American International University- Bangladesh

Department of Computer Science

Lab Report Cover Sheet

Course Name	MICROPROCESSOR AND EMBEDDED SYSTEMS
Lab Report No.	09
Lecturer Name	**MD. ALI NOOR**
Semester	SUMMER 2021-22
Submission Date	11/08/2022
Section	O
Group No.	03

Student Name	Student ID	Contribution (out of 100%)
1. EMON, MEHEDI HASAN	19-41118-2	
2. ASIF AHAMMED	19-40020-1	
3. EASINUL ABEDIN SAYEM	19-40291-1	
4. MD REDOY SHEIKH	19-40101-1	
5. MD. NAZIM HASAN	19-40118-1	
6. MOHI UDDIN ANANDO	20-43585-1	

Title: Familiarization with Raspberry Pi

Abstract:

The objective of this experiment is to get familiarized with Raspberry Pi, a really small, really powerful, 'micro-computer' and learn how to simply glow an LED.

Theory and Methodology:

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries.

Hardware

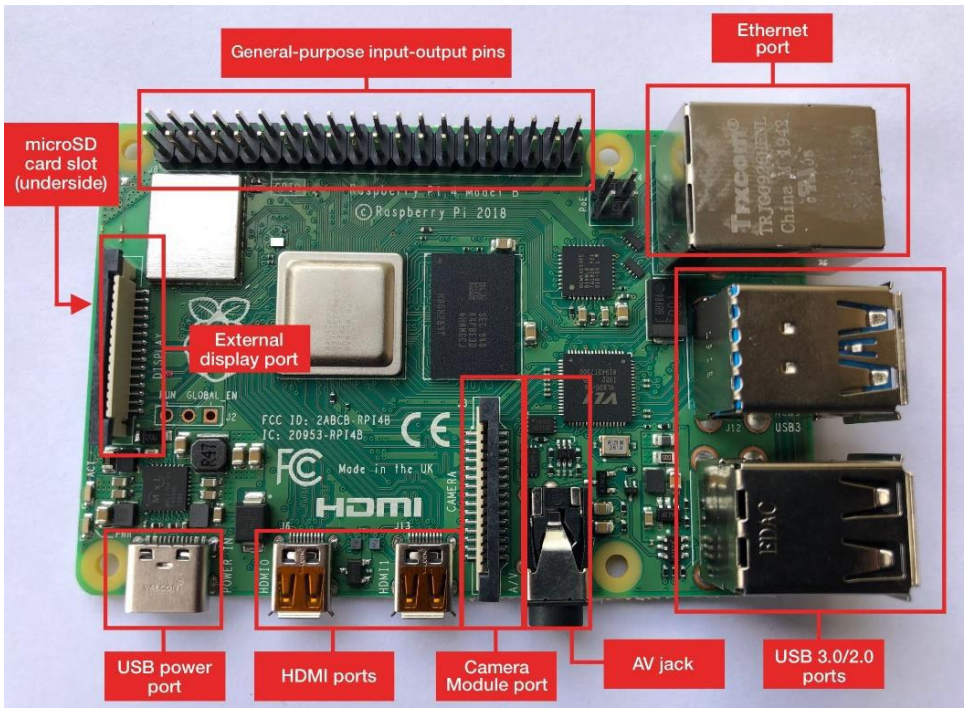

Figure 1: Raspberry Pi 3 - Model B

Technical Specification

1. Processor

• Broadcom BCM2387 chipset.

• 64-bit 1.2GHz Quad-Core ARM Cortex-A53.

2. 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)

• IEEE 802.11 b / g / n Wi-Fi. Protocol: WEP, WPA WPA2, algorithms AES-CCMP (maximum key length of 256 bits), the maximum range of 100 meters.

• IEEE 802.15 Bluetooth, symmetric encryption algorithm Advanced Encryption Standard (AES)with 128-bit key, the maximum range of 50 meters.

3. GPU

• Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware accelerated Open VG, and 1080p30 H.264 high-profile decode.

• Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure

4. Memory

• 1GB LPDDR2.

5. Operating System

• Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT

6. Dimensions

• 85 x 56 x 17mm

7. Power

• Micro USB socket 5V1, 2.5A

8. Ethernet

• 10/100 Base T Ethernet socket.

