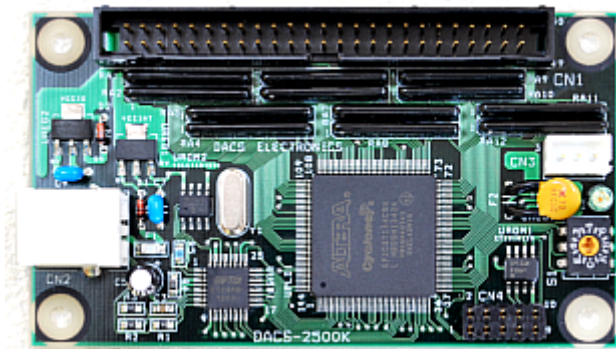


USB接続
デジタル入出力基板
DACS-2500K

取扱説明書



DACS

機器使用に関する注意と警告

- (1) 接続の間違い、または操作の誤りによって、万一、対象となる相手方装置、または本装置のいずれかが故障しても、本装置は一切の責任を負いません。
- (2) 本装置を接続することにより、対象機器の電気的な回路状態が変化する場合は、直ちに本装置の使用を中止してください。
- (3) 本装置から、対象機器となる装置に異常電圧等がかかり、相手方装置が故障した場合においても、本装置は、相手方装置に関する一切の責任を負いません。
- (4) 本装置を使用した機器の安全に関しては、お客様にて十分な対策を立ててください。本装置を使用した機器の異常動作によるトラブルに関しては、本装置は一切の責任を負いません。

目次

1. 機能	2
2. 構成	4
3. コネクタピン配置と入出力信号仕様	5
4. 送受信データ形式	
4. 1 デジタル出力コマンド	8
4. 2 AD/DA制御コマンド	10
(注) 本基板には、AD/DA変換機能はありません。	
4. 3 PWMパルス出力コマンド	12
4. 4 サンプリング間隔設定コマンド	14
4. 5 デジタル入力データ形式	15
4. 6 デジタル出力状態読取コマンドデータ形式	16
4. 7 デジタル出力状態データ形式	16
5. 回転ディップスイッチとランプの説明	17
6. 解説	
6. 1 接続	18
6. 2 ボードID番号のセット	18
6. 3 デバイスドライバのインストール	19
6. 4 もっともシンプルな使用方法	20
6. 5 複数台のボードを 仮想COMポートとして使用する	20
6. 6 ダイレクトドライバを使用して 応答速度を向上	21
6. 7 データサンプリングを高速に実行する	21
DACS-2500K製品内容	22

1. 機能

USB接続デジタル入出力ボード DACS-2500K (USB-DIOボード) は、パソコンのUSBポートに接続して、パソコンから送信するコマンドにより、デジタル入出力信号を制御するためのボードです。

非絶縁のデジタル入力24点、デジタル出力24点があり、コマンドによりこの出力信号を制御し、また入力信号を読取ることができます。

本説明書では DACS-2500K-**STD** 標準版 について説明しています。
説明中にDACS-2500K と省略している部分は、
すべて DACS-2500K-**STD** に関する説明です。

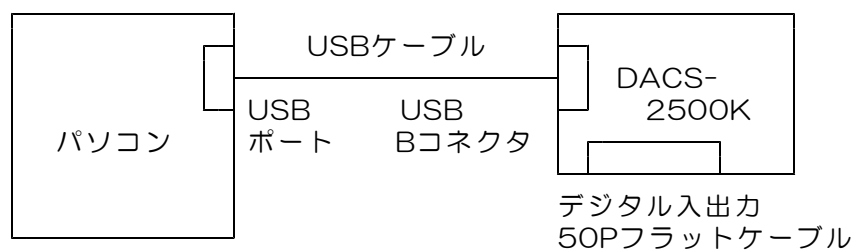
パソコン側からみると

このボードをUSBに接続すると、アプリケーションプログラムからは、高速版増設COMポートとして扱うことができます。たとえば、標準にてCOM1とCOM2をもっているパソコンでは、COM3がこのボードに対応する増設COMポートとなります。このボードを複数台接続すると、COM3、COM4、COM5 …… というように、COMポートが増えてゆきます。

また、ダイレクト版とよばれているデバイスドライバを使用すると、COMポートではなく、独自のUSBデバイスとして使用することができます。この場合は、基板と共に供給するドライバ独自の関数を用いて、基板とのREAD/WRITEを実行することになります。

READ/WRITEのデータ形式は

パソコンからは、たとえば W02A5B67☐ という簡単なアスキーコードの文字列を送信して、ボードのデジタル出力(24bit分)を設定し、ボードからはこの応答として、たとえば R01C4D58☐ というコードを返して、ボードのデジタル入力状態(24bit分)を通知します。



本ボードでは、FPGAとよばれる高密度集積回路を使用し、すべての動作を、ハードウェア論理回路にて並列に実行しています。このため、すべての入出力信号は、詳細仕様に記載しているタイミングにて、高速かつ正確に動作します。

主な機能

1	パソコンとの接続	<p>USBインターフェイス 高速拡張COMポートまたは専用USB機器として動作。 同時接続数 最大16 通信形式 アスキー文字列によるコマンド送信と アスキー文字列によるレスポンス受信。</p>
2	デジタル入力	<p>非絶縁 24bit TTLレベル 5V系およびLVTTTLいずれにも接続可能</p>
3	デジタル出力	<p>非絶縁 24bit TTLレベル TTL接続時最大負荷電流 2.5mA 2V5タイプ 出力電圧 最大 2.5V 短絡電流 12mA 3V3タイプ 出力電圧 最大 3.3V 短絡電流 20mA</p>
4	動作モード	<p>(a) デジタル入出力モード 各出力を指定通りにON/OFFし、 各入力状態を読取ります。 (b) AD/DA制御モード AD/DA変換器を接続して制御を行います。 (c) PWMパルス出力モード 市販のラジオコントロール用サーボモータを 動作させるためのPWMパルス、2ch分を 出力します。</p> <p>(a), (b), (c) のモードを複合して使用することが できます。</p>
5	動作速度 (目安)	<p>仮想COMドライバ使用時 最大繰返し周波数 50Hz ダイレクトドライバ使用時 最大繰返し周波数 1KHz 計測用サンプリング最大周波数 10KHz (注) 詳細は、6項の解説を参照ください。</p>
6	電源	<p>パソコンからUSBケーブルにて供給しますので、 基板用の別電源は不要です。消費電流 40mA この数値は、デジタル出力の負荷電流がない場合です。 デジタル出力に負荷電流が流れる場合は、 その電流値分が電源電流として増加します。</p>
7	動作周囲温度	0～50℃

