AIR POLLUTION MONITORING BASED ON IoT

A Project Report Submitted to
Jawaharlal Nehru Technological University Kakinada, Kakinada.
In partial fulfillment of the requirement for the Award of the degree of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS & COMMUNICATION ENGINEERING

Submitted by

K. MAHESH	**(19NA1A0427)**
V. VAMSI KRISHNA	**(19NA1A0422)**
SK. NISHITHA	**(19NA1A0417)**
V. AJAY	**(19NA1A0433)**

Under the supervision of
P. SOWMYA KEERTHANA, M.Tech
Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

LINGAYA'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY
(APPROVED BY AICTE- NEW DELHI, AFFILIATED TO JNTUK, ACCREDITED BY NAAC 'A' GRADE)
MADALVARIGUDEM (V), VIJAYAWADA-521 212.(Krishna District) [A.P.]

2022-2023

LINGAYA'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY
(APPROVED BY AICTE- NEW DELHI, AFFILIATED TO JNTUK, ACCREDITED BY NAAC 'A' GRADE)
MADALVARIGUDEM (V), VIJAYAWADA--521212.(Krishna District) [A.P.]

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled with "AIR POLLUTION MONITORING BASED ON IoT" is a bonafide record of the work done

By

K. MAHESH	**(19NA1A0427)**
V. VAMSI KRISHNA	**(19NA1A0422)**
SK. NISHITHA	**(19NA1A0417)**
V. AJAY	**(19NA1A0433)**

In the partial fulfillment for the award of degree of **Bachelor of Technology** in **ELECTRONICS AND COMMUNICATION ENGINEERING** from **Jawaharlal Nehru Technological University Kakinada,** during the Academic Year 2022- 2023.

Supervisor
P. SOWMYA KEERTHANA

Head of the Department
B. Lakshmi

EXTERNAL EXAMINER

LINGAYA'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY
(APPROVED BY AICTE- NEW DELHI, AFFILIATED TO JNTUK, ACCREDITED BY NAAC 'A' GRADE)
MADALVARIGUDEM (V), VIJAYAWADA- -521212.(Krishna District) [A.P.]

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DECLARATION

we declare that this thesis titled "AIR POLLUTION MONITORING BASED ON IoT " is the result of our own work and the materials which are not the results of our own work have been clearly acknowledged.

Signature of the Student

K. MAHESH	**(19NA1A0427)**
V. VAMSI KRISHNA	**(19NA1A0422)**
SK. NISHITHA	**(19NA1A0417)**
V. AJAY	**(19NA1A0433)**

ACKNOWLEDGEMENT

We Wish to take this opportunity to express my deep Gratitude to all the people who have extended their Cooperation in various ways during my work. It is my pleasure to acknowledge the help of all those individuals.

We take the opportunity to offer sincere Gratitude to our Honorable Chairman GADDE RAJLING, for providing us infrastructure and valuable support.

We convey my Profound Gratitude to the Principal Dr. N. SATYANARAYANA, without Whose blessings; it would not have been possible for me to carry out this work.

We convey my Profound Gratitude to the Vice Principal Dr. M. CHINNARAO, without Whose blessings; it would not have been possible for me to carry out this work.

We express my thanks to Dr. B. SATYANARAYANA, R&C, for his enthusiastic support and insight during this period.

We convey my gratitude and special thanks to Prof. B. LAKSHMI**,** Head of Department of ECE for her encouragement in bringing shape to this Report. She has been a constant source of Motivation and Inspiration.

We extend my sincere thanks with Gratitude to my Guide P.SOWMYA KEERTHANA, Assistant Professor, Electronics and Communication Engineering, LIMAT, VJA, for her valuable encouragement, guidance and suggestions during this period. The lively discussions we had with her helped me a lot to improve my focus.

Thanking You

Yours sincerely

K. MAHESH	**(19NA1A0427)**
V. VAMSI KRISHNA	**(19NA1A0422)**
SK. NISHIT1HA	**(19NA1A0417)**
V. AJAY	

AIR POLLUTION MONITORING BASED ON IoT

ABSTRACT

The problem of air pollution is a major global concern that affects human health, the environment IoT based air pollution monitoring system real-time data on air quality, which can help authorities to develop strategies to reduce pollution levels and protect public health. With increasing urbanization and industrialization, air pollution has become a significant global challenge, affecting the health and well -being of millions of people worldwide. IoT based air pollution monitoring systems are emerging as a powerful tool to address this challenge by providing accurate and reliable data on air quality. These systems can detect a wide range of pollutants, including particulate matter of harmful gases. They use sensors that can collect data in real-time and transmit it to cloud-based platforms for analysis and visualization.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

Air pollution is a serious problem which affects the health and well-being of people around the world. According to the World Health Organization (WHO), air pollution is responsible for 7 million premature deaths each year. Traditional air quality monitoring systems have been in place for several decades, but they have limitations, such as low spatial resolution and slow data reporting. IoT-based air pollution monitoring systems are a promising solution to address these limitations and provide real-time, accurate, and comprehensive air quality data.

Air quality monitoring systems based on IoT employ a network of sensors to measure various contaminants in the air. These sensors, which can be stationary or mobile, are meant to measure various pollutants such as particulate matter , carbon monoxide , nitrogen oxides , Sulphur dioxide , ozone , and volatile organic compounds. The sensors continuously collect data and wirelessly transfer it to a central server. The information is then processed and analyzed before being made available to the public via a web portal or mobile application.

One of the main advantages of IoT -based air quality monitoring systems is that they provide real-time data on air quality, which is essential for identifying pollution hotspots and implementing targeted pollution control measures. The data can also be used to develop predictive models to forecast air quality in the future, which can help city planners and policymakers make informed decisions on urban development, transportation, and other policies that affect air quality.

Despite these benefits, installing IoT-based air quality monitoring systems poses several problems. One problem is sensor calibration, which is required for reliable data. The sensors must be calibrated on a regular basis to guarantee that they accurately measure pollutants. Another problem is data transfer, as the sensors must be capable of wirelessly transmitting data to a central server. This necessitates the use of a strong wireless network capable of handling massive volumes of data. Finally, data security is an issue since the data collected bythe sensors may contain sensitive information that must be safeguarded. Air quality monitoring systems based on IoT are a viable answer to the problem of air pollution. They give real-time, accurate, and complete data on air quality that can be utilized to build targeted pollution - reduction initiatives. While installing these systems is difficult, developments in technology and rising awareness of the necessity of air quality monitoring are making IoT-based systems more widespread in cities around the world.

Although IoT-based air pollution monitoring systems have significant advantages over traditional air quality monitoring systems, they also confront a number of issues that must be solved. Here are some of the main issues with IoT-based air pollution monitoring systems:

SENSOR ACCURACY : The quality and dependability of sensor data are important for any air pollution monitoring system's performance. The sensors in IoT- based air pollution monitoring systems must be calibrated on a regular basis to ensure that they provide reliable data. Sensor drift, which occurs when the performance of the sensors changes over time, can also impair data accuracy.

DATA TRANSMISSION: The sensors used in IoT-based air pollution monitoring systems must be capable of wirelessly transmitting data to a central server. This necessitates the use ofa strong wireless network capable of handling massive volumes of data. The dependability of the wireless network might also have an impact on data accuracy.

DATA PROCESSIN AND STORAGE: The vast volume of data generated by Internet of Things-based air pollution monitoring devices might be difficult to handle and store. The data must be processed fast in order to offer real-time information on air quality, and it must be securely kept in order to safeguard the users' privacy.

DATA INTEGRATION AND ANALYSIS: The data collected by IoT-based air pollution monitoring systems needs to be integrated and analyzed to provide meaningful insights into air quality. This requires advanced data analytics tools and techniques to identify patterns and trends in the data.

COST: IoT-based air pollution monitoring systems can be expensive to implement and maintain, particularly in low-income countries or regions. The cost of sensors, wireless networks, and data processing and storage infrastructure can be a significant barrier to the adoption of these systems.

While IoT-based air pollution monitoring systems provide numerous benefits, they also confront a number of obstacles that must be solved. Sensor accuracy and calibration, data transmission, data processing and storage, data integration and analysis, cost, and public knowledge and acceptability are among the hurdles. These issues must be addressed if IoT-based air pollution monitoring systems are to be successfully implemented and used.

1.2 MOTIVATION

Air pollution is an important environmental concern that has an impact on both human and environmental health. Monitoring air pollution levels is crucial for understanding the scope of the problem and taking corrective action. The Internet of Things (IoT) has transformed the way we interact with our surroundings by enabling real-time data collection from several sources. By collecting and analyzing data on pollutants and their sources, IoT-based air pollution monitoring systems have become a crucial tool in regulating air quality.

The primary goal for designing an IoT-based air pollution monitoring system is to offer reliable and timely data on air quality. Traditional air pollution monitoring systems rely on fixed stations with limited coverage areas and data that may not be obtained in real time. Sensors are installed around an area or city with IoT-based air pollution monitoring systems, and data is collected in real-time, offering a more comprehensive and accurate picture of air quality.

Another motivation for an Air Pollution Monitoring System Based on IoT is to enable policymakers and citizens to take informed decisions based on the data collected. The data collected from the sensors can be analyzed to determine the sources of pollution, the pollutants' concentration levels, and their impact on the environment and human health. This information can be used to develop policies and regulations to control pollution levels and prevent health hazards.

Furthermore, an IoT-based air pollution monitoring system might assist raise public awareness of the issue of poor air quality. The sensors' data can be shared with the public via numerous channels, such as mobile applications and websites, allowing residents to make informed decisions regarding their everyday activities. They can, for example, avoid going outside during peak pollution hours, pick less polluting modes of transportation, and take steps to lessen their carbon footprint.

In decision, an IoT-based air pollution monitoring system is motivated by the need to deliver accurate and timely data on air quality, allowing policymakers and individuals to make educated decisions, and raising public awareness of the air quality issue. IoT-based air pollution monitoring systems can assist reduce the health hazards connected with air pollution and encourage sustainable environmental by giving this information.

1.3 OBJECTIVE

Air pollution is a serious global health and environmental concern, and measuring air quality is critical for identifying and managing pollution sources. Because of its capacity to deliver real-time data and remote monitoring capabilities, Internet of Things (IoT) technology is rapidly being used in air pollution monitoring systems. The following are the primary goals of selecting an IoT-based air pollution monitoring system

One of the primary goals of an IoT-based air pollution monitoring system is to deliver real-time data on air quality. This information can be used to identify pollution sources and take quick action to alleviate the effects of poor air quality on human health and the environment.

Precision and dependability The precision and dependability of air quality data are crucial for making effective policy decisions to enhance air quality. To ensure that the data collected is accurate and dependable, IoT -based air pollution monitoring systems employ a range of sensors and data collection technologies. Cost-effective Because they can be deployed with low-cost sensors and wireless communication networks, IoT-based air pollution monitoring systems are often more cost-effective than traditional monitoring systems.

Scalability and deployment simplicity Air pollution monitoring systems based on IoT are simple to set up and can be scaled up or down depending on the size of the area being monitored. As a result, they are perfect for monitoring air quality in both urban and rural settings. Data visualization and analysis Air pollution monitoring systems based on IoT collect massive volumes of data that may be analyzed and visualized to find trends and patterns in air quality. This information can be utilized to create efficient air quality management plans.
Connection to other systems Air pollution monitoring systems based on IoT can be connected with other systems, such as weather and traffic monitoring systems, to provide a more comprehensive picture of the elements affecting air quality.

Public awareness: By delivering real-time data on air quality levels, IoT-based air pollution monitoring devices can raise public awareness of air quality issues. This can aid in educating the public on the effects of air pollution on health and the environment, as well as encouraging people to take action to limit their own contribution to air pollution.

The management of air pollution in urban and industrial regions can be greatly aided by IoT-based air pollution monitoring systems. The technology can assist people and organizations in acting quickly to lower pollution levels by delivering accurate readings of various pollutants in real-time. The system's data collection can also be utilized to design effective pollution management methods by analyzing it to find trends and patterns.

As pollution levels rise to a risky level, the system should be able to notify people and authorities and provide reports and data visualizations that can be shared with stakeholders. In order to create a thorough pollution management strategy, the system should also be able to interface with other systems including weather monitoring systems, traffic monitoring systems, and emergency response systems.

IoT-based air pollution monitoring systems can offer insightful information on pollution levels, assisting in the preservation of the environment and public health. We can develop more efficient and effective methods to regulate air pollution and build a better, healthier future for everyone by utilizing the potential of IoT technology.

CHAPTER -2
EMBEDDED SYSTEM

2.1 SYSTEM

A framework is a set of instructions where each of its components stores up in sync as shown by regulated manner. It can also be characterized as a method for completing one or more tasks in accordance with a predetermined schedule. For instance, a watch indicates the passage of time. framework. Its components aspire to a lot of requirements to demonstrate time. The watch will stop functioning if one of its components is disrupted. As a result, we can state that the structure's many components are interdependent.

2.2 EMBEDDED SYSTEM

Embedded methods refer to something that is attached to another object, as the name implies. An installed framework can be compared to a piece of PC hardware that has had programming added to it. A framework that has been inserted may operate independently or generally form part of a larger framework. A microcontroller- or chip-based installation framework is one that is designed to carry out a specific purpose. A fire alert, for instance, is an embedded framework that only detects smoke. There are three parts to an embedded framework known as

Hardware

Application programming.

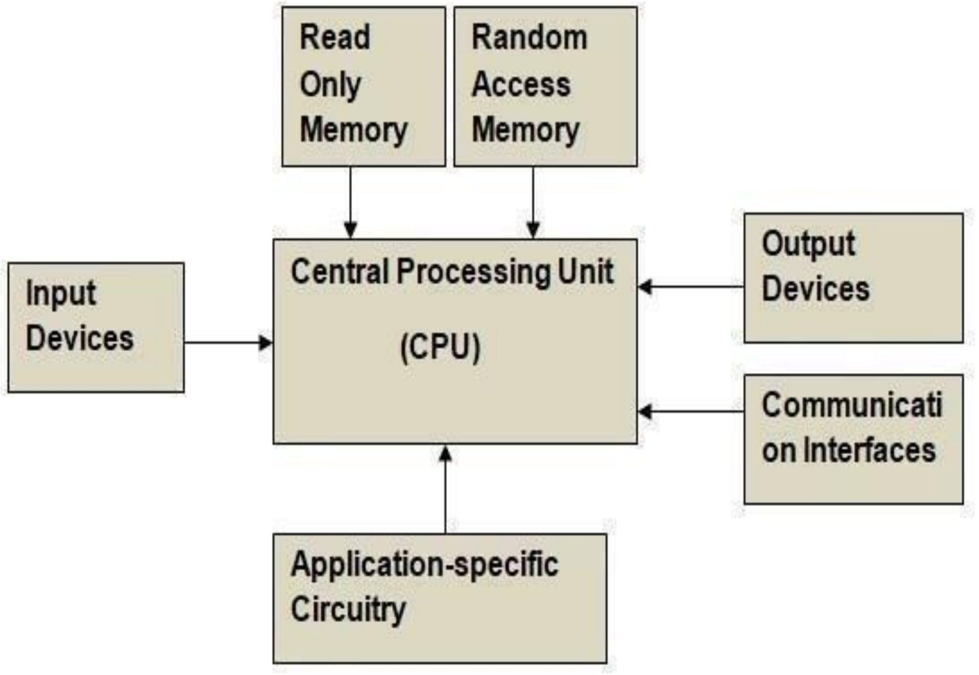


Fig 2.1: EMBEDDED SYSTEM

Real Time Operating Framework (RTOS), which coordinates application programming and provides the processor with an arrangement to conduct a procedure as per arranging by adhering to a technique to control latencies, may be part of an embedded framework. The framework's operation is described by RTOS.

It establishes a few standards whilst the application programming is running. A lightly deployed framework most likely lacks an RTOS.

2.3 EMBEDDED HARDWARE

The most focus required piece of an inserted framework is its equipment. The equipment laid on electronic Printed Circuit Board (PCB). This PCB is classified into five squares. These squares are Processor, Memory, Input gadgets, Output gadgets and Bus controllers.

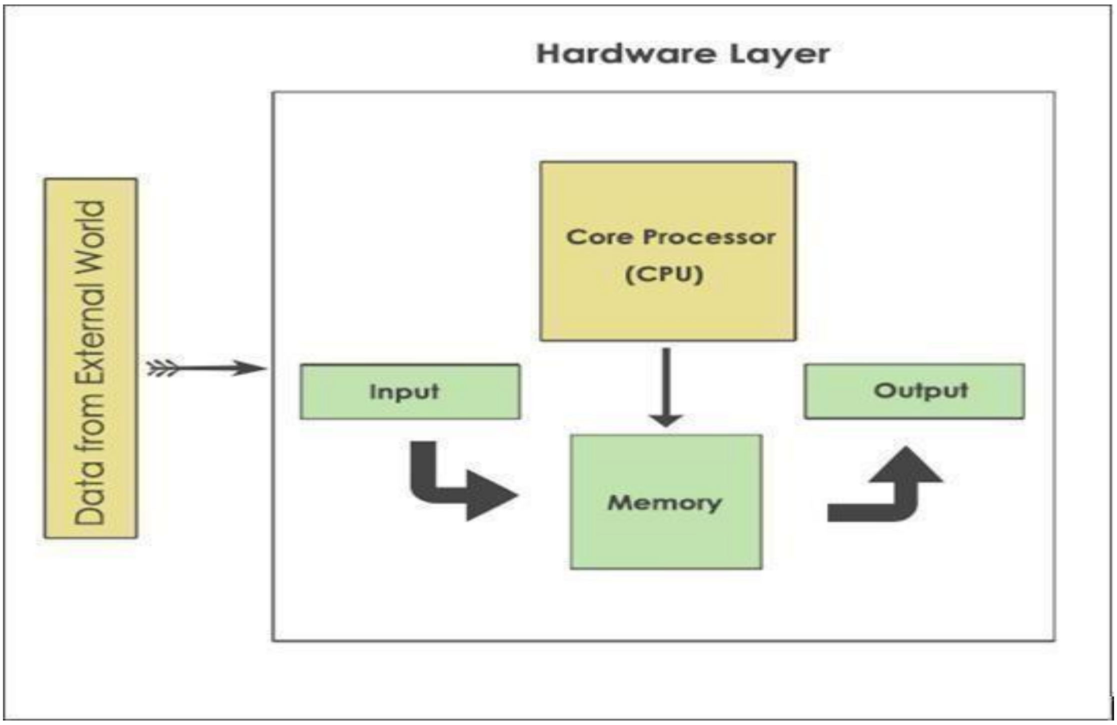


Fig 2.2: HARDWARE COMPONENTS OF AN EMBEDDEDXSYSTEM

Equipment is the focal resource of any inserted gadget and picking a particular component depends upon the essential and detail of the draftsman. In the overall market, there are various varieties of equipment made for different applications. Some of them are:

Microcontroller (CPU)

A Microcontroller is a superior choice for assembling little applications with definite estimation. As a general rule, they have a confined proportion of slam and less solid. A segment of the main associations are Altera, Atmel, Renesas, Infineon, NXP and impressively more. As a matter of fact, a microcontroller is a productive module that plays out the activities delegated by the customer in a capable manner.

System on Chip (SoC)

The SoC comprise of CPU, optional gadgets (Timers, counters), Communication interfaces (I^2C, SPI, UART), and Power Management Circuits on a solitary IC. With the end goal of less execution cost, great execution and high unwavering quality Soc is the most ideal alternative for usage. It likewise gives an opportunity to fuse at least one processor.

ASIC processor

ASIC means Application Specific Integrated Circuit. This kind of chip was specially implemented for the purpose of specific application which was owned by only a single company. The copyrights of this processor were not issued to anyone. ASIC require very low power for operation making it very attractive.

DSP processor

DSP processors are specially designed for digital image and video processing and fits best in Audio and video applications. It has the ability to cutoff noise from image or video and enhances the signal quality of VD player, Music player, and Gaming consoles.

2.4 EMBEDDED SYSTEM SOFTWARE

The product of an installed framework is created to execute a particular undertaking. It is normally written in a significant level arrangement and a short time later assembled down to offer code that can be stuck inside a non-unpredictable memory in the equipment. Installed framework programming is required to remain in context on the going with three cutoff points.

- Comfort capacity of framework memory
- Comfort ability of processor's speed
- If the inserted framework has been running constantly It ought to confine the force dissemination for activities like run, stop and wake up.

2.5 HOW THE SOFTWARE EMBEDDED INTO A SYSTEM

Actually, anyone writes program for embedded system either in assembly or in embedded c language. The program written is compiled first and then transformed into a HEX code. Then this code is loaded or burned into the ROM of the embedded system using some programmer.

These are the tools that are normally exploited in embedded system development

- Assembler
- Emulator
- Debugger
- Compiler

Assembler

The Assembler is used to transform the assembly language program into HEX code. After that we load the program onto the chip by utilizing programmer.

Emulator

An Emulator is a tool either hardware or software which is having analogous functionality of target or guest system. It will have a host system which is an exact duplicate of the target system and allows the host system to execute the functionality of all components and debugs the code, related issues. If the program code is accepted for its operation in the host system, then it would be transferred to the target system.

Debugger

A portion of the time we would not have accomplished the normal results or yield in view of slipups or bug. There are extraordinary instruments that are unequivocally used for the troubleshooting procedure, in which we can take a gander at the control stream and register an incentive to discover the issue.

Compiler

Compiler is programming which is utilized to change code in one programming language into machine language a machine knows about it. The compiler ordinarily presents for changing the elevated level language like c language into the low-level language like machine code, low level computing construct or item code.

2.6 RTOS (REAL TIME OPERATING SYSTEM)

A framework which is fundamental to finish its task and send its organization on schedule, by then just it said to be an ongoing working framework. The RTOS sorts out the application programming and deals with a module to allow the processor run. It is accountable for managing the different gear resources of a PC and besides has applications which continued running on the PC.

This working framework is incredibly proposed to run various applications with a precise arranging and an enormous proportion of consistency. Particularly, this can be enormous in estimation and mechanical robotization frameworks in which a deferral of a program could be an explanation behind danger.

2.7 CHARACTERISTICS OF AN EMBEDDED SYSTEM

- Single-worked – Only single focused on work is performed by an inserted framework on numerous occasions. For example, clothes washer is consistently performs same washing activity.

- Tightly compelled – All processing frameworks have restrictions on plan measurements, be that as it may, those on installed framework can be especially close. Structure measurements is a proportion of improvement features, for instance, its cost, size, force, and execution. Its size ought to be as meager as conceivable to get put on a solitary chip, must perform rapidly enough to process information continuously and use least capacity to extend battery life.

- Reactivex and Real time - some continuous implanted frameworks ought to continually react to varieties in the framework's conditions and should calculate certain results right away. For instance an instance of a vehicle venture controller; it industriously screens and reacts tospeed and brake sensors. It must enlist quickening or de-quickening speeds more than once inside an obliged time; a delayed estimation can prompts powerlessness to control of the vehicle.

- Microprocessors based – An implanted framework should comprise of either microchip or microcontroller.

- Memory – An implanted framework must comprise of a memory module so as to store the program. The product essentially dwells in ROM • Connected − It should gangs peripherals for giving association info and yield gadgets.

- HW-SW frameworks – Software fused for continuous activity and versatility to various conditions. Equipment present in the framework for execution and security.

2.9 APPLICATIONS

Present days embedded is existing every device.

Smart Homes

A large portion of the items in your house are embedded which give brilliant experience and solace to the client. Some devices having embedded systems included are Home Security system, Setup Box, Digital Camera, washing machine, Television, Microwave Oven, Air-conditioned, Refrigerator and much more.

Offices

Embedded systems are also included in commercial outlets for the working of the internet. The examples are Router, Modem, Printer, and Gateways.

Transportation

The car business is well, contending around the world. A portion of the Embedded subsystems in automobiles are Anti-lock Braking framework (ABS), Air conditioning control, Ignition control, Airbag control, Rain detecting wipers.

Healthcare

The health is a basic one, and utilization of embedded frameworks for healthcare services is a bad dream. The odd decision may prompt a heartbreaking impact on the general public or a person. Models are Blood weight screens, Heartbeat screens, pacemakers, tele supervision and medical procedure, Scanners.

Industrial world

The ongoing difficulties of embedded frameworks broadened its extension towards automation. Automation is the way toward completing an assignment redundantly. Automation expands machine efficiency, decreasing product cost and configuration time. Models are Industrial hardware and control, Temperature observing, 3D printing machines.

Aerospace and Defense

Aerospace and Defense is a rough territory where security and execution are generally significant. To accomplish this dependable firmware and embedded programming must be constructed. Some models are Flight control frameworks, actuation, Air and Thermal Management, Engine control, Vehicle turbochargers, and Navigation framework.

CHAPTER-3

LITERATURE REVIEW

3.1 REVIEW OF EXISTING METHOD

There is no current system to monitor the air pollution. The pollution will affect the people in that area. It also affects ozone because the amount of pollutant is not monitored properly and can't know the amount Location update unavailable. This application is not there in GSM mobile. According to the existing system it is not that much eases and comfort for the peoples to view the amount of pollution which is prevailing in the locality. The particular location cannot be found because it does not consist, this cannot be used that much in the society. Thereis no specific level of Indications about of the amount of pollution that has been occurred. This is not handy and not applicable for all the peoples of the society. Less possibility for an early detection of any problem.

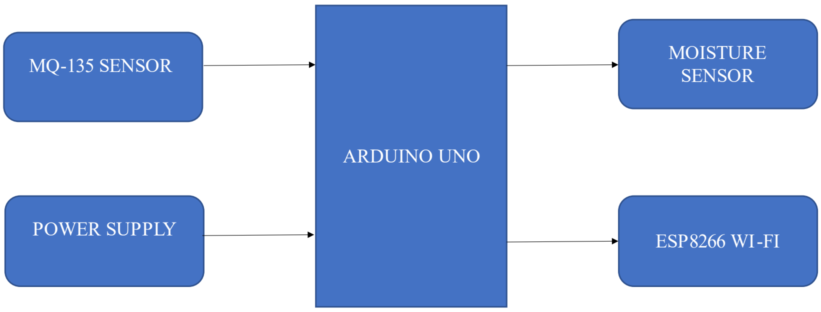


Fig:3.1 Block Diagram of Existing System

The sensors are used to sense the current status in the air. The sensors are connected to the controller. The controller receives the value and converts into digital form and those values are updated in the web server through GSM modem. The default values are already stored in the web server. The current data will be compared with the default values to analyze the result. The pollution status will be uploaded in the web server. So, the user can view anywhere through internet. When there is an pollution that occurs this shows the total amount of gases which is present on the particular locality example CFC (chlorofluorocarbon) carbon-monoxide and many toxic gases can identified by the amount of percentage which is present on it.

The main working principle behind this is IOT which collects information from the cloud which consists of information about the pollution status which is present in our environment. The microcontroller which is used in this device is that microcontroller which consists of 6 outputs and 6inputs so that many sensors can be clubbed together which totally sums up together as an pollution detector and monitoring using an IOT device.

PROCESSING OF SENSORS

The main process of the sensors is that many sensors are being clubbed together as an single device in which many sensors has been used such as humidity sensor temperature sensor gas sensor smoke sensor are all together clubbed as an single device by using the IOT in the cloud which receives amount of pollution which is being existed in locality. Each and every sensor works accordingly so that the percentage of the pollution can be determined which is being shown in the display.

