HW 4

Course Name: Special Topics/Problems in CS: Cyber Physical Systems

Course Number: CS-5331

Submitted By: AMIR FAIYAZ

Student ID: R11772642

Submission Date: 2/28/2023

a.

Objective: The objective of this project was to simulate a traffic light system using an Arduino microcontroller and related hardware components. The system needed to include a normal traffic light sequence, as well as the ability to detect pedestrian crossings and trigger a pedestrian crossing sequence.

Procedure: The project was built with an Arduino Uno board, LED lights, switches, and resistors. The system was designed with two sets of traffic lights for opposing traffic directions, as well as pedestrian crossing buttons at each set of lights. The Arduino code was written to implement the standard traffic light sequence, as well as to detect when a pedestrian button was pressed and switch to the pedestrian crossing sequence.

Key Results: This project's goal was to simulate a traffic light system using an Arduino microcontroller and related hardware components. The system had to support both a standard traffic light sequence and the ability to detect pedestrian crossings and initiate a pedestrian crossing sequence.

b)

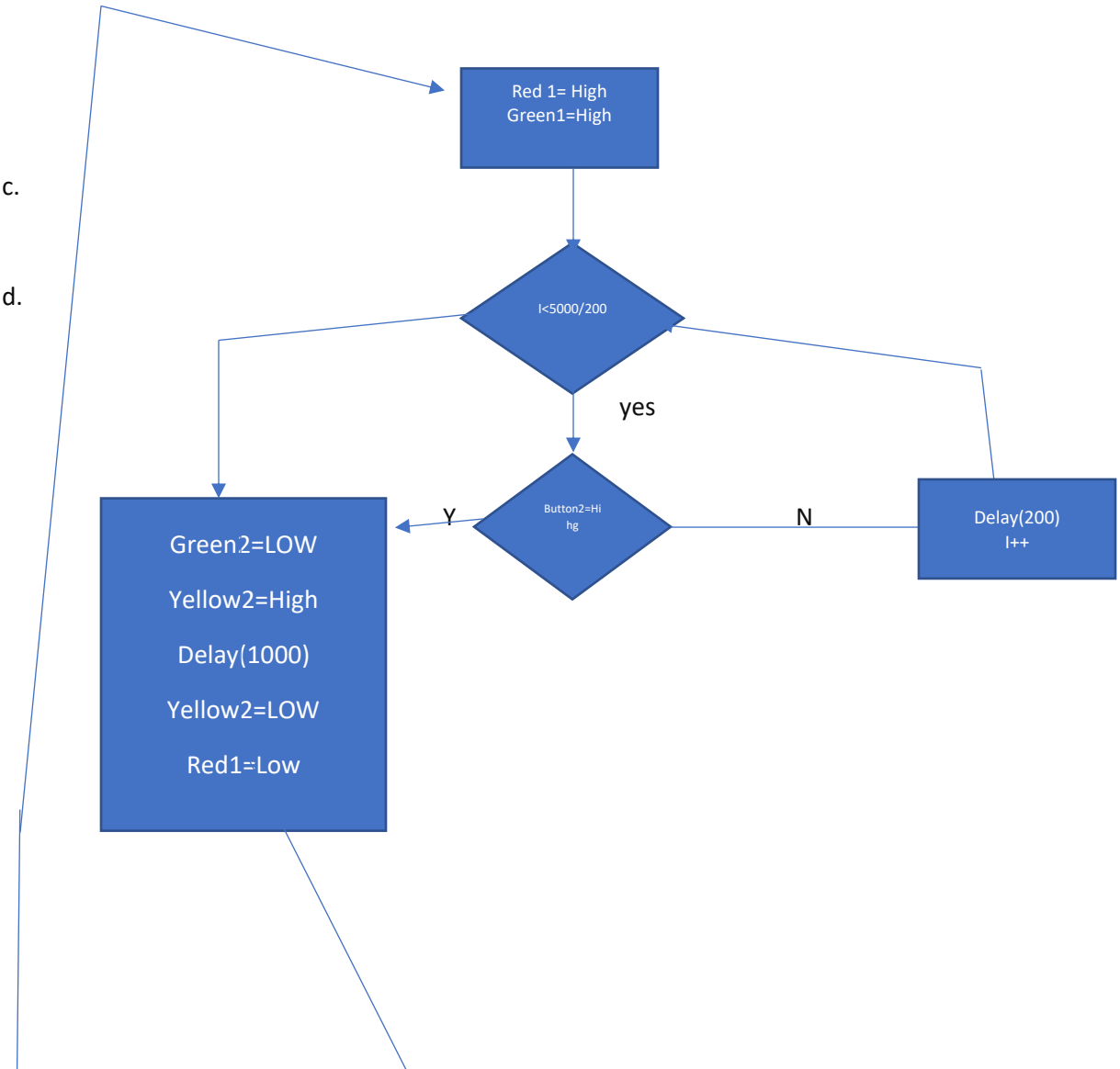
```
 int red1 = 2;

int green1 = 3;
int yellow1 = 4;
int red2 = 6;
int green2 = 7;
int yellow2 = 8;
const int button1 = 5;
const int button2 = 9;

void setup() {
  pinMode(red1, OUTPUT);
  pinMode(green1, OUTPUT);
  pinMode(yellow1, OUTPUT);
  pinMode(red2, OUTPUT);
  pinMode(green2, OUTPUT);
  pinMode(yellow2, OUTPUT);
  pinMode(button1, INPUT);
```

