SIMATS ENGINEERING

Engineer to Excel

SAVEETHA

INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

(Declared as Deemed to be University under Section 3 of UGC Act 1956)

INSTITUTE OF ECE

ECA14 – EMBEDDED SYSTEMS

Exp No:

FLASHING OF LED USING AT89C51 MICROCONTROLLER USING PROTEUS

AIM:

To Write an assembly language program to flash the LED using AT89C51

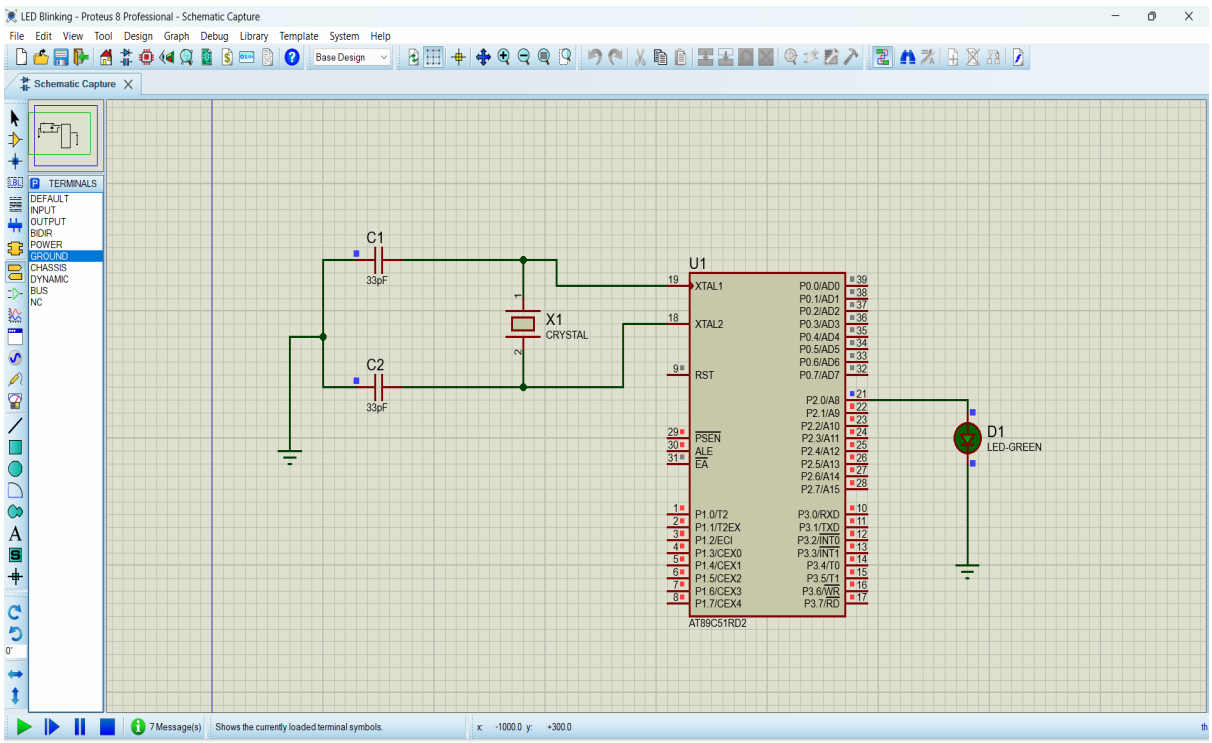
SOFTWARES REQUIRED:

- Proteus software

PROGRAM

```
        ORG 0000H
    UP: SETB P2.0
        ACALL DELAY
        CLR P2.0
        ACALL DELAY
        SJMP UP
DELAY: MOV R4,#35
    H1:MOV R3,#255
    H2:DJNZ R3,H2
       DJNZ R4,H1
       RET
       END
```

CIRCUIT DIAGRAM:



RESULT

Thus the program has been successfully verified and executed.

Exp No:

GENERATION OF SQUARE WAVE USING PROTEUS

AIM:

To write an assembly language program to Generate square wave using AT89C51.

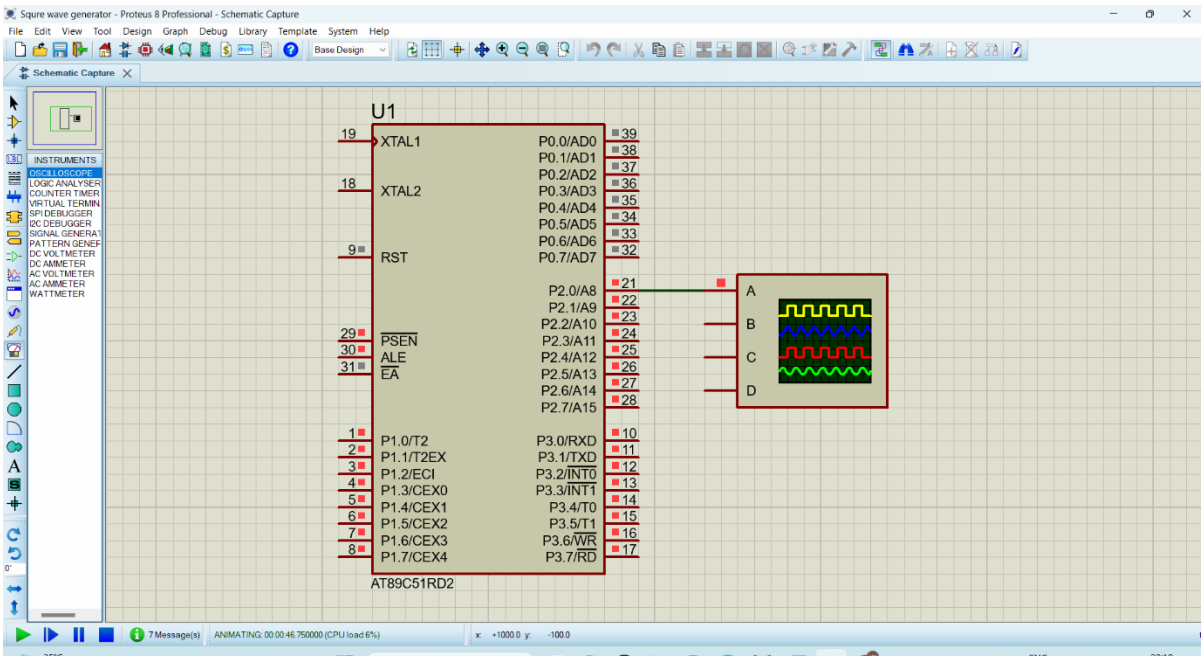
SOFTWARE REQUIRED:

- Proteus 8 software.

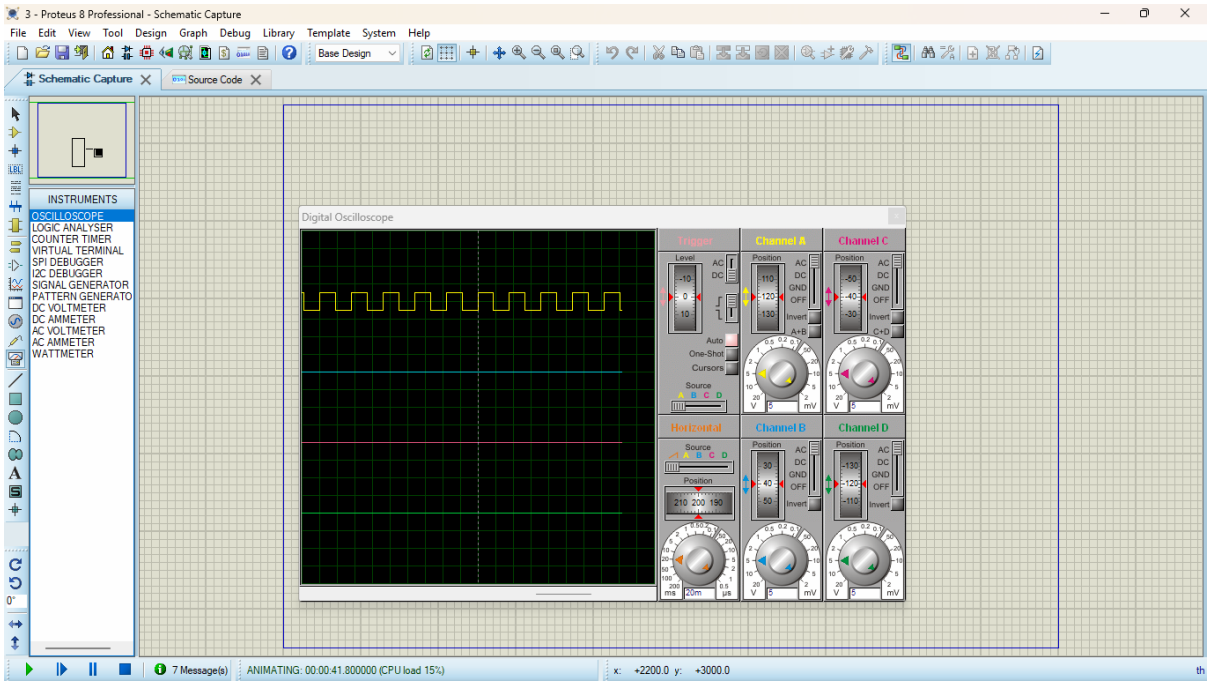
PROGRAM

```
        ORG 0000H
UP:  SETB P2.0
     ACALL DELAY
     CLR P2.0
     ACALL DELAY
     SJMP UP
DELAY: MOV R4,#35
H1:MOV R3,#255
H2:DJNZ R3,H2
   DJNZ R4,H1
   RET
   END
```

CIRCUIT DIAGRAM:



OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

SMOOTH BRIGHTNESS CHANGE OF LED USING AT89C51 USING PROTEUS

AIM:

To write an assembly language program for Fade in Fade out of LED Using AT89C51 using Keil and Proteus

SOFTWARE REQUIRED:

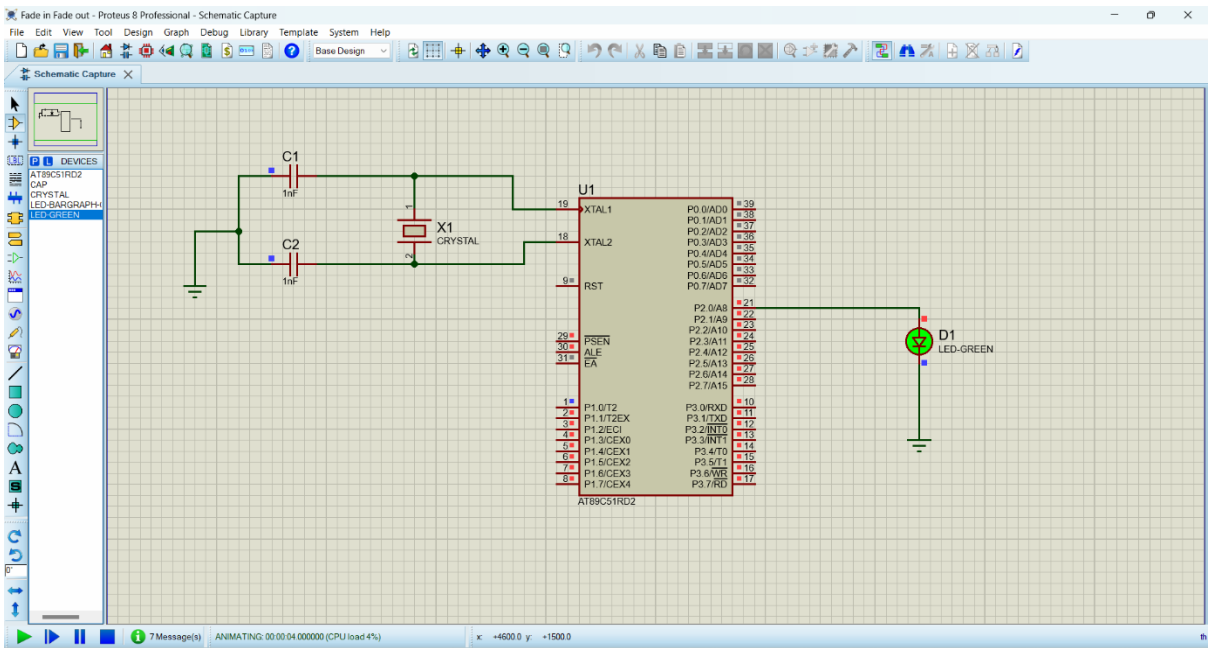
- Proteus 8 software.

PROGRAM:

```
#include <REGX52.h>
delay(unsigned int y)
{
  unsigned int i,j;
for(i=0;i<y;i++)
{
  for(j=0;j<1275;j++){}
}
}
```

```
main()
{
   while(1)
{
      delay(100);
      P1_0 = 0;
      delay(100);
      P1_0 = 1;
   }
}
```

CIRCUIT DIAGRAM:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

STEPPER MOTOR USING AT89C51 USING PROTEUS

AIM:

To write an assembly language program for Stepper Motor Using AT89C51 using Keil and Proteus

SOFTWARE REQUIRED:

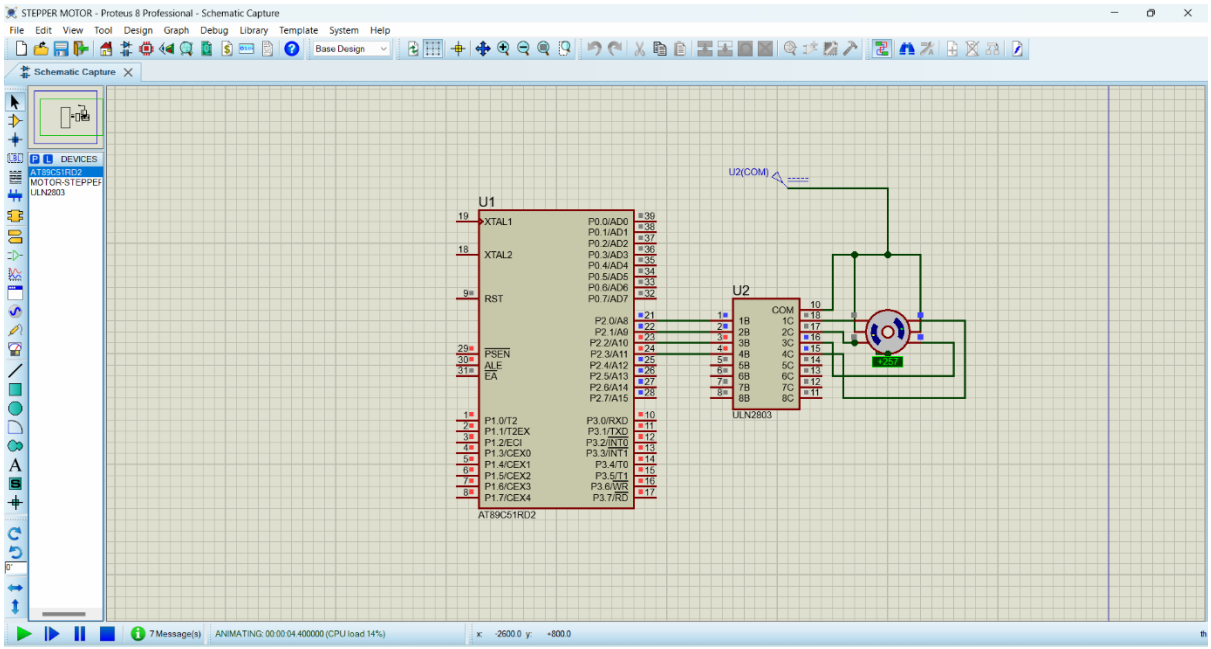
- Proteus 8 software.

