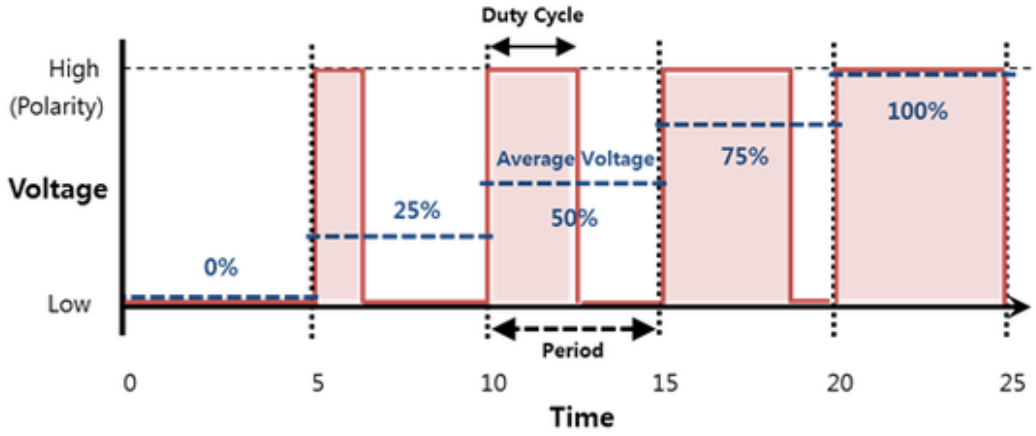
Pertemuan 05
Pulse Width Modulation (PWM)

Dasar Teori PWM

Bentuk digital Pulse Width Modulation (PWM; juga ditulis sebagai modulasi lebar pulsa) biasanya digunakan untuk menggerakkan beban berat seperti motor, aktuator, pemanas, dan sebagainya. PWM pada dasarnya adalah gelombang persegi positif yang lebar pulsanya dapat diubah. Dengan mengubah lebar pulsa, kita dapat secara efektif mengubah nilai rata-rata tegangan yang disuplai ke beban. Gambar 1 menunjukkan contoh PWM



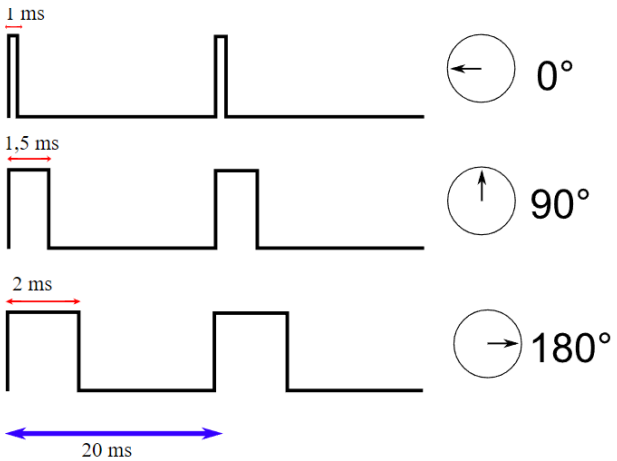
Gambar 1. Contoh PWM

Praktik 1 (Servo SG90)

Sesuai dengan datasheet servo SG90, untuk dapat bekerja, servo membutuhkan sinyal PWM dengan ketentuan sebagai berikut :

- Rotational Range: 180°
- Pulse Cycle: ca. 20 ms
- Pulse Width: 500-2400 µs

Dimana, kalau digambarkan menggunakan grafik maka akan seperti grafik yang ditunjukkan oleh Gambar 2.



Gambar 2. Duty cycle dan sudut pada motor servo

Untuk itu, perlu dicari nilai konversi dari satuan waktu (ms) ke nilai 16 bit untuk dapat menggerakkan servo ke sudut yang diinginkan. Caranya seperti berikut ini :

