AIR QUALITY MONITORING

INTRODUCTION

Continued exposure to environments with poor air quality is a major public health concern in developed and developing countries. It is estimated that the pollutants responsible for poor air quality cause nearly 2.5 million premature deaths per year world-wide. Significantly, around 1.5 million of these deaths are due to polluted indoor air, and it is suggested that poor indoor air quality may pose a significant health risk to more than half of the world's population.

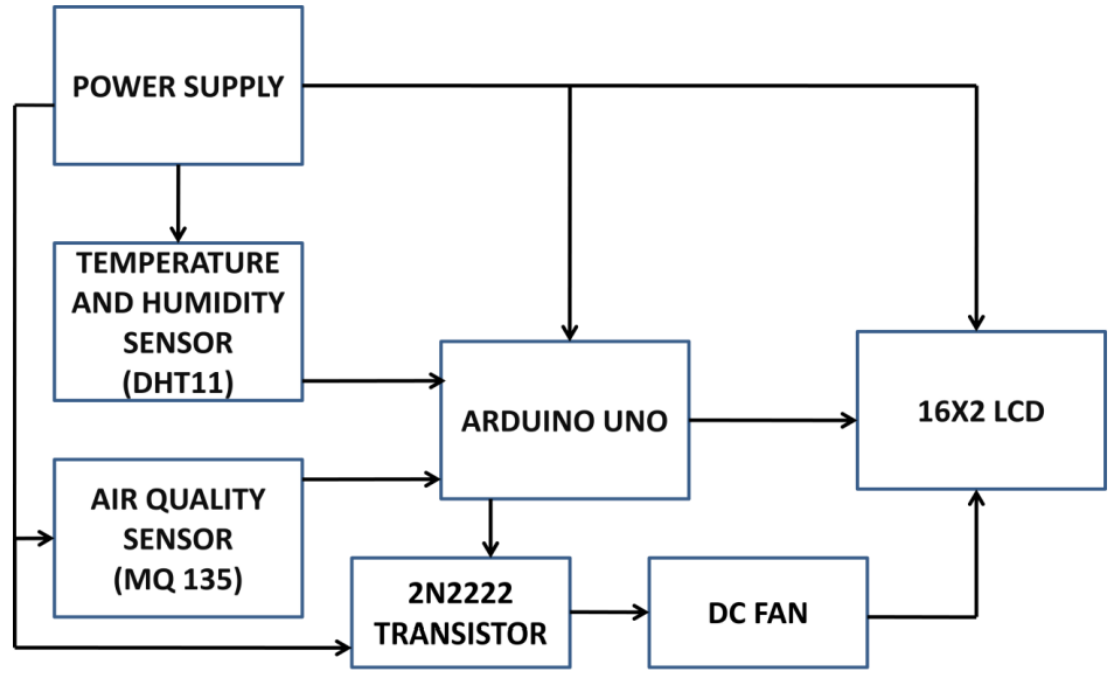
OVERVIEW OF PROJECT

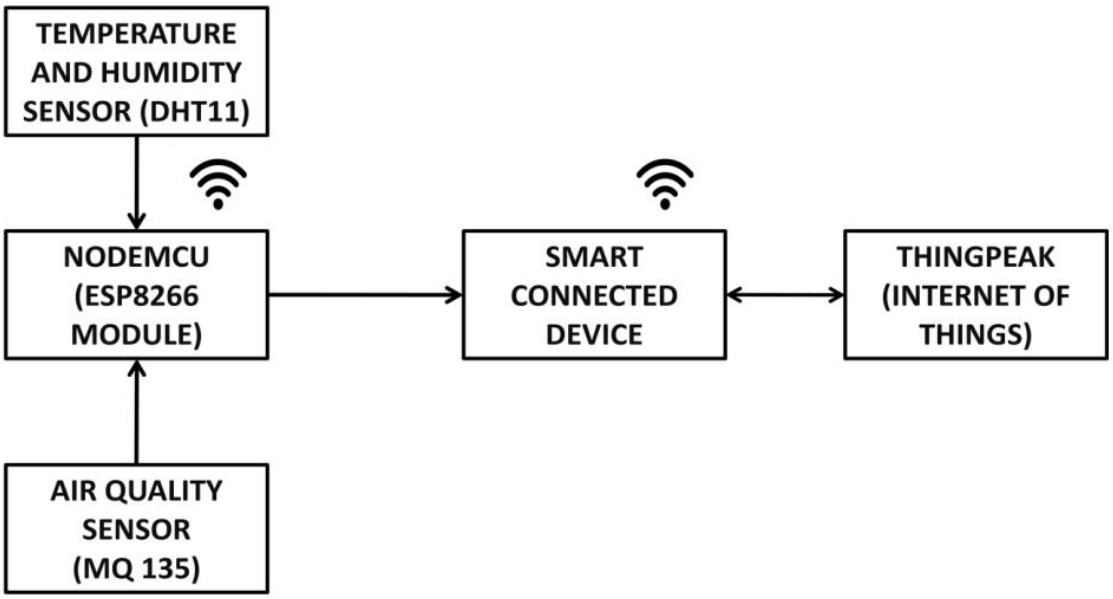
- **project provides a combination of process of sensing several gas levels in the air and also the ambient temperature and humidity, thus sensing the quality of the air.**
- **The levels of the gases and the temperature is displayed in a LCD display panel , which continuously shows the real time output values of the gas sensors , temperature and humidity sensor**

