

USB-PIO UNIT

- ・USB-PIO-8I (絶縁型)
- ・USB-PI-16I (絶縁型)
- ・USB-PO-16I (絶縁型)

(3機種共通)取扱説明書

USB I/F Insulation type parallel (8bit/16bit) I/O UNIT



システムサコム工業株式会社

このマニュアルは<http://www.sacom.co.jp>からダウンロードできます。
予告なく仕様を変更することがございますのでご了承下さい。詳細は、お問い合わせ下さい。

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。

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。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。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、システムサコム工業株式会社までご連絡下さい。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。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。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。

©2000 SYSTEM SAOM Co., Ltd. All rights reserved.

システムサコム工業株式会社の許可なく、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

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	1
1.1	製品概要	
1.2	製品構成	
2	各部の名称	2
2.1	フロントパネル	
2.2	リアパネル	
3	各端子説明	3
3.1	入出力コネクタ	
3.2	電源入力コネクタ	
3.3	USBコネクタ	
4	インストールの方法	5
4.1	サンプルソフトのインストールと使い方	
5	機能説明	6
5.1	絶縁入力について	
5.2	絶縁出力について	
6	ドライバソフトウェアの使用	8
6.1	開発環境の設定	
6.2	基本的な関数使用の流れ	
7	ドライバ関数リファレンス	10
7.1	USB-PIO-8Iのドライバ関数	
7.2	USB-PI-16Iのドライバ関数	
7.3	USB-PO-16Iのドライバ関数	
8	その他	24
8.1	うまく動作しないとき	
8.2	USBについて	
8.3	連絡先	
9	仕様	27

1 はじめに

この度は、システムサコム工業株式会社製のUSBインターフェース付き絶縁型パラレルI/OユニットUSB-P (IO-8I、I-16I、O-16I)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いたします。

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。

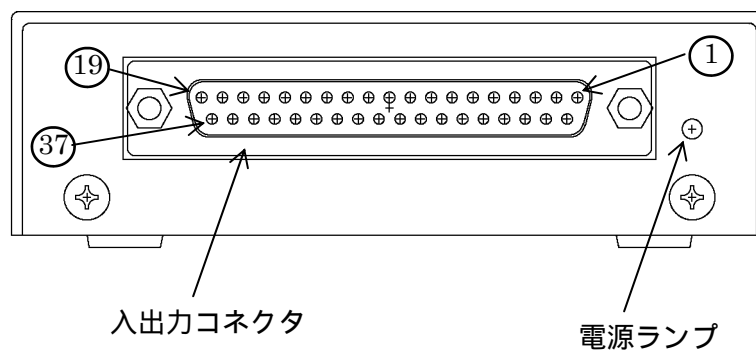
1.2 製品構成

本製品には以下の物が含まれます。

- USB-P (IO-8I、I-16I、O-16Iのいずれか) 本体
- USBケーブル(1m)
- 取扱説明書(本書)
- インストールマニュアル説明書
- 添付ソフトウェア(CD-ROMディスク 1枚)
- USBドライバソフト、サンプルソフト(VisualC++6.0,VisualBasic6.0)

2 各部の名称

2.1 フロントパネル

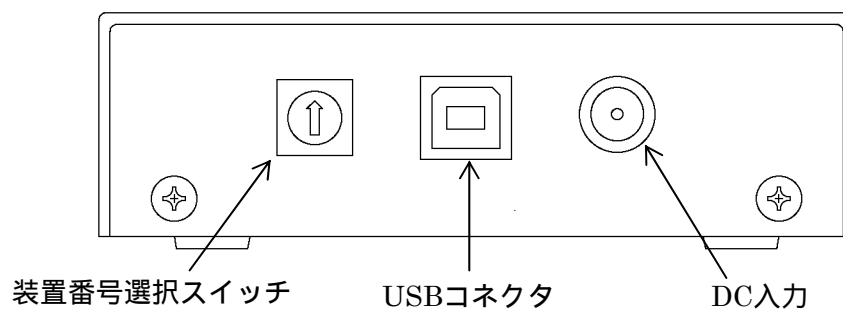


入出力コネクタ : フラットケーブル用37ピンコネクタ(ロック付き)です。
ADコンバータ、DAコンバータ、デジタル入出力の入出力をここから行います。

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

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	PIO-8I	PI-16I	PO-16I
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5～25mA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ることになります。、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

