SNL Mini Project

Traffic Management for Emergency Vehicles in a City

Shrineeth Kotian - XIEIT181925
Manish Kumavat - XIEIT181926
Dixit Patel -XIEIT181936

May 17, 2021

List of Figures

Contents

INTRODUCTION TO WIRELESS SENSOR NETWORKS (LO1)

INTRODUCTION:

1.1 IDENTIFY THE REQUIREMENTS FOR THE REAL WORLD PROBLEMS:

1.1.1 Wireless Sensor Network (WSN):

Wireless Sensor Network (WSN) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions.

Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System. Base Station in a WSN System is connected through the Internet to share data.

1.1.2 Characteristics of Wireless Sensor Network:

The characteristics of WSN include the following :
a) The consumption of Power limits for nodes with batteries
b) Capacity to handle node failures
c) Some mobility of nodes and Heterogeneity of nodes
d) Scalability to a large scale of distribution
e) Capability to ensure strict environmental conditions
f) Simple to use
g) Cross-layer design

1.1.3 Advantages of Wireless Sensor Networks:

The advantages of WSN include the following :
a) Network arrangements can be carried out without immovable infrastructure.
b) Apt for the non-reachable places like mountains, over the sea, rural areas, and deep forests.
c) Flexible if there is a casual situation when an additional workstation is required.
d) Execution pricing is inexpensive.
e) It avoids plenty of wiring.
f) It might provide accommodations for the new devices at any time.

g) It can be opened by using centralized monitoring.

1.1.4 Wireless Sensor Network Applications:

Wireless sensor networks may comprise numerous different types of sensors like low sampling rate, seismic, magnetic, thermal, visual, infrared, radar, and acoustic, which are clever to monitor a wide range of ambient situations. Sensor nodes are used for constant sensing, event ID, event detection & local control of actuators. Wireless sensor networks have been useful in a variety of areas. The primary areas in which these networks are deployed follows :

1. Environmental observations : Wireless sensor networks can be used to monitor environmental changes. An example is water pollution detection in a lake that is located near a factory that uses chemical substances.

   Sensor nodes are randomly deployed in unknown and hostile areas and relay the exact origin of a pollutant to a centralized authority which takes appropriate measures to limit the spreading of pollution.

   Other examples include forest fire detection, air pollution, and rainfall observation in agriculture.

2. Military monitoring : The military uses sensor networks for battlefield surveillance.
   Sensors can monitor vehicular traffic, track the position of the enemy, or even safeguard the equipment of side deploying sensors.

3. Building monitoring : Wireless sensor networks can also be used in large buildings or factories to monitor climate changes.
   Thermostats and temperature sensor nodes are deployed all over the building's area. In addition, sensors can be used to monitor vibration that can damage the structure of a building.

4. Health care : Sensors can be used in biomedical applications to improve the quality of provided care.
   Sensors are implanted in the human body to monitor medical problems such as cancer and help patients maintain their health.

5. Agriculture and Farming : The employment of WSNs has been reported assist farmers in various aspects such as the maintenance of wiring in a problematic environment, irrigation mechanisation which aids more resourceful water use and reduction of wastes.

INTRODUCTION TO TRAFFIC MANAGEMENT FOR EMERGENCY VEHICLES IN A CITY (LO2)

2.1 PROBLEM DEFINITION:

There is the need for detailed approach towards handling prioritized vehicles (also known as emergency vehicles) like Fire Truck, Ambulance, Police Vans, etc by taking into consideration all possible scenarios from best case to worst case.
In our proposed system we offer a solution to tackle the worst case (deadlock) condition for management of emergency vehicles.Our Project have proposed a RFID-based system, which manages and regulates the traffic signals at junctions when the emergency vehicle approaches, by allowing the straight forward passage out of the traffic congestions. In our system, RFID receiver present in the Traffic Signal detects the emergency vehicles with the help of RFID tag attached to emergency vehicle, and the emergency light turn on at the junction to alert everyone present over there. And then the signal turns green for the road from where the emergency vehicle is coming, rest all the paths are blocked to avoid any obstacle to emergency vehicle. Also the ambulance or any emergency vehicle, it is tracked by using GPS. This location is send to the application. This GPS System will send the coordinates of the vehicle at every moment to application server.

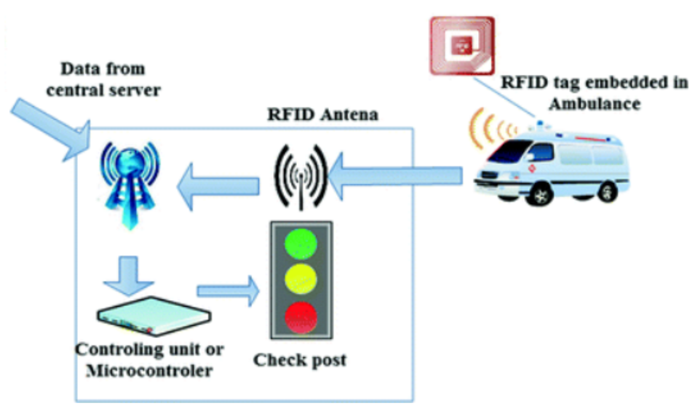

Figure 2.1: Traffic Management for Emergency Vehicles

2.2 AIMS AND OBJECTIVES:

A) To prevent emergency vehicles from being stuck in an unnecessary traffic.
B) To help the emergency vehicles to reach their destination within less time.
C) To be precise in providing location of the emergency vehicles with respect to time.
D) To consider the data delivery rate while monitoring and detecting emergency vehicles.
E) To conserve the network energy for lifetime better monitoring.

