IOT - AIR POLLUTION MONITORING SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in

Computer Science and Engineering

By

AQEEB KHAN(37110046)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC I 12B Status by UGC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600 119

MARCH - 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC I 12B Status by UGC I Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AQEEB KHAN (37110046)** who carried out the project entitled "**IOT – AIR POLLUTION MONITORING SYSTEM**" under my supervision from November 2020 to March 2021.

Internal Guide

Dr. S.VIGNESHWARI, M.E., Ph.D.

Dr. L. LAKSHMANAN, M.E., Ph.D.

Head of the Department

Submitted for Viva voce Examination held on____________________

Internal Examiner **External Examiner**

DECLARATION

I, **AQEEB KHAN (37110046)** hereby declare that the Project Training Report entitled **"IOT – AIR POLLUTION MONITORING SYSTEM"** done by me under the guidance of Dr. S. Vigneshwari, M.E., Ph.D., at Sathyabama Institute of Science and Tehnology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.SASIKALA M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI M.E., Ph.D., and Dr. L. LAKSHMANAN M.E., Ph.D.**, Heads of the Department, Dept. of Computing for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. VIGNESHWARI M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computing who were helpful in many ways for the completion of the project.

ABSTRACT

Today technology has become an integral part of our day to day life. In this modern world, many technological advancements are taking place. It is important for us toexploit them as much as possible in order to gain more benefits. Technology provides many useful features for keeping our surroundings healthy. Here we come with an innovation which emphasizes on providing an IoT-Enabled Air Pollution Meter With Digital Dashboard On Smartphone that displays real-time air quality readings for the immediate surroundings. This is done to have a better understanding of our surroundings and ensure the quality of air is not compromised by constantly monitoring air quality on our smartphone. There is a wide range of scope for this product to be used in many situations. This system can be upgraded into a more advanced system with added features in the future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IoT	Internet of Things
LCD	Liquid Crystal Display
WIFI	Wireless Fidelity
TCP/IP	Transmission Control Protocol / Internet Protocol
PCB	Printed Circuit Board
GPIO	General-Purpose Input/Output
RF	Radio Frequency
APSD	Automatic Power Save Delivery
WPA	WIFI Protected Access
USB	USB Serial Bus
STA	STAtion (In WIFI Terminology)
DFU	Device Firmware Update
TWI	Two-Wire Interface
LED	Light Emitting Diode
AREF	Analog REFerence
GND	Ground
Vin	Input Voltage
Vout	Output Voltage
AC	Alternating Current
DC	Direct Current
EEPROM	Electrically Erasable Programmable Read-Only Memory
PWM	Pulse Width Modulation
TWI	Two Wire Interface
SCK	Serial Clock
MOSI	Master Out Slave In
MISO	Master In Slave Out

CHAPTER 1

INTRODUCTION

A healthy environment is the first and foremost thing for our happiness. We need a pollution free surrounding for living a safe and secure life. The recent increase in pollution levels in metropolitan cities, especially in Delhi, is a worrying sign. In the world that is advancing rapidly with technology where cars could drive on their own and drones could capture your food, air pollution should not be of much concern but the above statistics just proves it wrong. Our application is one such thing which can provide the surrounding air quality index to the user. This is a basic level system which notifies the user of the various pollutants and their levels present in air. We have also additionally included a buzzer alarm which notifies the user when the pollutants level breaches a certain threshold mark. This will make the user understand that the place is not healthy and safe to live. So, the user can now take necessary steps to reduce air pollution, or move to a safer location. The concept of IoT allows us to store the data containing the types and amount of pollutants present in air so that the user can analyze the changes that happens over a period of time. So the user can decide whether the air quality is improving or reducing over a period of time. There is an urgent need for these kind ofsystems which can be availed by any person especially in places which areheavily polluted. We can therefore think of curbing our daily human activities which leads to air pollution after making thorough analysis of the data. This gives us an opportunity to research on why pollution level is increasing or what good things need to be continued when the pollution level is decreasing. These kind of decisions could be arrived through the data that the user gets through our system.

1.1 Purpose of the Project:

In this modern world, technology provides us many useful features which can be implemented to have a healthy and safe environment. We all are aware of the fact that in places like Delhi and other metropolitan cities, the air quality index is changing from bad to worse. People living in such places should necessarily carry an air pollution monitoring system consisting of all facilities to keep track of air quality index and take necessary steps to improve quality of air in immediate surroundings.

1.2 Existing System:

The existing system is in such a way that it cannot store the data that the air pollution monitoring system provides as an output. It naturally means that it just provides a facility to display the data but the user cannot perform analysis on the data and employ his own methods to tackle air pollution in his location. Also, the existing system does not make use of dust sensor to detect particle pollution.Hence our system would solve the aforementioned problems faced by the users in storing the data apart from detecting particle pollution.

Demerits:

a) The existing system does not have provision for storing data.
b) Does not detect particle pollution.
c) There is no low cost affordable system available.
d) No system available for daily usage.
e) Complex systems and requires high maintenance.

1.3 Proposed System:

Our proposed system is mainly to overcome the aforementioned problems, as well as provide a provision to visualize the data from the air pollution monitoring system in a graphical and tabular manner. The system is designed to detect various harmful gases present in air apart from detecting particle pollutants. It also keeps track of the temperature and humidity of the surrounding. The system also provides an alarm so that the user is made aware of the fact that the surrounding air is not safe for living and the user should take necessary actions to help reduce air pollution. This makes it a simple system which effectively provides valuable inputs to the user regarding the air quality.

Merits:

a) Computer-based system, so it is environment friendly.
b) Stores the data of surrounding air for future use and reference.
c) Cost effective.
d) It works on basic configurations.
e) It can be developed easily since it is developed mainly for adding additional features.
f) Safe and Secure.
g) Low cost and affordable for all.
h) No risk involved.
i) It can adapt batteries or external power supply could be given through any means. So during power shutdowns, it works on external batteries.
j) Efficiency and storage of data for future analytics.

CHAPTER 2

LITERATURE SURVEY

1. Aravindhar, D. J. (2019, November). Iot Based Air Pollution Monitoring System Using Esp8266-12.
2. Kim, S. H., Jeong, J. M., Hwang, M. T., & Kang, C. S. (2017, October). Development of an IoT-based atmospheric environment monitoring system.
3. Nasution, T. H., Muchtar, M. A., & Simon, A. (2019, October). Designing an IoT-based air quality monitoring system.
4. Yamunathangam, D., Pritheka, K., & Varuna, P. (2019). IoT enabled air pollution monitoring and awareness creation system.
5. Al Ahasan, M. A., Roy, S., Saim, A. H. M., Akter, R., & Hossain, M. Z. (2018). Arduino-Based real time air quality and pollution monitoring system.
6. Kumar, S., & Jasuja, A. (2017, May). Air quality monitoring system based on IoT using Raspberry Pi.
7. Khot, R., & Chitre, V. (2017, March). Survey on air pollution monitoring systems.
8. Gokul, P., Srikanth, J., Inbarasu, G., Subramaniyam, K., & Venkatesan, G. P. (2019, July). Internet of Things Based Air Pollution Monitoring and Forecasting System.
9. Kamble, S., Mini, S., & Panigrahi, T. (2018, March). Monitoring Air Pollution: An IoT Application.
10. Singh, R., Gaur, N., & Bathla, S. (2020, November). IoT based Air Pollution Monitoring device using Raspberry Pi and Cloud Computing.
11. Patil, A. G., Mandle, S. S., Lamdade, S. D., & Keripale, N. A. (2019). IOT Based Air Pollution Monitoring System.
12. Gupta, K., & Rakesh, N. (2018, January). IoT based automobile air pollution monitoring system.
13. Patil, D., Thanuja, T. C., & Melinamath, B. C. (2019). Air pollution monitoring system using wireless sensor network (WSN).
14. Chaturvedi, A., & Shrivastava, L. (2020, April). IOT Based Wireless Sensor Network for Air Pollution Monitoring.

15. Kaivonen, S., & Ngai, E. C. H. (2020). Real-time air pollution monitoring with sensors on city bus.
16. Srivastava, H., Mishra, S., Das, S. K., & Sarkar, S. (2020). An IoT-Based Pollution Monitoring System Using Data Analytics Approach.
17. Patil, N. M., Jain, R., Sankhe, S., Vichare, K., & Wankhede, A. (2018). IoT based Environment Pollution Monitoring System.
18. Guanochanga, B., Cachipuendo, R., Fuertes, W., Salvador, S., Benítez, D. S., Toulkeridis, T.,.....& Meneses, F. (2018, November). Real-time air pollution monitoring systems using wireless sensor networks connected in a cloud-computing, wrapped up web services.
19. Balasubramaniyan, C., & Manivannan, D. (2016). Iot enabled air quality monitoring system (AQMS) using raspberry Pi.
20. Bapat, P., Sengunthar, K., Shenvi, K., & Khade, A. IOT based Air and Sound Pollution Monitoring System.

We have also taken a cue from works done by Parveen Sulthana, Pramod Sharma and Vijayakumar Sajjan. We have taken all these works as a reference and have done it using different methodologies apart from doing as much as possible to eliminate the cons in the related work.

One such work is an IoT-based real time air pollution monitoring system proposed to monitor the pollution levels of various pollutants in Coimbatore city. The geographical area is classified as industrial, residential and traffic zones. It proposes an IoT system that could be deployed at any location and store the measured value in a cloud database, perform pollution analysis, and display the pollution level at any given location.

Since the existing work involves too much cost apart from complex implementation and high maintenance, we have eliminated all such factors.Also, our work is done in such a way that it can accommodate further enhancements to it. Ours is a work which focuses on providing a low cost air pollution monitoring system for regular and daily usage for people living in metropolitan and highly polluted cities to be aware of their surroundings.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 Aim of the System:

- The system focuses mainly on one thing, that is to ensure every person gets to afford a low cost air pollution monitoring system for daily usage.
- To ensure the user gets to store the data containing the pollutant types and their amount. This is necessary for the user to be aware of the surroundings and how it has changed over the past few days or years in terms of air quality.
- To ensure that the user gets a warning when the pollution increases to an extent where it is not feasible for survival.
- To cover all types of harmful gases present in air, apart from particle pollutants.
- To keep track of the temperature and humidity readings to infer more about the surrounding air.
- To facilitate the user in carrying out detailed analysis regarding the airquality and take effective work to neutralize the shortcomings.
- To provide uninterrupted service even during power shutdowns.
- To visualize the data in whatever format the user wishes to, i.e., graphical form, tables, bar graphs, etc.
- To provide a risk free experience throughout the time the user makes use of the system.
- To ensure a provision to upgrade into a newer version consisting of additional features.

