كلية الهندسه

Systems & Biomedical Engineering Department

Elevator

Course**: Bioelectronics**

Report By:

Rawan Sayed	sec:1	BN:30
Sara Adel	sec:1	BN:35
Al-Zahraa Eid	sec:1	BN:16
Ahmed Adel Ahmed	sec:1	BN:6
Kirolos Dawood	sec:1	BN:15

Submitted to Dr. Ahmed Ehab

ABSTRACT

We design a simple and creative Elevator system for building consists of 4 floors, consists of four floors. Each floor has two buttons one for going up, and one for going down except for the ground and last floor have one button only. Elevator should have at most four people ins.

Table of Contents

List of Figures

Introduction

User specifies/select the floor from Keypad on which he wants to move to. Push buttons are used to call elevator to go up or down. There are total 4 dummy floors in our lift control system. After selecting the floor lift/elevator starts to move to the user selected floor by motor. Once the elevator reaches to the users desired floor, there is a 7-segment shows the current floor/level and another 7-segment shows number of people inside the elevator.

Conditions:

The elevator door waits for 5 seconds before closing, and the door can be stopped from closing by pushing an open button. Also, the elevator door opens if someone blocks the door.

If the elevator is going up, it should not stop for a " going down" request, and vice versa: if it is going down it should not stop for a " going up request ".The elevator system save the requests, for example if the elevator is going up, and you pressed on the " going down " button, the elevator shall ignore the request initially, but when it reach the top floor it should go back to pick up that one who want to go down.

SECTION II: Circuit diagram

Our whole Elevator `circuit diagram:

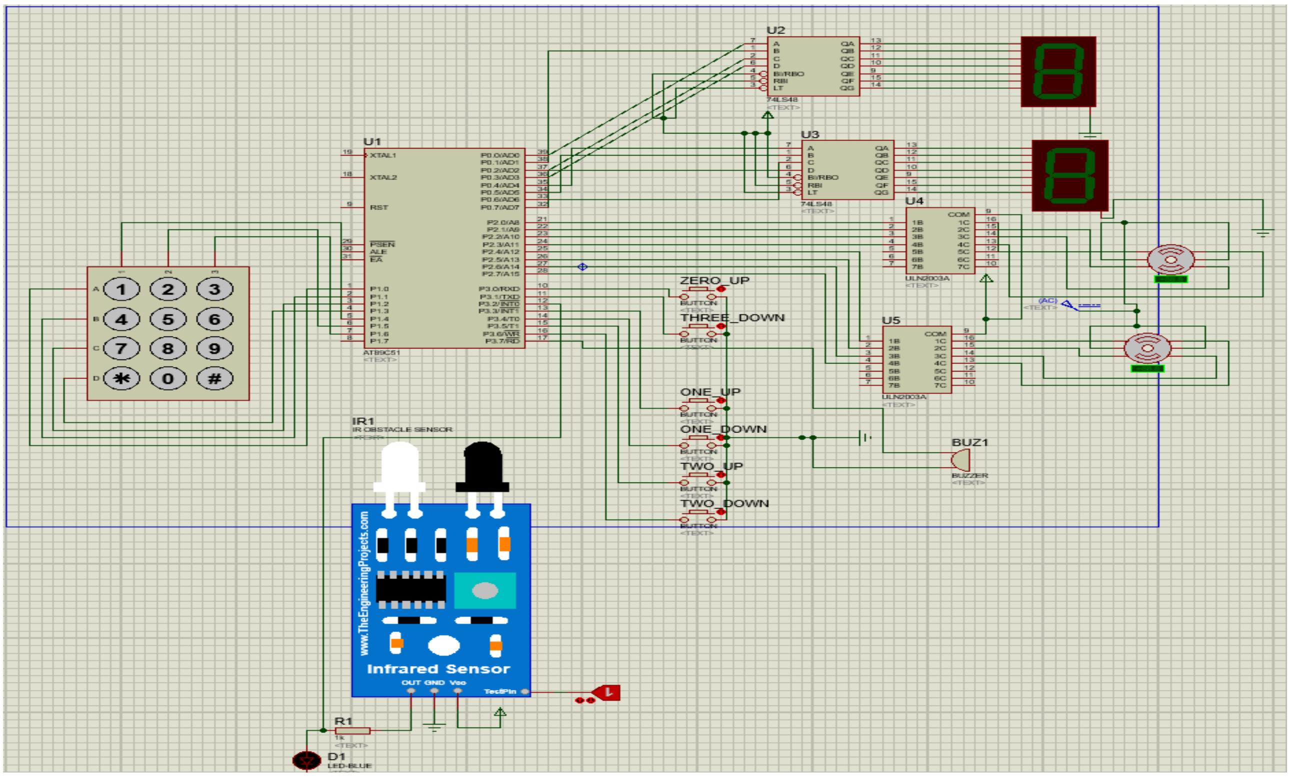


Figure 1: Schematic Diagram

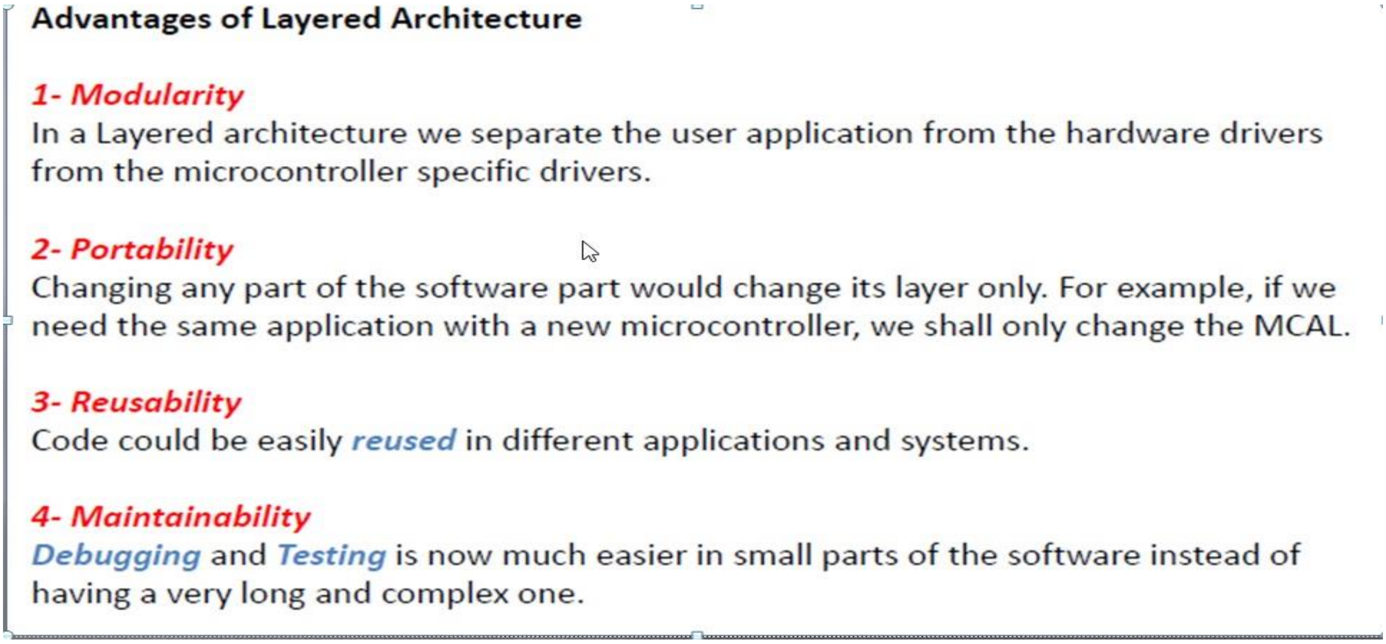
Used components:

- ✓ (2) 7-Segment.
- ✓ (2) BCD.
- ✓ (2) Stepper motor.
- ✓ (6) Push Button.
- ✓ Keypad.
- ✓ IR sensor.
- ✓ At89c51 microcontroller.

SECTION III: Code

Summary about our implementation:

We use AUTOSAR layers for its advantages which is:



Advantages of Layered Architecture

1- Modularity

In a Layered architecture we separate the user application from the hardware drivers from the microcontroller specific drivers.

2- Portability

Changing any part of the software part would change its layer only. For example, if we need the same application with a new microcontroller, we shall only change the MCAL.

