

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG



ĐỒ ÁN MÔN HỌC 1

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG

**THIẾT KẾ THIẾT BỊ ĐO NỒNG ĐỘ KHÍ GAS,
NHIỆT ĐỘ, ĐỘ ẨM SỬ DỤNG ESP32 THEO DÕI
DỮ LIỆU QUA FIREBASE HIỂN THỊ TRÊN MÀN
HÌNH TFT**

SVTH: Lê Anh Tuấn

MSSV: 22161206

Thành phố Hồ Chí Minh, tháng 5 năm 2025

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG



ĐỒ ÁN MÔN HỌC 1

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG

**THIẾT KẾ THIẾT BỊ ĐO NỒNG ĐỘ KHÍ GAS,
NHIỆT ĐỘ, ĐỘ ẨM SỬ DỤNG ESP32 THEO DÕI
DỮ LIỆU QUA FIREBASE HIỂN THỊ TRÊN MÀN
HÌNH TFT**

SVTH: Lê Anh Tuấn

MSSV: 22161206

GVHD: Th.S Trương Quang Phúc

Thành phố Hồ Chí Minh, tháng 5 năm 2025

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Thầy Trương Quang Phúc, giảng viên Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh, người đã tận tình hướng dẫn em trong suốt quá trình thực hiện. Đồ án 1 với đề tài: *"Thiết kế thiết bị đo nồng độ khí gas, nhiệt độ, độ ẩm sử dụng ESP32 theo dõi dữ liệu qua Firebase hiển thị trên màn hình TFT."*

Trong suốt quá trình nghiên cứu và hoàn thiện đồ án, Thầy không chỉ truyền đạt cho em những kiến thức chuyên môn quý báu mà còn tận tâm định hướng, góp ý chi tiết, giúp em từng bước hoàn thiện sản phẩm một cách hiệu quả nhất. Em vô cùng trân trọng sự tận tụy, trách nhiệm và nhiệt huyết của Thầy trong từng lời hướng dẫn.

Do còn hạn chế về thời gian, kinh nghiệm và kiến thức thực tế, chắc chắn đồ án vẫn còn nhiều thiếu sót. Em rất mong nhận được những góp ý quý báu từ Thầy để có thể tiếp tục rèn luyện, nâng cao kỹ năng và cải thiện chất lượng trong những nghiên cứu, dự án sau này.

Một lần nữa, em xin gửi lời cảm ơn chân thành đến Thầy!

TP. Hồ Chí Minh, tháng 5 năm 2025

Sinh viên thực hiện đề tài

Lê Anh Tuấn

TÓM TẮT

Ngày nay, cùng với sự phát triển mạnh mẽ của khoa học và công nghệ, các hệ thống nhúng ngày càng được ứng dụng rộng rãi vào nhiều lĩnh vực trong đời sống – từ những ứng dụng đơn giản như điều khiển LED, bật/tắt thiết bị điện tử, đến những hệ thống phức tạp như đèn giao thông, thang máy, cửa tự động, robot, tàu vũ trụ hay điều khiển nhà máy công nghiệp. Nền tảng của các hệ thống tự động hiện đại ngày nay phần lớn đều dựa vào vi điều khiển và hệ thống nhúng, thay thế dần cho các công nghệ cũ như khí nén, thủy lực, rơ-le cơ điện, mạch điện tử số, các thiết bị máy móc tự động thấp so với các hệ thống tự động hiện đại được xây dựng trên nền tảng của các hệ thống nhúng.

Từ những kiến thức đã học trong nhà trường và mong muốn đóng góp cho xã hội, em thực hiện đề tài: *“Thiết kế thiết bị đo nồng độ khí gas, nhiệt độ, độ ẩm sử dụng ESP32, theo dõi dữ liệu qua Firebase hiển thị trên màn hình TFT”*

Mục tiêu của đề tài là thiết kế và thi công một hệ thống có khả năng giám sát môi trường theo thời gian thực, đặc biệt tập trung vào nồng độ khí gas, nhiệt độ và độ ẩm – những yếu tố ảnh hưởng trực tiếp đến an toàn và chất lượng sống. Thiết bị sử dụng cảm biến MQ-2 để phát hiện khí gas, cảm biến DHT11 để đo nhiệt độ và độ ẩm, vi điều khiển ESP32 làm trung tâm xử lý, gửi dữ liệu liên tục lên nền tảng đám mây Firebase và hiển thị lên màn hình TFT để người dùng tiện theo dõi.

Đề tài xuất phát từ thực trạng hiện nay: tình trạng cháy nổ do rò rỉ khí gas, đặc biệt từ các bình gas mini tái sử dụng ngày càng trở nên phổ biến và nguy hiểm. Nhiều hộ gia đình, hàng quán vẫn sử dụng bình gas mini đã bị tái chiết nhiều lần – vốn không được thiết kế để sử dụng lại – dẫn đến nguy cơ rò rỉ, cháy nổ rất cao. Các cơ sở sang chiết thường sử dụng hỗn hợp khí có áp suất cao vượt mức cho phép, làm tăng nguy cơ mất an toàn. Trong khi đó, người sử dụng lại thiếu kiến thức cơ bản về dấu hiệu rò rỉ và cách sử dụng an toàn.

Với đề tài này, em mong muốn xây dựng một thiết bị cảnh báo sớm, hỗ trợ người dùng trong việc phát hiện các điều kiện bất thường, góp phần nâng cao ý thức phòng tránh rủi ro, đảm bảo an toàn cho cá nhân và cộng đồng.

MỤC LỤC

LỜI CẢM ƠN	I
TÓM TẮT	II
MỤC LỤC	III
DANH MỤC HÌNH ẢNH	VIII
DANH MỤC BẢNG	IXX
DANH MỤC CÁC TỪ VIẾT TẮT	X
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI	1
1.1 GIỚI THIỆU	1
1.2 TÌNH HÌNH NGHIÊN CỨU	2
1.2.1 Tình hình nghiên cứu trong nước	2
1.2.2 Tình hình nghiên cứu ngoài nước	2
1.3 MỤC TIÊU CỦA ĐỀ TÀI	3
1.4 PHƯƠNG PHÁP NGHIÊN CỨU	5
1.4.1. Phương pháp thu thập tài liệu	5
1.4.2. Phương pháp phân tích - thiết kế hệ thống	5
1.4.3. Phương pháp thực nghiệm	5
1.4.4. Phương pháp kiểm tra và đánh giá	6
1.5 BỐ CỤC ĐỀ TÀI	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	7
2.1 GIỚI THIỆU VỀ HỆ THỐNG IOT	7
2.1.1 Khái niệm và ứng dụng	7
2.1.2 Cấu trúc của hệ thống IoT	7
2.2. VI ĐIỀU KHIỂN ESP32	9

2.3 GIAO TIẾP VỚI FIREBASE REALTIME DATABASE	11
2.3.1 Giới thiệu	11
2.3.2 Cấu trúc Firebase	12
2.3.3 Giao tiếp giữa ESP32 và Firebase	12
2.4 PHẦN MỀM LẬP TRÌNH VI ĐIỀU KHIỂN	13
2.4.1 Giới thiệu về Arduino IDE	13
2.4.2 Giới thiệu về Visual Studio Code	14
2.5 GIAO DIỆN NGƯỜI DÙNG	14
2.5.1 Giao diện web	14
2.6. CÁC CHUẨN GIAO TIẾP SỬ DỤNG TRONG HỆ THỐNG	15
2.6.1. Giao tiếp SPI: màn hình TFT ILI9225	16
2.6.2. Giao tiếp GPIO: Cảm biến MQ-2, DHT11, Relay, còi, nút nhấn	17
2.6.3. Giao tiếp PWM: Điều khiển Servo	18
2.7. CÁC MODULE PHẦN CỨNG	18
2.7.1. Cảm biến khí gas MQ-2	18
2.7.2. Cảm biến nhiệt độ và độ ẩm DHT11	21
2.7.3. Màn hình TFT ILI9225	23
2.7.4. Còi báo động, servo, quạt	24
2.7.5. Nút nhấn	29
2.7.6. Các thiết bị ngoại vi	32
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG	37
3.1. Yêu cầu hệ thống	37
3.1.1. Chức năng theo dõi dữ liệu cảm biến	37
3.1.2. Chức năng điều khiển thiết bị từ Web Dashboard và nút nhấn	38

3.1.3. Chức năng hiển thị dữ liệu lên màn hình TFT	40
3.2. Đặc tả hệ thống	42
3.2.1. Sơ đồ khối hệ thống	42
3.2.2. Mô tả hoạt động hệ thống	43
3.3. Thiết kế phần cứng	46
3.3.1. Khối xử lý trung tâm (ESP32)	46
3.3.2. Khối ngoại vi	47
3.3.3. Khối giao diện tương tác	47
3.3.4. Khối cảnh báo	49
3.3.5. Khối xử lý và chấp hành	50
3.3.6. Sơ đồ nguyên lý & sơ đồ mạch in	51
3.4. Thiết kế phần mềm	54
3.4.1. Lưu đồ thuật toán chương trình chính	54
3.4.2. Lưu đồ chương trình con	55
3.4.3. Cấu trúc dữ liệu Firebase	58
3.4.4. Giao diện phần mềm Web Dashboard	60
CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG	63
4.1 MÔ HÌNH PHẦN CỨNG THỰC TẾ	63
4.2. KẾT QUẢ GIAO DIỆN WEB	65
4.2.1. Giao diện trang trình	65
4.2.2. Giao diện trang cài đặt	67
4.3 KIỂM TRA VÀ ĐÁNH GIÁ CHỨC NĂNG CHÍNH	69
4.3.1. Đọc và hiển thị dữ liệu cảm biến trên màn hình	69
4.3.2 Gửi dữ liệu lên Firebase	70

4.3.3 Điều khiển thiết bị từ Firebase và nút nhấn	71
4.4. ĐÁNH GIÁ KẾT QUẢ	71
4.4.1. Độ chính xác của cảm biến	71
4.4.2. Độ trễ cập nhật dữ liệu trên Firebase	72
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	74
5.1. KẾT LUẬN	74
5.2. HƯỚNG PHÁT TRIỂN	74
TÀI LIỆU THAM KHẢO	75

DANH MỤC HÌNH ẢNH

Hình 2-1 Cấu trúc hệ thống IoT	8
Hình 2-2 ESP32 (ESP32-WROOM-32)	10
Hình 2-3 Mô hình Firebase Realtime Database	11
Hình 2-4 Cấu trúc dữ liệu của Firebase Realtime Database	12
Hình 2-5 Giao tiếp giữa ESP32 và Firebase Realtime Database	13
Hình 2-6 Mối quan hệ giữa HTML, CSS và JavaScript trong thiết kế web	14
Hình 2-7 Cảm biến khí gas MQ2	19
Hình 2-8 Kết nối chân của ESP với cảm biến MQ2	20
Hình 2-9 Cảm biến nhiệt độ và độ ẩm DHT11	21
Hình 2-10 Kết nối chân của ESP với cảm biến DHT11	22
Hình 2-11 Màn hình TFT	24
Hình 2-12 Sơ đồ kết nối esp với màn hình TFT	24
Hình 2-13 Còi chirp 5V	25
Hình 2-14 Sơ đồ kết nối giữa esp với khối cảnh báo	26
Hình 2-15 Servo SG90 180 độ	27
Hình 2-16 Sơ đồ kết nối esp với Servo SG90	27
Hình 2-17 Quạt 12V	28
Hình 2-18 Kết nối chân giữa ESP32 và quạt 12V	29
Hình 2-19 Nút nhấn	30
Hình 2-20 Kết nối chân của ESP với khối nút nhấn	31
Hình 2-21 C1815 NPN Transistor	32
Hình 2-22 PC817	34
Hình 2-23 IC7805	35
Hình 2-24 Adapter 12V2A	36
Hình 3-1 Sơ đồ khối của hệ thống	42
Hình 3-2 Sơ đồ nguyên lý khối xử lý trung tâm	46

Hình 3-3 Sơ đồ kết nối của esp với khối ngoại vi	47
Hình 3-4 Sơ đồ kết nối esp với khối giao diện tương tác	48
Hình 3-5 Sơ đồ kết nối giữa esp với khối cảnh báo	49
Hình 3-6 Sơ đồ kết nối esp với khối xử lý và chấp hành	50
Hình 3-7 Sơ đồ nguyên lý toàn hệ thống	52
Hình 3-8 Board mạch PCB hoàn thiện	53
Hình 3-9 Hình 3d mạch PCB hoàn thiện	53
Hình 3-10 Lưu đồ chương trình chính	54
Hình 3-11 Lưu đồ chương trình đọc khí gas	55
Hình 3-12 Lưu đồ chương trình đọc nhiệt độ, độ ẩm	56
Hình 3-13 Lưu đồ chương trình Web Dashboard	57
Hình 3-14 Cây dữ liệu Firebase	58
Hình 3-15 Giao diện Web Dashboard	60
Hình 4-1 Board sau khi hàn và kết nối linh kiện	63
Hình 4-2 Mô hình hệ thống	64
Hình 4-3 Giao diện web hiển thị	65
Hình 4-4 Bảng theo dõi dữ liệu	65
Hình 4-5 Bảng trạng thái các thiết bị khi ở chế độ manual	66
Hình 4-6 Bảng trạng thái các thiết bị khi ở chế độ auto	66
Hình 4-7 Giao diện web cài đặt	67
Hình 4-8 Kết quả hiển thị trên màn hình TFT	69
Hình 4-9 Giao diện firebase	70

DANH MỤC BẢNG

BẢNG 2-1 Thông số kỹ thuật ESP32 (ESP32-WROOM-32)	10
BẢNG 2-2 Thông số kỹ thuật của MQ2	20
BẢNG 2-3 Kết nối chân của ESP với cảm biến MQ2	20
BẢNG 2-4 Thông số kỹ thuật cảm biến DHT11	22
BẢNG 2-5 Kết nối chân của ESP với cảm biến DHT11	22
BẢNG 2-6 Thông số kỹ thuật của màn hình TFT	23
BẢNG 2-7 Sơ đồ kết nối chân giữa ESP32 và TFT	24
BẢNG 2-8 Thông số kỹ thuật còi	25
BẢNG 2-9 Kết nối chân của ESP với còi	26
BẢNG 2-10 Thông số kỹ thuật của động cơ Servo	27
BẢNG 2-11 Sơ đồ nối giữa esp và Servo SG90 180 độ	27
BẢNG 2-12 Thông số kỹ thuật của quạt 12V	29
BẢNG 2-13 Kết nối chân giữa ESP32 và quạt 12V	29
BẢNG 2-14 Thông số kỹ thuật của nút nhấn dán 6x6x10mm (4 chân)	31
BẢNG 2-15 Kết nối chân của ESP với các nút nhấn	31
BẢNG 2-16 Thông số kỹ thuật C1815	33
BẢNG 2-17 Thông số kỹ thuật PC817	34
BẢNG 2-18 Thông số kỹ thuật IC7805	35
BẢNG 2-19 Thông số kỹ thuật adapter	36
BẢNG 3-1 Bảng chức năng của 4 nút nhấn	40
BẢNG 4-1 Cài đặt ngưỡng cảnh báo	68
BẢNG 4-2 Chế độ điều khiển	69

