ANTI CONFINEMENT SECURITY AND AUTOMATIC AMBULANCE RESCUE SYSTEM

A PROJECT REPORT

Submitted by

KAARTHIKEYAN R **Reg. No.:13BEC058**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 049

ANNA UNIVERSITY : CHENNAI

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report titled **"ANTI CONFINEMENT SECURITY AND AUTOMATIC AMBULANCE RESCUE SYSTEM"** is the bonafide work of **MR KAARTHIKEYAN.R [Reg.NO.13BEC058]** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Mr. D.ALLIN JOE, M.E.,
PROJECT SUPERVISOR
Department of ECE
Kumaraguru College of Technology
Coimbatore

SIGNATURE
Dr.K.MALARVIZHI Ph.D.,
HEAD OF THE DEPARTMENT
Department of ECE
Kumaraguru College of Technology
Coimbatore.

The candidates with Register number 13BEC058 are examined by us in the project viva-voce examination held on …………………….

INTERNAL EXAMINER **EXTERNAL EXAMINAR**

ACKNOWLEDGEMENT

First we would like to express our praise and gratitude to the Lord, who has showered his grace and blessing enabling us to complete this project in an excellent manner. He has made all things in beautiful in his time.

We express our sincere thanks to our beloved Joint Correspondent,

Shri. Shankar Vanavarayar for his kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr.R.S.Kumar M.E., Ph.D.,** who encouraged us with his valuable thoughts.

We would like to express our sincere thanks and deep sense of gratitude to our HOD, **Dr. K.Malarvizhi, Ph.D.,** for her valuable suggestions and encouragement which paved way for the successful completion of the project.

We are greatly privileged to express our deep sense of gratitude to the Project Coordinator **Mrs.Thilagam,M.E.**, Assistant Professor (SRG), for her continuous support throughout the course.

In particular, We wish to thank and express our everlasting gratitude to the Supervisor **Mr.D.Allin joe, M.E.**, Assistant Professor for his expert counselling in each and every steps of project work and we wish to convey our deep sense of gratitude to all teaching and non-teaching staff members of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

TABLE OF CONTENTS

ABSTRACT

Traffic congestion and tidal flow management were recognized as major problems in modern urban areas, which have caused much uncomfortable for the ambulance. Moreover road accidents in the city have been nonstop and to bar the loss of life due to the accidents is even more crucial. To implement this we introduce a scheme called AARS (Automatic ambulance rescue system). The main theme behind this scheme is to provide a smooth flow for the ambulance to reach the hospitals in time and thus minifying the expiration. The idea behind this scheme is to implement a ITS which would control mechanically the traffic lights in the path of the ambulance. The ambulance is controlled by the central unit which furnishes the most scant route to the ambulance and also controls the traffic light according to the ambulance location and thus reaching the hospital safely. The server also determines the location of the accident spot through the sensor systems in the vehicle which encountered the accident and thus the server walks through the ambulance to the spot. This scheme is fully automated, thus it finds the accident spot, controls the traffic lights, helping to reach the hospital in time.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. **GPS-GLOBAL POSITIONING SYSTEM**

2. **GSM-GLOBAL SYSTEM FOR MOBILE COMMUNCATION**

3. **RSSI-RECEIVED SIGNAL STRENGTH INDICATOR**

4. **TDMA-TIME DIVISION MULTIPLE ACCESS**

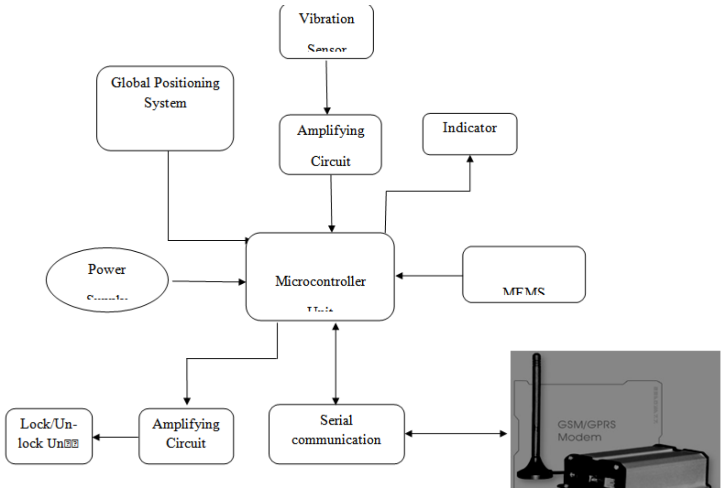
5. **CDMA-CODE DIVISION MULTIPLE ACCESS**

CHAPTER 1

INTRODUCTION

When a vehicle met with an accident the message came to control room or a rescue team by using GPS and GSM Technology. GPS is a fleet of more than 24 communications satellites that transmit signals globally around the clock. With a GPS receiver, one can quickly and accurately determine the latitude, the longitude, and in most cases the altitude of a point on or above Earth's surface. GSM use a Subscriber Identity Module (SIM) smart card that contains user account information. Any GSM phone becomes immediately programmed after plugging in the SIM card, thus allowing GSM phones to be easily rented or borrowed. Here an accelerometer (MEMS) is used in a car alarm application; MEMSENSER is a powerful yet simple software tool for engineers, researchers and students working in the field of Micro Electro Mechanical Systems Dangerous driving can be detected with an accelerometer. It can be used as a crash recorder of the vehicle movements before, during and after a crash. With signals from an accelerometer, a severe accident can be recognized. According to this project when a vehicle met with an accident immediately the vehicle number and persons contact number will be transferred to control room or a rescue team. So the rescue team can immediately trace the location from where the message came. Then after conforming the location necessary action will be taken. In second application on an uncertain situation many of vehicles that have center locking system, Such as door locking system faces many problems due to automatic locking system. At that situation there is no way to open the lock. Our project will provide a suitable solution for this situation.

1.1 BLOCK DIAGRAM:



POWER SUPPLY UNIT

1.2 CIRCUIT DIAGRAM:

Figure1.2 shows the circuit diagram of power supply.