ARDUINO MICROCONTROLLER

Arduino microcontroller is the one in which the many sensors are clubbed together as an single deice so that the values which is obtained can be viewed through the percentage .This microcontroller has an 6 output and 6 input which consists of many sensors which is being attached to it. This microcontroller is used for many applications. The cost is normal and it is can be used mainly clubbing devices which shows output for many connections which is being connected to it.

FUNCTIONS OF SENSORS

The main function of the sensor is that there are many sensors which is present each and every sensor has its different kinds of features which are associated with A sensor device that detects and responds to some type of input from the physical environment. The output is generally a signal that is converted to human readable display at the sensor location or transmitted electronically over a network for reading or further processing.

IOT FUNCTIONS WITH THE CLOUD

Cloud to IOT provides a fully managed service for managing devices. This includes Registration Authentication, and authorization inside the cloud platform resource hierarchy as well as device metadata stored in cloud and the ability to send device configuration from the service to the device.

3.2 OBJECTIVE OF PROJECT

Air pollution is a serious global health and environmental concern, and measuring air quality is critical for identifying and managing pollution sources. Because of its capacity to deliver real-time data and remote monitoring capabilities, Internet of Things (IoT) technology is rapidly being used in air pollution monitoring systems. The following are the primary goals of selecting an IoT-based air pollution monitoring system

One of the primary goals of an IoT-based air pollution monitoring system is to deliver real-time data on air quality. This information can be used to identify pollution sources and take quick action to alleviate the effects of poor air quality on human health and the environment.

Precision and dependability The precision and dependability of air quality data are crucial for making effective policy decisions to enhance air quality. To ensure that the data collected is accurate and dependable, IoT-based air pollution monitoring systems employ a range of sensors and data collection technologies.

Cost-effective Because they can be deployed with low-cost sensors and wireless communication networks, IoT-based air pollution monitoring systems are often more cost-effective than traditional monitoring systems.

Scalability and deployment simplicity Air pollution monitoring systems based on IoT are simple to set up and can be scaled up or down depending on the size of the area being monitored. As a result, they are perfect for monitoring air quality in both urbanand rural settings. Data visualization and analysis Air pollution monitoring systems based on IoT collect massive volumes of data that may be analyzed and visualized to find trends and patterns in air quality. This information can be utilized to create efficient air quality management plans.

Connection to other systems Air pollution monitoring systems based on IoT can be connected with other systems, such as weather and traffic monitoring systems, to provide a more comprehensive picture of the elements affecting air quality.

Public awareness: By delivering real-time data on air quality levels, IoT-based air pollution monitoring devices can raise public awareness of air quality issues. This can aid in educating the public on the effects of air pollution on health and the environment, as well as encouraging people to take action to limit their own contribution to air pollution.

3.3 SCOPE OF THE PROJECT

Air pollution is a global environmental issue that affects human health and the environment. With the advancement of technology, Internet of Things (IoT) has emerged as a promising solution for monitoring air pollution levels. An IoT-based air pollution monitoring system can provide real-time insights into air quality, enabling proactive measures to reduce pollution levels. This project aims to develop an INTERNET OF THINGS BASED AIR POLLUTION MONITORING that can measure various air pollutants such as particulate matter (PM), carbon monoxide (CO), nitrogen dioxide (NO2), and sulfur dioxide (SO2) levels in real-time**.**

Hardware Development:

This aspect of the project involves the development of an IoT device that can measure air pollutants. The IoT device should be portable, lightweight, and easily installable. It should include various sensors that can measure different pollutants such as PM, CO, NO2, and SO2 in real-time.

Sensor Integration:

The IoT device requires the integration of multiple sensors to measure air pollutants accurately. The sensors should be reliable, sensitive, and provide accurate data. Different sensors may be used to measure different pollutants. For example, PM sensors use optical or laser technology to measure particulate matter, while CO sensors use electrochemical technology.

Data Collection and Analysis:

The IoT device collects data continuously and transmits it to a centralized server. The server stores and analyzes the data, providing real-time insights on air quality. The data may be analyzed using statistical methods or machine learning algorithms to detect trends and patterns in air pollution levels.

Data Visualization:

The system generates reports, graphs, and other visual aids that show air pollution levels over a specified period. The visualization tools should be user-friendly and provide meaningful insights. For example, the data may be displayed on a map to show the distribution of pollutants across different areas.

Alerts and Notifications:

The system notifies relevant authorities and individuals when pollution levels exceed safe limits. The notification system should be reliable and timely to enable proactive measures to reduce pollution levels.

Integration with Existing Systems:

The air pollution monitoring system should integrate with existing air quality management systems, such as weather forecasting, to provide comprehensive insights into air quality. For example, the system may use weather data to predict air pollution levels in specific areas.

Environmental Impact Assessment:

The system should evaluate the impact of air pollution on the environment and human health. The assessment should consider the effects of different pollutants and their concentrations. For example, the system may analyze the impact of PM on respiratory health and the environment.

User-Friendly Interface:

The system should have an easy-to-use interface that allows users to access real-time data, generate reports, and receive notifications. The interface should be intuitive and user-friendly, enabling users to navigate the system easily.

The scope of a project on INTERNET OF THINGS BASED AIR POLLUTION MONITORING involves multiple aspects, including hardware development, sensor integration, data collection and analysis, data visualization, alerts and notifications, integration with existing systems, environmental impact assessment, and a user-friendly interface. The project can have a significant impact on the environment and human health by providing real- time insights into air quality and enabling proactive measures to reduce pollution levels.

CHAPTER-4

Materials and Methods

MATERIALS AND METHODS

DESCRIPTION OF COMPONENTS

4.1 ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Revision Of The Board Has The Following New Features

1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. Infuture, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin that is reserved for future purposes. Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous version

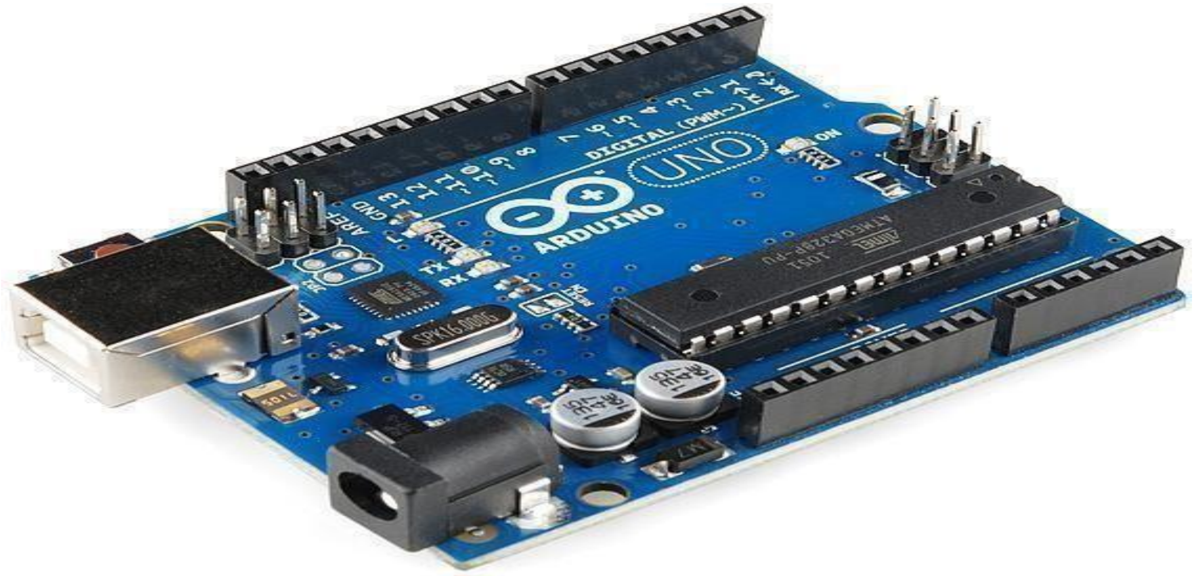


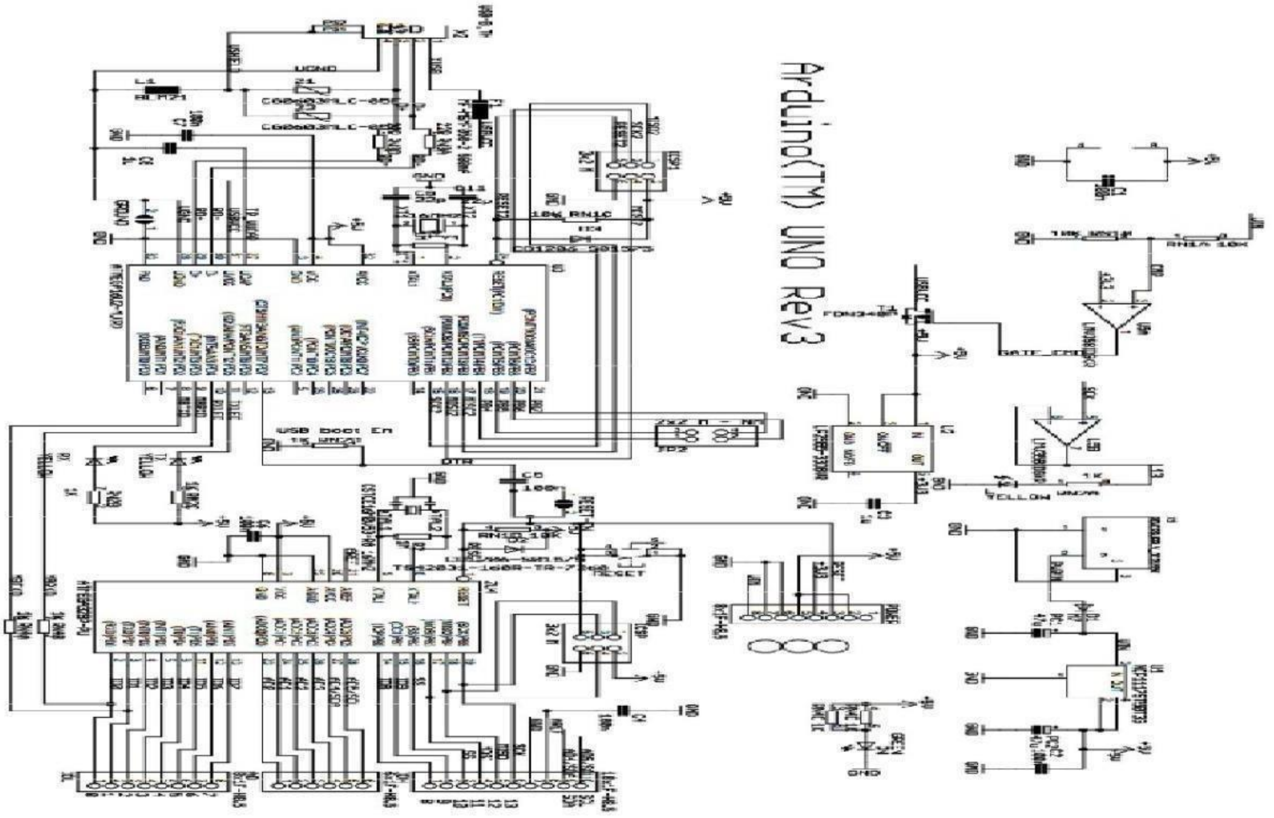
Fig.4.1 ARDUINO UNO

SPECIFICATIONS

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7-12V (recommended)
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mADC Current for 3.3V Pin 50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design:

Figure.4.5: Schematic & Reference Design.



POWER

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) orbattery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supplyvoltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be suppliedwith power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pinof the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

IOREF This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

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

Input And Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write (), and digital Read () functions. They operate at 5 volts. Each pin can provide or receive a maximumof 40 mA and has an internal pull-up resistor (disconnected bydefault) of 20-50 ohms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

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

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

TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analog Reference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shieldswhich block the one on the board.

Communication

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

The Arduino software includes a serial monitor which allows simple textual data to besent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).A Software Serial library allows for serial communication on any ofthe Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software. The ATmega328 onthe Arduino Uno comes pre burned with a boot loader that allows you to upload new code to

it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).You can also bypass the boot loader andprogram the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by: On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easierto put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OSX and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops longenough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OSX or Linux, it resets each time a connection is made to it from software (via USB). For the following half- second or so, the boot loader is running on the Uno.

While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after aconnection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

USB Over Current Protection

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

Physical Characteristics

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

It get information from sensor and process on it. It compares the received data with the threshold level set and accordingly output is generated.

4.2 POWER SUPPLY

Power supplies in recent times have greatly improved in reliability but, because they have to handle considerably higher voltages and currents than any or most of the circuitry they supply, they are often the most susceptible to failure of any part of an electronic system. Modern power supplies have also increased greatly in their complexity, and can supply very stable output voltages controlled by feedback systems. Many power supply circuits also contain automatic safety circuits to prevent dangerous over voltage or over current situations.

A DC Power Supply Unit (commonly called a PSU) deriving power from the AC mains (line) supply performs a number of tasks:

1. It changes (in most cases reduces) the level of supply to a value suitable for driving the load circuit.

2. It produces a DC supply from a pure AC wave.

3. It prevents any AC from appearing at the supply output.

4. It will ensure that the output voltage is kept at a constant level, independent of changes in:

- The AC supply voltage at the supply input.
- The Load current drawn from the supply output.
- Temperature

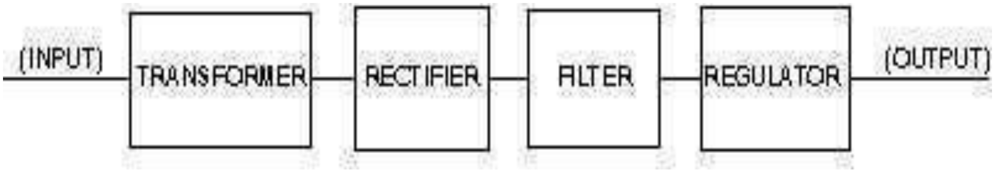


Fig 4.3: Block diagram of power supply

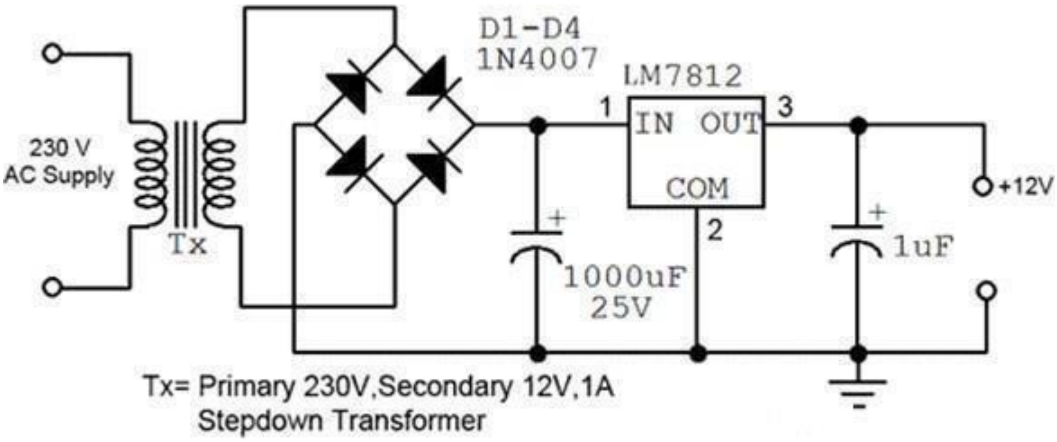


Fig 4.4: circuit diagram of power supply

4.3 LCD

LCD (Liquid Crystal Display) is one sort of a feature found in various contraptions. A liquid valuable stone introduction (LCD) has a level panel display or any electronically modulated optical device which breakers the light-modifying principles of liquid crystals. As LCD show board couldn't release light by liquid diamonds. They have a need of external light source to prepare an image to be seen.

Liquid Crystal Display is fundamental to check the status of any motorized and semi robotized contraptions. This ought to be possible by demonstrating their status on an introduction module, for example, a LCD (Liquid Crystal Display).

16x2 LCD module is a boss among the most extensively seen contraptions open. The liquid jewel show has been supplanted different assorted introductions like 7 piece and others. This is a quick eventual outcome of the various focal points of LCDs for example that they are moderate. It may be balanced reasonably and doesn't have any containments' of demonstrating extraordinary and even custom characters.

A 16x2 LCD show is a critical unit and is in all points of view normally utilized in different contraptions and circuits. A 16x2 LCD addresses, it can indicate 16 characters for every line and there are2 such lines. These devices are much positive than seven pieces and other multi segment LEDs. The reasons being: LCDs are wise; adequately programmable; have no repression of indicating unique and even custom characters (not in the slightest degree like in seven fragments), movements, and so forth.

Pin Description:

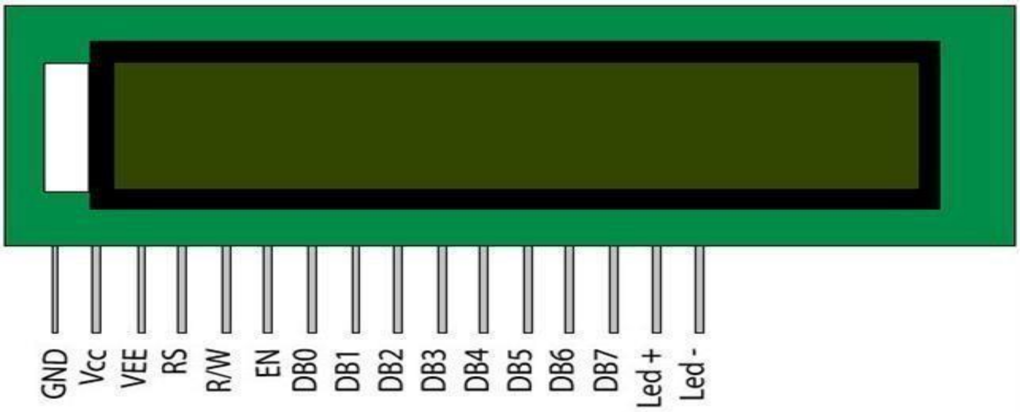
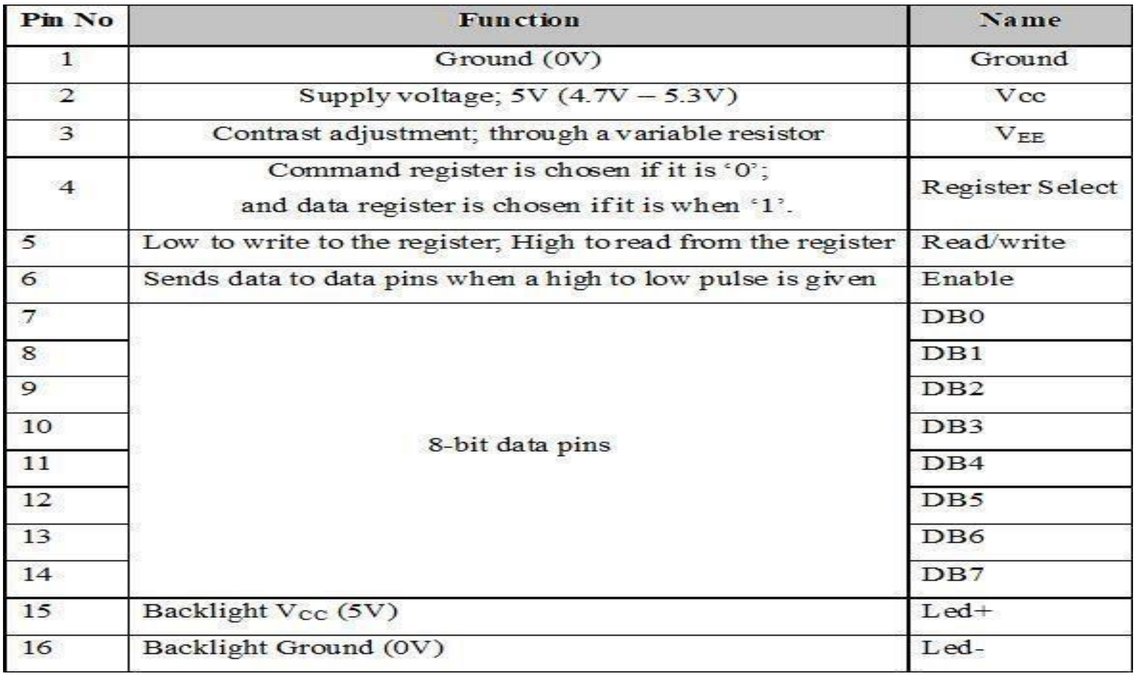


Fig 4.: 5 LCD DISPLAY

The Register select register includes the heading rules that must be gave to the LCD. A coma is an instruction which is given to LCD to finish a fated movement, for instance, presenting it, clearing its screen, setting the cursor position, controlling of introduction so on. The data register involves the data tobe showed up on the LCD. The data stays in the data register is the ASCII code of the character to be appeared on the LCD. Snap to consider internal structure of the LCD.

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	Command register is chosen if it is '0'; and data register is chosen if it is when '1'.	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 4.1: PIN DESCRIPTION OF LCD DISPLAY

RS (Register select)

A 16X2 LCD is having two registers known as direction and information. The registers select empowers the LCD hurl switch between two registers. RS=0 for command register, while RS=1 for information register.

Course Register: The Register select register (request register) contains the request rules that must be offered to the LCD. A request is a direction which is presented to LCD to play out a destined movement,for instance, presenting it, clearing its screen, setting the cursor position, controlling of feature, etc. The system as for headings performed in the request register.

Data Register: The data register includes the data to be showed up on the LCD. The data lives in the data register is the ASCII code of the character to be showed up on the LCD. The data to be showed up on the screen sent to data register and is taken care of there. If RS=1, data register is picked.

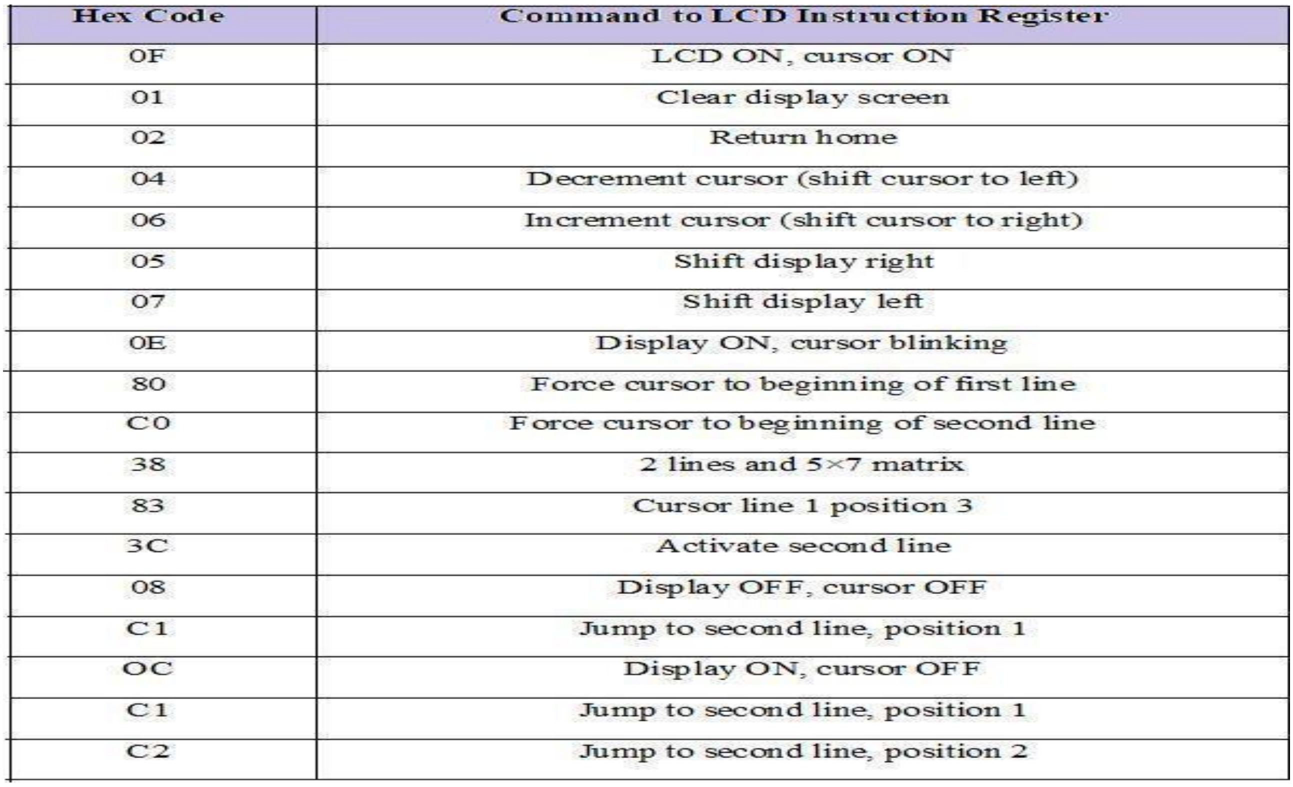
LCD Interfacing

LCD can without a great deal of a stretch be interfaced with a microcontroller to exhibit a message or status of a device. For showing anything in LCD, it must be initialised by giving course of action of bearings to present the LCD or within reset circuit. It is up to the decision of the client to utilize which system. We will utilize the bearings to instate LCD. Initialisation by internal reset gathering When the LCD is turned on, an inside reset circuit displays regularly.

To reasonably process the bearing or information, a delay is required. The concede which is basic is made by defer circle. It is significantly higher than the time taken to process heading or by looking at the clamoring pennant. The last procedure is regularly embraced. The motivation to utilize delay flag is that deferral made is the precise extent of time for which LCD need to process the time. So is most legitimate for each application.

LCD Commands:

Some fixed directions are available in the LCD. These directions are required to work LCD and given by utilizing microcontroller to perform important activities in LCD. Some significant direction guidelines are given underneath:



Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table 4.2: LCD COMMANDS

LCDs are used in a wide extent of electronic circuits, including LCD TVs, PC screens, instrument loads up, plane cockpit shows, and inside and outside signage. Little LCD screens are typical in reduced buyer gadgets, for instance, mechanized cameras, watches, number mini-computers, and phones, including PDAs.

4.4 GSM Module:

GSM resembles the mobile phones where few of the mobile features are not available for the GSM. Similar to mobile phones it can connect to network operator where we can communicate through the sms. The frequency band of the gsm generally varies over 900MHz or 1800MHz. It also has LEDs where it can glow upo giving the power supply of 12v to the gsm sensor. Blue light which indicates the network signal it glows for every 3 seconds .overall the sole purpose of gsm is for communication.

A GSM modem or GSM module is a device that uses GSM mobile telephone technology to provide a wireless data link to a network. GSM modems are used in mobile telephones and other equipment that communicates with mobile telephone networks. They use SIMs to identify their device to the network. A customised Global System for Mobile communication (GSM) module is designed for wireless radiation monitoring through Short Messaging Service (SMS). This module is able to receive serial data from radiation monitoring devices such as survey meter or area monitor and transmit the data as text SMS to a host server.

