

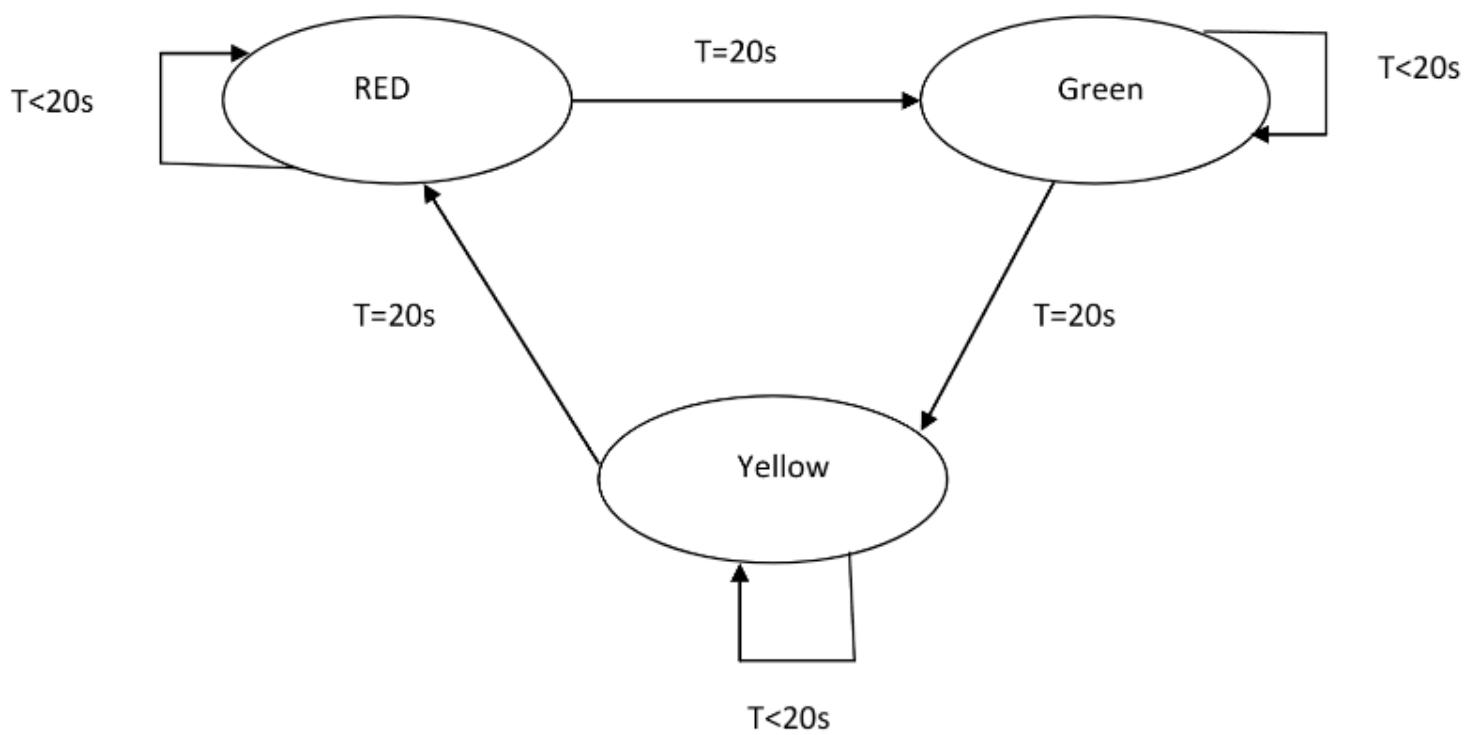
EMBEDDED SYSTEMS  
TRAFFIC LIGHT

BY

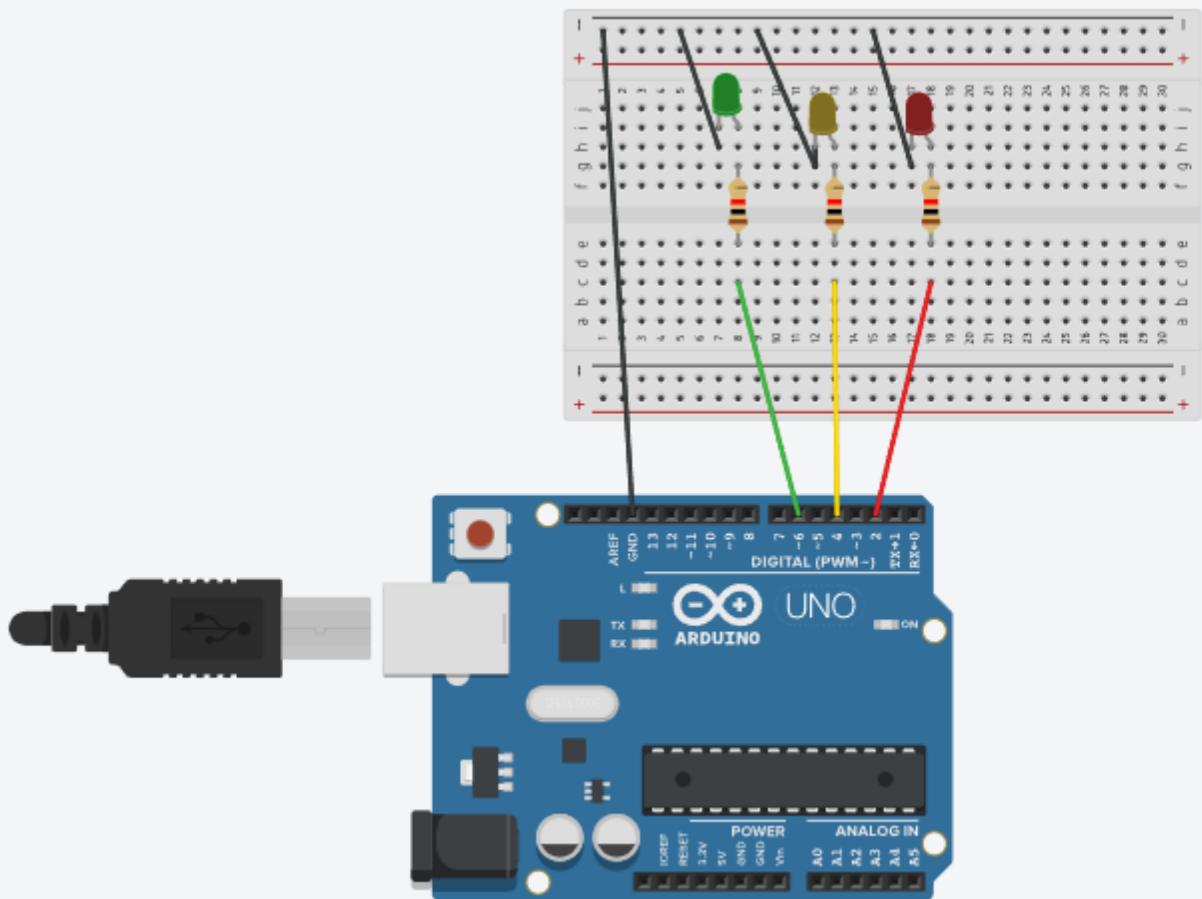
SYED MUHAMMAD ABIS RIZVI  
IBRAHIM ABDEMALEK

