ABSTRACT

Air pollution is a critical problem that is currently plaguing Asia in the perilous fight against environmental pollution. It refers to the harmful substances and pollutants in the air that play an essential role in the depletion of the ozone layer while being a grave hazard to the ecological system. A principal cause of air pollution is the increase in vehicular traffic, especially in Asia, that has been noted in recent years. The level of pollution is increasing rapidly due to factors like industries, urbanization, increase in population, and vehicle use which can affect human health. According to many recent studies, Carbon Monoxide is fatal enough to cause health issues that may lead to death in some extreme cases.

Internet will play an important role in the communication of physical things. Physical objects can be empowered through the embedded electronics into it, to make them smart and at the same time, IoT gives the connection among these objects to give high quality of life to the people. Hardware and software components work cooperatively to build the IoT. IoT Based Air Pollution Monitoring System is used to monitor the Air Quality over a web server using the Internet. It will trigger an alarm when the air quality goes down beyond a certain level, which means when there are sufficient amounts of harmful gases present in the air. It will show the air quality in PPM on the LCD and as well as on the webpage so that air pollution can be monitored very easily.

This device is intended to provide the user with a cost-efficient means of determining air quality. Our sensor focuses on the five components of the Environmental Protection Agency's Air Quality Index: ozone, particulate matter, carbon monoxide, sulphur dioxide, and nitrous oxide.

The system uses MQ135 sensor for monitoring Air Quality as it detects the most harmful gases and can measure their amount accurately.

CHAPTER 1:
INTRODUCTION

1.1 Introduction

Air pollution is the biggest problem of every nation, whether it is developed or developing. Health problems are growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of a lot of gaseous pollutants. Air Pollution is the presence of substances in the atmosphere that are harmful to the health of humans and other animals. There are different types of air pollutants such as gases (Ozone, Nitrogen Dioxide, Carbon Monoxide, etc.) and particulates (PM10 and PM2.5). Harmful effects of pollution include mild allergic reactions such as irritation of the throat, eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma.

According to a study published in the scientific paper titled 'Health and economic impact of air pollution in the States of India: The Global Burden of Disease Study 2019', 1.7 million deaths were attributable to air pollution in India in 2019, which was 18% of the total deaths in the country, while the economic loss due to the lost output from premature deaths and morbidity from air pollution was 1.4% of the GDP in India during this time, which is equivalent to ₹260,000 crore. Furthermore, emerging researches, including a study from Harvard T.H. Chan School of Public Health, finds that breathing more polluted air over many years may itself worsen the effects of COVID-19.

Currently, the monitoring of air quality levels is achieved by placing sensors in various locations and the sensors alert if the pollution levels exceed the threshold level. However, there hardly exists a mobile solution wherein an individual can contribute to the pollution level checking from the convenience of their homes. Moreover, solutions for forecasting gas and particulate matter levels are location specific and require expensive training data (Meteorological data, wind speeds, etc.) which is unavailable in the reach of the common people.

Therefore, we aim to collect data about the concentration of NO2, CO_2, CO, as well as temperature and humidity via a compact and affordable solution, and host its visualization on a website as well as on a mobile app. We calculate the Air Quality Index (AQI) and also predict their future concentrations using machine learning approaches. The future concentration levels that are being predicted is aimed to provide maximum accuracy based on the regions in which the system is placed. The proposed device can be placed anywhere in the user's home and the user can obtain real time data of the pollution levels in their immediate locality. The accuracy of the prediction model can be improved based on the data sensed from the user's surroundings.

The common pollutant and their health effects with its sources are tabulated in Table 1.1 .

Pollutant	**Health Effects**	**Sources**
Carbon Monoxide (CO)	Exposure to elevated levels of carbon monoxide can adversely affect the functioning of the heart, resulting in cardiac ischaemia, increased hospital admissions, and possibly increased cardiac mortality.	By the incomplete combustion of fossil fuels, largely from motor vehicles and other mobile sources.
Nitrogen Dioxide (N02)	Exposure to elevated levels of nitrogen oxides can contribute to respiratory illness, aggravation of asthma in children, and reduced lung growth.	By the combustion of fossil fuels.
Ozone (O_3)	Ozone irritates the respiratory tract. Exposure to ozone in sensitive people can cause chest tightness, coughing, and wheezing.	By atmospheric reactions involving nitrogen oxides, volatile organic compounds, and sunlight.
Sulphur Dioxide	Exposure to sulphur dioxide causes severe problems for people with asthma and is also associated with increased risk of lung cancer and chronic bronchitis	By the combustion of fossil fuels containing sulphur, including coal, oil, gasoline, and diesel
Particulate Matter	causes premature mortality from cardiovascular and respiratory diseases	By the combustion of fossil fuels, while coarse particulate matter originates from road dust, diesel engines, and crushing and grinding operations.
Lead (Pb)	Lead exposure can cause cognitive deficits, developmental delays, hypertension, impaired hearing, attention deficit disorder, reduced intelligence, and learning disabilities.	By gasoline smelters, dust, paint chips, consumer products, and lead shot are other important sources of exposure to lead.

TABLE 1.1: THE SIX COMMON POLLUTANTS AND THEIR HEALTH EFFECTS

1.2 Purpose of Project

The project is an implementation of IoT (Internet of Things) Based Air Pollution Monitoring System Using Arduino. Air pollution is a growing issue and it is necessary to monitor air quality for a better future and healthy living for all. IoT is getting popular day-by-day and standards are on its way. Therefore, collection of air quality information is easier. Analysis of monitoring data allows us to assess how bad air pollution from day to day. According to the recent survey, Dhaka, the capital of Bangladesh is the third in the list of most air-polluted city. Thus because of this expansion in the quantity of vehicles contamination is developing quickly and it influencing people groups wellbeing too. This air contamination makes disease and harm safe, neurological, regenerative and respiratory framework. In extraordinary cases, it can likewise cause passing. As indicated by overview 50000 to 100000 unexpected losses occurred to us only because of air contamination [2]. Along these lines, there is a requirement for checking air quality and to monitor it. IoT is the system of physical gadgets, vehicles, home apparatuses, and different things implanted with hardware, programming, sensors, and availability which empowers these articles to associate and trade information. IoT permits articles to be noticed or controlled. In this paper, I am proposing and going to piloting a model which IoT to screen air contamination.

1.3 Objectives of Monitoring Air Quality

The air quality monitoring program design dependent upon the monitoring specific objectives specified for the air quality management in the selected area of interest. Defining the output influence, the design of the network and optimize the resources used for monitoring. It also ensures that the network is specially designed to optimize the information on the problems at hand. There might be different objectives for the development of the environmental monitoring and surveillance system. Normally, the system has to provide on-line data and information transfer with a direct /automatically/ on-line quality control of the collected data. Several monitors, sensors and data collection systems to be applied to make on-line data handover and control likely. The main objectives stated for the development of an air quality measurement and surveillance program might be to facilitate the background concentration(s) measurements, monitor current levels as a baseline for assessment, check the air quality relative to standards or limit values, detect the importance of individual sources, enable comparison of the air quality data from different areas and countries, collect data for the air quality management, traffic and land-use planning purposes, observe trends (related to emissions), develop abatement strategies, determine the exposure and assess the effects of air pollution on health, vegetation 3 or building materials, inform the public about the air quality and raise the awareness, develop warning systems for the prevention of undesired air pollution episodes, facilitate the source apportionment and identification, supply data for research investigations, develop/validate management tools (such as models), develop and test analytical instruments and to support legislation in relation to the air quality limit values and guidelines. The relationships between the data collected and the information to be derived from them must be taken into account when a monitoring program is planned, executed and reported. This emphasizes the need for users and potential users of the data to be involved in planning surveys, not only to ensure that the surveys are appropriate to their needs but also to justify committing the resources.

1.4 Air Quality Parameters

The important parameters that are considered in the proposed framework include:

Carbon Dioxide (CO2) – CO2 is colorless, odourless gas and non-combustible gas. Also, it is measured under the category of smother gases that have ability of interfering the availability of oxygen for tissues. Carbon Dioxide is a gas vital to life in the world, because it is one of the most vital elements evolving photosynthesis process, which converts solar into chemical energy. The concentration of CO2 has amplified due mainly to massive remnant fuels boiling. This increase makes plants grow rapidly. The rapid growth of undesirable plants leads to the increase use of chemicals to eliminate them.

Sulphur Dioxide (SO2) - Sulphur Dioxide is a colorless gas, detectable by the distinct odour and taste. Like CO2, it is mainly due to fossil fuels boiling and to manufacturing processes. In high attentions may cause breathing problems, especially in sensitive groups, like asthmatics. It contributes to acid rains.

Smoke - About 1 million people are in custom of tobacco smoking globally of which majority population is from rising countries. Every year nearly 4.9 million people expired due to smoking allow to 2007 report. In addition, second hand smoke is serious threat to the health of people of all age's causes 41000 deaths each year.

Temperature and humidity- Quantity of temperature is an important for safety of people and affects our life skills. Greenhouse outcome can be observed by measuring temperature and comparing temperature changes from historical to present time especially since the industrial revolution using climate data. Humidity is a type of gas that guards us from UV rays from the sun and helps trick heat on Earth, thereby making the climate on Earth, a pleasant one for living. However, as humidity increases, the warmth on Earth also increases which makes our life uncomfortable. Humidity is essential for various storage and food processing facilities.

1.5 Beneficiaries

1.5.1 Society People

Whether closed or open premises, human health gets affected by the increase in air pollutants or particulate matter. Thus, installing an air quality monitoring system helps monitor the presence of pollutants, resulting in better environmental conditions for humans to reside. This also impacts their health and reduces the chances of occurring any health issues by maintaining a moderate ambiance or as required. Hence, IoT-powered solutions provide better services to the industrialists, which in turn, provides better services to their customers. It creates a positive impact on human health by eliminating unwanted air pollutants and particulate matter by allowing the authorities to take decisions according to the situation.

