IOT BASED AIR POLLUTION MONITORING SYSTEM

submitted in partially fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science Engineering

By

Pushpak Kumar Mishra(17SCSE101975)

Under the Guidance of

R. RAJKUMAR

Assistant Professor

School of Computer Science and Engineering

(*B. Tech SCSE*)

Panelist

Ms. N. Gayathri

Assistant Professor

School of Computer Science and Engineering

(*B. Tech SCSE*)

Galgotias University, Greater Noida.

DECLARATION

I hereby declare that the thesis entitled "**IOT Based Air Pollution Monitoring System**" submitted by me, for the award of the degree of *Bachelor of Technology in Computer Science Engineering* to Galgotias University, Greater Noida is a record of bonafide work carried out by me under the supervision of R Rajkumar.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Greater Noida

Date : **Signature of the Candidates**

CERTIFICATE

This is to certify that the thesis entitled "**IOT Based Air Pollution Monitoring System**" submitted by Pushpak Kumar Mishra, School Of Computer Science and Engineering, Galgotias University, Greater Noida, for the award of the degree of *Bachelor of Technology in Computer Science Engineering*, is a record of bonafide work carried out by him under my supervision, as per University code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Greater Noida

Date :

Signature of the Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

Head of the Department

ABSTRACT

The level of pollution is increasing rapidly due to factors like industries, urbanization, increasing in population, vehicle use which can affect human health .We are going to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a web server using internet and will trigger alarm when the air quality goes down beyond a certain level , means when there are sufficient amount of harmful gases are present in the air like CO_2, smoke, alcohol, benzene and NH_3. It will show the air quality in PPM and the LCD and as well as on webpage so that we can monitor it very easily.

We have used MQ135 sensor which is the best choice for monitoring Air Quality as it can detect most harmful gases and can measure their amount accurately. In this IOT project, you can monitor the pollution level from anywhere using your mobile or computer. We can install this system anywhere and can also trigger some devices when pollution goes beyond some level , like we can switch on the Exhaust fan or can send alert SMS/Mail to the user.

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to my beloved **Chairman Shri. Suneel Galgotia and CEO Shri. Dhruv Galgotia** for proving necessary facilities to carry out and finish the project successfully. I am grateful to my **Vice Chancellor Dr. Preeti Bajaj** for his support and encouragement.

My special thank goes to Mr. Munish Sabharwal **Dean, Computer Science Engineering,** for providing an environment that encouraged me in working towards my goal and supported in completion of my project. This project gave me an opportunity to apply creative and critical thinking skills.

I would like to present my deep sense of gratitude to my project guide, **R. Rajkumar , Assistant Professor, School of Computer Science Engineering** who has always been a source of motivation and firm support for carrying out the project. The supervision and support that he gave truly helped in the progression of the project. His cooperation is much indeed appreciated. I am highly obliged to him for his valuable advices and moral support during research period.

Finally, my greatest and special gratitude goes to my family for their love and support.

Place: Greater Noida   Pushpak Kumar Mishra
Date:

CONTENT

Chapter No.	Particulars	Page No.
	Abstract	**4**
	Acknowledgement	**5**
	Table of Content	**6**
	List of Figure	**8**
	List of Table	**9**
	List of Abbreviations	**10**
1.	**Introduction**	**11**
	1.1 Internet of Things	**11**
	1.1.1 Trends and Characteristics	**12**
	1.1.2 Enabling Technologies for IOT	**14**
	1.1.3 Security of IOT	**15**
	1.1.4 Privacy of IOT	**16**
	1.2 Arduino	**16**
	1.2.1 Hardware of Arduino	**17**
	1.2.2 Software of Arduino	**18**
	1.2.3 Applications	**20**
	1.3 Overview of Project	**20**
	1.4 Objective	**20**
	1.5 Motivation and Contribution	**20**
	1.6 Thesis Outline	**21**
2.	**Literature Review**	**22**
	2.1 Introduction	**22**
	2.2 Why This Project	**22**
	2.3 Existing Model	**22**
	2.4 Air Quality Parameter	**23**
	2.5 Advantage of pure air	**24**
	2.6 Classification and Analysis of Different Techniques	**25**
3.	**Proposed Model and Component Description**	**30**
	3.1 Proposed Model	**30**
	3.2 Block Diagram	**31**
	3.3 Hardware Requirement	**32**
	3.3.1 For Different Parameter Sensing	**32**
	3.3.2 For Power Supply	**32**
	3.4 Software Requirement	**32**
	3.5 Component Description	**33**
	3.6 Cost Estimation	**40**
	3.7 Work Plan	**40**
4.	**Circuit Diagram and Code Implementation**	**41**
	4.1 Work Done	**41**

	4.1.1 Connection	**41**
	4.1.2 Flowchart for displaying the Temperature , Humidity and Air Quality on LCD	**43**
	4.1.3 Flowchart for Controlling the speed of fan based on Temperature	**44**
	4.1.4 Flowchart for send the Temperature , Humidity and Air Quality on THINGSPEAR	**45**
	4.2 Work Explaination	**46**
	4.3 Code Explaination	**46**
	4.4 Advantages	**51**
	4.5 Application	**51**
5.	**Testing and Result**	**52**
	5.1 Testing and Output of the Project	**52**
	5.1.1 Code	**53**
	5.2 Output	**56**
	5.3 Result and Description	**57**
	5.4 Conclusion	**57**
	References	**58**

LIST OF FIGURES

Figure No.	Title	Page No.
1	Early Arduino Board	**17**
2	Official Arduino UNO R2	**18**
3	Power and User LED Attached to pin 13	**19**
4	Block Diagram of an Existing Air Quality Monitor	**23**
5	Block Diagram of a Proposed Model	**30**
6	Block Diagram of an Air Quality Monitoring Sensing	**31**
7	Block Diagram of Sending the data to THINGSPEAK using NodeMCU	**31**
8	DTH11	**33**
9	MQ135 Air Quality Sensor	**34**
10	2N2222 Transistor	**35**
11	DC Fan with DC Motor	**35**
12	Potentiometer	**36**
13	16X2 LCD Panel	**37**
14	Arduino Uno	**38**
15	NodeMCU	**39**
16	Connection of DTH11 with Arduino	**41**
17	Connection of LCD with Arduino	**41**
18	Connection of DHT11 and MQ135 with NodeMCU	**42**
19	Connection of DTH11 with fan	**42**
20	Output testing window 1	**52**
21	Output Testing Window 2	**52**
22	Output of DHT11 sensor	**56**
23	Output of DHT11 and MQ135 in THINGSPEAK app	**56**

LIST OF TABLES

Table No.	Title	Page No.
1	Classification and Analysis of existing Air Pollution Monitoring Techniques	25
2	Pin description of 16X2 LCD Panel	37
3	Pin description of Arduino UNO	38
4	Technical Specification of Arduino UNO	39
5	Cost of different Components	40
6	Work Plan	40

LIST OF ABBREVIATIONS

IOT	INTERNET OF THINGS
LCD	LIQUID CRYSTAL DISPLAY
LED	LIGHT EMITTING DIODE
UIDs	UNIQUE IDENTIFIER
HVAC	HEAT VENTILATION AND AIR CONDITIONING
LSTM	SERVICE ORIENTED ARCHITECTURE
SOA	STRUCTURED QUERRY LANGUAGE
HTTPs	HYPER TEXT TRANFER PROTOCOL SECURE
IETF	INTERNET ENGINEERING TASK FORCE
MQTT	MESSAGE QUEING TELEMETRY TRANSPORT
SDN	SOFTWARE DEFINED NETWORKING
BLE	BLUETOOTH LOW ENERGY
Li-Fi	LIGHT FIDELITY
NFC	NEAR FIELD COMMUNICATION
RFID	RADIO FREQUENCY IDENTIFICATION
SQL	STRUCTURED QUERY LANGUAGE
IEEE	INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERING
LTE	LOW TERM EVOLUTION
LPWAN	LOW POWER WIDE AREA NETWORKING
VSAT	VERY SMALL APERTURE TERMINAL
PLC	POWER LINE COMMUNICATION

Chapter 1

INTRODUCTION

Continued exposure to environments with poor air quality is a major public health concern in developed and developing countries. It is estimated that the pollutants responsible for poor air quality cause nearly 2.5 million premature deaths per year world-wide. Significantly, around 1.5 million of these deaths are due to polluted indoor air, and it is suggested that poor indoor air quality may pose a significant health risk to more than half of the world's population. Due to its link with 11ulfil11ialization, societal health problems associated with poor air quality disproportionately affects developed and developing nations – it is estimated that air pollution is responsible for the premature deaths. Remedial action to improve air quality is often easy to implement once airborne pollutants have been detected.

The main objective of IOT Air Monitoring System is that the Air pollution is a growing issue these days. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Due to flexibility and low cost Internet of things (IoT) is getting popular day by day. With the urbanization and with the increase in the vehicles on road the atmospheric conditions have considerably affected. Harmful effects of pollution include mild allergic reactions such as irritation of the throat eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma. Monitoring gives measurements of air pollutant and sound pollution concentrations, which can then be analysed interpreted and presented. This information can then be applicable in many ways. Analysis of monitoring data allows us to assess how bad air pollution is from day to day.

1.1 INTERNET OF THINGS:

The **Internet of things** (**IoT**) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

The definition of the Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers.

There are a number of serious concerns about dangers in the growth of IoT, especially in the areas of privacy and security, and consequently industry and governmental moves to address these concerns have begun.

1.1.1 TRENDS AND CHARACTERISTICS:

The IoT's major significant trend in recent years is the explosive growth of devices connected and controlled by the Internet. The wide range of applications for IoT technology mean that the specifics can be very different from one device to the next but there are basic characteristics shared by most.

The IoT creates opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions.

The number of IoT devices increased 31% year-over-year to 8.4 billion in the year 2017 and it is estimated that there will be 30 billion devices by 2020. The global market value of IoT is projected to reach $7.1 trillion by 2020.

a). Intelligence

Ambient intelligence and autonomous control are not part of the original concept of the Internet of things. Ambient intelligence and autonomous control do not necessarily require Internet structures, either. However, there is a shift in research (by companies such as Intel) to integrate the concepts of the IoT and autonomous control, with initial outcomes towards this direction considering objects as the driving force for autonomous IoT. A promising approach in this context is deep reinforcement learning where most of IoT systems provide a dynamic and interactive environment. Training an agent (i.e., IoT device) to behave smartly in such an environment cannot be addressed by conventional machine learning algorithms such as supervised learning. By reinforcement learning approach, a learning agent can sense the environment's state (e.g., sensing home temperature), perform actions (e.g., turn HVAC on or off) and learn through the maximizing accumulated rewards it receives in long term.

IoT intelligence can be offered at three levels: IoT devices, Edge/Fog nodes, and Cloud computing. The need for intelligent control and decision at each level depends on the time sensitiveness of the IoT application. For example, an autonomous vehicle's camera needs to make real-time obstacle detection to avoid an accident. This fast decision making would not be possible through transferring data from the vehicle to cloud instances and return the predictions back to the vehicle. Instead, all the operation should be performed locally in the vehicle. Integrating advanced machine learning algorithms including deep learning into IoT devices is an active research area to make smart objects closer to reality. Moreover, it is possible to get the most value out of IoT deployments through analyzing IoT data, extracting hidden information, and predicting control decisions. A wide variety of machine learning techniques have been used in IoT domain ranging from traditional methods such as regression, support vector machine, and random forest to advanced ones such as convolutional neural networks, LSTM, and variational autoencoder.

