Traffic Management

A Traffic Management IoT project involves the use of Internet of Things (IoT) technology to improve the efficiency, safety, and sustainability of traffic and transportation systems. Such projects can have a wide range of applications, from monitoring and controlling traffic flow to providing real-time information to drivers and city planners.

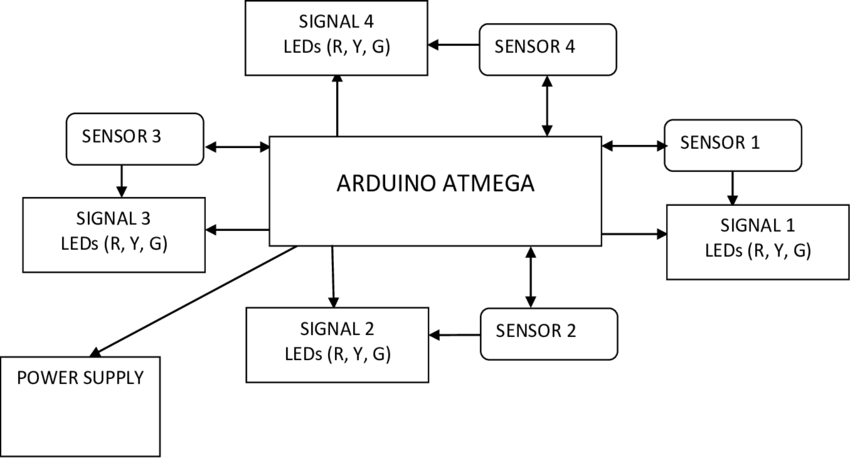
When a vehicle approaches an intersection or a busy road section, the nearby LCD display continuously updates drivers on the traffic status. If the road is congested, it advises drivers to take alternative routes or reduce their speed.

The system monitors traffic flow and can prioritize specific lanes or routes based on the data collected from the vehicle detection sensors. For example, during rush hours, the system may allocate more green light time to the main traffic direction.

In the event of an accident or roadwork, the system can display detour information on the variable message signs, guiding drivers to choose alternative routes to avoid delays.

If a road section becomes too congested, the system can also alert traffic management authorities to take action, such as deploying additional traffic control personnel or sending emergency services.

**Block Diagram of Traffic Management Project**

****

Traffic Management Project Block Diagram

**Components Required**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Components** | **Quantity** |
| 1. | Arduino or Microcontroller | 1 |
| 2. | IR Sensors | 4 |
| 3. | Ultrasonic Sensors | 4 |
| 4. | Servo Motors | 2 |
| 5. | Traffic Lights | 4 |
| 6. | Communication Modules (e.g., Wi-Fi, GSM) | 1 |
| 7. | LCD Displays | 1 |
| 8. | 15v Power Supply | 1 |
| 9. | Cables and Connectors | As Needed |

|  |  |  |
| --- | --- | --- |
| **S.no** | **Tools** | **Examples** |
| 1. | Arduino IDE | Arduino IDE |
| 2. | CAD Software (Optional) | Autodesk Eagle. |
| 3. | Version Control System | GitHub, GitHub Desktop. |
| 4. | Project Management Software | ThingSpeak |
| 5. | Simulation Software (Optional) | Wokwi |
| 6. | Data Analysis Tools | Excel. |
| 7. | Communication Tools | Microsoft Teams, Discord, Zoom |

**Tools Required**

**Testing and Safety Equipment:**

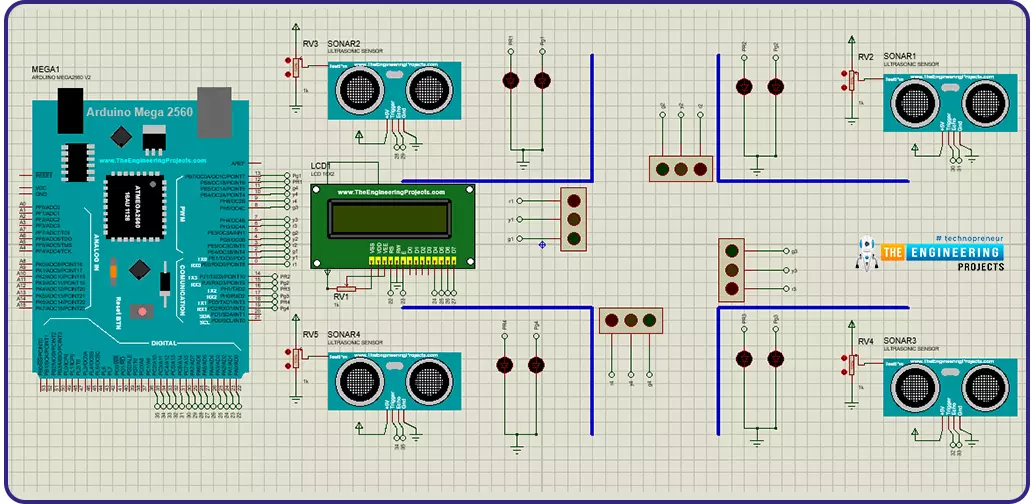
**Safety Gear**: As needed - Safety glasses, gloves, and other personal protective equipment.

