A Mini-Project Report

on

Air Quality Monitoring System Using IoT

Submitted for partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

Mr. Abdul Basith (2451-17-733-131)

Mr. Batal Al Farooqui (2451-17-733-145)

Mr. Mohammed Faiyaz Ahmed (2451-17-733-141)

Under the guidance of

G. Vijay Kumar

Associate Professor

Department of CSE

M.V.S.R.E.C., Hyderabad.

Department of Computer Science and Engineering

M.V.S.R. ENGINEERING COLLEGE

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510

2018-19.

M.V.S.R. ENGINEERING COLLEGE

(Affiliated to Osmania University, Hyderabad)

Nadergul (P.O.), Hyderabad-501510

Certificate

This is to certify that the mini-project work entitled **“Air Quality Monitoring System Using IoT”** is a bonafide work carried out by **Mr. Mohammed Faiyaz Ahmed(2451-17-733-141), Mr. Abdul Basith(2451-17-733-131) and Mr. Batal Al Farooqui(2451-17-733-145)**
in partial fulfillment of the requirements for the award of degree of **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING** from M.V.S.R. Engineering College, affiliated to OSMANIA UNIVERSITY, Hyderabad, under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide	Project Co-ordinator
G. Vijay Kumar	**D. Sirisha**
Associate Professor	Assistant Professor
Department of CSE	Department of CSE

 MVSREC, Hyderabad.

DECLARATION

This is to certify that the work reported in the present mini-project entitled "**Air Quality Monitoring System Using IOT"** is a record of bonafide work done by us in the Department of Computer Science and Engineering, M.V.S.R Engineering College, Osmania University. The reports are based on the mini-project work done entirely by us and not copied from any other source.

The results embodied in this mini-project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Abdul Basith	Batal Al Farooqui	Mohammed Faiyaz Ahmed
(2451-17-733-131)	(2451-17-733-145)	(2451-17-733-141)

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude and indebtedness to my mini-project guide **G. Vijay Kumar** for his valuable suggestions and interest throughout the course of this mini-project.

We are also thankful to our principal **Dr. G. Kanaka Durga** and **Dr. Akhil Khare**, Professor and Head, Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad for providing excellent infrastructure for completing this mini-project successfully as a part of our B.E. Degree (CSE). We would like to thank our mini-project coordinators **D. Sirisha and B Janaiah** for their constant monitoring, guidance and support.

We convey our heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank my family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

Abdul Basith (2451-17-733-131)

Batal Al Farooqui (2451-17-733-145)

Mohammed Faiyaz Ahmed (2451-17-733-141)

Abstract

Air pollution in India is a serious problem with major sources being fuel,wood and biomass burning, fuel adulteration, vehicle emission and traffic congestion.The system that we designed is used to detect the harmful air pollutants and alert the public belonging to areas having dedicated hardware by the buzzer sound or through the website when the air quality goes beyond a certain level.The air quality detector measures the level of CO_2 along with other gases.This project consists of one sensor namely, MQ135 Gas Sensor as it can detect most harmful gases in an accurate manner.The project also has an LCD attached to it which displays the value of CO_2 level measured along with other gases. The project shows the air quality in PPM on the LCD.

TABLE OF CONTENTS

LIST OF FIGURES

A list of figures with figure number, figure title and page number and a list of tables with table number, table name and page number.

LIST OF TABLES

CHAPTER I

1. INTRODUCTION

The reason behind choosing this project is that Air pollution is a serious issue in India. Growth in population, rapid growth in Urbanization and Industrialization, rising demands for energy and motor vehicles are worsening air pollution. Pollutant air contains one or more hazardous substrate, pollutants that create hazards to general health. About 1.2 million premature deaths per year happen in India. First 7 out of 10 and 14 out of 20 most polluted cities in the world are from India. According to 2018 World Health Organization (WHO) Report, first 14 of 15 cities with most PM 2.5 concentrations are from India, with Kanpur being 1st worldwide. Effect of Human health: 90% of the diseases due to air pollution are non-infectious, of which 24% (Heart diseases), 25% (heart stroke), 29% (lung cancer) and 43% (chronic pulmonary diseases).A journal from science direct also states that Air pollution has effect on digestive, urinary and nervous system.

- First 7 out of 10 and 14 out of 20 most polluted cities in the world are from India.

- Air Pollution adversely effects Respiratory system, cardiovascular system, nervous system etc.

- As many as 1.2 million deaths take place every year due to air pollution in India.