In the future, the Internet of Things may be a non-deterministic and open network in which auto-organized or intelligent entities (web services, SOA components) and virtual objects (avatars) will be interoperable and able to act independently (pursuing their own objectives or shared ones) depending on the context, circumstances or environments. Autonomous behavior through the collection and reasoning of context information as well as the object's ability to detect changes in the environment (faults affecting sensors) and introduce suitable mitigation measures constitutes a major research trend, clearly needed to provide credibility to the IoT technology. Modern IoT products and solutions in the marketplace use a variety of different technologies to support such context-aware automation, but more sophisticated forms of intelligence are requested to permit sensor units and intelligent cyber-physical systems to be deployed in real environments.

b). Architecture

IoT system architecture, in its simplistic view, consists of three tiers: Tier 1: Devices, Tier 2: the Edge Gateway, and Tier 3: the Cloud. Devices include networked things, such as the sensors and actuators found in IioT equipment, particularly those that use protocols such as Modbus, Bluetooth, Zigbee, or proprietary protocols, to connect to an Edge Gateway. The Edge Gateway consists of sensor data aggregation systems called Edge Gateways that provide functionality, such as pre-processing of the data, securing connectivity to cloud, using systems such as WebSockets, the event hub, and, even in some cases, edge analytics or fog computing. Edge Gateway layer is also required to give a common view of the devices to the upper layers to facilitate in easier management. The final tier includes the cloud application built for IioT using the microservices architecture, which are usually polyglot and inherently secure in nature using HTTPS/Oauth. It includes various database systems that store sensor data, such as time series databases or asset stores using backend data storage systems (e.g. Cassandra, PostgreSQL). The cloud tier in most cloud-based IoT system features event queuing and messaging system that handles communication that transpires in all tiers.[1] Some experts classified the three-tiers in the IioT system as edge, platform, and enterprise and these are connected by proximity network, access network, and service network, respectively.

Building on the Internet of things, the web of things is an architecture for the application layer of the Internet of things looking at the convergence of data from IoT devices into Web applications to create innovative use-cases. In order to program and control the flow of information in the Internet of things, a predicted architectural direction is being called BPM Everywhere which is a blending of traditional process management with process mining and special capabilities to automate the control of large numbers of coordinated devices.

Network architecture

The Internet of things requires huge scalability in the network space to handle the surge of devices. IETF 6LoWPAN would be used to connect devices to IP networks. With billions of devices being added to the Internet space, Ipv6 will play a major role in handling the network layer scalability. IETF's Constrained Application Protocol, ZeroMQ, and MQTT would provide lightweight data transport.

Fog computing is a viable alternative to prevent such large burst of data flow through Internet. The edge devices' computation power to analyse and process data is extremely limited. Limited processing power is a key attribute of IoT devices as their purpose is to supply data about physical objects while remaining autonomous. Heavy processing requirements use more battery power harming IoT's ability to operate. Scalability is easy because IoT devices simply supply data through the internet to a server with sufficient processing power.

c). Complexity

In semi-open or closed loops (i.e. value chains, whenever a global finality can be settled) the IoT will often be considered and studied as a complex system due to the huge number of different links, interactions between autonomous actors, and its capacity to integrate new actors. At the overall stage (full open loop) it will likely be seen as a chaotic environment (since systems always have finality). As a practical approach, not all elements in the Internet of things run in a global, public space. Subsystems are often implemented to mitigate the risks of privacy, control and reliability. For example, domestic robotics (domotics) running inside a smart home might only share data within and be available via a local network. Managing and controlling a high dynamic ad hoc IoT things/devices network is a tough task with the traditional networks architecture, Software Defined Networking (SDN) provides the agile dynamic solution that can cope with the special requirements of the diversity of innovative IoT applications.

d). Size considerations

The Internet of things would encode 50 to 100 trillion objects, and be able to follow the movement of those objects. Human beings in surveyed urban environments are each surrounded by 1000 to 5000 trackable objects. In 2015 there were already 83 million smart devices in people's homes. This number is expected to grow to 193 million devices by 2020.

The figure of online capable devices grew 31% from 2016 to 2017 to reach 8.4 billion.

e). Space considerations

In the Internet of Things, the precise geographic location of a thing—and also the precise geographic dimensions of a thing—will be critical. Therefore, facts about a thing, such as its location in time and space, have been less critical to track because the person processing the information can decide whether or not that information was important to the action being taken, and if so, add the missing information (or decide to not take the action). (Note that some things in the Internet of Things will be sensors, and sensor location is usually important.) The GeoWeb and Digital Earth are promising applications that become possible when things can become organized and connected by location. However, the challenges that remain include the constraints of variable spatial scales, the need to handle massive amounts of data, and an indexing for fast search and neighbour operations. In the Internet of Things, if things are able to take actions on their own initiative, this human-centric mediation role is eliminated. Thus, the time-space context that we as humans take for granted must be given a central role in this information ecosystem. Just as standards play a key role in the Internet and the Web, geospatial standards will play a key role in the Internet of things.

1.1.2 ENABLING TECHNOLOGIES FOR IOT:

There are many technologies that enable the IoT. Crucial to the field is the network used to communicate between devices of an IoT installation, a role that several wireless or wired technologies may 14ulfil:

a). Addressability

The original idea of the Auto-ID Center is based on RFID-tags and distinct identification through the Electronic Product Code. This has evolved into objects having an IP address or URI. An alternative view, from the world of the Semantic Web focuses instead on making all things (not just those electronic, smart, or RFID-enabled) addressable by the existing naming protocols, such as URI. The objects themselves do not converse, but they may now be referred to by other agents, such as powerful centralized servers acting for their human owners. Integration with the Internet implies that devices will use an IP address as a distinct identifier. Due to the limited address space of Ipv4 (which allows for 4.3 billion different addresses), objects in the IoT will have to use the next generation of the Internet protocol (Ipv6) to scale to the extremely large address space required. Internet-of-things devices additionally will benefit from the stateless address auto-configuration present in Ipv6, as it reduces the configuration overhead on the hosts, and the IETF 6LoWPAN header compression. To a large extent, the future of the Internet of things will not be possible without the support of Ipv6; and consequently, the global adoption of Ipv6 in the coming years will be critical for the successful development of the IoT in the future.

b). Application Layer

- ADRC defines an application layer protocol and supporting framework for implementing IoT applications.

c). Short-range wireless

- Bluetooth mesh networking – Specification providing a mesh networking variant to Bluetooth low energy (BLE) with increased number of nodes and standardized application layer (Models).

- Light-Fidelity (Li-Fi) – Wireless communication technology similar to the Wi-Fi standard, but using visible light communication for increased bandwidth.
- Near-field communication (NFC) – Communication protocols enabling two electronic devices to communicate within a 4 cm range.
- Radio-frequency identification (RFID) – Technology using electromagnetic fields to read data stored in tags embedded in other items.
- Wi-Fi – Technology for local area networking based on the IEEE 802.11 standard, where devices may communicate through a shared access point or directly between individual devices.
- ZigBee – Communication protocols for personal area networking based on the IEEE 802.15.4 standard, providing low power consumption, low data rate, low cost, and high throughput.
- Z-Wave – Wireless communications protocol used primarily for home automation and security applications

d). Medium-range wireless

- LTE-Advanced – High-speed communication specification for mobile networks. Provides enhancements to the LTE standard with extended coverage, higher throughput, and lower latency.
- 5G – 5G wireless networks can be used to achieve the high communication requirements of the IoT and connect a large number of IoT devices, even when they are on the move.

e). Long-range wireless

- Low-power wide-area networking (LPWAN) – Wireless networks designed to allow long-range communication at a low data rate, reducing power and cost for transmission. Available LPWAN technologies and protocols: LoRaWan, Sigfox, NB-IoT, Weightless, RPMA.
- Very small aperture terminal (VSAT) – Satellite communication technology using small dish antennas for narrowband and broadband data.

f). Wired

- Ethernet – General purpose networking standard using twisted pair and fiber optic links in conjunction with hubs or switches.
- Power-line communication (PLC) – Communication technology using electrical wiring to carry power and data. Specifications such as HomePlug or G.hn utilize PLC for networking IoT devices.

1.1.3 SECURITY OF IOT:

While security considerations are not new in the context of information technology, the attributes of many IoT implementations present new and unique security challenges. Addressing these challenges and ensuring security in IoT products and services must be a fundamental priority.Users need to trust that IoT devices and related data services are secure from vulnerabilities, especially as this technology become more pervasive and integrated into our daily lives. Poorly secured IoT devices and services can serve as potential entry points for cyber attack and expose user data to theft by leaving data streams inadequately protected.

The interconnected nature of IoT devices means that every poorly secured device that is connected online potentially affects the security and resilience of the Internet globally. This challenge is amplified by other considerations like the mass-scale deployment of homogenous IoT devices, the ability of some devices to automatically connect to other devices, and the likelihood of fielding these devices in unsecure environments.

As a matter of principle, developers and users of IoT devices and systems have a collective obligation to ensure they do not expose users and the Internet itself to potential harm. Accordingly, a collaborative approach to security will be needed to develop effective and appropriate solutions to IoT security challenges that are well suited to the scale and complexity of the issues.

1.1.4 PRIVACY OF IOT:

The full potential of the Internet of Things depends on strategies that respect individual privacy choices across a broad spectrum of expectations. The data streams and user specificity afforded by IoT devices can unlock incredible and unique value to IoT users, but concerns about privacy and potential harms might hold back full adoption of the Internet of Things. This means that privacy rights and respect for user privacy expectations are integral to ensuring user trust and confidence in the Internet, connected devices, and related services.

Indeed, the Internet of Things is redefining the debate about privacy issues, as many implementations can dramatically change the ways personal data is collected, analyzed, used, and protected. For example, IoT amplifies concerns about the potential for increased surveillance and tracking, difficulty in being able to opt out of certain data collection, and the strength of aggregating IoT data streams to paint detailed digital portraits of users. While these are important challenges, they are not insurmountable. In order to realize the opportunities, strategies will need to be developed to respect individual privacy choices across a broad spectrum of expectations, while still fostering innovation in new technology and services.

1.2 ARDUINO:

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its products are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (‘shields’) or breadboards (For prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers can be programmed using C and C++ programming languages. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors.

The name *Arduino* comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

1.2.1 HARDWARE OD ARDUINO:

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name *Arduino* to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in *-duino*.

An early Arduino board with an RS-232 serial interface (upper left) and an Atmel Atmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left.

Figure 1: Early Arduino Board

Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8,] ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed *shields*. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino Uno is the Optiboot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

An official Arduino Uno R2 with descriptions of the I/O locations

Figure 2: Official Arduino UNO R2

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The *Diecimila*, *Duemilanove*, and current *Uno* provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent, but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

1.2.2 SOFTWARE OF ARDUINO:

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

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

A). Pro IDE

On October 18th, 2019, Arduino Pro IDE (alpha preview) was released. The system still uses Arduino CLI (Command Line Interface), but improvements include a more professional development environment, autocompletion support, and Git integration. The application frontend is based on the Eclipse Theia Open Source IDE. The main features available in the alpha release are:

- Modern, fully featured development environment
- Dual Mode, Classic Mode (identical to the Classic Arduino IDE) and Pro Mode (File System view)
- New Board Manager
- New Library Manager

- Board List
- Basic Auto-Completion (Arm targets only)
- Git Integration
- Serial Monitor
- Dark Mode

b). Sketch

A *sketch* is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension **.ino**. Arduino Software (IDE) pre-1.0 saved sketches with the extension **.pde**.

A minimal Arduino C/C++ program consists of only two functions:

- `setup()`: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function `main()`.
- `loop()`: After `setup()` function exits (ends), the `loop()` function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. It is analogous to the function `while(1)`.

