DEPARTMENT OF TECHNICAL EDUCATION

GOVERNMENT POLYTECHNIC FOR WOMEN

Department of Electronics and Communication Engineering

CERTIFICATE

Certified that this project report entitled E-CLASSROOM MANAGEMENT which is being submitted by kumari____________________________ bearing Reg. No.____________for partial fulfillment for the award of diploma in Electronics and Communication Engineering during the year 2019-20. Suggestions indicated for internal assessment have been incorporated in the report and one copy of it being deposited in the Polytechnic library. The project report has been approved as it is satisfied the academic requirements in respect of project work prescribed for the said diploma.

It is further understood that by this certificate the under signed do not endorse or approve any statement made, opinion expressed or conclusion drawn the rein but approve the project only for the purpose for which it is submitted.

Guide(s) Name and Signature

Examiners

1

2

Head of Department

CANDIDATE’S DECLARATION

I, **____________________________** a student of Diploma **in Electronics and Communication Engineering Department** bearing Reg. No. **______________**of Government Polytechnic for Women, Bangalore, hereby declare that I own full responsibility for the information, results and conclusions provided in this project work titled **E-CLASS ROOM MANAGEMENT** submitted to **State Board of Technical Examinations, Government of Karnataka**, for the award of Diploma in **2019-20**. To the best of my knowledge, this project work has not been submitted in par tor full elsewhere in any other institution/organizations for the award of any certificate/diploma/degree. I have completely taken care in acknowledging the contribution of others in this academic work. I further declare that incase of any violation of intellectual property rights and particulars declared, found at any stage, I as the candidate will be solely responsible for the same.

Candidate Signature

ABSTRACT

The project entitled E-CLASS ROOM MANAGEMENT SYSTEM is a microcontroller-based electronic system designed and developed to manage the important classroom activities/status such as availability of class room, completion of class, arrival of teacher to the class room, call to class representative, response from the class representative, interaction between classroom and staffroom, monitoring the status of the class room at staff room.

Normally, CCTV is prevalent in monitoring the classroom activities. Although it provides many more details of the classroom such as video and audio of the classroom activities, it has several disadvantages too, such as privacy risk, cost, lack of interaction between classroom and staffroom. CCTV does not allow indication of class-end and interaction. Further, because of cost factor, many educational institutes do not employ CCTVS for classroom monitoring. E-Classroom management is designed and developed to overcome these difficulties.

E-Classroom management is designed and developed at low cost employing microcontrollers and relevant minimal hardware. It uses LEDs and LCD to indicate the status and communication alerts, simple pushbuttons to input controlling signals, and microcontrollers to process the controlling information, data and communication. The basic idea is to have intercommunication between the staffroom and two classrooms in pilot phase, and the design has potential for extension for multiple class rooms.

This project is designed and developed using MCS-51 microcontrollers and embedded C as programming language, thus illustrating processor-based design skill, and tested using Multisim simulation tool and PCB design using Multisim. By doing this project, several skills such as programming, simulation, PCB design, hardware design and report preparation skills are acquired.

ACKNOWLEDGEMENTS

We take this opportunity to express our profound and sincere gratitude to the Department of Electronics and Communication, Government Polytechnic for women, Bangalore. We salute our beloved and highly esteemed institution for growing us in to true Electronics and Communication diploma holder.

We would like to express our thanks to our principal Smt. SALMA SHAHEEN, for her support. We would also express our gratitude to our HOD Sri R.SURESH BABU.

We would also like to express our thanks to guide Sri S. M. PRASAD, Dept. of E&C for his guidance and support.

Our sincere thanks also go for teaching and non-teaching staff of Electronics & Communication department.

CONTENTS

1. Introduction

1.1 Introduction

1.2 Organization of the report

2. System Block Diagram

2.1 Block Diagram

2.2 Description

3. Hardware Description

3.1 LCDdisplay

3.2 Buzzer

3.3 LED

3.4 Push button switches

3.5 AT89C51Microcontroller

3.6 Multisim

4. Software and Hardware Design

4.1 Flowchart

4.2 Program Code

4.3 System Circuit

4.4 Circuit 3-D model

5. Operation Instructions, Future direction and Conclusion

5.1 Operating Instructions

5.2 Conclusion and Future Direction

5.3 Reference

Chapter-1 INTRODUCTION

1.1 Introduction

There are several places such as Colleges/Schools/Institutions that contain class rooms and teacher/staff rooms normally physically separated/away. Teachers have the tendency to reach/leave class rooms late or early. This might lead to students' scattering out of the class room or it might lead other teachers to wait. Further, there are circumstances where some kind of interaction needs to be established between class room and staff room such as class representative is required to reach staff room or responding to teachers calls. Presently, Close Circuit Television (CCTV) is employed in the class rooms to observe the activities in class room, and through them some of the mentioned problems are resolved. However, CCTVs are expensive and not many schools can afford; CCTVs do not support class room and staff room interaction; CCTVs intrude on privacy; and CCTVs are normally of star-topology having only one control.

The main use of this system is that the time can be saved. By less co stand simple components this system can be achieved.

This project is an attempt to overcome the said difficulties. It is designed and developed using low cost hardware such as microcontrollers, LEDs, LCD, buzzer and Push-button keys which make the system cheap. With the design and development of this real-world project, several technical skills such as hardware design, software development, testing under simulated environment, PCB design, and report preparation were acquired.

1.2 Organization of Report

The rest of the report is organized as follows. Chapter-2 provides block diagram and its description. Chapter-3 contains detailed hardware description. Chapter-4 contains software design and description of the project. It also contains program code and flowchart of the system. Chapter-5 gives operating instruction and conclusion of the project.

