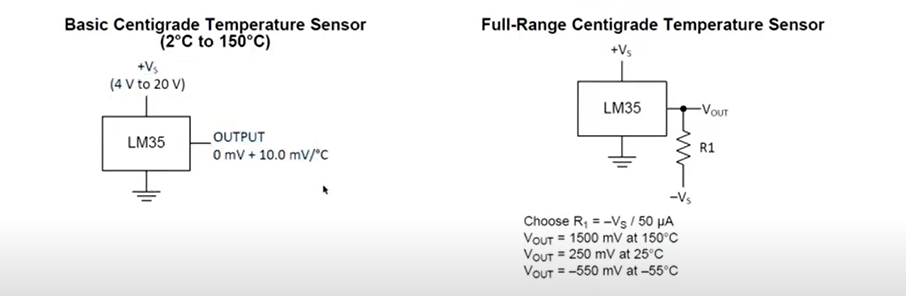
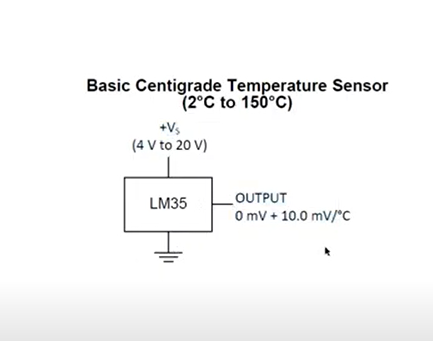
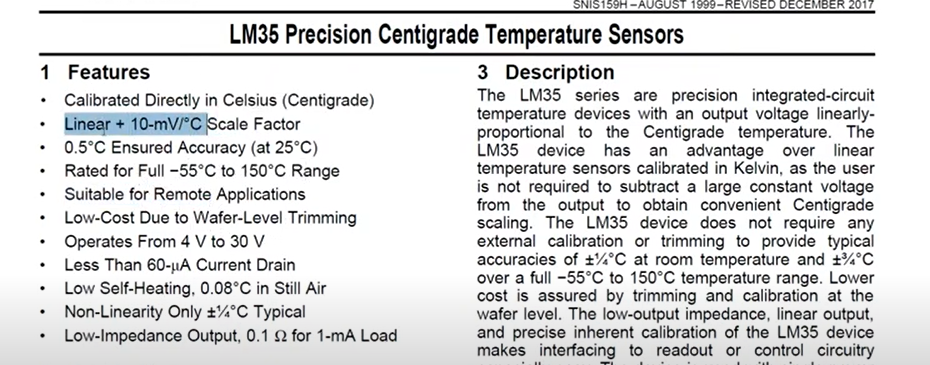
**MICROPROCESSOR AND MICROCONTROLLER-BASED DESIGNS**

**PROJECT FINAL REPORT**

**TEMPERATURE SENSOR USING MICROCONTROLLER**

**PROGRESS WEEK 1: DIGITAL THERMOMETER**

****

**EQUIPMENT USED:**

* PIC18F452
* LM35 (Temperature sensor)
* Potentiometer 10k
* Resister 1k
* Wires
* LCD 16X2
* Crystal oscillator 4Mz
* Capacitors 22pf

