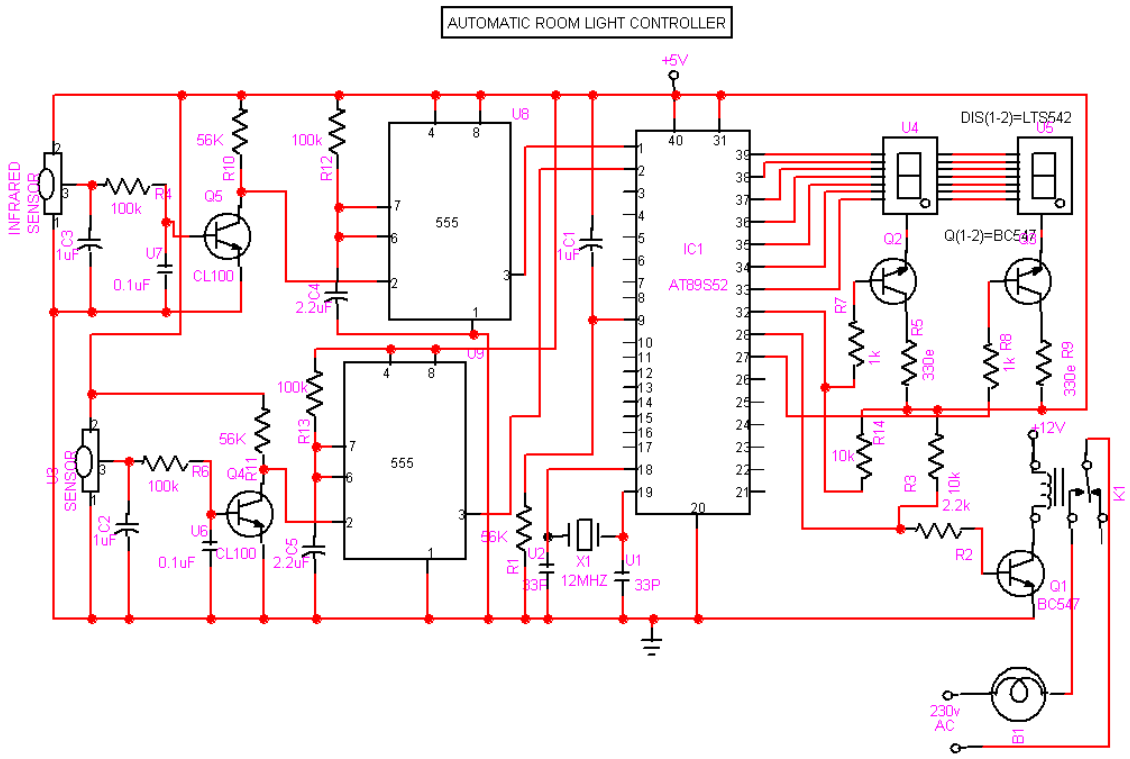
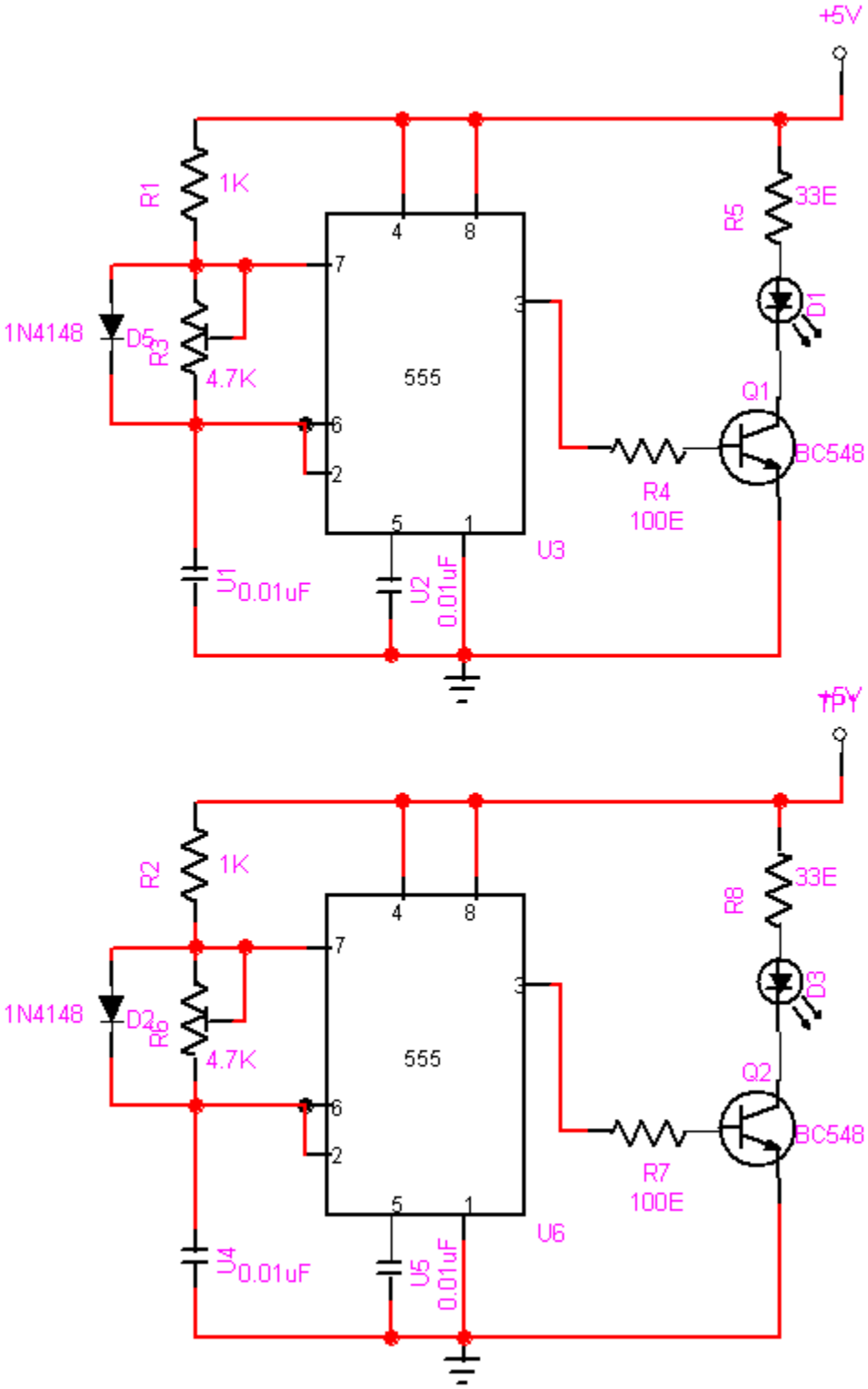
Automatic light

This Project “Automatic Room Light Controller with Visitor Counter using Microcontroller” is a reliable circuit that takes over the task of controlling the room lights as well us counting number of persons/ visitors in the room very accurately. When somebody enters into the room then the counter is incremented by one and the light in the room will be switched ON and when any one leaves the room then the counter is decremented by one. The light will be only switched OFF until all the persons in the room go out. The total number of persons inside the room is also displayed on the seven segment displays. The microcontroller does the above job. It receives the signals from the sensors, and this signal is operated under the control of software which is stored in ROM. Microcontroller AT89S52 continuously monitor the Infrared Receivers, When any object pass through the IR Receiver's then the IR Rays falling on the receivers are obstructed , this obstruction is sensed by the Microcontroller




+5V
R1
1K
R5
33E
4
8
7
3
1N4148
D5
R3
4.7K
555
D1
Q1
6
2
BC548
R4
100E
5
1
U3
U1
0.01uF
U2
0.01uF
+5V
R2
1K
R8
33E
4
8
7
3
1N4148
D2
R6
4.7K
555
D3
Q2
6
