**Phase 5 project**

**Project Title: AIR QUALITY MONITORING**

**Project ID:** proj\_223737\_Team\_3

**College:** Gnanamani College of Technology

**College code:** 6208

**Branch:** B.Tech-Information Technology

**Year:** IIIrd year

**Team Members:**

Balachandar.K(620821205006)

Jana.V(620821205020)

Kathirvanan.S(620821205028)

Pravinkumar.K(620821205041)

Srikanth.S(620821205056)

**PHASE 5**

**Air Quality Monitoring Project Documentation**

**Project Objectives:**

The objective of this project is to develop a real-time air quality monitoring system using IoT devices. This system will provide continuous monitoring and data collection of air quality parameters, such as particulate matter (PM2.5 and PM10), carbon monoxide (CO), nitrogen dioxide (NO2), ozone (O3), and temperature, at various locations within a target area. The project aims to raise public awareness about air quality and its health impacts by making the collected data accessible through a user-friendly online platform.

**IoT Device Setup:**

The IoT device setup consists of the following components:

* **Air Quality Sensors**: We use high-precision sensors to measure MQ135 levels in the air.
* **Microcontroller**: Each device is equipped with a microcontroller Arduino to collect data from the sensors.

**Required Components:**

* MQ135 Gas sensor
* Arduino Uno
* Wi-Fi module ESP8266
* 16X2 LCD
* Breadboard
* 10K potentiometer
* 1K ohm resistors
* 220 ohm resistor
* Buzzer

**Air Quality Sensor (MQ135):**

**Product Description:**

Air quality click is suitable for detecting ammonia (NH3), nitrogen oxides (NOx) benzene, smoke, CO2 and other harmful or poisonous gases that impact air quality. The MQ-135 sensor unit has a sensor layer made of tin dioxide (SnO2), an inorganic compound which has lower conductivity in clean air than when polluting gases are present. To calibrate Air quality, use the on-board potentiometer to adjust the load resistance on the sensor circuit.



**Fig:** MQ135 Sensor

**Pin Description:**

* 1, the VDD power supply 5V DC
* 2,GND , used to connect the module to system ground
* 3, DIGITAL OUT, You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer
* 4, ANALOG OUT, This pin outputs 0-5V analog voltage based on the intensity of the gas.

**Potentiometer:**

**Product Description:**

A potentiometer is a three[-terminal](https://en.wikipedia.org/wiki/Terminal_%28electronics%29) [resistor](https://en.wikipedia.org/wiki/Resistor) with a sliding or rotating contact that forms an adjustable [voltage divider.](https://en.wikipedia.org/wiki/Voltage_divider) If only two terminals are used, one end and the wiper, it acts as a variable resistor or [rheostat.](https://en.wikipedia.org/wiki/Potentiometer#Rheostat) The measuring instrument called a [potentiometer](https://en.wikipedia.org/wiki/Potentiometer_%28measuring_instrument%29) is essentially a [voltage divider](https://en.wikipedia.org/wiki/Voltage_divider) used for measuring [electric potential](https://en.wikipedia.org/wiki/Electric_potential) (voltage); the component is an implementation of the same principle, hence its name.



**Fig:** Potentiometer

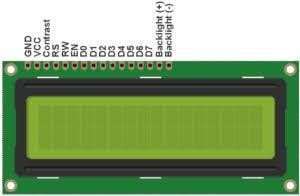
**16X2 LCD Panel:**

**Product Description:**

A liquid-crystal display (LCD) is a [flat-panel display](https://en.wikipedia.org/wiki/Flat_panel_display) or other [electronically modulated optical device](https://en.wikipedia.org/wiki/Electro-optic_modulator) that uses the light-modulating properties of [liquid crystals.](https://en.wikipedia.org/wiki/Liquid_crystal) Liquid crystals do not emit light directly, instead using a [backlight](https://en.wikipedia.org/wiki/Backlight) or [reflector](https://en.wikipedia.org/wiki/Reflector_%28photography%29) to produce images in color or [monochrome](https://en.wikipedia.org/wiki/Monochrome)[.[1]](https://en.wikipedia.org/wiki/Liquid-crystal_display#cite_note-1) LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and [seven-segment displays.](https://en.wikipedia.org/wiki/Seven-segment_display)

**Pin Description:**

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Ground (0V) | Ground |
| 2 | Supply voltage; 5V (4.7V – 5.3V) | Vcc |
| 3 | Contrast adjustment; through a variable resistor | VEE |
| 4 | Selects command register when low; and data register when high | Register Select |
| 5 | Low to write to the register; High to read from the register | Read/write |
| 6 | Sends data to data pins when a high to low pulse is given | Enable |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |
| 13 | DB6 |
| 14 | DB7 |
| 15 | Backlight VCC (5V) | Led+ |
| 16 | Backlight Ground (0V) | Led- |



**Fig:** 16X2 LCD Display

**Buzzer:**

Buzzer is also called as Beeper. It is a sound signalling mechanical device.

