

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

ĐỀ TÀI:
HỆ THỐNG CẢNH BÁO CHÁY DỰA TRÊN NHIỆT ĐỘ
SỬ DỤNG ESP8266 (WEMOS D1 R2)

Giảng viên hướng dẫn: **Lê Anh Tuấn**
Nguyễn Văn Bảo

Sinh viên thực hiện:
Hoàng Thị Diễm Quỳnh - 19520894
Nguyễn Lê Thảo Ngọc - 21521191

TP. HỒ CHÍ MINH, 2024

LỜI CẢM ƠN

Lời đầu tiên, nhóm chúng em xin trân trọng cảm ơn giảng viên Lê Anh Tuấn người đã giúp chúng em hiểu rõ hơn về cơ sở lý thuyết và giảng viên Nguyễn Văn Bảo đã hỗ trợ nhóm về mặt kỹ thuật cho nhóm em trong quá trình hoàn thành đồ án này.

Mặc dù đã có những đầu tư nhất định trong quá trình làm bài song cũng khó có thể tránh khỏi những sai sót, em kính mong nhận được ý kiến đóng góp của quý thầy/cô để đồ án của chúng em được hoàn thiện hơn nữa.

Nhóm em xin chân thành cảm ơn!

Mục lục

Chương 1. GIỚI THIỆU VỀ ĐỀ TÀI	6
1. Tổng quan đề tài.....	6
2. Lý do chọn đề tài.....	6
3. Tóm tắt đề tài.....	7
Chương 2. DỤNG CỤ VÀ THIẾT BỊ SỬ DỤNG.....	8
1. Board ESP8266 Wemos D1 - R2.....	8
2. Phần mềm lập trình Arduino IDE	9
3. ThingSpeak	11
4. Cảm biến nhiệt độ DHT22.....	12
5. Cảm Biến Khí Gas (LPG/CO/CH4) MQ-2.....	14
6. Các thiết bị khác.....	15
7. MATLAB® Analysis in ThingSpeak.....	17
Chương 3. CƠ SỞ LÝ THUYẾT	20
1. Mô hình triển khai	20
2. Quy trình hoạt động	20
3. Thư viện được sử dụng trong Arduino IDE	21
3.1 ESP8266WiFi.h.....	21
3.2 DHT.h.....	21
3.3 ArduinoJson.h.....	22
Chương 4. KỊCH BẢN TRIỂN KHAI	23
1. Mô hình thiết kế	23
2. Quy trình hoạt động	24

3. Trục quan hoá dữ liệu	25
Chương 5: KẾT QUẢ THỰC NGHIỆM	28
1. Hiện thực phần cứng.....	28
2. Hiện thực chương trình.....	29
3. Giao diện người dùng.....	31
4. Mobile application.....	34
TÀI LIỆU THAM KHẢO	37

DANH MỤC HÌNH ẢNH

Hình 2. 1 : WeMos D1 R2 WiFi UNO Based ESP8266.....	9
Hình 2.2. 1: Phần mềm code Arduino IDE.....	10
Hình 2.2. 2: Giao diện Arduino IDE.....	11
Hình 2. 3: Thingspeak.....	12
Hình 2. 4: Cảm biến nhiệt độ DHT22.....	13
Hình 2.5. 1: Mặt sau của cảm biến MQ-2	14
Hình 2.5. 2: Mặt trước cảm biến khí gas MQ-2.....	15
Hình 2.6. 1: Bảng mạch.	15
Hình 2.6. 2: Điện trở.....	16
Hình 2.6. 3: Đèn led.....	16
Hình 2.6. 4: Dây nối.....	17
Hình 4. 1 Mô hình thiết kế hệ thống cảnh báo cháy	23
Hình 4. 2 Quy trình hoạt động của mô hình	24
Hình 4.3. 1 Biểu đồ nhiệt độ	25
Hình 4.3. 2 Biểu đồ độ ẩm.....	25
Hình 4.3. 3 Theo dõi trạng thái đèn.....	26
Hình 4.3. 4 Theo dõi nồng độ khí gas	26
Hình 5. 1: Mạch thiết kế hoàn thiện.....	28
Hình 5.3 1 Hiển thị thông tin nhiệt độ	31
Hình 5.3 2 Hiển thị thông tin độ ẩm	32
Hình 5.3 3 Hiển thị thông tin khí gas (các loại khí dễ gây cháy)	32
Hình 5.3 4 Hiển thị tín hiệu cảnh báo.....	33
Hình 5.3 5 Hiển thị các thông tin khác	33
Hình 5.3 6 Email cảnh báo nhận được khi hệ thống phát hiện nhiệt độ	34
Hình 5.4 1 Giao diện mobile.....	35
Hình 5.4 2 Giao diện mobile.....	35
Hình 5.4 3 Giao diện mobile.....	36
Hình 5.4 4 Giao diện mobile.....	36

Chương 1. GIỚI THIỆU VỀ ĐỀ TÀI

1. Tổng quan đề tài

Ngày nay dưới tác động của biến đổi khí hậu, điều kiện thời tiết oi bức, kết hợp nền nhiệt cao khiến nhu cầu sử dụng các thiết bị điện của người dân tăng nhanh. Điều này tiềm ẩn rất nhiều nguy cơ cháy nổ. Bởi các thiết bị điện khi dùng quá nhiều, công suất lớn rất dễ bị quá tải khiến chập điện hay nóng tới mức gây cháy đồ đạc xung quanh.

Có thể thấy rõ, vụ cháy dù lớn hay nhỏ đều sẽ gây ra các thiệt hại lớn nên tốt nhất là chúng ta cần chủ động đề phòng và tìm cách ngăn chặn nó. Tuy nhiên với các nguyên nhân gây cháy khách quan như nhiệt độ môi trường cao, bởi quên tắt bếp, thiết bị điện quá tải hay bị chập,...thì con người khó mà phòng bị được bởi chúng là ngẫu nhiên, không mong muốn. Do đó chúng ta cần một hệ thống nhận biết được dấu hiệu của hoả hoạn như là nhiệt độ cao, khói, tia lửa,... để giúp chúng ta dễ dàng phát hiện sớm các vụ cháy trước khi chúng bùng lớn và gây nhiều thiệt hại hơn. Thêm vào đó, hệ thống cũng cần có chức năng cảnh báo từ xa để tiện cho việc phát hiện và xử lý kịp thời cho những người liên quan tới vụ cháy.

Với đề tài “HỆ THỐNG CẢNH BÁO CHÁY DỰA TRÊN NHIỆT ĐỘ”, nhóm chúng em sẽ xây dựng hệ thống cảnh báo cháy từ xa với chức năng cung cấp thông tin về nhiệt độ, độ ẩm của môi trường xung quanh hệ thống. Điều này không những giúp ta kịp thời biết được sự cháy nổ của thiết bị khác mà còn giúp ta nhận biết nguy cơ cháy nổ từ chính hệ thống này.

Bên cạnh đó, việc ứng dụng ThingSpeak giúp hệ thống trở nên thân thiện và dễ sử dụng với người dùng hơn.

2. Lý do chọn đề tài

Những năm gần đây, tỉ lệ hoả hoạn xảy ra ngày càng cao nhất là ở các nơi đông dân cư như chung cư, nhà trọ,...Các vụ cháy dù lớn hay nhỏ đều có kịp thời ứng phó và giải quyết thì cũng đều để lại thiệt hại lớn cho những người liên quan. Do đó việc phòng ngừa, ngăn chặn cháy nổ trước khi nó ngoài tầm kiểm soát là việc hết sức quan trọng.

Nhưng không phải lúc nào chúng ta cũng có thể can thiệp, phát hiện kịp thời các dấu hiệu trước khi xảy ra hoả hạn, vì vậy nhóm đã thiết kế mô hình cảnh báo cháy dựa trên cảm biến nhiệt độ để có thể phát hiện nguy cơ hoả

hoạn trước khi nó bùng lên. Nhóm 02 đã chọn đề tài này với mong muốn tìm hiểu về công nghệ cảnh báo cháy, và có thể phát triển một dự án cảnh báo cháy tốt hơn trong tương lai.

3. Tóm tắt đề tài

Hệ thống gồm một ứng dụng web (dùng ThingSpeak) để hiển thị các số liệu về nhiệt độ, độ ẩm. Phần cứng gồm vi điều khiển ESP8266 thu thập dữ liệu từ các cảm biến và báo động bằng đèn khi phát hiện nhiệt độ/nồng độ khí dễ cháy cao hơn ngưỡng an toàn và một mạch IoT (như DHT22) có khả năng theo dõi nhiệt độ và độ ẩm môi trường, một mạch cảm biến khí gas (MQ2).

Người dùng sẽ truy cập ThingSpeak để xem các thông tin về nhiệt độ, độ ẩm, nồng độ khí gas khi có nhu cầu.

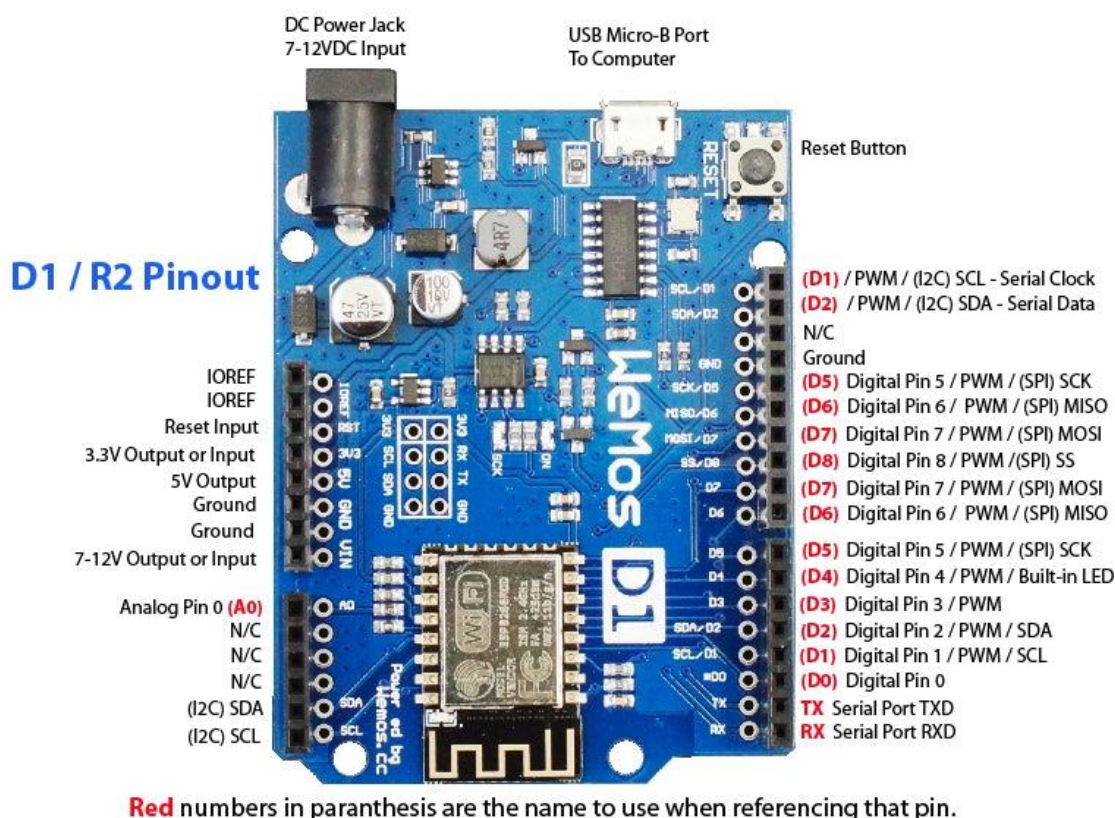
Chương 2. DỤNG CỤ VÀ THIẾT BỊ SỬ DỤNG

1. Board ESP8266 Wemos D1 - R2

ESP8266 Wemos D1 R2 là một phiên bản của bo mạch phát triển dựa trên vi điều khiển ESP8266 của Espressif Systems - một con chip Wi-Fi rất phổ biến và có khả năng xử lý cao; được sử dụng nhiều trong các ứng dụng Internet of Things (IoT). Một số thông tin cơ bản về Wemos D1 R2 bao gồm:

- + Vi điều khiển: ESP8266EX SoC (System-on-a-Chip) có xung nhịp 80MHz, dựa trên kiến trúc Tensilica Xtensa LX106.
- + Điện áp hoạt động: Wemos D1 R2 hoạt động với điện áp 3.3V.
- + Khe cắm Micro USB: Được sử dụng để cấp nguồn và giao tiếp với máy tính thông qua cổng USB.
- + Kích thước: Board có kích thước nhỏ gọn, tương đương với một board Arduino Uno, đồng thời cung cấp các tính năng của ESP8266.
- + Hỗ trợ lập trình: Wemos D1 R2 có thể được lập trình bằng Arduino IDE, sử dụng ngôn ngữ Arduino và các thư viện hỗ trợ ESP8266.
- + EEPROM: Board hỗ trợ bộ nhớ EEPROM có kích thước 512 bytes để lưu trữ dữ liệu người dùng.
- + Đèn LED: Board có đèn LED báo trạng thái và phát triển.
- + Wi-Fi: Board hỗ trợ kết nối Wi-Fi chuẩn IEEE 802.11 b/g/n. Nó có anten tích hợp và cung cấp khả năng kết nối mạng không dây.
- + GPIO: Wemos D1 R2 có 11 chân GPIO (General Purpose Input/Output) có thể được sử dụng để kết nối với các thiết bị ngoại vi khác.
- + UART: Board hỗ trợ giao tiếp UART (Universal Asynchronous Receiver-Transmitter) để kết nối với các thiết bị ngoại vi qua cổng serial.

ESP8266 Wemos D1 R2 có các tính năng và giao diện tương tự như Arduino Uno, nhưng thay vì sử dụng vi điều khiển AVR truyền thống như Arduino Uno, ESP8266 Wemos D1 R2 lại sử dụng vi điều khiển ESP8266. Hai vi điều khiển trên có sự khác nhau về các đặc điểm như: trong khi ESP8266 được thiết kế đặc biệt để hỗ trợ kết nối Wi-Fi thì AVR lại không, điều này cho phép ESP8266 dễ dàng kết nối và giao tiếp thông qua mạng không dây; AVR sử dụng kiến trúc RISC truyền thống, trong khi ESP8266 sử dụng kiến trúc RISC của Tensilica (kiến trúc Tensilica cung cấp hiệu năng cao hơn và khả năng xử lý tốt hơn so với AVR); ESP8266 hoạt động ở tốc độ xung nhịp cao hơn so với các vi điều khiển AVR thông thường, mang lại khả năng xử lý nhanh chóng và hiệu suất cao hơn trong các ứng dụng phức tạp.



Hình 2. 1 : WeMos D1 R2 WiFi UNO Based ESP8266

2. Phần mềm lập trình Arduino IDE

Arduino IDE (Integrated Development Environment) là một phần mềm mã nguồn mở được thiết kế đặc biệt cho việc lập trình vi điều khiển Arduino. Chủ yếu được sử dụng để chỉnh sửa, biên dịch và tải lên mã trong Thiết bị Arduino.

Hầu hết tất cả các mô-đun Arduino đều tương thích với phần mềm này.



Hình 2.2. 1: *Phần mềm code Arduino IDE*

Arduino IDE cung cấp một thư viện mã nguồn mở phong phú, bao gồm các chức năng và module tiện ích cho việc lập trình Arduino. Bên cạnh đó, nó cũng hỗ trợ việc quản lý thư viện mở rộng và tải chúng từ nhiều nguồn tài nguyên trực tuyến. Với Arduino IDE, người dùng có thể viết mã nguồn, xem thông báo lỗi và cảnh báo, nạp chương trình vào vi điều khiển, và theo dõi quá trình chạy của chương trình trên vi điều khiển Arduino.



Hình 2.2. 2: Giao diện Arduino IDE

Một số mô-đun Arduino có sẵn bao gồm: Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro và nhiều mô-đun khác. Mỗi mô-đun trong số đó chứa một bộ vi điều khiển trên bo mạch được lập trình và tiếp nhận thông tin dưới dạng mã (code). Mã chính (maincode), còn được gọi là “sketch” (bản phác thảo), được tạo trên nền tảng IDE sẽ tạo ra một tệp Hex, sau đó được chuyển và tải lên trong bộ điều khiển trên board. Môi trường IDE chủ yếu chứa hai phần cơ bản: Trình chỉnh sửa và Trình biên dịch, nơi trước đây được sử dụng để viết mã được yêu cầu và sau đó được sử dụng để biên dịch và tải mã lên Mô-đun Arduino nhất định. Môi trường này hỗ trợ cho cả ngôn ngữ C và C++.

3. ThingSpeak

ThingSpeak là một nền tảng IoT (Internet of Things) được phát triển bởi MathWorks. Nó cung cấp các tính năng và dịch vụ để thu thập, lưu trữ, phân tích và hiển thị dữ liệu từ các thiết bị IoT.



Hình 2. 2: *Thingspeak*.

Một số tính năng chính của ThingSpeak bao gồm:

Kết nối thiết bị: Cho phép kết nối các thiết bị IoT như Arduino, Raspberry Pi, ESP8266, v.v. để gửi dữ liệu lên ThingSpeak. Hỗ trợ nhiều giao thức kết nối như HTTP, MQTT, WebSocket.

Lưu trữ và quản lý dữ liệu: Cung cấp kênh (channel) để lưu trữ và quản lý dữ liệu từ các thiết bị. Mỗi kênh có thể chứa nhiều trường để lưu trữ các loại dữ liệu khác nhau. Dữ liệu được lưu trữ trong các kênh có thể truy cập, tải xuống và phân tích dễ dàng.

Phân tích và hiển thị dữ liệu: Cung cấp công cụ phân tích và trực quan hóa dữ liệu như biểu đồ, bảng và các tính năng phân tích dữ liệu. Cho phép tạo các bản tin (app) để hiển thị dữ liệu trên web hoặc thiết bị di động.

Tự động hóa và Ứng dụng: Hỗ trợ các tính năng tự động hóa như đặt lịch gửi dữ liệu, cảnh báo dựa trên ngưỡng. Tích hợp với nhiều dịch vụ khác như IFTTT, Zapier để liên kết với các ứng dụng khác.

Trong hệ thống cảnh báo cháy rừng này, ThingSpeak sẽ đóng vai trò là một nền tảng cung cấp các dịch vụ để thu thập, lưu trữ, phân tích và hiển thị dữ liệu.

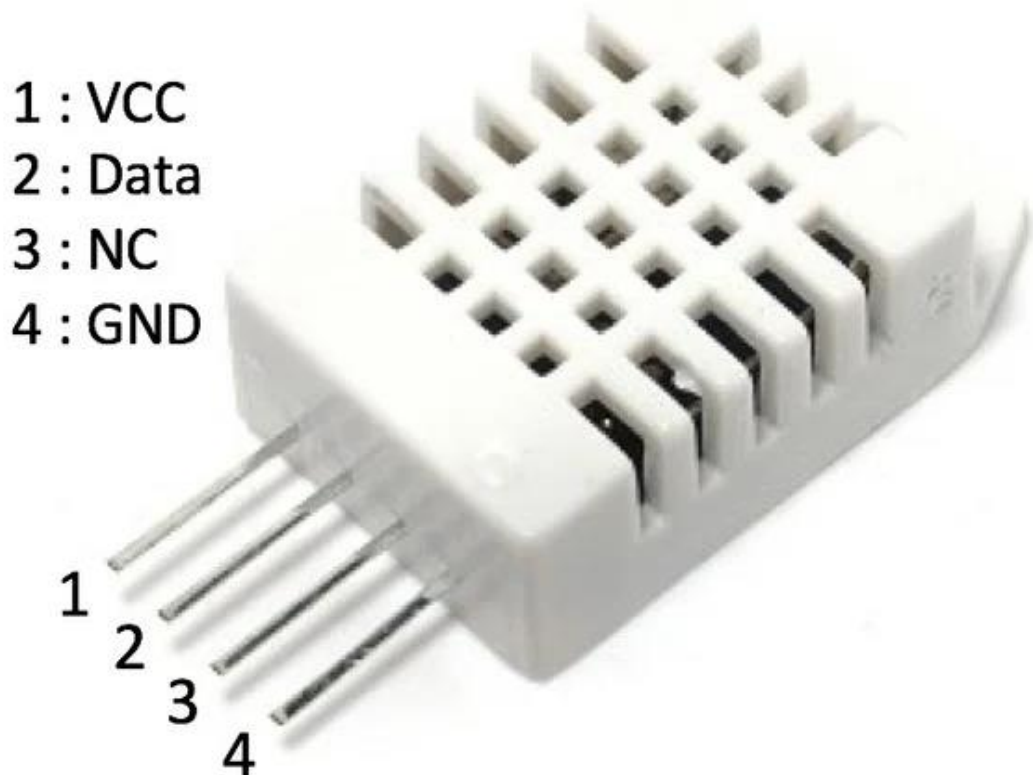
4. Cảm biến nhiệt độ DHT22

Cảm biến nhiệt độ DHT22 là một cảm biến kỹ thuật số được sử dụng để đo nhiệt độ và độ ẩm trong môi trường, được phổ biến trong các dự án điện tử và IoT.

Cảm biến DHT được chia thành hai phần, một cảm biến độ ẩm dựa trên nguyên lý điện dung và một linh kiện nhiệt trở. Bên trong cảm biến cũng có một con chip đơn giản thực hiện một số chuyển đổi tín hiệu tương tự sang tín hiệu số và đưa ra kết quả về nhiệt độ và độ ẩm dưới dạng tín hiệu số. Tín hiệu kỹ thuật số này có thể đọc bằng bất kỳ vi điều khiển nào.

Cảm biến DHT22 sử dụng giao tiếp dựa trên giao thức 1-Wire, cho phép dễ dàng kết nối và đọc dữ liệu từ cảm biến thông qua một chân dữ liệu duy nhất (chân Data). Cảm biến này có khả năng đo nhiệt độ trong khoảng từ -40 đến 80 độ Celsius và độ ẩm từ 0% đến 100%.

Để sử dụng cảm biến nhiệt độ DHT22, cần kết nối chân dữ liệu của cảm biến với một chân GPIO của vi điều khiển (ví dụ: Arduino). Sau đó, ta có thể sử dụng các thư viện hỗ trợ như thư viện DHT hoặc thư viện Adafruit DHT để đọc dữ liệu từ cảm biến.



Hình 2. 3: Cảm biến nhiệt độ DHT22.

