CHAPTER 1

INTRODUCTION

Air pollution is one of the most significant environmental issues facing humanity today, with over 90% of the world's population living in areas with poor air quality. Exposure to air pollution can lead to serious health problems, including respiratory illnesses, heart disease, and even premature death. To tackle this issue, governments and organizations worldwide are implementing air quality monitoring systems to measure and analyze the level of pollutants in the air.

An IoT-based air quality monitoring system is a modern solution to this problem that utilizes the power of technology to provide real-time data on air quality. The system consists of a network of sensors that are installed in different locations to measure the level of pollutants in the air. The data collected from these sensors is then transmitted to a cloud-based platform, where it is analyzed and displayed on a web-based dashboard.

One of the primary advantages of an IoT-based air quality monitoring system is its ability to provide real-time data on air quality. This data can be used to alert individuals and organizations to high levels of pollution, allowing them to take appropriate measures to protect their health. Additionally, the system can provide valuable insights into the sources and impact of pollutants, enabling policymakers to develop effective strategies to reduce air pollution.

Without an air quality monitoring system, it can be difficult to accurately measure the level of air pollution in a particular area. This can lead to a lack of awareness and understanding of the severity of the problem, and ultimately result in inadequate measures to address it. Furthermore, the absence of real-time data on air quality can lead to individuals and organizations being exposed to high levels of pollution without their knowledge.

In conclusion, an IoT-based air quality monitoring system is a valuable tool in the fight against air pollution. It provides real-time data on air quality, enabling individuals and organizations to take appropriate measures to protect their health. It also provides valuable insights into the sources and impact of pollutants, enabling policymakers to develop effective strategies to reduce air pollution and improve public health.

CHAPTER 2

DESIGN

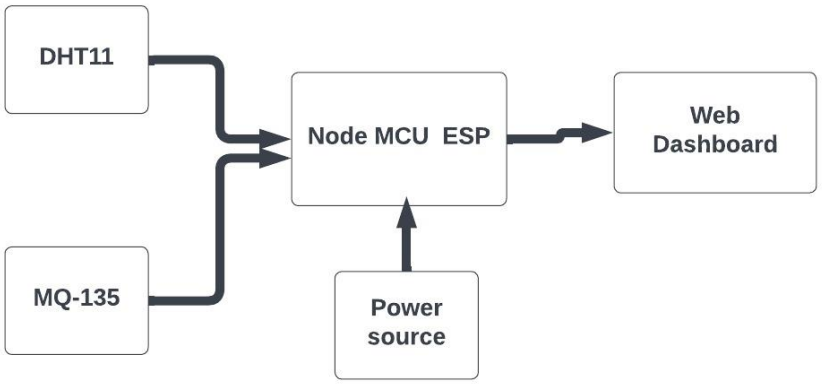

Fig 2.1: Block Diagram of the Model

2.1 DHT Sensor

The DHT sensor is a type of digital temperature and humidity sensor. "DHT" stands for "Digital Humidity and Temperature." It is commonly used in various projects and applications that require monitoring and controlling temperature and humidity levels The DHT sensor is a low-cost and reliable digital temperature and humidity sensor that is commonly used in various projects and applications that require monitoring and controlling temperature and humidity levels.

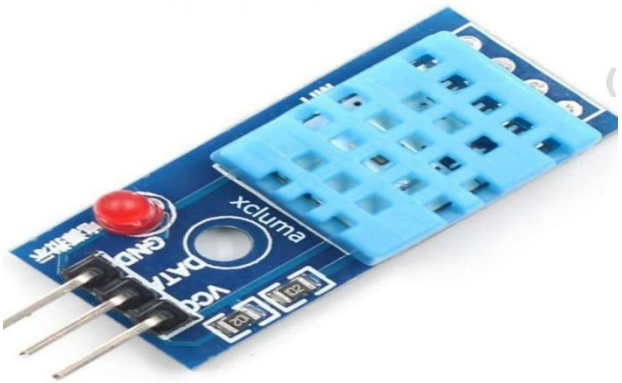

Fig 2.1 DHT Sensor

It uses a capacitive humidity sensor and a thermistor to measure humidity and temperature, respectively. The DHT sensor operates on a DC voltage of 3.3V to 5V and communicates with the microcontroller using a one-wire digital protocol that requires a minimum of three wires (power, ground, and data). The DHT sensor is available in different models such as DHT11 and DHT22, with varying accuracy levels and measurement ranges.