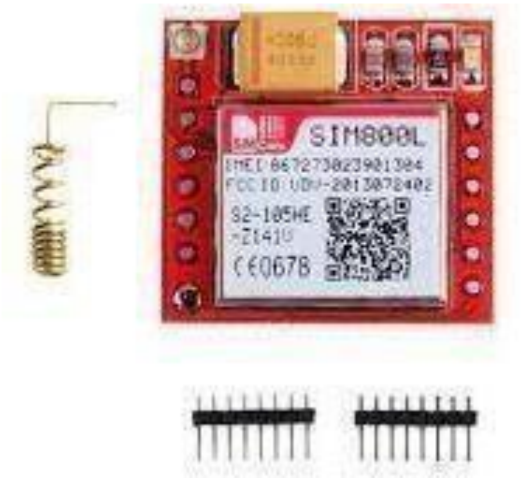


Figure 4.6: GSM (SIM800L)

GSM (Global System for Mobile communication) is a standard created by the European Telecommunications Standards Institute (ETSI) to portray the conventions for second-age advanced cell systems utilized by cell phones.

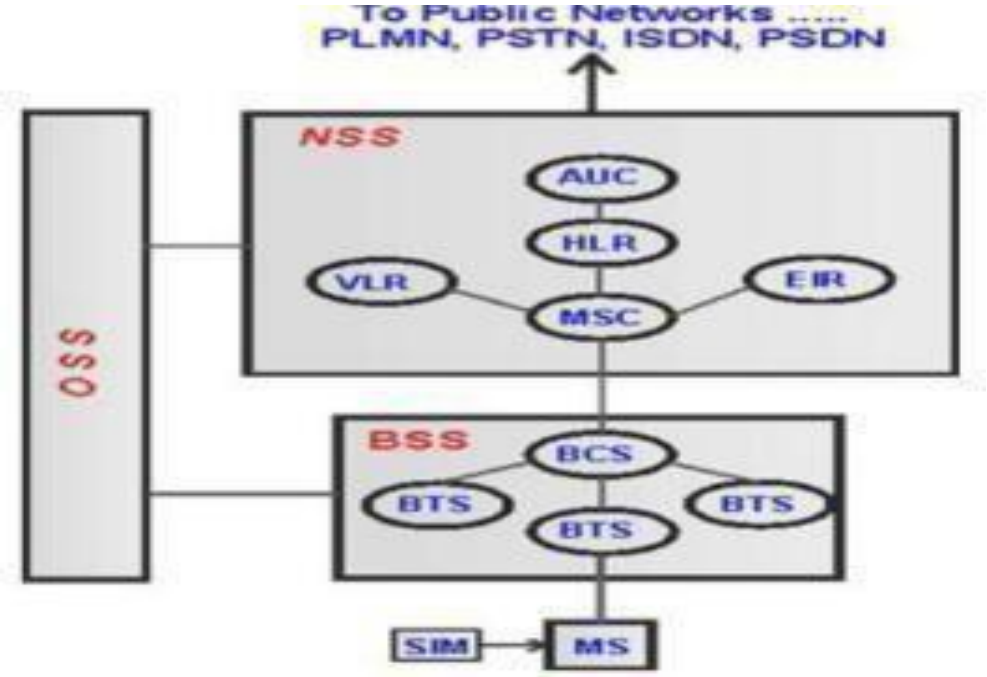


Figure 4.7 GSM NETWORK ARCHITECTURE

The GSM network architecture consists of different elements that all interact together to form the overall GSM system. These include elements like the base-station, controller, MSC, AuC, HLR, VLR, etc.

1. Mobile station

Mobile stations (MS), mobile equipment (ME) are most widely known, cell or mobile phones are the section of a GSM cellular network that the user sees and operates.

There are a number of elements to the cell phone, although the two main elements are the main hardware and the SIM.

2. Base Station Subsystem (BSS)

The Base Station Subsystem (BSS) section of the GSM network architecture that is fundamentally associated with communicating with the mobiles on the network.

3. Network Switching Subsystem (NSS)

The GSM system architecture contains a variety of different elements, and is often termed the core network. It provides the main control and interfacing for the whole mobile network

.

4. **Operation and Support Subsystem (OSS)**

The OSS or activity bolster subsystem is a component inside the general GSM organize engineering that is associated with segments of the NSS and the BSC. It is utilized to control and screen the general GSM system and it is additionally used to control the traffic heap of the BSS. It must be noticed that as the quantity of BS increments with the scaling of the supporter populace a portion of the upkeep undertakings are moved to the BTS, permitting investment funds in the expense of responsibility for framework.

4.5 Wi-Fi Module(ESP8266):

This. ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and MCU (Micro Controller Unit) capability. ESP8266 Serial Wifi Wireless Transceiver Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network.

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much

WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

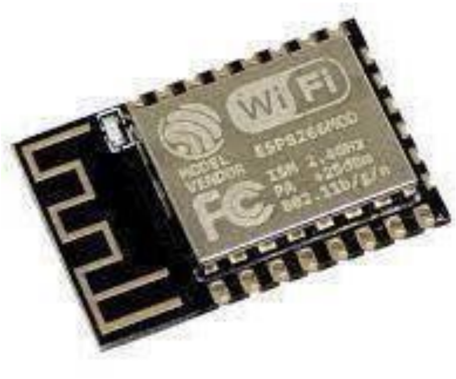


Figure 4.8: Wi-Fi Module(ESP8266)

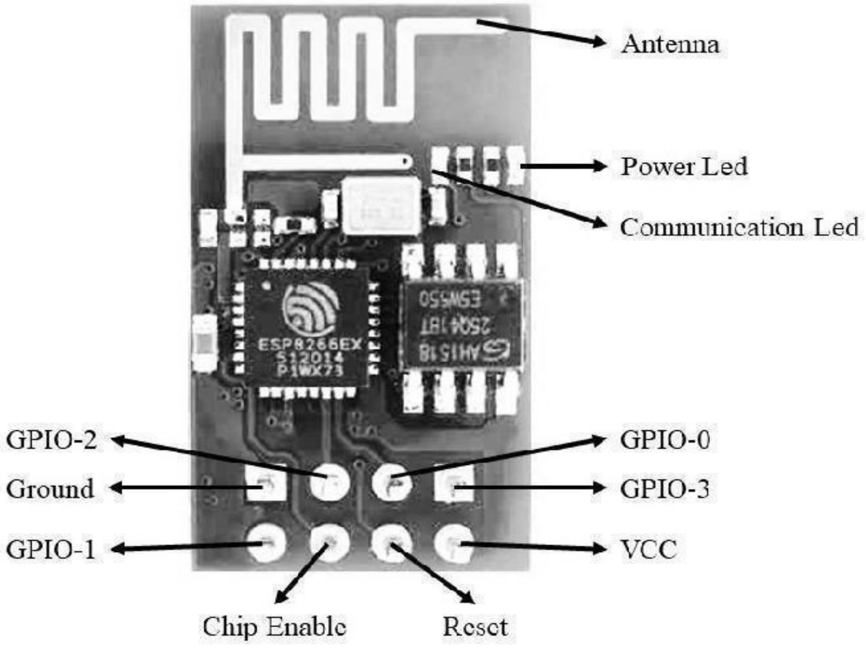


Figure 4.9 : Wi-Fi Module(ESP8266) Pin Configuration

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

Note: The ESP8266 Module is not capable of 5 -3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from your 5V dev board.

The **ESP8266** is a low-cost Wi-Fi microchip, with built-in TCP/IP networking software, and microcontroller capability, produced by Espressif Systemsin Shanghai, China.

The chip was popularized in the English-speaking maker community in August 2014 via the **ESP-01** module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. However, at first, there was almost no English-language documentation on the chip and the commands it accepted. The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, the chip, and the software on it, as well as to translate the Chinese documentation.

Features:

- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Diamond Standard 106Micro running at 80 or 160 MHz
- Memory:
  - 32 KiB instruction RAM
  - 32 KiB instruction cache RAM
  - 80 KiB user-data RAM
  - 16 KiB ETS system-data RAM
- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- IEEE 802.11 b/g/n Wi-Fi
  - Integrated TR switch, balun, LNA, power amplifier and matching network
  - WEP or WPA/WPA2 authentication, or open networks
- 17 GPIO pins
- Serial Peripheral Interface Bus (SPI)
- I²C (software implementation)
- I²S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10 Bit ADC

4.6 MQ-2 GAS SENSOR

The MQ-2 gas sensor is a popular type of gas sensor module used to detect different types of smoke . It is a small-sized and low-cost sensor that is widely used in Smoke detection systems, fire detection systems, and other safety equipment. The MQ-2 sensor module consists of a gas sensor unit and a signal conditioning circuit. The gas sensor unit contains a sensing element which is made up of a tin dioxide (SnO2) semiconductor material that has a high sensitivity to different types of smoke. The signal conditioning circuit amplifies the output signal from thegas sensor unit and converts it into a voltage signal that can be easily read by a microcontroller or other digital device.

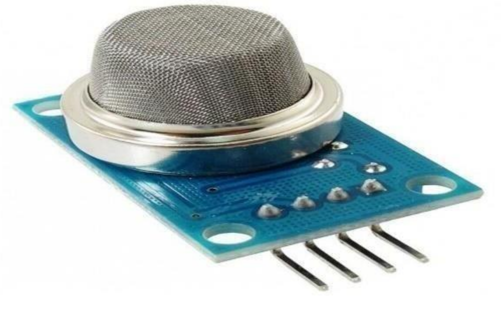


Fig. 4. 10 MQ-2 Gas Sensor

The MQ-2 gas sensor operates on the principle of chemical reactions between the target gas and the SnO2 sensing material. When a gas molecule comes into contact with the sensing material, it undergoes an oxidation or reduction reaction which changes the resistance of the sensing material. This change in resistance is then converted into a voltage signal by the signal conditioning circuit.

The MQ-2 gas sensor module requires a stable power supply of 5V DC and consumes very low power (less than 150mW). It has a wide detection range for different gases, with a sensitivity of 300 to 10,000 parts per million (ppm) for different gases.

4.7 MQ-135 GAS SENSOR

The MQ-135 gas sensor is a popular type of gas sensor module used to detect different types of gases such as benzene, nitrogen oxides, ammonia, and other harmful gases. It is a low-cost sensor that is widely used in air quality monitoring systems, indoor air quality monitoring, and other applications.

The MQ-135 sensor module consists of a gas sensor unit and a signal conditioning circuit. The gas sensor unit contains a sensing element made up of a tin dioxide (SnO2) semiconductor material that has a high sensitivity to different types of gases. The signal conditioning circuit amplifies the output signal from the gas sensor unit and converts it into a voltage signal that can be easily read by a microcontroller or other digital device.

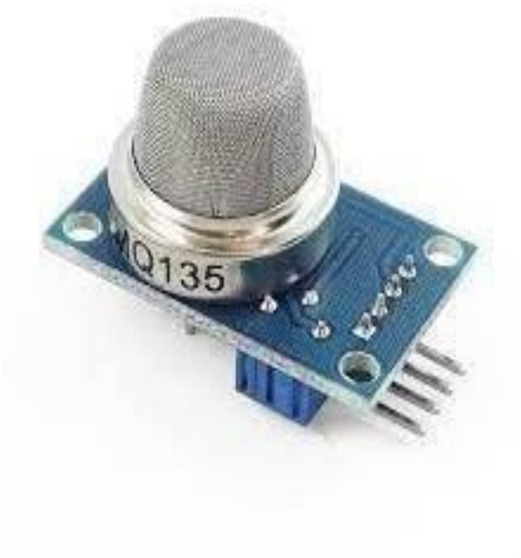


Fig. 4. 11 MQ-135 GAS SENSOR

The MQ-135 gas sensor operates on the principle of chemical reactions between the target gas and the SnO2 sensing material. When a gas molecule comes into contact with the sensing material, it undergoes an oxidation or reduction reaction which changes the resistance of the sensing material. This change in resistance is then converted into a voltage signal by the signal conditioning circuit.

The MQ-135 gas sensor module requires a stable power supply of 5V DC and consumes very low power (less than 150mW). It has a detection range of 10 to 1000 ppm for different gases, with a sensitivity of 1 ppm for some gases.

it is important to note that the MQ-135 sensor has some limitations. It is sensitive to changes in temperature and humidity, which can affect its accuracy. It is also not able to detect gases such as hydrogen and helium due to their small molecular size.

Overall, the MQ-135 gas sensor is a versatile and low-cost sensor module that can be used for gas detection in a variety of applications. It is commonly used in air quality monitoring systems to detect harmful gases and pollutants, and to provide alerts and warnings to users when air quality is poor.

4.8 MQ-5 GAS SENSOR

The MQ-5 gas sensor is a popular type of gas sensor module used to detect different types of gases such as liquefied petroleum gas (LPG), propane, natural gas, methane, and other combustible gases. It is a small and low-cost sensor that is widely used in gas leakage detection systems, fire detection systems, and other safety equipment.

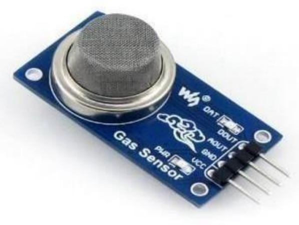


Fig. 4. 12 MQ-5 GAS SENSOR

The MQ-5 sensor module consists of a gas sensor unit and a signal conditioning circuit. Thegas sensor unit contains a sensing element made up of a tin dioxide (SnO2) semiconductor material that has a high sensitivity to different types of gases. The signal conditioning circuit amplifies the output signal from the gas sensor unit and converts it into a voltage signal that can be easily read by a microcontroller or other digital device.

The MQ-5 gas sensor operates on the principle of chemical reactions between the target gas and the SnO2 sensing material. When a gas molecule comes into contact with the sensing material, it undergoes an oxidation or reduction reaction which changes the resistance of the sensing material. This change in resistance is then converted into a voltage signal by the signal conditioning circuit.

The MQ-5 gas sensor module requires a stable power supply of 5V DC and consumes very low power (less than 150mW). It has a detection range of 200 to 10,000 ppm for different gases, with a sensitivity of 300 ppm for LPG and 500 ppm for methane.

However, it is important to note that the MQ-5 sensor has some limitations. It is sensitive to changes in temperature and humidity, which can affect its accuracy. the MQ-5 gas sensor is a versatile and low-cost sensor module that can be used for gas detection in a variety of applications. It is commonly used in gas leakage detection systems to detect combustible gases and to provide alerts and warnings to users when gas leaks are detected.

4.9 MQ-7 GAS SENSOR

The MQ-7 gas sensor is a popular type of gas sensor module used to detect carbon monoxide (CO) gas. It is a small-sized and low-cost sensor that is widely used in gas leakage detection systems, fire detection systems, and other safety equipment.

Fig 4. 13. MQ-7 GAS SENSOR

contains a sensing element made up of a tin dioxide (SnO2) semiconductor material that has a high sensitivity to carbon monoxide gas. The signal conditioning circuit amplifies the outputsignal from the gas sensor unit and converts it into a voltage signal that can be easily read bya microcontroller or other digital device.

The MQ-7 gas sensor operates on the principle of chemical reactions between the carbon monoxide gas and the SnO2 sensing material. When carbon monoxide comes into contact with the sensing material, it reduces the resistance of the sensing material. This change in resistance is then converted into a voltage signal by the signal conditioning circuit.

The MQ-7 gas sensor module requires a stable power supply of 5V DC and consumes very low power (less than 150mW). It has a detection

range of 20 to 2000 ppm for carbon monoxide gas, with a sensitivity of 1 ppm.

it is important to note that the MQ-7 sensor has some limitations. It is sensitive to changes in temperature and humidity, which can affect its accuracy. It may also give false readings if exposed to certain types of gases such as ethanol and propane.

Overall, the MQ-7 gas sensor is a versatile and low-cost sensor module that can be used for gas detection in a variety of applications. It is commonly used in gas leakage detection systems to detect carbon monoxide gas and to provide alerts and warnings to users when carbon monoxide levels are high.

4.10 MQ-8 GAS SENSOR

The MQ-8 gas sensor is a type of gas sensor that is commonly used to detect hydrogen gas. It is a small, affordable, and versatile sensor that can be used in a variety of applications, suchas gas leakage detection, fuel cell monitoring, and hydrogen-powered vehicle applications.

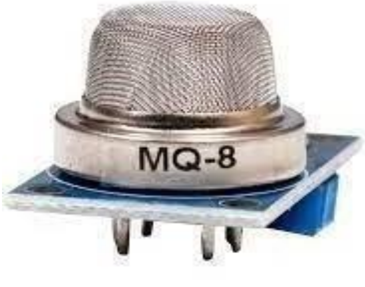


Fig. 4. 14 MQ-8 GAS SENSOR

Here is some complete information about the MQ-8 gas sensor The MQ-8 gas sensor works on the principle of the interaction between hydrogen gas and tin dioxide (SnO2) semiconductor material. When hydrogen gas comes in contact with the SnO2 material, it changes the electrical conductivity of the material, which can be measured by the sensor's circuitry. The change in conductivity is proportional to the concentration of hydrogen gas in the air.

Sensitivity and detection range:

The MQ-8 gas sensor has a high sensitivity to hydrogen gas and can detect concentrations as low as 100 ppm

(parts per million) up to 10,000 ppm. The detection range can be adjusted by varying the load resistance of the

sensor's circuit.

Response time:

The response time of the MQ-8 gas sensor is typically less than 10 seconds when exposed to a sudden change

in the concentration of hydrogen gas.

Power consumption:

The MQ-8 gas sensor operates on a low voltage of 5V and has a low power consumption of about 150mW.

Output:

The MQ-8 gas sensor provides an analog output voltage that is proportional to the concentration of hydrogen gas in the air. The output voltage can be measured using an analog- to-digital converter (ADC) or microcontroller.

Calibration:

The MQ-8 gas sensor needs to be calibrated before use to ensure accurate readings. Calibration involves exposing the sensor to a known concentration of hydrogen gas and adjusting the load resistance of the circuit until the output voltage matches the expected value.

Environmental factors:

The MQ-8 gas sensor is sensitive to temperature and humidity, which can affect its accuracy. Therefore, it should be operated within a specific temperature and humidity range to ensure accurate readings.

the MQ-8 gas sensor is a reliable and affordable option for detecting hydrogen gas in various applications. However, it is important to follow proper calibration and operating procedures to ensure accurate readings and reliable performance.

4.11 BMP180 SENSOR

The BMP180 sensor is a highly precise and low-power digital barometric pressure sensor that is widely used in various applications such as mobile devices, GPS modules, weather forecasters, and more. It is a single -chip package that includes a MEMS pressure sensor, an analog-to-digital converter, and a control unit.

The BMP180 sensor is capable of measuring atmospheric pressure in the range of 300 to 1100 hPa with an accuracy of ±0.12 hPa, equivalent to ±1m altitude difference. Additionally, it can measure temperature in the range of -40°C to +85°C with an accuracy of ±1°C. The sensor uses the Inter-Integrated Circuit (I2C) digital interface, which is a common protocol used in microcontrollers and other host devices.

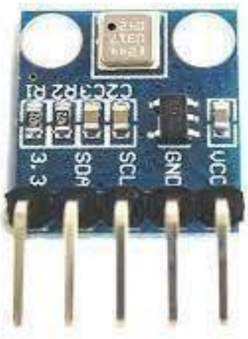


Fig.4.15 BMP180 SENSOR

The BMP180 sensor is a small form factor device with dimensions of 3.6mm x 3.8mm x 0.93mm, which makes it an ideal choice for compact designs. It has a low power consumption of 5μA in standard mode and 0.5μA in ultra-low power mode, making it suitable for battery-powered devices that require extended operating times.

The BMP180 sensor uses a capacitive pressure sensor to measure atmospheric pressure, and an on -board temperature sensor to measure ambient temperature. The sensor converts the analog signals from the pressure and temperature sensors into digital signals, which are then sent over the I2C interface to the microcontroller or other host device.

the BMP180 sensor is a versatile and accurate sensor that is commonly used in many different applications that require pressure and temperature measurements. Its small form factor, low power consumption, and high precision make it a popular choice for many different projects

4.12 Temperature Humidity Sensor(DHT11):

The DHT-11 Digital Temperature And Humidity Sensor is a basic, ultra low-cost 37 digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). Fig.3.18. Temperature and Humidity Sensor.

The **DHT11** is a commonly used **Temperature and humidity sensor that** comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

The DHT-11 Digital Temperature And Humidity Sensor is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so in your code please use sensor reading interval at 2 seconds or more. Compared to the DHT22 this sensor is less precise, less accurate and works in a smaller range of temperature/humidity.

But despite its disadvantages over DHT22, it is smaller and less expensive sensor for temperature and humidity measurement.

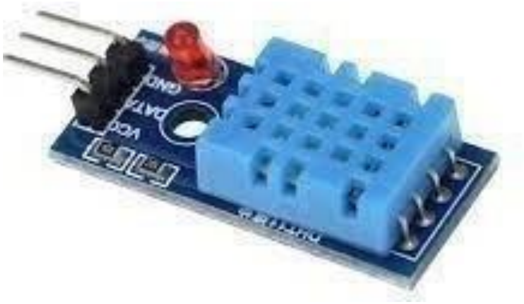


Figure 4. 16 : DHT11 Sensor

4.13 Buzzer:

Buzzer is used to add the sound feature. It is light in weight,Good performance, general purpose musical buzzer are commonly used in alerting / alarming circuits, kids toys etc.. This buzzer is used as an external buzzer which operates in wide range of voltage (3V to 12V). Most commonly used buzzers operates in at 9V & 12v. They have long life, stable performance, High Quality with the SOT plastic package.

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, train and confirmation of user input such as a mouse click or keystroke.

Piezoelectric buzzers, or piezo buzzers, as they are sometimes called, were invented by Japanese manufacturers and fitted into a wide array of products during the 1970s to 1980s. This advancement mainly came about because of cooperative efforts by Japanese manufacturing companies. In 1951, they established the Barium Titanate Application Research Committee, which allowed the companies to be "competitively cooperative" and bring about several piezoelectric innovations.

Figure 4. 17: Buzzer

4.14 MOTOR DRIVEN

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a highercurrent signal. This higher current signal is used to drive the motors. L293D contains two inbuilt H -bridge driver circuits. The L293D is a 16-pin Motor Driver IC which can control a set of two DC motors simultaneously in any direction. The L293D is designed to provide bidirectional drive currents of up to 600 mA (per channel) at voltages from 4.5 V to 36 V (at pin 8!). You can use it to control small dc motors - toy motors.

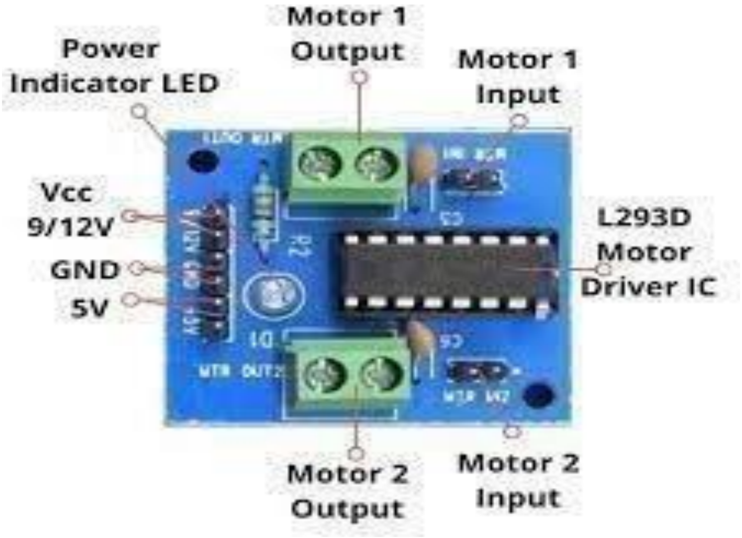


Figure 4.18: Motor driver pin description

4.15 Exhaust fan

They remove smoke and odors – we've all experienced the smoky dinner, or maybe you have a smoker in your home. Exhaust fans help to quickly remove smoke and odors, making your indoor air more breathable.

They help improve comfort – exhaust fans help maintain circulation and remove excess moisture, increasing your overall indoor comfort.

Fig. 4.19 Exhaust fan

4.2 SOFTWARE REQUIREMENTS

SOFTWARE USED

4.2.1 ARDUINO IDE-SOFTWARE

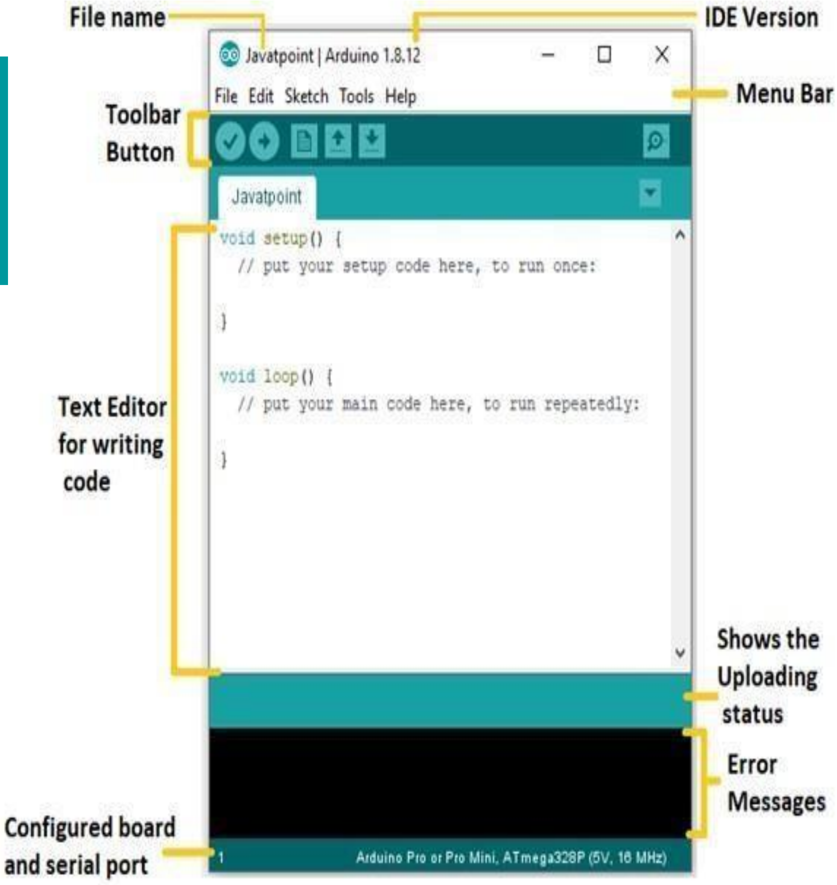
Node MCU flasher is a firmware programmer for Node MCU DEVKIT V0.9. You can use it to program Node MCU DEVKIT or your own ESP8266 board. You MUST set GPIO0 to LOW before programming, and Node MCU DEVKIT V0.9 will do it automatically.

Step 1: Installing Arduino IDE Software

Install Arduino IDE software from the link http://www.arduino.cc/en/main/software.

