

GAS SENSOR BY USING ARDUINO UNO

Abstract

Gas leak detection is of paramount importance for ensuring safety in various industrial, commercial, and residential settings. This paper presents a detailed investigation into the development of an LPG gas sensor system using Arduino, an open-source electronics platform. The gas sensor system utilizes an MQ-series gas sensor to detect LPG gas concentrations and triggers safety measures when the gas concentration exceeds a predefined threshold. The project's design, implementation, and performance evaluation showcase its potential for cost-effective gas leak detection applications

Keywords: component; formatting; style; styling; insert (key words)

Authors

Durgesh Vijay Patil

Department Electrical Engineering
G.H Rasoni Institute of Engineering &
Business Management Jalgaon
durgeshvijay.patil.ee@ghribmjal.raisoni.net

Parikshit Sharadkumar Mali

Department Electrical Engineering
G.H Rasoni Institute of Engineering
&business Management Jalgaon
Parikshit.mali.ee@ghribmjal.raisoni.net

Om Kamlakar Chaudhari

Department Electrical Engineering
G.H Rasoni Institute of Engineering
&business Management Jalgaon
om.chaudhari.ee@ghribmjal.raisoni.net

Aniket Sharad Chaudhari

Department Electrical Engineering
G.H Rasoni Institute of Engineering
&business Management Jalgaon
aniket.chaudhari.ee@ghribmjal.raisoni.net

Lokesh Dayaneshwar Patil

Department Electrical Engineering
G.H Rasoni Institute of Engineering
&business Management Jalgaon
lokesh.patil.ee@ghribmjal.raisoni.net

Gunpesh Chandan Tayade

Department Electrical Engineering
G.H Rasoni Institute of Engineering
&business Management Jalgaon
gunpesh.tayade.ee@ghribmjal.raisoni.net

I. INTRODUCTION

Gas leaks, especially from Liquefied Petroleum Gas (LPG) sources, pose significant safety hazards, leading to fires, explosions, and health risks. The objective of this project is to design an LPG gas sensor using Arduino for real-time gas detection and safety applications. The affordability and simplicity of the Arduino platform make it an ideal choice for developing a cost-effective gas sensor system.

- 1. Literature Review:** The paper provides an overview of existing gas sensor technologies, highlighting the benefits and limitations of different sensor types. Previous research and projects related to gas leak detection using Arduino and MQ-series gas sensors are also reviewed to contextualize this study.
- 2. Methodology:** The gas sensor system is designed using an Arduino board (Arduino Uno) interfaced with an MQ-6 LPG gas sensor. The connections and circuit diagram between the gas sensor and Arduino are explained. The calibration process of the gas sensor is detailed to ensure accurate gas concentration readings.

II. EASE OF USE

- 1. Implementation:** The paper describes the step-by-step implementation of the gas sensor system using Arduino. The Arduino code for gas detection and threshold-based safety measures is provided, along with code snippets explaining the gas sensor's data acquisition and processing.
- 2. Results:** The gas sensor system is tested under controlled conditions to assess its performance in detecting varying concentrations of LPG gas. The obtained data is presented in graphical format, illustrating the sensor's response to different gas levels.
- 3. Discussion:** The results are analyzed, and the system's performance in detecting LPG gas leaks is discussed. The advantages and limitations of the gas sensor system are highlighted, along with suggestions for possible improvements.

III. PREPARE YOUR PAPER BEFORE STYLING

- 1. Abbreviations and Acronyms:** Here are some common abbreviations and acronyms related to gas sensors used with Arduino:

- LPG - Liquefied Petroleum Gas
- MQ - Gas sensor series produced by the company "Winsen Electronics," commonly Used with Arduino (e.g., MQ-2, MQ-6).
- CO - Carbon Monoxide
- CO₂ - Carbon Dioxide
- H₂S - Hydrogen Sulfide
- VOC - Volatile Organic Compounds
- PID - Photoionization Detector
- PPM - Parts Per Million
- ADC - Analog-to-Digital Converter

- I2C - Inter-Integrated Circuit (a communication protocol used for sensor interfacing with Arduino)
- UART - Universal Asynchronous Receiver/Transmitter (a communication protocol used for sensor interfacing with Arduino)
- GPIO - General Purpose Input/Output
- LED - Light Emitting Diode
- PCB - Printed Circuit Board
- IC - Integrated Circuit
- UART - Universal Asynchronous Receiver/Transmitter
- LCD - Liquid Crystal Display
- SD - Secure Digital (for SD card data logging)
- IoT - Internet of Things
- PWM - Pulse Width Modulation (for controlling actuators based on gas sensor readings)

These abbreviations and acronyms are commonly used in the context of gas sensors and Arduino-based projects for gas detection and monitoring applications. They provide a convenient way to refer to specific components, gases, or technologies involved in the development and implementation of gas sensor systems using Arduino.

