Traffic Control by Using Arduino UNO

**Project Tasks**

**Supervisor**

**Khandaker Iftakher Ahmed**

Lecturer C.C. of CSE

European University of Bangladesh

**Submitted By**

[**230122003, 230122004, 230122015**

**230122031, 230122077**]

Students of CSE

European University of Bangladesh

[21 March, 2025]

**Department of Computer Science & Engineering**

European University of Bangladesh.

**Traffic Control by Using Arduino UNO**

*Faculty of Computer Science & Engineering, European University of Bangladesh.*

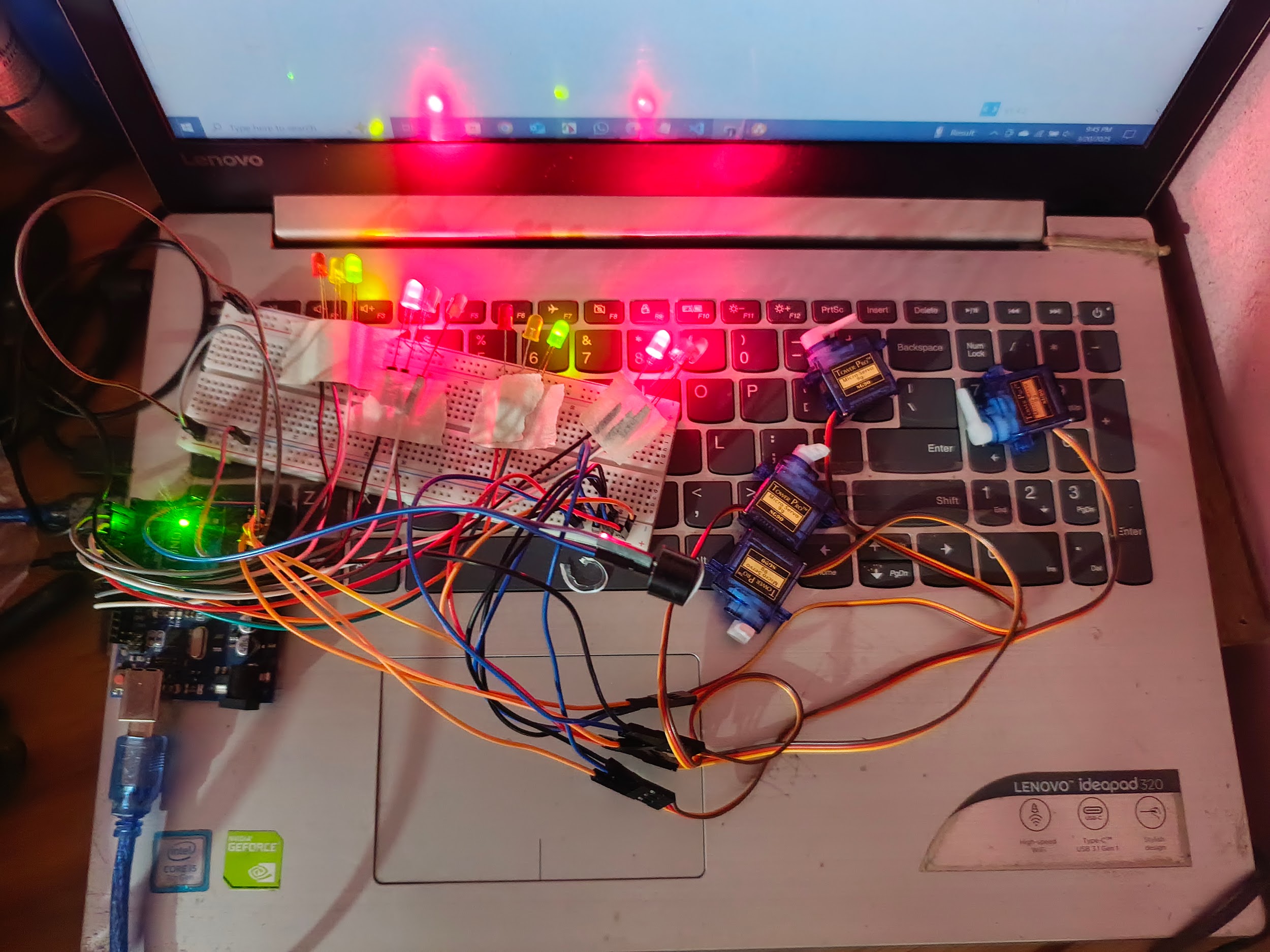
***A****bstract*— Our main goal is to develop an ultramodern and reliable traffic control by using arduino uno. Which is faced in almost every field in our daily life. For example road crossing, human, car, bus, motor-cycle, three-wheelers, etc. To develop this system, we used an Arduino UNO as the main processor, LEDs, servo motors, buzzers, resistors, jumper wires, a 9V battery, and other components.

