INDEX

ABSTRACT

Liquefied Petroleum Gas (LPG) is a main source of fuel in urban areas. Most of the fire-break outs in houses and industries are due to leakage of gas. This cause damages to human life, environment, and equipment's and too many more things. Such accidents can be avoided using modern technology. In the present work, Node-MCU based gas leakage detection and alerting system is developed to monitor the concentration of LPG gas. System is built using Node-MCU ESP 32. MQ6 gas sensor is used for detection of LPG. Node-MCU takes the decision based on information provided by gas sensor. Firmware has been developed and deployed into Node MCU using arduino IDE. Presence of gas is displayed on LCD display module in terms of ppm. It also alerts the person by generating alarm. The designed system helps to achieve portability, light weight and cost-effective solution to commercially available system.

CHAPTER – 1

INTRODUCTION

Introduction to Project

By enabling common things to connect to other systems and devices and share data through sensors, software, and other technologies, the Internet of Things (IoT) has completely changed the way we live. IoT devices are widely employed in a variety of industries, including healthcare, agriculture, traffic monitoring, safety management, and environmental monitoring. These devices range from simple home goods to sophisticated industrial machinery. IoT, for instance, has the ability to automate farming procedures and tackle issues like an increasing population and unreliable human resources in agriculture. IoT may be quite useful in the chemical sector for maintaining safety procedures and keeping track of the environmental impacts of chemical manufacturing. IoT devices, however, have security issues, therefore developers must make security a major focus. [1]. Chemical catastrophes may have terrible effects, including the loss of life, destruction of property, and pollution of the environment.

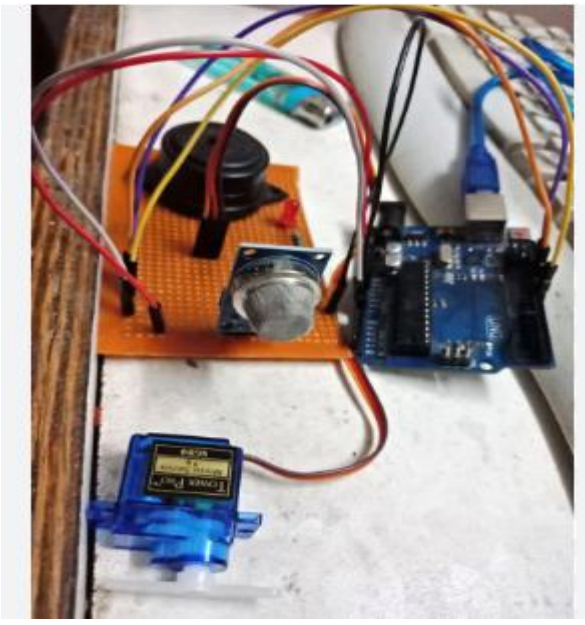


Fig:1.1: Gas leakage detection

The greatest chemical catastrophe in history happened in 1984 in Bhopal, India, where the leak of methyl isocyanine gas resulted in the deaths of over 2,500 people. India has nevertheless had chemical catastrophes that have caused deaths, injuries, and environmental harm even after this tragedy. Early gas leak identification and notification can be crucial in reducing mortality from such catastrophes [2]. In this project, we suggest a gas leak detection system that employs a gas sensor to find potentially harmful gases, setting off an alert and notifying the proper person through buzzer sound However, maintaining the security of IoT devices continues to be a major priority in dealing with such problems.

1.1 Introduction to Embedded System

An **embedded system** is a computer system that is designed to perform a specific task orset of tasks. It is a combination of computer hardware and software that is integrated into a larger system. Embedded systems are used in various applications such as home appliances, transportation, healthcare, business sector & offices, defence sector, aerospace, and agricultural sector. The three main components of an embedded system are hardware, software, and firmware. Hardware refers to the physical components of the system such as microprocessors or microcontrollers.

Software refers to the programs that run on the hardware. Firmware is a type of software that is embedded in the hardware and is responsible for controlling the system. An Embedded system is a special- purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Unlike a general-purpose computer, such as a personal computer, an Embedded System performs one or few predefined Tasks usually with very specific requirements. Since the system is dedicated to specified tasks, design engineers can optimize it, reducing the size and cost of the product. Embedded Systems are often mass- produced, benefiting from economies of scale.

Characteristics of Embedded System:

An embedded System is any computer system hidden inside a product other than a computer.

Throughput – Our system may need to handle a lot of data in short period of time.

Response – Our system may need to react to events quickly.

Test ability- Setting up equipment to test embedded software can be difficult.

Debug ability- Without a screen or a keyboard, finding out what the software is doing wrongis a troublesome problem.

Reliability – Embedded Systems must be able to handle any situation without human intervention.

Memory Space - Memory is limited on Embedded Systems, and you must make the softwareand the data fit into whatever memory exists.

Power Consumption – Portable systems must run on battery power, and the software in thesesystems must conserve power.

Processor hogs- Computing that requires large amounts of CPU time can complicate theresponse problem.

1.2 Introduction to IOT

INTERNET OF THINGS (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order tocommunicate Interaction amongst each other or with respect to the external environment. In the upcoming years, IoT-based technology will offer advanced levels of services and practically away people lead their daily lives. Advancements in medicine, power, gene therapy agriculture, smart cities, and smart homes are just a very few of the categorical example where IoT is strongly established.

IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data. With more than 7 billion

connected IOT devices today, experts are expecting this number to grow to 10 billion by 2020 and 22 billion by 2025. Oracle has a network of device partners.

The most important features of IoT on which it works are connectivity, integrating, active engagement, and many more. Connectivity refers to establish a proper connection between all the things of IoT platform it may be server or cloud. After connecting the IoT devices, it needs a highspeed messaging between the devices and cloud to enable reliable, secure and bi-directional communication. IoT makes things smart and enhances life through the use of data. For example, if we have a coffee machine whose beans have going to end, then the coffee machine it orders the coffee beans of your choice from the retailer.The most important features of IoT on which it works are connectivity,analysing, integrating, active engagement, and many more. Some of themare listed below:

Connectivity: Connectivity refers to establish a proper connection between all the things of IoT platform it may be server or cloud. After connecting the IoT devices, it needs a high speed messaging between the devices and cloud to enable reliable,secure and bi-directional communication.

Analysing: After connecting all the relevant things, it comes to real-time analysing the data collected and use them to build effective business intelligence. If we have a good insight into data gathered from all these things, then we call our system has a smart system.

Integrating: IoT integrating the various models to improve the user experience as well.

Artificial Intelligence: IoT makes things smart and enhances life through the use of data. For example, if we have a coffee machine whose beans have going to end, then the coffee machine itorders the coffee beans of your choice from the retailer.

Sensing: The sensor devices used in IoT technologies detect and measure any change in the environment and report on their status. IoT technology brings passive networks to

active networks. Without sensors,there could not hold an effective or true Iot environment.

Active Engagement: IoT makes the connected technology, product, or services to activeengagement between each other.

Endpoint Management: It is important to be the endpoint management of all the IoT system otherwise,it makes the complete failure of the system. For example, if a coffee machine itself order the coffee beans when it goes to end but what happens when it orders the beans from a retailer and we are not present at home for a few days, it leads to the failure of the IoT system.

1.3 Need of IoT

The Internet of Things (IoT) stands as a transformative force, reshaping our interactions with the world and revolutionizing diverse aspects of our daily lives. At its core, IoT thrives on connectivity, fostering seamless communication between devices and promoting interoperability.. Through automation, IoT enhances efficiency by enabling devices to operate autonomously based on predefined conditions or real-time data, reducing the need for constant human intervention. In the realm of smart cities, IoT contributes to urban development by introducing intelligent transportation systems, energy management, and sustainable practices, thereby enhancing overall quality of life.

Health care benefits from IoT through wearables and remote monitoring tools, offeringpersonalized insights and timely interventions. Industries leverage Industrial IoT (IIoT) to optimize manufacturing processes, monitor equipment health, and implement predictive maintenance strategies, leading to increased productivity and cost savings. From smart homes with connected appliances to environmental monitoring and supply

chain optimization, IoT's impact is far-reaching, creating a more connected, efficient, and intelligent world across variousdomain

CHAPTER -2

LITERATURE SURVEY

2.1 Introduction

Gas leakage detection systems are crucial for ensuring safety in residential, commercial, and industrial environments. These systems employ different technologies and configurations to detect the presence of hazardous gases promptly. Here's an introduction to the various ways or systems through which gas leakage detection can be implemented:

2.2 Standalone Systems:

- **Basic Alert Systems:**
  - Utilize simple gas sensors (e.g., MQ series) connected to a microcontroller (e.g., Arduino).
  - Emit audible alerts (e.g., buzzer) when gas levels exceed predefined thresholds.
  - Suitable for residential use where basic detection and alerting are sufficient.

2.3 Wired Systems:

- **Centralized Monitoring:**
  - Deploy gas sensors connected via wired networks to a central control panel.
  - Provide real-time monitoring and localized alerts within a building or facility.
  - Commonly used in industrial settings for comprehensive gas detection coverage.

2.4 Wireless Systems:

- **WiFi-enabled Systems:**
  - Connect gas sensors wirelessly to a local WiFi network.

- Transmit data to a central monitoring station or cloud platform for remote monitoring and alerts.
- Ideal for applications requiring flexibility and scalability across large areas.