The accuracy of the DHT11 is +/-5%RH for humidity and +/-2°C for temperature, while the DHT22 has an accuracy of +/-2%RH for humidity and +/-0.5°C for temperature. The DHT11 can measure temperature in the range of 0°C to 50°C and humidity in the range of 20%RH to 90%RH, while the DHT22 can measure temperature in the range of -40°C to 125°C and humidity in the range of 0%RH to 100%RH. The DHT sensor is widely used in applications such as weather monitoring, HVAC systems, indoor climate control, agriculture, and environmental monitoring.

2.2 MQ-135 Sensor

The MQ-135 air quality sensor is a type of gas sensor that is commonly used to detect the presence of various gases in the air. It is particularly effective at detecting gases such as ammonia, nitrogen oxides, benzene, smoke, and carbon dioxide.

The sensor works by detecting changes in the electrical conductivity of its internal material when exposed to different gases. It then produces a0n analog output voltage that can be read and processed by a microcontroller or other circuit.

Fig 2.2 MQ-135 Sensor

The MQ-135 sensor is often used in air quality monitoring systems and can be found in devices such as air purifiers, smoke detectors, and gas leak detectors. It is a low-cost sensor and widely available in the market.

However, it is important to note that the accuracy and reliability of the sensor can be affected by various factors such as temperature and humidity, and it may need to be calibrated periodically to ensure accurate readings. Additionally, the sensor is not capable of distinguishing between different types of gases, so it is important to use it in conjunction with other sensors or testing methods for more precise results.

2.3 NodeMCU Board

NodeMCU is an open-source firmware and development kit designed to run on the ESP8266 Wi-Fi module. The ESP8266 is a powerful, low-cost microcontroller chip that can be programmed to connect to the internet, control electronic devices, and interact with sensors. NodeMCU provides a simple programming interface for the ESP8266 that is based on the Lua scripting language, making it easy for beginners to build IoT projects with minimal setup time and effort.

One of the key features of NodeMCU is its built-in Wi-Fi support, which allows developers to connect their projects to the internet without the need for additional hardware. This feature makes NodeMCU an ideal platform for building IoT devices that require internet connectivity, such as smart home devices, environmental sensors, and remote monitoring systems. NodeMCU also provides access to the GPIO pins on the ESP8266, which can be used to control electronic devices and interact with sensors. This feature makes it easy to interface with a wide range of electronic components and build custom hardware projects.

The Lua scripting language used by NodeMCU is easy to learn and use, making it a great choice for beginners who are new to programming. This feature makes it easy to deploy updates and bug fixes to a large number of devices.Overall, NodeMCU is a powerful and versatile platform for building IoT projects with the ESP8266. Its simplicity and ease of use make it a great choice for beginners, while its powerful features and flexibility make it an ideal choice for more advanced projects.

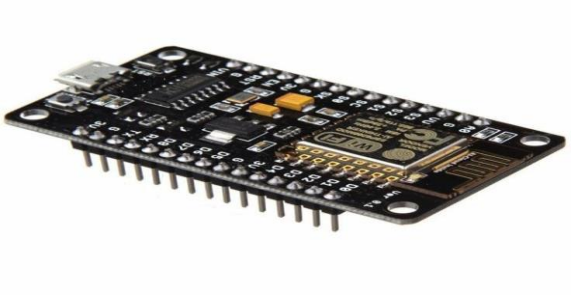

Fig 2.3 NodeMCU Board

2.4 Arduino IDE

The Arduino IDE (Integrated Development Environment) is a software application that is used to write, compile and upload code to Arduino boards. Arduino boards are small microcontroller-based devices that are designed to be easy to use and highly adaptable, and the Arduino IDE is the primary tool used for programming them.

