RFID and GPS based Automatic Lane Clearance System for Ambulance

Rashmi Hegde, Rohith R. Sali & M. S. Indira

Sir MVIT

E-mail : raregem24@gmail.com, rohi265@gmail.com

Abstract - The exponential growth of the metropolitan cities of the country has generated and magnified urban sprawl into problematic proportions. Lack of efficient traffic control and management has many a times lead to loss of lives due to ambulances getting stuck in traffic jams. To overcome this problem, we propose a RFID and GPS based Automatic Lane Clearance System for Ambulance. The focus of this paper is to reduce the delay in arrival of the ambulance to the hospital by automatically clearing the lane in which ambulance is travelling, before it reaches the traffic signal. This can be achieved by turning the traffic signal, in the path of the ambulance, to green when the ambulance is at a certain distance from the traffic junction. The use of RFID distinguishes between the emergency and non-emergency cases, thus preventing unnecessary traffic congestion. The communication between the ambulance and the traffic signal post is done through transceivers and GPS. The system is fully automated and thus, requires no human intervention at the traffic junctions.

Index Terms— Ambulance, GPS, RFID, Traffic signal, Transceiver

I. INTRODUCTION

Bangalore has grown exponentially in the past two decades. Improvement in the quality of life along with substandard public transportation has resulted in spiralling growth of private automobiles. The resultant offshoot of such a high automobile growth is that now Bangalore is one of the most accident-prone cities in India [1].Moreover, the ambulances often get stuck at the traffic signals where all other vehicles try to squeeze in to all the available space so as to move ahead as soon as the signal turns green. Unlike western countries, Indian cities cannot think of having separate lanes for emergency purpose due to lack of road planning and infrastructure. With the lives of the patients depending on the speedy arrival of the ambulances to hospital, an alternative solution to the above problem is the need of the hour.

The problem of ambulance getting stuck in a traffic jam can be addressed by ensuring that the lane in which the ambulance is travelling is cleared. That is, the arrival of the ambulance is to be communicated to the nearest traffic signal, so that it can turn the light to green and hence clear the traffic. However, all the ambulances will not be carrying emergency cases. Hence, the traffic clearing system, if done for all the ambulances, will certainly pose a traffic problem. To overcome this difficulty, we propose to make a system combining RFID (Radio Frequency Identification) and GPS.

II. BACKGROUND

The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather, anywhere on or near the Earth, where there is an unobstructed line of sight to four or more GPS satellites. (2) A GPS receiver's job is to locate four or more of these satellites, figure out the distance to each, and use this information to deduce its own location.

The Radio Frequency Identification (RFID) usually refers a whole system, which usually contains three main parts: The RFID tag, RFID Reader, and the Application. RFID tag refers to small electronic devices that consist of a small chip and an antenna. The chip typically is capable of carrying some kind of information depending on applications. The RFID readers, also called interrogators, are used to recognize the presence of nearby RFID tags. The *Table (i)* shows the read range for different frequencies of passive RFID. (3)

K.Athavan *et al* have proposed an ambulance rescue system using GSM and GPS [4]. Fang-Yie et al suggested a traffic control scheme where the transportation authority and the ambulance communicate using encrypted messages [5].

III. RFID AND GPS BASED AUTOMATIC LANE CLEARANCE SYSTEM FOR AMBULANCE:

The proposed system consists of two units. Namely:

1. Ambulance Unit
2. Junction Unit

Ambulance Unit:

The Ambulance Unit, to be installed in the ambulance, consists of an RFID reader, GPS receiver and a transceiver interfaced with a microcontroller. The GPS receiver continuously receives the GPS co-ordinates of the ambulance by calculating its position using the timing signals from the GPS satellites. When an ambulance leaves the hospital for an emergency case, a RFID card is swiped near the RFID reader, which when authenticated activates the transmission of GPS co-ordinates through the transceiver. *Table (ii*) lists out the components used for the Ambulance Unit. *Fig. (a)* shows the flowchart of the Ambulance Unit.

