Summer Training Report

On

DESIGN & DEVELOPMENT OF AN IOT-BASED GAS LEAKAGE DETECTOR

At

Defense Research and Development Laboratory
Defense Research and Development Organisation
Kanchanbagh, Hyderabad -500058

Submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in
MECHATRONICS ENGINEERING

JABALPUR ENGINEERING COLLEGE , JABALPUR 482011

By

Aditya Patel (0201MT211004)
Pranay Shukla (0201MT211042)
Vaibhav Thakur (0201MT211059)

Under the guidance of
Mr Baubual Kumhar(Scientist 'F')

DECLARATION

We Aditya Patel (0201MT211004) Pranay Shukla (0201MT211042) Vaibhav Thakur (0201MT211059) hereby declares that the Project Report entitled " **IOT BASED GAS LEAKAGE DETECTOR SYSTEM**" done by us under the guidance of Mr.Babulal Kumhar is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Mechatronics Engineering.

1.Aditya Patel

2.Pranay Shukla

3.Vaibhav Thakur

Date: **Name & Signature Candidate**

Place:

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to the Defence Research and Development Laboratory (DRDL), a premier lab of the Defence Research and Development Organisation (DRDO), for allowing me to undertake this enriching internship. The exposure and experience gained here have been invaluable to my academic and professional growth.

First and foremost, I extend my heartfelt thanks to Mr. Babulal Kumhar, my project guide, for their continuous guidance, support, and encouragement throughout the internship. Their insightful feedback and unwavering patience were instrumental in completing this project.

Lastly, I am thankful to my academic institution, Jabalpur Engineering College], for their support and allowing me to undertake this internship.

This internship has been a significant milestone in my academic journey, and I deeply appreciate all the support and guidance received.

Thank you.

Aditya Patel (0201MT211004)
Pranay Shukla (0201MT211042)
Vaibhav Thakur (0201MT211059)

ORGANISATIONAL PROFILE

Defence Research & Development Organisation (DRDO) works under the Department of Defence Research and Development of the Ministry of Defence. DRDO dedicatedly worked towards enhancing self-reliance in Defence Systems and undertakes design & development leading to the production of world-class weapon systems and equipment following the expressed needs and the qualitative requirements laid down by the three services.

DRDO is working in various areas of military technology which include aeronautics, armaments, combat vehicles, electronics, instrumentation engineering systems, missiles, materials, naval systems, advanced computing, simulation and life sciences. DRDO while striving to meet the Cutting edge weapons technology requirements provides ample spinoff benefits to the society at large thereby contributing to nation building.

Defence Research and Development Laboratory (DRDL) is an Indian missile development laboratory, part of the Defence Research and Development Organization (DRDO). Their charter is centred on the design, development, and flight evaluation of various types of missile systems for the Indian armed forces.

VISION

"Be a design and development house for missile-based weapon systems required for tactical applications from multiple platforms."

MISSION

"Develop the state-of-the-art infrastructure and technologies required for different classes of missiles. Transfer the technology to a production agency for guided missile products."

ABSTRACT

Liquefied Petroleum Gas (LPG) is a main source of fuel, especially in urban areas because it is clean compared to firewood and charcoal. Gas leakage is a major problem in the industrial sector, residential premises, etc. Nowadays, home security has become a major issue because of increasing gas leakage. Gas leakage is a source of great anxiety with ateliers, residential areas, and vehicles like Compressed Natural Gas (CNG), buses, and cars which are run on gas power. One of the preventive methods to stop accidents associated with gas leakage is to install a gas leakage detection kit at vulnerable places. This paper aims to propose and discuss a design of a gas leakage detection system that can automatically detect, alert, and control gas leakage. This proposed system also includes an alerting system for the users. The system is based on a sensor such as MQ-2, MQ-3, MQ-4, MQ-5, and MQ-6 that easily detect a gas leakage.

CONTENT

CHAPTER 1

INTRODUCTION

Safety plays a crucial role in today's world as accidents are susceptible to happen anywhere. Places that make use of flammable and not easily detectable gases are susceptible to the occurrence of accidental fires. The Web of Things may be a futuristic technology during which the interconnection of devices and therefore the Internet is proposed. This often changes the automation of the many daily chores. Within the proposed gas detection system, we'll use IoT to detect leakage and alert the user preventing any longer gas leakage.

Toxic gases cause serious health impacts but also are utilised in industries in large quantities. These gases need to be monitored; such an increase within the normal level might be known and proper precautionary measures are often taken.

Arduino is going to be wont to perform the specified task by interfacing a gas sensor and LCD to display, an Ethernet shield to send alert messages to the user via an Android application, and a servo motor to turn on the window. The system will detect the gas leakage by using a gas sensor and it'll inform the Arduino board which can perform further actions i.e. opening the window or turning on the fan. The people within the neighbourhood also can be included just in case of an emergency. MQ6 LPG gas sensor is employed for input. The gas leakage event may involve danger at all times.

There are many deaths around the world due to gas leakage. Thus, it's ensured that one doesn't need to worry about the gas leakage becoming so intense and out of control that it can cause damage to life or the encompassing environment and also notifying and alerting the workers or residents about the gas leakage.

