

Design assignment on Smart Lighting System

In the partial fulfillment of the course

CS F241 – Microprocessor Programming & Interfacing

BITS- Pilani Goa Campus



Submitted to Prof. K.R. Anupama

By batch no. B142

Shubham Jain

2013A7PS049G

Nisarg Shah

2013A7PS694G

Tejas Dilip Jogi

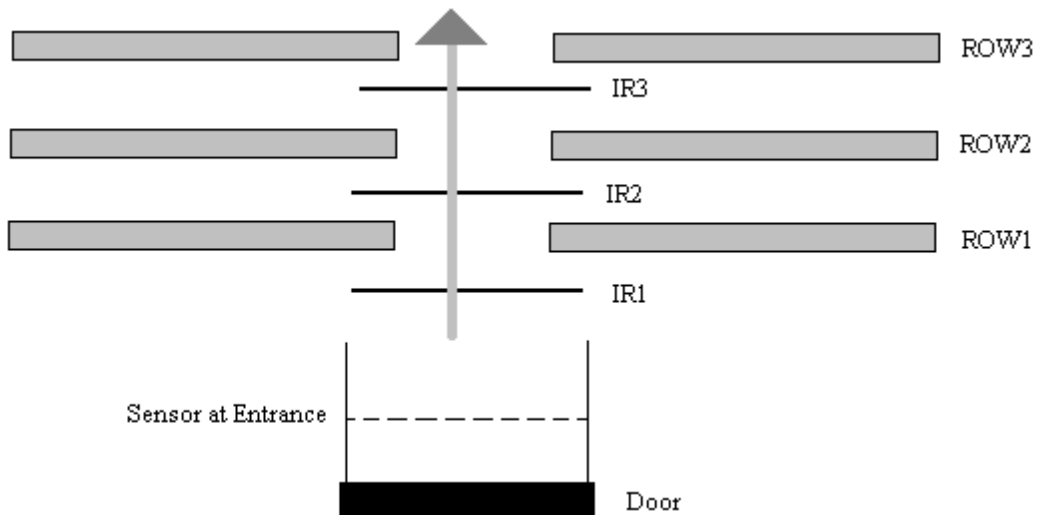
2013A7PS010G

Abhishek Sukhwani

2013A3PS284G

P14. System to be designed Smart Lighting System Description:

This is a lighting system for a conference room. As the seats get filled the light should be turned on. The rows are filled from row1 onwards. There are 2 lights per row. As each row begins to get filled the lights get turned on as each row empties completely the light gets turned off. You can assume there are atleast 10 rows.



1.Specifications:

1. The system to be designed is Smart Lighting System.
2. There are a total of 10 rows.
3. There is an IR proximity sensor present before any row and at gate.
4. When a person passes through the sensor the sensor gives a logic-1.
5. The sensor is connected to two lights just above its corresponding row.
6. Every time a person comes in the room and sits in a row, the lights corresponding to that row must be turned on automatically.

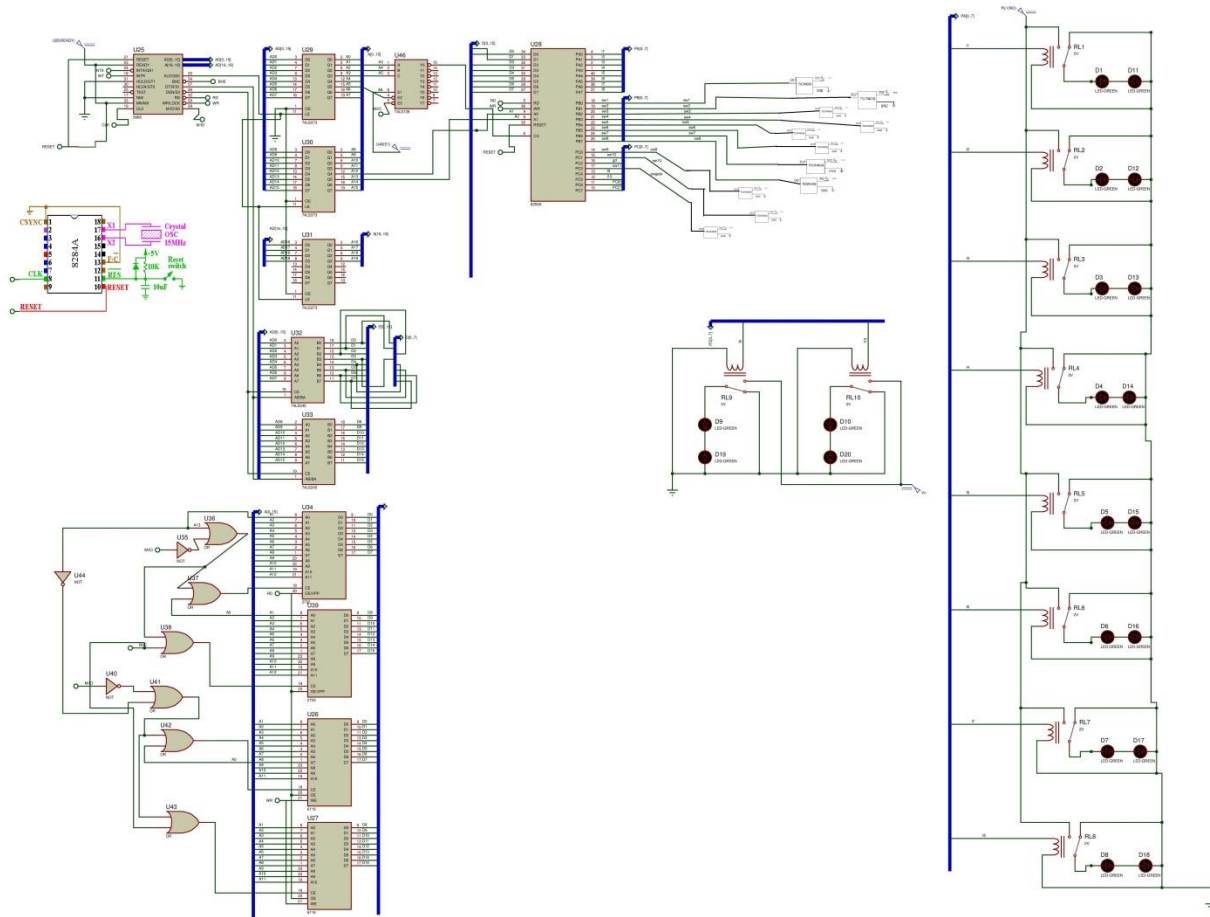
2.Assumptions made:

1. The room first gets fully filled, and then it gets fully empty.
2. When the people will be coming in the room, they will be sitting at the first row (from the gate) that would be empty.
3. When people will be leaving the room, any person from any row may stand and go out of the room.
4. At any particular instance of time, only one sensor of the 11 sensors (10 rows and 1 of gate) will be high.
5. When a person would be coming inside the room, he would simply come and sit and while leaving he would just stand up and go out, i.e., any person will not be making any transitions inside the room.

3.Components Used:

1. Microprocessor 8086 -1
2. Decoder 74LS138 -1
3. Programmable Peripheral Interface 8255A -1
4. IR sensor TSOP 4038 -11
5. Relay -10
6. Buffer 74LS373 -2
7. Octal Bus Transceivers 74LS245 -2
8. Clock generator 8284A -1
9. ROM 2732 -2
10. RAM 6116 -2
11. AND 7408
12. OR 7432
13. NOT 7404
14. Green LEDs

4.Circuit Diagram (also attached in zip):



5.Memory Mapping:-

2732 (8K) ROM :- 00000h – 01FFFh

6116 (4K) RAM :- 02000h – 02FFFh

Base Address for 8255 – 00000h

6.ALP:

#make_bin#

#LOAD_SEGMENT=FFFFh#

#LOAD_OFFSET=0000h#

#CS=0000h#

#IP=0000h#

#DS=0000h#

#ES=0000h#

#SS=0000h#

#SP=FFFEh#

#AX=0000h#

#BX=0000h#

#CX=0000h#

#DX=0000h#

#SI=0000h#

#DI=0000h#

#BP=0000h#

creg equ 06h

porta equ 00h

portb equ 02h

portc equ 04h

; constants

max_cap db 03h ; max capacity of the room

total db 00h ; total no. of people entered

row_counter db 00h ; no. of people currently sitting in a row

row_number db 1 ; row no. currenty being filled

; counters for each row (used when leaving)

row1 db 00h

row2 db 00h

row3 db 00h

row4 db 00h

row5 db 00h

row6 db 00h

row7 db 00h

row8 db 00h

row9 db 00h

row10 db 00h

pc4 db 10h

pc5 db 20h

off db 0ffh

; add your code here

 jmp st1

 db 509 dup(0)

;IVT entry for 80H

 dw t_isr

 dw 0000

 db 508 dup(0)

;main program

st1: cli

; intialize ds, es,ss to start of RAM

```
mov    ax,0200h
mov    ds,ax
mov    es,ax
mov    ss,ax
mov    sp,0FFFEH
```

;intialise portb as input &portc as output

t_isr:

```
mov max_cap, 03h
mov total, 00h
mov row_counter, 00h
mov row_number, 1
```

```
mov    al, 10000011b
out 06h, al
```

