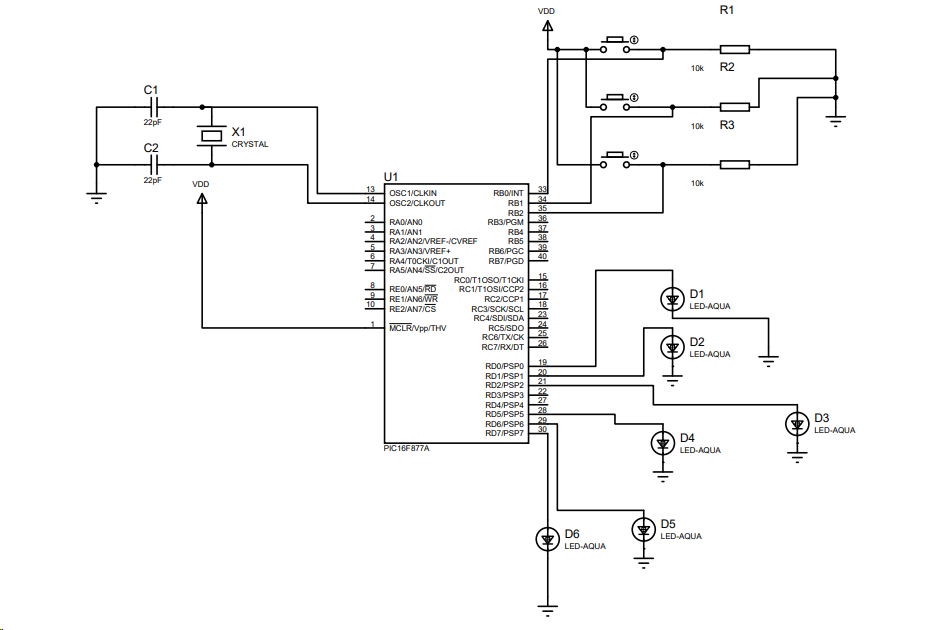
**Task1:**

**Description:**

1. RB0, RB1, RB2 – Configured as Inputs
2. RD0,1,2,5,67 – Configured as Outputs
3. As Long as RB0 is pressed it is stuck in While loop toggling the RD0 and RD1 every 1 Second
4. As Long as RB1 is pressed it is stuck in While loop toggling the RD7 and RD6 every 1 Second
5. As Long as RB2 is pressed it is stuck in While loop toggling the RD0,RD1,RD2,RD5,RD6,RD7 every 1 Second