2.3 SCOPE OF THE PROJECT:

This project can be used to monitor the traffic conditions in specific zones, gathering data that improves the traffic management and it can be implemented in today's time seeing the number of patients dying only because of the delay of ambulance caused due to traffic. In this system a wireless sensor may communicate to other traffic controllers. This is comparitively more effecient and faster way and may also reduce human labour. More ideas can be used to extend the project, such as different types of emergency vehicles like Ambulance, Police Van, Fire Truck, etc can be distinguished for different priorities and making them reach their destination as soon as possible without any delays.

2.4 FEATURES OF THE PROJECT:

Project features includes all the features that WSN have :
a) The consumption of Power limits for nodes with batteries
b) Capacity to handle node failures
c) Some mobility of nodes and Heterogeneity of nodes
d) Scalability to a large scale of distribution
e) Capability to ensure strict environmental conditions
f) Simple to use
g) Cross-layer design
And the features related to the Project are :
a) Ability to detect the emergency vehicles using RFID sensor.
b) Tracking of emergency vehicles.
c) Sending the real time location of the emergency vehicles using GPS system.
d) Avoid any kind of loss or delay of emergency vehicles to reach their destination.

REVIEW OF LITERATURE

Since inefficiency of effective traffic system results in huge economic loss. It will also return loss of human lives. So, a great work has been done to deal with these problems.

Haneen Hassan Al-Ahmadi, Aalaa Mojahed proposed Real-Time Simulation of Traffic Monitoring System in Smart City. This paper proposed an intelligent traffic monitoring system using a wireless sensor network for the smart city. Our system has various nodes in the whole city, including roads, objects, and traffic signals, and each node is connected, and each node is sharing the data of the vehicle, traffic rate. When a vehicle enters the smart city, initially, vehicles are verified through the registration system, whereas all the vehicles and persons must be registered. Whereas when the new vehicle enters the city, it notifies the sensors and gives information about the new vehicle. They are also providing the facility of the shortest and traffic-free path in terms of distance and time.

Adil Hilmani , Abderrahim Maizate, and Larbi Hassouni proposed Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks. This paper propose an intelligent traffic control system based on the design of a wireless sensor network (WSN) in order to collect data on road traffic in a smart city. In addition, the proposed system has innovative services that allow drivers to view the traffic rate and the number of available parking spaces to their destination remotely using an Android mobile application to avoid traffic jams and to take another alternative route to avoid getting stuck and also to make it easier for drivers when looking for a free parking space to avoid unnecessary trips. Our system integrates three smart subsystems connected to each other (crossroad management, parking space management, and a mobile application) in order to connect citizens to a smart city.

A paper is published by Dr. r. s. Deshpande, J. G. Rana on Traffic Control System Based on Embedded Technology. This paper uses sensor nodes and networks in addition to embedded technology to manage traffic congestion through communication between every junction and controls congestion based on information received from other previous junctions.

According to proposed method of Chakkaphong Suthaputchakun, Zhili Sun and Mehrdad Dianati the system will display status oftraffic lights in advance toall emergency vehicles. So that driver of vehicle can movedown the junction according to the received status. But there may be a chance of occurrence overspeed concept results in accidents.

Rajeshwari Sundar, Santhoshshebbar, and Varaprasad golla proposed Traffic Control System for Emergency Vehicle Clearance and Stolen Vehicle Detection. This system uses RFID readers and RFID tags.

Prajakta Waghere, Priyanka Nalawade, Nisha Vanare, Prajakta Kalbhor, Prof.A.J.Jadhav published IJARSE paper on Dynamic Traffic Control System using RFID technology. It uses IR sensors, microcontroller for controlling traffic flow based on status of IR sensors. This paper deals with decision making algorithm(DMA).So system flow is designed based on this algorithm.Do not use abbreviations in the title or heads unless they are unavoidable.

SYSTEM DESCRIPTION:

4.1 DESIGN:

1. Block Diagram:

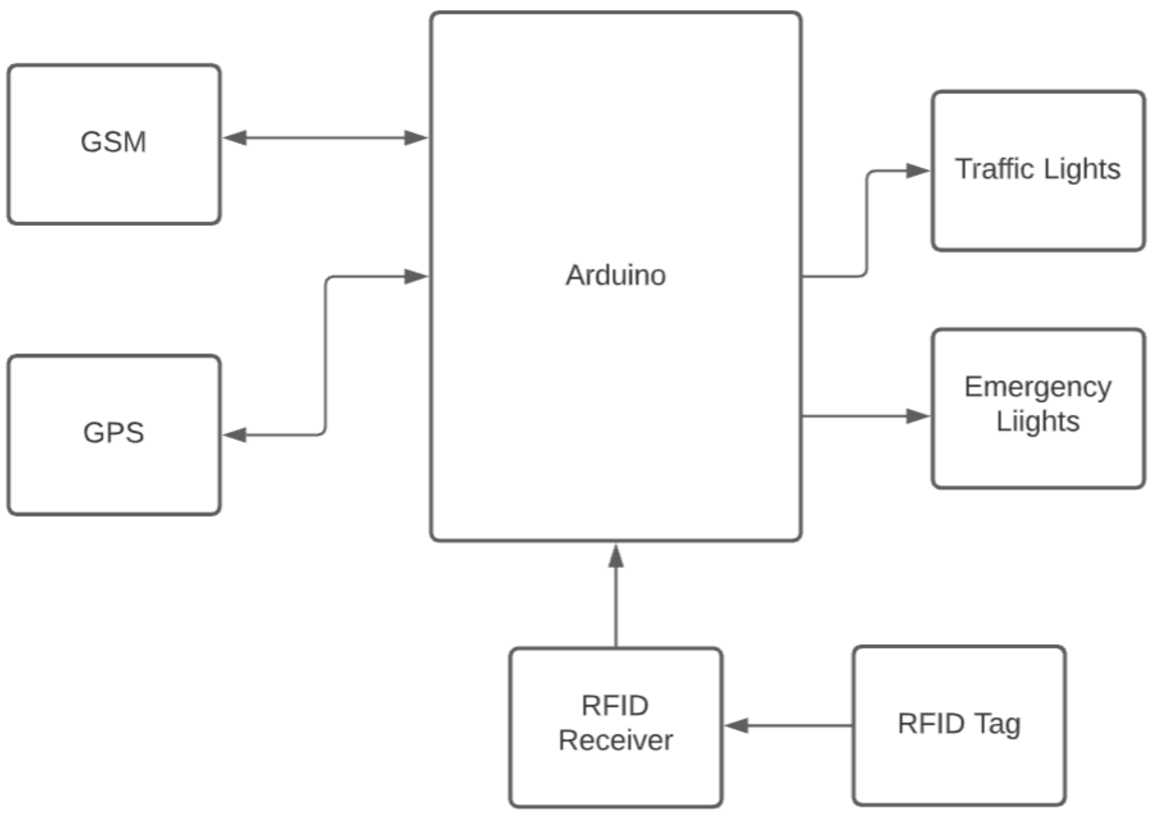

Figure 4.1: Block Diagram

4.2 SOFTWARE USED:

1. PROTEUS SOFTWARE:

   Proteus is a circuit designing software invented by Labcenter Electronics. It used to design different circuits on PCB (printed circuit board) and simulation of different circuits. The use of proteus for any electronic circuit project makes that project cost-effective and less errored due to schematic construction on the proteus. In 1988 the first version of Proteus known as PCB-B was created by John Jameson who was chairman of the company.

   - Introduction to Proteus:

     Proteus is used to simulate, design and drawing of electronic circuits. It was invented by the Labcenter electronic. By using proteus you can make two-dimensional circuits designs as well. With the use of this engineering software, you can construct and simulate different electrical and electronic circuits on your personal computers or laptops. There are numerous benefits to simulate circuits on proteus before make them practically. Designing of circuits on the proteus takes less time than practical construction of the circuit. The possibility of error is less in software simulation such as loose connection that takes a lot of time to find out connections problems in a practical circuit. Circuit simulations provide the main feature that some components of circuits are not practical then you can construct your circuit on proteus. There is zero possibility of burning and damaging of any electronic component in proteus. The electronic tools that are very expensive can easily get in proteus such as an oscilloscope. Using proteus you can find different parents of circuits such as current, a voltage value of any component and resistance at any instant which is very difficult in a practical circuit.

   - Features of Proteus:

     There are 2 main parts of proteus first is used to design and draw different circuits and the second is for designing of PCB layout.
     First is ISIS that used to design and simulate circuits. And second is ARES that used for designing of a printed circuit board.
     It also provides features related to the three-dimensional view of design in PCB.

2. ARDUINO:

   Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. The Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

- Advantages of Arduino :

(a) 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.

(b) Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

(c) Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

(d) Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

(e) Open source and extensible hardware - The plans of the Arduino boards are published under a Creative 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 in order to understand how it works and save money.

4.3 IMPLEMENTATION AND METHODOLOGY:

The project is implemented as follows:

1. Firstly, we displayed a four-way intersection.
2. Then using one Arduino we have implemented 4 signals using the component Traffic Lights in the Proteus Software.
3. We have used a RFID Sensor which is used to detect an Emergency Vehicle.
4. The Signal from the RFID Sensor is sent to the Arduino of the Traffic Lights.
5. This Traffic light arduino checks if the signal is high, which inturn means that the Emergency Vehicle has been detected successfully.
6. Then, the Emergency Lights will be turned on and the traffic light of the road on which the emergency vehicle is present at that moment ,will go green.
7. We have inserted GSM and GPS Modules to send the location of the emergency vehicle to either a hospital or a specified person (who may be the patients relatives).