1.5.2 Special Care Units in the hospitals

Hospital is a place that is prone to diseases and now after the current pandemic it has become more evident that the air around should be free from all kind of toxics. A air pollution detector at low cost will help the patients breath better quality of air. This is be a add on to the hospital services as, whenever the air quality would show alert on the monitor adequate actions can be taken by the hospital authorities and all it would be easy to monitor the data using cloud services.

1.5.3 Urban and Rural planning sector and green building management system

Environmental health has been a topic of discussion for decades. Different policies and regulations pertaining to the emission of pollutants in the air have been imposed to keep the air quality high. Hence, to keep the emission rate well under control as per the determined guidelines, it is important for industries to monitor the production of harmful gases. By using outdoor air quality monitoring systems, companies can track the air quality index around their manufacturing units and subsequently control their emission rates. This helps them to adhere to regulations and prevent any lawfully enforced consequences from air quality administering organizations when air pollution levels exceed its limits.

1.6 Advantages of Proposed Project

This system monitors the current pollution status. This will update in the web server. So, we can monitor anywhere through internet. Web server is also to monitor the current pollution status.

1.6.1 Portability

It is a compact device which consists of many sensors including cloud which are all combined by using internet of things (IoT).

1.6.2 Safety

One can avoid from going to particular location by redirecting themselves or by taking safety protections such as wearing mask and can reduce over dumped wastages in a particular locality area.

1.6.3 Cost

Compared to others it's efficient and low cost because sensor is clubbed by using internet of things (IoT) and Arduino microcontroller.

1.6.4 Simple maintenance

As the Project deals with the software embedded C, so maintenance will be easy and this can also be installed in GSM android mobiles.

CHAPTER 2:
COMPONENT DESCRIPTION

Hardware Component:-

1) MQ-135 Gas sensor

2) MQ-2 Gas Sensor

3) MQ-7 Gas Sensor

4) 16x2 LCD

5) Wi-Fi module ESP8266

6) Breadboard

7) 10K potentiometer

8) 220-Ohm Resistor

9) Buzzer

10) Jumper Wire M to M and M to F

Software Requirement:-

1) Arduino IDE 1.6.13 – To code the Node MCU

2) Thingspeak IoT Cloud – For reading and visualizing sensor values

Bill of Materials:

Items	Quantity	Cost
MQ135, MQ2 and MQ7 Gas Sensor	1 each	₹ 525
Node MCU	1	₹ 290
Breadboard	1	₹ 135
Jumper Wire	12	₹ 20
10 kΩ Resistor	2	₹ 1
LCD 16*2 Display	1	₹150
Grand Total		₹1121

2.1 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.

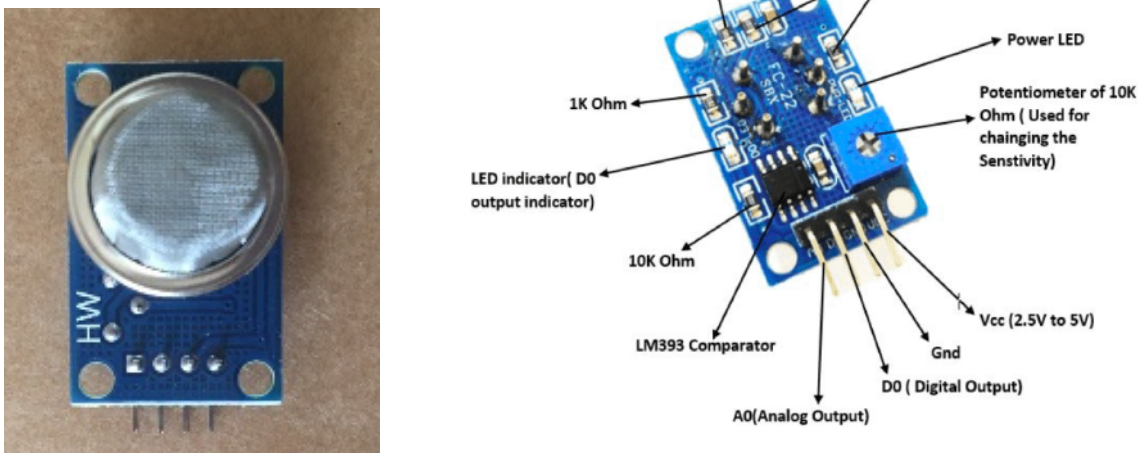


Fig. 2.1: MQ-135 Gas Sensor

Pin 1	VDD	power supply 5V DC
Pin 2	GND	used to connect the module to system ground
Pin 3	D0 Digital Out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer
Pin 4	A0 Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas.

Table 2.1: Pin Description of MQ-135

Technical Specifications of MQ135 Gas Sensor:-

- Operating Voltage is +5V
- Detects/Measures NH3, NOx, alcohol, Benzene, smoke, CO2, etc.
- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)
- Preheat time over 24 hours
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

2.1 MQ-2 Gas Sensor

MQ2 is one of the commonly used gas sensors in MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type Gas Sensor also known as Chemiresistors as the detection is based upon change of resistance of the sensing material when the Gas comes in contact with the material. Using a simple voltage divider network, concentrations of gas can be detected.

MQ2 Gas sensor works on 5V DC and draws around 800mW. It can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide c oncentrations anywhere from 200 to 10000ppm.

Fig 2.2: MQ-2 Gas Sensor

Here are the complete specifications

- Operating Voltage is +5V
- Detects/Measures LPG, Alcohol, Propane, Hydrogen, and even methane
- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)
- Preheat time over 48 hours
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

2.2 MQ-7 Gas Sensor

Product Description:

MQ7 Gas sensor is another one of Metal Oxide Semiconductor (MOS) type Gas Sensor of MQ Gas Sensors family involving MQ 2, MQ 4, MQ 3, MQ 8, MQ 135, etc. It is mainly used to detect Carbon Monoxide. This sensor contains a sensing element, mainly aluminum-oxide based ceramic, coated with Tin dioxide (SnO_2), enclosed in a stainless-steel mesh. Whenever CO gas comes into contact with the sensing element, the resistivity of the element changes. The change is then measured to get the concentration of the gases present. The MQ7 Sensor has a small heating element present which is needed to preheat the sensor to get it in the working window. It can detect Carbon Monoxide Gas in the range of 20 PPM to 2000 PPM in the air. It finds uses in Alarm application in case of CO gas concentration build-up in the home or your car as CO is a very harmful gas and can kill a person if present over 300PPM.

- Operating Voltage is +5V
- Detects/Measures Carbon Monoxide (CO)
- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)
- Preheat time no less than 48 hours
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

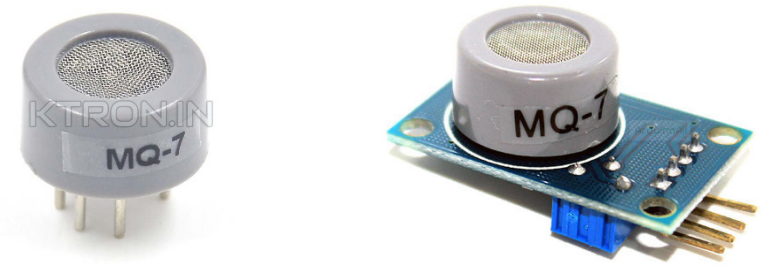


Fig 2.3: MQ-7 Gas Sensor

2.3 16X2 LCD Panel:-

Product Description:

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in colour or monochrome. [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.

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
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 V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Table 2.3: Pin Description of 16x2 LCD Panel

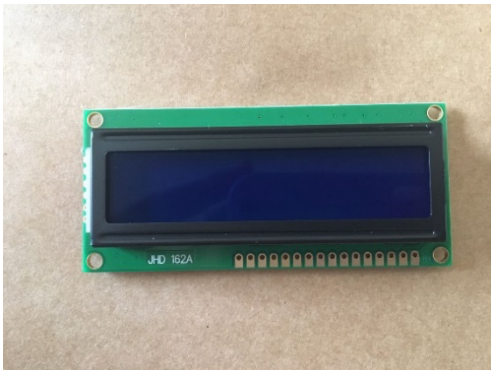


Fig 2.4: 16x2 LCD Display

2.5 NodeMCU:-

Product Description:

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.

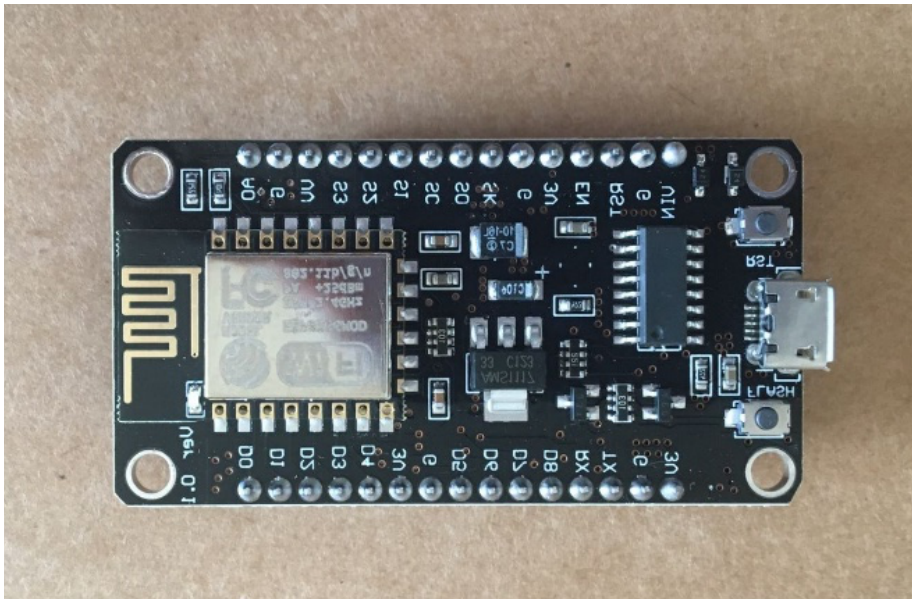


Fig 2.5: NodeMCU-ESP8266

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but du is I2C.

