Arduino Lab Progressive Documentation

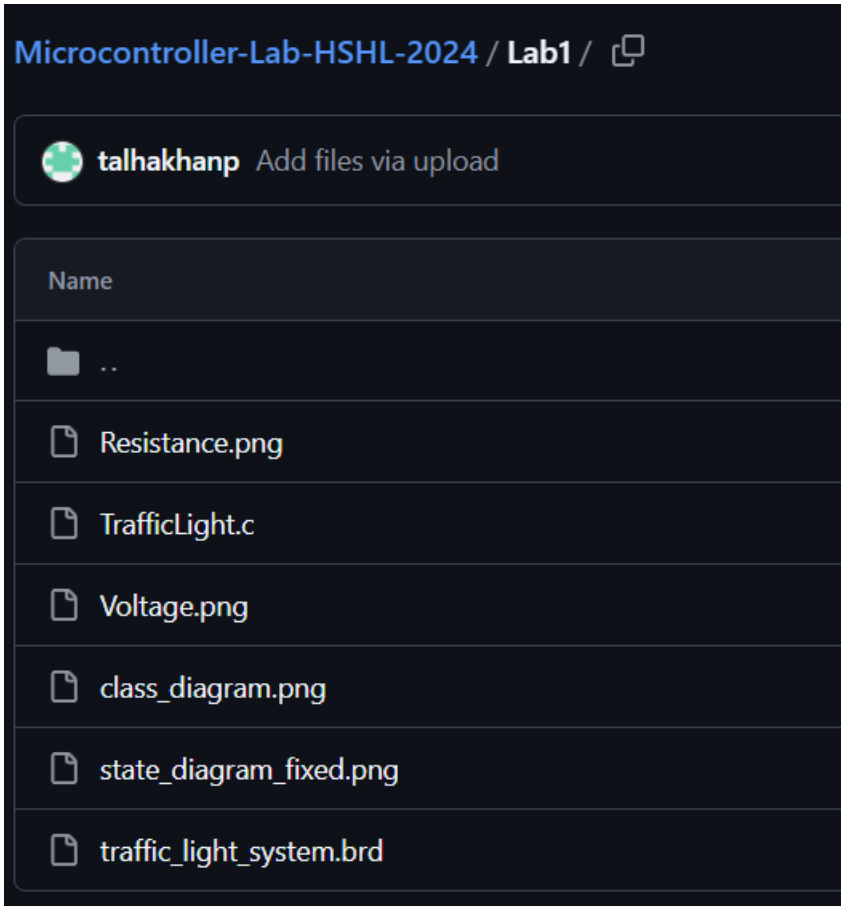
Lab 1: Basic Traffic Signal

In this task, a basic traffic signal was implemented using three LEDs: red, yellow, and green. The system cycles through the standard traffic light phases.

Code:

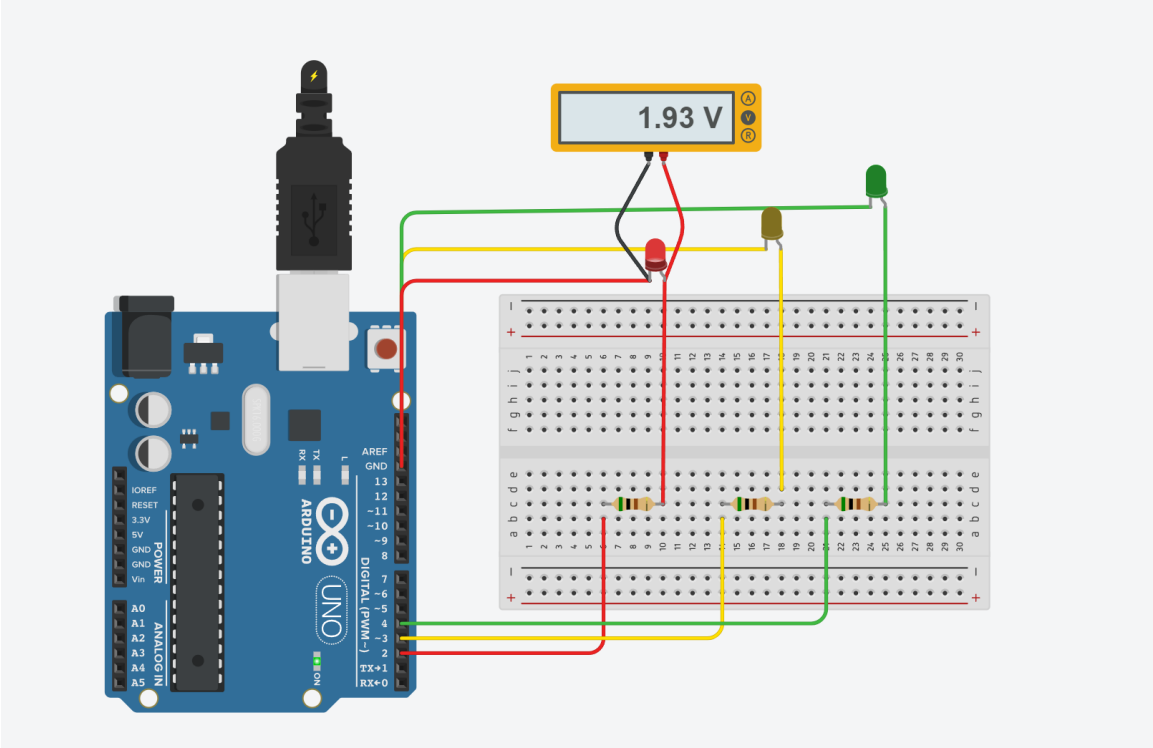
```
/// code for the tinkercad simulation

// BY TALHA KHAN ONLY FOR EDUCATIONAL PURPOSES

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  digitalWrite(3, LOW);
  digitalWrite(2, HIGH);  // red only
  delay(5000);

  digitalWrite(3, HIGH); // // red + yellow
  delay(2000);

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
   digitalWrite(4, HIGH); // green
  delay(5000);

  digitalWrite(4, LOW);
  digitalWrite(3, HIGH); // // yellow
  delay(2000);

}
```