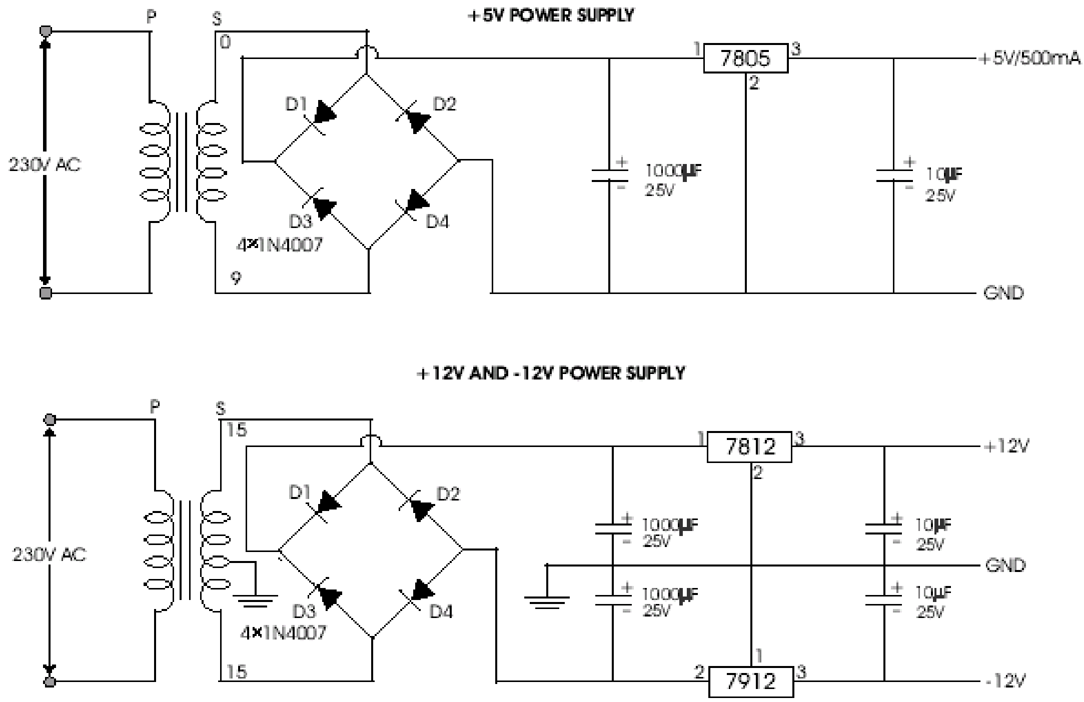


Fig.1.2. Circuit Diagram of Power Supply

1.2.1 WORKING PRINCIPLE:

The AC voltage, typically 220V rms, is connected to a transformer, which steps that ac voltage down to the level of the desired DC output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.

1.3 ARM PROCESSOR

1.3.1 ARM7 FAMILY:

The ARM7 family includes the ARM7TDMI, ARM7TDMI-S, ARM720T, and ARM7EJ-S processors. The ARM7TDMI core is the industry's most widely used 32-bit embedded RISC microprocessor solution. Optimized for cost and power-sensitive applications, the ARM7TDMI solution provides the low power consumption, small size, and high performance needed in portable, embedded applications.The ARM7TDMI-S core is the synthesizable version of the ARM7TDMI core, available in both VERILOG and VHDL, ready for compilation into processes supported by in-house or commercially available synthesis libraries. Optimized for flexibility and featuring an identical feature set to the hard macro cell, it improves time-to-market by reducing development time while allowing for increased design flexibility, and enabling >>98% fault coverage. The ARM720T hard macro cell contains the ARM7TDMI core, 8kb unified cache, and a Memory Management Unit (MMU) that allows the use of protected execution spaces and virtual memory. This macro cell is compatible with leading operating systems including Windows CE, Linux, palm OS, and SYMBIAN OS.

1.3.2 ADVANTAGES

a. Small device
b. Lower Power Consumption
c. Simple decoding
d. Higher performance

CHAPTER 2

LPC 2148 MICROCONTROLLER

2 .1 LPC2148 MICROCONTROLLER:

LPC2148 microcontroller board based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine microcontrollers with embedded high-speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30% with minimal performance penalty. The meaning of LPC is Low Power Low Cost microcontroller. This is 32 bit microcontroller manufactured by Philips semiconductors (NXP).Due to their tiny size and low power consumption, LPC2148 is ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale.

2.2 FEATURES OF LPC2148 MICROCONTROLLER:

- 16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
- 8 kB to 40 kB of on-chip static RAM and 32 kB to 512 kB of on-chip flash memory; 128-bit wide interface/accelerator enables high-speed 60 MHz operation.
- In-System Programming/In-Application Programming (ISP/IAP) via on-

chip boot loader software, single flash sector or full chip erase in 400 ms and programming of 256 B in 1 ms Embedded ICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip Real Monitor software and high-speed tracing of instruction execution.

- USB 2.0 Full-speed compliant device controller with 2 kB of endpoint RAM. In addition, the LPC2148 provides 8 kB of on-chip RAM accessible to USB by DMA.

SERIAL COMMUNICATION

2.3 INTRODUCTION:

Serial communication is basically the transmission or reception of data one bit at a time. Today's computers generally address data in bytes or some multiple thereof. A byte contains 8 bits. A bit is basically either a logical 1 or zero. Every character on this page is actually expressed internally as one byte. The serial port is used to convert each byte to a stream of ones and zeroes as well as to convert a stream of ones and zeroes to bytes. The serial port contains a electronic chip called a Universal Asynchronous Receiver/Transmitter (UART) that actually does the conversion.The serial port has many pins. We will discuss the transmit and receive pin first. Electrically speaking, whenever the serial port sends a logical one (1) a negative voltage is effected on the transmit pin. Whenever the serial port sends a logical zero (0) a positive voltage is affected. When no data is being sent, the serial port's transmit pin's voltage is negative (1) and is said to be in a MARK state. Note that the serial port can also be forced to keep the transmit pin at a positive voltage (0) and is said to be the SPACE or BREAK state. (The terms MARK and SPACE are also used to simply denote a negative voltage (1) or a positive voltage (0) at the transmit pin respectively).When transmitting a byte, the UART (serial port) first sends a START BIT which is a positive voltage (0), followed by the data (general 8 bits, but could be 5, 6, 7, or 8 bits) followed by one or two STOP Bits which is a negative(1) voltage. The sequence is repeated for each byte sent. Figure 1 shows a diagram of what a byte transmission would look like.

when transmitting a character there are other characteristics other than the baud rate that must be known or that must be setup. These characteristics define the entire interpretation of the data stream. The first characteristic is the length of the byte that will be transmitted. This length in general can be anywhere from 5 to 8 bits.The second characteristic is parity. The parity characteristic can be even, odd, mark, space, or none. If even parity, then the last data bit transmitted will be a logical 1 if the data transmitted had an even amount of 0 bits. If odd parity, then the last data bit transmitted will be a logical 1 if the data transmitted had an odd amount of 0 bits. If MARK parity, then the last transmitted data bit will always be a logical 1. If SPACE parity, then the last transmitted data bit will always be a logical 0. If no parity then there is no parity bit transmitted.

The third characteristic is the amount of stop bits. This value in general is 1 or 2. Assume we want to send the letter 'A' over the serial port. The binary representation of the letter 'A' is 01000001. Remembering that bits are transmitted from least significant bit (LSB) to most significant bit (MSB), the bit stream transmitted would be as follows for the line characteristics 8 bits, no parity, 1 stop bit and 9600 baud.LSB (0 1 0 0 0 0 0 1 0 1) MSB The above represents (Start Bit) (Data Bits) (Stop Bit). To calculate the actual byte transfer rate simply divide the baud rate by the number of bits that must be transferred for each byte of data. In the case of the above example, each character requires 10 bits to be transmitted for each character. As such, at 9600 baud, up to 960 bytes can be transferred in one second.

The above discussion was concerned with the "electrical/logical" characteristics of the data stream. We will expand the discussion to line protocol. Serial communication can be half duplex or full duplex. Full duplex communication means that a device can receive and transmit data at the same time. Half duplex means that the device cannot send and receive at the same time. It can do them both, but not at the same time. Half duplex communication is all but outdated except for a very small focused set of applications.

Half duplex serial communication needs at a minimum two wires, signal ground and the data line. Full duplex serial communication needs at a minimum three wires, signal ground, transmit data line, and receive data line. The RS232 specification governs the physical and electrical characteristics of serial communications. This specification defines several additional signals that are asserted (set to logical 1) for information and control beyond the data signal

These signals are the Carrier Detect Signal (CD), asserted by modems to signal a successful connection to another modem, Ring Indicator (RI), asserted by modems to signal the phone ringing, Data Set Ready (DSR), asserted by modems to show their presence, Clear To Send (CTS), asserted by modems if they can receive data, Data Terminal Ready (DTR), asserted by terminals to show their presence, Request To Send (RTS), asserted by terminals if they can receive data. The section RS232 Cabling describes these signals and how they are connected.

