ACCIDENT DETECTION AND ALERTING SYSTEM

A PROJECT REPORT

Submitted by

KAVYA REDDY N	BL.EN.U4CSE19091
PREETHI REDDY M	BL.EN.U4CSE19110
SHRAVYA V	BL.EN.U4CSE19140

for the course

19CSE303- Embedded Systems

Evaluated by

<<Faculty Signature>>

AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDHYAPEETHAM

BANGALORE-560 035

November-2021

ACCIDENT DETECTION AND ALERTING SYSTEM USING 8051 MICROCONTROLLER

1. Project Description

This project is an 8051 Microcontroller-based accident detection and alerting system. When a person riding a bike is involved in an accident, there is a potential that he or she could suffer a catastrophic injury or die instantly, and no one will be able to assist him. This system, on the other hand, provides a solution to the issue. The system functions as an accident identification system, gathering and transmitting information on vehicles involved in accidents to the nearest control room.

The user car is equipped with a GSM module, a vibration sensor, and a microprocessor for this purpose. The vibration sensor detects and outputs whenever a user vehicle is involved in an accident. The microcontroller then detects this output. This change detection signal is now sent to a GSM Module by the microcontroller. The GSM Module sends the accident data to the GSM Module via SMS. We can provide anyone with a phone number. For instance, a police number, an ambulance number, a doctor's number, and so forth. We're also using an LCD Module in this case. This shows the current state. When an accident occurs, the buzzer will sound.

2. Software and Hardware tools used:

HARDWARE REQUIREMENTS:

8051 Microcontroller

8051 microcontroller is designed by Intel in 1981. It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, 2 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement. An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 12 MHz

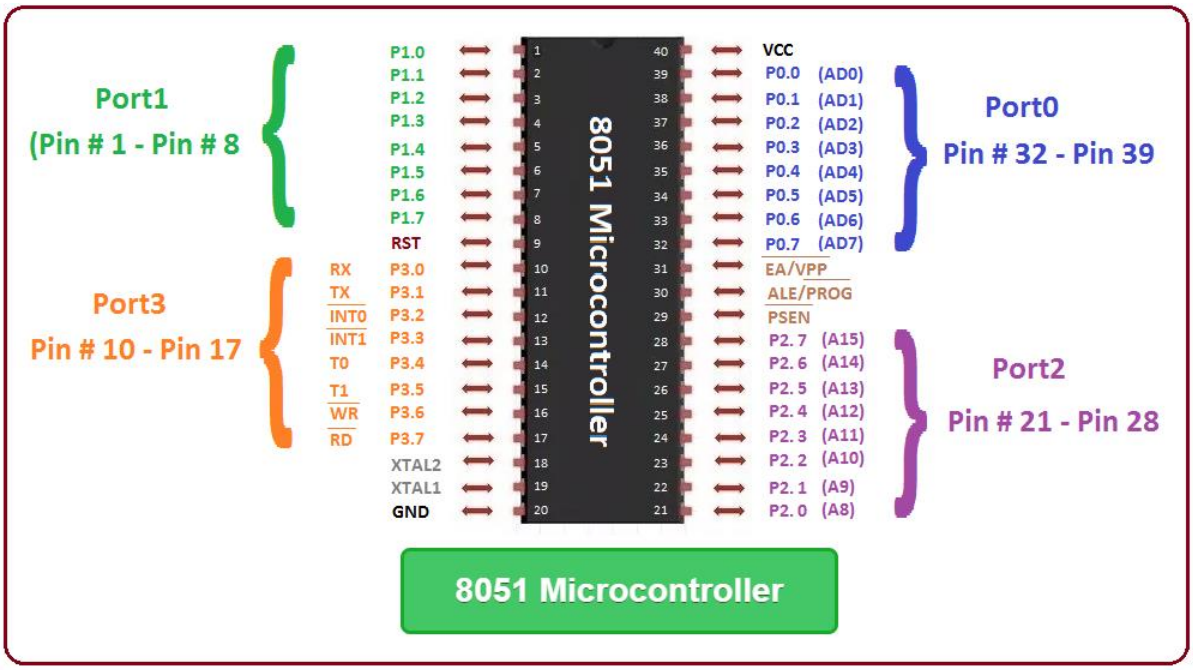

Figure 2.0: 8051 Microcontroller with its different ports and pins.

