

## PWMパルス出力 DACS-2500KB-RSW4

Python 関数	windows版	Drsw.py
	Raspberry_Pi版	Drsw_rpi.py

Python関数 Drsw(\_rpi).py で、PWMパルス出力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\_RSW¥Python(win)¥Drsw\_py にある  
Drsw.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、Drsw\_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\_RSW¥Python(rpi)¥Drsw\_rpi\_py にある  
Drsw\_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Drsw\_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 Drsw\_rpi\_sample.py、Drsw\_rpi\_py に読替えてください。

サンプルプログラム Drsw\_sample.py の表示横幅について  
12チャンネル分のパルス幅を1行で表示するために、表示画面の横幅を広げる必要があります。

### Windows版

Drsw\_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した場合、表示横幅が不足しますので、横幅を広げて使用してください。

### Raspberry\_Pi版

Drsw\_rpi\_sample.py では、表示横幅が 90文字以上であることを確認しています。  
プログラミングツール Thonny で動作させた場合、表示横幅を拡大しても、  
横幅取得の戻り値が、80 固定となるため、Drsw\_rpi\_sample.py の表示横幅確認  
部分を、90 から 80 に変更して使用してください。  
Terminal では、横幅を拡大すれば、プログラムはそのまま正常に動作します。

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

まずは、サンプルプログラム Drsw\_sample.py を起動してください。

下図、1行目のように、パルス出力基準クロック周波数とパルス出力周期を表示します。起動直後の表示は未定となっていますが、デバイス電源投入時の初期値は 1MHz 20ms です。コマンドで設定をしない場合は、電源投入時の初期値が継続します。

2～3行に、ch0～11 12チャンネル分のパルス幅（単位はクロック数、16進数表記）、s パルス出力変化状態、DI デジタル入力の状態、chQ（初期はch0 チャンネル番号は変更可能）表示指定チャンネルのパルス幅（ $\mu s$  単位）を表示します。

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。

基準クロック周波数と 周期を設定	I, 1, 20000 (71)	1 : クロック周波数 1MHz 単位MHz 20000 : 20000 $\mu s$ 単位 $\mu s$
各チャンネルの パルス幅を設定	W, 0, 1000	0 : チャンネル番号 0 1000 : 1000 $\mu s$ 単位 $\mu s$

これにて、基準クロック周波数 1 MHz、周期 20ms (50Hz)、第0チャンネルのパルス幅が 1000  $\mu s$ となります。

各チャンネルの パルス幅を設定	W, 11, 2000	11 : チャンネル番号 11 2000 : 2000 $\mu s$ 単位 $\mu s$
--------------------	-------------	---

第11チャンネルのパルス幅が 2000  $\mu s$ となります。

パルス幅表示チャンネル 変更	11	11 : チャンネル番号 11
-------------------	----	-----------------

```

clock= ***MHz  interval= ***.**us
|ch0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | s | DI | ch0
03E8 05F0 000 FFFFFFF ***.**us
I,1,20000
OK
clock = 1.0MHz  interval = 20000.00us 04e20
|ch0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | s | DI | ch0
03E8 05F0 000 FFFFFFF 1000.00us
W,0,1000
OK
clock = 1.0MHz  interval = 20000.00us 04e20
|ch0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | s | DI | ch0
03E8 05F0 000 FFFFFFF 1000.00us
W,11,2000
OK
clock = 1.0MHz  interval = 20000.00us 04e20
|ch0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | s | DI | ch0
03E8 05F0 07D0 000 FFFFFFF 1000.00us
11
clock = 1.0MHz  interval = 20000.00us 04e20
|ch0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | s | DI | ch11
03E8 05F0 07D0 000 FFFFFFF 2000.00us

```



## def Drsw\_read(handle, rmode) :

パルス幅の現在値取得

速度指定をして変化中は変化中の現在値

引数 : デバイスハンドル, 読取モード (文字 0~12)  
0~11 指定チャンネルのパルス幅  
12 全チャンネルのパルス幅

戻り値 : 文字列

単独チャンネル指定のとき 16進数4桁

指定チャンネルのパルス幅 (単位 クロック数)

全チャンネル指定のとき

第0チャンネルから第11チャンネルまでのパルス幅 (単位 クロック数) を、  
カンマ区切りで各16進数4桁

使用例 readdata = Drsw.Drsw\_read(handle, '0')

