A Traffic Light System for Safe Crossings Between President University Buildings A and B

Microcontroller - Project 1

Group member(s) :

- Dimas Budi Nugraha (001201700023)

Problem Statement

The crossing area between Building A and Building B at President University experiences a high volume of pedestrian and vehicular traffic. A significant safety concern has arisen due to the occurrence of students crossing the road without paying attention to the vehicles passing by, posing a risk to their safety. To address this issue, there is a need to implement a set of traffic lights, including Red, Yellow, and Green lights, as a safety measure.

The specific requirements for the traffic lights are as follows:

- The Red light should be illuminated for 15 seconds.
- The Yellow light should be illuminated for 5 seconds.
- The Green light should be illuminated for 8 seconds.

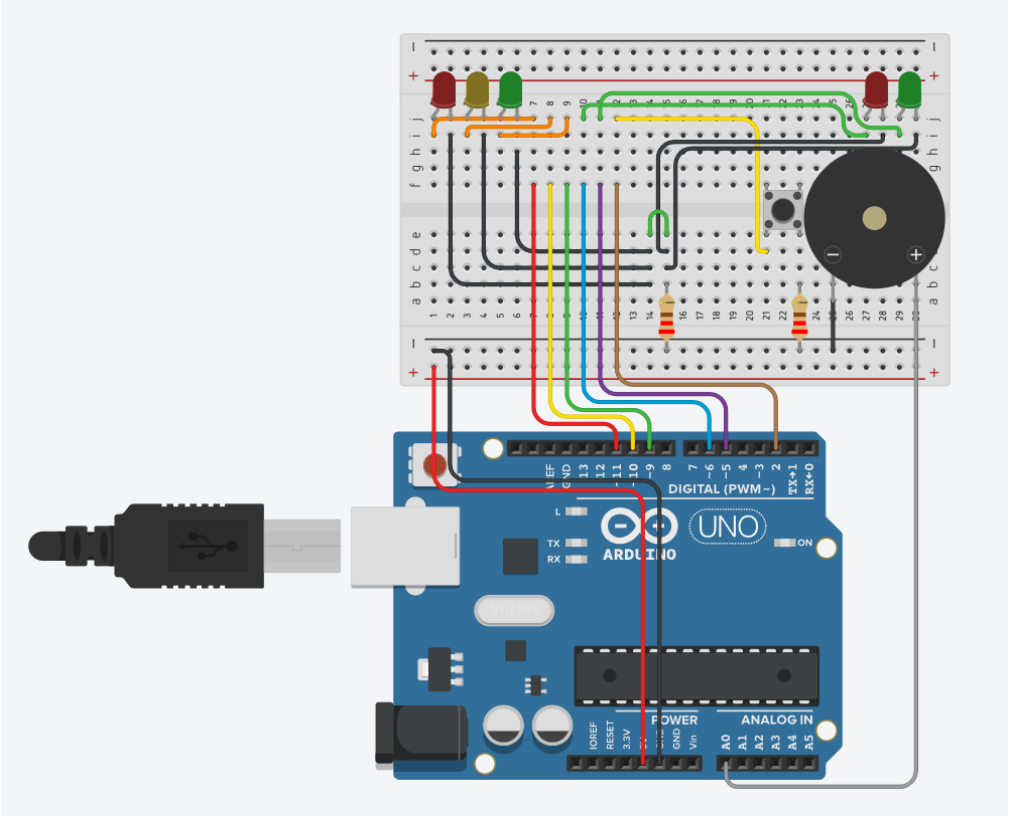
Additionally, a button is to be integrated into the circuit to provide an override function. When the button is pressed, all the traffic lights should be turned off, and an additional Green light should be activated for a duration of 7 seconds to facilitate the safe passage of students. After the additional Green light has completed its cycle, the traffic lights should revert to the initial state, starting with the Red light.