4.4 CODE:

1. Arduino Code for controlling the Traffic Lights

```
int signal1[] = {A0,2,3};
int signal2[] = {A1,4,5};
int signal3[] = {A2,6,7};
int signal4[] = {A3,8,9};

int greenDelay = 250;
int yellowDelay = 250;

void setup() {
  Serial.begin(9600);
    pinMode(signal1[0], OUTPUT);
    pinMode(signal1[1], OUTPUT);
    pinMode(signal1[2], OUTPUT);

    pinMode(signal2[0], OUTPUT);
    pinMode(signal2[1], OUTPUT);
    pinMode(signal2[2], OUTPUT);


    pinMode(signal3[0], OUTPUT);
    pinMode(signal3[1], OUTPUT);
    pinMode(signal3[2], OUTPUT);


    pinMode(signal4[0], OUTPUT);
    pinMode(signal4[1], OUTPUT);
    pinMode(signal4[2], OUTPUT);


  pinMode(12,INPUT);
  pinMode(11, OUTPUT);

}

void loop() {
  s1();
  s2();
  s3();
  s4();

}


int s1()
```

```
{
  digitalWrite(signal1[1], HIGH);
  digitalWrite(signal1[0], LOW);
  digitalWrite(signal2[0], HIGH);
  digitalWrite(signal3[0], HIGH);
  digitalWrite(signal4[0], HIGH);
  delay(yellowDelay);
  emer();




  digitalWrite(signal1[2], HIGH);
  digitalWrite(signal1[1], LOW);
  delay(greenDelay);
  emer();

  digitalWrite(signal1[2], LOW);

}


int s2()
{

  digitalWrite(signal1[0], HIGH);
  digitalWrite(signal2[1], HIGH);
  digitalWrite(signal2[0], LOW);
  digitalWrite(signal3[0], HIGH);
  digitalWrite(signal4[0], HIGH);
  delay(yellowDelay);
  emer();
  digitalWrite(signal2[0], LOW);



  digitalWrite(signal2[2], HIGH);
  digitalWrite(signal2[1], LOW);
  delay(greenDelay);
  emer();
  digitalWrite(signal2[0], HIGH);
  digitalWrite(signal2[2], LOW);


}

int s3()
{
```

```
  digitalWrite(signal1[0], HIGH);
  digitalWrite(signal2[0], HIGH);
  digitalWrite(signal3[1], HIGH);
  digitalWrite(signal3[0], LOW);
  digitalWrite(signal4[0], HIGH);

  delay(yellowDelay);
emer();


  digitalWrite(signal3[2], HIGH);
  digitalWrite(signal3[1], LOW);
  delay(greenDelay);
emer();
  digitalWrite(signal3[2], LOW);

}

int s4()
{

  digitalWrite(signal1[0], HIGH);
  digitalWrite(signal2[0], HIGH);
  digitalWrite(signal3[0], HIGH);
  digitalWrite(signal4[1], HIGH);
  digitalWrite(signal4[0], LOW);
  delay(yellowDelay);
  emer();



  digitalWrite(signal4[2], HIGH);
  digitalWrite(signal4[1], LOW);
  delay(greenDelay);
emer();
  digitalWrite(signal4[2], LOW);

}



int emer()
{
  int i=digitalRead(12);
Serial.println(i);
if(i==1)
{
```

```
  digitalWrite(11,HIGH);

  digitalWrite(signal1[0], HIGH);
  digitalWrite(signal1[1], LOW);
  digitalWrite(signal1[2], LOW);

  digitalWrite(signal2[0], LOW);
  digitalWrite(signal2[1], LOW);
  digitalWrite(signal2[2], HIGH);

  digitalWrite(signal3[0], HIGH);
  digitalWrite(signal3[1], LOW);
  digitalWrite(signal3[2], LOW);

  digitalWrite(signal4[0], HIGH);
  digitalWrite(signal4[1], LOW);
  digitalWrite(signal4[2], LOW);

 delay(500);
  digitalWrite(signal2[0], HIGH);
  digitalWrite(11,LOW);
   digitalWrite(signal2[2], LOW);

}
else
{
  digitalWrite(11,LOW);
}

}
```

2. Arduino code of RFID Transmitter:

```
#include <VirtualWire.h>
char *controller;
const int buttonPin1 = 2;
int buttonState1 = 0;
void setup()
{
Serial.begin(9600);
pinMode(buttonPin1, INPUT);

pinMode(13,OUTPUT);
vw_set_ptt_inverted(true);
vw_set_tx_pin(12);
vw_setup(4000);
delay(100);}
void loop()
{
   buttonState1 = digitalRead(buttonPin1);
   Serial.println(controller);

  if (buttonState1 == HIGH)
{

controller="1" ;
vw_send((uint8_t *)controller, strlen(controller));
vw_wait_tx();
digitalWrite(13,HIGH);

}
 else
 {
 controller="0" ;
vw_send((uint8_t *)controller, strlen(controller));
vw_wait_tx();
digitalWrite(13, LOW);
 }
}
```

3. Arduino Code of RFID Receiver:

```
#include <VirtualWire.h>
void setup()
{
vw_set_ptt_inverted(true);
vw_set_rx_pin(12);
vw_setup(4000);
Serial.begin(9600);
pinMode(13, OUTPUT);
vw_rx_start();
}

void loop()
{
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;
   Serial.println();


if (vw_get_message(buf, &buflen))

{
if(buf[0]=='1')
{

digitalWrite(13,HIGH);

}
else
{

digitalWrite(13,LOW);

}
}

}
```

