Introduction. Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on(HIGH = 3.3V) and off(LOW = 0V). This on-off pattern can simulate voltages by changing the portion of the time the signal spends on versus the time that the signal spends off. The duty cycle is proportional to the average *(time_high/(time_high + time_low) * 100%)* voltage on the selected PWM pin.
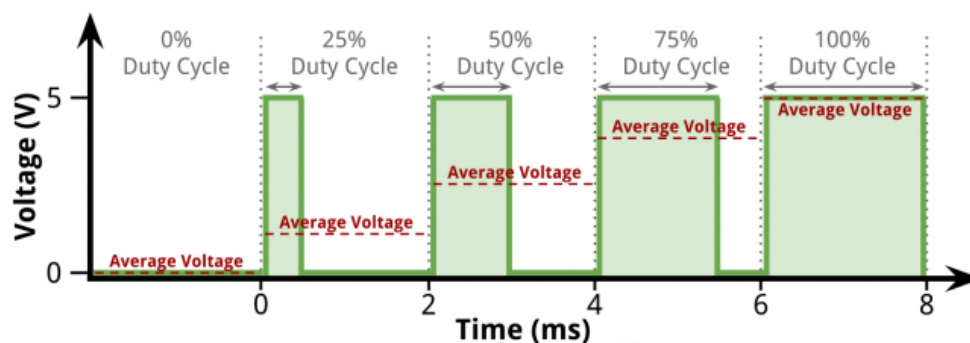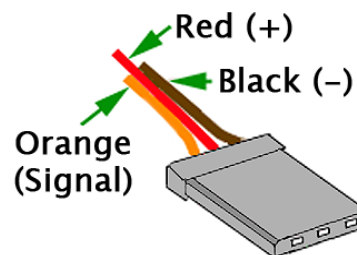

robotic-controls.com

Raspberry Pi Pico has PWM capability on all GPIO pins. When a GPIO pin is set to PWM mode, the output of each pin varies with the frequency cycle. The lowest PWM frequency output is 10[Hz].

Task.1. Create a MicroPython script to control the SG90 servomechanism.

Pico board	Servo
Gnd	brown
VSYS	red
GP15	orange


```
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)
```

Task.2. Connect an RGB module to the Raspberry Pi Pico board. You need to use 100R resistors to prevent LED burnout. Those resistors should be connected between Pico board GPIOs and the KY-009 board R, G, and B pins.
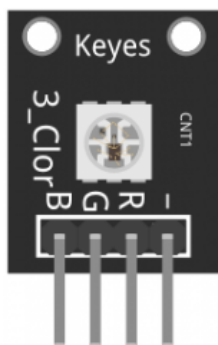
Arduino board	KY-009 board	
GP2	B	blue component
GP3	G	green component
GP4	R	red component
Gnd	-	cathode

```
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)
```

Task.4. Use the circuit from Task 1. The position of the servo shaft should be set according to the joystick inclination in the Y axis, in the full range of motion. Pressing the joystick button should freeze the current shaft angle.

Task.5. Write a program to control the servomechanism using two buttons. Pressing one button forces CW rotation. The other button makes the servo rotate CCW.

Task.6. Create a simple HMI consisting of a joystick and 3 buttons for controlling the color of light emitted from the RGB module.

Task.7. Connect Pico-LCD-0.96 module.

Open the *Demo code* from the *Resources* section on the website:

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

Analyze the code. Create a MicroPython script to draw a shape on the display with the joystick. Use the SW button as a tool for changing the current pixel color.

For those interested:

1. MicroPython web page:

<p align="center">micropython.org/download/rp2-pico/</p>

2. Control a Servo Motor with Raspberry Pi Pico Using PWM in MicroPython tutorial:

<p align="center">circuitdigest.com/microcontroller-projects/control-a-servo-motor-with-raspberry-pi-pico-using-pwm-in-micropython</p>