Solution Analysis and Design

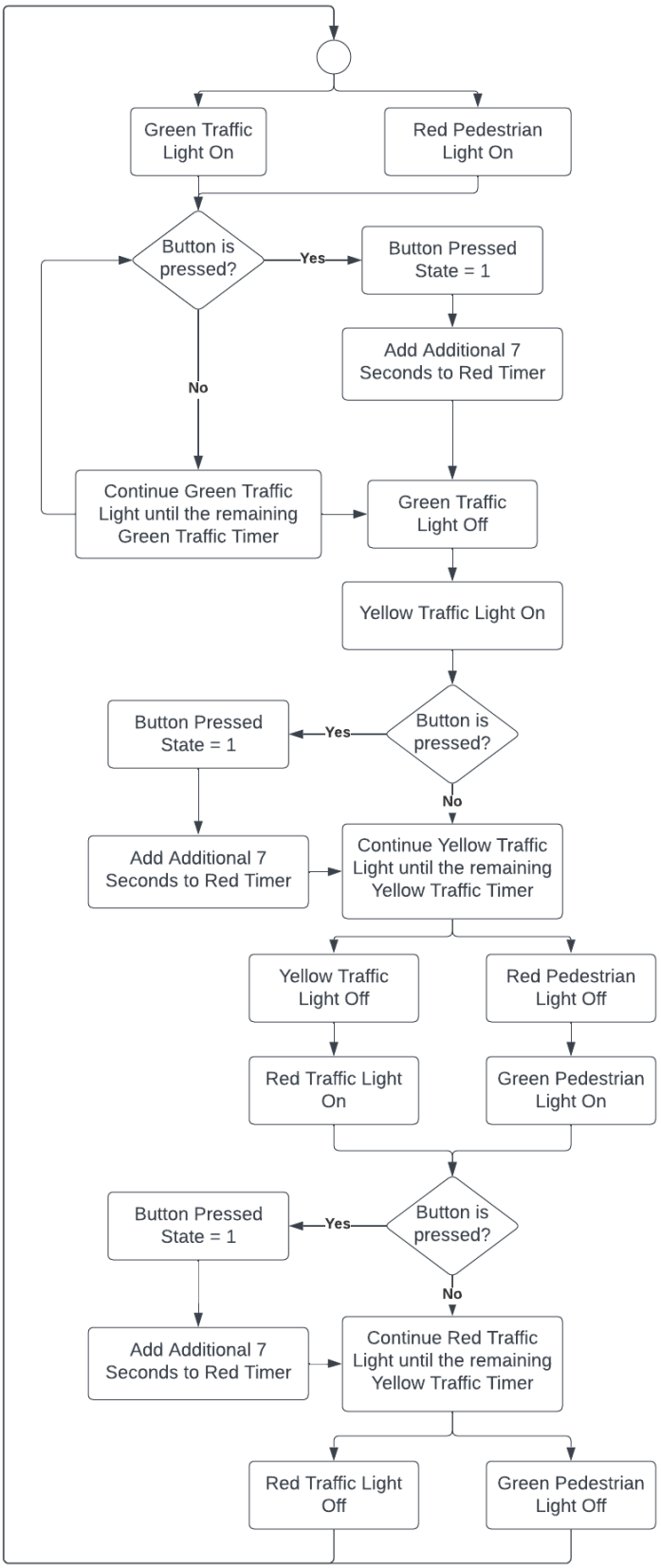
Components used for this project:

- Arduino UNO
- Jumper Wires
- 220 Ohm Resistor
- LEDs
- Piezo Buzzer
- Button

Schematics



Flowchart


Green Traffic Light On
Red Pedestrian Light On
Button is pressed?
Yes
Button Pressed State = 1
Add Additional 7 Seconds to Red Timer
No
Continue Green Traffic Light until the remaining Green Traffic Timer
Green Traffic Light Off
Yellow Traffic Light On
Button Pressed State = 1
Yes
Button is pressed?
No
Add Additional 7 Seconds to Red Timer
Continue Yellow Traffic Light until the remaining Yellow Traffic Timer
Yellow Traffic Light Off
Red Pedestrian Light Off
Red Traffic Light On
Green Pedestrian Light On
Button Pressed State = 1
Yes
Button is pressed?
No
Add Additional 7 Seconds to Red Timer
Continue Red Traffic Light until the remaining Yellow Traffic Timer
Red Traffic Light Off
Green Pedestrian Light Off

Solution implementation

1. Setup Function

```
// Classic setup
// Set the pin modes for each peripherals and set the default traffic light state
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(PEDESTRIAN_BUTTON_PIN, INPUT_PULLUP);
  for (int i = 0; i < 3; i++) {
    pinMode(trafficLights[i], OUTPUT);
  }
  for (int i = 0; i < 2; i++) {
    pinMode(pedestrianLights[i], OUTPUT);
  }
  currentLightPtr = &trafficLights[TRAFFIC_GREEN];
}
```

2. Switch Lights and Turn Off Lights Functions

```
// Function for switching the current traffic light state
void switchLights(uint8_t* lights, int light, unsigned long& currentMillis) {
  turnOffLights(lights);
  currentLightPtr = &lights[light]; // Set current light pointer to the selected light
  previousMillis = currentMillis; // Update the previousMillis everytime the light switches
}

// Function to turn off a set of lights given an array of pin numbers
void turnOffLights(uint8_t* lights) {
  // Calculate the number of lights in the array
  int numberOfLights = sizeof(lights) + 1;

  // Loop trough each light and turn it off
  for (int i = 0; i < numberOfLights; i++) {
    digitalWrite(lights[i], LOW);
  }
}
```