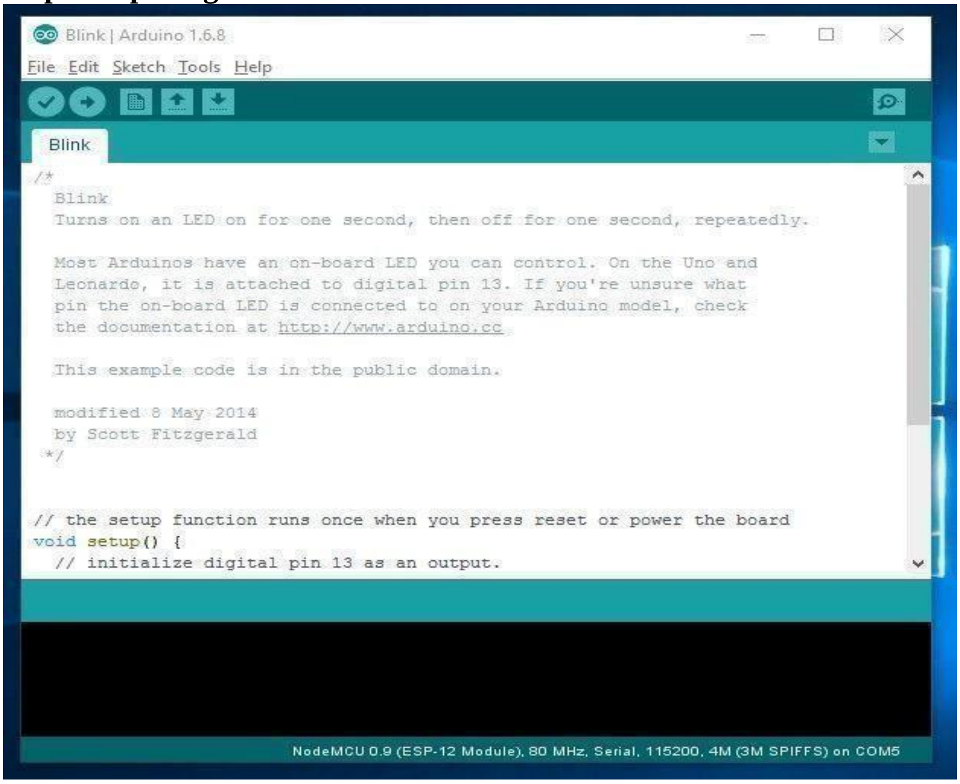
Step 2: Arduino IDE Icon

After installing Arduino IDE icon is created on the Desktop as show in the figure


File name
IDE Version
Javatpoint | Arduino 1.8.12
File Edit Sketch Tools Help
Menu Bar
Toolbar Button
Javatpoint
Text Editor for writing code
Shows the Uploading status
Error Messages
Configured board and serial port
1
Arduino Pro or Pro Mini, ATmega328P (5V, 16 MHz)

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

}

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

}
```

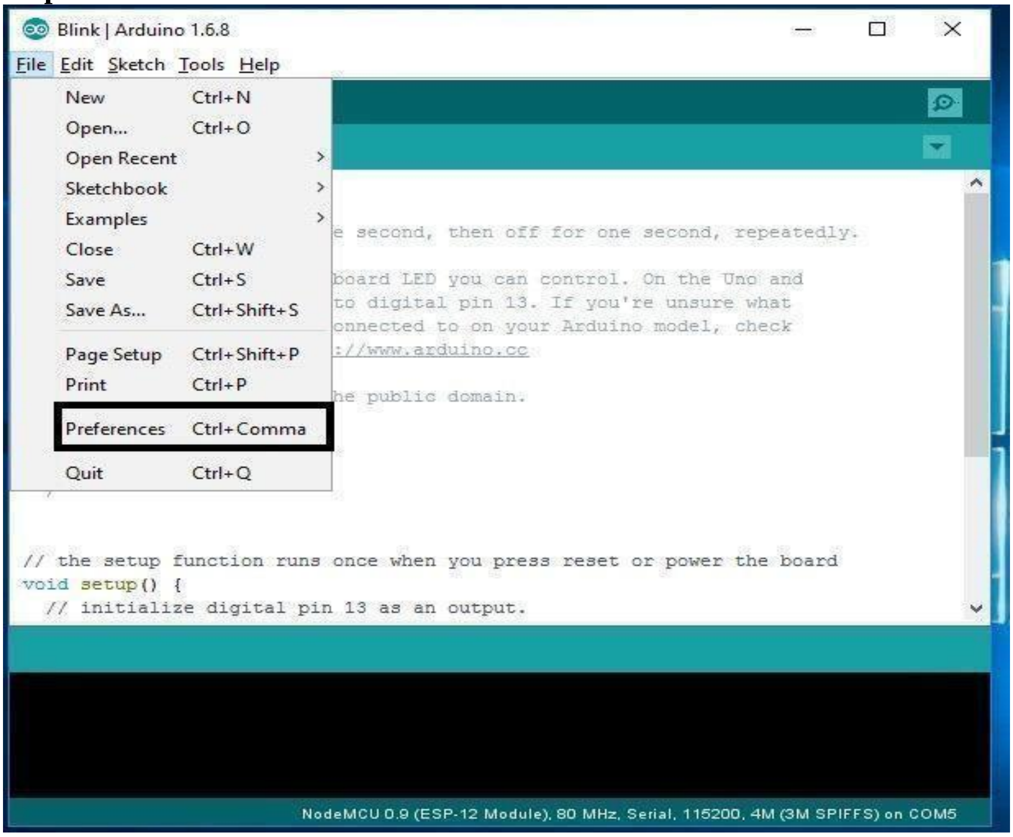
Step 3: Opening Arduino IDE


Blink | Arduino 1.6.8
File Edit Sketch Tools Help
Blink
NodeMCU 0.9 (ESP-12 Module), 80 MHz, Serial, 115200, 4M (3M SPIFFS) on COM5

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
```

Click on the Icon to open the Arduino window as shown in the figure

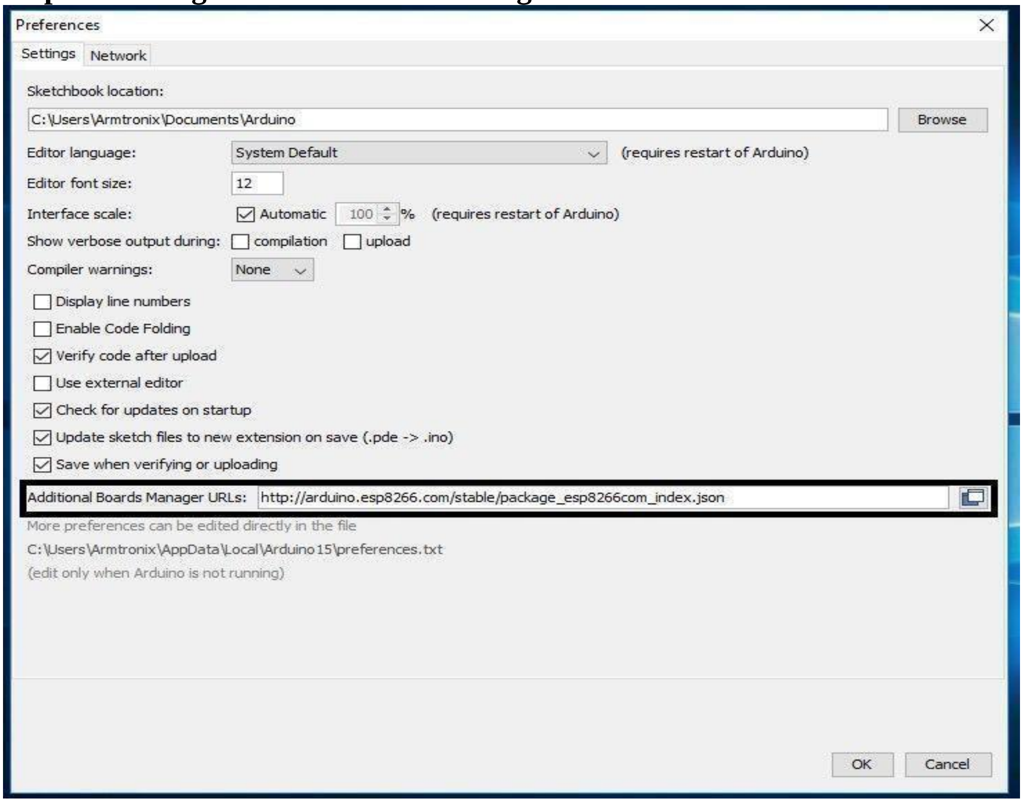
Step 4: Preferences



```
e second, then off for one second, repeatedly.

board LED you can control. On the Uno and
to digital pin 13. If you're unsure what
onnected to on your Arduino model, check
://www.arduino.cc

he public domain.


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
```

Open the File and click on the Preferences as shown in the figure

Step 5: Adding ESP8266 Board Manager



Preferences

Settings Network

Sketchbook location:

C:\Users\Armtronix\Documents\Arduino Browse

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers

□ Enable Code Folding

☑ Verify code after upload

□ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

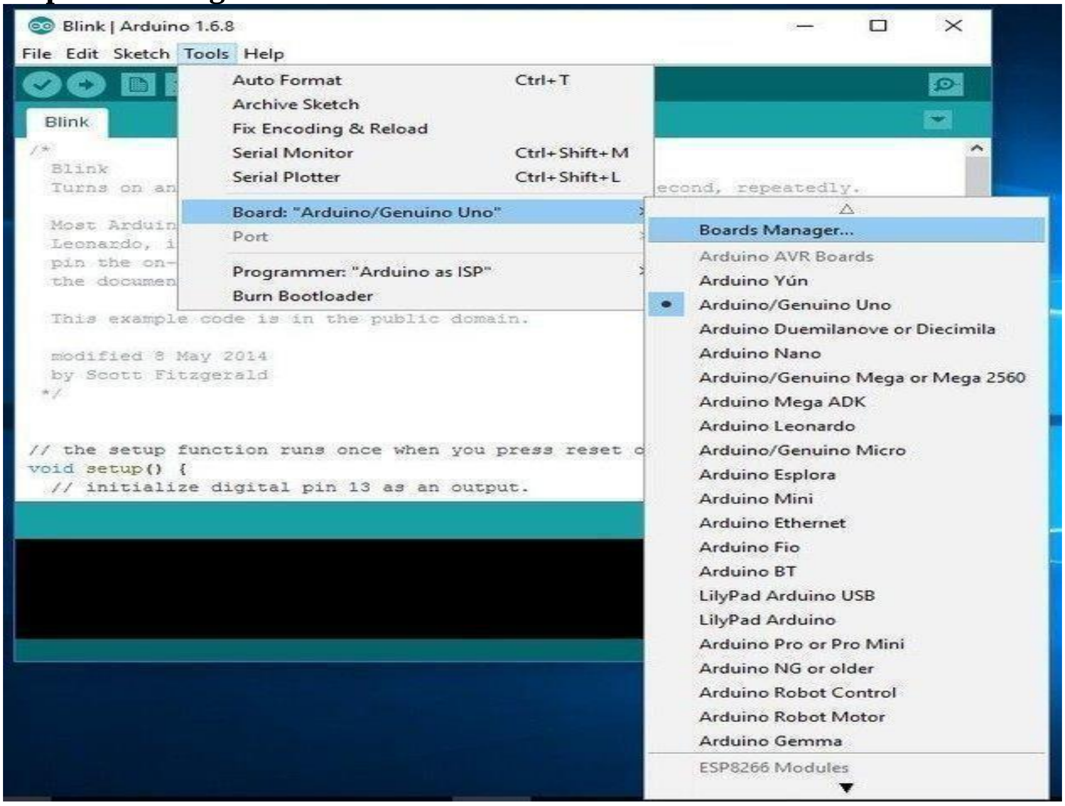
C:\Users\Armtronix\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

OK Cancel

In the Additional Boards Manager enter below URL. http://arduino.esp8266.com/stable/package_esp8266com_index.json As highlighted in the figure and enter OK.

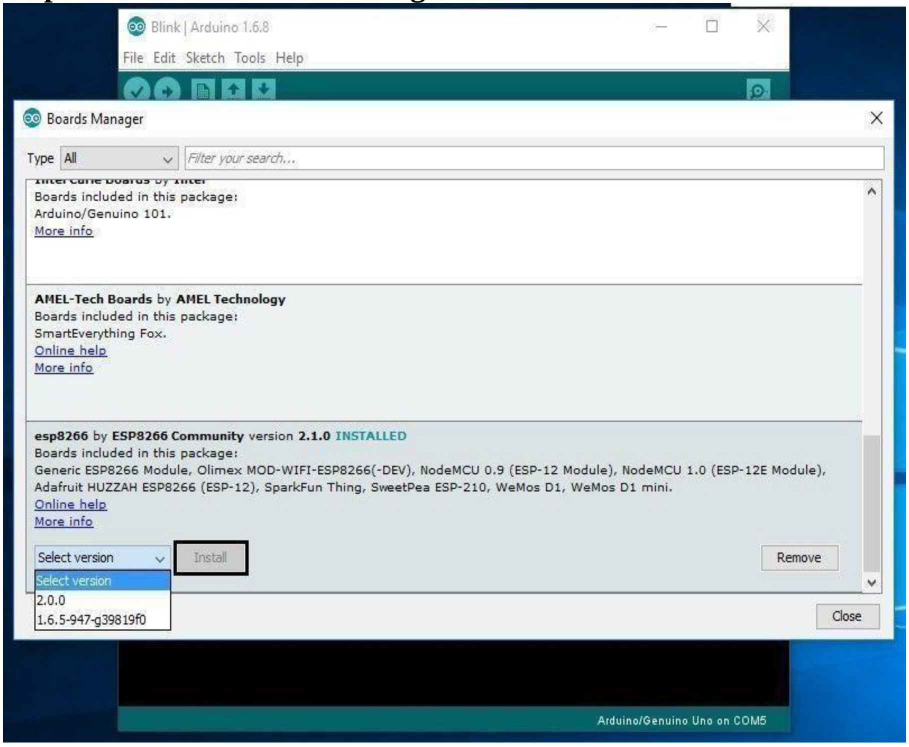
Step 6: Selecting Board



```
/*
  Blink
  Turns on an                                   econd, repeatedly.

  Most Arduin
  Leonardo, i
  pin the on-
  the documen

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset o
void setup() {
  // initialize digital pin 13 as an output.
```

Now open the tools in that select **Board: “Arduino/Genuino Uno”** and click on the **Boards Manager** asshown in the figure

Step 7: ESP8266 Board Package



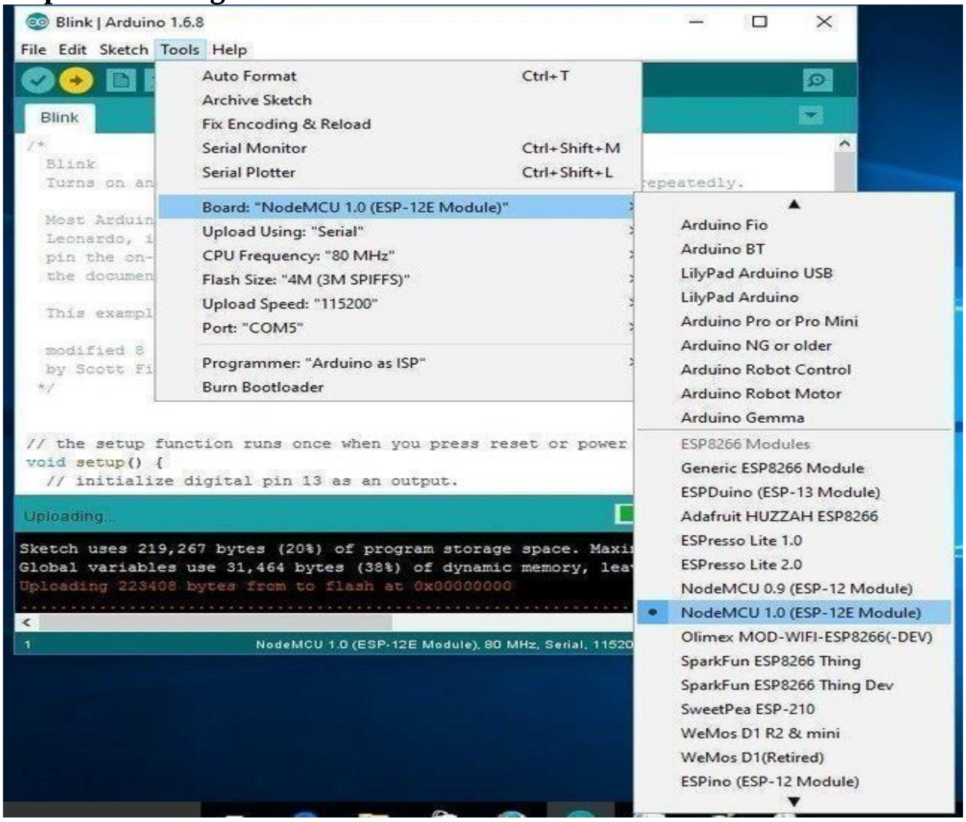
Intel Curie Boards by Intel
Boards included in this package:
Arduino/Genuino 101.
More info

AMEL-Tech Boards by **AMEL Technology**
Boards included in this package:
SmartEverything Fox.
Online help
More info

esp8266 by **ESP8266 Community** version **2.1.0** INSTALLED
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini.
Online help
More info

The Boards Manager window opens, scroll the window page to bottom till you see the module with the name ESP8266. Once we get it, select that module and select version and click on the Install button. When it is installed it shows Installed in the module as shown in the figure and then close the window.

Step 8: Selecting ESP8266 Arduino Board



Board: "NodeMCU 1.0 (ESP-12E Module)"
Upload Using: "Serial"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"
Port: "COM5"

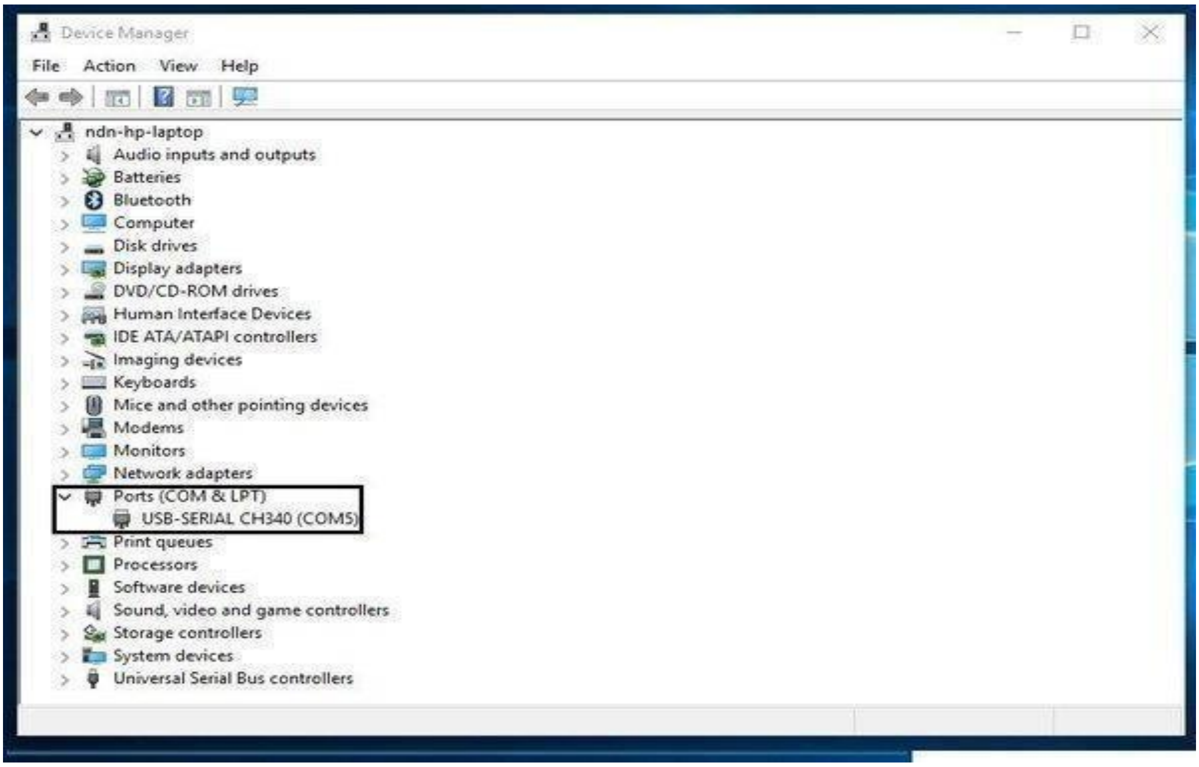
```
/*
  Blink
  Turns on an
  ...repeatedly.

  Most Arduin
  Leonardo, i
  pin the on-
  the documen

  This exampl

  modified 8
  by Scott Fi
 */

// the setup function runs once when you press reset or power
void setup() {
  // initialize digital pin 13 as an output.
```

```
Sketch uses 219,267 bytes (20%) of program storage space. Maxi
Global variables use 31,464 bytes (38%) of dynamic memory, lea
Uploading 223408 bytes from to flash at 0x00000000
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Serial, 11520

To run the esp8266 with Arduino we have to select the **Board: “Arduino/Genuino Uno”** and then change it to **NodeMCU 1.0 (ESP-12E Module)** or other esp8266 modules depending on what you have .This can be done by scrolling down, as shown in the figure.

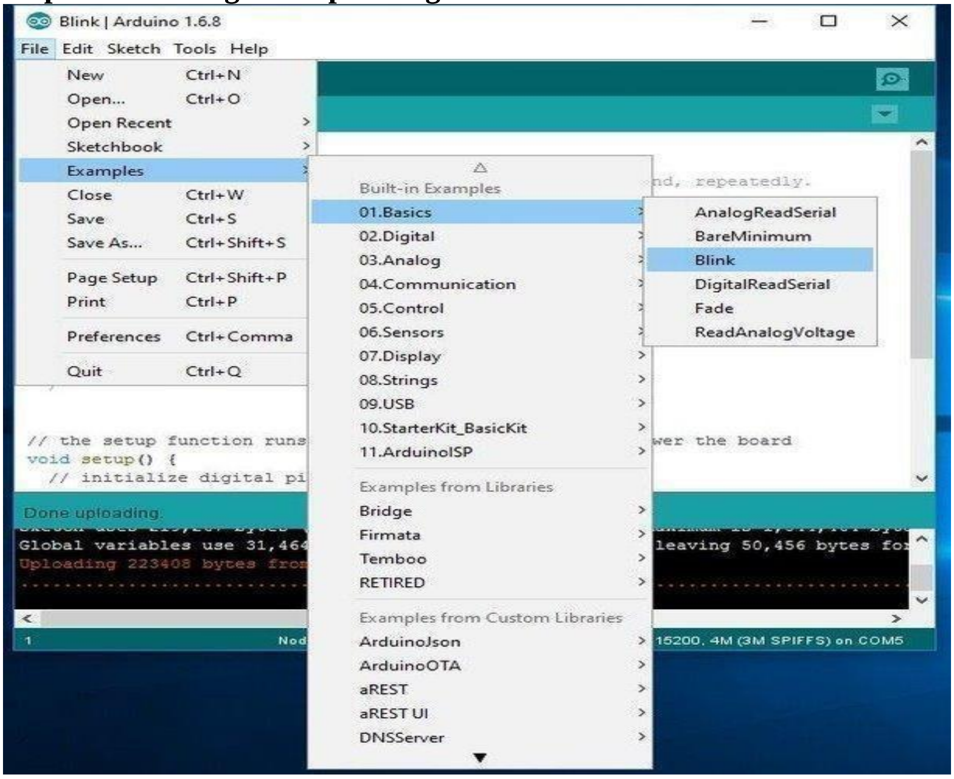
Step 9: Connecting ESP8266 to the PC



ndn-hp-laptop
- Audio inputs and outputs
- Batteries
- Bluetooth
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Imaging devices
- Keyboards
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- Ports (COM & LPT)
  - USB-SERIAL CH340 (COM5)
- Print queues
- Processors
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

Now Let’s connect the ESP8266 module to your computer through USB cable as shown in the figure.When module is connected to the USB, COM port is detected eg: here COM5 is shown in the figure.

Step 10: Selecting Example Program in Arduino IDE

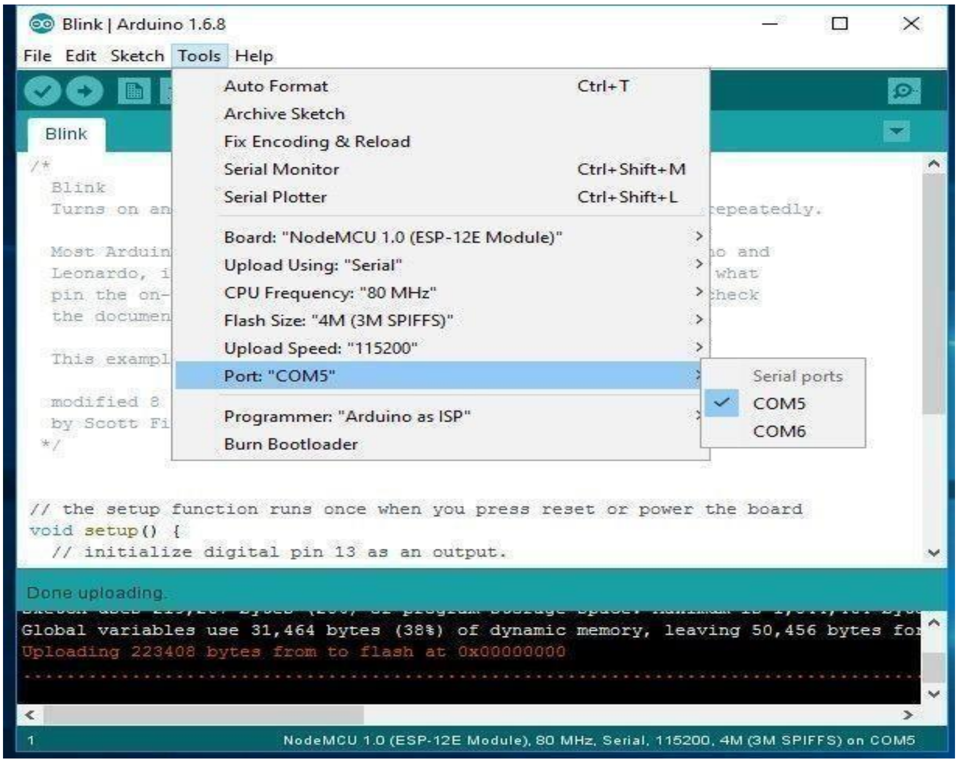


```
nd, repeatedly.