OBJECTIVE

- **To measure and display temperature and humidity level of the environment.**
- **To combine advanced detection technologies to produce an air quality sensing system with advanced capabilities to provide low cost comprehensive monitoring.**
- **To display the sensed data in user friendly format in LCD display panel.**

BLOCK DIAGRAM





HARDWARE REQUIREMENTS

- **For Different Parameter Sensing**

Temperature and Humidity sensor (DHT11)

• Air Quality sensor (MQ 135)

• 2n2222 Transisitor

• DC Fan

• Potentiometer

• 16x2 LCD Panel

• NodeMCU

• Arduino Uno

For Power Supply

Step down transformer (12-0-12 V,1 A)

• Diodes

• Voltage Regulator (7805)

• Capacitors (0.01 micro Farad, 470 micro Farad)

• Wires

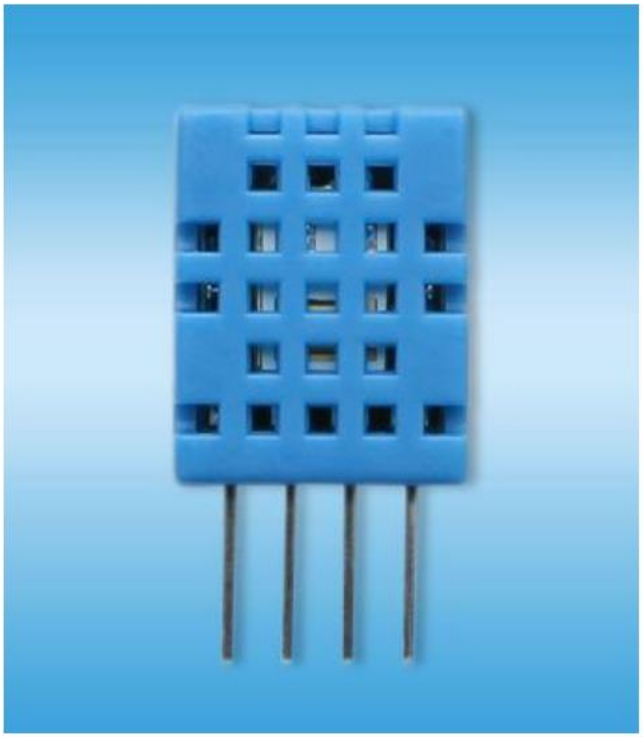
SOFTWARE REQUIREMENTS

- **Arduino (Version 1.8.2)**
- **THINGSPEAK website**

COMPONENT DESCRIPTION

Temperature and humidity sensor (DHT11)

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability.