**Schematic Diagram**



**Code:**

*void main() {*

*TRISB = 0x07; //RB0, RB1, RB2 as Input Pins*

*TRISD = 0x00; //All the Pins in Port configured as Output*

*PORTD = 0x00;*

*while(1)*

*{*

*PORTD =0x00;*

*while(!(PORTB & 0x1) )*

*{*

*PORTD ^= 0x3;*

*Delay\_ms(1000);*

*}*

*while(!(PORTB & 0x2) )*

*{*

*PORTD ^= 0xC0;*

*Delay\_ms(1000);*

*}*

*while(!(PORTB & 0x4) )*

*{*

*PORTD ^= 0xE7;*

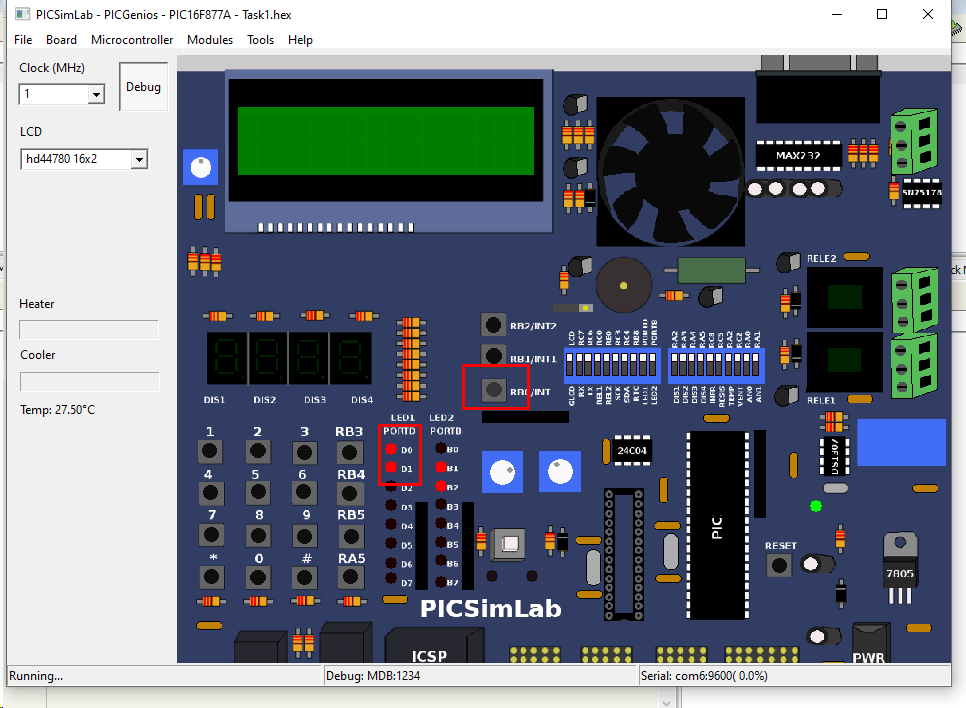
*Delay\_ms(1000);*

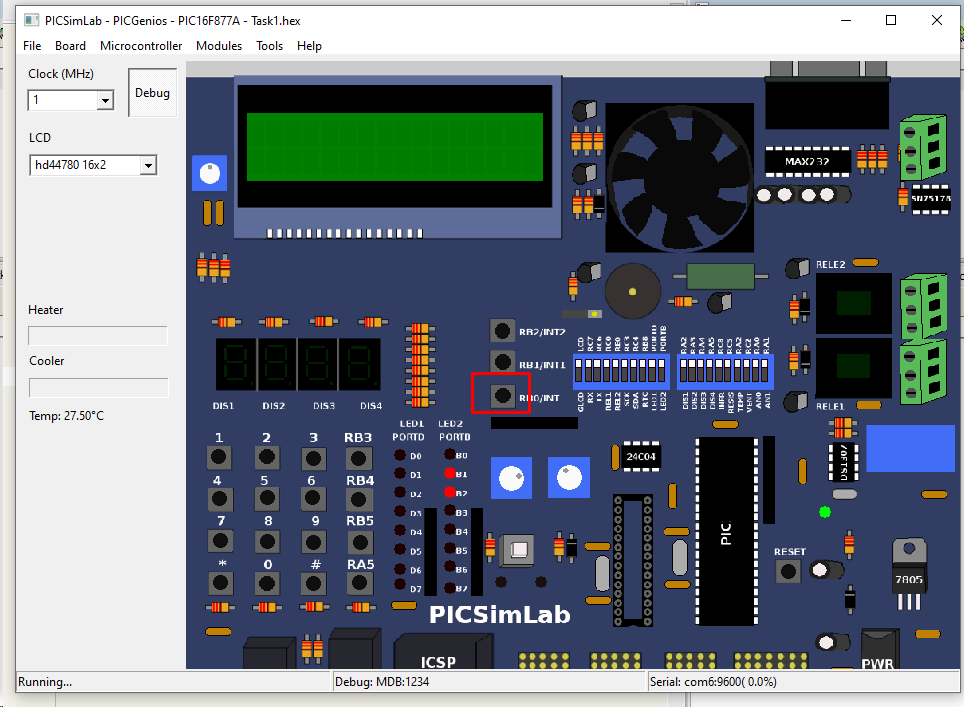
*}*

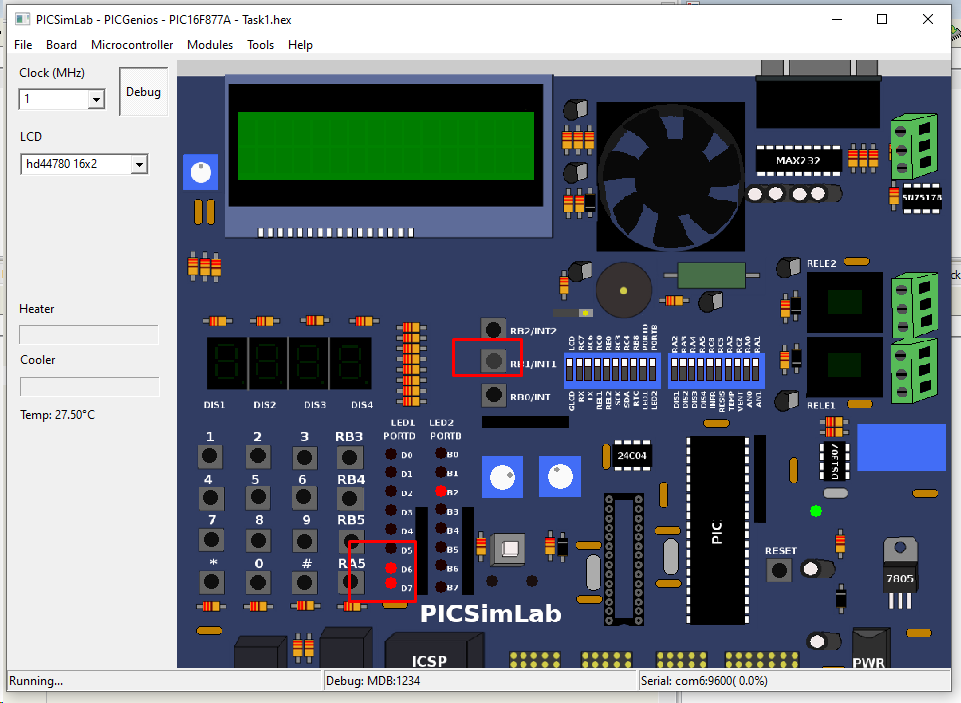
*}*

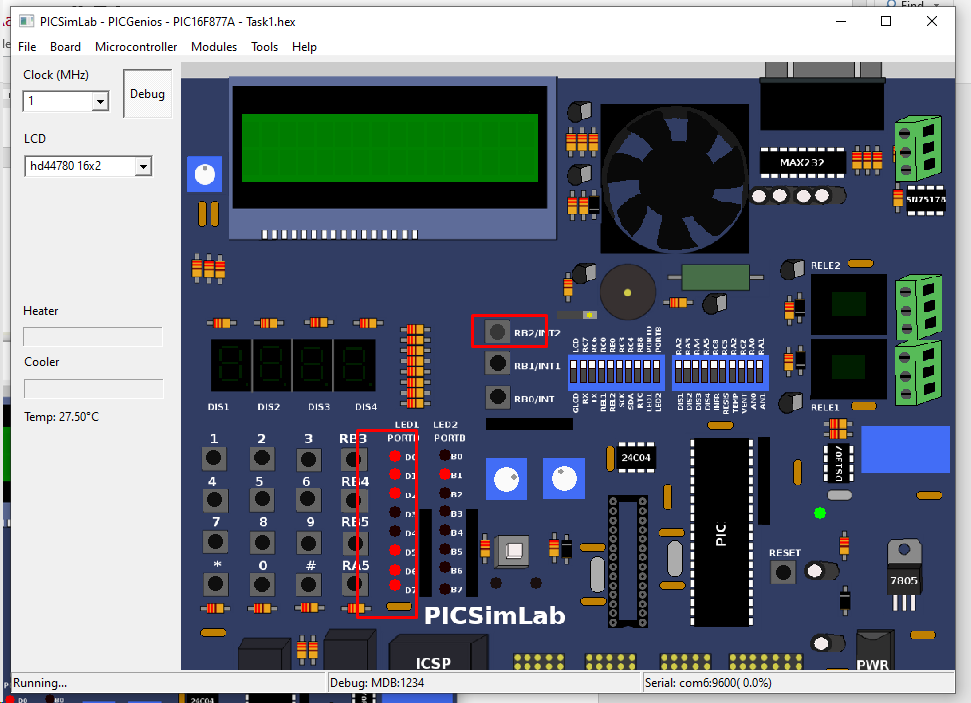
*}*

**PICSIMLAB Output**







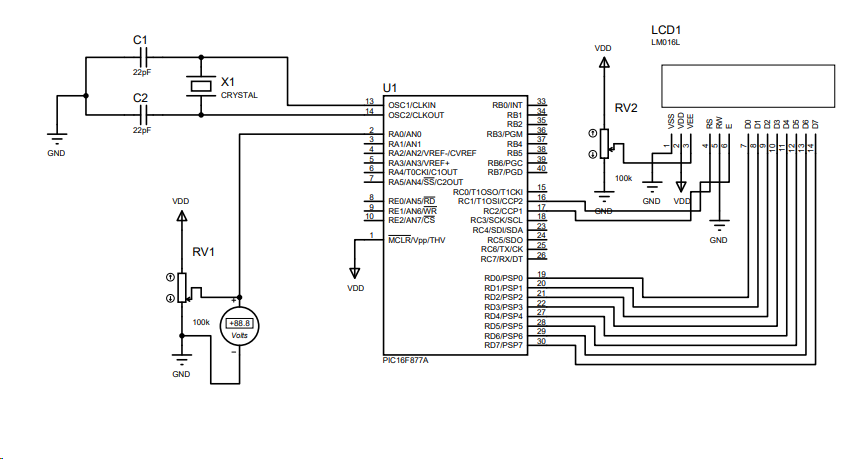


**Task2:**

**Description:**

1. AN0 is configured as Analog Input Pin, and Interanl Vref+ (5v) and VSS (0V) is used for converting the input voltage
2. Max 5V – will give ADC output value 1023
3. We convert max 5 volt value to 150 --> i.e 15.0V
