ROAD ODDITY ALERTING DEVICE

(R.O.A.D)

ABSTRACT

The objective of this project is to make a Road Oddity Alerting Device that augments road safety for vehicle drivers. The device installed inside the driver's cabin will deliver electronic notifications of approaching road oddities visually to the driver on an LCD screen by making use of Radio Frequency Receiver and Transmitter module. The system is capable of providing real-time road-based information to the vehicle driver: providing directions, cautions on speed limits, school zones and can eventually replace cluttered and unreliable road signs creating a new standard in on-road safety for vehicles.

INTRODUCTION

One of the main aspects in designing a car is visibility. Visibility is the measure of distance at which an object or light can be clearly discerned. There are 2 aspects to improving visibility of a driver: 1. Design the vehicle such that the visibility range is maximum to avoid obstacles or 2. Addition of sign boards along the path to warn the driver about the obstacles ahead.

Due to some unavoidable circumstances, such as, absence, rusting, blocking of signboards, also absence or damaged street lights, the visibility of the driver is hampered. This causes an error in the judgement of the driver, which is an important aspect to road safety. This has led to fatal consequences causing road safety a major concern these days.

The hassles of vehicular commuting in crowded Indian metropolitans form a much discussed issue. Waiting hours in traffic, taking tortuous detours due to on-road constructions, navigating blind turns, one-ways and so on – all a part of our daily routine. Forked roads, railway crossings, steep ascents and descents are few samples from the vast gamut of road oddities we encounter on the average drive.

Road Oddity Alerting Device helps in overcoming these drawbacks by digitally displaying appropriate messages on an LCD screen, fitted inside the driver's cabin. This in turn warns the driver about what lies ahead and hence, improving the overall safety.

MOTIVATION

Year	Number of Accidents		Number of Persons		Accident Severity*
	Total	Fatal	Killed	Injured	
2002	4,07,497	73,650 (18.1)	84,674	408,711	20.8
2003	4,06,726	73,589 (18.1)	85,998	435,122	21.1
2004	4,29,910	79,357 (18.5)	92,618	464,521	21.5
2005	4,39,255	83,491 (19.0)	94,968	465,282	21.6
2006	4,60,920	93,917 (20.4)	105,749	496,481	22.9
2007	4,79,216	1,01,161 (21.1)	114,444	513,340	23.9
2008	4,84,704	1,06,591 (22.0)	119,860	523,193	24.7
2009	4,86,384	1,10,993 (22.8)	125,660	515,458	25.8
2010	4,99,628	1,19,558 (23.9)	134,513	527,512	26.9
2011 (P)	4,97,686	1,21,618 (24.4)	1,42,485	5,11,394	28.6

(P): Provisional.
Source: Information supplied by States/UTs (Police Departments).
Figures within parentheses indicate share of fatal accidents to total accidents.
* Accident Severity : No. of Persons Killed per 100 Accidents

- Around 1.4 million road accidents occur in India
- Only .5 million accidents are reported.
- Every hour 16 people lose their life and 60 are injured

It is clearly evident from the above table that there has been an increase in the number of accidents over the years. More than 24% of the accidents that occur result in fatalities. This rise can be attributed to increased congestion on roads and powerful vehicles which can easily cross 100KMPH. With these two factors driving has become dangerous rather than pleasurable. Most of the accidents occur due to loss of control of driver over the vehicle especially on curves and crossings. This lack of control over the vehicle is mainly due to the driver's negligence to the sign boards mounted on the side of the road which are meant to warn him of approaching oddity. These boards guide the driver and alert him regarding the impending oddity. However most of the drivers tend to neglect this traffic signs. And in some cases the board itself may be damaged or obstructed by trees.

Project ROAD aims at delivering the message directly inside the passenger cabin so that 100 percent visibility is achieved. The message is transmitted electronically via RF waves and is displayed on a LCD screen mounted in front of the driver thereby ensuring 100 percent visibility.

PROBLEM

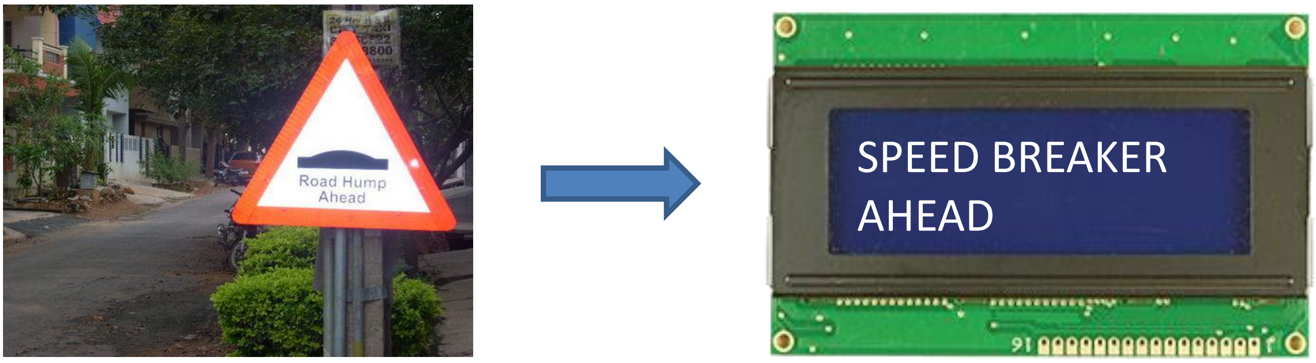
Most drivers don't pay attention to the traffic sign board mounted along the road. In some cases the sign board may be damaged or obstructed. The visibility reduces considerably during the night. The reasons for accident are listed as below.

- **Fault of driver 78.5%**
- Fault of pedestrian 2.2%
- Fault of cyclist 1.2%
- Defect in road conditions 1.3%
- Defect in condition of motor vehicle 1.8%
- Weather condition 0.8%
- All other causes 14.2%

SOLUTION

- We intend to build a device that notifies the driver well in advance of the impending obstacle.
- While driving, a small screen will display the notification for eg. "SPEED BREAKER 50M AHEAD", thus allowing user to slow down his car and avoid bumping into it.
- And also reduce body dive and body squatting effects due to excessive braking.
- From speed breakers to intersections to curves etc. can be known well in time to control your car according to the need.

- This means, one can manoeuvre the acceleration of the vehicle and hence, increase its efficiency by avoiding unnecessary acceleration and braking.