➢ **Bluetooth-enabled Systems:**

- Utilize Bluetooth connectivity between gas sensors and a central monitoring unit or mobile device.
- Suitable for smaller-scale deployments or portable gas detection needs.

2.5 IoT-enabled Systems:

➢ **Cloud-connected Systems:**

- Integrate gas sensors with IoT platforms (e.g., AWS IoT, Google Cloud IoT).
- Enable data storage, analysis, and remote monitoring.
- Send notifications or alerts via email, SMS, or push notifications based on detected gas levels.

2.6 Portable Systems:

➢ **Battery-powered Systems:**

- Include gas sensors powered by batteries for portable or temporary deployment.
- Provide flexibility in monitoring gas levels in various locations without relying on mains power.

2.7 Smart Home Integration:

➢ **Integration with Home Automation:**

- Incorporate gas sensors into existing smart home ecosystems (e.g., SmartThings, Home Assistant).

- Trigger automated responses such as HVAC shutdown or alert notifications to homeowners.

2.8 Industrial Systems:

- **PLC-based Systems:**

  - Employ Programmable Logic Controllers (PLCs) for automated gas detection and control in industrial environments.
  - Integrate with industrial control systems for enhanced safety protocols and operational efficiency.

2.9 Advanced Detection Systems:

- **Multi-gas Detection Systems:**

  - Utilize advanced sensors capable of detecting multiple types of gases simultaneously (e.g., methane, carbon monoxide).
  - Provide comprehensive monitoring and alerting capabilities in complex industrial or hazardous environments.

Literature review

A literature review on various gas leakage detection systems provides insights into the technologies, applications, and advancements in each category. Here's a summary based on the systems you've outlined:

1. Wired Systems: Centralized Monitoring Systems

Wired systems traditionally use a centralized control panel connected via wires to gas sensors strategically placed throughout a building or facility. These systems are reliable and typically employed in industrial settings where continuous monitoring and immediate local response are critical. The control panel displays real-time gas levels and triggers alarms onsite, ensuring rapid intervention to mitigate risks.

2. Wireless Systems:

WiFi-enabled Systems

WiFi-enabled gas detection systems utilize microcontrollers such as ESP8266 or ESP32 to wirelessly connect gas sensors to a local network. Data from sensors can be monitored remotely through a web interface or mobile app, enabling facility managers or homeowners to access real-time gas concentration data from anywhere. Alerts can be sent via notifications, enhancing responsiveness to potential gas leaks and enabling timely actions to ensure safety.

Bluetooth-enabled Systems

Bluetooth-enabled gas detection systems are suitable for smaller-scale applications or areas where WiFi connectivity may not be feasible or necessary. Sensors communicate wirelessly with a central monitoring unit or mobile device equipped with Bluetooth capability. These systems offer flexibility and ease of deployment in residential or commercial settings, providing localized monitoring and alerting capabilities.

3. IoT-enabled Systems:

IoT-enabled gas detection systems leverage cloud-connected platforms such as AWS IoT or Google Cloud IoT for data storage, analysis, and remote monitoring. Gas sensors transmit real-time data to the cloud, where it can be accessed and analyzed by stakeholders. Alerts are generated based on predefined thresholds and sent via email, SMS, or push notifications, enabling proactive measures to be taken in response to detected gas leaks. These systems are scalable and offer enhanced integration with other IoT devices and systems, making them ideal for smart buildings and industrial IoT applications.

4. Portable Systems:

Battery-powered gas detection systems are designed for temporary monitoring or environments without access to mains power. These portable units are equipped with gas sensors and alert mechanisms such as buzzers or LEDs, ensuring mobility and flexibility in deployment. Portable systems are commonly used in construction sites, temporary work areas, or during maintenance activities where continuous monitoring of gas levels is necessary to ensure safety.

5. Smart Home Integration:

Gas sensors integrated into home automation systems like SmartThings or Home Assistant enable seamless integration with existing smart home ecosystems. Integration allows gas detection events to trigger automation routines, such as HVAC shutdowns or notifications to homeowners via mobile apps. These systems enhance residential safety and provide homeowners with peace of mind by enabling proactive management of gas leak risks.

6. Industrial Systems:

PLC-based gas detection systems utilize Programmable Logic Controllers (PLCs) for automated and robust gas detection in industrial environments. PLCs integrate with existing control systems to provide real-time response and safety

protocols in industrial settings where gas leaks can pose significant risks to personnel and operations. These systems ensure continuous monitoring and rapid intervention, minimizing downtime and enhancing workplace safety.

7. Advanced Detection Systems:

Multi-gas detection systems utilize advanced sensors capable of detecting multiple types of gases simultaneously, such as methane, propane, and carbon monoxide. These systems provide comprehensive monitoring and alerting capabilities in diverse industrial or commercial environments where multiple gas types may be present. Advanced detection systems are essential for ensuring comprehensive safety measures and regulatory compliance in complex industrial settings.

CHAPTER-3

DESIGNED SYSTEM

3.1 Introduction

This system provides a robust solution for monitoring and detecting gas leaks in residential and commercial environments. By leveraging the capabilities of the ESP8266 microcontroller and various gas sensors, this system ensures timely detection of potentially hazardous situations, enabling prompt intervention and ensuring safety.

System Overview

The Gas Leakage Detection System comprises:

- **ESP8266 Microcontroller**: Acts as the central processing unit, responsible for data acquisition from sensors and communication with the monitoring interface.
- **Gas Sensors:** Utilizes MQ series gas sensors (e.g., MQ-2, MQ-5) for detecting specific types of gases
  such as LPG, propane, methane, and carbon monoxide.
- **Alarm Module:** Triggers audible and visual alerts in the presence of gas leaks, alerting occupants to evacuate and take necessary precautions.
- **Data Communication:** Utilizes Wi-Fi connectivity provided by ESP8266 to transmit real-time gas concentration data to a monitoring dashboard.

Key Features

- **Real-time Monitoring:** Continuously monitors gas levels in the environment, providing real-time updates to ensure immediate action can be taken upon detection of a gas leak.
- **Alert Mechanisms:** Issues audible alarms and visual indicators upon detecting gas leaks, ensuring prompt evacuation and response.

Documentation Structure

This documentation is structured to provide comprehensive guidance on:

- **System Installation:** Steps to install and set up the Gas Leakage Detection System.
- **Configuration:** Guidance on configuring the ESP8266 and integrating various gas sensors.
- **Usage:** Instructions on using the monitoring interface and interpreting gas concentration data.
- **Maintenance:** Tips for maintaining the system to ensure optimal performance and reliability.

3.2 Objectives

Developing a Functional Prototype: Create a working model that demonstrates real-time detection of gas leaks using an ESP8266 microcontroller and a gas sensor module.

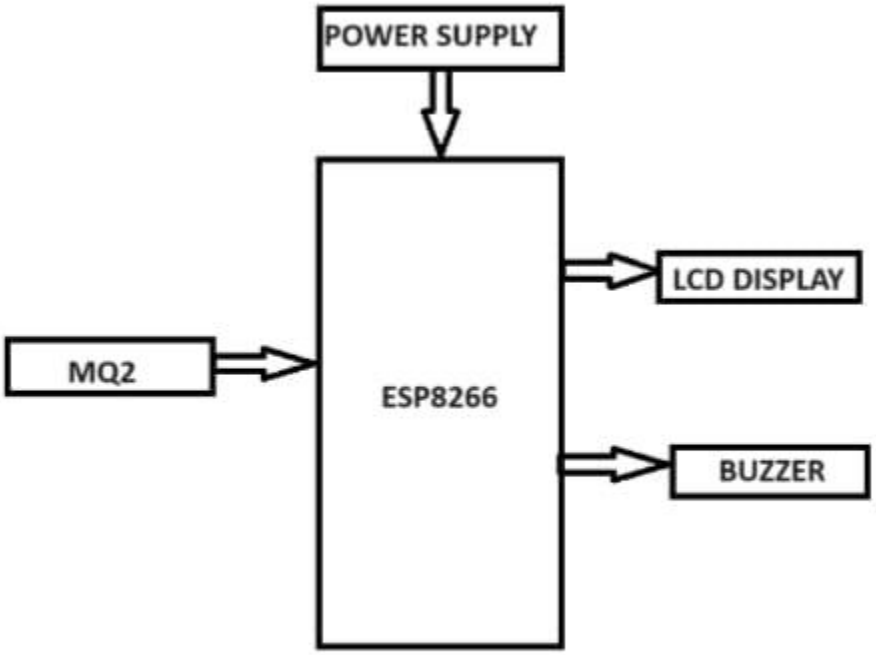
Promoting Safety: Enhance safety measures in residential or industrial environments by detecting and alerting about potential gas leaks promptly.

Learning Experience: Serve as an educational resource for learning about IoT hardware integration, sensor interfacing, and network communication.Integration with IoT

Ecosystem: Enable integration with IoT platforms or local networks for remote monitoring and alert notifications

.Scalability and Adaptability: Design the system to be scalable for different environments and adaptable to various gas sensors and communication protocols.

