ADVANCE SYSTEM TO IMPLEMENT CONGESTION CONTROL AND TRAFFIC REGULATIONS

Submitted in partial fulfillment of the
Requirements for the degree of

BACHELOR OF ENGINEERING

By

Zaid Shaikh
Saad Shaikh
Alok Yadav

Under Guidance of

Prof. Medha Somalwar

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

THADOMAL SHAHANI ENGINEERING COLLEGE

UNIVERSITY OF MUMBAI

2019-2020

CERTIFICATE

This project report entitled **Advance System to implement congestion control and traffic regulations** is approved for the degree of **Bachelor of Engineering**

External Examiner

Project guide
(prof. Medha Somalwar)

Internal Examiner

HOD(EXTC Department)

(Dr.Ashwini Kunte)

College Stamp

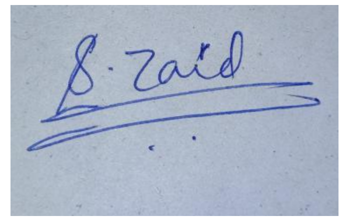
Principal

Date:

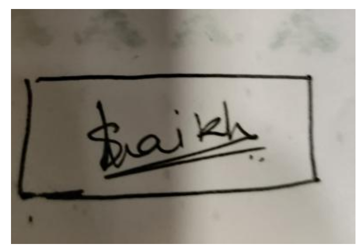
Place:

DECLARATION

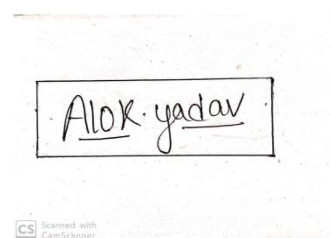
We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misinterpreted or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



ZAID SHAIKH



SAAD SHAIKH



ALOK YADAV

ACKNOWLEDGEMENT

We take this opportunity in representing the report on our project **“Advance system to implement congestion control and traffic regulations”.** The completion of any project brings with it a sense of satisfaction, but it is never complete without the appreciating those people who made it possible and whose constant support crowned each effort with success. One cannot even imagine the power of the force that guides us and neither can we succeed without acknowledging it.

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions, and inspired guidance offered by our Project Guide, **Prof. Medha Somalwar**, in bringing this report to a successful completion. We also thank her for showing keen interest and for providing all the facilities as and when we needed them.

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our esteemed Head of the Department **Dr. Ashwini Kunte** and respected Principal, **Mr. G. T. Thampi** for their revered guidance throughout our dissertation work, which made this task a pleasant job. It was a real delight to work under their guidance.

Finally we would like to thank our entire Electronics and Telecommunication Department and Staff which instilled in us the self-discipline that was necessary in building the project.

ABSTRACT

Ceaseless increase in number of vehicles is one of the major reasons for various problems in Road Traffic Control.Though the increased number can not be reduced,the existing systems can be improved.The existing systems can be modified by adding some new features as well as increasing accuracy in the existing system.One of the problems that often leads to traffic congestion is violation of signals by vehicles.There are several rules being already placed for road safety of drivers and the pedestrians.But over speeding leads to a number of accidents daily.

The modifications that the system proposes takes into account some of the above problems and also have some more ideas that can be implemented along with the solution.The system also propose a method to maintain discipline on road by making vehicles to follow the signals and lanes accordingly.The methods seems to prove efficient ways to control traffic congestion and reduce noise pollution by reducing honking.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1. Increased number of vehicles

The increase in number of vehicles also causes a situation where the vehicles exceed the capacity of a road and this often leads to traffic congestion. In metropolitan cities,this has become major problem as there is no way to increase road space. The only solution possible is to maintain traffic more efficiently. For this, the existing systems needs some more support systems and modifications.

2. Current traffic maintenance

The traffic maintenance currently is done by using two important methods one of which is to use man power that is the traffic inspectors who supervise the vehicles on road and makes sure that traffic regulations are being followed by everyone. The other major method is the use of CCTV cameras which are installed on various places on roads. The feed from these cameras are monitored by traffic personnel and makes sure that all the people who are not following rules are fined properly.

Along with these traffic signals are programmed to work on a particular algorithm that changes the signals time after a fixed period. This sometimes creates unnecessary traffic on roads.

3. Image and video processing

Image and video processing are two of the main techniques used to extract information from cameras to detect the vehicles not following rules. For Image processing, the cameras require to capture the image at a particular required minimum resolution for which sometimes the installed camera is not able to capture that image properly and thus the processing is not done. On the other hand for video processing, vehicles can be detected using some processing techniques but for which also cameras are required to be of specific minimum requirement and sometimes it is not possible to get that camera there.

Apart from processing techniques, many times a traffic personnel is given the job of checking the extracted footage and locating vehicles not following rules. This method is somewhat better than IP and VP as their is no specific requirement of cameras here but the personnel has to be attentive. Though there is some advantage but there is a chance human error here and hence this method is also not that reliable. Despite of that, this is the most widely used method of all. Even if the processing techniques are used, then too personnel are sometimes appointed with this method to ensure efficiency.

4. What is RFID and Speed sensor?

Radio Frequency Identification Reader(RFID) is sensor based on radio frequencies which detects what is known as a Tag which contains a microchip along with small antenna in it. RFID has many applications where wireless authentication is required. It basically is a reader and tags where tags are the user or given to users for detection.

Speed sensors which are already placed in cars and bikes are usually known as speedometer. A speed sensor basically counts the revolutions per minute or second of the tires of vehicles and calculates speed from that using tires specifications.

2. LITERATURE REVIEW

2.1 Existing Systems

The various systems that are already used for traffic management uses some decent technologies such as use of IP or VP. One of the other method used to identify the vehicles that violates the traffic rules is using CCTV. The cameras are installed near the junctions of roads,next to traffic signals. The footage of these cameras are monitored in a control room. In the control room, there are officials who monitor these footage closely and note details of the vehicles that violates signals and rules.

If a vehicle violates the traffic signal or a rule, the official note the details and capture images of the vehicle breaking the rule and uploads it to the official website of traffic control authority. The drivers get a notice of the fine or challan issued to them and also can check for the same on the website.

CENTRALLY CONTROLLED MONITORS[1]: The normal function of traffic lights requires more than slight control and coordination to ensure that traffic and pedestrians move as smoothly, and safely as possible. A variety of different control systems are used to accomplish this, ranging from simple clockwork mechanisms to sophisticated computerized control and coordination systems that self-adjust to minimize delay to people using the junction.

Traffic controllers use the concept of phases, which are directions of movement grouped together. For instance, a simple T-junction may have three vehicle movement phases, one for each arm of the junction. There may be additional phases for other movements such as pedestrians, cyclists, bus lanes or tramways.

A stage is a group of non-conflicting phases which move at the same time.

A traffic signal is typically controlled by a controller mounted inside a cabinet. Controllers are solid state. The cabinet typically contains a power panel, to distribute electrical power in the cabinet; a detector interface panel, to connect to loop detectors and other detectors; detector amplifiers; the controller itself; a conflict monitor unit; flash transfer relays; a police panel, to allow the police to disable the signal; and other components.

Solid state controllers are required to have an independent conflict monitor unit (CMU), which ensures fail-safe operation. The CMU monitors the outputs of the controller, and if a fault is detected, the CMU uses the flash transfer relays to put the intersection to FLASH, with all red lights flashing, rather than displaying a potentially hazardous combination of signals. The CMU is programmed with the allowable combinations of lights, and will detect if the controller gives conflicting directions a green signal for instance.

INDUCTION LOOP TRAFFIC SENSORS[2]: An induction or inductive loop is an electromagnetic communication or detection system which uses a moving magnet or an alternating current to induce an electric current in a nearby wire. Induction loops are used for transmission and reception of communication signals, or for detection of metal objects in metal detectors or vehicle presence indicators.

Vehicle detection loops, called inductive-loop traffic detectors, can detect vehicles passing or arriving at a certain point, for instance approaching a traffic light or in motorway traffic. An insulated, electrically conducting loop is installed in the pavement.

