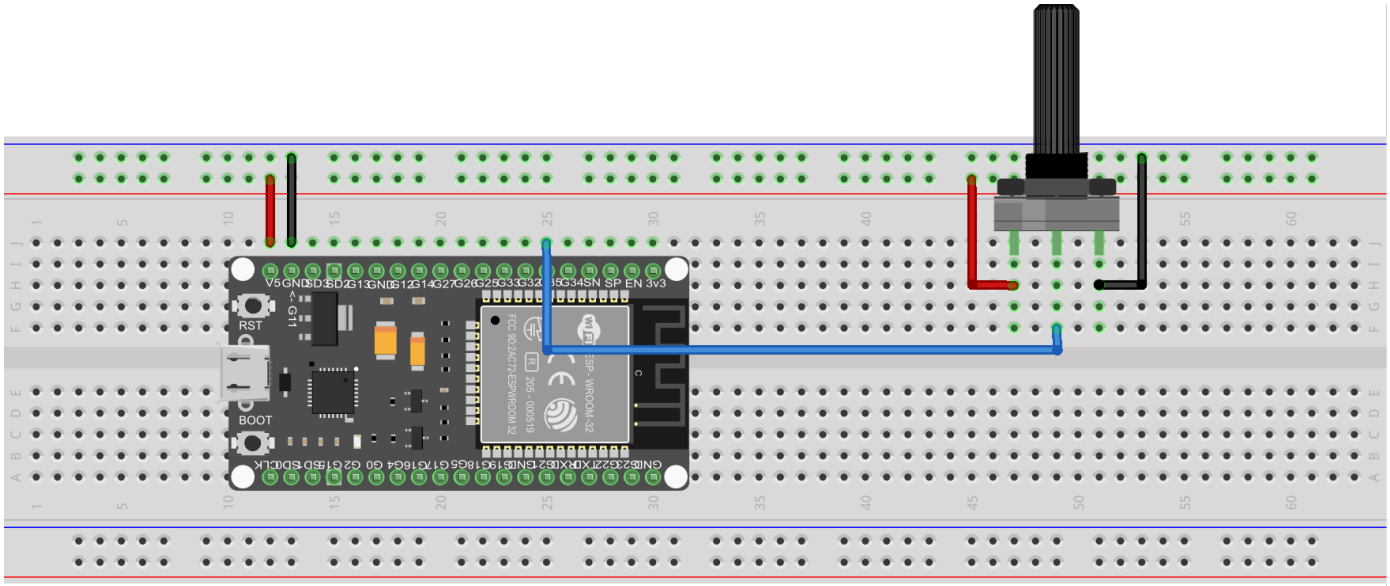
ESP32 Lab Assignment III guide

Problem

Using a potentiometer and esp32 board, control the brightness of one led by adjusting the potentiometer.

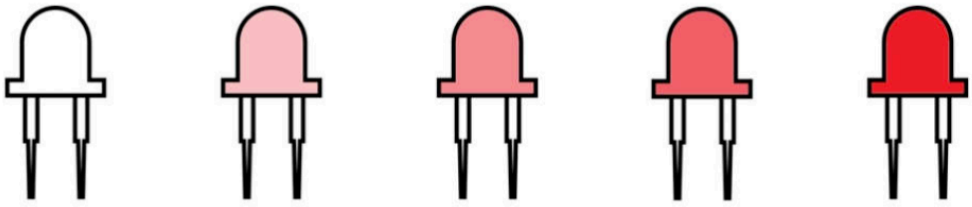
Circuit



PWM (pulse width modulation)

The ESP32/ESP8266 GPIOs can be set either to output 0V or 3.3V, but they can't output any voltages in between (apart from the DAC pins on the ESP32). However, you can output "fake" mid-level voltages using pulse-width modulation (PWM), which is how you'll produce varying levels of LED brightness for this project.

If you alternate an LED's voltage between HIGH and LOW very fast, your eyes can't keep up with the speed at which the LED switches on and off; you'll simply see some gradations in brightness.



That's basically how PWM works -- by producing an output that changes between HIGH and LOW at a very higy frequency. The duty cycle is the fraction of the time period at which the LED is set to HIGH. The following figure illustrates how PWM works.

works.

For instance, a duty cycle of 50 percent results in 50 percent LED brightness, a duty cycle of 0 means the LED is fully off, and a duty cycle of 100 means the LED is fully on. Changing the duty cycle is how you produce different levels of brightness, and that's what we're going to do here.

How the code works

To create a PWM pin, import the *PWM* class from the *machine* module.

```
from machine import Pin, PWM
```

Then, create a PWM object called *led*.

```
led = PWM(Pin(5), frequency)
```

To create a PWM object, you need to pass as parameters, the pin it is connected to, the signal frequency and the duty cycle.

Frequency: The frequency can be a value between 0 and 78125. A frequency of 5000 Hz can be used to control an LED brightness.

Duty cycle: The duty cycle can be a value between 0 and 1023. In which 1023 corresponds to 100% duty cycle (full brightness), and 0 corresponds to 0% duty cycle (unlit LED).

If you don't set the duty cycle when instantiating the PWM object, it will be 0 by default.

To set the duty cycle use the *duty()* method on PWM object and pass the duty cycle as an argument:

```
led.duty(duty_cycle)
```