

TUSB-INSxx

USB インタフェース付き絶縁型パラレル I/O ユニット

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。

| | |
|---|--|
|  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。万が一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。

©2019 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..... | 4 |
| 1.1 製品概要..... | 4 |
| 1.2 製品構成..... | 4 |
| 2. 各部の名称..... | 5 |
| 2.1 フロントパネル..... | 5 |
| 2.2 リアパネル..... | 5 |
| 3. 各部説明..... | 6 |
| 3.1 入出力コネクタ..... | 6 |
| 3.2 電源入力コネクタ..... | 7 |
| 3.3 USB コネクタ..... | 7 |
| 4. 機能解説..... | 8 |
| 4.1 絶縁入力について..... | 8 |
| 4.2 絶縁出力について..... | 9 |
| 5 付属のディスクについて..... | 10 |
| 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..... | 10 |
| 6.プログラミング..... | 11 |
| 6.1 Visual C++での使用..... | 11 |
| 6.1.1 使用準備..... | 11 |
| 6.1.2 関数の呼び出し方法(TUSB-INS8/8 の例)..... | 11 |
| 6.2 Visual Basic での使用..... | 12 |
| 6.2.1 使用準備..... | 12 |
| 6.2.2 プロシージャの呼び出し方法(TUSB-INS8/8 の例)..... | 12 |
| 6.3 Visual C# での使用..... | 12 |
| 6.3.1 使用準備..... | 12 |
| 6.3.2 関数の呼び出し方法(TUSB-INS8/8 の例)..... | 13 |
| 6.4 TUSB-INS8/8 関数説明..... | 14 |
| 6.5 TUSB-INS16I 関数説明..... | 20 |
| 6.6 TUSB-INS16O 関数説明..... | 25 |
| 7. その他..... | 28 |
| 7.1 USB について..... | 28 |
| 7.2 連絡先..... | 29 |
| 8. 仕様..... | 30 |
| 8.1 仕様概要..... | 30 |

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絶縁型パラレル I/O ユニット TUSB-INS 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)を使用した光絶縁型デジタル I/O ユニットです。フォトカプラを内蔵し、計測制御において電位差のあるシステム間やノイズが多い環境での使用など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を利用できますので、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。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

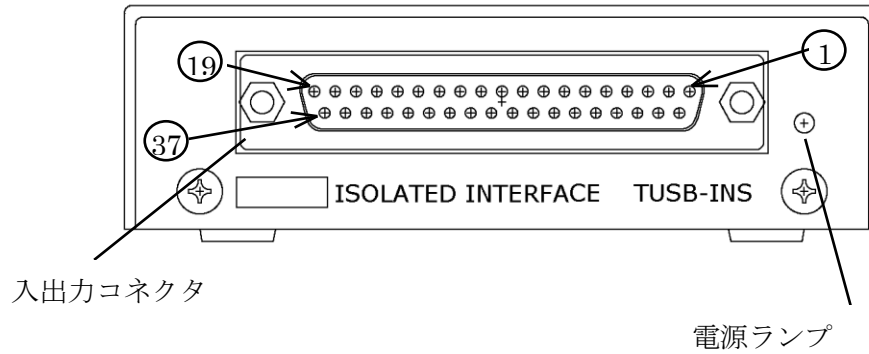
本製品には以下の物が含まれます。

- ① TUSB-INSxx 本体(xx は 8/8、16I、16O のいずれか)
- 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。

2. 各部の名称

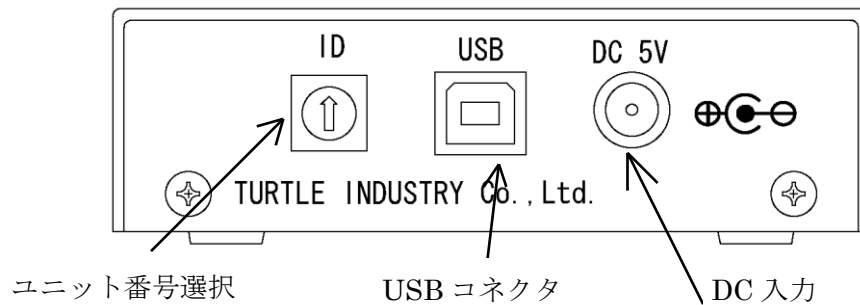
2.1 フロントパネル



| | |
|---------|--|
| 入出力コネクタ | フラットケーブル用 37 ピンコネクタ(ロック付き)です。デジタル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. 各部説明

