UNIVERSITY OF NAIROBI

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

INTERNET OF THINGS-BASED INTELLIGENT GAS LEAKAGE DETECTOR
PRJ INDEX 005

BY:
KISALI IAN KIPKOECH
F17/1341/2018

SUPERVISOR: Dr. GEVIRA OMONDI
EXAMINER: Prof H. A. OUMA, Mr K. WACHIRA

Project report submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Electrical and Electronics Engineering from the University of Nairobi

Date of Submission: 9th June 2023

DECLARATION OF ORIGINALITY

NAME OF STUDENT:	**Kisali Ian Kipkoech**
REGISTRATION NUMBER :	**F17/1341/2018**
COLLEGE/SCHOOL/FACULTY:	**Faculty of Engineering**
DEPARTMENT:	**Electrical and Information Engineering**
COURSE NAME:	**Bachelor of Science in Electrical and Electronics Engineering**
PROJECT TITLE:	**Internet of Things-based Intelligent Gas Leakage Detector**

1. I understand what plagiarism is and I am aware of the university policy in this regard**.**
2. I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.
3. I have not sought or used the services of any professional agencies to produce this work.
4. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
5. I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti-plagiarism policy.

Signature: ………………………

Date: …………………………….

CERTIFICATION

This report has been submitted to the Department of Electrical and Information Engineering, University of Nairobi with my approval as supervisor:

Sign………..……………………………… Date……………………………………

Dr. Gevira Omondi

DEDICATION

I would like to dedicate this project to my parents, siblings, friends and mentors who have played a significant role in my project journey. My heartfelt appreciation for the support, encouragement and guidance throughout my journey in engineering.

ACKNOWLEDGEMENTS

First, I would like to thank God for giving me the strength, good health and guidance throughout my studies till the accomplishment of this project.

I am grateful to my supervisor, Dr. Gevira Omondi for the immense guidance, feedback and unwavering support throughout the project. It has been a great pleasure for me to get an opportunity to work under him.

I am grateful to my parents and family members who provided the much needed emotional and financial support and encouragement.

A project of this nature could not be successful without reference to and inspiration from works of others whose details are mentioned in the reference section. I acknowledge all of them.

Lastly, I express my heartfelt appreciation to all my lecturers, classmates and friends in the department of Electrical and Information at University of Nairobi who extended help to the accomplishment of this undertaking.

ABSTRACT

The intent of this paper is to design an efficient, effective and low cost microcontroller-based intelligent gas leakage device that could be used to detect an occurrence of liquefied petroleum gas leakage and alert occupants to prevent loss of life and property. This proposed project uses an Arduino connected to gas sensor which measures the concentration of gas in air and if the limit is exceeded, a warning message will be displayed on liquid-crystal display and an alarm set on to alert residents to vacate the premises and take necessary actions to avert damages.

Keywords (microcontroller-based intelligent, Arduino, liquefied petroleum gas, sensor, liquid-crystal display)

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

LPG……………………………………… Liquefied Petroleum Gas

GSM……………………………………… Global System for Mobile communication

IoT……………………………………….. Internet of Things

ADC…………………………………….. Analog to Digital Conversion

WSN…………………………………….. Wireless System Network

PPM……………………………………… Parts Per Million

PWM…………………………………….. Pulse Width Modulation

LCD……………………………………… Liquid-Crystal Display

IC……………………………………....... Integrated Circuit

USB…………………………………….. Universal Serial Bus

DO……………………………………… Digital Output

AO………………………………………. Analog Output

GND…………………………………….. Ground

EEPROM……………………………….. Electronically Erasable Programmable ROM

LC………………..................................... Liquid Crystals

CRT……………………………………... Cathode Ray Tube

IDE………………………………………. Integrated Development Environment

UART…………………………………… Universal Asynchronous Receiver Transmitter

CHAPTER 1: INTRODUCTION

1.1 General Background

The goal of the Internet of Things is to simplify life by automating all of the little tasks around us. In as much as IoT helps in automation, the benefits can also be extended for enhancing existing safety standards in our environment. Using IoT technology to build an Intelligent Gas Leakage Detector system the environment with smart alerting techniques and protective measures is equally important for our safety.

Liquid petroleum gas – LPG, commonly referred to us as cooking gas, is a source of fuel composing of flammable mixture of hydrocarbon and an important element in our day to day living as it provides an equivalent to clean and environment friendly source of energy. Gas cylinders are considered to be the most efficient and also effective way to store and transport gas that can be utilized in commercial, home, and industrial processes.

Propylene, butylene, and other hydrocarbons like isobutene and n-butane make up a small percentage of the LPG mixture, which is predominantly consisting of butane and propane. At normal temperature, every substance is odorless and in gaseous form. For easy detection and safety, the gases are mixed with a powerful odorant called ethane thiol (commonly referred to as thiol) therefore adding odor and making the gas detectable through smell by human beings. LPG currently provide about 3% of energy consumed and burns relatively cleanly without producing soot and very minimal sulfur emissions [1].

1.2 Problem Statements

LPG is evidently a basic need in majority of Kenyan homes and LPG leakage can pose great harm to environment and people, including but not limited to explosions and fire. A practical case is in Shauri Moyo, Nairobi on September 28, 2011 where the estate was sealed off after a poisonous gas leaked emanating from metal yard leaving 30 people admitted at Kenyatta National Hospital after inhaling the gas [2].

The need to develop a cost effective Gas Leakage Detector system using innovative and smart technologies greatly influenced the development of this project. This project will aim to design and practically implement a Gas Leakage detection system that is based on Arduino microcontroller.

This system should be capable of detecting gas leakage and include protective measures such as fan and buzzer activation to inform the occupants or the facility owner through alarm and other protective measures.

1.3 Objectives

The main aim of this project is to design and develop smart Internet of Things (IoT)-based intelligent gas leakage detector.

Specific objectives include:

- To detect gas leakage such as LPG, Methane using gas sensor
- To sound an alert on occurrence of gas leakage
- To study the functioning of ESP8266 module and how to send data to Thingpeak.
- To develop and build a prototype of the Gas Leakage Detector

1.4 Scope of Project

This project is a proposal to implement a security system for detecting leakage of a gas. This system detects and monitors the levels of Liquid Petroleum Gas (LPG) using MQ-2 gas sensor and sends signal to Arduino microcontroller. The microcontroller will alert a person that there is a gas leakage using a buzzer and also sending the data to an online platform, in this case Thingspeak, using a Wi-Fi module. Simultaneously to take temporary necessary actions on gas leakage, the microcontroller will automatically switch on exhaust fan to decrease the gas concentration in the surrounding. A constraint of the project is environmental factors. The sensitivity of gas sensors is affected by temperature, humidity and other gases present in environment. A solution to this constraint is not considered in the scope of this project.

1.5 Significance of Project

The system will ensure less accidents caused by gas leaking though early notifications and necessary temporary precautions. Additionally, this study offers helpful direction for industrial practitioners to pinpoint ways to enhance the security and measures already in place in the workplace.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter provides a review of previous research and existing projects that has previously been conducted in gas leakage detection and prevention using references sources and research papers to get a scope of design, conception and any information related to improve the project. Further innovations and improvements on the project have been made with different concepts and designs.

2.2 Concept

Gas Leakage Detector with smart alerting system and prevention is an innovation from the existing Gas Detectors. Applications of this project will not only be in residential but also industrial areas to safeguard everyone from risks of inhaling dangerous gas and preventing burns. Microcontroller Arduino Uno used as the processor produces inputs from gas sensor. This detector is more efficient as fatal accidents will be avoided.

2.3 Statistics

A statistics conducted by Statistica reveals that domestic consumption of Liquefied Petroleum Gas has seen a steady increase in usage from 151 thousand metric tons in 2016 to roughly 326.2 thousand metric tons in 2020, a sharp increase of 116% [3].

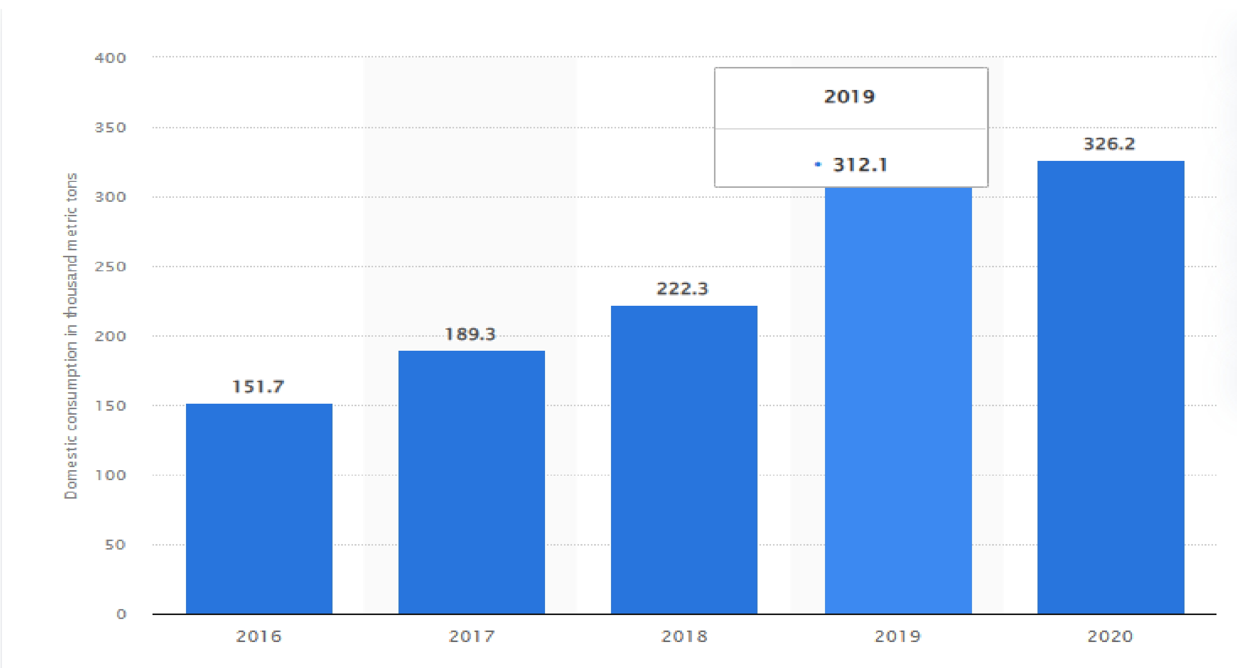


Figure 2. 1: Domestic Consumption of LPG in Kenya from 2016 to 2020

With its growing popularity and usage in urban areas, the LPG cylinder has come to symbolize rapid transformation in lives of Kenyans. It's almost ubiquitous presence in homes of both middle class and low income citizens signals change from firewood and charcoal to less smoky and clean source of energy and just as its presence symbolizes efficient cooking, it has also proved to be deadly enemy.

The Kenyan Ministry of Health recorded a significant increase in the number of burn victims in the country with a large number of cases originating from informal settlements. Incidents reported in the year 2020 surpassed 200,000 people, which had been annual average, to 213,770 compared to 73,292 cases reported in the year 2019 according information from Burns Awareness Statistical Data done by government of Kenya [4]. At Kenyatta National Hospital, gas related accidents are the leading cause of burns for admission at the hospital.

2.4 Characteristics of LPG

LPG is produced by refining crude oil, petroleum or wet natural gas obtained from fossil fuel sources. Although there are several classifications of LPG gases based on composition percentages, most of them contain primarily propane (C_3H_8), butane (C_4H_{10}), or a mixture of the two. Propylene, butylene, and other hydrocarbons, including C_2H_6, C_3H_8, are present in small quantities for detection, in addition to odorant called ethane thiol. Gas analysis methods like chromatography can be used to analyze the gas and determine the specific composition of LPG [5]. However, in general, LPG consists of:

Gas	Percentage of Total LPG
Propane	30 – 60%
Butane	10 – 30%
Isobutene	5 – 15%
Propylene	1 – 10%
Butylene	1 – 10%

Table 1: General composition of LPG

The percentages are an approximation and vary depending on the source of LPG. The composition can also change since the gases within the mixture can evaporate or react with each other depending on existing environmental conditions.

LPG has a calorie content of 46.1 MJ/kg and a boiling point of -42^{O}c, at which it becomes vapor, compared to 42.5 MJ/kg for fuel oil and 43.5 MJ/kg for gasoline. Energy content per unit of volume is 26MJ/L and relative density of about 0.5 – 0.58 kg/l of LPG is less compared to that of petroleum. As both concentration and gas pressure of LPG components vary greatly depending on underlying conditions such as temperature, this properties should be important every time the application is considered as far as safety in concerned. [1].