3.2 *Scope of the system:*

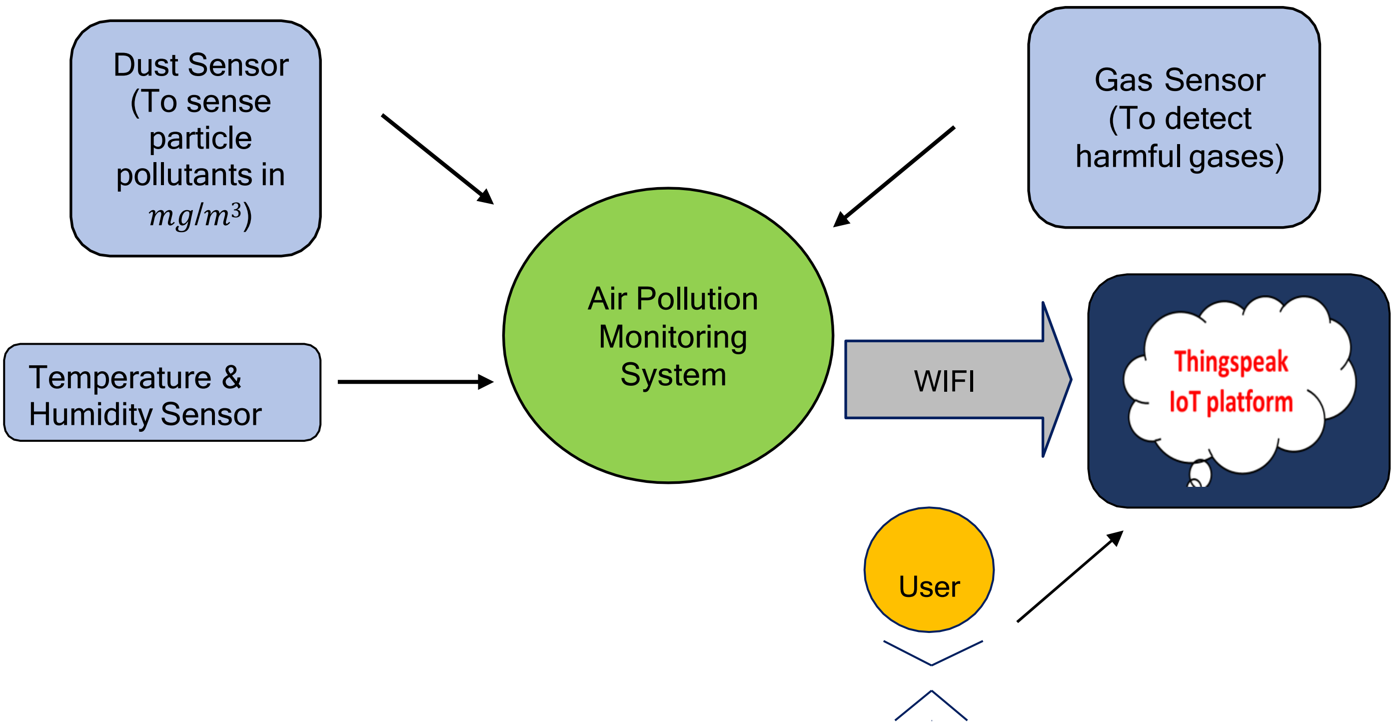
- The system focuses on covering people who lives in highly polluted metropolitan cities like Mumbai, Chennai, Bengaluru, Calcutta, Delhi and so on.
- There are also other places where day by day the pollution level is increasing rapidly.
- Since our system is a low cost one, it covers most of the people since it can be afforded by anyone. And anybody with the knowledge of using smart phones or computers can easily use our system.
- Since our system can also detect LPG gas, there is another scenario in which our system can also be used for detecting LPG gas leakage. So, ours is a multi-purpose system.
- Our system makes use of MQ135 Gas sensor. It can cover most of the harmful gases including CO_2, Ammonium gases, Nitrogen gases, and other harmful gases.
- In the future, there is a provision to go even deeper by making use of additional sensors and we can display indepth details of the surrounding air. By this way, we can cover more properties of the air.
- We can also make use of other concepts like Big Data and Machine Learning which can allow the system itself to perform predictive analysis on the air quality and provide details like how the quality of air might turn out to be after few years, by making use of the data collected over a period of time. These kind of provisions are left in our system where we can do the necessary coding and upload it to our Arduino UNO board, which in turn carries out the instructions.
- We can also add features to send real-time data continuously without any interruptions. And along with an alarm, we can also send a warning message and email to the user.

CHAPTER 4

SYSTEM DESIGN AND METHODOLOGY

4.1 UML Diagram:

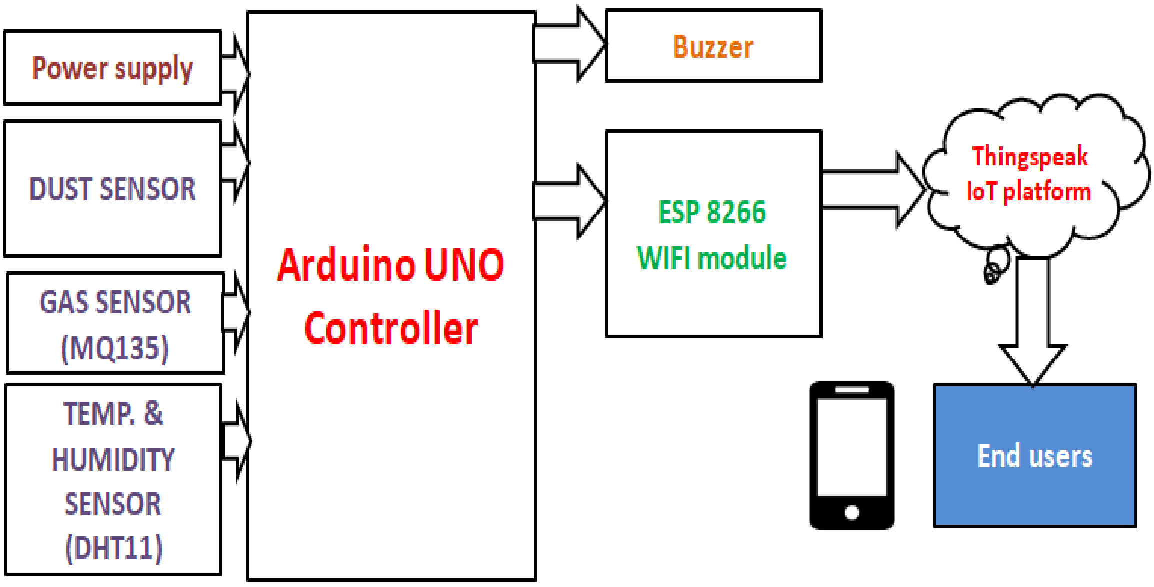
4.1.1 Use case Diagram



(Fig 4.1 : Use Case Diagram)

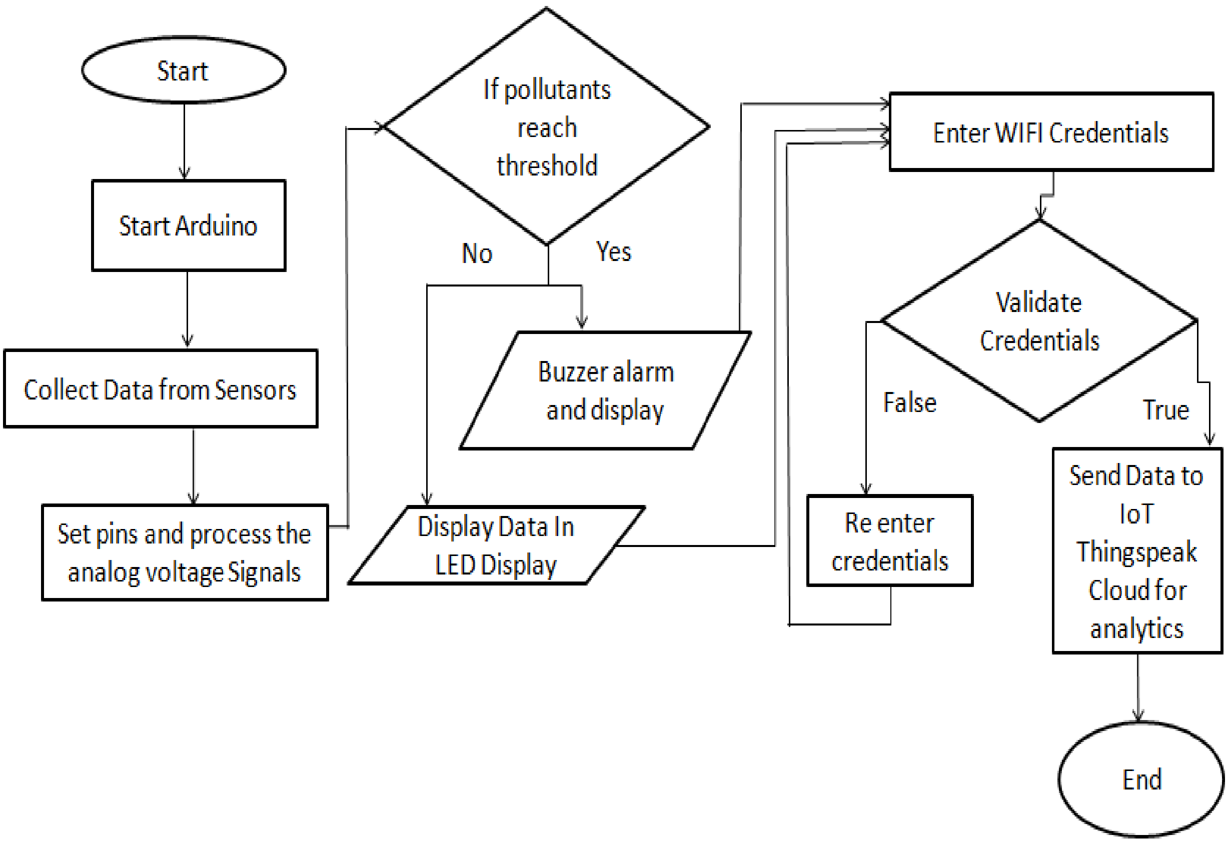
4.2 System Architecture:

4.2.1 Block Diagram



(Fig:4.2 : Block Diagram)

4.3 Flowchart:



(Fig: 4.3: Flowchart)

CHAPTER 5

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

5.1 Introduction:

This project is loaded with arduino software IDE which controls the microprocessor connected with sensor and the ESP8266 WIFI module. This is connected to an IoT cloud platform to display the result and the output data of the system. The Internet of things (IoT) describes the network of physical objects, that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet. The internet of things helps people live and work smarter, as well as gain complete control over their lives. In addition to offering smart devices to automate homes, IoT is essential to business. IoT provides businesses with a real-time look into how their systems really work, delivering insights into everything from the performance of machines to supply chain and logistics operations.

.

5.2 Hardware and Software Requirements :

- Arduino UNO Atmega 16U2.
- WIFI module – ESP8266.
- Dust Sensor.
- DHT11 Temperature and Humidity Sensor.
- MQ135 Gas Sensor.
- Buzzer Module and LCD Display Board.
- Voltage Regulators and ICs.
- 12 V 1 Amp Transistor, 9 Ohm Resistor and Connecting Wires.

The following are the minimum system requirements:

(1) 1.4 GHz Pentium Processor
(2) 90 MB memory and 512 MB RAM
(3) Windows 7 or any other superior version.