It gives a HIGH output when LPG, i-butane, propane, methane, alcohol, hydrogen, and smoke gas are sensed. This module is extremely easy to interface with microcontrollers and Arduino and simply available in the market by the name "LPG Gas Sensor Module".

Various kinds of anthropogenic emissions named primary pollutants are pumped into the atmosphere that undergo chemical reactions and further lead to the formation of new pollutants normally called secondary pollutants. For instance, according to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC), nearly all climate-altering pollutants either directly or indirectly (by contributing to secondary pollutants in the atmosphere) are responsible for health problems.

Almost every citizen spends 90% of their time in indoor air. Outdoor air quality of the cities of developed countries improved considerably in recent decades. In contrast to this, indoor

air quality degraded during this same period because of many factors like reduced ventilation, energy conservation, and the introduction to new sources and new materials that cause indoor pollution. The design of buildings for lower power consumption resulted in a decrease of ventilation which further decreases the quality of air inside the building. This increases the need for indoor air quality.

1.2 WORKING OF PROJECT

Creating a gas leakage detector using Arduino, an MQ-6 sensor, Blynk, and a NodeMCU involves several steps. Here's a step-by-step guide:

Components Required

1. **Arduino Uno** (or any other compatible board)
2. **MQ-6 Gas Sensor**
3. **NodeMCU (ESP8266)**
4. **Blynk App** (installed on your smartphone)
5. **Breadboard and jumper wires**
6. **Power supply** (for the Arduino and NodeMCU)
7. **Resistors** (as required for the sensor)

Circuit Diagram

1. **MQ-6 Sensor Connections**:
    - **Vcc** to **5V** on Arduino
    - **GND** to **GND** on Arduino
    - **Analog Out** to **Analog Input A0** on Arduino
2. **NodeMCU Connections**:
    - **Vin** to **5V** on Arduino
    - **GND** to **GND** on Arduino
    - **TX** to **RX** on Arduino
    - **RX** to **TX** on Arduino

Software Setup

1. Arduino IDE

- Install the Arduino IDE if you haven't already.
- Install the **ESP8266** board in the Arduino IDE.
- Go to **File** > **Preferences**.
- In the **Additional Board Manager URLs**, add this link: `http://arduino.esp8266.com/stable/package_esp8266com_index.json`

- Go to **Tools** > **Board** > **Boards Manager** and install **esp8266**.

2. Blynk Library

- Install the Blynk library from the Library Manager:
    - Go to **Sketch** > **Include Library** > **Manage Libraries** and search for **Blynk**.

3. Blynk App Setup

- Create a new project in the Blynk app.
- Select **ESP8266** as the device.
- Note the **auth token** that Blynk sends to your email.

Testing and Deployment

1. Upload the Arduino code to the Arduino board.
2. Upload the NodeMCU code to the NodeMCU.
3. Power both the Arduino and NodeMCU.
4. Open the Blynk app on your smartphone.
5. Monitor the gas levels on the Blynk app.

1.3 SCOPE OF THE PROJECT:-

A gas detector is a device that detects the presence of gases in an area, often as part of a safety system. A gas detector can sound an alarm to operators in the area where the leak is occurring, allowing them to leave. This type of device is important because many gases can be harmful to organic life, such as humans or animals.

Gas detectors can be used to detect combustible, flammable, and toxic gases and oxygen depletion. This type of device is used widely in industry and can be found in locations, such as on oil rigs, to monitor manufacturing processes and emerging technologies such as photovoltaic. They may be used in firefighting.

In this work whenever the gas is leaked it is sensed by the sensor and it sends a signal through the processor to the buzzer. The ppm can be seen in the LCD display whenever the gas leakage reaches the ppm of 100 it will send a signal to the buzzer and the buzzer will ring and it will alert the people around that the gas is leaking.

1.4 ADVANTAGES OF THE GAS DETECTION SYSTEM:-

You can monitor the amount of gases in your environment.
• Can escape from higher chances of poisoning, explosion, fire or asphyxiation.
• Get real-time -alerts about the gaseous presence in the atmosphere
• Prevent the fire hazards and explosion
• Supervise gas concentration levels
• Ensure worker's health
• Real-time update about leakage
• Cost-effective installation
• Data analytics for improved decision
• Measure oxygen level accuracy
• Get immediate gas leak alerts

1.5 APPLICATIONS OF USING GAS SENSOR:-

Harmful Gas Detection

The sensing of toxic gases such as H2S, Methane, and CO is of great importance in any industry to avoid unwanted leakage and consequences like poisoning or explosions. The presence of these gases can be easily detected in industrial facilities and commercial buildings with the help of IoT-powered gas monitoring solutions. Moreover, a gas detector or sensor device is a crucial part of carrying out safe industrial operations. The sensor-enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises.

Fire Hazard Prevention

The gas sensors help detect the concentration of the gases present in the atmosphere to avoid hazardous consequences like fire breakouts. Also, it is an imperative solution to keep the plant workers and equipment safe from fire hazards. It effectively detects the presence of hazardous gases like propane and methane and alerts the plant authorities, preventing the premises from unexpected ignition. Moreover, a gas monitoring solution uses gas analyzers to generate alerts regarding the temperature increase. This allows the management to take immediate action to curb harmful fire explosions.