The relatively crude nature of the loop's structure means that only metal masses above a certain size are capable of triggering the relay. This is good in that the loop does not thus produce very many "false positive" triggers (say, for example, by a pedestrian crossing the loop with a pocket full of loose metal change) but it sometimes also means that bicycles, scooters, and motorcycles stopped at such intersections may never be detected by them (and therefore risk being ignored by the switch/signal). Most loops can be manually adjusted to consistently detect the presence of scooters and motorcycles at the least.

2.2 Regulations in India

The amendments to the Motor Vehicle Act were passed by the Parliament recently but most have come into effect from September 1, 2019. This means that driving errors are going to make a huge dent in your monthly budget and some of them may cost you the same or more than your monthly fuel bill.

The transport ministry issued a notification dated August 28, listing out all the laws that have come into effect from September 1, 2019. Here is a list of the important new motor vehicle laws that are coming into effect from September 1, according to the transport ministry's press release.

Enhancement of penalties: The new Motor Vehicles Act has enhanced the penalties for driving errors. According to the transport ministry's press release explaining the summary of changes in the Motor Vehicle Act, it has enhanced "penalty for offences where no penalty is specifically provided for- first offence from up to Rs 100 to Rs 500 and second/subsequent offend from up to Rs 100 to Rs 500 and second/subsequent offence from up to Rs 300 to Rs 1,500."

2.3 Proposed Solutions

An efficient algorithm for detecting traffic congestion and a framework for smart traffic control system[3]: In this paper the author proposed a real time video processing system which can measure traffic density and accordingly control the traffic signal. The video processing does object detection.

An intelligent automatic traffic light controller[4]: In this paper the author proposed a system to detect traffic density using IR sensors. Based on this the system calculates time for red and green signals.

Intelligent traffic management technique based on image processing[5]: The paper proposes a system using image processing and computer vision to calculate time for red and green signal and also adapt according to roads accordingly.

Smart traffic management system for congestion control and warnings using IoT[6]: This paper mainly suggests priority to emergency vehicles like ambulance, fire-brigade, etc. The RF transmitter and receiver is mounted on the vehicles that sends signal to receiver and the signal turns green for that particular lane so that the vehicle can pass.

Advanced Traffic Signal Control System: The system provides real time coordination between group of traffic signals using phase timings,cycle lengths and signal coordination route. It can operate on local and central level where the signals are monitored using a computer software.

These were some of the ideas or projects that were proposed and tried related to our topic and problem statement. Most of the systems used were either using some processing techniques such as IP or VP. Other than that using sensors is one of the way to implement a system which is what we aimed for during this project. Using sensors means to use IoT along with basic electronics

2.4 Technologies available in vehicles

Tesla Autopilot is an advanced driver-assistance system feature offered by Tesla that has lane centering, adaptive cruise control, self-parking, the ability to automatically change lanes, and the ability to summon the car to and from a garage or parking spot[10].

SOFTWARE UPDATES: Autopilot-enabled cars receive Autopilot software updates through servers the same as all other Tesla software updates.

ADAPTIVE CRUISE CONTROL: Autopilot has the ability to follow another car, maintaining a safe distance from it as it speeds up and slows down. It can observe a second vehicle in front of the vehicle that it is following as well as differentiate between pedestrians, bicyclists/motorcyclists, small cars, and large SUVs/trucks. It also slows on tight curves, on interstate ramps, and when a car crosses the road in front of it. It can be enabled at any speed between 18 mph and 90 mph. By default, it sets the limit at the current speed limit plus/minus any driver-specified offset, then adjusting speed according to changes in speed limits.

ALERTS: Autopilot alerts the driver under various circumstances, such as a surprising situation on the road or excessive inattention by the driver. If the driver dismisses three audio warnings within an hour, Autopilot is disabled until the car is parked. This is to prevent experienced drivers from excessive reliance on built-in safety features.

AUTOPARK/SUMMON: Auto-park drives the car into a parking spot, while Summon drives it out. Configuration settings control maximum distance, side clearance and bumper clearance. This feature activates Home-link to open and close garage doors and it is available using the fob or the Tesla mobile app.

SAFETY FEATURES: The Autopilot can detect a potential front or side collision with another vehicle, bicycle or pedestrian within a distance of 525 feet (160 m), if one is found it sounds a warning. Autopilot has automatic emergency braking that detects objects that may hit the car and applies the brakes. The car may also automatically swerve out of the way to prevent fast moving collisions

3. PROBLEM STATEMENT

Due to a continuous increase in number of vehicles, the existing or current systems are facing a difficulty to cope up with the huge increase in demand of proper traffic control. The existing systems use image processing and video processing with the help of CCTV cameras but this requires human help to complete the entire system.Sometimes due to fixed period of signal duration, unnecessary traffic is generated and this traffic is very difficult to clear as it is based on signal duration.

Having such a huge amount of technology available with us, we can improve the systems more to provide efficient traffic control. Also the traffic discipline that is made in order to make a proper flow of vehicles, must be followed by everyone. But that is not the actual scenario as there are some loop holes in the existing system.

With the help of technologies available, there is a need to make the systems better and also try to maintain traffic discipline. Along with the discipline, safety is also a major concern and requirement for traffic management and reducing over speeding can be one of the ways to achieve safety in the system.

4. PROPOSED SOLUTION

In the proposed system there will be a patch(Unit) RFID reader installed on the road. The important part here is the location of this reader. The exact location of this reader will be between the space of stop line(for vehicles) and the zebra crossing. Once the reader is installed properly, then the reader will be operated using the processor. The processor will be programmed in such a way that when the signal for that particular road goes red, the reader will be activated. And when the signal goes green, the reader will be deactivated. And if a vehicle crosses the reader unit between this time, it will be detected through the tag and its identity will be sent to the server using the processor.

The basic idea behind this feature is to prevent the vehicle from over speeding and alarm the driver whenever the speed limit is crossed. There will be a sensor installed in the vehicle that will be used to detect the speed of the vehicle. Practically every car has a built-in speedometer that detects the speed of car. The speed of car can also be detected by using this speedometer for the actual implementation. But for demonstration of how the system will work, we have used a LM393 speed sensor.The data from the sensor will be stored and monitored in the processor continuously.A threshold value for speed will be determined and the speed will be checked with that value. If the speed is less than threshold value, the process of checking is repeated. If the speed is greater than threshold value, the processor will note the speed and raise an alarm inside the vehicle.

Using ultrasonic sensors we will be measuring the density of traffic. Depending upon levels required the number of sensors can be used. Here we used three sensors to differentiate the traffic as Low, Medium and High. The ultrasonic sensors are programmed to do a sweep or check the trigger after every half cycle of respective signal. Then depending upon how many sensors detect vehicles, the density is differentiated. Here when ultrasonic sensors are detecting vehicles, it basically refers to objects in front of them.

