Logo, company name

Description automatically generated

Data Engineering

**Data Acquisition Technologies and Sensor Networks Project Report on**

SMART TRAFFIC JUNCTION

**Supervisor:**

Ph.D. Fangning Hu

Dehao Zhao.

**Submitted By:**

Chidambar Prabhakar Bangre

Jitesh Balesh Kumar

Lenwyn Lobo Denis

ACKNOWLEDGEMENTS

We would like to thank our supervisors **Prof. Ph.D. Fangning Hu** and **Dehao Zhao.** for providing a chance to work on such project and for providing all the components to finish this project.

We would like to thank all my team members (**Chidambar Prabhakar Bangre**, **Jitesh Balesh Kumar,** and **Lenwyn Lobo Denis**) who all helped me in solving any problems, providing resources, and providing advice.

**Chidambar Prabhakar Bangre**

**Jitesh Balesh Kumar**

**Lenwyn Lobo Denis**

Table of Contents

[INTRODUCTION 1](#_Toc121275908)

[PROBLEM STATEMENT 1](#_Toc121275909)

[MOTIVATION 3](#_Toc121275910)

[OBJECTIVE 3](#_Toc121275911)

[OUTLINE OF THE REPORT 4](#_Toc121275912)

[LITERATURE SURVEY 4](#_Toc121275913)

[Related Works 4](#_Toc121275914)

[State of the Art In this project 7](#_Toc121275915)

[Challenges with IoT 7](#_Toc121275916)

[Privacy and Security 7](#_Toc121275917)

[Cost versus Usability 8](#_Toc121275918)

[PROPOSED METHODOLOGY 9](#_Toc121275919)

[SYSTEM REQUIREMENTS 9](#_Toc121275920)

[Python Arduino IDE 9](#_Toc121275921)

[Package Requirements 10](#_Toc121275922)

[Hardware Requirements 10](#_Toc121275923)

[DESIGN 11](#_Toc121275924)

[System Architecture 11](#_Toc121275925)

[IMPLEMENTATION 12](#_Toc121275926)

[Snapshots 12](#_Toc121275927)

[Sample Code 14](#_Toc121275928)

[RESULTS 20](#_Toc121275929)

[Test Cases 20](#_Toc121275930)

[CONSLUSION 21](#_Toc121275931)

[FUTURE WORKS 21](#_Toc121275932)

[REFERENCES 22](#_Toc121275933)

# INTRODUCTION

In today’s world as the vehicle is increasing the traffic is also increasing and emergency vehicles are the major target. Neither manual control by police officers nor using predefined timers has proved effective. In order to reduce the traffic congestion, a number of researches have already been done to provide a clear pathway to the emergency vehicles. However, it often fails to meet the target travel time of an emergency. Due to this the emergency vehicles are stuck and are unable to reach the destination on time, which leads to loss of many life.

Using IoT technology we have built smart traffic junction. The smart traffic junction will solve all traffic problem and will allow the emergency vehicle to reach to destination on time. In this system every lane in junction will have sound detecting. Sound detecting is one that detects sound wave through its intensity and convert it to electrical signal. Sound sensor works quite similar to our ears, has diaphragm which converts vibration into signal. Sound sensor consists of inbuilt microphone, peak detection and an amplifier which are very sensitive to sound. Sound waves propagate through air molecules, such sound waves cause the diaphragm in microphone to vibrate, which result in capacitance change Capacitance change is then amplified and digitalized for processing of sound intensity. When the emergency vehicle enters a particular lane the Siren sound of ambulance will be detected and that particular lane signal is turned green and other lane signal is turned red, this allows emergency vehicle to pass easily without any kind of problems. And every lane will be equipped with WIFI system where the driver in the will have the control over the traffic lights to turn the signal to green when it is required(In case of emergency).

# PROBLEM STATEMENT

Due to the ever increasing population of motor vehicles in modern developed industrialized and urban areas, traffic congestion is recognized as one of the major problems. Travelling to different places within the city is becoming more difficult, there is a loss in productivity from workers, trade opportunities are lost, delivery gets delayed and thereby the cost goes on increasing which ultimately leads to frustration and imbalanced life.

