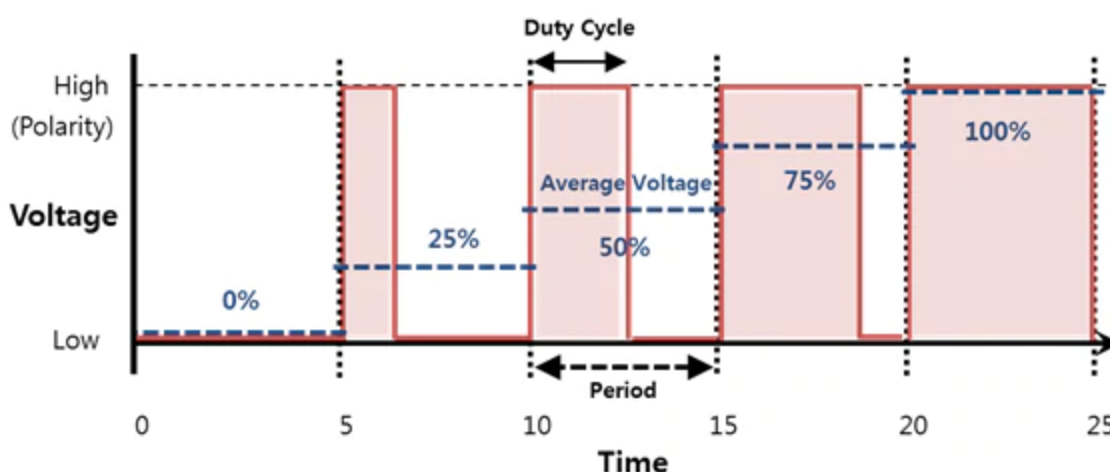


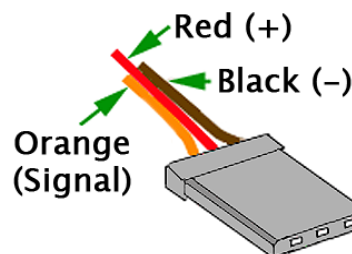
Wprowadzenie. PWM jest techniką generowania sygnału pseudo analogowego za pomocą linii cyfrowych. Na wybranym wyjściu cyfrowym generowany jest przebieg prostokątny o wartości maksymalnej (on) równej 3.3V i wartości minimalnej (off) 0V. W przypadku odpowiednio dużej częstotliwości przełączeń taki sygnał jest odbierany przez urządzenie zewnętrzne tak jak sygnał stały. Współczynnik wypełnienia równy $(\text{czas_on})/(\text{czas_on} + \text{czas_off}) * 100\%$ jest proporcjonalny do średniej wartości napięcia.



Moduł Raspberry Pi Pico jest zdolny do generowania sygnałów PWM na dowolnym GPIO. Najniższa, dostępna, częstotliwość sygnału prostokątnego wynosi 10[Hz].

Zad.1. Napisz skrypt w MicroPython do sterowania serwomechanizmem SG-90.

moduł Pico	serwomechanizm SG-90
Gnd	brązowy
VSYS	czerwony
GP15	pomarańczowy/żółty



Ćwiczenie nr 3: PWM

```
from machine import Pin, PWM

from utime import sleep

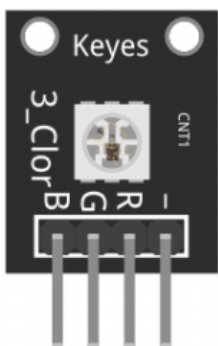
MIN_POS = 2500
MAX_POS = 9000

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

#angle from 0 to 180
def set_servo_pos(shaft_position):
    servo.duty_u16(shaft_position)
    sleep(0.2)

while True:
    for position in range(MIN_POS, MAX_POS, 50):
        set_servo_pos(position)
    for position in range(MAX_POS, MIN_POS, -50):
        set_servo_pos(position)
```

Zad.2. Podłącz moduł RGB KY-009 do płytki Raspberry Pi Pico. W celu uniknięcia przegrzania LED wchodzących w skład modułu należy zastosować rezystory ograniczające prąd 100R. Rezystory powinny być włączone (szeregowo) pomiędzy wyprowadzenia GPIO na płytce Pico a piny R,G,B modułu Keyes KY-009.



RPi Pico	KY-009	
GP2	B	składowa niebieska
GP3	G	składowa zielona
GP4	R	składowa czerwona
Gnd	-	katoda

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

#Pins
R_pin = PWM(Pin(2))
G_pin = PWM(Pin(3))
B_pin = PWM(Pin(4))

while True:
    for intensity in range(0, 65000,5000):
        R_pin.duty_u16(intensity)
        sleep(1)
```

Zad.3. Wykorzystaj układ z zadania 2. Napisz skrypt w MicroPython, który pozwoli na sterowanie pozycją serwomechanizmu za pomocą joystick'a. Kąt obrotu wału powinien być proporcjonalny do wychylenia drążka, w pełnym zakresie ruchu. Naciśnięcie przycisku joysticka powinna "zamrażać" bieżącą pozycję.

Zad.4. Napisz skrypt w MicroPython, który pozwoli na sterowanie położeniem wału serwomechanizmu za pomocą 2 przycisków. Naciśnięcie 1 z przycisków powinno wymuszać obrót zgodnie(CW) z kierunkiem ruchu wskazówek zegara, a drugiego w kierunku przeciwnym do kierunku ruchu wskazówek zegara (CCW).

Zad.5. Zbuduj prosty interfejs użytkownika (HMI), składający się z joysticka i 3 przycisków do kontroli koloru światła emitowanego przez moduł RGB.

Zad.6. Podłącz wyświetlacz Pico-LCD-0.96 lub Pico-LCD-1.8 do płytki Raspberry Pi Pico.



Pobierz *Demo code* z sekcji *Resources* na odpowiedniej stronie:

- www.waveshare.com/wiki/Pico-LCD-0.96
- www.waveshare.com/wiki/Pico-LCD-1.8

Dokonaj analizy kodu, pod kątem nazw przydatnych metod i funkcji. Stwórz skrypt w MicroPython, który pozwoli na rysowanie dowolnego kształtu na ekranie przy pomocy joysticka. Wykorzystaj przycisk SW do zmiany koloru.

Dla zainteresowanych:

1. MicroPython - strona:

micropython.org/download/rp2-pico/

2. Sterowanie serwomechanizmem przy zastosowaniu techniki PWM w MicroPython:

circuitdigest.com/microcontroller-projects/control-a-servo-motor-with-raspberry-pi-pico-using-pwm-in-micropython