

デジタル入出力 DACS-2500KB-ST3 (2500K-STD)

Python 関数	windows版	Ddio.py
	Raspberry_Pi版	Ddio_rpi.py

Windows版

Windows11/10 (64bit) 、Python 3 にて動作
ダックス技研HPよりダウンロード
フォルダ dacs2500KB_STD¥Python(win)¥Ddio_py にある
Ddio.py と FT.py を同じディレクトリに格納
以下のサンプルプログラムを動作させる場合は、
Ddio_sample.py、Ddio_IDsample.py も格納します。
デバイスドライバをインストールしておきます。

Raspberry_Pi版

Raspberry pi の Python にて動作
フォルダ dacs2500KB_STD¥Python(rpi)¥Ddio_rpi_py にある
Ddio_rpi.py と FT_rpi.py を同じディレクトリに格納
以下のサンプルプログラムを動作させる場合は、
Ddio_rpi_sample.py、Ddio_rpi_IDsample.py も格納します。
デバイスドライバをインストールしておきます。

以下は、Window版のプログラム名で記述しています。

Raspberry_Pi版は、プログラム名を

Ddio_rpi_sample.py、Ddio_rpi_IDsample.py、Ddio_rpi_py に読替えてください。

サンプルプログラム Ddio_sample.py、Ddio_IDsample.py は、キー入力したデータ (文字列) を、そのまま、関数 Ddio.py の Ddio_write に送りますので、Ddio.py の機能確認ができます。

サンプルプログラム Ddio_sample.py と Ddio_IDsample.py の相違点

サンプルプログラム Ddio_sample.py

デバイスのID番号は 0番固定です。同種デバイスの複数接続はできません。
カウンタ/モーションコントロール/PWMパルス出力の異種デバイスと識別接続します。

サンプルプログラム Ddio_IDsample.py

デバイスにID番号を設定した複数デバイスを接続し、ID番号を指定してデバイスを選択することが可能。

サンプルプログラム Ddio_sample.py を起動すると、キー入力待ちになります。

例1 **DW, 0100CA** と16進数6桁で24bitのデジタル出力を指定します。
この例では、bit16, 7, 6, 3, 1 がON。その他のbitはOFFとなります。
変化をさせないbitがある場合は、16進数にはない文字(例えばx)を指定します。

例 **DW, 01xxCA**

出力のあと、デジタル入力状態を **DI, 020053** のように返してきます。

```
DW, 0100CA  
DI, 020053
```

例2 **DR** とデジタル入力読取りのみを指定します。デジタル出力は変化しません。

```
DR  
DI, 020053
```

例3 コマンド実行間隔の設定

IT, 1000000 とコマンド実行間隔を設定できます。実行間隔 1秒の例
10進数 最大 1秒 単位 1 μ s

IT, 0 コマンド実行間隔を解除(最小間隔)とします。

いずれも、デジタル入力状態を **DI, 020053** のように返してきます。

例4 PWMパルス出力の設定 デジタル出力bit0 -- チャンネル0
デジタル出力bit1 -- チャンネル1

WP, 0, 1000 チャンネル0のパルス幅を1000 μ sに指定

WP, 1, 3000 チャンネル1のパルス幅を3000 μ sに指定

G パルス出力開始

N パルス出力終了(デジタル出力bit0と1が汎用出力に戻る)

いずれも、デジタル入力状態を **DI, 020053** のように返してきます。

/ (スラッシュ)で複数の指示データを連結できます。

例5 指示データの連結

```
WP, 0, 1000/WP, 1, 3000/G
```

デジタル入力状態の指示データ分を **DI, 020053/DI, 020053/DI, 020053**
のように連結して返してきます。

サンプルプログラム Ddio_IDsample.py を起動すると、ID番号のキー入力待ちになります。

ID番号を 0~F (16進数) の範囲で指定すると、一致するID番号のデバイスを検索して
接続します。

その後の操作は、サンプルプログラム Ddio_sample.py と同じです。

関数 `Ddio.py / Ddio_rpi.py` の説明

`def Ddio_init(devnum):`

デバイスの初期化

引数 : デバイス番号 (int型) 通常は0
複数デバイス接続時の自動選択は9
自動選択は、カウンタ/モーションコントロール/PWMパルス出力の
異種デバイスを接続したときのみ有効
同種デバイスが複数ある場合は、自動選択不可。

(参考) デジタル入出力DACS-2500KB-STD が複数ある場合は、
デバイスにID番号を設定し、初期化関数 `Ddio_initID` を使用すると、
ID番号を指定してデバイスを選択することが可能。
デジタル入出力DACS-2500KB-STDのID番号を0以外に設定すれば、
カウンタ/モーションコントロール/PWMパルス出力のいずれかとの
同時使用も可能。

戻り値 : (1)'OK' 正常終了 'NG' 異常 , (2)デバイスハンドル

デバイスをOPEN

使用例 `ret = Ddio.Ddio_init(0)`
`if ret[0] == 'OK':`
`handle = ret[1] # デバイスハンドルをセット`
↑ 戻り値からデバイスハンドルをセットしてください。

`def Ddio_initID(IDnum):`

デバイスの初期化 ID番号使用

引数 : デバイスID番号 (16進数1文字 '0'~'F')
デバイスにID番号を設定し、この関数でデバイスの初期化をすると、
ID番号を指定してデバイスを選択することが可能。

戻り値 : (1)'OK' 正常終了 'NG' 異常 , (2)デバイスハンドル

デバイスをOPEN

使用例 `ret = Ddio.Ddio_initID('5')`
`if ret[0] == 'OK':`
`handle = ret[1] # デバイスハンドルをセット`
↑ 戻り値からデバイスハンドルをセットしてください。
↓ 文字で指定 ID=5 の例

def Ddio_write(handle, wdata):

コマンド送信とレスポンス受信

引数 : デバイスハンドル, 指示データ

戻り値 : いずれの応答も文字列 **DI, y--y** 異常時 戻り値なし

y--y デジタル入力(16進数6桁 bit23~0)

指示データ(文字列)

DW, y--y デジタル出力(16進数6桁 bit23~0)とデジタル入力読取り
y--y が6桁未満の場合はbit23からの指定、以降は未指定にて変化なし

DR デジタル入力読取りのみ

WP, x, y--y PWM出力パルス幅設定 デジタル出力bit0 : ch0 bit1 : ch1
x チャンネル番号 (0または1) **y--y** パルス幅 (整数値 単位: 1 μ s)
PWM出力パルス幅設定例 **WP, 0, 1500**

G パルス出力開始 デジタル出力bit0、bit1がPWMパルス出力

N パルス出力終了 (初期状態) デジタル出力bit0、bit1が汎用デジタル出力

IT, y--y コマンド実行間隔設定 **y--y** 実行間隔 (整数値 単位: 1 μ s)
以後コマンド実行間隔がすべて指定時間となります。解除は **y--y** に0を指定

/ (スラッシュ)で複数の指示データ連結可能。 戻り値も / (スラッシュ)で連結

例 **IT, 500000/DW, 56AF01/DW, 06AF00**

0.5秒間隔で出力を変化させる例

データ連結では、一部のデータが不正のとき、すべての転送なし

def Ddio_close(handle):

デバイスをCLOSE

引数 : デバイスハンドル 戻り値 : なし

使用例 **Ddio.Ddio_close(handle)**

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
------	--

DACS25DDIO24828A