Blink example

Power LED (red) and User LED (green) attached to pin 13 on an Arduino compatible board

Figure 3: Power and User LED attached to pin 13

Most Arduino boards contain a light-emitting diode (LED) and a current limiting resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions. A typical program used by beginners, akin to Hello, World!, is “blink”, which repeatedly blinks the on-board LED integrated into the Arduino board. This program uses the functions `pinMode()`, `digitalWrite()`, and `delay()`, which are provided by the internal libraries included in the IDE environment. This program is usually loaded into a new Arduino board by the manufacturer.

```
#define LED_PIN 13                  // Pin number attached to LED.

Void setup() {
   pinMode(LED_PIN, OUTPUT);       // Configure pin 13 to be a digital output.
}

void loop() {
   digitalWrite(LED_PIN, HIGH);    // Turn on the LED.
   Delay(1000);                    // Wait 1 second (1000 milliseconds).
   DigitalWrite(LED_PIN, LOW);     // Turn off the LED.
   Delay(1000);                    // Wait 1 second.
}
```

c). Libraries

The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

1.2.3 APPLICATIONS:

- Arduboy, a handheld game console based on Arduino
- Arduinome, a MIDI controller device that mimics the Monome
- Ardupilot, drone software and hardware
- ArduSat, a cubesat based on Arduino.
- C-STEM Studio, a platform for hands-on integrated learning of computing, science, technology, engineering, and mathematics (C-STEM) with robotics.
- Data loggers for scientific research.
- OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars
- OpenEVSE an open-source electric vehicle charger
- XOD, a visual programming language for Arduino

1.3 OVERVIEW OF PROJECT:

a). This project provides a combination of process of sensing several gas levels in the air and also the ambient temperature and humidity, thus sensing the quality of the air.

b).The levels of the gases and the temperature is displayed in a LCD display panel , which continuously shows the real time output values of the gas sensors , temperature and humidity sensor.

1.4 OBJECTIVE:

a). To measure and display temperature and humidity level of the environment.

b). To combine advanced detection technologies to produce an air quality sensing system with advanced capabilities to provide low cost comprehensive monitoring.

c). To display the sensed data in user friendly format in LCD display panel.

1.5 MOTIVATION AND CONTRIBUTION :

This project aims at benefitting the modern society aided with the constantly increasing use of internet. IoT (Internet of Things) IoT is simply the network of interconnected things/devices which are embedded with sensors, software, network connectivity and necessary electronics that enables them to collect and exchange data making them responsive. Internet of Things is essentially an architectural framework more than a concept in which data exchange and integration allowed between the computer systems and physical world over consisting network infrastructure which exists in real world. With the help of sensors and actuators when IoT is augmented, the technology becomes an instance of the more normal class of cyber- physical systems, which also circumscribe technologies such as an intelligent transportation, smart grids, smart homes, and smart cities. Through its embedded computing system each thing is uniquely classifiable but within the existing internet infrastructure it is able to interoperate. With the help of various existing technologies the devices collect useful data and then autonomously flow the data between other devices. Current project and such other projects like smart homes such as control and automation of heating, ventilation, air conditioning etc. uses Wi-Fi for remote monitoring. This project would also work in taking care of home and atmosphere as we could monitor the parameters from anywhere under the Wi-Fi access. This allows authorities to monitor air pollution in different areas and take action to control the issue. Main motive of this project is to prevent the harmful effects of pollutants present in air so that healthy surroundings can be maintained by data analysis of stored data in IoT cloud.

1.6 THESIS OUTLINE:

In Chapter 2 we have done literature survey regarding Internet of Things and the existing models. In Chapter 3, we discuss the existing system model. Chapter 4 we analyze the air quality parameter, MQ135 Gas sensor , Arduino Uno and Wi-Fi module ESP8266. The simulation results and discussion of the thesis is given in Chapter 5 followed by conclusion of this thesis.

Chapter 2

LITERATURE REVIEW

This chapter represents the background and the literature survey of the Internet of things , the air quality parameter, MQ135 Gas sensor , Arduino Uno and Wi-Fi module ESP8266. All the information are collected using these device so that we can measure air quality.

2.1 INTORDUCTION:

The main objective of IOT Air Monitoring System is that the Air pollution is a growing issue these days. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Due to flexibility and low cost Internet of things (IoT) is getting popular day by day. With the urbanization and with the increase in the vehicles on road the atmospheric conditions have considerably affected. Harmful effects of pollution include mild allergic reactions such as irritation of the throat eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma. Monitoring gives measurements of air pollutant and sound pollution concentrations, which can then be analysed interpreted and presented. This information can then be applicable in many ways. Analysis of monitoring data allows us to assess how bad air pollution is from day to day.

2.2 WHY THIS PROJECT:

We propose an air quality as well as sound pollution monitoring system that allows us to monitor and check live air quality as well as sound pollution in an area through IOT.System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data. Also, system keeps measuring sound level and reports it. The sensors interact with 22rduino uno which processes this data and transmits it over the application. This allows authorities to monitor air pollution in different areas and act against it. Also, authorities can keep a watch on the noise pollution near schools, hospitals and no honking areas.

2.3 EXISTING MODEL:

The commercial meters available in the market are Fluke CO-220 carbon monoxide meter for CO, Amprobe CO2 meter for CO2, ForbixSemicon LPG gas leakage sensor alarm for LPG leakage detection. The researchers in this field have proposed various air quality monitoring systems based on WSN, GSM and GIS. Now each technology has limited uses
according to the intended function, as Zigbee is meant for users with Zigbee trans-receiver, Bluetooth. GIS based system is designed, implemented and tested to monitor the pinpoints of air pollution of any area. It consists of a microcontroller, gas sensors, mobile unit, a temporary memory buffer and a web server with internet connectivity which collects data from different locations along with coordinate's information at certain time of a day. The
readings for particular location are averaged in a closed time and space. The Global Positioning System (GPS) module is attached to a system to provide accurate representation of pollution sources in an area. The recorded data is periodically transferred to a computer through a General Packet Radio Service (GPRS) connection and then the data will be displayed on the dedicated website with user acceptance. As a result large number of people can be benefited with the large.

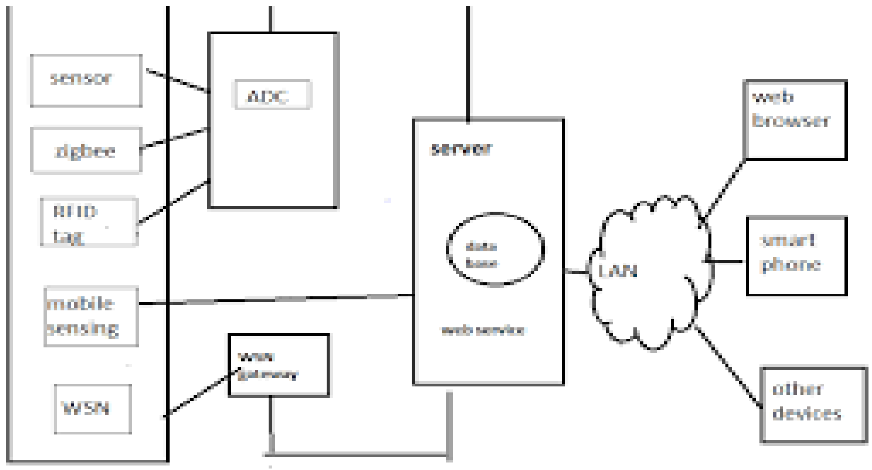


Figure 4:Block Diagram of an existing air quality monitor

2.4 AIR QUALITY PARAMETER:

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

a). Carbon Dioxide (CO2) – CO2 is colorless, odorless gas and non-combustible gas. Moreover, it is considered under the category of asphyxiate gases that have capability of interfering the availability of oxygen for tissues. Carbon Dioxide is a gas essential to life in the planet, because it is one of the most important elements evolving photosynthesis process, which converts solar into chemical energy. The concentration of CO2 has increased due mainly to massive fossil fuels burning. This increase makes plants grow rapidly. The rapid growth of undesirable plants leads to the increase use of chemicals to eliminate them.

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

c). Nitrogen Dioxide (NO2) – Nitrogen Dioxide is a brownish gas, easily detectable for its odour, very corrosive and highly oxidant. It is produced as the result of fossil fuels burning. Usually NO thrown to the atmosphere is converted in NO2 by chemical processes. In high concentrations, NO2 may lead to respiratory problems. Like SO2, it contributes to acid rains.

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

e). LPG-Liquefied petroleum gas (LPG) is an odorless and colorless liquid which evaporates readily into a gas. Leakage is normally detected by adding an odorant into it. It is considered under the category of highly flammable gases and it can be classified as a carcinogen and mutagen if Butadiene content is more than 0.1%. LPG may leak in the form of a gas or a liquid. If it leaks in the form of a liquid it evaporates quickly and will eventually form large cloud of gas in air which is relatively heavier than air thus drops to the ground. Whereas, LPG vapors travel along the ground for a long distance and gets collected in drains or basements. Gas leads to burn or explode after getting in touch with a source of ignition.

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

2.5 ADVANTAGE OF PURE AIR:

a). Easy breathing: Breathing is, for the most part, an unconscious action. You do it without thinking about it much, but if your home’s air quality is low, breathing can feel like trying to suck air through a coffee stirrer. Even if you don’t realize it, your body does. Surviving on shallow breaths puts a lot of stress on your body, especially your heart and lungs, making even climbing a flight of stairs a physically tiring task. With clean air, you can take big, easy breaths and give your body the oxygen it needs.

b). Better sleep: Our respiration changes when we’re asleep such that our breathing rate decreases and becomes much steadier. But during REM sleep, breathing rate increases and fluctuates as much as it would when you’re awake.So as important as air is when you’re awake, it’s just as important when you’re asleep. Try to sleep with something covering your face or mouth. You definitely won’t be comfortable and will spend most of the night tossing and turning. Respiratory irritation and airborne allergens affect the quality of your sleep, leading to sleep-disordered breathing problems such as sleep apnea. With clean indoor air, you won’t have to worry about breathing problems, so you can wake up feeling refreshed.

c). Elimination of allergens: Reportedly, somewhere between 10 and 30 percent of the global population suffers from allergic rhinitis—what we know as hay fever—which is characterized by sneezing, congestion, an itchy throat, and irritated eyes. Hay fever is caused by allergens, which are airborne particles that you may be allergic to. The most common of which include:

- Dust
- Animal dander
- Mold spores
- Cockroach debris
- Pollen

These allergens float in the air but eventually fall onto flat surfaces, leaving a thin, seemingly furry film on surfaces. The heavier the allergen, the faster it settles out of the air and onto a flat surface. Eliminating allergens ensures that you’re not sneezing up a storm or otherwise suffering in your own home. Aside from frequent vacuuming, maintaining your home’s ventilation and air conditioning system keeps allergens at bay, filtering out the particles from outdoor air to give you just the good stuff. Having the right type of filtration and having it properly installed, will make a big difference on how clean your air is.

d). Reduced odours: Whether you cook a lot of fish, forget to clean your cat’s litter box, or have housemates who eat exclusively bean burritos, odours are always a potential problem. The smell of onion breath or B.O. isn't anyone's cup of tea, but bad smells also have a very real effect on your mood, thoughts, behaviours, and dreams. Bad odours can leave you uncharacteristically grumpy and irritable. On the other hand, good smells—like chamomile, flowers, and citrus—improve your mood, reduce stress, and lower your heart rate.

Furthermore, odours are about as distracting as someone slurping their food. If you're trying to study or get work done at home, you have to try that much harder to focus on your work because of the pungent, musty mystery smell creeping through your home. You shouldn't have to walk through your home with a clothespin on your noise. Traditional filters, even HEPA filters, will not remove odour and sprays may just mask them for a while. The key is to break down the organic material (VOC's) in the air that causes the smell in the first place. UV filters with a special catalyst like this PureAir do just that.

e). Balanced humidity: Good indoor air quality also means a good balance of humidity. Humidity, which is the amount of water in the air, affects all aspects of your immediate environment. Too much humidity makes your home feel stuffy and moist, which makes breathing feel heavy. The extra moisture causes mold growth and increased dust mite populations while also attracting pests.
At the same time, not enough humidity makes for cold and dry conditions that lead to irritated skin, static shock, frequent nosebleeds, and completely uncooperative hair.