GSM

GSM stands for "global system for mobile communication," and it refers to a second-generation (2G) mobile network. This is commonly used for mobile communication all over the world. This GSM gadget features a sim slot into which a sim with a unique number can be inserted; this unique number is used for contact. This GSM gadget has a unique number called an imei number, which varies depending on the hardware kit. The device is utilised to transfer data in our project. The data from the GPS is sent to a specific mobile phone through GSM.

Figure 2.1: GSM module with RS232

LCD

LCD stands for “Liquid Crystals Displays” is a flat panel display which uses liquid crystals in its primary form of operation. It is commonly used for portable electronic games, as view finders for digital cameras, monitors for computers, etc. The most used LCD is of 16 x 2 dimensions. It can display 16 character and has 2 lines. Each character takes 5 x 7-pixel matrix.
LCD contains 2 registers – Command registers which is used to store commands given to the LCD and Data register which is used to store the data that is to be displayed in the screen.
The LCD display module requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The 3 control lines are EN, RS and RW.
EN – It is called “Enable”. This control line is used to tell the LCD that you are sending it data. Data is sent to the data pins when a high (1) to low (0) pulse is given.
RS – It is called as “Register Select”. When RS is low (0), the data is treated as a command. The command register is selected. When RS is high (1), the data is sent as text to be displayed on the screen. Data register is selected.
RW – It is the Read/ Write control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is efficiently querying the LCD.
The data bus consists of 4 or 8 lines. They are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, DB7.
RS is connected to Port 2.0 (P2.0), RW is connected to Port 2.1 (P2.1), EN is connected to Port 2.2 (P2.2).
Data lines are connected into Port 1 (P1)

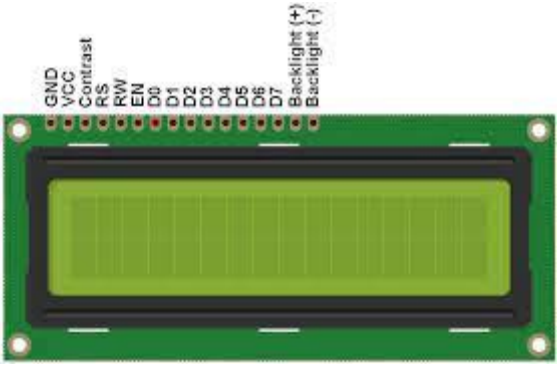

Figure 2.2: LCD module with its pins.

Vibration Sensor

Vibration sensors are sensors used to measure, display and analyze frequency, displacement, velocity or acceleration. There are various types but the most commonly used vibration sensor is accelerometer. It produces electrical signal proportional to acceleration of vibrating component.
Vibration sensors respond to repetitive mechanical motion. Most of the sensors are available with their contacts "normally open" type. The contacts close when the sensor vibrates in its designed frequency range. Frequency range and sensitivity are manually adjustable in some of the sensors.

Figure 2.3: Vibration Sensor

Buzzer

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.
The pin configuration of the buzzer is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas

the negative terminal is represented with the '-'symbol or short terminal and it is connected to the GND terminal.

Figure 2.4: Buzzer

SOFTWARE REQUIREMENTS

Keil

Keil compiler is a software used where the machine language code is written and compiled. After compilation, the machine source code is converted into hex code which is to be dumped into the microcontroller for further processing. Keil compiler also supports C language code.

Figure 2.5: Keil Software logo

Proteus

Proteus is used to simulate, design and drawing of electronic circuits. It was invented by the Lab center electronic.
By using proteus you can make two-dimensional circuits designs as well.
With the use of this engineering software, you can construct and simulate different electrical and electronic circuits on your personal computers or laptops. There are numerous benefits to simulate circuits on proteus before make them practically.

