

# **AUTOMATIC SPEED CONTROLLER OF VEHICLES NEAR SCHOOLS AND COLLEGES**

**Submitted in partial fulfillment of the requirements for the award of**

**Bachelor of Engineering Degree in  
Electrical and Electronics Engineering**

**by**

**G.KAVITHA(Register No. 37140042)**

**K.KOWSALYA (Register No. 37140046)**



**DEPARTMENT OF ELECTRICAL AND ELECTRONICS  
ENGINEERING  
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**APRIL 2021**



# **SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

---

## **DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of  
**G.KAVITHA** (Reg. No. 37140042) and **K.KOWSALYA** (Reg. No. 37140046) who  
carried out the project entitled "**AUTOMATIC SPEED CONTROLLER OF  
VEHICLES NEAR SCHOOLS AND COLLEGES**" under our supervision from  
December 2020 to April 2021.

**Mrs. V. MEENAKSHI**  
**Internal Guide**

**Dr. V. SIVACHIDAMBARANATHAN**  
**Head of the Department**

---

**Submitted for Viva voce Examination held on\_\_\_\_\_**

**Name:**

**Signature:**

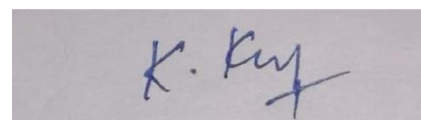
**Internal Examiner**

**External Examiner**

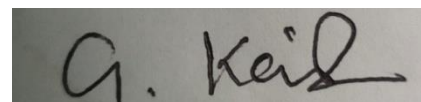
## DECLARATION

We **G.KAVITHA** (Reg. No. 37140042) and **K.KOWSALYA** (Reg. No. 37140046) Here by declare that the Project Report entitled “**AUTOMATIC SPEED CONTROLLER OF VEHICLES NEAR SCHOOL AND COLLEGES**” done by us under the guidance of **Mrs. V. MEENAKSHI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electrical and Electronics Engineering.

1.

Handwritten signature of K. Kowsalya in blue ink on a light background.

2.

Handwritten signature of G. Kavitha in black ink on a dark background.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATES**

## ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA., Dean, School of Electrical and Electronics Engineering** and **Dr. V. SIVACHIDAMBARANATHAN, Head of the Department, Dept. of Electrical and Electronics Engineering** and **Internal Guide Mrs. V. MEENAKSHI**, for providing us necessary support and details at the right time during the progressive reviews.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electrical and Electronics Engineering who were helpful in many ways for the completion of the project.

## **ABSTRACT**

As far as automobiles are concerned, safety is very important to reduce the occurrence of accidents in speed restricted zones. It minimizes the loss of property and life. According to the recent surveys, in the past few years, an accident near the school zones, hospital zones and sharp turnings have increased tremendously, because of their hurry to get the targeted place soon. Therefore controlling vehicle speed has been a crucial issue to be considered. This paper aims to give a practical, compact and simple design to develop an automatic vehicle speed control system, which has to be quickly get implemented in school, college, hospital, sharp turning zones to reduce the number of accidents. This automated speed controlling system is built using the microcontroller- based platform of the Arduino Uno board. Here the Arduino is programmed in such a way that, the prescribed speed limit was incorporated in the transmitter unit which transmits the signals, and it was received by the receiver in the vehicle using wireless communication technology and the speed of the vehicle was automatically controlled by the input signals by the receiver, with the help of speed encoder sensor.

## TABLE OF CONTENTS

| CHAPTER NO. | TITLE                               | PAGE NO. |
|-------------|-------------------------------------|----------|
|             | ABSTRACT                            | v        |
|             | LIST OF FIGURES                     | ix       |
| <b>1</b>    | <b>INTRODUCTION</b>                 |          |
|             | 1.1GENERAL                          | 1        |
|             | 1.2LITERATURE SURVEY                | 2        |
|             | 1.3 SCOPE OF THE PROJECT            | 6        |
|             | 1.4 ADVANTAGES                      | 6        |
| <b>2</b>    | <b>PROJECT DESCRIPTION</b>          |          |
|             | 2.1 BLOCK DIAGRAM                   | 7        |
| <b>3</b>    | <b>CIRCUIT DIAGRAM</b>              |          |
|             | 3.1 CIRCUIT DIAGRAM FOR TRANSMITTER | 8        |
|             | 3.2 CIRCUIT DIAGRAM FOR RECEIVER    | 9        |
| <b>4</b>    | <b>PROPOSED RESULT</b>              |          |
|             | 4.1 PROPOSED RESULT(NOT DETECT)     | 1        |
|             | 4.2 PROPOSED RESULT                 | 10       |
|             | 4.3 PROGRAMMING:PROTEUS             | 10       |

|          |                                 |    |
|----------|---------------------------------|----|
| <b>5</b> | <b>HARDWARE DESCRIPTION</b>     |    |
|          | 5.1 POWER SUPPLY UNIT           | 14 |
|          | 5.2 ECU(ENGINE CONTROL UNIT)    | 15 |
|          | 5.2.1 ACTIVE RFID READER        | 15 |
|          | 5.2.2 APPLICATION CIRCUIT       | 16 |
|          | 5.2.3 SPEED CONTROL OF DC MOTOR | 17 |
|          | 5.3 ARDUINO REQUIREMENTS        | 19 |
|          | 5.4 EMBEDDED C                  | 21 |
|          | 5.5 PIN DIAGRAM                 | 22 |
|          | 5.5.1 PI DESCRIPTION            | 24 |
|          | 5.6 ATMEGA328 PINOUT            | 23 |
|          | 5.7 ARDUINO                     | 23 |
|          | 5.8 ARCHITECTURE DIAGRAM        | 25 |
|          | 5.9 STACK POINTER               | 30 |
| <br>     |                                 |    |
| <b>6</b> | <b>CONCLUSION</b>               |    |
|          | CONCLUSION                      | 35 |
|          | REFERENCES                      | 36 |

## LIST OF FIGURES

| FIGURE NO. | TITLE                   | PAGE NO. |
|------------|-------------------------|----------|
| 2.1        | BLOCK DIAGRAM           | 8        |
| 2.1        | TRANSMITTER             | 8        |
| 3.1        | CIRCUIT FOR TRANSMITTER | 8        |
| 3.2        | CIRCUIT FOR RECEIVER    | 9        |
| 4.1        | SAFE ZONE NOT DETECT    | 10       |
| 4.2        | SAFE ZONE DETECT        | 10       |
| 3.2        | ARCHITECHURE DIAGRAM    | 17       |
| 4.1        | POWER SUPPLY UNIT       | 14       |
| 5.1        | APPLICATION CIRCUIT     | 16       |
| 5.3        | DC MOTOR                | 18       |
| 5.5        | PIN DIAGRAM             | 22       |
| 5.6        | ATMEGA328 PINOUT        | 23       |
| 5.7        | ARDUINO                 | 23       |
| 5.8        | ARCHITECTURE DIAGRAM    | 25       |

## **INTRODUCTION**

### **GENERAL INTRODUCTION**

Recent studies show that one third of the number of road accidents occur due to over speeding and traffic violation. Running vehicle disregarding the red light has become a major problem now a day. Reduction of the number of accidents and mitigation of their consequences are a big concern for traffic authorities, the automotive industry and transport research groups. To reduce the rate of road accident we propose a system which controls the speed of vehicle automatically in any critical zone, without major inconvenience to driver .A controlling module in the vehicle then takes the decision and control the speed accordingly. Index Terms-ECU (Electronic/Engine Control Unit), Driver assistance systems, RFID (Radio Frequency Identification).

## **LITERATURE REVIEW**

### **1. vinayak Ikke, Nilam Chavan, “Car Over-Speed Detection with Remote Alerting”,, IJIRST –International Journal for Innovative Research in Science & Technology | March 2016 | Volume 2 | Issue 10 |**

The aim of this project is to develop a device to detect rash driving on highways and to alert the traffic authorities wirelessly about the speed details and any speed violation. Accidents due to rash driving on highways are on the rise and people are losing their lives because of others mistakes. In the present system, to detect rash driving the police has to use a handheld radar gun and then aim at the vehicle to record its speed. If the speed of the vehicle exceeds the speed limit, nearest police station is informed to stop the speeding vehicle. This is an ineffective process as after detecting one has to inform the same and a lot of time is wasted. The proposed system will check on rash driving by calculating the speed of a vehicle using the time taken to travel between the two set points at a fixed distance and then transmit the data over 2.4GHz to the central control room. A set point consists of a pair of sensors comprising of an IR transmitter and an IR receiver, each of which are installed on either sides of the road. The speed limit set by the device is kept at the very location depending upon the traffic. The time taken by the vehicle to travel from one set point to the other is calculated by a microcontroller program. Based on that time it then calculates the speed and displays that on an LCD and also transmits the same. Moreover if the vehicle crosses the speed limit, a buzzer sounds alerting the police both at the location and wirelessly at the control room.