Oxygen Level Measurement

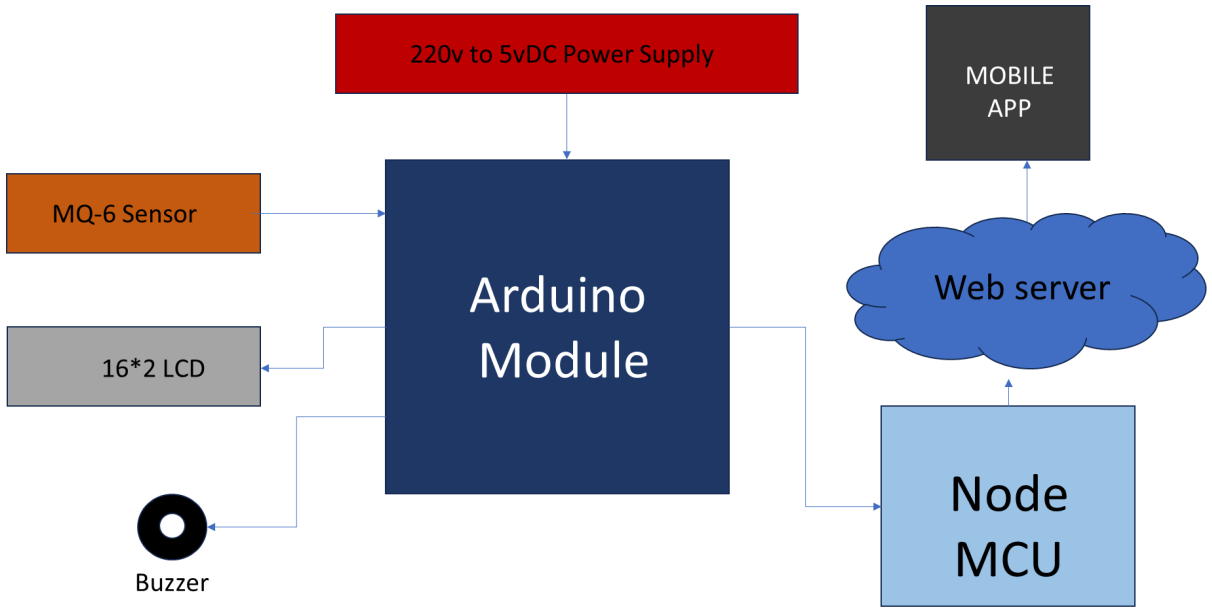
Sensing the presence of gases is a necessity to conduct industrial operations as several pitmen had lost their lives due to lack of oxygen in the process of mining explorations. A sudden decrease in oxygen levels can result in dizziness, brain damage, or even death among workers working in mines or close-packed industrial premises. A gas monitoring system significantly benefits industries by maintaining proper oxygen levels that reflect the optimal performance of your workers. This system also creates alerts in real-time about the decreasing oxygen levels, which

gives enough time to take necessary measures to evacuate the facilities much before the health gets affected.

Chapter 2

PROPOSED SYSTEM

2.1 BLOCK DIAGRAM :



2.2 COMPONENTS REQUIRES:

1. ARDUINO
2. MQ-6 GAS SENSOR
3. 16*2 LCD
4. BUZZER
5. 10 KV VARIABLE RESISTOR
6. MALE TO MALE/FEMALE WIRE
7. GAS LIGHTER
8. 9V BATTERY / CHARGER TO GIVE SUPPLY

2.3 MQ-6 Sensor

A gas sensor is a device that detects the presence or concentration of gas in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage. based on this voltage value the type and the concentration of the gas can be estimated.

MQ-6 Sensor detects gases like:- LPG, Alcohol, Propane, Hydrogen, Methane, Carbon, Nitrogen etc.

COMPONENTS OF SENSOR

- SnO_2 Layer
- Measuring Electrode
- Heater
- Plastic & Stainless Steel Net
- Micro Al_2O_3 ceramic tube

Specification of MQ sensor-

- Working Condition:

Circuit Voltage	-	5V ± 0.1 V (AC or DC)
Heating Voltage	-	5V ± 0.1 V (AC or DC)
Load Resistance	-	20 KΩ
Heater Resistance	-	33 Ω ± 5%
Heating Consumption	-	<750mW

- Environmental Condition:

Operation Temperature	-	-10°C to 50°C
Storage Temperature	-	20°C to 70°C
Relative humidity (RH)	-	Less than 95%
Oxygen Concentration	-	21%

- Sensitivity Characteristics

Sensing Resistance - 10 KΩ - 60 KΩ (1000 ppm LPG)
Concentration slope rate - ≤ 0.6 (1000 ppm / 4000 ppm LPG
Detecting concentration scope- 200-10000 ppm

2.4 MQ-6 Sensor Structure

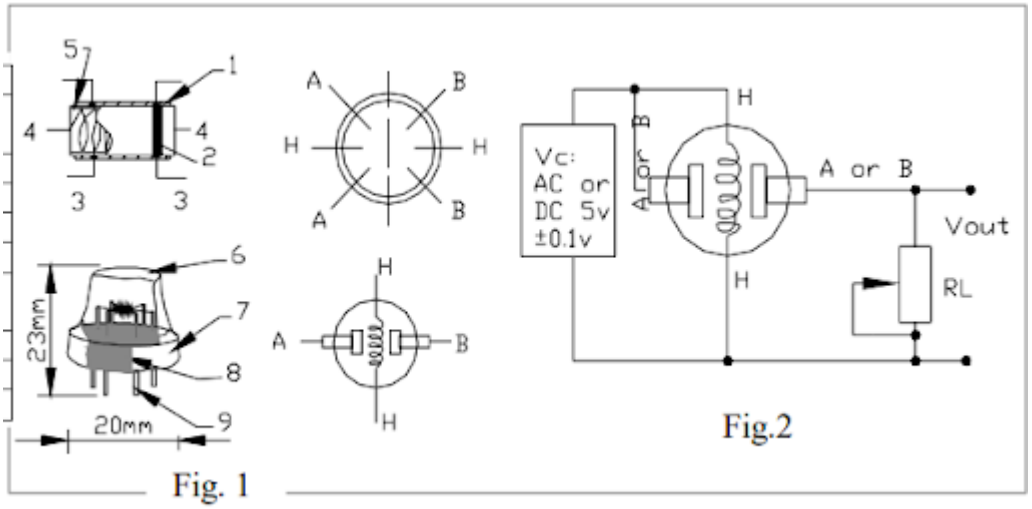


Fig. 1

Fig.2

The detail of each label and its description is written here in the table below:

Component	Material	Description
1	SnO2	Gas sensing layer - responsible for detecting target gases through changes in resistance.
2	Au	Electrodes - facilitate electrical contact with the gas sensing layer.
3	Pt	Electrode line - connects the electrodes to the external circuitry.
4	Ni-Cr alloy	Heater coil - provides heat to activate the gas sensing layer and increase sensitivity.

5	Al2O3	Tubular ceramic - housing and support for the gas sensor components.
6	Stainless steel gauze (SUS316 100-mesh)	Anti-explosion network - prevents flame propagation into the sensor body.
7	Copper plating Ni	Clamp ring - secures the sensor components within the tubular ceramic.
8	Bakelite	Resin base - provides mechanical support and electrical insulation for the sensor.
9	Copper plating Ni	Tube pin - electrical connection point for the sensor.

2.5 MQ-6 Sensor Pinout Configuration

Pin Name	Description
Vcc	This is the power Pin that requires an operating voltage of 5V.
GND	Ground pin connected to the ground terminal of the circuit
DO	It is a digital output pin that needs to set the threshold value using a Pot.
AO	Analog out the pin. It based the output of this pin on the intensity of the LPG or other gas.

2.6 MQ-6 Sensor Alternatives

Sr.No.	Name of sensor	Detects
1	MQ-2	METHANE, BUTANE,LPG, SMOKE
2	MQ-3	ALCOHOL, ETHANOL, SMOKE.
3	MQ-4	METHANE, CNG GAS.
4	MQ-5	NATURAL GAS,LPG
5	MQ-6	LPG,BUTANE GAS.

6	MQ-7	CARBON MONOXIDE
7	MQ-8	HYDROGEN GAS
8	MQ-9	CARBON MONOXIDE, FLAMMABLE GASES.
9	MQ-131	OZONE
10	MQ-135	AIR QUANTITY (BENZENE, ALCOHOL, SMOKE).

11	MQ-136	HYDROGEN SULPHIDE GAS
12	MQ-137	AMMONIA.
13	MQ-138	BENZENE,TOLUENE,ALCOHOL,ACETONE,PROPANE,FORMALDEHYDE GAS,HYDROGEN
14	MQ-241	METHANE,NATURAL GAS
15	MQ-216	NATURAL GAS, COAL GAS
16	MQ-303A	ALCOHOL,ETHANOL,SMOKE.
17	MQ-306A	LPG,BUTANE GAS
18	MQ-307A	CARBON MONOXIDE.
19	MQ-309A	CARBON MONOXIDE,FLAMMABLE GASES.
20	MQ-811	CARBON DIOXIDE(C02).

2.7 WORKING PRINCIPLE OF MQ-6 SENSOR

The working principle is based on - Heating Process
Ion Creation
Gas Absorption
Resistance Measurement
Digital Output

Heating Process: When the sensor is turned on, the circuit starts heating the core of this sensor that has the sensitive element SnO2 layer. This process is done to maintain a temperature of around 300°C (572°F) which activates the sensing element and it starts absorbing the oxygen from the surrounding air.

Ion Creation: The result of the reaction in the previous step creates the depletion region around the sensing element. As a result, the electrical conductivity of the circuit decreases because of the high resistance.

Gas Absorption: Once the MQ-6 LPG sensor comes into contact with the target gas, the oxygen ions from the depletion region start reacting with the gas molecules and as a result, the depletion region starts adsorbing. This causes the reduction in the number of oxygen ions and the overall conductivity increase.

Resistance Measurement: The circuit of the sensor measures the change in the resistance and the electrical current as well. The change in the current is directly proportional to the amount of the target gas in the environment. In this way, the analog values are sent to the output device through the analog output pin.

Digital Output: The MQ-6 provides the feature of digital output as well. The analog values, when exceeding the threshold value set through the potentiometer, are converted into the digital output and the sensor sends the signal through the digital pin. This is useful because usually, this pin is connected to the alarm and in the systems like an automatic alarm that shows the signal of the LPT butane gas presence.

