Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Information Technology

(Department of Electronics & Telecommunication)

Hardware implementation entitled

"INTELLIGENT AMBULANCE WITH AUTOMATIC CONTROL SYSTEM"

Submitted By:				
Roll No.	**Division**	**Name of Student**	**E-mail**	**Contact Number**
311021	A	Devesh Aasheesh Khanzode	Devesh.21820079@viit.ac.in	8788003997
211065	A	Pranali pravin sonawane	Pranali.21820006@viit.ac.in	9156982822

For the course

ROBOTICS & MECHATRONICS

T.Y. B. Tech

Under supervision of

Prof. S. T. Kulkarni

Year 2019 – 2020

INDEX

ABSTRACT

The use of Embedded technology has proved to be very beneficial in present Traffic Light controller (TLC) and that will minimize waiting time of vehicle and also manage traffic load. In this project we exploit the emergence of new technology called as intelligent traffic light controller; this makes the use of RFID, ESP82, and GPS along with embedded technology. Where traffic light will be intelligently decided based on detection of ambulance by RFID & traffic flow rate on the region of ambulance's current location. Thus optimization of traffic light switching increases road Capacity, traffic flow and can prevent traffic congestions.

Traffic lights play an important role in traffic management. Traffic lights are the signaling devices that are placed on the intersection points and used to control the flow of traffic on the road. In 1868, the traffic lights only installed in London and today these have installed in most cities around the world. Most of the traffic lights around the world follow a predetermined timing circuit. Sometime the vehicles on the red light side have to wait for green signal even though there 5 little or no traffic. It results in the loss of valuable time.

Traffic control at intersections is a matter of concern in large cities. Several attempts have been made to make traffic light's sequence dynamic so that these traffic lights operate according the current volume of the traffic. Most of them use the sensor to calculate current volume of traffic but this approach has the limitation that these techniques based on counting of the vehicles and treats emergency vehicles as the ordinary vehicles means no priority to ambulance, fire brigade or V.I.P vehicles. As a result, emergency vehicles stuck in traffic signal and waste their valuable time.

So in this project, to avoid such valuable time of emergency vehicles RFID tags are used. An RFID reader detects them and clears traffic for easy access to emergency vehicle. There is also one better thing, when ambulance is called up, driver can select appropriate hospital location and sends it current location.

1. INTRODUCTION

This particular project is designed for the cities with heavy traffic. E.g. in Bangalore the roads are full jammed every time. Most of the time the traffic jam occurs for at least 100 meters or more than that. In this distance the traffics police can't hear the siren form the ambulance so unfortunately he ignores this. Then the ambulance has to wait till the traffic is there. Some times to leave the traffic, it takes at least 30 minutes. So by this time anything can happen to the patient. So this project avoids such situations.

Basically in this project if any ambulance comes near to any traffic post the traffic control system automatically stops the ongoing signals (completing its predefined delay) and gives green signal to lane in which ambulance is detected.

The road accidents in modern urban areas are increased to uncertain level. The loss of human life due to accident is to be avoided. Traffic congestion and tidal flow are major facts that cause delay to ambulance. To bar loss of human life due to accidents we introduce a scheme called ITLS (Intelligent Traffic Light system). The main theme behind this scheme is to provide a smooth flow for the emergency vehicles like ambulance to reach the hospitals in time and thus minimizing the delay caused by traffic congestion. The idea behind this scheme is to implement ITLS which would control automatically the traffic lights in the path of the ambulance. The ambulance is controlled by the control unit which furnishes adequate route to the ambulance and also controls the traffic light according to the ambulance location and thus reaching the hospital safely.

There is another issue related to patient's health, many times critical parameters of patient, like heartrate, temperature etc. which are to be measured continuously, may play vital role while monitoring patient for doctors. To overcome such requirements, we have internet connectivity and inbuilt sensors in the systems with the help of that we can send data to respective hospital. As well as blood group can be informed to hospital in advance so that required resources may get collected before the patient reaches hospital.