2. 構成

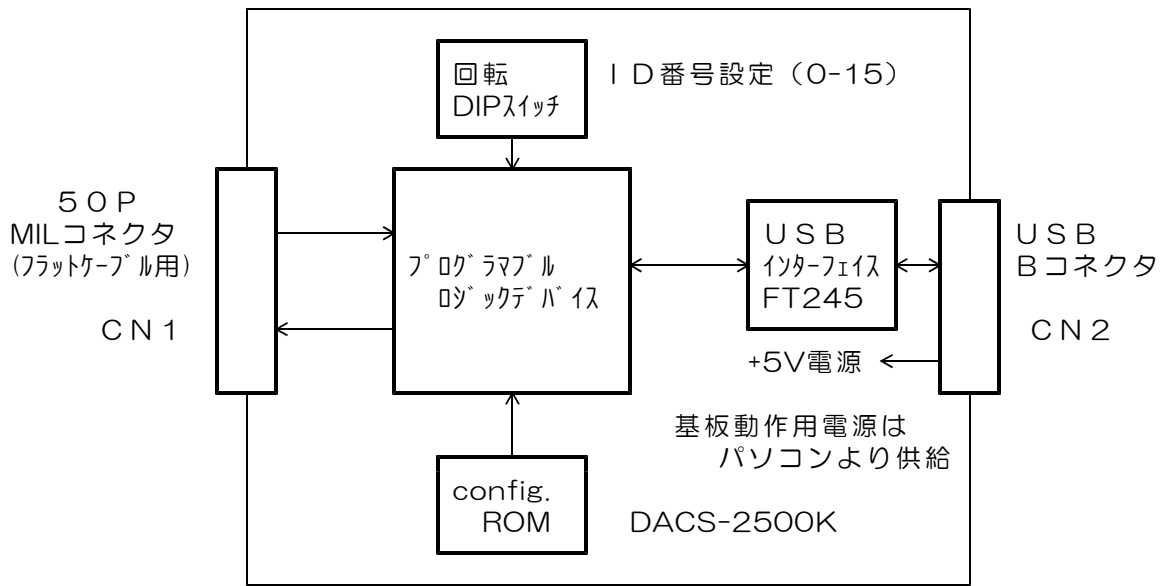


図2.1 DACS-2500K ブロック図

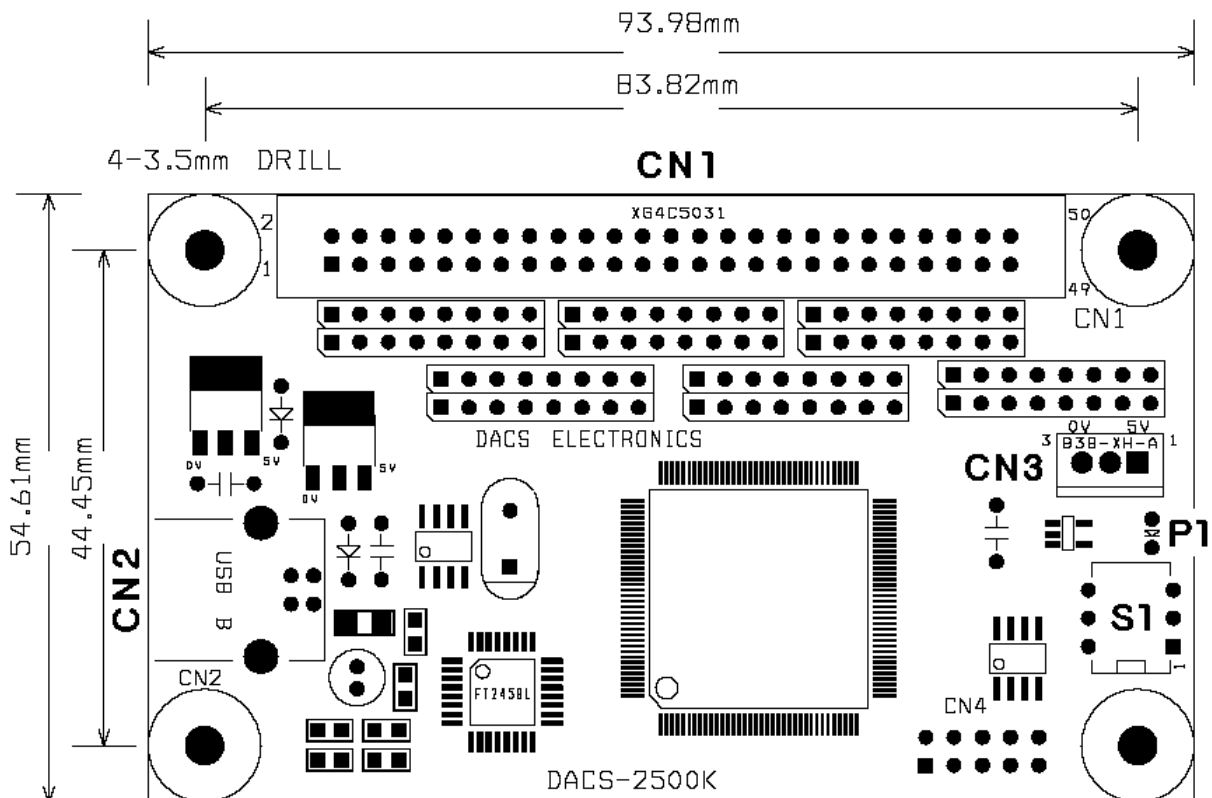


図2.2 DACS-2500K 外形図

3. コネクタピン配置と入出力信号仕様

C N 1 デジタル入出力コネクタ (50Pフラットケーブル用)

基板側 型式 オムロン XG4C5031

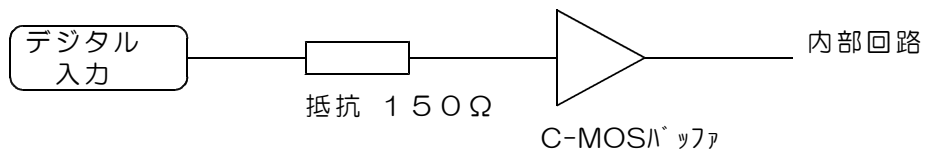
ケーブル側 型式 オムロン XG4M5030

(注) ケーブル側コネクタは、30cmケーブル付きにて標準添付となっています。
添付ケーブルの機器側は、解放端(コネクタなし)となっています。

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49
△																								

1	デジタル入力	bit 0 (LSB)	2	デジタル入力	bit 1
3	デジタル入力	bit 2	4	デジタル入力	bit 3
5	デジタル入力	bit 4	6	デジタル入力	bit 5
7	デジタル入力	bit 6	8	デジタル入力	bit 7
9	デジタル入力	bit 8	10	デジタル入力	bit 9
11	デジタル入力	bit 10	12	デジタル入力	bit 11
13	デジタル入力	bit 12	14	デジタル入力	bit 13
15	デジタル入力	bit 14	16	デジタル入力	bit 15
17	デジタル入力	bit 16	18	デジタル入力	bit 17
19	デジタル入力	bit 18	20	デジタル入力	bit 19
21	デジタル入力	bit 20	22	デジタル入力	bit 21
23	デジタル入力	bit 22	24	デジタル入力	bit 23 (MSB)
25	0V		26	0V	
27	デジタル出力	bit 0 (LSB)	28	デジタル出力	bit 1
29	デジタル出力	bit 2	30	デジタル出力	bit 3
31	デジタル出力	bit 4	32	デジタル出力	bit 5
33	デジタル出力	bit 6	34	デジタル出力	bit 7
35	デジタル出力	bit 8	36	デジタル出力	bit 9
37	デジタル出力	bit 10	38	デジタル出力	bit 11
39	デジタル出力	bit 12	40	デジタル出力	bit 13
41	デジタル出力	bit 14	42	デジタル出力	bit 15
43	デジタル出力	bit 16	44	デジタル出力	bit 17
45	デジタル出力	bit 18	46	デジタル出力	bit 19
47	デジタル出力	bit 20	48	デジタル出力	bit 21
49	デジタル出力	bit 22	50	デジタル出力	bit 23 (MSB)

