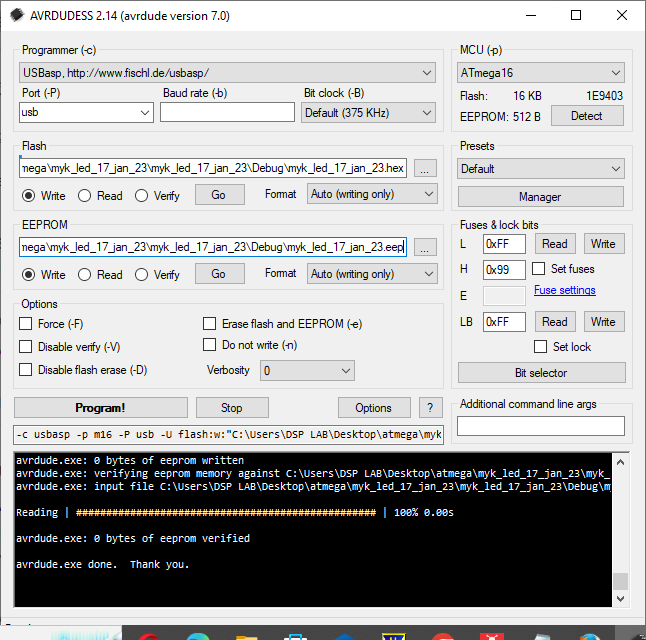
**77Programmer required for AVR ATMEGA16 microcontroller kit**

Version - AVRDUDESS-2.14-setup.exe ( Required .net framework)

GUI & settings required



**ATMEL Studio 7 programming steps for ATMEGA 16**

* Open ATMEL Studio 7
* Goto File 🡪New 🡪Project
* From C/C++ 🡪 Select GCC C executable project
* Give filename for your project
* For eg Name 🡪Project\_1 ( you can also specify your location )
* For eg Desktop/ATMEL
* Click on OK
* Click on Device Selection 🡪Microcontroller 🡪 ATMEGA16 🡪OK
* To add clock frequency for your microcontroller
  + Goto Project 🡪Project\_1 properties
  + Click on Toolchain
  + Select symbols 🡪 click on ADD icon
  + Write F\_CPU=16000000 🡪click on OK this will set clock frequency for controller
* Now under main.c write your code for your choice