GITHUB-LINK: [https://github.com/ibrahimabdelmalek31/Embedded-  
Systems-Team-D01](https://github.com/ibrahimabdelmalek31/Embedded-Systems-Team-D01)

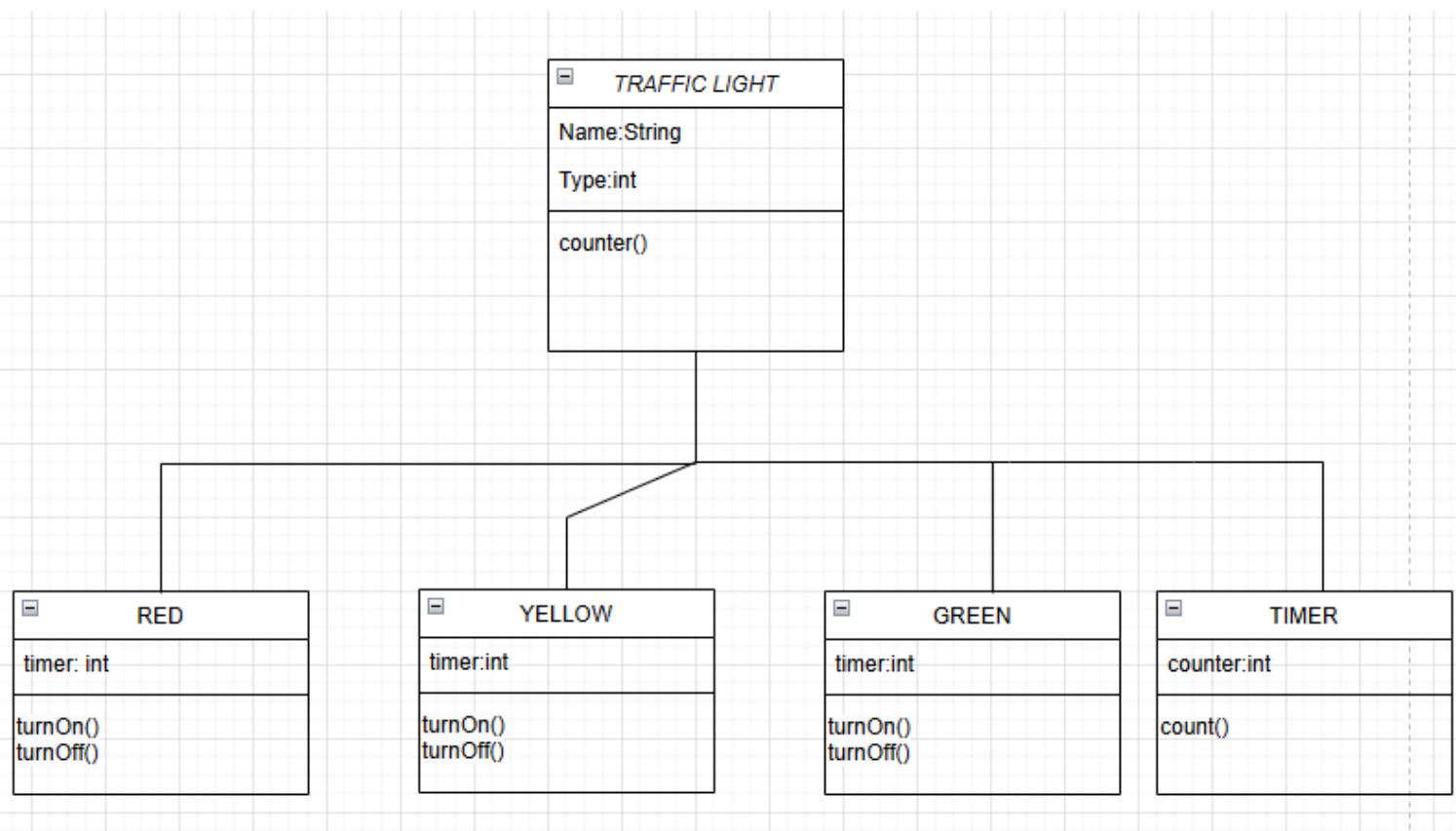
## STATE MACHINE DIAGRAM TASK 1



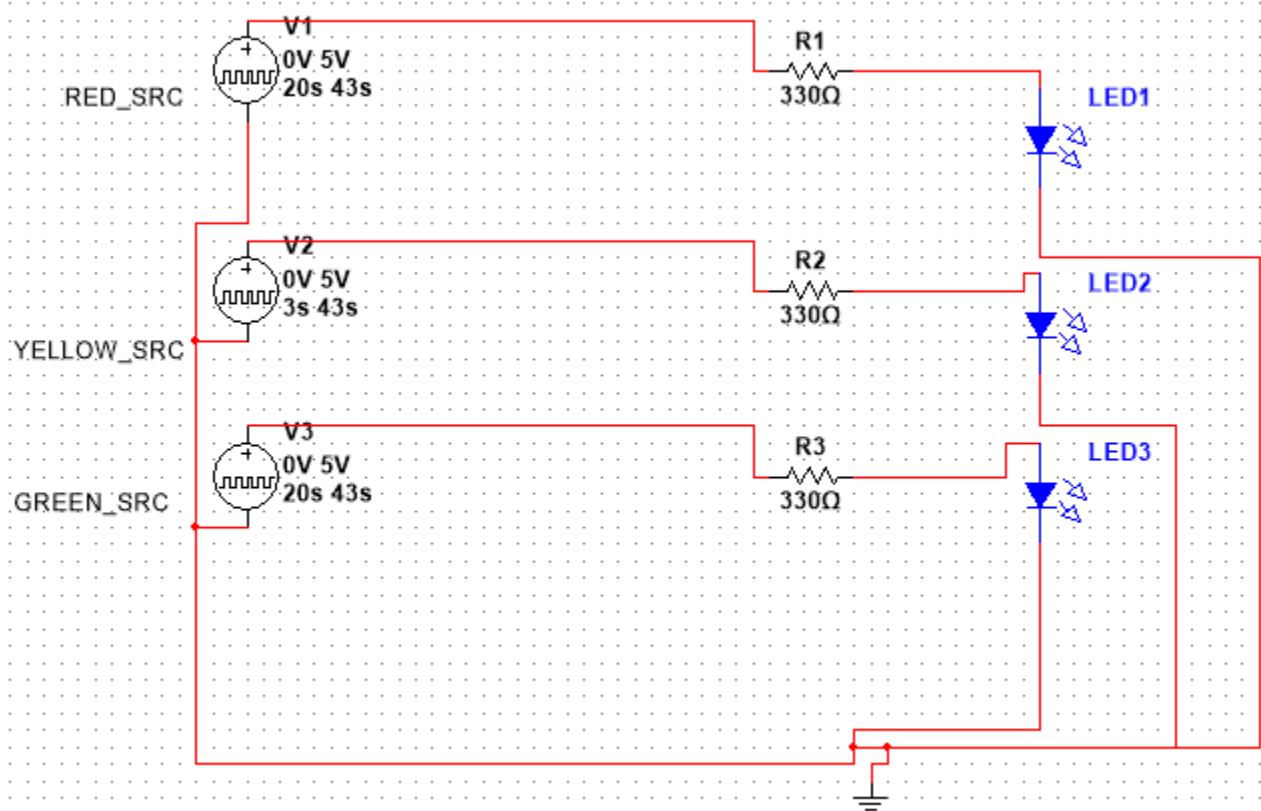
## TASK 1



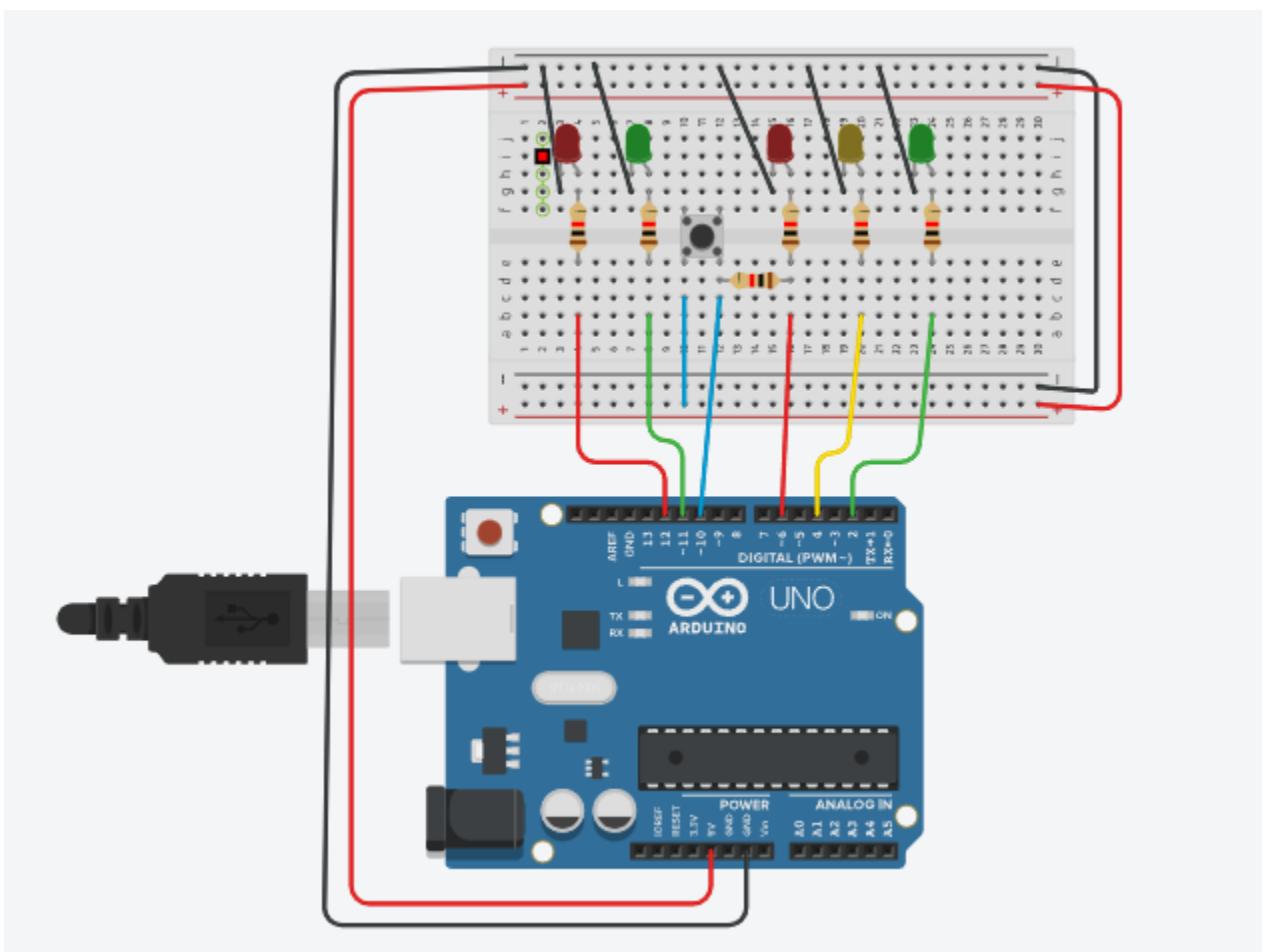
## CLASS DIAGRAM TASK 1



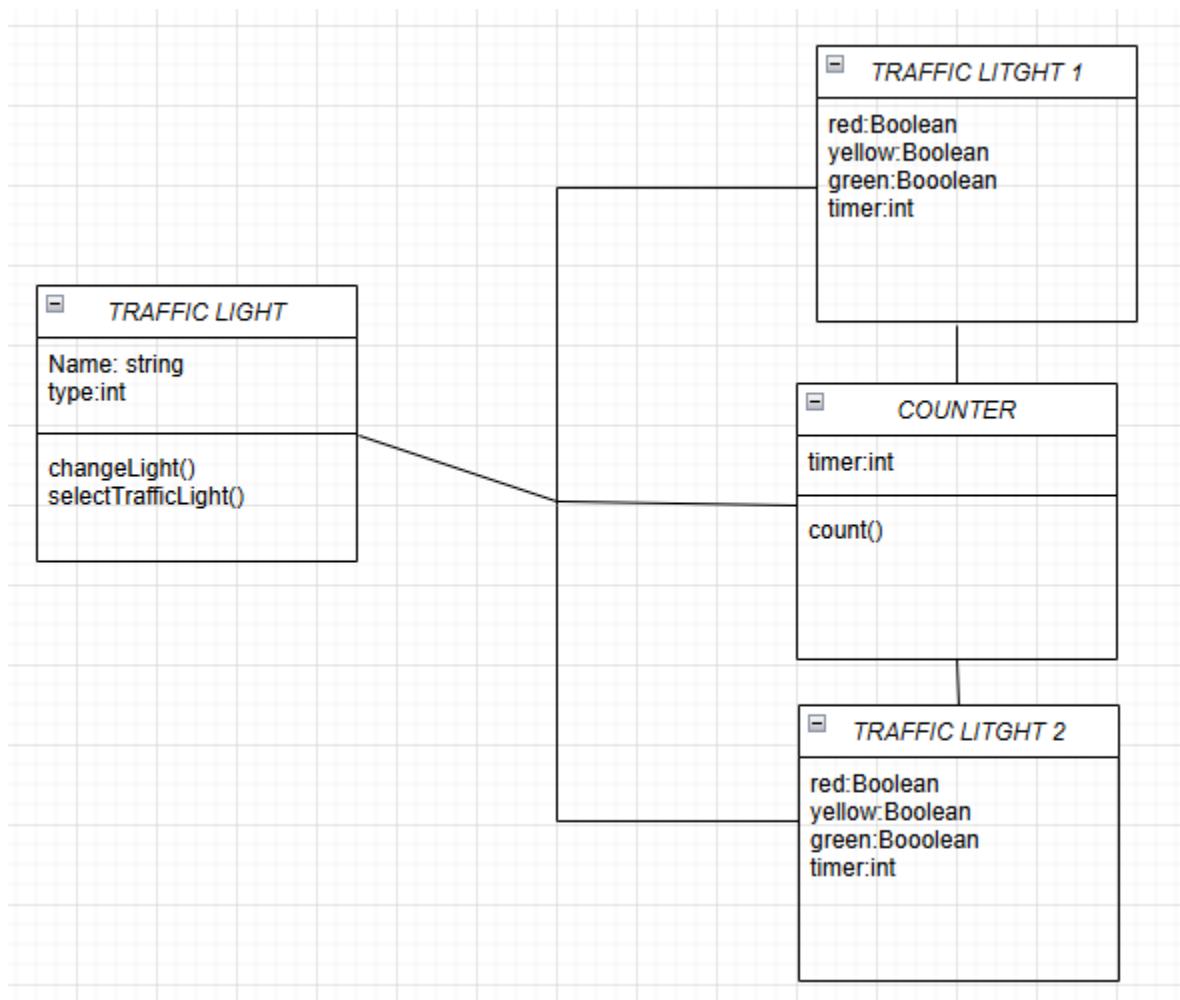
## MULTI-SIM IMPLEMENTATION



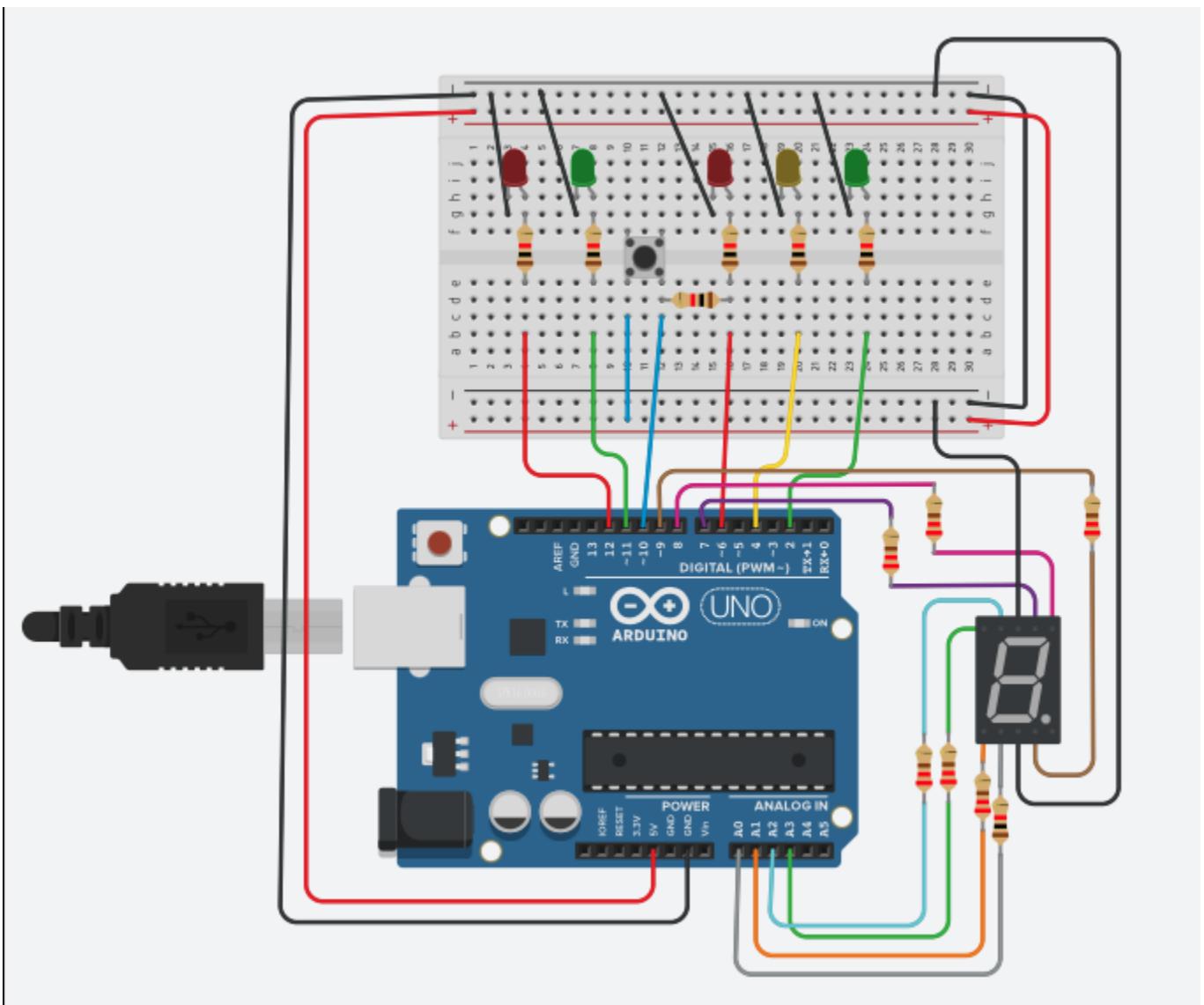
## TASK 2



## CLASS DIAGRAM TASK 2



## TASK 2 WITH SEVEN SEGMENT



## SEGMENT IMPLEMENTATION CODE

Simulator time: 00:00:03      [Code](#) [Stop Simulation](#) [Send To](#)