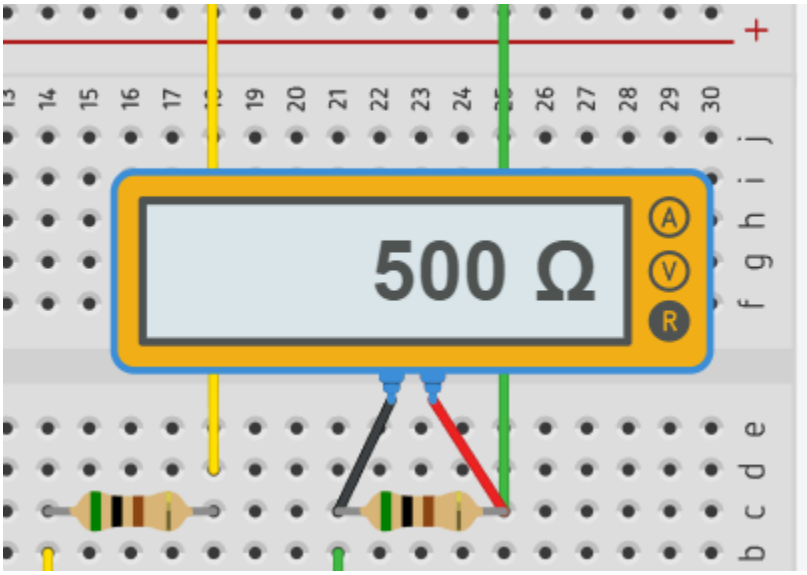
GitHub

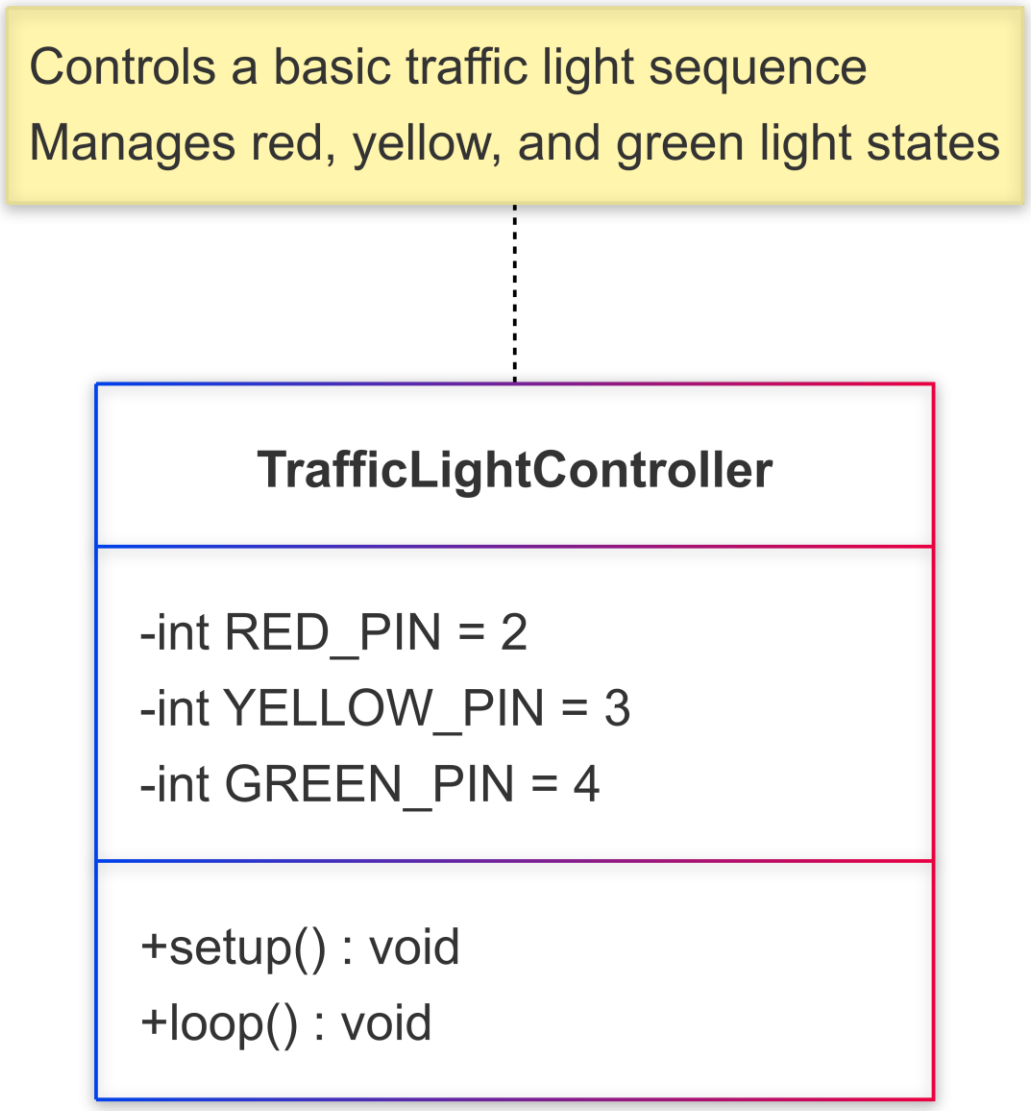

Microcontroller-Lab-HSHL-2024 / Lab1 /
talhakhanp Add files via upload
Name
..
Resistance.png
TrafficLight.c
Voltage.png
class_diagram.png
state_diagram_fixed.png
traffic_light_system.brd

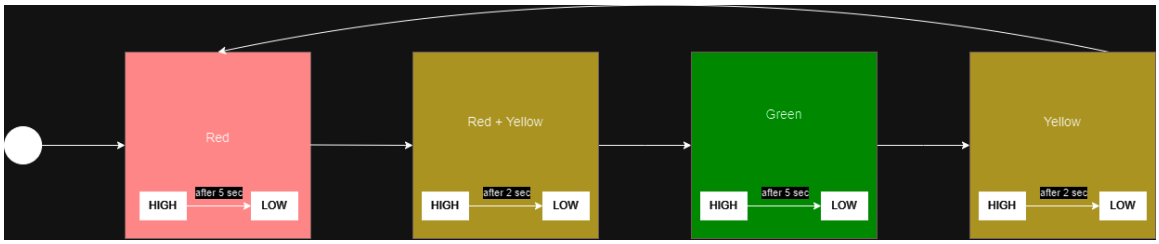

Rename code-for-tinkercad to Lab1/TrafficLight.c, Moving file to Lab1 folder
talhakhanp authored 2 days ago
Verified
3482bbe

Diagrams


1.93 V
ARDUINO
UNO
POWER
ANALOG IN
DIGITAL (PWM~)


500 Ω
A
V
R





Lab 2: Traffic and Pedestrian Light Integration

In this task, a pedestrian light controlled by an RGB LED and a button was added to the system. When the traffic light was red and the button was pressed, the pedestrian light turned green, allowing pedestrians to cross.