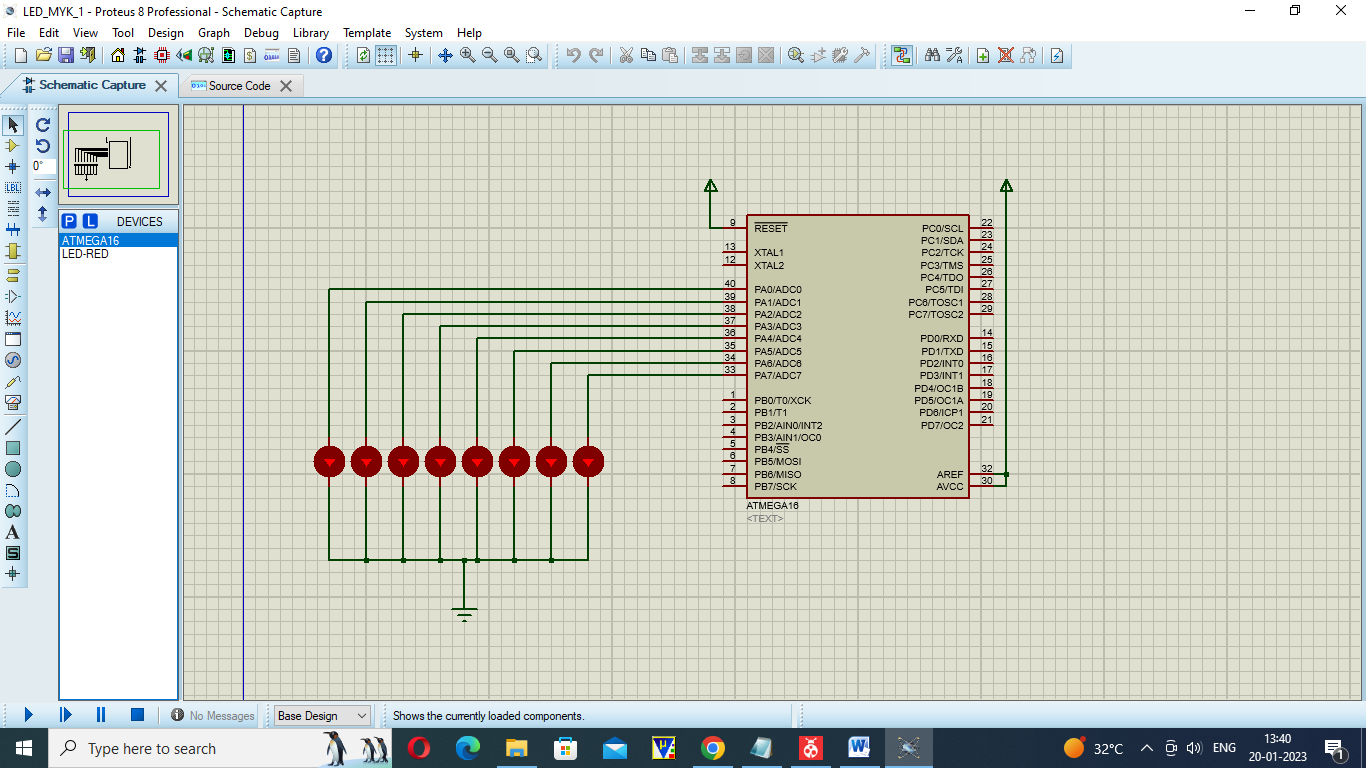
**Interfacing single LED to ATMEGA16 microcontroller**

Main.c program

**#include<avr/io.h>  
#include<util/delay.h>  
int main(void)  
{  
 DDRB | =(1<< DDB0); // Output mode   
 while(1)  
 {  
 PORTB |=(1<<PB0); // LED ON  
 \_delay\_ms(1000); // 1 sec delay   
 PORTB &=~(1<<PB0); // LED OFF  
 \_delay\_ms(1000); //1 sec delay  
 }  
}**Now Build your project by clicking on build icon

* Connect usb programmer to pc
* Connect 12v/1A adapter to the development board
* Power on the supply
* Make the necessary connection as per interfacing diagram
* Open the AVRDUDESS-2.14 software
* Set the necessary information properly fuse and lock bit information , programmer, device etc.
* Select the \*.hex file under flash & select \*.eep file under EEPROM
* Click on program button. That’s it !!!!!!!!!!!!!!!!!

**LED interfacing with ATMEGA16 microcontrolller ( PORT A )**



Program

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRA=0XFF; // port pin as output pins

while(1)

{

PORTA=0xff; // all LEDs are ON

*\_delay\_ms*(30);