PROGRAM:

```
ORG 0000H
UP: MOV P2,#09H
ACALL DELAY
MOV P2,#0CH
ACALL DELAY
MOV P2,#06H
ACALL DELAY
MOV P2,#03H
ACALL DELAY
SJMP UP
DELAY:MOV R4,#18
H1:MOV R3,#255
H2:DJNZ R3,H2
DJNZ R4,H1
RET
```

END

CIRCUIT DIAGRAM:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

INTERFACING OF RELAY USING AT89C51 USING PROTEUS

AIM:

To write an assembly language program for Interfacing of Relay Using AT89C51 using Keil and Proteus

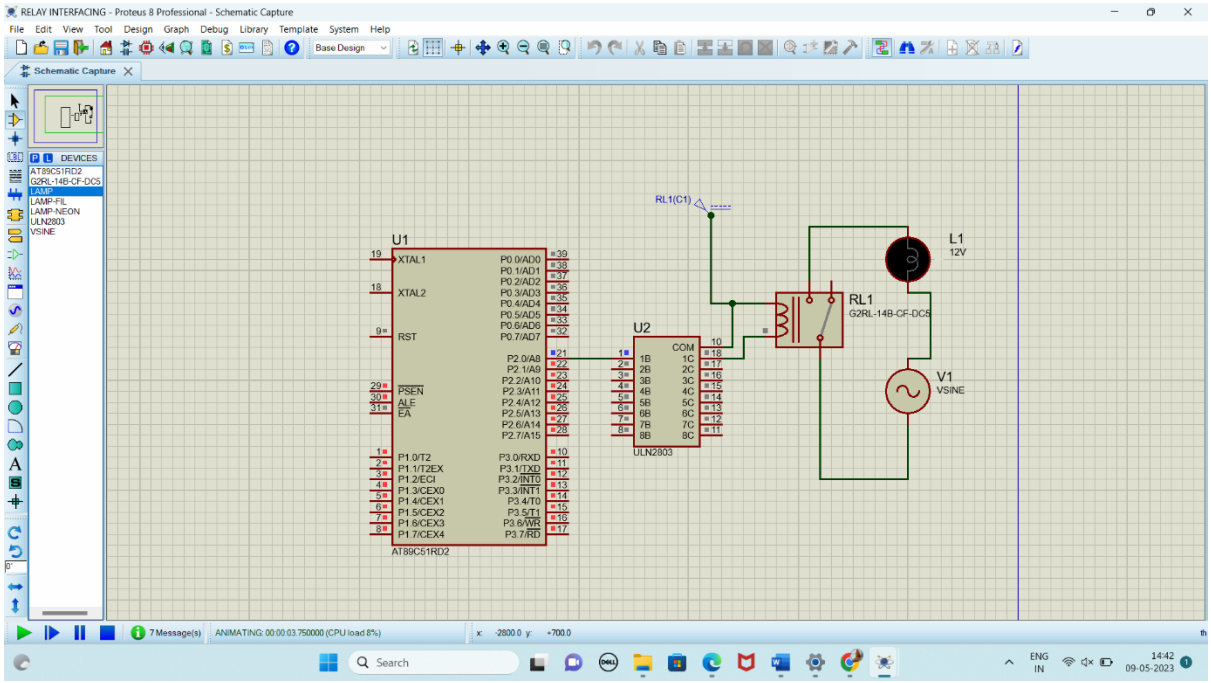
SOFTWARE REQUIRED:

- Proteus 8 software.

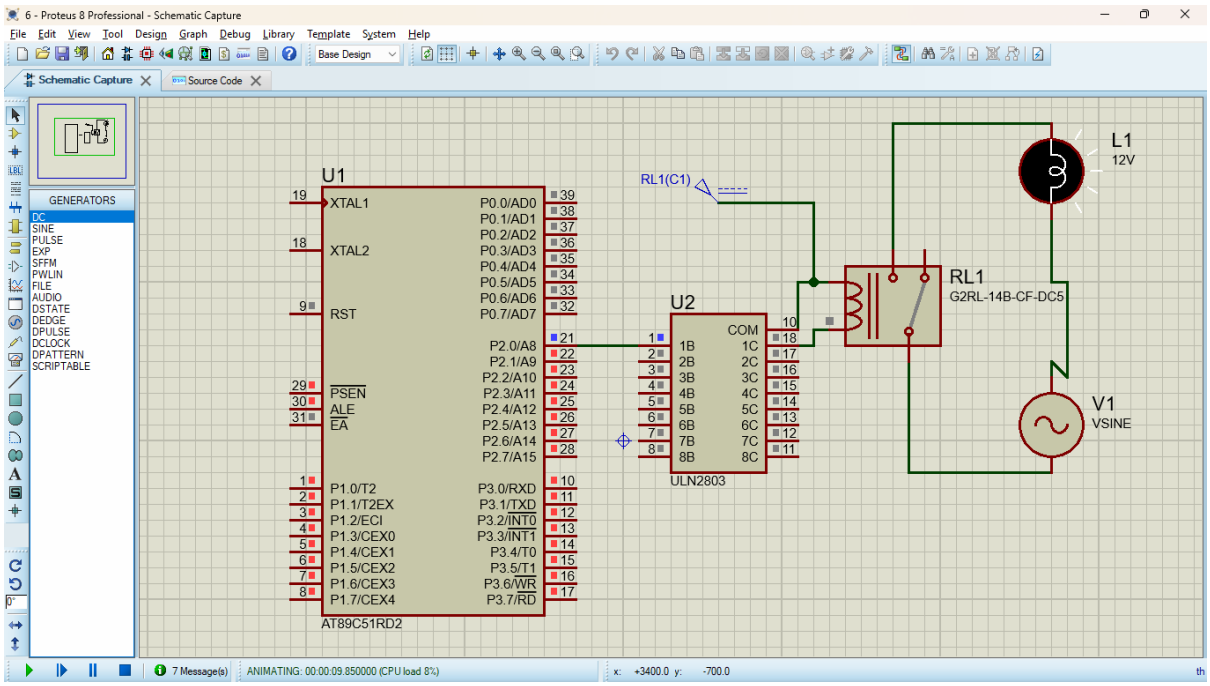
PROGRAM:

```
ORG 0000H
UP:SETB P2.0
ACALL DELAY
CLR P2.0
ACALL DELAY
SJMP UP
DELAY:MOV R4,#18
H1:MOV R3,#255
H2:DJNZ R3,H2
DJNZ R4,H1
RET
END
```

CIRCUIT DIAGRAM:



OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

LED TOGGLE USING AT89C51 USING PROTEUS

AIM:

To write an assembly language program for LED Toggle Using AT89C51 using Keil and Proteus

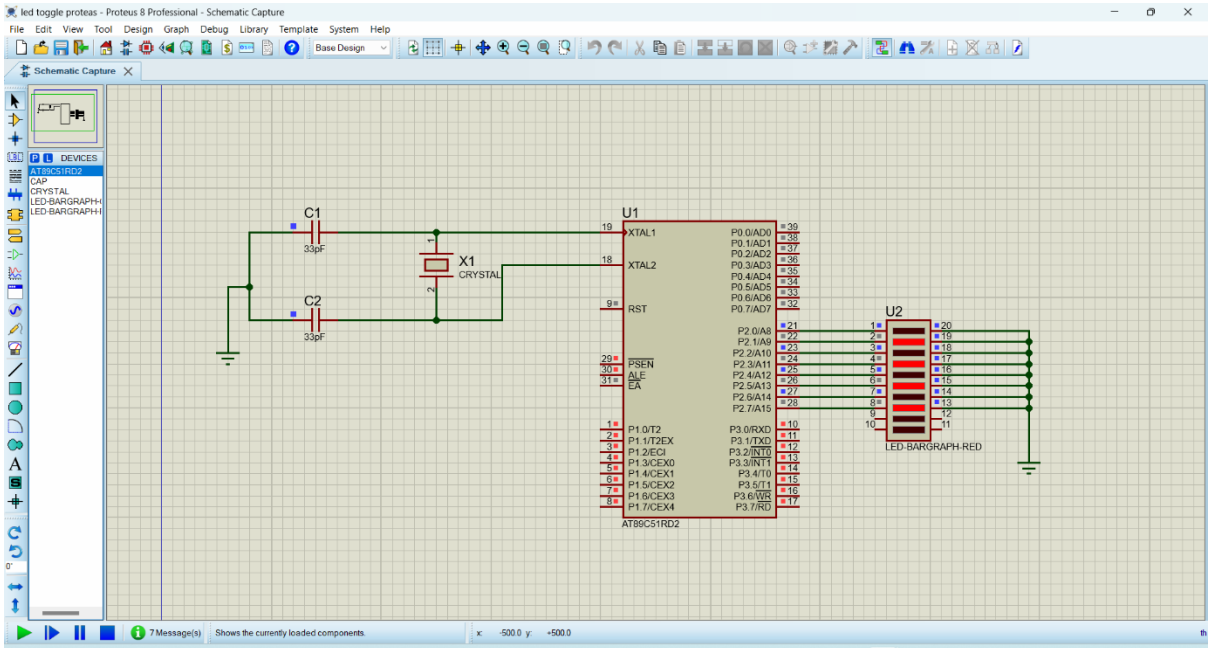
SOFTWARE REQUIRED:

- Proteus 8 software.

PROGRAM:

```
ORG 0000H
UP: MOV P2,#55H
ACALL DELAY
MOV P2,#0AAH
ACALL DELAY
SJMP UP

DELAY:MOV R4,#10
H1:MOV R3,#255
H2:DJNZ R3,H2
DJNZ R4,H1
RET
END
```

CIRCUIT DIAGRAM:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

7 SEGMENT DISPLAY USING AT89C51 USING PROTEUS

AIM:

To write an assembly language program for 7 Segment Display Using AT89C51 using Keil and Proteus

SOFTWARE REQUIRED:

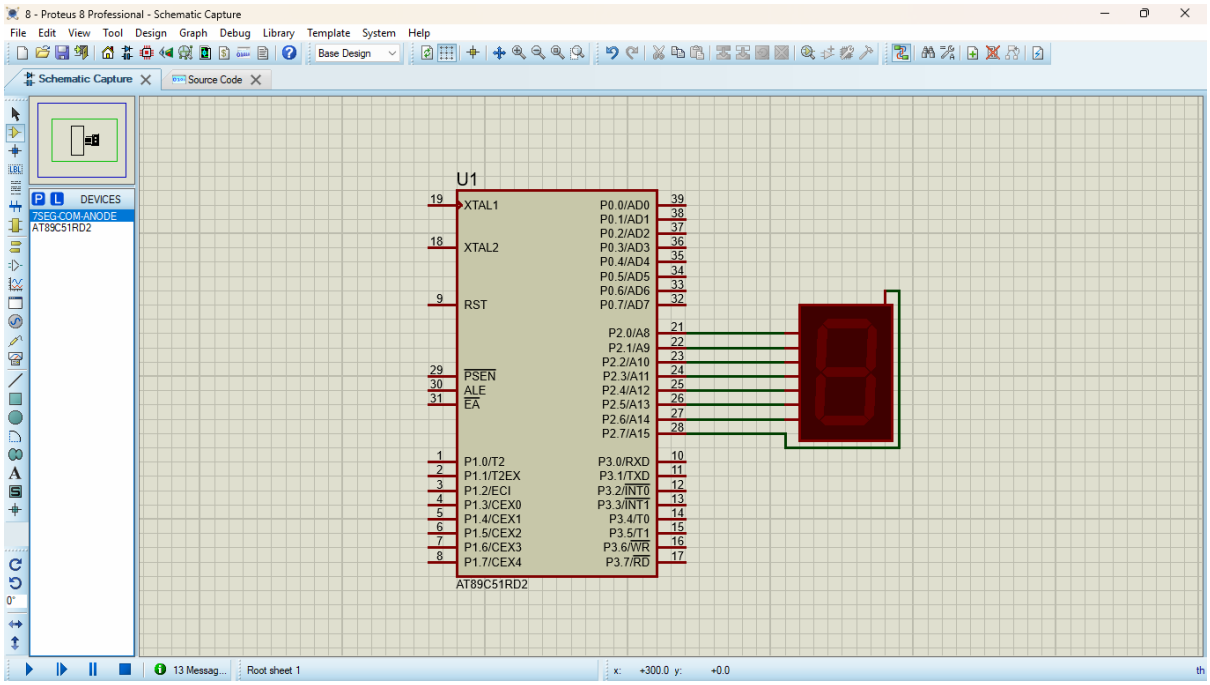
- Proteus 8 software.

PROGRAM:

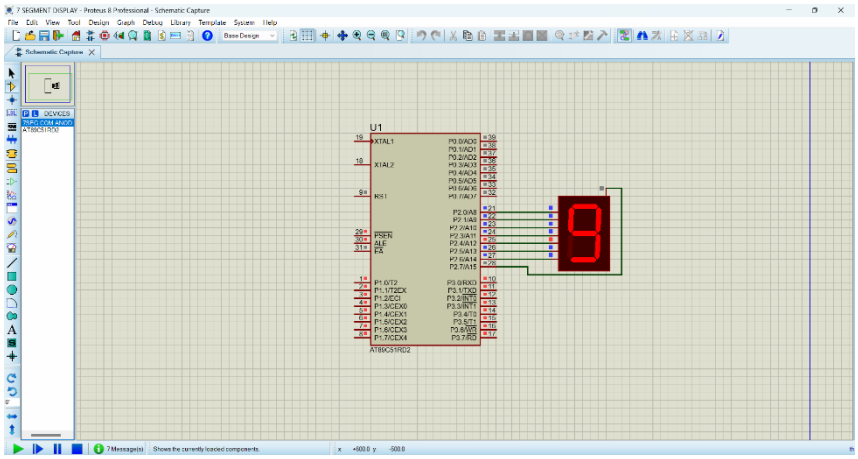
```
ORG 000H
UP:MOV P2,#0C0H
ACALL DELAY
MOV P2,#0F9H
ACALL DELAY
MOV P2,#0A4H
ACALL DELAY
MOV P2,#0B0H
ACALL DELAY
MOV P2,#99H
ACALL DELAY
MOV P2,#92H
ACALL DELAY
```

```
MOV P2,#82H
ACALL DELAY
MOV P2,#0F8H
ACALL DELAY
MOV P2, #80H
ACALL DELAY
MOV P2,#90H
ACALL DELAY

DELAY: MOV R5,#10
H1:MOV R4,#180
H2:MOV R3,#255
H3:DJNZ R3,H3
DJNZ R4,H2
DJNZ R5,H1
RET
END
```

CIRCUIT DIAGRAM:



OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

LED CHASER USING AT89C51 USING PROTEUS

AIM:

To write an assembly language program for LED Chaser Using AT89C51 using Keil and Proteus

SOFTWARE REQUIRED:

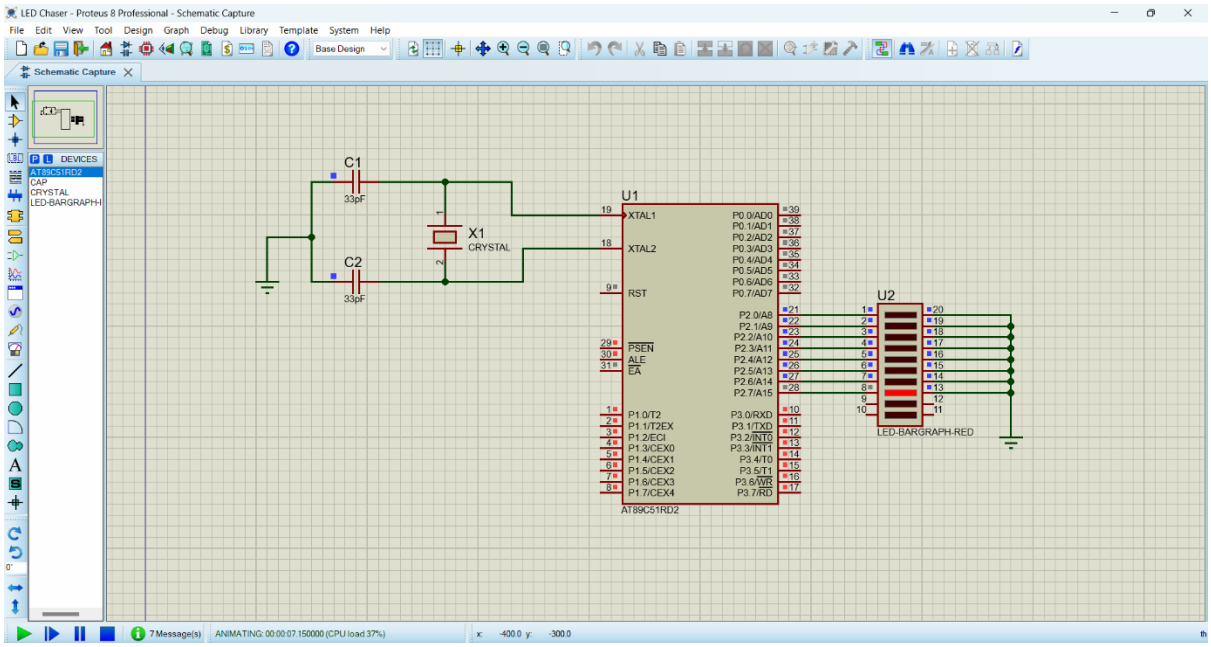
- Proteus 8 software.

PROGRAM:

```
ORG 0000H
UP: MOV P2,#01H
ACALL DELAY
MOV P2,#02H
ACALL DELAY
MOV P2,#04H
ACALL DELAY
```

```
MOV P2,#08H
ACALL DELAY
MOV P2,#10H
ACALL DELAY
MOV P2,#20H
ACALL DELAY
MOV P2,#40H
ACALL DELAY
MOV P2,#80H
ACALL DELAY
SJMP UP

DELAY: MOV R4,#255
H1: DJNZ R4,H1
RET
END
```

CIRCUIT DIAGRAM:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

ARDUINO BASED MANUAL ELECTRONIC COUNTER USING PROTEUS

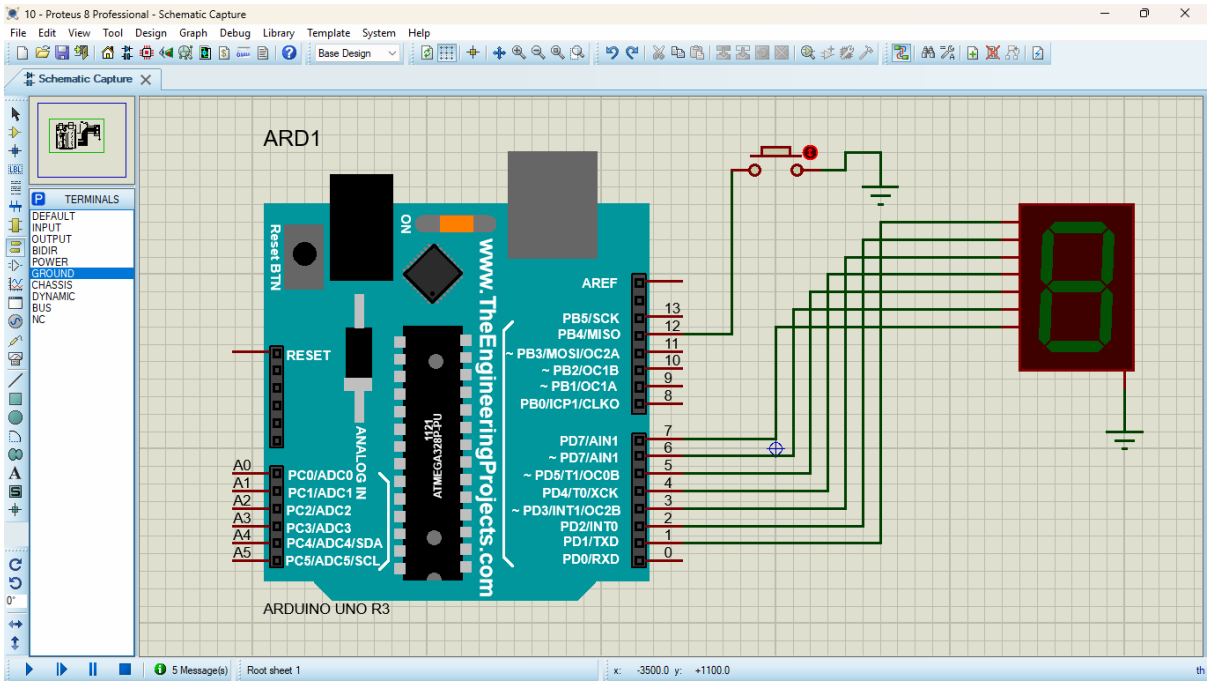
AIM:

To write an Embedded C program for Manual Electronic counter using Arduino IDE and Proteus

SOFTWARE REQUIRED:

- Proteus 8 software
- Arduino IDE

CIRCUIT DIAGRAM:



PROGRAM:

```
int x0,x1,x2,x3,x4,x5,x6,x7,x8,x9;
int delay_time=200;

void setup() {
  //configure pin2 as an input and enable the internal pull-up resistor
 pinMode(12, INPUT_PULLUP);
   pinMode(1,OUTPUT);
 pinMode(2,OUTPUT);
 pinMode(3,OUTPUT);
 pinMode(4,OUTPUT);
 pinMode(5,OUTPUT);
 pinMode(6,OUTPUT);
 pinMode(7,OUTPUT);

}
void loop() {
  //read the pushbutton value into a variable
```

```
int sensorVal = digitalRead(12);


 if (sensorVal == LOW) {
   x0=true;
 }
  while(x0){
   zero();
     sensorVal = digitalRead(12);
     if (sensorVal == LOW) {
        x1=true;
        x0=false;
      }
        }
  while(x1){
   one();
     sensorVal = digitalRead(12);
     if (sensorVal == LOW) {
        x2=true;
        x1=false;
      }
        }
   while(x2){
   two();
     sensorVal = digitalRead(12);
     if (sensorVal == LOW) {
        x3=true;
        x2=false;
      }
        }
  while(x3){
```

```
  three();
    sensorVal = digitalRead(12);
    if (sensorVal == LOW) {
       x4=true;
       x3=false;
      }
        }
  while(x4){
   four();
    sensorVal = digitalRead(12);
    if (sensorVal == LOW) {
       x5=true;
       x4=false;
      }
        }
  while(x5){
   five();
    sensorVal = digitalRead(12);
    if (sensorVal == LOW) {
       x6=true;
       x5=false;
      }
        }
while(x6){
  six();
    sensorVal = digitalRead(12);
    if (sensorVal == LOW) {
       x7=true;
       x6=false;
      }
```

```
            }
    while(x7){
     seven();
       sensorVal = digitalRead(12);
       if (sensorVal == LOW) {
          x8=true;
          x7=false;
        }
           }
 while(x8){
     eight();
       sensorVal = digitalRead(12);
       if (sensorVal == LOW) {
          x9=true;
          x8=false;
        }
           }
 while(x9){
     nine();
       sensorVal = digitalRead(12);
       if (sensorVal == LOW) {
          x0=true;
          x9=false;
        }
           }

}

void zero()
{
```