3.1 入出力コネクタ

絶縁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: 17JE23370-02 又は同等品(かん合ネジはインチ)

| ピン番号 | INS8/8 | INS16I | INS16O |
|------|----------|-----------|-----------|
| 1 | BI0 アノード | BI0 アノード | BO0 コレクタ |
| 20 | BI0 カソード | BI0 カソード | BO0 エミッタ |
| 2 | BI1 アノード | BI1 アノード | BO1 コレクタ |
| 21 | BI1 カソード | BI1 カソード | BO1 エミッタ |
| 3 | BI2 アノード | BI2 アノード | BO2 コレクタ |
| 22 | BI2 カソード | BI2 カソード | BO2 エミッタ |
| 4 | BI3 アノード | BI3 アノード | BO3 コレクタ |
| 23 | BI3 カソード | BI3 カソード | BO3 エミッタ |
| 5 | BI4 アノード | BI4 アノード | BO4 コレクタ |
| 24 | BI4 カソード | BI4 カソード | BO4 エミッタ |
| 6 | BI5 アノード | BI5 アノード | BO5 コレクタ |
| 25 | BI5 カソード | BI5 カソード | BO5 エミッタ |
| 7 | BI6 アノード | BI6 アノード | BO6 コレクタ |
| 26 | BI6 カソード | BI6 カソード | BO6 エミッタ |
| 8 | BI7 アノード | BI7 アノード | BO7 コレクタ |
| 27 | BI7 カソード | BI7 カソード | BO7 エミッタ |
| 9 | BO0 コレクタ | BI8 アノード | BO8 コレクタ |
| 28 | BO0 エミッタ | BI8 カソード | BO8 エミッタ |
| 10 | BO1 コレクタ | BI9 アノード | BO9 コレクタ |
| 29 | BO1 エミッタ | BI9 カソード | BO9 エミッタ |
| 11 | BO2 コレクタ | BI10 アノード | BO10 コレクタ |
| 30 | BO2 エミッタ | BI10 カソード | BO10 エミッタ |
| 12 | BO3 コレクタ | BI11 アノード | BO11 コレクタ |
| 31 | BO3 エミッタ | BI11 カソード | BO11 エミッタ |
| 13 | BO4 コレクタ | BI12 アノード | BO12 コレクタ |
| 32 | BO4 エミッタ | BI12 カソード | BO12 エミッタ |
| 14 | BO5 コレクタ | BI13 アノード | BO13 コレクタ |
| 33 | BO5 エミッタ | BI13 カソード | BO13 エミッタ |
| 15 | BO6 コレクタ | BI14 アノード | BO14 コレクタ |
| 34 | BO6 エミッタ | BI14 カソード | BO14 エミッタ |
| 16 | BO7 コレクタ | BI15 アノード | BO15 コレクタ |
| 35 | BO7 エミッタ | BI15 カソード | BO15 エミッタ |
| 17 | 非接続 | 非接続 | 非接続 |
| 36 | 非接続 | 非接続 | 非接続 |
| 18 | 非接続 | 非接続 | 非接続 |
| 37 | 非接続 | 非接続 | 非接続 |
| 19 | 非接続 | 非接続 | 非接続 |

- ※ 入力端子に接続されている LED には直列に 1K Ω の抵抗器が接続されています。入力 ON 時の電流が 4~15mA の範囲になるように外部で入力抵抗を調整してください。
- ※ 入力端子の直流逆電圧の最大定格は 5V です。逆電圧がかかる場合には外部にダイオード等を入れて逆電圧による破壊から装置を保護してください。
- ※ 出力端子は各ビット電氣的に独立しておりますが、各ビット間の最大印可電圧はコレクタの最大印可電圧と同様に 55V です。

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 μ 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ります。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供給電流を超えない様にシステムを構築しなければなりません。