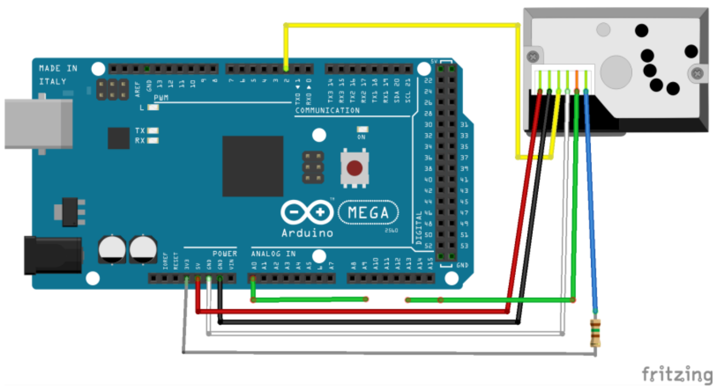
(4) Arduino software IDE version- 1.8.5

(5) Thingspeak IOT platform

5.3 Circuit Diagram:

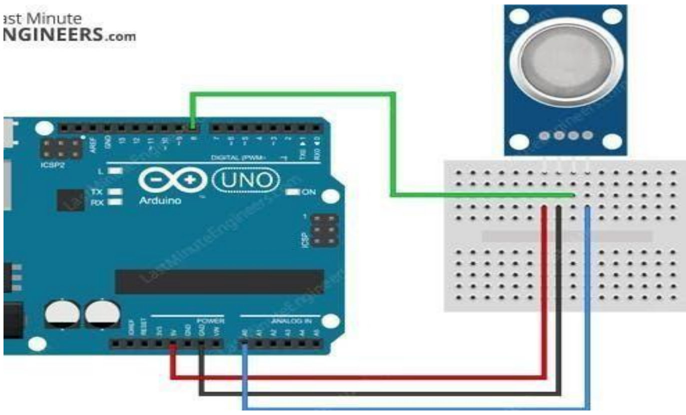
The hardware part of this project is very easy to put together. First of all, makethe connections for all the sensors and the display board with the Arduino UNO board. The illustration of all the sensors, buzzer and the display board connections with Arduino UNO board is shown in the following figures. We have used analog pins to recognize the variations in the values detected by the sensors.

The Dust Sensor needs to be connected to analog pin A2 of the Arduino UNO Board.



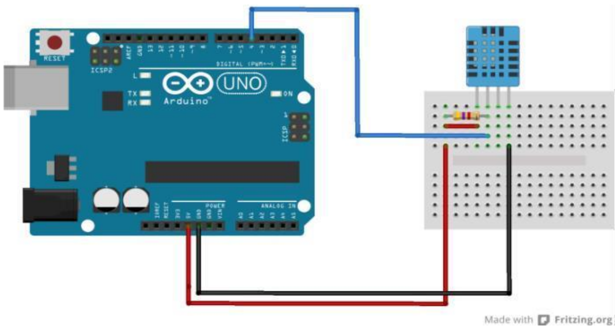
(Fig: 5.1: Dust Sensor Connection)

The MQ135 Gas Sensor needs to be connected with analog pin A0 of the Arduino UNO board which has been demonstrated in the following figure.



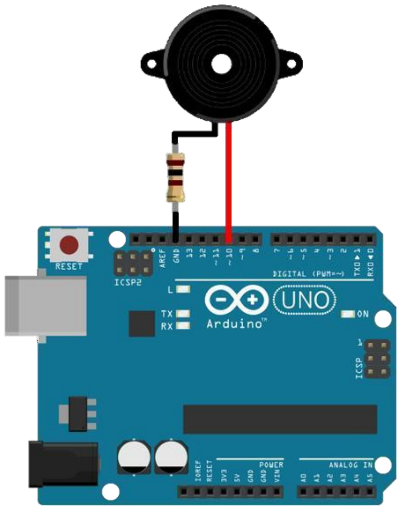
(Fig: 5.2: Gas Sensor Connection)

The DHT11 Temperature and Humidity Sensor needs to be connected to the analog pin A1 of the Arduino UNO Board.



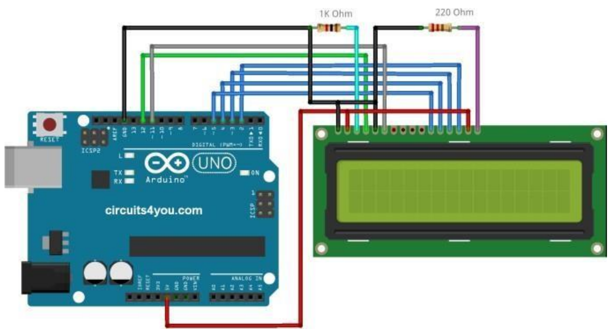
(Fig: 5.3: Temperature & Humidity Sensor Connection)

Make the connections for the buzzer module and the Arduino. Connect the positive pin on the buzzer with pin 7 on the Arduino and the buzzer's negative pin with the GND pin on the Arduino as illustrated below.



(Fig: 5.4: Buzzer connection)

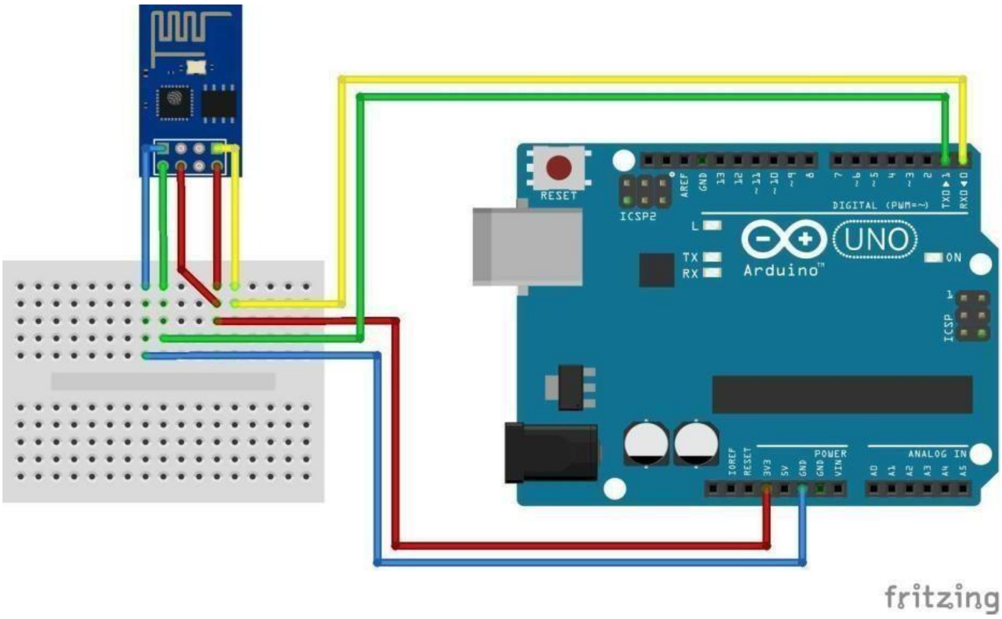
The analog pin A3 is used for connecting the display board to the Arduino UNO Board. It is illustrated as in the following figure.



(Fig: 5.5: Display Board Connection)

It is important to have a WIFI Module to send the data to the cloud platform for visualization purposes. Hence, we have used an ESP8266 WIFI Module to send data to Thingspeak IoT platform.

The WIFI module is connected to the Arduino board as follows:



(Fig : 5.6:   WIFI Module Connection)

5.4 Arduino UNO Board

5.4.1. Overview:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSPheader, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to getstarted. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 up to version R2) Programmed as a USB-to-serial converter.

Revision 3 of the board has the following new features:

Pinout: added SDA and SCL pins that are near to the AREF pin and two othernew pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board. In future, shields will be compatible with both the boards that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V.

The second one is not connected pin that is reserved for future purposes.

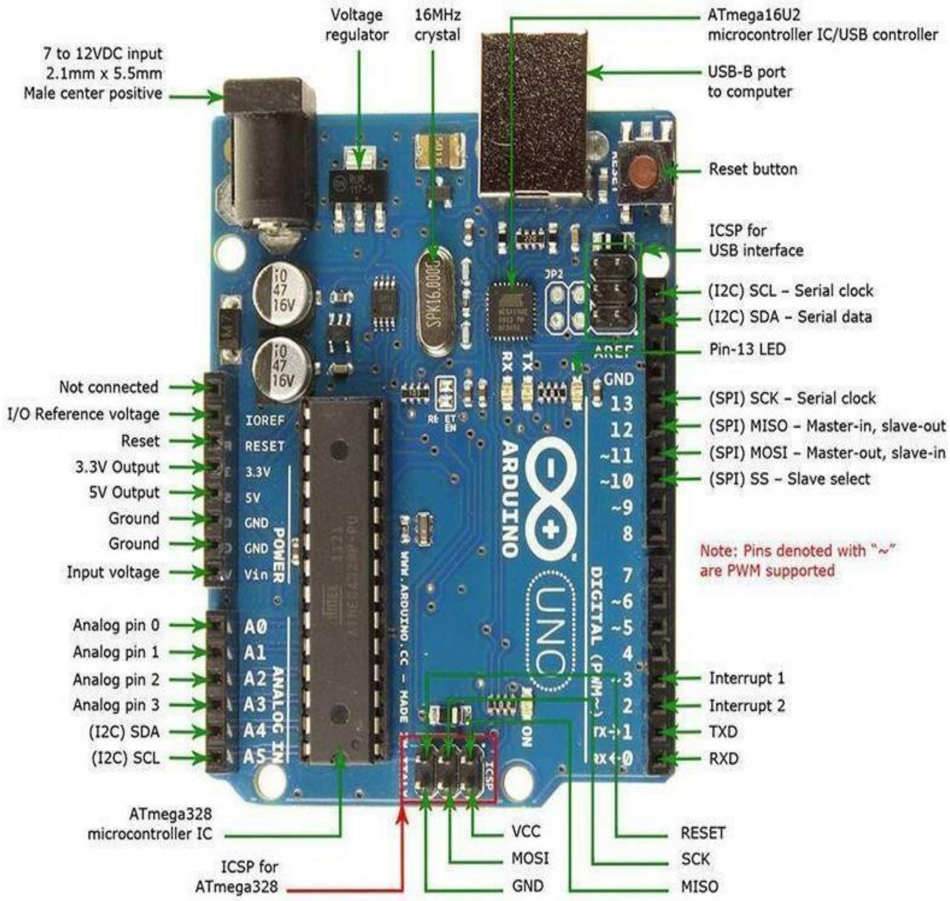
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0.

The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

5.4.2 Summary:

- Microcontroller Atmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40mA
- DC Current for 3.3V Pin 50mA
- Flash Memory 32 KB (Atmega328) of which 0.5KB used by boot loader
- SRAM 2KB (Atmega328)
- EEPROM 1KB (Atmega328)
- Clock Speed 16 MH



(Fig: 5.7: Arduino Uno PIN Description)

5.4.3 Power:

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the ground and VIN pin headers of the POWERconnector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltagevia the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from DC power jack (7-12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage viathe 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advice it.
- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum currentdraw is 50mA.
- **GND:** Ground pins.

5.4.4 Memory:

The Atmega328 has 32 KB (with 0.5 KB used for the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output: Each of the digital pins on the Uno can be used as an input or output, using pinmode (), Digital write (), and digital read () functions. They operate at 5 volts. Each pincan provide or receive a maximum of 40mA and has an internal pull-up resistor (disconnected by default) of 20-50 kms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1(TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of Atmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach interrupt () function for details.
- **PWM:** 3, 5, 6, 9, 10 and 11. Provide 8-bit PWM output with the analogue write () function. SPI: 10(SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins supportSPI communication using the SPI library.

- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analogue inputs, labeled A0 through A5, each of which provides 10 bits of resolution (i.e. 1024 different values). By default they ensure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogue reference () function. Additionally, some pins have specialized functionality:

- **TWI:** A4 or SDA pin and A5 or SCL pin. Support TWI communication using theWire library.

There are a couple of other pins on the board:

- **AREF:** Reference voltage for the analogue inputs. Used with analogue reference ().
- **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add areset button to shields which block the one on the board. See also the mappingbetween Arduino pins and ATmega328 ports. The mapping between Arduino pins and ATmega328 ports. The mapping for ATmega8, 168 and 328 is identical.