3- Reusability

Code could be easily ***reused*** in different applications and systems.

4- Maintainability

Debugging and ***Testing*** is now much easier in small parts of the software instead of having a very long and complex one.

Our Layers:

- DIO file
- 7-Segment file (.c / .h)
- Keypad file (.c/ .h)
- Stepper file (.c / .h)

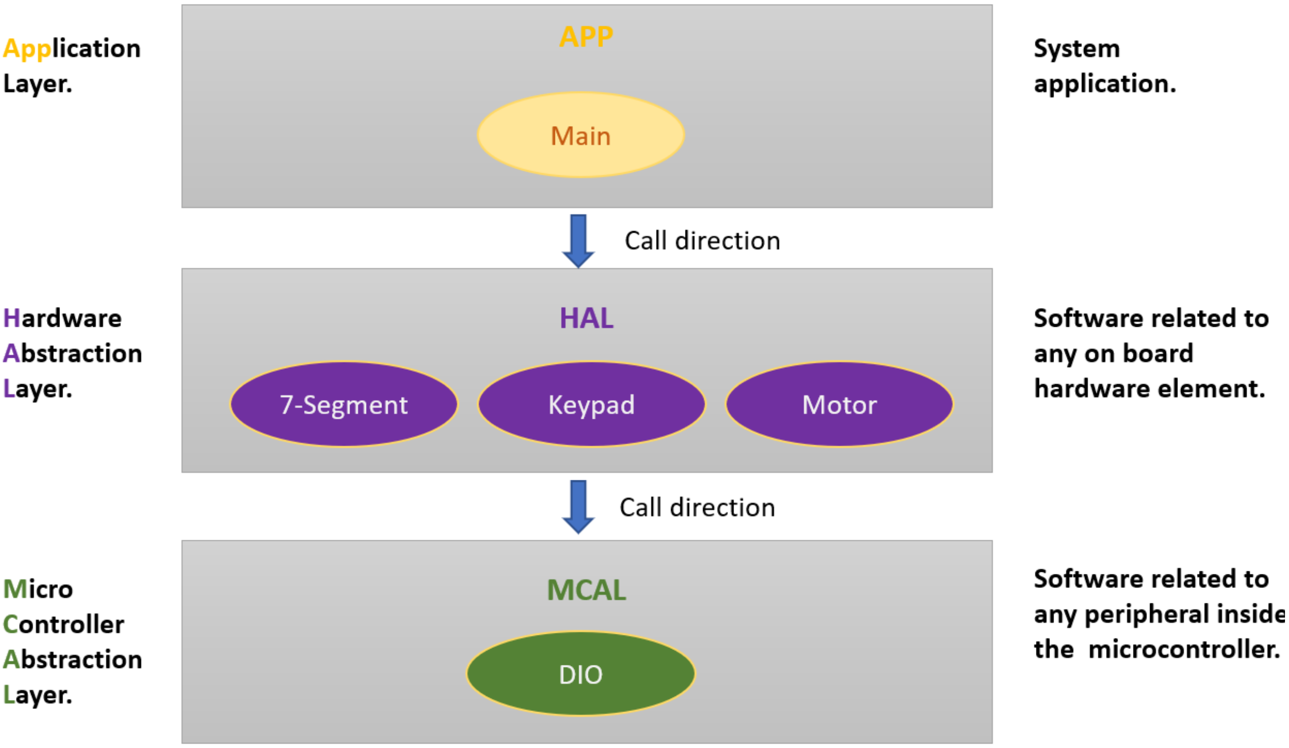
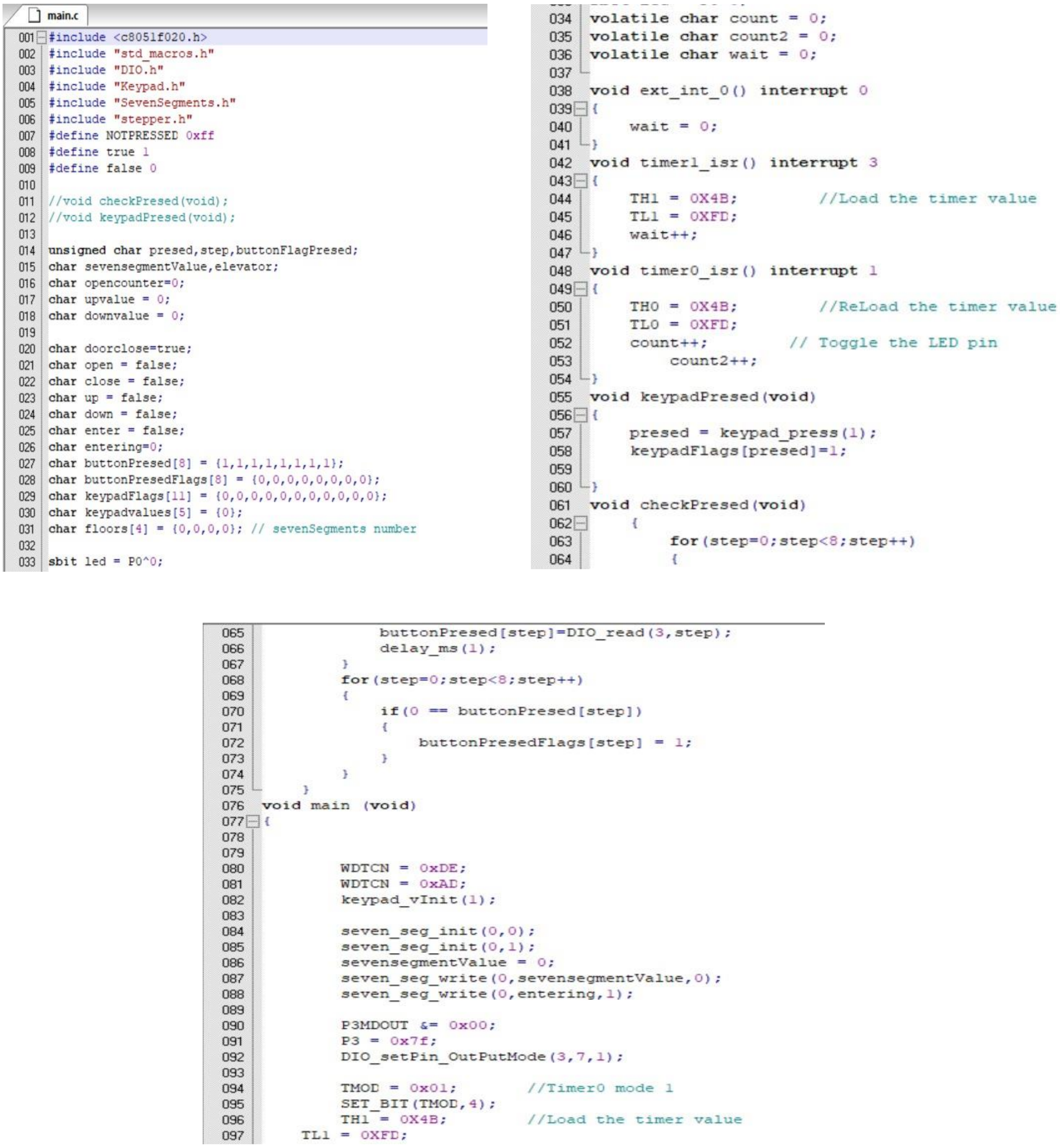