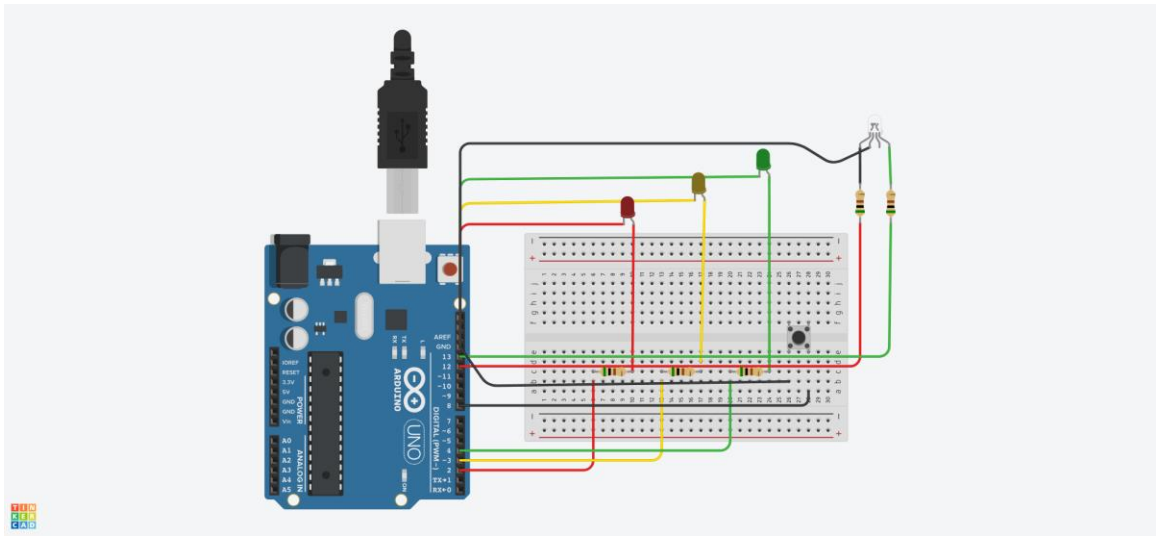
Code:

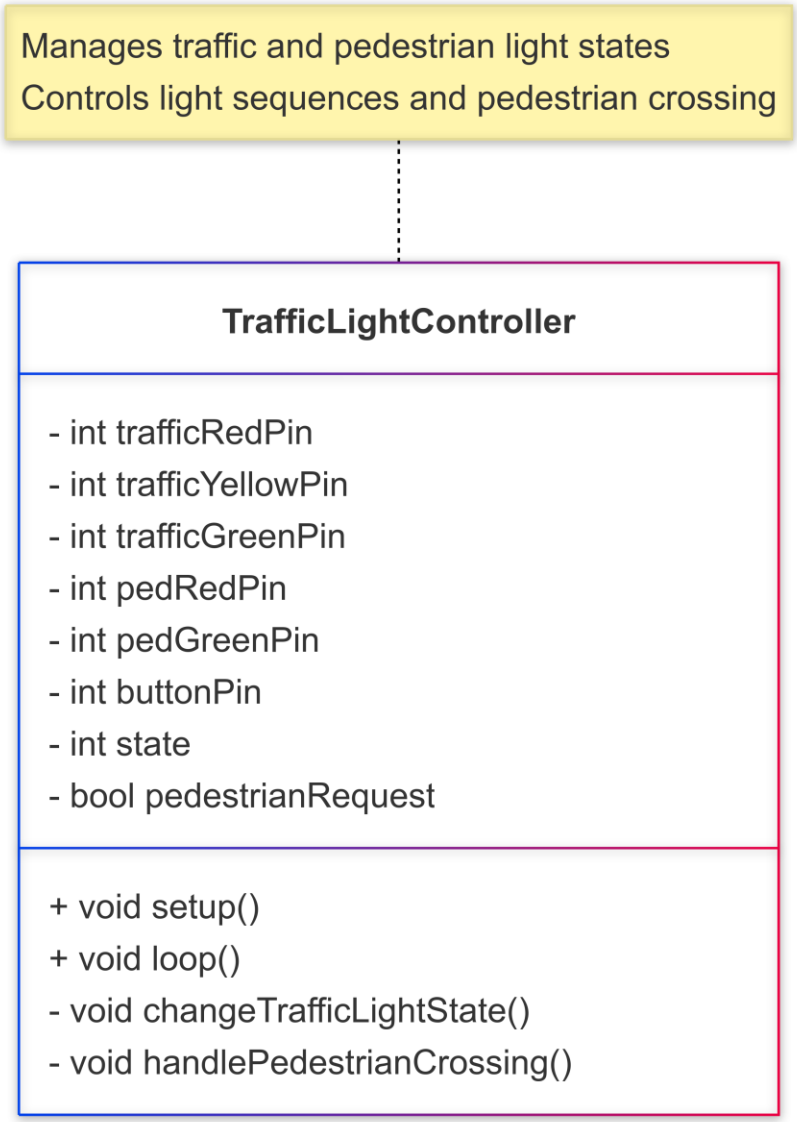
```
// Pin Definitions
int trafficRedPin = 2;
int trafficYellowPin = 3;
int trafficGreenPin = 4;
int pedRedPin = 12;
int pedGreenPin = 13;
int buttonPin = 8;

// Traffic Light States
int state = 0;  // // 0 = red, 1 = red + yellow, 2 = green, 3 = yellow
bool pedestrianRequest = false;

void setup() {
  pinMode(trafficRedPin, OUTPUT);
  pinMode(trafficYellowPin, OUTPUT);
  pinMode(trafficGreenPin, OUTPUT);
  pinMode(pedRedPin, OUTPUT);
  pinMode(pedGreenPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);

  // start with the traffic light + ped light both on red
  digitalWrite(trafficRedPin, HIGH);
  digitalWrite(pedRedPin, HIGH);
}

void loop() {
  // check if the button is pressed for pedestrian light
  if (digitalRead(buttonPin) == LOW) {
    pedestrianRequest = true;  // someone pressed the button
  }

  switch (state) {
    case 0:  // red light phase
      digitalWrite(trafficRedPin, HIGH);
      digitalWrite(trafficYellowPin, LOW);
      digitalWrite(trafficGreenPin, LOW);
      digitalWrite(pedRedPin, HIGH);
      digitalWrite(pedGreenPin, LOW);

      delay(10000);  // wait for 10 seconds on red
      if (pedestrianRequest) {
        // pedestrian walk phase
```

