# Smart Kitchen Air Quality Monitoring and Controlling System

April 11th, 2025

Divya Sivakumaran
Adam Pietrewicz

# Contents

# 1. Overview

## 1.1 Introduction

This report outlines the development and evaluation of the Smart Kitchen Air Quality Monitoring and Controlling System: a system that monitors air quality, and controls the ventilation when hazardous gas levels are detected. The primary use case would be for general appliances, especially household kitchens.

## 1.2 Motivation

According to the World Health Organization, approximately 3.2 million people face premature death annually due to household air pollution caused by cooking [1]. Although this is more common in developing countries, it still remains a problem in North America, as it can introduce lung diseases. After cooking the particulate matter can increase almost 65 times, and it can enter the bloodstream and lungs [2]. With the right ventilation in the kitchen, the concentration of harmful gases can be reduced which can decrease the risks of smoke inhalation. This project aims to address this issue by creating an IoT system that can monitor the air quality in the kitchen and control the ventilation system to increase airflow and remove hazardous gases. This real-time monitoring can decrease the smoke inhalation risks and allow for a safer cooking environment.

## 1.3 Project Description

The proposed project is a smart kitchen air quality system that a user can install in their home kitchen to monitor the air quality status wirelessly through a web application. The system comprises portable air quality sensor modules that can be placed around the desired room (even installed on an oven air vent) which connect to a cloud server and upload their air quality readings through an edge device. This data will be analyzed and displayed to the user on a computer application where they can read their air quality status from each sensor module. In addition, the system includes fans that are placed within the kitchen and automatically controlled by the system depending on the detection of harmful gases or smoke. For the scope of this project, a prototype was developed to show the feasibility of the product.

### 1.3.1 Functional Requirements:

1. Monitor and detect the air quality metrics such as gas levels of carbon monoxide, carbon dioxide, methane, volatile organic compounds (VOCs), Liquefied Petroleum Gas using air quality sensors
2. Send air quality readings to cloud using edge device
3. Access historical readings and measurements of air sensors from the cloud
4. Operate fan ventilation automatically based on detected gas levels and manually through the user application
5. Receive notifications through email or mobile text when there are threatening gas levels

### 1.3.2 Non-Functional Requirements:

1. Securely store sensor data on cloud to prevent unauthorized access
2. User application must be easy to operate and interact with

