DESIGN AND ANALYSIS OF IOT BASED AIR MONITORING SYSTEM

A PROJECT REPORT

Submitted by

G LOHITH KRISHNA (17113228)
D ARJUN (17113227)
M ABHISHEK (18113218)

Under the guidance of

Dr. MAGDALENE.R

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)
CHENNAI

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE
CHENNAI - 603 103

MAY 2022

BONAFIDE CERTIFICATE

Certified that this project report **"DESIGN AND ANALYSIS OF IOT BASED AIR MONITORING SYSTEM"** is the bonafide work of **"G LOHITH KRISHNA(17113228), D ARJUN(17113227), M ABHISHEK(18113218)** who carried out the project work under my supervision during the academic year 2**021-2022**.

Dr. J. THANGAKUMAR,
HoD,
Department of CSE.

Dr.R.MAGDALENE
SUPERVISOR
Ast Professor of CSE

INTERNAL EXAMINER
Name:______________________
Designation: ________________

EXTERNAL EXAMINER
Name: ______________________
Designation:___________________

Project Viva - voce conducted on _______________

TABLE OF CONTENTS

APPENDIX A: SAMPLE SCREEN

APPENDIX B: SAMPLE CODE

APPENDIX C: PLAGIARISM REPORT

APPENDIX D: FUNDING / PATENT / PUBLICATION DETAILS

APPENDIX E: TEAM DETAILS

ACKNOWLEDGEMENT

First and foremost we would like to thank **ALMIGHTY** who has provided us the strength to do justice to our work and contribute our best to it.

We wish to express our deep sense of gratitude from the bottom of our heart to our guide **DR.R.MAGDALENE Assistant Professor of**, **Computer Science and Engineering,** for her motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

We would like to extend our heartfelt gratitude to **Dr. J. Thngakumar, Ph.D., Professor & Head, Department of Computer Science and Engineering** for his valuable suggestions and support in successfully completing the project.

We thank the management of **HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE** for providing us the necessary facilities and support required for the successful completion of the project.

As a final word, we would like to thank each and every individual who have been a source of support and encouragement and helped us to achieve our goal and complete our project work successfully.

ABSTRACT

The level of pollution has increased with times by IoT of factors like the increase in population, increased vehicle use, industrialization and urbanization which results in harmful effects on human wellbeing by directly affecting health of population exposed to it. In order to monitor in this project we are going to make an IoT Based Air Pollution Monitoring Systems in which we will monitor the air quality over a cloud server using internet and will trigger a alarm where the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in air like CO2, smoke, alcohol, benzene and NH3. It will show the air quality in PPM on the LCD and as well as on mobile application so that we it very every easily. In this IoT project, you can monitor the pollution level from anywhere using your computer or mobile. Thus, poor air quality causes several health hazards like heart disease, lung cancer, and respiratory problems. The need of the hour is not only to control air pollution but also materializing technologies, devices and software systems to keep a close check on rapidly growing air pollution. The efforts in this paper are made to develop one of such systems by which the real-time air quality monitoring can be done to take preventative measures to make our living environment safe to live a good life.

LIST OF FIGURES

LIST OF ABBREVIATIONS(sample,use your own)

WHO	World Health Organization
IOT	Internet Of Things
AQI	Air Quality Index
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable read-only Memory
LCD	Liquid Crystal Display

CHAPTER 1
INTRODUCTION

1.1 Introduction

Climate change has been causing significant impacts on people's living conditions, stability, and socio-economic development in day by day. Besides, due to population growth, the rapid increase of industrialization, as well as indiscriminate use of chemical fertilizers and pesticides in agriculture, are causing serious impacts on the environment, climate, and public health. Air quality is getting worse noticeably, especially in urban areas where high population density and many industrial parks. World Health Organization (WHO) announced that there are approximately 4.2 million premature deaths globally are linked to ambient air pollution. Thus, poor air quality causes several health hazards like heart disease, lung cancer, and respiratory problems. The need of the hour is not only to control air pollution but also materializing technologies, devices and software systems to keep a close check on rapidly growing air pollution. The efforts in this paper are made to develop one of such systems by which the real-time air quality monitoring can be done to take preventative measures to make our living environment safe to live a good life. The implementation of air quality monitoring systems is essential for areas that are at high risk of air pollution, directly affecting human health such as industrial parks, production complexes, university laboratory complexes, and so on. The goal of this study is to design and implement an affordable air quality monitoring systems in areas with several kilometers radius. The monitoring AQI values are reported on a public website for the purposes of early warning and preventive action.

1.2 PROBLEM STATEMENT

Air pollution is one of environmental issues that cannot be ignored. Inhaling pollutants for a long time causes damages in human health. Traditional air quality monitoring methods, such as building air quality monitoring stations, are typically expensive. This project is suitable for air quality monitoring in real time. Design a tool which will sense quality of air and display it in the form of percentage, Sense how much carbon mono-oxide(CO) is present in air and display in the form of percentage, Sense the temperature and display it in degree celcius.

1.3 SCOPE

In the future work, we can modify the system to notify a user about the air quality when it reaches beyond when it reaches beyond a permissible level through SMS or APP. Notification, furthermore, the sensors in the system can be calibrated more so that we can get more accurate and get data for more harmful gases such as Ammonia, oxides of nitrogen, etc. Our work can demonstrate vast opportunities to work on the device, on the app and also on the field using the device that we have worked with. The device can be used any time efficiently in different locations of a city and then research with the achieved data for that particular area in that city. The device can be updated with additional sensors that can sense data from the existence of other gases such as O2 and H2. These gases will provide the condition of the atmosphere and authority can take into further decisions accordingly. The sensors that we have been worked with can also be reset according to most recent time update. The android app which we have developed for turning on and off the device can be updated with newer features by implementing necessary codes.

1.4 MOTIVATION

As seen in the previous overview to the current IoT, most IoT applications in domain specific or application specific solutions. The architectures of these IoT systems are split and cannot correlate and integrate the data from different storage tower, these isolated IoT keys use private protocols and cause much difficulties in information sharing, technology multiplexing, network managements, and advancement.

1.5 APPLICATION

- It is used to calculate the air quality in surroundings.
- It stores the data in cloud storage (google cloud).
- Sensors are interacting with traffic system.
- Mobile app is used to monitor the air pollution level.

1.6 Organisation of The Thesis

Chapter 2 discusses about the references and research papers we used as a supplementary guide for completion of this project.

Chapter 3 describes the project description used in the system and also discusses about the existing system and its flaws. Then shows the importance of proposed system.

Chapter 4 describes the system requirements such as the hardware and software requirements.

Chapter 5 describes the requirements block diagram and complete explanation of the architecture of the system. and software requirements to complete the system design. Chapter 6 describes the modules present in the project and gives the short introduction about each module.

Chapter 7 contains most of the implementation part and the division of complete implementation into various modules. And explanation of each module in a detailed manner. And also shows the snaps of the outputs and the model and brief explanation of the results achieved after implementation of the project.

Chapter 8 includes different test cases to run and check the working of the device and the results obtained for all the test cases.

Chapter 9 is the conclusion part of the project wherein we declare what is achieved after implementation and what is the Future Enhancement or Future Scope of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter is about analysis of the different papers that are published up to now and about the project details that given and discussed in the analysis of paper in detail.

2.2 Inference from Literature

Air pollution in large urban areas has a drastic effect on humans and the environment. Ecological issues in India are growing quickly. Air contamination is mainly caused by vehicles and industries which cause various respiratory diseases, such as asthma and sinusitis. The quality of air is inferior in metropolitan cities like Kolkata, Delhi, and Mumbai due to a large amount of carbon dioxide (CO2) and other harmful gases emitted from vehicles and industries. An extensive number of projects have been described in the literature that utilize low-cost air pollution sensing devices that can be carried by individuals or by versatile vehicles. In two studies, Peterová and Hybler and Bhatt et al. Demonstrated an environmental sensing approach that reinvigorate attention and sympathy of citizens toward pollution. Exposure Sense is a portable participatory sensing framework that is used to screen one’s everyday activities. In another study Zheng et al. presented a cloud-based system that uses knowledge-based discovery to find real-time air quality data. The data are collected by monitoring stations that are placed in various geo-locations. This system uses mobile clients for monitoring purposes. Re et al. Presented an Android application which provides users with information about air quality. By joining user area information and metropolitan air quality data provided by monitoring stations, this application provides a ubiquitous and unobtrusive monitoring framework that is ready to advise users about their daily air pollution exposure. Reshi et al. Designed a WSN platform, called VehNode, that provided automobiles with the capacity to monitor the level of pollutants in smoke released by the vehicle. Mujawar et al. outlined an air pollution contamination measuring system utilizing WSN for use in Solapur City. Micro-sensor nodes detect the target gas by measuring the sensing layer’s electrical conductivity. When the gases touch the surface of the sensor they are assimilated and the conductivity changes. Also, a semiconductor sensor is used at the emission outlet of the vehicle to sense the level of pollutants and transmit this level to the microcontroller. In another study by De Nazelle et al. Demonstrated environmental sensing approaches that reinvigorated the awareness and sympathy of individuals toward pollution. “Air pollution is a challenge that threatens basic human welfare, damages natural and physical capital, and constrains economic growth. We hope this paper will translate the cost of premature deaths into an economic language that resonates with policy makers so that more resources will be devoted to improving air quality. By supporting healthier cities and investments in cleaner sources of energy, we can reduce dangerous emissions, slow climate change, and most importantly save lives,” said Laura Tuck, Vice President for Sustainable Development at the World Bank. “The report of air pollution is a burden of disease associated to urgent call the government for necessary action,” said Dr. Chris Murray, Director of IHME. “Of all the different risk factors for premature deaths, this is one area, the air we breathe, over which individuals have little control. Policy makers in health and environment agencies, as well as leaders in various industries, are facing growing demands—and expectations—to address this problem.”

2.3SUMMARY

This chapter briefly discusses the ideas used in the research papers which are used as reference. And gives advantages and disadvantages in the existing system.

CHAPTER 3

PROJECT DESCRIPTION

3.1 Existing system

In this existing system is the IoT-Mobair App for pollution management, industries based, specifically, streamline and integrate environmental processes, including air emissions analyses, water and energy management, and waste reduction. Such apps provides visibility for users, specific industries into the risk of incidents such as chemical leaks, oil spills and toxic substance disposal, while compliance strengthening with environmental standards and regulations.

3.2 PROPOSED SYSTEM:

In this proposed system is roadside pollution monitoring system for asthma patients. We can also implement zero tolerance fast big data real-time stream analytical tools to process such a complex system. Mobile app is implemented in this project and this app is used to monitor pollution levels in the air. More sensor node is using to monitor the overall traffic pollution level in the particular area.

3.3 REQUIREMENT SPECIFICATIONS

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

CHAPTER 4

4.1 HARDWARE AND SOFTWARE SPECIFICATION

Software Requirement:

Language	: Embedded ‘C’
Compiler	: Arduino IDE

Hardware Requirement:

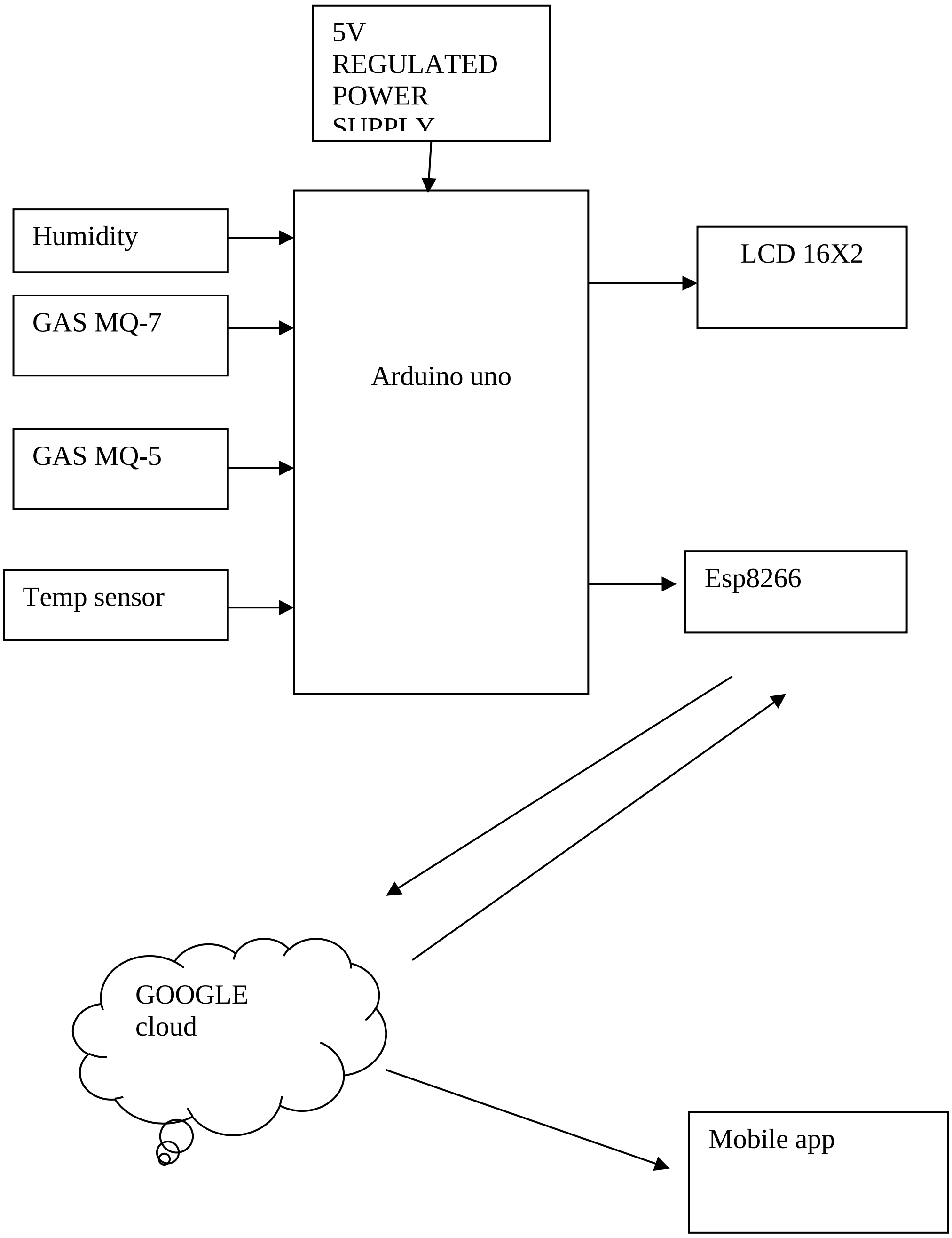
- Arduino Uno
- LCD Display (16x2)-1
- MQ-7 -1
- MQ-5
- Temperature sensor-1
- Humidity sensor
- Esp8266

