Department of Telecommunications
and Networking
Capstone Project - Report

Smart Emergency System

Horn Bunhak

Men Rithydet

Ho Chandara

Choeng Cheatong

(Telecommunications and Network Engineering: Gen. 9 – Year 2_Term 3)

Advisor: Lihour NOV (Ph.D.)

Submission Date: 19 Jul 2024

Table of Contents

I. Introduction

Overview

Fire incidents are a major concern worldwide, causing significant damage and loss of life. Many homes do not have an effective fire detection system, which increases the risk of serious accidents. Our project aims to solve this problem by creating a smart home safety system that can detect fire and gas leaks. Using modern Internet of Things (IoT) technology, our system will send real-time alerts to homeowners, helping them respond quickly to emergencies. This report explains how we developed and built our system, from the initial research to the final product, and shows how our solution can improve home safety.

Problem Statement

According to our research data with both foreign and our country. We have found the same problem in each country. For instance, SCDF Annual Statistics 2023 have shown that in the last two years burn incidents still increase and it happen mostly every day. The overall number of fires increased by 8.6% to 1,954 cases in 2023 which means an average of 5 to 6 fires every day that SCDF responded to. The largest increase involves Active Mobility Devices (AMDs) which have increased by 31% compared to 2022 (Figure 1). The reason that mostly caused to burned incident is because of cooking and electricity (Figure 2). Also, Cambodia almost has a vast number of fire incidents. Mr. Tith Kongnov from Khmer TIme has reported “In 2022, there were 618 fires, which caused 17 deaths and 24 injuries, destroying 697 houses (109 stalls) that were damaged. In 2023, 761 cases of fire accidents occurred, killing 54 people, and injuring 97 others, while 579 houses (57 stalls).” (1)

As result, we can conclude that burning houses and building cause a lot of problems nowadays not in just some countries but a lot of countries are suffer from this problem as well, most importantly causing someone's life and finances in the family household due to small mistakes and sometimes due to the heat that led to the house burning down that why we saw that problem surfaces and start to find the solution.

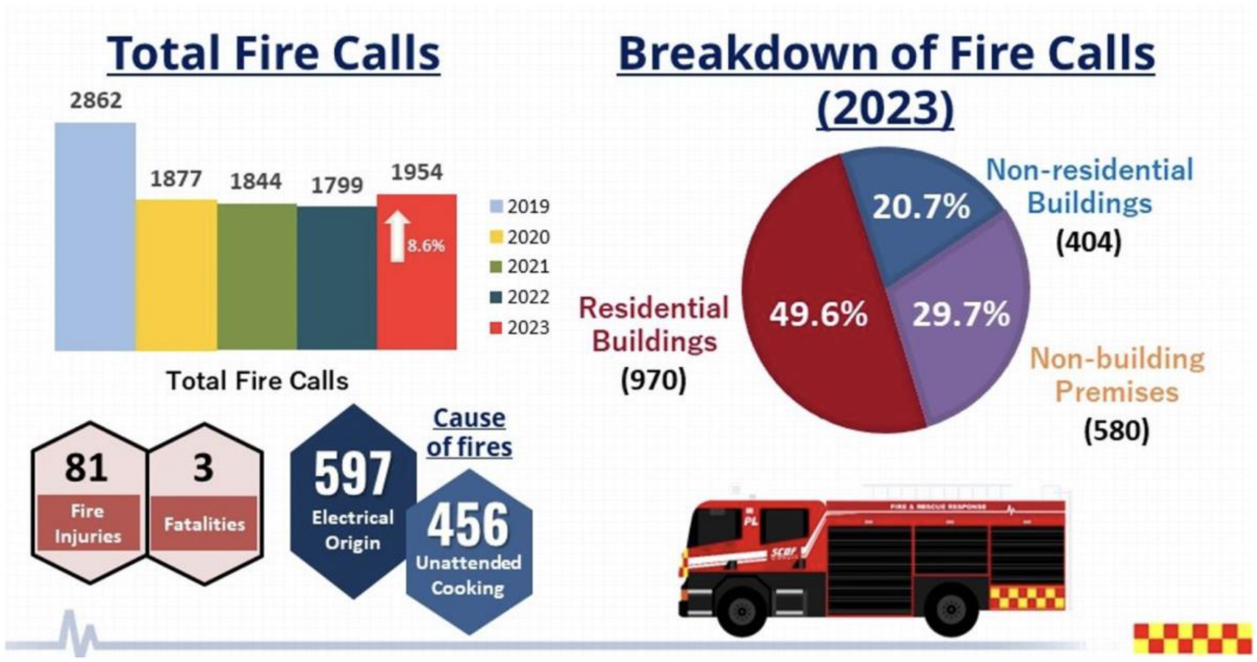
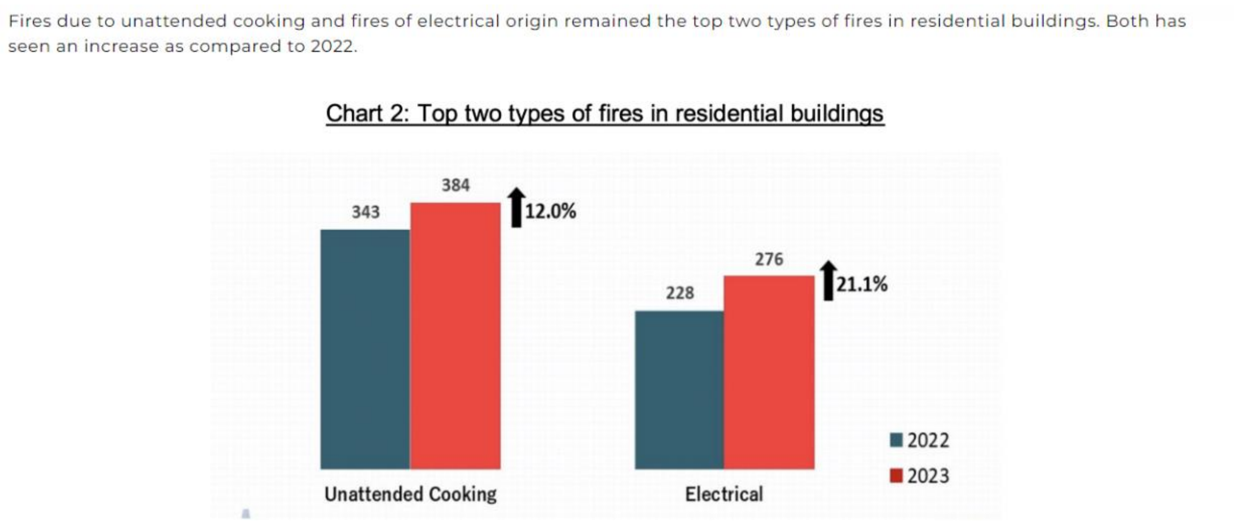