2. SYSTEM BLOCK DIAGRAM

2.1 Block Diagram

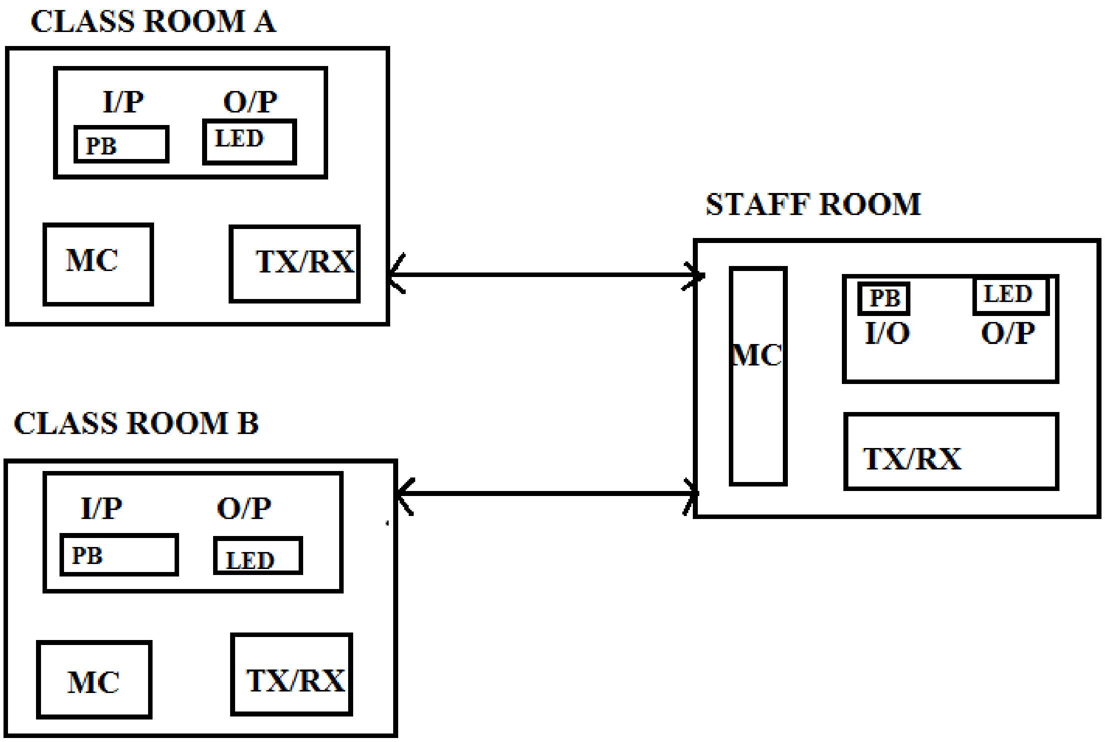


Figure 2.1: E-Classroom Management Block Diagram

2.1 Description

The proposed system is a pilot phase and designed for two class rooms (designated as Room-B and Room-C) and a staff/faculty room (Room-A). Staff room has a microcontroller, status indicating LEDs and LCD and controlling push-buttons. Similarly, each class room has a microcontroller, status LEDs and controlling push-buttons, as shown in Fig. 2.1.

Communication between Room-A and Room-B or Room-C is through TX/RX lines of microcontrollers, control signals using push-buttons through Interrupts and Response/Status is through LEDs. LCD is also used at staff room to indicate the status of class rooms.

Chapter-3 HARDWARE DESCRIPTION

3.1 LCD Display

Liquid Crystal Display which is commonly Known as LCD is an Alphanumeric Display it means that it can display Alphabets, numbers as well as special symbols thus LCD is user friendly Display device which can be used for displaying various messages unlike seven segment display only numbers and some of the alphabets .The only disadvantage of the robust display is visualized from a longer distance as compared to LCD. Here we have used 16*2 alphanumeric displays which mean on this display we can display two lines with maximum of 16 characters in one line. This LCD has two registers, namely, command and Data.

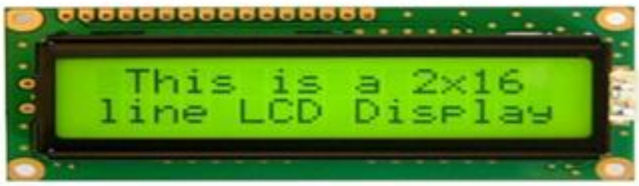


Fig. 3.1: LCD

The command register stores the command instructions given to the LCD.A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD .The data is the ASCII value of the character to be displayed on the LCD.

3.2 Buzzer

A buzzer or beeper is an audio signaling device which may be mechanical, electromechanical, or piezoelectric (piezo for short) .we have used magnetic buzzer.

Fig. 3.2: Buzzer

Magnetic buzzer specifications:

- Current Rating- 30mA
- Frequency- 2.3 kHz
- Voltage Rating-5VDC
- Size / Dimension Circular - 12mm Dia x 9.5mm H
- Mounting Type- through Hole
- Voltage Range -3 ~ 7VDC

3.3 LED

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable current is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. LEDs are typically small (less than 1 mm2) and integrated optical components may be used to shape the radiation pattern.

Fig. 3.3: LED

3.4 Push-button Switches

A **push-button** (also spelled **pushbutton**) or simply **button** is a simple switch mechanism to control some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. These buttons are used to generate control signals.

Fig. 3.4: PB Switches

3.5 MCS-51 Microcontroller AT89C51

The AT89C51 is an old 8 bit microcontroller from the Atmel family. It works with the popular 8051 architecture and hence is used by most beginners till date. It is a 40 pin IC package 4kb flash memory. It has four ports and all together provides 32 programmable GPIO pins. It does not have in built ADC module and supports only USART communication although it can be interfaced with external ADC IC like the ADC084 or the ADC0808.The AT89C51 is no longer in production and Atmel does not support.

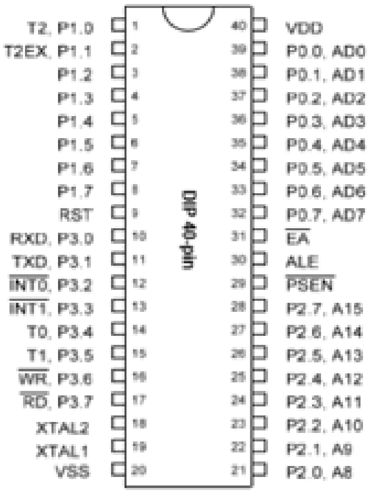


Fig. 3.5: MCS-51 Microcontroller PIN diagrams

1. 80c51 Central Processing unit
2. 5Voperating voltagefrom0Mhz to 40Mhz
3. 64kBofon-chipFlashusercodememorywithISP(In-SystemProgramming)andIAP(In-Application Programming)
4. 256 bytes of on-chip scratchpad RAM
5. 16K/8K/4Kbyteselectrically erasable/programmable Flash EPROM
6. 64KB program memory address space

7. 64KB data memory address space
8. Four 8-bitbi-directionalports
9. 8-sources, 4-levelinterruptcapability
10. Three16-bit timer/counters
11. One full duplex serial port

3.6 Multisim Software

NI Multisim is an electronic schematic capture and simulation program which is part of a suite of circuit design programs, along with NI Ultiboard. Multisim is one of the few circuit design programs to employ the original Berkeley SPICE based software simulation. Multisim was originally created by a company named Electronics Workbench, which is now a division of National Instruments. Multisim includes microcontroller simulation (formerly known as MultiMCU), as well as integrated import and export features to the printed circuit board layout software in the suite, NI Ultiboard**.**

Multisim Software provides SPICE simulation, analysis, and printed circuit board (PCB) tools to help you quickly iterate through designs and improve prototype performance. Move from schematic to layout seamlessly to save time and reduce prototype iterations.

Multisim was originally called **Electronics Workbench** and created by a company called Interactive Image Technologies. At the time it was mainly used as an educational tool to teach electronics technician and electronics engineering programs in colleges and universities. National Instruments has maintained this educational legacy, with a specific version of Multisim with features developed for teaching electronics.