4.2 TECHNOLOGIES USED:

A Software Requirements Specification (SRS), often known as a software requirements specification, is a detailed description of the behavior of a system that is being created. It provides a series of scenarios that explain all of the software's interactions with users. The SRS also includes non-functional requirements in addition to use cases. Non-functional requirements are those that place limitations on the design or implementation of a system (such as performance engineering requirements, quality standards, or design constraints). Specification of System Requirements It's a collection of data that encapsulates a system's needs. A business analyst, also known as a system analyst, is responsible for studying the business demands of their customers and stakeholders in order to discover and suggest solutions to business challenges. Three types of needed elements apply to projects. In business words, business requirements outline what must be given or performed in order to generate value.

CHAPTER 5

5.1 BLOCK DIAGRAM



5.2 BLOCK DIAGRAM DESCRIPTION:

Above the block diagram is contain power unit, Arduino uno controller, MQ-7, MQ-5, and temperature sensor. Gas sensor and temperature sensor are interfacing with traffic system, then it gets monitored the pollution level in traffic area. This sensor is interfaced with Arduino uno analog pins. The controller, which process the sensor value and send to the server room and mobile app. Mobile app is used by people for monitoring air pollution level. The pollution level is increased the controller is intimate with people and traffic server rooms. This proposed system is used to prevent people from air pollution. Reduce the level of pollution from such sources and to protect humans and the environment from harmful gasses, this air pollution kit was developed that helps a person to detect, monitor, and test air pollution in a given area.

CHAPTER 6

6.1 MODULES:

- **Module 1**:
- In module 1 we have interfaced gas sensor and DHT11 with Arduino UNO microcontroller to gets the air quality and temperature of the product installed location. The first dataset, A Medicare payment data collection includes hospitals level payments for approximately 30k hospitals for the top 100 most frequently billed Día gnostic-Related Groups (DRGS). The top 1oo DRGS account for 60 present of all inpatient related Medicare payments.
- **Module 2**:
- In module 2 we have interfaced LCD to display the sensor values and creating account for cloud database for connecting Node MCU to get the sensor values and update it on the cloud database. The location of the provider is represented by columns such as provider address, ZIP code, state, city, and hospital area referral description. As a result, rather than examining all of them, we will just include one of them in our feature set. We'll go with 'hospital region referral description' because it's not as particular as a provider's location or city, but it's also not as wide as a state.
- **Module 3**:
- In module 3 Development of android application to get the values from the cloud database and display it to the user. The data that will be utilized to answer the problem is one of the most significant aspects of machine learning difficulties. Data preparation accounts for around sixty to seventy percent of the overall time spent on a typical machine learning project. In order to get successful outcomes, it is critical to have the proper data for the situation at hand.

CHAPTER 7

SYSTEM DESIGN

7.1 HARDWARE MODULES DETAILS:

- Arduino Uno
- LCD Display (16x2)-1
- MQ-7   -1
- MQ-5
- Temperature sensor-1
- Humidity sensor
- Esp8266

7.1.1 ARDUINO UNO:

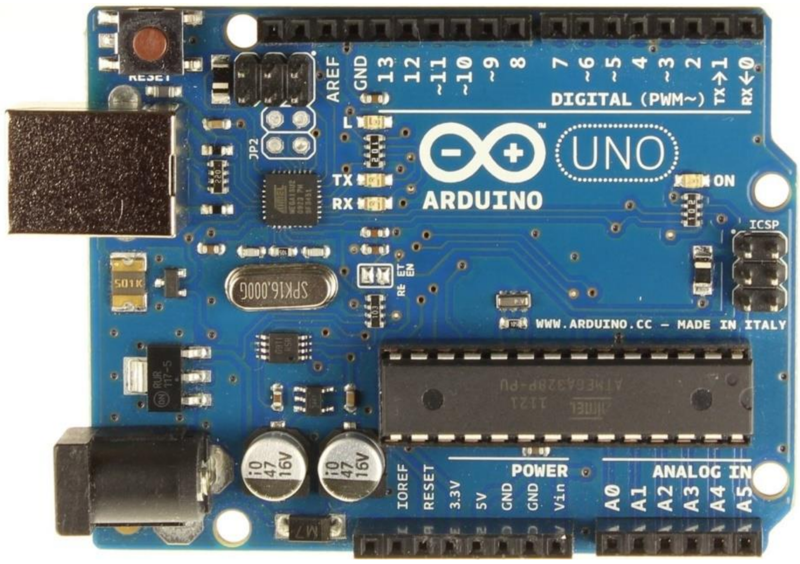


Fig 7.1

Specifications:

- The ATmega2560 is a Microcontroller.
- The operating voltage of this microcontroller is 5volts.
- The recommended Input Voltage will range from 7volts to 12volts.
- The input voltage will range from 6volts to 20volts.

- The digital input/output pins are 54 where 15 of these pins will supply PWM o/p.
- Analog Input Pins are 16.
- DC Current for each input/output pin is 40 mA.
- DC Current used for 3.3V Pin is 50 mA.
- The static random access memory (SRAM) is 8 KB.
- The electrically erasable programmable read-only memory (EEPROM) is 4 KB.
- The length of this board is 101.52 mm.
- The width of this board is 53.3 mm.
- The weight of this board is 36 g.