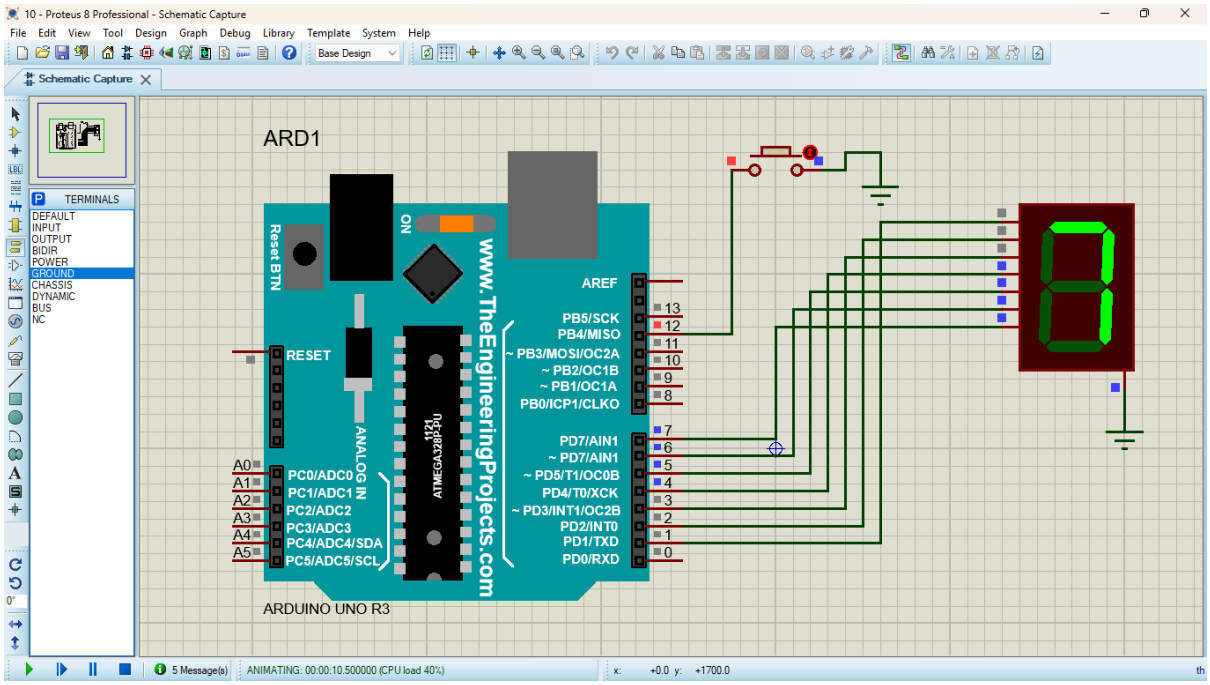
```
  for(int i=1;i<7;i++)
  {
 digitalWrite(i,HIGH);
 digitalWrite(7,LOW);
  }
  delay(delay_time);
}

void one()
{
  digitalWrite(1,LOW);
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  delay(delay_time);
}

 void two()
{
  digitalWrite(1,HIGH);
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  delay(delay_time);
```

```
}

 void three()
{
  digitalWrite(1,HIGH);
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  delay(delay_time);
}

 void four()
{
  digitalWrite(1,LOW);
  digitalWrite(2,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,HIGH);
  digitalWrite(7,HIGH);
  delay(delay_time);
}

 void five()
{
  digitalWrite(1,HIGH);
  digitalWrite(2,LOW);
```

```
 digitalWrite(3,HIGH);
 digitalWrite(4,HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,HIGH);
 digitalWrite(7,HIGH);
 delay(delay_time);
}

  void six()
{
 digitalWrite(1,HIGH);
 digitalWrite(2,LOW);
 digitalWrite(3,HIGH);
 digitalWrite(4,HIGH);
 digitalWrite(5,HIGH);
 digitalWrite(6,HIGH);
 digitalWrite(7,HIGH);
 delay(delay_time);
}

 void seven()
{
 digitalWrite(1,HIGH);
 digitalWrite(2,HIGH);
 digitalWrite(3,HIGH);
 digitalWrite(4,LOW);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 digitalWrite(7,LOW);
 delay(delay_time);
```

```
  }

   void eight()
  {
    digitalWrite(1,HIGH);
    digitalWrite(2,HIGH);
    digitalWrite(3,HIGH);
    digitalWrite(4,HIGH);
    digitalWrite(5,HIGH);
    digitalWrite(6,HIGH);
    digitalWrite(7,HIGH);
    delay(delay_time);
  }

   void nine()
{
    digitalWrite(1,HIGH);
    digitalWrite(2,HIGH);
    digitalWrite(3,HIGH);
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(7,HIGH);
    delay(delay_time);
  }
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

ARDUINO TO 16x2 LCD DISPLAY USING PROTEUS

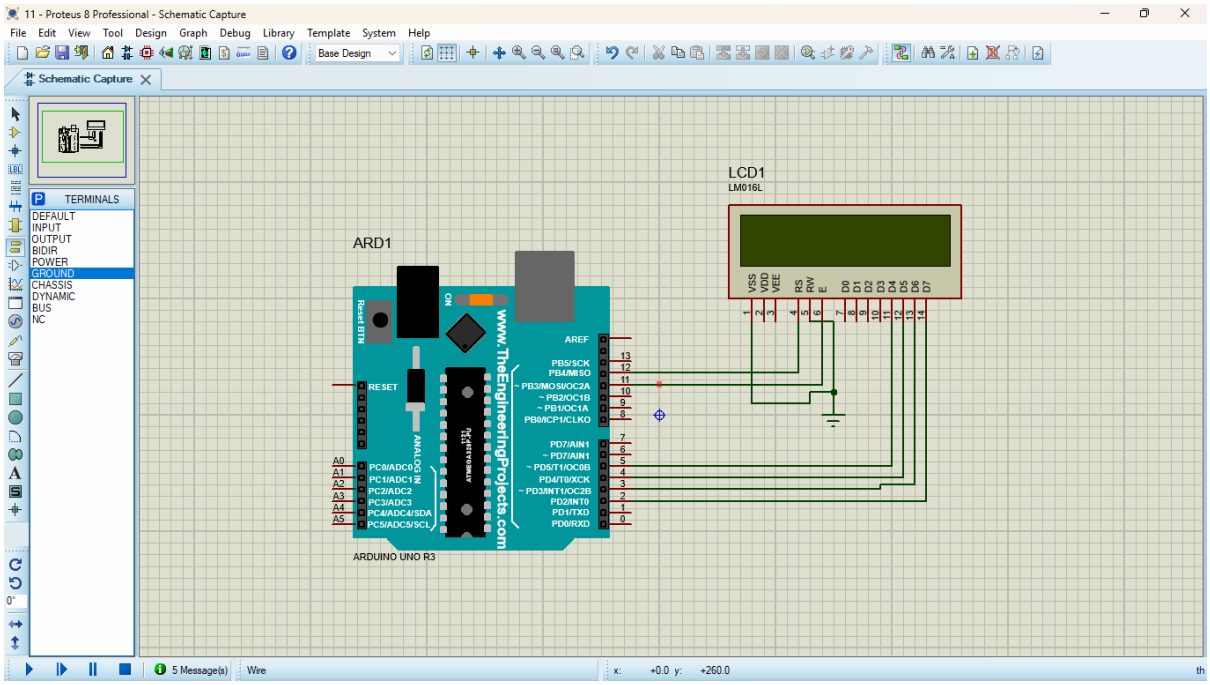
AIM:

To write an Embedded C program to interface Arduino Uno and 16 x 2 LCD display using Arduino Uno and Proteus

SOFTWARE REQUIRED:

- Proteus 8 software.
- Arduino IDE software

CIRCUIT DIAGRAM:



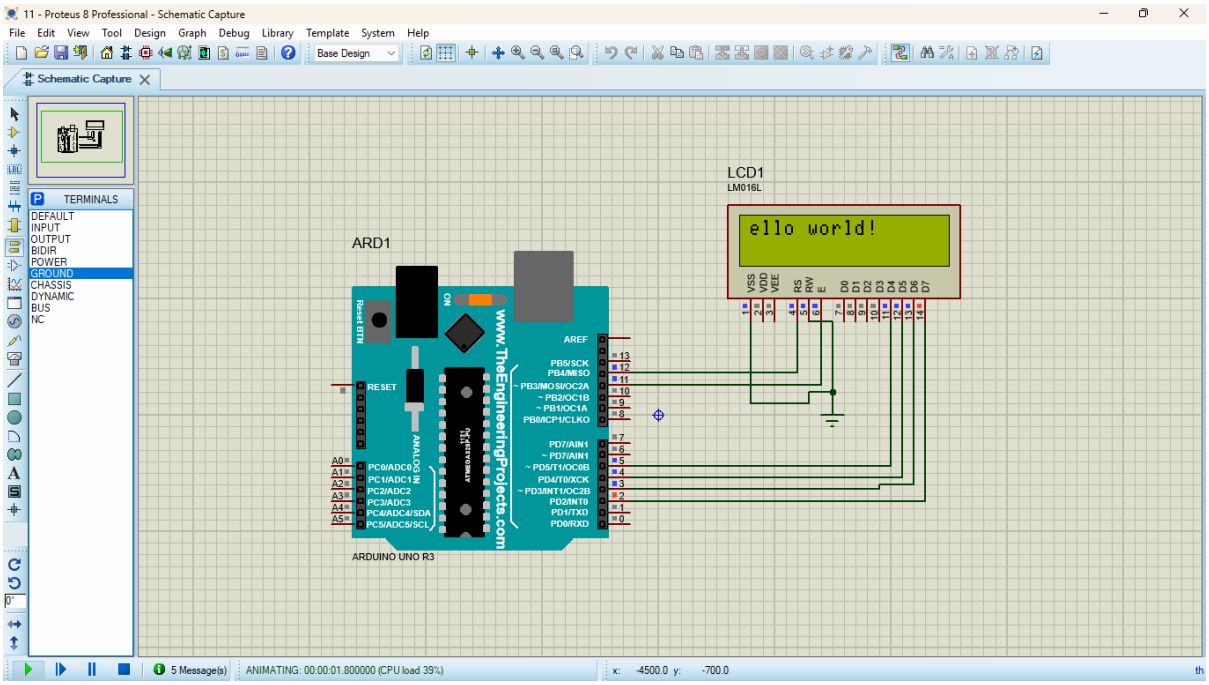
PROGRAM:

Arduino to 16x2 LCD Display using Proteus

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (7, 8, 9, 10, 11, 12);
// your PWM numbers may be different

void setup()  {

lcd.begin(16,  2);
lcd.print("put your message here");

}

void loop ()  {

for (int  i=0;  1<13;  i++)
```

```
}

lcd.scrollDisplayLeft();
delay(700);

}
}
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

SERIAL COMMUNICATION USING ARDUINO WITH PROTEUS

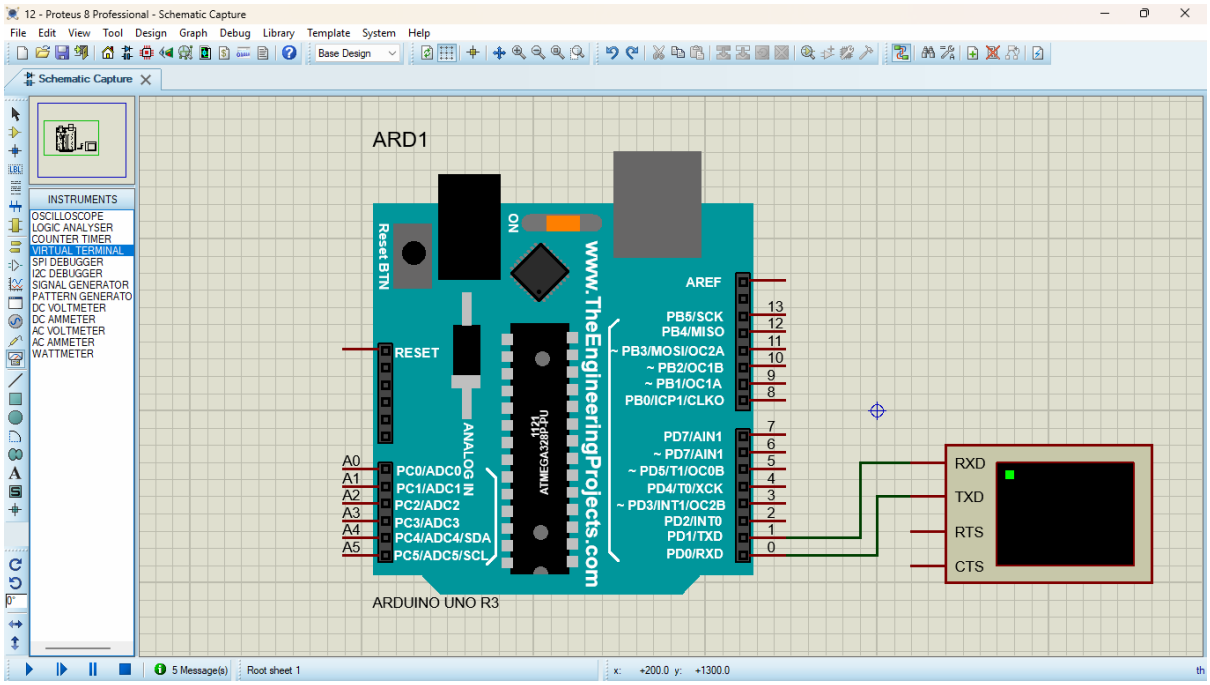
AIM:

To write an Embedded C program for serial communication using Arduino Uno and Proteus

SOFTWARE REQUIRED:

- Proteus 8 software.
- Arduino IDE software

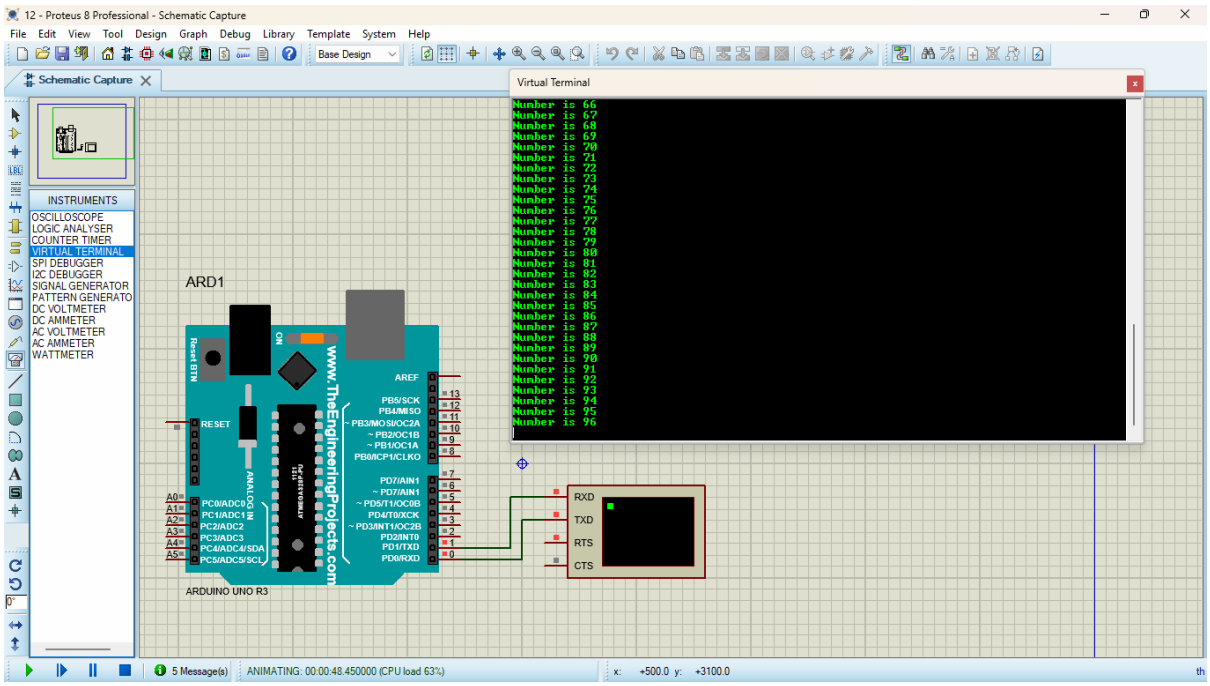
CIRCUIT DIAGRAM:



PROGRAM:

```
void setup()
{
Serial.begin(9600); // send and receive at 9600 baud
}

int number = 0;

void loop()
{
Serial.print("Number is ");
Serial.println(number); // print the number
delay(500); // delay half second between numbers
number++; // to the next number
}
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

TEMPERATURE SENSOR IN PROTEUS USING ARDUINO

AIM:

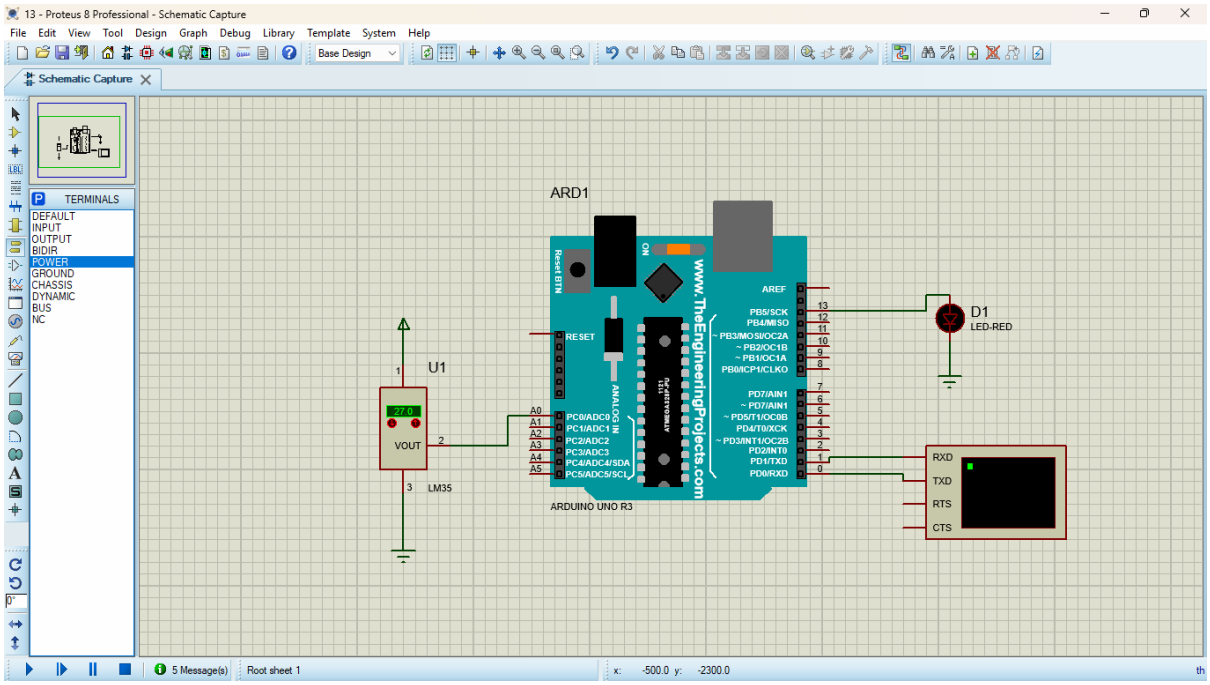
To write an Embedded C program for interfacing Temperature Sensor using Arduino Uno and Proteus

SOFTWARE REQUIRED:

- Proteus 8 software.