3. Loop Function (Lines 60-80)

```
60  void loop() {
61    unsigned long currentMillis = millis(); // Get the current board uptime and store it as currentMillis
62    unsigned long elapsedMillis = currentMillis - previousMillis; // Calculate the elapsed time since the previous timestamp
63
64    // Check if the button is pressed and whether it has been pressed before
65    if (!digitalRead(PEDESTRIAN_BUTTON_PIN) && pedestrianButtonPressed == 0) {
66      // Check if the current traffic light is green
67      if (*currentLightPtr == TRAFFIC_LIGHT_G_PIN) {
68        // Reset the elapsed time and update the previous timestamp
69        previousMillis = currentMillis;
70        elapsedMillis = 0;
71        delay(50); // Add delay to fix some shenanigans
72
73        // Immediately change the traffic light from green to yellow
74        currentLightPtr = &trafficLights[TRAFFIC_YELLOW];
75        Serial.print("CHANGE YELLOW");
76      }
77      // Turn off the traffic lights for updates and set the button state to pressed
78      turnOffLights(trafficLights);
79      pedestrianButtonPressed = 1;
80    }
```

In this code snippet, it checks whether the button is pressed. If the button is pressed, the traffic light will immediately change to yellow, and an additional 7 seconds will be added to the red traffic light timer.

4. Loop Function: Traffic Light State Switching Between Green and Yellow (Lines 81-118)

```
81
82    switch (*currentLightPtr) {
83      case TRAFFIC_LIGHT_G_PIN:
84        // Check if it's time to switch from green light
85        if (elapsedMillis >= greenInterval) {
86          switchLights(trafficLights, TRAFFIC_YELLOW, currentMillis);
87          Serial.print("CHANGE YELLOW ");
88        } else {
89          // If the elapsed time has not reached turn on green traffic light and red pedestrian light
90          noTone(PEDESTRIAN_PIEZO);
91          analogWrite(*currentLightPtr, 255);
92          analogWrite(PEDESTRIAN_LIGHT_R_PIN, 255);
93        }
94        break;
95      case TRAFFIC_LIGHT_Y_PIN:
96        // Check if it's time to switch from yellow light and turn off pedestrian lights to update it
97        if (elapsedMillis >= yellowInterval) {
98          switchLights(trafficLights, TRAFFIC_RED, currentMillis);
99          turnOffLights(pedestrianLights);
100         Serial.print("CHANGE RED ");
101       } else {
102         // Make it so that it will sound alarm if the button is pressed
103         if (pedestrianButtonPressed) {
104           if (currentMillis - previousFlickerTime >= 500) {
105             if (isFlickering) {
106               tone(PEDESTRIAN_PIEZO,0);
107             } else {
108               tone(PEDESTRIAN_PIEZO, 400);
109             }
110             isFlickering = !isFlickering;
111             previousFlickerTime = currentMillis;
112           }
113         }
114         // If the elapsed time has not reached turn on yellow traffic light and red pedestrian light
115         analogWrite(*currentLightPtr, 255);
116         analogWrite(PEDESTRIAN_LIGHT_R_PIN, 255);
117       }
118       break;
```