COMPONENTS

Below is the exhaustive list of all the components used for implementing each block:

COMPONENTS	QUANTITY	DESCRIPTION
433MHz TX module	1	To transmit data via RF waves
433MHz RX module	1	To receive data via RF waves
PIC16F886 microcontroller	2	Control unit
LCD display(20x4)	1	Display unit
Battery	2	Power supply
Circuit board	2	Components housing and connection
7805 regulator	2	Power regulation
4MHz Crystal	2	Clock source
22pF ceramic capacitor	4	Clock generation
Single strand wire	1	Interconnections
Push buttons	2	Reset switch
1uF electrolytic capacitor	2	Reset
bc548 npn transistor	1	Power management
4.7k resistors	4	General purpose
LED	2	Indicate Status

Microchip PIC16F886

PIC is a family of modified Harvard architecture microcontrollers made by Microchip Technology. The name PIC initially referred to "Peripheral Interface Controller".

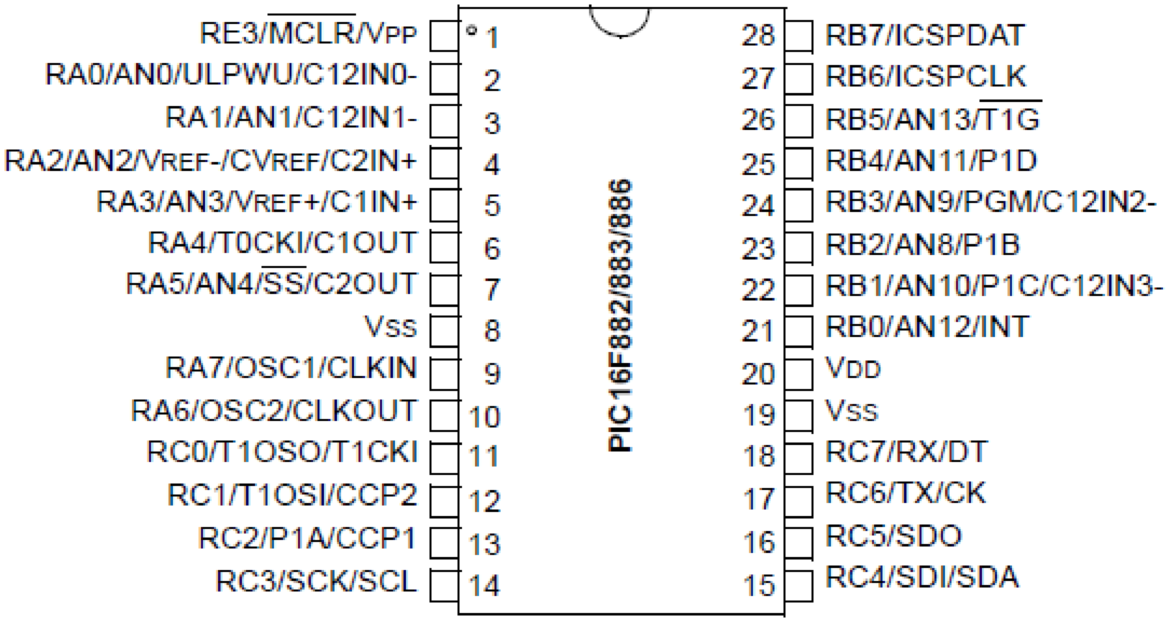
The architectural decisions of PIC16F886 are directed at the maximization of speed-to-cost ratio. It uses Harvard architecture—in which instructions and data come from separate sources—simplifies timing and microcircuit design greatly, and this benefits clock speed, price, and power consumption.

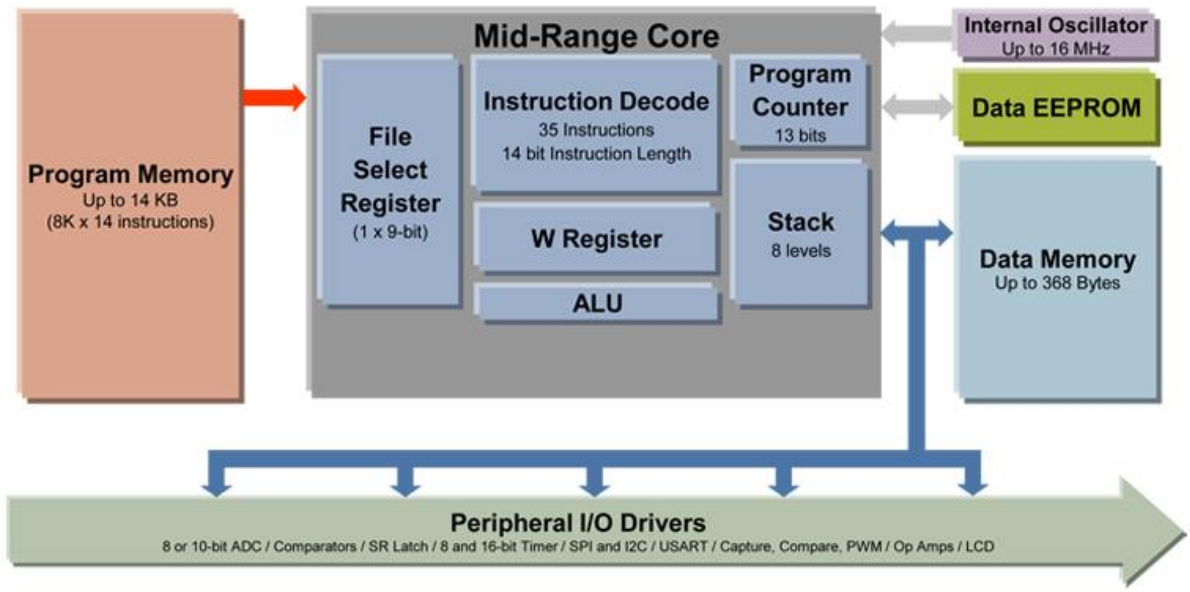
The PIC instruction set is suited to implementation of fast lookup tables in the program space. Such lookups take one instruction and two instruction cycles. Many functions can be modelled in this way. Optimization is facilitated by the relatively large program space of the PIC and by the design of the instruction set, which allows for embedded constants.

