R.V. COLLEGE OF ENGINEERING
BENGALURU-560059
(Autonomous Institution Affiliated to VTU, Belgaum)

MICROCONTROLLERS AND EMBEDDED SYSTEMS (18CS44)

MCES LAB PART B

Topic: Instant Tele-Emergency Alarm Module - iTEAM

Submitted by

NISHCHAL J	**1RV18CS107**
PAVAN KR	**1RV18CS112**
NISARG	**1RV18CS106**

Submitted to

Prof. Mamatha T.

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

R.V. College of Engineering, Bengaluru – 560059

CONTENTS

Abstract

This project aims to propose and build a state-of-the-art technology called the **Instant Tele-Emergency Alarm Module - iTEAM** used for automatic calling and messaging service to the nearest FIRE-STATION during a fire breakout in a hospital. The same device can be used in hospitals to contact a doctor through a Call or SMS during emergency cases. This device uses GSM Technology based on Cellular Networks which does not require internet services for its operation, thus, can be installed in all the remote locations across the country.

Introduction

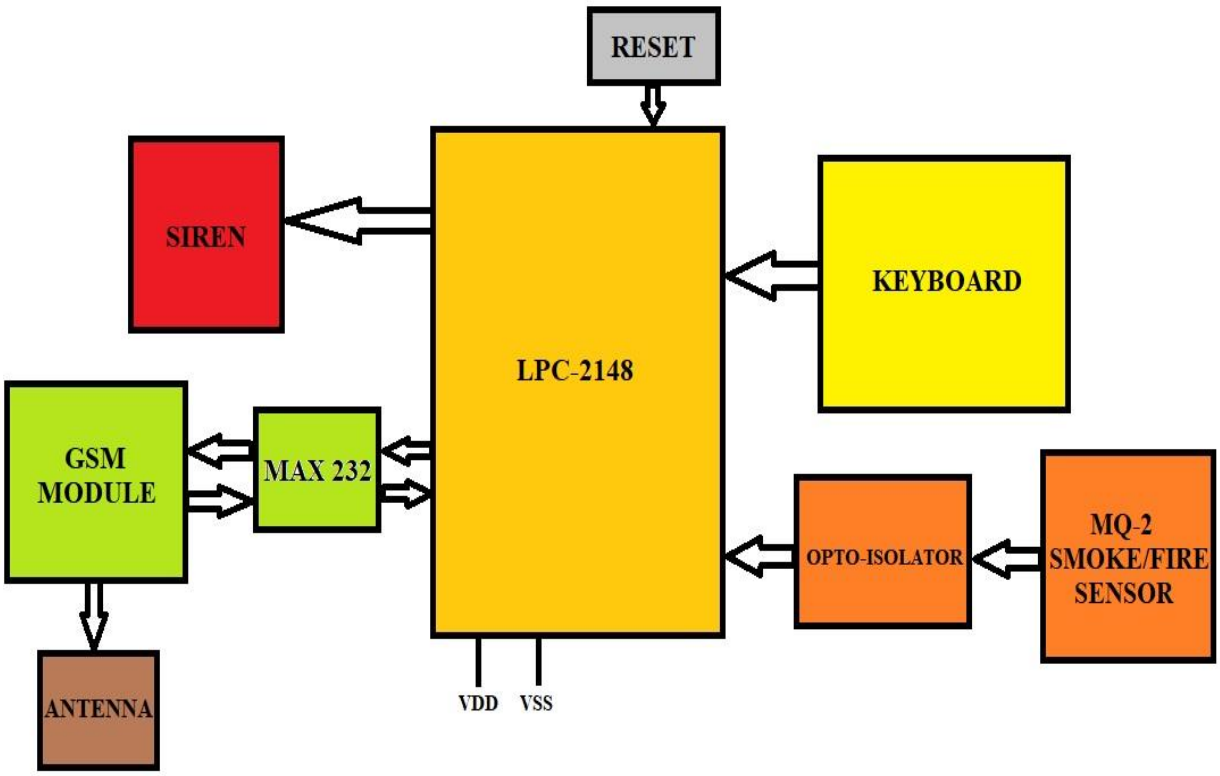
The hospitals and health departments play a crucial role in the proper functioning of a nation. The health sector is just as vital to the economy as the industrial and agricultural sectors. This simple fact couldn't be stressed more than in the ongoing scenario of a pandemic. It has been realised that efficient health services and infrastructure have greatly helped in putting the situation under hold and minimising the health risks of the citizens.

While some countries have been well prepared, some have suffered grave consequences due to poor health infrastructure. In addition to the availability of health personnel, clean wards and essential equipments, presence of efficient emergency and immediate alert services can be the matter of life and death, not just in the case of a pandemic but for everyday operations. The aim of our project is to develop one such general purpose utility device that can vastly improve the health services while ensuring the security of the health personnel and patients.

This state-of-the-art device built on LPC2148 microcontroller helps to detect anonymous smoke and fire breakout in hospitals with the help of the MQ-2 smoke sensor, which will then intimate the nearest Fire-station through a phone call or SMS with the help of SIM900A GSM module based on Cellular networks, thus eliminating the requirement of internet services. The same device can be used to reach out a particular doctor in the hospital during emergency cases, just from a click of a button on the keypad attached to the device.

Thus, the multi-utility device is a promising technology which can reform the emergency healthcare services in the country.

FUNCTIONAL BLOCK DIAGRAM



Benefits of iTEAM

- Extremely Low-cost device used for multi-purpose utilities
- Flexibility / modularity in control
- Scalable, Robust and Reliable
- Low maintenance cost and No Internet Services required
- Calls and Messages can be sent at nominal tariffs given by cellular network operators like BSNL, AIRTEL, JIO

Components Required

- ARM7 LPC2148 microcontroller
- SIM900A GSM/GPRS module
- Gas/smoke sensor MQ2
- 4x4 matrix keypad
- Piezoelectric buzzer

Working

The project intends to develop a general purpose utility device by interfacing the GSM module, gas sensor, keypad and buzzer with the LPC2148 microcontroller. The device serves two specific purposes:

(i) To alert the doctors about their immediate requirement in a specific ward

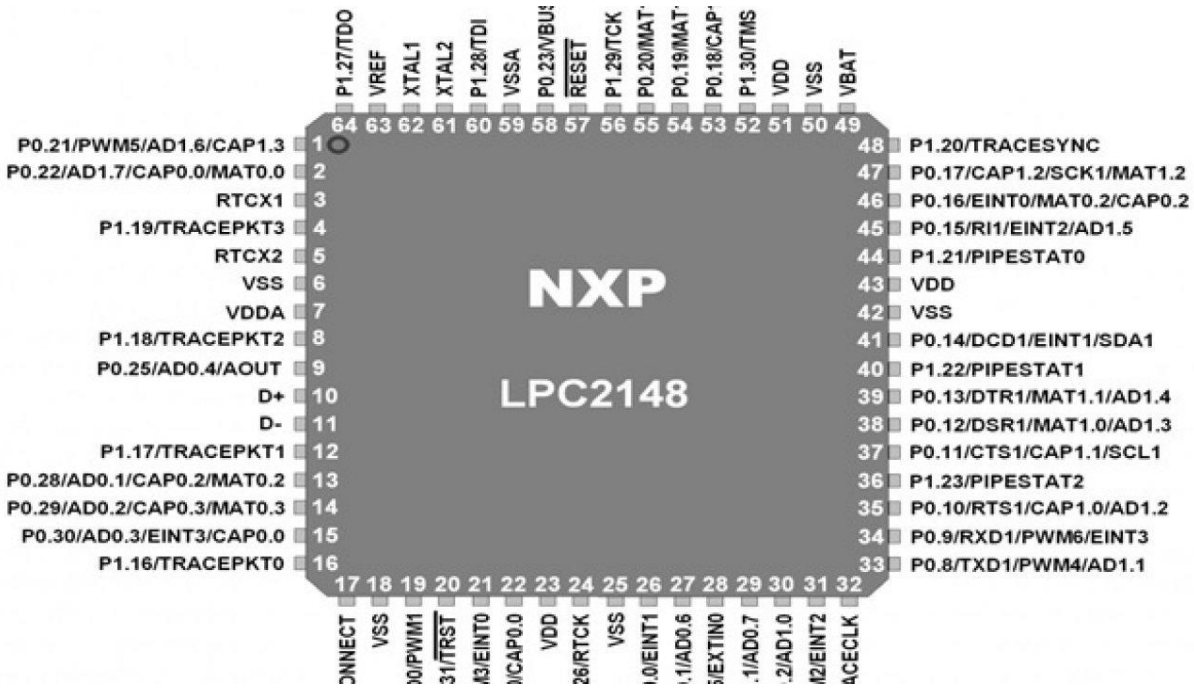
- This is done by pressing the desired key on the device by health staff that sends a message to the doctor's phone

(ii) To act as a fire alarm

- This is achieved by detecting smoke in the building and alerting the occupants with a buzzer alarm, while sending a message to the fire station instantly.

LPC2148 Microcontroller

The LPC2148 microcontroller is designed by Philips (NXP Semiconductor) with several in-built features & peripherals. Due to these reasons, it will make more reliable as well as the efficient option for an application developer. LPC2148 is a 16-bit or 32-bit microcontroller based on ARM7 family.



General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either input pin or output pin, whose behaviour can be controlled at the run time. A group of these pins is called a port (Example, Port 0 of LPC2148 has 32 pins).

LPC2148 has two 32-bit General Purpose I/O ports.

1. PORT0
2. PORT1

PORT0 is a 32-bit port

- Out of these 32 pins, 28 pins can be configured as either general purpose input or output.
- 1 of these 32 pins (P0.31) can be configured as general-purpose output only.
- 3 of these 32 pins (P0.24, P0.26 and P0.27) are reserved. Hence, they are not available for use. Also, these pins are not mentioned in pin diagram.

PORT1 is also a 32-bit port. Only 16 of these 32 pins (P1.16 – P1.31) are available for use as general-purpose input or output.

SIM900A GSM/GPRS module

SIM900A Modem can work with any GSM network operator SIM card just like a mobile phone with its own unique phone number.

SIM900A GSM/GPRS modem is plug and play modem with RS232 serial communication supported. Hence, the advantage of using this modem will be that its RS232 port can be used to communicate and develop embedded applications.

Applications like SMS Control, data transfer, remote control and logging can be developed. SIM900 modem supports features like voice call, SMS, Data/Fax, GPRS etc.

MQ2 Gas/Smoke Sensor

MQ2 is one of the commonly used gas sensors in MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type Gas Sensor also known as chemiresistors as the detection is based upon change of resistance of the sensing material when the gas comes in contact with the material. Using a simple voltage divider network, concentrations of gas can be detected.

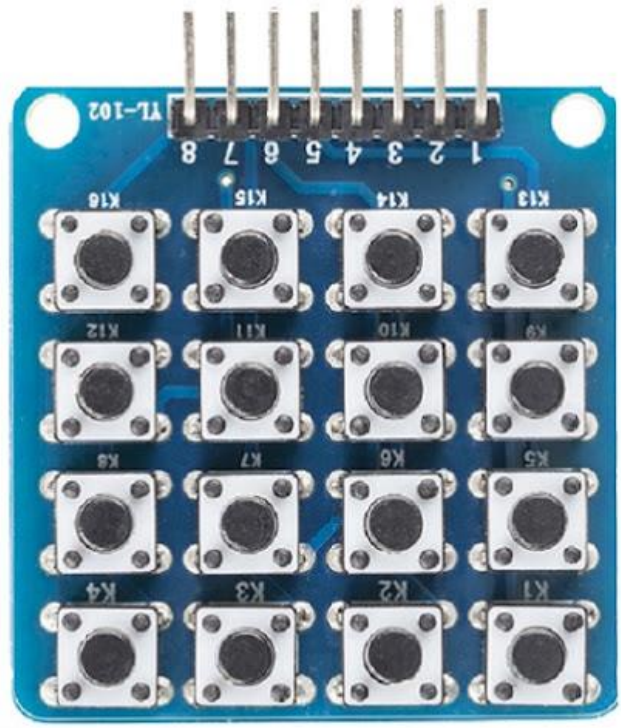
MQ2 Gas sensor works on 5V DC and draws around 800mW. It can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide concentrat ions anywhere from 200 to 10000ppm.

The analog output voltage provided by the sensor changes in proportional to the concentration of smoke/gas. The greater the gas concentration, the higher is the output voltage; while lesser gas concentration results in low output voltage. The following animation illustrates the relationship between gas concentration and output voltage.

The analog signal from MQ2 Gas sensor is further fed to LM393 High Precision Comparator (soldered on the bottom of the module), of course to digitize the signal. Along with the comparator is a little potentiometer you can turn to adjust the sensitivity of the sensor. You can use it to adjust the concentration of gas at which the sensor detects it.

4x4 Matrix Keypad

The 4x4 matrix keypad usually is used as input in a project. It has 16 keys in total, which means the same input values. The 4x4 matrix keypad module is a matrix non-encoded keypad consisting of 16 keys in parallel. The keys of each row and column are connected through the pins outside – pin R1-R4 as labelled beside to control the rows, while C1-C4, the columns.