4. Arduino code of GPS and GSM Module

```
#include <TinyGPS.h>
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8);

TinyGPS gps;

void setup()
{
  Serial.begin(9600);
  SIM900.begin(9600);

}

void loop()
{
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;


  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (Serial.available())
    {
      char c = Serial.read();
      //Serial.print(c);
      if (gps.encode(c))
        newData = true;
    }
  }

  if (newData)
  {
    float flat, flon;
    unsigned long age;
    gps.f_get_position(&flat, &flon, &age);

    delay(400);
    SIM900.println("Recipient = \"+91xxxxxxxxxx\"");
    delay(300);
    SIM900.print("Latitude = ");
    SIM900.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
    SIM900.print(" Longitude = ");
    SIM900.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
    delay(200);
    SIM900.println((char)26);
```

```
      delay(200);
      SIM900.println();


    }

    Serial.println(failed);
  // if (chars == 0)
    // Serial.println("* No characters received from GPS: check wiring *");
}
```

4.5 FINAL PROTOTYPE:

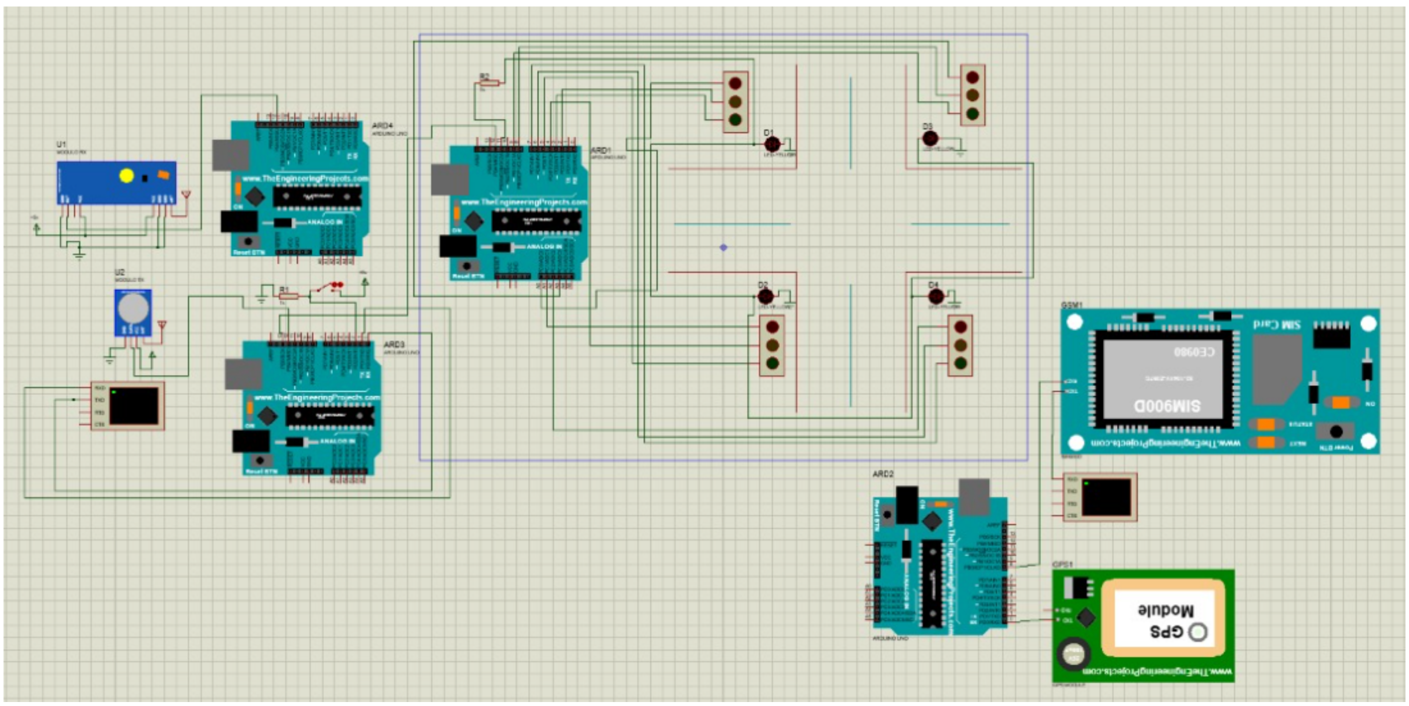

Figure 4.2: Overview of Traffic Management for Emergency Vehicles model

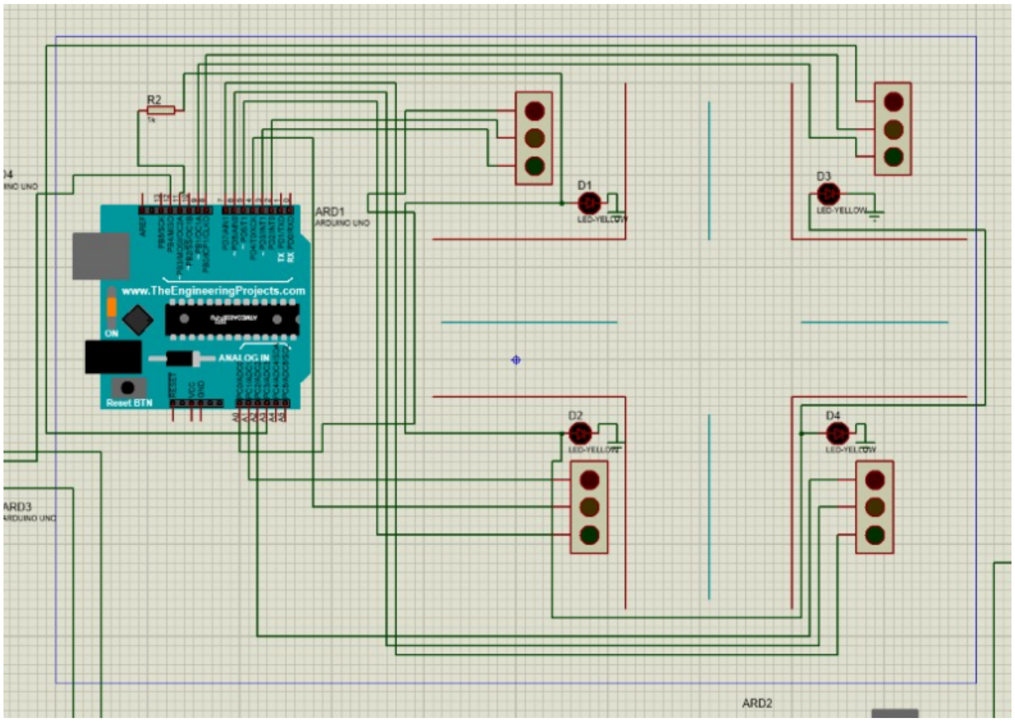

Figure 4.3: Traffic Lights System

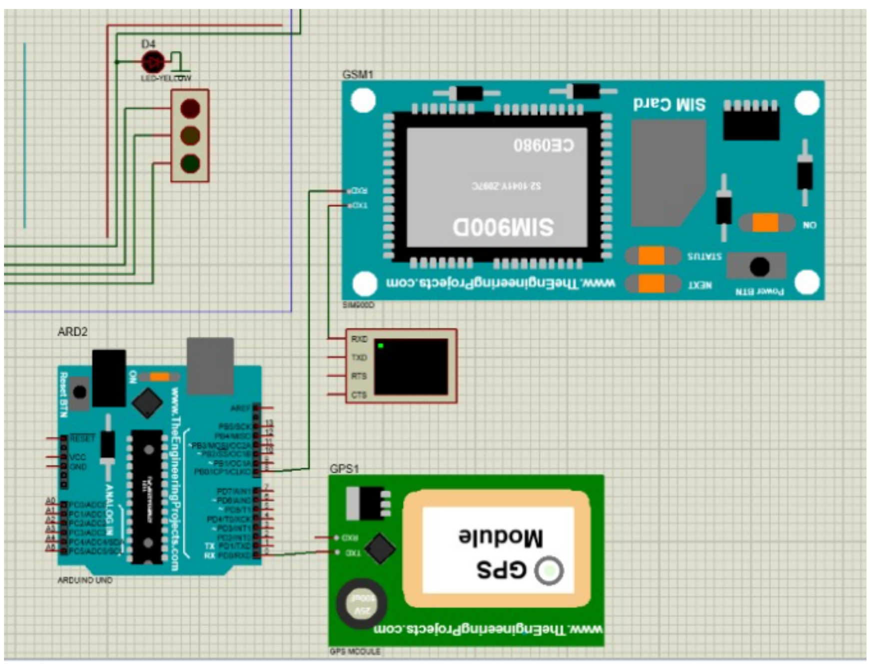

Figure 4.4: GPS and GSM Modules

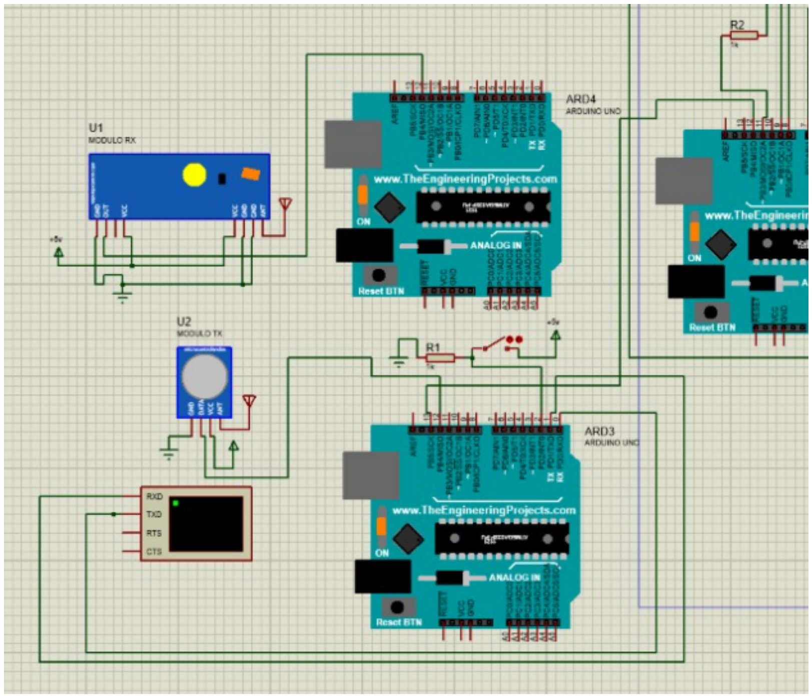

Figure 4.5: RFID Transmitter and receiver

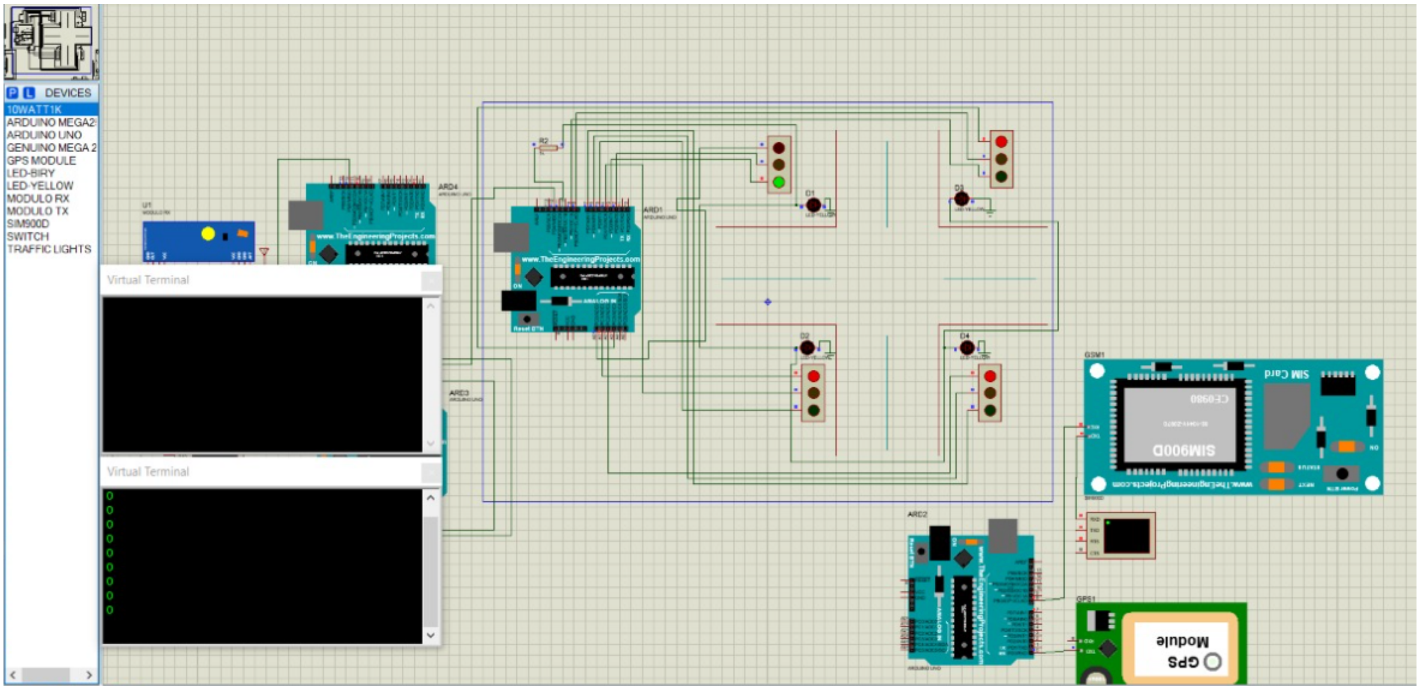

Figure 4.6: In Normal Conditions

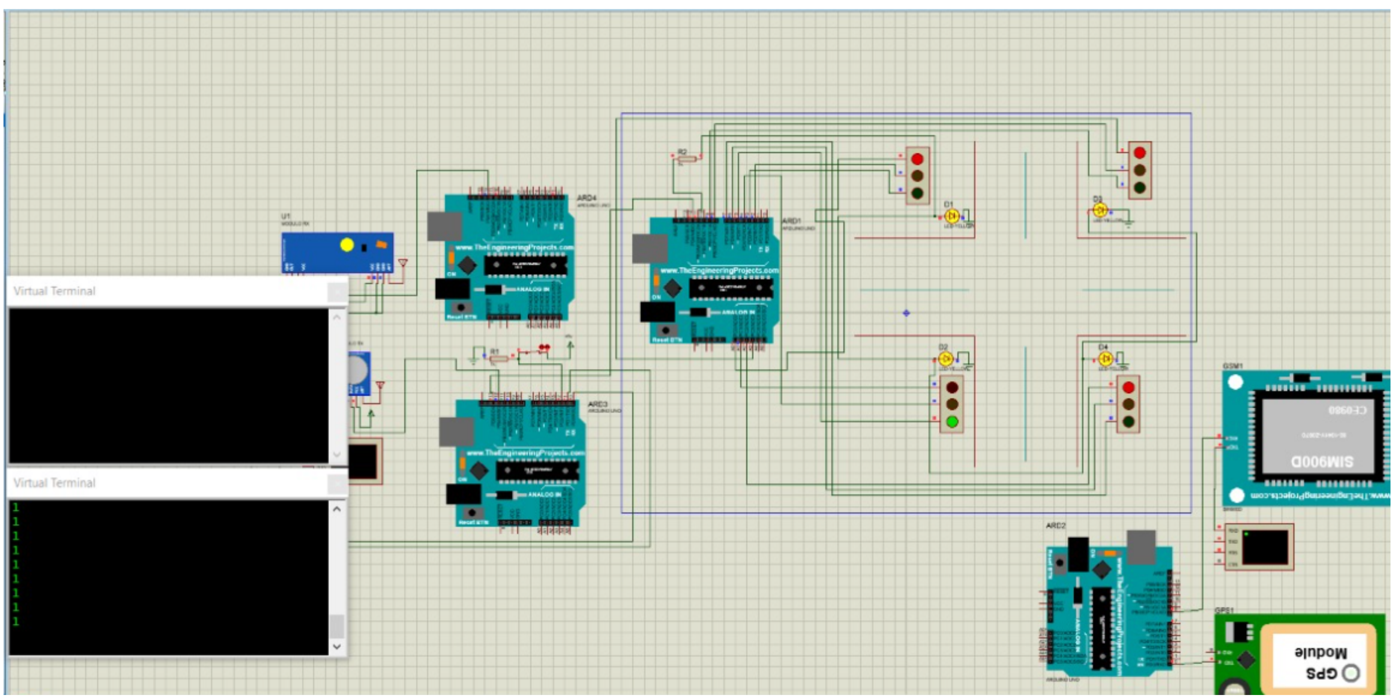

