VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering (SCOPE)

PROJECT REPORT

COURSE CODE / TITLE	BECE204L– Microprocessors and Microcontrollers		
PROGRAM / YEAR/ SEM	B.Tech (BCE/BPS/BAI/BRS)/II Year/ FALL 2023-2024		
DATE OF SUBMISSION			
TEAM MEMBERS DETAILS	**REGISTER NO.**	**NAME**	
	22BCE1089	Kishore S	
	22BCE1149	M. Rohit	
	22BCE1139	Krishnan Narayanan	
	22BCE5168	Bala Adhithiya	
PROJECT TITLE	**Atmospheric gas and Air Quality Monitoring System**		
COURSE HANDLING FACULTY	**Dr. S. REVATHI Associate Professor, SENSE**	**REMARKS**	
FACULTY SIGNATURE			

OBJECTIVE:

The Primary objective of this project is to make a personal environment monitoring assistant which will be capable of finding whether the surrounding atmosphere is either clean or polluted. The device is designed to help humans check if they could stay in a place thereby ensuring it is environmentally safe from harmful gasses. The device we build, primarily measures the air quality; that is the concentration of harmful gasses like Ammonia (NH_3), sulphur (S), Benzene (C_6H_6), CO_2, and other harmful gases and smoke; and gives you suggestions of whether you could stay in that place. The device also measures the oxygen concentration of a particular place and displays the levels. Even here, it makes suggestions of whether it sufficient or not.

INTRODUCTION:

Introducing the "Personal Environment Monitoring Assistant": In an era marked by increasing concerns about air quality and environmental safety, our project endeavors to create a cutting-edge device designed to empower individuals with real-time information about their immediate surroundings. This innovative assistant employs advanced sensors to measure concentrations of critical pollutants such as ammonia, sulphur, benzene, carbon dioxide, and more. What sets our project apart is not just the provision of raw data but the integration of an intelligent assistant capable of offering personalized suggestions on whether the environment is safe for occupancy. From assessing air quality to gauging oxygen levels, our device aims to be a comprehensive tool, providing actionable insights to help users make informed decisions about their living spaces. Embrace a new level of control over your environment with our Personal Environment Monitoring Assistant.

The project is a personal environment monitoring assistant which keeps track of the biological environment surrounding the user. Various air quality monitoring equipment are available on the market in applications of personal environment monitoring.

Existing Works in the technical aspect of our project:

These devices frequently use sensors to measure pollutant concentrations such as ammonia, sulphur, benzene, carbon dioxide (CO_2), and other dangerous gases. Some may additionally have oxygen sensors to monitor the amount of oxygen in the air. These devices often provide real-time air quality data and may deliver alarms or notifications if specific criteria are surpassed.

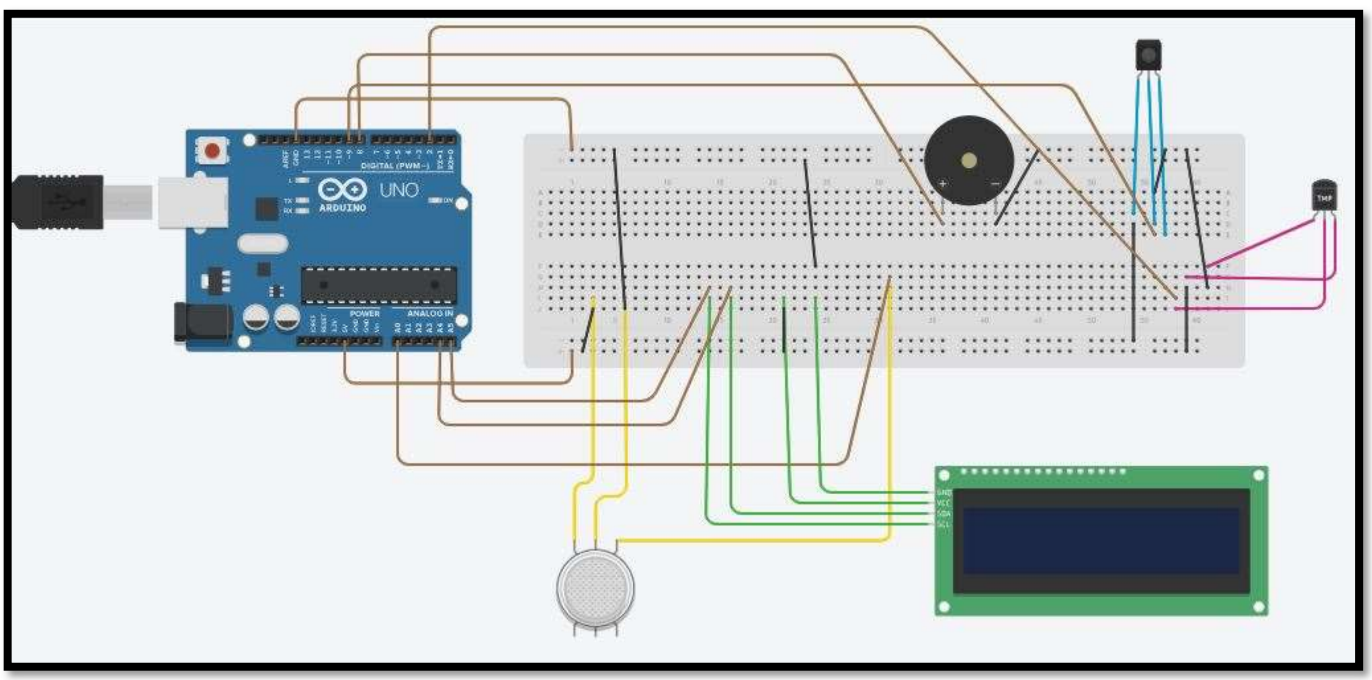
Our Proposed Idea:

The inspiration for creating a personal environment monitoring assistant, as stated in our project, arises from the necessity for individuals to have a tool that allows them to swiftly and easily analyse the safety of their immediate surroundings. What distinguishes our approach from existing solutions is likely the incorporation of an intelligent assistant that not only provides statistics on air quality but also makes recommendations based on the measured factors. The personalized recommendations on whether it is safe or not to stay in a specific place give an extra degree of usability and user-friendliness to the gadget. This could improve the overall user experience and make the gadget more useful for everyday use. Furthermore, the precise combination of gases, which includes ammonia, sulphur, benzene, and carbon dioxide, indicates a comprehensive approach to air quality monitoring that addresses a wide spectrum of potential pollutants.