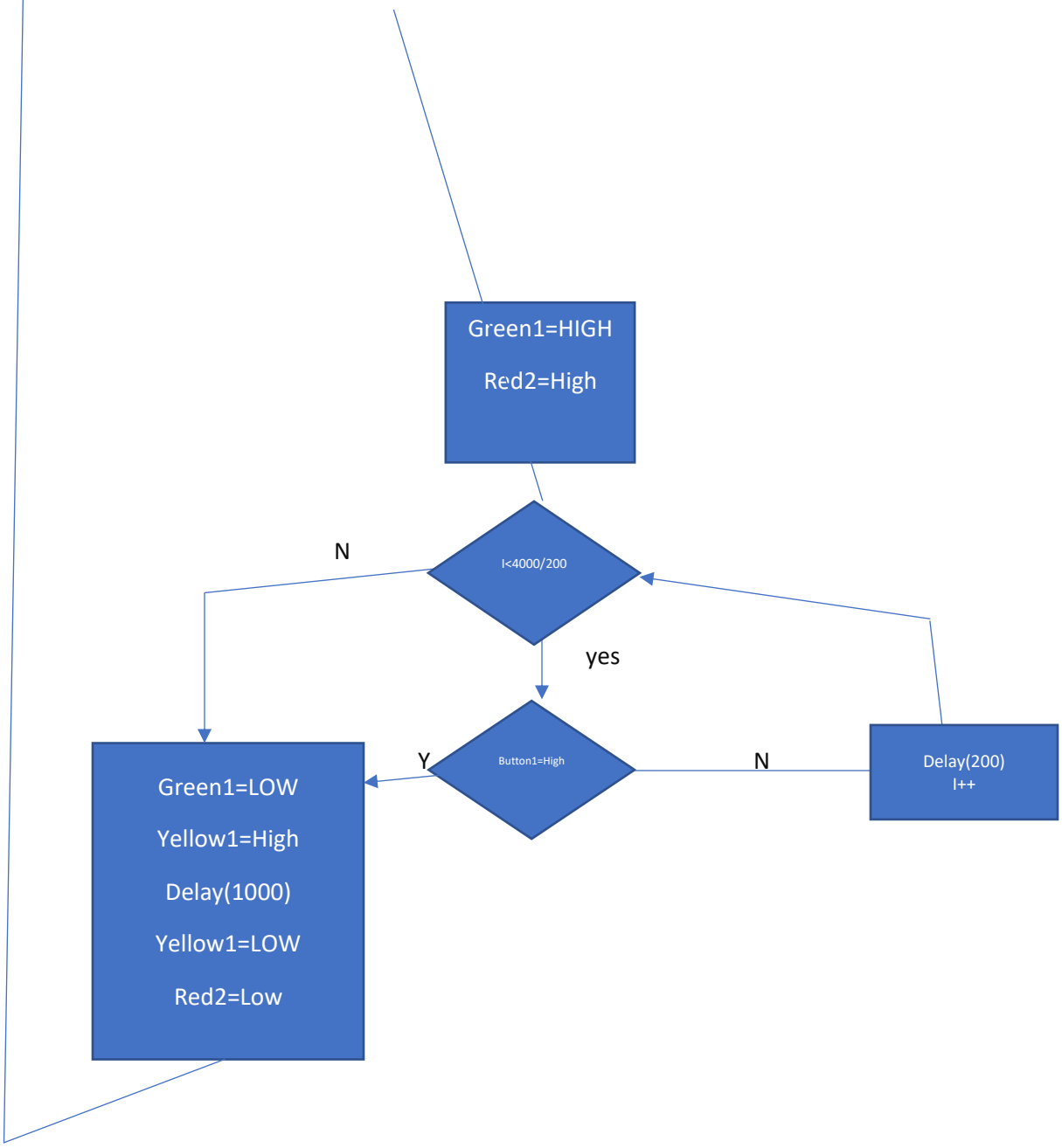
```
  pinMode(button2, INPUT);
}

void loop() {
  // Street 1 - Red, Street 2 - Green
  digitalWrite(red1, HIGH);
  digitalWrite(green2, HIGH);

  for(int i = 0; i < 5000/200; i++) {
    if(digitalRead(button2) == HIGH) {
      // Button pressed on Street 2
      digitalWrite(green2, LOW);
      digitalWrite(yellow2, HIGH);
      delay(1000);
      digitalWrite(red1, LOW);
      digitalWrite(yellow2, LOW);
      break;
    }
    delay(200);
  }

  if(digitalRead(button2) == LOW) {
    // No button press on Street 2
    digitalWrite(green2, LOW);
    digitalWrite(yellow2, HIGH);
    delay(1000);
    digitalWrite(yellow2, LOW);
    digitalWrite(red1, LOW);
  }

  // Street 1 - Green, Street 2 - Red
  digitalWrite(green1, HIGH);
  digitalWrite(red2, HIGH);

  for(int i = 0; i < 4000/200; i++) {
    if(digitalRead(button1) == HIGH) {
      // Button pressed on Street 1
      digitalWrite(green1, LOW);
      digitalWrite(yellow1, HIGH);
      delay(1000);
      digitalWrite(yellow1, LOW);
      digitalWrite(red2, LOW);
      break;
    }
    delay(200);
```

```
  }

  if(digitalRead(button1) == LOW) {
    // No button press on Street 1
    digitalWrite(green1, LOW);
    digitalWrite(yellow1, HIGH);
    delay(1000);
    digitalWrite(yellow1, LOW);
    digitalWrite(red2, LOW);
  }
}
```