- Arduino IDE software

CIRCUIT DIAGRAM:

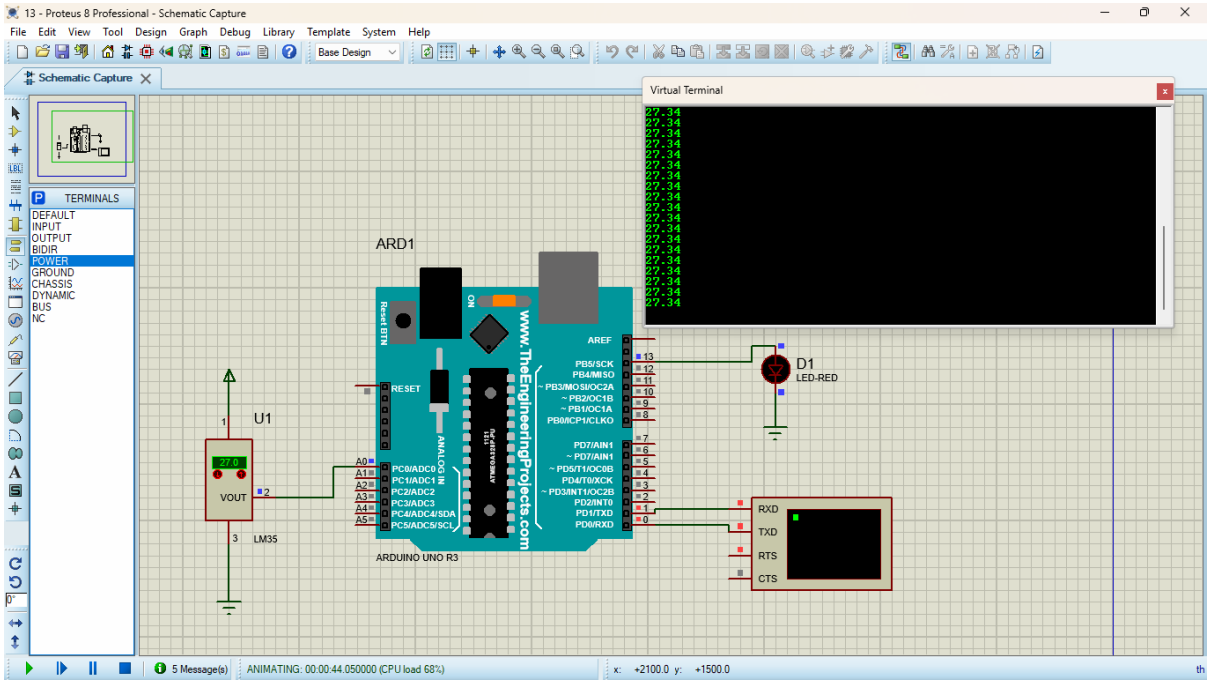


PROGRAM:

```
float temp;
void setup() {
pinMode (13, OUTPUT);
Serial.begin (9600);
}
void loop() {
temp= analogRead (A0);
temp= (temp*500)/1024;
Serial.println (temp);
if (temp>30)
digitalWrite (13, HIGH);
else
digitalWrite (13, LOW);
delay (1000);
```

}

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

GAS SENSOR MQ-2 IN PROTEUS USING ARDUINO

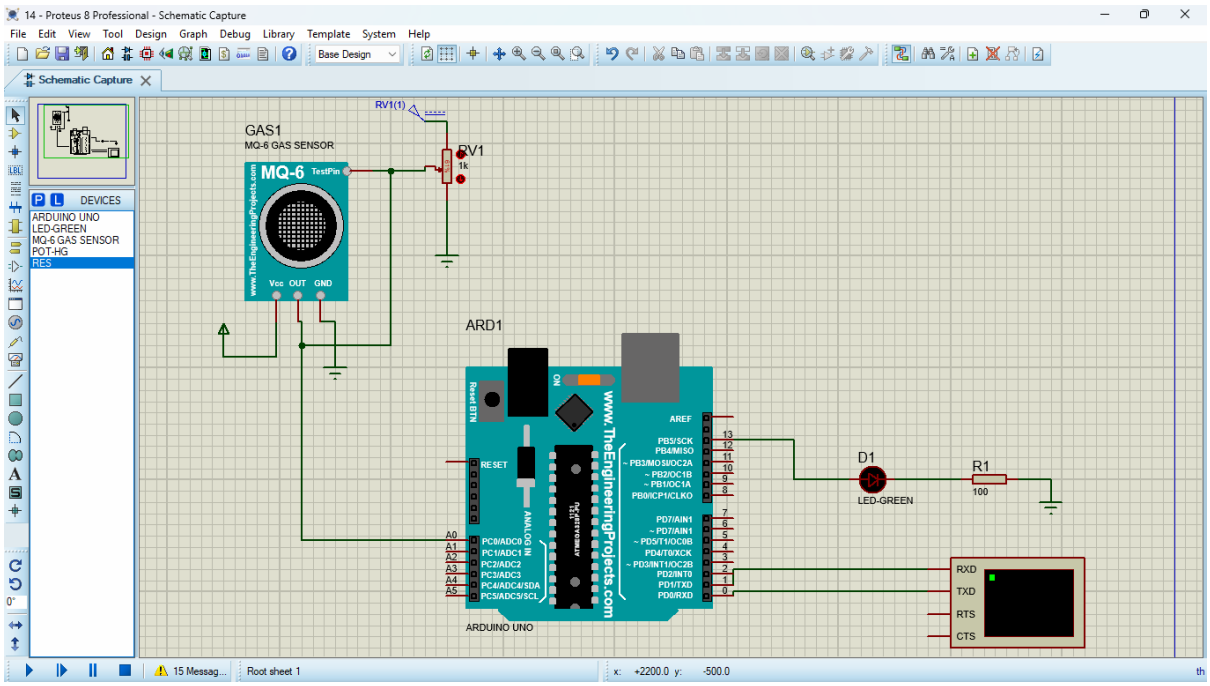
AIM:

To write an Embedded C program for interfacing Gas Sensor MQ-2 using Arduino Uno and Proteus

SOFTWARE REQUIRED:

- Proteus 8 software.
- Arduino IDE software

CIRCUIT DIAGRAM:



PROGRAM:

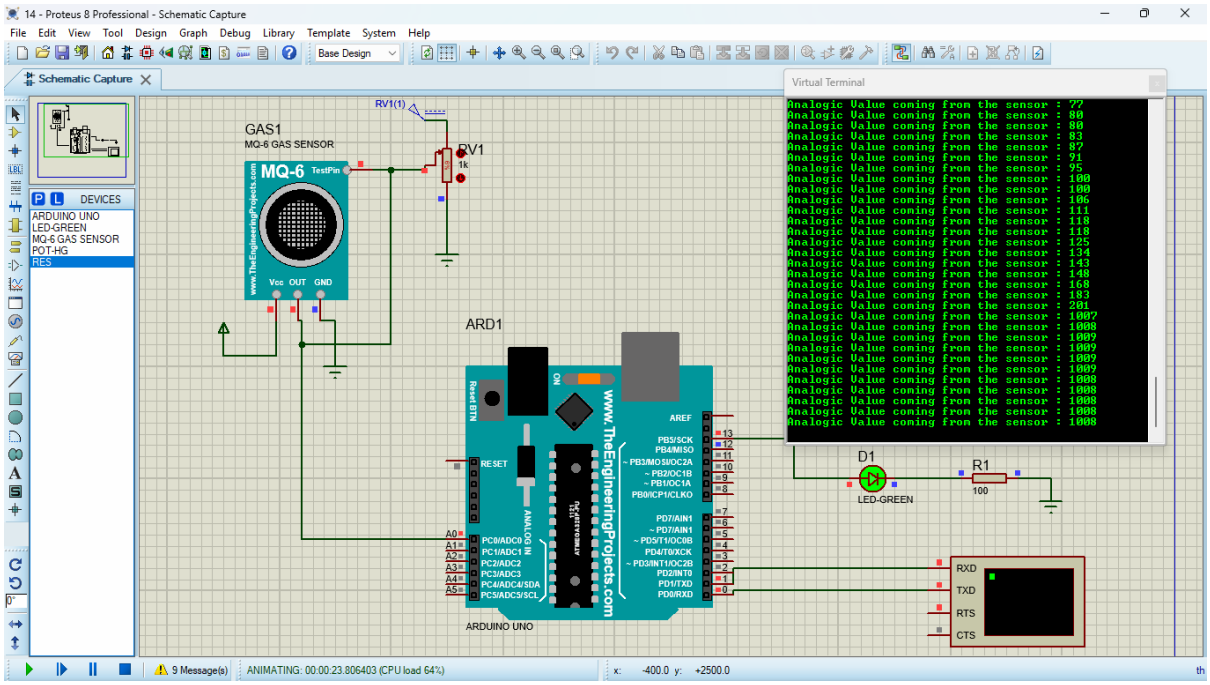
```
#define GreenLed 13
#define Sensor A0


void setup(){
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(A0,INPUT);
  Serial.begin(9600);
}


void loop() {
  int value = analogRead(A0);
  Serial.print("Analogic Value coming from the sensor : ");
  Serial.println(value);
  delay(100);
```

```
  if(value>600){
    digitalWrite(13,HIGH);
  }
  else{
    digitalWrite(13,LOW);
  }
  delay(20);
}
```

Code for MQ6 sensor:

```
int mq6Pin = A0; // MQ-5 analog pin connected to A0

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(mq5Pin);
  Serial.print("MQ5 Sensor Value: ");
  Serial.println(sensorValue);
  delay(1000); // 1-second delay
}
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

ULTRASONIC SENSOR IN PROTEUS USING ARDUINO

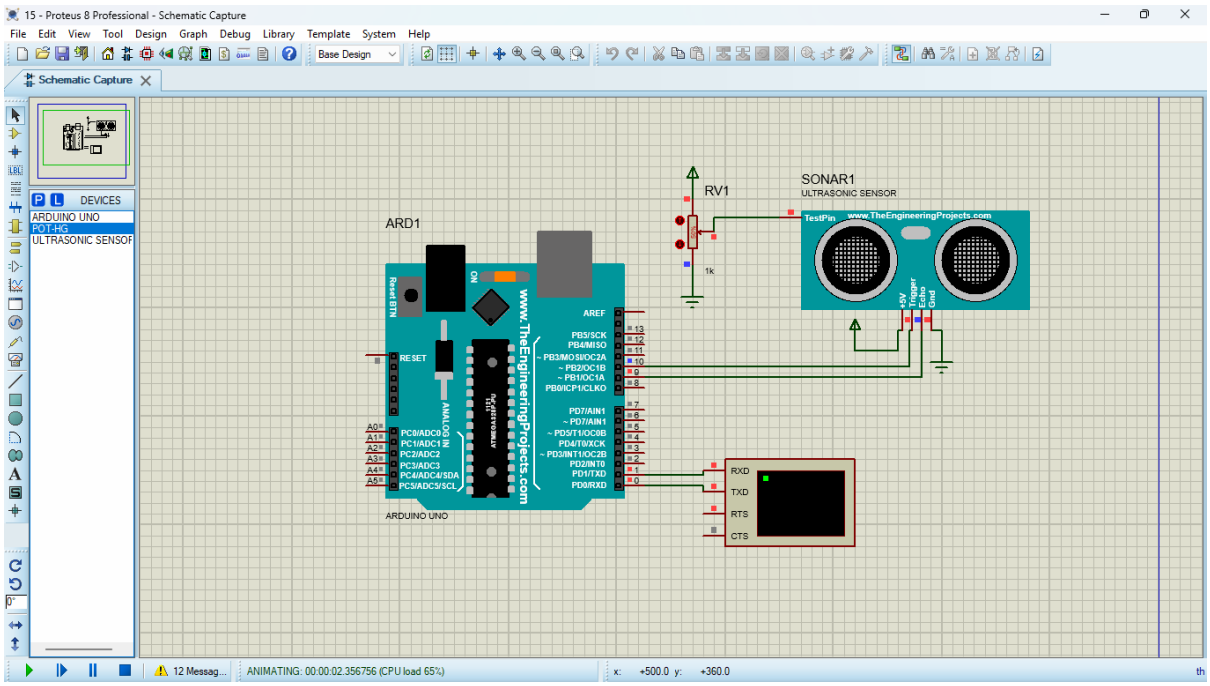
AIM:

To write an Embedded C program for interfacing Ultrasonic Sensor using Arduino Uno and Proteus

SOFTWARE REQUIRED:

- Proteus 8 software.
- Arduino IDE software

CIRCUIT DIAGRAM:

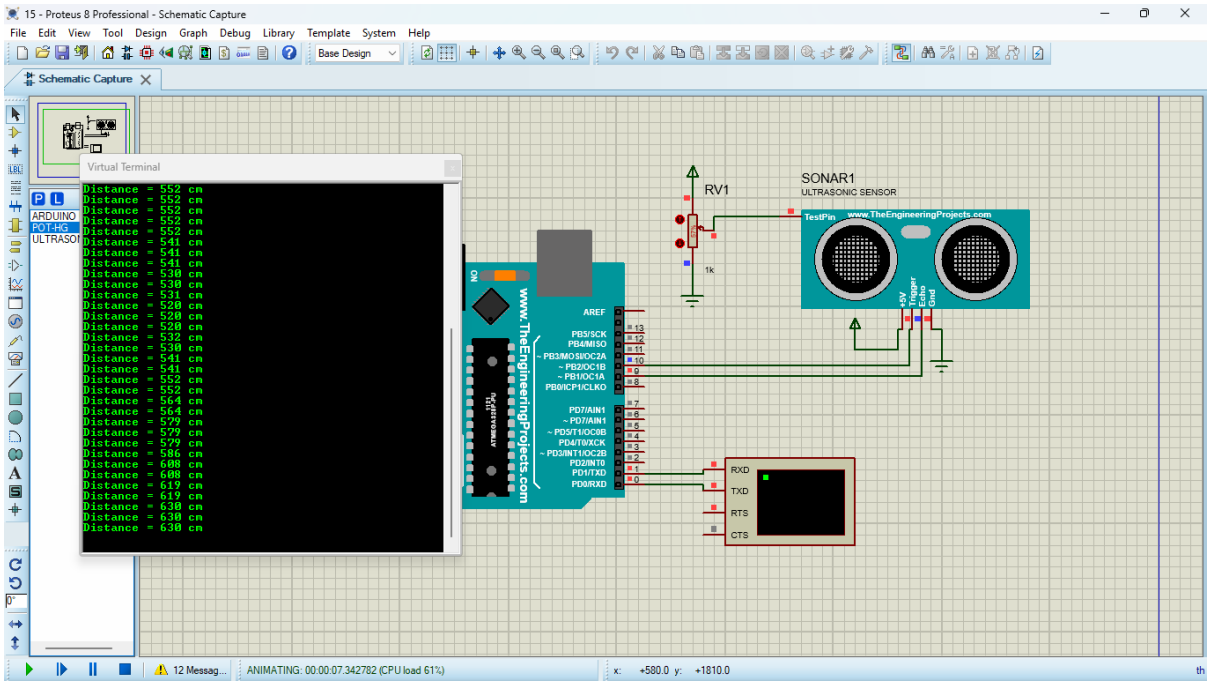


PROGRAM:

```
// Program Sketch
    int trig = 10;
   int echo = 9;
   long duration;
   int cm;
void setup() {
  // put your setup code here, to run once:
  pinMode(trig, OUTPUT);
```

```
  pinMode(echo,INPUT);
  Serial.begin(9600);
}
void loop() {
  // Sketch to calculate ultrasonic distance
        digitalWrite(trig, LOW);
        delayMicroseconds(10);
        digitalWrite(trig, HIGH);
        delayMicroseconds(10);
        digitalWrite(trig, LOW);
        delayMicroseconds(10);
        duration = pulseIn(echo, HIGH);
        cm = (duration/2) * 0.034;

    // Print Value on Serial Monitor
          Serial.print(" Distance = ");
          Serial.print(cm);
          Serial.println(" cm");
}
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

PIR SENSOR IN PROTEUS USING ARDUINO

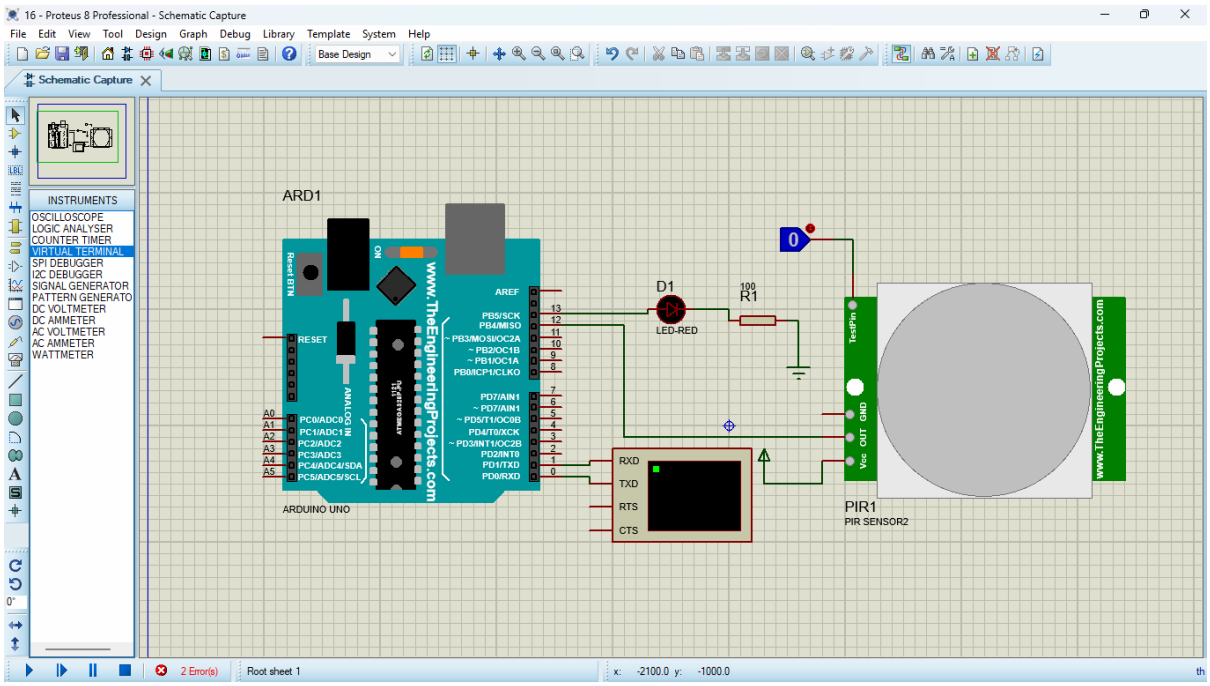
AIM:

To write an Embedded C program for interfacing PIR Sensor using Arduino Uno and Proteus

SOFTWARE REQUIRED:

- Proteus 8 software.
- Arduino IDE software

CIRCUIT DIAGRAM:

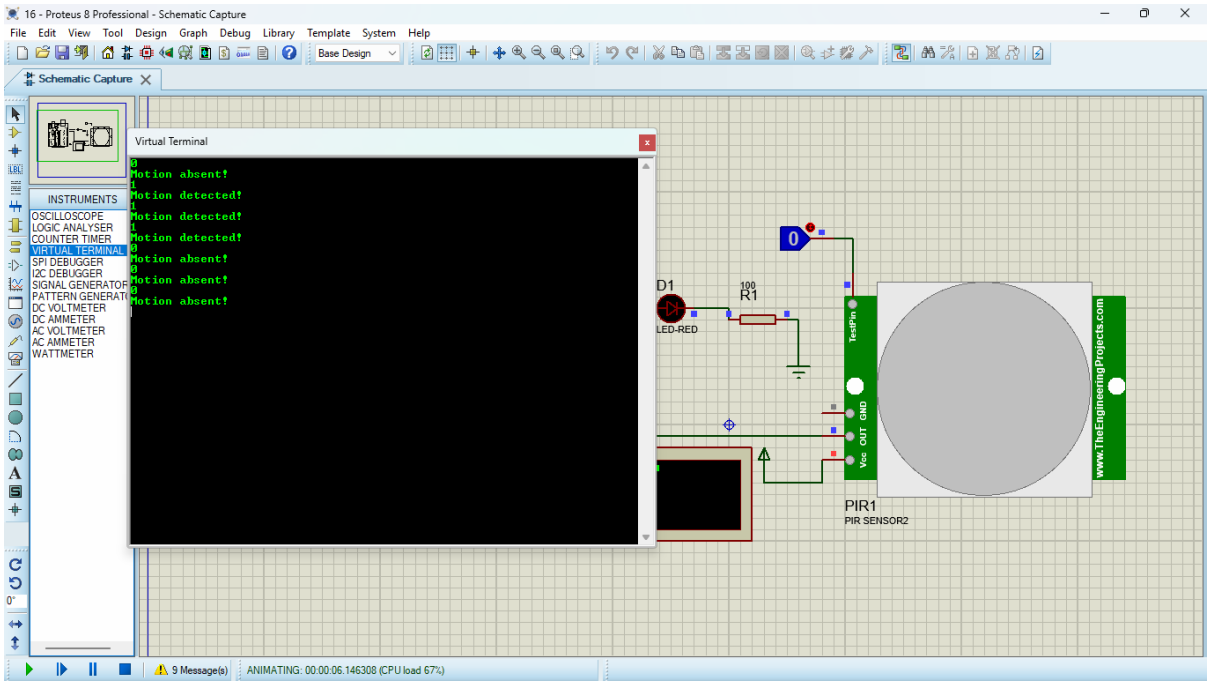


PROGRAM:

```
int Status = 13;  // Digital pin D6
int sensor = 12;  // Digital pin D7
void setup() {
  Serial.begin(9600);
  pinMode(sensor, INPUT);   // declare sensor as input
  pinMode(Status, OUTPUT);  // declare LED as output
}
```

```
void loop() {
  long state = digitalRead(sensor);
  Serial.println(state);
  if(state == HIGH) {
    digitalWrite (Status, HIGH);
    Serial.println("Motion detected!");
    delay(1000);
  }
  else {
    digitalWrite (Status, LOW);
    Serial.println("Motion absent!");
    delay(1000);
    }
}
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

EMBEDDED C PROGRAM TO MULTIPLY TWO 16-BIT BINARY NUMBERS

AIM:

To write an Embedded C program to multiply two 16-bit numbers using Keil software

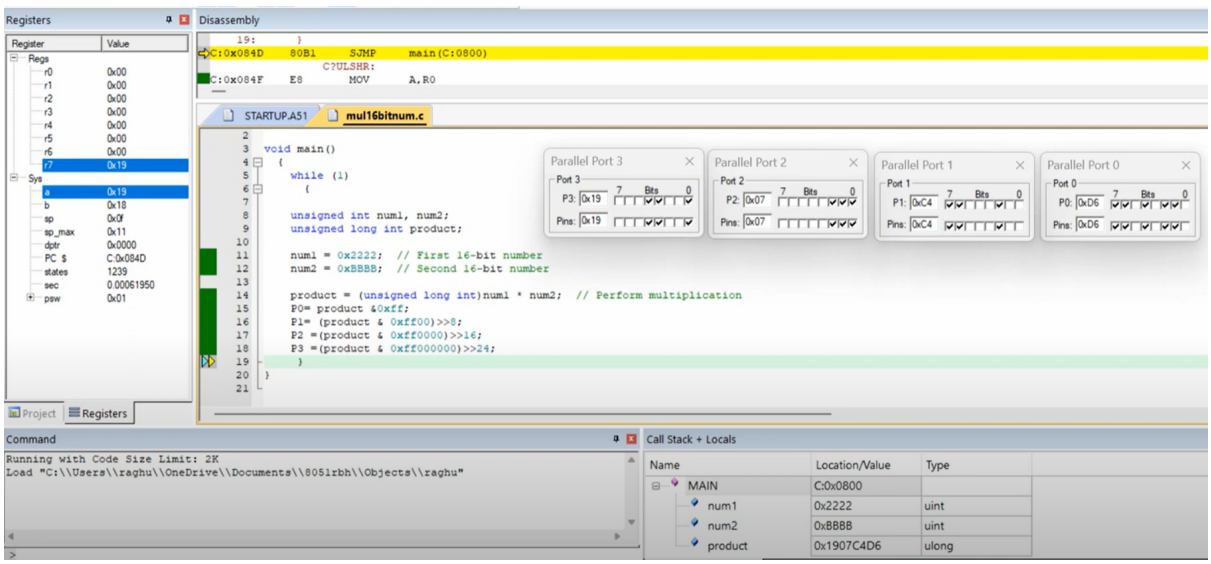
SOFTWARE REQUIRED:

- Keil

Program:

```
#include <reg51.h> void main()
{
while (1)
{
unsigned int num1, num2; unsigned long int product; num1 = 0x2222;
num2 = 0xBBBB;
product = (unsigned long int)num1 * num2; P0= product &0xff;
P1= (product & 0xff00)>>8;
P2 =(product & 0xff0000)>>16; P3 =(product & 0xff000000)>>24;
}
}
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

EMBEDDED C PROGRAM TO FIND THE SUM OF FIRST 10 INTEGER NUMBERS

AIM:

To write an Embedded C program to find the sum of first 10 integer numbers using Keil software

SOFTWARE REQUIRED:

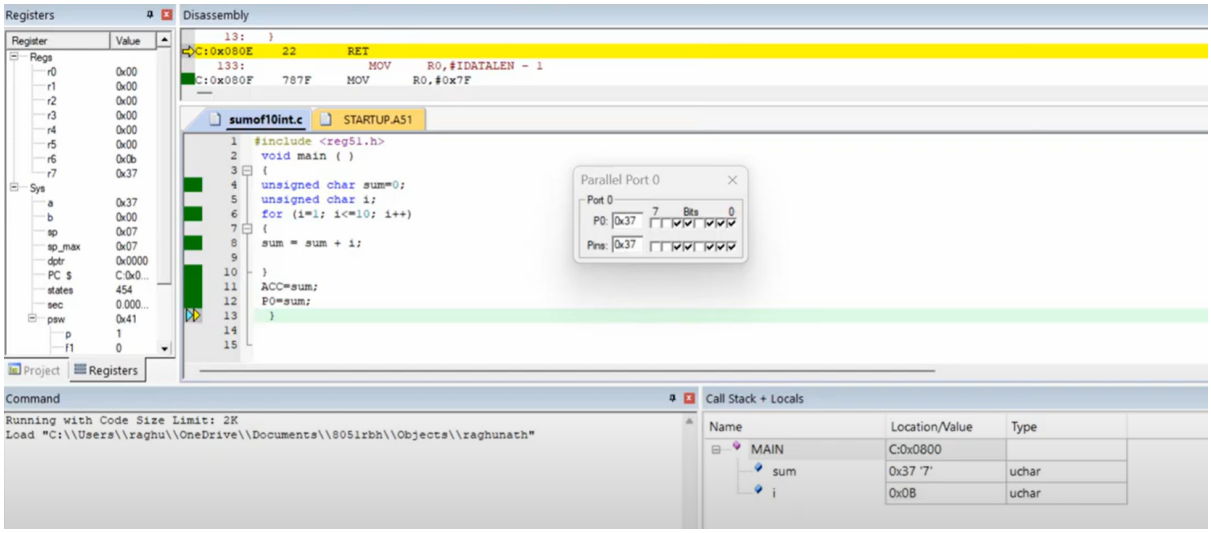
- Keil

PROGRAM:

```
#include <reg51.h> void main ( )
{
unsigned char sum=0; unsigned char i;
for (i=1; i<=10; i++)
{
sum = sum + i;

}
ACC=sum; P0=sum;
}
```

OUTPUT:

1+2+3+4+5+6+7+8+9+10 = 55 (0x37) in Hexadecimal



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

EMBEDDED C PROGRAM TO FIND FACTORIAL OF A GIVEN NUMBERS

AIM:

To write an Embedded C program to find factorial of a given numbers using Keil software.

