Assignment 4

Application using Beagle board to control the operation of a hardware simulated traffic signal

Shreya Pawaskar
Cno - C22018881961
Roll no - 4947
Division - C

Problem Statement: Write an application using Beagle board to control the operation of a hardware simulated traffic signal.

Theory :

1. Using Adafruit_BBIO Library with BBB

To use, import the library ; different options to import: GPIO, PWM, ADC

1. first one is for GPIO:

```
import Adafruit_BBIO.GPIO as GPIO
setup()
output()
input()
cleanup()
```

2. Second is for PWM

```
import Adafruit_BBIO.PWM as PWM
```

Example for GPIO :

```
import Adafruit_BBIO.GPIO as GPIO
GPIO.setup("P8_10", GPIO.OUT)
GPIO.output("P8_10", GPIO.HIGH)
GPIO.output("P8_10", GPIO.LOW)
GPIO.cleanup()
```

2. Time Library in python

This module provides various time-related functions

```
time.sleep(secs)
```

```
e.g.: time.sleep(10)
```

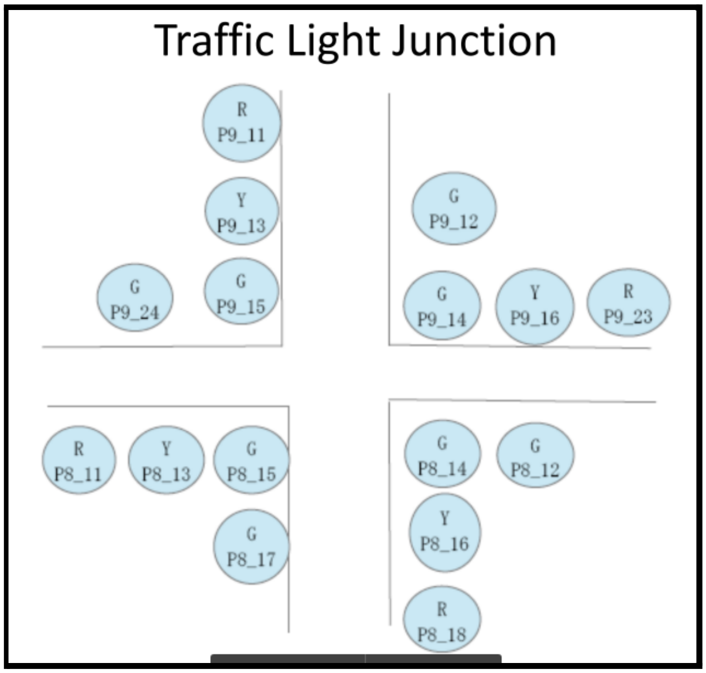
Suspend execution of the current thread for the given number of seconds. The argument may be a floating point number to indicate a more precise sleep time. Import time module in our programs to generate delays.

3. Traffic Light Controller

Mapping P8 and P9 expansion connector pins to GPIO pins

- LED used 16 in no, therefore 16 GPIO pins required!
- GPIO pins are mapped to P8 and P9 Expansion Connectors on BBB
- P8_11 => GPIO1_13, P8_12 => GPIO1_12,
P8_13 => GPIO0_23, P8_14 => GPIO0_26,
P8_15 => GPIO1_15, P8_16 => GPIO1_14,
P8_17 => GPIO0_27, P8_18 => GPIO2_1,

- P9_11 => GPIO0_30, P9_12 => GPIO1_28,
P9_13 =>GPIO0_31, P9_14 => GPIO1_18,
P9_15 => GPIO1_16, P9_16 => GPIO1_19,
P9_23 => GPIO1_17, P9_24 => GPIO0_15



1) Explain the following use of libraries with examples.