**Methodology**

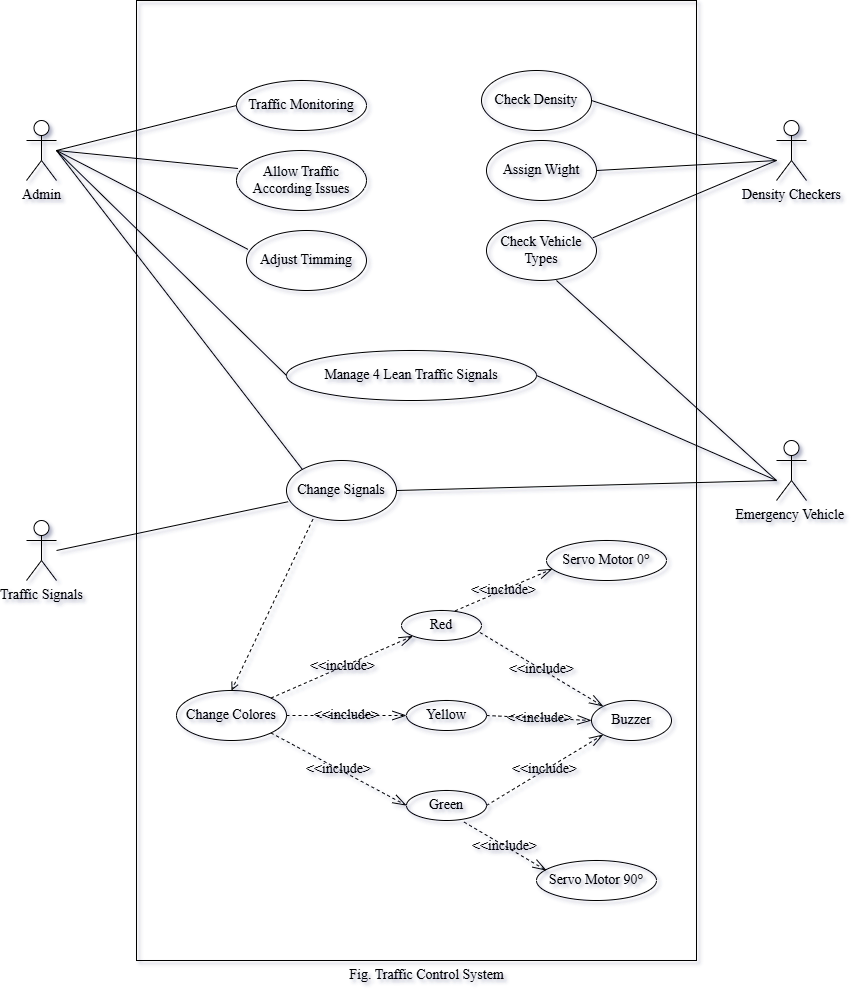
**Equipments:**

* Arduino UNO
* BreadBoard
* Buzzer
* Servo Motor
* LEDs (Red, Yellow, Green)
* Resistors
* Jumper Wires
* 9V Battery

