



**BACHELOR OF SCIENCE IN ELECTRONIC AND
TELECOMMUNICATION ENGINEERING**

**Development of an Iot based air pollution
and weather monitoring system with
android application**

Submitted by

Md Moinul Islam

T-163012

Muhammad Whahedul Alam

T-161031

Supervised by

Abdul Gafur

Associate professor

Department of ETE

International Islamic University Chittagong

Department of Electronic and Telecommunications Engineering

International Islamic University Chittagong

Kumira, Sitakunda, Chattogram

September, 2021

DEDICATION

This thesis work is dedicated to all of our honourable teachers and our beloved parents for their prayers and support towards the achievement of our goal.

CERTIFICATE OF APPROVAL

The project entitled as “Development of an Iot based air pollution and weather monitoring system with android application” submitted by Md. Moinul Islam bearing ID No: T-163012, Muhammad Wahedul Alam bearing ID No: T-161031, to the Department of Electronic and Telecommunications Engineering (ETE) of International Islamic University Chittagong (IIUC) has been accepted as satisfactory to partially fulfill the bachelor's degree criteria in Electronic and Telecommunications Engineering and approved as to its style and contents for the examination held on Spring 2021.

Approved by:

Abdul Gafur

Associate professor

Department of Electronics and Telecommunications Engineering

Faculty of Science and Engineering International Islamic University Chittagong Kumira,
Sitakunda, Chittagong.

CANDIDATES DECLARATION

The work presented in this thesis is hereby stated not to have been submitted elsewhere for the award of any degree or diploma, and does not contain any illegitimate declaration.



Md. Moinul Islam

T163012



Muhammad Wahedul Alam

T161031

ACKNOWLEDGEMENT

The most generous and the most merciful, in the name of Almighty Allah. We are first and foremost thankful to Allah Almighty, Who gave us the power and the intelligence to think. We thank our Honorable Supervisor Engr. Abdul Gafur, Associate Professor, Department of Electronic and Telecommunications Engineering International Islamic University Chittagong who gave us considerable recommendations and inspections during the project. He has contributed to the success of this project by making useful ideas during the experiment and project work. We also thank our family, instructors, friends, classmates, seniors and staff, who provided us with endless help and support in all respects. We're glad to have a wonderful person like you next to us.

Table of Contents

DEDICATION	ii
CERTIFICATE OF APPROVAL.....	iii
CANDIDATES DECLARATION.....	iv
ACKNOWLEDGEMENT	v
LIST OF FIGURES	viii
ABSTRACT.....	xii
List of Abbreviation.....	xiii
CHAPTER 1 INTRODUCTION	1
1.1 Motivation.....	2
1.2 Objective	2
1.3 Thesis Outline.....	3
CHAPTER 2 LITERATURE REVIEW	3
CHAPTER 3 METHODOLOGY	5
3.1 Proposed Model block diagram	5
3.2 Flow Chart of Proposed System	6
3.3 Circuit Diagram of Proposed System	7
3.4 Proposed Model in Real Life	9
3.5 Components of the Device.....	10
3.5.1 Arduino Uno	10
3.5.2 ESP 8266 Wi-Fi Module	11
3.5.3 DHT 11 Temperature and Humidity sensor	12
3.5.4 MQ-7 CO Gas sensor	13
3.5.5 M135 Air quality sensor module.....	14
3.5.6 MQ-2 LPG gas sensor	15
3.5.7 Liquid-crystal display 16×2.....	15
3.5.8 I2C Module.....	18
3.5.9 Relay.....	19
3.5.10 Problem Formulation and Decision	20
3.5.11 Dataset	21
3.5.12 Summary	22
CHAPTER 4 RESULT AND DISCUSSION	23

4.1 Introduction	23
4.2 Dataset Description.....	23
4.2.1 Displaying data in LCD Display	23
4.2.2 Sending data to the cloud platform	37
4.2.3 Cost Estimation Structure	45
CHAPTER 5 CONCLUSION AND FUTURE WORKS	46
5.1 Future work.....	46
5.2 Advantages.....	46
5.3 Applications.....	46
5.4 Conclusion	47
References	48
APPENDIX.....	50

LIST OF FIGURES

Fig. 3.1. Proposed model of the system	5
Figure 3.2: Block diagram of the Proposed system	5
Figure 3.3: Flow Chart of Proposed System	6
Fig. 3.4. Circuit diagram for proposed model of the system	7
Figure 3.5: Implementation schema of an Iot based air pollution and weather monitoring system	8
Fig. 3.5. Proposed Model in Real Life.....	9
Fig. 3.6. Arduino Uno	10
Figure 3.7: ESP 8266 Wi-Fi Module	12
Figure 3.8: DHT 11 Temperature and Humidity sensor	13
Figure 3.9: MQ-7 CO Gas sensor	13
Figure 3.10: M135 Air quality sensor module.....	14
Figure 3.11: MQ-2 gas sensor.....	15
Figure 3.12: Liquid-crystal display 16×2.....	16
Figure 3.13: LCD-16×2-pin-diagram	17
Figure 3.14: I2C Module for 16×2 Character LCD	18
Figure 3.15: 5V DC Relay	19
Figure 4.1 : LCD display view Temperature & Humidity(1st place).....	23
Figure 4.1.2: LCD display view CO (1st place)	24
Figure 4.1.3: LCD display view LPG (1st place)	24
Figure 4.1.4: LCD display view Air quality (1st place)	24
Figure 4.1.5: LCD display view LPG(2nd place).....	25
Figure 4.1.6: LCD display view CO(2nd place).....	25
Figure 4.1.7: LCD display view Air quality(2nd place)	25
Figure 4.1.8: LCD display view Temperature & Humidity(2nd place).....	26
Figure 4.1.9: LCD display view Air quality(3rd place).....	26
Figure 4.1.10: LCD display view CO(3rd place).....	26
Figure 4.2.1: LCD display view Temperature & Humidity (3rd place)	27
Figure 4.2.2: LCD display view LPG(3rd place).....	27
Figure 4.2.3: LCD display view LPG(4th place).....	27

Figure 4.2.4: LCD display view Temperature & Humidity (4th place).....	28
Figure 4.2.5: LCD display view CO (4th place).....	28
Figure 4.2.6: LCD display view Air quality(4th place)	28
Figure 4.2.7: LCD display view CO(5th place).....	29
Figure 4.2.8: LCD display view Air quality(5th place)	29
Figure 4.2.9: LCD display view LPG(5th place).....	29
Figure 4.2.10: LCD display view Temperature & Humidity (5th place).....	30
Figure 4.3.1: LCD display view CO(6th place).....	30
Figure 4.3.2: LCD display view Air quality(6th place)	30
Figure 4.3.3: LCD display view Temperature & Humidity (6th place).....	31
Figure 4.3.4: LCD display view LPG(6th place)	31
Figure 4.3.5: LCD display view Temperature & Humidity (7th place).....	31
Figure 4.3.6: LCD display view LPG(7th place).....	32
Figure 4.3.7: LCD display view CO(7th place).....	32
Figure 4.3.8: LCD display view Air quality(7th place)	32
Figure 4.3.9: LCD display view Temperature & Humidity (8th place).....	33
Figure 4.3.10: LCD display view Air quality(8th place)	33
Figure 4.5.1: LCD display view CO(8th place).....	33
Figure 4.5.2: LCD display view LPG(8th place).....	34
Figure 4.5.3: LCD display view CO(9th place).....	34
Figure 4.5.4: LCD display view Air quality(9th place)	34
Figure 4.5.5: LCD display view Temperature & Humidity (9th place).....	35
Figure 4.5.6: LCD display view LPG(9th place).....	35
Figure 4.5.7: LCD display view Air quality(10th place)	35
Figure 4.5.8: LCD display view Temperature & Humidity (10th place).....	36
Figure 4.5.9: LCD display view LPG(10th place).....	36
Figure 4.5.10: LCD display view CO(10th place).....	36
Figure 4.6.1: output of the device in the cloud platform(Air quality)	37
Figure 4.6.2: output of the device in the cloud platform(CO)	37
Figure 4.6.3: output of the device in the cloud platform(LPG)	38
Figure 4.6.4: output of the device in the cloud platform(Temperature)	38

Figure 4.6.5: output of the device in the cloud platform(Humidity)	39
Figure 4.7.1: Android App Data from first place.....	39
Figure 4.7.2: Android App Data from 2nd place	40
Figure 4.7.3: Android App Data from 3rd place.....	40
Figure 4.7.4: Android App Data from 4th place	41
Figure 4.7.5: Android App Data from 5th place.....	41
Figure 4.7.6: Android App Data from 6th place	42
Figure 4.7.7: Android App Data from 7th place.....	42
Figure 4.7.8: Android App Data from 8th place	43
Figure 4.7.9: Android App Data from 9th place.....	43
Figure 4.7.10: Android App Data from 10th place	44

LIST OF TABLE

Table 1 Arduino Uno Technical Specifications.....	11
Table 2 AQI scale for CO	20
Table 3 US Embassy Dataset.....	21
Table 4 Second Dataset of our device.....	22

ABSTRACT

Air pollution is one of the biggest environmental problems that affects the health, ecology, and environment and produces many severe impacts on human health. Vehicles are the biggest cause of air pollution in all large cities and the second main source of pollution is industry. Vehicle mass usage has led to a significant increase in atmospheric pollutants. This is the source of pollution in the human health. Other breathing issues such as asthma episodes and skin rashes have also occurred. The contaminants that pollute the air are unseen, leading to people's carelessness. Public recognition is therefore today's main requirement. The developments in the internet of things (IoT) allowed the quantity of harmful compounds in air particles to be monitored and air quality controlled. The goal is to use the MQ135 sensor to build a system that can detect the quantity of harmful particles in the air. A dedicated server can be used to manage air quality through the internet. When the amount of toxic particles in the air, such as CO (Carbon Monoxide) an exhaust fan will automatically switched on and remove harmful air. Air quality in PPM (Parts Per million) is displayed on the web server when the toxicity in the air crosses the threshold value and automatically air purifier gets switched on, and it detoxifies the air. This device also able to detect LPG gas by the help of MQ-2 sensore and this device also detect temperature and humidity by the help of DHT-22 sensor . In this device we've added a dummy exhaust fan that will turn on automatically when the air quality cross the limit of moderate air quality. The device may be put anywhere, and some external devices can be activated if pollution passes the safe area such as the Android application can send a notification to all users. Our effort is designed to provide indoor air quality monitoring and analysis with a low cost budget system in real time. Our project work is important for all general users who can easily achieve the level of indoor air quality and can thoroughly monitor indoor air quality [1].

