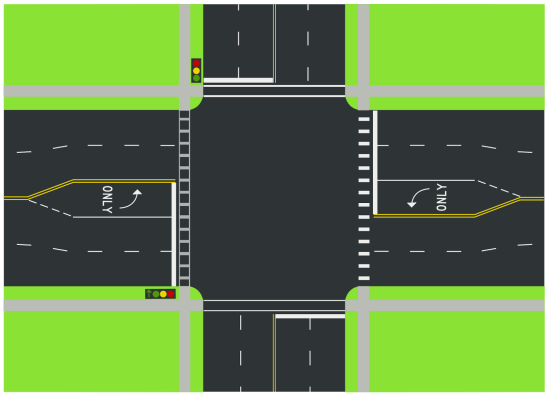
Traffic Lights

using

Python and Raspberry Pi

Overview

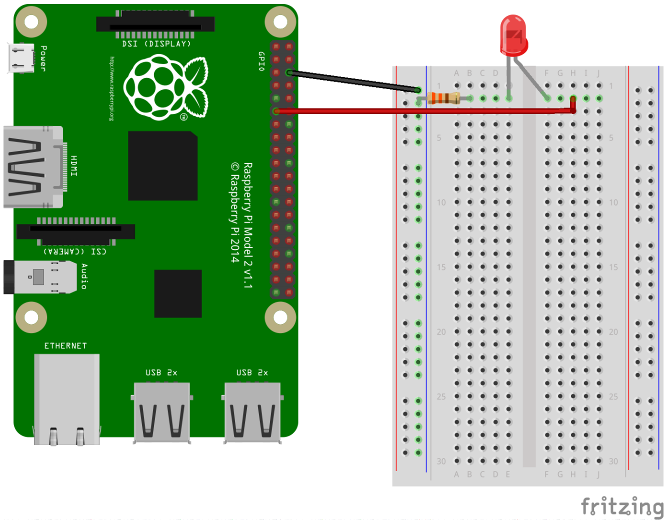
You are going to learn how to connect lights (LEDs) to your Raspberry Pi. You will then program a set of working traffic lights to control a road junction like the one below.



Step 1 - Open the Python IDE

1. On the desktop, click **Menu > Programming > Python 3 (IDLE)**
2. When the Python Shell appears, open a new file. Click **File > New File**

Step 2 - Light up a Red LED



1. You need to build a circuit as shown below, you will need:
   a. 1 x Red LED
   b. 1 x 330 Ohm Resistor
   c. 1 x Breadboard
   d. 2 x Male to Female cables

Notice how the LED has one leg longer than the other. The long leg goes to the right-hand side, the short leg goes the same side as the resistor.

2. Lets test our wiring. **Add** this code to the new file:

```
from gpiozero import LED
from time import sleep
```

3. Then **Add**:

```
red_led = LED(17)
```

This tells the Raspberry Pi that we are going to use pin 17 for output. The Red LED is connected to pin 17.

4. Now create a loop to turn the led on and off once a second:

```
while True:
    red_led.on()
    sleep(1)
    red_led.off()
    sleep(1)
```

Note: Make sure you tab in all of the lines under the while True:

5. Test out your circuit by pressing F5, saving your file when prompted. Does the Red LED light up?

If it doesn’t light up check:

a. you have the wires plugged into the right pins on the Raspberry Pi
b. the wires are connected to the right places on the breadboard
c. the LED is connect the right way
d. the code is correct. It should look like this:

```
from gpiozero import LED
from time import sleep

red_led = LED(17)

while True:
    red_led.on()
    sleep(1)
    red_led.off()
    sleep(1)
```