Application Layer.
APP
Main
System application.
Call direction
Hardware Abstraction Layer.
HAL
7-Segment
Keypad
Motor
Software related to any on board hardware element.
Call direction
Micro Controller Abstraction Layer.
MCAL
DIO
Software related to any peripheral inside the microcontroller.

Figure 2:AUTOSAR Layered Architecture

Part 1:



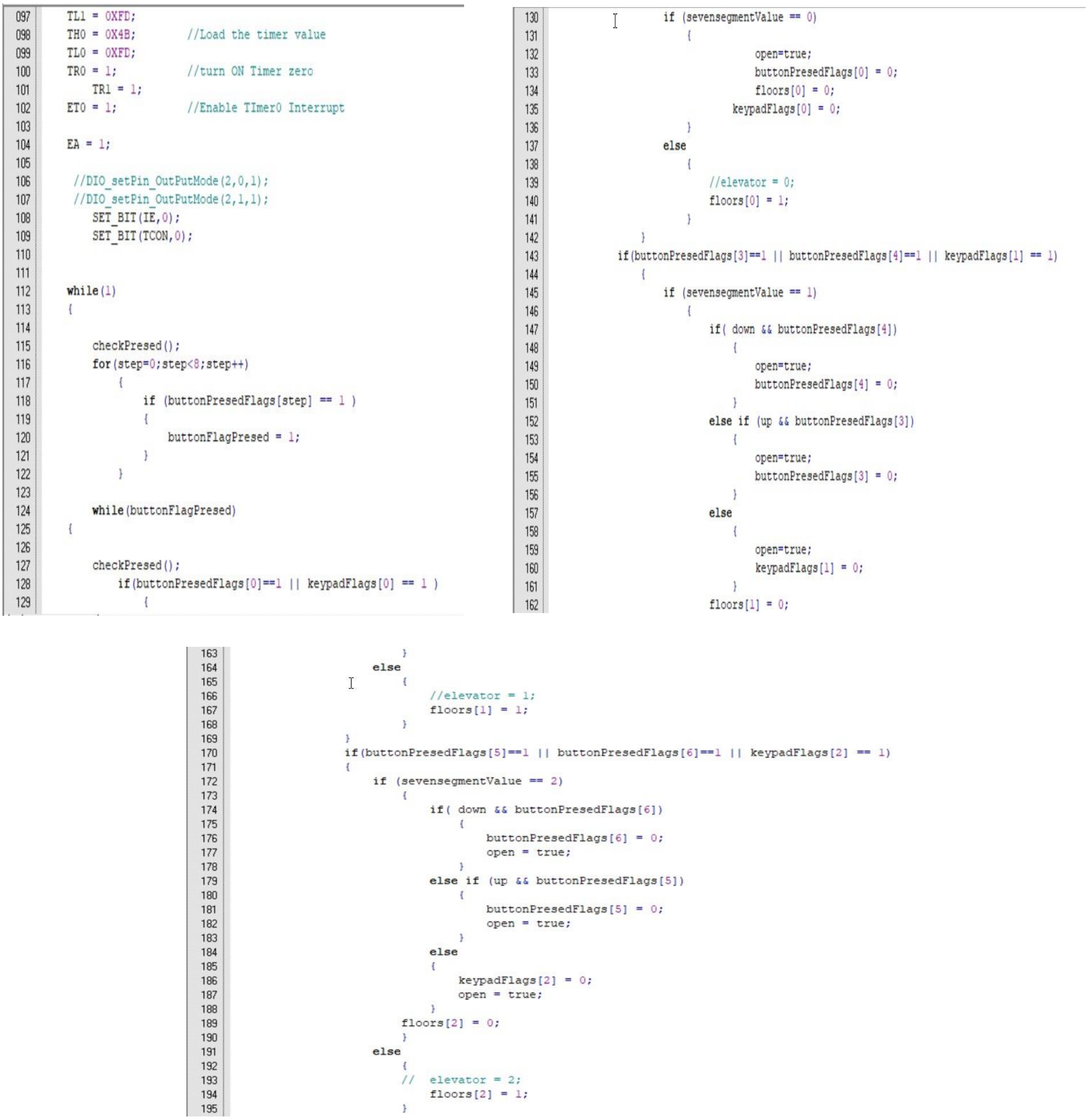
```
#include <c8051f020.h>
#include "std_macros.h"
#include "DIO.h"
#include "Keypad.h"
#include "SevenSegments.h"
#include "stepper.h"
#define NOTPRESSED 0xff
#define true 1
#define false 0

//void checkPresed(void);
//void keypadPresed(void);

unsigned char presed,step,buttonFlagPresed;
char sevensegmentValue,elevator;
char opencounter=0;
char upvalue = 0;
char downvalue = 0;

char doorclose=true;
char open = false;
char close = false;
char up = false;
char down = false;
char enter = false;
char entering=0;
char buttonPresed[8] = {1,1,1,1,1,1,1,1};
char buttonPresedFlags[8] = {0,0,0,0,0,0,0,0};
char keypadFlags[11] = {0,0,0,0,0,0,0,0,0,0,0};
char keypadvalues[5] = {0};
char floors[4] = {0,0,0,0}; // sevenSegments number

sbit led = P0^0;
```

```
volatile char count = 0;
volatile char count2 = 0;
volatile char wait = 0;

void ext_int_0() interrupt 0
{
    wait = 0;
}
void timer1_isr() interrupt 3
{
    TH1 = 0X4B;         //Load the timer value
    TL1 = 0XFD;
    wait++;
}
void timer0_isr() interrupt 1
{
    TH0 = 0X4B;         //ReLoad the timer value
    TL0 = 0XFD;
    count++;        // Toggle the LED pin
        count2++;
}
void keypadPresed(void)
{
    presed = keypad_press(1);
    keypadFlags[presed]=1;

}
void checkPresed(void)
    {
        for(step=0;step<8;step++)
        {
```

```
                buttonPresed[step]=DIO_read(3,step);
                delay_ms(1);
            }
            for(step=0;step<8;step++)
            {
                if(0 == buttonPresed[step])
                {
                    buttonPresedFlags[step] = 1;
                }
            }
        }
void main (void)
{


        WDTCN = 0xDE;
        WDTCN = 0xAD;
        keypad_vInit(1);

        seven_seg_init(0,0);
        seven_seg_init(0,1);
        sevensegmentValue = 0;
        seven_seg_write(0,sevensegmentValue,0);
        seven_seg_write(0,entering,1);

        P3MDOUT &= 0x00;
        P3 = 0x7f;
        DIO_setPin_OutPutMode(3,7,1);

        TMOD = 0x01;        //Timer0 mode 1
        SET_BIT(TMOD,4);
        TH1 = 0X4B;         //Load the timer value
    TL1 = 0XFD;
```