3.3 Block Diagram



3.4 Tools Required

3.4.1 Hardware Components

- ✓ Power Supply
- ✓ Node MCU
- ✓ Jumper wires
- ✓ LCD

3.4.2 Soft Ware Requirements

- ✓ Arduino IDE
- ✓ Proetus
- ✓ Code develops through Embedded C

CHAPTER-4

HARDWARE IMPLEMENTATION

4.1 Node MCU ESP8266

4.1.1 Description

Node MCU ESP8266 Description Node MCU is an open-source firmware for which open- source prototyping board designs are available. The name “Node MCU” combines “node” and “MCU” (micro-controller unit). The term “Node MCU” strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs areopen source. Node MCU ESP8266 and Node MCU ESP32

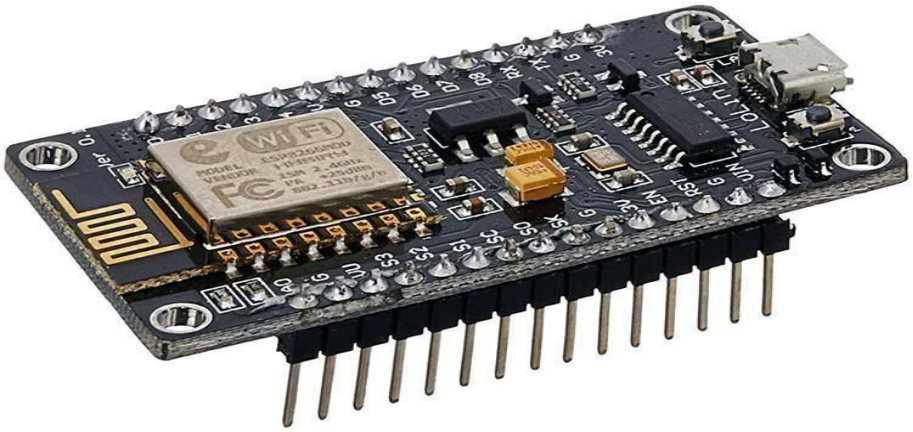


Fig 4.1: Node MCU

are becoming very popular and are almost used in more than 50% IoT based projects today.

The firmware uses the Lua scripting language. The firmware is based on the eLua project and built on the Espressif Non-OS SDK for ESP8266. It uses many open-source projects, such as luacjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented. The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller

surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototypingon breadboards.

The design was initially was based on the ESP-12 module of the ESP8266, which is a Wi- Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

About the Node MCU ESP8266 Pinout:

Node MCU ESP8266 Wi-Fi Module is an open-source Lua based firmware and developmentboard specially targeted for IoT based applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

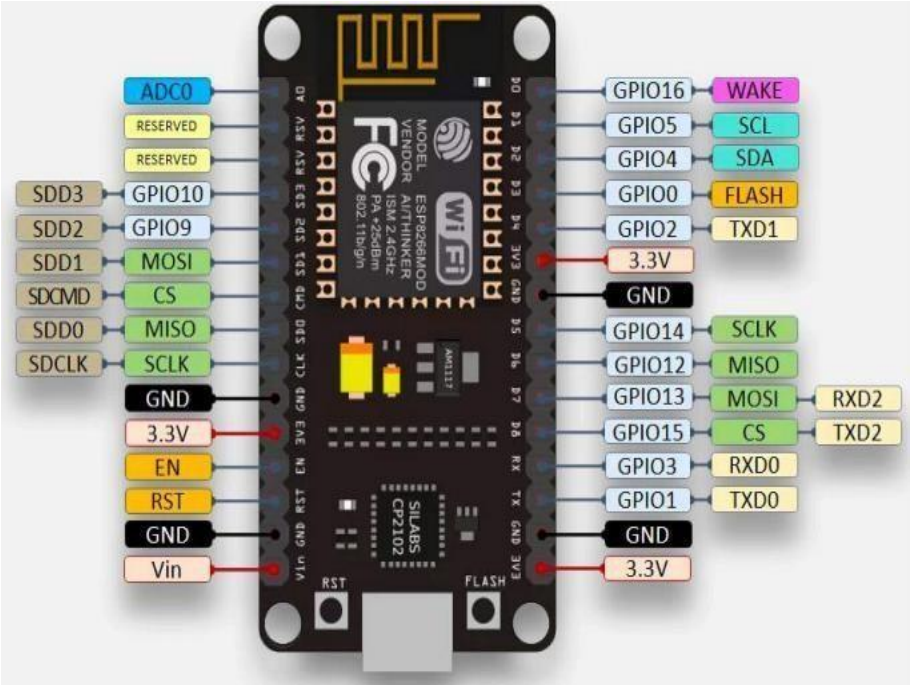


Fig 4.2 Pin Diagram of Node MCU

4.1.2 Node MCU ESP8266 Features:

Microcontroller: Tensilica 32-bit RISC CPU Xtensa

LX106Operating Voltage: 3.3V

Input Voltage: 7-12V

Digital I/O Pins (DIO):

16 Analog Input Pins

(ADC): 1UARTs: 1

SPIs: 1

I2Cs: 1

Flash Memory: 4

MBSRAM: 64 KB

Clock Speed: 80 MHz

USB-TTL based on CP2102 is included onboard, Enabling Plug n

Play PCBAntenna Small Sized module to fit smartly inside your IoT

projects

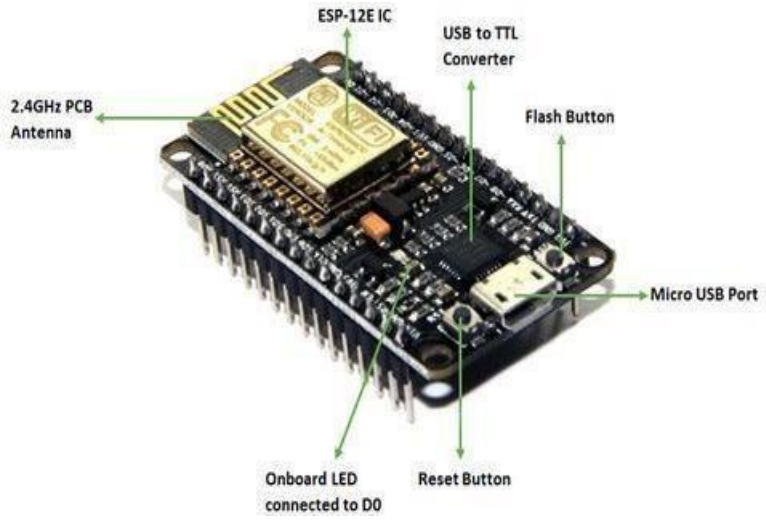


Fig 4.3: Layout of the Node MCU

4.1.3 Node MCU ESP8266 Pinout:

For practical purposes ESP8266 Node MCU V2 and V3 boards present identical pinouts. Whileworking on the Node MCU based projects we are interested in the following pins.

Power pins (3.3 V).

Ground pins

(GND).Analog

pins (A0).

Digital pins (D0 – D8, SD2, SD3, RX, and TX – GPIO XX)

Most ESP8266 Node MCU boards have one input voltage pin (Vin), three power pins (3.3v), fourground pins (GND), one analog pin (A0), and several digital pins (GPIO XX).

Pin Code Arduino alias

A0 A0 A0

D0 GPIO 16 16

D1 GPIO 5 5

D2 GPIO 4 4

D3 GPIO 0 0

D4 GPIO 2 2

D5 GPIO 14 14

D6 GPIO 12 12

D7 GPIO 13 13

D8 GPIO 15 15SD2 GPIO 9 9

SD3 GPIO 10 10

RX GPIO 3 3

TX GPIO 1 1

4.2 Gas Sensor

As for the main sensor for detecting gas, we use the MQ-2 gas sensor as the main detector. MQ-2 is a gas sensor that has a high sensitivity to types of flammable gases such as LPG, Propane, and Hydrogen. Besides having high sensitivity, MQ-2 was chosen because of its low cost and suitable for different applications. The gas sensor will be configured via OpenCR, which is like Arduino. In OpenCR, the gas sensor is set according to the lower limit and the upper limit of the gas to be detected, thus avoiding reading the misleading data. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. By placing it in four different directions, it is expected that the value obtained is more accurate and precise.

4.2 MQ-2 sensor

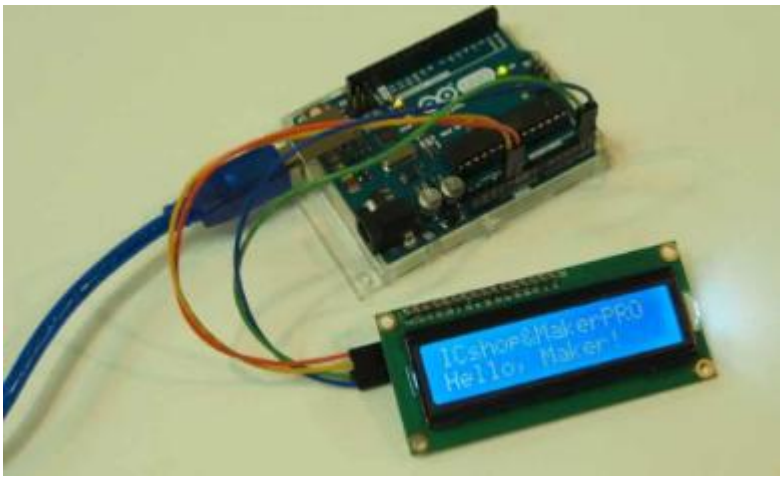
4.3 Liquid Crystal Display

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly but instead use a backlight or reflector to produce imagesin colour or monochrome.