4. Whenever the voltage drops below 10.5 – Low Voltage is Displayed
5. Whenever the voltage raises above 10.5 and below 13.5 –Normal Voltage is Displayed
6. Whenever the voltage raises above 13.5 High Voltage is Displayed

**Schematic:**



**Code:**

void Init(void);

void LCD\_Command(unsigned char);

void LCD\_Data(unsigned char);

void LCDOutput(unsigned int);

void Delay(unsigned int);

unsigned char k[10],x;

unsigned char n,m;

unsigned int hivalue,lovalue,adcv;

long value;

char Lowstring[] = "Low Voltage";

char Highstring[] = "High Voltage";

char Normstring[] = "Normal Voltage";

char \*ptr;

void update\_lcd(unsigned int num)

{

LCD\_Command(0x80); //Initialize cursor to first Position

LCDOutput(num);

if(num > 135)

{

ptr = (char \*)Highstring;

}

else if(num < 105)

{

ptr = (char \*)Lowstring;

}

else

{

ptr = (char \*)Normstring;

}

LCD\_Command(0xC0);

while(\*ptr != '\0')

LCD\_Data(\*ptr++);

Delay(100);

}

void main()

{

Init();

ADCON0=0x00; // sampling freq=osc\_freq/2,ADC off initially

ADCON0=0x81; //configure the A/D control registers

ADCON1=0x8E;

while(1)

{

ADCON0|=0X04; //start ADC conversion

while(ADCON0&0X04); //wait for conversion to complete

lovalue=ADRESL; //read the low 8 bit value

hivalue=ADRESH; //read the upper 8 bit value

value=((unsigned int)hivalue<<8)+(unsigned int)lovalue;

adcv = (value\*150)/1023 ;

update\_lcd(adcv);

}

}

/\*end main program\*/

void Init(void)

{

TRISD = 0x00; //Initialize the PORTD as output

TRISC = 0x00; //Initialise the PORT C as output

TRISA = 0x01;

LCD\_Command(0x38); //Initialize the 2 lines and 5\*7 Matrix LCD

Delay(100);

LCD\_Command(0x38);

Delay(100);

LCD\_Command(0x38);

Delay(100);

LCD\_Command(0x38);

Delay(100);

LCD\_Command(0x06); //Increment cursor (shift cursor to right)

Delay(100);

LCD\_Command(0x0C); //Display on,cursor off

Delay(100);