List of Abbreviation

- LPG: Liquefied Petroleum Gas
- SDG: Sustainable Development Goals
- UN: United Nations
- AQI: Air Quality Index
- AQHI: Air Quality Health Index
- GPRS: General Packet Radio Services
- GUI: Graphical User Interface
- GPIO: General-Purpose input/output
- PCB: Printed circuit board
- HTML: Hypertext Markup Language
- PHP: Personal Home Page
- API: Application Programming Interface
- LDAP: Lightweight Directory Access Protocol
- IMAP: Internet Message Access Protocol
- SNMP: Simple Network Management Protocol
- NNTP: Network News Transfer Protocol
- HTTP: Hypertext Transfer Protocol
- PPM: Parts Per Million

CHAPTER 1

INTRODUCTION

In light of the daily newspapers and other electronic or print media, a horrible news spreads every day that causes people to get sick and the climate is changing so that people live in misery. Everyone is suffering from the climate change curse from top to bottom aspect. Air pollution is the major cause of climate change and people's health. Climate changes including global warming, world dimming, rainfall, drought, storm, acid rain, foggy weather, etc. The living on Earth and beneath water have numerous challenges, such life changes, owing to a lack of suitable living necessities. Air for every living creature is the most useful thing [2]. The major objective of this method was to evaluate air quality for people and for every living creature on earth on this critical issue. For our lives it is very essential to know how safe we are and how the environment and weather have changed in order to pollute the air and it will be sound. This method makes it easy to know air quality responses. Some significant gas sensors are mostly in use throughout the system to know the ideal outcome of the entire air condition. CO, Humidity has been declared the main cause of air pollution and all are employed in the system [3]. In order to monitor the LPG gas presence in the surroundings the MQ-2 sensing is also installed. In this project, the Arduino UNO is based on air pollution and weather monitoring devices. The sensor data is also displayed on a character LCD interfaced in the monitoring IOT device. The statistics were made known via a server and android application, because currently, almost everyone has an android device and Internet connection.

1.1 Motivation

Not one living creature without air can live. Air is the most vital part of life. According to the study, 50,000 to 100,000 early deaths each year in the United States are caused by air pollution alone. In the EU it is 300,000 and more than 3,000,000 in the world. Many people become sick because of air pollution alone. In metropolitan regions there are the primary sources of air pollution, the growing number of petrol and diesel cars, and the development of industrial zones on the outskirts of large cities [4]. In the big cities the situation is significantly worsened. Climate change is now visible . According to the UN (United Nations) Sustainable Development Goals (SDG), there are seventeen aim to change the globe into a clean, healthy and natural way of life. The SDG states that it ensures a healthy life and promotes the access to affordable, durable and modern energy for all ages for sustainable consumption and production patterns Take immediate action against and effect on climate change Sustainably protecting and using marine waters and resources, sustainable forest management, fighting desertification, stopping and reversal land degradation, stopping losses to biodiversity. This is the major reason why the subject was selected to be researched. [5]

1.2 Objective

To design and develop an Android Application IoT-based air pollution and weather monitoring system The following task is carried out by the implemented project.

- 1) Combine advanced air-quality detecting technology with upgraded capacities to give comprehensive low cost monitoring.
- 2) The temperature and humidity of the environment are measured and shown.
- 3) Check air quality on the web using a dedicated server.
- 4) Show detected data in an easy-to-use Android app.

1.3 Thesis Outline

Chapter 1: Includes introduction, motivation and thesis outline.

Chapter 2: Describes the previous studies done on air pollution

Chapter 3: Describes the methodology and a summary of the past studies on air pollution components of the proposed model.

Chapter 4: Describes result and discussion.

Chapter 5: Concludes the report on the problems identified and the project's next development.

CHAPTER 2 LITERATURE REVIEW

Every day the incidence of pollution increases alarmingly. Air is the environment's most delicate element, which is temporarily contaminated by airborne pollutants. To understand the air pollution level and air quality this system is presented as a wireless sensor network, which primarily monitors pollution in air [6]. This is a low-cost, yet efficient sensor monitoring system. In 2015, several earlier projects like the Smart car environment surveillance system [2] were launched. It essentially found the rate of emissions of toxic gasses that pollute the air. In order to know the dangerous gases and their influence, industrial air pollution [2] safety monitoring system and health enhancement was created. The system was discussed at low-cost air quality in 2008 since the sensors were very costly and the system too. Air pollution might be detected through the use of mobile GPRS [5] systems. In urban cities the wireless sensors network-based surveillance systems for air quality were implemented. Even earlier Pollution Dynamic Monitoring System is used. In looking at previous studies, we can state that air pollution is increasing at an alarming rate. The entire planet will experience terrible weather for the future if it is not halted right now. We can state from our previous studies, that air pollution is the most worrying issue and this should be investigated in order to save the planet [1]. There are other pollutants such as water pollution, noise pollution, plastics and Pollution of soil. According to the World Health Organization, air

pollution poses a serious danger to health and the climate from haze above cities to smoking within the house. Ambient (outdoor) and home air pollution have combined impacts leading to over 7 million premature deaths each year, mainly because of higher levels of stroke, heart conditions, chronic obstructive lung disease, lung cancer and acute breathing infections. In urban areas which monitor air pollution, more than 80% are exposed to air qualities which exceed the WHO $10\mu\text{g}/\text{m}^3$ limit with low and medium income nations with most exposures. Vehicles, energy generation, building heating systems, agriculture/waste incineration, industry are among the main causes of outside pollution. More than 3 billion people worldwide also rely on residential cooking, lighting, smoke releases and the outdoor discharge of pollutants (including biomass, coal, and kerosene) for the use of harmful technologies and fuels [7].

Worldwide air is contaminated by 9 out of 10 people. Green energy should be introduced to avoid air pollution. WHO estimates that the pollution of the environment alone caused around 4.2 million deaths in 2016, and that domestic air pollution caused 3.8 billion deaths over the same time through cooking harmful fuels and processes. The goal was thus to create such a system that let individuals know what harmful air is being breathed [8]. This system comprises study into the importance of working on this issue from earlier investigations. The principal goal was to make such a gadget portable and easy to install. For simplicity of use, both a website and Android app can view the results of the device. By monitoring air, water and sound pollution in daily life, the measurement of the pollution level would be of enormous importance to humans' health. Different types of pollution surveillance gas sensors will be installed at various places around the city to detect air pollution [9] [10]. The contaminated region and the area containing dangerous particles to humans will be the main priority. The sensors will collect realistic data on different gasses e.g. Carbon Monoxide (CO), LPG, Temperature and humidity from diverse regions of the air. The method that has been presented allows mainly air quality monitoring and pollution of a smart city on a desktop/laptop computer, utilizing a programming software developed using graphical user interface [10].

CHAPTER 3 METHODOLOGY

The Proposed model of the system is as follows . On Figure 3, we can see a diagram of the system's operation. Using the system's block diagram, we may see how the system will work in a given location. The device will be set up to capture environmental data, and there will be a base standard value. The device will collect data and, based on the given values, display the output [13].



Fig. 3.1. Proposed model of the system

3.1 Proposed Model block diagram

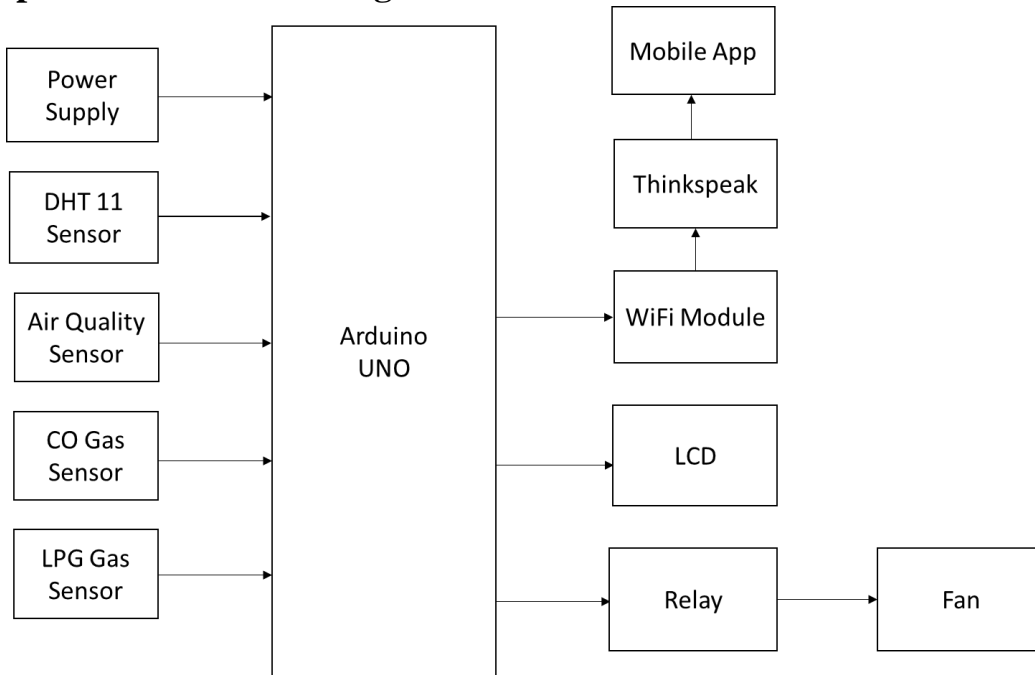


Figure 3.2: Block diagram of the Proposed system

The block diagram of proposed model of the system is shown in Fig.3.2. Here four sensors- air quality sensor (MQ 135), temperature and humidity sensor (DHT 22), CO gas sensor (MQ 7) and LPG gas sensor (MQ-2) is connected to microcontroller (Arduino Uno) as input and LCD display is connected to see the output of the project. A Wi-Fi module (ESP 8266) is also connected through which we can see the result in think speak web server and we can also see the result in our android application. In this project we have added a dummy exhaust fan which will turn on automatically if the surrounding air cross the limit of moderate air quality and then to get the polluted air out [12].