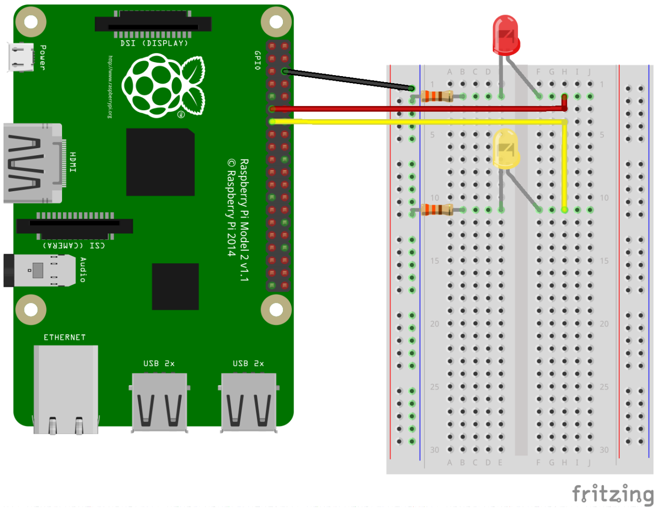
Job Done!!

Step 3: Adding an Amber Light!

*After the Red light, show a Red **and** Amber light together. You need to add an Amber (Yellow) LED to your circuit and update your code to leave the Red LED on for **3** seconds and then turn on the Red and Amber LED **together**.*

1. Add an Amber (Yellow) LED to the circuit board.

Double check that you put the LED the correct way around.



2. Let's add code to make the Amber LED work. After the red_led setup code, add the setup for the amber_led.

```
red_led = LED(17)
amber_led = LED(27)
```

3. Can you see which pin the Amber LED is connected to?
   That's right it's pin 27.

4. Change the loop so that it starts to follow the traffic light sequence.

```
while True:
    red_led.on()
    sleep(3)
    red_led.on()
    amber_led.on()
    sleep(1)
```

5. Test out your circuit by pressing F5, saving your code. Does the Red LED light up for 3 seconds and then Red and Amber?
6. Test out your circuit again by pressing F5. What happened? **Did the Red and Amber light stay on all the time?**

   > We haven't turned any lights out yet. When we first turn on the Red LED, we need to turn off the Amber LED.

7. **Update** the loop so that it looks like this:

```
while True:
    red_led.on()
    amber_led.off()
    sleep(3)
    red_led.on()
    amber_led.on()
    sleep(1)
```

   Can you spot the difference, it is now telling the amber_led to turn Off.

8. Test out your circuit by pressing F5.

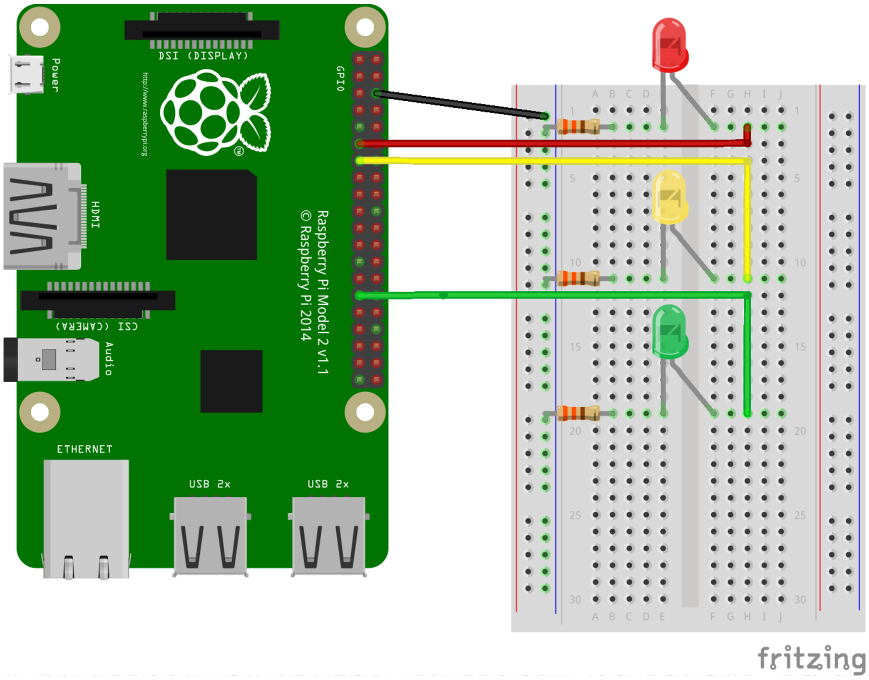
   Does the Red LED light up for 3 seconds and then Red and Amber?