5.4.5 Communication

- The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0(RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer.
- The '16U2' firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDS on the board will

flash when the data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

- A software serial library allows for serial communication on any of theUNO's digital pins.

- The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes the wire library to simplify use of the I2C; see the documentation fordetails. For SPI communication, use the SPI library.

5.4.6 Programming:

The Arduino UNO can be programmed with the Arduino software (download). Select "Arduino UNO from the tools > Board menu" (according to the microcontroller on yourboard). For details see the reference and tutorials.

The ATmega328 on the Arduino UNO comes pre-burned with a boot larder that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference header files).

You can also bypass the boot loader and program the microcontroller through the ICSP (in circuit serial programming) header; See these instructions for details.

The ATmega16U2 (or 8U2in the rev1 and rev2 boards) firmware code is available. The Atmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

- **On REV1 boards:** connecting the shoulder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- **On REV2 or later boards:** there is a resistor that pulling the 8U2/16U2 HWB line to ground; making a teaser to put into DFU mode. You can then use atmen's FLIP (windows) or the DFU programmer (Mac os X and Linux) to load a new firmware, or u can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user tutorial for

more information. Automatic (software) reset rather than requiring a physical press of the reset button beforean upload, the Arduino UNO is design in a way that allows it to be reset by software running on a connected computer. One of the hardware flow controls line (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via 100 Nano farad capacitor. When this line is asserted (taken low), the reset lines drop long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter a timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB).For the following half a second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e., anything besides an upload of the new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to reenable it. It's labeled "RESET-EN". You may also be able to disable the autoreset by connecting a 110 ohm resistor from 5V to the reset line;

5.4.7 USB Overcurrent Protection

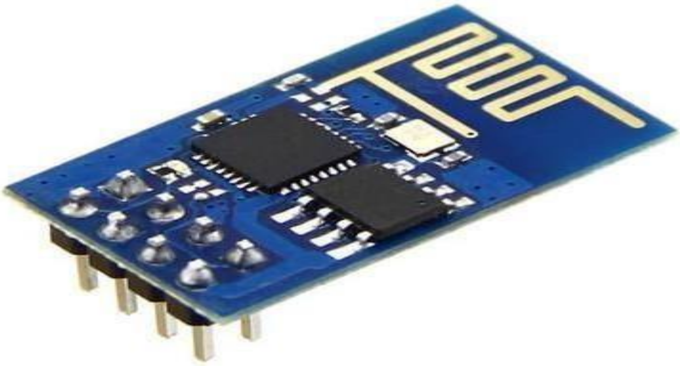
The Arduino Uno has a resettable polyfused that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

5.4.8 Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160mil (0.16”), not an even multiple of the 100mil spacing of the other pins.

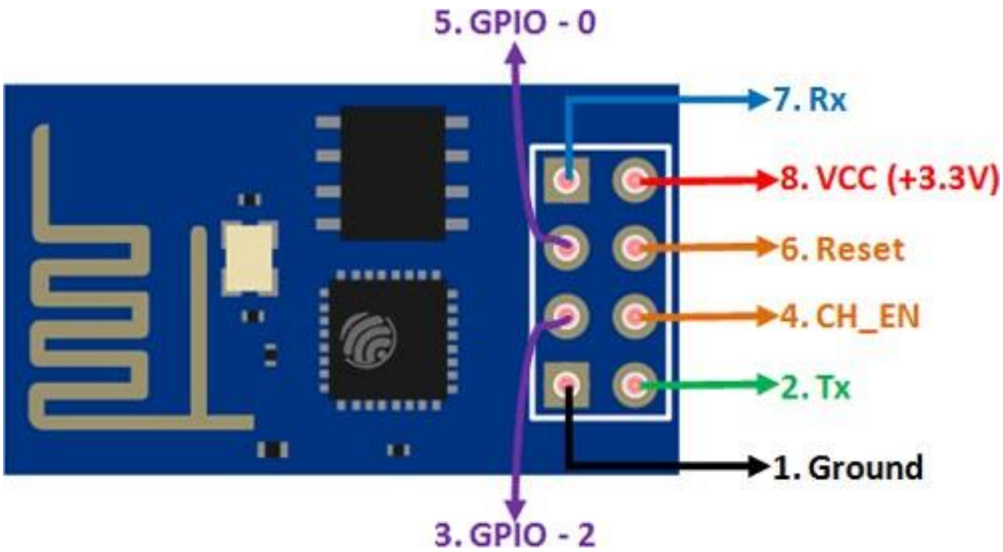
5.5 ESP8266 WIFI Module:

- The **ESP8266** is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by manufacturer Espressif Systems in Shanghai, China.
- The chip first came to the attention of Western makers in August 2014 with the **ESP-01** module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and makesimple TCP/IPconnections using Hayes-style commands.
- The **ESP8285** is an ESP8266 with 1 MiB of built-in flash, allowing for single-chip devices capable of connecting to Wi-Fi. The successor to these microcontroller chips is the ESP32, released in 2016.
- To **install** additional **ESP8266WiFi** library: Click Sketch > **Include** Library > Manage Libraries, search for **ESP8266WiFi** and then **install** withthe latest version.
- AP **mode** allows it to create its own network and have other devices (your phone) connect to it and **STA mode** allows the ESP8266 to connect to a **Wi-Fi** network (one created by your wireless router).
- **SSID** is simply the technical term for a **network** name. When you set up a wireless home **network**, you give it a name to distinguish it from other **networks** in your neighbourhood. You'll see this name when you connect your computer to your wireless **network**. WPA2 is a standard for wireless security.



(Fig 5.8: ESP8266 WIFI Module)

- This module allows microcontrollers to connect to a Wi-Fi network.
- It is mostly **used** for development of IoT (Internet of Things) embedded applications. It employs a 32-bit RISC CPU based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz)
- The ESP8266 is capable of either hosting an **application** or offloading all **Wi- Fi** networking functions from another **application** processor.



(Fig: 5.9: ESP8266 PIN Description)

5.5.1 ESP8266-01 Features:

- Low cost, compact and powerful Wi-Fi Module
- Power Supply: +3.3V only
- Current Consumption: 100mA
- I/O Voltage: 3.6V (max)
- I/O source current: 12mA (max)
- Built-in low power 32-bit MCU @ 80MHz
- 512kB Flash Memory
- Can be used as Station or Access Point or both combined
- Supports Deep sleep (<10uA)
- Supports serial communication hence compatible with many development platformlike Arduino
- Can be programmed using Arduino IDE or AT-commands or Lua Script

TABLE 5.1: ESP8266 PIN Description

Pin	Name	Description
1	TXD	1) UART_TXD, sending; 2) General Purpose Input/Output：GPIO1； 3) Pull-down is not allowed when startup;
2	GND	GND
3	CU_PD	1) Working at high level; 2) Power off when low level is supplied

4	GPIO2	1) It should be high level when power on, hardware pull-down is not allowed; 2) Pull-up by default;
5	GPIO16	External Reset signal, reset when low level is supplied; work when high level is supplied (high level by default);
6	GPIO0	1) WiFi Status indicator; 2) Operation mode selection: Pull-up: Flash Boot, operation mode Pull-down: UART Download, download mode
7	VCC	Power Supply(3.3V)
8	RXD	1) UART_RXD, Receiving 2) General Purpose Input/Output: GPIO3;

5.5.2 : Working of WIFI Module:

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook thisup to your Arduino device and get about as much WiFi- ability as a WiFi Shield offers. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on- chip integration allows for minimal external circuitry, including the front-end module,is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces,it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

5.6 Buzzer Module:

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board incombination, can complete a simple circuit design, to “plug and play”.

5.6.1 Specifications:

- On-board passive buzzer
- On-board 8550 triode drive
- Can control with single-chip microcontroller IO directly
- Working voltage: 5V
- Board size: 22 (mm) x12 (mm)

5.6.2 Working:

Buzzer is a kind of pronunciation device that can convert audio signal into sound signal.

The vibrating disk in a magnetic **buzzer** is attracted to the pole by the magnetic field. When an oscillating signal is moved through the coil, it produces a fluctuating magnetic field which vibrates the disk at a frequency equal to that of the drive signal.

(Fig: 5.10 Buzzer Module)

5.7 Dust Sensor:

Dust Sensor is a simple air monitoring module with onboard Sharp GP2Y1010AU0F. It is capable of detecting fine particle larger than 0.8μm in diameter, even like the cigarette smoke. Analog voltage output of the sensor is linear with dust density. The module has embedded voltage boost circuit to support wide range of power supply.

A dust sensor detects the dust particle concentration in air by using optical sensing method. An infrared light emitting diode (IR LED) and a photo-sensor are optically arranged in the device. The photo-sensor detects the reflected IR LED light by dust particles in air. The SMART Dust Sensor can detect the small particles like cigarette smoke and it can distinguish small particles like smoke from large house dust by pulse pattern of signal output.

It works on the principle of laser scattering. The air enters through the air inlet where a light source illuminates the particles and the scattered light is transformed into a signal by a phototransistor. These signals are amplified by the amplifier circuit and then processed to get the particle concentration. The intensity of the scattered light depends on the dust particles. More the dust particles in the air, the greater will be the intensity of light. Output voltage at the V_{OUT} pin of the sensor changes according to the intensity of scattered light.

A **sensor** and a beam of light sit at an angle to each other. As a particle passes in front of the light, some light is reflected towards the **sensor**. The **sensor** registers a pulse for as long as the particle reflects light to the **sensor**.

A dust sensor is able to distinguish small particles of cigarette smoke from large particles of house dust. The Low pulse width is proportional to the particle size and concentration. Constant forced air convection flow by heater resister in dust sensor.

Applications of Dust Sensor include:

- Air purifier
- Air conditioner
- Air monitor
- PM2.5 Detector

(Fig: 5.11: Dust Sensor)

5.8 MQ135 Gas Sensor:

Gas sensors (also known as gas detectors) are electronic devices that detect and identify different types of gasses. They are commonly used to detect toxic or explosive gasses and measure gas concentration. Gas sensors are employed in factories and manufacturing facilities to identify gas leaks, and to detect smoke and carbon monoxide in homes. Gas sensors vary widely in size (portable and fixed), range, and sensing ability. They are often part of a larger embedded system, such as hazmat and security systems, and they are normally connected to an audible alarm or interface. Because gas sensors are constantly interacting with air and other gasses, they have to be calibrated more often than many other types of sensors.

Depending on their intended environments and functions, the physical makeup and sensing process can vary notably between sensors.

The MQ-135 sensor module comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when you are only trying to detect one particular gas. It is highly sensitive to ammonia (NH_3), sulphide and benzene steams, smoke, NOx and CO_2. Low cost sensor. To measure the gases in PPM we need to use analog pin.

Sensitive for benzene, alcohol, smoke. Output voltage boosts along with the concentration of the measured gases increases. It gives a Fast response and recovery period is very small too. The sensitivity of the MQ135 gas sensor is adjustable. It also contains an output indicator and provides the signal to the user.

Ideal sensor to detect the presence of a dangerous LPG leak in your car or in a service station, storage tank environment. This unit can be easily incorporated into an alarm unit, to sound an alarm or give a visual indication of the LPG concentration. The sensor has excellent sensitivity combined with a quick response time. The sensor can also sense iso-butane, propane, LNG and cigarette smoke.