LCD\_Command(0x01); //clear display screen

Delay(100);

}

/\*define the output function\*/

/\*BCD conversion\*/

void LCDOutput(unsigned int num)

{

unsigned int j;

unsigned int i;

unsigned int tdata;

tdata = num ;

if(tdata == 0)

{

LCD\_Data(0x30); //assign formal argument to other variable

LCD\_Data(0x30);

LCD\_Data(0x2E);

LCD\_Data(0x30);

LCD\_Data('V');

}

else

{

j=0;

while (tdata != 0)

{

i = tdata - (tdata / 10) \* 10;

k[j] = i+0x30;

tdata = tdata / 10;

j++;

}

k[j] = '\0';

//LCD\_Data(k[3]);

LCD\_Data(k[2]);

LCD\_Data(k[1]);

LCD\_Data(0x2E);

LCD\_Data(k[0]);

LCD\_Data('V');

}

}

void LCD\_Command(unsigned char i)

{

PORTC&=~0x04; // RS=0

PORTD=i;

PORTC |=0x02; // RS=0,R/W=0,EN=1

PORTC&= ~0x02; // RS=0,R/W=0,EN=0

Delay(100);

}

void LCD\_Data(char i)

{

PORTC|=0x04; //RS=1

PORTD=i; //Assign the value to PORTD to display

PORTC|=0x02; // RS=1,R/W=0,EN=1

PORTC&=~0x02; // RS=1,R/W=0,EN=0

Delay(100);