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下さい。

- ※ 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要です。

4. 機能解説

4.1 絶縁入力について

[入力電流]

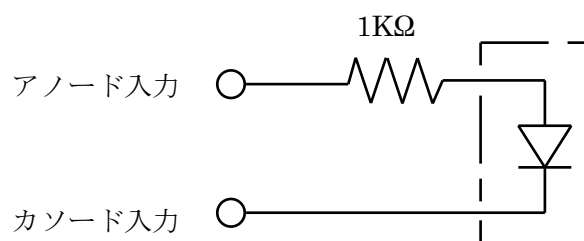
入力電流は 4mA 以上流してください。入力電流は最大でも 15mA までとしてください。

[入力抵抗]

装置内には 1K Ω の入力抵抗が入っております。駆動側の電圧により必用に応じて直列抵抗を付加してください。

[逆電圧]

入力への逆電圧は最大定格 5V となっております。逆電圧がかかる場合にはダイオードなどにより入力部を逆電圧保護してください。



入力回路構成

4.2 絶縁出力について

[出力電流]

出力電流の最大定格は **150mA** です。

ただし、出力電流が **15mA** を超える場合は出力電流と出力電圧の積が **100mW** を超えない範囲となる事を確認してください。

[出力飽和電圧]

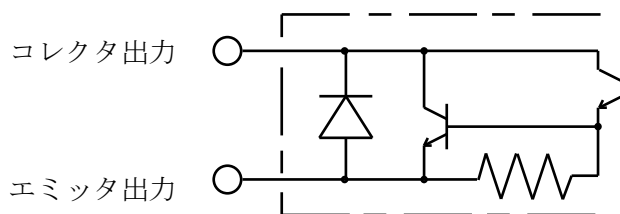
出力電流が **40mA** の時 **1V(Typ)**です。(ダーリントン接続)

[コレクタ・エミッタ間電圧]

コレクタ・エミッタ間電圧の絶対最大定格は **55V** です。

[エミッタ・コレクタ間電圧]

エミッタ・コレクタ間電圧の絶対最大定格は **0.3V** です。交流回路の使用時には回路構成に注意してください。



出力回路構成

[電源投入後の出力状態について]

電源投入後の初期状態では出力は全て **OFF** です。

5 ソフトウェアについて

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

|-[TUSBINSx] (注 1)

|-[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。

※ ソフトウェアディスクは付属しません。ソフトウェアは弊社 Web サイトよりダウンロードしてください。

(注 1)

フォルダ名 TUSBINSx は使用するユニットの種類によって異なります。

TUSB-INS16I TUSBINSI

TUSB-INS16O TUSBINSO

TUSB-INS8/8 TUSBINS8

6.プログラミング

ここでは、Visual C++、Visual Basic、Visual C# で TUSB-INSxx 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、ここではソフトウェアセット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。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++での使用

6.1.1 使用準備

Visual C++で使用するために以下 2つのファイルをソフトウェアセット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。(製品によりファイル名が異なります)

| | |
|------------|----------------------|
| TUSBINS8.H | TUSB-INS8/8 用ヘッダファイル |
| TUSBINSI.H | TUSB-INS8/8 用ヘッダファイル |
| TUSBINSO.H | TUSB-INS8/8 用ヘッダファイル |

※ ソフトウェアセット内の 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 TUSBINS8.h(又は TUSBINSI.h、TUSBINSO.h)をインクルードし、TUSBINS8.lib(又は TUSBINSI.lib、TUSBINSO.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(TUSB-INS8/8 の例)

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。Tusbins88_Device_Open 以外の関数は Tusbins88_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 `Tusbins88_Device_Open` を実行して機能関数の使用が終了したら `Tusbins88_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。1つの TUSB-INS8/8 デバイスを 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。1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进行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つのファイルをソフトウェアセット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。

TUSBINS8.vb TUSB-INS8/8 用標準ライブラリファイル

TUSBINSI.vb TUSB-INS8/8 用標準ライブラリファイル

TUSBINSO.vb TUSB-INS8/8 用標準ライブラリファイル

※ ソフトウェアセット内の DEVTOOLS フォルダにあります。

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(TUSB-INS8/8 の例)

