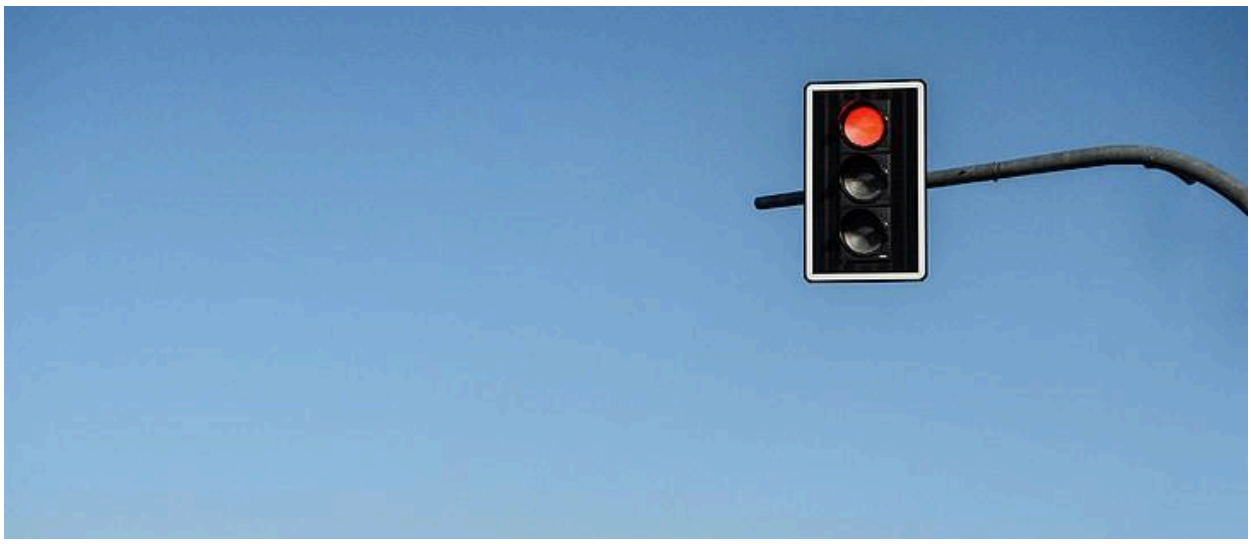
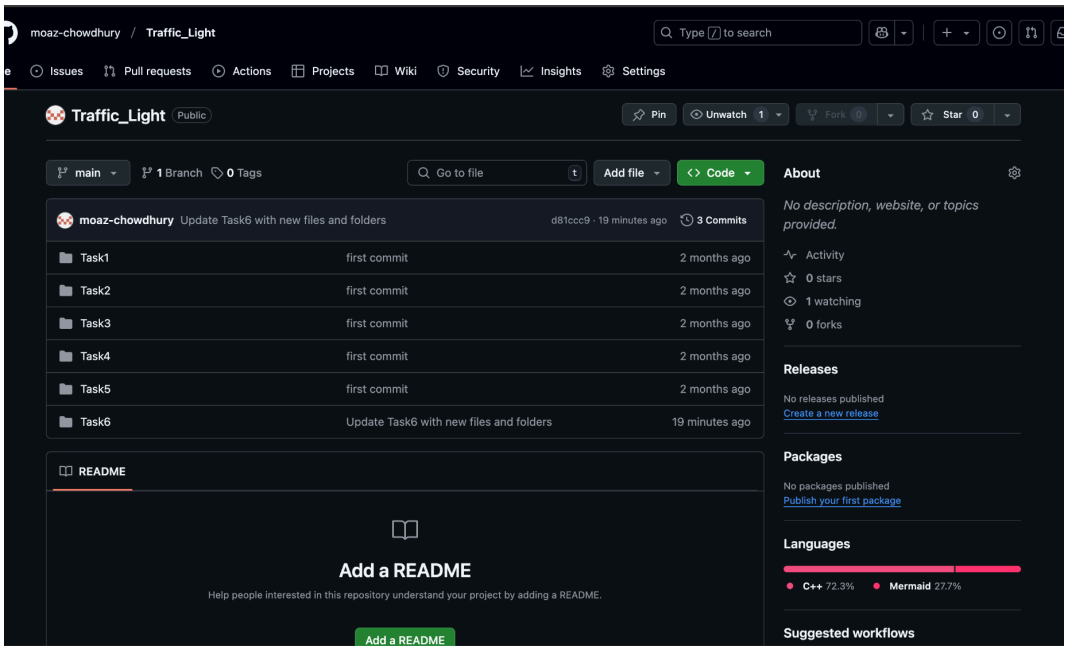