Urban traffic control is one of the most challenging problems of the day. Roads and highways are unlikely to, expand much due to cost and dwindling land supply, so intelligent systems such as advanced traffic signal control is critical for operating current roadway systems at maximum capacity. In a street network with poorly timed traffic signals, fuel consumed by vehicles stopping and idling accounts for approximately 40% of network wide vehicular fuel consumption [1]. Most of the junctions have reached a bottleneck stage. The case is evident from commuting experiences and the statistical surveys revealing the face that there are at least 34.77 lakh vehicles, 71% are two wheelers, 16% cars and 3% autos in a city like Bangalore alone [2]. Road traffic control strategies like pre-timed, progression schemes, actuated, semi actuated control, traffic response, adaptive control strategies have inherent limitations even today [3]. What is needed is an inexpensive model with less human intervention to control traffic. A perfect proportion of civilization and decentralization of control process is required. Complete central control on a network is not feasible due to reasons stated in [4].

The traditional public policy measures to relieve congestion is widening of the roads which carry heavy traffic and building of new roads. These measures are not only costly but also inefficient because of the acute shortage of space available for road construction in the over-crowded metropolitan cities. Hence there is a need to change the system rather than making new infrastructure twice. Therefore many countries are working on efficiently managing their existing transportation systems to render improved mobility and safety. By enhancing public transport, route guidance systems, traffic signal improvements and incident management, congestion can be reduced greatly.

In typical conventional traffic light controller, there are so many problems occurs which are mentioned below: Heavy traffic jams with increasing number of vehicles on the road and heavy traffic congestion problem is increased in cities. This usually happens in the morning and in the evening. Due to this, people spend unnecessary time on the road. By developing the program with different setting delay for different junctions, these problems can be solved.

Even car struck in a traffic jam at traffic light junction, a road user waiting for traffic light is a solution to this problem which detects traffic light. It can cause the emergency case become complicated.

The proposed traffic control of vehicles system solves this problem in the most effective way. When an emergency car came and number of vehicles are present in front of emergency car then system give green signal to pass the emergency and hence for it will give a particular time period to pass the signals. Market is very vast and difficult to understand. It is considered too uncertain to be predictable due to huge fluctuation of the market. Stock market prediction is interesting as well as divides researchers and academics into two groups, those who believe that we can devise mechanism to predict the market and those who believe that the market is efficient and whenever new information comes up to the market absorbs it by correcting itself, thus is no space for prediction.

Investing in a good stock but at a bad time can have disastrous result, while investing in a right time can bear profits. Financial investors of today are facing this problem of trading as do not properly understand as to which stock to buy or which stock to sell in order to get optimum result. So, the purposed project will reduce the problem with suitable accuracy faced in such real time scenario.

# MOTIVATION

Traffic congestion and transportation delays are increasing worldwide. The excellence of the emergency vehicle service depends on how fast the emergency vehicles can reach the destination on time.

If the emergency vehicle gets stuck in a traffic jam and its arrival at the destination is delayed it causes loss of lives. The record shows that more than 20 percent of people needing emergency treatment have died on their way to the hospital because of delays due to traffic jams.one of the major reason behind this delay is unawareness among people. So monitoring traffic congestion is a major step to reduce deaths. Designing this IoT system will help to solve all these problems. In this system, every lane in-junction will have sound detecting. When the emergency vehicle enters a particular lane the Siren sound of the ambulance will be detected and that particular lane signal is turned green and another lane signal is turned red, this allows the emergency vehicles to pass easily without any kind of problems.

# OBJECTIVE

The aims of this project are as follows: The primary objective of this project is to introduce smart traffic junction in IoT (Internet of Things) platform which will provide emergency vehicles to pass through the traffic junction easily and quickly. Vehicles waiting for the fixed timer to go green, this system reduces the waiting for the emergency vehicles.

# OUTLINE OF THE REPORT

Starting with the Introduction to Smart Traffic Junction we have included the Literature Survey and the challenges. The proposed methodologies are mentioned thereof. The software requirements and detailed information about the algorithm is given. The flow of the project is then followed which is represented by professional flow diagram. Implementation of offline as well as online training and recognition, assisted with code description for the ease of reader is given. The validation of project is proved with various test cases. The experimental results are shown along with the help of snapshots and graphs. The report is concluded along with discussion and future works at the end.