2. HARDWARE & SOFTWARE USED

2.1 COMPONENT SPECIFICATION

i. Arduino Nano

ii. ESP8266

iii. Heart bit pulse sensor

iv. LM35

v. GPS module

vi. RFID module and tag

vii. Power Supply 5V, 3.3V

2.2 SOFTWARE USED

i. Arduino IDE

ii. Proteus

iii. Blynk

3. BLOCK DIAGRAMS

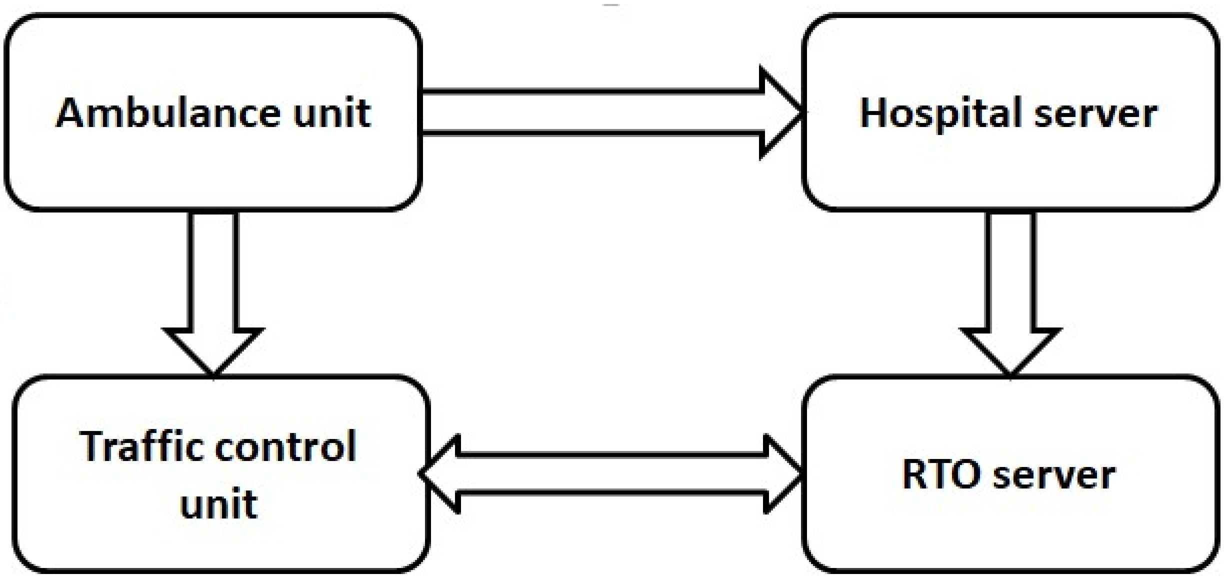

Fig No 3.1 OVERALL UNITS IN PROPOSED SYSTEM

3.1. AMBULANCE UNIT

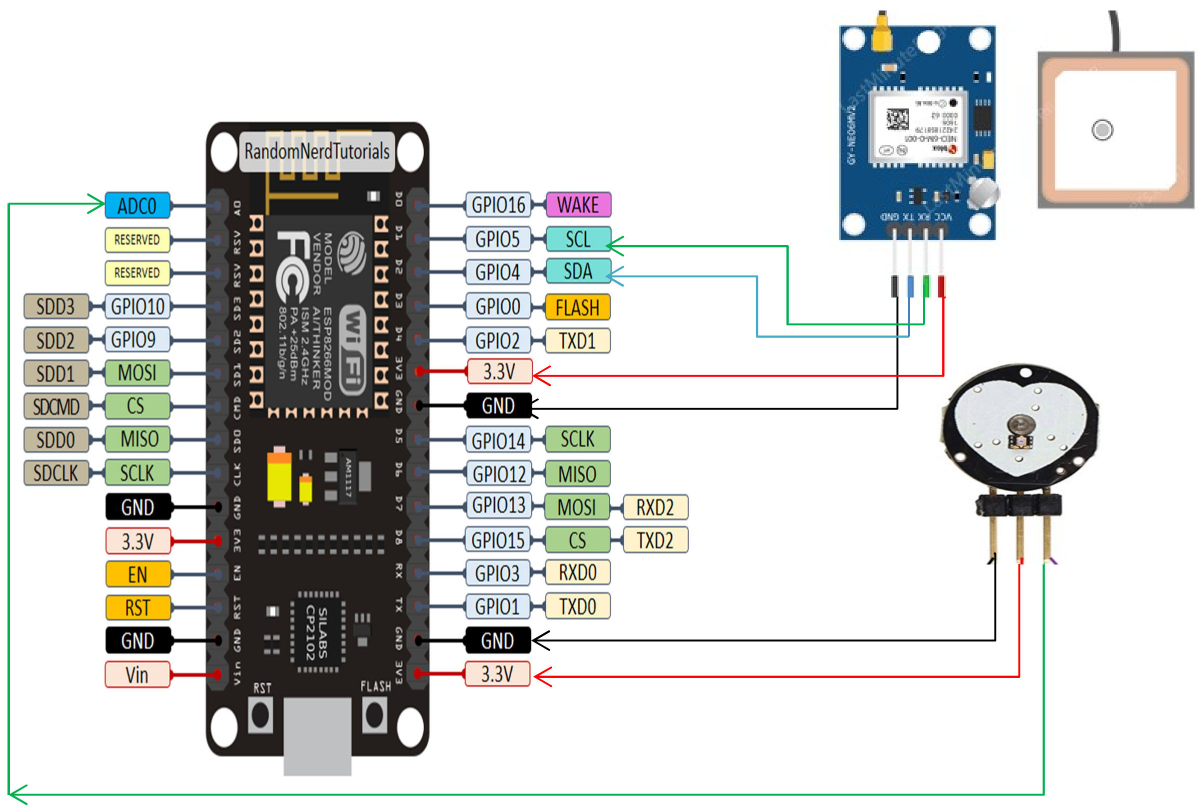