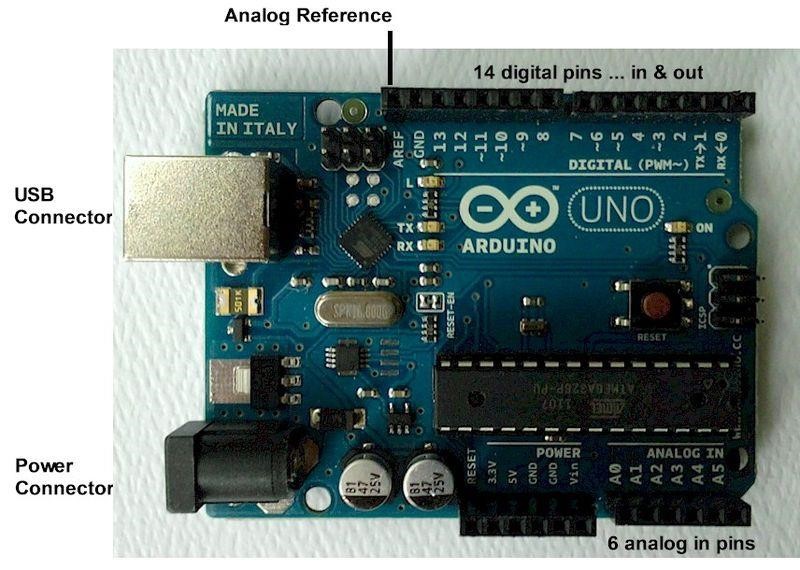


**Fig:** Buzzer

**Arduino UNO:**

**Product Description:**

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures [single-board microcontrollers](https://en.wikipedia.org/wiki/Single-board_microcontroller) and [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) kits for building digital devices and interactive objects that can sense and control objects in the physical world.



**Fig:** Arduino UNO

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various expansion boards (*shields*) and other circuits. The boards feature serial communications interfaces, including [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages [C](https://en.wikipedia.org/wiki/C_%28programming_language%29) and [C++.](https://en.wikipedia.org/wiki/C%2B%2B) In addition to using traditional compiler tool chains, the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on the [Processing](https://en.wikipedia.org/wiki/Processing_%28programming_language%29) language project.

**Pin Description:**

|  |  |  |
| --- | --- | --- |
| **Pin Category** | **Pin Name** | **Details** |
| Power | Vin, 3.3V, 5V,  GND | Vin: Input voltage to Arduino when using an external power source.  5V: Regulated power supply used to power microcontroller and other components on the board.  3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA.  GND: ground pins. |
| Reset | Reset | Resets the microcontroller. |
| Analog Pins | A0 – A5 | Used to provide analog input in the range of 0-5V |
| Input/Output Pins | Digital Pins 0 - 13 | Can be used as input or output pins. |
| Serial | 0(Rx), 1(Tx) | Used to receive and transmit TTL serial data. |
| External Interrupts | 2, 3 | To trigger an interrupt. |
| PWM | 3, 5, 6, 9, 11 | Provides 8-bit PWM output. |
| SPI | 10 (SS), 11 (MOSI),  12 (MISO) and 13 (SCK) | Used for SPI communication. |
| Inbuilt LED | 13 | To turn on the inbuilt LED. |
| TWI | A4 (SDA), A5 (SCA) | Used for TWI communication. |
| AREF | AREF | To provide reference voltage for input voltage. |

