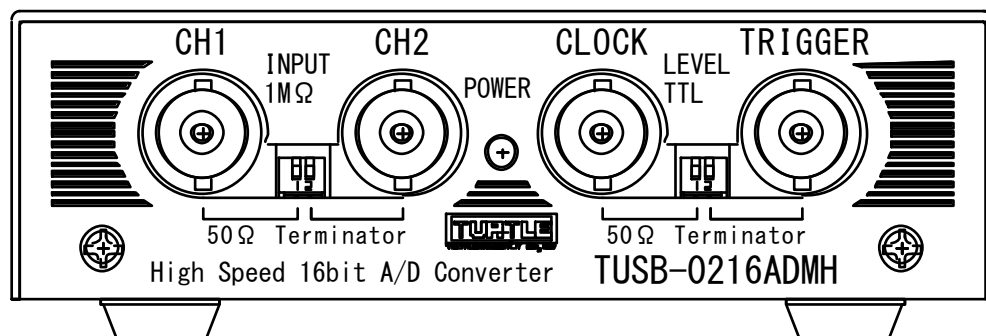


# TUSB-0216ADMH

USB インタフェース付き高分解能高速 AD コンバータユニット

## 取扱説明書



(Windows 7 64bit Windows 8 Windows 10 Windows 11 対応)



## 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但、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。

©2025 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
1.3 ご使用の前に.....	5
2. 各部の名称.....	6
2.1 フロント(前面).....	6
2.2 リア(後面).....	7
3. 各部説明.....	8
3.1 アナログ信号入力コネクタ.....	8
4 付属のディスクについて.....	10
4.1 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.....	10
5. AD コンバータの機能.....	11
5.1 ドライバ関数操作手順.....	11
5.2 取り込み動作モード.....	11
5.3 トリガ機能.....	12
5.4 アナログトリガ誤動作防止機能.....	12
5.5 変換クロック.....	13
5.6 連続取り込み速度について.....	13
5.7 入力レンジと変換値.....	13
5.8 デジタル入出力.....	14
6.プログラミング.....	15
6.1 Visual C++(C++/CLI) での使用.....	15
6.1.1 使用準備.....	15
6.1.2 関数の呼び出し方法.....	15
6.2 Visual Basic での使用.....	16
6.2.1 使用準備.....	16
6.2.2 プロシージャの呼び出し方法.....	16
6.3 Visual C#での使用.....	16
6.3.1 使用準備.....	16
6.3.2 関数の呼び出し方法.....	17
6.4 連続取り込みの方法.....	18
6.4.1 連続取り込みの仕組み.....	18
6.4.2 連続取り込みの方法.....	18
6.4.3 プレトリガ機能について.....	19
6.5 関数説明.....	20

---

6.6 エラーコード表.....	36
7. その他 .....	37
7.1 USB について .....	37
7.2 連絡先.....	38
8. 仕様.....	39
8.1 仕様概要.....	39
8.2 寸法図.....	40

## 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ADコンバータユニット TUSB-0216ADMH 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をお願いします。

### 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)を使用したコンピュータインターフェースユニットです。コンピュータを使用して高速アナログ電圧信号の計測ができます。ドライバソフトウェア、Visual C++ , Visual Basic , Visual C# の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-0216ADMH 本体
- ② USB ケーブル(1m)