Figure 3:Import Libraries and Create Interrupt functions

Part 2:

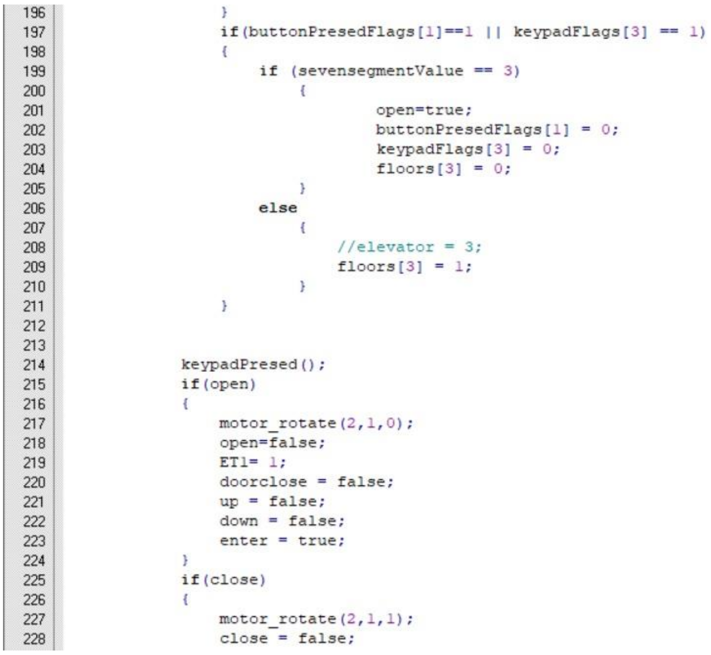


```
TL1 = 0XFD;
TH0 = 0X4B;         //Load the timer value
TL0 = 0XFD;
TR0 = 1;            //turn ON Timer zero
    TR1 = 1;
ET0 = 1;            //Enable TImer0 Interrupt

EA = 1;

 //DIO_setPin_OutPutMode(2,0,1);
 //DIO_setPin_OutPutMode(2,1,1);
    SET_BIT(IE,0);
    SET_BIT(TCON,0);


while(1)
{

    checkPresed();
    for(step=0;step<8;step++)
        {
            if (buttonPresedFlags[step] == 1 )
            {
                buttonFlagPresed = 1;
            }
        }

    while(buttonFlagPresed)
{

    checkPresed();
        if(buttonPresedFlags[0]==1 || keypadFlags[0] == 1 )
            {
```