Fig No 3.2 INTERFACING OF SENSOR IN AMBULANCE UNIT

3.2. TRAFFIC CONTROL UNIT

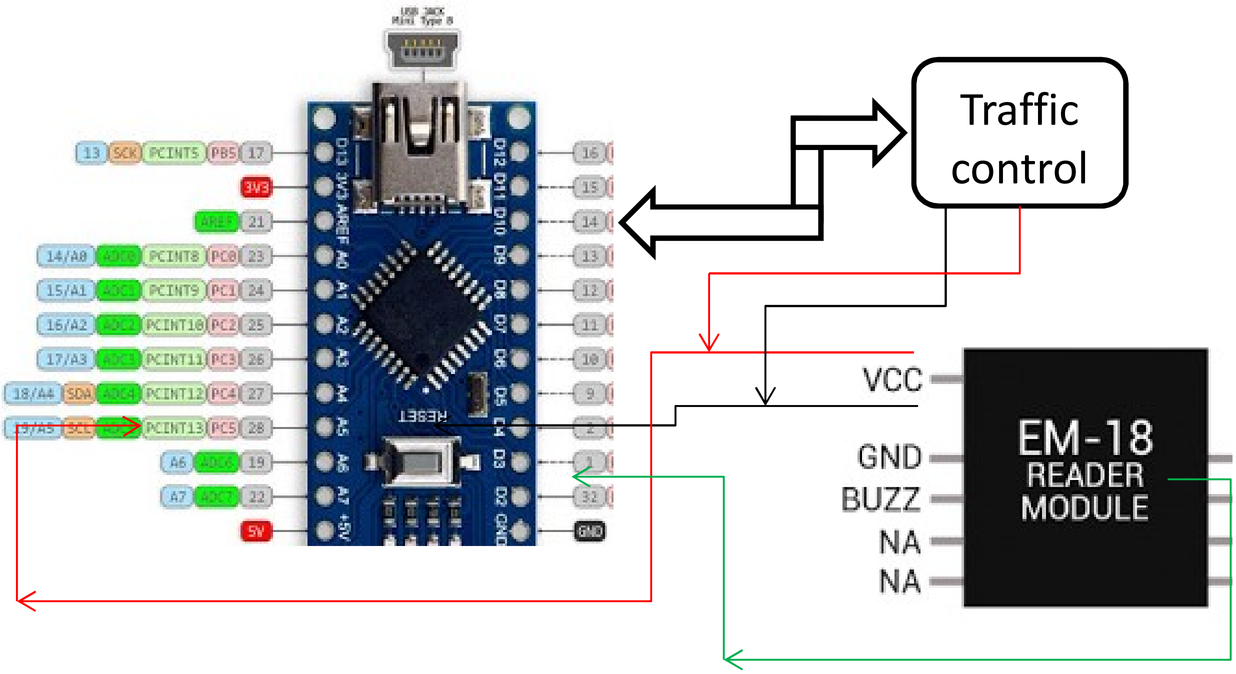

Fig No 3.3 INTERFACING OF SENSOR IN TRAFFIC CONTROL UNIT

3.2. HOSPITAL SERVER & RTO SERVER

This is the virtual System through which we can access real time data of patient & ambulance locations to determine paths also with the help of lot much data traffic conditions in day to day life can be adjusted.

