SMART TRAFFIC MANAGEMENT SYSTEM-DEVELOPMENT PART1

Creating a basic Smart Traffic Management system using the Wokwi Simulator with Arduino involves simulating sensors, traffic lights, and vehicles to demonstrate how the system would work in a real-world scenario. Here's a simplified example to help you get started:

Components Needed:

1. **Arduino Board (e.g., Arduino Uno, Nano, or ESP8266):** This will serve as the core controller for your traffic management system.

2. **LEDs (Red, Yellow, and Green):** These will simulate traffic lights.

3. **Ultrasonic Sensors:** To simulate vehicle presence detectors.

4. **Servo Motor:** To control a gate or barrier that opens and closes the road.

5. **Breadboard and Jumper Wires:** To connect and prototype the components.

Steps to Create the Smart Traffic Management System:

1. **Hardware Setup:**

   - Connect the LEDs to appropriate pins on the Arduino to simulate traffic lights.

   - Connect the ultrasonic sensors to the Arduino to simulate vehicle presence detectors.

   - Connect the servo motor to the Arduino to simulate a gate or barrier.

2. **Programming the Arduino:**

   - Write Arduino code to control the behavior of the traffic lights, sensors, and gate.

   - Use ultrasonic sensors to detect the presence of vehicles and change the state of traffic lights accordingly (e.g., red light when a vehicle is detected).

   - Simulate the traffic light sequence (e.g., red, green, yellow) based on timing or sensor inputs.

   - Use the servo motor to control the gate's opening and closing based on sensor inputs.

   - Implement any additional logic, such as pedestrian crossings or timers.

Program

`cpp

```
const int mainRoadRedPin = 2;

const int mainRoadGreenPin = 3;
const int sideRoadRedPin = 4;
const int sideRoadGreenPin = 5;
const int mainRoadSensorPin = 6;
```

```
const int sideRoadSensorPin = 7;

void setup() {
  pinMode(mainRoadRedPin, OUTPUT);
  pinMode(mainRoadGreenPin, OUTPUT);
  pinMode(sideRoadRedPin, OUTPUT);
  pinMode(sideRoadGreenPin, OUTPUT);
  pinMode(mainRoadSensorPin, INPUT_PULLUP);
  pinMode(sideRoadSensorPin, INPUT_PULLUP);
}

void loop() {
  // Check if there are vehicles on the main road
  if (digitalRead(mainRoadSensorPin) == LOW) {
    // Main road has vehicles, so stop side road traffic
    digitalWrite(mainRoadRedPin, LOW);
    digitalWrite(mainRoadGreenPin, HIGH);
    digitalWrite(sideRoadRedPin, HIGH);
    digitalWrite(sideRoadGreenPin, LOW);
  } else if (digitalRead(sideRoadSensorPin) == LOW) {
    // Side road has vehicles, so stop main road traffic
    digitalWrite(mainRoadRedPin, HIGH);
    digitalWrite(mainRoadGreenPin, LOW);
    digitalWrite(sideRoadRedPin, LOW);
    digitalWrite(sideRoadGreenPin, HIGH);
  } else {
    // No vehicles, all lights are red (4-way stop)
    digitalWrite(mainRoadRedPin, LOW);
    digitalWrite(mainRoadGreenPin, HIGH);
    digitalWrite(sideRoadRedPin, LOW);
    digitalWrite(sideRoadGreenPin, HIGH);
  }
}
```

3. **Simulation on Wokwi:**

- Go to the Wokwi platform (wokwi.com) and create an account if you don't have one.

- Set up a new Arduino simulation project.

- Build the virtual circuit in the Wokwi simulator, replicating your hardware setup virtually.

- Upload your Arduino code to the virtual Arduino board.

- Test and simulate the traffic management system to see how it behaves.