Name	**Frequency Range**	**Typical max. read range for passive tags**
Low Frequency (LF)	30-300 kHz	50 cm
High Frequency (HF)	3-30 MHz	3 m
Ultra High Frequency (UHF)	300 MHz-3 GHz	9m
Microwave	>3 GHz	>10m

Table (i)

Sl. No.	**Components**	
1.	Microcontroller	Arduino Uno[6]
2.	Transceiver	Xbee S2[7]
3.	RFID Reader	EM-18[9]
4.	GPS Receiver	

Table (ii)

Program for the Ambulance Unit:

```
#include <SoftwareSerial.h>
#include<string.h>
#define tx 2
#define rx 3
floatlonf=0,latf=0,altf,t;
char
ch,data[600],format[6],valid=0,lon[10],lat[10],alt[6],j,k,
l=0,time[7];
SoftwareSerialgps(tx,rx);
int i=0;
inttxhardware=1;
intrxhardware=0;
void setup()
{
Serial.begin(9600);
gps.begin(9600);
}
void loop()
{
gps.listen();
for(i=0;i<600;)
  {
if (gps.available())
    {
data[i++]=gps.read();
  }
for(i=0;i<600;i++)
  {
if (data[i]=='$')
    {
for(j=0;j<5;j++)
format[j]=data[++i];
if (strcmp(format,"GPRMC")==0)
      {
if (data[i+13]=='A')
valid=1;
else if (data[i+13]=='V')
Serial.println("waiting for valid data");
        }
else if (strcmp(format,"GPGGA")==0 && valid==1)
      {
for(j=0,i+=13,l=0;j<9;j++,i++)
        {
if (data[i]>='0'&&data[i]<='9')
          {
lat[l++]=data[i];
          }
        }
  latf=(lat[0]-'0')*10.0+(lat[1]-'0')+((lat[2]-
'0')*10.0+(lat[3]-'0')+(lat[4]-'0')/10.0+
(lat[5]-'0')/100+(lat[6]-'0')/1000.0+(lat[7]-
'0')/10000.0)/60;
Serial.print("$");
Serial.println(latf,6);
for (j=0,i+=3,l=0;j<10;j++,i++)
```

```
        {
if (data[i]>='0'&&data[i]<='9')
         {
lon[l++]=data[i];
         }
        }
        lonf=(lon[0]-'0')*100.0+(lon[1]-'0')*10.0+(lon[2]-
'0')+((lon[3]-'0')*10.0+(lon[4]-'0')+(lon[5]-
'0')/10.0+(lon[6]-'0')/100+(lon[7]-'0')/1000.0+(lon[8]-
'0')/10000.0)/60.0;
Serial.print("#");
Serial.println(lonf,6);
valid=0;
      }
    }
    }
    }
    }
    }
}}
```

B. Junction Unit:

The Junction Unit, to be installed at the traffic signal post, consists of a transceiver interfaced with a microcontroller. The GPS co-ordinates transmitted by the Ambulance Unit are received by the transceiver. The co-ordinates of a point at a particular distance are specified in the Junction Unit's microcontroller program, which when crossed by the ambulance turns the traffic signal green. At the same time, a LED screen displays the message that an ambulance is approaching, indicating the motorists to give way.The distance of the point from the signal may differ according to the traffic scenario across different junction and may be programmed as per its need. A suitable delay is given for the signal to remain green till the ambulances passes through the junction. A four-lane junction is assumed in this case. In case of presence of ambulance at more than one lane of the junction, the ambulance which arrives first is given priority. *Table (iii)* lists out the components used for the Junction Unit. *Fig. (b)* shows the flowchart of the Junction Unit.

Sl. No.	**Components**	
1.	Microcontroller	Arduino Uno[6]
2.	Transceiver	Xbee S2[7]
3.	LED Display	JHD 162A[8]

Table (iii)

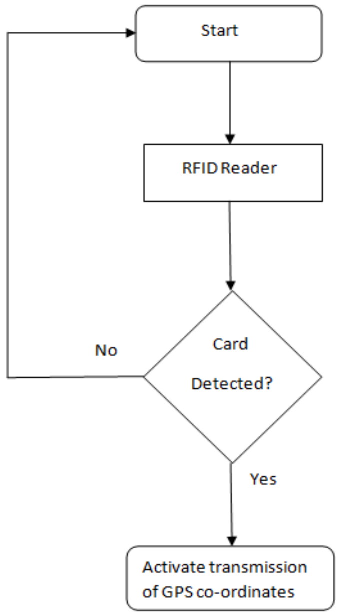


Fig. (a). Flowchart of Ambulance Unit

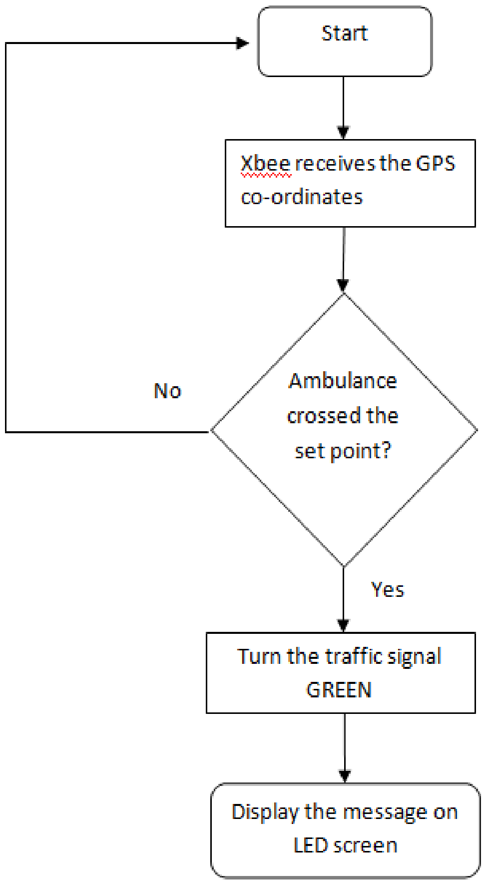


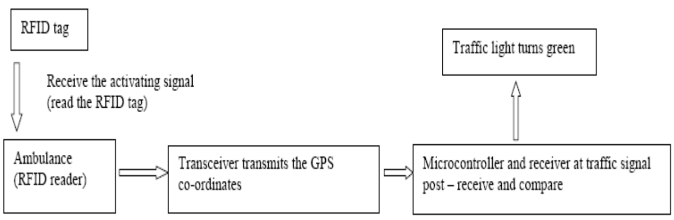
Fig. (b). Flowchart of Junction Unit

Program for Junction Unit:

```
#include <LiquidCrystal.h>

inttx=1;

intrx=0;

floatlonf=0,latf=0,altf,t;

char ch,data[600],format[6],valid=0,lon[10],lat[10],alt[6],j,k,l=0,time[7];

const float x1=13.128487,x2=13.128355,x3=13.028355,x4=13.171167;

const float y1=77.41352,y2=77.43303,y3=77.543303,y4=77.6123331;

const float klat=0.00034,klon=0.00033;

constintledup =8,leddown =9,ledleft =10,ledright=7;

int i=0,trigger=0;

LiquidCrystallcd(12, 11, 5, 4, 3, 2);


void setup()

{

Serial.begin(9600);

lcd.begin(16, 2);

lcd.setCursor(0, 1);

pinMode(rx,INPUT);
pinMode(leddown, OUTPUT);
pinMode(ledup, OUTPUT);
pinMode(ledleft, OUTPUT);
pinMode(ledright, OUTPUT);
lcd.print("Amb");
}
void loop()
{
Serial.flush();
for(i=0;i<50;)
{
if (Serial.available())
  {
data[i]=0;
data[i++]=Serial.read();
Serial.print(data[i]); //testing
  }
 }
for(i=0;i<50;i++)
 {
if (data[i]=='$')
 {
for(j=0,l=0;j<20;j++,i++)
   {
if (data[i]>='0'&&data[i]<='9')
    {
lat[l++]=data[i];
    }
if(data[i]=='\n')
break;
   }
  latf=(lat[0]-'0')*10.0+(lat[1]-'0')+(((lat[2]-'0')*10.0+(lat[3]-'0')+(lat[4]-'0')/10.0+
(lat[5]-'0')/100+(lat[6]-'0')/1000.0+(lat[7]-'0')/10000.0)/100.0);
Serial.print("LAT=");
Serial.println(latf,6);
  }


if (data[i]=='#')
  {
i++;
for(j=0,l=0;j<20;j++,i++)
   {
if (data[i]>='0'&&data[i]<='9')
    {
lon[l++]=data[i];
    }
```

```
if(data[i]=='\n')
break;
      }
      lonf=(lon[0]-'0')*10.0+(lon[1]-'0')+((lon[2]-'0')*10.0+(lon[3]-'0')+(lon[4]-'0')/10.0+
(lon[5]-'0')/100+(lon[6]-'0')/1000.0+(lon[7]-'0')/10000.0)/100.0;
Serial.print("LON=");
Serial.println(lonf,6);
break;
    }
  }
while (latf<=x1+klat &&latf>=x1-klat)
  {
if (lonf<=y1+klon || lonf>=y1-klon)
    {
digitalWrite(ledup,LOW);
digitalWrite(leddown,HIGH);
digitalWrite(ledleft,LOW);
digitalWrite(ledright,LOW);
trigger=1;
    }
break;
  }
while ( lonf<=y2+klon &&lonf>=y2-klon)
  {
if (latf<=x2+klat || latf>=x2-klat)
    {
digitalWrite(ledup,LOW);
digitalWrite(leddown,LOW);
digitalWrite(ledleft,HIGH);
digitalWrite(ledright,LOW);
trigger=1;
    }
break;
  }
while (lonf<=y3+klon &&lonf>=y3-klon)
  {
if (latf<=x3+klat || latf>=x3-klat )
    {
digitalWrite(ledup,HIGH);
digitalWrite(leddown,LOW);
digitalWrite(ledleft,LOW);
digitalWrite(ledright,LOW);
trigger=1;
    }
break;
  }
while (lonf<=y4+klon &&lonf>=y4-klon)
  {
if (latf<=x4+klat || latf>=x4-klat)
    {
digitalWrite(ledup,LOW);
digitalWrite(leddown,LOW);
digitalWrite(ledleft,LOW);
digitalWrite(ledright,HIGH);
trigger=1;
    }
break;
 }
if(trigger==1)
  {
delay(50000);
  }
trigger=0;
}
```

c. Block Diagram of the complete system:



IV. LIMITATION

The drawback of this arrangement is that the Government ambulances, like the 108s, do not have a particular place from which they regularly leave to pick up the patients. Hence, the implementation of this system is difficult for Government ambulances. However, the proposed system can be easily implemented for the ambulances in service of hospitals.

V. REFERENCES

[1] Mahesh Chand: Accident Scenario in Metropolitan Cities of India- published in Urban Transport journal-September2002

[2] http://en.wikipedia.org/wiki/Global_Positioning_System

[3] RFID Essentials, by Bill Grover and Himanshu Bhatt, O'Reilly

[4] K. Athavan *et al*, International Journal of Advanced Technology & Engineering Research, Vol. 2, Issue 2, May 2012.

[5] Fang-yieu et al, Controlling Traffic Lights for Ambulances, 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Application.

[6] Arduino Uno Datasheet: http://arduino.cc/en/Main/arduinoBoardUno

[7] Xbee S2 Transceiver Datasheet: ftp://ftp1.digi.com/support/documentation/90000866_A.pdf

[8] LED Display Datasheet: http://pdf1.alldatasheet.com/datasheet-pdf/view/127934/ETC1/JHD162A.html

[9] EM-18 RFID Reader Datasheet: http://www.jayconsystems.com/fileuploader/download/download/?d=1&file=custom%2Fupload%2FFile-1341977419.pdf

❖❖❖