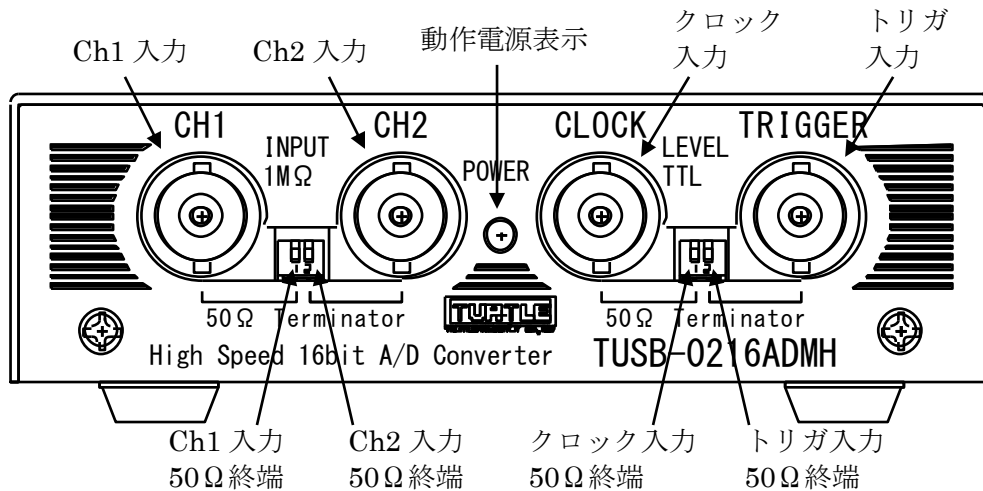
不足品などがあれば、当社サービス課までご連絡下さい。

### 1.3 ご使用の前に

本製品のご使用前にはデバイスドライバのインストールが必要です。ドライバインストールの手順に従って正しくドライバインストールを行って下さい。

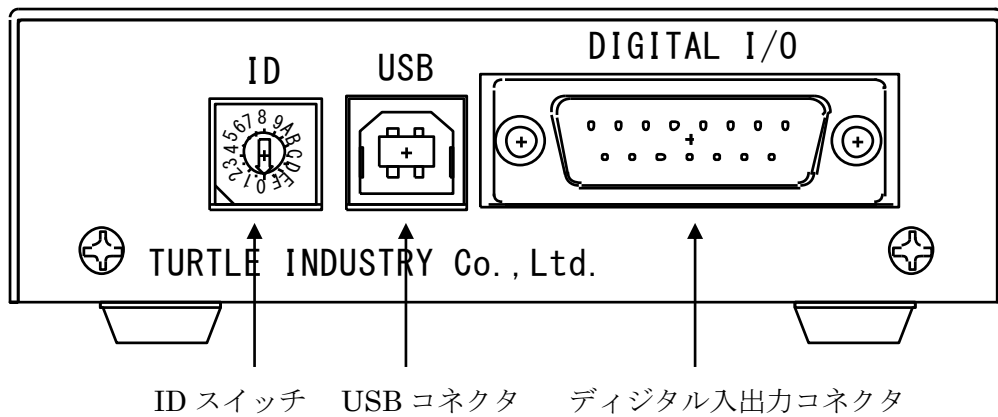
## 2. 各部の名称

### 2.1 フロント(前面)



Ch1,Ch2 入力	測定するアナログ電圧の入力コネクタです。入力約 1M $\Omega$ です。終端抵抗設定スイッチで 50 $\Omega$ にする事もできます。入力電圧は 0～2V および 1～1V から選択できます。
クロック入力	外部クロック使用時に入力します。 [外部クロック条件] <ul style="list-style-type: none"> <li>○ TTL 互換です。0-0.8V : Low 2.0-5.5V : High</li> <li>○ Low,High は 1:1 にして下さい</li> <li>○ 1MHz～25MHz</li> </ul> ※ 使用しない時は 50 $\Omega$ 終端を ON して下さい。
トリガ入力	外部デジタルトリガ使用時に入力します。 [外部トリガ条件] <ul style="list-style-type: none"> <li>○ TTL 互換です。0-0.8V : Low 2.0-5.5V : High</li> <li>○ 立ち上がりでトリガとします。</li> </ul> ※ 使用しない時は 50 $\Omega$ 終端を ON して下さい。
50 $\Omega$ 終端	各信号入力を 50 $\Omega$ で GND に終端します。下に倒すと ON(50 $\Omega$ 終端)です。
動作電源表示	ユニットの電源が入っている時に点灯します

## 2.2 リア(後面)



ID スイッチ	装置の ID を設定します。 一つの PC に複数の TUSB-0216ADMH を接続した時に 操作対象を指定するために使用します。
USB コネクタ	USB ケーブルを使用して PC と接続します。電源もここ から供給されます。
デジタル入出力 コネクタ	入力 4 ビット、出力 4 ビットのデジタル入出力です。 ○ TTL 互換です。 ○ 入力 0-0.8V : Low 2.0-5.5V : High ○ 出力 Low : 0.2V 以下 High 3V 以上 (何れも出力 100 $\mu$ A 以下の時)

※ 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。



### 3. 各部説明

#### 3.1 アナログ信号入力コネクタ

AD コンバータのアナログ入力には 2 つ有ります。入力電圧を同時又は交互で取込む事が可能です。入力コネクタの脇に配置されたスイッチによって入力抵抗を  $1\text{M}\Omega$  又は  $50\Omega$  にする事が出来ます。

#### 3.2 外部トリガ入力

AD 変換のサンプリング開始トリガを外部から入力することが出来ます。トリガ入力は信号の立ち上がりで有効となります。

- 信号のロジックレベルについて

信号の論理レベルは TTL となっております。

0-0.8V : Low    2.0-5.5V : High

スイッチの切り替えによって、 $50\Omega$  で終端可能です。この場合、終端後の電圧として High が十分 2.0V 以上である事を確認してください。

[外部トリガとサンプリング時点について]

外部トリガ入力後、次にサンプリングされたデータをトリガ時点とします。ただし、ハードウェアおよび同期処理の遅れなどにより 2~27ns 程度の遅れが発生します。

内部クロックを使用する場合は外部と同期しておりませんので、サンプリング周期程度のあいまいさがあります。(遅延時間は計算上の値で保証されません)

#### 3.3 外部クロック入力

AD 変換のクロックを外部から入力することが出来ます。

- 信号のロジックレベルについて

信号の論理レベルは TTL となっております。

0-0.8V : Low    2.0-5.5V : High

- Low , High レベルの各区間は 1:1 になる様にしてください。

- 周波数は 1MHz~25MHz の間で設定してください。

スイッチの切り替えによって、 $50\Omega$  で終端可能です。この場合、終端後の電圧として High が十分 2.0V 以上である事を確認してください。

[外部クロックとサンプリング時点について]