The above paragraph alluded to hardware flow control. Hardware flow control is a method that two connected devices use to tell each other electronically when to send or when not to send data. A modem in general drops (logical 0) its CTS line when it can no longer receive characters. It re-asserts it when it can receive again. A terminal does the same thing instead with the RTS signal. Another method of hardware flow control in practice is to perform the same procedure in the previous paragraph except that the DSR and DTR signals.

2.4SOFTWARE TOOLS

2.4.1 SOFTWARE TOOLS

The list of software tools used are

- KIEL IDE
- Flash Magic
- Orcad design
- Embedded-C language

2.5 INTRODUCTION

OrCAD-Circuit Design:

- This tool is used to design the schematic of the hardware.
- Using Orcad the PCB layout is designed

Keil IDE's:

- This tool is used to develop the source code needed for the design.
- The tool helps us not only to develop but also compile the code and simulate the code.
- The keil tool is also used to convert the compiled Embedded C code to its equivalent hex code.

Flash Programmer:

- Flash programmer is used to fuse the built hex code into the Microcontroller AT89c51 (here).

- **Language**: Embedded C.

2.6 ORCAD CAPTURE CIS

OrCAD Capture CIS is designed to reduce production delays and cost overruns through efficient management of components. It reduces the time spent searching existing parts for reuse, manually entering part information content, and maintaining component data. Users search parts based on their electrical characteristics and OrCAD Capture CIS automatically retrieves the associated part. Flexible and scalable, the solution is quickly implemented. OrCad Capture CIS is ideal for individual design teams or multi-site teams who need to collaborate across multiple locations, OrCAD Capture CIS gives designers access to correct part data early in the design process and enables complete component

specifications to be passed to board designers and other members of the design team, reducing the potential for downstream errors. It provides access to cost information so designers can use preferred, lower cost, and instock parts. The embedded part selector accesses information stored in MRP/ERP systems and engineering databases and synchronizes externally sourced data with the schematic design database, so bills of materials can be automatically generated.

2.7 KEIL C COMPILER:

Keil Software publishes one of the most complete development tool suites for 8051 software, which is used throughout industry. For development of C code, their Developer's Kit product includes their C51 compiler, as well as an integrated 8051 simulator for debugging. A demonstration version of this product is available on their website, but it includes several limitations

The C programming language was designed for computers, though, and not embedded systems. It does not support direct access to registers, nor does it allow for the reading and setting of single bits, two very important requirements for 8051 software. In addition, most software developers are accustomed to writing programs that will by executed by an operating system, which provides system calls the program may use to access the hardware. However, much code for the 8051 is written for direct use on the processor, without an operating system. To support this, the Keil compiler has added several extensions to the C language to replace what might have normally been implemented in a system call, such as the connecting of interrupt handlers.

The purpose of this manual is to further explain the limitations of the Keil compiler, the modifications it has made to the C language, and how to account for these in developing software for the 8051 microcontroller.

Keil LIMITATIONS

There are several very important limitations in the evaluation version of Keil's

2.8 ORCAD

ORCAD really consists of tools. Capture is used for design entry in schematic form. You will probably be already familiar with looking at circuits in this form from working with other tools in your university courses. Layout is a tool for designing the physical layout of components and circuits on a PCB. During the design process, you will move back and forth between these two tools.

CHAPTER 3

HARDWARE TOOLS

The hardware tools used are

- Power supply
- PIC Microcontroller(PIC16F877A)
- Vibration sensor
- MEMS accelerometer sensor
- Serial communication
- GSM modem

3.1 GSM MODEM:

3.1.1 DEFINITION:

Global system for mobile communication (GSM) is a globally accepted standard for digital cellular communication. GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 MHz.

3.1.2 THE GSM NETWORK

GSM provides recommendations, not requirements. The GSM specifications define the functions and interface requirements in detail but do not address the hardware. The reason for this is to limit the designers as little as possible but still to make it possible for the operators to buy equipment from different suppliers. The

GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS).

Figure 3.1 shows the basic GSM network elements.

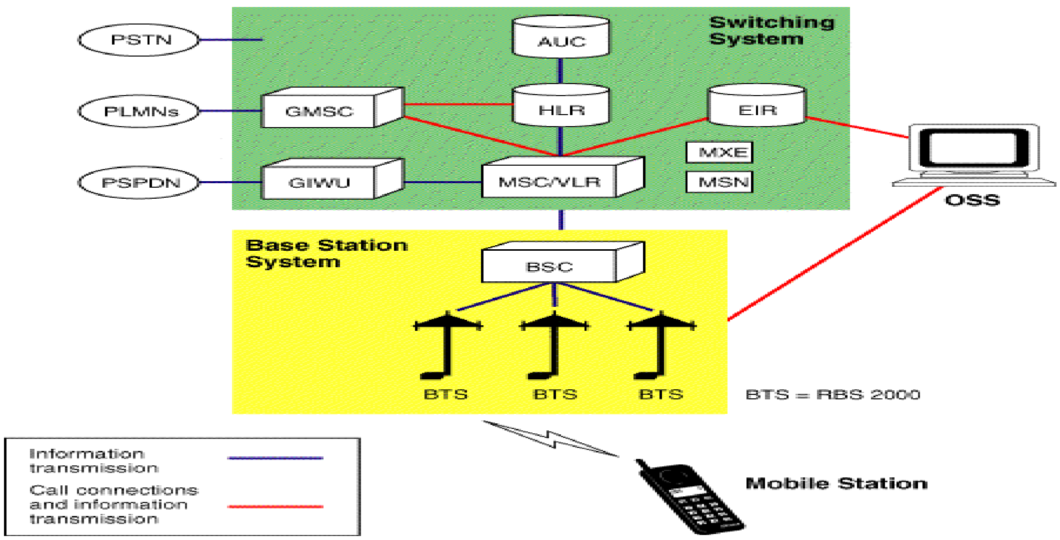


Figure 3.1 GSM network element

3.2 VIBRATION SENSOR

3.2.1 Introduction

Vibration sensors detect the vibration of the ground soil in case of a debris flow. Prior to installing a vibration sensor, it is extremely important to determine what level of vibration is appropriate to activate the sensor in case of a debris flow. It is also important to keep in mind the risk of unintentional activation caused by earthquakes, as well as areas in which there is construction traffic and other vibration causes that may activate the sensor.

3.2.3 Applications

- Vibrations produced by industrial machinery are vital indicators of machinery health.
- Machinery monitoring programs record a machine's vibration history.
- Monitoring vibration levels over time allows the plant engineer to predict problems before serious damage occurs.