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
IoT	Internet of Things – Internet kết nối vạn vật
MCU	Microcontroller Unit – Vi điều khiển
ESP32	Vi điều khiển tích hợp Wi-Fi & Bluetooth
TFT	Thin Film Transistor (loại màn hình LCD sử dụng transistor màng mỏng)
IDE	Integrated Development Environment – Môi trường phát triển tích hợp
HTML	HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản
CSS	Cascading Style Sheets – Ngôn ngữ định kiểu cho HTML
JS	JavaScript – Ngôn ngữ lập trình cho web
PWM	Pulse Width Modulation – Điều chế độ rộng xung
UART	Universal Asynchronous Receiver/Transmitter – Giao tiếp nối tiếp không đồng bộ
SPI	Serial Peripheral Interface – Giao tiếp nối tiếp ngoại vi
I2C	Inter-Integrated Circuit – Giao tiếp nối tiếp nội mạch
API	Application Programming Interface – Giao diện lập trình ứng dụng
MQTT	Message Queuing Telemetry Transport – Giao thức truyền thông nhẹ cho IoT
PCB	Printed Circuit Board – Bảng mạch in
JSON	JavaScript Object Notation – Định dạng trao đổi dữ liệu
DHT11	Digital Humidity and Temperature Sensor – Cảm biến nhiệt độ và độ ẩm
MQ2	Gas Sensor Module – Cảm biến khí gas
SRAM 8T	Static Random Access Memory 8 Transistor
ADC	Analog to Digital Converter (Bộ chuyển đổi tương tự sang số)

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1 GIỚI THIỆU

Trong thời đại công nghiệp hóa – hiện đại hóa và phát triển mạnh mẽ của công nghệ thông tin, việc giám sát các yếu tố môi trường như khí gas, nhiệt độ và độ ẩm trở thành một nhu cầu cấp thiết, đặc biệt trong bối cảnh ngày càng có nhiều sự cố liên quan đến rò rỉ khí gas, cháy nổ hay điều kiện thời tiết bất lợi gây ảnh hưởng đến sức khỏe con người và tài sản. Các tai nạn do rò rỉ khí gas không chỉ gây thiệt hại vật chất nghiêm trọng mà còn ảnh hưởng trực tiếp đến tính mạng con người, đặc biệt tại các khu dân cư, nhà hàng, nhà máy sản xuất, phòng thí nghiệm hoặc các khu vực nấu nướng tập trung.

Từ thực tế đó, việc thiết kế và triển khai một thiết bị có khả năng đo lường liên tục các thông số quan trọng như nồng độ khí gas, nhiệt độ và độ ẩm là hết sức cần thiết. Thiết bị này không chỉ giúp phát hiện kịp thời các nguy cơ tiềm ẩn mà còn hỗ trợ người dùng trong việc theo dõi và kiểm soát môi trường một cách chủ động.

Với sự phát triển mạnh mẽ của công nghệ IoT (Internet of Things), việc kết nối các thiết bị phần cứng với nền tảng dữ liệu đám mây như *Firebase* đã mở ra cơ hội lớn cho việc theo dõi và điều khiển thiết bị từ xa. *ESP32*, với ưu điểm là vi điều khiển có tích hợp Wi-Fi và Bluetooth, trở thành lựa chọn lý tưởng trong việc phát triển các ứng dụng giám sát thông minh. Khi kết hợp *ESP32* với các cảm biến đo khí gas (như *MQ-2*), cảm biến nhiệt độ và độ ẩm (như *DHT11* hoặc *DHT22*), người dùng có thể thu thập dữ liệu môi trường một cách chính xác và nhanh chóng.

Ngoài ra, để tăng tính trực quan và tiện lợi cho người sử dụng tại chỗ, việc *hiển thị dữ liệu trực tiếp lên màn hình TFT* là một giải pháp hiệu quả. Màn hình sẽ giúp người dùng dễ dàng quan sát các thông số môi trường và trạng thái của hệ thống ngay tại vị trí lắp đặt, mà không cần phụ thuộc hoàn toàn vào thiết bị di động hoặc mạng internet.

Từ những lý do trên, đề tài “*Thiết kế thiết bị đo nồng độ khí gas, nhiệt độ, độ ẩm sử dụng ESP32, theo dõi dữ liệu qua Firebase, hiển thị trên màn hình*”

TFT” không chỉ có ý nghĩa thực tiễn cao mà còn mang tính ứng dụng rộng rãi trong đời sống hằng ngày, đặc biệt trong việc nâng cao an toàn cháy nổ và kiểm soát điều kiện môi trường. Đề tài này là một minh chứng rõ ràng cho việc ứng dụng công nghệ vào đời sống, góp phần xây dựng những hệ thống thông minh, hiện đại và an toàn hơn cho cộng đồng.

1.2 TÌNH HÌNH NGHIÊN CỨU

1.2.1 Tình hình nghiên cứu trong nước

Trong những năm gần đây, tại Việt Nam đã có nhiều nghiên cứu và ứng dụng liên quan đến hệ thống giám sát khí gas nhằm phục vụ nhu cầu an toàn trong sinh hoạt và sản xuất. Một số đề tài tiêu biểu có thể kể đến như:

❖ **Đề tài tốt nghiệp tại các trường đại học kỹ thuật:** Nhiều sinh viên đã thực hiện các mô hình cảnh báo khí gas sử dụng cảm biến MQ-2 kết hợp Arduino hoặc ESP8266, hiển thị dữ liệu lên LCD hoặc gửi cảnh báo qua tin nhắn SMS, Zalo hoặc qua mạng nội bộ.

❖ **Các sản phẩm thương mại:** Một số công ty và startup đã phát triển hệ thống giám sát khí gas cho nhà ở, trạm xăng, nhà bếp công nghiệp, tuy nhiên các sản phẩm này thường chỉ hiển thị tại chỗ, chưa tích hợp chức năng theo dõi từ xa qua Internet hoặc điều khiển qua ứng dụng.

❖ **Ứng dụng IoT trong giáo dục:** Nhiều trung tâm đào tạo, CLB nghiên cứu khoa học đã tổ chức các dự án nhỏ sử dụng ESP32 và Firebase nhằm giám sát các thông số môi trường, trong đó có khí gas, nhưng chưa phát triển thành hệ thống hoàn chỉnh tích hợp nhiều chức năng.

Tuy vậy, hầu hết các đề tài trong nước vẫn dừng lại ở mức cơ bản, ít có sự tích hợp toàn diện giữa phần cứng (ESP32, cảm biến), phần mềm (Firebase, giao diện web), và khả năng điều khiển từ xa – vốn là điểm mạnh của đề tài này.

1.2.2 Tình hình nghiên cứu ngoài nước

Trên thế giới, các nghiên cứu và ứng dụng giám sát khí gas trong môi trường đã được triển khai sớm và đa dạng hơn, đặc biệt là trong bối cảnh phát triển *IoT (Internet of Things)* và *Smart Home*:

- **Nghiên cứu khoa học:** Nhiều công trình nghiên cứu đã đề xuất mô hình sử dụng ESP32 hoặc ESP8266 kết nối với cảm biến khí gas (MQ-series), hiển thị dữ liệu qua Blynk, ThingSpeak, hoặc Firebase, kèm theo điều khiển từ xa và gửi thông báo về điện thoại khi phát hiện nồng độ khí vượt ngưỡng.
- **Ứng dụng thực tiễn:** Các thiết bị thông minh giám sát khí gas trong gia đình, kho hàng, nhà máy ở Mỹ, Hàn Quốc, Trung Quốc... đã được thương mại hóa, với khả năng cập nhật dữ liệu thời gian thực, gửi cảnh báo qua Wi-Fi, điều khiển tự động thiết bị như quạt thông gió, còi báo động, đóng mở van gas tự động.
- **Mã nguồn mở và cộng đồng phát triển:** Trên các diễn đàn kỹ thuật như GitHub, Hackster.io, Instructables... có nhiều dự án mẫu sử dụng ESP32, Firebase, MQTT, kết hợp với cảm biến khí gas, phục vụ mục đích học tập và nghiên cứu. Tuy nhiên, không phải dự án nào cũng có tính hoàn chỉnh và ứng dụng thực tế cao.

Nhìn chung, các nghiên cứu ngoài nước có mức độ tích hợp và ứng dụng thực tiễn cao, là cơ sở tham khảo quan trọng để nhóm phát triển và điều chỉnh đề tài phù hợp với điều kiện trong nước.

1.3 MỤC TIÊU CỦA ĐỀ TÀI

Em thực hiện đề tài xây dựng một hệ thống giám sát và cảnh báo nồng độ khí gas, nhiệt độ, độ ẩm trong môi trường, kết hợp phần cứng và phần mềm, với các mục tiêu cụ thể sau:

- ❖ **Xây dựng mô hình mạch in hoàn chỉnh:** Thiết kế và triển khai mạch in sử dụng vi điều khiển ESP32, kết nối với các cảm biến như MQ-2 để đo khí gas, và DHT11 để đo nhiệt độ, độ ẩm môi trường. Mô hình được tối ưu hóa để đảm bảo độ chính xác, độ ổn định và độ tin cậy trong việc phát hiện khí gas độc hại cũng như nhiệt độ vượt ngưỡng an toàn.
- ❖ **Xử lý dữ liệu và cảnh báo:** Hệ thống sẽ được lập trình để xử lý dữ liệu từ cảm biến theo thời gian thực. Nếu giá trị gas vượt ngưỡng an toàn, hệ thống sẽ:

- Bật quạt (FAN) để khuếch tán và làm loãng khí gas.
- Kích hoạt còi báo động (BUZZER) để cảnh báo người xung quanh.
- Gửi cảnh báo lên Firebase, từ đó hiển thị thông báo trên web dashboard.

❖ **Hiển thị và điều khiển từ xa:** Sử dụng Firebase Realtime Database làm trung gian lưu trữ và truyền nhận dữ liệu. Cho phép người dùng theo dõi dữ liệu từ xa thông qua giao diện web. Hỗ trợ điều khiển thiết bị từ xa (bật/tắt quạt, còi, mở/đóng cửa) trong chế độ Manual.

❖ **Hiển thị dữ liệu trực tiếp:** tích hợp màn hình TFT ILI9225 để hiển thị:

- Nồng độ khí gas.
- Nhiệt độ, độ ẩm môi trường.
- Trạng thái thiết bị (còi, quạt, cửa).
- Ngưỡng cảnh báo hiện tại và chế độ hoạt động (Auto/Manual).

❖ **Cài đặt và điều chỉnh ngưỡng linh hoạt:** cho phép điều chỉnh ngưỡng cảnh báo gas và nhiệt độ bằng:

- Nút nhấn phần cứng (MODE, UP, DOWN, ON/OFF).
- Giao diện người dùng trên web.
- Tăng tính linh hoạt, giúp hệ thống phù hợp với nhiều môi trường khác nhau.

❖ **Tăng cường tính an toàn và tiện ích:** cung cấp một giải pháp an toàn và tiện ích cho việc giám sát và xử lý khí gas độc hại. Bằng cách sử dụng web dashboard, người dùng có thể nhận được thông báo cảnh báo ngay khi có sự vượt ngưỡng gas và có thể điều khiển quạt để xử lý tình huống nguy hiểm.

❖ **Đơn giản hóa quá trình giám sát:** Nhằm đơn giản hóa quá trình giám sát bằng cách cung cấp một giao diện trực quan và dễ sử dụng. Người dùng có thể dễ dàng xem thông tin cảm biến, nhận cảnh báo và điều khiển từ xa chỉ thông qua web dashboard.

Tổng thể, mục tiêu đề tài là cung cấp một giải pháp tích hợp phần cứng và phần mềm để giám sát và xử lý khí gas độc hại một cách hiệu quả, đồng thời tăng cường tính an toàn và tiện ích cho người dùng.

1.4 PHƯƠNG PHÁP NGHIÊN CỨU

Để thực hiện đề tài, nhóm đã áp dụng các phương pháp nghiên cứu sau:

1.4.1. Phương pháp thu thập tài liệu

Thu thập, tổng hợp các tài liệu, giáo trình, sách chuyên ngành và các bài báo khoa học liên quan đến:

- Hệ thống IoT
- Vi điều khiển ESP32.
- Giao tiếp và cấu trúc dữ liệu Firebase.
- Cảm biến khí gas MQ-2, DHT11 và các thiết bị ngoại vi khác.

Nghiên cứu hướng dẫn kỹ thuật của các linh kiện, datasheet, thư viện lập trình.

1.4.2. Phương pháp phân tích - thiết kế hệ thống

- Phân tích yêu cầu thực tế để xác định chức năng chính của hệ thống: đo lường, hiển thị, điều khiển và cảnh báo.
- Thiết kế hệ thống gồm các khối chức năng: cảm biến, vi điều khiển, hiển thị, điều khiển thiết bị và giao tiếp không dây.
- Xây dựng sơ đồ khối, sơ đồ nguyên lý và thiết kế mạch in.

1.4.3. Phương pháp thực nghiệm

- Tiến hành lắp ráp phần cứng theo sơ đồ nguyên lý.
- Lập trình vi điều khiển ESP32 để thu thập dữ liệu từ cảm biến, xử lý dữ liệu và giao tiếp với Firebase.
- Hiển thị dữ liệu đo được lên màn hình TFT và gửi dữ liệu lên Firebase Realtime Database.

- Tạo giao diện Web Dashboard để giám sát và điều khiển thiết bị từ xa.

1.4.4. Phương pháp kiểm tra và đánh giá

- Thực hiện đo kiểm trong nhiều điều kiện khác nhau để đánh giá độ chính xác của cảm biến.
- So sánh dữ liệu thực tế và dữ liệu ghi nhận trên Firebase để xác định độ trễ, độ ổn định hệ thống.
- Thử nghiệm chức năng điều khiển thiết bị qua nút nhấn và qua giao diện web để đánh giá độ phản hồi và tính khả thi.