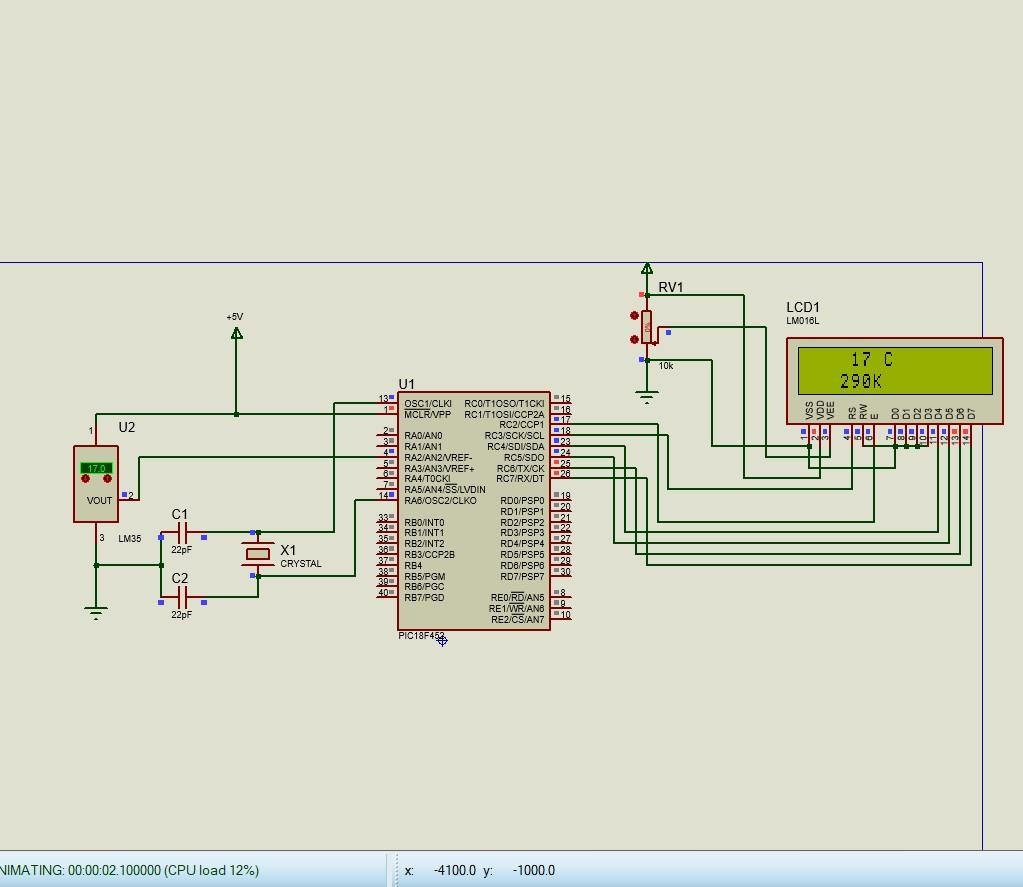
**COMPILER USED:**

* The compiler we used in this project is **MIKROC PRO 7.6.** The reason is that it is the best compiler for microcontrollers in C language.

**C LANGUAGE CODE:**

|  |
| --- |
| // connections between lcd and pic sbit LCD\_RS at RC3\_bit; sbit  LCD\_EN at RC2\_bit; sbit LCD\_D4 at RC4\_bit; sbit LCD\_D5 at RC5\_bit; sbit LCD\_D6 at RC6\_bit; sbit  LCD\_D7 at RC7\_bit;    //directions  sbit LCD\_RS\_Direction at TRISC3\_bit;  sbit LCD\_EN\_Direction at TRISC2\_bit; sbit LCD\_D4\_Direction at TRISC4\_bit; sbit  LCD\_D5\_Direction at TRISC5\_bit; sbit  LCD\_D6\_Direction at TRISC6\_bit; sbit  LCD\_D7\_Direction at TRISC7\_bit;      //prototype void  mydelay(int); unsigned int AD\_Read(); void  ADC\_ini();      char tp1[7]; char tp2[7]; unsigned int temp\_celc; unsigned int temp\_kel;    void main()  {  TRISA=0xFF;  lcd\_Init();  mydelay(10); lcd\_out(1,1,"Temperature is:-"); mydelay(10); delay\_ms(500);  while(1)  {  ADC\_ini();  temp\_celc= AD\_read()\*0.489;  temp\_kel=temp\_celc + 273; IntTostr(temp\_celc,tp1);  IntTostr(temp\_kel,tp2);  Lcd\_Cmd(\_LCD\_CLEAR);  Lcd\_cmd(\_LCD\_FIRST\_ROW);  Lcd\_cmd(\_LCD\_MOVE\_CURSOR\_LEFT);  mydelay(2); lcd\_out(1,1,tp1); lcd\_out(1,7," C"); mydelay(10); lcd\_out(2,1,tp2); lcd\_out(2,7,"K");  lcd\_cmd(\_LCD\_CURSOR\_OFF);  mydelay(5000);  }    }    void mydelay(int del)  {  int n=0; int  b=0; for(; n<del; n++)  {  for(; b<1000; b++){int s=0;}  }  }      void ADC\_ini()  {  ADCON0=0b11010101; //RC OSC, CHANNEL 2, GO=1, ADC ON ADCON1=0b10000000; //Right justified, vref=5 volt mydelay(600);  }  unsigned int AD\_read()  {    unsigned int ADC\_RES=0;  while(ADCON0.DONE=1){};  ADC\_RES=ADRESL;  ADC\_RES|=((unsigned int)ADRESH)<<8; return ADC\_RES;  } |