9. Video Output

• HDMI (rev 1.3 & 1.4)

• Composite RCA (PAL and NTSC)

10. Audio Output

• Audio Output 3.5mm jack

• HDMI

• USB 4 x USB 2.0 Connector

11. GPIO Connector

• 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip

• Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines

Figure 2: Raspberry Pi 3 - Model B GPIO pin

12. Camera Connector

• 15-pin MIPI Camera Serial Interface (CSI-2)

13. Display Connector

• Display Serial Interface (DSI) 15-way flat flex cable connector with two data lanes and a clock lane

14. USB

• Four built-in USB ports provide enough connectivity for a mouse, keyboard, or anything else that you feel the RPi needs.

15. Antenna

• There's no need to connect an external antenna to the Raspberry Pi 3. Its radios are connected to this chip antenna soldered directly to the board

16. HDMI connector:

• The HDMI port provides digital video and audio output.14 different video resolutions are supported, and the HDMI signal can be converted to DVI (used by many monitors), composite (analog video signal usually carried over a yellow RCA connector), or SCART (a European standard for connecting audio-visual equipment) with external adapters

17. Status LED

There are five status LEDs on the corner of the board.

ACT	Green	Lights when the SD card is accessed/used
PWR	Red	steady ON when Pi is connected to 3.3V power

FDX	Green	On if network adapter is full duplex
LNK	Green	Network activity light On when Ethernet is connected
100	Yellow	On if the network connection is 100Mbps

The status LEDs give information about the operating condition and any problems of the board e.g.

Status LED	Possible Problem
Red power LED does not light, nothing on display	The power is not properly connected
The LED light is blinking	The red power LED should never blink. Blinking RED LED (PWR) means the 5V power supply is dropping out. Use a different power supply.
Red power LED is on, green LED is glowing faintly and steadily	Power supply is OK. But faint and steady green light (ACT) means SD card has some problem in starting the operating system (no boot code).

LAB Task: Design a traffic control system using RED, YELLOW, and GREEN LEDs.

In this experiment an LED will be controlled by using Raspberry Pi. Python will be used to blinking Red, Yellow and Green LEDs to create a traffic control system. This experiment will give a basic idea of Python language as well as importing GPIO pins of Raspberry Pi.

Equipment List:

1) Activated Raspberry pi

2) LED (Red, Yellow, Green)

3) Resistor (220)

4) Breadboard

5) Jumper wires

Experimental Procedure:

At first, we open the proteus 8.9 software. And then we create a new project withschematic from the selected template. We are not creating any PCB layout. And, then create a flowchart project. Because we implement this experiment using flowchart. We select one Raspberry pi 3, Raspberry pi LED Red Breakout Board, Raspberry pi LED Yellow Breakout Board, Raspberry pi LED Green Breakout Board for Schematic Capture. Then we design flowchart in the Visual designer. In setup function, all three LEDS are turn off. Then in the loop function Firstly we use red led on, then give 5s time delay and after that we use red led off. Then we give 1s time delay. Secondly, we use yellow led on, then give 3s time delay and after that we use yellow led off.

Then we give 1s time delay. Thirdly, we use green led on, then give 4s time delay and after that we use red led off. Then we give 1s time delay.

Hardware Set-Up:

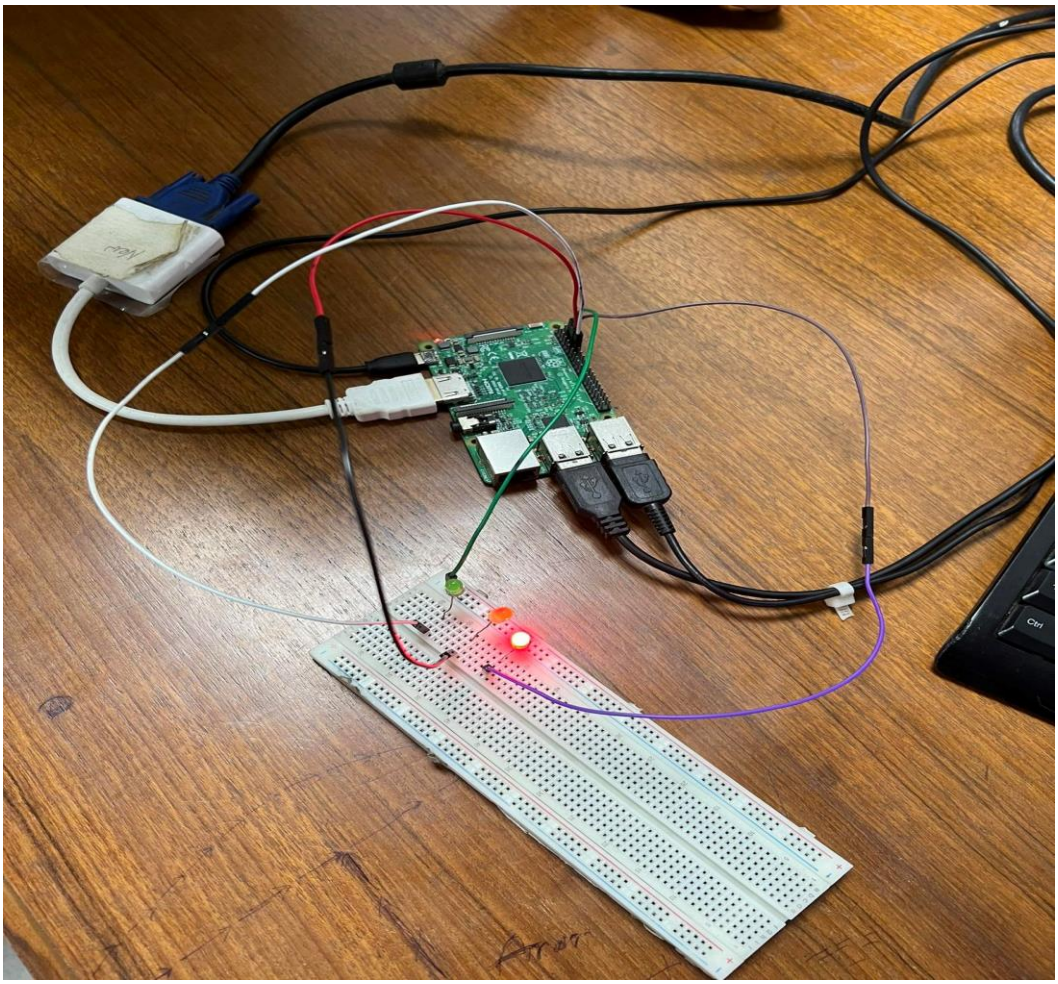

Figure 3: Blinking of Red LED

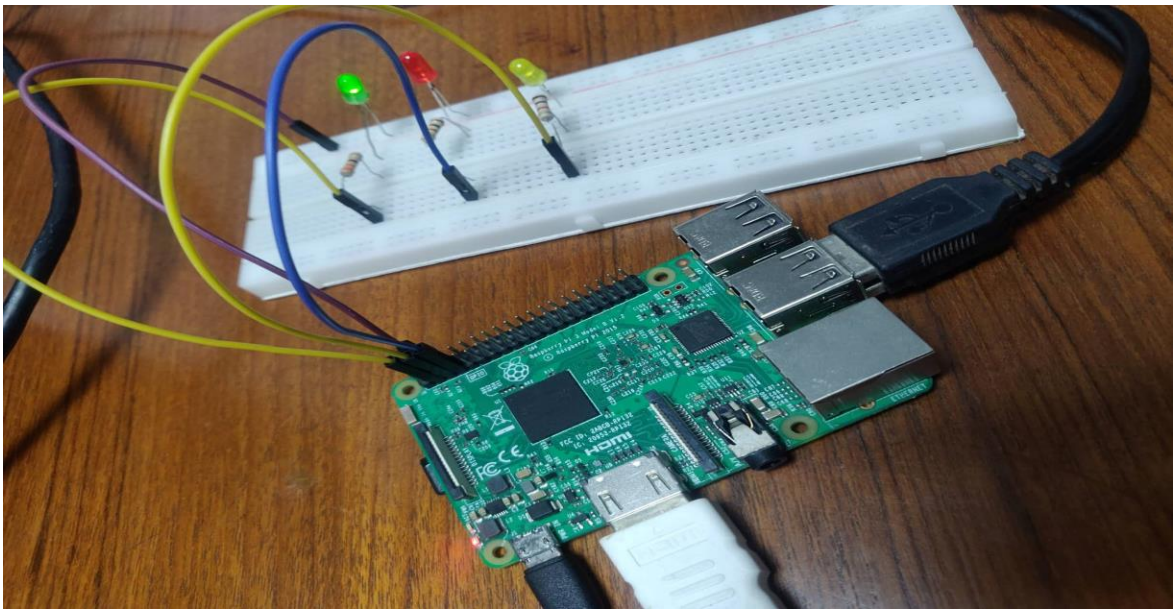

Figure 4: Blinking of Green LED

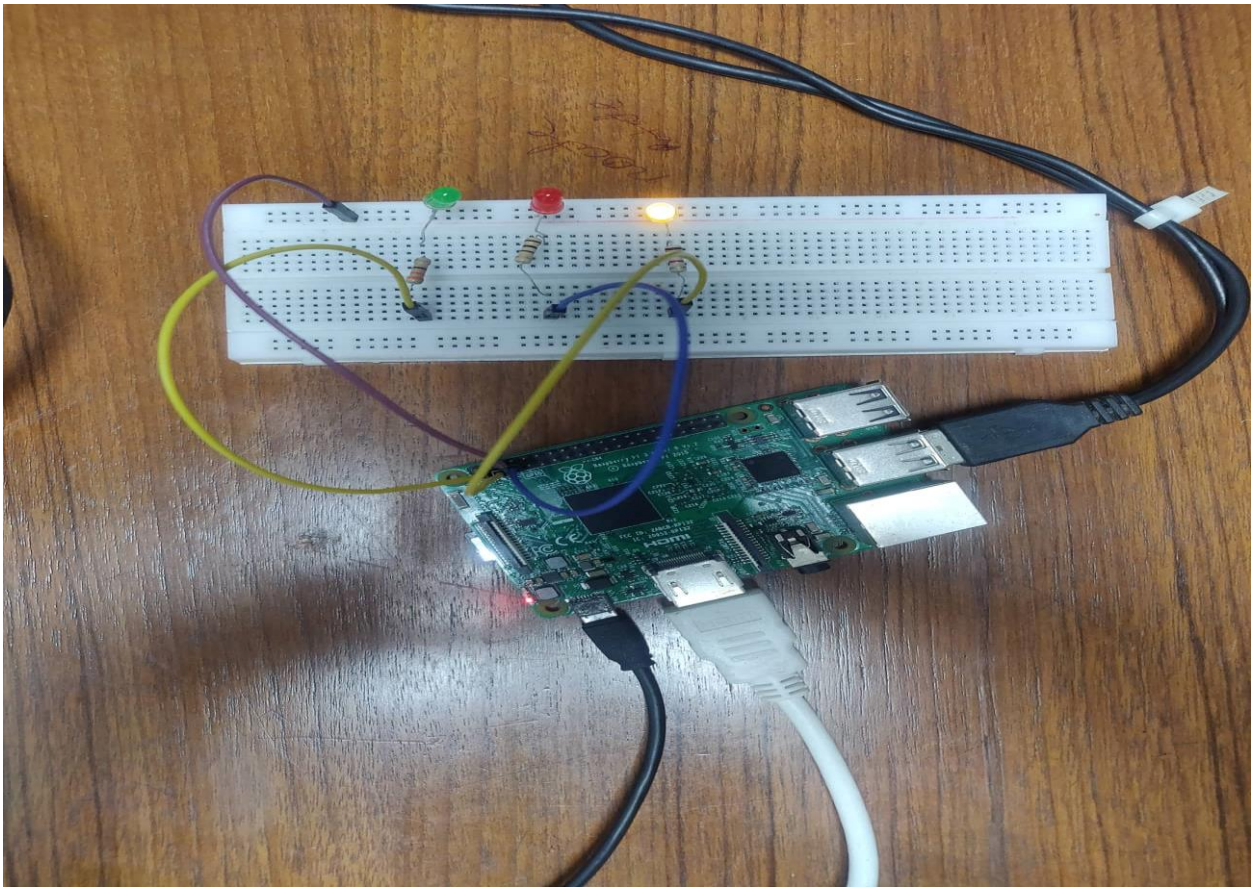

Figure 5: Blinking of Yellow LED

Program/Lab Code:

```
import Rp1.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(3, GPIO,OUT)
GPIO.setup(5, GPIO,OUT)
GPIO.setup(7, GPIO,OUT)
#Forever loop
While True:
        GPIO.output(3, GPIO, HIGH)
        time.sleep(5)
      GPIO.output(3, GPIO, LOW)
      GPIO.output(5, GPIO, HIGH)
      time.sleep(2)
      GPIO.output(5, GPIO, LOW)
      GPIO.output(7, GPIO, HIGH)
      time.sleep (6)
      GPIO.output(7, GPIO, LOW)

pi@raspberrypi: ~ $ nano blinl1.py
pi@raspberrypi: ~ $ sudo python blinl1.py
  File “blinl1.py”. line l1
    GPIO.output( 3,GPIO,HIGH)
IndentationError: expected an indented block
pi@raspberrypi: ~ $ nano blinl1.py
pi@raspberrypi: ~ $ sudo python blinl1.py
  File “blinl1.py”. line l1
    GPIO.output( 5,GPIO,HIGH)
```

pi@raspberrypi: ~ $ nano blinl1.py

pi@raspberrypi: ~ $ sudo python blinl1.py

File “blinl1.py”. line l1

GPIO.output(7,GPIO,HIGH)