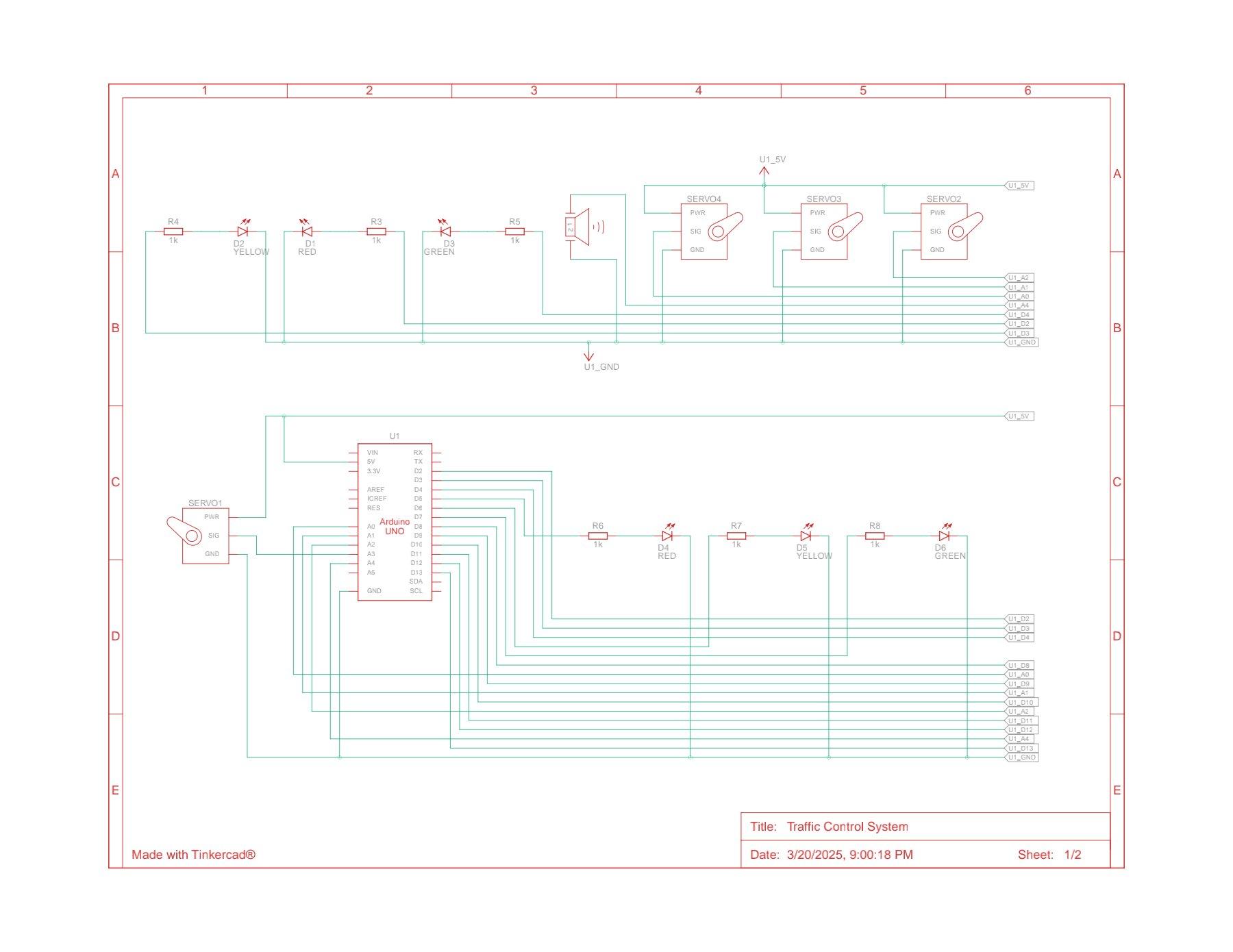
\*To enhance the project's quality, we can modify the materials as needed. The selection of materials will depend on the tasks that can be completed within the project's limited timeframe.



**Use Case Diagram:** A use case diagram, a type of Unified Modeling Language (UML) diagram, visually represents the interactions between users (actors) and a system, outlining the different ways a user can interact with and achieve goals within the system.



**Circuit Diagram:** To develop this system, we used an Arduino UNO as the main processor, LEDs, servo motors, buzzers, resistors, jumper wires, a 9V battery, and other components.



