

# TUSB-ADAPIOZ

USB I/F 付き組込型デジタル I/O ボード

## 取扱説明書



(64bitOS 対応ドライバ版)



## 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。

 <b>警告</b>	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。
 <b>注意</b>	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。万が一不審な事やお気付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。

©2014 Turtle Industry Co., Ltd. All rights reserved.

株式会社タートル工業の許可なく、本書の内容の複製、改変などを行うことはできません。

Microsoft, Windows, Windows NT, は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他、記載されている会社名、製品名は、各社の商標および登録商標です。

## 使用上の警告と注意



### 警告

接続機器の電源を全て切断してから端子台への接続および取り外しを行ってください。接続機器によっては感電の危険があります。



### 注意

端子に印加する電圧、電流は仕様に規定された値を守ってください。過熱による火災や漏電のおそれがあります。

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。

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。

煙や異臭の発生した時は直ちにご使用をおやめ下さい。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。

1.	はじめに.....	5
1.1	製品概要.....	5
1.2	製品構成.....	5
2.	各部の名称.....	6
2.1	フロントパネル.....	6
2.2	リアパネル.....	6
3.	各部説明.....	7
3.1	IO コネクタ.....	7
3.2	電源入力コネクタ.....	8
3.3	USB コネクタ.....	8
4.	機能説明.....	9
4.1	AD コンバータの機能.....	9
4.1.1	シングル取込.....	9
4.1.2	変換範囲.....	9
4.1.3	外部トリガ連続取込.....	9
4.1.4	アナログレベル検出トリガ連続取込.....	9
4.2	DA コンバータの機能.....	9
4.2.1	出力範囲.....	9
4.3	デジタル IO の機能.....	10
4.3.1	入出力選択.....	10
4.3.2	ロジックレベル.....	10
4.3.3	出力時のシンク、ソース電流.....	10
4.3.4	クロック出力.....	10
4.3.5	AD コンバータトリガ入力.....	10
4.4	AD コンバータ連続取込の方法.....	10
5	付属のディスクについて.....	11
5.1	ドライバ、アプリケーションディスクのディレクトリについて.....	11
6.	プログラミング.....	12
6.1	Visual C++(C++/CLD) での使用.....	12
6.1.1	使用準備.....	12
6.1.2	関数の呼び出し方法.....	12
6.2	Visual Basic での使用.....	13
6.2.1	使用準備.....	13
6.2.2	プロシージャの呼び出し方法.....	13
6.3	Visual C#での使用.....	13

---

6.3.1	使用準備 .....	13
6.3.2	関数の呼び出し方法 .....	13
6.4	関数説明 .....	14
7.	その他 .....	31
7.1	USB について .....	31
7.2	連絡先 .....	32
8.	仕様 .....	33
8.1	仕様 .....	33

## 1. はじめに

この度は、(株) タートル工業製の USB インタフェース付き計測制御ユニット TUSB-ADAPIOZ 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、サンプルソフトウェアが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。

※ 初めて接続される時にはドライバのインストール作業が必用です。付属のインストールマニュアルを参照してください。

### 1.2 製品構成

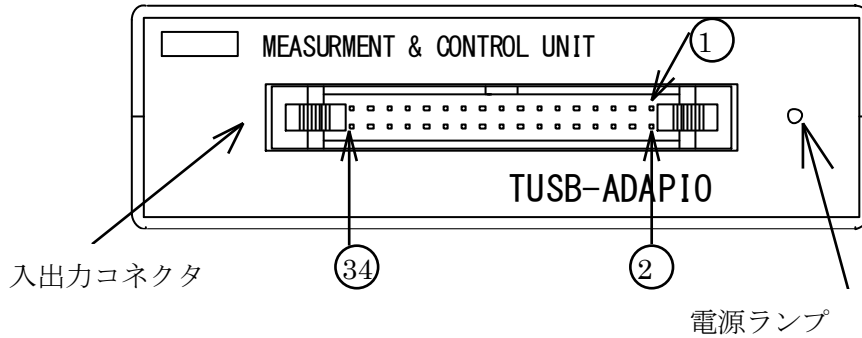
本製品には以下の物が含まれます。

- ① TUSB-ADAPIO 本体
- ② USB ケーブル(1m)
- ③ 添付ソフトウェア(CD-ROM 1 枚)