1. **Adafruit_BBIO Library with BBB**

- Adafruit BBIO is an API to enable GPIO, PWM, ADC, UART, SPI and eQEP (Quadrature Encoder) hardware access from Python applications running on the BeagleBone.
- Adafruit_BBIO supports Linux kernels 3.8 through 4.19.
- It is recommended to use Python 3.
- To use, import the library ; different options to import: GPIO, PWM, ADC

1. first one is for GPIO:

```
import Adafruit_BBIO.GPIO as GPIO
setup()
output()
input()
cleanup()
```

2. Second is for PWM

```
import Adafruit_BBIO.PWM as PWM
```

Example for GPIO :

```
import Adafruit_BBIO.GPIO as GPIO
GPIO.setup("P8_10", GPIO.OUT)
GPIO.output("P8_10", GPIO.HIGH)
GPIO.output("P8_10", GPIO.LOW)
GPIO.cleanup()
```

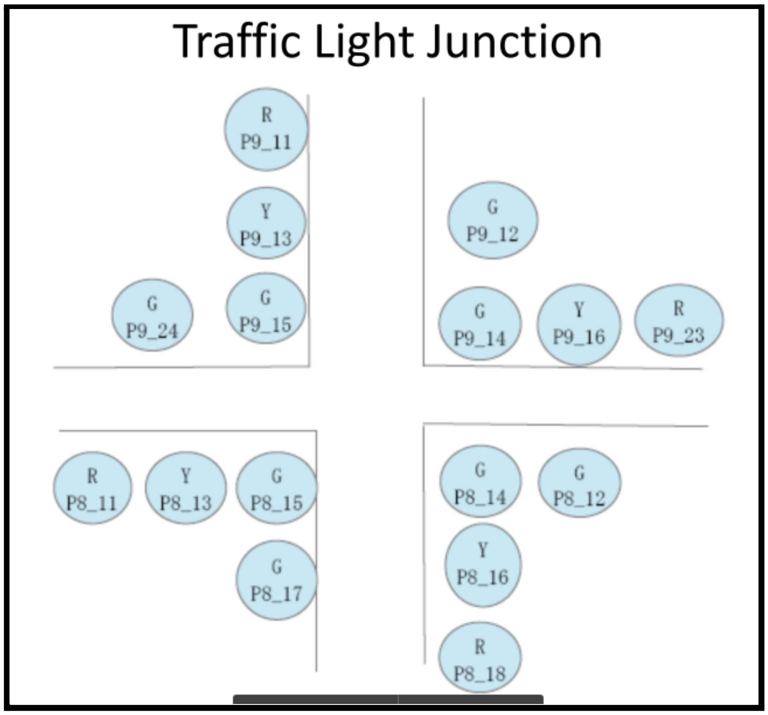
2. Time Library in python

This module provides various time-related functions.

time.sleep(secs) e.g.: time.sleep(10)

Suspend execution of the current thread for the given number of seconds. The argument may be a floating point number to indicate a more precise sleep time. Import time module in our programs to generate delays.

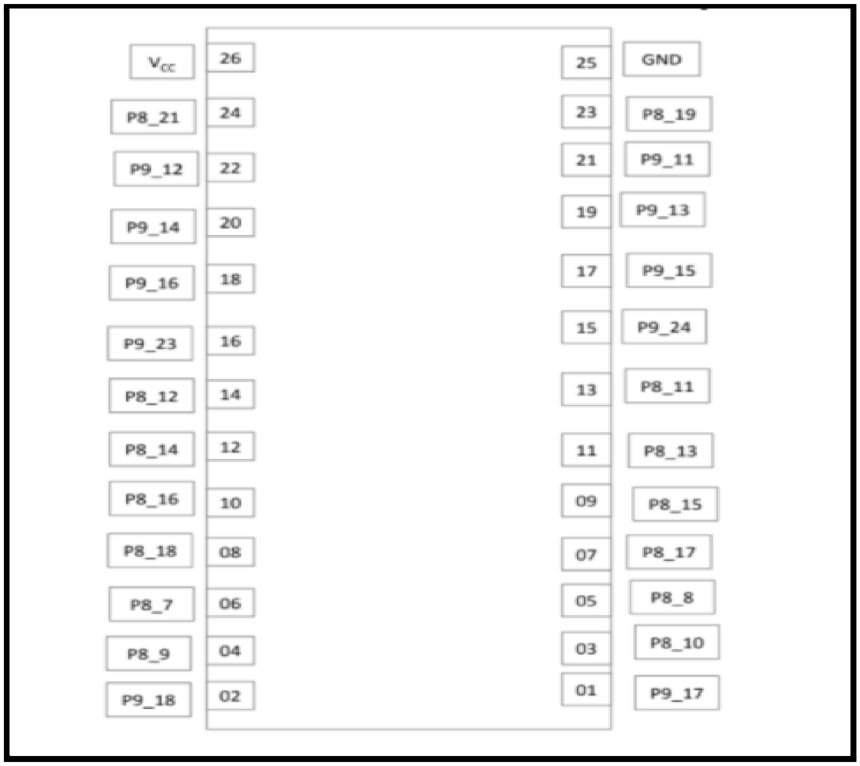
2) Draw the layout of the Traffic light controller board



3) List the pin nos used for implementation.

