IOT Road Safety

PROJECT REPORT

DEPARTMENT OF COMPUTER SCIENCE

Contents

1 Introduction 1
- Overview 1
- Project Profile 1
- Contributions 1

2 Problem Definition and Methodology 2
- Problem definition 2
- Objectives 2
- Motivation 2
- Methodology 2
- Scope 3

3 Requirement Analysis and Specification 4
- Requirement Analysis 4
- Existing System 4
- Proposed System 4
- Requirement Specification 4
  - Functional Requirements 4
  - Non-Functional Requirements 5
- Feasibility Study 5
  - Technical Feasibility 5
  - Economical Feasibility 5
  - Operational feasibility 5
- Project Planning and Scheduling 5
  - Gantt Chart 5
- Software Requirement Specification 6
  - Introduction 6
  - Overall Description 7
  - Design and Implementation Constraints 8
  - Assumptions and Dependencies 8
  - External Interface Requirements 8
  - System Features 9
  - Other Nonfunctional Requirements 9

4 System Design 10
- Users of the System 10
- Modularity criteria 10
- Design Methodologies 10
- Architecture diagrams 11
  - Architecture 11
- User Interface Layouts 11

5 Implementation 12
- Tools/Scripts for Implementation 12
  - Blynk 12
  - ArduinoDroid 12
- Module hierarchy 12
- Coding 13
- Problems Encountered 13

6 Testing And Implementation 14
- Test Plans 14
- Unit testing 14
- Integration testing 14
- System testing 14
- Implementation - Changeover Plans 14

7 Conclusion 15

8 Bibliography 16

Appendix 17

A Circuit Diagram 17

B Code 18

C Hardware Design 23

List of Figures

1 Working Diagram-Control Side 11
2 Working Diagram-Road Side and Other Vehicle 11
3 Vehicle Side 17
4 Control Side 17
5 Another Vehicle 18
6 Setting up of control side and vehicle side units 23
7 Blynk app activation 24
8 Connection of control side and vehicle side units by Blynk app 25

INTRODUCTION

Chapter 1

1 Introduction

Overview

Technology has brought fine changes into every portion of our life by making it smart and reliable. There are many situations in which technologies can be used to avoid accidents in roads which opens a wide window for the requirement of Smart Road System. With the dynamic changes in the models of the vehicles the roads need to have same ability to face them. Evolving towards the future, the roads needs to build with advanced sensors and antenna systems to have a pace with the new era. The design involves the road side units and vehicle side units as part of intelligent transport system involving Internet of things(IOT).

This project has desighned a system to alert the driver about the speed limits in specific areas by reducing the speed of the vehicles in sensitive public zones without any interference of the drivers where controls are taken automatically by the use of a wireless local area network. The main objective of the proposed system is to operate the vehicles in a safe speed at critical zones minimizing the possible risk of unwitting accidents and casualties. Besides, the system is capable of detecting the accidents and give notification to the control room. The system operates in such way that the accident information is passed to the vehicles entering the same zone to take diversion to avoid traffic congestion.

The basic steps of this system are:

- Block and circuit preparation
- Hardware Implementation
- Setting up IOT

Project Profile

- **Title** :IOT Road Safety
- **Domain** :Embedded System
- **Language** :C
- **Library** :TinyGPS,ESP8266WIFI and Blynk

Contributions

The major contributions of this system are:

- Create a system which gives alert about the speed limits in specific areas.
- Create control side and vehicle side units whose controlls will be taken by a wireless area network.

PROBLEM DEFINITION AND METHODOLOGY

Chapter 2

2 Problem Definition and Methodology

Problem definition

The early effects to prevent road accidents and to ensure road safety includes the use of speed detection devices,CCTVs,speed limiters and emergency accident units.Old approaches emphasize the concept of problem-solving in Road safety, but it is more correct to recognize that Road safety activities doesn't solve problems. For instance, when a safer road design is implemented, hopefully the number of crashes, or their seriousness, will go down, but they will not disappear. It is more correct to say the implementation of correct policies, programs and measures will reduce numbers or consequences of crashes, but they will no be solved.This realization is important, because it changes the focus from a problem that will go away if we devote enough resources to it, to a situation requiring on-going management. This management in turn requires the development of scientifically based techniques, witch will enable us to predict with confidence that safety resources are well-spent and likely to be effective.The standard measures used in assessing road safety interventions are fatalities and killed or seriously injured (KSI) rates, usually per billion (109) passenger kilometres.Vehicle speed within the human tolerances for avoiding serious injury and death is a key goal of modern road design because impact speed affects the severity of injury to both occupants and pedestrians.

Objectives

The main objective of the proposed system is to operate the vehicles in a safe speed at critical zones minimizing the possible risk of unwitting accidents and casualties.

Motivation

This project paves a system to alert the driver about the speed limits in specific areas and reduce the speed of the vehicles in sensitive public zones without any interference of the drivers where controls are taken automatically by the use of a wireless local area network. These systems can be used to overcome some problems such as,

- Over Speed
- Exact location of the accident occured
- Heavy traffic congestion

Methodology

IoT is used in our system with Arduino Code and Node MCU controller with Blynk app .ArduinoDroid is a open source platform as a combination of software and hardware and Blynk server is used to control the communications between smartphone and hardware.