Specifications

• Program memory	Flash memory	8192 bytes
• Data memory	SRAM	368 bytes
	EEPROM	256 bytes
• I/O pins	24 pins	
• 10 bit A/D	11 channels	
• EUSART	1 channel	
• TIMERS	8bit	2
	16bits	1

Pin Diagram





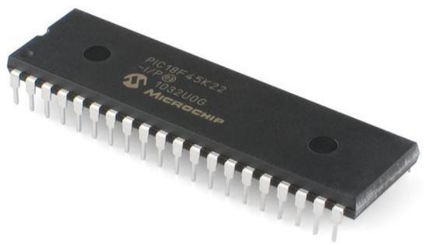
Advantages

The PIC architectures have these advantages:

- Small instruction set to learn.
- RISC architecture.
- Easy entry level, in circuit programming plus in circuit debugging PICKit units available for less than Rs. 2000/-.
- Inexpensive microcontroller.
- Power-Saving Sleep mode

- Wide Operating Voltage Range (2.0V-5.5V).
- High current source/sink for direct LED drive.
- Ultra Low-Power Wake-up (ULPWU).
- Low-Power Features:
  - >Standby Current:
    - 50 nA @ 2.0V, typical
  - > Operating Current:
    - 11uA @ 32 kHz, 2.0V, typical
    - 220uA @ 4 MHz, 2.0V, typical

- Availability of processors in DIL package.



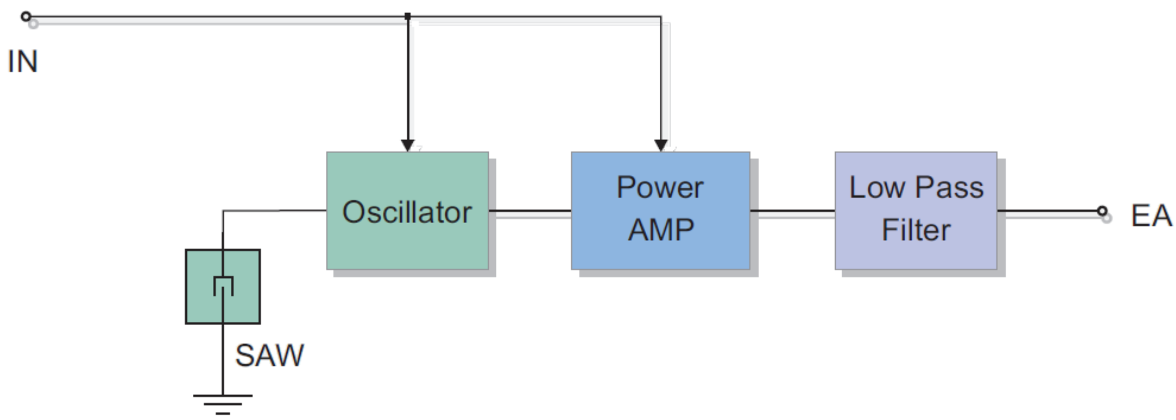
TX MODULE

Radio transmitter is an electronic device which, with the aid of an antenna, produces radio waves. The transmitter itself generates a radio frequency alternating current, which is applied to the antenna. When excited by this alternating current, the antenna radiates radio waves. The purpose of most transmitters is radio communication of information over a distance. The information is provided to the transmitter in the form of an electronic signal through a controller or processor. The transmitter combines the information signal to be carried with the radio frequency signal which generates the radio waves, which is often called the carrier. This process is called modulation. The information can be added to the carrier in several different ways, in different types of transmitter. The antenna may be enclosed inside the case or attached to the outside of the transmitter.

The features of TX module used are:

- Frequency Range: 433.92 MHz
- Modulate Mode: ASK
- Circuit Shape: SAW
- Date Rate: 2400bps
- Supply Voltage: 3~ 12 V
- Power Supply and All Input / Output Pins: -0.3 to +12.0

Block Diagram



Pin description

1 ANT (antenna out)
2 GND (ground)
3 DATA (data in)
4 VCC (+5V)

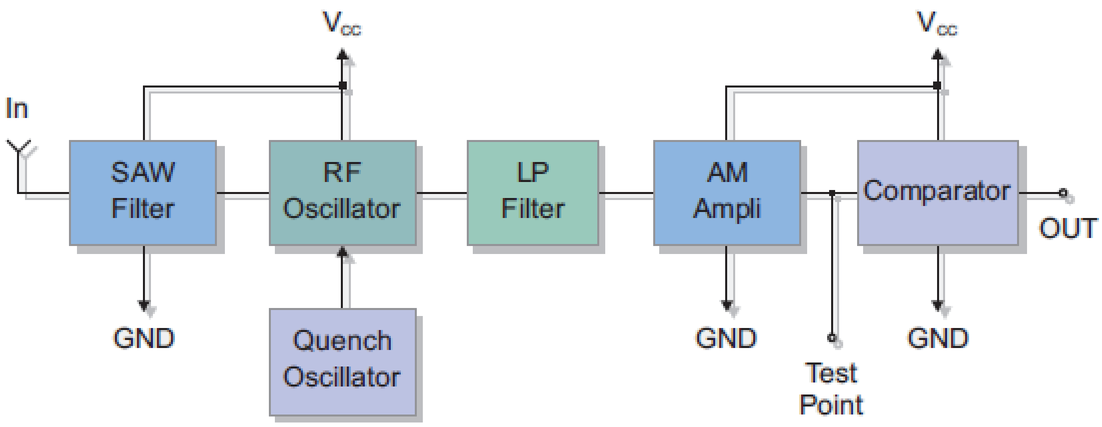
RX Module

The RX module is basically a receiver, which receives data transmitted at 433.92 MHz. A radio receiver is an electronic device that receives radio waves and converts the information carried by them to a usable form. It needs to be used with an antenna. The antenna intercepts radio waves (electromagnetic waves) and converts them to tiny alternating currents which are applied to the receiver, and the receiver extracts the desired information. The receiver uses electronic filters to separate the wanted radio frequency signal from all other signals, an electronic amplifier to increase the power of the signal for further processing, and finally recovers the desired information through demodulation. The information produced by the receiver may be in the form of sound (an audio signal), images (a video signal) or data (a digital signal). In project ROAD, the RX module receives a digital signal and sends it to the controller, which displays an appropriate message on the LCD screen.