Also, if any crisis occurs then RTO will have some enough data so that this crisis can be easily sorted.

4. SCOPE OF PROJECT IN REAL LIFE

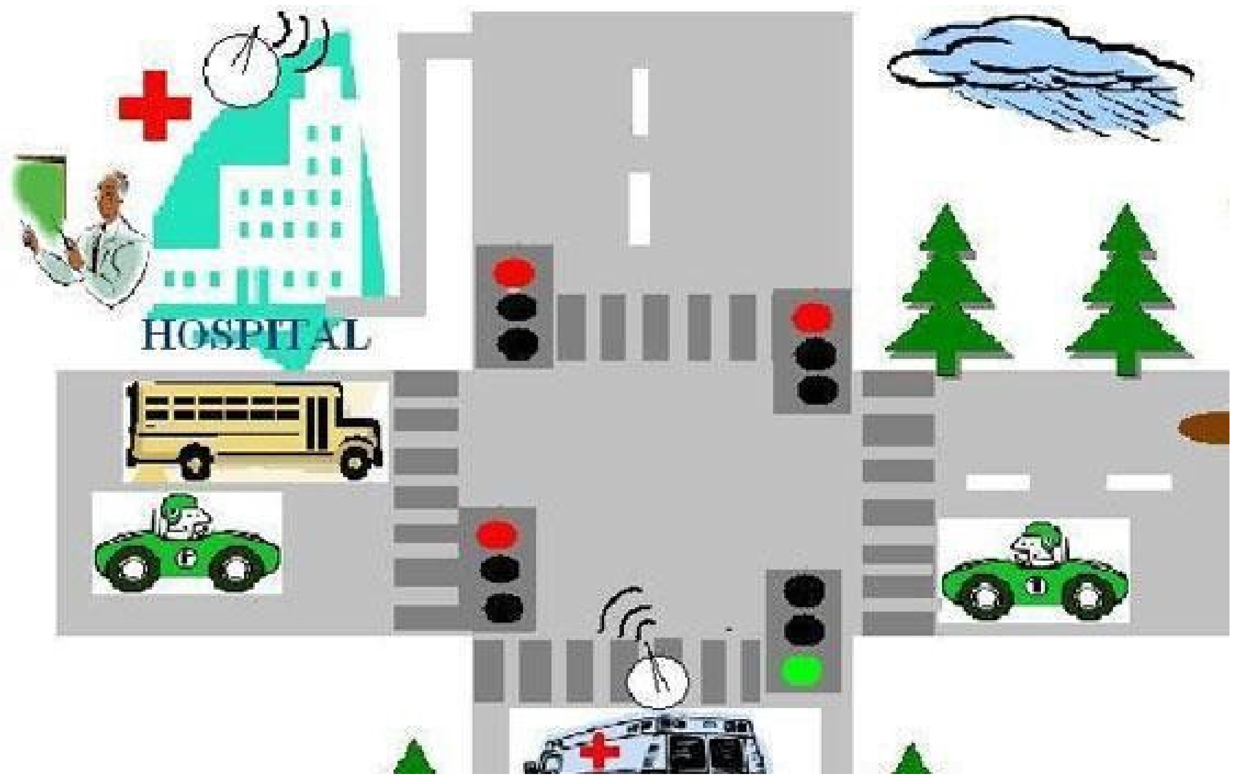

Fig No 4.1 PROPOSED SYSTEM

When an ambulance is called for emergency by patient, driver will select nearest hospital and then ambulance current location will be transmitted to the desired hospital selected by driver.

Now in street light, RFID receiver module is fixed in such a way so that it will receive or detect the passage of the ambulance. Once ambulance is detected traffic signal in particular direction is turned to green.