c.

d.





d) The pedestrian switch is tested in the loop function of the implemented code using the amberDelay variable after a delay.

Due to the delay caused by the execution of the code and the processing of the switch input, the switch does not respond instantly. The timing of the code execution will determine the best-case delay, and the worst-case delay will be determined by adding the processing times for the code and the switch input.

One can optimize the code by eliminating pointless delays and utilizing interrupt-based programming to increase reaction time.

Interrupts decrease the time it takes for the software to check the switch state in the loop function and enable it to respond nearly instantly to events like the switch input. Moreover, removing noise from the switch signal and avoiding erroneous pedestrian signal triggering can be accomplished by either hardware debouncing circuits or software debouncing approaches.

E)

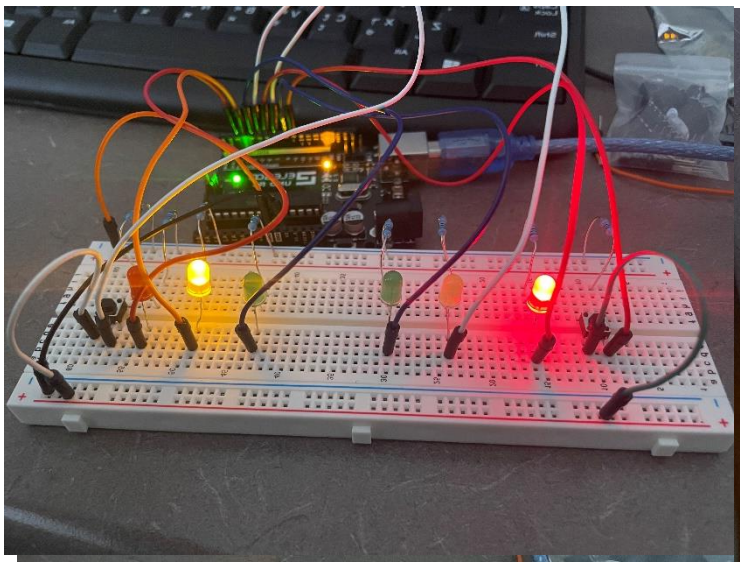
sketch_feb28a.ino

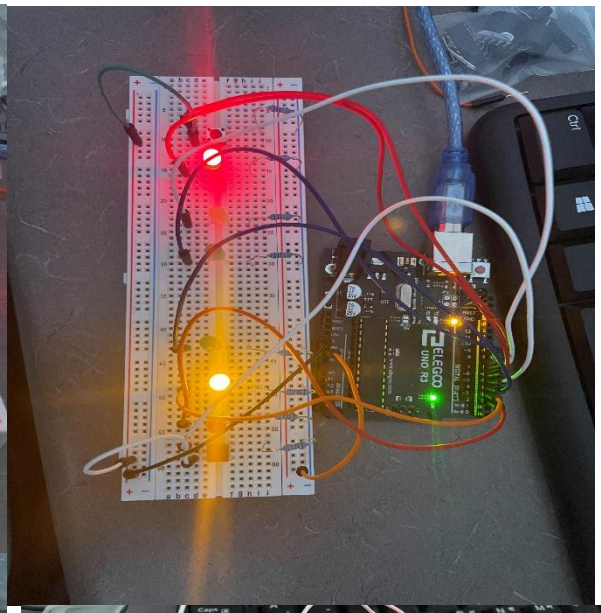
```
int red1 = 2;
int green1 = 3;
int yellow1 = 4;
int red2 = 6;
int green2 = 7;
int yellow2 = 8;
const int button1 = 5;
const int button2 = 9;

void setup() {
  pinMode(red1, OUTPUT);
  pinMode(green1, OUTPUT);
  pinMode(yellow1, OUTPUT);
  pinMode(red2, OUTPUT);
  pinMode(green2, OUTPUT);
  pinMode(yellow2, OUTPUT);
  pinMode(button1, INPUT);
  pinMode(button2, INPUT);
}

void loop() {
  // Street 1 - Red, Street 2 - Green
  digitalWrite(red1, HIGH);
```