Figure 3.2 shows the vibration sensor

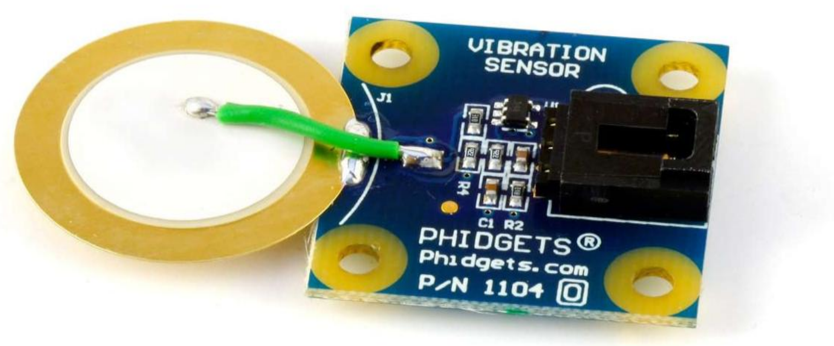


Figure 3.2 vibration sensor

CHAPTER 4

SOFTWARE CODING

VEHICLE SECTION

```
#include <LPC214X.H>
#include <stdio.h>
#include <string.h>
#include "UART_ARM.H"
#include "LCD.H"
#include "GPS.H"
#include "utility.h"
//#define RF1       19
#define Switch 24
#define Relay 31
#define Buzzer 30
unsigned char       //SORC = DEST_1,
DEST,
BUS = FALSE,
REACHD = FALSE,
TEMP,
DONE=FALSE;
signed char         TOTAL_BLNC = 60,
CURRENT_BLNC = 60,
RE_MAX = 70,
FARE_1   = 20,
FARE_2   = 40;
unsigned long TEMP1;
unsigned char       GPS_TIME[11],
GPS_LATT[12],
```

```
GPS_LONG[13],

GMT_TIME[9],sms=0,GPS_SEND=0,y[80];

#define Enable_U1IER()       U1IER=1

#define Disable_U1IER()      U1IER=0

void UART1_ISR (void)__irq;

void UART1_Intrpt(void);

void DELAY_MS(unsigned int n)

{

        int i,j;

        for(i=0;i<n;i++)

        {

                for(j=0;j<0x100;j++)

                {;}

        }

}

void UART1_Intrpt(void)

{

        U1IER    =          1;

        VICIntSelect        = 0<<7;

        VICVectCntl6        = 0x020 | 7;

        VICVectAddr6        = (unsigned long)UART1_ISR;

        VICIntEnable        = 1 << 7;

}

void UART_START(void)

{

        UART0_INIT(9600);

        UART1_INIT(9600);

        UART1_Intrpt();

}


//------------------------------------//

//***********LCD Initialize Fn ********//
```

```
//--------------------------------------//
void LCD_START(void)
{
	LCD_CONFIG(&IOPIN0,20,&IOPIN1,20);
	LCD_INIT();
	LCD_CLEAR();
//	LCD_PUTS("Wind Turbine Mon", 1, 1);
	LCD_PUTS(" Blind Bus Passenger", 1, 1);
	LCD_PUTS("   Alert System     ", 2, 1);
	delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);
	LCD_CLEAR();
}


//--------------------------------//
//***********GPIO Init Fn ********//
//--------------------------------//
void GPIO_INIT(void)
{
	IODIR1 &=  ~(1 << Switch);
	IODIR1 |=  1 << Relay;		//VOICE
	IODIR1 |=  1 << Buzzer;		//VOICE
//	IODIR0 &=  ~(1 << IR);
//	IODIR0 &=  ~(1 << PIR);
//	IODIR0 &=  ~(1 << ULTRA);
////	IOCLR1 = 0x7<<16;
//	IODIR1 &= ~(0XF << 16);			//VOICE RECOGNITION
//	IODIR1 |=  0XF << 24;		//VOICE
//	IOSET1 |=  1 << 24;
////	IOCLR1 |=  0XF << 16;
//
	IOCLR1 |=  1 << Relay;
```

```
	IOCLR1 |=  1 << Buzzer;
//	IOCLR1 |=  1 << SPEECH_3;
//
//	IOSET1 |= 1 << VOICE_1;
//	IOSET1 |= 1 << VOICE_2;
//	IOSET1 |= 1 << VOICE_3;
}


//--------------------------//
//***********Fn Main********//
//--------------------------//
void main()
{
	PINSEL0 = 0;
	PINSEL1 = 0;
	PINSEL2 = 0;
	VPBDIV = 2;

	GPIO_INIT();
	UART_START();
	LCD_START();


	while(1)
	{

			if(!(IOPIN1 & (1 << Switch)))
			{
				//IOSET1 |= 1 << Relay;
				delay_Nx10cyc(5999999);
				UART1_PUTS("AT+CMGS=\"+919944099109\"\r");delay_Nx10cyc(5999999);
```

```
UART1_PUTS("vehicle accident\r");delay_Nx10cyc(5999999);
GPS_RECEIVE_FIX(1);
UART1_PUTS("Latitude:");LATTITUDE(&GPS_LATT[0]);
UART1_PUTS(&GPS_LATT[0]);
delay_Nx10cyc(5999999);
UART1_PUTS("Longitude:");    LONGITUDE(&GPS_LONG[0]);
UART1_PUTS(&GPS_LONG[0]);
UART1_PUTC(0x1A);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);
UART1_PUTS("AT+CMGD=1,4\"\r");

}
//      if(sms == 1)
//      {
//          //UART1_PUTC('Z');
//          Disable_U1IER();  U1IER=0;
//          UART1_GETS(y);
//          //UART1_PUTS(y);
//          if(strstr(y,"2345"))
//          {
//              GPS_SEND=1;
//          }
//          if(GPS_SEND == 1)
//          {
//              GPS_RECEIVE_FIX(1);
//              IOCLR1 |= 1 << Relay;
//              IOSET1 |= 1 << Buzzer;
//
    UART1_PUTS("AT+CMGS=\"+919940027053\"\r");delay_Nx10cyc(5999999);
//              UART1_PUTS("Latitude:");LATTITUDE(&GPS_LATT[0]);
UART1_PUTS(&GPS_LATT[0]);
//              UART1_PUTS("Longitude:");    LONGITUDE(&GPS_LONG[0]);
UART1_PUTS(&GPS_LONG[0]);
//
    UART1_PUTC(0x1A);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);GPS_SEND=0;