   Try it again and again. Does it always work?

Job Done!!!

Step 4 - Adding the Green LED

1. Add a Green LED to the circuit board as shown.



Make sure you connect the correct pin on the Pi and have the LED the correct way around.

2. **Update** the setup code to create a Green LED.

```
red_led = LED(17)
amber_led = LED(27)
green_led = LED(5)
```

3. Now we need to update the loop to include the Green LED:

```
while True:
    red_led.on()
    amber_led.off()
    green_led.off()
    sleep(3)
    red_led.on()
    amber_led.on()
    green_led.off()
    sleep(1)
    red_led.off()
    amber_led.off()
    green_led.on()
    sleep(3)
```

4. Test out your circuit by pressing F5.

   Does the Red LED light up for 3 seconds and then Red and Amber for 1 second and finally Green for 3 seconds?

Job Done!!!

Step 5 - Finish The Sequence

The last step in the traffic light sequence is Amber by itself.

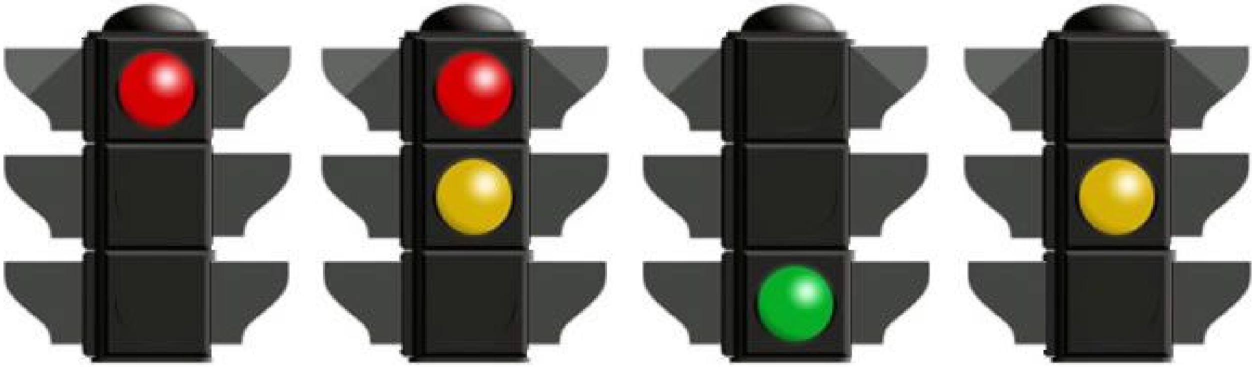
1. **Update** the loop to show amber after the green light.

```
while True:
    red_led.on()
    amber_led.off()
    green_led.off()
    sleep(3)
    red_led.on()
    amber_led.on()
    green_led.off()
    sleep(1)
    red_led.off()
    amber_led.off()
    green_led.on()
    sleep(3)
    red_led.off()
    amber_led.on()
    green_led.off()
    sleep(1)
```

2. Test out your circuit by pressing F5

.