- ArduinoDroid
- Blynk Server

Scope

Connected technologies and the IoT improves in:

- vehicle safety
- efficiency
- convenience
- overall customer experience
- operational performance

REQUIREMENT ANALYSIS AND SPECIFICATION

Chapter 3

3 Requirement Analysis and Specification

Requirement Analysis

Requirement analysis results in the specification of operational characteristics of software: indicates interface of software with other system elements and establishes constrains the software must meet. Requirement analysis allows the software engineer to elaborate on basic requirement established during earlier requirement engineering tasks and build models that depict user scenarios, functional activities, problem classes and their relationships, system and class behavior and flow of data as it is transformed.

Existing System

The early effects to prevent road accidents and to ensure road safety includes the use of speed detection devices,CCTVs,speed limiters and emergency accident units as the first phase.Despite achieving the state-of-the-art performance, the existing systems suffer from two main problems,

- Over Speed : These systems cannot control speed at some specific zones.
- Exact location of accident occured: These systems cannot give the precise location of accident .

Proposed System

I evaluate the proposed Road Safety system with the IoT connected technology.

- I propose to use node MCUs connected with IoT which controls and limits the speed in some specific areas and GPS to get the exact location where the accident may occur.

Requirement Specification

Functional Requirements

In software engineering and system engineering, functional requirement defines function of a system and its components. A function is described as a set of inputs, the behavior and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form ”system must do requirement”, while non-functional requirements are ”system shall be requirement”. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with nonfunctional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system. This system does:

- Any vehicle entering the network zone cannot overcome the speed limit defined by the system and the controls will be automatically taken by the use of a wireless local area network.

Non-Functional Requirements

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing nonfunctional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements. Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of ”system shall do requirement”, an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of ”system shall be requirement”, an overall property of the system as a whole or of a particular aspect and not a specific function. The system’s overall properties commonly mark the difference between whether the development project has succeeded or failed. Non-functional requirements are often called ”quality attributes” of a system. Other terms for non-functional requirements are ”qualities”, ”quality goals”, ”quality of service requirements”, ”constraints” and ”non-behavioral requirements”.. Qualities—that is non-functional requirements—can be divided into two main categories: Execution qualities, such as safety, security and usability, which are observable during operation (at run time). Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

Feasibility Study

Technical Feasibility

Technical feasibility assesses the current resources (hardware and software) and technologies, which are required to accomplish user requirements. Today every organization has computer, so it is not an extra cost.

Economical Feasibility

Economic feasibility is the most frequently used method for evaluating the effectiveness of proposed system.The proposed model is cost effective.

Operational feasibility

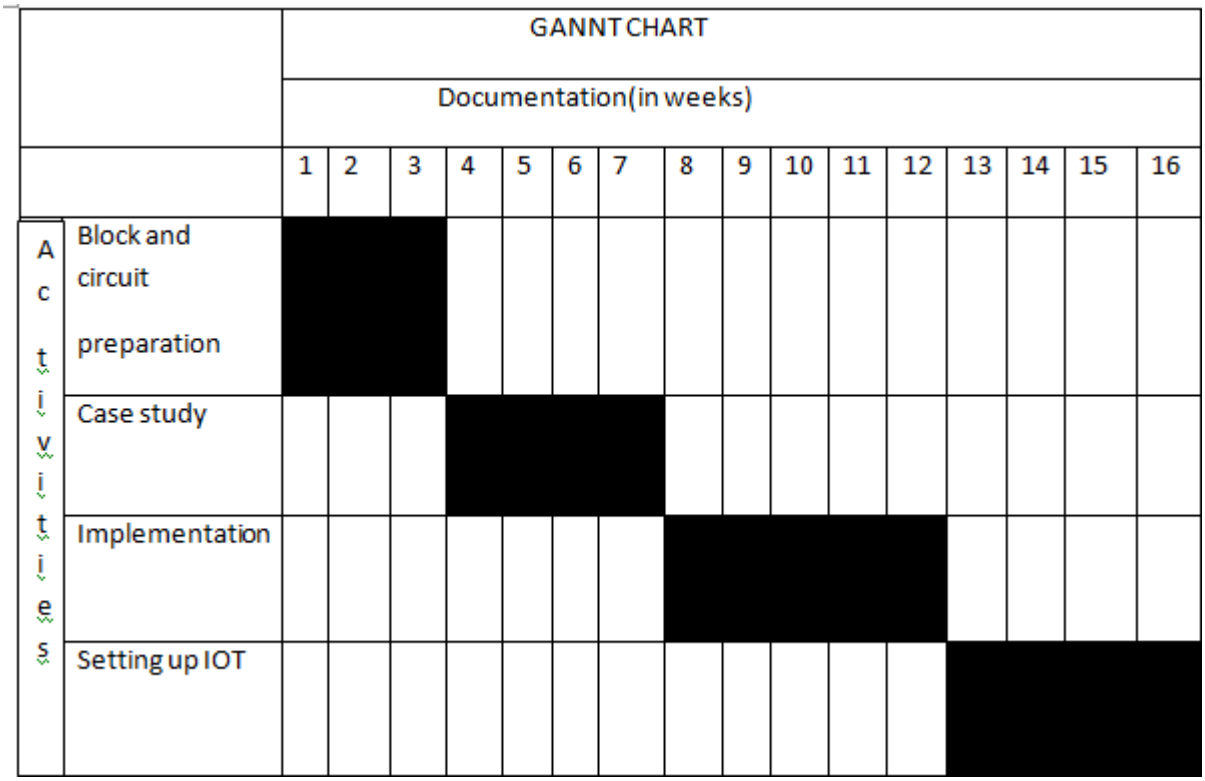
The proposed system performs effective than the existing system.This system controls speed automatically when they reach at some specific areas with the help of local area network.