Chapter-4 SOFTWARE AND HARDWARE DESIGN

4.1 Flow Chart

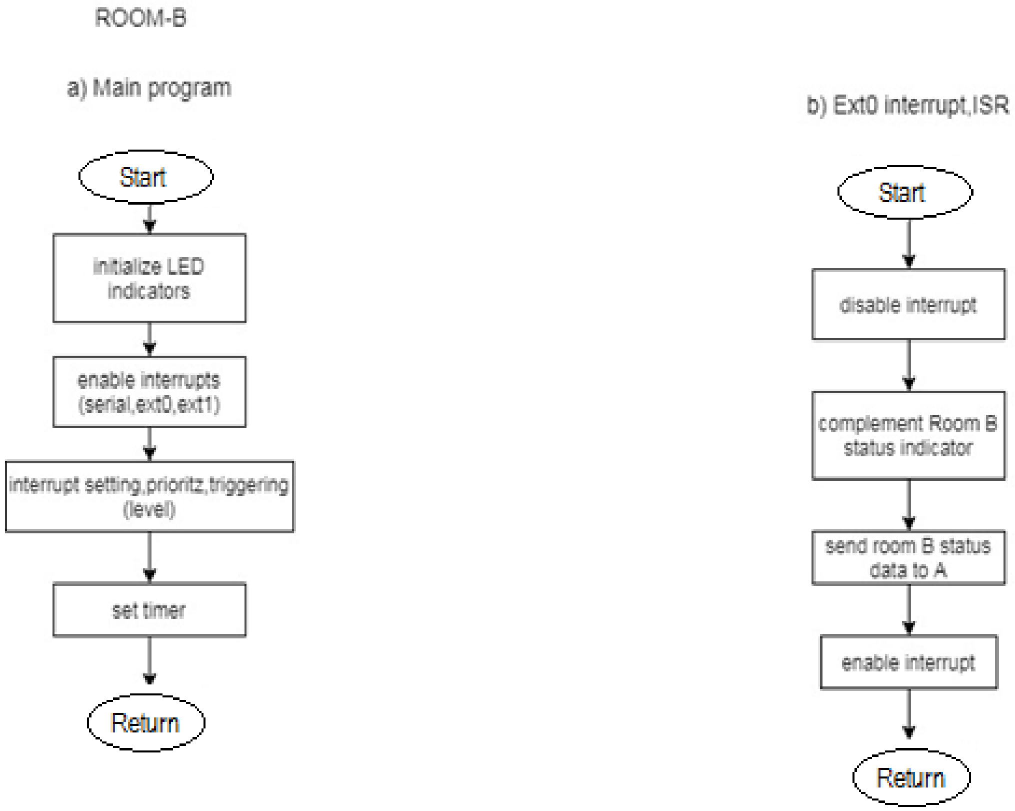
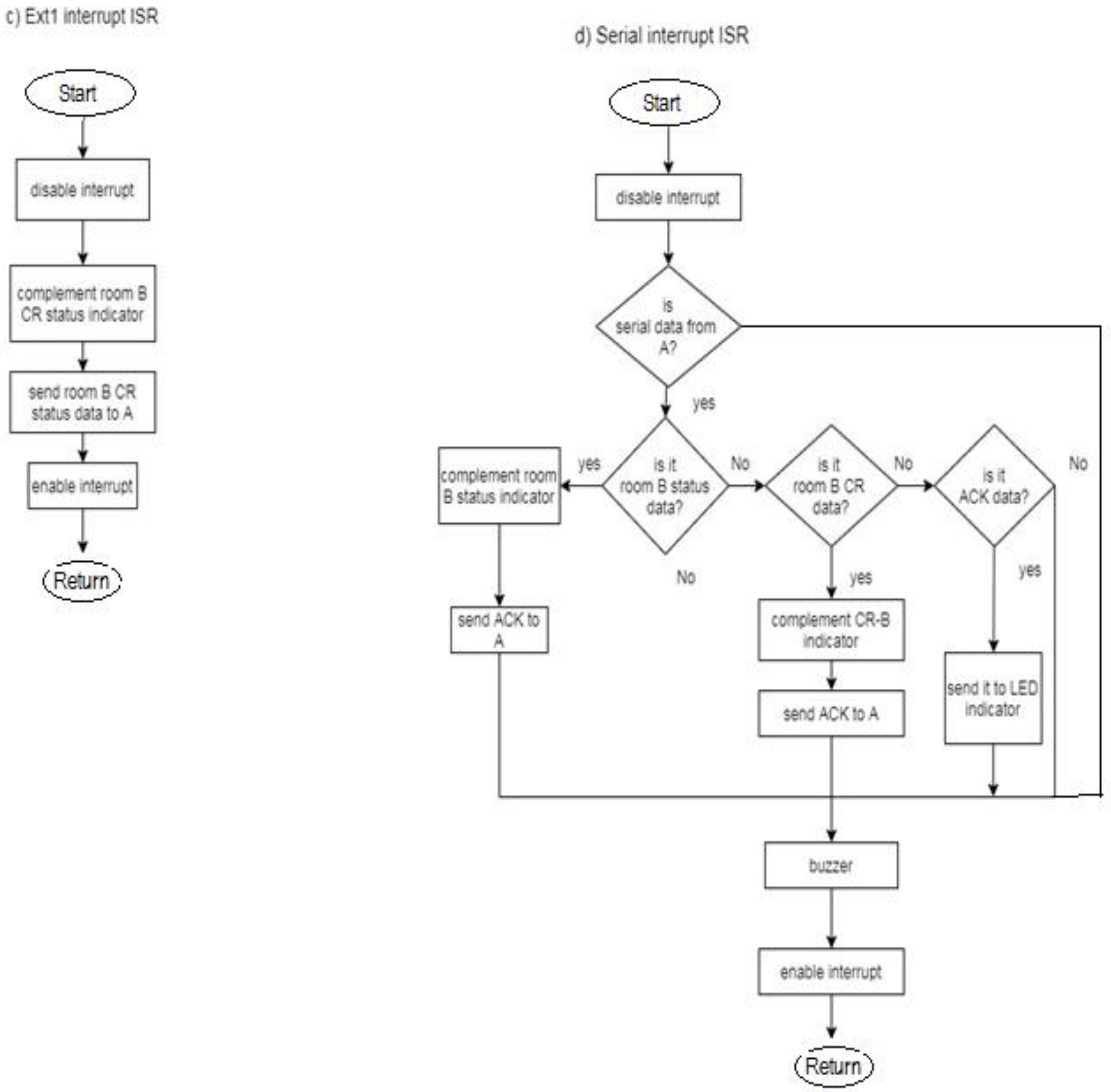

ROOM-B
a) Main program
Start
initialize LED indicators
enable interrupts (serial,ext0,ext1)
interrupt setting,prioritz,triggering (level)
set timer
Return
b) Ext0 interrupt,ISR
Start
disable interrupt
complement Room B status indicator
send room B status data to A
enable interrupt
Return