外部クロックの立下りでデータがサンプリングされます。ただし、回路上約 21ns の遅れがあります。(遅延時間は計算上の値で保証されません)

### 3.4 デジタル入出力コネクタ

入出力各 4 ビットのデジタル入出力コネクタです。論理レベルは TTL です。

ピン番号	名称	機能
1	NC	非接続※
2	GND	グラウンド
3	PI0	デジタル入力0
4	PI1	デジタル入力1
5	PI2	デジタル入力2
6	PI3	デジタル入力3
7	GND	グラウンド
8	PO0	デジタル出力0
9	PO1	デジタル出力1
10	PO2	デジタル出力2
11	PO3	デジタル出力3
12	GND	グラウンド
13	NC	非接続
14	NC	非接続
15	NC	非接続

- 入力 0-0.8V : Low 2.0-5.5V : High
- 出力 Low : 0.2V 以下 High 3V 以上  
(何れも出力 100  $\mu$  A 以下の時)

[ケーブル側コネクタについて]

D サブ 15 ピンオスをご使用下さい。とめネジはインチ(#4-40UNC)です。

### 3.5 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 付属のディスクについて

### 4.1 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

|-[TUSBADMH]

  |-[DRIVER] : 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。

## 5. AD コンバータの機能

本コンバータの変換データ取り込み方法には大きく分けて 2 種類の方法があります。繰り返しトリガ動作と非繰り返しトリガ動作です。

非繰り返し動作では AD コンバータスタート後にトリガを検出すると (プレトリガを設定している場合はトリガ前から) データのサンプリングを開始し、停止命令が来るかメモリが溢れるまでサンプリングし続けます。

繰り返し動作では AD コンバータスタート後にトリガを検出すると指定データ数分だけサンプリングをして待機状態になります。再度トリガを検出すると指定データ数分だけサンプリングをします。停止命令が来るかメモリが溢れるまでこの動作を繰り返します。

### 5.1 ドライバ関数操作手順

ドライバ関数を使用した AD 変換の操作手順の概要は以下の通りです。

- (1) 入力レンジ、内部クロックの設定を行う。
- (2) `Tusbadmh_Adc_Start` で変換開始する
- (3) (ソフトウェアトリガの場合のみ) `Tusbadmh_Trigger` で適切にトリガを発生させます。
- (4) `Tusbadmh_Status_Get` や `Tusbadmh_Length` を使用して現在の取り込み状態を確認する。
- (5) 取り込み状態に応じて `Tusbadmh_Data_Get` でデータを取得する。
- (6) 必要なデータを取り込み完了するまで (4) - (5) を繰り返す。
- (7) `Tusbadmh_Adc_Stop` で停止する。

### 5.2 取り込み動作モード

取り込み動作には 2 種類あります。連続取込とトリガ毎の繰り返し取り込みです。

連続取り込みでは開始後トリガ待ち状態になり、トリガ条件が発生すると連続的に取り込みを行いメモリに保存し続けます。FIFO 方式の動作をしますので、ご使用のコンピュータが許す限り制限無くデータを取り込む事ができます。プレトリガ値を設定するとトリガ前の変換値を取得できます。

繰り返し取り込みでは開始後トリガ待ち状態になり、トリガ条件が発生すると繰り返しモードでの指定数だけデータを取り込んだ後トリガ待ち状態になります。トリガ待ち状態の時にトリガ条件が発生すると指定数だけデータを取り込みます。何度でもトリガを受け付ける事が可能ですが、データ取りこぼしの無い様にデータを取得してください。繰り返しモードではプレトリガは無効です。

### 5.3 トリガ機能

連続取り込みでは次の 5 種類のトリガ方式が選択可能です。

- ① 外部デジタル : 外部トリガ入力の立上りでトリガします
- ② 信号立上り : チャンネル 1 入力の指定閾値立上りでトリガします
- ③ 信号立下り : チャンネル 1 入力の指定閾値立下りでトリガします
- ④ ソフトウェア : ソフトウェアによりトリガします

### 5.4 アナログトリガ誤動作防止機能

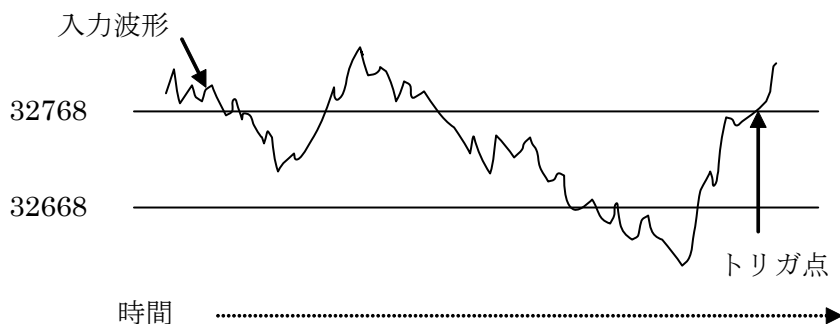
信号には一般的に何らかのノイズが含まれています。チャンネル 1 信号入力トリガを使用している場合、このノイズ信号の影響によって意図せずにトリガがかかる場合があります。そのため信号立上り、立下りトリガはノイズによる誤動作防止帯が設けられます。