### **2. D.Guru Pandi, J.Navarajan, R.Vishal, D.Vibuvasan, “Embedded Based Accident Prevention Technique Using Image Processing”, IJARIE-ISSN (O)-2395-4396 Vol-2 Issue-2 2016.**

The accident prevention system fully works on the basis of a continuous video processing system, a camera is placed at the backside of a rear view mirror which is placed in the front windshield of the car which works like a perception system collecting all the changes in the environment and gives it to an PIC controller which does processing and senses any obstacles present on road and

if the size of the obstruct is minimum and doesn't cause any damage that it is well below the car. The car doesn't slow down and will be able to move continuously with the same speed, but if there is any obstacle that hinders the progress of the car the processor will check whether the car is slowing down or not . If it doesn't slow down the automated system applies brakes and slows down the car before getting collided with the obstruction. Initially our idea was to give a notification PIC controller to the driver inside the car but the response speed of human is less so we decided to make it fully automated.

**3. “Distributed Traffic Control for Reduced Fuel Consumption and Travel Time In-transportation Networks” IJIRST –International Journal for Innovative Research in Science & Technology| Volume 2 | Issue 10 | March 2015.**

This paper proposes a distributed framework for optimal control of vehicles in transportation networks. The objective is to reduce the balanced fuel consumption and travel time through hybrid control on speed limit and ramp metering rate. The dual decomposition theory associated with the sub gradient method is then applied in order to decompose the optimal control problem into a series of suboptimal problems and then solve them individually via networked road infrastructures (RIs). Coordination among connected RIs is followed in each iteration to update the individual controls.

**4. Heung-Sub Shin, Sang-Joong Jung, Jong-Jin Kim and Wan-Young Chung, “Real Time Car Driver ’s Condition Monitoring System”, 978-1-4244-8168-2/10/\$26.00 2010 IEEE SENSORS 2010 Conference**

Driver's drowsiness and fatigue have been major causes of the serious traffic accidents, which make this an area of great socioeconomic concern. This paper describes the design of ECG (Electrocardiogram) sensor with conductive fabric electrodes and PPG (Photoplethysmogram) sensor to obtain physiological signals for car driver's health condition monitoring. ECG and PPG signals are transmitted to base station connected to the server PC via personal area network for practical test. Intelligent health condition monitoring system is designed at the server to analyze the PPG and ECG signals. Our purpose for intelligent health condition monitoring

system is managed to process HRV signals analysis derived from the Physiological signals in time and frequency domain and to evaluate the driver's drowsiness status.

**5. Mayuresh Desai, Arati Phadke, "Internet of Things based vehicle monitoring system", 978-1-5090-4884-7/17/\$31.00 2017 IEEE.**

Advances in technologies and availability of economical open source hardware systems are setting a new trend in system designing. Use of technologies like Internet of Things (IoT) can ease the process of data collection and analysis. The main objective of the paper is to describe a system which can monitor or track the location and vehicle parameters of different test vehicles from a centralized place for research and development purposes and to store data of testing parameters of those vehicles on the server for further analysis and records. System design will be generalized for monitoring different parameters like Location, Vehicle speed, Engine compartment temperature, Fuel consumption and many more. Proposed system uses Open source controller and GPS/GSM/GPRS module for data transfer application.

**6. Manuel Fogue, Piedad Garrido, Francisco J. Martinez, "Prototyping an Automatic Notification Scheme for Traffic Accidents in Vehicular Networks", 978-1-4577-2028-4/11/\$26.00 c 2011 IEEE**

The new communication technologies integrated into the automotive sector offer an opportunity for better assistance to people injured in traffic accidents, reducing the response time of emergency services, and increasing the information they have about the incident. Determining more accurately the human and material resources required for each particular accident could significantly reduce the number of victims. The proposed system requires each vehicle to be endowed with an On-Board Unit responsible for detecting and reporting accident situations to an external Control unit that estimates its severity, allocating the necessary resources for its assistance.

**7. B.Karthiga, B.Arun kumar, "A Novel System for Severity Estimation of Automotive Accident with Automatic Notification", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 10, October 2014.**

New communication technologies integrated into modern vehicles offer an opportunity for better assistance to people injured in traffic accidents. Recent studies show how communication capabilities should be supported by artificial intelligence systems. To improve the overall rescue process, a fast and accurate estimation of the severity of the accident represent a key point to help emergency services better estimate the required resources. This paper proposes a novel intelligent system which is able to automatically detect road accidents, notify them through vehicular networks, and estimate their severity based on the concept of data mining and knowledge inference.

Our system considers the most relevant variables that can characterize the severity of the accidents (variables such as the vehicle speed, the type of vehicles involved, the impact speed ). Results show that a complete Knowledge Discovery in Databases (KDD) process, with an adequate selection of relevant features, allows generating estimation models that can predict the severity of new accidents.

**8. Hiwot Teshome and Tibebe Beshah, “Traffic Accident Analysis from Drivers Training Perspective Using Predictive Approach”, *HiLCoE Journal of Computer Science and Technology*, Vol. 2, No. 1**

Road traffic accidents are among the top leading causes of deaths and injuries, especially in the developing world. Ethiopia in particular experiences the highest rate of such accidents. One of the solutions to reduce the problem of traffic accident is finding the causes through research studies. In this work, factors for the relationship between driver’s training and road traffic accident are identified using data mining classification technique. Using WEKA as a tool, different rules are generated so that different domain experts get new insights related to road traffic accidents.

**9. Jeffrey Miller, “Vehicle-to-Vehicle-to-Infrastructure (V2V2I) Intelligent Transportation System Architecture”, 2008 IEEE Intelligent Vehicles Symposium June 4-6, 2008.**

The V2V2I architecture leverages the benefits of fast queries and responses from the V2I architecture, but with the advantage of a distributed architecture not having a single point-of-failure from the V2V architecture. In the V2V2I architecture, the transportation network is broken into zones in which a single vehicle is known as the Super Vehicle. Only Super Vehicles are able to communicate with the central infrastructure or with other Super Vehicles, and all other vehicles can only

communicate with the Super Vehicle responsible for the zone in which they are currently traversing. I describe the Super Vehicle Detection (SVD) algorithm for how a vehicle can find or become a Super Vehicle of a zone and how Super Vehicles can aggregate the speed and location data from all of the vehicles within their zone to still ensure an accurate representation of the network.

## **SCOPE OF THE PROJECT**

- This system proves to be highly effective in minimizing the over speeding and unwanted accidents in restricted zones.
- In current systems, there is no autonomous speed restriction in the vehicle to avoid accidents.
- Hence further research and optimizations of the automatic vehicle speed control system will allow us to implement in vehicles for improved safety for roadside pedestrians, passengers, and other road users.

## **ADVANTAGES:**

- ❖ Easy adaption in vehicles
- ❖ Low power consumption (5 to 12V)
- ❖ Quick response (10sec approx...)
- ❖ Durability (20 sec approx...)
- ❖ Highly secured to accident prevent
- ❖ Highly effective