1.5 BỐ CỤC ĐỀ TÀI

Bài báo cáo được trình bày gồm 7 chương:

- Chương 1 – Tổng quan đề tài :

Trình bày lý do chọn đề tài, tình hình nghiên cứu trong và ngoài nước, mục tiêu, phương pháp nghiên cứu và bố cục tổng thể của báo cáo.

- Chương 2 – Cơ sở lý thuyết:

Cung cấp kiến thức nền tảng về ESP32, hệ thống IoT, Firebase Realtime Database, các cảm biến, chuẩn giao tiếp và phần mềm lập trình.

- Chương 3 – Thiết kế hệ thống:

Phân tích yêu cầu, thiết kế phần cứng, phần mềm, xây dựng sơ đồ nguyên lý, mạch in và cấu trúc dữ liệu Firebase.

- Chương 4 – Thực nghiệm và đánh giá hệ thống:

Trình bày mô hình thực tế, kết quả thực nghiệm, giao diện người dùng, đánh giá hoạt động của hệ thống về độ chính xác, độ trễ và độ ổn định.

- Chương 5 – Kết luận và hướng phát triển:

Tổng kết quá trình thực hiện đề tài, đánh giá mức độ hoàn thành và đề xuất các hướng phát triển trong tương lai.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 GIỚI THIỆU VỀ HỆ THỐNG IOT

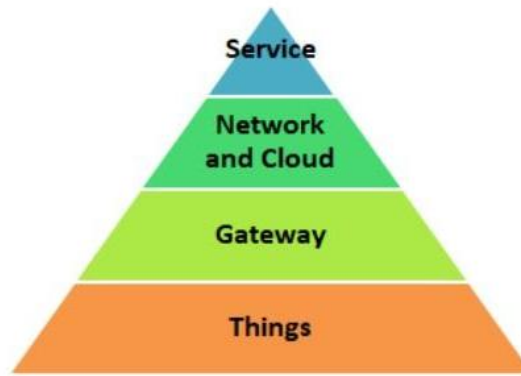
2.1.1 Khái niệm và ứng dụng

IoT được viết tắt từ Internet of things được gọi là Internet kết nối vạn vật hiện nay đang là một xu hướng công nghệ hiện đại đang thu hút sự quan tâm rộng rãi nhờ tính ứng dụng đa dạng và khả năng tạo ra những bước đột phá trong cuộc cách mạng công nghiệp 4.0. Mặc dù khái niệm IoT đã được biết đến từ nhiều thập kỷ trước, nhưng mãi đến năm 1999, thuật ngữ này mới thực sự được phổ biến và ứng dụng mạnh mẽ trong lĩnh vực công nghiệp hóa, mở ra cơ hội phát triển không ngừng cho đến ngày nay.

Một hệ thống IoT là một mạng lưới liên kết giữa các thiết bị điện tử với nhau thông qua các chuẩn giao tiếp không dây như Wi-Fi, LoRa, Zigbee, Bluetooth,... Các thiết bị trong hệ thống thường được tích hợp các bộ phận điện tử, phần mềm, cảm biến và các công nghệ mạng, cho phép thu thập dữ liệu từ môi trường xung quanh, xử lý thông tin và truyền tải dữ liệu lên các hệ thống máy tính hoặc đám mây để phân tích và điều khiển. Nhờ đó, IoT đóng vai trò quan trọng trong việc tối ưu hóa quá trình giám sát, tự động hóa và ra quyết định trong nhiều lĩnh vực như: nông nghiệp, công nghiệp, giao thông, y tế và nhà thông minh.

2.1.2 Cấu trúc của hệ thống IoT

Cấu trúc của hệ thống IoT gồm 4 phần chính: Thiết bị (Things), trạm kết nối (Gateways), hạ tầng mạng (Network and Cloud), dịch vụ (Services).



Hình 2-1 Cấu trúc hệ thống IoT

- **Thiết bị (Things):** Hiện nay trên thị trường các thiết bị công nghệ thông minh được sử dụng rất rộng rãi điển hình như các thiết bị điều khiển, cảm biến, đồng hồ và điện thoại thông minh với đặc điểm chung là đều được kết nối thông qua Internet. Đối với các thiết bị thông minh đã được tích hợp sẵn cơ sở hạ tầng sẽ được sàng lọc kết nối và quản lý dữ liệu một cách cục bộ. Đối với các thiết bị chưa được tích hợp thông minh có thể kết nối được thông qua các trạm kết nối.
- **Trạm kết nối (Gateway):** Gateway đóng vai trò là cầu nối giữa các thiết bị vật lý và hệ thống mạng. Nó thu thập dữ liệu từ thiết bị và truyền tải đến các dịch vụ xử lý dữ liệu ở tầng cao hơn. Trong một số hệ thống nhỏ, *vi điều khiển ESP32* có thể đồng thời đóng vai trò thiết bị và gateway khi được tích hợp Wi-Fi.
- **Hạ tầng mạng và điện toán đám mây (Network and Cloud):** Cơ sở hạ tầng mạng bao gồm các thiết bị tổng hợp, trạm kết nối và các thiết bị định tuyến,... để có thể kiểm soát được lưu lượng dữ liệu lưu thông bên cạnh đó cũng được kết nối với mạng lưới viễn thông triển khai bởi các nhà cung cấp. Điện toán đám mây sẽ bao gồm hệ thống các máy chủ, hệ thống mạng, lưu trữ ảo sẽ được kết nối.
- **Dịch vụ (Services):** là các ứng dụng giao diện người dùng (web, mobile app, dashboard) giúp trực quan hóa dữ liệu, đưa ra cảnh báo và cho phép điều khiển thiết bị từ xa. Ví dụ: *dashboard web hoặc ứng dụng Android* trong đề

tài cho phép người dùng xem nhiệt độ, độ ẩm, khí gas và điều khiển quạt, còi, cửa thông qua Firebase.

2.2. VI ĐIỀU KHIỂN ESP32

ESP32 là một vi điều khiển mạnh mẽ do Espressif Systems phát triển, được trang bị bộ vi xử lý lõi kép Tensilica Xtensa LX6 với tốc độ xử lý lên đến 240 MHz. ESP32 tích hợp sẵn các tính năng kết nối không dây như WiFi và Bluetooth, giúp nó trở thành lựa chọn lý tưởng cho các ứng dụng IoT.

Tính năng nổi bật của ESP32:

- ❖ Kết nối Wi-Fi và Bluetooth: Hỗ trợ chuẩn Wi-Fi 802.11 b/g/n và Bluetooth BLE, giúp giao tiếp không dây tiện lợi và linh hoạt.
- ❖ Bộ xử lý hiệu suất cao: CPU Xtensa dual-core 32-bit, xung nhịp lên đến 240MHz, giúp xử lý dữ liệu nhanh chóng, hiệu quả.
- ❖ Bộ nhớ Flash và RAM: RAM tích hợp 520KB và bộ nhớ lưu trữ lên đến 4MB, hỗ trợ các ứng dụng phức tạp.
- ❖ Hỗ trợ nhiều giao thức giao tiếp: Bao gồm UART, SPI, I2C, PWM, ADC, DAC, giúp ESP32 dễ dàng kết nối với nhiều loại cảm biến, màn hình, relay và các thiết bị ngoại vi khác.
- ❖ Tiêu thụ năng lượng thấp: ESP32 có nhiều chế độ tiết kiệm điện năng, thích hợp cho các thiết bị IoT hoạt động dài hạn.
- ❖ Hỗ trợ các giao thức mạng: TCP/IP, DNS, DHCP, MQTT, HTTP, giúp dễ dàng kết nối với các hệ thống đám mây như Firebase, AWS IoT, Google Cloud Platform.

Đối với các dự án IoT và nhúng, ESP32 rất phổ biến bởi tính tiện dụng, độ tin cậy và giá thành thấp. Các dự án có thể được xây dựng bằng cách sử dụng các thư viện có sẵn và các hướng dẫn được cung cấp bởi cộng đồng. Ngoài ra, có thể sử dụng ESP32 để kết nối với các dịch vụ đám mây như AWS IoT hoặc Google Cloud Platform để lưu trữ và xử lý dữ liệu từ các thiết bị IoT.



Hình 2-2 ESP32 (ESP32-WROOM-32)

❖ Thông số kỹ thuật

Thông số kỹ thuật	Mô tả
CPU	32-bit RISC CPU - Tensilica XtensaL106
Tốc độ xử lý	Lên đến 80MHz
Bộ nhớ flash tích hợp	512KB hoặc 1MB
Bộ nhớ RAM tích hợp	80KB
Chế độ ngủ tiêu thụ	0.9mW (DTIM3)
Kết nối Wi-Fi	802.11 b/g/n, tốc độ tối đa lên đến 72.2Mbps
Anten tích hợp	PCB hoặc IPEX
Cổng giao tiếp	UART, SPI, I2C, GPIO
Giao thức mạng hỗ trợ	TCP/IP, DNS, DHCP, MQTT, HTTP
Điện áp hoạt động	3.0 - 3.6V DC
Dòng tiêu thụ tối đa	200mA
Kích thước	18 x 24 mm
Hệ điều hành	Non-OS hoặc FreeRTOS

BẢNG 2-1 Thông số kỹ thuật ESP32 (ESP32-WROOM-32)

Sau khi phân tích và so sánh các dòng vi điều khiển phổ biến, em quyết định sử dụng ESP32 DevKit V1 vì các lý do sau:

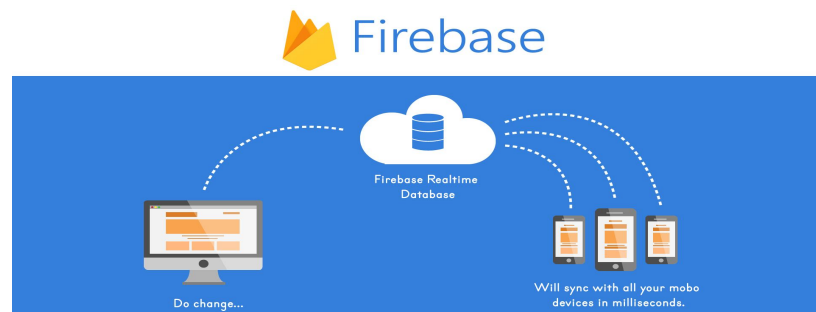
- ESP32 DevKit V1 có số lượng chân I/O lớn hơn so với ESP8266, đáp ứng đầy đủ nhu cầu kết nối với các thiết bị ngoại vi trong hệ thống.
- Được tích hợp sẵn cả Wi-Fi và Bluetooth, thuận tiện cho việc giao tiếp không dây và mở rộng ứng dụng IoT.
- Hỗ trợ đa dạng các giao tiếp như UART, SPI, I2C, ADC, PWM, giúp linh hoạt trong việc kết nối và điều khiển các cảm biến, thiết bị ngoại vi.
- Bộ xử lý mạnh mẽ với kiến trúc dual-core, đảm bảo hiệu suất cao cho việc xử lý tín hiệu từ cảm biến, điều khiển thiết bị và giao tiếp mạng đồng thời.

Với những ưu điểm trên, ESP32 DevKit V1 là lựa chọn tối ưu cho hệ thống giám sát và điều khiển trong đề tài.

2.3 GIAO TIẾP VỚI FIREBASE REALTIME DATABASE

2.3.1 Giới thiệu

Firebase Realtime Database là một cơ sở dữ liệu NoSQL lưu trữ dữ liệu dưới dạng JSON và cho phép đồng bộ dữ liệu theo thời gian thực giữa các thiết bị. Đây là một nền tảng đám mây do Google phát triển, rất phù hợp để sử dụng trong các ứng dụng IoT, nơi dữ liệu cần được cập nhật và truy xuất nhanh chóng.

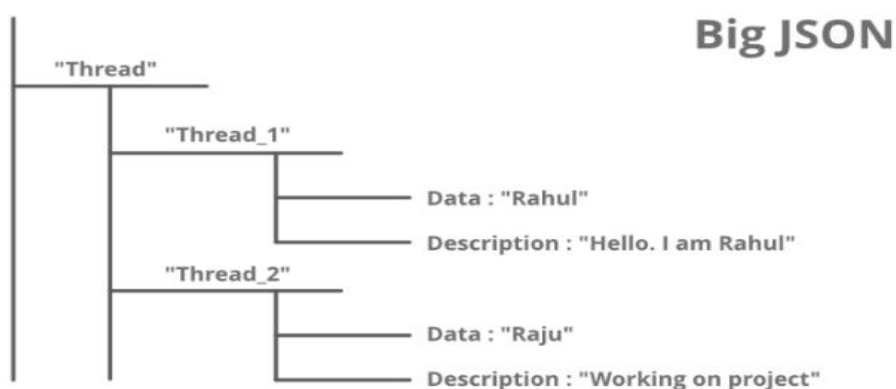


Hình 2-3 Mô hình Firebase Realtime Database

2.3.2 Cấu trúc Firebase

Firebase Realtime Database lưu trữ dữ liệu theo cấu trúc cây JSON và được đồng bộ hóa theo thời gian thực đối với mọi kết nối. Đối với các ứng dụng được xây dựng trên đa nền tảng như Android, IOS và WebApp, tất cả client sẽ cùng sử dụng một cơ sở dữ liệu. Bên cạnh đó, hệ thống dữ liệu này sẽ tự động cập nhật khi lập trình viên phát triển ứng dụng. Sau đó, tất cả dữ liệu này sẽ được truyền tải thông qua các kết nối SSL với độ bảo mật cao.

Với cấu trúc này, các thiết bị có thể ghi dữ liệu lên Firebase và đọc dữ liệu từ Firebase ngay khi có sự thay đổi.

