GREEN CORRIDOR

A

PROJECT REPORT

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

SUBMITTED BY:

SUKANT DHAMIJA (UE165114)

SUMIT NEGI (UE165117)

SUSHANT SINGH (UE165118)

SEVENTH SEMESTER

UNDER THE SUPERVISION OF:

SUMIT BUDHIRAJA

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY,

PANJAB UNIVERSITY, CHANDIGARH, INDIA

NOVEMBER 2019

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY, PANJAB UNIVERSITY, CHANDIGARH

CANDIDATE’S DECLARATION

We hereby certify that the work presented in the Project entitled “**GREEN CORRIDOR**” in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in Electronics and Communications Engineering from University Institute of Engineering and Technology, Panjab University, Chandigarh, is an authentic record of my own work carried out under the supervision and guidance of Sumit Budhiraja (Assistant Professor).

Date:

Sukant Dhamija (UE165114)

Sign:

Sumit Negi (UE165117)

Sign:

Sushant Singh (UE165118)

Sign:

Place: Chandigarh

Semester 6

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date:
Place: Chandigarh

Sumit Budhiraja
Assistant Professor
ECE, UIET, Panjab University,
Chandigarh

ABSTRACT

The growth of industrialization and urbanization has led to an immense increase in the population invariably leading to rise in the number of vehicles on road. The resulting traffic congestion and traffic jams are the major hurdles for emergency vehicles such as ambulance carrying critical patients as these emergency vehicles are not able to reach their destination in time, resulting into a loss of human life. To tackle this problem to some extent we have come up with “GREEN CORRIDOR”. The proposed system clears the traffic congestion by turning all the red lights to green on the path of the ambulance, hence helping in clearing the traffic and providing way towards its destination. The system consists of a web page which registers the ambulance on the network. In case of emergency situation, if the ambulance halts on its way, the it sends an emergency command to the traffic signal server and also the direction where it wants to travel along with this the current position with the help of Global Positioning System (GPS). The nearest signal is identified based upon the current position of the ambulance. And that particular signal is made green till the ambulance passes by and later it regains its original flow of control. In this way it acts like a lifesaver project as it saves time during emergency by controlling the traffic lights.

CONTENTS :-

CHAPTER 1
INTRODUCTION

1.1 Green Corridor Introduction

The pace at which the world is developing is very high today. Reformations in technology every day is evolving and improving efficiency in healthcare sector is one of the most difficult and challenging jobs also with the advent of Industrialization and Urbanization, as the population increases day by day the number of vehicles also increases on the roads. This leads to high traffic jams in big cities. Traffic congestion causes many adversary effects on countries transportation. One of the widely affected service due to traffic jams is that of an ambulance. Many a times, ambulance consist of emergency or critical patients which needs to be taken to the hospital in minimum amount of time providing proper treatment to the patient so that chances of surviving increases in critical condition. A Patient may lose his life if there is delay in reaching of ambulance to the hospital. According to the surveys 95% of the heart attacks cases can be treated, if the ambulance can reach the hospital at current time without stucking into the traffic. For this, it is needed that the vehicles on the road to make way for the ambulance. But sometimes, the ambulance gets stuck in the traffic which in turn wastes a lot of time waiting for the traffic to get clear. We can overcome these limitations by the emerging technology such as IoT i.e. Internet of Things. Various software implementations and hardware devices can be connected with the help of wireless networking tools or wired tools. In IoT the components are connected and controlled by the internet. Thus, the impact of IoT in today's era is significant as it helps to represent the object digitally and makes itself something greater than the object by itself. In this paper, we have come up with the '**GREEN CORRIDOR**'. The main objective of this system is to make it possible for the ambulance to reach a particular location without having it to stop anywhere until the destination is reached. This project proposes monitoring of traffic lights and its controlling by the driver of the ambulance. Depending upon the emergency, the driver sends the direction towards which it wants to travel. Depending upon the command, that particular signal is made green to provide way to the ambulance and simultaneously the others are changed to red. Using this method, way is provided to the ambulance resulting it to reach the destination in minimum time.

1.2 THEORY RELATED TO PROJECT

As the country is developing, so is the growth in terms of population which eventually increases the number of vehicles running on the road which leads to high traffic jams in big cities. Traffic congestion causes many adversary effects on countries transportation and one of the widely affected service due to traffic jams is that of an ambulance.

When a patient is in a serious condition, ambulance has to be as fast as it can be to save the life of a patient, and we might have even come across the saying often that if the patient had reached the hospital in time, then his or her life could have been saved. Such a problem is what we are confronting these days, where no one is accountable for this cause but the motor machines driven by humans and the inefficient technology due to which the patient is unable to reach the hospital on time and he/she had to pay the price of his/her life. Cardiac attack patients need this service most as a little delay in the service could be fatal.