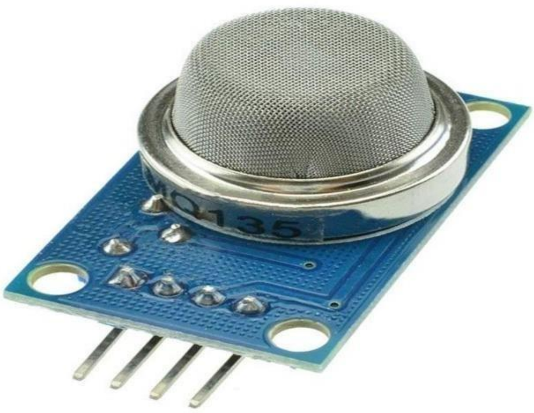
The gas sensor output terminals are connected to non inverting input terminal of the comparator.

Here the comparator is constructed with operational amplifier LM 358. The reference voltage is given to inverting input terminal. The reference voltage depends on the desired gas intensity. When there is no leakage the non inverting input is greater then inverting input so the output of the comparator is positive voltage which is given to the base of the switching transistor BC 547. Hence the transistor is conducting. Here the transistor acts as switch so the collector and emitter will be closed. The output is taken from collector terminal. Now the outputis zero which is given to hex inverter 40106.

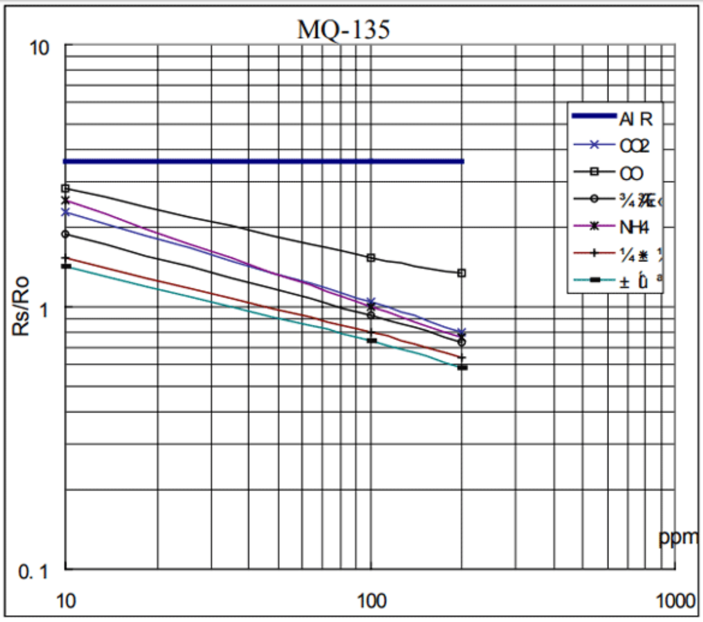
When there is gas leakage the inverting input voltage is greater than non inverting input. Now the comparator output is -12V so the transistor is in cutoff region. The 5v is given to hex inverter 40106 IC. Then the final output data is directly given to microcontroller to determine the gas leakage.

Sensitive material of MQ135 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, the sensor conductivity is more higher along with the gas concentration rising. It is with low cost and suitable for different application.

Used for family, surrounding environment noxious gas detection device, Can be applied to ammonia, aromatics, sulfur, benzene vapor, and other harmful gases/smoke, gas detection, tested concentration range: 10 to 1000ppm.



(Fig: 5.12: MQ135 Gas Sensor)



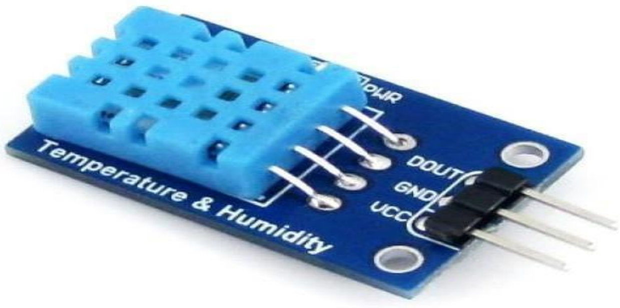
(Fig: 5.13: MQ135 Sensitivity Characteristics)

5.9 DHT11 Temperature and Humidity Sensor:

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement devices, and connected with a high- performance 8-bit microcontroller.

DHT11 sensor consists of capacitive humidity sensor and thermistor for sensing temperature. This composite sensor contains calibrated digital signal outputs of temperature and humidity. The DHT11 only returns integers (e.g. 20). The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

Various applications are: HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.



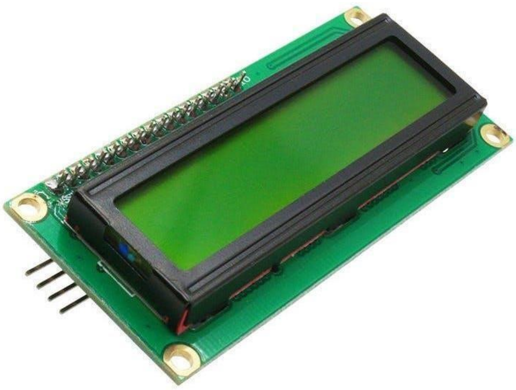
(Fig: 5.14: DHT11 Sensor)

5.10 LCD Display :

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 combined with polarizers. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in **color** or monochrome.

An **LCD** (**Liquid Crystal Display**) **screen** is an electronic **display** module and has a wide range of applications. A **16x2 LCD display** is very basic module and is very commonly used in various devices and circuits. A **16x2 LCD** means it can **display** 16 characters per line and there are 2 such lines.

A liquid crystal display (LCD) has liquid crystal material sandwiched between two sheets of glass. Without any voltage applied between transparent electrodes, liquid crystal molecules are aligned in parallel with the glass surface. When voltage is applied, they change their direction and they turn vertical to the glass surface. They vary in optical characteristics, depending on their orientation. Therefore, the quantity of light transmission can be controlled by combining the motion of liquid crystal molecules and the direction of polarization of two polarizing plates attached to the both outer sides of the glass sheets. LCDs utilize these characteristics to display images.



(Fig: 5.15: LCD Display)

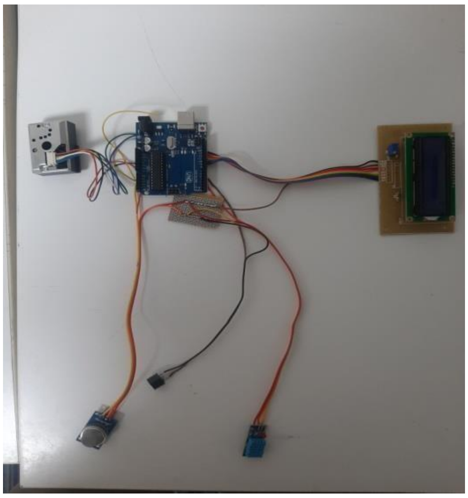
5.11 Thingspeak IoT Platform

- Thingspeak can control hardware remotely. It can display sensor data, store and visualise it, among other useful things.
- Thingspeak app : It allows you to create amazing interfaces for your projects using various widgets.
- Thingspeak server: It is responsible for all communication between the smartphone and hardware.
- Thingspeak Libraries:
    - These are available for all popular hardware platforms.
    - The libraries enable communication with the server, process all incoming and outgoing commands.

(Fig: 5.16: Thingspeak IoT Platform)

5.12 Working Explanation

- The sensors and other components are connected to the Arduino UNO Board as follows.



(Fig: 5.17: Basic Hardware Connection)

- The Arduino UNO is programmed to collect the data from the sensors (Dust Sensor, Gas Sensor, Temperature and Humidity Sensor) and process the signals.
- If the pollutant level is more than the threshold set in the Arduino program, then an alarm is raised through the buzzer module.

- The values are sent to the Thingspeak IoT Cloud Platform through WIFI Module. The WIFI credentials are set in the code. If the credentials matches with the WIFI hotspot name and password, then the WIFI module is connected to it and can now send the data to the URL, which is the Thingspeak Channel API Key. The API Key is set in the code, and if it matches to the Thingspeak Channel, then the data is processed by the IoT cloud platform.
- The user can now visualize the data in any form that the user wants to.
- The user can also export the data from Thingspeak to his system, where he can make his own predictive analysis.
- Thingspeak stores the data and displays the data in graphical format which is convenient for the user to understand the data.
- The system provides the temperature and humidity readings, amount ofdust present in air in mg / m^3units, and the amount of harmful gasespresent in air. These details can also be viewed from the LCD Display Module, which shows the real time readings. The previous data can be viewed in the Thingspeak Platform.

CHAPTER 6

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS:

The various functionalities used in the system are working fine. The Arduino UNO board is used to implement the project. The code for the system is written and embedded into the Arduino UNO board. The Arduino UNO board is programmed in such a way that it processes the input from the three sensors viz. dust sensor, temperature and humidity Sensor, and gas sensor. These three sensors detect various possibilities of air pollution and provide the input to the Arduino board for processing.

The sensors provide the input successfully to the Arduino Uno Board. The Arduino UNO board is powered by an external power supply. It processes the input data from the sensors which are in analog form. The data is processed and sent to the IoT cloud Platform “Thingspeak” through ESP8266 WiFi Module. The Thingspeak platform can be used to design our application interface, store and visualize it using various widgets. It acts as a communication medium between the hardware and the smartphone.

The WiFi module successfully transfers the data to the cloud platform (Thingspeak). The Thingspeak Platform successfully displays the data in an user friendly and visualised manner. So, the user can monitor the changes in air quality via a smart phone seamlessly through the Thingspeak IoT platform.

The output expected is working seamlessly. The idea which we wanted to implement earlier is what exactly we have done in this project successfully. So, this project overcomes the demerits that we discussed in the existing projects.

The use of cost efficient sensors makes it an affordable system which is really important for people living in metropolitan cities. The system works even during power shutdown as it also uses an external battery apart from the power supply.

CHAPTER 7

SUMMARY AND CONCLUSIONS:

Everyone would agree to the fact that a healthy surrounding is the foremost thing we need in our life. Here we provide a pollution detection system with the help of technology. This proves the fact that technology is touching each and every field which in turn shows that everything is dependent on technology. With the use of technology, our system focuses on providing the air quality index of the surroundings to the user. Since our system uses dust sensor, gas sensor and a temperature and humidity sensor, it can be availed of low cost. The system also alerts the user when the pollutants level reach a threshold value, so that the user can take necessary action or move to a safe location. For Storage of Data and detailed and visualized analysis, Thingspeak IoT platform is useful for future analytics. Many inventions are taking place in the field of technology. With much more innovations in the future, some interesting features could be added to our system. Our system is developed in such a way that it can accommodate new features for upgradation. Our air pollution monitoring system is implemented successfully and could be enhanced with further developments and extended use of technology.

REFERENCES

[1] Xiaojun, C., Xianpeng, L., & Peng, X. (2015, January). IOT-based air pollution monitoring and forecasting system. In *2015 international conference on computer and computational sciences (ICCCS)* (pp. 257-260). IEEE.

[2] Pal, P., Gupta, R., Tiwari, S., & Sharma, A. (2017). IoT based air pollution monitoring system using Arduino. *International Research Journal of Engineering and Technology (IRJET)*, *4*(10), 1137-1140.

[3] Parmar, G., Lakhani, S., & Chattopadhyay, M. K. (2017, October). An IoT based low cost air pollution monitoring system. In *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)* (pp. 524-528). IEEE.

[4] Desai, N. S., & Alex, J. S. R. (2017, March). IoT based air pollution monitoring and predictor system on Beagle bone black. In *2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)* (pp. 367-370). IEEE.