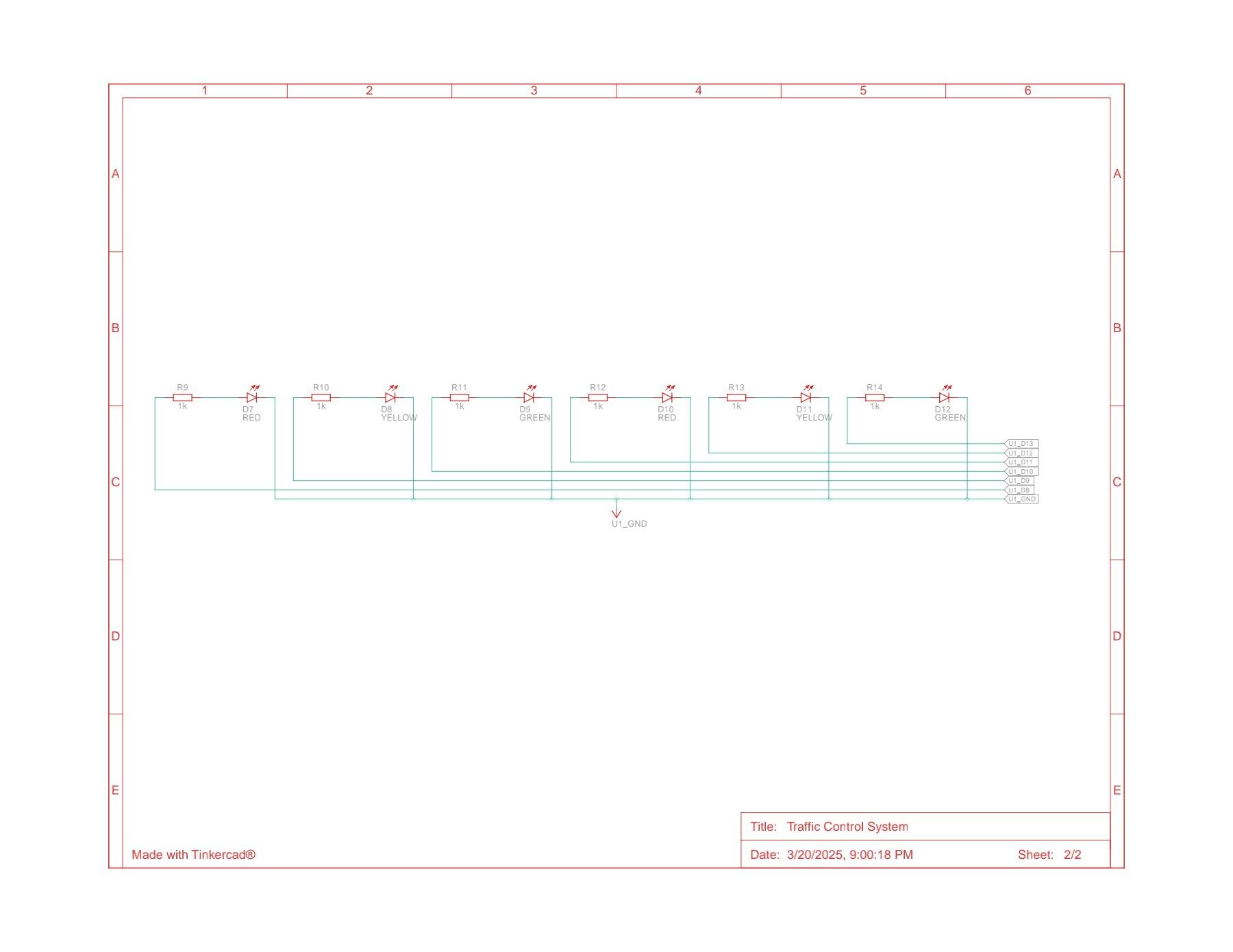


Fig. Circuit Diagram

**Connection Diagram:** A connection diagram, also known as a wiring diagram, is a visual representation that shows how electrical components are connected, including their relative positions and the connections between them, often using standardized symbols and lines.