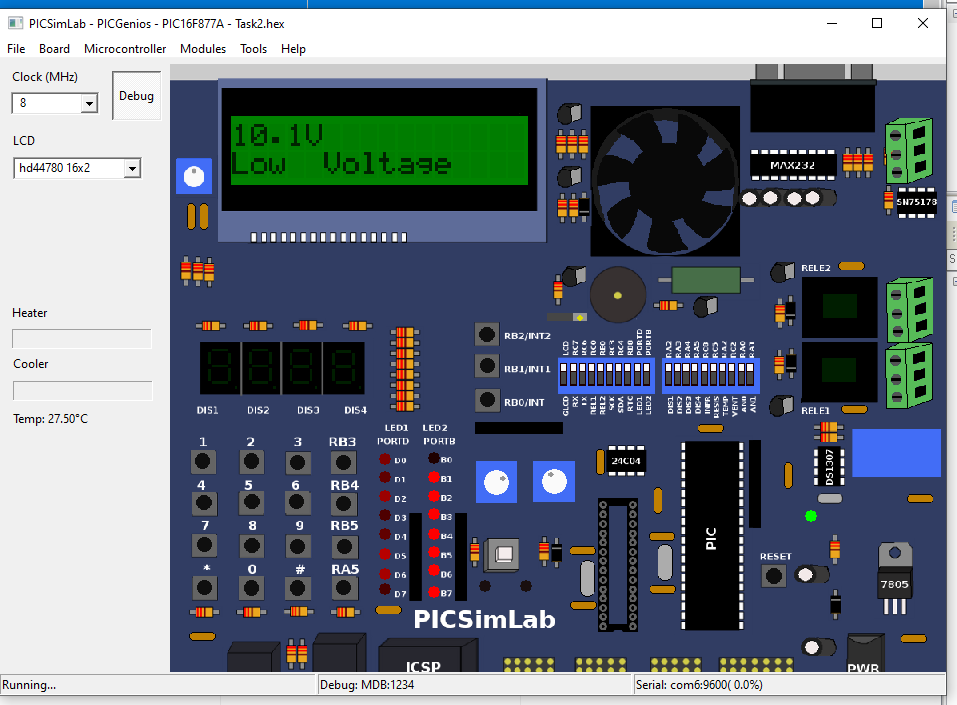
}

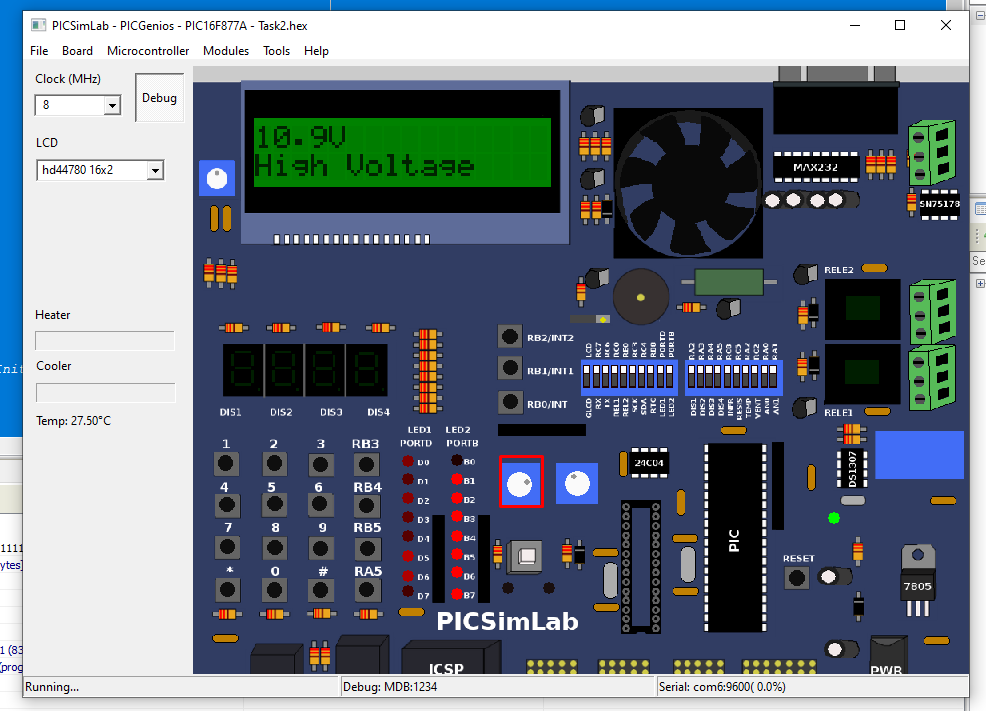
void Delay(unsigned int DelayCount)

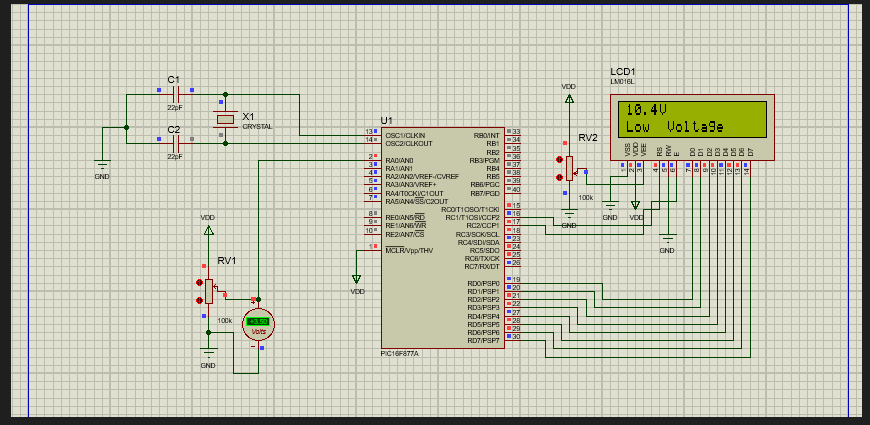
{

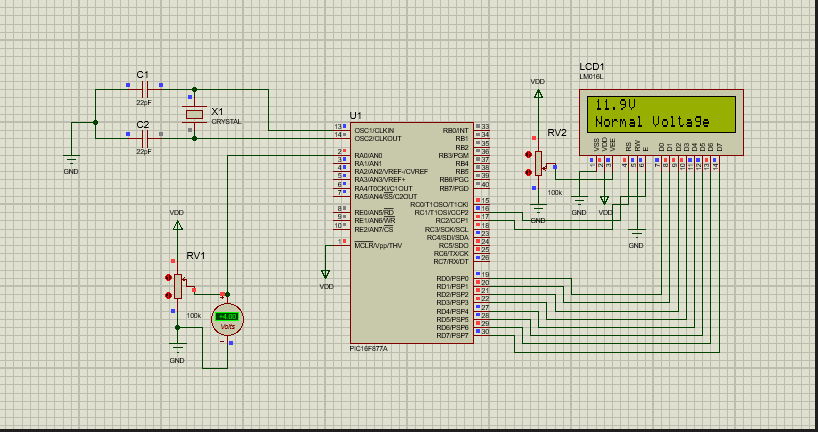
while(--DelayCount);

