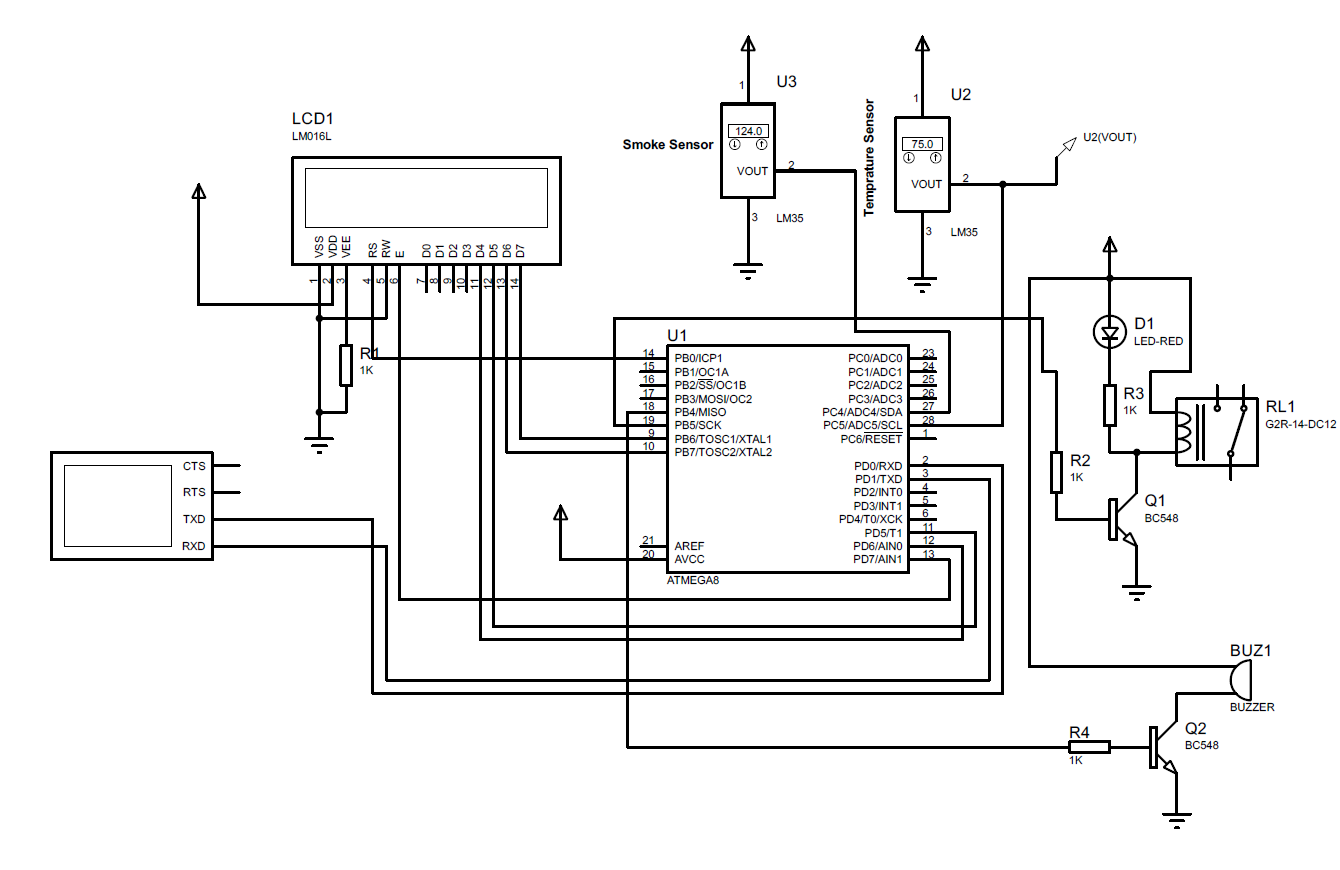
**Object-** GSM based SMS Alert Fire Alarm System using Arduino (ATMEGA8)

**Software Used:-** Proteus 8.6 Professional & Arduino Atmega8

**Components Used :**- Atmega8, BC548, LM35, LM016L LCD, RED LED, G2R-14-DC12 Relay , RESISTANCE , BUZZER, BUTTON, & VIRTUAL TERMINAL AS GSM MODULE.