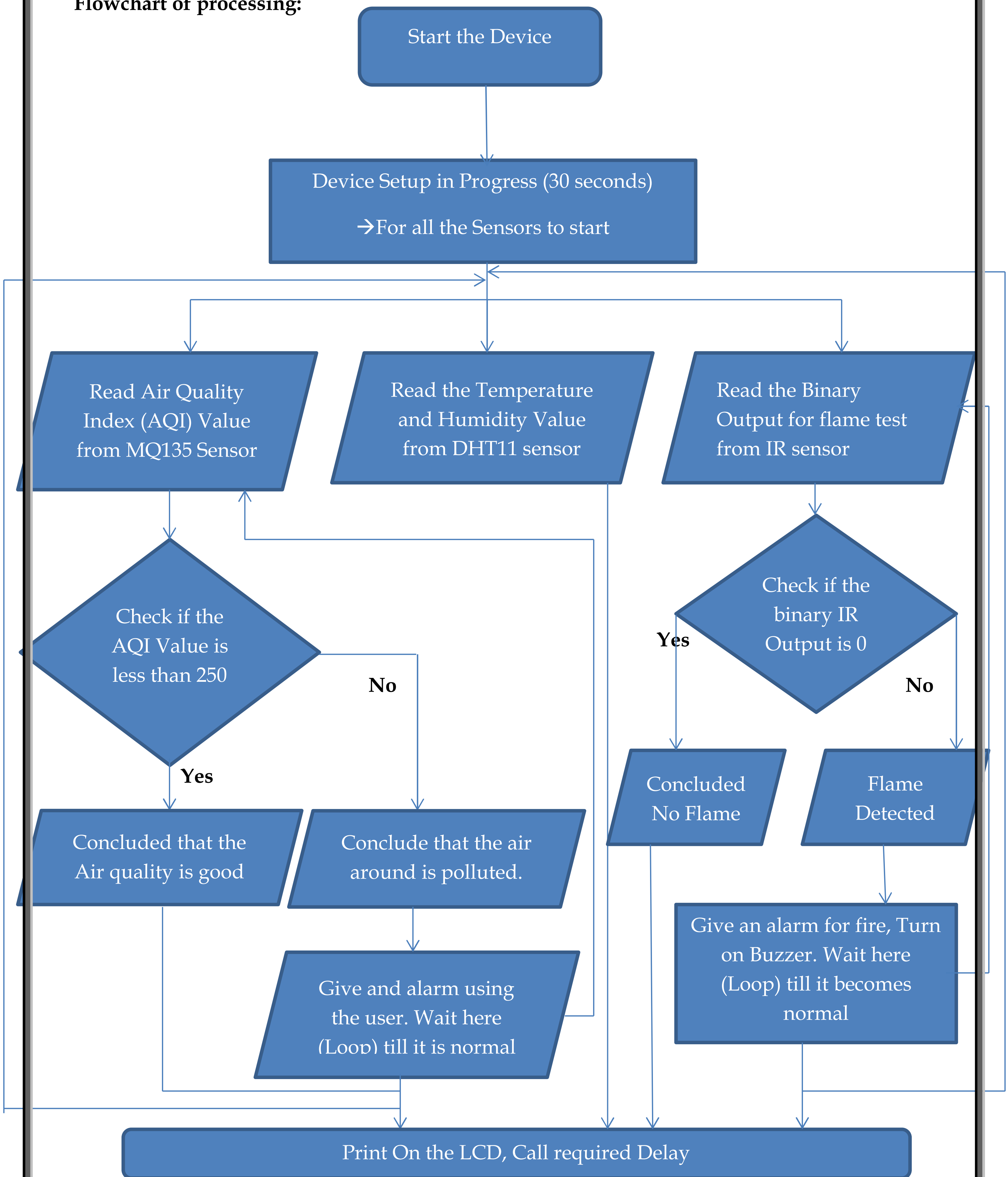
(Write a brief introduction about what is the project about? What is existing ? What motivates you to take this work and How your work is different from existing.

BLOCK DIAGRAM:

Circuit Connections**:**



Flowchart of processing:



(Give the BLOCK diagram/FLOW diagram of your project)

COMPONENTS/ SOFTWARE REQUIRED:

Hardware:

S.NO	**Name of the Component**	**Type Description**
1.	Arduino Uno	8- bit Micro-Controller
2.	16*2 LCD Display	16 characters in a row; 2 rows of this kind. Operational Voltage: 5V – 9V, Operated with I^2C converter.
3.	MQ 135 Air Sensor	To detect gases like Ammonia (NH3), sulfur (S), Benzene (C6H6), CO2, and other harmful gases and smoke. Operational Voltage : 5V and consumes around 150mA. It requires some pre-heating before it could actually give accurate results.
4.	Super Debug IR Infrared Flame Detection Sensor	Specialized sensor designed to detect the presence of flames through the infrared spectrum. Operational Voltage : 3.3V. and consumes around 200mA to 300mA of current depending on the model. It uses infrared technology to detect Flames
5.	DHT11 Temperature and Humidity Sensor	The DHT11 is a digital sensor used for measuring temperature and humidity in the surrounding environment. The DHT11 sensor typically has a temperature measurement range of 0°C to 50°C (32°F to 122°F). The DHT11 sensor typically operates on a voltage supply of 3.3V to 5V.
6.	Buzzer	output voltage-5v sound with blinking light alarm system
7.	Jumper wires for connection	Male – Male; Male - Female

(List the required components/software in the tabular form)

PROJECT DESCRIPTION:

The project is about personal environment monitoring assistant which is to enable a common man monitor his surroundings day to day and come to a know whether the surrounding is safe for him or not. Three major processes are involved in this project. These Include:

- Finding the Air Quality Index (AQI)
- Finding the Temperature and Humidity
- Finding the Flame probability

Finding the Air Quality Index:

The system's basic functionality is focused around constant monitoring of air quality via the MQ135 gas sensor. Within the loop, the analogue output from the sensor, which represents air quality in parts per million (ppm), is read and processed. For real-time monitoring, the measured air quality is then printed to both the serial monitor and the LCD.

Working of MQ135:

Sensor Theory:

The MQ135 works on the idea of resistance modification in response to certain gases. It consists of a SnO_2 (tin dioxide) sensor element and a heating element. When the sensor is turned on, the heating element raises the temperature of the SnO_2, making it sensitive to the gases in the environment.

Mechanism for Detecting Gases:

The detecting process is based on variations in the conductivity of the SnO_2 sensor layer caused by gas. SnO_2 has a strong resistance in the absence of target gases. When certain gases are present, they react with the SnO_2 surface, resulting in a decrease in resistance. The amount of resistance change is related to the concentration of gas.

- Based on this mechanism of detecting the gasses, the threshold value for the AQI from this sensor is found to be 250 ppm of overall pollutants.

The default condition is checked with 250 ppm. That is, the analog output from the sensor is compared with the threshold value of 250 and if the value is greater than 250, then it is concluded that the air is polluted. Upon this conclusion,

1. The buzzer is turned on to give an alarm for the user.
2. The necessary prompting to indicate air pollution is given on the display

These processes are continued until the AQI becomes normal.

If the value is normal, then it is just displayed and the message that air quality is normal is also displayed. These processes are continued repeatedly.

Air Quality Alert:

The detection of higher levels of air pollution is an essential component of loop functioning. The loop determines whether the measured air quality (ppm) exceeds a predetermined threshold. When the air quality surpasses this threshold, the system issues an alert:

Monitoring of Air Quality:

The basic functioning of the system is centred on the continuous measurement of air quality using the MQ135 gas sensor. The sensor's analogue output, which indicates air quality in parts per million (ppm), is read and processed within the loop. The measured air quality is subsequently printed to both the serial monitor and the LCD for real-time monitoring.