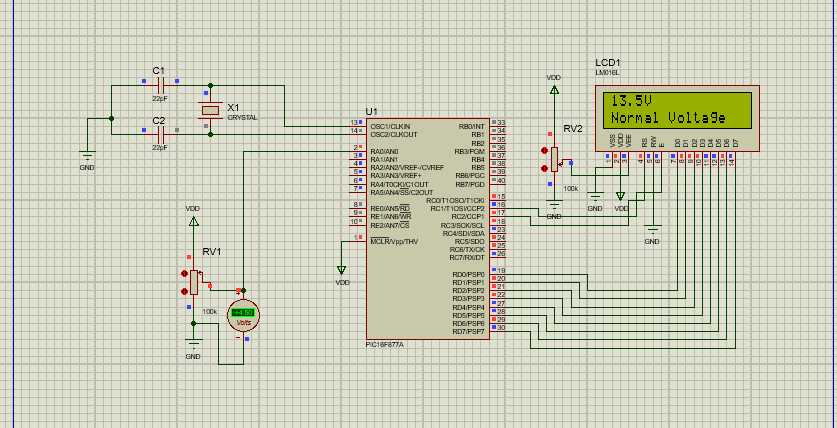
}

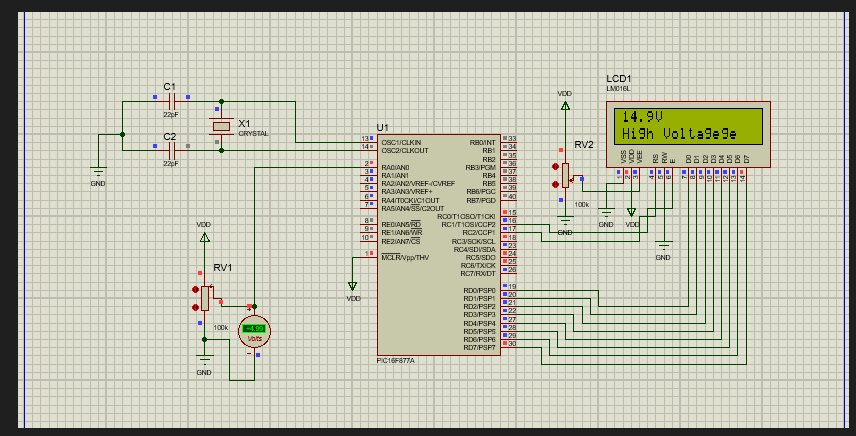
**PICSIMLAB - Output**











**Task 3: 7 Segment Display with Transistor Multiplexer**

**Description:**

1. PORT D is configured as common inputs for the 4 7 Segment Display
2. PORTA is used as Multiplexer used to select particular 7 Segment Display

**Code – PICSIMLAB**

void main()

{

unsigned int i,j;

TRISD = 0x00; //7 Segment Display output

TRISA = 0x00; //Transistor Multiplexer output

while(1)

{

for(i =0; i < 10; i++)

{

PORTD = digits[i];

for(j= 0; j <4 ; j++)

{

PORTA = 0x20 >> j;

Delay\_ms(30);