// the setup function runs                wer the board
void setup() {
  // initialize digital pi
```

```
Global variables use 31,464               leaving 50,456 bytes fo
Uploading 223408 bytes from
```

Now open the File tab in that go to the Examples in that enter into Built-in example, go to 01.Basics andclick on Blink to open the window.

Step 11: Selecting COM Port



Board: "NodeMCU 1.0 (ESP-12E Module)"
Upload Using: "Serial"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"
Port: "COM5"
Programmer: "Arduino as ISP"

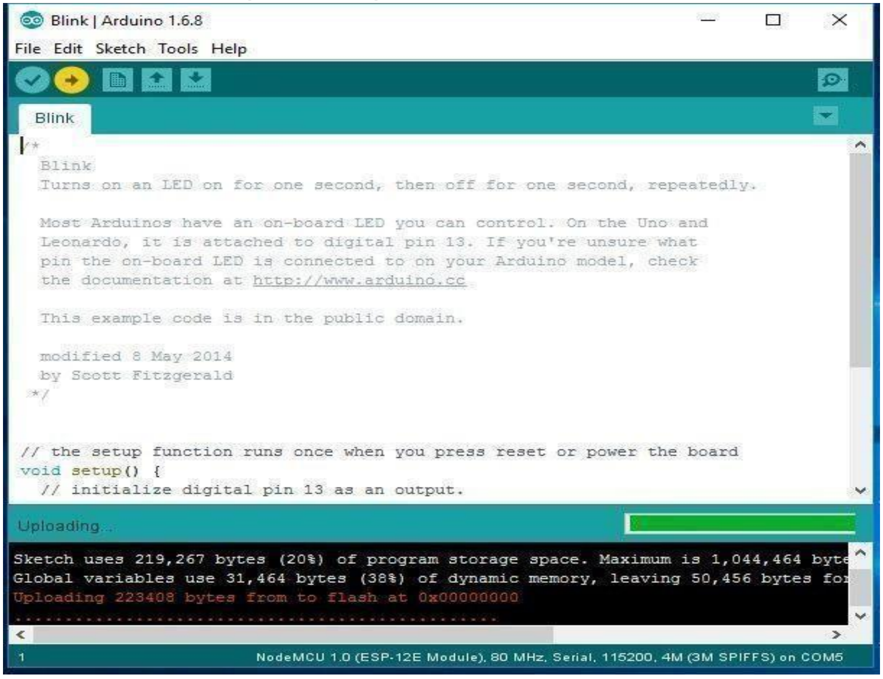
```
/*
  Blink
  Turns on an                              repeatedly.

  Most Arduin                              o and
  Leonardo, i                              what
  pin the on-                              check
  the documen

  This exampl

  modified 8
  by Scott Fi
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
```

```
Global variables use 31,464 bytes (38%) of dynamic memory, leaving 50,456 bytes for
Uploading 223408 bytes from to flash at 0x00000000
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Serial, 115200, 4M (3M SPIFFS) on COM5

The Blink example will open on a new window , click on tools to select the port: “COM” based on which esp8266 module is connected to your respected COM port of the computer. To select COM port refer previous steps.

Step 12: Uploading the Program to ESP8266 Module



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
```

Uploading...

```
Sketch uses 219,267 bytes (20%) of program storage space. Maximum is 1,044,464 byte
Global variables use 31,464 bytes (38%) of dynamic memory, leaving 50,456 bytes for
Uploading 223408 bytes from to flash at 0x00000000
..........................................
```

1 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Serial, 115200, 4M (3M SPIFFS) on COM5

On the blink example code change all number 13 to number 16 and then click on the right arrow shown in the figure to upload the program to the module. This will start blinking the on board led on the nodemcu module.

void setup()

{

// initialize digital pin 16 as an output.pinMode(16, OUTPUT);

}

/ the loop function runs over and over again forevervoid loop()

{

delay(10);

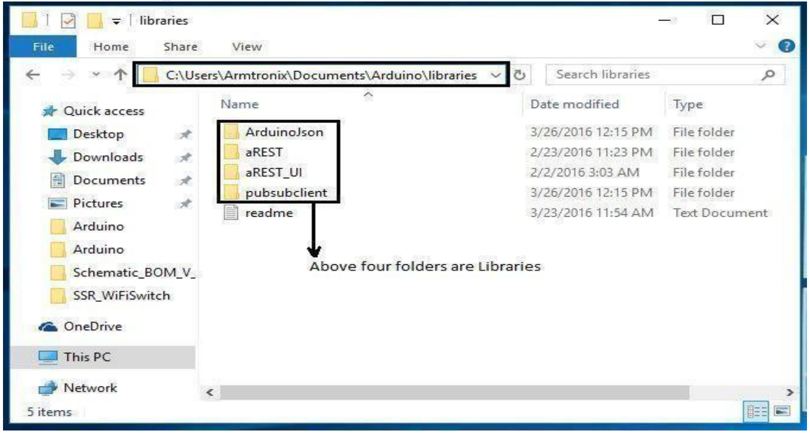
digitalWrite(16, HIGH); // turn the LED on (HIGH is the voltage level)delay(1000); // wait for a second

digitalWrite(16, LOW); // turn the LED off by making the voltage LOWdelay(1000); // wait for a second

}

Step 13: Adding Libraries

In case you need to add the libraries to the Arduino follow the example path is shown in the figure i.e C:\Users\Armtronix\Documents\Arduino\libraries. Enter into the libraries folder then paste the file in thatas shown below



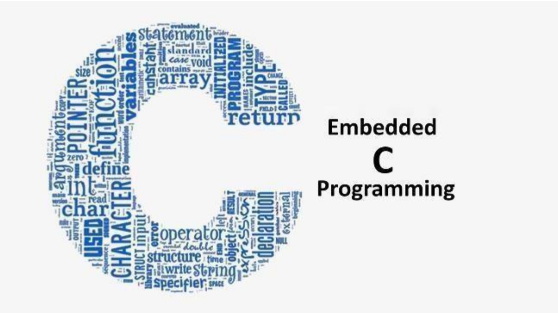
Name	Date modified	Type
ArduinoJson	3/26/2016 12:15 PM	File folder
aREST	2/23/2016 11:23 PM	File folder
aREST_UI	2/2/2016 3:03 AM	File folder
pubsubclient	3/26/2016 12:15 PM	File folder
readme	3/23/2016 11:54 AM	Text Document

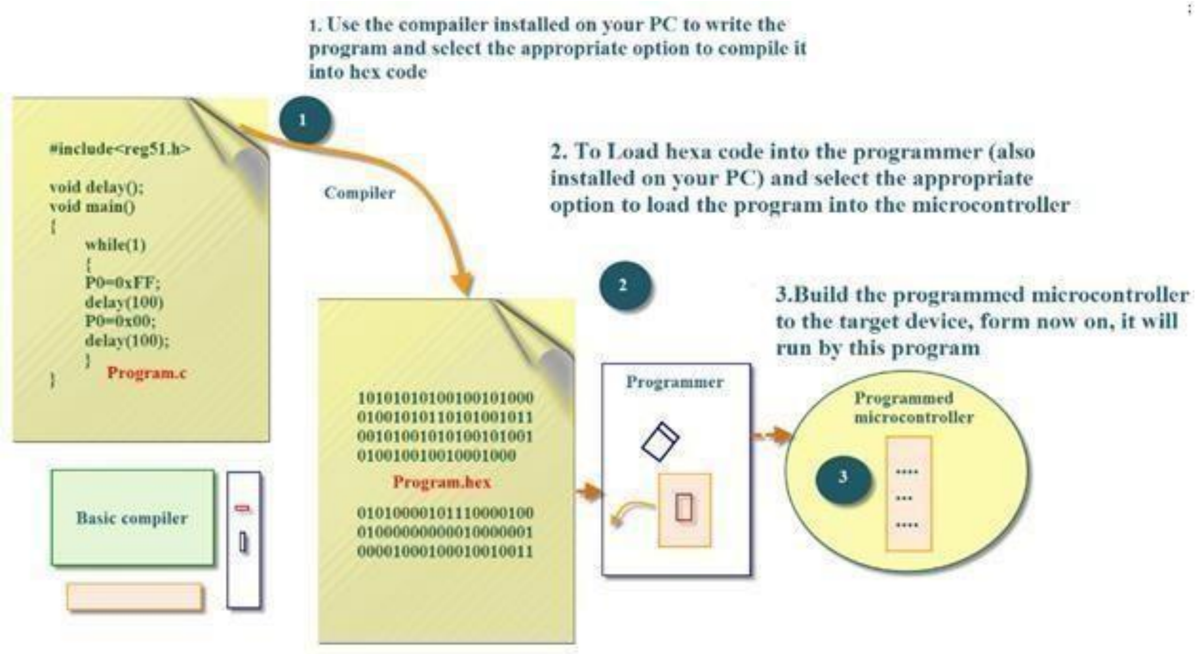
Above four folders are Libraries

4.3 EMBEDDED C

Embedded C is generally used to develop microcontroller-based applications. C is a high-level programming language. Embedded C is just the extension variant of the C language. This programming language is hardware independent.

Embedded C programming plays a key role in performing specific function by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all device working is based on microcontroller that are programmed by embedded C.




1. Use the compailer installed on your PC to write the program and select the appropriate option to compile it into hex code

2. To Load hexa code into the programmer (also installed on your PC) and select the appropriate option to load the program into the microcontroller

3.Build the programmed microcontroller to the target device, form now on, it will run by this program

```
#include<reg51.h>

void delay();
void main()
{
     while(1)
     {
     P0=0xFF;
     delay(100)
     P0=0x00;
     delay(100);
     }
}
```

Program.c

```
10101010100100101000
01001010110101001011
00101001010100101001
010010010010001000
```

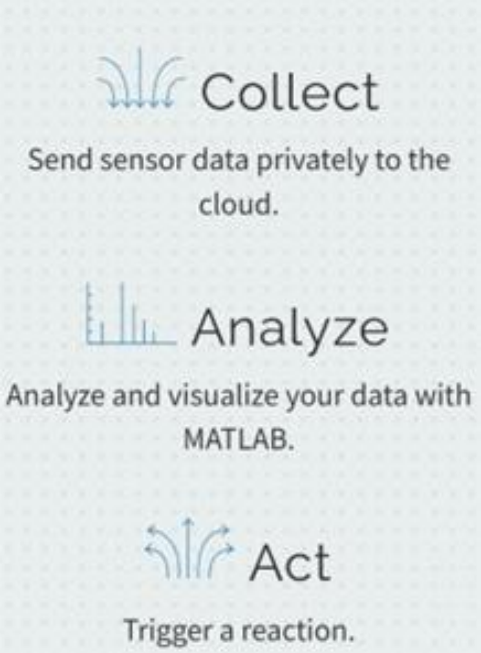
Program.hex

```
01010000101110000100
01000000000010000001
00001000100010010011
```

'Embedded C' is rather embedded target aware programming using traditional 'C' programming language. Embedded C **is a set of language extensions for the C programminglanguage by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems.**

4.4 THINGSPEAK

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts. ThingSpeak is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.



Collect

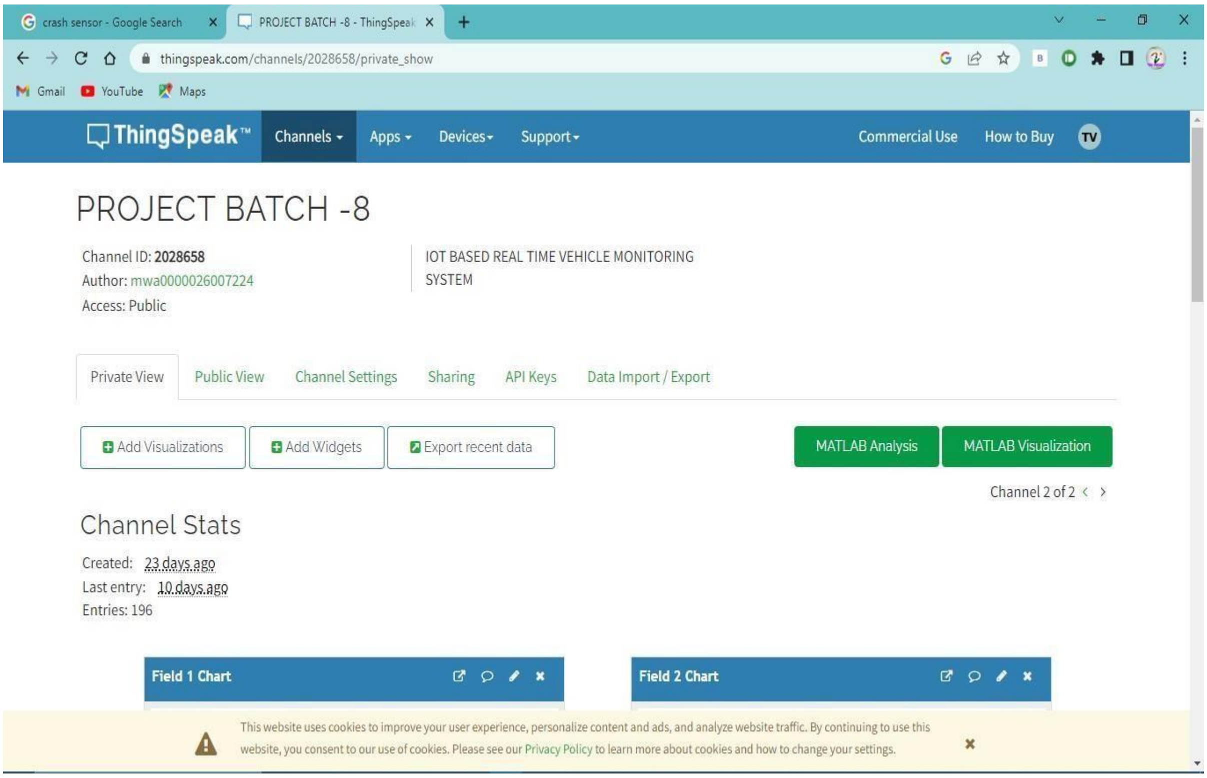
Send sensor data privately to the cloud.

Analyze

Analyze and visualize your data with MATLAB.

Act

Trigger a reaction.



PROJECT BATCH -8

Channel ID: 2028658
Author: mwa0000026007224
Access: Public

IOT BASED REAL TIME VEHICLE MONITORING SYSTEM

Channel Stats

Created: 23 days ago
Last entry: 10 days ago
Entries: 196

This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our Privacy Policy to learn more about cookies and how to change your settings.

Figure4.24: ThingSpeak Channel

The Above Figure4.24 shows the channel representation of my channel named .

4.4 CONTEXT DIAGRAM OF THE PROJECT

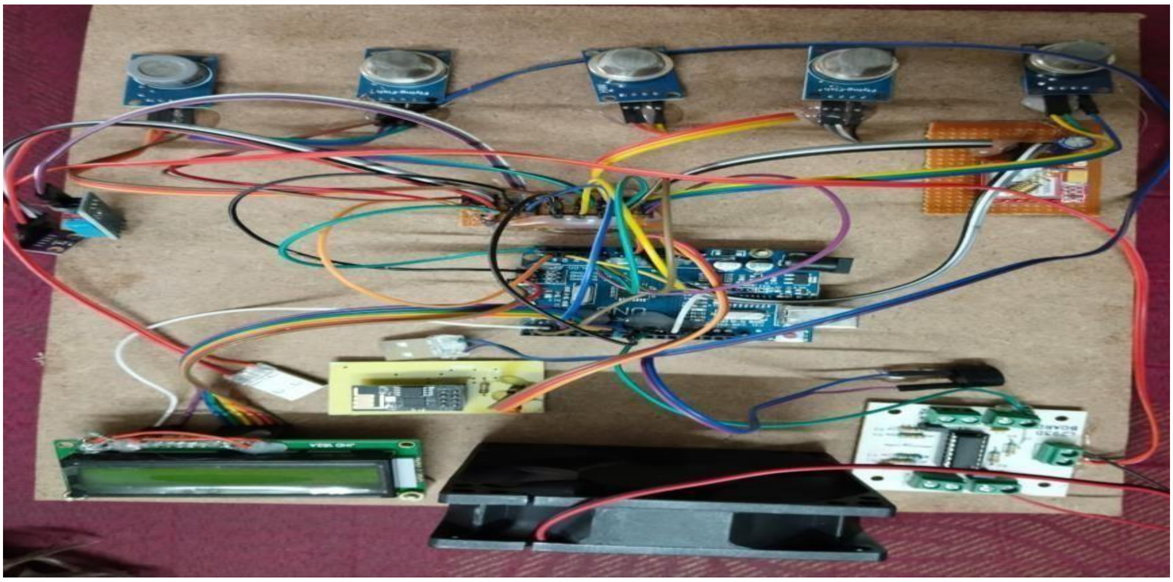


Figure 4.25: Hardware representation of our kit.

CHAPTER–5

DESIGN

5.1 SYSTEM ARCHITECTURE-

The system architecture that can be designed using the components listed includes a control system that continuously monitors the environment for gas levels, temperature, pressure, and humidity. The Arduino Uno will act as the main controller, responsible for executing the code and controlling all the other components of the system.

The system will include several gas sensors, including the MQ -2, MQ-5, MQ-7, MQ-8, and MQ-135 sensors. These sensors will detect the presence of different types of gases, such as combustible gases, carbon monoxide, and ammonia. The MQ-2 sensor is commonly used to detect smoke, methane, propane, and butane. The MQ-5 sensor is used to detect natural gas, propane, and hydrogen. The MQ -7 sensor is used to detect carbon monoxide and the MQ-8 sensor is used to detect hydrogen gas. The MQ-135 sensor is used to detect air quality including benzene, alcohol, smoke, and nitrogen oxides. The BMP 180 sensor will be used to measure the temperature and pressure of the environment, while the DHT11 sensor will measure the humidity. These sensors will provide additional information about the environment and can be used to detect changes in weather conditions or other factors that could impact the gas levels.

The GSM module will enable the system to communicate through the mobile network, while the WiFi module will allow the system to connect to a local wireless network and communicate through the internet. This will enable the system to send alerts to the user or a remote monitoring station if dangerous levels of gas are detected. The LCD display will be used to display the sensor data and system status.

In the event that a dangerous level of gas is detected, the system will sound an alarm using the buzzer. The exhaust fan will be used to ventilate the environment and reduce the concentration of gas if necessary. The system will continuously monitor the environment and provide real-time feedback to the user through the LCD display and mobile or internet communication.

All the components will be connected using connecting wires, and the system will be powered using an external power source. The code for the system will be written in Arduino programming language, and it will include logic to control the sensors, display data on the LCD display, sound the alarm, and control the exhaust fan.

Overall, this system architecture provides a comprehensive approach to monitoring the environment for gas levels and ensuring the safety of the occupants. The system can be customized to meet specific needs and can be expanded to include additional sensors or other components as necessary.

5.2 CONTROLLER SECTION

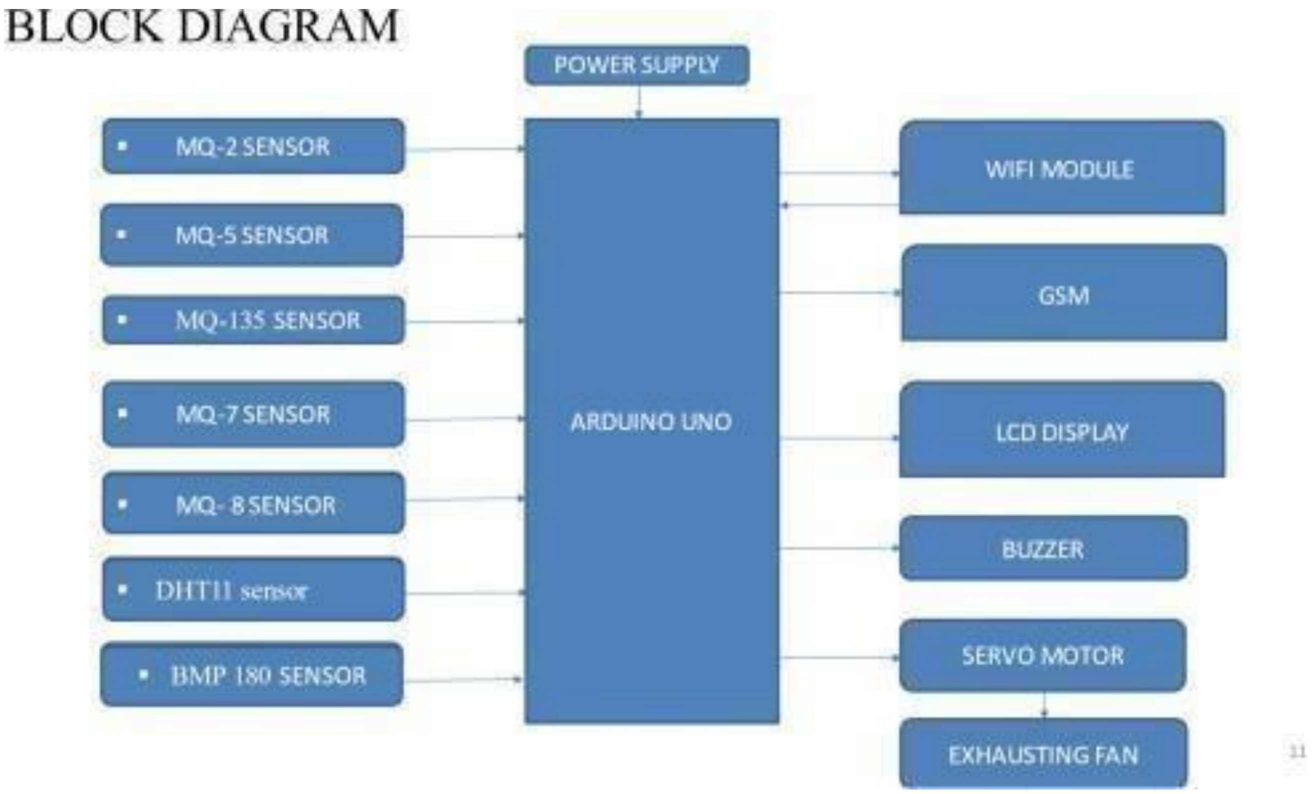


FIG. 5.1 BLOCK DIAGRAM

The project's block diagram shows that the system includes sensors, a microcontroller board, a GSM module, an LCD display, a Wi-Fi module, a buzzer, and an exhaust fan. The sensors measure the air quality and temperature, and the microcontroller board processes and controls the system based on the sensor data. If the air quality is hazardous, the GSM module sends SMS alerts to the user, and the Wi-Fi module uploads sensor data to Thing Speak. The LCD display displays sensor readings and system status, whereas the Android app displays sensor data graphically and allows the user to set alert thresholds and receive alerts. The exhaust fan expels hazardous gases from the enclosed space. Overall, the project is a comprehensive air quality monitoring system that provides the user with real-time data and alerts. By detecting hazardous gases and expelling them from the enclosed space, the system ensures a safe and healthy environment. The Android app allows the user to access sensor data, view graphs, and set alert thresholds, making it simple to maintain the system and ensure accurate readings.

5.3 CLOUD SECTION:

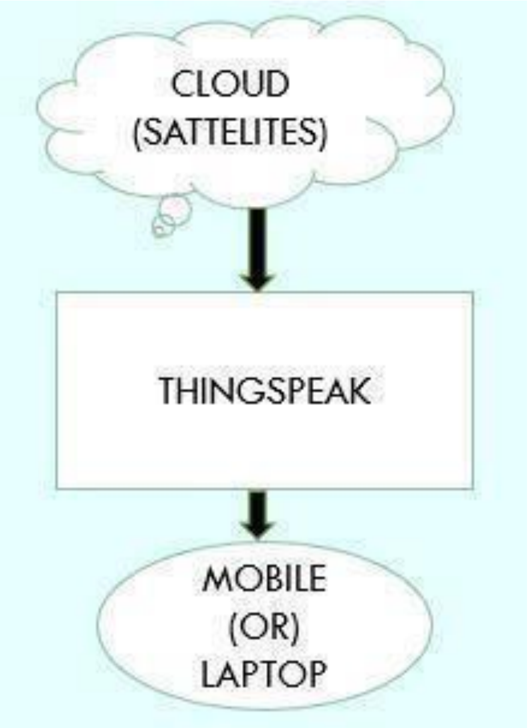


FIG. 5.2: BLOCK DIAGRAM OF CLOUD SECTION

In Cloud section as shown in Figure4.26. we have satellite which receives the information about co-ordinates and Vehicle information through the SIM800A GSM GPRS module which has the SIM (Subscriber identity Module) in it 2G. By using this gsm the information is transmitted to the satellite. We can collect this data transmitted to the satellite from vehicle using the application called Thingspeak. Opening the application we can see the data in our mobile/pc in the form of graphical representation. ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.

5.4 OPERATION

The project is intended to monitor the air quality and temperature of a closed space and to notify the user if the air quality is hazardous. The system measures air quality and temperature using various sensors, and a microcontroller board processes the data and controls the system. Sensor Reading

The sensors are linked to the microcontroller board and continuously measure the enclosed space's air quality and temperature. The microcontroller board then processes the sensor readings. Data Processing

The sensor data is processed by the microcontroller board, which determines whether the air quality is hazardous or not. If the air quality is hazardous, the system activates the buzzer to sound an alarm and sends an SMS alert to the user via the GSM module. The system also activates the exhaust fan to expel the hazardous gases from the enclosed space. The LCD display displays sensor readings as well as system status. The user can view sensor data and system status on the LCD display.

The Wi-Fi module sends sensor data to Thing Speak, which is accessible via the Android app. The app allows the user to view sensor data, set alert thresholds, and receive alerts. The Android app allows the user to access sensor data, view graphs, and set alert thresholds. If the air quality is hazardous, the user can also receive SMS alerts.

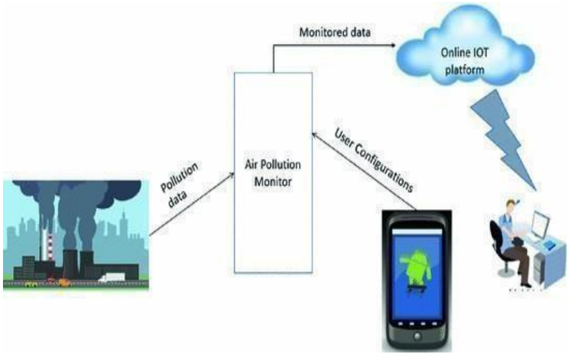


FIG. 5.3 OPERATION

The system should be maintained on a regular basis by checking the sensors and their connections. To ensure, accurate readings, the system should be cleaned on a regular basis. Overall, this system is intended to monitor the air quality and temperature of an enclosed space, providing the user with real -time data and alerts. Overall, the goal of this project is to monitor the air quality and temperature of an enclosed space and notify the user if the air quality is hazardous. The system provides the user with real-time data and alerts, ensuring a safe and healthy environment. The system is intended to monitor the air quality and temperature of a closed space and to notify the user if the air quality is hazardous. A microcontroller board, a GSM module, an LCD display, a Wi -Fi module, a buzzer, and an exhaust fan comprise the system.

Sensors: Several sensors are used in the system to monitor the air quality and temperature of the space.

Among these sensors are:

MQ-2 gas detector for smoke, methane, propane, and other gases. MQ-5 sensor for natural gas, propane, and

hydrogen detection

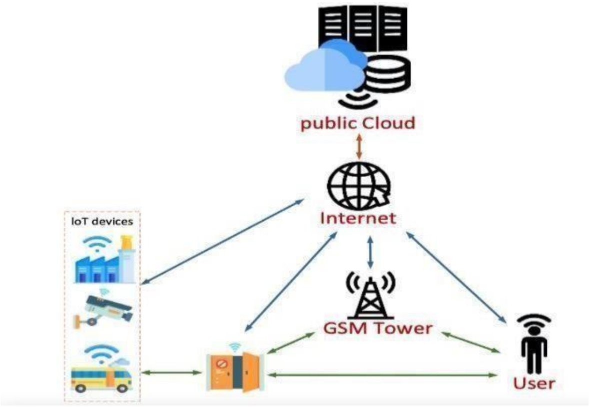


FIG. 5.4 SOURCES RECIVED FROM AIR POLLUTION

MQ-7 sensor for carbon monoxide detection, MQ-135 air quality sensor detects hazardous gases such as ammonia, nitrogen oxides, and sulphur dioxide. MQ-8 sensor for hydrogen gas detection. DHT11 sensor for temperature and humidity measurement. BMP180 sensor for pressure, altitude, and temperature measurement. The Arduino Uno board with Atmega328p serves as the main microcontroller board, processing data from sensors and controlling the system.

The GSM module is used to send SMS alerts to the user if any hazardous gases are detected.

LCD Display: The LCD display is used to show sensor readings and system status.

The Wi-Fi module is used to upload sensor readings to Thing Speak, which is accessible via an Android app.

Buzzer: A buzzer is used to sound an alarm if dangerous gases are detected.

Exhaust Fan: The exhaust fan expels hazardous gases from the enclosed space.

Software Requirements: The Arduino IDE and Embedded C are used to programming the system. To write, compile, and upload code to the microcontroller board, the Arduino IDE is used. The Embedded C programming language is used to programming the microcontroller board in order to read sensor data, process it, and control the system.

Thing Speak Android App: The Thing Speak Android app allows you to remotely view sensor data and system status. The user can view the graphs and set the alert thresholds after accessing the sensor data.

CHAPTER -6

RESULTS

6.1 Key function of Results

An Air pollution monitoring based on IoT system with

GSM and LCD display can provide real-time information on the air quality of a particular area. The system consists of sensors that can detect various pollutants such as carbon monoxide, sulfur dioxide, nitrogen dioxide, and particulate matter. These sensors are connected to a microcontroller board such as Arduino which collects the data and sends it to a cloud-based platform.

The cloud-based platform processes the data and generates a report on the air quality of the area. This report can be accessed through a mobile application. Additionally, the system can also send alerts to the concerned authorities or individuals via GSM, in case of dangerous levels of pollutants.

The LCD display can be used to display the air quality index (AQI) of the area in real-time. The AQI provides an indication of the air quality and the associated health effects. The display can also show the concentration of individual pollutants and their associated limits

6.2 CODING

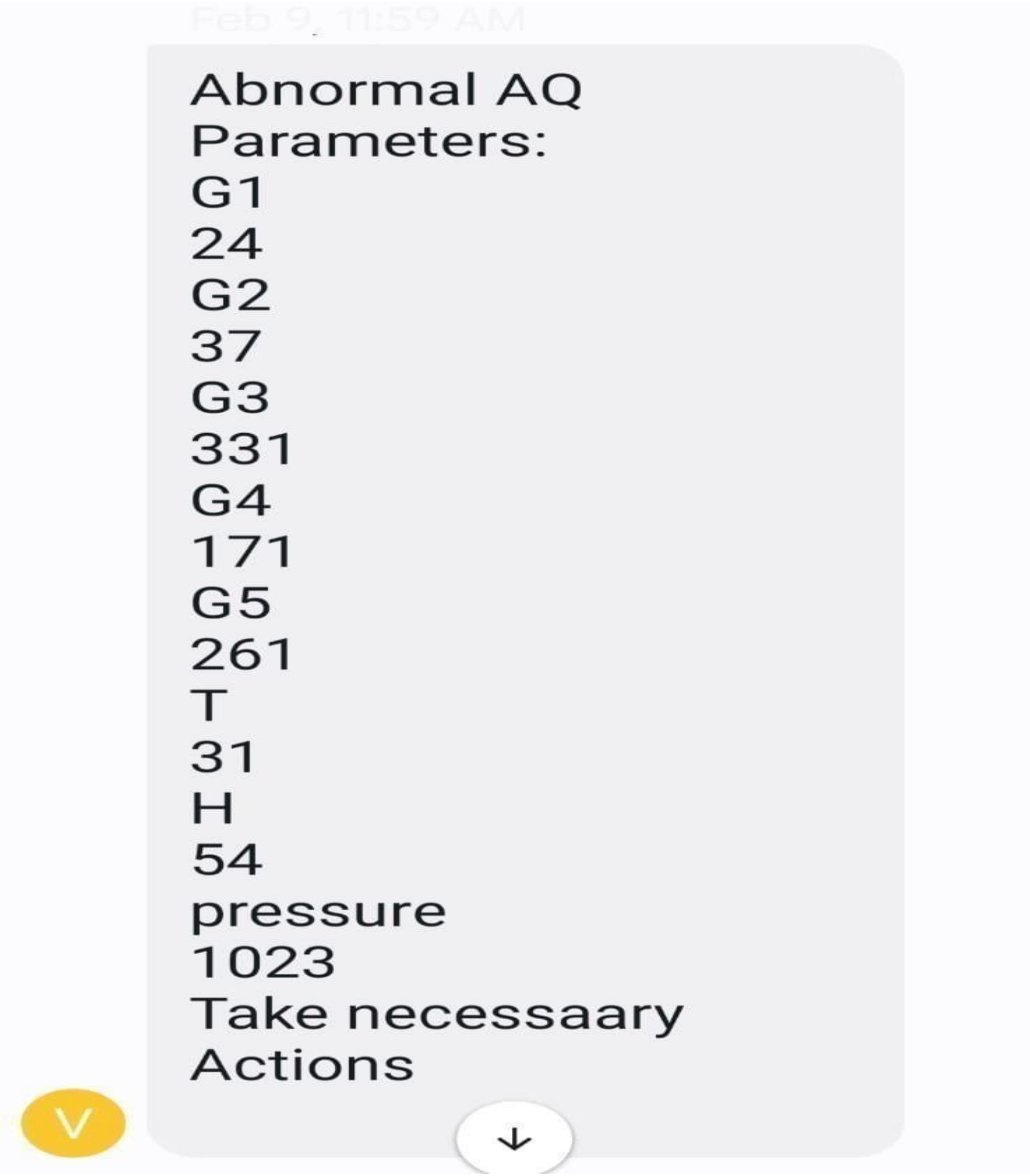
By using embedded C language