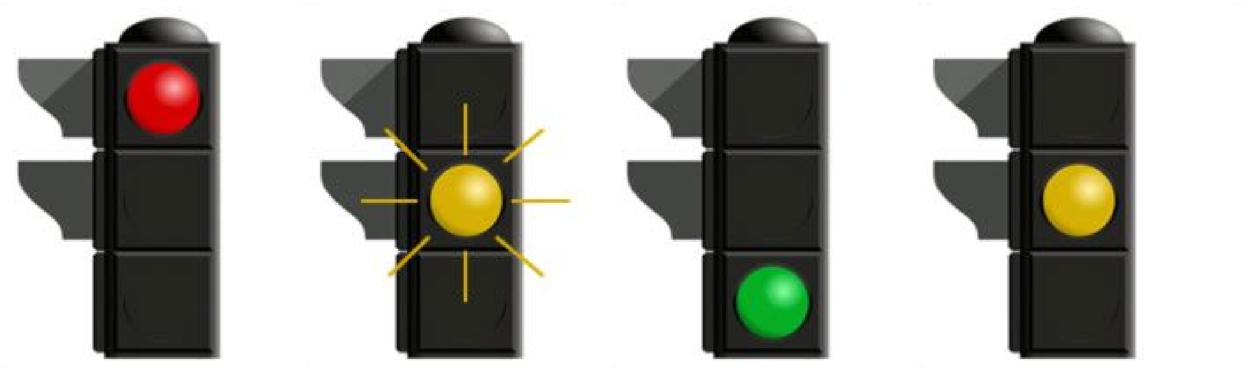
Did you see this?



Job Done!!!

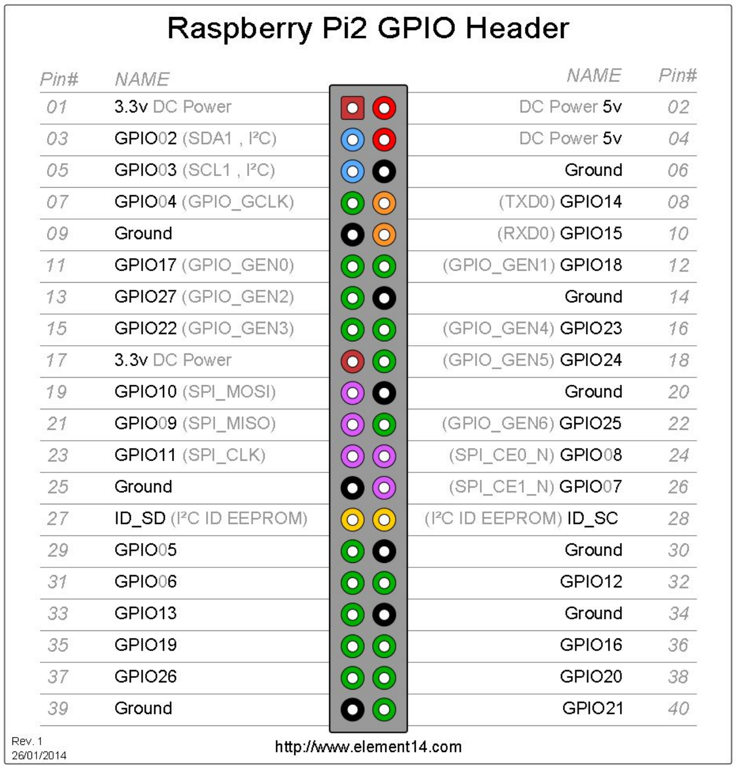
Challenge 1

Some traffic lights have a different sequence. Pedestrian Crossings called Pelican Crossings have a flashing Amber light instead of Red Amber. Can you modify the project to be a Pelican Crossing?



Challenge 2

Add another Red LED and another Green LED so the Pedestrian knows when to cross. You can use GPIO 22 for Red and GPIO 23 for Green.