Pin Description

- **1, the VDD power supply 3.5～5.5V DC**
- **2 DATA serial data, a single bus**
- **3, NC, empty pin**
- **4, GND, used to connect the module to system ground**

Arduino Uno

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world.

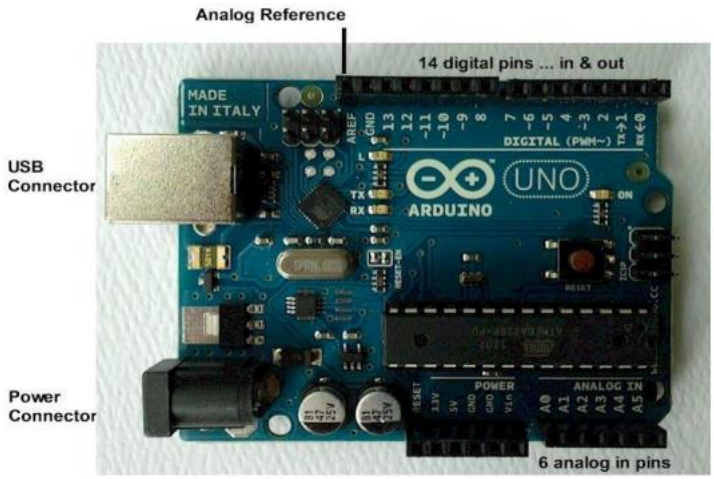


FIG : Arduino Uno

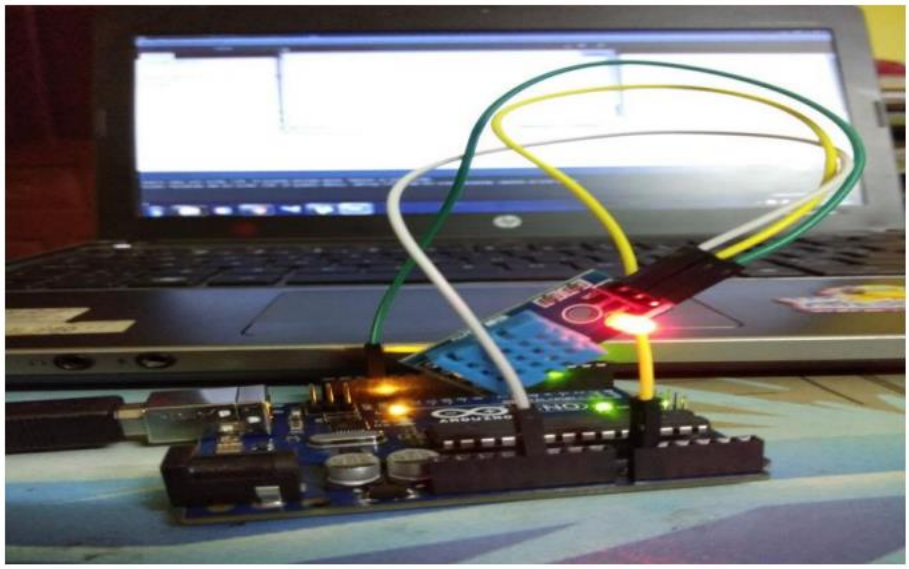
NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 WiFi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif NonOS SDK for ESP8266

WORK DONE

Connection

- DTH11's voltage, ground is connected to +5V and 0V and signal can be connected to any Digital Pin 8 of Arduino Uno.



- MQ135's voltage and ground are connected to +5V and 0V and analog output pin is connected to analog Pin A0 of Arduino Uno.