Fig. 1: Fire Incident Annual Statistics 2023 from Singapore



Fires due to unattended cooking and fires of electrical origin remained the top two types of fires in residential buildings. Both has seen an increase as compared to 2022.

Fig. 2: Annual statistic of the main reason that caused fire incident

Objective

In Cambodia, most homes lack fire detection systems or alarms to alert homeowners in case of gas leaks, fires, or smoke. Moreover, there is often no solution for a quick emergency response when a fire occurs while the residents are away. To address these issues and reduce the risk of catastrophic fires, we propose a comprehensive product called “ពន្លត់ PONLUT” designed to ensure greater safety for homeowners.

II. Methodology

Our project began with an extensive literature review to identify fire safety issues in society, leading us to define our project topic and compile a list of necessary components. We drafted the initial design, procured hardware such as the IR Flame Sensor, MQ2 Sensor, and ESP-32 Dev Module, and developed algorithms and flowcharts (Figure 3). We then integrated the sensors with the ESP-32 on a PCB board to ensure accurate readings (Figure 4) and spent several weeks coding and troubleshooting to ensure all functions operated correctly. Following iterative testing and problem-solving, we assembled the complete system and conducted comprehensive tests to confirm its reliability, culminating in a detailed presentation showcasing our project's features and results.

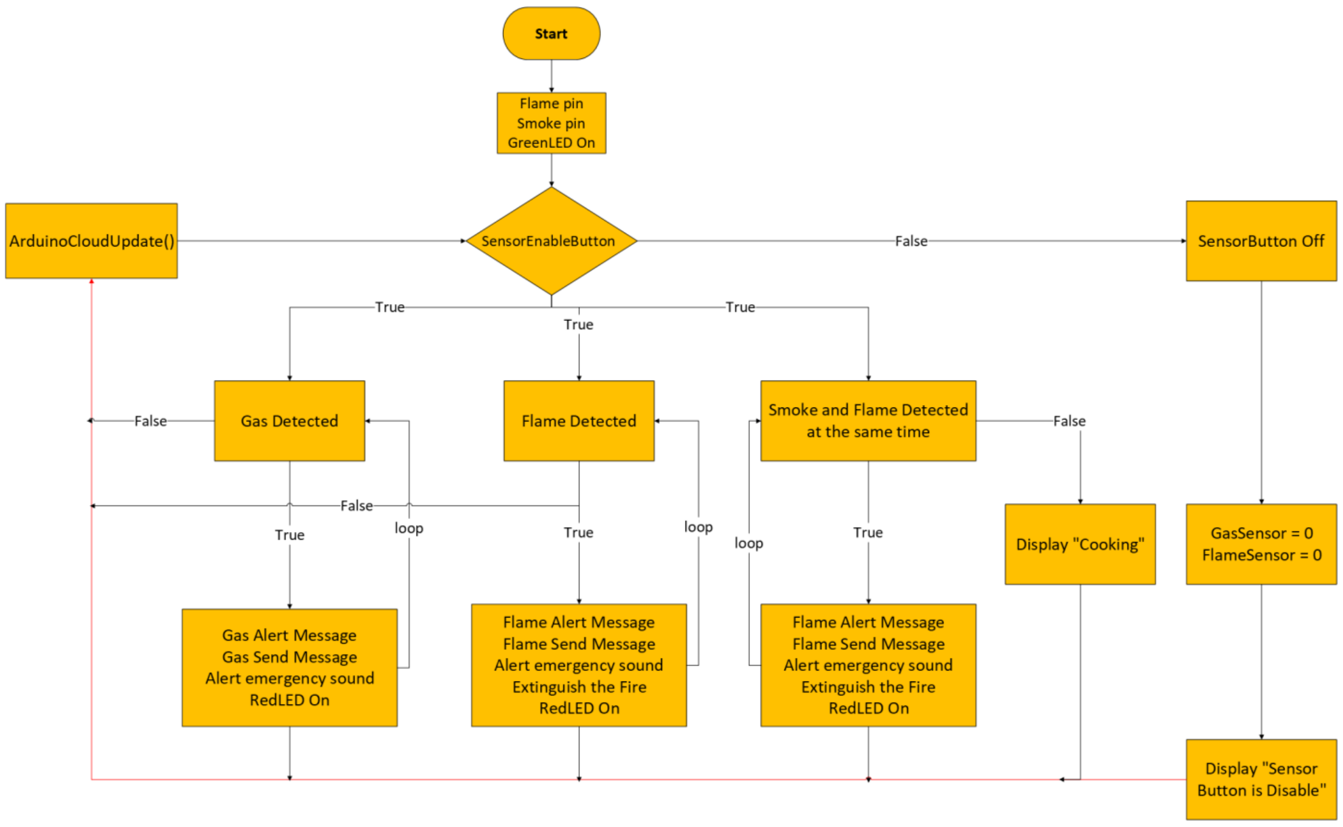


Fig. 3: Smart Emergency System Mechanism Flowchart Structure

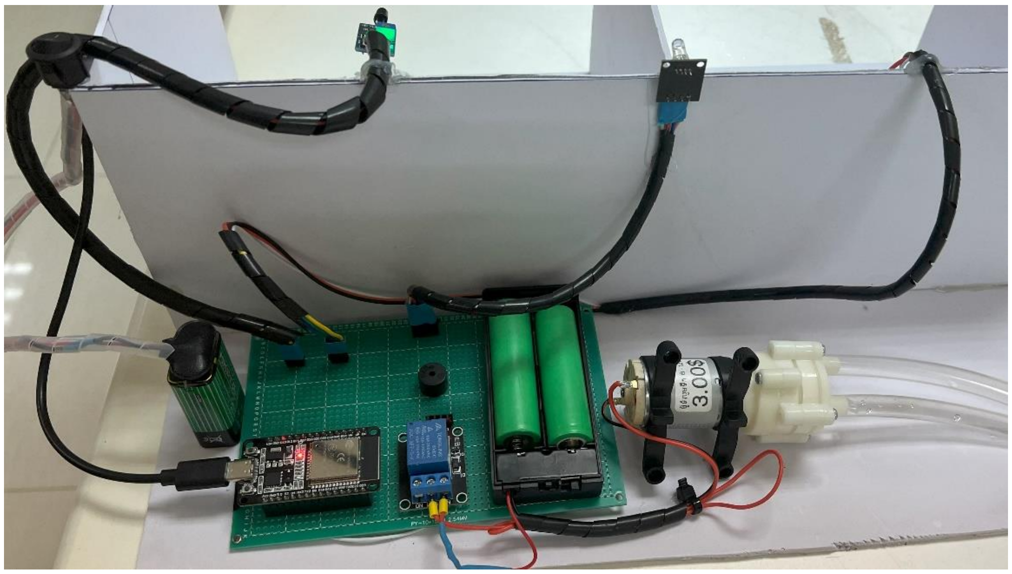