The feature of RX Module are:

- Frequency Range: 433.92MHz
- Modulate Mode: ASK
- Circuit Shape: LC
- Date Rate: 2400bps
- Selectivity: -106dBm
- Channel Spacing: ±500KHz
- Supply Voltage: 5V
- High Sensitivity Passive Design.
- Simple To Apply with Low External Count

Block diagram



Pin description

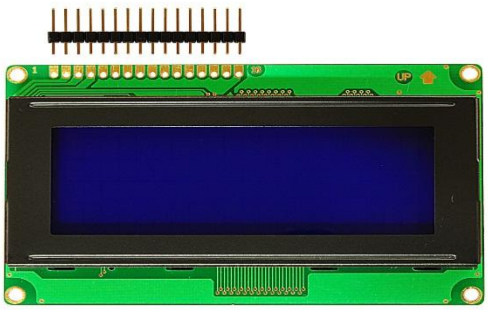
1 ANT (atenna in)
2 GND (digital ground)
3 GND (analog ground)
4 VCC (source)
5 VCC (source)
6 DATA (linear data in)
7 DATA (analog data in)
8 GND (ground)

LCD DISPLAY

This is an industry standard based controlled 4 lines x 20 characters LCD display with WHITE characters on BLUE background and backlight. It is a parallel interface so you will need 7 GPIO pins for 4-bit mode or 11 GPIO pins for 8-bit mode to interface to this LCD screen.

Features:

- Wide viewing angle and high contrast
- +5V DC LED backlight
- Don't need separate power supply for backlight
- Supported 4 or 8 bit parallel interface
- Display 4-line X 20-character
- Operate with 5V DC
- Free 16 positions male header



POWER AND OTHER CIRCUITRY

BATTERY

A battery is a device consisting of one or more electrochemical cells that convert stored chemical energy into electrical energy. This is a rectangular prism shape with rounded edges and a polarized snap connector at the top. Most nine-volt alkaline batteries are constructed of six individual 1.5V LR61 cells enclosed in a wrapper. The battery has both terminals in a snap connector on one end. The smaller circular (male) terminal is positive, and the larger hexagonal or octagonal (female) terminal is the negative contact. The connectors on the battery are the same as on the connector itself; the smaller one connects to the larger one and vice versa.

The power circuit includes a 9V battery to power up the receiver and the transmitter circuits. The controller too, requires a power supply.

OSCILLATORS:

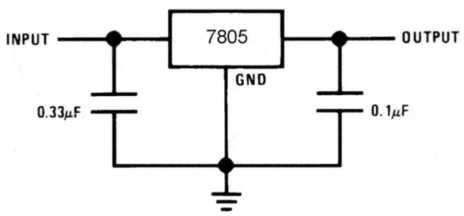
A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digital integrated circuits, and to stabilize frequencies for radio transmitters and receivers. Quartz crystals are manufactured for frequencies from a few tens of kilohertz to tens of megahertz. The one used here is 4 MHz.

REGULATOR IC 7805

A voltage regulator is designed to automatically maintain a constant voltage level. They stabilize the DC voltages used by the processor. It is a Feedback voltage regulator, which operate by comparing the actual output voltage to some fixed reference voltage. Any difference is amplified and used to control the regulation element in such a way as to reduce the voltage error. This forms a negative feedback control loop; increasing the open-loop gain tends to increase regulation accuracy but reduce stability (avoidance of oscillation, or ringing during step changes). There will also be a trade-off between stability and the speed of the response to changes. If the output voltage is too low, the regulation element is commanded, to produce a higher output voltage–by dropping less of the input voltage (for linear series regulators and buck switching regulators), or to draw input current for longer periods (boost-type switching regulators); if the output voltage is too high, the regulation element will normally be commanded to produce a lower voltage.

ADVANTAGES

- 7805 ICs do not require additional components to provide a constant, regulated source of power, making them easy to use, as well as economical and efficient uses of space.
- 7805 ICs have built-in protection against a circuit drawing too much power. They have protection against overheating and short-circuits, making them quite robust in most applications. In some cases, the current-limiting features of the 7805 devices can provide protection not only for the 7805 itself, but also for other parts of the circuit.



STAGES OF IMPLEMENTATION

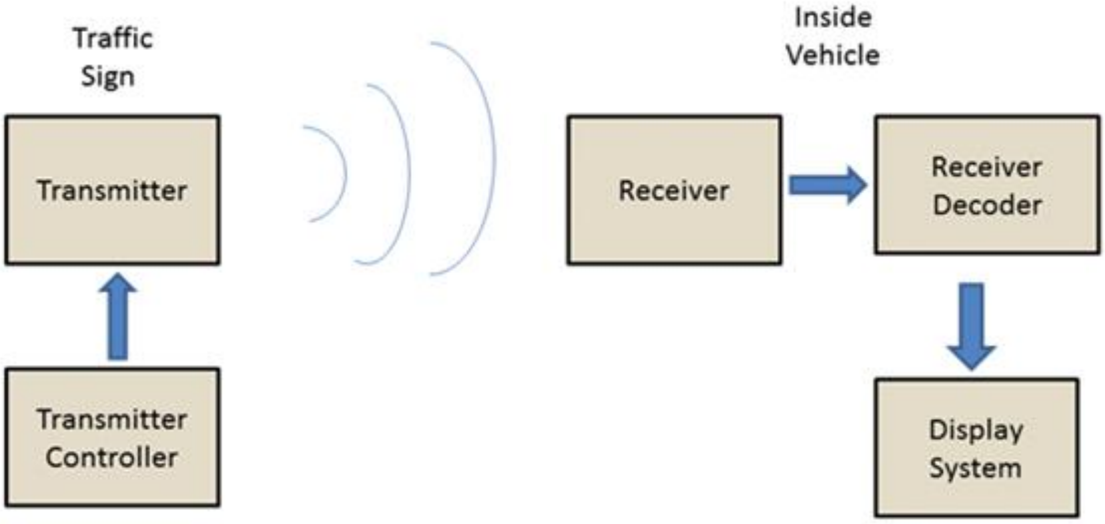
PROJECT OUTLINE

STAGES OF COMPLETION FOR R.O.A.D PROJECT

Sl.	STAGE	DATE OF COMPLETION	DESCRIPTION
1.	Identification Of Team Members	7.2.2013	After carefully analysing the character of various classmates, a balanced team was formed to carry out the project.
2.	Ideation	14.2.2013	The basic idea and the motivation behind same was discussed by the team members. Various statistical reports were read and the problem to be tackled was understood.
3.	Ideation stage 2	21.2.2013	After understanding the problem, a low cost and practical solution needed to be implemented. Hence, a general idea of the solution was discussed. A basic block diagram was made to depict the solution.
4.	Feasibility Check	28.2.2013	The feasibility of the project was checked for.
5.	Identification of Components and Cost analysis	14.3.2013	Post the block diagram design, analysis was done as to how each block needs to be implemented. The components required for same were identified and a cost report was made for the same.
6.	Design Stage	21.3.2013	The numerical values of the various components required was carefully designed to ensure the proper working of the system. Also, the algorithms required were designed.
7.	Procurement Of components	28.3.2013	The components identified from the analysis conducted were procured from their place of availability.
8.	Testing Of components	4.4.2013	The components procured were then tested for their working condition under various environments.
9.	Testing of components: Stage 2	11.4.2013	The components were then wired to each other as a circuit and their working was checked for. This stage also required the code to be ready, which was done.
10.	Aesthetics	18.4.2013	The system we aimed to design was working. Next came the part of making it look presentable and appealing.
11.	Innovation	25.4.2013	Another discussion took place as to how the same product/project can be used for various other circumstances.
12.	Implementation and Presentation	08.05.2013	The project was implemented where required and presented to the respective experts.