5. FLOWCHART

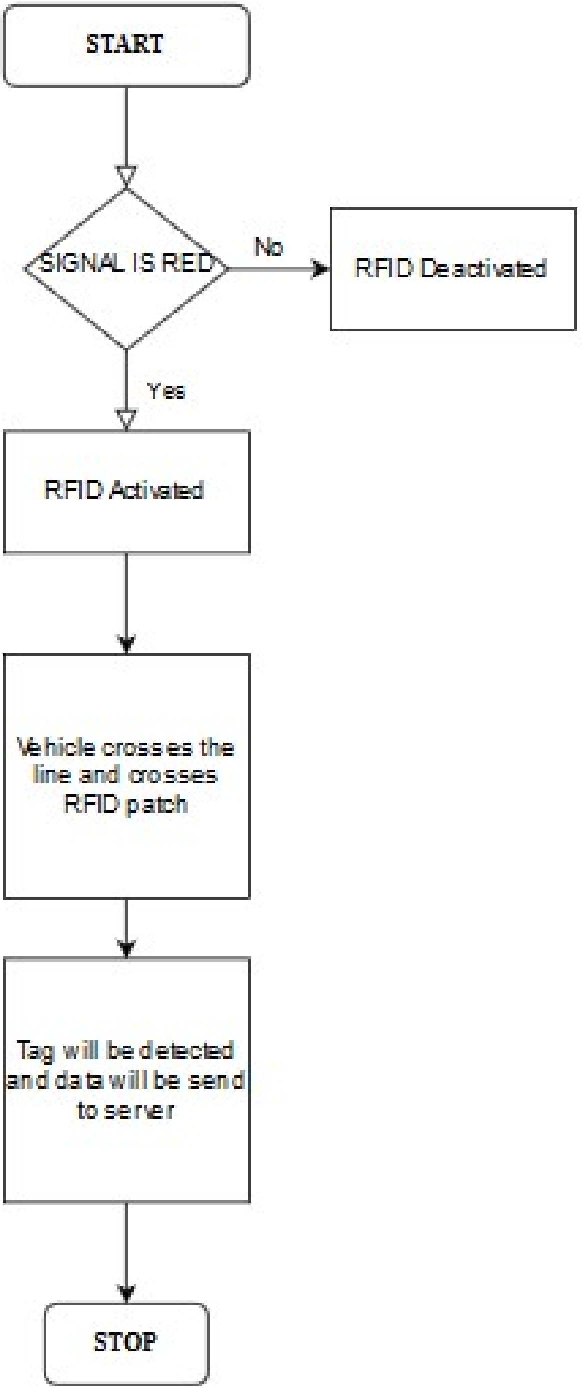


Fig 1. Part 1- RFID Detection of signal violation

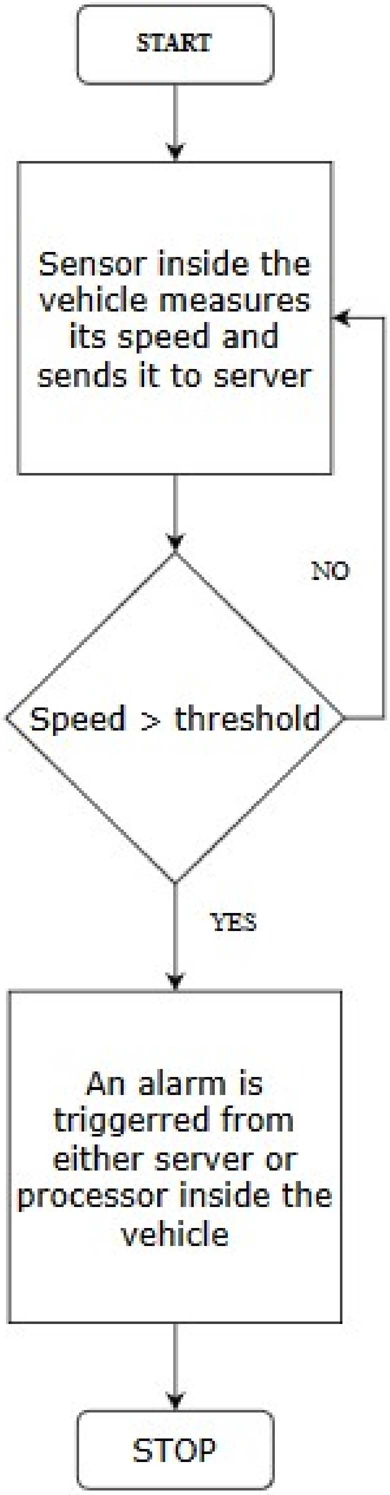


Fig 2. Part 2- Over Speeding control

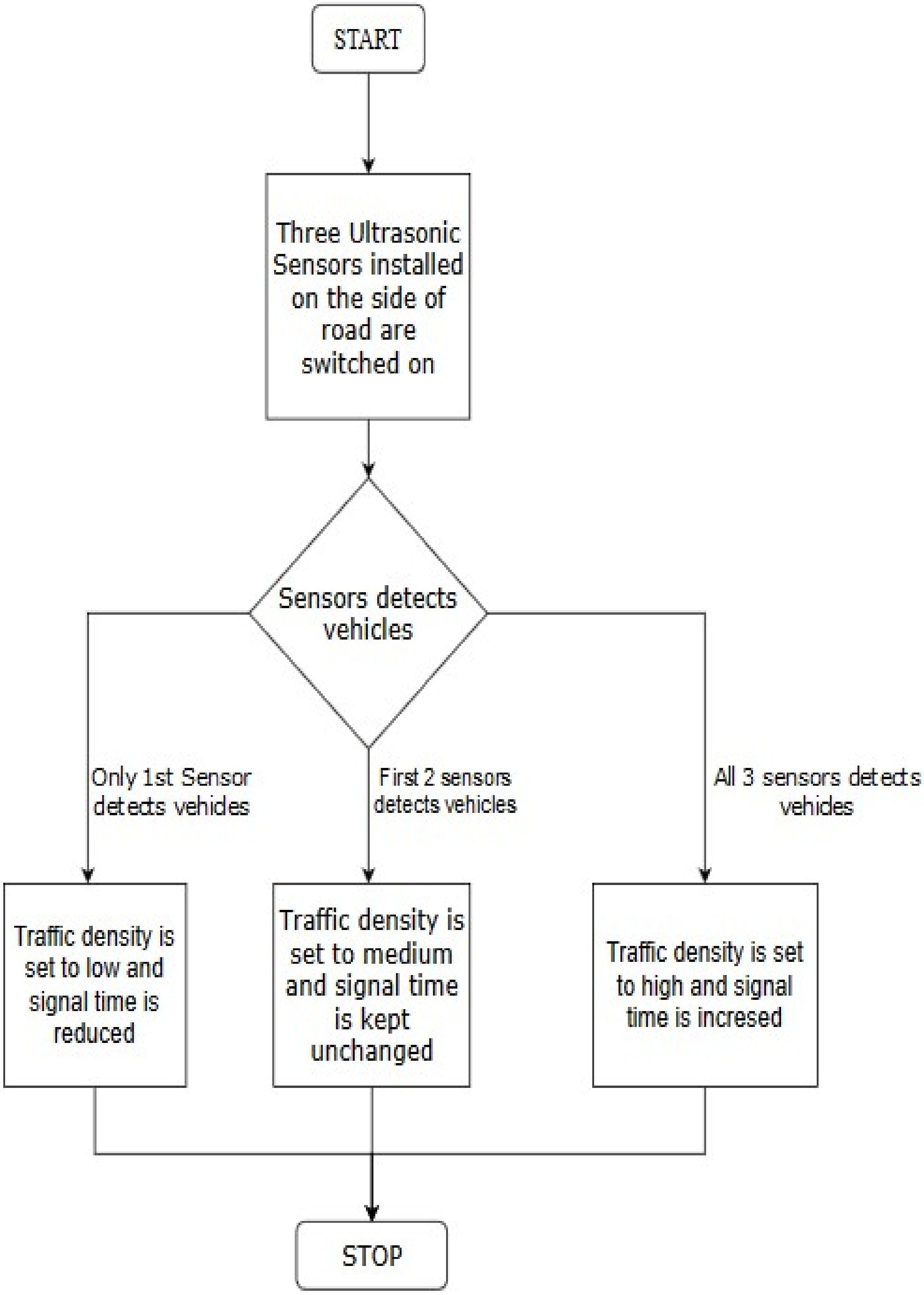


Fig 3. Part 3- Signal control based on traffic density

The above flowcharts explains briefly the working of different parts of the projects. The charts also states the basic algorithm of parts which is explained in detail ahead.