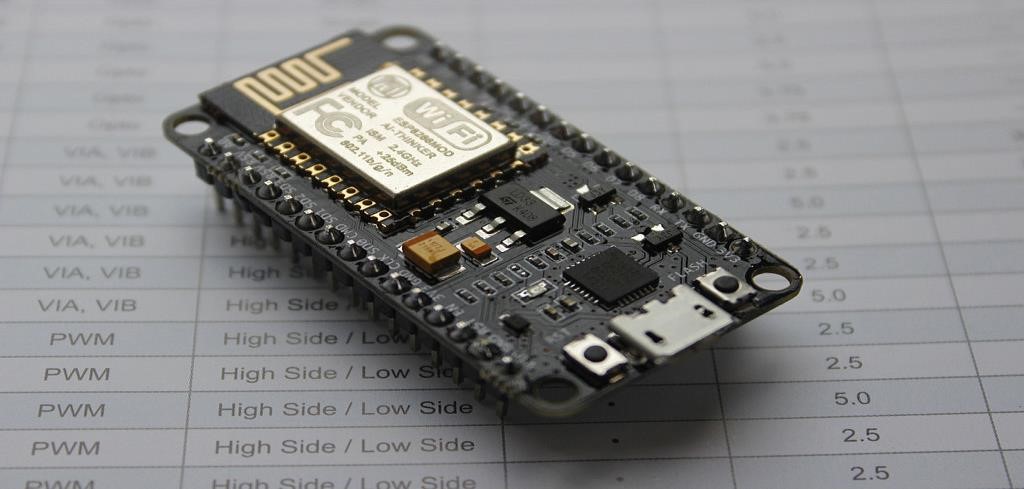
**Technical Specification:**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet) – 8 bit AVR family microcontroller |
| Operating Voltage | 5V |
| Recommended Input Voltage | 7-12V |
| Input Voltage Limits | 6-20V |
| Analog Input Pins | 6 (A0 – A5) |
| Digital I/O Pins | 14 (Out of which 6 provide PWM output) |
| DC Current on I/O Pins | 40 mA |
| DC Current on 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (0.5 KB is used for Bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

**NodeMCU:**

**Product Description:**

NodeMCU is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform. It includes [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [WiFi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from [Espressif Systems,](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1) 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](https://en.wikipedia.org/wiki/Lua_%28programming_language%29) scripting language. It is based on the eLua project, and built on the Espressif NonOS SDK for ESP8266.



**Fig:** NodeMCU

**Software Requirement:**

Here in this project we are using Arduino IDE it is a open source IDE developed by Arduino.cc it supports c++ embedded language. C++ is platform dependent language. Any language is said to be platform dependent whenever the program is being execute in the same operating system where that was developed and compiled but it don't execute on other operating system.

The Arduino IDE is is supported by all operating systems (like Windows, mac OS, Linux)it is developed with java programming.

It originated from the IDE for the languages for interfacing with the devices and that includes a code editing with features such as text cutting and pasting, searching and replacing text, automatic indexing, braces matching, and syntax highlighting, and that provides a simple one click mechanisms for compiling and uploading programs to an arduino board. It also contains a message area, a text console, a toolbar with buttons and many common features which os required for an IDE.

Pollution Monitoring and Notification System Requirements Specification It is an open-source Arduino Software which helps in writing code and uploading it to the Arduino board. It supports all the operating systems. This software works with any of the a aforementioned Arduino boards. The programming language used to write codes is C and C++. The program written or the code written in IDE is known as sketch. Once the sketch is ready, it is compiled. And the code will be uploaded to arduino board kit.



**Fig:** Arduino IDE Software.

**Circuit Diagram and Explanation:**

First of all we will connect the ESP8266 with the Arduino. ESP8266 runs on 3.3V and if you will give it 5V from the Arduino then it won’t work properly and it may get damage. Connect the VCC and the CH\_PD to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting three resistors in series like we did in the circuit. Connect the TX pin of the ESP8266 to the pin 10 of the Arduino and the RX pin of the esp8266 to the pin 9 of Arduino through the resistors.

ESP8266 Wi-Fi module gives your projects access to Wi-Fi or internet. It is a very cheap device and make your projects very powerful. It can communicate with any microcontroller and it is the most leading devices in the [IOT platform](http://circuitdigest.com/internet-of-things-iot-projects). Learn more about [using ESP8266 with Arduino here](http://circuitdigest.com/microcontroller-projects/sending-arduino-data-to-webpage).

Then we will connect the MQ135 sensor with the Arduino. Connect the VCC and the ground pin of the sensor to the 5V and ground of the Arduino and the Analog pin of sensor to the A0 of the Arduino.

Connect a buzzer to the pin 8 of the Arduino which will start to beep when the condition becomes true.