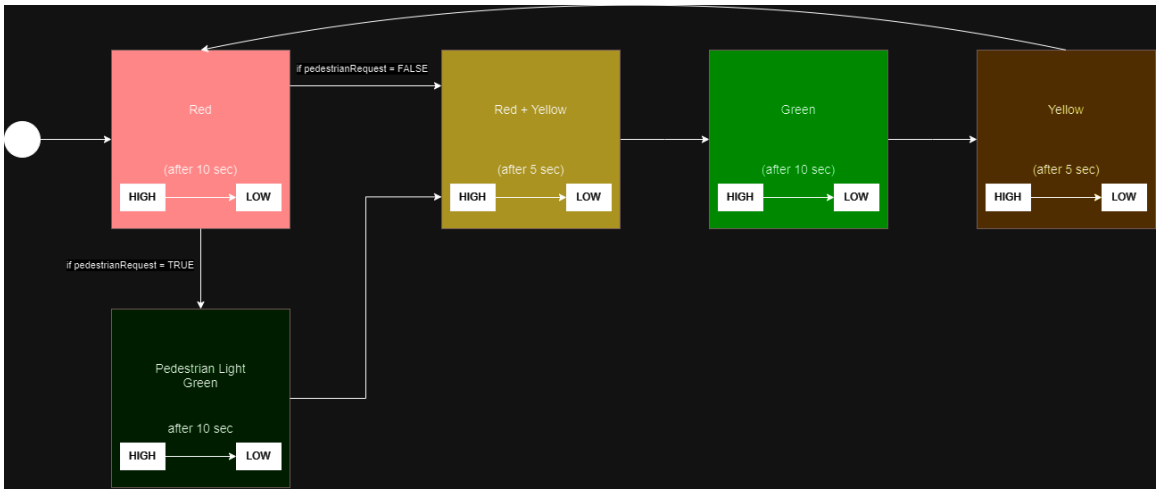
```
      digitalWrite(pedRedPin, LOW);
      digitalWrite(pedGreenPin, HIGH);
      delay(10000);  // let pedestrians walk for 10 seconds
      pedestrianRequest = false;  // reset the button press
    }
    state = 1;  // go to the next phase
    break;

  case 1:  // red + yellow light phase
    digitalWrite(trafficRedPin, HIGH);
    digitalWrite(trafficYellowPin, HIGH);
    digitalWrite(trafficGreenPin, LOW);
    delay(5000);  // stay here for 5 seconds
    state = 2;  // move to green
    break;

  case 2:  // green light phase
    digitalWrite(trafficRedPin, LOW);
    digitalWrite(trafficYellowPin, LOW);
    digitalWrite(trafficGreenPin, HIGH);
    digitalWrite(pedRedPin, HIGH);
    digitalWrite(pedGreenPin, LOW);

    delay(10000);  // green light stays for 10 seconds
    state = 3;  // move to yellow light
    break;

  case 3:  // yellow light phase
    digitalWrite(trafficRedPin, LOW);
    digitalWrite(trafficYellowPin, HIGH);
    digitalWrite(trafficGreenPin, LOW);
    delay(5000);  // yellow light stays for 5 seconds
    state = 0;  // go back to red light
    break;
 }
}
```