//              //LCD_PUTS("LAT:", 1, 1);  LCD_PUTS(GPS_LATT, 1, 5);
```

```
//					//LCD_PUTS("LON:", 2, 1);  LCD_PUTS(GPS_LONG, 2, 5);
//					Enable_U1IER();    U1IER=1;
//				}
//				sms = 0;
//			 }
////			if((IOPIN1 & (1 << SPEECH_1)))
////			{
////				IOCLR1 |= 1 << VOICE_1;
////				delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);
////				IOSET1 |= 1 << VOICE_1;
////			}
////			if((IOPIN1 & (1 << SPEECH_2)))
////			{
////				IOCLR1 |= 1 << VOICE_2;
////				delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);
////				IOSET1 |= 1 << VOICE_2;
////			}
////			if((IOPIN1 & (1 << SPEECH_3)))
////			{
////				IOCLR1 |= 1 << VOICE_3;
////				delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);
////				IOSET1 |= 1 << VOICE_3;
////			}
////		if((IOPIN1 & (1 << IR)))
////		{
////			LCD_PUTS("PERSON ENTERED", 1, 1);
////			GPS_RECEIVE_FIX(1);
////			LATTITUDE(&GPS_LATT[0]); //UART0_PUTS(&GPS_LATT[0]);
////			LONGITUDE(&GPS_LONG[0]); //UART0_PUTS(&GPS_LONG[0]);
////			LCD_PUTS("LAT:", 1, 1);    LCD_PUTS(GPS_LATT, 1, 5);
////			LCD_PUTS("LON:", 2, 1);    LCD_PUTS(GPS_LONG, 2, 5);
////
```

```
////			}
////			if(!(IOPIN1 & (1 << PIR)))
////			{
////				LCD_PUTS("INTRUDER DETECTED", 1, 1);
////				GPS_RECEIVE_FIX(1);
////				LATTITUDE(&GPS_LATT[0]); //UART0_PUTS(&GPS_LATT[0]);
////				LONGITUDE(&GPS_LONG[0]); //UART0_PUTS(&GPS_LONG[0]);
////				LCD_PUTS("LAT:", 1, 1);	LCD_PUTS(GPS_LATT, 1, 5);
////				LCD_PUTS("LON:", 2, 1);	LCD_PUTS(GPS_LONG, 2, 5);
////
////			}
////			if(!(IOPIN1 & (1 << ULTRA)))
////			{
////				LCD_PUTS("OBSTACLE DETECTED", 1, 1);
////				GPS_RECEIVE_FIX(1);
////				LATTITUDE(&GPS_LATT[0]); //UART0_PUTS(&GPS_LATT[0]);
////				LONGITUDE(&GPS_LONG[0]); //UART0_PUTS(&GPS_LONG[0]);
////				LCD_PUTS("LAT:", 1, 1);	LCD_PUTS(GPS_LATT, 1, 5);
////				LCD_PUTS("LON:", 2, 1);	LCD_PUTS(GPS_LONG, 2, 5);
////
//			}

	} //while

} //main

//void UART0_ISR(void) __irq
//{
//	char ch;
//	ch = ch;
////	ch = UART0_GETC();
//
```

```
///*	if(ch==DEST) BUS = TRUE;
//
//	if(ch== REACH_ACK)	REACHD = TRUE;
//
//	if(ch== 0xA1)	DONE = TRUE;
// */
//	VICVectAddr = 6;
//}

void UART1_ISR (void)__irq
{

	 char  h;
	h = (U1RBR);
	if (h == '#')
	{

//	//if((!(Msg = U1IIR) & 0x01) == 0)			//Check Flag Status of Recieve Interrupt
//	{
//		ch = U1RBR;
//		if(ch)
//		{
//
//		if(ch== '#')


					IOCLR1 |= 1 << Relay;
					IOSET1 |= 1 << Buzzer;
					delay_Nx10cyc(5999999);

	UART1_PUTS("AT+CMGS=\"+919944099109\"\r");delay_Nx10cyc(5999999);
```

```
                    UART1_PUTS("Latitude:");LATTITUDE(&GPS_LATT[0]);
UART1_PUTS(&GPS_LATT[0]);

                    delay_Nx10cyc(5999999);

                    UART1_PUTS("Longitude:");          LONGITUDE(&GPS_LONG[0]);
UART1_PUTS(&GPS_LONG[0]);

                    delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);

    UART1_PUTC(0x1A);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);GPS_SEND=0;

                    //LCD_PUTS("LAT:", 1, 1);  LCD_PUTS(GPS_LATT, 1, 5);

                    //LCD_PUTS("LON:", 2, 1);  LCD_PUTS(GPS_LONG, 2, 5);

                    Enable_U1IER();    U1IER=1;

                }

            //case '+' :

                //sms = 1;

                //Disable_U1IER();U1IER=0;

                //UART1_GETS(y);


                //LCD_PUTS(" SEAT AVAILABLE ", 1, 1);

                //LCD_PUTS("                ", 2, 1);

//delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);

//delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);

                //      break;

            //case 'B' :

//                  sms = 2;

                //LCD_PUTS("SEAT NOT AVAILAB", 1, 1);

                //LCD_PUTS("                ", 2, 1);

//delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);

//delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);delay_Nx10cyc(5999999);

                    //break;


    VICVectAddr = 7;
```

```
}
```