# LITERATURE SURVEY

## Related Works

**IOT BASED TRAFFIC SIGNAL CONTROL FOR REDUCING TIME DELAY OF AN EMERGENCY VEHICLE**

They have proposed a model that allow emergency vehicle pass through when stuck in a traffic lane. They have used IoT technology to control traffic signal. Along with this they have used ESP8266 system with an android application and a server. ESP8266 system is placed in every emergency vehicle, this will help in sending exact location of emergency vehicle to the server. Driver selects the route and which is accessed by the server. When an emergency vehicle approaches a signal, server make sure that the vehicle need not to wait at signal. A signal is also sent to a hospital so that actions can be performed much faster. To avoid confusion among people they have also placed blue light in traffic signal indicating an emergency vehicle is going to pass through.

**RFID-BASED SMART TRAFFIC CONTROL FRAMEWORK FOR EMERGENCY VEHICLES**

This paper uses RFID technology. It consists of RFID tag, RFID reader. All vehicles are equipped with RFID tag on them. RFID reader are positioned at certain distance from traffic signal. RFID reads RFID signals from the traffic lane and sends it to Arduino controller. Arduino processes these signals and necessary action will be taken. It also calculates traffic density, depending on traffic density signal gets changed. When RFID reader reads signal of an emergency vehicle the respective lane will be changed to green if it was red. A seven segment display is placed along with traffic signal. This display shows letter 'E' indicating that an emergency vehicle is passing in that lane

**SMART AMBULANCE SYSTEM USING IOT**

This paper is about an intelligent ambulance. In this system, ambulance is incorporated with sensors such as heart rate sensor, blood pressure, ECG which can judge status of parameters, this information are then sent to hospital so that quick actions can be taken. Along with this, with the help of GPRS technology traffic signals are monitored. When ambulance comes closer to the signal, it will be changed to green so that it can pass easily.

**IOT BASED INTELLIGENT TRAFFIC SIGNAL SYSTEM FOR EMERGENCY VEHICLES**

They paper speaks about traffic signal that uses IoT technology for easy movement of an emergency vehicle. This system consists of components such as Node MCU, Raspberry pi, RFID tag and its reader. Every emergency vehicle has to be fitted with RFID tag. When emergency vehicle approaches lane, RFID reader detects and identifies it. After reading information is sent to Node MCU. If the signal is red it changes to green. If signal is green extra time is added so emergency vehicles is passed through the lane. This is cost effective and also eliminates manual process which has to be done by traffic police office

**PRIORITY BASED REAL TIME SMART TRAFFIC CONTROL SYSTEM USING DYNAMIC BACKGROUND**

This system was developed to extra traffic information from image. Combination of gradient magnitude and direction subtraction method is used to calculate TD (traffic density).This paper consists of three module , first module consists of pre-processing and DBTCA. In pre-processing frame by frame streaming of video data is done. DBTCA (Dynamic background traffic cycle calculator algorithm) based on traffic density weightage time allocated of each road is calculated. Second modules RGB image is converted to HSV colour space and classification of vehicle is done. The third module work with time allocation and priority controlling. This helps to reach emergency destination on time and thus save lives of people.

**REAL TIME TRAFFIC CONTROL FOR EMERGENCY SERVICE VEHICLES**

This system was developed to reduce delay in emergency vehicles. All emergency vehicle has control system and communication and this system has unique MAC address which helps to identify the vehicle. The information collected from control system and communication including ESP8266 location is sent to control centre. This information is used to locate the emergency vehicle. The collected information from emergency vehicles is transmitted to traffic control centre is used to change traffic light. Clear route can be obtained by Changing signal of that road to green and the other signals at junction to red or yellow to let people know emergency vehicle is arriving.

**AUTOMATIC TRAFFIC SIGNALS IN SMART CITIES FOR SPEEDY CLEARANCE OF EMERGENCY VEHICLES**

Automatic Traffic Signals was presented using controller, ESP8266. This system consists of two sections transmitter and receiver. At first internet connection is checked at both sections and then ESP8266 coordinates is collected from both side and the data is send over cloud. The receiver controller on checking whether or not transmitter coordinate is near, if coordinate is near receiver controller changes the signal if not than continuously check emergency vehicles status near or not.

## State of the Art In this project

the development of a rapidly deployable and easily maintainable IoT platform is presented. The objective of this platform is to provide rapid response time of detecting emergency vehicle in the traffic junctions. Compared to the state-of-the-art systems, the platform introduced in this project has improved behavioural performance on several aspects. Firstly, the low cost of the components in the platform bring a cost-effective solution for detecting the emergency vehicles and maintain the traffic vehicles in the junction. In addition, our platform achieves fast deployment and easy maintenance, which simplifies the initial deployment and follow-up maintenance procedures.