不足品などがあれば、当社サービス課までご連絡下さい。

## 2. 各部の名称

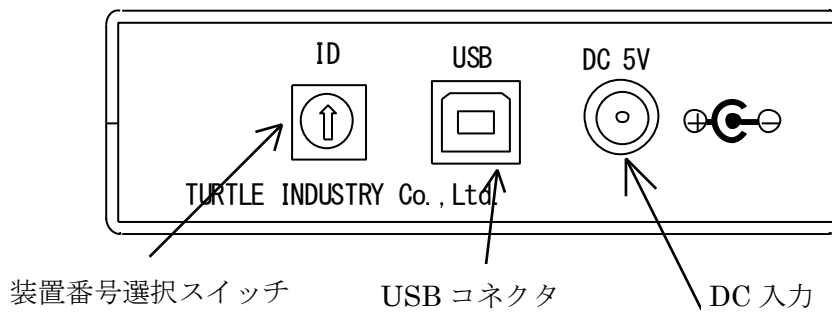
### 2.1 フロントパネル



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、デジタル信号の入出力を行います。
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)	ユニット選択に使用するユニット番号を設定します。

### 3. 各部説明

#### 3.1 IO 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
29	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°C 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ください。
- ※ 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 500  $\mu$ A 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ハブと自己電源をもたないバスパワーハブがあります。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ください。

※ 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. 機能説明

### 4.1 AD コンバータの機能

#### 4.1.1 シングル取込

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。

#### 4.1.2 変換範囲

-2.5V~2.5V の範囲の電圧をデ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。

#### 4.1.3 外部トリガ連続取込

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。

#### 4.1.4 アナログレベル検出トリガ連続取込

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。

### 4.2 DA コンバータの機能

#### 4.2.1 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 10 $\mu$ 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。

### 4.3 デジタル IO の機能

#### 4.3.1 入出力選択

TUSB-ADAPIOZ のデジタル入出力は 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。

#### 4.3.2 ロジックレベル

デジタル入出力の論理レベルは全て TTL レベルです。

#### 4.3.3 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。

#### 4.3.4 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。

#### 4.3.5 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。立ち下がリエッジでトリガされます。

### 4.4 AD コンバータ連続取込の方法

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。

## 5 付属のディスクについて

### 5.1 ドライバ、アプリケーションディスクのディレクトリについて

[ROOT] : ドライバ、アプリケーションディスクルート

|-[TUADAPIO]

|-[DRV] : ドライバ

|-[DOC] : ドキュメント(取扱説明書等)

|-[DEV] :

|-[TOOLS] :開発用 API 定義ファイル等

|-[VB] :Visual Basic .NET 用 サンプルプロジェクト

|-[CppCLI] :Visual C++(C++/CLI) 用 サンプルプロジェクト

|-[CSharp] :Visual C#用 サンプルプロジェクト

○ [DRV]ディレクトリ

この階層にはドライバファイルが入っております。ドライバのインストール時にはこのディレクトリをご指定下さい。

○ [DOC]ディレクトリ

取扱説明書等が PDF 形式で入っております。

○ [VB]ディレクトリ

Visual Basic のサンプルプログラムがソースファイルを含めたプロジェクトとして入っています。

○ [CppCLI]ディレクトリ

Visual C++(C++/CLI)のサンプルプログラムがソースファイルを含めたプロジェクトとして入っています。

○ [CSharp]ディレクトリ

Visual C#のサンプルプログラムがソースファイルを含めたプロジェクトとして入っています。

○ [TOOLS]ディレクトリ

開発時に必要な各種ファイルが入っています。

## 6. プログラミング

ここでは、Visual C++、Visual Basic、Visual C# で TUSB-ADAPIO 応用アプリケーションを開発する方法を説明します。本装置用のドライバをインストールするとドライバ操作用の DLL(ダイナミックリンクライブラリ)ファイルが同時にインストールされます。応用アプリケーションではこの DLL を介してドライバを操作します。

DLL を直接ロードして操作する方法もありますが、ここでは添付ディスクに付属の定義ファイルを利用した方法を説明します。

開発ツール(Visual C++、Visual Basic など)の使用方法についてはご説明いたしません。それぞれに付属のマニュアルかその他の資料をご参照ください。

※ 初めて接続される時にはドライバのインストール作業が必用です。付属のインストールマニュアルを参照してください。

注 1) 64bitOS 対応ドライバでは Visual Basic 6 は未対応です。

### 6.1 Visual C++(C++/CLI) での使用

#### 6.1.1 使用準備

Visual C++で使用するために以下のファイルを付属ディスクから適当な場所にコピーしてください。コピー先は指定しませんが、通常はプロジェクトのフォルダで支障ありません。