Air conditioning, ventilation, and humidifiers or dehumidifiers balance humidity for better health and comfortable living. Some thermostats can display the current percentage of humidity in your home. Ideally, your indoor humidity should be around 45 percent.

f). Reduced energy costs: The steps required to maintain your air quality—improving airflow and ventilation, maintaining humidity and carbon dioxide levels—all go toward optimizing your HVAC system and ensuring that it doesn't work too hard to keep your home healthy. Efficiency inevitably leads to reduced costs on your utility bill.You shouldn't be getting sick from the air in your own home, so take the necessary steps to ensure good indoor air quality.

2.6 CLASSIFICATION AND ANALYSIS OF DIFFERENT TECHNIQUES:

Table 1: Classification and Analysis of Existing Air Pollution Monitoring Techniques

S.No.	Researcher	Contribution	Advantages
1.	Xiaoke Yang, Lingyu Yang, Jing Zhang (School of Automation Science and Electrical Engineering, Beihang University,	A WiFi-enabled indoor air quality monitoring and control system:- Published in: Control & Automation (ICCA), 2017 13th IEEE	This paper proposes an open platform of a WiFi-enabled indoor air quality monitoring and control system, which could be incorporated into such a 'smart building' structure. The complete software and hardware design of this system is presented, along with a series of

	Beijing, 100191, China**)**	International Conference	control experiments. The proposed system operates over an existing WiFi wireless network utilizing the MQTT protocol. It is capable of monitoring the indoor air quality as well as controlling an air purifier to regulate the particulate matters concentration. Experiment results under a real world office environment demonstrate the effectiveness of the proposed design.
2.	Sujuan Liu, Chuyu Xia, Zhenzhen Zhao (College of Electronic Information and Control Engineering, Beijing University of Technology, 100124, China)	A low-power real-time air quality monitoring system using LPWAN based on LoRa:- Published in: Solid-State and Integrated Circuit Technology (ICSICT), 2016 13th IEEE International Conference	This paper presents a low-power real-time air quality monitoring system based on the LoRa Wireless Communication technology. The proposed system can be laid out in a large number in the monitoring area to form sensor network. The system integrates a single-chip microcontroller, several air pollution sensors (NO2, SO2, O3, CO, PM1, PM10, PM2.5), Long Range (LoRa) -Modem, a solar PV-battery part and graphical user interface (GUI). As communication module LoRa sends the data to the central monitoring unit and then the data would be saved in the could. The range tests at an outdoor area show that LoRa is able to reach to approximately 2Km. The TX power is only about 110mA which is lower compared with other used wireless technology. An easy to use GUI was designed in the system. Based on LoRa technology, GUI, and Solar PV-battery part the system has several progressive features such as low cost, long distance, high coverage, long device battery life, easy to operate.
3.	M.F.M Firdhous, B.H Sudantha, P.M	IoT enabled proactive indoor air	This paper proposes an IoT based indoor air quality

	Karunaratne (Dept. of Information Technology, University of Moratuwa, Sri Lanka)	quality monitoring system for sustainable health management:- Published in:Computing and Communications Technologies (ICCCT), 2017 2nd International Conference	monitoring system for tracking the ozone concentrations near a photocopy machine. The experimental system with a semiconductor sensor capable of monitoring ozone concentrations was installed near a high volume photocopier. The IoT device has been programmed to collect and transmit data at an interval of five minutes over blue tooth connection to a gateway node that in turn communicates with the processing node via the WiFi local area network. The sensor was calibrated using the standard calibration methods. As an additional capability, the proposed air pollution monitoring system can generate warnings when the pollution level exceeds beyond a predetermined threshold value.
4.	R du Plessis, A Kumar, GP Hancke (Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa)	A wireless system for indoor air quality monitoring:- Published in: Industrial Electronics Society , IECON 2016 - 42nd Annual Conference of the IEEE	This paper describes the development of a wireless monitoring system which can be deployed in a building. The system measures carbon dioxide, carbon monoxide and temperature. The system developed in this paper can serve as the monitoring component of a HVAC control system and function as an indoor air quality monitor independently.
5.	Giovanni B. Fioccola ; Raffaele Sommese ; Imma Tufano ; Roberto Canonico ; Giorgio Ventre	Polluino: An efficient cloud-based management of IoT devices for air quality monitoring:- Published in: Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum	The Internet of Things paradigm originates from the proliferation of intelligent devices that can sense, compute and communicate data streams in a ubiquitous information and communication network. The great amounts of data coming from these devices introduce some challenges related to the storage and processing capabilities of the information. This strengthens the novel paradigm known as Big Data. In such a complex scenario, the

			Cloud computing is an efficient solution for the managing of sensor data. This paper presents Polluino, a system for monitoring the air pollution via Arduino. Moreover, a Cloud-based platform that manages data coming from air quality sensors is developed.
6.	Sneha Jangid, Sandeep Sharma (School of ICT, Gautam Buddha University, Greater Noida, India)	An embedded system model for air quality monitoring:- Published in: Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference	Objective of the paper is to present a system model which can facilitate the assessment of health impacts caused due to indoor air pollutant as well as outdoor and can intimate the human prior about the risk he/she going to have, here we are focusing our work in context to allergic patients as they will be informed by this tool such that they can secure themselves without actually experiencing the risk factors, here a sensing network based microcontroller equipped with gas sensors, optical dust particle sensor, humidity and temperature sensor has been used for air quality monitoring. The design included various units mainly: sensing unit, processing unit, power unit, display unit, communication unit. This work will apply the techniques of electrical engineering with the knowledge of environmental engineering by using sensor networks to measure Air Quality Parameters.
7.	H. Ali, J. K. Soe, Steven. R. Wel (School of Electrical Engineering & Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia)	A real-time ambient air quality monitoring wireless sensor network for schools in smart cities:- Published in: Smart Cities Conference (ISC2), 2015 IEEE First International	In this paper, a low-cost solar-powered air quality monitoring system based on ZigBee wireless network system technology is presented. The solar powered network sensor nodes can be deployed by schools to collect and report real-time data on carbon monoxide (CO), nitrogen dioxide (NO2), dust particles, temperature, and relative humidity. The proposed system

			allows schools to monitor air quality conditions on a desktop/laptop computer through an application designed using LabVIEW and provides an alert if the air quality characteristics exceed acceptable levels. They tested the sensor network successfully at the Singapore campus of the University of Newcastle, Australia. The experimental results obtained by them demonstrated that the sensor network can provide high-quality air quality measurements over a wide range of CO, NO2 and dust concentrations.
8.	Yonggao Yang, Lin Li (Department of Computer Science, Prairie View A&M University, Prairie View, TX 77446, U.S.A)	A smart sensor system for air quality monitoring and massive data collection:- Published in: Information and Communication Technology Convergence (ICTC), 2015 International Conference	Air pollution has been a global challenge for environment protection. Effectively collecting and scientifically visualizing the air quality data can better help us monitor the environment and address related issues. This article presents a smart sensor system for air quality monitoring which consists of three units: the smart sensor unit, the smartphone, and a server.

Chapter 3

PROPOSED MODEL AND COMPONENT DESCRIPTION

3.1 PROPOSED MODEL:

The paper aims at designing an air pollution monitoring system which can be installed in a specific locality and to enhance the system from the previously developed systems beating the earlier disadvantages by developing an android app available for the public. This app can be used by anyone to get in live updates about the pollution in their region. It uses Arduino integrated with individual gas sensors like carbon monoxide, ammonia along with particulate matter, humidity, and smoke which measures the concentration of each gas separately. The collected data is uploaded to the cloud using thingspeak platform at regular time intervals. Ethernet shield is used for connecting arduino and cloud.Pictorial or graphical representation of values can be shown in Thingspeak.

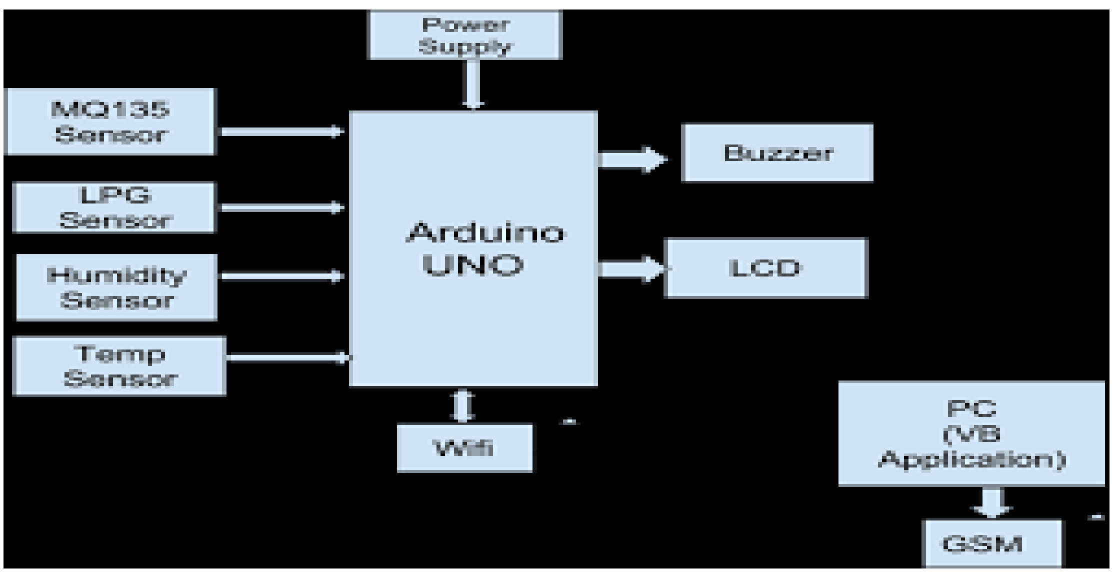


Figure 5:Block Diagram of a Proposed Model

The users can install an android application through which they get the recent updates and graphical content up to date. The average concentration of each gas is analyzed using matlab. Then certain time control is assigned based on the standard level of each gas measured and the result can be viewed in android application. The architecture of air pollution monitoring and awareness creation system is shown in figure 1. The concentration level of each gas can be viewed both as a graph and in numerical format. Based on these values the air quality index value is calculated and the nature of the air quality in that area is determined which is also displayed through the app. Along with this, the health effects for the corresponding air quality is displayed to create awareness among the public. Additionally, they could also get to know the temperature and weather in that region. The users will not get disturbed with irrelevant data as the values displayed are location specific and help them stay tuned to the current status of air pollution. The hardware setup of proposed

