IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 3, Ver. V (May-Jun. 2016), PP 07-22
www.iosrjournals.org

Vehicle Security System with Theft Identification and Accident Notification

Anagha Kotre
(M.C.A.) Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application University of Mumbai

***Abstract*:** *The rapid development of electronics provides secured environment to the human. As a part of this 'Vehicle Security System With Theft Identification And Accident Notification' is designed to reduce the risk involved in losing the vehicle and providing accident notification which will reduce the rate of deaths. This tracking system is composed of a GPS receiver, Microcontroller and a GSM Modem. GPS Receiver gets the location information from satellites in the form of latitude and longitude. This is an inexpensive device which reduces the problem associated with accident notification and antitheft control.*

Keywords: *GPS receiver,Microcontroller GSM Modem*

I. Introduction

This device presents an automotive localization system using GPS and GSM SMS services. The system permits localization of the automobile and transmitting the position to the owner on his mobile phone as a short message (SMS) at his request. The system can be interconnected with the car alarm system and alert the owner on his mobile phone. This tracking system is composed of a GPS receiver, Microcontroller and a GSM Modem. GPS Receiver gets the location information from satellites in the form of latitude and longitude. The Microcontroller processes this information and this processed information is sent to the user/owner using GSM modem.

The presented application is very useful in case of car theft situations, for monitoring adolescent drivers by their parents as well as in car tracking system applications. This enable it to monitor the accident situations and it can immediately alerts the police/ambulance service with the location of accident.

II. Proposed System

Working:

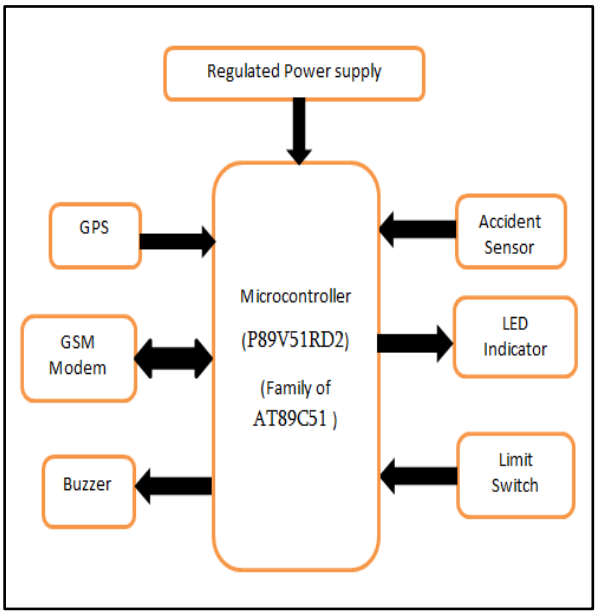

Fig.1(Vehicle Security Block diagram)

In this device the GPS is used to provide the exact position of the vehicle. The information that is collected by the GPS modem is passed to the microcontroller on its request. The information provided by the GPS system contains longitudinal and latitude positions.It also provides the speed and time of the vehicle.
Here we use AT89C51 microcontroller. It mainly controls the all function of the project. It gets the information for the GPS modem and passed it to the GSM modem. It controls the accident sensor, limit switch and led

indicator. GSM modem is used to send messages to the predefined numbers stored in the microcontroller. This GSM modem uses AT commands in order to send messages to the predefined number. This project has three features such as follows.

1. If an unauthorized person tries to open the door then LED gets on and simultaneously we can hear buzzer.
2. In order to identify accident we have used accident sensor that will detect the accident and the device will send the latitude and longitude of the accident location to the predefined number.
3. Whenever we find that our vehicle is missing then simply send a SMS to GSM Modem from the authorized mobile .Then the Microcontroller gets the location of vehicle and sends the message to the same mobile. The message includes the location at which the vehicle is present at that particular instant of time.