Project Planning and Scheduling

3.6.1 Gantt Chart

GANTT chart is a graphical representation of the project. This chart shows the start and end dates of each activity in the project. It shows week, month or quarter required to complete each activity. It is also known as timeline chart. It shows information about activities in the form of horizontal bars. It is prepared on graph paper. In case if it is complex it is prepared using application such as Microsoft Excel.

In the First Phase of our project we want to prepare block diagram and circuit diagram of the system.It takes three weeks.After that we enter to case study and it tooks 5 weeks to complete.Third part is for implementation and this phase tooks four weeks to complete and the



predicting model.After that final stage, setting the IoT the system takes four weeks to complete , in this we test various speed fordifferent zones and how controls are made possible.

Software Requirement Specification

Introduction

Purpose

The purpose of this document is to provide a debriefed view of requirements and specifications of the project called IOT Road Safety.

•The goal of this project is to operate the vehicles in a safe speed at critical zones minimizing the possible risk of unwitting accidents and casualties.

Document Conventions

- All terms are in Times New Roman style.
- Main features or important terms are in bold.
- Use LateX for documentation.

Intended Audience and Reading Suggestions

Anyone with some programming experience, with familiarity in C language and IOT, can understand this document.The document is intended for developers, software architects, testers, project managers and documentation writers. This Software Requirement Specification also includes:

- Overall description of the product
- External interface requirements
- System Features
- Other nonfunctional requirements

Product Scope
IOT Road Safety, which aims to minimise accident rates. IoT Road Safety helps in problems such as,

- vehicle safety
- efficiencyl
- convience
- overall customer experience
- operational performance

References

- IEEE Standard 830-1998 Recommended Practice for Software Requirements Specifications.

Overall Description

Product Perspective
The early effects to prevent road accidents and to ensure road safety includes the use of speed detection devices,CCTVs,speed limiters and emergency accident units as the first phase.Despite achieving the state-of-the-art performance, the existing systems suffer from two main problems,

- Over Speed
  These systems cannot control speed at some specific zones.
- Exact location of accident occured
  These systems cannot give the precise location of accident.

The IoT Road Safety System will give solution for these problems .
Product Functions

- Block and Circuit preparation
- Hardware Implementation
- Setting up IoT

User Classes and Characteristics

- Accident control authorities
- Customers

Operating Environment

- Operating System: Windows 8
- Processor: Intel I3 or Higher
- Memory: 4GB or more

Design and Implementation Constraints

- Control Side and Vehicle side units
- Bluetooth transmitter and receiver
- GPS

Assumptions and Dependencies

Assumptions

The vehicles will operate in a safe speed at critical zones minimizing the possible risk of unwitting accidents and casualties. Besides, the system is capable of detecting the accidents and give notification to the control room. **Dependencies**

- Node MCUs
- Blynk Server

External Interface Requirements User

Interfaces

Once accident occurs, piezoelectric sensors in vehicle unit get activated and send the information along with GPS location (When any accident occur, the system takes the location data using GPS Modem) to control room through IoT communication.Alert message will be displayed in the LCD unit at the vehicle

We use Blynk App for GUI. It is designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data and , vizualize it .

Hardware Interfaces

- Operating System: Windows or any other Platform

- Hardware: Intel I5
- Internet Connection

Software Interfaces

- Arduinoid
- Blynk Server

Communications Interfaces
Standard HTTP COMMUNICATION interface required for internet connection.

System Features System

Feature 1 : Alerts
• Description This system will alert the driver about the speed limits in specific areas by reducing the speed of the vehicles in sensitive public zones without any interference of the drivers where controls are taken automatically by the use of a wireless local area network. Functional Requirements Any vehicle entering the network zone cannot overcome the speed limit defined by the system.

Other Nonfunctional Requirements

Performance Requirements

- Quickness: System should be fast enough for getting alerts.
- Failure Handling: In case of failures due to unavoidable reasons, the system should be able to recover quickly.
- Detection and Response time: High (According to the processor).
- Typical throughput required: model needs to update.

SYSTEM DESIGN

Chapter 4

4 System Design

Users of the System

User: The user who handles the System.

Modularity criteria

The proposed system has following modules :

- Block and Circuit preparation
- Hardware Implementation
- Setting up IoT

Design Methodologies

ArduinoDroid
ArduinoDroid is a free app that will let you edit, compile and upload sketches to your Arduino board directly from an Android phone or tablet.They have following features,

- They works offline
- They compile sketches (no root required) and upload sketches

Currently only Arduino Uno r3 and the FTDI-based boards (Duemilanove, Diecimila) are supported but support for the Mega, Due and Nano is planned. Other features to be implemented include advanced configurable syntax highlighting, smart code suggestions (autocomplete) and 3rd-party apps integration (manuals, compile sketch requests).

Blynk Server
Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things. The server responsible for all the communications between the smartphone and hardware.