6. BLOCK DIAGRAM

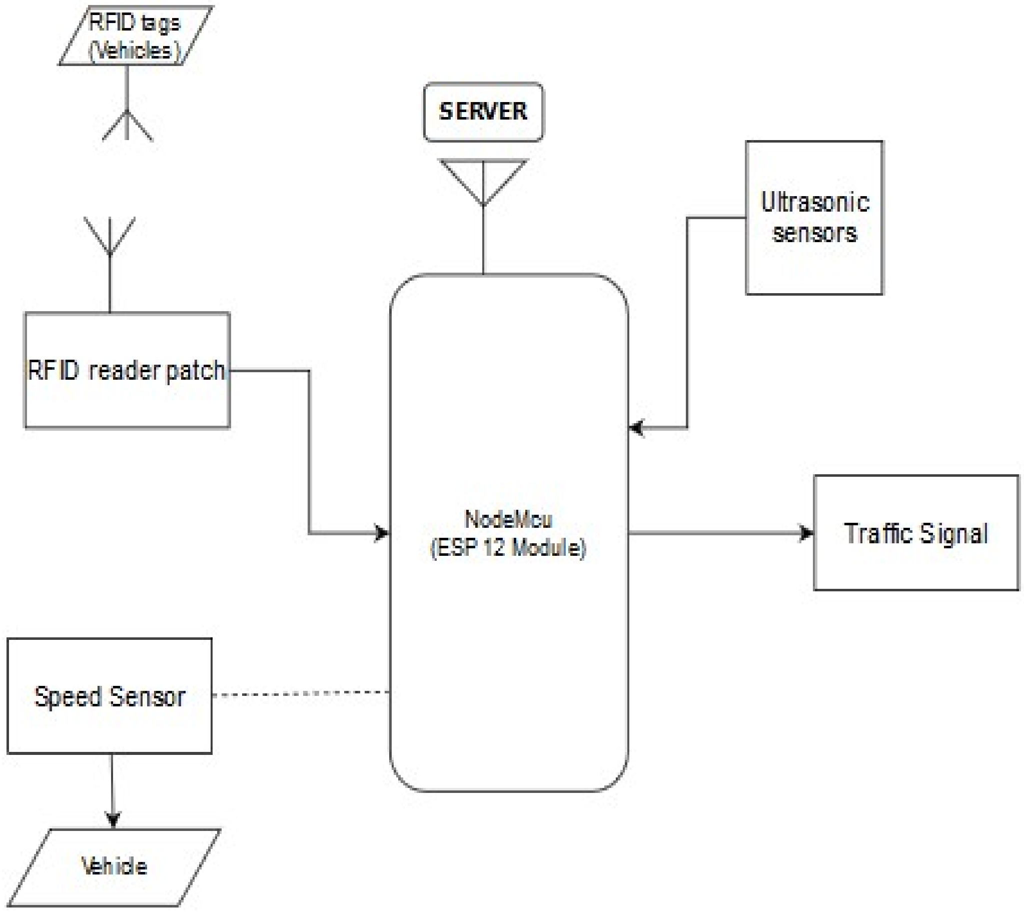


Fig 4. Block diagram of project

The block diagram explains the overall structure of hardware connections and how components are inter related. Here RFID reader will be connected to the processor which is NodeMcu and the tags which are actually vehicles will be read by the reader and values passed by NodeMcu to the server will be stored on the server.For the speed control, the sensor will be installed in the vehicle which will be connected to a processor that will be help in monitoring the speed via server.

For the traffic density measurement, Ultrasonic sensors will be installed on the side of roads which are then connected to the processor. The processor will then differentiate the traffic and change the signal duration.

7. COMPONENTS

7.1 NODEMCU

The Node MCU is an open source firmware and development kit that helps you to prototype your IoT product with Arduino IDE or in few Lua script lines.
It includes firmware which runs on the ESP8266 Wi-Fi SoC. And hardware which is based on the ESP-12 module.The on chip Wi-Fi module helps to connect to server or networks without any issue of voltage fluctuations. The programming can be done using C language on the Arduino IDE.

Fig 5. NodeMcu

7.2 RFID READER AND TAG

Fig 6. RFID reader and tag

A radio frequency identification reader (RFID reader) is a device used to gather information from an RFID tag, which is used to track individual objects. Radio waves are used to transfer data from the tag to a reader.
RFID is a technology similar in theory to bar codes. However, the RFID tag does not have to be scanned directly, nor does it require line-of-sight to a reader. The RFID tag it must be within the range of an RFID reader, which ranges from 3 to 300 feet, in order to be read. RFID technology allows several items to be quickly scanned and enables fast identification of a particular product, even when it is surrounded by several other items.

The RFID reader emits radio waves of specific frequencies through RFID antennas. The waves "give energy" to the tags so that they can communicate by emitting a unique ID. They do not need batteries and can be used for many years. The reader processes the data so that we can integrate them into our application and give them meaning. The typical reading range is 0-12 meters. Gen2 UHF RFID systems consist of: readers, antennas, printers, and RFID tags or tags.

RFID tag in its most simplistic form, is comprised of two parts – an antenna for transmitting and receiving signals, and an RFID chip (or integrated circuit, IC) which stores the tag's ID and other information. RFID tags are affixed to items in order to track them using an RFID reader and antenna. RFID tags transmit data about an item through radio waves to the antenna/reader combination. RFID tags typically do not have a battery (unless specified as Active or BAP tags); instead, they receive energy from the radio waves generated by the reader. When the tag receives the transmission from the reader/antenna, the energy runs through the internal antenna to the tag's chip. The energy activates the chip, which modulates the energy with the desired information, and then transmits a signal back toward the antenna/reader.

For the demonstration purpose the RFID module used is RC5222,which is an integrated reader and tag chip that operates on 13.56Mhz.The range of this module is low compared to the required one.

7.3 LED

A Light-Emitting-Diode is a semiconductor light source that lights up when current floes through it. The source used here is standard 3Volt LED's used for basic purposes such as indication,presence,signals

Fig 7. LED

7.4 LM393 SPEED SENSOR

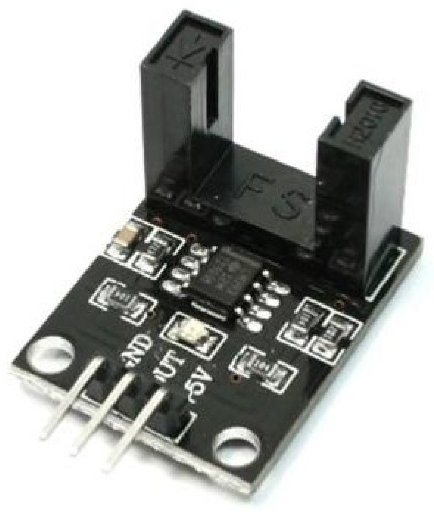


Fig 8. Speed Sensor

The LM393 Speed Sensor Module is basically an Infrared Light Sensor integrated with LM393 Voltage Comparator IC. The sensor can also be divided into two parts: the sensor part and the control part.The sensor part of the LM393 Speed Sensor module consists of an Infrared LED and an NPN Photo Transistor. These two components are placed facing each other is a special housing made of black thermoplastic.

This special housing ensures that the Photo Transistor receives light only from the Infrared LED and all the external source of light is eliminated.Coming to control unit, it is made up of LM393 Voltage Comparator and a few passive electronic components. The signal from the photo transistor is given to the LM393 and based on the presence or absence of an object between the Infrared LED and the Photo Transistor, the Output of the LM393 IC will either be HIGH or LOW

7.5 ULTRASONIC SENSOR

Fig 9. Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound. Ultrasonic sensors have two main components: the transmitter and the receiver.In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver[9].

Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. In comparison to infrared (IR) sensors in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles

7.6 Thingsboard

Thingsboard is an open source IoT platform for data collection, processing and visualization and device management[7]. The platform can be used for various IoT projects data has to be collected on a server or displayed on a server. Also remote control of devices using server can be done using this platform.

The platform provides features as provision, monitor and control your IoT entities in secure way using rich server-side APIs. Define relations between your devices, assets, customers or any other entities.

Fig 10. Thingsboard logo

7.7 Arduino

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board micro-controllers and micro-controller kits for building digital devices[8].