## Challenges with IoT

There are key challenges and implications today that need to be addressed before mass adoption of IOT can occur.

### Privacy and Security

As the IoT become a key element of the Future Internet and the usage of the Internet of Things for large-scale, partially mission-critical systems creates the need to address trust and security functions adequately. New challenges identified for privacy, trust and reliability are:

• providing trust and quality-of-information in shared information models to enable re-use across many applications.

• Providing secure exchange of data between IoT devices and consumers of their information.

• Providing protection mechanisms for vulnerable devices. Table 1 shows various security & privacy requirement at different layers of IOT.

Table

Description automatically generated with medium confidence

Table 1: Various security & privacy requirement at different layers of IOT.

### Cost versus Usability

IOT uses technology to connect physical objects to the Internet. For IOT adoption to grow, the cost of components that are needed to support capabilities such as sensing, tracking and control mechanisms need to be relatively inexpensive in the coming years.

# PROPOSED METHODOLOGY

The proposed system is comprised of Arduino UNO, Sound detecting sensor, bread board, and WIFI module. Sound sensor and WIFI will be incorporated to every lane in the traffic junction, when the emergency vehicle arrives at the junction along with the siren the sound sensor will detect the siren sound and sends the signal to the Arduino saying that an emergency vehicle has arrived and then the traffic signal of that particular lane turns green and also making all the other lane signal to red so that the emergency vehicle has a clear path and can arrive the destination on time. Every driver in an emergency vehicle will have access to a web application where the driver has the control to turn the signal to green and to achieve this, we are using WIFI module to transmit the data, once the driver turns a particular lane green the remaining lanes signal will turn red to achieve the same result that is to reach the destination on time.

# SYSTEM REQUIREMENTS

## Python Arduino IDE

The Arduino IDE is a cross-platform application that is written in Java programming language. This is originated from processing open source IDE. This processing IDE used for to code interactive programs with 2D, 3D or PDF output, this is compatible with GNU Linux, MAC OS X, and Windows.

It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

## Package Requirements

To work with WIFI module (ESP 32) with Arduino IDE we first need install the board packages to the Arduino IDE. To do that first we have to go to the preferences in the IDE and then add board with URL and then enter the URL “https://raw.githubusercontent.com/espressif/arduino-esp32/ghpages/package\_esp32\_index.json”, which will download all the required packages to operate the WIFI module (ESP32).

## Hardware Requirements

* **System :** Any Windows or MacOS incorporated systems
* **Hard Disk :** 128 GB
* **Input Devices :** Keyboard, Mouse
* **Ram : 4 GB**
* **Processor :** Arduino UNO Board, WIFI module (ESP32)
* **Sensors :** Sound Detecting module, WIFI module
* **Other Modules :** LED lights, Connecting wires, Bread Board, USB to B type convertor