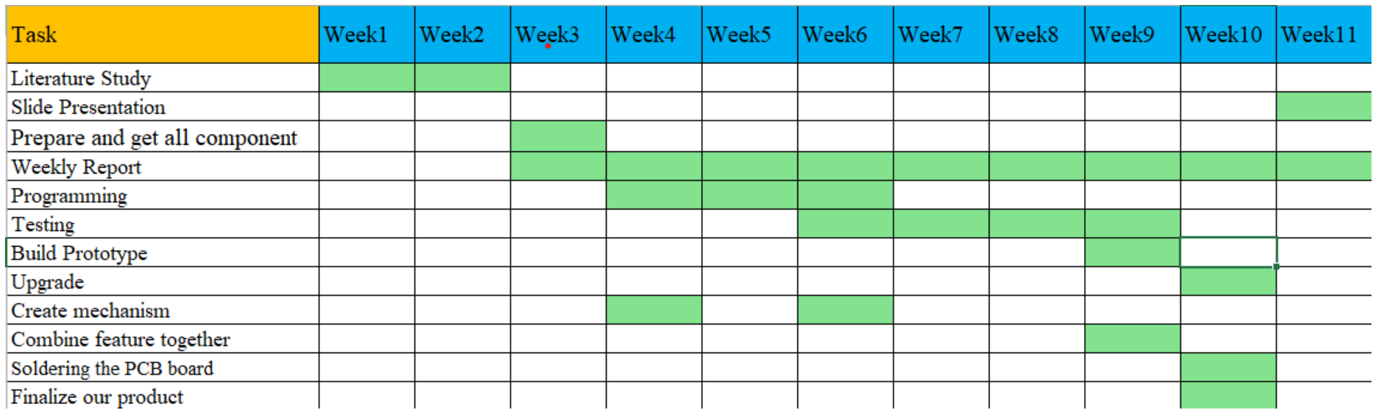


Fig. 4: Components integrated on PCB Board

Project Timeline

We have assigned our work and task with the table below:



Task	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11
Literature Study	🟩	🟩									
Slide Presentation											🟩
Prepare and get all component			🟩								
Weekly Report			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Programming				🟩	🟩	🟩					
Testing						🟩	🟩	🟩	🟩		
Build Prototype									🟩		
Upgrade										🟩	
Create mechanism				🟩		🟩					
Combine feature together									🟩		
Soldering the PCB board										🟩	
Finalize our product										🟩	

III. Implementation

In our project, we are integrating the IR Flame Sensor and MQ2 Sensor (for smoke and gas detection) with the ESP-32 Dev Module to create a smart home safety system. The ESP-32 Dev Module, developed by Espressif Systems, is a powerful microcontroller with built-in Wi-Fi and Bluetooth capabilities, making it ideal for Internet of Things (IoT) applications.

Component used

1. **ESP-32 Dev Module:** A versatile microcontroller with integrated Wi-Fi and Bluetooth, enabling wireless communication and internet connectivity. (Figure 5)

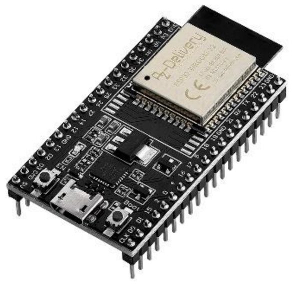


Fig. 5: ESP-32 Dev Module

2. **IR Flame Sensor:** Detects the presence of flames by sensing specific wavelengths of infrared light emitted by fire. (Figure 6)

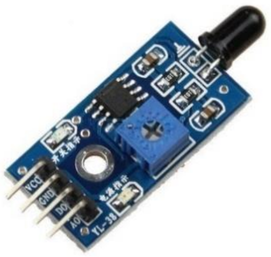


Fig. 6: Flame Sensor

3. **MQ2 Sensor:** Detects smoke and various gases, providing an early warning for potential fire hazards and gas leaks. (Figure 7)

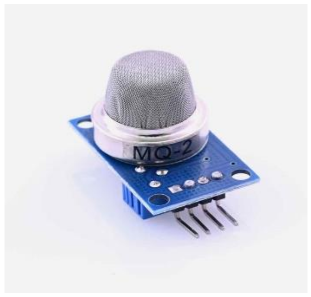


Fig. 7: MQ2 Sensor

Software (IDE)