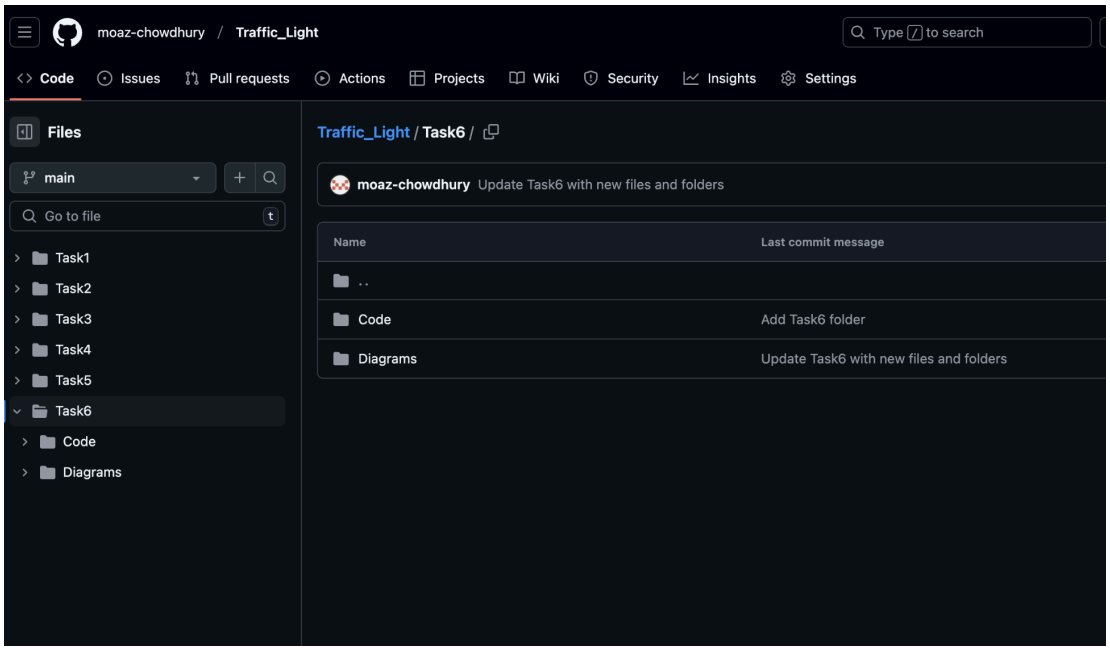