1 (Arduino Uno R3) ▶

```
// ----- ENUM & VARIABLES -----
enum Status {green, yellow, red, pgreen, pred};
Status status = green;

unsigned long lastMillis = 0;
unsigned long interval = 10000;
bool pedestrianRequest = false;

// ----- 7-SEGMENT SETUP -----
int segPins[7] = {7, 8, 9, A0, A1, A2, A3}; // Segment order: a,b,c,d,e,f,g
byte digits[10][7] = {
    {1,1,1,1,1,0}, {0,1,1,0,0,0,0}, {1,1,0,1,1,0,1},
    {1,1,1,1,0,0,1}, {0,1,1,0,0,1,1}, {1,0,1,1,0,1,1},
    {1,0,1,1,1,1,1}, {1,1,1,0,0,0,0}, {1,1,1,1,1,1,1},
    {1,1,1,1,0,1,1}
};

// ----- PROTOTYPES -----
void handleTrafficLights(unsigned long currentMillis);
void handlePedestrianLights(unsigned long currentMillis);
void displayDigit(int num);
void clearDisplay();
void displayCountdown(unsigned long currentMillis);

// ----- SETUP -----
void setup() {
    pinMode(12, OUTPUT); // Ped red
    pinMode(11, OUTPUT); // Ped green
    pinMode(10, INPUT); // Button
    pinMode(6, OUTPUT); // Traffic red
    pinMode(4, OUTPUT); // Traffic yellow
    pinMode(2, OUTPUT); // Traffic green

    for (int i = 0; i < 7; i++) pinMode(segPins[i], OUTPUT);
}

// Initial states: traffic green, ped red
digitalWrite(2, HIGH);
digitalWrite(4, LOW);