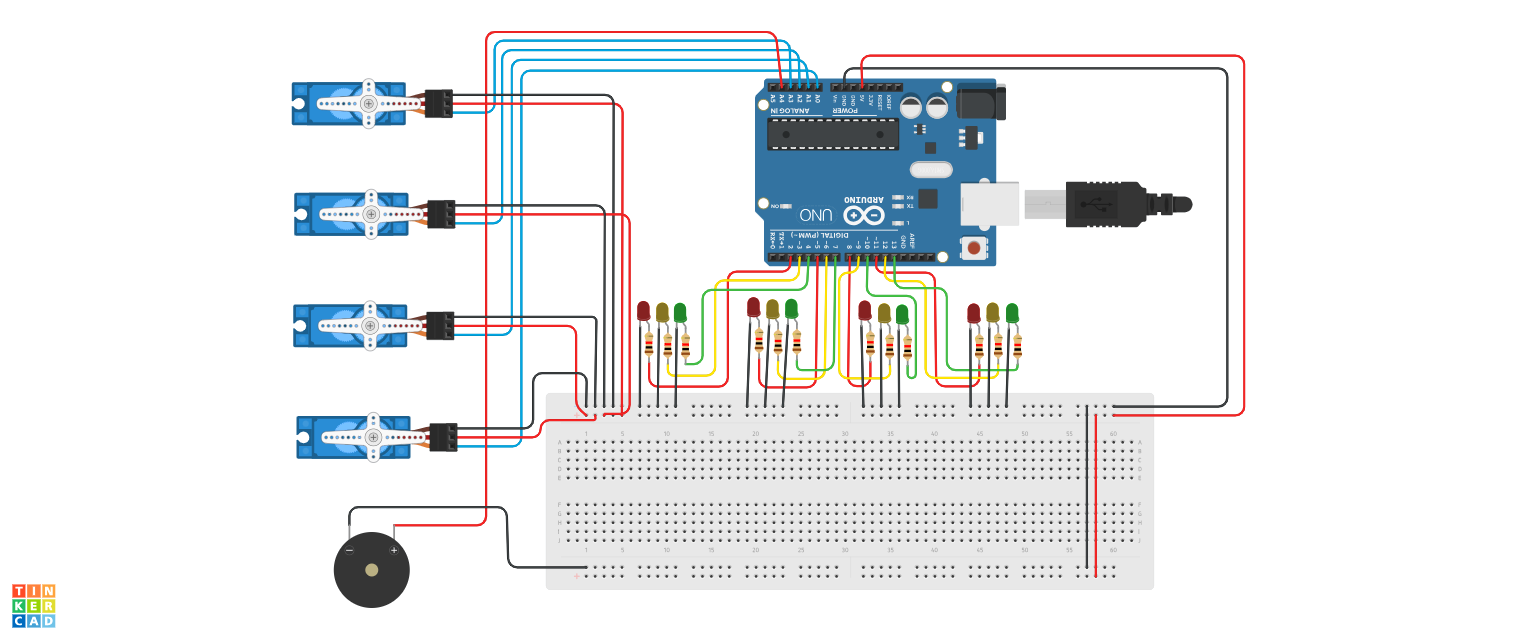


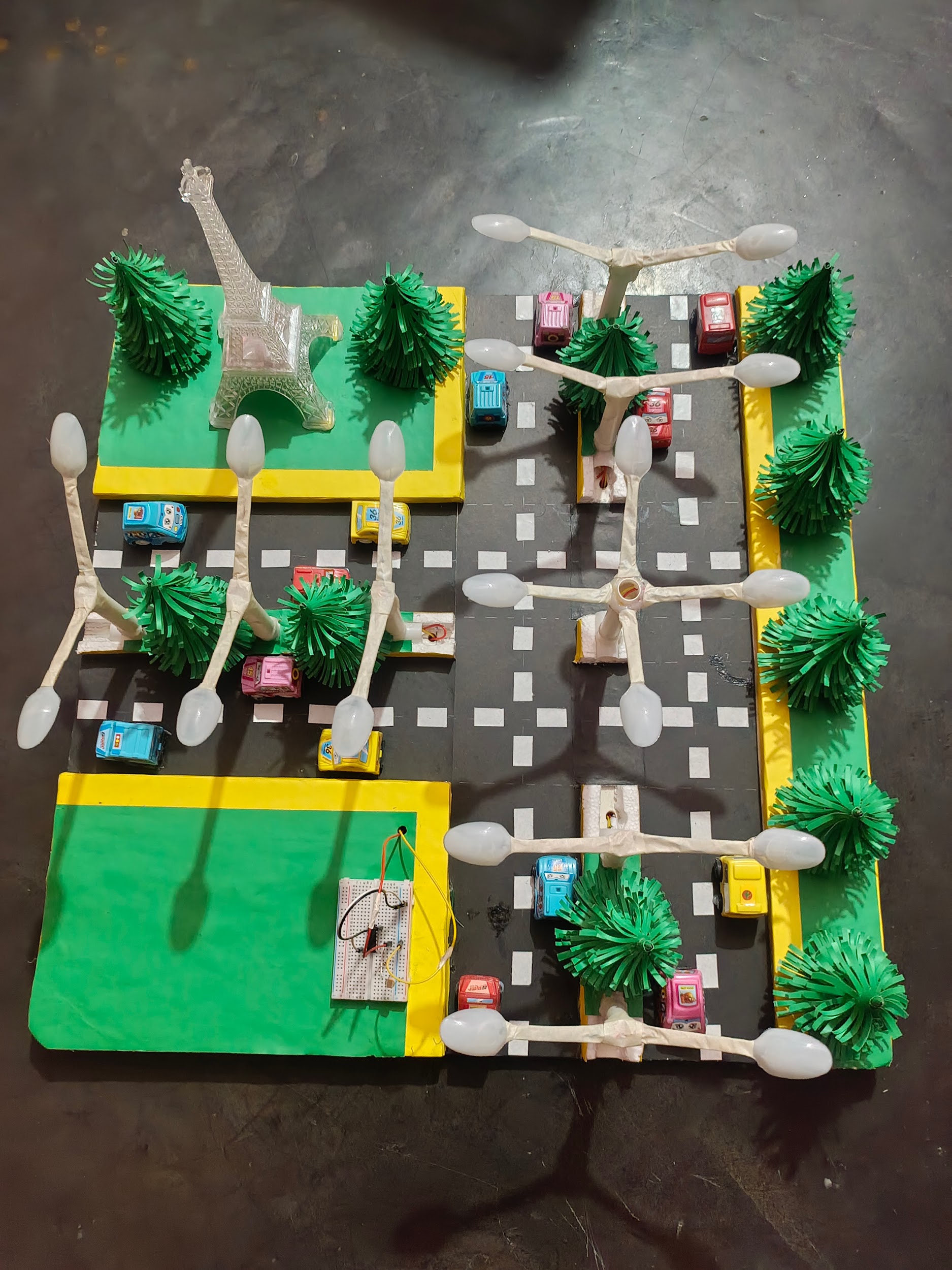
Fig. Connection Diagram

**Arduino UNO Code:**

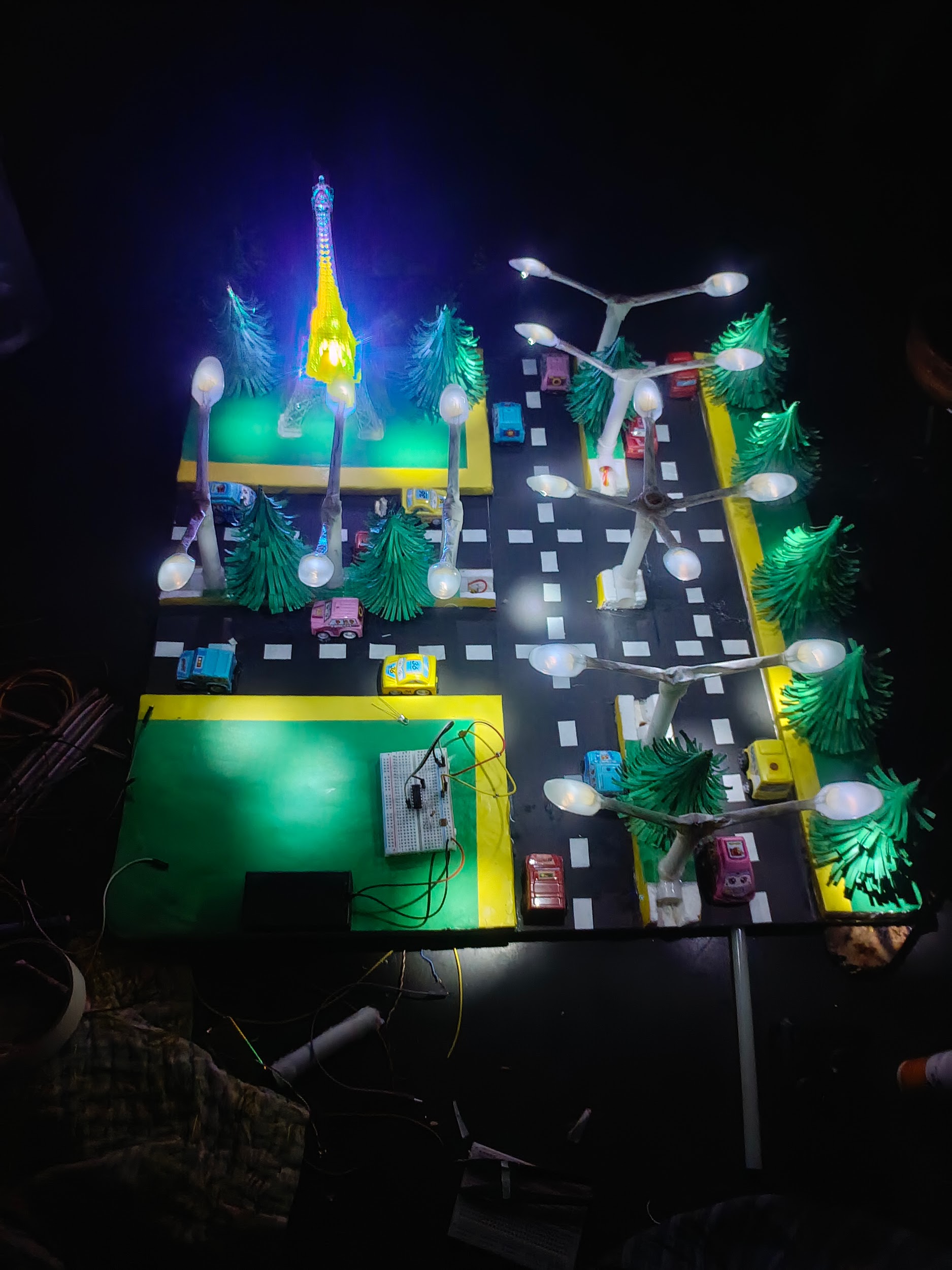
| #include <Servo.h>  *// Create servo objects for 4 lanes* Servo servo1, servo2, servo3, servo4;  *// Define servo motor pins (separate from LED pins)* int servoPins[4] = {A0, A1, A2, A3}; *// Using Analog pins for servos (A0-A3)*  *// Define LED pins (4 lanes × 3 LEDs each)* int ledPins[4][3] = {  {2, 3, 4}, *// Lane 1: Red, Yellow, Green*  {5, 6, 7}, *// Lane 2: Red, Yellow, Green*  {8, 9, 10}, *// Lane 3: Red, Yellow, Green*  {11, 12, 13} *// Lane 4: Red, Yellow, Green* };  *// Define Buzzer pin* const int buzzer = A4;  void setup() {  *// Set LED pins as outputs*  for (int i = 0; i < 4; i++) {  for (int j = 0; j < 3; j++) {  pinMode(ledPins[i][j], OUTPUT);  }  }   *// Attach servos*  servo1.attach(servoPins[0]);  servo2.attach(servoPins[1]);  servo3.attach(servoPins[2]);  servo4.attach(servoPins[3]);   *// Set all servos to initial closed position*  servo1.write(0);  