To program and manage our smart home safety system, we use the Arduino IDE in conjunction with the Arduino IoT Cloud. The Arduino IoT Cloud provides a user-friendly platform for developing IoT applications, allowing us to seamlessly integrate our sensors and the ESP-32 Dev Module. Through the Arduino IDE, we write and upload the necessary code to the ESP-32, enabling it to process data from the IR Flame Sensor and MQ2 Sensor. The Arduino IoT Cloud facilitates remote monitoring and control, providing real-time insights and alerts directly to our smart devices. This combination ensures an efficient development process and robust functionality for our IoT project. (Figure 8)

Fig. 8: Arduino IoT Cloud

Implementation Step

1. **Sensor Integration:**
   - IR Flame Sensor: Connect the IR Flame Sensor to the ESP-32 Dev Module using the appropriate digital or analog pins. This sensor will continuously monitor for the presence of flames.
   - MQ2 Sensor: Connect the MQ2 Sensor to the ESP-32 Dev Module using the appropriate analog or digital pins. This sensor will monitor smoke and gas levels in the environment.

2. **Data Processing:**
   - The ESP-32 Dev Module collects data from both the IR Flame Sensor and MQ2 Sensor.
   - Using its built-in processing capabilities, the ESP-32 evaluates the sensor data to determine if there is a potential fire or gas leak.

3. **Communication and Alerts:**

   - When the IR Flame Sensor detects a flame or the MQ2 Sensor detects smoke or gas, the ESP-32 processes these signals.
   - The integrated Wi-Fi module in the ESP-32 allows it to connect to the internet and communicate with external smart devices such as smartphones or laptops.
   - The ESP-32 also communicates with the Arduino IoT Cloud, enabling it to send messages and monitor sensor data remotely.
   - Upon detecting any hazardous conditions, the ESP-32 sends an alert message or emergency notification to the user. The message will clearly indicate the nature of the hazard, such as "Flame Detected", "Smoke Detected" and "Gas Leak Detected".

4. **User Interaction:**
   - Users receive real-time notifications on their smart devices, allowing them to take immediate action.
   - The system ensures that users are always aware of potential dangers in their environment, enhancing home safety.

- Through the Arduino IoT Cloud, users can monitor sensor readings and system status, ensuring they have continuous insight into the safety of their home.

By combining the capabilities of the IR Flame Sensor, MQ2 Sensor, and ESP-32 Dev Module, our IoT project provides an effective solution for monitoring and alerting users to fire, and gas leaks. The integration of Wi-Fi connectivity and communication with the Arduino IoT Cloud ensures that alerts are promptly communicated and that users can monitor or track their system remotely.

Project Expenses

To implement our Smart Emergency Alarm project, we have budgeted for the following components. The table below outlines the items list of our components with their quantity, unit price and sub-total costs.

Table 1: Material and Components

Item List	Amount (pc/set)	Unit Price (USD)	Sub-Total (USD)
ESP-32 Dev	1	5	5
Flame Sensor	1	0.58	0.58
MQ2 Sensor	1	3	3
Water Pump	1	3	3
PCB Board DIY	1	1.20	1.20
Relay Module	1	1	1
		Total (USD):	13.78

IV. Results and Discussion

Achievement

As a result, we have completed all the features as we had planned, and it works smoothly and accurately. When it has a problem, the red-light alerts (Figure9), and sends a message to the owner to know about it (Figure10) as when it is normal it shows the green light (Figure11).