Table 2.4: Pin Description of NodeMCU ESP-8266

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3 V
- Input Voltage: 7-12 V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UART: 1
- SPI: 1
- I2C: 1
- Flash Memory: 4 MB
- SRAM: 64 KB

Clock Speed: 80 MHz USB-TTL based on CP2102 is included onboard, Enabling Plug and Play.

2.6 BreadBoard

Breadboards are temporary work boards for electronic circuits. The general shape of a breadboard is shown below. Compatible with most breadboards, 24-gauge wire is used to connect circuits; solid wire, not stranded. Sometimes, kits may be available with various colors of fixed lengths to specifically fit breadboards.

A bare board already has some connections. The horizontal rows are connected throughout the row and may make a complete row with the addition of a simple jumper at the center point. These rows are noted with red and blue or black markings.

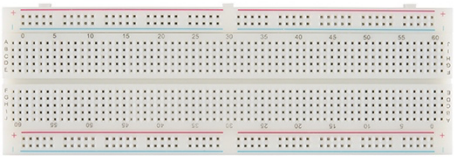


Fig 2.6- BreadBoard

2.7 10K potentiometer

Potentiometers are very useful in changing the electrical parameters of a system. It is a single-turn 10k Potentiometer with a rotating knob. These potentiometers are also commonly called as rotary potentiometer or just POT in short. These three-terminal devices can be used to vary the resistance between 0 to 10k ohms by simply rotating the knob. A potentiometer knob can also be used along with this POT for aesthetic purposes.

Specifications

- Rotary type shaft potentiometer
- 15mm shaft length[1]
- Total Resistance 10kΩ

Fig 2.7- 10k Potentiometer

2.8 220-Ohm Resistor

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Resistors act to reduce current flow, and, at the same time, act to lower voltage levels within circuits. In electronic circuits, resistors are used to limit current flow, to adjust signal levels, bias active elements, and terminate transmission lines among other uses.

Features of 220 Ohm 1/4 Watt Resistor

- Resistance: 220Ω
- Tolerance: 5%.
- Power rating: 1/4Watt.

Applications of 220 Ohm 1/4 Watt Resistor

- Voltage limiter.
- Current Limiter.
- DIY projects requiring impedance matching.

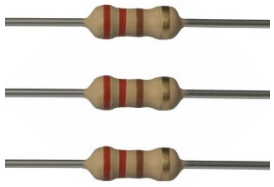


Fig 2.8- 220-Ohm Resistor

2.9 Buzzer

Buzzer is also known as Piezo Speakers (buzzers). You want to generate sound in our project you can use this simple magnetic buzzer. This is the same buzzer we have used in our development boards. This one generates a continuous beep usually when supplied with power but you can generate any tone as you wish by interfacing it with a microcontroller with proper coding.

Fig 2.9- Buzzer

A “piezo buzzer” is basically a tiny speaker that you can connect directly to an Arduino. From the Arduino, you can make sounds with a buzzer by using tone. You have to tell it which pin the buzzer is on, what frequency (in Hertz, Hz) you want, and how long (in milliseconds) you want it to keep making the tone.

2.10 Jumper wire

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

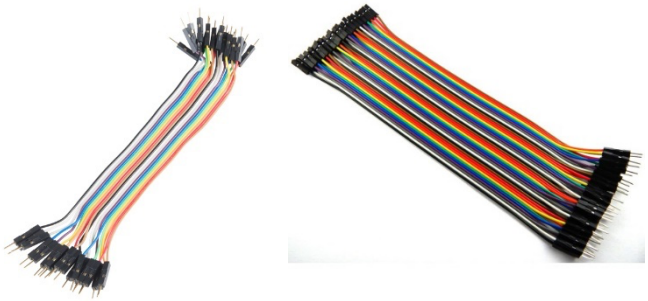


Fig 2.10- Jumper Wire M to M and M to F

2.11 Arduino IDE 1.6.13 Software

The Arduino integrated development environment (IDE) is a cross-platform application (for Microsoft Windows, macOS, and Linux) that is written in the Java programming language. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

Fig 2.11- Arduino IDE Software

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

2.12 Thingspeak IoT Cloud

ThingSpeak is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites. ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications.

ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks,[4] allowing ThingSpeak users to analyze and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from MathWorks.

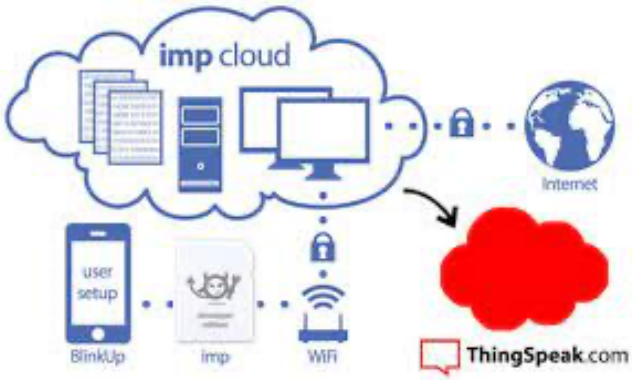


Fig 2.12- ThingSpeak IoT Cloud

CHAPTER 3
LITERATURE REVIEWS

3.1 Introduction

This chapter reviews some of the past works in processing and understanding IoT based air pollution detection monitoring system. Air pollution is not only natural medical matters impact on creating nations alike. The strong effect of air pollution on wellbeing are extremely mind blowing as there are a broad area of sources and their particular influence differ from one another. The synthetic substances reason an assortment of mankind and natural medical issues enlarge in air contamination impacts on condition also on human wellbeing. The proposed framework unit incorporates an Arduino, MQ135 Gas sensor, LCD and ESP8266 Wi-Fi Device. Almost all the past and recent works in IoT based on methods that implement these steps sequentially and independently.

3.2 Opportunity, Status & Capability of IoT

3.2.1 Opportunity of IoT

The IoT create a huge network of billions or trillions of "Things" communicating each other. The IoT is not dissident revolution over the existing technologies, it is comprehensive uses of existing technologies, and it is the creation of the new communication modes. The IoT blends the virtual world and the physical world by transporting different concepts and technical components together: pervasive networks, reduction of devices, mobile communication, and new ecosystem. In IoT, applications, services, middleware components, networks, and end nodes to structurally planned and used in entire new ways. IoT proposals a means to look into complex procedures and dealings. The IoT implies a symbiotic communication between the physical and the digital worlds: physical entities have digital complements and virtual illustration; things become context aware and they sense, communicate, interact, and exchange data, information, and knowledge. New chances meet business requirements, and new services to be created based on real-time physical world data. All from the physical or virtual world possibly be connected by the IoT. Connectivity between the things to be available to all with low cost and cannot be owned by private objects. For IoT, intelligent learning, fast placement,

best information understanding and interpreting, against fraud and malicious attack, and privacy protection are vital requirements.

3.2.2 Status of IoT

The IoT regarded as an extension of existing interaction between people and applications through a new dimension of "Things" for communication and integration. The IoT development process is a multifaceted large-scale technological novelty process. The IoT is developing from the vertical application to polymeric application. At the early stage of IoT placement, driving of domain specific requests is the main development approach. A domainspecific application might be an industrial control system with its own industry features. The application can provide various enterprise management services being combined with the industry manufacture and business processes. Polymeric requests are cross-industry applications founded on public information service stages. These requests provision both home users and industry users. The application is provided and promoted by communication operators and solution providers with large scale. For example, a vehicle integrated with sensor networks, a global positioning system (GPS), and radio communication technology provide inclusive detection, navigation, entertainment, and other information services. By preserving such information through the public service platform, consumers, original equipment manufacturers (OEMs), maintenance providers, and vehicle organization agencies can share this information and segment services to improve the vehicle, the vehicle component design, and the fabrication process through the vehicle growth management.

3.2.3 Capability of IoT

In summary, the IoT applications have the following capabilities. Location Identifying and Sharing of Location Info: The IoT system can gather the location information of IoT stations and end nodes, and then offer services based on the collected location information. The location information includes geographical position information got from the GPS, Cell-ID, RFID, etc., and complete or relative position information between things. More representative IoT applications include at least the following.

Mobile asset tracking: This application track and display the status of product using the position sensing device and statement function installed on the commodity.

Fleet management: The manager of the fleet schedule the vehicles and drivers established on the business supplies and the real-time position information collected by the vehicles.

Traffic information system: This application gets traffic information such as road traffic conditions and congested locations by tracking the location information of a large number of vehicles. The system thus contributions the driver to select the most efficient route.

Environment Sensing: The IoT system collect and process all kinds of physical or chemical environmental parameters via the locally or widely organized terminals. Typical environmental information includes temperature, humidity, noise, visibility, light intensity, spectrum, radiation, pollution (CO, CO2, etc.), images, and body pointers. Representative applications include at least the following.

Secure Communication: IoT system further establish secure data transmission channel between the application or service platform and IoT terminals based on service requirements

3.3 Open and General IoT Architecture

Standards Association planned a reference model for the IoT, which consists of sensing layer, network and business layers, and application layer. Fulfilling with this reference model, its open and general architecture, which is layered, open, and elastic. The architecture includes three functional stages as follows.

Sensing and Gateway Platform: This platform connects sensors, controllers, RFID readers, and location detecting device (e.g., GPS) to IoT network layer. Modularization of hardware, data format, and software edge is proposed for IoT terminal, IoT Gateway, and angle node. IoT terminal, IoT gateway, and tip node can include flexible modules combined with control module, common interface module, and communication module. Joint interface module collects physical interfaces of various sensors into a shared interface. Common control module can connect sensors, controllers, GPS, and RFID booklovers with a common connection protocol. The software and application parameters of an IoT terminal and IoT gateway ought to be able to self-configure and self-adapt. Modularization, shared interface, intelligent operation, self-adaption, and self-configuration are important characteristics of this stage.