Figure 2.6: Proteus logo

3. SYSTEM MODEL

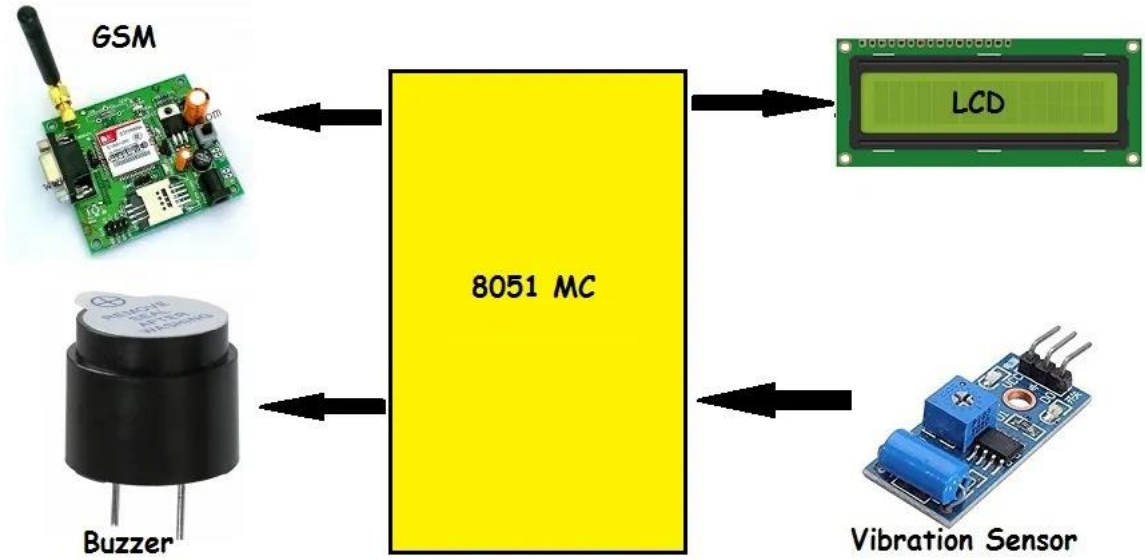

Figure 3.0: Block diagram of the Accident detection and alerting system. The vibration sensor senses an accident and send a signal to the 8051 microcontroller. The controller then sends this signal to the GSM module and the LCD module which will turn on the buzzer and send a SMS.