servo2.write(0);  servo3.write(0);  servo4.write(0);   *// Set Buzzer as output*  pinMode(buzzer, OUTPUT); }  void loop() {  *// \*\*Step 1 & 3: Lane 1 & Lane 3 Green (Lane 2 & Lane 4 Red)\*\**  controlTraffic(0, 2, 90); *// Lane 1: Green, Servo Open*  controlTraffic(2, 2, 90); *// Lane 3: Green, Servo Open*  controlTraffic(1, 0, 0); *// Lane 2: Red, Servo Closed*  controlTraffic(3, 0, 0); *// Lane 4: Red, Servo Closed*  buzzerSignal(2000, 3); *// Short "Go" beeps for Green*  delay(5000); *// Green light duration*   *// \*\*Step 2 & 4: Lane 1 & 3 Yellow -> Red, Lane 2 & 4 Green\*\**  controlTraffic(0, 1, 0); *// Lane 1: Yellow*  controlTraffic(2, 1, 0); *// Lane 3: Yellow*  buzzerSignal(1500, 1); *// Short warning beep for Yellow*  delay(2000); *// Yellow delay*   controlTraffic(0, 0, 0); *// Lane 1: Red*  controlTraffic(2, 0, 0); *// Lane 3: Red*  controlTraffic(1, 2, 90); *// Lane 2: Green, Servo Open*  controlTraffic(3, 2, 90); *// Lane 4: Green, Servo Open*  buzzerSignal(2000, 3); *// Short "Go" beeps for Green*  delay(5000); *// Green light duration*   *// \*\*Repeat the cycle\*\**  controlTraffic(1, 1, 0); *// Lane 2: Yellow*  controlTraffic(3, 1, 0); *// Lane 4: Yellow*  buzzerSignal(1500, 1); *// Short warning beep for Yellow*  delay(2000);   controlTraffic(1, 0, 0); *// Lane 2: Red*  controlTraffic(3, 0, 0); *// Lane 4: Red*  buzzerSignal(1000, 1); *// Long beep for Red* }  *// Function to control traffic lights and servos* void controlTraffic(int lane, int signal, int servoAngle) {  *// Turn OFF all LEDs for the lane*  for (int i = 0; i < 3; i++) {  digitalWrite(ledPins[lane][i], LOW);  }  *// Turn ON selected LED*  digitalWrite(ledPins[lane][signal], HIGH);   *// Move servo to correct position*  switch (lane) {  case 0: servo1.write(servoAngle); break;  case 1: servo2.write(servoAngle); break;  case 2: servo3.write(servoAngle); break;  case 3: servo4.write(servoAngle); break;  } }  *// Function to signal using the buzzer* void buzzerSignal(int frequency, int count) {  for (int i = 0; i < count; i++) {  tone(buzzer, frequency);  delay(300);  noTone(buzzer);  delay(300);  } } |
| --- |