Resource and Administration Platform: Network and service layer includes backbone networks and resource administration stages. The backbone network contains 3G, 4G, internet, optical

fiber network, Ethernet network, satellite networks, and private network. The resource and administration stage provides common capabilities which can be used by different IoT applications, such as data processing, data storage, security management, and application supporting. These abilities may also be invoked by specific IoT application support capabilities, e.g., to build other specific IoT application support capabilities. This stage also provides relevant control functions of network connectivity, such as access and transport resource control functions, mobility management, or authentication, authorization, and accounting for IoT terminals, services, applications, users, and developers.

3.4 Challenge & Prospect of IoT

3.4.1 Challenge of IoT

The IoT provides many new chances to the industry and end user in many application fields. Currently, however, the IoT itself lacks theory, technology architecture, and standards that integrate the simulated world and the real physical world in an integrated framework. Following key challenges are thus recorded.

Technical Challenge: IoT technology to be complex for variety of reasons. First, there are legacy heterogeneous architectures in the present networking technologies and applications, e.g., different applications and environments need different networking technologies, and the ranges as well as other features of cellular, wireless local area network, and RFID technologies are much unlike from each other. Second, communication technologies, including fixed and mobile communication systems, power stripe communications, wireless communication, and short-range wireless communication technologies, for both fixed and mobile devices, either simple or complicated, should be low cost and with reliable connectivity. At last, there are thousands of unlike applications; it is in natural to have different requirements on what parties need to communicate with each other, what kind of security solutions are appropriate, and so on. To recap, complexity and alternative technologies may introduce problems; unnecessary competition and deployment barriers in markets may also introduce problems; systems and communication mechanisms with needless dependencies may block the relocation of IoT systems to the most economic and efficient stages. All the above may block IoT to join as many “Things” as possible.

Privacy and Security Challenge: Compared with outdated networks, security and privacy issues of IoT become more prominent. Much information includes privacy of users, so that defense of privacy becomes an important security issues in IoT. Because of the mixtures of things, services, and networks, security of IoT needs to cover more administration objects and levels than traditional network security. Existing security architecture is intended from the perception of human communication, may not be suitable and directly applied to IoT system. Using existed security instruments block logical relationship between things in IoT. IoT needs lowcost- and M2M-oriented technical solutions to assurance the privacy and the security.

3.4.2 Prospect of IoT

With the development and adulthood of distributed intelligent information processing technologies, IoT systems make intelligent sensing widely available through information sharing and teamwork. The gradual founding and improvement of the standards system inescapably bring IoT into our daily life. The IoT creates an opportunity for the web-based services, thus attractive the commercial and social potential future of IoT. The growth of IoT keeps going forward along scale, collaborative, and intelligent. Promoted by technology, standardization, and application experiences, IoT applications expand the scale in the different industries, and more initiatives to be attracted to come in.

Intelligent System: The IoT bring seamless business and social networking over fast reliable and protected networks into our society. System intelligence important for the development of IoT and the key point context awareness and inter-things information exchange. Therefore, increasing and familiarizing the intelligence at the device level will be a focus of research, such as the integration of sensors and actuators, high efficiency, multi standard and adaptive communication subsystems, and adaptable antennae. Intelligences can be presented using micro control unit (MCU) on upper layers. However, physical layer so far has been far behind the mandatory intelligent level, for example, to adapt IoT devices under different radio infrastructures.

CHAPTER 4:
PROPOSED SYSTEM ARCHITECTURE

4.1 Introduction

The paper aims at designing an air pollution monitoring system which can be installed in a specific locality and to enhance the system from the previously developed systems beating the earlier disadvantages by developing an android app available for the public. This app can be used by anyone to get in live updates about the pollution in their region. It uses Arduino integrated with individual gas sensors like carbon monoxide, ammonia along with particulate matter, humidity, and smoke which measures the concentration of each gas separately. The collected data is uploaded to the cloud using thing speak platform at regular time intervals. Ethernet shield is used for connecting Arduino and cloud. Pictorial or graphical representation of values can be shown in Thing speak The users can install an android application through which they get the recent updates and graphical content up to date. The average concentration of each gas is analyzed using matlab. Then certain time control is assigned based on the standard level of each gas measured and the result can be viewed in android application. The architecture of air pollution monitoring and awareness creation system. The concentration level of each gas can be viewed both as a graph and in numerical format. Based on these values the air quality index value is calculated and the nature of the air quality in that area is determined which is also displayed through the app. Along with this, the health effects for the corresponding air quality is displayed to create awareness among the public. Additionally, they could also get to know the temperature and weather in that region. The users will not get disturbed with irrelevant data as the values displayed are location specific and help them stay tuned to the current status of air pollution.

4.2 Proposed System Architecture

Internet of Things (IoT) mainly deals with connecting smart devices to internet by joining the advantage of OSI layered Architecture. In the context of this work we propose a cluster of Air Quality Monitoring Gas Sensor MQ135 motes, which are used

to measure the concentration of Air pollutants in the air. The Gas Sensors MQ135 is interface with a tiny entrenched platform equipped with other. We have mainly used the ESP8266-12E which is an open source development boards chip. MQ135 Gas Sensor is used to collect gas concentration measurements. This sensor data would be captured and sent to the ESP8266-12E for IoT (Internet of Things) based data acquirement

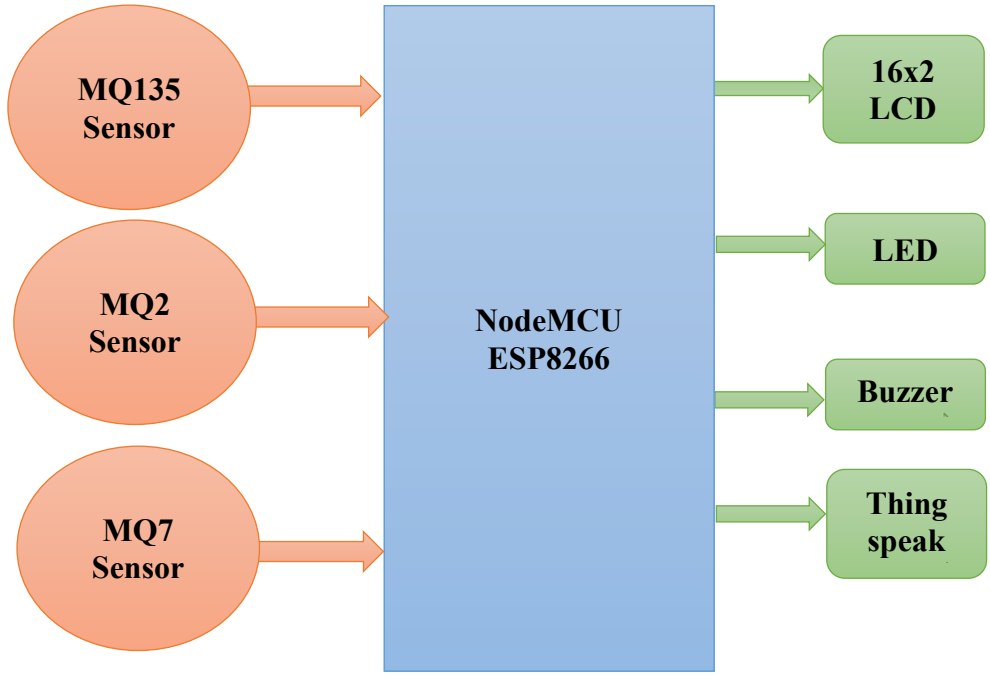


Fig 4.1: Proposed System Architecture

4.3 Circuit Diagram

I connect the ESP8266 with the Arduino. ESP8266 runs on 3.3V and connect the VCC and the CH_EN to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it communicate with the Arduino when I connect it directly to the Arduino. So, I have to make a voltage divider for it which convert the 5V into 3.3V. This can be done by connecting three resistors in series like I did in the circuit. Connect the TX pin of the ESP8266 to the pin 8 of the Arduino and the RX pin of the esp8266 to the pin 9 of Arduino. ESP8266 Wi-Fi module gives my projects access to Wi-Fi or internet.

Then I 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 7 of the Arduino which start to beep when the condition becomes true. In last, I connect LCD with the Arduino [8]. The connections of the LCD are as follows Connect pin VCC to the 5V of the Arduino; Connect pin GND to the GND of the Arduino; Connect pin RS to the pin 12 of the Arduino; Connect pin RW & GND (Read/Write) to used Jumper pin; Connect pin E to the pin 11 of the Arduino; The following four pins are data pins which are used to communicate with the Arduino; Connect pin D4 to pin 5 of Arduino; Connect pin D5 to pin 4 of Arduino; Connect pin D6 to pin 3 of Arduino; Connect pin D7 to pin 2 of Arduino.