5. FLOW CHART

5.1. AMBULANCE UNIT

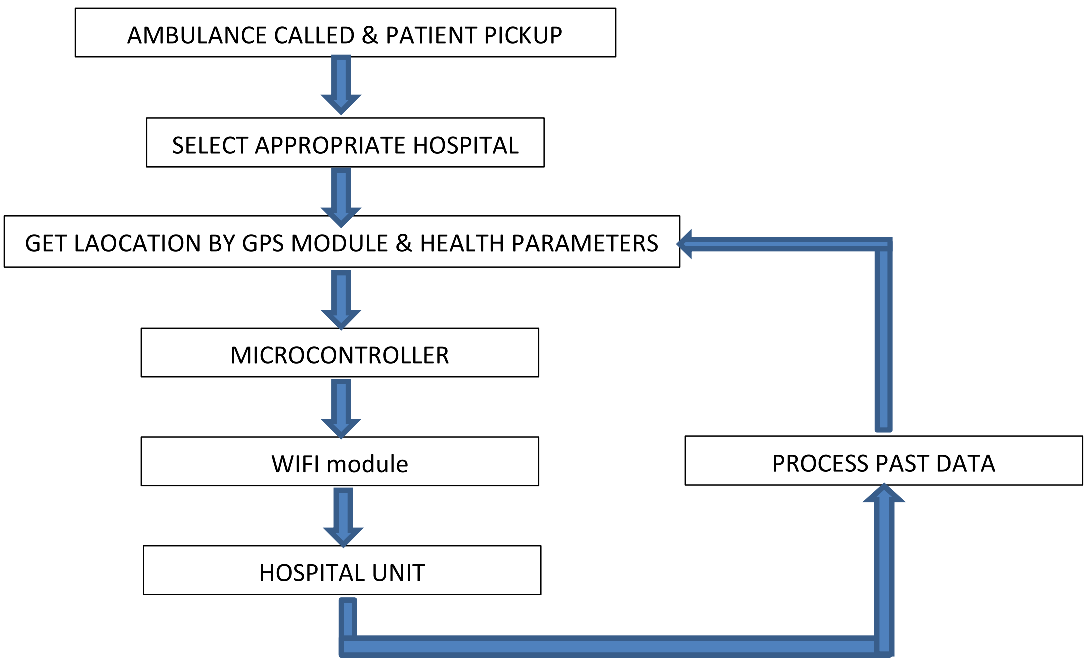

Fig No 5.1 Ambulance Unit

5.2. TRAFFIC CONTROL UNIT

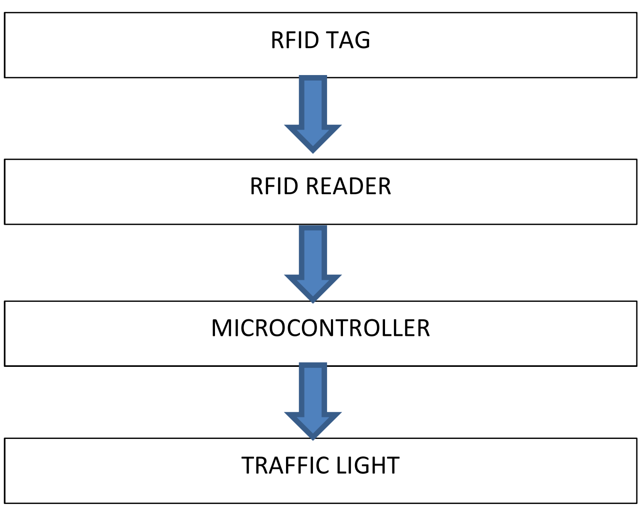

Fig No 5.1 Traffic Unit

6. SOURCE CODE

6.1. AMBULANCE UNIT

```
#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.
#include <PulseSensorPlayground.h>          // Includes the PulseSensorPlayground Library.



#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

static const int RXPin = 4, TXPin = 5;   // GPIO 4=D2(conneect Tx of GPS) and GPIO 5=D1(Connect Rx of GPS
static const uint32_t GPSBaud = 9600; //if Baud rate 9600 didn't work in your case then use 4800
const int PulseWire = A0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.
                               // Use the "Gettting Started Project" to fine-tune Threshold Value beyond default setting.
                               // Otherwise leave the default "550" value.

TinyGPSPlus gps; // The TinyGPS++ object
WidgetMap myMap(V0);  // V0 for virtual pin of Map Widget
```

```
SoftwareSerial ss(RXPin, TXPin);  // The serial connection to the GPS device

BlynkTimer timer;
PulseSensorPlayground pulseSensor;     // Creates an instance of the
PulseSensorPlayground object called "pulseSensor"


float spd;       //Variable  to store the speed
float sats;      //Variable to store no. of satellites response
String bearing;  //Variable to store orientation or direction of GPS

char auth[] = "b7hxnLsTJASIai1BMul2D6Hc5IzsrfLS";          //Your Project
authentication key
char ssid[] = "WIZARD";                        // Name of your network (HotSpot or
Router name)
char pass[] = "india@11";                     // Corresponding Password



//unsigned int move_index;         // moving index, to be used later
unsigned int move_index = 1;       // fixed location for now


void setup()
{
  Serial.begin(115200);
  Serial.println();
  ss.begin(GPSBaud);
  Blynk.begin(auth, ssid, pass);
```

```
    // Configure the PulseSensor object, by assigning our variables to it.
  pulseSensor.analogInput(PulseWire);
  pulseSensor.setThreshold(Threshold);

  timer.setInterval(5000L, checkGPS); // every 5s check if GPS is connected, only really
needs to be done once
    timer.setInterval(1000L, bpm);
}

void checkGPS(){
  if (gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
      Blynk.virtualWrite(V4, "GPS ERROR");  // Value Display widget  on V4 if GPS not
detected
  }
}

void loop()
{

    while (ss.available() > 0)
    {
      // sketch displays information every time a new sentence is correctly encoded.
      if (gps.encode(ss.read()))
        displayInfo();
  }
  Blynk.run();
  timer.run();
}
```

```
void displayInfo()
{

 if (gps.location.isValid() )
 {

   float latitude = (gps.location.lat());     //Storing the Lat. and Lon.
   float longitude = (gps.location.lng());

   Serial.print("LAT:  ");
   Serial.println(latitude, 6);  // float to x decimal places