3.2 Flow Chart of Proposed System

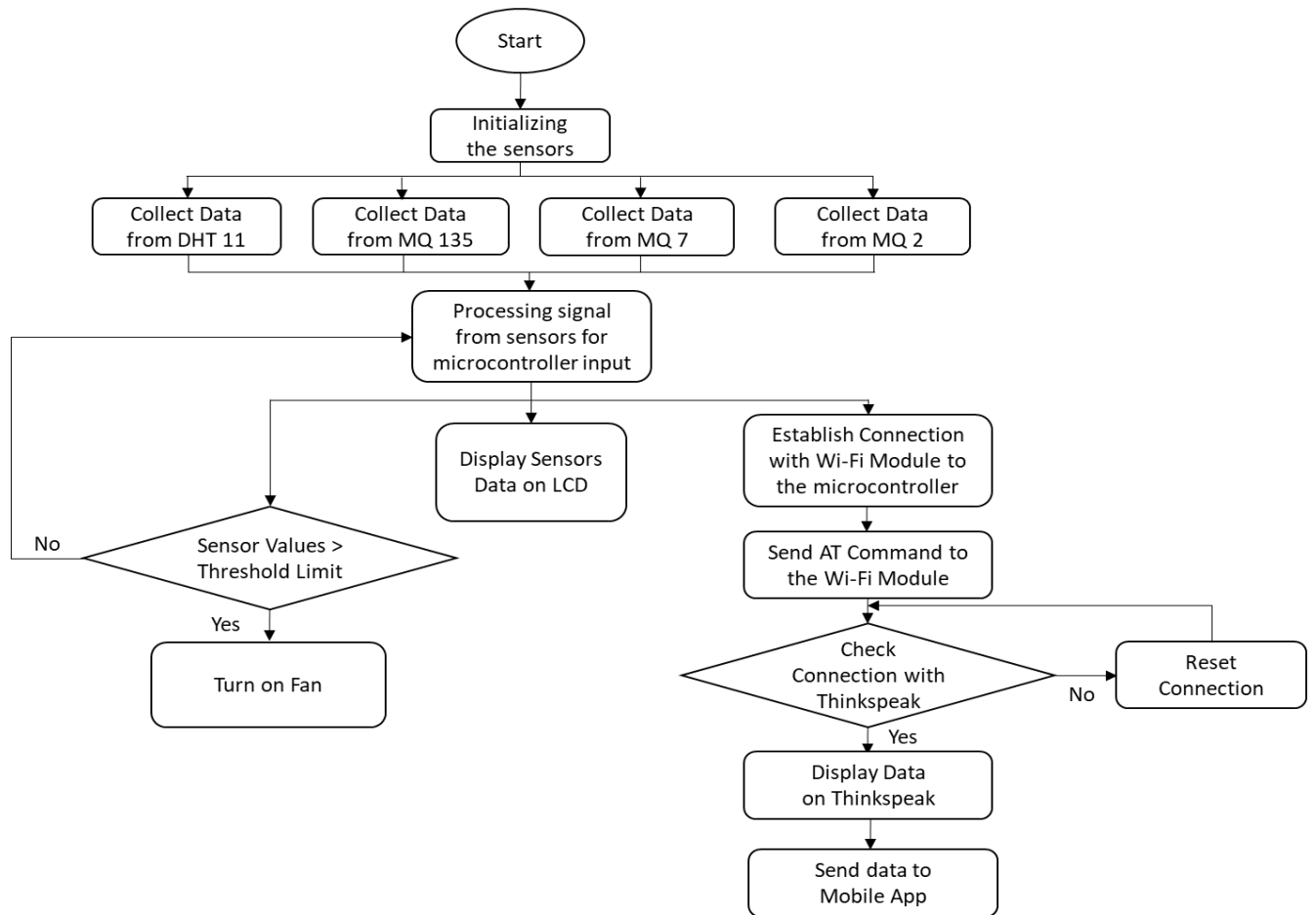


Figure 3.3: Flow Chart of Proposed System

3.3 Circuit Diagram of Proposed System

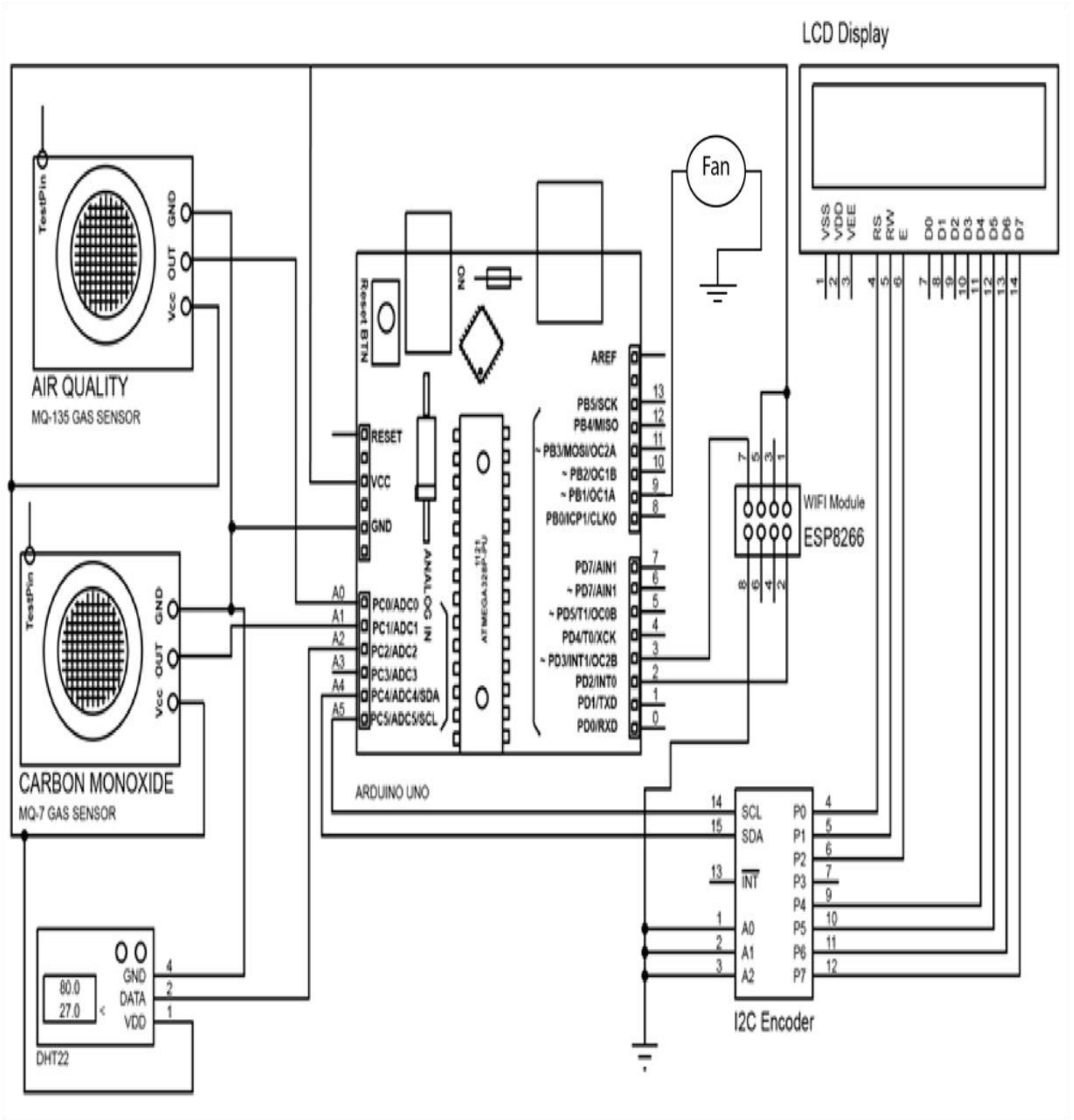


Fig. 3.4. Circuit diagram for proposed model of the system

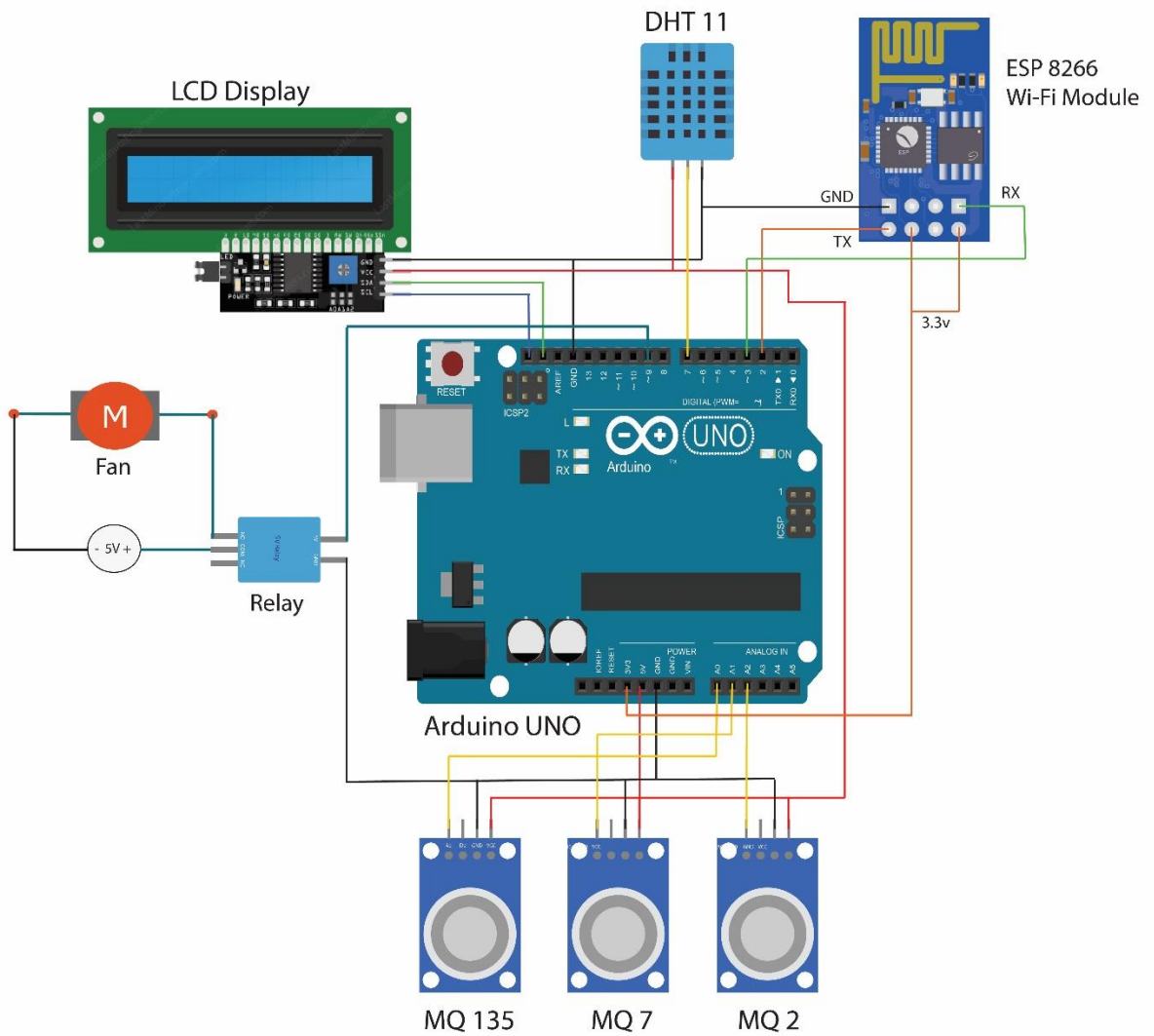


Figure 3.5: Implementation schema of an Iot based air pollution and weather monitoring system