The pin no’s used for implementation are P9_11, P9_13, P9_15, P9_24, P9_16, P9_12, P9_14, P9_23, P8_11, P8_13, P8_15, P8_17, P8_16, P8_12, P8_14, P8_12.

4) Draw FRC pin Diagram with pins on it.



V_{cc}	26	25	GND
P8_21	24	23	P8_19
P9_12	22	21	P9_11
P9_14	20	19	P9_13
P9_16	18	17	P9_15
P9_23	16	15	P9_24
P8_12	14	13	P8_11
P8_14	12	11	P8_13
P8_16	10	09	P8_15
P8_18	08	07	P8_17
P8_7	06	05	P8_8
P8_9	04	03	P8_10
P9_18	02	01	P9_17

5) Write sample code for Dancing LED.

```
import Adafruit_BBIO.GPIO as GPIO
GPIO.setup(“P9_15”,GPIO.OUT)
while True:
GPIO.output(“P9_15”,GPIO.HIGH)
print(“high”)
time.sleep(5)
GPIO.output(“P9_15”,GPIO.LOW)
print(“low”)
time.sleep(5)
```

6) Attach Program for traffic signals with program output as HIGH, LOW after every 5 seconds.

```
#Program for Traffic light controller

import Adafruit_BBIO.GPIO
import time

GPIO.setup("P9_11", GPIO.OUT)
GPIO.setup("P9_13", GPIO.OUT)
GPIO.setup("P9_15", GPIO.OUT)
GPIO.setup("P9_24", GPIO.OUT)
GPIO.setup("P9_12", GPIO.OUT)
GPIO.setup("P9_14", GPIO.OUT)
GPIO.setup("P9_16", GPIO.OUT)
GPIO.setup("P9_23", GPIO.OUT)

GPIO.setup("P9_11", GPIO.OUT)
GPIO.setup("P9_13", GPIO.OUT)
GPIO.setup("P9_15", GPIO.OUT)
GPIO.setup("P9_17", GPIO.OUT)
GPIO.setup("P9_14", GPIO.OUT)
GPIO.setup("P9_16", GPIO.OUT)
GPIO.setup("P9_18", GPIO.OUT)
GPIO.setup("P9_12", GPIO.OUT)

while true:
    num = 0
    GPIO.output("P8_16", GPIO.LOW)
    GPIO.output("P8_13", GPIO.LOW)
    GPIO.output("P8_18", GPIO.LOW)
    GPIO.output("P8_15", GPIO.LOW)
    GPIO.output("P8_17", GPIO.LOW)
    GPIO.output("P9_12", GPIO.LOW)
    GPIO.output("P9_14", GPIO.LOW)
    GPIO.output("P9_15", GPIO.HIGH)
    GPIO.output("P9_24", GPIO.HIGH)
    GPIO.output("P8_14", GPIO.HIGH)
    GPIO.output("P8_12", GPIO.HIGH)
    GPIO.output("P8_11", GPIO.HIGH)
    GPIO.output("P8_23", GPIO.HIGH)
    time.sleep(8)
    while num<5:
        GPIO.output("P8_16", GPIO.LOW)
        GPIO.output("P9_13", GPIO.LOW)
        time.sleep(1)
        GPIO.output("P8_16", GPIO.HIGH)
        GPIO.output("P9_13", GPIO.HIGH)
        time.sleep(1)
        num = num + 1
```

7) Steps to copy file from PC to BeagleBone Black

1. Edit your IOTL program in gedit and save file as test.py
2. Open a new terminal
3. Type ‘su’ and input password
4. Now use sftp command connect to BBB as sftp <ip addr of BBB>
5. Now use put command to copy file from PC to BBB – put test.py
6. Now for BBB terminal compile file as – python test.py
7. Every time you modify test.py on PC perform step 4 to upload changes to BBB.