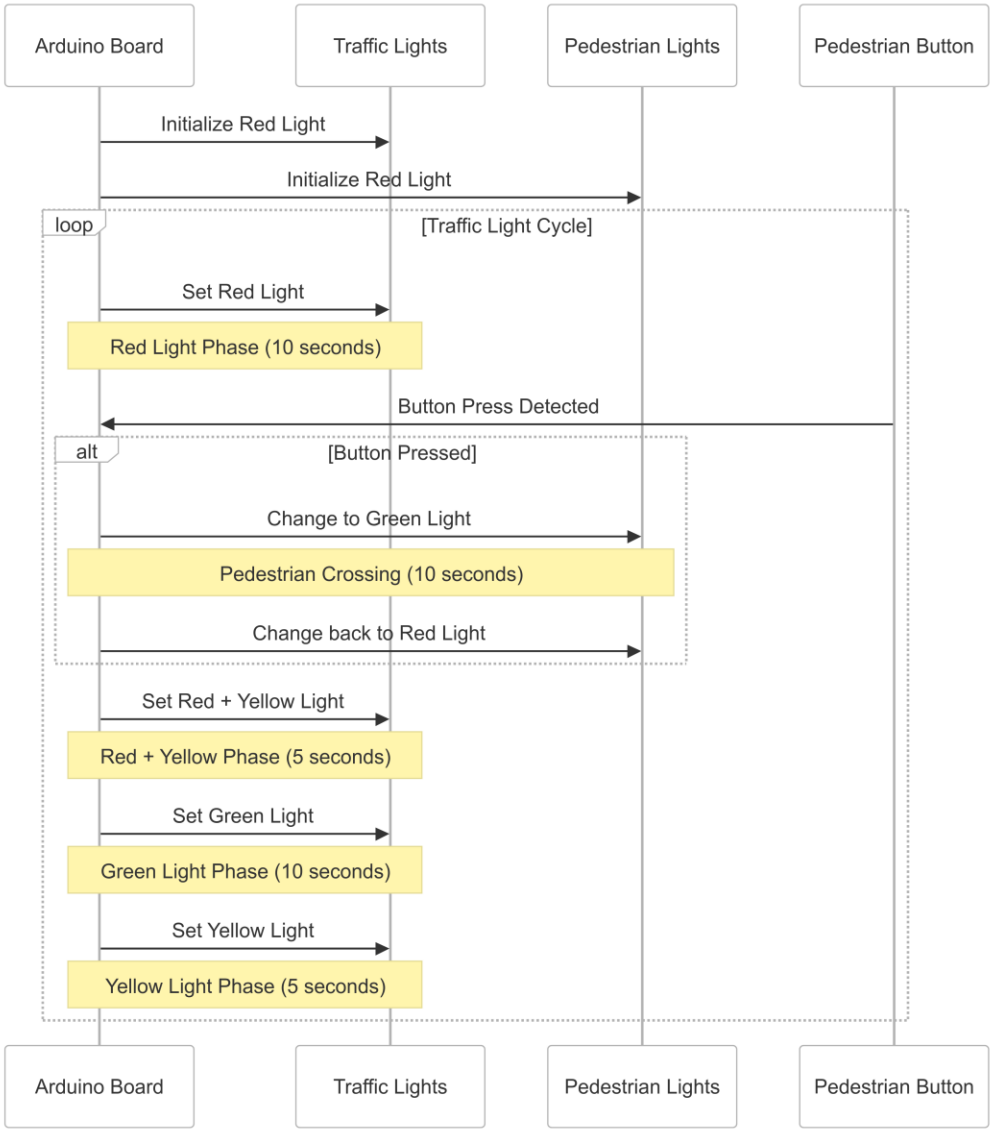
GitHub

Diagrams:



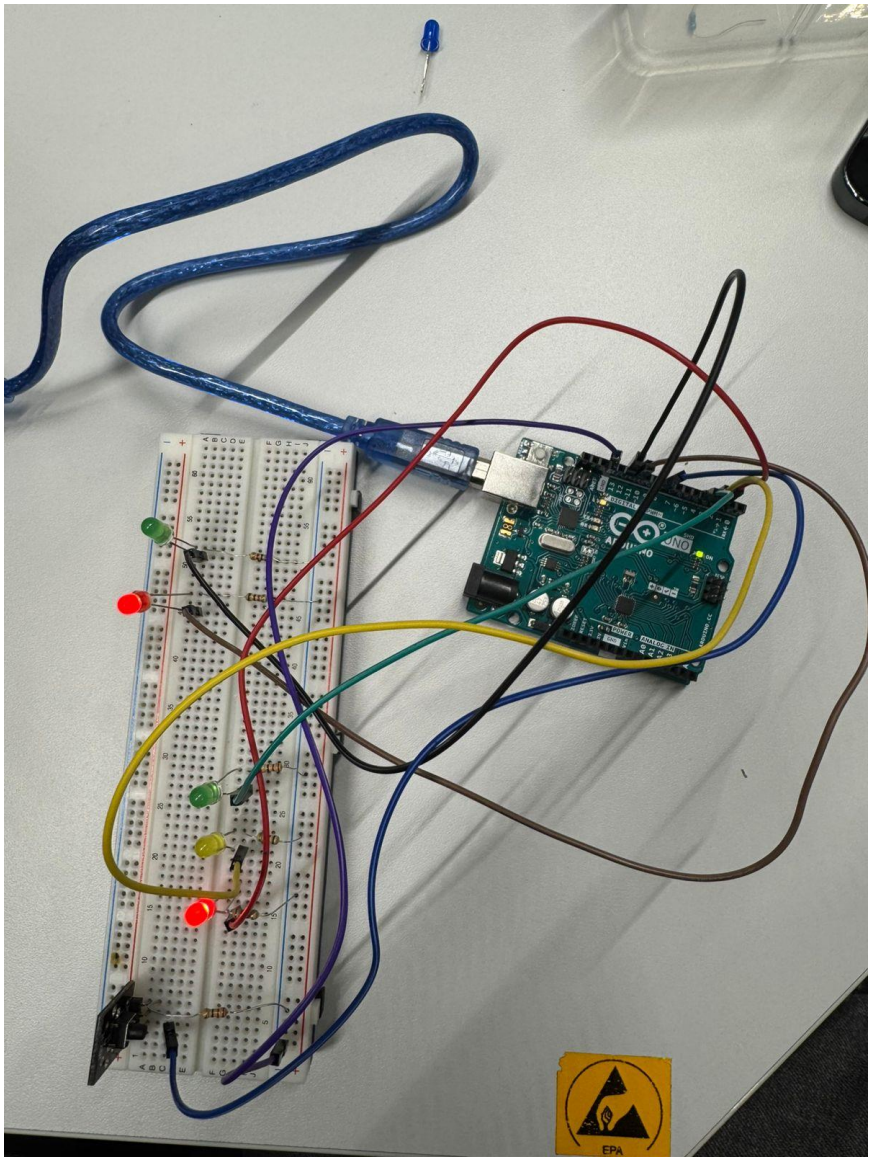

Manages traffic and pedestrian light states
Controls light sequences and pedestrian crossing
TrafficLightController
- int trafficRedPin
- int trafficYellowPin
- int trafficGreenPin
- int pedRedPin
- int pedGreenPin
- int buttonPin
- int state
- bool pedestrianRequest
+ void setup()
+ void loop()
- void changeTrafficLightState()
- void handlePedestrianCrossing()


Red
(after 10 sec)
HIGH
LOW
if pedestrianRequest = FALSE
Red + Yellow
(after 5 sec)
HIGH
LOW
Green
(after 10 sec)
HIGH
LOW
Yellow
(after 5 sec)
HIGH
LOW
if pedestrianRequest = TRUE
Pedestrian Light
Green
after 10 sec
HIGH
LOW


Arduino Board
Traffic Lights
Pedestrian Lights
Pedestrian Button
Initialize Red Light
Initialize Red Light
loop
[Traffic Light Cycle]
Set Red Light
Red Light Phase (10 seconds)
Button Press Detected
alt
[Button Pressed]
Change to Green Light
Pedestrian Crossing (10 seconds)
Change back to Red Light
Set Red + Yellow Light
Red + Yellow Phase (5 seconds)
Set Green Light
Green Light Phase (10 seconds)
Set Yellow Light
Yellow Light Phase (5 seconds)
Arduino Board
Traffic Lights
Pedestrian Lights
Pedestrian Button

Lab 4: Task 2 Realization

Task 2 was realized in the lab.



Lab 5: Peer-to-Peer Communication

In this task, two Arduino boards were connected using serial communication. The master board had a button, and the slave board controlled the traffic lights. When the button on the master was pressed, it sent a signal to the slave to activate the lights.