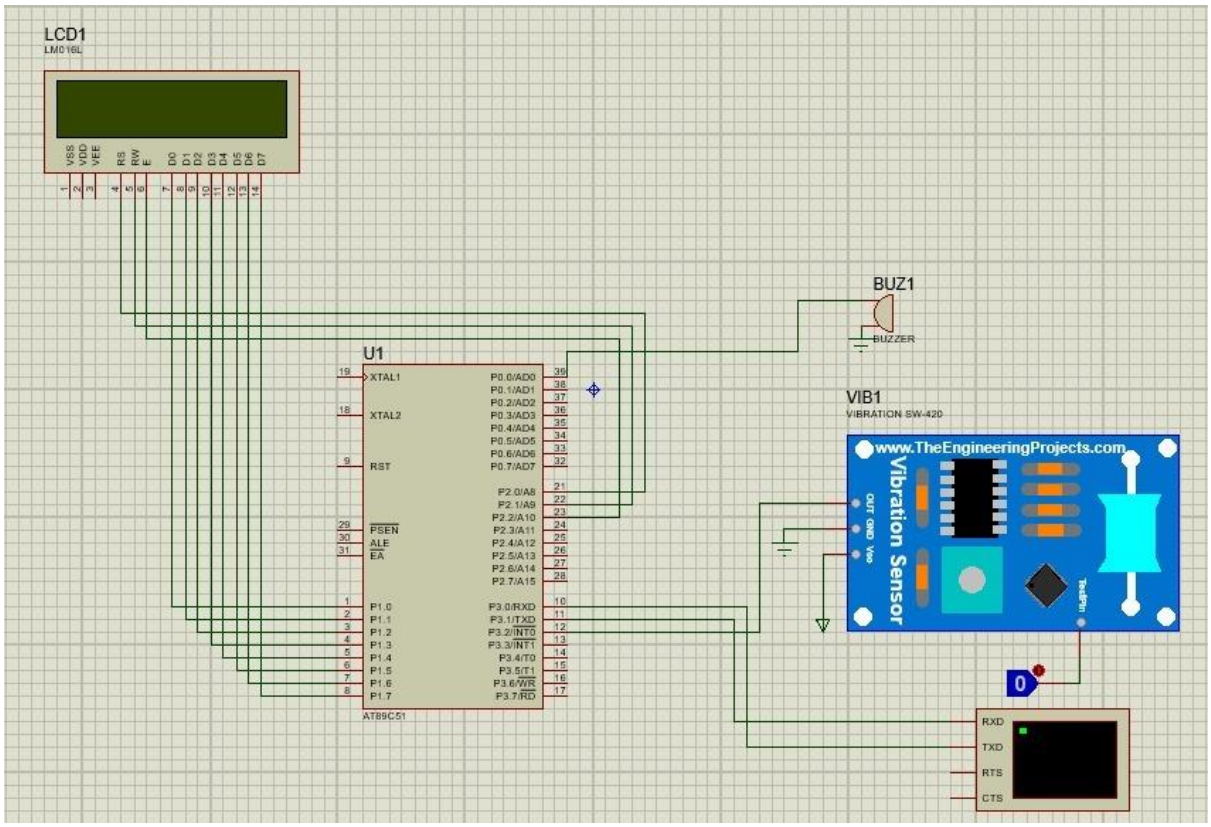

Figure 3.1: The proteus implementation of the block diagram.

4. IMPLEMENTATION DETAILS

LCD.h

```
#define LCDDATA P1
#define DELAY for(i=0;i<1200;i++)

sbit RS = P2^0;
sbit RW = P2^1;
sbit EN = P2^2;

void cmdwrt(unsigned char);
void datawrt(unsigned char);
void  LCD_init();
void display(unsigned char *str);

void LCD_init()
{
    unsigned int i,j;
    int com[5]={0x38,0x0C,0x01,0x06,0x80};
    for(j=0;j<=4;j++) {
    cmdwrt(com[j]);
    DELAY;
    }
}

void cmdwrt(unsigned char dat)
{
    unsigned int i;
    LCDDATA=dat;
    RS = 0;
    RW = 0;
    EN = 1;
    DELAY;
    EN = 0;
}

void datawrt(unsigned char dat)
{
    unsigned int i;
    LCDDATA=dat;
    RS = 1;
```

```
    RW = 0;
    EN = 1;
    DELAY;
    EN = 0;
}

void display(unsigned char *str)
{
    int i;
    for(;*str!=0;str++) {
    datawrt(*str);
         DELAY;
    }

}
```

GSM.h

```
code unsigned char SMS1[2] = "AT" ;
code unsigned char SMS2[9] = "AT+CMGF=1" ;
code unsigned char SMS3[8]= "AT+CMGS=" ;
code unsigned char SMS4[3]= "ATD" ;
code unsigned char SMS5[3]= "ATH" ;


void sendMSG(unsigned char *num , unsigned char *msg);
void delay1(unsigned int tim);
void sendserial(unsigned char mydata1);
void call(unsigned char *num1);

unsigned char i;

void sendMSG(unsigned char *num , unsigned char *msg)
{
    for (i=0;i<2;i++)
    sendserial(SMS1[i]);
    sendserial(0X0D);
    delay1(20);

    for (i=0;i<9;i++)
    sendserial(SMS2[i]);
    sendserial(0X0D);
    delay1(20);
```

```
    for (i=0;i<8;i++)
    sendserial(SMS3[i]);
    sendserial(0x22);

    for(;*num!=0;num++)
    sendserial(*num);
    sendserial(0x22);  // "
    sendserial(0X0D);
    delay1(20);

    for(;*msg!=0;msg++)
    sendserial(*msg);
    sendserial(0X1A);
    delay1(30);

}

void call(unsigned char *num1)
{
    for (i=0;i<2;i++)
    sendserial(SMS1[i]);
    sendserial(0X0D);
    delay1(20);

    for (i=0;i<9;i++)
    sendserial(SMS2[i]);
    sendserial(0X0D);
    delay1(20);

    for (i=0;i<3;i++)
    sendserial(SMS4[i]);

    for(;*num1!=0;num1++)
    sendserial(*num1);
    sendserial(0x3b);
    sendserial(0X0D);
    delay1(20);
    delay1(100);

    for (i=0;i<3;i++)
    sendserial(SMS5[i]);
    delay1(40);
}
```

```
void delay1(unsigned int tim)
{
    unsigned int h;
    for(h=0;h<=tim;h++) {
        TMOD=0X21;
        TH0=0x4B;
        TL0=0xFD;
        TR0=1;
        while(TF0==0);
        TF0=0;
    }
}

void sendserial(unsigned char mydata1)
{
    TI=0;
    SBUF= mydata1;
    while(TI==0);
}

void GSM_init()
{
    SCON=0x50;
    TMOD=0x21;
    TH1=0xFD;
    TL1=0xFD;
    TR1=1;
}
```

