EE381 EC Project
INDOOR AIR QUALITY SENSOR

Section: D **Table no. - 12**

Names and Roll no. of group members:

1. **Sumedha Shekhar 201015**
2. **Sanskriti Raj 200875**
3. **P Priya Satwika 200663**

Pre-project work:

Q1. What problem are you trying to solve, and why is it important/interesting?
Ans. Air pollution is a major environmental issue caused by harmful substances in our air. These substances include gases, particulate matter, and biological molecules, which harm human health and the environment.
Air quality refers to the level of pollutants in the air we breathe. Low levels of pollutants characterize good air quality, while high levels characterize poor air quality.
We plan to make an indoor air quality sensor to solve this problem. It will check various air pollutants that will help us take appropriate measures to reduce them to lead a healthy life.

Q2. What are the existing solutions? Describe a few of them and list any shortcomings in them. Is your approach unique in some way?
Ans. Today, gas sensors detect noxious and harmful gases and natural gas leakage but are mainly used for outdoor applications. We need something that can check indoor air quality so that we can take appropriate measures to improve the air we breathe in to reduce its negative impact on our health and lead a healthy life.

Q3. What resources do you require to complete the project? Give a breakup of tasks that you need to accomplish week by week to complete the project.
Ans. The following components are required:

1. Arduino Uno
2. DTH11 sensor
3. 16x2 LCD Display
4. Two 10k potentiometer
5. 220Ω, 47kΩ, two 1kΩ, two 10Ω, 180Ω resistors
6. 5V Buzzer
7. Trigger LED

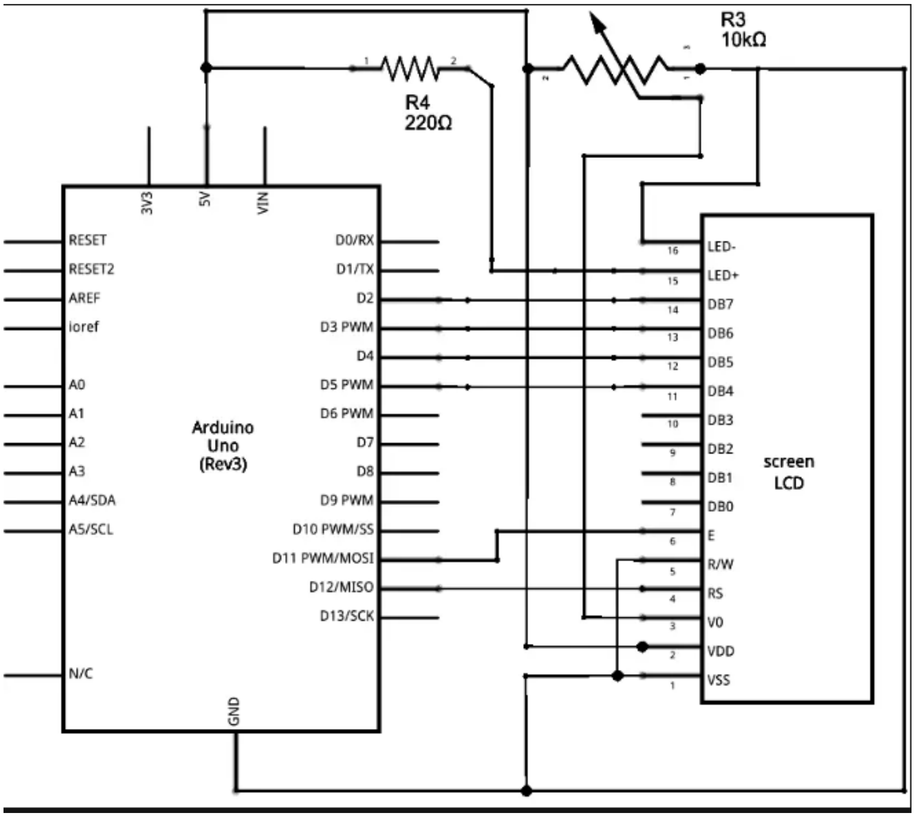
8. Power LED
9. LM393 op amp
10. Two 0.1uF capacitor
11. USB cable
12. Two Breadboards
13. Jumper wires
14. Connecting wires

Timeline

Week 1: LCD 16X2 interfacing with Arduino Uno
Week 2: Breadboarding a circuit to convert the analog output received from the MQ2 sensor to digital output.
Week 3: Connect the DHT11 sensor and MQ2 sensor with Arduino Uno and run the code to check if it's working or not.

Project work:

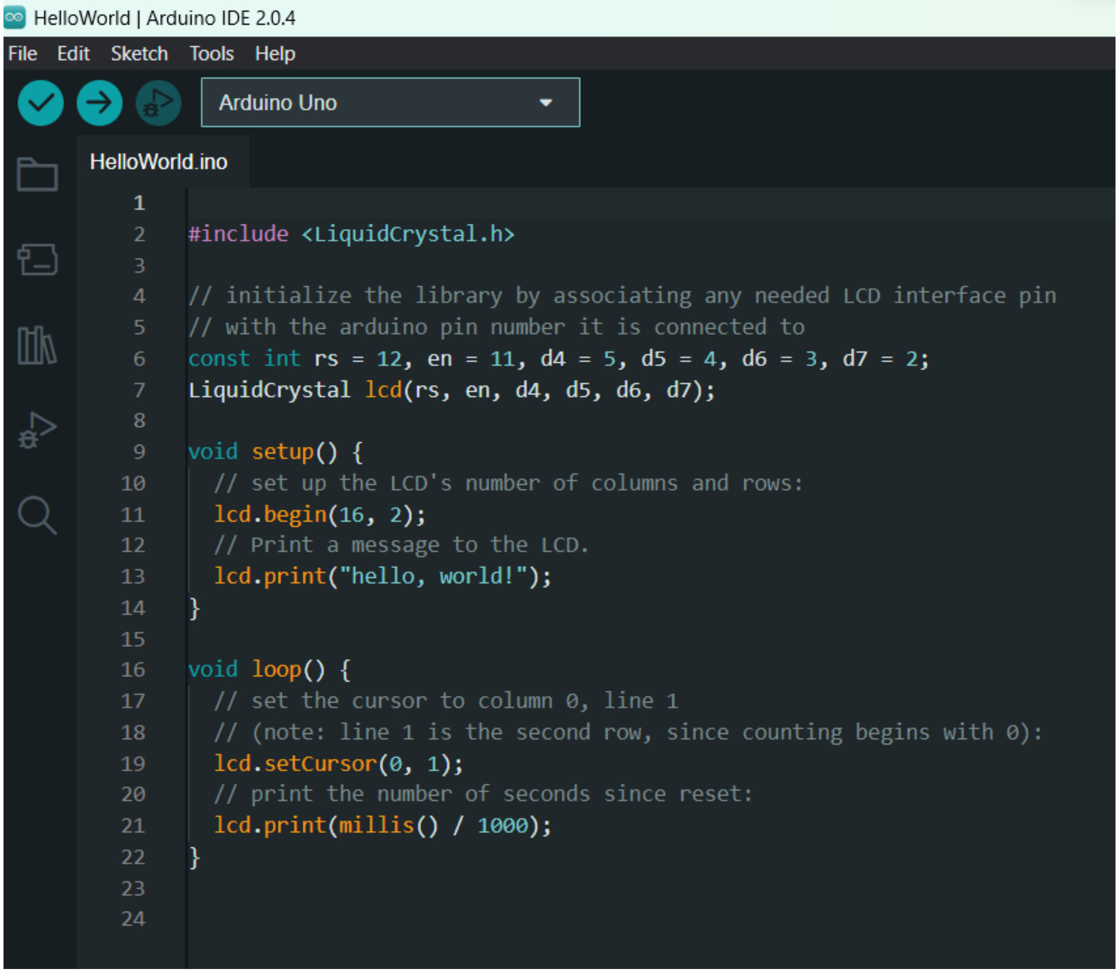
LCD 16X2 interfacing with Arduino circuit diagram:



Implementation details:

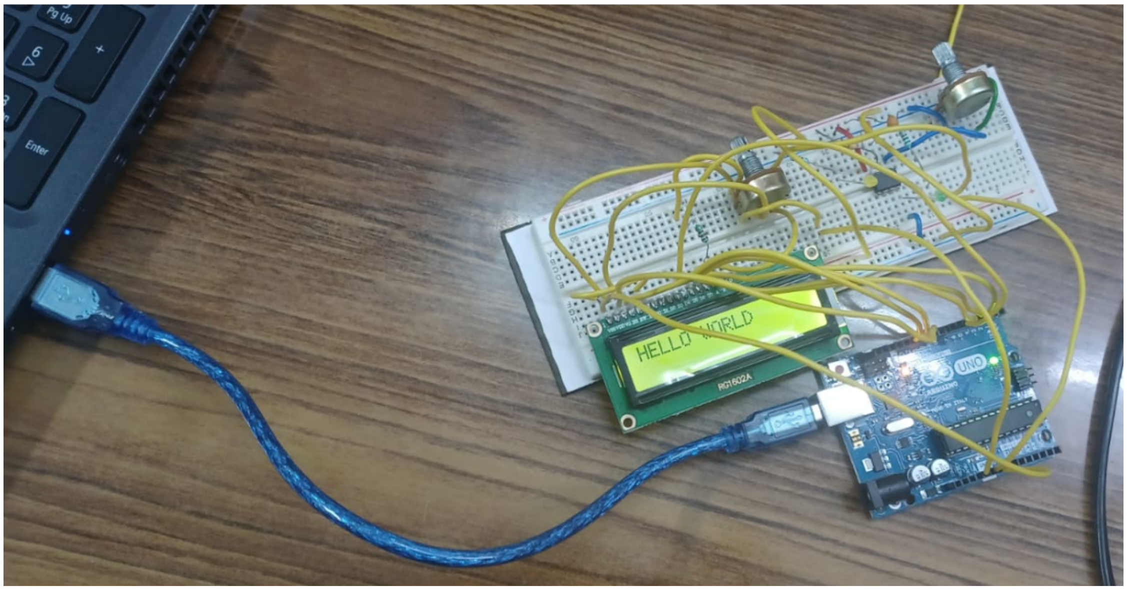
We used LCD 16X2 to display data in 4-bit mode via interfacing it with Arduino Uno.

Code used for implementation:



```

#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}


```

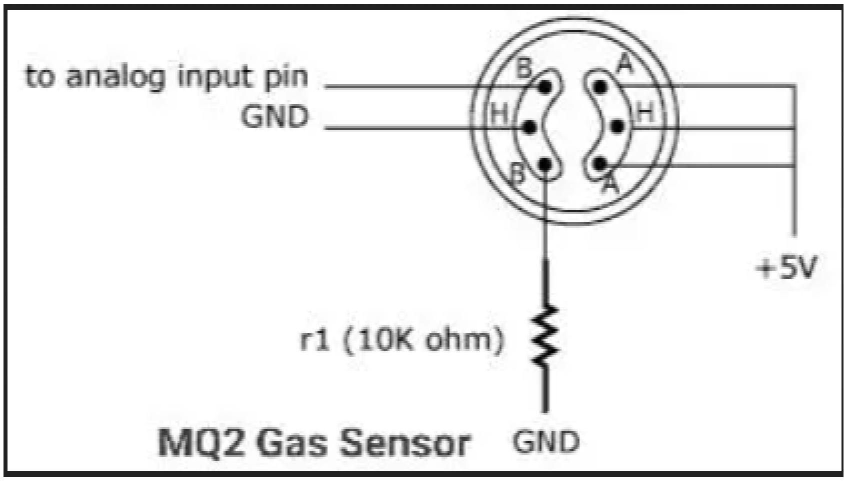
Breadboard circuit - showing output on LCD



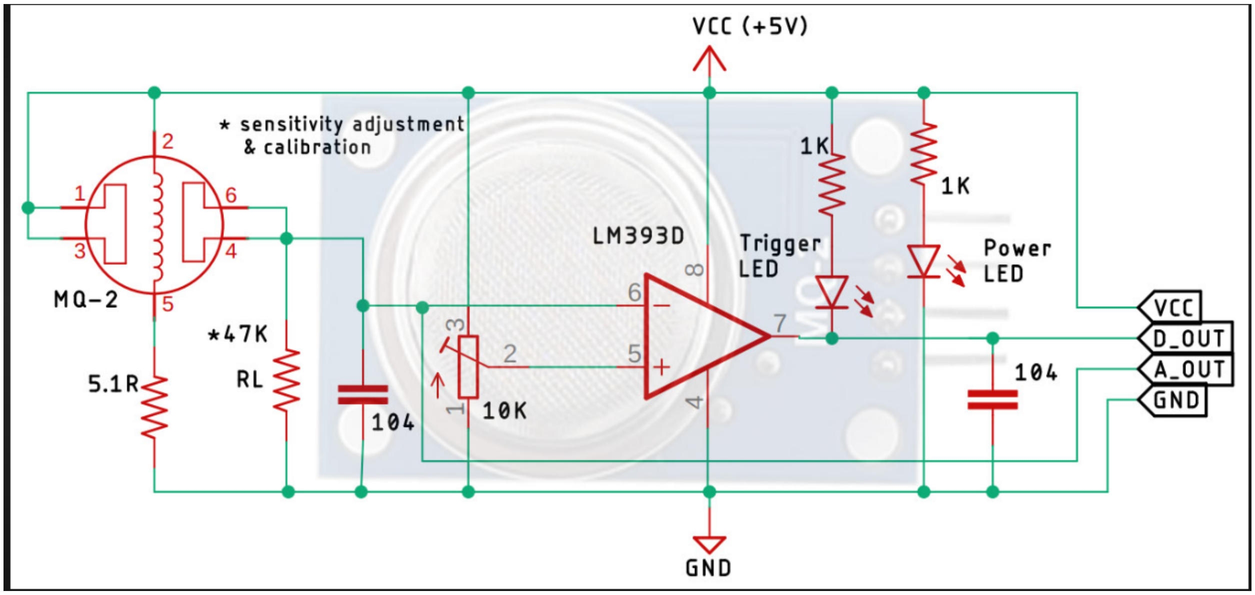
MQ2 Sensor:

The MQ2 gas sensor operates on 5V DC and consumes approximately 800mW. It can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane, and Carbon Monoxide concentrations ranging from 200 to 10000 ppm.