[5] Rushikesh, R., & Sivappagari, C. M. R. (2015, October). Development of IoT based vehicular pollution monitoring system. In *2015 international conference on green computing and internet of things (ICGCIoT)* (pp. 779-783). IEEE.

[6] Shah, H. N., Khan, Z., Merchant, A. A., Moghal, M., Shaikh, A., & Rane, P. (2018). IOT based air pollution monitoring system. *International Journal of Scientific & Engineering Research*, *9*(2), 62-

[7] Singh, A., Pathak, D., Pandit, P., Patil, S., & Golar, P. C. (2017). IOT based air and sound pollution monitoring system. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *6*(3).

[8] Okokpujie, K. O., Noma-Osaghae, E., Odusami, M., John, S. N., & Oluga, O. (2018). A smart air pollution monitoring system. *International Journal of Civil Engineering and Technology (IJCIET)*, *9*(9), 799-809.

[9] Gupta, H., Bhardwaj, D., Agrawal, H., Tikkiwal, V. A., & Kumar, A. (2019). An IoT Based Air Pollution Monitoring System for Smart Cities. In *2019 IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS)* (pp. 173-177). IEEE.

[10] Saha, A. K., Sircar, S., Chatterjee, P., Dutta, S., Mitra, A., Chatterjee, A., ... & Saha, H. N. (2018, January). A raspberry Pi controlled cloud based air and sound pollution monitoring system with temperature and humidity sensing. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)* (pp 607-611). IEEE.

[11] Muthukumar, S., Mary, W. S., Jayanthi, S., Kiruthiga, R., & Mahalakshmi, M. (2018, July). IoT based air pollution monitoring and control system. In *2018 International Conference on inventive research in computing applications (ICIRCA)* (pp. 1286-1288). IEEE.

[12] Ayele, T. W., & Mehta, R. (2018, April). Air pollution monitoring and prediction using IoT. In *2018 second international conference on inventive communication and computational technologies (ICICCT)* (pp. 1741-1745). IEEE.

[13] Saikumar, C. V., Reji, M., & Kishoreraja, P. C. (2017). IOT based air quality monitoring system. *International Journal of Pure and Applied Mathematics*, *117*(9), 53-57.

[14] BC, K., & Jose, D. (2018). IoT Based Pollution Monitoring System using Raspberry-PI. *International Journal of Pure and Applied Mathematics*, *118*(24).

[15] Zakaria, N. A., Abidin, Z. Z., Harum, N., Hau, L. C., Ali, N. S., & Jafar, F. A. (2018). Wireless internet of things-based air quality device for smart pollution monitoring. *Int. J. Adv. Comput. Sci. Appl*, *9*(11), 65-69.

[16] Grace, R. K., & Manju, S. (2019). A comprehensive review of wireless sensor networks based air pollution monitoring systems. *Wireless Personal Communications*, *108*(4), 2499-2515.

[17] Senthilkumar, R., Venkatakrishnan, P., & Balaji, N. (2020). Intelligent based novel embedded system based IoT enabled air pollution monitoring system. *Microprocessors and Microsystems*, *77*, 103172.

[18] Dhingra, S., Madda, R. B., Gandomi, A. H., Patan, R., & Daneshmand, M. (2019). Internet of Things mobile–air pollution monitoring system (IoT-Mobair). *IEEE Internet of Things Journal*, *6*(3), 5577-5584.

[19] Kumar, H., Karthikeyan, S., Manjunath, G. M., & Rakshith, M. S. (2018). Solar power environmental air pollution & water quality monitoring system based on IoT. *Perspectives in Communication, Embedded-systems and Signal-processing-PiCES*, *2*(8), 197-199.

[20] Chowdhury, S., Islam, M. S., Raihan, M. K., & Arefin, M. S. (2019, September). Design and Implementation of an IoT Based Air Pollution Detection and Monitoring System. In *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)* (pp. 296-300). IEEE.

[21] Alam, S. S., Islam, A. J., Hasan, M. M., Rafid, M. N. M., Chakma, N., & Imtiaz, M. N. (2018, September). Design and development of a low-cost IoT based environmental pollution monitoring system. In *2018 4th international conference on electrical engineering and information & communication technology (iCEEiCT)* (pp. 652-656). IEEE.

[22] Sai, K. B. K., Subbareddy, S. R., & Luhach, A. K. (2019). IOT based air quality monitoring system using MQ135 and MQ7 with machine learning analysis. *Scalable Computing: Practice and Experience*, *20*(4), 599-606.

[23] Yamunathangam, D., Pritheka, K., & Varuna, P. (2019). IoT enabled air pollution monitoring and awareness creation system. *Int. Jrnl. of Recent Techn. and Eng*, *7*(4), 398-400.

[24] Srivastava, C., Singh, S., & Singh, A. P. (2020). IoT-enabled air monitoring system. In *Intelligent Systems, Technologies and Applications* (pp. 173- 180). Springer, Singapore.

[25] https://www.arduino.cc/reference/en/

[26] https://www.tutorialspoint.com/arduino/index.htm

[27] https://create.arduino.cc/projecthub

[28] https://www.arduino.cc/en/Main/Software

[29] https://thingspeak.com/

[30] https://electronut.in/

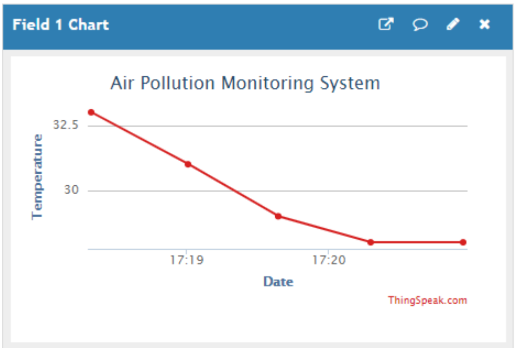
[31] https://maker.pro/arduino/projects

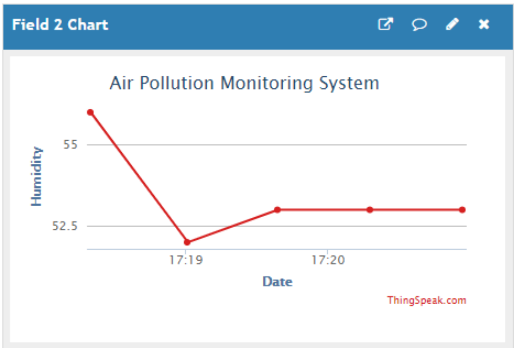
APPENDIX

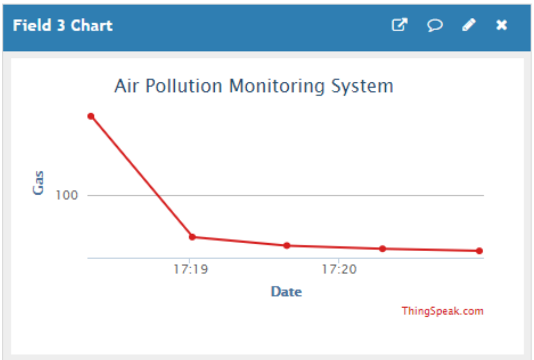
IMPLEMENTED SCREENSHOTS:

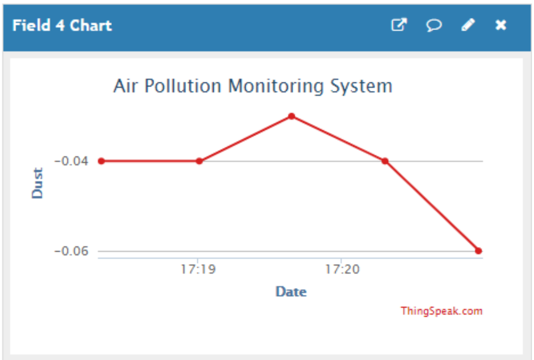
Graphical representation of variation in data :

(Screenshots taken from Thingspeak IoT Platform)





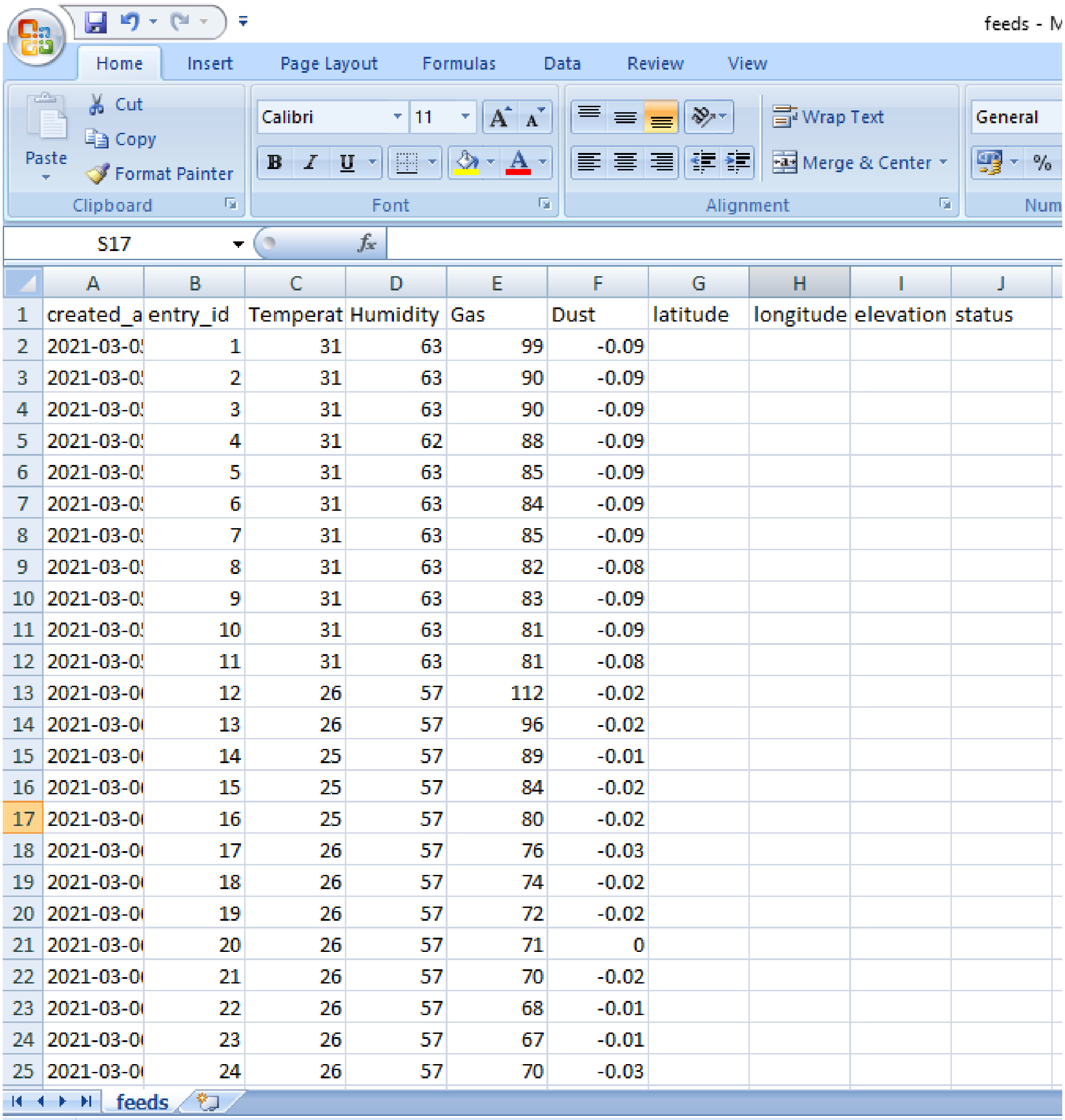




X- Axis: Temperature, Humidity, Gas and Dust

Y-Axis: Time(in hh:mm 24-Hour Format)