Figure 4.7: When Emergency Vehicle is detected, emergency lights glow

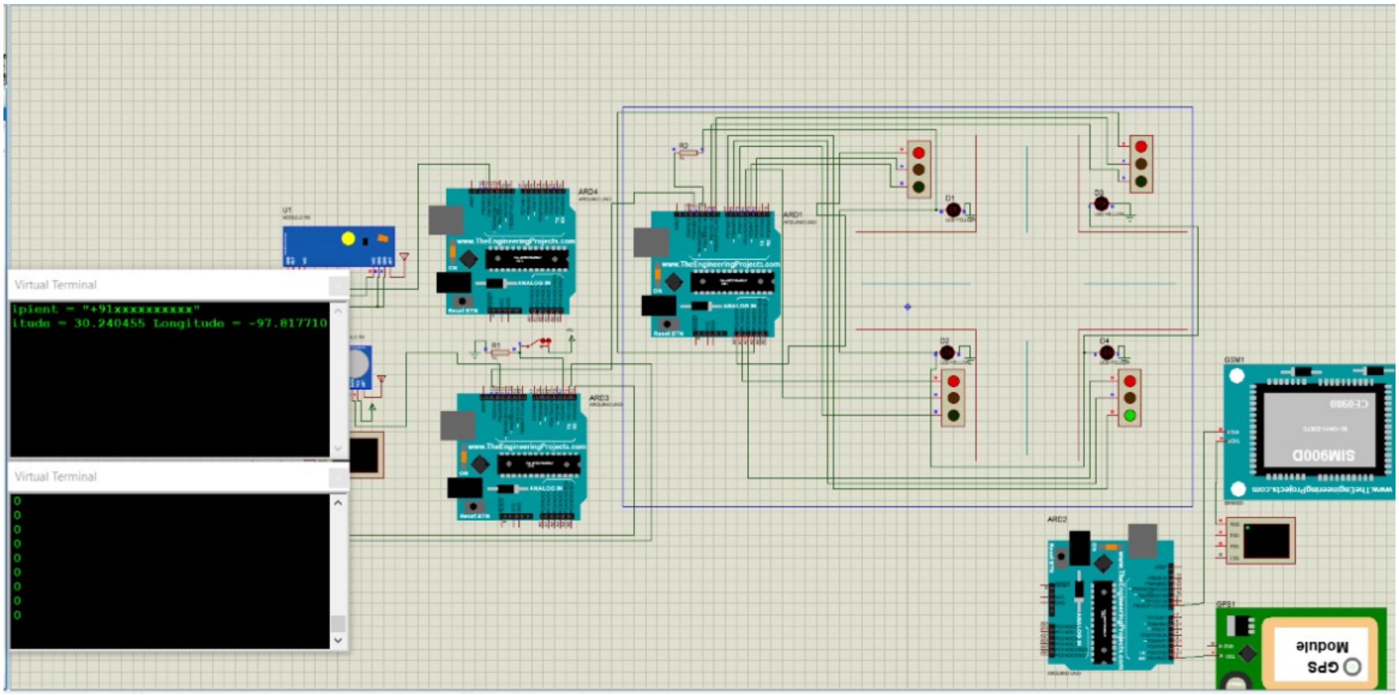

Figure 4.8: Location traced

4.6 FUTURE SCOPE:

Further enhancements can be done to the prototype by testing it with longer range RFID modules. At present, we have implemented system using simulation on Proteus Software, this can be done in real world application and on a large scale. More ideas can be implemented to extend the project, such as different types of emergency vehicles like Ambulance, Police Van, Fire Truck, etc can be distinguished for different priorities. Hence, integrating the entire traffic management system with above ideas opens the door for future scope.

4.7 CONCLUSION:

The project "Traffic Management for Emergency Vehicles in a City" has been successfully designed and tested. In this implementation, we have used Radio Frequency Technology. It is developed with integration of required hardware components in the Proteus Software. This system reduces risk of accidents of emergency vehicles due to unnecessary traffic, and help the emergency vehicles to reach their destination within less time and without facing any obstacles. Hence, a simulation model of "Traffic Management for Emergency Vehicles in a City" is designed and implemented on Proteus Software successfully.

REFERENCES:

1. Proteus User Manual Guide

2. Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks, Published 11, September 2020.

3. Real-Time Simulation of Traffic Monitoring System in Smart City, Published 11, November 2019