digitalWrite(6, LOW);
digitalWrite(11, LOW);
digitalWrite(12, HIGH);

lastMillis = millis();
}

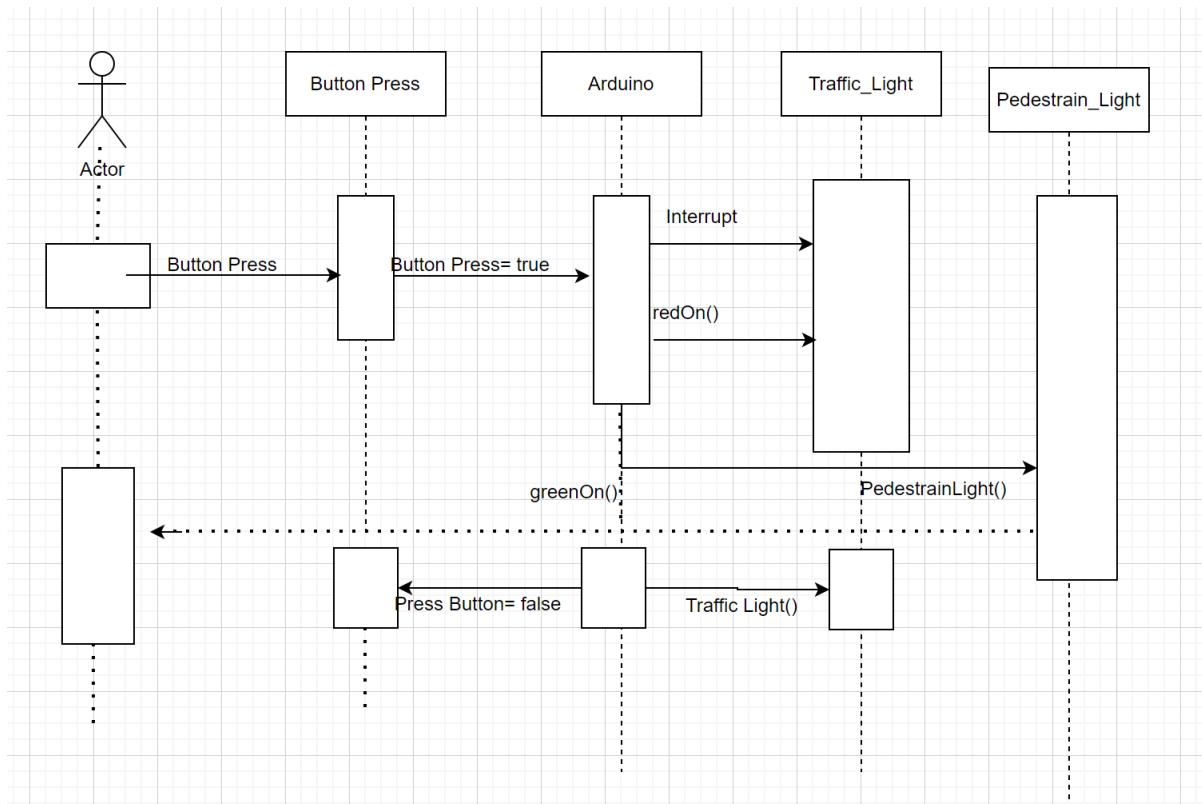
// ----- LOOP -----
void loop() {
    unsigned long currentMillis = millis();
    int buttonState = digitalRead(10);

    static unsigned long lastButtonPress = 0;
    if (buttonState == HIGH && !pedestrianRequest && status != pgreen) {
        currentMillis - lastButtonPress > 500) {
            pedestrianRequest = true;
            lastButtonPress = currentMillis;
            lastMillis = currentMillis;
            status = pgreen;
            interval = 3200;
        }
    }
}
```