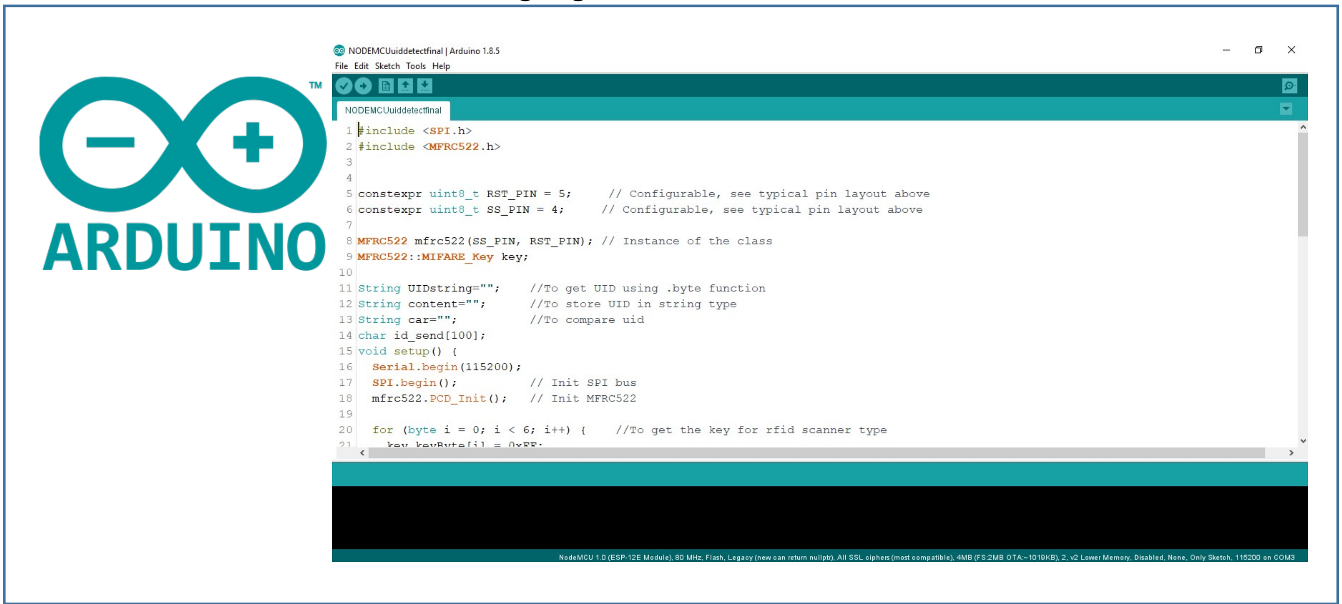


Fig 11. Arduino IDE

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based micro-controllers: AVR Studio (older) and Atmel Studio (newer).The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. 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. The source code for the IDE is released under the GNU General Public License, version 2.

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 tool chain, 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.

8. PROGRAMMING LOGIC

8.1 RFID Detection of signal violation

The NodeMcu is programmed for detecting the tags when the signal goes red for that specific road or lane. The cricuit diagram for connecting RFID to NodeMCu is given below:

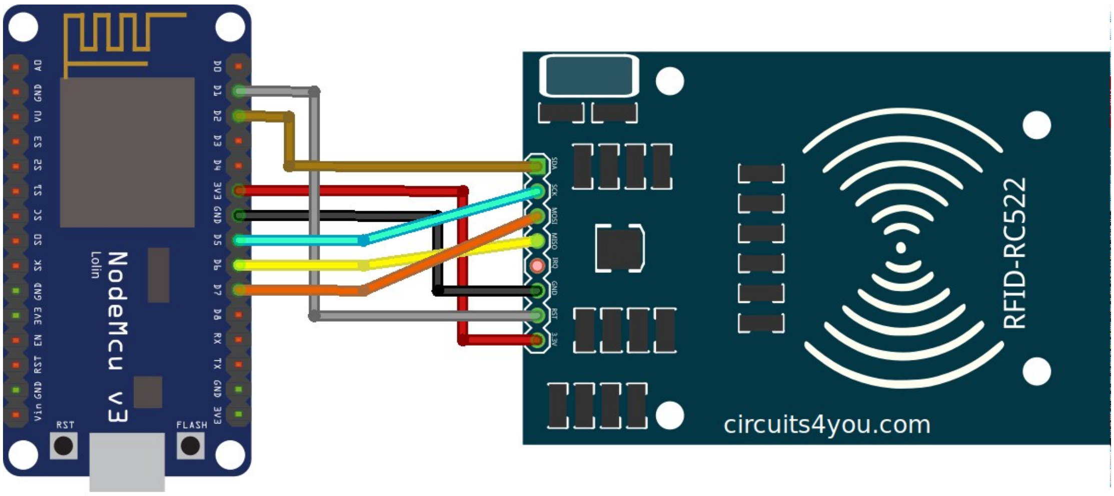


Fig 11. RFID and NodeMcu connection

By attaching led to above we can make the signal and use the following code that will be uploaded to the processor using arduino IDE:

```
#include <ArduinoJson.h>

#include "ThingsBoard.h"

#include <ESP8266WiFi.h>

#include <SPI.h>

#include <MFRC522.h>


//---------- WIFI and Device Configuration----------//

#define WIFI_AP                 "realme X"

#define WIFI_PASSWORD           "whateverittakes"

#define TOKEN                   "KDhtyDyIDFQ4t3iDZXtU"
```

```
#define THINGSBOARD_SERVER   "demo.thingsboard.io"

//**************************************************
WiFiClient espClient;       //Initialize ThingsBoard client
ThingsBoard tb(espClient);        // Initialize ThingsBoard instance
int status = WL_IDLE_STATUS;      // the Wifi radio's status
#define SERIAL_DEBUG_BAUD      115200      //--- Baud rate for debug serial

//------------RFID pins and class configuration---------/
constexpr uint8_t RST_PIN = 5;        // Configurable, see typical pin layout above
constexpr uint8_t SS_PIN = 4;        // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class
MFRC522::MIFARE_Key key;

String UIDstring="";      //To get UID using .byte function
String content="";        //To store UID in string type
String car="";            //To compare uid
char akash[100];

void setup() {

  Serial.begin(SERIAL_DEBUG_BAUD);          // initialize serial for debugging
  SPI.begin();             // Init SPI bus
  mfrc522.PCD_Init();    // Init MFRC522
```

```
  for (byte i = 0; i < 6; i++)
  {
    key.keyByte[i] = 0xFF;              //To get the key for rfid scanner
type
  }
  WiFi.begin(WIFI_AP, WIFI_PASSWORD);    // Initialising WiFi
  InitWiFi();
}

void loop() {
  delay(1000);

  if (WiFi.status() != WL_CONNECTED) {
    reconnect();
  }

  if (!tb.connected()) {                          //TO connect to thingsboard
server
    // Connect to the ThingsBoard
    Serial.print("Connecting to: ");
    Serial.print(THINGSBOARD_SERVER);
    Serial.print(" with token ");
    Serial.println(TOKEN);
    if (!tb.connect(THINGSBOARD_SERVER, TOKEN)) {
      Serial.println("Failed to connect");
      return;
    }
  }
```

```
  Serial.println("Sending data...");

  if ( ! mfrc522.PICC_IsNewCardPresent())     //------- Look for new cards---------//
    return;

  if ( ! mfrc522.PICC_ReadCardSerial())    //------- Verify if the NUID has been readed--------//
    return;

  for (byte i = 0; i < mfrc522.uid.size; i++) {       //------ Store UID into UIDstring and content------//
    UIDstring = UIDstring + String(mfrc522.uid.uidByte[i]);
    //Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "); //Conditional operator - If the byte is less than hexadecimal 16 then
    //Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX)); // These two lines used for storing UID
  }

  content.toUpperCase();
  /*if(content != car){                          //compares old and new uid to make sure an ID is not repeated twice in one cycle
    Serial.println("A CAR BROKE SIGNAL!!!");
    Serial.print("CAR ID:");
    Serial.println(content);
    //uid=content.toInt();
```

```
    //Serial.print("uid> ");
    //Serial.println(uid);
  }
  car=content;
  delay(1000);
  content="";
*/
  content.toCharArray(akash,12);
  Serial.println(akash);
  tb.sendTelemetryString("UID",  akash);        //Send  value  to
thingsboard server

  //tb.loop();
}

void InitWiFi()
{
  Serial.println("Connecting to AP ...");
  // attempt to connect to WiFi network

  WiFi.begin(WIFI_AP, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to AP");
}
```

```
void reconnect() {
  // Loop until we're reconnected
  status = WiFi.status();
  if ( status != WL_CONNECTED) {
    WiFi.begin(WIFI_AP, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("Connected to AP");
  }
}
```

By using the above program and logic we get the data of vehicles displayed on the thingsboard server as a table with details of vehicles in the form ID of tags. The output is obtained using a dashboard with widget of card that displays the incoming data in the form of a table.