TUADAPIO.H                      ヘッダファイル

※ 付属ディスク内の DEVTOOLS フォルダに有ります。

ライブラリファイルはプロジェクトに追加してください。

ヘッダファイルは関数を使用するソースコードファイルの適当な場所にインクルードしてください。

※ ネイティブコードで使用する場合は TOOLS フォルダ下の Native フォルダ内の TUADAPIO.h をインクルードし、TUADAPIO.lib をプロジェクトに追加してください。

#### 6.1.2 関数の呼び出し方法

デバイスドライバの操作は全て機能毎の関数を呼ぶ事によって実現されます。Adapio\_Device\_Open 以外の関数は Adapio\_Device\_Open 関数が正常に処理された後でないと有効にはなりません。各機能関数を呼び出す前に Adapio\_Device\_Open を実行して機能関数の使用が終了したら Adapio\_Device\_Close 関数を呼び出してドライバを開放してください。ドライバ関数をアプリケーションで実行する場合にはアプリケーションの初めに Open し、アプリケーションの終了時に Close します。 TUSB-ADAPIO を 2 つのアプリ

ケーションで同時に操作する事は出来ません。

## 6.2 Visual Basic での使用

### 6.2.1 使用準備

Visual Basic で使用するために以下のファイルを付属ディスクから適当な場所にコピーしてください。コピー先は指定しませんが、通常はプロジェクトのフォルダで支障ありません。

TUADAPIO.vb                      標準ライブラリファイル

※ 付属ディスク内の DEVTOOLS フォルダに有ります。

ライブラリファイルは既存項目の追加でプロジェクトに追加してください。

### 6.2.2 プロシージャの呼び出し方法

デバイスドライバの操作は全て機能毎のプロシージャを呼ぶ事によって実現されます。Adapio\_Device\_Open 以外のプロシージャは Adapio\_Device\_Open が正常に処理された後でないと有効にはなりません。各機能のプロシージャを呼び出す前に Adapio\_Device\_Open を実行してプロシージャの使用が終了したら Adapio\_Device\_Close を呼び出してドライバを開放してください。デバイス機能をアプリケーションで実行する場合にはアプリケーションの初めに Open し、アプリケーションの終了時に Close します。

TUSB-ADAPIO を 2 つのアプリケーションで同時に操作する事は出来ません。

## 6.3 Visual C#での使用

### 6.3.1 使用準備

Visual C#で使用するために以下のファイルを付属ディスクから適当な場所にコピーしてください。コピー先は指定しませんが、通常はプロジェクトのフォルダで支障ありません。

TUADAPIO.cs                      ライブラリファイル

※ 付属ディスク内の DEVTOOLS フォルダに有ります。

ライブラリファイルはプロジェクトに追加してください。

### 6.3.2 関数の呼び出し方法

デバイスドライバの操作は全て機能毎の関数を呼ぶ事によって実現されます。Adapio\_Device\_Open 以外の関数は Adapio\_Device\_Open 関数が正常に処理された後でないと有効にはなりません。各機能関数を呼び出す前に Adapio\_Device\_Open を実行して機能関数の使用が終了したら Adapio\_Device\_Close 関数を呼び出してドライバを開放してください。ドライバ関数をアプリケーションで実行する場合にはアプリケーションの初めに Open し、アプリケーションの終了時に Close します。 TUSB-ADAPIO を 2 つのアプリケーションで同時に操作する事は出来ません。

#### 6.4 関数説明

ここでは、各関数(プロシージャ)のもつ機能などの詳細を説明します。

## Adapio\_Device\_Open

C 宣言	short Adapio_Device_Open ( short id )
	各言語での宣言については付属の定義ファイルを参照してください。

### 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-F)
----	----------------------

### 戻り値

- 0:成功
- 1: ID 番号が正しくありません
- 3:このデバイスは既にオープンされています
- 4:接続台数が多すぎます
- 5:オープンできません
- 6:デバイスが見つかりませんでした



## Adapio\_Device\_Close

C 宣言	void Adapio_Device_Close( short id )
	各言語での宣言については付属の定義ファイルを参照してください。

### 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-F)
----	----------------------

### 戻り値

なし

## Adapio\_Pio\_LowDirection

C 宣言	short Adapio_Pio_LowDirection(short id, unsigned char dir)
	各言語での宣言については付属の定義ファイルを参照してください。