Piezoelectric Buzzer

A buzzer or beeper is an audio signal generator type device, which may be mechanical, electromechanical or piezoelectric (in our case) depending upon the operating principle of the device. Buzzers are widely used in alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

In embedded systems, buzzer is widely used to notify various events such as interrupt generation, end of any process or as an alarming on emergency situation such as detection of fire, exceeding radiation in nuclear reactor plant etc

Interfacing devices with LPC2148

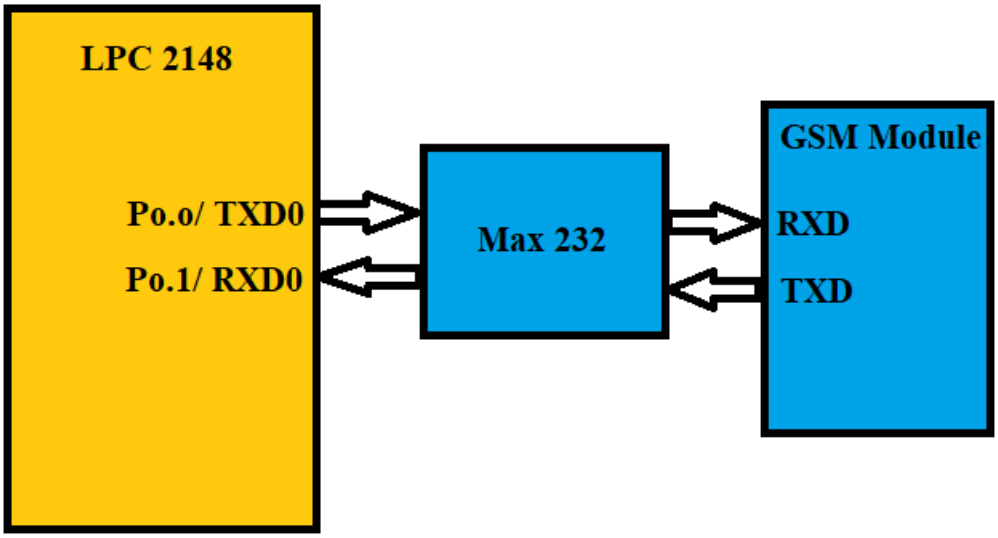
SIM900A GSM module

GSM modem interfaces with microcontroller for SMS control of industrial equipments. Sending an SMS through GSM modem when interfaced with microcontroller or PC is much simpler as compared to sending an SMS through UART.

Text messages may be sent through the modem by interfacing only three signals of the serial interface of modem with microcontroller i.e., TxD, RxD and GND.

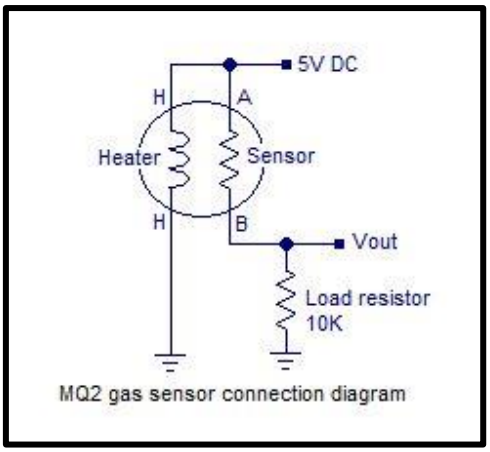
RTS and CTS signals of serial port interface of GSM Modem are connected with each other. The transmit and receive signals of the microcontroller serial port are connected to the corresponding signals of the serial interface of GSM module.

Max232 is used to interface the GSM Module with LPC 2148 to standardise the operating voltages across all the devices in UART Communication.



MQ2 Gas/Smoke Sensor

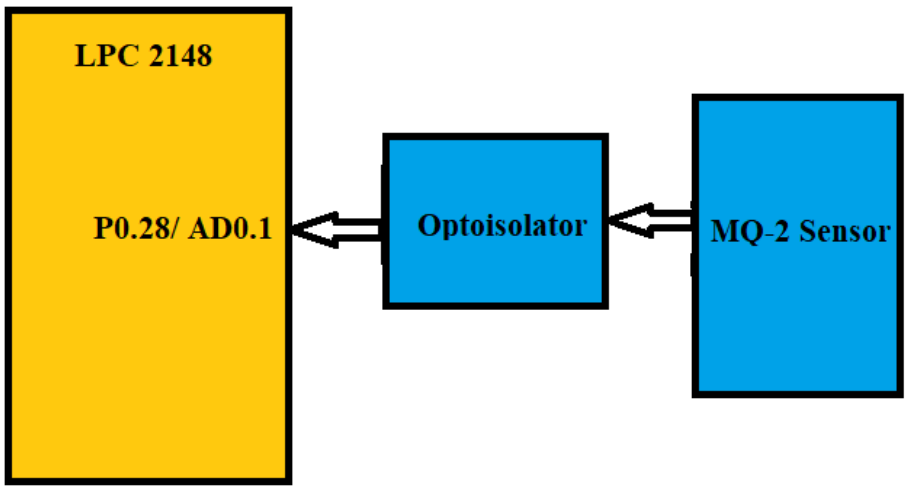
MQ2 sensor senses the flammable gases by the increase in temperature when they are oxidized by the heating element. Consider the figure given above. If there is any flammable gas present in the sample, the oxidization of the same gas results in increased temperature and the resistance of the sensor resistor will drop. That means more current will flow through the load resistor and so the voltage across it will shoot up.



MQ2 gas sensor connection diagram

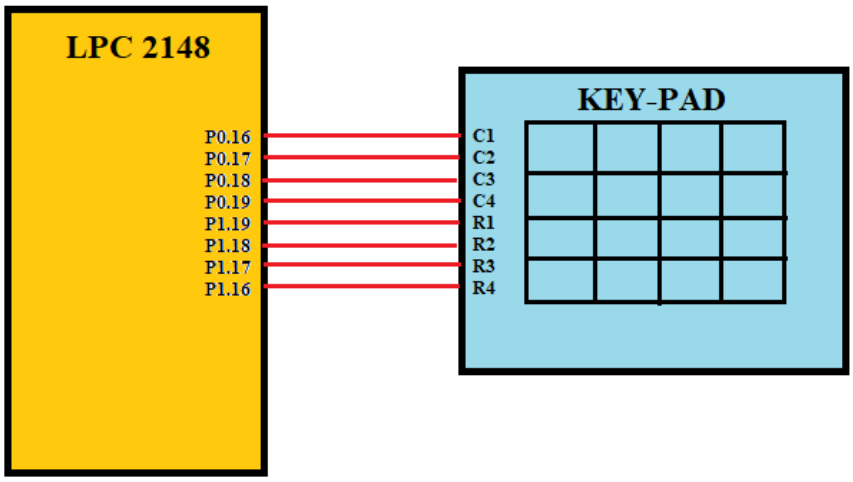
Under normal conditions (no LPG in the air), the sensor resistor will be very high around 850K. So the voltage drop V_{out} across the load resistor will be around zero. When the sensor is fully exposed to LPG, the sensor resistance drops to around 800 ohms and the voltage drop across the load resistance will be around 4.62 volts.