Description of components

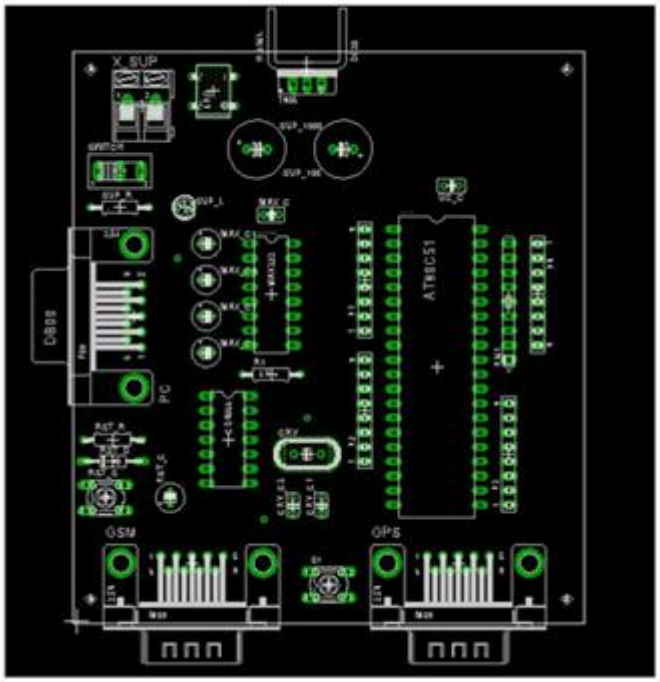

Fig.2 (Device which contains placing of all the components)

- **Hardware**

- **Power Supply:** Power supply is a reference to a source of electrical power. Power supply is the major concern for every electronic device .Since the controller and other devices used are low power devices there is a need to step down the voltage and as well as rectify the output to convert the output to a constant dc.
- **Microcontroller:** The P89V51RD2 is an 80C51 microcontroller with 64 kB Flash and 1024 bytes of data RAM. A key feature of the P89V51RD2 is its X2 mode option. The Flash program memory supports both parallel programming and in serial In-System Programming (ISP). ISP allows a device to be reprogrammed in the end product under software control. The capability to field/update the application firmware makes a wide range of applications possible. The P89V51RD2 is also In-Application Programmable (IAP), allowing the Flash program

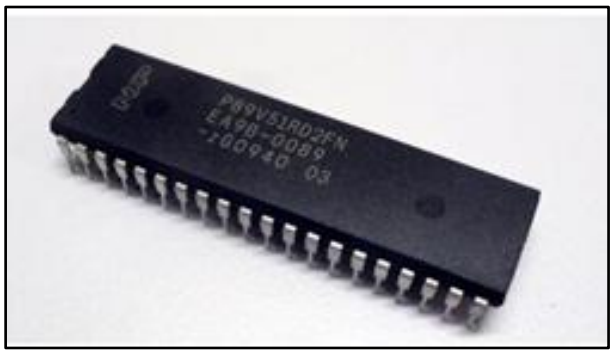

Fig.3(Microcontroller) memory to be reconfigured even while the application is running.

- **Buzzer:** It is used for the theft indication. If an unauthorized person try to open the door then LED indicator which is placed inside the car will blink and along with that buzzer gets ON.

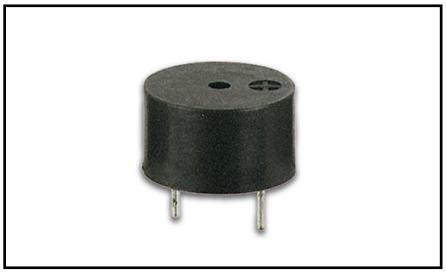

Fig.4 (Buzzer)

- **Limit_Switch: -** A limit Switch detects the physical motion of an object by directly contact with that object. It is used to detect when unauthorized person try to open the door.

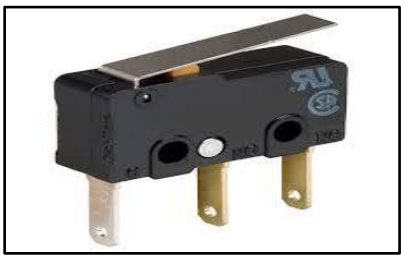

Fig.5 (Limit Switch)

- **Accident_Sensor**: Accident of a vehicle is identified by using this sensor.Whenever the accident will happen the information (Latitude and Longitude) will be sent to the GSM modem.