TRAFFIC SECTION

```
#include <LPC214X.H>

#include "UART_ARM.H"
#include "LCD.H"
#include "ADC_ARM.H"
#include "Utility.h"


//------------------------------------//
//***********Macros Definition********//
//------------------------------------//
#ifndef TRUE
#define TRUE        1
#endif

#ifndef FALSE
#define FALSE       0
#endif

#define Enable_U0IER()      U0IER=1
#define Disable_U0IER()     U0IER=0
#define Enable_U1IER()      U1IER=1
#define Disable_U1IER()     U1IER=0


#define RF1          16
#define RF2  17
```

```
#define RF3		18
#define RF4  19
#define RED1 23
#define RED2 28
#define RED3 26
#define RED4 27
#define GREEN1 5
#define GREEN2 6
#define GREEN3 7
#define GREEN4 10
#define YELLOW1 29
#define YELLOW230
#define YELLOW3 20
#define YELLOW421
//#define Ultrasonic31
//#define R_90			15
//#define R_180			30

#define CT_DIR_FWD		1
#define CT_DIR_BWD		2
#define CT_DIR_CLK		3
#define CT_DIR_ACLK		4
#define CT_DIR_D1		5
#define CT_DIR_D2		6
//-----------------------------------//
//***********Global Variables********//
//-----------------------------------//

unsigned char		OBST1[2],
					OBST2[2],
					OBST3[2],
					dummy,
```

```
                              TEMPR[6],
                              GYRO_X[6],
                              GYRO_Y[6],
                              MEMS_X[6],
                              MEMS_Y[6],
                              CT_DIR=0;drowsy =0;car_on = 0;
int pos=0;
unsigned long       ADC_VAL0,
                              ADC_VAL1,
                              ADC_VAL2,
                              ADC_VAL3,
                              ADC_VAL4,
                              ADC_TEMP;

unsigned char  /*ARRAY[6] ,*/ flag=0 ,c1[9] , u;
  double x1 , x2 , x;

  void float2char1(float x);

  void delay(unsigned int n)
{
        int i,j;
        for(i=0;i<n;i++)
        {
                for(j=0;j<0x2700;j++)
                {;}
        }
}

//---------------------------------------//
//***********ISR Fn Declaration *********//
//---------------------------------------//
```

```
void UART0_ISR (void)__irq;

void UART1_ISR (void)__irq;

void UART0_Intrpt(void);

void UART1_Intrpt(void);



//----------------------------------------//

//************ UART0 Intrpt *************//

//----------------------------------------//

void UART0_Intrpt(void)

{

        U0IER   =       0;

        VICIntSelect    = 0<<6;                         //UART0 ('0' - irq   '1'-fiq)

        VICVectCntl6    = 0x020 | 6;                    //VIC slot enabled

        //VICVectAddr6  = (unsigned long)UART0_ISR;     //pass address of UART0

        VICIntEnable    = 1 << 6;                       //Enable  UART0 Interrupt

}



//-------------------------------------//

//***********GPIO Initialize Fn *******//

//-------------------------------------//

void INIT_GPIO(void)

{
```

```
	IODIR1 &= ~(0xF<<16);			//configure p1.16 - p1.19 as input
	IODIR1 |= (0xF<<28);			//configure p1.28 - p1.31 as output
	IODIR1 |= (0xF<<24);			//configure p1.24 - p1.27 as output



}


//--------------------------------------//
//***********LCD Initialize Fn ********//
//--------------------------------------//
void LCD_START(void)
{
	LCD_CONFIG(&IOPIN0,20,&IOPIN1,20);
	LCD_INIT();
	LCD_CLEAR();
	delay_Nx10cyc(5999999);
}



//--------------------------//
//***********Fn Main********//
//--------------------------//
void main (void)
{
	PINSEL0 = 0;
	PINSEL1 = 0X00000000;
	PINSEL2 &= 0x00000000;
	PINSEL0 |= 0x20;
	VPBDIV = 0x02;
	UART1_INIT(9600);
	UART0_INIT(9600);
	UART0_Intrpt();
```

```
        UART0_PUTC('K');
        UART1_PUTC('Q');
        IODIR1 = 0X00F00000;
        IODIR0 = 0X0000FFFF;
        LCD_START();

        INIT_GPIO();
        Enable_U0IER();

while (1)
        {

          IOCLR1 = 1<<YELLOW4|1<<YELLOW1|1<<RED1;
          IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
          IOSET0 = 1<<GREEN1;
          IOSET1 =  1<<RED2|1<<RED3|1<<RED4;
          delay(3000);
          IOCLR0 = 1<<GREEN1;
          IOSET1 = 1<<YELLOW1|1<<YELLOW2| 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
          delay(1500);

          if(((IOPIN1 & 1<<YELLOW1))&&((IOPIN1 & 1<<YELLOW2)))          ////CHECKING          RF
TRANSMITTER  STREET 1
                {

                        if(IOPIN1 & (1<<RF1))
                         {
```

```
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN1;
        IOSET1 =  1<<RED2|1<<RED3|1<<RED4;
        while(IOPIN1 & (1<<RF1));
}

else if(IOPIN1 & (1<<RF2))
{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN2;
        IOSET1 =  1<<RED1|1<<RED3|1<<RED4;
        while(IOPIN1 & (1<<RF2));
}



else if(IOPIN1 & (1<<RF3))

{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN3;
        IOSET1 =  1<<RED1|1<<RED2|1<<RED4;
        while(IOPIN1 & (1<<RF3));
}



else if(IOPIN1 & (1<<RF4))
```

```
			{
				IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
				IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
				IOCLR0 = 1<<GREEN1|1<<GREEN2|1<<GREEN3|1<<GREEN4;
				IOSET0 = 1<<GREEN4;
				IOSET1 =  1<<RED1|1<<RED2|1<<RED3;
				while(IOPIN1 & (1<<RF4));
			}

		}

IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<RED2;
IOCLR0 = 1<<GREEN1|1<<GREEN3|1<<GREEN4;
IOSET0 = 1<<GREEN2;
IOSET1 = 1<<RED1|1<<RED3|1<<RED4;delay(3000);
IOCLR0 = 1<<GREEN2;
IOSET1 = 1<<YELLOW2|1<<YELLOW3| 1<<RED1|1<<RED2|1<<RED3|1<<RED4;delay(1500);

		if(((IOPIN1 & 1<<YELLOW2))&&((IOPIN1 & 1<<YELLOW3)))
////CHECKING RF TRANSMITTER  STREET 2
			{

			if(IOPIN1 & (1<<RF1))
			{
				IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
				IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
				IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
				IOSET0 = 1<<GREEN1;
```

```
        IOSET1 =  1<<RED2|1<<RED3|1<<RED4;
        while(IOPIN1 & (1<<RF1));
}

else if(IOPIN1 & (1<<RF2))
{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN2;
        IOSET1 =  1<<RED1|1<<RED3|1<<RED4;
        while(IOPIN1 & (1<<RF2));
}

else if(IOPIN1 & (1<<RF3))

{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN3;
        IOSET1 =  1<<RED1|1<<RED2|1<<RED4;
        while(IOPIN1 & (1<<RF3));
}

else if(IOPIN1 & (1<<RF4))

{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
```

```
                IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;

                IOCLR0 = 1<<GREEN1|1<<GREEN2|1<<GREEN3|1<<GREEN4;

                IOSET0 = 1<<GREEN4;

                IOSET1 =  1<<RED1|1<<RED2|1<<RED3;

                while(IOPIN1 & (1<<RF4));

        }

        }



IOCLR1 = 1<<YELLOW2|1<<YELLOW3|1<<RED3;

IOCLR0 = 1<<GREEN1|1<<GREEN2|1<<GREEN4;

IOSET0 = 1<<GREEN3;

IOSET1 = 1<<RED1|1<<RED2|1<<RED4;delay(3000);

IOCLR0 = 1<<GREEN3;

IOSET1 = 1<<YELLOW3|1<<YELLOW4| 1<<RED1|1<<RED2|1<<RED3|1<<RED4;delay(1500);



    if(((IOPIN1 & 1<<YELLOW3))&&((IOPIN1 & 1<<YELLOW4)))
////CHECKING RF TRANSMITTER  STREET 3

        {


        if(IOPIN1 & (1<<RF1))

        {

                IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;

                IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;

                IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;

                IOSET0 = 1<<GREEN1;

                IOSET1 =  1<<RED2|1<<RED3|1<<RED4;

                while(IOPIN1 & (1<<RF1));

        }
```

```
else if(IOPIN1 & (1<<RF2))
{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN2;
        IOSET1 =  1<<RED1|1<<RED3|1<<RED4;
        while(IOPIN1 & (1<<RF2));
}


else if(IOPIN1 & (1<<RF3))

{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN3;
        IOSET1 =  1<<RED1|1<<RED2|1<<RED4;
        while(IOPIN1 & (1<<RF3));
}


else if(IOPIN1 & (1<<RF4))

{
        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
        IOCLR0 = 1<<GREEN1|1<<GREEN2|1<<GREEN3|1<<GREEN4;
        IOSET0 = 1<<GREEN4;
        IOSET1 =  1<<RED1|1<<RED2|1<<RED3;
        while(IOPIN1 & (1<<RF4));
```

```
            }
        }

IOCLR1 = 1<<YELLOW3|1<<YELLOW4|1<<RED4;
IOCLR0 = 1<<GREEN1|1<<GREEN2|1<<GREEN3;
IOSET0 = 1<<GREEN4;
IOSET1 = 1<<RED1|1<<RED2|1<<RED3;delay(3000);
IOCLR0 = 1<<GREEN4;
IOSET1 = 1<<YELLOW4|1<<YELLOW1| 1<<RED1|1<<RED2|1<<RED3|1<<RED4;delay(1500);


    if(((IOPIN1 & 1<<YELLOW4))&&((IOPIN1 & 1<<YELLOW1)))
////CHECKING RF TRANSMITTER  STREET 4
            {

            if(IOPIN1 & (1<<RF1))
            {
                IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
                IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
                IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
                IOSET0 = 1<<GREEN1;
                IOSET1 =  1<<RED2|1<<RED3|1<<RED4;
                while(IOPIN1 & (1<<RF1));
            }

            else if(IOPIN1 & (1<<RF2))
            {
                IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
                IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
                IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
                IOSET0 = 1<<GREEN2;
                IOSET1 =  1<<RED1|1<<RED3|1<<RED4;
```

```
                        while(IOPIN1 & (1<<RF2));
                }

                else if(IOPIN1 & (1<<RF3))

                {
                        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
                        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
                        IOCLR0 = 1<<GREEN2|1<<GREEN3|1<<GREEN4;
                        IOSET0 = 1<<GREEN3;
                        IOSET1 =  1<<RED1|1<<RED2|1<<RED4;
                        while(IOPIN1 & (1<<RF3));
                }

                else if(IOPIN1 & (1<<RF4))

                {
                        IOCLR1 = 1<<YELLOW1|1<<YELLOW2|1<<YELLOW3|1<<YELLOW4;
                        IOCLR1 = 1<<RED1|1<<RED2|1<<RED3|1<<RED4;
                        IOCLR0 = 1<<GREEN1|1<<GREEN2|1<<GREEN3|1<<GREEN4;
                        IOSET0 = 1<<GREEN4;
                        IOSET1 =  1<<RED1|1<<RED2|1<<RED3;
                        while(IOPIN1 & (1<<RF4));
                }

        }



    }
}
```