Diagram showing MQ2 Pins:

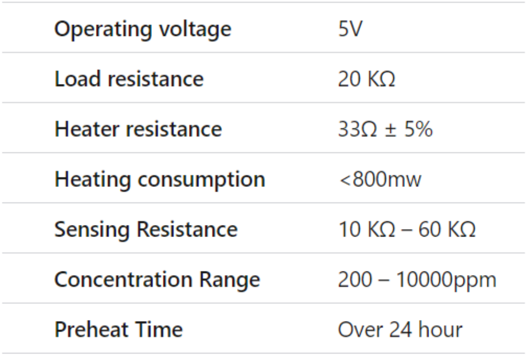


Circuit diagram to convert the analog output received from the MQ2 sensor to digital output



Implementation details:

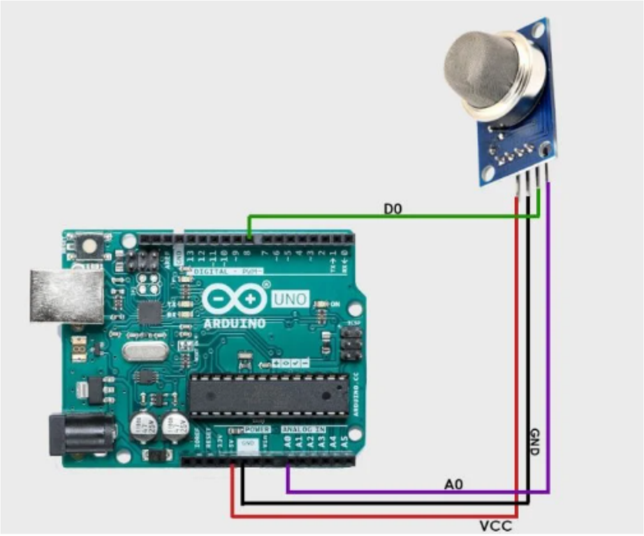
Technical Specification:



Operating voltage	5V
Load resistance	20 KΩ
Heater resistance	33Ω ± 5%
Heating consumption	<800mw
Sensing Resistance	10 KΩ – 60 KΩ
Concentration Range	200 – 10000ppm
Preheat Time	Over 24 hour

Since the MQ2 sensor gives analog output and we need the digital output to display our final CO2 concentration in ppm on the LCD, we breadboard a circuit that would convert analog output from the MQ2 sensor to digital output.
For that, we used LM 393 op-amp, a low-power, low-offset voltage op-amp powered from a +5V supply to convert analog signal to digital signal, 10k potentiometer used to adjust the sensitivity of the MQ2 Gas Sensor module, power LED which turns on when power is applied on the board, trigger LED which turns on when a certain set threshold is reached and, two 0.1uF decoupling capacitors used to reduce noise in the board.

Circuit diagram of interfacing MQ2 gas sensor with Arduino Uno



Connection:

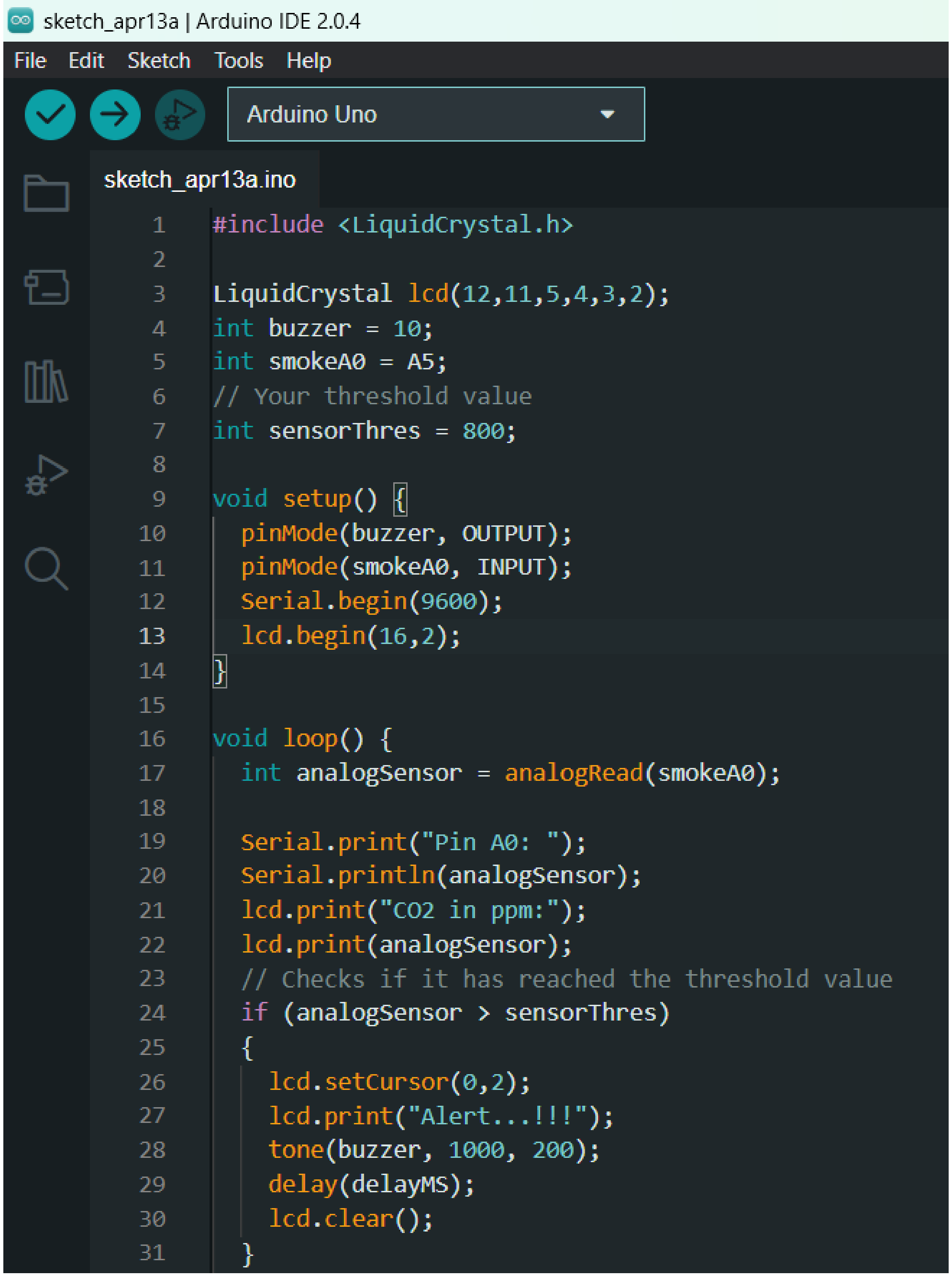
1. Vcc of MQ2 sensor with 5V of Arduino Uno
2. Gnd of MQ2 sensor with Gnd of Arduino Uno
3. The analog output of MQ2 sensor to A5 of Arduino Uno
4. The digital output of MQ2 sensor to D7 of Arduino Uno