デジタル入力回路



入力電圧範囲 0～+5V

入力リーク電流 入力 low 0.2mA以下 (入力端子から接続機器方向へ)
入力 High 10μA以下

しきい値 TTLレベル

High Level 最小値 +1.7V

Low Level 最大値 +0.7V

High Level : 論理1 Low Level : 論理0

(注意) 入力解放状態では、High/Lowのいずれになるかは不定です。
入力解放状態で入力をプログラムにて読み取ると、読取るごとに0と1とが変換することがあり、あたかもボードが不安定な動作をしているようにみえてしまいます。

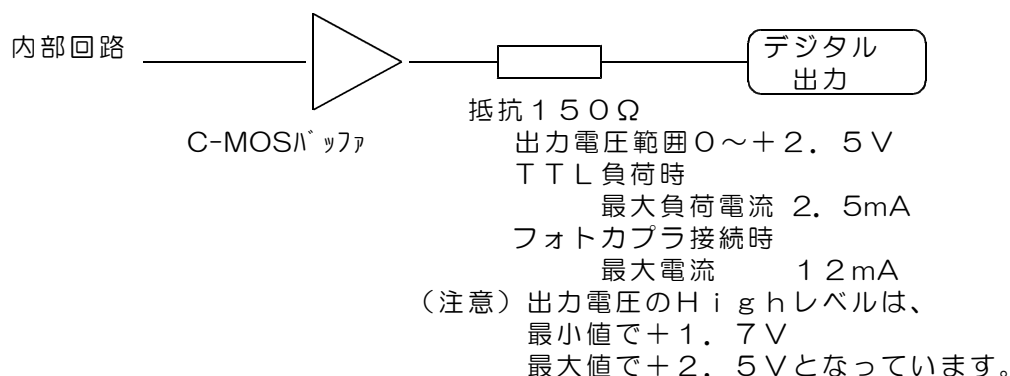
入力の動作試験を行うときは、

入力0とするためには、0～10KΩの抵抗にて、0Vに接続してください。

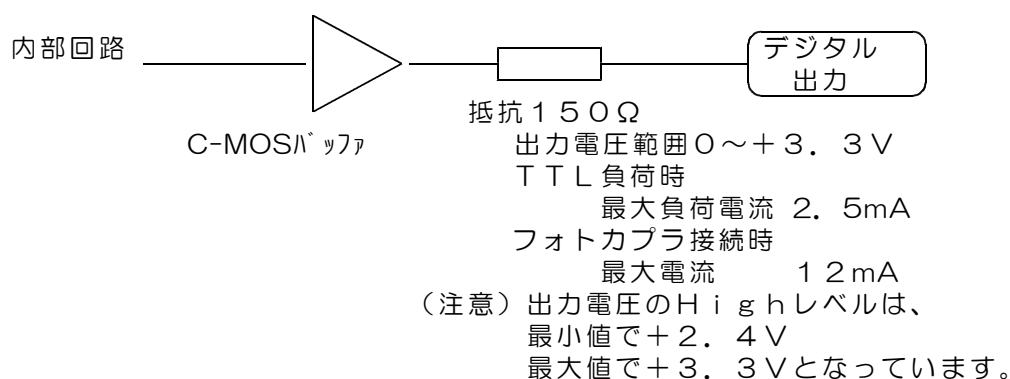
入力1とするためには、10KΩ程度の抵抗にて、+2V～+5Vの電源に接続してください。

(警告) 入力電圧範囲を超える電圧または負電圧を入力すると、ボードに使用してあるプログラムロジックデバイスが壊れます。該当する入力回路部分だけでなく、デバイス全体の機能が壊れます。

デジタル出力回路 TTL 2.5V出力 DACS-2500K- *** -2V5



デジタル出力回路 TTL 3.3V出力 DACS-2500K- *** -3V3



C N 2 USBコネクタ (Bタイプ)

(注) USBケーブルは、別途に準備ください。

- 1 +5 V 電源入力 (消費電流 40 mA デジタル出力負荷電流 0 のとき)
- 2 USB データ (-)
- 3 USB データ (+)
- 4 0 V

C N 3 電源出力コネクタ (3P アダプタ基板への電源供給用)

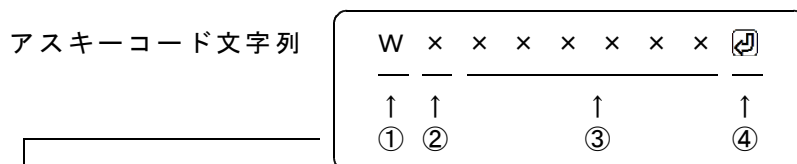
- 1 +5 V 電源出力 (最大出力電流 200 mA)
- 2 2 V5 タイプ のとき
 +2.5 V 電源出力 (最大出力電流 +5 V との合計値で 200 mA)
- 3 V3 タイプ のとき
 +3.3 V 電源出力 (最大出力電流 +5 V との合計値で 200 mA)
- 3 0 V

C N 4 出荷時にのみ使用するコネクタです。

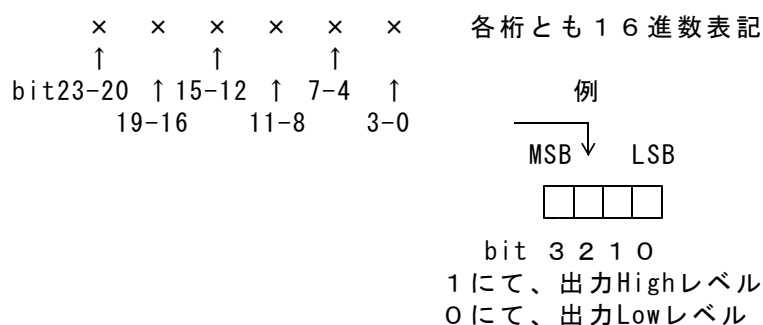
4. 送受信データ形式

4. 1 デジタル出力コマンド

(1) データ形式



- ① W (大文字) デジタル出力コマンド識別文字コード
- ② 0~9, A~F 基板識別IDコード (16進数文字表記 英字は小文字も可)
基板のディップスイッチ設定と同一とすること。
- ③ 000000~FFFFFF 16進数6桁表記 (英字は小文字も可)
デジタル出力する内容を指定。



- 16進数に該当しない文字を指定した場合。
その位置のデジタル出力は、直前に送信したコマンドの同一位置のデータとなります。
これを、4bit単位の Don't Care として利用することができます。
データの例 W1X12XXX␣
(注意) 直前のコマンドとは異なる種類のコマンドを送信する場合に、Don't Care を利用すると、出力が不正になります。

データの省略

- ③項のデータのすべて、あるいはその途中からを省略することができます。省略した場合は、上記のDon't Care と同じ扱いになります。
データの例 W1␣ W1A8␣

- ④ 区切りマーク
アスキー OD (H) キャリッジリターンコード または & 文字コード
キャリッジリターン、または&文字のうちのいずれかを指定します。
通常はキャリッジリターンコードを使用してください。
使用上の区別については、6項の解説を参照ください。

(2) 動作

DACS-2500K基板は、基板識別IDコードが一致するWコマンドを受信すると、直ちにデータ内容に従ってデジタル出力を実行します。この出力は、次のコマンドを受信するまで変化しません。

(参考) 電源投入時には、すべてのデジタル出力がLowになっています。

このコマンドの受信を完了した時点で、入力データをラッチし、デジタル入力データをホストに返します。レスポンスのデータ形式は、デジタル入力データ形式に記述しています。

