ACMEGRADE

MINI PROJECT ON

ARDUINO BASED FIRE ALARM SYSTEM

MADE BY:

1. **DIPENDU HAZRA(BTECH. IN ELECTRONICS AND COMMUNICATION ENGINEERING)**
2. **KRUSHNA PRAMOD KAPARE(BTECH. IN ELECTRONICS AND TELECOMMUNICATION ENGINEERING)**

UNDER THE GUIDANCE OF:

PROF. MANJUNATH G

COURSE NAME – EMBEDDED SYSTEMS

OFFICIAL TRAINING PARTNER

TABLE OF CONTENTS

ABSTRACT

Fire outbreak is a major concern at homes, offices, industries etc. It is dangerous and requires high security and control to avoid destruction of lives and property. One of the preventive measures to avoid the danger is to install an automatic fire alarm detector at vulnerable locations, hence the Arduino based fire alarm detection and control system is proposed. It is capable of automatically detecting temperature and gas concentration in a given environment, sound an alarm, switch on LEDs for fire indication, display warning message in a LCD screen and also turn on a fan to reduce the intensity of fire. The system uses a LM35 temperature sensor along with a gas sensor for temperature and smoke detection, a buzzer for producing alarming sound, a series of LEDs for visualization purpose, a LCD screen 16X2 for display purpose, and a DC motor that can be attached with a fan to cool the system in case of fire, and Atmeg328p Microcontroller. At the end, the objectives of this project were achieved and the system worked effectively.

INTRODUCTION

Ever since mankind first began building structures out of wood rather than stone, fire has been a part of the learning process. In fact, so common have these infernos been throughout history that nearly every major city in the world has been largely burnt to the ground at one time or another in its history.

Firefighting requires skills in combating, extinguishing, and preventing fires, responding to emergency calls, operating and maintaining fire department equipment and quarters, and extensive training in performing firefighting activities. The earliest known firefighters were in the city of Rome. In 60 A.D., emperor Nero established a Corps of Vigils (Vigils) to protect Rome after a disastrous fire. It consisted of 7,000 people equipped with buckets and axes, and they fought fires and served as police. In the 4th century B.C., an Alexandrian Greek named Ctesibius made a double force pump called a siphon. As water rose in the chamber, it compressed the air inside, which forced the water to eject in a steady stream through a pipe and nozzle.

Nowadays, some factories and buildings have proper installation and fire safety and control arrangements such as fire alarm, fire extinguishers, water supply system etc. But the problem is these conventional fire extinguishing systems are not enough to take prompt action during fire outbreak and hence, save life. The best way to reduce these losses is to respond to the emergency situation as quickly as possible. So, there comes the necessity of a standalone fire detection systems. This project therefore seeks to design a microcontroller fire alarm and control system that will continuously monitor the presence of significant amount of heat and activate an alarm, turn on a series of LEDs for visualization purpose, display an alert message in a LCD screen, and turn on a dc motor that can be attached with a fan to cool the system in case of fire.

PROBLEM STATEMENT

Safety is a crucial consideration in the design of residential and commercial buildings in order to safeguard against loss of life and damage to property.

The existing fire alarm system in market nowadays is too complex in terms of its design and structure. Since the system is too complex, it needs regular maintenance to be carried out to make sure the system operates well. Meanwhile, when the maintenance is being done to the existing system, it could raise the cost of the system.

PROPOSED SYSTEM

The project is designed with a low cost and all level users can have one for a safety purpose. This project therefore seeks to design a microcontroller fire alarm and control system that will continuously monitor the presence of significant amount of heat and activate an alarm, turn on a series of LEDs for visualization purpose, display an alert message in a LCD screen, and turn on a dc motor that can be attached with a fan to cool the system in case of fire.

HARDWARE COMPONENTS REQUIRED

1.Arduino Uno

2.LM35 temperature sensor

3.Gas sensor

4. 3 LEDs – 1 green LED, 1 yellow LED, 1 red LED

5. 5V piezo buzzer

6. 10kΩ potentiometer

7. 5 1kΩ resisters

8. 16×2 LCD

9. Breadboard

10. L293D motor driver module

11. 5V DC motor

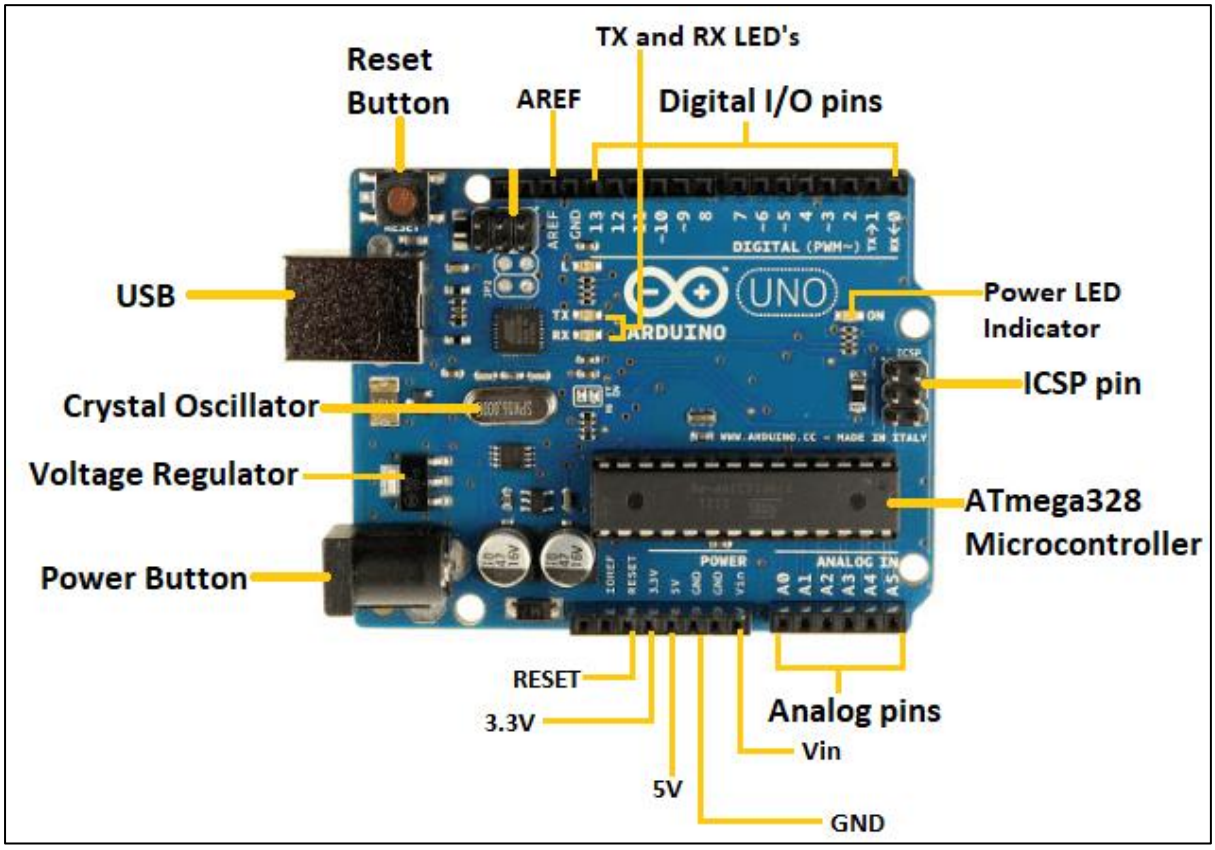
12. Jumper wires

1.Arduino Uno:

The Arduino Uno is an open source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc . The board is equipped with sets of digital and analog input/output(I/O) pins that may be interfaced to various expansion

boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

In this project, 1 arduino uno is required. The arduino code is uploaded in the microcontroller through an USB cable. It serves as the heart of the system because all the components of the system are connected to the arduino via jumper wires.

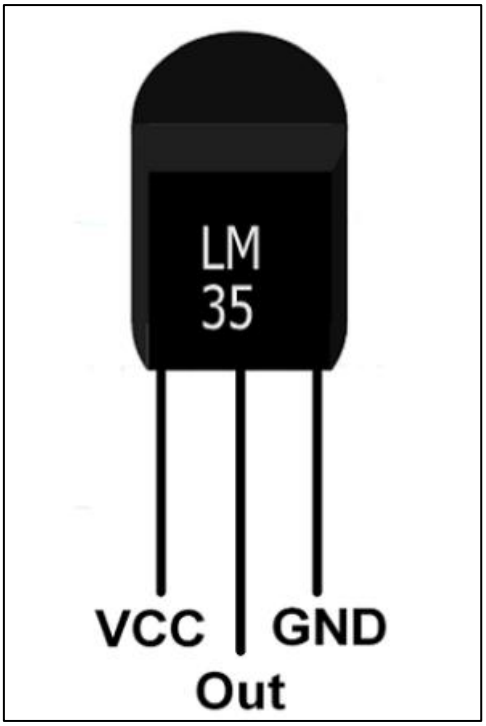


Arduino Uno pin configuration

2.LM35 temperature sensor:

LM35 is a temperature measuring device having an analog output voltage proportional to the temperature. It provides output voltage in Centigrade (Celsius). It does not require any external calibration circuitry. The sensitivity of LM35 is 10 mV/degree Celsius. As temperature increases, output voltage also increases. For, e.g. 250 mV means 25°C. It is a 3-terminal sensor used to measure surrounding temperature ranging from -55 °C to 150 °C. LM35 gives temperature output which is more precise than thermistor output.

In this system, LM35 temperature sensor is used to detect the surrounding temperature in deg. Celsius.



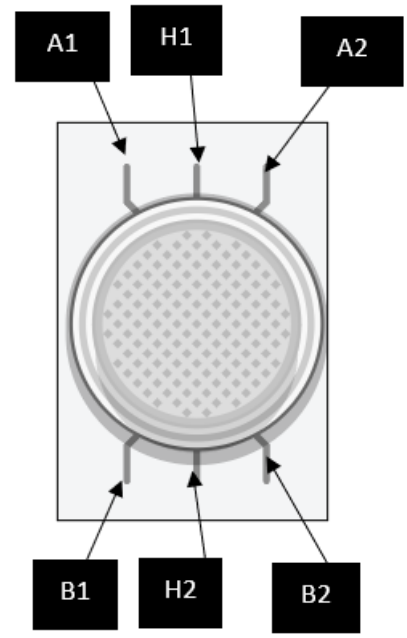
LM35 temperature sensor pin diagram

3. Gas sensor:

Gas sensors (also known as gas detectors) are electronic devices that detect and identify different types of gasses. They are commonly used to detect toxic or explosive gasses and measure gas concentration. Gas sensors are employed in factories and manufacturing facilities to identify gas leaks, and to detect smoke and carbon monoxide in homes. Gas sensors vary widely in size (portable and fixed), range, and sensing ability. They are often part of a larger embedded system, such as hazmat and security systems, and they are normally connected to an audible alarm or interface. Because gas sensors are constantly interacting with air and other gasses, they have to be calibrated more often than many other types of sensors.