# 2. System Design

## 2.1 Block Diagram

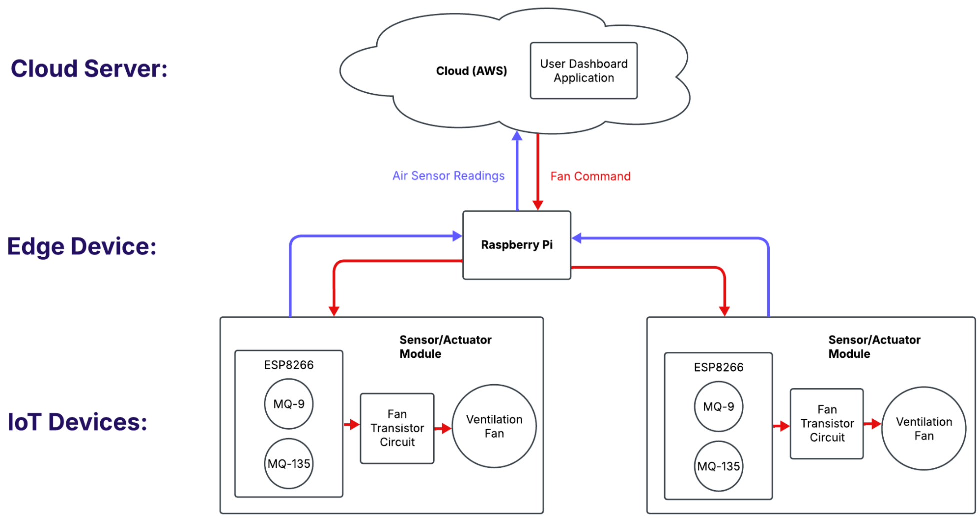


*Figure 1: Smart Kitchen Air System Block Diagram*

## 2.2 System Components

| **Component** | **Description** |
|---|---|
| MQ-9 Gas Sensor | This sensor outputs the carbon monoxide (CO), methane ($CH_4$), LPG (Liquified Petroleum Gas) PPM through an analog pin and notifies of dangerous gas levels using a digital pin as a boolean |
| MQ-135 Air Quality Sensor | This sensor outputs if dangerous levels toxic gases (carbon dioxide, ammonia, benzene, smoke) is detected using a digital pin |
| Ventilation Fan | 5V mini fan that is operated ON/OFF by the ESP8266 through the Fan Transistor Circuit |
| Fan Transistor Circuit:<br>- 2N2222 NPN Transistor<br>- Diode | Circuit that uses a transistor to switch between powering the fan ON/OFF by using a ESP digital pin. Diode is added for safe amperage/voltage levels to the fan |
| ESP8266 Board | Microcontroller that controls the readings of the air sensors and operation of the ventilation fan. Full setup of the module can be viewed in Appendix A. |
| Raspberry Pi | Edge device that receives air sensors readings from ESP and fan commands from AWS using MQTT |
| Cloud Server/User Application: AWS | Stores/visualizes air sensor readings, sends notifications to user and sends fan signal to turn ON when dangerous gas level detected |

*Table 1: Description of System Block Diagram Components*

## 2.3 Workflow

1. The ESP8266 board reads the gas PPM values from the MQ9 analog pin, and read the digital pin of the MQ135 to check if no dangerous gases are detected every second
2. Published the sensor readings from the ESP to the Raspberry Pi broker using MQTT
3. Raspberry Pi aggregates the data received by taking the average of the last 10 gas readings and sends this to the AWS Cloud
4. AWS IoT routes each field of the message to Cloudwatch Metrics
5. AWS IoT reroutes Air Quality field back to a “fan signal” topic which is used to control the fan
6. Raspberry PI is subscribed to this fan signal and will publish this message to the ESP via MQTT
7. ESP will turn the fan ON/OFF depending on the received MQTT fan signal (1 or 0)

## 2.4 Implementation

### 2.4.1 Sensor Calibration

For the MQ9 sensor, calibration was needed to convert the values from the analog pin into ppm. First the sensor needed to be calibrated in a clean air environment, and then that value was used for an exponential regression equation that was provided by the MQ Unified Sensors example code [3]. Additionally, MQ9 and MQ135 need to be heated up for around 20 seconds, in order to provide stable measurements.

### 2.4.2 Fan Circuit

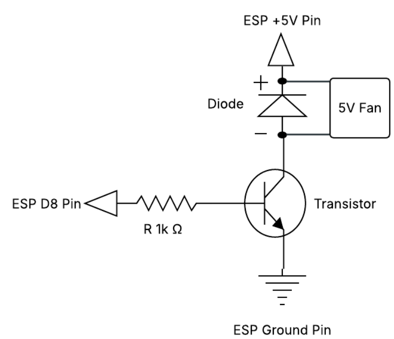


*Figure 2: Transistor Circuit to power fan ON/OFF by writing HIGH/LOW to ESP D8 Pin*

We constructed this circuit (photo available in Appendix A) in order to control the fan ON/OFF by writing HIGH/LOW to the ESP D8 Pin connected to the transistors base pin. When the transistor switches it connects the fan to the 5V ESP source. In addition a diode is added in between the fan voltage pins in order to safely make sure it is powered properly and no current will flow in the opposite direction, which would destroy the fan.

### 2.4.3 Cloud Architecture

To enhance user experience, functionalities were added with AWS Cloud that allows increased connectivity with real-time updates. To allow immediate action, when a message is published from the Raspberry Pi, there are a set of routing rules for the incoming topic. One set of this rule will create Cloudwatch Metrics for each of the fields (Air Quality, LPG, CH4, CO, and Gas Detected). The Cloudwatch Dashboard will visualize these metrics and allow adjustable periods (can be viewed in Appendix B). Additionally, an alarming threshold was set for each gas threshold and when it was breached an action was set. For this project, there were alarms set for the ppm of each gas, and if the max value exceeds the threshold, a few actions will be taken. First an SMS message will be sent to the user with information on the breaching alarm and levels. Second, an email will be sent with more information on the breach. These notifications are sent with Simple Notification Service (SNS), and it will notify the user in real-time (examples of notifications are available in Appendix C).