デバイスのインストールは、別冊のインストールマニュアルをご参照ください。

4.1 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CD-ROMディスク>
|[16O] (又は[16I]、[88IO])
|-[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ください。

5 機能説明

5.1 絶縁入力について

入力電流

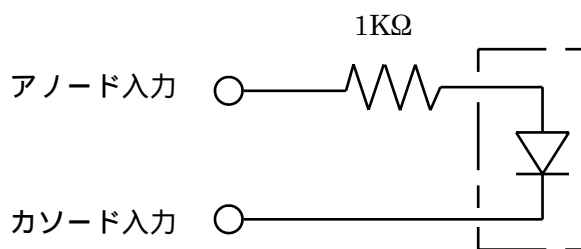
入力電流は5mA以上流してください。入力電流は最大でも15mAまでとしてください。

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

5.2 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。

- a) 150mA以内とする事。
- b) 15mAを超える場合は、電流と出力電圧の積が0.1Wを超えない事。

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)

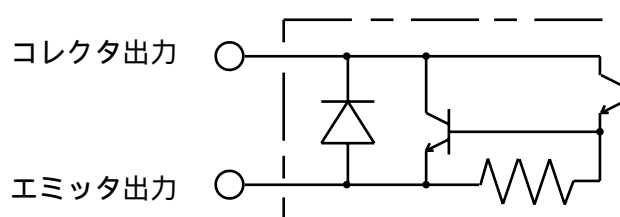
コレクタ・エミッタ間電圧

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



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CD-ROMディスクより(TOOLSディレクトリの中)
SUSBINSI.LIB^{1 2}
SUSBINSI.H^{1 2}
を適当な場所にコピーします。
- 2 SUSBINSI.LIB^{1 2}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SUSBINSI.H^{1 2}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CD-ROMディスクより(TOOLSディレクトリの中)
SUSBINSI.BAS^{1 2}
を適当な場所にコピーします。
- 2 SUSBINSI.BAS^{1 2}ファイルをプロジェクトに追加します。