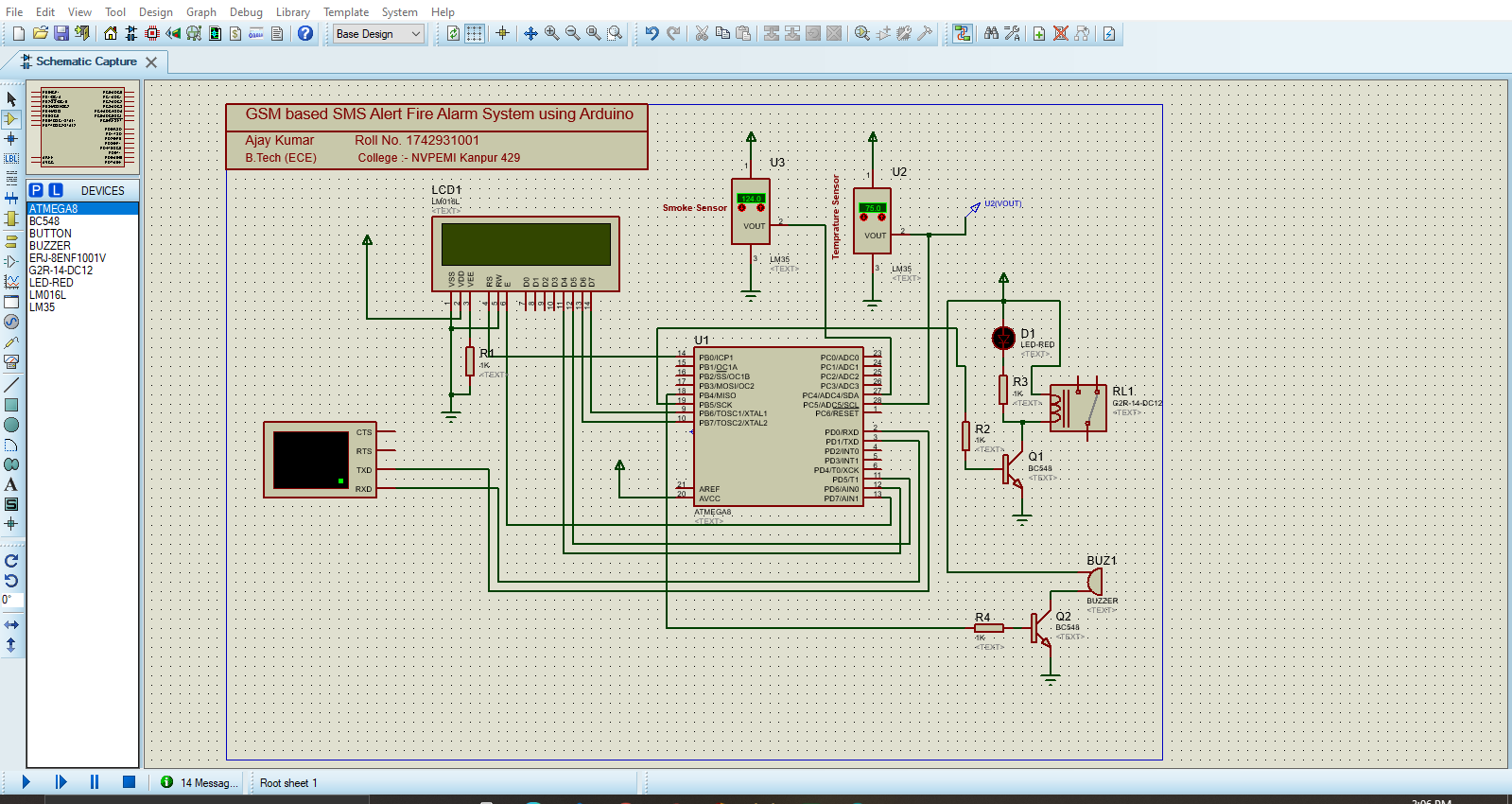
**CIRCUIT DIAGRAM:-**



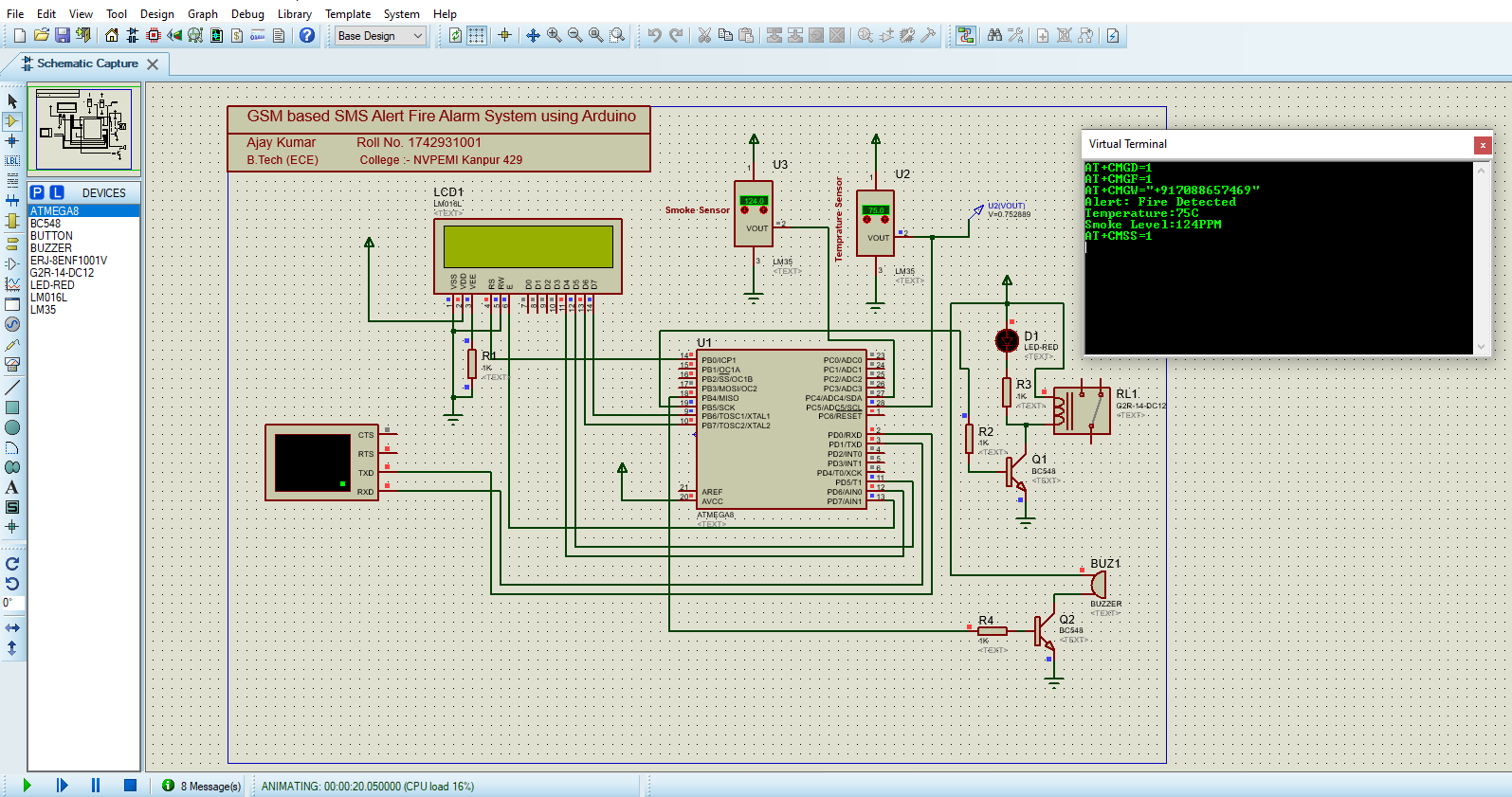
**Theory:-** Proteus 8 Professional is a software which can be used to draw schematics, PCB layout, code and even simulate the schematic . You can simulate your work and be more efficient in completing the task at hand. It is developed by Labcenter Electronic Ltd.