立上りトリガは[トリガ設定閾値 - ノイズ防止レベル]未満の値を観測しないとトリガ検知を開始しません。

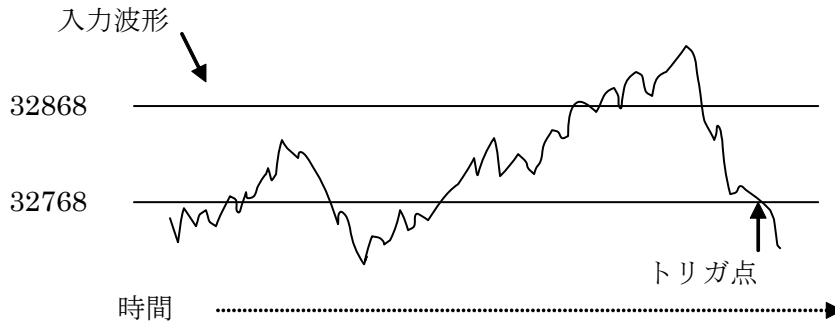
立下りトリガは[トリガ設定閾値 + ノイズ防止レベル]より大きい値を観測しないとトリガ検知を開始しません。

[ 関連ドライバ関数 : `Tusbadmh_ThLevel_Set` ]

< 閾値=32768 に立ち上がりトリガ設定した時の例 >



< 閾値=32768 に立ち下がりトリガ設定した時の例 >



- ※ 立上りトリガで[トリガ設定閾値 - ノイズ防止レベル]が 0 以下の場合は 0 を観測するとトリガ検知を開始します。
- ※ 立下りトリガで[トリガ設定閾値 + ノイズ防止レベル]が 65535 以上の場合は 65535 を観測するとトリガ検知を開始します。

## 5.5 変換クロック

取り込み間隔は内部クロック又は外部クロックが選択できます。外部クロックは 1MHz ～25MHz の範囲で任意に設定可能です。内部クロックは 1MHz～25MHz の範囲で設定できます。

[ 関連ドライバ関数 : Tusbadmh\_Clock\_Select ]

1MHz より低い周波数でのサンプリングが必要な場合は平均化機能を使用してください。

2 の n 乗の平均化が可能で、メモリに格納されるデータ数も(1/平均化回数)となります。

※ 外部変換クロックの Low , High レベルの各区間は 1:1 になる様にして、一定周波数としてください。

## 5.6 連続取り込み速度について

本装置の変換速度は 25MHz/ch です。装置内蔵メモリは 1ch 当たり 1M サンプルあり、25MHz で 2 チャンネル同時取り込みが可能です。しかし USB 転送速度の限界を超えておりますので内蔵メモリ長を超えての 25MHz 連続取り込みはできません。コンピュータにもよりますが、2ch 同時連続取り込みは 100kHz 程度までです。

## 5.7 入力レンジと変換値

入力範囲をバイポーラ(±1V)、ユニポーラ(0～2V)から選択出来ます。

取込データは 16 ビットストレートバイナリとなっております。

変換値	ユニポーラ	バイポーラ
<b>FFFF (HEX)</b>	> 1.99997V	> 0.99997V
<b>C000 (HEX)</b>	1.50000V	0.50000V
<b>8000 (HEX)</b>	1.00000V	0.00000V
<b>4000 (HEX)</b>	0.50000V	-0.50000V
<b>0000 (HEX)</b>	< 0.00000V	<-1.00000V

※ 上の表は設計値であって、精度を保証するものではありません。

※ 温度やサンプリング周波数などにより数百マイクロ V 程度のオフセットずれが発生しますので、必要に応じて 0V 値を差し引いて下さい。

## 5.8 デジタル入出力

デジタル入出力は、入力と出力各々4ビットあります。出力電流は最大 24mA です。

### 出力電圧特性

	出力状態	出力流入電流	出力電圧
出力電圧	“H”レベル	-100 $\mu$ A	3.0V 以上
		-18mA	2.4V 以上
		-24mA	2.2V 以上
	“L”レベル	100 $\mu$ A	0.2V 以下
		16mA	0.4V 以下
		24mA	0.55V 以下

## 6. プログラミング

ここでは、Visual C++、Visual Basic、Visual C#で TUSB-0216ADMH 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、Visual C#)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。

TUSBADMH.H                      ヘッダファイル

※ 付属ディスク内の DEV\TOOLS 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 TUSBADMH.h をインクルードし、TUSBADMH.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。Tusbsadmh\_Device\_Open 以外の関数は Tusbadmh\_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 Tusbadmh\_Device\_Open を実行して機能関数の使用が終了したら Tusbadmh\_Device\_Close 関数を呼び出してデバイ



スを開放してください。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** すれば充分です。

一つの TUSB-0212ADMH デバイスを 2 つのアプリケーションで同時にオープンする事は出来ません。一つのアプリケーションでの **Open**～**Close** の間は他のアプリケーションで同じデバイスを操作する事はできません。