Concerning to this problem we have come up with a solution which could solve it to a large extent, thus minimizing the number of deaths due to delay in ambulance service. GREEN CORRIDOR is one such smart solution in which whenever an ambulance with an emergency case will pass through a traffic light, it will turn ON the red signal for all the lanes expect for the one, which leads to the destination of an ambulance by keeping the green signal ON and this strategy will allow ambulance to surpass the heavy traffic and saving the crucial minutes in an hour of need to carry patient to hospital without any delay.

For the above stated problem, we have a model of our solution shown in fig.2.1, which consists of arduino NANO, NRF transceiver module, web page to register for an emergency case or non-emergency case, NodeMCU, RFID reader and RFID tag.

We have embedded the RFID Tag in the ambulance having a unique identification code which will be detected by the RFID module, located between two roundabouts, during the case of an emergency situation and will remain off during the non-emergency situation transmitting necessary signal through NRF module to the other NRF module located at the roundabout to cause the necessary changes in the signal flow of traffic control system. While GPS module will keep on monitoring the coordinates of the ambulance to locate the nearest traffic control system. Arduino NANO and NodeMCU both act as the brain of the model giving necessary commands to the sensors, where NodeMCU has an additional feature of internet connectivity for registration of an ambulance under the emergency or non-emergency situation.

The ambulance gets a facility to put it on emergency or non-emergency condition through web page where NodeMCU will provide internet connectivity. For an emergency case, as soon as RFID reader detects RFID tagged ambulance it will transmit signal to traffic lights to turn the green signal ON for the desired lane. For a non-emergency case, RFID reader will not operate and hence will not detect the RFID tag and everything will work normally. Therefore, with the aid of such a project a good solution is provided to the above stated problem.
We can conclude that it will definitely be a life saviour for the needy as it will give full assistance in turning the lights green on for the desired lane, thus helping an ambulance to reach the hospital on time.

