Assignment Number - 9

Title: Controlling the operation of a hardware simulated traffic signal using Raspberry-Pi / BeagleBoard circuit.

Problem Definition: Write an application using Raspberry-Pi / BeagleBoard to control the operation of a hardware simulated traffic signal.

Objectives:

- ➔ Understanding the controlling of devices through Raspberry-Pi / BeagleBoard.
- ➔ To understand the actuation.

Outcomes:

Students will be able to:

- ➔ Simulate traffic signals through LEDs.
- ➔ Control this simulated traffic signal through Raspberry-Pi / BeagleBoard.
- ➔ Perform actuation.

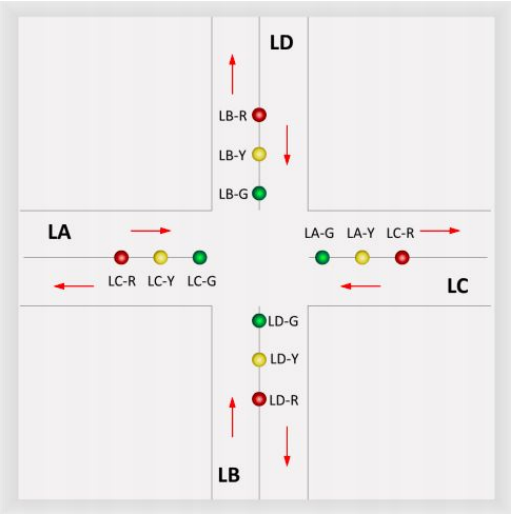
Software and Hardware Requirements:

Raspberry-Pi / BeagleBoard, Breadboard, Green LEDs, Yellow LEDs, Red LEDs, Raspbian (OS), Cables, PC, Mouse, Keyboard, Python.

Theory:

A simple traffic light system for a 4 way intersection is implemented using Raspberry-Pi where the traffic is controlled in a predefined timing system. There are 4 lanes LA, LB, LC and LD going towards the signal. At the cross road there are 4 sets of Traffic lights opposite to each lane. These sets are:

1. LA (LA-G, LA-Y, LA-R)
2. LB (LB-G, LB-Y, LB-R)
3. LC (LC-G, LC-Y, LC-R)
4. LD (LD-G, LD-Y, LD-R)



Traffic from any lane moves when its corresponding Green Light is ON. The "ON time" of any Red Light is dependent on the "ON time" of Yellow Light and Green Light of the other 3 signal lights. The "ON time" of the Yellow Light is the same for all the lanes. Users can specify and change the "ON time" of the Green Light and Red Light of each signal separately. The Traffic light pattern keeps on repeating till the next change made by the user.

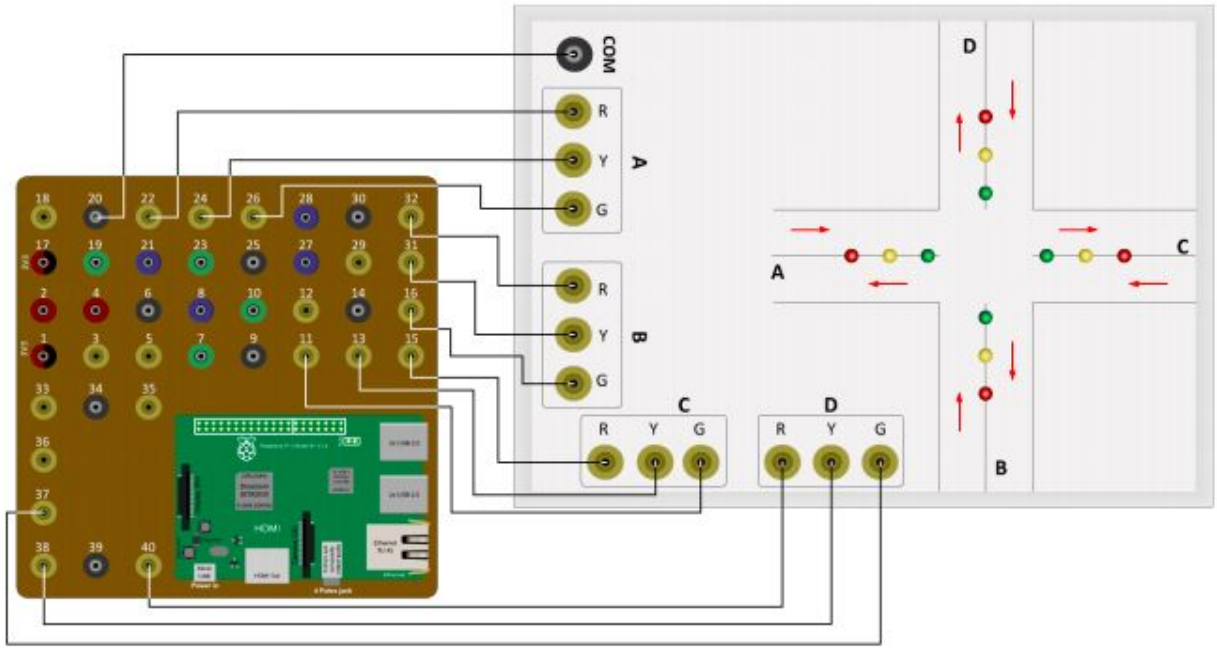
Algorithm:

1. Import RPi.GPIO library.
2. Import Time library.
3. Declare all the LED pins which are connected to the GPIO pins of Raspberry-Pi board.
4. Set mode i.e. GPIO.BOARD.
5. Take delay time from the user for each lane.
6. Set all the LED pins as output.
7. Define 4 functions to control the traffic light as:
    a. trafficState1
    b. trafficState2
    c. trafficState3
    d. trafficState4
8. Firstly for LA, the signal becomes Green.
9. Hence, for all the other lanes (LB, LC, LD), the corresponding Red Signal is on.
10. After a time delay, as a warning indicator, the Yellow Light in LA signal is turned on indicating that the Red Light is about to light up.
11. After a time delay for Lane 3, the signal becomes Green. So, at the same time, the signal for Lane 1 becomes Red.
12. Second time for LB, the signal becomes Green.
13. Hence, for all the other lanes (LA, LC, LD), the corresponding Red Signal is on.
14. After a time delay, as a warning indicator, the Yellow Light in LB signal is turned on indicating that the Red Light is about to light up.
15. Third time for LC, the signal becomes Green.
16. Hence, for all the other lanes (LA, LB, LD), the corresponding Red Signal is on.
17. After a time delay, as a warning indicator, the Yellow Light in LC signal is turned on indicating that the Red Light is about to light up.
18. Fourth time for LD, the signal becomes Green.
19. Hence, for all the other lanes (LA, LB, LC), the corresponding Red Signal is on.
20. After a time delay, as a warning indicator, the Yellow Light in LD signal is turned on indicating that the Red Light is about to light up.
21. This process shall be repeated all over again.

Circuit Diagram:

1. Connect the R, Y, G pins of Lane A to 22, 24, 26 pins of Raspberry-Pi module respectively.
2. Connect the R, Y, G pins of Lane B to 32, 31, 16 pins of Raspberry-Pi module respectively.
3. Connect the R, Y, G pins of Lane C to 11, 13, 15 pins of Raspberry-Pi module respectively.
4. Connect the R, Y, G pins of Lane D to 40, 38, 37 pins of Raspberry-Pi module respectively.

5. Connect the COM pin of the Traffic Signal module to the GND pin of Raspberry-Pi module.



Source Code:

```
import time
Import RPi.GPIO as GPIO
LC-R=22
LC-Y=24
LC-G=26
LD-R=32
LD-Y=31
LD-G=16
LA-R=15
LA-Y=13
LA-G=11
LB-R=40
LB-Y=38
LB-G=37
LA=int(input("LA:"))
print(LA)
LB=int(input("LB:"))
print(LB)
LC=int(input("LC:"))
print(LC)
LD=int(input("LD:"))
print(LD)
RUNNING = True
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False);
```

```
GPIO.setup(LA-R, GPIO.OUT)
GPIO.setup(LA-Y, GPIO.OUT)
GPIO.setup(LA-G, GPIO.OUT)
GPIO.setup(LB-R, GPIO.OUT)
GPIO.setup(LB-Y, GPIO.OUT)
GPIO.setup(LB-G, GPIO.OUT)
GPIO.setup(LC-R, GPIO.OUT)
GPIO.setup(LC-Y, GPIO.OUT)
GPIO.setup(LC-G, GPIO.OUT)
GPIO.setup(LD-R, GPIO.OUT)
GPIO.setup(LD-Y, GPIO.OUT)
GPIO.setup(LD-G, GPIO.OUT)
def trafficState1(red1, yellow1, green1):
        GPIO.output(LA-R, red1)
        GPIO.output(LA-Y, yellow1)
        GPIO.output(LA-G, green1)
def trafficState2(red2, yellow2, green2):
        GPIO.output(LB-R, red2)
        GPIO.output(LB-Y, yellow2)
        GPIO.output(LB-G, green2)
def trafficState3(red3, yellow3, green3):
        GPIO.output(LC-R, red3)
        GPIO.output(LC-Y, yellow3)
        GPIO.output(LC-G, green3)
def trafficState4(red4, yellow4, green4):
        GPIO.output(LD-R, red4)
        GPIO.output(LD-Y, yellow4)
        GPIO.output(LD-G, green4)
print "Traffic Light Simulation. Press CTRL + C to quit"
try:
        while RUNNING:
                print("Green Light ON for LA ")
                trafficState1(0,0,1)
                trafficState2(1,0,0)
                trafficState3(1,0,0)
                trafficState4(1,0,0)
                time.sleep(LA)
                trafficState1(0,1,0)
                trafficState2(1,0,0)
                trafficState3(1,0,0)
                trafficState4(1,0,0)
                time.sleep(5)
                print("Green Light ON for LB")
                trafficState1(1,0,0)
```

```
                trafficState2(0,0,1)
                trafficState3(1,0,0)
                trafficState4(1,0,0)
                time.sleep(LB)
                trafficState1(1,0,0)
                trafficState2(0,1,0)
                trafficState3(1,0,0)
                trafficState4(1,0,0)
                time.sleep(5)
                print("Green Light ON for LC")
                trafficState1(1,0,0)
                trafficState2(1,0,0)
                trafficState3(0,0,1)
                trafficState4(1,0,0)
                time.sleep(LC)
                trafficState1(1,0,0)
                trafficState2(1,0,0)
                trafficState3(0,1,0)
                trafficState4(1,0,0)
                time.sleep(5)
                print("Green Light ON for LD")
                trafficState1(1,0,0)
                trafficState2(1,0,0)
                trafficState3(1,0,0)
                trafficState4(0,0,1)
                time.sleep(LD)
                trafficState1(1,0,0)
                trafficState2(1,0,0)
                trafficState3(1,0,0)
                trafficState4(0,1,0)
                time.sleep(5)
    except KeyboardInterrupt:
            RUNNING = False print "\Quitting"
    finally:
            GPIO.cleanup()
```

Conclusion:

We have successfully implemented the operation of traffic signals using Raspberry-Pi.