CHAPTER 5

RESULTS AND DISCUSSION

Figure 5.1 shows the proposed architecture diagram

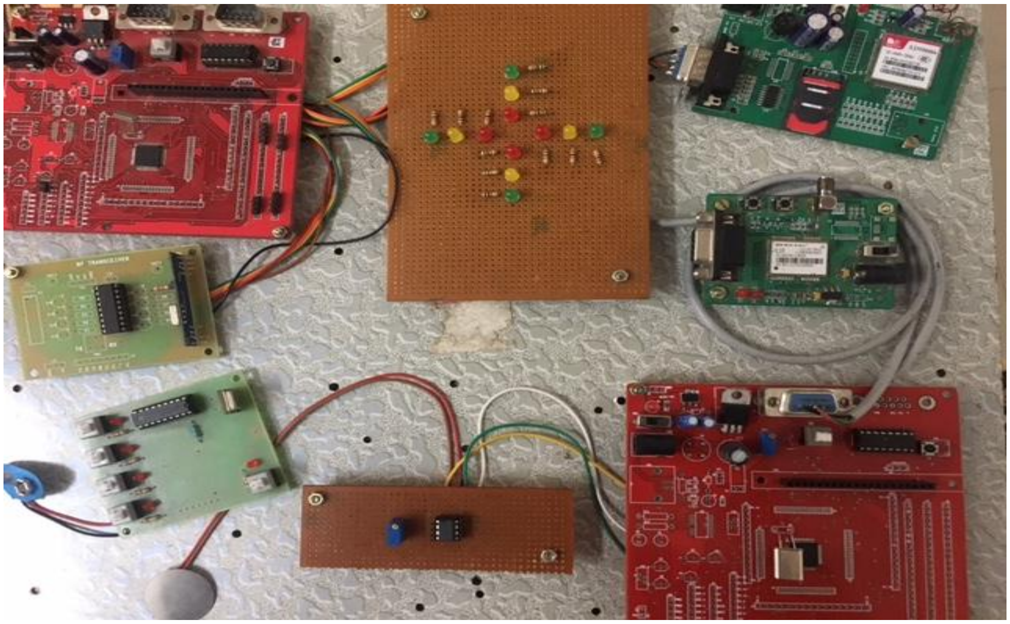


Figure 5.1 Proposed architecture

Human life is affected due to delay in the arrival of ambulance. The ambulance is not able to reach the hospital in the golden hour. It gets stuck in the traffic signals. It would be of great use to the patient if the traffic signals in the path of the ambulance are ON. There must be a system by which the ambulance would reach the accident spot and then hospital as soon as possible to carry out health services . The existing systems are post accident detection systems. It has lack of intelligence. It fails to track the rear-end collision and pre damage status. It depends on the way of monitoring people to be manual. It requires manual work to save human life which results in time delay and because of that first aid cannot be provided to the patient on time. This leads to loss of human life. In Pre-collision system, one or more systems may not activate due to sensing and tracking limitations. The actual field performance may be less effective. Limitations in the algorithms and sensors may cause difficulty in real world applications. Moreover, it may use more complex algorithms to determine collision risk. There will be different effectiveness for different algorithms. For the driver's state, there was only limited information available prior to the collision. There was no effect of pre-collision systems on driver maneuvers such as steering, other than breaking. Further simulation of driver braking deceleration without instrumentation in real-world collision was not feasible beyond constant magnitudes. It did not capture all braking inputs of driver that were possible. These are the disadvantages of existing system. In this paper, we have described a design for automatically controlling the traffic signals so that the ambulance would be able to cross all the traffic junctions without time delay. The server keeps a database for each node for easy access. Hence, each node will have a unique id for addressing the data. The ambulance is guided to the hospital by the server through the shortest route. The sensor installed in the vehicle senses the accident and Global Positioning System (GPS) tracks the /location of the accident. Through GSM (Global System for Mobile Communications), it sends the location of the accident to the ambulance section. The buzzer produces sound when accident occurs. The central unit finds the ambulance, nearest to the accident spot and also the shortest path between the location of the accident, ambulance and the nearest hospital. Here,wireless technologies are used for information transferring .The traffic signals on the path of the ambulance are controlled. When the ambulance reaches the traffic junction, the encoder converts the serial data into parallel data when it passes from the transmitter to the receiver. If the signal is red, it comes to green automatically. The decoder in the receiver section converts the parallel data into serial data when it is sent back. This helps the ambulance to cross the traffic junction as soon as possible. The prioritized traffic switching is done priority wise, i.e. if two ambulances are coming at the same time, the ambulance which will arrive first at the traffic junction will be given the priority to cross the traffic junction before the next ambulance arrives. In this way, using wireless technologies, the information is transferred and the traffic signals are controlled so that the ambulance would be able to reach the hospital on time.

5.2 GSM AND GPS

Figure 5.2 shows the GSM and GPS module

Figure 5.2 GSM and GPS module

In Vehicle Section, all the equipments are connected to microcontroller. The Piezoelectric sensor is used as vibration sensor to measure flex, touch, vibration and shock. Piezoelectric sensor is a device that uses piezoelectric effect to measure changes in acceleration, pressure, temperature, strain or force by converting them to an electric charge. Sensor based on piezoelectric effect can operate from transverse, longitudinal, shear forces and are insensitive to electric field and electromagnetic radiation. This piezoelectric sensor measures dynamic pressure which includes blast, ballistics and engine combustion under varying condition. An electronic amplifier is an electronic device that increases power of a signal and converts alternating current into direct current. Here, we have used Microcontroller ARM7 for this vehicle section. ARM7 is a group of older 32-bit ARM processor.ARM is a family of instruction set architecture for computer processor based on a reduced instruction set computing. A RISC-based computer design approach means ARM processor requires significantly fewer transistors. LPC2148 is the widely used IC from ARM7 family which we have used in vehicle section. It is pre-loaded with many inbuilt peripherals making it more efficient. Power supply, crystal oscillator, reset circuit, UART are the minimum listed hardware needed for LPC2148. It works on 3.3V power supply, transformer is used to step down 230V AC to 9V AC supply and provide isolation between power grid and circuit. Rectifier in LPC2148 is used to convert AC supply into DC and regulator is used to regulate DC supply output, reset button is essential to avoid programming pitfalls and provide clock for RTC operation. LPC2148 has inbuilt ISP which means we can program it within the system using serial communication on COM0. Indicator indicates through buzzer whether accident has occurred. The buzzer produces sound when accident occurs. MEMS sensor used for this project is one of the most promising technologies for 21st century. It is an enabling technology for pressure and acceleration sensors. MEMS based sensors provide an interface that can sense and process. They are a class of devices which makes small mechanical and electrical components on a single chip. They are crucial components in hard disk drives, automotive electronics, computer peripherals, wireless devices, medical equipment and smart mobile electronic devices such as PDAs and cell phones. The benefits of MEMS are high performance, miniaturization, integration, low power and low cost. GSM (Global System for Mobile communications) is a widely used digital mobile telephony system. GSM uses time division multiple access (TDMA) and is the most widely used of the three digital wireless telephony technologies (TDMA, GSM , and CDMA). The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The sensor installed in the vehicle unit senses the accident and GPS tracks the location of the accident. Through GSM, it sends the location of the accident to the ambulance section. The buzzer produces sound when accident occurs. The central unit finds the ambulance, nearest to the accident spot and also the shortest path between the location of the accident, ambulance and the nearest hospital. The ambulance crosses all the traffic junctions by automatically controlling the traffic signals and reaches the nearest hospital. Here, wireless technologies are used to transfer information.