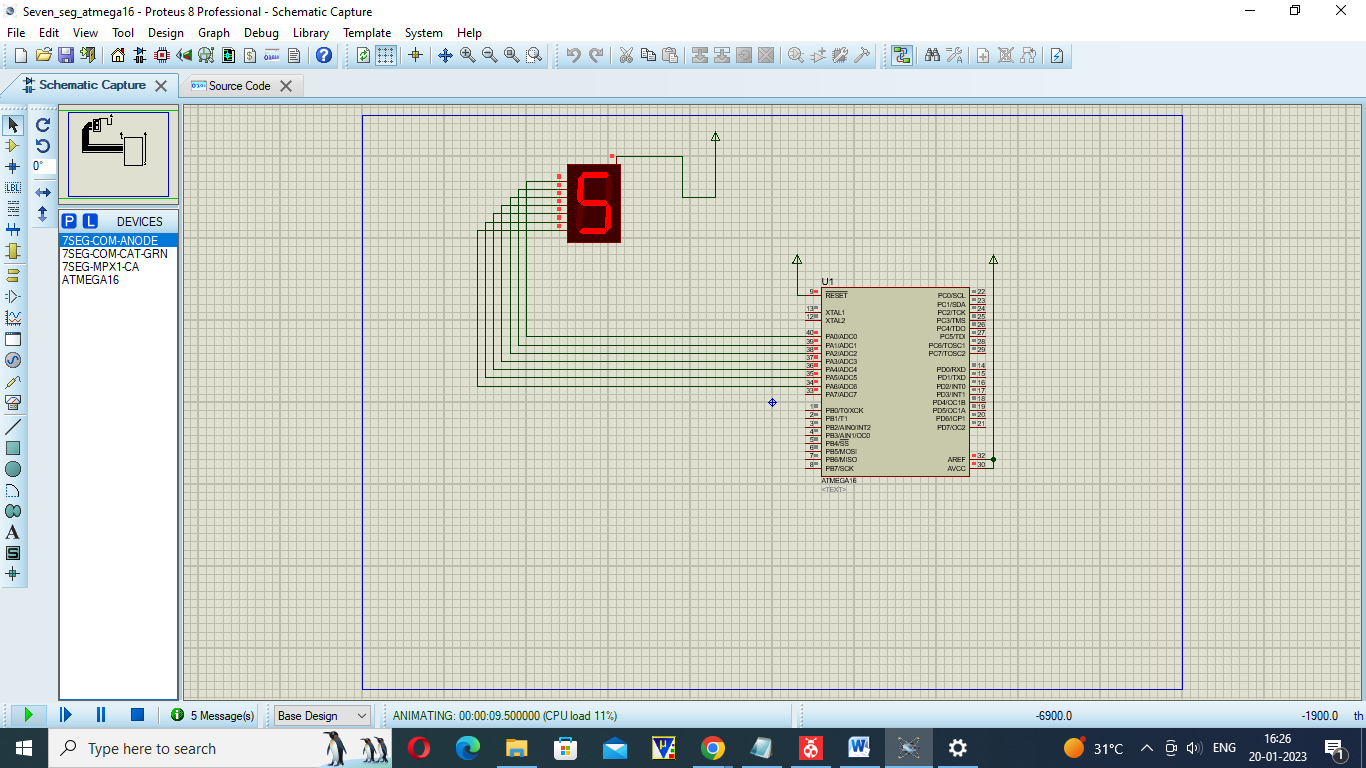
PORTA=0X00; // all LEDs are OFF

*\_delay\_ms*(30);

}

}

**SSD interfacing with ATMEGA16 microcontrolller ( PORT A )**



Program

#include <avr/io.h>

#include <util/delay.h>

int ar[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90}; // for proteus

//int ar[]={0x7e,0x30,0x6d,0x79,0x33,0x5b,0x5f,0x70,0x7f,0x7b};

// int ar[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f}; for kit

int main()

{

unsigned int i;

DDRA=0xFF;

while(1)

{

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

{

PORTA=ar[i];