## 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 1 つ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。

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

※ 付属ディスク内の DEV\TOOLS 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。Tusbadmh\_Device\_Open 以外のプロシージャは Tusbadmh\_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 Tusbadmh\_Device\_Open を実行してプロシージャの使用が終了したら Tusbadmh\_Device\_Close を呼び出してデバイスを開放してください。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** すれば充分です。

一つの TUSB-0212ADMH デバイスを 2 つのアプリケーションで同時にオープンする事は出来ません。一つのアプリケーションでの **Open**～**Close** の間は他のアプリケーションで同じデバイスを操作する事はできません。

## 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 1 つ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。

TUSBADMH.cs                      ライブラリファイル

※ 付属ディスク内の DEV\TOOLS フォルダに有ります。

ライブラリ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。

### 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。  
**Tusbadmh\_Device\_Open** 以外の関数は **Tusbadmh\_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。各機能の関数を呼び出す前に **Tusbadmh\_Device\_Open** を実行して関数の使用が終了したら **Tusbadmh\_Device\_Close** を呼び出してデバイスを開放してください。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** すれば充分です。

一つの TUSB-0212ADMH デバイスを 2 つのアプリケーションで同時にオープンする事は出来ません。一つのアプリケーションでの **Open**～**Close** の間は他のアプリケーションで同じデバイスを操作する事はできません。

## 6.4 連続取り込みの方法

ここでは、連続取り込みの仕組みとプログラミングの方法を説明します。

### 6.4.1 連続取り込みの仕組み

TUSB-0216ADMH 内には FIFO 方式のメモリが 1 チャンネルあたり 1M サンプル分あります。連続取り込みが開始されると、サンプリングされたデータはこのメモリ内に格納されてゆきます。同時にドライバは適時この FIFO 方式メモリからデータを読み出してコンピュータ内のメモリ(これも FIFO 方式の動作をします。容量は 2M サンプルで以後 PC メモリとします)に保存します。アプリケーションソフトウェアはドライバを使用してこの PC メモリ内のデータを読み出します。FIFO 方式メモリ、PC メモリ共に読み出されたデータは自動的に消去されます。

この動作が続く限り制限無く連続データを取り込み事が可能です。しかし、PC 内の処理が間に合わない場合や USB の通信が十分に確保できない場合はデータが失われる可能性があります。コンピュータや動作状態にもよりますが、おおよそ 100kHz 程度までであれば連続的に取り込み可能です。

### 6.4.2 連続取り込みの方法

Tusbadmh\_Adc\_Start でサンプリングを開始します。開始後にトリガ条件が成立するとデータ蓄積を開始します。ソフトウェアトリガの場合は必要な時点でトリガコマンド Tusbadmh\_Trigger を実行してください。トリガ条件の成立は Tusbadmh\_Status\_Get で確認する事ができます。

トリガ条件成立後は PC メモリにデータを蓄積しますが、データバッファがいっぱいになると自動的に取り込みを終了します。連続的に取り込む場合には適時 PC バッファを Tusbadmh\_Data\_Get で読み出して下さい。

- ※ USB の転送が間に合わない場合 FIFO バッファが一杯になる事があります。この場合には AD 変換が停止し新たなサンプリングデータは記録されません。FIFO バッファが一杯になった事を Tusbadmh\_Status\_Get で確認する事が出来ます。
- ※ USB の転送が間に合っている場合でもアプリケーションで PC バッファの読み込みが間に合わずに一杯になるとそこで連続取込は停止します。

#### 6.4.3 プレトリガ機能について

AD 変換開始時にプレトリガ長を 0 以外にするとプレトリガ機能が有効となります。プレトリガとはトリガ前の事です。トリガ発生前のデータが必要な場合は必要数を設定します。プレトリガ長を 100 と設定した場合は先頭の 100 データがプレトリガデータとなります。

※ 連続取り込み開始からプレトリガ長のデータ取得期間の間はトリガが無視されます。

## 6.5 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。

**Tusbadmh\_Device\_Open**

C 宣言	short Tusbadmh_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(0-15)
----	-----------------------

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_Device\_Close**

C 宣言	void Tusbadmh_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(0-15)
----	-----------------------

**戻り値**

なし

## Tusbadmh\_Dio\_Read

C 宣言	short Tusbadmh_Dio_Read ( short Id ,unsigned char *Data )
	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)のデバイスのデジタル入力ポートの入力値および現在の出力ポートの出力値を読み込みます。下位 4 ビットが入力値、上位 4 ビットが出力値です。取得した数値は 2 進数にした時の 0,1 のパターンで **Low,High** が示されます。

4 ビット単位の入出力読み込み値と **High**、**Low** の関係を下表に示します。

Data (HEX)	2 進 値	入出力ビット			
		3	2	1	0
0	0000	L	L	L	L
1	0001	L	L	L	H
2	0010	L	L	H	L
3	0011	L	L	H	H
4	0100	L	H	L	L
5	0101	L	H	L	H
6	0110	L	H	H	L
7	0111	L	H	H	H