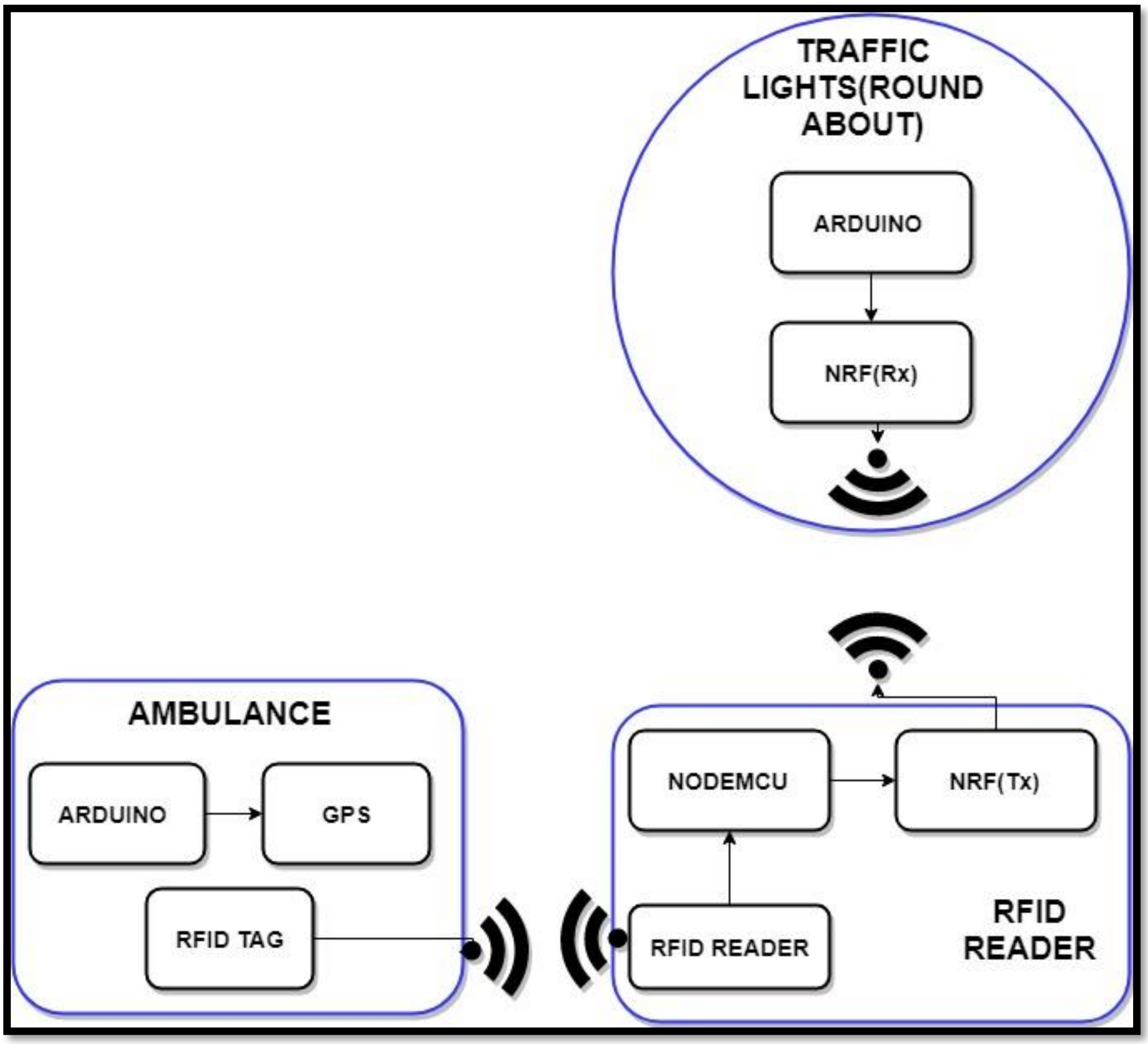


Fig. 1.1 Block Diagram of GREEN CORRIDOR

CHAPTER 2

IMPLEMENTATION

2.1 MATERIAL REQUIRED

This project cumulates both hardware and software environment. The information regarding the materials has been provided in the section 2.1.1 and section 2.1.2.

2.1.1 HARDWARE REQUIREMENTS

We have six hardware components in this project which are as follows:

1. Arduino NANO
2. Node MCU
3. NRF24L01
4. RFID reader and RFID tag
5. GPS (GLOBAL POSITIONING SYSTEM)
6. LED (LIGHT EMITTING DIODE)

1. ARDUINO NANO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards as shown in fig 3.1, are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. Arduino NANO in contrast to Arduino UNO board is compact in size and has two additional input output pins.

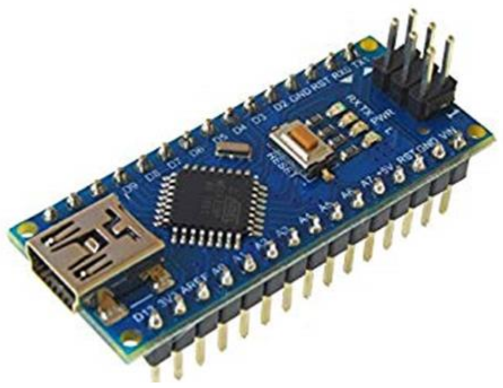


Fig. 3.1 Arduino NANO Board

Advantages of Arduino board:

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 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.

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.

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.

2. NODE MCU

NodeMCU as shown in fig.3.2, is a wi-fi SOC (system on a chip) produced by Espressif Systems. It is based ESP8266 -12E Wi-Fi module. It is a highly integrated chip designed to provide full internet connectivity in a small package.

It can be programmed directly through USB port using LUA programming or Arduino IDE. By simple programming we can establish a Wi-Fi connection and define input/output pins according to your needs exactly like arduino, turning into a web server and a lot more.

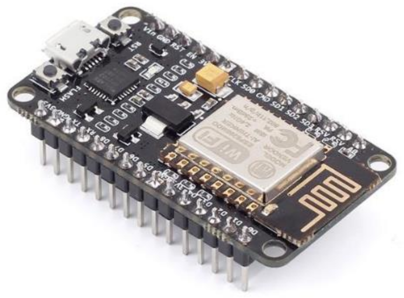


Fig. 3.2 Node MCU

NodeMCU is the Wi-Fi equivalent of ethernet module. It combines the features of Wi-Fi access point and microcontroller. These features make the NodeMCU extremely powerful tool for Wi-Fi networking. It can be used as access point and/or station, host a web server or connect to internet to fetch or upload data.

Features

1. Finally, programable Wi-Fi module.
2. Arduino-like (software defined) hardware IO.
3. Can be programmed with the simple and powerful Lua programming language or Arduino IDE.
4. USB-TTL included, plug & play.
5. 10 GPIOs D0-D10, PWM functionality, IIC and SPI communication, 1-Wire and ADC A0 etc. all in one board.
6. Wi-fi networking (can be used as access point and/or station, host a web server), connect to internet to fetch or upload data.
7. Event-driven API for network applications.
8. PCB antenna.

3. NRF24L01

The **NRF24L01** as shown in fig 3.3, is a wireless **transceiver module**, meaning each module can both send as well as receive data. They operate in the frequency of 2.4GHz, which falls under the ISM band and hence it is legal to use in almost all countries for engineering applications. The modules when operated efficiently can cover a distance of 100 meters (200 feet) which makes it a great choice for all wireless remote-controlled projects.