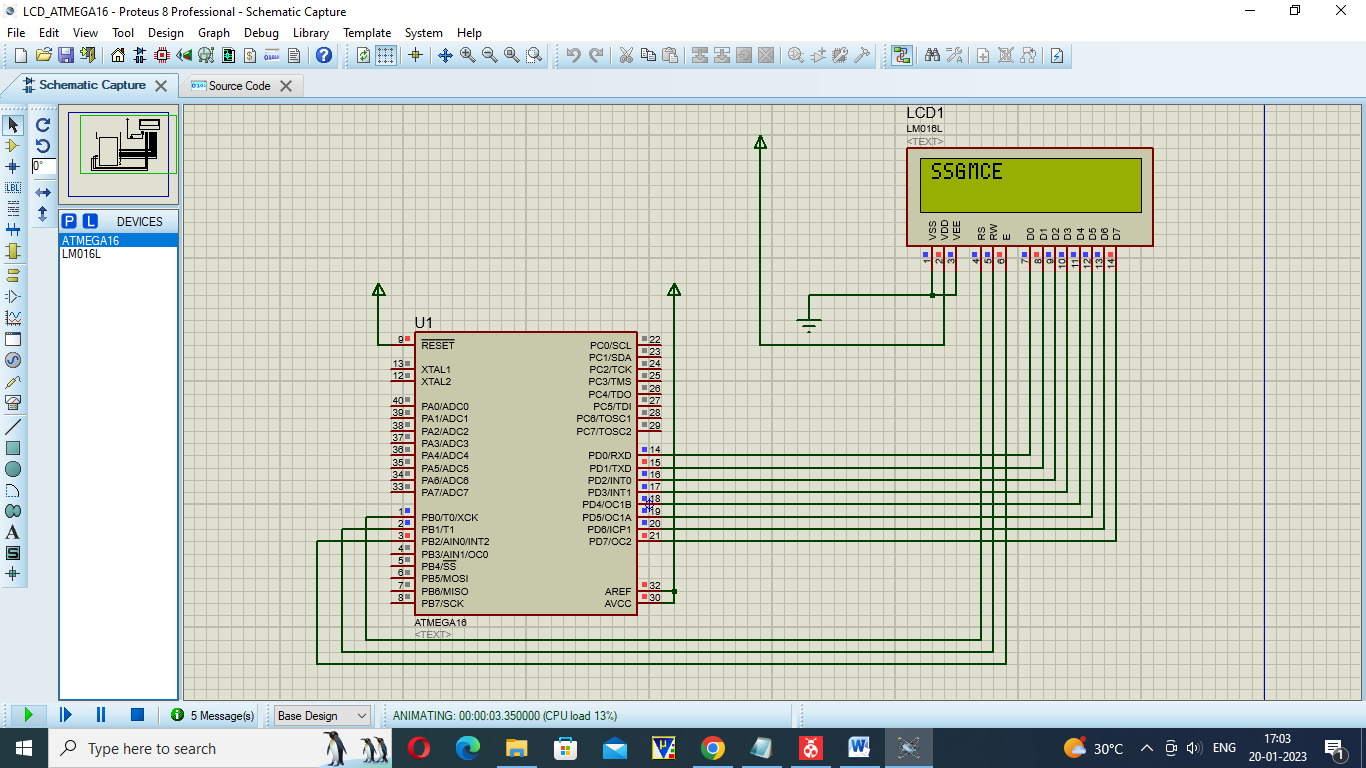
*\_delay\_ms*(100);

}

}

}

**LCD interfacing with ATMEGA16 microcontrolller**



Program

#include<avr/io.h>

#include<util/delay.h>

// writing data to the LCD

void lcd\_data(unsigned char data)

{

PORTD = data;

PORTB |=(1<<PB0); // RS=1

PORTB &=(~(1<<PB1)); //RW =0

PORTB |=(1<<PB2); // EN=1

*\_delay\_ms*(10); // delay of 10ms

PORTB &=(~(1<<PB2)); //EN=0

}

// writing command to the LCD

void lcd\_cmd(unsigned char command)

{

PORTD = command;

PORTB &=(~(1<<PB0)); //RS=0

PORTB &=(~(1<<PB1)); //RW=0

PORTB |=(1<<PB2); //EN=1

*\_delay\_ms*(10); // delay of 10ms

PORTB &=(~(1<<PB2)); //EN=0

}

void lcd\_initialise()

{

lcd\_cmd(0x38); // lcd 16x2 format

lcd\_cmd(0x06); // auto increment of cursor

lcd\_cmd(0x0c); // display on cursor off

lcd\_cmd(0x01); // clear screen

}

int main(void)

{

DDRD=0xFF; // set all pins as output pins

DDRB |=(1<<DDB0)|(1<<DDB1) |(1<<DDB2); // set 0th 1st and 2nd pin as ouput pins

lcd\_initialise();

while(1)

{

lcd\_cmd(0x80); // 1st row 1st col

lcd\_data('S');

lcd\_cmd(0x81); //

lcd\_data('S');

lcd\_cmd(0x82); //

lcd\_data('G');

lcd\_cmd(0x83); //

lcd\_data('M');

lcd\_cmd(0x84); //

lcd\_data('C');

