Page No.
Date

• Name:- Pritam Gotiram Mogal.
Div: TE-B
Roll No: 17
Sub:- ESIOT

Experiment No:-8

Aim :- Write an application using Rasberry-Pi / beagle board to control operation of a hardware simulated traffic signal.

Theory:-

Attaching traffic lights:-

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

Programming the traffic lights:-

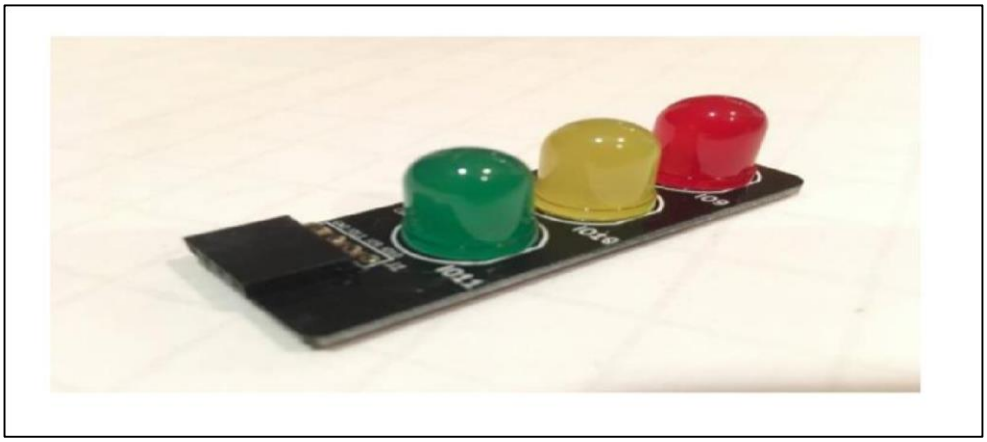
You need to install a couple of extraa software packages needed to allow you to download. Sample code & to give python access to GPIO pins on Pi

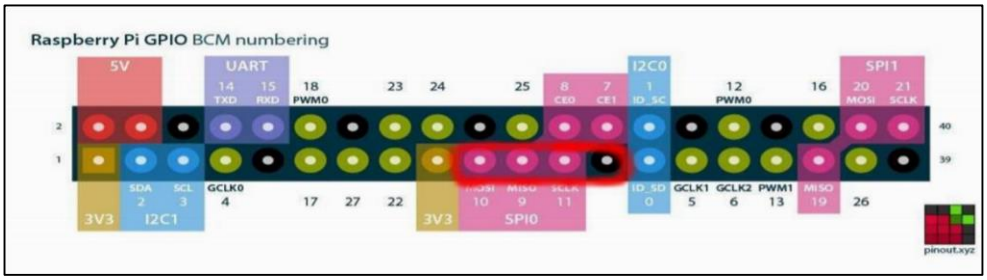
sudo apt-get install python-dev python-rpi gpio git

How it works:

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

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




Raspberry Pi GPIO BCM numbering
5V
UART
I2C0
SPI1
3V3
I2C1
SPI0
pinout.xyz

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 ctrlC. It's a good idea to add handler fun^n 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 ambee light, waits then cycles through rest of traffic light pattern by turning appropriate LED's on & off.

Conclusion :-

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