4.3.1 Description

A **liquid-crystal display** (**LCD**) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with

polarizers. Liquid crystals do not emit light directly but instead use a backlight or reflector to produce images in colour or monochrome. LCDs are available to display arbitrary images (as in a general- purpose computer display) or fixed images with low information content, which can be displayed or hidden: preset words, digits, and seven-segment displays (as in a digital clock) are all examples of devices with these displays. They use the same basic technology, except that arbitrry imagesare made from a matrix of small pixels, while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement.



4.3 Liquid Crystal Display

4.3.2Working

An LCD panel is made of many layers. These consist of a polariser, polarised glass, LCD fluid, conductive connections etc. Polarisation is a process in which the vibration of light waves is restricted to a single plane, resulting in the formation of light waves known as polarised light. Since liquid crystals do not produce light of their own, they need an external light source to work. An LCD panel has sets of polarised glass consisting of liquid crystal materials in between them. When the external light passes through one of the polarised glasses and electric current is applied on the liquid crystalmolecules, they align themselves in such a way that polarised light travels from the first layer to the second polarised glass, causing an image to appear on the screen.

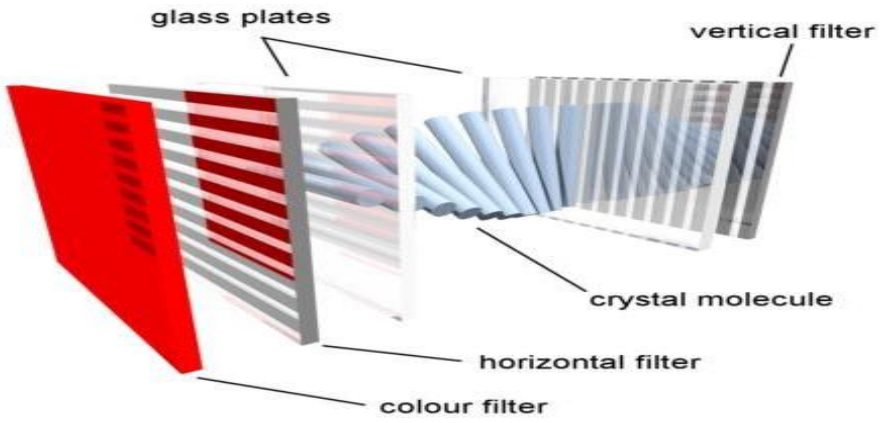


Fig 4.10: Internal Blocks of LCD

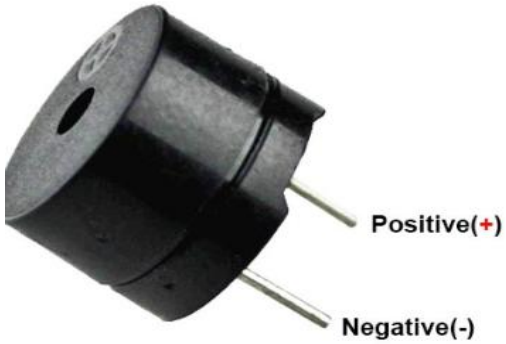
4.1.4 Types of LCD

Reflective: This type of LCD has a mirror layer. When a light ray within an LCD is reflected by the mirror layer, then visible patterns are produced on the LCD.

Transmissive: Here the LCD has a backlight, which passes through the LCD polarised glass to produce visible pattern. But because it uses backlight for working, the images displayed in such LCD types appear very dim when used under bright sunligh

4.4 Buzzer

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.



4.4 Buzzer

4.4.1 Description

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

4.4.2 Working

The working principle of a buzzer depends on the theory that, once the voltage is given across a piezoelectric material, then a pressure difference is produced. A piezo type includes piezo crystals among two conductors.

Once a potential disparity is given across these crystals, then they thrust one conductor & drag the additional conductor through their internal property. So this continuous action will produce a sharp sound signal.

CHAPTER-5

SOFTWARE IMPLEMENTATION

5.1 Arduino IDE

5.1.1 Introduction to Arduino IDE

IDE stands for Integrated Development Environment - An official software introduced by Arduino.cc that is mainly used for writing, compiling and uploading the code in almost all Arduino modules/boards. Arduino IDE is open-source software and is easily available to download & installfrom Arduino Official Site.

In this post, I'll take you through the brief Introduction of the Software, how you can install it, and make it ready for your required Arduino module.
Let's dive in and get down to the nitty-gritty of this Software.

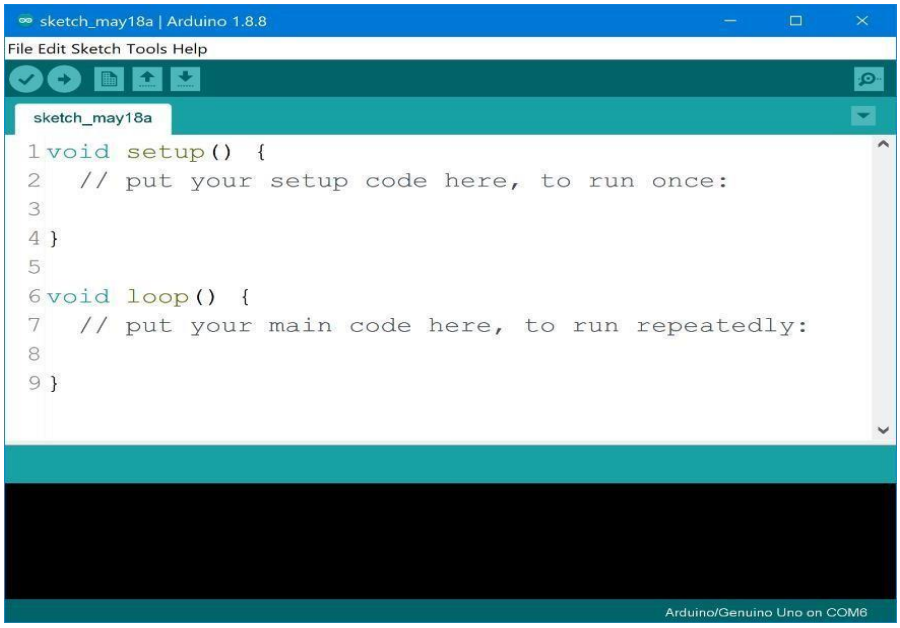


Fig 5.1: Arduino IDE Editor page

Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.

It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.It is available for all operating systems i.e., MAC, Windows, Linux and runs on the Java Platform that comes with inbuiltfunctions and commands that play a vital role in debugging, editing and compiling the code.A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.The IDE environment mainly contains two basic parts: Editorand Compiler where former is used for writing the required code and later is used for compiling anduploading the code into the given Arduino Module.

This environment supports both C and C++ languages.

5.1.2 How to Download Arduino IDE

You can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

8.1 or Windows 10, as the app version is not compatible with Windows 7 or older version of thisoperating system.

You can download the latest version of Arduino IDE for Windows (Non admin standalone version),by clicking below button:

Arduino IDE Download

The IDE environment is mainly distributed into three sections

1. Menu Bar
2. Text Editor

3. Output Pane

As you download and open the IDE software, it will appear like an image below:

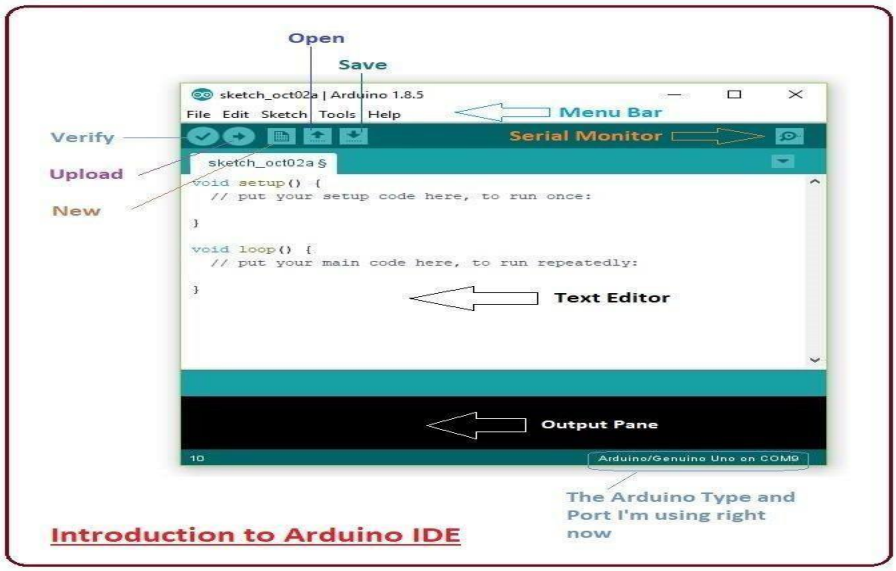


Fig 5.2: Introduction to Arduino IDE

The bar appearing on top is called Menu Bar that comes with five different options as

- File You can open a new window for writing the code or open an existing one. The followingtable shows number of further subdivisions the file option is categorized into:

File	
New	This is used to open new text editor window to write your code
Open	Used for opening the existing written code
Open Recent	The option reserved for opening recently closed program
Sketchbook	It stores the list of codes you have written for your project
Examples	Default examples already stored in the IDE software
Close	Used for closing the main screen window of recent tab. If two tabs are open, it will ask you again as you aim to close the second tab
Save	It is used for saving the recent program
Save as	It will allow you to save the recent program in your desired folder
Page setup	Page setup is used for modifying the page with portrait and landscape options. Some default page options are already given from which you can select the page you intend to work on
Print	It is used for printing purpose and will send the command to the printer
Preferences	It is page with number of preferences you aim to setup for your text editor page
Quit	It will quit the whole software all at once

Fig 5.3: File subdivisions in Arduino IDE

-

As you go to the preference section and check the compilation section, the Output pane will showthe code compilation as you click the upload button.