5. Cảm Biến Khí Gas (LPG/CO/CH4) MQ-2

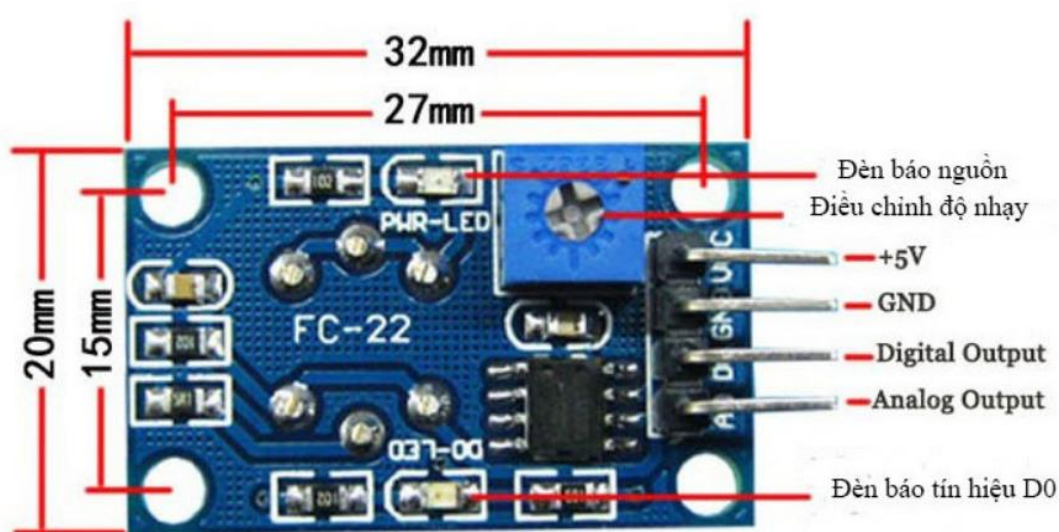
Cảm biến khí Gas MQ2 là một trong những cảm biến được sử dụng rộng rãi nhất trong các dòng cảm biến MQ. Nó là một cảm biến MOS (Metal Oxide Semiconductor). Cảm biến oxit kim loại hay còn được gọi là (Điện trở hóa trị) vì cảm biến dựa trên sự thay đổi điện trở của cảm biến khi tiếp xúc với khí.

Cảm biến khí gas arduino hoạt động trên 5V DC và tiêu thụ khoảng 800mW. Nó có thể phát hiện nồng độ LPG, Khí, Rượu, Propane, Hydrogen, Methane và Carbon Monoxide từ 200 đến 10000 ppm.

Tuy cảm biến MQ2 phát hiện nhiều loại khí, nhưng không thể xác định chúng đó là loại khí nào mà hầu hết các cảm biến khí trên thị trường hiện nay đều hoạt động theo cách này. Do đó, nó phù hợp để đo hoặc phát hiện nồng độ khí.

Cảm biến khí Gas MQ2 hoạt động dựa trên nguyên lý phản ứng hóa học khi tiếp xúc các khí trong môi trường. Nguyên tắc hoạt động của nó là khi các khí trong môi trường như khí CO, khí LPG, khí methane, khí Hydro, khói... tiếp xúc với phần tử bên trong cảm biến, làm cho các electron được giải phóng vào Thiếc Dioxide cho phép dòng điện chạy qua cảm biến một cách tự do. Khi được làm nóng, phần tử cảm biến sẽ tạo ra phản ứng hóa học với các khí tiếp xúc và làm thay đổi điện trở của phần tử cảm biến. Cảm biến MQ2 đo lường các biến đổi điện trở này và chuyển đổi chúng thành tín hiệu điện analog hoặc Digital.

SƠ ĐỒ MẠCH CẢM BIẾN KHÍ GAS MQ2:

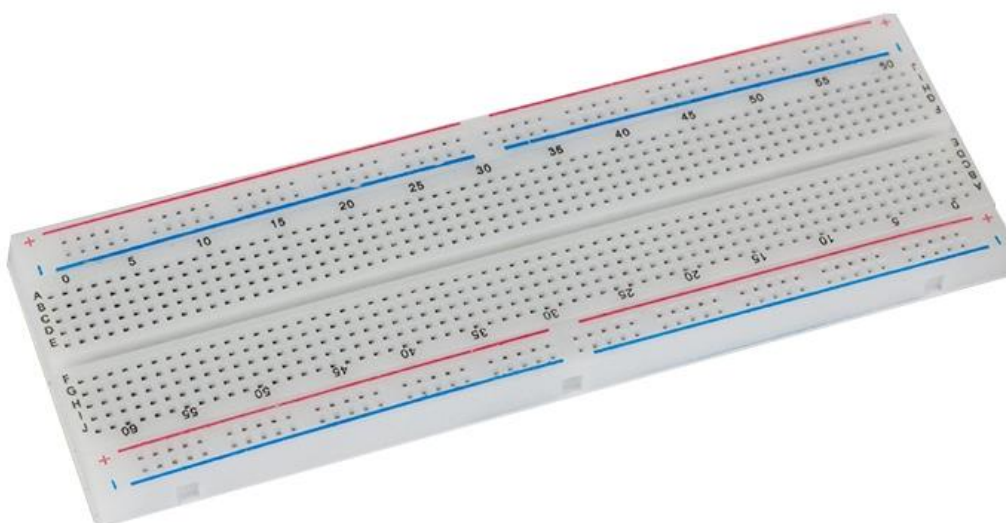


Hình 2.5. 1: Mặt sau của cảm biến MQ-2



Hình 2.5. 2: Mặt trước cảm biến khí gas MQ-2

6. Các thiết bị khác



Hình 2.6. 1: Bảng mạch.



Hình 2.6. 2: Điện trở.



Hình 2.6. 3: Đèn led.



Hình 2.6. 4: Dây nối.

7. MATLAB® Analysis in ThingSpeak



Hình 2.7 1

Matlab (tên viết tắt của Matrix laboratory) là phần mềm cung cấp môi trường tính toán số và lập trình, do công ty MathWorks thiết kế. Matlab cho phép tính toán số với ma trận, vẽ đồ thị hàm số hay biểu đồ thông tin, thực hiện thuật toán, tạo các giao diện người dùng và liên kết với những chương trình máy tính viết trên nhiều ngôn ngữ lập trình khác.

Matlab dùng để giải quyết các bài toán về giải tích số, xử lý tín hiệu số, xử lý đồ họa mà không phải lập trình cổ điển. Hiện nay, Matlab có đến hàng ngàn lệnh và hàm tiện ích. Ngoài các hàm cài sẵn trong chính ngôn ngữ, Matlab còn có các lệnh và hàm ứng dụng chuyên biệt trong các Toolbox để mở rộng môi trường Matlab, nhằm giải quyết các bài toán thuộc các phạm trù riêng.

Một số chức năng nổi bật của MATLAB:

- Cung cấp môi trường tương tác để khảo sát, thiết kế và giải quyết các vấn đề.
- Cung cấp thư viện lớn các hàm toán học cho đại số tuyến tính, thống kê, phân tích Fourier, bộ lọc, tối ưu hóa, tích phân và giải các phương trình vi phân bình thường.
- Matlab cung cấp các đồ thị được tích hợp sẵn để hiển thị hình ảnh dữ liệu và các công cụ để tạo đồ thị tùy chỉnh.
- Giao diện lập trình của Matlab cung cấp các công cụ phát triển để nâng cao khả năng bảo trì chất lượng mã và tối đa hóa hiệu suất.
- Cung cấp các công cụ để xây dựng các ứng dụng với các giao diện đồ họa tùy chỉnh.
- Cung cấp các hàm để tích hợp các thuật toán dựa trên Matlab với các ứng dụng bên ngoài và các ngôn ngữ khác như C, Java, NET và Microsoft Excel.

Chúng ta có thể sử dụng MATLAB Analysis để chuẩn bị, sàng lọc và thực hiện phân tích dữ liệu như tính độ ẩm trung bình, loại bỏ giá trị ngoại ra trong channel của ThingSpeak. ThingSpeak cho phép chúng ta sử dụng MATLAB để phân tích và trực quan hóa dữ liệu của mình.

Trong ThingSpeak, chúng ta thường sử dụng MATLAB Analysis để khai thác dữ liệu lưu trữ trong các channel của ThingSpeak. Sau đó sử dụng TimeControl để lập lịch thực thi chương trình viết bằng MATLAB hoặc sử dụng React để thực thi chương trình một khi thỏa điều kiện đã thiết lập. Ngoài ra MATLAB Analysis App còn cung cấp các mẫu sẵn có giúp chúng ta dễ dàng thực hiện các hành động như: tính toán giá trị trung bình của 1 field trong channel; chuyển đổi đơn vị (ví dụ như chuyển đổi từ Celsius sang Fahrenheit); thay thế giá trị null, loại bỏ giá trị ngoại lai,...Ngoài ra ThingSpeak còn hỗ trợ các hàm của MATLAB và toolboxes trong Access MATLAB

Add-On Toolboxes, chúng ta cũng có thể viết dữ liệu vào 1 channel hoặc tạo 1 giao diện bằng cách dùng MATLAB Visualizations App.

Tuy nhiên MATLAB Analysis cũng có những giới hạn như:

- Ứng dụng Phân tích MATLAB không hỗ trợ tạo figure, vì vậy bạn không thể tạo bất kỳ biểu đồ nào bằng ứng dụng này.
- Dữ liệu được ghi vào các channel ThingSpeak bằng MATLAB Analysis phải tuân thủ các giới hạn về tốc độ cập nhập thêm dữ liệu. Chúng ta có thể mất dữ liệu nếu cố gắng ghi vào một channel nhiều lần trong khoảng thời gian được cho phép.
- Đối với các tài khoản có giấy phép miễn phí, việc không hoạt động kéo dài sẽ vô hiệu hóa việc thực thi MATLAB Analysis. Đối với các ứng dụng MATLAB Analysis được liên kết với TimeControl và React, chúng ta cần đăng nhập vào ThingSpeak ít nhất 60 ngày một lần để đảm bảo chương trình tiếp tục thực thi mà không bị gián đoạn.

Chương 3. CƠ SỞ LÝ THUYẾT

1. Mô hình triển khai

ESP 8266 Wemos D1 R2: Đây là một vi mạch phát triển dựa trên nền tảng ESP8266, cung cấp khả năng kết nối Wi-Fi và xử lý dữ liệu. Nó được sử dụng làm bộ điều khiển trung tâm trong hệ thống cảnh báo cháy rừng.

Cảm biến nhiệt độ: Sử dụng một cảm biến nhiệt độ để đo và ghi nhận nhiệt độ hiện tại trong khu vực giám sát. Cảm biến này gửi tín hiệu cho Wemos D1 R2 khi nhiệt độ vượt quá ngưỡng đã được định trước.

Hệ thống phun nước: Wemos D1 R2 có khả năng kích hoạt và điều khiển các van và bơm nước để bắt đầu quá trình phun nước và dập tắt đám cháy. Hệ thống phun nước có thể lấy nước từ các nguồn nước xung quanh như hệ thống nước của các hộ dân, bờ hồ, sông, hoặc các nguồn nước khác.

Giao tiếp mạng: Framework sử dụng kết nối Wi-Fi của Wemos D1 R2 để gửi thông báo cảnh báo đến các bên liên quan. Thông báo có thể được gửi đến người quản lý hoặc cơ quan chức năng để thông báo về tình trạng cháy rừng và yêu cầu hỗ trợ.