3.4 Proposed Model in Real Life

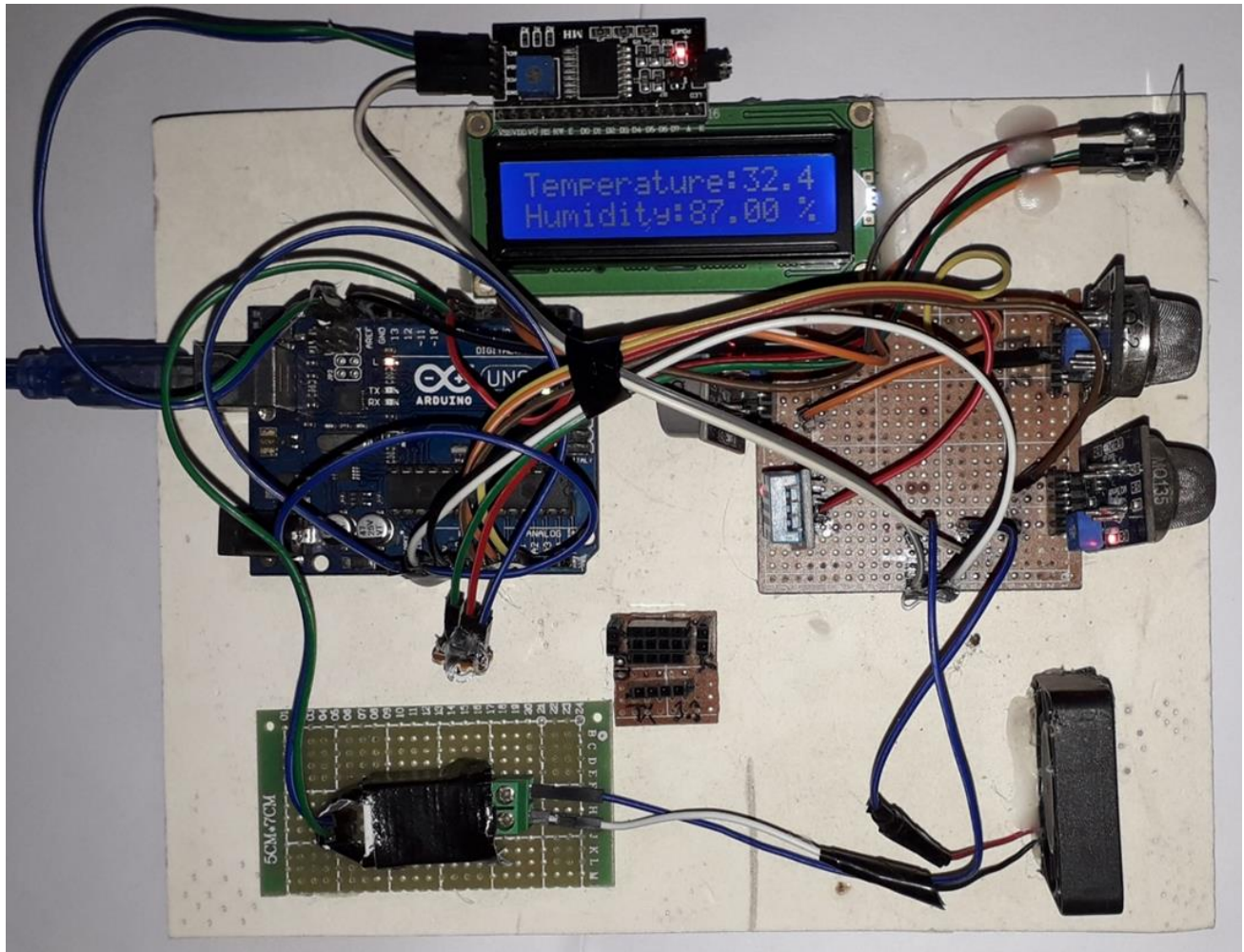


Fig. 3.5. Proposed Model in Real Life

Basically, it's a device that collects data from the environment we've chosen for our research. A variety of sensors are installed into the device, and they collect data through sensing from the environment. This data is then transformed to digital and sent to a server where it is stored.

3.5 Components of the Device

This device is consisting of the following components:

- 1) Arduino Uno R3
- 2) DHT 11 Temperature and Humidity sensor
- 3) MQ-7 CO Gas sensor
- 4) MQ 135 Air quality sensor module
- 5) MQ-2 LPG gas sensor
- 6) ESP 8266 Wi-Fi Module
- 7) Wire
- 8) DC Fan
- 9) Breadboard
- 10) LCD Display
- 11) Relay
- 12) I2C Module for 16×2 LCD

These components are briefly discussed in the following section

3.5.1 Arduino Uno



Fig. 3.6. Arduino Uno

Arduino UNO is a prototype board among the most widely used. It is tiny and rich in characteristics. The board is provided with an Arduino boot loader included in it. Arduino Uno is an ATmega328P-based microcontroller board. Arduino Uno is seen in Fig.3.6. There are 14 digital input/output pins available (6 of them for PWM output), 6 analog inputs, a ceramic CSTCE16M0V53-R0 resonator (16MHz), USB connectivity, power connector, an ICSP header and a reset button. The mikrocontroller is supported with everything necessary. Just connect it to a computer with a USB cable or power it to start using an AC-to-DC adapter or battery [11]. The Aurdino Uno Arduino Uno technical specifications is shown below:

Table 1 Arduino Uno Technical Specifications

Microcontroller	ATmega328
Operating Voltage	5 v
Input Voltage	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (PWM Output - 6)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (for Bootloader - 0.5 KB)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

3.5.2 ESP 8266 Wi-Fi Module

The Wi-Fi Module ESP8266 is used to connect to any internet hotspot and send sensor data over wireless Internet to the ThingSpeak Platform. The Wi-Fi module ESP8266 is an autonomous SOC with a built-in TCP/IP protocol stack that can provide Wi-Fi connectivity to any microcontrollers

[22]. The ESP8266 is able to host or disable any Wi-Fi network functionality from a different application processor. Each ESP8266 module has AT command set firmware pre-programmed. So, you can just attach it to a device called Arduino [14]. The monitoring are uploaded into the cloud here. ESP8266EX has a 32-bit Tensilica Processor, standard digital peripheral interfaces, antenna connection, RF balun, power amplifier, low noise amplifier, filters and power management modules. Due to its large working temperature range, it can be used reliably in industrial situations. Highly integrated functionalities on the chip and minimum discrete external component count, The chip is reliable, compactness and robustness. The Fig. 3.7 shows ESP 8266 Wi-Fi Module.

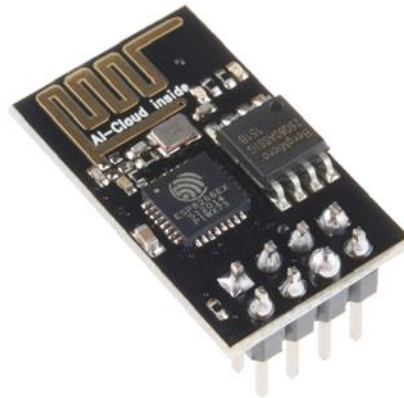


Figure 3.7: ESP 8266 Wi-Fi Module

A Tensilica 32 bit RISC processor with an extra-low power consumption reaching a maximum clock speed of 160 MHz is integrated into the ESP8266EX microcontroller [15]. Around 80% of the processing power can be used for user application programming and development using the Real Time Operating System (RTOS) and the Wi-Fi stack [19] .

3.5.3 DHT 11 Temperature and Humidity sensor

DHT 11 Temperature & Humidity Sensor provides a sensor compound with temperature & humidity and a digital calibrated signal output. It provides great reliability and outstanding long-term stability by using high-class digital signal acquisition technology with temperature and

humidity sensing technology. This sensor has a high-performance 8bit microcontroller that delivers great quality with quick response, ant interference capacity and cost-effectiveness, including a resistive humidity measurement component and a temperature measurement equipment [23]. The temperature and humidity sensors of DHT 11 are presented in Fig. 3.8.

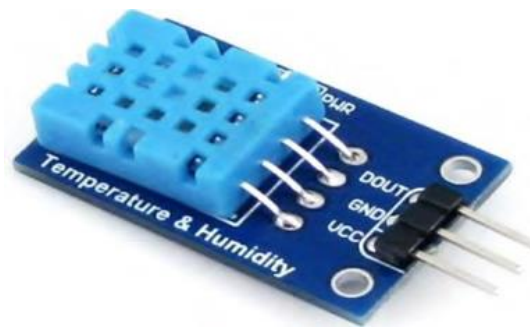


Figure 3.8: DHT 11 Temperature and Humidity sensor

There are four pins on the DHT11 sensor: VCC (power), GND (ground), Data Pin (data), and a not-connected pin. Sensors and microcontrollers communicate via a pull-up resistor of 5 to 10k ohms .In DHT 11 Operating Voltage is 3.5V to 5.5V and Operating current is 0.3mA.

3.5.4 MQ-7 CO Gas sensor



Figure 3.9: MQ-7 CO Gas sensor

The gas sensor MQ7 is used to detect carbon monoxide presence. The MQ-7 CO gas sensor is displayed in Fig. 3.9. Methane and other combustion steam are also detected. It is cheap and can

be applied in many ways. This sensor has an inflammable gas and smoke sensitivity. The smoke sensor is supplied with 5 volts. The smoke sensor indicates smoke output by the voltage, more smoke output by the voltage [20]. To control the sensitivity, a potentiometer is supplied. SnO₂ is a low-conductivity sensor used when the air is pure [16]. However, if the smoke sensor exists, the smoke concentration produces an analog resistive output. A heater is provided on the circuit. Heating power from the power source is provided via VCC and GND. A variable resistor is present on the circuit. The resistance of the pin is dependent on the air smoke in the sensor [21]. When the voltage is increased between the sensor and load resistor, the resistance is reduced [25].

3.5.5 M135 Air quality sensor module



Figure 3.10: M135 Air quality sensor module

MQ-135 is a gas sensor used for measuring combustible gas concentration. It is less conductive in clean air as it improves its conductivity when combustible gasses are present in the air. The sensor can detect gas concentrations from 100 PPM to 1000 PPM [18]. The M135 Air Sensor quality module is shown in Fig. 3.10. It provides a safe and environmentally friendly gas detection component ideal for ammonia, aromatic compounds, sulfur, benzene vapour, and other gasses. Air quality sensor detects a variety of gasses, such as NH₃, NO_x, alcohol, benzene, smoke and CO₂. It is excellent for use with a simple drive and monitoring system in office or factory [17].

3.5.6 MQ-2 LPG gas sensor

The gas sensor MQ2 is an electronic sensor used to sensitize gasses such as LPG, propane , methane, or hydrogen concentrations in the air. The MQ2 gas sensor is sometimes called chemical storage system. It contains a sensing material, the resistance of which varies as it comes to gas. This variation in the resistance value is used for gas detection [24].



Figure 3.11: MQ-2 gas sensor

MQ2 is a gas sensor kind of metal oxide semiconductor. The 5V DC voltage is operated by the sensor. It can detect gasses between 200 and 10000ppm in concentration. These sensors detect the presence of methane, butane, LPG and smoke gasses in the air but they cannot distinguish between gasses. So they don't know what gas it is. The modular form of this sensor can be utilized for the detection of only one particular gas without interacting with a single microcontroller. Only the gas can be found. But the sensor must be utilized without a module if ppm is to be determined. This sensor is also utilized for air quality monitoring, gas leak alert.

3.5.7 Liquid-crystal display 16×2

LCD (Liquid Crystal Display) screens are electrical display modules that have a wide range of uses. A 16×2 LCD display is a relatively basic module and is widely utilized in numerous devices and circuits. A 16×2 LCD can display 16 characters per line, and there are two such lines. Each

character is presented in a 5×7 pixel matrix in this LCD. The 16×2 alphanumeric dot matrix display can display 224 distinct characters and symbols.



Figure 3.12: Liquid-crystal display 16×2

The key advantages of this LCD device are its low power consumption and affordable price. In addition to taking up a lot of space, these LCD displays are slow and their lifespan will be shortened due to the direct current. Each of the two registers on a 16×2 LCD is called the Data Register and Command Register. To move from one register to another, the RS (register select) is utilized. Command registers are used when the register set is set to '0' It's also known as a "data register" when the register set number is "1".