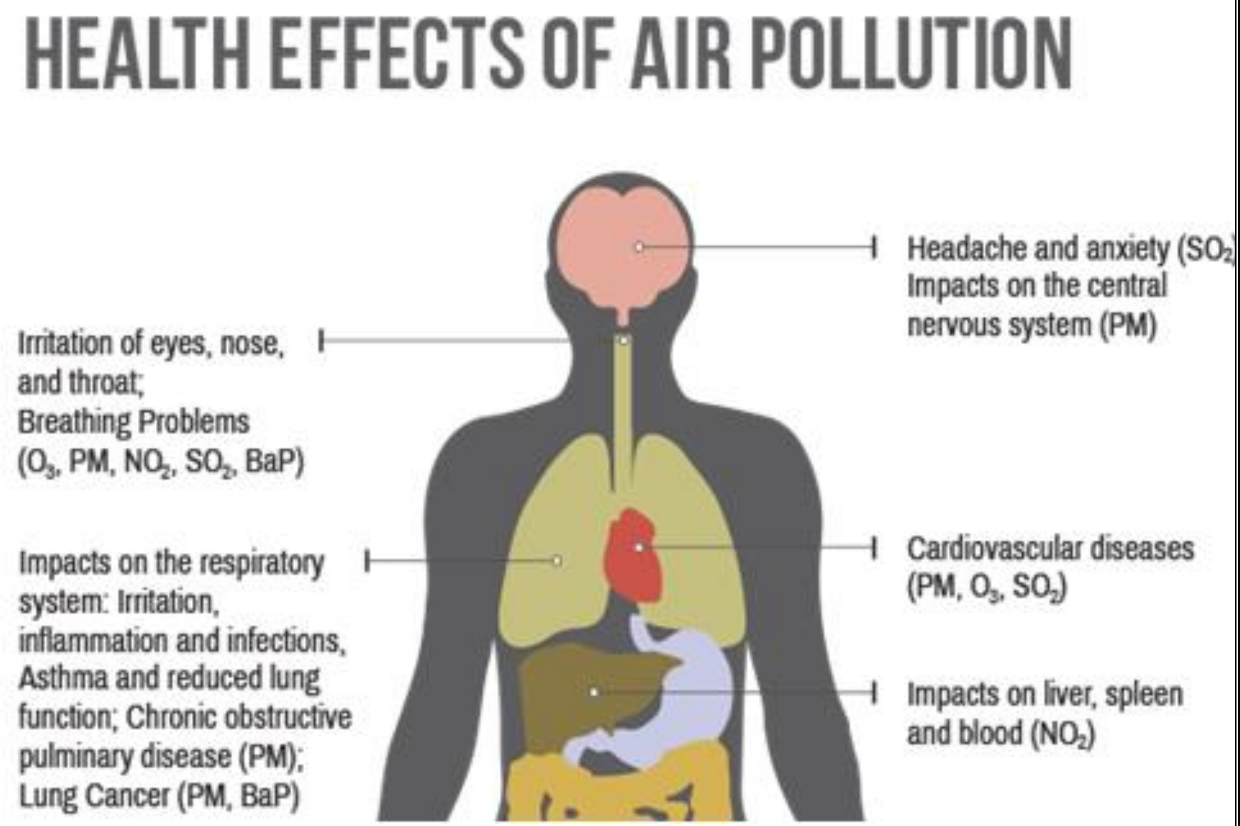

Figure-1.1

1.1 PROBLEM STATEMENT

To develop a system using MQ135 sensor that can detect and alert about the concentration level of harmful pollutants present in ambient air to the end users connected to the system.

1.2 PROPOSED SYSTEM

The system that we are designing is to detect the harmful air pollutants and alert the public belonging to the areas having dedicated hardware about cautions and precautions of outdoor air quality. The system is going to be cost effective finished product which can work anywhere using renewable energy sources to produce the desired output that can be freely accessible to the public. The system consists of monitoring sensor, MQ135 Gas sensor that communicates with the Arduino. The back-end of Arduino collects real time data from the sensor and converts it into information delivered to users through web portals and LCD. The sensor(MQ135 Gas sensor) is equipped with gaseous data as well as data logging and communication capabilities.

1.3 SCOPE OF THE MINI PROJECT

This system is useful to keep a check of various pollutants that affect our lives directly or indirectly. Since this system consists of a dedicated hardware and runs on low power, this is deemed to be a cost effective system.Possible Future Work: Using Location based services, hardware mounted in multiple areas, can alert the public in certain areas.

CHAPTER II

2. TOOLS AND TECHNOLOGIES

2.1 LITERATURE SURVEY

Some of the existing methodologies for the air pollution monitoring are described as below, In plug and sense device method, it Uses multiple sensors with location co-ordinate, AQI LED indicator is actuated as per pollution level and the Real time pollution level visualized using line graph [2].In distributed sensor data computing, it uses distributed intelligence for the sensor nodes and uses spatial database for locations .[3] In Arduino based method it uses sensor devices for data, Uses ESP8266 Wi-Fi module for connection to server, Uses Node.js and Node RED for displaying data on the server side.[4]In personal assessment methods, Biochemical dose assessment methods are used Ex .Biomarkers [5].In ZigBee technology, ZigBee transmitters and receivers are used, GPS module is used for locations for pollution level on map .[6]

1. The effects of air pollution on human health range from good to severe depending on

2. The ‘AQI Category, Pollutants and Health Breakpoints’, released under Swacch Bharat mission.

3. These values can be obtained from solid state sensors and alerts can be sent to user when the AQI exceeds a breakpoint.

4. Web portal is selected as the GUI to display the alerts.

2.1.1. Pollutants can be grouped to four categories

1. Gaseous pollutants (e.g. SO2, NOx, CO, ozone, Volatile Organic Compounds).

2. Persistent organic pollutants (e.g. dioxins).

3. Heavy metals (e.g. lead, mercury).

4. Particulate Matter.

2.1.2. Health effects:

The different composition of air pollutants, the dose and time of exposure and the fact that humans are usually exposed to pollutant mixtures than to single substances, can lead to diverse impacts on human health.