3.2 BLOCK DIAGRAM:

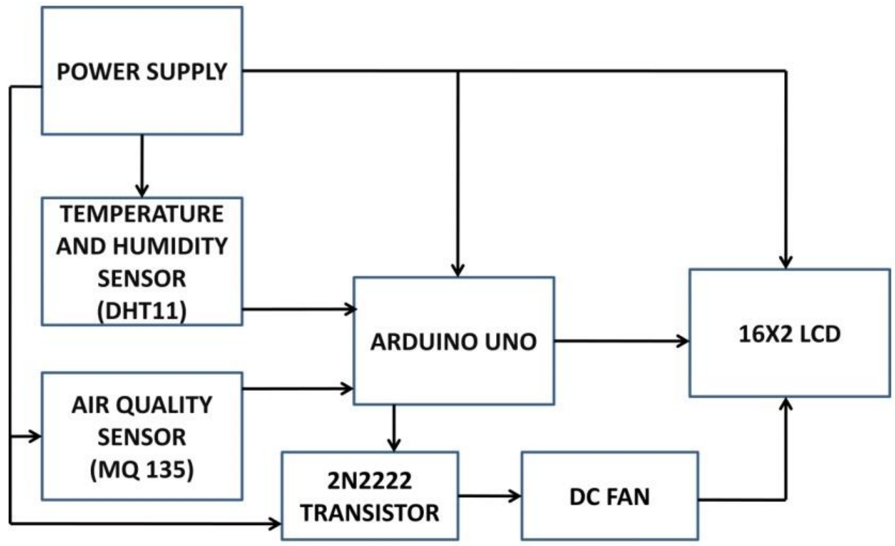


Figure 6: Block Diagram of Air Quality Monitoring Sensing

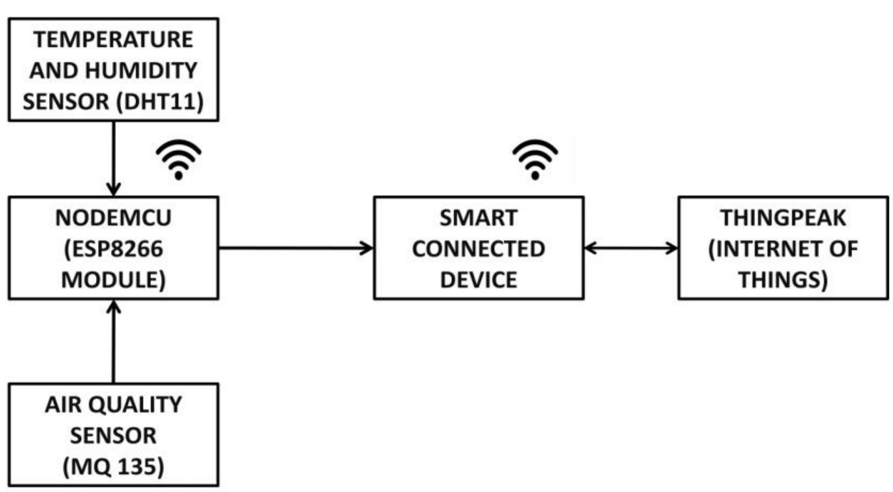


Figure 7: Block Diagram of sending the data to THINGSPEAK using NodeMCU

3.3 HARDWARE REQUIREMENTS:

3.3.1 For Different Parameter Sensing:-

- Temperature and Humidity sensor (DHT11)
- Air Quality sensor (MQ 135)
- 2n2222 Transisitor
- DC Fan
- Potentiometer
- 16x2 LCD Panel
- NodeMCU
- Arduino Uno

3.3.2 For Power Supply :-

- Step down transformer (12-0-12 V,1 A)
- Diodes
- Voltage Regulator (7805)
- Capacitors (0.01 micro Farad, 470 micro Farad)
- Wires

3.4 SOFTWARE REQUIREMENTS:

- Arduino (Version 1.8.2)
- THINGSPEAK website

3.5 COMPONENT DESCRIPTION:

a).Temperature and humidity sensor (DHT11):-

Product Description:

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.

Figure 8: DHT11

Pin Description:

- 1, the VDD power supply 3.5～5.5V DC
- DATA serial data, a single bus
- 3, NC, empty pin
- 4, GND, used to connect the module to system ground

b).Air Quality Sensor (MQ135):-

Product Description:

Air quality click is suitable for detecting ammonia (NH_3), nitrogen oxides (NO_x) benzene, smoke, CO_2 and other harmful or poisonous gases that impact air quality. The MQ-135 sensor unit has a sensor layer made of tin dioxide (SnO_2), an inorganic compound which has lower conductivity in clean air than when polluting gases are present. To calibrate Air quality, use the on-board potentiometer to adjust the load resistance on the sensor circuit.

Figure 9: MQ135 Air Quality Sensor

Pin Description:

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

c). 2N2222 Transistor:-

Product Description:

The 2N2222 is a common NPN bipolar junction transistor (BJT) used for general purpose low-power amplifying or switching applications. It is designed for low to medium current, low power, medium voltage, and can operate at moderately high speeds. It was originally made in the TO-18 metal can as shown in the picture. The 2N2222 is considered a very common transistor and is used as an exemplar of an NPN transistor. It is frequently used as a small-signal transistor and it remains a small general purpose transistor of enduring popularity.

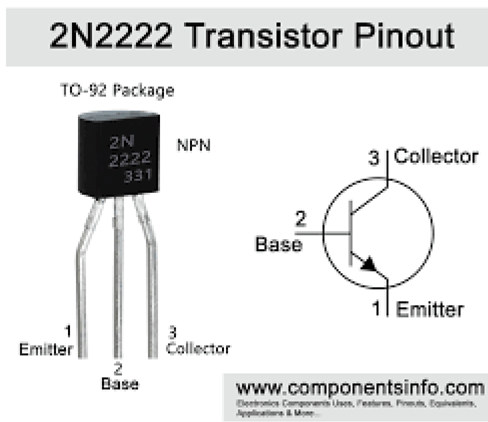


Figure 10: 2N2222 Transistor

d). DC Fan:-

Product Description:

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.


Figure 11: DC fan with DC Motor

e).Potentiometer:-

Product Description:

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat. The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Figure 12: Potentiometer

f).16X2 LCD Panel:-

Product Description:

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome.[1] LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and seven-segment displays.

Figure 13: 16X2 LCD Panel

Pin Description:

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

Table 2: Pin Description of 16X2 LCD Panel

g).Arduino Uno:-

Product Description:

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (*shields*) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

Figure 14: Arduino UNO

Pin Description:

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

Table 3: Pin Description of Arduino UNO

Technical Specification:

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

Table 4: Technical Specification of Arduino UNO

h). NodeMCU:

Product Description:

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266.

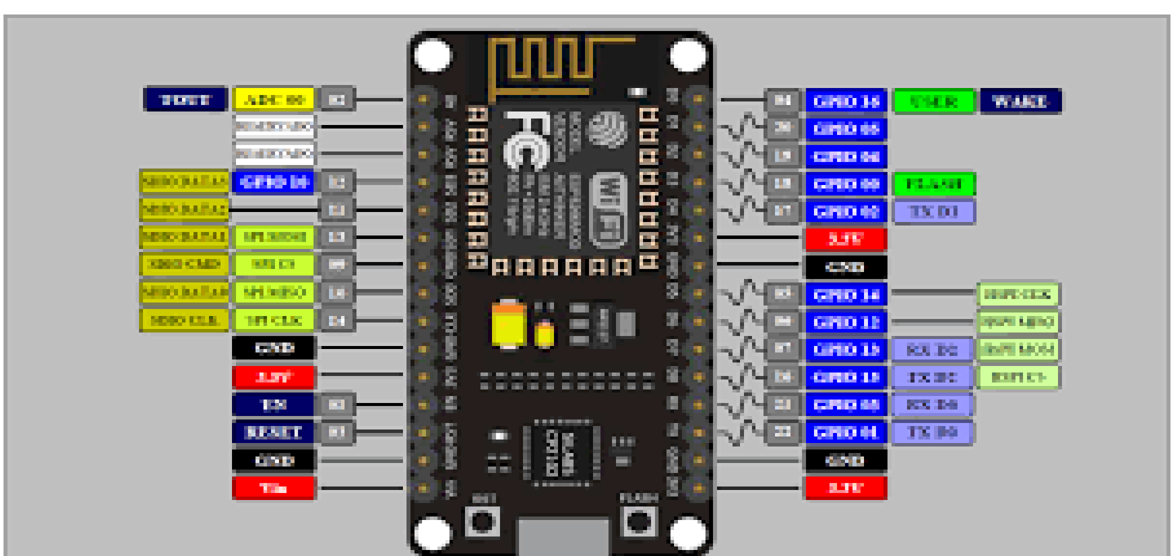


Figure 15: NodeMCU

3.6 COST ESTIMATION:

Sl. No.	Name of Components	Cost (Rs.)
1.	Arduino Uno	400
2.	DTH11	150
3.	Bread board	60
4.	Jumper Wires	3/piece
5.	12-0-12 Transformer	130
6.	7805	20
7.	Capacitor	5/piece
8.	Resistances	3/piece
9.	16x2 LCD	140
10.	MQ 135	150
11.	NodeMCU	400
12.	DC Fan	20
13.	Potentiometer	10/piece
14.	2N222 Transistor	5/piece

Table 5: Cost of Different Components

3.7 WORK PLAN:

Sl No.	Duration	Work done/will be done
1.	August'17-September'17	Selection of Project topic and study of the project topic.
2.	October '17	Study of previous work done related to our topic and preparing a literature review.
3.	November'17	Purchase of required components and making of a 5V power supply, and sensing of temperature and humidity using DTH11 sensor.
4.	December'17-May'18	Sensing of air quality using MQ135 and displaying the values on an LCD panel and sending the data to ThingSpeak Platform using NodeMCU. And controlling the speed of Fan based on temperature sensed by DHT11.

Table 6: Work Plan

Chapter 4

CIRCUIT DIAGRAM AND CODE INPLEMENTATION

4.1 WORK DONE:

4.1.1 Connections:

a).DTH11’s voltage, ground is connected to +5V and 0V and signal can be connected to any Digital Pin 8 of Arduino Uno.

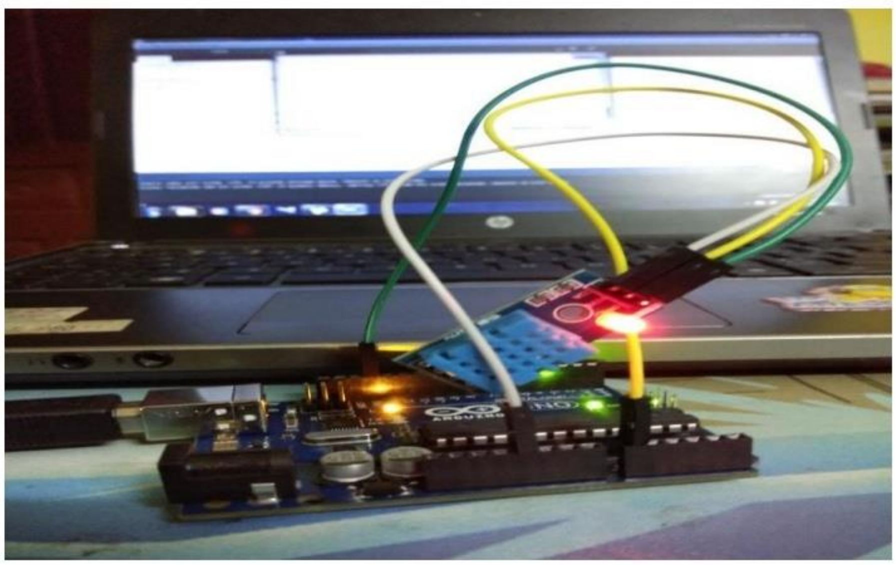


Figure 16: Connection of DTH11 with Arduino

b). MQ135’s voltage and ground are connected to +5V and 0V and analog output pin is connected to analog Pin A0 of Arduino Uno.

c). LCD RS pin to digital pin 12, Enable pin to digital pin 11,D4 pin to digital pin 5, D5 pin to digital pin 4, D6 pin to digital pin 3, D7 pin to digital pin 2, R/W pin to ground, VSS pin to ground, VCC pin to 5V, 10K resistor ends to +5V and ground and wiper to LCD VO pin.