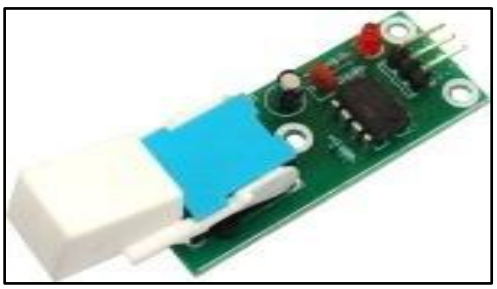

Fig.6 (Accident Sensor)

- **GSM_Modem**: GSM network is used to provide communication from one place to other. The GSM module using consists of a mobile station. The commands that are using to provide communication is AT commands specify the GSM technology and are related to SMS service.

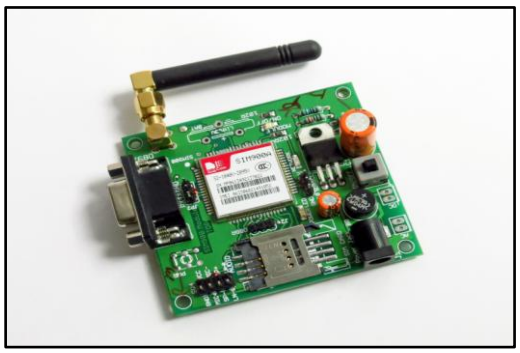

Fig.7 (GSM Modem)

- **GPS:** GPS technology is used to find the location of any object or vehicle to monitor a child continuously using satellite signals. Three satellite signals are necessary to locate the receiver in 3D space and fourth

satellite is used for time accuracy.GPS will give the information of parameters like longitude, latitude and attitude. With the help of these parameters one can easily locate the position of any object. In this GPS technology, the communication takes place between GPS transceiver and GPS satellite.

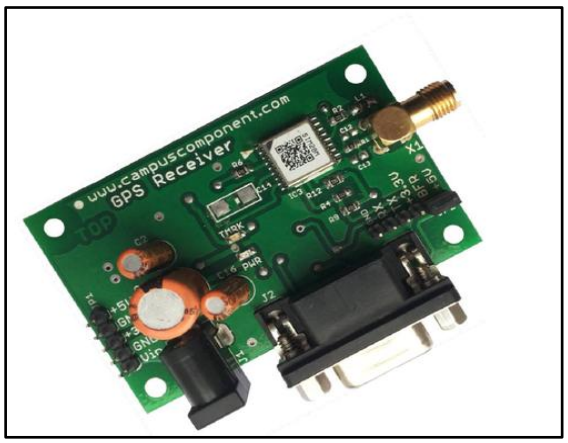

Fig.8 (GPS Modem)

- **Max232:**To convert RS (Recommended Standard) into TTL (Transistor-Transistor Logic) and vice versa. It is used for GSM, GPS and microcontroller to communicate serially

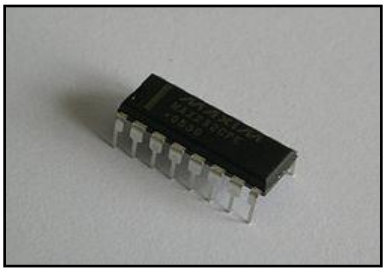

Fig.9 (Max232)

- **CD4066**:It is used to perform the work of GPS and GSM Modem simultaneously. Generally it will as a switch.

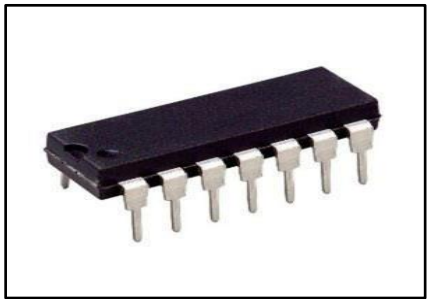

Fig.10 (CD4066)

➢ **Features**

- Wide supply voltage range
- High noise immunity
- Wide range of digital and analog switching

➢ **Software**

- **Keil's "C51" C Compiler (µVision IDE)**

These simple instructions will walk you through the process of configuring Keil's µVision IDE and C51 C Compiler with settings that produce code that can be download and run on the 8051 development board.

➢ **Step 1:** Copy Startup Code To Your Project

- **Step 2:** Edit Startup and Other Code
- **Step3:** Define a New Target Name for This Board
- **Step 4:** Configure Build Target
- **Step 5:** Compile and Download Intel-Hex Output
- **Step 6** Reconfigure For Flash ROM.