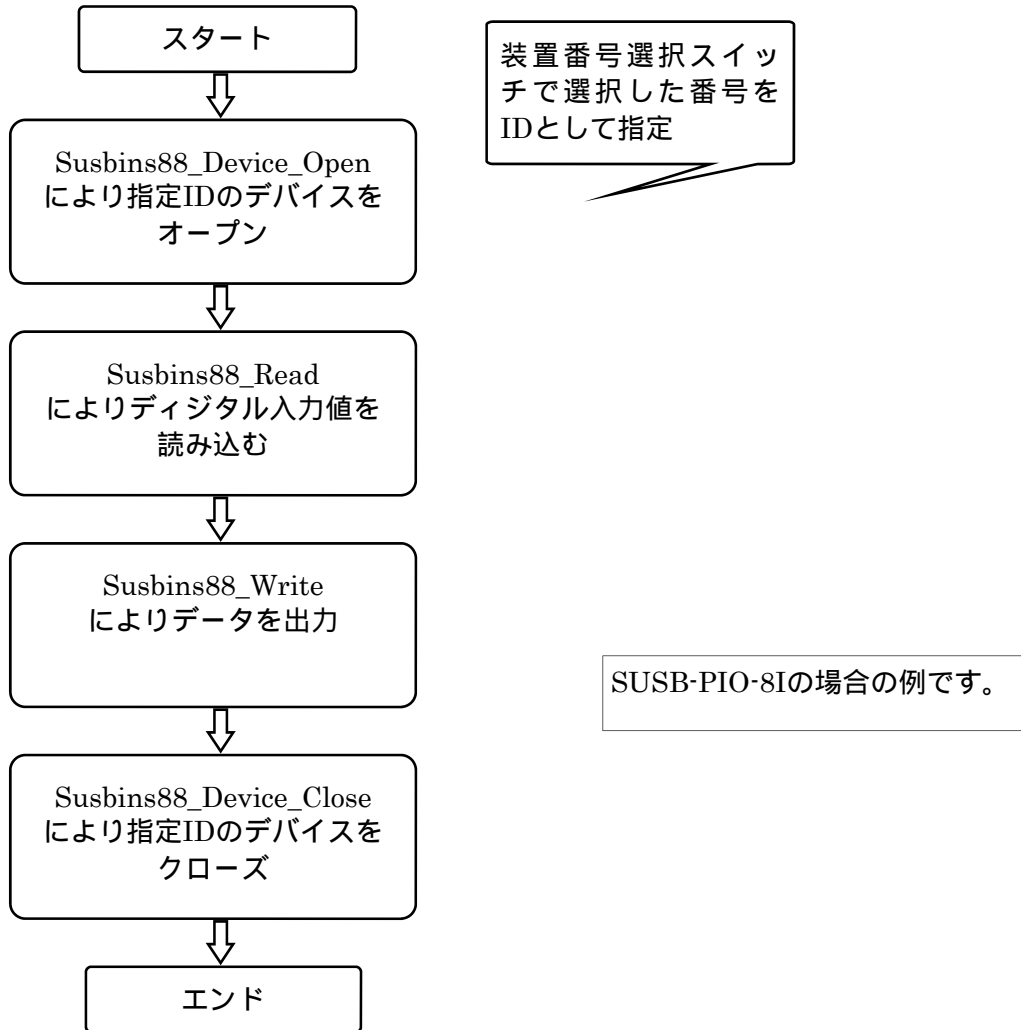
設定は以上です。

本サンプルプログラムはVisual Basic 6.0で作成されました。

^{1 2} SUSB-PI-16Iの場合のファイル名です。
SUSB-PO-16Iの場合はそれぞれ SUSBINSO.LIB SUSBINSO.H SUSBINSO.BASとなります。
SUSB-PIO-8Iの場合はそれぞれ SUSBINS8.LIB SUSBINS8.H SUSBINS8.BASとなります。

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、入力ポートからデータを入力し、出力ポートからデータを出し、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

7.1 USB-PIO8/8Iのドライバ関数

Susbins88_Device_Open

C,C++宣言	short __stdcall Susbins88_Device_Open(short id)
---------	---

BASIC定義	[Private] Declare Function Susbins88_Device_Open Lib "SUSBINS8.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

Susbins88_Device_Close

C,C++宣言	<code>void __stdcall Susbins88_Device_Close(short id)</code>
---------	---

BASIC定義	<code>[Private] Declare Sub Susbins88_Device_Close Lib "SUSBINS8.DLL" (ByVal id As Integer)</code>
---------	--

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

Susbins88_Write

C,C++宣言	short __stdcall Susbins88_Write (short id, unsigned char dat);
---------	---

BASIC定義	[Private] Declare Function Susbins88_Write Lib "SUSBINS8.DLL" (ByVal id As Integer , ByVal dat As Byte) As Integer
---------	--

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

Susbins88_Read

C,C++宣言	short __stdcall Susbins88_Read (short id, unsigned char *dat);
---------	---

BASIC定義	[Private] Declare Function Susbins88_Read Lib "SUSBINS8.DLL" (ByVal id As Integer , ByRef dat As Byte) As Integer
---------	---

解説

絶縁入力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。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

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

Susbins88_EdgeSet

C,C++宣言 `short __stdcall Susbins88_EdgeSet
 (short id, unsigned char dat,
 unsigned char time);`

BASIC定義 `[Private] Declare Function Susbins88_EdgeSet
 Lib "SUSBINS8.DLL"
 (ByVal id As Integer , ByVal dat As Byte,
 ByVal time As Byte) As Integer`

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から15)
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

Susbins88_EdgeRead

C,C++宣言	short __stdcall Susbins88_EdgeRead (short id, unsigned char *dat);
---------	---

BASIC定義	[Private] Declare Function Susbins88_EdgeRead Lib "SUSBINS8.DLL" (ByVal id As Integer , ByRef dat As Byte) As Integer
---------	---