Fig. 4.1: Flow charts for Room B/C



c) Ext1 interrupt ISR

d) Serial interrupt ISR

Fig. 4.2: Flow charts for Room-A

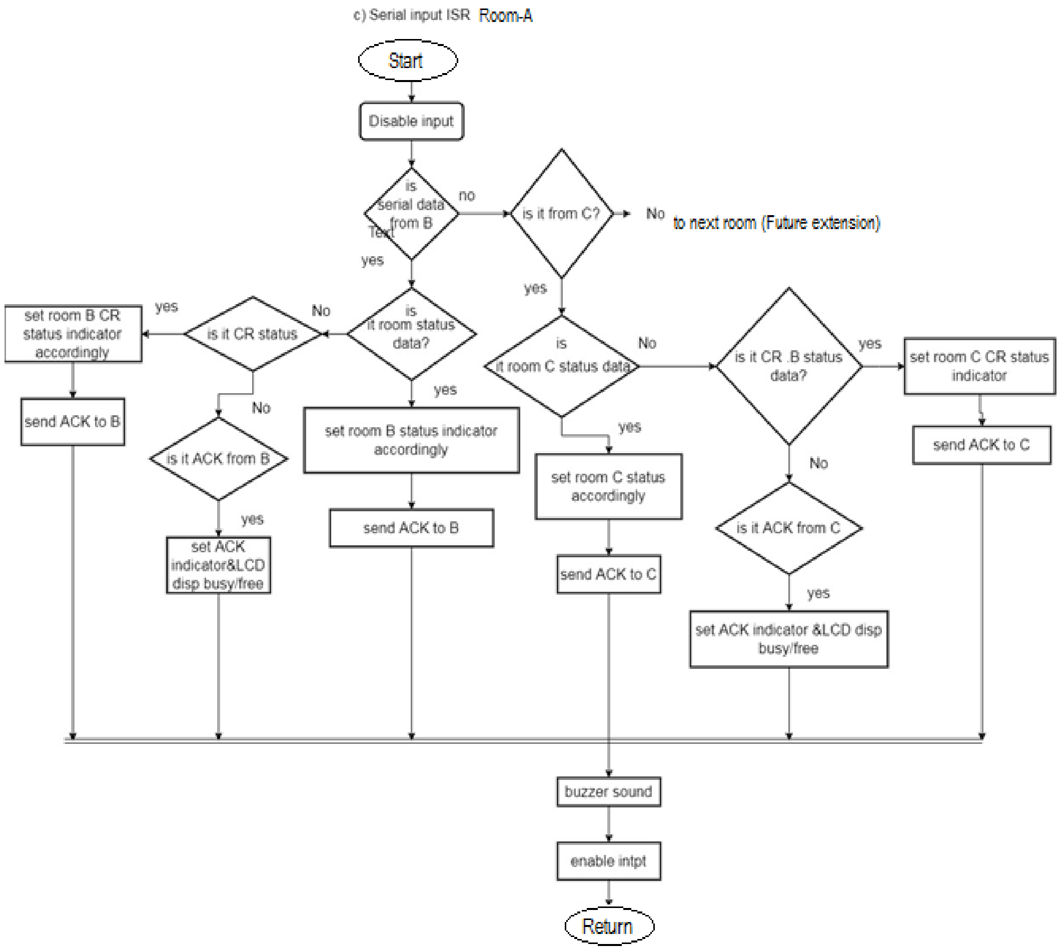


Fig. 4.3: Flow charts for Room-A (continued)

4.2 Program Code

```
//Program to manage Class Room B
#include<htc.h>
#define LEDC0Cls P10//Classroom status
#define LEDC1Crc P11//CR call status
#define LEDC2Ack P12//Acknowledgement indicator
#define Buzz P13//Buzzer line

//const char a[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
//void RAM_VECTOR(0x73, t2int_handler);

//Macro to create ISR for serial-port interrupt
#asm
        GLOBAL _serialisr// ISR name = serialisr()
        PSECT vectors,ovrld
        ORG 0x23                        // Int vector
```

```
            LJMP _serialisr                 // ISR function name again with a preceding underscore added.
            PSECT text
#endasm

//Macro to create ISR for Ext. Interrupt 0
#asm
            GLOBAL _ext0isr                 //ISR name = ext1isr()
            PSECT vectors,ovrld
            ORG 0x03                                    //Int vector
            LJMP _ext0isr                   //ISR function name again with a preceding underscore added.
            PSECT text
#endasm

#asm
            GLOBAL _ext1isr// ISR name = ext1isr()
            PSECT vectors,ovrld
            ORG 0x13// Int vector
            LJMP _ext1isr// ISR function name again with a preceding underscore added.
            PSECT text
#endasm

//Functions
#pragma interrupt level 0
void reentrant delay(unsigned int msec)    //Function to provide time delay in msec for LCD.
{           unsigned int i,j ;
            for (i=0;i<msec; i++)
            for (j=0; j<100;j++);
}

//Buzzer function
#pragma interrupt_level 0
void reentrant buzzer(unsigned char Length, unsigned int msec)    //Function to provide time delay in msec for LCD.
{           unsigned int i;
   Buzz=1;
for (i=0;i<Length;i++)
             {
     delay(msec);
     Buzz=!Buzz;
    }
}

//ISR for serial-port interrupt
interrupt void serialisr(void)
{unsigned int i,j,Outer=50, Inner=50;

 ES=0;                  //Disable serial-port interrupt
 RI=0;                  // Clear intpt flag
 ACC=SBUF;              // Reading from the port buffer

if(ACC5==0&&ACC4==1)        //is it for B?
if(ACC7==0&&ACC6==0)      //is it from A?
     {if(ACC3==0&&ACC2==0)   //Room status data?
       {LEDC0Cls=!LEDC0Cls;
```

```
         ACC7=0; ACC6=1;     //from B
         ACC5=0;ACC4=0;     // to A
         ACC3=1;ACC2=0;     //Setting Ack
         SBUF=ACC;                    //Preparing to send Ack to A
         while(TI==0);      //Sending Ack to A
         TI=0;
        }                  //Sent ack to A
if(ACC3==0&&ACC2==1)   //CR status data?
        {LEDC1Crc=!LEDC1Crc;
         ACC7=0;ACC6=1;     // from B
         ACC5=0;ACC4=0;     // to A
         ACC3=1;ACC2=0;     //Setting Ack
         ACC0=LEDC1Crc;
         SBUF=ACC;                    //Preparing to send Ack to A
         while(TI==0);      //Sending Ack to A
         TI=0;
        }

if(ACC3==1&&ACC2==0)   //Ack data?
        {LEDC2Ack=1;
         delay(5);
         LEDC2Ack=0;
                     buzzer(3, 2);
        }
      }
  ES=1;                                                            //Enable serial-port interrupt
 }

//ISR for Ext. Interrupt-0
interrupt void ext0isr(void)
{EX0=0; //Disable ext intpt-0
 LEDC0Cls=!LEDC0Cls; //Complementing class-room status LED
 ACC=0x00;
 ACC7=0;ACC6=1;  //from B
 ACC5=0;ACC4=0;  //to A
 ACC3=0;ACC2=0;  //Room status data
 ACC0=LEDC0Cls;  //Current room-status data in room C
 SBUF=ACC;
 while(TI==0);
 TI=0;
 EX0=1;                          //Enable external interrupt0
}

//ISR for Ext. Interrupt-1
interrupt void ext1isr(void)
{EX1=0; //Disable ext intpt-0
 LEDC1Crc=!LEDC1Crc; //Complementing class-room status LED
 ACC=0x00;
 ACC7=0;ACC6=1;  //from B
 ACC5=0;ACC4=0;  //to A
 ACC3=0;ACC2=1;  //CR status data
 ACC0=LEDC1Crc;  //Current room-status data in room C
 SBUF=ACC;
 while(TI==0);
 TI=0;
 EX1=1;                              //Enable external interrupt0
}

//main program
void main()
{  unsigned int i;
   LEDC0Cls=1;
   LEDC2Ack=1;
   LEDC1Crc=1;
for (i=1;i<4;i++)  //led blinking
   {LEDC0Cls=!LEDC0Cls;
    LEDC2Ack=!LEDC2Ack;
    LEDC1Crc=!LEDC1Crc;
    delay(20);
   }
   LEDC0Cls=0; //led initial setting
   LEDC2Ack=0;
   LEDC1Crc=0;
   EA=1; //interrupt settings
   ES=1;
   EX0=1;IT0=1;
   EX1=1;
   TMOD=0x20; //timer settiing
   TH1=0xFD;
   SCON=0x90;
   TR1=1;
while(1);}
```