The digital output (DO) is connected directly to the interrupt 1 and analog output (AO) is connected to AD0.1



Keypad

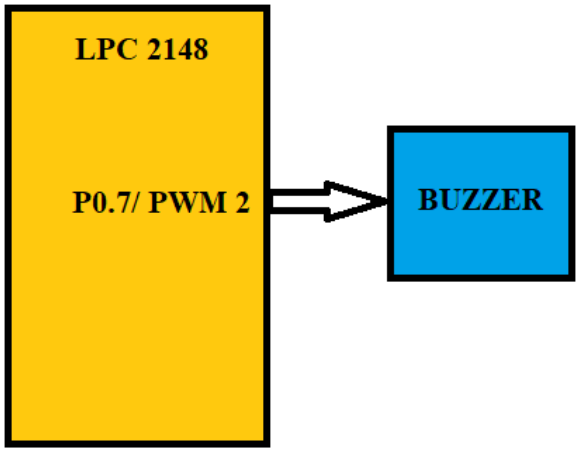
The row and column pins of the 4x4 matrix keypad are connected to the pins of LPC2148 as follows:



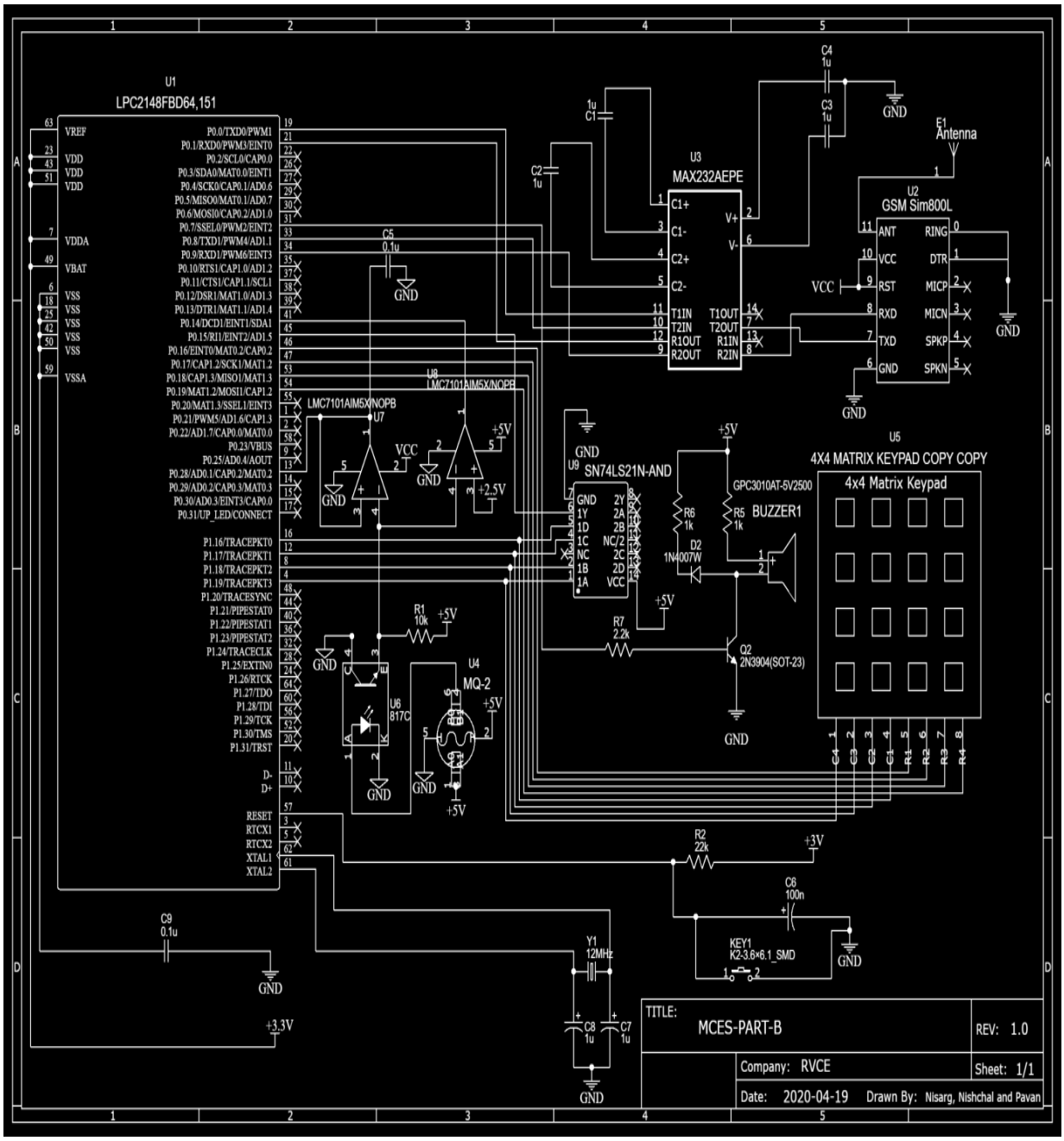
Buzzer

A small piezoelectric buzzer can be interfaced with LPC2148 by pulling pin P0.7 low, due to which the current will flow through the buzzer and a relatively sharp, single-tone frequency will be heard.

The alternative PWM feature of pin P0.7 (the PWM2 signal) can be used to modulate the buzzer to oscillate around different frequencies. Then the volume of the sound will be changed by alternating the pulse width. The buzzer can be disconnected by removing jumper JP1, and this is also the default position for this jumper since the buzzer sound can be quite annoying if always left on.