DESIGN DEVELOPMENT

The functional block diagram of the basic design is as shown below.



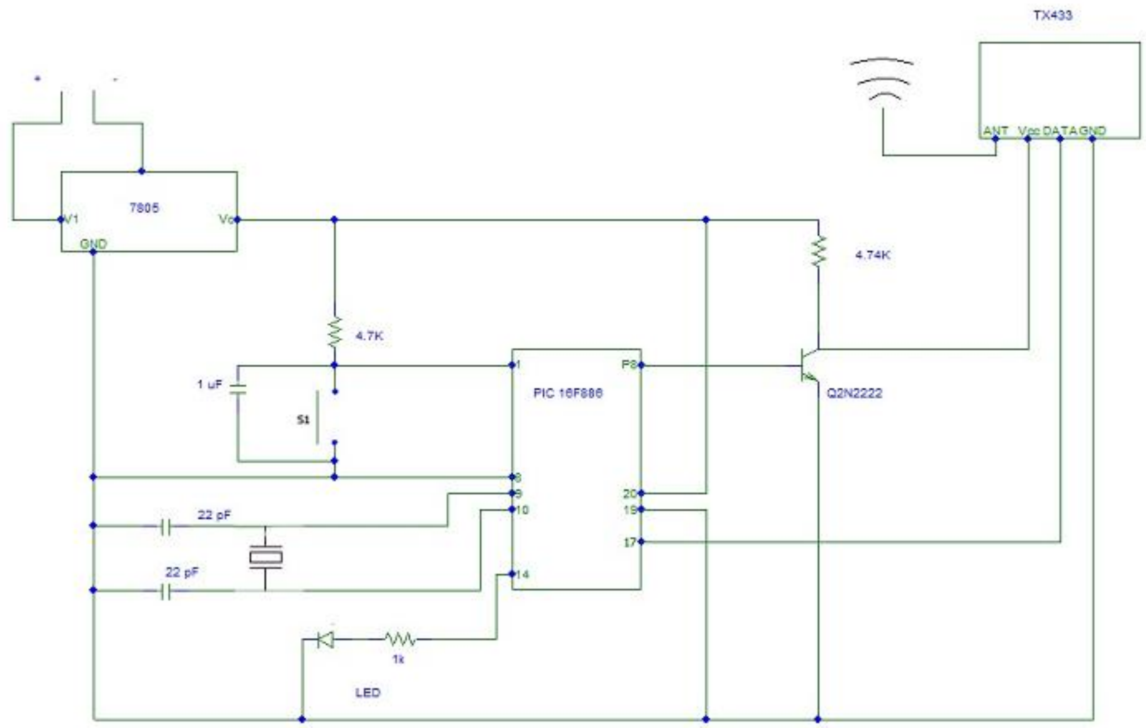
The devices present along the road will be transmitting the message with help of transmitter which is interfaced with the microcontroller. A receiver fitted in the vehicle will pick up the signal once it is in the vicinity of transmitter. . This signal is retrieved and used to detect, identify, and display information.

PROCURMENT OF COMPONENTS

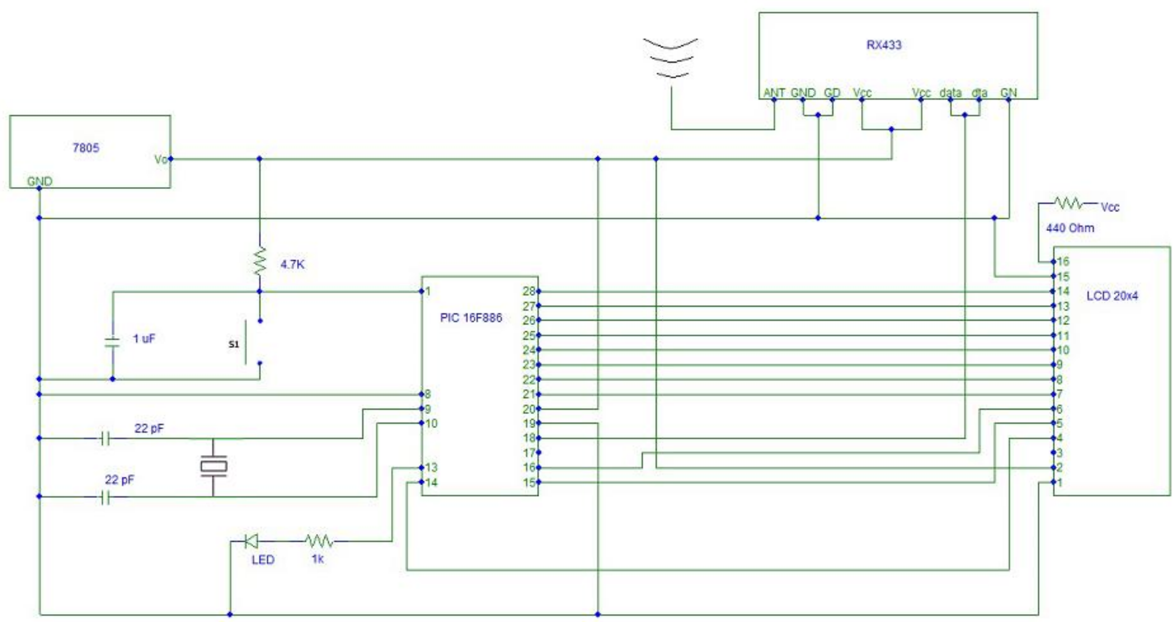
The components were procured on the date mentioned in the schedule for the project. Various locales were scanned for the required parts. Finally, S.P Road was chosen as the destination to procure the components.

