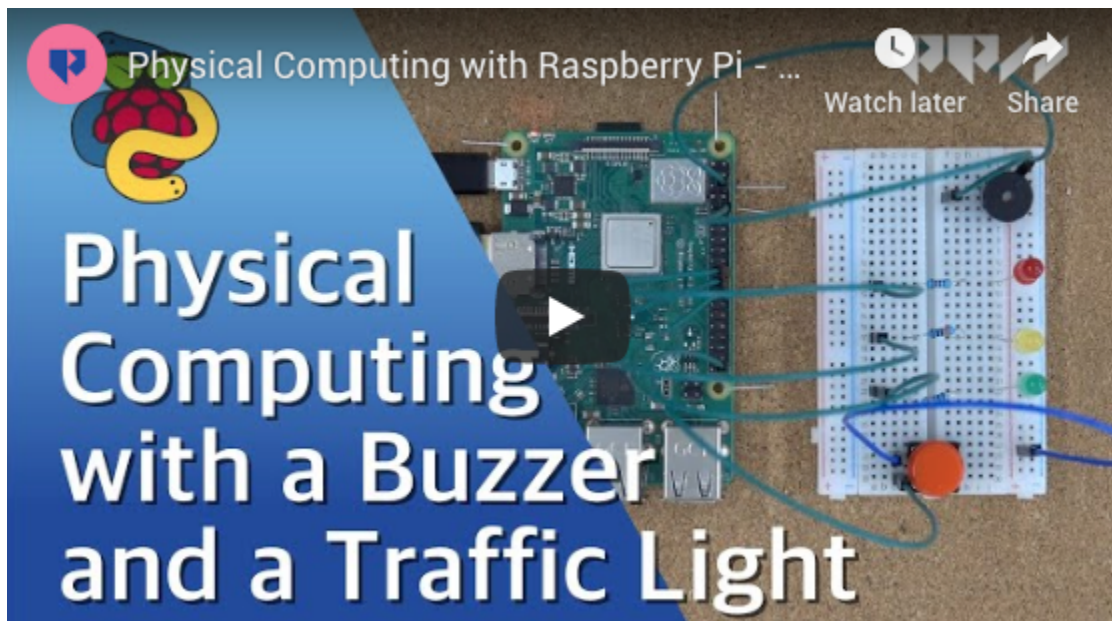


# Physical Computing with a Buzzer and a Traffic Light

<b>Physical Computing with a Buzzer and a Traffic Light</b>	<b>1</b>
<b>Project-1: What's the Buzz?</b>	<b>2</b>
<b>Project-1: What's the Buzz? - the code</b>	<b>4</b>
<b>Upgrade the Buzzer program</b>	<b>5</b>
<b>Project-2: Traffic Light!</b>	<b>5</b>
The hardware	6
<b>Project-2: Traffic Light - the hardware</b>	<b>7</b>
<b>Project-2: Traffic Light - the code</b>	<b>9</b>
<b>The Traffic Light Program</b>	<b>11</b>



We have created a button and light program, so let's add a bit of noise to the project. The Buzzer is another simple component that can work very similarly to the LED and Button. We will be able to wire up the buzzer and connect an LED and Button to create our own traffic light. These components have so many different applications in different projects.