Giao diện người dùng: Framework có thể bao gồm giao diện người dùng đơn giản để hiển thị thông tin về nhiệt độ, trạng thái hoạt động của hệ thống và cung cấp các tùy chọn cấu hình.

2. Quy trình hoạt động

Trong hệ thống cảnh báo cháy rừng sử dụng ESP8266 và Wemos D1 R2, chúng ta có thể sử dụng một cảm biến nhiệt độ để phát hiện sự xuất hiện của nhiệt độ bất thường, đó là một tín hiệu cho thấy có ngọn lửa trong khu vực giám sát. Cảm biến này sẽ gửi tín hiệu đến ESP8266 khi nhiệt độ vượt quá ngưỡng đã được thiết lập trước, đánh dấu nguy cơ có thể làm bùng phát cháy rừng. Bằng việc sử dụng cảm biến nhiệt độ và ESP8266, ta có thể xây dựng một hệ thống giám sát cháy rừng hiệu quả và linh hoạt.

Cảm biến nhiệt độ được đặt trong khu vực cần giám sát và liên tục đo và ghi nhận nhiệt độ hiện tại. Bằng cách so sánh nhiệt độ hiện tại với ngưỡng nhiệt độ đã được thiết lập trước, cảm biến sẽ xác định xem có sự tăng đột ngột và vượt quá ngưỡng đó hay không. Khi nhiệt độ vượt quá ngưỡng, cảm biến sẽ gửi tín hiệu đến ESP8266.

ESP8266, là bộ điều khiển trung tâm trong hệ thống, sẽ nhận tín hiệu từ cảm biến nhiệt độ và xử lý nó. Khi nhận được tín hiệu cảnh báo từ cảm biến, ESP8266 có thể xử lý một loạt các thao tác gồm thực hiện các hành động cần thiết, chẳng hạn như gửi thông báo cảnh báo cho các bên liên quan như người quản lý hoặc cơ quan chức năng

thông qua giao thức Wi-Fi hoặc mạng di động, kích hoạt hệ thống phun nước, lưu trữ dữ liệu liên quan đến cảnh báo cháy.

Khi ESP8266 nhận được tín hiệu cảnh báo từ cảm biến nhiệt độ, nó có thể kích hoạt hệ thống phun nước, điều khiển các van và bơm nước để bắt đầu quá trình phun nước và dập tắt đám cháy. Điều này đảm bảo rằng hệ thống phun nước được kích hoạt ngay lập tức để kiểm soát và ngăn chặn sự lan rộng của cháy rừng.

ESP8266 cũng có thể gửi tín hiệu báo động đến các hộ dân xung quanh hoặc trạm cứu hỏa gần nhất thông qua kết nối mạng, cung cấp thông tin về vị trí và tình trạng cháy rừng. Điều này giúp cơ quan chức năng và nhân viên cứu hỏa nắm bắt thông tin kịp thời và có thể đáp ứng nhanh chóng để kiểm soát tình hình cháy rừng.

Ngoài ra, ESP8266 còn có khả năng lưu trữ dữ liệu liên quan đến cảnh báo cháy, các thông tin quan trọng như thời gian xảy ra cảnh báo, vị trí cảnh báo và mức độ cháy có thể được ghi lại và lưu trữ trên ESP8266.

Việc lưu trữ dữ liệu này cung cấp một nguồn thông tin quan trọng cho quá trình giám sát và phân tích sau này. Nhờ vào khả năng lưu trữ, các chuyên gia có thể xem xét các mẫu dữ liệu cảnh báo cháy trước đó để phân tích xu hướng, xác định các khu vực có nguy cơ cháy cao, và đưa ra các biện pháp phòng ngừa hiệu quả hơn trong tương lai.

3. Thư viện được sử dụng trong Arduino IDE

3.1 ESP8266WiFi.h

Trong dự án cảnh báo cháy rừng, thư viện "ESP8266WiFi.h" có chức năng hỗ trợ kết nối và giao tiếp với mạng WiFi sử dụng mô-đun ESP8266 trên nền tảng Arduino hoặc ESP8266. Một số chức năng chính của thư viện này bao gồm: Kết nối và ngắt kết nối Wi-Fi, Gửi và nhận dữ liệu qua giao thức TCP/IP, Lấy địa chỉ IP của module, Điều khiển chế độ hoạt động của module.

3.2 DHT.h

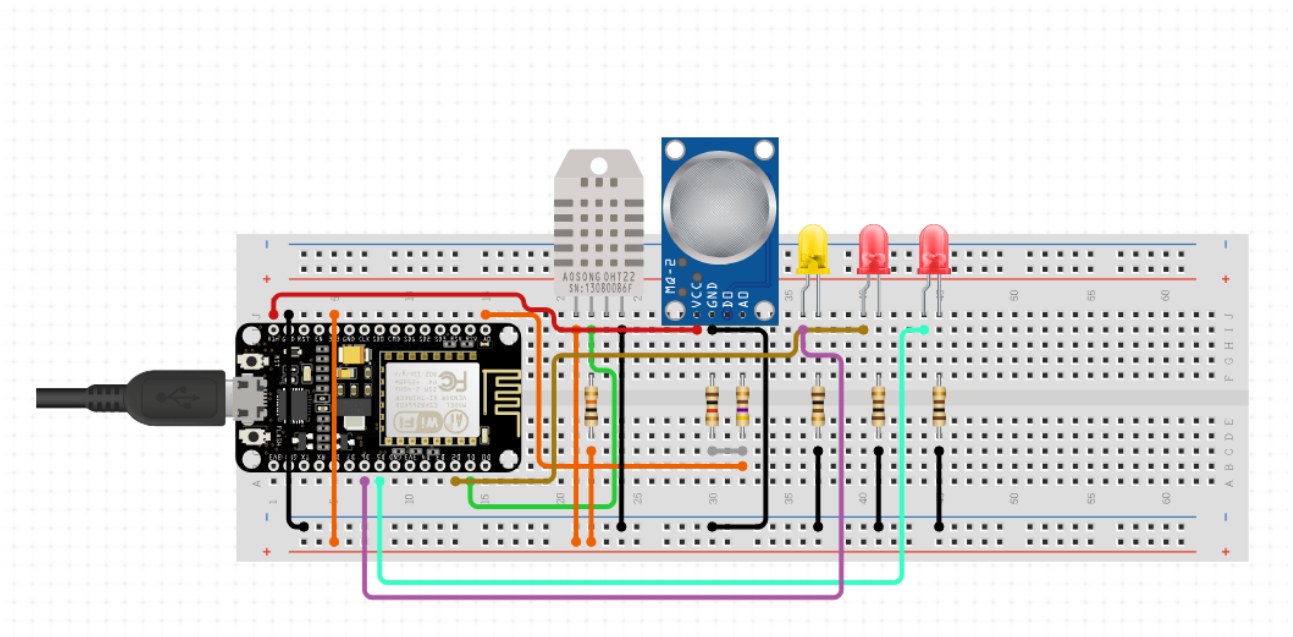
Thư viện "DHT.h" trong dự án này có chức năng hỗ trợ đọc dữ liệu từ cảm biến nhiệt độ và độ ẩm DHT, thông qua vi điều khiển ESP8266. Chức năng chính của thư viện này gồm: Đọc dữ liệu nhiệt độ và độ ẩm, Xử lý lỗi và kiểm tra tính hợp lệ của dữ liệu, Điều khiển và cài đặt cảm biến.

3.3 ArduinoJson.h

Thư viện "ArduinoJson.h" trong dự này có chức năng hỗ trợ xử lý và tạo chuỗi JSON trên nền tảng Arduino hoặc ESP8266. Chức năng chính của thư viện này gồm: Phân tích và truy cập dữ liệu JSON, Tạo chuỗi JSON, Kiểm tra tính hợp lệ của JSON, Nhận và gửi dữ liệu JSON qua giao thức mạng.

Chương 4. KỊCH BẢN TRIỂN KHAI

1. Mô hình thiết kế

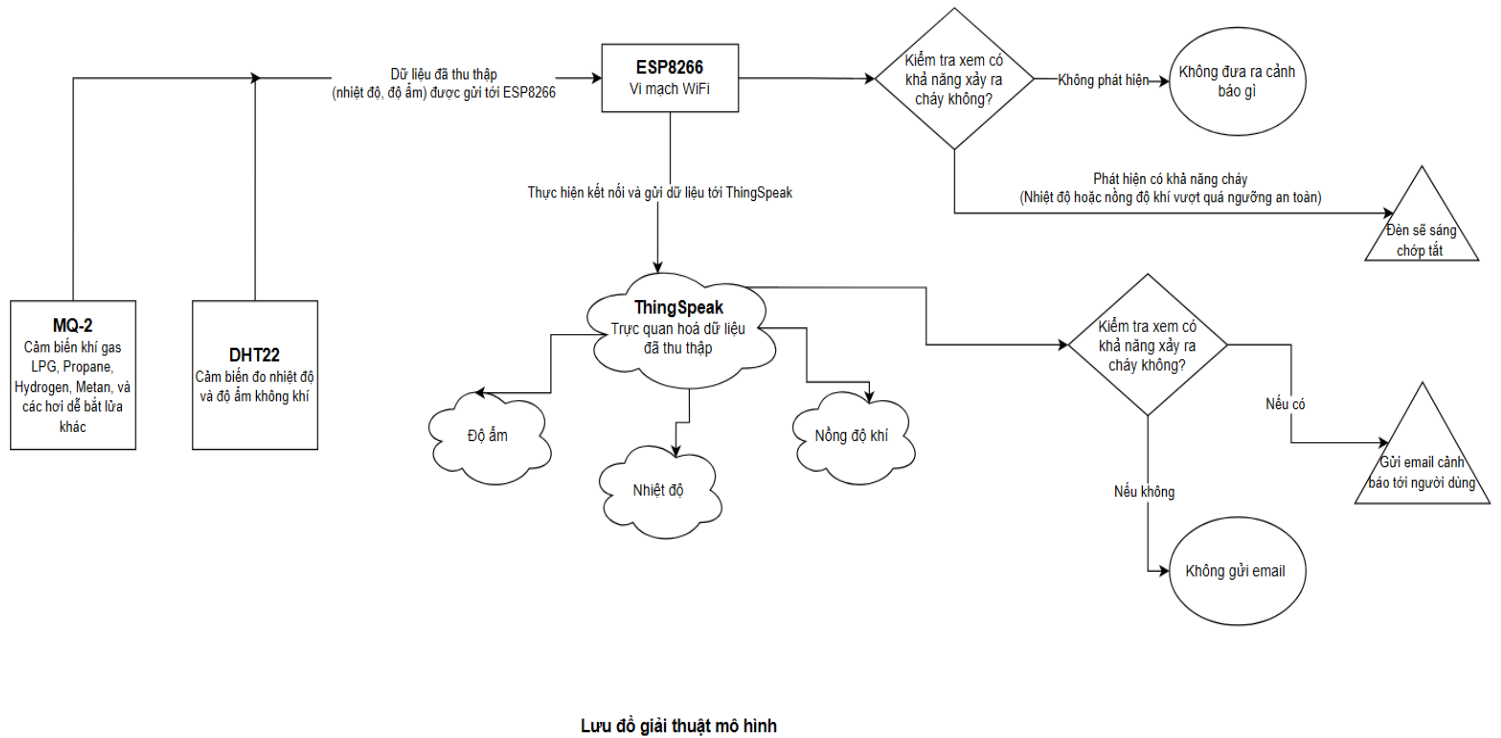


Hình 4. 1 Mô hình thiết kế hệ thống cảnh báo cháy

Các thiết bị sử dụng:

- +1 Kit Arduino Wifi ESP8266 NodeMCU Lua WeMos D1 R2
- + 1 Cảm biến độ ẩm, nhiệt độ DHT22
- +1 Cảm Biến Khí Gas (LPG/CO/CH4) MQ-2
- + 3 đèn led

2. Quy trình hoạt động



Hình 4. 2 Quy trình hoạt động của mô hình

Quy trình cụ thể của mô hình thiết kế:

Đầu tiên DHT22 sẽ thu thập các dữ liệu về nhiệt độ và độ ẩm của không khí xung quanh. Các dữ liệu này sẽ được gửi tới ESP8266 để in ra Serial và chuyển tiếp tới ThingSpeak. Riêng giá trị nhiệt độ sẽ được ESP8266 dùng để so sánh với 1 ngưỡng an toàn để kiểm tra xem có khả năng xảy ra hỏa hoạn không. Nếu kết quả vượt ngưỡng thì sẽ xuất ra cảnh báo bằng hành động 2 đèn led đỏ sẽ thay phiên nhau chớp tắt. Nếu kết quả ngược lại thì 2 đèn led này sẽ không sáng

Đồng thời, cảm biến MQ-2 được dùng để phát hiện khi có quá nhiều khói/khí gas trong môi trường. Nếu giá trị MQ-2 trả về vượt ngưỡng nồng độ khí cho phép thì đèn led vàng sẽ sáng nhấp nháy

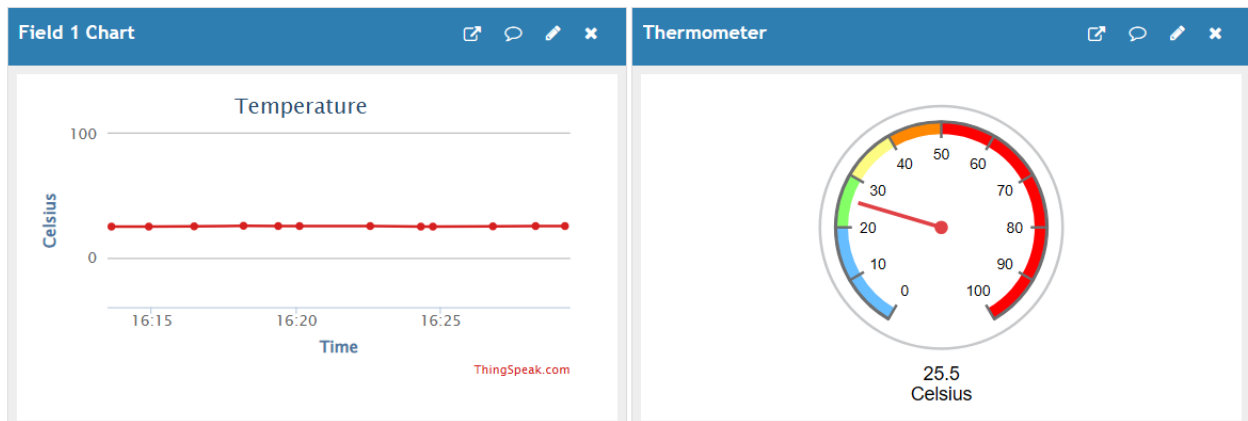
Tại giao diện ThingSpeak, cứ mỗi 15 giây từ khi ESP8266 bắt đầu gửi dữ liệu, các biểu đồ về nhiệt độ, độ ẩm, nồng độ khói sẽ cập nhập lại thông số.

ThingSpeak sẽ kiểm tra và gửi email cảnh báo cho người dùng nếu phát hiện nhiệt độ hoặc độ ẩm vượt quá ngưỡng đã đặt ra mỗi khi có dữ liệu mới được cập nhập trên ThingSpeak. Việc này được thực hiện thông qua chương trình viết bằng ngôn ngữ MATLAB.

3. Trực quan hoá dữ liệu

Sử dụng ThingSpeak để tạo các biểu đồ và widget để hiển thị dữ liệu như sau:

+ Nhiệt độ

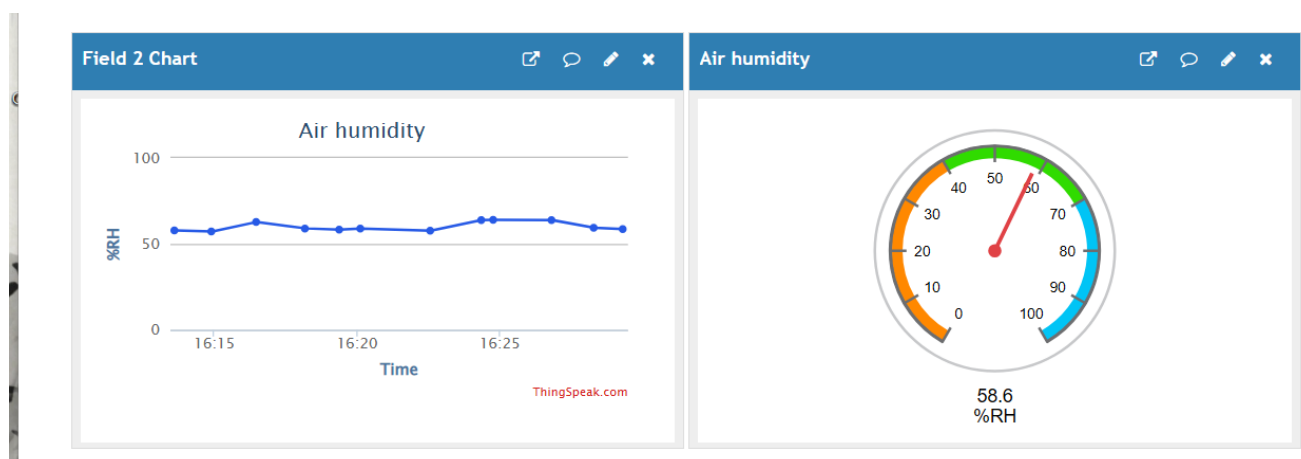


Hình 4.3. 1 Biểu đồ nhiệt độ

Widget “Thermometer” sẽ biểu diễn chính xác nhiệt độ xung quanh dựa trên giá trị cảm biến mà DHT22 đã gửi. Tuy nhiên vì Thingspeak mất tối thiểu 15 giây để cập nhập dữ liệu mới nên sẽ có độ trễ so với thực tế.

Chart “Temperature” là biểu đồ thống kê nhiệt độ tại 20 thời điểm gần nhất. Mỗi một chấm trong biểu đồ tương ứng với giá trị nhiệt độ thu thập được tại 1 thời điểm thực tế.

+ Độ ẩm

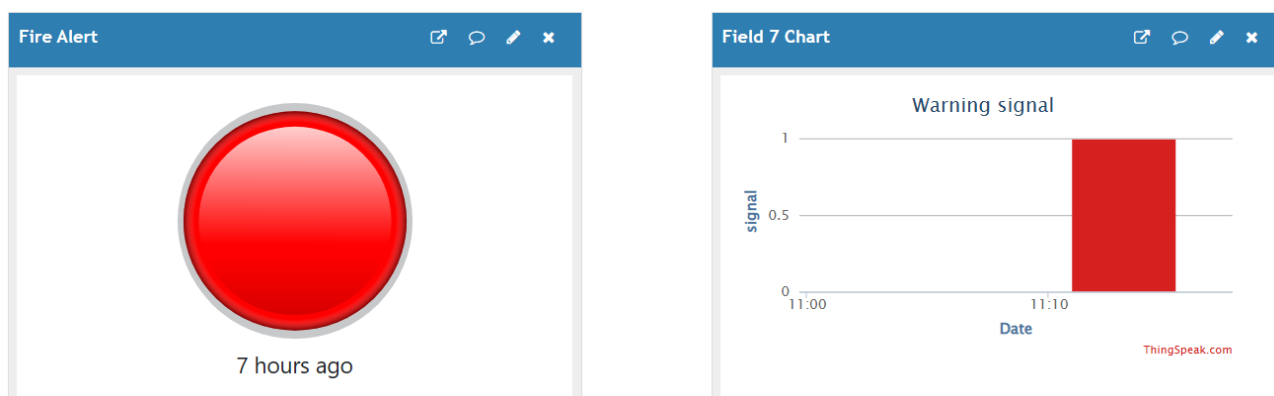


Hình 4.3. 2 Biểu đồ độ ẩm

Widget “Air humidity” sẽ biểu diễn chính xác độ ẩm xung quanh dựa trên giá trị cảm biến mà DHT22 gửi. Tuy nhiên Thingspeak mất tối thiểu 15 giây để cập nhập dữ liệu mới nên giá trị độ ẩm thấy được là giá trị thực tế của 15 giây trước đó.

Chart “Air humidity” là cũng biểu đồ thông kê độ ẩm tại 20 thời điểm gần nhất. Mỗi một chấm trong biểu đồ tương ứng với giá trị độ thu thập được tại 1 thời điểm thực tế. Cứ 15 giây biểu đồ sẽ cập nhập 1 lần nếu có thêm dữ liệu mới.

+Cảnh báo

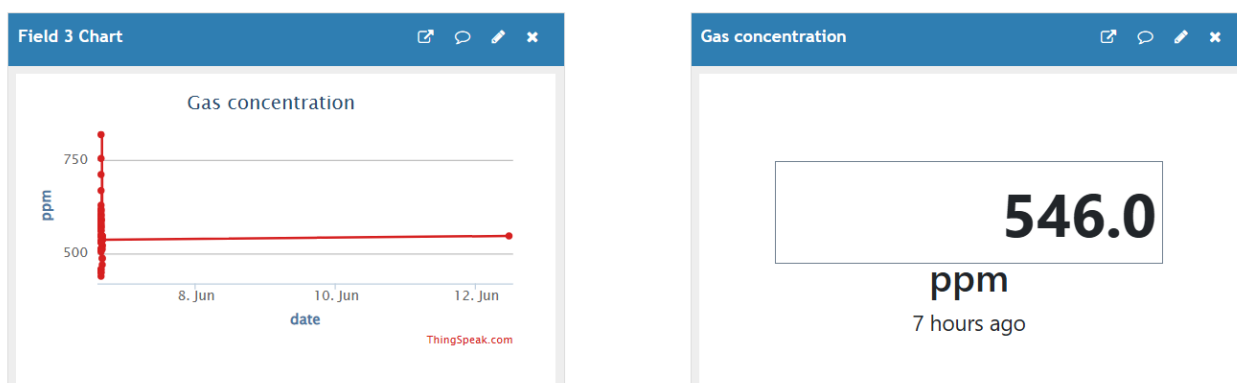


Hình 4.3. 3 Theo dõi trạng thái cảnh báo cháy

Widget “Fire alert” sẽ bật màu đỏ (thể hiện sự nguy hiểm) ngay khi ThingSpeak nhận được giá trị nhiệt độ vượt quá ngưỡng an toàn mà nhóm đã chọn. Nếu nhiệt độ dưới ngưỡng thì sẽ bật màu xanh lá, thể hiện sự an toàn.