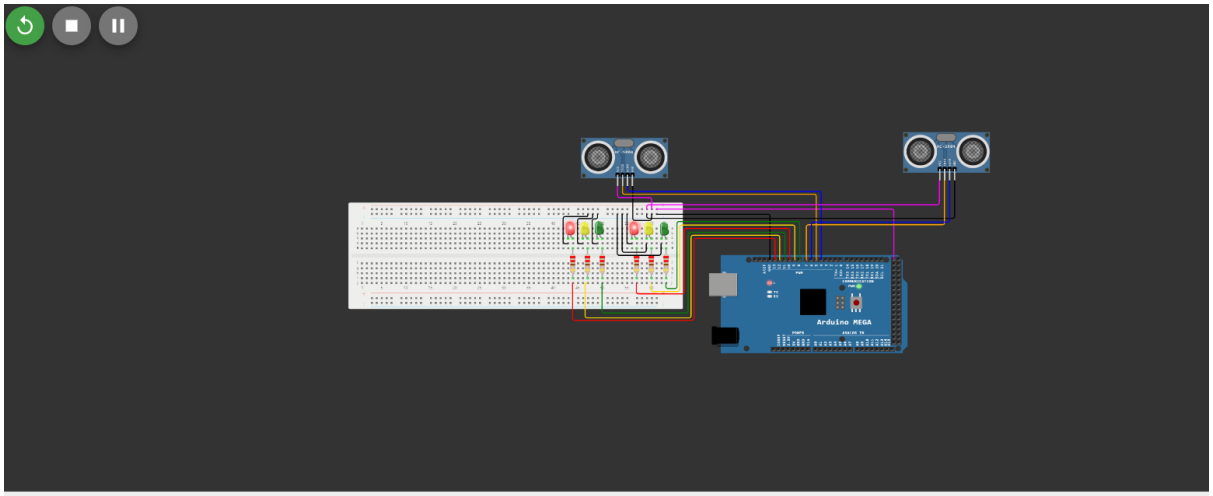
4. **Simulation Interactions:**

- In the simulation, you can interact with the virtual traffic management system to observe its behavior. For example, you can:

  - Simulate vehicles approaching the ultrasonic sensors to trigger traffic light changes.

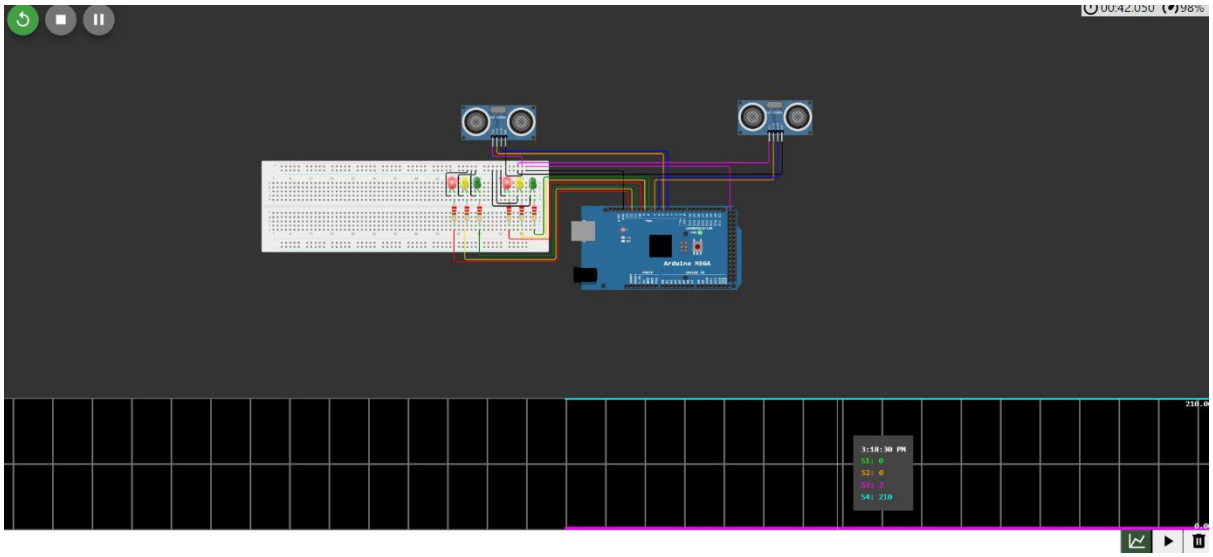
  - Toggle the state of the traffic lights.

  - Manually control the servo motor to open and close the gate.




```
1
S1: 0  S2: 0  S3: 2  S4: 21
```

a.output



b.simulation

5. **Observations and Improvements:**

- Observe how your traffic management system behaves in different scenarios.

- Make any necessary improvements or adjustments to the code and simulation to better simulate real-world conditions.

This is a basic example to get you started. In a real-world implementation, you would have more complex sensors, a more intricate traffic light control system, and integration with a central management system. Additionally, you could connect your system to a mobile application or a web interface for remote monitoring and control.