In last, we will [connect LCD with the Arduino](http://circuitdigest.com/microcontroller-projects/arduino-lcd-interfacing-tutorial). The connections of the LCD are as follows

* Connect pin 1 (VEE) to the ground.
* Connect pin 2 (VDD or VCC) to the 5V.
* Connect pin 3 (V0) to the middle pin of the 10K potentiometer and connect the other two ends of the potentiometer to the VCC and the GND. The potentiometer is used to control the screen contrast of the LCD. Potentiometer of values other than 10K will work too.
* Connect pin 4 (RS) to the pin 12 of the Arduino.
* Connect pin 5 (Read/Write) to the ground of Arduino. This pin is not often used so we will connect it to the ground.
* Connect pin 6 (E) to the pin 11 of the Arduino. The RS and E pin are the control pins which are used to send data and characters.
* The following four pins are data pins which are used to communicate with the Arduino.

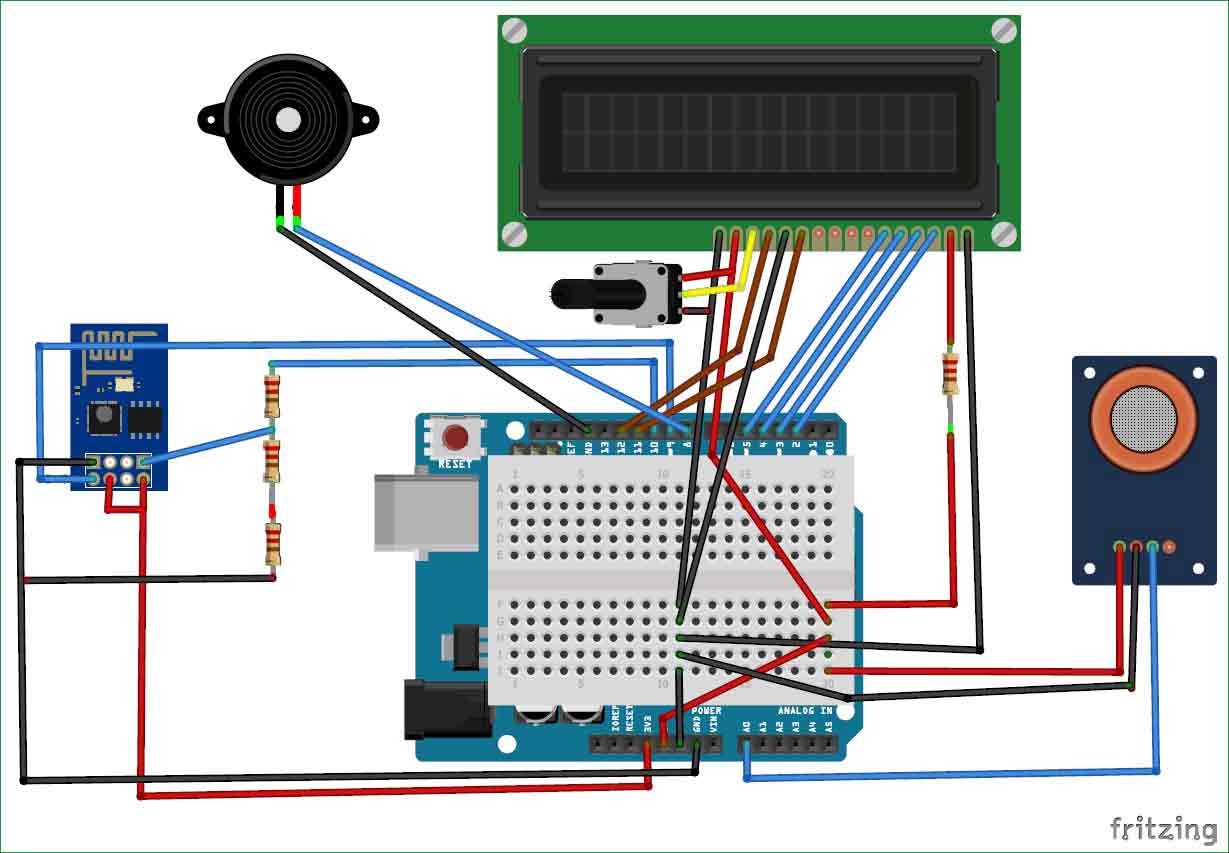
Connect pin 11 (D4) to pin 5 of Arduino.

Connect pin 12 (D5) to pin 4 of Arduino.

Connect pin 13 (D6) to pin 3 of Arduino.

Connect pin 14 (D7) to pin 2 of Arduino.