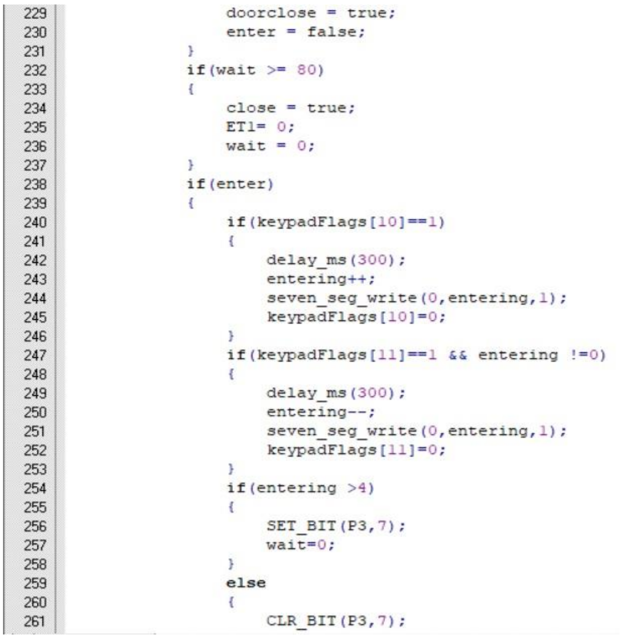
```
                if (sevensegmentValue == 0)
                    {
                            open=true;
                            buttonPresedFlags[0] = 0;
                            floors[0] = 0;
                        keypadFlags[0] = 0;
                    }
                else
                    {
                        //elevator = 0;
                        floors[0] = 1;
                    }
            }
        if(buttonPresedFlags[3]==1 || buttonPresedFlags[4]==1 || keypadFlags[1] == 1)
            {
                if (sevensegmentValue == 1)
                    {
                        if( down && buttonPresedFlags[4])
                            {
                                open=true;
                                buttonPresedFlags[4] = 0;
                            }
                        else if (up && buttonPresedFlags[3])
                            {
                                open=true;
                                buttonPresedFlags[3] = 0;
                            }
                        else
                            {
                                open=true;
                                keypadFlags[1] = 0;
                            }
                        floors[1] = 0;
```

```
                    }
                else
                    {
                        //elevator = 1;
                        floors[1] = 1;
                    }
            }
            if(buttonPresedFlags[5]==1 || buttonPresedFlags[6]==1 || keypadFlags[2] == 1)
            {
                if (sevensegmentValue == 2)
                    {
                        if( down && buttonPresedFlags[6])
                            {
                                buttonPresedFlags[6] = 0;
                                open = true;
                            }
                        else if (up && buttonPresedFlags[5])
                            {
                                buttonPresedFlags[5] = 0;
                                open = true;
                            }
                        else
                        {
                            keypadFlags[2] = 0;
                            open = true;
                        }
                    floors[2] = 0;
                    }
                else
                    {
                    //  elevator = 2;
                        floors[2] = 1;
                    }
```