Since LPG's boiling point (-42^{0}c) is below room temperature, the gas easily evaporate at normal pressure and temperature hence need to store the gas in pressurized steel cylinder. The cylinders are filled to approximately 80 to 85 percent of the capacity to allow thermal expansion and the remaining volume contains vapor and pressure that varies with temperature. However the ratio of volume of gas that is vaporized and liquefied gas is different depending of factors such as pressure, composition and temperature.

LPG vapor has a density about 1.9 times that of natural air and therefore heavier than air. This is dangerous as there is a possibility of explosion if the mixture of LPG and air is within explosive limits and there is a source of fire ignition. The gas can also cause suffocation since LPG will displace air causing a decrease in oxygen concentration in air. Buildings used for LPG cylinders storage have adequate floor level ventilation and event of gas leak the vapor being heavier to air flows along the ground to lowest level and remain there.

The burning of LPG produces a greenhouse gas, carbon dioxide, and a small concentration of carbon monoxide. LPG emits 81% of carbon (IV) oxide per kWh produced by oil. The gas burns cleanly than hydrocarbons with high molecular weight since it releases few particulates compared to traditional fuel stoves. Coal is used in some developing countries and LPG is provided as an alternative, hence awareness is an important strategy to be adopted with an aim of reducing air pollution and mitigating climate change [1].

Physical data for various LPG cylinders [6]:

Capacity(kg)		**Water of Capacity(l)**	**Mass(kg)**	**Height inclusive Valve (mm)**	**Diameter (mm)**
LPG	Propane				
48.0	45.0	113.4	45.0	1288	375
19.0	18.0	45.4	21.0	890	295
14.0	10.0	34.0	17.5	720	295
9.0	8.0	22.7	13.3	545	295
6.0					295

Table 2: Composition of Different LPG Cylinders

2.5 Review of Past Related Work

Gas detectors measure and indicate gas concentration in the air using different design and technologies to prevent toxic exposure and fire. This aims to prevent air pollution which is harmful to environment and fire explosions that can lead to fatal accidents. The detectors are manufactured as either portable or stationary devices and work by signifying high gas levels using audible or visible indicators such as alarms and light signals. While many of older standard gas detector devices were used it is important to improve on the existing technologies for faster detection and action on an event of leakage. There are various types of detectors in the market which serves same function to monitor and warn of dangerous LPG gas levels.

2.5.1 Prototype of Gas Leak Detector

In [7], with the help of an Arduino UNO and SIM900 GSM/GPRS gateway, a system for gas leakage detection was created to alert people on gas leaks via Short Message Service (SMS) to the phone of the person to be contacted..

The procedures of the gas leak detector were as follows: to monitor parameter of LPG leak using MQ2 Gas Sensor, all parameters obtained from sensor is sent to the microcontroller (Arduino UNO) using a cable. The microcontroller will process parameters and give warning message and alert users via alarm when input data shows existing gas leak in air. The alarm will be turned on.

This prototype however does not include instant measures to be taken once there is a gas leakage. Such measures include activating a fan and opening of window or an air outlet to reduce concentration of LPG gas in the surrounding. The design also does not include an LCD to display the concentration of gas to the occupants.

The workflow process architecture of the system is as shown:

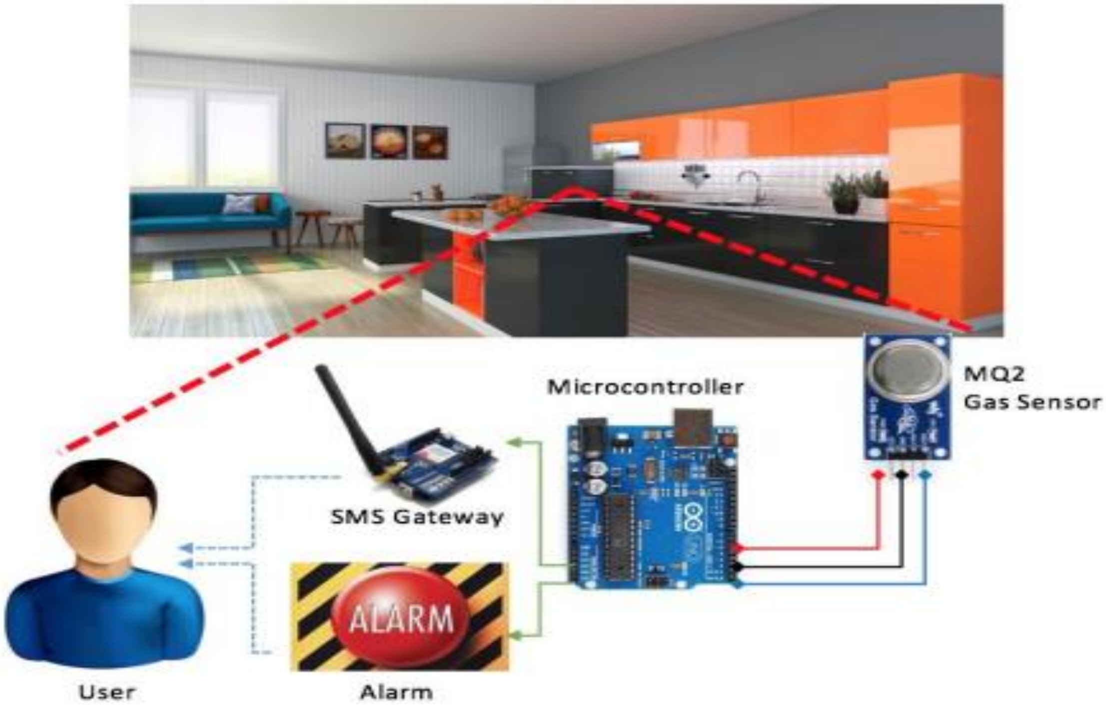


Figure 2. 2: System Architecture Design

2.5.2 Wireless Sensor Network System for LPG Gas Leakage

The methodology used in this research [8] employed use of a gas sensor, an Arduino UNO microcontroller to create a visual LabVIEW tool that enabled accurate predictions and indicate the gas leakage concentration. The system is an improvement as it involved use of a wireless method of gas leakage detection and at a lower cost, low power and broader coverage for detecting and monitoring a small concentration of gas leak.

The output voltage of a gas sensor is calculated in LabVIEW in terms of leaked gas concentration using ZigBee, making it simple to monitor and display on the web. When MQ-2 gas sensor detects presence of LPG, it sends an analog signal to ADC converter of Arduino microcontroller which processes the signal and transfers the values to LabVIEW using X Bee.

The LabVIEW platform was used as a software and hardware interface so that on the set thresholds, the sensor will keep sensing leakages which is remotely monitored.

This study provided a different approach by incorporating WSN to gas leakage detection by use of LabVIEW therefore enabling monitoring of gas via software. The design also incorporated a phone service using GSM module where users receive SMS notification alerts in case of gas leakage. The research was successful in creating wireless system network using both internet and software tools using ZigBee wireless system.

2.6 Internet of Things

The Internet of Things refers to a network of interconnected physical devices such as vehicles, home appliances and other devices that are embedded with electronic gadgets such as sensors, micro-controllers and software enabling connectivity and communication with each other over the internet. IoT devices include smart homes, autonomous vehicles, wearables and the primary aim is to enhance efficiency, productivity and convenience in various sectors.

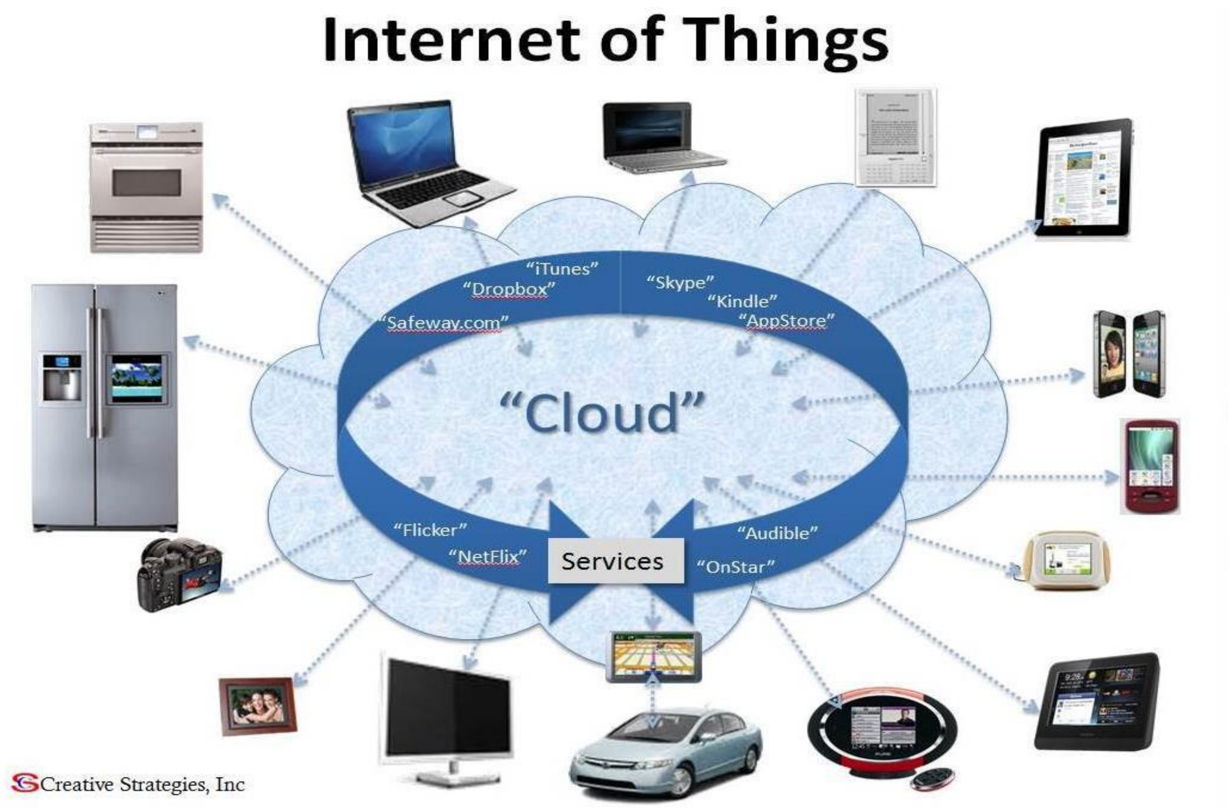


Figure 2. 3: Internet of Things

A fundamental advantage of IoT is its potential to automate various processes hence improving efficiency and reducing costs in sectors such as healthcare, agriculture, energy, industry and automotive. For example, IoT-enabled agriculture systems can use data on soil moisture, temperature and weather patterns to optimize crop yields, reduce water usage and limit use of harmful pesticides.

IoT, however, poses considerable risks and obstacles. Security is one of the key issues brought up because there are so many connected gadgets on the internet, which leaves systems open to hacker infiltrations, attacks and data leaks. Furthermore, jobs that were formerly performed by people may become redundant due to the rapid growth of technology and automation, which will significantly alter the job market.

This technology can be used in disaster management to help mitigate impacts of disasters, improving response time and reducing loss of life and property. Such applications has a wide range including early warning systems, monitoring critical infrastructures such as bridges and smart emergency response systems. Employing warning systems such as Gas Leakage Detection is an application of IoT in disaster management and control reducing loss of life and property.

2.7 Description of Materials

2.7.1 Arduino Uno Board

A microcontroller is a small type of computer on an Integrated Circuit (IC) designed to control specific function or an application. It combines function of a microprocessor, which processes data and instructions, with input/output interfaces, memory and other peripherals.

Arduino is an open-source hardware and software platform designed to build devices and applications specific to an environment. It consists of a circuit board which can be programmed (Arduino Uno) and a software tool called Arduino IDE (Integrated Development Environment) used to write and upload code to the physical board. [9]

An Arduino Uno is a type of Arduino microcontrollers designed for building electronic projects and can be used for various applications. The board is compatible with variety of programming languages, including C and C++ hence flexible choice for developers.

The important components that make up an Arduino board and their functions are:

- Reset Button – Restarts any code loaded in the Arduino
- Digital Input/output – Pins 0-13 can be used for input or output
- USB Connection – Used for powering up Arduino and uploading sketches
- ATMEGA Microcontroller – Brains of board and stores the programs.
- Analog Pins (A0 - A5) – Can read signal from an analog sensor and convert to digital value
- Voltage Regulator – Controls the amount of voltage going into Arduino Board
- TX/RX – Transmit and receive data indication LEDs.
- 5V and 3.3V that supply the respective voltage
- AREF – Analog reference used to set external reference voltage between 0 and 5V.

