリの中)
TUSBADAPIO.BAS
を適当な場所にコピーします。

2 TUSBADAPIO.BASファイルをプロジェクトに追加します。

設定は以上です。
本サンプルプログラムはVisual Basic 6.0で作成されました。

6.2　基本的な関数使用の流れ

ここでは、関数の使用方法を簡単な例を元にご説明いたします。この関数はデバイスをオープンし、ディジタルIOポート0～7にデータを出力し、デバイスをクローズします。



オープンおよびクローズはプログラムの開始時および終了時に一回ずつ行う必要があります。一回の作業後毎にオープン、クローズを行う必要はありません。
その他の関数については関数リファレンスをご参照下さい。
２台以上使用する場合には装置の選択番号を変えて、それぞれについてオープンクローズを行って下さい。

7 ドライバ関数リファレンス

Adapio_Device_Open

C,C++宣言	short __stdcall Adapio_Device_Open(short id)
BASIC定義	[Private] Declare Function Adapio_Device_Open Lib "TUSBADAPIO.DLL" (ByVal id As Integer) As Integer

解説

指定ID(ユニット番号選択スイッチの値) のデバイスをオープンします。
このデバイスに関する各種関数を使用する前に必ず呼び出す必要が有ります。

引数

id	ユニット番号選択スイッチの番号(0から15)

戻り値

0:成功
1:ID番号が不正
2:ドライバがインストールされていない
3:デバイスはすでにオープンされている
4:接続されている台数が多すぎる(最高16台まで)
5:オープンできなかった
6:デバイスが見つからない

Adapio_Device_Close

C,C++宣言	void __stdcall Adapio_Device_Close(short id)
BASIC定義	[Private] Declare Sub Adapio_Device_Close Lib "TUSBADAPIO.DLL" (ByVal id As Integer)

解説

指定ID(ユニット番号選択スイッチの値) のデバイスをクローズします。

引数

id	ユニット番号選択スイッチの番号(0から15)

戻り値

なし

Adapio_Pio_LowDirection

C,C++宣言	short __stdcall Adapio_Pio_LowDirection (short id, unsigned char dir);
BASIC定義	[Private] Declare Function Adapio_Pio_LowDirection Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal dir As Byte) As Integer

解説

ディジタル入出力のLowByte(0～7ビット) の入出力方向を指定します。指定方法はビットパターンで行い、例えば引数dirにFF(HEX) を指定すれば全て出力、00(HEX)を指定すれば全て入力となります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dir	入出力方向の指定(00～FF)

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_HighDirection

C,C++宣言	short __stdcall Adapio_Pio_HighDirection (short id, unsigned char dir);
BASIC定義	[Private] Declare Function Adapio_Pio_HighDirection Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal dir As Byte) As Integer

解説

ディジタル入出力のHighByte(8～15ビット)の入出力方向を指定します。指定方法はビットパターンで行い、例えば引数dirにFF(HEX) を指定すれば全て出力、00(HEX)を指定すれば全て入力となります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dir	入出力方向の指定(00～FF)

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_LowByteWrite

C,C++宣言	short __stdcall Adapio_Pio_LowByteWrite (short id, unsigned char dat);
BASIC定義	[Private] Declare Function Adapio_Pio_LowByteWrite Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal dat As Byte) As Integer

解説

ディジタル入出力のLowByte(0～7ビット) の出力データを指定します。指定方法はビットパターンで行い、例えば引数datにFF(HEX)を指定すれば全てHigh、00(HEX)を指定すれば全てLowとなります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dat	出力データの指定(00～FF)

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_HighByteWrite

C,C++宣言	short __stdcall Adapio_Pio_HighByteWrite (short id, unsigned char dat);
BASIC定義	[Private] Declare Function Adapio_Pio_HighByteWrite Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal dat As Byte) As Integer

解説

ディジタル入出力のHighByte(8～15ビット)の出力データを指定します。指定方法はビットパターンで行い、例えば引数datにFF(HEX)を指定すれば全てHigh、00(HEX)を指定すれば全てLowとなります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dat	出力データの指定(00～FF)

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_LowByteRead

C,C++宣言	short __stdcall Adapio_Pio_LowByteRead (short id, unsigned char *dat);
BASIC定義	[Private] Declare Function Adapio_Pio_LowByteRead Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByRef dat As Byte) As Integer

