

Module 6.4: Controlling Multiple LEDs

Available digitally at: sites.google.com/ahschools.us/ahsmakerspace

This module will show you how to control multiple LED in a variety of ways.

You will be introduced to these components and concepts:

- A. PWM
- B. Pulse
- C. LED Board

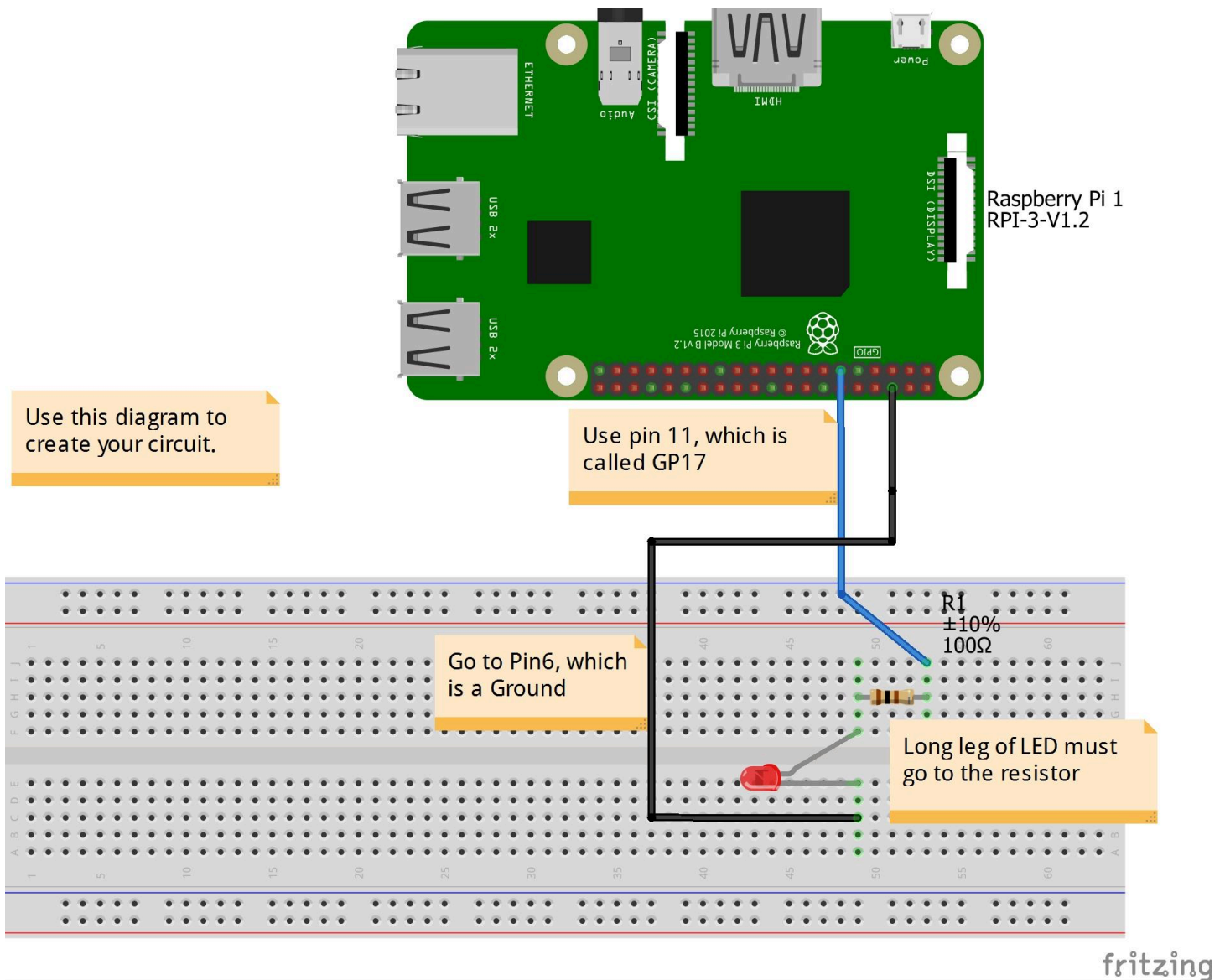
Use [Module 6.2](#) as reference if you need help with resistors, LEDs, jumper cables, and GPIO pins.

PART 1: Get your RPi Ready.



PART 2: Build it.