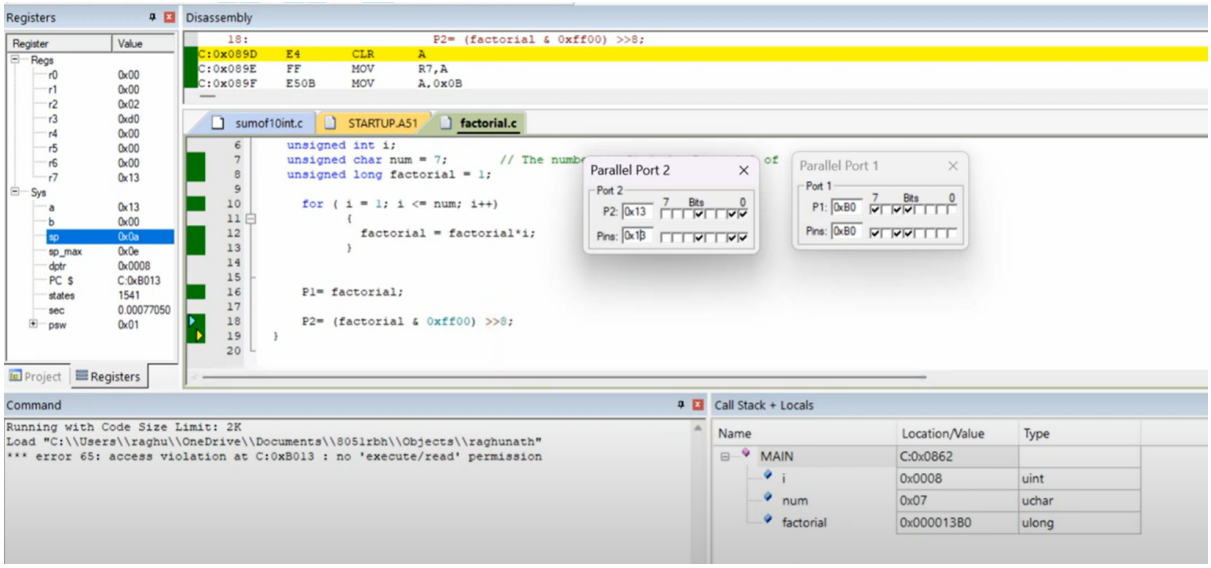
SOFTWARE REQUIRED:

- Keil

PROGRAM:

```
#include <reg51.h> #include <stdio.h>

void main()

{

unsigned int i;

unsigned char num = 12;        // The number to find the factorial of unsigned long factorial = 1;

for ( i = 1; i <= num; i++)

{

factorial = factorial*i;

}

P0= factorial;

P1= (factorial & 0xff00) >>8; P2= (factorial & 0xff0000) >>16;

P3= (factorial & 0xff000000) >>24;

}
```

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

EMBEDDED C PROGRAM TO ADD AN ARRAY OF 16-BIT NUMBERS AND STORE THE 32-BIT RESULT IN INTERNAL RAM

AIM:

To write an Embedded C program to add an array of 16-bit numbers and store the results in internal RAM using Keil software.

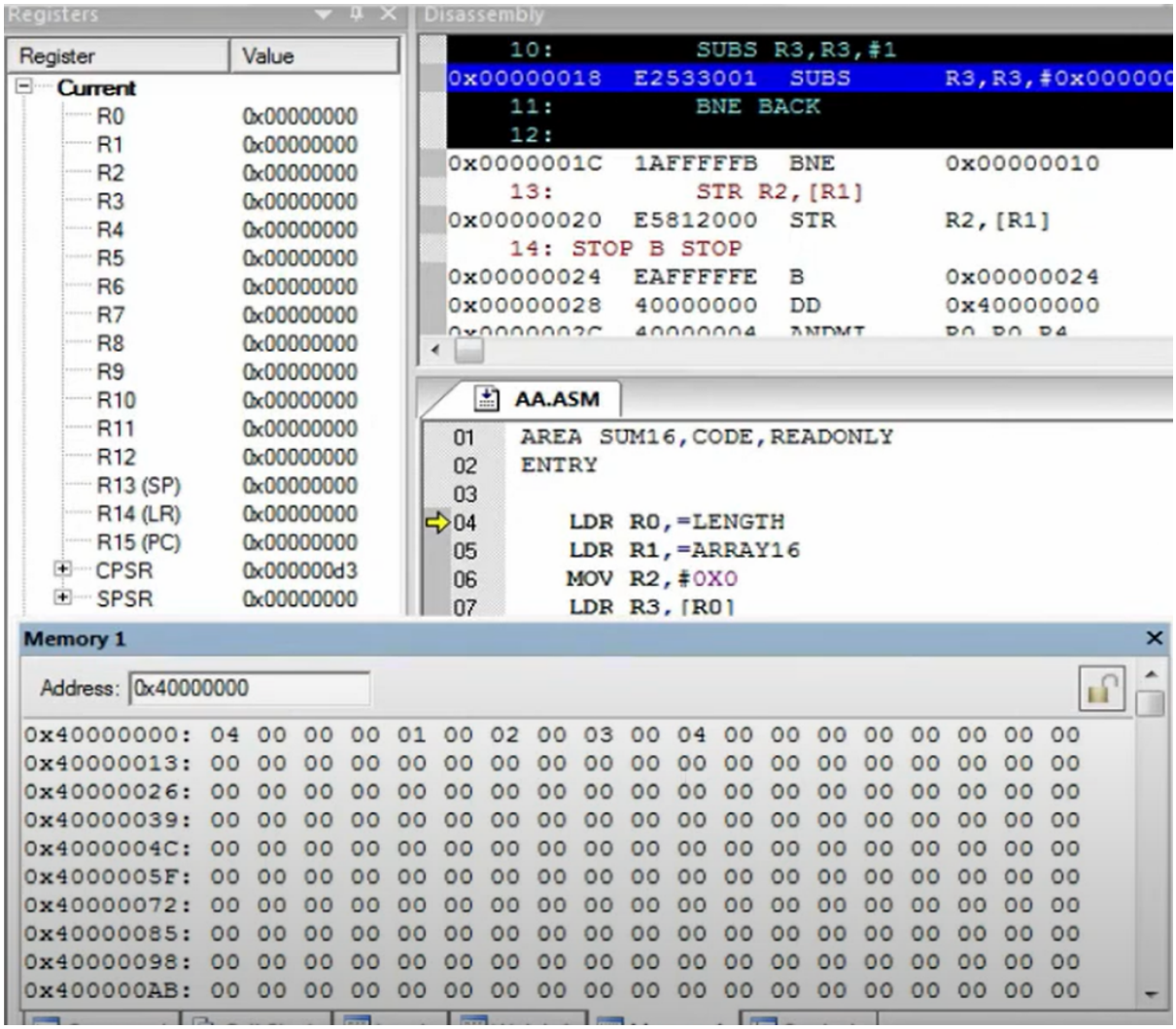
SOFTWARE REQUIRED:

- Keil

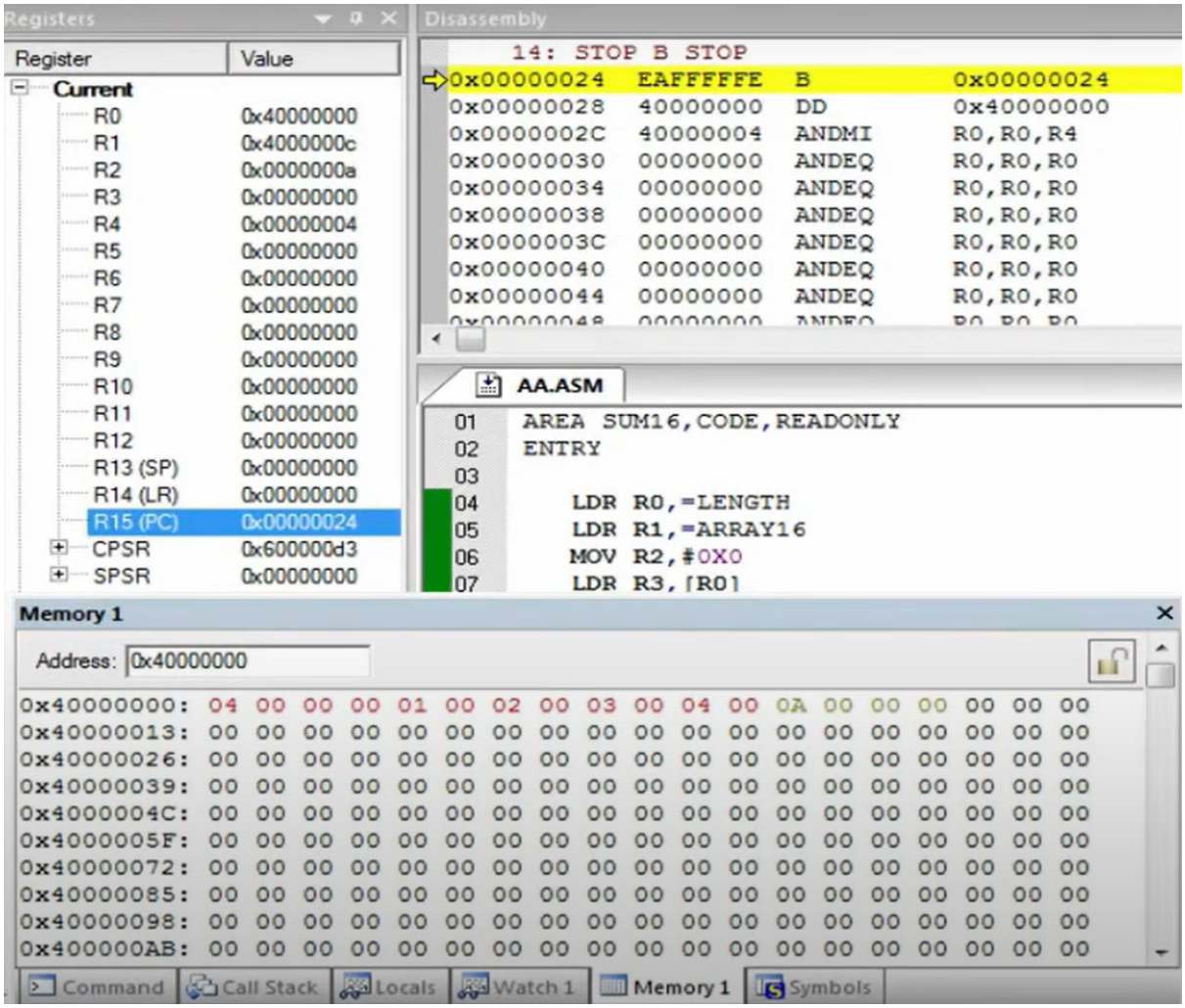
PROGRAM:

```
#include <reg51.h> void main()
{
unsigned int i, array[5] = { 0x1111, 0x2222, 0x8888, 0x4444, 0xABCD }; unsigned long sum = 0;
for (i = 0; i < 5; i++)
{
sum =sum + array[i];
}
P0 = sum & 0xFF;
P1= (sum & 0xff00)>>8; P2=(sum & 0xFF0000)>>16;
}
```

INPUT:


Registers
Register
Value
Current
R0 0x00000000
R1 0x00000000
R2 0x00000000
R3 0x00000000
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000000
CPSR 0x000000d3
SPSR 0x00000000
Disassembly
10: SUBS R3,R3,#1
0x00000018 E2533001 SUBS R3,R3,#0x000000
11: BNE BACK
12:
0x0000001C 1AFFFFFB BNE 0x00000010
13: STR R2,[R1]
0x00000020 E5812000 STR R2,[R1]
14: STOP B STOP
0x00000024 EAFFFFFE B 0x00000024
0x00000028 40000000 DD 0x40000000
AA.ASM
01 AREA SUM16,CODE,READONLY
02 ENTRY
03
04 LDR R0,=LENGTH
05 LDR R1,=ARRAY16
06 MOV R2,#0X0
07 LDR R3,[R0]
Memory 1
Address: 0x40000000
0x40000000: 04 00 00 00 01 00 02 00 03 00 04 00 00 00 00 00 00 00 00
0x40000013: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000026: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000039: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000005F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000072: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000085: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000098: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000AB: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

EMBEDDED C PROGRAM TO DISPLAY “HELLO WORLD” MESSAGE

AIM:

To write an Embedded C program to display “HELLO WORLD” message using Keil software.