- **Eagle Software**

EAGLE (**E**asily **A**pplicable **G**raphical **L**ayout **E**ditor) by Cad soft is a flexible and expandable EDA schematic capture, PCB layout, auto router and CAM program. EAGLE contains a schematic editor, for designing diagrams. Eagle is software limited to Printed Circuit Design.

III. Circuit Diagram

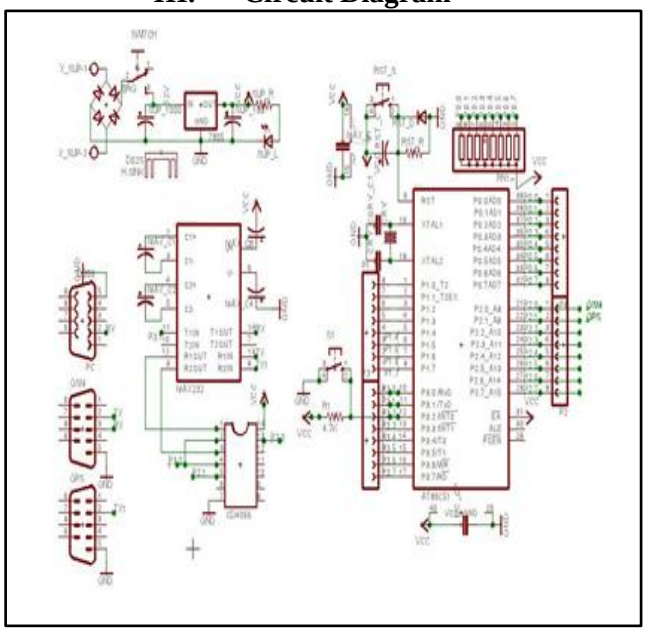

Fig.11(Circuit Diagram of the device which contains all the components)

A) Bottom-View

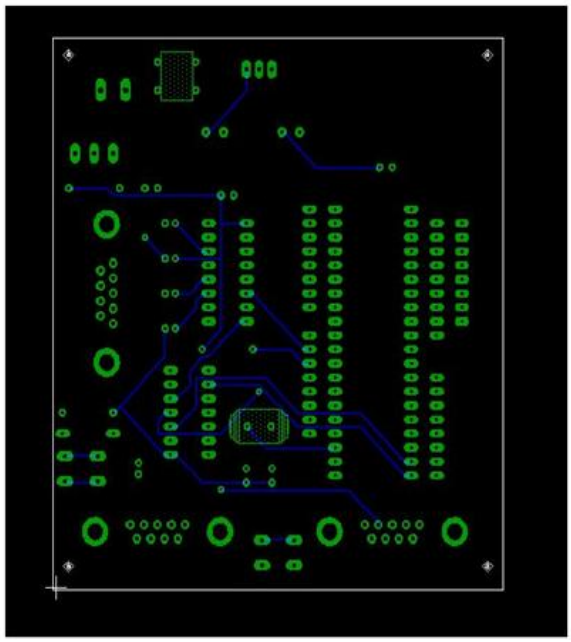

Fig.12(Bottom view of the device with all the components)

B) Top-View

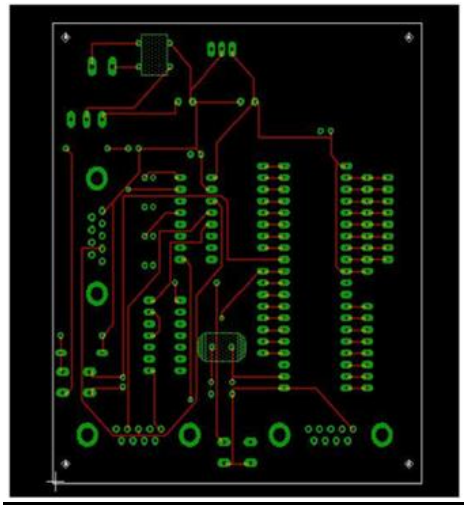

Fig.13(Top view of the device with all the components)

IV. Communication Protocols And Commands

AT commands

AT commands are instructions used to control a modem. AT is the abbreviation of Attention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems. These are also supported by GSM/GPRS modems and mobile phones. Besides this common AT command set, GSM/GPRS modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands. Extended commands are AT commands that start with "+". All GSM AT commands are extended commands. For example, +CMGS (Send SMS message), +CMSS (Send SMS message from storage), +CMGL (List SMS messages) and +CMGR (Read SMS messages) are extended commands. Here are some of the tasks that can be done using AT commands with a GSM/GPRS modem or mobile phone.