Depending on their intended environments and functions, the physical makeup and sensing process can vary notably between sensors. One of the most commonly used gas sensors for toxic identification and smoke detection is the metal oxide based gas sensot. This type of sensor employs a chemi-resistor which comes in contact and reacts with target gasses. Metal oxide gas sensors increase their electrical resistance as they come into contact with gasses such as carbon monoxide, hydrogen, methane, and butane. Most home based smoke detection systems are oxide based sensors.

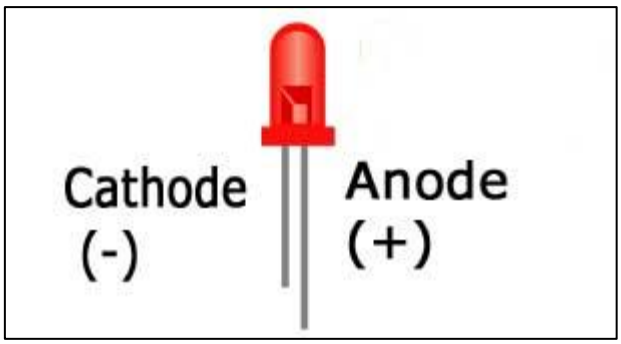
In this system, gas sensor is used to detect the gas concentration around the system.



Pin Configuration of gas sensor

4. LEDs:

The LED is a two-terminal semiconductor light source that emits light when current flows through it. The word LED meaning or LED full form is Light Emitting Diode. The Light Emitting Diode is a special type of p-n junction diode which is made of special type doped semiconductor materials. The LED or Light Emitting Diode allows the flow of current in the forward direction and blocks the current in the reverse direction. When the current flow in the forward direction then LED releases energy in the form of photons**.**



Pin configuration of LED

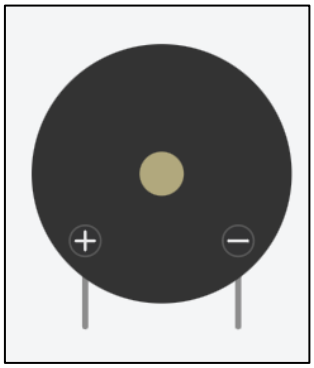
In this project, 3 LEDs(1 green LED, 1 yellow LED, 1 red LED) are used for indication purposes.

5. Piezo buzzer:

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.

A piezoelectric buzzer/beeper also depends on acoustic cavity resonance or Helmholtz resonance to produce an audible beep.

In this project, 1 piezo buzzer is used for producing alarming sound in case of fire.



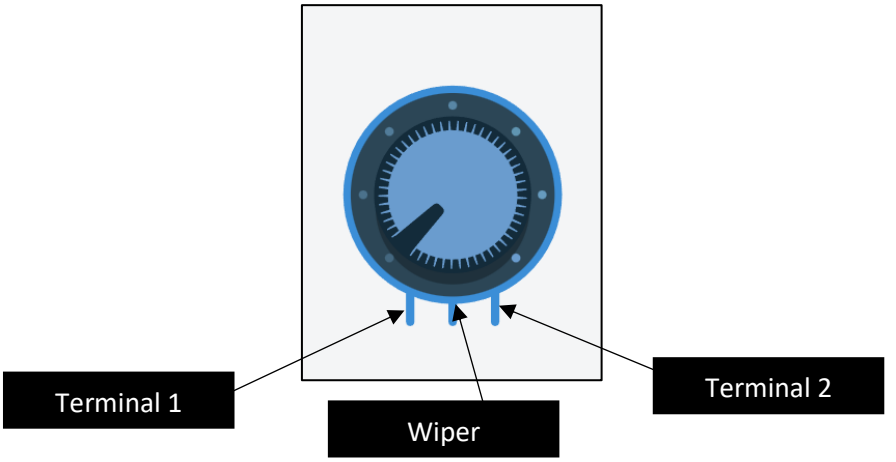
5V piezo buzzer

6. 10kΩ potentiometer:

A 10kΩ potentiometer is an electronic component that can be used to control the flow of electricity through a circuit, much like a faucet regulates the flow of water in your home.

A 10k pot has three terminals at one end - known as the wiper, outside leads, and ground - just like any other potentiometer does but its total resistance on either side is approximately 10K ohms when completely turned clockwise and 0 ohms when completely turned counter-clockwise. Potentiometers with values less than 10k are called regular ("linear") pots whereas those above are called logarithmic because resistive track increases by orders of magnitude as you turn the knob from one end to the other.

In this project, a 10kΩ potentiometer is used to control the contrast of a 16×2 LCD screen.



Pin configuration of 10kΩ potentiometer

7. Resistors:

A passive electrical component with two terminals that are used for either limiting or regulating the flow of electric current in electrical circuits. The main purpose of resistor is to reduce the current flow and to lower the voltage in any particular portion of the circuit.

In this project, resistors are used to limit the flow of current through the electrical components.

1 kΩ resistor

8. 16×2 LCD:

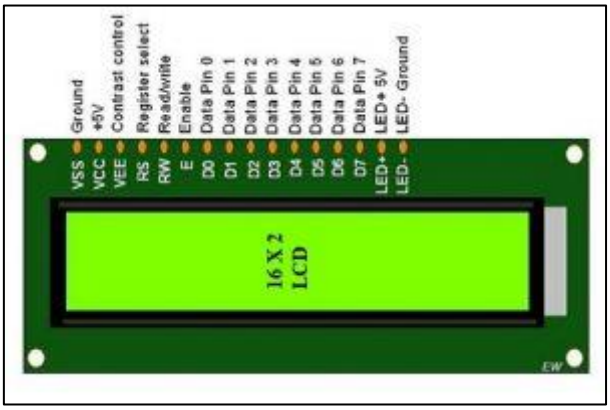
An electronic device that is used to display data and the message is known as LCD 16×2. As the name suggests, it includes 16 Columns & 2 Rows so it can display 32 characters (16×2=32) in total & every character will be made with 5×8 (40) Pixel Dots. So the total pixels within this LCD can be calculated as 32 x 40 otherwise 1280 pixels.

LCD 16×2 Pin Configuration:

The pin configuration of LCD 16×2 is discussed below so that LCD 16×2 connection can be done easily with external devices.

- Pin1 (Ground): This pin connects the ground terminal.
- Pin2 (+5 Volt): This pin provides a +5V supply to the LCD

- Pin3 (VE): This pin selects the contrast of the LCD.
- Pin4 (Register Select): This pin is used to connect a data pin of an MCU & gets either 1 or 0. Here, data mode = 0 and command mode =1.
- Pin5 (Read & Write): This pin is used to read/write data.
- Pin6 (Enable): This enables the pin must be high to perform the Read/Write procedure. This pin is connected to the data pin of the microcontroller to be held high constantly.
- Pin7 (Data Pin): The data pins are from 0-7 which are connected through the microcontroller for data transmission. The LCD module can also work on the 4-bit mode through working on pins 1, 2, 3 & other pins are free.
- Pin8 – Data Pin 1
- Pin9 – Data Pin 2
- Pin10 – Data Pin 3
- Pin11 – Data Pin 4
- Pin12 – Data Pin 5
- Pin13 – Data Pin 6
- Pin14 – Data Pin 7
- Pin15 (LED Positive): This is a +ve terminal of the backlight LED of the display & it is connected to +5V to activate the LED backlight.
- Pin16 (LED Negative): This is a -ve terminal of a backlight LED of the display & it is connected to the GND terminal to activate the LED backlight.



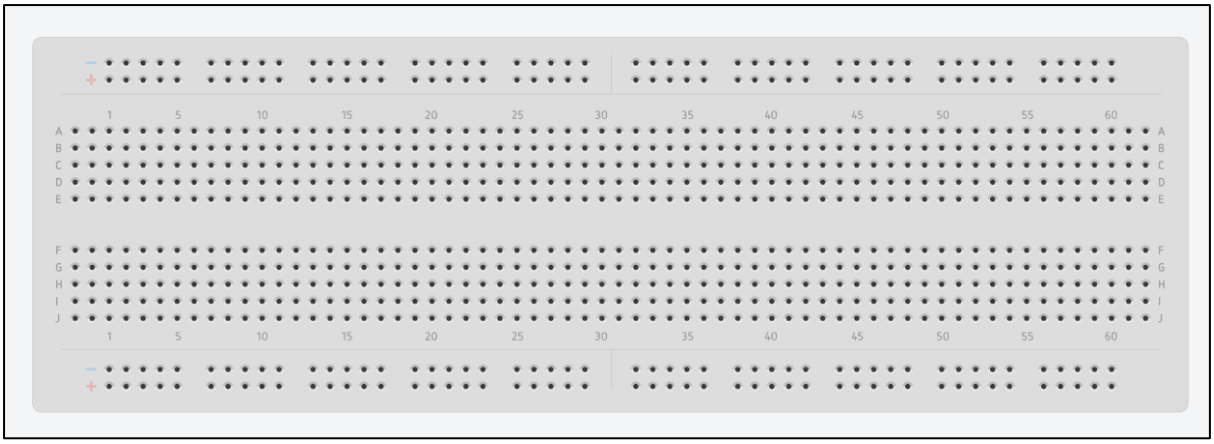
16×2 LCD Pin Diagram

In this project, 16×2 LCD is used to display warning messages.

9. Breadboard:

A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electrical circuits. Unlike a perfboard or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable. For this reason, breadboards are also popular with students and in technological education.

In this project, breadboard is used to construct the system with Arduino Uno.



Breadboard

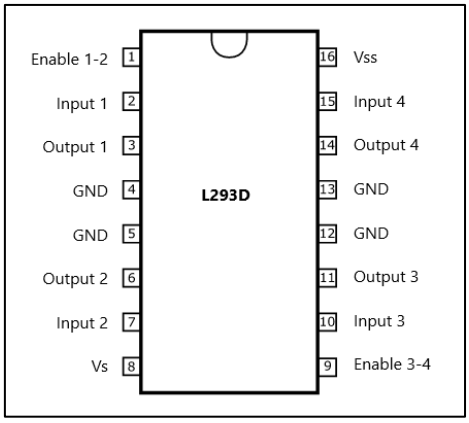
10. L293D motor driver module:

L293D Motor Driver Module is a medium power motor driver perfect for driving DC Motors and Stepper Motors. It uses the popular L293 motor driver IC. It can drive 2 DC motors with directional and speed control.

The driver greatly simplifies and increases the ease with which we may control motors, relays, etc from micro-controllers. It can drive motors up to 12V with a total DC current of up to 600mA.

We can connect the two channels in parallel to double the maximum current or in series to double the maximum input voltage. This motor driver is perfect for robotics and mechatronics projects for controlling motors from microcontrollers, switches, relays, etc.

In this project, it is used to drive a 5V DC motor, which can be attached with a fan to cool the system in case of fire.