The board is an 8 bit 16MHZ which can support 14 digital inputs and output pins, 6 PWM and 6 analog pins, each pin having a specific function. Non-volatile storage and EEPROM is used for storage. The primary distinction between non-volatile storage and EEPROM is that non-volatile storage contents are easily erasable unlike EEPROM. The availability of the memory during which blocks are divided and thus portion is erased block by block, where the erased portion is provided for the EEPROM byte. [9]

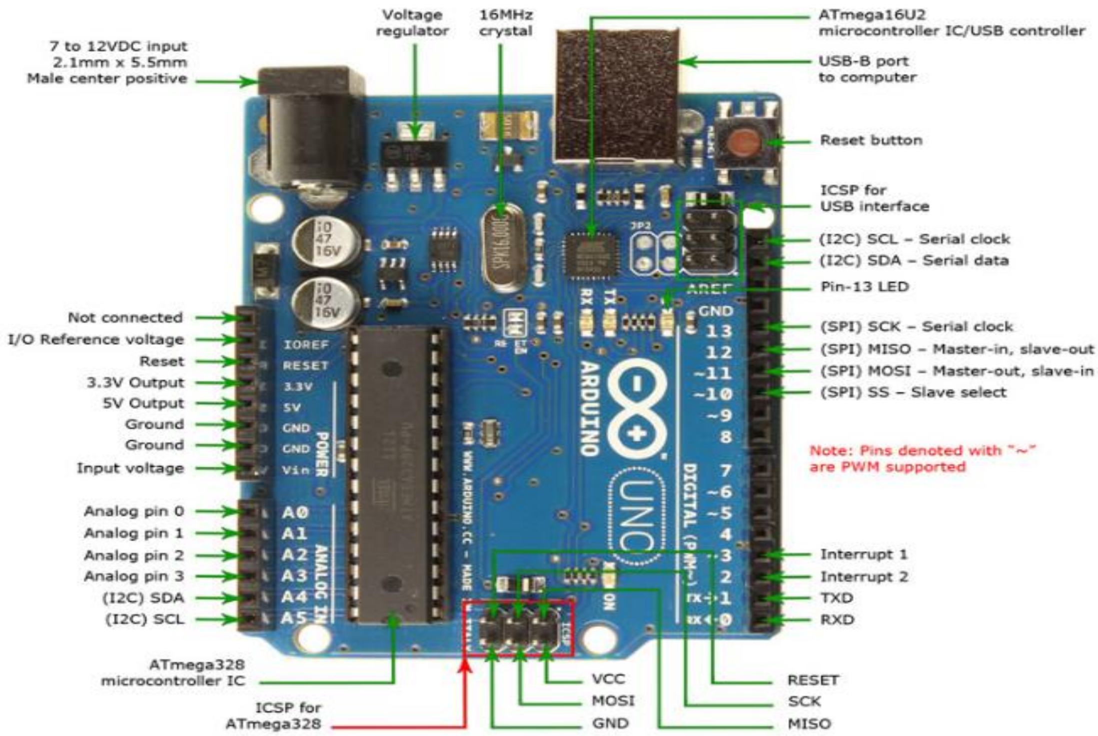


Figure 2. 4: Arduino Uno Board

2.7.2 ESP8266 Wi-Fi Module

This module is a low-cost, powerful and versatile module that is becoming a popular choice for building IoT and smart devices. It is a small, single-chip system that combines a powerful 32-bit microcontroller with built in Wi-Fi capabilities. The module is significantly less expensive and can be used in a wide range of projects. [10]

A distinct feature of the module is its built-in Wi-Fi capabilities with wider range of abilities including 802.11 b/g/n Wi-Fi standards, WPA/WPA2 security and access to point and station modes making it easier for the module to receive and send data over the internet.

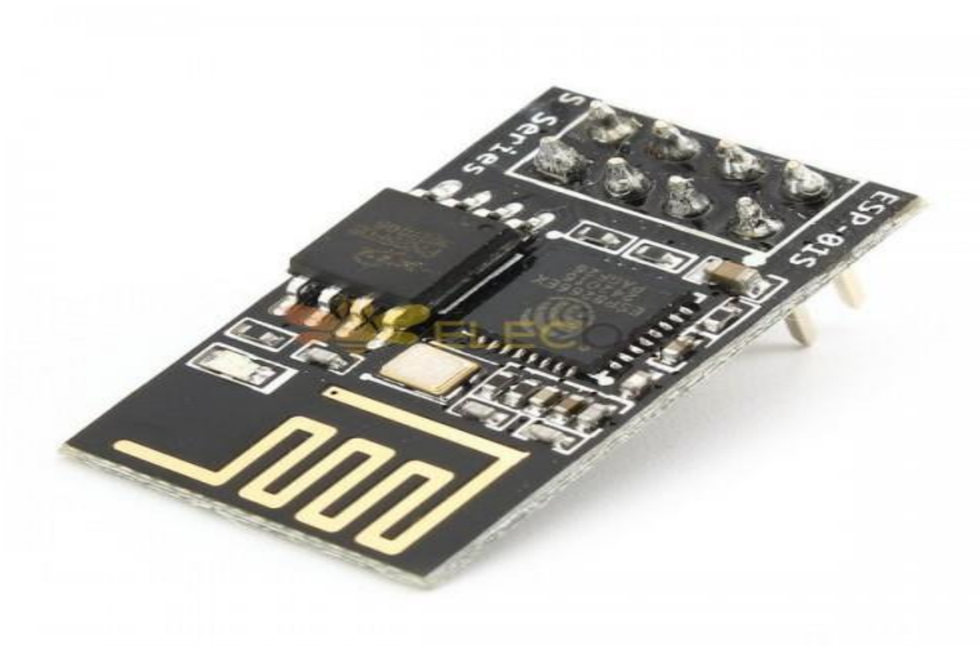


Figure 2. 5: ESP8086 Wi-Fi Module

2.7.3 MQ-2 Gas Sensor

This is a type of gas sensor that is versatile and can be used to detect LPG, smoke, propane, carbon monoxide and methane concentration in air. The sensor works by detecting presence of a specific gas in the air and providing an output signal that can be used to activate a device or trigger an alarm.

The sensor is a Metal Oxide Semiconductor (MOS) that detects gas based on the change in resistance of the sensing material when the gaseous elements pass through the chamber.

Figure 2. 6: MQ-2 Gas Sensor

The 800mW-powered MQ2 gas sensor can detect concentrations between 200 and 10,000 ppm and runs on a 5V DC power supply. The ratio of one gas to another is measured in ppm. For instance, 800 parts per million of LPG means that out of a million gas molecules, 800 of them are LPG and the rest are composed of different gases. The sensor detects a variety of gases but cannot differentiate them. Instead of detecting the specific gas, it is most effective in detecting changes in a specific gas's density in air, in this case LPG. [11]

The sensor gives a binary signal for the presence or absence of LPG and an analog signal for the amount in the atmosphere. In direct proportion to the gas concentration, the sensor's analog output voltage (AO) changes. In general, output voltage increases with LPG concentration, while output voltage decreases with LPG concentration. Using an LM393 comparator, the analog signal is transformed into its digital equivalent and made available at the digital output (DO) pin. A potentiometer is also included in the module, which is used to modify the sensitivity of DO and define a minimum threshold above which gas concentration is HIGH and below which it is LOW. Additionally, there are GND and VCC, which represent the module's ground and power supply, respectively. When the module is turned on, the Power LED illuminates, and the Status LED goes on when the gas concentration reaches the threshold value. [11]

Quantity	Values and Limit
Operating Voltage	5V
Load Resistance	20 KΩ
Heater Resistance	33Ω ± 5%
Heating Consumption	<800mw
Sensing Resistance	10 KΩ – 60 KΩ
Concentration Scope	200 – 10000ppm
Preheat Time (hrs.)	>24

Table 3 Operating Properties of MQ-2 Sensor

2.7.4 Extractor Fan

An Arduino extractor fan is used as a ventilation system designed to reduce concentration of contaminated air from an enclosed space. The fan is connected to Arduino Uno board allowing it to be controlled remotely. This allows the fan to be turned on and off depending on the condition, in our case LPG concentration in the air. This will temporarily contain a situation of gas leakage hence minimizing damages.

The working principle of an Arduino extractor fan is based on feedback control system. Arduino is programmed to read data from MQ-2 sensor to determine when the fan will be turned on or off based on the set threshold. The fan is connected to a motor driven circuit that provides necessary voltage and current to make the fan rotate. The Pulse Width Modulation (PWM) signal provided to the motor driver circuit controls the speed of the fan.

Figure 2. 7: Arduino Extractor Fan

2.7.5 LCD Screen Display

According to Mart Clary [12], a Liquid-Crystal Display (LCD) is a video display that utilizes light modulating properties of liquid crystals to display pictures or texts in a screen.

A flat panel display, electronic visual display, or video display that uses liquid crystals (LCs) to modulate light is known as a liquid crystal display (LCD) [13]. LCs are utilized in a variety of applications, including computer monitors, televisions, signage, etc. They do not directly emit light. They are common in consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones. LCDs have replaced cathode ray tube (CRT) displays in most applications. They are available in a wider range of screen sizes than CRT and plasma displays, and they cannot suffer image burn-in since they do not use phosphors.

However, image persistence might influence LCDs. In comparison to CRTs, LCDs are safer to dispose off and have higher energy efficiency. It can be utilized in battery-powered electronic devices due to its low electrical power consumption. It is an electronically modulated optical device that creates color or monochromatic images by arranging any number of segments filled with liquid crystals in front of a light source (backlight) or reflector. The most adaptable ones make use of a collection of tiny pixels. LCDs come in two different models: Graphics LCD and LCD with characters. The LCD's character displays ASCII data and images Displays graphics on LCD [13].

It is possible to set up the Arduino microcontroller to display the level of LPG in the air and notify residents of the presence of a gas leak by connecting the microcontroller to the pins of an LCD display.

Figure 2. 8: 16x2 LCD Screen

2.7.6 SG90 Servo Motor

Servos are types of motors that allow precise control of physical movement since they generally move a position rather than a continuous rotation and are simple to connect and control since the drivers is in-built. The rotary actuator allows control of angular position, velocity and acceleration. It is a micro-servo motor meaning they are small-sized with dimensions of about 23 x 12.2 x 29mm and a weight of around 9 grams. The motor operates at a voltage range of 4.8 to 6 volts and can rotate up to 180 degrees.

By comparing the motor's present position to its desired position, the potentiometer delivers position feedback to the error amplifier in the control unit. The control unit then adjusts the motor’s current position matching the desired position. The mechanism is called servomechanism or servo in control engineering.

The motor has three wires, orange connected to PWM, red connected to VCC and brown connected to GND.

Servos include a tiny DC motor that is coupled to the output shaft through gears and powers the servo horn which being attached to a potentiometer, as shown in [14]:

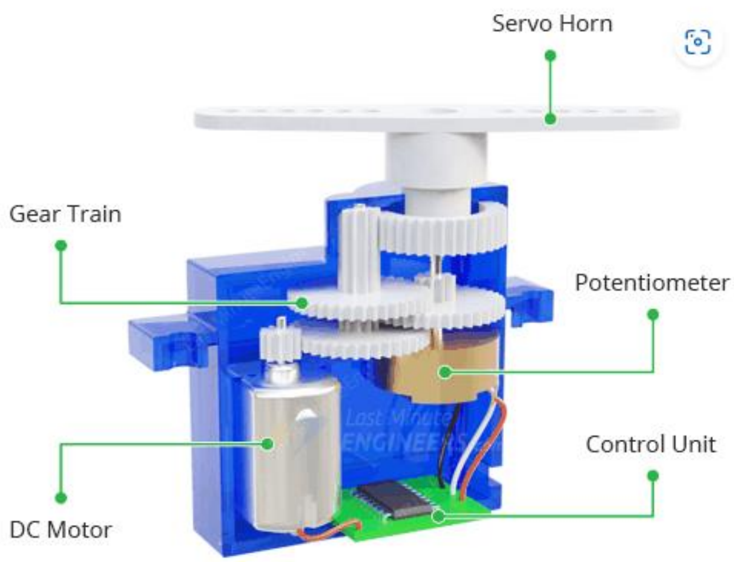


Figure 2. 9: SG90 Micro-servo Motor

2.7.7 Piezoelectric Buzzer

There are many ways of communicating between the occupants of an area and the gas leakage detector system and in the case of an audio communication, a buzzer would be an important technology. A buzzer is an audio signaling device that converts signal from audio to sound. It is powered using DC voltage and in this case, it is used to generate alarm sound in case of a gas leakage.

The figure below shows the pin configuration of buzzer:

Figure 2. 10: Pin Configuration of Piezoelectric Buzzer

Buzzer consists of two pins, positive and negative represented as '+' or longer terminal and powered through 6 volts while the negative terminal is represented with '-'or short terminal and connected to GND [15].