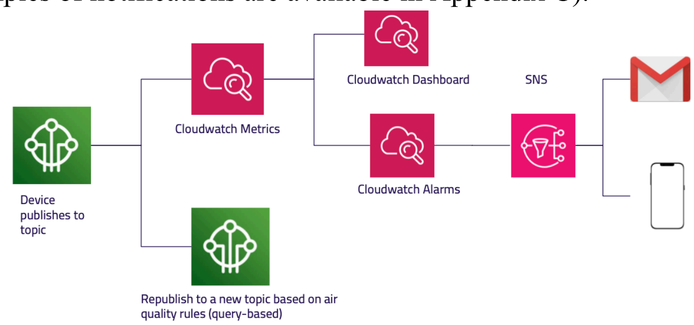


*Figure 3: AWS Cloud Architecture*

# 3. Evaluations

## 3.1 Testing

The prototype was tested in a controlled indoor environment to evaluate its responsiveness, accuracy, and system latency without exposure to real harmful gases using the following methods:

1. To calibrate the gas sensors, they were powered on for at least 5 minutes before running the system
2. The MQ9 and MQ135 sensors contain an on-board potentiometer that adjusts the threshold for which qualifies a gas reading as dangerous (which triggers a digital pin HIGH). Thus to simulate gas presence, the onboard potentiometers were manually adjusted, allowing the digital output to trigger threshold breaches
    a. The ESP8266 sent gas readings every second to the Raspberry Pi via MQTT. After collecting 10 readings, the Pi calculated the average and sent the aggregated result to AWS. Fan actuation occurred based on AWS messages received via MQTT. Thus the ventilation fan was actuated every 10 seconds due to little to no latency of MQTT data publishings.
3. Published manual data entries in AWS IoT Core MQTT queue to trigger Cloud actions such as Cloudwatch Alarms (SMS and email)
    a. On average the SMS and email were delivered within 30 seconds of breaching the threshold

## 3.2 Current Limitations

1. With the current setup, only the MQ9 ppm values are read. This is because the ESP8266 board only has one analog input pin. MQ9 was chosen to use this pin as it measures carbon monoxide which is a more common and harmful gas present when cooking. In order to improve and widen the range of gasses detected, using an Arduino board with 2 or more analog pins (such as the modern Arduino UNO) can allow better use of both sensors.
2. The current system sends the fan control signal from the AWS cloud. However if there is a Wifi connection issue, this can result in unsent messages, which therefore makes the system vulnerable to harmful gases during those time periods and the ventilation system will not be activated. For future development, the fans can be controlled locally by the microcontroller to mitigate this. However, setting the fan signal from the cloud layer allows the user to change the gas thresholds easily and exhibit more control. The cloud layer in this project also provides the data visualization and wireless notification.

3. Power constraint: the initial design was to power the sensor/actuator IoT module using a battery in order to make the module portable for easy installation in the kitchen. However, the 9V battery pack used lacked the voltage and current capacity to simultaneously power the ESP, fan, and sensors. A higher-capacity LiPo battery or power-boosting circuit is recommended for a truly portable and robust solution.

# 4. Related Publications

This project builds upon previous research in smart air quality monitoring but prioritizes affordability, ease of installation, and real-time user interaction over complex ML algorithms or expensive hardware.

1. **IoT and Machine Learning in Smart Kitchen Monitoring for Enhanced Worker Health [4]**

This publication explores using sensors to identify poor air quality in commercial kitchens. It uses similar hardware equipment for monitoring and controlling the air quality with gas sensors and fans. However there are differences in how the fan signals and user interface is designed. For this approach, the input sensor data is passed as an input into a classification machine learning model that labels the danger level. This level is then used to control the fan speeds. Additionally, this approach focuses on storing the data in the cloud for further analysis and new data to train the model. The advantage of this approach is that it has a model that combines all readings rather than having multiple thresholds for each sensor like the solution proposed in this paper. Furthermore, this solution only focuses on the immediate action but does not cover the user interaction with the data and will not notify users when an emergency occurs.

2. **Cooking emission control with IoT sensors and connected air quality interventions for smart and healthy homes: Evaluation of effectiveness and energy consumption [5]**

This publication solves a similar issue of improving the air quality while cooking. However, their implementation is far more advanced than a simple prototype, which results in better performance and direct real-life implementation. One advantage of this approach was the use of PM2.5 sensors which measure the concentration of particulate matter that is less than 2.5 micrometers, as these particles are more likely to enter the bloodstream and lungs. However, this sensor is quite expensive, priced at roughly $35 compared to the $4 MQ9 sensor used in our prototype. Additionally, this project uses a wide range of actuators used to clean the air such as a stove hood, portable air cleaners, and bathroom exhausts. These options allow a quicker air purification process, and can drastically reduce the particulate matter. However, it comes with more expensive equipment which may restrict the accessibility for lower income households and integrate it in developing countries. Their experimentation excels in performance and automation, however it lacks the monitoring and user interface that our project prioritizes.

