

モーションコントローラ DACS-2500KB シリーズ

Python 関数 windows版 Dpmv.py
 Raspberry_Pi版 Dpmv_rpi.py

Python関数 Dpmv(_rpi).py で、モーションコントローラ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_PM6¥Python(win)¥Dpmv_py にある
Dpmv.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、Dpmv_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_PM6¥Python(rpi)¥Dpmv_rpi_py にある
Dpmv_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Dpmv_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 Dpmv_rpi_sample.py、Dpmv_rpi_py に読替えてください。

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

次ページの図、上2行のように、各軸位置、デジタル入力状態、ステータスを表示します。

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。

加速度を設定	A, 10	単位：KHz/s (分解能 1.25KHz/s)
速度を設定	V, 1000	単位：Hz (分解能 0.25Hz)
第1軸の移動量を指定	1, 1000	加速度/速度は小数点可
移動開始	G	単位：パルス数 (整数)

これにて、第1軸が移動を開始します。
複数軸を同時に移動させる場合は、移動開始前に、複数軸を指定します。

第2軸の移動量指定例	2, 1000	移動中に強制停止する場合は S
第3軸の移動量指定例	3, -1000	
移動開始	G	

多くの機能のなかで、ここまでが基本的な操作です。

```

| p.1 | p.2 | p.3 | p.4 | p.5 | p.6 | DI | A 6 state |
| 0 | 0 | 0 | 0 | 0 | 0 | FFFFFFF /,N,00000
A,10
OK
| p.1 | p.2 | p.3 | p.4 | p.5 | p.6 | DI | A 6 state |
| 0 | 0 | 0 | 0 | 0 | 0 | FFFFFFF /,N,00000
V,1000
OK
| p.1 | p.2 | p.3 | p.4 | p.5 | p.6 | DI | A 6 state |
| 0 | 0 | 0 | 0 | 0 | 0 | FFFFFFF /,N,00000
I,1000
OK
| p.1 | p.2 | p.3 | p.4 | p.5 | p.6 | DI | A 6 state |
| 0 | 0 | 0 | 0 | 0 | 0 | FFFFFFF /,N,00000
G
OK1
| p.1 | p.2 | p.3 | p.4 | p.5 | p.6 | DI | A 6 state |
| 1000 | 0 | 0 | 0 | 0 | 0 | FFFFFFF /,N,00000