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。`Tusbins88_Device_Open` 以外のプロシージャは `Tusbins88_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 `Tusbins88_Device_Open` を実行してプロシージャの使用が終了したら `Tusbins88_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。

1つの TUSB-INS8/8 デバイスを 2つのアプリケーションで同時にオープン(操作)する事は出来ません。1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进行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つのファイルをソフトウェアセット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。

| | |
|-------------|------------------------|
| TUSBINS8.cs | TUSB-INS8/8 用ライブラリファイル |
| TUSBINSI.cs | TUSB-INS8/8 用ライブラリファイル |
| TUSBINSO.cs | TUSB-INS8/8 用ライブラリファイル |

※ ソフトウェアセット内の DEVYTOOLS 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。

6.3.2 関数の呼び出し方法(TUSB-INS8/8 の例)

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。
Tusbins88_Device_Open 以外の関数は Tusbins88_Device_Open 関数が正常に処理された後でないとは有効にはなりません。各機能関数を呼び出す前に Tusbins88_Device_Open を実行して機能関数の使用が終了したら Tusbins88_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。1つの TUSB-INS8/8 デバイスを 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。1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进行操作する事はできません。

6.4 TUSB-INS8/8 関数説明

Tusbins88_Device_Open

| | |
|------|--|
| C 宣言 | short Tusbins88_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

- 0:成功
- 1:ID 番号が不正
- 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

Tusbins88_Device_Close

| | |
|------|--|
| C 宣言 | void Tusbins88_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

Tusbins88_Write

| | |
|------|---|
| C 宣言 | short Tusbins88_Write (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

絶縁出力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)**を指定すれば全て出力 **ON**、**00(HEX)**を指定すれば全て **OFF** となります。

引数

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

戻り値

- 0:成功
- 1:オープンされていない
- 2:失敗

Tusbins88_Read

| | |
|------|---|
| C 宣言 | short Tusbins88_Read (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

絶縁入力の入力データを読みみます。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。例えば引数 `dat` が `FF(HEX)`であれば入力が全て ON、`00(HEX)`であれば全て OFF となります。

引数

| | |
|------------------|-----------------------|
| <code>id</code> | ユニット番号選択スイッチの番号(0-15) |
| <code>dat</code> | 入力したデータを格納するバッファのポインタ |

戻り値

0:成功
1:オープンされていない
2:失敗

Tusbins88_EdgeSet

| | |
|------|--|
| C 宣言 | short Tusbins88_EdgeSet (short id, unsigned char dat, unsigned char time) |
| | 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

エッジ検出機能の動作設定を行います。エッジ検出機能は指定時間間隔毎にポートの入力状態をチェックして前値との比較を行います。していされた変化が発生した時には内部エッジ検出用レジスタに 1 をセットします。

引数

| | |
|------|---|
| id | ユニット番号選択スイッチの番号(0-F) |
| dat | エッジの検出方向(ビットパターン) 各ビット 0 の時は OFF→ON の時にエッジ検出します。 各ビット 1 の時は ON→OFF の時にエッジ検出します。 |
| time | エッジの検出間隔を設定します。0~255 設定値+1 (mS) |

戻り値

0:成功

1:オープンされていない

2:失敗

Tusbins88_EdgeRead

| | |
|------|---|
| C 宣言 | short Tusbins88_EdgeRead (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

エッジ検出用レジスタの値を読みみます。ビットパターンで参照変数 **dat** に入力されて戻されます。前回エッジ読み込み後に検出されたエッジのビットが 1 になっています。この関数の実行時にはエッジ検出用レジスタはリセットされます。

引数

| | |
|-----|-----------------------|
| id | ユニット番号選択スイッチの番号(0-15) |
| dat | 入力したデータを格納するバッファのポインタ |

戻り値

- 0:成功
- 1:オープンされていない
- 2:失敗

6.5 TUSB-INS16I 関数説明

Tusbins16i_Device_Open

| | |
|------|---|
| C 宣言 | short Tusbins16i_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

- 0:成功
- 1:ID 番号が不正
- 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

Tusbins16i_Device_Close

| | |
|------|---|
| C 宣言 | void Tusbins16i_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