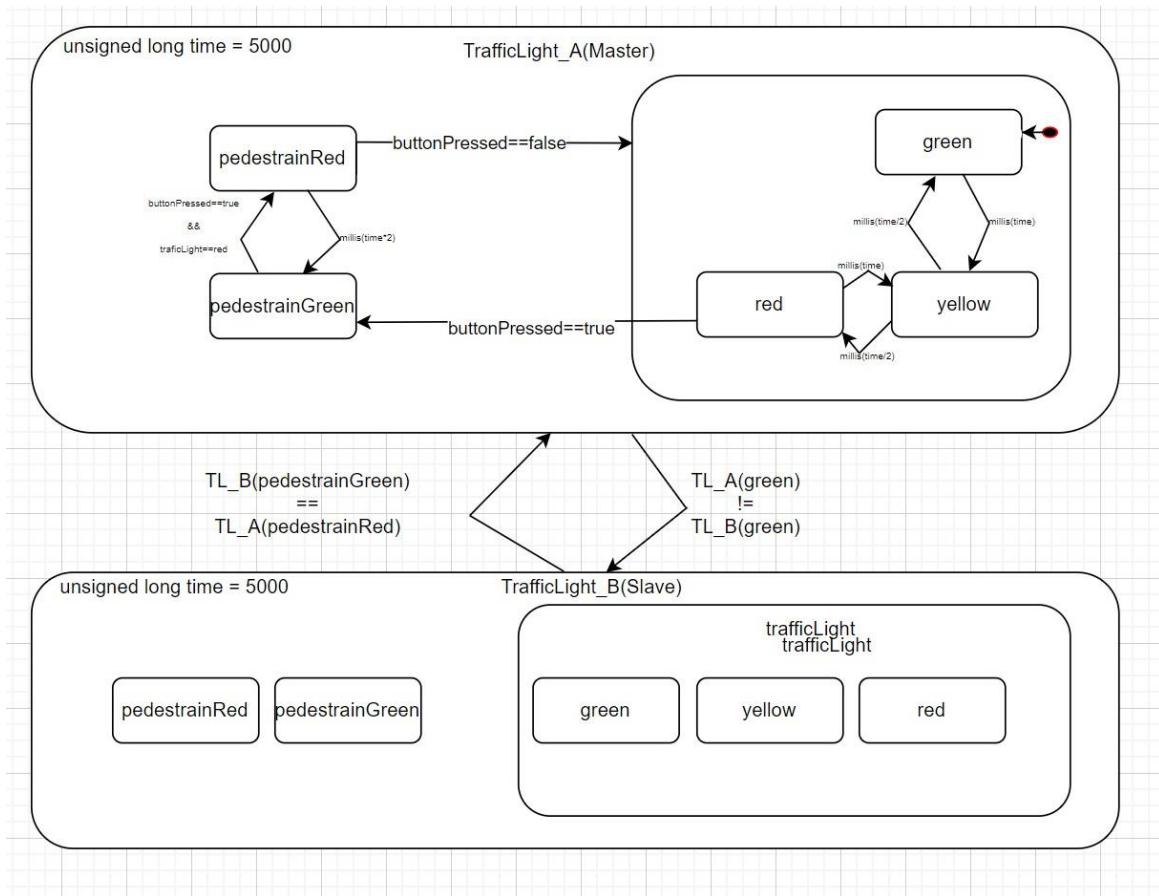
Serial Monitor

## TASK 5

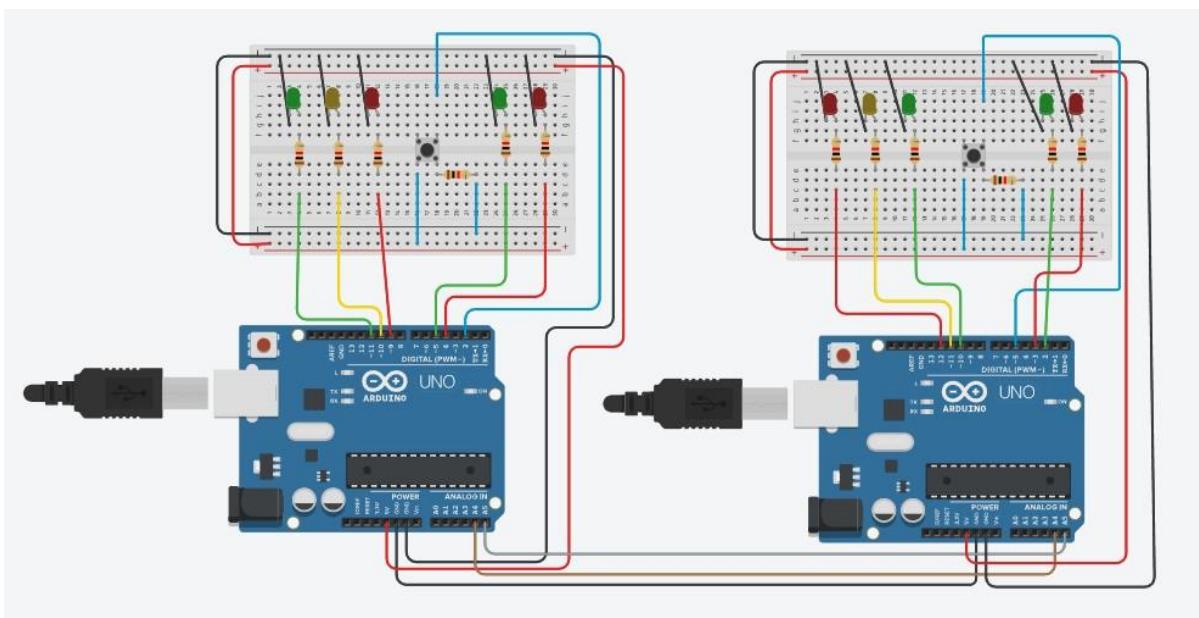
### SEQUENCE DIAGRAM



## STATE MACHINE DIAGRAM



## TRAFFIC LIGHT HAVING TWO MICROCONTROLLERS



MASTER -CODE

## SLAVE-CODE

```

task_5BSlave.ino

1 #include <Wire.h>
2
3 // C++ code
4 volatile boolean pushButton;
5 int redState = 1; // Start in red state for traffic lights
6 enum State {red, yellow, green, pedestrianGreen, pedestrianRed};
7 State _currentState = green;
8 State _targetState = pedestrianGreen;
9 unsigned long time = 5000;
10 unsigned long previousMillis = 0; // Used for timing (non-blocking)
11 unsigned long pedestrianMillis = 0; // Used for pedestrian light timing
12 int set;
13 int event;
14
15 void setup() {
16   Wire.begin(1);
17   Wire.onReceive(receiveEvent); // Set up Wire communication
18   Serial.begin(9600);
19
20   // Pin assignments for the lights
21   pinMode(3, OUTPUT); // Pedestrian red light
22   pinMode(2, OUTPUT); // Pedestrian green light
23   pinMode(12, OUTPUT); // Traffic red light
24   pinMode(11, OUTPUT); // Traffic yellow light
25   pinMode(10, OUTPUT); // Traffic green light
26   pinMode(5, INPUT); // Button to trigger pedestrian light
27   pinMode(4, OUTPUT); // Traffic red light (alternate pin)
28
29   // Attach interrupt for button press
30   attachInterrupt(digitalPinToInterrupt(5), buttonPressed, RISING);
31 }
32
33 void loop() {
34   unsigned long currentMillis = millis(); // Get current time in milliseconds
35
36   if (redState == 0) {
37     pedestrian(currentMillis); // Handle pedestrian light logic
38
39     digitalWrite(12, LOW); // Traffic red light off
40     previousMillis = currentMillis;
41     if (redState == 1) {
42       _currentState = yellow; // Transition to yellow
43     } else {
44       _currentState = pedestrianGreen; // Transition to pedestrian green
45     }
46     set = 1;
47   }
48   break;
49 }
50
51 void pedestrian(unsigned long currentMillis) {
52   switch (_targetState) {
53     case pedestrianGreen:
54       digitalWrite(12, HIGH); // Traffic red light on
55       digitalWrite(2, HIGH); // Pedestrian green light on
56       if (currentMillis - pedestrianMillis >= time / 5) {
57         digitalWrite(2, LOW); // Pedestrian green light off
58         pedestrianMillis = currentMillis;
59         redState = 1; // Set redState to 1 after pedestrian light
60         _targetState = yellow; // Transition to yellow pedestrian light
61       }
62     break;
63
64     case yellow:
65       digitalWrite(12, HIGH); // Traffic red light on
66       digitalWrite(11, HIGH); // Traffic yellow light on
67       if (currentMillis - pedestrianMillis >= time / 2) {