* Connect pin 15 to the VCC through the 220 ohm resistor. The resistor will be used to set the back light brightness. Larger values will make the back light much more darker.
* Connect pin 16 to the Ground.



**Working Explanation:**

The MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and some other gases, so it is perfect gas sensor for our **Air Quality Monitoring Project**. When we will connect it to Arduino then it will sense the gases, and we will get the Pollution level in PPM (parts per million). MQ135 gas sensor gives the output in form of voltage levels and we need to convert it into PPM. So for converting the output in PPM, here we have used a library for MQ135 sensor, it is explained in detail in “Code Explanation” section below.

Sensor was giving us value of 90 when there was no gas near it and the safe level of air quality is 350 PPM and it should not exceed 1000 PPM. When it exceeds the limit of 1000 PPM, then it starts cause Headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 2000 PPM then it can cause increased heart rate and many other diseases.

When the value will be less than 1000 PPM, then the LCD and webpage will display “Fresh Air”.  Whenever the value will increase 1000 PPM, then the buzzer will start beeping and the LCD and webpage will display “Poor Air, Open Windows”. If it will increase 2000 then the buzzer will keep beeping and the LCD and webpage will display “Danger! Move to fresh Air”.

**Code Explanation:**

Before beginning the coding for this project, we need to first Calibrate the MQ135 Gas sensor. There are lots of calculations involved in converting the output of sensor into PPM value, we have done this calculation before in our previous Smoke Detector project. But here we are using the Library for MQ135, you can download and install this MQ135 library from here:

Using this library you can directly get the PPM(parts per million) values, by just using the below two lines:

MQ135 gasSensor = MQ135(A0);

float air\_quality = gasSensor.getPPM();

But before that we need to **calibrate the MQ135 sensor**, for calibrating the sensor upload the below given code and let it run for 12 to 24 hours and then get the RZERO value.

#include "MQ135.h"

void setup (){

Serial.begin (9600);

}

void loop() {

MQ135 gasSensor = MQ135(A0); // Attach sensor to pin A0

float rzero = gasSensor.getRZero();

Serial.println (rzero);

delay(1000);

}

After getting the RZERO value. **Put the RZERO value in the library file** you downloaded "MQ135.h": #define RZERO 494.63

Now we can begin the actual code for our Air quality monitoring project.

In the code, first of all we have defined the libraries and the variables for the Gas sensor and the LCD. By using the Software Serial Library, we can make any digital pin as TX and RX pin. In this code, we have made Pin 9 as the RX pin and the pin 10 as the TX pin for the ESP8266. Then we have included the library for the LCD and have defined the pins for the same. We have also defined two more variables: one for the sensor analog pin and other for storing air\_quality value.

#include <SoftwareSerial.h>

#define DEBUG true

SoftwareSerial esp8266(9,10);

#include <LiquidCrystal.h>

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

const int sensorPin= 0;

int air\_quality;

Then we will declare the pin 8 as the output pin where we have connected the buzzer. lcd.begin(16,2) command will start the LCD to receive data and then we will set the cursor to first line and will print the ‘circuitdigest’. Then we will set the cursor on the second line and will print ‘Sensor Warming’.

pinMode(8, OUTPUT);

lcd.begin(16,2);

lcd.setCursor (0,0);

lcd.print ("circuitdigest ");

lcd.setCursor (0,1);

lcd.print ("Sensor Warming ");

delay(1000);

Then we will set the baud rate for the serial communication. Different ESP’s have different baud rates so write it according to your ESP’s baud rate. Then we will send the commands to set the ESP to communicate with the Arduino and show the IP address on the serial monitor.

Serial.begin(115200);

esp8266.begin(115200);

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

sendData("AT+CWMODE=2\r\n",1000,DEBUG);

sendData("AT+CIFSR\r\n",1000,DEBUG);

sendData("AT+CIPMUair\_quality=1\r\n",1000,DEBUG);

sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG);

pinMode(sensorPin, INPUT);

lcd.clear();

For printing the output on the webpage in web browser, we will have to use **HTML programming**. So, we have created a string named webpage and stored the output in it. We are subtracting 48 from the output because the read() function returns the ASCII decimal value and the first decimal number which is 0 starts at 48.

if(esp8266.available())

{

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

{

delay(1000);

int connectionId = esp8266.read()-48;

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

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

webpage+= " Air Quality is ";

webpage+= air\_quality;

webpage+=" PPM";

webpage += "<p>";

The following code will call a function named sendData and will send the data & message strings to the webpage to show.

sendData(cipSend,1000,DEBUG);

sendData(webpage,1000,DEBUG);

cipSend = "AT+CIPSEND=";

cipSend += connectionId;

cipSend += ",";

cipSend +=webpage.length();

cipSend +="\r\n";

The following code will print the data on the LCD. We have applied various conditions for checking air quality, and LCD will print the messages according to conditions and buzzer will also beep if the pollution goes beyond 1000 PPM.