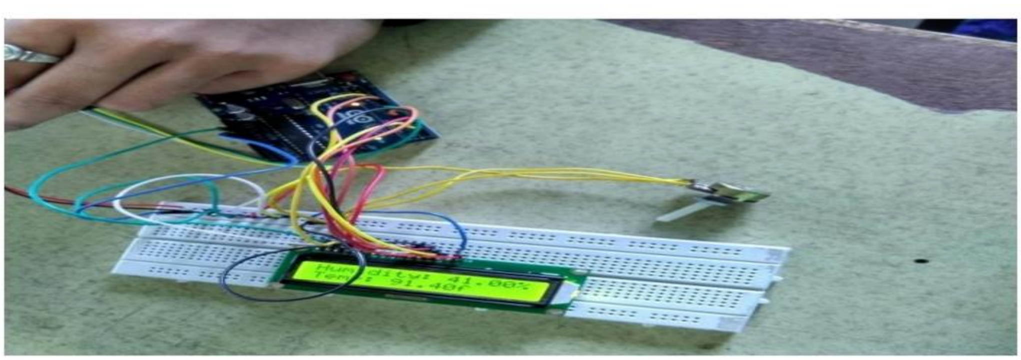


Figure 17: Connection of LCD with Arduino

d). The data pins of DHT11 are connected to Digital pin of NodeMCU and that of MQ135 is connected to Analog Pin.

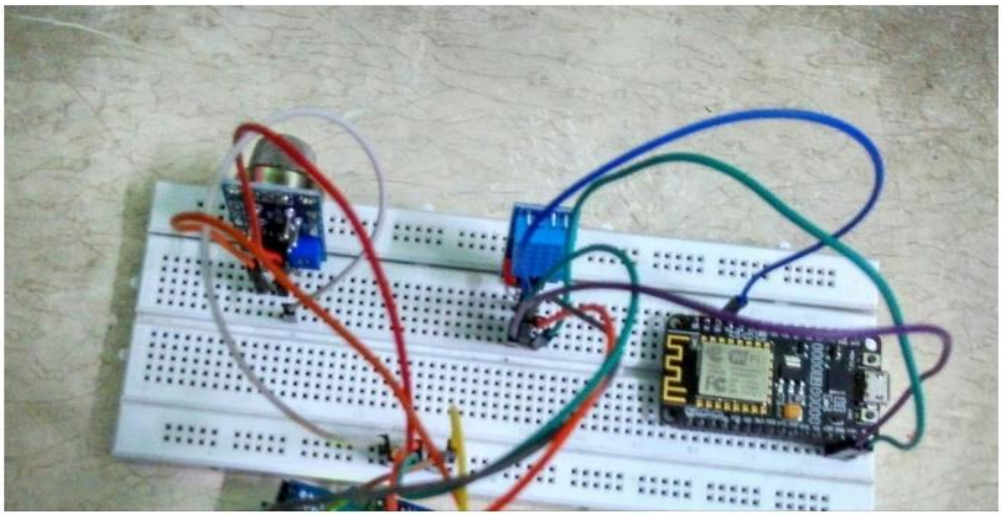


Figure 18: Connection of DHT11 and MQ135 with NodeMCU

e). The base of 2N2222 transistor is connected to a pwm pin of Arduino Uno and the emitter and collecter of transisitor is connected to the DC Fan and supplied with 9V battery (in Forward Bias). The other connections are kept the same.

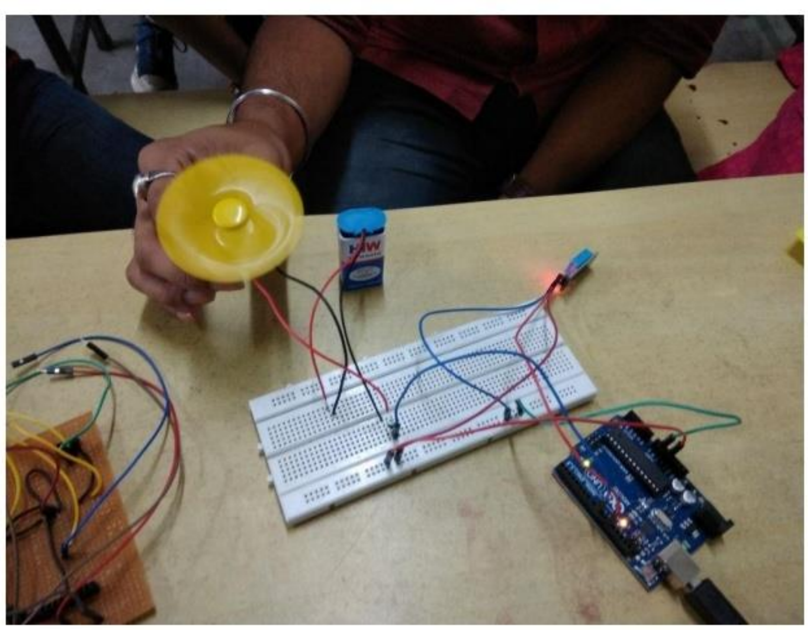
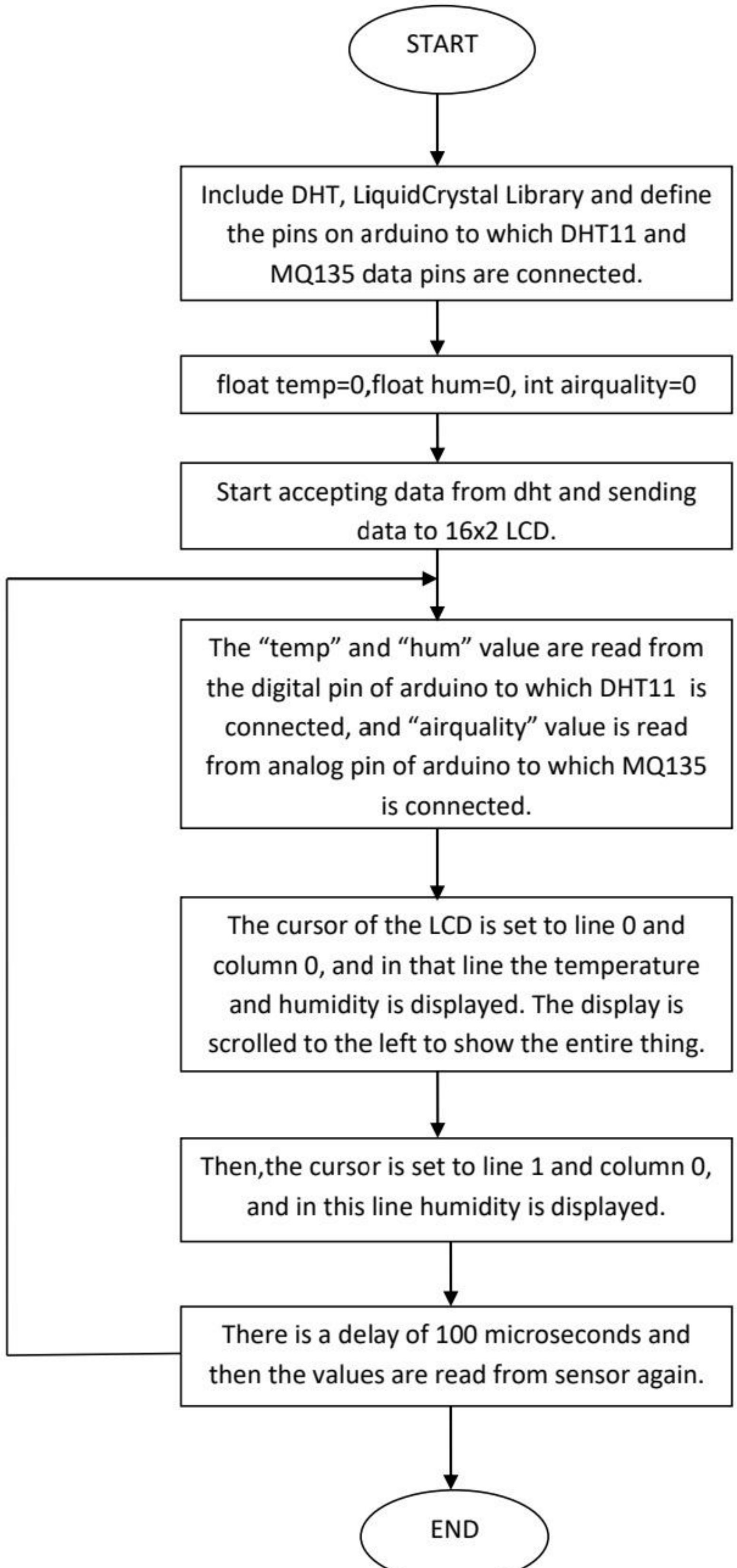


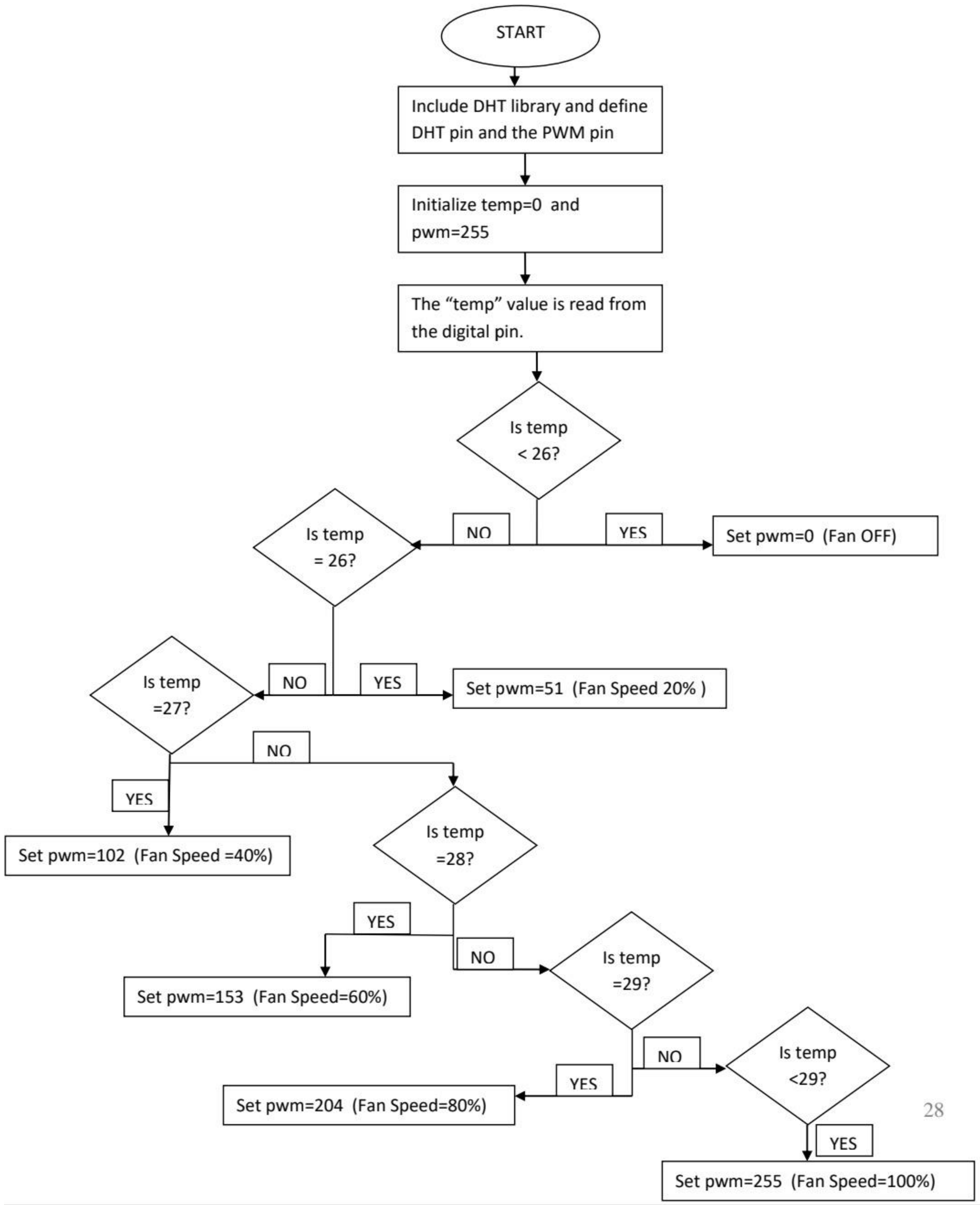
Figure 19: Connection of DHT11 with DC fan

