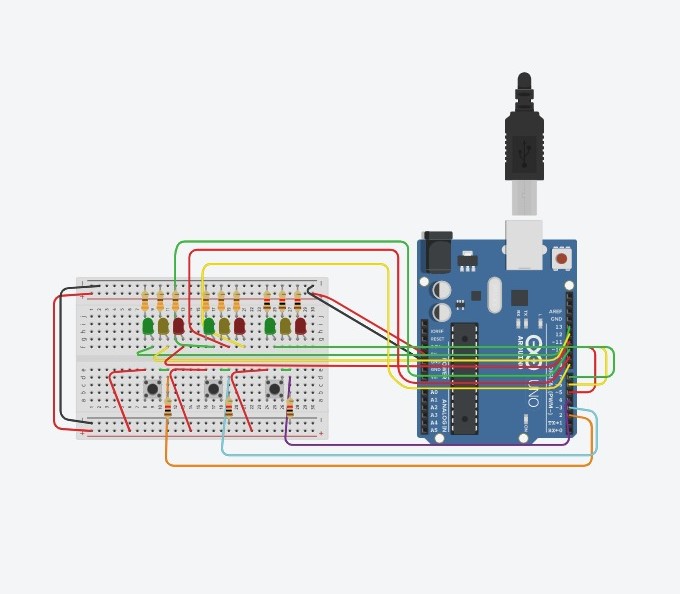
Phase 3 – development part 1

# **Project title - TRAFFIC MANAGEMENT SYSTEM**



## **CODING**

int s1=1;

int s2=0;

void setup()

{

pinMode(2, INPUT);

pinMode(13, OUTPUT);

pinMode(12, OUTPUT);

pinMode(11, OUTPUT);

pinMode(3, INPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

pinMode(8, OUTPUT);

pinMode(2, INPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(5, OUTPUT);

pinMode(3, INPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

pinMode(8, OUTPUT);

}

void loop()

{

if (digitalRead(2) == 1) {

on1();

}else if(digitalRead(3) == 1) {

on2();

}else{

if(s1){

on2();

}else if (s2){

on1();

}

}

}

void on1(){

off3();

s1=1;

s2=0;

digitalWrite(11, LOW);

digitalWrite(13, HIGH);

delay(3000); // Wait for 3000 millisecond(s)

digitalWrite(13, LOW);

digitalWrite(12, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(12, LOW);

digitalWrite(11, HIGH);

}

void off1(){

digitalWrite(13, LOW);

digitalWrite(12, LOW);

digitalWrite(11, HIGH);

}

void on2(){

off1();

s2=1;

s1=0;

digitalWrite(8, LOW);

digitalWrite(10, HIGH);

delay(3000); // Wait for 3000 millisecond(s)

digitalWrite(10, LOW);

digitalWrite(9, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(9, LOW);

digitalWrite(8, HIGH);

}

void on3(){

off3();

s1=1;

s2=0;

digitalWrite(6, LOW);

digitalWrite(7, HIGH);

delay(3000); // Wait for 3000 millisecond(s)

digitalWrite(7, LOW);

digitalWrite(6, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(6, LOW);

digitalWrite(5, HIGH);

}

void off3(){

digitalWrite(7, LOW);

digitalWrite(6, LOW);

digitalWrite(5, HIGH);

}

## **THE ASPECTS:**

**1.Components and Hardware**:

* **Arduino Uno R3**: The Arduino will serve as the main microcontroller for your system.
* **Sensors**: You will need various sensors, such as ultrasonic sensors for distance measurement, IR sensors for vehicle detection, and potentially cameras for image processing.
* **Actuators**: For controlling traffic lights or gates, you'll need servos or relays.
* **Communication Module**: To connect your Arduino to the internet, you'll need a Wi-Fi or Ethernet shield/module.
* **Power Supply**: Ensure you have a reliable power supply for your Arduino and sensors.

**2.Data Acquisition:**

Use sensors to detect traffic conditions. For instance, use ultrasonic sensors to measure vehicle presence and IR sensors to detect the movement of vehicles.

**3.Data Processing**:

The Arduino will process the data from the sensors. You might need to implement algorithms for traffic flow analysis.

**4.Communication:**

Connect your Arduino to the internet using Wi-Fi or Ethernet. You can use libraries like MQTT or HTTP to send data to a cloud platform.

**5.Cloud Platform**:

Set up a cloud platform (e.g., AWS, Azure, Google Cloud, or an IoT platform like ThingSpeak) to receive and store the data from your Arduino.

**6.Data Analysis and Decision-Making**:

Implement logic on the cloud platform to analyze the traffic data and make decisions about traffic light control, roadblocks, or other management actions.

**7.User Interface**:

Create a web-based or mobile application to provide a user interface for traffic management. Users, such as traffic operators or authorities, can monitor and control the system through this interface.

**8.Control Mechanisms:**

Use the actuators connected to the Arduino to control traffic lights or gates based on the decisions made by the cloud platform.

**9.Security:**

Ensure data encryption and implement security measures to protect your IoT traffic management system from unauthorized access.

**10.Testing and Deployment**:

Thoroughly test your system in a controlled environment before deploying it in real-world scenarios.

**11.Maintenance**:

Regularly maintain and update your system to ensure its reliability and security.

**12.Regulatory Compliance**:

Ensure that your system complies with local traffic regulations and safety standards.