

REAL TIME AIR QUALITY MONITORING SYSTEM

*A Project report submitted in partial fulfillment of the requirements for the award of
the degree of*

**BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING**

Submitted by

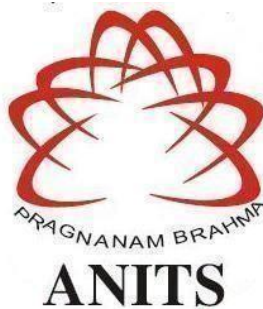
M. DEEPSHIKA (318126512152)

M.VIKRAM (318126512156)

S.GANESH(318126512169)

A. DEVA KALYAN(319126512L16)

**Under the guidance
Mr. N. RAM KUMAR,M.E.,(Ph.D)
Assistant Professor**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC
AUTONOMOUS)**

*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)
Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)2020-2021*

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC
AUTONOMOUS)


(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC
with 'A' Grade)


Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)



ANITS
CERTIFICATE

This is to certify that the project report entitled "REAL TIME AIR QUALITY MONITORING SYSTEM" submitted by M.Deepshika(318126512152), M.Vikram(318126512156), S.Ganesh(318126512169), A.Deva Kalyan(31912651216) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics & Communication Engineering of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.


Project Guide
Mr.N.Ram Kumar
Assistant Professor
Department of E.C.E
ANITS


Head of the Department
Dr. V.Rajyalakshmi
Professor & HOD
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department
Department of ECE
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa-531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mr.N.Ram Kumar**, Assistant professor ,Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very muchthankful to the **Principal and Management, ANITS, Sangivalasa**, for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS:

M.Deepshika (318126512152)

M.Vikram(318126512156)

S.Ganesh (318126512169)

A.Deva Kalyan(319126512L16)

ABSTRACT

Air pollution refers to the release of pollutants into the air—pollutants which are harmful to human health and the planet as a whole such as Carbon Monoxide(CO), Methane, Nitrous Oxide ,Carbon Dioxide(CO₂), Flourinated gases(F-gases) which as a whole effect the climatic changes. As the issue becomes more dominant it is constantly required to monitor these harmful gases and take necessary actions to eradicate this issue. This project presents the idea of detecting harmful gases in the environment and provide the data to a administrator.The main aim of this system is to achieve pollutants monitoring using wireless sensors connected to the Internet, which send the measurements to a centralized server. IOT allows the devices to send the location of the places that breached the threshold levels .Low power sensors are used to detect the parameters and interact with the microcontroller to process the data and transmit it through the internet using IOT .The ultimate goal of this project is to detect harmful gasses and send the data to an administrator to monitor the conditions. In this project we are targeting to build a system that can fetch the values of harmful pollutants present in that location and raise an alarm whenever the levels are breached, so that we can effectively monitor the changes and take necessary actions to normalize the pollutant levels. High levels of Carbon Monoxide intake can cause several health related issues like dull headache, weakness. dizziness. nausea or vomiting. shortness of breath, blurred vision .In the same way excess intake of nitrous oxide can cause dizziness, unconsciousness, and even death. Previously we have built the LPG detector using MQ6 sensor and Smoke detector using MQ2 sensor but this time we have used MQ135 sensor which is the best choice for monitoring Air Quality as it can detects most harmful gases and can measure their amount accurately. In this IOT project, you can monitor the pollution level from anywhere using your computer or mobile. We can install this system anywhere and can also trigger some device when pollution goes beyond some level, like we can switch on the Exhaust fan or can send alert SMS/mail to the user. Index Terms: Arduino, Gas sensor, Buzzer.

CONTENTS

ABSTRACT	iv
LIST OF FIGURES	viii
LIST OF TABLES	viii
CHAPTER 1 INTRODUCTION	01
1.1 Purpose	02
1.2 Project Objective	03
CHAPTER 2 LITERATURE SURVEY	05
2.1 Literature Review	05
2.2 Problem Statement	05
2.3 Aim	05
2.4 Existing System	05
2.5 Proposed System	06
CHAPTER 3 HARDWARE IMPLEMENTATION	07
3.1 Block Diagram Description	07
3.2 Program for Arduino in a few simple steps	08
3.3 Arduino UNO	09
3.3.1 Arduino pin Diagram	10
3.3.2 Pin description	11
3.3.3 Applications	17
3.4 MQ135 Multi gas sensor	18
3.4.1 Pin configuration	18

3.4.2 Specifications	19
3.4.3 Features	19
3.4.4 Applications	22
3.5 Wi-Fi Module ESP8266	22
3.5.1 Specifications	23
3.5.2 Pin out	24
3.5.3 Applications	24
3.6 Liquid Crystal Display	25
3.6.1 Features	26
3.6.2 Pin Diagram	27
3.6.3 Advantages	30
3.6.4 Disadvantages	30
3.7 DHT11 Sensor	31
3.7.1 Working Principle	31
3.7.2 Pin Identification & Configuration	32
3.7.3 Applications	33
3.8 Buzzer	33
3.8.1 Buzzer Driver	34
3.9 Power Supply	35
3.9.1 Description	36
3.9.2 Circuit Diagram	36
3.10 Software Implementation	39
3.10.1 Arduino IDE	39
3.11 System Design and Implementation	42

3.11.1 System Design	42
3.11.2 Circuit Diagram	43
3.11.3 System Implementation	44
CHAPTER 4 RESULT	47
APPLICATIONS AND ADVANTAGES	50
CHAPTER 5 CONCLUSION	51
REFERENCES	52
APPENDIX:SOURCE CODE	53

LIST OF FIGURES

Fig. 3.1	Block Diagram Description	07
Fig. 3.3.1	Arduino pin Diagram.	10
Fig. 3.3.2	Arduino Layouts	15
Fig. 3.4	MQ135 gas sensors	18
Fig. 3.4.1	WiFi module pinout	22
Fig. 3.6.1	LCD Display	27
Fig. 3.7	DHT11 Sensor	31
Fig. 3.8.1	Buzzer	34
Fig. 3.8.2	Buzzer driver	34
Fig.3.9	Block Diagram of power supply	35
Fig.3.9.1	Circuit Diagram of power supply	36
Fig. 3.9.2	Circuit Diagram	36
Fig. 4.1	Hardware Kit	47
Fig. 4.1.1	Display of output in Thingspeak	49

LIST OF TABLES

Pin Description of Arduino UNO	11
Pin Configuration of MQ135	18
Pin diagram of LCD	27
Pin identification of DHT11	32

CHAPTER -1

INTRODUCTION

Air pollution is the biggest problem of every nation, whether it is developed or developing. Health problems have been growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of lot of gaseous pollutants. Harmful effects of pollution include mild allergic reactions such as irritation of the throat, eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma. According to a survey, due to air pollution 50,000 to 100,000 premature deaths per year occur in the U.S. alone. Whereas in EU number reaches to 300,000 and over 3,000,000 worldwide.

Here to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air quality over a web server using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO₂, smoke, alcohol ,benzene and NH₃. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily.

Previously we have built the LPG detector using MQ6 sensor and Smoke detector using MQ2 sensor but this time we have used MQ135 sensor which is the best choice for monitoring Air Quality as it can detects most harmful gases and can measure their amount accurately. In this IOT project, you can monitor the pollution level from anywhere using your computer or mobile. We can install this system anywhere and can also trigger some device when pollution goes beyond some level, like we can switch on the Exhaust fan or can send alert SMS or mail to the user.

PURPOSE:

Air pollution occurs when harmful substances including particulates and biological molecules are introduced into Earth's atmosphere. It may cause diseases, allergies or death in humans; it may also cause harm to other living organisms such as animals and food crops, and may damage the natural or built environment. Human activity and natural processes can both generate air pollution. We propose an air quality as well as sound pollution monitoring system that allows us to monitor and check live air quality as well as sound pollution in an area through IOT.

System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data. Also, system keeps measuring sound level and reports it. The sensors interact with arduino uno which processes this data and transmits it over the application. This allows authorities to monitor air pollution in different areas and act against it. Also, authorities can keep a watch on the noise pollution near schools, hospitals and no honking areas.

Air pollution is one of the most common issues in the world. Severe Health issues have been recorded in the past few years due to urbanization and growing number of automobiles which release a lot of air pollutants. The harm caused by these pollutants includes many allergic conditions to human health such as irritation in the throat, eyes and nose and also some serious problems like bronchitis, heart diseases, pneumonia, lung and asthma.

Our system mainly focusses on monitoring the harmful pollutants in air using wireless gas sensors and Wi-Fi module which helps in monitoring the results. Air pollution is growing day-by-day and it is high time that we consider this as an issue and find a solution.

PROJECT OBJECTIVE:

Nowadays the air condition is very polluted. In recent years, car emissions, chemicals from factories, smoke, and dust are everywhere. That is the reason why now air condition is very polluted. The effect of air pollution is very bad for our health, especially for a place where the air in our body is taken for breathing.

Air pollution cannot be detected by human feelings. Air pollution may contain a lot of dangerous substances such as ozone, particulate matter sulfur dioxide, nitrogen dioxide, carbon

monoxide, and lead. This proposed system uses a wireless sensor network with low-cost sensors and hardware components along the necessary software to effectively monitor the air pollution phenomenon. Air pollution is the worst environmental problem and it causes a multitude of adverse effects on human health, water bodies, and climate. The main source of air pollution in all major cities is due to vehicles and the second major source remains the industries. The air pollution monitoring system

is installed in a particular locality where there are traces of acute air pollution to detect the constituent gases of air which may lead to harmful effects on human health and other leaving beings.

Substances in the polluted air are very dangerous. For example, if the carbon monoxide is above 100ppm, it makes humans feel dizzy, nauseous, and within minutes they could die. This research makes humans find out which content of the air is polluted. With module node MCU

esp8266, we can monitor the air pollution remotely, because there is a Wi-Fi module in node MCU esp8266. This makes the air condition can be monitored every time.

Air is getting polluted because of release of toxic gases by industries, vehicle emissions and increased concentration of harmful gases and particulate matter in the atmosphere. The level of pollution is increasing rapidly due to factors like industries, urbanization, increasing in population, vehicle use which can affect human health. Particulate matter is one of the most important parameter having the significant contribution to the increase in air pollution.

This creates a need for measurement and analysis of real time air quality monitoring so that appropriate decisions can be taken in a timely period. This paper presents a real-time standalone air quality monitoring. Internet of Things is nowadays finding profound use in each and every sector, plays a key role in our air quality monitoring system too. The setup will show the air quality in PPM in webpage so that we can monitor it very easily. In this IoT project, you can monitor the pollution level from anywhere using your computer or mobile [1]. The setup will show the air quality in PPM in webpage so that we can monitor it very easily. In this IoT project, you can monitor the pollution level from anywhere using your computer or mobile. Air condition is much polluted. In recent years, car emissions, chemicals from factories, smoke and dust are everywhere. That is the reason why now air condition is much polluted. The effect of air pollution is very bad for our health, especially for place where the air in our body is taken for breathing. In our lungs may cause some diseases, such as asthma, cough, lung disorders [1]. The air pollution cannot be detected by human feelings. The air pollution may contain a lot of dangerous substances, such as LPG gas, carbon monoxide, and methane [2]. Substances in the polluted air are very dangerous. For example, if the carbon monoxide is above 100ppm, it makes human feel dizzy, nauseous, and within minutes they could die. This research makes human find out which content of the air is polluted. With module node mcu esp8266, we can monitor the air pollution remotely, because there is a Wi-Fi in nodemcu esp8266. This makes the air condition can be monitored every time.

CHAPTER -2

LITERATURE SURVEY

2.1 Literature Review

The drawbacks of the conventional monitoring instruments are their large size, heavy weight and extraordinary expensiveness. These lead to sparse deployment of the monitoring stations. In order to be effective, the locations of the monitoring stations need careful placement because the air pollution situation in urban areas is highly related to human activities (e.g. construction activities) and location-dependent (e.g., the traffic choke-points have much worse air quality than average).

