

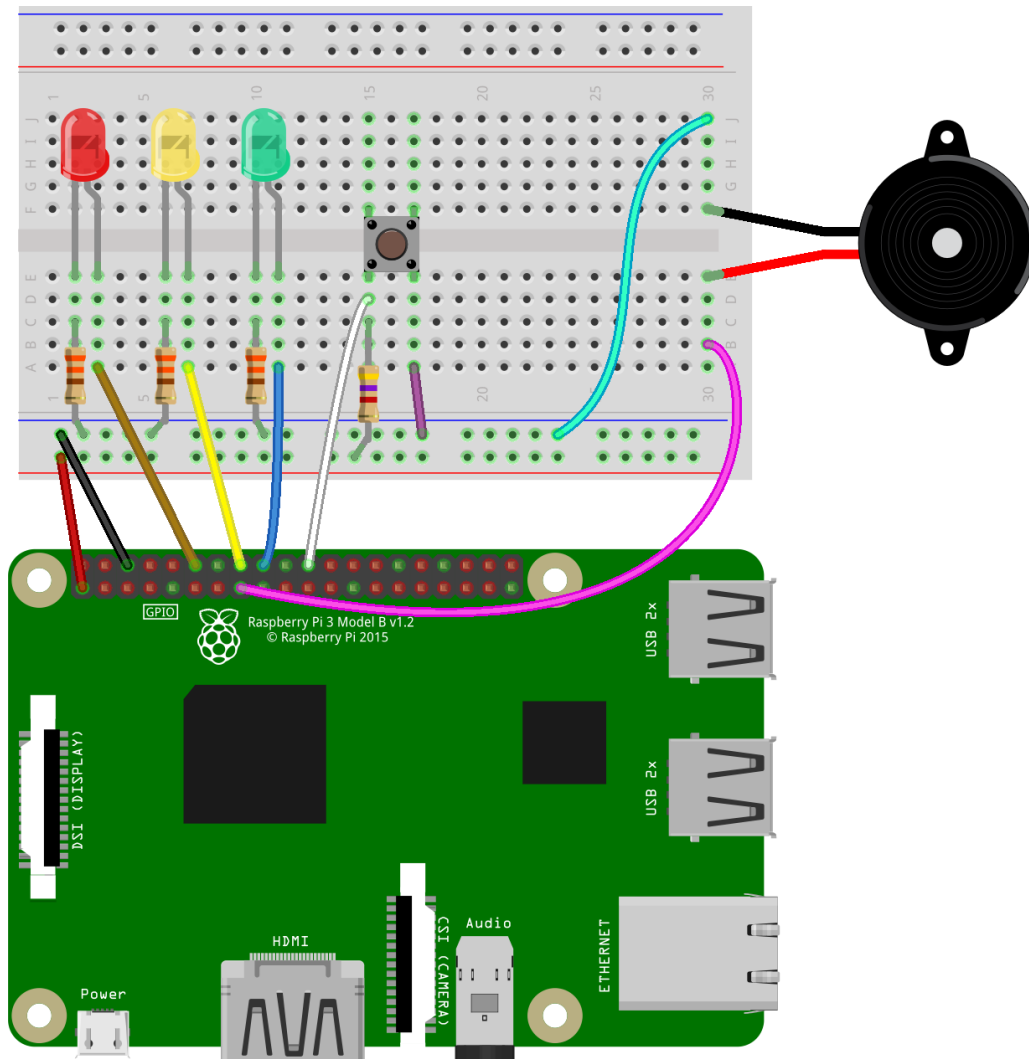
CamJam EduKit Worksheet Seven

Project Traffic Light Simulator

Description In this project, you will program a traffic light simulator using the circuit from CamJam EduKit Worksheet Six.

Equipment Required

The circuit built in CamJam EduKit Worksheet Six.



fritzing

Exercise

In this worksheet, you are going to program your Raspberry Pi with EduKit to act like a standard UK Pelican Crossing traffic lights. All UK lights work in the same way so that drivers know what to expect when they approach them.

Using the techniques learnt in the previous worksheets, your aim is to make the kit act like the traffic lights at a Pelican Crossing, reacting to the button press to allow pedestrians to cross. The LEDs will act as signals to the vehicles; the buzzer will act as a signal to the pedestrians.