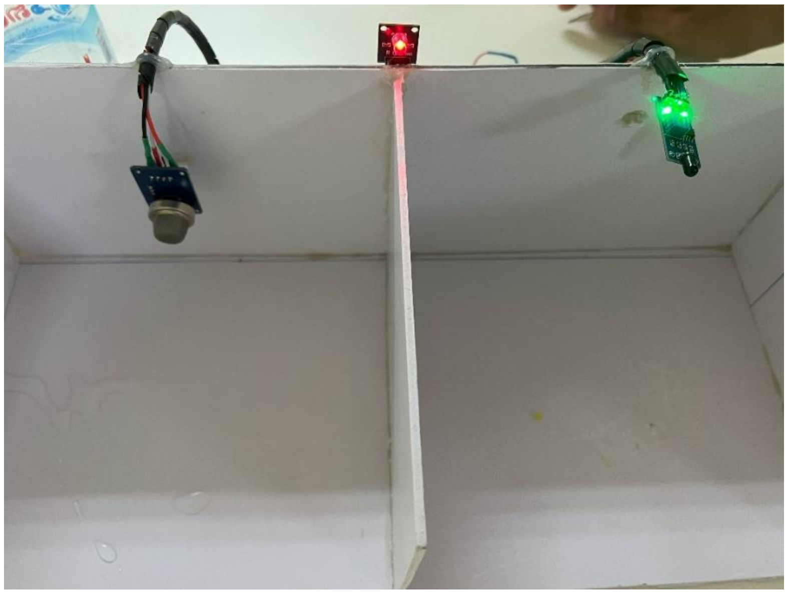


Fig9: When the problem occurs, the LED is RED

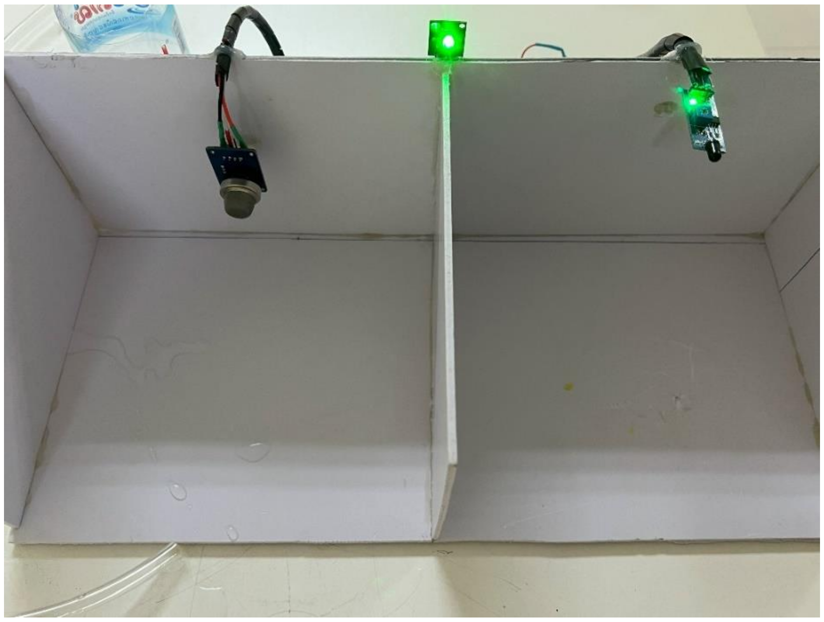
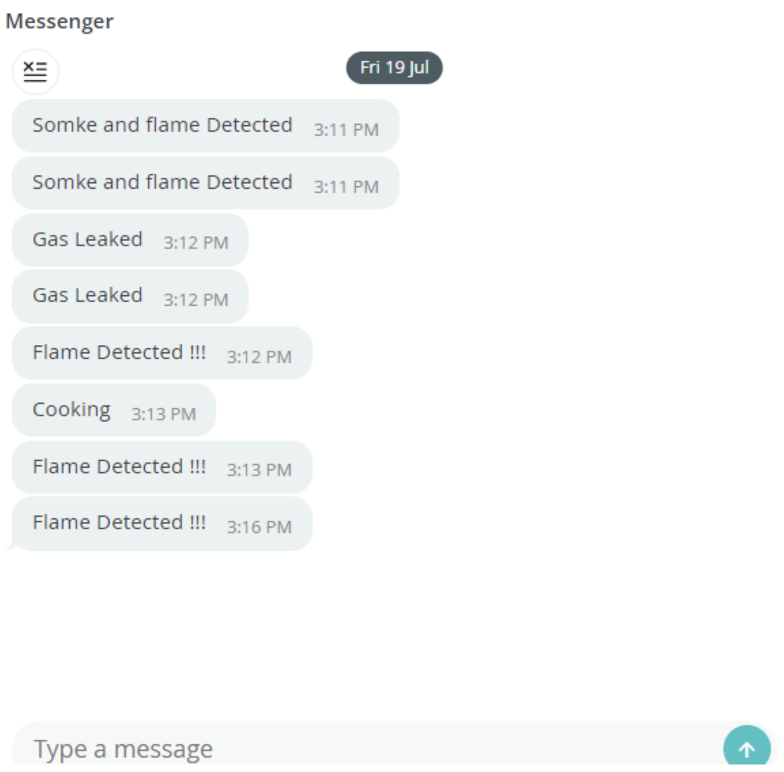


Fig10: Normal or safe stat, the LED is green



Somke and flame Detected 3:11 PM

Somke and flame Detected 3:11 PM

Gas Leaked 3:12 PM

Gas Leaked 3:12 PM

Flame Detected !!! 3:12 PM

Cooking 3:13 PM

Flame Detected !!! 3:13 PM

Flame Detected !!! 3:16 PM

Figure11: Message sent via cloud when the conditions true

Future development

Along that way, we also want to improve and change the sensors to enhance their performance and cover range. Furthermore, we hope to combine our sensor into multi-function to make it easy to implement and save space as well as to upgrade its beauty and appearance. In addition, we are also looking forward to improving our software and applications to get more functionality and other useful features like Blink or our app rather than Arduino Cloud.