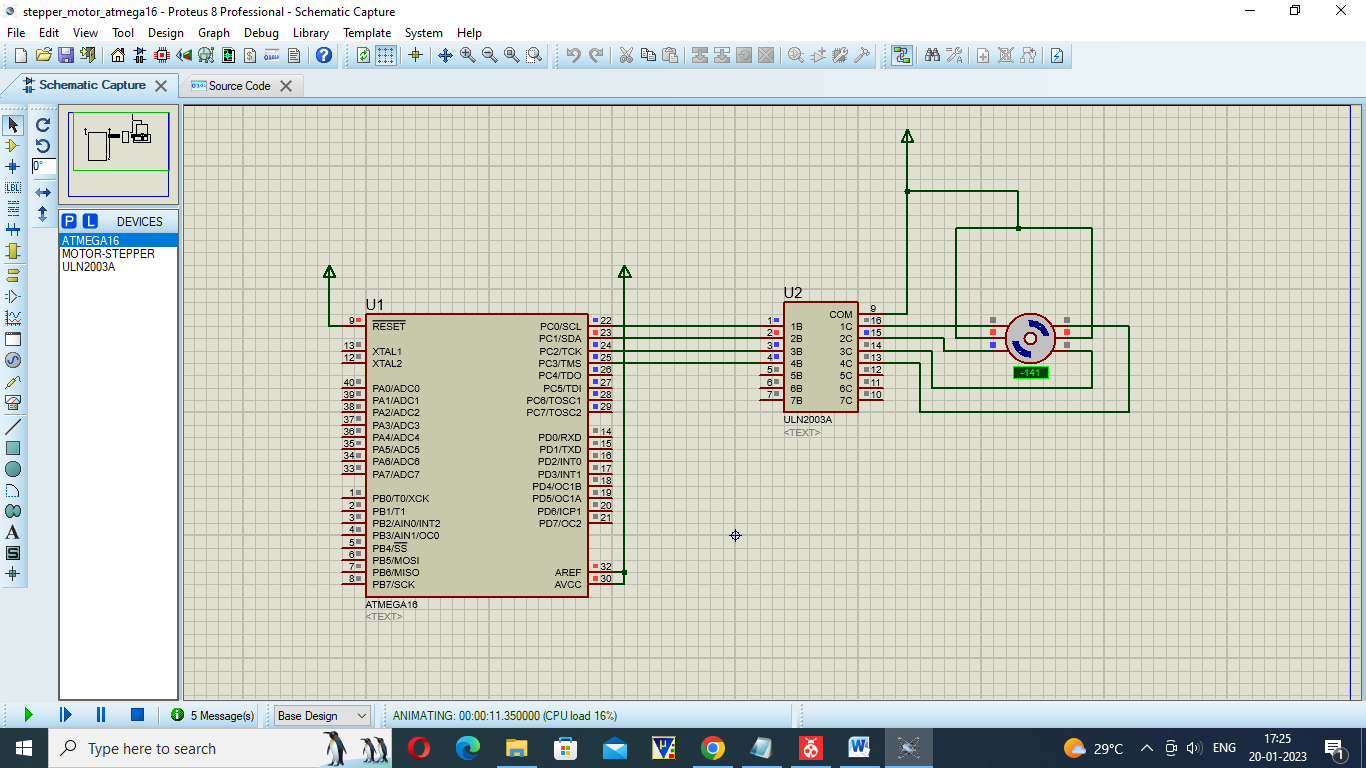
lcd\_cmd(0x81); //

lcd\_data('E');

}

}

**Stepper Motor interfacing with ATMEGA16 microcontrolller**



Program

#include<avr/io.h>

#include<util/delay.h>

int main(void)

{

DDRC=0xFF;

while(1)

{

PORTC=0X01;

*\_delay\_ms*(5);

PORTC=0X02;

*\_delay\_ms*(5);

PORTC=0X04;