戻り値の例 文字列 05F0

使用例 readdata = Drsw.Drsw\_read(handle, '11')

戻り値の例 文字列 07D0

使用例 readdata = Drsw.Drsw\_read(handle, '12')

戻り値の例 文字列

03E8, 05F0, 07D0

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑  
ch0 1 2 3 4 5 6 7 8 9 10 11

## def Drsw\_sread(handle) :

ステータスとデジタル入力の取得

引数 : デバイスハンドル

戻り値 : 文字列 パルス幅変化状態, デジタル入力

パルス幅変化状態 第11~第0チャンネルの順で、12bitの16進数3桁

ON (1) : 変化中 OFF (0) : 停止中

デジタル入力 16進数6桁 bit23~0

使用例 readdata = Drsw.Drsw\_sread(handle)

戻り値の例 文字列 A01, 82CA

第11, 9, 0チャンネルが変化中 その他は停止中

デジタル入力 bit23, 9, 7, 6, 3, 1 がHigh その他はLow

## def Drsw\_iread(handle) :

クロック周波数と周期の取得

引数 : デバイスハンドル

戻り値 : 基準クロック周波数 (MHz) 実数値, 周期 (クロック数) 整数値

デバイスから読取った値ではありません。

Drsw\_wite で設定している値を返します。

未設定のときは、基準クロック周波数の値が 99.0 で戻ります。

使用例 readdata = Drsw.Drsw\_iread(handle)

戻り値の例 16.0 (実数値), 320000 (整数値)

## def Drsw\_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 W,1,1000/W,2,1600/W,3,2000 第1~3チャンネルのパルス幅を設定

**G** パルス出力開始

**N** パルス出力終了 (パルス出力なし)

**I, y--y, Z--Z** 周期設定

(アイ) **y--y** クロック周波数設定(MHz) 下記のいずれか  
0.5, 1, 2, 4, 8, 16, 32, 64

**Z--Z** 周期 小数点可 (単位:  $\mu\text{s}$ )

**IP, y--y, Z--Z** 周期設定

(アイピ) **y--y** クロック周波数設定(MHz) 上記コマンドと同じ

**Z--Z** 周期 (単位: 基準クロック数)

**W, xx, y--y** 目標パルス幅設定

**xx** チャンネル番号 (0~11) 全チャンネル指定(12)

**y--y** 目標パルス幅 小数点可 (単位:  $\mu\text{s}$ )

**WP, xx, y--y** 目標パルス幅設定

**xx** チャンネル番号 (0~11) 全チャンネル指定(12)

**y--y** 目標パルス幅 (単位: 基準クロック数)

**V, xx, y--y** 変化速度設定

**xx** チャンネル番号 (0~11) 全チャンネル指定(12)

**y--y** 変化速度 小数点可 (単位: パルス幅  $\mu\text{s} / \text{s}$ )

(注1) 最低変化速度 毎秒  $1.53 \mu\text{s}$

(注2) 「直ちに指定パルス幅にする利用」では、本コマンドを使用しない。  
このコマンド使用後に、「直ちに指定パルス幅とする」設定に戻す  
場合は、このコマンドで、変化速度に100000以上を指定する

**VP, xx, y--y** 変化速度設定

**xx** チャンネル番号 (0~11) 全チャンネル指定(12)

**y--y** 変化速度 (単位: 基準クロック数 / s)

「直ちに指定パルス幅にする利用」では、本コマンドを使用しない。

**S** パルス幅変化停止

パルス幅変化を停止し、

停止後のパルス幅を目標パルス幅に変更して後、停止を解除

**DP, y--y** デジタル出力極性設定

**y--y** 16進数6桁 bit23~0

**DO, y--y** デジタル出力設定

(ディオ) **y--y** 16進数6桁 bit23~0

パルス出力を開始している場合、bit11~0は指定しても無効

使用例

```
readdata = Drsw.Drsw_write(handle, 'G')
          パルス出力開始           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

```
'G'           パルス出力開始
'N'           パルス出力終了（パルス出力なし）

'I, 16, 20000'   周期設定   クロック周波数16MHz  周期  20ms
(アイ)

'IP, 0.5, 30000'  周期設定   クロック周波数0.5MHz  周期  30000クロック

'W, 1, 1600'     目標パルス幅設定  第1チャンネル  パルス幅  1600μs
'WP, 10, 1100'  目標パルス幅設定  第10チャンネル  パルス幅  1100クロック