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);

Finally the below function will send and show the data on the webpage. The data we stored in string named ‘webpage’ will be saved in string named ‘command’. The ESP will then read the character one by one from the ‘command’ and will print it on the webpage.

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);

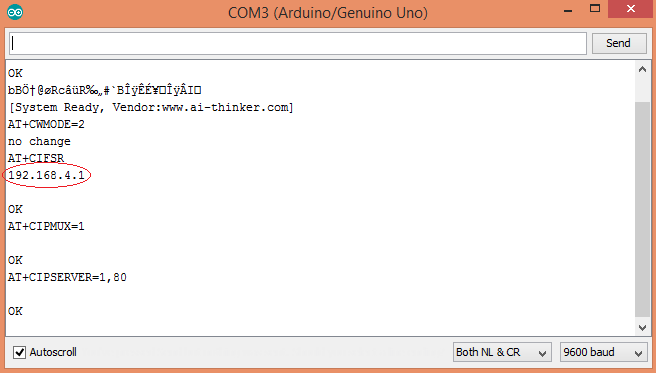
}

return response;

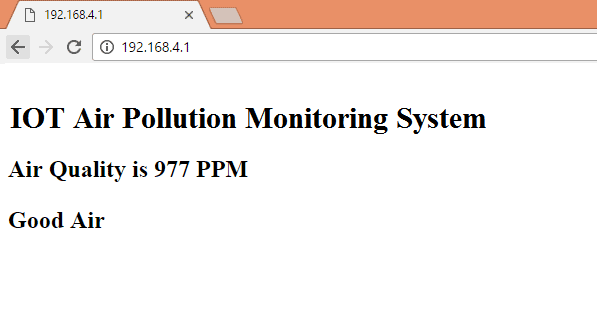
}

**Testing and Output of the Project:**

Before uploading the code, make sure that you are connected to the Wi-Fi of your ESP8266 device. After uploading, open the serial monitor and it will show the IP address like shown below.



Type this IP address in your browser, it will show you the output as shown below. You will have to refresh the page again if you want to see the current Air Quality Value in PPM.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

We have setup a local server to demonstrate its working, you can check the **Video** below. But to monitor the air quality from anywhere in the world, you need to **forward the port 80 (used for HTTP or internet) to your local or private IP address** (192.168\*) of you device. After port forwarding all the incoming connections will be forwarded to this local address and you can open above shown webpage by just entering the public IP address of your internet from anywhere. You can forward the port by logging into your router (192.168.1.1) and find the option to setup the port forwarding.

**Data Sharing and Public Awareness:**

The real-time air quality monitoring system has several benefits for raising public awareness:

1. **Immediate Information**: By providing real-time air quality data, individuals can make informed decisions about outdoor activities and exposure to pollutants.
2. **Health Education**: The platform educates users about the health risks associated with poor air quality, such as respiratory diseases and allergies.
3. **Community Engagement**: Users can report air quality issues, encouraging community engagement and collaboration to address pollution sources.
4. **Policy Influence**: Researchers and policymakers can access this data to formulate effective air quality management policies.

**Advantages:**

* Sensors are easily available.
* Simple, compact, easy to handle.
* Sensors have long life and less cost.
* Quality of air can be checked indoor as well as outdoor.
* Detecting a wide range of physical parameters including temperature ,humidity and carbon dioxide.

**Application:**

* Indoor air quality monitoring.
* Industrial perimeter monitoring.
* Roadside pollution monitoring.
* To make this data available to common man.

**Conclusion:**

The system to watch the air of environment using Arduino microcontroller, IOT Technology is proposed to enhance quality of air. With the utilization of IOT technology enhances the method of monitoring various aspects of environment like air quality monitoring issue proposed during this paper. Here the using of MQ135 gas sensor gives the sense of various sort of dangerous gas and Arduino is that the heart of this project. We implemented our system on the Arduino system monitor and the 16x2 LCD display by using the MQ135 sensor and the system displays the values that it received from the MQ135 sensor and will show the corresponding air quality rating as per the PPM value.