## CHAPTER 2

### PROJECT DESCRIPTION

#### 2.1 BLOCK DIAGRAM

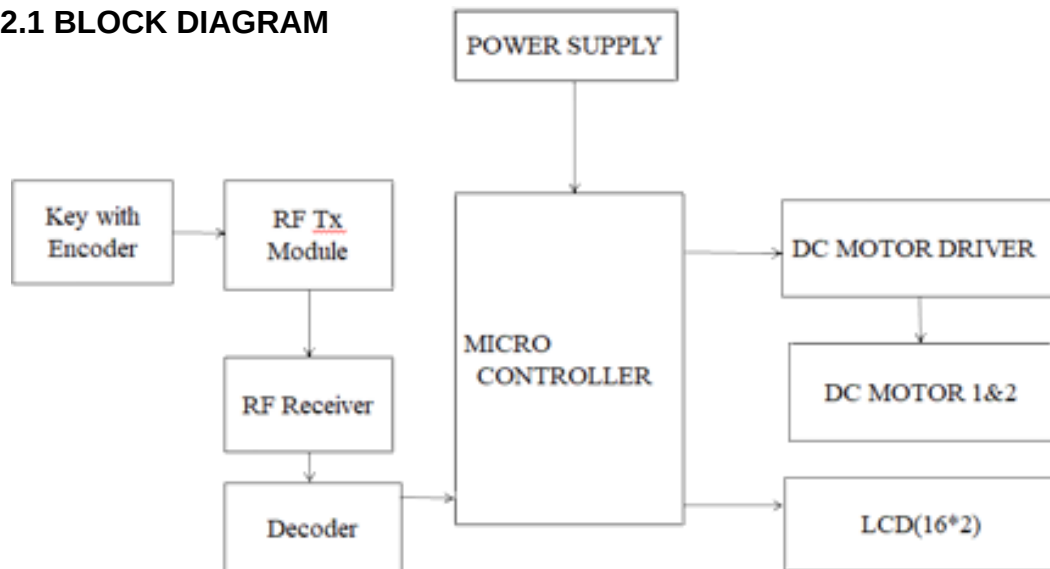


FIGURE 2.1 BLOCK DIAGRAM

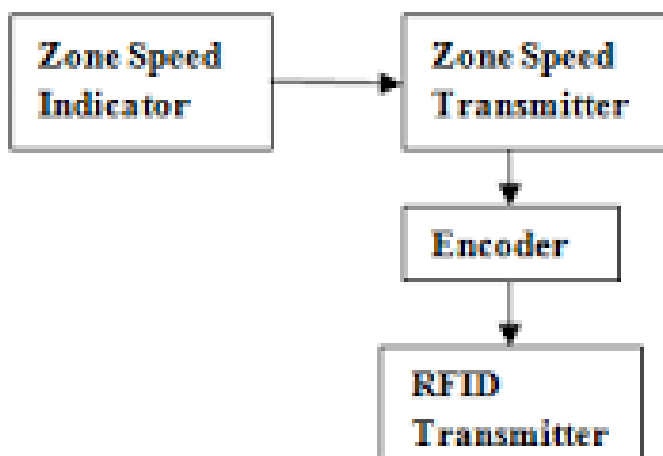


FIGURE 2.1 TRANSMITTER

## 2.2 IMPLEMENTATION:

- The transmitter block consists of the power supply, Arduino UNO and transmitter. In this block, the Arduino microcontroller was programmed with a predefined speed limit and transmits the signal with the help of transmitter wireless communication.
- **Receiver block diagram:** receiver block diagram In the receiver section, it receives the speed of the vehicle through speed encoder input signal, and other components of the receiver unit are a dc motor, motor driver, LCD display, transmitter receiver, Arduino UNO as a microcontroller.
- Based on the signal received from the transmitter placed in a special zone the speed of the vehicles reduces automatically with the help of these devices.

## CHAPTER 3

### CIRCUIT DIAGRAM

#### CIRCUIT DIAGRAM FOR TRANSMITTER

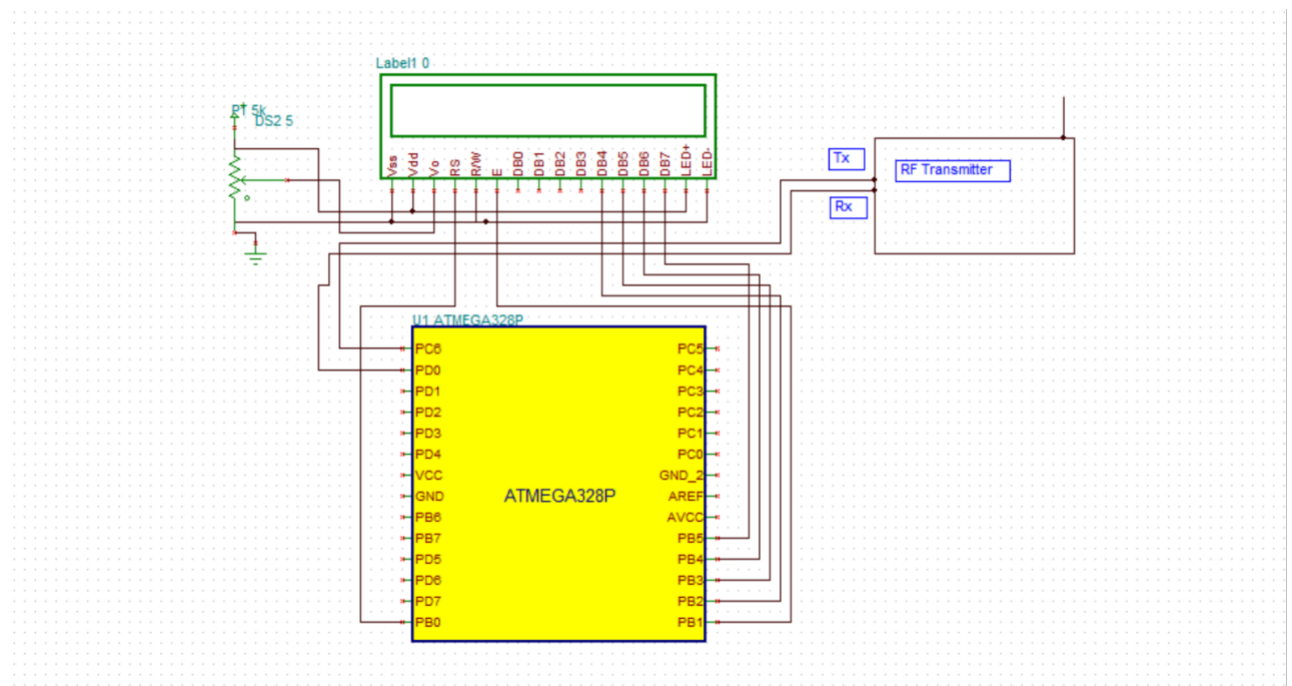


FIGURE 3.1 CIRCUIT FOR TRANSMITTER

## CIRCUIT DIAGRAM FOR RECEIVER

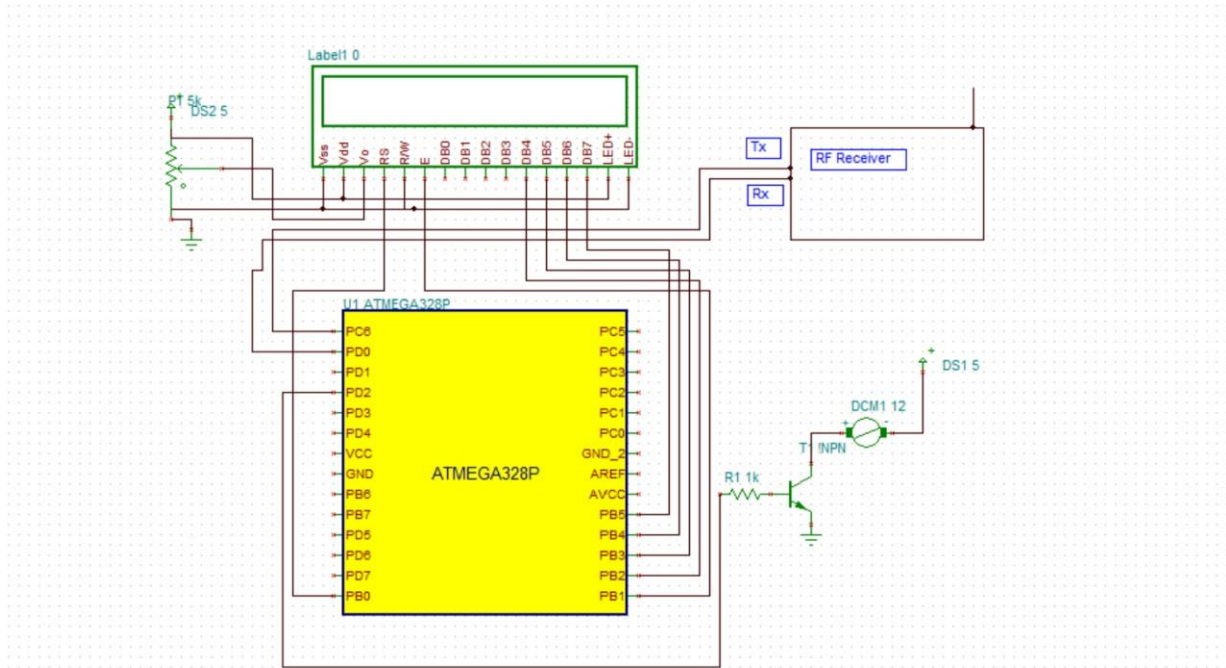


FIGURE 3.2 CIRCUIT FOR RECEIVER

### EXPLANATIONS:

- The above figure can represent the circuit diagram of our design. Coming to the operation, many accidents at High-ways are taking place due to the close running of vehicles, all of sudden, if the in front vehicle driver reduces the speed or applied breaks, then it is quite difficult to the following vehicle driver to control his vehicle, resulting accident. To avoid this kind of accident, the warning system, which contains display system can arrange at rear side of vehicle.
- The current speed of the vehicle is sensed by the dc motor and the output of it was given to the microcontroller where it compares the speed with the prescribed limit and the speed is controlled automatically.