Adding 5V Buzzer to the circuit to alarm the user to renew the air

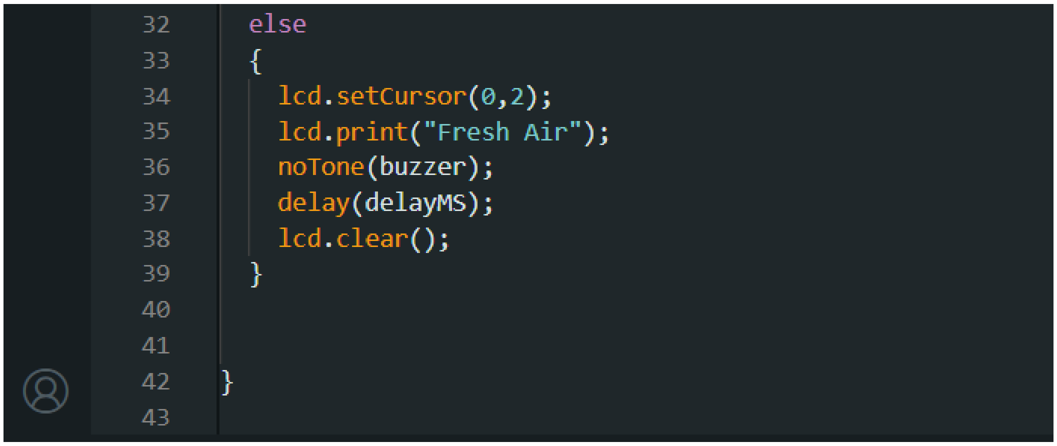
Connection:

1. Positive terminal to D10 of Arduino Uno
2. Negative terminal to Gnd of Arduino Uno

Code used for Implementation:



```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);
int buzzer = 10;
int smokeA0 = A5;
// Your threshold value
int sensorThres = 800;

void setup() {
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
  lcd.begin(16,2);
}

void loop() {
  int analogSensor = analogRead(smokeA0);

  Serial.print("Pin A0: ");
  Serial.println(analogSensor);
  lcd.print("CO2 in ppm:");
  lcd.print(analogSensor);
  // Checks if it has reached the threshold value
  if (analogSensor > sensorThres)
  {
    lcd.setCursor(0,2);
    lcd.print("Alert...!!!");
    tone(buzzer, 1000, 200);
    delay(delayMS);
    lcd.clear();
  }
```



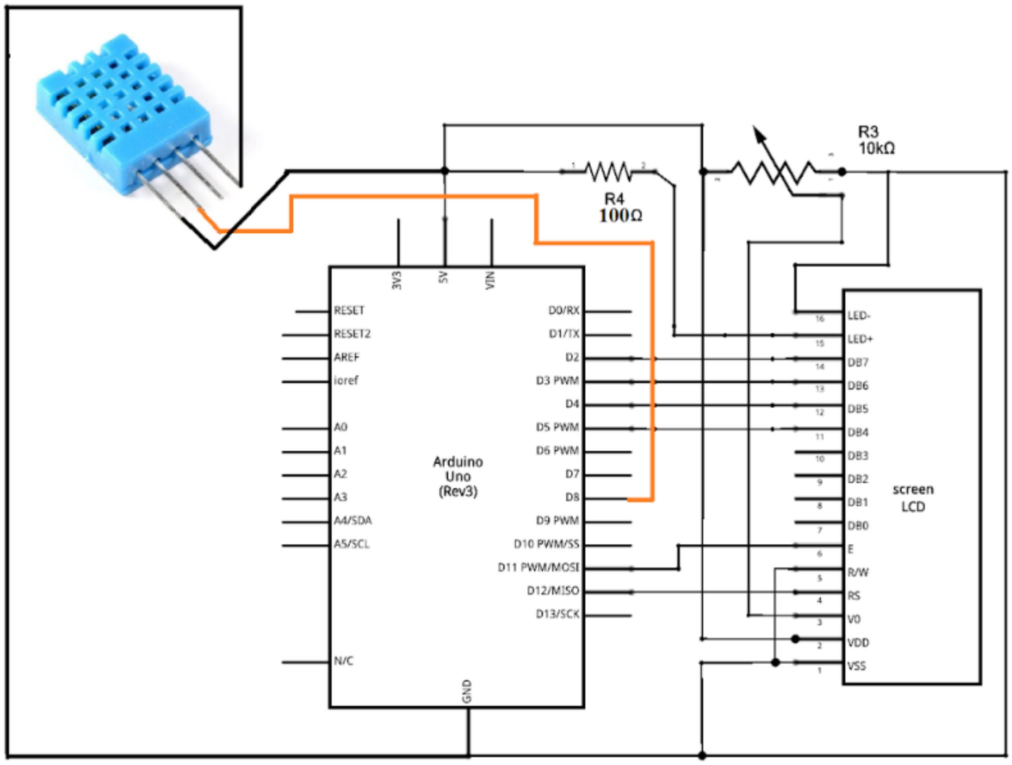
```
  else
  {
    lcd.setCursor(0,2);
    lcd.print("Fresh Air");
    noTone(buzzer);
    delay(delayMS);
    lcd.clear();
  }


}

```

If we run this code, we get the concentration of CO2 in ppm displayed on the LCD. If this value is less than the set threshold value, the buzzer will remain silent, but if it exceeds the threshold value, it starts making noise.

DHT11 Sensor:

The DHT11 is a primary, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits a digital signal on the data pin.

Circuit Diagram of interfacing DHT11 sensor with Arduino Uno



Implementation details:

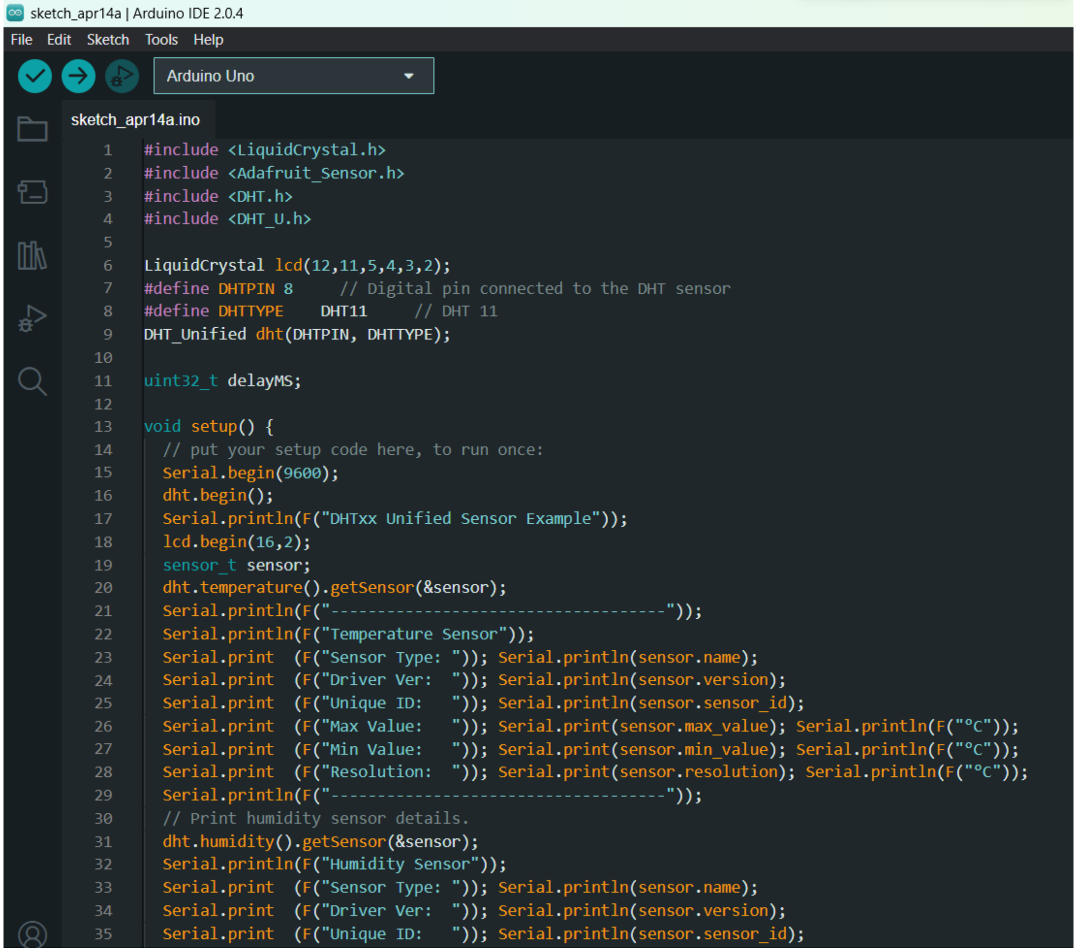
Technical Specification:

- Low cost3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1″ spacing

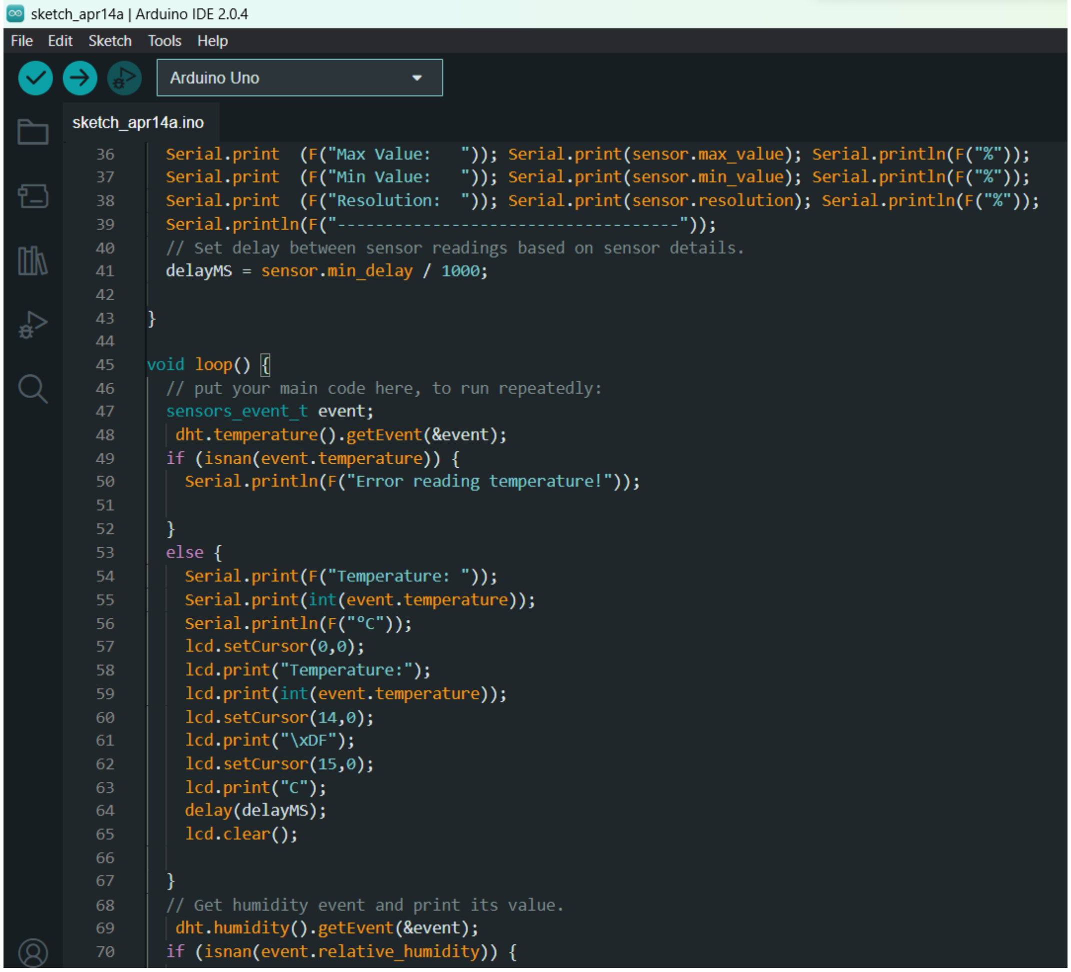
Connection:

1. Signal pin of the DHT11 sensor to D8 of Arduino Uno
2. Vcc of the DHT11 sensor to 3.3V pin of Arduino Uno

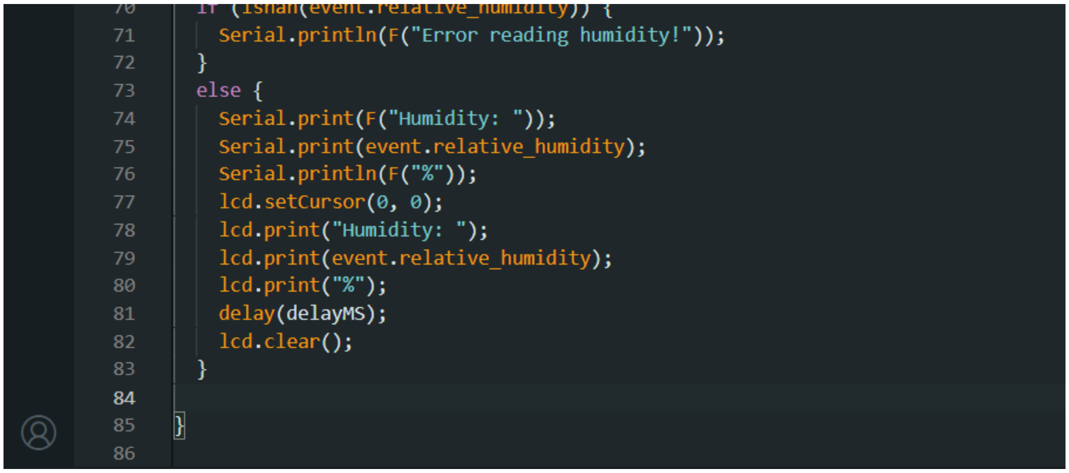
Code used for Implementation:



```cpp
#include <LiquidCrystal.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

LiquidCrystal lcd(12,11,5,4,3,2);
#define DHTPIN 8     // Digital pin connected to the DHT sensor
#define DHTTYPE    DHT11     // DHT 11
DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dht.begin();
  Serial.println(F("DHTxx Unified Sensor Example"));
  lcd.begin(16,2);
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println(F("------------------------------------"));
  Serial.println(F("Temperature Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("°C"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("°C"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("°C"));
  Serial.println(F("------------------------------------"));
  // Print humidity sensor details.
  dht.humidity().getSensor(&sensor);
  Serial.println(F("Humidity Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
```


sketch_apr14a | Arduino IDE 2.0.4
File Edit Sketch Tools Help
Arduino Uno
sketch_apr14a.ino