Raspberry Pi2 GPIO Header

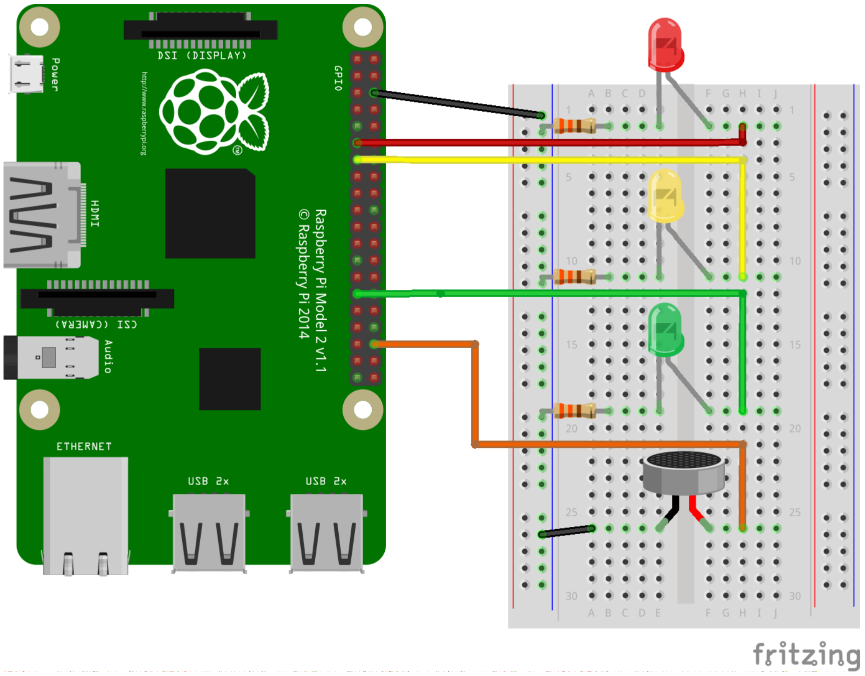
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 1
26/01/2014

http://www.element14.com

Challenge 3

Connect a buzzer that beeps when it’s safe to cross. We’ll connect it to GPIO pin 16.



The code that you will need to add is similar to the code for an LED. First you need to import the Buzzer class. **Add** the Buzzer to your first line:

```
from gpiozero import LED, Buzzer
```

After the green_led setup code, add the setup for the buzzer:

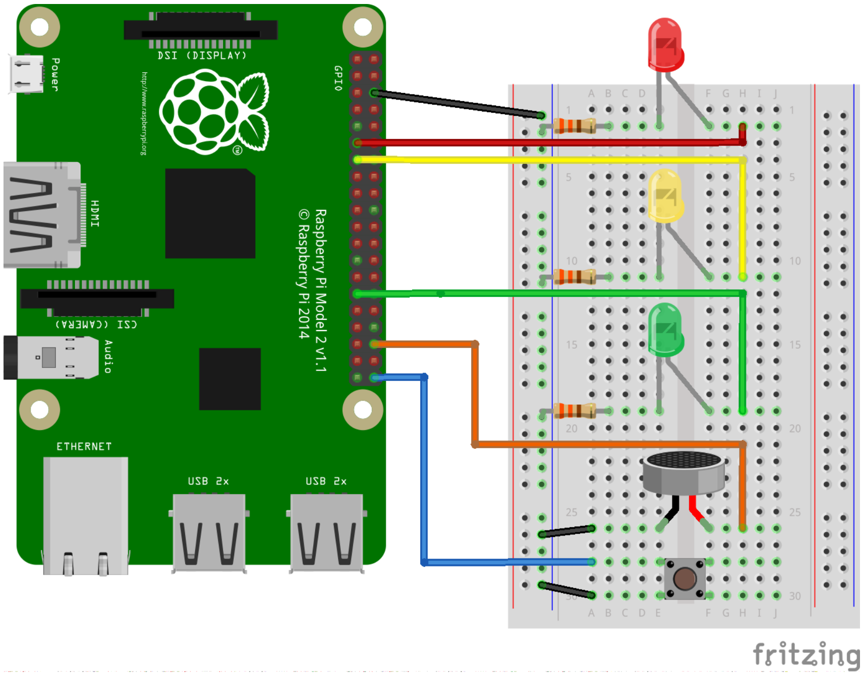
```
buzzer = Buzzer(16)
```

To turn the buzzer on and off, you can use the following code:

```
buzzer.on()
buzzer.off()
```

Challenge 4

Connect a button for a pedestrian to press when they want to cross. We'll connect it to GPIO pin 21.



Use the following code to understand how the button works before completing this challenge:

```
from gpiozero import LED, Button

red_led = LED(17)
button = Button(21)

while True:
    if button.is_pressed:
        red_led.on()
    else:
        red_led.off()
```