CIRCUIT DIAGRAM

TX UNIT CIRCUIT



RX UNIT CIRCUIT



MICROCONTROLLER PROGRAMMING

The two PIC16F886 microcontroller were programmed to transmit and receive data respectively. The microcontroller code was written in C/C++ language and converted to HEX file using software called MPLAB X IDE v 1.30 by Microchip Technology and XC8 compiler.

MPLAB® X IDE is a software program that runs on a PC (Windows®, Mac OS®, Linux®) to develop applications for Microchip microcontrollers and digital signal controllers. It is called an Integrated Development Environment (IDE), because it provides a single integrated "environment" to develop code for embedded microcontrollers.

MPLAB XC8 compiler:

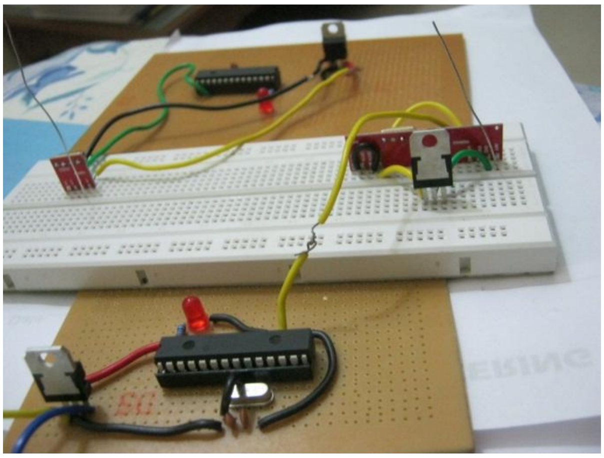
- Integrates with MPLAB® X IDE to provide a full graphical front end:
- Editing errors and breakpoints match the corresponding lines in source code
- Single step through C and C++ source code to inspect variables and structures at critical points
- Data structures with defined data types, including floating point, display in watch windows
- Runs on Windows, Linux and Mac OS X

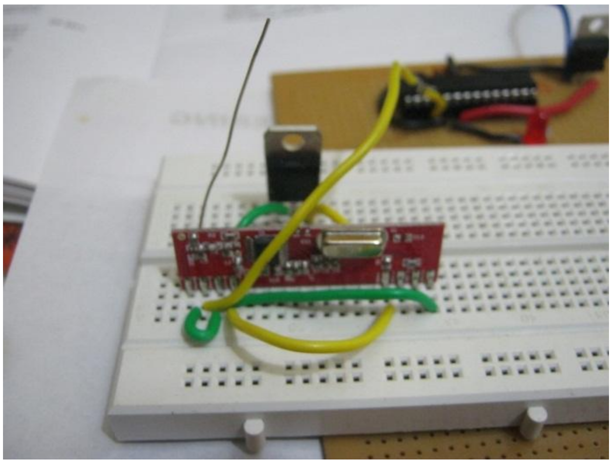
The code was burned onto the microcontroller using PIC programmer using burning tool called **PICkit 2 v2.61** by Microchip technology. The PICkit™ 2 Development Programmer/Debugger (PG164120) is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers.

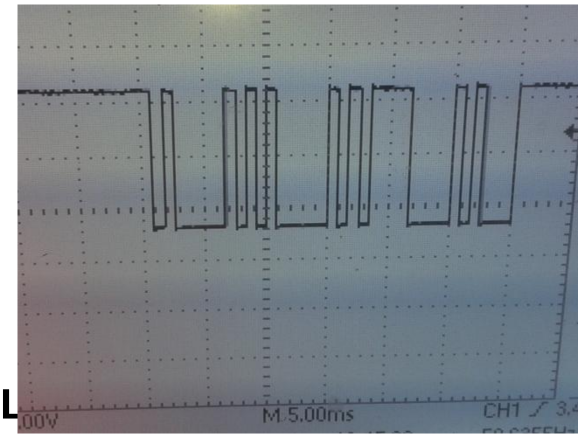
The microcontroller was programmed to transmit and receive data at baud rate of 1200 following Manchester Scheme of serial data transmission.

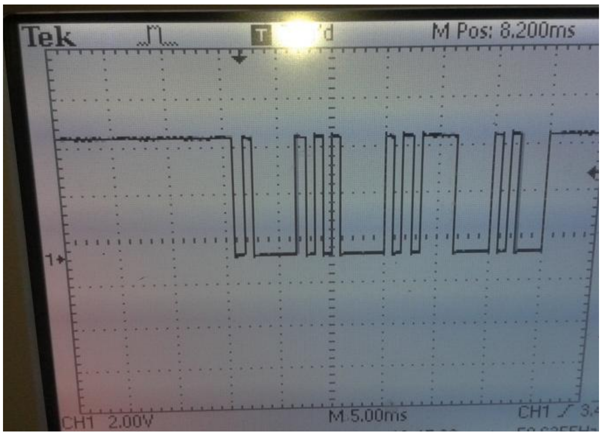
RX AND TX 433Mhz module interfacing:

After successfully burning the code onto the microcontrollers the RX and TX modules were connected to the microcontrollers and the outputs across the RX and TX pin were observed on an oscilloscope. The outputs of both the modules were matching and hence it was inferred that the signal transmission was taking place successfully.









L

The 20x4 white on blue background character LCD display was interfaced to the circuit. Sample messages were tested on the LCD screen by placing the receiver and transmitter at various locations.