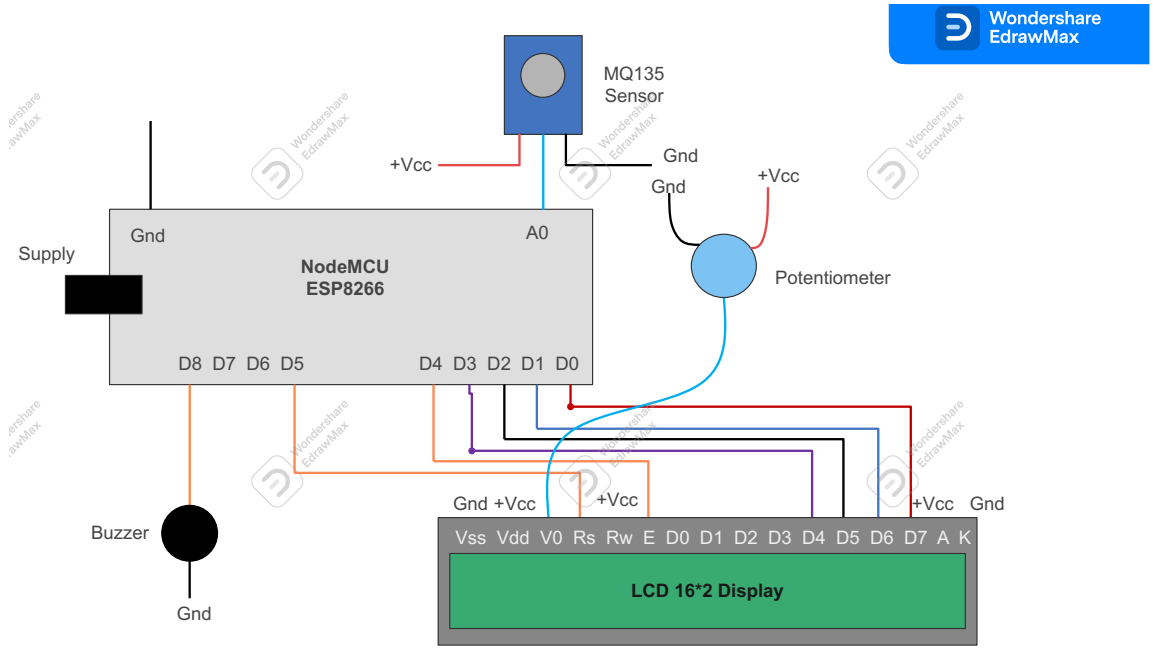


Fig 4.2: Circuit Diagram

4.4 Block diagram for proposed model of the system

The major components of my project have been mentioned in the above section. Here I explain the architecture. Gas sensor & Wi-Fi Device are connected to the Arduino board. Also LCD connected to the Arduino board for displaying information. I am monitor the Air Quality over a serial monitor & LCD using Gas sensor and trigger an alarm when the air quality goes down beyond a certain level, means when there is

sufficient amount of harmful gases are present in the air like CO2, smoke, alcohol, benzene and NH3. It is shown the air quality in PPM on the LCD as well as serial monitor so that I can monitor it very easily [9]. I have used MQ135 sensor which is the best choice for monitoring Air Quality as it can detect most harmful gases and can measure their amount accurately. In this Internet of Things (IoT) project, I can monitor the pollution level from anywhere using computer.

4.5 Algorithm & Working Process

I have connected the MQ135 gas sensor and ESP8266 Wi-Fi device with the Arduino. Connected 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. Connected a buzzer to the pin 7 of the Arduino which is start to beep when the condition becomes true. The MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and some other gases, so it is faultless gas sensor for our Air Quality Observing Detection Project. When I connect it to Arduino then it senses the gases, and I get the Pollution level in PPM (parts per million). MQ135 gas sensor gives the output in form of voltage levels and I need to convert it into PPM. Sensor is giving us value of 0.1 when there is no gas near it and the safe level of air quality is 0.5 PPM and it is not exceeding 0.5 PPM. When it exceeds the limit of 0.5 PPM, then it starts cause Headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 1 PPM then it can cause increased heart rate and many other diseases. When the value is being less than 0.5 PPM, then the LCD and serial monitor is displayed “Fresh Air”. Whenever the value is increased 0.5 PPM, then serial monitor is displayed “Poor Air, Open Windows”. If it is increased 1 PPM, then the buzzer is kept beeping and the LCD is displayed “Danger! Move to fresh Air”. After uploading the code, I am connected to the Wi-Fi of my ESP8266 device, the serial monitor has opened and it is showing the IP address like shown below (192.168.43.57). If I have typed mentioned IP address in my browser, it is shown the output as below. I have to refresh the page again if I want to see the current Air Quality Value in PPM. After uploading code, the value is being less than 0.5 PPM, then the LCD and Web Browser is displayed “Fresh Air”. After uploading code, the value is increased 0.5 PPM, then the LCD and web browser are displayed “Poor Air, Open Windows”. After uploading code, When the value is increased 1.00 PPM then the buzzer is kept beeping and the LCD and Web Browser are displayed “Danger! Move to fresh Air”

4.6 Working Principle of Proposed Model

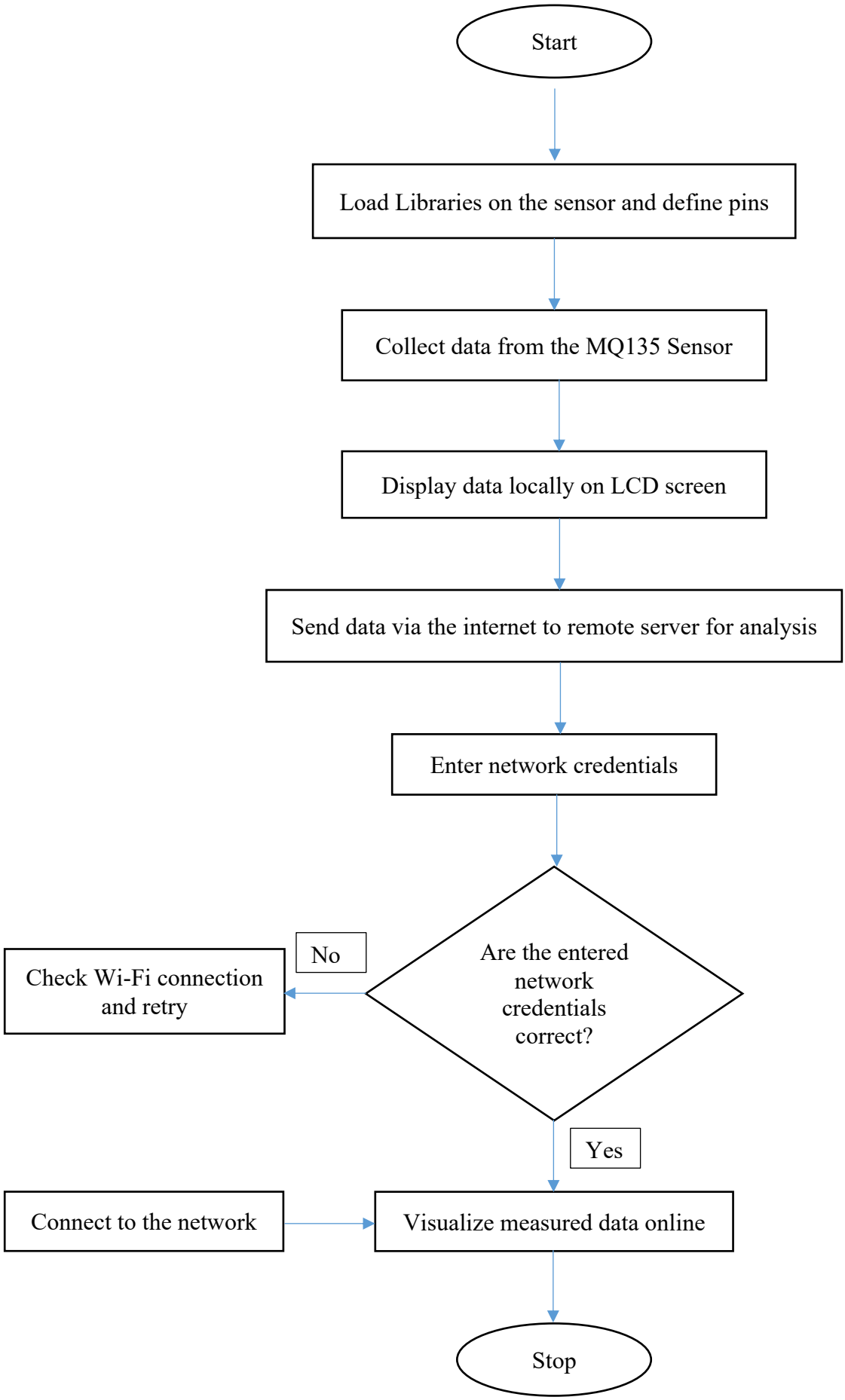
As described by Figure 3, the library in the Arduino was loaded and a message was sent to the LCD. Air quality data was collected using the MQ135 sensor. The calibrated sensor made the analog output voltage proportional to the concentration of polluting gases in Parts per Million (ppm). The data is first displayed on the LCD screen and then sent to the Wi-Fi module. The Wi-Fi module transfers the measured data valve to the server via internet. The Wi-Fi module is configured to transfer measured data an application on a remote server called “Thing speak”. The online application provides global access to measured data via any device that has internet connection capabilities. Data collected from the sensor was converted into a string and used to update the information sent to the remote server.

4.7 Methodology

The model was designed using an Arduino Uno microcontroller, Wi-Fi module 8266, MQ135 Gas Sensor and a 16 by 2 liquid crystal display (LCD) Screen. Figure 1 shows the proposed system overview and the functional block diagram is depicted in figure 2. The proposed flow chart is presented in figure 3.

The system overview procedure was classified into Five (5) layers as shown in figure 1. The first layer was the environmental parameters which are obtained by measurement. The second layer was the study of the characteristics and features of the sensors. The third layer was the decision making, sensing, measuring, fixing of the threshold valve, periodicity of sensitivity, timing and space. The fourth layer was the sensor data acquisition. The fifth layer was the ambient intelligence environment. The sensor collected data when operated by the microcontroller and forwarded it over the internet for analysis via the Wi-Fi module. Users were able to monitor measured

Fig 4.3: Flowchart for the Proposed System



4.8 Mathematical Analysis of Proposed Model

The level concentration of pollutants in the air is measured in parts per million (ppm) or percentage.

Conversion factors include the following: 1 ppm = 1.145 mg/m3

1 mg/m3 = 0.873 ppm

1% = 1/100

1 ppm = 1/1000000

1 ppm = 0.0001%

Table 2 shows PPM to percentage conversion.