Architecture diagrams

4.4.1 Architecture

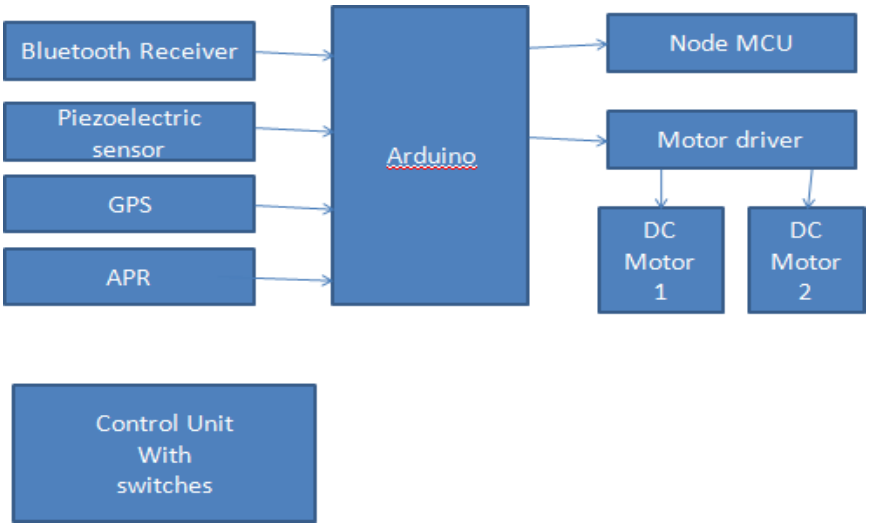


Figure 1: Working Diagram-Control Side

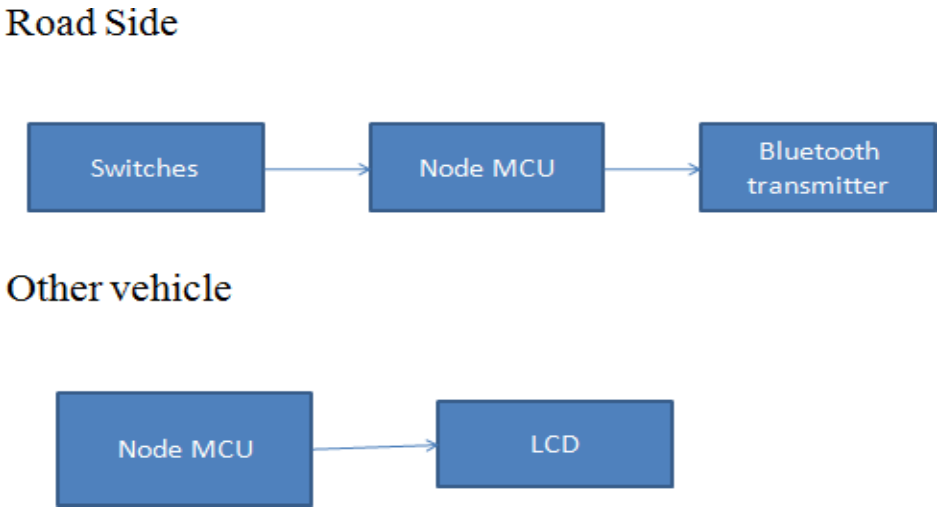


Figure 2: Working Diagram-Road Side and Other Vehicle

User Interface Layouts

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the Interface.The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

IMPLEMENTATION AND MAINTENANCE

Chapter 5

5 Implementation

Tools/Scripts for Implementation

Blynk

Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

ArduinoDroid

ArduinoDroid is a free app that will let you edit, compile and upload sketches to your Arduino board directly from an Android phone or tablet. It also needs an Android device with USB-host support.

Sometimes we don't have any PC or laptop to program our Arduino boards. We can still program it using our Android mobile, Thanks to OTG (On the Go) adaptor. You might have used OTG adaptor for connecting Pendrives and game controllers, and give power to small devices.

Module hierarchy

- **Block and Circuit preparation**
  A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks..A circuit diagram (electrical diagram, elementary diagram, electronic schematic) is a graphical representation of an electrical circuit. A pictorial circuit diagram uses simple images of components.Here we have Node MCU,GPS,APR module and bluetooth transmitter and receiver

- **Hardware Implementation**
  Road side unit transmit the speed limit in the area using bluetooth transmitter.Speed limit information from road side unit is received by Vehicle unit using bluetooth receiver and produce a voice announcement for alerting the driver using APR module According to the zonal details the vehicle speed will be reduced automatically controlling the motors for navigation.Once accident occurs, piezoelectric sensors in vehicle unit get activated and send the information along with GPS location (When any accident occur, the system takes the location data using GPS Modem) to control room through IoT communication.Control room send the alert to the road side unit using IoT communication.Road side transmit the accident information through bluetooth to the vehicles entering the accident zone to take diversion to avoid traffic congestion.Alert message will be displayed in the LCD unit at the vehicle

- **Setting up IoT**
  The system will alert the driver about the speed limits in specific areas and reduce the speed of the vehicles in sensitive public zones without any interference of the drivers where controls are taken automatically by the use of a wireless local area network. Any vehicle entering the network zone cannot overcome the speed limit defined by the system. The system is capable of detecting the accidents and give notification to the control room.

  The system operates in such way that the accident information is passed to the vehicles entering the same zone to take diversion to avoid traffic congestion.

Coding

- **ArduinoDroid**
  ArduinoDroid is a free app that will let you edit, compile and upload sketches to your Arduino board directly from an Android phone or tablet.It also needs an Android device with USB-host support.