V. Conclusion

In conclusion, we have completed the project successfully. During the process, we gained lots of knowledge about IoT things that we have never done before such as doing a literature study, knowing much more about sensors and cloud on how they work together, knowing how to handle problems with the best solution, and managing the timeline for a project. Moreover, we have gained some experience in how to do group work as well. Lastly, this project will help me prepare for my future education.

Reference

1. Kongnov, T. (2024, January 3). Big rise in fire deaths last year. *Khmer Times*.

Webpage: https://www.khmertimeskh.com/501416880/big-rise-in-fire-deaths-last-year/#:~:text=He%20added%20that%20in%202023,and%20109%20stalls%20were%20damaged

2. Alphabusinessdesigns.com. (2024, March 2). How much does it cost to install smoke alarms in 2024?

Webpage: https://bhullarelectricalsandsolar.com.au/how-much-does-it-cost-to-install-smoke-alarms-in-2024/

3. Toscano, M. (2024, March 25). How many house fires occur each year. *Consumer Affairs Journal of Consumer Research*.

Webpage: https://www.consumeraffairs.com/homeowners/how-many-house-fires-occur-each-year.html

4. Munir, M. (2022, November 24). Fire alarm with buzzer and water sprinkler system. *hacker.io*.

Webpage: https://www.hackster.io/munir03125344286/fire-alarm-with-buzzer-and-water-sprinkler-system-baf035

5. Nandanwar, U. (2021, September 5). Flame sensor. *github.com*

Webpage: https://github.com/un0038998/lessons/blob/main/Flame_Sensor/Flame_Sensor.ino