2
BC548
R7
100E
5
1
U6
U4
0.01uF
U5
0.01uF

Project Programming

```
INCLUDE reg_51.pdf

RB0     EQU     000H    ; Select Register Bank 0
RB1     EQU     008H    ; Select Register Bank 1  ...poke to PSW to use


DIS_A EQU    P0.2
DIS_B EQU    P0.3
DIS_C EQU    P0.4
DIS_D EQU    P0.6
DIS_E EQU    P0.5
DIS_F EQU    P0.1
DIS_G EQU    P0.0

DIS1    EQU    P0.7
DIS2    EQU    P2.6

LIGHTEQU    P2.7

SEN1   EQU    P1.0
SEN2   EQU    P1.1

DSEG            ; This is internal data memory
ORG     20H     ; Bit adressable memory
COUNT:          DS      1
SPEED:          DS      1
VALUE_1:        DS      1
VALUE_2:        DS      1

NUMB1:          DS      1
NUMB2:          DS      1
NUMB3:          DS      1
VISITOR:        DS      1
STACK           DATA 3FH

CSEG     AT      0       ; RESET VECTOR
;---------==========---------==========---------=========---------
;            PROCESSOR INTERRUPT AND RESET VECTORS
;---------==========---------==========---------=========---------

          ORG     00H                     ; Reset
          JMP     MAIN
```

```
        ORG   000BH          ;Timer Interrupt0
        JMP   REFRESH

;---------==========----------==========----------=========---------
;  Main routine. Program execution starts here.