解説

ディジタル入出力のLowByte(0～7ビット) の入力データを読込みます。入力値はビットパターンで参照変数datに入力されて戻されます。引数datがFF(HEX) であれば入力が全てHigh、00(HEX) であれば全てLowとなります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dat	入力データを格納するバッファ

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_HighByteRead

C,C++宣言	short __stdcall Adapio_Pio_HighByteRead (short id, unsigned char *dat);
BASIC定義	[Private] Declare Function Adapio_Pio_HighByteRead Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByRef dat As Byte) As Integer

解説

ディジタル入出力のHighByte(8～15ビット)の入力データを読込みます。入力値はビットパターンで参照変数datに入力されて戻ります。引数datがFF(HEX) であれば入力が全てHigh、00(HEX) であれば全てLowとなります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dat	入力データを格納するバッファ

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_WordWrite

C,C++宣言	short __stdcall Adapio_Pio_WordWrite (short id, int dat);
BASIC定義	[Private] Declare Function Adapio_Pio_HighByteWrite Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal dat As Long) As Integer

解説

ディジタル入出力のの出力データを指定します(16ビット)。指定方法はビットパターンで行い、例えば引数datにFFFF(HEX) を指定すれば全てHigh、0000(HEX)を指定すれば全てLowとなります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dat	出力データの指定(0000～FFFF)

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_WordRead

C,C++宣言	short __stdcall Adapio_Pio_WordRead (short id, int *dat);
BASIC定義	[Private] Declare Function Adapio_Pio_WordRead Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByRef dat As Long) As Integer

解説

ディジタル入出力の入力データを読込みます (16ビット)。入力値はビットパターンで参照変数datに入力されて戻ります。
引数datがFFFF(HEX)であれば入力が全てHigh、0000(HEX)であれば全てLowとなります。

引数

id	ユニット番号選択スイッチの番号(0から15)
dat	入力データを格納するバッファ

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_Clock_Out

C,C++宣言	short __stdcall Adapio_Pio_Clock_Out (short id, int count);
BASIC定義	[Private] Declare Function Adapio_Pio_Clock_Out Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal count As Long) As Integer

解説

ディジタル入出力のビット8にデューティー50%のクロックを出力します。入出力方向の設定に関わりなくビット8は出力となります。指定出来る周期は1(μ秒)～65535(μ秒)です。

引数

id	ユニット番号選択スイッチの番号(0から15)
count	周期 (count+ 1) μ秒の設定。範囲は(0～65535)

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Pio_Clock_Stop

C,C++宣言	short __stdcall Adapio_Pio_Clock_Stop (short id);
BASIC定義	[Private] Declare Function Adapio_Pio_Clock_Stop Lib “TUSBADAPIO.DLL” (ByVal id As Integer) As Integer

解説

出力しているクロックを停止します。

引数

id	ユニット番号選択スイッチの番号(0から15)

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Dac_Out

C,C++宣言	short __stdcall Adapio_Dac_Out (short id, short ch,unsigned char dat);
BASIC定義	[Private] Declare Function Adapio_Dac_Out Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal ch As Integer, ByVal dat As Byte) As Integer

解説

DAコンバータ出力値を設定します。出力値は0～+2.5V(0～FF) です。

引数

id	ユニット番号選択スイッチの番号(0から15)
ch	出力値を設定するチャンネル0又は1
dat	出力データ 00(HEX)～FF(HEX)

戻り値

0:成功
1:オープンされていない
2:失敗
3:ch不正(0か1でない)

Adapio_Adc_SingleSample

C,C++宣言	short __stdcall Adapio_Adc_SingleSample (short id, short ch,short *dat)
BASIC定義	[Private] Declare Function Adapio_Adc_SingleSample Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal ch As Integer, ByRef dat As Integer) As Integer

解説

ADコンバータの変換値を取得します。連続サンプリング実行時にこの関数を使用すると連続サンプリング動作が停止します。入力値は-2.5V～+2.5V(000～3FF)となります。

引数

id	ユニット番号選択スイッチの番号(0から15)
ch	入力するチャンネル0～5
dat	入力データ 000(HEX)～3FF(HEX)

戻り値

0:成功
1:オープンされていない
2:失敗
3:ch不正(0～5でない)