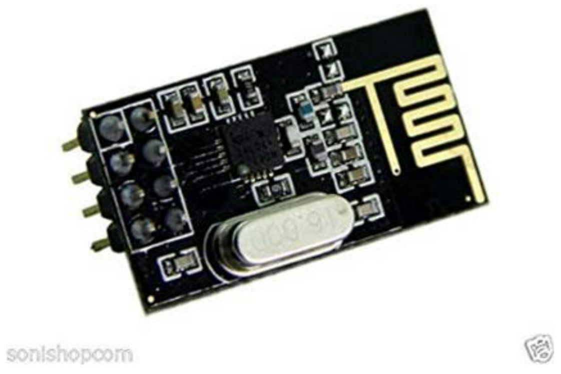


Fig. 3.3 NRF24L01

The module operates at 3.3V hence can be easily used with 3.2V systems or 5V systems. Each module has an address range of 125 and each module can communicate with 6 other modules hence it is possible to have multiple wireless units communicating with each other in a particular area. Hence mesh networks or other types of networks are possible using this module.

4. GPS MODULE

GPS receiver module gives output in standard (National Marine Electronics Association) NMEA string format. It provides output serially on Tx pin with default 9600 Baud rate.

This NMEA string output from GPS receiver contains different parameters separated by commas like longitude, latitude, altitude, time etc. Each string starts with '$' and ends with carriage return/line feed sequence.

PIN DESCRIPTION:

VCC: Power Supply 3.3 – 6 V

GND: Ground

TX: Transmit data serially which gives information about location, time etc.

RX: Receive Data serially. It is required when we want to configure GPS module

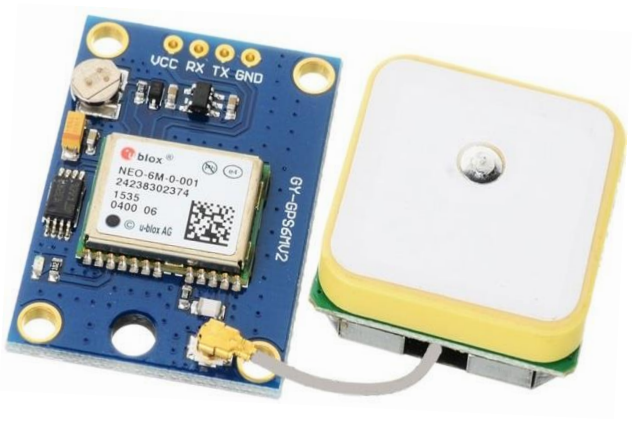


Fig. 3.4 GPS Module

5. RFID READER AND RFID TAG

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically stored information. Passive tags collect energy from a nearby interrogating radio waves of the RFID reader, as in fig.3.5a. Active tags have a local power source (such as a battery) and may operate hundreds of meters from the RFID reader. Unlike a barcode, the tag need not be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method of automatic identification and data capture (AIDC).

RFID tags (fig 3.5b) are used in many industries. For example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line

The RFID reader (fig 3.5a) has a radio transmitter and receiver inside. It is also called as an interrogator. The reader transmits radio frequency signals continuously upon powering. When an RFID tag (fig. 3.5b) is placed inside the range area of a reader, it energizes the tag through electromagnetic induction and collects the information from it.

The image given above is that of an RFID tag (smart card shaped tag). RFID tags are available in different types of size and shapes. The Tag contains an IC for storing the data, an antenna for transmitting and receiving, and also a modulator. Tags are very small in size and they can hold only few bits of data.

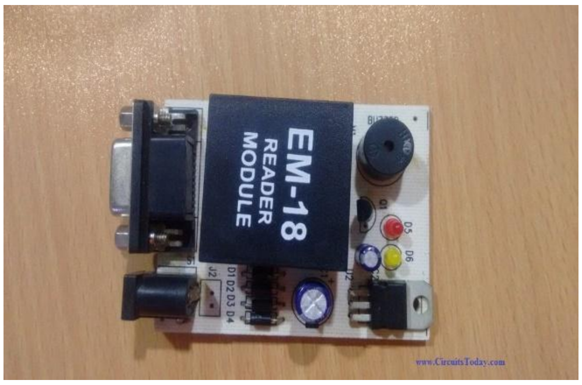


Fig. 3.5a RFID Module

Fig. 3.5b RFID Tag

6. LED (LIGHT EMITTING DIODE)

Fig. 3.6 Light Emitting Diode

A **light-emitting diode** (**LED**) as shown in fig.3.6, is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. This effect is called electroluminescence. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device. LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching.