```
#include<SoftwareSerial.h>
SoftwareSerial gsm_Serial(11,12);
#include "DHT.h"
#define DHTPIN 8
#define DHTTYPE DHT11
DHT dht(DHTPIN,DHTTYPE);
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_BMP280.h>
//#define BMP_SCK (13)
//#define BMP_MISO (12)
//#define BMP_MOSI (11)
//#define BMP_CS (10)
Adafruit_BMP280 bmp; // I2C
//Adafruit_BMP280 bmp(BMP_CS); // hardware SPI
//Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK); #include
<LiquidCrystal.h>
const int rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7; LiquidCrystal lcd(rs, en, d4, d5, d6,
d7); int mq2=A0;
int mq135=A1;
int mq5=A2;
int mq6=A3;
int  mq7=10;
int buz=13;
int ef=10;
int t,h,s1,s2,s3,s4,s5;
void setup()
{
dht.begin();
pinMode(ef,OUTPUT);
pinMode(buz,OUTPUT);
pinMode(mq2,INPUT);
pinMode(mq135,INPUT);
pinMode(mq5,INPUT);
pinMode(mq6,INPUT);
pinMode(mq7,INPUT);
pinMode(buz,OUTPUT);
Serial.begin(115200);
gsm_Serial.begin(9600);
gsm_Serial.println("AT");
delay(1500);
gsm_Serial.println("AT+CMGF=1");
lcd.begin(16, 2);
lcd.print(" WELCOME");
delay(1500);
```

```
    pinMode(buz,OUTPUT);
    wifi_init();
    digitalWrite(buz,0);
    //delay(2000);
  //  Serial.println(F("BMP280 test"));
   // if (!bmp.begin()) {
//    Serial.println(F("Could not find a valid BMP280 sensor, check wiring!"));
//    while (1);
   // }
   // /* Default settings from datasheet. */
    // bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,     /* Operating Mode. */
    //   Adafruit_BMP280::SAMPLING_X2,     /* Temp. oversampling */
    //   Adafruit_BMP280::SAMPLING_X16,    /* Pressure oversampling */
    //   Adafruit_BMP280::FILTER_X16,      /* Filtering. */
    //   Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */
    }
    void loop()
    {
    delay(200);
    lcd.clear();
    s1=analogRead(mq2);
    s2=analogRead(mq135);
    s3=analogRead(mq5);
    s4=analogRead(mq6);
    s5=analogRead(mq7);
    t = dht.readTemperature();
    h=dht.readHumidity();
     int pval = bmp.readPressure();
   //  Serial.print(F("Pressure = "));
  //  Serial.print(bmp.readPressure());
  //  Serial.println(" Pa");
  // Serial.print(F("Approx altitude = "));
  //  Serial.print(bmp.readAltitude(1013.25)); /* Adjusted to local forecast! */
   // Serial.println(" m");
    lcd.setCursor(0,0);
    lcd.print("S1:");
    lcd.setCursor(3,0);
    lcd.print(s1);
    lcd.setCursor(6,0);
    lcd.print("T:");
    lcd.setCursor(8,0);
    lcd.print(t);
    lcd.setCursor(11,0);
    lcd.print("H:");
    lcd.setCursor(13,0);
    lcd.print(h);
    lcd.setCursor(0,1);
    lcd.print("S2:");
    lcd.setCursor(3,1);
lcd.print(s2);
lcd.setCursor(6,1);
lcd.print("S:");
lcd.setCursor(8,1);
lcd.print(s3);
lcd.setCursor(11,1);
lcd.print("G:");
lcd.setCursor(13,1);
lcd.print(s4);
if((s1>300)||(s2>300)||(s3>500)||(s4>500)||(s5>500)||(t>40)||(h>90))
{
digitalWrite(buz,1);
digitalWrite(ef,1);
send_sms();
upload_iot(s1,s2,s3,s4,s5,t,h,1023);
digitalWrite(buz,0);
}
else
{
digitalWrite(buz,0);
digitalWrite(ef,0);
}

}
void send_sms()
{
gsm_Serial.println("Sending SMS...");
gsm_Serial.println("AT");
delay(1000);
gsm_Serial.println("ATE0");
delay(1000);
gsm_Serial.println("AT+CMGF=1");
delay(1000);
gsm_Serial.print("AT+CMGS=\"6301578276\"\r\n");// Replace x with mobile number
delay(1000);
//if(sts==1)
gsm_Serial.println("Abnormal AQ Parameters: ");
gsm_Serial.println("G1");
gsm_Serial.println(s1);
gsm_Serial.println("G2");
gsm_Serial.println(s2);
gsm_Serial.println("G3");
gsm_Serial.println(s3);
gsm_Serial.println("G4");
gsm_Serial.println(s4);
gsm_Serial.println("G5");
gsm_Serial.println(s5);
gsm_Serial.println("T");
gsm_Serial.println(t);
gsm_Serial.println("H");
gsm_Serial.println(h);
gsm_Serial.println("pressure");
            gsm_Serial.println("1023");
            gsm_Serial.println("Take necessaary Actions");
            delay(100);
            gsm_Serial.println((char)26);// ASCII code of CTRL+Z delay(2000);
            gsm_Serial.println("AT");
            delay(1000);
            gsm_Serial.println("ATE0");
            delay(1000);
            gsm_Serial.println("AT+CMGF=1");
            delay(1000);
            gsm_Serial.print("AT+CMGS=\"9347130040\"\r\n");// Replace x with mobile number
            delay(1000);
            //if(sts==1)
            gsm_Serial.println("Abnormal AQ Parameters: ");
            gsm_Serial.println("G1");
            gsm_Serial.println(s1);
            gsm_Serial.println("G2");
            gsm_Serial.println(s2);
            gsm_Serial.println("G3");
            gsm_Serial.println(s3);
            gsm_Serial.println("G4");
            gsm_Serial.println(s4);
            gsm_Serial.println("G5");
            gsm_Serial.println(s5);
            gsm_Serial.println("T");
            gsm_Serial.println(t);
            gsm_Serial.println("H");
            gsm_Serial.println(h);
            gsm_Serial.println("pressure");
            gsm_Serial.println("1023");
            gsm_Serial.println("Take necessaary Actions");
            delay(100);
            gsm_Serial.println((char)26);// ASCII code of CTRL+Z delay(2000);
            }

            void wifi_init()
            {
            Serial.println("AT+RST");
            delay(2000);
            Serial.println("AT+CWMODE=1");
            delay(1000);
            Serial.print("AT+CWJAP=");
            Serial.write("");
            Serial.print("mahesh"); // ssid/user name
            Serial.write("");
            Serial.write(',');
            Serial.write("");
  Serial.print("12345678"); //password
Serial.write("");
            Serial.println();
            delay(1000);
            }
            void upload_iot(int x,int y,int z,int p,int q,int r,int s,int t) //ldr copied int to - x and gas copied into -
            y
            {