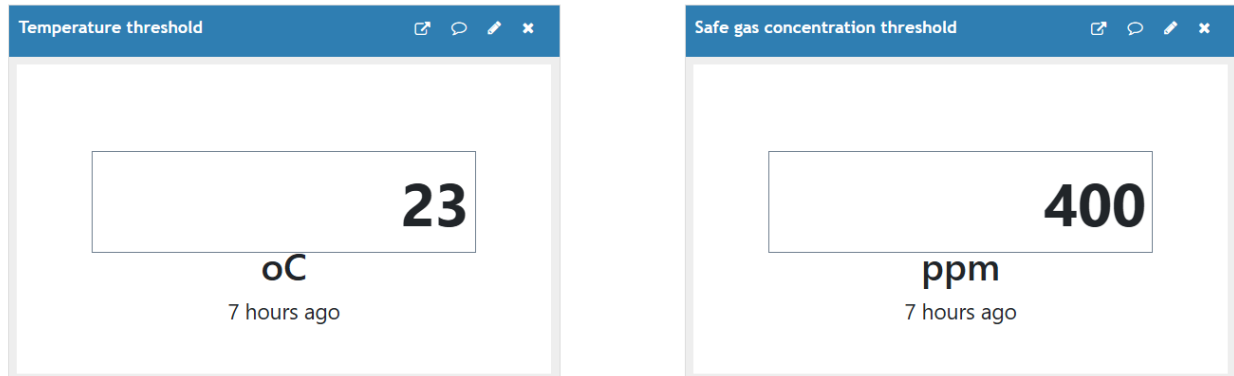
Biểu đồ Warning signal sẽ lưu trữ các giá trị của signal, signal=1 đại diện cho việc phát hiện nguy cơ cháy, signal=0 thì bình thường

+Nồng độ khói



Hình 4.3. 4 Theo dõi nồng độ khí gas

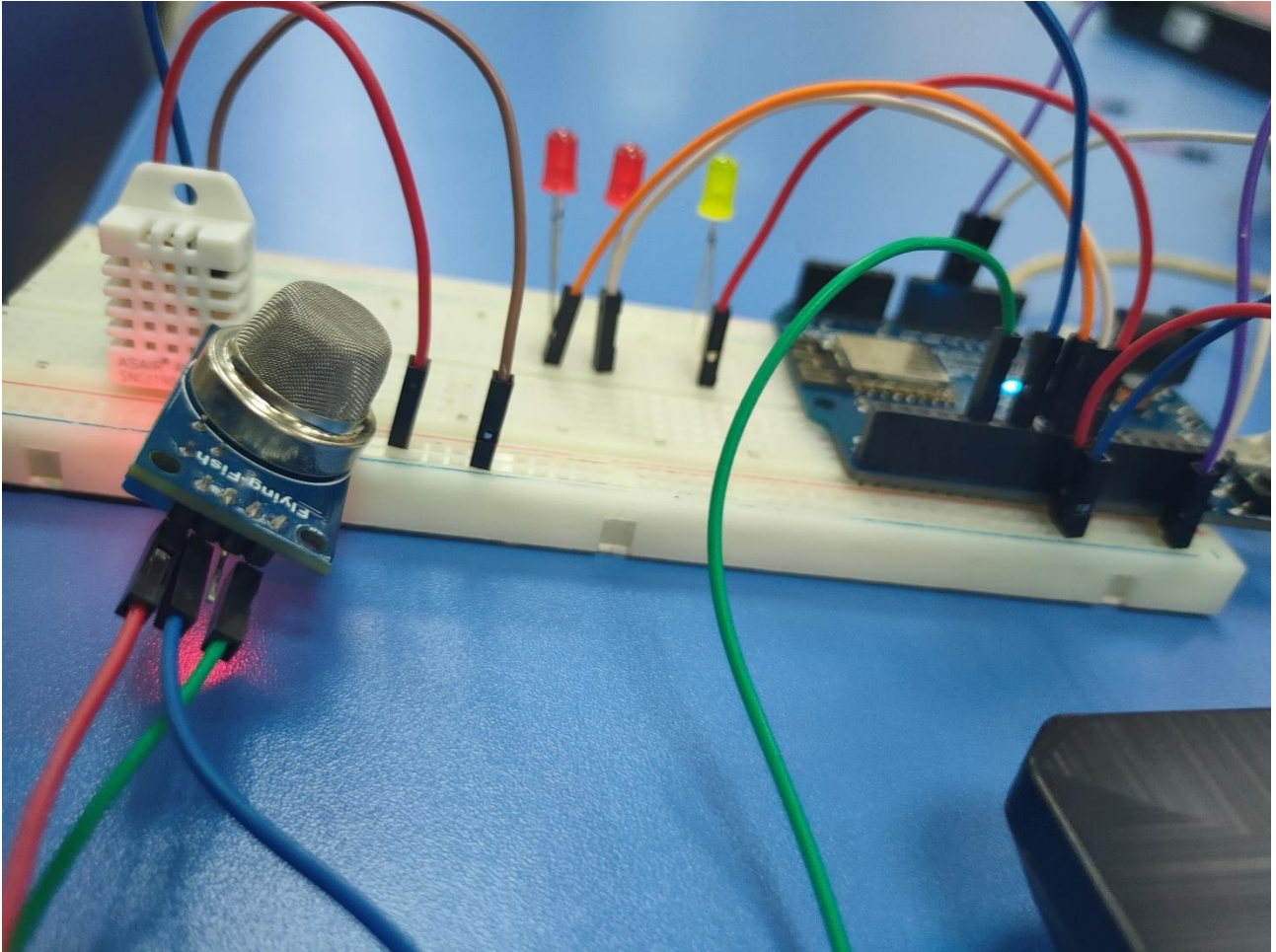
Widget “Gas concentration” sẽ biểu diễn nồng độ khí gas mà cảm biến MQ-2 cảm nhận được.
+Ngưỡng giá trị được thiết lập



Hình 4.3. 5 Ngưỡng nhiệt độ và ngưỡng nồng độ khí

Chương 5: KẾT QUẢ THỰC NGHIỆM

1. Hiện thực phần cứng



Hình 5. 1: Mạch thiết kế hoàn thiện

2. Hiện thực chương trình

Thực hiện lấy dữ liệu (nhiệt độ, độ ẩm) từ cảm biến DHT22

```
118 //Vì DHT22 gửi tín hiệu sau mỗi 2s nên chúng ta phải chờ 2s mỗi vòng lặp
119 |   delay(2000);
120 //Đọc giá trị nhiệt độ từ sensor
121 |   temperature =dht22_sensor.readTemperature();
122 //Đọc giá trị độ ẩm từ sensor
123 |   humidity =dht22_sensor.readHumidity();
124
125 //Kiểm tra xem có đọc được giá trị từ sensor không. Nếu nhận được NULL thì sẽ thông báo
126 |   if (isnan(temperature) || isnan(humidity) )
127 |   {
128 |       Serial.println(F("Failed to read from DHT sensor!"));
129 |       return;
130 |   }
131 . //In ra nhiệt độ trên Serial monitor
132 |   Serial.print(" Temperature: " );
133 |   Serial.print(temperature );
134 |   Serial.println(" °C");
135 //In ra độ ẩm trên Serial monitor
136 |   Serial.print(" Humidity: " );
137 |   Serial.print(humidity );
138 |   Serial.println(" %RH");
139
```

Kiểm tra nhiệt độ môi trường và đưa ra cảnh báo nếu nhiệt độ vượt quá ngưỡng cho phép

```
146 //Kiểm tra nhiệt độ. Nếu quá ngưỡng thì sẽ bật đèn cảnh báo
147 |   if( temperature <= threshold)
148 |   {
149 |       Serial.println("Threshold value at normal");
150 |       Serial.println("=>No flame detected. ");
151 |       digitalWrite(ledPin1, 0);
152 |       digitalWrite(ledPin2, 0);
153 |   }
154 |   else
155 |   {
156 |
157 |       Serial.println("=> Flame detected ! ");
158 |       // ThingSpeak cần ít nhất 15 giây để cập nhập dữ liệu vô biểu đồ nên 2 đèn sẽ liên tục chớp tắt 15 giây
159 |       for(int i=0;i<15;i++)
160 |       {
161 |           digitalWrite(ledPin1, 1);
162 |           digitalWrite(ledPin2, 0);
163 |           delay(500);
164 |
165 |           digitalWrite(ledPin1, 0);
166 |           digitalWrite(ledPin2, 1);
167 |           delay(500);
168 |       }
169 |   }
170
171 }
```

Thực hiện lấy dữ liệu nồng độ khí từ cảm biến MQ-2

```
139 |  
140 | gasValue=analogRead(gasPin);  
141 | Serial.println(gasValue);  
... |
```

Kiểm tra nồng độ khí và đưa ra cảnh báo nếu nồng độ vượt quá giới hạn cho phép

```
... |  
174 |     if(gasValue > limit)  
175 |     {  
176 |         digitalWrite(ledPin3,1);  
177 |         delay(500);  
178 |         digitalWrite(ledPin3,0);  
179 |     }  
180 |     else{  
181 |         digitalWrite(ledPin3,0);  
182 |     }
```

Kiểm tra xem cò khả năng cháy không

```
|  
| if((temperature <= threshold) && (gasValue <= limit))  
| {  
|     signal=0;  
| }  
| else {  
|     signal=1;  
| }
```

Hàm giao tiếp với ThingSpeak

```
171 | //Gửi dữ liệu nhận được từ DHT22 tới Thingspeak  
172 | thingSpeak(temperature, humidity, gasValue, threshold, signal, limit);  
173 |
```