OT Based Air Pollution Monitoring System monitors the Air Quality over a webserver using internet and will trigger an alarm when the air quality goes down beyond a certain level, means when there are amount of harmful gases present in the air like CO₂, smoke, alcohol, benzene, NH₃, NO_x and LPG. The system will show the air quality in PPM on the LCD and as well as on webpage so that it can be monitored very easily. Temperature and Humidity is detected and monitored in the system.

2.2 Problem statement

To know about which area contains more pollution and escape from that particular place easily.

2.3 Aim

To prevent, control and abate pollution of streams, wells, land and air to protect the environment from any degradation by effective monitoring and implementation of pollution.

2.4 EXISTING SYSTEM:

The commercial meters available in the market are Fluke CO-220 carbon monoxide meter for CO, Amprobe CO₂ meter for CO₂, ForbixSemicon LPG gas leakage sensor alarm for LPG leakage detection. The researchers in this field have proposed various air quality monitoring systems based on WSN, GSM and GIS. Now each technology

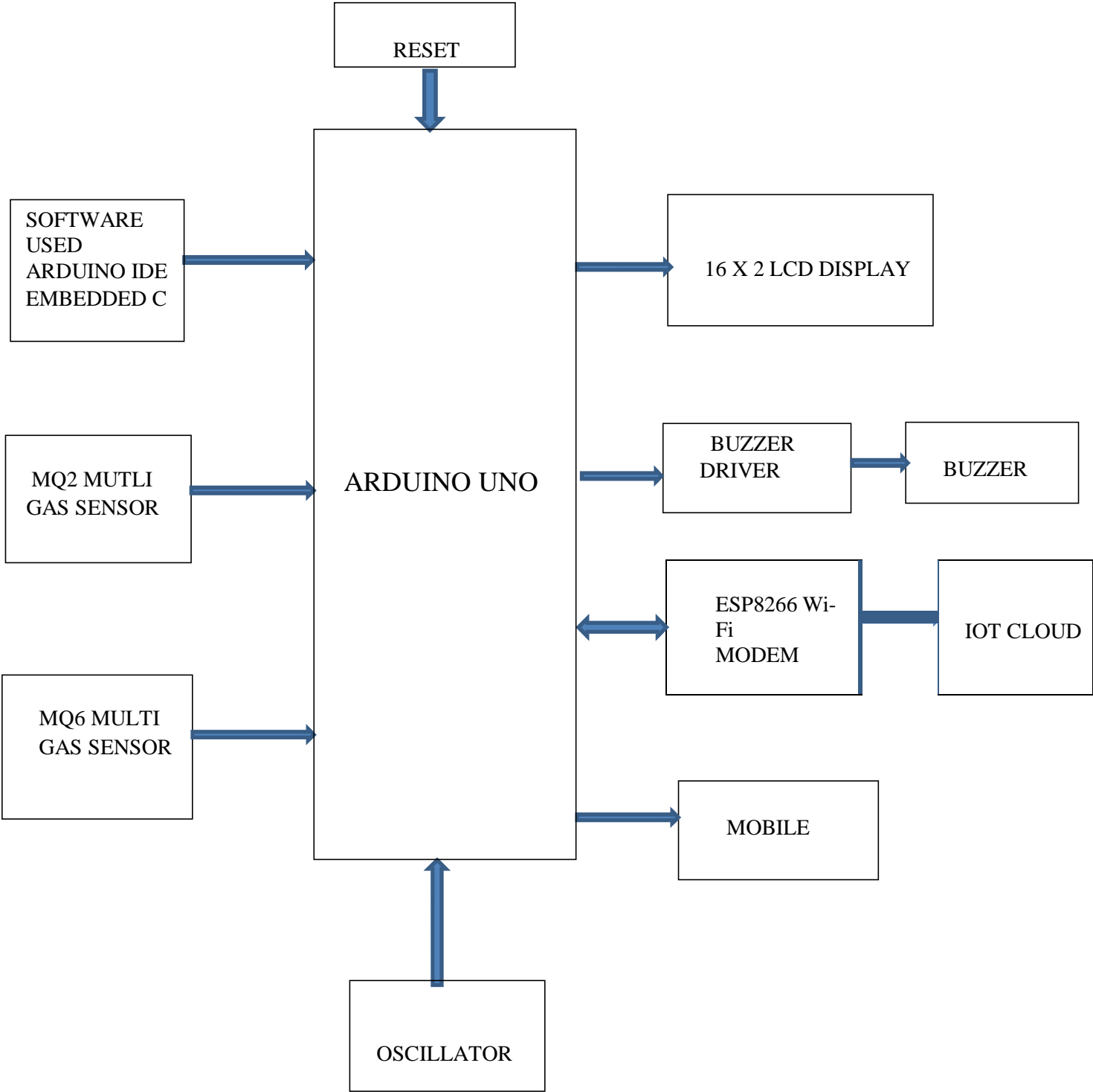
has limited uses according to the intended function, as Zigbee is meant for users with Zigbee-receiver, Bluetooth.

2.5 PROPOSED SYSTEM:

In this project we are going to make IOT based Air Pollution Monitoring System in which we will monitor the Air Quality over a web server using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO₂, smoke, alcohol, benzene and NH₃. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily.

CHAPTER 3:HARDWARE IMPLEMENTATION

3.1 Block diagram Description:



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

3.2 Program for Arduino In A Few Simple Steps:

STEP1:

Arduino microcontrollers come in a variety of types. The most common is the Arduino UNO, but there are specialized variations. Before you begin building, do a little research to figure out which version will be the most appropriate for your project.

STEP2:

To begin, you'll need to install the Arduino Programmer, aka the integrated development environment (IDE).

STEP3:

Connect your Arduino to the USB port of your computer. This may require a specific USB cable. Every Arduino has a different virtual serial-port address, so you'll need to reconfigure the port if you're using different Arduinos.

STEP 4:

Set the board type and the serial port in the Arduino Programmer.

STEP 5:

Test the microcontroller by using one of the preloaded programs, called sketches, in the Arduino Programmer. Open one of the example sketches, and press the upload button to load it. The Arduino should begin responding to the program: If you've set it to blink an LED light, for example, the light should start blinking.

STEP6:

To upload new code to the Arduino, either you'll need to have access to code you can paste into the programmer, or you'll have to write it yourself, using the Arduino programming language to create your own sketch. An Arduino sketch usually has five parts: a header describing the sketch and its author; a section defining variables; a setup routine that sets the initial conditions of variables and runs preliminary code; a loop routine, which is where you add the main code that will execute repeatedly until you stop running the sketch; and a section where you can list other functions that activate during the setup and loop routines. All sketches must include the setup and loop routines.

STEP7:

Once you've uploaded the new sketch to your Arduino, disconnect it from your computer and integrate it into your project as directed.

3.3 Arduino Uno

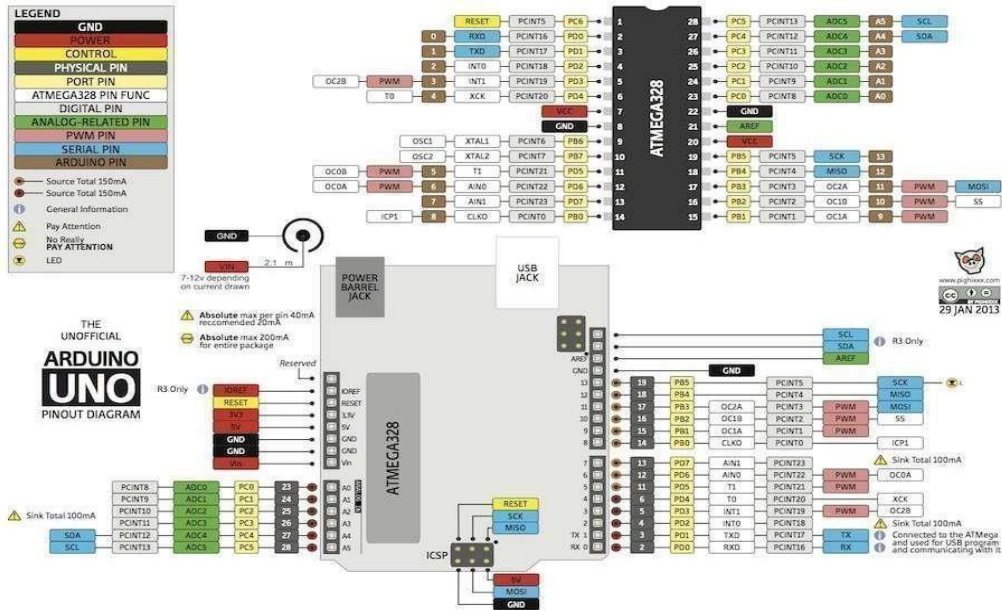
The Arduino Uno is a microcontroller board based on the ATmega328 . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the 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.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. The Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

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. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

3.3.1 Arduino pin Diagram:



3.3.2 Pin Description:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

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:

- **V I N** : 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**. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of 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.
- **3V3**. 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 micro-controller 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 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:

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 [attachInterrupt\(\)](#) function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` 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. 1024 different values). By default they measure from ground to 5 volts, though it is 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 micro-controller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the ATmega8, 168, and 328 is identical.

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 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 USB-to-serial chip and USB Connection to the computer (but not for serial communication on pin 0 and pin 1).

1).USB Overcurrent Protection:

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

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 ATmega8U2/16U2 is connected to the reset line of the ATmega328 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 bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

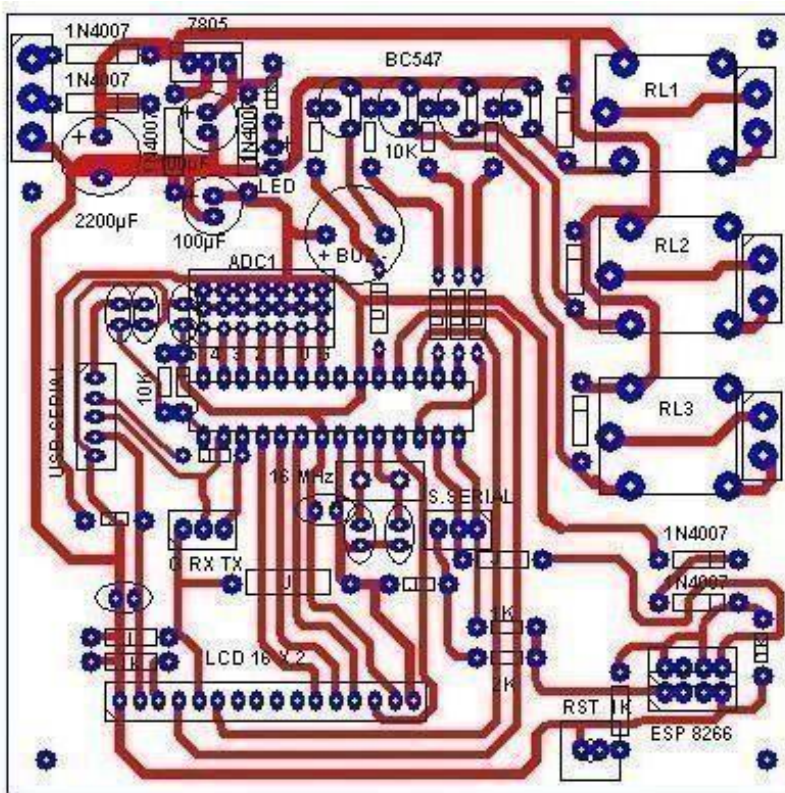
This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (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.