**PROGRESS WEEK 2: DIGITAL THERMOMETER**



**PROGRESS WEEK 3: DIGITAL THERMOMETER**

Diagram

Description automatically generated with medium confidence

**PROGRESS WEEK 4: TEMPERATURE SENSOR**

**EQUIPMENT USED:**

* Arduino-UNO
* LM35 (Temperature sensor)
* Resisters 100ohm
* Wires
* Red and Green Led

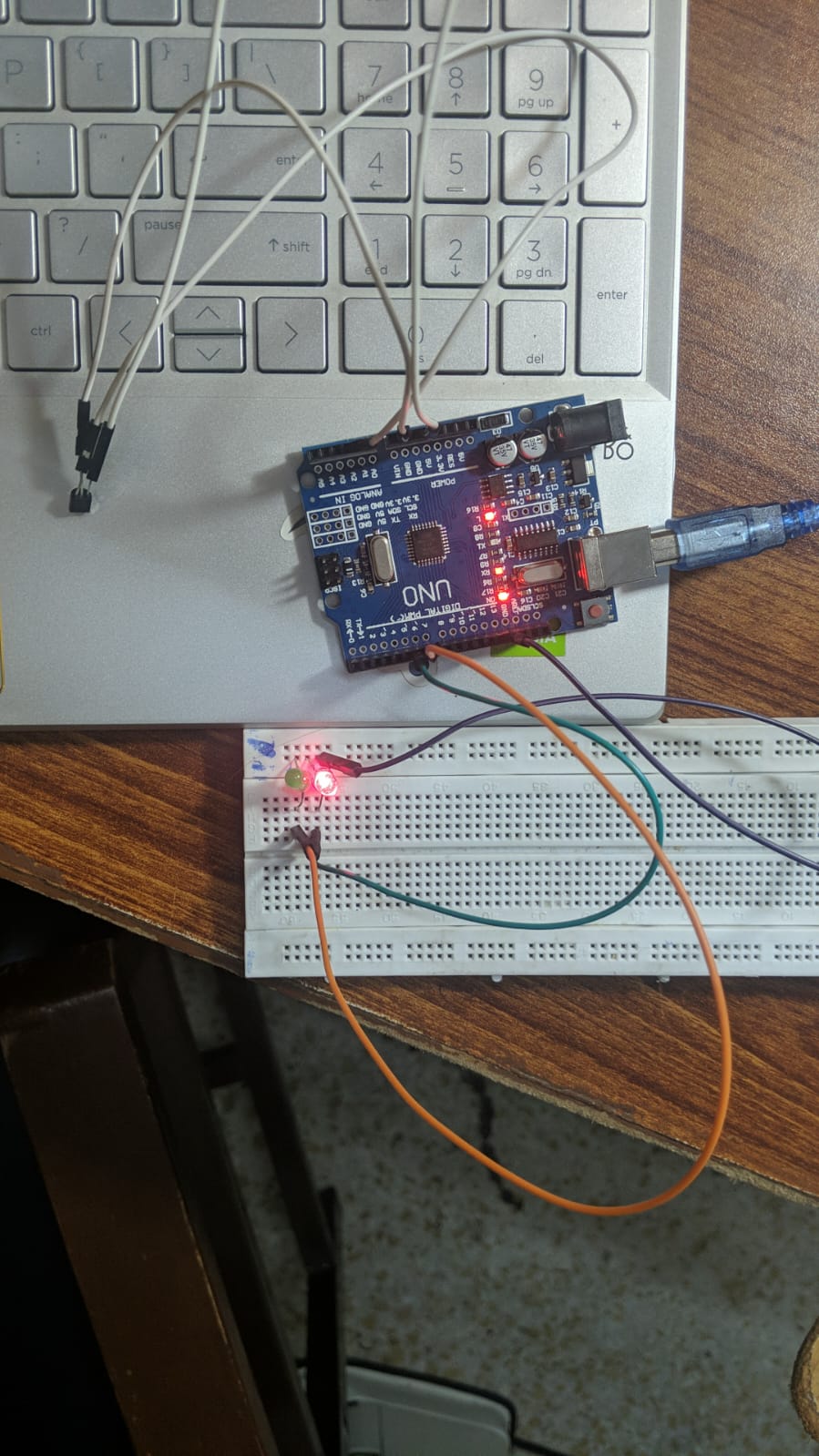
**COMPILER USED:**

* ARDUINO is a very handy and very powerful Integrated Development Editor that will let you write code as well as upload it to an Arduino board for testing purposes. This cross-platform IDE works along with an Arduino controller for writing, compiling as well as uploading the code to the board.