; int 3h ; breakpoint

```
; reset o/p ports
mov al, 00h
out porta, al
```

; int 3h ; breakpoint

```
mov al, 00001001b
out portc, al
mov al, 00001011b
out portc, al
mov al, 00001101b
out portc, al
```

```
mov al, 00001111b
```

```
out portc, al
```

```
arrive:
```

```
; int 3h ; breakpoint
```

```
in al, portc
```

```
and al, 08h
```

```
jz arrive ; loop if no activity
```

```
arrive1:
```

```
in al, portc
```

```
and al, 08h
```

```
jnz arrive1
```

```
inc total
```

```
inc row_counter
```

```
mov al, row_counter
```

```
cmp al, max_cap ; check if row is full
```

```
jne x2
```

```
mov row_counter, 0 ; row full, increase row number
```

```
inc row_number
```

```
cmp row_number, 11 ; room full
```

```
je leaving_init
```

```
jmp arrive
```


x2:

cmp row_counter, 1 ; if row not empty, do nothing

jne arrive

cmp row_number, 1

jl arrive

mov al, 00000001b

out porta, al

cmp row_number, 2

jl arrive

mov al, 00000011b

out porta, al

cmp row_number, 3

jl arrive

mov al, 00000111b

out porta, al

cmp row_number, 4

jl arrive

mov al, 00001111b

out porta, al

cmp row_number, 5

jl arrive

mov al, 00011111b

out porta, al

cmp row_number, 6

jl arrive

mov al, 00111111b

out porta, al

cmp row_number, 7

jl arrive

mov al, 01111111b

out porta, al

cmp row_number, 8

jl arrive

mov al, 11111111b

out porta, al

cmp row_number, 9

jl arrive

mov al, 10h

out portc, al

cmp row_number, 10

jle arrive

mov al, 30h

out portc, al

jmp arrive

; leaving starts

leaving_init: ; init all counters to max capacity

mov al, max_cap

mov row1, al

mov al, max_cap

mov row2, al

mov al, max_cap

mov row3, al

mov al, max_cap

mov row4, al

mov al, max_cap

mov row5, al

```
mov al, max_cap  
mov row6, al
```

```
mov al, max_cap  
mov row7, al
```

```
mov al, max_cap  
mov row8, al
```

```
mov al, max_cap  
mov row9, al
```

```
mov al, max_cap  
mov row10, al
```

```
mov al, 10h  
mov pc4, al
```

```
mov al, 20h  
mov pc5, al
```

```
mov off, 0ffh
```

```
leaving:
```

```
;      int 3
```

```
leave1:
```

```
    cmp row1, 0
```

```

        jz leave2
        in al, portb
            and al, 01h
            jz leave2
;        int 3
        dec row1
        jnz sec
        mov bl, off
        and bl, 0feh
        mov off, bl
        mov al, off
        out porta, al
        jmp sec

```

```

leave2:
        cmp row2, 0
        jz leave3
        in al, portb
            and al, 02h
            jz leave3
;        int 3
        dec row2
        jnz sec
        mov bl, off
        and bl ,0fdh
        mov off, bl
        mov al ,off
        out porta, al
        jmp sec

```

leave3:

 cmp row3, 0

 jz leave4

 in al, portb

 and al, 04h

 jz leave4

 dec row3

 jnz sec

 mov bl, off

 and bl, 0fbh

 mov off, bl

 mov al, off

 out porta, al

 jmp sec

leave4:

 cmp row4, 0

 jz leave5

 in al, portb

 and al, 08h

 jz leave5

 dec row4

 jnz sec

 mov bl, off

 and bl, 0f7h

 mov off, bl

 mov al, off

 out porta, al

jmp sec

leave5:

cmp row5, 0

jz leave6

in al, portb

and al, 10h

jz leave6

dec row5

jnz sec

mov bl, off

and bl, 0efh

mov off, bl

mov al, off

out porta, al

jmp sec

leave6:

cmp row6, 0

jz leave7

in al, portb

and al, 20h

jz leave7

dec row6

jnz sec

mov bl, off

and bl, 0dfh

mov off, bl

mov al, off

out porta, al

jmp sec

leave7: cmp row7, 0

jz leave8

in al, portb

and al, 40h

jz leave8

dec row7

jnz sec

mov bl, off

and bl, 0bfh

mov off, bl

mov al, off

out porta, al

jmp sec

leave8:

cmp row8, 0

jz leave9

in al, portb

and al, 80h

jz leave9

dec row8

jnz sec

mov bl, off

and bl, 07fh

mov off, bl

mov al, off


```
    out porta, al
    jmp sec
```

```
leave9: cmp row9, 0
```

```
    jz leave10
```

```
        in al, portc
        and al, 01h
        jz leave10
        dec row9
        jnz sec
        mov pc4, 00h
        mov al, 00h
        or al, pc4
        or al, pc5
        out portc, al
        jmp sec
```

```
leave10: cmp row10, 0
```

```
    jz leaving
```

```
    in al, portc
```

```
        and al, 02h
        jz leaving
        dec row10
        jnz sec
        mov pc5, 00h
        mov al, 00h
        or al, pc4
        or al, pc5
        out portc, al
```

```
;          int 3  
          jmp sec
```

```
sec:      in al, portc ; Wait till gate sensor 1  
          and al, 08h  
          jz sec
```

```
sec1:    in al, portc ; Wait till the gate sensor 0  
          and al, 08h  
          jnz sec1
```

```
;      int 3  
          dec total  
          jz exit ; if class empty, exit  
          jmp leaving
```

```
exit:
```

```
end
```

7.Flow Chart:

