

Arduino Air Quality Monitoring System

What is AQI Sensor or Air Quality Index Sensor?

The Air Quality Index (AQI) is a standardized measurement used to communicate how polluted or clean the air is in a specific area at a given time. It provides an easily understandable way to convey information about air quality to the public.

AQI typically takes into account several major air pollutants regulated by environmental agencies worldwide, such as ground-level ozone, particulate matter (PM_{2.5} and PM₁₀), carbon monoxide (CO), sulfur dioxide (SO₂), and nitrogen dioxide (NO₂). Each pollutant has its own scale and threshold levels that determine the AQI value.

The AQI scale ranges from 0 to 500, with specific categories indicating the level of health concern associated with the air quality:

- **0 to 150: Normal to Moderately Polluted**-This range encompasses air quality conditions that are considered acceptable for most individuals, with occasional concerns for sensitive groups in the higher end of the scale.
- **151 to 200: Poor**- Everyone may start to experience health effects, and sensitive groups are more likely to be significantly affected.
- **201 to 300: Very bad** - Health warnings of emergency conditions. The entire population is likely to be affected, and serious health effects may arise.
- **301 to 500: Toxic** - Health alert: everyone may experience more serious health effects due to the extremely poor air quality.

How does Air Quality Monitoring System Work?

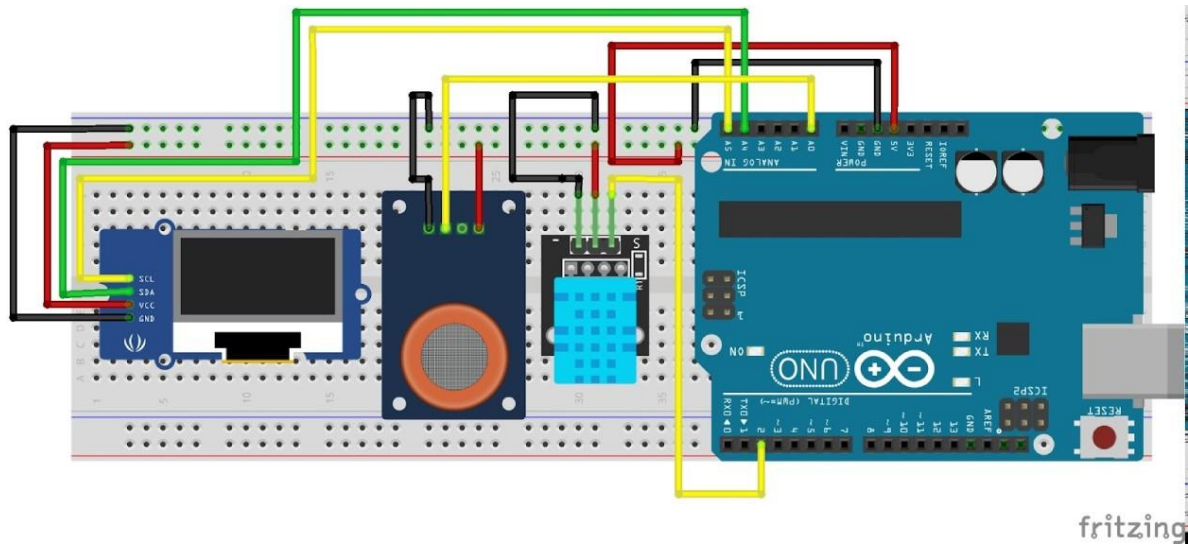
The Air Quality Monitoring System operates by utilizing sensors to detect environmental parameters like gas levels, temperature, and humidity. Specifically, analog readings from the gas sensor, connected to an Arduino's analog pin, provide data on the air quality. These readings are mapped to predefined thresholds, categorizing the air quality as "Good," "Poor," "Very Bad," or "Toxic." Simultaneously, a [DHT11 sensor](#) measures humidity and temperature. The collected data is displayed in real-time on an OLED screen using the Adafruit libraries. By continuously monitoring these parameters, the system offers a snapshot of air quality conditions, enabling users to assess and respond to changes in their environment.

Components Required for this Project are:

- Arduino board
- MQ135 gas sensor for detecting various gases.
- DHT11 temperature and humidity sensor
- OLED display for visual output
- Breadboard and jumper wires

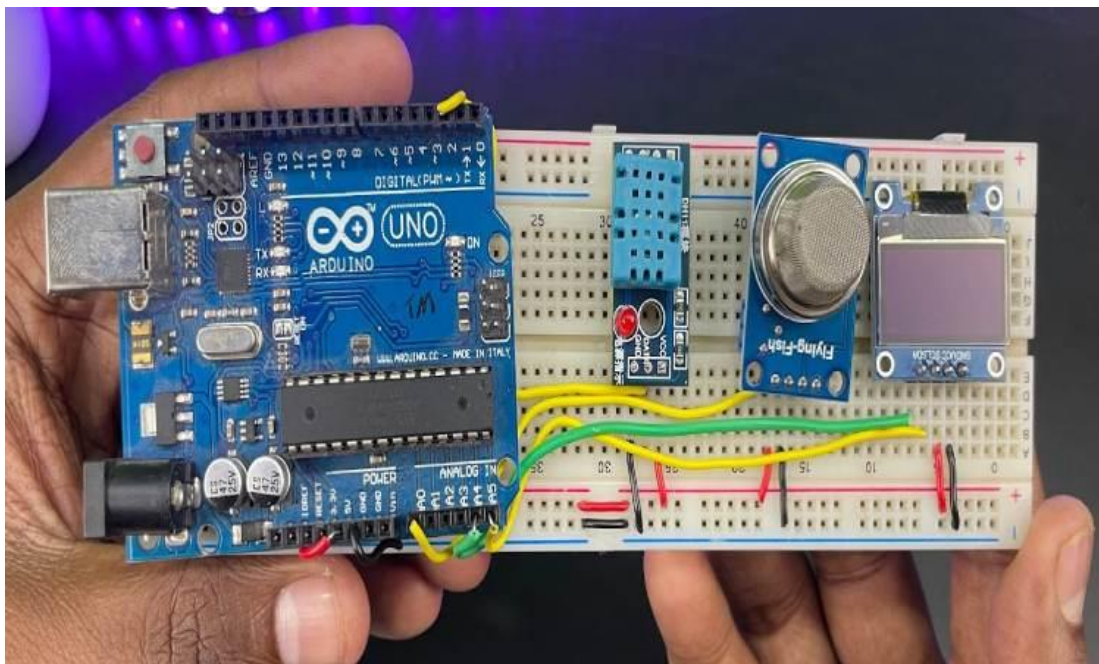
Arduino Air Quality Monitoring System Circuit Diagram

The complete air quality monitoring system project connection will be as follows with an OLED, DHT11 and MQ135 gas sensor connected to Arduino Uno on a breadboard.



The arduino is connected to the OLED using I2C protocol with the help of the SDA and the SCL pins. The MQ135 gas sensor is outputting analog signals which are sent to the analog input pin A0 of the Arduino. The DHT11 sensor has one digital data output pin which is connected to the D2 pin of Arduino.

Here's how the project will look after it is interfaced on a breadboard.



Code explanation for Arduino based Air quality monitoring System

This code monitors air quality and displays sensor readings (temperature, humidity, and gas level) on an OLED display. Let's break it down:

```
#include <SPI.h>

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <dht.h>
```

These lines include necessary libraries for communication protocols (SPI, Wire), graphics handling (Adafruit_GFX), OLED display control (Adafruit_SSD1306), and for the DHT sensor.

```
#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define OLED_RESET 4
```

Constants are defined for the OLED display's width, height, and reset pin.

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
```

An object named display of type Adafruit_SSD1306 is created to control the OLED display using the specified width, height, communication (Wire), and reset pin.