Pseudo code of implementation:

```
Initialize LCD
Initialize buzzer pin
Initialize MQ135 sensor pin
Create degree symbol for LCD display
Main Setup:
    Turn on LCD backlight
    Set up pin modes for buzzer, MQ135 sensor, and push button
    Begin DHT sensor
    Initialize start time for progress display
    Create custom character for degree symbol on LCD
Main Loop:
    Turn off buzzer
    Read analog output from MQ135 sensor and store it in ppm
    If (time elapsed since start <= disp_time):
      Display "Setup Progress" on LCD
      Display progress dots on LCD
      Clear LCD
    Else:
        Print "Air Quality:" to serial monitor
```

```
Print ppm value to serial monitor
Display "Air Quality:" on LCD
Display ppm value on LCD
If (ppm > threshold):
    While (ppm > threshold):
        Display "AQ Level HIGH" on LCD
        Delay for 500 milliseconds
        Clear LCD
        Display "Air is Polluted!" on LCD
        Print "AQ Level HIGH" to serial monitor
        Turn on Buzzer
        Read analog output from MQ135 sensor and update ppm
Else:
    Turn off Buzzer
    Display "AQ Level Good" on LCD
    Print "AQ Level Good" to serial monitor
    Clear LCD
```

Finding the Temperature and Humidity:

The current temperature and humidity of the surroundings is found out using the DHT11 sensor. The overview of the processing involved is that first, the output from the sensor is digitally read from the sensor for the required parameters. For this, the pre-installed libraries for sensor working are utilized.

Measurement of Humidity:

- The capacitive humidity sensor measures the amount of water vapour in the air. The capacitance of the sensor element fluctuates as the humidity changes.
- This capacitance shift is converted by the sensor into a digital signal that represents the relative humidity.

Temperature Calculation:

- The DHT11's thermistor measures the temperature outside. The resistance of the thermistor fluctuates with temperature.
- The sensor transforms the change in resistance into a digital signal that represents the temperature.

Setup (Initialization):

The process starts by including the libraries required for I2C communication and LCD control. This function starts the LCD display, backlight, and serial communication. Starts a timer to keep track of how much time has passed. The next process is configuring the buzzer, MQ135 sensor, and push button (push) pin modes. This function starts a DHT humidity and temperature sensor (HT). This function generates a custom character for the degree sign on the LCD.

Processing Involved:

Primary Operation is to check whether the time since the program's start is within a given setup progress time. If it is within the setup progress time, a progress message is displayed on the LCD, offering a visual indicator of the initialization status. The LCD shows "Setup Progress" on the first row and cycles through seven dots on the second row. Following the completion of the setup progress time, the code reads the humidity and temperature from the DHT sensor and displays the results on the LCD. Using the custom character, the temperature is displayed alongside the degree symbol. The humidity level is also shown.

Pseudo Code:

```
Include Wire.h

Include LiquidCrystal_I2C.h

Setup:

  lcd.init()

  lcd.backlight()

  Serial.begin(9600)

  startTime = millis()

  pinMode(buz, OUTPUT)

  pinMode(aqsensor, INPUT)

  pinMode(push, INPUT)

  HT.begin()

  start_time = millis()

  lcd.createChar(1, degreeSymbol)


Loop:

  digitalWrite(buz, LOW)

  If (millis() - start_time <= disp_time):
```

```
    lcd.setCursor(0, 0)
    lcd.print("Setup Progress")
    lcd.setCursor(1, 1)
    For i in range(7):
      delay(200)
      lcd.print(".")
    lcd.clear()
  Else:
    humidity = HT.readHumidity()
    tempc = HT.readTemperature()
    lcd.print("Temperature:")
    lcd.print(tempc)
    lcd.write(1)
    lcd.print("C")
    lcd.setCursor(1, 1)
    lcd.print("Humidity: ")
    lcd.print(humidity)
    lcd.print("%")
```

Finding the Flame probability:

The flame probability is determined using the IR sensor. The sensor detects the flame using a potentiometer.

Working of IR Sensor:

During combustion, flames release distinctive infrared radiation. This radiation appears as flickering infrared light. The detector is intended to distinguish the precise wavelength of infrared light produced by flames from background IR radiation. The electrical signal is then analysed to see if it matches the typical pattern of flame-generated IR radiation. To decrease false alerts, IR flame detectors frequently include filters that distinguish between flame-related and non-flame-related IR sources. This aids in discriminating between radiation from flames and radiation from other heat sources. When the sensor senses a flame, an output signal is generated. This signal can be used to trigger alarms, control systems, or other safety precautions.

Since IR sensors are made with Photo-sensitive coils, the sensor produces a high output (5V) when there is intense light detected near the coil. If there is any flame near the sensor, the sensor produces a digital high (1) output. So, the presence of any flame is detected with this condition.

This implies that if the sensor output is low (0), then it is concluded that there is no flame and safe. Instead if the sensor output is high (1), then, it is concluded that there is flame. Also, if flame is detected, the alarm sound is given out by turning the buzzer on. The buzzer is kept on till the flame probability is 0; ie until there is no flame.

Setup:

- Analogue pin A0 is connected to the MQ135 sensor.
- Digital pin 9 is connected to the fire sensor.
- Digital pin 8 is connected to the buzzer.
- Digital pin 6 is connected to the push button
- For environmental monitoring, a DHT temperature and humidity sensor (attached to pin 2) is also employed.

The main loop:

- For the first 30 seconds, the system displays a setup progress notification.
- Following initialization, the loop checks for the presence of a fire.
- If a fire is detected, the LCD shows a fire alarm message, activates the buzzer, and waits for the fire to be extinguished.
- If there is no fire, the LCD shows a clear and safe message.
- The loop is repeated every second, and the LCD is cleared in preparation for the next iteration.

Pseudo code of implementation:

```
// Include necessary libraries

Include Wire library

Include LiquidCrystal_I2C library

Include DHT library

// Initialize LCD, sensors, and variables

Initialize LiquidCrystal_I2C object with address 0x27 and dimensions 16x2

Initialize variables for timing and thresholds

Initialize pins for sensors, buzzer, and push button

// Setup function

Setup LCD, Serial communication, and sensors

Set up initial conditions and create custom characters
```

```
// Main loop
Repeat indefinitely {
  Read air quality value from MQ135 sensor
  // Display setup progress for the first 30 seconds
  If (elapsed time < 30 seconds) {
    Display "Setup Progress" on LCD
    Wait and display progress dots
  }
 Else {
    // Check for fire presence
    Read fire sensor status
    If (fire detected) {
      // Display fire alert and activate buzzer
      Display "Fire Alert! Run!" on LCD
      Activate buzzer
      Wait for fire condition to clear
    }
  Else {
      // Display clear and safe message
      Display "Clear and Safe No Fire Danger" on LCD
    }
    // Delay for one second and clear the LCD
    Delay for 1 second
    Clear LCD
  }
}
```

As mentioned earlier, all these processes and repeated again and again continuously until there is any pollution or fire alert.