Problems Encountered

The following are some of the problems faced in this system:

- The system takes more time because of short range communication.
- Latency

TESTING
AND
IMPLEMENTATION

Chapter 6

6 Testing And Implementation

Test Plans

A test plan documents strategy that will be used to verify and ensure that a product or system meets its design specification and other requirements. A test plan is usually prepared by or with significant input from the engineer.This document describes the plans for testing the architectural prototype of System.

In my Project the system has to be tested to get the Desired Output.I use different speed for testing the system.

Unit testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. In our system,

- Test to check whether block and circuit diagrams are well designed.
- Test to check whether hardware implementation work properly.
- Test to check whether the IoT connections are guaranteed.

Integration testing

Integration testing (sometimes called integration and testing) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

- Check whether the system limits the speed in specific areas.
- Check whether the system gives alerts.
- Check whether the controls are taken by a wireless local system.

System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

Implementation - Changeover Plans

- Provide the model with variety speed limits.

CONCLUSIONS
AND
FUTURE WORKS

Chapter 7

7 Conclusion

We have presented a system, to alert the driver about the speed limits in specific areas and reduce the speed of the vehicles in sensitive public zones without any interference of the drivers where controls are taken automatically by the use of a wireless local area network. In the initial phase, we designed the basic block and circuit diagram for the system. In the implementation phase, we executed the hardware with the help of IoT connecting tecnologies such as Blynk app. Extensive experiments conducted on IoT and other connecting technologies.

Future enhancement

We can be enhanced this system by implementing camera using Raspberri pi, GSM module in case of network unavailability and low RAM module/zigbee module for long range communication.

REFERENCES

8 Bibliography

- Software Engineering a practitioners approach by Roger S.Pressman
- IoT-driven road safety system by Dasari Vishal, H. Saliq Afaque, Harsh Bhardawaj and T. K. Ramesh
- Programming with Arduino by Hans-Petter Halvorsen
- www.github.com
- www.stackoverflow.com
- www.codeacademy.com