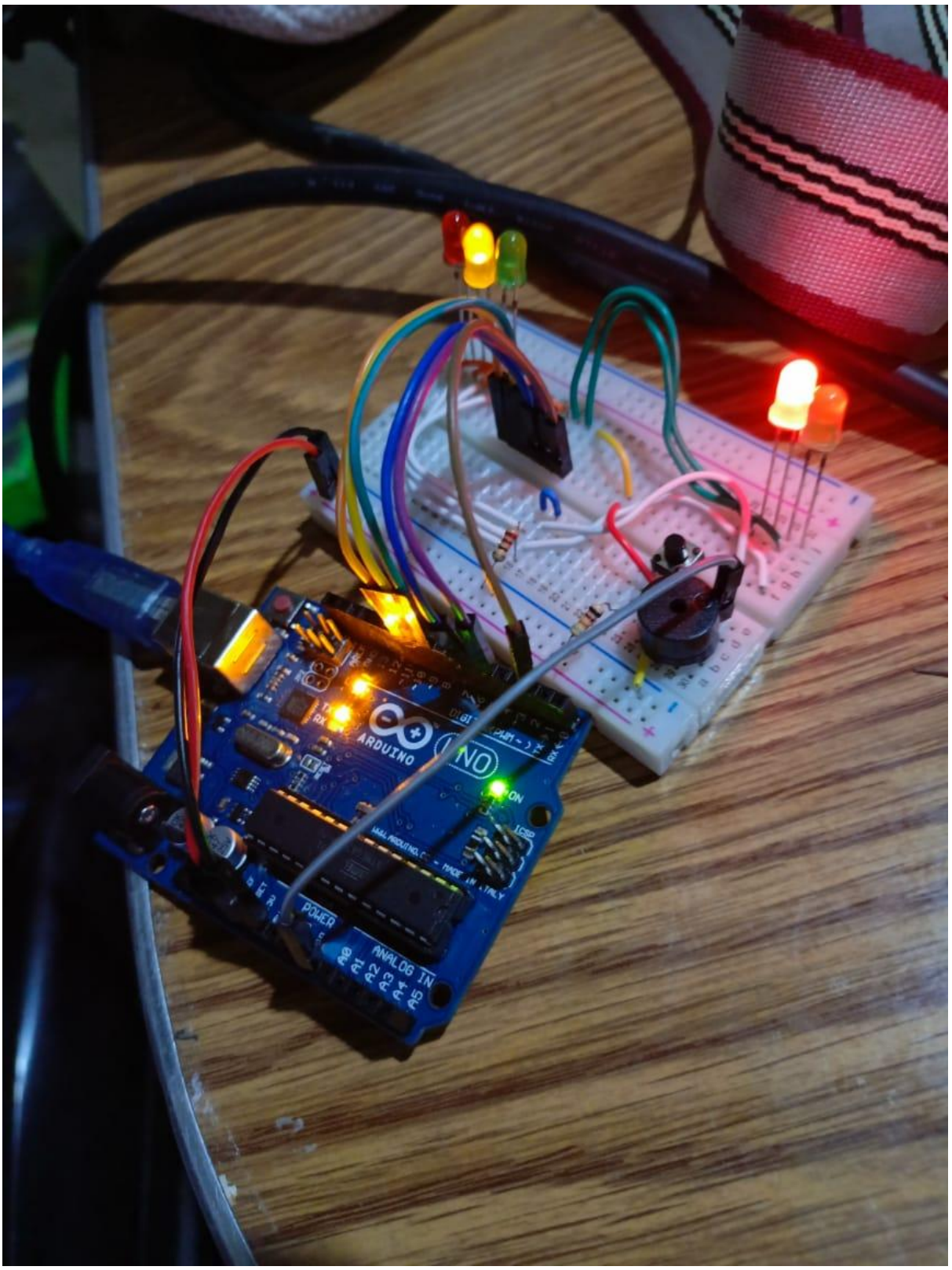
In this code snippet, the light will change based on the current light pointer. If the duration of each set light has elapsed, it will then switch to the next light in the sequence.