Table.1 (List of AT commands)

AT Command	Functionality
AT+CGMI	Name of the manufacture
AT+CGMM	Model number
AT+CGSN	International mobile subscriber identity
AT+CGMR	Software version
AT+CIMI	International mobile subscriber identity
AT+CSQ	Radio signal strength
AT+CBC	Charging status
AT+CMGS	Send message
AT+CMGR	Read message
AT+CMGW	Write message
AT+CMGD	Delete message
AT+CNMI	Notifications of received messages
AT+CPBR	Read phone book
AT+CPBW	Write to phone book
AT+CPBF	Search phone book
AT+CLCK	Checking whether a facility is locked
AT+CPWD	Change password
ATO	Return to online data state
ATH	Hook control
ATA	Answer call
ATD	Dial call

V. Snapshot of The Device

Fig.14 (Snapshot of the device with all the components)

VI. Advantages

- Security Purpose.
- Theft identification
- To minimize the deaths and the severe conditions due to accidents the GPS and GSM technologies are used where the immediate action would be taken placed by ambulance police services which might reduce the severity.
- This project is to find out theft vehicle easily and with the least amount of time and also we can save people from accident like to give fast treatment for them.

VII. Source Code

```
# include <reg51.h>          /////// Header file for at89c51

 sbit gsmselect = P2^0;
 sbit gpsselect = P2^1;
 sbit twist= P2^5;
 sbit buzzer=P0^0;
 sbit bump=P0^5;

 unsigned char data rx_buf[50];        //// Store SMS received
 unsigned char xdata rx_buf1[40];      //// Store GPS data
 unsigned char data tx_buf[22];    //// Array to store AT commands
 unsigned char  data data1, recd_byte_cnt, j,recd_byte_cnt1, k;
 unsigned char  data  byte_count, octet_to_send;
 unsigned int   data delay,a;
 unsigned char  data  cw, dw,p;
 unsigned char  data sms_no,t;
 unsigned char GGA,delete_5_sms_flag,loop_count; acci=0;

 bdata char  sflg1;///// Bit addressable character variable
 sbit star_flg        = sflg1^0;
 sbit starhash_flg    = sflg1^1;
 sbit all_data_recd   = sflg1^2;
```

```
 bdata sendflg;
 sbit send_location_flg=sendflg^0;
 sbit all_data_recd1   =sendflg^1;

 delay1()
{
  for (delay=0; delay<=65500; delay++);
}
delay2()
{
  for ( delay=0; delay<=5000; delay++ );
}
delay3()
{
  for (delay=0; delay<=50000; delay++)         /// 65500
  {
    if( all_data_recd ) break;
  }
}

set_echo_off()
{
  tx_buf[0] = 'a';
  tx_buf[1] = 't';
  tx_buf[2] = 'e';
  tx_buf[3] = '0';
  tx_buf[4] = 0xd;
  tx_buf[5] = 0xa;

  SBUF = tx_buf[0];
  byte_count     = 0;
  octet_to_send  = 5;
  while (byte_count != octet_to_send) {
  };
}

set_text_mode()
{
  tx_buf[0] = 'a';
  tx_buf[1] = 't';
  tx_buf[2] = '+';
  tx_buf[3] = 'c';
  tx_buf[4] = 'm';
  tx_buf[5] = 'g';
  tx_buf[6] = 'f';
  tx_buf[7] = '=';
  tx_buf[8] = '1';
  tx_buf[9] = 0xd;
  tx_buf[10] = 0xa;

  SBUF = tx_buf[0];
  byte_count     = 0;
  octet_to_send  = 10;
  while (byte_count != octet_to_send) {
  };
}
delete_sms()
{
  sflg1 = 0;
```

```
  tx_buf[0] = 'a';
  tx_buf[1] = 't';
  tx_buf[2] = '+';
  tx_buf[3] = 'c';
  tx_buf[4] = 'm';
  tx_buf[5] = 'g';
  tx_buf[6] = 'd';
  tx_buf[7] = '=';
  tx_buf[8] = j;          // sms no to be deleted
  tx_buf[9] = 0xd;                                  /// Carriage return /// to press enter
  tx_buf[10]= 0xa;                                   /// line forward

  SBUF = tx_buf[0];
  byte_count     = 0;
  octet_to_send  = 10;
  while (byte_count != octet_to_send) {
  };

  delay2();
   sflg1 = 0;
}
read_sms()
{
 sflg1 = 0;
  tx_buf[0] = 'a';
  tx_buf[1] = 't';
  tx_buf[2] = '+';
  tx_buf[3] = 'c';
  tx_buf[4] = 'm';
  tx_buf[5] = 'g';
  tx_buf[6] = 'r';
  tx_buf[7] = '=';
  tx_buf[8] = sms_no;
  tx_buf[9] = 0xd;
  tx_buf[10] = 0xa;

  SBUF = tx_buf[0];
  byte_count     = 0;
  octet_to_send  = 10;
  while (byte_count != octet_to_send) {
  };

  delay3();         // wait for response

  if( all_data_recd )
  {
           all_data_recd = 0;
  if( rx_buf[0] == '1' )     //sms received
           {
    if( rx_buf[2] == 'N' )        // ON
    {

        j = sms_no;     ////////////////////////////////
delete_sms();


     send_location_flg = 1;             /// Now send location     rx_buf[0] = 0;
rx_buf[1] = 0;
           rx_buf[2] = 0;
```

```
 rx_buf[3] = 0;

     }

 }
     else