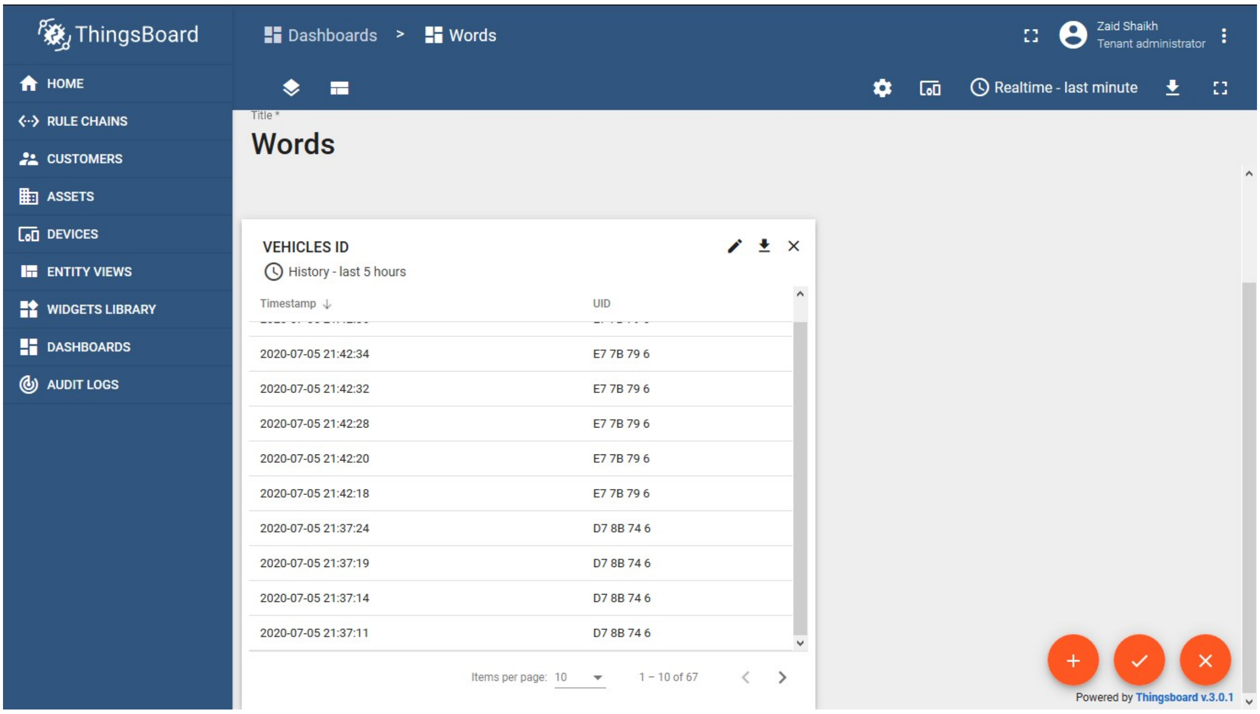


Fig 12. ID display on thingsboard server

8.2 Over speeding control

The connections of speed sensor with the processor is given below:

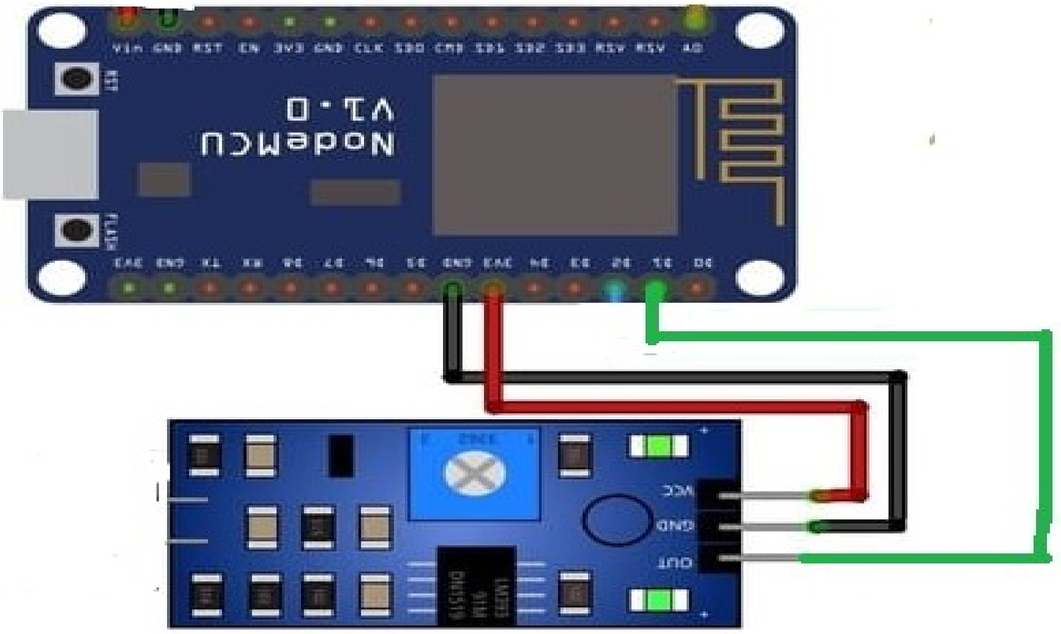


Fig 13. Speed sensor circuit diagram

The programming logic for the sensor part is based on the sections or slots present in the wheel. For our wheel, there was 20 slots and therefore for calculating the rpm we need to divide the steps by 20. Using arduino IDE we can upload the following program:

```
#include <ArduinoJson.h>

#include "ThingsBoard.h"

#include <ESP8266WiFi.h>


//---------- WIFI and Device Configuration----------//

#define WIFI_AP                "realme X"

#define WIFI_PASSWORD            "whateverittakes"

#define TOKEN                  "KDhtyDyIDFQ4t3iDZXtU"
```

```
#define THINGSBOARD_SERVER   "demo.thingsboard.io"

//*************************************************

WiFiClient espClient;       //Initialize ThingsBoard client

ThingsBoard tb(espClient);        // Initialize ThingsBoard instance

int status = WL_IDLE_STATUS;      // the Wifi radio's status

#define  SERIAL_DEBUG_BAUD      115200      //---  Baud  rate  for
debug serial

//-----*** SPEED SENSOR pin and required variabless--------//

int number = 0;

int sensor = 15;      //Sensor's ouput pin attached here

unsigned long start_time = 0;     //start and end time for calculating
rps using steps

unsigned long end_time = 0;

int steps=0;

float steps_old=0;              //For counting steps using old and new
steps

float temp=0;                  //To store rps temporarily

float rps=0;                   //To store revolutions per minute

float speed_veh;               //For displaying speed

void setup() {

  Serial.begin(SERIAL_DEBUG_BAUD);          //  initialize  serial  for
debugging

  pinMode(sensor,INPUT_PULLUP);          // Defining pin config
```

```
  WiFi.begin(WIFI_AP, WIFI_PASSWORD);    // Initialising WiFi
  InitWiFi();
}

void loop() {
  delay(1000);

  if (WiFi.status() != WL_CONNECTED) {
    reconnect();
  }

  if (!tb.connected()) {                          //TO connect to thingsboard
server
    // Connect to the ThingsBoard
    Serial.print("Connecting to: ");
    Serial.print(THINGSBOARD_SERVER);
    Serial.print(" with token ");
    Serial.println(TOKEN);
    if (!tb.connect(THINGSBOARD_SERVER, TOKEN)) {
      Serial.println("Failed to connect");
      return;
    }
  }

  Serial.println("Sending data...");

 start_time=millis();          //***The logic to calculating speed using
sensor
```

```
 end_time=start_time+1000;
 while(millis()<end_time)
 {
   if(digitalRead(sensor))
   {
     steps=steps+1;
     while(digitalRead(sensor));
   }

 }
   //Serial.println("STEPS: ");
   //Serial.print(steps);
     temp=steps-steps_old;
     steps_old=steps;
     rps=(temp/20);
     speed_veh=rps*100;

  //tb.sendTelemetryString("words", "trial");
  //delay(1000);
  tb.sendTelemetryFloat("integer", speed_veh);

  tb.loop();
}

void InitWiFi()
{
  Serial.println("Connecting to AP ...");
```

```
  // attempt to connect to WiFi network


  WiFi.begin(WIFI_AP, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to AP");
}


void reconnect() {
  // Loop until we're reconnected
  status = WiFi.status();
  if ( status != WL_CONNECTED) {
    WiFi.begin(WIFI_AP, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("Connected to AP");
  }
}
```

The speed can be monitored by using one of the widgets on the dashboard. The digital gauge is one of such example and here is the output that we get on the server. Here if the speed crosses the threshold automatically an alarm is generated from the processor side.