Documentation of Task 6

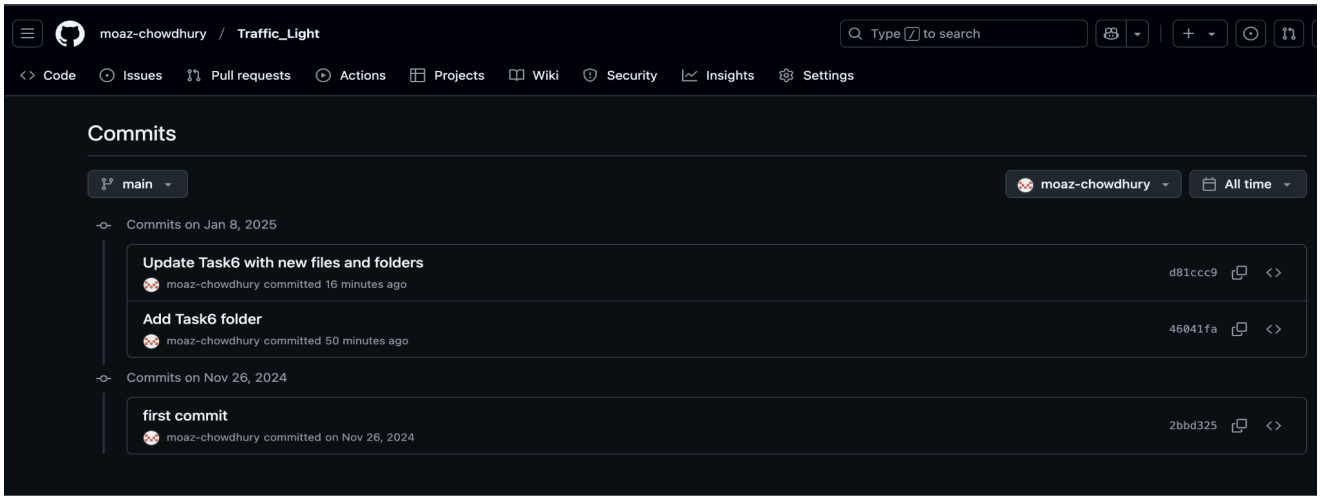
Moaz Bin Alamgir Chowdhury

GitHub and Git Usages:



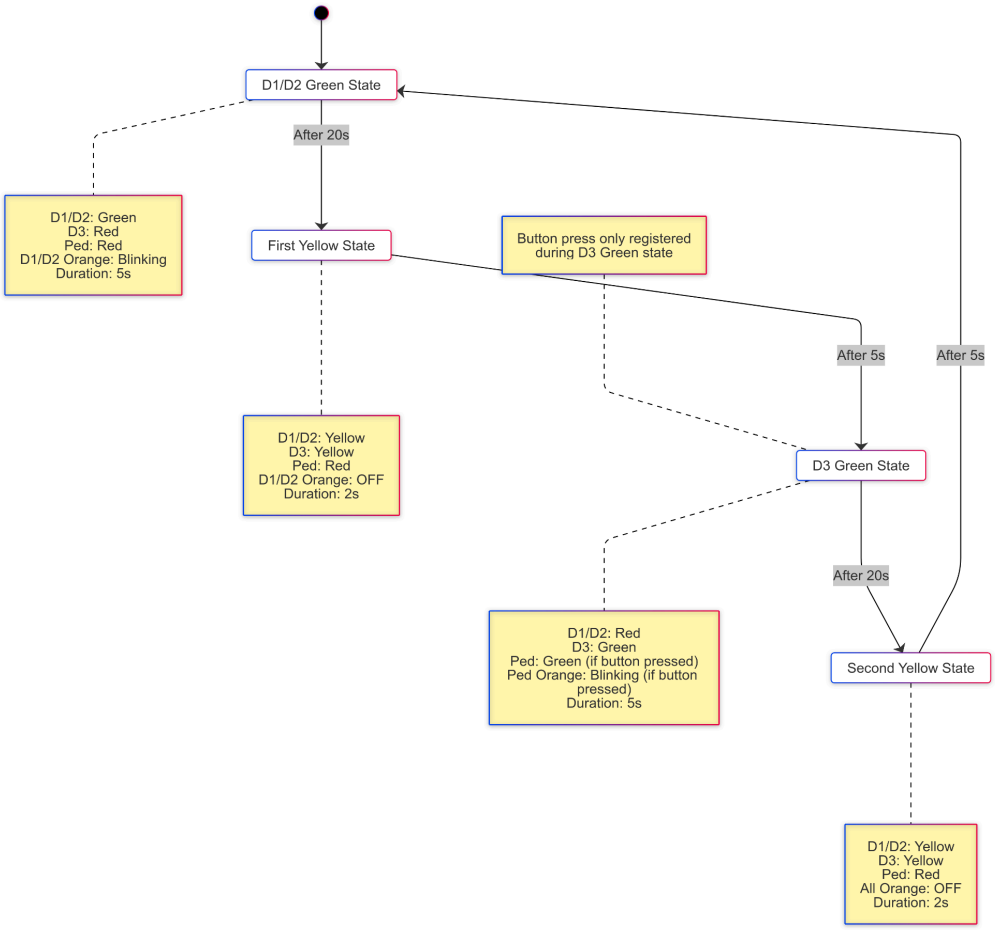


Commits:

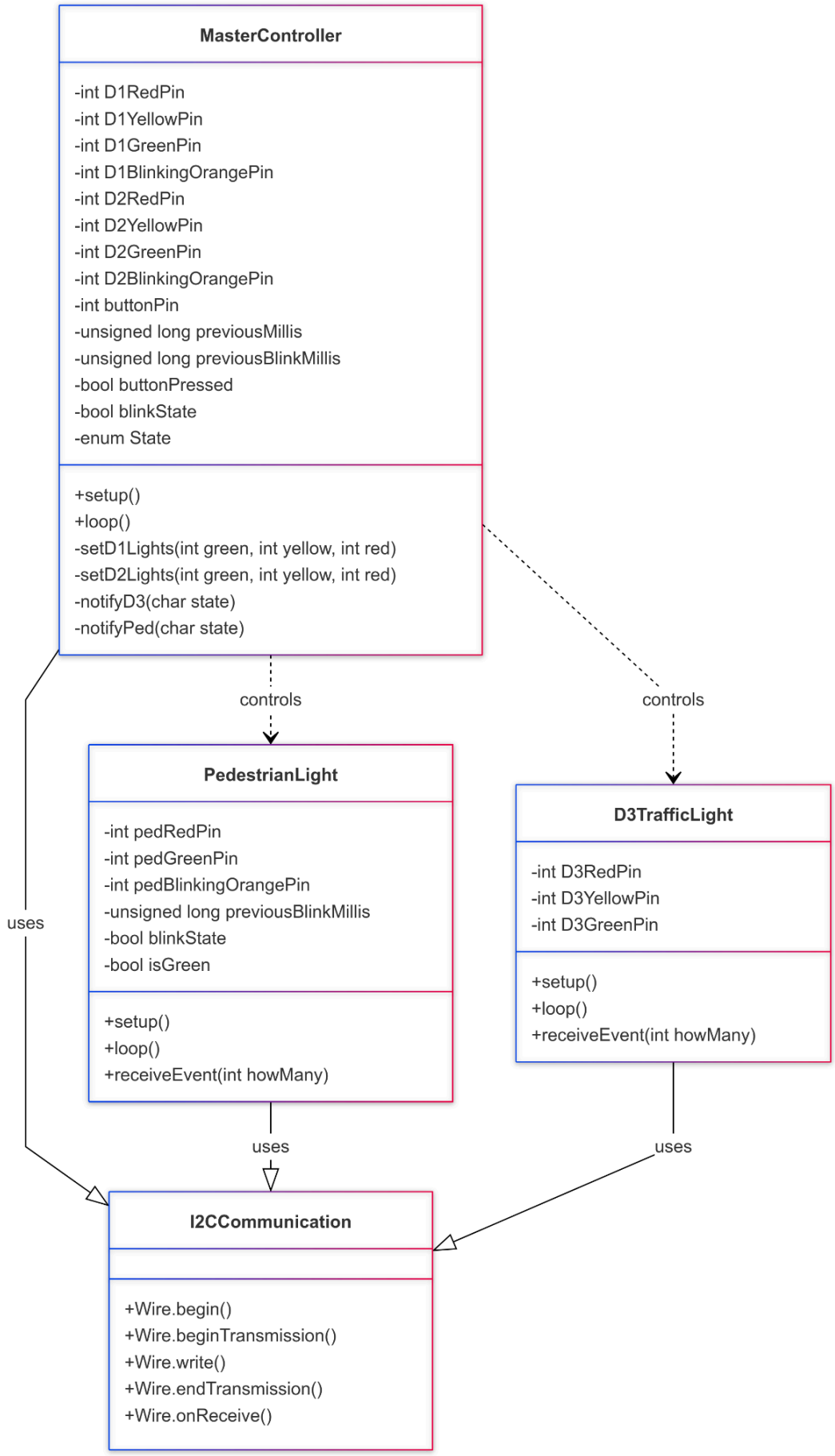

moaz-chowdhury / Traffic_Light
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Commits
main
moaz-chowdhury
All time
Commits on Jan 8, 2025
Update Task6 with new files and folders
moaz-chowdhury committed 16 minutes ago
d81ccc9
Add Task6 folder
moaz-chowdhury committed 50 minutes ago
46041fa
Commits on Nov 26, 2024
first commit
moaz-chowdhury committed on Nov 26, 2024
2bbd325

Diagrams:

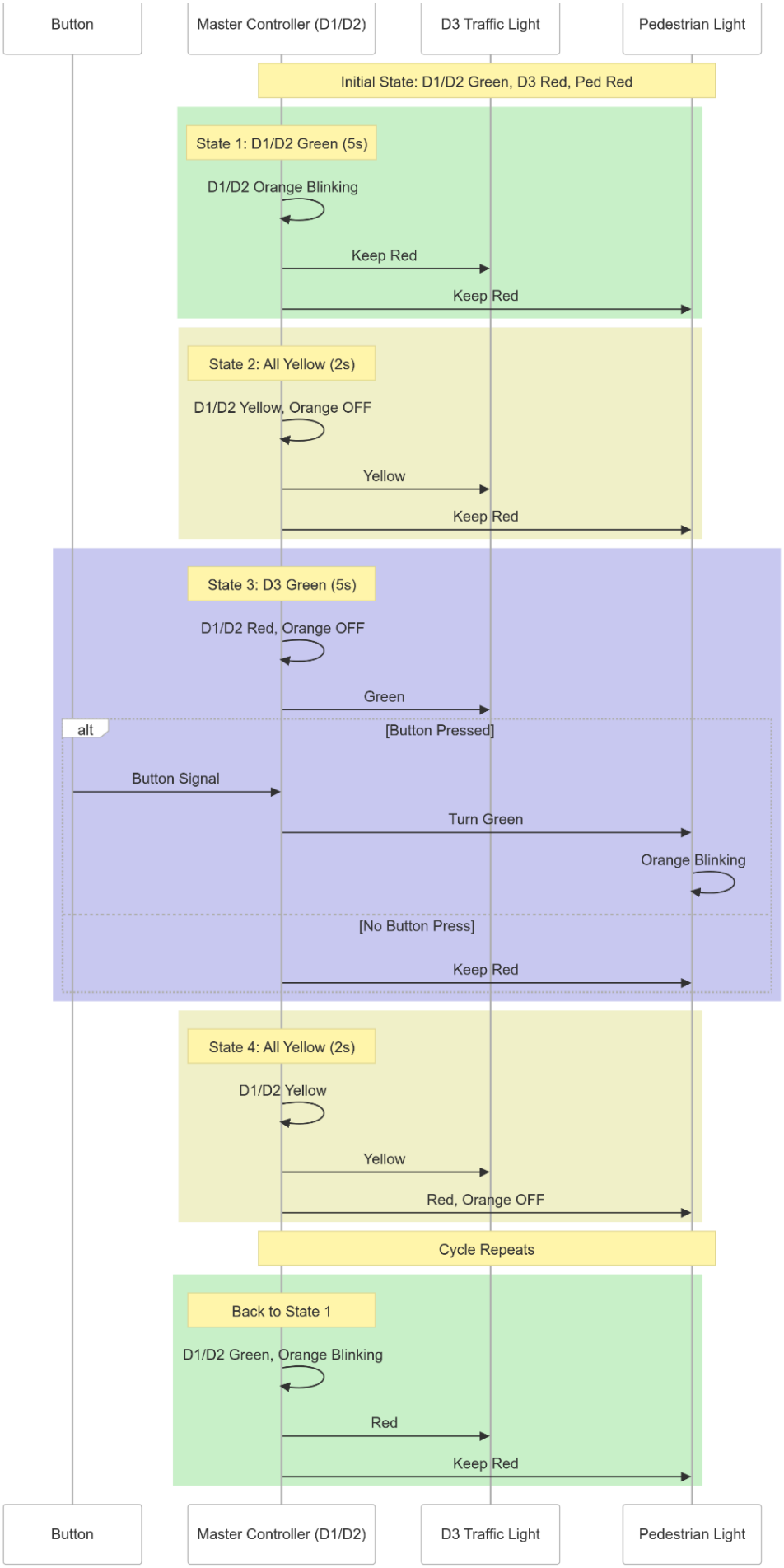
State Machine:



Class Diagram:


MasterController
-int D1RedPin
-int D1YellowPin
-int D1GreenPin
-int D1BlinkingOrangePin
-int D2RedPin
-int D2YellowPin
-int D2GreenPin
-int D2BlinkingOrangePin
-int buttonPin
-unsigned long previousMillis
-unsigned long previousBlinkMillis
-bool buttonPressed
-bool blinkState
-enum State
+setup()
+loop()
-setD1Lights(int green, int yellow, int red)
-setD2Lights(int green, int yellow, int red)
-notifyD3(char state)
-notifyPed(char state)
controls
controls
uses
PedestrianLight
-int pedRedPin
-int pedGreenPin
-int pedBlinkingOrangePin
-unsigned long previousBlinkMillis
-bool blinkState
-bool isGreen
+setup()
+loop()
+receiveEvent(int howMany)
D3TrafficLight
-int D3RedPin
-int D3YellowPin
-int D3GreenPin
+setup()
+loop()
+receiveEvent(int howMany)
uses
uses
I2CCommunication
+Wire.begin()
+Wire.beginTransmission()
+Wire.write()
+Wire.endTransmission()
+Wire.onReceive()