Main.c

```
#include<reg51.h>
#include"GSM.h"
#include"LCD.h"

#define NUMBER1 "9182328024"
#define NUMBER2 "9441233636"


sbit vib = P3^2;
sbit buzzer = P0^0;
```

```
void main()
{
    unsigned int r;
    GSM_init();
    LCD_init();
    cmdwrt(0x80);
    display("INITIALISING....");
    for(r=0;r<40000;r++);
    cmdwrt(0x80);
    display("DETECTING ACCIDENT");
      for(r=0;r<20000;r++);
    buzzer=1;
    while(1) {
        if(vib==1) {
            buzzer=0;
            cmdwrt(0x80);
            display("DETECTING VIBRATION");
                        for(r=0;r<20000;r++);
            cmdwrt(0xC0);
            display(" VIBRATION: YES ");
                        for(r=0;r<50000;r++);
                        cmdwrt(0x01);
            cmdwrt(0x80);
            display("SENDING MSG.....");
            sendMSG(NUMBER1,"ACCIDENT DETECTED");
            sendMSG(NUMBER2,"ACCIDENT DETECTED");
            cmdwrt(0xC0);
            display("    MSG SENT    ");
            for(r=0;r<20000;r++);
                        cmdwrt(0x01);
            for(r=0;r<20000;r++);
            cmdwrt(0x80);
            display("CALLING.........");
            call(NUMBER1);
            call(NUMBER2);
            cmdwrt(0x80);
            display("ACCIDENT DETECTED");

        }
        else
        {
            cmdwrt(0x80);
            display("DETECTING VIBRATION");
                        for(r=0;r<20000;r++);
```

```
                buzzer=1;
               cmdwrt(0xC0);
               display(" VIBRATION: NO  ");
          }

     }
}
```

