

カウンタ DACS-2500KB シリーズ

Python 関数	windows版	Dcnt.py
	Raspberry_Pi版	Dcnt_rpi.py

Python関数 Dcnt(_rpi).py で、カウンタのプログラミングが容易に。

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_CNT¥Python(win)¥Dcnt_py にある
Dcnt.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、Dcnt_sample.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_CNT¥Python(rpi)¥Dcnt_rpi_py にある
Dcnt_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Dcnt_rpi_sample.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 Dcnt_rpi_sample.py、Dcnt_rpi_py に読替えてください。

サンプルプログラム Dcnt_sample.py で使用している、
print行を戻すエスケープシーケンスの動作について

Windows版

Dcnt_sample.py をダブルクリックで起動するか
ツール → コマンドプロンプト から起動してください。
コマンドラインで起動すると正常な動作をしません。

Raspberry_Pi版

プログラミングツール Thonny で動作させた場合、サンプルプログラムの
print行を戻すエスケープシーケンスが正常に動作しないため、
カウンタ値のprint繰返し時間10msを、1秒程度に変更して使用してください。
Terminal上では正常に動作します。

サンプルプログラム Dcnt_sample.py は、キー入力したデータ（文字列）を、そのまま関数 Dcnt.py の Dcnt_write に送りますので、Dcnt.py の機能確認ができます。まずは、サンプルプログラム Dcnt_sample.py を起動してください。

下図の上3行のように、カウンタ6chのカウンタ値とホールド値を連続表示します。Cに続く行は、各チャンネルのカウンタ値です。Hに続く行は、各チャンネルのゲート信号によるホールド値です。

続いて、キー入力にて、以下の操作をします。
操作方法はサンプルプログラムに記述している注釈をご覧ください。

チャンネル0番の カウンタリセット入力を 無効にします。	↓ OR0 (ゼロ、アール、ゼロ)	カウンタのリセット入力が無接続の場合 入力Highでリセット側になっていますので、 リセット入力を仮に無効にします。 カウンタのリセット入力が適切に 接続されている場合は、この操作は必要 ありません。
カウント開始	OG	チャンネル0番がカウントを始めます。

先頭の数字はチャンネル番号です。
0~5の数字でチャンネル番号を指定し、指定チャンネルの操作をします。

操作内容の詳細は、関数 Dcnt_write 使用例をご覧ください。サンプルプログラムでは、キー入力した内容が、そのまま Dcnt_write の指示データ文字列となります。

```

| ch0 | 1 | 2 | 3 | 4 | 5 |
C      0   0   0   0   0   0
H      0   0   0   0   0   0
OR0
OK
| ch0 | 1 | 2 | 3 | 4 | 5 |
C      0   0   0   0   0   0
H      0   0   0   0   0   0
OG
OK
| ch0 | 1 | 2 | 3 | 4 | 5 |
C      2084812   0   0   0   0   0
H      0   0   0   0   0   0

```

/ (スラッシュ)で複数の指示データを連結できます。

例 OEC,1000/OAB/OG
 ↑ ↑
 連結 連結
 ch0 最終値を1000にて、カウント繰返し
 ch0 AB相カウントモード
 ch0 カウント開始

関数 Dcnt.py / Dcnt_rpi.py の説明

def Dcnt_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 (最終ページ参照)
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 = Dcnt.Dcnt_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 Dcnt_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 Dcnt.Dcnt_close(handle)

def Dcnt_read(handle, rmode):

カウント値またはホールド値の取得

引数 : デバイスハンドル, 読取モード (文字 0~5、9、0H~5H、9H、D)

0~5	指定チャンネルのカウント値
9	全6チャンネルのカウント値
0H~5H	指定チャンネルのホールド値
9H	全6チャンネルのホールド値
D	デジタル入力

戻り値 : 文字列

単独チャンネル指定のとき

指定チャンネルのカウント値またはホールド値 10進数

全チャンネル指定のとき

第0チャンネルから第5チャンネルまでのカウント値またはホールド値を、
10進数でカンマ区切り

デジタル入力のとき 16進数6桁 bit23~0

使用例 readdata = Dcnt.Dcnt_read(handle, '0')

使用例 readdata = Dcnt.Dcnt_read(handle, '0H')

戻り値の例 文字列 120

使用例 readdata = Dcnt.Dcnt_read(handle, '9')

使用例 readdata = Dcnt.Dcnt_read(handle, '9H')

戻り値の例 文字列 120, -250, 0, 0, 654300, -420000

使用例 readdata = Dcnt.Dcnt_read(handle, 'D')

戻り値の例 文字列 030C0A

def Dcnt_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 : 'OK' 正常終了 'NG' 異常終了

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

指示データ(文字列)

/ (スラッシュ)で複数の指示データを連結可能

例 OEC,1000/OAB/OG

ch0の最終値を1000にて繰返し

ch0をAB相カウントモード

ch0のカウント開始

X:チャンネル番号(0~5)

xG カウント開始

xS カウント停止(電源投入初期)

xR カウント値リセット

xF, Z--Z フィルタ設定 Z--Z 小数点可 単位 μs

最小単位 0.015625 μs

最大値 16383 μs

xAB A相/B相カウント

xUD up/downカウント(電源投入初期)

xR1 カウントリセット入力有効(電源投入初期)

xR0 カウントリセット入力無効(例 ch0ではDI02)

xT1 ゲート入力有効

xT0 ゲート入力無効(電源投入初期)

xMI 周期計測

(74)

xMW パルス幅計測

xMN 通常カウント動作(電源投入初期)

xEC, Z--Z 最終値設定(Z--Z 10進数で最終値) くり返し動作

xES, Z--Z 最終値設定(Z--Z 10進数で最終値) 最終値で停止

xEN 最終値設定解除(電源投入初期)