(Describe in detailed operation about your project)

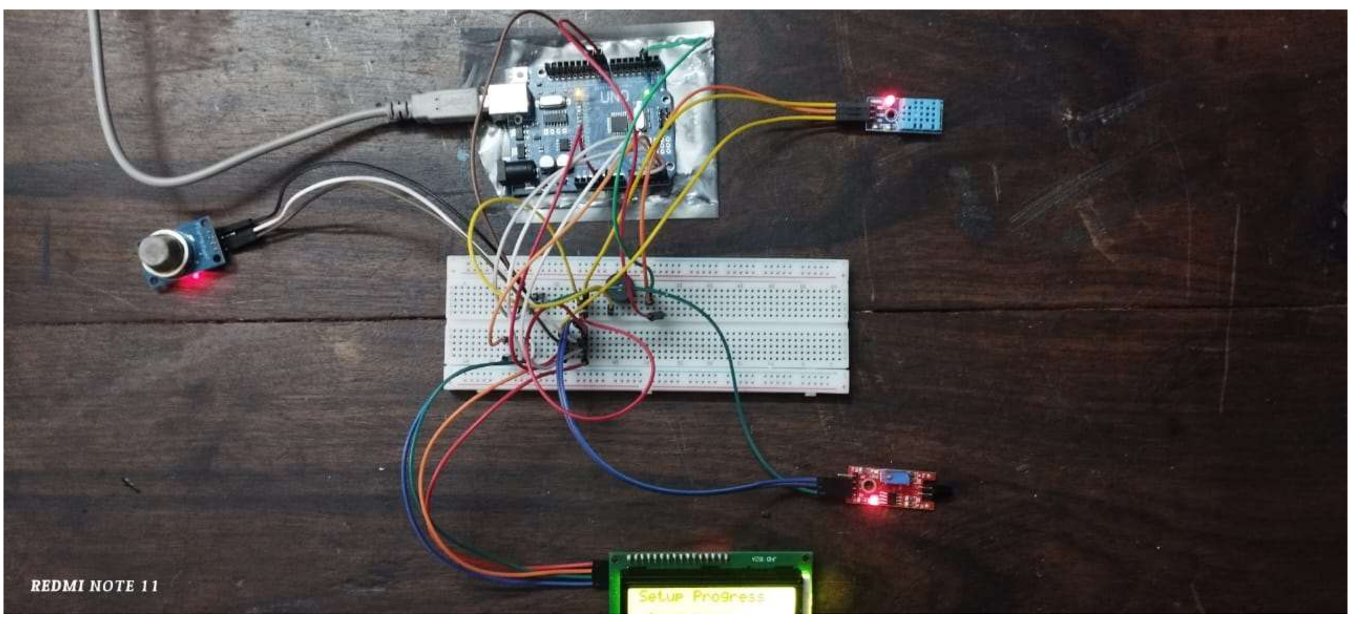
CONCEPT LEARNED:

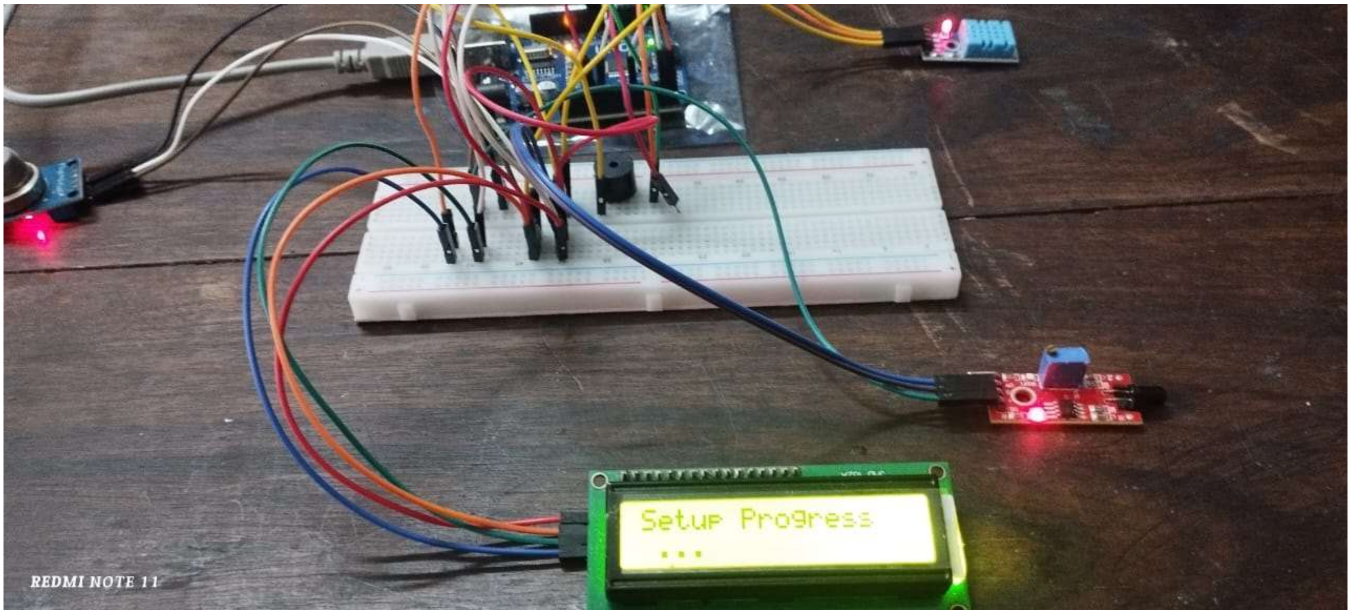
1. Basic Arduino programing:
   →Input and Output pins declaration.
   →Finding the address type and baud rate of LCD.
   →Reading the analog output and processing.
   → Reading the digital output and processing.
   →Basic conditional statements and their syntax.
   → Basic loop statements (for and while) and their syntax.
2. Arduino port and pin details.
3. I^{2}C interface module description and working.
4. LCD Interfacing with Arduino.
5. Hardware output determination.
6. Digital sensor Interfacing with Arduino (IR,DHT11).
7. Analog sensor Interfacing with Arduino (MQ135).
8. Hardware and software analysis.

   (List out the main concept you have learned from this project)

IMPLEMENTATION:

Overall Hardware Connection:

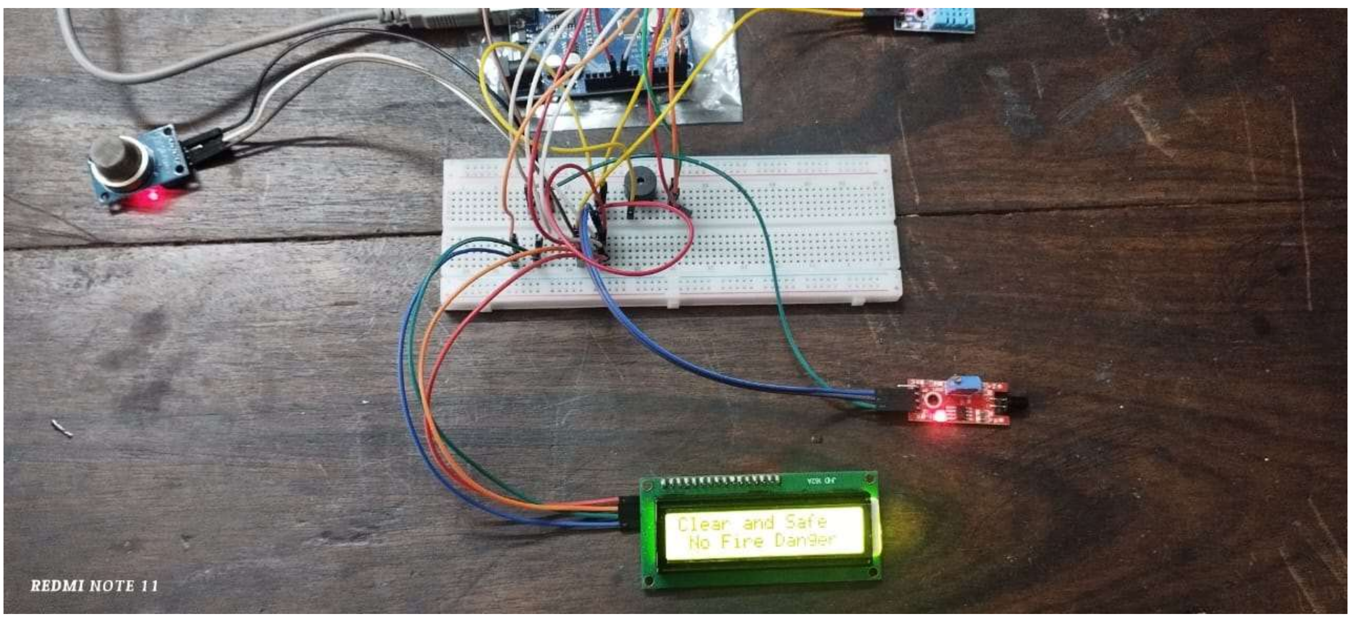




In this stage, the system displays 'Setup Progress' as it waits for all the sensors to start. This because, the MQ135 sensor takes a heating time of 30 seconds, DHT sensor takes a time of 15-18 seconds and also, the IR flame test sensor takes 10 seconds time to start.

