Interrupt handling with PIC Microcontroller

Task1

Team names

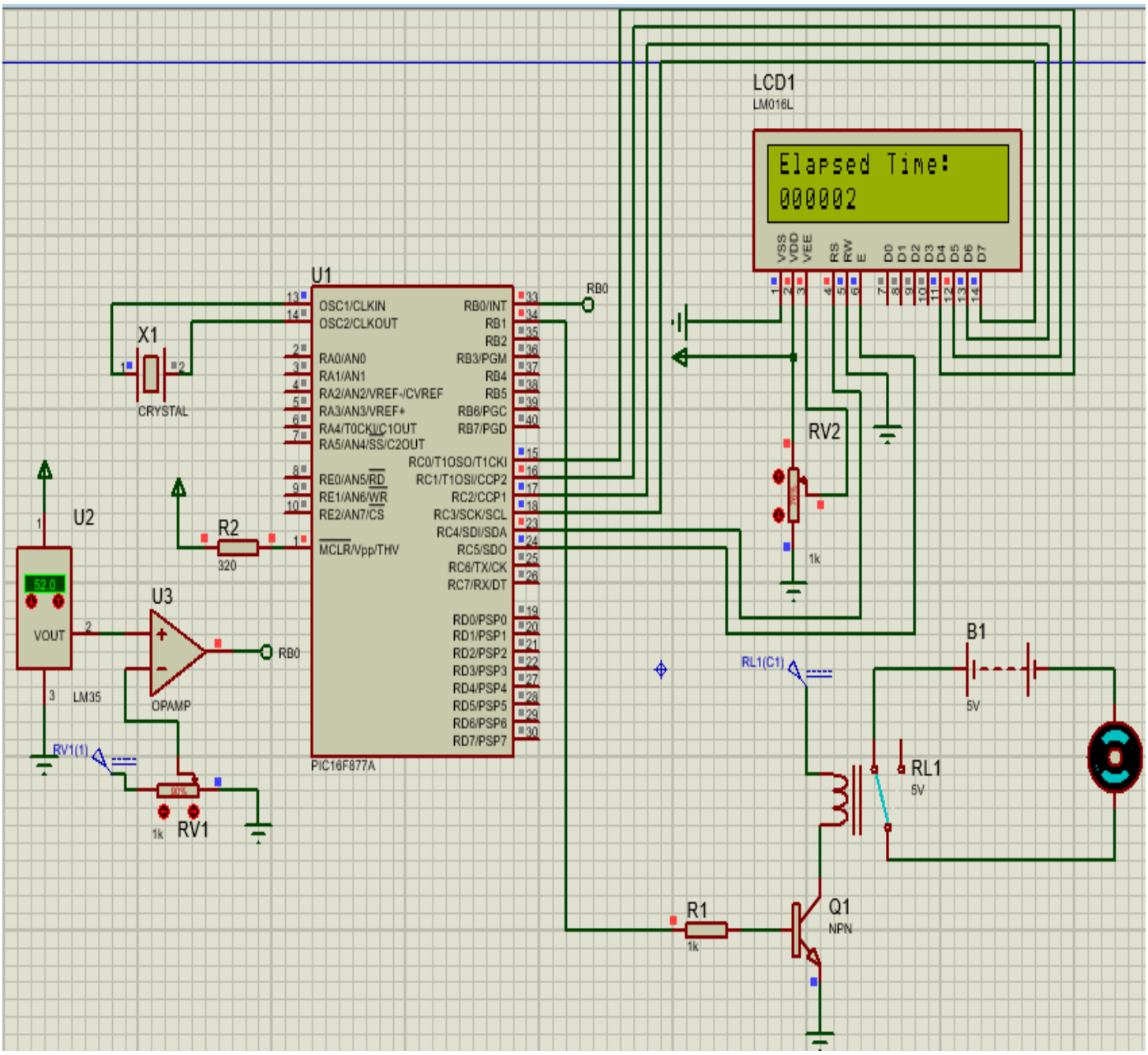
١- احمد محمد عبدالمنعم محمد سكشن ١

٢- احمد محمد عربي اسماعيل حسن سكشن ١

٣- ابراهيم محمود محمد السيد سكشن ١

٤- محمد سعيد محمد مرسي فرج سكشن ٣

Simulation


LCD1
LM016L
Elapsed Time:
000002
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1
OSC1/CLKIN
OSC2/CLKOUT
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
MCLR/Vpp/THV
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
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
PIC16F877A
RB0
X1
CRYSTAL
R2
320
RV2
1k
U2
52.0
VOUT
LM35
U3
OPAMP
RB0
RV1(1)
1k
RV1
B1
5V
RL1(C1)
RL1
5V
R1
1k
Q1
NPN