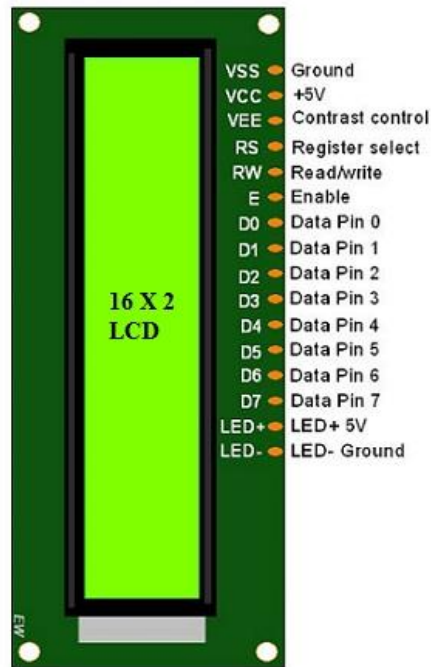


Figure 3.13: LCD-16×2-pin-diagram

- Ground (Pin 1): On the display, there is a GND pin for connecting a microcontroller unit or power source to the display's GND terminal.
- VCC (Pin 2): This is the display's voltage supply pin, which is used to link the power source's voltage supply pin.
- Control Pin (Pin 3): A variable POT that can supply 0 to 5V is connected to this pin to manage the display's difference.
- Register Select (Pin 4): For connecting a microcontroller unit, this pin toggles between command and data register, obtaining either 0 or 1.
- Read/Write Pin (Pin 5): If the microcontroller unit pin is set to 0 or 1, this pin toggles the display between read and write operation.
- Enable Pin (Pin 6): This pin is attached to the microcontroller unit and is constantly held high in order to execute read/write operations.
- Data Pin (7-14 Pin): These pins are used to communicate information to the display. These pins are connected in two-wire modes such as 4-wire mode and 8-wire mode. Four pins are connected to the microcontroller in 4-wire mode, whereas eight pins are connected in 8-wire mode, ranging from 0 to 7.

- Pin 15: For connected to +5V
- Pin 16: For connected to GND

3.5.8 I2C Module

As part of the I2C Module, there is a PCF8574 serial-to-parallel converter that transforms I2C serial data to parallel . The default I2C address for these modules is either 0x27 or 0x3F. Check the black I2C adaptor board on the underside of the module to see which version you have. There will be a default address of 0x3F if there are three sets of pads labeled as A0, A1, and A2. The default address is 0x27 if there are no pads. In addition, the module features a potentiometer for adjusting the display' The screen may need to be adjusted in order for the text to appear correctly.

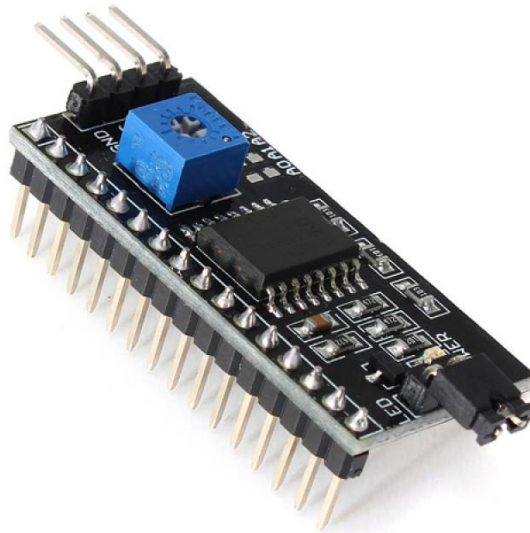


Figure 3.14: I2C Module for 16×2 Character LCD

3.5.9 Relay

It is a switch that is electrically operated. We will find input terminals for a single or many control signals, as well as a set of contact terminals. If the switch has several contact types such as make and break contacts, it can have any number of contacts. Whenever a circuit must be controlled by a low-power signal, or when numerous circuits must be controlled by a single signal, a relay is required. They were originally employed as signal repeaters in long-distance telegraph circuits: they refresh the signal coming in from one circuit by sending it on another. For logical tasks, relays were widely utilized in telephone exchanges and in early computer systems. Traditionally, a relay relies on an electromagnet to close or open the contacts, but various working principles have been developed, such as solid-state relays, which rely on semiconductor features for control without relying on mechanical elements. In order to protect electrical circuits from overload or defects, protective relays are utilized. If we want to operate a latching relay just need a single pulse of control power. Another pulse applied to a second set of control terminals, or a pulse with the opposite polarity, resets the switch, but repeated pulses of the same type have no effect. If the circuits that the relay controls should not be affected by a power outage, then magnetic latching relays come.



Figure 3.15: 5V DC Relay

3.5.10 Problem Formulation and Decision

We have considered CO and H2 to measure AQI. So we will only discuss CO and H2 concentration [10]. Consider,

R_0 = Sensor resistance in fresh air.

R_s = Sensor resistance in displayed gases.

R_0 clean air factor = 10 (provided in the datasheet of the sensor). $p_{curve} = 3$, values for a specific gas obtained from the curve provided in the sensor datasheet.

Dividing the sensor resistance by R_0 clean air factor is calibrated the sensor. The gas percentage has to be calculated first to find the gas concentration.

$$\text{Gas percentage} = \left(\frac{\log(R_s/R_0)}{p_{curve}} - p_{curve} + p_{curve} \right)^{10}$$

To find the gas concentration, gas percentage need to be divided by the R_0 ,

$$\text{Gas concentration} = \frac{\text{gas percentage}}{R_0}$$

According to the gas concentration, AQI level needs to be measured. Table 3.2 shows AQI scale for continues concentration of CO.

Table 2 AQI scale for CO

CO	AQI	Category
1	11	Good
0	0.50	Good
1	11	Good
3	34	Good
5	56	Moderate
8	86	Moderate
10	109	Unhealthy
15	193	Unhealthy

3.5.11 Dataset

Data from the gas sensors that were attached to the Arduino was used to create the following we exported them to an Excel Sheet in a certain format. They will be used for further analysis. Our ability to find more accurate results increases as more data is collected. It's also possible to estimate the air quality in the future.

Table 3 US Embassy Dataset

CO(ppm)	Temperature (Celsius)	Humidity (%)	AQI
1	31	61	Good
0	31	61	Good
1	32	63	Good
2	32	63	Good
3	31	61	Good
5	34	67	Moderate
8	34	67	Moderate
10	34	67	Unhealthy
15	34	67	Unhealthy
12	35	69	Unhealthy

Table 4 Second Dataset of our device

CO(ppm)	Temperature (Celsius)	Humidity (%)	AQI
1	30	62	Good
2	32	63	Good
3	31	63	Good
2	31	63	Good
3	31	61	Good
5	34	67	Moderate
8	34	68	Moderate
12	34	72	Unhealthy
15	34	69	Unhealthy
12	35	70	Unhealthy

3.5.12 Summary

In this section, we've explained about our system device, which we made ourselves. In this section, the circuit diagram and the block diagram were illustrated. We also introduced and quickly detailed our formula for detecting gas percentage and gas concentration.

CHAPTER 4 RESULTS AND DISCUSSION

4.1 Introduction

During this chapter, we will discuss the findings of our device experiment. We put our device in a variety of situations in order to gather as much data.

4.2 Dataset Description

We used the US Embassy of Bangladesh dataset to measure and compare the performance of our device. AQI scores are assigned to each of the 2500 records in this collection. These datasets were taken from their website. During every hour of a 24-hour period (including weekends), as well as every day of a month.

4.2.1 Displaying data in LCD Display

In our project we have take data from ten different places to get more accurate information.

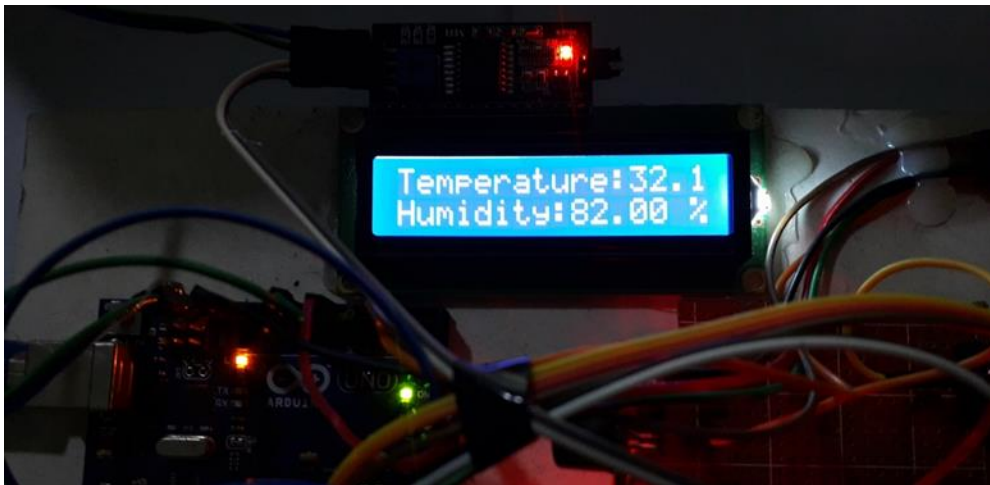


Figure 4.1 : LCD display view Temperature & Humidity(1st place)



Figure 4.1.2: LCD display view CO (1st place)



Figure 4.1.3: LCD display view LPG (1st place)



Figure 4.1.4: LCD display view Air quality (1st place)



Figure 4.1.5: LCD display view LPG(2nd place)



Figure 4.1.6: LCD display view CO(2nd place)



Figure 4.1.7: LCD display view Air quality(2nd place)

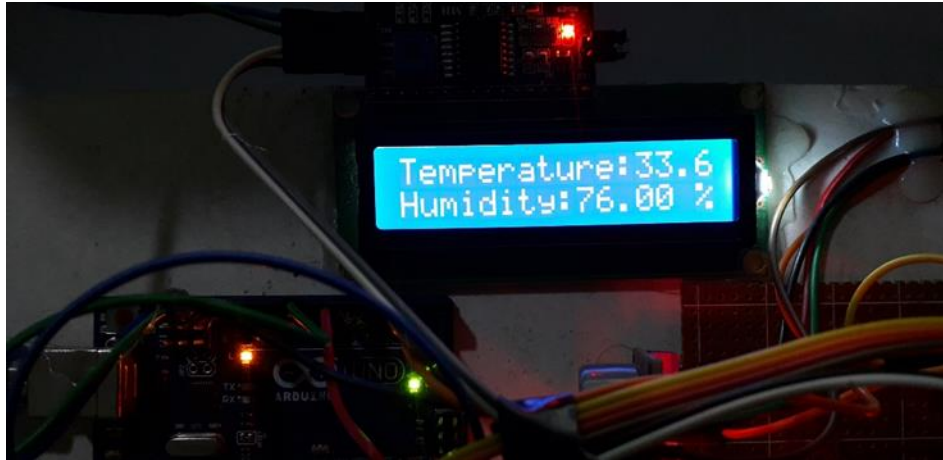


Figure 4.1.8: LCD display view Temperature & Humidity(2nd place)



Figure 4.1.9: LCD display view Air quality(3rd place)



Figure 4.1.10: LCD display view CO(3rd place)



Figure 4.2.1: LCD display view Temperature & Humidity (3rd place)