****Master Code****

```
// Pin Definitions
int buttonPin = 8;  // pin connected to the button

void setup() {
  Serial.begin(9600);              // initialize serial communication
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  // check if the button is pressed
  if (digitalRead(buttonPin) == LOW) {
    delay(50); // debounce delay
    if (digitalRead(buttonPin) == LOW) {  // check again to confirm press
      Serial.println(1);  // send signal to slave
      delay(1000);           // prevent multiple signals for one press
    }
  }
}
```

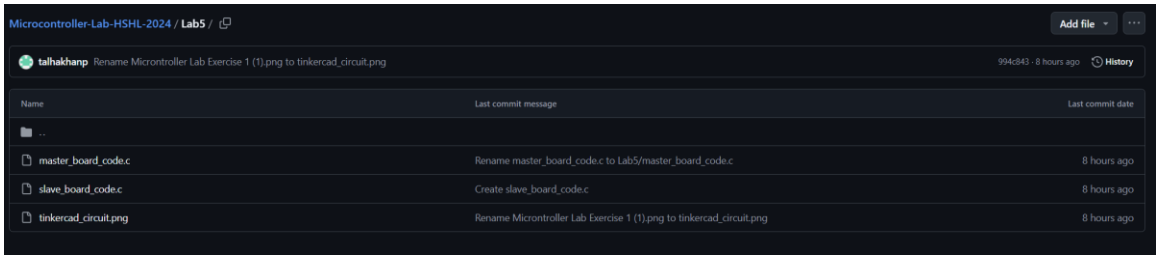
****Slave Code****

```
// Pin Definitions
int trafficRedPin = 2;
int trafficYellowPin = 3;
int trafficGreenPin = 4;
int pedRedPin = 12;
int pedGreenPin = 13;

// Traffic Light States
int state = 0;  // 0 = red, 1 = red + yellow, 2 = green, 3 = yellow
bool pedestrianRequest = false;

void setup() {
  pinMode(trafficRedPin, OUTPUT);
  pinMode(trafficYellowPin, OUTPUT);
  pinMode(trafficGreenPin, OUTPUT);
  pinMode(pedRedPin, OUTPUT);
  pinMode(pedGreenPin, OUTPUT);

  // starting serial communication
  Serial.begin(9600);

  // start with the traffic light + ped light both on red
```

```
  digitalWrite(trafficRedPin, HIGH);
  digitalWrite(pedRedPin, HIGH);
}

void loop() {
  // check for incoming data from the master
  if (Serial.available() > 0) {
    char incomingData = Serial.read();
    if (incomingData == '1') {
      pedestrianRequest = true;  // set the pedestrian request flag
    }
  }

  switch (state) {
    case 0:  // red light phase
      digitalWrite(trafficRedPin, HIGH);
      digitalWrite(trafficYellowPin, LOW);
      digitalWrite(trafficGreenPin, LOW);
      digitalWrite(pedRedPin, HIGH);
      digitalWrite(pedGreenPin, LOW);

      delay(10000);  // wait for 10 seconds on red
      if (pedestrianRequest) {
        // pedestrian walk phase
        digitalWrite(pedRedPin, LOW);
        digitalWrite(pedGreenPin, HIGH);
        delay(10000);  // let pedestrians walk for 10 seconds
        pedestrianRequest = false;  // reset the button press
      }
      state = 1;  // go to the next phase
      break;

    case 1:  // red + yellow light phase
      digitalWrite(trafficRedPin, HIGH);
      digitalWrite(trafficYellowPin, HIGH);
      digitalWrite(trafficGreenPin, LOW);
      delay(5000);  // stay here for 5 seconds
      state = 2;  // move to green
      break;

    case 2:  // green light phase
      digitalWrite(trafficRedPin, LOW);
      digitalWrite(trafficYellowPin, LOW);
```

```
    digitalWrite(trafficGreenPin, HIGH);
    digitalWrite(pedRedPin, HIGH);
    digitalWrite(pedGreenPin, LOW);

    delay(10000);  // green light stays for 10 seconds
    state = 3;  // move to yellow light
    break;

  case 3:  // yellow light phase
    digitalWrite(trafficRedPin, LOW);
    digitalWrite(trafficYellowPin, HIGH);
    digitalWrite(trafficGreenPin, LOW);
    delay(5000);  // yellow light stays for 5 seconds
    state = 0;  // go back to red light
    break;
 }
}
```

****GitHub****



****Diagrams****