応答は不要な場合でも必ずホスト側で読取ってください。

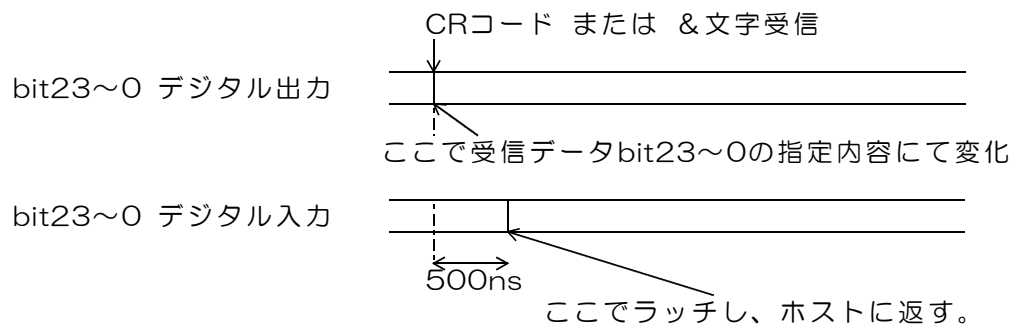
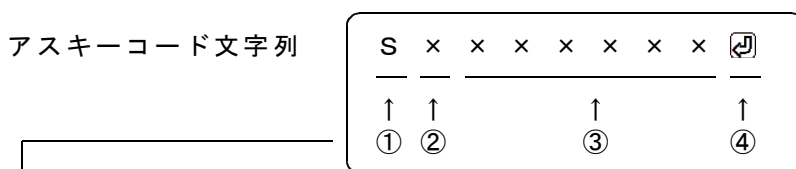


図4. 1 デジタル出力コマンド受信時の動作

4. 2 AD/DA制御コマンド

(注) 本基板にはAD/DA変換機能はありません。

(1) データ形式



- ① S (大文字) AD/DA制御コマンド識別文字コード
- ② 0~9, A~F 基板識別IDコード (16進数文字表記 英字は小文字も可)
基板のディップスイッチ設定と同一とすること。
- ③ 000000~FFFFFF 16進数6桁表記 (英字は小文字も可)
出力する内容を指定
左端より bit23~20 右端が bit3~0

bit23 DAデータラッチ用ストロープコントロール
1: ストロープ有効 0: ストロープなし

bit22 AD変換用RCコントロール
1: 遅延あり 0: 遅延なし

bit21~12
AD変換用マルチプレクサ制御信号
アンプゲイン制御信号など、
AD変換後に出力を変化させる必要のあるもの

bit11~0
汎用出力データ

- 16進数に該当しない文字を指定した場合。
その位置のデジタル出力は、直前に送信したコマンドの同一位置のデータとなります。
これを、4bit単位の Don't Care として利用することができます。
データの例 S1X12XXX␣
(注意) 直前のコマンドとは異なる種類のコマンドを送信する場合に、
Don't Care を利用すると、出力が不正になります。

データの省略

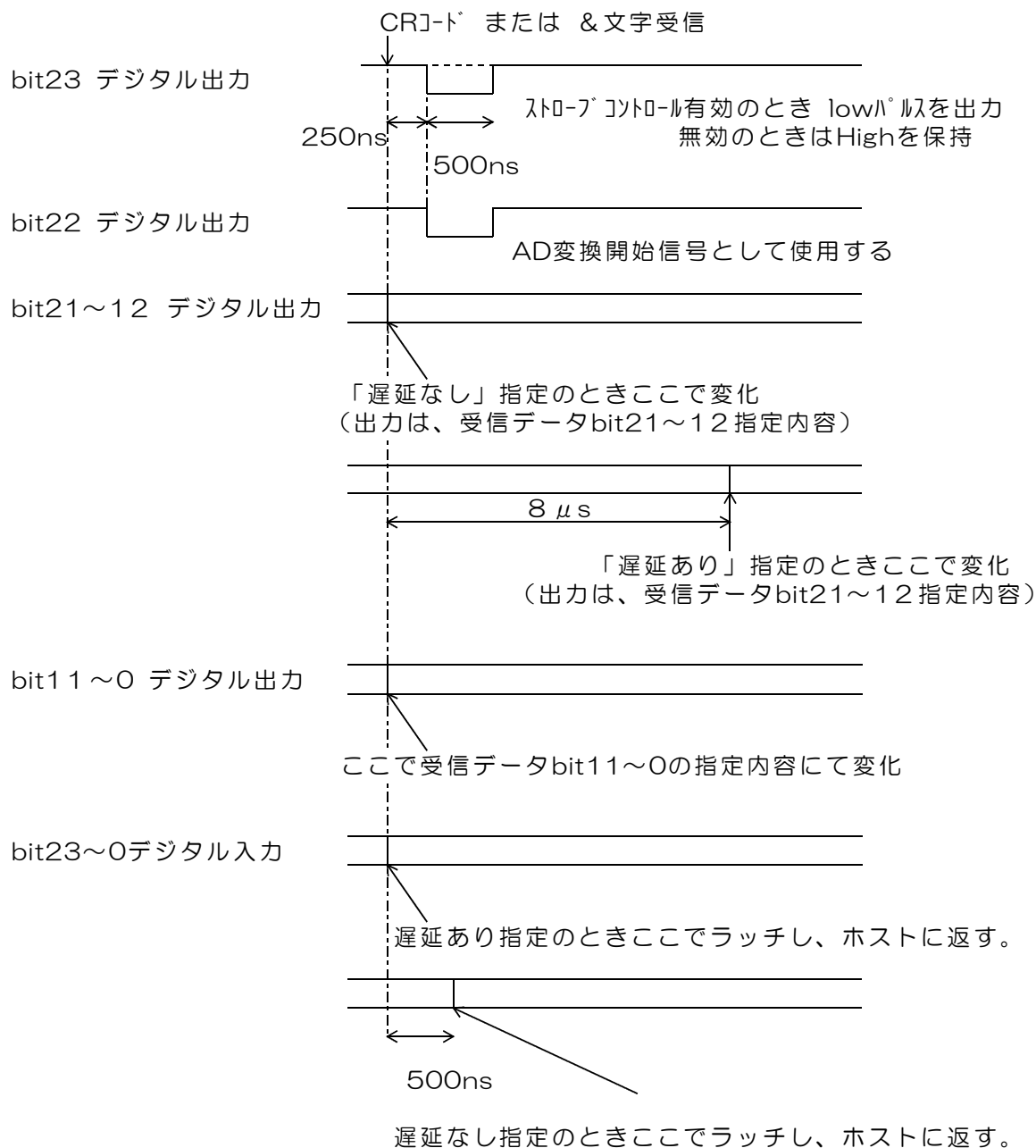
③項のデータのすべて、あるいはその途中からを省略することができます。省略した場合は、上記のDon't Care と同じ扱いになります。

データの例 S1␣ S1A8␣

- ④ 区切りマーク
アスキー OD (H) キャリッジリターンコード または & 文字コード
キャリッジリターン、または&文字のうちのいずれかを指定します。
通常はキャリッジリターンコードを使用してください。
使用上の区別については、6項の解説を参照ください。