}

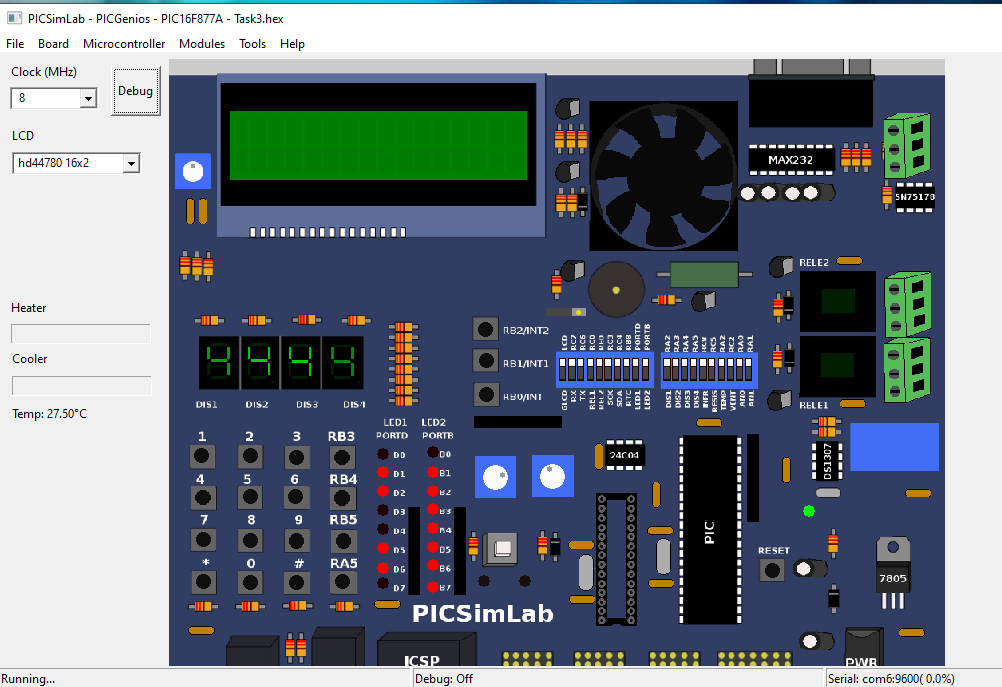
Delay\_ms(50);

}

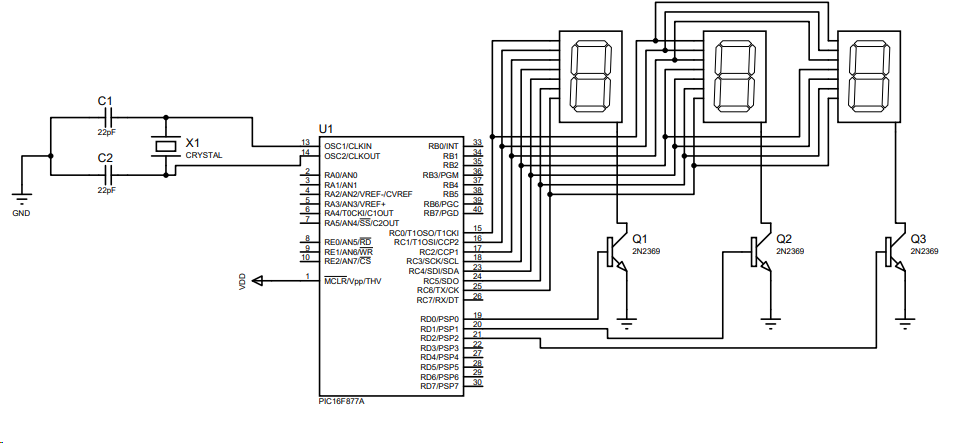
}

}

**PICSIMLAB - Output**



**Schematic: with 3 Segment Display**



**Code:**

//RA2-RA5 - Transistor Mux

//RD0-RD7 - LED OUTPUT

#define D0 0x3F

#define D1 0x06

#define D2 0x5B

#define D3 0x4F

#define D4 0x66

#define D5 0x6D

#define D6 0x7D

#define D7 0x07

#define D8 0x7F

#define D9 0x6F

unsigned int digits[10] = {D0, D1, D2, D3, D4, D5, D6, D7, D8, D9};

void main()

{

unsigned int i,j;

TRISC = 0x00; //7 Segment Display output

TRISD = 0x00; //Transistor Multiplexer output

while(1)

{

for(i =0; i < 10; i++)

{

PORTC = digits[i];

for(j= 0; j <3 ; j++)

{

PORTD = 0x4 >> j;

Delay\_ms(30);

}

}

}

}

**Proteus Output:**