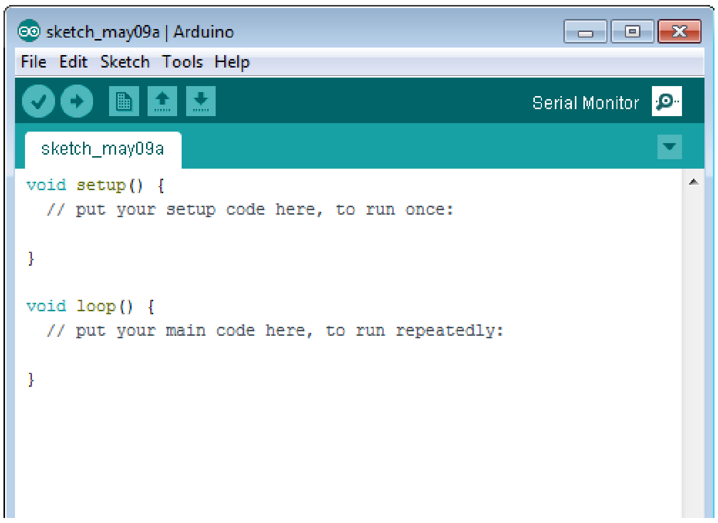
2.1.2 SOFTWARE REQUIREMENTS

The platform that we use in this project for the software part is as follows:

1. Arduino IDE

1. Arduino IDE

The Arduino integrated development environment (IDE) as in fig.3.7 is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig. 3.7 Arduino IDE Interface

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

The hardware components in Table 1 and the software platforms in Table 2 are shown along with the quantities required of each and the respective cost of each. The total cost incurred in this project is approximately Rs. 1830.

HARDWARE

S. No	COMPONENT	DESCRIPTION	QUANTITY	COST PER ITEM (Rs)
1	Arduino Nano	Microcontroller unit which acts as brain of each module	2	250
2	Node MCU	To provide internet connectivity	1	300
3	NRF24L01	To establish wireless communication	2	100
4	RFID reader and RFID tag	To detect the presence of ambulance in emergency conditions	1	200
5	GPS	To locate the coordinates of the ambulance	1	800
6	LED	For traffic lights	3	10

Table 1

SOFTWARE

S. No	PLATFORM	DESCRIPTION
1	ARDUINO IDE	Platform to write down the codes for hardware

Table 2

2.2 METHODOLOGY

We have three modules in the **Intelligent Traffic System for Ambulance** as shown in block diagram in fig.3.8.