1. Build a circuit on your breadboard to match this:



- Now we need to write our program for the button.... Start with a new file by clicking **File > New file**.
- Save the new file by clicking **File > Save**. Save the file as **gpio_pwmled.py**
- This time you'll need the PWMLED class. Write the following code in your new file:

```
from gpiozero import PWMLED
from time import sleep

led = PWMLED(17)

while True:
    led.value = 0 # off
    sleep(1)
    led.value = 0.5 # half brightness
    sleep(1)
    led.value = 1 # full brightness
    sleep(1)
```

5. Save your program.
 6. Run the code.
 7. The LED should progress through off, half, and full brightness
 - a. YES! Nice work, move on to Part 3
 - b. NO! Troubleshoot this sucker. Double check your code, your circuit, get a second set of eyes... or a third. Make it happen!
-

PART 3: Pulse

8. Alter your code to the following:

```
from gpiozero import PWMLED
from signal import pause

led = PWMLED(17)

led.pulse()

pause()
```

9. Save this program as **pulse_LED.py**
 10. Run your program and see if the LED pulses.
 - a. YES! Nice work, move on to Part 4
 - b. NO! Troubleshoot this sucker. Double check your code, your circuit, get a second set of eyes... or a third. Make it happen!
-

PART 4: Multiple LEDs

11. Now we will write program that can control multiple LEDs in a LEDBoard. Use this code in a new file, saving it as **LED_Board.py**

```
from gpiozero import LEDBoard
from time import sleep
from signal import pause

leds = LEDBoard(5, 6, 13, 19, 26)

leds.on()
sleep(1)
leds.off()
sleep(1)
leds.value = (1, 0, 1, 0, 1)
sleep(1)
leds.blink()

pause()
```

12. Using the code above as your guide, try putting the circuit together on your own with these helpful hints:
- There are 5 LEDs.
 - Each LED must be put in series with a resistor of 50 Ohms or greater.
 - Each LED must be grounded.
 - Each LED is controlled by a different GPIO pin (which is numbered in the code)

Done? [Check your circuit](#) with this diagram here.

13. Save, Run, and try it out.
- It works! I push the button and the LED turns on, then turns off when I push it again. Now go on to Part 5.
 - NOPE!!! Troubleshoot it.

PART 5: Changing Multiple LED Brightness

14. By incorporating PWM ([learn about Pulse Width Modification here](#)) we can control the brightness of the individual LEDs in the LEDBoard. Use this code with the circuit you built in Part 4:

```
from gpiozero import LEDBoard
from signal import pause

leds = LEDBoard(5, 6, 13, 19, 26, pwm=True)

leds.value = (0.2, 0.4, 0.6, 0.8, 1.0)

pause()
```

15. Save as **PWM_LED**, Run, and try it out.
- It works! Try changing the numbers so you can understand the connection between the value and what the LED looks like. Then go on to Part 6.
 - NOPE!!! Troubleshoot it.

PART 6: LED Bar Graph

16. A collection of LEDs can be treated like a bar graph. Using the same circuit you built in Part 5 but with the code below give it a shot! Save this program as **LED_Bar_Graph**

```
from gpiozero import LEDBarGraph
from time import sleep

graph = LEDBarGraph(5, 6, 13, 19, 26, pwm=True)

graph.value = 1/10 # (0.5, 0, 0, 0, 0)
sleep(1)
graph.value = 3/10 # (1, 0.5, 0, 0, 0)
sleep(1)
```

```
graph.value = -3/10 # (0, 0, 0, 0.5, 1)
sleep(1)
graph.value = 9/10 # (1, 1, 1, 1, 0.5)
sleep(1)
graph.value = 95/100 # (1, 1, 1, 1, 0.75)
sleep(1)
```

17. Save and Run.

18. Notice the # signs in the code. Can you connect what is in the parenthesis after the # sign with what the LEDs are doing? The # is a comment and tells the computer to skip over whatever comes next on the line.

PART 7: Module 6.4 CHALLENGE

19. You are on your own for this one. Using the code below, build and operate a Traffic Light system.

```
from gpiozero import LED
from time import sleep

red = LED(2)
amber = LED(3)
green = LED(4)

green.on()
amber.off()
red.off()

while True:
    sleep(10)
    green.off()
    amber.on()
    sleep(1)
    amber.off()
    red.on()
    sleep(10)
    amber.on()
    sleep(1)
    green.on()
    amber.off()
    red.off()
```

20. With the same circuit you built, try this code:

```
from gpiozero import TrafficLights
from time import sleep
from signal import pause

lights = TrafficLights(2, 3, 4)
```

```
def traffic_light_sequence():
    while True:
        yield (0, 0, 1) # green
        sleep(10)
        yield (0, 1, 0) # amber
        sleep(1)
        yield (1, 0, 0) # red
        sleep(10)
        yield (1, 1, 0) # red+amber
        sleep(1)

lights.source = traffic_light_sequence()

pause()
```

21. **Are you starting to get the idea that there are always multiple ways to code the same thing?**
22. **Done? DO NOT TURN OFF POWER.** First, tell the RPi to Shutdown. Once shutdown is complete then power off all the hardware.