## CHAPTER 4

### PROPOSED RESULT

#### PROPOSED RESULT (SAFE ZONE NOT DETECT)

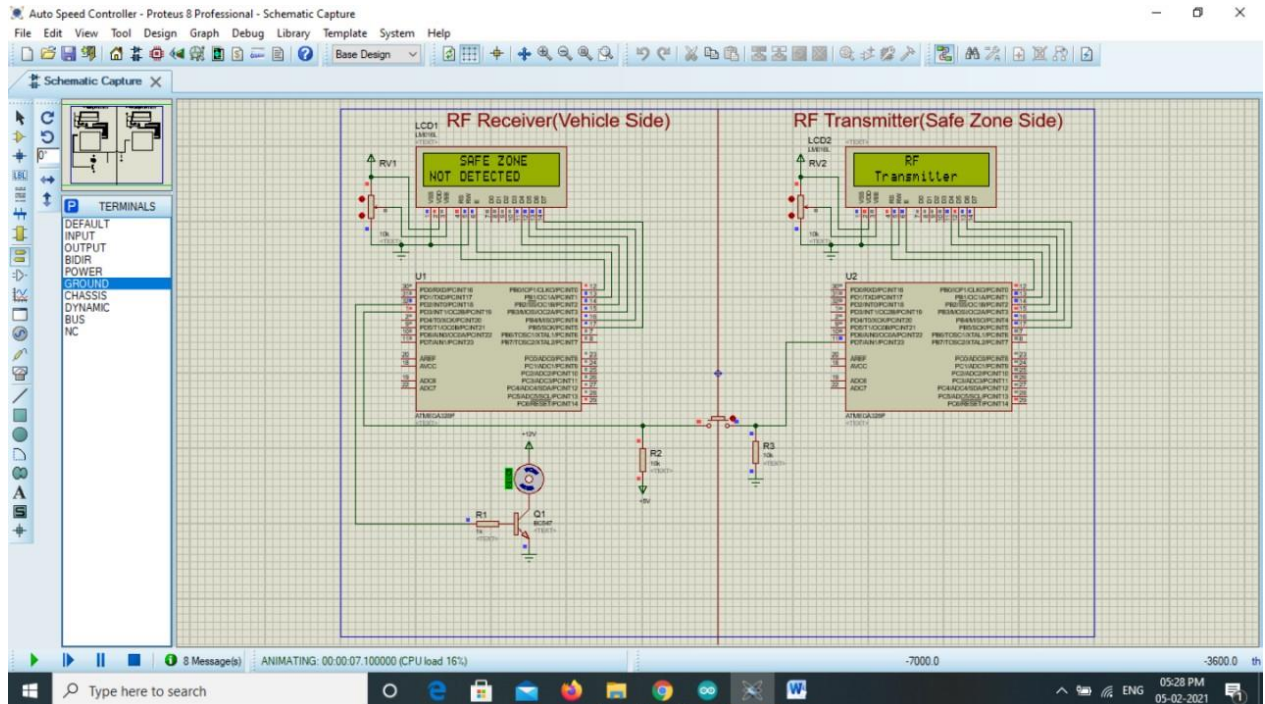
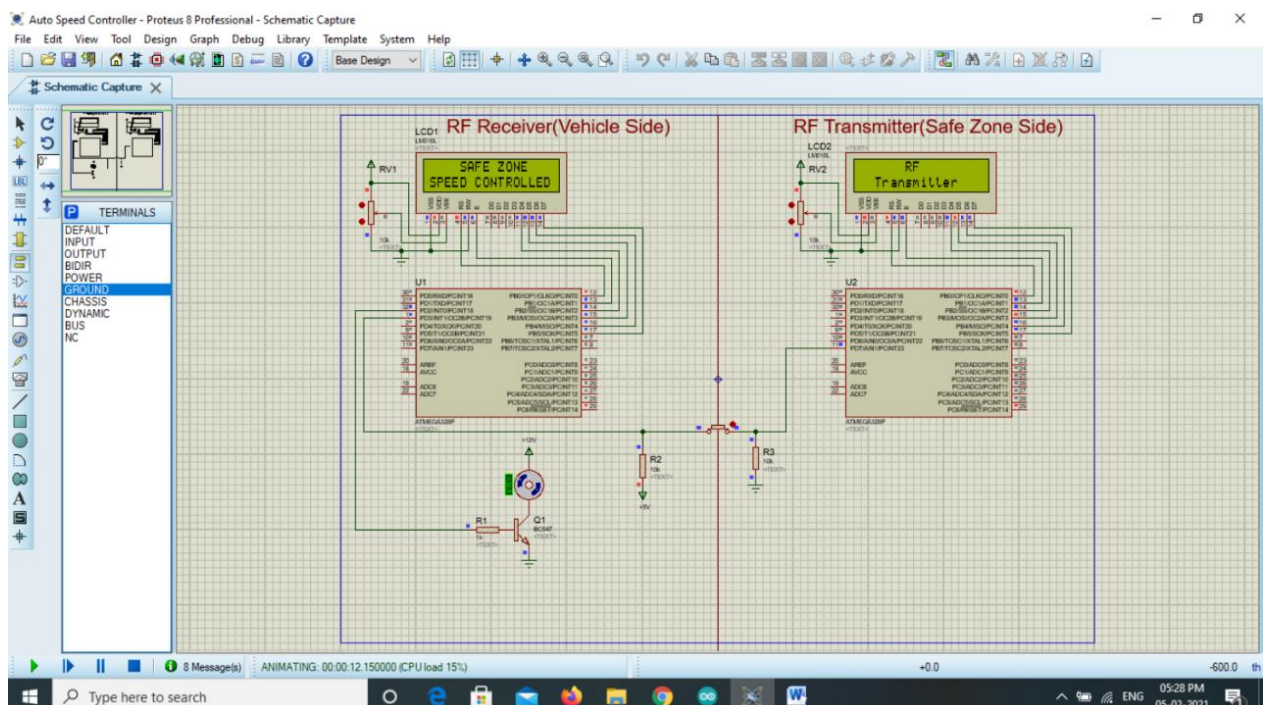


Figure 4.1 SAFE ZONE NOT DETECT

#### PROPOSED RESULT (SAFE ZONE DETECTED)



10  
Figure 4.2 SAFE ZONE NOT DETECT

## **DESCRIPTION OF PROPOSED RESULT**

The proposed result for this topic is to prepare a model which will show the experimental view of this project covering the theoretical as well as practical areas related to this project this model will also show the practical implementation of the device which could be fitted in to the automobiles for safety purposes.

We are trying to work with the all-pros and cons related to this project. Hopefully we could come with a model which will show the experimental view of Smart Display and control device through which the Idea of automated speed control concept to prevent the accident and control traffic would be more clearly understood.

We are actually presenting the layout of the project on which we will be working. Basically it consists of two sections: zone status transmitter according to the zone, it waits for few seconds, otherwise unit and receiver (speed display and control) unit. Once the information is received from the zones, the vehicle's embedded unit is automatically alert the driver, to reduce the speed vehicle's SDC unit automatically

## PROGRAMMING : PROTEUS SOFTWARE

### **CODING:**

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(13,12,11,10,9,8);

const int motor = 6;

int j;

char i;

void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.print("Automatic Speed ");
  lcd.setCursor(0,1);
  lcd.print("Control - RFID");
  delay(2000);
}

void loop()
{
  if(Serial.available() > 0)
  {
    while(Serial.available())
    i = Serial.read();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Safe Zone Detected");
    lcd.setCursor(0,1);
    lcd.print("Speed Controlled");
    41analogWrite(motor,80);
    delay(1000);
```