```

G \square と入力すると、関数 Dpmv.py が、移動量(絶対値)の最も多い軸をマスター軸と決めて移動を開始します。

Vで指定した速度は、マスター軸の速度となり、その他の軸は直線補間速度となります。

G \square の戻り値 OK の後に続く数字は、関数 Dpmv.py が判定したマスター軸番号です。

ご注意 移動停止後も、各軸に指定した移動量は有効になっています。
先に移動させた軸を、次に移動させない場合は、その軸の移動量に0を指定する必要があります。

ご参考 移動量指定で 9, 0 \square とすると全軸が0指定になります。

/ (スラッシュ)で複数の指示データを連結できます。

例 1,100/2,-1000/G \square 第1軸と第2軸の移動量を指定し移動開始

↑ ↑

連結 連結

関数 `Dpmv.py / Dpmv_rpi.py` の説明

`def Dpmv_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

各軸指定移動量を初期化します。

電源投入時に、デバイス側でも初期化をしますが、このプログラムでの初期化処理は、デバイスを再接続しないで、プログラムを再起動した場合の対策です。

第6軸独立動作の設定があれば、独立動作を継続します。

使用例

```
ret = Dpmv.Dpmv_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 Dpmv_read(handle, rmode):`

移動量または位置の取得

引数 : デバイスハンドル, 読取モード (M:移動量 P:位置)

戻り値 : 文字列

1軸から6軸の移動量(または位置)を、10進数でカンマ区切り、
デジタル入力(24bit 16進数 6桁)

使用例 `readdata = Dpmv.Dpmv_read(handle, 'P')`

戻り値の例 文字列 0, 0, 100, -20, 0, 0, A00230
1軸, 2, 3, 4, 5, 6, デジタル入力

`def Dpmv_sread(handle):`

ステータスの取得

引数 : デバイスハンドル

戻り値 : 文字列 X, y, Z--Z

X: 1~6軸(第6軸独立のときは1~5)の状態

G:移動中 D:ドウェル時間中 /:待機 S:停止

y: 第6軸独立動作の状態

N:第6軸独立なし G:移動中 D:ドウェル時間中 /:待機 S:停止

Z--Z: ステータス (16進数 下位5桁)

モーションコントローラ説明書 Qコマンドの応答をご覧ください。

使用例 `readdata = Dpmv.Dpmv_sread(handle)`

戻り値の例 文字列 G, N, 00003

先頭の文字が / (スラッシュ) のとき待機中

先頭の文字が G のとき移動中と判断できます。

def Dpmv_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

G、GC、GD 移動開始の場合 'OKx' x:マスター軸番号1~6
複数指示データ連結では、最後が移動開始指示のとき軸番号付加
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 1,100/2,-1000/G

X, y--y	各軸の移動量を指定 X :軸番号(1~6) y--y :指定移動量(整数 単位 パルス) X=9 のとき、全軸に同じ移動量をセット
V, y--y	移動速度を指定(小数点可 単位 Hz)
A, y--y	y--y :加速度を指定(小数点可 単位 KHz/s) 台形加減速
A, y--y, Z--Z	y--y :加速度を指定(小数点可 単位 KHz/s) Z--Z :S字時間を指定(小数点可 単位 s)
G	移動開始
GC	エンドレス移動開始
DT, y--y	ドウェルタイム設定 y--y :ドウェルタイム(単位 ms)
GD	ドウェルタイム後の移動開始
S	強制停止
R	位置リセット (全軸の位置を0とする)
DH, xxx	デジタル出力(bit23~12) 16進数3桁で指定
LL, xxxxx	リミットスイッチ low ON 16進数5桁で指定
LH, xxxxx	リミットスイッチ High ON 16進数5桁で指定 16進数5桁の指定は、ボード説明書の「リミット入力信号による自動停止機能と原点設定への応用」などをご覧ください。
DP, xxx	パルス出力と移動方向出力極性 16進数3桁で指定 16進数3桁の指定は、ボード説明書のPコマンド「出力極性指定」を参照ください。
WY	ウォッチドクタイマー有効
WN	ウォッチドクタイマー無効
6Y	第6軸独立動作指定
6N	第6軸独立動作解除
6V, y--y	第6軸独立動作の速度を指定
6A, y--y	第6軸独立動作の加速度を指定 台形加減速 数値設定は A と同じ
6A, y--y, Z--Z	第6軸独立動作の加速度を指定 S字加減速 数値設定は A と同じ

6G 第6軸独立移動開始
6GC 第6軸独立エンドレス移動開始
6DT, y--y 第6軸独立動作ドウェルタイム設定 **y--y**:ドウェルタイム(単位ms)
6GD 第6軸独立動作 ドウェルタイム後の移動開始
6S 第6軸独立移動強制停止

使用例 `readdata = Dpmv.Dpmv_write(handle, '3, -1000')`
 第3軸の移動量に -1000パルス 戻り値の例 'OK'

使用例 `readdata = Dpmv.Dpmv_write(handle, 'V, 100')`
 移動速度 100Hz 戻り値の例 'OK'

使用例 `readdata = Dpmv.Dpmv_write(handle, 'G')`
 移動開始 戻り値の例 'OK2'
G、GC、GD 移動開始の場合は、戻り値 'OKx' x:マスター軸番号1~6

複数の指示データを連結する例

`readdata = Dpmv.Dpmv_write(handle, '1, 100/2, -1000/G')`
 第1軸に100パルス、第2軸に-1000パルス、移動開始 戻り値の例 'OK2'

def Dpmv_moveA(handle, adata):

全軸MOVE 最大6軸分の位置または移動量と移動速度を指定して目標位置に移動する
 位置指定 MD, 移動量指定 MI

引数 : デバイスハンドル, 指示データ

戻り値 : 'OKx' 正常終了 x:マスター軸番号1~6

'NG' データ異常 'ER' 送受信異常 'NA' 移動中につき不可
 指示データ(文字列)

`uu/z--z/x, y--y/x, y--y/ -- /x, y--y`

uu 移動モード **MD**:位置指定 **MI**:移動量指定

ドウェルタイム終了後に移動開始 **MDW**:位置指定 **MIW**:移動量指定

Z--Z 移動速度(小数点可 単位 Hz) 最も移動量の多い軸の速度となる

X 軸番号 1~6

y--y 移動量または目標位置 整数 (単位: パルス数)

対象となる軸データの最大6軸分を / (スラッシュ) で連結

指定のない軸は、移動対象とならないので、移動量0を指定する必要なし。

使用例

`readdata = Drsw.Drsw_moveA(handle, 'MD/55.5/1, -200/2, 1200/5, 2500')`
 位置指定、速度 55.5Hz
 第1軸位置 -200、2軸位置 1200、5軸位置 2500 ^移動開始
 マスター軸は移動量の最も多い軸となる 戻り値の例 'OK5'

