DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Subject: Microprocessor and Microcontroller Based Design

SUBMITTED TO:
Dr Saghir Khan
LE Ayesha

SUBMITTED BY:
Waseem Ghulam (409600)
Ameer Hamza (409662)
Waqar Ahmed (398417)
Muskan Fatima (414875)

LAB # 11: Use of Timer modules to program PIC-18

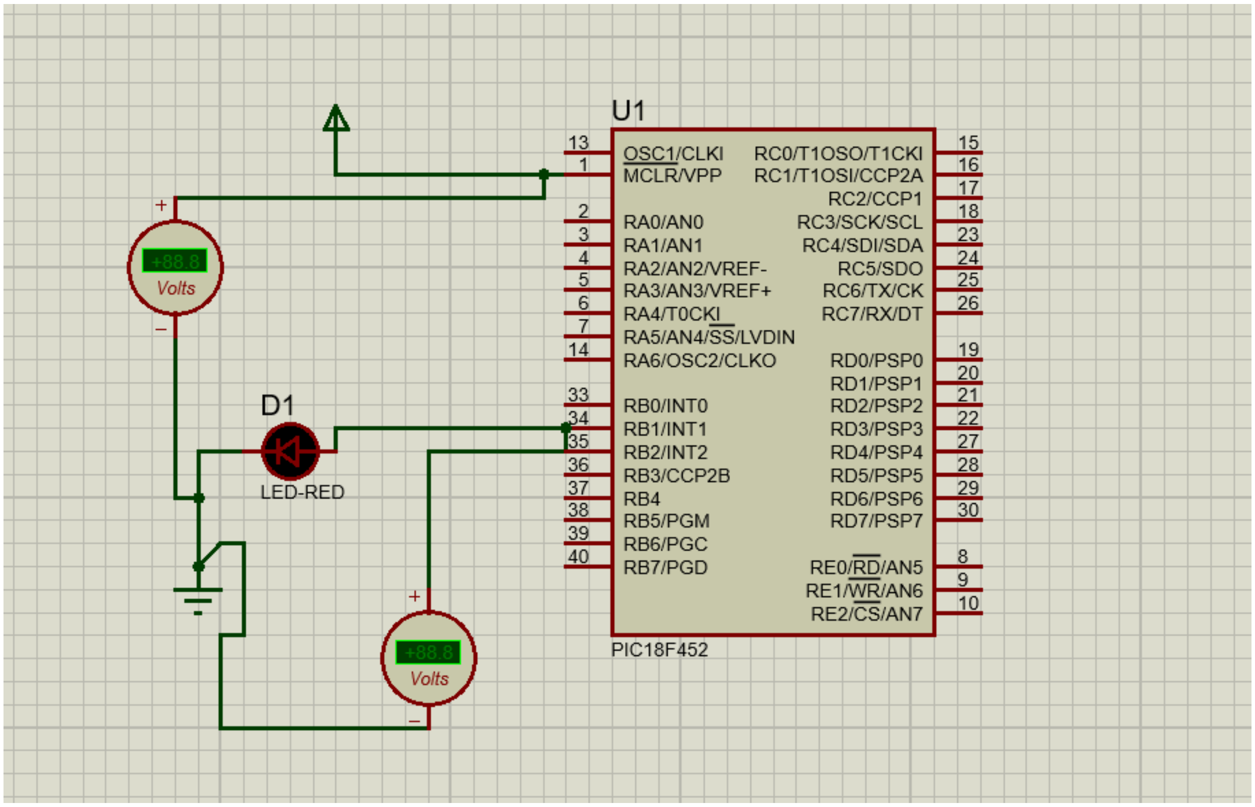
Task #01:

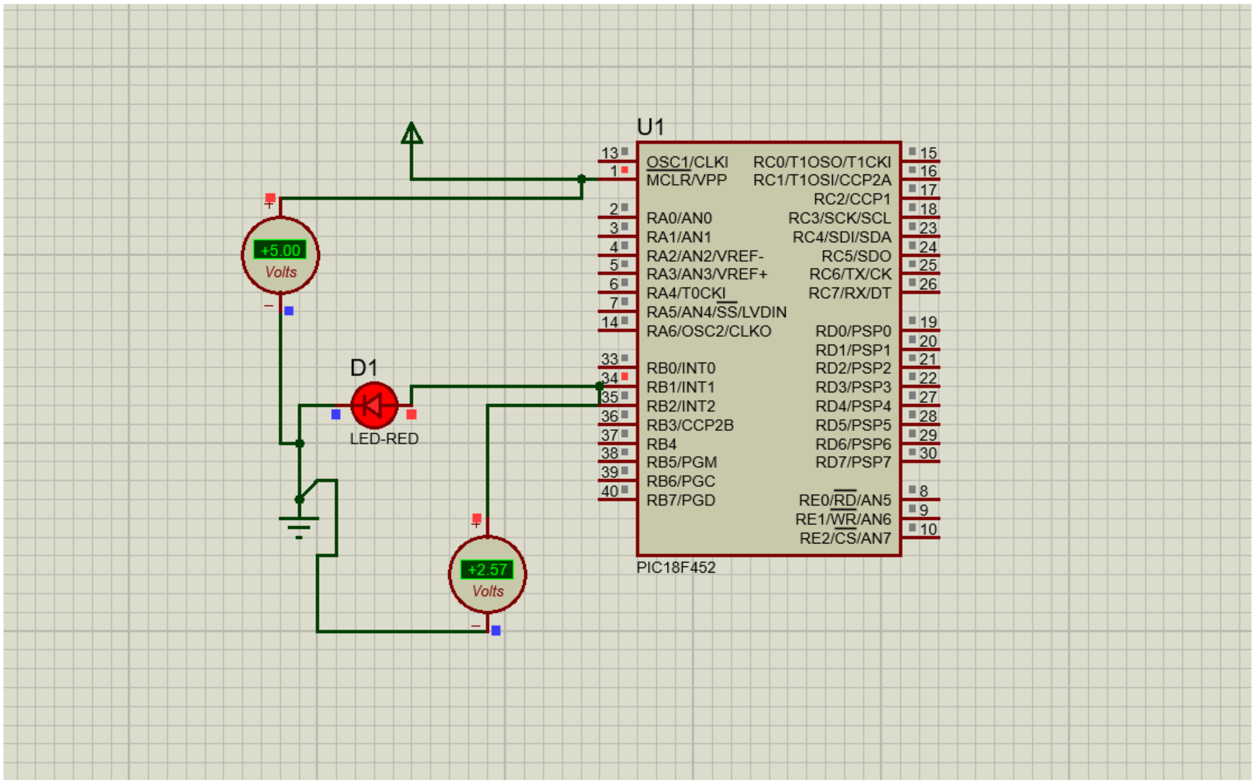
Code:

```
INCLUDE <p18f452.inc>      ; Include the PIC18F452
header file

; Define file registers for delay counters
COUNT1      EQU 0x20       ; Outer loop counter
COUNT2      EQU 0x21       ; Inner loop counter

    ORG 0x0000             ; Reset vector
    GOTO MAIN              ; Jump to the main program

MAIN:
    ; Configure RB1 as output
    CLRF PORTB             ; Clear PORTB
    BCF TRISB, 1           ; Set RB1 as output (clear
bit 1 of TRISB)

TOGGLE_LOOP:
    ; Toggle RB1
    BSF LATB, 1            ; Turn RB1 ON
    CALL DELAY_1S          ; Wait for 1 second
    BCF LATB, 1            ; Turn RB1 OFF
    CALL DELAY_1S          ; Wait for 1 second
    GOTO TOGGLE_LOOP       ; Repeat

; 1-second delay subroutine
DELAY_1S:
    MOVLW D'250'           ; Outer loop (repeat 250
times)
    MOVWF COUNT1

DELAY_LOOP1:
    MOVLW D'250'           ; Inner loop (repeat 250
times)
    MOVWF COUNT2
```

```
DELAY_LOOP2:
    NOP                         ; No operation (short
delay)
    DECFSZ COUNT2, F            ; Decrement inner loop
counter
    GOTO DELAY_LOOP2            ; Repeat inner loop
    DECFSZ COUNT1, F            ; Decrement outer loop
counter
    GOTO DELAY_LOOP1            ; Repeat outer loop
    RETURN                      ; Return after 1-second
delay

    END                         ; End of program
```

Proteus:



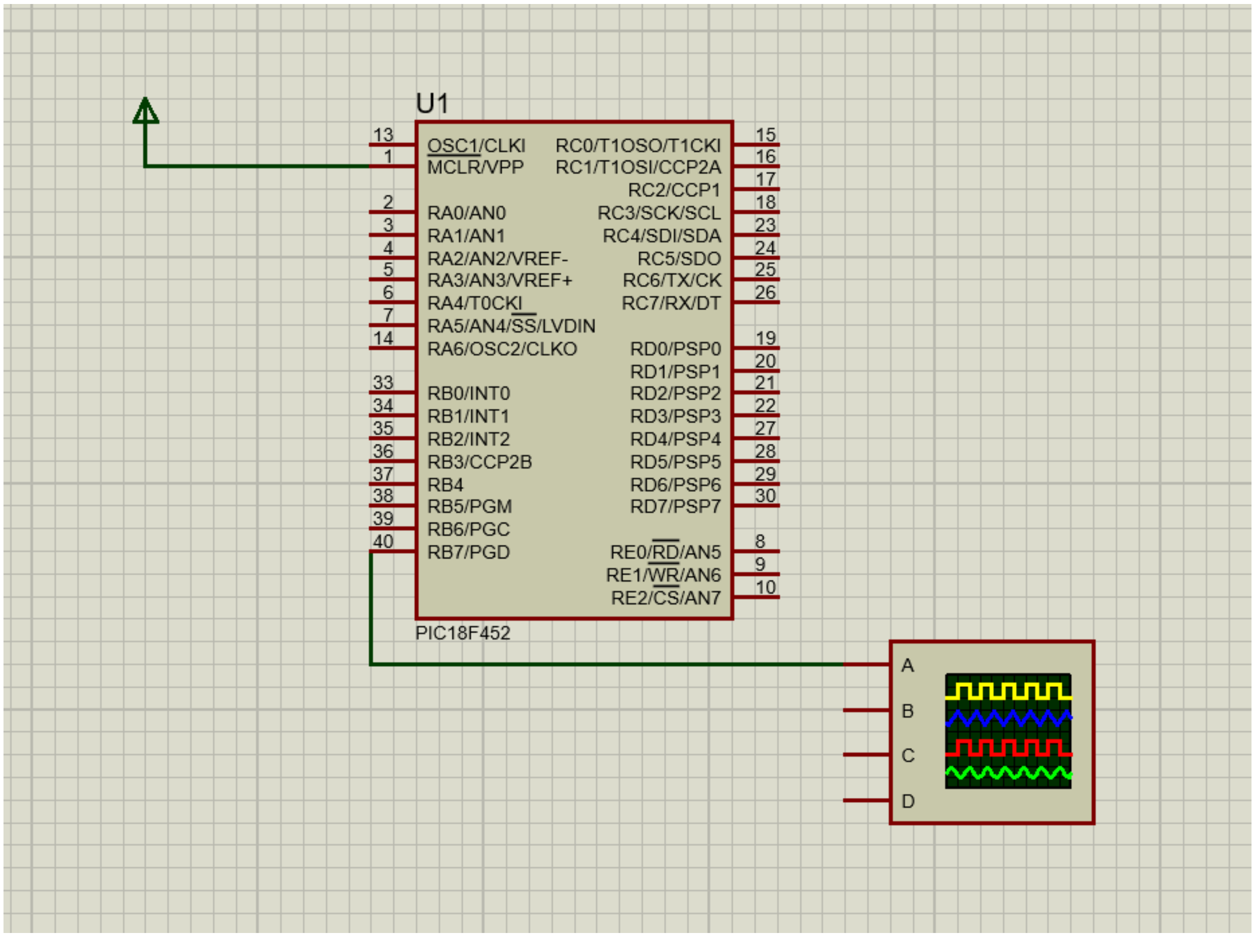


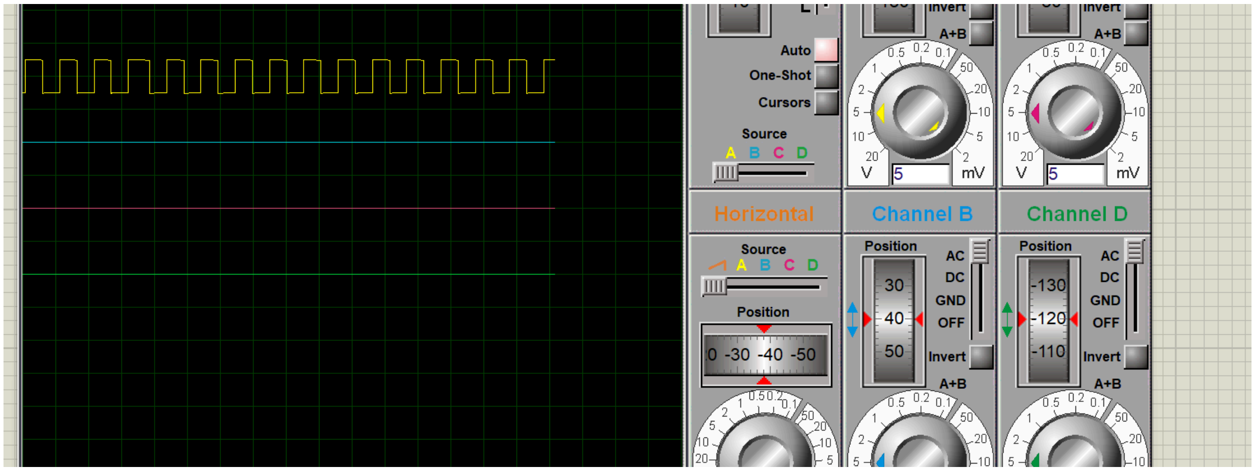
Task #02:

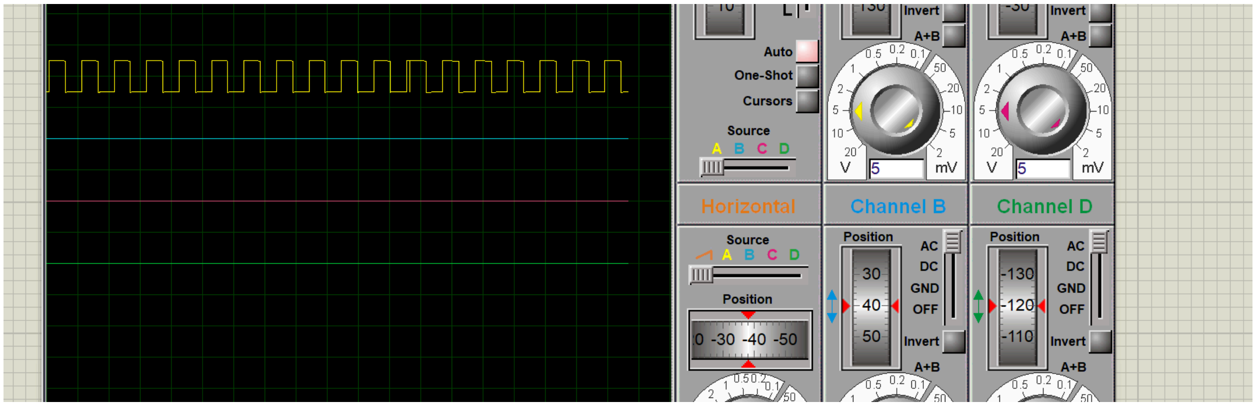
Code:

```
INCLUDE <p18f452.inc>       ; Include the PIC18F452
header file

; Define constants and file registers
COUNT1      EQU 0x20        ; Outer loop counter for
delay
COUNT2      EQU 0x21        ; Inner loop counter for
delay

    ORG 0x0000              ; Reset vector
    GOTO MAIN               ; Jump to the main program

MAIN:
    ; Configure RB7 as output
    CLRF PORTB            ; Clear PORTB
    BCF TRISB, 7          ; Set RB7 as output (clear
bit 7 of TRISB)

SQUARE_LOOP:
    BSF LATB, 7           ; Set RB7 high (start of
square wave)
    CALL DELAY_60HZ       ; Delay for half-period (ON
time)
    BCF LATB, 7           ; Set RB7 low (end of
square wave)
    CALL DELAY_60HZ       ; Delay for half-period
(OFF time)
    GOTO SQUARE_LOOP      ; Repeat indefinitely

; ******************************
; DELAY_60HZ Subroutine
; Generates a delay of 8.33 ms (half-period of 60 Hz)
; XTAL = 12 MHz, Timer0 with Prescaler = 64
; ******************************
DELAY_60HZ:
    ; Configure Timer0
    CLRF TMR0L            ; Clear Timer0 Low register
    CLRF TMR0H            ; Clear Timer0 High
register
    MOVLW B'10000100'     ; Timer0: 16-bit mode,
prescaler = 64
    MOVWF T0CON           ; Load Timer0 control
register

    ; Load Timer0 registers for 8.33 ms
    ; Timer count = 65536 - (Delay / Clock Period) /
Prescaler
    ; = 65536 - (8.33ms / 0.33us) / 64
    ; = 65536 - 395
    MOVLW 0xF9            ; Load high byte of 395
(65536 - 395 = F9A5)
    MOVWF TMR0H           ; High byte
    MOVLW 0xA5            ; Load low byte of 395
    MOVWF TMR0L           ; Low byte
    BSF T0CON, TMR0ON       ; Turn on Timer0

WAIT_TMR0:
    BTFSS INTCON, TMR0IF    ; Check Timer0 overflow
flag
    GOTO WAIT_TMR0          ; Wait until Timer0
overflows

    BCF INTCON, TMR0IF      ; Clear Timer0 overflow
flag
    RETURN                  ; Return to the main
program

    END                     ; End of program
```