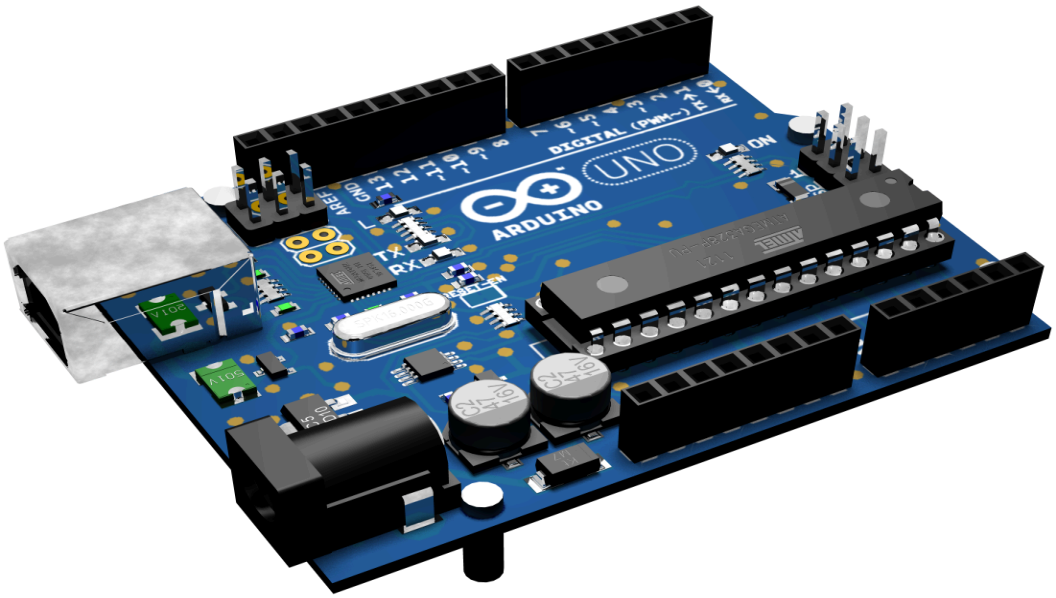
2.8 Applications Of MQ-6 Sensor

Here is a list of simple and basic examples that show the applications where the MQ-6 LPG butane gas sensor is extensively used:

- Home
- Industries
- Vehicles
- Indoor environments

- Industrial settings
- Landfill gas detection
- VOC monitoring
- Portable Gas Detection Devices
- Personal safety monitoring
- Industrial inspections
- Flammable gas sensing experiment

2.9 ARDUINO-UNO:-



The Arduino Uno is a popular open-source microcontroller board used for building electronics projects. It is based on the ATmega328P microcontroller and provides a simple, easy-to-use environment for programming and interfacing with hardware components. Here are some key features and details about the Arduino Uno:

2.9.1 Key Features:

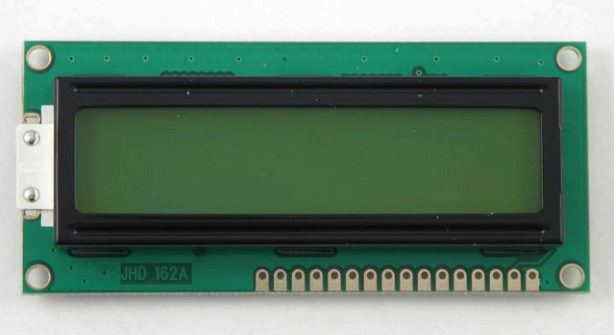
1. **Microcontroller**: ATmega328P
2. **Operating Voltage**: 5V
3. **Input Voltage (recommended)**: 7-12V
4. **Input Voltage (limits)**: 6-20V
5. **Digital I/O Pins**: 14 (of which 6 can provide PWM output)
6. **Analog Input Pins**: 6
7. **DC Current per I/O Pin**: 20 mA
8. **DC Current for 3.3V Pin**: 50 mA
9. **Flash Memory**: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
10. **SRAM**: 2 KB (ATmega328P)
11. **EEPROM**: 1 KB (ATmega328P)
12. **Clock Speed**: 16 MHz
13. **LED_BUILTIN**: 13

2.9.2 Pin Configuration:

1. **Power Pins**:
    - **Vin**: Input voltage to the Arduino board when using an external power source.
    - **5V**: Regulated power supply used to power microcontrollers and other components.
    - **3.3V**: A 3.3V supply is generated by the on-board regulator.
    - **GND**: Ground pins.
2. **Analog Pins**: A0 to A5, used to read analog voltage values.

3. **Digital Pins**: D0 to D13, used for both input and output. Pins D0 (RX) and D1 (TX) are used for serial communication.
4. **PWM Pins**: Pins with a tilde (~) symbol (D3, D5, D6, D9, D10, D11) can be used for PWM output.
5. **Special Function Pins**:
    - **Serial**: Pins D0 (RX) and D1 (TX).
    - **External Interrupts**: Pins D2 and D3.
    - **SPI**: Pins D10 (SS), D11 (MOSI), D12 (MISO), and D13 (SCK).
    - **I2C**: Pins A4 (SDA) and A5 (SCL).
    - **AREF**: Reference voltage for the analog inputs.