5. SAMPLE OUTPUT

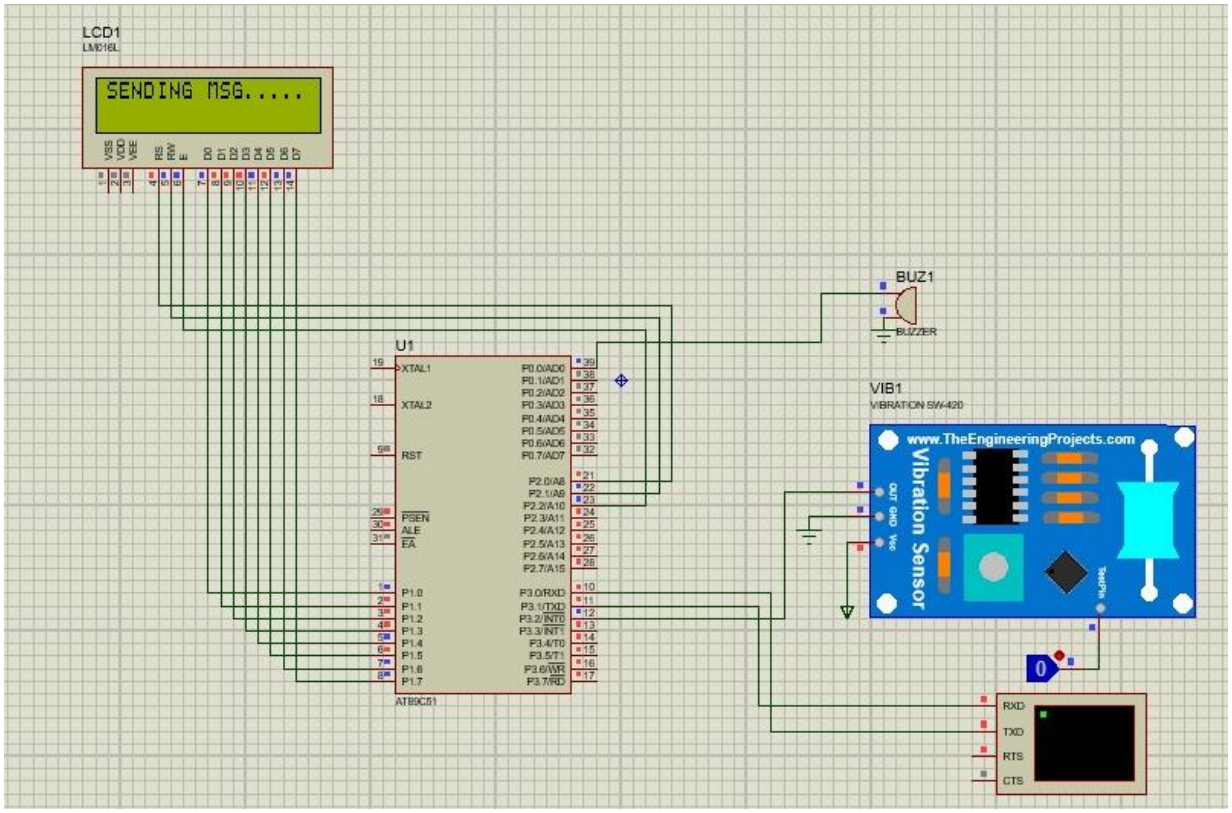

Figure 5.0: Snapshot of the project from proteus when accident has been detected and message is being sent.