# 5. Conclusion

This project addresses a common yet overlooked problem in everyday life. Cooking is a regular activity in every household that requires attention to maintaining a safe and healthy environment. With an easily installable and distributed system, users can be more aware of their home kitchen air quality and take steps to monitor and prioritize their health. With precautionary actions, the fan can increase ventilation to promote cleaner air and healthier breathing environments. This project advances the field by demonstrating how low-cost, open-source software/hardware can be integrated with commercial cloud services (AWS) to build a reliable, real-time air quality monitoring and ventilation system, bridging the gap between low-complexity sensor systems and existing high end machine learning based solutions.

# References


[1] World Health Organization, "Household air pollution and health," Oct. 16, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/household-air-pollution-and-health. [Accessed: Feb. 14, 2025].

[2] H. Canada, “Government of Canada,” Canada.ca, https://www.canada.ca/en/health-canada/services/publications/healthy-living/factsheet-cooking-and-indoor-air-quality.html [Accessed Apr. 11, 2025].

[3] miguel5612, “Unified library for MQ sensors”, https://github.com/miguel5612/MQSensorsLib [Accessed Apr. 11, 2025].

[4] D. K. Shah et al., “Smart kitchen: Real time monitoring of kitchen through IOT,” 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), pp. 718–722, Apr. 2022. doi:10.1109/iciem54221.2022.9853161

[5] J. Pantelic, Y. J. Son, B. Staven, and Q. Liu, “Cooking emission control with IOT sensors and connected air quality interventions for smart and healthy homes: Evaluation of effectiveness and energy consumption,” Energy and Buildings, vol. 286, p. 112932, May 2023. doi:10.1016/j.enbuild.2023.112932

# Appendix A: Sensor/Actuator Module

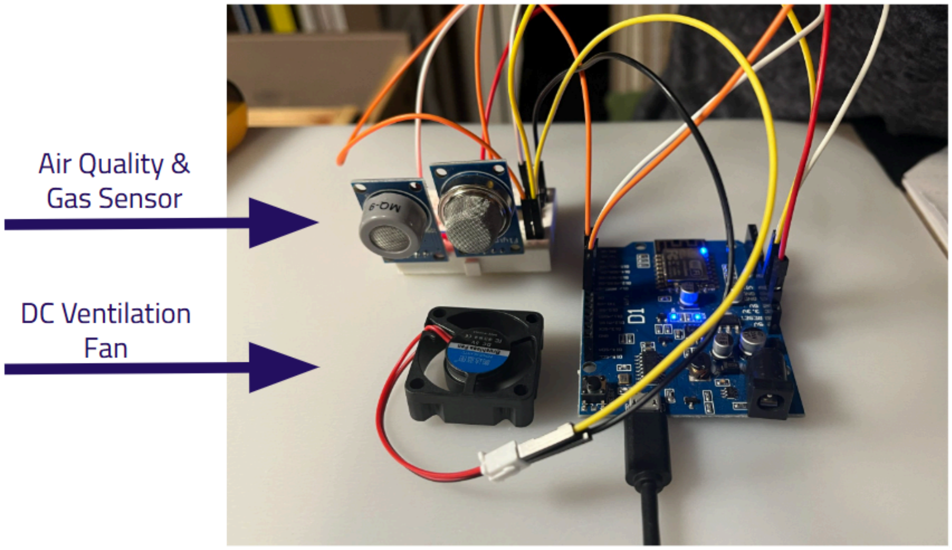


**Figure A1:** Photo of sensor/actuator module which includes: MQ9 and MQ135 Sensors, ESP8266 board and 5V DC ventilation fan.