2.1.3. Effects of air pollutants on different organs and systems

2.1.3.1. Respiratory system

Numerous studies describe that all types of air pollution, at high concentration, can affect the airways. Nevertheless, similar effects are also observed with long-term exposure to lower pollutant concentrations. Symptoms such as nose and throat irritation, followed by bronchoconstriction and dyspnoea, especially in asthmatic individuals, are usually experienced after exposure to increased levels of sulphur dioxide, nitrogen oxides (Kagawa, 1985), and certain heavy metals such as arsenic, nickel or vanadium. In addition, particulate matter that penetrates the alveolar epithelium and ozone initiate lung inflammation. In patients with lung lesions or lung diseases, pollutant-initiated inflammation will worsen their condition. Moreover, air pollutants such as nitrogen oxides increase the susceptibility to respiratory infections. Finally, chronic exposure to ozone and certain heavy metals reduces lung function, while the later are also responsible for asthma, emphysema, and even lung cancer. Emphysema-like lesions have also been observed in mice exposed to nitrogen dioxide.

2.1.3.2. Cardiovascular system:

Carbon monoxide binds to haemoglobin modifying its conformation and reduces its capacity to transfer oxygen. This reduced oxygen availability can affect the function of different organs (and especially high oxygen consuming organs such as the brain and the heart), resulting in impaired concentration, slow reflexes, and confusion. Apart from lung inflammation, systemic inflammatory changes are induced by particulate matter, affecting equally blood coagulation. Air pollution that induces lung irritation and changes in blood clotting can obstruct (cardiac) blood vessels, leading to angina or even to myocardial infarction.Symptoms such as tachycardia, increased blood pressure and anaemia due to an inhibitory effect on haematopoiesis have been observed because of heavy metal pollution.

Finally, epidemiologic studies have linked dioxin exposure to increased mortality caused by ischemic heart disease, while in mice, it was shown that heavy metals can also increase triglyceride levels.

2.1.3.3. Nervous system

The nervous system is mainly affected by heavy metals (lead, mercury and arsenic) and dioxins. Neurotoxicity leading to neuropathies, with symptoms such as memory disturbances, sleep disorders, anger, fatigue, hand tremors, blurred vision, and slurred speech, have been observed after arsenic, lead and mercury exposure. Especially, lead exposure causes injury to the dopamine system, glutamate system, and N-methyl-D-Aspartate (NMDA) receptor complex, which play an important role in memory functions. Mercury is also responsible for certain cases of neurological cancer. Dioxins decrease nerve conduction velocity and impaired mental development of children.

2.1.3.4. Urinary system:

Heavy metals can induce kidney damage such as an initial tubular dysfunction evidenced by an increased excretion of low molecular weight proteins, which progresses to decreased glomerular filtration rate (GFR). In addition, they increase the risk of stone formation or nephron calcinosis and renal cancer.

2.1.3.5. Digestive system:

Dioxins induce liver cell damage, as indicated by an increase in levels of certain enzymes in the blood, as well as gastrointestinal and liver cancer.

2.1.3.6. Exposure during pregnancy

It is rather important to mention that air pollutants can also affect the developing foetus. Maternal exposure to heavy metals and specially to lead, increases the risks of spontaneous abortion and reduced foetal growth (preterm delivery, low birth weight). There are also evidence suggesting that parental lead exposure is also responsible for congenital malformations, and lesions of the developing nervous system, causing important impairment in new-born's motor and cognitive abilities. Similarly, dioxins were found to be transferred from the mother to the foetus via the placenta. They act as endocrine disruptors and affect growth and development of the central nervous system of the foetus. In this respect, TCDD is considered as a developmental toxin in all species examined.