**CODE:**

|  |
| --- |
| #define LM35 A0  int GREEN=7;  int RED=6;  void setup()  {  Serial.begin(9600);  pinMode (7,OUTPUT);  pinMode (6,OUTPUT);    }  void loop()  {  float Imvalue = analogRead(LM35);  float temp=(Imvalue\*500)/1023;  Serial.println(temp); //Temperature in clecius  delay(1000);  if (temp>60)  {  digitalWrite(GREEN,LOW);  digitalWrite( RED,HIGH);  }  else  {  digitalWrite(GREEN,HIGH);  digitalWrite(RED,LOW);  delay (200);  }  } |

**CONNECTIONS:**

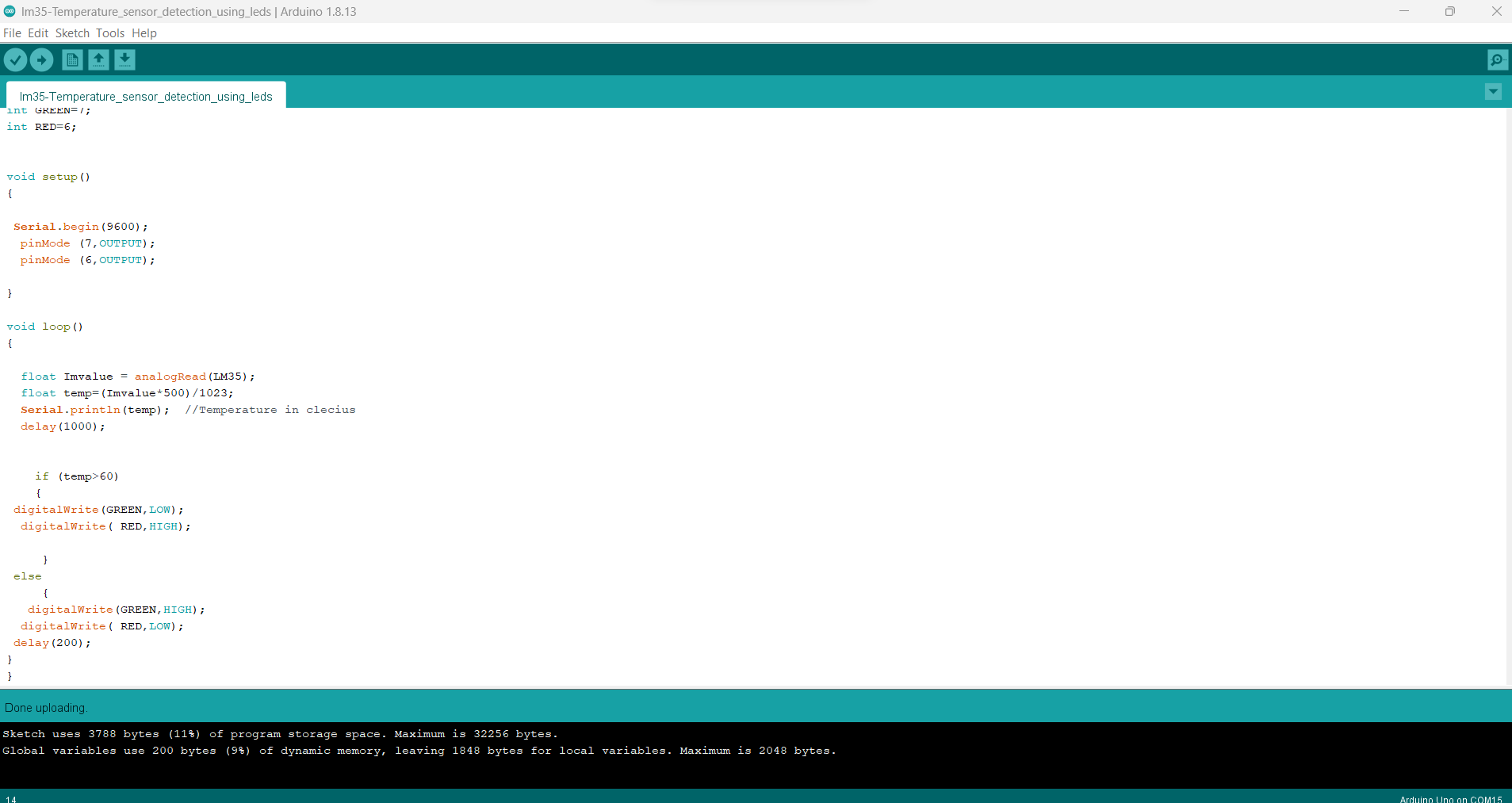
****

1) Connect the Components as per the Schematic Circuit Diagram

2) Upload "Lm35.ino"