1) **MODULE 1**- RFID TAGGED AMBUALNCE
2) **MODULE 2**- RFID READER
3) **MODULE 3**- TRAFFIC LIGHTS

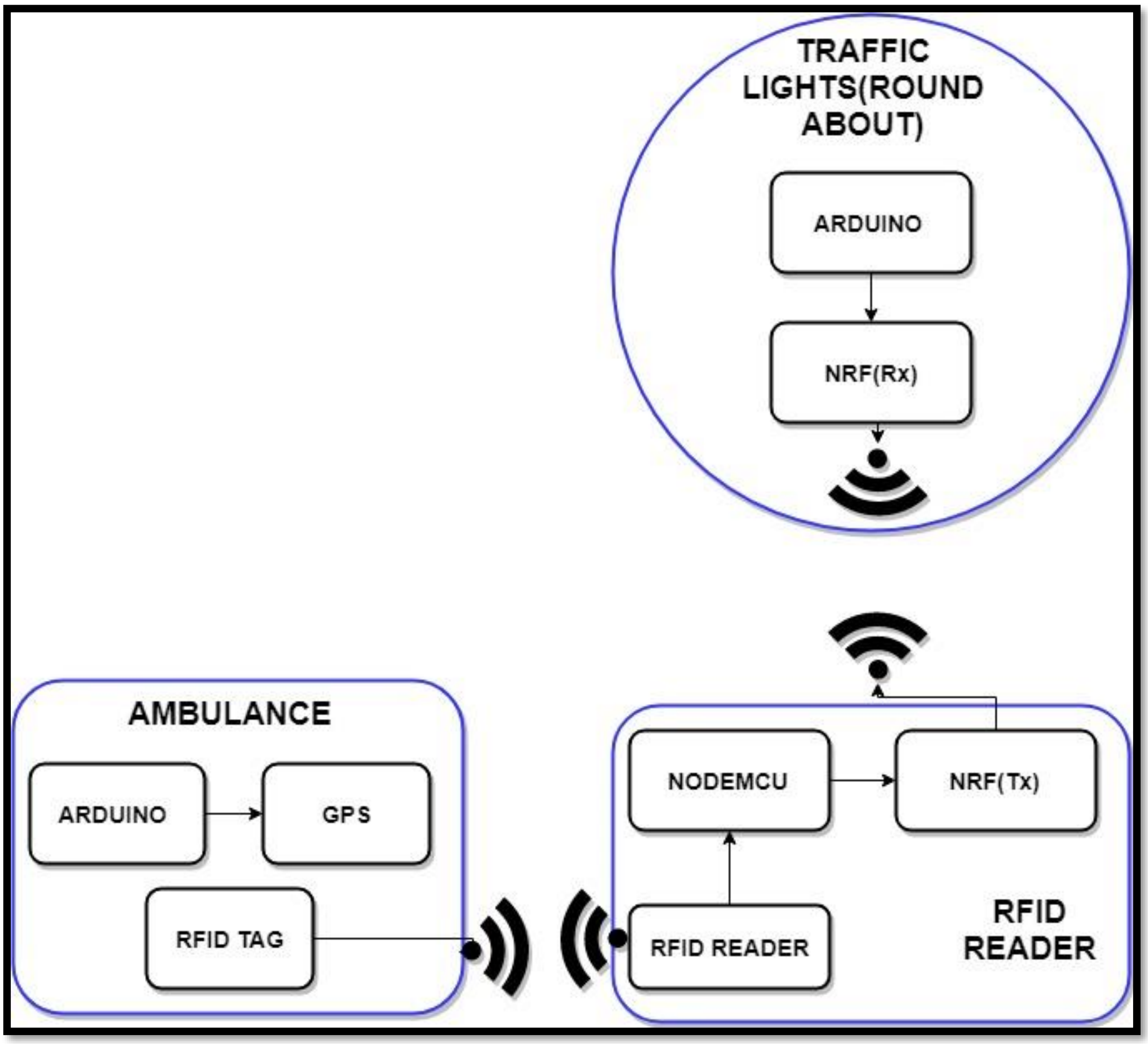


Fig. 3.8 Block Diagram of GREEN CORRIDOR

1. MODULE 1

Ambulance will consist:

a) **Arduino NANO**- Microcontroller unit controlling all the sensors i.e. GPS module and RFID tag according to the code embedded in it.
b) **GPS module-** It will locate the coordinates of the ambulance.
c) **RFID tag-** A unique identification given to the ambulance which will be authenticated by the RFID reader to change the flow control of lights.

2. MODULE 2

RFID reader module will be placed between two roundabouts. It consists:

a) **RFID reader-** It will authenticate the RFID tagged ambulance in case of emergency situation and will send the signal to change the flow of control of traffic lights.
b) **NRF-** It is a transceiver to establish a wireless communication between the RFID reader module and the traffic lights module at the roundabout.
c) **NODEMCU**- It acts as a microcontroller but with an additional feature of internet connectivity.

3. MODULE 3

Traffic lights module will be placed at roundabout. It consists:

a) **Arduino NANO-** Microcontroller unit controlling all the sensors i.e. NRF module according to the code embedded in it.
b) **NRF-** It is a transceiver to establish a wireless communication between the RFID reader module placed between the roundabouts and the traffic lights module at the roundabout.

2.3 WORKING

Ambulance will be given a uniquely identified RFID tag. The ambulance driver will make the use of web page using local IP address as shown in fig.3.9, to register the ambulance for an emergency or non-emergency condition. The GPS will keep on monitoring the coordinates of the ambulance which will help in turning on the nearest possible RFID reader module. The webpage will consist two options:

a) **EMERGENCY**
b) **NON-EMERGENCY**

In case of **emergency situation**, the ambulance driver will put the ambulation in emergency situation and this will turn on the RFID reader placed between the two roundabouts. So, whenever the ambulance will pass that lane, the RFID reader will read the unique identification tag of the ambulance and will transmit the signal through NRF to another NRF placed at roundabout and thus changing the flow of control of traffic light system in favour of ambulance by changing the red light to the green light for the desired lane.

In case of **non-emergency situation**, the RFID reader will not be turned on and thus even if the ambulance passes by the RFID reader it will not cause any change in the signal. The prepared model for the Intelligent Traffic System for Ambulance is shown in fig.3.8.