```

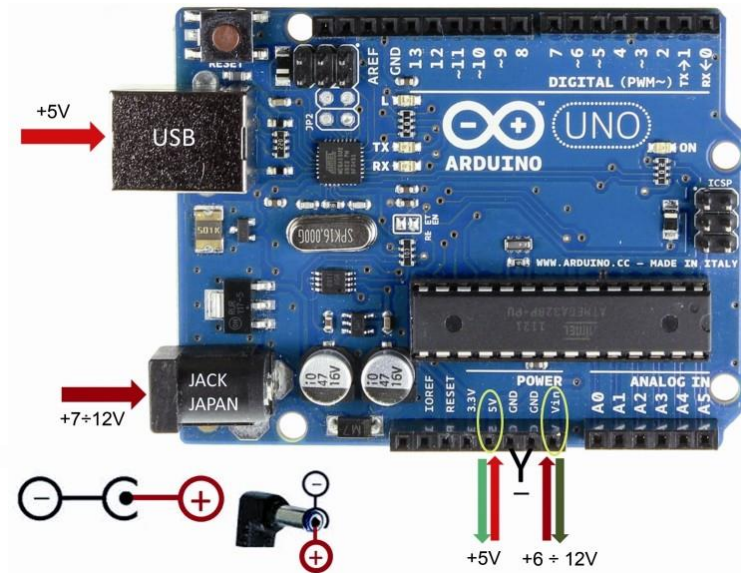
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Speed= 40Kmph");
delay(1000);
}
//
else{
while (!Serial.available()){
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Safe Zone ");
lcd.setCursor(0,1);
lcd.print(" Not Detect...");
analogWrite(motor,255);
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Speed= 80Kmph");
delay(1000);
}
}
delay(100);
}

```

## CHAPTER 5

### HARDWARE DESCRIPTION

#### 5.1 POWER SUPPLY UNIT



**Figure 5.1 POWER SUPPLY UNIT**

- ❖ Must be a DC adapter (i.e. it has to put out DC, not AC); should be between 9V and 12V DC (see note below);
- ❖ Must be rated for a minimum of 250mA current output, although you will likely want something more like 500mA or 1A output, as it gives you the current necessary to power a servo or twenty LEDs if you want to.
- ❖ Must have a 2.1mm power plug on the Arduino end, and The plug must be "centre positive", that is, the middle pin of the plug has to be the + connection. automatically reduce the speed.

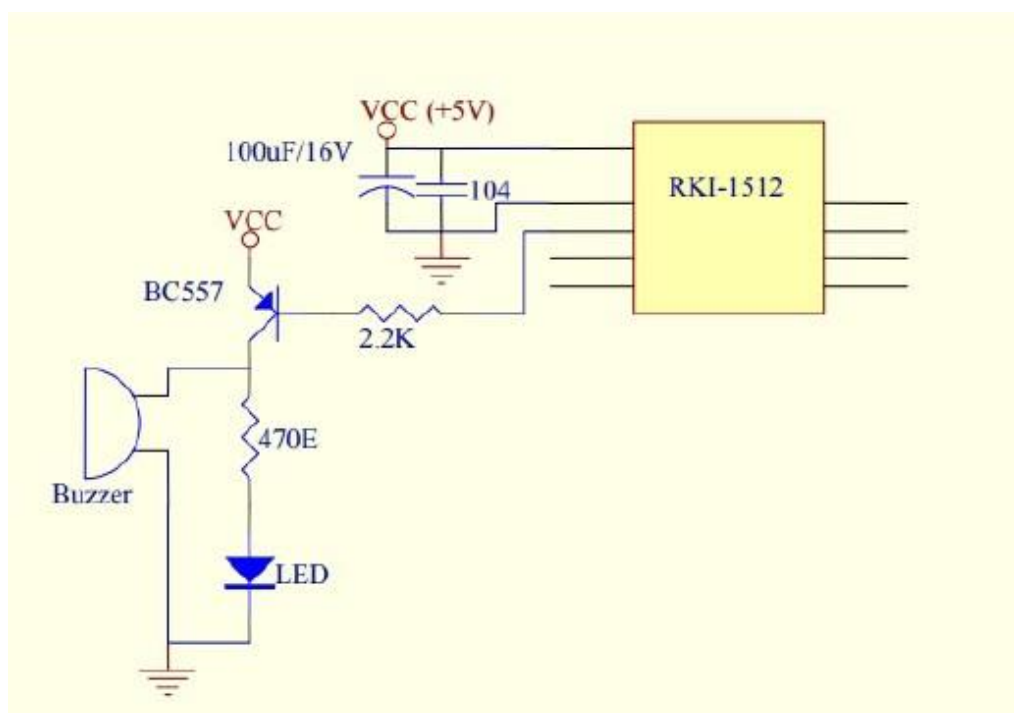
## **ECU (ENGINE CONTROL UNIT):**

- ❖ An engine control unit (ECU) may be a sort of electronic control unit incorporated within the vehicle that controls a series of actuators on an indoor combustion engine to make sure optimal engine performance.
- ❖ ECU does this by reading values from different sensors within the engine bay, interpreting the data and adjusting the engine actuators accordingly. For the vehicles when facility of ECU has not been incorporated, the speed control of auto is formed by controlling air-fuel mixture and ignition timing. Vehicle speed is mechanically set and dynamically controlled by mechanical and pneumatic means..

## **ACTIVE RFID READER:**

- ❖ An active RFID system consists of a reader, tag and antenna. Unlike passive RFID tags that contains merely an antenna and a microchip with no internal power source, a lively RFID tag has its own power source -- an on-board, long-lasting battery that permits the tag to transmit data continuously, no matter whether it's within the field range of a reader.
- ❖ There are two sorts of active RFID tags: transponders and beacons. A transponder only communicates when it's within the immediate vicinity of a reader. A beacon broadcasts constantly. A number of unique characteristics are specific to active RFID tags. Because the tags are often required to survive in harsh environmental conditions like extreme temperatures or precipitation, some are encased during a protective shell. Given the dimensions requirements of the battery, wiring and therefore the durable exterior, active RFID tags are larger than passive tags.

- ❖ Some even have on-board sensors that track environmental parameters just like the temperature, humidity, fluid and power consumption. They can even be used for highly specific use cases, like sensing when a container door is open and there's a change in temperature.
- ❖ This gives shippers an additional insight that permits them to form important adjustments, like rerouting a shipment if it spoils. Active RFID tags require low signal strength to speak and may broadcast up to and even beyond a variety of 100 meters. The high cost generally makes active RFID tags too expensive for easy inventory applications.
- ❖ They are better for tracking high-value items like cargo. Industries that commonly use active RFID tags additionally to shipping and logistics include automobile sales manufacturing, health and medical, construction, mining, remote monitoring and IT asset management.



**Figure 5.2 APPLICATION CIRCUIT**

### **SPECIFICATIONS:**

- ❖ 5VDC through USB (External 5V supply will boost range of the module) Current: <50Ma Operating Frequency: 125Khz Read Distance: 10cm Size 32mm(length) \* 32mm(width) \* 8mm(height)

### **SPEED CONTROL OF DC MOTORS**

- ❖ Basically, there are three ways to vary the speed of DC motors: 1. With the use of mechanical gears to achieve the desired speed. This method is generally beyond the capability of most hobbyist home workshops. 2. Reducing the motor voltage with a series resistor. However this is inefficient (energy wasted in resistor) and reduces torque.
- ❖ The current drawn by the motor increases as the load on the motor increases. More current means a larger voltage drop across the series resistor and therefore less voltage to the motor. The motor now tries to draw even more current, resulting in the motor "stalling". 3. By applying the full supply voltage to the motor in bursts or pulses, eliminating the series dropping effect.
- ❖ This is called pulse width modulation (PWM) and is the method used in this kit. Short pulses means the motor runs slowly; longer pulses make the motor run faster.

### **PRODUCT SPECIFICATIONS**

- ❖ Module: M274
- ❖ Type: Incremental electromechanical
- ❖ Working voltage: 10 v
- ❖ Dimensions: 32 mm x 19 mm x 30 mm
- ❖ Material: PCB and Brass
- ❖ Working voltage: 0 to 5 v

Cycles per rotations: 20



**Figure 5.3 DC MOTOR**

## **ARDUINO REQUIREMENTS**

- ❖ ARDUINO IDE – 1.8.5 Arduino is an open-source electronics platform supported easy-to-use hardware and software. Arduino boards are ready to read inputs - light on a sensor, a finger on a button, or a Twitter message - and switch it into an output - activating a motor, turning on an LED, publishing something online.
- ❖ You can tell your board what to try to to by sending a group of instructions to the microcontroller on the board. To do so you employ the Arduino programing language (based on Wiring), and therefore the Arduino Software (IDE), supported Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to a fantastic amount of accessible knowledge which will be of great help to novices and experts alike.

- ❖ Arduino was born at the Ivrea Interaction Design Institute as a simple tool for fast prototyping, aimed toward students without a background in electronics and programming. As soon because it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.
- ❖ All Arduino boards are completely open-source, empowering users to create them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it's growing through the contributions of users worldwide. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to create low cost scientific instruments, to prove chemistry and physics principles, or to urge started with programming and robotics.
- ❖ Designers and designers build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example.
- ❖ Arduino may be a key tool to find out new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and lots of others offer similar functionality.

- ❖ All of those tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the method of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:  
Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms.
- ❖ The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50  
Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- ❖ Simple, clear programming environment - The Arduino Software (IDE) is easy-to- use for beginners, yet flexible enough for advanced users to require advantage of also . For teachers, it's conveniently supported the Processing programming environment, so students learning to program therein environment are going to be conversant in how the Arduino IDE works.
- ❖ Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language are often expanded through C++ libraries, and other people eager to understand the technical details can make the leap from Arduino to the AVR C programing language on which it's based.