Figure 4.2.2: LCD display view LPG(3rd place)



Figure 4.2.3: LCD display view LPG(4th place)

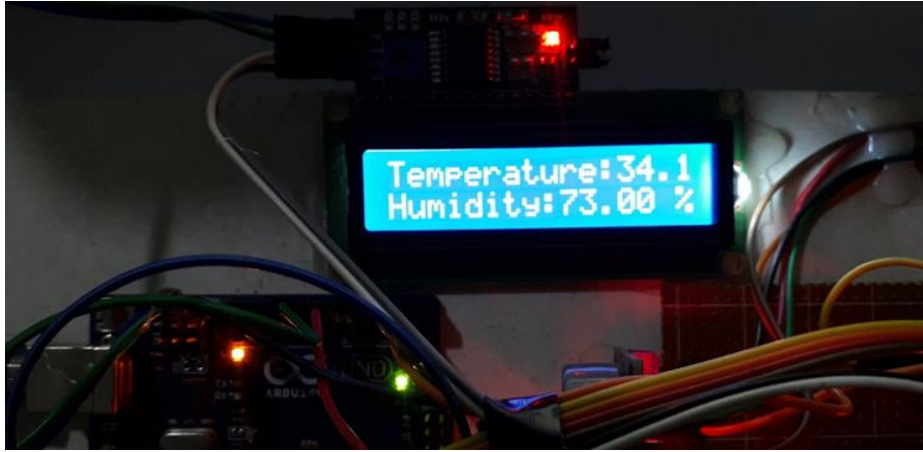


Figure 4.2.4: LCD display view Temperature & Humidity (4th place)



Figure 4.2.5: LCD display view CO (4th place)



Figure 4.2.6: LCD display view Air quality(4th place)



Figure 4.2.7: LCD display view CO(5th place)



Figure 4.2.8: LCD display view Air quality(5th place)



Figure 4.2.9: LCD display view LPG(5th place)

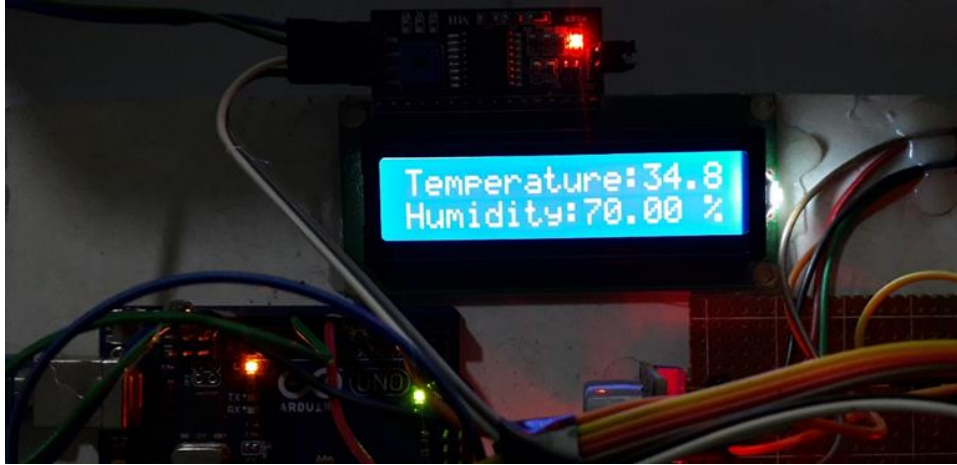


Figure 4.2.10: LCD display view Temperature & Humidity (5th place)



Figure 4.3.1: LCD display view CO(6th place)



Figure 4.3.2: LCD display view Air quality(6th place)

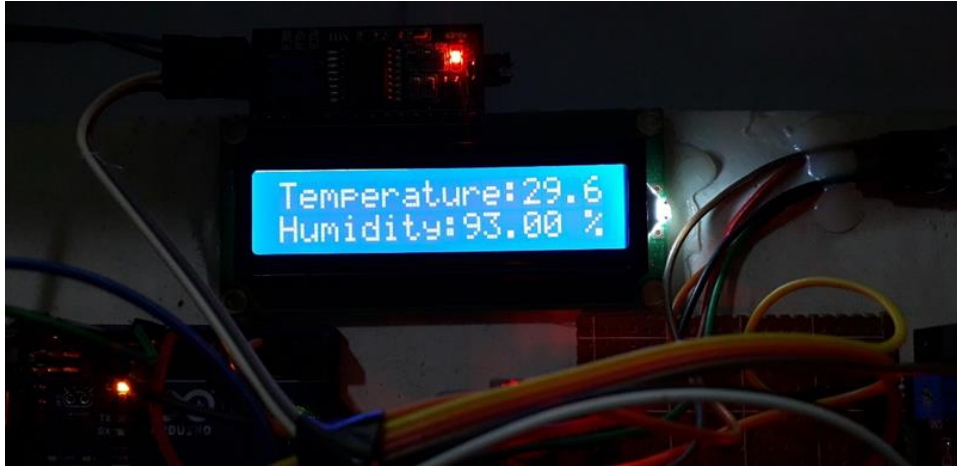


Figure 4.3.3: LCD display view Temperature & Humidity (6th place)



Figure 4.3.4: LCD display view LPG(6th place)

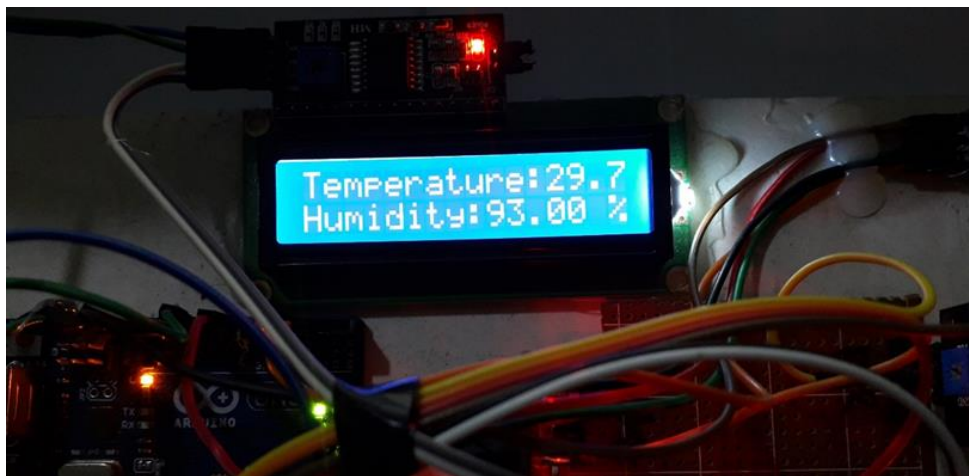


Figure 4.3.5: LCD display view Temperature & Humidity (7th place)



Figure 4.3.6: LCD display view LPG(7th place)



Figure 4.3.7: LCD display view CO(7th place)



Figure 4.3.8: LCD display view Air quality(7th place)



Figure 4.3.9: LCD display view Temperature & Humidity (8th place)



Figure 4.3.10: LCD display view Air quality(8th place)



Figure 4.5.1: LCD display view CO(8th place)



Figure 4.5.2: LCD display view LPG(8th place)



Figure 4.5.3: LCD display view CO(9th place)



Figure 4.5.4: LCD display view Air quality(9th place)



Figure 4.5.5: LCD display view Temperature & Humidity (9th place)



Figure 4.5.6: LCD display view LPG(9th place)



Figure 4.5.7: LCD display view Air quality(10th place)



Figure 4.5.8: LCD display view Temperature & Humidity (10th place)



Figure 4.5.9: LCD display view LPG(10th place)



Figure 4.5.10: LCD display view CO(10th place)

In this all figure show the output of air quality, CO, LPG , temperature and humidity in LCD Display respectively. We can see in these figures that different values are seen in different places.

4.2.2 Sending data to the cloud platform

We've selected 5 data points to send to the cloud platform. Every 10 seconds, the server receives a single input from the Arduino. We used 5 variables to collect the data and sent it to 5 separate fields of the cloud channel and then send data to mobile android App.

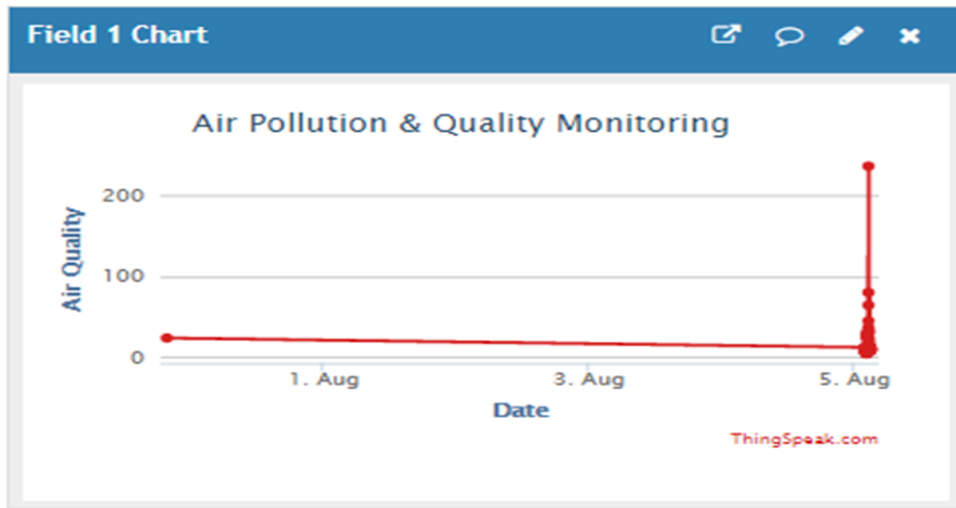


Figure 4.6.1: output of the device in the cloud platform(Air quality)

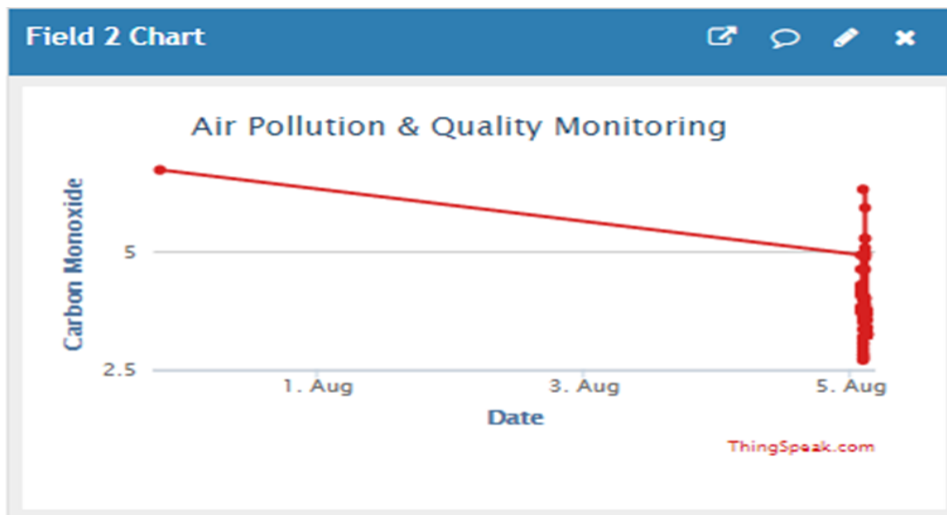


Figure 4.6.2: output of the device in the cloud platform(CO)