The Arduino IDE is an open-source software application that can be downloaded and used free of charge. It provides a simple and user-friendly interface for writing and editing code, as well as a built-in compiler and uploader that allow you to quickly test and deploy your code to an Arduino board.

The Arduino IDE supports a variety of programming languages, including C and C++. It also includes a number of libraries and tools that make it easy to work with the various hardware components that are commonly used with Arduino boards.

Overall, the Arduino IDE is an essential tool for anyone who is interested in working with Arduino boards and developing projects that make use of microcontrollers and other electronic components.

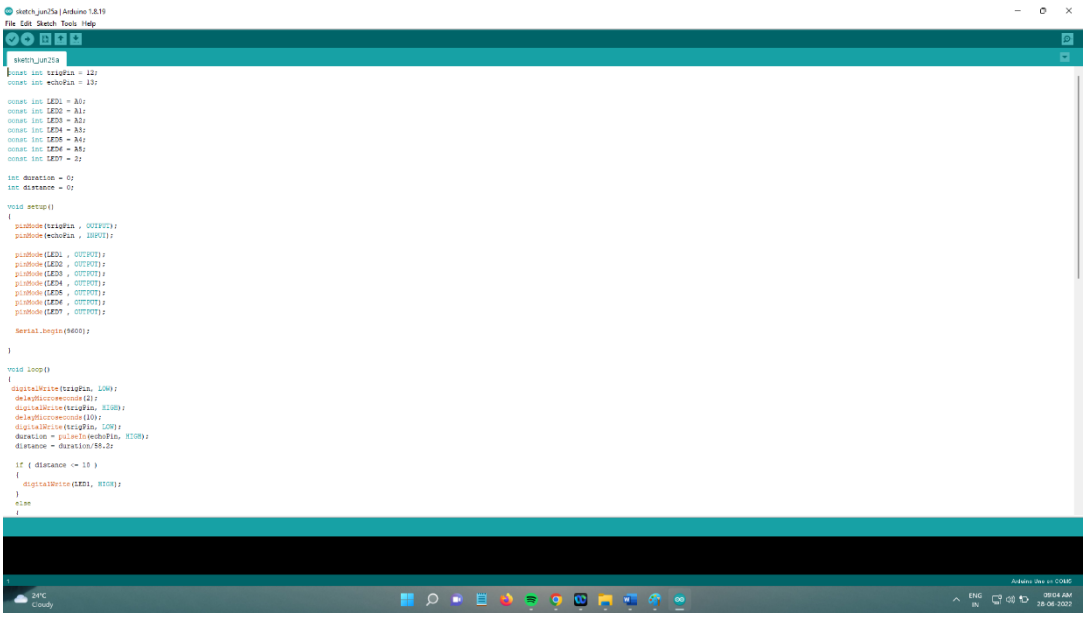

Fig 2.4 Arduino IDE

CHAPTER 3

IMPLEMENTATION

The Air Quality Monitoring System used to measure air quality, with the help of NodeMCU board, MQ-135 gas sensor, DHT-11 sensor and Arduino IDE. Firstly, hardware circuit connection has follows below.

To connect the MQ135 sensor module to the NodeMCU board, you need to connect the VCC pin of the MQ135 module to the 5V pin of the NodeMCU board and the GND pin of the MQ135 module to the GND pin of the NodeMCU board. The AO (analog output) pin of the MQ135 module should be connected to the A0 pin of the NodeMCU board.