5.3 SIGNAL SECTION

Figure 5.3 shows the traffic signal

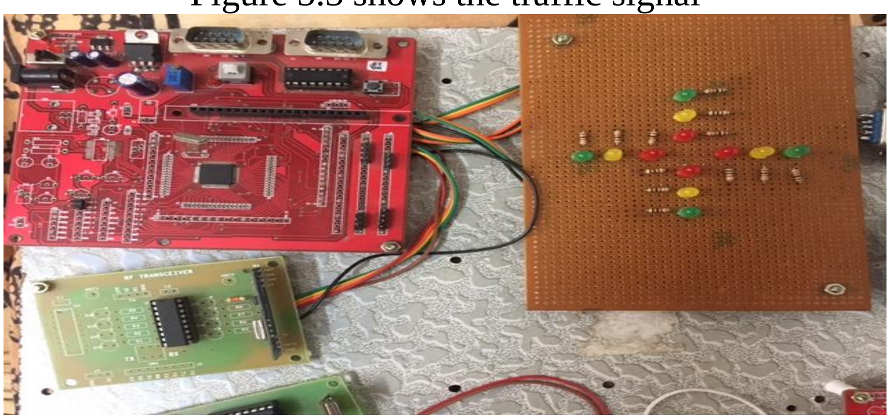


Figure 5.3 Traffic signal block

In Signal Section, the functions of the microcontroller section are same as in the ambulance section. Here, we used reader device, decoder unit and signal indicator. The reader device receives the data which the RF transmitter sends from the ambulance section. Any device may act as a reader that can display text on a screen. A decoder is a device that performs the reverse operation of an encoder. To recover the original information, it undoes the encoding. Normally, the same method which is used to encode is reversed to decode. It is a combinational circuit that converts binary information. Here, the decoder unit converts the parallel data into serial data and sends it to the microcontroller section. The received signal strength indicator (RSSI) measures the power present in a radio signal which is received. RSSI is a radio receiver technology metric, which is normally invisible to the user of the device which consists of receiver, but is directly known to users of wireless networking. The output of RSSI is a DC analog level. The ambulance unit is the transmitter and each signal is the receiver. When the data is transmitted to the receiver, the signal comes to green automatically. The decoder converts the data from parallel to serial because the controller knows only serial language. In this way, this system helps the ambulance to reach the emergency site and then to hospital without time delay so that intensive care can be given to the patient in the golden hour and many lives can be saved.

5.4VEHICLE SECTION

Figure 5.4 shows the vehicle section

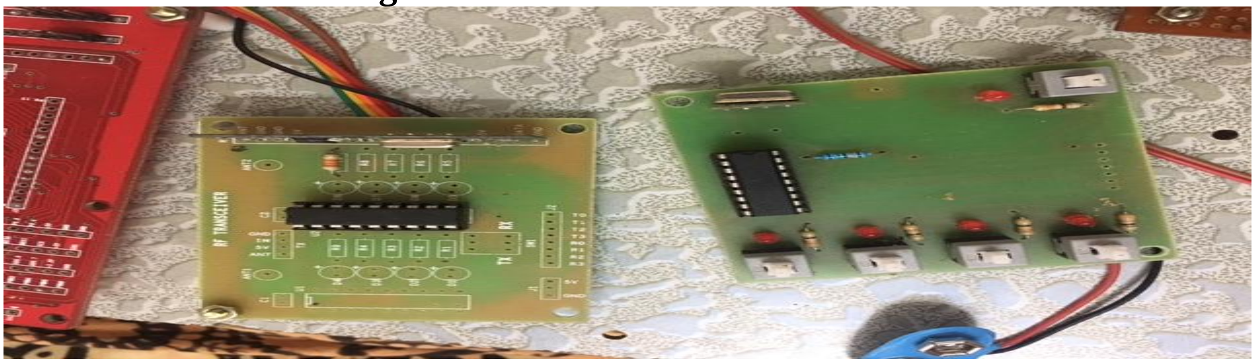


Figure 5.4 vehicle section

In Ambulance Section, we used PIC microcontroller. The serial number of the ICs 16F877A. This section consists of crystal oscillator, power circuit and serial communication. 9V input supply is given and the operating voltage is 5V. The crystal oscillator is used to work according to the frequency change. In a PIC microcontroller, there are totally 40 pins and 5 ports; port A to port E. PIC is a family of modified architecture microcontrollers. The name PIC initially referred to Peripheral Interface Controller. The MAX232 is an IC, which converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. Here, it is used for the purpose of serial communication. It is a dual driver/receiver. The display unit is for the purpose of driver's reference. An encoder is a device, transducer, circuit, algorithm, software program, or person, which converts data from one format or code to another, for the purposes of speed, standardization, security, secrecy or compressions. Here, the encoder converts the serial data into parallel data because the controller performs only serial functions and wireless is parallel communication. The RF transmitter consists of switches, the signal changes according to it. An RF module is an electronic device used to transmit or receive radio signals between two devices. It is often preferable to communicate with other device wirelessly in an embedded system. The wireless communication may be performed through optical communication or RF communication. The choice is RF for many applications as it does not require line of sight. RF communications use a transmitter or receiver.

CONCLUSION

Our project has been successfully completed. By doing this project we have obtained the valuable knowledge.

In this project, we have described a design for automatically controlling the traffic signals so that the ambulance would be able to cross all the traffic junctions and reach hospital without time delay. Human life is affected due to delay in the arrival of ambulance. The ambulance is not able to reach the hospital in the golden hour. The existing system has many disadvantages. It depends on the way of monitoring people to be manual which results in time delay and because of that health services cannot be provided to the patient on time which leads to loss of human life. In our proposed system, the ambulance is guided to the hospital by the central unit through the shortest route. The sensor installed in the vehicle senses the accident and Global Positioning System (GPS) tracks the location of the accident. Through GSM (Global System for Mobile Communications), it sends the location of the accident to the ambulance section. The central unit finds the ambulance, nearest to the accident spot and also the shortest path between the location of the accident, ambulance and the nearest hospital. Here, wireless technologies are used to transfer information.

REFERENCES

1. Xiaolin Lu, Develop Web GIS Based Intelligent Transportation Application Systems with Web Service Technology, Proceedings of International Conference on ITS Telecommunications, 2006.
2. Kristofer D. Kusano and Hampton C. Gabler , Member, IEEE, "Safety Benefits of Forward Collision Warning, Brake Assist, and Autonomous Braking Systems in Rear-End Collisions", 2011.
3. Wang Wei, Fan Hanbo, Traffic Accident Automatic Detection and Remote Alarm Device.
4. G. Derekenaris, J. Garofalakis, C. Makris, J. Prentzas, S. Sioutas, A. Tsakalidis, "An Information System for the Effective Management of Ambulances", 2000.
5. Zhaosheng Yang. Study on the schemes of traffic signal timing for Priority Vehicles Based on Navigation System, 2000.
6. Maria Teresa Aviles, Roberto Becchini, "Mission support for emergency operators", 2007.
7. Samir El-Masri, Basema Saddik, "Mobile Emergency System and integration", 2011.
8. Sara Nazari, M. Reza Meybodi, M. Ali Salehigh, Sara taghipour, "An Advanced Algorithm for Finding Shortest Path in Car Navigation System", Proceedings of 1st International Conference on Intelligent Network and Intelligent Systems, pages: 671-674, 2008.
9. Katsunori Tawara, Naoto Mukai, Traffic Signal Control by using Traffic Congestion Prediction based on Pheromone Model, Proceedings of 22nd International Conference on Tools with Artificial Intelligence, 2010.
10. Dian-liang XIAO, Yu-jia TIAN, "Reliability of Emergency Rescue System on Highway", 2009.
11. XU Guang-hui, Deng Jun, Huang Yong-bo, "The Research and Design of the Control System of the Omnidirectional Self-balancing Intelligent Ambulance", 2011.
12. Noraimi Azlin Mohd Nordin, Norhidayah Kadir, Zati Aqmar Zaharudin and Nor Amalina Nordin, "An Application of the A* Algorithm on the Ambulance Routing", 2011.
13. S.Jagadeeshwaran, N.Dinesh, "Automatic Ambulance Rescue System. K.Athavan, G.Balasubramanian", 2012.
14. Wei Yan Ma Zhigang, Qiu sihai, "System of Medical Emergency Ambulance for Community based on Zigbee", 2010.