

SMART TRAFFIC MANAGEMENT SYSTEM

Phase – 4

INTRODUCTION :

Smart traffic management systems using IoT are becoming increasingly popular in many countries around the world. These systems use a variety of techniques to identify traffic congestion, such as image processing, laser tracking, and inductive loop.

However, there are significant problems with the available methods, and hence, the proposed model makes use of infrared sensors, which play a constructive role in handling traffic. Infrared sensors are used to count the density of automobiles using which the signals are controlled by ESP8266-based Node MCU and the data is sent to the central cloud system.

This system can be interfaced with existing models and takes less time to install. The proposed system allocates a smart period for green lights on roads. To overcome the existing traffic problem, a solution of profile signal control system called Autonomous and Real-Time signal control based on Estimation traffic demand for Minimization of Signal waiting time (ARTEMIS) has been proposed. ARTEMIS is a new signal-based control system that makes optimized signaling by predicting traffic flow from upstream intersections. Using the proposed system, traffic congestion can be reduced by 35%.

COMPONENTS REQUIRED :

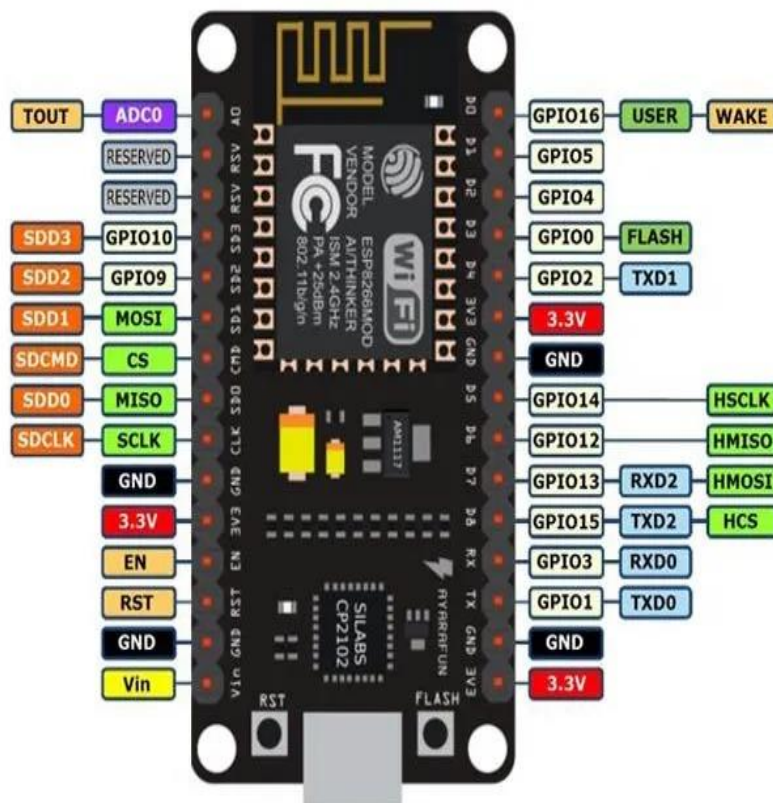
- 1.Arduino Board - ESP8266
- 2.Sensors
3. GSM/GPRS/3G/4G/5G Modules
- 4.Traffic Light
5. Central Server
6. Power supply

ESP8266 :

The ESP8266 uses a 32bit processor with 16 bit instructions. It is Harvard architecture which mostly means that instruction memory and data memory are completely separate.

The ESP8266 has on die program Read-Only Memory (ROM) which includes some library code and a first stage boot loader.