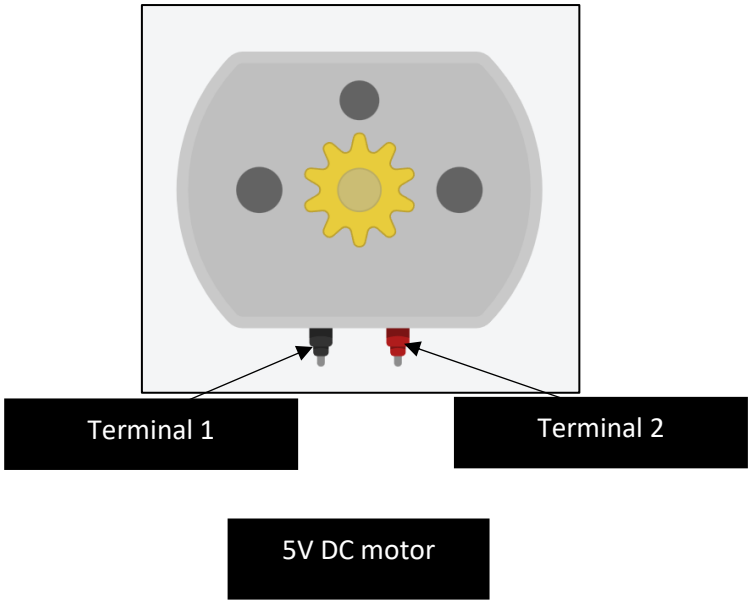
L293D motor driver pin diagram

11. 5V DC motor:

A DC motor is any of a class of rotary electrical motors that converts direct current (DC) electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor.

DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings.

In this project, a 5V DC motor is used and it can be attached with a fan to cool the system in case of fire.



12. Jumper wires:

These are used for making the connections between the components of the system.

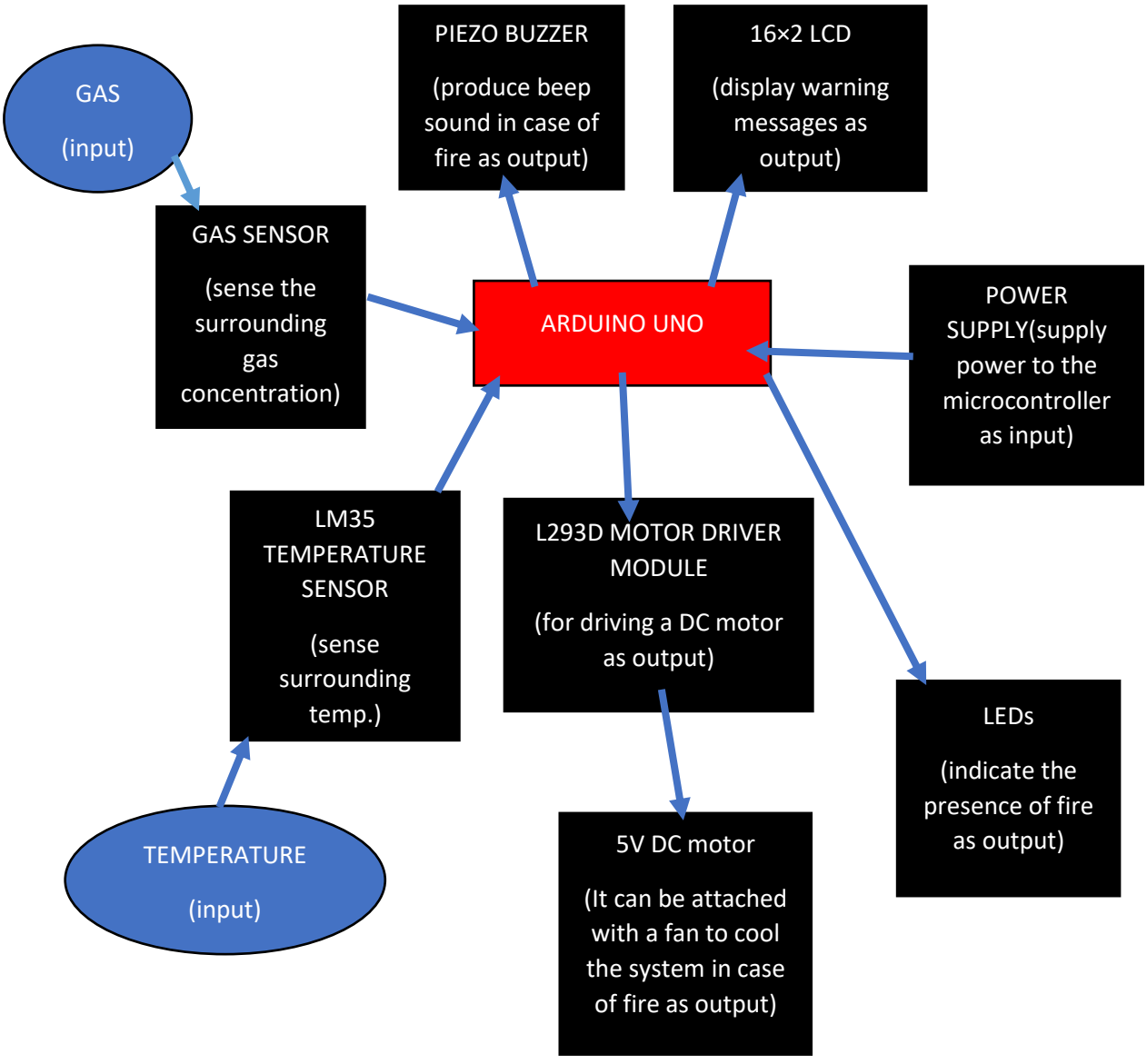
SOFTWARE REQUIRED

In this project, Tinkercad software is used to design the whole system.

The Circuits section of Tinkercad is a simulator for an electronic circuit with a Arduino Uno or a Micro bit board or a ATiny chip in the browser. The code can be made with CodeBlocks, which are graphical code pieces that can be put together by shifting them with the mouse cursor. Programming with code text is also possible.

Tinkercad has included libraries for some components, such as the Adafruit Neopixel library, the Arduino Servo library and a library for a I2C display. It is not possible to select or upload other libraries. The circuit can have analog components which are fully simulated.

BLOCK DIAGRAM



WORKING PRINCIPLE

The fire detection system is designed using Arduino Uno as the microcontroller unit. A gas sensor is used to measure the gas concentration around the system and produce analog values as output to the Arduino Uno. A LM35 temperature sensor is used to measure the surrounding temperature of the system and produce analog values as output to the Arduino Uno. The Arduino Uno controls a piezo buzzer, a series of LEDs, a L293D motor driver module, and a 16×2 LCD.

The piezo buzzer is used to produce beep sound as output when the surrounding temperature goes high and the gas concentration exceeds a certain desired limit. A series of LEDs is used to indicate the situation around the fire detection system. A L293D motor driver module is used to drive a 5V DC motor because the Arduino Uno does not have enough power to drive the 5V DC motor. A 16×2 LCD is used to display warning messages for visualization purposes.

The Arduino Uno is programmed to read the analog values of the LM35 temperature sensor and that of the gas sensor. The analog values of the LM35 temperature sensor are converted to degree Celsius values.

If the gas sensor value is less than or equal to 85, or the temperature is less than 30 degree Celsius, the green LED will glow, and “STAY COOL” message will be displayed in the LCD screen.

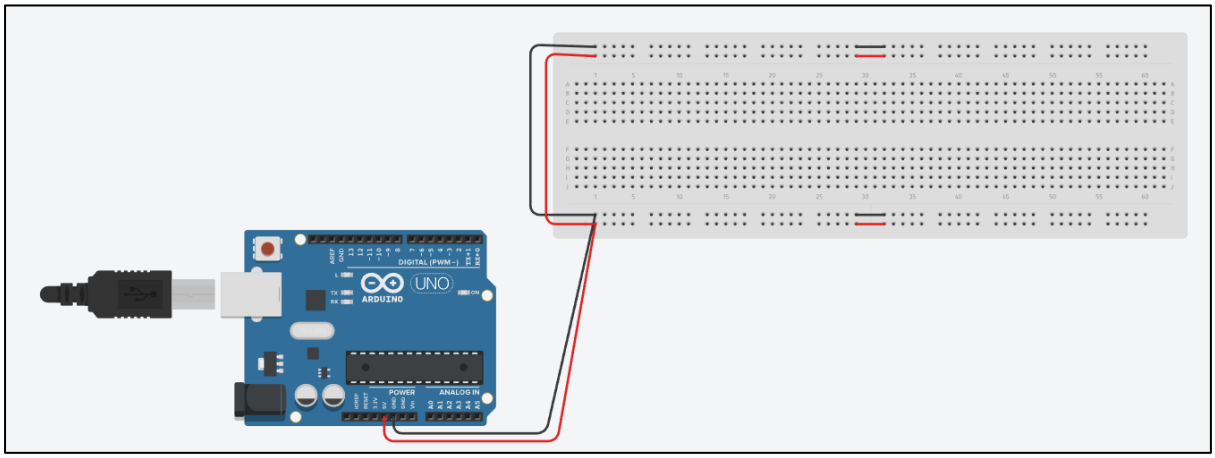
If the gas sensor value is greater than 85 and less than 100, or the temperature is greater or equal to 30 degree Celsius or less than 40 degree Celsius, the green and yellow LED will glow, and “STAY COOL” message will be displayed in the LCD screen.

If the gas sensor value is greater than or equal to 100, or the temperature is greater than or equal to 40 degree Celsius, the green, yellow, and red LED will glow, the buzzer will produce beep sound as alarming sound, the motor that can be attached with a fan, will rotate to cool the system, and “DANGER” message will be displayed in the LCD screen.

IMPLEMENTATION

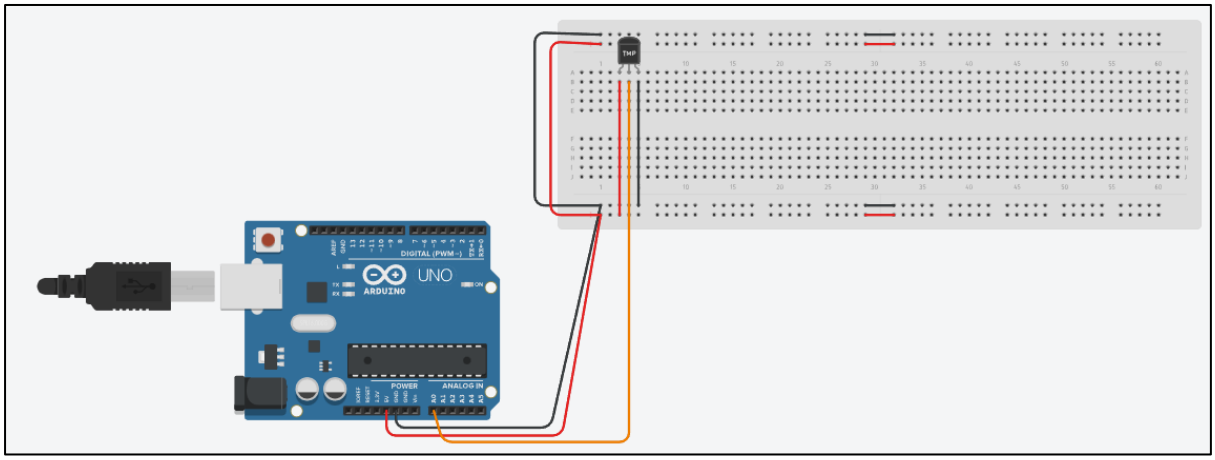
STEP 1: CONNECT ARDUINO UNO WITH BREADBOARD

Connect the 5V pin of Arduino Uno to the positive line of the breadboard and connect the Gnd pin of the Arduino Uno to the negative line of the breadboard.