**Test Vehicles**: As needed - For testing traffic flow and response under controlled conditions.

**Traffic Cones and Barriers**: As needed - For creating controlled test environments.

**Traffic Signs and Road Markings**: As needed - For creating realistic traffic scenarios during testing.

**Circuit Diagram of Traffic Management Project using Arduino, IR Sensor and UltraSonic Sensor.**

****

**Traffic Management**

Traffic Management Project Circuit Diagram

**Working Principle:**

After assembling all components according to the circuit diagram and uploading the code to the Arduino board. Now place the sensors and servo motor at accurate positions.

The IR sensors detect vehicles at intersections and entry/exit points, while ultrasonic sensors measure distances to objects. Arduino collects this data.

Arduino processes the sensor data to monitor traffic conditions, identifying congestion or obstacles. It analyzes the data in real-time.

Based on the analysis, Arduino adjusts traffic signals, variable message signs, or gate control using servo motors to optimize traffic flow.

LED displays and alerts provide real-time feedback to drivers, guiding them and enhancing road safety.

In case of accidents or emergencies, the system can send alerts to traffic management authorities or emergency services for swift action.

The project's working principle centers on data collection, analysis, and control to improve traffic flow and safety, with Arduino acting as the central controller.

**Arduino Code:**

int signal1[] = {23, 25, 27};

int signal2[] = {46, 48, 50};

int signal3[] = {13, 12, 11};

int signal4[] = {10, 9, 8};

int redDelay = 5000;

int yellowDelay = 2000;

void setup() {

for (int i = 0; i < 3; i++) {

pinMode(signal1[i], OUTPUT);

pinMode(signal2[i], OUTPUT);

pinMode(signal3[i], OUTPUT);

pinMode(signal4[i], OUTPUT);

}

// put your setup code here, to run once:

}

void loop() {

digitalWrite(signal1[2], HIGH);

digitalWrite(signal1[0], LOW);

digitalWrite(signal2[0], HIGH);

digitalWrite(signal3[0], HIGH);

digitalWrite(signal4[0], HIGH);

delay(redDelay);

// Making Green LED at signal 1 LOW and making yellow LED at signal 1 HIGH for 2 seconds

digitalWrite(signal1[1], HIGH);

digitalWrite(signal1[2], LOW);

delay(yellowDelay);

digitalWrite(signal1[1], LOW);

// Making Green LED at signal 2 and red LED's at other signal HIGH

digitalWrite(signal1[0], HIGH);

digitalWrite(signal2[2], HIGH);

digitalWrite(signal2[0], LOW);

digitalWrite(signal3[0], HIGH);

digitalWrite(signal4[0], HIGH);

delay(redDelay);

// Making Green LED at signal 2 LOW and making yellow LED at signal 2 HIGH for 2 seconds

digitalWrite(signal2[1], HIGH);

digitalWrite(signal2[2], LOW);

delay(yellowDelay);

digitalWrite(signal2[1], LOW);

// Making Green LED at signal 3 and red LED's at other signal HIGH

digitalWrite(signal1[0], HIGH);

digitalWrite(signal2[0], HIGH);

digitalWrite(signal3[2], HIGH);

digitalWrite(signal3[0], LOW);

digitalWrite(signal4[0], HIGH);

delay(redDelay);

// Making Green LED at signal 3 LOW and making yellow LED at signal 3 HIGH for 2 seconds

digitalWrite(signal3[1], HIGH);

digitalWrite(signal3[2], LOW);

delay(yellowDelay);

digitalWrite(signal3[1], LOW);

// Making Green LED at signal 4 and red LED's at other signal HIGH

digitalWrite(signal1[0], HIGH);

digitalWrite(signal2[0], HIGH);

digitalWrite(signal3[0], HIGH);

digitalWrite(signal4[2], HIGH);

digitalWrite(signal4[0], LOW);

delay(redDelay);

// Making Green LED at signal 4 LOW and making yellow LED at signal 4 HIGH for 2 seconds

digitalWrite(signal4[1], HIGH);

digitalWrite(signal4[2], LOW);

delay(yellowDelay);

digitalWrite(signal4[1], LOW);

}

The above code is for only connecting the Arduino UNO with Traffic Lights. Using Wokwi Simulator. The Further code development will be executed and submitted on the next Assignment.