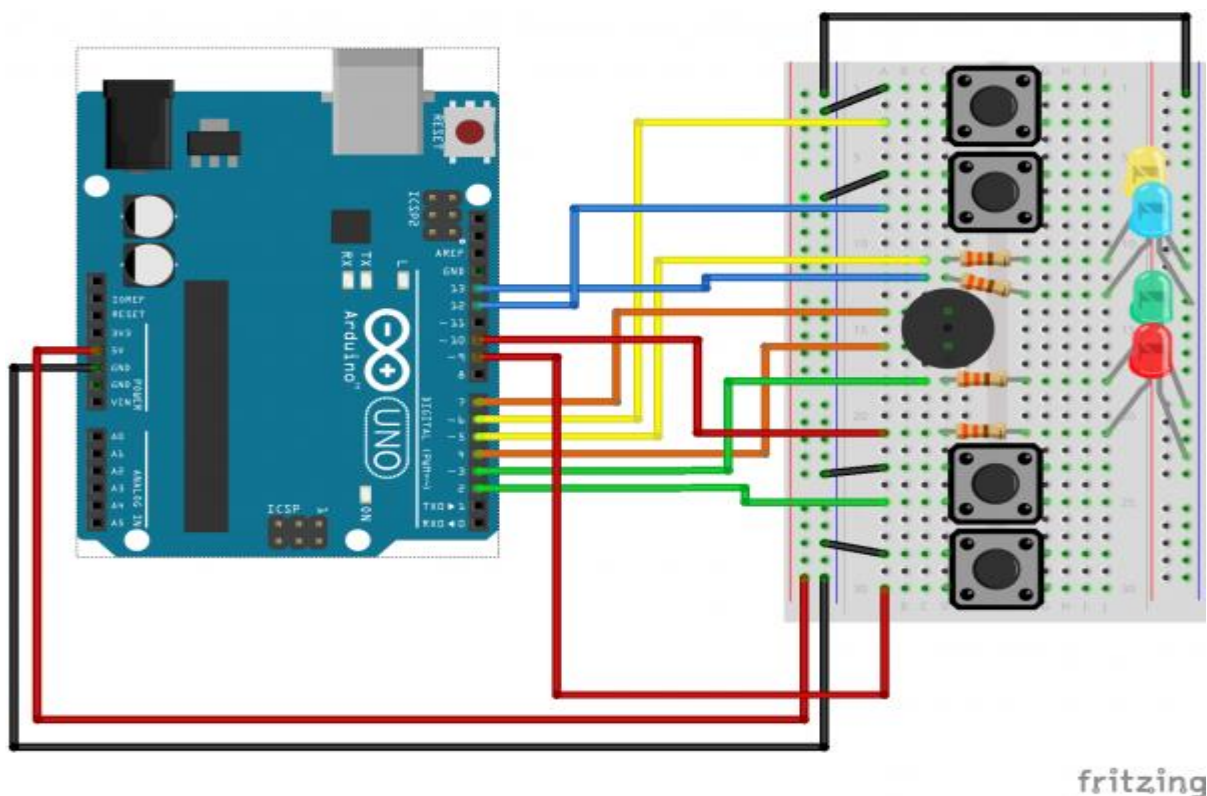
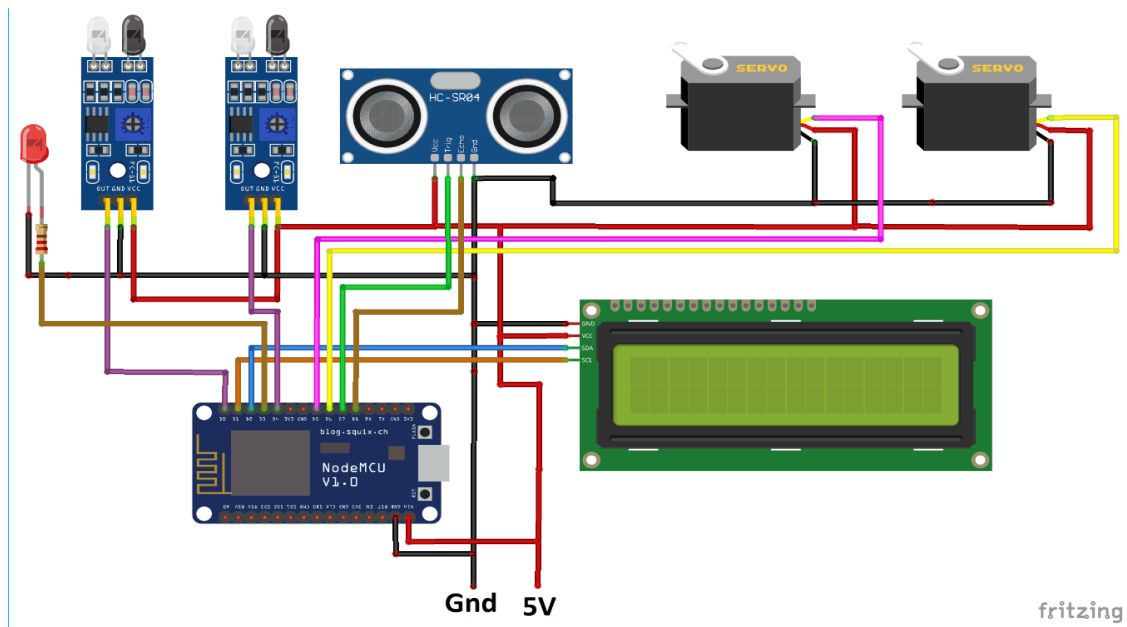
PIN—DIAGRAM :



Our ESP8266 has 17 GPIO pins but only 11 can be used (among 17 pins, 6 are used for communication with the on-board flash memory chip). It also

has an analog input (to convert a voltage level into a digital value that can be stored and processed in the ESP8266).