IndentationError: expected an indented block

pi@raspberrypi: ~ $ nano blinl1.py

pi@raspberrypi: ~ $ sudo python blinl1.py

Simulation:

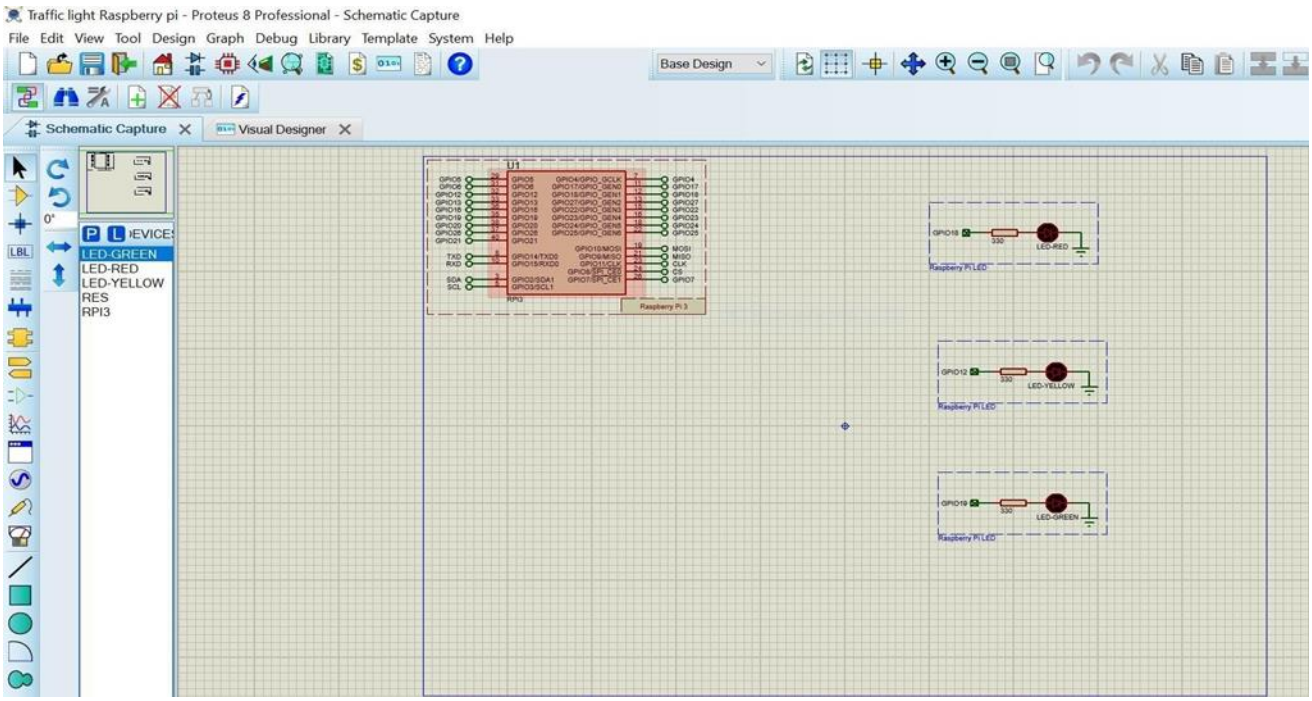

Figure 6:Schematic Capture (Traffic light management system)