Specification is buzzer include [12]:

Property	Specification
Frequency	3300Hz
Operating Temperature	-200C - +600C
Operating Voltage	3V – 24V DC
Sound Pressure Level	85dBA OR 10 cm
Supply Current	<15mA

Table 4: Operating Properties of Piezoelectric Buzzer

CHAPTER 3: DESIGN AND IMPLEMENTATION

3.1 Introduction

The design process involved determination of the appropriate microcontroller, actuators, display and sensors. The microcontroller, Arduino Uno, is the brain or hub which is the interface of the components used in the project.

The design is based on two aspects:

- Detection of gas leakage using MQ-2 gas sensor
- Intervention measures to contain gas leakage

The following aspects were considered in designing the detector:

1. Communication from the microcontroller to users
    - Buzzer to produce siren sound on gas leakage
    - LCD screen to display gas concentration to occupants
    - ESP8266 sending data (gas concentration in PPM) in real time to Thingspeak platform for graphical representation of the data.
    - Red LED to warn users
2. Intervention measures by the microcontroller
    - Extractor fan used to reduce gas concentration in air on occurrence of a leakage
    - Servo motor to open an emergency window to also reduce LPG gas

The components interfaced to the Arduino is shown:

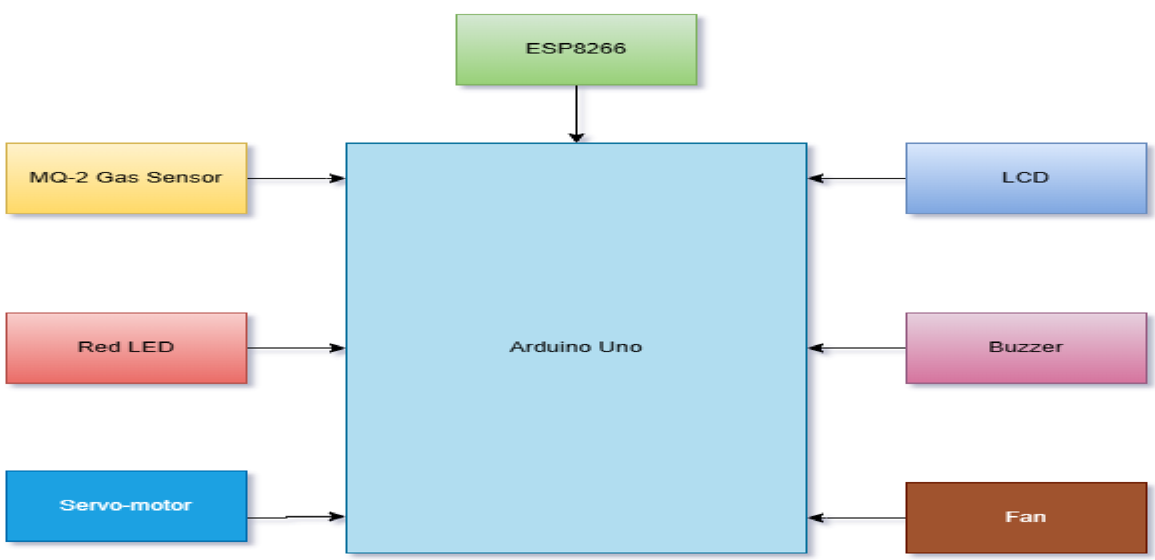


Figure 3: 1 Interfacing Arduino with peripherals

3.2 Software Configuration

3.1.2 Arduino Integrated Development Environment

The Arduino software (IDE) allows one to write programs or sketches and upload them to the board. There are two options of using the software:

- Online IDE also called Arduino web editor allows one to save sketches in cloud and have them available from any device and backed up
- To work offline, one has to install the latest version of desktop IDE depending on the operating system of the computer.

The programs that are written using Arduino IDE are called sketches, which are written in text editor and saved with a file extension .ino. The console displays text output while toolbar allows verification and upload of programs, create, open, save sketches and open the serial monitor. The software can be downloaded from the link [16].

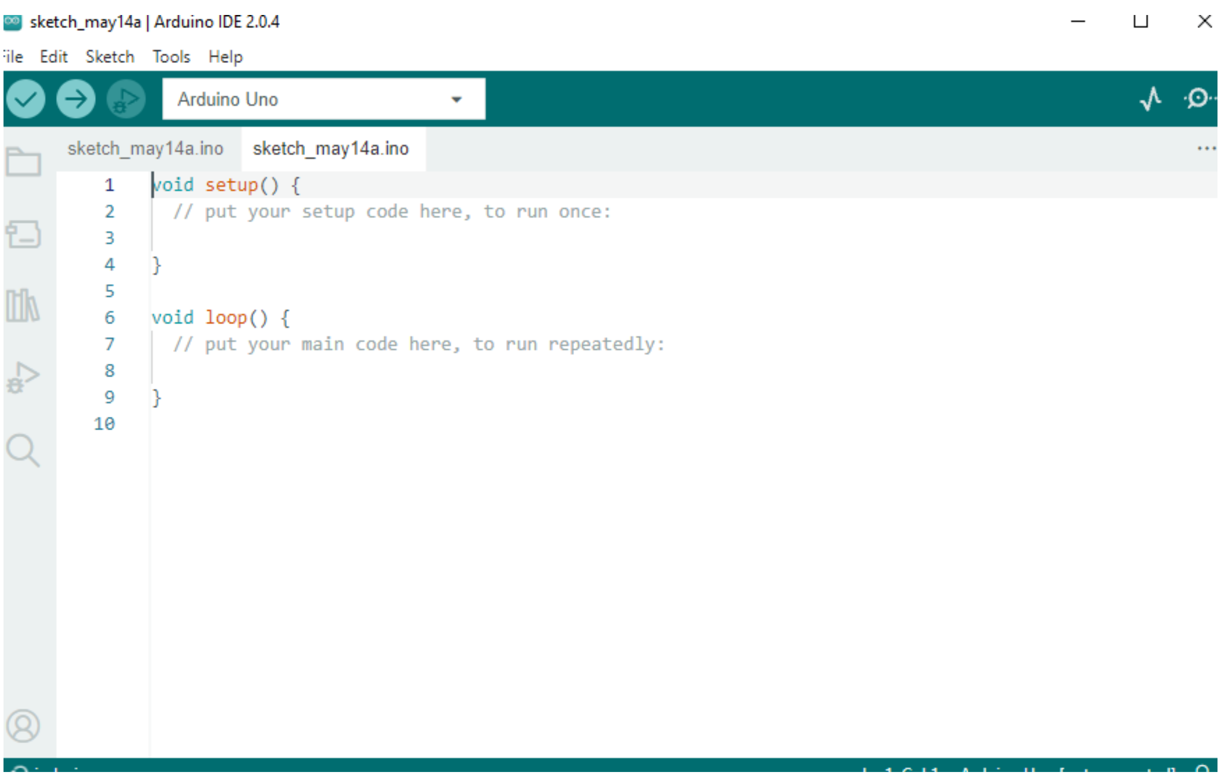


Figure 3: 2 Arduino IDE with Bare Minimum sketch file

Before uploading any sketch, ensure the correct port and serial board is used by the IDE to the hardware connected (Arduino UNO). Libraries provide extra functionalities for use in sketches and are inserted using #include <LibraryHeaderFile>. Libraries used in this design include SoftwareSerial for serial communication on the digital pins, LiquidCrystal for controlling liquid crystal displays and Servo for controlling the position of servo motor used.
The sketch uploaded to IDE for the LPG leakage is attached as Appendix C.

4.1.3 Thingspeak Configuration

This is an Internet of Things platform for collecting, analyzing and visualizing gas levels from the connected MQ-2 gas sensor. It provides a cloud based platform enabling users store the data in cloud and analyze using range of tools and visualizations. It supports various communications protocols including HTTP, MQTT and UDP allowing transmission from wide range of devices. In order to communicate data and retain incoming gas concentration, Thingspeak requires a user account and a channel. Data gets transmitted every 15 seconds and a channel can have up to 8 data fields, location fields, and status fields.

- Create an account for a new user. – https://thingspeak.com
- Create a new *Channel* by clicking on *Channels*, *My Channels*, and *New Channel*
- Fill in name and description of the project and the field parameter to be displayed for example *LPG Sensor value* then save the channel

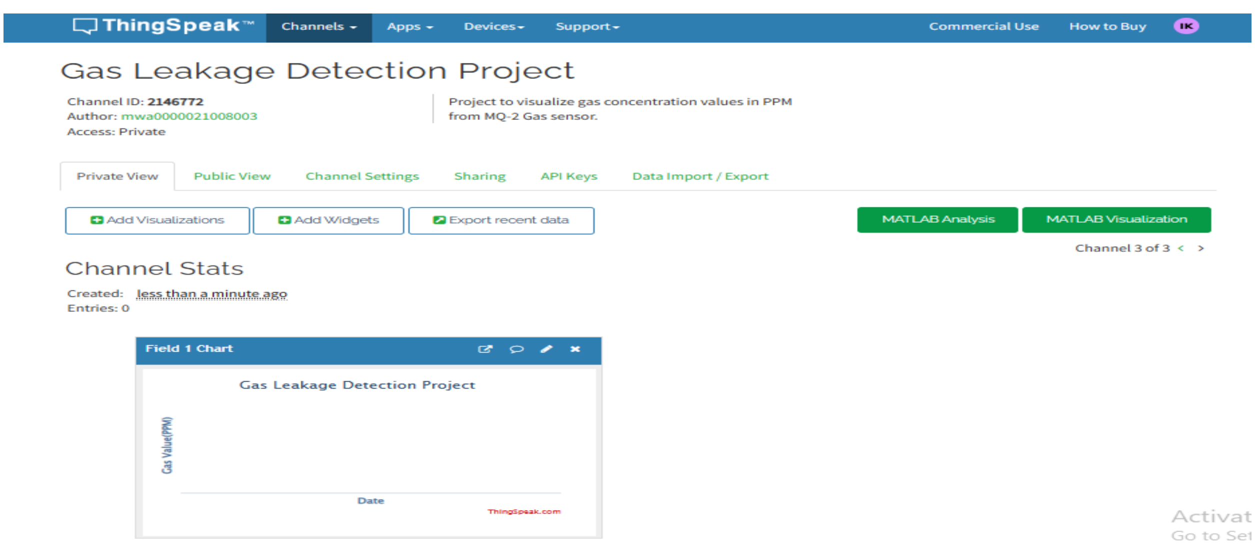


Figure 3: 3 Thingspeak Configuration to Visualize Gas Concentration

The channel created is linked to my ESP8266 module from the code attached in Appendix C. This channel created will display gas values obtained from the sensor in real time with a delay of 15 seconds. API keys enable writing of data to the channel or reading data from a private channel. Write API key is used in this case and used in the code to link the channel and gas sensor. Thingspeak uses the GET HTTP protocol to enable transfer of data. The command used to access Thingspeak API is:

```
GET https://api.thingspeak.com/update?api_key= UV0TOP5EXKABH1XF&field1=0
```