6. Alestair SG Life Protector. (2024, March 3). SCDF annual statistics 2023.

Webpage: https://alestairsg.com/scdf-annual-statistics-2023/

Appendix

Below is our source code implementation for integrating with the Arduino IoT Cloud.

```
#include "thingProperties.h"

const int BUZZER_PIN = 4;

const int redPin = 33;

const int greenPin = 32;

const int bluePin = 19;

const int GAS_SENSOR_PIN = 34;

const int FLAME_SENSOR_PIN = 27;

int gasSensorData = 0;

bool gasAlert = false;
bool fireAlert = false;

int flameState = 0;  // Variable to store the sensor state

const int relay_pin = 15;


void setup() {

  Serial.begin(9600);

  delay(1500);

  initProperties();

  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  setDebugMessageLevel(2);

  ArduinoCloud.printDebugInfo();


  pinMode(BUZZER_PIN, OUTPUT);

  pinMode(GAS_SENSOR_PIN, INPUT);

  pinMode(27,INPUT);

  pinMode(redPin, OUTPUT);

  pinMode(greenPin, OUTPUT);

  pinMode(bluePin, OUTPUT);

  pinMode(relay_pin,OUTPUT);

  digitalWrite(redPin, LOW);

  digitalWrite(greenPin, LOW);

  digitalWrite(bluePin, LOW);

  digitalWrite(relay_pin,1);

}


void loop() {
  ArduinoCloud.update();
  digitalWrite(relay_pin, 1);
  int flameState =  digitalRead(FLAME_SENSOR_PIN);


  //  SensorEnable Switch
  if (GasDetectionEnable == 1) {
    gasSensorData = analogRead(GAS_SENSOR_PIN);
    Serial.print("Gas Sensor Data: ");
    Serial.print(gasSensorData);
    Serial.print("  ");
    GAS_Value = gasSensorData;
    Serial.print("Flame Value: ");
    Serial.println(digitalRead(FLAME_SENSOR_PIN));


//Gas true only
    if (gasSensorData > 1000 && flameState == 1) {
      gasAlert = true;
      GASSiren();
      redLED_Blink(10, 100);
      digitalWrite(redPin, HIGH);
      digitalWrite(greenPin, LOW);
      digitalWrite(bluePin, LOW);
      redLED_Blink(10, 100);
      Serial.println("GAS LEAKED!!!");
      message = " GAS LEAKED !!!";
      ArduinoCloud.update(); // Ensure the message is sent immediately
```

```
    } else {
      gasAlert = false;
    }

    if (gasAlert) {
      delay(1000);
      return;
    }

//gas trure && flame true
     else if (gasSensorData >= 500 ) {
        if (flameState==0){
        digitalWrite(relay_pin,0);
        gasAlert = true;
        GASSiren();
        redLED_Blink(10, 200);
        digitalWrite(redPin, HIGH);
        digitalWrite(greenPin, LOW);
        digitalWrite(bluePin, LOW);
        redLED_Blink(10, 200);
        Serial.println("Flamed and Smoke Dected!!!");
        message = " Flamed and Smoke Dected!!!";
        ArduinoCloud.update(); // Ensure the message is sent immediately
        }else{
          Serial.println("Coocking!!!");
          message = "Coocking!!!";
          ArduinoCloud.update(); // Ensure the message is sent immediately
        }

    } else {
      gasAlert = false;
      digitalWrite(relay_pin,1);
    }

    if (gasAlert) {
      delay(1000);
      return;
    }

  // Flame Sensor Detection
    else if (digitalRead(FLAME_SENSOR_PIN) == 0) {
    fireAlert = true;
    Serial.println("Flame detected!!!!!!");
    message = " Flame Detected !!!";
    digitalWrite(redPin, HIGH);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, LOW);