2.2 HARDWARE REQUIREMENTS

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

2.3 SOFTWARE REQUIREMENT

- Arduino IDE
- Embedded C
- MQ135 Gas Sensor Library

CHAPTER III

3. SYSTEM DESIGN

3.1 FLOWCHART

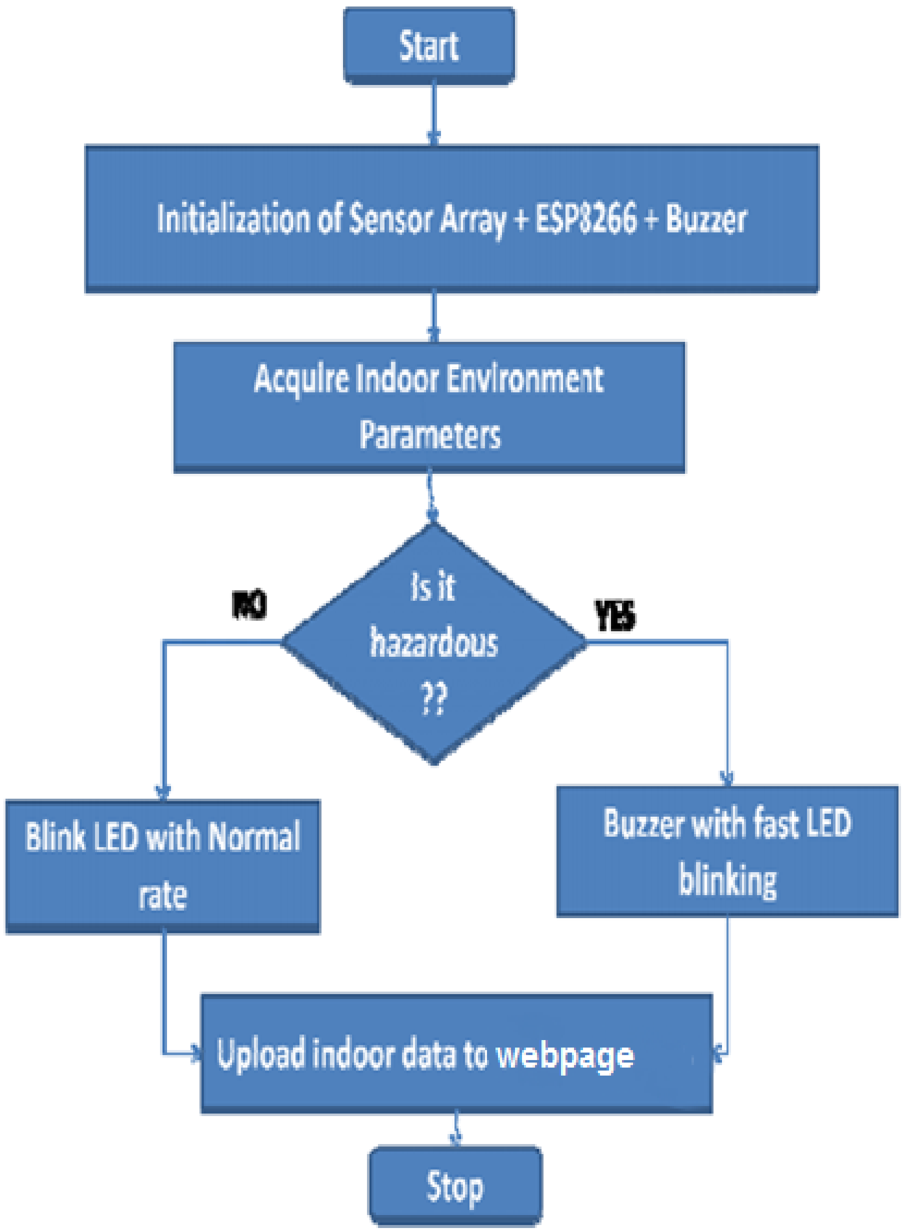

Figure-3.1.1

3.2 SYSTEM ARCHITECTURE

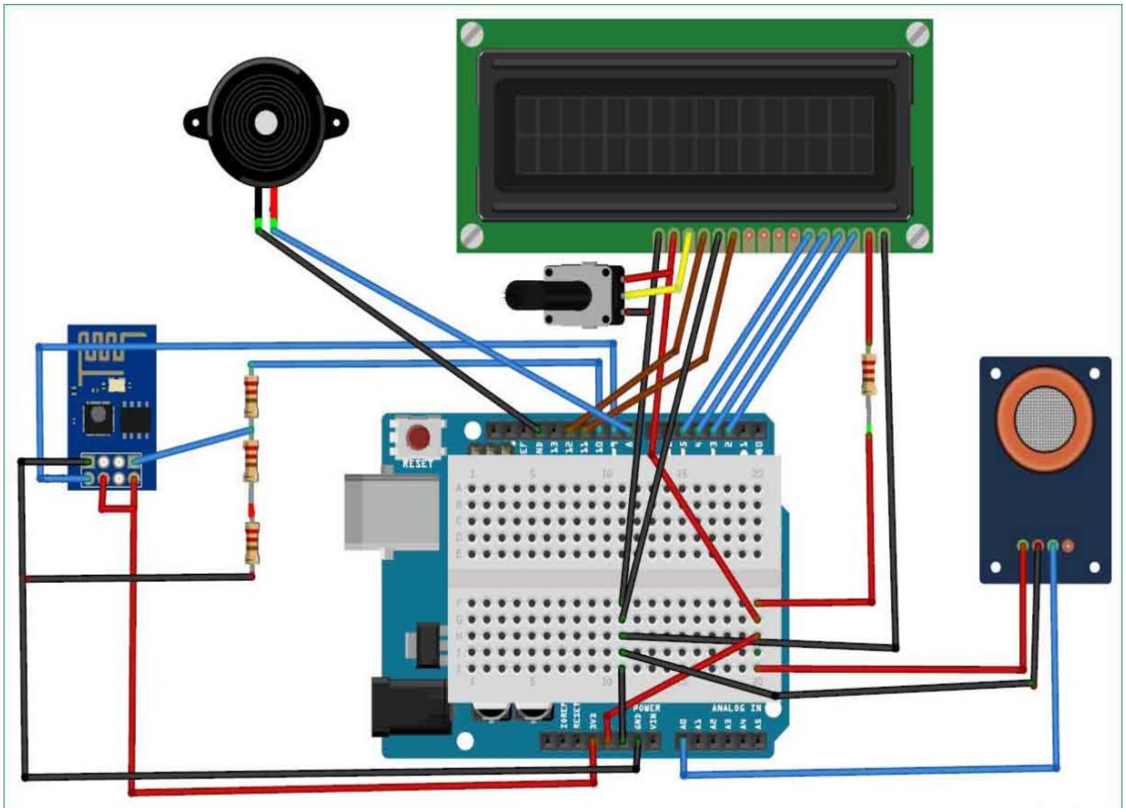

Figure-3.2.1

1. **Arduino Uno**

Arduino Uno is one of the most popular prototyping boards. It is small in size and packed with rich features. The board comes with built-in Arduino boot loader. It is an Atmega 328 based controller board which has 14 GPIO pins, 6 PWM pins, 6 Analog inputs and on board UART, SPI and TWI interfaces. In this IOT device, 9 pins of the board are utilized. There are six pins used to interface the character LCD. There are two pins utilized to interface the ESP8266 Wi-Fi Module and an analog input pin is used to connect the MQ-135 sensor.

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, which is reserved for future purposes. Stronger RESET circuit. Atmega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

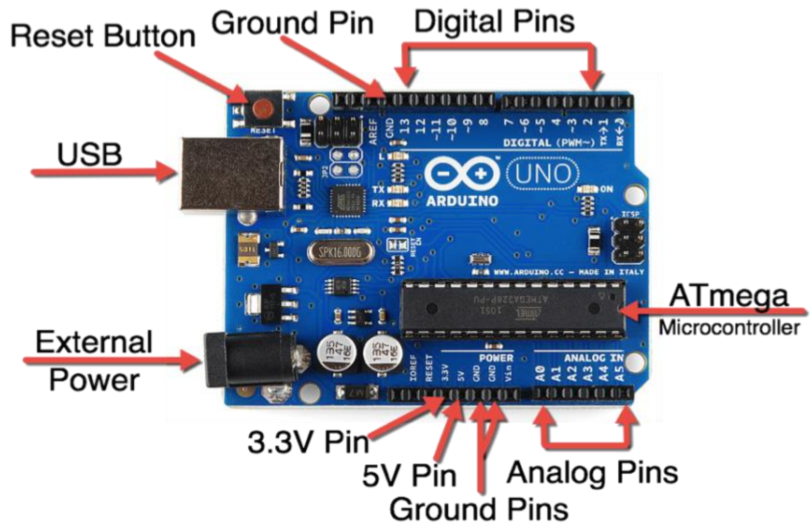

Figure-3.2.2

SPECIFICATIONS

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip

(NOTE: works with Eagle 6.0 and newer) Schematic: arduino-uno-Rev3-schematic.pdf Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows: VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it. 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA. GND. Ground pins. IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kohms. In addition, some pins have specialized functions: Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. External Interrupts: 2 and 3.These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details. PWM: 3, 5, 6, 9, 10, and 11.Provide 8-bit PWM output with the analogWrite() function. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library. LED: 13.There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality: TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board: AREF. Reference voltage for the analog inputs. Used with analogReference(). Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip.

This setup has other implications. When the Uno is connected to a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program themicrocontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by: On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

General Pin functions

- LED: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- VIN: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source).You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- 3V3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND: Ground pins.