Sequence Diagram:


Button
Master Controller (D1/D2)
D3 Traffic Light
Pedestrian Light
Initial State: D1/D2 Green, D3 Red, Ped Red
State 1: D1/D2 Green (5s)
D1/D2 Orange Blinking
Keep Red
Keep Red
State 2: All Yellow (2s)
D1/D2 Yellow, Orange OFF
Yellow
Keep Red
State 3: D3 Green (5s)
D1/D2 Red, Orange OFF
Green
alt
[Button Pressed]
Button Signal
Turn Green
Orange Blinking
[No Button Press]
Keep Red
State 4: All Yellow (2s)
D1/D2 Yellow
Yellow
Red, Orange OFF
Cycle Repeats
Back to State 1
D1/D2 Green, Orange Blinking
Red
Keep Red
Button
Master Controller (D1/D2)
D3 Traffic Light
Pedestrian Light

Codes:

Master Arduino:

```
#include <Wire.h>

// Traffic light pins for D1
const int D1RedPin = 9;
const int D1YellowPin = 10;
const int D1GreenPin = 11;
const int D1BlinkingOrangePin = 5;  // New pin for blinking orange
const int buttonPin = 2;

// Timing durations
const unsigned long greenDuration = 5000;  // 5 seconds
const unsigned long yellowDuration = 2000; // 2 seconds
const unsigned long redDuration = 5000;    // 5 seconds
const unsigned long blinkDuration = 500;   // 0.5 seconds for blinking

unsigned long previousMillis = 0;
unsigned long previousBlinkMillis = 0;
bool buttonPressed = false;
bool blinkState = false;

enum State {
  D1_GREEN,   // State 1: D1 Green, D3 Red
  D1_YELLOW,  // State 2: D1 Yellow, D3 Yellow
  D1_RED,     // State 3: D1 Red, D3 Green
  D3_YELLOW,  // State 4: D1 Yellow, D3 Yellow
  D3_RED      // State 5: D1 Green, D3 Red
};

State currentState = D1_GREEN;

void setup() {
  Wire.begin();
  pinMode(D1RedPin, OUTPUT);
  pinMode(D1YellowPin, OUTPUT);
  pinMode(D1GreenPin, OUTPUT);
  pinMode(D1BlinkingOrangePin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  Serial.begin(9600);
```

```
    // Initial state: D1 Green, D3 Red
    setD1Lights(HIGH, LOW, LOW); // Green
    notifyD3('R'); // Red
    notifyPed('R'); // Red
}

void loop() {
    unsigned long currentMillis = millis();

    // Handle blinking orange light
    if (currentState == D1_GREEN) {
        if (currentMillis - previousBlinkMillis >= blinkDuration) {
            blinkState = !blinkState;
            digitalWrite(D1BlinkingOrangePin, blinkState);
            previousBlinkMillis = currentMillis;
        }
    } else {
        digitalWrite(D1BlinkingOrangePin, LOW);
    }

    // Check button press
    if (digitalRead(buttonPin) == LOW && !buttonPressed) {
        buttonPressed = true;
    }

    // State machine
    switch (currentState) {
        case D1_GREEN:  // State 1
            if (currentMillis - previousMillis >= greenDuration) {
                setD1Lights(LOW, HIGH, LOW); // Yellow
                notifyD3('Y'); // Yellow
                notifyPed('R'); // Red
                previousMillis = currentMillis;
                currentState = D1_YELLOW;
            }
            break;

        case D1_YELLOW:  // State 2
            if (currentMillis - previousMillis >= yellowDuration) {
                setD1Lights(LOW, LOW, HIGH); // Red
                notifyD3('G'); // Green
```

```
            if (buttonPressed) {
                notifyPed('G'); // Green if button was pressed
            }
            previousMillis = currentMillis;
            currentState = D1_RED;
        }
        break;

    case D1_RED:  // State 3
        if (currentMillis - previousMillis >= redDuration) {
            setD1Lights(LOW, HIGH, LOW); // Yellow
            notifyD3('Y'); // Yellow
            notifyPed('R'); // Red
            previousMillis = currentMillis;
            currentState = D3_YELLOW;
        }
        break;

    case D3_YELLOW:  // State 4
        if (currentMillis - previousMillis >= yellowDuration) {
            setD1Lights(HIGH, LOW, LOW); // Green
            notifyD3('R'); // Red
            notifyPed('R'); // Red
            buttonPressed = false; // Reset button press
            previousMillis = currentMillis;
            currentState = D1_GREEN;
        }
        break;
  }
}

void setD1Lights(int green, int yellow, int red) {
  digitalWrite(D1GreenPin, green);
  digitalWrite(D1YellowPin, yellow);
  digitalWrite(D1RedPin, red);
}

void notifyD3(char state) {
  Wire.beginTransmission(10);
  Wire.write(state);
  Wire.endTransmission();
}
```

```
void notifyPed(char state) {
  Wire.beginTransmission(9);
  Wire.write(state);
  Wire.endTransmission();
}
```