     {
send_location_flg=0;
               }
                delay1();

   }

   rx_buf[50] = 0;

 } /// End of read_sms

void Send_Location_Sms()
{
     tx_buf[0] = 'a';
     tx_buf[1] = 't';
     tx_buf[2] = '+';
     tx_buf[3] = 'c';
     tx_buf[4] = 'm';
     tx_buf[5] = 'g';
     tx_buf[6] = 's';
     tx_buf[7] = '=';
     tx_buf[8] = '"';
    tx_buf[9] = '9'; // send sms mobile number
     tx_buf[10] = '7';
     tx_buf[11] = '7';
     tx_buf[12] = '3';
     tx_buf[13] = '4';
     tx_buf[14] = '3';
     tx_buf[15] = '6';
     tx_buf[16] = '6';
     tx_buf[17] = '3';
     tx_buf[18] = '3';
     tx_buf[19] = '"';
     tx_buf[20] = 0xd;
     tx_buf[21] = 0xa;

     SBUF = tx_buf[0];
     byte_count     = 0;
     octet_to_send  = 21;
     while (byte_count != octet_to_send) {
     };
     delay1();
tx_buf[0] = 'L';    //           lattitude
     tx_buf[1] = 'a';     //
     tx_buf[2] = 't';     //
     tx_buf[3] = ':';                          //
     tx_buf[4] = rx_buf1[11];
     tx_buf[5] = rx_buf1[12];
     tx_buf[6] = rx_buf1[13];
     tx_buf[7] = rx_buf1[14];
     tx_buf[8] = rx_buf1[15];
               tx_buf[9] = rx_buf1[16];
     tx_buf[10] = rx_buf1[17];
```

```
    tx_buf[11] = rx_buf1[18];
    tx_buf[12] = rx_buf1[19];
            tx_buf[13] = ' ';
            tx_buf[14] = rx_buf1[21];
    tx_buf[15] = 0xd;
    tx_buf[16] = 0xa;

    SBUF = tx_buf[0];
    byte_count     = 0;
    octet_to_send  = 16;
    while (byte_count != octet_to_send) {
    };
    delay1();
            tx_buf[0] = 'L';   //longitue
    tx_buf[1] = 'o';//
    tx_buf[2] = 'n';//
    tx_buf[3] = ':';//
    tx_buf[4] = rx_buf1[24];
    tx_buf[5] = rx_buf1[25];
    tx_buf[6] = rx_buf1[26];
    tx_buf[7] = rx_buf1[27];
    tx_buf[8] = rx_buf1[28];
            tx_buf[9] = rx_buf1[29];
    tx_buf[10] = rx_buf1[30];
    tx_buf[11] = rx_buf1[31];
    tx_buf[12] = rx_buf1[32];
            tx_buf[13] = ' ';
            tx_buf[14] = rx_buf1[34];
    tx_buf[15] = 0xd;
    tx_buf[16] = 0xa;
    tx_buf[17] = 0x1A;

    SBUF = tx_buf[0];
    byte_count     = 0;
    octet_to_send  = 17;
    while (byte_count != octet_to_send) {
    };
    delay1();
           for(p=0;p<40;p++)
            {
             rx_buf1[p]=0;
            }
}

void Send_Location_Sms1()
{
    delay1();
    tx_buf[0] = 'A';          //          lattitude
    tx_buf[1] = 'C';    //
    tx_buf[2] = 'C';    //
    tx_buf[3] = 'I';                      //
    tx_buf[4] = 'D';
    tx_buf[5] = 'E';
    tx_buf[6] = 'N';
    tx_buf[7] = 'T';
    tx_buf[8] = ' ';
            tx_buf[9] = 'A';
    tx_buf[10] = 'T';
    tx_buf[11] = ' ';
```

```
    tx_buf[12] = 'L';
            tx_buf[13] = 'O';
            tx_buf[14] = 'C';
    tx_buf[15] = 'A';
    tx_buf[16] = 'T';
            tx_buf[17] = 'I';
    tx_buf[18] = 'O';
            tx_buf[19] = 'N';
    tx_buf[20] = ' ';
            tx_buf[21] = 0xd;
    tx_buf[22] = 0xa;
            // tx_buf[22] = 0x1A;

    SBUF = tx_buf[0];
    byte_count    = 0;
    octet_to_send  = 22;
    while (byte_count != octet_to_send) {
    };
}

void GpsBaud()              //// Function to set 4800 baudrate
 {
  TH1  = 0xFA;      // 0xFA for 4800 baud rate  at 11.0592Mhz
  PCON = 0x80;   ///For doubling baud rate
  TMOD = 0x20;
  SCON = 0x50;
  EA = 1;             /// Enable acknowledge
  ES = 1;  /// Enable Serial Intr
  EX0 = 1;            //// Enable External intr
  TR1= 1;             /// Timer1 ON
 }
 void GsmBaud()             //// Function to set 9600 baudrate
 {
  PCON = 0x80;
 }

void main()
{

            buzzer=0;
            bump=1;

   GpsBaud();
   gpsselect = 0 ;
   gsmselect = 0 ;
   GGA=0;
   GsmBaud();        /////  Set 9600 baudrate
   gsmselect = 1;
   send_location_flg = 0;
   all_data_recd1 = 0;
   recd_byte_cnt1 = 0;
   all_data_recd = 0;
   recd_byte_cnt = 0;
   delete_5_sms_flag =1;

  while(1) //// Repeat Forever
 {

   if(bump==1 && acci==0)
```

```
  {
          send_location_flg=1;

          GpsBaud();
          gsmselect=0;
   gpsselect=1;
          REN = 0;
          buzzer=1;
          delay2();
            GsmBaud();
          delay2();
          Send_Location_Sms1();
           acci=1;
           delay1();
           buzzer=0;