```
#define sensor A0  
  
#define DHT11PIN 2
```

Constants are defined for the analog pin connected to the gas sensor (sensor) and the pin connected to the DHT11 sensor (DHT11PIN).

```
int gasLevel = 0;  
  
String quality = "";  
  
dht DHT;
```

Variables are declared to store gas levels and quality descriptions, and an instance DHT of the DHT sensor is created.

```
void sendSensor() {  
  
    // Function to read temperature and humidity from the DHT sensor  
  
    int readData = DHT.read11(DHT11PIN);  
  
    float h = DHT.humidity;  
  
    float t = DHT.temperature;  
  
    if (isnan(h) || isnan(t)) {  
  
        Serial.println("Failed to read from DHT sensor!");  
  
        Return;  
  
    }  
}
```

```
display.setTextColor(WHITE);  
  
display.setTextSize(1);  
  
display.setFont();  
  
display.setCursor(0, 43);  
  
display.println("Temp  :");  
  
display.setCursor(80, 43);  
  
display.println(t);  
  
display.setCursor(114, 43);  
  
display.println("C");  
  
display.setCursor(0, 56);  
  
display.println("RH   :");  
  
display.setCursor(80, 56);  
  
display.println(h);  
  
display.setCursor(114, 56);  
  
display.println("%");  
  
}
```

This function is defined to read temperature and humidity data from the DHT sensor and display it on the OLED screen.

```
void air_sensor() {  
  
gasLevel = analogRead(sensor);
```

```
if(gasLevel<151){  
    quality = " GOOD!";  
}  
  
else if (gasLevel >151 && gasLevel<200){  
    quality = " Poor!";  
}  
  
else if (gasLevel >200 && gasLevel<300){  
    quality = "Very bad!";  
}  
  
else if (gasLevel >300 && gasLevel<500){  
    quality = "Toxic!";  
}  
  
else{  
    quality = " Toxic";  
}  
  
display.setTextColor(WHITE);  
display.setTextSize(1);  
display.setCursor(1,5);  
display.setFont();
```

```
display.println("Air Quality:");  
  
display.setTextSize(1);  
  
display.setCursor(5,23);  
  
display.println(gasLevel);  
  
  
display.setCursor(20,23);  
  
display.println(quality);  
  
}
```

This function is defined to read gas levels from the analog pin connected to the gas sensor and determine the air quality based on predefined thresholds. It also displays the gas level and quality on the OLED screen.

```
void setup() {  
  
  Serial.begin(9600);  
  
  pinMode(sensor,INPUT);  
  
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3c)) { // Address 0x3D for 12  
8x64  
    Serial.println(F("SSD1306 allocation failed"));  
  }  
  
  display.clearDisplay();  
  
  display.setTextColor(WHITE);  
  
}
```



```
display.setTextSize(2);  
display.setCursor(50, 0);  
display.println(" Air");  
display.setTextSize(1);  
display.setCursor(23, 20);  
display.println("Quality monitor");  
display.display();  
delay(1200);  
display.clearDisplay();
```

```
display.setTextSize(1.5);  
display.setCursor(20, 20);  
display.println("BY Circuit");  
display.setCursor(20, 40);  
display.println("Digest");  
display.display();  
delay(1000);  
display.clearDisplay();
```

```
}
```

The setup function initializes serial communication, sets the sensor pin as input, and initializes the OLED display. It displays introductory messages on the screen.

```
void loop() {  
  display.clearDisplay();  
  air_sensor();  
  sendSensor();  
  display.display();  
  
}
```

The loop function clears the display, reads the gas sensor and DHT sensor data using the defined functions, and continuously updates the display with the latest readings.

This code essentially initializes the sensors, reads their data, determines air quality based on gas levels, and displays all the information on the OLED screen in a loop.

[Working Demo for Arduino Air Quality Monitoring Project](#)

So, this is the complete working demo of the project. You can see that it shows the temperature and humidity values at the bottom of the OLED. at the top, you can see that the level of toxicity is shown which goes from good to toxic after gas from the lighter is introduced.