Tusbins16i_Read

| | |
|------|--|
| C 宣言 | short Tusbins16i_Read (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

デジタル入出力の入力データを読み込みます(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 ON、`0000(HEX)` であれば全て OFF となります。

引数

| | |
|------------------|-----------------------|
| <code>id</code> | ユニット番号選択スイッチの番号(0-15) |
| <code>dat</code> | 入力したデータを格納するバッファのポインタ |

戻り値

- 0:成功
- 1:オープンされていない
- 2:失敗

Tusbins16i_EdgeSet

| | |
|------|--|
| C 宣言 | short Tusbins16i_EdgeSet (short id, int dat, unsigned char time) |
| | 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

エッジ検出機能の動作設定を行います。エッジ検出機能は指定時間間隔毎にポートの入力状態をチェックして前値との比較を行います。していされた変化が発生した時には内部エッジ検出用レジスタに 1 をセットします。

引数

| | |
|------|---|
| id | ユニット番号選択スイッチの番号(0-F) |
| dat | エッジの検出方向(ビットパターン) 各ビット 0 の時は OFF→ON の時にエッジ検出します。 各ビット 1 の時は ON→OFF の時にエッジ検出します。 |
| time | エッジの検出間隔を設定します。0~255 設定値+1 (mS) |

戻り値

- 0:成功
- 1:オープンされていない
- 2:失敗

Tusbins16i_EdgeRead

| | |
|------|--|
| C 宣言 | short Tusbins16i_EdgeRead (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

エッジ検出用レジスタの値を読みみます。ビットパターンで参照変数 **dat** に入力されて戻されます。前回エッジ読み込み後に検出されたエッジのビットが 1 になっています。この関数の実行時にはエッジ検出用レジスタはリセットされます。

引数

| | |
|-----|-----------------------|
| id | ユニット番号選択スイッチの番号(0-15) |
| dat | 入力したデータを格納するバッファのポインタ |

戻り値

- 0:成功
- 1:オープンされていない
- 2:失敗

6.6 TUSB-INS16O 関数説明

Tusbins16o_Device_Open

| | |
|------|---|
| C 宣言 | short Tusbins16o_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

- 0:成功
- 1:ID 番号が不正
- 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

Tusbins16o_Device_Close

| | |
|------|---|
| C 宣言 | void Tusbins16o_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

Tusbins16o_Write

| | |
|------|---|
| C 宣言 | short Tusbins16o_Write (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

絶縁出力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 **FFFF(HEX)**を指定すれば全て出力 ON、**0000(HEX)**を指定すれば全て OFF となります。

引数

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

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 仕様概要

仕様概要

絶縁入力部

| | |
|---------|-------------------|
| ビット数 | 8bit(TUSB-INS8/8) |
| ターンオン時間 | 50 μ 秒 |
| ターンオフ時間 | 15 μ 秒 |
| 入力オン電流 | 4mAmin |
| 入力最大定格 | 15mA |

絶縁出力部

| | |
|----------------|-------------------|
| ビット数 | 8bit(TUSB-INS8/8) |
| ターンオン時間 | 50 μ 秒 |
| ターンオフ時間 | 15 μ 秒 |
| 出力シンク電流 | 15mAmin |
| コレクタ・エミッタ間最大定格 | 55V |
| コレクタ・エミッタ間最大定格 | 0.3V(逆電圧) |

その他

| | |
|-----------|---|
| 信号入出力コネクタ | 37 D-sub コネクタ(メス) |
| 入出力間耐圧 | AC1000V / 1 分間 |
| 入出力間絶縁抵抗 | 100M Ω / DC500V |
| 電源 | USB または外部安定化 DC |
| 消費電流 | 約 100mA |
| 外部電源電圧 | 5.0VDC 安定化されたもの |
| 大きさ | 30(h) \times 100(w) \times 140(d)mm(突起物含みません) |
| 重さ | 約 300g |

TUSB-INSxx 取扱説明書

(64bitOS 用ドライバ対応版)

発行年月 2019年4月 第16a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

©2019 株式会社 タートル工業