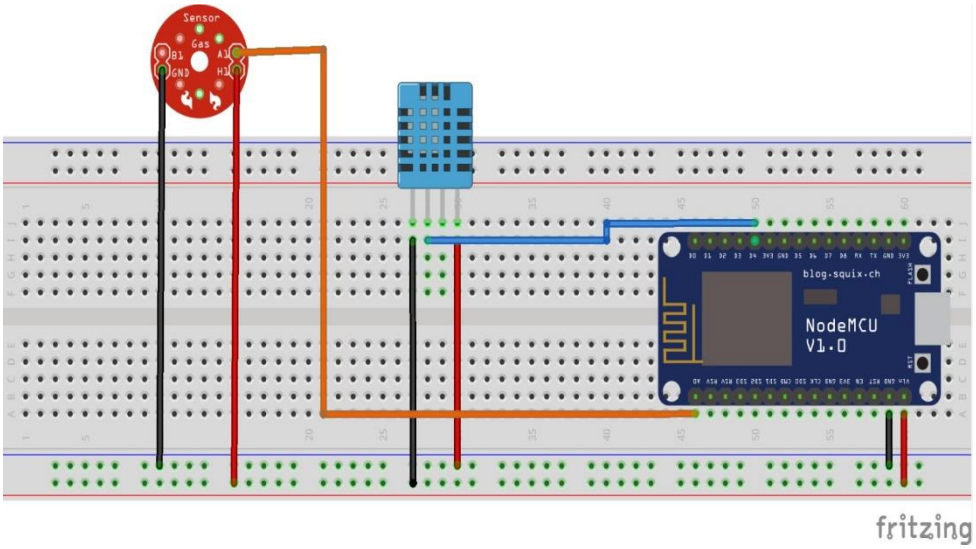

Fig 3.1 Circuit Diagram

To connect the DHT11 sensor module to the NodeMCU board, you need to connect the VCC pin of the DHT11 module to the 3V3 pin of the NodeMCU board and the GND pin of the DHT11 module to the GND pin of the NodeMCU board. The data pin of the DHT11 module should be connected to the D4 pin of the NodeMCU board.

Use a breadboard to connect the different components together. Make sure to use jumper wires to make the necessary connections between the different pins of the components and the NodeMCU board. Once the components are connected, now write the code in Arduino IDE then compile the code after successful compilation, upload the code to the NodeMCU board press RST button on NodeMCU to restart from the beginning and start monitoring the air quality.

PROGRAM:

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#define DHTPIN D4
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

float t = 0.0;
float h = 0.0;

const int aqsensor = A0;
const char *ssid = "Super user";
const char *pass = "qwertyui";

WiFiServer server(80);
void setup() {
  pinMode (aqsensor,INPUT);
  dht.begin();
  Serial.begin (115200);
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected")
 Serial.print("IP Address: ");
  delay(5000);
```

```
  Serial.println(WiFi.localIP());
  delay(5000);
  server.begin();
 }
 void loop() {
  int ppm = analogRead(aqsensor);
  Serial.print("Air Quality: ");
  Serial.println(ppm);
  float newT = dht.readTemperature();
    if (isnan(newT)) {
      Serial.println("Failed to read from DHT sensor!");
    }
    else {
      t = newT;
      Serial.println(t);
    }
    float newH = dht.readHumidity();
    if (isnan(newH)) {
      Serial.println("Failed to read from DHT sensor!");
    }
    else {
      h = newH;
      Serial.println(h);
    }
  delay(1000);
  WiFiClient client = server.available();
  if ( client.connected()) {
  client.println("HTTP/1.1 200 OK");
  client.println("Content-type:text/html");
  client.println("Connection: close");
  client.println("Refresh: 3");
  client.println();
  client.println("<!DOCTYPE html><html>");
```

```
  client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");
  client.println("<link rel=\"icon\" href=\"data:,\">");
  client.println("<body><h1 style=\"color:black;\"> Air Quality Monitoring </h1>");
  client.println("<body><p style=\"color:blue;\"> Pollution Content(in PPM) = " +
String(ppm) +" ppm"+ " </p>");

  if(ppm <= 130)
  {
    client.println("<body><p style=\"color:green;\"> Normal </p>");
    Serial.println("AQ Level Normal");
  }
  else if (ppm > 130 && ppm < 250)
  {
    client.println("<body><p style=\"color:purple;\"> Medium </p>");
    Serial.println("AQ Level Medium");
  }
  else
  {
    client.println("<body><p style=\"font-size:200%; color:red\"> Danger!!! </p>");