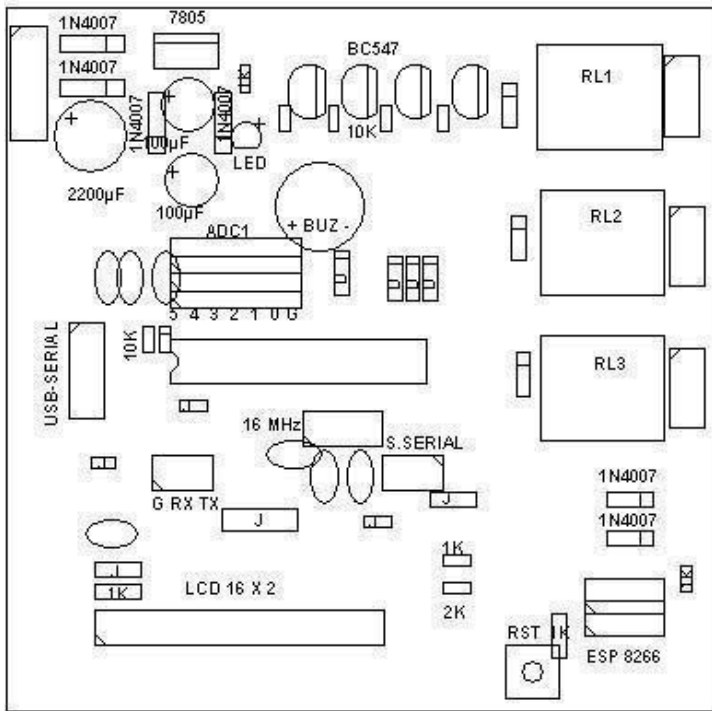
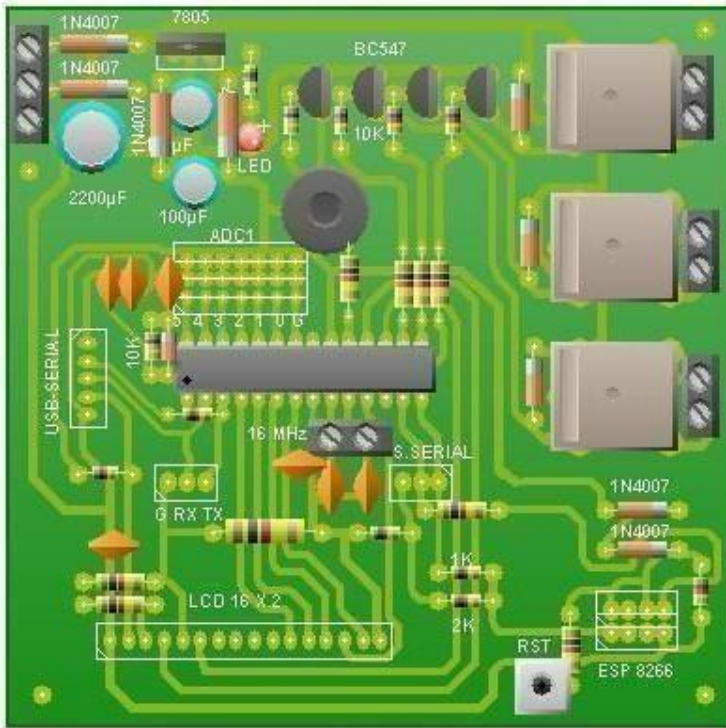
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 10 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

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.

ARDUINO LAYOUTS:





3.3.3 APPLICATIONS OF ARDUINO

UNO:

- Embedded system
- Robotics
- Motion control rig
- DC motor control (using H-bridge)
- Ardupilot(drone hardware and software)
- Defense and security defense
- Parking lot counter
- Home and industrial automation
- Game Duino(for creating retro 2D games)
- Water quality testing
- Data loggers (used in scientific research)
- Xoscillo(open-source oscilloscope)
- Count down timer for traffic lights

3.4 MQ135 Multi Gas

Sensors:

MQ135 Gas Sensor module for Air Quality having Digital as well as Analog output. Sensitive material of MQ135 gas sensor is SnO₂, which with lower conductivity in clean air. When the target combustible gas exist, The sensors conductivity is more higher along with the gas concentration rising. MQ135 gas sensor has high sensitivity to Ammonia, Sulphide and Benze steam, also sensitive to smoke and other harmful gases. It is with low cost and suitable for different application.

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

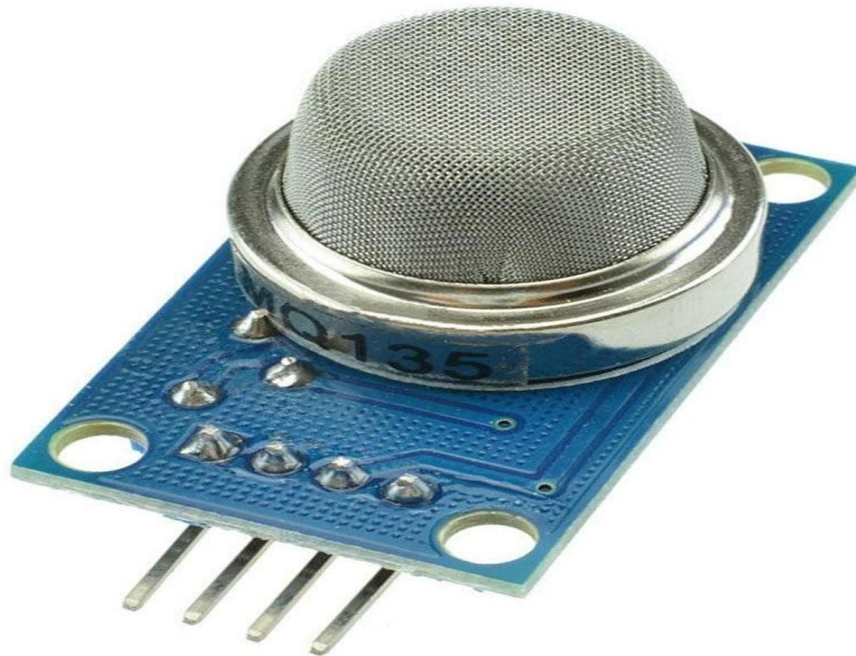


Figure:MQ135 Gas Sensor

3.4.1 Pin Configuration:

Pin	Pin Name	Description
1	Vcc	Used to power the sensor, Generally the operating voltage +5V.
2	Ground	Used to connect the module to system ground.
3	Digital Out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer.
4	Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas

For Sensor:

Pin	Pin Name	Description
1	H -Pins	Out of the two H pins, one pin is connected to supply and the other to ground
2	A-Pins	The A pins and B pins are interchangeable. These pins will be tied to the Supply voltage.
3	B-Pins	A pins and B pins are interchangeable. One pin will act as output while the other will be pulled to ground.

3.4.2 Specifications of MQ135 Gas Sensor Module:-

- Working voltage: DC 5V
- Working Current: 150mA
- DOUT: TTL output
- AOUT: Analog output
- Preheat time: Over 20s
- Dimension: 32mm x 22mm x 27mm (HIGH 27mm)

3.4.3 MQ-135 Sensor

Features:

- Wide detecting scope
- Fast response and High sensitivity
- Stable and long life
- Operating Voltage is +5V
- Detect/Measure NH₃, NO_x, alcohol, Benzene, smoke, CO₂, etc.
- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)

- Preheat duration 20 seconds
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

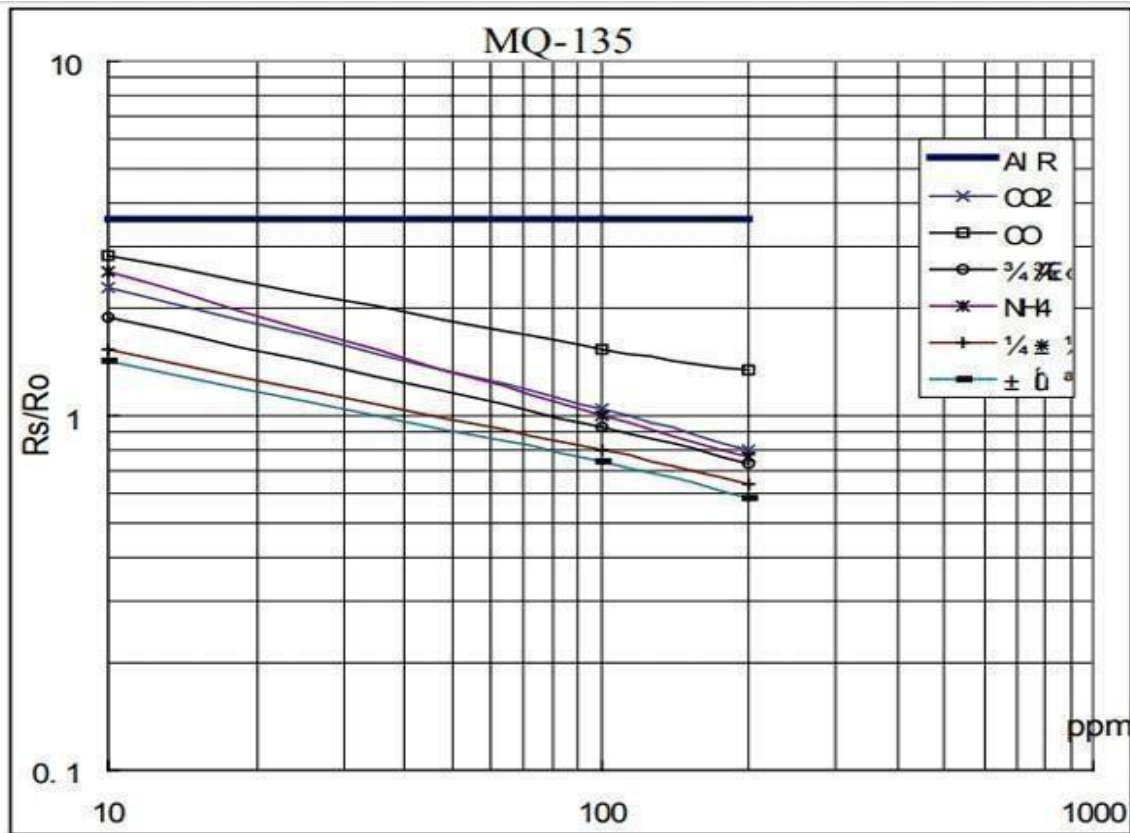
How to use MQ-135 Sensors to detect gases

You can either use the digital pin or the analog pin to do this. Simply power the module with 5V and you should notice the power LED on the module to glow and when no gas is detected the output LED will remain turned off meaning the digital output pin will be 0V. Remember that these sensors have to be kept on for pre-heating time (mentioned in features above) before you can actually work with it. Now, introduce the sensor to the gas you want to detect and you should see the output LED to go high along with the digital pin, if not use the potentiometer until the output gets high. Now every time your sensor gets introduced to this gas at this particular concentration the digital pin will go high (5V) else will remain low (0V).

You can also use the analog pin to achieve the same thing. Read the analog values (0- 5V) using a micro controller, this value will be directly proportional to the concentration of the gas to which the sensor detects. You can experiment with these values and check how the sensor reacts to different concentrations of gas and develop your program accordingly.

How to use MQ-135 sensor to measure PPM

MQ-135 gas sensor applies SnO₂ which has a higher resistance in clear air as a gas-sensing material. When there is an increase in polluting gases, the resistance of the gas sensor decreases along with that. To measure PPM using MQ-135 sensor we need to look into the (Rs/Ro) v/s PPM graph taken from the MQ135 datasheet.



The above figure shows the typical sensitivity characteristics of the MQ-135 for several gases. in their: Temp: 20, Humidity: 65%, O₂ concentration 21%, RL=20kΩ, Ro: sensor resistance at 100ppm of NH₃ in the clean air. Rs:sensor resistance at various concentrations of gases.