Data Exported from Thingspeak IoT Platform:



	A	B	C	D	E	F	G	H	I	J
1	created_a	entry_id	Temperat	Humidity	Gas	Dust	latitude	longitude	elevation	status
2	2021-03-0!	1	31	63	99	-0.09				
3	2021-03-0!	2	31	63	90	-0.09				
4	2021-03-0!	3	31	63	90	-0.09				
5	2021-03-0!	4	31	62	88	-0.09				
6	2021-03-0!	5	31	63	85	-0.09				
7	2021-03-0!	6	31	63	84	-0.09				
8	2021-03-0!	7	31	63	85	-0.09				
9	2021-03-0!	8	31	63	82	-0.08				
10	2021-03-0!	9	31	63	83	-0.09				
11	2021-03-0!	10	31	63	81	-0.09				
12	2021-03-0!	11	31	63	81	-0.08				
13	2021-03-0(	12	26	57	112	-0.02				
14	2021-03-0(	13	26	57	96	-0.02				
15	2021-03-0(	14	25	57	89	-0.01				
16	2021-03-0(	15	25	57	84	-0.02				
17	2021-03-0(	16	25	57	80	-0.02				
18	2021-03-0(	17	26	57	76	-0.03				
19	2021-03-0(	18	26	57	74	-0.02				
20	2021-03-0(	19	26	57	72	-0.02				
21	2021-03-0(	20	26	57	71	0				
22	2021-03-0(	21	26	57	70	-0.02				
23	2021-03-0(	22	26	57	68	-0.01				
24	2021-03-0(	23	26	57	67	-0.01				
25	2021-03-0(	24	26	57	70	-0.03				

SOURCE CODE:

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

#include "DHT.h"

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
SoftwareSerial esp(3, 2);

//btcs
#define splash splash1

String network = "Dust";
String password = "1234567890";
#define IP "184.106.153.149"
String GET = "GET /update?api_key=LOHXFQKTO4W6S8SO";

#define Gas A0
#define DHTPIN A1
#define dusled A3
#define dussens A2
#define buz 7

#define DHTTYPE DHT11
int a, b;

DHT dht(DHTPIN, DHTTYPE);

int DispDelay;
int gas_r, hum;
float dus_out;
float temp;

int tu = 35;
String out;
void setup()
 {

  // setup code, to run once:

  Serial.begin(115200);
  dht.begin();
  pinMode(Gas, INPUT);
  pinMode(buz, OUTPUT);
  digitalWrite(buz, LOW);
  esp.begin(115200);
  LcDSet();
  setupEsp8266();
```

```
  lcd.clear();

}

void LcDSet() {

  lcd.begin(16, 2);

  // Print a message to the LCD.

  lcd.clear();

  splash(0, "Air Pollution");
  splash(1, "Monitoring");
  delay(1000);

}


void loop() {
  tu++;
  setDisp();
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  gas_r = analogRead(Gas);
  dus_out = dust(dusled, dussens);
  temp = t;
  hum = h;
  if (gas_r > 550) {
    digitalWrite(buz, HIGH);
    delay(100);
    digitalWrite(buz, LOW);
    delay(25);
    digitalWrite(buz, HIGH);
    delay(100);
    digitalWrite(buz, LOW);
    delay(25);
    digitalWrite(buz, HIGH);
  }
  else {
    digitalWrite(buz, LOW);
  }
  Serial.println(dus_out);
  delay(300);
  if (tu >= 60) {
    send2();
    tu = 0;
  }
```

```
}

void setDisp() {

  if (DispDelay >= 0 and DispDelay < 20) {

    display_T(temp, hum);

  }
  else if (DispDelay >= 20 and DispDelay < 40) {

    display_ard(gas_r, dus_out);
  }
  else  {
    DispDelay = 0;
  }


  DispDelay++;
}

void send2() {
  splash(0, "Please Wait");
  splash(1, "Sending Data");
  updateServer(String(temp), String(hum), String(gas_r), String(dus_out));

  delay(3000);
  lcd.clear();
}
void display_ard(int val1 , float val2) {

  lcd.setCursor(0, 0);
  lcd.print("G:              ");
  lcd.setCursor(3, 0);
  lcd.print(val1);
  lcd.setCursor(0, 1);
  lcd.print("D:              ");
  lcd.setCursor(3, 1);
  lcd.print(val2);
  lcd.print(" mg/m3 ");

}

void display_T(float val1 , int val2) {


  lcd.setCursor(0, 0);
  lcd.print("T:              ");
  lcd.setCursor(3, 0);
  lcd.print(val1);
```

```
  lcd.print(char(223));
  lcd.print(" C ");
  lcd.setCursor(0, 1);
  lcd.print("H:                ");
  lcd.setCursor(3, 1);
  lcd.print(val2);
  lcd.print(" % ");