;---------==========----------==========----------=========---------
MAIN:
        MOV PSW,#RB0       ; Select register bank 0
  MOV SP,STACK
        CLR LIGHT

  MOV VISITOR,#00H
        MOV SPEED,#00H
        MOV COUNT,#00H
        MOV VALUE_1,#15H
        MOV VALUE_2,#15H
        CLR DIS1
        CLR DIS2

        MOV TMOD,#01H              ;enable timer0 for scanning
        MOV TL0,#00H
        MOV TH0,#0FDH
        SETB ET0
        SETB EA
        SETB TR0               ;Start the Timer


                MOV VALUE_1,#00H
                MOV VALUE_2,#00H
                SETB SEN1
                SETB SEN2
UPP:    JNB SEN1,UP_COUNT
                JB SEN2,UPP

                MOV A,VISITOR                       ;DOWN COUNTING
                CJNE A,#00,UAPS
                CLR LIGHT
                JNB SEN2,$
                CALL DELAY
                JB SEN1,$
                CALL DELAY
                JNB SEN1,$
                CALL DELAY
                AJMP UPP
UAPS: DEC VISITOR
```

```
                MOV A,VISITOR
                CJNE A,#00,UAPA
                CLR LIGHT
UAPA:           MOV R2,VISITOR
    MOV R1,#00H
                MOV R3,#00D
    MOV R4,#00D
    MOV R5,#00D
    MOV R6,#00D
    MOV R7,#00D
                CALL HEX2BCD
                MOV VALUE_2,R3
            MOV VALUE_1,R4
                JNB SEN2,$
            CALL DELAY
            JB SEN1,$
            CALL DELAY
            JNB SEN1,$
            CALL DELAY
                AJMP UPP

UP_COUNT:
                SETB LIGHT
                INC VISITOR
                MOV A,VISITOR
                CJNE A,#99,UPPS
                MOV VISITOR,#98
                JNB SEN1,$
                CALL DELAY
                JB SEN2,$
                CALL DELAY
                JNB SEN2,$
                CALL DELAY
                AJMP UPP
UPPS: MOV R2,VISITOR
    MOV R1,#00H
                MOV R3,#00D
    MOV R4,#00D
    MOV R5,#00D
    MOV R6,#00D
    MOV R7,#00D
                CALL HEX2BCD
                MOV VALUE_2,R3
            MOV VALUE_1,R4
                JNB SEN1,$
                CALL DELAY
```

```
        JB SEN2,$
        CALL DELAY
            JNB SEN2,$
            CALL DELAY
            AJMP UPP


;************************************************
HEX2BCD:
                MOV B,#10D
    MOV A,R2
    DIV AB
    MOV   R3,B          ;
    MOV   B,#10         ; R7,R6,R5,R4,R3
    DIV   AB
    MOV   R4,B
    MOV   R5,A
    CJNE R1,#0H,HIGH_BYTE   ; CHECK FOR HIGH BYTE
    SJMP ENDD

HIGH_BYTE:
                    MOV   A,#6
      ADD   A,R3
                    MOV   B,#10
                    DIV   AB
      MOV   R3,B
                    ADD   A,#5
      ADD   A,R4
                    MOV   B,#10
                    DIV   AB
      MOV   R4,B
                    ADD   A,#2
      ADD   A,R5
                    MOV   B,#10
                    DIV   AB
      MOV   R5,B
      CJNE R6,#00D,ADD_IT
      SJMP CONTINUE
ADD_IT:
      ADD A,R6
CONTINUE:
       MOV R6,A
       DJNZ R1,HIGH_BYTE
       MOV B, #10D
       MOV A,R6
       DIV AB
```

```
        MOV R6,B
        MOV R7,A
ENDD:   RET
;*************************************************************************