The value of Ro is the value of resistance in fresh air (or the air with we are comparing) and the value of Rs is the value of resistance in Gas concentration. First you should calibrate the sensor by finding the values of Ro in fresh air and then use that value to find Rs using the below formula:

$$\text{Resistance of sensor}(R_s): R_s=(V_c/V_{RL}-1)\times RL$$

Once we calculate Rs and Ro we can find the ratio and then using the graph shown above we can calculate The equivalent value of PPM for that particular gas.

3.4.4 Applications:

Used to detect leakage/excess of gases like Ammonia, nitrogen oxide, alcohols, aromatic compounds, sulfide and smoke.

Air quality monitors.

3.5 Wi-Fi module ESP8266:

ESP8266 is Wi-Fi enabled system on chip (SoC) module developed by Espressif system. It is mostly used for the development of the Internet of Things (IoT) embedded applications. The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by Shanghai-based Chinese manufacturing company Espressif Systems.

The ESP8266 is capable of either hosting an application or offloading all the Wi-Fi networking functions from another application processor. Each ESP8266 Wi-Fi module comes pre-programmed with an AT command set firmware, now you can simply hook this up to your Arduino device and get as much Wi-Fi ability as a Wi-Fi Shield offers.



Fig-3: Wifi module

3.5.1 ESP8266 wifi module Specification:

- 2.4 GHz Wi-Fi(802.11 b/g/n, supporting WPA/WPA2).
- General-purpose input/output (16 GPIO).

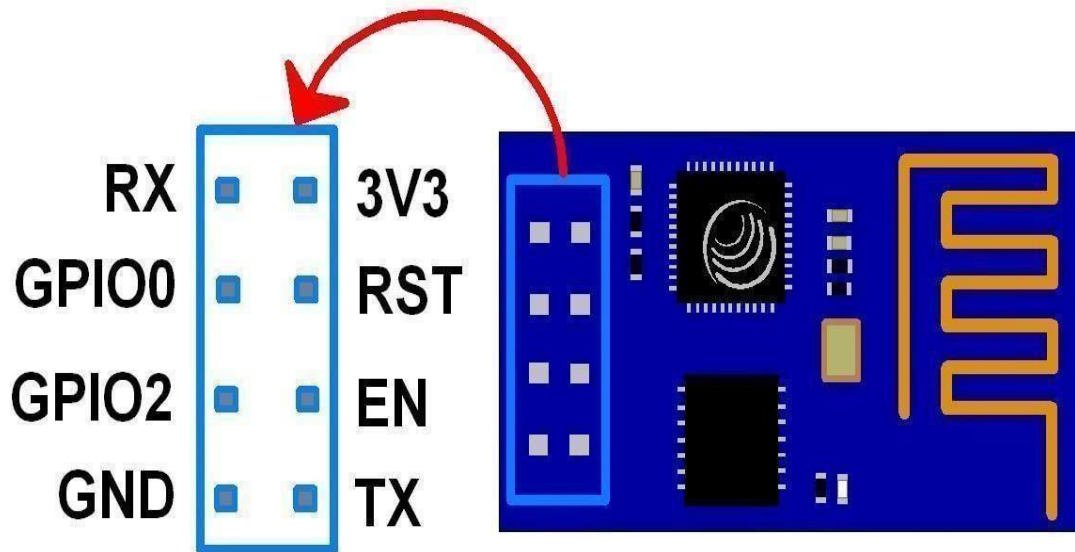
- Inter-Integrated Circuit (I²C) serial communication protocol.
- Analog-to-digital conversion (10-bit ADC).
- Serial Peripheral Interface (SPI) serial communication protocol
- I²S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO).
- UART (on dedicated pins, plus a transmit-only UART can be enabled onGPIO2).
- Pulse-width modulation (PWM).

It employs a 32-bit RISC CPU (reduced instruction set computer) based on the TensilicaXtensa L106 running at 80 MHz (or overclocked to 160 MHz).It has a 64 KB boot ROM, 64 KB instruction RAM, and 96 KB data RAM. External flash memory can be accessed through SPI.To communicate with the ESP8266 module, the microcontroller needs to use AT commands.The microcontroller communicates with the ESP8266-01 module using UART (Universal Asynchronous Receiver/Transmitter) having specified Baud rate.

There are many third-party manufacturers that produce different modules based on this chip (ESP8266). So, the module may come with different pin availability options like,

- ESP-01 comes with 8 pins (2 GPIO pins) with PCB (printed circuit board) trace antenna.
- ESP-02 comes with 8 pins, (3 GPIO pins) with a U-FL antenna connector.
- ESP-03 comes with 14 pins, (7 GPIO pins) with Ceramic antenna.
- ESP-04 comes with 14 pins, (7 GPIO pins) and without an antenna.

352 ESP8266 WiFi Module Pinout:



- 3V3: 3.3 V Power Pin.
- GND: Ground Pin.
- RST: Active Low Reset Pin.
- EN: Active High Enable Pin.
- TX: Serial Transmit Pin of UART.
- RX: – Serial Receive Pin of UART.
- GPIO0 & GPIO2: – General Purpose I/O Pins. It also known as TX/RX pins are used for Programming the module or for serial I/O purpose.

353 Applications:

- Wireless Web Server
- Geolocation using ESP8266
- Pressure Sensors on Railway Tracks
- Air Pollution Meter

- Temperature logging system
- World's smallest IoT project
- Wi-Fi controlled robot

3.6 LIQUID CRYSTAL DISPLAY:

In 1968, RCA Laboratories developed the first liquid crystal display (LCD). Since then, LCD's have been implemented on almost all types of digital devices, from watches to computer to projection TVs .LCD's operate as a light "valve", blocking light or allowing it to pass through. An image in an LCD is formed by applying an electric field to alter the chemical properties of each LCC (Liquid Crystal Cell) in the display in order to change a pixel's light absorption properties. These LCC's modify the image produced by the backlight into the screen output requested by the controller. Through the end output may be in color, the LCC's are monochrome, and the color is added later through a filtering process. Modern laptop computer displays can produce 65,536 simultaneous colors at resolution of 800 X 600.

To understand the operation of an LCD, it is easiest to trace the path of a light ray from the backlight to the user. The light source is usually located directly behind the LCD, and can use either LED or conventional fluorescent technology. From this source, the light ray will pass through a light polarizer to uniformly polarize the light so it can be acted upon by the liquid crystal(LC) matrix. The light beam will then pass through the LC matrix, which will determine whether this pixel should be "on" or "off". If the pixel is "on", the liquid crystal cell is electrically activated, and the molecules in the liquid will align in a single direction. This will allow the light to pass through unchanged. If the pixel is "off", the electric field is removed from the liquid, and the molecules will scatter. This dramatically reduces the light that will pass through the display at that pixel.

In a color display, after the light passes through the liquid crystal matrix, it passes through a color filter (usually glass). This filter blocks all wavelengths of light except those within the range of that pixel. In a typical RGB display, the color filter is integrated into the upper glass colored microscopically to render each individual pixel red, green or blue. The areas in between the colored pixel

filter areas are printed black to increase contrast. After a beam of light passes through the color filter, it passes through yet another polarizer to sharpen the image and eliminate glare. The image is then available for viewing.

In an AMLCD, each LCC is stimulated individually by a dedicated transistor or diode. The two existing AMLCD technologies are Thin Film Transistor (TFT) and metal-insulator-metal (MIM). In an MIM display, dedicated diodes are fabricated at each pixel. MIM displays, currently being manufactured by Toshiba and Seiko-Epson, are not advantageous that TFT displays.

INTERFACING LCD TO THE MICROCONTROLLER:

This is the first interfacing example for the parallel port. We will start with something simple. This example does not use the Bi-directional feature found on newer ports, thus it should work with most, if not all Parallel Ports. It however does not show the use of the status port as an input. So what are we interfacing? A 16 Character X 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required running them is on board.

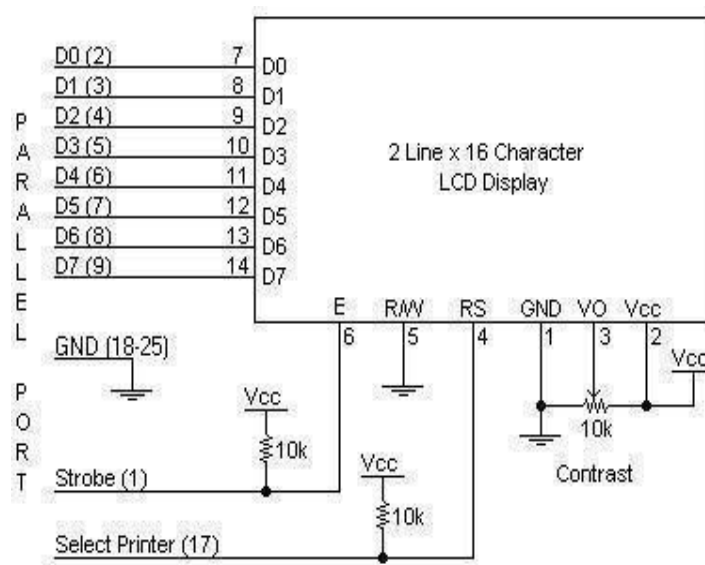
3.6.1 Features:

- Interface with either 4-bit or 8-bit microprocessor.
- Display data RAM
- 80 × 8 bits (80 characters).
- Character generator ROM
- 160 different 5 × 7 dot-matrix character patterns.
- Character generator RAM
- 8 different user programmed 5 × 7 dot-matrix patterns.
- Display data RAM and character generator RAM may be accessed by the microprocessor.
- Numerous instructions
- Clear Display, Cursor Home, Display ON/OFF, Cursor ON/OFF, Blink Character, Cursor Shift, Display Shift.
- Built-in reset circuit is triggered at power ON.



A general purpose alphanumeric LCD, with two lines of 16 characters

3.6.2 Pin diagram:



Pin Diagram of LCD

Pins Definition

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Pin Definition of LCD

In the above table Vcc and Vss are supply pins and VEE (Pin no.3) is used for controlling LCD contrast. Pin No.4 is Rs pin for selecting the register, there are two very important registers are there in side the LCD. The RS pin is used for their selection as follows. If RS=0, the instruction command code register is selected, allowing the user to send data to be displayed on the LCD. R/W is a read or writes Pin, which allows the user to write information to the LCD or read information from it. R/W=1 when reading R/W=0 when writing. The LCD to latch information presented to its data pins uses the enable (E) pin. The 8-bit data pins, D0-D7, are used to send information to the LCD or read the contents of the LCD's internal registers. To display letters and numbers, we must send ASCII codes for the letters A-Z, and number 0 -9 to these pins while making RS=1. So these are the pins which make as 1 and symbol A is for Power Supply for LED B/L and K is for the power supply for LED B/L and Rs is for instruction or data register select.

ABSOLUTE MAXIMUM RATINGS:

1. ELECTRICAL ABSOLUTE MAXIMUM RATINGS:

ITEM	SYMBOL	CONDITION	MIN	MAX	UNIT
Supply Voltage (Logic)	Vdd – Vss	-	0	7.0	V
Supply Voltage (LCD Drive)	Vdd – V0	-	0	13.0	V
Input Voltage	Vi	-	-0.3	Vdd +0.3	V

Quality control:

Some LCD panels have defective transistors, causing permanently lit or unlit pixels which are commonly referred to as stuck pixels or dead pixels respectively. Unlike integrated circuits (ICs), LCD panels with a few defective pixels are usually still usable. It is also economically prohibitive to discard a panel with just a few defective pixels because LCD panels are much larger than ICs.

Color displays:

In color LCDs each individual pixel is divided into three cells, or subpixels, which are colored red, green, and blue, respectively, by additional filters (pigment filters, dye filters and metal oxide filters). Each subpixel can be controlled independently to yield thousands or millions of possible colors for each pixel. CRT monitors employ a similar 'subpixel' structures via phosphors, although the analog electron beam employed in CRTs do not hit exact 'subpixels'.