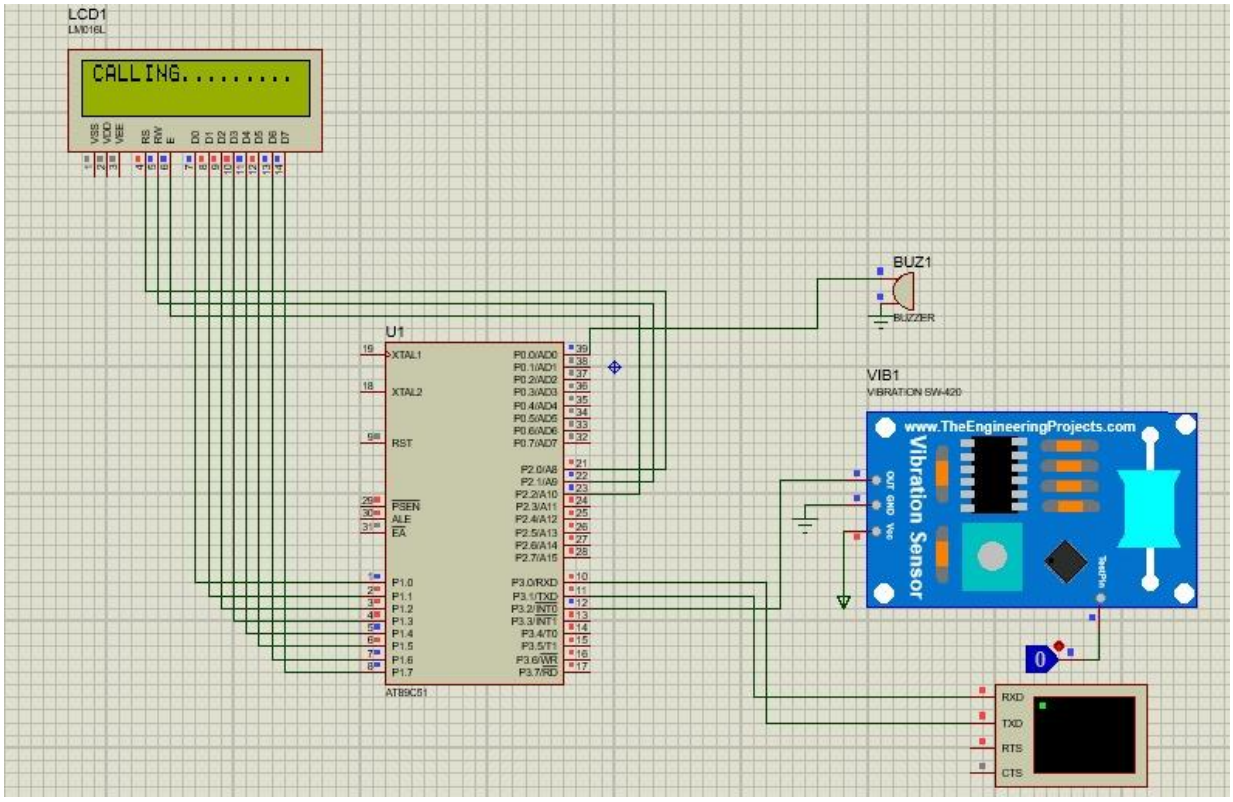

Figure 5.1: Snapshot of the project from proteus when accident has been detected and call is being sent.

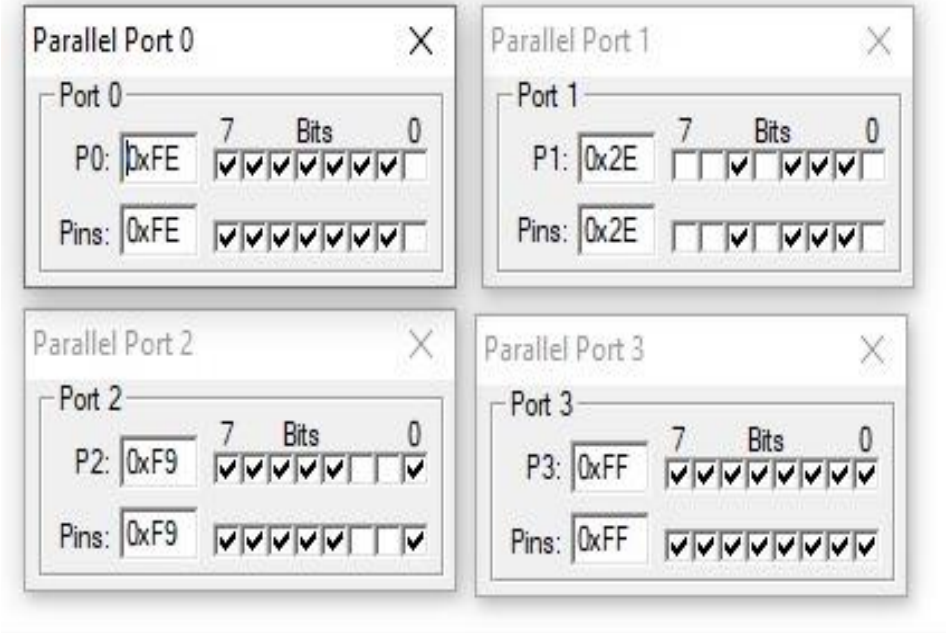

Figure 5.2: Snapshot of port peripherals in Keil software

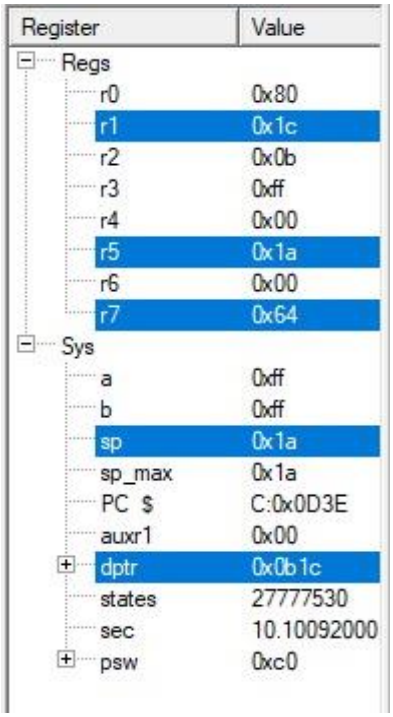

Figure 5.3: Snapshot of Registers form the Keil software.