Parts per Million (ppm)	Percent (%)
0	0
5	0.005
50	0.005
500	0.05
1000	0.1

Table 4.2: PPM to Percentage conversion

CHAPTER 5:

Implementation

5.1 Code

Code to obtain ppm value from MQ-135 Sensor on ThingSpeak:

```
#include <ESP8266WiFi.h>
#include <LiquidCrystal.h>

// NodeMCU LCD pin
const int RS = 4, EN = 0, d4 = 12 , d5 = 13, d6 = 15, d7 =
3;

LiquidCrystal lcd(RS, EN, d4, d5, d6, d7);

const int aqsensor = A0;

//int rled = D7;  //red led
//int buz = D8;   //buzzer

String apiKey = "ZLLUAM4PYH99Z3AJ";     // Write API key
const char *ssid =  "Kars";     //  WiFi Name
const char *pass =  "c503@1234"; // WiFi Password
const char* server = "api.thingspeak.com";
WiFiClient client;

void setup()
{
  // put your setup code here, to run once:

  pinMode (aqsensor,INPUT); // MQ135 is connected as INPUT
to ESP-8266
```

```
  Serial.begin (115200);      //baud rate

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

  Serial.println("Connecting to ");
  lcd.print("Connecting.... ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
    lcd.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP Address: ");
  delay(5000);
  Serial.println(WiFi.localIP());
  delay(5000);
}

void loop()
{
  // put your main code here, to run repeatedly:

    int ppm = analogRead(aqsensor);
    int CO = 0.0407 * (ppm) ;

  Serial.print("Air Quality: ");
  Serial.println(ppm);
  Serial.println(" Carbon Monoxide: ");
  Serial.println(CO);
```

```
  lcd.setCursor(0,0);
  lcd.print("Air Quality: ");
  lcd.print(ppm);
  delay(1000);

  if (client.connect(server,80))
                        {
                                String postStr = apiKey;
                                postStr +="&field1=";
                                postStr += String(ppm);
                                postStr += "\r\n\r\n";

                                client.print("POST     /update
HTTP/1.1\n");
                                client.print("Host:
api.thingspeak.com\n");
                                client.print("Connection:
close\n");
                                client.print("X-
THINGSPEAKAPIKEY: "+apiKey+"\n");
                                client.print("Content-Type:
application/x-www-form-urlencoded\n");
                                client.print("Content-Length:
");

client.print(postStr.length());
                                client.print("\n\n");
                                client.print(postStr);
                                Serial.print("Air Quality: ");
                                Serial.print(ppm);
                                Serial.println("  PPM.Send  to
Thingspeak.");
                          }
  if(ppm <= 130)
```

```
  {
    lcd.setCursor(1,1);
    lcd.print ("AQ Level Normal");
    Serial.println("AQ Level Normal");
  }
  else if (ppm > 130 && ppm < 250)
  {
    lcd.setCursor(1,1);
    lcd.print ("AQ Level Medium");
    Serial.println("AQ Level Medium");
  }
  else
  {
    lcd.setCursor(1,1);
    lcd.print("AQ Level Danger!");
    Serial.println("AQ Level Danger!");
  }

          client.stop();
          Serial.println("Waiting...");
          delay(1000);

}
```

Code to obtain ppm value from MQ-135 Sensor on Webpage:

```
#include "ESP8266WiFi.h"
#include <LiquidCrystal.h>  // include the library code:


// initialize the library by associating any needed LCD
interface pin
// NodeMCU 12E the suggested pins
const int RS = 4, EN = 0, d4 = 12 , d5 = 13, d6 = 15, d7 =
3;
```

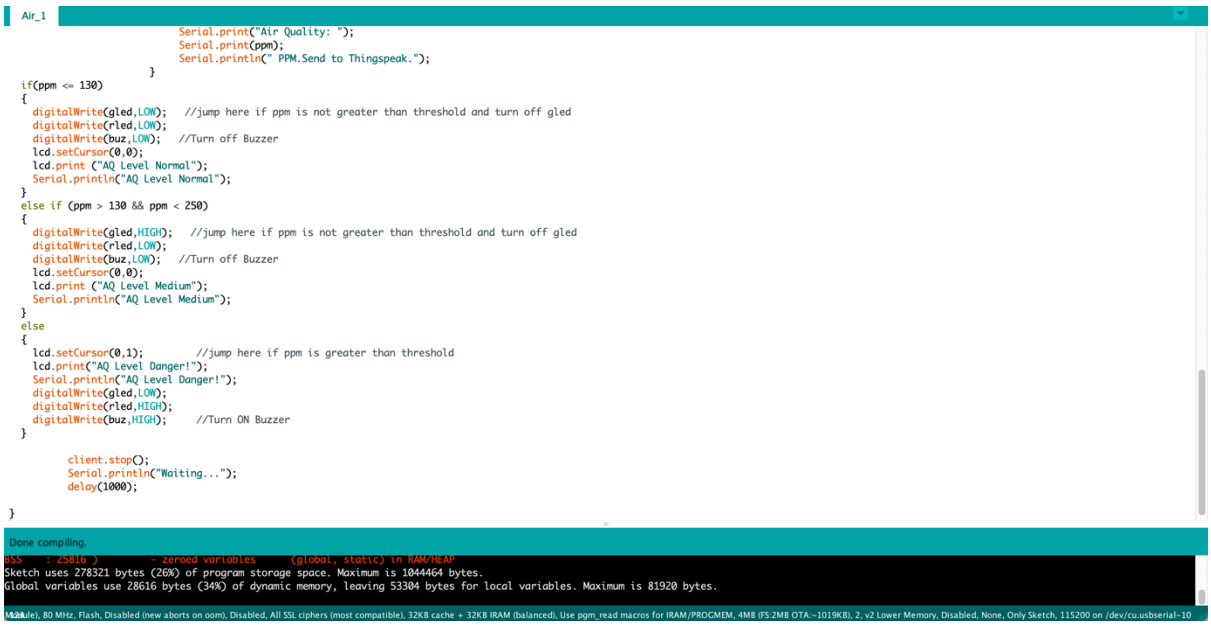
```
LiquidCrystal lcd(RS, EN, d4, d5, d6, d7);

const int aqsensor = A0;

int gled = D6;
int rled = D7;
int buz = D8;

const char *ssid = "Kars"; // WiFi Name
const char *pass = "c503@1234"; // WiFi Password

WiFiServer server(80);

void setup()
{
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Setup done");

  pinMode (aqsensor,INPUT);   // MQ135 is connected as INPUT
to ESP8266
  pinMode (rled,OUTPUT);      // rled is connected as output
from ESP8266
  pinMode (buz,OUTPUT);       // buzz is connected as output
from ESP8266

  Serial.begin (115200);

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

  Serial.println("Connecting to ");
```

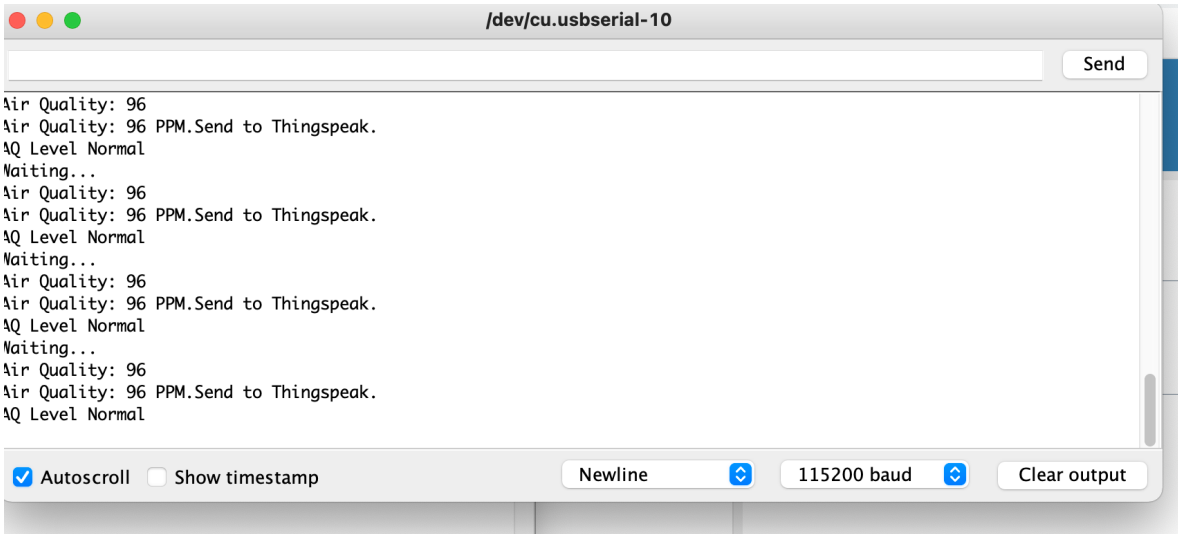
```
  lcd.print("Connecting.... ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
    lcd.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP Address: ");
  delay(5000);
  Serial.println(WiFi.localIP());
  delay(5000);
  server.begin();
}

void loop()
{
  int ppm = analogRead(aqsensor);

  Serial.print("Air Quality: ");
  Serial.println(ppm);

  lcd.setCursor(0,0);
  lcd.print("Air Quality: ");
  lcd.print(ppm);
  delay(1000);

  WiFiClient client = server.available();
  if ( client.connected())
  {
  client.println("HTTP/1.1 200 OK");
  client.println("Content-type:text/html");
```