- ❖ Similarly, you'll add AVR-C code directly into your Arduino programs if you would like to.  
Open source and extensible hardware - The plans of the Arduino boards are published under an ingenious Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module so as to know how it works and economize.

## EMBEDDED C:

- ❖ Embedded c may be a set of language extension for the C programming language by the C Standards committee to deal with commonality issues that exist between C extensions for various embedded systems. Historically embedded C programming requires nonstandard extensions to the C language so as to support exotic features like fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.
- ❖ Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, data type declaration, conditional statements(if, switch, case), loops(while, for), functions, arrays and strings, structures and union, bit operations, macros etc
- ❖ During infancy years of microprocessor based systems, programs were developed using assemblers and fused into the EPROMs. There wont to be no mechanism to seek out what the program was doing. LEDs, switches, etc. were used to check for correct execution of the program. But they were too costly and were not quite reliable as well .
- ❖ As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/ controller

## PIN DIAGRAM

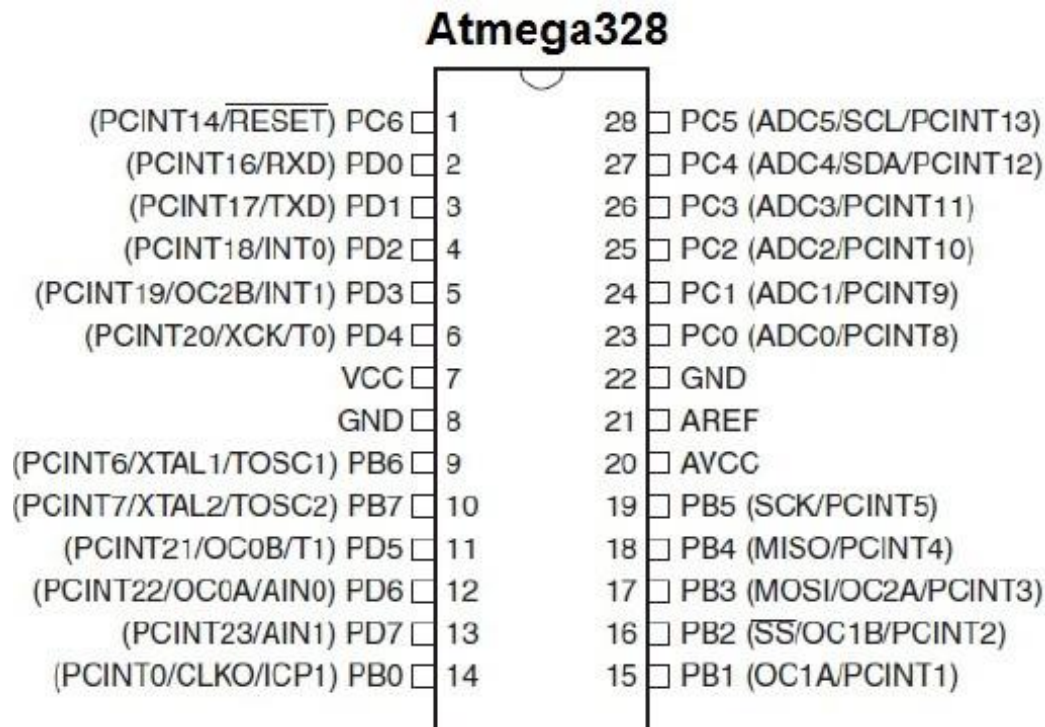


Figure 5.5 PIN DIAGRAM

## ATMEGA 328 PINOUT



Figure 5.6 ATMEGA 328 PINOUT

## 5.7.ARDUINO

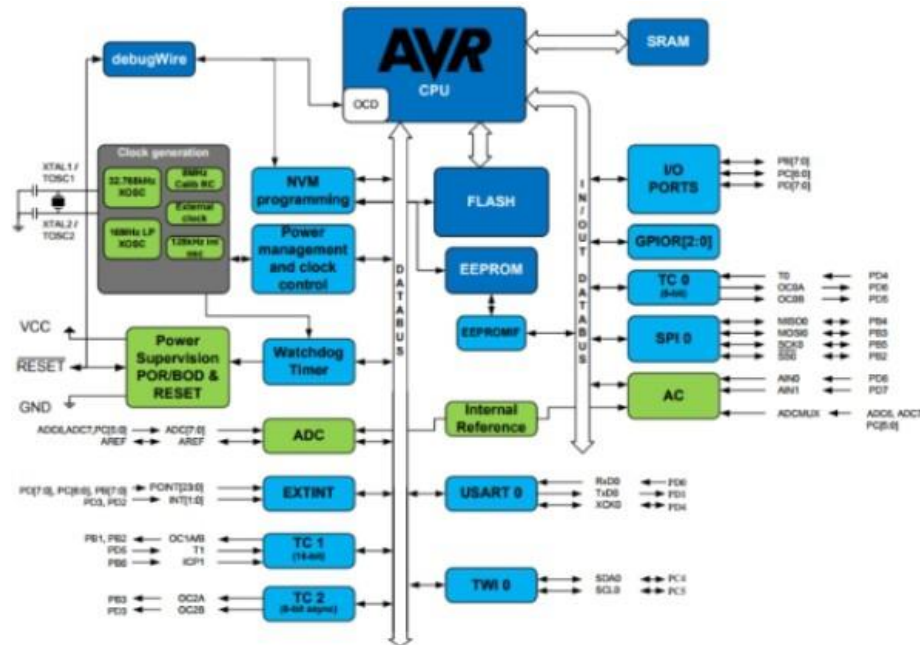


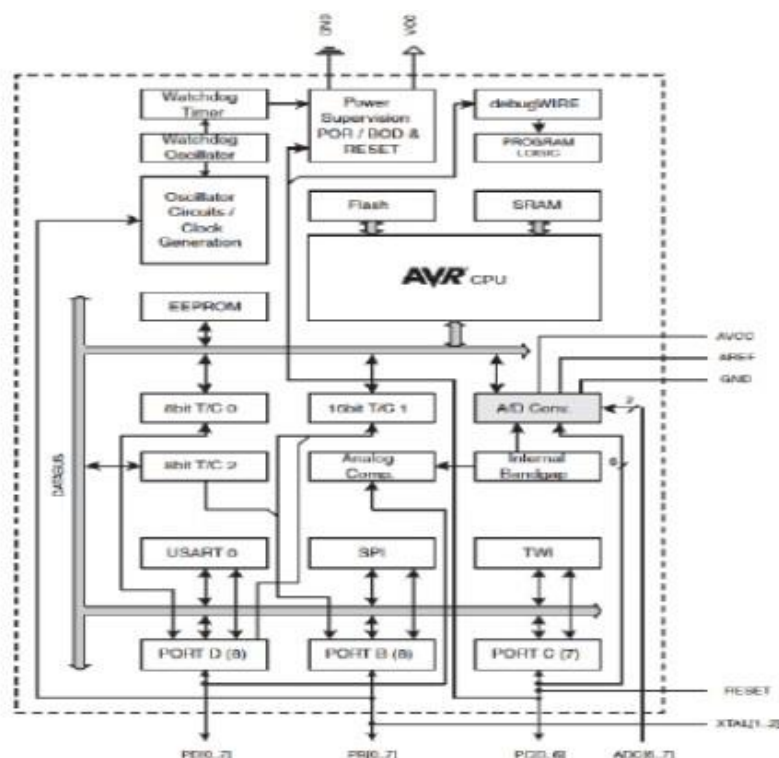
FIGURE 5.7 ARDUINO DIAGRAM

### PIN DESCRIPTION

- ❖ VCC Digital supply voltage.
- ❖ GND Ground.
- ❖ Port B (PB[7:0]) XTAL1/XTAL2/TOSC1/TOSC2 Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for every bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, albeit the clock isn't running. Depending on the clock selection fuse settings, PB6 are often used as input to the inverting Oscillator amplifier and input to the interior clock operating circuit. Depending on the clock selection fuse settings, PB7 are often used as output from the inverting Oscillator amplifier. If the interior Calibrated RC Oscillator is employed as chip clock source, PB[7:6] is employed as TOSC[2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is about .

- ❖ Port C (PC[5:0]) Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC[5:0] output buffers have symmetrical drive characteristics with both high sink and source
- ❖ PC6/RESET If the RSTDISBL Fuse is programmed, PC6 is employed as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the opposite pins of Port C. If the RSTDISBL Fuse is unprogrammed, PC6 is employed as a Reset input. A low level on this pin for extended than the minimum pulse length will generate a Reset, albeit the clock isn't running. Shorter pulses aren't bound to generate a Reset. The various special features of Port C are elaborated within the Alternate Functions of Port C section.
- ❖ Port D (PD[7:0]) Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for every bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

## ARCHITECTURE DIAGRAM



- ❖ AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions within the program memory are executed with one level pipelining. While one instruction is being executed, subsequent instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory. The fast-access Register File contains 32 x 8-bit general purpose working registers with one clock cycle time interval .
- ❖ This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle. Six of the 32 registers are often used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers also can be used as an address pointer for search tables in Flash program memory.
- ❖ These added function registers are the 16-bit X-, Y-, and Z-register, described later during this section. The ALU supports arithmetic and logic operations between registers or between a continuing and a register. Single register operations also can be executed within the ALU. After an mathematical process , the Status Register is updated to reflect information about the results of the operation. Program flow is provided by conditional and unconditional jump and call instructions, ready to directly address the entire address space. Most AVR instructions have one 16-bit word format.
- ❖ Every program memory address contains a 16- or 32-bit instruction. Program non-volatile storage space is split in two sections, the Boot Program section and therefore the application section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the appliance non- volatile storage section must reside within the Boot Program section.

- ❖ During interrupts and subroutine calls, the address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated within the general data SRAM, and consequently the Stack size is merely limited by the entire SRAM size and therefore the usage of the SRAM.
  - ❖ All user programs must initialize the SP within the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible within the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported within the AVR architecture. The memory spaces within the AVR architecture are all linear and regular memory maps. A flexible interrupt module has its control registers within the I/O space with a further Global Interrupt Enable bit within the Status Register.
  - ❖ All interrupts have a separate Interrupt Vector within the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the upper the priority. The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions.
  - ❖ The I/O Memory are often accessed directly, or because the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, this device has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions are often used.
- 3.2.1 ALU – Arithmetic Logic Unit The high-performance AVR ALU operates in direct reference to all the 32 general purpose working registers. Within one clock cycle, arithmetic operations between general purpose registers or between a register and an instantaneous are executed.
- ❖ The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a strong multiplier supporting both signed/unsigned multiplication and fractional format. See Instruction Set Summary section for an in depth description.

- ❖ Status Register The Status Register contains information about the results of the foremost recently executed arithmetic instruction. This information are often used for altering program flow so as to perform conditional operations.
- ❖ The Status Register is updated in any case ALU operations, as laid out in the Instruction Set Reference. This will in many cases remove the necessity for using the dedicated compare instructions, leading to faster and more compact code. The Status Register isn't automatically stored when entering an interrupt routine and restored when coming back from an interrupt. This must be handled by software. Status Register When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used.
- ❖ When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, leading to an I/O address offset within 0x00 - 0x3F.  
Name: SREG Offset: 0x5F Reset: 0x00 Property: When addressing as I/O Register: address offset is 0x3F. FIGURE 3.2.1 ALU Global Interrupt Enable the worldwide Interrupt Enable bit must be set for the interrupts to be enabled.