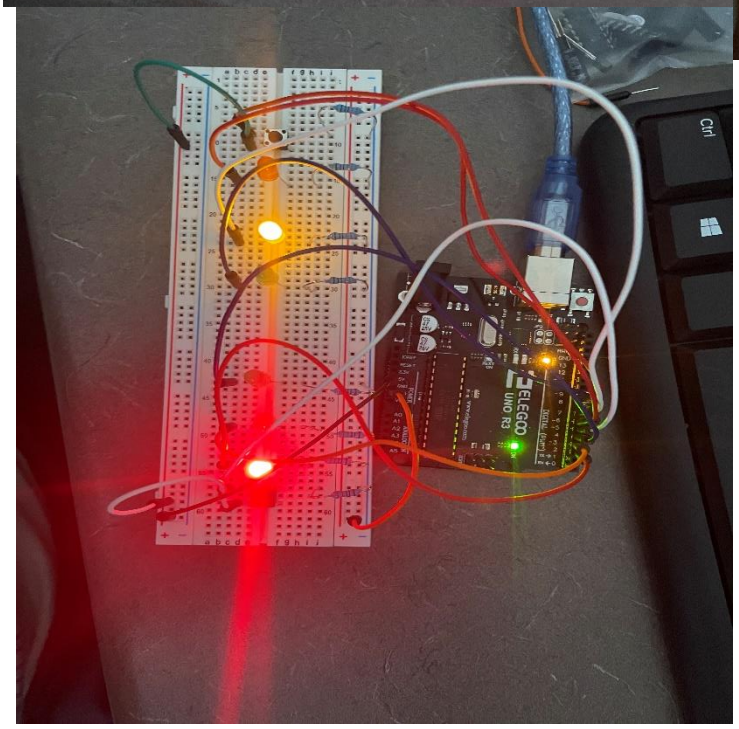
Output

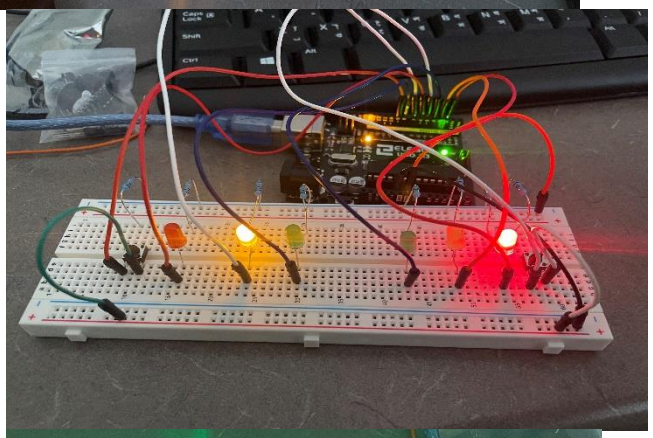
```
Sketch uses 1430 bytes (4%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

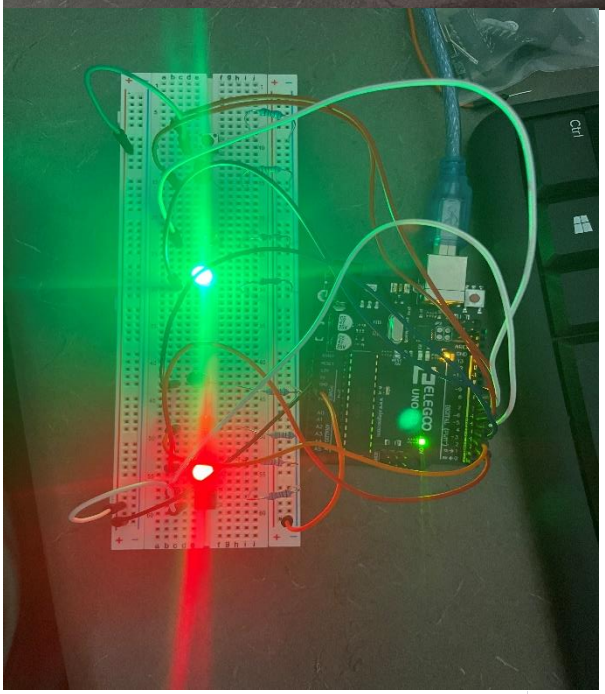
F) Images below: -











g.

Technical difficulties can arise when working with hardware systems such as a traffic system. Issues with wiring, incorrect coding syntax or logic, or malfunctions with components such as LEDs or switches are all possible challenges.