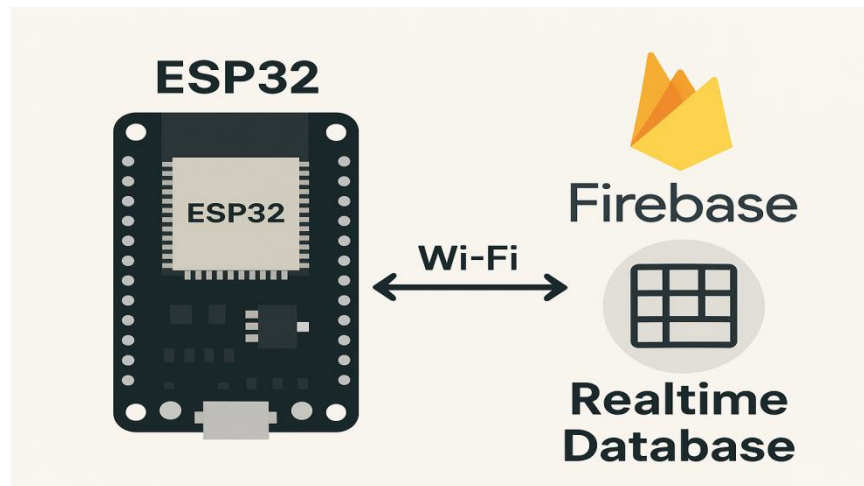


Hình 2-4 Cấu trúc dữ liệu của Firebase Realtime Database

2.3.3 Giao tiếp giữa ESP32 và Firebase

ESP32 có thể kết nối với Firebase Realtime Database thông qua giao thức HTTP hoặc giao thức WebSocket. Để giao tiếp, ESP32 sẽ sử dụng thư viện Firebase hỗ trợ trong môi trường Arduino IDE. Dữ liệu cảm biến được ESP32 thu thập sẽ được gửi lên Firebase, đồng thời ESP32 cũng có thể nhận dữ liệu điều khiển từ Firebase để điều khiển các thiết bị ngoại vi như relay, còi báo động hay servo.

Việc sử dụng Firebase giúp hệ thống dễ dàng mở rộng, quản lý từ xa và theo dõi dữ liệu thời gian thực qua giao diện web.



Hình 2-5 Giao tiếp giữa ESP32 và Firebase Realtime Database

2.4 PHẦN MỀM LẬP TRÌNH VI ĐIỀU KHIỂN

2.4.1 Giới thiệu về Arduino IDE

Arduino IDE (Arduino Integrated Development Environment) là một môi trường lập trình phổ biến dành cho việc phát triển hệ thống nhúng, đặc biệt là các ứng dụng liên quan đến vi điều khiển. Phần mềm này hỗ trợ lập trình bằng ngôn ngữ C và C++, giúp sinh viên và người mới tiếp cận lập trình nhúng dễ dàng hơn.

Arduino IDE hoạt động trên đa nền tảng gồm Windows, macOS, Linux và được xây dựng trên nền Java, cung cấp nhiều tính năng như soạn thảo mã nguồn, biên dịch và nạp chương trình trực tiếp vào vi điều khiển.

Phần mềm hỗ trợ nhiều dòng vi điều khiển Arduino như Arduino Uno, Arduino Mega, cũng như các module ESP như ESP8266, ESP32. Mỗi board mạch chứa một bộ vi điều khiển, có khả năng tiếp nhận và thực thi mã nguồn (sketch) được viết trong Arduino IDE. Sau khi lập trình, phần mềm biên dịch mã thành một tệp tin dạng Hex và nạp trực tiếp lên vi điều khiển để điều khiển hoạt động của hệ thống.

Arduino IDE là một công cụ quan trọng trong việc phát triển các hệ thống nhúng, hỗ trợ mạnh mẽ cho các dự án IoT, điều khiển tự động và lập trình vi điều khiển.

2.4.2 Giới thiệu về Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn nhẹ, mã nguồn mở do Microsoft phát triển, hỗ trợ nhiều ngôn ngữ lập trình như C/C++, Python, JavaScript,... và đặc biệt hữu ích trong lập trình vi điều khiển khi kết hợp với tiện ích mở rộng PlatformIO. Với khả năng hoạt động đa nền tảng (Windows, macOS, Linux), giao diện hiện đại, tùy biến linh hoạt và hỗ trợ kiểm tra lỗi, gợi ý cú pháp thông minh, VS Code giúp quá trình lập trình trở nên hiệu quả và chuyên nghiệp hơn. Khi sử dụng VS Code cùng PlatformIO để lập trình ESP32, người dùng có thể dễ dàng tạo, xây dựng, upload chương trình và giám sát dữ liệu qua Serial Monitor.

So với Arduino IDE, VS Code vượt trội về khả năng tổ chức dự án, hỗ trợ debug, quản lý thư viện và tích hợp Git. Đây là lựa chọn lý tưởng cho các dự án IoT có quy mô trung bình đến lớn, yêu cầu kiểm soát mã nguồn tốt và phát triển dài hạn.

2.5 GIAO DIỆN NGƯỜI DÙNG

2.5.1 Giao diện web

Trong hệ thống, giao diện người dùng được thiết kế dưới dạng một trang web dashboard cho phép theo dõi dữ liệu từ các node cảm biến theo thời gian thực. Giao diện này được lập trình bằng các ngôn ngữ cơ bản của web bao gồm:



Hình 2-6 Mối quan hệ giữa HTML, CSS và JavaScript trong thiết kế web

- **HTML (HyperText Markup Language):** là ngôn ngữ đánh dấu được sử dụng để xây dựng cấu trúc của trang web. Nó giống như "bộ khung" để định

nghĩa các thành phần như: tiêu đề, đoạn văn, hình ảnh, bảng, biểu mẫu nhập liệu, nút nhấn,...

Các thành phần HTML được viết dưới dạng thẻ (tag) và có thể lồng vào nhau.

- **CSS (Cascading Style Sheets):** được dùng để thiết kế và trang trí giao diện cho các phần tử HTML. Nếu HTML là bộ khung, thì CSS là lớp “sơn” tạo nên màu sắc, bố cục, phông chữ, hiệu ứng,...

CSS có thể viết trực tiếp trong HTML (inline), trong phần <style>, hoặc tách ra file riêng (.css). Khi áp dụng CSS này, giao diện web trở nên đẹp mắt, dễ nhìn hơn và phù hợp với trải nghiệm người dùng.

- **JavaScript (Ngôn ngữ lập trình tương tác):** là ngôn ngữ lập trình giúp xử lý logic và tương tác trên trang web. Nhờ JavaScript, người dùng có thể tương tác với dashboard như:

- Nhấn nút bật/tắt thiết bị
- Tự động cập nhật dữ liệu cảm biến từ Firebase
- Hiện thị cảnh báo khi khí gas vượt ngưỡng

JavaScript có thể kết hợp với Firebase thông qua REST API hoặc thư viện Firebase để lấy dữ liệu theo thời gian thực.

2.6. CÁC CHUẨN GIAO TIẾP SỬ DỤNG TRONG HỆ THỐNG

Trong hệ thống thiết bị đo nồng độ khí gas, nhiệt độ, độ ẩm sử dụng vi điều khiển ESP32, việc lựa chọn chuẩn giao tiếp phù hợp giữa vi điều khiển và các thiết bị ngoại vi đóng vai trò quyết định trong việc đảm bảo truyền dữ liệu ổn định, chính xác và hiệu quả. Hệ thống sử dụng các chuẩn giao tiếp phổ biến và phù hợp nhất hiện nay bao gồm: SPI, GPIO và PWM. Mỗi chuẩn giao tiếp có đặc điểm kỹ thuật và cách thức hoạt động riêng biệt, phù hợp với từng loại thiết bị ngoại vi cụ thể.

2.6.1. Giao tiếp SPI: màn hình TFT ILI9225

Khái niệm và nguyên lý hoạt động:

SPI là chuẩn giao tiếp nối tiếp tốc độ cao, trong đó vi điều khiển ESP32 đóng vai trò master điều khiển tín hiệu đồng hồ (clock) và truyền dữ liệu đến các thiết bị ngoại vi (slave). Chuẩn SPI sử dụng 4 dây tín hiệu chính gồm:

- **SCLK (Serial Clock):** dây tín hiệu đồng hồ đồng bộ do vi điều khiển (master) phát ra.
- **MOSI (Master Out Slave In):** dây truyền dữ liệu từ master (ESP32) đến slave (màn hình TFT).
- **MISO (Master In Slave Out):** dây truyền dữ liệu từ slave về master, thường không dùng khi giao tiếp một chiều như với màn hình TFT.
- **CS (Chip Select):** chân chọn thiết bị, khi kéo thấp, thiết bị slave được kích hoạt để nhận dữ liệu.

SPI hoạt động theo mô hình master-slave, vi điều khiển làm master điều khiển tín hiệu clock và truyền dữ liệu đến các thiết bị slave được chọn.

Ưu điểm của giao tiếp SPI:

- **Tốc độ truyền dữ liệu rất cao:** SPI có thể hoạt động với tần số clock lên đến vài chục MHz, cho phép truyền lượng lớn dữ liệu nhanh chóng, phù hợp với việc hiển thị hình ảnh và giao diện đồ họa trên màn hình TFT.
- **Độ tin cậy và ổn định cao:** SPI sử dụng đồng hồ để đồng bộ dữ liệu, giảm thiểu lỗi truyền và giúp dữ liệu được truyền chính xác.
- **Đơn giản trong thiết kế phần mềm:** Cơ chế truyền dữ liệu tuần tự và rõ ràng giúp lập trình dễ dàng.
- **Hỗ trợ nhiều thiết bị trên cùng bus:** Mỗi thiết bị slave có chân CS riêng, vi điều khiển dễ dàng lựa chọn thiết bị cần giao tiếp.

Ứng dụng trong hệ thống: Màn hình TFT ILI9225 sử dụng giao tiếp SPI để nhận dữ liệu hình ảnh, màu sắc và các lệnh điều khiển từ ESP32. Do đặc thù truyền dữ liệu đồ họa với dung lượng lớn, tốc độ cao là yêu cầu bắt buộc để hiển thị mượt mà, SPI là giao tiếp phù hợp nhất.

2.6.2. Giao tiếp GPIO: Cảm biến MQ-2, DHT11, Relay, còi, nút nhấn

Khái niệm: GPIO là các chân số trên vi điều khiển ESP32 có thể được cấu hình linh hoạt làm tín hiệu vào (Input) hoặc ra (Output) để giao tiếp trực tiếp với các thiết bị ngoại vi. Giao tiếp GPIO là cách thức truyền dữ liệu đơn giản nhất, thường dùng để điều khiển hoặc nhận tín hiệu dạng nhị phân (bật/tắt) hoặc tín hiệu analog (điện áp thay đổi).

Ứng dụng cụ thể với các cảm biến trong hệ thống:

- **Cảm biến DHT11:** Đây là cảm biến đo nhiệt độ và độ ẩm kỹ thuật số sử dụng giao tiếp digital một dây (single-wire protocol). Vi điều khiển ESP32 gửi tín hiệu bắt đầu (start signal) qua chân GPIO được kết nối với chân data của DHT11. Sau đó DHT11 phản hồi dữ liệu nhiệt độ và độ ẩm dưới dạng chuỗi bit số trên cùng chân đó. Chuẩn giao tiếp này không thuộc giao tiếp I2C hay SPI mà là giao tiếp custom dựa trên truyền nhận bit đơn giản, yêu cầu thời gian đáp ứng chính xác.
- **Cảm biến MQ-2:** MQ-2 là cảm biến phát hiện khí gas đa năng, có thể phát hiện các loại khí như gas, LPG, CO, và khói. MQ-2 cung cấp tín hiệu ra dạng analog (điện áp thay đổi theo nồng độ khí) hoặc tín hiệu digital khi vượt qua ngưỡng định trước. Vi điều khiển ESP32 sử dụng chân ADC (Analog-to-Digital Converter) thuộc GPIO để đọc điện áp analog từ MQ-2, sau đó chuyển đổi thành giá trị số đo nồng độ khí. Ngoài ra, nếu cần phát hiện nhanh tín hiệu vượt ngưỡng, chân digital output cũng được kết nối với GPIO để nhận tín hiệu bật/tắt.

- **Relay, còi báo động:** Các thiết bị này được điều khiển bật/tắt trực tiếp bằng tín hiệu số xuất ra từ chân GPIO. Vì điều khiển cung cấp mức điện áp 3.3V hoặc 5V tùy thiết kế để kích hoạt relay (thường qua mạch điều khiển transistor), còi báo hiệu.
- **Nút nhấn:** Chuyển chế độ Auto/Manual, tăng/giảm ngưỡng, điều khiển thủ công.

2.6.3. Giao tiếp PWM: Điều khiển Servo

Khái niệm: PWM là kỹ thuật điều chế độ rộng xung nhằm kiểm soát năng lượng trung bình cung cấp cho tải bằng cách điều chỉnh tỷ lệ thời gian tín hiệu ở mức cao so với toàn bộ chu kỳ. Tín hiệu PWM có tần số cố định, chỉ thay đổi độ rộng xung.

Ứng dụng trong hệ thống: Dùng để điều khiển servo, ví dụ như điều khiển Servo để thực hiện đóng/mở cửa khi phát hiện rò rỉ khí gas hoặc theo điều khiển thủ công từ nút nhấn hay Firebase. ESP32 sinh tín hiệu PWM với tần số và độ rộng xung tùy chỉnh, điều khiển chính xác vị trí góc quay servo.

2.7. CÁC MODULE PHẦN CỨNG

Hệ thống giám sát nồng độ khí gas, nhiệt độ và độ ẩm sử dụng các module cảm biến, thiết bị điều khiển và hiển thị để thu thập, xử lý và truyền dữ liệu đến người dùng. Dưới đây là các module phần cứng chính được sử dụng trong hệ thống.

2.7.1. Cảm biến khí gas MQ-2

- **Lý do chọn linh kiện:** Cảm biến khí gas MQ-2: được lựa chọn để phát hiện nồng độ khí gas trong không khí. MQ-2 có ưu điểm giá thành rẻ, kích thước nhỏ gọn, dễ sử dụng và dễ tích hợp vào hệ thống nhúng. Cảm biến này có thể phát hiện nhiều loại khí như LPG, propane, methane, alcohol, hydrogen... với độ nhạy cao và ngưỡng phát hiện có thể điều chỉnh được.

❖ Giới thiệu cảm biến MQ2:

MQ-2 là một loại cảm biến khí, thường được sử dụng để phát hiện và đo lượng khí trong không khí. Cảm biến MQ-2 có khả năng phát hiện các khí như LPG, i-butane, propane, methane, alcohol, Hydrogen và fume.

Cảm biến MQ-2 hoạt động trên nguyên lý tự nhiên điện hóa, nghĩa là nó đo sự thay đổi của điện trở khi được phơi nhiễm với khí. Khi cảm biến được phơi nhiễm với khí, chất nào đó trong khí sẽ tương tác với màng bảo vệ của cảm biến, làm thay đổi điện trở. Sau đó, mạch điện tử trong cảm biến sẽ đo thay đổi này và xuất ra một tín hiệu analog hoặc số để cho biết lượng khí hiện tại trong không khí.