'V, 9, 89.5'    変化速度設定  第9チャンネル  毎秒  89.5μs
'VP, 9, 50'     変化速度設定  第9チャンネル  毎秒  50基準クロック数

'S'            パルス幅変化停止
              停止後のパルス幅を目標パルス幅に変更した後、停止を解除

'DP, 400A04'    デジタル出力極性設定
              bit22、bit11、bit9、bit2 を反転した例

'DO, 2A0000'    デジタル出力設定
(ディオー)     bit21、bit19、bit17 をHigh(1)、その他はLow(0)
              パルス出力を開始している場合、bit11~0は指定しても無効
```

複数の指示データを連結する例

```
readdata = Drsw.Drsw_write(handle, 'V, 9, 89.5/W, 9, 1600')
          第9チャンネルの変化速度を 毎秒89.5μs
          第9チャンネルの目標パルス幅を1600μs
```

## 複数チャンネルのパルス幅変化を同時スタート&ストップ

関数 Drsw\_pmove は、以下の処理(1~7)を実行

(1) 変化対象となる各チャンネルの現在値を読取ります。

```
readdata = Drsw.Drsw_read(handle, 'xx')
```

(2) 各チャンネルのパルス幅現在値を μs 単位に変換

(3) パルス幅の変化量「目標 - 現在値」を算出

(4) 合成変化パルス幅算出  $\sqrt{\text{対象チャンネルのパルス幅変化量}^2 \text{の合計}}$

(5) 対象チャンネルの変化速度を算出

変化速度 \* パルス幅の変化量 / 合成変化パルス幅

(6) デバイスに変化速度を設定

```
Drsw.Drsw_write(handle, 'V, xx, yyy/V, xx, yyy/--/V, xx, yyy')
```

(7) デバイスに目標パルス幅を設定

必ず、各チャンネルの目標パルス幅を連結して送信します。

```
Drsw.Drsw_write(handle, 'W, xx, yyy/W, xx, yyy/--/W, xx, yyy')
```

指定複数チャンネルのパルス幅変化が同時にスタートし、同時に終了します。

## def Drsw\_pmove(handle, pdata):

複数チャンネルの同時スタート&ストップ

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

指示データ(文字列) 'L/z--z/xx, y--y/xx, y--y/ -- /xx, y--y'

Z--Z 変化速度 (単位:  $\mu\text{s} / \text{s}$ ) 小数点可

XX チャンネル番号 0~11

y--y 目標パルス幅 (単位:  $\mu\text{s}$ ) 小数点可

対象となるチャンネルのデータを / (スラッシュ) で連結

指示データ(文字列) 'LP/z--z/xx, y--y/xx, y--y/ -- /xx, y--y'

Z--Z 変化速度 (単位: クロック数 / s)

XX チャンネル番号 0~11

y--y 目標パルス幅 (単位: クロック数)

対象となるチャンネルのデータを / (スラッシュ) で連結

任意の複数グループに分けた並行動作も可能です。

例 チャンネル0,1,3 を第1グループとして **pmove**で動作

チャンネル5,6,11 を 第2グループとして **pmove** で動作

使用例