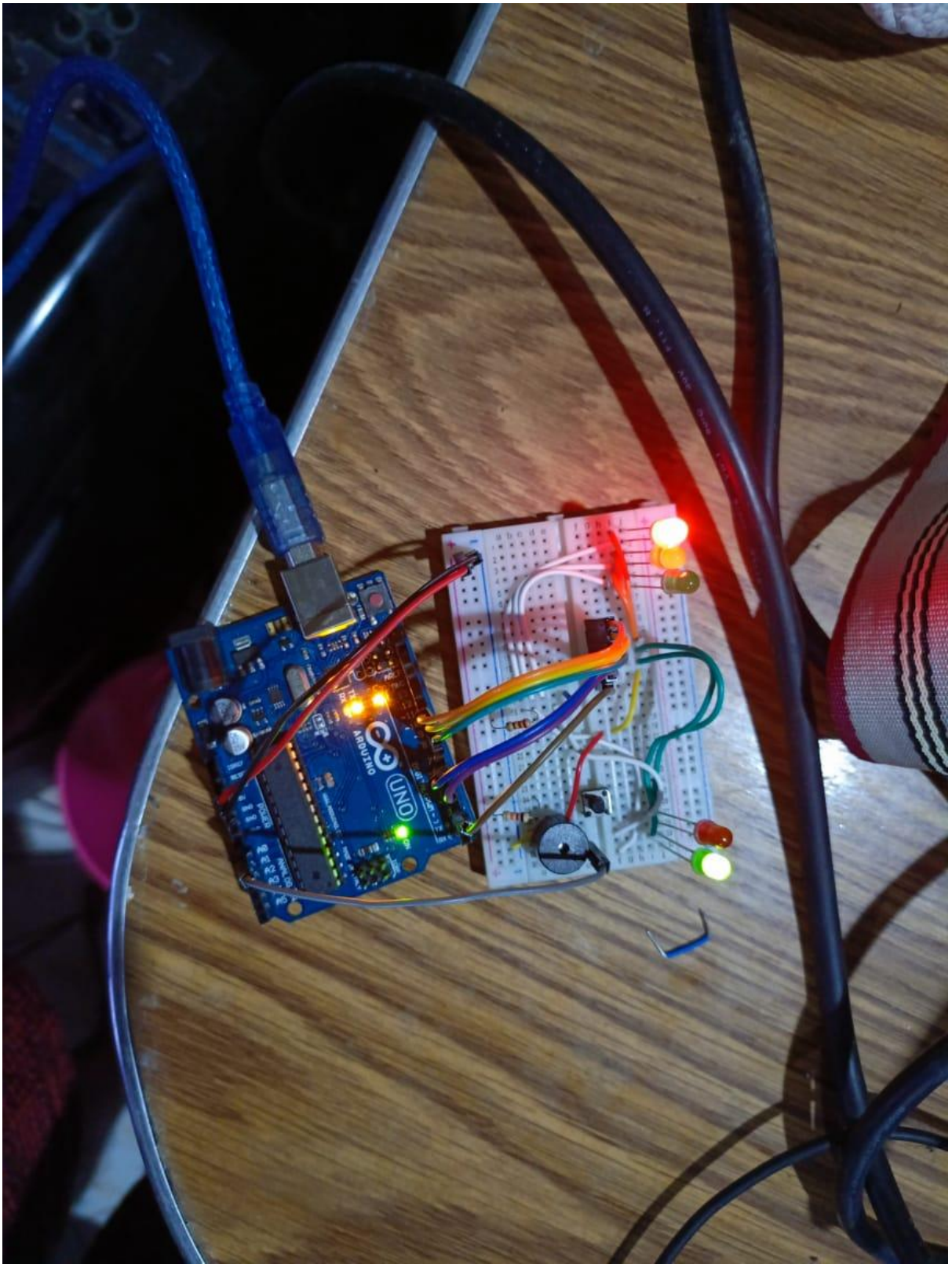
5. Loop Function: Red Light State Switching(Lines 119-160)

```
119     case TRAFFIC_LIGHT_R_PIN:
120       // Calculate the red traffic light duration depending on button press
121       long realRedInterval = redInterval + redAdditionalInterval * pedestrianButtonPressed;
122
123       // Check if it's time to switch from red light, turn off pedestrian lights to update it and reset the button pressed state
124       if (elapsedMillis >= realRedInterval) {
125         switchLights(trafficLights, TRAFFIC_GREEN, currentMillis);
126         turnOffLights(pedestrianLights);
127         Serial.print("CHANGE GREEN ");
128         pedestrianButtonPressed = 0;
129       } else if (elapsedMillis >= realRedInterval - 3000 && pedestrianButtonPressed) {
130         // When the red traffic light timer reaching the end it will flicker the green pedestrian light additionaly it will sound different alarm
131         if (currentMillis - previousFlickerTime >= 500) {
132           if (isFlickering) {
133             analogWrite(PEDESTRIAN_LIGHT_G_PIN, 255);
134             tone(PEDESTRIAN_PIEZO, 600);
135           } else {
136             analogWrite(PEDESTRIAN_LIGHT_G_PIN, 0);
137             noTone(PEDESTRIAN_PIEZO);
138           }
139           isFlickering = !isFlickering;
140           previousFlickerTime = currentMillis;
141         }
142       } else {
143         // Make it so that it will sound alarm if the button is pressed, the alarm sound different from the yellow alarm
144         if (pedestrianButtonPressed) {
145           if (currentMillis - previousFlickerTime >= 500) {
146             if (isFlickering) {
147               tone(PEDESTRIAN_PIEZO, 600);
148             } else {
149               tone(PEDESTRIAN_PIEZO, 400);
150             }
151             isFlickering = !isFlickering;
152             previousFlickerTime = currentMillis;
153           }
154         }
155         // If the elapsed time has not reached turn on red traffic light and green pedestrian light
156         analogWrite(PEDESTRIAN_LIGHT_G_PIN, 255);
157         analogWrite(*currentLightPtr, 255);
158       }
159       break;
160   }
```

In this code snippet, which is a continuation of the previous section, an alarm will sound when the button is pressed.when the traffic light is red.

Appendix


ARDUINO
UNO
POWER
ANALOG IN


ARDUINO
UNO