Data (HEX)	2 進 値	入出力ビット			
		3	2	1	0
8	1000	H	L	L	L
9	1001	H	L	L	H
A	1010	H	L	H	L
B	1011	H	L	H	H
C	1100	H	H	L	L
D	1101	H	H	L	H
E	1110	H	H	H	L
F	1111	H	H	H	H

## 引数

Id	ユニット番号選択スイッチの番号(0-15)
Data	データを格納するバッファのアドレス(ポインタ)

## 戻り値

エラーコード(エラーコード表参照)



**Tusbadmh\_Dio\_Write**

C 宣言	short Tusbadmh_Dio_Write( short Id, unsigned char Data )
	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)のデバイスのデジタル出力ポートの出力値を書き込みます。

書き込みは下位 4 ビットで、数値と High,Low レベルの関係は Tusbadmh\_Dio\_Read と同じです。

**引数**

Id	ユニット番号選択スイッチの番号(0-15)
Data	書き込むデータ

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_Adc\_Start**

C 宣言	short Tusbadmh_Adc_Start (short Id,int CycLen, int PreLen, unsigned char TrgSel, unsigned char Mode,unsigned char Ch1Only)
	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**

連続取り込みを開始します。サンプリングクロックの設定など連続取り込みに必要な設定はこの関数を呼ぶ前に行ってください。

**引数**

Id	ユニット番号選択スイッチの番号(0-15)
CycLen	1～1048576 繰り返しモードで使用する場合の 1 トリガあたりの取り込みデータ数
PreLen	0～1048576 プレトリガ長。プレトリガ機能を使用しない場合は 0 としてください。
TrgSel	トリガ種類の選択 0:外部トリガ      1:立上りトリガ 2:立下りトリガ    3:ソフトトリガ
Mode	トリガ動作モードの選択 0:トリガ後連続取り込みモードで動作します。 1:繰り返しトリガモードで動作します。
Ch1Only	0:Ch1 と Ch2 の両取り込みをする。      1:Ch1 のみ取り込む ※チャンネル当たりの転送速度は上がりますが、メモリは増えません

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_Adc\_Stop**

C 宣言	short Tusbadmh_Adc_Stop (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**

連続取り込みを停止します。

**引数**

id	ユニット番号選択スイッチの番号(0-15)
----	-----------------------

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_Status\_Get**

C 宣言	short Tusbadmh_Status_Get (short Id ,unsigned char *Status, unsigned char *OvfSt)
	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**

連続取り込み状態を確認する。

**引数**

Id	ユニット番号選択スイッチの番号(0-15)
Status	ステータス 0:停止中 1:トリガ待ち 2:トリガ後変換中
OvfSt	オーバーフロー状態 (オーバーフローしたチャンネルは連続取り込み停止します) 1: ch1 オーバーフロー 2: ch2 オーバーフロー 3: ch1 と ch2 オーバーフロー。 ※ 開始時に Ch1Only をセットした場合でも設定や動作状態で ch2 のオーバーフローが発生し、2 や 3 が返ります。Ch1Only が有効の場合、1 または 3 が返った時に Ch1 がオーバーフローしたと判断してください。

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_Length**

C 宣言	short Tusbadmh_Length (short Id,int *Len1, int *Len2, int *Rate1, int *Rate2 )
	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**

取り込み完了データ数を取得します。データバッファはデバイス内と PC 内にあります。デバイスから PC へのデータ転送はドライバで自動的に行います。この関数では PC 内のバッファに取り込まれたデータ長を返します。

Rate は装置内バッファの使用率で、100%になると自動的に取り込みを停止します。

**引数**

Id	ユニット番号選択スイッチの番号(0-15)
Len1	Ch1 取り込み済み長
Len2	Ch2 取り込み済み長
Rate1	Ch1 装置内バッファ使用率(%)
Rate2	Ch2 装置内バッファ使用率(%)

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_Data\_Get**

C 宣言	short Tusbadmh_Data_Get (short Id,unsigned char ch,int*Data, int *Leng)
	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**

取り込み済みデータを取得します。取得したデータはバッファ内から消去されます。

**引数**

Id	ユニット番号選択スイッチの番号(0-15)
ch	チャンネル 0:ch1 1:ch2
Data	取得のデータの格納先のポインタ
Leng	取り込み要求長。 呼び出し時は要求データ数を設定します。 戻る時は実際に取得できた数が入っています。

**戻り値**

エラーコード(エラーコード表参照)

## Tusbadmh\_Clock\_Select

C 宣言	short Tusbadmh_Clock_Select (short Id ,unsigned char ClkSel, unsigned char Div, unsigned char Ave)
	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