### 解説

デジタル入出力の 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 [hex] )

### 戻り値

0:成功  
 1:オープンされていない  
 2:失敗

## Adapio\_Pio\_HighDirection

C 宣言	short Adapio_Pio_HighDirection(short id, unsigned char dir)
	各言語での宣言については付属の定義ファイルを参照してください。

### 解説

デジタル入出力の 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 [hex] )

### 戻り値

- 0:成功
- 1:オープンされていない
- 2:失敗た

**Adapio\_Pio\_LowByteWrite**

C 宣言	short Adapio_Pio_LowByteWrite(short id, unsigned char dat)
	各言語での宣言については付属の定義ファイルを参照してください。

**解説**

デジタル入出力の 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 [hex] )

**戻り値**

- 0:成功
- 1:オープンされていない
- 2:失敗

## Adapio\_Pio\_HighByteWrite

C 宣言	short Adapio_Pio_HighByteWrite(short id, unsigned char dat)
	各言語での宣言については付属の定義ファイルを参照してください。

### 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 [hex] )

### 戻り値

0:成功  
 1:オープンされていない  
 2:失敗

## Adapio\_Pio\_LowByteRead

C 宣言	short Adapio_Pio_LowByteRead(short id, unsigned char *dat)
	各言語での宣言については付属の定義ファイルを参照してください。

### 解説

デジタル入出力の LowByte(0~7 ビット)の入力データを読み込みます。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 宣言	short Adapio_Pio_HighByteRead(short id, unsigned char *dat)
	各言語での宣言については付属の定義ファイルを参照してください。

### 解説

ディジタル入出力の HighByte(8~15 ビット)の入力データを読み込みます。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 宣言	short Adapio_Pio_WordWrite(short id, int dat)
	各言語での宣言については付属の定義ファイルを参照してください。

**解説**

デジタル入出力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 [hex] )

**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 宣言	short Adapio_Pio_Clock_Out(short id, int count)
	各言語での宣言については付属の定義ファイルを参照してください。

### 解説

デ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( $\mu$  秒)~65535( $\mu$  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)
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\_Dac\_Out

C 宣言	short Adapio_Dac_Out(short id, short ch,unsigned char dat)
	各言語での宣言については付属の定義ファイルを参照してください。

### 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 宣言	short Adapio_Adc_SingleSample(short id, short ch,short *dat)
	各言語での宣言については付属の定義ファイルを参照してください。

### 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 宣言	short Adapio_Adc_DigitalTrg(short id,short endCh,short buffSize)
	各言語での宣言については付属の定義ファイルを参照してください。

### 解説

外部デ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 16.6 $\mu$  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 宣言	short Adapio_Adc_AnalogTrg(short id,short endCh,short buffSize, short threshold, short trgCh,short upOrDown)
	各言語での宣言については付属の定義ファイルを参照してください。

### 解説

外部デ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ル(0~3)
upOrDown	立ち上がり(0) 立ち下がり(1) 指定

### 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 宣言	short Adapio_Adc_GetStatus(short id,short *running, short *sampledNum)
	各言語での宣言については付属の定義ファイルを参照してください。

### 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\_TriggerまたはAdc\_Digital\_Trigger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 宣言	short Adapio_Adc_GetDatas(short id,short *databuf,short len)
	各言語での宣言については付属の定義ファイルを参照してください。

### 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)

## 7. その他

### 7.1 USBについて

※ ここでの記述は USB の一般的な記述となっております。

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フェースバスです。インタ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。USB2.0 の仕様では、1.5Mbps ロースピードデバイスおよび 12Mbps フルスピードデバイス、480Mbps のハイスピードデバイスが定義されています。

USB(フルスピード)の主な特長	
高速	12Mbps のバススピード(ハイスピードでは 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。
バス電源供給可能	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ハブとセルフパワーハブがあり、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となっております。

#### 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。



## 7.2 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、メーカー  
ご使用 OS(Windows 7 Home...など)  
OS の Edition( Home Professional など)  
OS のサービスパック  
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	support@turtle-ind.co.jp
FAX	029-843-2024
郵送	〒300-0842 茨城県土浦市西根南 1-12-4

## 8. 仕様

## 8.1 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**

**(64bitOS 用ドライバ対応版)**

発行年月      2014 年 2 月    第 13 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

©2014 株式会社   タートル工業