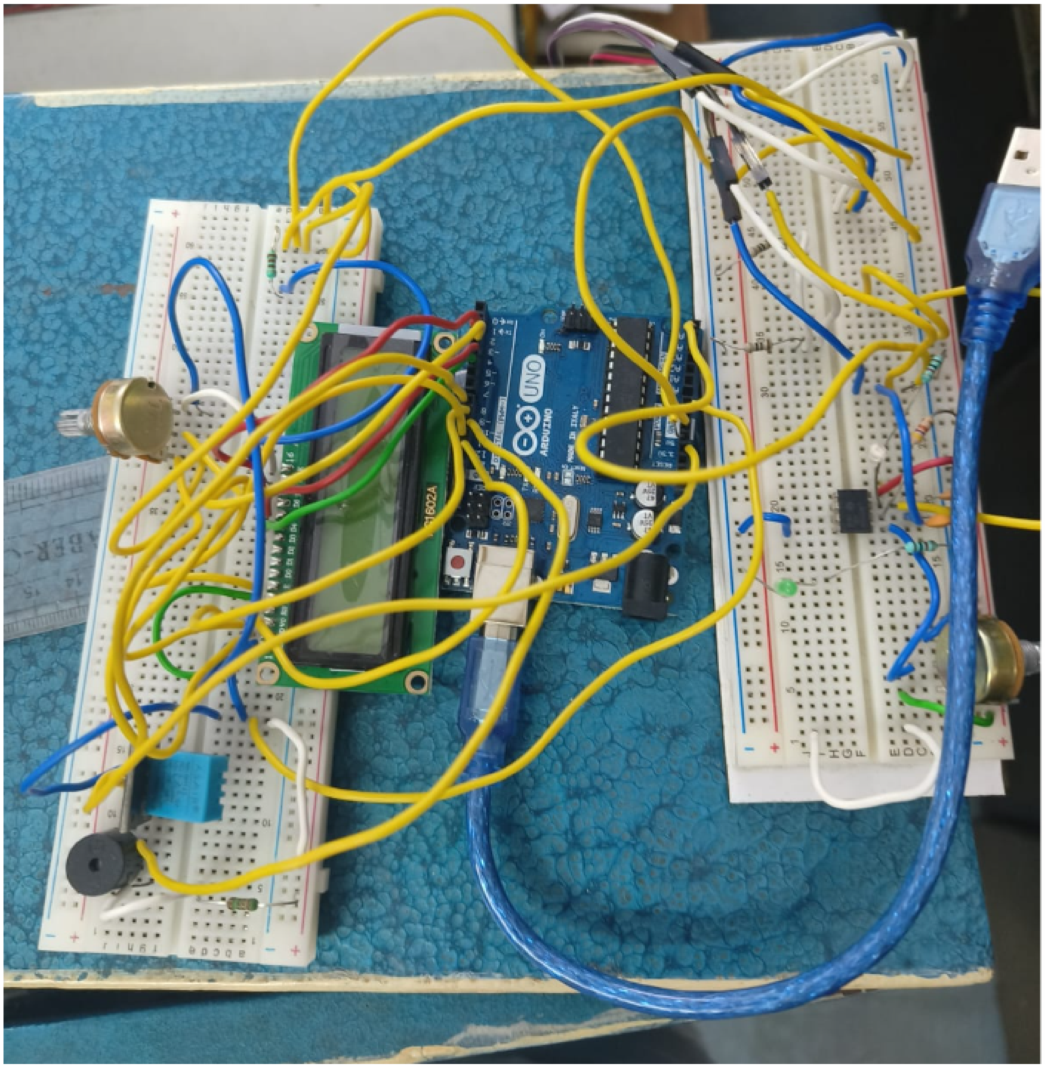
```
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("%"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("%"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("%"));
  Serial.println(F("------------------------------------"));
  // Set delay between sensor readings based on sensor details.
  delayMS = sensor.min_delay / 1000;

}

void loop() {
  // put your main code here, to run repeatedly:
  sensors_event_t event;
   dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println(F("Error reading temperature!"));

  }
  else {
    Serial.print(F("Temperature: "));
    Serial.print(int(event.temperature));
    Serial.println(F("°C"));
    lcd.setCursor(0,0);
    lcd.print("Temperature:");
    lcd.print(int(event.temperature));
    lcd.setCursor(14,0);
    lcd.print("\xDF");
    lcd.setCursor(15,0);
    lcd.print("C");
    delay(delayMS);
    lcd.clear();

  }
  // Get humidity event and print its value.
   dht.humidity().getEvent(&event);
  if (isnan(event.relative_humidity)) {
```



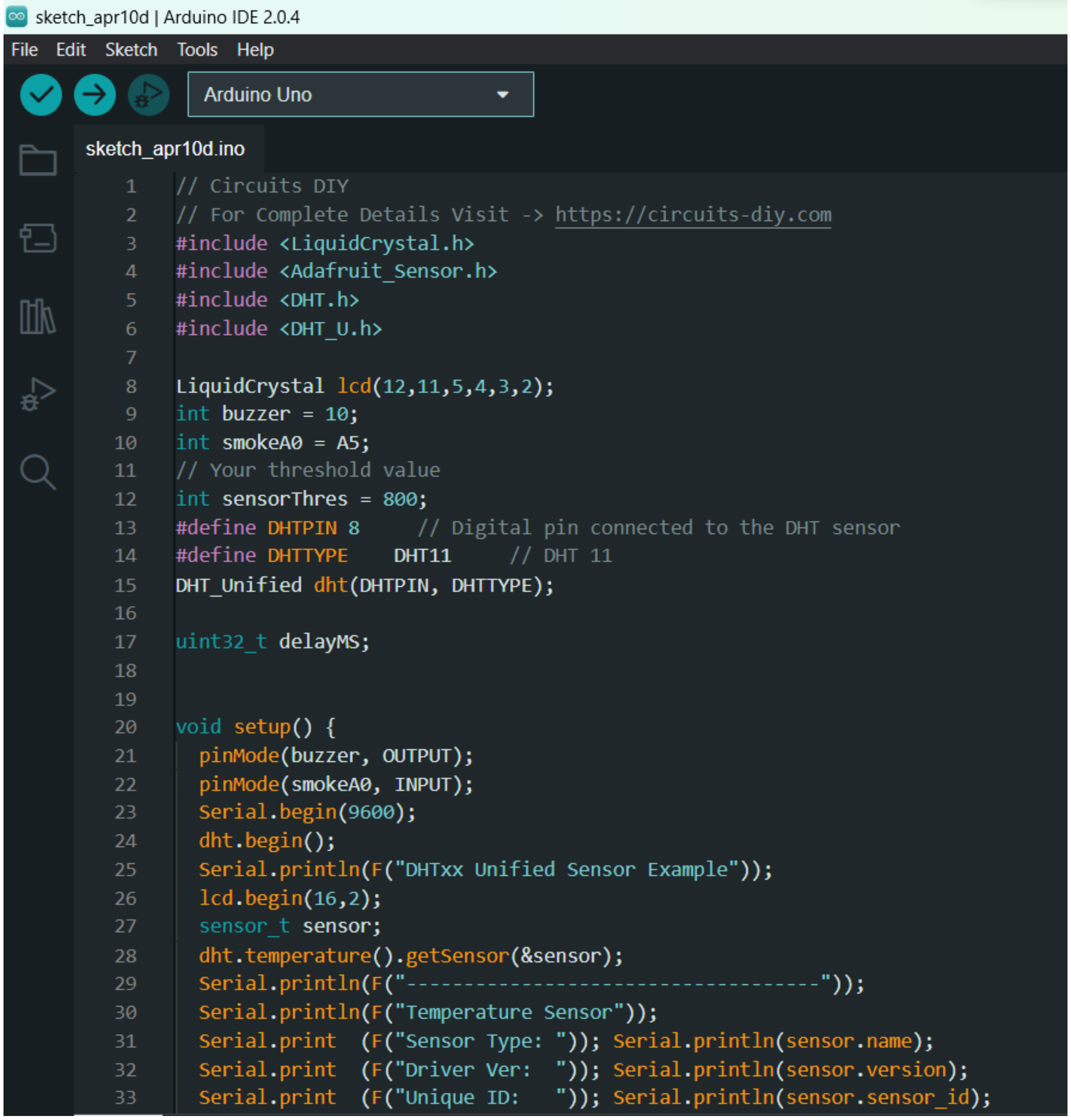
```
    Serial.println(F("Error reading humidity!"));
  }
  else {
    Serial.print(F("Humidity: "));
    Serial.print(event.relative_humidity);
    Serial.println(F("%"));
    lcd.setCursor(0, 0);
    lcd.print("Humidity: ");
    lcd.print(event.relative_humidity);
    lcd.print("%");
    delay(delayMS);
    lcd.clear();
  }

}
```