Key Benefits:

- Low cost
- Consistent board format
- 10x faster processing
- Added connectivity.

7.2 ARDUINO UNO

1. Introduction

1.1 Overview

The Arduino uno is a microcontroller board based on the ATmega1280 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 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. The uno is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: arduino-mega-reference-design.zip

Schematic: arduino-mega-schematic.pdf.

Summary

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

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

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm 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 than 7V, however, the 5V pin may supply less than five volts and the board may be unstable.

If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions.

Communication

The Arduino uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega1280 provides four hardware UARTs for TTL (5V) serial communication. An FTDI FT232RL on the board channels one of these over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the uno's digital pins.

The ATmega1280 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega1280 datasheet.

Programming

The Arduino uno can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The ATmega1280 on the Arduino uno comes preburned with a bootloader 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 bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

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 the FT232RL is connected to the reset line of the ATmega1280 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-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 a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to 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; see this forum thread for details.

7.1.2 ESP8266:

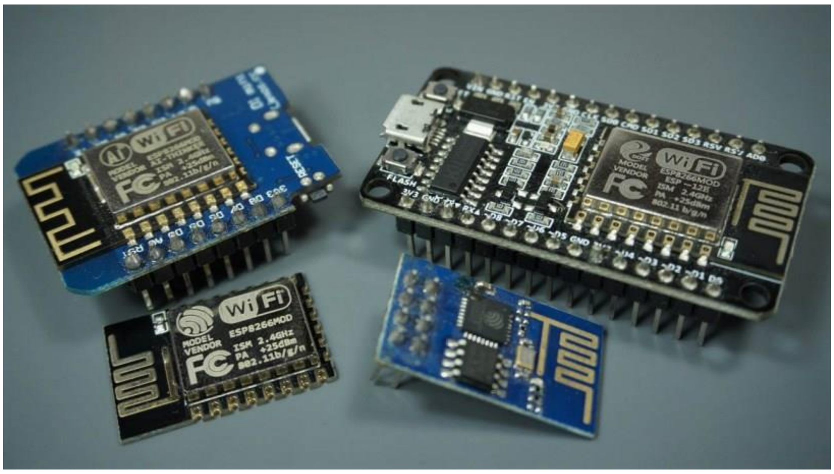


Fig 7.2

The ESP8266 12-E chip comes with 17 GPIO pins. Not all GPIOs are exposed in all ESP8266 development boards, some GPIOs are not recommended to use, and others have very specific functions. With this guide, you’ll learn how to properly use the ESP8266 GPIOs and avoid hours of frustration by using the most suitable pins for your projects.

ESP8266 Peripherals

The ESP8266 peripherals include:

- 17 GPIOs
- SPI
- I2C (implemented on software)
- I2S interfaces with DMA
- UART
- 10-bit ADC

Best Pins to Use – ESP8266

One important thing to notice about ESP8266 is that the GPIO number doesn't match the label on the board silkscreen. For example, D0 corresponds to GPIO16 and D1 corresponds to GPIO5.

The following table shows the correspondence between the labels on the silkscreen and the GPIO number as well as what pins are the best to use in your projects, and which ones you need to be cautious.

The pins highlighted in green are OK to use. The ones highlighted in yellow are OK to use, but you need to pay attention because they may have unexpected behavior mainly at boot. The pins highlighted in red are not recommended to use as inputs or outputs.

Analog Input

The ESP8266 only supports analog reading in one GPIO. That GPIO is called **ADC0** and it is usually marked on the silkscreen as **A0**.
The maximum input voltage of the ADC0 pin is 0 to 1V if you're using the ESP8266 bare chip. If you're using a development board like the ESP8266 12-E NodeMCU kit, the voltage input range is 0 to 3.3V because these boards contain an internal voltage divider.

You can learn how to use analog reading with the ESP8266 with the following guide.

On-board LED

Most of the ESP8266 development boards have a built-in LED. This LED is usually connected to GPIO2.

7.1.3 GAS SENSOR:

In current technology scenario, monitoring of gases produced is very important. From home appliances such as air conditioners to electric chimneys and safety systems at industries monitoring of gases is very crucial. **Gas sensors** are very important part of such systems. Small like a nose, gas sensors spontaneously react to the gas present, thus keeping the system updated about any alterations that occur in the concentration of molecules at gaseous state.

Fig 7.3

The **gas sensor module** consists of a steel exoskeleton under which a sensing element is housed. This sensing element is subjected to current through connecting leads. This current is known as heating current through it, the gases coming close to the sensing element get ionized and are absorbed by the sensing element. This changes the resistance of the sensing element which alters the value of the current going out of it.

7.1.4 Power Supply:

Alternating current (AC) is used for power line transmission and for high power devices like appliances and lights. The characteristics of AC make it ideal for transmission over long lines and for delivering large amounts of power for relatively unregulated uses, such as generating heat and light. Lower power appliances and devices require the closely regulated control of direct current power (DC). As a normal house is supplied with AC, it must be converted to DC for many uses. Use these tips to learn how to make an AC DC converter.

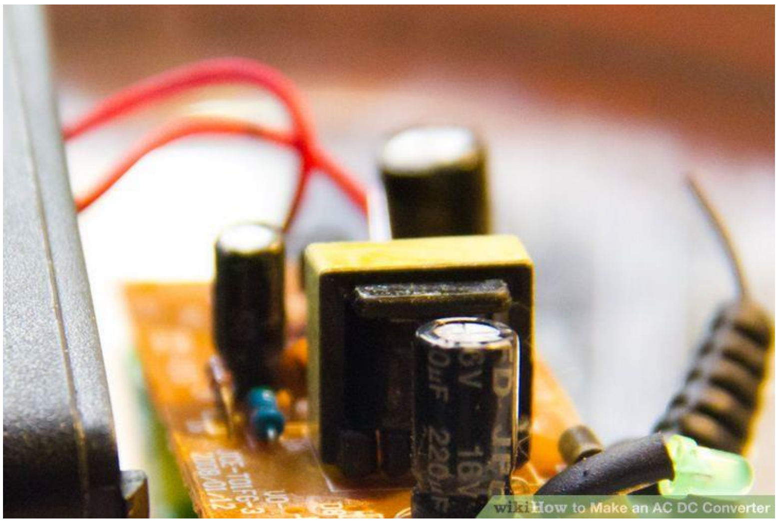
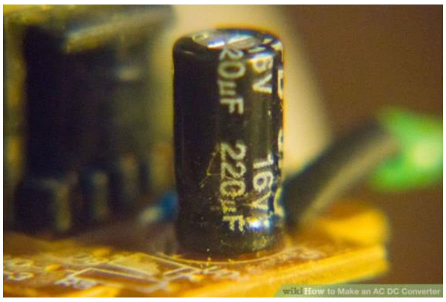


Fig 7.4

A transformer contains 2 magnetically coupled wire windings. One winding is called the primary. The primary is driven by the main AC supply. The other winding is called the secondary. The secondary serves as the power input to the AC DC converter. This transformer and all of the other items needed to build the AC DC converter are readily available at electronic stores and hobby stores.