APPENDIX

A Circuit Diagram

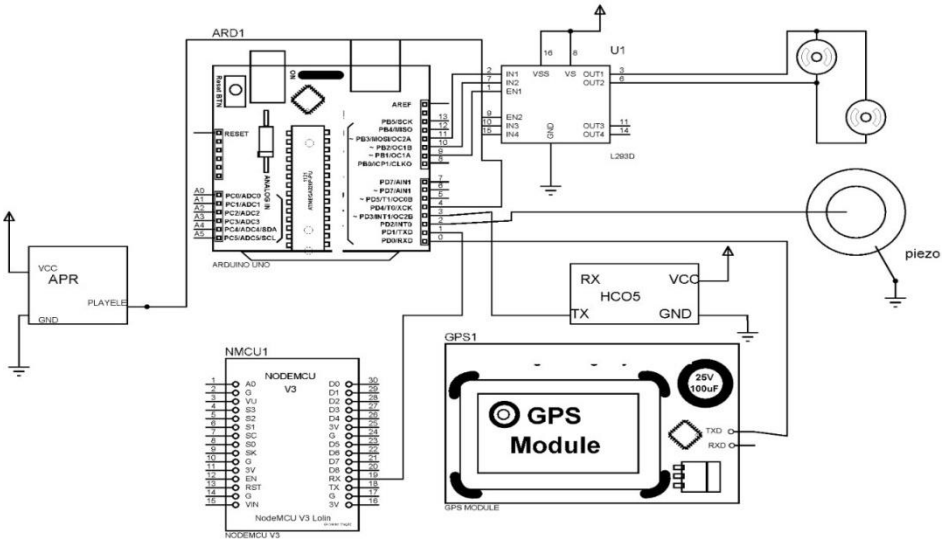


Figure 3: Vehicle Side

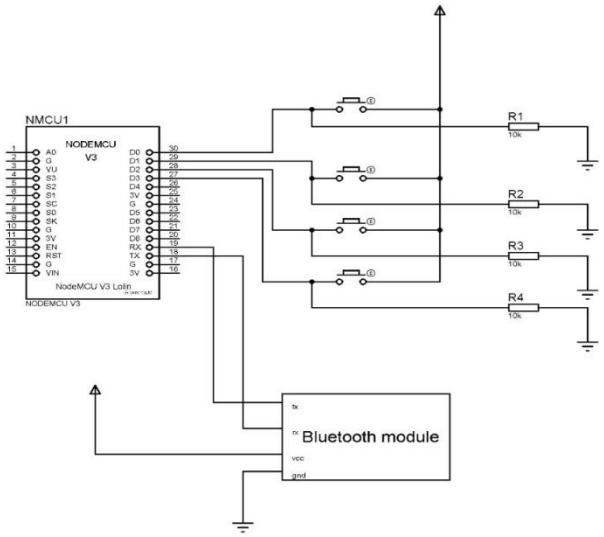


Figure 4: Control Side

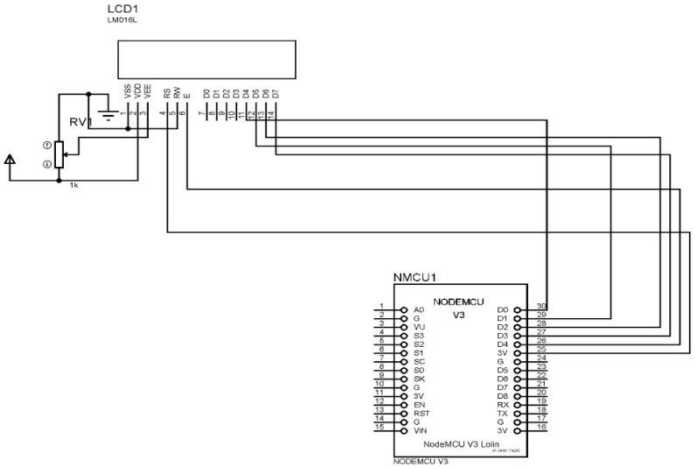


Figure 5: Another Vehicle

B Code

```
#include <LiquidCrystal.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include<BlynkSimpleEsp8266.h>

char auth[] = "Y4DsRJfvnDUee9LMZHCWT7pdmVLolRGU";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "hellow";
char pass[] = "12345678";
const int rs = D5, en = D6, d4 = D1, d5 = D2, d6 = D3, d7 = D4;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int a=0;
BLYNK_WRITE(V2)
{
  a=param.asInt();
}
void setup() {
  // set up the LCD's number of columns and rows:
     Serial.begin(9600);
       Blynk.begin(auth, ssid, pass);
  lcd.begin(16, 2);

}

void loop() {
 Blynk.run();
 lcd.setCursor(0, 1);
 if(a==1)
 {
```

```
   lcd.print("accident occured");
   Serial.print("hi");
  }
  else
  {
   lcd.print("welcome");
  }
}

#define BLYNK_PRINT  Serial


#include <ESP8266WiFi.h>
#include  <BlynkSimpleEsp8266.h>
char auth[] = "q6FAQlggdlxznS2kMIbxAPn8E6nnv116";
char ssid[] = "hellow";
char  pass[] = "12345678";

String str;
void setup() {
 Serial.begin(9600);
Blynk.begin(auth, ssid, pass);

}

void loop() {
   Blynk.run();
if(Serial.available()>0)
{
   str=Serial.readStringUntil('/n');
 // Serial.print(str);
 // Blynk.notify("location:");
   Blynk.notify(str);
}

}

#include <TinyGPS++.h>
#include <SoftwareSerial.h>
TinyGPSPlus gps;
SoftwareSerial   ss(3,4);
char  n;
int  a;
void setup() {
 Serial.begin(9600);
 ss.begin(9600);
    pinMode(2, INPUT);
   pinMode(6,OUTPUT);
   pinMode(11,OUTPUT);
   pinMode(10,OUTPUT);
   pinMode(9,OUTPUT);
   pinMode(12,OUTPUT);//apr
   digitalWrite(11,HIGH);
   digitalWrite(6,HIGH);
   attachInterrupt(digitalPinToInterrupt(2),piezo,     CHANGE);
}
```

```
void loop() {
   n=Serial.read();
//Serial.println(" ");
   delay(200);
 if(n==’3’)
 {
 digitalWrite(6,HIGH);
 digitalWrite(11,HIGH);
 digitalWrite(12,HIGH);
 delay(200);
 digitalWrite(12,LOW);
 }
 else if(n==’2’)
 {
 digitalWrite(6,LOW);
 digitalWrite(11,LOW);
 digitalWrite(10,LOW);
 digitalWrite(9,LOW);
 digitalWrite(12,HIGH);
 delay(200);
 digitalWrite(12,LOW);

 }
 else if(n==’1’)
 {
 analogWrite(11,100);
 analogWrite(6,100);
 digitalWrite(12,HIGH);
 delay(200);
 digitalWrite(12,LOW);

 }

}

 // while (ss.available() > 0)
 //    if (gps.encode(ss.read()))
    //    displayInfo();


void displayInfo()
{
 // Serial.print(F("Location: "));
  if (gps.location.isValid())
  {
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(","));
    Serial.print(gps.location.lng(), 6);
  }
  else
  {
   // Serial.print(F("INVALID"));
   Serial.print("10.305125");
   Serial.print(’,’);
   Serial.print("76.389582");

  }
```

```
 /*  Serial.print(F("   Date/Time: "));
  if (gps.date.isValid())
  {
    Serial.print(gps.date.month());
    Serial.print(F("/"));
    Serial.print(gps.date.day());
    Serial.print(F("/"));
    Serial.print(gps.date.year());
  }
  else
  {
    Serial.print(F("INVALID"));
  }

  Serial.print(F("  "));
  if (gps.time.isValid())
  {
    if (gps.time.hour() < 10) Serial.print(F("0"));
    Serial.print(gps.time.hour());
    Serial.print(F(":"));
    if (gps.time.minute() < 10) Serial.print(F("0"));
    Serial.print(gps.time.minute());
    Serial.print(F(":"));
    if (gps.time.second() < 10) Serial.print(F("0"));
    Serial.print(gps.time.second());
    Serial.print(F("."));
    if (gps.time.centisecond() < 10) Serial.print(F("0"));
    Serial.print(gps.time.centisecond());
  }
  else
  {
   // Serial.print(F("INVALID"));

  }*/

  Serial.println();

}
void piezo()
{
   while (ss.available() > 0)
    if (gps.encode(ss.read()))
      displayInfo();
}

int a=0,b=0,c=0,d=0;
void setup() {
  pinMode(D1,INPUT);
   pinMode(D2,INPUT);
   pinMode(D3,INPUT);
      pinMode(D4,INPUT);
digitalWrite(D1,LOW);
digitalWrite(D2,LOW);
digitalWrite(D3,LOW);
digitalWrite(D4,LOW);
Serial.begin(9600);
```

```
}

void  loop()
{
 a=digitalRead(D1);
if(a==1)
 {
  Serial.print("1");
 }
b=digitalRead(D2);
 if(b==1)
 {
  Serial.print("2");
 }
 d=digitalRead(D4);
 if(d==1)
 {
  Serial.print("3");
 }
}
```

