

IOT PHASE III

TRAFFIC MANAGEMENT SYSTEMS

PROJECT TITLE:

Building a Traffic Control Management System using Dataset Loading and Preprocessing

INTRODUCTION:

Traffic control management is a crucial aspect of ensuring efficient and safe transportation systems. With the advent of advanced technologies and data-driven approaches, it has become increasingly important to build projects that utilize datasets for effective traffic management. In this project, we will develop a traffic control management system by loading and preprocessing a dataset.

OBJECTIVE:

The objective of this project is to develop a traffic control management system that utilizes a preprocessed dataset to make informed decisions and optimize traffic flow. By analyzing the dataset, we aim to identify patterns, predict traffic conditions, and suggest optimal solutions for traffic control decisions and optimize traffic flow. By analyzing the dataset, we aim to identify patterns, predict traffic conditions, and suggest optimal solutions for traffic control.

Next, will the steps to be followed for applying our project model

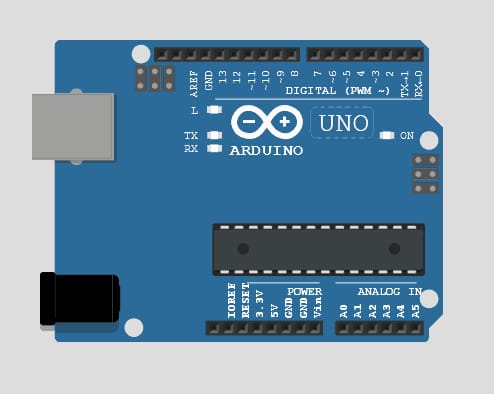
Now we start our first phase of our project development.

REQUIRED COMPONENTS:

* Arduino Uno R3
* 3 LEDS (Green, Red, Yellow)
* 1 Small Breadboard
* 1 Green Wire
* 1 Yellow Wire
* 1 Red wire
* 3 Black Wires
* Sensors
* cameras

ARDUINO UNO R3:

The LEDS has been powered by Arduino UNO (board). It contains which uploaded to the board and once it simulated LED’S start blinking like a traffic light.



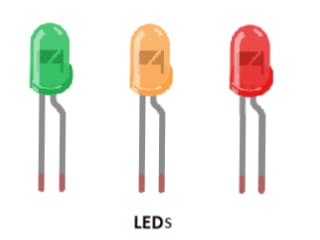
3 LED’S (GREEN RED YELLOW)

* Traffic signals consists of three colors: Red Yellow Green

RED: A Driver should stop his vehicle.

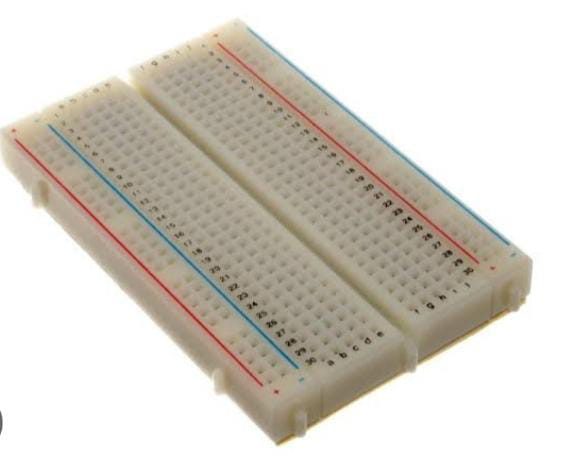
YELLOW: A Driver has to slow down and be ready to stop

GREEN: A Driver can start driving or keep driving



SMALL BREADBOARD:

* A Solderless breadboard is a matrix of electrical connection points in a plastic enclosure. They come in various sizes: The common sizes are:
* Full,60 rows,800 points including positive/negative bus strip.
* Half 30 rows, 400 points including positive/negative bus strip.