Slave Arduino 1:

```
#include <Wire.h>

const int pedRedPin = 12;
const int pedGreenPin = 13;
const int pedBlinkingOrangePin = 4;  // New pin for blinking orange

unsigned long previousBlinkMillis = 0;
const unsigned long blinkDuration = 500;  // 0.5 seconds for blinking
bool blinkState = false;
bool isGreen = false;

void setup() {
  Wire.begin(9);  // Address 9 for pedestrian light
  Wire.onReceive(receiveEvent);
  pinMode(pedRedPin, OUTPUT);
  pinMode(pedGreenPin, OUTPUT);
  pinMode(pedBlinkingOrangePin, OUTPUT);

  // Initial state: Red
  digitalWrite(pedRedPin, HIGH);
  digitalWrite(pedGreenPin, LOW);
  digitalWrite(pedBlinkingOrangePin, LOW);
  Serial.begin(9600);
}

void loop() {
  unsigned long currentMillis = millis();
```

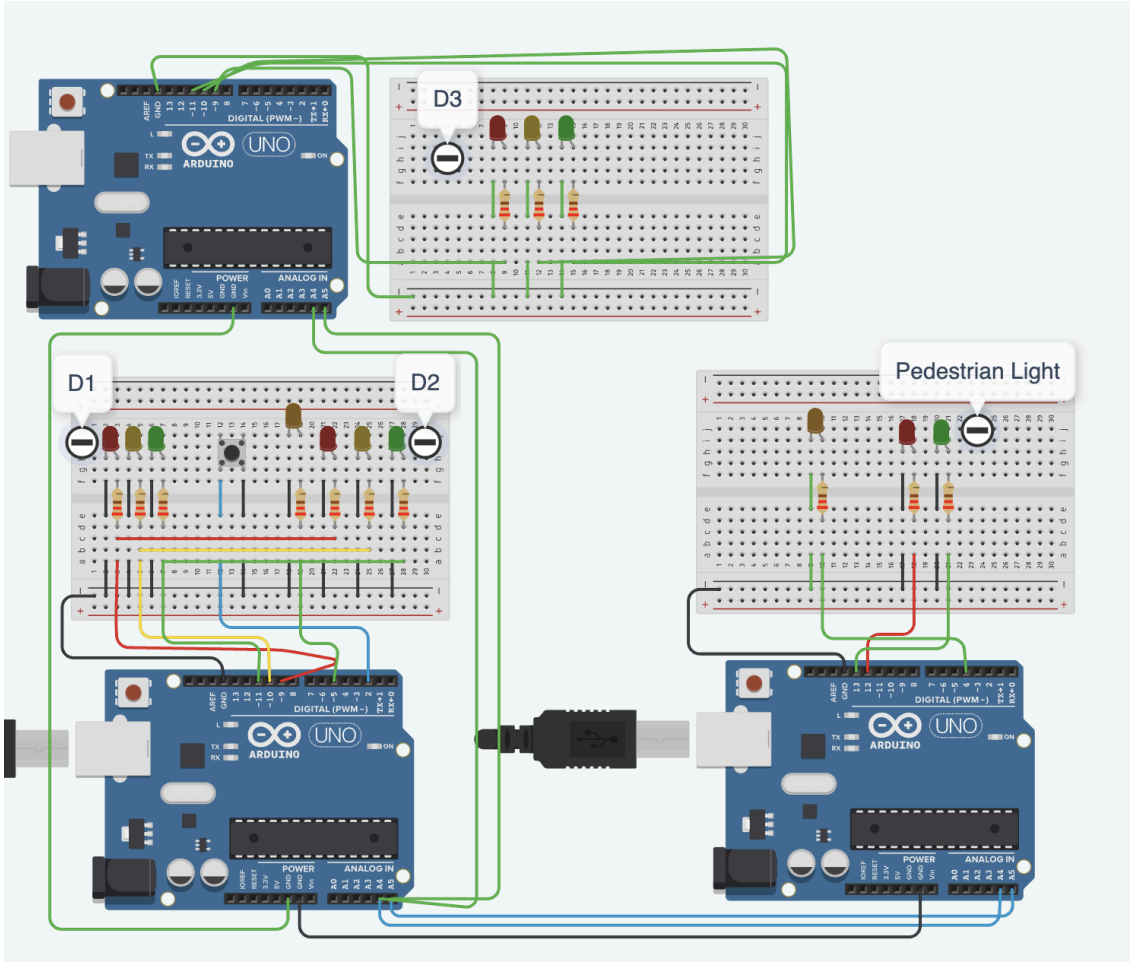
```
    // Handle blinking orange light
    if (isGreen) {
        if (currentMillis - previousBlinkMillis >= blinkDuration) {
            blinkState = !blinkState;
            digitalWrite(pedBlinkingOrangePin, blinkState);
            previousBlinkMillis = currentMillis;
        }
    } else {
        digitalWrite(pedBlinkingOrangePin, LOW);
    }
}

void receiveEvent(int howMany) {
    char command = Wire.read();

    switch(command) {
        case 'G':  // Green
            digitalWrite(pedRedPin, LOW);
            digitalWrite(pedGreenPin, HIGH);
            isGreen = true;
            Serial.println("Pedestrian: Green");
            break;

        case 'R':  // Red
            digitalWrite(pedRedPin, HIGH);
            digitalWrite(pedGreenPin, LOW);
            isGreen = false;