CIRCUIT DIAGRAM:



PROGRAM :

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char ssid[] = "WiFi Name";
char pass[] = "password";
char auth[] = "Auth Token";
int red_led = 16;
int green_led = 5;
int yellow_led = 4;

void setup(){
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);

  pinMode(red_led, OUTPUT);
  pinMode(green_led, OUTPUT);
  pinMode(yellow_led, OUTPUT);
  pinMode(red_led, HIGH);
  pinMode(green_led, HIGH);
  pinMode(yellow_led, HIGH);
}
```

```
void loop(){  
  Blynk.run();  
}
```

```
BLYNK_WRITE(V1){  
  Serial.println("Red");  
  digitalWrite(green_led, LOW);  
  digitalWrite(yellow_led, LOW);  
  digitalWrite(red_led, HIGH);  
}
```

```
BLYNK_WRITE(V2)  
{  
  Serial.println("Yellow");  
  digitalWrite(green_led, LOW);  
  digitalWrite(red_led, LOW);  
  digitalWrite(yellow_led, HIGH);  
}
```

```
BLYNK_WRITE(V3)  
{  
  Serial.println("Green");  
  digitalWrite(yellow_led, LOW);  
  digitalWrite(red_led, LOW);  
}
```

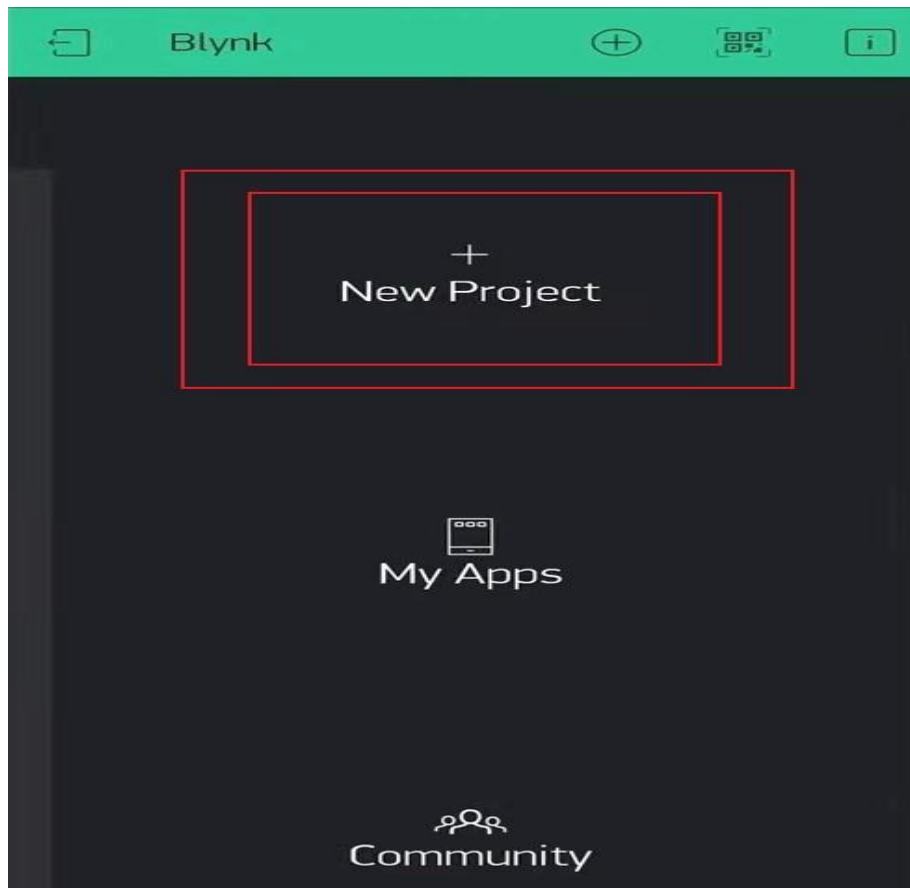
```
digitalWrite(green_led, HIGH);  
}
```

CONNECTING TO MOBILE APPLICATION (BLYNK APK) :

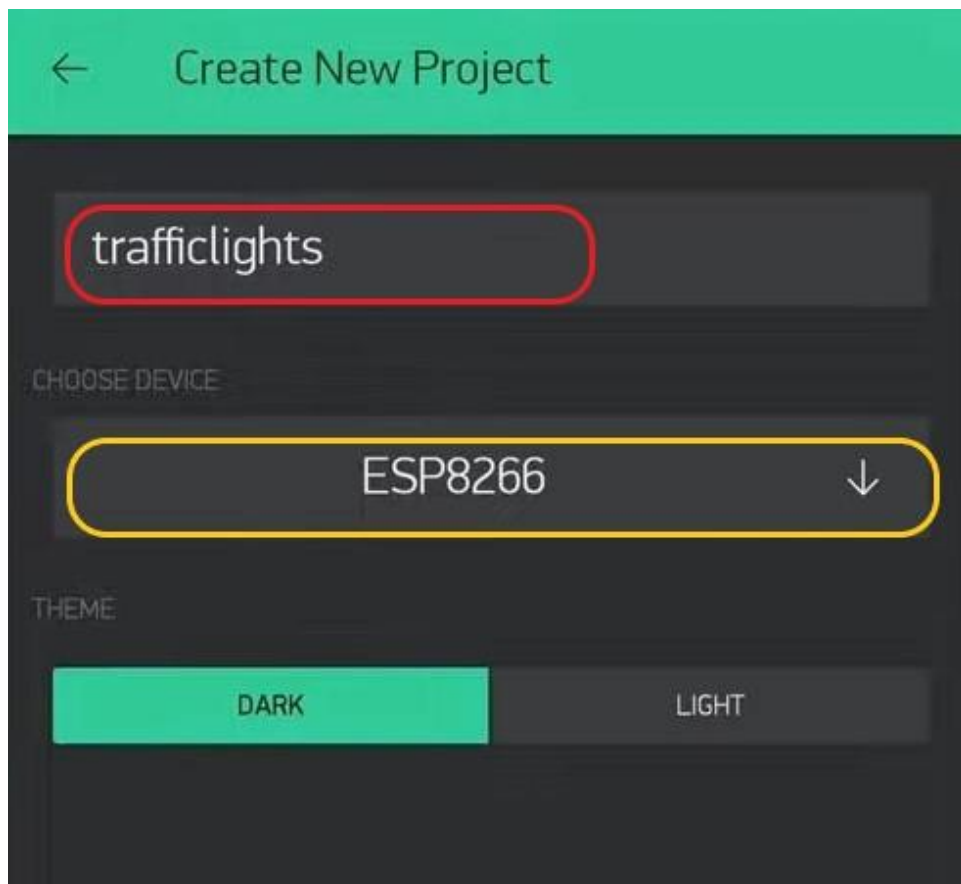
Blynk is an IoT software platform that provides infrastructure for the internet of Things. It supports 400+ development boards, SBC's, and other modules. It allows users to build and manage connected products with no code, and to remotely control, monitor, and automate them with mobile and web applications. It also offers secure cloud, data analytics, user and access management, alerts, and Over-The-Air firmware updates

1 . To create a device using Blynk, you can manually create a device using Blynk.Console for initial prototyping (works for any