Proteus (Result/Output):


U1
13 OSC1/CLKI
1 MCLR/VPP
2 RA0/AN0
3 RA1/AN1
4 RA2/AN2/VREF-
5 RA3/AN3/VREF+
6 RA4/T0CKI
7 RA5/AN4/SS/LVDIN
14 RA6/OSC2/CLKO
33 RB0/INT0
34 RB1/INT1
35 RB2/INT2
36 RB3/CCP2B
37 RB4
38 RB5/PGM
39 RB6/PGC
40 RB7/PGD
RC0/T1OSO/T1CKI 15
RC1/T1OSI/CCP2A 16
RC2/CCP1 17
RC3/SCK/SCL 18
RC4/SDI/SDA 23
RC5/SDO 24
RC6/TX/CK 25
RC7/RX/DT 26
RD0/PSP0 19
RD1/PSP1 20
RD2/PSP2 21
RD3/PSP3 22
RD4/PSP4 27
RD5/PSP5 28
RD6/PSP6 29
RD7/PSP7 30
RE0/RD/AN5 8
RE1/WR/AN6 9
RE2/CS/AN7 10
PIC18F452
A
B
C
D


Auto
One-Shot
Cursors
Source
A B C D
Invert
A+B
0.5 0.2 0.1
1 50
2 20
5 10
10 5
20 2
V 5 mV
Horizontal
Channel B
Channel D
Position
AC
DC
GND
OFF
30
40
50
-130
-120
-110
0 -30 -40 -50