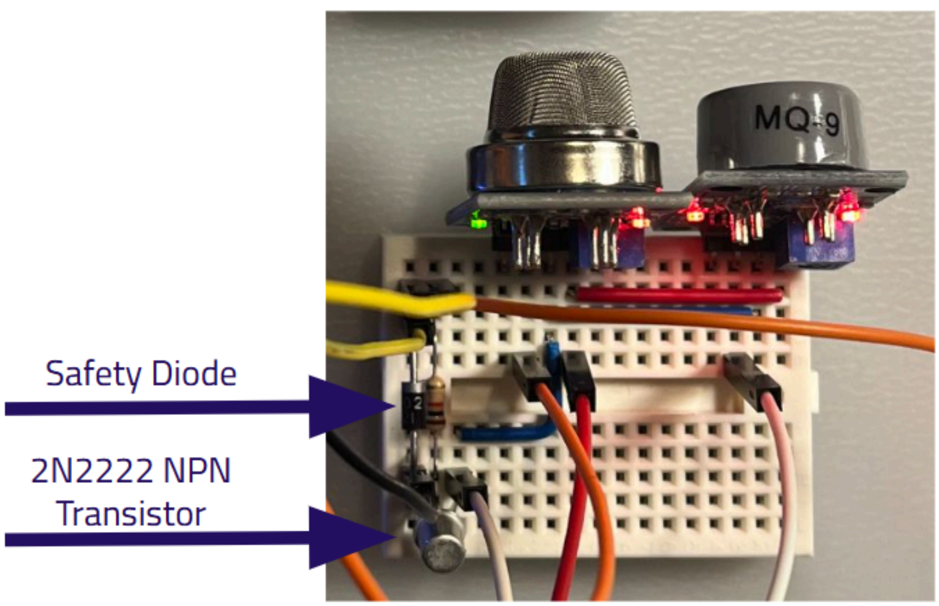


**Figure A2:** Close-up look at breadboard with transistor circuit to control ventilation fan.

# Appendix B: User Cloud Dashboard

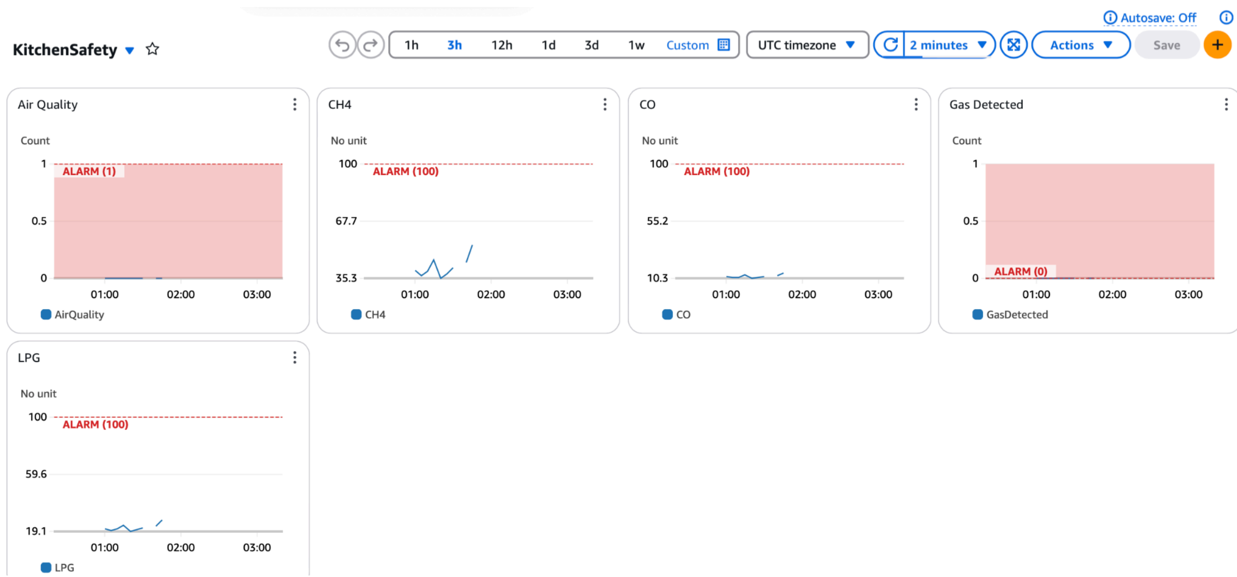


**Figure B1:** AWS Cloud Dashboard displaying all data topics.

# Appendix C: Cloud Notifications

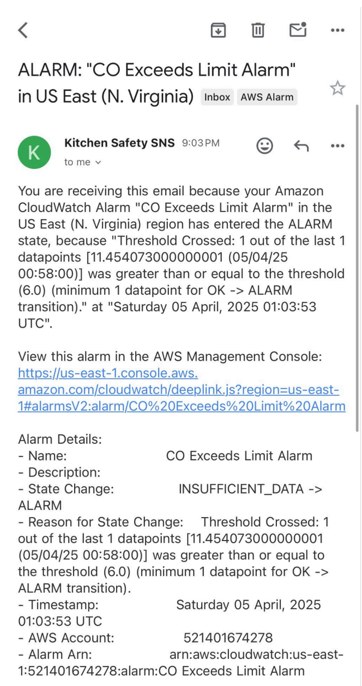


**Figure C1:** AWS Email Notification example.


**Figure C2:** AWS SMS text notification example.