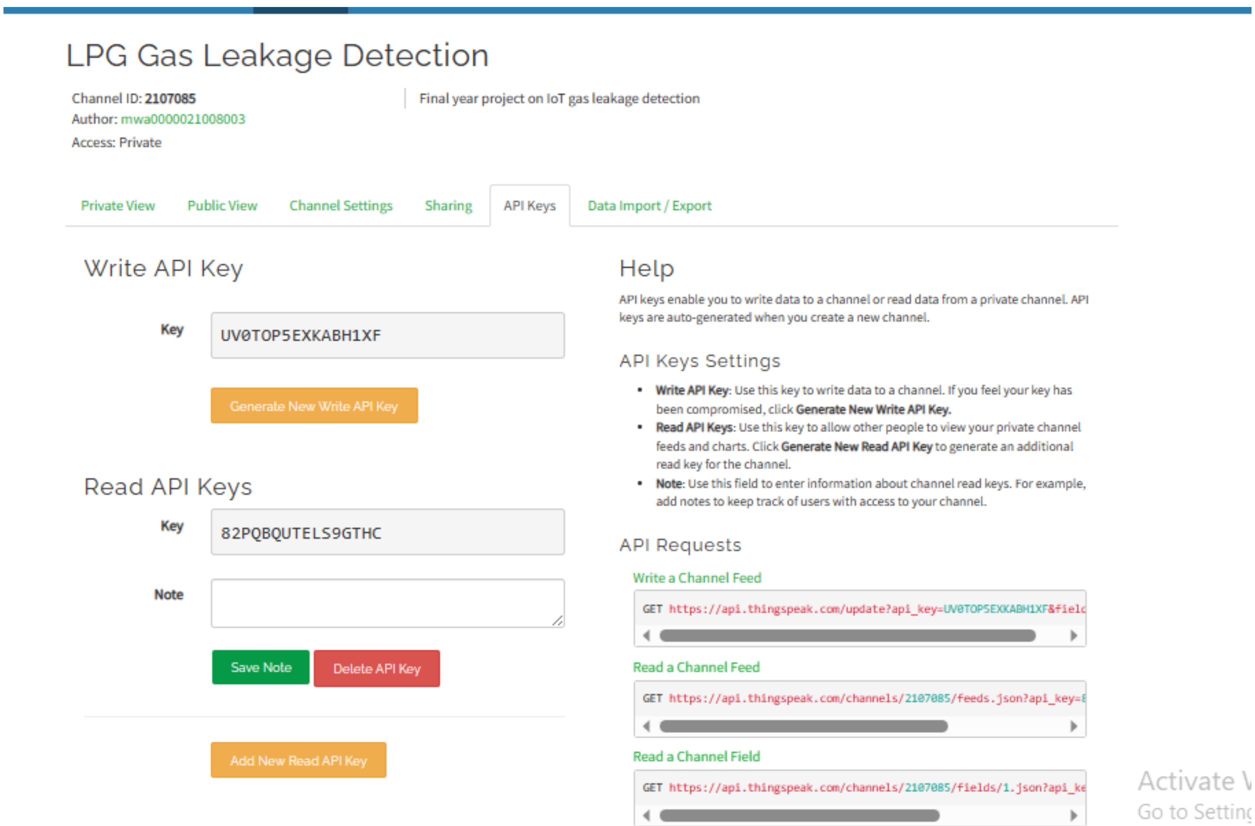


Figure 3: 4 Thingspeak API using GET HTTP Protocol

3.3 Hardware Connection

The ESP8266 Wi-Fi module is initially connected to the Arduino through the UART, and the microcontroller can then control the ESP8266 (Wi-Fi connection, firmware update, etc.) using AT commands. In order to use the module to upload gas concentration to Thingspeak, the module should first be connected to Wi-Fi then accessing it from the internet.

ESP8266 can be programmed using Arduino IDE but some changes have to make on the IDE. Go to File -> Preferences in IDE and in the Additional Boards Manager URLs section enter the following link:

https://arduino.esp8266.com/stable/package_esp8266com_index.json

Now, go to Tools -> Board -> Boards Manager and search for ESP8266 and install the packages. The module has three different Wi-Fi operation modes, including [17]:

- **Station Mode (STA)** – A network that has already been set up by an access point, such as a Wi-Fi router, is connected to the module.
- **Soft Access Point (AP)** – Users can access the internet through the module, which serves as an access point. Before configuring the module in station mode, it is first set up in soft AP mode.
- **Soft AP + Station** – The module is set up to operate in soft AP mode and station.

The module works on a 3.3V power supply and consists of 8 pins which are VCC, GND, TX, RX, RST, CH_PD, GPIO0, and GPIO2. Both VCC and CH_PD pin of the ESP8266 are connected to 3.3V pin of the Arduino, GND pin connected to GND of Arduino and RST connected to GND pin of Arduino via a pushbutton. This is used for resetting the module. Arduino Pin 2 is connected to TX pin of ESP8266 and Pin 3 connected to RX pin of module. After connecting the module in programming mode and with necessary firmware installed, the Baud Rate of Arduino IDE and “Both NL & CR” options are selected in the serial monitor to enable the module receive instructions and be able to send data to the platform for visualization.

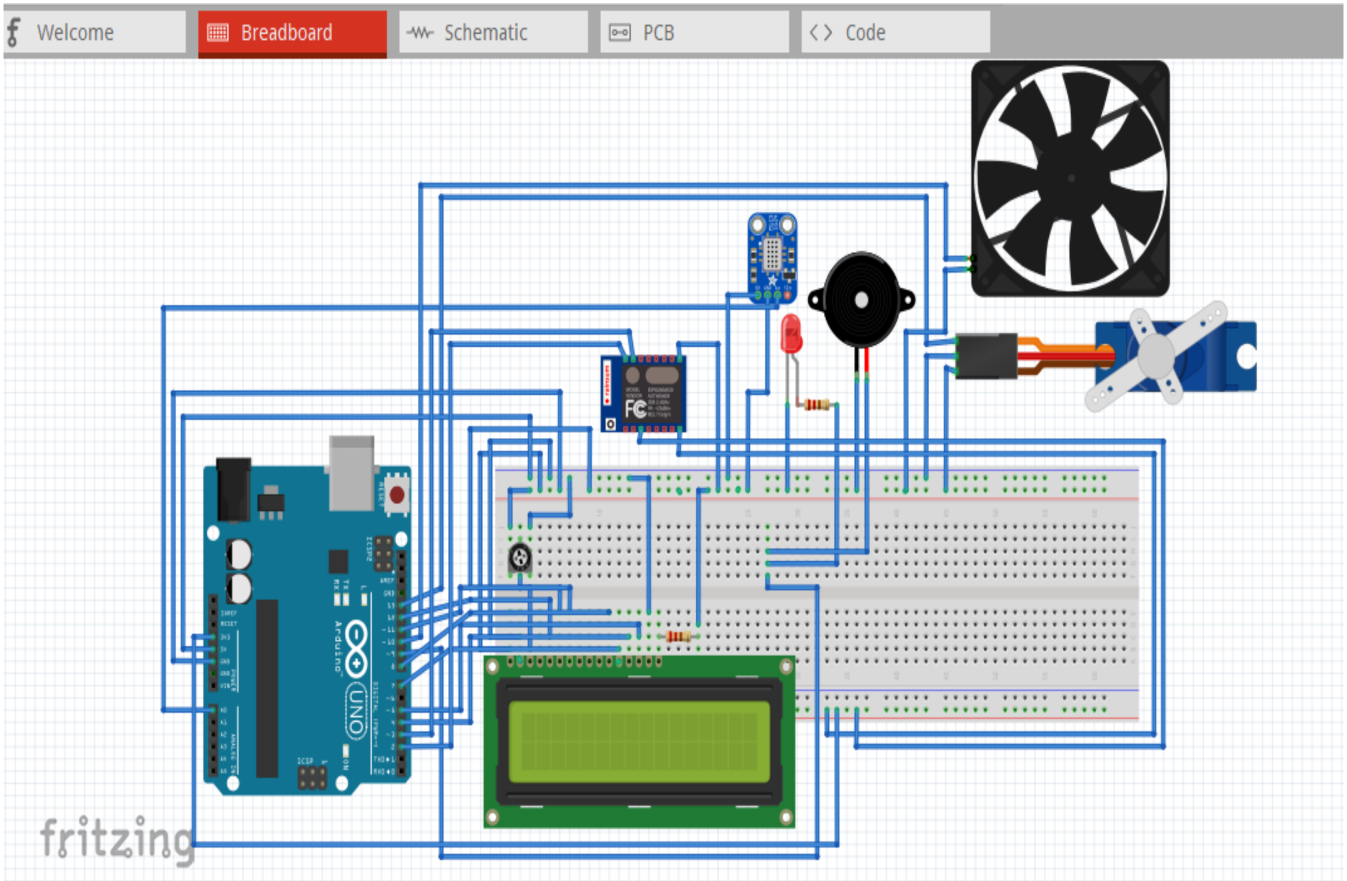


Figure 3: 5 Simulation using Fritzing Software

MQ2 Gas sensor module has four pins, VCC, GND, A0 and D0. VCC pin supplies power to the module and is connected to 5V of Arduino while the GND pin is connected to GND pin of Arduino. A0 produces analog output voltage that is proportional to gas concentration, with higher voltage resulting from higher concentrations and lower voltage resulting from lower concentrations. The pin is attached to Arduino's analog pin A0. D0 is not connected since it only indicates the presence or absence of LPG leakage yet we want to determine the concentration and minimize the effects of a leakage. DO becomes LOW when gas levels exceed value set by potentiometer and HIGH otherwise.

LCD has a total of 16 pins on the display module. Two pins are for power (VCC and GND) connected to 5V and GND of Arduino respectively. Pin VO/VEE is used for contrast adjustment and connected to a variable resistor, in this case a potentiometer. The output of potentiometer is connected to this pin and rotating the knob clockwise or anti-clockwise will adjust the contrast of the LCD. Pin RS selects command registers when low and data register when high is connected to pin 12 of Arduino and R/W pin to read or write to register connected to GND. Enable pin EN sends data to data pins with high to low pulse connected to pin 11 of Arduino. Pins D4, D5, D6 and D7 of LCD are data pins connected to pins 8, 7, 4 and 5 of Arduino respectively. LED backlights are connected to GND (0V) and 5V, respectively, on the A and K pins.

Piezo buzzer has two pins, a positive and negative. The positive pin with a longer pin is connected to pin 9 of Arduino while negative to the GND via a 100ohm resistor. Positive terminal of fan connected to pin 10 of Arduino. The SG90 motor has three wires, orange connected to PWM, red connected to VCC and brown connected to GND of Arduino.

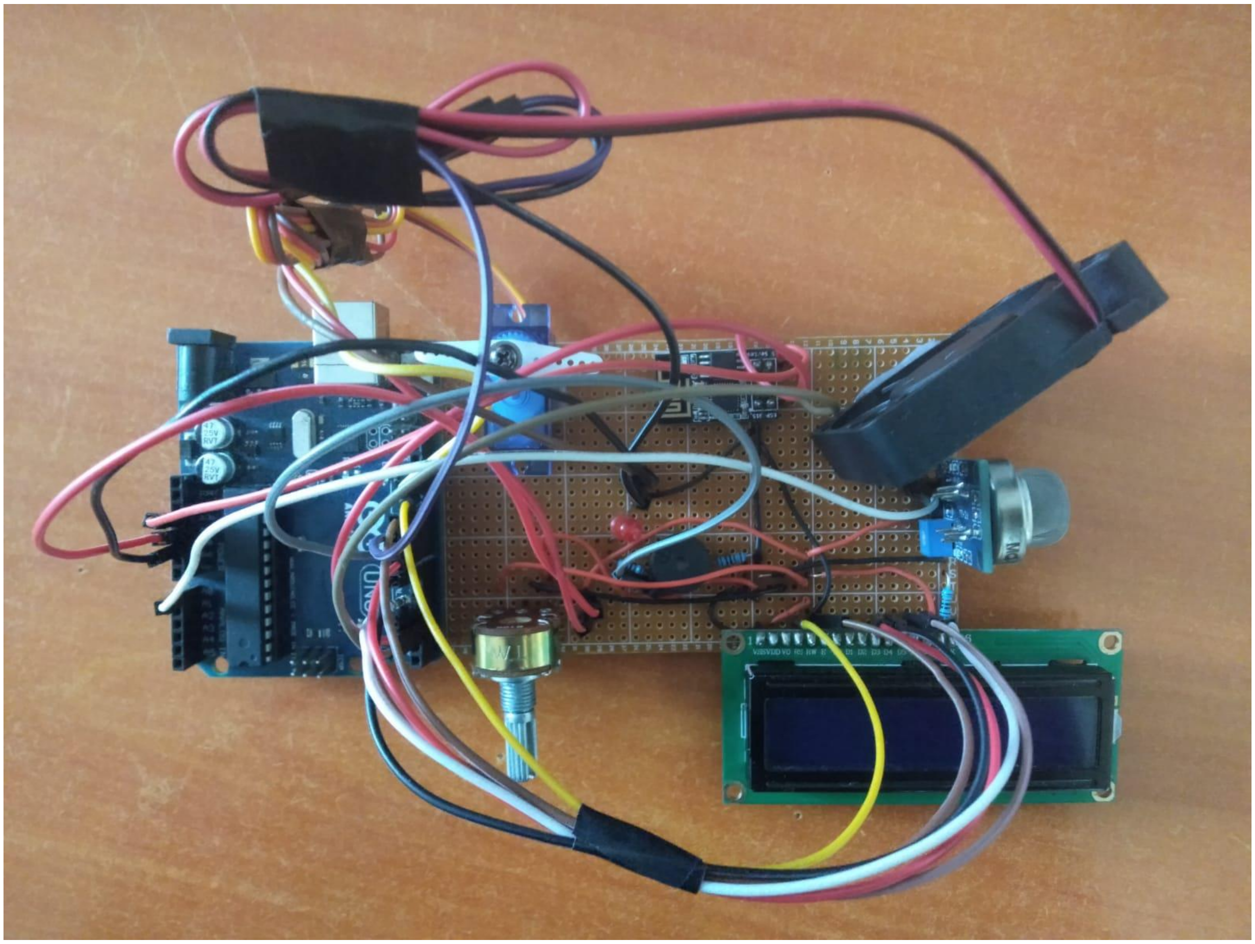


Figure 3: 6 Hardware Connections on Perforated Board

3.4 Design Process

To design the working of the gas leakage sensor, I categorized the levels of gas concentration into three:

- Less than or equal to 100ppm
- Greater than 101 and less than or equal to 300 ppm
- Greater than 300ppm

Once the Arduino is powered up using power cable, it powers on the peripherals that are connected to it. MQ-2 gas sensor measures the levels of gas in the surrounding. If the gas level is less than or equal to 100ppm, the value are uploaded to Thingspeak using Wi-Fi module and the values displayed on the LCD. In the intermediary gas level of between 101 – 300 ppm, data is also uploaded to the platform, LCD displays the gas concentration in PPM, red LED blinks to indicate the level is rising and the fan rotates to reduce the concentration. For levels greater than 300ppm, the code is designed to upload the values, display that the levels are extreme using the LCD, activate the fan to rotate to reduce concentration, light on the red LED and activate the servo motor to open an emergency opening to contain the high gas concentration levels.