(2) 動作

DACS-2500K基板は、基板識別IDコードが一致するSコマンドを受信すると、次の手順にてデータを出力し、入力データをラッチします。
ラッチした入力データは、デジタル入力データ形式の項に記述する形式にて、レスポンスとしてホストに返します。
応答は不要な場合でも必ずホスト側で読取ってください。

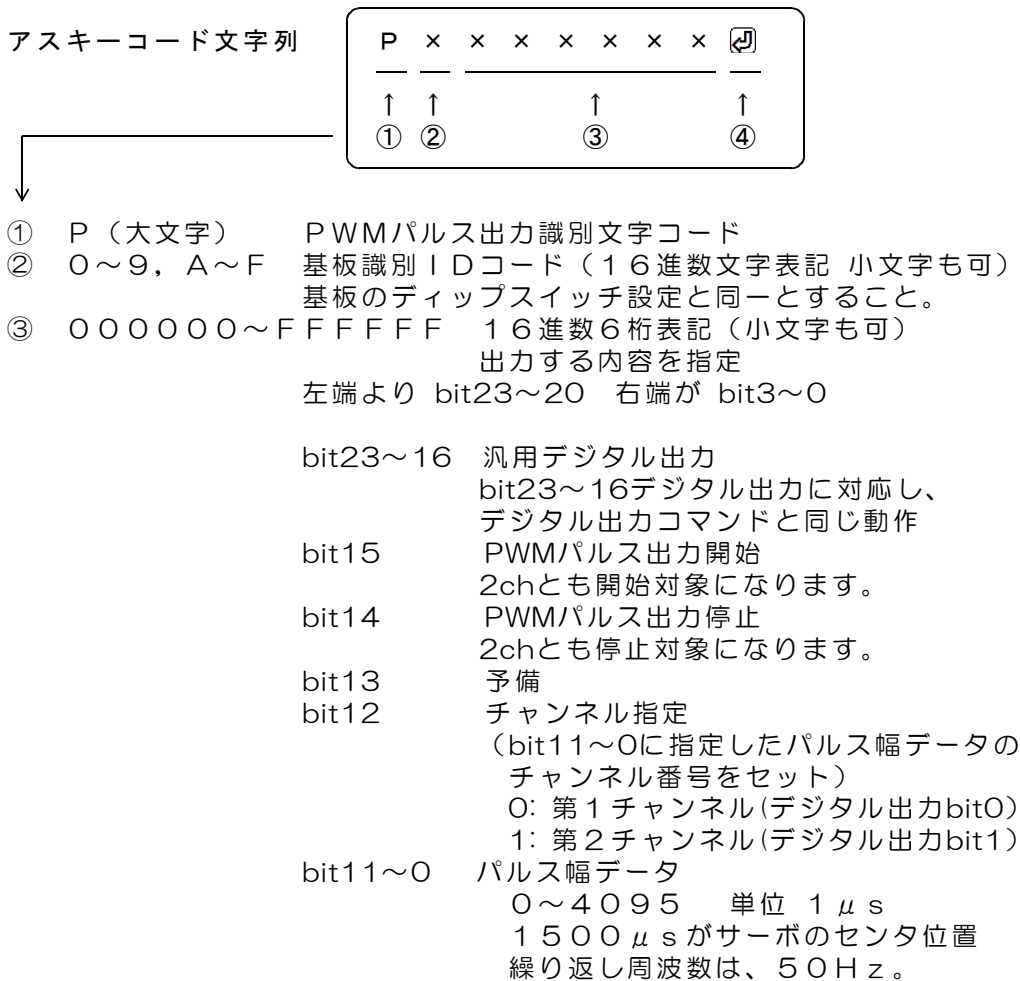


参考 遅延あり指定の動作は、主にAD変換器に接続して、マルチプレクサ（入力チャンネル）を切換ながら動作させる場合に使用します。
遅延なし指定の動作は、主にメモリなどの読取り書き込み動作に使用します。

図4.2 AD/DA制御コマンド受信時の動作

4. 3 PWMパルス出力コマンド

(1) データ形式



- 16進数に該当しない文字を指定した場合。
その位置のデジタル出力は、直前に送信したコマンドの同一位置の
データとなります。
これを、4bit単位の Don't Care として利用することができます。
(注意) 直前のコマンドとは異なる種類のコマンドを送信する場合に、
Don't Care を利用すると、出力が不正になります。

データの省略

③項のデータのすべて、あるいはその途中からを省略することができます。
省略した場合は、上記のDon't Care と同じ扱いになります。

- ④ 区切りマーク
アスキー OD (H) キャリッジリターンコード または & 文字コード
キャリッジリターン、または&文字のうちのいずれかを指定します。
通常はキャリッジリターンコードを使用してください。
使用上の区別については、6項の解説を参照ください。

(2) 動作

DACS-2500K基板は、基板識別IDコードが一致するPコマンドを受信すると、指定内容に従って、PWMパルス出力制御を実行し、入力データをラッチします。デジタル出力bit23~16は、デジタル出力コマンドと同じように、指定内容にて出力します。

デジタル出力bit15~2は、このコマンドを受信しても変化しません。

デジタル出力bit1~0は、PWMパルス出力開始指定にてパルス出力となり、PWMパルス出力停止指定にて、通常のデジタル出力動作（以前に受信したデジタル出力コマンドまたはAD/DA制御コマンドの指定内容を出力）となります。

ラッチした入力データは、デジタル入力データ形式の項に記述する形式にて、レスポンスとしてホストに返します。入力データのラッチタイミングは、デジタル出力コマンドの場合と同じです。

応答は不要な場合でも必ずホスト側で読取ってください。

PWMパルス出力	指定データ幅（単位 $1\mu\text{s}$ ）のパルスを、繰り返し周波数50Hzにて連続出力します。 1500 μs がサーボのセンタ位置 サーボの可動範囲があるため、 実質的には1000~2000の範囲が有効となります。
----------	--

4. 4 サンプルング間隔設定コマンド

(1) データ形式

アスキーコード文字列

I	x	x	x	x	x	x	x	↵
↑	↑			↑				↑
①	②			③				④

- ① I (大文字 アイ) サンプルング間隔設定コマンド識別文字コード
- ② 0~9, A~F 基板識別IDコード (16進数文字表記 小文字も可)
基板のディップスイッチ設定と同一とすること。
- ③ 000000~0FFFFFFF 16進数6桁表記 (小文字も可)

受信データを実行する間隔を指定。

単位 $1\mu\text{s}$ 設定範囲 $5 \sim 1,048,575\mu\text{s}$
正確な値を設定する場合の注意
実際の実行間隔は、ここに指定する間隔に、
(送信文字数+1) $\times 0.5\mu\text{s}$ が加算されます。

電源投入時には最小値になっています。

- ④ 区切りマーク
アスキー OD (H) キャリッジリターンコード または & 文字コード
キャリッジリターン、または&文字のうちのいずれかを指定します。
通常はキャリッジリターンコードを使用してください。
使用上の区別については、6項の解説を参照ください。

(2) 動作

DACS-2500K基板は、基板識別IDコードが一致する I コマンドを受信すると、データ内容に従って「受信データの実行間隔」を設定します。
実行間隔は、このコマンドを受信した直後から、その後に受信するコマンドすべてについて有効になります。

DACS-2500K基板は、受信バッファに蓄積しているデータを、この間隔にて順次実行してゆきます。

(参考) 電源投入時には、最小値の $5\mu\text{s}$ になっています。

実行間隔設定内容および利用方法の詳細については、6項の解説を参照ください。

また、その他のコマンドと同様に、入力データをラッチし、レスポンスとしてホストにデータを返します。入力データのラッチタイミングは、デジタル出力コマンドの場合と同じです。