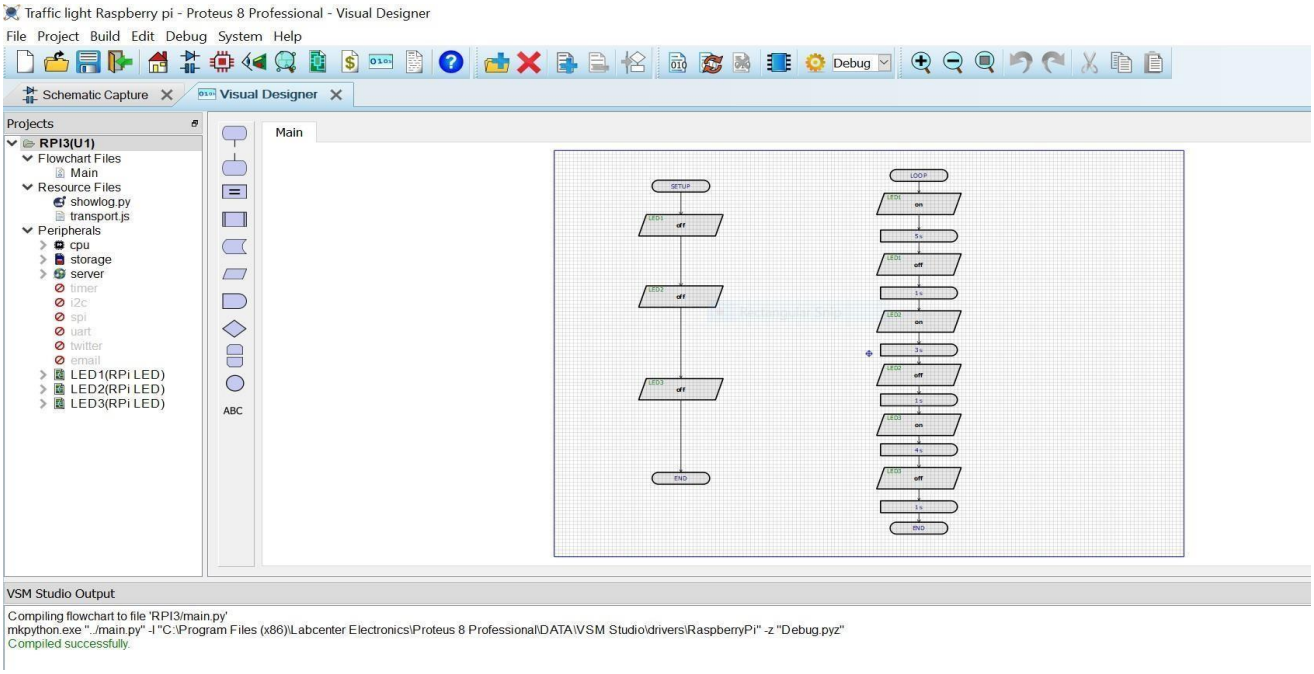

Figure 7: Visual designer (Traffic light management system)

Results:

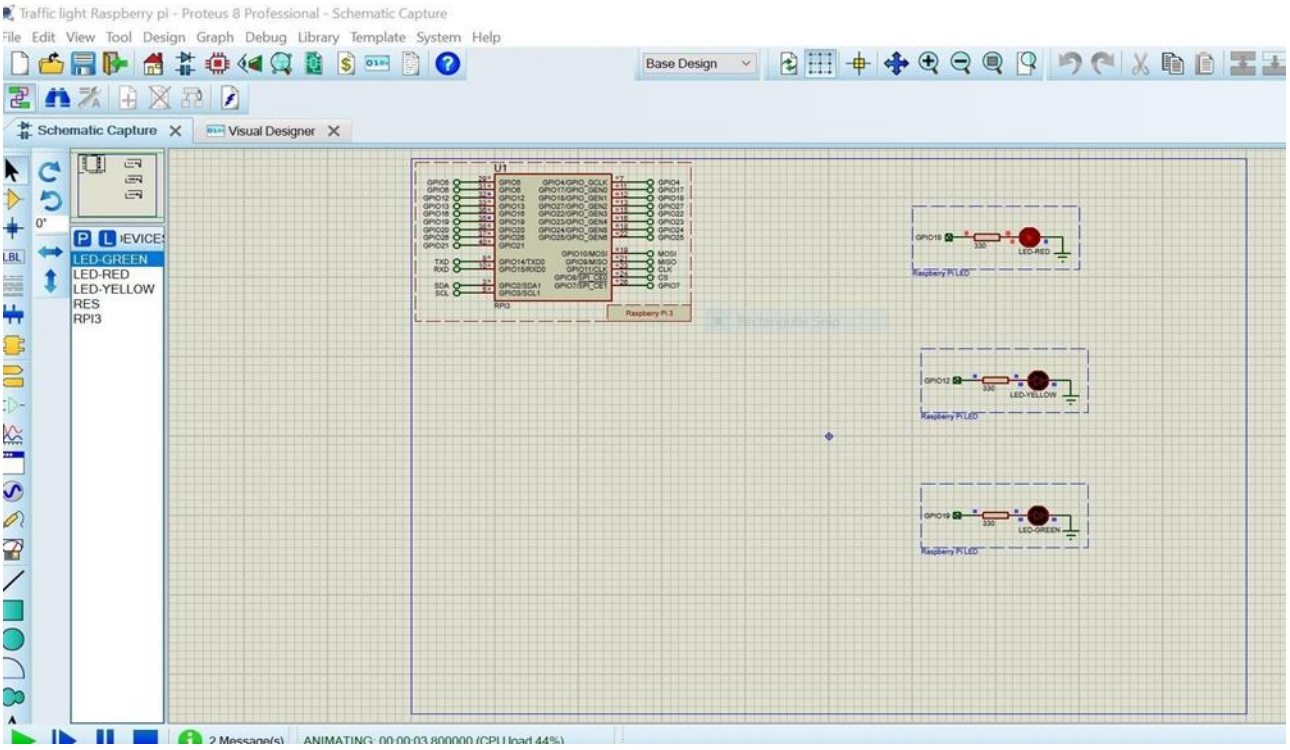

Figure 8: Traffic light management system (RED LED turn on)

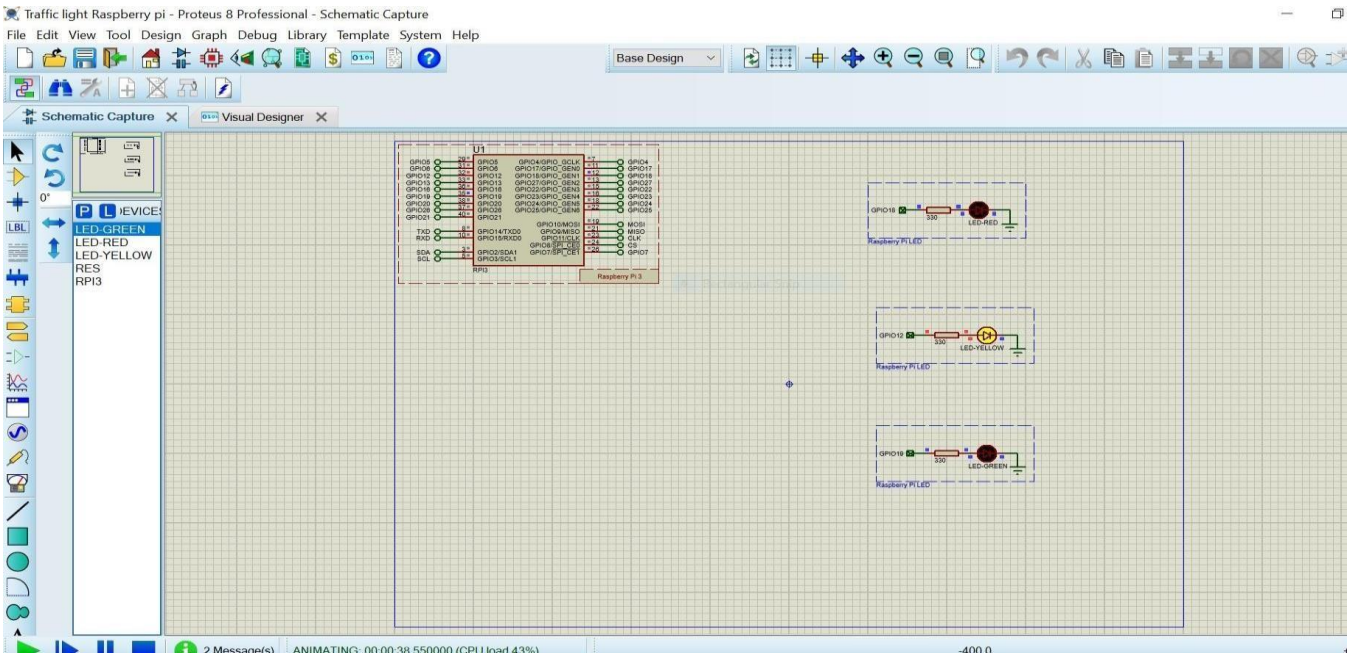

Figure 9: Traffic light management system (Yellow led turn on)