## Materials

- Raspberry Pi Model 3 B+ 8GB

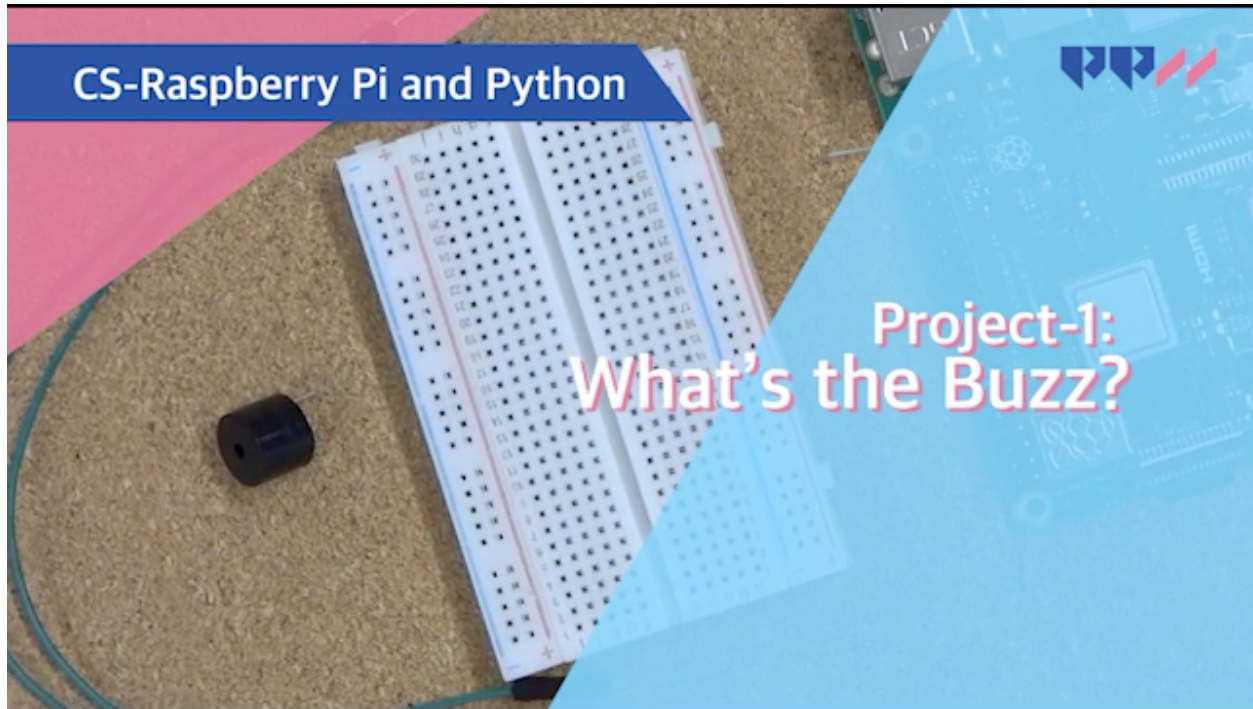
- Micro SD Card Buzzer
- Micro USB Power Supply
- Mouse
- 3 x LEDs
- 3 x 330ohm Resistor
- Jumper Wires
- HDMI Cord
- Keyboard
- HDMI Monitor
- Computer
- SD Card Reader

### **Key Concepts**

- GPIO
- Time
- Sleep
- Button
- LED
- Buzzer

## Project-1: What's the Buzz?

We are going to explore using the buzzer for our physical computing.

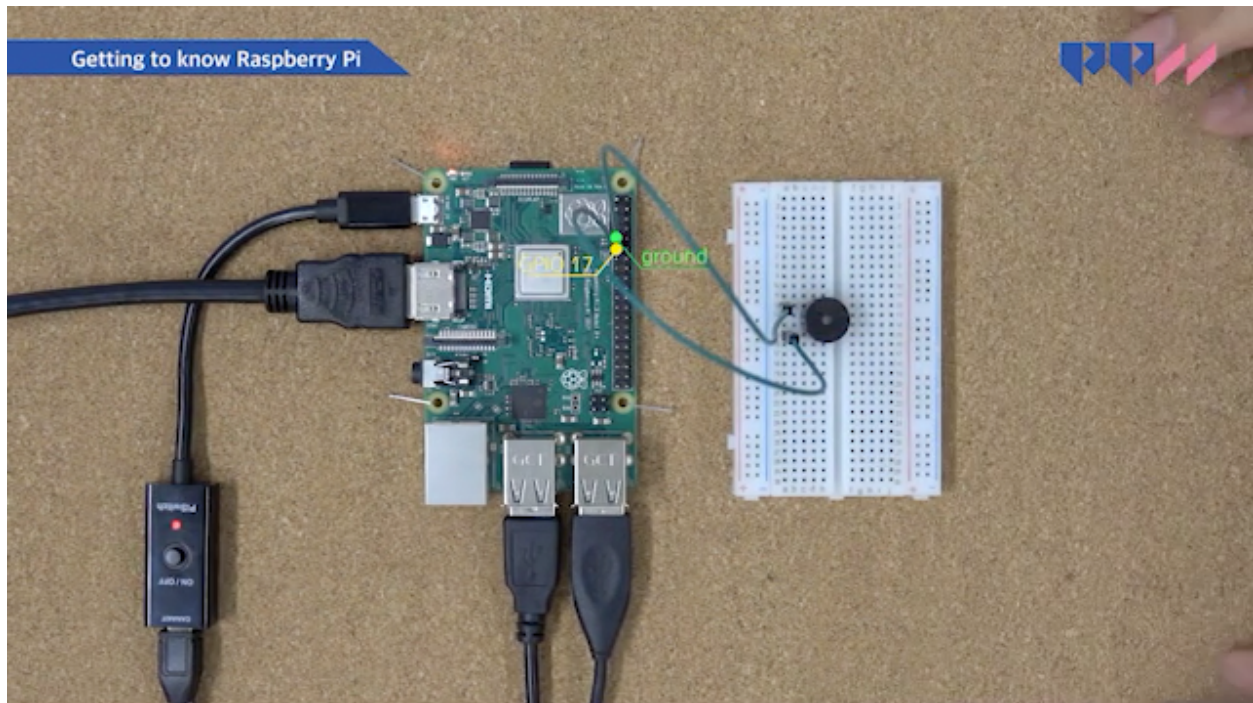


### Materials

- Breadboard
- Buzzer
- 2 male/female jumper wires

### The Hardware

The wiring for a buzzer is like the LED wiring minus the resistor. That's because the buzzer needs more power than the LED does. Hook your buzzer up like this,