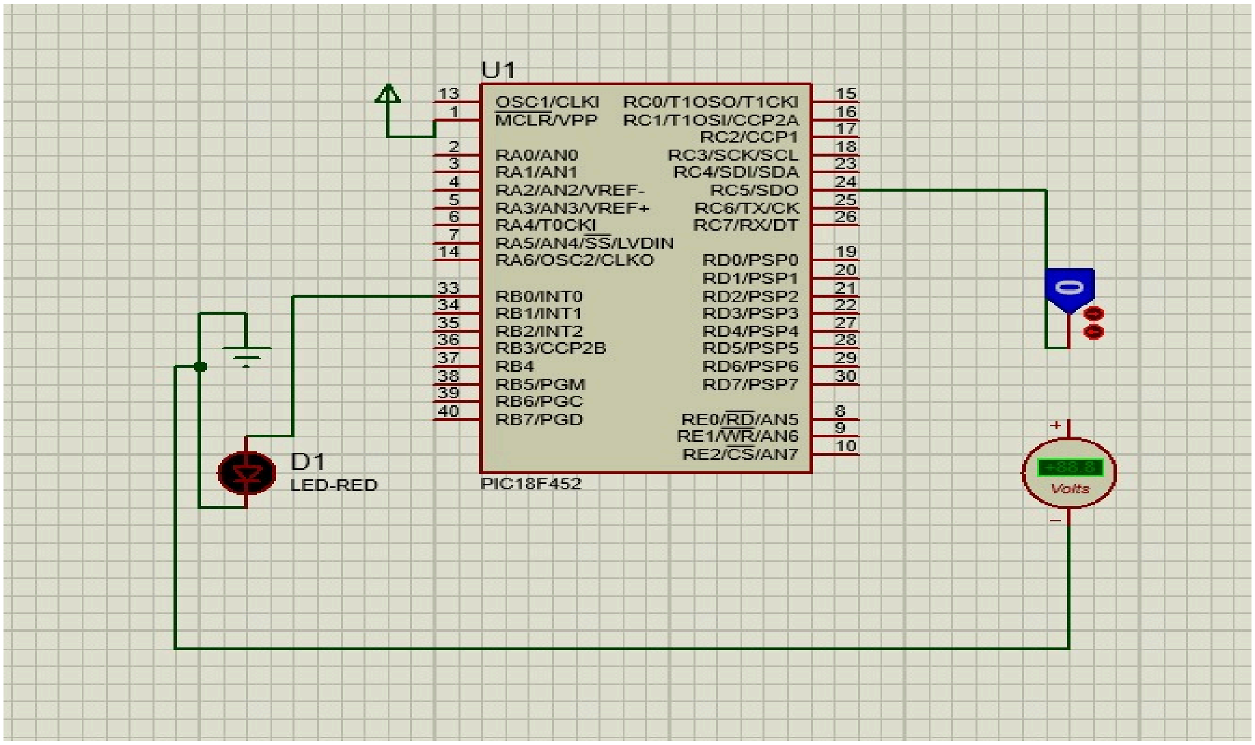


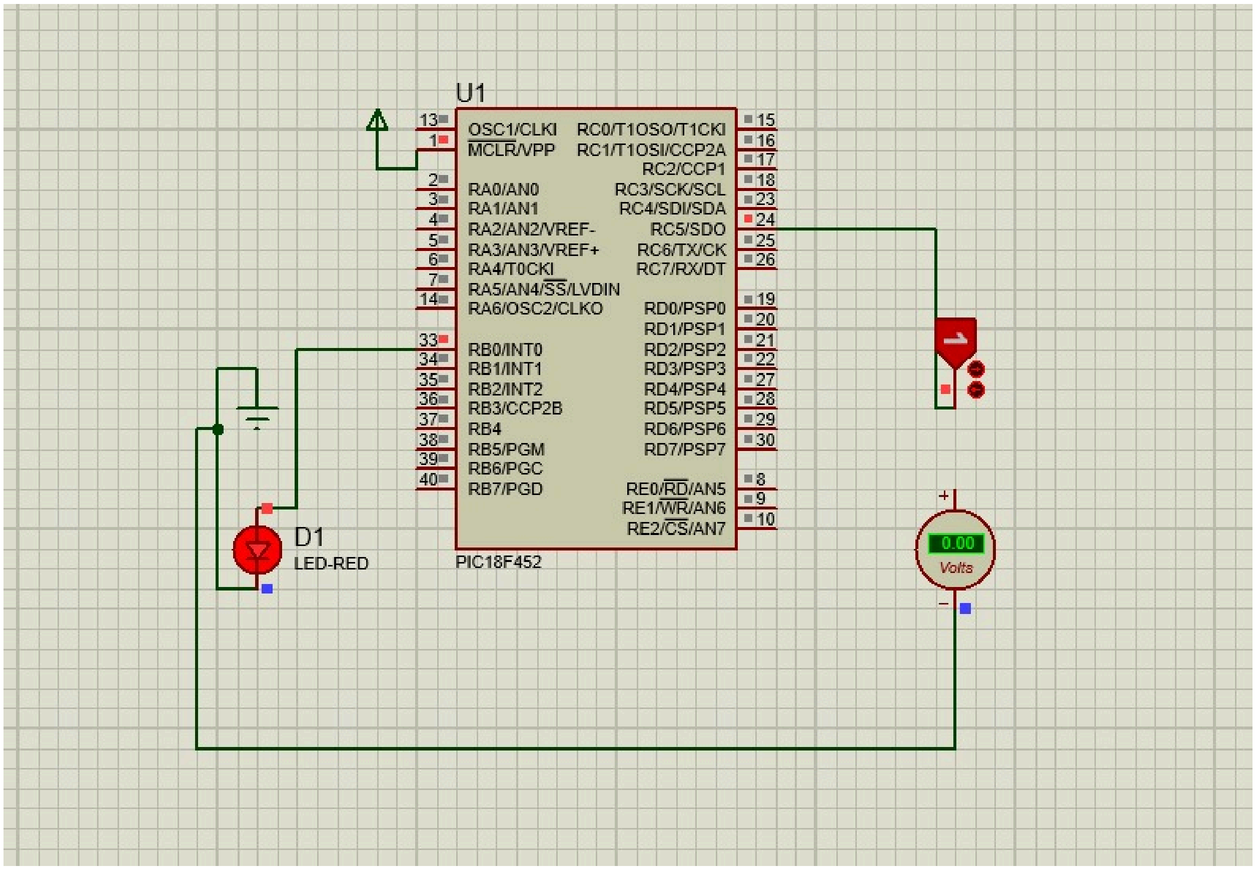
Task #03

Code:

```
LIST    P=18F452
    INCLUDE <P18F452.INC>

    ORG     0x0000                  ; Reset vector
    GOTO    MAIN                    ; Jump to main program

    CBLOCK  0x20                   ; Start of General Purpose
RAM
        OVERFLOW_COUNT             ; Reserve 1 byte for
overflow counter
    ENDC

MAIN:
    ; ====== Initialization ======
    CLRF    PORTB                   ; Clear PORTB
    CLRF    LATB                    ; Clear output latch
    BCF     TRISB, 0                ; Set RB0 (LED) as output
    BSF     TRISC, 5                ; Set RC5 as input

    MOVLW   0x00                    ; Configure Timer0:
16-bit mode, no prescaler
    MOVWF   T0CON
    CLRF    TMR0H                   ; Clear Timer0 high byte
    CLRF    TMR0L                ; Clear Timer0 low byte
    CLRF    OVERFLOW_COUNT       ; Clear overflow counter

WAIT_SIGNAL:
    BTFSS   PORTC, 5             ; Check if RC5 is HIGH
    GOTO    RESET_TIMER          ; If LOW, reset Timer0
    BSF     T0CON, TMR0ON        ; Start Timer0

CHECK_OVERFLOW:
    BTFSS   INTCON, TMR0IF       ; Wait for Timer0
overflow
    GOTO    CHECK_OVERFLOW
    BCF     INTCON, TMR0IF       ; Clear Timer0 overflow
flag
    INCF    OVERFLOW_COUNT, F ; Increment overflow
counter

    MOVLW   0x0A                 ; Compare overflow
counter to 10
    CPFSGT  OVERFLOW_COUNT       ; Skip if overflow
counter >= 10
    GOTO    WAIT_SIGNAL          ; Continue monitoring

    ; ====== Overflow Count Reached ======
    BSF     LATB, 0              ; Turn on LED at RB0
    GOTO    WAIT_SIGNAL          ; Keep monitoring PORTC5

RESET_TIMER:
    BCF     T0CON, TMR0ON        ; Stop Timer0
    CLRF    TMR0H                ; Clear Timer0 high byte
    CLRF    TMR0L                ; Clear Timer0 low byte
    CLRF    OVERFLOW_COUNT       ; Reset overflow counter
    BCF     LATB, 0              ; Turn off LED
    GOTO    WAIT_SIGNAL          ; Go back to monitoring

    END                          ; End of program
```

Proteus:


U1
13 OSC1/CLKI
1 MCLR/VPP
2 RA0/AN0
3 RA1/AN1
4 RA2/AN2/VREF-
5 RA3/AN3/VREF+
6 RA4/T0CKI
7 RA5/AN4/SS/LVDIN
14 RA6/OSC2/CLKO
33 RB0/INT0
34 RB1/INT1
35 RB2/INT2
36 RB3/CCP2B
37 RB4
38 RB5/PGM
39 RB6/PGC
40 RB7/PGD
RC0/T1OSO/T1CKI 15
RC1/T1OSI/CCP2A 16
RC2/CCP1 17
RC3/SCK/SCL 18
RC4/SDI/SDA 23
RC5/SDO 24
RC6/TX/CK 25
RC7/RX/DT 26
RD0/PSP0 19
RD1/PSP1 20
RD2/PSP2 21
RD3/PSP3 22
RD4/PSP4 27
RD5/PSP5 28
RD6/PSP6 29
RD7/PSP7 30
RE0/RD/AN5 8
RE1/WR/AN6 9
RE2/CS/AN7 10
PIC18F452
D1
LED-RED
0
+
+88.8
Volts
−



Task #04:

Design:

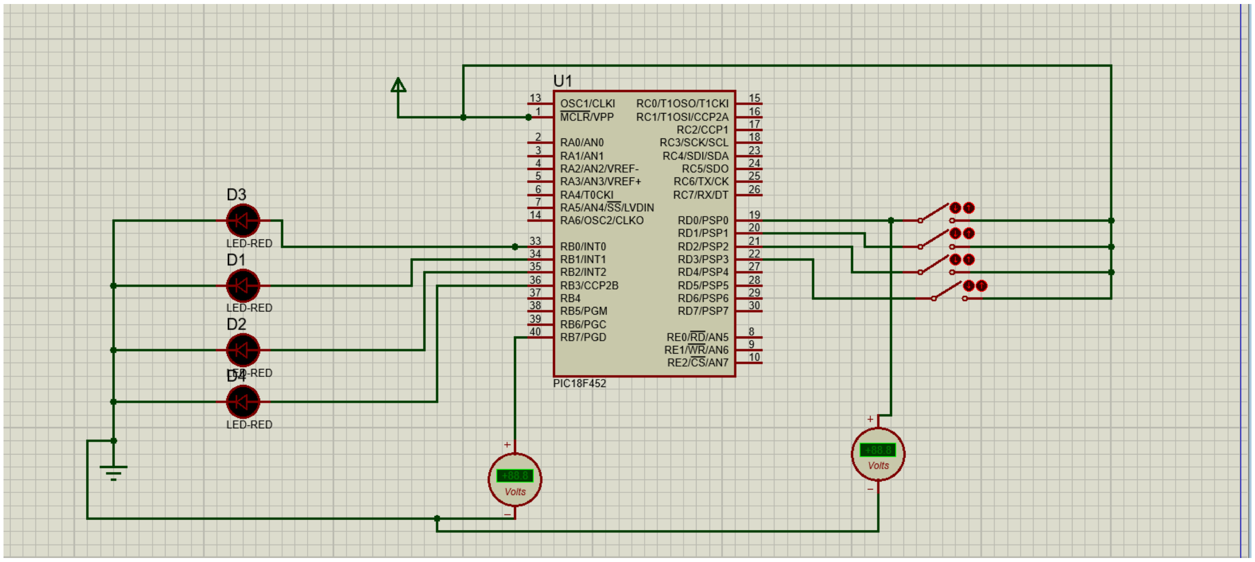
For the number of input on PORTD , the corresponding number of LEDs will turn on on PORTB.

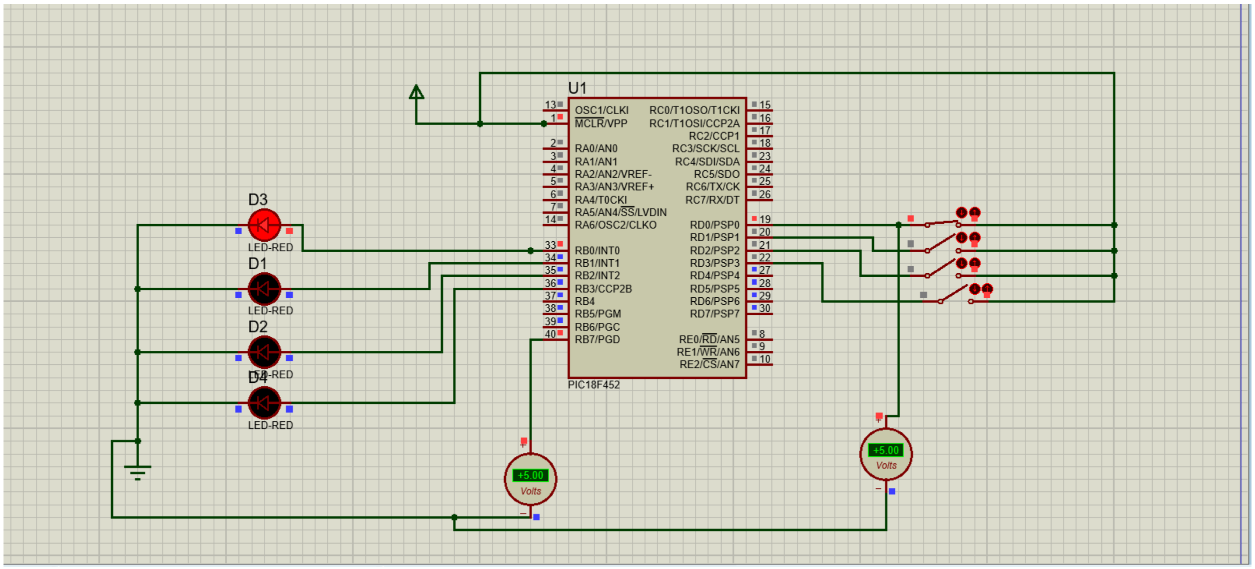
Code:

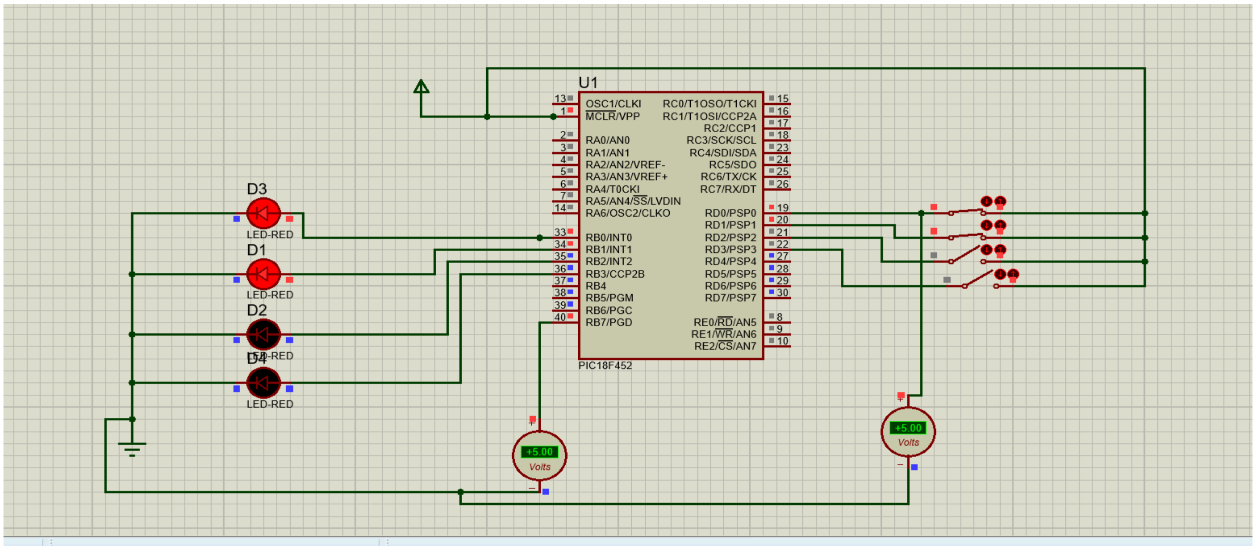
```
INCLUDE <p18f452.inc>       ; Include header file for
PIC18F452

; Define constants
INPUT_MASK      EQU 0x0F     ; Mask for 4-bit input
(00001111)

; *****************************
; Main Program
; *****************************
                ORG 0x0000    ; Program start address
                GOTO MAIN     ; Jump to main program

MAIN:
                ; Initialization
                CLRF PORTB    ; Clear PORTB (turn off
all LEDs)
                MOVLW 0x00    ; Set all PORTB pins as
output
                MOVWF TRISB   ; Configure PORTB as
output (RB7 will be output)
                MOVLW 0x0F    ; Set lower 4 bits of
PORTD as input
                MOVWF TRISD   ; Configure RD0-RD3 as
input

                ; Enable pull-ups for PORTD (if
needed)
                MOVLW 0xF0     ; Enable pull-ups on
RD0-RD3 (if needed)
                MOVWF INTCON2

MAIN_LOOP:
                ; Read 4-bit input from PORTD
                MOVF PORTD, W   ; Move PORTD value
into WREG
                ANDLW INPUT_MASK ; Mask only lower 4
bits (RD0-RD3)

                ; *** DEBUGGING: Display PORTD value
on PORTB ***
                MOVWF PORTB    ; Display PORTD value
on PORTB for debugging

                ; *** DEBUGGING: Blink an LED to
indicate loop execution ***
                BSF PORTB, 7   ; Turn on an indicator
LED (e.g., RB7)
                CALL DELAY     ; Short delay
                BCF PORTB, 7   ; Turn off the
indicator LED

                CALL LIGHT_UP  ; Call subroutine to
light up LEDs
                GOTO MAIN_LOOP ; Repeat the process

; *****************************
; Subroutine: LIGHT_UP
; Inputs: WREG contains the 4-bit value
; Outputs: Lights up the corresponding LEDs on PORTB
; *****************************
LIGHT_UP:
                CLRF PORTB      ; Clear PORTB before
lighting LEDs
                MOVF WREG, W    ; Copy the value from
WREG to WREG again

                ; Check the bits in WREG and turn on
corresponding LEDs
                BTFSS WREG, 0   ; Check bit 0
                BSF PORTB, 0    ; If set, light up LED
0
                BTFSS WREG, 1   ; Check bit 1
                BSF PORTB, 1    ; If set, light up LED
1
                BTFSS WREG, 2   ; Check bit 2
                BSF PORTB, 2    ; If set, light up LED
2
                BTFSS WREG, 3   ; Check bit 3
                BSF PORTB, 3    ; If set, light up LED
3

                RETURN

; *****************************
; Delay Subroutine
; *****************************
DELAY:
```