2.10 LCD DISPLAY:-



A 16x2 LCD is a very common type of liquid crystal display used in various electronics projects for showing text. It has 16 columns and 2 rows, allowing it to display up to 32 characters at a time. Here's a detailed overview of the 16x2 LCD and how to interface it with an Arduino Uno.

2.10.1 Key Features:

1. **Resolution**: 16 columns x 2 rows
2. **Interface**: Parallel (usually with a Hitachi HD44780 or compatible controller)
3. **Supply Voltage**: Typically 5V
4. **Backlight**: Some models come with a backlight for visibility in low-light conditions
5. **Character Set**: ASCII characters and some special symbols

2.10.2 Pin Configuration:

The typical pin configuration for a 16x2 LCD display is as follows:

1. **VSS (Pin 1)**: Ground
2. **VDD (Pin 2)**: +5V power supply
3. **VO (Pin 3)**: Contrast adjustment (connect to a potentiometer)
4. **RS (Pin 4)**: Register Select (0: command, 1: data)
5. **RW (Pin 5)**: Read/Write (0: write, 1: read)
6. **E (Pin 6)**: Enable
7. **D0-D7 (Pins 7-14)**: Data pins
8. **A (Pin 15)**: LED+ (Anode for backlight)
9. **K (Pin 16)**: LED- (Cathode for backlight)

2.10.3 Interfacing 16x2 LCD with Arduino Uno:

To interface the 16x2 LCD with an Arduino, you can use the LiquidCrystal library which simplifies the communication process. Here's a basic wiring setup and example code.

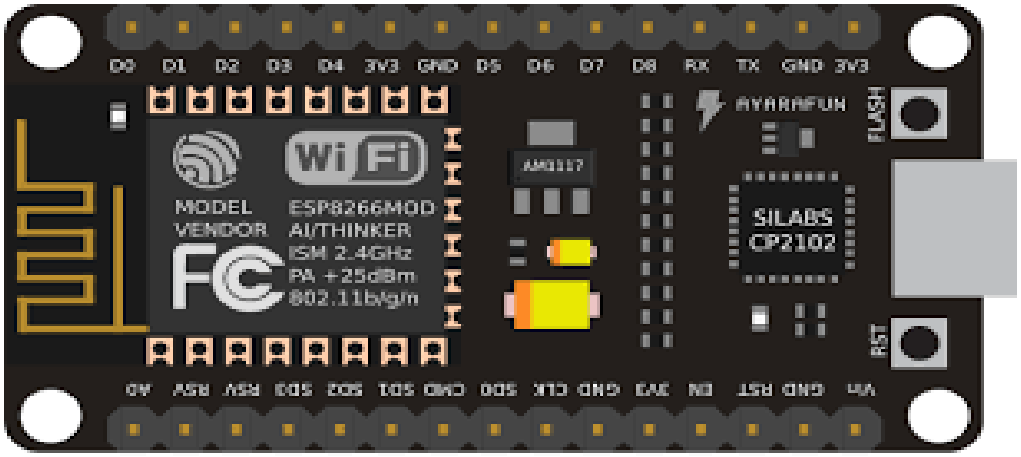
Wiring:

- **VSS** -> GND
- **VDD** -> 5V
- **VO** -> Center pin of a 10k potentiometer (other two pins to 5V and GND)
- **RS** -> Digital pin 12

- **RW** -> GND
- **E** -> Digital pin 11
- **D4** -> Digital pin 5
- **D5** -> Digital pin 4
- **D6** -> Digital pin 3
- **D7** -> Digital pin 2
- **A (Backlight Anode)** -> 5V (or through a current-limiting resistor to 5V)
- **K (Backlight Cathode)** -> GND

2.11 NodeMCU

An open-source IoT platform called NodeMCU is built on the ESP8266 chip. It has many GPIO pins, and built-in Wi-Fi, and can be programmed using the Lua programming language or the Arduino IDE. Because of its low cost, small size, and simplicity of use, it is extensively utilized for Internet of Things applications, including industrial automation, home automation, and environmental monitoring.



2.11.1 NodeMCU Specifications

- Microcontroller: ESP8266
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins: 16
- Analog Input Pins: 1 (10-bit ADC)
- Clock Speed: 80 MHz (up to 160 MHz)
- Flash Memory: 4 MB (ESP-12E module)
- RAM: 128 KB
- Wi-Fi: 802.11 b/g/n
- Interfaces: UART, SPI, I2C, PWM, GPIO
- USB: Micro USB for power and programming
- Dimensions: Approximately 48mm x 26mm

2.11.2 Characteristics of NodeMCU

- **Low Cost**: Affordable, making it accessible for hobbyists and professionals.
- **Wi-Fi Connectivity**: Integrated Wi-Fi for easy internet connection and IoT applications.
- **Power Efficient**: Can be used in low-power applications with deep sleep mode.
- **Compact Size**: Small form factor, suitable for embedded systems.
- **Open Source**: Both hardware and software are open source, with a large community for support.
- **Easy to Program**: Can be programmed using the Arduino IDE or Lua scripting language.

2.11.3 WORKING PRINCIPLE:

The working principle of NodeMCU revolves around the integration of the ESP8266 Wi-Fi module with additional hardware to facilitate the easy development of IoT applications. Here's a step-by-step explanation:

1. Power Supply and Initialization

- **Power Supply**: NodeMCU operates on 3.3V DC, which can be provided through the micro USB port or external power supply pins (VIN and GND).
- **Initialization**: Upon powering up, the ESP8266 microcontroller initializes and sets up the necessary hardware and software configurations.