**Project Controlling:** Project controlling, a crucial element of project management, involves monitoring and managing a project's progress to ensure it stays on track, achieves goals, and stays within budget and timeline, by comparing actual performance against planned objectives and taking corrective actions when necessary.

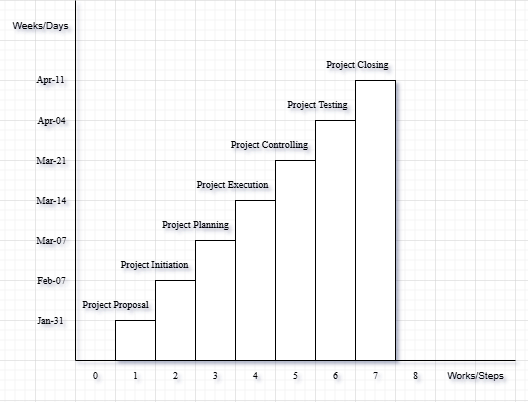
\*When light mode,

****

\*When night mode,

****

# **Milestones**



**References**

We mentioned all the literature or web references, etc here-

* Internet web page & resources: [ResearchGate](https://www.researchgate.net/profile/Md-Abu-Sayed-24), Gemini, etc.
* TinkerCad circuit simulation software:

[](https://www.tinkercad.com/things/a0ubEAdp67Q-traffic-control-system?sharecode=xx0k-L8L7O1WeQGpxyaN1SCgtBETq1BPRXW4US2GfGc)

<https://www.tinkercad.com/things/a0ubEAdp67Q-traffic-control-system?sharecode=xx0k-L8L7O1WeQGpxyaN1SCgtBETq1BPRXW4US2GfGc>

End!