応答は不要な場合でも必ずホスト側で読取ってください。

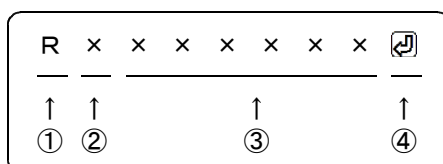
このコマンドにより、デジタル出力の変化はありません。

4. 5 デジタル入力データ形式

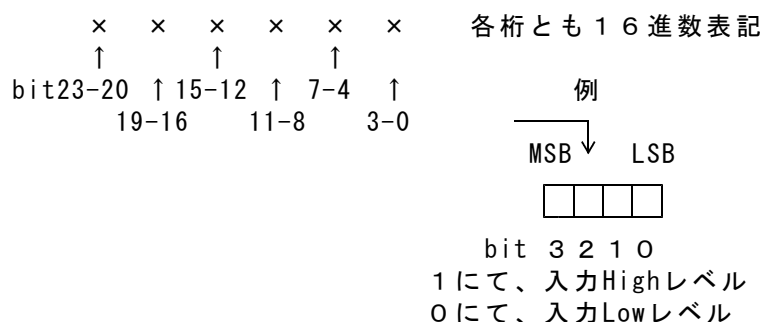
ご注意 本項にて説明するデジタル入力データ形式は、パソコンから送信するコマンドではありません。パソコンから送信する「Wコマンド」などに、DACS-2500Kが応答するデータ形式を説明しています。

(1) データ形式

アスキーコード文字列



- ① R (大文字) デジタル入力応答識別文字コード
- ② 0~9, A~F 基板識別IDコード (16進数文字表記 大文字)
基板のディップスイッチ設定により決まる。
- ③ 000000~FFFFFF 16進数6桁表記 (大文字)
デジタル入力内容。



対応するコマンドデータの省略があっても、応答内容には省略はなく、常に固定長です。

- ④ 区切りマーク アスキー OD (H) キャリッジリターンコード
または & 文字コード
対応するコマンドの末尾と同じコードを返します。

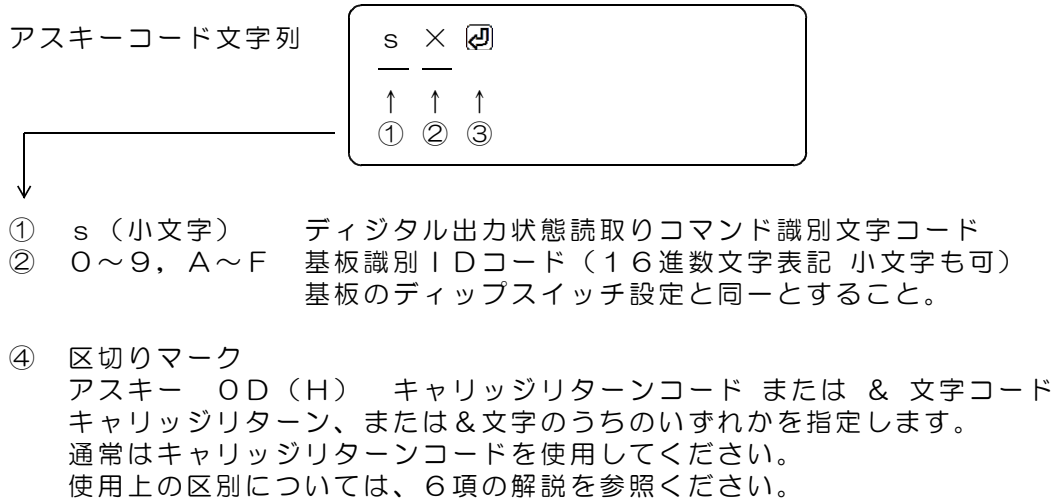
(2) 動作

DACS-2500K基板は、基板識別IDコードが一致する各コマンドを受信すると、デジタル入力信号をラッチし、レスポンスとして、本形式にて、データをホストに返します。

応答は不要な場合でも必ずホスト側で読取ってください。

4. 6. デジタル出力状態読取りコマンドデータ形式

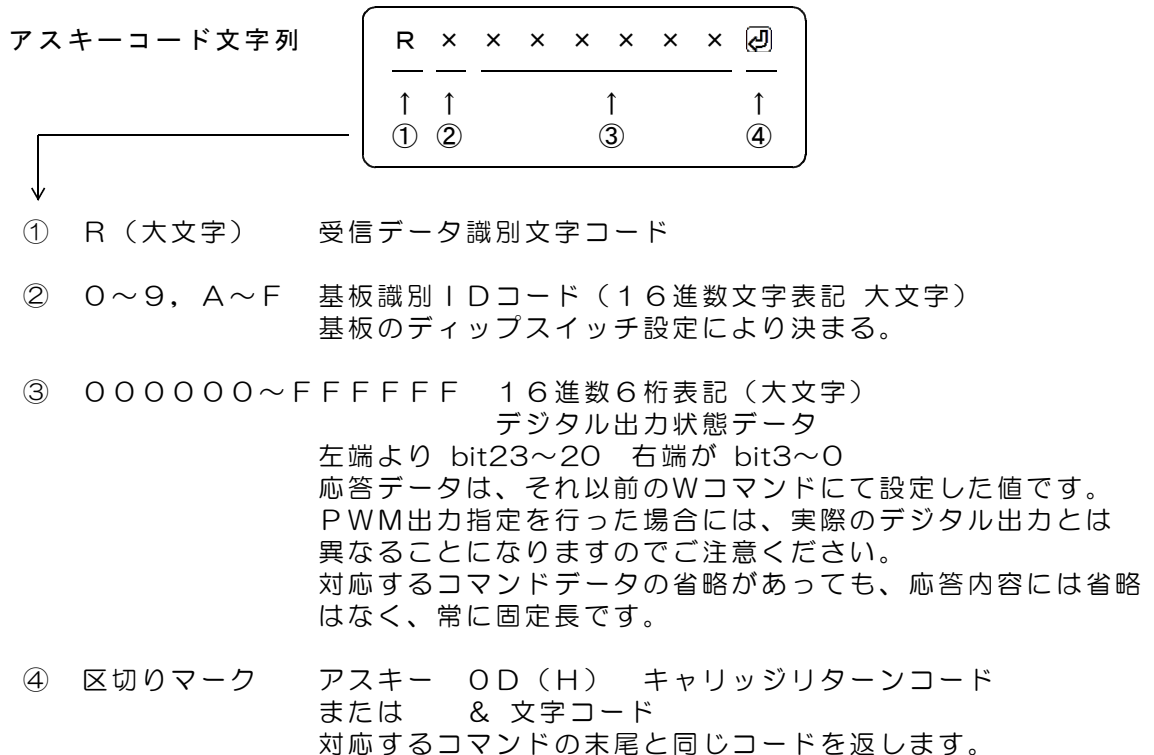
本項のデジタル出力状態読取りコマンドを送信すると、その応答としてDACS-2500Kが、4.7項のデータをホストに返します。