**OUTPUT:**

1. **Verify code**
2. **Upload code**
3. **Open screen monitor**

****

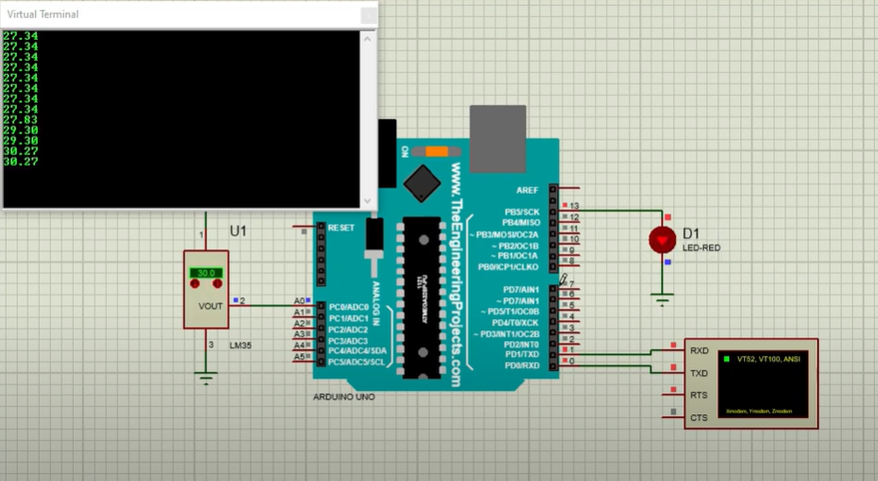
**SCREEN MONITOR OUTPUT:**

**PROTEUS SIMULATION:**

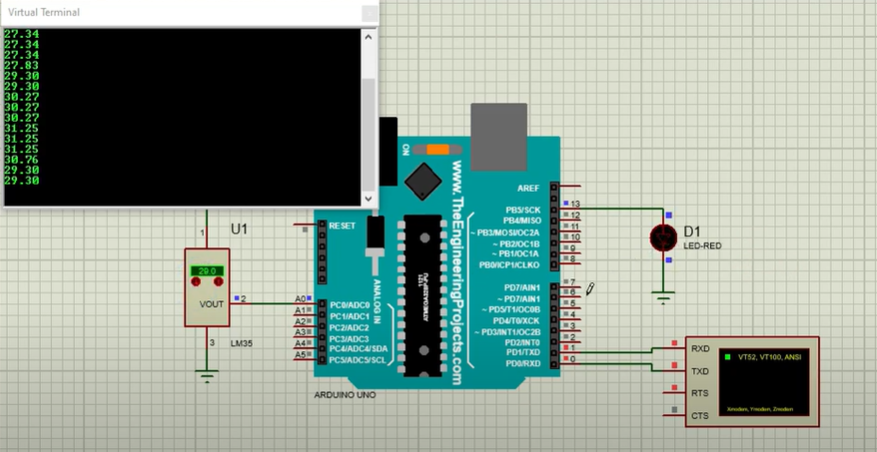
**Chart, bar chart

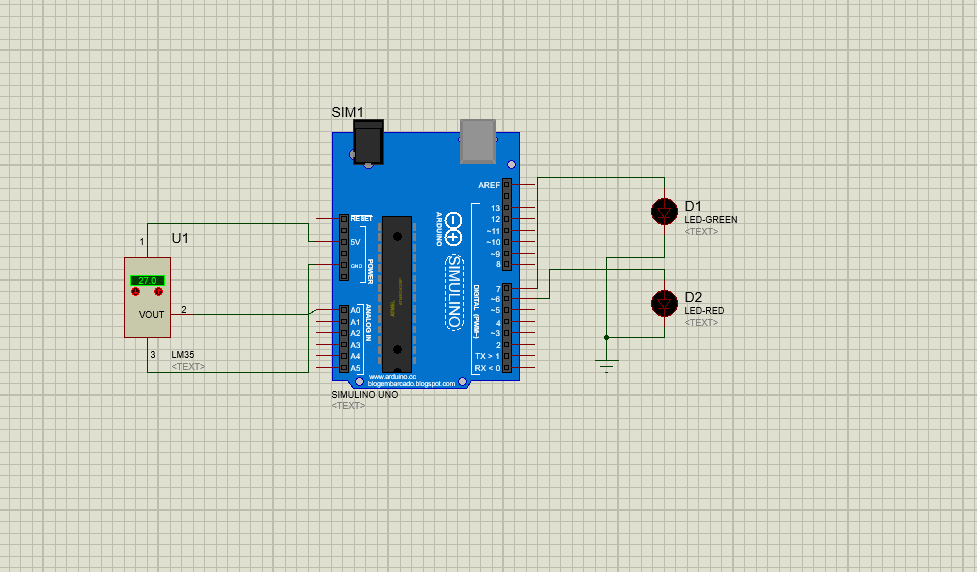
Description automatically generated**

