

Forum:[Hack Forums](#)

Thread: [Half step sequence with Steppermotor](#)

SIM-1517404 - September 27, 2017, 3:05 pm

I thought I had the sequencing figured out for the stepper motor, but I was brutally wrong.

I'm running a unipolar 6-wire stepper motor with RPi 3 model b and Python. The code I'm running is:

```
Code import RPi.GPIO as GPIO
import time
```

```
# Variables
```

```
delay = 0.0055
steps = 200
```

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
```

```
# Enable pins for IN1-4 to control step sequence
```

```
coil_A_1_pin = 18
coil_A_2_pin = 23
coil_B_1_pin = 24
coil_B_2_pin = 25
```

```
# Set pin states
```

```
GPIO.setup(coil_A_1_pin, GPIO.OUT)
GPIO.setup(coil_A_2_pin, GPIO.OUT)
GPIO.setup(coil_B_1_pin, GPIO.OUT)
GPIO.setup(coil_B_2_pin, GPIO.OUT)
```

```
# Function for step sequence
```

```
def set_step(w1, w2, w3, w4):
    GPIO.output(coil_A_1_pin, w1)
    GPIO.output(coil_A_2_pin, w2)
    GPIO.output(coil_B_1_pin, w3)
    GPIO.output(coil_B_2_pin, w4)
```

```
# loop through step sequence based on number of steps
```

```
# gives us a list so we can essentially iterate through each of the different step values
# meaning we can iterate through each "part" of the original stepping sequence
step_values = [{"w1":1, "w2":0, "w3":1, "w4":0},
               {"w1":0, "w2":1, "w3":1, "w4":0},
```

```
{"w1":0, "w2":1, "w3":0, "w4":1},  
{"w1":1, "w2":0, "w3":0, "w4":1}]
```

```
step_values1 = [{"w1":1, "w2":0, "w3":0, "w4":0},  
                {"w1":1, "w2":1, "w3":0, "w4":0},  
                {"w1":0, "w2":1, "w3":0, "w4":0},  
                {"w1":0, "w2":1, "w3":1, "w4":0},  
                {"w1":0, "w2":0, "w3":1, "w4":0},  
                {"w1":0, "w2":0, "w3":1, "w4":1},  
                {"w1":0, "w2":0, "w3":0, "w4":1},  
                {"w1":1, "w2":0, "w3":0, "w4":1}]
```

```
step_values2 = [{"w1":1, "w2":0, "w3":1, "w4":0},  
                {"w1":1, "w2":0, "w3":0, "w4":0},  
                {"w1":1, "w2":0, "w3":0, "w4":1},  
                {"w1":0, "w2":1, "w3":0, "w4":1},  
                {"w1":0, "w2":1, "w3":0, "w4":1},  
                {"w1":0, "w2":1, "w3":0, "w4":0},  
                {"w1":0, "w2":1, "w3":1, "w4":0},  
                {"w1":0, "w2":0, "w3":1, "w4":0}]
```

```
current_step_count = 0
```

```
def step_motor():
```

```
    global current_step_count
```

```
    needed_step_values = step_values[current_step_count % 4]
```

```
    set_step(needed_step_values["w1"], needed_step_values["w2"], needed_step_values["w3"],
```

```
needed_step_values["w4"])
```

```
    time.sleep(delay)
```

```
    current_step_count += 1
```

```
for i in range(steps):
```

```
    step_motor()
```

```
set_step(0,0,0,0)
```

step_values works, it rotates one full rotation, but it's a full step sequence.

I've tried the "standard" step sequence, as shown in step_values1, and another I found, as shown in step_values2.

When I run the step_values1 it does:

- Nothing happens for a brief moment
- One full step clockwise
- One half step counter-clockwise
- One full step counter-clockwise

and repeat.

When I run the step_values2 it does:

- One full step clockwise
- One half step counter-clockwise
- One half step counter-clockwise
- One full step counter-clockwise

and repeat.

Also wondering what the code for 1/8 microstep would look like, if anyone knows how to deal with microstepping.

Any help is much appreciated.