SOFTWARE REQUIRED:

- Keil sofware

Program:

#include <reg51.h> #include <stdio.h>

void main (void)

{

SCON = 0x50; /* SCON: mode 1, 8-bit UART, enable rcvr */ TMOD = 0x20; /* TMOD: timer 1, mode 2, 8-bit reload */ TH1 = 0xFD; /* TH1: reload value for 9600 baudrate */

TR1 = 1; /* TR1: timer 1 run */

TI = 1; /* TI: set TI to send first char of UART */ while (1)

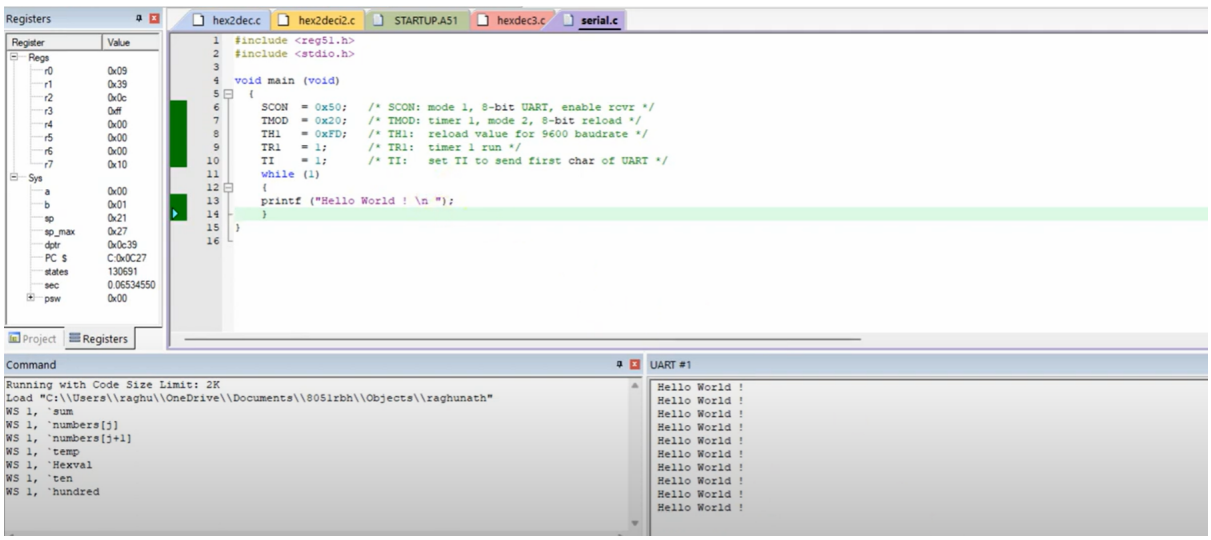
{

printf ("Hello World ! \n ");

}

}

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

EMBEDDED C PROGRAM TO CONVERT THE HEXADECIMAL DATA 0XCFH TO DECIMAL AND DISPLAY THE DIGITS ON PORTS P0, P1 AND P2

AIM:

To write an Embedded C program to convert Hexadecimal numbers into Decimal and display the digits on ports using Keil software.

SOFTWARE REQUIRED:

- Keil software

PROGRAM:

include <reg51.h> void main (void)

{

unsigned char hexa=0xFF; unsigned char hundreds, tens, units;

hexa=hexa/10; P0=B;

units=B;

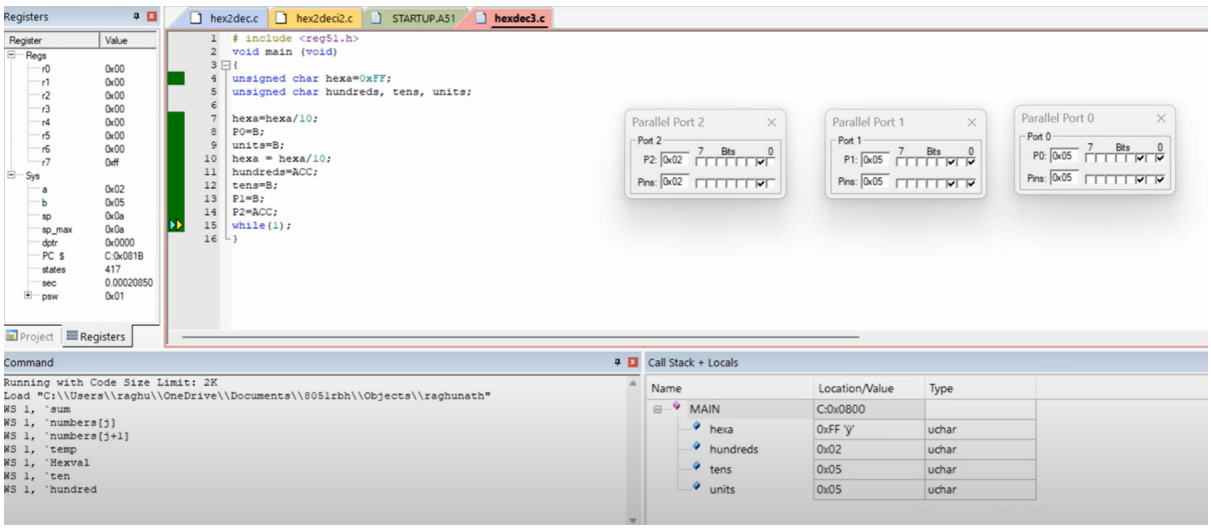
hexa = hexa/10; hundreds=ACC; tens=B;

P1=B; P2=ACC;

while(1);

}

OUTPUT:



RESULT:

Thus the program has been successfully verified and executed.

Exp No:

BLINKING LEDS USING SOFTWARE DELAY ROUTINE IN LPC2148 KIT

AIM:

To write and execute a C program to blink LEDs using software delay routine in LPC 2148 kit

APPARATUS REQUIRED:

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

PROGRAM:

```
#include "lpc214x.h"
void delay (unsigned int k);
void main(void)
{
IODIR0 = 0xFFFFFFFF; //Configure Port0 as output Port
PINSEL0 = 0;                    //Configure Port0 as General Purpose IO
while(1)
{
IOSET0 = 0x0000ff00;  //Set P0.15-P0.8 to '1'
delay(1000);          //1 sec Delay
IOCLR0 = 0x0000ff00;  //Set P0.15-P0.8 to '0'
delay(1000);          //1 Sec Delay
}
}
//Delay Program
//Input - delay value in milli seconds
void delay(unsigned int k)
{
        unsigned int i,j;
        for (j=0; j<k; j++)
```

```
                for(i = 0; i<=800; i++);
}
```

OUTPUT: LEDs P0.15-P0.8 are blinking

RESULT:

Thus the C program to blink LEDs using software delay routine was written and executed in LPC 2148 kit

Exp No:

PROGRAM TO READ THE SWITCH AND DISPLAY IN THE LEDS USING LPC2148 KIT

AIM:

To write and execute C program to read the switch and display in the LEDs using LPC2148 kit

APPARATUS REQUIRED:

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

PROGRAM:

```
#include "lpc214x.h"
int main(void)
{
  unsigned int sw_sts;
  IODIR0 = 0x0000ff00; //Configure Port0
        PINSEL0 = 0;                    //Configure Port0 as General Purose IO
while(1)
{
  sw_sts = IOPIN0;
  IOSET0 = 0x0000ff00;   //Set P0.15-P0.8 to '1'
  IOCLR0 = sw_sts >> 8;  //Set P0.15-P0.8 to '0'
}
}
```

OUTPUT: LEDs P0.15-P0.08 displayed the bits entered in the switches

RESULT:

Thus C program was written read the switch and display in the LEDs using LPC2148 kit

Exp No:

DISPLAY A NUMBER IN SEVEN SEGMENT LED IN LPC2148 KIT

AIM:

To write and execute C program to display a number in seven segment LED in LPC2148 kit

APPARATUS REQUIRED:

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

PROGRAM:

```
//SEVEN SEGMENT LED DISPLAY INTERFACE IN C
/* Program to Count 0-9 and Display it in 7 segment Display (MUX) DS4
 * Display Select DS3 ==> "P0.13"  Enable --> '0',  Disable --> '1'
 * Display Select DS4 ==> "P0.12"  Enable --> '0',  Disable --> '1'
 */

/* Segment Connection Display 1 & 2            Enable --> '1',  Disable --> '0'
 *-----------------------------------------------------------------
 * MSB                                        LSB
 *   Dp     G      F      E      D      C      B      A
 * P0.23  P0.22  P0.21  P0.20   P0.19  P0.18  P0.17  P0.16
 *   0      0      0      0      0      1      1      0   --> 6 => '1'
 *-----------------------------------------------------------------*/

#include <LPC214X.H>
#define DS3   1<<13              // P0.13
#define DS4   1<<12              // P0.12
#define SEG_CODE 0xFF<<16     // Segment Data from P0.16 to P0.23
unsigned char const seg_dat[]={0x3F, 0x6, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x7, 0x7F,
0x67};
```

```
void delayms(int n)
{
int i,j;
for(i=0;i<n;i++)
 {for(j=0;j<5035;j++)   //5035 for 60Mhz ** 1007 for 12Mhz
    {;}
 }
}

int main (void)
{
  unsigned char count;

PINSEL0 = 0; // Configure Port0 as General Purpose IO => P0.0  to P0.15
PINSEL1 = 0; // Configure Port0 as General Purpose IO => P0.16 to P0.31
IODIR0 = SEG_CODE | DS3 | DS4; //Configure Segment data & Select signal as output
IOSET0 = SEG_CODE | DS3 ;  //Disable DS3 display
IOCLR0 = DS4;  //Enable DS4 Display
count  = 0; //Initialize Count

//Display Count value
IOCLR0 = SEG_CODE;
IOSET0 = seg_dat[count]<<16;

while(1)
 {
          delayms(1000); //1 sec delay
    count++;              //Increment count
    if(count>9) count=0;  //Limit 0-9
```

```
    //Display Count value
    IOCLR0 = SEG_CODE;
    IOSET0 = seg_dat[count]<<16;


   }
}
```

OUTPUT: 7-Segment display counting from 0 to 9

RESULT:

Thus C program, was written and executed to display a number in seven segment LED in LPC2148 kit

Exp No:

SERIAL TRANSMISSION AND RECEPTION USING ON-CHIP UART IN LPC2148 KIT

AIM:

To write and execute C program for serial transmission and reception using on-chip UART in LPC2148 kit.

APPARATUS REQUIRED:

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

PROGRAM:

```
#include <lpc214x.h>
void UART0_Init(void)
{
 PLL0CON = 0;
 PLL0FEED=0xAA;
 PLL0FEED=0x55;
 VPBDIV = 1;
 // Fpclk = 12.000.000 MHz
 // DLM,DLH = Fpclk / (19200*16) = 39 = 0x27
 PINSEL0 |= 0x5; // Select UART0 RXD/TXD
 U0FCR = 0; // Disable FIFO's
 U0LCR = 0x83; // 8N1, enable Divisor latch bit
 U0DLL = 0x27; // baud rate fixed to 19200 @ PCLK = 12 Mhz
 U0DLM = 0;
 U0LCR = 3; // Disable Divisor latch bit
}
/*----------------------------------------------------------------*/
/* Function to send one char. to Serial Port */
void sout(unsigned char dat1)
{
```

```
 while(!(U0LSR & 0x20));//Wait for Tx Buffer Empty
 U0THR = dat1;   //Send to UART1
}
/*---------------------------------------------------*/


int main (void)
{  int dat;
   UART0_Init();
 do
  {
   if(U0LSR & 1)  /* Check for RDR (Receiver Data Ready)command */
    {
      dat = U0RBR;// Receive Data from Srial Port
      sout(dat);  // Send Data to Srial Port
          }
   }while(1);
}
```

OUTPUT: Data was serially transmitted

RESULT:

Thus a C program was Written and executed for serial transmission and reception using on-chip UART in LPC2148 kit.

Exp No:

ACCESS AN INTERNAL ADC AND DISPLAY THE BINARY OUTPUT IN LEDS IN LPC2148 KIT

AIM:

To write and execute C program for accessing an internal ADC and display the binary output in LEDS in LPC2148 kit.

APPARATUS REQUIRED:

Keil uVision5 Software

Philips Flah Programmer

LPC 2148 kit

PROGRAM:

```
#include <LPC214X.H>
#define LEDS  0xFF<<8   //LED => P0.8 to P0.15
////////////////////////////////////////
/*--- ADC Signal Declaration */
////////////////////////////////////////
#define AD0_1 1<< 24
#define CLK_DIV 1<<8
#define PDN 1<<21
#define SOC 1<<24
#define BURST 1<<16
#define DONE 1<<31

/*-------------------------------------------------------*/
//Delay Program
//Input - delay value in milli seconds
void delay(unsigned int k)
{
        unsigned int i,j;
        for (j=0; j<k; j++)
                for(i = 0; i<=800; i++);
```

```
}
/*-------------------------------------------------------------*/
void adc_init()
{
unsigned long int ADC_CH;

        ADC_CH = 0 | 1 << 1; //Channel AD0.1
   AD0CR = SOC | PDN | CLK_DIV | ADC_CH | BURST ;
}
/*-------------------------------------------------------------*/
unsigned int adc_read( unsigned char channel)
{
        unsigned int aval;
        unsigned long int val;

   if (channel == 1) val = AD0DR1;
   else if (channel == 2) val = AD0DR2;
   else if (channel == 3) val = AD0DR3;

   val = val >> 6;
   val = val & 0x3FF;
   aval = val;
   return (aval);
}
/*-------------------------------------------------------------*/
//////////////////////////
/*----Main Program------*/
//////////////////////////
int main(void)
{
```

```
unsigned int tp1;


  IODIR0   = LEDS; //Configure Port0 as output Port
  PINSEL0 = 0; //Configure Port0 as General Purpose IO
  PINSEL1 = 0 | AD0_1; // Enable AD0.1


    adc_init(); //Initialise on-chip ADC


 do
  {  tp1 = adc_read(1); // Channel AD0 0.1
                    tp1 = tp1 >> 2;  // ADC 10 bit But LED 8bit, Truncate lsb 2 bits
     IOSET0 =  LEDS;  //Switch OFF all LEDS
     IOCLR0 = tp1 << 8;  //Set VAlue
     delay(1000);
  }while(1);
}
```

OUTPUT: The Potentiometer knob was adjusted to generate Analog input and Digital display is observed

RESULT:

Thus C program was Written and executed for accessing an internal ADC and display the binary output in LEDS in LPC2148 kit.