   Serial.print("LONG: ");
   Serial.println(longitude, 6);
   Blynk.virtualWrite(V1, String(latitude, 6));
   Blynk.virtualWrite(V2, String(longitude, 6));
   myMap.location(move_index, latitude, longitude, "GPS_Location");
   spd = gps.speed.kmph();               //get speed
     Blynk.virtualWrite(V3, spd);

     sats = gps.satellites.value();    //get number of satellites
     Blynk.virtualWrite(V4, sats);

     bearing = TinyGPSPlus::cardinal(gps.course.value()); // get the direction
     Blynk.virtualWrite(V5, bearing);

      Blynk.virtualWrite(V7, "MH12 AB XXXX");

 }


 Serial.println();
```

```
}

void bpm()
{
 int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM as an "int".
                                               // "myBPM" hold this BPM value now.

if (pulseSensor.sawStartOfBeat()) {            // Constantly test to see if "a beat happened".
 Serial.println("♥  A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".
 Serial.print("BPM: ");                        // Print phrase "BPM: "
 Serial.println(myBPM);                        // Print the value inside of myBPM.
/* if(myBPM<L_hr || myBPM>H_hr )
 {
  abnormal_hr=1;
  Serial.println(abnormal_hr);

 }*/
}
  delay(20);                    // considered best practice in a simple sketch.
  Blynk.virtualWrite(V6, myBPM);
}
```

6.2. TRAFFIC CONTROL UNIT

```
#include <SoftwareSerial.h>

SoftwareSerial R1(6, 7);

SoftwareSerial R2(4, 5);

SoftwareSerial R3(2, 3);

SoftwareSerial R4(0, 1); //RX=0 TX=1


const int L1 = 8;

const int L2 = 9;

const int L3 = 10;

const int L4 = 11;

const char pass[]="1234567890ab";


void setup() {

  // put your setup code here, to run once:

Serial.begin(9600);

R1.begin(9600);

R2.begin(9600);

R3.begin(9600);

R4.begin(9600);

digitalWrite(L1,LOW);

digitalWrite(L2,LOW);

digitalWrite(L3,LOW);

digitalWrite(L4,LOW);
```

```
}

void loop() {
  int i,count;
 while(1)
{
  R1.listen();
  if(R1.available())
  {
    count=0;
    for(i=0;i<12;i++)
    {
      if(pass[i]== R1.read())
      count++;
    }
    if(count>=11)
    {
      digitalWrite(L1,HIGH);
      digitalWrite(L2,LOW);
      digitalWrite(L3,LOW);
      digitalWrite(L4,LOW);
    }
  }

  R2.listen();
```

```
if(R2.available())
{
  count=0;
  for(i=0;i<12;i++)
  {
    if(pass[i]== R2.read())
    count++;
  }
  if(count>=11)
  {
    digitalWrite(L2,HIGH);
    digitalWrite(L1,LOW);
    digitalWrite(L3,LOW);
    digitalWrite(L4,LOW);
  }
}

R3.listen();
if(R3.available())
{
  count=0;
  for(i=0;i<12;i++)
  {
    if(pass[i]== R3.read())
    count++;
```

```
  }
  if(count>=11)
  {
    digitalWrite(L3,HIGH);
    digitalWrite(L1,LOW);
    digitalWrite(L2,LOW);
    digitalWrite(L4,LOW);
  }
}
R4.listen();
if(R4.available())
{
  count=0;
  for(i=0;i<12;i++)
  {
    if(pass[i]== R4.read())
    count++;
  }
  if(count>=11)
  {
    digitalWrite(L4,HIGH);
    digitalWrite(L1,LOW);
    digitalWrite(L2,LOW);
    digitalWrite(L3,LOW);
}}}}
```