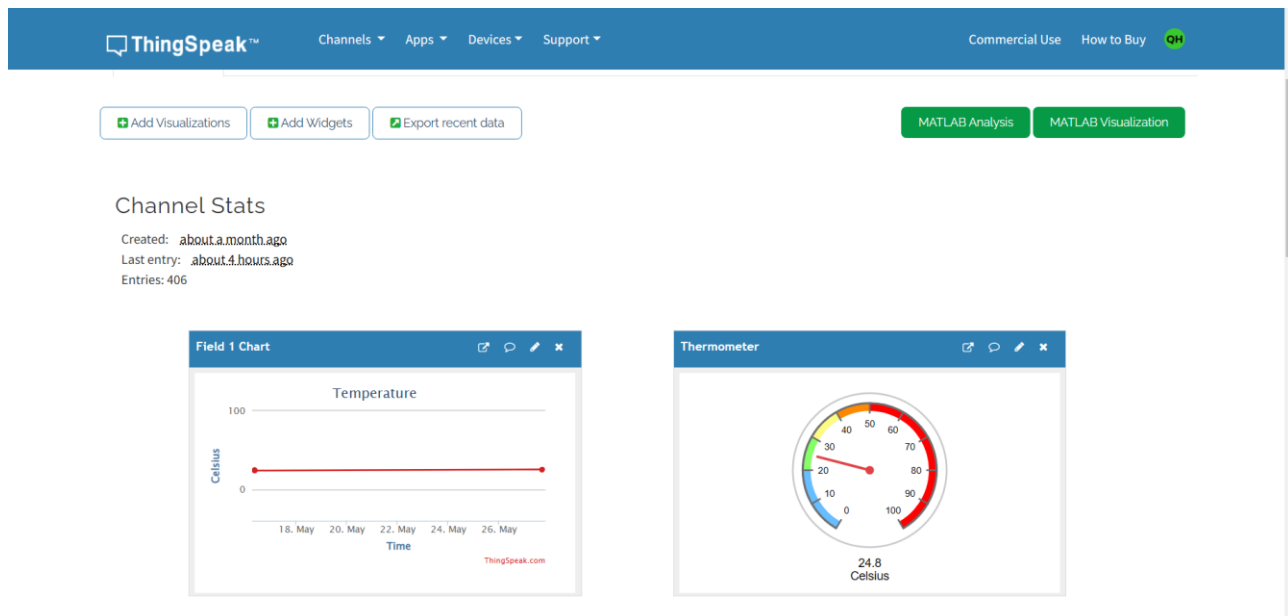
Sau khi chúng ta tạo 1 kết nối tới channel ThingSpeak, chúng ta sẽ sử dụng HTTP POST để gửi dữ liệu tới ThingSpeak và cập nhật chúng cho các widget

```

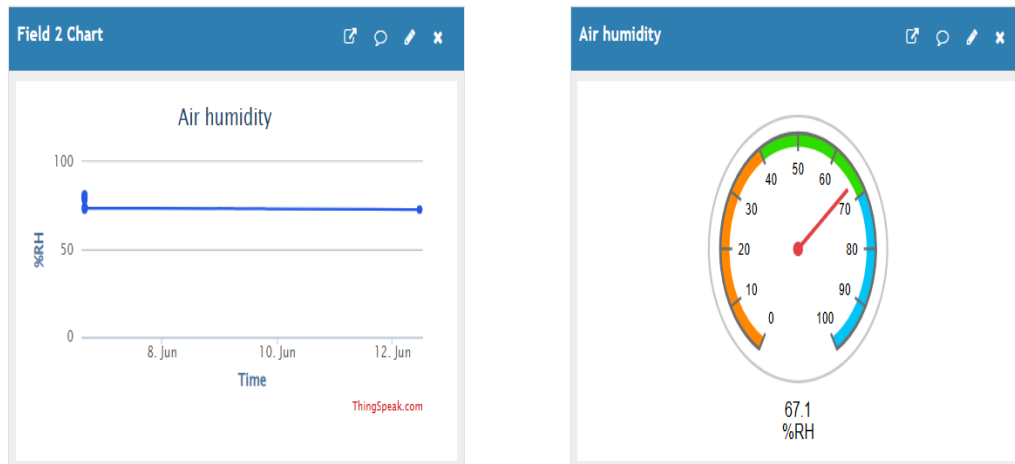
87 | void thingSpeak(float temp, float hum, float gas, int threshold, bool signal, int limit)
88 | {
89 |     String postStr = apiKey;
90 > //Tạo chuỗi chứa nội dung gửi tới Thing Speak...
109 //Thực hiện kết nối tới ThingSpeak và gửi các thông tin cần thiết tới ThingSpeak
110 |     client.print("POST /update HTTP/1.1\n");
111 |     client.print("Host: api.thingspeak.com\n");
112 |     client.print("Connection: close\n");
113 |     client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
114 |     client.print("Content-Type: application/x-www-form-urlencoded\n");
115 |     client.print("Content-Length: ");
116 |     client.print(postStr.length());
117 |     client.print("\n\n");
118 |     client.print(postStr);
119 |
120 |

```

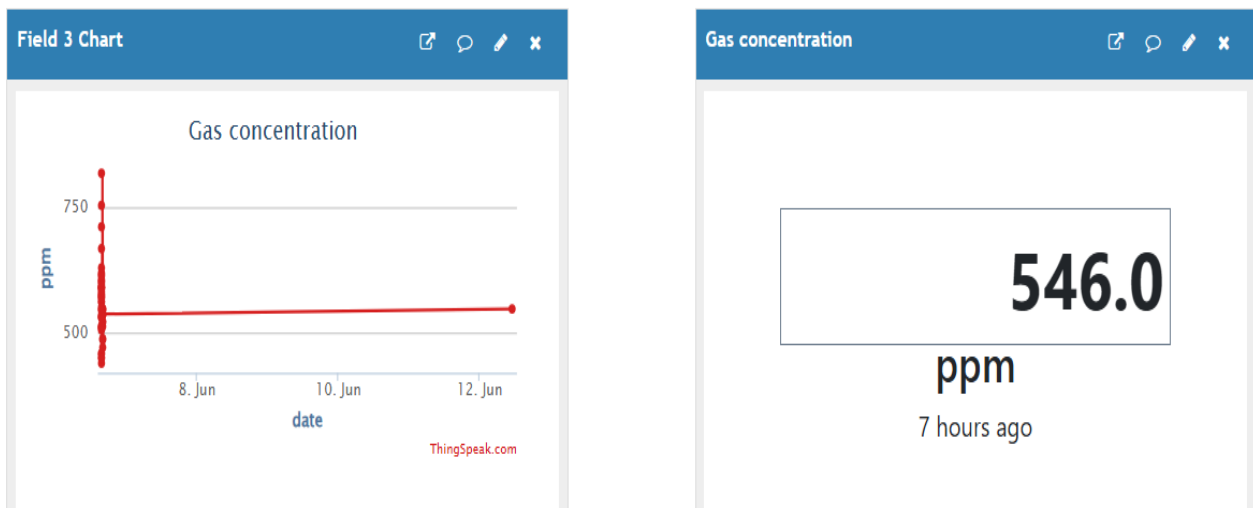
3. Giao diện người dùng



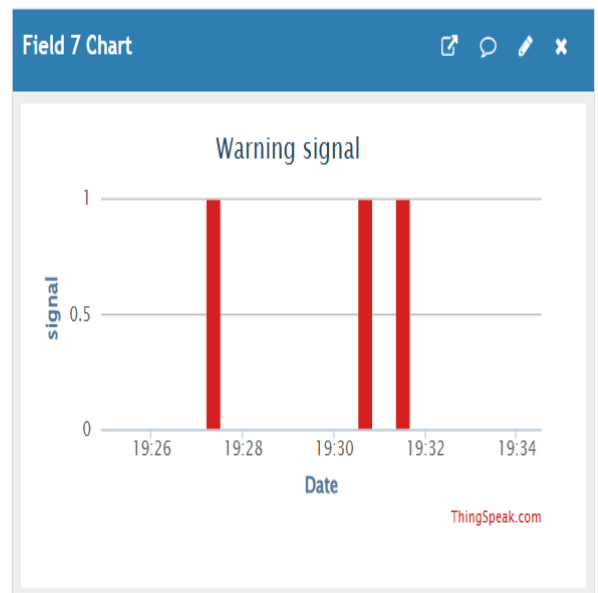
Hình 5.3 1 *Hiển thị thông tin nhiệt độ*