- IOREF: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- Reset: Typically used to add a reset button to shields which block the one on the board.

Special Pin Functions

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode(),digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the analogReference () function.

In addition, some pins have specialized functions:

- Serial / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM (Pulse Width Modulation): 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite() function.
- SPI (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- TWI (Two Wire Interface) / I²C: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- AREF (Analog Reference): Reference voltage for the analog inputs.

Figure-3.2.3

2. 16X2 Characters LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix.

This LCD has two registers, namely,

(i) Command and (ii) Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

The 16X2 LCD display is used to monitor the sensor values read by the Arduino board from MQ-135. It is interfaced with the Arduino Uno by connecting its data pins D4 to D7 with pins 6 down to 3 of the controller respectively. The RS and E pins of the LCD are connected to pins 13 and 12 of the controller respectively. The RW pin of the LCD module is connected to the ground.

Figure-3.2.4

Pin No.	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; the best way is to use a variable resistor such as a potentiometer. The output of the potentiometer is connected to this pin. Rotate the potentiometer knob forward and backwards to adjust the LCD contrast.	Vo / VEE
4	Selects command register when low, and data register when high	RS (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; Extra voltage push is required to execute the instruction and EN(enable) signal is used for this purpose. Usually, we make it en=0 and when we want to execute the instruction we make it high en=1 for some milliseconds. After this we again make it ground that is, en=0.	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-

Table-3.2.1

FEATURES

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle • B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

3. ESP8266 Wi-Fi Module

The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application networking functions from another application Each ESP8266 module comes pre-programmed with an AT command.

The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network

- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- For connecting ESP8266 Module with Arduino Uno, you need 3.3 voltage regulators because Arduino is not capable of providing 3.3v to ESP8266.