Color components may be arrayed in various pixel geometries, depending on the monitor's usage. If software knows which type of geometry is being used in a given LCD, this can be used to increase the apparent resolution of the monitor through sub pixel rendering. This technique is especially useful for text anti-aliasing.

To reduce smudging in a moving picture when pixels do not respond quickly enough to color changes, so-called pixel overdrive may be used.

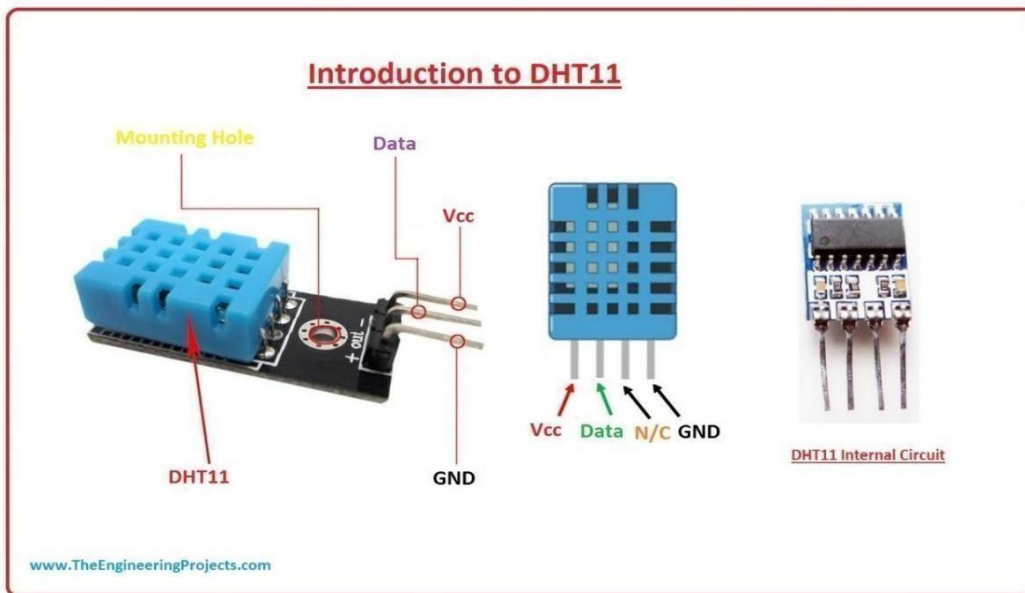
3.6.3 Advantages of an LCD's:

1. LCD's consumes less amount of power compared to CRT and LED.
2. LCD's are consist of some microwatts for display in comparison to some mill watts for LED's.
3. LCDs are of low cost.
4. Provides excellent contrast.
5. LCD's are thinner and lighter when compared to cathode ray tube and LED.
6. NO BURN-IN
7. Supports Low and small profilesizes
8. It doesnot affect by magnetic fields.

3.6.4 Disadvantages of an LCD's:

1. Require additional light sources
2. Range of temperature is limited for operation
3. Low reliability
4. Speed is very low
5. LCD's need an AC drive
6. Limited viweing angle and brightness
7. Consumes lot of electricity so produces moreheat
8. It has individually Liquid crystals which cant complete all blocks of blackLight.

3.7 DHT11 SENSOR



DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc... to measure humidity and temperature instantaneously. DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

3.7.1 Working Principle of DHT11 Sensor

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form. For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro- controller.

3.7.2 Pin Identification and Configuration:

For DHT11 Sensor

Pin No	Pin Name	Description
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	NC	No Connection and hence not used
4	Ground	Connected to the ground of the circuit

For DHT11 Sensor module

1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circu

Functionality of DHT11 Sensor

Applications

This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to predict weather conditions. The humidity sensor is used as a preventive measure in homes where people are affected by humidity. Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure. Its compact size and sampling rate made this sensor popular among hobbyists. Some of the sensors which can be used as an alternative to DHT11 sensor are DHT22, AM2302, SHT71.

BUZZER:

A buzzer or beeper is a signalling device, usually electronic, typically used in automobiles, household appliances such as a microwave oven, or game shows. It most commonly consists of a number of switches or sensors connected to a control unit that determines if and which button was pushed or a preset time has lapsed, and usually illuminates a light on the appropriate button or control panel, and sounds a warning in the form of a continuous or intermittent buzzing or beeping sound. Initially this device was based on an electromechanical system which was identical to an electric bell without the metal gong (which makes the ringing noise). Often these units were anchored to a wall or ceiling and used the ceiling or wall as a sounding board. Another implementation with some AC-connected devices was to implement a circuit to make the AC current into a noise loud enough to drive a loudspeaker and hook this circuit up to a cheap 8-ohm speaker. Now-a-days, it is more popular to use a ceramic-based piezo-electric sounder like a Sonalert which makes a high-pitched tone. Usually these were hooked up to driver" circuits which varied the pitch of the sound or pulsed the sound on and off.

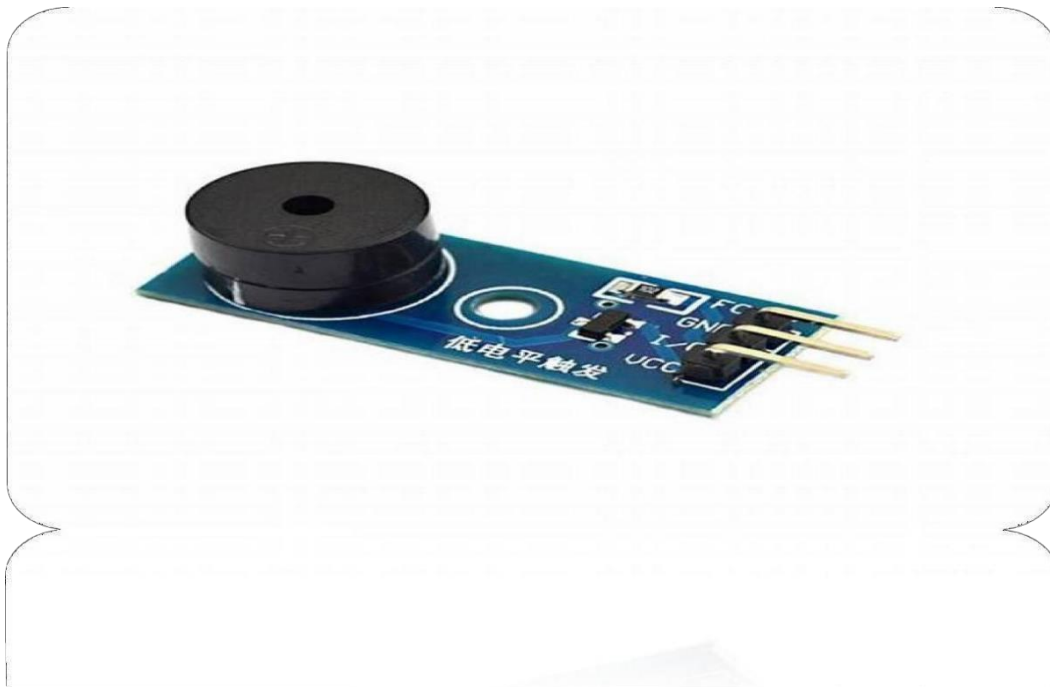
The specifications of the buzzer include the following.

- Color is black
- The frequency range is 3,300Hz
- The supply current is below 15mA

3.8.1 Buzzer Driver:

The circuit is designed to control the buzzer. The buzzer ON and OFF is controlled by the pair of switching transistors (BC 547). The buzzer is connected in the Q2 transistor collector terminal. When high pulse signal is given to base of the Q1 transistors, the transistor is conducting and close the collector and emitter terminal so zero signals is given to base of the Q2 transistor. Hence Q2 transistor and buzzer is turned OFF state.

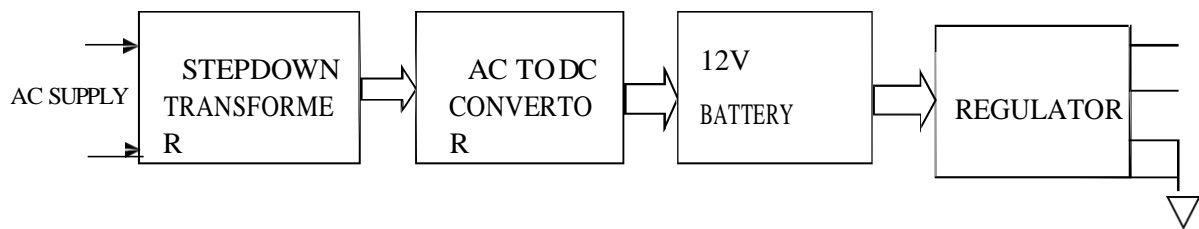
When low pulse is given to base of transistor Q1, the transistor is turned OFF. Now 12V is given to base of Q2 transistor so the transistor is conducting and buzzer is energized and produces the sound signal. They are either of the passive or of the active type. The former are driven by an AF signal source, while the latter feature a built-in oscillator, and require a direct voltage only. This circuit is a double AF oscillator for driving passive buzzers. It ensures a richer output sound than normally obtainable from a piezo buzzer due to the use of two oscillators, N1 and N2, whose output signal lies between 1 and 10 kHz. Gates N3-N4 form an S-R bistable which is controlled by the outputs of N1-N2, and drives the buzzer directly.



3.9 Power Supply:

The Power Supply is a Primary requirement for the project work. The required DC power supply for the base unit as well as for the recharging unit is derived from the mains line. For this purpose center tapped secondary of 12V-0-12V transformer is used. From this transformer we getting 5V power supply. In this +5V output is a regulated output and it is designed using 7805 positive voltage regulator. This is a 3 Pin voltage regulator, can deliver current up to 800 milliamps.

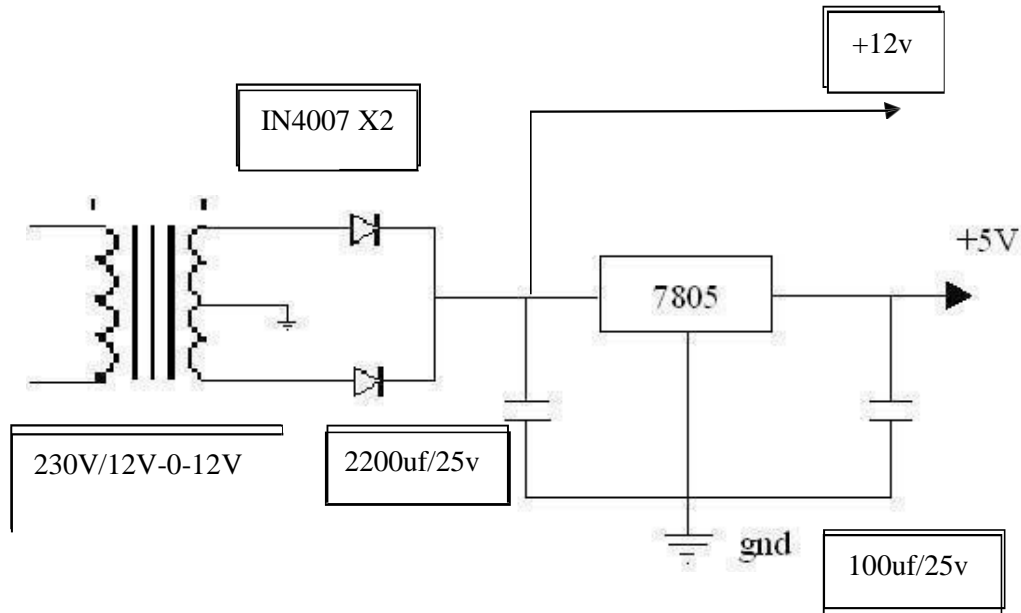
Rectification is a process of rendering an alternating current or voltage into a unidirectional one. The component used for rectification is called 'Rectifier'. A rectifier permits current to flow only during positive half cycles of the applied AC voltage. Thus, pulsating DC is obtained to obtain smooth DC power additional filter circuits required. A power supply is an electrical device that supplies electric power to an electrical load. The main purpose of a power supply is to convert electric current from a source to the correct voltage, current, and frequency to power the load. As a result, power supplies are sometimes referred to as electric power converters. Some power supplies are separate standalone pieces of equipment, while others are built into the load appliances that they power. Examples of the latter include power supplies found in desktop computers and consumer electronics devices. Other functions that power supplies may perform include limiting the current drawn by the load to safe levels, shutting off the current in the event of an electrical fault power conditioning to prevent electronic noise or voltage surges on the input from reaching the load, power-factor correction, and storing energy so it can continue to power the load in the event of a temporary interruption in the source power



3.9.2 Circuit diagram:

A diode can be used as rectifier. There are various types of diodes. However, semiconductor diodes are very popularly used as rectifiers. A semiconductor diode is a solid-state device consisting of two elements: one being an electron emitter or cathode, the other an electron collector or anode. Since electrons in a semiconductor diode can flow in one direction only—from emitter to collector—the diode provides the unilateral conduction necessary for rectification.