68         digitalWrite(11, LOW); // Traffic yellow light off
69         pedestrianMillis = currentMillis;
70         _targetState = pedestrianRed;
71       }
72     break;
73
74     case pedestrianRed:
75       digitalWrite(11, HIGH); // Traffic yellow light on
76       if (currentMillis - pedestrianMillis >= time / 2) {
77         digitalWrite(11, LOW); // Traffic yellow light off
78         pedestrianMillis = currentMillis;
79         _targetState = pedestrianGreen; // Set to pedestrian green state
80       }
81     break;
82   }
83 }
84
85 void pedestrian(unsigned long currentMillis) {
86   switch (_targetState) {
87     case pedestrianGreen:
88       digitalWrite(12, HIGH); // Traffic red light on
89       digitalWrite(2, HIGH); // Pedestrian green light on
90       if (currentMillis - pedestrianMillis >= time / 5) {
91         digitalWrite(2, LOW); // Pedestrian green light off
92         pedestrianMillis = currentMillis;
93         redState = 1; // Set redState to 1 after pedestrian light
94         _targetState = yellow; // Transition to yellow pedestrian light
95       }
96     break;
97
98     case yellow:
99       digitalWrite(12, HIGH); // Traffic red light on
100      digitalWrite(11, HIGH); // Traffic yellow light on
101      if (currentMillis - pedestrianMillis >= time / 2) {
102        digitalWrite(11, LOW); // Traffic yellow light off
103        pedestrianMillis = currentMillis;
104        _targetState = pedestrianRed;
105      }
106    break;
107
108    case pedestrianRed:
109      digitalWrite(11, HIGH); // Traffic yellow light on
110      if (currentMillis - pedestrianMillis >= time / 2) {
111        digitalWrite(11, LOW); // Traffic yellow light off
112        pedestrianMillis = currentMillis;
113        _targetState = pedestrianGreen; // Set to pedestrian green state
114      }
115    break;
116  }
117 }
118
119 void buttonPressed() {
120   // When the button is pressed, set the pedestrian phase
121   if (digitalRead(5) == HIGH) { // Button pressed (active HIGH)
122     redState = 0; // Trigger pedestrian phase
123     _targetState = pedestrianGreen; // Set to pedestrian green state
124     previousMillis = millis(); // Reset the timing
125   }
126 }
127
128 void receiveEvent(int event) {
129   // Handle Wire communication event and set redState
130   redState = Wire.read();
131 }
132
133 }

pedestrian(currentMillis); // Handle pedestrian light logic
}
if (redState == 1) {
  trafficLight(currentMillis); // Handle traffic light logic
}

oid trafficLight(unsigned long currentMillis) {
switch (_currentState) {
  case green:
    digitalWrite(10, HIGH); // Traffic green light on