- Size the transformer windings. AC mains provide 120 volts AC. If 120 volts AC were directly converted to a DC voltage, the resulting DC voltage would be far too high a voltage for use by appliances and devices. The primary and secondary windings of the transformer are scaled to each other in order to produce a lower voltage on the secondary winding.
- Choose a secondary winding. The AC output of the secondary winding should be rated as the same voltage of DC that is being created.



Attach a smoothing capacitor. Attach a polarized capacitor across the output connections of the rectifier. The positive terminal of the polarized capacitor must connect to the positive output of the regulator. This capacitor should be sized such that the capacitance in farads (F) is equal to (5 times the current to be supplied by the AC DC converter) divided by (transformer secondary rating times 1.4 times frequency). Frequency varies from country to country, but is typically either 50 Hertz (Hz) or 60 Hertz.

Capacitor

Image:

Fig 7.5

Capacitor is a passive component used to store charge. The charge (q) stored in a capacitor is the product of its capacitance (C) value and the voltage (V) applied to it. Capacitors offer infinite reactance to zero frequency so they are used for blocking DC components or bypassing the AC signals. The capacitor undergoes through a recursive cycle of charging and discharging in AC circuits where the voltage and current across it depends on the RC time constant. For this reason, capacitors are used for smoothing power supply variations.

Other uses include, coupling the various stages of audio system, tuning in radio circuits etc. These are used to store energy like in a camera flash.

Capacitors may be non-polarized/polarized and fixed/variable. Electrolytic capacitors are polarized while ceramic and paper capacitors are examples of non polarized capacitors. Since capacitors store charge, they must be carefully discharged before troubleshooting the circuits. The maximum voltage rating of the capacitors used must always be greater than the supply voltage. Click to learn more about working of a capacitor along with its internal structure.

Pin Diagram:

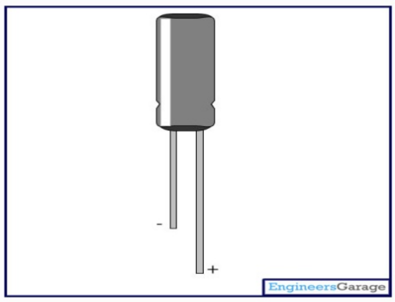


Fig 7.5.2

Capacitor is a widely used electronic component. It stores electric charge and then discharges it into the circuit. It blocks the direct current and allows the alternating current to pass through it. Depending on the purpose, there are a variety of capacitors being used like ceramic, electrolytic, mylar, mica, etc. We will explore an electrolytic capacitor through this article.

Structure of a Capacitor: A capacitor contains two conductor plates which are generally made of metal and an insulator between them. This insulator also known as dielectric is made up of material like paper, plastic, ceramic or glass. The two plates are electrically connected to the external circuit with the help of two thin metal rods also known as the legs of the capacitor.

These two plates are used to store charge between them. One is connected with positive voltage and other one with negative voltage. A capacitor is characterized by the parameter capacitance. Capacitance is measured as ratio of difference of charges between the plates and total voltage drop between the plates.

$C = dQ/dV$

The unit of capacitance is FARAD.

Let's have a closer look into the structure and how a capacitor is able to store charge.

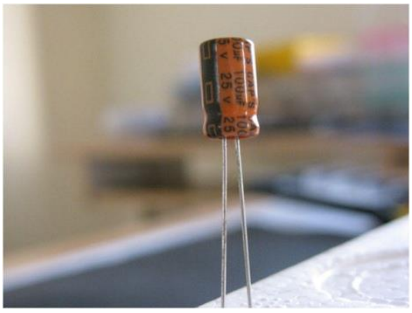


Fig 7.5.3

The above image shows a simple electrolytic capacitor with two thin rods coming out from the cylindrical container. The capacitor is wrapped up using a plastic covering. This covering is done to label the capacitor.

Metal Container

Inside the metal case is a folded layer of dielectric in between metal plates. Next images give a clear perception of the internal structure of the capacitor.

Dielectric

In the above image, we can clearly see two different types of layers, folded like a swiss roll. A dielectric layer is sandwiched between two metal plates. These metal plates are used to store charge and the dielectric works as an insulator between them. These plates are folded round to minimize the size of the capacitor.

One plate works as cathode and another as anode. To increase the value of a capacitor and the same time to keep the size smaller, we use electrolyte. However depending on the size and application, there are different types of electrolytes used in different ways within a capacitor. Generally, anode is soaked into liquid electrolyte to increase the surface area of the plate as well as efficiency.

Crystal Oscillator

A quartz crystal resonator plays a vital role in electronics oscillator circuitry. Sometimes mispronounced as crystal oscillator, it is rather a very important part of the feedback network of the oscillator circuitry. Electronics oscillators are used in frequency control application finding their usage in almost every industry ranging from small chips to aerospace.

A quartz crystal is the heart of such type of resonators. Their characteristics like high quality factor (Q), stability, small size and low cost make them superior over other resonators like LC circuit, turning forks, ceramic resonator etc.

The basic phenomenon behind working of a **quartz crystal oscillator** is the inverse piezo electric effect i.e., when electric field is applied across certain materials they start producing mechanical deformation. These mechanical deformation/movements are dependent on the elementary structure of the quartz crystal. Quartz is one of the naturally occurring materials which show the phenomena of piezo electricity, however for the purpose of resonator it is artificially developed since processing the naturally occurring quartz is difficult and costly process.

The image above shows a commonly used quartz crystal resonator. It is widely used in electronic oscillators circuitry used in digital circuits and microcontroller/processors.

7.1.5 TEMPERATURE SENSOR

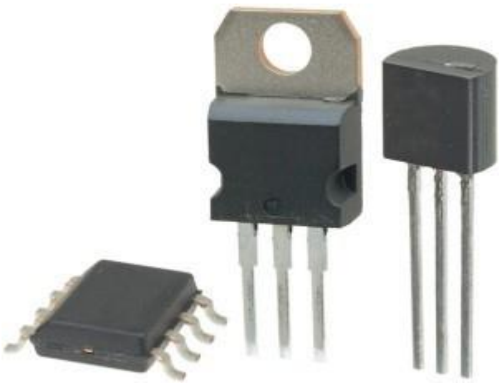


Fig 7.7

The most commonly measured physical parameter is temperature whether in process industry applications or in laboratory settings. Exact measurements are critical part of success. Exact measurements are needed for many applications such as medical applications, materials research in labs, studies of electronic or electrical components, biological research, and geological studies. Most commonly, temperature sensors are used to measure temperature in circuits which control a variety of equipment's.