Circuit Diagram



Code

```
#include <LPC214x.H>
#include <string.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#define PLOCK 0x00000400
#define LED_OFF (IO0SET = 1U << 31)
#define LED_ON  (IO0CLR = 1U << 31)
#define COL0 (IO1PIN & 1 <<19)
#define COL1 (IO1PIN & 1 <<18)
#define COL2 (IO1PIN & 1 <<17)
#define COL3 (IO1PIN & 1 <<16)

//function_list

 //interrupts
void init_sensor_keyboard_interrupts(void);
__irq void Keyboard_ISR(void);
__irq void Sensor_ISR(void);

void delay(int count);
void callDoc_msgDoc();
void keyboard_check();
void GSM_init();
void UART0_init(void);
void UART0_TxChar(char ch);
void UART0_SendString(char* str);
void GSM_ReceiveMsg(void);
void GSM_Response(void);
void GSM_Calling(char *Mob_no);
void GSM_HangCall(void);
void GSM_Response_Display(void);
void callFire_msgFire();
void ring_bell(int dutycycle);

//Global_variables
char buff[160];                 /* buffer to store responses and messages */
bool status_flag = false;       /* for checking any new message */
volatile int buffer_pointer;
char Mobile_no[14];             /* store mobile no. of received message */
char message_received[60];              /* save received message */
int position = 0;         /* save location of current message */
```

```
char lookup_table[4][4][10]={
{"9029292201", "9029292102", "9023892201","9039292201"},
{"9029292201", "9029292201", "9029292251","9029292201"},
{"9029292101", "9029292201", "9029293201","9029292201"},
{"9029292401", "9339292201", "9999592201","9929292201"}};

int main (void)
{
  init_sensor_keyboard_interrupts() ;            // initialize the external interrupts
          GSM_init();
          //PINSEL2 |= 1 << 24;  // P0.28 AS AD0.1(01) for sensor
          //AD0CR= (1 << 1 | 1 << 21 | 1 << 24) ; //Sensor_init
          //PINSEL0 |= 2 << 14;   //select P0.7 as PWM2 (option 2)
  while (1)
  { /*
    AD0GDR|=(1<<31)|(0x1180); //0001 0001 1000 0000 = 70
                    while ( (AD0GDR & (unsigned long) 1 << 31) == 0){};
    int i = (AD0GDR >> 6 ) & 0xFF ;
    if (i > 60){
                     int duty=i/10;
      ring_bell(duty);
                    }
                    IO0CLR|=0x8000;       */
  }
}
__irq void Keyboard_ISR(void) // Interrupt Service Routine-ISR
{
 //keyboard
          keyboard_check();
  callDoc_msgDoc(Mobile_no);
  EXTINT |= 0x4;        //clear interrupt
  VICVectAddr = 0;     // End of interrupt execution
}
__irq void Sensor_ISR(void) // Interrupt Service Routine-ISR
{
 callFire_msgFire();
 ring_bell(100);
 EXTINT |= 0x2;        //clear interrupt
 VICVectAddr = 0;     // End of interrupt execution
}
void init_sensor_keyboard_interrupts()  //Initialize Interrupt
{
  EXTMODE = 0x6;         //Edge sensitive mode on EINT1 and EINT2
  EXTPOLAR = 0;       //Falling Edge Sensitive
  PINSEL0 = 0xA0000000;    //Select Pin function P0.14 and P0.15 as EINT1 and EINT2

  /* initialize the interrupt vector */
```

```
  VICIntSelect &= ~ (0x3<<15);          // EINT1 selected as IRQ 15&16
  VICVectAddr0 = (unsigned int)Sensor_ISR; // address of the ISR
  VICVectCntl0 = (1<<5) | 15;          //
  VICVectAddr5 = (unsigned int)Keyboard_ISR; // address of the ISR
  VICVectCntl5 = (1<<5) | 16;          //
  VICIntEnable = (0x3<<15);             // EINT1&2 interrupt enabled
  EXTINT = 0x6;
}
void GSM_init(){
        buffer_pointer = 0;
        bool is_msg_arrived;
        memset(message_received, 0, 60);
        UART0_init();
        UART0_SendString("GSM Initializing...");
        delay(300);
        while(1)
        {
                UART0_SendString("ATE0\r\n");                /* send ATE0 to check module is
ready or not */
                delay(5);
                GSM_ReceiveMsg();
                strcpy(buff,"OK\r\n");
                if(strstr(buff,"OK"))
                {
                        GSM_Response();                /* get Response */
                        memset(buff,0,160);
                        break;
                }
                else
                {
                        //UART0_SendString("Error");
                }
        }
        delay(1000);
        //UART0_SendString("Text Mode");
        UART0_SendString("AT+CMGF=1\r\n");       /* select message format as text */
        GSM_ReceiveMsg();
        strcpy(buff,"OK\r\n");
        GSM_Response();
        delay(10);
}
void UART0_init(void)
{
        PINSEL0 = PINSEL0 | 0x00000005;     /* Enable UART0 Rx0 and Tx0 pins of UART0 */
        U0LCR = 0x83; /* DLAB = 1, 1 stop bit, 8-bit character length */
        U0DLM = 0x00;          /* For baud rate of 9600 with Pclk = 15MHz */
        U0DLL = 0x61; /* We get these values of U0DLL and U0DLM from formula */
```

```
	U0LCR = 0x03; /* DLAB = 0 */
	U0IER = 0x00000001; /* Enable RDA interrupts */
	IO0DIR|=0xFF<<16;
}
void UART0_TxChar(char ch) /* A function to send a byte on UART0 */
{
	U0IER = 0x00000000; /* Disable RDA interrupts */
	U0THR = ch;
	while( (U0LSR & 0x40) == 0 ); /* Wait till THRE bit becomes 1 which tells that
transmission is completed */
	U0IER = 0x00000001; /* Enable RDA interrupts */
}
void UART0_SendString(char* str) /* A function to send string on UART0 */
{
	U0IER = 0x00000000; /* Disable RDA interrupts */
	uint8_t i = 0;
	while( str[i] != '\0' )
	{
		UART0_TxChar(str[i]);
		i++;
	}
	U0IER = 0x00000001; /* Enable RDA interrupts */
}
void GSM_ReceiveMsg(void){
	int i=0;
	for(i=0; i<2; i++){
	while((U0LSR & (0x01<<0))==0x00){};
                                        IO0CLR |= 0xFF << 16;
                                        IO0SET |= U0RBR << 16;
	}
}
void GSM_Response(void)
{
	unsigned int timeout=0;
	int CRLF_Found=0;
	char CRLF_buff[2];
	int Response_Length=0;
	while(1)
	{
		if(timeout>=60000)		/*if timeout occur then return */
		return;
		Response_Length = strlen(buff);
		if(Response_Length)
		{
			delay(1);
			timeout++;
			if(Response_Length==strlen(buff))
```