2. Firmware and Programming

- **Firmware**: NodeMCU comes preloaded with NodeMCU firmware, which is based on the eLua project, enabling scripting with the Lua programming language. Alternatively, it can be programmed using the Arduino IDE with C/C++.
- **Programming**: The microcontroller is programmed via the micro USB connection using the Arduino IDE, Lua, or other supported environments. The code is uploaded to the onboard flash memory.

3. Wi-Fi Connectivity

- **Wi-Fi Initialization**: The ESP8266 initializes the Wi-Fi module, scans for available networks, and connects to a specified network using SSID and password.
- **Network Communication**: Once connected, the NodeMCU can send and receive data over the internet using standard TCP/IP protocols.

4. GPIO and Peripherals

- **GPIO Control**: The NodeMCU has multiple General Purpose Input/Output (GPIO) pins that can be programmed to read inputs from sensors or control outputs such as LEDs, motors, and other devices.

- **Peripheral Interfaces**: It supports interfaces like UART, SPI, I2C, PWM, and ADC, allowing it to interact with various peripheral devices and modules.

5. Execution of User Code

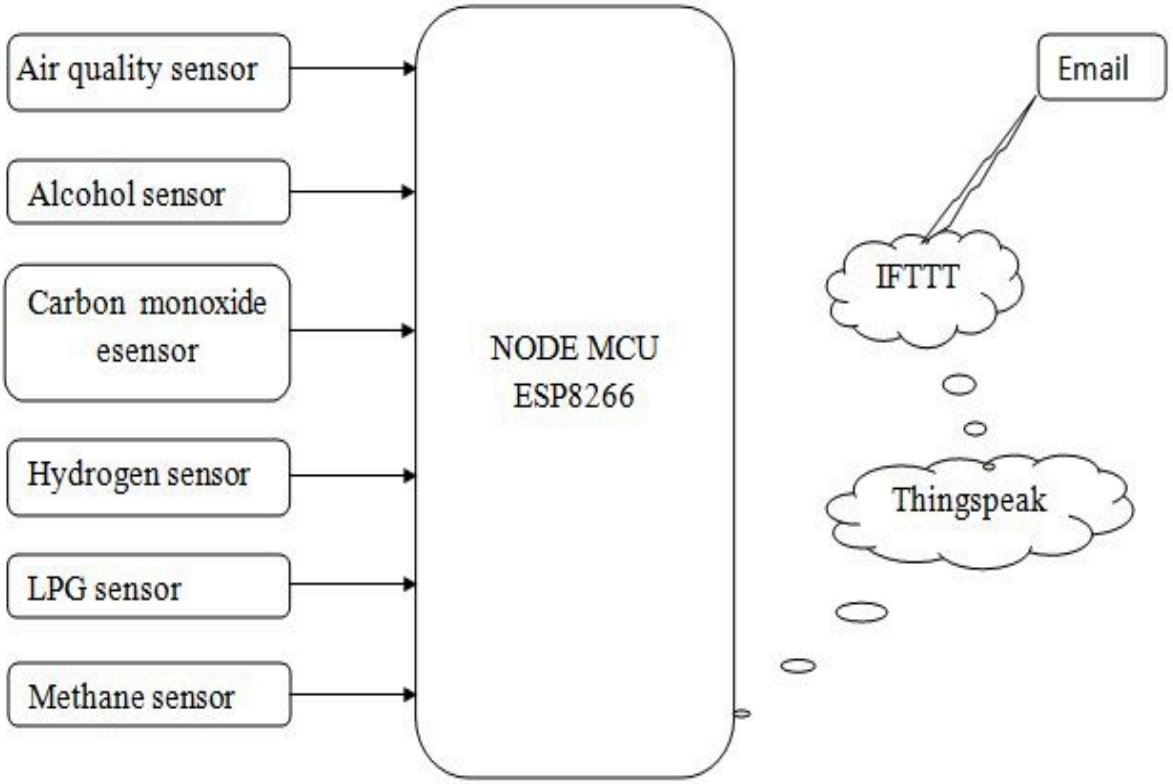
- **Loop Execution**: The main user code typically runs in a loop, continuously performing tasks such as reading sensor data, controlling actuators, and communicating with other devices or servers.
- **Event Handling**: NodeMCU can also handle events such as interrupts, which are triggered by external signals on GPIO pins.

6. Data Processing and Communication

- **Data Acquisition**: NodeMCU acquires data from connected sensors or devices through GPIO or communication interfaces.
- **Data Processing**: The acquired data is processed according to the programmed logic, which may include filtering, aggregating, or converting the data.
- **Data Transmission**: Processed data can be transmitted to remote servers, cloud platforms, or other devices over Wi-Fi. This enables real-time monitoring, control, and data logging.

7. Power Management

- **Power Modes**: NodeMCU supports various power modes, including deep sleep, which significantly reduces power consumption by turning off most of the microcontroller's functions except for a timer or external interrupt.
- **Wake-Up**: The microcontroller can wake up from deep sleep mode based on a timer or external signal to perform tasks and then return to sleep mode, optimizing battery life in portable applications.