hardware) or use Static Tokens for cellular, Ethernet, and other non-WiFi connection methods. Activating devices with manually generated AuthTokens is recommended for prototyping stages or when you build a device for yourself.

- First we need to install the Blynk app from PlayStore and create an account.
- Click on the create button and create your new project



- Give your project a name and choose the board as NodeMCU and connection type as Wi-Fi.



← Create New Project

trafficlights

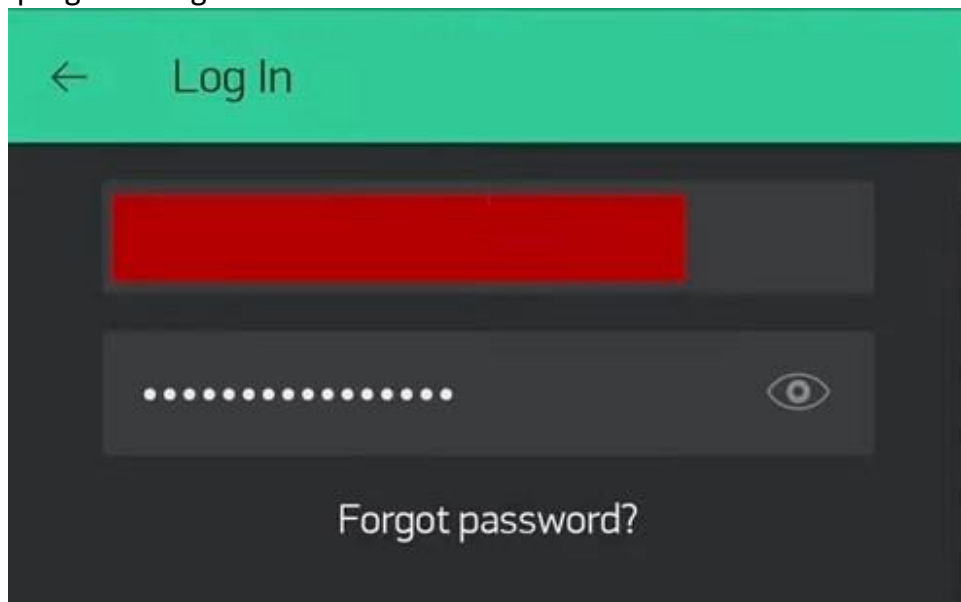
CHOOSE DEVICE

ESP8266 ↓

THEME

DARK LIGHT

- An auth token will be sent to your registered email id. Keep it safe as it will be used later on while programming.



← Log In

[Redacted Email]

..... [Eye Icon]

Forgot password?

←

My Devices

trafficlights

HARDWARE MODEL

NodeMCU

↓

CONNECTION TYPE

Wi-Fi

↓