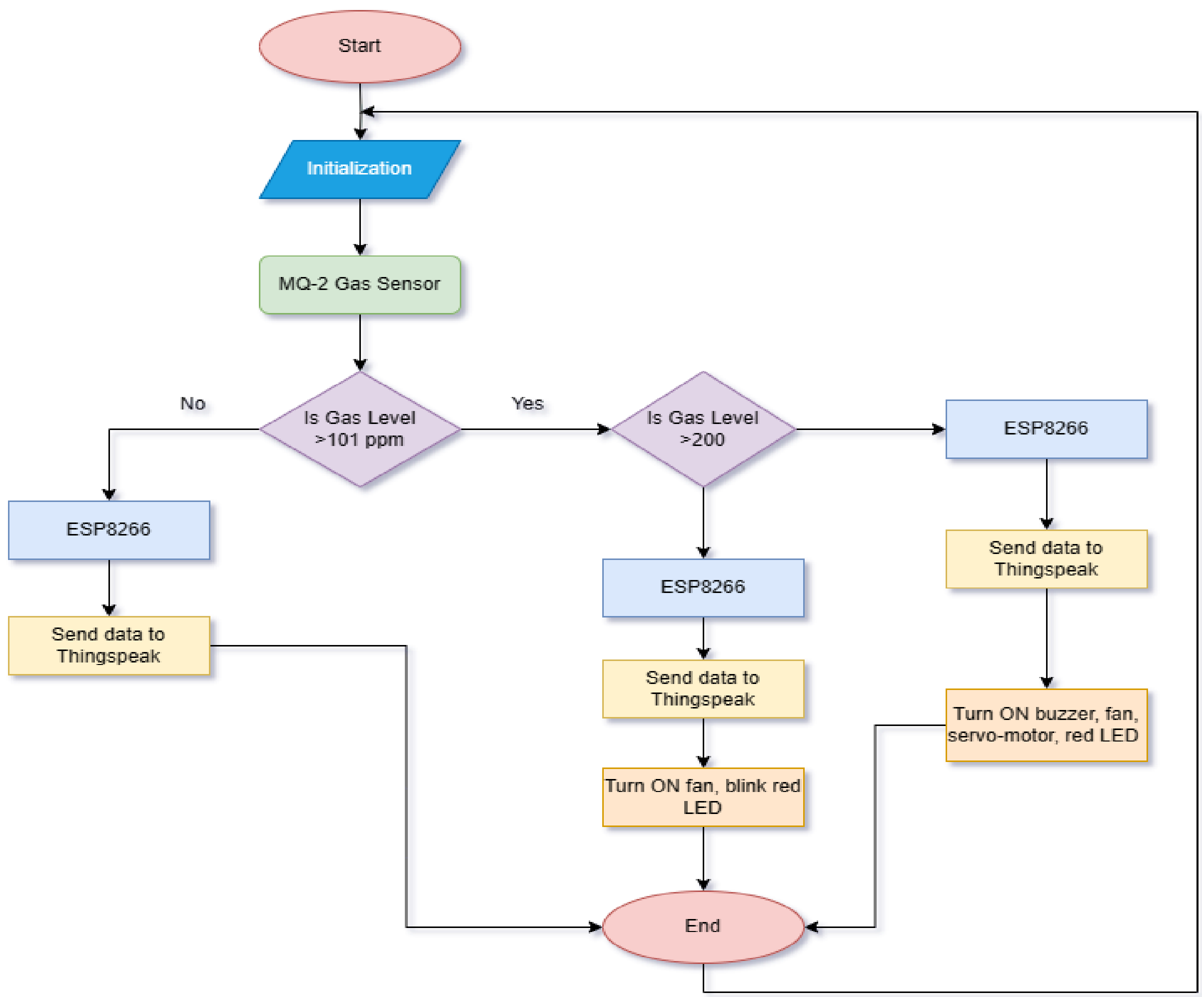


Figure 3: 7 Flowchart of Design Process

CHAPTER 4: RESULTS AND ANALYSIS

4.1 Introduction

After following through the specifications in the design and implementation chapter and using the code attached in appendix C, the results from the various levels of gas concentration are presented. This include results for normal levels, medium level and extreme concentration.

4.2 Normal gas level

On powering the design, the LCD first displays hello then displays the values. When there is no gas leakage, the LCD displays the levels are normal and the specific concentration of the gas. The buzzer connected did not make a tone. The servo motor and fan did not move and the red LED light was off.

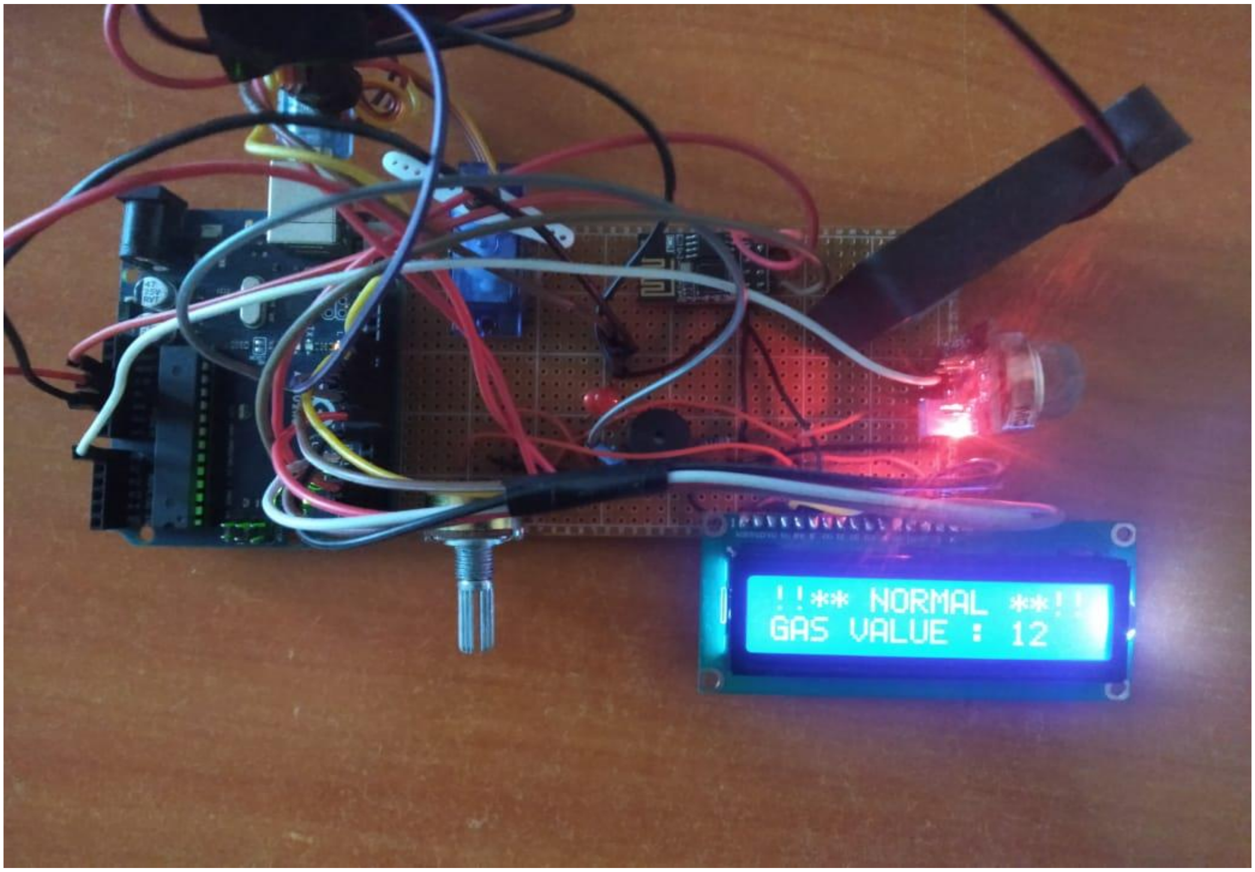


Figure 4: 1 LCD Display of Normal Gas Levels

The serial monitor of Arduino IDE also displays the gas levels and the commands for ESP8266 to send data to Thingspeak. If the AT command provided to Wi-Fi module is successful, an OK is displayed at the end of line and data is sent. Otherwise a fail is displayed and values not uploaded to the API.

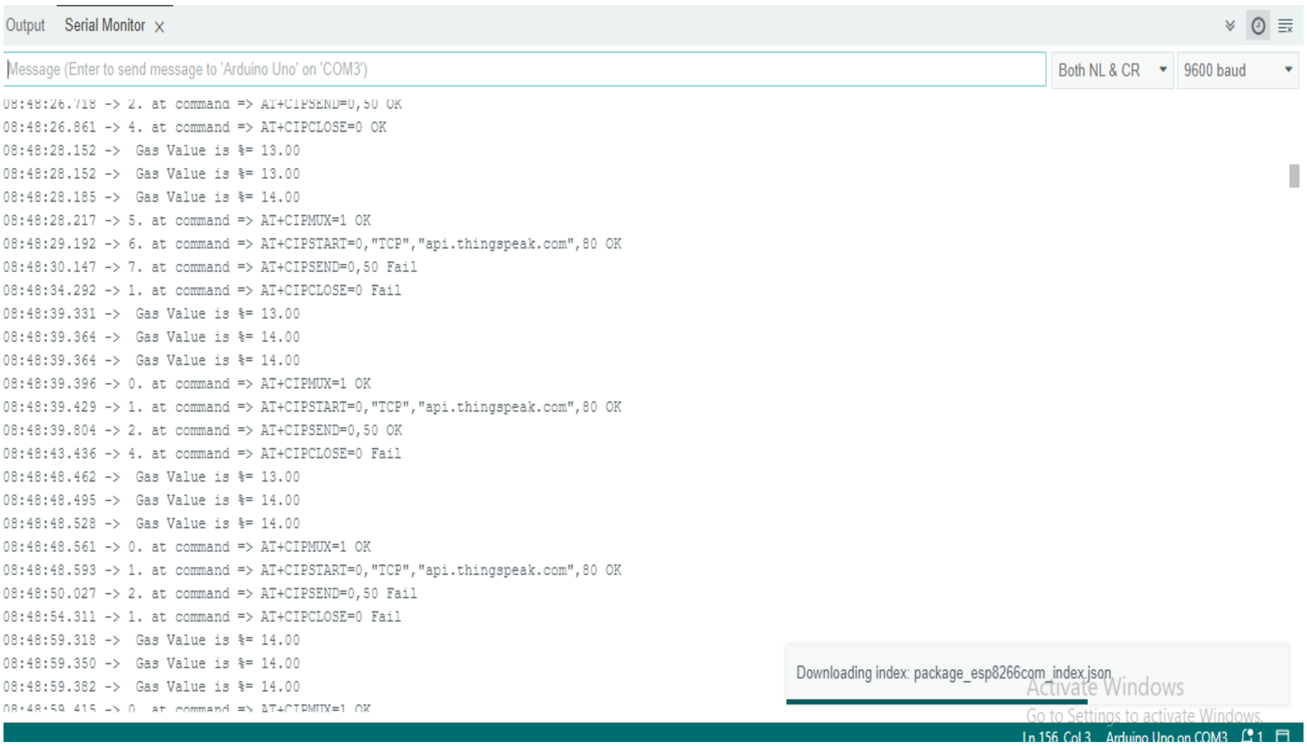


Figure 4: 2 Arduino IDE Serial Monitor Display

These values in the serial monitor are uploaded to the API and the levels can easily be monitored graphically from the software. Thingspeak displays the value of gas concentration in PPM against time in days. In figure 4.3 below, the value obtained from gas sensor and uploaded via ESP module at date 18th May 2023 Thursday 08:46AM is 12 PPM. The values are uploaded in threes and a small change in concentration is not easily noticed in the platform due to the scaling of gas concentration values in the y-axis and large differences in the concentration.