Cảm biến MQ-2 có thiết kế nhỏ gọn, dễ dàng sử dụng và được tích hợp sẵn trên nhiều thiết bị điện tử, chẳng hạn như module cảm biến MQ-2. Nó được sử dụng rộng rãi trong các ứng dụng đo lường khí, như trong hệ thống báo khí gas tự động hoặc các hệ thống kiểm tra khí trong phòng thí nghiệm.



Hình 2-7 Cảm biến khí gas MQ2

➤ **Thông số kĩ thuật cảm biến MQ2**

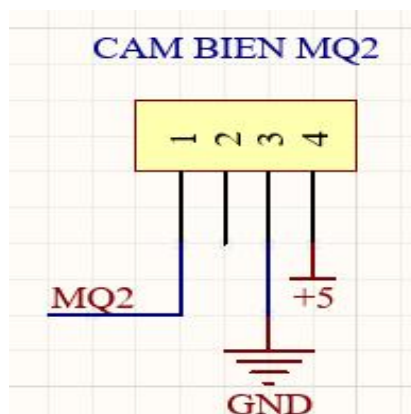
Thông số kĩ thuật	Mô tả
Điện áp hoạt động	3.3-5V DC
Dòng tiêu thụ	Khoảng 150mA
Các khí có thể phát hiện	LPG, i-butane, propane, methane, alcohol, Hydrogen
Điện trở đầu ra	1K Ω ~ 10K Ω tại nồng độ khí khác nhau
Độ nhạy	Nồng độ khí có thể phát hiện từ vài ppm đến vài trăm ppm
Nhiệt độ hoạt động	Từ -20 đến +50 độ C
Độ ẩm tương đối hoạt động	Từ 20 đến 90% không ngưng tụ

BẢNG 2-2 Thông số kĩ thuật của MQ2

➤ **Kết nối chân giữa ESP32 và cảm biến MQ2**

ESP32	Cảm biến MQ2
D34	AOUT(analog)
3.3V	VCC
GND	GND

BẢNG 2-3 Kết nối chân của ESP với cảm biến MQ2



Hình 2-8 Kết nối chân của ESP với cảm biến MQ2

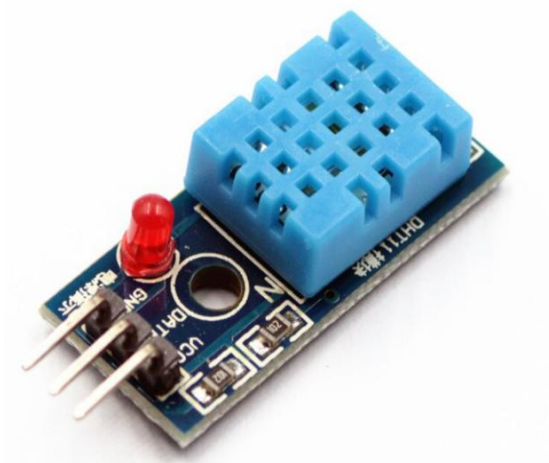
2.7.2. Cảm biến nhiệt độ và độ ẩm DHT11

- **Lý do chọn linh kiện:** Cảm biến nhiệt độ và độ ẩm DHT11: được sử dụng để đo nhiệt độ và độ ẩm môi trường. DHT11 sử dụng giao tiếp one-wire, giúp tiết kiệm chân GPIO trên vi điều khiển. Đây là lựa chọn phù hợp khi sử dụng ESP32 – mặc dù có nhiều chân GPIO hơn ESP8266 nhưng vẫn cần tối ưu tài nguyên phần cứng cho các chức năng khác.

❖ Giới thiệu cảm biến DHT11:

Cảm biến DHT11 là một cảm biến đo nhiệt độ và độ ẩm trong không khí với độ chính xác và độ ổn định tương đối cao. Nó được sử dụng rộng rãi trong các ứng dụng như hệ thống điều khiển tự động, các thiết bị đo nhiệt độ và độ ẩm và các dự án IoT.

Cảm biến DHT11 có một cảm biến nhiệt độ và một cảm biến độ ẩm được tích hợp trên cùng một bo mạch. Nó hoạt động ở mức điện áp 3.3V hoặc 5V và sử dụng giao tiếp 1-Wire thông qua chân GPIO để giao tiếp với vi điều khiển hoặc các thiết bị khác.



Hình 2-9 Cảm biến nhiệt độ và độ ẩm DHT11

➤ **Thông số kĩ thuật cảm biến DHT11**

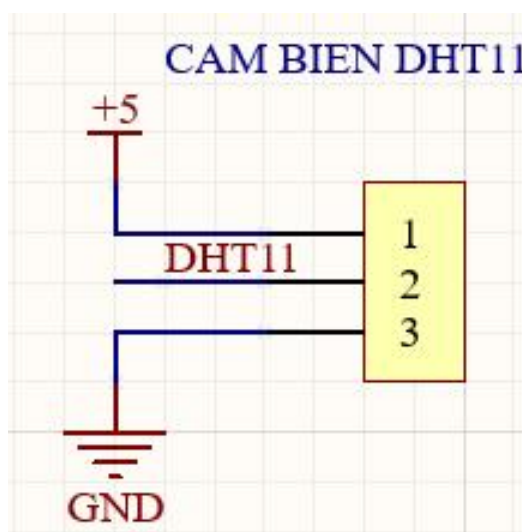
Thông số kĩ thuật	Mô tả
Điện áp hoạt động	3.3V hoặc 5V
Dòng tiêu thụ	khoảng 2.5mA
Dải đo nhiệt độ	0°C – 50°C (sai số $\pm 2^{\circ}\text{C}$)
Dải đo độ ẩm	20% – 90% RH (sai số $\pm 5\%$)
Độ phân giải	1°C cho nhiệt độ và 1% RH cho độ ẩm
Tần số giao tiếp	1Hz (1 lần/giây)

BẢNG 2-4 Thông số kĩ thuật cảm biến DHT11

➤ **Kết nối chân giữa ESP32 và cảm biến DHT11**

ESP32	Cảm biến DHT11
D21	OUT
GND	GND

BẢNG 2-5 Kết nối chân của ESP với cảm biến DHT11



Hình 2-10 Kết nối chân của ESP với cảm biến DHT11

2.7.3.Màn hình TFT ILI9225

- **Lý do chọn linh kiện:** Màn hình TFT ILI9225 được lựa chọn vì kích thước nhỏ gọn (2.2 inch) cùng độ phân giải phù hợp (176x220 pixels) giúp hiển thị rõ ràng các thông số môi trường và trạng thái thiết bị. Giao tiếp SPI tiết kiệm chân GPIO trên ESP32, đồng thời màn hình hoạt động ổn định ở mức điện áp 3.3V và có thư viện lập trình phổ biến, thuận tiện cho việc phát triển hệ thống IoT.

❖ Giới thiệu màn hình TFT ILI9225:

TFT ILI9225 là màn hình LCD màu sử dụng công nghệ Thin-Film Transistor (TFT), cho hình ảnh sáng, sắc nét với độ tương phản cao. Màn hình có độ phân giải 176x220 pixel trên kích thước 2.2 inch, phù hợp để hiển thị số liệu, biểu đồ và các biểu tượng trạng thái trong hệ thống giám sát.

Màn hình giao tiếp với vi điều khiển thông qua chuẩn SPI, cho tốc độ truyền nhanh và ổn định, đồng thời tiết kiệm chân GPIO. Với nguồn điện 3.3V, màn hình tương thích trực tiếp với ESP32 mà không cần mạch chuyển đổi điện áp, giúp đơn giản hóa thiết kế phần cứng.

➤ Thông số kỹ thuật của màn hình

Thông số kỹ thuật	Mô tả
Điện áp hoạt động	3.3 – 5VDC
Độ phân giải	176×220 pixel
Sử dụng giao tiếp	SPI
RAM	87120B

BẢNG 2-6 Thông số kỹ thuật của màn hình TFT

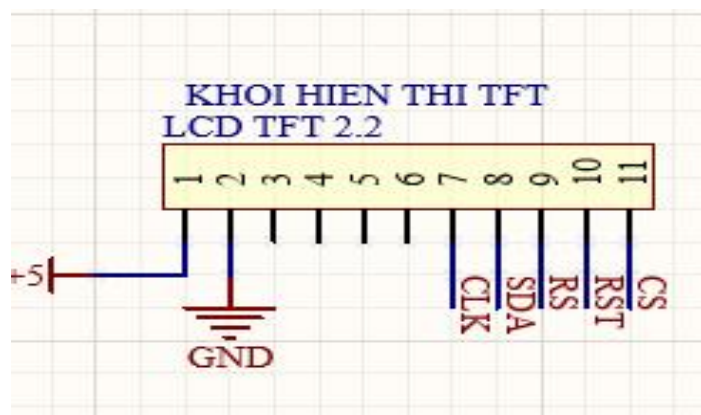


Hình 2-11 Màn hình TFT

❖ **Kết nối chân giữa ESP32 và TFT**

ESP32	Khối xử lý và chấp hành
D23	SDA
D18	CLK
D4	RST
D2	RS
D15	CS
GND	GND

BẢNG 2-7 Sơ đồ kết nối chân giữa ESP32 và TFT



Hình 2-12 Sơ đồ kết nối esp với màn hình TFT

2.7.4. Còi báo động, servo, quạt

➤ **Còi báo động**

- **Lý do chọn linh kiện:** Buzzer (Còi cảnh báo): dùng để phát ra âm thanh khi có cảnh báo, ví dụ như khi phát hiện khí gas vượt ngưỡng cho phép hoặc hệ thống có lỗi. Loại còi sử dụng là active buzzer 5V vì dễ điều khiển (chỉ cần bật/tắt mức logic).

❖ **Giới thiệu còi:**

- Còi là một thiết bị điện tử tạo ra âm thanh để thu hút sự chú ý hoặc cảnh báo. Có nhiều loại còi khác nhau, từ còi xe đạp đến còi cảnh báo đám đông trong trường học hoặc tòa nhà cao tầng. Các còi điện tử thường được sử dụng trong các ứng dụng như báo động, cảnh báo an toàn và các thiết bị cảnh báo khác.

➤ **Thông số kỹ thuật của còi:**

Thông số kỹ thuật	Mô tả
Điện áp hoạt động	2-5V
Dòng điện hoạt động	thường từ 20mA
Công suất tiêu thụ	thường từ 0,1W đến 2W
Độ ồn	thường từ 85dB đến 120dB
Tuổi thọ	thường từ 10.000 giờ đến 50.000 giờ

BẢNG 2-8 Thông số kỹ thuật còi

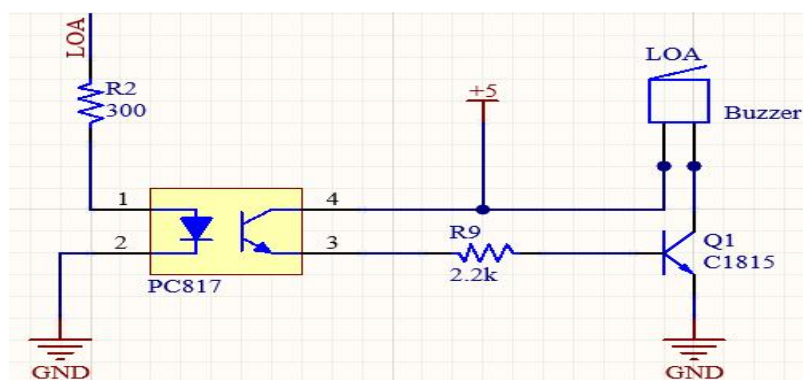


Hình 2-13 Còi chip 5V

❖ **Kết nối chân giữa ESP32 và khối cảnh báo**

ESP32	Khối cảnh báo
D12	Còi
GND	GND

BẢNG 2-9 Kết nối chân của ESP với còi



Hình 2-14 Sơ đồ kết nối giữa esp với khối cảnh báo

➤ **Servo**

- **Lý do chọn linh kiện:** Servo được chọn vì khả năng điều khiển chính xác góc quay, dễ dàng tích hợp với vi điều khiển qua tín hiệu PWM. Kích thước nhỏ gọn, tiêu thụ điện năng thấp và phản hồi nhanh giúp servo phù hợp cho các ứng dụng mở/đóng cửa hoặc van khí trong hệ thống giám sát môi trường. Ngoài ra, servo MG90S có giá thành hợp lý và độ bền cao, đáp ứng tốt yêu cầu sử dụng lâu dài.

❖ **Giới thiệu Servo:**

Servo là thiết bị truyền động có thể điều khiển vị trí góc quay thông qua tín hiệu điều khiển PWM từ vi điều khiển. Servo MG90S sử dụng động cơ DC kết hợp bộ điều khiển vi mạch tích hợp, cho phép điều chỉnh góc quay trong phạm vi 0° đến 180° với độ chính xác cao. Servo có cấu tạo nhỏ gọn, dễ lắp đặt và thường được sử dụng trong các ứng dụng robot, tự động hóa và điều khiển cơ khí nhỏ gọn.



Hình 2-15 Servo SG90 180 độ

➤ **Thông số kĩ thuật của động cơ Servo**

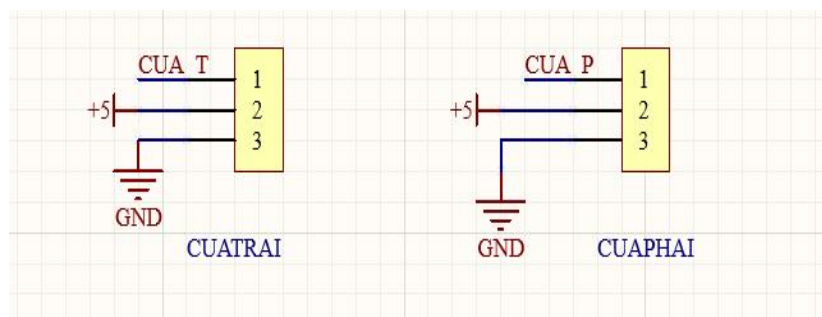
Thông số kĩ thuật	Mô tả
Điện áp hoạt động	5V DC
Kích thước	22.2x11.8.32 mm
Momen xoắn	1.8kg/cm
Góc xoay	180 độ

BẢNG 2-10 Thông số kĩ thuật của động cơ Servo

❖ **Kết nối chân giữa ESP32 và Servo SG90 180 độ**

ESP32	Khối xử lý và chấp hành
D14, D27	CUA_T và CUA_P
GND	GND

BẢNG 2-11 Sơ đồ nối giữa esp và Servo SG90 180 độ



Hình 2-16 Sơ đồ kết nối esp với Servo SG90

➤ Quạt

- **Lý do chọn linh kiện:** như đã đề cập ở trên khi khí gas cao sẽ có ảnh hưởng xấu tới sức khỏe nên cần một thiết bị giúp giảm bớt lượng khí gas trong không khí. Vì vậy quạt 12V là một lựa chọn phù hợp để giảm bớt lượng khí gas trong lúc khí gas vượt ngưỡng cảnh báo.