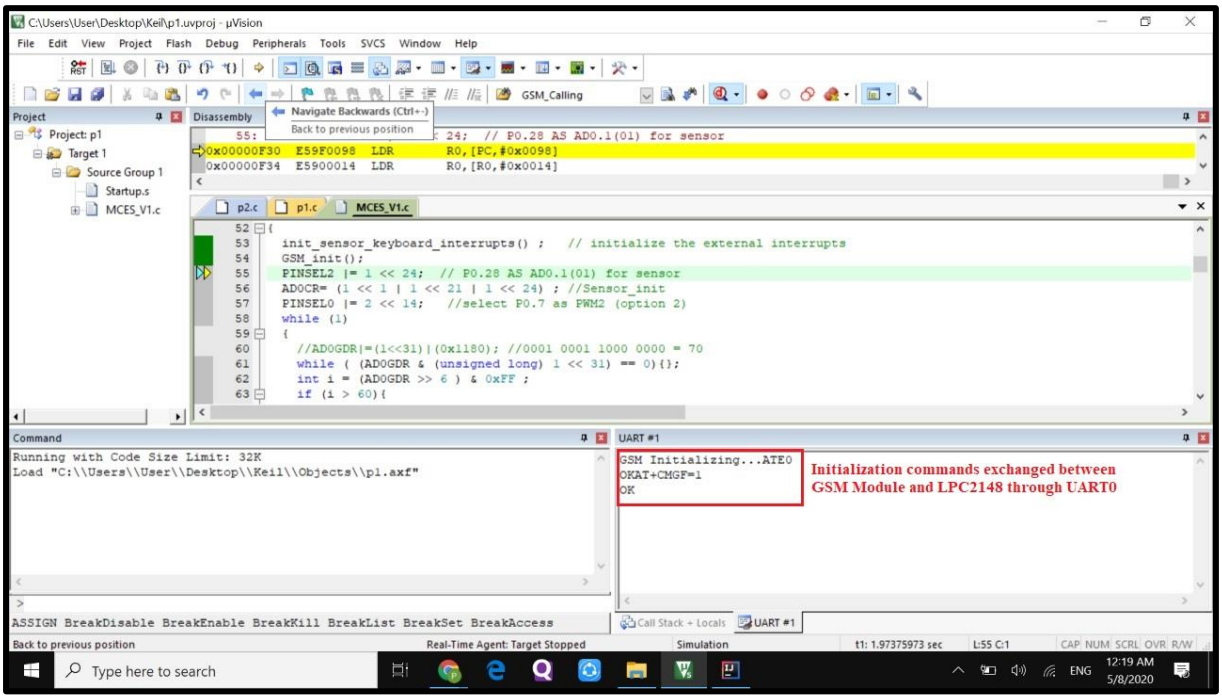
```
                        {
                                for(int i=0;i<Response_Length;i++)
                                {
                                        memmove(CRLF_buff,CRLF_buff+1,1);
                                        CRLF_buff[1]=buff[i];
                                        if(strncmp(CRLF_buff,"\r\n",2))
                                        {
                                                if(CRLF_Found++==2)          /* search for \r\n
in string */
                                                {
                                                        GSM_Response_Display();              /*
display response */
                                                        return;
                                                }
                                        }

                                }
                                CRLF_Found = 0;

                        }

                }
                delay(1);
                timeout++;
        }
        //status_flag = false;
}

void GSM_Response_Display(void)
{
        buffer_pointer = 0;
        while(1)
        {
                if(buff[buffer_pointer]== '\r' || buff[buffer_pointer]== '\n')          /* search
for \r\n in string */
                {
                        buffer_pointer++;
                }
                else
                        break;
        }


        while(buff[buffer_pointer]!='\r')               /* display response till "\r" */
        {
                UART0_TxChar(buff[buffer_pointer]);
```

```
                buffer_pointer++;
        }
        buffer_pointer=0;
        memset(buff,0,strlen(buff));
}

void GSM_Calling(char *Mob_no)
{
        char call[20];
        sprintf(call,"ATD%s;\r\n",Mob_no);
        UART0_SendString(call);                 /* send command ATD<Mobile_No>; for
calling*/
}

void GSM_HangCall(void)
{
        UART0_SendString("ATH\r\n");                /*send command ATH\r to hang call*/

}

void GSM_Send_Msg(char *num,char *sms)
{
        char sms_buffer[35];
        buffer_pointer=0;
        sprintf(sms_buffer,"AT+CMGS=\"%s\"\r\n",num);
        UART0_SendString(sms_buffer);               /*send command AT+CMGS="Mobile
No."\r */
        delay(200);
        while(1)
        {
                if(buff[buffer_pointer]==0x3e)              /* wait for '>' character*/
                {
                        buffer_pointer = 0;
                        memset(buff,0,strlen(buff));
                        UART0_SendString(sms);              /* send msg to given no. */
                        UART0_TxChar(0x1a);                         /* send Ctrl+Z then only
message will transmit*/
                        break;
                }
                buffer_pointer++;
                buff[buffer_pointer]='>';
        }
        delay(300);
        buffer_pointer = 0;
        memset(buff,0,strlen(buff));
        memset(sms_buffer,0,strlen(sms_buffer));
}
```