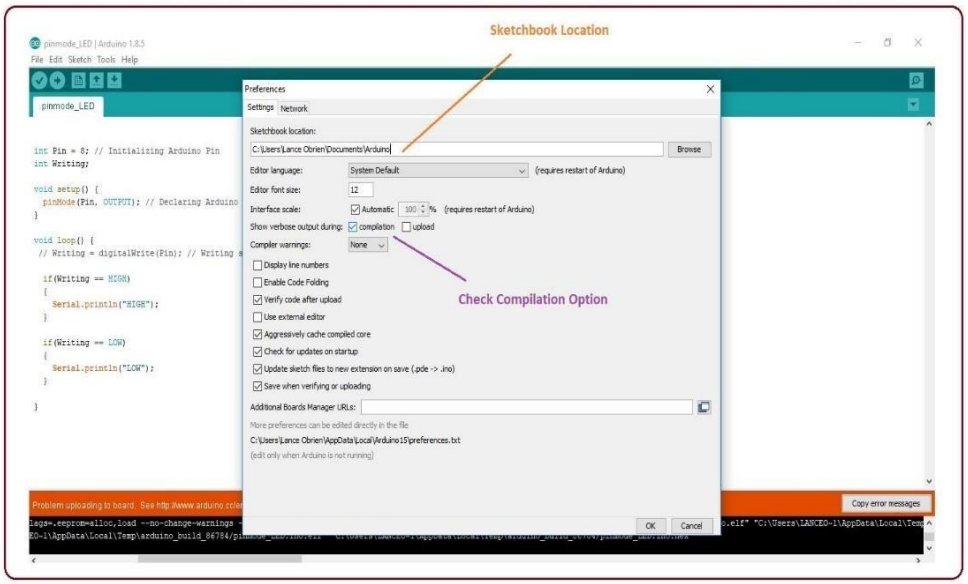


Fig 5.4: Selection of compilation

And at the end of the compilation, it will show you the hex file it has generated for the recentsketch that will send to the Arudino Board for the specific task you aim to achieve.

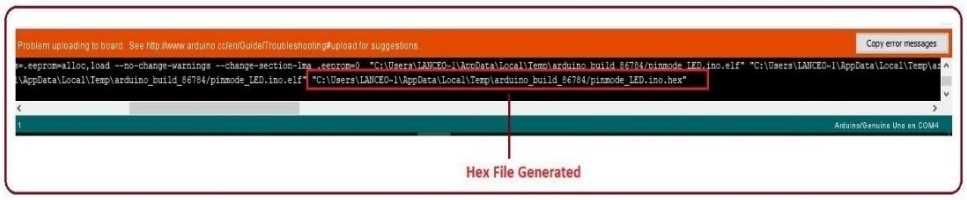


Fig 5.5: Hex file generation

- Sketch - For compiling and programming
- Tools - Mainly used for testing projects. The Programmer section in this panel is used forburning a boot loader to the new microcontroller.
- Help - In case you are feeling Edit - Used for copying and pasting the code with furthermodification for font
- sceptical about software, complete help is available from getting started to troubleshooting.
- The Six Buttons appearing under the Menu tab are connected with the running program asfollows.

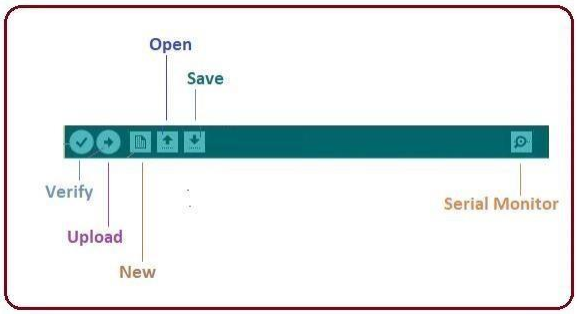


Fig 5.6: Serial monitor

- The check mark appearing in the circular button is used to verify the code. Click this once youhave written your code.
- The arrow key will upload and transfer the required code to the Arduino board.
- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.
- The downward arrow is used to save the current running code.
- The button appearing on the top right corner is a Serial Monitor - A separate pop-up window that acts as an independent terminal and plays a vital role in sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at oncewill open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connectedto your computer by USB cable in order to activate the Serial Monitor.
- You need to select the baud rate of the Arduino Board you are using right now. For my ArduinoUno Baud Rate is 9600, Monitor, the output will show as the image below.

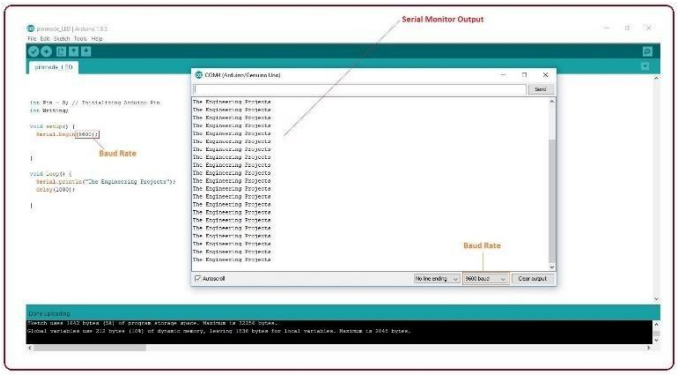


Fig 5.7: output of the serial monitor

 The main screen below the Menu bard is known as a simple text editor used for writing therequired code.

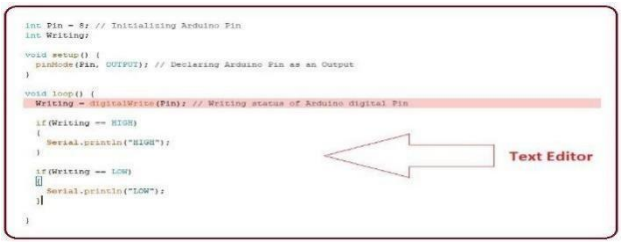


Fig 5.8: Text editor

 Output Pane that mainly highlights the compilation status of the running code: the memory usedby the code, and errors that occurred in the program. You need to fix the bottom of the main screen is described as those errors before you intend to upload the hex file into your Arduino Module.

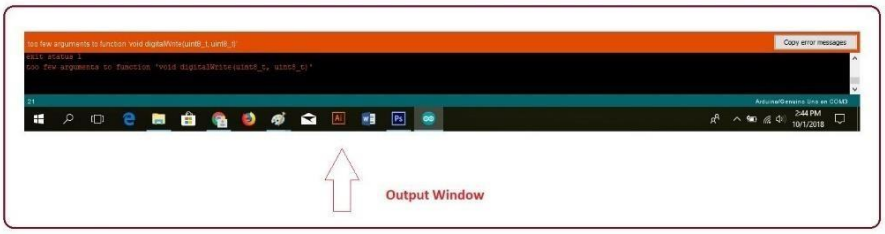


Fig 5.9: output window

 More or less, Arduino C language works similar to the regular C language used for any embeddedsystem microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

5.1.3 Libraries

- Libraries are very useful for adding extra functionality into the Arduino Module.
- There is a list of libraries you can check by clicking the Sketch button in the menu bar and goingto Include Library.
- As you click the Include Library and Add the respective library it will be on the top of the sketchwith a #include sign. Suppose, I Include the Liquid Crystal library, it will appear on the text editor as

#include <Liquid Crystal. h>

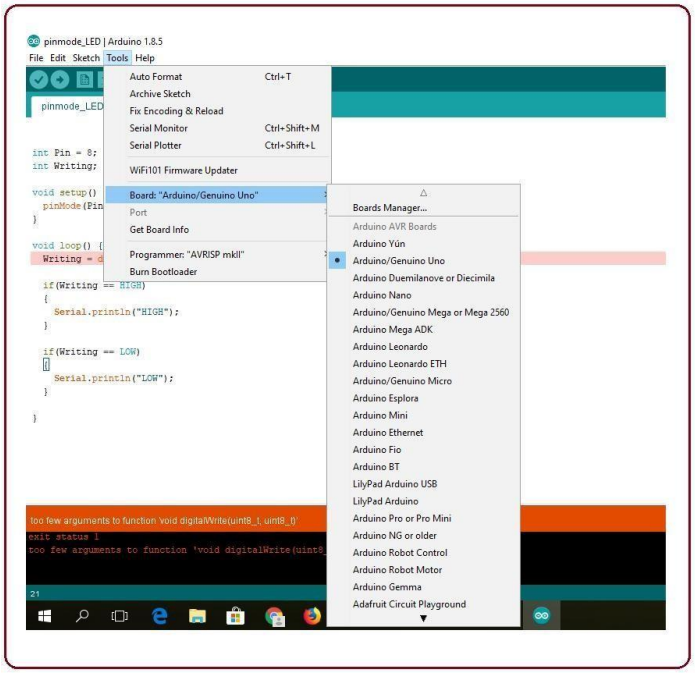


Fig 5.10: Selection of tools

- As you click the Include Library and Add the respective library it will be on the top of the sketchwith a #include sign. Suppose, I Include the LiquidCrystal library, it will appear on the text editor as#include <Liquid Crystal.h>
- Most of the libraries are preinstalled and come with the Arduino software.However, you can also download them from external sources.