            String cmd = "AT+CIPSTART=\"TCP\",\"";
            cmd += "184.106.153.149"; // api.thingspeak.com
            cmd += "\",80";
            Serial.println(cmd);
            delay(1500);
           String getStr ="GET /update?api_key=LNNOFC6JGB5HNAU9&field1=";
            getStr += String(x);
            getStr +="&field2=";
            getStr += String(y);
            getStr +="&field3=";
            getStr += String(z);
            getStr +="&field4=";
            getStr += String(p);
            getStr +="&field5=";
            getStr += String(q);
            getStr +="&field6=";
            getStr += String(r);
            getStr +="&field7=";
            getStr += String(s);
            getStr +="&field8=";
            getStr += String(t);
            getStr += "\r\n\r\n";
            cmd = "AT+CIPSEND=";
            cmd += String(getStr.length());
            Serial.println(cmd);
            delay(1500);
            Serial.println(getStr);
            delay(1500);
            }
```

6.3 OUTPUT SCREENS

Notification: Alerts from the system is shown in below fig 6,1

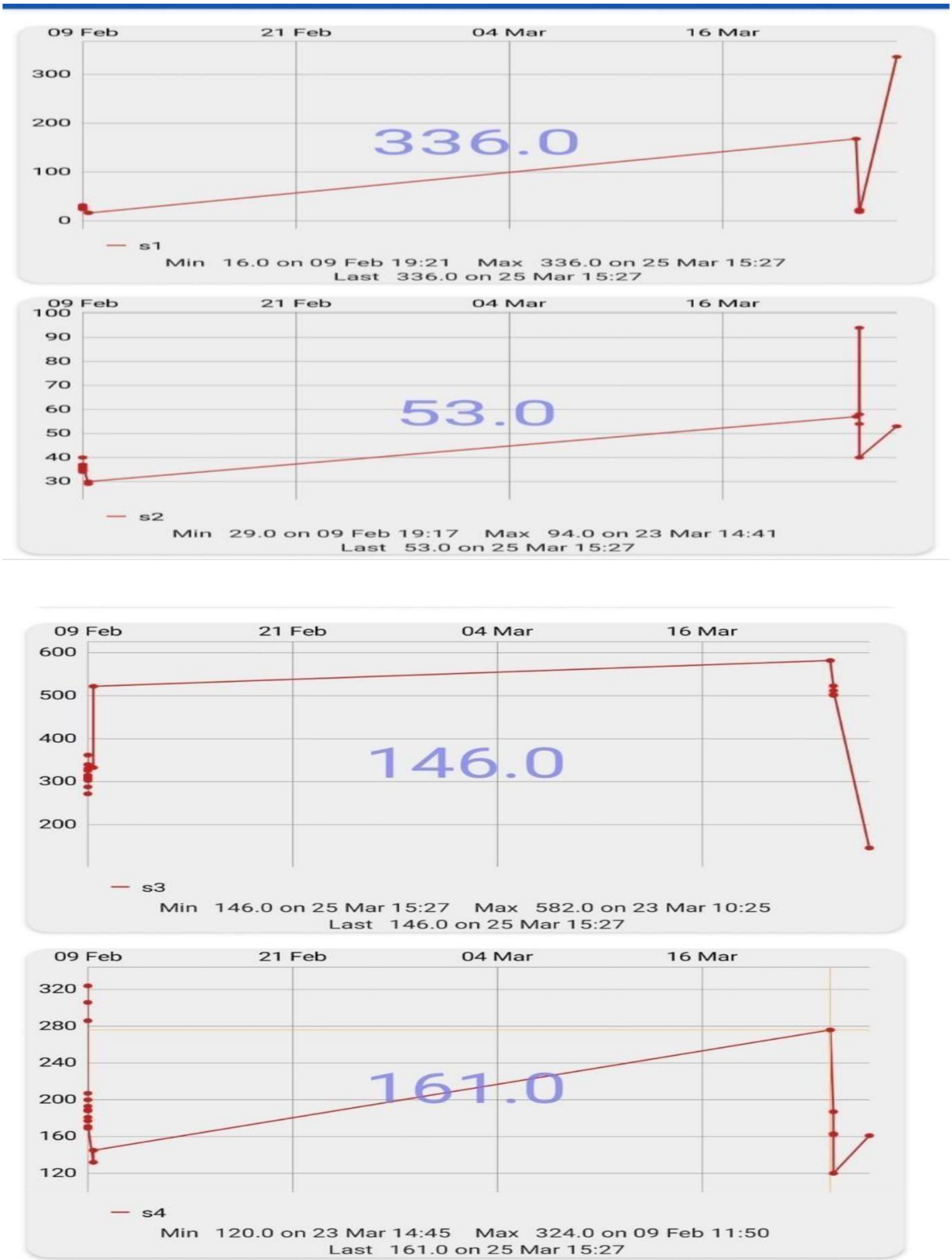


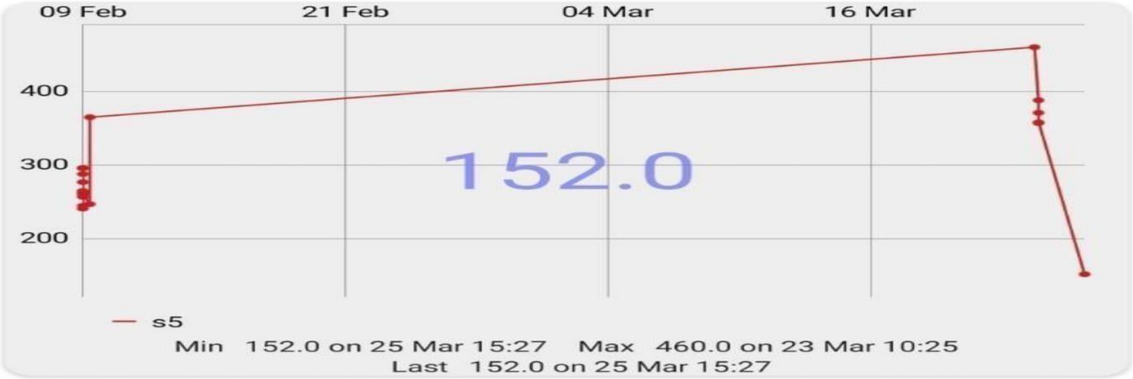
Feb 9, 11:59 AM

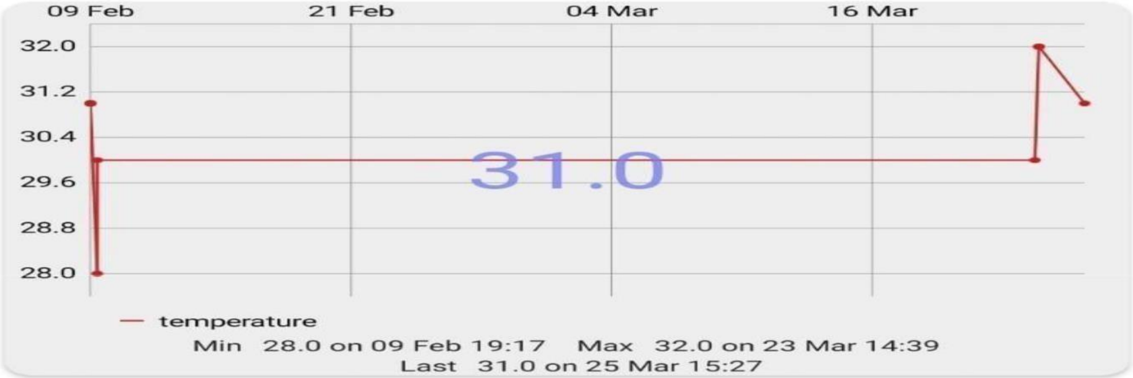
Abnormal AQ
Parameters:
G1
24
G2
37
G3
331
G4
171
G5
261
T
31
H
54
pressure
1023
Take necessaary
Actions

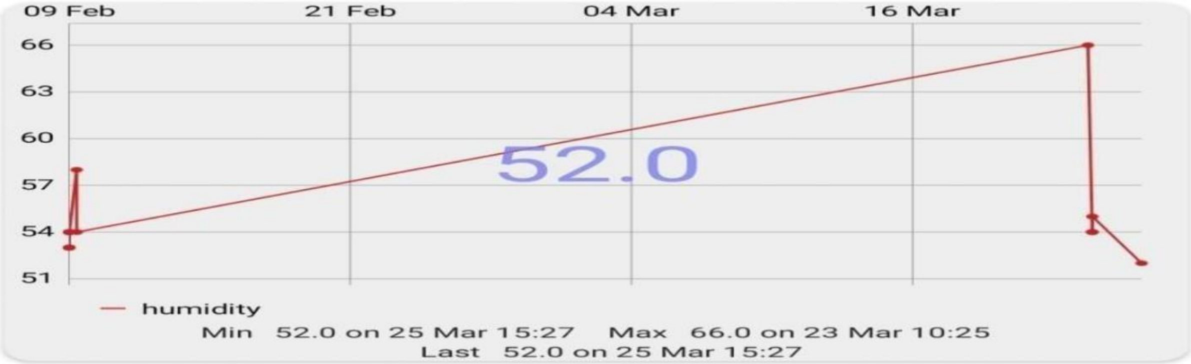
Fig 6,1 Screenshot of the Text message received using GSM

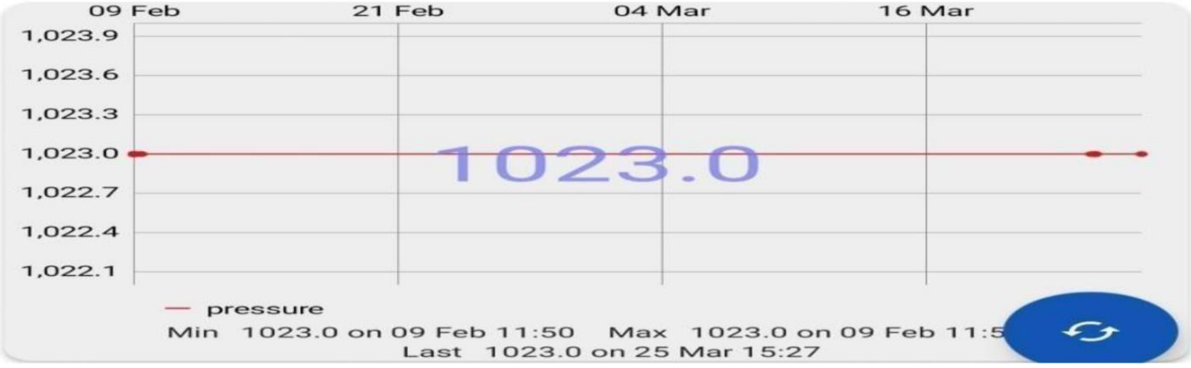
Fig 6.2 outputs on the IoT platform called Things Speak




09 Feb
21 Feb
04 Mar
16 Mar
400
300
200
152.0
s5
Min 152.0 on 25 Mar 15:27 Max 460.0 on 23 Mar 10:25
Last 152.0 on 25 Mar 15:27


09 Feb
21 Feb
04 Mar
16 Mar
32.0
31.2
30.4
29.6
28.8
28.0
31.0
temperature
Min 28.0 on 09 Feb 19:17 Max 32.0 on 23 Mar 14:39
Last 31.0 on 25 Mar 15:27


09 Feb
21 Feb
04 Mar
16 Mar
66
63
60
57
54
51
52.0
humidity
Min 52.0 on 25 Mar 15:27 Max 66.0 on 23 Mar 10:25
Last 52.0 on 25 Mar 15:27


09 Feb
21 Feb
04 Mar
16 Mar
1,023.9
1,023.6
1,023.3
1,023.0
1,022.7
1,022.4
1,022.1
1023.0
pressure
Min 1023.0 on 09 Feb 11:50 Max 1023.0 on 09 Feb 11:5
Last 1023.0 on 25 Mar 15:27

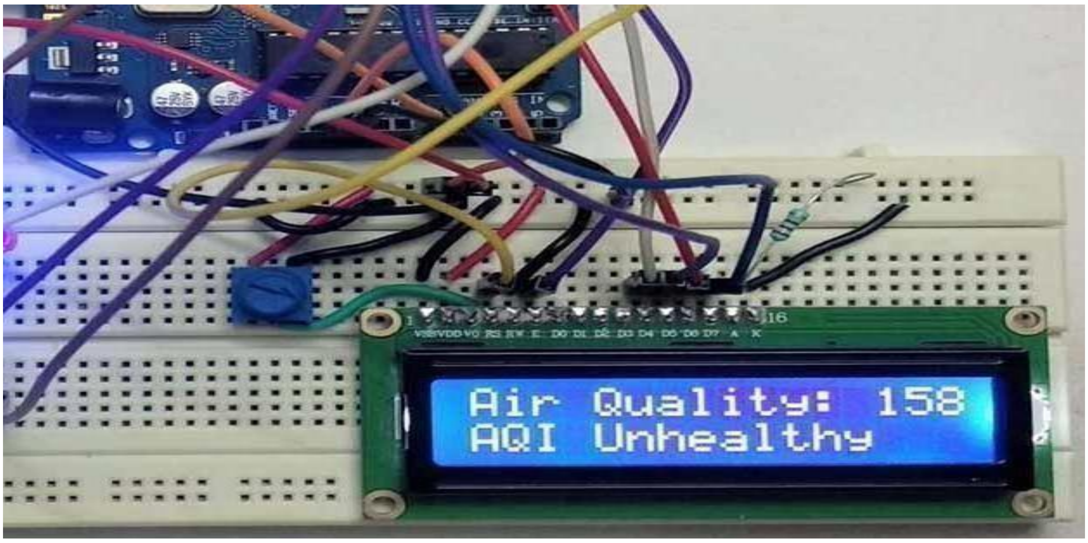


Fig 6.3 outputs on the LCD Display

An Air Pollution Monitoring Based on IoT system with GSM and LCD display can provide real-time information on the air quality of a particular area. The system consists of sensors that can detect various pollutants such as carbon monoxide, sulfur dioxide, nitrogen dioxide, and particulate matter. These sensors are connected to a microcontroller board such as Arduino which collects the data and sends it to a cloud-based platform.

The cloud-based platform processes the data and generates a report on the air quality of the area. This report can be accessed through a mobile application. Additionally, the system can also send alerts to the concerned authorities or individuals via GSM, in case of dangerous levels of pollutants.

The LCD display can be used to display the air quality index (AQI) of the area in real-time. The AQI provides an indication of the air quality and the associated health effects. The display can also show the concentration of individual pollutants and their associated limits.

IoT-based air pollution monitoring systems offer numerous advantages over traditional monitoring methods. Here are some of the advantages and applications of Air Pollution Monitoring Based on IoT

Real-time monitoring: IoT-based air pollution monitoring systems provide real-time data on air quality, allowing for quick response times and improved decision-making.

Continuous monitoring: IoT-based air pollution monitoring systems can continuously monitor air quality, providing a more comprehensive understanding of pollution levels over time.

Remote monitoring: IoT-based air pollution monitoring systems can be monitored remotely, reducing the need for manual data collection and improving efficiency.

Cost-effective: IoT-based air pollution monitoring systems can be less expensive than traditional monitoring methods, making them more accessible to communities and governments.

Scalability: IoT-based air pollution monitoring systems can be easily scaled to cover larger areas or more pollutants.

Applications:

Environmental monitoring: IoT-based air pollution monitoring systems can be used to monitor air quality in urban and rural areas, as well as in industrial and commercial settings.

Health monitoring: IoT-based air pollution monitoring systems can be used to monitor air quality in hospitals and healthcare facilities to ensure safe and healthy environments for patients and staff.

Traffic monitoring: IoT-based air pollution monitoring systems can be used to monitor air quality in areas with high traffic, such as highways and urban centers.

Agricultural monitoring: IoT-based air pollution monitoring systems can be used to monitor air quality in agricultural settings to ensure crop health and worker safety.

Disaster management: IoT-based air pollution monitoring systems can be used to monitor air quality in areas affected by natural disasters, such as wildfires, floods, and hurricanes.

Overall, IoT-based air pollution monitoring systems offer a powerful tool for monitoring and managing air quality, with applications in a wide range of industries and settings.

CHAPTER - 7

FUTURE SCOPE

FUTURE SCOPE

The future scope of An Air pollution monitoring based on IoT

systems is vast and has the potential to be a game-changer in the field of environmental monitoring. Some potential advancements that could be made in an advanced version of the system include

Integration with smart city infrastructure: An Air pollution monitoring based on IoT system can be integrated with other smart city infrastructure such as traffic management systems, weather stations, and public transport systems. This can provide more accurate and real-time data on air quality and enable authorities to take preventive measures more effectively

Multi-sensor integration: An advanced version of the system can integrate multiple sensors that can detect a wide range of pollutants. This can provide a more comprehensive picture of air quality in the area. Integration with AI: An Air pollution monitoring based on IoT system integrated with AI technology can improve the accuracy and reliability of the system. The system can learn to identify patterns in air pollution data and provide predictions on future pollution levels

CHAPTER-8

CONCLUSION

CONCLUSION

Air pollution monitoring based on IoT have revolutionized the way we monitor andmanage air quality. By leveraging the power of the Internet of Things (IoT), these systems offer real-time, continuous, and remote monitoring of air quality, making them more effective, efficient, and accessible than traditional monitoring methods.

Air pollution monitoring based on IoT provide numerous advantages over traditional monitoring methods, such as real-time monitoring, continuous monitoring, remote monitoring, cost-effectiveness, and scalability. These advantages make IoT-based air pollution monitoring systems more suitable for use in a wide range of industries and settings, including environmental monitoring, health monitoring, traffic monitoring, agricultural monitoring, and disaster management.

Air pollution monitoring based on IoT are also capable of detecting a wide range of pollutants, including particulate matter, volatile organic compounds, carbon monoxide, and nitrogen oxides. This makes them useful in identifying the sources of pollution and developing targeted interventions to reduce pollution levels.

Moreover, the data collected by IoT-based air pollution monitoring systems can be used to inform policy and decision-making at the local, national, and global levels, helping to improve air quality and protect public health.

In summary, air pollution monitoring based on IoT systems have the potential to transform the way we monitor and manage air quality, providing a powerful tool for identifying, tracking, and mitigating air pollution in a wide range of industries and settings. With the continued development of IoT technology and the increasing demand for better air quality, we can expect to see the adoption of air pollution monitoring based on IoT continue to grow in the coming years.

REFERENCES

[1] Manglani T., Srivastava A., Kumar A. and Sharma R. IoT based Air and Sound Pollution Monitoring System for Smart Environment. 2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022, pp. 604-607.

[2] Chen H. Z., Cheng S. G., Chen Y. Q., Du D. and Feng X. X. Spatial and temporal distribution Characteristic of China acid rain and its harm. 2011 International Conference on Multimedia Technology, 2011, pp. 1228-1230.

[3] Abas A. M., Azim K. A., Fadzil H. M. and Dahlui M. Monitoring the quality of ozone towards the prevention of further global warming. 2011 2nd International Conference on Instrumentation Control and Automation, 2011, pp. 194-199.

[4] Biondi S. M., Catania V., Monteleone S. and Polito C. Bus as a sensor: A mobile sensor nodes network for the air quality monitoring. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2017, pp. 272-277.

[5] Pramanik P., Nandi S. and Saha M. AirCalypse: Revealing Fine-Grained Air Quality from Social Media. 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2018, pp. 1507-1508.

[6] Marjovi A., ArfireA. and Martinoli A. High Resolution Air Pollution Maps in Urban Environments Using Mobile Sensor Networks. 2015 International Conference on Distributed Computing in Sensor Systems, 2015, pp. 11-20.

[7] Chen Y., Chen D., Song T. and Song K. An Intelligent and Portable Air Pollution Monitoring System Based on Chemical Sensor Array. 2020 IEEE 4th International Conference on Frontiers of Sensors Technologies (ICFST), 2020, pp. 21-25.

Science, Technology and Development ISSN : 0950-0707

IoT based Air Pollution Monitoring

KESAM MAHESH[1] V. VAMSI KRISHNA[2] SHAIK NISHITHA[3] V. AJAY[4]
Electronics And Communication Engineering,
Lingayas Institute of Management And Technology Vijayawada, Andhra Pradesh, India.
maheshkesam@gmail.com, 19na1a0422.lingayas@limat.edu.in, 19na1a0417.lingayas@limat.edu.in, 19na1a0433.lingayas@limat.edu.in,

Author Corresponding *
P.SOMYA KEERTHANA, M.Tech.
Assistant Professor, Dept. of Electronics And Communication Engineering, Lingayas Institute of Management And Technology Vijayawada, Andhra Pradesh, India.
Keerthana.lingayas@limat.edu.in.

***Abstract*— In recent years, the Internet of Things (IoT) has emerged as a significant area of research and development. It opens up a plethora of possibilities for developing smart and connected devices capable of sensing, monitoring, and controlling various aspects of the physical world. Monitoring air pollution levels in urban areas is one of the critical applications of IoT. This paper proposes an IoT-based air pollution monitoring system that measures the concentration of various air pollutants using low-cost sensors. The system combines multiple sensors with an IoT gateway, which collects and processes sensor data. The information gathered is then sent to a cloud-based server for storage, analysis, and visualization. The proposed system monitors air pollution levels in real time and serves as a platform for environmental monitoring and management. Because of its low cost and ease of deployment, the system is an appealing solution for monitoring air pollution in urban areas, which is critical for public health and environmental sustainability. The proposed system can also be used for other types of environmental monitoring, such as water quality monitoring, noise level monitoring, and weather monitoring.**

Keywords*— *Arduino, Microcontroller, Hardware, Software, Air Pollution Monitoring, GSM, IoT.

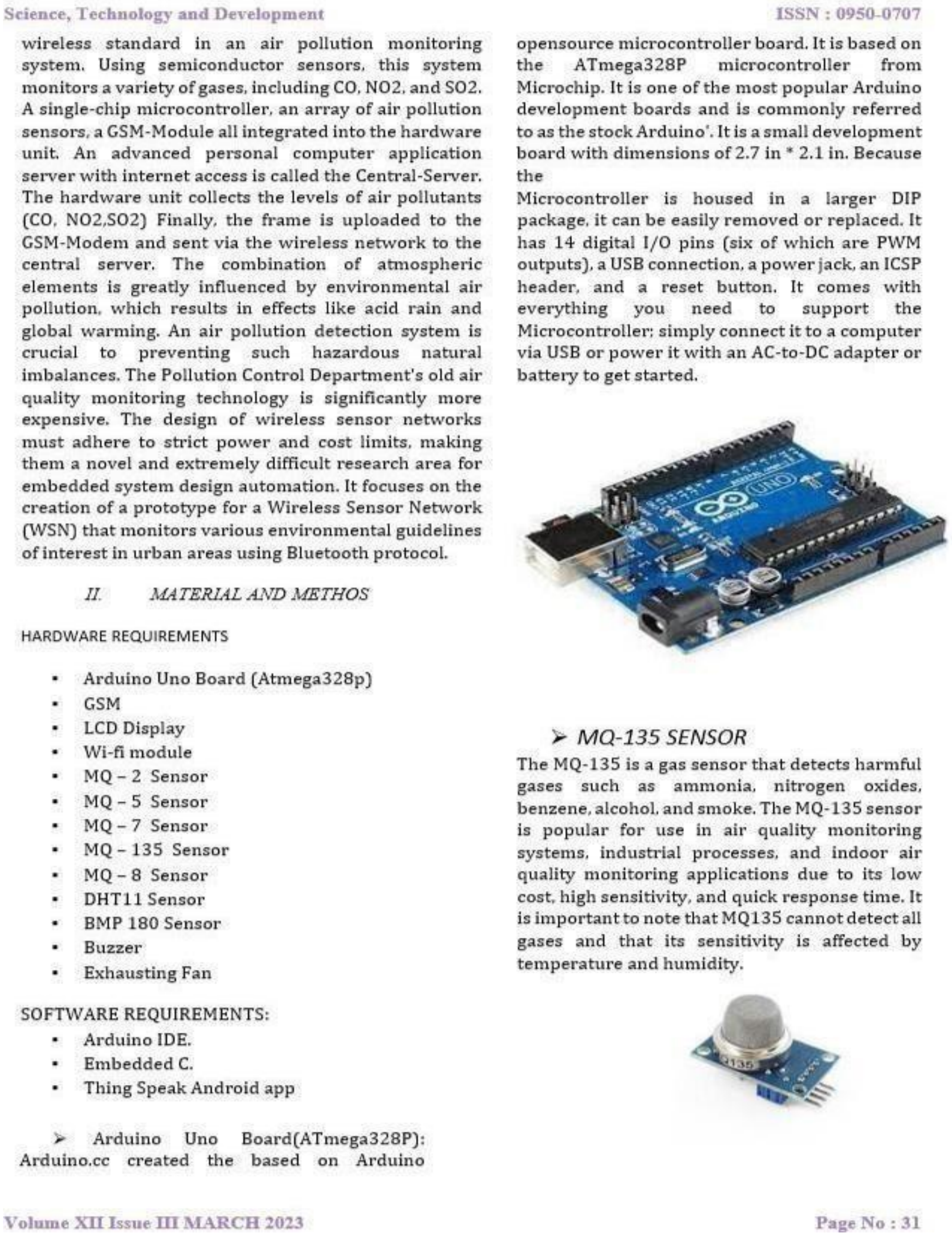
I. INTRODUCTION

Air pollution is a growing global concern because it harms both the environment and human health. To address this issue, researchers and engineers created an air pollution monitoring system based on the Internet of Things (IoT). This system collects real-time data on air quality using IoT sensors, which is then transmitted and analyzed to provide insights into pollution levels. The Internet of Thingsbased air pollution monitoring system has the potential to change the way we monitor air quality. When compared to traditional monitoring methods, it provides a more efficient and accurate method of data collection. To provide a comprehensive picture of pollution levels, the system can be deployed in a variety of locations, including urban and rural areas. In this paper, we will discuss the design and implementation of an IoT-based air pollution monitoring system. We will cover the various components of the system, including the sensors used, data transmission protocols, and data analysis techniques. We will also highlight the benefits of using an IoT-based approach for air pollution monitoring and the potential applications of this technology. Overall, this paper aims to provide a comprehensive overview of the IoT-based air pollution monitoring system and its potential impact on the environment and public health.

The difficulty with traditional monitoring instruments is their large size, heavy weight, and exorbitant price.

This results in insufficient monitoring station deployment. Because the air pollution situation in urban areas is highly related to human activities (e.g., construction activities) and locationdependent (e.g., traffic choke-points have much worse air quality than average), monitoring station locations must be carefully chosen. The IOT Based Air Pollution Monitoring System monitors the air quality over a webserver via the internet and will activate an alarm when the air quality certain level high , indicating the presence of harmful gases such as CO2, smoke, alcohol, benzene, NH3, NOx, and LPG. The system will display the air. PPM quality is displayed on the LCD as well as on the website, allowing for easy monitoring. The system detects and monitors temperature and humidity.

The combination of the main air pollutant gases has been conceived, developed, and observed using a



Science, Technology and Development ISSN : 0950-0707

wireless standard in an air pollution monitoring system. Using semiconductor sensors, this system monitors a variety of gases, including CO, NO2, and SO2. A single-chip microcontroller, an array of air pollution sensors, a GSM-Module all integrated into the hardware unit. An advanced personal computer application server with internet access is called the Central-Server. The hardware unit collects the levels of air pollutants (CO, NO2,SO2) Finally, the frame is uploaded to the GSM-Modem and sent via the wireless network to the central server. The combination of atmospheric elements is greatly influenced by environmental air pollution, which results in effects like acid rain and global warming. An air pollution detection system is crucial to preventing such hazardous natural imbalances. The Pollution Control Department's old air quality monitoring technology is significantly more expensive. The design of wireless sensor networks must adhere to strict power and cost limits, making them a novel and extremely difficult research area for embedded system design automation. It focuses on the creation of a prototype for a Wireless Sensor Network (WSN) that monitors various environmental guidelines of interest in urban areas using Bluetooth protocol.

II. MATERIAL AND METHOS

HARDWARE REQUIREMENTS

- Arduino Uno Board (Atmega328p)
- GSM
- LCD Display
- Wi-fi module
- MQ – 2 Sensor
- MQ – 5 Sensor
- MQ – 7 Sensor
- MQ – 135 Sensor
- MQ – 8 Sensor
- DHT11 Sensor
- BMP 180 Sensor
- Buzzer
- Exhausting Fan

SOFTWARE REQUIREMENTS:

- Arduino IDE.
- Embedded C.
- Thing Speak Android app

➢ Arduino Uno Board(ATmega328P): Arduino.cc created the based on Arduino opensource microcontroller board. It is based on the ATmega328P microcontroller from Microchip. It is one of the most popular Arduino development boards and is commonly referred to as the stock Arduino'. It is a small development board with dimensions of 2.7 in * 2.1 in. Because the

Microcontroller is housed in a larger DIP package, it can be easily removed or replaced. It has 14 digital I/O pins (six of which are PWM outputs), a USB connection, a power jack, an ICSP header, and a reset button. It comes with everything you need to support the Microcontroller; simply connect it to a computer via USB or power it with an AC-to-DC adapter or battery to get started.

➢ *MQ-135 SENSOR*

The MQ-135 is a gas sensor that detects harmful gases such as ammonia, nitrogen oxides, benzene, alcohol, and smoke. The MQ-135 sensor is popular for use in air quality monitoring systems, industrial processes, and indoor air quality monitoring applications due to its low cost, high sensitivity, and quick response time. It is important to note that MQ135 cannot detect all gases and that its sensitivity is affected by temperature and humidity.

Science, Technology and Development ISSN : 0950-0707

- **MQ – 2 SENSOR**

MQ-2 is a gas sensor module capable of detecting a wide range of gases such as smoke, propane, methane, and carbon monoxide. MQ-2 sensor modules are commonly found in gas detection systems, gas leakage detectors, and smoke detectors.

- **MQ – 5 SENSOR**

The MQ-5 gas sensor is a type of gas sensor that detects the presence of flammable gases such as LPG, natural gas, and propane. It, like other MQ gas sensors, operates on the metal oxide semiconductor (MOS) principle, detecting changes in electrical resistance when the target gas comes into contact with the sensor.The MQ-5 sensor is widely used in gas leak detection systems for both residential and industrial applications. It should be noted that the MQ-5 sensor is only sensitive to flammable gases and may not be appropriate for detecting other gases.

- MQ – 8 Sensor

The MQ-8 gas sensor is a gas sensor that detects the presence of hydrogen gas. It works by detecting changes in electrical resistance when hydrogen gas comes into contact with the sensor, which is based on the principle of metal oxide semiconductors (MOS).The MQ-8 sensor is suitable for a wide range of applications, including gas leak detection, hydrogen fuel cell monitoring, and industrial process control. It should be noted that the MQ-8 sensor detects only hydrogen gas and may not be suitable for detecting other gases.

- MQ- 7 Sensor

The MQ-7 gas sensor is a gas sensor that detects the presence of carbon monoxide (CO) gas. It operates on the metal oxide semiconductor (MOS) principle, detecting changes in electrical resistance when CO gas comes into contact with the sensor. The MQ-7 sensor is widely used in carbon monoxide detection systems for residential and industrial use. It should be noted that the MQ-7 sensor is only sensitive to carbon monoxide and may not be capable of detecting other gases.

- DHT- 11Sensor

The DHT11 sensor is a digital temperature and humidity sensor that is widely used in electronic projects. It measures the humidity and temperature of the surrounding air using a capacitive humidity sensor and a thermistor, respectively. The DHT11 sensor is reasonably priced, simple to see, and only requires a few connections to a microcontroller or other digital circuit. It can measure temperatures from 0 to 50°C with a 2°C accuracy and relative humidity from 20 to 90% with a 5% accuracy.