Once all the sensors are started and set to work, the system first displays the AQI value and gives a suggestion of whether it is good or if the air is polluted. In this case, the AQI is good.

Once the air quality id displayed, the system prints the temperature and humidity of the surrounding environment. This Step is just displaying to the user.

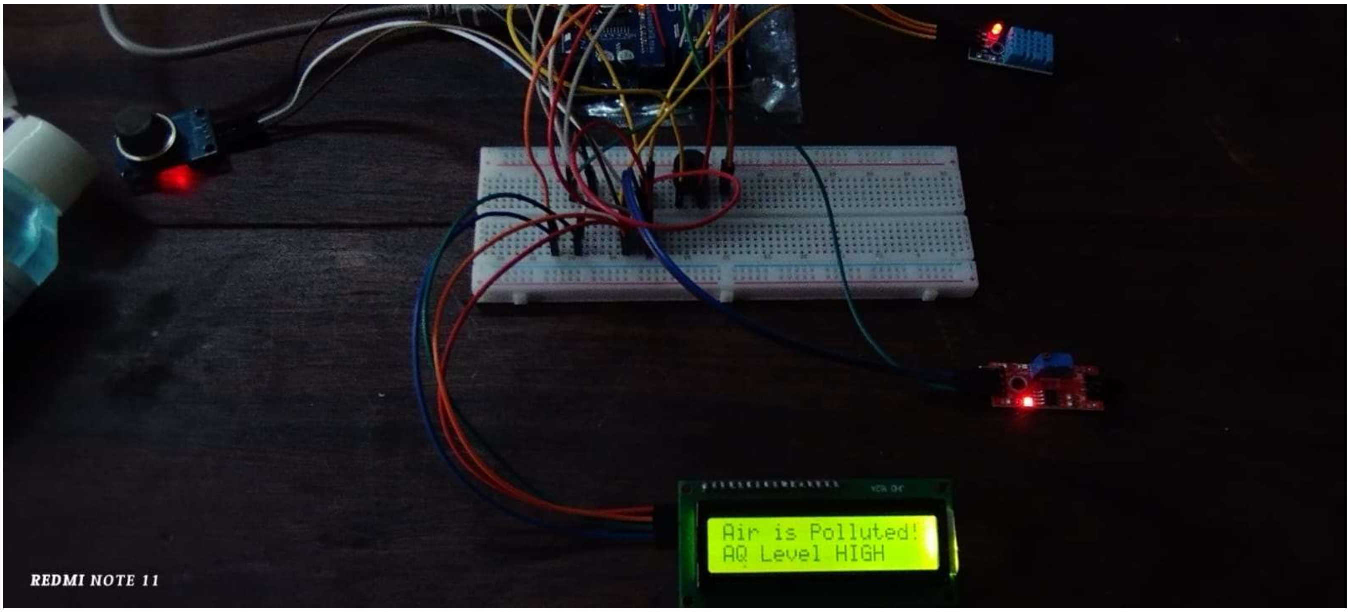


Finally, the system prints the fire probability, ie whether there is any flame or not. In this case, there is no flame. Hence, the system displays “Clear and Safe. No fire Danger”. But, if there is any fire, appropriate message will be displayed and an alarm would be given to the user by turning the buzzer on.

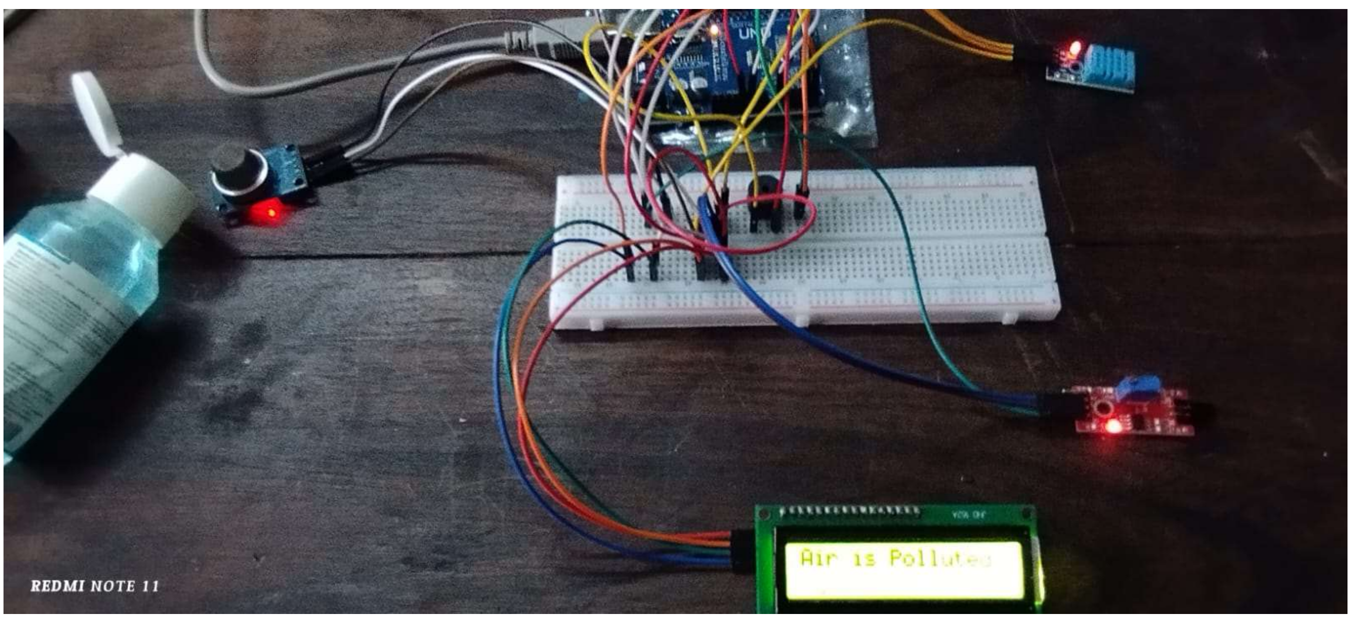
These processes keep repeating continuously and the recordings are displayed every time. However , this cycle is disturbed if there is any discrepancy.