5.1.4 Making Pins Input or Output.

The digitalRead and digitalWrite commands are used for addressing and making the Arduino pins as an input and output respectively. These commands are text sensitive i.e., you need to write them down the exact way they are given like digitalWrite starting with small "d" and write with capital "W". Writing it down with DigitalWrite or digitalWrite won't be calling or addressingany function.

5.1.5 How to Select the Board

- In order to upload the sketch, you need to select the relevant board you are using and the portsfor that operating system.
- As you click the Tools on the menu, it will open like the figure below:

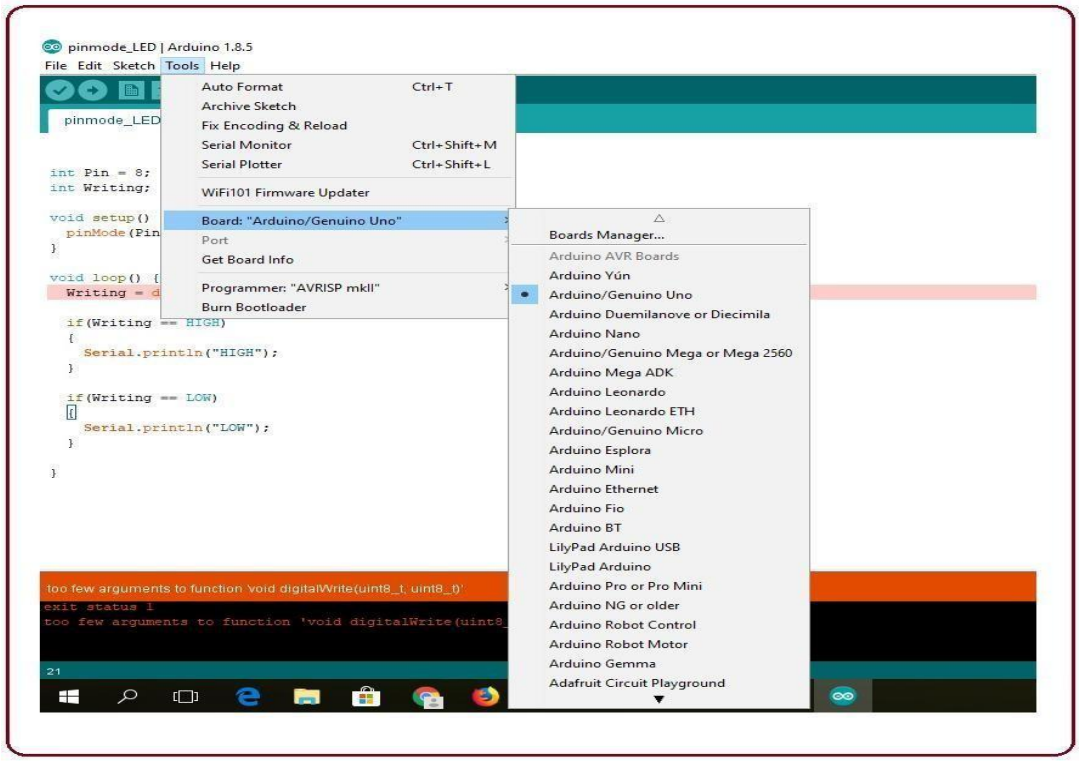


Fig 5.11: Selection of board manager

- Just go to the "Board" section and select the board you aim to work on. Similarly, COM1, COM2,COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for theUSB serial device in the port section of the Windows Device Manager.
- The following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with the COM4 port at the right bottom corner of the screen.
- After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six-button section or you can go to the Sketch section andpress verify/compile and then upload.

  The sketch is written in the text editor and is then saved with the file extension into. It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, the older versions may require the physical reset on theboard.

- Once you upload the code, TX and RX LEDs will blink on the board, indicating the desiredprogram is running successfully.

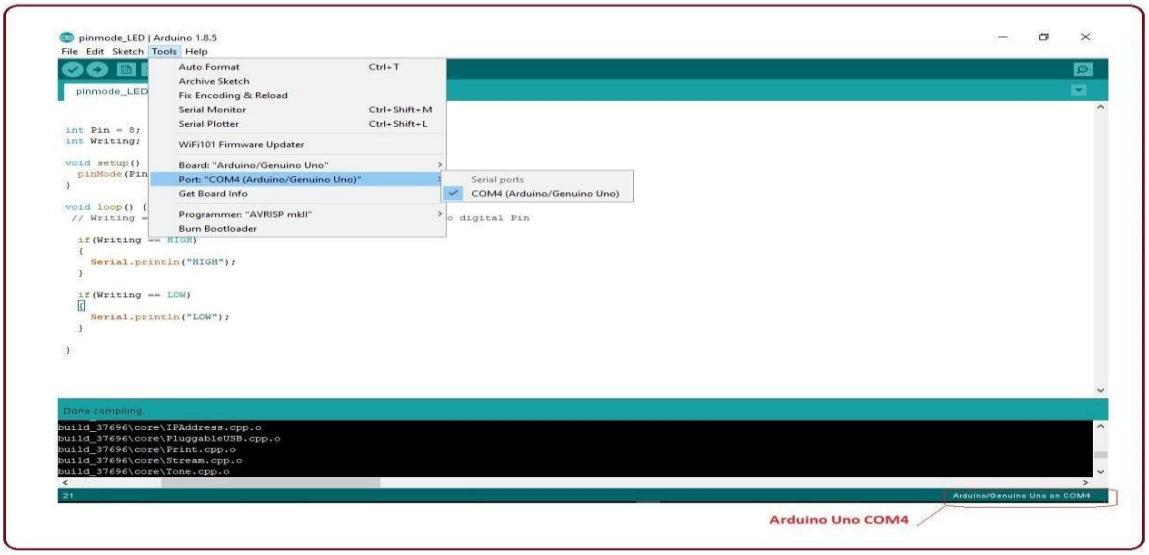


Fig 5.12: Selection of port

Note: The port selection criteria mentioned above are dedicated to Windows operating system only,you can check this Guide if you are using MAC or
Linux.

The amazing thing about this software is that no prior arrangement or bulk of the mess is required toinstall this software, you will be writing your first program within 2 minutes after the installation ofthe IDE environment.

5.1.6 Uploading

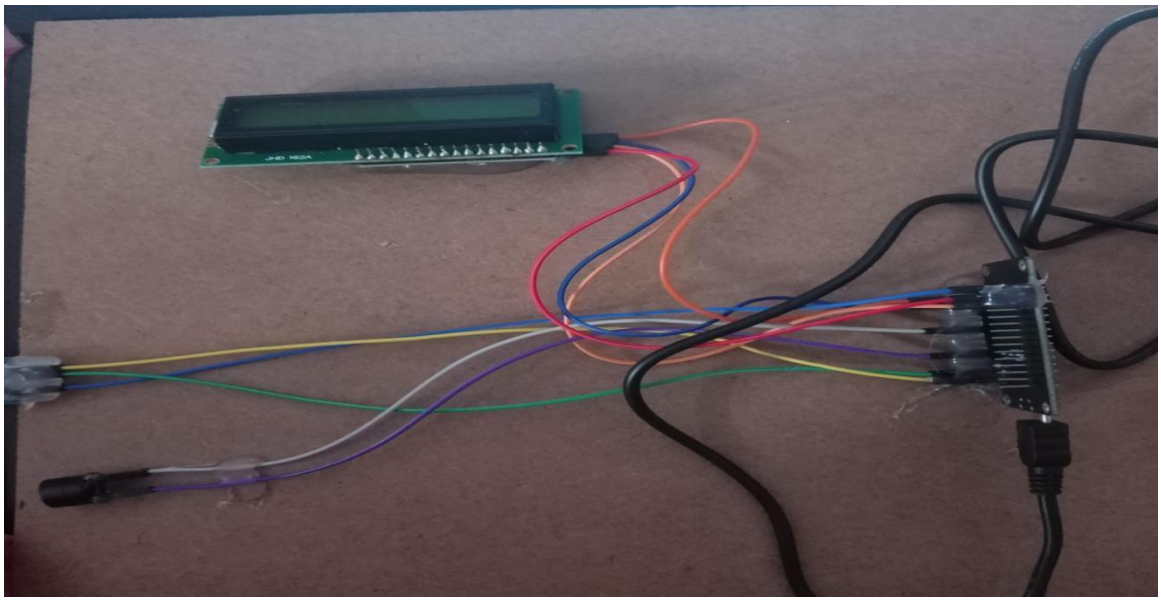
After writing your code, click on the upload button which is above the window and the code willbe directly uploaded into the Node MCU with a cable wire connector.