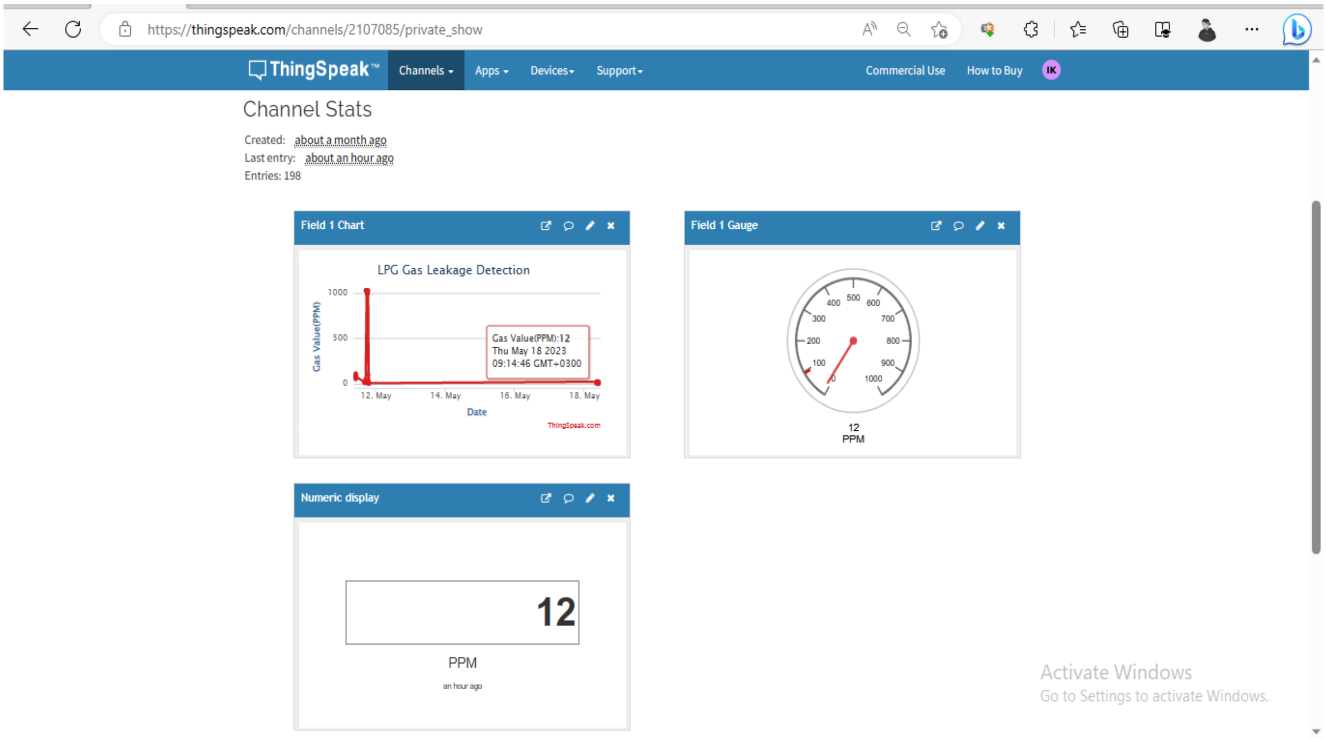


Figure 4: 3 Thingspeak Representation of Values in PPM

4.3 Presence of Gas Leakage

I used a lighter to test the presence of a gas leakage in the surrounding. When the lighter is brought close the gas sensor, the values of gas concentration in the serial monitor is seen to be rising and the values is uploaded to the platform after 15 seconds delay.

The fan was activated which reduced the concentration of the gas levels and the levels were seen to decrease. The buzzer went on for 5 seconds at a frequency of 400MHz and the red LED flashes to indicate leakage. The servo motor also rotated at an angle of 180^0 to open an emergency window or opening to also reduce the levels.

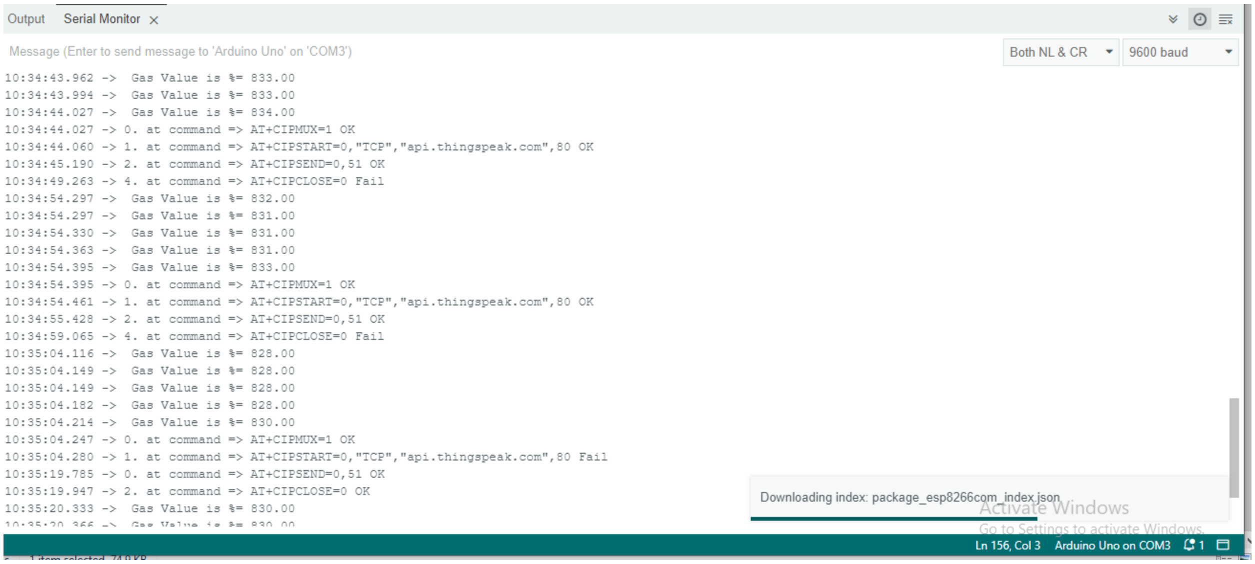


Figure 4: 4 Arduino IDE Serial Monitor Display Extreme Values

On receiving the gas level, the ESP8266 module sends data to Thingspeak immediately or at intervals of 10 seconds depending on the speed of internet connected to the module. The AT+CIPSEND, 51 OK command shows that the data has been sent and received successfully by the API.

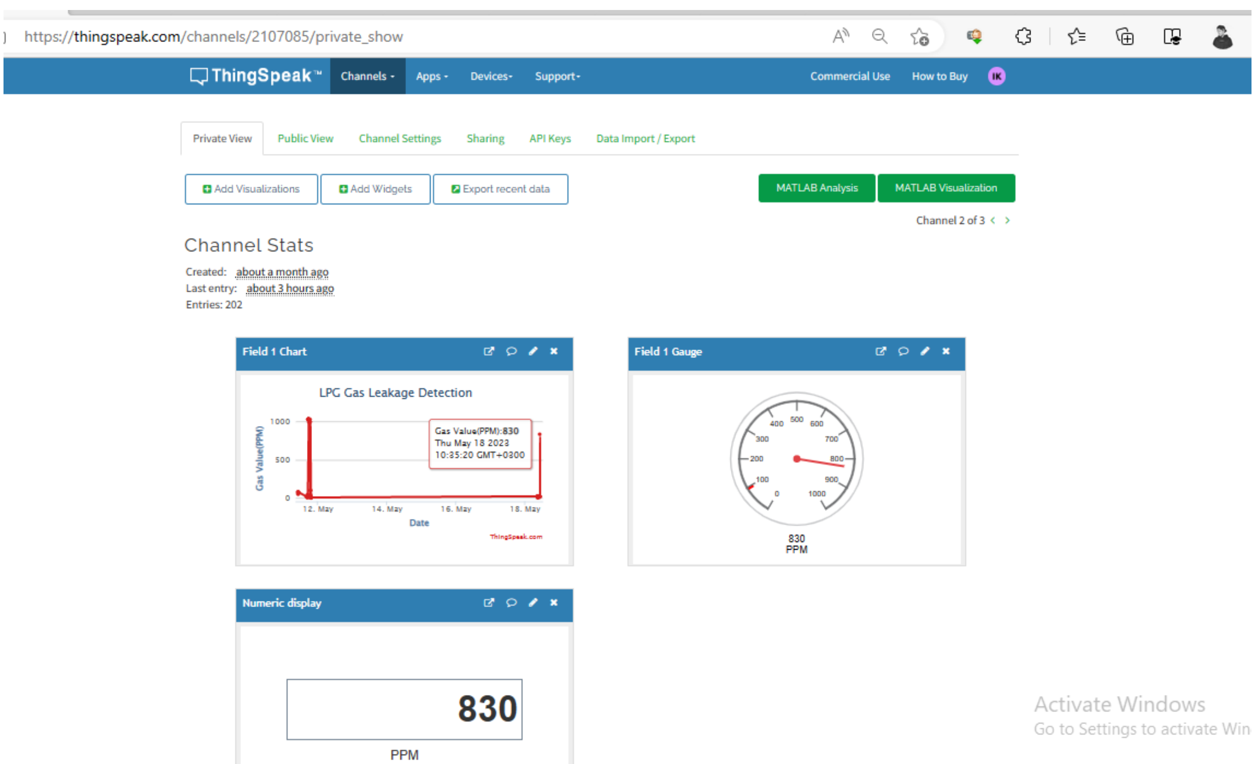


Figure 4: 5 Thingspeak Representation of Extreme Values in PPM

Figure 4.5 shows that that value obtained from the sensor during a gas leakage on Thursday 18th May 2023 at 10:35AM is 830 PPM. This level is greater than the set threshold value of 300 PPM thus activating control measures. The value can also be observed from field 1 gauge and the numeric display using Thingspeak as shown in Figure 4.5.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this project an MQ2 gas sensor was used to monitor the levels of LPG gas in an enclosure and appropriate intervention measures were set to be able to deal with an occurrence of gas leakage. The following were objectives were achieved:

- Detection of gas leaks (such as LPG, Butane or Methane) or any other petroleum-based gaseous material that an MQ2 sensor can detect.
  The sensor was able to detect data from the surrounding and send the values via an analogue signal to microprocessor, in this case Arduino.
- To display the data collected from the sensor over an online platform to visualize the gas levels and store the values for future references
  Thingspeak platform was used to display the data using graph, gauge levels and numeric display.
- To sound an alert whenever there is a gas leakage, and turn it off when the gas levels are normal.
  A buzzer was used as an alarm system to notify occupants of an emergency of gas leakage and be able to vacate for safety to avoid a catastrophe.
- To turn on the extractor fan when gas levels are high hence reducing the concentration in the surrounding.
  A 12V extractor fan was used hence reducing the high values of the flammable LPG gas from the premises as a protective measure.
- To display the specific levels of LPG gas in real time and if the value is normal, medium and extreme.
  This is achieved using a 16x2 LCD which is always on and shows the data obtained from the sensor via Arduino.
- To open an emergency window or opening in case of extreme LPG values.
  An SG90 servo motor is used as protective measure by rotating a window to open it if the leakage occurs in an enclosed room. The motor will maintain the position of the window for high gas values.

5.2 Shortcomings

Several challenges were encountered in the working of the project. They include:

- LCD display being unstable
- Delay in uploading of gas values to thingspeak in case of unstable internet connections

5.3 Recommendations

The following are suggestions on how the project can further be improved:

- Adding a GSM module to send SMS to occupants in case of gas leakage
- Modelling of house or an enclosing and connecting components i.e. servo motor a door or an emergency exit

References

[1] "Liquefied Petroleum Gas," Wikipedia, March 2023. [Online]. Available: https://en.wikipedia.org/wiki/Liquefied_petroleum_gas.

[2] C. News, "Shauri Moyo Declared Safe After Gas Leak," Capital FM, 28 September 2011. [Online]. Available: https://www.capitalfm.co.ke/news/2011/09/shauri-moyo-declared-safe-after-gas-leak/.

[3] L. Kamer, "Domestic consumption of Liquefied Petroleum Gas (LPG) in Kenya from 2016 to 2020," Statista, 1 August 2022. [Online]. Available: https://www.statista.com/statistics/1274010/domestic-consumption-of-lpg-in-kenya/.

[4] M. Waweru, "Kenya: Burn Incidents Exceeded Annual Average in 2020 - Data," Capital FM, 23 September 2021. [Online]. Available: https://allafrica.com/stories/202109230863.html.

[5] B. Authors, "Liquefied Petroleum Gas composition," Byju's, 2022. [Online]. Available: https://byjus.com/chemistry/lpg-composition/#:~:text=The%20primary%20components%20of%20liquefied,fuel%20in%20household%20cooking%20applications..