2. Units: To measure gas levels using Arduino, you can use various types of gas sensors, such as MQ-series sensors. These sensors come in different models, like MQ-2, MQ-4, MQ-5, etc., each designed to detect specific gases like LPG, methane, propane, and so on. To interface them with Arduino, you usually need to connect the sensor's analog output to an analog pin on the Arduino board. By reading the analog values, you can determine the gas concentration. Remember to check the datasheet of the specific gas sensor for detailed wiring and calibration instructions.

3. Equations:

- Connect the gas sensor to the Arduino board according to the sensor's datasheet. Typically, this involves connecting the analog output of the sensor to an analog pin on the Arduino.
- Set up the Arduino environment and open a new sketch.
- Define the analog pin that is connected to the gas sensor using the `analogRead()` function to read the analog value from the pin.
- Convert the analog reading to a voltage value, if necessary, depending on the sensor's output range.
- Calibrate the sensor by taking readings in clean air and in the presence of a known concentration of the gas you want to detect. Adjust your code to map the sensor readings to gas concentrations.
- Print or display the gas concentration using the Serial Monitor or any other output method you prefer.

Here's a basic code outline for reading an analog gas sensor on Arduino:

```
Arduino const int gasSensorPin = A0; // Replace A0 with the analog pin connected to the gas sensor
```

```
void setup() {Serial.begin(9600); // Initialize Serial Monitor for output}
void loop() { int sensorValue = analogRead(gasSensorPin); // Read analog value from the
gas sensor
float voltage = (sensorValue * 5.0) / 1023.0; // Convert analog value to voltage
(assuming 5V Arduino)
// Calibrate your sensor and convert the voltage to gas concentration using a calibration
curve
// Print the gas concentration to the Serial Monitor Serial.print("Gas Concentration: ");
Serial.print(gasConcentration); // Replace gasConcentration with your calculated gas
concentration value
Serial.println(" ppm"); // Units for gas concentration (e.g., ppm)
delay(1000); // Add a delay if needed between readings }
```

Please note that the calibration process and gas concentration conversion depend on the specific gas sensor you are using. Always refer to the sensor datasheet and manufacturer's guidelines for accurate calibration and gas concentration calculation.

IV. WORKING PRINCIPLE

The working principle of a gas sensor using Arduino typically involves an analog gas sensor and an Arduino board. The gas sensor, such as MQ-series sensors, operates based on the change in electrical conductivity when exposed to a specific gas.

Here's a general explanation of how the gas sensor works with Arduino:

- 1. Gas Detection:** Gas sensors are designed to detect the presence and concentration of specific gases. When the target gas comes in contact with the sensor's sensitive material, it undergoes a chemical reaction that alters the electrical conductivity of the sensor.
- 2. Sensor Response:** The change in electrical conductivity is proportional to the concentration of the gas. The higher the gas concentration, the greater the change in conductivity.
- 3. Analog Output:** The gas sensor provides an analog output voltage that varies depending on the gas concentration. This analog output voltage is usually in the range of 0 to 5 volts.
- 4. Arduino Interface:** The analog output of the gas sensor is connected to one of the analog input pins of the Arduino board.
- 5. Analog-to-Digital Conversion:** Arduino, being a digital device, needs to convert the analog voltage from the sensor to a digital value for processing. It does this using the `analogRead()` function.
- 6. Calibration and Conversion:** To determine the gas concentration accurately, the gas sensor needs to be calibrated. This calibration involves taking readings in clean air and in the presence of known concentrations of the gas. By mapping the sensor readings to gas

concentrations using a calibration curve, you can convert the analog values to meaningful gas concentration values.

- 7. Output:** Once the gas concentration is determined, you can display the result on the Serial Monitor, LCD screen, or any other output device you prefer.
- 8. Continuous Monitoring:** The Arduino can continuously monitor the gas concentration and provide real-time data.

It's important to note that different gas sensors may have slightly different working principles and calibration procedures. Always refer to the datasheet and guidelines provided by the manufacturer to properly use and calibrate your specific gas sensor with Arduino.