CHAPTER – 6

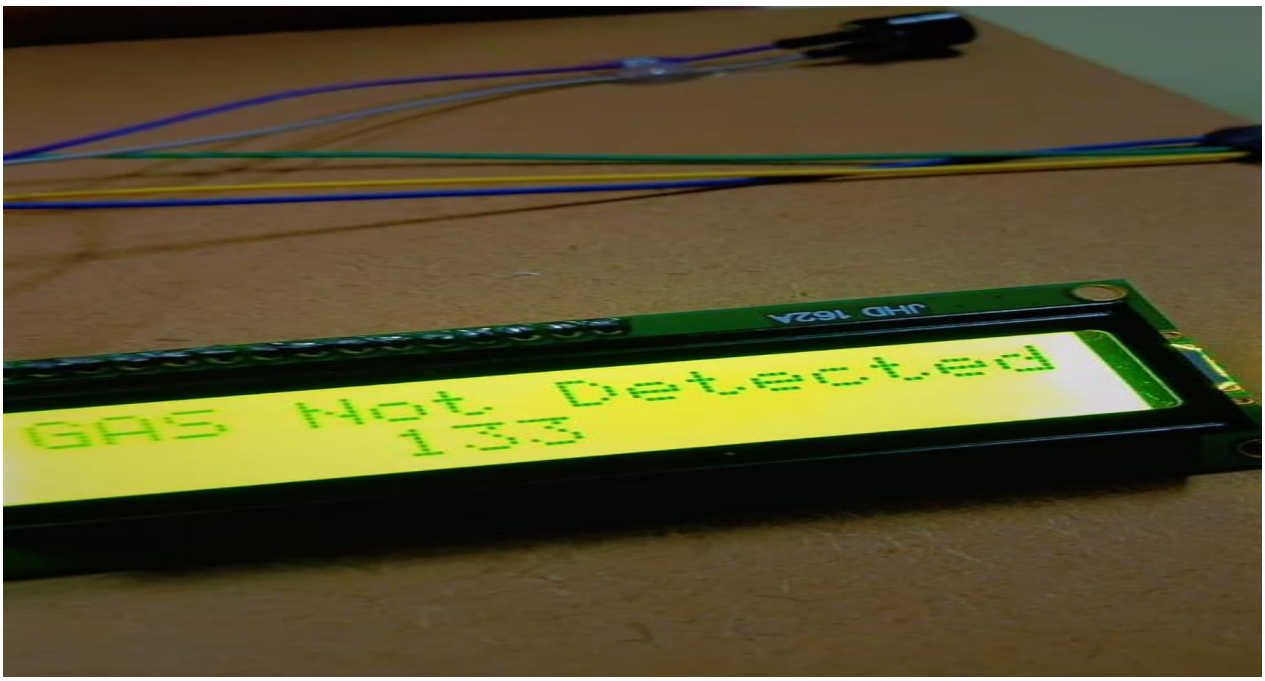
RESULT

The result of a gas leakage detection system using ESP8266, MQ-2 sensor, and a buzzer is a functional setup that can detect the presence of gases such as propane, methane, alcohol, and smoke based on the sensitivity of the MQ-2 sensor.

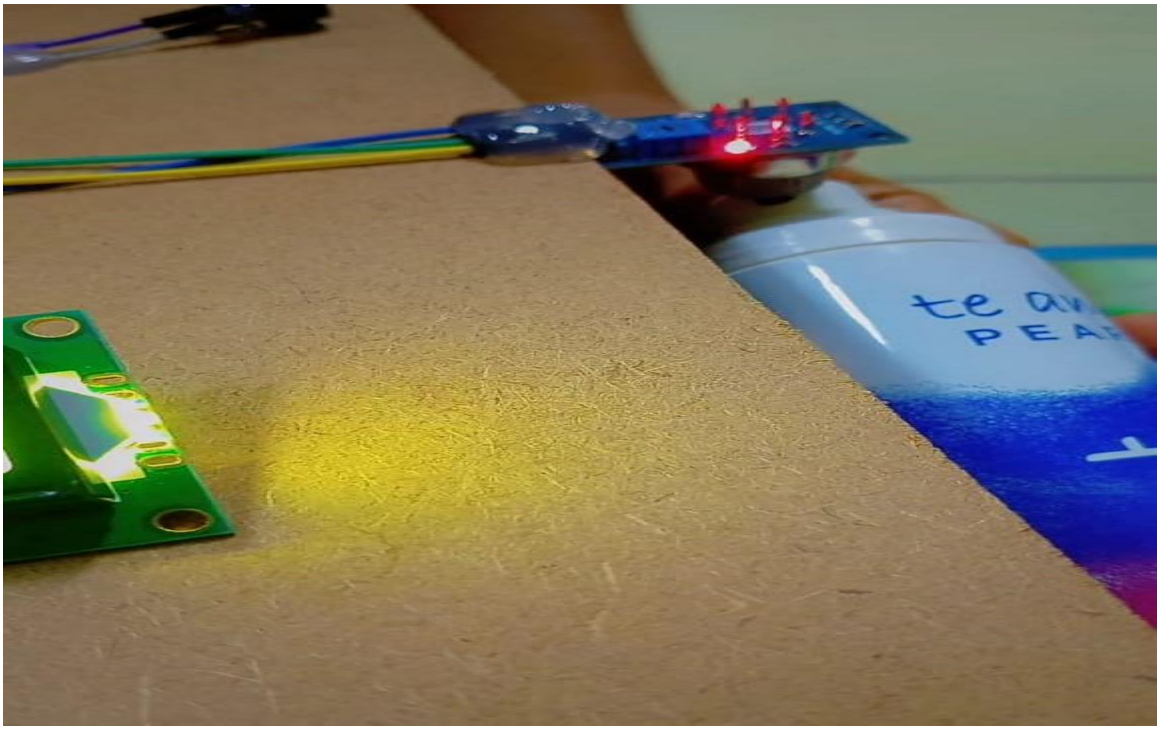
Gas leakage detection systems play a crucial role in safeguarding environments against potential hazards posed by escaping gases. These systems utilize various technologies and configurations, such as centralized monitoring with wired connections, wireless setups using WiFi or Bluetooth, IoT-enabled solutions leveraging cloud platforms for remote monitoring, portable units powered by batteries for temporary use, integration into smart home automation for seamless control, PLC-based systems ensuring robust industrial safety, and advanced multi-gas detection systems for comprehensive monitoring. By continuously monitoring gas levels and promptly alerting stakeholders through alarms or notifications, these systems enable proactive responses to mitigate risks, protect lives, and safeguard properties in residential, commercial, and industrial settings. Their versatility and integration capabilities not only enhance safety protocols but also contribute to operational efficiency and regulatory compliance, making them indispensable tools in modern safety management practices.