*\_delay\_ms*(5);

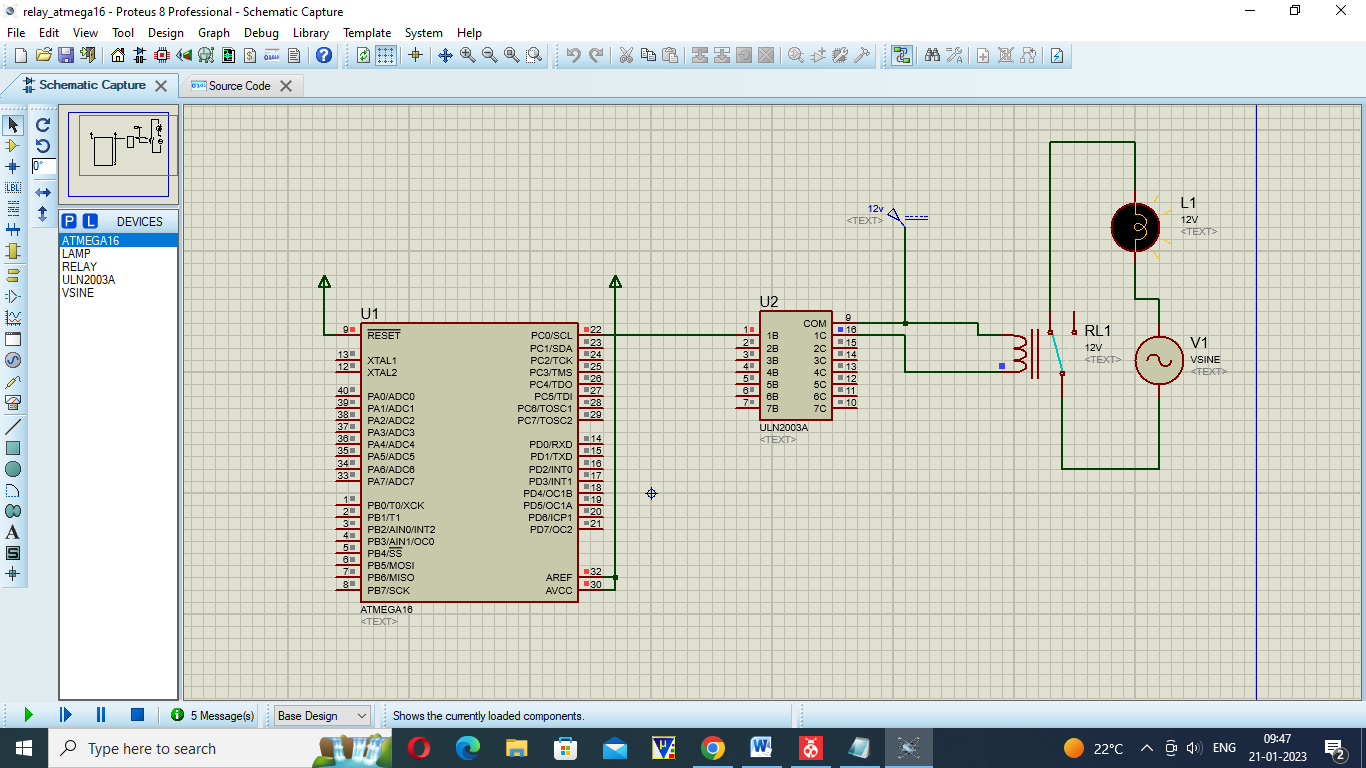
PORTC=0X08;

*\_delay\_ms*(5);

}

}

**Relay interfacing with ATMEGA16 microcontrolller**



**PROGRAM**

#include <avr/io.h>

#include <util/delay.h>

int main()

{

DDRC=0x01;

PORTC=0x00;

while(1)

{

PORTC=0x01;

*\_delay\_ms*(500);

PORTC=0x00;

*\_delay\_ms*(500);

}

}

**LM35(Temperature Sensor) interfacing with ATMEGA16 microcontrolller**