// Room A Programs

```
#include<htc.h>
#include <string.h>
#define LED0BCls P10        //Room B status
#define LED1CCls P11        //Room C status
#define LED2BAck P12        //Room B ack
#define LED3CAck P13        //Room C ack
#define LED4BCrc P14        //CR B status
#define LED5CCrc P34        //CR C status
#define Buzz P36        // Buzzer
#define dataport P2     //LCD 8-bit data line
#define rs P06
#define en P05
#define rw P07

//ISR macros
//Macro to create ISR for ext interrupt 0
#asm
            GLOBAL _ext0isr                // ISR name = ext0isr()
            PSECT vectors,ovrld
            ORG 0x03                                   // Int vector
            LJMP _ext0isr                  // ISR function name again with a preceding underscore added.
            PSECT text
#endasm
```

```
//Macro to create ISR for serial interrupt
#asm
            GLOBAL _serialisr//ISR name = serialisr()
            PSECT vectors,ovrld
            ORG 0x23                                   //Int vector
            LJMP _serialisr                //ISR function name again with a preceding underscore added.
            PSECT text
#endasm

//Functions

//Delay function
#pragma interrupt_level 0
void reentrant delay(unsigned int msec)    //Function to provide time delay in msec for LCD.
{           unsigned int i,j ;
            for(i=0;i<msec;i++)
            for(j=0;j<10;j++);
}

//Buzzer function
#pragma interrupt_level 0
void reentrant buzzer(unsigned char Length, unsigned int msec)    //Function to provide time delay in msec for LCD.
{           unsigned int i;
   Buzz=1;
for (i=0;i<Length;i++)
             {
     delay(msec);
     Buzz=!Buzz;
    }
}

//LCD commands write function
#pragma interrupt_level 0
void reentrant lcdcmd(unsigned char item)    //Function to send command to LCD
{           dataport = item;
            rs=0;
   rw=0;
            en=1;
            delay(1);
            en=0;
   rw=0;
}

// LCD data writing function
#pragma interrupt_level 0
void reentrant lcddata(unsigned char item)    //Function to send data to LCD
{           dataport=item;
            rs=1;
   rw=0;
            en=1;
            delay(1);
            en=0;
```

```
  rw=0;
}

//LCD Initialization function
//#pragma interrupt_level 1
void reentrant lcdini2(void)
{          lcdcmd(0x38);   //for using 8-bit 2 row mode of LCD
           delay(100);
           lcdcmd(0x0C);   //turn display ON and cursor OFF
           delay(100);
           lcdcmd(0x01);   //clear screen
           delay(100);
           lcdcmd(0x80);   //bring cursor to position 0 of line 1
           delay(100);
}

//LCD scrolling display function
//#pragma interrupt_level 1
void disp_scrol_name(char *item)
 {unsigned int lcdloc=0x80,i=0,j,k, lcdwidth=16;
  j=strlen(item);

           for(k=0;k<lcdwidth;k++) // scrolling
   {
    lcdcmd(lcdloc);
    while(i<=j)
    {if ((k+i)<lcdwidth) // cutting name tail
     lcddata(item[i]);

i++;
    }
    lcdloc++;
    i=0;
lcdcmd(0x01);// clearing display for every iteration
   }
  }

// LCD Normal display
#pragma interrupt_level 0
void reentrant displcd(unsigned int lcdloc, char *item)
 {unsigned int i=0,j,k, lcdwidth=16;//lcdloc=0x80,
  j=strlen(item);
  //lcdini2();
    // lcdcmd(0x01);
    lcdcmd(lcdloc);
    while(i<=j)
    {  lcddata(item[i]);
i++;
    }

 }
```

```
//Ext Intpt 0
#pragma interrupt_level 1
interrupt void ext0isr(void)
{ EX0=0;                //disable interrupt
  ACC=0x00;

  if (P15==0)           //if status B switch
  {LED0BCls=!LED0BCls;   //toggle Room B status
   ACC7=0;ACC6=0;        //from A
   ACC5=0;ACC4=1;                        //to B
   ACC3=0;ACC2=0;        //Room status data
   ACC0=LED0BCls;}       //ACC0 current status of room B at room A

  if (P16==0)           //if status C switch
  {LED1CCls=!LED1CCls;   //toggle Room C status
   ACC7=0;ACC6=0;        //from A
   ACC5=1;ACC4=0;        //to C
   ACC3=0;ACC2=0;        //Room status data
   ACC0=LED1CCls;        //ACC0 current status of room C at room A
  }

  if (P17==0)                 //if CR-B  switch
  {LED4BCrc=!LED4BCrc;   //CR-B status disp at A is complelemnted
   ACC7=0;ACC6=0;       // from A
   ACC5=0;ACC4=1;       // to B
   ACC3=0;ACC2=1;       //CR status data
   ACC0=LED4BCrc;
  }                  //CR-B status passed

  if (P35==0)           //if CR-C switch
  {LED5CCrc=!LED5CCrc;   //CR-C status disp complelemnted
   ACC7=0;ACC6=0;        // from A
   ACC5=1;ACC4=0;        //to C
   ACC3=0;ACC2=1;        // CR C status
   ACC0=LED5CCrc;        //CR-C status passed
  }

   SBUF=ACC;
   while(TI==0);       //transmitting data to B or C
   TI=0;
//buzzer(1, 4);
   EX0=1;                  //enable interrupt
}

//serial port
#pragma interrupt_level 1
interrupt void serialisr(void)
{   unsigned int i,j,Outer=10,Inner=10,lcdln1=0x80, lcdln2=0xC0;
    unsigned char *busy10="LH-10 Busy", *busy11="LH-11 Busy", *nbusy10="LH-10 Free",*nbusy11="LH-11 Free";

    ES=0;RI=0;//disable serial interrupt
             ACC=SBUF;
```

```
if (ACC5==0&&ACC4==0)// to A?
    {
if (ACC7==0&&ACC6==1)// is it from B
      {
if (ACC3==0&&ACC2==0)// is it Room Status Data?
        {LED0BCls=ACC0;
         ACC7=0;ACC6=0;// from A
         ACC5=0;ACC4=1; // to B
         ACC3=1;ACC2=0;// Ack
         SBUF=ACC;
         while(TI==0); // sending ack
         TI=0;
        }

       if (ACC3==0&&ACC2==1) // is it CR status?
         {LED4BCrc=ACC0;
          ACC7=0;ACC6=0;   // from A
          ACC5=0;ACC4=1;   // to B
          ACC3=1;ACC2=0;// Ack
          SBUF=ACC;
          while(TI==0); //sending ack
          TI=0;
         }

if (ACC3==1&&ACC2==0) // is it Ack
        { LED2BAck=1;
          //delay(1);
                                    if (LED0BCls==1)
            displcd(lcdln1,busy10);
          else
           displcd(lcdln1,nbusy10);

         LED2BAck=0;//ack indicator goes off
        }
      }

     if (ACC7==1&&ACC6==0)   // is it from C
      { if (ACC3==0&&ACC2==0)// is it Room Status Data?