Adapio_Adc_DigitalTrg

C,C++宣言	short __stdcall Adapio_Adc_DigitalTrg (short id,short endCh,short buffSize)
BASIC定義	[Private] Declare Function Adapio_Adc_DigitalTrg Lib “TUSBADAPIO.DLL” (ByVal id As Integer ,ByVal endCh As Integer, ByVal buffSize As Integer) As Integer

解説

外部ディジタルトリガに連動して連続的にADコンバータデータを取得します。取得されたデータは装置内部の一時メモリに保存されます。取り込みは一回のトリガで0chから開始され、endChで指定されたチャンネルまで取り込みます。最高10KHzのTTLクロック入力で4chサンプルする事が出来ます。チャンネル間の時間差は約16.6μ秒程度です。バッファに全てデータが入ると変換を終了します。
指定するデータバッファの大きさは全取込チャンネルの合計です。0～2チャンネルを各10データずつ記録する場合はbuffSizeの値を30とします。

引数

id	ユニット番号選択スイッチの番号(0から15)
endCh	取り込み最終チャンネル(0から3)
buffSize	データバッファの大きさ(1～512)

戻り値

0:成功
1:オープンされていない
2:失敗
3:ch不正(0～3でない)
4:BufSize不正(0～512MAXでない)

Adapio_Adc_AnalogTrg

C,C++宣言	short __stdcall Adapio_Adc_AnalogTrg (short id,short endCh,short buffSize short threshold, short trgCh, short upOrDown)
BASIC定義	[Private] Declare Function Adapio_Adc_AnalogTrg Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByVal endCh As Integer, ByVal buffSize As Integer, ByVal threshold As Integer, ByVal trgCh As Integer, ByVal upOrDown As Integer) As Integer

解説

外部ディジタルトリガに連動して連続的にADコンバータデータを取得します。ディジタルトリガが入力される度にthresholdで設定された値と比較します。この値を超えて立ち上がりまたは、立ち下がりが検出された時からメモリへの保存を開始します。この立ち上がりまたは立ち下がりが検出されるまではメモリへの保存は行いません。その他の動作はAdapio_Adc_DigitalTrg関数と同様です。

引数

id	ユニット番号選択スイッチの番号(0から15)
endCh	取り込み最終チャンネル(0から3)
buffSize	データバッファの大きさ(1～512)
threshold	取り込み閾値001(HEX)～3FE(HEX)
trgCh	閾値を判定するチャンネつつル(0～3)
upOrDown	立ち上がり(0) 立ち下がり(1)　指定

Adapio_Adc_AnalogTrg

戻り値

0:成功
1:失敗
2:オープンされていない
3:EndCh不正(0～3でない)
4:BufSize不正(0～512MAXでない)
5:TrgCh不正(0～3でない)
6:UpOrDown不正(0か1でない)

Adapio_Adc_GetStatus

C,C++宣言	short __stdcall Adapio_Adc_GetStatus (short id,short *running, short *sampledNum)
BASIC定義	[Private] Declare Function Adapio_Adc_GetStatus Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByRef running As Integer, ByRef sampledNum As Integer) As Integer

解説

連続サンプリング動作のステータスを取得します。
Adc_Analog_TrgまたはAdc_Digital_Trgの実行後でないとステータス値には意味がありません。

引数

id	ユニット番号選択スイッチの番号(0から15)
running	動作状況　　動作中(1)　　サンプリング完了(0)
buffSize	サンプリング済みデータ数

戻り値

0:成功
1:オープンされていない
2:失敗

Adapio_Adc_GetDatas

C,C++宣言	short __stdcall Adapio_Adc_GetDatas (short id,short *databuf, short len)
BASIC定義	[Private] Declare Function Adapio_Adc_GetDatas Lib “TUSBADAPIO.DLL” (ByVal id As Integer , ByRef databuf As Integer, ByVal Leng As Integer) As Integer

解説

連続サンプリング済みデータバッファの内容を取得します。
取り込まれたデータは一回目より各チャンネルを繰り返しながら格納されています。チャンネル0の1回目のデータを0ch(1) とすると、endChが2の時データは次の様に格納されています。
0ch(1),1ch(1),2ch(1),0ch(2),1ch(2),2ch(2),0ch(3),1ch(3),2ch(3),…

引数

id	ユニット番号選択スイッチの番号(0から15)
databuf	取り込みデータ格納用バッファ
len	取得するデータ数

戻り値

0:成功
1:失敗
2:オープンされていない
3:len不正(0～512MAXでない)