❖ Giới thiệu khối nút nhấn:

Quạt là thiết bị cơ điện dùng để tạo luồng không khí, hỗ trợ lưu thông gió hoặc làm mát môi trường. Trong hệ thống giám sát môi trường, quạt được sử dụng để giảm nồng độ khí gas, hạ nhiệt độ khi vượt ngưỡng, hoặc duy trì thông gió ổn định.

Quạt thường hoạt động ở điện áp 5V, 12V hoặc 220V, tùy loại, và được điều khiển gián tiếp thông qua module relay. Relay đóng vai trò như một công tắc điện tử, cho phép vi điều khiển bật/tắt quạt một cách an toàn. Khi cảm biến phát hiện nhiệt độ hoặc nồng độ khí vượt mức cài đặt, hệ thống sẽ kích hoạt relay để bật quạt, đảm bảo phản ứng kịp thời và hiệu quả.



Hình 2-17 Quạt 12V

➤ **Thông số kĩ thuật của quạt**

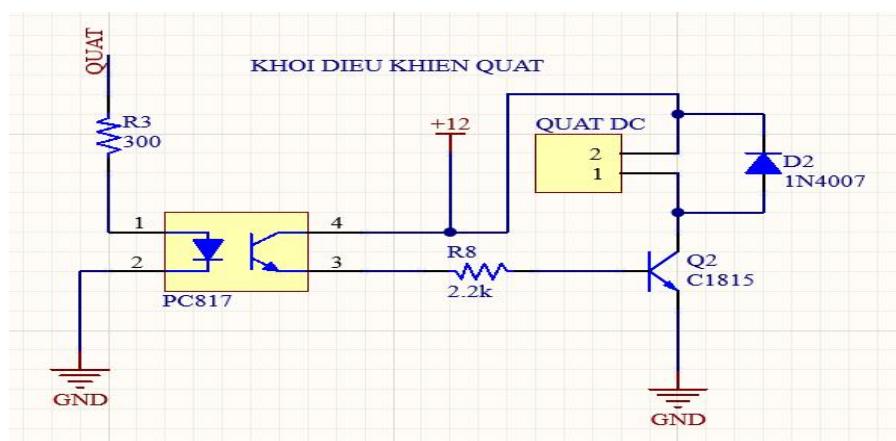
Thông số kĩ thuật	Mô tả
Điện áp hoạt động	12V DC
Dòng hoạt động	Dao động từ 0.1A đến 0.5A
Tốc độ quay	Dao động từ 1000 vòng/phút đến 3000 vòng/phút
Đường kính cánh quạt	120mm
Tuổi thọ	Từ 30.000 giờ đến 50.000 giờ hoạt động liên tục.

BẢNG 2-12 Thông số kĩ thuật của quạt 12V

❖ **Kết nối chân giữa ESP32 và quạt 12V**

ESP32	Khối xử lý và chấp hành
D13	QUAT
GND	GND

BẢNG 2-13 Kết nối chân giữa ESP32 và quạt 12V



Hình 2-18 Kết nối chân giữa ESP32 và quạt 12V

2.7.5. Nút nhấn

- **Lý do chọn linh kiện:** khối nút nhấn đóng vai trò là thiết bị nhập liệu thủ công, cho phép người dùng tương tác trực tiếp với hệ thống thông qua các thao tác cơ bản. Đây là một phần quan trọng trong các hệ thống điều khiển nhúng có giao diện người dùng đơn giản.

❖ Giới thiệu khối nút nhấn:

Khối nút nhấn bao gồm bốn nút: UP, DOWN, MODE và ON/OFF, đóng vai trò là thiết bị nhập liệu thủ công, cho phép người dùng tương tác trực tiếp với hệ thống. Thông qua các nút này, người dùng có thể chuyển đổi giữa các chế độ hoạt động (Auto, Manual, Cài đặt), cũng như điều khiển trực tiếp các thiết bị như quạt, còi báo động và cửa servo.

- Các nút được sử dụng trong hệ thống gồm:

➤ Nút UP / DOWN: dùng để tăng hoặc giảm giá trị ngưỡng của các tham số như nhiệt độ hoặc nồng độ khí gas trong chế độ cài đặt. Ngoài ra, trong chế độ điều khiển thủ công (Manual), các nút này có thể đảm nhận vai trò bật quạt hoặc điều khiển cửa.

➤ Nút MODE: dùng để chuyển đổi giữa các chế độ thiết lập ngưỡng, như chuyển giữa ngưỡng nhiệt độ và ngưỡng khí gas. Có thể mở rộng thêm để chuyển chế độ hoạt động chính của hệ thống (Auto / Manual).

➤ Nút ON/OFF: được sử dụng để bật hoặc tắt còi báo động thủ công trong chế độ Manual, hoặc kích hoạt chế độ cài đặt ngưỡng khi hệ thống đang ở chế độ Auto.

Các nút nhấn sử dụng cấu trúc kết nối pull-up hoặc pull-down với điện trở để đảm bảo tín hiệu ổn định. Khi người dùng nhấn nút, một mức điện áp (HIGH hoặc LOW) sẽ được ghi nhận bởi các chân GPIO của ESP32 và được xử lý thông qua phần mềm.



Hình 2-19 Nút nhấn

➤ **Thông số kĩ thuật của nút nhấn dán 6x6x10mm (4 chân)**

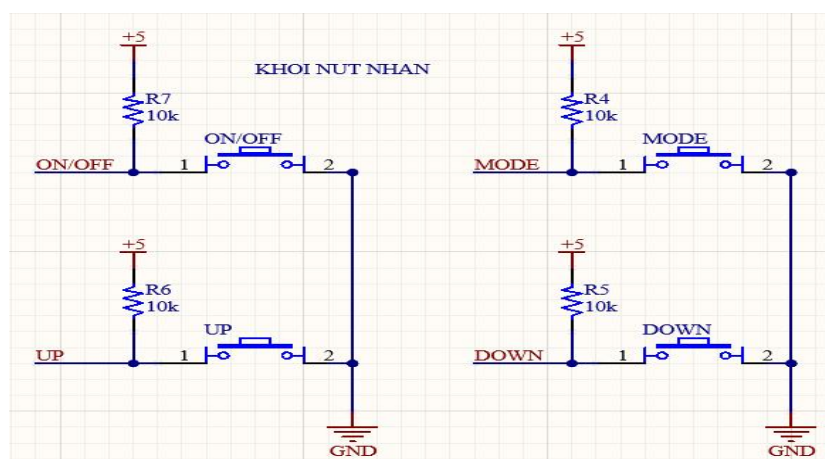
Thông số kĩ thuật	Mô tả
Kích thước thân nút	6mm x 6mm (rộng x dài)
Chiều cao nút	10mm (tính từ mặt PCB lên đỉnh nút)
Số chân	4 chân (kết nối 2 cặp chân đối xứng)
Kiểu kết nối	Hàn dán qua lỗ (THT) hoặc dán bề mặt (SMD), tùy loại
Dòng điện định mức	50mA @ 12VDC
Điện áp hoạt động	Tối đa 12V DC
Điện trở tiếp xúc	$\leq 100\text{ m}\Omega$
Nhiệt độ hoạt động	-25°C đến +70°C

BẢNG 2-14 Thông số kĩ thuật của nút nhấn dán 6x6x10mm (4 chân)

➤ **Kết nối chân giữa ESP32 và các nút nhấn**

ESP32	Cảm biến DHT11
D32	ON/OFF
D26	MODE
D33	UP
D25	DOWN

BẢNG 2-15 Kết nối chân của ESP với các nút nhấn



Hình 2-20 Kết nối chân của ESP với khối nút nhấn

2.7.6. Các thiết bị ngoại vi

➤ Transistor C1815

- **Lý do chọn linh kiện:** Transistor C1815 được lựa chọn vì có đặc tính khuếch đại tốt, dòng thu lớn ($I_C \text{ max} \approx 150\text{mA}$), điện áp hoạt động phù hợp ($V_{CE \text{ max}} = 50\text{V}$), rất thích hợp để điều khiển các thiết bị công suất nhỏ như relay, còi báo hoặc LED trong hệ thống. Ngoài ra, linh kiện này phổ biến, dễ tìm, giá thành rẻ và dễ tích hợp vào mạch điều khiển với vi điều khiển như ESP32.

❖ Giới thiệu Transistor C1815:

- Transistor C1815 là một trong hai loại transistor NPN phổ biến. Trong transistor C1815, có ba lớp bán dẫn, transistor C1815 được sử dụng trong nhiều ứng dụng khác nhau như mạch khuếch đại, bộ chuyển đổi tín hiệu và công tắc. Khi được sử dụng như một công tắc, transistor NPN sẽ ở trạng thái dẫn điện khi có dòng điện vào cực b và cho phép dòng điện lớn hơn chạy qua cực emitter.



Hình 2-21 C1815 NPN Transistor

➤ **Thông số kỹ thuật của Transistor C1815:**

Thông số kỹ thuật	Mô tả
Điện áp tối đa giữa collector và emitter (VCEO)	50V
Điện áp tối đa giữa collector và base (VCBO)	60V
Điện áp tối đa giữa emitter và base (VEBO)	5V
Dòng điện collector tối đa (IC)	150mA
Công suất tiêu thụ tối đa (Pd)	625mW
Hệ số khuếch đại (hfe)	từ 70 đến 700
Tần số chuyển đổi tối đa (fT)	100MHz

BẢNG 2-16 Thông số kỹ thuật C1815

➤ **Optocoupler PC817**

- **Lý do chọn linh kiện:** Linh kiện PC817 được chọn vì tính năng cách ly tốt giữa đầu vào và đầu ra, khả năng chịu được điện áp và dòng tốt, độ tin cậy cao và chi phí phù hợp. Nó là sự lựa chọn lý tưởng cho các ứng dụng điều khiển với đầu vào điện áp thấp và đầu ra tải nặng. Việc này giúp cách ly tín hiệu giữa các thành phần trong hệ thống, giảm thiểu nguy cơ nhiễu tín hiệu và đảm bảo tính ổn định của hệ thống.

❖ **Giới thiệu PC817:**

-PC817 là một optocoupler, được sử dụng để cách ly tín hiệu giữa hai mạch điện. Nó bao gồm một đèn LED và một phototransistor được tích hợp trong cùng một vỏ. Khi đèn LED được kích hoạt, nó phát ra ánh sáng và kích thích phototransistor, giúp truyền tín hiệu giữa hai mạch điện một cách an toàn và cách ly. PC817 được sử dụng rộng rãi trong các mạch điện tử, đặc biệt là trong các ứng dụng điều khiển và bảo vệ mạch điện.



Hình 2-22 PC817

➤ **Thông số kỹ thuật của PC817**

Thông số kỹ thuật	Mô tả
Điện áp cách ly	5,000Vrms
Dòng điện đầu vào	50mA
Điện áp đầu vào	1.2V-1.5V
Dòng điện đầu ra	50mA
Điện áp đầu ra tối đa	80V
Số chân	4 chân
Nhiệt độ hoạt động	-55 độ C đến 100 độ C
Loại linh kiện	Optocoupler

BẢNG 2-17 Thông số kỹ thuật PC817

➤ **IC7805**

- **Lý do chọn linh kiện:** IC 7805 được chọn vì là bộ ổn áp tuyến tính phổ biến, cung cấp điện áp đầu ra ổn định 5V với dòng điện lên đến 1A, phù hợp để cấp nguồn cho các module cảm biến, relay và thiết bị ngoại vi trong hệ thống. IC có cấu tạo đơn giản, dễ dàng sử dụng và có khả năng bảo vệ quá nhiệt, giúp tăng độ bền và ổn định cho mạch nguồn.

❖ **Giới thiệu IC7805:**

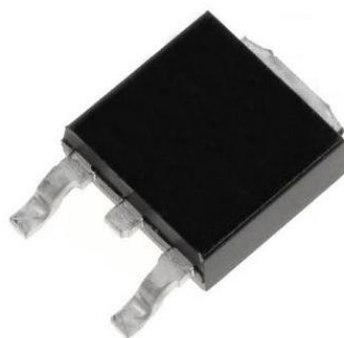
-IC 7805 là bộ điều áp tuyến tính thuộc họ 78xx, chuyên dùng để cung cấp điện áp ổn định 5V từ nguồn đầu vào có điện áp cao hơn (thường từ 7V đến 35V). IC này có 3 chân: đầu vào (Input), đầu ra (Output) và chân mass (Ground). Với khả năng cung cấp dòng điện tối đa khoảng 1A, 7805 thường được dùng trong

các mạch điện tử nhúng để đảm bảo cung cấp nguồn ổn định cho các linh kiện hoạt động chính xác và bền bỉ.

➤ **Thông số kĩ thuật của IC7805**

Thông số kĩ thuật	Mô tả
Điện áp đầu vào	7V đến 35V
Điện áp đầu ra	5V DC
Dòng điện đầu ra	1.5A
Công suất	Tối đa 15W
Nhiệt độ hoạt động	Từ -40 đến +125 độ C

BẢNG 2-18 Thông số kĩ thuật IC7805



Hình 2-23 IC7805

➤ **Adapter 12V 2A**

- **Lý do chọn linh kiện:** Adapter 12V công suất 2A được lựa chọn vì cung cấp đủ dòng điện ổn định để cấp nguồn cho toàn bộ hệ thống, bao gồm vi điều khiển ESP32, cảm biến, màn hình TFT, relay, còi báo động và servo. Công suất 2A đảm bảo hệ thống hoạt động liên tục, không bị sụt áp khi các thiết bị tiêu thụ đồng thời, đồng thời tạo điều kiện thuận lợi cho việc chuyển đổi điện áp xuống các mức thấp hơn như 5V và 3.3V.

➤ **Giới thiệu Adapter 12V 2A:**

Adapter 12V 2A là thiết bị chuyển đổi nguồn điện từ lưới điện 220V/110V sang điện áp một chiều 12V với dòng điện tối đa 2A. Adapter này cung cấp nguồn điện ổn định, phù hợp cho các hệ thống điện tử vừa và nhỏ.