C Hardware Design

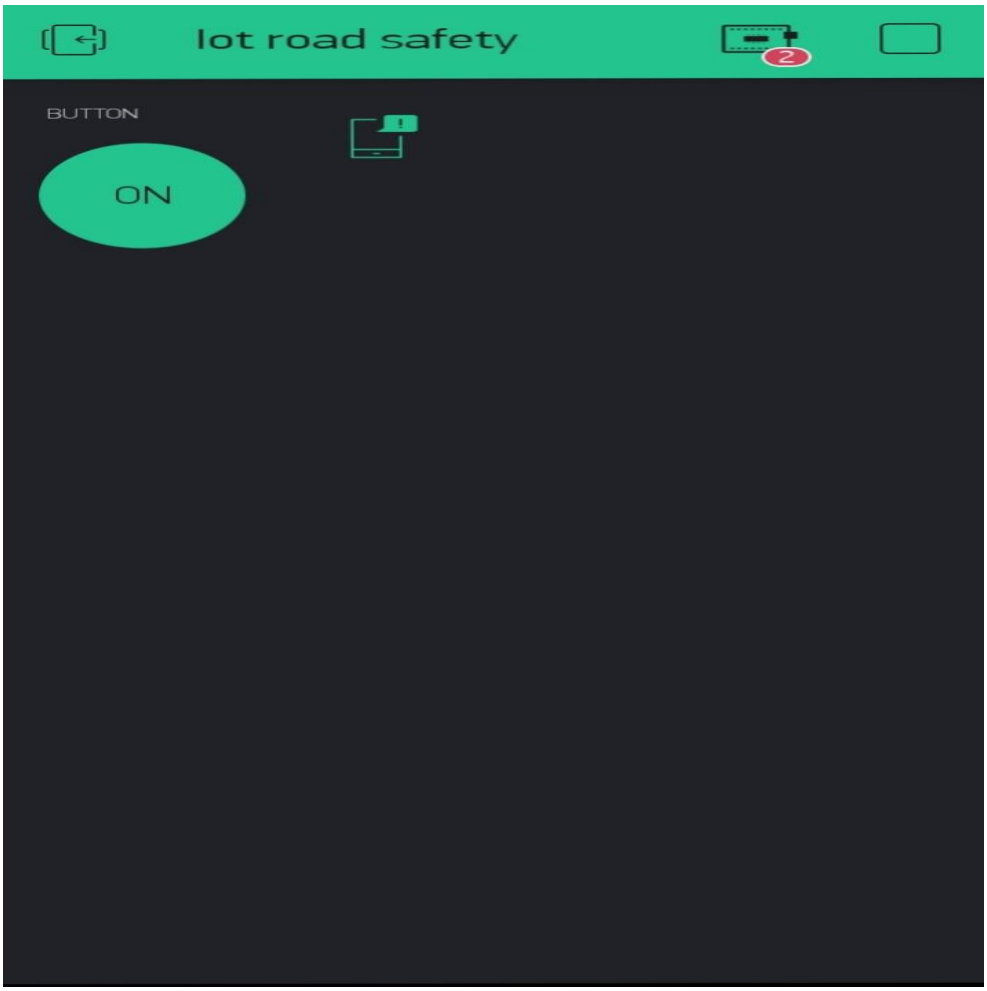


Figure 6: Setting up of control side and vehicle side units

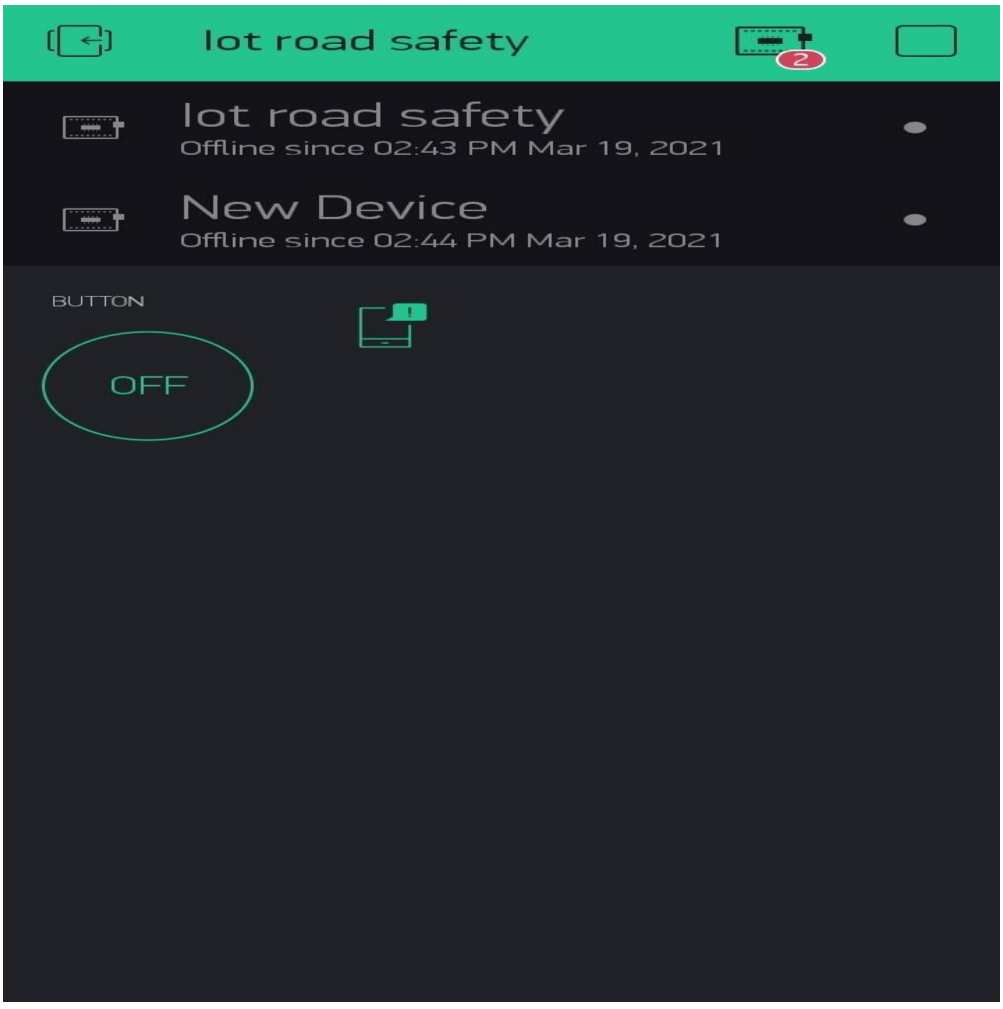