- BMP 180 Sensor

Volume XII Issue III MARCH 2023 Page No : 32

The BMP180 sensor is a digital barometric pressure and temperature sensor that is commonly used in weather monitoring, altitude sensing, and other applications where atmospheric pressure measurements are required. To measure barometric pressure and temperature, it employs a piezoresistive sensor and an integrated temperature sensor. The BMP180 sensor is low-cost and simple to use, with only a few connections to a microcontroller or other digital circuit required. It can measure atmospheric pressure from 300 to 1100 (hectopascals) with a 0.1 accuracy and temperature from -40 to 85°C with a 1°C accuracy.

- GSM

A GSM modem, also known as a GSM module, is a device that provides a wireless data link to a network by utilising GSM mobile telephone technology. GSM modems are employed in

Mobile phones and other devices that communicate with mobile phone networks. SIM cards are used to identify their device to the network. A customised Global System for Mobile Communication (GSM) module is being developed for wireless radiation monitoring via Short Messaging Service (SMS). This module can receive serial data from radiation monitoring devices such as a survey or an area monitor and send it to a host server as text SMS.

- LCD Display

The LCD display 16x2 is an alphanumeric display with 16 columns and 2 rows of characters. It can display up to 32 characters at once and is widely used in a variety of electronic projects, including microcontroller-based systems, Arduino projects, and other digital circuits. The LCD display 16x2 is an alphanumeric display with 16 columns and 2 rows of characters. It can display up to 32 characters at once and is widely used in a variety of electronic projects, including microcontroller-based systems, Arduino projects, and other digital circuits. The LCD display 16x2 functions by receiving digital signals from a microcontroller or other digital circuit and converting them into characters or symbols that appear on the screen. It typically employs a Hitachi HD44780 or compatible controller chip to facilitate communication between the display and the microcontroller. The LCD display 16x2 can display a variety of information, including text, numbers, and symbols. It can also be used to display graphics by implementing more advanced display algorithms or by using custom characters.

- WIFI Module

A Wi-Fi module is an electronic component that allows a microcontroller or other digital circuit to communicate wirelessly. It usually consists of a Wi-Fi chip, an antenna, and firmware that allows wireless communication over a Wi-Fi network.

Wi-Fi modules are available in a variety of sizes, shapes, and functionalities. Some modules are intended to be integrated into existing devices or circuits, while others are intended to be used as standalone modules to add WiFi connectivity to a project. Wi-Fi modules can support a wide range of Wi-Fi standards, including 802.11 b/g/n/ac, and operate in a variety of frequency bands. They can also support a variety of security protocols, including WEP, WPA, and WPA2.

- Exhaust fan

An exhaust fan is a common electronic component used to help dissipate heat generated by electronic components such as microcontrollers, power transistors, and voltage regulators.

Heat can cause electronic components to malfunction or even fail, so keeping them cool is critical. An exhaust fan can aid in this process by drawing hot air away from the components and expelling it outside the enclosure or case.

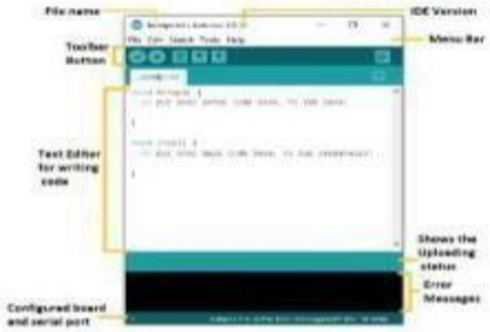
➢ Buzzer

Buzzer is used to add the sound feature. It is light in weight, Good performance, general purpose musical buzzer are commonly used in alerting / alarming circuits, kids toys etc.. This buzzer is used as an external buzzer which operates in wide range of voltage (3V to 12V). Most commonly used buzzers operates in at 9V & 12v. They have long life, stable performance, High Quality with the SOT plastic package.

➢ Arduino IDE

The IDE we used here is 'Arduino IDE' and the programming language used is 'Embedded C'. The Arduino has all the features like-

- Editor
- Cross compiler
- Debugger
- Serial monitor


File name
IDE Version
Menu Bar
Toolbar Button
Text Editor for writing code
Shows the Uploading status
Error Messages
Configured board and serial port

➢ Thing Speak:

Thing Speak is an IoT analytics platform service that allows you to aggregate, visualize, and live data streams in the cloud. You can send data to Thing Speak from your devices, create instant visualization of live data, and send alerts. Thing Speak is an opensource software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.

III . SYSTEM DESIGN

The system is intended to monitor the air quality and temperature of a closed space and to notify the user if the air quality is hazardous. A microcontroller board, a GSM module, an LCD display, a Wi-Fi module, a buzzer, and an exhaust fan comprise the system. Sensors: Several sensors are used in the system to monitor the air quality and temperature of the space. Among these sensors are:
MQ-2 gas detector for smoke, methane, propane, and other gases
MQ-5 sensor for natural gas, propane, and hydrogen detection
MQ-7 sensor for carbon monoxide detection

MQ-135 air quality sensor detects hazardous gases such as ammonia, nitrogen oxides, and sulphur dioxide. MQ-8 sensor for hydrogen gas detection
DHT11 sensor for temperature and humidity measurement
BMP180 sensor for pressure, altitude, and temperature measurement
The Arduino Uno board with Atmega328p serves as the main microcontroller board, processing data from sensors and controlling the system.
The GSM module is used to send SMS alerts to the user if any hazardous gases are detected.

LCD Display: The LCD display is used to show sensor readings and system status.

The Wi-Fi module is used to upload sensor readings to Thing Speak, which is accessible via an Android app. Buzzer: A buzzer is used to sound an alarm if dangerous gases are detected.

Exhaust Fan: The exhaust fan expels hazardous gases from the enclosed space.

Software Requirements: The Arduino IDE and Embedded C are used to programming the system. To write, compile, and upload code to the microcontroller board, the Arduino IDE is used. The Embedded C programming language is used to programming the microcontroller board in order to read sensor data, process it, and control the system.

Thing Speak Android App: The Thing Speak Android app allows you to remotely view sensor data and system status. The user can view the graphs and set the alert thresholds after accessing the sensor data.

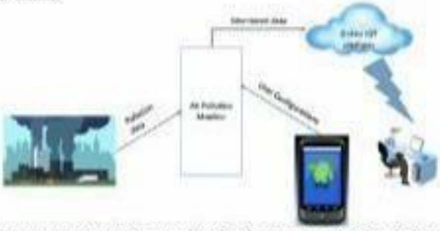
IV . WORKING

The project is intended to monitor the air quality and temperature of a closed space and to notify the user if the air quality is hazardous. The system measures air quality and temperature using various sensors, and a microcontroller board processes the data and controls the system. Sensor Reading

The sensors are linked to the microcontroller board and continuously measure the enclosed space's air quality and temperature. The microcontroller board then processes the sensor readings. Data Processing

The sensor data is processed by the microcontroller board, which determines whether the air quality is hazardous or not. If the air quality is hazardous, the system activates the buzzer to sound an alarm and sends an SMS alert to the user via the GSM module. The system also activates the exhaust fan to expel the hazardous gases from the enclosed space. The LCD display displays sensor readings as well as system status. The user can view sensor data and system status on the LCD display.

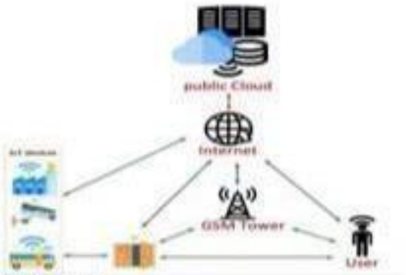
The Wi-Fi module sends sensor data to Thing Speak, which is accessible via the Android app. The app allows the user to view sensor data, set alert thresholds, and receive alerts. The Android app allows the user to access sensor data, view graphs, and set alert thresholds. If the air quality is hazardous, the user can also receive SMS alerts.



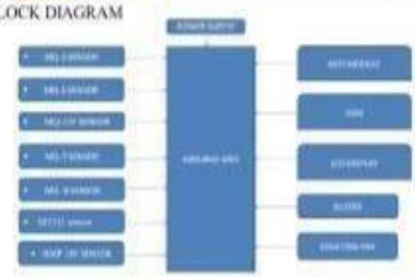
The system should be maintained on a regular basis by checking the sensors and their connections. To ensure accurate readings, the system should be cleaned on a regular basis. Overall, this system is intended to monitor the air quality and temperature of an enclosed space, providing the user with real-time data and alerts. Overall, the goal of this project is to monitor the air quality and temperature of an enclosed space and notify the user if the air quality is hazardous. The system provides the user with realtime data and alerts, ensuring a safe and healthy environment.

V . BLOCK DIAGRAM

Here the simulation circuit of system design is shown below. We can see Arduino as main controlling unit in this connected with GSM, wi-fi module, lcd display, Gas sensors, DHT-11 Sensor,
BMP-180 Sensor, buzzer and exhausting fan



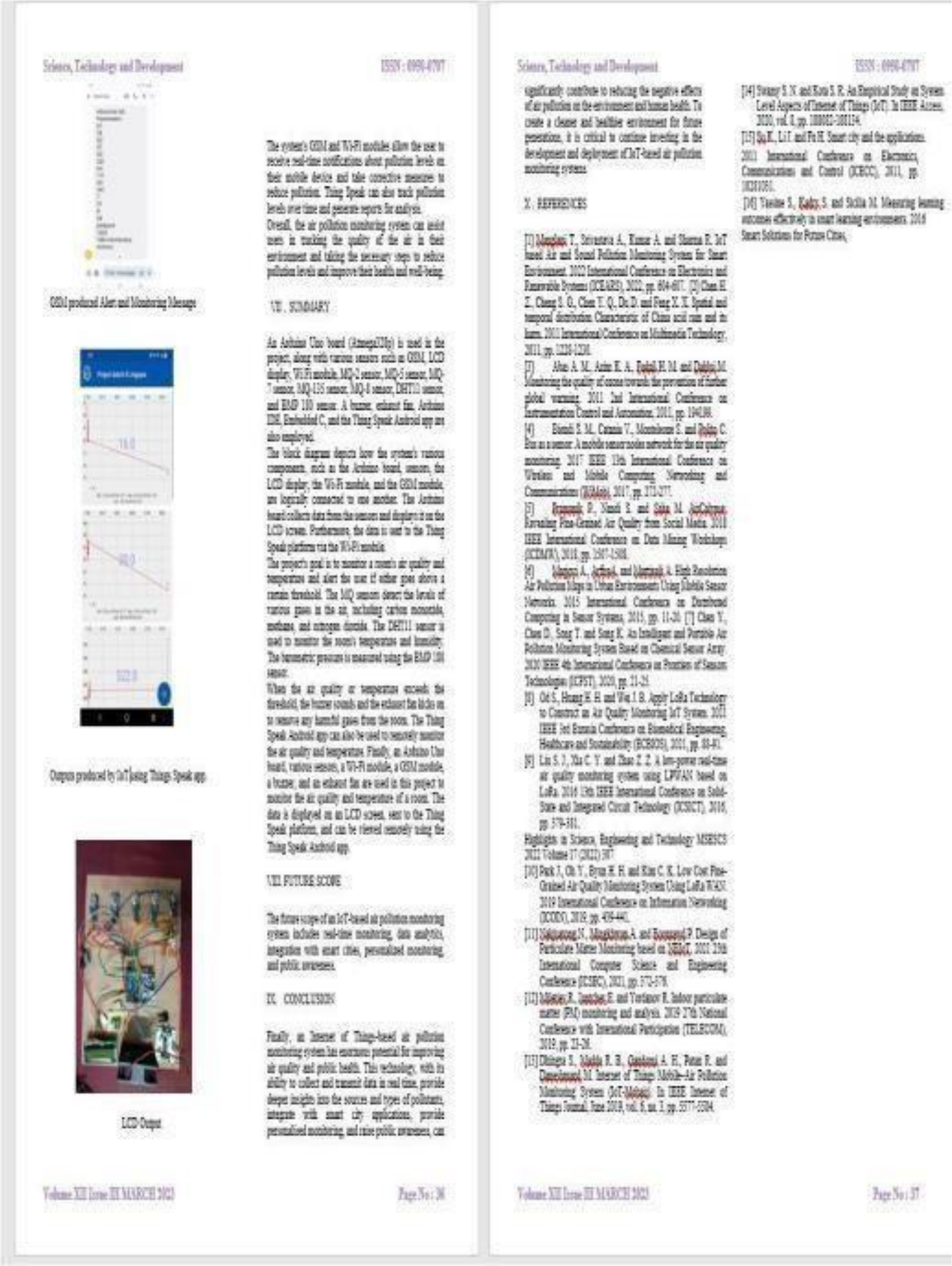
BLOCK DIAGRAM



The project's block diagram shows that the system includes sensors, a microcontroller board, a GSM module, an LCD display, a Wi-Fi module, a buzzer, and an exhaust fan. The sensors measure the air quality and temperature, and the microcontroller board processes and controls the system based on the sensor data. If the air quality is hazardous, the GSM module sends SMS alerts to the user, and the Wi-Fi module uploads sensor data to Thing Speak. The LCD display displays sensor readings and system status, whereas the Android app displays sensor data graphically and allows the user to set alert thresholds and receive alerts. The exhaust fan expels hazardous gases from the enclosed space. Overall, the project is a comprehensive air quality monitoring system that provides the user with real-time data and alerts. By detecting hazardous gases and expelling them from the enclosed space, the system ensures a safe and healthy environment. The Android app allows the user to access sensor data, view graphs, and set alert thresholds, making it simple to maintain the system and ensure accurate readings.

VI . RESULTS & DISCUSION

Now we see the output of the project message received through GSM mobile



Science, Technology and Development

ISSN : 0950-0707

GSM produced Alert and Monitoring Message

Output produced by IoT using Thing Speak app

LCD Output

The system's GSM and Wi-Fi modules allow the user to receive real-time notifications about pollution levels on their mobile device and take corrective measures to reduce pollution. Thing Speak can also track pollution levels over time and generate reports for analysis.

Overall, the air pollution monitoring system can assist users in tracking the quality of the air in their environment and taking the necessary steps to reduce pollution levels and improve their health and well-being.

VII . SUMMARY

An Arduino Uno board (Atmega328p) is used in the project, along with various sensors such as GSM, LCD display, Wi-Fi module, MQ-2 sensor, MQ-5 sensor, MQ-7 sensor, MQ-135 sensor, MQ-8 sensor, DHT11 sensor, and BMP 180 sensor. A buzzer, exhaust fan, Arduino IDE, Embedded C, and the Thing Speak Android app are also employed.

The block diagram depicts how the system's various components, such as the Arduino board, sensors, the LCD display, the Wi-Fi module, and the GSM module, are logically connected to one another. The Arduino board collects data from the sensors and displays it on the LCD screen. Furthermore, the data is sent to the Thing Speak platform via the Wi-Fi module.

The project's goal is to monitor a room's air quality and temperature and alert the user if either goes above a certain threshold. The MQ sensors detect the levels of various gases in the air, including carbon monoxide, methane, and nitrogen dioxide. The DHT11 sensor is used to monitor the room's temperature and humidity. The barometric pressure is measured using the BMP 180 sensor.

When the air quality or temperature exceeds the threshold, the buzzer sounds and the exhaust fan kicks on to remove any harmful gases from the room. The Thing Speak Android app can also be used to remotely monitor the air quality and temperature. Finally, an Arduino Uno board, various sensors, a Wi-Fi module, a GSM module, a buzzer, and an exhaust fan are used in this project to monitor the air quality and temperature of a room. The data is displayed on an LCD screen, sent to the Thing Speak platform, and can be viewed remotely using the Thing Speak Android app.

VIII FUTURE SCOPE

The future scope of an IoT-based air pollution monitoring system includes real-time monitoring, data analytics, integration with smart cities, personalized monitoring, and public awareness.

IX. CONCLUSION

Finally, an Internet of Things-based air pollution monitoring system has enormous potential for improving air quality and public health. This technology, with its ability to collect and transmit data in real time, provide deeper insights into the sources and types of pollutants, integrate with smart city applications, provide personalized monitoring, and raise public awareness, can

Volume XII Issue III MARCH 2023

Page No : 36

Science, Technology and Development

ISSN : 0950-0707

significantly contribute to reducing the negative effects of air pollution on the environment and human health. To create a cleaner and healthier environment for future generations, it is critical to continue investing in the development and deployment of IoT-based air pollution monitoring systems.

X . REFERENCES

[1] Mangla T., Srivastava A., Kumar A. and Sharma R. IoT based Air and Sound Pollution Monitoring System for Smart Environment. 2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022, pp. 604-607. [2] Chen H. Z., Cheng S. G., Chen T. Q., Du D. and Feng X. X. Spatial and temporal distribution Characteristic of China acid rain and its harm. 2011 International Conference on Multimedia Technology, 2011, pp. 1228-1230.

[3] Alias A. M., Asim K. A., Fadzil M. M. and Dahlan M. Monitoring the quality of ozone towards the prevention of further global warming. 2011 2nd International Conference on Instrumentation Control and Automation, 2011, pp. 104-108.

[4] Bisandi S. M., Catania V., Monteleone S. and Polito C. Bus as a sensor: A mobile sensor nodes network for the air quality monitoring. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2017, pp. 272-277.

[5] Pramanik P., Nandi S. and Saha M. AirCalypse: Revealing Fine-Grained Air Quality from Social Media. 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2018, pp. 1507-1508.

[6] Marjovi A., Arfire A. and Martinoli A. High Resolution Air Pollution Maps in Urban Environments Using Mobile Sensor Networks. 2015 International Conference on Distributed Computing in Sensor Systems, 2015, pp. 11-20. [7] Chen Y., Chen D., Song T. and Song K. An Intelligent and Portable Air Pollution Monitoring System Based on Chemical Sensor Array. 2020 IEEE 4th International Conference on Frontiers of Sensors Technologies (ICFST), 2020, pp. 21-25.

[8] Ok S., Huang H. H. and Wu J. B. Apply LoRa Technology to Construct an Air Quality Monitoring IoT System. 2021 IEEE 3rd Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS), 2021, pp. 88-91.

[9] Liu S. J., Xia C. Y. and Zhao Z. Z. A low-power real-time air quality monitoring system using LPWAN based on LoRa. 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2016, pp. 379-381.

Highlights in Science, Engineering and Technology MSESCS 2022 Volume 17 (2022) 307

[10] Park J., Oh Y., Byun H. H. and Kim C. K. Low Cost Fine-Grained Air Quality Monitoring System Using LoRa WAN. 2019 International Conference on Information Networking (ICOIN), 2019, pp. 439-441.

[11] Sukhwong N., Mungkhrun A. and Boonsawad P. Design of Particulate Matter Monitoring based on NB-IoT. 2021 25th International Computer Science and Engineering Conference (ICSEC), 2021, pp. 372-376.

[12] Mileva R., Iontchev E. and Yordanov R. Indoor particulate matter (PM) monitoring and analysis. 2019 27th National Conference with International Participation (TELECOM), 2019, pp. 23-26.

[13] Dhingra S., Madda R. B., Gandomi A. H., Patan R. and Daneshmand M. Internet of Things Mobile–Air Pollution Monitoring System (IoT-Mobair). In IEEE Internet of Things Journal, June 2019, vol. 6, no. 3, pp. 5577-5584.

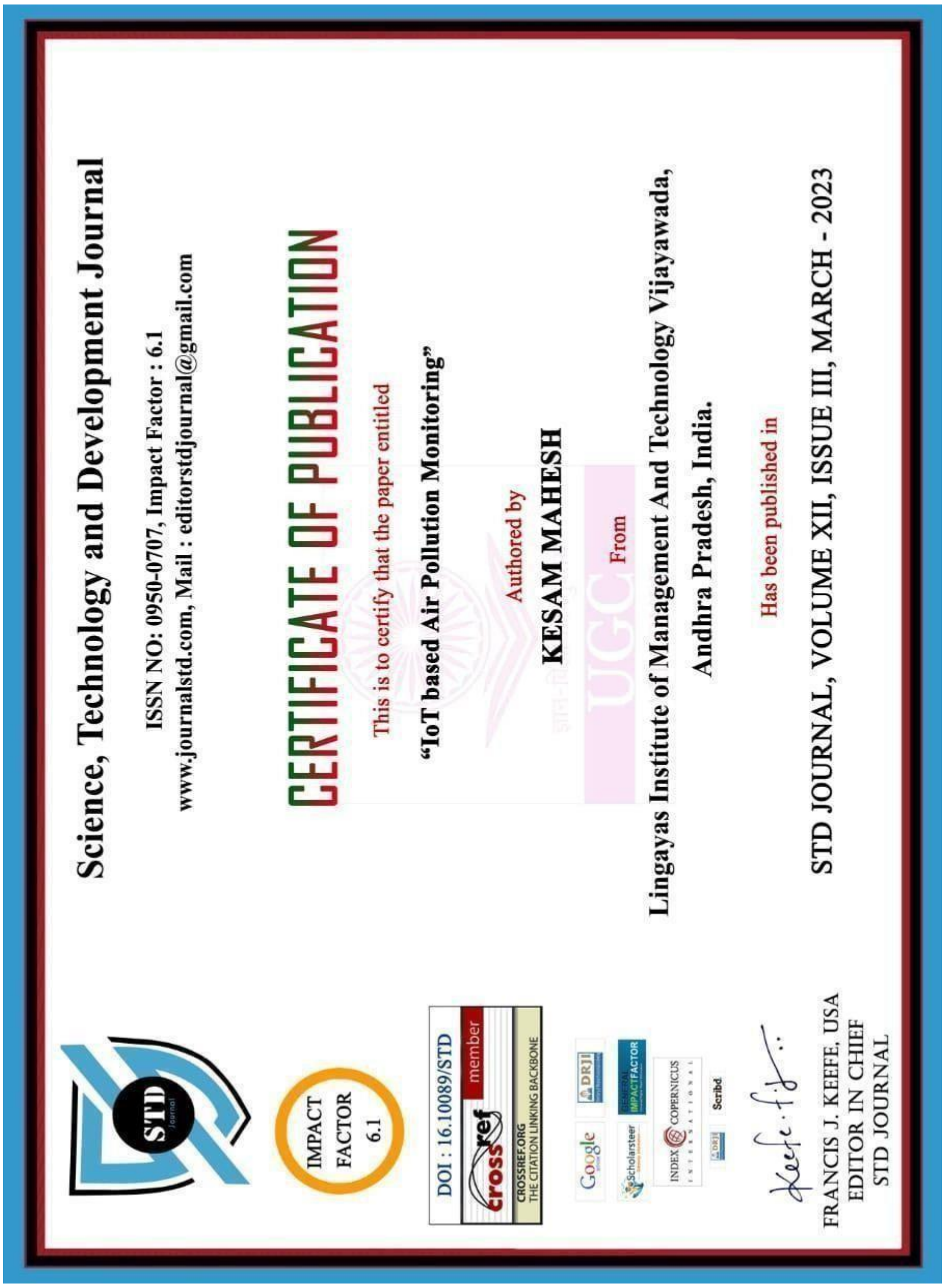
[14] Swamy S. N. and Kota S. R. An Empirical Study on System Level Aspects of Internet of Things (IoT). In IEEE Access, 2020, vol. 8, pp. 188082-188134.

[15] Su K., Li J. and Fu H. Smart city and the applications. 2011 International Conference on Electronics, Communications and Control (ICECC), 2011, pp. 1028-1031.

[16] Yassine S., Kadry S. and Sicilia M. Measuring learning outcomes effectively in smart learning environments. 2016 Smart Solutions for Future Cities,

Volume XII Issue III MARCH 2023

Page No : 37



Science, Technology and Development Journal

ISSN NO: 0950-0707, Impact Factor : 6.1

www.journalstd.com, Mail : editorstdjournal@gmail.com

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

"IoT based Air Pollution Monitoring"

Authored by

KESAM MAHESH

From

Lingayas Institute of Management And Technology Vijayawada, Andhra Pradesh, India.

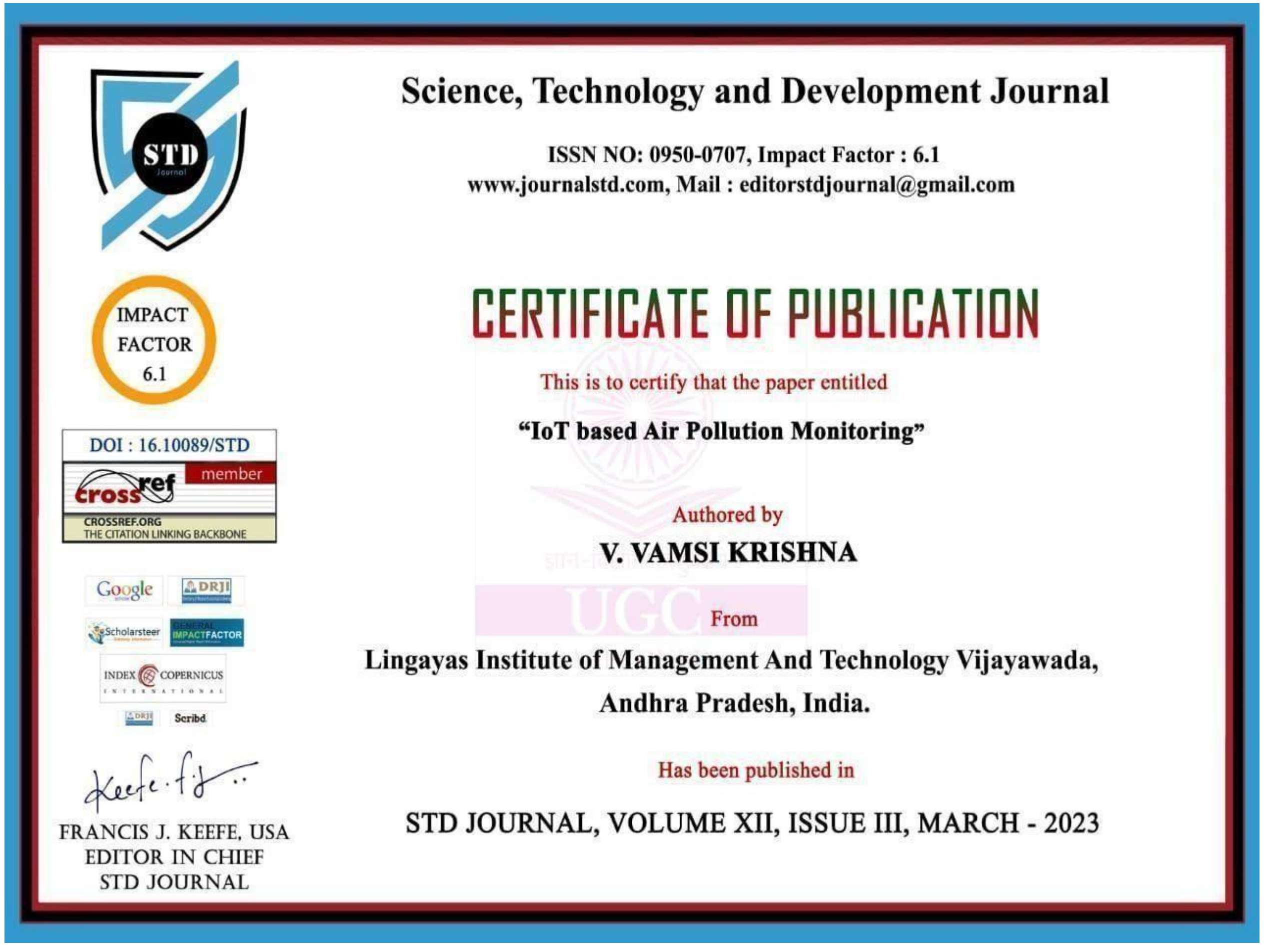
Has been published in

STD JOURNAL, VOLUME XII, ISSUE III, MARCH - 2023

IMPACT FACTOR 6.1

DOI : 16.10089/STD

FRANCIS J. KEEFE, USA
EDITOR IN CHIEF
STD JOURNAL



Science, Technology and Development Journal

ISSN NO: 0950-0707, Impact Factor : 6.1
www.journalstd.com, Mail : editorstdjournal@gmail.com

IMPACT
FACTOR
6.1

DOI : 16.10089/STD

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

“IoT based Air Pollution Monitoring”

Authored by

V. VAMSI KRISHNA

From

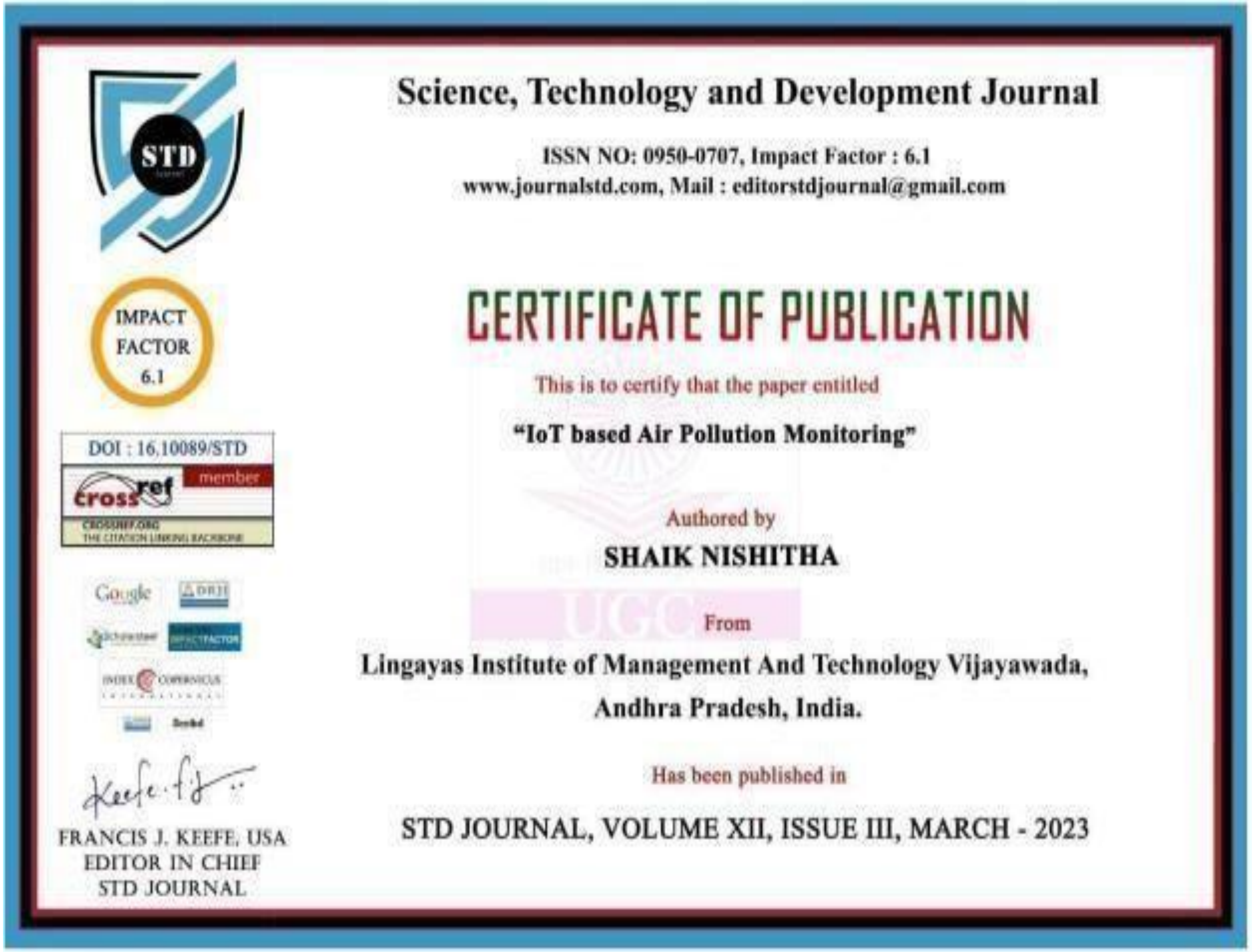
**Lingayas Institute of Management And Technology Vijayawada,
Andhra Pradesh, India.**

Has been published in

STD JOURNAL, VOLUME XII, ISSUE III, MARCH - 2023

Keefe.f.j

FRANCIS J. KEEFE, USA
EDITOR IN CHIEF
STD JOURNAL



Science, Technology and Development Journal

ISSN NO: 0950-0707, Impact Factor : 6.1
www.journalstd.com, Mail : editorstdjournal@gmail.com

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

"IoT based Air Pollution Monitoring"

Authored by

SHAIK NISHITHA

From

Lingayas Institute of Management And Technology Vijayawada, Andhra Pradesh, India.

Has been published in

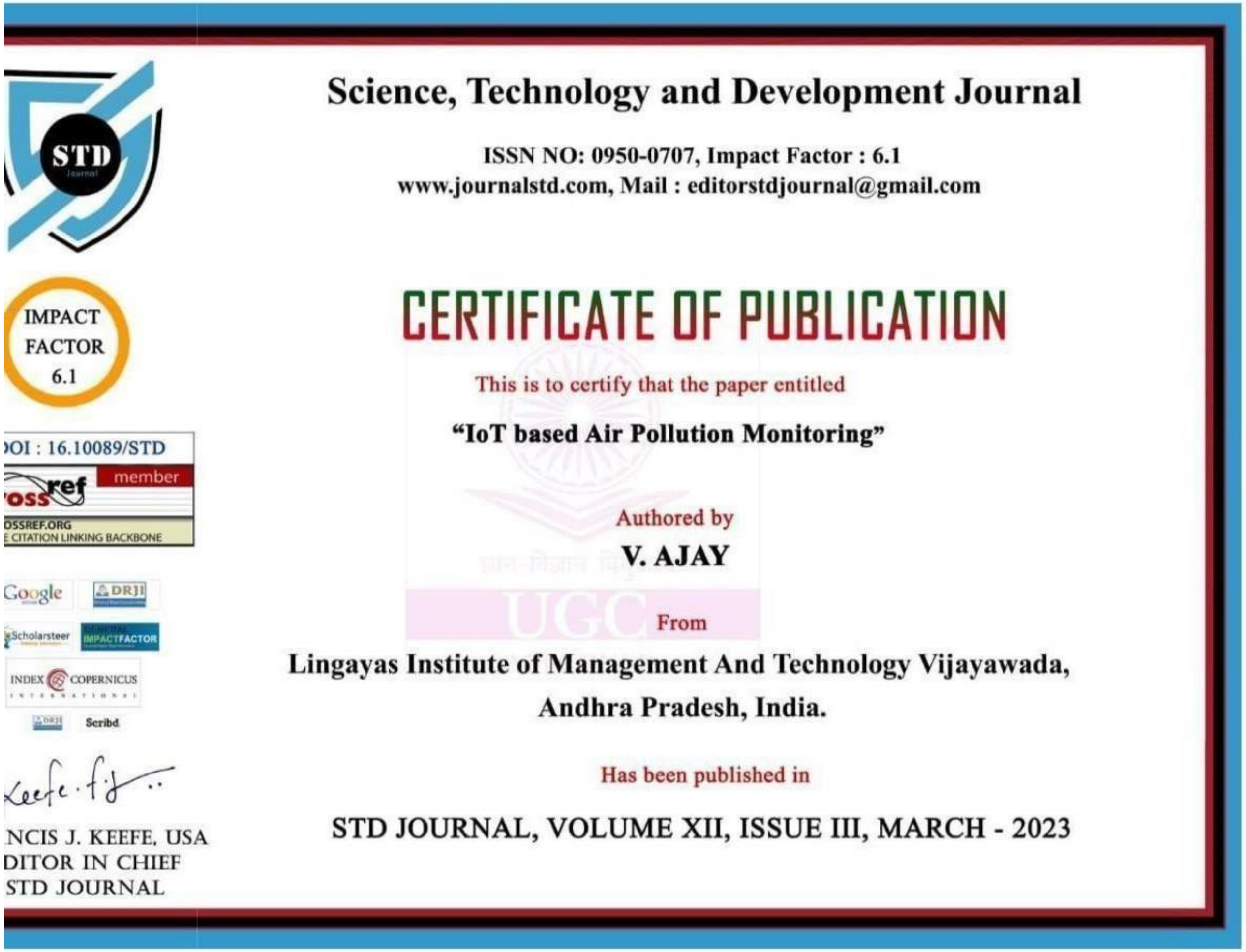
STD JOURNAL, VOLUME XII, ISSUE III, MARCH - 2023

IMPACT FACTOR 6.1

DOI : 16.10089/STD

crossref member

FRANCIS J. KEEFE, USA
EDITOR IN CHIEF
STD JOURNAL



Science, Technology and Development Journal

ISSN NO: 0950-0707, Impact Factor : 6.1
www.journalstd.com, Mail : editorstdjournal@gmail.com

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

"IoT based Air Pollution Monitoring"

Authored by

V. AJAY

From

Lingayas Institute of Management And Technology Vijayawada,
Andhra Pradesh, India.

Has been published in

STD JOURNAL, VOLUME XII, ISSUE III, MARCH - 2023

Keefe

NCIS J. KEEFE, USA
DITOR IN CHIEF
STD JOURNAL