4.1.2 Flowchart for displaying the Temperatutre , Humidity and Air Quality on LCD:

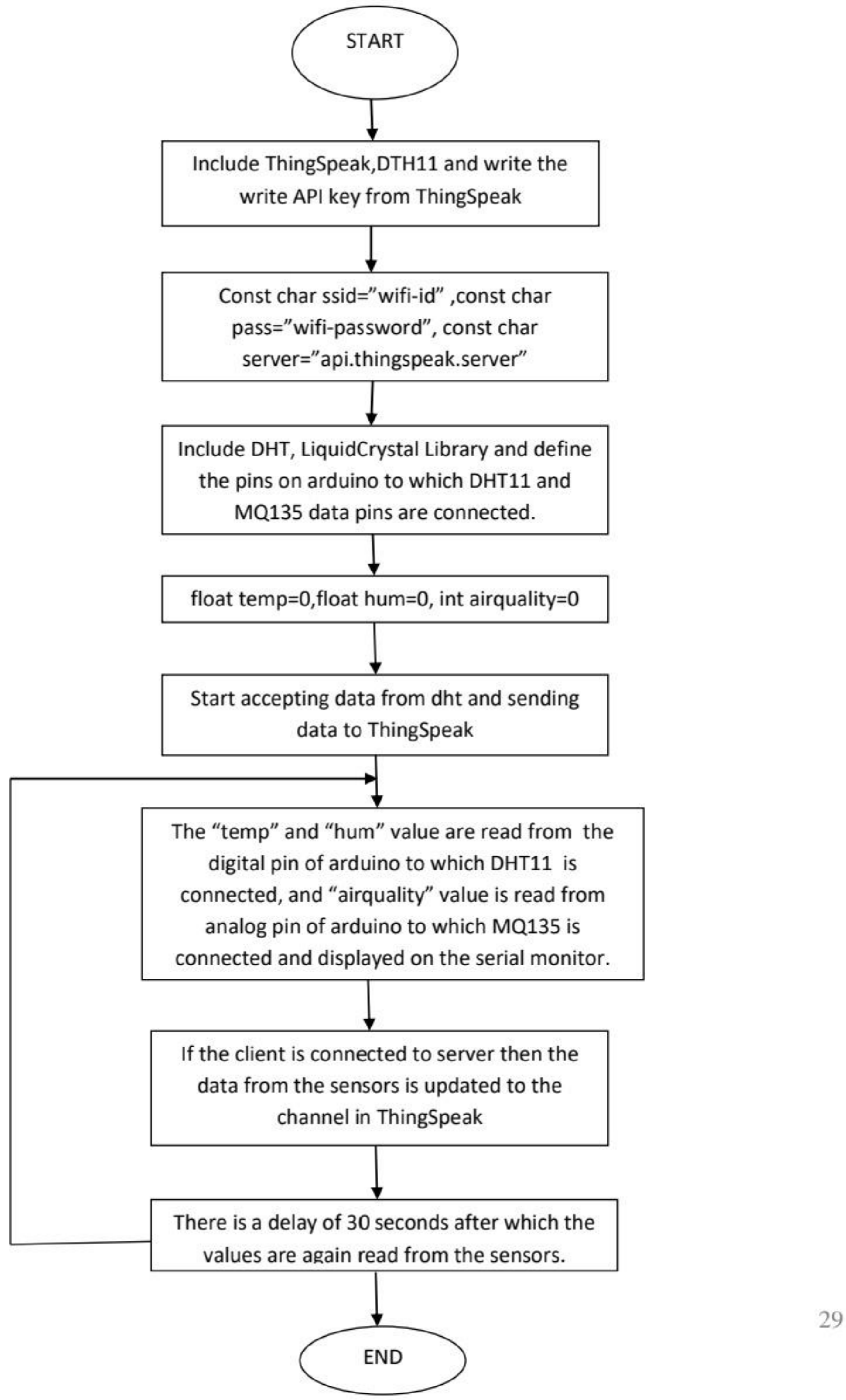


27

4.1.3 Flowchart for Controlling the speed of Fan Based on Temperature:



4.1.4 Flowchart for Sending the Temperature , Humidity and Air Quality on THINGSPEAK:



4.2 WORKING EXPLAINATION:

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

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

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

4.3 CODE EXPLAINATION:

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

Using this library you can directly get the PPM values, by just using the below two lines:

```
MQ135 gasSensor = MQ135(A0);

float air_quality = gasSensor.getPPM();
```

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

```
#include "MQ135.h"

void setup (){

Serial.begin (9600);

}
```

```
void loop() {

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

float rzero = gasSensor.getRZero();

Serial.println (rzero);

delay(1000);

}
```

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

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

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

```
#include <SoftwareSerial.h>

#define DEBUG true

SoftwareSerial esp8266(9,10);

#include <LiquidCrystal.h>

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

const int sensorPin= 0;

int air_quality;
```

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

```
pinMode(8, OUTPUT);

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

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

lcd.print ("circuitdigest ");

lcd.setCursor (0,1);

lcd.print ("Sensor Warming ");

delay(1000);
```

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

```
Serial.begin(115200);

esp8266.begin(115200);

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

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

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

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

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

pinMode(sensorPin, INPUT);

lcd.clear();
```

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

```
if(esp8266.available())

  {

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

    {

     delay(1000);
```

```
    int connectionId = esp8266.read()-48;

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

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

      webpage+= " Air Quality is ";

      webpage+= air_quality;

      webpage+=" PPM";

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

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

```
    sendData(cipSend,1000,DEBUG);

    sendData(webpage,1000,DEBUG);



    cipSend = "AT+CIPSEND=";

    cipSend += connectionId;

    cipSend += ",";

    cipSend +=webpage.length();

    cipSend +="\r\n";
```

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

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

lcd.print ("Air Quality is ");

lcd.print (air_quality);

lcd.print (" PPM ");

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

```
if (air_quality<=1000)

{

lcd.print("Fresh Air");

digitalWrite(8, LOW);
```

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

```
String sendData(String command, const int timeout, boolean debug)

{

  String response = "";

  esp8266.print(command); // send the read character to the esp8266

  long int time = millis();

  while( (time+timeout) > millis())

  {

    while(esp8266.available())

    {

      // The esp has data so display its output to the serial window

      char c = esp8266.read(); // read the next character.

      response+=c;

    }

  }

  if(debug)

  {

    Serial.print(response);

  }
```

```
    return response;

}
```

4.4 ADVANTAGES:

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

4.5APPLICATIONS:

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

Chapter 5

TESTING AND RESULTS

5.1 TESTING AND OUTPUT OF THE PROJECT:

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

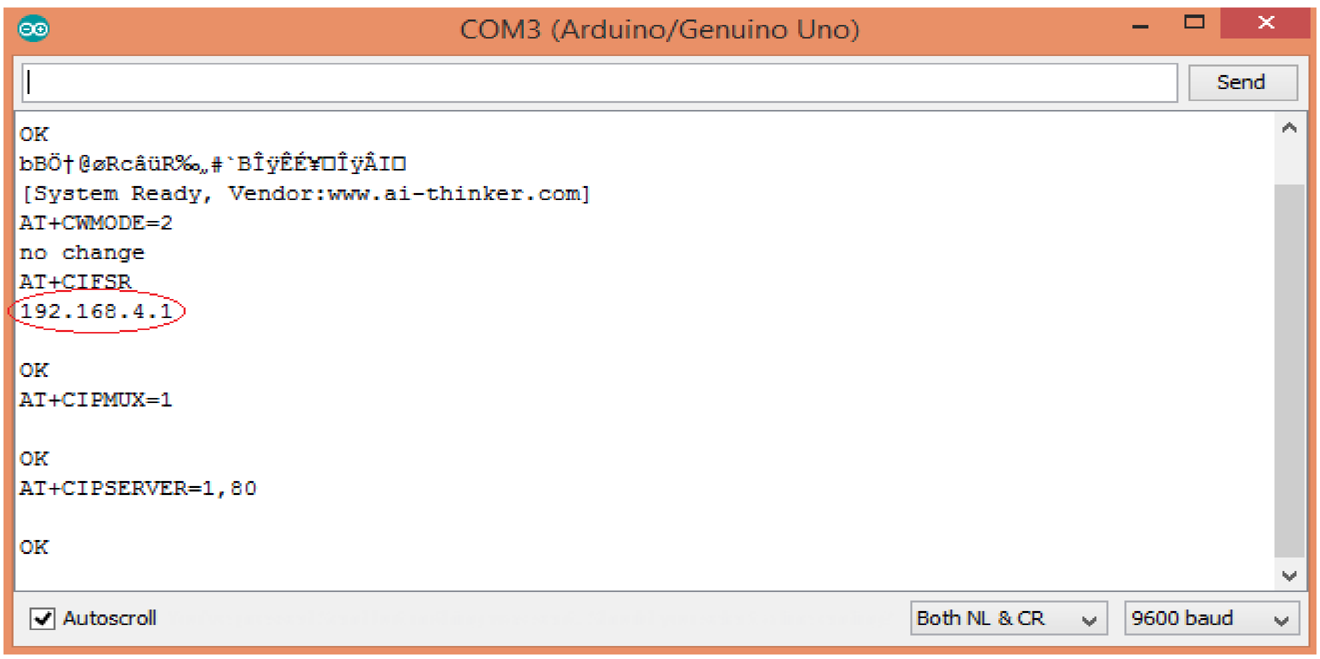


Figure 20: Output testing window 1

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

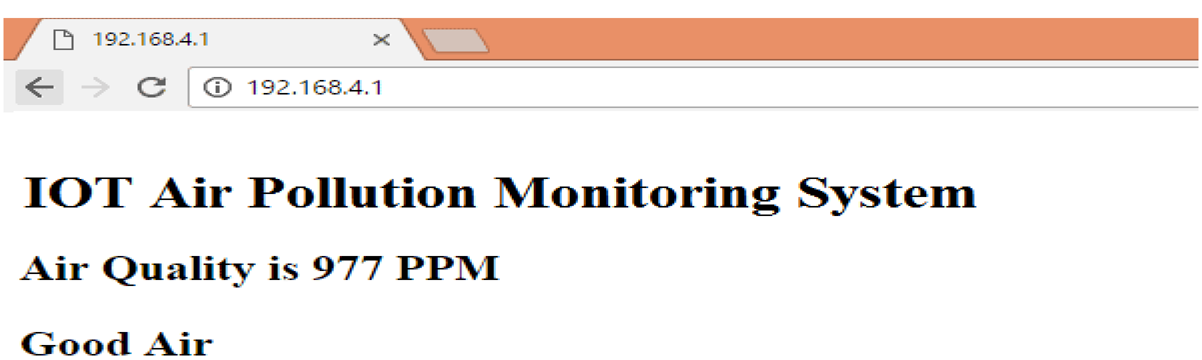


Figure 21: Output Testing Window 2

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

5.1.1 CODE:

```
#include "MQ135.h"
#include <SoftwareSerial.h>
#define DEBUG true
SoftwareSerial esp8266(9,10); // This makes pin 9 of Arduino as RX pin and pin 10 of Arduino as the TX
pin
const int sensorPin= 0;
int air_quality;
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11, 5, 4, 3, 2);

void setup() {
pinMode(8, OUTPUT);
lcd.begin(16,2);
lcd.setCursor (0,0);
lcd.print ("circuitdigest ");
lcd.setCursor (0,1);
lcd.print ("Sensor Warming ");
delay(1000);
Serial.begin(115200);
esp8266.begin(115200); // your esp's baud rate might be different
  sendData("AT+RST\r\n",2000,DEBUG); // reset module
  sendData("AT+CWMODE=2\r\n",1000,DEBUG); // configure as access point
  sendData("AT+CIFSR\r\n",1000,DEBUG); // get ip address
  sendData("AT+CIPMUair_quality=1\r\n",1000,DEBUG); // configure for multiple connections
  sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // turn on server on port 80
pinMode(sensorPin, INPUT);        //Gas sensor will be an input to the arduino
lcd.clear();
}

void loop() {

MQ135 gasSensor = MQ135(A0);
float air_quality = gasSensor.getPPM();

if(esp8266.available()) // check if the esp is sending a message
  {
```

```
   if(esp8266.find("+IPD,"))
   {
    delay(1000);
    int connectionId = esp8266.read()-48; /* We are subtracting 48 from the output because the read()
function returns the ASCII decimal value and the first decimal number which is 0 starts at 48*/
    String webpage = "<h1>IOT Air Pollution Monitoring System</h1>";
      webpage += "<p><h2>";
      webpage+= " Air Quality is ";
      webpage+= air_quality;
      webpage+=" PPM";
      webpage += "<p>";
    if (air_quality<=1000)
{
  webpage+= "Fresh Air";
}
else if(air_quality<=2000 && air_quality>=1000)
{
  webpage+= "Poor Air";
}

else if (air_quality>=2000 )
{
webpage+= "Danger! Move to Fresh Air";
}

webpage += "</h2></p></body>";
    String cipSend = "AT+CIPSEND=";
    cipSend += connectionId;
    cipSend += ",";
    cipSend +=webpage.length();
    cipSend +="\r\n";

    sendData(cipSend,1000,DEBUG);
    sendData(webpage,1000,DEBUG);

    cipSend = "AT+CIPSEND=";
    cipSend += connectionId;
    cipSend += ",";
    cipSend +=webpage.length();
    cipSend +="\r\n";

    String closeCommand = "AT+CIPCLOSE=";
    closeCommand+=connectionId; // append connection id
    closeCommand+="\r\n";

    sendData(closeCommand,3000,DEBUG);
   }
  }

lcd.setCursor (0, 0);
lcd.print ("Air Quality is ");
```

```
lcd.print (air_quality);
lcd.print (" PPM ");
lcd.setCursor (0,1);
if (air_quality<=1000)
{
lcd.print("Fresh Air");
digitalWrite(8, LOW);
}
else if( air_quality>=1000 && air_quality<=2000 )
{
lcd.print("Poor Air, Open Windows");
digitalWrite(8, HIGH );
}
else if (air_quality>=2000 )
{
lcd.print("Danger! Move to Fresh Air");
digitalWrite(8, HIGH);   // turn the LED on
}
lcd.scrollDisplayLeft();
delay(1000);
}
String sendData(String command, const int timeout, boolean debug)
{
    String response = "";
    esp8266.print(command); // send the read character to the esp8266
    long int time = millis();
    while( (time+timeout) > millis())
    {
      while(esp8266.available())
      {
        // The esp has data so display its output to the serial window
        char c = esp8266.read(); // read the next character.
        response+=c;
      }
    }
    if(debug)
    {
      Serial.print(response);
    }
    return response;
}
```

5.2 OUTPUT:

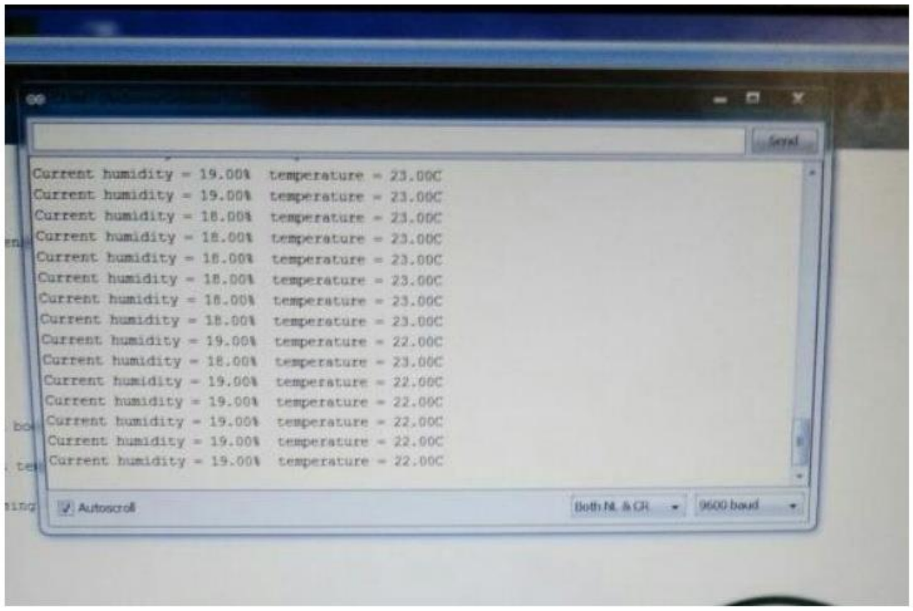


Figure 22: Output of DHT11 sensor(on serial monitor)

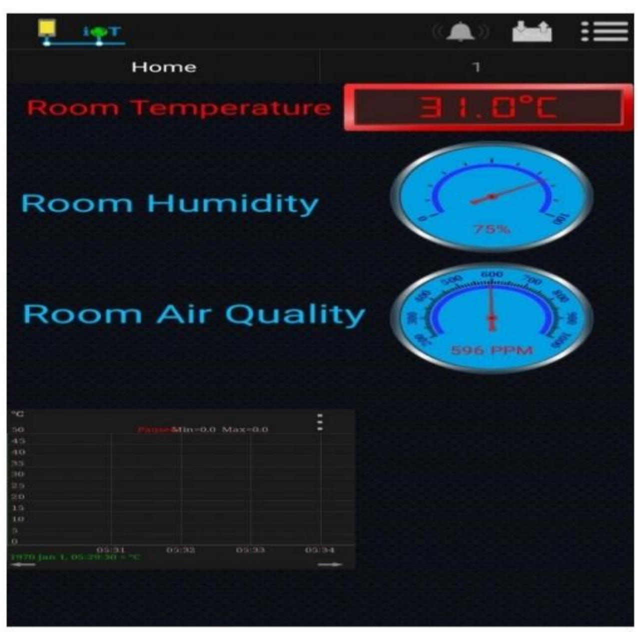


Figure 23: Output of DHT11 and MQ135 in THINGSPEAK app

5.3 RESULT AND DISCUSSION:

Last step after wiring all parts to PCB board is covering all parts with the box. All sensors must be contacted to the air directly. The box needs to drill until the sensors and LCD fit with the box. And then, the all parts seemgood looking with the box.The dimension of box is18 cm x 11 cm x 6 cm.The Figure shows the front and rear view of the box. It can be seen, the box is not too small nor too big as well,so the box is easy to carry anywhere.

Next, we will illustrate the result of the measurement from sensors. The data is taken from the measurement of a cigarette:

The results show that if the red line is upward, the particle level of pollution is higher and dangerous. At level of 0-50 ppm, healthy air falls into the healthy category, while in the range 50 – 100 ppm is categorized medium air (no effect on health). The range of 100-200 ppm, air is not healthy, 200-300 ppm is very unhealthy, while 300 upwards is said to be very dangerous [8]. As shown in figure 10, the smoke point indicates that the smoke level is above 10k, indicating that the ambient air condition is very dangerous.

Carbon monoxide (CO) when inhaled into the lungs will participate in blood circulation and will block the entry of oxygen needed by the body. This can happen because CO gas is toxic metabolism, participating to react metabolically with blood. Like oxygen, CO gas readily reacts with blood (hemoglobin) [9]. Figure 11 shows that no CO content contained in cigarettes, indicating that there is no CO content in the air.

Air pollution problems generally are caused by solid particles of TSP (Total Suspended Particulate or total drift particle) with a maximum diameter of about 45 mm, particles of PM10 (particulate matter) with diameters less than 10 mm and PM2.5 with diameters less than 2.5 mm. These particles are believed by environmental and public health experts to trigger respiratory infections, because solid particles of PM10 and PM2,5 can precipitate in the respiratory tract of the bronchi and alveoli areas, whereas TSP cannot be inhaled into the lungs, but only at upper respiratory tract [10]. Figure 15 shows that the ppm content is almost 500, implying that the air is dangerous and Figure 16 illustrates that the air condition is dangerous because the ppm content is almost 1000.

5.4 CONCLUSION:

The system to monitor the air of environment using Arduino microcontroller, IOT Technology is proposed to improve quality of air. With the use of IOT technology enhances the process of monitoring various aspects of environment such as air quality monitoring issue proposed in this paper. Here the using of MQ135 gas sensor gives the sense of different type of dangerous gas and arduino is the heart of this project which controls the entire process. Wi-Fi module connects the whole process to internet and LCD is used for the visual Output. The Automatic Air & Sound management system is a step forward to contribute a solution to the biggest threat. The air & sound monitoring system overcomes the problem of the highly-polluted areas which is a major issue. It supports the new technology and effectively supports the healthy life concept. This system has features for the people to monitor the amount of pollution on their mobile phones using the application.

REFRENCES

[1] https://www.tinkercad.com/

[2] https://circuits.io/

[3] https://www.arduino.cc/

[4] https://circuitdigest.com/ microcontroller - projects /iot -air-pollution-monitoring-using-arduino

[5] Riteeka Nayak, Malaya Ranjan Panigrahy , Vivek Kumar Rai and T Appa Rao:IOT based air pollution monitoring system Vol-3, Issue-4, 2017

[6] Navreetinder Kaur , Rita Mahajan and Deepak Bagai: Air Quality Monitoring System based on Arduino Microcontroller Vol. 5, Issue 6, June 2016

[7] Palaghat Yaswanth Sai: An IoT Based Automated Noise and Air Pollution Monitoring System Vol. 6, Issue 3,March 2017

[8] 1 L.Ezhilarasi, 2 K.Sripriya, 3 A .Suganya , 4 K.Vinodhini.: A System for Monitoring Air and Sound Pollution using Arduino Controller with IOT TechnologyVol. 3 Issue 2 (2017) Pages 1781 – 1785

[9] Exploring Arduino : Tools and Techniques forEngineering Wizardry by Jeremy Blum 1st edition

[10] Ms. Sarika Deshmukh, Mr.Saurabh surendran and Prof.M.P. Sardey:Air and Sound Pollution Monitoring System using IoT Volume: 5 Issue: 6

[11].https://securedstatic.greenpeace.org/india/Global/india/Airpoclypse--Not-just-Delhi--Air-in most-Indian-cities-hazardous--Greenpeace report.pdf [12].content/uploads/2208/04/5v-regulator-using-7805.JPG

[13]. https://store.arduino.cc/usa/arduino-uno-rev3

[14].https://www.aliexpress.com/item/32620995019.html?spm=2114.4001

[15]. https://en.wikipedia.org/wiki/Arduino#/media/File:UnoConnections.jpg

[16]. http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/

[17]. https://playground.arduino.cc/Main/DHT11Lib

[18]. https://www.robot-r-us.com/vmchk/sensor-temp/humid/dht11-temperature-and-humiditysensor. Html

[19]. https://electrosome.com/power-supply-design-5v-7805-voltage-regulator/

[20]. https://www.epa.gov/air-research/air-monitoring-measuring-and-emissions-research

[21]. http://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humiditytutorial-
using-arduino/

[22]. https://www.elprocus.com/ever-wondered-lcd-works/

[23]. https://www.google.co.in/search?q=dht11&client=firefox-a&rls=org.mozilla:en
US:official&channel=np&dcr=0&source=lnms&tbm=isch&sa=X&ved=0ahUKEwj8sMu0o-PXAhUlTY8KHW8wB6kQ_AUICigB&biw=1366&bih=598

[24]. https://en.wikipedia.org/wiki/Environmental_monitoring

THANK YOU