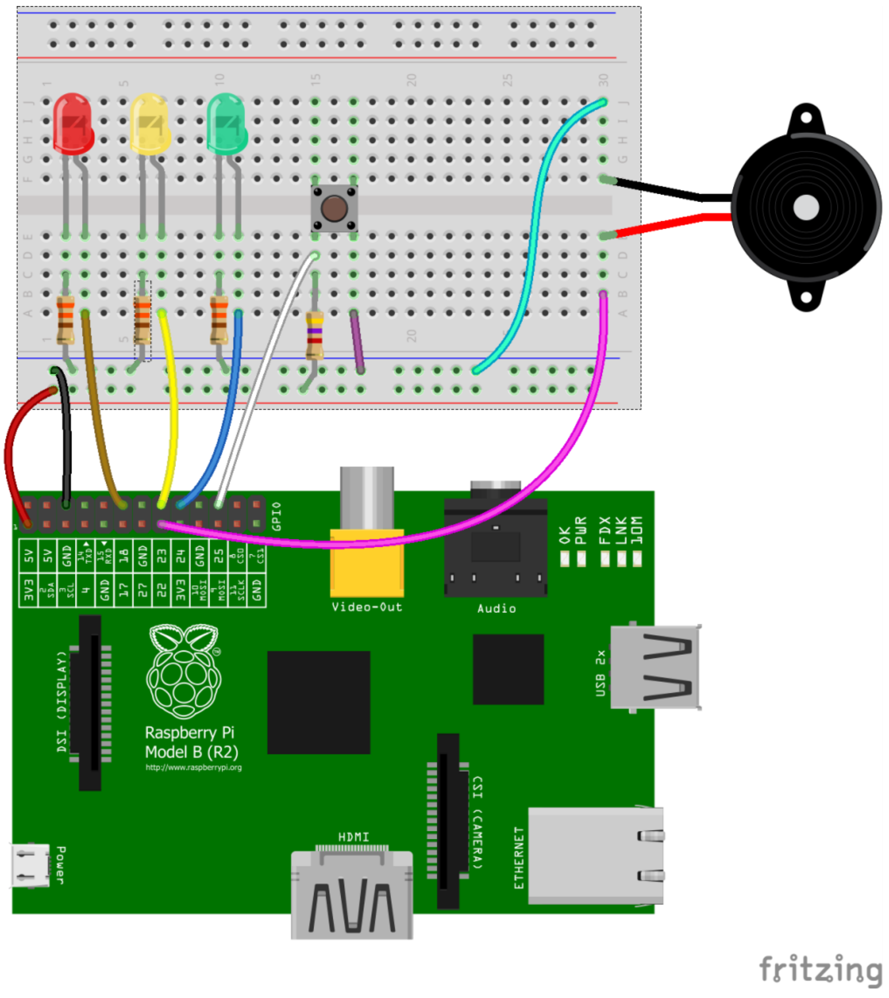
CamJam EduKit Worksheet Seven

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

Equipment Required

The circuit built in CamJam EduKit Worksheet Six.



Exercise

In this worksheet, you are going to program your Raspberry Pi with EduKit to act like 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.

Exercise

Using the techniques learnt in the previous worksheets, your aim is to make the kit to 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 signals 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:

`import library`	Remember to import required libraries at the start of the code. You will definitely need time and RPi.GPIO.
`GPIO.setup(nn,GPIO.IN)`	Sets the pin *nn* to be used for input
`time.sleep(x)`	Wait for *x* seconds
`GPIO.output(nn,GPIO.HIGH)`	Turns on GPIO pin *nn*

Code Hints

`while condition`
 `    xxxx`
 `    yyyy`
	While condition is true (e.g. x<1), run the commands *xxxx* and *yyyy*.

`if condition:`
 `    xxxx`
 `else:`
 `    yyyy`
	If *condition* is true, run code *xxxx*, otherwise run code *yyyy*

Code Outline

There are comments where you need to fill in some code. Create the file and open the editor with the `nano` command:

```
# Import Libraries
import os
import time
import RPi.GPIO as GPIO

# Set the GPIO pin naming mode
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Set up variables for the LED, Buzzer and switch pins

# Set up each of the input (swich) and output (LEDs, Buzzer) 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 abber 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)
```

Code Outline

```
def StartWalking():
    #Try and 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 throught the traffic light sequence by calling each function
# one after the other.
def TrafficLightSequence():
    # Remember all code in the function is indented

os.system('clear') #Clears the screen
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 pressedS
while True: #Loop around forever
    ButtonNotPressed = True # Button has not been pressed
    start = time.clock() # Records the current time
    while ButtonNotPressed: # While the button as not been pressed
        time.sleep(0.1) # Wait for 0.1s
        if GPIO.input(ButtonPin) == False: # If the button is pressed
            ButtonNotPressed = False # Button has been pressed
            if time.clock()-start<=20 # If under 20 seconds
                time.sleep (20-start)  # Wait until 20s is up
            TrafficLightSequence() # Runthe traffic light sequence
```

Code Outline

```
GPIO.cleanup()
```

Running the Code

To run your code using the **`sudo python`** command. It may not run perfectly first time, but you can always re-edit it with **`nano`**.

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

Challenge

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