         {LED1CCls=ACC0;
                                    ACC7=0;ACC6=0; //from A
          ACC5=1;ACC4=0;// to C
          ACC3=1;ACC2=0; // ack
                                   SBUF=ACC;
                  while(TI==0); //sending ack C
                  TI=0;
          }
       if (ACC3==0&&ACC2==1)// is it CR status?
         {LED5CCrc=ACC0;
                                    ACC7=0;ACC6=0;// from A
          ACC5=1;ACC4=0;// to C
          ACC3=1;ACC2=0; // ack
                                   SBUF=ACC;
                  while(TI==0); //sending ack C
```

```
                         TI=0;
             }
if (ACC3==1&&ACC2==0)// is it Ack
             {LED3CAck=1;
              delay(5);
                                                   if (LED1CCls==1)
                                     displcd(lcdln2,busy11);
                          else
                                     displcd(lcdln2,nbusy11);
             LED3CAck=0;
            }
                           }
       buzzer(5, 4);
      }

  ES=1;// enable serial interrupt

}


void main()
{  unsigned int i;
unsigned char item[]="CL-ROOM";
   buzzer(1, 1);
   LED0BCls=1;
   LED1CCls=1;
   LED2BAck=1;
   LED3CAck=1;
   LED4BCrc=1;
   LED5CCrc=1;
lcdini2();// initializing LCD
   //disp_scrol_name(item);// scroll Displaying in LCD
   //displcd(0x08,item);// Displaying project name

for (i=1;i<4;i++) //LEDs blinking to indicate project is powered on
   {LED0BCls=!LED0BCls;
    LED1CCls=!LED1CCls;
    LED2BAck=!LED2BAck;
    LED3CAck=!LED3CAck;
    LED4BCrc=!LED4BCrc;;
    LED5CCrc=!LED5CCrc;
    delay(2);
   }

   // interrupt and timer settings
   EA=1;EX0=1;EX1=1;ES=1;IT0=1;IT1=1;
   TMOD=0x20;
   TH1=0xFD;
   SCON=0x90;
   TR1=1;

   while(1); // keeping in microcontroller wait condition
```

```
}

//Program to manage Class Room C
#include<htc.h>
#define LEDC0Cls P10//Classroom status
#define LEDC1Crc P11//CR call status
#define LEDC2Ack P12//Acknowledgement indicator
#define Buzz P13//Buzzer line


//Macro to create ISR for serial-port interrupt
#asm
        GLOBAL _serialisr// ISR name = serialisr()
        PSECT vectors,ovrld
        ORG 0x23                        // Int vector
        LJMP _serialisr           // ISR function name again with a preceding underscore added.
        PSECT text
#endasm

//Macro to create ISR for Ext. Interrupt 0
#asm
        GLOBAL _ext0isr           //ISR name = ext1isr()
        PSECT vectors,ovrld
        ORG 0x03                        //Int vector
        LJMP _ext0isr             //ISR function name again with a preceding underscore added.
        PSECT text
#endasm

#asm
        GLOBAL _ext1isr// ISR name = ext1isr()
        PSECT vectors,ovrld
        ORG 0x13// Int vector
        LJMP _ext1isr// ISR function name again with a preceding underscore added.
        PSECT text
#endasm

//Functions
#pragma interrupt_level 0
void reentrant delay(unsigned int msec)    //Function to provide time delay in msec for LCD.
{       unsigned int i,j ;
        for(i=0;i<msec;i++)
        for(j=0;j<100;j++);
}

//Buzzer function
#pragma interrupt_level 0
void reentrant buzzer(unsigned char Length, unsigned int msec)    //Function to provide time delay in msec for LCD.
{       unsigned int i;
   Buzz=1;
for (i=0;i<Length;i++)
         {
    delay(msec);
    Buzz=!Buzz;
    }
}

//ISR for serial-port  interrupt
interrupt void serialisr(void)
{unsigned int i,j,Outer=50, Inner=50;

 ES=0;                  //Disable serial-port interrupt
 RI=0;                  // Clear intpt flag
 ACC=SBUF;          // Reading from the port buffer

 if(ACC5==1&&ACC4==0)        //to C?
   if(ACC7==0&&ACC6==0)      // from A?
     {if(ACC3==0&&ACC2==0)   //Room status data?
        {LEDC0Cls=!LEDC0Cls;
         ACC7=1;ACC6=0;     //from C
         ACC5=0;ACC4=0;     // to A
         ACC3=1;ACC2=0;      //Setting Ack
         SBUF=ACC;                     //Preparing to send Ack to A
         while(TI==0);       //Sending Ack to A
         TI=0;
        }                  //Sent ack to A
       if(ACC3==0&&ACC2==1)   //CR status data?
        {LEDC1Crc=!LEDC1Crc;
         ACC7=1;ACC6=0;      // from C
         ACC5=0;ACC4=0;      // to A
         ACC3=1;ACC2=0;      //Setting Ack
         SBUF=ACC;                     //Preparing to send Ack to A
         while(TI==0);       //Sending Ack to A
         TI=0;
        }

        if(ACC3==1&&ACC2==0)   //Ack data?
        {LEDC2Ack=1;
         delay(5);
         LEDC2Ack=0;
         buzzer(3, 2);
        }
      }
  ES=1;                                                            //Enable serial-port interrupt
 }

//ISR for Ext. Interrupt-0
interrupt void ext0isr(void)
{           EX0=0; //Disable ext intpt-0
            LEDC0Cls=!LEDC0Cls; //Complementing class-room status LED
            ACC=0x00;
            ACC7=1;ACC6=0;  //from C
            ACC5=0;ACC4=0;  //to A
            ACC3=0;ACC2=0;  //Room status data
            ACC0=LEDC0Cls;  //Current room-status data in room C
            SBUF=ACC;
            while(TI==0);
            TI=0;
            EX0=1;                              //Enable external interrupt0
}

//ISR for Ext. Interrupt-1
interrupt void ext1isr(void)
{EX1=0; //Disable ext intpt-0
 LEDC1Crc=!LEDC1Crc; //Complementing class-room status LED
 ACC=0x00;
 ACC7=1;ACC6=0;   //from C
 ACC5=0;ACC4=0;   //to A
 ACC3=0;ACC2=1;   //CR status data
 ACC0=LEDC1Crc;   //Current room-status data in room C
 SBUF=ACC;
 while(TI==0);
 TI=0;
 EX1=1;                              //Enable external interrupt0
}

// Main program
void main()
{  unsigned int i;
   LEDC0Cls=1;
   LEDC2Ack=1;
   LEDC1Crc=1;
for (i=1;i<4;i++)               // Indicators blinking
   {LEDC0Cls=!LEDC0Cls;
    LEDC2Ack=!LEDC2Ack;
    LEDC1Crc=!LEDC1Crc;
    delay(20);
   }
   LEDC0Cls=0;  // Indicator initialising
   LEDC2Ack=0;
   LEDC1Crc=0;
   EA=1;   //interrupt settings
   ES=1;
   EX0=1;IT0=1;
   EX1=1;
   TMOD=0x20; //timer setting
   TH1=0xFD;
   SCON=0x90;
   TR1=1;
   while(1);
}
```