V. APPLICATIONS:

Gas sensors interfaced with Arduino have various applications across different industries and environments. Some common applications include:

- 1. Air Quality Monitoring:** Gas sensors can be used to monitor air quality in indoor and outdoor environments, detecting pollutants like carbon monoxide (CO), nitrogen dioxide (NO₂), sulfur dioxide (SO₂), and volatile organic compounds (VOCs). This is helpful in homes, offices, factories, and urban areas to ensure healthy air quality.
- 2. Industrial Safety:** Gas sensors are used in industrial settings to monitor the presence of toxic or combustible gases, helping to ensure worker safety. Common gases monitored include methane (CH₄), hydrogen sulfide (H₂S), ammonia (NH₃), and others.
- 3. Fire Detection:** Gas sensors can be part of fire detection systems, detecting smoke and hazardous gases to trigger alarms and evacuation protocols in case of fire.
- 4. Gas Leak Detection:** In residential and commercial buildings, gas sensors can detect gas leaks from natural gas pipelines or LPG cylinders, alerting occupants to potential dangers.
- 5. Environmental Monitoring:** Gas sensors are used to assess pollution levels in the environment, helping in environmental research and compliance monitoring.
- 6. Smart Agriculture:** In greenhouse applications, gas sensors can monitor the concentration of gases like carbon dioxide (CO₂) for optimized plant growth.
- 7. Vehicle Emission Monitoring:** Gas sensors can be integrated into vehicles to monitor exhaust emissions, contributing to emission control and compliance.
- 8. Mining Safety:** Gas sensors are utilized in mines to monitor the presence of harmful gases and prevent potential accidents.
- 9. Gas Detection in Labs :** In laboratories handling hazardous chemicals, gas sensors help to ensure a safe working environment

The versatility of gas sensors combined with the flexibility and programmability of Arduino allows for creative and diverse applications in numerous industries. When using gas sensors, it's essential to ensure proper calibration, maintenance, and adherence to safety guidelines to achieve accurate and reliable results.

VI. ADVANTAGES AND LIMITATIONS

1. Advantages of Gas Sensors using Arduino

- **Cost-Effective:** Gas sensors interfaced with Arduino are relatively affordable, making them accessible for various applications.
- **Real-time Monitoring:** Arduino allows gas sensors to provide real-time data, enabling immediate response to changes in gas concentrations.
- **Customization:** Arduino's flexibility allows users to customize gas sensor applications according to specific needs.
- **Ease of Use:** Arduino's user-friendly interface and extensive community support make gas sensor integration straightforward, even for beginners.
- **Portable and Compact:** The compact size of Arduino boards and gas sensors allows for portable and space-efficient monitoring solutions.
- **Versatile:** Arduino supports a wide range of gas sensors, accommodating diverse gas detection requirements.

2. Limitations of Gas Sensors using Arduino

- **Accuracy and Calibration:** Gas sensors might require careful calibration to ensure accurate readings, which can be a time-consuming process.
- **Limited Precision:** Some gas sensors may have limited precision in detecting low concentrations of gases accurately.
- **Cross-Sensitivity:** Certain gas sensors may exhibit cross-sensitivity, reacting to multiple gases, affecting specificity in detecting a single gas.
- **Environmental Influences:** Environmental conditions like humidity and temperature might influence gas sensor performance and readings.
- **Lifespan:** Gas sensors have a finite lifespan and might require replacement after prolonged use.
- **Complexity for Some Gases:** Detecting specific gases may require specialized sensors that can be more complex and expensive.

VII. FUTURE DEVELOPMENT

The future development of gas sensors using Arduino is likely to see several exciting advancements and improvements, driven by technological advancements and user needs. Some potential future developments include:

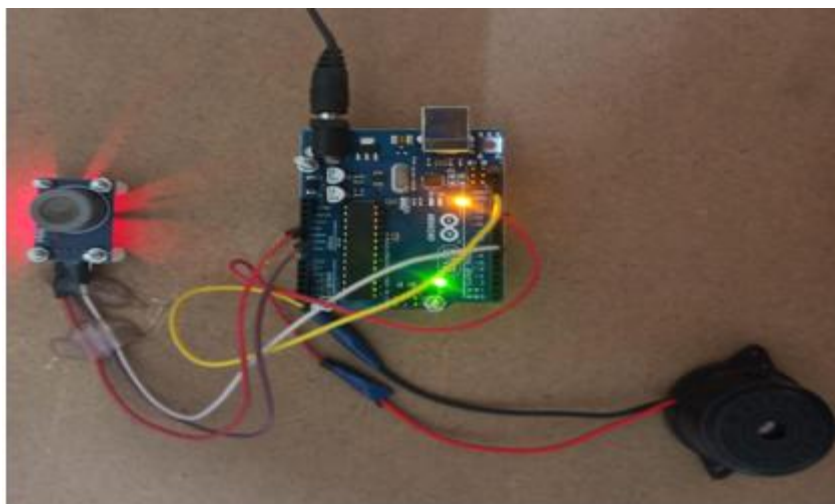
- **Higher Sensitivity and Selectivity:** Gas sensors are likely to become more sensitive and selective, allowing them to detect even lower concentrations of gases accurately and distinguish between different gases with higher precision.

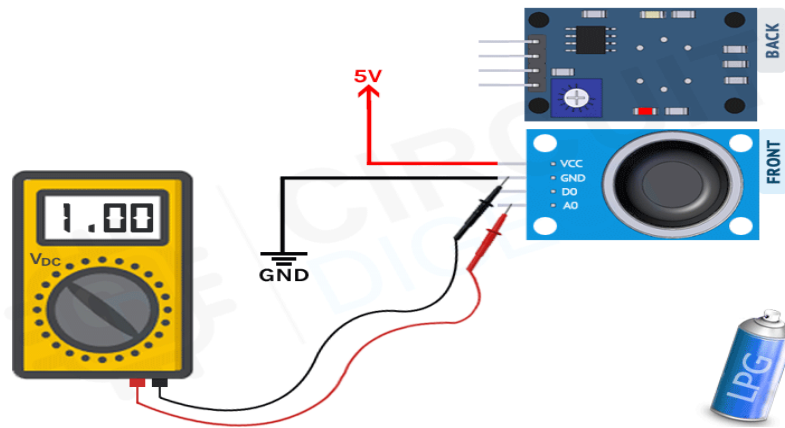
- **Miniaturization:** Continued miniaturization of gas sensors will lead to even smaller and more compact devices, enabling integration into various IoT and wearable applications.
- **Energy Efficiency:** Future gas sensors are expected to be more energy-efficient, extending the battery life of portable devices and reducing power consumption in general

1. Program:

```
int RedLED = 11;  
int GreenLED = 10;  
int Buzzer = 9;  
int MQ5 = A5;  
int SENSORTHRESHOLD = 120;  
void setup() { pinMode(RedLED, OUTPUT);  
pinMode(GreenLED, OUTPUT);  
pinMode(Buzzer, OUTPUT);  
pinMode(MQ5, INPUT);  
Serial.begin(9600);}  
void loop() { int analogSensor = analogRead(MQ5);  
Serial.print("Pin A0: ");  
Serial.println(analogSensor);  
if  
(analogSensor > SENSORTHRESHOLD) { digitalWrite(RedLED, HIGH);  
digitalWrite(GreenLED, LOW);  
digitalWrite(Buzzer, HIGH);}  
else { digitalWrite(RedLED, LOW);  
digitalWrite(GreenLED, HIGH);  
digitalWrite(Buzzer, LOW); }  
delay(100);}
```

2. Figures and Tables





VIII. TABLE

Sr no	Name of Componantes	Quantity
1	Arduino UNO Board	1
2	Gas sensor	1
3	Buzzer	1
4	Jumper wires	-

IX. CONCLUSION

The conclusion of using a gas sensor with Arduino depends on the specific project and the results obtained. Generally, gas sensors interfaced with Arduino can effectively detect and measure the concentration of certain gases in the environment. By analyzing the sensor readings, you can monitor air quality, detect gas leaks, or implement safety measures.

However, the effectiveness and accuracy of the gas sensor's performance will depend on factors like sensor quality, calibration, environmental conditions, and the gases being detected. It is essential to thoroughly test and calibrate the sensor to ensure reliable and precise results.

In conclusion, using a gas sensor with Arduino can be a valuable tool for various applications, but its performance and reliability rely on proper calibration and understanding the limitations of the specific sensor used.

REFERENCES

- [1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529-551, April 1955. (references)
- [2] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
- [3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [4] K. Elissa, "Title of paper if known," unpublished.
- [5] R. Nicole, "Title of paper with only first word capitalized," *J. Name Stand. Abbrev.*, in press.