Schematic Diagram Representation of Node-MCU Working

CHAPTER-3

APPENDIX SOURCE CODE

```
/*************************************************************

  This is a simple demo of sending and receiving some data.
  Be sure to check out other examples!
 *************************************************************/

/* Fill-in information from Blynk Device Info here */
#define BLYNK_TEMPLATE_ID           "TMPL3F2R_J3hJ"
#define BLYNK_TEMPLATE_NAME         "Quickstart Device"
#define BLYNK_AUTH_TOKEN            "ydZ-2zsVv9vMO1MwJRAvhtRRScMf4t0x"

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Airtel_Vinod Kumar Soni";
char pass[] = "chotiadvanced";

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();

  // Update state
  Blynk.virtualWrite(V1, value);
}

// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()
{
  // Change Web Link Button message to "Congratulations!"
  Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
  Blynk.setProperty(V3, "onImageUrl",  "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
  Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}

// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()
{
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V2, millis() / 1000);
}

void setup()
```

```
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  // You can also specify server:
  //Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, IPAddress(192,168,1,100), 8080);

  // Setup a function to be called every second
  timer.setInterval(1000L, myTimerEvent);
}

void loop()
{
  Blynk.run();
  timer.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```

This code is a simple demonstration of using an ESP8266 microcontroller with the Blynk platform to send and receive data. Here's a breakdown of the key components:

```
#include <ESP8266WiFi.h>
```

ESP8266WiFi.h: This library allows the ESP8266 to connect to a WiFi network.

```
#include <BlynkSimpleEsp8266.h>
```

BlynkSimpleEsp8266.h: This library facilitates communication between the ESP8266 and the Blynk platform.

```
#define BLYNK_TEMPLATE_ID           "TMPL3F2R_J3hJ"
#define BLYNK_TEMPLATE_NAME         "Quickstart Device"
#define BLYNK_AUTH_TOKEN            "ydZ-2zsVv9vMO1MwJRAvhtRRScMf4t0x"
```

Blynk Configuration:

These lines define the Blynk template ID, template name, and authorization token, which are necessary for connecting your device to the Blynk platform.

```
#define BLYNK_PRINT Serial
```

Debugging:
This line enables serial output for debugging purposes.

```
char ssid[] = "Airtel_Vinod Kumar Soni";
char pass[] = "chotiadvanced";
```

WiFi Credentials:

These lines define the SSID and password for your WiFi network.

```
BLYNK_WRITE(V0)
{
  int value = param.asInt();
  Blynk.virtualWrite(V1, value);
}
```

Handling Virtual Pin 0 Changes:

This function is called whenever the state of Virtual Pin 0 changes. It reads the new value from Virtual Pin 0 and writes it to Virtual Pin 1.

```
void myTimerEvent()
{
  Blynk.virtualWrite(V2, millis() / 1000);
}
```

Sending Uptime to Virtual Pin 2:

This function sends the microcontroller's uptime (in seconds) to Virtual Pin 2 every second.

```
void setup()
{
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(1000L, myTimerEvent);
}
```

Setup Function:

In the `setup` function, serial communication is initialized, the device connects to the Blynk platform using the provided credentials, and a timer is set up to call `myTimerEvent` every second.

```
void loop()
{
  Blynk.run();
  timer.run();
}
```

Loop Function:

The `loop` function runs continuously, ensuring that Blynk communication and timer events are handled properly.

This code sets up an ESP8266 microcontroller to communicate with the Blynk platform, handle changes on Virtual Pin 0, update properties on connection, and send the device's uptime to Virtual Pin 2 every second. It provides a simple but effective way to demonstrate the capabilities of Blynk in an IoT project.

CHAPTER-4

CONCLUSION

The design of a sensor-based automatic gas leakage detector with an alert and control system has been proposed and discussed in this paper. This is a low-cost, low power, lightweight, portable, safe, user-friendly, efficient, multi-featured and simple system device for detecting gas. Gas leakage detection will not only provide us with significance in the health department but it will also lead to a rise in our economy, because when gas leaks it not only contaminates the atmosphere but also wastage of gases will hurt our economy. The proposed system is easily affordable even for poor people. In the open literature it is noticed that much work has not been done for a smart gas detection system. In future, more advanced features will be integrated with this system which will provide users with more safety and relaxation. The proliferation of handheld devices has led to developments in the field of smart gas sensors, which has considerably widened their scope of application. The need for ensuring safety in workplaces is expected to be the key driving force for the market over the coming years.

FUTURE SCOPE:-

Overall, the software and hardware parts of the systems have been developed and tested by introducing a small amount of LPG near the gas sensor module. One of the notable future functions of this system is to add a sub-system where wastage of gas and the uses of

gas can be monitored using this system. The system is flexible as a greater number of sensors and relays can be added to it according to the whole LPG supply setup in those premises. The author is adding more software-based intelligent functions to this system. This is an automatic gas detection, control and alert system. In future, this system will have a feature where it can notify the emergency services if any accidents happen. A mobile app and web-based app for real time monitoring also will be added. In the user app for this system, many smart features will be added. The overall features will make the system more safe for the users. The system will be optimised for use in many places like the car, the home, industries and many other places. After designing the final prototype with smart multifunctional features, the system will be implemented in real life scenarios as a pilot project.