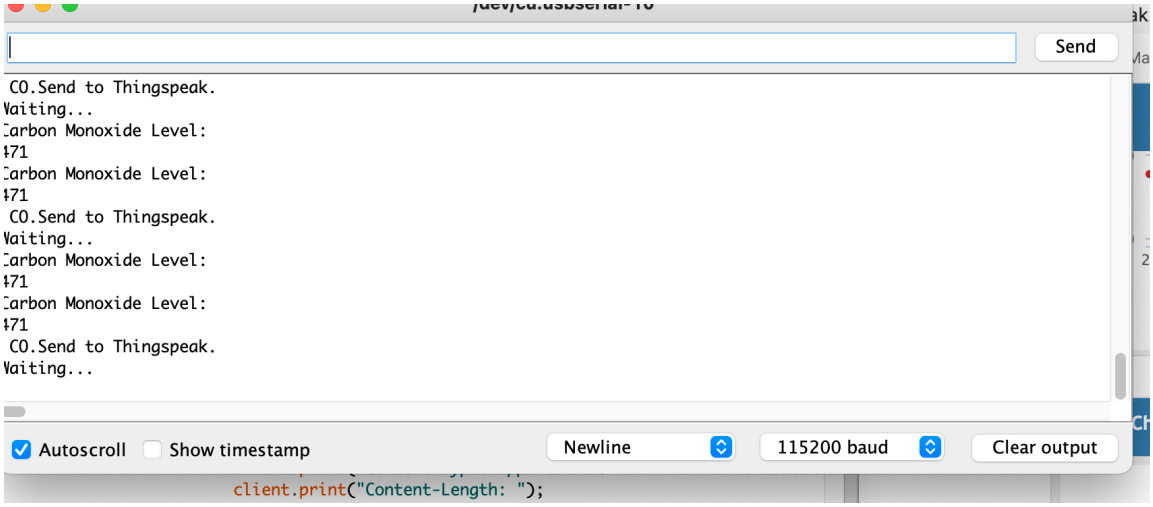
```
  client.println("Connection: close");
  client.println("Refresh: 3");
  client.println();

  // Display the HTML web page
  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:,\">");

  // Web Page Heading
  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>");
    digitalWrite(gled,LOW);
    digitalWrite(rled,LOW);
    digitalWrite(buz,LOW);
    lcd.setCursor(1,1);
    lcd.print ("AQ Level Normal");
    Serial.println("AQ Level Normal");
  }
  else if (ppm > 130 && ppm < 250)
  {
    client.println("<body><p       style=\"color:purple;\">
Medium </p>");
    digitalWrite(gled,HIGH);
    digitalWrite(rled,LOW);
    digitalWrite(buz,LOW);
    lcd.setCursor(1,1);
```

```
      lcd.print ("AQ Level Medium");
      Serial.println("AQ Level Medium");
    }
    else
    {
      client.println("<body><p        style=\"font-size:200%;
  color:red\"> Danger!!! </p>");
      lcd.setCursor(1,1);
      lcd.print("AQ Level Danger!");
      Serial.println("AQ Level Danger!");
      digitalWrite(gled,LOW);
      digitalWrite(rled,HIGH);
      digitalWrite(buz,HIGH);
    }

    client.println("</body></html>");
    delay(500);

  }
  }
```

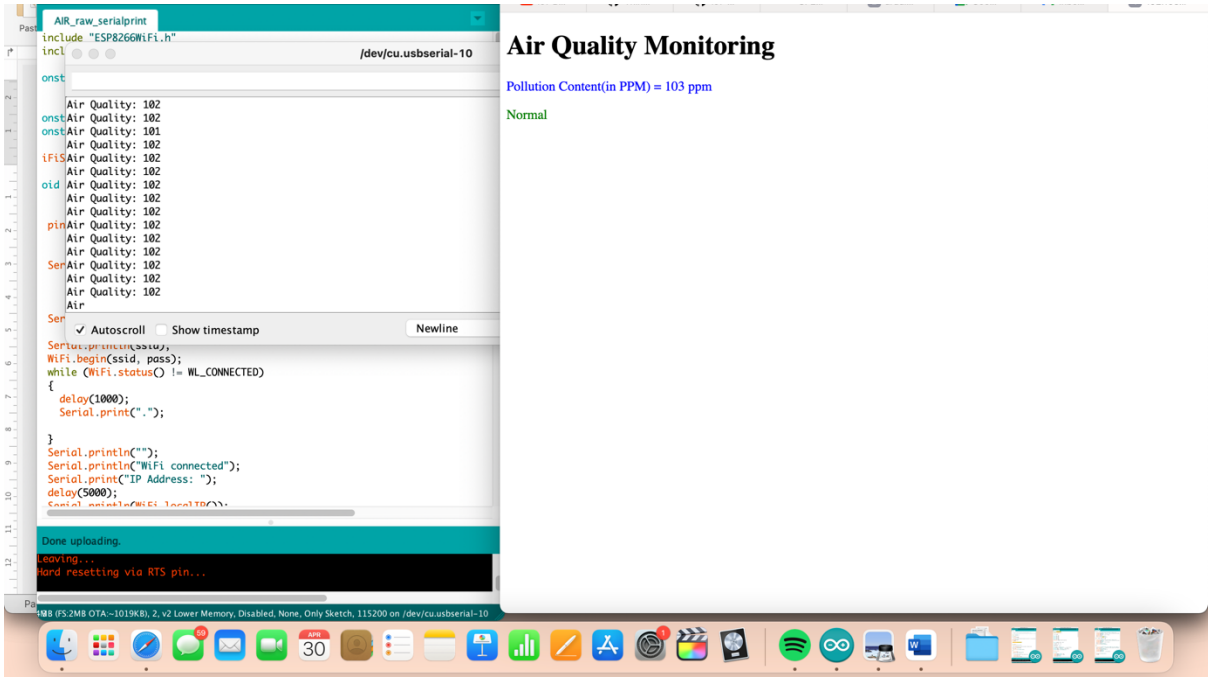
Serial Monitor Output:



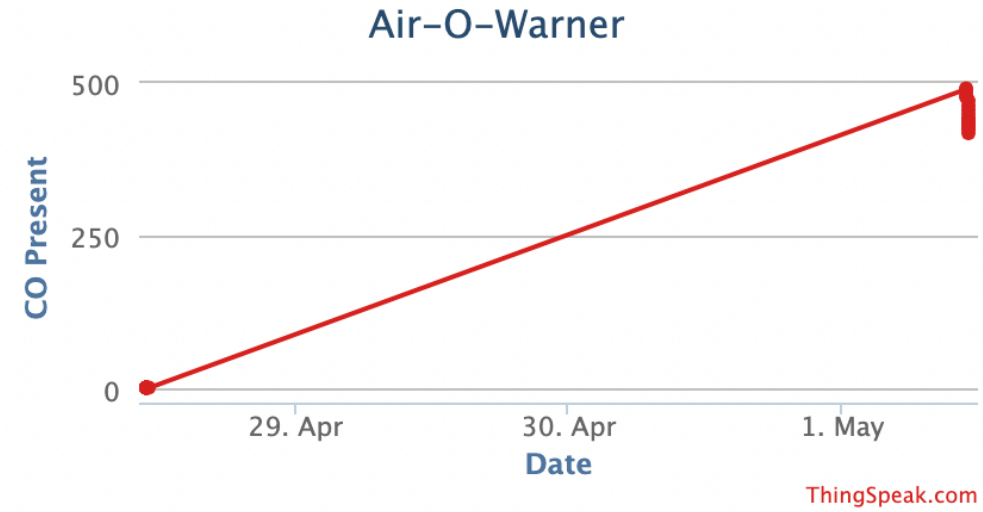


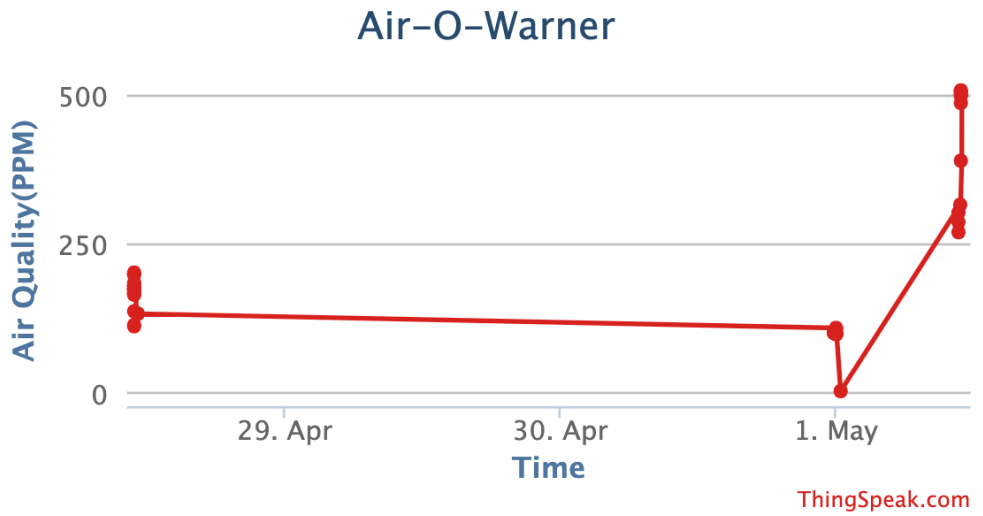


Webpage Display:



Graph on ThingSpeak





5.2 IOT Interfacing

The system to monitor the air of environment using Arduino microcontroller, IOT Technology is proposed to improve quality of air. The use of IoT technology enhances the process of monitoring various aspects of the environment such as the air quality monitoring issue proposed in this paper. Here, using the MQ135 and MQ6 gas sensors gives the sense of the different types of dangerous gas and Arduino is the heart of this project.

Which controls the entire process. Wi-Fi module connects the whole process to the internet and LCD is used for the visual Output.

ThingsBoard

To send data to the ThingsBoard, we will use Arduino SDK that has all high-level functions for this. To do this we have to define device token and ThingsBoard host:

If you have ThingsBoard up and running and corresponding to use case device entity exists:

• GotoDeviceGroupsofthedeviceowner,selectAll.
• Click on the device row in the table to open device details
• Click "Copy access token". The Token will be copied to your clipboard.