AUTH TOKEN

*****BuiF

Refresh

Email

+

Widget Box

Displays

3.141

Value Display


i

25 °C

Labeled Value


UPGRADE

i




LED

i



Gauge


i



Radial Gauge

UPGRADE


i



Enhanced Gauge

UPGRADE


i



LCD


UPGRADE

i



SuperChart

i



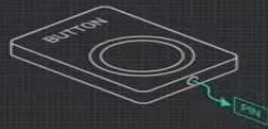
SimpleChart

UPGRADE

i



Button Settings



Button

OUTPUT

D0

0

1

MODE

PUSH



SWITCH

ON/OFF LABELS

OFF

OFF

ON

ON

DESIGN

FONT SIZE

T

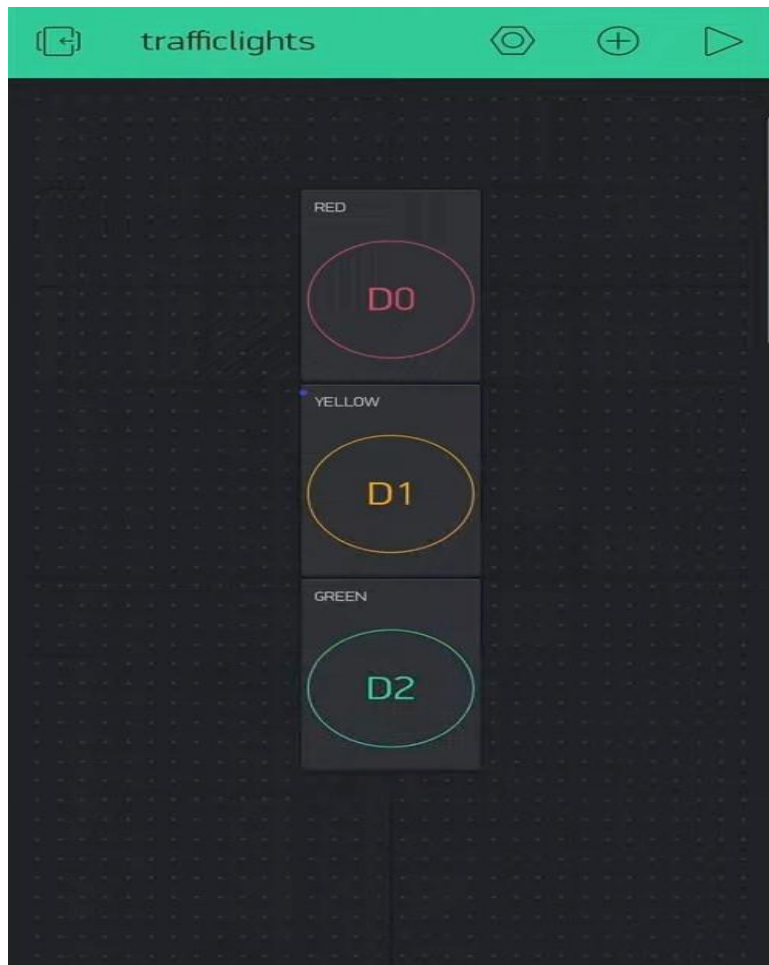
T

T

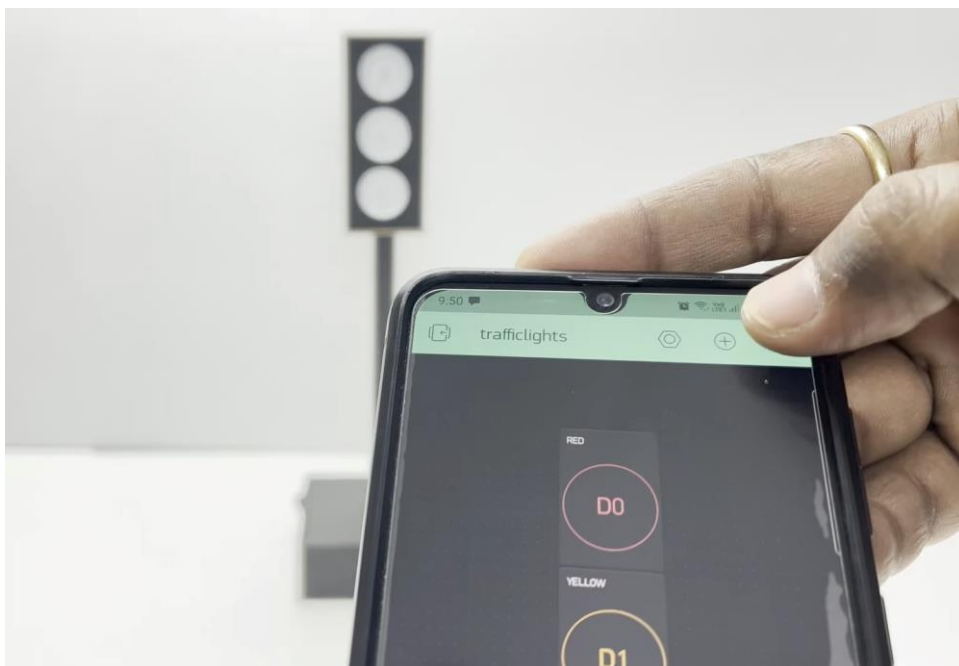
TEXT

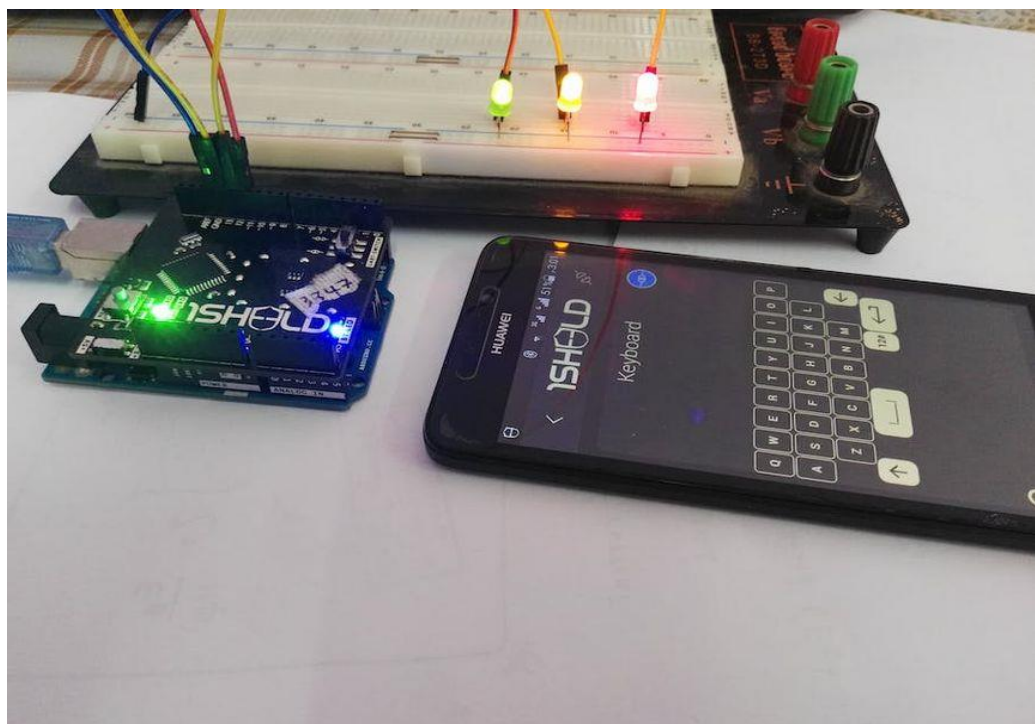
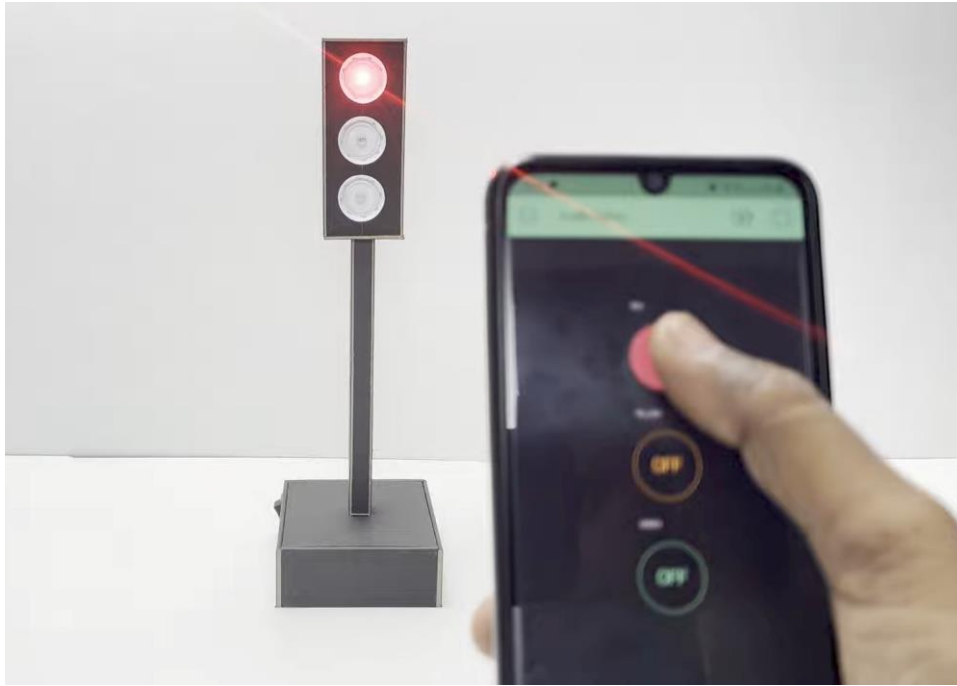


Delete



MOBILE APP OUTPUT :





HTML PROGRAM :

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title>Smart Traffic Management</title>
</head>
<body>
  <header>
    <h1>Smart Traffic Management</h1>
  </header>

  <section id="traffic-status">