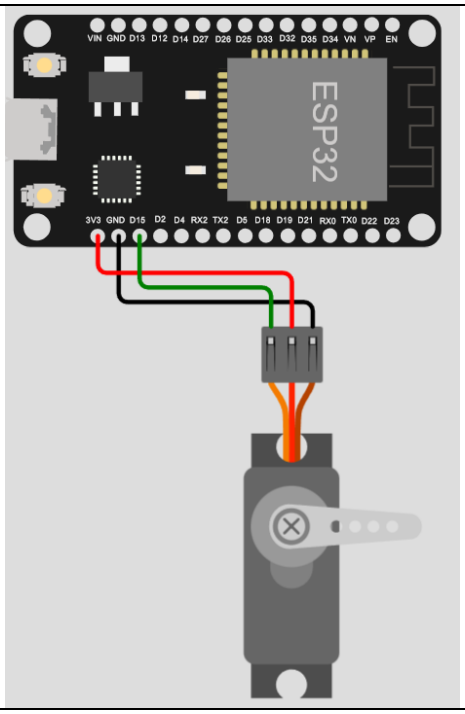
- Time (t) = 1 detik = 1000 ms
- Untuk mendapatkan pulse cycle (pc) sebesar 20 ms maka dibutuhkan frekuensi (freq) sebesar 50 Hz, perhitungannya :

  pc = t / freq
  pc = 1000 / 50
  pc = 20 ms

Praktik 1 (Servo SG90)

- Untuk sudut 0° membutuhkan pulse width (pw) sebesar 500 µs, maka duty cycle (dc) nya sebesar :
  dc = pw / pc
  dc = 500 µs / 20 ms = 500 µs / 20000 µs
  dc = 0.025 = 2.5%
- Untuk sudut 180° membutuhkan pulse width (pw) sebesar 2400 µs, maka duty cycle (dc) nya sebesar :
  dc = pw / pc
  dc = 2400 µs / 20 ms = 2400 µs / 20000 µs
  dc = 0.12 = 12%
- Merubah nilai duty cycle (dc) ke integer 16-bit
  - Sudut 0° = duty cycle 2.5% = 0.0250*65536 = 1638
  - Sudut 180° = duty cycle 12% = 0.1200*65536 = 7864

```
from machine import Pin, PWM
from utime import sleep_ms

servo = PWM(Pin(15))
servo.freq(50)
servo.duty_u16(1638)

while True:
    servo.duty_u16(1638)
    sleep_ms(1000)
    servo.duty_u16(7864)
    sleep_ms(1000)
```



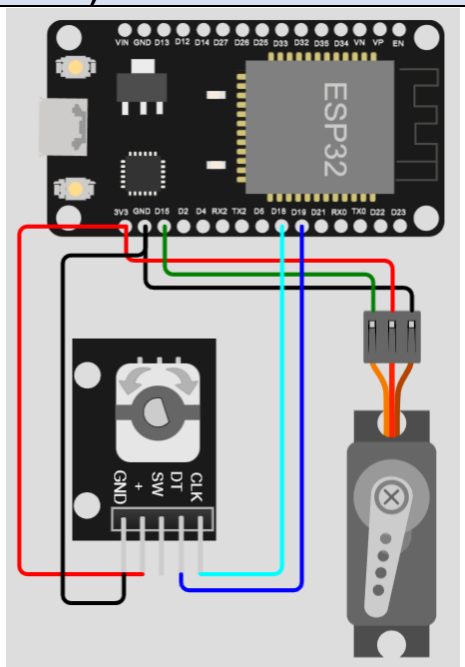
Praktik 2 (Servo, Rotary Encoder)

Download dua buah library rotary encoder dari link berikut ini

1. https://github.com/miketeachman/micropython-rotary/blob/master/rotary.py
2. https://github.com/miketeachman/micropython-rotary/blob/master/rotary_irq_esp.py

Kemudian upload kedua buah library tersebut ke ESP32 dan tuliskan kode berikut ini pada file main.py