[6] AFROX, "Product Reference Manual - Section 5 - Liquefied Petroleum Ga," June 2020. [Online]. Available: http://www.awsgroup.co.za/data/L.P.G.pdf.

[7] M. T. Tombeng, "Prototype of Gas Leak Detector System using Microcontroller and SMS Gateway," *Cogito Smart Journal,* vol. 3, no. 1, pp. 3 - 7, 2017.

[8] T. H. B. V. D. K. M. S. S. A. D. &. D. L. P. Mujawar, "Development of wireless sensor network system for LPG gas leakage detection system," *International Journal of Scientific & Engineering Research,* vol. 6, no. 4, pp. 558-563, 2015.

[9] A. D. Authors, "Arduino Uno Documentation Datasheet," Arduino Inc, [Online]. Available: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf. [Accessed 22 March 2023].

[10] I. Grokhotkov, "ESP8266 Arduino Core Documentation," 3 April 2023. [Online]. Available: https://readthedocs.org/projects/arduino-esp8266/downloads/pdf/latest/. [Accessed 15 April 2023].

[11] A. Authors, "How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino," Last Minute Engineers, [Online]. Available: https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/. [Accessed 15 April 2023].

[12] M. Clary, "Interfacing to an LCD Screen Using an Arduino.," no. 1, pp. 1-9, 2015.

[13] A. K. K. A. A. S. Hitendra Rawat, "LPG Gas Leakage Detection & Control System," *National Conference on Synergetic Trends in engineering and Technology (STET-2014) International Journal of Engineering and Technical Research,* Vols. ISSN: 2321-0869, no. Special Issue, p. 5, 2014.

[14] L. M. Engineers, "How Servo Motor Works & Interface it With Arduino," Last Minute Engineers, [Online]. Available: https://lastminuteengineers.com/servo-motor-arduino-tutorial/#:~:text=Servos%20contain%20a%20small%20DC,motor%20to%20the%20target%20position.. [Accessed 12 April 2023].

[15] Elprocus, "Buzzer : Working, Types, Circuit, Advantages & Disadvantages," El-pro-cus, [Online]. Available: https://www.elprocus.com/buzzer-working-applications/. [Accessed 12 April 2023].

[16] A. Authors, "Arduino Software Download," Arduino, [Online]. Available: https://www.arduino.cc/en/software/. [Accessed 13 April 2023].

[17] E. H. Authors, "How to Connect ESP8266 to WiFi | A Beginner’s Guide," Electronics Hub, 30 March 2018. [Online]. Available: https://www.electronicshub.org/connect-esp8266-to-wifi/. [Accessed 2023].

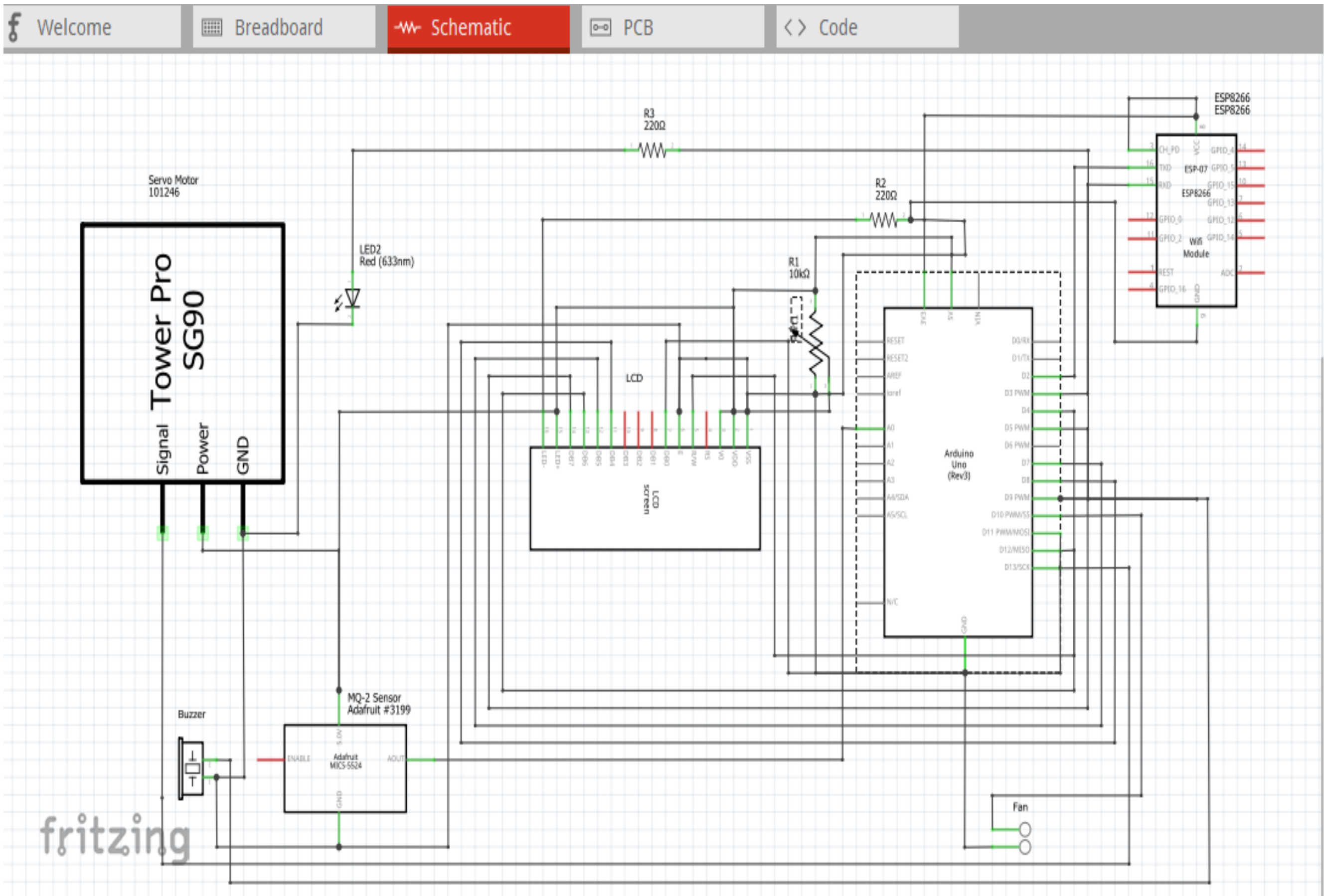
[18] K. Sharma, "CNG-LPG-gas-leakage-detecton-and-alert-using-IOT-thingspeak," Github, 27 April 2017. [Online]. Available: https://github.com/kavinsharma/CNG-LPG-gas-leakage-detecton-and-alert-using-IOT-thingspeak/blob/master/thingspeak_esp8266_mq5.ino. [Accessed 24 March 2023].

[19] K. Tachikawa, "ArduinoWifi," Github, 12 March 2019. [Online]. Available: https://github.com/Aeres-u99/ArduinoWifi/blob/master/FInaly_working_no_push.ino. [Accessed 24 March 2023].

Appendices

Appendix A

Hardware Schematic Diagram



Appendix B

Cost Analysis

Component	Quantity	Price(Ksh)
Arduino Uno	1	1000
ESP8266 Wi-Fi Module	1	300
Extractor Fan	1	80
16x2 LCD	1	500
Buzzer	1	100
Servo motor	1	300
MQ-2 gas sensor	1	400
Soldier Wire	3 meters	90
Breadboard	1	200
Jumper Wires	1 Packet	150

Appendix C

C++ Code [18] [19]

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <Servo.h>

LiquidCrystal lcd(12, 11, 8, 7, 4, 5); // Arduino pins connected to LCD. Another
option is using I2C
#define RX 2 // Arduino Pin 2 connected to TX Pin of ESP8266
#define TX 3 // Arduino Pin 3 connected to RX Pin of ESP8266
int servoPin = 13; // Arduino Pin servo_motor is connected to
Servo myServo;
int window_door = 0;
int pos = 0;
int piezoPin = 9; //Arduino Pin connected to Buzzer
int fanPin = 10; // Arduino Pin connected to mini-fan
String AP = "neek";       // Internet Username to connect to ESP
String wifi_password = "43218765"; // Internet password
String write_api = "UV0TOP5EXKABH1XF";   // Thingspeak Write API KEY
String host_api = "api.thingspeak.com"; //Thingspeak api
String PORT = "80"; // Connection to port 80
int trueCount;
int timeCount;
boolean found = false;
int valSensor = 1;
int const gas_pin = A0;

SoftwareSerial esp8266(RX,TX);

void setup() {
  lcd.begin(16, 2);
  lcd.print(" HELLO ");
  myServo.attach(servoPin);
  myServo.write(pos);
  pinMode(piezoPin, OUTPUT);
  pinMode(fanPin, OUTPUT);
  Serial.begin(9600);
  esp8266.begin(115200);
  sendingCommands("AT",5,"OK");
  sendingCommands("AT+CWMODE=1",5,"OK");
  sendingCommands("AT+CWJAP=\""+ AP +"\",\""+ wifi_password +"\"",20,"OK");
}
```

```
void loop() {
  if (getGasValue()<100){
    //low level of gas concentration in the air
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" NORMAL ");
    lcd.setCursor(0,1);
    lcd.print("Gas Value : ");
    lcd.print(getGasValue());
    digitalWrite(piezoPin, LOW);
    digitalWrite(fanPin, LOW);

    if (window_door == 1){
      for (pos = 180; pos >= 0; pos -= 1){
        myServo.write(pos);
        delay(15);
        }
        window_door = 0;
    }
    String getData = "GET /update?api_key="+ write_api +"&field1="+getGasValue();
    sendingCommands("AT+CIPMUX=1",5,"OK");
    sendingCommands("AT+CIPSTART=0,\"TCP\",\""+ host_api +"\","+ PORT,15,"OK");
    sendingCommands("AT+CIPSEND=0," +String(getData.length()+4),4,">");
    esp8266.println(getData);delay(150);trueCount++;
    sendingCommands("AT+CIPCLOSE=0",5,"OK");
  }
  //Moderate gas value will be updated to thingspeak
  else if(getGasValue()>101 && getGasValue()<300) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" MEDIUM ");
    lcd.setCursor(0,1);
    lcd.print("Gas Value : ");
    lcd.print(getGasValue());
    digitalWrite(piezoPin, HIGH);
    delay(10);
    digitalWrite(piezoPin, LOW);
    delay(10);
    digitalWrite(fanPin, HIGH);
    String getData = "GET /update?api_key="+ write_api +"&field1="+getGasValue();
    sendingCommands("AT+CIPMUX=1",5,"OK");
    sendingCommands("AT+CIPSTART=0,\"TCP\",\""+ host_api +"\","+ PORT,15,"OK");
    sendingCommands("AT+CIPSEND=0," +String(getData.length()+4),4,">");
    esp8266.println(getData);delay(150);trueCount++;
    sendingCommands("AT+CIPCLOSE=0",5,"OK");
```

```
  }
  else {
    // Extreme level of gas concentration in the air
    // Data obtained will be updated to Thingspeak
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("!!** EXTREME **!!");
    lcd.setCursor(0,1);
    lcd.print("Gas Value : ");
    lcd.print(getGasValue());
    tone(piezoPin, 4000, 5000);
    digitalWrite(piezoPin, HIGH);
    digitalWrite(fanPin, HIGH);
    if(window_door==0){
      for(pos = 0;pos <= 180; pos += 1){
        myServo.write(pos);
        delay(15);
      }
      window_door=1;
    }
    String getData = "GET /update?api_key="+ write_api +"&field1="+getGasValue();
    sendingCommands("AT+CIPMUX=1",5,"OK");
    sendingCommands("AT+CIPSTART=0,\"TCP\",\""+ host_api +"\","+ PORT,15,"OK");
    sendingCommands("AT+CIPSEND=0," +String(getData.length()+4),4,">");
    esp8266.println(getData);delay(150);trueCount++;
    sendingCommands("AT+CIPCLOSE=0",5,"OK");
  }
}


int getGasValue(){

   float g = analogRead(gas_pin);
   Serial.print(" Gas Value is %= ");
   Serial.println(g);
   delay(10);
   return int(g);

}

void sendingCommands(String command, int maxTime, char readReplay[]) {
  Serial.print(trueCount);
  Serial.print(". at command -> ");
  Serial.print(command);
  Serial.print(" ");
```

```
  while(timeCount < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
      command = true;
      break;
    }

    timeCount++;
  }

  if(found == true)
  {
    Serial.println("OK");
    trueCount++;
    timeCount = 0;
  }

  if(found == false)
  {
    Serial.println("FAIL UPLOAD");
    trueCount = 0;
    timeCount = 0;
  }

  found = false;
}
```