4. 7. デジタル出力状態データ形式

デジタル出力状態読取りコマンドの応答としてDACS-2500Kがホストに返します。

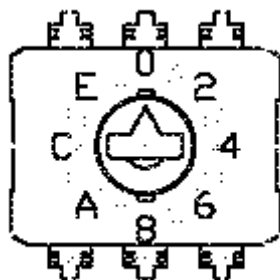
ご注意 本項にて説明するデジタル入力データ形式は、パソコンから送信するコマンドではありません。DACS-2500Kが応答するデータ形式を説明しています。



5. 回転ディップスイッチとランプの説明

(1) 回転ディップスイッチの設定

基板上にある回転ディップスイッチ S1 にて、ID番号を設定します。
ID番号の使用方法については、6項の解説を参照ください。



この図ではID=0の設定例

図5. 1 回転ディップスイッチの設定

(2) LEDランプの表示

デジタル出力の最上位ビット bit23 がON (1) となると、LEDランプP1が点灯します。

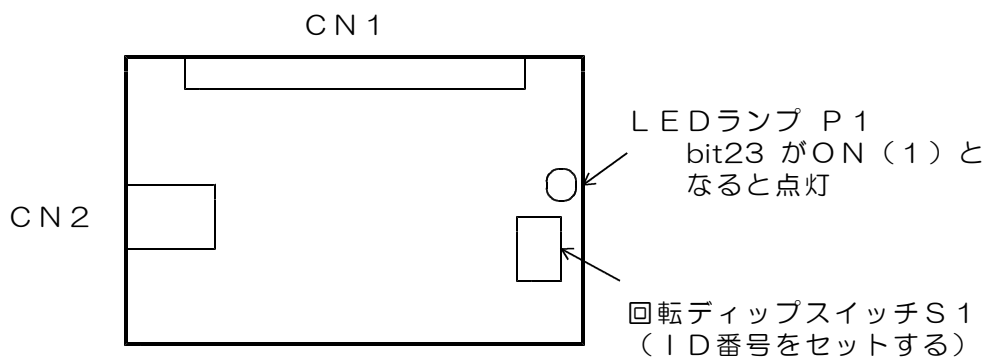


図5. 2 回転ディップスイッチとLEDランプの位置

6. 解説

6. 1 接続

USBケーブルにて、パソコンとUSB-DIO基板を接続します。ケーブルは別途ご購入ください。パソコン側がAタイプコネクタ、USB-DIO側がBコネクタのケーブルを使用します。ケーブルの最大長は5mです。

複数台のUSB-DIOを接続する場合で、パソコン側のUSBインターフェイスポートが不足するときは、汎用USBハブを使用して接続可能ポート数を増やしてください。

基板の電源は、パソコンからUSBケーブルを介して供給されますので、基板用に特別な電源を用意する必要はありませんが、5台以上の基板を使用する場合は、パソコンから供給する電源容量が不足しますので、USBハブに外部電源を接続して別電源を供給してください。USBハブによっては、接続する機器の条件にかかわらず、必ず外部電源を必要とするものもあります。（使用するUSBハブの説明書を参照してください。）

また、ノートパソコンをご使用の場合で、2台以上の基板を使用する場合は、必ずUSBハブに外部電源を供給して、基板を動作させるようにしてください。

6. 2 ボードID番号のセット

ボードID番号をすべて同一（たとえば0）の設定として複数台の基板を動作させたとしても、特に問題が発生するわけではありません。しかし、基板の故障などにて基板を交換するようなことがあると、複数台の基板を接続したシステムで、ID番号を利用したプログラムを採用していない場合は、プログラムの一部を変更しない限り、システムが正常に動作しなくなってしまう可能性があります。この理由は次の通りです。

一般のUSB接続機器と同様に、USB-DIOボードも、ベンダID、プロダクトID、ボードシリアルNo. という番号により区別されています。このうち、ベンダIDとプロダクトIDは、USB-DIOシリーズ固有のもので、すべての基板で同一のものとなっています。ボードシリアルNo.については、弊社より出荷するボードごとに個別に異なった番号が設定されています。このボードシリアルNo.は、基板上のEPROM内に書込んであり、書込プログラムツールを使用しない限り変更することはできません。

このシリアルNo.により、パソコン側のデバイスドライバが、それぞれのボードごとにデバイス番号を、0, 1, 2, 3, ... というように、0番からの連番で割り当てて動作するようになっています。（同じシリアルNo. では、デバイスドライバが、ボードを区別することができません。）

このため、複数の基板を利用しているシステムで、基板の故障などで、一部またはすべての基板を交換するようなことがあると、デバイスドライバが割り当てるデバイス番号の順番が変わってしまうことになり、アプリケーションプログラムからみたデバイス対応が、それまでのものと一致しなくなってしまう。

- | | |
|----------|---|
| 単独使用の場合 | ボードID番号は固定（たとえば0）にて使用してください。
仮想COMポートドライバを使用した場合、基板を交換すると、あらたなCOMポートが追加になります。
Windowsのデバイスマネージャーをひらいて、COMポートをもとの番号にもどせば、正常に動作するようになります。
ダイレクトドライバを使用した場合、基板交換があっても、デバイス番号は0番しかありませんので、そのまま動作します。 |
| 複数台使用の場合 | ボードID番号をそれぞれに割り当てて、ディップスイッチにて設定してください。 |

仮想COMポートドライバを使用した場合、基板を交換すると、あらたなCOMポートが追加になります。Windowsのデバイスマネージャーをひらいて、COMポートをもとの番号にもどせば、正常に動作するようになります。

しかし、すべてのCOMポート番号とボードとの対応が、きちんと保持できているかどうかを管理するのは、なかなか困難なので、複数台使用の場合は、仮想COMポートドライバではなく、ダイレクトドライバを使用することをお勧めします。

ダイレクトドライバを使用した場合、基板交換があるとデバイス番号の順番が変わります。このため、ボードID番号をもとにしたプログラムを作成するようにしてください。

具体的には、デバイス番号とボードIDとの対応を、システムの稼働時に検索する方法となります。

基板と共にご提供している「ダイレクトドライバを使用したサンプルプログラム」では、この方法を採用しています。

サンプルプログラムのソースファイルを参考にしてください。

6.3 デバイスドライバのインストール

デバイスドライバには、仮想COMポートドライバと、ダイレクトドライバの2種類があります。

(参考) 複合板ドライバを使用すれば、この2種類のドライバを同時にインストールすることができます。

対応OS Windows 10/8/7

仮想COMポートドライバ

このドライバをインストールすると、拡張COMポートが追加となります。

インストール後、WindowsのデバイスマネージャーにてCOMポートが増えていることと、増えたCOMポートの番号を確認してください。

アプリケーションプログラムからは、通常のシリアルポートと同様の扱いにて、プログラミングができます。

基板と共にご提供しているサンプルプログラムにより、インストール後の動作確認を行ってください。

ダイレクトドライバ

アプリケーションプログラムからは、ダイレクトドライバ専用の関数を使用してOPEN/READ/WRITE/CLOSEなどを実行します。

複数の基板を使用する場合、あるいは高速動作をさせる必要のある場合は、このダイレクトドライバを使用されることをお勧めします。

基板と共にご提供しているサンプルプログラムにより、インストール後の動作確認を行ってください。サンプルプログラムの動作については、サンプルプログラムと共にご提供する説明資料を参照ください。