---

## Project-1: What's the Buzz? - the code

Once you have everything wired, open a new tab in Thonny and copy this code,

```
from gpiozero import Buzzer
from time import sleep
```

```
buzzer = Buzzer(17)
```

```
buzzer.on()
```

This code works very much like the code for the LED. We pull the function from the library, we designate pin 17 as Buzzer and then we tell it to turn on. When you run this program, the buzzer will turn on and stay on until you tell it to turn off. Replace `buzzer.on` with `buzzer.off` to shut it down.

---

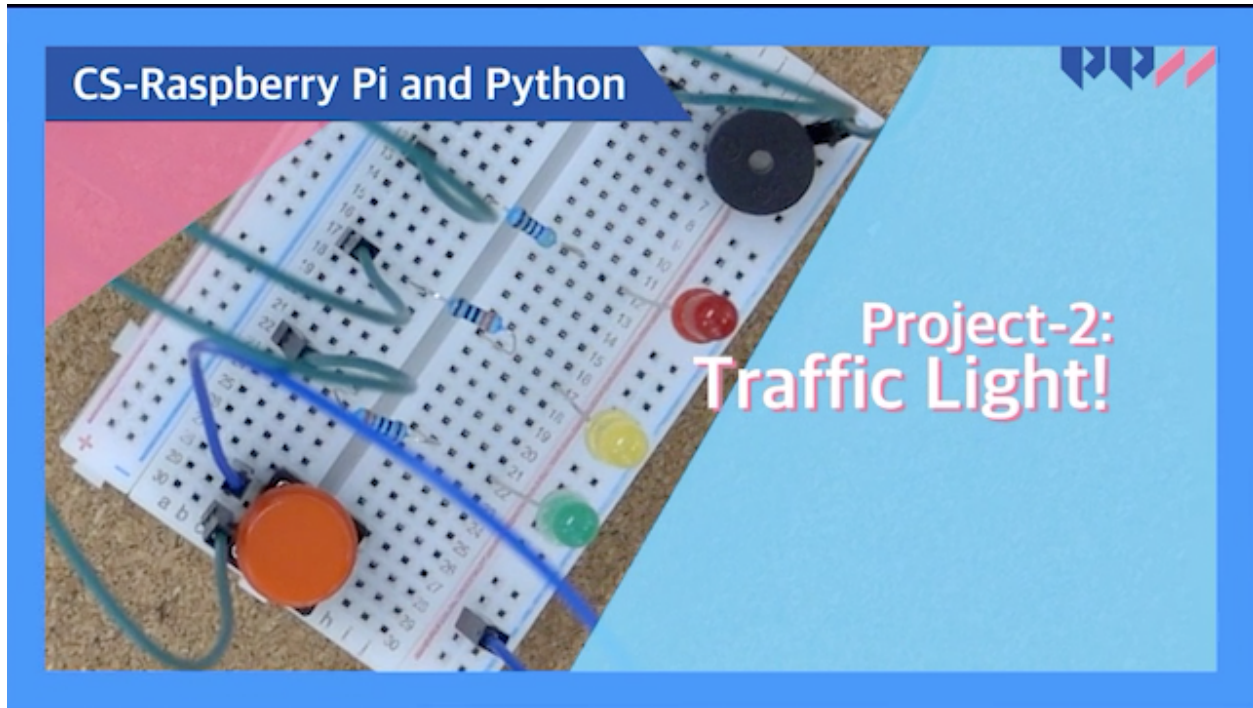
## Upgrade the Buzzer program

```
from gpiozero import Buzzer
from time import sleep
buzzer = Buzzer(17)
while True:
    buzzer.on()
    sleep(1)
    buzzer.off()
    sleep(1)
```

We added the forever loop to get the buzzer on for one second and then off for when second. If you want to shut the buzzer off, make sure to stop the program during an off moment so it stays off. If you kill the program while it is on, it will stay on.