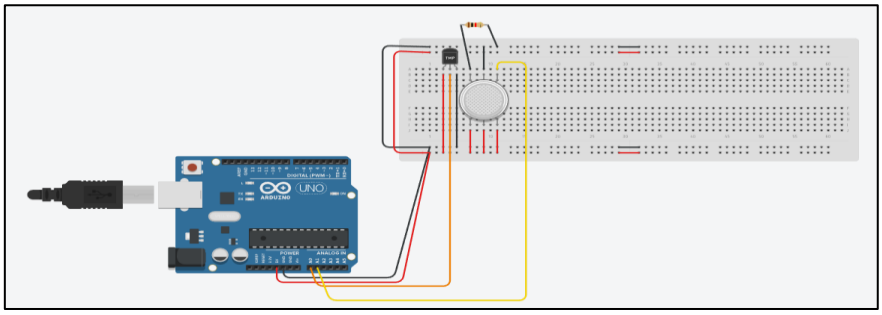
STEP 2: CONNECT LM35 TEMPERATURE SENSOR TO THE ARDUINO UNO

Connect the Vout pin, Power pin and Gnd pin of LM35 temperature sensor to A0 pin of Arduino Uno, positive line of breadboard and negative line of the breadboard, respectively.



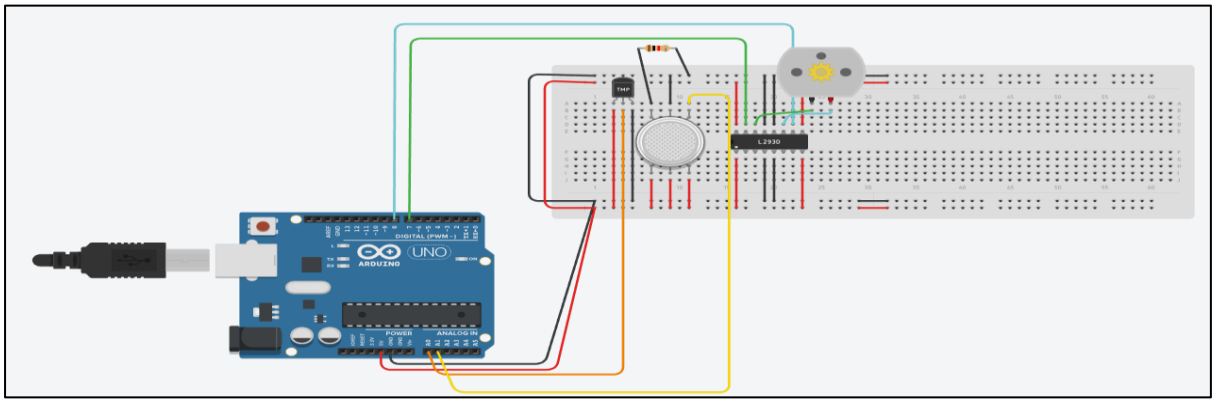
STEP 3: CONNECT GAS SENSOR TO THE ARDUINO UNO

Connect the B1, H2, and B2 pins of the gas sensor to the 5V line in the breadboard. Connect the A1 pin of the gas sensor to the ground line of the breadboard via the 1kΩ resistor. Connect the A2 pin of the gas sensor to the analog pin A1 of the Arduino board. Connect the H1 pin of the gas sensor to ground line of the breadboard.



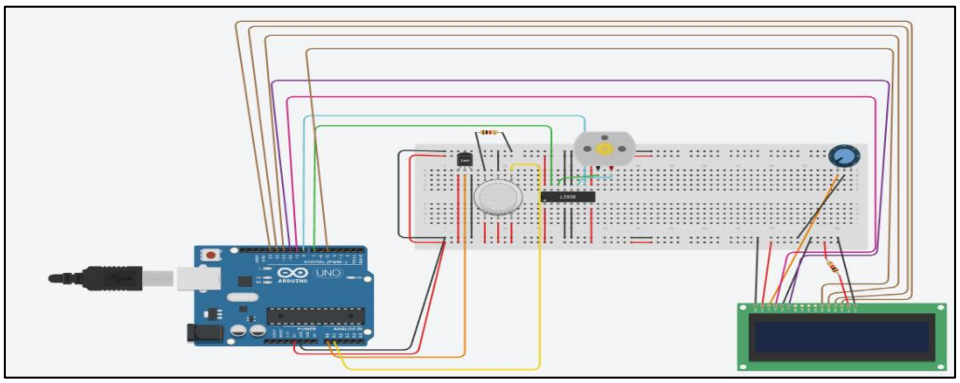
STEP 4: CONNECT L293D MOTOR DRIVER MODULE TO THE ARDUINO UNO AND CONNECT 5V DC MOTOR TO THE MOTOR DRIVER MODULE

Connect Enable 1&2 pin, Enable 3&4, Power1 and Power2 pins of the motor driver to the positive line of the breadboard. Connect the 4 Ground pins of the motor driver to the negative line of the breadboard. Connect Input4 pin of the motor driver to D7 pin of the Arduino Uno. Connect Input3 pin of the motor driver to the D8 pin of the Arduino Uno. Connect Output4 pin to the Terminal1 pin of the DC motor and connect Output3 pin of the motor driver to Terminal2 pin of the DC motor.



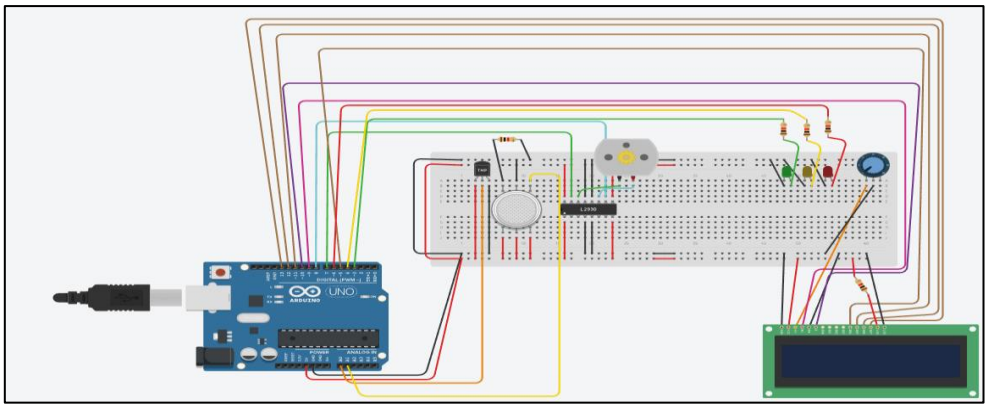
STEP 5: CONNECT LCD TO THE ARDUINO UNO

Connect the Ground pin of LCD to the negative line of the breadboard. Connect the Power pin of the LCD to the positive line of the breadboard. Connect the LCD's Register select pin and Enable pin to the Arduino's D9 and D10 pin respectively. Connect DB pins 4,5,6,7 to digital pins D5, D11, D12,D13 of Arduino board respectively. Connect the contrast pin to the Terminal1 pin of the 10kΩ potentiometer and connect the Wiper pin of the potentiometer to the negative line of the breadboard. Connect the read/write pin to the ground line of the breadboard. Connect the LED anode pin to the positive line of the breadboard via the 1KΩ resister. Connect the LED cathode pin to the ground line of the breadboard.



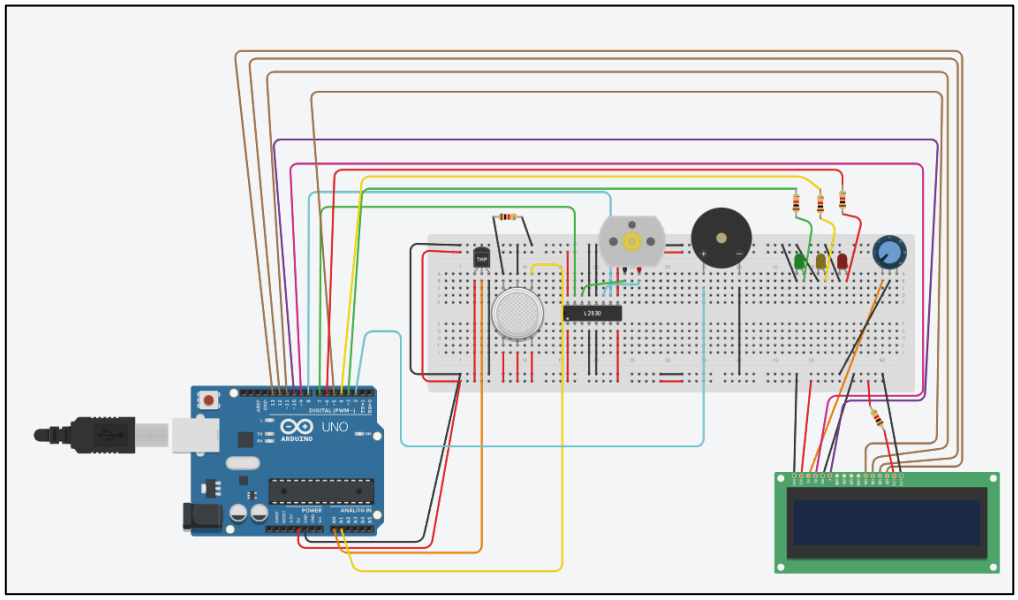
STEP 6: CONNECT THE LEDs TO THE ARDUINO UNO

Connect the cathode pins of the 3 LEDs to the negative line of the breadboard. Connect the anode pin of the green LED to D3 pin of Arduino Uno via 1kΩ resistor. Connect the anode pin of the yellow LED to D4 pin of Arduino Uno via 1kΩ resistor. Connect the anode pin of the red LED to D6 pin of Arduino Uno via 1kΩ resistor.