Responses to Discrepancies:

1. Air Pollution:

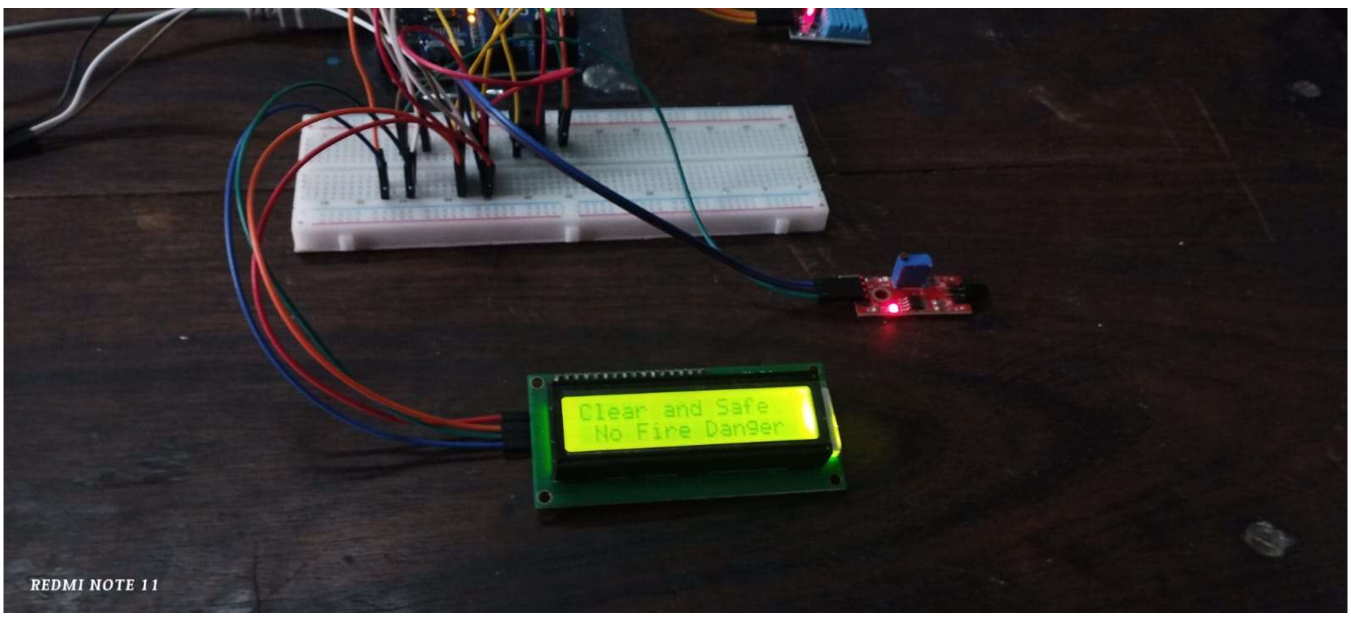


Here, to check the response for rising air pollution, a hand sanitizer bottle is purposefully kept open near the sensor. Such a test is done because from the open hand sanitizer bottle, ammino benzene gas is released, which is a pollutant. Hence, by this way, the pollution test is done and also the system senses it. Corresponding outputs are also displayed.

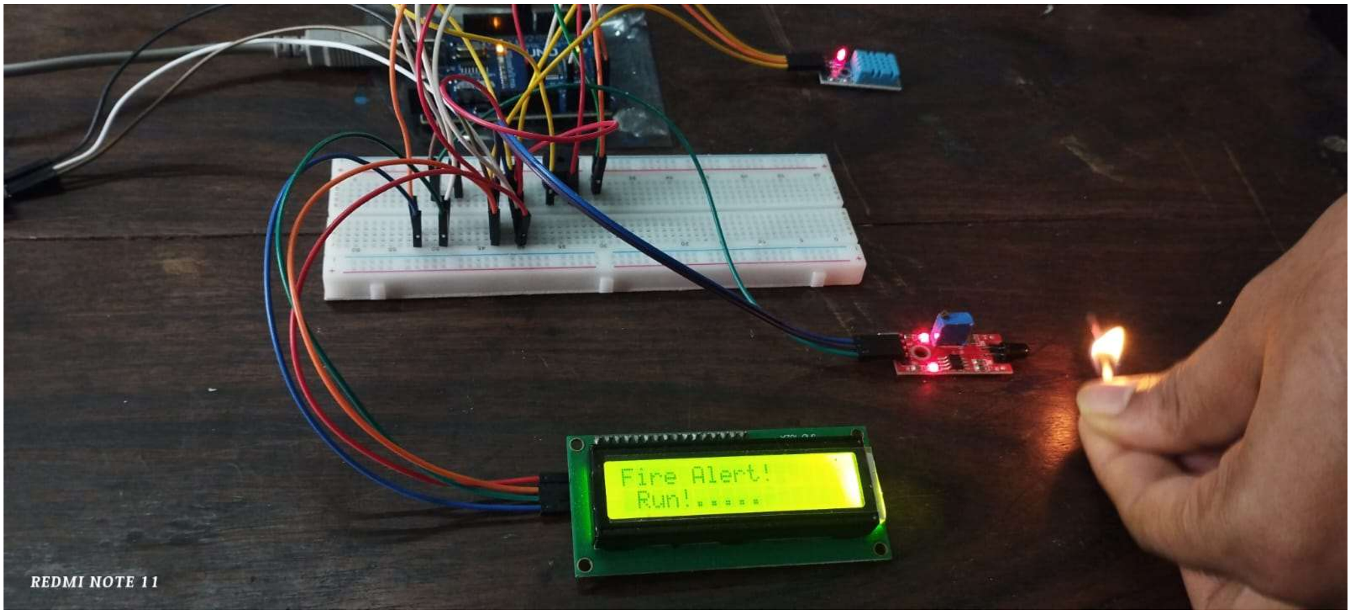


Once, the bottle is removed, the system stabilizes back to normal state and the message "AQ Level Good" is also being printed.

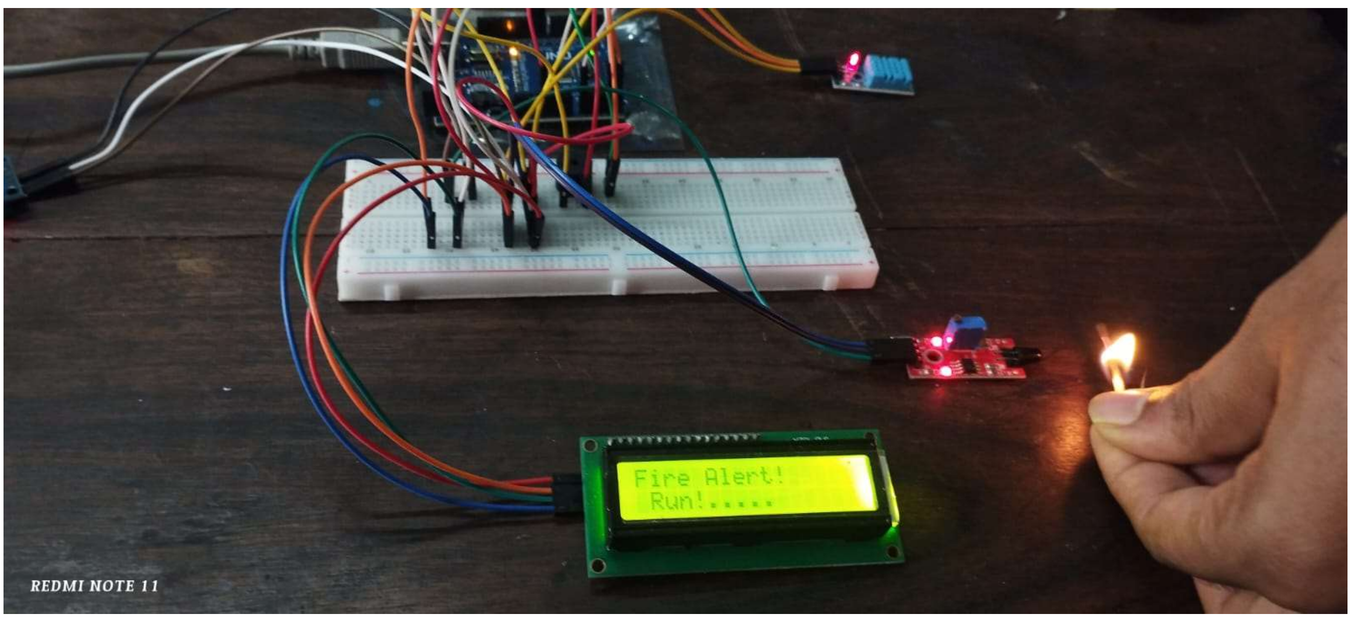
2. Flame Test:



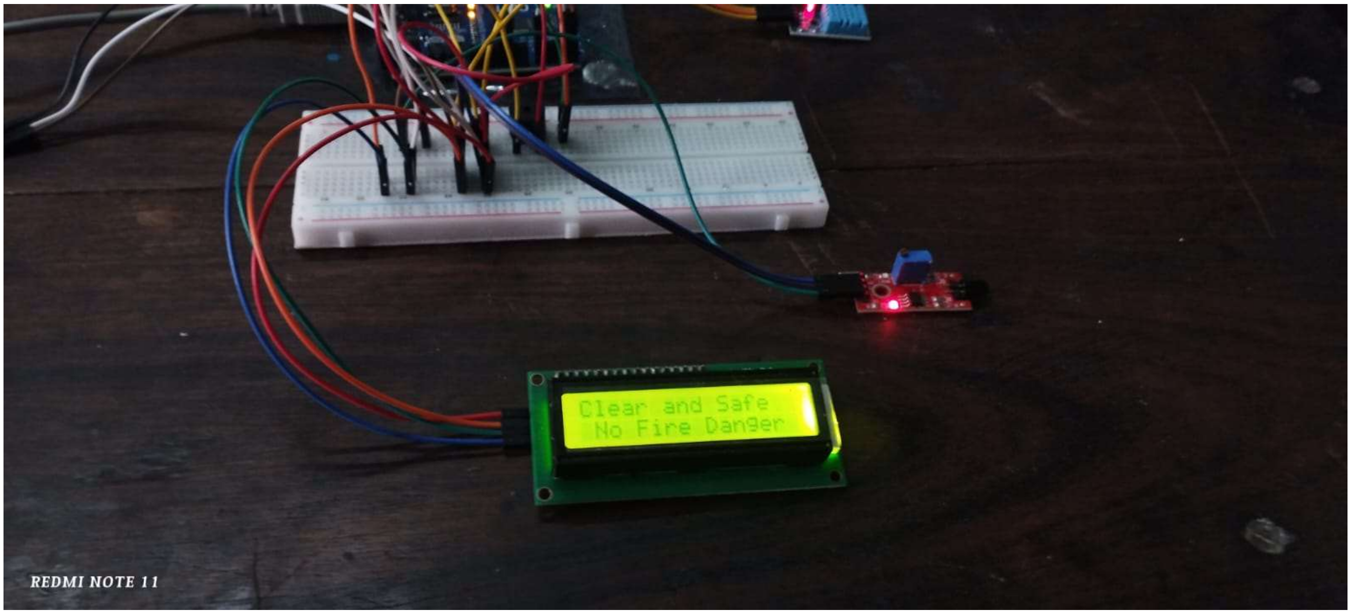
Before performing the flame test, the system prints that there is no flame as usual.



Then a match stick is burnt to check how the system responds. It is evident from the image that the system is able to sense it. It produces an output of “Fire Alert! Run!......”.



The same message is being displayed as the match stick is not removed.



Once the match stick is removed, the system jumps back to normal state and hence it prints that there is no fire.

(Provide the photographs of developed prototype model of your projects under different working condition or scenario. Give a short description under the photograph for all pictures and also provide the YouTube link for your project demo)

CHALLENGES FACED:

Accuracy and dependability of sensors:

- The challenge is to ensure the accuracy and dependability of the sensors used to monitor environmental data.
- Sensors may vary over time, give inconsistent readings, or be influenced by other variables.
- inaccurate data might lead to inaccurate conclusions and suggestions.
- Regular sensor calibration, the implementation of error-checking algorithms, and the use of high-quality, well-maintained sensors can all help to address this issue.

Integration and compatibility of data:

Integrating data from numerous sensors and guaranteeing connectivity with multiple devices is a challenge.

- Different sensor types may have different data formats, which might cause compatibility concerns when connecting to different user devices.
- Difficulty combining and comprehending data from multiple sources.

- During development, standardise data formats, employ suitable communication protocols, and thoroughly test data integration processes.

Privacy and security of users:

- The challenge is to protect user privacy while also safeguarding the data acquired.
- Collecting personal environmental data may raise privacy concerns, and storing and transmitting such data requires stringent security measures.
- Legal and ethical difficulties arise, as well as a loss of user trust.
- Use encryption for data transit and storage, clearly disclose privacy policies to users, and follow applicable data protection regulations.

Power Requirements:

- Managing the power consumption of monitoring equipment is a challenge.
- Continuous monitoring can quickly deplete device batteries, especially in portable applications.
- Limited device uptime, user annoyance can also occur.
- Improve the efficiency of code and algorithms, implement power-saving modes, and examine alternative power sources or efficient energy harvesting approaches.

Design of the User Interface:

- The challenge is to create an intuitive and user-friendly interface for a wide range of users.
- Users' technical backgrounds and preferences may differ. So User dissatisfaction and lower adoption rates can occur.
- Conduct user testing, collect feedback, and iterate on the interface design to ensure that it fulfils the needs and expectations of a wide range of users.

(List out various challenges or difficulties faced while implementing your project)

APPLICATIONS:

The integration of air quality monitoring, temperature and humidity sensing, and fire detection capabilities brings up new opportunities for improving safety, health, and environmental awareness.

1. Indoor Air Quality Management:

The system is useful in indoor environments such as houses, offices, and commercial premises where air quality can have a substantial impact on health. Continuous air quality monitoring aids in identifying and controlling indoor pollution sources, resulting in a healthier living and working environment.

2. Occupational Health and Safety in the Workplace:

The system is a crucial safety tool in industrial or manufacturing environments where workers may be exposed to numerous contaminants or hazardous circumstances. Workers and managers are alerted to possible hazards by real-time air quality data, enabling for rapid interventions and the application of safety regulations.

3. Environmental Studies and Research:

The technology can be used by researchers and environmentalists to collect data for environmental studies. The combination of air quality, temperature, and humidity measurements yields a comprehensive dataset for analysing environmental trends and determining the influence of human activity on the environment.

4. Home Security and Comfort:

In residential applications, the system contributes to occupant safety and comfort. Residents can protect their homes by receiving alerts for excessive levels of air pollution or fire danger. Furthermore, the temperature and humidity monitoring functions improve overall living conditions.

5. Educational Institutions:

The system can be used to educate students about environmental monitoring in educational institutions. It is a hands-on tool for teaching air quality, temperature, and fire safety topics. Students can actively interact with the system to gain an understanding of the significance of environmental awareness.

6. Emergency reaction and Disaster Management:

The system is critical in early identification and reaction in emergency scenarios such as natural disasters or industrial accidents. Rapid detection of poor air quality or the existence of a fire allows for prompt evacuation and emergency intervention.