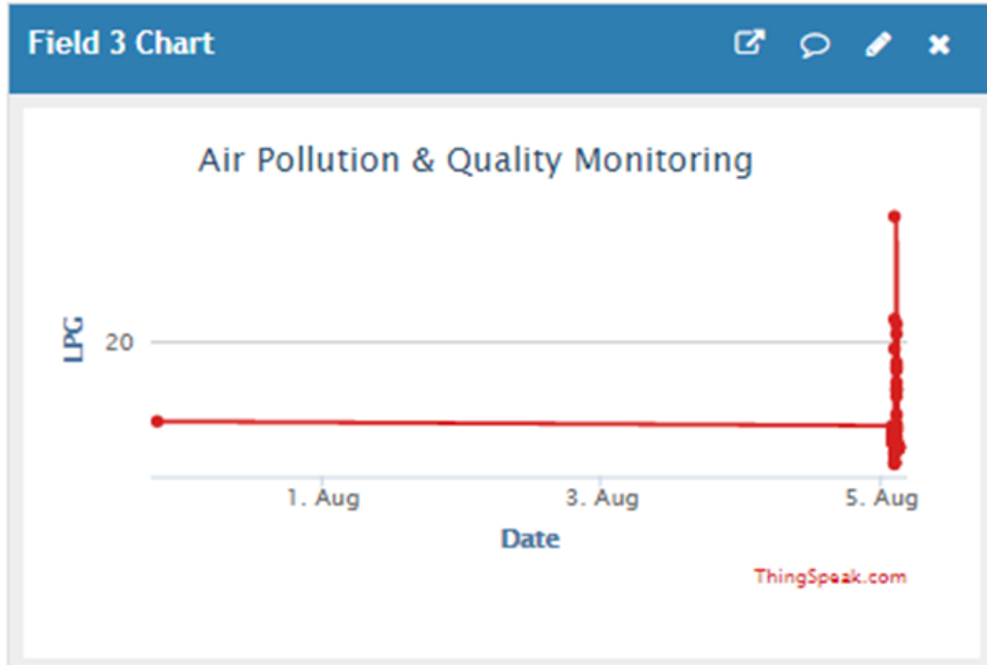


Figure 4.6.3: output of the device in the cloud platform(LPG)

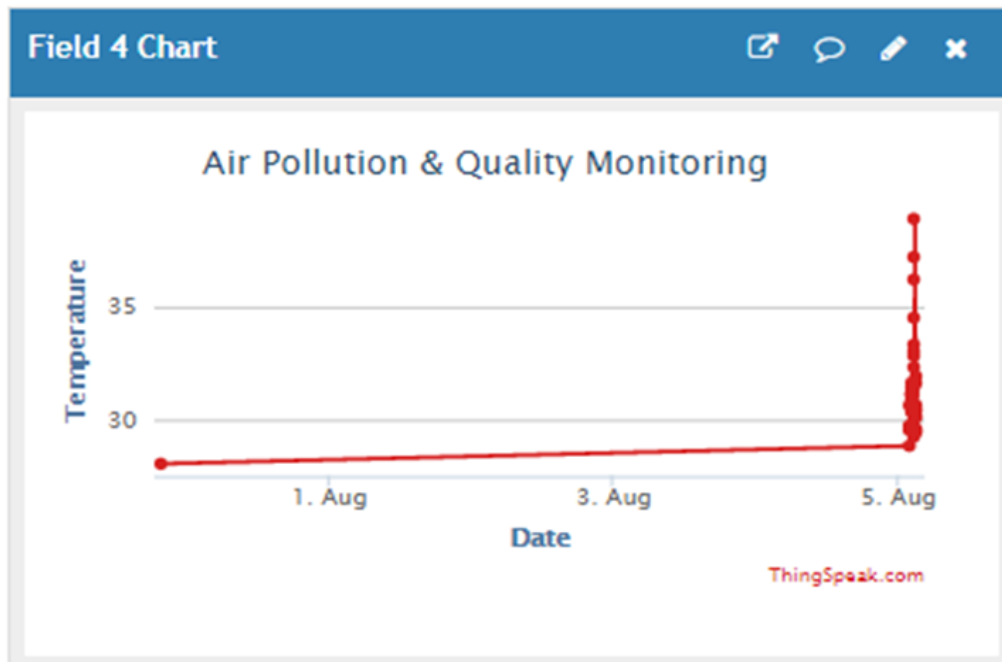


Figure 4.6.4: output of the device in the cloud platform(Temperature)

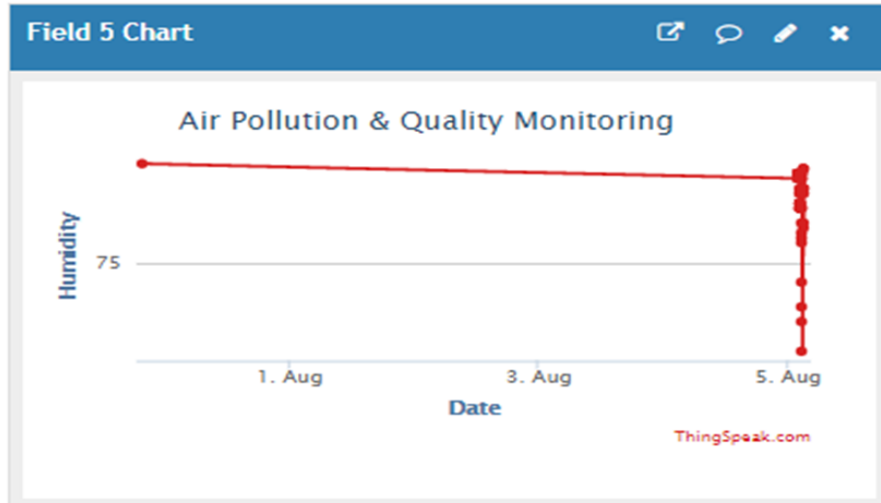


Figure 4.6.5: output of the device in the cloud platform(Humidity)



Figure 4.7.1: Android App Data from first place



Figure 4.7.2: Android App Data from 2nd place



Figure 4.7.3: Android App Data from 3rd place

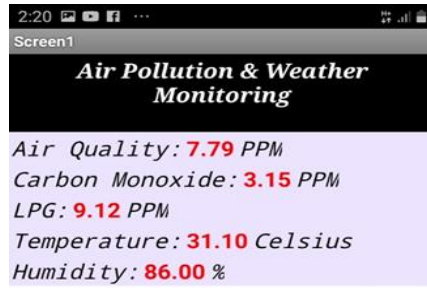


Figure 4.7.4: Android App Data from 4th place



Figure 4.7.5: Android App Data from 5th place



Figure 4.7.6: Android App Data from 6th place

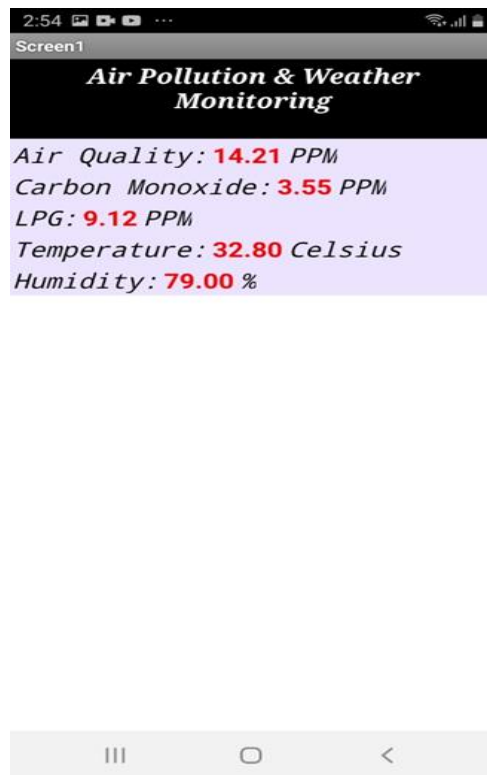


Figure 4.7.7: Android App Data from 7th place



Figure 4.7.8: Android App Data from 8th place



Figure 4.7.9: Android App Data from 9th place

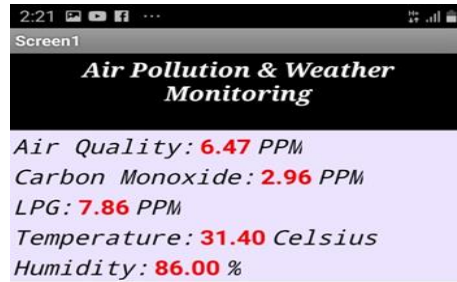


Figure 4.7.10: Android App Data from 10th place

In his all Figures we can see that the Arduino library was loaded, and a message was sent to the LCD display as shown in the figures. The calibrated sensor made the analog output voltage proportionate to the concentration of polluting gases in parts per million (ppm). The data is initially shown on the LCD screen, and then communicated to the Wi-Fi module (ESP 8226). The Wi-Fi module transmits the measured data value to the server ("Thingspeak") over the internet. The Wi-Fi module is designed to send measured data to an application on a cloud server and then cloud server send measured parameters in our Android app.

4.2.3 Cost Estimation Structure

Sl.No	Equipment name	Quantity	Cost (BDT)
1.	Arduino UNO	1	450
2.	Esp8266 wifi module	1	220
3.	MQ-135	1	280
4.	MQ-7	1	180
5.	MQ-2	1	120
6.	DHT-11	1	170
7.	LCD 16×2	1	160
8.	I2C Module	1	160
9.	Relay 5v	1	40
10.	PCB Board	1	50
11.	DC Fan	1	60
Total			1880 BDT

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Future work

We did not utilize dust sensors and alerting equipment to alert us in this Project Model. In future, we'll do that. We haven't done this in our model for larger houses like buildings, schools. We should need a wireless sensor network to operate in such situation. But we didn't have the funds to make these technologies . We will work in future on improving the quality of the air with radiation detection and machine learning algorithms.

5.2 Advantages

- A wide variety of sensors are readily available in the marketplace
- Easy to use, small size, and easy to handle
- Sensors have a long lifespan and relatively inexpensive.
- Indoor and outdoor air quality can be examined.
- Detection of a wide range of physical characteristics
- Accident caused by LPG gas will be reduced.
- Electrical cost will be reduced as unnecessary exhaust fan must be kept on and exhaust fan will be on automatically if required.

5.3 Applications

- Monitoring Indoor air quality
- Monitoring Industrial perimeter
- Monitoring Roadside pollution
- Easy to use common man

5.4 Conclusion

This research reveals the intelligent approach to monitor the environment and air at cheap cost yet efficient and integrated systems. Multiple sensors and their functioning processes were explored in the proposed architectural functions. Here, too, we explain their working conditions, their functioning, their optimal application and their processes of data collection and comparison with standard base data. The air pollution monitoring device was tested in several regions of the city to measure gas levels. The sensor parameters have also been submitted to the data server. Our project has shown that it's efficient and cost-effective and that with certain highly operating sensors it can truly be trusted to everyone, and that its data will serve as a means of taking certain essential actions for bettering the society.