# DESIGN

## System Architecture

Diagram

Description automatically generated

Diagram

Description automatically generated

System Architecture

Diagram, schematic

Description automatically generated

Pinout Diagram 1

Diagram

Description automatically generated

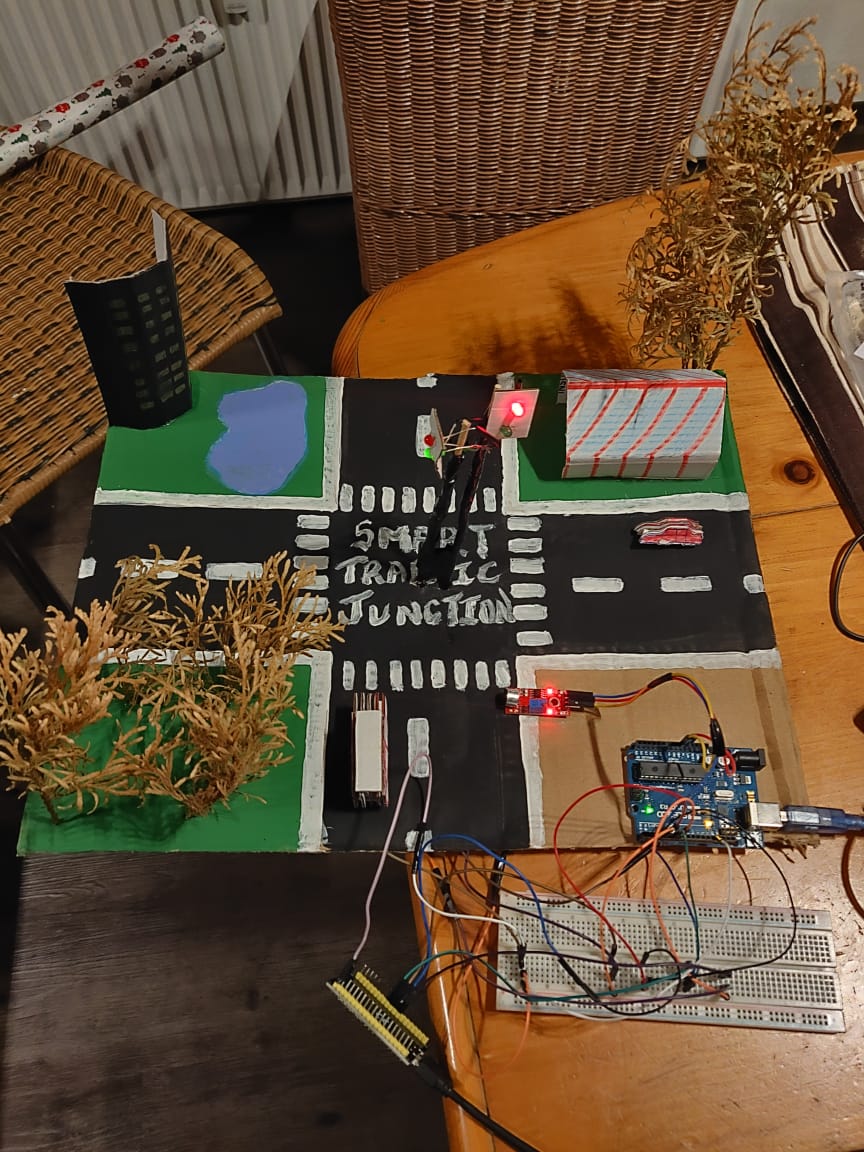
Pinout Diagram 2

# IMPLEMENTATION

There are two different models that has been implemented using different sensor modules. There is basically two parts included, the first one is hardware & second one is software. The hardware part has sensors which help to measure the real time values, Wi-Fi module gives the connection between hardware and software.

The connections for the sensors can be seen in the two pin out diagrams given (Pinout diagram 1 and Pinout diagram 2 ).

## Snapshots



Snapshot of the build model

Chart, waterfall chart

Description automatically generated

Snapshot of the web user interface for WIFI module (When the green signal if off)

Graphical user interface, application

Description automatically generated

Snapshot of the web user interface for WIFI module (When the green signal if on)

## Sample Code

#define R1 4

#define G1 5

#define R2 6

#define G2 7

#define Amb 2

int count1=0;

int count2=0;

int count3=0;

int count4=0;

void setup()

{

Serial.begin(9600);

// Serial.println("Density based Traffic Control");

delay(1000);

pinMode(Amb, INPUT);

pinMode(R1, OUTPUT);

pinMode(G1, OUTPUT);

pinMode(R2, OUTPUT);

pinMode(G2, OUTPUT);

}

void loop()

{

digitalWrite(G1, LOW);

digitalWrite(R1, HIGH);

digitalWrite(R2, LOW);

digitalWrite(G2, HIGH);

// traffic();

if(digitalRead(Amb)==0)

{

ROAD3();

}

}

// void traffic()

// {

// digitalWrite(G1, HIGH);

//

digitalWrite(R1, LOW);

// digitalWrite(R2, HIGH);

// digitalWrite(G2, LOW);

// delay(1000);

// digitalWrite(G1, LOW);

// digitalWrite(R1, HIGH);

// digitalWrite(R2, LOW);

// digitalWrite(G2, HIGH);

// }

void ALL\_RED()

{

// Serial.println("ALL RED");

digitalWrite(G1, LOW);

digitalWrite(R1, HIGH);

}

void ROAD3()

{

// Serial.println("ROAD3");

digitalWrite(R1, LOW);

digitalWrite(G1, HIGH);

digitalWrite(R2, HIGH);

digitalWrite(G2, LOW);