If we run this code, we will get the temperature and humidity values displayed on the LCD.

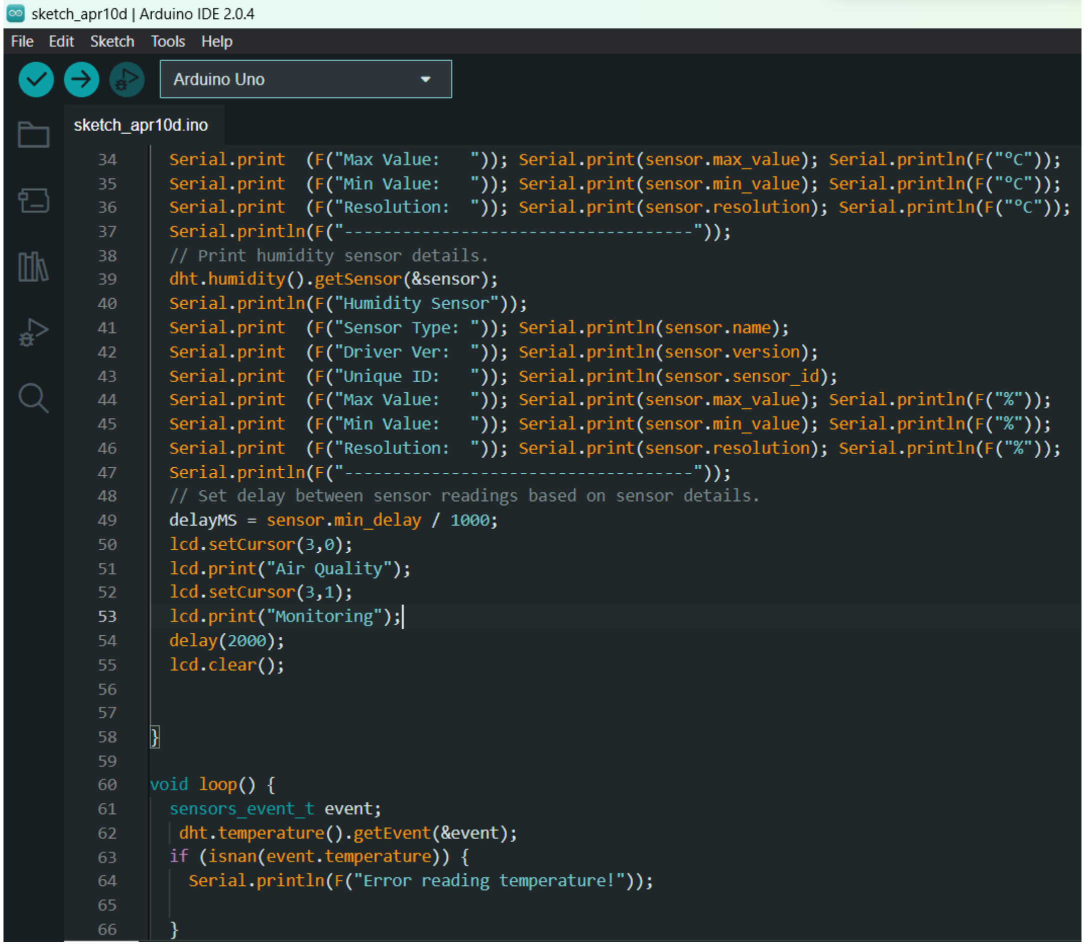
Final Circuit:



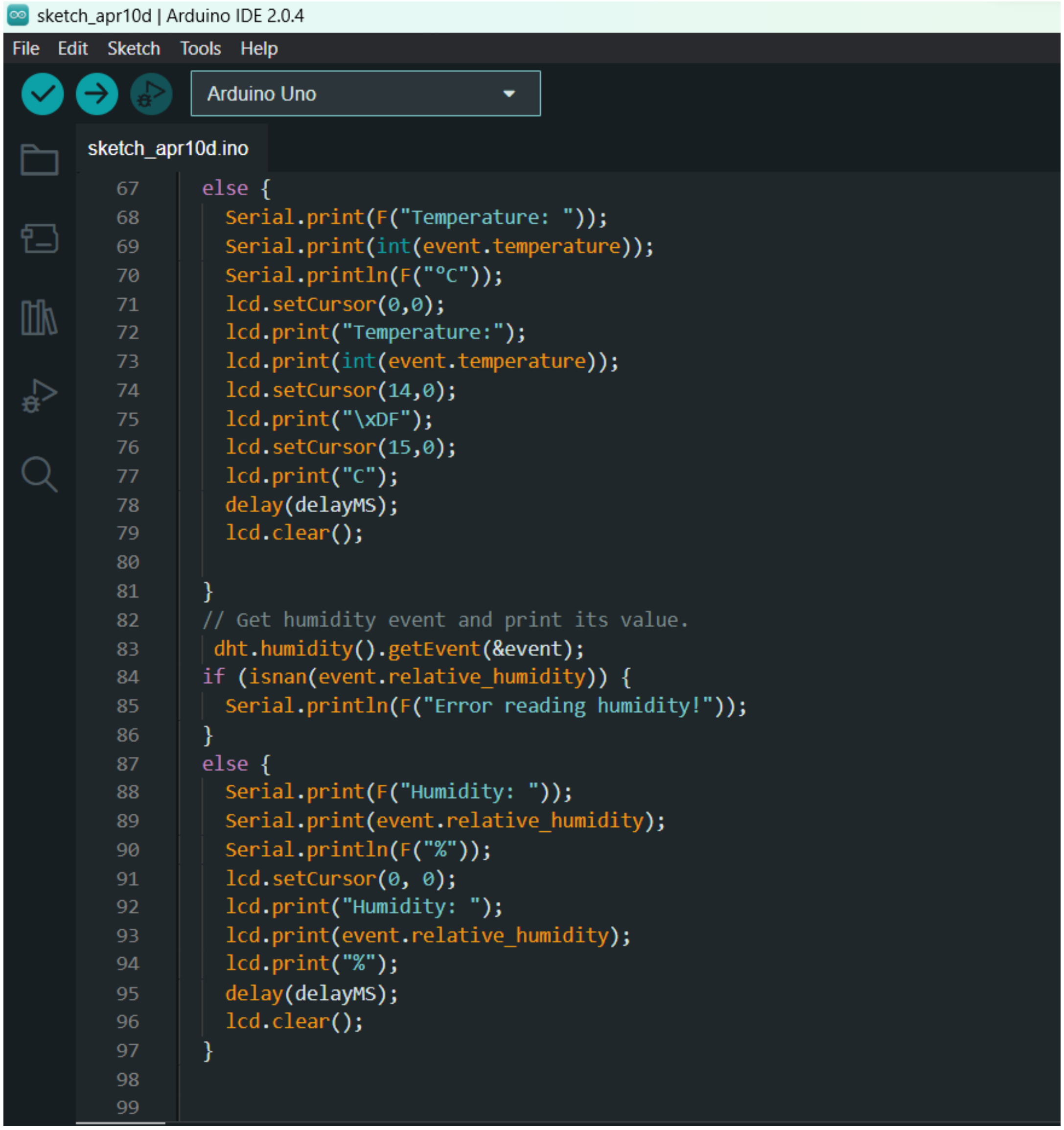
Code used for Implementation:



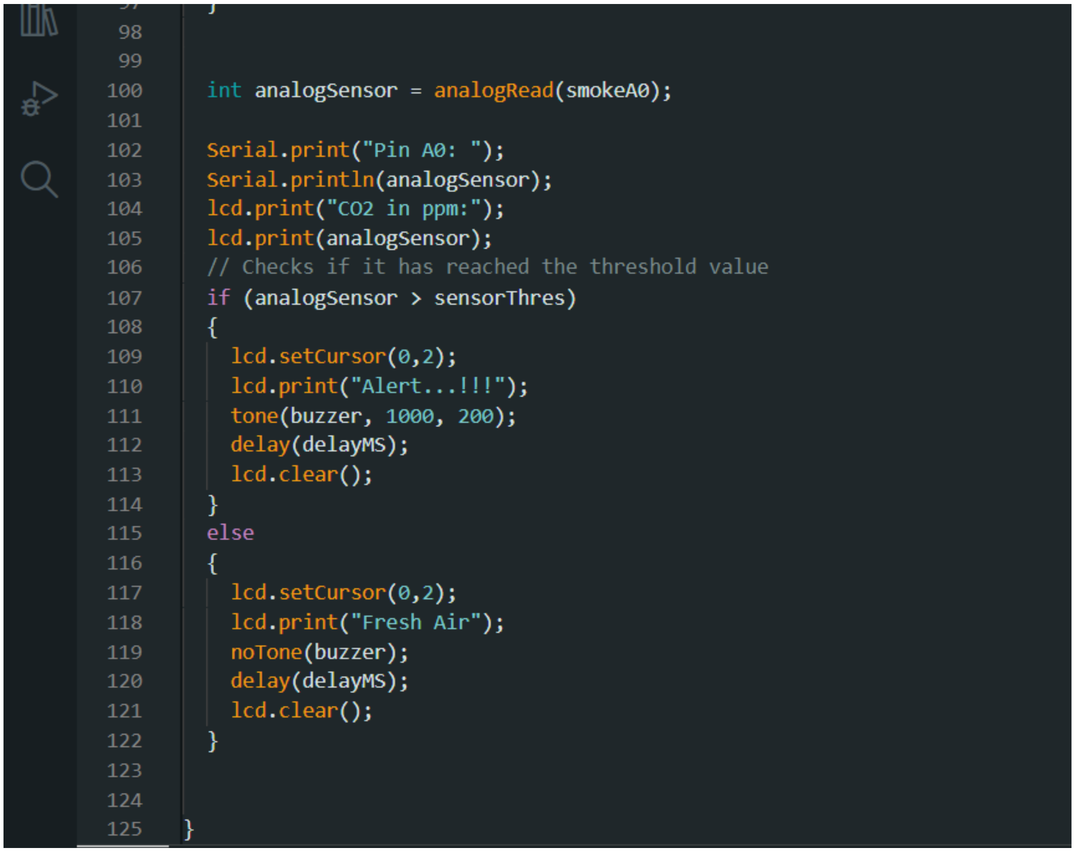
```
// Circuits DIY
// For Complete Details Visit -> https://circuits-diy.com
#include <LiquidCrystal.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

LiquidCrystal lcd(12,11,5,4,3,2);
int buzzer = 10;
int smokeA0 = A5;
// Your threshold value
int sensorThres = 800;
#define DHTPIN 8     // Digital pin connected to the DHT sensor
#define DHTTYPE    DHT11     // DHT 11
DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;


void setup() {
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
  dht.begin();
  Serial.println(F("DHTxx Unified Sensor Example"));
  lcd.begin(16,2);
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println(F("------------------------------------"));
  Serial.println(F("Temperature Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
```



```
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("°C"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("°C"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("°C"));
  Serial.println(F("------------------------------------"));
  // Print humidity sensor details.
  dht.humidity().getSensor(&sensor);
  Serial.println(F("Humidity Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("%"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("%"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("%"));
  Serial.println(F("------------------------------------"));
  // Set delay between sensor readings based on sensor details.
  delayMS = sensor.min_delay / 1000;
  lcd.setCursor(3,0);
  lcd.print("Air Quality");
  lcd.setCursor(3,1);
  lcd.print("Monitoring");
  delay(2000);
  lcd.clear();



}

void loop() {
  sensors_event_t event;
   dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println(F("Error reading temperature!"));

  }
```



```
else {
  Serial.print(F("Temperature: "));
  Serial.print(int(event.temperature));
  Serial.println(F("°C"));
  lcd.setCursor(0,0);
  lcd.print("Temperature:");
  lcd.print(int(event.temperature));
  lcd.setCursor(14,0);
  lcd.print("\xDF");
  lcd.setCursor(15,0);
  lcd.print("C");
  delay(delayMS);
  lcd.clear();

}
// Get humidity event and print its value.
 dht.humidity().getEvent(&event);
if (isnan(event.relative_humidity)) {
  Serial.println(F("Error reading humidity!"));
}
else {
  Serial.print(F("Humidity: "));
  Serial.print(event.relative_humidity);
  Serial.println(F("%"));
  lcd.setCursor(0, 0);
  lcd.print("Humidity: ");
  lcd.print(event.relative_humidity);
  lcd.print("%");
  delay(delayMS);
  lcd.clear();
}


```



```
  }


  int analogSensor = analogRead(smokeA0);

  Serial.print("Pin A0: ");
  Serial.println(analogSensor);
  lcd.print("CO2 in ppm:");
  lcd.print(analogSensor);
  // Checks if it has reached the threshold value
  if (analogSensor > sensorThres)
  {
    lcd.setCursor(0,2);
    lcd.print("Alert...!!!");
    tone(buzzer, 1000, 200);
    delay(delayMS);
    lcd.clear();
  }
  else
  {
    lcd.setCursor(0,2);
    lcd.print("Fresh Air");
    noTone(buzzer);
    delay(delayMS);
    lcd.clear();
  }


}
```

Result

If we run this code, we will get temperature, humidity, and CO2 concentration value in ppm displayed on the LCD screen. If the value of CO2 in ppm is above the set threshold value buzzer will make a noise giving us an indication that we need to renew our air.

Here is the video link to our working project, the Indoor Air Quality Sensor

https://drive.google.com/file/d/1U0hKVh91bnGgNusakhLHONU0NyPZpXMx/view?usp=sharing