7. ADVANTAGES

i. Improves traffic flow for emergency vehicle.
ii. Possible to serve patient according to doctor's suggestions in ambulance.
iii. Saves patient's life in critical cases.
iv. Preparation can be done before reaching the ambulance at the hospital.

FUTURE SCOPE

i. Ambulance can be encrypted by hospital for safety.

ii. By using worldwide network ambulance can be passed from any traffic post.

iii. Operation Theater can be inbuilt inside the ambulance.

iv. 4G module can be used for faster network.

v. By using higher level GPS module, accuracy can increased

CONCLUSION

The following paper will help to assuage blockage of emergency vehicles in traffic by the usage of electronics and communication technique.

The most common use of these systems is to manipulate traffic signals in the path of an emergency vehicle, stopping conflicting traffic and allowing the emergency vehicle right-of-way, to help reduce response times and enhance traffic safety.

The overall paper along with the process of traffic light control also shows the message and visual communication using GSM with 3G connection in Ambulance.

REFERENCE

[1] Claudio De Capua, Antonella Meduri, and Rosario Morello. "*A Smart ECG Measurement System Based on Web-Service Oriented Architecture for Telemedicine Applications*" IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 59, NO. 10, OCTOBER 2010.

[2] Ankit Jha, Lalit Kanwar, Mayur Solanki, Shyam Sunder Joshi, Smt. Sarita Chauhan "*An Advance Intelligent Ambulance With Online Patient Monitoring System*" , IPASJ International Journal of Electronics & Communication (IIJEC) Volume 3, Issue 4, April 2015 ISSN 2321-5984.

[3] Md. Manirul Islam, Fida Hasan Md. Rafi, Mohiuddin Ahmad and Abu Farzan Mitul,T. M. N. Tunku Mansur, M. A. Rashid "*Microcontroller Based Health Care Monitoring System Using Sensor Network*", 2012 7th International Conference on Electrical and Computer Engineering 20-22 December, 2012, Dhaka, Bangladesh.

[4] Bor-Shyh Lin, Willy Chou, Hsing-Yu Wang, Yan-Jun Huang and jeng-Shyang Pan, "*Development of Novel Non-Contact Electrodes for Mobile Electrocardiogram Monitoring System*", IEEE journal of Translational Engineering in Health and Medicine, Digital Object Identifier 10.1109/JTEHM.2013.2253598.

[5] Rajeshwari Sundar, Santhoshs Hebbar, and Varaprasad Golla "*Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection*", IEEE sensor journal, VOL. 15, NO. 2, FEBRUARY 2015.

[6] Sarika B. Kale, Gajanan P. Dhok "*Design of Intelligent Ambulance and Traffic Control*" , International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-5, April 2013.

[7] Bharadwaj. R, Deepak.J, Baranitharan. M, V. Vaidehi "*Efficient Dynamic Traffic Control System using Wireless Sensor Networks*" ,2013 International Conference on Recent Trends in Information Technology (ICRTIT).

[8] Soufiene Djahel, Mazeiar Salehie, Irina Tal and Pooyan Jamshidi , "*Adaptive Traffic Management for Secure and Efficient Emergency Services in SmartCities*" Work in progress session at Percom 2013, San Diego,19March2013.