Serial.println("Ambulance detected");

delay(20000);

digitalWrite(R1, HIGH);

digitalWrite(G1, LOW);

digitalWrite(G2, LOW);

digitalWrite(R2, HIGH);

}

// Load Wi-Fi library

#include <WiFi.h>

// Replace with your network credentials

const char\* ssid = "nova\_1958";

const char\* password = "6jq3rABaJUeMEnJK";

// Set web server port number to 80

WiFiServer server(80);

// Variable to store the HTTP request

String header;

// Auxiliar variables to store the current output state

String output26State = "off";

String output27State = "off";

// Assign output variables to GPIO pins

const int output26 = 26;

const int output27 = 27;

const int output25 = 25;

const int output14 = 14;

// Current time

unsigned long currentTime = millis();

// Previous time

unsigned long previousTime = 0;

// Define timeout time in milliseconds (example: 2000ms = 2s)

const long timeoutTime = 2000;

void setup() {

Serial.begin(115200);

// Initialize the output variables as outputs

pinMode(output26, OUTPUT);

pinMode(output27, OUTPUT);

pinMode(output25, OUTPUT);

pinMode(output14, OUTPUT);

// Set outputs to LOW

digitalWrite(output26, LOW);

digitalWrite(output27, LOW);

digitalWrite(output25, LOW);

digitalWrite(output14, LOW);

// Connect to Wi-Fi network with SSID and password

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

// Print local IP address and start web server

Serial.println("");

Serial.println("WiFi connected.");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}

void loop(){

WiFiClient client = server.available(); // Listen for incoming clients

if (client) { // If a new client connects,

currentTime = millis();

previousTime = currentTime;

Serial.println("New Client."); // print a message out in the serial port

String currentLine = ""; // make a String to hold incoming data from the client

while (client.connected() && currentTime - previousTime <= timeoutTime) { // loop while the client's connected

currentTime = millis();

if (client.available()) { // if there's bytes to read from the client,

char c = client.read(); // read a byte, then

Serial.write(c); // print it out the serial monitor

header += c;

if (c == '\n') { // if the byte is a newline character

// if the current line is blank, you got two newline characters in a row.

// that's the end of the client HTTP request, so send a response:

if (currentLine.length() == 0) {

// HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)

// and a content-type so the client knows what's coming, then a blank line:

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("Connection: close");

client.println();

// turns the GPIOs on and off

// if (header.indexOf("GET /26/on") >= 0) {

// Serial.println("GPIO 26 on");

// output26State = "on";

// digitalWrite(output26, HIGH);

// } else if (header.indexOf("GET /26/off") >= 0) {

// Serial.println("GPIO 26 off");

// output26State = "off";

// digitalWrite(output26, LOW);

// } else if (header.indexOf("GET /27/on") >= 0) {

// Serial.println("GPIO 27 on");

// output27State = "on";

// digitalWrite(output27, HIGH);

// } else if (header.indexOf("GET /27/off") >= 0) {

// Serial.println("GPIO 27 off");

// output27State = "off";

// digitalWrite(output27, LOW);

// }

if(header.indexOf("GET /27/on")>=0)

{

Serial.println("Green light is on");

output27State = "on";

digitalWrite(output27, HIGH);

digitalWrite(output26, LOW);

digitalWrite(output25, HIGH);

digitalWrite(output14, LOW);

}

else if (header.indexOf("GET /27/off") >= 0)

{

Serial.println("Green Light is off");

output27State = "off";

digitalWrite(output27, LOW);

digitalWrite(output26, HIGH);

digitalWrite(output25, LOW);

digitalWrite(output14, HIGH);

}

// Display the HTML web page

client.println("<!DOCTYPE html><html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:,\">");

// CSS to style the on/off buttons

// Feel free to change the background-color and font-size attributes to fit your preferences

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");

client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

client.println(".button2 {background-color: #555555;}</style></head>");

// Web Page Heading

client.println("<body><h1>EMERGENCY SIGNAL CONTROL</h1>");

// Display current state, and ON/OFF buttons for GPIO 26

// client.println("<p>GPIO 26 - State " + output26State + "</p>");

// If the output26State is off, it displays the ON button

// if (output26State=="off") {

// client.println("<p><a href=\"/26/on\"><button class=\"button\">ON</button></a></p>");