The **Arduino Integrated Development Environment ([IDE](https://en.wikipedia.org/wiki/Integrated_development_environment" \o "Integrated development environment))** is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform" \o "Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows" \o "Windows), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS), [Linux](https://en.wikipedia.org/wiki/Linux" \o "Linux)) that is written in functions from [C](https://en.wikipedia.org/wiki/C_(programming_language)" \o "C (programming language)) and [C++](https://en.wikipedia.org/wiki/C++_(programming_language)" \o "C++ (programming language)). It is used to write and upload programs to [Arduino](https://en.wikipedia.org/wiki/Arduino" \o "Arduino) compatible boards, but also, with the help of third-party cores, other vendor development boards.

Before Simulation



After simulation



Procedure:-

1. Fist of all we are open the proteus8 software .
2. Now we Choose all the components those use in this projects .
3. Now draw the schematics .
4. After the schematics , we open Arduino Ide and write code .
5. After write code generate hex file .
6. this hex file upload in Atmega8 micro controller .
7. Now simulate the schematic .

#### **Program/Code for Fire Alarm System using Arduino**

#include <SoftwareSerial.h>

#include<LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

SoftwareSerial mySerial(9, 10);

int sensor=A1;

float temp\_read,Temp\_alert\_val,Temp\_shut\_val;

int sms\_count=0,Fire\_Set;

void setup()

{

  pinMode(sensor,INPUT);

  mySerial.begin(9600);

  Serial.begin(9600);

  lcd.begin(16,2);

  delay(500);

}

void loop()

{

CheckFire();

CheckShutDown();

}

void CheckFire()

{

lcd.setCursor(0,0);

lcd.print("Fire Scan - ON");

Temp\_alert\_val=CheckTemp();

if(Temp\_alert\_val>45)

{

 SetAlert(); // Function to send SMS Alerts

}

}

float CheckTemp()

{

temp\_read=analogRead(sensor); // reads the sensor output (Vout of LM35)

temp\_read=temp\_read\*5;    // converts the sensor reading to temperature

temp\_read=temp\_read/10;  // adds the decimal point

return temp\_read; // returns temperature value in degree celsius

}

void SetAlert()

{

while(sms\_count<3) //Number of SMS Alerts to be sent

{

SendTextMessage(); // Function to send AT Commands to GSM module

}

Fire\_Set=1;

lcd.setCursor(0,1);

lcd.print("Fire Alert! SMS Sent!");

}

void CheckShutDown()

{

if(Fire\_Set==1)

{

Temp\_shut\_val=CheckTemp();

if(Temp\_shut\_val<28)

{

lcd.setCursor(0,1);

lcd.print("Fire Shut! SAFE NOW");

sms\_count=0;

Fire\_Set=0;

}}}

void SendTextMessage()

{

  mySerial.println("AT+CMGF=1");    //To send SMS in Text Mode

  delay(2000);

  mySerial.println("AT+CMGS=\"+919544xxxxxx\"\r"); // change to the phone number you using

  delay(2000);

  mySerial.println("Fire in NEW ROOM!");//the content of the message

  delay(200);

  mySerial.println((char)26);//the stopping character

  delay(5000);

   mySerial.println("AT+CMGS=\"+919847xxxxxx\"\r"); // change to the phone number you using

  delay(2000);

  mySerial.println("Fire in NEW ROOM!");//the content of the message

  delay(200);

  mySerial.println((char)26);//the message stopping character

  delay(5000);

  sms\_count++;

}