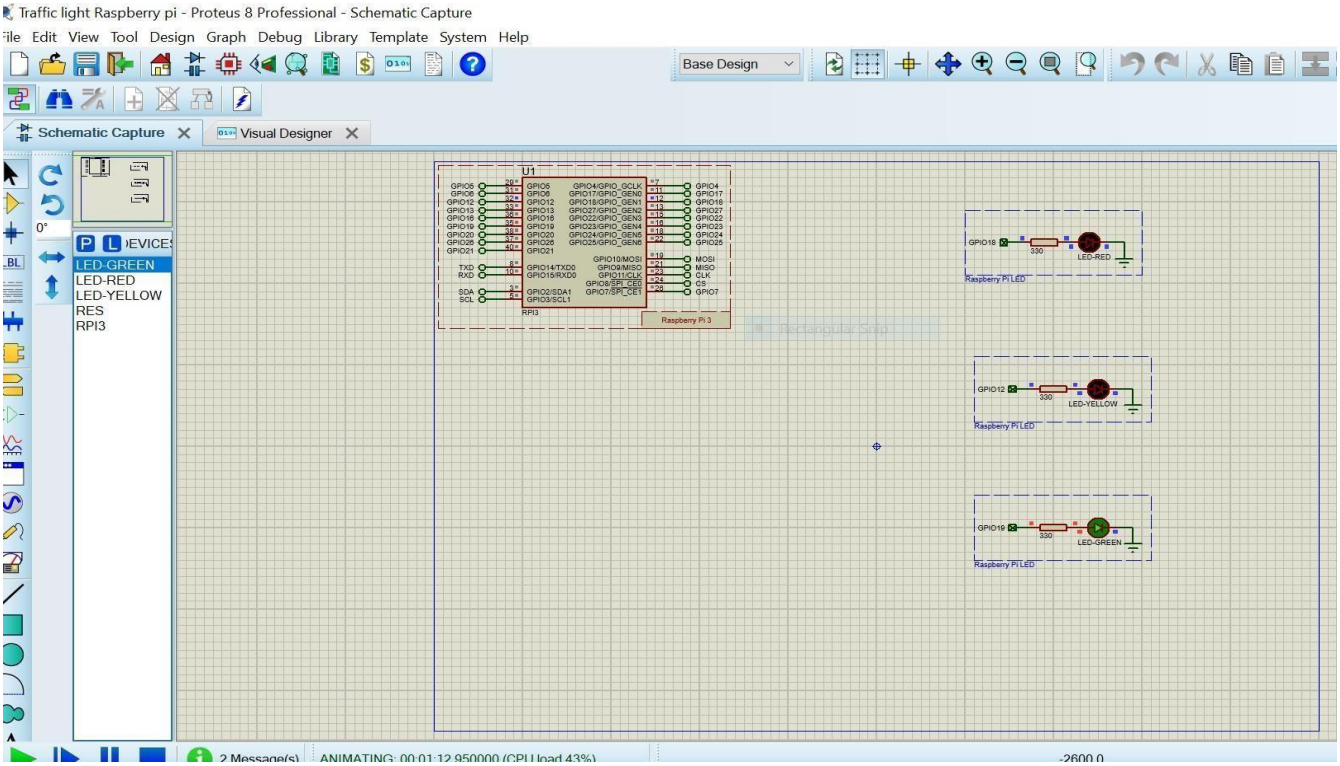

Figure 10: Traffic light management system (Green LED turn on)

Discussion:

In this Traffic light management system experiment, we can see in result section that, our project run perfectly. At first the Red LED turned on for 5 second. Because at first, we want to turn on the RED LED for traffic management system. Then it turned off. Then the Yellow LED turn on and it stayed for 3 second. Then it turned off. After that the Green LED turned on and stayed for 4 second. Then it turned off. This will move on continuously.

Conclusion:

Despite various setbacks, we were able to obtain the desired outcome, and the experiment was successfully carried out inside the proteus. Due to the success of the experiment, the traffic management system was created using Raspberry pi.

Reference(s):

1) Raspberry pi datasheet.
2) https://www.raspberrypi.org/documentation/linux/
3) https://www.raspberrypi.org/documentation/remote-access/ssh/
4) https://www.raspberrypi.org/documentation/remote-access/ssh/scp.md
5) https://www.raspberrypi.org/documentation/linux/usage/root.md
6) https://www.raspberrypi.org/documentation/usage/python