Thiết bị có kích thước nhỏ gọn, dễ dàng lắp đặt, đồng thời được trang bị các tính năng bảo vệ như chống quá tải và chống ngắn mạch nhằm đảm bảo an toàn cho thiết bị và người sử dụng.

➤ **Thông số kỹ thuật của adapter**

Thông số kỹ thuật	Mô tả
Điện áp đầu vào	100-240V AC, 50-60Hz
Điện áp đầu ra	12V DC
Dòng điện đầu ra	2A (max)
Công suất đầu ra	24W (max)
Đầu cắm	5.5mm x 2.1mm
Màu sắc	Đen hoặc trắng
Chất liệu vỏ	Nhựa ABS hoặc hợp kim nhôm
Bảo vệ	Quá dòng, quá áp, ngắn mạch

BẢNG 2-19 Thông số kỹ thuật adapter



Hình 2-24 Adapter 12V2A

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

3.1. Yêu cầu hệ thống

Hệ thống thiết bị đo nồng độ khí gas, nhiệt độ, độ ẩm sử dụng vi điều khiển ESP32 được thiết kế nhằm mục đích giám sát, cảnh báo và điều khiển thiết bị trong môi trường thực tế. Để đảm bảo hệ thống vận hành ổn định và hiệu quả, các yêu cầu chức năng và phi chức năng được xác định như sau:

3.1.1. Chức năng theo dõi dữ liệu cảm biến

Hệ thống phải có khả năng theo dõi liên tục các thông số môi trường xung quanh, bao gồm:

Đo nồng độ khí gas (cảm biến MQ-2): Cảm biến MQ-2 được sử dụng để phát hiện và đo lường nồng độ các khí cháy như LPG, butan, propane, methane và khói. Giá trị đo được là tín hiệu analog, sẽ được vi điều khiển ESP32 đọc thông qua chân ADC. Hệ thống phải có khả năng xử lý và chuyển đổi dữ liệu đầu vào thành giá trị nồng độ dễ hiểu để hiển thị và cảnh báo kịp thời khi nồng độ khí vượt ngưỡng an toàn.

Đo nhiệt độ và độ ẩm (cảm biến DHT11): Cảm biến DHT11 thu thập dữ liệu về nhiệt độ và độ ẩm trong môi trường. Dữ liệu được truyền qua giao tiếp một dây (1-wire) đến ESP32. Hệ thống phải đảm bảo ghi nhận dữ liệu chính xác và cập nhật thường xuyên.

Tần suất cập nhật: Hệ thống đọc và xử lý dữ liệu cảm biến mỗi chu kỳ 1 giây (delay = 1000ms). Tần suất này đảm bảo khả năng phản hồi nhanh với các thay đổi môi trường mà vẫn tối ưu hóa tài nguyên và thời gian xử lý của ESP32.

Gửi dữ liệu lên Firebase: Sau mỗi lần đọc, dữ liệu cảm biến được gửi lên Firebase Realtime Database để phục vụ mục đích giám sát từ xa thông qua giao diện web hoặc thiết bị di động.

Hiển thị dữ liệu tại chỗ: Tất cả các giá trị đo được được hiển thị trực tiếp trên màn hình TFT giúp người dùng theo dõi tức thời. Ngoài ra hiện giá trị

ngưỡng cảnh báo, trạng thái của thiết bị và chế độ hoạt động giúp người dùng hiển thị thông tin rõ ràng và dễ hiểu.

3.1.2. Chức năng điều khiển thiết bị từ Web Dashboard và nút nhấn

Hệ thống hỗ trợ điều khiển các thiết bị như quạt, còi báo động và cửa servo thông qua hai phương thức: điều khiển từ xa qua giao diện Web Dashboard kết nối Firebase và điều khiển trực tiếp bằng nút nhấn vật lý gắn trên thiết bị. Việc kết hợp này giúp tăng độ linh hoạt, tính tiện lợi và đảm bảo khả năng kiểm soát hệ thống trong cả môi trường có hoặc không có kết nối Internet.

➤ Các chế độ chính

Hệ thống hoạt động theo 3 chế độ:

Đầu tiên là “chế độ tự động”, hệ thống hoạt động hoàn toàn tự động dựa trên dữ liệu từ cảm biến nhiệt độ DHT11 và cảm biến khí gas MQ2. Khi nhiệt độ hoặc nồng độ khí gas vượt quá các ngưỡng an toàn đã được cấu hình, hệ thống sẽ tự động kích hoạt quạt để thông gió, bật còi để cảnh báo, và mở cửa để thoát khí, sau đó cập nhật trạng thái này lên Firebase. Khi các giá trị cảm biến trở lại mức an toàn, các thiết bị sẽ tự động tắt.

Tiếp theo, người dùng có thể dễ dàng chuyển sang “chế độ cài đặt” từ “chế độ tự động” để tinh chỉnh các ngưỡng an toàn cho nhiệt độ và khí gas thông qua các nút bấm vật lý, với các thay đổi được lưu trữ ngay lập tức trên Firebase.

Cuối cùng là “chế độ thủ công”, người dùng có toàn quyền kiểm soát các thiết bị. Thông qua các nút bấm trên thiết bị hoặc giao diện Firebase, người dùng có thể chủ động bật/tắt quạt, còi, và mở/đóng cửa theo ý muốn. Chế độ này phù hợp cho các tình huống cần sự can thiệp trực tiếp từ người dùng, và mọi thay đổi điều khiển cũng được đồng bộ hóa với Firebase. Thông tin về nhiệt độ, độ ẩm, nồng độ khí gas, trạng thái thiết bị và chế độ hoạt động hiện tại luôn được hiển thị rõ ràng trên màn hình TFT và xuất ra Serial Monitor, giúp người dùng dễ dàng theo dõi và quản lý hệ thống.

❖ Điều khiển từ xa qua Web Dashboard

Giao diện Web Dashboard được lập trình bằng HTML, CSS và JavaScript, kết nối với Firebase Realtime Database để nhận và gửi dữ liệu điều khiển theo thời gian thực.

Các chức năng điều khiển:

- ❖ **Bật/tắt quạt thông gió:** giúp làm loãng khí gas hoặc giảm nhiệt quá ngưỡng khi có cảnh báo.
- ❖ **Bật/tắt còi báo động:** để cảnh báo người dùng khi xảy ra rò rỉ khí gas hoặc nhiệt độ vượt ngưỡng.
- ❖ **Mở/đóng cửa (servo):** cho phép điều hướng dòng khí độc và nhiệt độ ra ngoài khi có nồng độ khí gas rò rỉ hoặc nhiệt độ cao.
- ❖ **Chuyển chế độ Auto/Manual:** cho phép người dùng thiết lập hệ thống hoạt động tự động hay điều khiển thủ công.
- ❖ **Điều chỉnh ngưỡng cảnh báo:** chỉnh các giá trị ngưỡng cho nhiệt độ và khí gas từ xa.

Đồng bộ trạng thái:

Mọi thao tác điều khiển được cập nhật tức thời lên Firebase, từ đó truyền xuống ESP32 để thực thi. Đồng thời, trạng thái thực tế của các thiết bị cũng được hiển thị lại trên giao diện Web để người dùng dễ theo dõi.

❖ Điều khiển trực tiếp bằng nút nhấn vật lý

Để đảm bảo tính an toàn và khả năng điều khiển trong tình huống mất kết nối mạng, hệ thống được trang bị 4 nút nhấn vật lý gắn trực tiếp trên mạch, mỗi nút đảm nhiệm một chức năng cụ thể:

Tên nút nhấn	Chức năng chính
MODE	Chuyển đổi giữa chế độ tự động(auto) và chế độ thủ công (manual). Nếu trong chế độ cài đặt ngưỡng, có chức năng chuyển đổi ngưỡng khí gas và ngưỡng nhiệt độ
ON/OFF	Ở chế độ tự động, có chức năng vào hoặc thoát chế độ cài đặt ngưỡng. Nếu ở chế độ thủ công, có chức năng bật/tắt còi.
DOWN	Trong chế độ cài đặt ngưỡng, có chức năng giảm ngưỡng khí hoặc ngưỡng nhiệt độ. Trong chế độ thủ công, mở/đóng cửa.
UP	Trong chế độ cài đặt ngưỡng, có chức năng tăng ngưỡng khí hoặc ngưỡng nhiệt độ. Trong chế độ thủ công, bật/tắt quạt.

BẢNG 3-1 Bảng chức năng của 4 nút nhấn

3.1.3. Chức năng hiển thị dữ liệu lên màn hình TFT

Màn hình TFT là thành phần quan trọng trong hệ thống, đóng vai trò cung cấp thông tin trực quan cho người dùng về trạng thái môi trường và thiết bị. Hệ thống sử dụng màn hình ILI9225 TFT 2.2 inch, giao tiếp SPI với vi điều khiển ESP32.

Mọi dữ liệu từ cảm biến và trạng thái thiết bị được cập nhật và hiển thị theo chu kỳ 1 giây, giúp người dùng theo dõi kịp thời mà không cần phụ thuộc vào thiết bị ngoại vi hay kết nối Internet. Tăng độ tin cậy trong vận hành hệ thống, đặc biệt trong môi trường dễ cháy hoặc yêu cầu cảnh báo nhanh.

❖ Thông tin hiển thị

Trên màn hình TFT, các thông tin được trình bày theo từng dòng rõ ràng, bao gồm:

Giá trị cảm biến:

- Nhiệt độ (°C) – đo bằng cảm biến DHT11.
- Độ ẩm (%) – đo bằng cảm biến DHT11.
- Nồng độ khí gas (ppm) – quy đổi từ cảm biến MQ2.

Ngưỡng cảnh báo:

- Ngưỡng nhiệt độ cho phép (có thể điều chỉnh).
- Ngưỡng khí gas cho phép (có thể điều chỉnh).

Trạng thái thiết bị:

- Quạt: Bật / Tắt
- Còi báo: Bật / Tắt
- Cửa servo: Mở / Đóng
- Chế độ hoạt động: Auto / Manual
- Trạng thái cài đặt: Điều chỉnh ngưỡng (Gas / Temp / None)

❖ Cập nhật và phản hồi thời gian thực

- Mỗi 1 giây (delay 1000 ms), dữ liệu mới được ghi nhận từ cảm biến và hiển thị lên màn hình.
- Khi có thay đổi về chế độ, trạng thái thiết bị hoặc ngưỡng cài đặt, màn hình sẽ cập nhật ngay lập tức để phản ánh chính xác tình trạng hệ thống.

❖ Phân biệt trạng thái bằng màu sắc

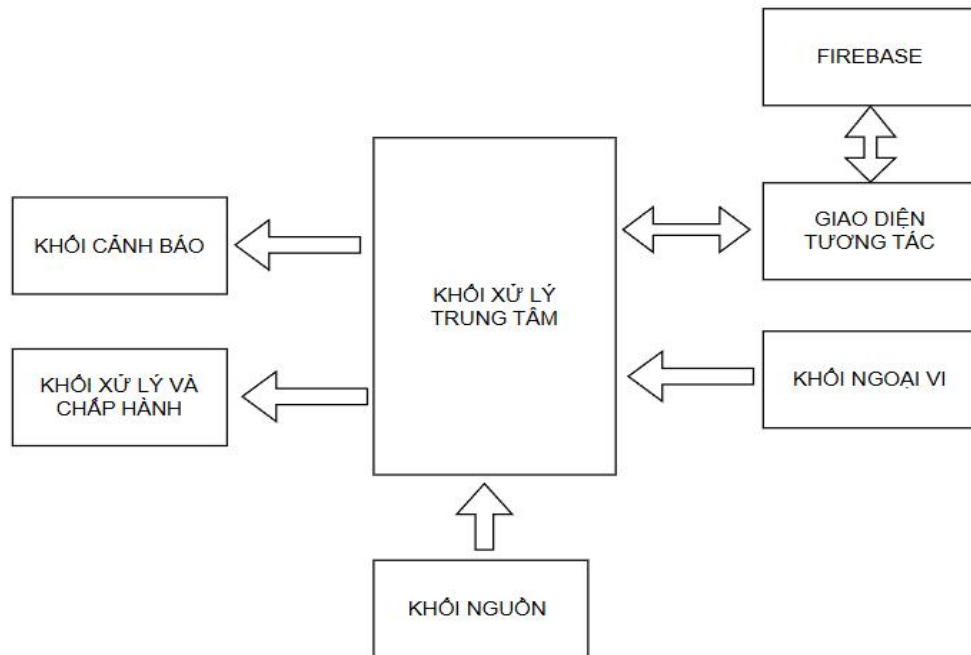
Giao diện sử dụng màu sắc để người dùng dễ nhận biết:

- **Trắng:** giá trị cảm biến (nhiệt độ, độ ẩm, gas).
- **Vàng:** ngưỡng cảnh báo (tempThreshold, gasThreshold).
- **Xanh lá cây:** trạng thái BẬT của thiết bị (quạt, còi).
- **Đỏ:** trạng thái TẮT của thiết bị.
- **Xanh da trời:** trạng thái cửa (Mở/Đóng).
- **Cam:** chế độ hoạt động (Auto / Manual).

3.2. Đặc tả hệ thống

3.2.1. Sơ đồ khối hệ thống

Với những yêu cầu đặt ra như trên của đề tài thì sơ đồ khối của hệ thống sẽ như sau:



Hình 3-1 Sơ đồ khối của hệ thống

Chức năng từng khối:

Khối nguồn: có nhiệm vụ cung cấp điện năng cho toàn bộ hệ thống. Sử dụng nguồn adapter 12V DC làm nguồn đầu vào chính. Điện áp này được chia thành 2 phần:

- Cung cấp trực tiếp 12V cho các thiết bị công suất lớn như quạt DC.
- Hạ áp xuống 5V thông qua mạch ổn áp sử dụng IC 7805 để cấp nguồn cho các khối còn lại: ESP32, LED, cảm biến, nút nhấn và còi.

Nguồn điện đầu vào cần đảm bảo ổn định và liên tục để hệ thống hoạt động chính xác và tin cậy.

Khối xử lý trung tâm: Linh kiện chính là vi điều khiển ESP32 DevKit. Đây là "bộ não" của hệ thống, thực hiện các nhiệm vụ nhận dữ liệu từ các khối

ngoại vi như cảm biến MQ2 (khí gas), DHT11 (nhiệt độ, độ ẩm), và các nút nhấn và xử lý thông tin với các khối khác trong toàn hệ thống. Bộ điều khiển trung tâm sẽ nhận tín hiệu từ cảm biến. Thông qua Hình 3.1 Sơ đồ khối của hệ thống qua chương trình vi điều khiển, tín hiệu kỹ thuật số được xử lý, các tín hiệu sau đó gửi đến các cổng đầu ra như còi, quạt, giao diện hiển thị (TFT) hoặc Firebase.