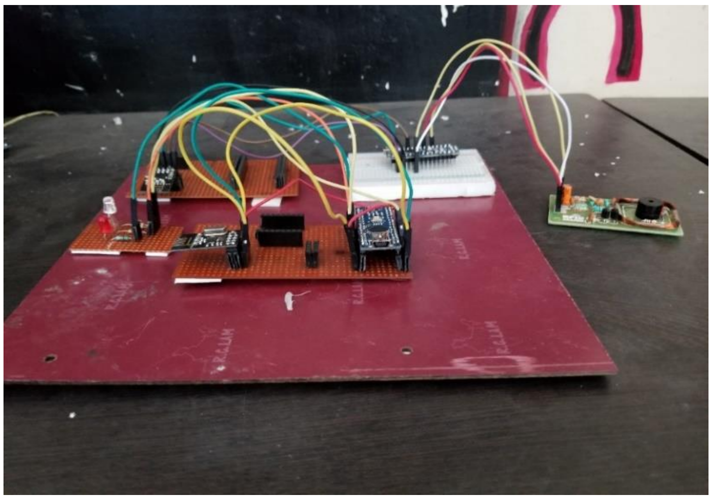


Fig. 2.8 Project model

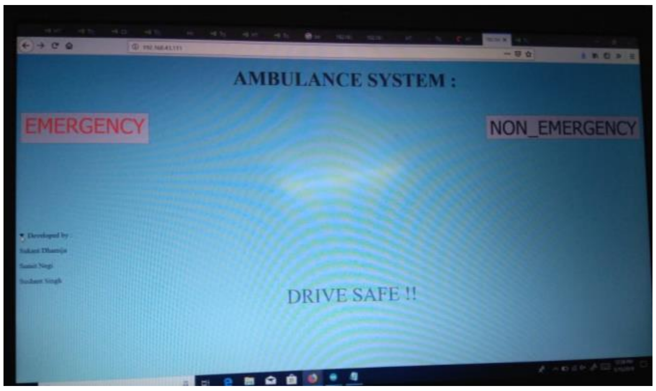


Fig. 2.9 Web Page

CONCLUSION AND FUTURE SCOPE

Human life is precious and we must follow safety measures in all aspects of life which includes ambulances services too. In this, by using intelligent ambulance system we can achieve the uninterrupted service of the traffic control system by implementing the alternate methods for signal change to allow flow control. The accuracy of the RFID is more than that of Camera's, so our project also improves the performance of the traffic light in changing the flow control during the emergency situation. This system uses the techniques of the world changing technology, IOT (Internet of Things), efficient and reliable. This system will reduce accidents which often happen at the traffic signal intersections as other vehicles have to huddle to give way to the ambulance services. And the patient in critical condition in the ambulance will have a higher probability to survive as ambulance will not be halted in its route and will reach on time in the hospital. This life saver project must be implemented in the traffic forums to aid the public in good manner.

Our idea in future can be extended by implementing the shortest way to hospital, to treat patients and also alert the recommended Doctor in that hospital by providing initial medical details of patient like blood pressure, blood group, heart rate, medical history etc. which will be merit in the treatment of critical patient and reduce the time delay. Facility to store details of several patients over long periods of time can be provided in cloud environment.

REFERENCES

1) A. Balamuruganl, G. Navin, Siva Kumar, S. Raj Thilak, P. Selva kumar, "Automated Emergency System in Ambulance to Control Traffic Signals using IoT", *ProfessorSri Krishna College of Technology Coimbatore - 42 Tamilnadu India*.
2) Sensor node and RFID information via Traffic_light_control_and_coordination, [online] Available: www.aten.wikipedia.org/wiki/
3) "Smart Traffic System for Ambulance Using RFID and Internet of Things".....M. Agila, P. Karthikeyan, M.Arun Kumar
4) Arduino Wireless Communication – NRF24L01 Tutorial…….https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/
5) "EM-18 RFID Reader Module Interfaced With Arduino Uno" …..https://www.instructables.com/id/EM-18-RFID-Reader-Module-Interfaced-With-Arduino-U/

APPENDIX

1. RFID READER MODULE CODE

```
#include <ESP8266WiFi.h>
#include<SPI.h>
#include<nRF24L01.h>
#include<RF24.h>
#include<SoftwareSerial.h>
#include<String.h>
const char* ssid = "sukant";
const char* password = "Sukant22";
WiFiServer server(80);
int ambulance1 = 2;
int val = 0;
int i;
int indication = D3;
RF24 radio(D4,D8);
String amb = {"$0007016533"};
unsigned char ADDRESS0[5] = {0xb1, 0x43, 0x88, 0x99, 0x45};
SoftwareSerial ss(D1,D2);
void setup()
{
  ESP.wdtDisable();
  Serial.begin(115200);   //initialise Serial
  delay(10);
  ss.begin(9600);         //initialise UART
  pinMode(indication,OUTPUT);
  /*_________Wifi Network__________*/

  Serial.println();
  Serial.println();
  Serial.print("Connecting to : \t");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
   delay(500);
   Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  server.begin();
  Serial.println("Server started");
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
  digitalWrite(indication,LOW);
  for(i=0;i<=2;i++)
  {
  digitalWrite(indication,HIGH);
  delay(200);
  digitalWrite(indication,LOW);
  delay(200);
  }
  /*________________________________________*/

/*__________NRF Interfacing______________*/
  radio.begin();
  radio.setRetries(15,15);
  radio.setDataRate(RF24_250KBPS);
  radio.setPALevel(RF24_PA_MAX);
  radio.openWritingPipe(ADDRESS0);
  radio.openReadingPipe(0,ADDRESS0);
  radio.stopListening();