連続サンプリングのクロックの設定を行います。

### 引数

id	ユニット番号選択スイッチの番号(0-15)
ClkSel	クロックソース選択 0: 200MHz    1:20.48MHz    2:16.384MHz    3:12.8MHz 4:1.92MHz    5:Ext ※ 0～4 は内部クロックソースです
Div	クロックの分周比 0～199 ClkSel で選択したクロックをを Div+1 で割った値が内部クロックの周波数になります。 例 ClkSel = 0 (200MHz)    Div = 7 の場合 200MHz / (7+1) -> 25MHz ※ 必ず 1MHz～25MHz となる様に設定してください。 ※ 外部クロックの時は Div は最大 24 です。
Ave	平均化設定 0～8 取り込んだデータの平均化機能を設定します。平均化回数は 2 の Ave 乗となります。(サンプリング個数は平均回数分の 1 になります) 例) Ave = 4 の時 $2^4 = 16$ となり 16 回平均

### 戻り値

エラーコード(エラーコード表参照)

**Tusbadmh\_ThLevel\_Set**

C 宣言	short Tusbadmh_ThLevel_Set (short Id ,int ThLevel, int NLevel)
	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**

連続サンプリング時のアナログトリガ基準レベルの設定を行います。

**引数**

id	ユニット番号選択スイッチの番号(0-15)
ThLevel	アナログ信号立ち上がりおよび立下りトリガの時の基準レベル設定 1～65534(変換値単位)
NLevel	ノイズ除去レベル アナログ信号トリガのときの誤動作防止レベルの設定値 0～3277(変換値単位) ※ ノイズレベルよりも十分大きく、信号振幅よりも十分小さい値が適切ですが、不明な場合は 800 程度を設定してください。

**戻り値**

エラーコード(エラーコード表参照)



**Tusbadmh\_InputType**

C 宣言	short Tusbadmh_InputType (short Id ,unsigned char Type1, unsigned char Type2)
	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**

入力レンジの設定を行います。

**引数**

id	ユニット番号選択スイッチの番号(0-15)
Type1	Ch1 入力タイプ 0->バイポーラ(-1V~1V) 1->ユニポーラ(0V~2V)
Type2	Ch2 入力タイプ 0->バイポーラ(-1V~1V) 1->ユニポーラ(0V~2V)

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_CheckInputType**

C 宣言	short Tusbadmh_CheckInputType (short Id ,unsigned char *Type1, unsigned char *Type2)
	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**

現在設定されている入力レンジの確認を行います。

**引数**

id	ユニット番号選択スイッチの番号(0-15)
Type1	Ch1 入力タイプ 0->バイポーラ(-1V~1V) 1->ユニポーラ(0V~2V)
Type2	Ch2 入力タイプ 0->バイポーラ(-1V~1V) 1->ユニポーラ(0V~2V)

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_Trigger**

C 宣言	short Tusbadmh_Trigger (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**

ソフトウェアトリガをかけます。

**引数**

id	ユニット番号選択スイッチの番号(0-15)
----	-----------------------

**戻り値**

エラーコード(エラーコード表参照)

**Tusbadmh\_TransLimit**

C 宣言	short Tusbadmh_Trigger (short Id ,int Limit)
	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**

転送サイズ制限を設定します。USB 通信における転送データブロックのサイズを設定します。高速サンプリングかつ 1 回あたりの必要データ数が少ない場合にこの値を小さくすると応答性が良くなる場合があります。大量にデータを取得する時にこの値を小さくしすぎると転送効率が低下します。通常使用ではデフォルト値の 50000 データのままで設定不要です。

設定目安としては(スタート~ストップ間の)必要なデータ長より少し大きい値に設定してみてください。

**引数**

id	ユニット番号選択スイッチの番号(0-15)
Limit	転送サイズ制限(100~100000)

**戻り値**

エラーコード(エラーコード表参照)

## 6.6 エラーコード表

各機能関数(プロシージャ)から戻る処理結果コードの値の表です。

戻り値	状態
0	正常終了
1	ID 番号が異なる
2	ドライバが正常にインストールされていない
3	すでにデバイスはオープンされている
4	接続台数が多すぎる
5	デバイスをオープンできなかった
6	指定のデバイスが見つからない
7	指定のデバイスはオープンされてない
8	指定パラメータのエラー
9	USB 通信エラー
10	メモリが確保できない
11	連続取り込み動作中
12	連続取り込みデータは無い
13	連続取り込みが開始されていない
14	メモリがオーバーフローしました
15	データ列順エラー
99	その他のエラー

## 7. その他

### 7.1 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。

※ ここでの記述は USB の一般的な記述となっております。

USB の主な特長	
高速	12Mbps のバススピード(USB 2.0 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。
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  
OS の種類(Windows 10 など)  
OS のサービスパック  
OS のビット数 32bit / 64bit  
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