## Project-2: Traffic Light!

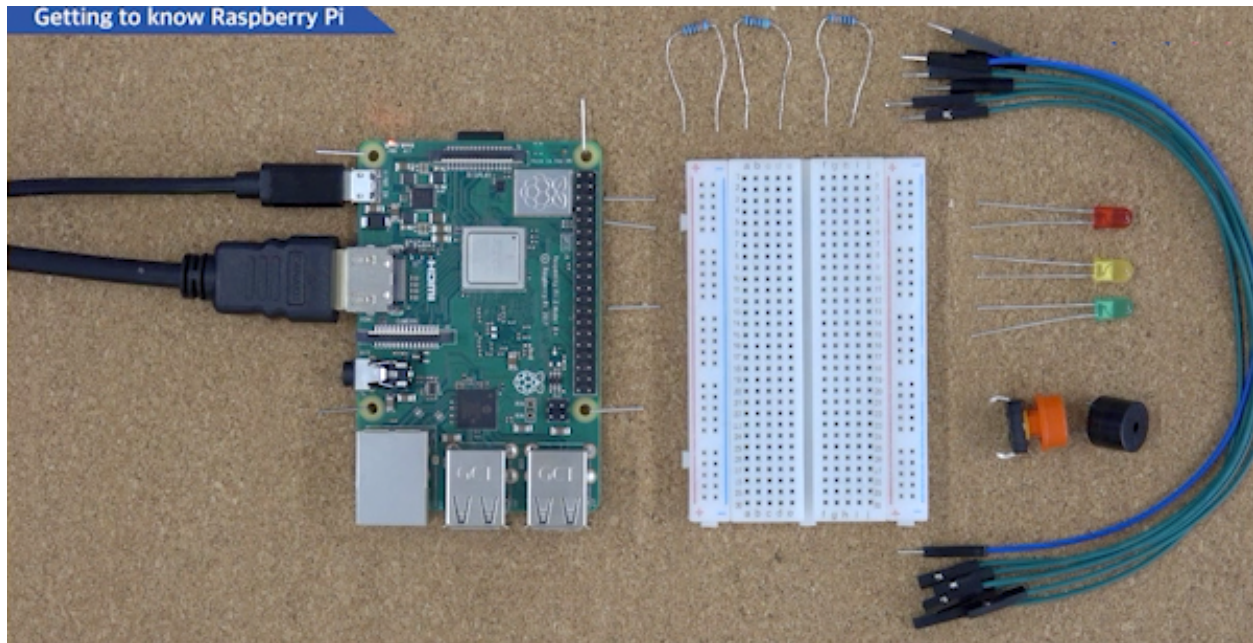


### Materials:

3 LEDs, a button, and a buzzer

### Steps

The hardware



Component	GPIO pin
Button	21
Red LED	25
Amber LED	8
Green LED	7
Buzzer	15

One of the standard projects used when teaching Python on a Raspberry Pi is the Traffic Jam project. This uses 3 LEDs, a button, and a buzzer.

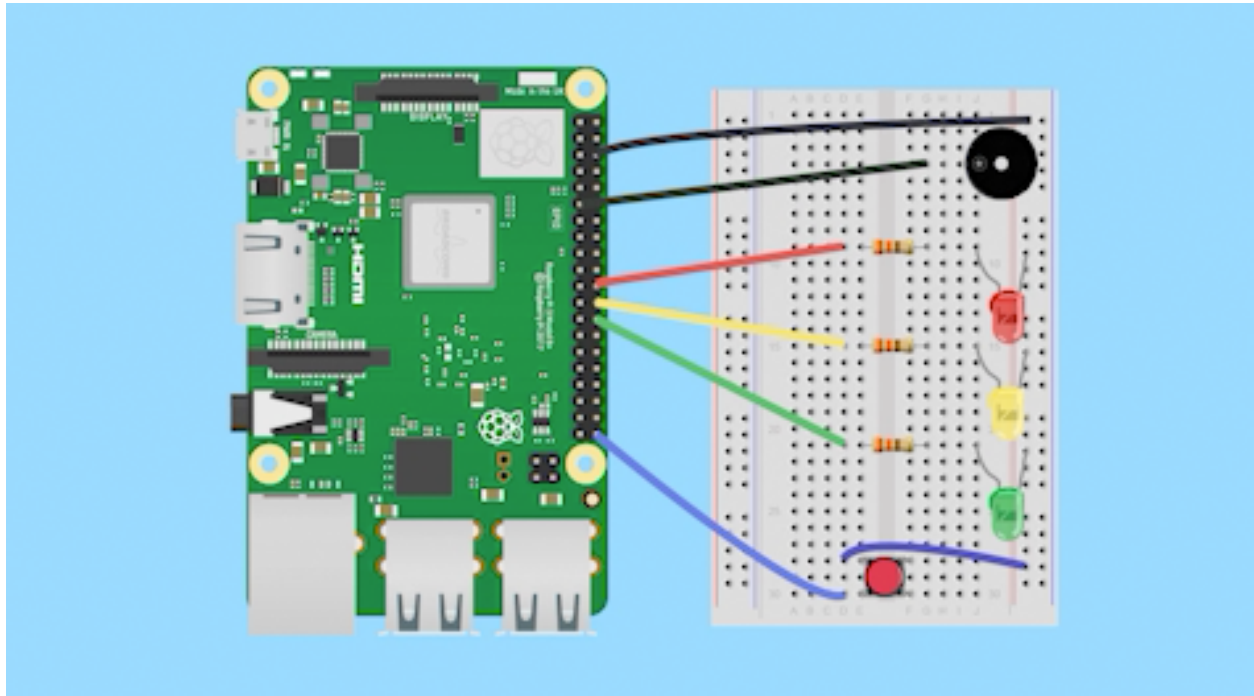
We are going to combine all three for this small project.