PROGRAM CODES AND FLOWCHART

TRANSMITTER PROGRAM

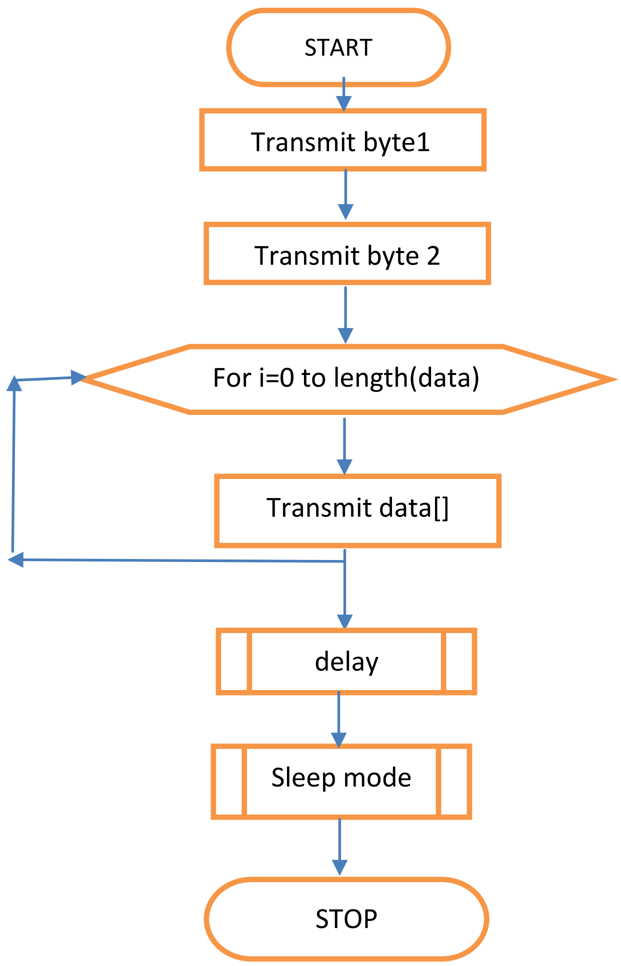
```
//******************CODE FOR TRANSMISSION OF DATA BEGINS******************
/*

 This program transmits the message via TX pin i.e PORTC.6

 */

#include <stdio.h>

#include <stdlib.h>

#include<pic16f886.h>

#include<htc.h>
```

```
#define synbyte1 'A'

#define synbyte2 'A'

#define length    35

#define mssg      "SPEED BREAKER AHEAD       GO SLOW!!"

#define TRUE     1


//Configuration BITS

__CONFIG(FOSC_HS & WDTE_ON & PWRTE_OFF & MCLRE_OFF & CP_OFF & CPD_OFF &
BOREN_ON & IESO_ON & FCMEN_ON & LVP_OFF);

__CONFIG(BOR4V_BOR40V & WRT_OFF);


//UART initialization subroutine

void inituart(void)

{

        TRISC6 = 0;                 //TX Pin outpin

        SPBRG =103;                 //BAUD rate of 2400bps

        BRGH  = 0;                  //normal baudrate

         SPEN  = 1;                 //enable serial port pins

        TXEN  = 1;                  //Enable Trasmission

        SYNC  = 0;                  //asynchronous

        TX9   = 0;                  //8 bits transmission

}


//Transmit Subroutine

void    transmit(unsigned char text)

{
```

```
    while(!TXIF);                        //wait for previous transmission to finish
    TXREG = text;                        //Transmit Character
}

//delay subroutine
void    delay(unsigned int n)
{
    unsigned int j;
    for(j=0;j<n;j++);
}

//main program
int main()
{
    unsigned char i,sum,chksum;
    unsigned char x[length];
    x[length]=mssg;                      //message to be transmitted
    TRISB=0X00;                          //PortB as output port
    inituart();                          //initialise UART
    sum=0;

    for(i=0;i<length-1;i++)              //calculation of checksum
    {
        sum=sum+x[i];
    }
    chksum=0xFF-sum+1;                   //checksum generated
```

```
    x[length-1]=chksum;                    //append checksum to mssg


    while(TRUE)
    {
        PORTB=0X80;                         //status indicator
        delay(60000);
        transmit(synbyte1);                //trasmit synbyte 1
        transmit(synbyte2);                //transmit synbyte 2
        transmit(length);                  //transmit message length
        for(i=0;i<length;i++)
        {
            transmit(x[i]);                //transmit message
        }
        PORTB=0X00;
        delay(60000);
    }
    return (EXIT_SUCCESS);
}


//*******************CODE FOR TRANSMISSION OF DATA ENDS******************
```

Flowchart for transmitter program:



RECEIVER PROGRAM

```
//******************CODE FOR RECEIVER OF DATA BEGINS******************
/*

 This program receives message serially and displays it on LCD

 */
```

```
#include <stdio.h>

#include <stdlib.h>

#include <pic16f886.h>

#include <htc.h>

#define         synbyte1        'A'

#define         synbyte2        'A'

#define         DATA            PORTB                   //Data PORT

#define         RS                          PORTCbits.RC1       // RS signal for LCD

#define         RW              PORTCbits.RC2       // R/W signal for LCD

#define         E               PORTCbits.RC3       // E signal for LCD

#define         TRUE             1


unsigned char   x[80];                                  //global variable

unsigned char   length=0;                               //global variable


/*Configuration bits*/

__CONFIG(FOSC_HS & WDTE_ON & PWRTE_OFF & MCLRE_OFF & CP_OFF & CPD_OFF &
BOREN_ON & IESO_ON & FCMEN_ON & LVP_OFF);

__CONFIG(BOR4V_BOR40V & WRT_OFF);


/*subroutines declaration*/

void inituart(void);                            //initialise UART

unsigned char receive(void);                            //receive character

void Init_LCD(void);                            //Initialize LCD dispaly

void cmd(unsigned char);                        // 8-bit control word for LCD
```

```
void data(unsigned char);                    // 8-bit text data for LCD

void display(void);                          //display the message

void clear(void);                            //clear LCD screen

void delay(unsigned int);                    // delay


/*main program*/

int main()

{

   unsigned char syn_byte1=0,syn_byte2=0;

   unsigned char i,sum;

   TRISB=0x00;                               //PORTB as output

   inituart();                               //initialise UART


   while(TRUE)

   {

      clear();

      syn_byte1=receive();                   //get the synbyte1

      if(syn_byte1!=synbyte1)

      {

         continue;                           //goto while if mismatch

      }

      syn_byte2=receive();                   //get the synbyte2

      if(syn_byte2!=synbyte2)

      {

         continue;                           //goto while if mismatch

      }
```

```
        length=receive();                           //receive LENGTH number of byte
        sum=0;
        for(i=0;i<length;i++)                       //calculate sum
        {
          x[i]=receive();
          sum=sum+x[i];
        }
        if(sum==0)                                  //if data not corrupt display message
        {
          display();                                //display message
          for(i=0;i<10;i++)
          {
            delay(60000);
          }
          clear();
        }
        else                                        //if data corrupt clear display
        {
          Init_LCD();
          delay(2000);
        }
      }
      return (EXIT_SUCCESS);
    }
```