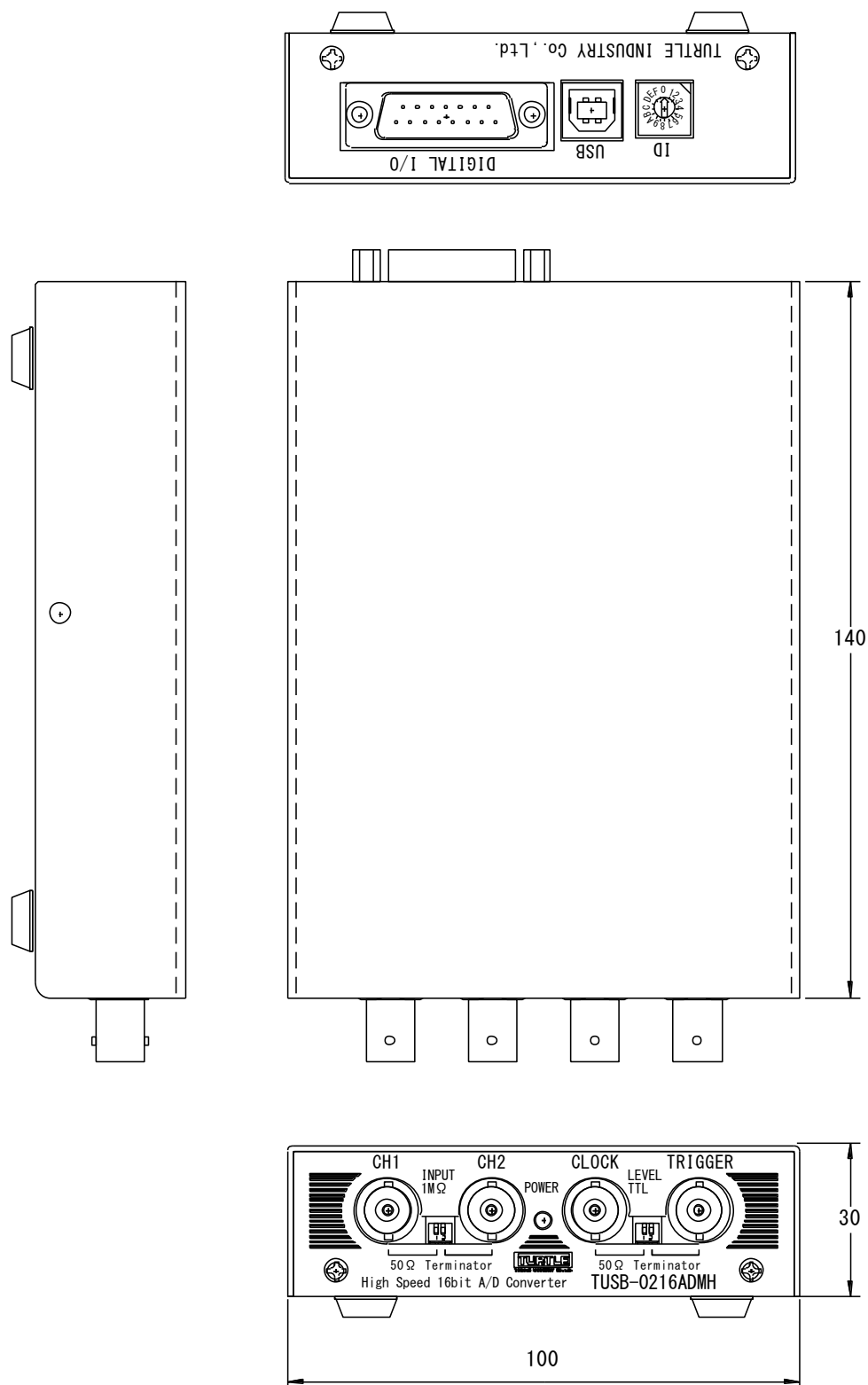
## 8. 仕様

### 8.1 仕様概要

入力チャンネル数	2ch(BNC コネクタ)
最高変換周波数	25MHz(2ch 同時)
変換分解能	1/65536(16 ビット)
高調波歪	約-95dB
ノイズレベル	約 3LSB(実効値)
入力抵抗	1M $\Omega$ /50 $\Omega$ 切り替え
入力電圧	-1～+1V/0～2V 切り替え
許容入力電圧	約±5V、(入力抵抗 50 $\Omega$ の場合は短時間のみ)
アナログ帯域幅	DC～約 10MHz(-3dB)
変換ビットパターン	ストレートオフセットバイナリ
メモリ容量	16bit 2M ワード(1M ワード×2ch FIFO 方式リングメモリ)
連続変換時の最高変換レート	200kSPS(1ch のみ転送時) [ パソコンの種類、使用状況によります ] 100kSPS(2ch 同時サンプリング時)
変換開始/停止繰り返し機能	1 回の最大データ量：1048576 ワード、繰り返し回数：制限なし
トリガ方式(プレトリガ機能)	コンピュータコマンド/入力信号±レベル/ 外部パルス入力(TTL レベル、BNC コネクタ)
プレトリガデータ量	1048576 ワード(最大)
クロック	内部搭載 1MHz～25MHz 外部(BNC コネクタ)1MHz～25MHz(デューティ比 50%±5%以内)
パラレル I/O	入力 4bit / 出力 4bit (各々TTL レベル)
電源電圧	DC5V(USB より供給) 350mA(最大)
使用温度範囲	5℃～45℃
大きさ	本体 30(H)×100(W)×140(D)mm(突起部含まず)
質量	約 300g(ケーブル等含まず)



## 8.2 寸法図



**TUSB-0216ADMH 取扱説明書**

発行年月      2025 年 5 月    第 8c 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

c2025   株式会社   タートル工業