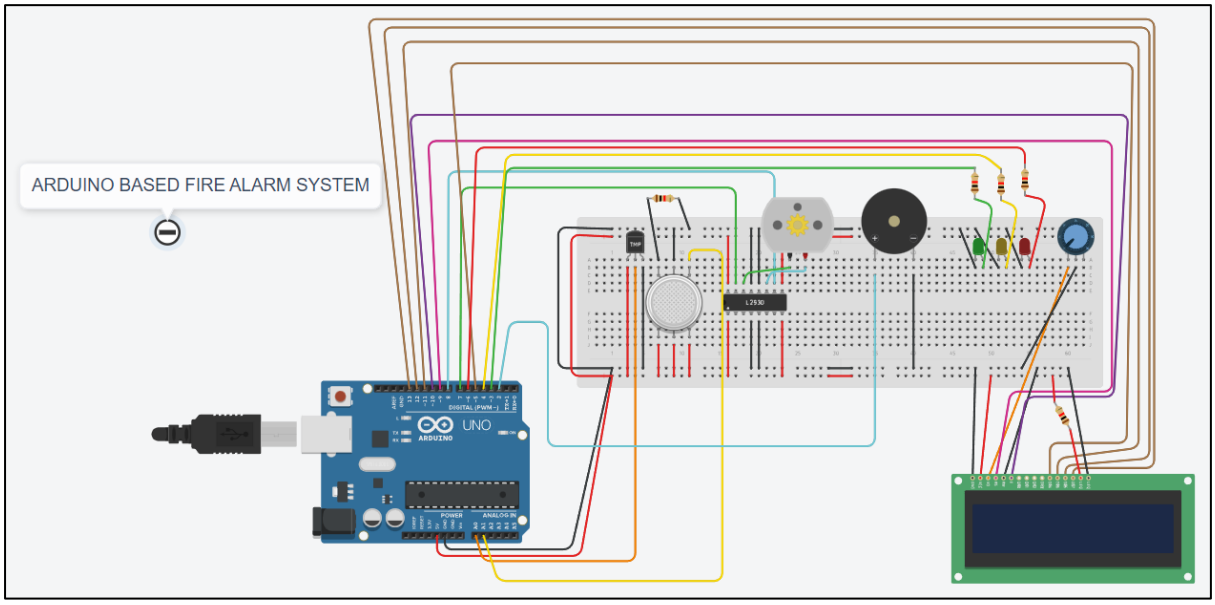


STEP 7: CONNECT THE BUZZER TO THE ARDUINO UNO

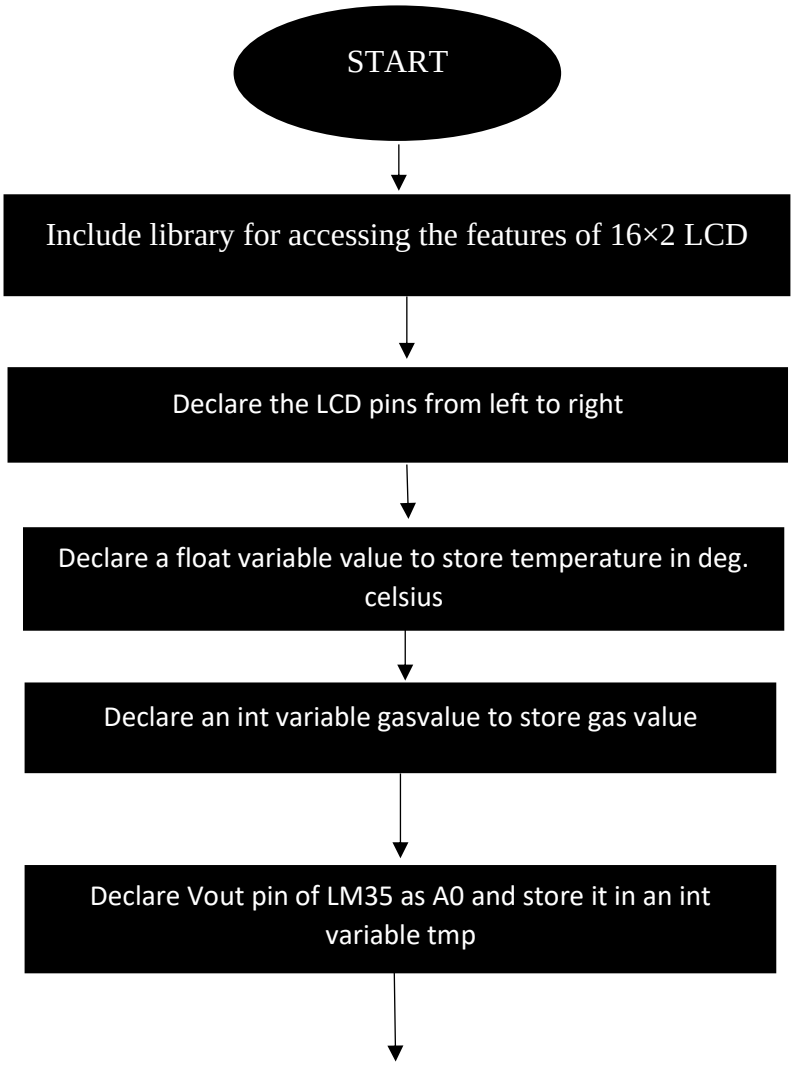
Connect the positive terminal of the buzzer to the D2 pin of the Arduino Uno. Connect the negative terminal of the buzzer to the negative line of the breadboard.

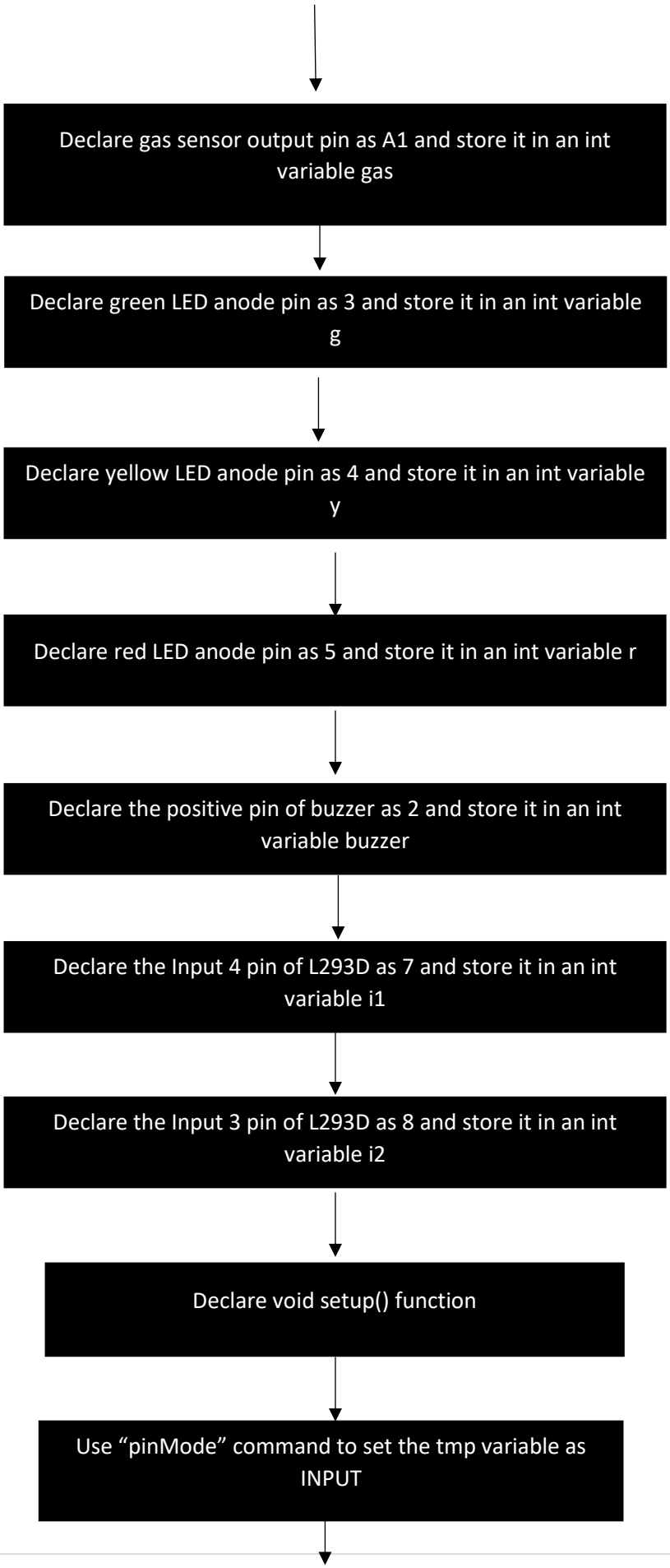


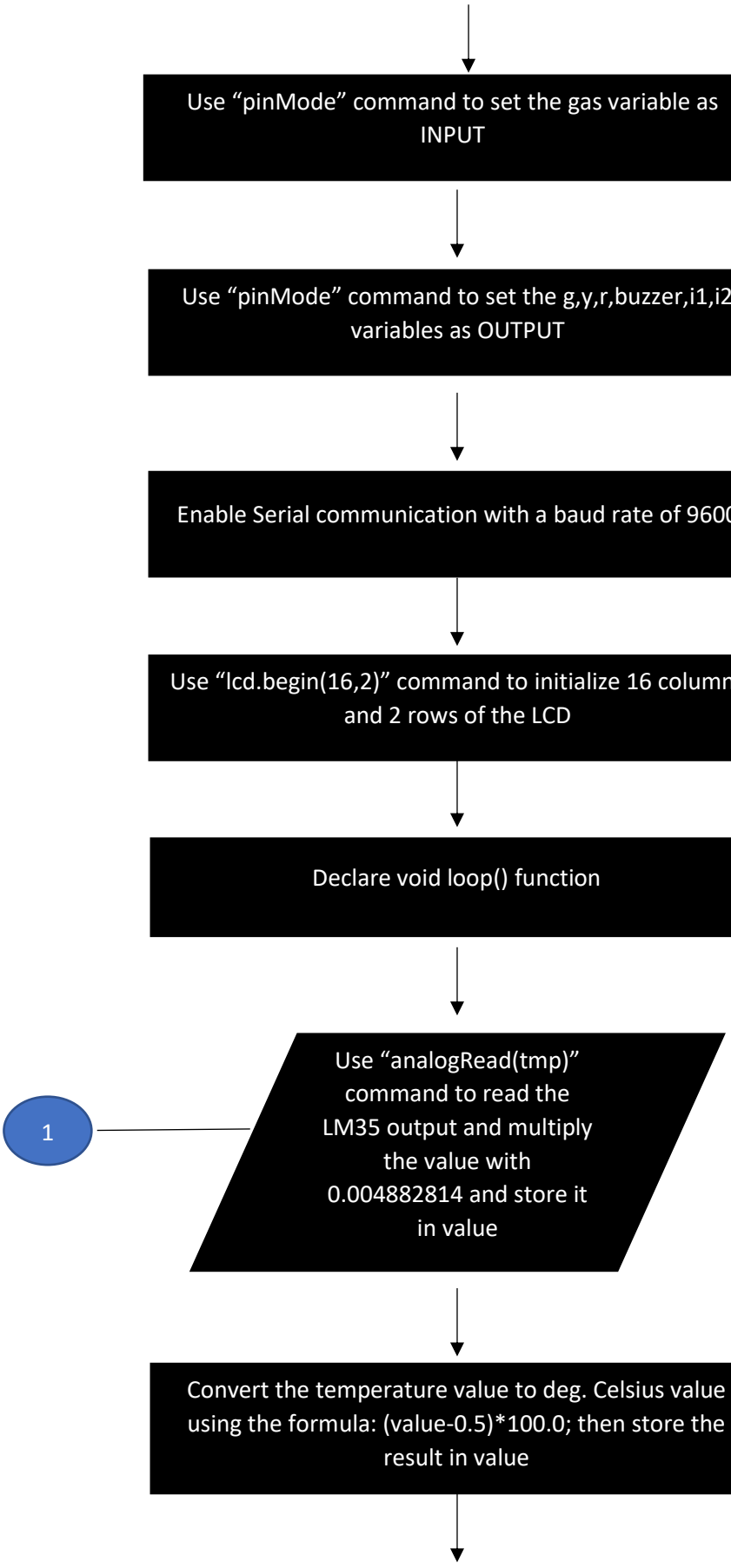
FINAL CIRCUIT:

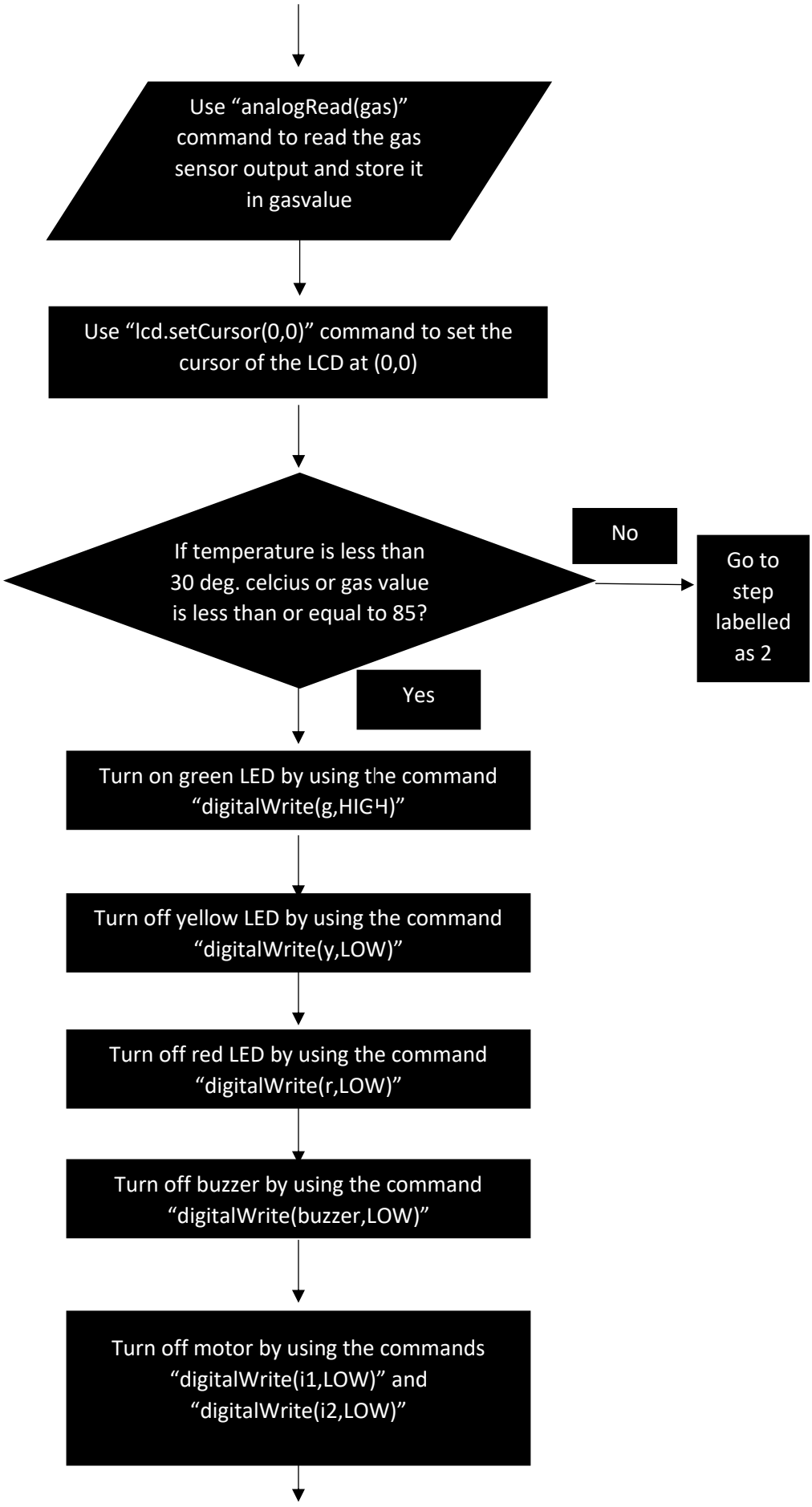


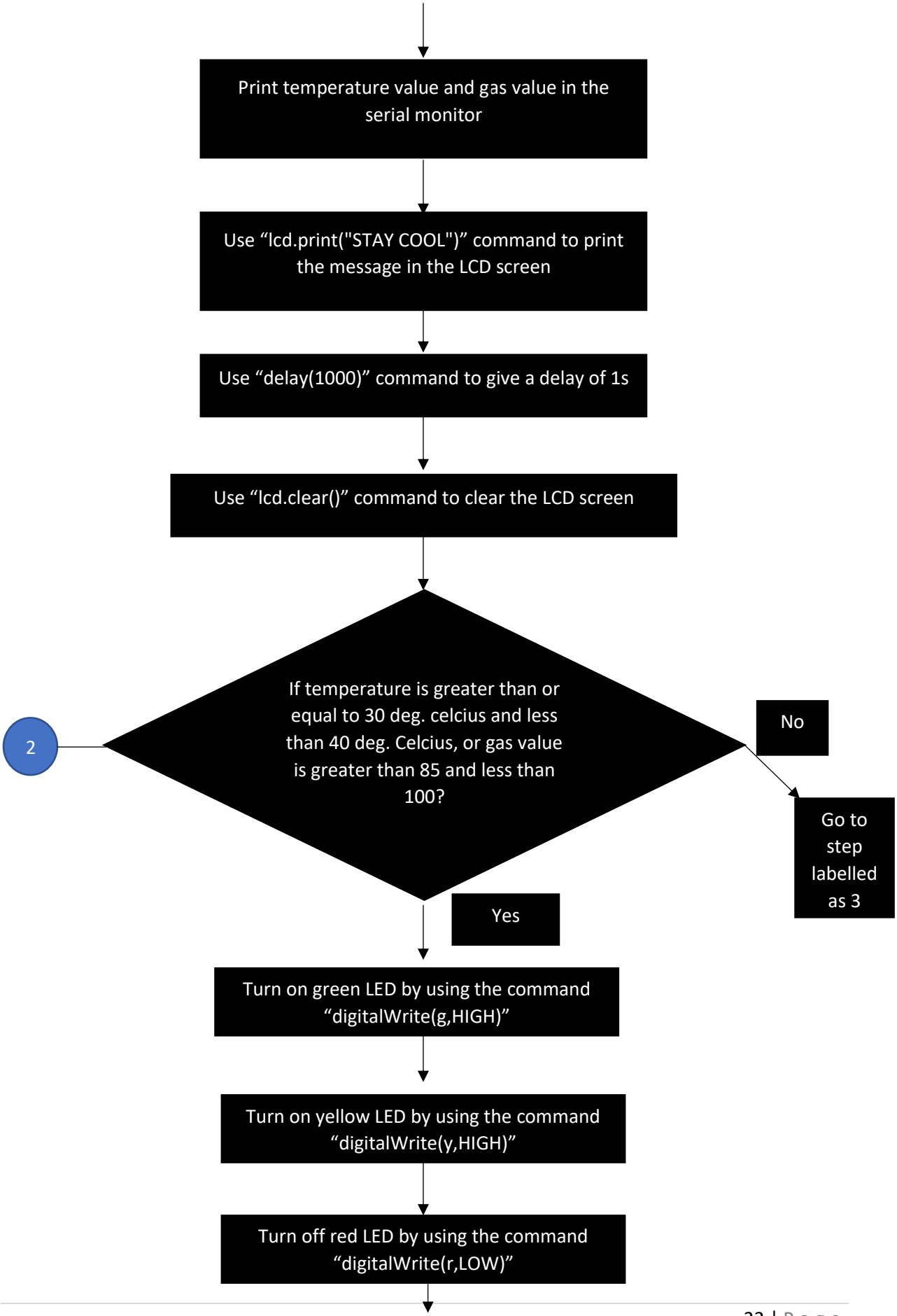
FLOWCHART OF ARDUINO CODE



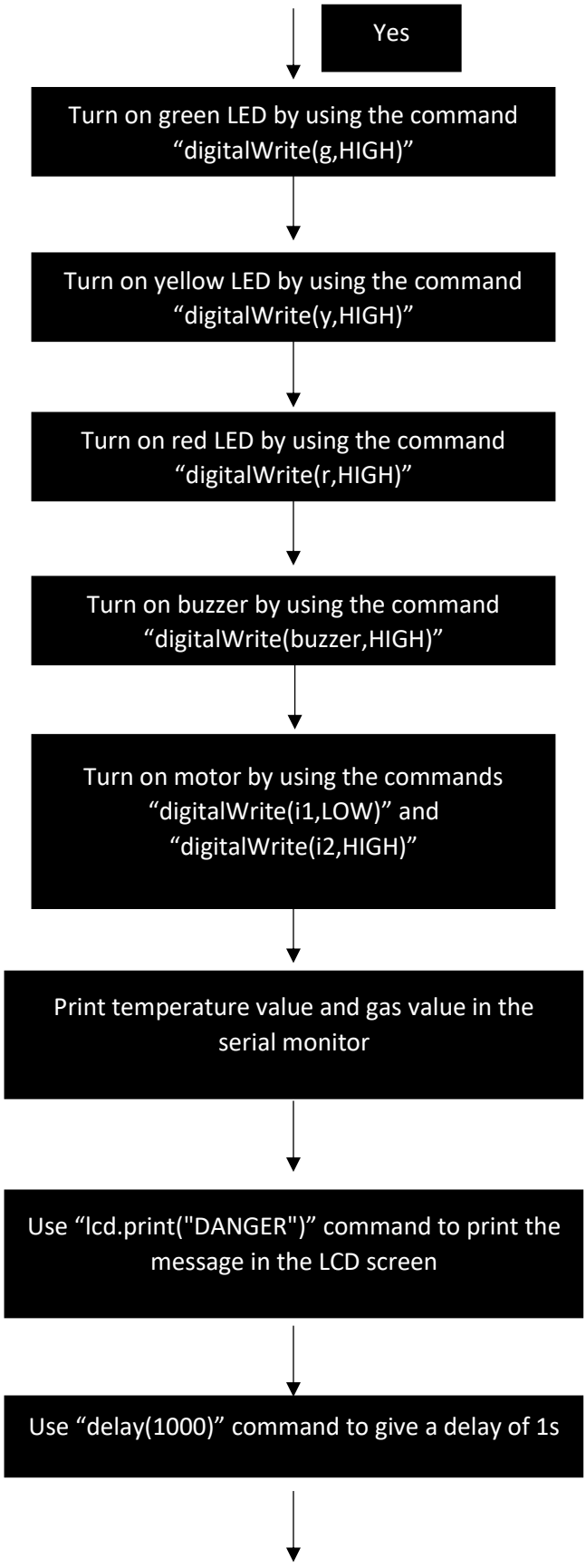

Declare gas sensor output pin as A1 and store it in an int variable gas
Declare green LED anode pin as 3 and store it in an int variable g
Declare yellow LED anode pin as 4 and store it in an int variable y
Declare red LED anode pin as 5 and store it in an int variable r
Declare the positive pin of buzzer as 2 and store it in an int variable buzzer
Declare the Input 4 pin of L293D as 7 and store it in an int variable i1
Declare the Input 3 pin of L293D as 8 and store it in an int variable i2
Declare void setup() function
Use “pinMode” command to set the tmp variable as INPUT