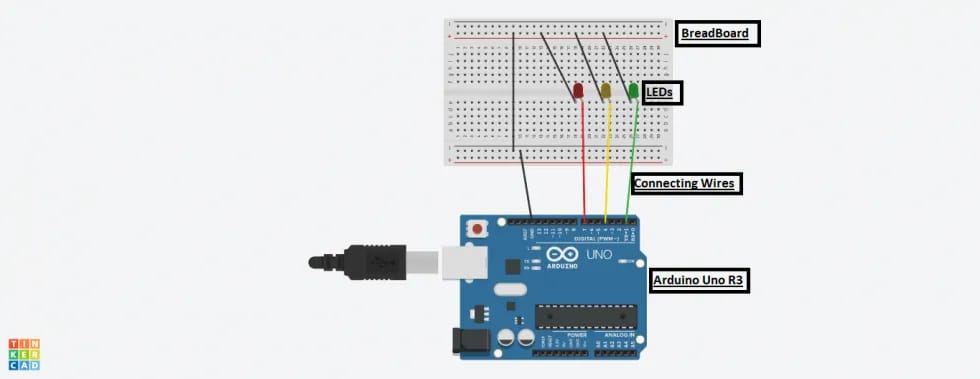
PLATFORM USED FOR EXECUTING OUR SOURCE CODE:

WowKi is a platform that offers a variety of components for building Internet of Things (IoT) projects. To create a traffic light control management system using Wowki, you will need the following components:

1. Wowki IoT Development Board: This board serves as the main controller for your project. It is equipped with a microcontroller and various input/output interfaces to connect and control other components.

2. LED Modules: These modules represent the traffic lights. You will need three LED modules, each representing the colors red, green, and yellow.

3. Jumper wires: These wires are used to establish connections between the Wowki development board and the LED modules.

CIRCUIT DIAGRAM: 

STEPS TO CREATE TRAFFIC MANAGEMENT SYSTEM IN ARDUINO:

Here's an overview of how to connect the components:

1. Connect the red LED module to one GPIO pin on the Wowki development board. This will be the pin used to control the red light.

2. Connect the green LED module to another GPIO pin on the Wowki development board. This will be the pin used to control the green light.

3. Connect the yellow LED module to a third GPIO pin on the Wowki development board. This will be the pin used to control the yellow light.

4. Ensure that each LED module is properly connected to a power supply, such as the 3.3V or 5V output pins on the Wowki development board.

5.Change the colors of the LED wires necessary to show the differentiation on different colored wires.

6. Then give black color to the wires that has been placed on the ground (GND).

7.Do any changes if you want to change the color or size or any other thing if necessary.

8. Write the necessary code to control the traffic lights based on the desired logic. This can be done using the Wowki IoT coding platform or by using Python, Arduino, or other compatible programming languages.

SOURCE CODE:

Void setup ()

{

//put your source code here to run once:

Serial.begin (115200);

if(!Serial) Serial.println (“serial is not ok”);

pinMode (3, OUTPUT);

pinMode (4, OUTPUT);

pinMode (5, OUTPUT);

}

Void loop ()

{

//put your main code here to run repeatedly:

digitalWrite (3, HIGH);

Serial.println (“green is on”);

delay (2000);

digitalWrite (3, LOW);

Serial.println(“green is off”);

digitalWrite (4, HIGH);

Serial.println (“yellow is on”);

delay (600);

digitalWrite (4, LOW);

Serial.println(“yellow is off”);

digitalWrite (5, HIGH);

Serial.println (“red is on”);

delay (1800);

digitalWrite (4, HIGH);

Serial.println(“yellow is on”);

delay(400);

digitalWrite (5, LOW);

Serial.println(“red is off”);

digitalWrite(4, LOW);

Serial.println(“yellow is off”);