Figure 7: Blynk app activation

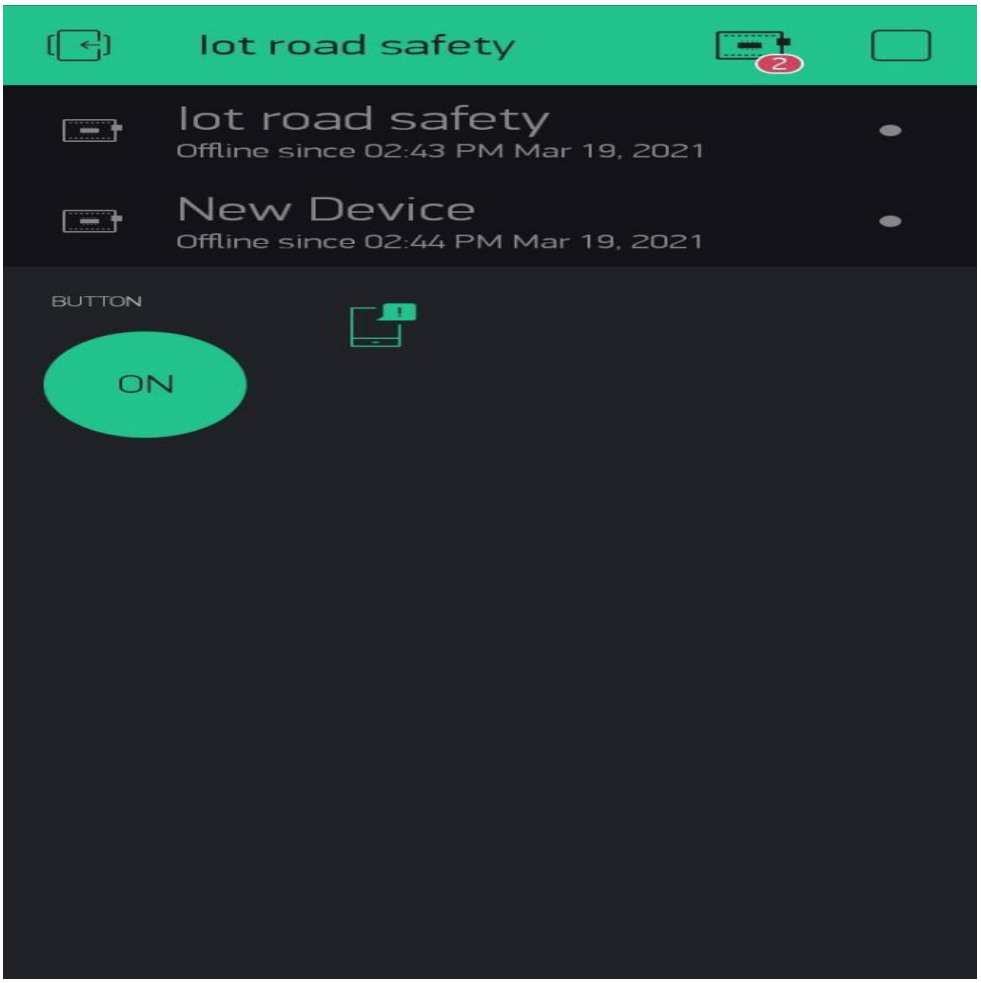


Figure 8: Connection of control side and vehicle side units by Blynk app