****
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&
;       7 SEGMENT DISPLAY ROUTINE
;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&
DISP:
        MOV R2,SPEED
        CJNE R2,#00H,AAS1
        CLR DIS_A
        CLR DIS_B
        CLR DIS_C
        CLR DIS_D
        CLR DIS_E
        CLR DIS_F
        SETB DIS_G
        RET
AAS1: CJNE R2,#01H,AS2
        CLR DIS_B
        CLR DIS_C
        SETB DIS_A
        SETB DIS_D
        SETB DIS_E
        SETB DIS_F
        SETB DIS_G
        RET
AS2:    CJNE R2,#02H,AS3
        CLR DIS_A
        CLR DIS_B
        CLR DIS_D
        CLR DIS_E
        CLR DIS_G
        SETB DIS_C
        SETB DIS_F
        RET
AS3:    CJNE R2,#03H,AS4
        CLR DIS_A
        CLR DIS_B
        CLR DIS_C
        CLR DIS_D
        CLR DIS_G
        SETB DIS_E
```

```
        SETB DIS_F
        RET
AS4:    CJNE R2,#04H,AS5
        CLR DIS_B
        CLR DIS_C
        CLR DIS_F
        CLR DIS_G
        SETB DIS_A
        SETB DIS_D
        SETB DIS_E
        RET
AS5:    CJNE R2,#05H,AS6
        CLR DIS_A
        CLR DIS_C
        CLR DIS_D
        CLR DIS_F
        CLR DIS_G
        SETB  DIS_B
        SETB DIS_E
        RET
AS6:    CJNE R2,#06H,AS7
        CLR DIS_A
        CLR DIS_C
        CLR DIS_D
        CLR DIS_E
        CLR DIS_F
        CLR DIS_G
        SETB DIS_B
        RET
AS7:    CJNE R2,#07H,AS8
        CLR DIS_A
        CLR DIS_B
        CLR DIS_C
        SETB DIS_D
        SETB DIS_E
        SETB DIS_F
        SETB DIS_G
        RET
AS8:    CJNE R2,#08H,AS9
        CLR DIS_A
        CLR DIS_B
        CLR DIS_C
        CLR DIS_D
        CLR DIS_E
        CLR DIS_F
        CLR DIS_G
```

```
        RET
AS9:    CJNE R2,#09H,AS10
        CLR DIS_A
        CLR DIS_B
        CLR DIS_C
        CLR DIS_D
        CLR DIS_F
        CLR DIS_G
        SETB DIS_E
        RET
AS10:   CJNE R2,#15H,AS11          ;symbol for -
        SETB DIS_A
        SETB DIS_B
        SETB DIS_C
        SETB DIS_D
        SETB DIS_E
        SETB DIS_F
        CLR DIS_G
        RET
AS11:   CJNE R2,#16H,AS12          ;switch off all disp
        SETB DIS_A
        SETB DIS_B
        SETB DIS_C
        SETB DIS_D
        SETB DIS_E
        SETB DIS_F
        SETB DIS_G
        RET
AS12:   MOV SPEED,#00H
        AJMP DISP
;************************************************************
;               INTRRUPT ROUTINE TO REFRESH THE DISPLAY
;************************************************************
REFRESH:
        PUSH   PSW          ; save current registerset
  MOV    PSW,#RB1
  PUSH   ACC
        INC COUNT
        MOV R4,COUNT
QA1:    CJNE R4,#01H,QA2
        MOV SPEED,VALUE_1
        SETB DIS1
        CLR DIS2
        CALL DISP
        AJMP DOWN
QA2:    CJNE R4,#02H,QA3
```

```
        MOV SPEED,VALUE_2
        CLR DIS1
        SETB DIS2
        CALL DISP
        AJMP DOWN
QA3:    MOV COUNT,#01H
        MOV R4,COUNT
        AJMP QA1
DOWN:       MOV TL0,#0FFH
        MOV TH0,#0F0H
        POP    ACC
  POP    PSW
        RETI
;*************************************************************
DELAY:
        MOV R1,#4FH
REP2:   MOV R2,#0FFH
REP1:   NOP
        DJNZ R2,REP1
        DJNZ R1,REP2
        RET
;*************************************************************



END
```