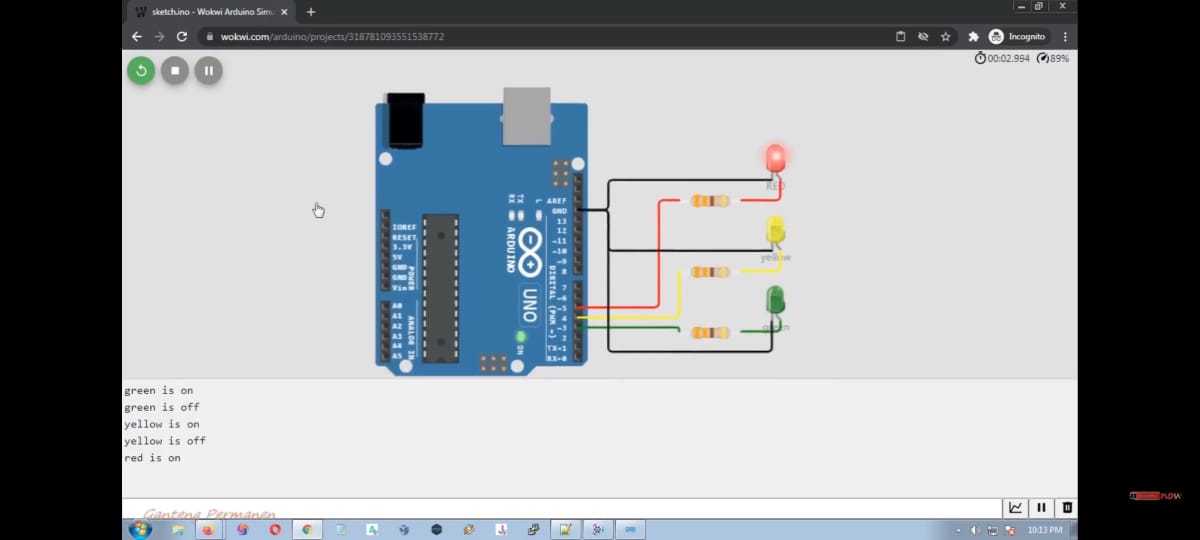
}

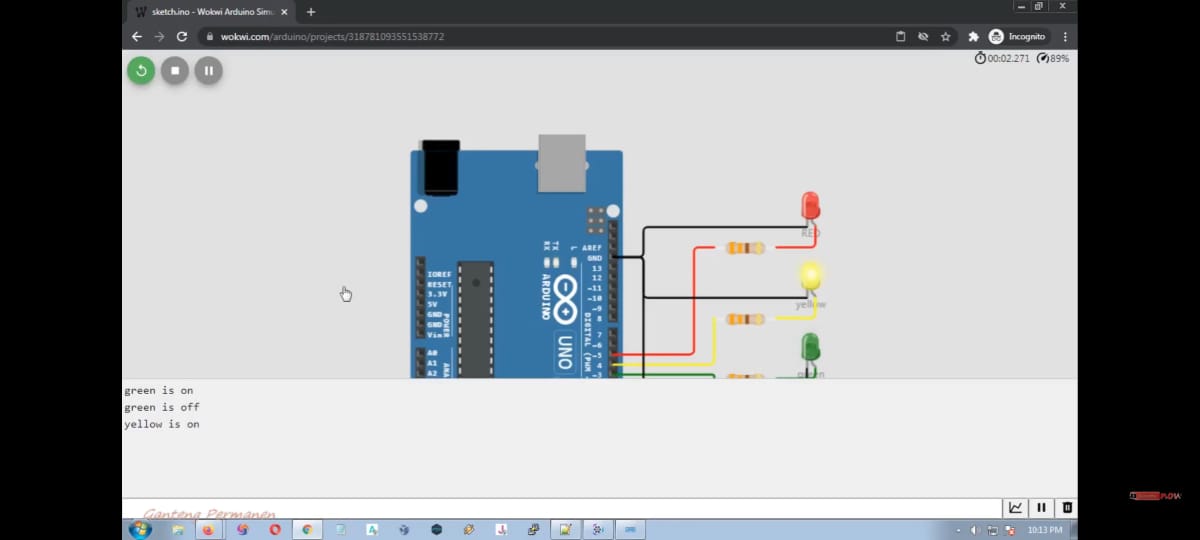
EXPLANATION OF SOURCE CODE:

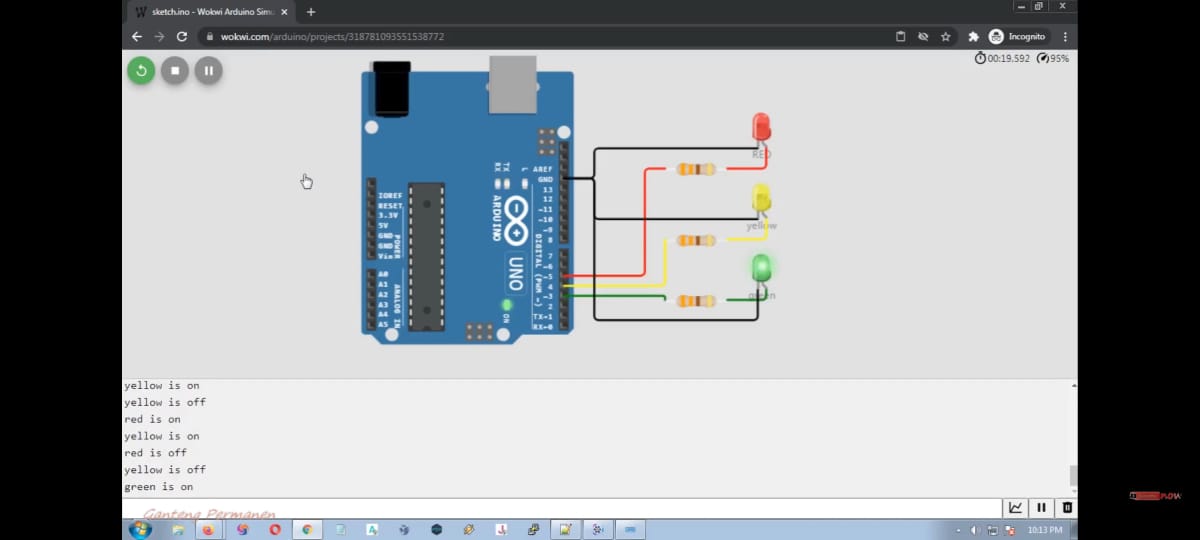
When writing the code, you will typically:

* Import the necessary libraries or modules to access the GPIO pins on the Wowki development board.
* Set up the GPIO pins as outputs to control the LED modules.
* Define the logic for transitioning between the different states of the traffic lights (e.g., red to green to yellow).
* Use appropriate delays to control the duration of each state.
* Write code to implement any additional features, such as pedestrian crossing signals or synchronization with other traffic lights.
  + In this code, we use pin numbers 2, 3, and 4 to control the red, yellow, and green LED modules respectively. The `setup( )` function is called once at the start of the program to configure those pins as output. The `loop( )` function is then executed repeatedly.
  + Within the `loop()` function, we turn on the red light for 3 seconds, followed by the yellow light for 1 second, then the green light for 5 seconds, and finally the yellow light again for 1 second. This sequence repeats continuously.
  + You can customize the timings or add additional logic based on your specific requirements. Be sure to adjust the pin numbers if you connect the LED modules to different GPIO pins on your Wowki board.
  + Remember to upload this code to your Wowki board using the Wowki IDE or any compatible programming tool.

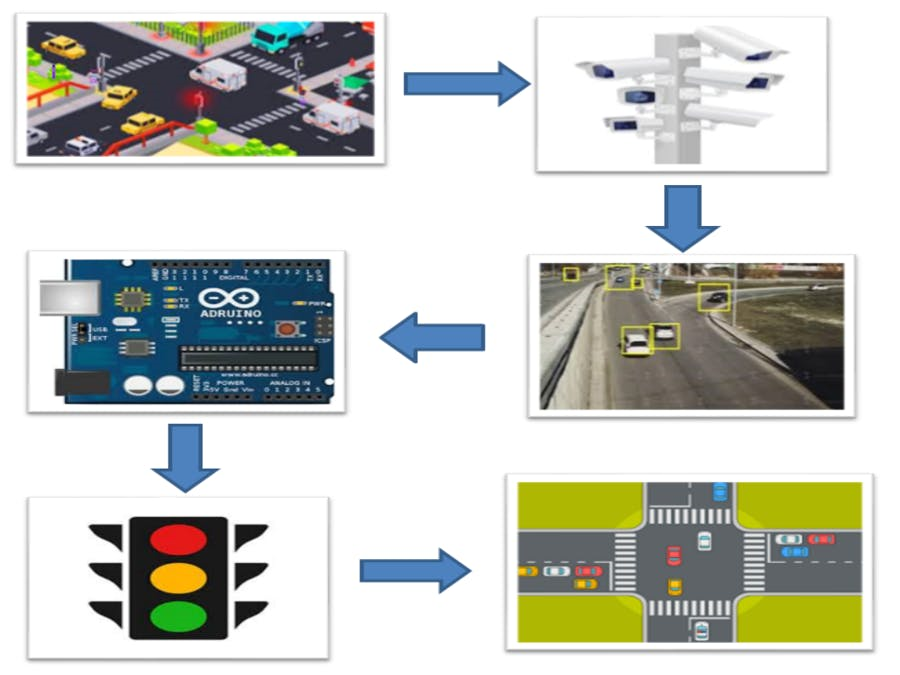
OUTPUT:







BLOCK DIAGRAM:



CONCLUSION:

By building a traffic control management system using dataset loading and preprocessing techniques, we can make informed decisions to optimize traffic flow and enhance overall road safety. This project highlights the importance of utilizing datasets and leveraging advanced technologies to improve traffic management effectiveness.