           REN = 1;
  }
  else
  { acci=0;
          buzzer=0;
  }

  loop_count++;  //// Variable to delete sms after 5 loops

          {if(send_location_flg ==0 && delete_5_sms_flag ==0 )
/// Communicate to gsm
       {

                    gpsselect=0;
                    gsmselect=1;
                    GsmBaud();
                    delay1();
       sflg1 = 0;
       set_echo_off();
       delay2();
       sflg1 = 0;
       set_text_mode();
       delay2();
       sflg1 = 0;
          if(twist==0)
          {
                   all_data_recd = 1;
                  send_location_flg = 1;/// Now send location

          }

              else
{
          for( sms_no=0x31; sms_no<0x36; sms_no++ )
                {
                  read_sms();
                  delay2();
                }
}
          delay2();
 }

 if(send_location_flg == 1 && delete_5_sms_flag ==0 ) //gps active and all msges r deleted
          {
```

```
                    GpsBaud();
                    gsmselect=0;
                    gpsselect=1;

          while(all_data_recd1==0 );          // Wait till all_data_recd1 becomes 1..
          {
                    all_data_recd1=0;
                    REN = 0;          // reception is disable
                    delay1();
           delay2();delay2();
                     GsmBaud();
                     delay2();delay2();
                      buzzer=1;
                     Send_Location_Sms();
                   delay1();
                   buzzer=0;


                     send_location_flg = 0;
                     REN = 1;
         // reception is enable

                   //  P1=0;
                    }
                   }
                   if(delete_5_sms_flag ==1)
                   {
                    for( j=0x31; j<0x36; j++ )          /// to delete 1st five sms
       {
        delete_sms();
                          delay2();delay2();delay2();delay2();
       }
                    delete_5_sms_flag =0;
                   }
                    if(loop_count==5)
                    {
                     for( j=0x31; j<0x36; j++ )
       {
        delete_sms();
                          delay2();delay2();delay2();delay2();
       }
                         loop_count=0;
                    }

 }         //// End of while(1)
 }
} ////////// End of main()
void serial() interrupt 4                //4 indiccates Serial ISR
{
          if(RI)

          {
            if(send_location_flg == 0)          //// when receiving from GSM
            {
                    RI         = 0;
                    data1 = SBUF;
   if( all_data_recd )  return;
```

```
 if( data1 == '*')           // receive *#1ON
  {
    star_flg = 1;
 }
 else if(data1 == '#')
  {
   if(star_flg)
    {
     starhash_flg = 1;
     recd_byte_cnt = 0;
     star_flg = 0;
    }
 }
  else
   {
     star_flg = 0;
    if( starhash_flg )
    {
       rx_buf[recd_byte_cnt] = data1;   // store only 1ON or 1OFF

       recd_byte_cnt++;

       if( data1 == 0xd)
       {
          all_data_recd = 1;
           starhash_flg  = 0;
        }
      }
    }
}
if(send_location_flg==1)                        //// When receiving from GPS
{
RI       = 0;

            data1 = SBUF;
  if( data1 == '$'  )
            {
             recd_byte_cnt1 = 0;

            }
  rx_buf1[recd_byte_cnt1] = data1;
  recd_byte_cnt1++;
           if(recd_byte_cnt1 >= 35 && GGA == 1)     /// 35
          {
           all_data_recd1 = 1;
           recd_byte_cnt1 = 0;
           GGA = 0;               // part of GPS from where we get logi&lati string.

          }
  if( data1 == ','  )                                    //// 0x3
                    {
                     if(rx_buf1[4] =='G')
                     {
                      recd_byte_cnt1 = 0;
                      GGA = 1;                              //// Indicates tht GGA string is received
                     }

             }
}
```

```
  } // End of if(RI)
          else
          {
                  TI = 0;
      byte_count++;
      if(byte_count > octet_to_send)
       return;
      SBUF = tx_buf[byte_count];
          }
}

void HWInt0() interrupt 0
{delete_5_sms_flag =1;
 delay2();}
```