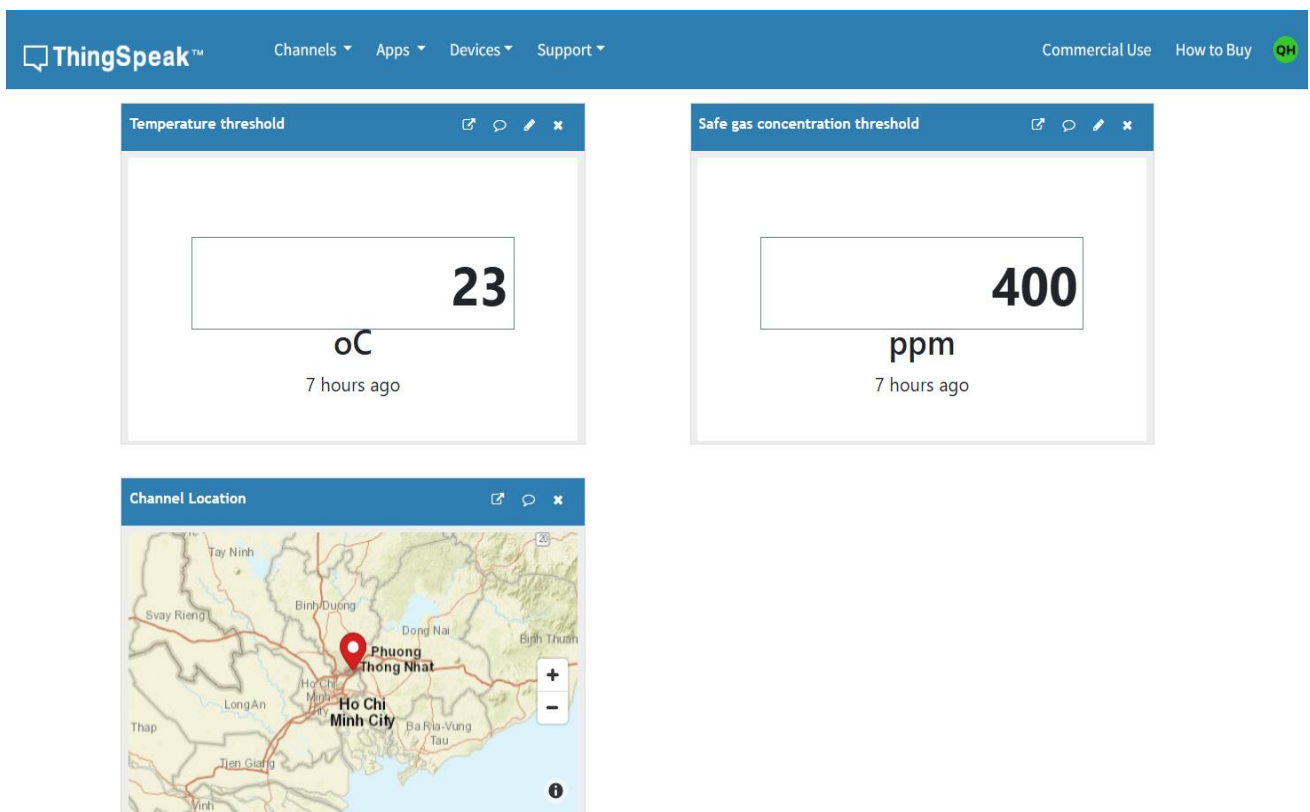
Hình 5.3 2 Hiển thị thông tin độ ẩm



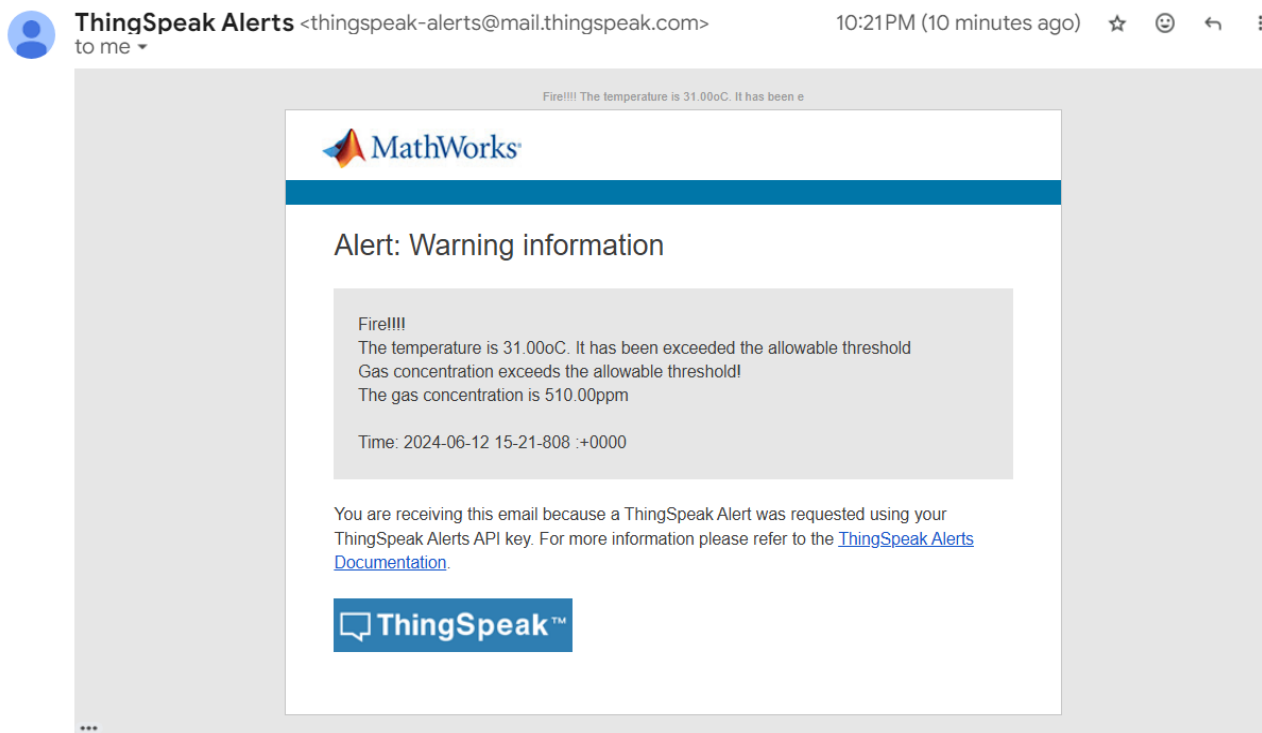
Hình 5.3 3 Hiển thị thông tin khí gas (các loại khí dễ gây cháy)



Hình 5.3 4 Hiển thị tín hiệu cảnh báo



Hình 5.3 5 Hiển thị các thông tin khác



Hình 5.3 6 Email cảnh báo nhận được khi hệ thống phát hiện nhiệt độ hoặc nồng độ khí vượt ngưỡng

4. Mobile application

Ứng dụng mobile được chỉ thiết kế tương thích với hệ điều hành Android

a) Chức năng

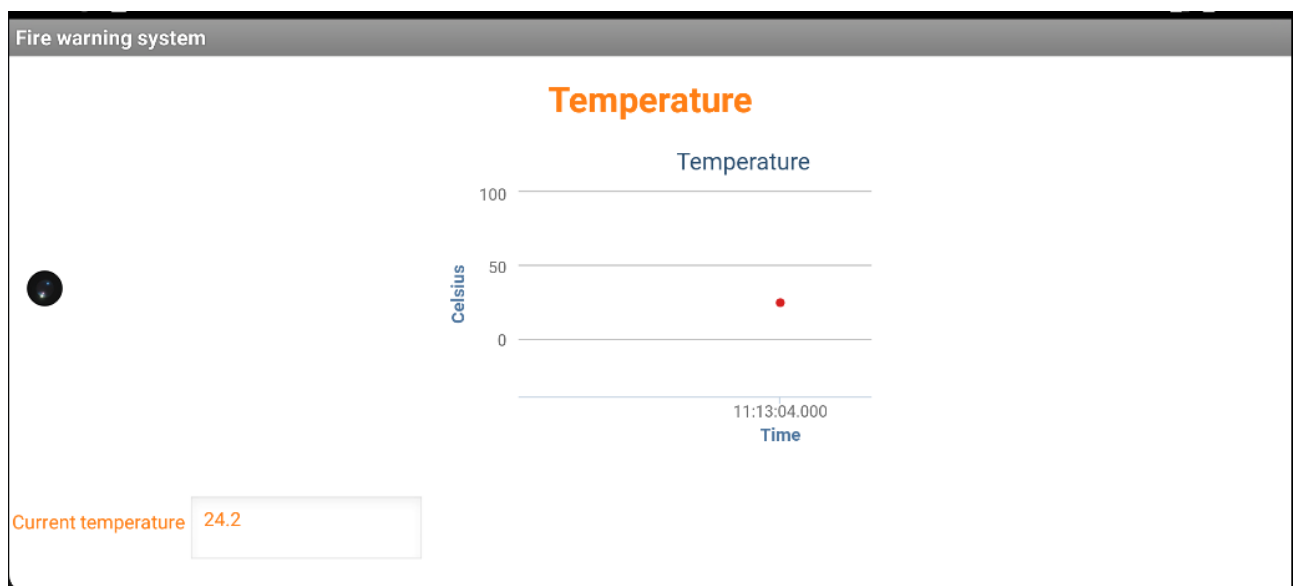
Biểu diễn các giá trị về nhiệt độ, độ ẩm và nồng độ khí gây cháy mà hệ thống cảnh báo cháy cảm nhận được.

Thông báo khi phát hiện có nguy cơ cháy (nhiệt độ cao hơn ngưỡng cho phép hoặc nồng độ khí cao vượt mức an toàn)

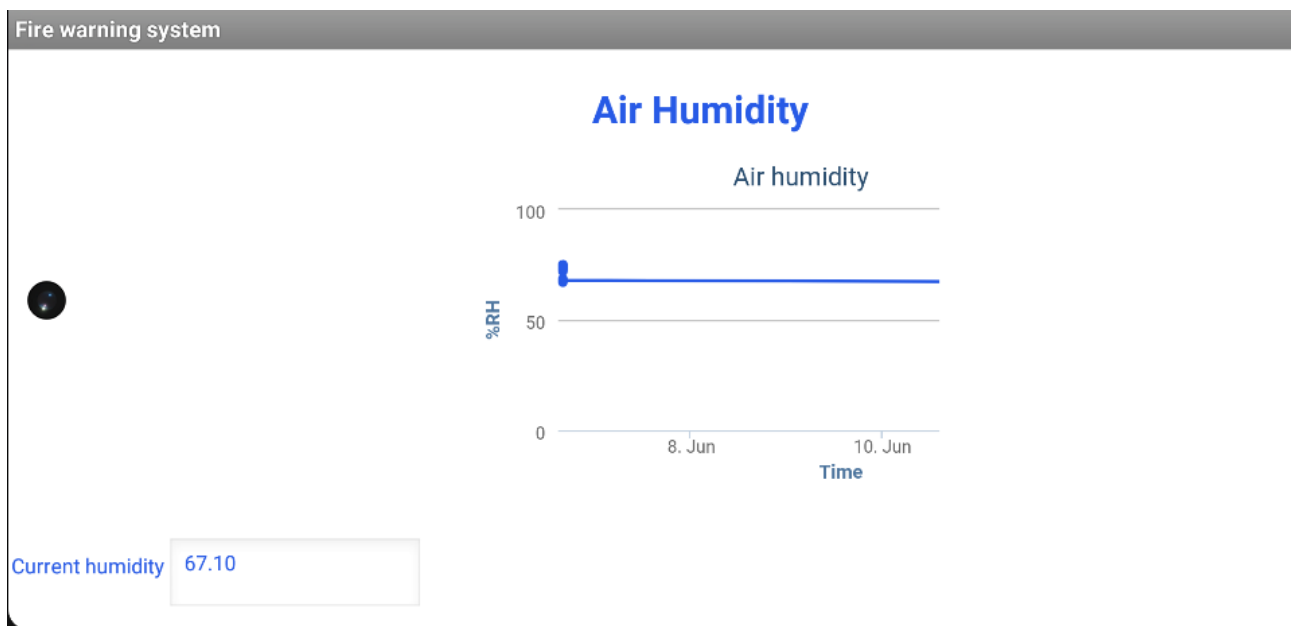
b) Giao diện



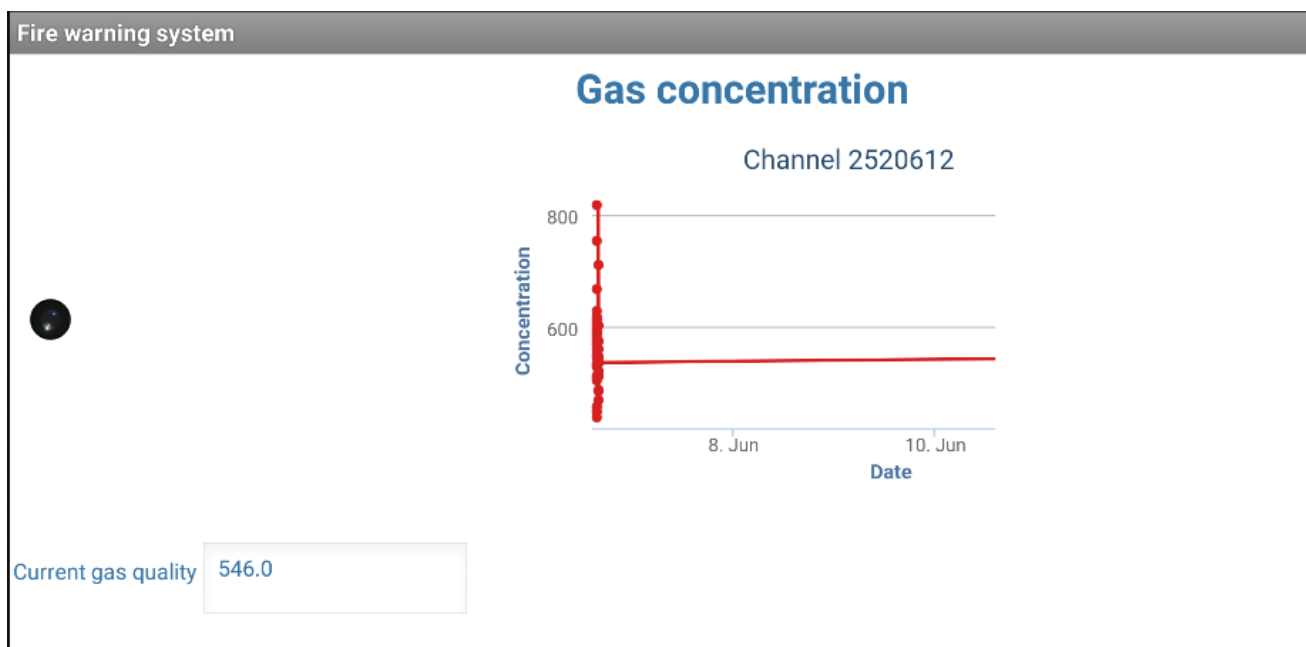
Hình 5.4 1 Giao diện mobile



Hình 5.4 2 Giao diện mobile



Hình 5.4 3 Giao diện mobile



Hình 5.4 4 Giao diện mobile

TÀI LIỆU THAM KHẢO

Trang thông tin chính thức của Arduino:

<https://www.arduino.cc/>

Thông tin về Board ESP8266 Wemos D1 R2: Mục “ESP8266 WeMos D1 Mini Datasheet”:

<https://diyi0t.com/esp8266-wemos-d1-mini-tutorial/>

Thông tin về Cảm biến nhiệt DHT22:

<https://learn.adafruit.com/dht>

Tài liệu về ThingSpeak

<https://www.mathworks.com/help/thingspeak/>

<https://www.mathworks.com/help/thingspeak/MoistureMonitor.html>