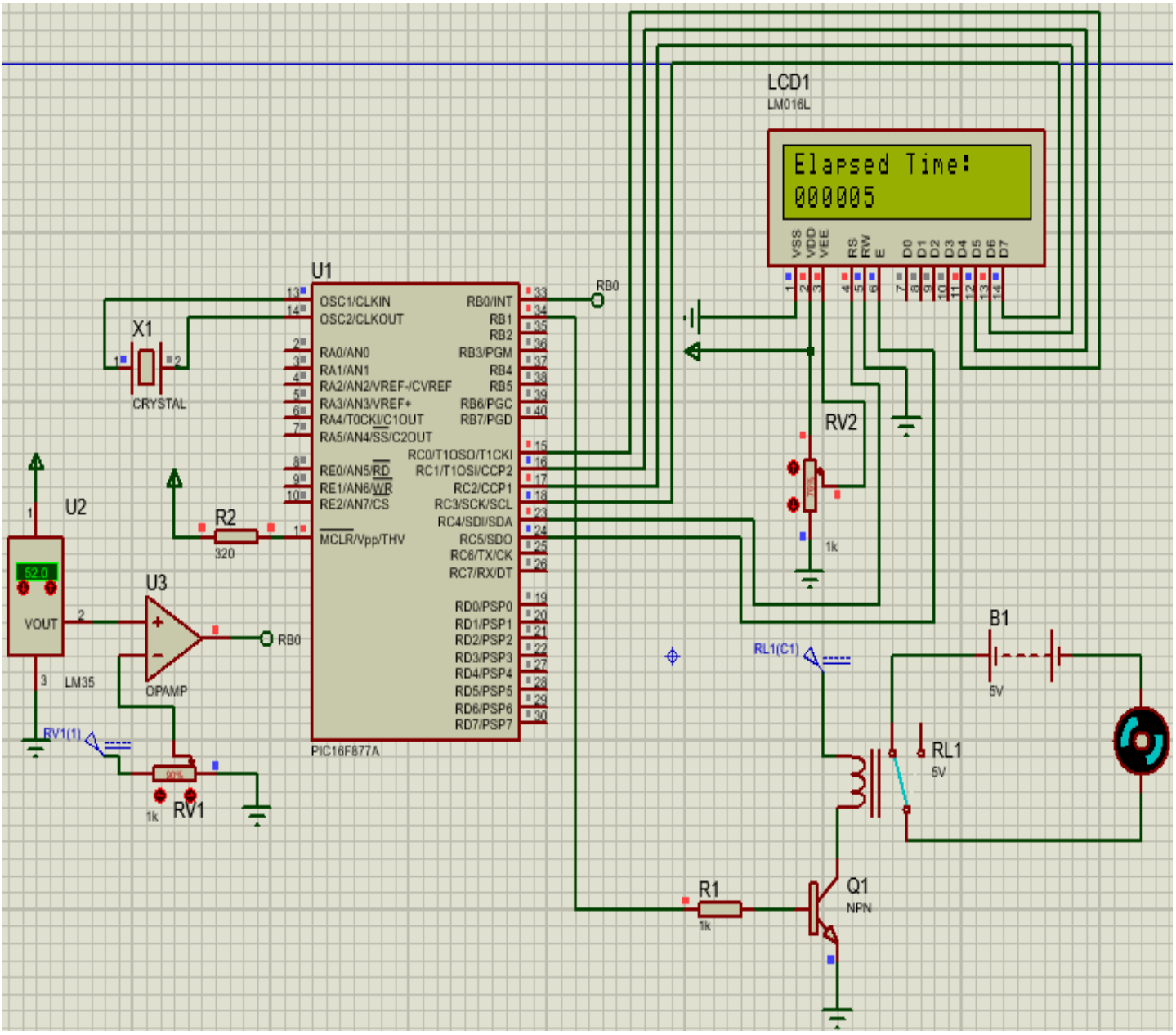
Code

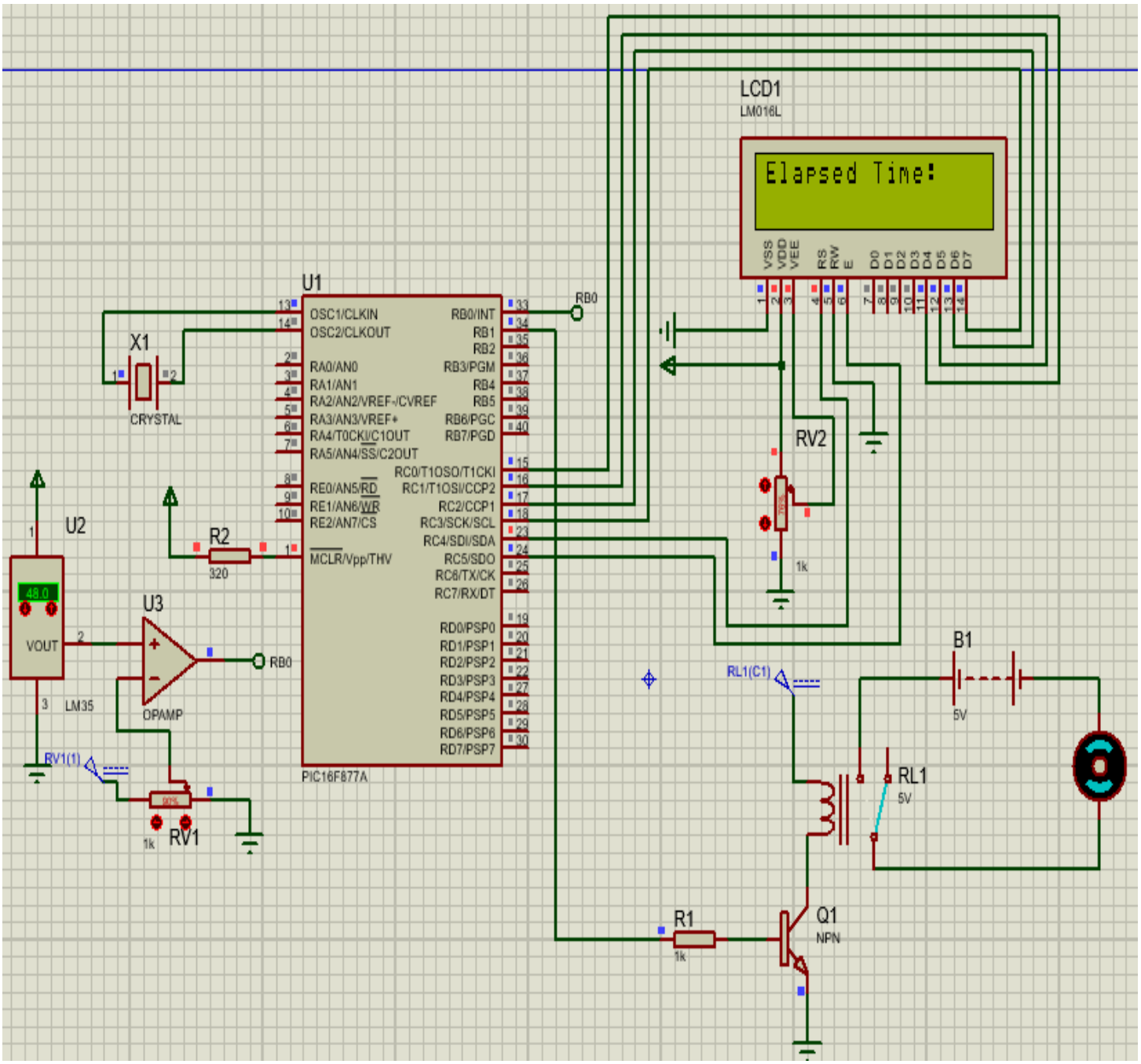
```
sbit LCD_RS at RC4_bit;
sbit LCD_EN at RC5_bit;
sbit LCD_D7 at RC3_bit;
sbit LCD_D6 at RC2_bit;
sbit LCD_D5 at RC1_bit;
sbit LCD_D4 at RC0_bit;

sbit LCD_RS_Direction at TRISC4_bit;
sbit LCD_EN_Direction at TRISC5_bit;
sbit LCD_D7_Direction at TRISC3_bit;
sbit LCD_D6_Direction at TRISC2_bit;
sbit LCD_D5_Direction at TRISC1_bit;
sbit LCD_D4_Direction at TRISC0_bit;
char txt[7];
int time, counter = 0;
void initialize_timer_interrupt(void){
  TMR0IE_bit = 1;
  GIE_bit = 1;
  T0CS_bit = 0;
  PSA_bit = 0;
  PS0_bit = 0;
  PS1_bit = 0;
  PS2_bit = 1;
  TMR0 = 6;
}
void main(){
```

```
  TRISB.B0 = 1;
  TRISB.B1 = 0;
  INTE_bit = 1;
  Lcd_Init();
  LCD_Cmd(_LCD_CURSOR_OFF);
  Lcd_Out(1, 1, "Elapsed Time: ");
  initialize_timer_interrupt();