1. If using a PaaS subscription, Log in or Sign Up to the thingsboard.cloud and perform previous steps.

2. If you do not have any device entity so far, Create a new device:

a. At the left navigation bar click on Device groups > All

b. At the right top menu click on the + button

c. Fill in all inputs in the form

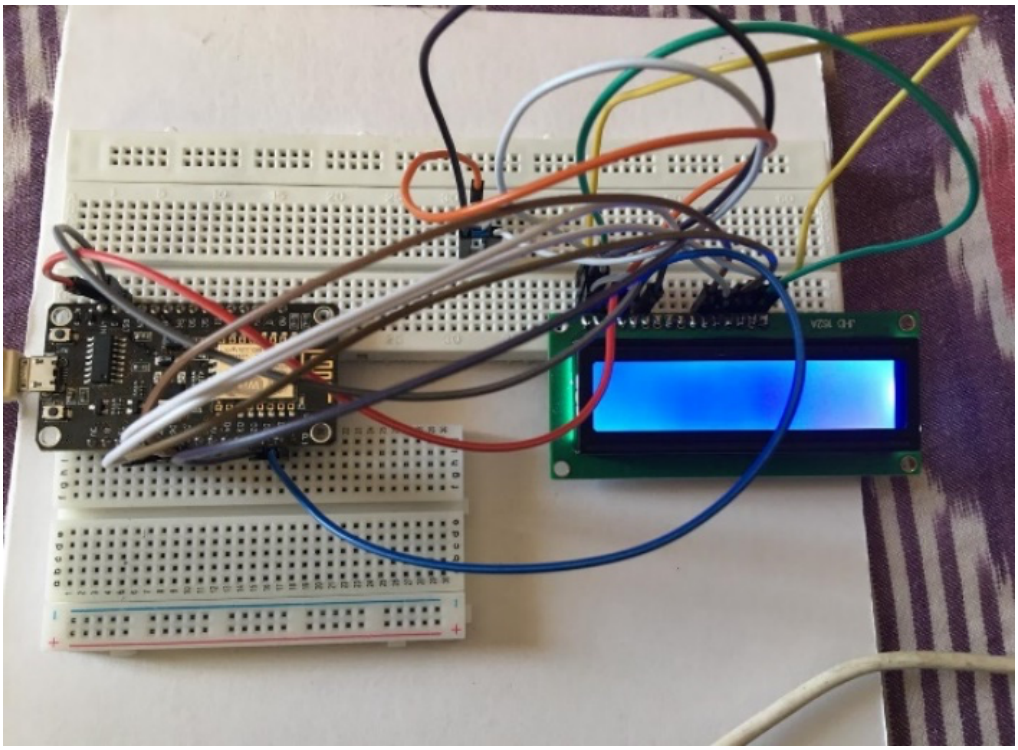
d. Click Add button

3. In the device list click on the created device 4. Click on Copy access token

5. Paste token to the global variable TOKEN

To send our decimal value from the sensor to ThingsBoard we should use sendTelemetryFloat() function in the main loop:

5.3 Output



Connection

Fig 5.1: Connection and display

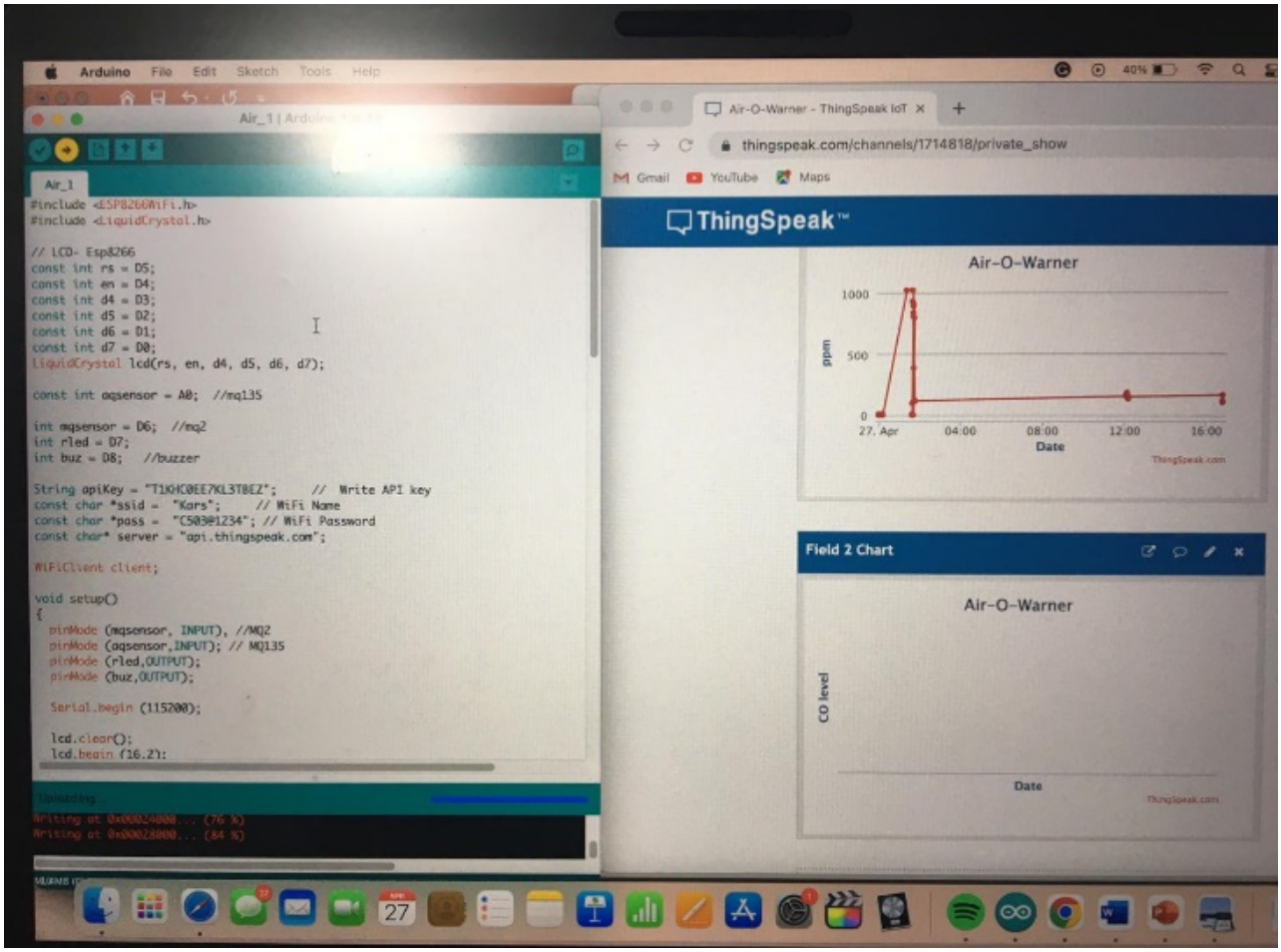


Fig 5.2: Output on Thingspeak in real time

CHAPTER 6

Conclusion

6.1 Conclusion

The system to monitor the air of environment using Arduino microcontroller, IoT Technology is proposed to improve quality of air. With the use of IoT technology enhances the process of monitoring various aspects of environment such as air quality monitoring issue proposed in this paper. Here, using the MQ135 gives the sense of different type of dangerous gas and Arduino is the heart of this project. Which control the entire process, Arduino module connects the whole process to LCD and serial monitor is used for the visual Output.

6.2 Real Time Applications:

- The readings collected by the sensors will be displayed on an interactive website.
- This will help the common people monitor the levels of air pollution in their neighborhood.
- The Trained Model can be used to gain insights on location specific future air quality levels.

The government can also use the data to implement dynamic pollution control measures. (Eg: Like Delhi's odd-even rule but over a localized area and on particular days rather than over weeks)

6.3 Applications:-

1) Industrial perimeter monitoring
2) Indoor air quality monitoring.
3) Site selection for reference monitoring stations. 4) Making data available to users.

6.4 Advantages:-

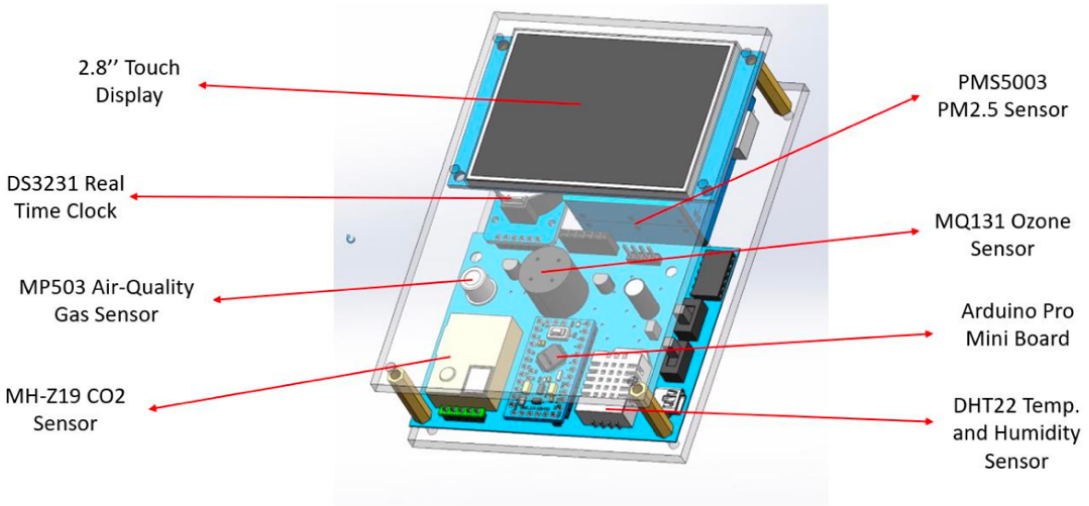
1) Easy to Install

2) Updates On mobile phone directly 3) Accurate Pollution monitoring

4) Remote location monitoring

6.5 Future Scope:

- The sensors can be miniaturized and integrated into smartphones.
- As a result, millions of datapoints can be generated for thousands of locations around the world.
- This can help to create better predictive models to estimate future patterns of air pollution across the world.
- With this information, governments can take more concrete steps to reduce air pollution like constructing Carbon Capture Systems (CCS) and rezoning industries in highly polluted regions.



The following CAD Design Model shows the envisioned miniaturized solution of the Air pollution monitoring system which is compact as well as portable.

6.6 REFERENCES:-

https://datasheetspdf.com/pdf/605076/Hanwei/MQ-135/1

https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf

https://www.pololu.com/file/0J309/MQ2.pdf

https://www.electronicoscaldas.com/datasheet/DHT11_Aosong.pdf

https://arduinojson.org/v6/doc/ https://www.mathworks.com/help/thingspeak/

https://how2electronics.com/iot-air-pollution-monitoring-esp8266/