```
readdata = Drsw.Drsw_moveA(handle, 'MI/1000/2, -100/3, 5000/6, 2500')
移動量指定、速度 1000Hz
第2軸移動量 -100、3軸移動量 5000、6軸移動量 2500 の移動開始
マスター軸は第3軸 戻り値の例 'OK3'
```

```
readdata = Drsw.Drsw_moveA(handle, 'MIW/1000/2, 100/4, -5000/6, 25')
移動量指定、速度 1000Hz
第2軸移動量 100、4軸 -5000、6軸 25 の移動をドウェルタイム後に開始
マスター軸は第4軸 戻り値の例 'OK4'
```

def Dpmv_move6(handle, data6) :

6軸単独MOVE 第6軸の位置または移動量と移動速度を指定して目標位置に移動する
位置指定 MD, 移動量指定 MI

Dpmv_move6 を使用すると、その後は、第6軸独立動作指定となります。Dpmv_writeで
第6軸独立動作 **6Y** を指定する必要はありませんが、全軸が第6軸独立動作なしの移動を
しているときは、Dpmv_move6で、第6軸独立動作を開始することはできませんので、
先に、Dpmv_write で第6軸独立動作 **6Y** を指定しておくか、全軸動作が停止している
ときに、Dpmv_move6 を使用してください。

引数 : デバイスハンドル, 指示データ

戻り値: 'OK' 正常終了 'NG' データ異常 'ER' 送受信異常 'NA' 移動中につき不可
指示データ(文字列)

uuu/z--z/y--y/

uuu 移動モード MD6 : 位置指定 MI6 : 移動量指定

ドウェルタイム終了後に移動開始 MD6W :位置指定 MI6W :移動量指定

Z--Z 移動速度(小数点可 単位 Hz)

y--y 移動量または目標位置 整数 (単位: パルス数)

使用例 readdata = Drsw.Drsw_move6(handle, 'MD6/55.5/-200')
位置指定、速度 55.5Hz -200 位置へ移動開始
戻り値の例 'OK'

```
readdata = Drsw.Drsw_move6(handle, 'MI6/1000/2500')
移動量指定、速度 1000Hz 移動量 2500 の移動開始
戻り値の例 'OK'
```