References

- [1] Z. K. 2. ., A. A. M. 3. ., M. M. 4. Harsh N. Shah 1, "IOT Based Air Pollution Monitoring System," *International Journal of Scientific & Engineering*, vol. 9, no. 2, pp. 10-20, 2018.
- [2] R. Rawal, "Air Quality Monitoring System," *International Journal of Computational Science and Engineering.*, vol. 9, no. 1, pp. 1-9, 2019.
- [3] K. S. S. Priya, "Review Paper on Air Pollution Monitoring," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 1, 2015.
- [4] H. Gupta, D. Bhardwaj, H. Agrawal, V. A. Tikkiwal and A. Kumar, "An IoT Based Air Pollution Monitoring System for Smart Cities," *IEEE*, 2019.
- [5] E. N.-O. O. M. S. J. a. Kennedy Okokpujie, "A SMART AIR POLLUTION MONITORING," *International Journal of Civil Engineering and Technology (IJCIET)*, vol. 9, no. 9, p. 799–809, 2018.
- [6] M. A. J. A. M. A. M. A. A. Muhammad Saqib, "Smart Environment Monitoring System by employing Wireless Sensor Networks on Vehicles For Pollution Free Smart Cities," *Procedia Engineering*, 2015.
- [7] M. D. a. G. M. Jagriti Saini1*, "A comprehensive review on indoor air," *Saini et al. Sustainable Environment Research*, vol. 9, no. 3, pp. 30-40, 2020.
- [8] R. K. Kodali, S. Pathuri and S. C. Rajnarayanan, "Smart Indoor Air Pollution Monitoring Station," *IEEE*, 2020.
- [9] R. U. R. M. Y. A. M. A. N. J. Md. Hasib Bin Hossain Khan, "AIR POLLUTION MONITORING SYSTEM FOR SMART CITY," 2019.
- [10] M. B. M. G. T. Boyanka Marinova Nikolova, "AIR QUALITY MONITORING SYSTEM," *ResearchGate*, vol. 9, no. 9, pp. 10-20, 2006.
- [11] H. Mokrani, R. Lounas, M. T. Bennai, D. E. Salhi and R. Djerbi, "Air Quality Monitoring Using IoT: A Survey," *IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2019.
- [12] M. B. Marinov, I. Topalov, E. Gieva and G. Nikolov, "Air quality monitoring in urban environments," *IEEE*, 2016.
- [13] S. Bedoui, S. Gomri, H. Samet and A. Kachouri, "Air pollution monitoring system using LabVIEW," *IEEE*, 2015.

- [14] G. Spandana and R. Shanmughasundram, "Design and Development of Air Pollution Monitoring System for Smart Cities," *IEEE*, 2018.
- [15] B. Bathiya, S. Srivastava and B. Mishra, "Air pollution monitoring using wireless sensor network," *IEEE*, 2016.
- [16] V. Barot and V. Kapadia, "Air Quality Monitoring Systems using IoT: A Review," *IEEE*, 2020.
- [17] H. P. L. d. Medeiros and G. Girão, "An IoT-based Air Quality Monitoring Platform," *IEEE*, 2020.
- [18] B. S. Nihal Kularatna, "An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements," *IEEE*, 2008.
- [19] S. B. A. M. R. P. Iqra Javid, "Air Pollution Monitoring System using IoT," *International Journal of Engineering and Advanced Technology (IJEAT)*, 2019.
- [20] E. N.-O. O. M. S. J. a. Kennedy Okokpujie, "A SMART AIR POLLUTION MONITORING," *International Journal of Civil Engineering and Technology (IJCIET)*, 2018.
- [21] J. T. Yu-Lin Zhao¹, "Development of IoT Technologies for Air Pollution Prevention and Improvement," : *Taiwan Association for Aerosol Research*, 2020.
- [22] M. D. Jagriti Saini¹, "Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review," *International Journal of Environmental Research and Public Health*, 2020.
- [23] Z. Haslam, "DHT11 Temperature/Humidity Sensor," *Project Hub*, 2017.
- [24] A. Mukherjee, "Smoke Detection using MQ-2 Gas Sensor," *Project Hub*, 2016.
- [25] I. Lohs, "Carbon Monoxide Gas Sensor MQ-7," *Project Hub*, 2020.

APPENDIX

Microcontroller Code of the proposed model

```
#include <Wire.h>

//#include <LCD.h>

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

#include <SoftwareSerial.h>
SoftwareSerial ser(2, 3);      // RX, TX

int relay =9;

int gas_sensor = A0; //Sensor pin
float m = -0.3376; //Slope
float b = 0.7165; //Y-Intercept
float R0 = 15.52; //Sensor Resistance in fresh air from previous code

int CO_sensor = A1; //Sensor pin
float m1 = -0.6527; //Slope
float b1 = 1.30; //Y-Intercept
float R01 = 6.42; //Sensor Resistance

int LPG_sensor = A2; //Sensor pin
float m2 = -0.6527; //Slope
float b2 = 1.30; //Y-Intercept
float R02 = 6.42; //Sensor Resistance

#include <DHT.h>
#define DHTPIN 7
```

```

#define DHTTYPE DHT11 // DHT 22 (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

String apiKey = "JQM3JYWA3LZDFZLU"; // Edit this API key according to your Account

void setup() {
  lcd.begin();
  // PC to Arduino Serial Monitor
  Serial.begin(115200); // Arduino to ESP01 Communication
  dht.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print(" Air Quality & ");
  lcd.setCursor(0, 1);
  lcd.print("Weather Station");
  delay(2000);
  lcd.clear();
  ser.begin(115200); // reset ESP8266
  ser.println("AT+RST"); // Resetting ESP8266
  char inv ="";
  String cmd = "AT+CWJAP";
  String Host_Name = "WALTON PRIMO RM3"; // Edit Host_Name
  String Password = "PHOTON2020"; // Edit
  cmd+= "=";
  cmd+= inv;
  cmd+= Host_Name;

```

```

cmd+= inv;
cmd+= ",";
cmd+= inv;
cmd+= Password;
cmd+= inv;
ser.println(cmd);          // Connecting ESP8266 to your WiFi Router

pinMode(gas_sensor, INPUT);
pinMode(CO_sensor,INPUT);
  pinMode(LPG_sensor,INPUT);
pinMode(7,INPUT);
pinMode(9,OUTPUT);

  Serial.println(F("DHTxx test!"));

dht.begin();
}

void loop() {

  // put your main code here, to run repeatedly:
float sensor_volt; //Define variable for sensor voltage
float RS_gas; //Define variable for sensor resistance
float ratio; //Define variable for ratio
float sensorValue = analogRead(gas_sensor); //Read analog values of sensor
sensor_volt = sensorValue*(5.0/1023.0); //Convert analog values to voltage
  RS_gas = ((5.0*10.0)/sensor_volt)-10.0; //Get value of RS in a gas
ratio = RS_gas/R0; // Get ratio RS_gas/RS_air

```

```

double ppm_log = (log10(ratio)-b)/m; //Get ppm value in linear scale according to the the ratio
value

double ppm = pow(10, ppm_log); //Convert ppm value to log scale

Serial.print("Air Quality = ");

Serial.println(ppm);

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Air Quality:");

  lcd.print(ppm);

  lcd.print(" ppm ");

  delay(2000);

  lcd.clear();

  if(ppm>=30){
    digitalWrite(9,HIGH);
  }
else{
  digitalWrite(9,LOW);
}

float sensor_volt1; //Define variable for sensor voltage

float RS_gas1; //Define variable for sensor resistance

float ratio1; //Define variable for ratio

float sensorValue1 = analogRead(CO_sensor); //Read analog values of sensor

sensor_volt1 = sensorValue1*(5.0/1023.0); //Convert analog values to voltage

RS_gas1 = ((5.0*10.0)/sensor_volt1)-10.0; //Get value of RS in a gas

ratio1 = RS_gas1/R01; // Get ratio RS_gas/RS_air

double ppm_log1 = (log10(ratio1)-b1)/m1; //Get ppm value in linear scale according to the the
ratio value

```

```

double ppm1 = pow(10, ppm_log1); //Convert ppm value to log scale
Serial.print("CO PPM = ");
Serial.println(ppm1);

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("CO:");
    lcd.print(ppm1);
    lcd.print(" ppm ");
    delay(2000);
    lcd.clear();

    float sensor_volt2; //Define variable for sensor voltage
float RS_gas2; //Define variable for sensor resistance
float ratio2; //Define variable for ratio
float sensorValue2 = analogRead(LPG_sensor); //Read analog values of sensor
sensor_volt2 = sensorValue2*(5.0/1023.0); //Convert analog values to voltage
RS_gas2 = ((5.0*10.0)/sensor_volt2)-10.0; //Get value of RS in a gas
ratio2 = RS_gas2/R02; // Get ratio RS_gas/RS_air

double ppm_log2 = (log10(ratio2)-b2)/m2; //Get ppm value in linear scale according to the the
ratio value

double ppm2 = pow(10, ppm_log2); //Convert ppm value to log scale
Serial.print("LPG PPM = ");
Serial.println(ppm2);

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

```

```
lcd.print("LPG:");
```

```
lcd.print(ppm2);
```

```
lcd.print(" ppm ");
```

```
delay(2000);
```

```
lcd.clear();
```

```
delay(2000);
```

```
float hum = dht.readHumidity();
```

```
Serial.println(hum);
```

```
float temp = dht.readTemperature();
```

```
Serial.println(temp);
```

```
lcd.clear();
```

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

```
lcd.print("Temperature:");
```

```
lcd.print(temp);
```

```
lcd.print(" Celcius ");
```

```
lcd.setCursor(0,1);
```

```
lcd.print("Humidity:");
```

```
lcd.print(hum);
```

```
lcd.print(" % ");
```

```
delay(2000);
```

```
lcd.clear();
```

```
String state1=String(ppm);
```

```
// Converting them to string
```

```

String state2=String(ppm1);
String state3=String(ppm2);
    String state4=String(temp);
String state5=String(hum);

String cmd = "AT+CIPSTART=\"TCP\", \"";    // Establishing TCP connection
cmd += "184.106.153.149";                // api.thingspeak.com
cmd += "\",80\r\n\r\n";                 // port 80

ser.println(cmd);
Serial.println(cmd);

if(ser.find("Error")){

    Serial.println("AT+CIPSTART error");
    return;
}

String getStr = "GET /update?api_key=";    // prepare GET string

getStr += apiKey;
getStr += "&field1=";
getStr += String(state1);                // Humidity Data
getStr += "&field2=";
getStr += String(state2);
    getStr += "&field3=";
getStr += String(state3);

```

```

    getStr += "&field4=";
getStr += String(state4);
    getStr += "&field5=";
getStr += String(state5);
getStr += "\r\n\r\n";

cmd = "AT+CIPSEND=";
cmd += String(getStr.length());           // Total Length of data

ser.println(cmd);
Serial.println(cmd);

if(ser.find(">")){

    ser.print(getStr);
    Serial.print(getStr);

}

else{

    ser.println("AT+CIPCLOSE");           // closing connection

    // alert user

    Serial.println("AT+CIPCLOSE");

```

```
}
```

```
// Update after every 15 seconds
```

```
}
```