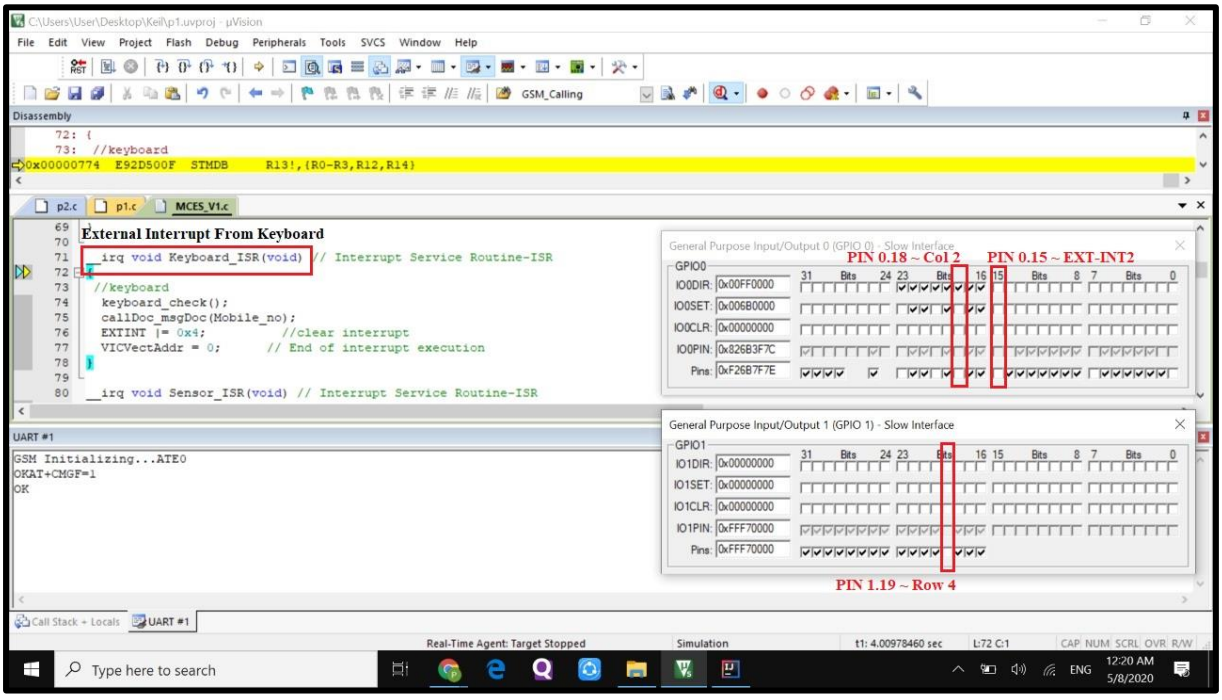
```
void keyboard_check(){

        int rowsel,colsel;
        IO0DIR |= 1U << 31 | 0x00FF0000; // to set P0.16 to P0.23 as o/ps

 while(1)
  {
//check for keypress in row0,make row0 '0',row1=row2=row3='1'
        rowsel=0;IO0SET |= 0X000F0000;IO0CLR |= 1 << 16;
        if(COL0==0){colsel=0;break;};if(COL1==0){colsel=1;break;};
        if(COL2==0){colsel=2;break;};if(COL3==0){colsel=3;break;};
//check for keypress in row1,make row1 '0'
        rowsel=1;IO0SET |= 0X000F0000;IO0CLR |= 1 << 17;
        if(COL0==0){colsel=0;break;};if(COL1==0){colsel=1;break;};
        if(COL2==0){colsel=2;break;};if(COL3==0){colsel=3;break;};
//check for keypress in row2,make row2 '0'
        rowsel=2;IO0SET |= 0X000F0000;IO0CLR |= 1 << 18;//make row2 '0'
        if(COL0==0){colsel=0;break;};if(COL1==0){colsel=1;break;};
        if(COL2==0){colsel=2;break;};if(COL3==0){colsel=3;break;};
//check for keypress in row3,make row3 '0'
        rowsel=3;IO0SET |= 0X000F0000;IO0CLR |= 1 << 19;//make row3 '0'
        if(COL0==0){colsel=0;break;};if(COL1==0){colsel=1;break;};
        if(COL2==0){colsel=2;break;};if(COL3==0){colsel=3;break;};
 }
        delay(50);  //allow for key debouncing
        while(COL0==0 || COL1==0 || COL2==0 || COL3==0);//wait for key release
        delay(50);   //allow for key debouncing
        IO0SET |= 0X000F0000; //disable all the rows
        memset(buff,0,strlen(Mobile_no));
        strcpy(Mobile_no,lookup_table[rowsel][colsel]);
}

void delay(int count)
{
  int j=0,i=0;
  for(j=0;j<count;j++)
  {
    /* At 60Mhz, the below loop introduces
    delay of 10 us */
    for(i=0;i<35;i++);
  }
}

void callDoc_msgDoc(){
        GSM_Calling(Mobile_no);
                /* call sender of message */
```

```
        UART0_SendString("Calling...");
        UART0_SendString(Mobile_no);
        delay(1500);
        GSM_HangCall();
                /* hang call */
        //UART0_SendString("Hang Call");
        delay(1500);
        GSM_Send_Msg("Come To ICU, Immediately",Mobile_no); //send message
  delay(100);
}

void callFire_msgFire(){
        GSM_Calling("101");
        /* call sender of message */
        UART0_SendString("Calling Fire Station ");
        UART0_SendString("101");
        delay(1500);
        GSM_HangCall();
                /* hang call */
        //UART0_SendString("Hang Call");
        delay(1500);
        GSM_Send_Msg("Fire Alert at XYZ Hospital, Start Immediatly ","101"); //send
message
  delay(100);
}



void ring_bell(int dutycycle){
   PWMPCR = (1 << 10);  // enable PWM2 channel
   PWMMR0 = 1000;   // set PULSE rate to value suitable for Bell operation
   PWMMR2 = (1000U*dutycycle)/100;  // set PULSE period
   PWMTCR = 0x00000009;  // bit D3 = 1 (enable  PWM), bit D0=1 (start the timer)
   PWMLER = 0X04;  // load the new values to PWMMR0 and PWMMR2 registers

}
```