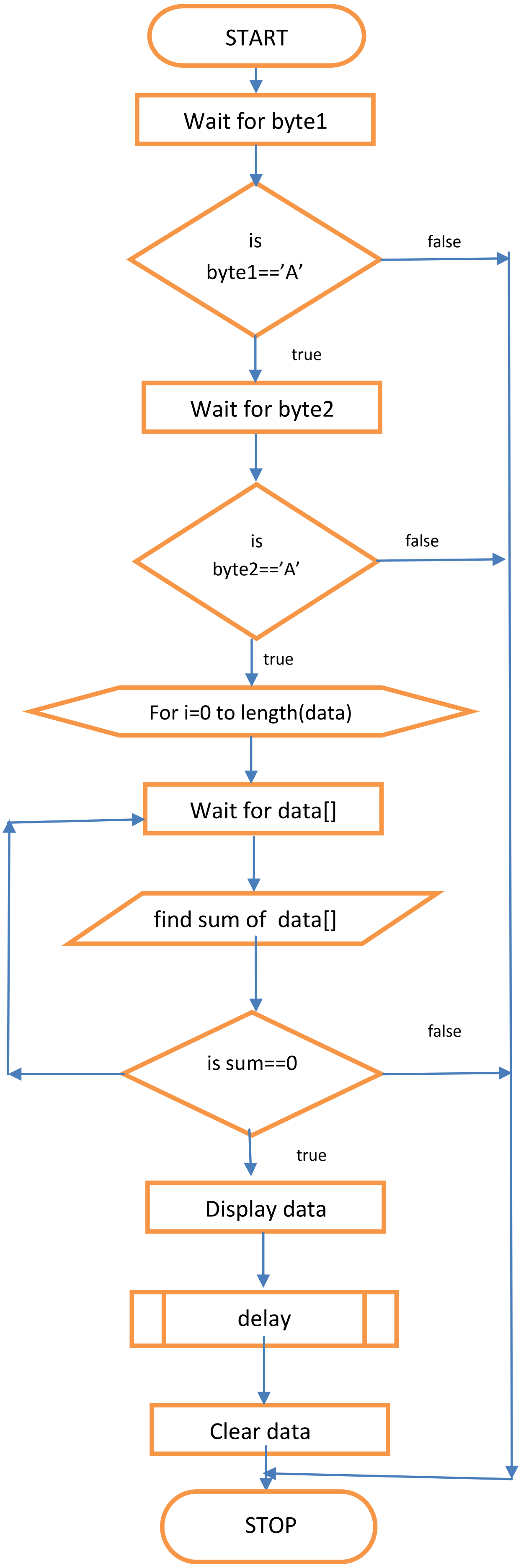
```
/*initialise UART*/

void inituart(void)

{

        TRISC7 = 1;                         //RX Pin

        SPBRG  = 103;                       //BAUD rate of 2400bps

        BRGH   = 0;                         //normal baudrate

        SYNC   = 0;                         //asynchronous

        SPEN   = 1;                         //enable serial port pins

        CREN   = 1;                         //enable reception

        SREN   = 0;                         //no effect

        RCIE   = 1;                         //enable rx interrupts

        RX9    = 0;                         //8-bit reception


}


/*Gets a character from the serial port*/

unsigned char receive(void)

{

        if(OERR)                            //if over run error, then reset the receiver

        {

                CREN = 0;

                CREN = 1;

        }


        while(!RCIF);                       //wait for transmission to receive
```

```
        return RCREG;                    //return the recieved byte
}


/*delay subroutine*/
void delay(unsigned int n)
{
   unsigned int j;
   for(j=0;j<n;j++);
}


/*display message*/
void display()
{


   unsigned char i;
   TRISB =  0x00;                          //PORTB as output
   TRISC = 0x80;                          // set PORTC bit(0:6) as output connected LCD
signal
   Init_LCD();                            // Init LCD 8bit interface multiple lines
   for(i=0; i<(length-1); i++)            // output message
   {
      data(x[i]);                         // write indiviual characters to LCD screen
   }


}
```

```
/* LCD display initialization */
void Init_LCD()
{
cmd(0x3C);                              // Function Set - 8-bit, 2 lines, 5X7
cmd(0b00001100);                        // Display on, cursor on
cmd(0b00000110);                        // Entry mode - inc addr, no shift
cmd(0b00000001);                        // Clear display & home position
cmd(0b00000010);
}
/* Write control word to LCD */
void cmd(unsigned char d)
{
   RW=0;                                //write mode
   RS = 0;                              // Data as command
   DATA = d;                            //data on PORTB
   E = 1;                               //Enable HIGH
   delay(50);
   E = 0;                               // Enable LOW
   delay(1000);
}

/* Write text data to LCD */
void data(unsigned char d)
{

   RW=0;                                //write mode
```

```
    RS = 1;                    //data as character
    DATA = d;                  //data on PORTB
    E = 1;                     //Enable HIGH
    delay(50);
    E = 0;                     // Enable LOW
    delay(1000);
}
/*clear the display*/
void clear()
{
    cmd(0x01);
    delay(10000);
    cmd(0x08);
    delay(10000);
}
```

Flowchart for receiver program:



RESULTS

The microcontroller testing board worked fine. The burning of program onto the PIC 16f886 microcontroller was successful.

The receiver and transmitter modules were working as expected and the floating pins were negated. The turn ON and OFF of red LED'S inferred to that the receiving and transmission of data was taking place effectively as per the program written.

The device was successfully tested for various messages displayed on the LCD by placing the transmitter at different locations and by moving the receiver in and out of the transmitter's vicinity. When the receiver is brought within the transmitter's range, the message is displayed on the receiver side LCD display.

The device was tested for various range of operation by altering the length of the antennas.

A list prepared after approaching our classmates and teachers regarding the road oddities encountered by them. The device was demonstrated in a simulated environment in an open space by placing card board and thermocol made road signs and oddities.

Demonstration was also carried out inside a car driven across a certain region and the video of the same was recorded.

FUTURE PROSPECTS

The horizon of such a project is vast. The following implementations can be further thought of, and integrated in this project.

1. Advertisements of restaurants, hotels, theme parks etc. in the vicinity could be displayed on the LCD screen.

2. Useful information about the road condition can be gathered using sensors and then passed on into the car via the transmitter and displayed on the LCD screen.

3. A trans-receiver can be fitted instead of a receiver in the car. Useful information about the driver and his blood group details can be fed into the controller. In case of any accident, the same can be detected and displayed onto the passing cars. This increases probability of the mishap being detected quicker.

4. If the stop time at a traffic signal is more than 30 seconds. Then, it is advisable to turn off engine to reduce emissions. Hence, this could be automated depending on the choice of the driver. That is, if the length of red light is more than 30 seconds, then it would suggest the driver to switch off the engine.