

# ～ローリングクロック プログラム説明書～

お買い上げいただきありがとうございます。

本書では付属しているモータで時計を動作させるためのプログラム例を解説しています。

「とりあえず動いているところを見てみたい!」という方は是非読んでみてください。

## はじめに

本プログラムは以下の環境で動作確認をしています。

- ◆ マイコンボード : Raspberry Pi Pico
- ◆ ファームウェア : MicroPython (v1.19.1.uf2)
- ◆ プログラム言語 : Python 3.7.9
- ◆ 開発環境 : Windows10、Thonny (Ver.3.3.13)

## プログラム

```
1 from machine import Pin, Timer
2 import utime
3 #IO PIN SET
4 IN1 = Pin(2, Pin.OUT)
5 IN2 = Pin(3, Pin.OUT)
6 IN3 = Pin(5, Pin.OUT)
7 IN4 = Pin(4, Pin.OUT)
8 SW1 = Pin(14, Pin.IN, Pin.PULL_UP)
9 SW2 = Pin(15, Pin.IN, Pin.PULL_UP)
10 pins = [IN1, IN2, IN3, IN4]
11 sequence = [[1,1,0,0],[0,1,1,0],[0,0,1,1],[1,0,0,1]]
12
13 def clock_move(timer):
14     for step in sequence:
15         for i in range(len(pins)):
16             pins[i].value(step[3-i])
17         utime.sleep_ms(8)
18     for step in sequence:
19         for i in range(len(pins)):
20             pins[i].value(0)
21
22 Timer().init(freq = 8.53325, mode = Timer.PERIODIC, callback = clock_move)
23
24 while True:
25     if SW1.value() == 1 and SW2.value() == 0:
26         for step in sequence:
27             for i in range(len(pins)):
28                 pins[i].value(step[3-i])
29             utime.sleep_ms(2)
30     elif SW1.value() == 0 and SW2.value() == 1:
31         for step in sequence:
32             for i in range(len(pins)):
33                 pins[i].value(step[i])
34             utime.sleep_ms(2)
35     else:
36         for step in sequence:
37             for i in range(len(pins)):
38                 pins[i].value(0)
```

## プログラム説明

本プログラムの役割ブロックごとに説明をしていきます。

```
1 from machine import Pin,Timer
2 import utime
```

1～2 行目で動作に必要なモジュール(ライブラリ)の読み込みを行います。

1 行目は、machine モジュール内の I/O ピン制御を使うため、クラス Pin をロードする。

また、同モジュール内のタイマ割り込みも使うため、クラス Timer もロードする。

2 行目は、時計の要となる utime モジュールをロードする。

```
4 IN1 = Pin(2, Pin.OUT)
5 IN2 = Pin(3, Pin.OUT)
6 IN3 = Pin(5, Pin.OUT)
7 IN4 = Pin(4, Pin.OUT)
8 SW1 = Pin(14, Pin.IN, Pin.PULL_UP)
9 SW2 = Pin(15, Pin.IN, Pin.PULL_UP)
10 pins = [IN1, IN2, IN3, IN4]
11 sequence = [[1,1,0,0],[0,1,1,0],[0,0,1,1],[1,0,0,1]]
```

4～7 行目で先ほど読み込んだクラス Pin を用いて、2～5pin を出力設定にし、

8～9 行目で、14pin と 15pin を入力設定にする。

10 行目は、変数 IN1～4 を後に繰り返し処理しやすいように配列としてまとめている。

11 行目は、励磁させるコイルごとの High/Low の組み合わせを配列としてまとめている。

```
13 def clock_move(timer):
14     for step in sequence:
15         for i in range(len(pins)):
16             pins[i].value(step[3-i])
17             utime.sleep_ms(8)
18         for step in sequence:
19             for i in range(len(pins)):
20                 pins[i].value(0)
21
22 Timer().init(freq = 8.53325, mode = Timer.PERIODIC, callback = clock_move)
```

13～22 行目はステッピングモータの正転動作を制御しているプログラムとなる。

13～20 行目で、clock\_move 関数を定義しており、22 行目で 8.53325Hz ごとに clock\_move 関数をタイマ割り込みで呼び出すための設定となる。関数の詳細は以下の通りとなる。

1. for 文が開始されると、まず step に配列 sequence の最初の要素である[1,1,0,0]が入る。
2. その後、中の for 文が開始、i に 0 が入り、pins[0] = IN1、step[3] = 0 のため IN1=0 となる。  
繰り返しが終わると、IN1=0、IN2=0、IN3=1、IN4=1 という組み合わせになる。
3. 最後に 5ms スリープが入り、配列 sequence の次の要素に進み、要素数分1～2を繰り返す。
4. 18～20 行目の for 文はすべて Low 出力として、時間が進まない休み区間となる。(モータの発熱防止のため)

```

24 while True:
25     if SW1.value() == 1 and SW2.value() == 0:
26         for step in sequence:
27             for i in range(len(pins)):
28                 pins[i].value(step[3-i])
29                 utime.sleep_ms(2)
30     elif SW1.value() == 0 and SW2.value() == 1:
31         for step in sequence:
32             for i in range(len(pins)):
33                 pins[i].value(step[i])
34                 utime.sleep_ms(2)
35     else:
36         for step in sequence:
37             for i in range(len(pins)):
38                 pins[i].value(0)

```

24～38 行目は、時刻調整用のプログラム部分となる。

14pin と 15pin に接続しているスイッチ ON/OFF の組み合わせによって、ステッピングモータを任意の方向に通常動作より速く回転させ、時刻の調整ができる。

【14pin＝ON、15pin＝OFF (25～29 行目)】

14pin のスイッチを押している間、ステッピングモータが正転(時間が進む)方向に回る。

for 文の制御部分に関しては、前ページにある clock\_move 関数のスリープ時間を短くしたもので、説明は割愛する。

【14pin＝OFF、15pin＝ON (30～34 行目)】

15pin のスイッチを押している間、ステッピングモータが逆転(時間が戻る)方向に回る。

基本的な流れは clock\_move 関数と同じだが、逆方向に回転させるため 33 行目の value の引数を step[i]に変更している。

【14pin＝OFF、15pin＝OFF or 14pin＝ON、15pin＝ON (35～38 行目)】

14pin と 15pin のスイッチがどちらも押されていない場合、もしくは両方とも押している間、ステッピングモータは停止する。(モータの発熱防止のため)

## 最後に

いかがでしたでしょうか。問題なく動かすことは出来ましたか？

拙い説明で分かりづらい部分はあったかと思いますが、楽しんでもらえていたら嬉しいです。

その他にも動かす方法はたくさんあると思いますので、色々と試してみてください。

それでは、またどこかで。一緒に楽しい電子工作ライフを歩みましょう！

以上