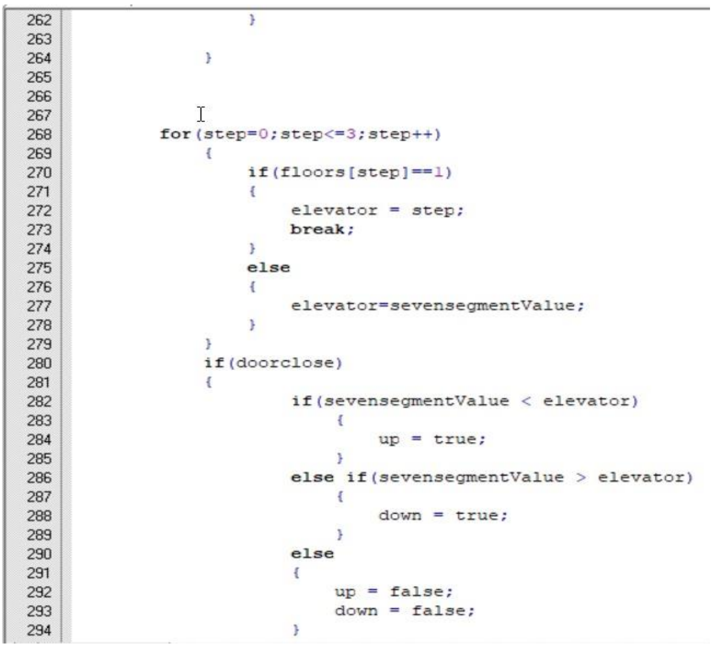
Figure 4: our conditions



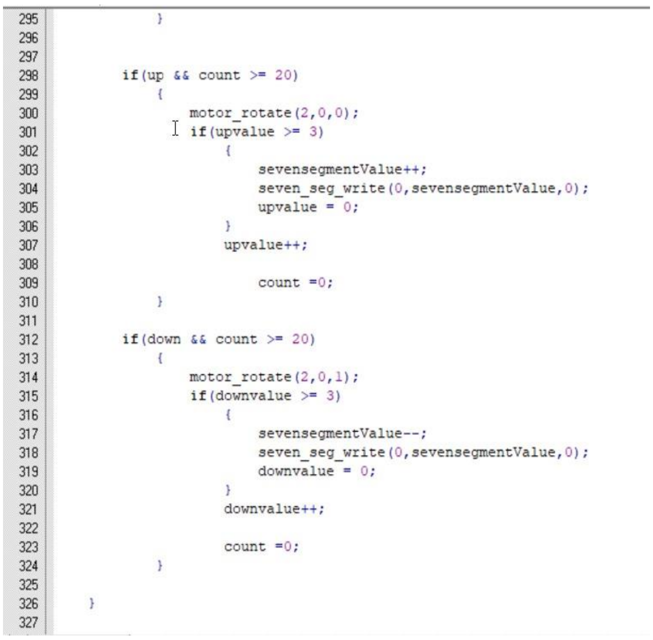
```
            }
            if(buttonPresedFlags[1]==1 || keypadFlags[3] == 1)
            {
                if  (sevensegmentValue == 3)
                    {
                            open=true;
                            buttonPresedFlags[1] = 0;
                            keypadFlags[3] = 0;
                            floors[3] = 0;
                    }
                else
                    {
                        //elevator = 3;
                        floors[3] = 1;
                    }
            }


        keypadPresed();
        if(open)
        {
            motor_rotate(2,1,0);
            open=false;
            ET1= 1;
            doorclose = false;
            up = false;
            down = false;
            enter = true;
        }
        if(close)
        {
            motor_rotate(2,1,1);
            close = false;
```



```
            doorclose = true;
            enter = false;
        }
        if(wait >= 80)
        {
            close = true;
            ET1= 0;
            wait = 0;
        }
        if(enter)
        {
            if(keypadFlags[10]==1)
            {
                delay_ms(300);
                entering++;
                seven_seg_write(0,entering,1);
                keypadFlags[10]=0;
            }
            if(keypadFlags[11]==1 && entering !=0)
            {
                delay_ms(300);
                entering--;
                seven_seg_write(0,entering,1);
                keypadFlags[11]=0;
            }
            if(entering >4)
            {
                SET_BIT(P3,7);
                wait=0;
            }
            else
            {
                CLR_BIT(P3,7);
```



```
                }

            }


        for(step=0;step<=3;step++)
            {
                if(floors[step]==1)
                {
                    elevator = step;
                    break;
                }
                else
                {
                    elevator=sevensegmentValue;
                }
            }
            if(doorclose)
            {
                    if(sevensegmentValue < elevator)
                        {
                            up = true;
                        }
                    else if(sevensegmentValue > elevator)
                        {
                            down = true;
                        }
                    else
                        {
                            up = false;
                            down = false;
                        }
```



```
            }


        if(up && count >= 20)
            {
                motor_rotate(2,0,0);
                if(upvalue >= 3)
                    {
                        sevensegmentValue++;
                        seven_seg_write(0,sevensegmentValue,0);
                        upvalue = 0;
                    }
                    upvalue++;

                        count =0;
            }

        if(down && count >= 20)
            {
                motor_rotate(2,0,1);
                if(downvalue >= 3)
                    {
                        sevensegmentValue--;
                        seven_seg_write(0,sevensegmentValue,0);
                        downvalue = 0;
                    }
                    downvalue++;

                    count =0;
            }

    }

```

Figure 5: Continue conditions