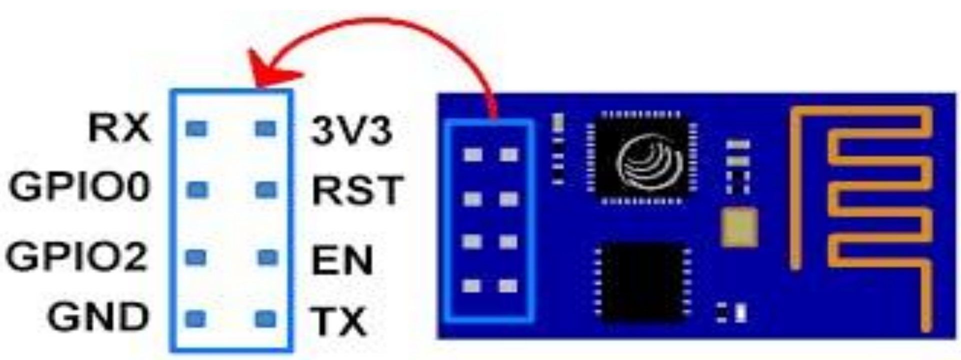

Figure-3.2.5

4. MQ-135 Gas sensor

The MQ-135 gas sensor senses the gases like ammonia nitrogen, oxygen, alcohols, aromatic compounds, sulphide and smoke. The operating voltage of this gas sensor is from 2.5V to 5.0V. MQ-135 gas sensor can be implementation to detect the smoke, benzene, steam and other harmful gases.

Figure-3.2.6

MQ-135 Sensor Features

- Wide detecting scope
- Fast response and High sensitivity
- Stable and long life
- Operating Voltage is +5V
- Detect/Measure NH_3, NO_x, alcohol, Benzene, smoke, CO_2, etc.
- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)
- Preheat duration 20 seconds
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

CHAPTER IV

4. SYSTEM IMPLEMENTATION AND METHODOLOGIES

4.1 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 https://github.com/GeorgK/MQ135.