```

```
/*______________________________________________*/
}
void loop()
{
	WiFiClient client = server.available();

	if (!client)
	{
	  ESP.wdtFeed();
	  return;
	}
	Serial.println("Waiting for new data");

	while(!client.available())
	{
	  ESP.wdtFeed();
	  delay(1);
	}
	String request = client.readStringUntil('\r');
	Serial.println(request);
	client.flush();
	int value = LOW;
	if(request.indexOf("/EMERGENCY")!= -1)
	{  value = HIGH;
	  Serial.println("Waiting for ambulance\n");
	  while(!ambulance())
	   {
	      ESP.wdtFeed();
	   };
	  radio.write(&ambulance1,sizeof(ambulance1));
	  Serial.println("Ambulance is coming\n");
	  Serial.println("Sending data to traffic light system");
	  Serial.println("------------------------------------");
	 }
```

```
    if (request.indexOf("/NON_EMERGENCY") != -1)
      {
        value = LOW;
      }

/*_________________________WEBPAGE________________________*/

      client.println("HTTP/1.1 200 OK");
      client.println("Content-Type: text/html");
      client.println("");
      client.println("<!DOCTYPE HTML>");
      client.println("<html>");
      client.println("<body style='background-color:powderblue;'>");
      client.print("<h1 style=text-align:center;><font size = 12 color = black>AMBULANCE SYSTEM</style></font></h1>");
      client.println("<br><br>");
      client.println("<a href=\"/EMERGENCY\"\"><button style='float:left'><font size = 8 color = red>EMERGENCY</style></font></p></button></a>");
      client.println("<a href=\"/NON_EMERGENCY\"\"><button style='float:right'><font size = 9 color = green>NON_EMERGENCY</style></font></button></a>");
      client.println("<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>");
      client.println("<details>");
      client.println("<summary>Developed by :</summary>");
      client.println("<p> Sukant Dhamija</p>");
      client.println("<p> Sumit Negi</p>");
      client.println("<p> Sushant Singh</p>");
      client.println("</details>");
<p>I have a date on <time datetime="2008-02-14 20:00">Valentines day</time>.</p>
      client.println("<p align = center><font size = 12 color =  #707b7c > DRIVE SAFE !!</font></p>");
      client.println("</html>");
```

```
/*__________________________OVER____________________________*/

 delay(1);
 Serial.println("Client disonnected");
 Serial.println("");
}
int ambulance()
 {
  String msg;
  char c;
  int val=0;
 if(ss.available()>0)
 {
   while(ss.available())
 {
   c=ss.read();
   msg=msg+c;
   delay(10);
 }
 msg=msg.substring(0,11);
 if(amb.indexOf(msg)<0)
 {
   Serial.print("Unauthorized ambulance");
   val=0;
 }
 else val=1;
 }
 return val;
 }
```

2. ROUNDABOUT MODULE CODE

```
#include<SPI.h>
#include<nRF24L01.h>
```

```
#include<RF24.h>
#include<SoftwareSerial.h>
#include<String.h>
RF24 radio(9,10);
byte pip;
byte newdata;
int rx;
int amb;
unsigned char ADDRESS2[1]= {0xb2};
unsigned char ADDRESS3[1]= {0xb3};
unsigned char ADDRESS4[1]= {0xb4};
unsigned char ADDRESS5[1]= {0xb5};
unsigned char ADDRESS1[5]  =
{
   0xb1,0x43,0x88,0x99,0x45
};
unsigned char ADDRESS0[5]  =
{
  0xb0,0x43,0x88,0x99,0x45
};
void setup() {
  Serial.begin(9600);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  radio.begin();
  radio.setPALevel(RF24_PA_MAX);
  radio.setDataRate(RF24_250KBPS);
  radio.openWritingPipe(ADDRESS0);
  radio.openReadingPipe(0,ADDRESS0);
  radio.openReadingPipe(1,ADDRESS1);
  radio.openReadingPipe(2,ADDRESS2);
  radio.openReadingPipe(3,ADDRESS3);
  radio.openReadingPipe(4,ADDRESS4);
  radio.openReadingPipe(5,ADDRESS5);
  radio.startListening();
  delay(1000);
}

void loop()
{
    if ( radio.available(&pip) )
    {
      radio.read(&rx,sizeof(rx));
      newdata = 1;
      Serial.println(pip);
    }
    else
    {
      Serial.println("Red\tOrange\tGreen");
      digitalWrite(4,HIGH);
      digitalWrite(5,LOW);
      digitalWrite(6,LOW);
      Serial.println("ON \t OFF \t OFF");
      Serial.println("Waiting for ambulance");
    }
  if(newdata == 1)
  {
   newdata = 0;
   if(pip==1)
   {
     amb = rx;
     Serial.print("Amb = ");
     Serial.println(amb);
   }
   if(amb==2)
   {
       Serial.println("-----------------------------------");
```

```
        Serial.println("Ambulance is coming\n\nSwitching Red Light to Green");
        Serial.println("Red\tOrange\tGreen");
        digitalWrite(4,LOW);
        digitalWrite(5,LOW);
        digitalWrite(6,HIGH);
        Serial.println("OFF \t OFF \t ON");
        Serial.println("--------------------");
        delay(10000);
    }
  }
}
```