4.3 Hardware Design (CIRCUIT)

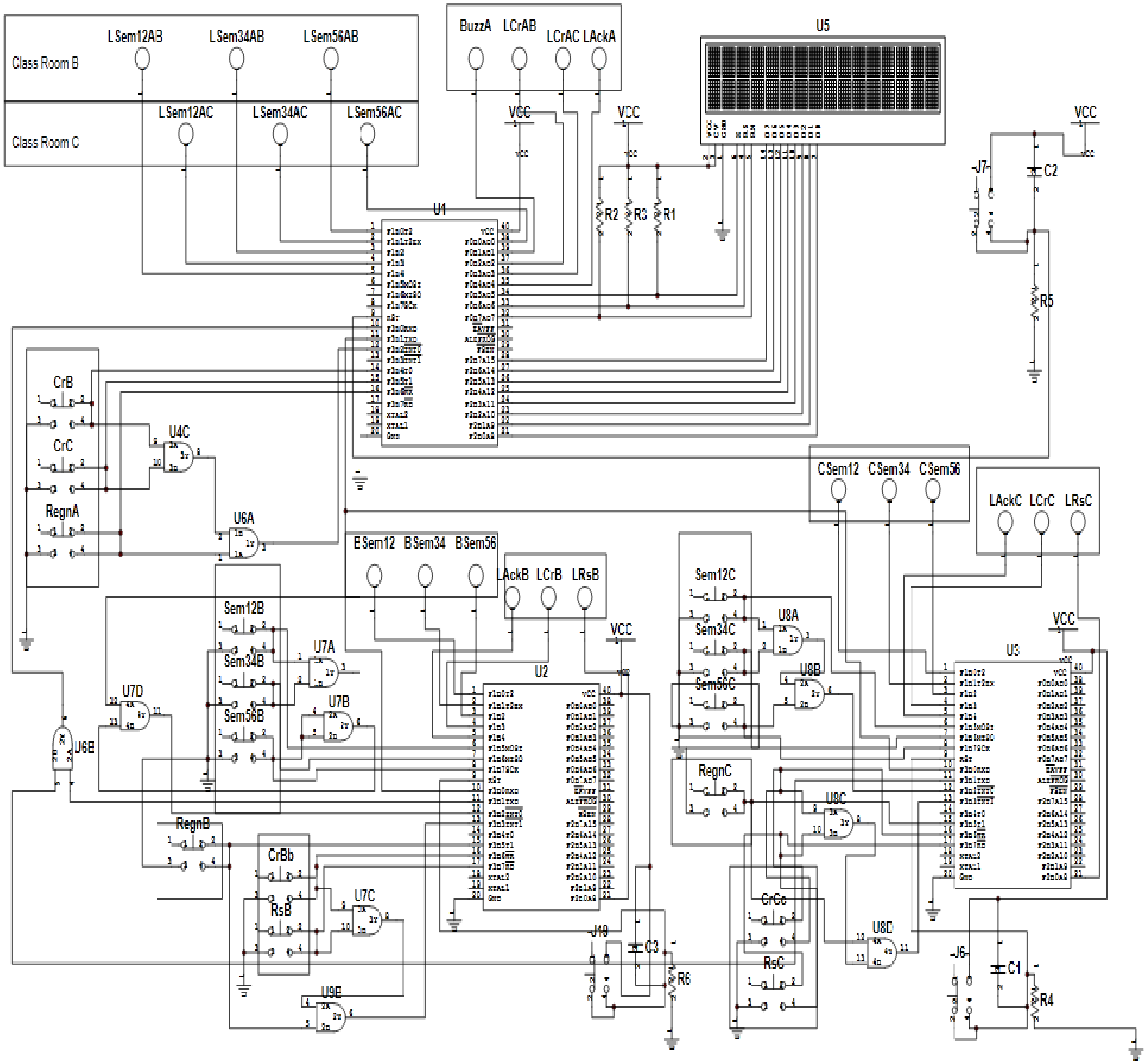


Fig. 4.4: System Circuit

4.4 PCB Design and 3-D model

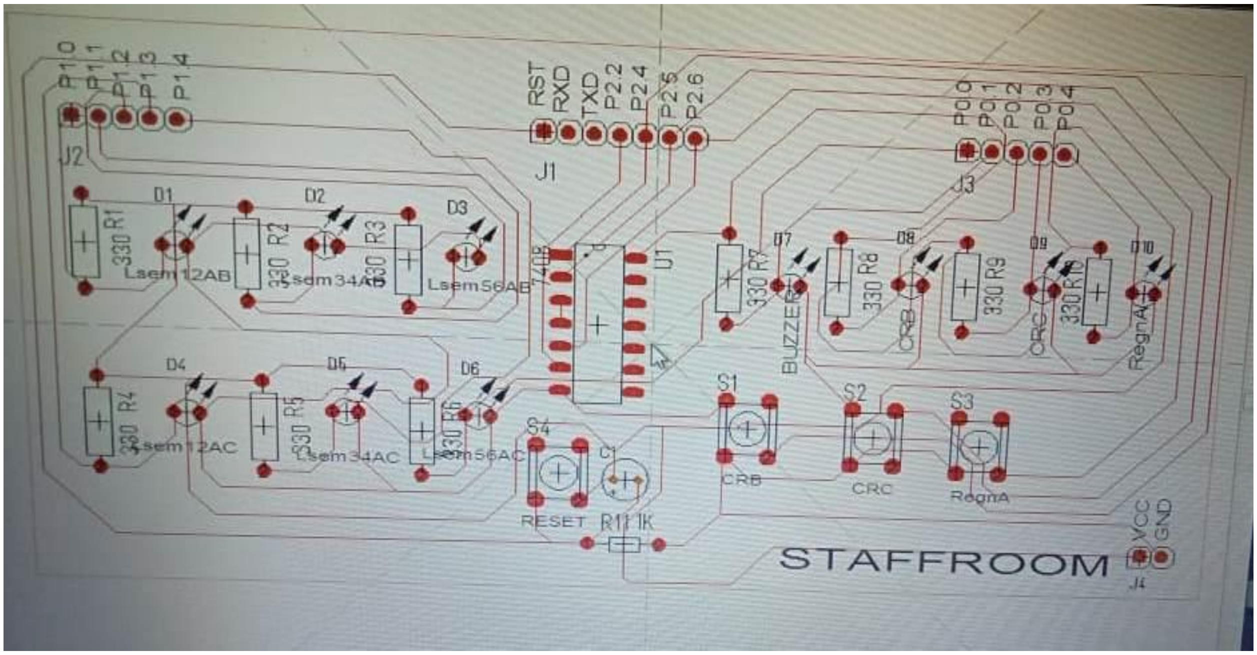


Fig. 4.5: Circuit PCB

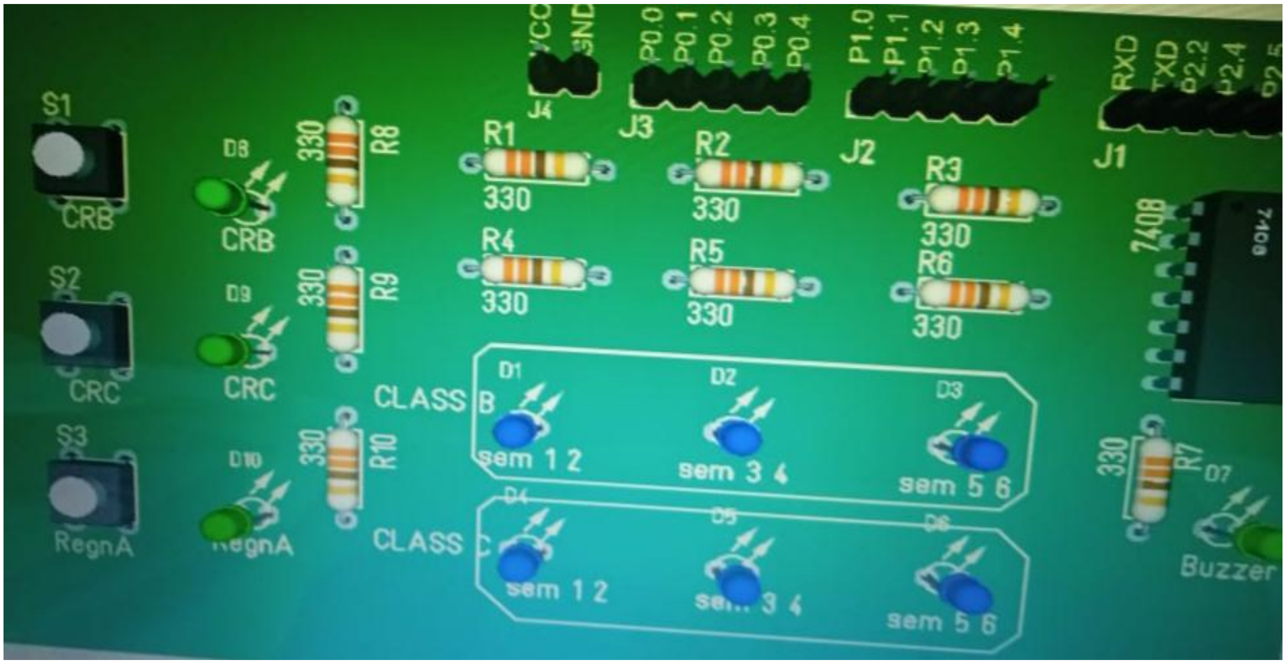


Fig. 4.6: Circuit 3-D model

Chapter- 5

OPERATION, CONCLUSION AND FUTURE DIRECTION

5.1 Operating Instructions (specifications)

1. The system works only if Room-B or Room-C is registered
2. They can be registered by pressing Semester key and Reg. keys (can be done by CR in the class room)
3. Registration of the room and class is visible in Room-A
4. Status (teacher is present or not) of the registered room is also visible in Room-A
5. Once the teacher enters class room (Room-B or Room-C), teacher needs to convey occupied status in the class room and status is reflected in the staff room
6. Once the class ends, teacher needs to convey it in the class room and it will be reflected in the staff room so that the next teacher can occupy that class room
7. And, this is applicable for both class rooms
8. CR call (Call to class representative in Room-C or B) can be made from staff room (Room-A)
9. In response to CR call, the CR call indicator lights in Room-C or B as the case may be.
10. Respective CR can respond in the respective class room by pressing the appropriate key
11. In response to CR response, CR response indicator lights up in staff room.
12. All calls/responses are also alerted through buzzer sounds
13. In staff room the class room and teacher status are displayed or scrolled in LCD.
14. This system will not function if not registered in the class rooms.

5.2 Conclusion and future directions

E-Class room management (for two class rooms) is designed and developed involving both software and hardware. Hardware includes three microcontrollers, LEDs, LCD, push-buttons and buzzer. Software is written in embedded C language. The software is developed with the help of designed flow-chart form of algorithm. The entire project was tested for functionality under simulated environment using Multisim. It worked as per the specifications. Its PCB was designed after the simulation of the circuit using Ultiboard tool available with Multisim. A 3-D model view is also seen to visualise the circuit.

In future, the system can be extended for multiple rooms, and communication can be made possible between several microcontrollers through wireless, thus it can be made commercialized for the intending applications.

5.3 References

- www.electronicwings.com
- www.gadgetronicx.com
- www.googlescholor.com