There are different types of temperature sensors used in the market today including resistance temperature detectors (RTDs), thermocouples, thermistors, infrared sensor, and semiconductor sensors. Each of them has particular operating parameters. These sensors come in different varieties, but have one common thing: they all measure temperature by sensing a change in the physical characteristic.

A temperature sensor is a device, usually an RTD (resistance temperature detector) or a thermocouple, that collects the data about temperature from a particular source and converts the data into understandable form for a device or an observer. Temperature sensors are used in many applications like HVand AC system environmental controls, food processing units, medical devices, chemical handling and automotive under the hood monitoring and controlling systems, etc.

4.1.6 HUMIDITY SENSOR:

Sometimes, after rains, the air feels moist. The water seems to have suspended in the air. However, in certain AC's you click some buttons and the atmosphere brightens up. How and why does it all happen? Moisture forms up in the air, resulting in humidity. However, the humidity sensor in your AC picks it up and cleans it up for you. Isn't that wonderful? Let's take a look at how it does that.

A humidity sensor (or hygrometer) senses, measures and reports both moisture and air temperature. The ratio of moisture in the air to the highest amount of moisture at a particular air temperature is called relative humidity. Relative humidity becomes an important factor when looking for comfort. Humidity sensors work by detecting changes that alter electrical currents or temperature in the air.

- Capacitive
- Resistive
- Thermal

All three types of sensors monitor minute changes in the atmosphere in order to calculate the humidity in the air.

4.1.8 LCD 16X2:

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments),animations and so on.

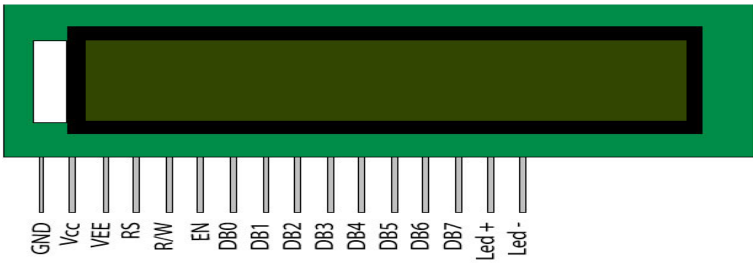


Fig 7.8

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

7.2 SOFTWARE REQUIREMENTS

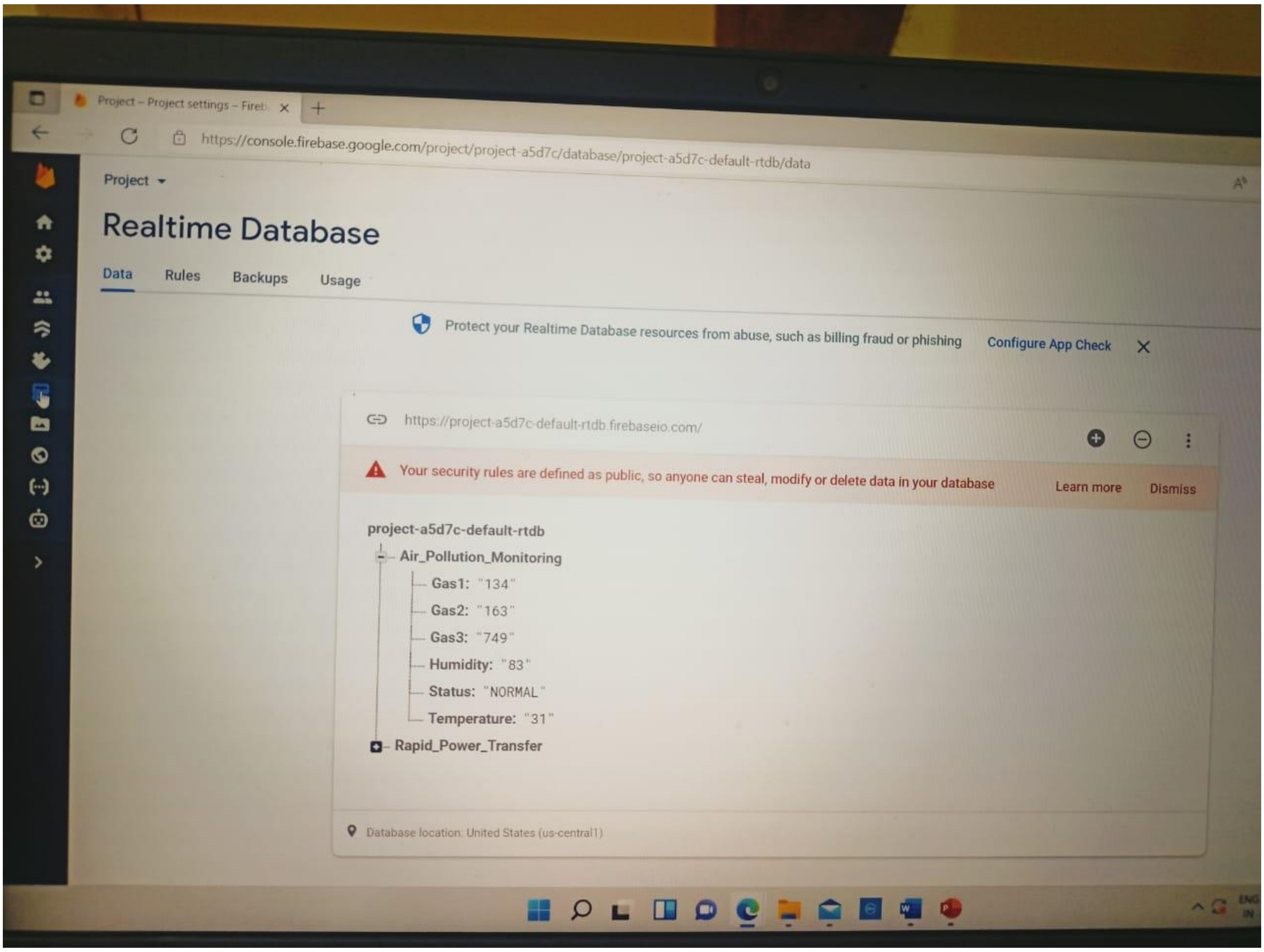
ARDUINO IDE:

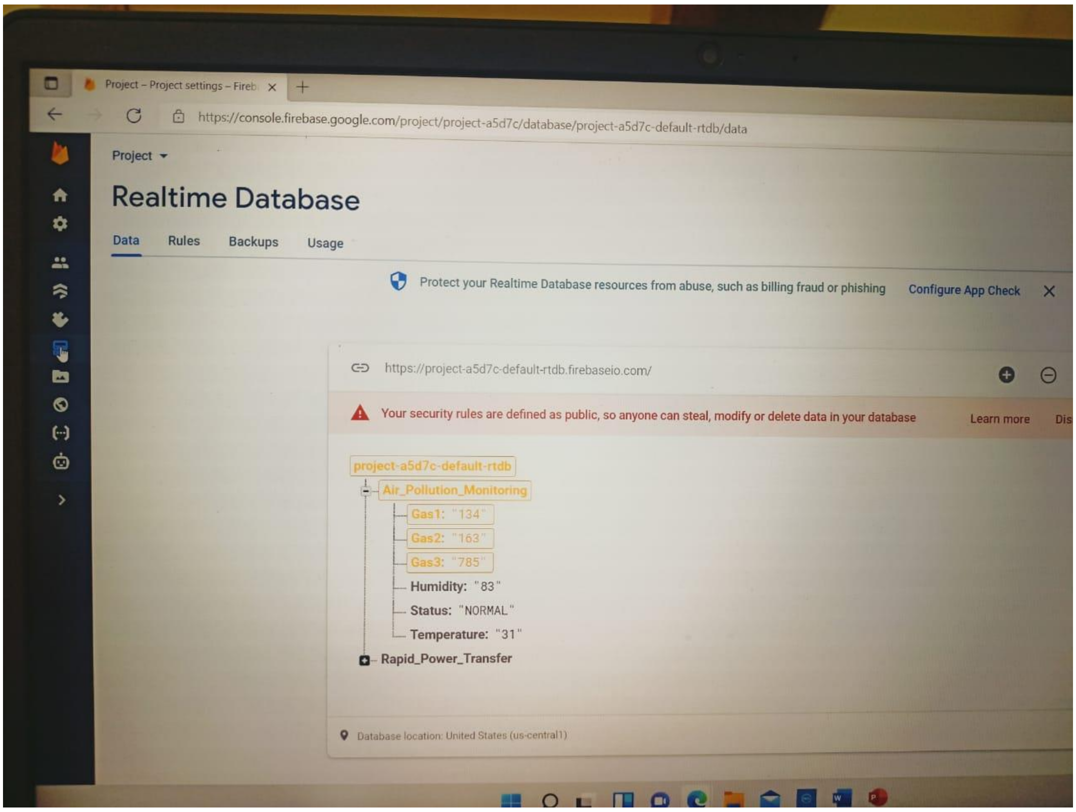
Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

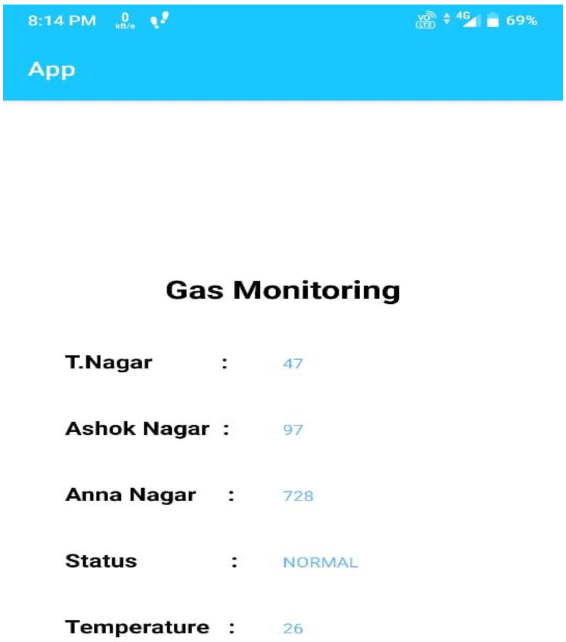
CHAPTER 8

OUTPUT

Firebase output:




Project – Project settings – Fireb
https://console.firebase.google.com/project/project-a5d7c/database/project-a5d7c-default-rtdb/data
Project
Realtime Database
Data
Rules
Backups
Usage
Protect your Realtime Database resources from abuse, such as billing fraud or phishing
Configure App Check
https://project-a5d7c-default-rtdb.firebaseio.com/
Your security rules are defined as public, so anyone can steal, modify or delete data in your database
Learn more
project-a5d7c-default-rtdb
Air_Pollution_Monitoring
Gas1: "134"
Gas2: "163"
Gas3: "785"
Humidity: "83"
Status: "NORMAL"
Temperature: "31"
Rapid_Power_Transfer
Database location: United States (us-central1)


8:14 PM
69%
App
Gas Monitoring
T.Nagar : 47
Ashok Nagar : 97
Anna Nagar : 728
Status : NORMAL
Temperature : 26

SAMPLE CODE:

FIREBASE DATABASE CODE:

```
#include"FirebaseESP8266.h"
#include<SoftwareSerial.h>
SoftwareSerial mySerial (D1, D2);
#include<ESP8266WiFi.h>
#define FIREBASE_HOST "https://project-a5d7c-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "Lw6uYACqlKmaNh8Kd1azkCjsvtKWdUnf6oCUCXCp"
#define WIFI_SSID "iotkit"
#define WIFI_PASS "12345678"
FirebaseData firebaseData;
int firebase;
int pin = D3;
String sensor_data;
String values;
String D;
void setup() {
  Serial.begin(115200);
  mySerial.begin(9600);
  pinMode(pin, OUTPUT);
  digitalWrite(pin, LOW);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.println("Wifi Connected ip: ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
```

```
  Firebase.reconnectWiFi(true);
  {
    Serial.println("reconnected");
  }
}
void loop() {
  bool Sr = false;
  while (mySerial.available()) {
    //get sensor data from serial put in sensor_data
    sensor_data = mySerial.readString();
    Sr = true;

  }

  delay(1000);
  if (Sr == true) {
    values = sensor_data;
    Serial.println(values);
    //get comma indexes from values variable
    int fristCommaIndex = values.indexOf(',');
    int secondCommaIndex = values.indexOf(',', fristCommaIndex + 1);
    int thirdCommaIndex = values.indexOf(',', secondCommaIndex + 1);
    int fourthCommaIndex = values.indexOf(',', thirdCommaIndex + 1);
    int fifthCommaIndex = values.indexOf(',', fourthCommaIndex + 1);

    //get sensors data from values variable by  spliting by commas and put in to variables
    String gas1 = values.substring(0, fristCommaIndex);
    String gas2 = values.substring(fristCommaIndex + 1, secondCommaIndex);
    String gas3 = values.substring(secondCommaIndex + 1, thirdCommaIndex);
    String t = values.substring(thirdCommaIndex + 1, fourthCommaIndex);
    String h = values.substring(fourthCommaIndex + 1);
    Serial.println(gas1);
```

```
    Serial.println(gas2);
    Serial.println(gas3);
    Serial.println(t);
    Serial.println(h);
    Serial.println("   ");

    if (gas1.toInt() > 900 || gas2.toInt() > 900 || gas3.toInt() > 900 || t.toInt() > 43)
    {
      D = "EMERGENCY";

    }
    else
    {
      D = "NORMAL";
    }
    Firebase.setString(firebaseData, "/Air_Pollution_Monitoring/Status", D);
    Firebase.setString(firebaseData, "/Air_Pollution_Monitoring/Gas1", gas1);
    Firebase.setString(firebaseData, "/Air_Pollution_Monitoring/Gas2", gas2);
    Firebase.setString(firebaseData, "/Air_Pollution_Monitoring/Gas3", gas3);
    Firebase.setString(firebaseData, "/Air_Pollution_Monitoring/Humidity", h.toInt());
    Firebase.setString(firebaseData, "/Air_Pollution_Monitoring/Temperature", t.toInt());
  }
}
```