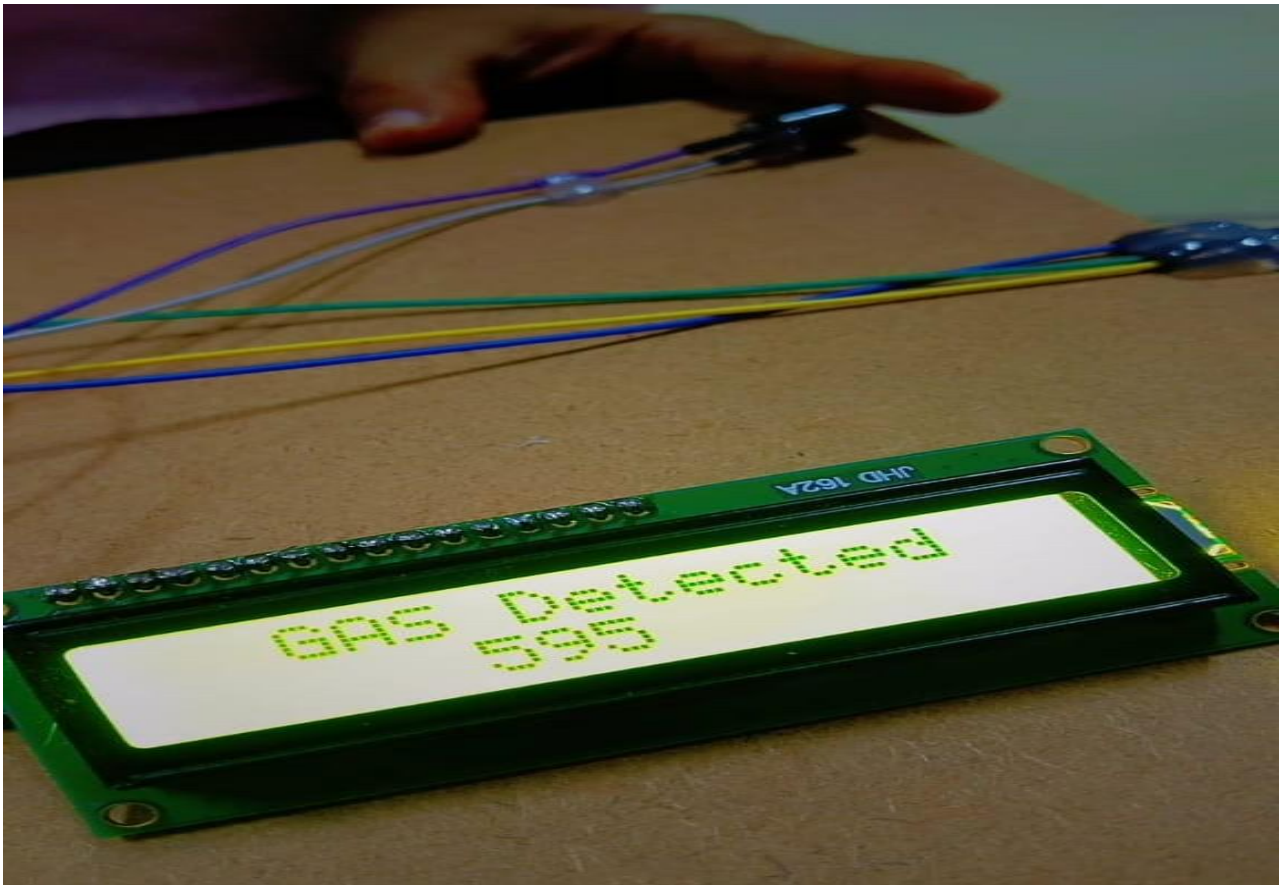
6.1 Circuit diagram



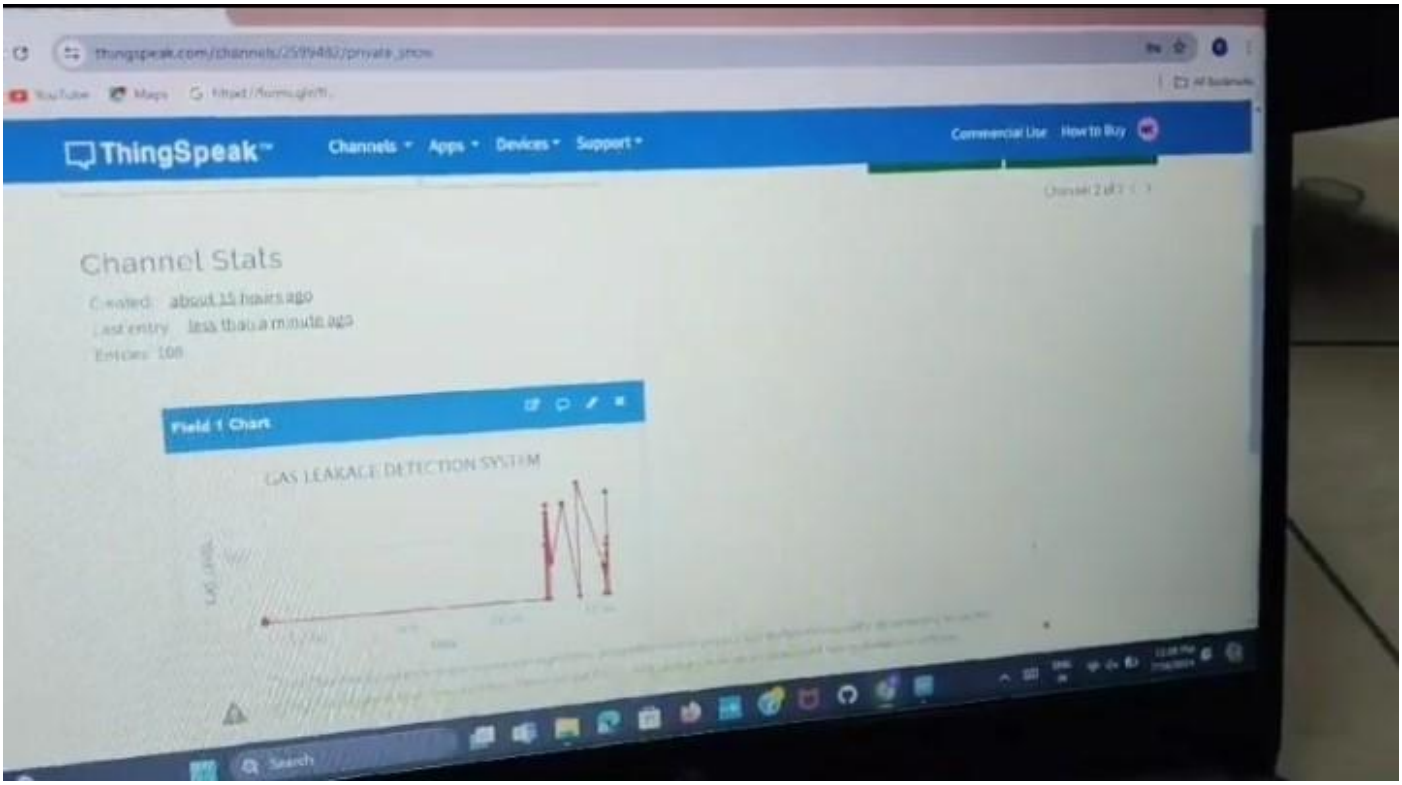
6.2 Gas not detected



6.3 Testing



6.4 Gas detected



6.5 Gas percentage values

CONCLUSION

In conclusion, gas leakage detection systems represent a critical component of safety infrastructure across residential, commercial, and industrial sectors. These systems utilize a range of technologies from traditional wired setups to advanced IoT-enabled solutions, each tailored to specific needs and environments. By providing real-time monitoring, immediate alerts, and integration with broader control systems, gas detection systems not only enhance safety protocols but also contribute to operational efficiency and regulatory compliance. As technology continues to advance, future innovations promise even greater precision, reliability, and integration capabilities, ensuring continued improvements in safety standards and proactive risk management strategies. Implementing robust gas leakage detection systems remains paramount in safeguarding human health, protecting assets, and fostering secure environments for communities worldwide.

FUTURE SCOPE

The LPG gas leakage detection and the alert system have a significant future in improving safety and preventing accidents in households and industries. With advancements in technology, the system can be further enhanced to provide more accurate and efficient detection of gas leakage. The integration of artificial intelligence and machine learning algorithms can improve the system's ability to identify potential hazards and trigger appropriate responses. Additionally, the system can be connected to a central monitoring system, enabling remote monitoring and management of gas leakage in real time. Overall, the future scope of the LPG gas leakage detection and alert system is promising, and continuous research and development can further enhance its capabilities.

SOURCE CODE

```
#include <ESP8266WiFi.h>
#include "secrets.h"
#include "ThingSpeak.h"
#include <LiquidCrystal_I2C.h> // always include thingspeak header file after other header files and custom macros

char ssid[] = "CSE_A_12";   // your network SSID (name)
char pass[] = "phani123";   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;

unsigned long myChannelNumber = 2599482;
const char * myWriteAPIKey = "1FGPP2RPN3HEVLCR";

// Initialize our values
int number1 = 0;
int number2 = random(0,100);
int number3 = random(0,100);
int number4 = random(0,100);
String myStatus = "";
LiquidCrystal_I2C lcd(0x27,16,2);
const int Sensor = A0;
const int buz = D8;
int GAS_VAL = 0;
void setup() {
  pinMode(Sensor, INPUT); // MQ-2 A0 Pin
  pinMode(buz,OUTPUT); // Buz
  lcd.begin();
  lcd.backlight();
  Serial.begin(115200);  // Initialize serial
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo native USB port only
  }
  WiFi.mode(WIFI_STA);
```

```
  ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or
WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  // set the fields with the values
  ThingSpeak.setField(1, GAS_VAL);
  ThingSpeak.setField(2, number2);
  ThingSpeak.setField(3, number3);
  ThingSpeak.setField(4, number4);
    GAS_VAL=analogRead(Sensor);
  Serial.println(GAS_VAL);
 if (GAS_VAL > 300)
 {
    lcd.setCursor(0,0);
    lcd.print("  GAS Detected    ");
    lcd.setCursor(5,1);
    lcd.print(GAS_VAL);
    digitalWrite(buz,HIGH);
  }
  else
 {
     lcd.setCursor(0,0);
     lcd.print("GAS Not Detected   ");
     lcd.setCursor(5,1);
```

```
  lcd.print(GAS_VAL);
   digitalWrite(buz,LOW);
 }
 delay(10);
 // figure out the status message
 if(number1 > number2){
  myStatus = String("field1 is greater than field2");
 }
 else if(number1 < number2){
  myStatus = String("field1 is less than field2");
 }
 else{
  myStatus = String("field1 equals field2");
 }

 // set the status
 ThingSpeak.setStatus(myStatus);

 // write to the ThingSpeak channel
 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
 if(x == 200){
  Serial.println("Channel update successful.");
 }
 else{
  Serial.println("Problem updating channel. HTTP error code " + String(x));
 }

 // change the values
 number1++;
 if(number1 > 99){
  number1 = 0;
 }
 number2 = random(0,100);
 number3 = random(0,100);
 number4 = random(0,100);

 delay(20000); // Wait 20 seconds to update the channel again
}
```

REFERENCES

[1.] Gonzalez-Martinez, A., & Aleixandre, M. (2018). IoT-based gas leak detection and localization using low-cost sensors. *Sensors*, 18(9), 3010.

[2.]Chen, C., Wang, H., Liu, Y., & Wu, J. (2019). Development of a portable gas leakage detection system based on IoT. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)* (pp. 180-184). IEEE.

[3.]Shanthini, K., & Alamelu, N. (2017). Wireless sensor network based gas leakage detection system. *International Journal of Pure and Applied Mathematics*, 116(21), 431-437.

[4.]Gao, X., Liu, Y., & Han, J. (2020). Multi-sensor fusion and intelligent decision-making for gas leakage detection systems. *Measurement Science and Technology*, 31(1), 015402.

[5.]Avci, E., Ozgur, M., & Hancke, G.P. (2019). Environmental monitoring and gas leakage detection system using IoT-based sensors. *Sustainability*, 11(5), 1250.

[6.]Lee, J. S. (2017). A review of gas sensors employed in electronic nose applications. *Sensor Letters*, 15(1), 1-10.

[7.]Mena, M., & Hernández, Á. (2020). Gas detection system integrated into a smart home environment. *Sensors*, 20(4), 982.