Using this library you can directly get the PPM 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. l***cd.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;
}
```

CHAPTER V

5. TESTING

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.

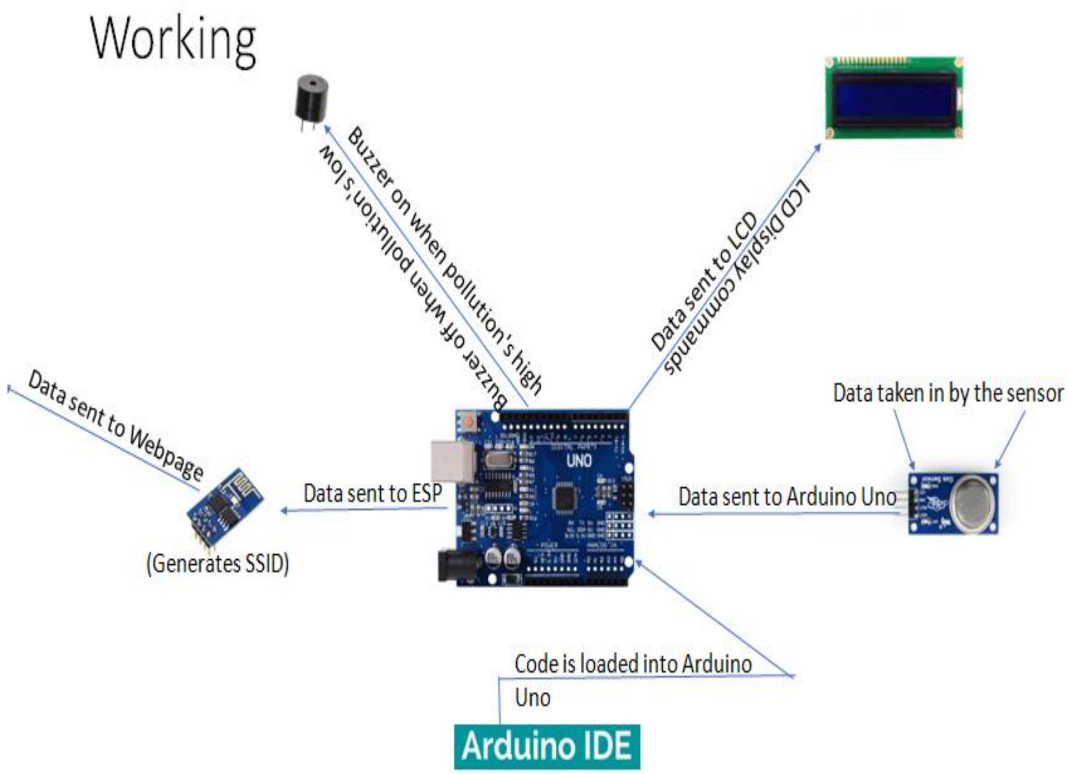

Figure-5.1

5.1 SCREENSHOTS

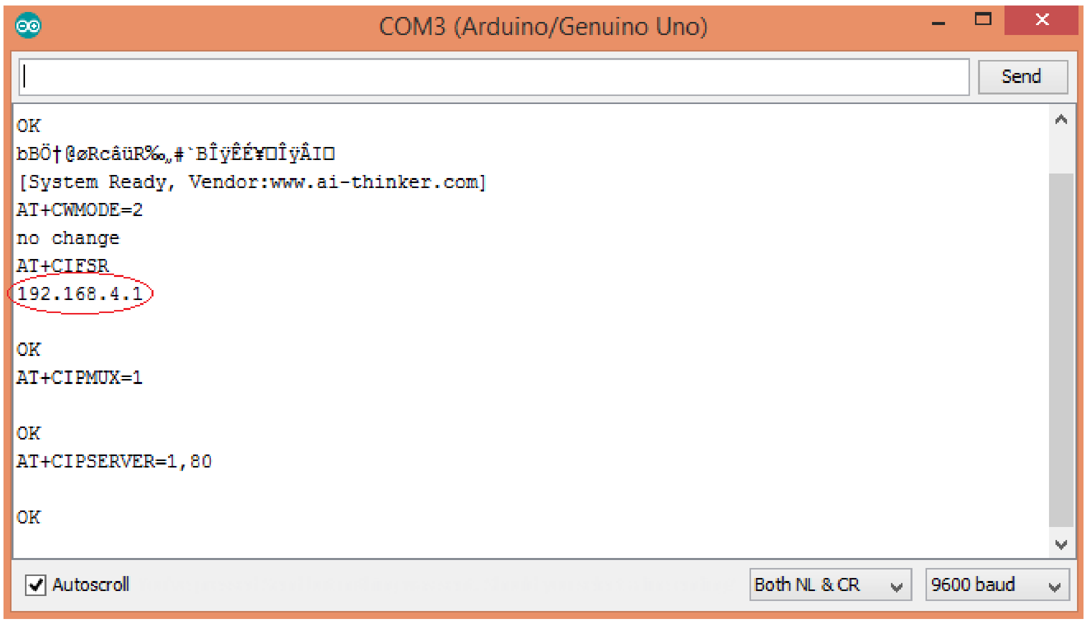

Figure-5.1.1

5.2 RESULTS SCREENSHOTS

Type this IP address in your browser, it will show you the output as shown below.

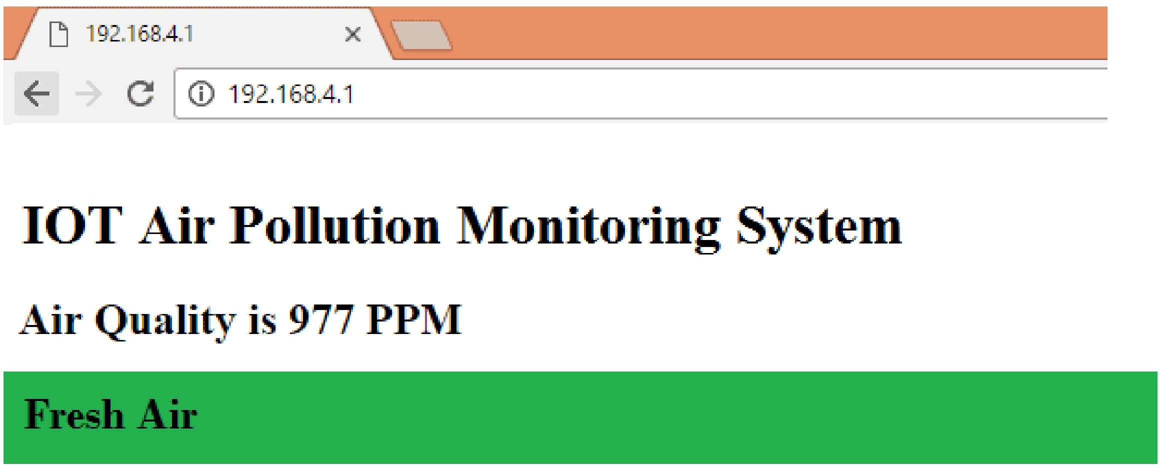

Figure-5.2.1

If the air quality is less than 1000ppm, "**Fresh air**" is displayed on the webpage with a **"Green"** background.

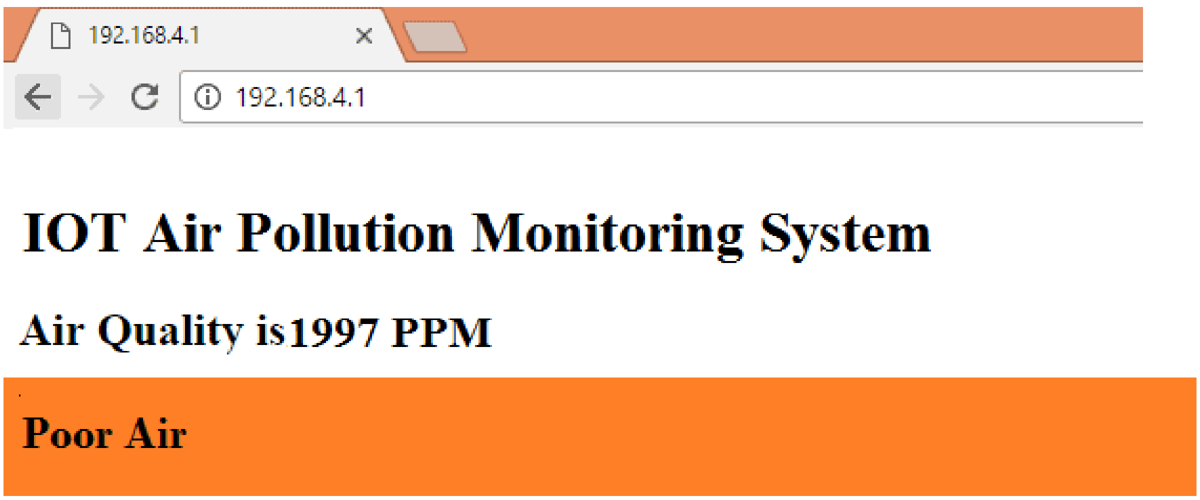
192.168.4.1
192.168.4.1
IOT Air Pollution Monitoring System
Air Quality is1997 PPM
Poor Air

Figure-5.2.2

If the air quality ranges between 1000ppm, “**poor air**” is displayed on the webpage with “**orange**” background.

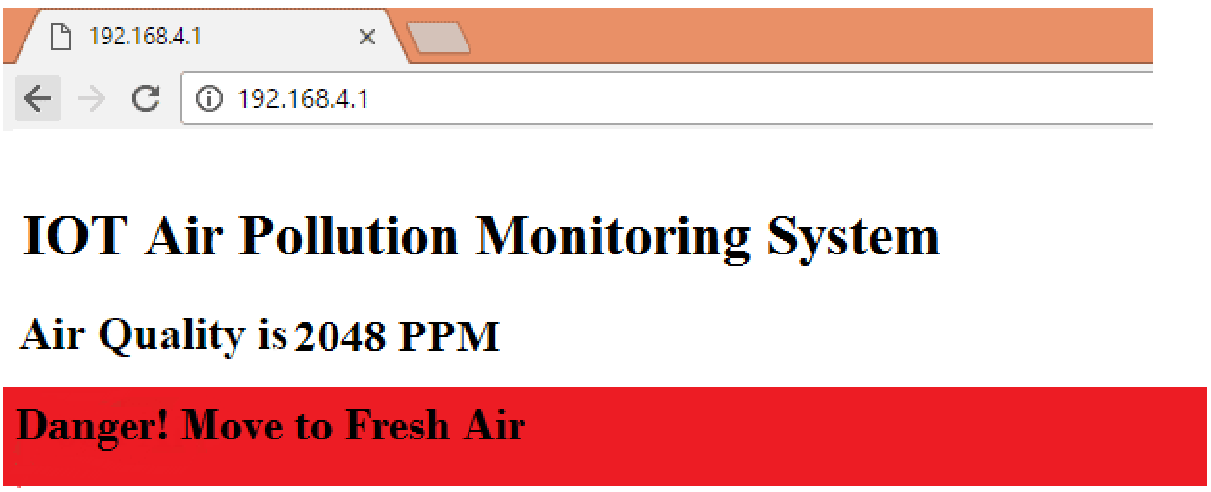

Figure-5.2.3

If the air quality is greater than 2000ppm,”**Danger! Move to Fresh Air**” is displayed on the webpage with a “**Red**” background.

CHAPTER VI

6. FUTURE SCOPE AND CHALLENGES

6.1 Future Scope:

Using Location based services, hardware mounted in multiple areas alerts the public in a certain area.

6.2 Challenges:

- Availability of sensor nodes in the market.

- Wide spread of the wireless sensor network over an entire area.

CHAPTER VII

Conclusion

This project taken up for Public Health monitoring is very helpful to create an awareness and alert public about the caution and precaution which in the case of negligence or unawareness might lead to various harmful effects on human health.

There is increasing interest from the public, city administrators and regulators in the air quality within cities, there is also much greater awareness of the costs and impact of poor air quality. There are strong market forecasts for continued growth in this sector, in part driven by the needs of governments to reduce air pollution. Governments, cities and entrepreneurs will continue to invest in this sector, as without safe, clean air, cities and communities cannot flourish.