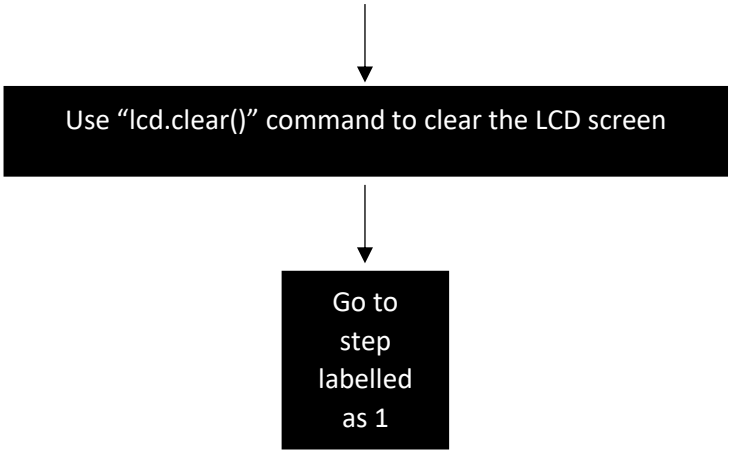

Use “pinMode” command to set the gas variable as INPUT
Use “pinMode” command to set the g,y,r,buzzer,i1,i2 variables as OUTPUT
Enable Serial communication with a baud rate of 9600
Use “lcd.begin(16,2)” command to initialize 16 columns and 2 rows of the LCD
Declare void loop() function
1
Use “analogRead(tmp)” command to read the LM35 output and multiply the value with 0.004882814 and store it in value
Convert the temperature value to deg. Celsius value using the formula: (value-0.5)*100.0; then store the result in value


Use “analogRead(gas)” command to read the gas sensor output and store it in gasvalue
Use “lcd.setCursor(0,0)” command to set the cursor of the LCD at (0,0)
If temperature is less than 30 deg. celcius or gas value is less than or equal to 85?
No
Go to step labelled as 2
Yes
Turn on green LED by using the command “digitalWrite(g,HIGH)”
Turn off yellow LED by using the command “digitalWrite(y,LOW)”
Turn off red LED by using the command “digitalWrite(r,LOW)”
Turn off buzzer by using the command “digitalWrite(buzzer,LOW)”
Turn off motor by using the commands “digitalWrite(i1,LOW)” and “digitalWrite(i2,LOW)”


Print temperature value and gas value in the serial monitor
Use “lcd.print("STAY COOL")” command to print the message in the LCD screen
Use “delay(1000)” command to give a delay of 1s
Use “lcd.clear()” command to clear the LCD screen
If temperature is greater than or equal to 30 deg. celcius and less than 40 deg. Celcius, or gas value is greater than 85 and less than 100?
2
No
Go to step labelled as 3
Yes
Turn on green LED by using the command “digitalWrite(g,HIGH)”
Turn on yellow LED by using the command “digitalWrite(y,HIGH)”
Turn off red LED by using the command “digitalWrite(r,LOW)”


Turn off buzzer by using the command "digitalWrite(buzzer,LOW)"
Turn off motor by using the commands "digitalWrite(i1,LOW)" and "digitalWrite(i2,LOW)"
Print temperature value and gas value in the serial monitor
Use "lcd.print("STAY COOL")" command to print the message in the LCD screen
Use "delay(1000)" command to give a delay of 1s
Use "lcd.clear()" command to clear the LCD screen
3
If temperature is greater than or equal to 40 deg. Celsius, or gas value is greater than or equal to 100
No
Go to step labelled as 1


Yes
Turn on green LED by using the command “digitalWrite(g,HIGH)”
Turn on yellow LED by using the command “digitalWrite(y,HIGH)”
Turn on red LED by using the command “digitalWrite(r,HIGH)”
Turn on buzzer by using the command “digitalWrite(buzzer,HIGH)”
Turn on motor by using the commands “digitalWrite(i1,LOW)” and “digitalWrite(i2,HIGH)”
Print temperature value and gas value in the serial monitor
Use “lcd.print("DANGER")” command to print the message in the LCD screen
Use “delay(1000)” command to give a delay of 1s



ARDUINO CODE

```
// ARDUINO BASED FIRE ALARM SYSTEM
#include<LiquidCrystal.h> //library for accessing the features of 16*2 LCD
LiquidCrystal lcd(9,10,5,11,12,13);//declare the LCD pins from left to right
float value;// store temperature in deg celcius
int gasvalue=0;//store gas value
int tmp=A0;//declare LM35 Vout pin
int gas=A1;//declare gas sensor output pin
int g=3;//declare green LED anode pin
int y=4;//declare yellow LED anode pin
int r=6;//declare red LED anode pin
int buzzer=2;//declare the positive pin of buzzer
int i1=7;//declare the Input 4 pin of L293D
int i2=8;//declare the Input 3 pin of L293D
void setup()
{
  pinMode(tmp,INPUT);//set the Vout pin of LM35 as INPUT
  pinMode(gas,INPUT);//set the outut pin of the gas sensor as INPUT
  pinMode(g,OUTPUT);//set the anode pin of green LED as OUTPUT
  pinMode(y,OUTPUT);//set the anode pin of yellow LED as OUTPUT
  pinMode(r,OUTPUT);//set the anode pin of red LED as OUTPUT
```

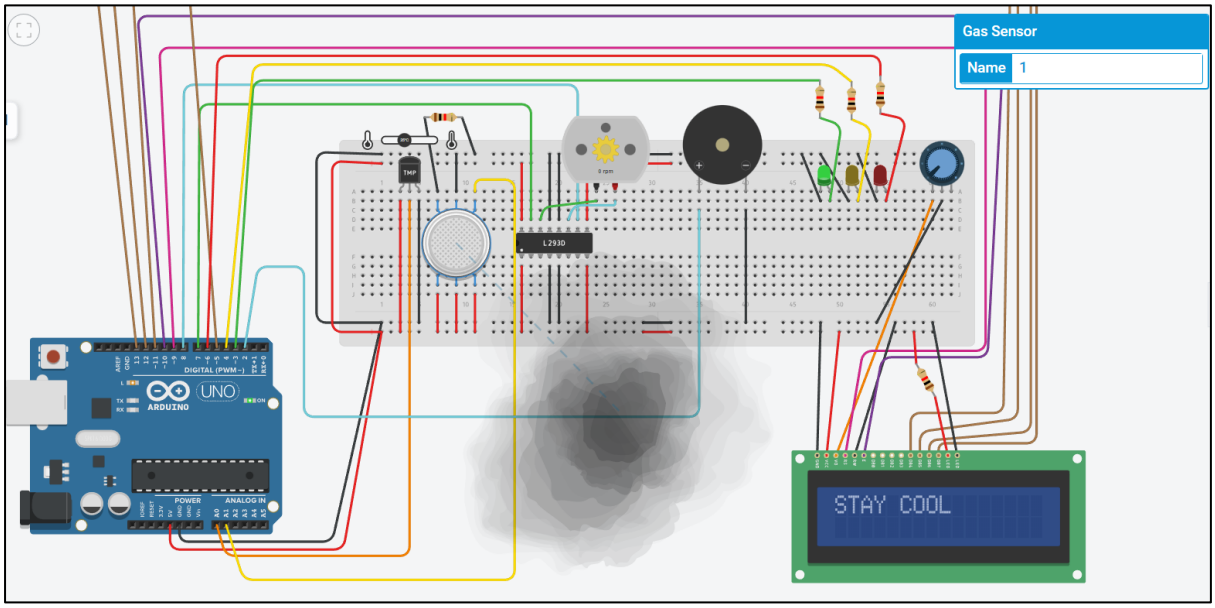
```
  pinMode(buzzer,OUTPUT);//set the positive pin of buzzer as OUTOUT
  pinMode(i1,OUTPUT);//set the input 4 pin of L293D module as OUTPUT
  pinMode(i2,OUTPUT);//set the input 3 pin of L293D module as OUTPUT
  Serial.begin(9600);//enable serial comm. with 9600 baud rate
  lcd.begin(16,2);//initialize 16 columns and 2 rows of the LCD
}

void loop()
{
  value=analogRead(tmp)*0.004882814;//read the LM35 output
  value=(value-0.5)*100.0;//calculate the temperature value in deg. celcius
  gasvalue=analogRead(gas);//read the output of the gas sensor
  lcd.setCursor(0,0);//set the cursor of the LCD at (0,0)
  if(value<30.0 || gasvalue<=85)
  /* If temperature is less than 30 deg. celcius or
  gas value is less than or equal to 85, green LED will be turned on, red LED and
  yellow LED will be turned off, buzzer will be turned off, the motor will be turned off,
  the temperature and gas values wil be printed in the serial monitor,
  "STAY COOL" message will be displayed in the LCD and a delay of 1s will be provided.
  Also the message on LCD will be cleared.
  */
  {
    digitalWrite(g,HIGH);
    digitalWrite(y,LOW);
    digitalWrite(r,LOW);
    digitalWrite(buzzer,LOW);
    digitalWrite(i1,LOW);
    digitalWrite(i2,LOW);
    Serial.print("Temperature in deg. celcius = ");
```

```
  Serial.print(value);
  Serial.print("\tGas value = ");
  Serial.print(gasvalue);
  Serial.println();
  lcd.print("STAY COOL");
  delay(1000);
  lcd.clear();
}
if(value>=30.0 && value<40.0 || gasvalue>85 && gasvalue<100)
/* If temperature is greater than or equal to 30 deg. celcius and less than 40 deg. celcius or
gas value is greater than 85 and less than 100, green and yellow LEDs will be turned on,
red LED will be turned off, buzzer will be turned off, the motor will be turned off,
the temperature and gas values wil be printed in the serial monitor,
"STAY COOL" message will be displayed in the LCD and a delay of 1s will be provided.
Also the message on LCD will be cleared.
*/
{
  digitalWrite(g,HIGH);
  digitalWrite(y,HIGH);
  digitalWrite(r,LOW);
  digitalWrite(buzzer,LOW);
  digitalWrite(i1,LOW);
  digitalWrite(i2,LOW);
  Serial.print("Temperature in deg. celcius = ");
  Serial.print(value);
  Serial.print("\tGas value = ");
  Serial.print(gasvalue);
  Serial.println();
  lcd.print("STAY COOL");
```

```
  delay(1000);
  lcd.clear();
 }
 if(value>=40 || gasvalue>=100)
 /* If temperature is greater than or equal to 40 deg. celcius or
 gas value is greater than or equal to 100, all LEDs will be turned on,
 buzzer will be turned on, the motor will be turned on,
 the temperature and gas values wil be printed in the serial monitor,
 "DANGER" message will be displayed in the LCD and a delay of 1s will be provided.
 Also the message on LCD will be cleared.
 */
 {
  digitalWrite(g,HIGH);
  digitalWrite(y,HIGH);
  digitalWrite(r,HIGH);
  digitalWrite(buzzer,HIGH);
  digitalWrite(i1,LOW);
  digitalWrite(i2,HIGH);
  Serial.print("Temperature in deg. celcius = ");
  Serial.print(value);
  Serial.print("\tGas value = ");
  Serial.print(gasvalue);
  Serial.println();
  lcd.print("DANGER");
  delay(1000);
  lcd.clear();
 }
}
```

SIMULATION RESULTS

CASE 1 : Here, surrounding temperature is less than 30 degree Celsius and the gas sensor value is less than or equal to 85, so, green LED turns on, “STAY COOL” message is displayed in the LCD screen, red and yellow LEDs are off, motor is off, and buzzer is off. This gives an indication that there is no fire around the system and the system’s surrounding is normal.