```
readdata = Drsw.Drsw_move6(handle, 'MI6W/1000/25')
移動量指定、速度 1000Hz
ドウェルタイム後に 移動量 25 の移動開始
戻り値の例 'OK'
```

def Dpmv_close(handle) :

デバイスをCLOSE

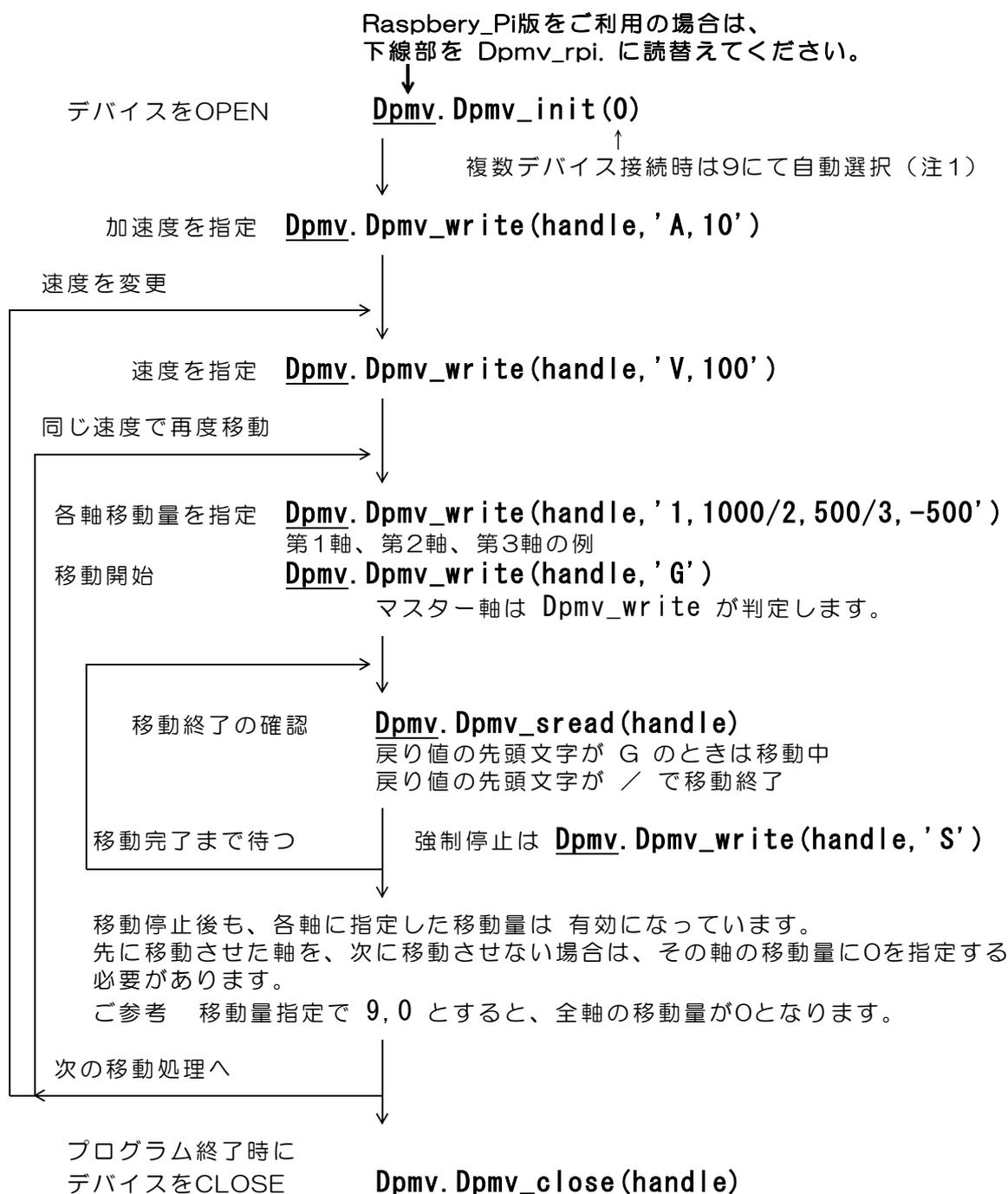
引数 : デバイスハンドル

戻り値 : なし

使用例 Dpmv.Dpmv_close(handle)

Dpmv.py を使ったプログラミング手順例

Dpmv_write を使用した移動例



(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。

Dpmv.py を使ったプログラミング手順例

Dpmv_moveA、Dpmv_move6 を使用した移動例



(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>

DACS25DPMV24816A