    digitalWrite(12, HIGH); // Pedestrian red light on
    if (currentMillis - previousMillis >= time) {
      digitalWrite(10, LOW); // Traffic green light off
      previousMillis = currentMillis;
      _currentState = yellow; // Transition to yellow
      set = 0;
    }
  break;

  case yellow:
    digitalWrite(11, HIGH); // Traffic yellow light on
    if (currentMillis - previousMillis >= time / 2) {
      digitalWrite(11, LOW); // Traffic yellow light off
      previousMillis = currentMillis;
      if (set == 0) {
        _currentState = red; // Transition to red
      } else {
        _currentState = green; // Transition to green
      }
    }
  break;

  case red:
    digitalWrite(12, HIGH); // Traffic red light on
    if (currentMillis - previousMillis >= time) {
      digitalWrite(12, LOW); // Traffic red light off
      digitalWrite(11, HIGH); // Traffic yellow light on
      if (currentMillis - pedestrianMillis >= time / 2) {
        digitalWrite(11, LOW); // Traffic yellow light off
        pedestrianMillis = currentMillis;
        _targetState = pedestrianRed;
      }
    }
  break;

  case pedestrianRed:
    digitalWrite(12, LOW); // Traffic red light off
    digitalWrite(3, HIGH); // Pedestrian red light on
    if (currentMillis - pedestrianMillis >= time * 2) {
      digitalWrite(3, LOW); // Pedestrian red light off
      redState = 0; // Reset redState after pedestrian phase
      _targetState = _currentState; // Return to the traffic state
    }
  break;
}

void buttonPressed() {
  // When the button is pressed, set the pedestrian phase
  if (digitalRead(5) == HIGH) { // Button pressed (active HIGH)
    redState = 0; // Trigger pedestrian phase
    _targetState = pedestrianGreen; // Set to pedestrian green state
    previousMillis = millis(); // Reset the timing
  }
}

void receiveEvent(int event) {
  // Handle Wire communication event and set redState
  redState = Wire.read();
}

```

## GIT USAGE

main ▾

1 Branch ▾

0 Tags

Go to file

t

Add file ▾

Code ▾



SyedMuhammadAbisRizvi Add files via upload

55c74bb · 5 days ago

22 Commits

Task 2

Add files via upload

2 months ago

Task 3

Add files via upload

last month

Task 5

Add files via upload

2 weeks ago

Task1

made some changes

last month

README.md

Initial commit

2 months ago

Traffic light system test.ms14

Add files via upload

2 months ago

Traffic light.pptx

Add files via upload

5 days ago

### Commits over time

Weekly from Sep 28, 2025 to Dec 7, 2025

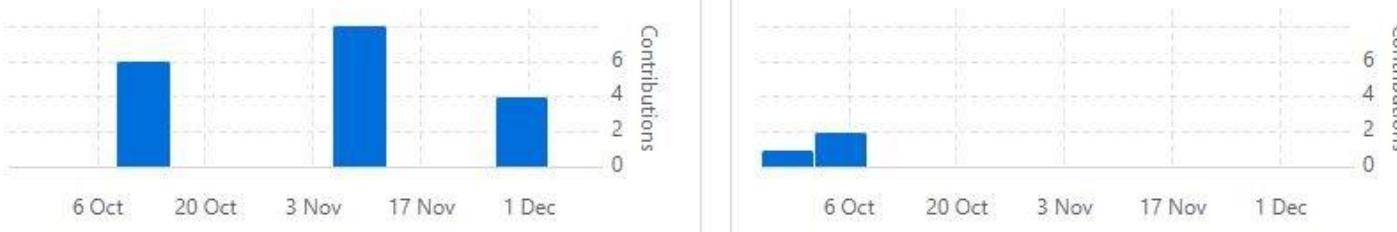


SyedMuhammadAbisRizvi

18 commits 736 ++ 181 --

#1

...



ibrahimabdelmalek31

3 commits 180 ++ 0 --

#2

...



THANK YOU.