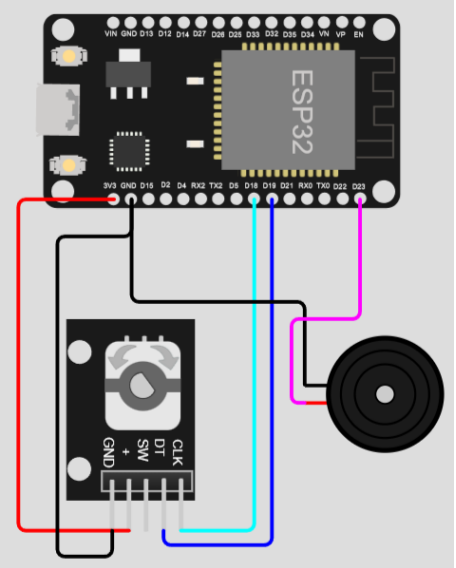
```
from machine import Pin, PWM
from utime import sleep_ms
from rotary_irq_esp import RotaryIRQ

start = 1638
servo = PWM(Pin(15))
servo.freq(50)
servo.duty_u16(start)

r = RotaryIRQ(pin_num_clk=18,
              pin_num_dt=19,
```



```
                min_val=0,
                max_val=18,
                reverse=False,
                range_mode=RotaryIRQ.RANGE_WRAP)
val_old = r.value()

while True:
    val_new = r.value()
    servo.duty_u16(val_new*346+start)
    if val_new != val_old:
        val_old = val_new
        print("value",val_new)
    sleep_ms(50)
```

Praktik 3 (Servo, Buzzer)

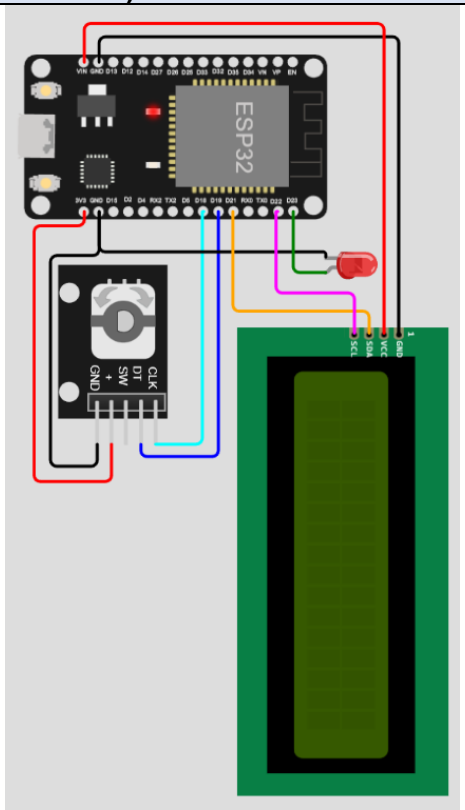
```
from machine import Pin, PWM
from utime import sleep_ms
from rotary_irq_esp import RotaryIRQ

buzzer = PWM(Pin(23))
r = RotaryIRQ(pin_num_clk=18,
                pin_num_dt=19,
                min_val=0,
                max_val=18,
                reverse=True,
                range_mode=RotaryIRQ.RANGE_WRAP)
val_old = r.value()

while True:
    val_new = r.value()
    if val_new != val_old:
        val_old = val_new
        print("value",val_new)
        buzzer.freq(val_new*100+100)
        buzzer.duty_u16(1000)
        sleep_ms(100)
    buzzer.duty_u16(0)
    sleep_ms(50)
```



Praktik 4 (Servo, LED, LCD 1602 I2C)

Download library LCD 1602 I2C dari link berikut ini :

- https://github.com/micropython-Chinese-Community/mpy-lib/blob/master/lcd/I2C_LCD1602/mp_i2c_lcd1602.py
- Lakukan sedikit perubahan pada file yang anda download pada baris ke-30 menjadi :
  `self.pb = bytearray(' '*16,'utf-8')`




```
from machine import Pin, PWM, I2C
from utime import sleep_ms
from rotary_irq_esp import RotaryIRQ
from mp_i2c_lcd1602 import LCD1602

i2c = I2C(1, sda=Pin(21), scl=Pin(22))
lcd = LCD1602(i2c)

led = PWM(Pin(23))
led.freq(50)

r = RotaryIRQ(pin_num_clk=18,
              pin_num_dt=19,
              min_val=0,
              max_val=18,
              reverse=True,
              range_mode=RotaryIRQ.RANGE_WRAP)
val_old = r.value()

lcd.puts("int cahaya LED")
while True:
    val_new = r.value()
    if val_new != val_old:
        val_old = val_new
        led.duty_u16(val_new*3640)
        lcd.clear()
        lcd.puts("int cahaya LED")
        lcd.puts(val_new,0,1)
    sleep_ms(50)
```

Praktik 5 (Gabungan)	
Gabungkan praktik 1,2,3 dan 4 sehingga menjadi satu sistem IoT seperti gambar berikut ini