8 その他

8.1 うまく動作しないとき

ユニットが認識(インストール)できない
OSはWindows 98または Windows 2000ですか ———>その他のOSには対応しておりません。 電源ランプが点灯していない時 ———>USBケーブルを差直してください。

8.2 USBについて

USBとはUniversal Serial Busの頭文字の略で、新しいコンピュータのインターフェースバスです。インターフェースのコストが低く使い易い事などからパーソナルコンピュータを中心に普及しました。USB1.1の仕様では、1.5Mbpsロースピードデバイスおよび12Mbpsハイスピードデバイスがあります。本ユニットでは12Mbpsハイスピード仕様になっております。

USBの主な特長	
高速	最高12Mbpsで通信可能(USB2.0では480Mbps)
接続が容易	ISAやPCIなどの拡張バスと違いケーブル1本で接続可能。コンピュータの動作中でも抜き差し可能。
多数接続可能	ハブの利用により最高127台 (ハブを含む) のデバイスを接続可能。
電源の供給	標準で100mA、最大で500mAの電源をバスで供給可能。
低コスト	多くのパーソナルコンピュータに標準で装備されており、安価なケーブル1本で接続可能。ただし、標準装備のポート数より多くのデバイスを接続する際にはハブが必要。

ハブについて
多数のUSBを接続するにはハブデバイスが必要です。ハブは1本のUSB線 (上流側) を複数のUSB線 (下流側) に分岐します。ハブにはバスパワードハブとセルフパワードハブがあり、前者は上流側の電源により動作しますが、後者は外部電源により動作します。ホストのポートからは標準で100mA、最大500mAの電流を供給する事が出来ます。バスパワードハブでは通常100mA未満の電流を消費するため、このハブに接続されたデバイスはバスから500mAを供給される事は出来ません。100mA以上の電流を消費するデバイスをバスパワードハブに接続する場合には注意が必要です。

ケーブルについて
USBケーブルはAタイプとBタイプに分かれます。ホストのポートはAタイプ、デバイス側はBタイプとなっており、誤挿入が起こらない仕様になっております。

転送速度について
USBの転送速度はきわめて高速ですが、接続されたデバイスの単位時間当たりのデータ転送量の総合計が最高転送量を超える事はありません。あるデバイスで大量のデータ転送を行うと他のデバイスの転送速度に影響の出る可能性があります。

8.3　連絡先

動作上の問題点および不明な点などのお問い合わせは下記までお願いします。
調査の上、当社よりご連絡差し上げます。

ご質問の際には動作環境等、なるべく詳細な情報を下さい。
特に次の情報は必ず記載してください。

ご使用のコンピュータの機種
ご使用OS(Windows98 , Windows98 SEなど)
メモリ容量
ハードディスクの容量
本ユニット以外でご使用されているUSB装置
こちらからご連絡差し上げる場合の貴ご連絡先

株式会社タートル工業
～　技術部　技術課　サービス係　～

E-mail	info@turtle-ind.co.jp
FAX	0298-43-2024
郵送	〒300-0842 茨城県土浦市西根南1-12-4

9 仕様

ADコンバータ部

チャンネル数	6ch
変換分解能	1/1024(10bit)
最大変換誤差	±4LSB
変換ビットパターン	オフセットストレートバイナリ
変換レート	10KHz/4ch
メモリ容量	512ワード
入力電圧範囲	±2.5V
入力抵抗	10KΩ以上

DAコンバータ部

チャンネル数	2ch
変換分解能	1/256(8bit)
最大変換誤差	±3LSB
変換ビットパターン	ストレートバイナリ
出力抵抗	50Ω
最大出力電流	＋10mA

ディジタル入出力部

ビット数	16bit
入出力割り当て	ビット単位で設定可能
出力レベル	TTL(2mA シンク, ソース) 8～16ビットのみシンク10mA ただし、ソースは同じ2mA

その他

信号入出力コネクタ	34Pフラットケーブル用(各部共通)
電源	USBまたは外部安定化DC
消費電流	約75mA
外部電源電圧	5.0VDC安定化されたもの
大きさ	30(h)×100(w)×140(d)mm(突起物含みません)
重さ	約300g

TUSB-ADAPIO取扱説明書
発行年月　　2000年10月　第1版
　　2001年8月　　第5版
発　　行　　株式会社　タートル工業
編　　集　　株式会社　タートル工業