            Serial.println("Pedestrian: Red");
            break;
    }
}
```

Slave Arduino 2:

```
#include <Wire.h>

const int D3RedPin = 9;
const int D3YellowPin = 10;
const int D3GreenPin = 11;

void setup() {
  Wire.begin(10);  // Address 10 for D3 traffic light
  Wire.onReceive(receiveEvent);
  pinMode(D3RedPin, OUTPUT);
  pinMode(D3YellowPin, OUTPUT);
  pinMode(D3GreenPin, OUTPUT);

  // Initial state: Red
  digitalWrite(D3RedPin, HIGH);
  digitalWrite(D3YellowPin, LOW);
  digitalWrite(D3GreenPin, LOW);
  Serial.begin(9600);
}

void loop() {
  // State controlled by I2C commands
}

void receiveEvent(int howMany) {
  char command = Wire.read();

  switch(command) {
    case 'R':  // Red
      digitalWrite(D3RedPin, HIGH);
      digitalWrite(D3YellowPin, LOW);
      digitalWrite(D3GreenPin, LOW);
```

```
        Serial.println("D3: Red");
        break;

      case 'Y':  // Yellow
        digitalWrite(D3RedPin, LOW);
        digitalWrite(D3YellowPin, HIGH);
        digitalWrite(D3GreenPin, LOW);
        Serial.println("D3: Yellow");
        break;

      case 'G':  // Green
        digitalWrite(D3RedPin, LOW);
        digitalWrite(D3YellowPin, LOW);
        digitalWrite(D3GreenPin, HIGH);
        Serial.println("D3: Green");
        break;
    }
}
```

Simulation:



The Blinking Orange Light in D2 reminds the drivers of the D2 lane to wait for the vehicles in the D1 lane to pass through before they can go onto the left road which is D3. The Blinking Orange Light in the Pedestrian Crossing reminds the drivers of the D3 to wait until all the pedestrians have crossed the road.