VIII. Application

- VIP vehicle tracking.
- Child and animal tracking.
- Accident Notification of Vehicle.
- Ambulance tracking.
- Vehicle Theft Control

IX. Conclusion

The device has been successfully designed and implemented for the "Vehicle Security System With Theft Control And Accident Notification". It has been developed by integrating features of all the hardware components used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced IC's and with the help of growing technology the project has been successfully implemented and tested. Finally we conclude that GPS and GSM based Security System add a huge for the rapid growth of Technology.

Acknowledgment

We would like to thank our guide Mr. Rakesh Jha for his help and support for this research paper.

References

Books Refered

[1]. Wilbert O. Introduction to GPS: the Global Positioning System - Ahmed El-Rabbany
[2]. Fundamentals Of Microprocessors and Microcomputers-B.Ram
[3]. GSM: evolution towards 3rd generation systems - Zoran Zvonar, Peter Jung, Karl Kammerlander
[4]. Car Security and Tracking System with Position, Route, and Speed Calculation, EDP Topic VG06, 2011
[5]. Guochang Xu, "GPS Theory, Algorithms and Applications" Springer, 2007.
[6]. Muhammad Ali Mazidi, Janice Gillspie, Mckinlay, Rolin D., " The Microcontroller in Embedded System: using Assembly and C," *2nd edition published by Pearson Education.*

Website Refered

[1]. www.national.com
[2]. www.atmel.com
[3]. www.wikipedia.com