Coding

```
#include "MQ135.h"

#include <SoftwareSerial.h>

#define DEBUG true

SoftwareSerial esp8266(9,10); // This makes pin 9 of Arduino as RX pin and pin 10
of Arduino as the TX pin

const int sensorPin= 0;

int air_quality;

#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11, 5, 4, 3, 2);

void setup() {

pinMode(8, OUTPUT);

lcd.begin(16,2);
```

```
lcd.setCursor (0,0);

lcd.print ("circuitdigest ");

lcd.setCursor (0,1);

lcd.print ("Sensor Warming ");

delay(1000);

Serial.begin(115200);

esp8266.begin(115200); // your esp's baud rate might be different

  sendData("AT+RST\r\n",2000,DEBUG); // reset module

  sendData("AT+CWMODE=2\r\n",1000,DEBUG); // configure as access point

  sendData("AT+CIFSR\r\n",1000,DEBUG); // get ip address

  sendData("AT+CIPMUair_quality=1\r\n",1000,DEBUG); // configure for multiple conn
ections

  sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // turn on server on port 80

pinMode(sensorPin, INPUT);        //Gas sensor will be an input to the arduino

lcd.clear();

}

void loop() {

MQ135 gasSensor = MQ135(A0);

float air_quality = gasSensor.getPPM();

if(esp8266.available()) // check if the esp is sending a message

  {

    if(esp8266.find("+IPD,"))

    {

     delay(1000);

     int connectionId = esp8266.read()-48; /* We are subtracting 48 from the outpu
t because the read() function returns the ASCII decimal value and the first decima
l number which is 0 starts at 48*/

     String webpage = "<h1>IOT Air Pollution Monitoring System</h1>";

       webpage += "<p><h2>";

       webpage+= " Air Quality is ";

       webpage+= air_quality;
```

```
       webpage+=" PPM";
       webpage += "<p>";
     if (air_quality<=1000)
{
  webpage+= "Fresh Air";
}
else if(air_quality<=2000 && air_quality>=1000)
{
  webpage+= "Poor Air";
}
else if (air_quality>=2000 )
{
webpage+= "Danger! Move to Fresh Air";
}
webpage += "</h2></p></body>";
     String cipSend = "AT+CIPSEND=";
     cipSend += connectionId;
     cipSend += ",";
     cipSend +=webpage.length();
     cipSend +="\r\n";

     sendData(cipSend,1000,DEBUG);
     sendData(webpage,1000,DEBUG);

     cipSend = "AT+CIPSEND=";
     cipSend += connectionId;
     cipSend += ",";
     cipSend +=webpage.length();
     cipSend +="\r\n";
```

```
     String closeCommand = "AT+CIPCLOSE=";
     closeCommand+=connectionId; // append connection id
     closeCommand+="\r\n";

     sendData(closeCommand,3000,DEBUG);
    }
  }
lcd.setCursor (0, 0);
lcd.print ("Air Quality is ");
lcd.print (air_quality);
lcd.print (" PPM ");
lcd.setCursor (0,1);
if (air_quality<=1000)
{
lcd.print("Fresh Air");
digitalWrite(8, LOW);
}
else if( air_quality>=1000 && air_quality<=2000 )
{
lcd.print("Poor Air, Open Windows");
digitalWrite(8, HIGH );
}
else if (air_quality>=2000 )
{
lcd.print("Danger! Move to Fresh Air");
digitalWrite(8, HIGH);   // turn the LED on
}
lcd.scrollDisplayLeft();
```

```
delay(1000);
}
String sendData(String command, const int timeout, boolean debug)
{
    String response = "";
    esp8266.print(command); // send the read character to the esp8266
    long int time = millis();
    while( (time+timeout) > millis())
    {
      while(esp8266.available())
      {
        // The esp has data so display its output to the serial window
        char c = esp8266.read(); // read the next character.
        response+=c;
      }
    }
    if(debug)
    {
      Serial.print(response);
    }
    return response;
}
```

CONCLUSION

IoT-based air pollution monitoring system is a revolutionary solution that can provide accurate and real-time data about the air quality in a particular area