  if(PORTB.B0 == 1){
    PORTB.B1 = 1;
  }
  else if(PORTB.B0 == 0){
    PORTB.B1 = 0;
  }
  while(1){
    if(counter == 250 && PORTB.B1 == 1){
      counter = 0;
      time++;
      IntToStrWithZeros(time, txt);
      Lcd_Out(2, 1, txt);
    }
    if(PORTB.B0 == 0){
        PORTB.B1 = 0;
        counter = 0;
        time = 0;
    }
  }
}
void interrupt(void){
```

```
    if(INTCON.TMR0IF == 1){
        counter++;
        INTCON.TMR0IF = 0;
        TMR0 = 6;
     }
    if(INTF_bit == 1){
        PORTB.B1 = 1;
        INTF_bit = 0;
    }
}
```

Screenshots




LCD1
LM016L
Elapsed Time:
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1
OSC1/CLKIN
OSC2/CLKOUT
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
MCLR/Vpp/THV
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
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
PIC16F877A
RB0
X1
CRYSTAL
RV2
1k
R2
320
U2
48.0
VOUT
LM35
U3
OPAMP
RB0
RV1(1)
1k
RV1
B1
5V
RL1(C1)
RL1
5V
R1
1k
Q1
NPN

Calculations & Notes

- **We used a crystal with frequency 8MHz. So, if we want to increment the counter every second we will make 250 interrupts.**
- **We also used pre scalar of 32.**

- **Timer interrupt delay Time =** $\frac{Prescalar*(256-TMR0)}{(FOSC/4)}$

$= \frac{32*(256-6)}{(8*10^6)/4} = 4 * 10^{-3}$ **seconds**

No. Interrupts = $\frac{1}{4* 10^{-3}} = 250$

- **While we are doing this task on Proteus, we found that the condition which checks if counter equals 250 does not increment the counter with 1 second. It takes more than 1 second to increment the counter. We think that it is normal to have like this problem with proteus due to our machine processor.**