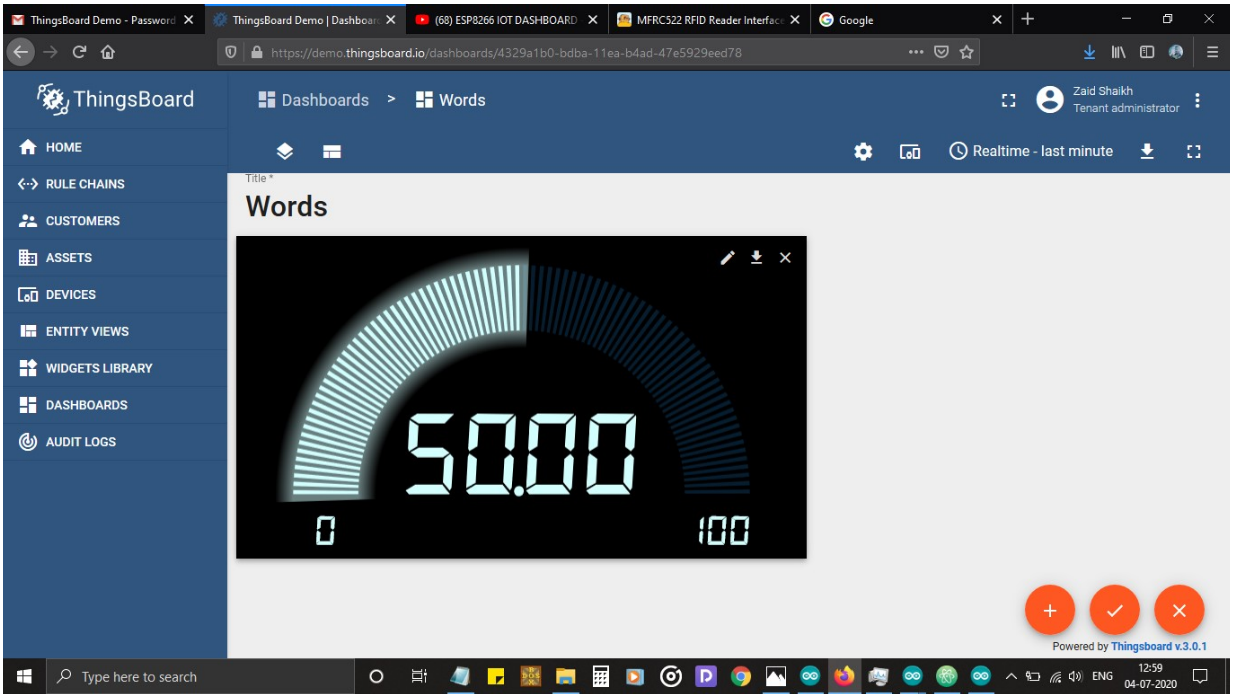


Fig 14. Speed monitor on thingsboard server

We can also combine the widgets to monitor multiple things in one dashboard for our convenience and use it more efficiently.

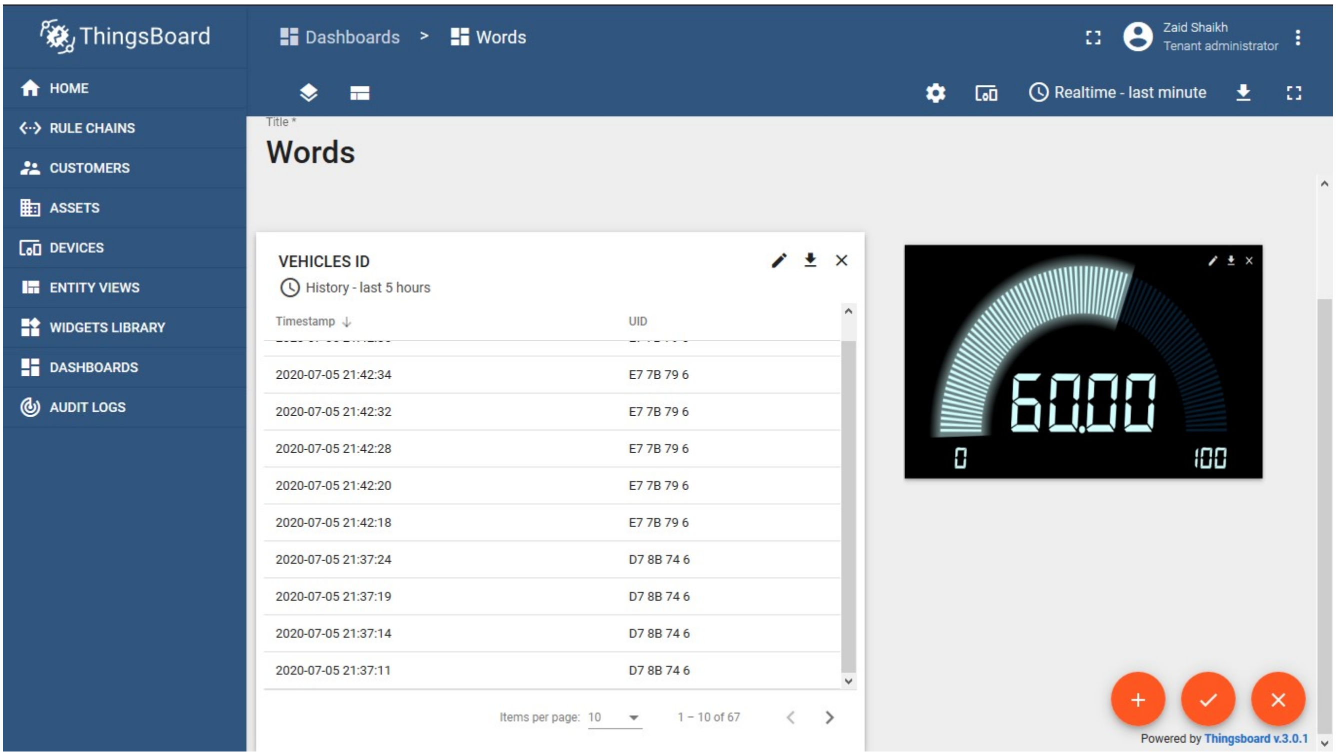


Fig 15. Combined output on thingsboard server

8.3 Signal control based on traffic density

For this part the ultrasonic sensors are interfaced with the board sequentially. For interfacing one Ultrasonic sensor with Nodemcu the below diagram explains:

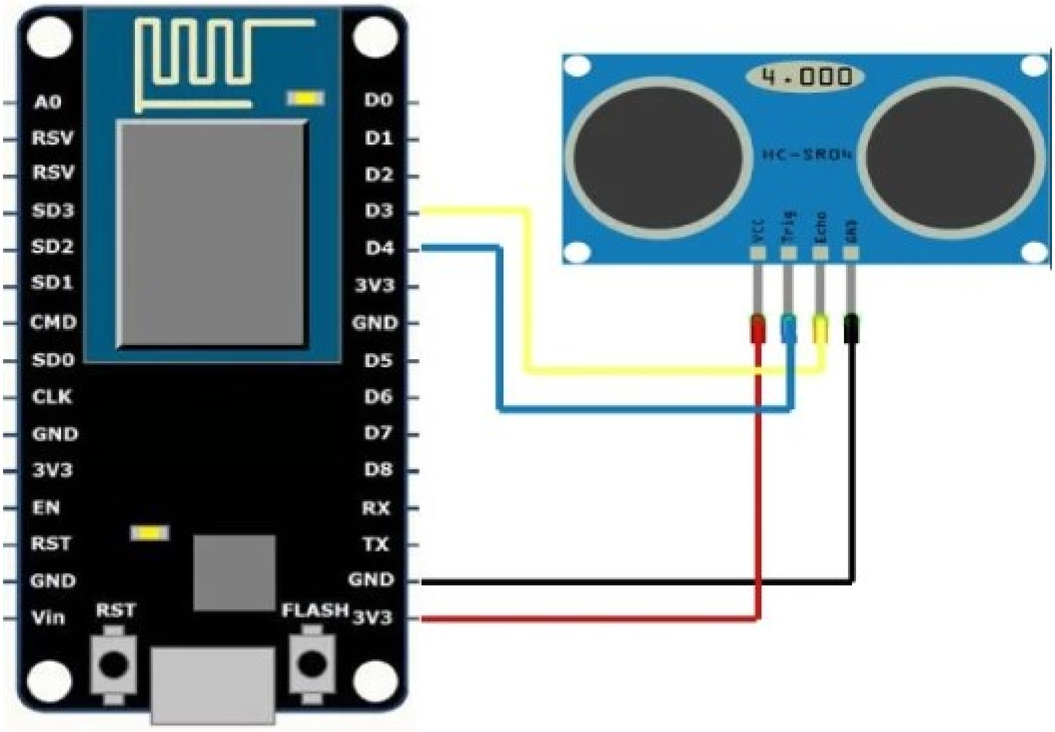


Fig 16. Ultrasonic connection with Nodemcu

Using the above logic we can connect the sensors in place. The programming logic that we used is based on the detection of objects by the sensors. If only one sensor detects the vehicle then the traffic is low, if 2 sensors simultaneously detects vehicles then the traffic is medium, and if all the 3 sensors detects vehicles then the traffic is High.

```
#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>
```

```
// utlrasonic pinout
#define ULTRASONIC_TRIG_PIN       5    // pin TRIG to D1
#define ULTRASONIC_ECHO_PIN       4    // pin ECHO to D2

// user config: TODO
const char* wifi_ssid = "...................";                  // SSID
const char* wifi_password = "...................";             // WIFI
const char* apiKeyIn = ".................";        // API KEY IN
const unsigned int writeInterval = 25000; // write interval (in ms)

// ASKSENSORS config.
String host = "http://api.asksensors.com";             // ASKSENSORS host name

ESP8266WiFiMulti WiFiMulti;

void setup() {

  Serial.begin(115200);

Serial.println("***************************************************");
  Serial.println("********** Program Start : Connect Ultrasonic HC-SR04 + ESP8266 to AskSensors over HTTP");
  Serial.println("Wait for WiFi... ");
  Serial.print("********** connecting to WIFI : ");
  Serial.println(wifi_ssid);
  WiFi.begin(wifi_ssid, wifi_password);
  while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("-> WiFi connected");
  Serial.println("-> IP address: ");
  Serial.println(WiFi.localIP());
  // ultraonic setup
  pinMode(ULTRASONIC_TRIG_PIN, OUTPUT);
  pinMode(ULTRASONIC_ECHO_PIN, INPUT);

}



void loop() {
    // wait for WiFi connection
  if (WiFi.status() == WL_CONNECTED){

        HTTPClient http;

        Serial.print("[HTTP] begin...\n");

        long duration, distance;
        digitalWrite(ULTRASONIC_TRIG_PIN, LOW);
        delayMicroseconds(2);

        digitalWrite(ULTRASONIC_TRIG_PIN, HIGH);
```

```
delayMicroseconds(10);

digitalWrite(ULTRASONIC_TRIG_PIN, LOW);
duration = pulseIn(ULTRASONIC_ECHO_PIN, HIGH);
distance = (duration/2) / 29.1;
Serial.print("********** Ultrasonic Distance: ");
Serial.print(distance);
Serial.println(" cm");

// Create a URL for the request
String url = "";
url += host;
url += "/write/";
url += apiKeyIn;
url += "?module1=";
url += distance;

Serial.print("********** requesting URL: ");
Serial.println(url);
http.begin(url); //HTTP

Serial.println("> Request sent to ASKSENSORS");

Serial.print("[HTTP] GET...\n");

if(i=1){
   Serial.println("Traffic density is low ");
```

```
}
elseif(i=2)
{
  Serial.println("Traffc density is MEDIUM");

}
elseif(i=3)
{
  Serial.println("Traffic densityis HIGH");
}
// start connection and send HTTP header
int httpCode = http.GET();

// httpCode will be negative on error
if(httpCode > 0) {
    // HTTP header has been send and Server response header has been handled
    Serial.printf("[HTTP] GET... code: %d\n", httpCode);

    // file found at server
    if(httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        Serial.println(payload);
    }
} else {
    Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
```

```
        }

        http.end();

        Serial.println("********** End ");

Serial.println("******************************************
***********");
    }

    delay(writeInterval);
}
```

9. CONCLUSION

The advanced techniques can be used to maintain traffic discipline along with traffic safety. The increasing number of vehicles seems to be unceasing and thus the systems which are in place have to be upgraded. The evolving technologies provide a better stage for modifications of the systems. The data that can be taken from vehicles using various sensors can be used for analyzing the traffic pattern and further make appropriate decisions for the modifications.

Thus the changes that are done for the traffic control must provide a better stability for the system. Also the over speeding problem is still one of the major problem for leading to accidents. Hence the system can decrease the the risk by some minor level that is also considered to be effective.

10. FUTURE SCOPE

The existing traffic management system uses a fine system based on the same IP technology. But if this system is modified using the proposed systems, then a more efficient Fine collection can be done. One of the reasons that leads to congestion often is the stopping of vehicles in the middle of road. Using RFID technology, this can be detected and stopped up to a great extent.

The speed detection proposed can be further advanced by using GPS along with the proposed system.By using GPS the traffic control authorities can also detect the location of the vehicle where the speed was over the threshold value. By using this, we can also set different threshold values for different places and thus make the system more convenient for the drivers as well as traffic controllers. One of the other advancement can be done to detect the conditions of the road on which vehicle is travelling. If the road is met with an accident or the condition of road is not proper, the system will inform the driver about the scenario and prevent the driver from going into improper road.

Also using proximity sensors and IR sensors along with RFID and Ultrasonic can increase the efficiency by considerable margin and also a stability to the system. Using GPS module to detect the location of the vehicle when needed for advice can be added as a feature along with speed control so that the user can see the traffic density of the location and the signal duration also. This can decrease the congestion up to a certain level.

11. REFERENCES

[1] https://en.wikipedia.org/wiki/Traffic_light_control_and_coordination

[2] https://www.fhwa.dot.gov/publications/research/operations/its/06108/02.cfm

[3] https://ieeexplore.ieee.org/document/7423566

[4] http://www.ijirset.com/upload/2015/tapsa/4_EC003-1.pdf

[5] https://pdfs.semanticscholar.org/229c/fc78c7124cc5d1f45d1ca15d9fb25b383822.pdf

[6] https://www.semanticscholar.org/paper/A-Smart-Traffic-Management-System-for-Congestion-of-Chandana-Sundaram/b89c48460e143dcb6d07bc0fad069344042b86f8

[7] https://demo.thingsboard.io/

[8] https://en.wikipedia.org/wiki/Arduino

[9] https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor

[10] https://www.cnbc.com/2017/11/17/tesla-model-s-best-features.html