ダイレクトドライバ専用関数の使用方法については、ドライバと共にご提供するPDFファイル(英文)とサンプルプログラムのソースファイルを参照してください。

インストール方法

インストール方法の詳細は、USB接続デバイスドライバインストール手順説明書を参照してください。

6. 4 もっともシンプルな使用方法

もっともシンプルな使用法は、1台のUSB-DIO基板を使用し、デバイスドライバとして仮想COMドライバを使用した場合です。

標準的なパソコンでは、USB-DIOは、デバイスドライバのインストールで、COM3に接続されます。動作試験は添付のサンプルプログラム（仮想COMシングル版）にて行います。

（注）パソコンによっては、COM3以外に接続される場合があります。

添付のサンプルプログラム（仮想COMシングル VS2015版）は、接続されているCOMポートを検索するようになっていまして、COM3以外に接続されている場合は、こちらのサンプルプログラムを使用してください。

サンプルプログラム起動後、キーボードから、たとえば W O 1 2 3 4 5 6 (⏎) と入力してみてください。ボードID番号が0に設定してあって、正常に接続できていれば、R O O O O O O O といった応答がかえってきます。
（受信データの最後には、キャリッジリターンコードがありますが、このコードは画面上では・となるか、全く表示されないかのいずれかになります。）

この使用方法では、パソコンからコマンドを送信し、その応答を待って、次のステップに進むという、コマンドとレスポンスの1対1対応のハンドシェイク方式となります。
コマンドを送出する繰り返し最小間隔は、およそ20ms となります。

この時間間隔は、次のような理由により決まります。

USBインターフェイスでは、64byte長のパケットを使用しています。

また、USB-DIOに使用しているUSBインターフェイスでは、送出するデータ長が64byte（ユーザデータは62byte）となるか、16msのタイムアウトとなるまで、このパケットを送りません。USB-DIOの送信データ長は9byteですので、USB-DIOは、毎回、16msのタイムアウトにてデータを送信します。

パソコンからのデータ送信にも、1~2msの時間がかかりますので、これらの合計時間として、繰り返し最小間隔は、およそ20ms となります。

6. 5 複数台のボードを仮想COMポートとして使用する

複数台のボードを、仮想COMポートとして使用することもできます。

標準的なパソコンでは、USB-DIOは、デバイスドライバのインストールで、COM3から順次、COM4、COM5 ー ー ー というように接続されてゆきます。

動作試験は、添付のサンプルプログラム（仮想COMマルチデバイス版）にて行います。サンプルプログラムでは、接続しているすべての USB-DIOボードにデータを送信し、その応答を待ちます。それぞれのボードに異なったID番号をセットしていれば、いずれかの1台がこれに回答してきます。サンプルプログラムでは、回答のあったCOMポートのデータをポート番号と共に表示します。

（注）仮想COMマルチデバイス版サンプルプログラム（VS2015版）が対応しているポート番号は、COM3~49のうち、最大4ポートです。

複数台のボードを使用する場合、仮想COMポートドライバを利用すると、COMポート番号とボードの対応を管理することが難しくなります。複数台のボードを使用する場合は、ダイレクトドライバを使用することをお勧めします。

6. 6 ダイレクトドライバを使用して応答速度を向上

(注) 動作速度に関する以下の説明は、ボードを1台のみ接続し、種々の条件を最良にした場合です。USBハブは使用していません。

ダイレクトドライバを使用することにより、6. 4項に記述しているタイムアウト時間を短縮することができます。

ダイレクトドライバでは、EventCharacter という特殊文字をUSB-DIOボードに送信して登録することができます。USB-DIOボードは、この文字が送信データ列にあると、タイムアウト時間を待たないで、直ちにデータをホスト（パソコン側）に送信します。添付のサンプルプログラム（ダイレクトドライバ マルチデバイス版）では、キャリッジリターンコードを、この EventCharacter とし、これにより、16msのタイムアウト時間を解除しています。サンプルプログラムでは、デバイスのOPENを行っている直後に、このEventCharacter設定関数を呼び出しています。サンプルプログラムのソースファイルを参照してください。

一方、パソコン側からUSB-DIOボードにデータを転送する間隔については、パソコンのUSBスケジューラのポーリングサイクルが1msとなっているために、パソコンからコマンドを送出する間隔を、このポーリング時間以下にすることができません。

アプリケーションプログラムで、データ受信（Read）から、次のデータ送信準備（Write）までを、1msよりも十分に短い時間で実行できるとすれば、コマンドとレスポンスのハンドシェイクを、最短時間の1msにて、繰り返して行うことができます。

機器制御のような用途で、データ出力とデータ入力を繰り返すような場合、この1msの時間間隔が最短の繰り返し時間となります。

6. 7 データサンプリングを高速に実行する

USB-DIOボードには、受信バッファ（FIFO BUFFER）として 128byte、送信バッファ（FIFO BUFFER）として 384byte があります。データ計測のような用途では、このバッファを利用して、最高10KHzにてデータサンプリングを実行することができます。

- (1) 均一なサンプリング間隔を確保するために、I（アイ）コマンドを用いて、サンプリング間隔（コマンド実行間隔）を設定します。

例 I 0 0 0 0 0 6 2 ☞ <--- 末尾は 0D(H)
送信文字数3とした場合の 100 μs 設定例

- (2) 次のような複数のコマンドからなる文字列を、WRITE関数の呼出しにて、USB-DIOに送信します。

S 0 & S 0 & S 0 & S 0 & S 0 & S 0 & S 0 & S 0 & S 0 & S 0 & S 0 ☞

Sコマンド12個を、省略形式にて&でつなぎます。末尾は 0D(H) とします。
&は、キャリッジリターンと同様に、各コマンドの区切りとなります。
唯一、キャリッジリターンと異なるのは、&で区切っているコマンドに対しては、レスポンスデータの末尾も&となるため、6. 6項にて説明しているEventCharacter とならないことです。
USB-DIOでは、Sコマンドを、0.1msの間隔にて12回繰り返し、送信バッファにたまったデータ列が、62byteのデータとなるか、またはキャリッジリターンのあったところで、レスポンスデータとしてホストに送信します。

さらに、このコマンド列を、あと2回、合計で3回、WRITE関数にて送信します。
 この回数は、受信バッファを有効に利用するためのものです。
 USB-DIOの受信バッファは128byteの容量ですので、USB-DIOの受信するデータが、 $36 \times 3 = 108$ byte となって、これが限界となります。4回送信するとバッファがいっぱいになり、WRITE関数を呼び出しても、バッファに空きができるまで、戻ってこなくなってしまう。

(3) 受信バッファに、USB-DIOボードからのレスポンス12個分 ($9 \times 12 = 108$ byte) が蓄積されるのを待って、READ関数で108byteを読取ります。
 USB-DIOの送信バッファは、384byte ありますので、3回分のデータ ($9 \times 12 \times 3 = 324$ byte) が残留してもオーバーフローすることはありません。
 もしもオーバーフローがおこると、レスポンスデータの一部が消滅するという致命的な問題が発生します。

(4) データを受信すると直ちに、(2) 項のコマンド列を1回送信します。

(3) と (4) を繰り返して、連続的にサンプリングを実行してゆきます。

(完)

DACS-2500K 製品内容

製品の名称	USB接続デジタル入出力基板 DACS-2500K
標準構成	DACS-2500K基板 1枚 デジタル入出力接続用ケーブル 30cm 1本 (機器接続側はコネクタなしの解放端となっています) デバイスドライバ/サンプルプログラム/取扱説明書は ダウンロードにて

製造販売	ダックス技研株式会社 ホームページ https://www.dacs-giken.co.jp
------	--

DACS25KSTD20702A