

Experiment no - 8

Aim :- write an application using Raspberry - Pi 1. beagle board to control operation of a hardware simulated traffic signal

Theory :-

Attaching traffic lights :-

The low voltage labs traffic lights connect to Pi using four pins. one of these needs to be ground, the other three being actual GPIO pins used to control each of individual LED's.

Programming the traffic lights :-

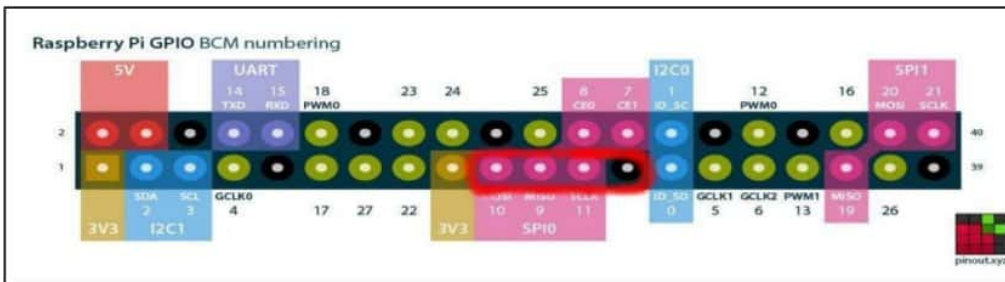
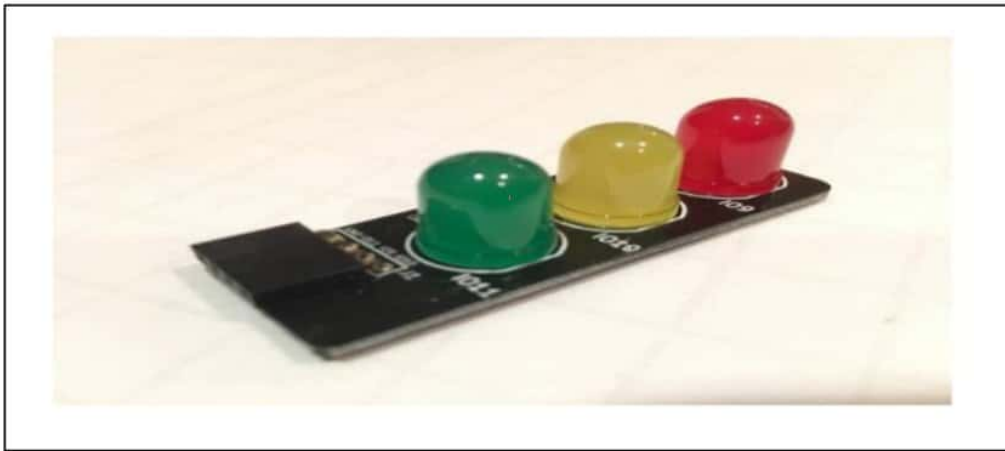
You need to install a couple of extra software packages needed to allow you to download sample code & to give python access to GPIO pins on Pi.

```
sudo apt-get install python-dev python-rpi-gpio git
```

How it works :-

The code for this is very simple. it starts by importing RPi.GPIO library, plus time which gives us a time wait function, signal that allows us to trap signal sent when user tries to quit program.

```
import RPi.GPIO as GPIO
import time
import signal
import sys
```



setup :-

```
GPIO.setmode(GPIO.BCM)
GPIO.setmode up (9, GPIO.OUT)
GPIO.setup (10, GPIO.OUT)
GPIO.setup (11, GPIO.OUT)
```

The main part of program will run in infinite loop until user exits it by stopping python with ctrl+c. It's a good idea to add a handler funⁿ that will run whenever this happens so that we can turn off all lights prior to existing.

Turn off all lights when user ends demo

```
def allLightsoff (signal, frame):
```

```
    GPIO.output (9, false)
```

```
    GPIO.output (10, false)
```

```
    GPIO.output (11, false)
```

```
    GPIO.cleanup ()
```

```
    sys.exit (0)
```

```
signal.signal (signal.SIGINT, allLightsoff)
```

The main body of code then consists of infinite while loop that turns on red light, waits, turn on amber light, waits then cycles through rest of traffic light pattern by turning appropriate LEDs on & off.

Conclusion :-

Thus, we have implemented application for traffic signals using Raspberry Pi