The rectified output is filtered for smoothing the DC, for this purpose a capacitor is used in the filter circuit. The filter capacitors are usually connected in parallel with the rectifier output and the load. The AC can pass through a capacitor but DC cannot, the ripples are thus limited and the output becomes smoothed. When the voltage across the capacitor plates tends to rise, it stores up energy back into voltage and current. Thus, the fluctuation in the output voltage is reduced considerably.



VOLTAGE REGULATR:

The LM 78XXX series of the three terminal regulations is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation. The voltages available allow these regulators to be used in logic systems, instrumentation and other solid state electronic equipment. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents. The LM78XX series is available in aluminum to 3 packages which will allow over 1.5A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. The LM 78XX is available in the metal 3 leads to 5 and the plastic to 92. For this type, with adequate heat sinking. The regulator can deliver 100mA output current.

The advantage of this type of regulator is, it is easy to use and minimize the number of external components.

The following are the features voltage regulators:

- Output current in excess of 1.5A for 78 and 78L series
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit.
- Available in aluminum 3 package.

POSITIVE VOLTAGE REGULATOR

The positive voltage regulator has different features like:

- Output current up to 1.5A
- No external components
- Internal thermal overload protection

- High power dissipation capability
- Internal short-circuit current limiting
- Output transistor safe area compensation
- Direct replacements for Fairchild microA7800s

Nominal Output Voltage	Regulator
5V	uA7805C
6V	uA7806C
8V	uA7808C
8.5V	uA7885C
10V	uA7810C
12V	uA7812C
15V	uA7815C
18V	uA7818C
24V	uA7824C

3.10 SOFTWARE IMPLEMENTATION:

3.10.1: ARDUINO IDE:

The Arduino IDE consists an editor for writing code, console area, and a series of buttons with inbuilt functionalities. The Arduino hardware is interfaced to the Arduino IDE with the help of cable to upload sketches and communicate with each other. Arduino is a software community, an open source platform, a user-friendly community that develops and promotes microcontroller established hardware kits for persuasive interaction with the world. The digital and analog pins assembled on the board gives the competence to associate with distinct shields. The GNU took the lead in releasing the Source code of the IDE. For the purpose of Programming and incorporating with the circuits the Arduino board has specialized pins available for the interfacing. The ATmega controllers consist of flash memory with a boot loader for uploading programs into it. The optiboot loader is the default loader for the Arduino boards. With the utilization of the cables the boards are loaded with the code. The sketch may be written in any programming language that has the capability in providing the binary code for the target machines. The Application code is written by the utilization of Java in the form of cross platform. It has a dedicated software library for wiring the modules or shields to the boards. In the body of the Arduino editor, specifically two functions play a predominant role they are the setup and loop functions. The setup body of code initializes the various variables, their pin modes, set of libraries. In general loop signifies the unending repetition of sketch.

The software tool used for implementing the software design is Arduino IDE. The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring

project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub `main()` into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution.

The Arduino IDE employs the program to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, it is used as the uploading tool to flash the user code onto official Arduino boards. With the rising popularity of Arduino as a software platform, other vendors started to implement custom open source compilers and tools (cores) that can build and upload sketches to other micro controllers that are not supported by Arduino's official line of micro-controllers. In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging and other advanced features.

Program For Arduino In A Few Simple Steps

STEP1:

Arduino microcontrollers come in a variety of types. The most common is the Arduino UNO, but there are specialized variations. Before you begin building, do a little research to figure out which version will be the most appropriate for your project.

STEP2:

To begin, you'll need to install the Arduino Programmer, aka the integrated development environment (IDE).

STEP3:

Connect your Arduino to the USB port of your computer. This may require a specific USB cable. Every Arduino has a different virtual serial-port address, so you'll need to reconfigure the port if you're using different Arduinos.

STEP4

Set the board type and the serial port in the Arduino Programmer.

STEP5:

Test the microcontroller by using one of the preloaded programs, called sketches, in the Arduino Programmer. Open one of the example sketches, and press the upload button to load it. The Arduino should begin responding to the program: If you've set it to blink an LED light, for example, the light should start blinking.

STEP6:

To upload new code to the Arduino, either you'll need to have access to code you can paste into the programmer, or you'll have to write it yourself, using the Arduino programming language to create your own sketch. An Arduino sketch usually has five parts: a header describing the sketch and its author; a section defining variables; a setup routine that sets the initial conditions of variables and runs preliminary code; a loop routine, which is where you add the main code that will execute repeatedly until you stop running the sketch; and a section where you can list other functions that activate during the setup and loop routines. All sketches must include the setup and loop routines.

STEP7:

Once you've uploaded the new sketch to your Arduino, disconnect it from your computer and integrate it into your project as directed.

Either a website or a mobile app is required in order to display the reading of the gas sensors. So that we have ThingSpeak as a website which displays the information in the form of graphical representations of the values for every 2 mins of the sensors readings and we have to connect the ssid key and password to the NodeMCU which helps to transmit the data to the ThingSpeak Server. ThingSpeak Server contains the private and public mode in order to make people view the particular gases in that particular area then the administrator has to keep the view as public. ThingSpeak can also send the values to the serial monitor so it can read and write the data. Serial Monitor contains the value of the gas sensors from which ThingSpeak server takes.

3.11 SYSTEM DESIGN AND IMPLEMENTATION:

3.11.1 SYSTEM DESIGN:

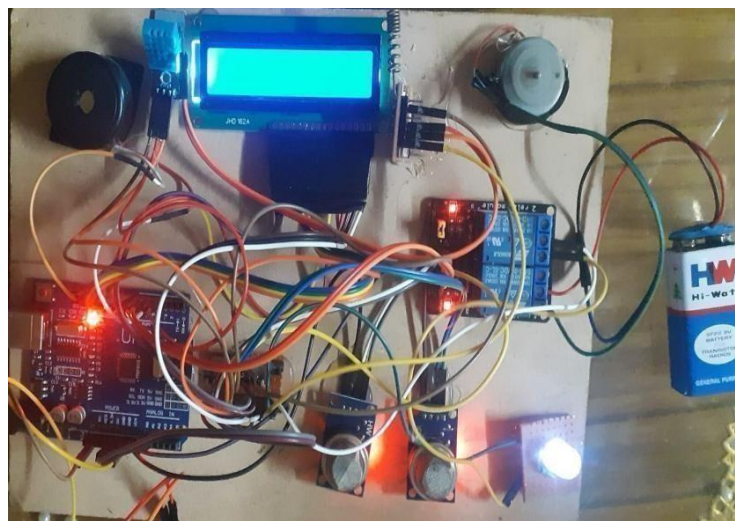
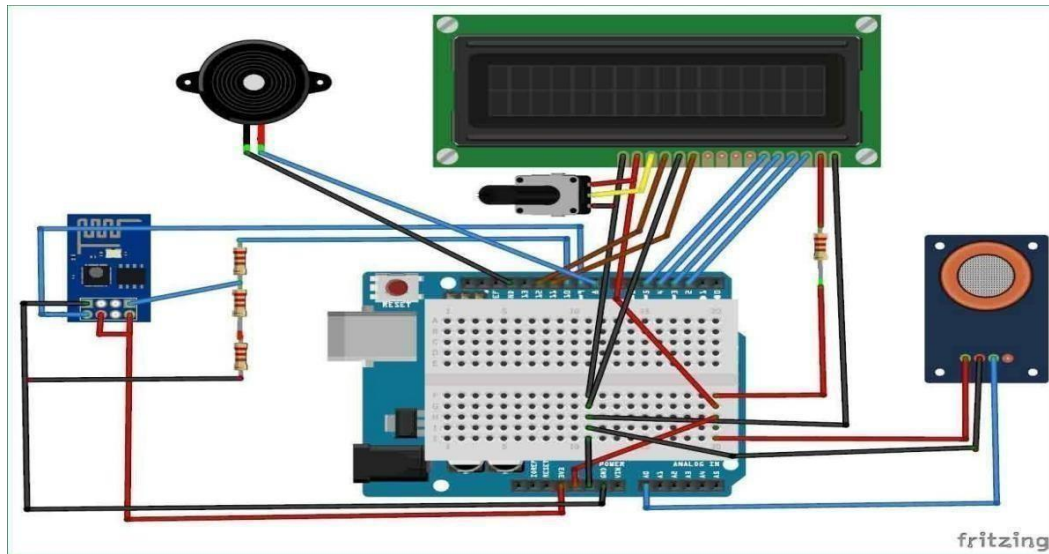
The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementations the candidate system. The design may be defined as the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient details to permit its physical realization. The designer's goal is how the output is to be produced and in what format samples of the output and input are also presented. Second input data and database files have to be designed to meet the requirements of the proposed output. The processing phases are handled through the program Construction and Testing.

Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation. The importance of software design can be stated in a single word "Quality". Design provides us with representations of software that can be assessed for quality. Design is the only way that we can accurately translate a customer's requirements into a finished software product or system without design we risk building an unstable system, that might fail if small changes are made or may be difficult to test, or on who's quality can't be tested. So it is an essential phase in the development of a software. We are implementing the proposed system using various hardware components like LM35 temperature sensor which operates from 4 to 30V and has a detection capability range of -55 centigrade to 150 degree centigrade, MQ2 smoke/gas sensor with a detection range of 300 to 10000 ppm (majorly used in domestic gas leakage alarms), MQ6 gas sensor which is mainly used to detect Methane, LCD display through which we would be able to see the pollutant levels locally.