}
```

PAPER PUBLICATION

AN IOT BASED AIR POLLUTION MONITORING SYSTEM

Ashwath R, Aqeeb Khan**,** Dr. S. Vigneshwari, ME., Ph.D.

Department of Computer Science and Engineering

Sathyabama Institute of Science and Technology, Chennai

ABSTRACT: Today technology has become an integral part of our day to day life. It is important for us to exploit the technological advancements as much as possible in order to gain more benefits. We all are aware of the fact that in places like Delhi and other metropolitan cities, the air quality index is changing from bad to worse. People living in such places should necessarily carry an air pollution monitoring system consisting of all facilities to keep track of airquality index and take necessary steps to improve quality of air in immediate surroundings. There is a wide range of scope for this product to be used in many situations. This system can be upgraded into a more advanced system with added features in the future.

Introduction:

A healthy environment is the first and foremost thing for our happiness. We need a pollution free surrounding for living a safe and secure life. The recent increase in pollution levels in metropolitan cities, especially in Delhi, is a worrying sign. In the world that is advancing rapidly with technology where cars could drive on their own and drones could capture your food, air pollution should not be of much concern but the above statistics just proves it wrong. Our application is one such thing which can provide the surrounding air quality index to the user. This is a basic level system which notifies the user of the various pollutants and their levels present in air.

Related work:

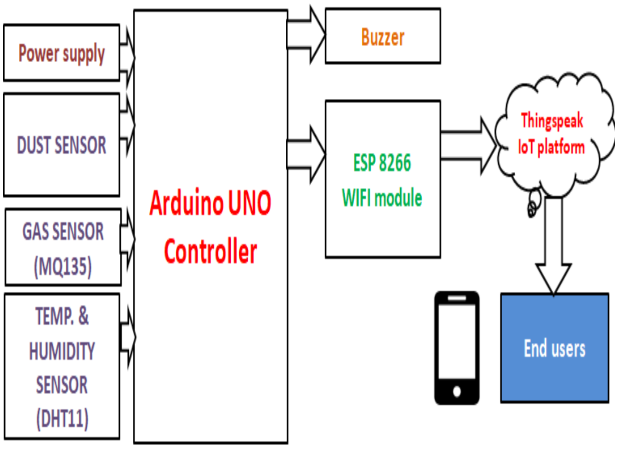
1. We have taken a cue from works done by Parveen Sulthana, PramodSharma and Vijayakumar Sajjan. We have taken those works as a reference and have done it using different methodologies apart from doing as much as possible to eliminate the cons in the related work. There were other interesting works focusing on air pollution detection.
2. Aravindhar, D. J. (2019, November). Iot Based Air Pollution Monitoring System Using Esp8266-12.
3. Kim, S. H., Jeong, J. M., Hwang, M. T., & Kang, C. S. (2017, October). Development of an IoT-based atmospheric environment monitoring system.

1. Nasution, T. H., Muchtar, M. A., & Simon, A. (2019, October). Designing an IoT-based air quality monitoring system.
2. One such work is an IoT-based real time air pollution monitoring system proposed to monitor the pollution levels of various pollutants in Coimbatore city. The geographical area is classified as industrial, residential and traffic zones. It proposes an IoT system that could be deployed at any location and store the measured value in a cloud database, perform pollution analysis, and display the pollution level at any given location.
3. Since the existing work involves too much cost apart from complex implementation and high maintenance, we have eliminated all such factors. Also, our work is done in such a way that it can accommodate further enhancementsto it. Ours is a work which focuses on providing a low cost air pollution monitoring system for regular and daily usage for people living in metropolitan and highly polluted cities to be aware of their surroundings.

Proposed work:

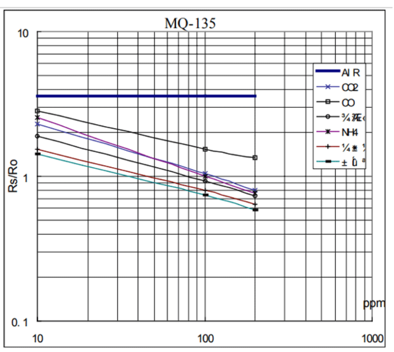
1. We are here with an innovation which emphasizes on providing an IoT-Enabled Air Pollution Meter With Digital Dashboard On Smartphone.
2. To display real-time air quality readings for the immediate surroundings.
3. This is done to have a better understanding of our surroundings and ensure the quality of air is not compromised by constantly monitoring air quality on our smart phone.

System Architecture Diagram:



MQ135 Gas Sensor:

The MQ-135 sensor module comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when you are only trying to detect one particular gas. It is highly sensitive to ammonia (NH3), sulphide and benzene steams, smoke, NOx and CO2. Low cost sensor. To measure the gases in PPM we need to use analog pin.

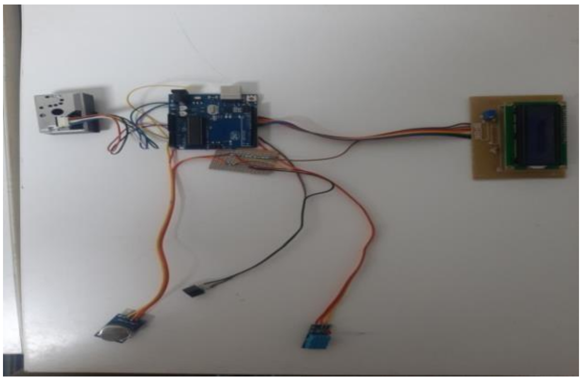


DHT11 Temperature and Humidity Sensor:

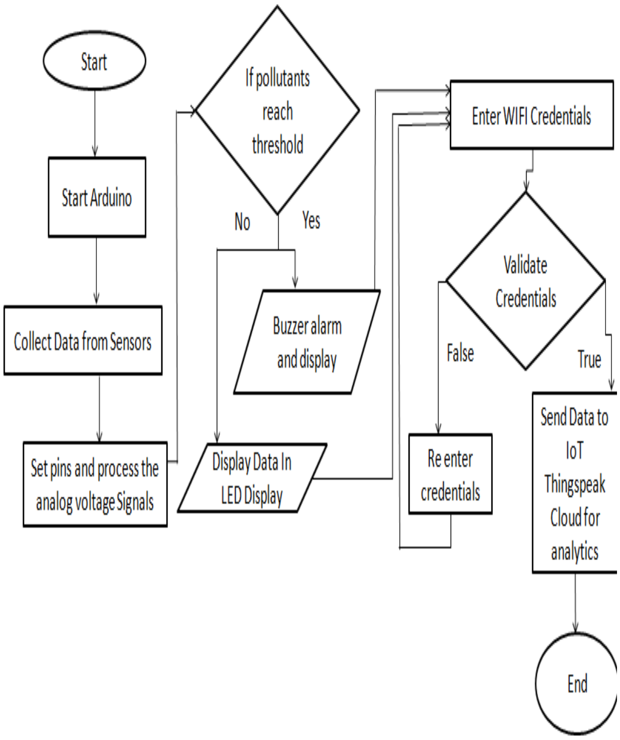
DHT11 sensor consists of capacitive humidity sensor and thermistor for sensing temperature. This composite sensor contains calibrated digital signal outputs of temperature and humidity. The DHT11 only returns integers (e.g. 20). The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

Dust Sensor:

It works on the principle of laser scattering. The air enters through the air inlet where a light source illuminates the particles and the scattered light is transformed into a signal by a phototransistor. These signals are amplified by the amplifier circuit and then processed to get the particle concentration. The intensity of the scattered light depends on the dust particles. More the dustparticles in the air, the greater will be the intensity of light. Output voltage at the V_{OUT} pin of the sensor changes according to the intensity of scattered light.



Flowchart:



Results and Discussion:

1. The Arduino UNO board is used to implement the project. It is programmed in such a way that it processes the input from the three sensors viz. dust sensor, temperature and humidity sensor, and gas sensor. These three sensors detect various possibilities of air pollution and provide the input to the Arduino board for processing.

2. The sensors provide the input successfully to the Arduino Uno Board.

3. The Arduino UNO board is powered by an external power supply. It processes the input data from the sensors which are in analog form.

4. The data is processed and sent to the IoT cloud Platform “Thingspeak” through ESP8266 WiFi Module. The Thingspeak platform can be used to design our application interface, store and visualize it using various widgets. It acts as a communication medium between the hardware and the smartphone.

5. The WiFi module successfully transfers the data to the cloud platform (Thingspeak). The Thingspeak Platform successfully displays the data in an user friendly and visualised manner.

6. So, the user can monitor the changes in air quality via a smart phone seamlessly through the Thingspeak IoT platform.

Conclusion:

1. Everyone would agree to the fact that a healthy surrounding is the foremost thing we need in our life. Here we provide a pollution detection system with the help of technology.

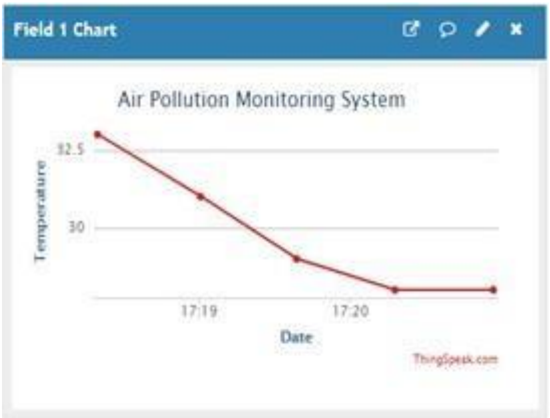
2. With the use of technology, our system focuses on providing the air quality index of the surroundings to the user.

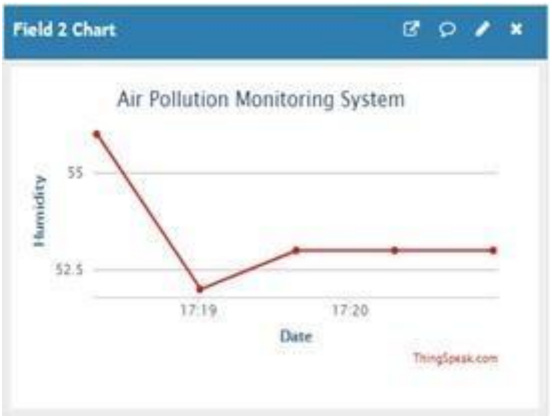
3. Since our system uses dust, gas and temperature and humidity sensors, it can be availed of low cost.

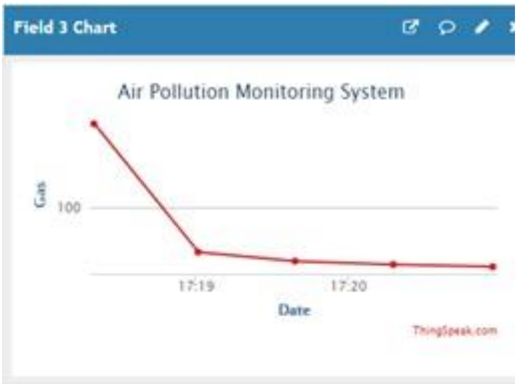
4. This can alert the user by providing the air quality index so that the user can take necessary action or move to a safe location.

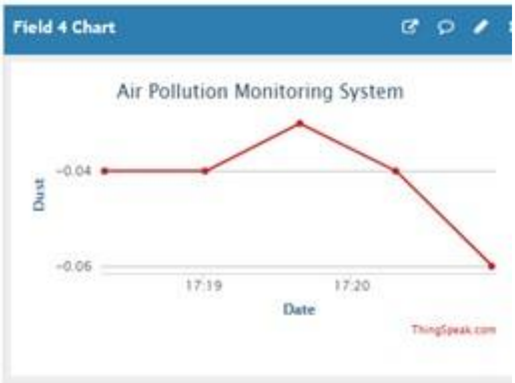
5. Many inventions are taking place in the field of technology. With much more innovations in the future, some interesting features could be added to our system.

6. For Storage of Data and detailed and visualized analysis, Thingspeak IoT platform is useful for future analytics.









1. Our system is developed in such a way that it can accommodate new features for upgradation. Our system is implemented successfully and could be enhanced with further developments and extended use of technology.

References:

[1] Xiaojun, C., Xianpeng, L., & Peng, X. (2015, January). IOT-based air pollution monitoring and forecasting system. In *2015 international conference on computer and computational sciences (ICCCS)* (pp. 257-260). IEEE.

[2] Pal, P., Gupta, R., Tiwari, S., & Sharma, A. (2017). IoT based air pollution monitoring system using Arduino. *International Research Journal of Engineering and Technology (IRJET)*, *4*(10), 1137-1140.

[3] Parmar, G., Lakhani, S., & Chattopadhyay, M. K. (2017, October). An IoT based low cost air pollution monitoring system. In *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)* (pp. 524-528). IEEE.

[4] Desai, N. S., & Alex, J. S. R. (2017, March). IoT based air pollution monitoring and predictor system on Beagle bone black. In *2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)* (pp. 367-370). IEEE.

[5] Rushikesh, R., & Sivappagari, C. M. R. (2015, October). Development of IoT based vehicular pollution monitoring system. In *2015 international conference on green*

computing and internet of things (ICGCIoT) (pp. 779-783). IEEE.

[6] Shah, H. N., Khan, Z., Merchant, A. A., Moghal, M., Shaikh, A., & Rane, P. (2018). IOT based air pollution monitoring system. *International Journal of Scientific & Engineering Research*, *9*(2), 62-

[7] Singh, A., Pathak, D., Pandit, P., Patil, S., & Golar, P. C. (2017). IOT based air and sound pollution monitoring system. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *6*(3).

[8] Okokpujie, K. O., Noma-Osaghae,E., Odusami, M., John, S. N., & Oluga, O. (2018). A smart air pollution monitoring system. *International Journal of Civil Engineering and Technology (IJCIET)*, *9*(9), 799-809.

[9] Gupta, H., Bhardwaj, D., Agrawal,H., Tikkiwal, V. A., & Kumar, A. (2019). An IoT Based Air Pollution Monitoring System for Smart Cities. In *2019 IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS)* (pp. 173-177). IEEE.

[10] Saha, A. K., Sircar, S., Chatterjee, P., Dutta, S., Mitra, A., Chatterjee, A., ... & Saha, H. N. (2018, January). A raspberry Pi controlled cloud based air and sound pollution monitoring system with temperature and humidity sensing. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 607-611). IEEE.

[11] Muthukumar, S., Mary, W. S., Jayanthi, S., Kiruthiga, R., & Mahalakshmi, M. (2018, July). IoT based air pollution monitoring and control system. In *2018 International Conference on inventive research in computing applications (ICIRCA)* (pp. 1286-1288). IEEE.

[12] Ayele, T. W., & Mehta, R. (2018, April). Air pollution monitoring and prediction using IoT. In *2018 second international conference on inventive communication and computational technologies (ICICCT)* (pp. 1741-1745). IEEE.

[13] Saikumar, C. V., Reji, M., & Kishoreraja, P. C. (2017). IOT based air quality monitoring system. *International Journal of Pure and Applied Mathematics*, *117*(9), 53-57.

[14] BC, K., & Jose, D. (2018). IoT Based Pollution Monitoring System using Raspberry-PI. *International Journal of Pure and Applied Mathematics*, *118*(24).

[15] Zakaria, N. A., Abidin, Z. Z., Harum, N., Hau, L. C., Ali, N. S., & Jafar, F. A. (2018). Wireless internet of things-based air quality device for smart pollution monitoring. *Int. J. Adv. Comput. Sci. Appl*, *9*(11), 65-69.

[16] Grace, R. K., & Manju, S. (2019). A comprehensive review of wireless sensor networks based air pollution monitoring systems. *Wireless Personal Communications*, *108*(4), 2499-2515.

[17] Senthilkumar, R., Venkatakrishnan, P., & Balaji, N. (2020). Intelligent based novel embedded system based IoT enabled air pollution monitoring system. *Microprocessors and Microsystems*, *77*, 103172.

[18] Dhingra, S., Madda, R. B., Gandomi, A. H., Patan, R., & Daneshmand, M. (2019). Internet of Things mobile–air pollution monitoring system (IoT-Mobair). *IEEE Internet of Things Journal*, *6*(3), 5577-5584.

[19] Kumar, H., Karthikeyan, S., Manjunath, G. M., & Rakshith, M. S. (2018). Solar power environmental air pollution & water quality monitoring system based onIoT. *Perspectives in Communication, Embedded-systems and Signal-processing-PiCES*, *2*(8), 197-199.

[20] Chowdhury, S., Islam, M. S., Raihan, M. K., & Arefin, M. S. (2019, September). Design and Implementation of an IoT Based Air Pollution Detection and Monitoring System. In *2019 5th International Conference on Advances inElectrical Engineering (ICAEE)* (pp. 296-300). IEEE.

[21] Alam, S. S., Islam, A. J., Hasan, M. M., Rafid, M. N. M., Chakma, N., & Imtiaz, M. N. (2018, September). Design and development of a low-cost IoT based environmental pollution monitoring system. In *2018 4th international conference on electrical engineering and information & communication technology (iCEEiCT)* (pp. 652-656). IEEE.

[22] Sai, K. B. K., Subbareddy, S. R., & Luhach, A. K. (2019). IOT based air quality monitoring system using MQ135 and MQ7 with machine learning analysis. *Scalable Computing: Practice and Experience*, *20*(4), 599-606.

[23] Yamunathangam, D., Pritheka, K., & Varuna, P. (2019). IoT enabled air pollution monitoring and awareness creation system. *Int. Jrnl. of Recent Techn. and Eng*, *7*(4), 398-400.

[24] Srivastava, C., Singh, S., & Singh, A. P. (2020). IoT-enabled air monitoring system. In *Intelligent Systems, Technologies and Applications* (pp. 173-180). Springer, Singapore.

PLAGIARISM REPORT