- ❖ The individual interrupt enable control is then performed in separate control registers. If the worldwide Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings.
- ❖ The Ibit is cleared by hardware after an interrupt has occurred, and is about by the RETI instruction to enable subsequent interrupts.
- ❖ The I-bit also can be set and cleared by the appliance with the SEI and CLI instructions, as described within the instruction set reference. Copy Storage The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit.

## STACK POINTER

- ❖ The Stack is especially used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack is implemented as growing from higher to lower memory locations. The Stack Pointer Register always points to the highest of the Stack. The Stack Pointer points to the info SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.
- ❖ The Stack within the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. Initial Stack Pointer value equals the last address of the interior SRAM and therefore the Stack Pointer must be set to point above start of the SRAM. See the table for Stack Pointer details .

## RESET AND INTERRUPT HANDLING

- ❖ The AVR provides several different interrupt sources. These interrupts and therefore the separate Reset Vector each have a separate program vector within the program memory space. All interrupts are assigned individual enable bits which must be written logic one along side the worldwide Interrupt Enable bit within the Status Register so as to enable the interrupt.
- ❖ counting on the Program Counter value, interrupts could also be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. rock bottom addresses within the program memory space are by default defined because the Reset and Interrupt Vectors.

- ❖ for program and data. Instructions within the program memory are executed with one level pipelining. While one instruction is being executed, subsequent instruction is pre-fetched from the program memory.
- ❖ This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory. The fast-access Register File contains 32 x 8-bit general purpose working registers with one clock cycle time interval .
- ❖ This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle. Six of the 32 registers are often used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers also can be used as an address pointer for search tables in Flash program memory.
- ❖ These added function registers are the 16-bit X-, Y-, and Z-register, described later during this section. The ALU supports arithmetic and logic operations between registers or between a continuing and a register. Single register operations also can be executed within the ALU. After an mathematical process , the Status Register is updated to reflect information about the results of the operation. Program flow is provided by conditional and unconditional jump and call instructions, ready to directly address the entire address space. Most AVR instructions have one 16-bit word format.
- ❖ Every program memory address contains a 16- or 32-bit instruction. Program non-volatile storage space is split in two sections, the Boot Program section and therefore the application section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the appliance non- volatile storage section must reside within the Boot Program section. During interrupts and subroutine calls, the address Program Counter (PC) is stored on the Stack.

- ❖ The Stack is effectively allocated within the general data SRAM, and consequently the Stack size is merely limited by the entire SRAM size and therefore the usage of the SRAM. All user programs must initialize the SP within the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible within the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported within the AVR architecture. The memory spaces within the AVR architecture are all linear and regular memory maps. A flexible interrupt module has its control registers within the I/O space with a further Global Interrupt Enable bit within the Status Register.
  - ❖ All interrupts have a separate Interrupt Vector within the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position.
  - ❖ The lower the Interrupt Vector address, the upper the priority. The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory are often accessed directly, or because the Data Space locations following those of the Register File, 0x20 - 0x5F.
  - ❖ In addition, this device has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions are often used.
  - ❖ ALU – Arithmetic Logic Unit The high-performance AVR ALU operates in direct reference to all the 32 general purpose working registers. Within one clock cycle, arithmetic operations between general purpose registers or between a register and an instantaneous are executed.
  - ❖ The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions.
  - ❖ Some implementations of the architecture also provide a strong multiplier supporting both signed/unsigned multiplication and fractional format. See Instruction Set Summary section for an in depth description.
- Status Register** The Status Register contains information about the results of the foremost recently executed arithmetic instruction.

- ❖ This information are often used for altering program flow so as to perform conditional operations. The Status Register is updated in any case ALU operations, as laid out in the Instruction Set Reference. This will in many cases remove the necessity for using the dedicated compare .
- ❖ The Status Register isn't automatically stored when entering an interrupt routine and restored when coming back from an interrupt. This must be handled by software. Status Register When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, leading to an I/O address offset within 0x00 - 0x3F.  
Name: SREG Offset: 0x5F Reset: 0x00 Property: When addressing as I/O Register: address offset is 0x3F.
- ❖ Global Interrupt Enable the worldwide Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers.
- ❖ If the worldwide Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is about by the RETI instruction to enable subsequent interrupts.
- ❖ The I-bit also can be set and cleared by the appliance with the SEI and CLI instructions, as described within the instruction set reference. Copy Storage The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit
- ❖ A bit from a register within the Register File are often copied into T by the BST instruction, and a touch in T are often copied into a touch in a

## MEMORY DESCRIPTION

- ❖ ATmega 328 has three types of memories e.g. EEPROM, SRAM etc. Flash Memory has 32KB capacity. It has an address of 15 bits. It is a Programmable Read Only Memory (ROM). It is non volatile memory. The ATmega328/P contains 32Kbytes On-chip In-System Reprogrammable non- volatile storage for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 16K x 16. For software security, the Flash Program memory space is split into two sections - Boot Loader Section and application Section within the device .

- ❖ The non-volatile storage has an endurance of a minimum of 10,000 write/erase cycles. The ATmega328/P Program Counter (PC) is 14 bits wide, thus addressing the 16K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described intimately in Boot Loader Support – Read- While-Write Self-Programming.