---

## Project-2: Traffic Light - the hardware

We need to wire our components to the breadboard. Check the diagram below,



1. First, place the LEDs in the breadboard with the short legs (negative) in the negative rail on the breadboard. Notice the color order of the LEDs in the diagram. We want the LEDs to resemble a traffic light. Place the long legs (positive) in the hole across the column and rail divider. Next, add a resistor from the positive leg across the center of the breadboard. For each LED, you will need a male/female jumper to connect them to the Raspberry Pi. For the green LED, connect the jumper to GPIO7. For yellow, connect it to GPIO 8 and for red, connect it to GPIO 25.
2. Next, add the buzzer so that one of the leads is in the negative rail and the other is across like the LED. Connect a male/female jumper wire from GPIO15 to the column below the positive buzzer lead.
3. Now set up the button across the center divide of the breadboard with a male/female jumper going from GPIO 2 and a male/male jumper wire going from the other lead to the negative rail. This connects the button to the Ground.

4. Lastly, take a male/female jumper wire and connect the ground to the negative rail on the breadboard that holds the LEDs and other components. This should mean you have everything wired. Check the diagram to make sure everything looks correct and make sure you have pressed the components and jumpers into the breadboard sufficiently.

---

## Project-2: Traffic Light - the code

### Step -1 Testing out the button

```
from gpiozero import Button
```

```
button = Button(21)
```

```
while True:
    if button.is_pressed:
        print("Hi")
    else:
        print("Bye")
```

Let's test each component one by one. First, the button. If the button is pressed, print "Hi". Else, print "Bye".

### Step -2 Testing Button with LED

```
from gpiozero import Button
```

```
from gpiozero import LED
```

```
button = Button(21)

led = LED(25)

while True:
    button.wait_for_press()

    led.on()

    button.wait_for_release()

    led.off()
```

Next, Let's check one of the LED. When the button is pressed, LED turns on. When the button is released, LED turns off.

### Step -3 Testing Button with LED and Buzzer

```
from gpiozero import Button
from gpiozero import LED
from gpiozero import Buzzer

button = Button(21)

led = LED(25)

buzzer = Buzzer(15)

while True:
    led.off()

    buzzer.off()

    button.wait_for_press()

    led.on()

    buzzer.on()

    button.wait_for_release()
```

Lastly, let's check the Buzzer, LED and the Button altogether. To begin, LED and Buzzer will be set to off. When the button is pressed, LED will light up. Then the Buzzer will turn on. When the button is released, we go back to the first line of the while loop and LED and Buzzer are turned off.

---

## The Traffic Light Program

```
from gpiozero import Buzzer
from gpiozero import Button
from gpiozero import LED
from time import sleep
```

```
button = Button(21)
buzzer = Buzzer(15)
redled = LED(25)
yellowled = LED(8)
greenled = LED(7)
```

```
while True:
    if button.is_pressed:
        greenled.on()
        sleep(2)
        greenled.off()
        yellowled.on()
        sleep(2)
```

```
yellowled.off()  
redled.on()  
buzzer.on()  
sleep(2)  
redled.off()  
buzzer.off()  
break
```

When you push the button, the traffic lights should change, the buzzer should sound, and then everything will shut off. They will restart when the button is pressed again.

---

## The Challenge

Can you change the duration of the lights being on?

Can you change the order they turn on?

Can you create a light and buzzer for people who are crossing the street?