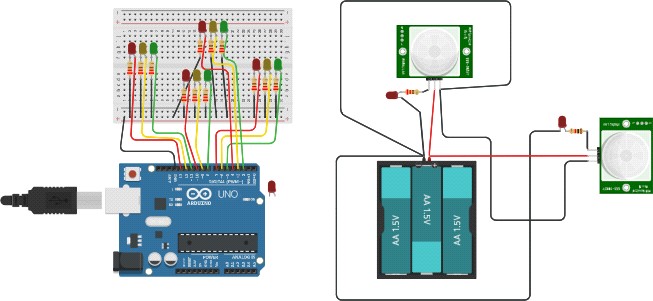
Phase 3 – development part 1

Project title - TRAFFIC MANAGEMENT SYSTEM



CODING

# First of all, we define the pins where we have connected the LEDs. red\_1 = 13

yellow\_1 = 12

green\_1 = 11

red\_2 = 10

yellow\_2 = 9

green\_2 = 8

red\_3 = 7

yellow\_3 = 6

green\_3 = 5

red\_4 = 4

yellow\_4 = 3

green\_4 = 2

def direction\_1\_green():

# green LED of direction 1 will turn ON digitalWrite(red\_1, LOW) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, HIGH) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_2\_yellow():

# yellow LED of direction 2 will turn ON digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, LOW) digitalWrite(yellow\_2, HIGH) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH)

digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_2\_green():

# green LED of direction 2 will turn ON digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, LOW) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, HIGH) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_3\_yellow(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, LOW) digitalWrite(yellow\_3, HIGH) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_3\_green(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, LOW) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, HIGH) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_4\_yellow(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, LOW) digitalWrite(yellow\_4, HIGH) digitalWrite(green\_4, LOW)

def direction\_4\_green(): digitalWrite(red\_1, HIGH)

digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, LOW) digitalWrite(yellow\_4, HIGH) digitalWrite(green\_4, LOW) import time

red\_1 = 2

yellow\_1 = 3

green\_1 = 4

red\_2 = 5

yellow\_2 = 6

green\_2 = 7

red\_3 = 8

yellow\_3 = 9

green\_3 = 10

red\_4 = 11

yellow\_4 = 12

green\_4 = 13

def direction\_1\_green(): digitalWrite(red\_1, LOW) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, HIGH) digitalWrite(red\_2, HIGH)

digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_2\_yellow(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, LOW) digitalWrite(yellow\_2, HIGH) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_2\_green(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, LOW) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, HIGH) digitalWrite(red\_3, HIGH)

digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_3\_yellow(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, LOW) digitalWrite(yellow\_3, HIGH) digitalWrite(green\_3, LOW) digitalWrite(red\_4, HIGH) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_3\_green(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, LOW) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, HIGH) digitalWrite(red\_4, HIGH)

digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, LOW)

def direction\_4\_yellow(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, LOW) digitalWrite(yellow\_4, HIGH) digitalWrite(green\_4, LOW)

def direction\_4\_green(): digitalWrite(red\_1, HIGH) digitalWrite(yellow\_1, LOW) digitalWrite(green\_1, LOW) digitalWrite(red\_2, HIGH) digitalWrite(yellow\_2, LOW) digitalWrite(green\_2, LOW) digitalWrite(red\_3, HIGH) digitalWrite(yellow\_3, LOW) digitalWrite(green\_3, LOW) digitalWrite(red\_4, LOW) digitalWrite(yellow\_4, LOW) digitalWrite(green\_4, HIGH)

def setup():

# Declaring all the LED's as output for i in range(2, 14):

pinMode(i, OUTPUT)

def loop():

# In the loop function, we controlled the signal one by one to control the flow of traffic. direction\_1\_green()

time.sleep(5) direction\_2\_yellow() time.sleep(3) direction\_2\_green() time.sleep(5) direction\_3\_yellow() time.sleep(3) direction\_3\_green() time.sleep(5) direction\_4\_yellow() time.sleep(3) direction\_4\_green() time.sleep(5) direction\_1\_yellow() time.sleep(3)

THE ASPECTS:

# 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.

# 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.

# Data Processing:

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

# 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.

# 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.

# 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.

# 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.

# Control Mechanisms:

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

# Security:

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

# Testing and Deployment:

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

# Maintenance:

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

# Regulatory Compliance:

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

# pin sensor:

three-pin sensor or a four-pin sensor. In general, ultrasonic rangefinders have one pin that connects to ground, another that connects