- ❖ Refer to Memory Programming for the outline on Flash data serial downloading using the SPI pins. Constant tables can be allocated within the entire program memory address space, using the Load Program Memory (LPM) instruction. Timing diagrams for instruction fetch and execution are presented in Instruction Execution Timing. 3.3.1 SRAM stands for Static Random Access Memory.
- ❖ It is a volatile memory i.e. data are going to be removed after removing the facility supply. Device SRAM Memory is organized. The device may be a complex microcontroller with more peripheral units than are often supported within the 64 locations reserved within the Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
- ❖ The lower 2303 data memory locations address both the Register File, the I/O memory, Extended I/O memory, and therefore the internal data SRAM. The first 32 locations address the Register File, subsequent 64 location the quality I/O memory, then 160 locations of Extended I/O

## REGISTERS

| 7   | 0 | Addr. |
|-----|---|-------|
| R0  |   | 0x00  |
| R1  |   | 0x01  |
| R2  |   | 0x02  |
| ... |   |       |
| R13 |   | 0x0D  |
| R14 |   | 0x0E  |
| R15 |   | 0x0F  |
| R16 |   | 0x10  |
| R17 |   | 0x11  |
| ... |   |       |
| R26 |   | 0x1A  |
| R27 |   | 0x1B  |
| R28 |   | 0x1C  |
| R29 |   | 0x1D  |
| R30 |   | 0x1E  |
| R31 |   | 0x1F  |

- ❖ ATmega-328 has thirty two (32) General Purpose (GP) registers.
- ❖ These all of the registers are the part of Static Random Access Memory (SRAM)
- ❖ address the internal data SRAM. The five different addressing modes for the info memory cover:
  - Direct – The direct addressing reaches the whole data space.
  - Indirect with Displacement – The Indirect with Displacement mode reaches 63 address locations from the bottom address given by the Y- or Z-register.
  - Indirect – within the Register File, registers R26 to R31 feature the indirect addressing pointer registers.
  - Indirect with Pre-decrement – The address registers X, Y, and Z are decremented.
  - Indirect with Post-increment – The address registers X, Y, and Z are incremented.
- ❖ The 32 general purpose working registers, 64 I/O Registers, 160 Extended I/O Registers, and therefore the 2K bytes of internal data SRAM within the device are all accessible through all these addressing modes.
- ❖ EEPROM stands for Electrically Erasable Programmable Read Only Memory. It has a long term data. The ATmega328/P contains 1K bytes of knowledge EEPROM memory. It is organized as a separate data space, during which single bytes are often read and written. The EEPROM has an endurance of a minimum of 100,000 write/erase cycles.

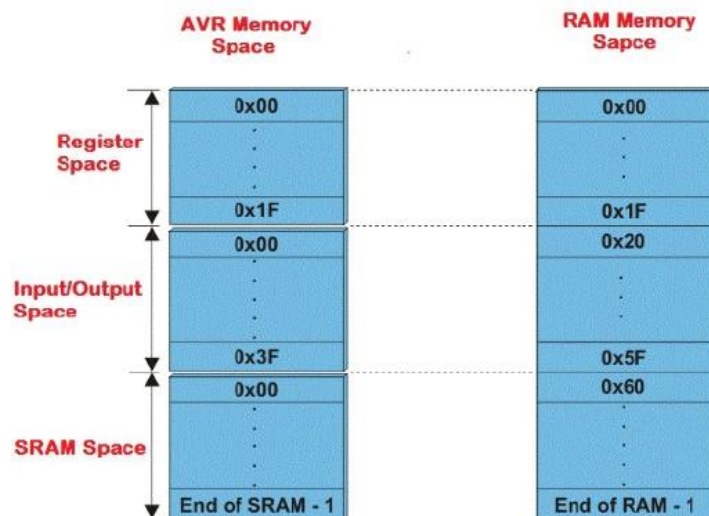


FIGURE 5.9.2 EEPROM

## 5.9.5 HARDWARE EXPERIMENTAL AND RESULTS



## 6. CONCLUSION:

- ❖ This Project discusses how RFID technology can be used to power intelligent vehicles. The RFID device alerts the driver when he or she is approaching the speed limit zone. If the driver is inattentive, the vehicle's speed can be held at a set limit without the driver's interference. This device was developed with the primary purpose of preventing accidents and reminding drivers of the speed limit in order to ensure safe driving. In hospitals, schools, and work zones, it is used to govern and regulate vehicle speed. Accidents that are caused by negligent driving or speeding by the driver can be avoided. As a result, many lives are saved.

## 6.2 REFERENCES

[1] Amulya A M,et.al., "Intelligent speed control system", Volume5, Issue 4, April 2018, pp.2537-2540.

[2] Vaishal B. Niranjane, et.al., "Automatic Vehicle Speed Control System Using Zigbee Technology", IJEECS ISSN 2348-117X Volume 6, Issue 3, March 2017.

[3] Amarnarayan, et.al., "Automatic Over speed Controlling of Vehicle", eISSN: 2321-225X, pISSN:2321-2241, Volume 5, Issue 5, May2016.

[4] Gummarekula Sattibabu, et.al., "Automatic Vehicle Speed Control With Wireless In-Vehicle Road Sign Delivery System Using ARM 7", Volume 2, Issue 8, pp.32-34.

[5] Amruta Ramase, et.al., "Automatic Speed Control of Vehicle Using RF Communication," pp.419-422.

[6] A Vengadesh, et.al., "Automatic Speed Control of Vehicle in Restricted Areas Using RF and GSM," Volume 02, Issue 09, December 2015, pp.875-877.

[7] Akash Batra, et.al., "Automatic Car Speed Control with Wireless RF Control", ISSN: 2277-9655, et al., 7(4): April 2018, pp.592-597.

[8] R. Ashok Kumar et.al., "A Beacon Based Automatic Vehicle Speed Control System for Restricted Zone", Volume 7, Issue X, October 2018, pp.1152-1159.

[9] Shankhavi K B, et.al., "Vehicle Speed Control using RF Communication", Volume 13, Issue 1, May 2016, ISSN: 2349 – 9303.

[10] R.Deepa. "Design of Vehicle Speed Control System Using Wireless Instrument Cluster", Volume 4, Issue 1, January 2015.

[11] Sunil R Kewate, et.al., "Automatic speed control system by the color sensor for automobiles – An innovative modelbased approach," ISSN 2250-3234, volume 4, number 2 , pp. 223-230.

[12] Dr.K.S.Tamilselvan, et.al., "Android Based Vehicle Speed Control System In Critical Zone Using GPS Technology", volume 7, issue 6, June 2018, pp.639-644.

[13] K.Govindaraju, et.al., "Embedded Based Vehicle Speed Control System Using Wireless Technology," Volume 2, Issue 8, August 2014, pp.1841-1844.

14. Gopal P. Gawande, et.al., "Review of Speed Control and Automatic Braking System", Volume 3, Issue 2, February 2014, pp.474-476.

15. Gangadhar, S ; R N shetty Inst. Of Techno, i"ntelligent road traffic Control system," IEEE conference Publication, kahargpur (2010).

16. Berndt, Don, "Real-Time Embedded Technology and Applications Symposium "(RTAS), IEEE Education & Learning (2005).
17. Kassem, N. Microsoft Corp., Redmond, WA, USA Kosba, A.E.; Yousuf, M.; "VRF-Based Vehicle Detection and Speed Estimation vehicular Technology Conference "(VTC Spring), IEEE (2012).
18. Telaprolu, m.k, sarma, V.V.; ratankanth, E.K.o, S. N.; Banda, v., "vehicular Electronics and safety(ICVES)", IEEE international conference pune (2009).
19. Baris\_S\_ims\_ek, Fatma Pakdil, Berna Dengiz, Murat Caner Testik, "Driver performance appraisal using GPS terminal measurements: A conceptual framework", Transportation Research, pp: 49-60, 2013.
20. J.W. Bolderdijk, J. Knockaert, E.M. Steg, E.T. Verhoef, "Effects of Pay-As-You-Drive vehicle insurance on young drivers speed choice: Results of a Dutch field experiment", Accident Analysis and Prevention, Elsevier Ltd, pp: 1181-1186, 2011.
21. Harry Lahrman, Niels Agerholm, Nerijs Tradisauskas, Teresa Naessa, Jens Juhla, Lisbeth Harms, "Pay as You Speed, ISA with incentives for not speeding: A case of test driver recruitment", Accident Analysis and Prevention , Elsevier Ltd, pp: 10-16, 2011.
22. Nourdine Aliane, Javier Fernández, Sergio Bemposta and Mario Mata, "Traffic Violation Alert and Management", 14th International IEEE Conference on Intelligent Transportation Systems, pp: 1716 - 1720, 2011.
23. Rajesh Kannan Megalingam, Vineeth Mohan, Paul Leons, Rizwin Shooja, Ajay M, "Smart Traffic Controller using Wireless Sensor Network for Dynamic Traffic Routing and Over Speed Detection", IEEE Global Humanitarian Technology Conference (GHTC), pp: 528 - 533, 2011.

24. Turgay Celik and Huseyin Kusetogullari, "Solar-Powered Automated Road Surveillance System for Speed Violation Detection", IEEE Transactions on Industrial Electronics, pp: 3216 - 3227, 2010.

[25] Vladimir Glavtchev, Pinar Muyan-Ozcelik, Jeffrey M. Ota, and John D. Owens, "Feature-Based Speed Limit Sign Detection Using a Graphics Processing Unit", IEEE Intelligent Vehicles Symposium, pp: 195 - 200, 2011.