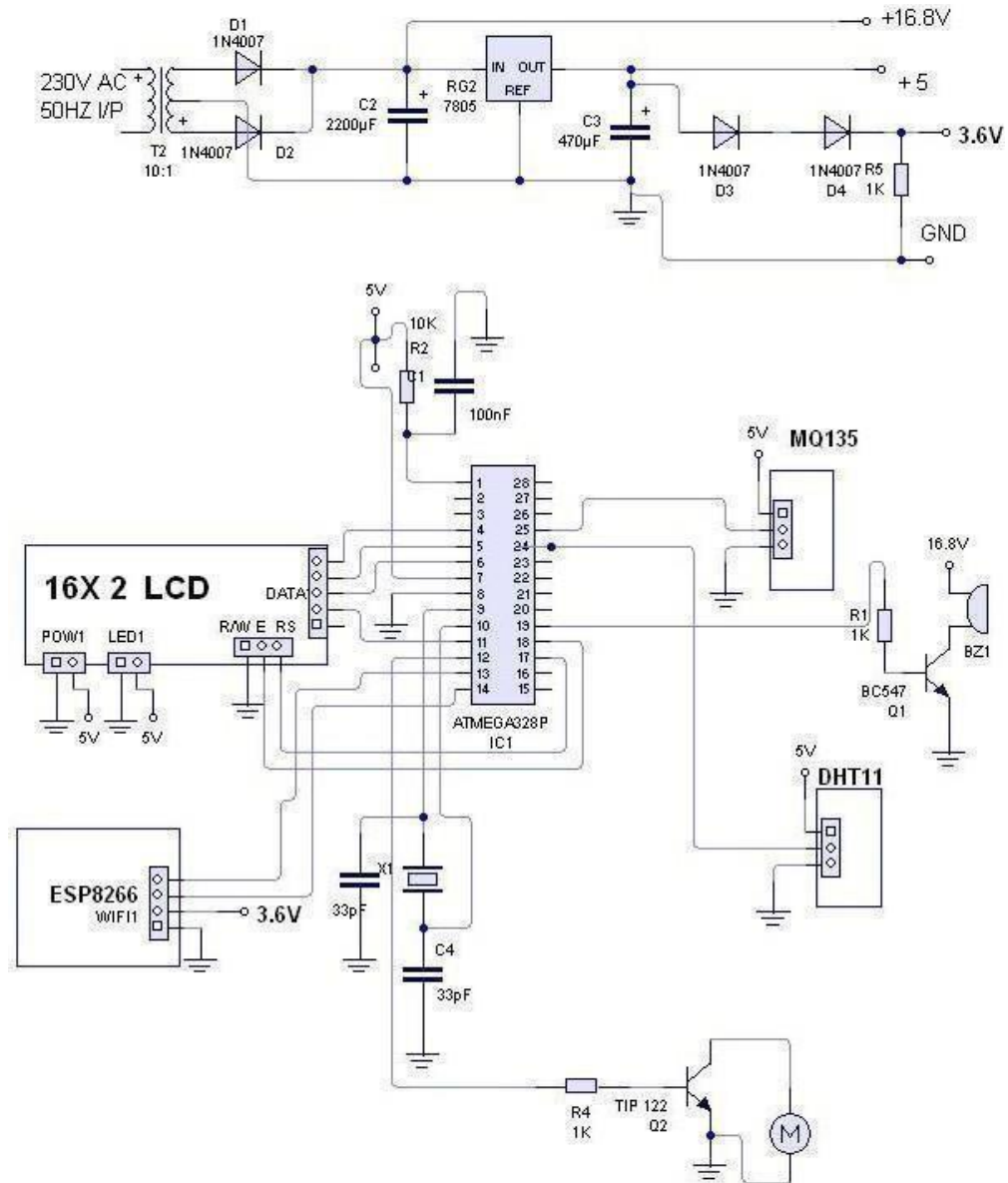
Arduino Uno is the microcontroller which interfaces with all of these sensors and collects the data and in turn displays it in the LCD and whenever the particular pollutant values are crossing the nominal values the buzzer is triggered, along with a led which triggers the motor which acts as a coolant. Node MCU esp8266

helps in sending the output data to a centralized server like thingspeak. Thingspeak is an open source website which helps in connecting with the internet enabled devices through wi-fi, it allows user to effectively monitor the data and visualize it collectively. It offers a great user interface which allows us to visualize the data graphically.

3.11.2 Circuit Diagram:



Detailed Diagram of Air quality Monitoring system:



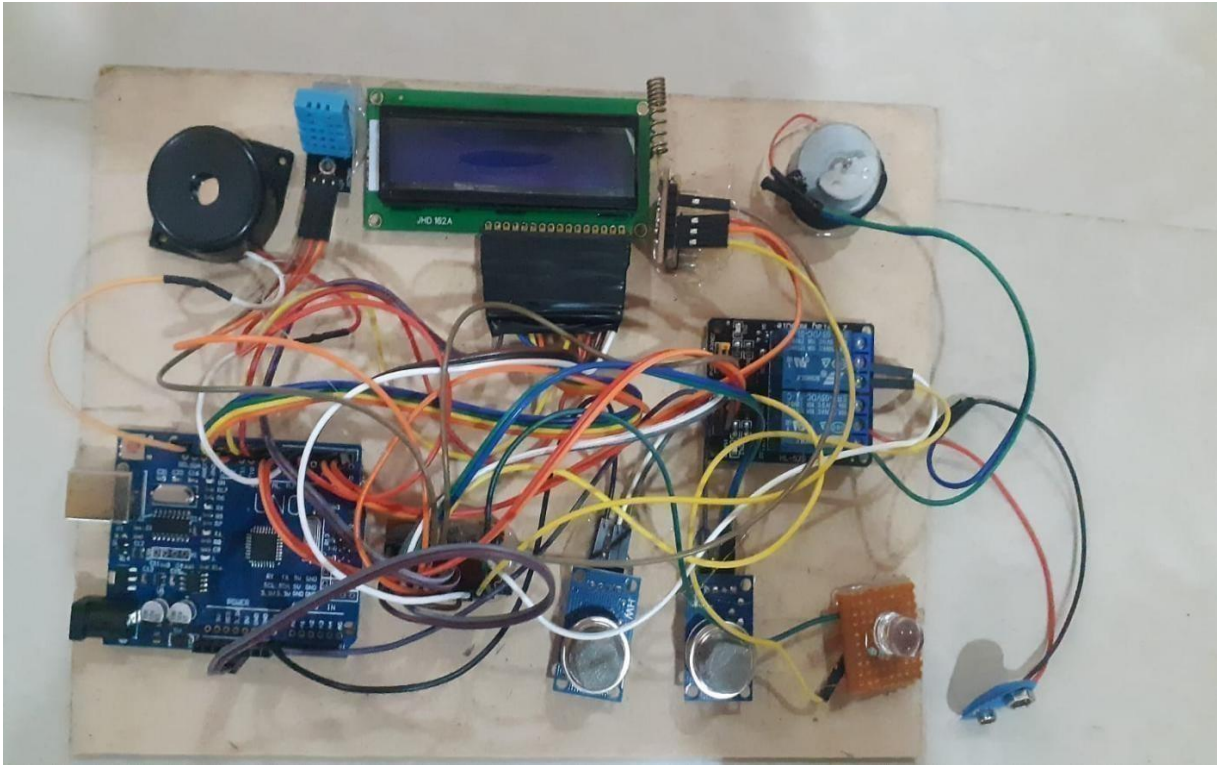
3.12 SYSTEM IMPLEMENTATION:

We start with connecting the ESP8266 with the Arduino. ESP8266 runs on 3.3V and if you will give it 5V from the Arduino then it won't work properly and it may get damaged. Connect the VCC and the CH_PD to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting three resistors in series like we did in the circuit. Connect the TX pin of the ESP8266 to the pin 10 of the Arduino and the RX pin of the esp8266 to the pin 9 of Arduino through the resistors.

ESP8266 Wi-Fi module gives your projects access to Wi-Fi or internet. It is a very cheap device and makes your projects very powerful. It can communicate with any microcontroller and it is the most leading device in the IOT platform. Learn more about here. Then we will connect the MQ135 sensor with the Arduino. Connect the VCC and the ground pin of the sensor to the 5V and ground of the Arduino and the Analog pin of sensor to the A0 of the Arduino. Connect a buzzer to the pin 8 of the Arduino which will start to beep when the condition becomes true. The MQ135 sensor can sense NH₃, NO_x, alcohol, Benzene, smoke, CO₂ and some other gases, so it is perfect gas sensor for our Air Quality Monitoring Project. When we will connect it to Arduino then it will sense the gases, and we will get the Pollution level in PPM (parts per million). MQ135 gas sensor gives the output in form of voltage levels and we need to convert it into PPM. So for converting the output in PPM, here we have used a library for MQ135 sensor, it is explained in detail in "Code Explanation" section below. Sensor was giving us value of 90 when there was no gas near it and the safe level of air quality is 350 PPM and it should not exceed 1000 PPM. When it exceeds the limit of 1000 PPM, then it starts cause Headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 2000 PPM then it can cause increased heart rate and many other diseases. When the value will be less than 1000 PPM, then the LCD and webpage will display "Fresh Air". Whenever the value will increase 1000 PPM, then the buzzer will start beeping and the LCD and webpage will display "Poor Air, Open Windows". If it will increase 2000 then the buzzer will keep beeping and the LCD.

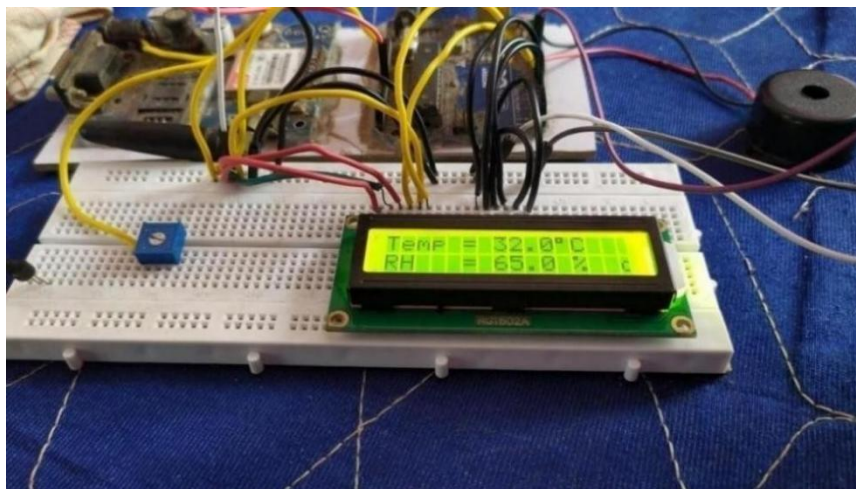
CHAPTER 4:RESULT

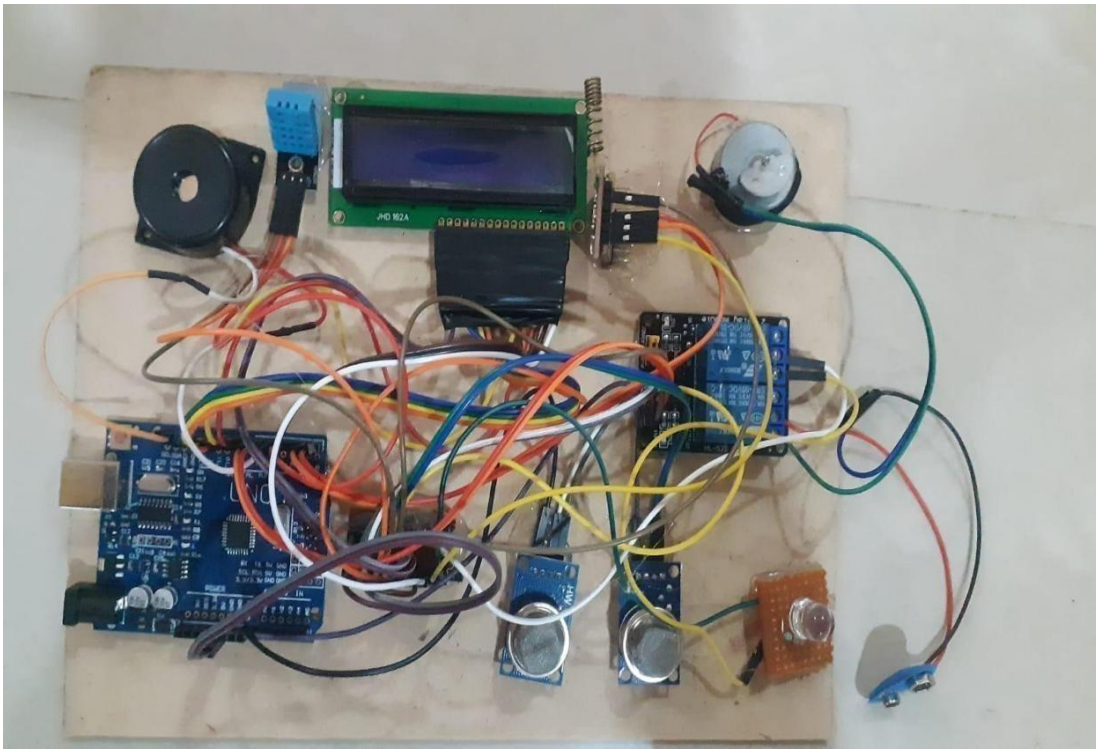
4.1 Hardware Kit:



4.1 Hardware kit of REAL TIME AIR QUALITY MONITORING SYSTEM

Display of Temp,RH



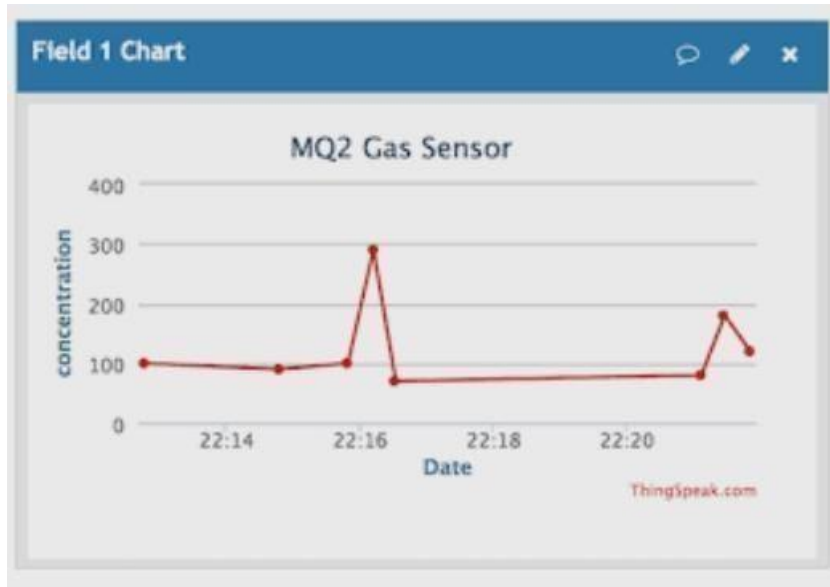


Circuit Diagram



Display of Smoke level

Results in ThingSpeak:



MQ-2 GAS SENSOR GRAPHICAL REPRESENTATION



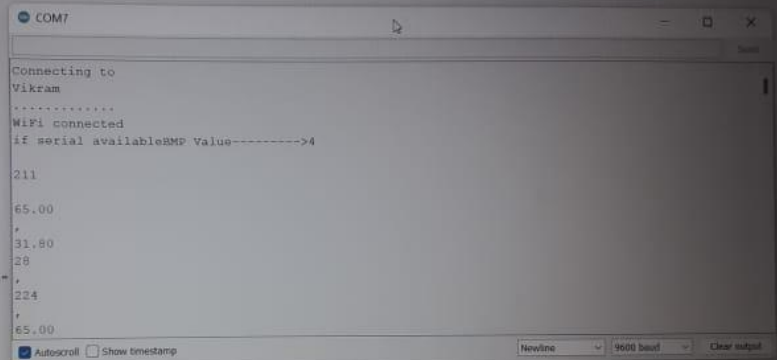
LM-35 GAS SENSOR GRAPHICAL REPRESENTATION



```
void loop()
{
  if (Serial.available() > 0)
  {
    Serial.print("if serial available");
    BMPValue = Serial.readStringUntil(',');
    Serial.read();
    TempValue = Serial.readStringUntil(',');
    Serial.read();
    TiltValue = Serial.readStringUntil('\0');
    Serial.print("BMP Value----->");
    Serial.print(BMPValue);
    Serial.print(TempValue);
    Serial.print(TiltValue);
  }

  if (client.connect(server, 80) // "184.106.153.149"
  {
    Serial.print("if client available");
    String postStr = apiKey;
    postStr += "field1=";
    postStr += BMPValue;
    postStr += "field2=";
    postStr += TempValue;
    postStr += "field3=";
    postStr += TiltValue;
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
  }
}
```



Serial Monitor Representation values of gas sensors

APPLICATIONS AND ADVANTAGES:

Applications:

Prevent smog dangers: This is one of the most important reasons why people use the air pollution control. ...

Protect their health: This is part of the reason why most people install the air pollution control. Most of these chemicals could damage the lungs.

Improve their indoor air quality: People use it to improve the air when they are indoors.

Advantages:

Air pollution control helps to protect the human health: To every human on earth, health is wealth; health is paramount for the running of our day to day activities, without which we would have to rely on others to live. Air Pollution Scrubber Suppliers come to help protect our health, which is very valuable.

Air pollution control helps prevent economic wastes: With air pollution control, the wastes accrued from dead crops and bad water will be limited or stopped. Some California wine producers complained about how their crops were polluted and customers complained that the wine had begun to taste like sulfur. With the presence of air pollution control, economic slowdowns like this will be prevented or at least managed to the barest minimum.

Increased worker productivity: No matter how strong the immune system is, there are times when it fails, especially when there is excess air pollution. As pollution is controlled, workers can now work for a longer period of time.

Helps improve indoor air quality: Air pollution control helps to secure the quality of the air inside your house.

Prevent smog dangers: This is one of the most important reasons why people use the air pollution control. Smog can be very hazardous, which is why air pollution control should be installed at an early stage to prevent smog.

Protect their health: This is part of the reason why most people install the air pollution control. Most of these chemicals could damage the lungs.

Improve their indoor air quality: People use it to improve the air when they are indoors.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

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

Our system mainly focussed on monitoring the harmful pollutants in air using wireless gas sensors and esp8266 wi-fi module which helps in monitoring the results. The proposed system is very much easy to implement. The concept is much new and we detect number of gases. The arduino and sensors are very less expensive so we can implement this system in high pollution areas. In future this type of systems have to be implemented because with help of this system we can actually detect and monitor the pollution of air and sound. In big cities this system is very much useful because implementation cost is very much less than the big systems. So we have to think about future and make world pollution free.

REFERENCES:

- G. Parmar, S. Lakhani, and M. Chattopadhyay, “An IoT based low cost air pollution monitoring system,” in 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), Bhopal, India, October 2017.
- K. Okokpujie, E. Noma-Osaghae, O. Modupe, S. John, and O. Oluwatosin, “A smart air pollution monitoring system,” International Journal of Civil Engineering and Technology, vol. 9, pp. 799–809, 2018.
- G. Rout, S. Karuturi, and T. N. Padmini, “Pollution monitoring system using IoT,” ARPN Journal of Engineering and Applied Sciences, vol. 13, pp. 2116–2123, 2018.
- B. C. Kavitha, D. Jose, and R. Vallikannu, “IoT based pollution monitoring system using raspberry–PI,” International Journal of Pure and Applied Mathematics, vol. 118, 2018.
- Asmi, A., Putra, J. C. P., & Rahman, I. B. A., “A Study of Indoor Air Quality of Public Toilet in University's Building”, in Humanities, Science and Engineering (CHUSER), IEEE Colloquium
- I-Haija, Q. A., Al-Qadeeb, H., & Alwaimi, A., “Case study: Monitoring of air quality in King Faisal University using a microcontroller and WSN”, Procedia Computer Science, volume

APPENDIX:

Code For Sensors:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);
#include <SoftwareSerial.h>
#include <DHT.h>
int DHTTYPE = DHT11;
int relay1 = 7;
int relay2 = 6;
int DHTPIN = 4;
int sensor1 = A0;
int sensor2 = A1;
int value1 = 0;
int value2 = 0;
DHT DHT(DHTPIN, DHTTYPE);
#define buz 5
void setup()
{
  Serial.begin(9600);
  lcd.begin(16,2);
  DHT.begin();
  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
  pinMode(sensor1,INPUT);
  pinMode(sensor2,INPUT);
  pinMode(buz,OUTPUT);
  lcd.print(" Air Monitoring");
  lcd.setCursor(0,1);
  lcd.print("  system");
  delay(1500);
  lcd.clear();
}
void loop()
```

```

{
float h = DHT.readHumidity();
float t = DHT.readTemperature();
value1 = analogRead(A0);
value2 = analogRead(A1);
Serial.println(value1);
Serial.println(",");
Serial.println(value2);
Serial.println(",");
Serial.println(h);
Serial.println(",");
Serial.println(t);
if(value1 < 200)
{
lcd.print("Gas PPM Level ");
lcd.setCursor(0,1);
lcd.print("High in Air");
digitalWrite(relay1,LOW);
digitalWrite(buz,HIGH);
delay(1500);
lcd.clear();
}
if(value2 > 300)
{
lcd.print("Smoke PPM Level");
lcd.setCursor(0,1);
lcd.print("High in Air");
digitalWrite(relay2,HIGH);
digitalWrite(buz,HIGH);
delay(1500);
lcd.clear();
}
else
{

```

```

digitalWrite(relay1,LOW);
digitalWrite(relay2,LOW);
digitalWrite(buz,LOW);
}
}

```

Code For Sending the Data To ThingSpeak:

```

#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
String apiKey = "M4M9XXZD229RQG2D";           // Enter your Write API key from ThingSpeak
const char* server = "api.thingspeak.com";
const char *ssid= "Vikram";                   // replace with your wifi ssid and wpa2 key
const char *pass = "8143805654";
#define net_led 14
String BMPValue;
String TempValue;
String TiltValue;
WiFiClient client;
void setup()
{
  lcd.begin(16, 2);
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
  delay(10);
  pinMode(net_led, OUTPUT);
  lcd.clear();
  lcd.print("IOT BASED");
  lcd.setCursor(0, 1);
  lcd.print("HEALTH MONITRING");
  delay(1000);
  lcd.clear();
}

```

```

lcd.print("USING NODEMCU");
lcd.setCursor(0, 1);
lcd.print("AND THINGSPEAK");
delay(1000);
Serial.println("Connecting to ");
Serial.println(ssid);
lcd.clear();
lcd.print("Connecting to ");
lcd.setCursor(0, 1);
lcd.print(ssid);
WiFi.begin(ssid, pass);

while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
lcd.clear();
lcd.print("WiFi connected");
digitalWrite(net_led, HIGH);
delay(500);
digitalWrite(net_led, LOW);
delay(500);
digitalWrite(net_led, HIGH);
delay(500);
digitalWrite(net_led, LOW);
delay(500);
digitalWrite(net_led, HIGH);
delay(500);
digitalWrite(net_led, LOW);
delay(500);
delay(1500);

```



```

}

void loop()
{
  if (Serial.available() > 0)
  {
    Serial.print("if serial available");
    BMPValue = Serial.readStringUntil(',');
    Serial.read();
    TempValue = Serial.readStringUntil(',');
    Serial.read();
    TiltValue = Serial.readStringUntil('\0');
    Serial.print("BMPValue----- >");
    Serial.print(BMPValue);
    Serial.print(TempValue);
    Serial.print(TiltValue);
  }

  if (client.connect(server, 80) // "184.106.153.149" or api.thingspeak.com
  {
    Serial.print("if client available");
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += BMPValue;
    postStr += "&field2=";
    postStr += TempValue;
    postStr += "&field3=";
    postStr += TiltValue;
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
  }
}

```

```
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\n\n");
client.print(postStr);
Serial.print("postStr====>");
Serial.print(postStr);
delay(500); lcd.clear();
lcd.print("Data sent to ");
lcd.setCursor(0, 1);
lcd.print(" Thingspeak");
delay(1000);
}
else
{
  Serial.print("else serial available");
  lcd.clear();
  lcd.print("Data not sent ");
  lcd.setCursor(0, 1);
  lcd.print(" to Thingspeak");
  delay(1000);
}
client.stop(); lcd.clear();
lcd.print("Reading...");
delay(2000);
}
```