    Serial.println("AQ Level Danger!");
  }

  client.println("<body><p style=\"color:blue;\"> Temperature(in °C) = " + String(t) +" °C"+
" </p>");
  client.println("<body><p style=\"color:blue;\"> Temperature(in %) = " + String(h) +" %"+ "
</p>");
  client.println("</body></html>");
  delay(500);
}
}
```

CHAPTER 4

RESULT

Air quality monitoring using NodeMCU, MQ135, and DHT11 is a popular DIY project that allows you to monitor the air quality in your home or office. The project involves using a NodeMCU development board, MQ135 gas sensor, and DHT11 temperature and humidity sensor to measure the concentration of pollutants and temperature and humidity levels in the air. The MQ135 gas sensor is a widely used sensor for detecting pollutants in the air such as smoke, benzene, alcohol, and carbon dioxide. The DHT11 temperature and humidity sensor, on the other hand, is used to measure the temperature and humidity levels in the air. The NodeMCU development board is used as the microcontroller to read data from the sensors and send it to localhost.

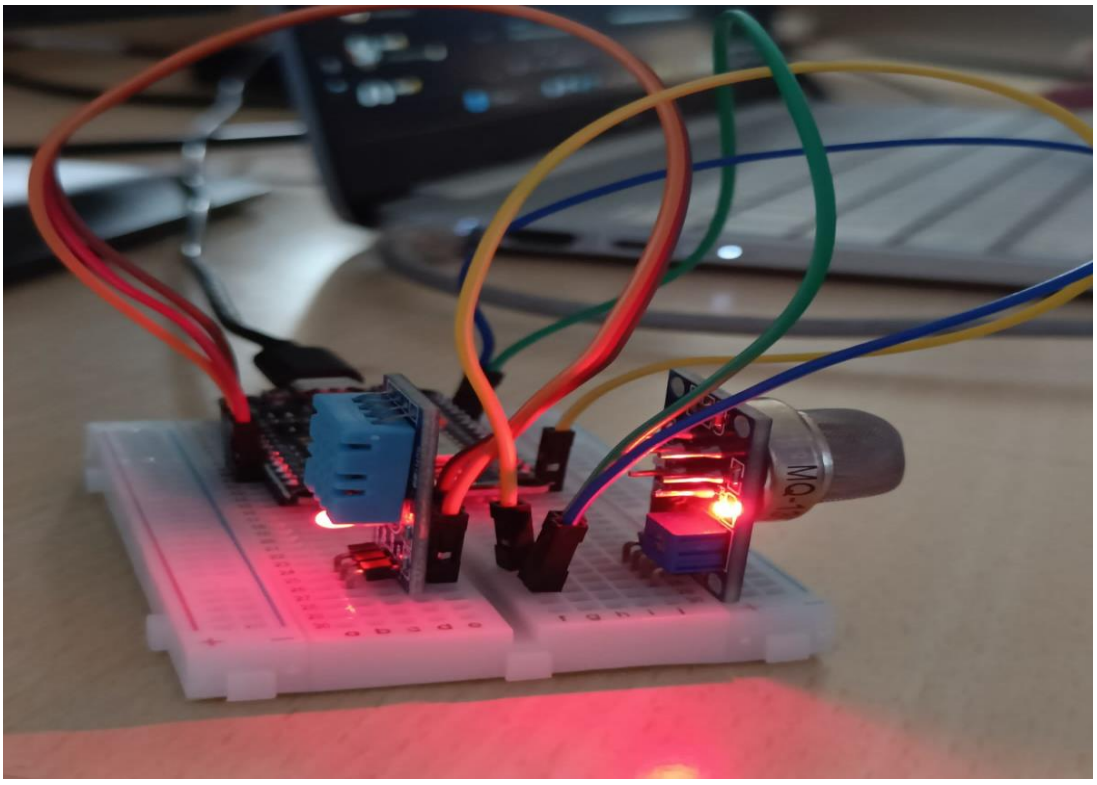

Fig 4.1 Circuit installation

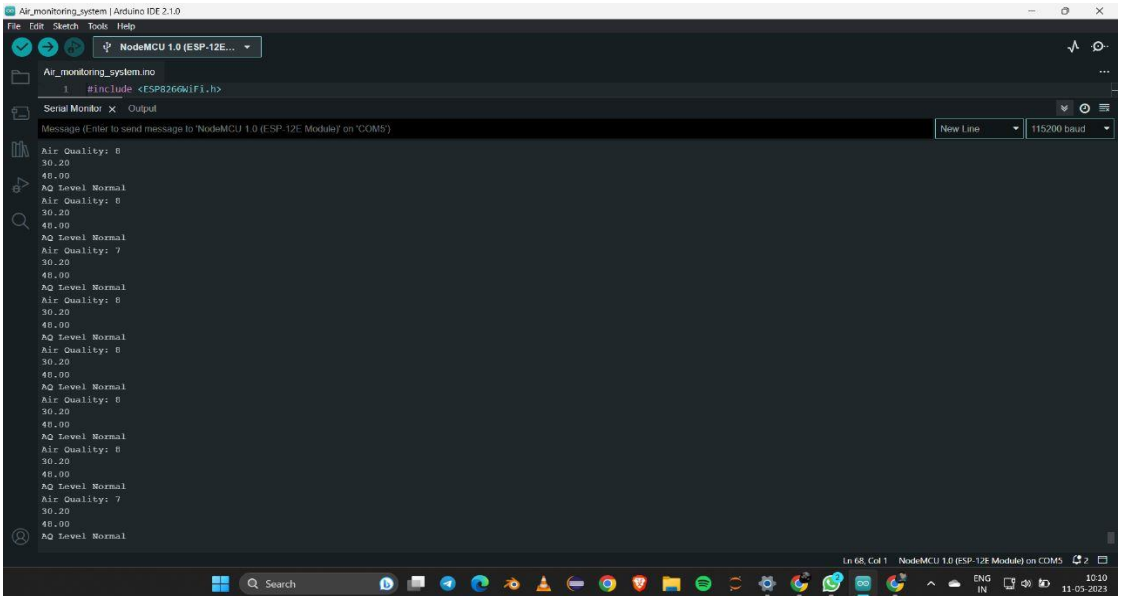

Fig 4.2 Sensor data on Serial Monitor

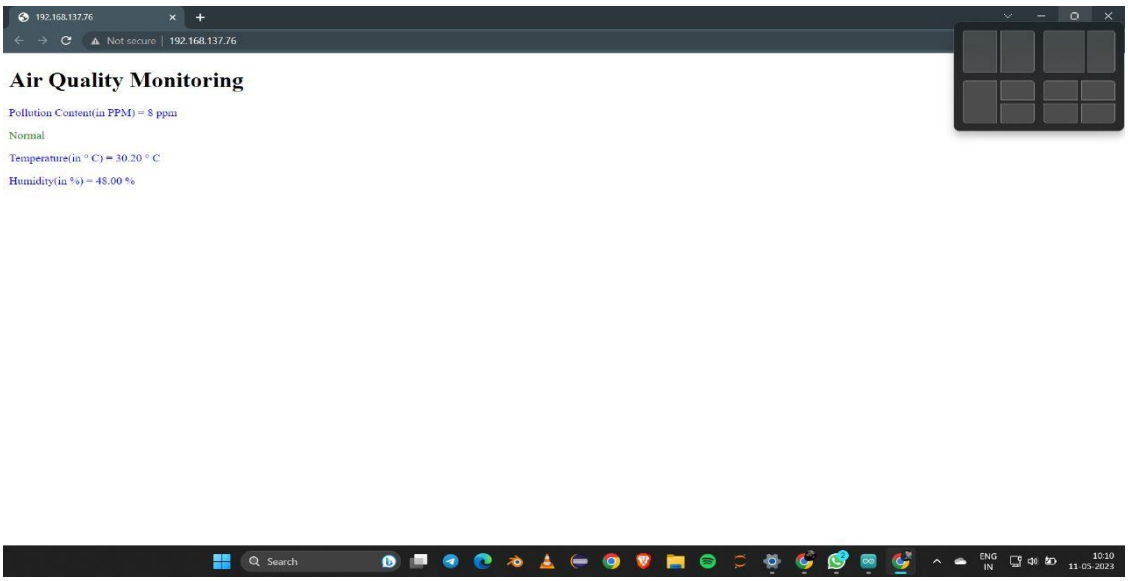

Fig 4.3 Output of air quality