    <h2>Traffic Status</h2>
    <p id="status-info">Loading...</p>
  </section>

  <section id="camera-feed">
    <h2>Live Camera Feed</h2>
    
  </section>

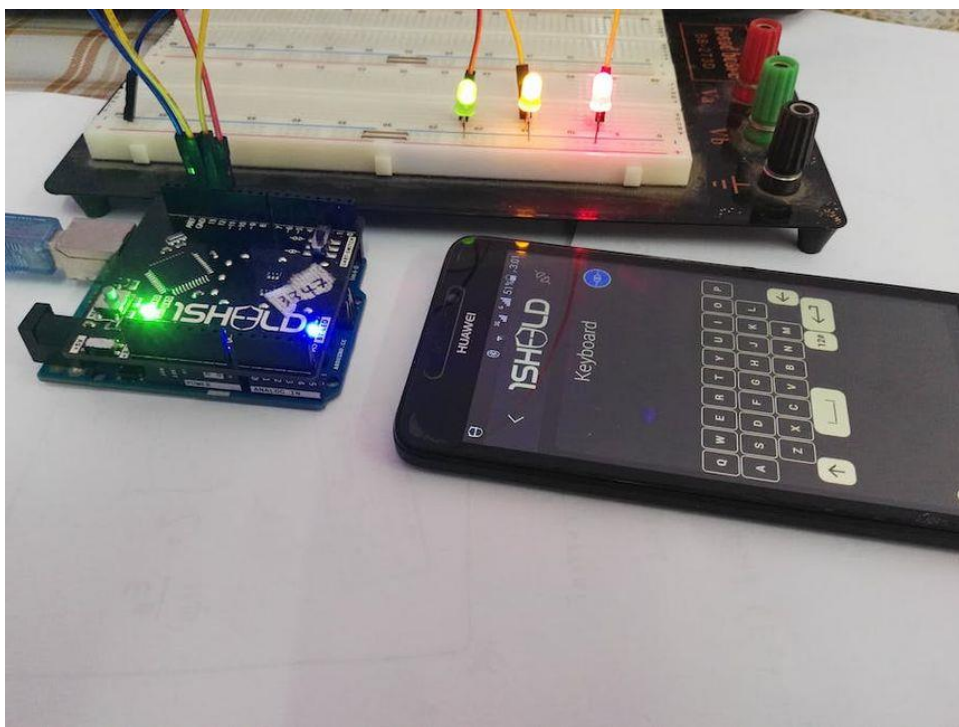
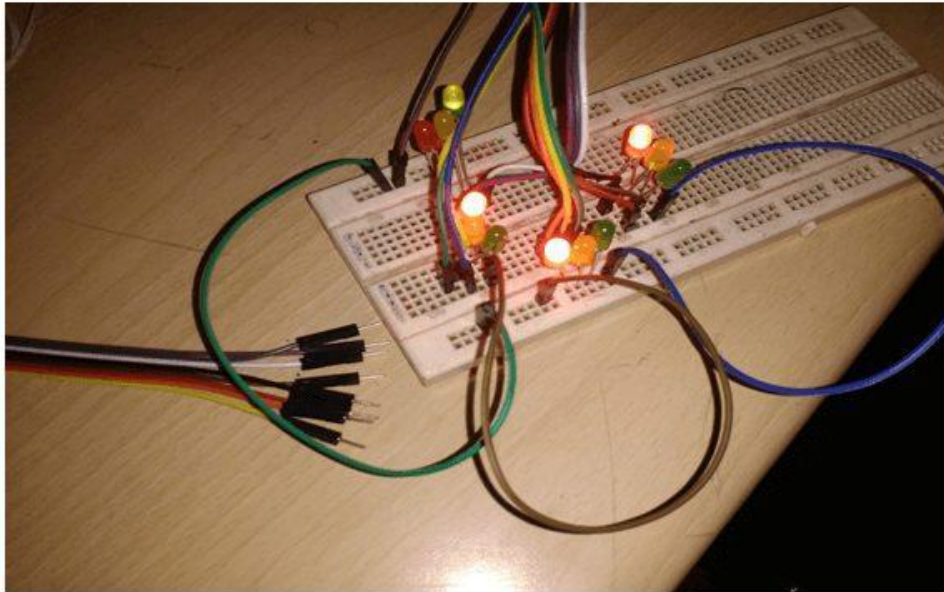
  <section id="traffic-lights-control">
    <h2>Traffic Lights Control</h2>
    <button id="green-light">Green Light</button>
    <button id="yellow-light">Yellow Light</button>
    <button id="red-light">Red Light</button>
  </section>

  <section id="sensors-data">
    <h2>Sensor Data</h2>
    <p id="temperature">Temperature: N/A</p>
    <p id="traffic-density">Traffic Density: N/A</p>
  </section>
```

```
<footer>  
  <p>&copy; 2023 Smart Traffic Management</p>  
</footer>  
</body>  
</html>
```

SAMPLE OUTPUT :





CONCLUSION :

A smart traffic management system in the context of the Internet of Things (IoT) holds great promise for addressing traffic congestion, enhancing safety, and improving overall traffic efficiency. The proposed work focuses on Smart Traffic management System using RFID which will eliminate the drawbacks of the existing system such as high implementation cost, dependency on the environmental conditions, etc. The proposed system aims at effective management of traffic congestion. It is also cost effective than the existing system. In conclusion, a smart traffic management system leveraging IoT has the potential to revolutionize urban mobility and address critical traffic-related challenges. With effective planning, collaboration between stakeholders, and addressing the associated challenges, IoTbased traffic management systems can create safer, more efficient, and sustainable urban environments.