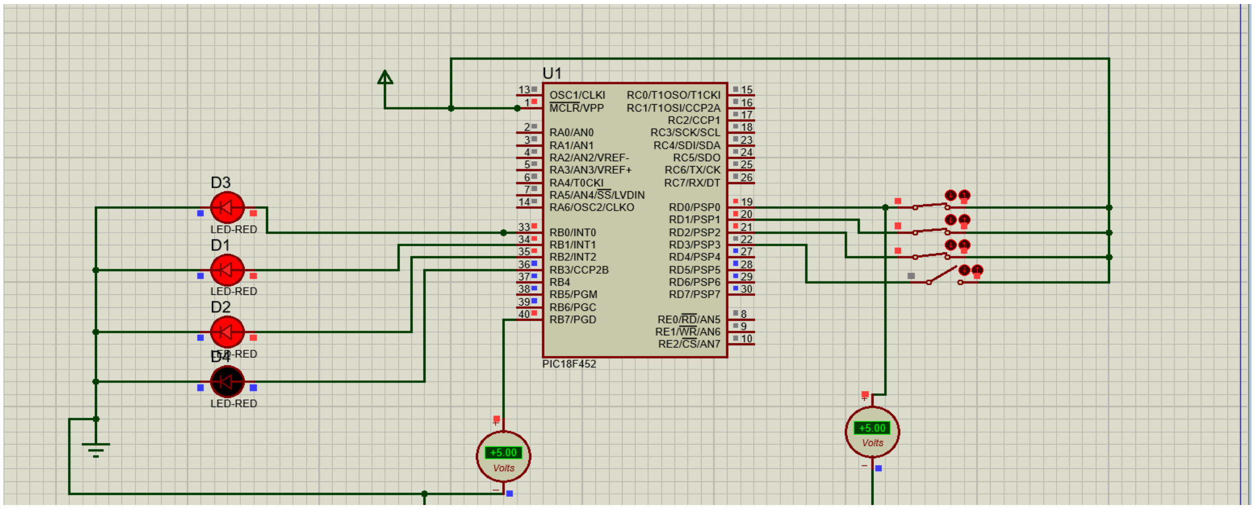
```
                MOVLW d'255'    ; Load WREG with a
value
                MOVWF COUNT1   ; Move the value to
COUNT1

DELAY_LOOP1:
                DECFSZ COUNT1, F ; Decrement COUNT1,
skip next if 0
                GOTO DELAY_LOOP1 ; Loop until COUNT1
is 0

                RETURN          ; Return from
subroutine

; *****************************
; Variables
; *****************************
COUNT1          EQU 0x20     ; Define a variable named
COUNT1

; End of Program
                END
```

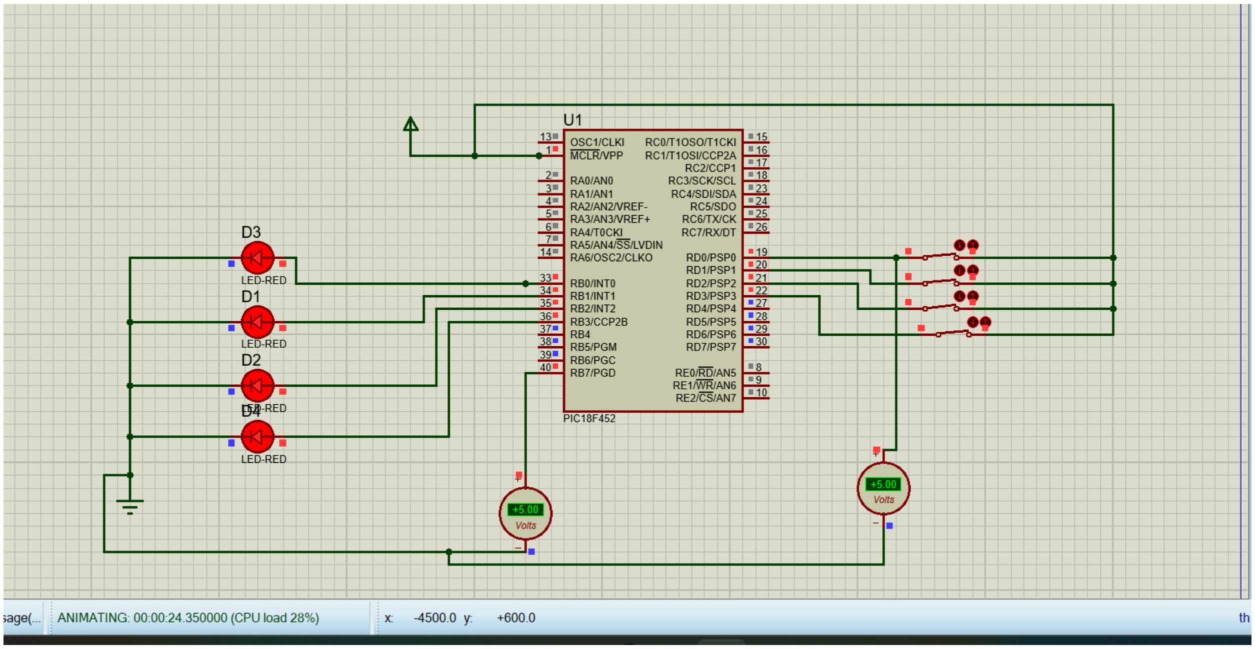
Proteus (Output/Result):


U1
OSC1/CLKI
MCLR/VPP
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS/LVDIN
RA6/OSC2/CLKO
RB0/INT0
RB1/INT1
RB2/INT2
RB3/CCP2B
RB4
RB5/PGM
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2A
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
RE0/RD/AN5
RE1/WR/AN6
RE2/CS/AN7
PIC18F452
D3
LED-RED
D1
LED-RED
D2
LED-RED
D4
LED-RED
Volts
Volts


U1
OSC1/CLKI
MCLR/VPP
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS/LVDIN
RA6/OSC2/CLKO
RB0/INT0
RB1/INT1
RB2/INT2
RB3/CCP2B
RB4
RB5/PGM
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2A
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
RE0/RD/AN5
RE1/WR/AN6
RE2/CS/AN7
PIC18F452
D3
LED-RED
D1
LED-RED
D2
LED-RED
D4
LED-RED
+5.00
Volts
+5.00
Volts







sage(... ANIMATING: 00:00:24.350000 (CPU load 28%) x: -4500.0 y: +600.0 th