```
readdata = Drsw.Drsw_pmove(handle, 'L/100/0, 2000/1, 1200/3, 2500')  
合成変化速度 100 $\mu\text{s}/\text{s}$   
ch0 2000 $\mu\text{s}$  ch1 1200 $\mu\text{s}$  ch3 2500 $\mu\text{s}$  ^変化
```

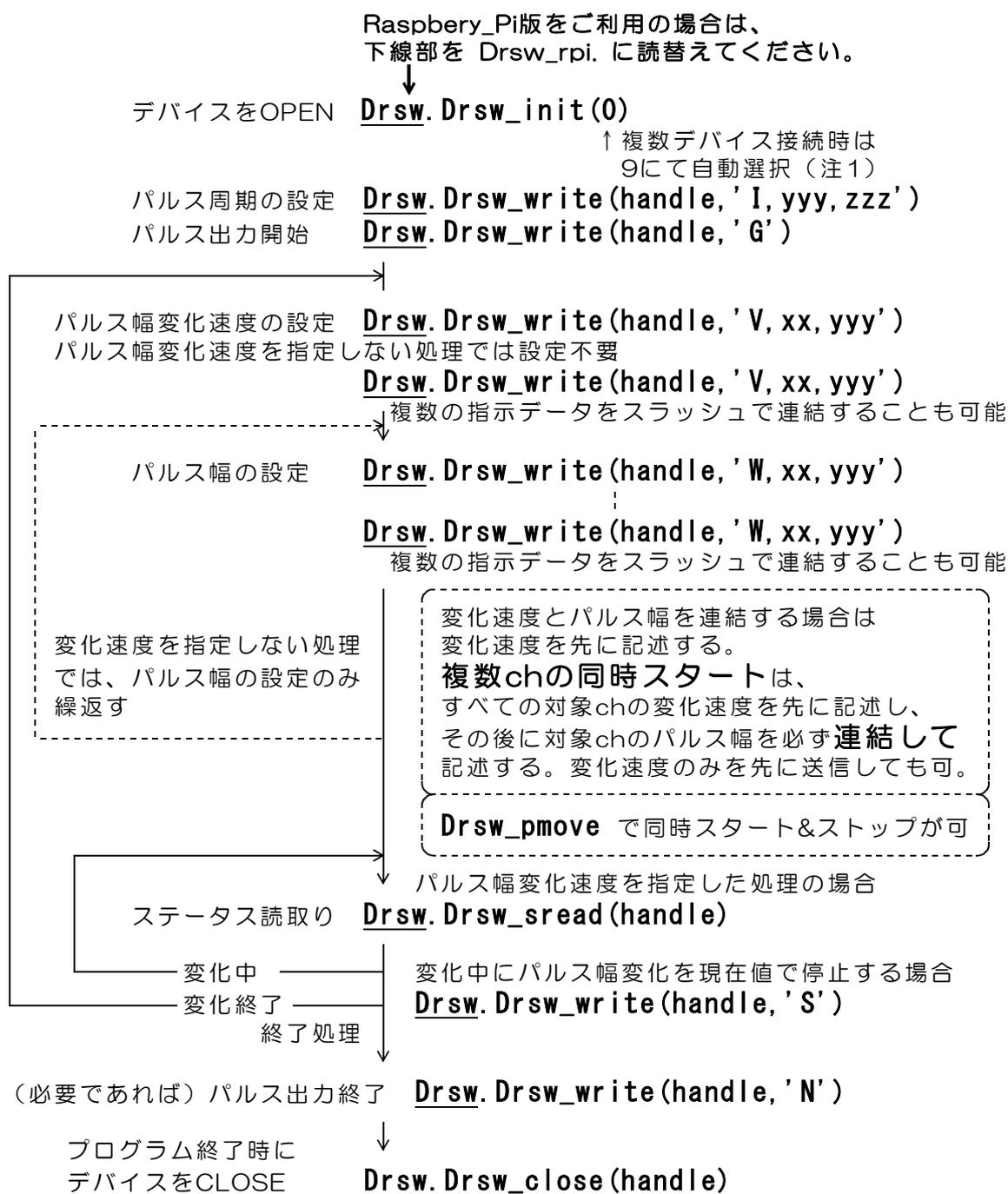
```
readdata = Drsw.Drsw_pmove(handle, 'L/180.5/5, 500/6, 1500/11, 250')  
合成変化速度 180.5 $\mu\text{s}/\text{s}$   
ch5 500 $\mu\text{s}$  ch6 1500 $\mu\text{s}$  ch11 250 $\mu\text{s}$  ^変化
```

サンプルプログラム Drsw\_sample.py のキー入力で Drsw\_pmove の動作確認可能  
キー入力例 L/180.5/0, 500/1, 1500/11, 250

(注1) 変化速度の分解能は  $1/65536$  最低変化速度は 毎秒  $1.53\mu\text{s}$   
遅い変化速度を指定した場合で、パルス幅変化が多いチャンネルと、パルス幅変化が少ないチャンネルがある場合、パルス幅変化の少ないチャンネルが、最低変化速度となると、同時ストップではなく、先に変化を終了します。

(注2) Drsw\_pmove を実行した後は、各チャンネルに設定した変化速度が、そのまま有効になっています。Drsw\_pmove の後で、Drsw\_write でパルス幅を設定すると、直前の Drsw\_pmove での変化速度で変化します。

## Drsw.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>

DACS25DRSW24717A