We anticipate that new air quality monitoring solutions will emerge, their development stimulated by the continued improvements to the capabilities of IoT sensors/devices, the wide availability of Mobile IoT technologies and the increased focus by governments on air quality.

We encourage mobile operators to explore these new revenue opportunities and to consider the additional value of including network derived mobility data (which is a unique network operator capability) as part of their offering.

REFERENCES

[1] Gomez Ruiz, Carlos Andres "CleanWiFi- the wireless network for air quality monitoring and environmental education in smart cities," IEEE ITU Kaleidoscope: ICTs for sustainable world, pp. 1-6, November 2016.

[2] Sitaram Tadepalli. "Implementation of a Wi-Fi based plug and sense device for dedicated air pollution monitoring using IoT," Online international conference on green engineering and technologies, pp. 1-7, Feb, 2016.

[3] Wein wang, Suparna de, Xin Huang. "Distributed sensor data computing in smart city applications," IEEE International Symposium on a world of wireless, mobile and multimedia networks, 2017.

[4] Philip Breitgger, Alexander Bergmann, "Air quality and health effects — how can wireless networks contribute? A critical review," International Conference on broadband communications for next generation networks and multimedia applications pages 457-462, 2016.

[5] Giovanni B. Fioccola, Raffaele Sommese "Polluino: An efficient cloud-based management of IoT devices for air quality monitoring"IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016.

[6] Philip Breitgger, Alexander Bergmann, "Air quality and health effects — how can wireless networks contribute? A review, "International Conference on broadband communications for and multimedia applications pages 457-462, 201.