The standard sequence is as follows:

| Action | Signal to Vehicles | Signal to Pedestrians | Timings |
|--|---|--|---------------------|
| | Steady Green | Red Standing Figure | |
| Pedestrian presses the button | Steady Amber (Cars should stop if they can) | Red Standing Figure | 3 seconds |
| | Steady Red (Cars must stop) | Red Standing Figure | 1 second |
| Pedestrians can start walking | Steady Red | Green Waking Figure and Beeping | 4 to 7 seconds |
| Pedestrians should not start to cross | Steady Red | Flashing Green Figure and no sound | 2 seconds |
| | Flashing Amber (Cars can start to go again if the crossing is clear) | Flashing Green Figure | 6 seconds |
| | Flashing Amber | Red Standing Figure | 1 second |
| | Steady Green (Cars can proceed) | Red Standing Figure | At least 20 seconds |

On the last step, the button can be pressed within the 20 seconds, but the lights should not change immediately. After the 20 seconds, the process can start again.

Using knowledge from the previous CamJam Worksheets, the Code Hints, and the outline code, write your traffic light code!

Code Hints

Here are some reminders of some of the code you will need:

| | |
|--|--|
| <code>import library</code> | Remember to import required libraries at the start of the code. You will definitely need time and GPIO Zero. |
| <code>colour = LED(xx)</code> | Sets the pin xx to be used as and LED pin |
| <code>time.sleep(x)</code> | Wait for x seconds |
| <code>colour.on()</code> | Turns on the <i>colour</i> LED |
| <code>while condition</code> xxxx yyyy | While condition is true (e.g. x<1), run the commands xxxx and yyyy. |
| <code>if condition:</code> xxxx | If <i>condition</i> is true, run code xxxx, otherwise run code yyyy |

```
else:  
    yyyy
```

Code Outline

There are comments where you need to fill in some code. Create the file in IDLE3 and start with the following code:

```
# CamJam EduKit 1 - Basics  
# Worksheet 7 - Traffic Lights Template  
  
# Import Libraries  
import os  
import time  
from gpiozero import LED, Button, Buzzer  
  
# Set up variables for the LED, Buzzer and switch pins  
  
# Define a function for the initial state (Green LED on, rest off)  
# (If you have the second 'pedestrian' LEDs, turn the red on & green  
# off)  
def startgreen():  
    # Remember all code in the function is indented  
  
# Turn the green off and the amber on for 3 seconds  
# ('Pedestrian' red LED stays lit)  
def steadyamber():  
    # Remember all code in the function is indented  
  
# Turn the amber off, and then the red on for 1 second  
def steadyred():  
    # Remember all code in the function is indented  
  
# Sound the buzzer for 4 seconds  
# (If you have the 'pedestrian' LEDs, turn the red off and green on)  
def startwalking():  
    # Make the buzzer buzz on and off, half a second of  
    # sound followed by half a second of silence  
  
# Turn the buzzer off and wait for 2 seconds  
# (If you have a second green 'pedestrian' LED, make it flash on and  
# off for the two seconds)  
def dontwalk():  
    # Remember all code in the function is indented  
  
# Flash the amber on and off for 6 seconds  
# (And the green 'pedestrian' LED too)  
def flashingambergreen():
```

```
# Remember all code in the function is indented

# Flash the amber for one more second
# (Turn the green 'pedestrian' LED off and the red on)
def flashingamber():
    # Remember all code in the function is indented

# Go through the traffic light sequence by calling each function
# one after the other.
def trafficlightqsequence():
    # Remember all code in the function is indented

os.system('clear') # Clears the terminal
print("Traffic Lights")
# Initialise the traffic lights
startgreen()

# Here is the loop that waits at lease 20 seconds before
# stopping the cars if the button has been pressed
while True: # Loop around forever
    buttonnotpressed = True # Button has not been pressed
    start = time.time() # Records the current time
    while buttonnotpressed: # While the button has not been pressed
        time.sleep(0.1) # Wait for 0.1s
        if button.ispressed: # If the button is pressed
            now = time.time()
            buttonnotpressed = False # Button has been pressed
            if (now - start) <= 20: # If under 20 seconds
                time.sleep(20 - (now - start)) # Wait until 20s is up
                trafficlightqsequence() # Run the traffic light sequence
```

Running the Code

Run your code from the menu or by pressing F5. If errors are reported, check your code again.

Press the button while the green LED is lit and see what happens.

Challenge

Use a second CamJam EduKit, or buy additional green and red LEDs and 330Ω resistors and use them as the Red and Green pedestrian figures.