EXPERIMENTAL SETUP

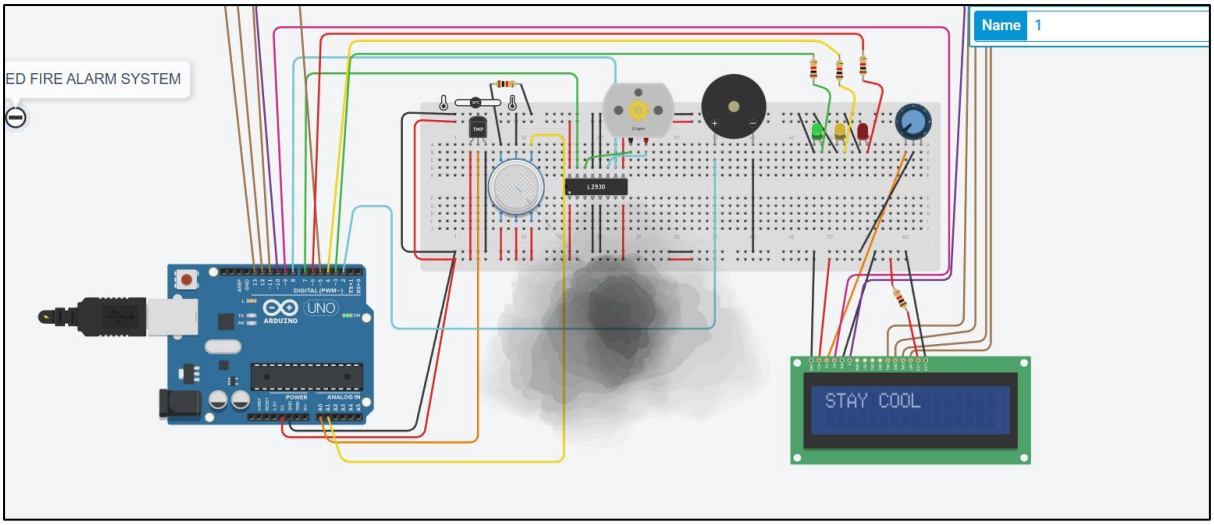
```
Temperature in deg. celcius = 24.71     Gas value = 85
Temperature in deg. celcius = 24.71     Gas value = 85
Temperature in deg. celcius = 24.71     Gas value = 85
Temperature in deg. celcius = 24.71     Gas value = 85
Temperature in deg. celcius = 24.71     Gas value = 85
Temperature in deg. celcius = 24.71     Gas value = 85
```

SERIAL MONITOR OUTPUT

CASE 2 : Here, surrounding temperature is greater than or equal to 30 deg. Celsius and less than 40 deg. Celsius, and the gas value is greater than 85 and less than 100, so, green and yellow LED turns on, “STAY COOL” message is displayed in the LCD screen, red LED is off, motor is off, and buzzer is off. This gives an indication that there is no fire around the system and the system’s surrounding is moderate.

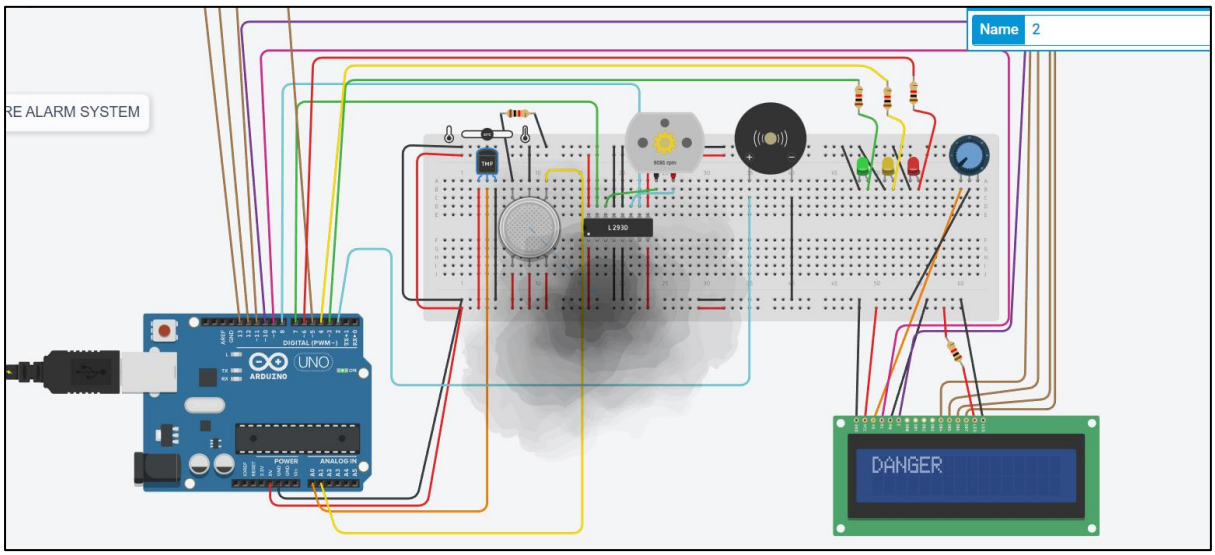
```
Temperature in deg. celcius = 33.01     Gas value = 94
Temperature in deg. celcius = 33.01     Gas value = 94
Temperature in deg. celcius = 33.01     Gas value = 94
Temperature in deg. celcius = 33.01     Gas value = 94
Temperature in deg. celcius = 33.01     Gas value = 94
Temperature in deg. celcius = 33.01     Gas value = 94
Temperature in deg. celcius = 33.01     Gas value = 94
```

SERIAL MONITOR OUTPUT

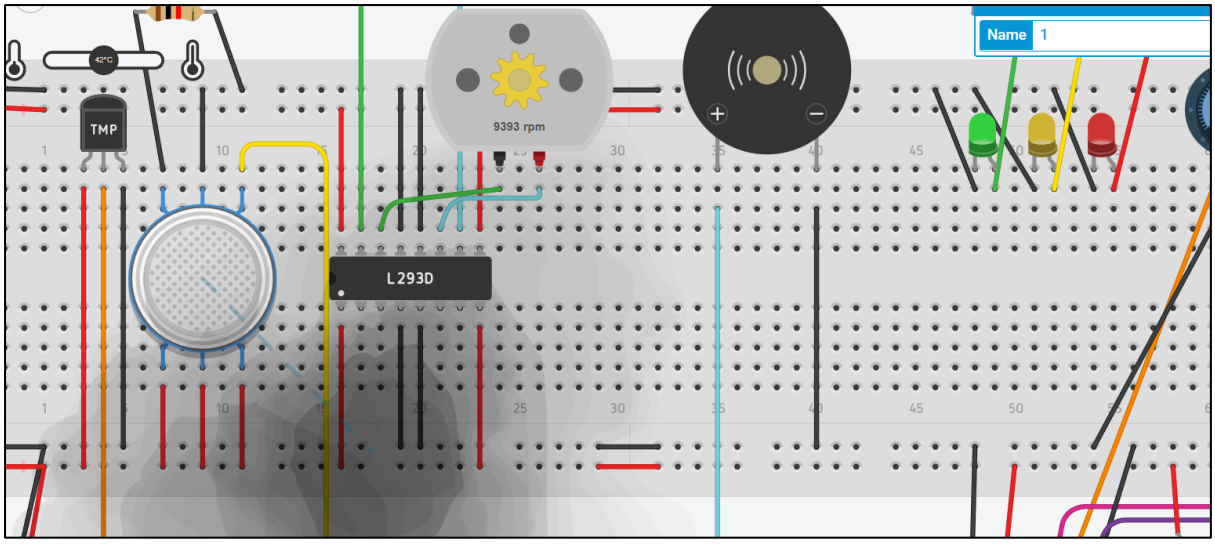


EXPERIMENTAL SETUP

CASE 3 : Here, surrounding temperature is greater than or equal to 40 deg. Celsius, and gas value is greater than or equal to 100, so, all the LEDs turn on, “DANGER” message is displayed in the LCD screen, motor is rotating with a speed of approximately 9393 rpm , and buzzer is producing beep sound as output. This gives an indication that there is fire around the system and the system’s surrounding is in danger. As a fire control mechanism, the DC motor can be attached with a fan to cool the system, thereby make the system’s surrounding normal.



EXPERIMENTAL SETUP



MAGNIFIED VIEW OF EXPERIMENTAL SETUP

```
Temperature in deg. celcius = 42.77     Gas value = 142
Temperature in deg. celcius = 42.77     Gas value = 142
Temperature in deg. celcius = 42.77     Gas value = 142
Temperature in deg. celcius = 42.77     Gas value = 142
Temperature in deg. celcius = 42.77     Gas value = 142
Temperature in deg. celcius = 42.77     Gas value = 142
Temperature in deg. celcius = 42.77     Gas value = 142
```

SERIAL MONITOR OUTPUT

ADVANTAGES

1. It detects fire at an early stage. This, in turn, helps in early reaction, saving lives and property.
2. It protects us and our residence.
3. It is cost effective.
4. It needs negligible maintenance.

DISADVANTAGES

1. This fire alarm system is not wireless. So, in case of fire, warning messages will not be conveyed to distant locations.
2. Faulty connections may exist, which may cause the system to not operate reliably.

APPLICATIONS

The Arduino based fire alarm system can be used in the many areas, out of which some are as follows:

1. Hydrogen stations
2. Combustion monitors for burners
3. Oil and gas pipelines
4. Automotive manufacturing facilities
5. Nuclear facilities
6. Aircraft hangars
7. Turbine enclosures

CONCLUSION

In this work, an attempt has been done to design a fire alarm system using LM35 Temperature sensor, gas sensor and Micro controller in Tinkercad for efficient use of electricity. It will help to reduce the wastage of electricity, save lives, reduce percentage of accident and reduce waste of electric appliance.

The results obtained from the measurement have shown that the system perform well under all the conditions. It is capable of automatically detecting temperature and gas concentration in a given environment, sound an alarm, switch on LEDs for fire indication, displays warning message in a LCD screen and also turns on a fan to reduce the intensity of fire. These objectives were met since the systems works effectively. This prototype can be used to design a real time Arduino based fire alarm system.

REFERENCES

1. https://circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing
2. https://www.fierceelectronics.com/sensors/what-a-gas-sensor
3. https://www.watelectronics.com/lcd-16x2/
4. https://www.anistar.com/employee-resources/what-are-the-advantages-and-disadvantages-of-a-wireless-fire-alarm-system/
5. https://www.ijeast.com/papers/447-453,Tesma411,IJEAST.pdf