// } else {

// client.println("<p><a href=\"/26/off\"><button class=\"button button2\">OFF</button></a></p>");

// }

// Display current state, and ON/OFF buttons for GPIO 27

client.println("<p>Green Light is " + output27State + "</p>");

// If the output27State is off, it displays the ON button

if (output27State=="off") {

client.println("<p><a href=\"/27/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/27/off\"><button class=\"button button2\">OFF</button></a></p>");

}

client.println("</body></html>");

// The HTTP response ends with another blank line

client.println();

// Break out of the while loop

break;

} else { // if you got a newline, then clear currentLine

currentLine = "";

}

} else if (c != '\r') { // if you got anything else but a carriage return character,

currentLine += c; // add it to the end of the currentLine

}

}

}

// Clear the header variable

header = "";

// Close the connection

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}

# RESULTS

The sound sensor successfully changed the traffic signal to green upon detecting the siren sound and made all the other lane traffic signal to red. The WIFI module was able to successfully make the traffic light green upon manually turning the signal green with the help of internet.

## Test Cases

|  |  |
| --- | --- |
| **Test Case #** | 01 |
| **Test Name** | Ambulance Siren Detection |
| **Test Description** | An emergency vehicle with siren passing through the lane equipped with sound sensor |
| **Input** | Sound |
| **Expected Output** | Turn the lanes traffic signal to green and remaining lanes traffic signal to red |
| **Actual Output** | Turned the lanes traffic signal to green and remaining lanes traffic signal to red |
| **Test Result** | Success |

|  |  |
| --- | --- |
| **Test Case #** | 02 |
| **Test Name** | Turn the traffic signal using web user interface |
| **Test Description** | The driver of the emergency vehicle has access to the web page where the driver can turn a particular lane green and all the other lanes red upon needed |
| **Input** | WIFI Signal |
| **Expected Output** | Turn the lanes traffic signal to green and remaining lanes traffic signal to red |
| **Actual Output** | Turned the lanes traffic signal to green and remaining lanes traffic signal to red |
| **Test Result** | Success |

# CONSLUSION

A traffic light system has been designed and developed with proper integration of both the hardware and the software. This interface is synchronized with the whole process of the traffic system. Automatically, this project could be programmed in any way to control the traffic light model and will be useful for planning proper road system. Smart junction which uses IoT that provides easy way for emergency vehicles to reach its destination. To avoid wastage of time spent at traffic signal we have used IoT sensors reduce this problem.

# FUTURE WORKS

On integrating this system with state and central government work flow, we can enable fast response rate from government officers thus improving the importance of life while travelling in emergency.

• Since emergency vehicle are detected by sound only, so by adding the image processing to the current project can give the best results.

• As in every junction a seven segment display can be added for displaying the EMERGENCY.

• When the traffic junction is crowded with many vehicles, signal lights will not be visible for far vehicles so by adding the warning lights along the dividers.

• Adding more high performance processors and sensors which can detect emergency vehicles early.

• An android application or web application can be built for real time tracking of an emergency vehicle.

• Multiple ways of power supply can be used like solar.

# REFERENCES

[1] Sayalee Deshmukh, and Dr. S.B.Vanjale, “IOT based Traffic Signal control for reducing time delay of an Emergency Vehicle using ESP8266,” in IEEE, 2018.

[2] Tejas Naik and Roopalakshmi R “A RFID-Based Smart Traffic Control Framework for Emergency Vehicle,” in IEEE, 2018

[3] Omkar Udawant, Nikhil Thombare,“ Smart Ambulance System using IoT,” in IEEE, 2017

[4] Shubhankar Vishwas Bhate and Prasad Vilas Kulkarni, “IoT based Intelligent Traffic Signal System for Emergency vehicles,” in IEEE, 2018.

[5] A.S.Dhatrak, and Dr.S.T.Gandhe, “Automatic Traffic Signals in Smart Cities for Speedy Clearance of Emergency Vehicles,” in IEEE, 2018.

[6] S Sarath, L.R Deepthi “Priority Based Real Time Smart Traffic Control System Using Dynamic Background ,” in IEEE, 2018.

[7] Dasari Vishal, Rishika J Reddy, Bala Abhirami M, T K Ramesh, "Real Time Traffic Control for Emergency Service Vehicles " in IEEE, 2017