解説

エッジ検出用レジスタの値を読み込みます。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から15)
dat	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

7.2 USB-PI16Iのドライバ関数

Susbins16i_Device_Open

C,C++宣言	short __stdcall Susbins16i_Device_Open(short id)
---------	--

BASIC定義	[Private] Declare Function Susbins16i_Device_Open Lib "SUSBINSI.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

Susbins16i_Device_Close

C,C++宣言	void __stdcall Susbins16i_Device_Close(short id)
---------	---

BASIC定義	[Private] Declare Sub Susbins16i_Device_Close Lib "SUSBINSI.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

Susbins16i_Read

C,C++宣言	short __stdcall Susbins16i_Read (short id, int *dat);
---------	--

BASIC定義	[Private] Declare Function Susbins16i_Read Lib "SUSBINSI.DLL" (ByVal id As Integer , ByRef dat As Long) As Integer
---------	--

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

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

Susbins16i_EdgeSet

C,C++宣言	short __stdcall Susbins16i_EdgeSet (short id, int dat, unsigned char time);
---------	---

BASIC定義	[Private] Declare Function Susbins88_EdgeSet Lib "SUSBINSI.DLL" (ByVal id As Integer , ByVal dat As Long, ByVal time As Byte) As Integer
---------	--

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から15)
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

Susbins16i_EdgeRead

C,C++宣言 `short __stdcall Susbins16i_EdgeRead
 (short id, int *dat);`

BASIC定義 `[Private] Declare Function Susbins16i_EdgeRead
 Lib "SUSBINSI.DLL"
 (ByVal id As Integer , ByVal dat As Long)
 As Integer`

解説

エッジ検出用レジスタの値を読み込みます。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から15)
dat	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

7.3 USB-PO16Iのドライバ関数

Susbins16o_Device_Open			
C,C++宣言	short __stdcall Susbins16o_Device_Open(short id)		
BASIC定義	[Private] Declare Function Susbins16o_Device_Open Lib "SUSBINSO.DLL" (ByVal id As Integer) As Integer		
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">解説</p> <p>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。</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">引数</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; padding: 5px;">id</td> <td style="padding: 5px;">ユニット番号選択スイッチの番号(0から15)</td> </tr> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">戻り値</p> <p>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</p> </div>		id	ユニット番号選択スイッチの番号(0から15)
id	ユニット番号選択スイッチの番号(0から15)		

Susbins16o_Device_Close

C,C++宣言	<code>void __stdcall Susbins16o_Device_Close(short id)</code>
---------	--

BASIC定義	<code>[Private] Declare Sub Susbins16o_Device_Close Lib "SUSBINSO.DLL" (ByVal id As Integer)</code>
---------	---

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

Susbins16o_Write

C,C++宣言	short __stdcall Susbins16o_Write (short id, int dat);
---------	--

BASIC定義	[Private] Declare Function Susbins16o_Write Lib "SUSBINSO.DLL" (ByVal id As Integer , ByVal dat As Long) As Integer
---------	---

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

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社ご連絡先

システムサコム工業株式会社
～ ユーザーサポート係 ～

E-mail

info@sacom.co.jp

TEL・FAX

・03-5623-5933 ・03-3660-0891

郵送

〒103-0002
東京都中央区日本橋馬喰町1-6-6
吉野第二ビル 4F

9 仕様

絶縁入力部

ビット数	8bit(USB-PIO-8I) 16bit(USB-PI-16I) USB-PO-16Iは入力なし
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	5mA _{min}
入力最大定格	50mA

絶縁出力部

ビット数	8bit(USB-PIO-8I) 16bit(USB-PO-16I) USB-PI16Iは出力なし
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	15mA _{min}
コルタ・ミツ間最大定格	55V
コルタ・ミツ間最大定格	7V(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) × 100(w) × 137(d)mm(突起物含みません)
重さ	約400g

USB-P (IO-8I、 I-16I、 O-16I) **取扱説明書**

発行年月 2007年6月 第3版
発 行 システムサコム工業株式会社
編 集 システムサコム工業株式会社

© システムサコム工業株式会社