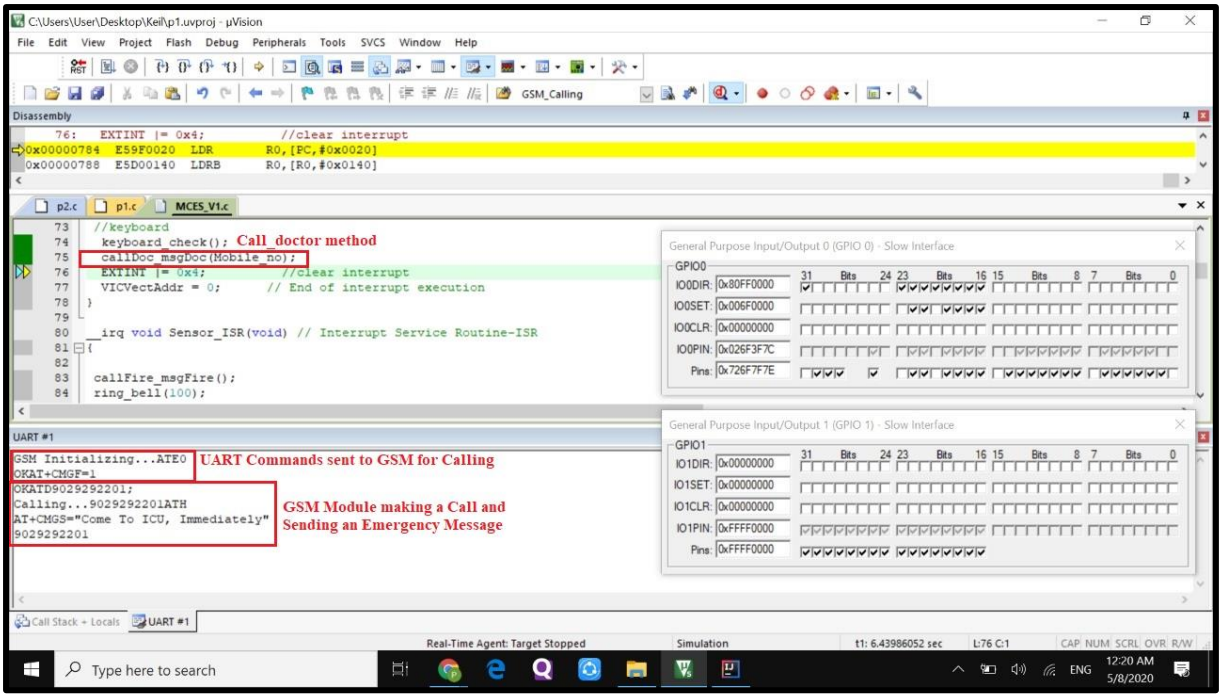
Output



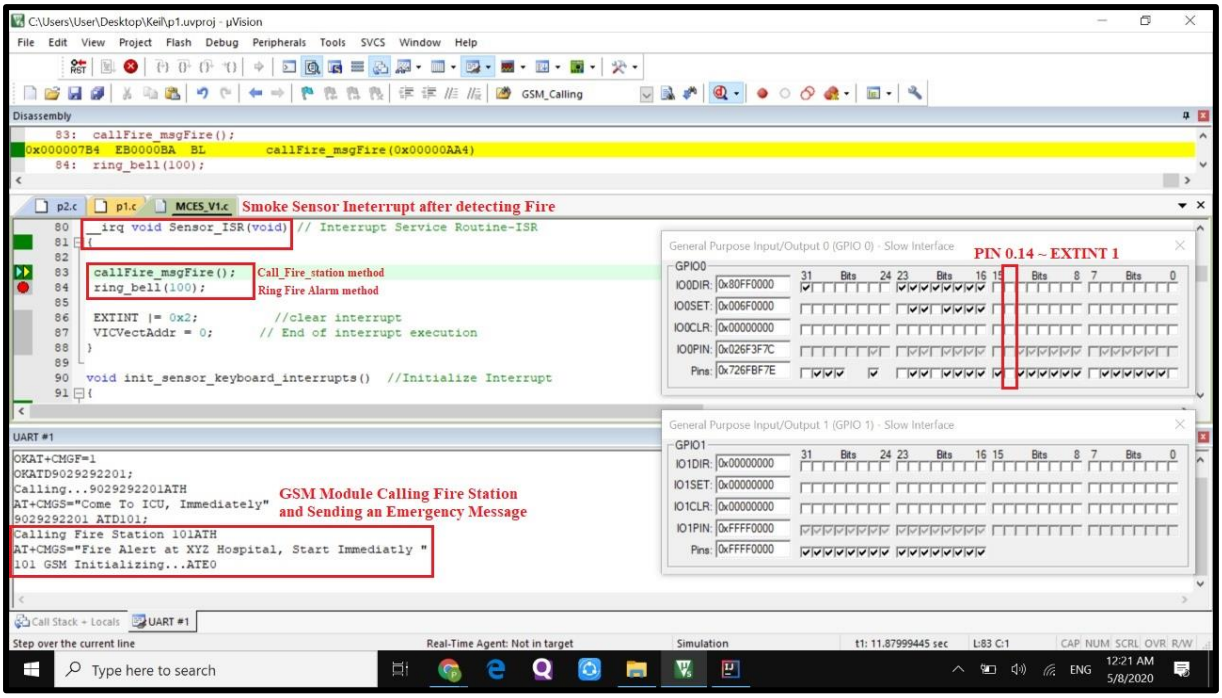
Initialization commands between GSM and LPC2148 using UART



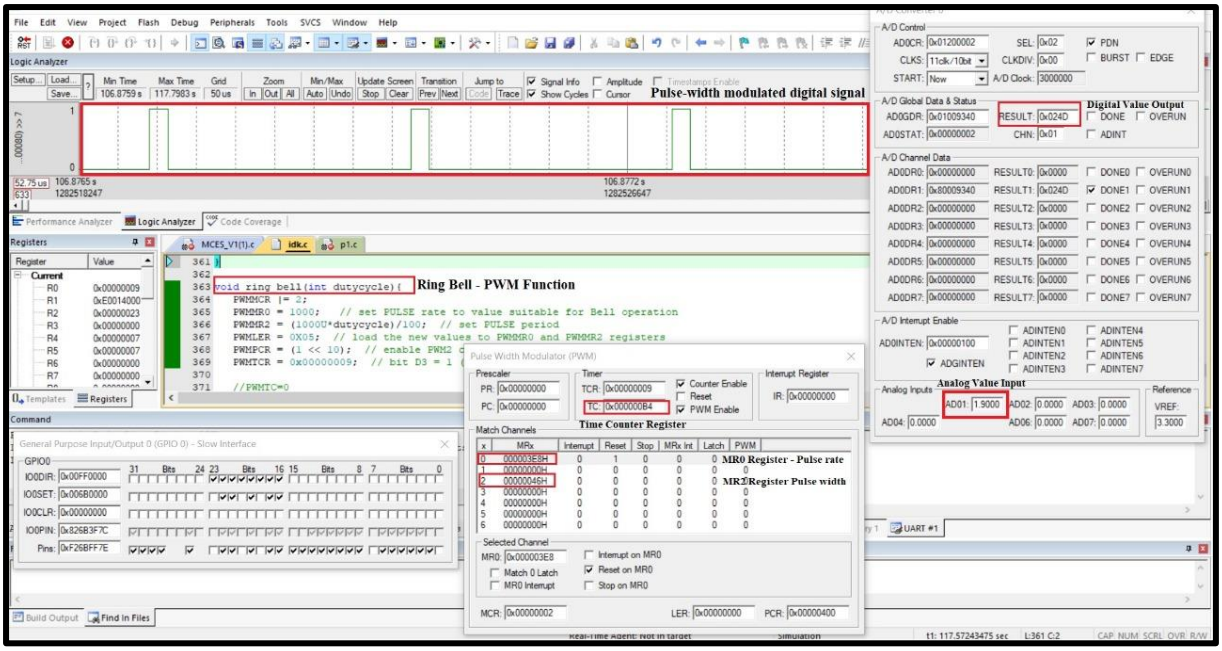
Keyboard interrupt generated when a key press is detected. The key pressed corresponds to the doctor's phone number that has to be alerted.



Transfer of alert message to the concerned number



Sending a message to the fire station in case of emergency



The PWM waveform of the fire siren generated using ADC to alert the occupants of the building

Conclusion

The testing of Instant Tele Emergency Alarm Module - iTEAM was successful. A key press could successfully send an alert message to the corresponding health official's phone number hardcoded in the device. A fire alert message was successfully sent in case of emergency to the fire station. A siren was also activated to alert the occupants in the building.

Thus, the objectives of this project were successfully accomplished. This provides an efficient, feasible option for the safety of the health workers while not compromising on the health services to the patients.