Once we increase the temperature beyond the limit(30) Then led starts to glow.

****

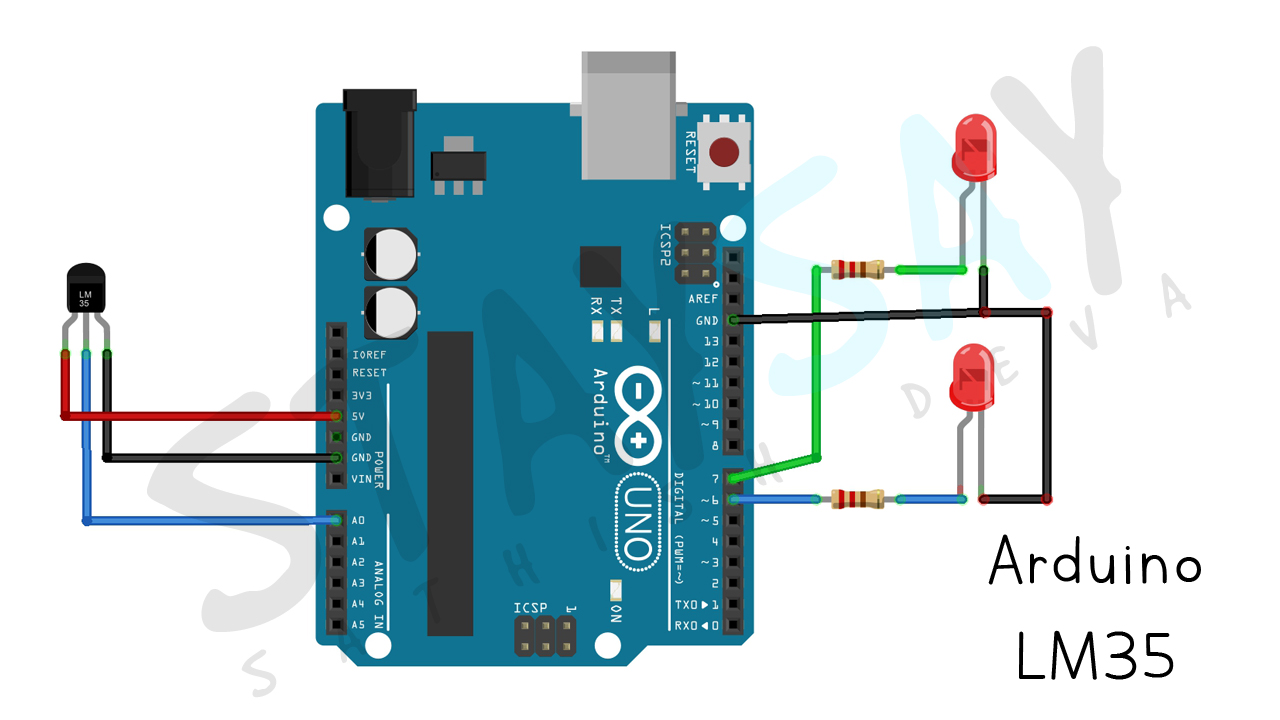
Likewise, if we decrease temperature Led stops glowing.

****

****

Diagram

Description automatically generated



**LIMITATIONS:**

The project does not give negative temperature just a low and high indication.

No Lcd display implemented.