DI, y--y デジタル入力極性(y--y 16進数6桁 bit23~0)

(74)

1:反転 0:反転なし

DL1 デジタル出力(bit11~0) カウンタ用出力に変更

DL0 デジタル出力(bit11~0) 汎用出力用途(初期)

DH1 デジタル出力(bit23~16) カウンタ用出力(カウンタ使用時)

DH0 デジタル出力(bit23~16) 汎用出力用途に変更

DO, y--y 汎用デジタル出力(y--y 16進数6桁 bit23~0)

(74)

使用例

```
readdata = Dcnt.Dcnt_write(handle, '3G')
          第3チャンネルのカウント開始          正常設定時の戻り値 'OK'
```

以下、指令データ文字列部分のみ表示

```
'3R0'      第3チャンネルのリセット入力無効
'3S'       第3チャンネルのカウント停止
'3R'       第3チャンネルのカウント値リセット

'2F, 12.5' 第2チャンネルのフィルタ設定 12.5  $\mu$ s
            入力のlow側とHigh側について、12.5  $\mu$ s以下の信号を除去

'4AB'      第4チャンネルをAB相カウントモードに設定
'4UD'      第4チャンネルをUp/Downカウントモードに設定

'5MI' (5、14、74) 第5チャンネルを周期計測に設定
'5MW'      第5チャンネルをパルス幅計測に設定
'5MN'      第5チャンネルを通常カウント動作に戻す

'1EC, 10000' 第1チャンネルの最終値に 10000を設定
            10000カウントの次に0値に戻ってカウントを繰り返す。
            Downカウントのときは、0の次に最終値となって繰り返す。
            分周出力を利用する場合の設定。

'1ES, 10000' 第1チャンネルの最終値に 10000を設定
            10000カウントにて停止。
            Downカウントのときは、0にて停止。

'1EN'      第1チャンネルの最終値を解除

'DI, 400A04' (74) デジタル入力極性を設定
            bit22、bit11、bit9、bit2 を反転した例

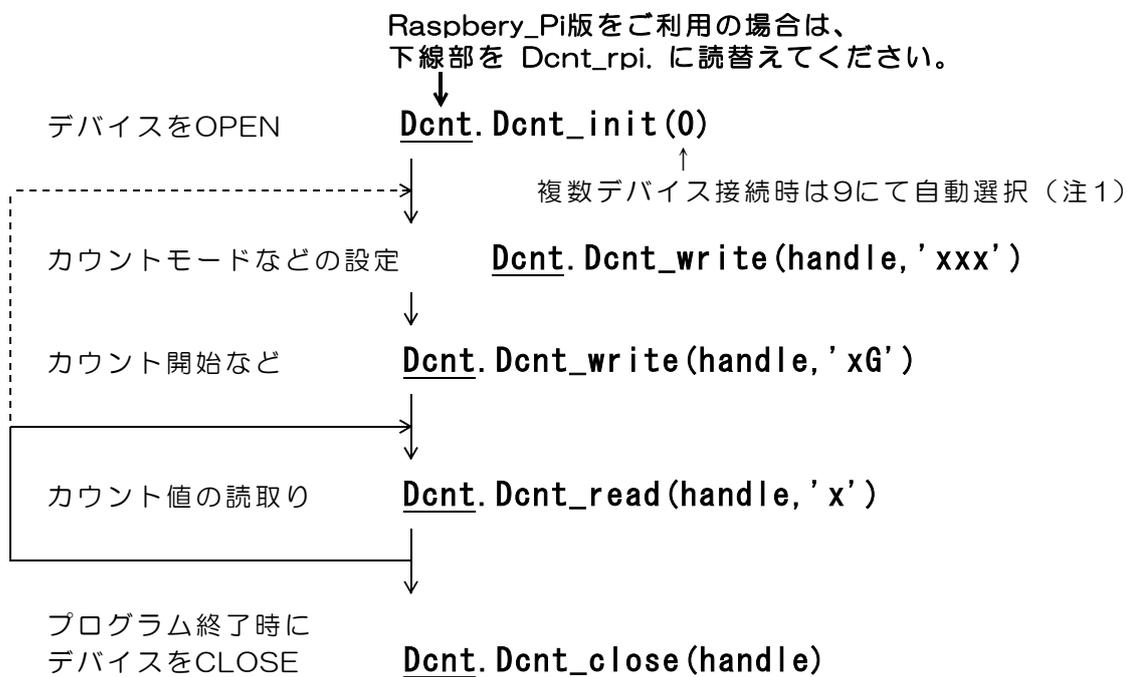
'DL1'      デジタル出力(bit11~0)をカウンタ用出力に変更
            デジタル出力(bit11~0)は、通常、汎用出力用になっています。
            この設定で、テスト用のパルス出力および分周出力になります。

'DH0'      デジタル出力(bit23~16)を汎用出力用に変更
            デジタル出力(bit23~16)は、カウンタ動作を選択すると、分周出力に
            なっていますが、この設定で汎用出力用になります。
            bit15~12はテスト用のパルス出力固定となっていますので、
            この設定でも汎用出力とはなりません。
```

複数の指示データを連結する例

```
readdata = Dcnt.Dcnt_write(handle, 'OEC, 1000/OAB/OG')
          ch0の最終値を1000にて繰返し
          ch0をAB相カウントモード
          ch0のカウント開始
```

Dcnt.py を使ったプログラミング手順例



(注1) 複数デバイス接続時はデバイス番号部分を9にて自動選択可。
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。

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>

DACS25DCNT24709B