    digitalWrite(relay_pin,0);
    GASSiren() ;
    ArduinoCloud.update(); // Ensure the message is sent immediately
   } else {
      delay(1000);
    fireAlert = false;
    digitalWrite(relay_pin,1);
    // Serial.println("No flame detected.");
  }

  if (fireAlert) {
    delay(1000); // Wait for 1 second before the next loop iteration
    return;
  }
  ArduinoCloud.update();

    digitalWrite(redPin, LOW);
    digitalWrite(greenPin, HIGH);
    digitalWrite(bluePin, LOW);

    delay(500);
  } else {
    gasSensorData = 0;
    Serial.println("Gas detection is disabled.");
    Serial.print("Gas Sensor Data: ");
    Serial.println(gasSensorData);
    GAS_Value = gasSensorData;
  }
  delay(500);
}
```

```
void GASSiren() {
  int tone1 = 1000; // First tone frequency
  int tone2 = 1500; // Second tone frequency
  int duration = 250; // Duration of each tone in milliseconds

  for (int i = 0; i < 4; i++) { // Repeat the pattern to create a continuous siren
    tone(BUZZER_PIN, tone1); // Play the first tone
    delay(duration);         // Wait for the duration
    tone(BUZZER_PIN, tone2); // Play the second tone
    delay(duration);         // Wait for the duration
  }
  noTone(BUZZER_PIN); // Turn off the buzzer
}
void FireSiren_1() {
  //digitalwrite(relay_pin,1);
  int tone1 = 1200; // First tone frequency
  int tone2 = 1800; // Second tone frequency
  int duration = 500; // Duration of each tone in milliseconds

  for (int i = 0; i < 3; i++) { // Repeat the pattern to create a continuous siren
    tone(BUZZER_PIN, tone1); // Play the first tone
    delay(duration);         // Wait for the duration
    redLED_Blink(10, 50);     // blink 10 time,each for 50ms
    tone(BUZZER_PIN, tone2); // Play the second tone
    delay(duration);         // Wait for the duration
    redLED_Blink(10, 50);
  }
  noTone(BUZZER_PIN); // Turn off the buzzer
}

void FireSiren_2() {
  int lowFreq = 500;  // Low frequency for the wail
  int highFreq = 2500; // High frequency for the wail
  int stepDelay = 5;  // Delay between frequency changes

  // Ramp up the frequency
  for (int i = lowFreq; i < highFreq; i += 5) {
    tone(BUZZER_PIN, i);
    delay(stepDelay);
  }
  // Ramp down the frequency
  for (int i = highFreq; i > lowFreq; i -= 5) {
    tone(BUZZER_PIN, i);
    delay(stepDelay);
  }
}
void redLED_Blink(int blinkCount, int blinkDuration) {
  for (int i = 0; i < blinkCount; i++) {
    digitalWrite(redPin, HIGH);
    delay(blinkDuration);
    digitalWrite(redPin, LOW);
    delay(blinkDuration);
  }
}
```

Team members and Responsibilities

Profile	Main Responsibility
	Team Leader: Horn Bunhak - Team management - Project management - Checkup and monitor work and task - Troubleshooting / Problem solving - Literature Study
	Member 1: Men Rithydet - Managing components and Testing - Implement code - Literature Study - Assist Troubleshooting
	Member 2: Choeng Cheatong - Implement hardware - Report / Slide Presentation - Prototype Design - Assist Troubleshooting - Literature Study

	Member 3: Ho Chandara - Implement hardware - Prototypes Design - Report / Slide Presentation - Literature Study - Assist solving problem