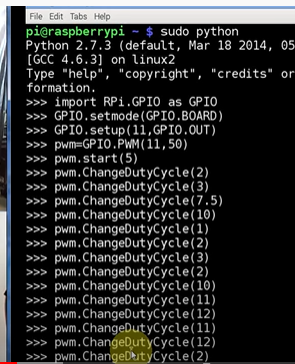
Tweaking to get the correct 0, 90 and 180 degrees

Duty Cycle is a percentage



Check the arm of the servo motor to see if the duty cycle setting corresponds to the 0, 90 and 180 position.

Final conclusion from my test is as follows:

pwm.ChangeDutyCycle(2) 0 degree

pwm.ChangeDutyCycle(12) 180 degree

pwm.ChangeDutyCycle(6.5) 90 degree

One way to stop jittering is to apply ChangeDutyCycle(0) after the arm has moved to the desired angle. Or pwm.stop()

Range 2 -> 12 % duty cycle represents 0 -> 180 degrees

To calculate duty cycle based on required angle

Remember slope of a line will be:

m=(y2-y1)/(x2-x1)=(12-2)/180-0)=10/180 = 1/18

We can now get the equation of the line using the point slope formula.

y-y1=m(x-x1)

y-2=1/18\*(x-0)

y = 1/18\*x + 2

Putting in our actual variables, we get

DutyCycle = 1/18\* (DesiredAngle) + **2 (+2 because left most duty cycle is 2%)**

DC of 90 degrees = ((1/18) \* 90 ) + 2

DC of 0 degrees = ((1/18) \* 0) + 2

https://toptechboy.com/raspberry-pi-lesson-28-controlling-a-servo-on-raspberry-pi-with-python/

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ratio 1/18 | Angle |  | Ratio \* Angle | Freq + 2 |
| 0.055556 | 0 |  | 0.00 | 2.0 |
| 0.055556 | 10 |  | 0.56 | 2.6 |
| 0.055556 | 20 |  | 1.11 | 3.1 |
| 0.055556 | 30 |  | 1.67 | 3.7 |
| 0.055556 | 40 |  | 2.22 | 4.2 |
| 0.055556 | 50 |  | 2.78 | 4.8 |
| 0.055556 | 60 |  | 3.33 | 5.3 |
| 0.055556 | 70 |  | 3.89 | 5.9 |
| 0.055556 | 80 |  | 4.44 | 6.4 |
| 0.055556 | 90 |  | 5.00 | 7.0 |
| 0.055556 | 100 |  | 5.56 | 7.6 |
| 0.055556 | 110 |  | 6.11 | 8.1 |
| 0.055556 | 120 |  | 6.67 | 8.7 |
| 0.055556 | 130 |  | 7.22 | 9.2 |
| 0.055556 | 140 |  | 7.78 | 9.8 |
| 0.055556 | 150 |  | 8.33 | 10.3 |
| 0.055556 | 160 |  | 8.89 | 10.9 |
| 0.055556 | 170 |  | 9.44 | 11.4 |
| 0.055556 | 180 |  | 10.00 | 12.0 |