7. Smart Home Integration:

As smart home technologies become more popular, the system can be connected into smart home platforms. Users can receive notifications on their smartphones or other smart devices, enabling remote monitoring and control of their homes' environmental conditions.

8. Public areas and Urban Planning:

Municipalities and urban planners can use the system in public areas to monitor air quality and estimate the environmental impact of urban development. The information gathered can be used to inform city planning and pollution management decisions.

With its broad applications, the Multi-Sensor Environmental Monitoring System fulfils the requirement for real-time environmental awareness in a variety of sectors. The system has the potential to have a substantial impact on both human well-being and broader environmental considerations, from maintaining health and safety in indoor places to contributing vital data for research and emergency response. Its adaptability and user-friendliness make it an invaluable resource in the domains of health, safety, education, and environmental management.

(How your project is useful in various other applications)

CONCLUSION:

The implemented multi-sensor environmental monitoring system based on Arduino is a diverse and complete solution for evaluating critical environmental parameters. The system offers users with real-time information critical for maintaining a safe and healthy environment by combining multiple sensors, including MQ135 for air quality, DHT11 for temperature and humidity, and a fire sensor.

Air Quality Monitoring:

The capacity of the system to monitor air quality via the MQ135 sensor is a vital feature. The customizable air quality threshold allows users to specify the level at which alarms are issued. This feature is useful in situations where air pollution levels may pose health risks, allowing individuals to take timely actions to reduce exposure.

Temperature and Humidity Sensing:

The DHT11 sensor gives an additional layer of environmental knowledge. Users obtain insight into ambient temperature and humidity levels, which are critical for determining comfort levels, particularly in enclosed environments. The user experience is improved by the visual depiction on the LCD, which includes a custom degree sign.

Fire Detection:

The fire sensor adds an extra degree of security to the system by acting as an early warning system for potential fire threats. The immediate alert displayed on the LCD, along with the audible signal from the buzzer, guarantees that people are instantly aware of any fire-related concerns, allowing for quick evacuation or action.

Improved Situational Awareness:

The LCD display acts as a centralised interface for displaying data from all sensors. This function improves situational awareness by combining information about air quality, temperature, humidity, and fire detection. Users can immediately interpret and respond to the information offered because to the clear and concise presentation.

Increasing Safety and Well-Being:

Finally, the multi-sensor environmental monitoring system does more than just gather data; it actively contributes to the safety and well-being of humans in the monitored area. The system enables users to make informed decisions, take preventive actions, and create a healthier and safer living or working environment by providing fast and accurate information.

In conclusion, the system's combination of sensor technologies, human interaction features, and warning mechanisms makes it a great tool for establishing a mindful and secure living environment.

(Give the conclusion about your project)

APPENDIX:

LCD definition and I²C address finder code:

```
#include <Wire.h>

void setup()

{

Wire.begin();

Serial.begin(9600);

Serial.println("\nI2C Scanner");

}

void loop()

{
```

```
byte error, address;
int Devices;
Serial.println("Scanning...");
Devices = 0;
for(address = 1; address < 127; address++ )
{

Wire.beginTransmission(address);
error = Wire.endTransmission();
if (error == 0)
{
Serial.print("I2C device found at address 0x");
if (address<16){
      Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");
Devices++;
}
else if (error==4)
{
      Serial.print("Unknown error at address 0x");
}
if (address<16){
   Serial.print("0");
   Serial.println(address,HEX);
}
```

```
}
if (Devices == 0){
    Serial.println("No I2C devices found\n");
}
else{
    Serial.println("done\n");
   delay(5000);
}
```

Code for exact Implementation:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>      //Header file for LCD
#include <DHT.h>
#define Type DHT11

LiquidCrystal_I2C lcd(0x27, 16, 2);
unsigned long startTime = 0;
unsigned long duration = 10000;

byte degreeSymbol[8] = {
  B00110,
  B01001,
  B01001,
  B00110,
  B00000,
  B00000,
  B00000,
```

```
  B00000
};

int push = 6; //push button for controling sensor activity is connected to 6th pin of arduino
int buz =8 ;  //buzzer connected to pin 8
const int aqsensor = A0;  //output of mq135 connected to A0 pin of arduino
int threshold = 250;      //Threshold level for Air Quality
unsigned long disp_time = 30000;  //set to 30s
unsigned long start_time;
int fire_check = 9;  //fire sensor connecter to pin 9
int temperature = 2;
DHT HT(temperature,Type);

void setup() {
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
  startTime = millis();
  pinMode (buz,OUTPUT);     // buzzer is connected as Output from Arduino
  pinMode (aqsensor,INPUT); // MQ135 is connected as INPUT to arduino
  pinMode (push,INPUT);
  HT.begin();
  start_time = millis();
  lcd.createChar(1, degreeSymbol);
}
```

```
void loop() {
 digitalWrite(buz,LOW);
 int ppm = analogRead(aqsensor); //read MQ135 analog outputs at A0 and store it in ppm

 if (millis()-start_time <= disp_time){
  lcd.setCursor(0,0);
  lcd.print("Setup Progress");
  lcd.setCursor(1,1);
  for (int i=0;i<7;i++){
   delay(200);
   lcd.print(".");
  }
  lcd.clear();
 }
 else{
  Serial.print("Air Quality: ");  //print message in serail monitor
  Serial.println(ppm);           //print value of ppm in serial monitor

  lcd.setCursor(0,0);            // set cursor of lcd to 1st row and 1st column
  lcd.print("Air Quality: ");
  lcd.print(ppm);                // print sensor op

  if (ppm > threshold)
   {
     while (ppm>threshold){
      lcd.setCursor(1,1);
```

```
      lcd.print("AQ Level HIGH");
      delay(500);
      lcd.clear();
      lcd.print("Air is Polluted!");
      Serial.println("AQ Level HIGH");
      digitalWrite(buz,HIGH);     //Turn ON Buzzer
      delay(600);
      digitalWrite(buz,LOW);
      ppm = analogRead(aqsensor);
     }
  }
else
  {
    digitalWrite(buz,LOW);   //Turn off Buzzer
    lcd.setCursor(1,1);
    lcd.print ("AQ Level Good");
    Serial.println("AQ Level Good");
  }
delay (1000);
lcd.clear();

int humidity = HT.readHumidity();
int tempc = HT.readTemperature();
lcd.print("Temperature:");
lcd.print(tempc);
lcd.write(1); // Display the degree symbol using the custom character
```

```
lcd.print("C");
lcd.setCursor(1,1);
lcd.print("Humidity: ");
lcd.print(humidity);
lcd.print("%");

delay (1000);
lcd.clear();

int fire = digitalRead(fire_check);

if (fire == 1){
  while (fire == 1){
    lcd.setCursor(0,0);
    lcd.print("Fire Alert!");
    lcd.setCursor(1,1);
    lcd.print("Run!");
     for (int i=0;i<5;i++){
      delay(200);
      lcd.print(".");
    }
    digitalWrite(buz,HIGH);    //Turn ON Buzzer
    delay(300);
    digitalWrite(buz,LOW);
    fire = digitalRead(fire_check);
  }
```

```
  }
  else{
   digitalWrite(buz,LOW);
   lcd.setCursor(0,0);
   lcd.print("Clear and Safe");
   lcd.setCursor(1,1);
   lcd.print("No Fire Danger");
  }

  delay(1000);
  lcd.clear();
 }
}
```