UNO CODE:

```
#include <LiquidCrystal.h>
#include "DHT.h"
#define DHTPIN 8
// Digital pin connected to the DHT sensor
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
```

```
const int gasPin1 = A0;//(gas1 pin)
const int gasPin2 = A1;//(gas2 pin)
const int gasPin3 = A2;//(gas2 pin)
float hum, Temp;
String D;
String values;
void setup() {
  Serial.begin(9600);
  dht.begin();
  lcd.begin(16, 2);
}
void loop()
{
  int gas1 = analogRead(gasPin1);
  int gas2 = analogRead(gasPin2);
  int gas3 = analogRead(gasPin3);
  gas3 = random(700, 800);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("GAS1: ");
  lcd.print(gas1);
  //Serial.println("Gas1:"+String(gas1));
  lcd.setCursor(0, 1);
  lcd.print("GAS2: ");
  lcd.print(gas2);
  //Serial.println("Gas2:"+String(gas2));
  delay(1000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMP: ");
```

```
  lcd.print(t);
  //Serial.println("TEMP:"+String(Temp));
  lcd.setCursor(0, 1);
  lcd.print("HUMIDITY: ");
  lcd.print(h);
  //Serial.println("HUMIDITY:"+String(hum));
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Gas3: ");
  lcd.print(gas3);
  delay(1000);
  if (gas1 > 900 || gas2 > 900 || gas3 > 900 || Temp > 43)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("EMERGENCY");
    delay(1000);
  }
  else
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("NORMAL");
    delay(1000);
  }
  values = (String(gas1) + ',' + String(gas2) + ',' + String(gas3) + ',' + String(t) + ',' + String(h));
  Serial.println(values);
  delay(1000);


}
```

CHAPTER 9

CONCLUSION:

Reduce the level of pollution from such sources and to protect humans and the environment from harmful gasses, this air pollution kit was developed that helps a person to detect, monitor, and test air pollution in a given area . Pollution in earlier days was negligible. Currently, however, pollution is increasing day-by-day because of various reasons, such as industrial growth, development of automobile industries, and chemical industries. Therefore, to reduce the level of pollution from such sources and to protect humans and the environment from harmful gasses, this air pollution kit was developed that helps a person to detect, monitor, and test air pollution in a given area. The kit has been integrated with the mobile application IoT-Mobair that helps the user in predicting the pollution level of their entire route. Further, data logging can be used to predict AQI levels. This proposed air pollution monitoring kit along with the integrated mobile application can be helpful to people suffering from respiratory diseases. The app had following features, indices of air quality for a specific city using real-time computation, air quality daily forecasts, timing outdoor activities for different recommendation of generation, air quality dips related to health risks, specific reports for air quality measures based on locations, and air quality maps generation.

FUTURE WORK:

Our work can demonstrate vast opportunities to work on the device, on the app and also on the field using the device that we have worked with. The device can be used any time efficiently in different locations of a city and then research with the achieved data for that particular area in that city. The device can be updated with additional sensors that can sense data from the existence of other gases such as O_2 and H_2. These gases will provide the condition of the atmosphere and authority can take into further decisions accordingly. The sensors that we have been worked with can also be reset according to most recent time update. The android app which we have developed for turning on and off the device can be updated with newer features by implementing necessary codes. In future time, our device can be kept testing for checking whether the sensors still runs properly and give real time data. The webpage that we have designed, there is more opportunities to add options like related tables, pie chart, diagram that will be implemented by back-end programming(server side) so that those options can be visible to the administrator and user as well.

CHAPTER 10

10.1 Individual Objective

G. Lohith Krishna – Searching data of charges from hospitals and relatives

D. Arjun – Building a model training and testing

M. Abhishek– Testing different algorithms and getting output

10.2 Contribution of Team Members

In this project all 3 team members have contributed equally under the guidance of Dr. R. Magdalene

G. Lohith Krishna – Coordination with teammates and developing project and paper

D. Arjun – Building the he models and searching for suitable algorithms and paperwork.

M. Abhishek – Testing, output and pictures of paperwork .

REFERENCES:

[1] U. Varshney, Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring. New York, NY, USA: Springer, 2009.

[2] J. J. Caubel, T. E. Cados, and T. W. Kirchstetter, “A new black carbon sensor for dense air quality monitoring networks,” Sensors, vol. 18, no. 3, p. 738, 2018.

[3] L. Morawska et al., “Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone?” Environ. Int., vol. 116, pp. 286–299, Jul. 2018.

[4] D. Santi et al., “Seasonal variation of semen parameters correlates with environmental temperature and air pollution: A big data analysis over 6 years,” Environ. Pollut., vol. 235, pp. 806–813, Apr. 2018.

[5] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, “Field calibration of a cluster of low-cost commercially available sensors for air quality monitoring. Part B: NO, CO and CO2,” Sensors Actuators B Chem., vol. 238, pp. 706–715, Jan. 2017.

[6] N. Castell et al., “Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?” Environ. Int., vol. 99, pp. 293–302, Feb. 2017.

[7] S. Gaglio, G. L. Re, G. Martorella, D. Peri, and S. D. Vassallo, “Development of an IoT environmental monitoring application with a novel middleware for resource constrained devices,” in Proc. 2nd Conf. Mobile Inf. Technol. Med. (MobileMed), 2014, pp. 1–4.

[8] A. J. Cohen et al., “Estimates and 25-year trends of the global burden of disease attributable to ambient air pollution: An analysis of data from the global burden of diseases study 2015,” Lancet, vol. 389, no. 10082, pp. 1907–1918, 2017.

[9] G. L. Re, D. Peri, and S. D. Vassallo, “Urban air quality monitoring using vehicular sensor networks,” in Advances Onto the Internet of Things. Cham, Switzerland: Springer, 2014, pp. 311–323.

[10] R. Peterová and J. Hybler, “Do-it-yourself environmental sensing,” Procedia Comput. Sci., vol. 7, pp. 303–304, Dec. 2011.