Khối ngoại vi: có chức năng thu thập dữ liệu đầu vào từ môi trường hoặc người dùng, gồm:

- Cảm biến MQ2: phát hiện khí gas dễ cháy.
- Cảm biến DHT11: đo nhiệt độ và độ ẩm không khí.
- Khối nút nhấn (UP, DOWN, MODE, ON/OFF): cho phép người dùng tương tác và điều chỉnh hệ thống thủ công.

Tất cả tín hiệu đầu vào từ khối này sẽ được gửi đến khối xử lý trung tâm để xử lý.

Khối giao diện tương tác: Có chức năng truyền và nhận dữ liệu, liên kết với wifi để thực hiện việc đẩy dữ liệu nhận được lên cơ sở dữ liệu (Firebase) và đồng thời Nhận lệnh điều khiển từ Firebase (ví dụ bật/tắt quạt, còi, cửa...) để thực hiện điều khiển thiết bị theo yêu cầu người dùng giao diện web.

Khối cảnh báo: Có chức năng nhận dữ liệu từ khối xử lý trung tâm để đi thực hiện các cảnh báo khác nhau tùy thuộc vào dữ liệu nhận được từ khối ngoại vi.

Khối xử lý và chấp hành: Có chức năng nhận dữ liệu từ khối xử lý trung tâm để đi thực hiện các tác vụ xử lý và điều khiển tùy thuộc vào dữ liệu nhận được từ khối ngoại vi hoặc khối giao diện tương tác.

3.2.2. Mô tả hoạt động hệ thống

Hệ thống giám sát và cảnh báo khí gas hoạt động dựa trên quy trình khép kín, trong đó vi điều khiển ESP32 đóng vai trò trung tâm điều khiển, thu thập dữ liệu cảm biến, xử lý logic điều kiện và điều khiển các thiết bị đầu ra. Dữ liệu đồng thời được cập nhật liên tục lên Firebase Realtime Database để phục vụ điều khiển và giám sát từ xa thông qua Web Dashboard.

- **Khởi tạo và kết nối hệ thống**

Khi cấp nguồn, ESP32 khởi động và tiến hành kết nối với mạng Wi-Fi đã cấu hình sẵn. Sau khi kết nối thành công, hệ thống thiết lập liên kết với cơ sở dữ liệu Firebase. Các giá trị cấu hình ban đầu bao gồm chế độ hoạt động (Auto hoặc Manual), ngưỡng cảnh báo nhiệt độ, và ngưỡng khí gas sẽ được đọc từ Firebase và lưu vào các biến hệ thống. Điều này giúp đảm bảo rằng trạng thái vận hành ban đầu được đồng bộ với dữ liệu trên đám mây.

- **Thu thập và xử lý dữ liệu cảm biến**

Sau khi hệ thống khởi tạo hoàn tất, ESP32 bắt đầu thu thập dữ liệu cảm biến theo chu kỳ 1 giây. Cảm biến DHT11 cung cấp giá trị nhiệt độ và độ ẩm môi trường, trong khi cảm biến MQ-2 phát hiện nồng độ khí gas qua tín hiệu analog, được quy đổi thành giá trị ppm tương đối. Các giá trị này được xử lý và lưu trữ tạm thời để hiển thị và đánh giá điều kiện môi trường.

- **Hiển thị thông tin tại chỗ**

Dữ liệu đo được từ các cảm biến sẽ được hiển thị lên màn hình TFT ILI9225 với bố cục rõ ràng, dễ theo dõi. Giao diện màn hình bao gồm: nhiệt độ, độ ẩm, nồng độ khí gas, ngưỡng cảnh báo hiện tại, trạng thái của các thiết bị (quạt, còi, cửa) và chế độ hoạt động hiện tại (Auto hoặc Manual). Người dùng có thể theo dõi toàn bộ thông tin hệ thống tại chỗ mà không cần truy cập ứng dụng web.

- **Cập nhật dữ liệu lên Firebase**

Cùng với quá trình hiển thị tại chỗ, hệ thống đồng thời gửi dữ liệu cảm biến lên Firebase Realtime Database theo cấu trúc JSON. Các giá trị được cập nhật tại nhánh */sensor/*, trong khi ngưỡng cảnh báo và trạng thái thiết bị được đồng bộ tại các nhánh */setting/* và */control/*. Điều này cho phép người dùng theo dõi và điều khiển từ xa thông qua Web Dashboard một cách chính xác và tức thời.

- **Điều khiển thiết bị theo chế độ hoạt động**

Tùy vào chế độ hiện tại, hệ thống sẽ điều khiển thiết bị theo hai phương thức:

- **Chế độ tự động (Auto):**

Hệ thống tự động so sánh giá trị cảm biến với ngưỡng đã cài đặt. Nếu nhiệt độ hoặc nồng độ khí gas vượt ngưỡng an toàn, hệ thống sẽ lập tức bật quạt thông gió, kích hoạt còi cảnh báo và mở cửa servo để giảm nồng độ khí độc và nhiệt. Khi các điều kiện trở lại bình thường, thiết bị sẽ tự động ngừng hoạt động. Mỗi thay đổi về trạng thái thiết bị đều được ghi nhận và cập nhật lên Firebase.

- **Chế độ thủ công (Manual):**

Trong chế độ này, người dùng có thể điều khiển thiết bị hoàn toàn theo ý muốn thông qua nút nhấn vật lý hoặc Web Dashboard. Trạng thái quạt, còi và cửa sẽ được cập nhật theo điều khiển của người dùng và được đồng bộ trở lại Firebase để đảm bảo tính nhất quán giữa thiết bị và hệ thống giám sát từ xa.

- **Chế độ cài đặt (Setting):**

Khi đang ở chế độ Auto, người dùng có thể kích hoạt chế độ cài đặt thông qua nút nhấn ON/OFF. Trong chế độ này, người dùng sử dụng nút MODE để lựa chọn loại ngưỡng cần chỉnh (nhiệt độ hoặc gas), sau đó dùng nút UP/DOWN để tăng hoặc giảm giá trị tương ứng. Các ngưỡng mới sẽ được lưu ngay lập tức lên Firebase, đảm bảo rằng hệ thống luôn sử dụng dữ liệu cấu hình mới nhất.

- **Cập nhật và điều khiển thiết bị đầu ra**

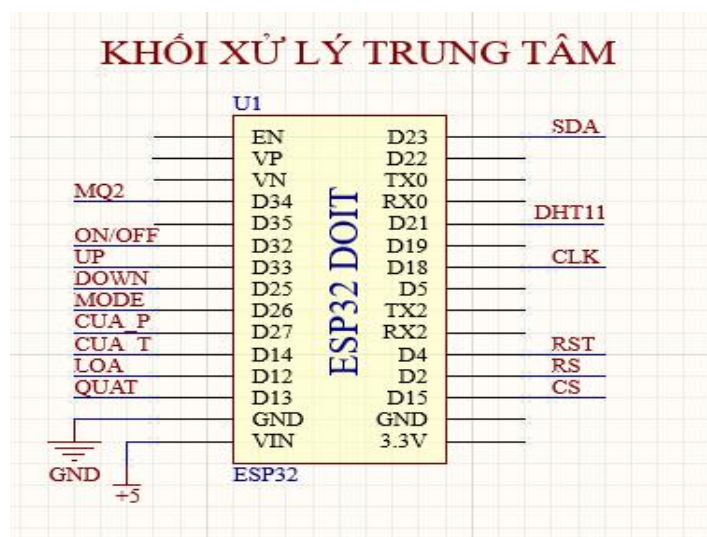
Sau khi xác định trạng thái điều khiển, ESP32 thực hiện thao tác điều khiển thiết bị đầu ra:

- Quạt và còi được bật/tắt thông qua các chân GPIO điều khiển relay.

- Cửa servo được điều khiển bằng tín hiệu PWM để mở hoặc đóng ở các góc phù hợp.
- Trạng thái thực tế của các thiết bị sẽ được phản hồi và cập nhật lên Firebase, đồng thời hiển thị lên màn hình TFT để người dùng theo dõi tức thời.

3.3. Thiết kế phần cứng

3.3.1. Khối xử lý trung tâm (ESP32)



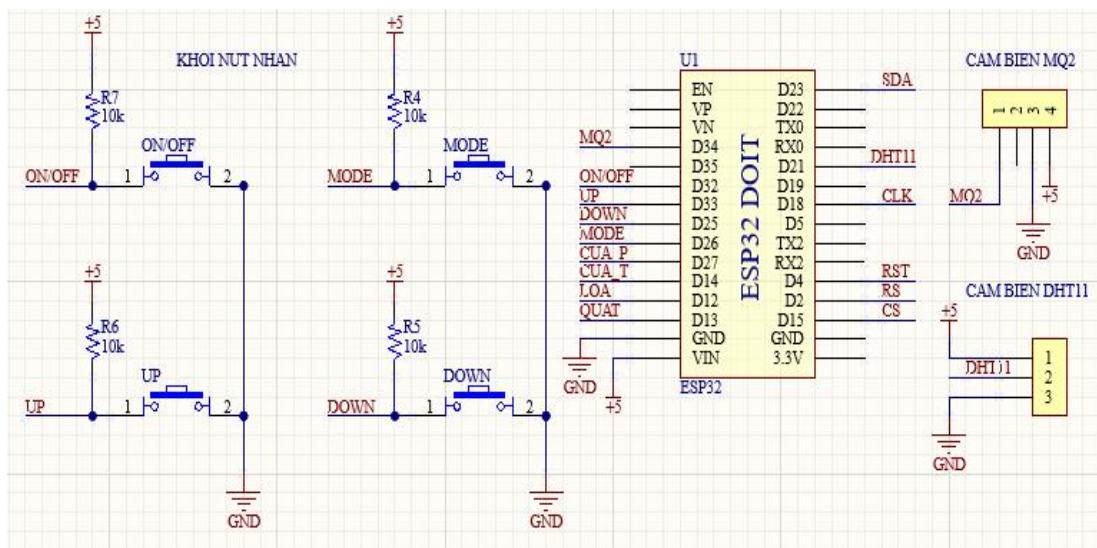
Hình 3-2 Sơ đồ nguyên lý khối xử lý trung tâm

Linh kiện chính của khối này là vi điều khiển ESP32 DevKit, được xem là bộ não của hệ thống. ESP32 có khả năng xử lý đa nhiệm, hỗ trợ Wi-Fi tích hợp sẵn và nhiều chân I/O linh hoạt, rất phù hợp với các ứng dụng IoT như hệ thống cảnh báo rò rỉ khí gas.

Các chân ADC và GPIO của ESP32 được dùng để nhận tín hiệu từ các cảm biến MQ2 và DHT11, cũng như từ các nút nhấn. Đồng thời, vi điều khiển cũng xuất tín hiệu điều khiển đến quạt (qua relay), còi (qua transistor), servo cửa và màn hình TFT. Ngoài ra, thông qua kết nối Wi-Fi, ESP32 gửi và nhận dữ liệu từ Firebase, đồng bộ trạng thái thiết bị và cảm biến lên Web Dashboard.

Khối xử lý trung tâm đảm nhận vai trò phân tích dữ liệu, đưa ra quyết định điều khiển và đảm bảo toàn bộ hệ thống vận hành nhịp nhàng theo logic lập trình..

3.3.2. Khối ngoại vi



Hình 3-3 Sơ đồ kết nối của esp với khối ngoại vi

Khối ngoại vi có nhiệm vụ thu thập dữ liệu đầu vào từ môi trường hoặc thao tác người dùng, từ đó cung cấp thông tin cho khối xử lý trung tâm để phân tích và ra quyết định điều khiển phù hợp. Các thành phần chính trong khối này bao gồm:

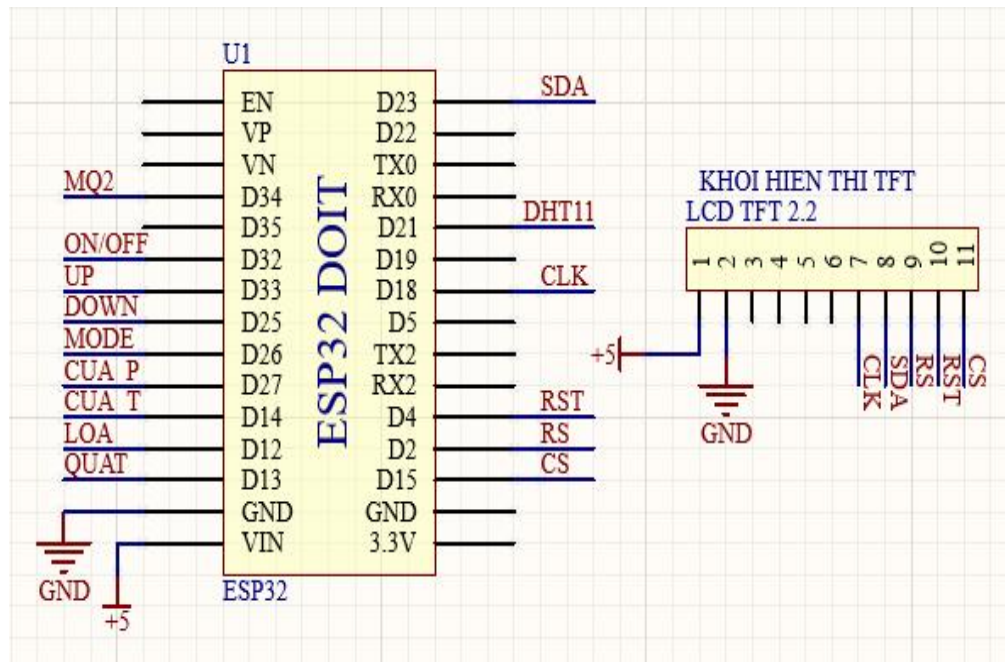
Cảm biến MQ2: phát hiện khí gas dễ cháy như LPG, butan, propane và khói. Tín hiệu đầu ra dạng analog được đưa vào chân ADC của ESP32 để quy đổi thành giá trị ppm tương đối.

Cảm biến DHT11: đo nhiệt độ và độ ẩm không khí trong khu vực giám sát. Cảm biến truyền tín hiệu số thông qua giao tiếp 1-Wire.

Nút nhấn vật lý (MODE, ON/OFF, UP, DOWN): cho phép người dùng điều chỉnh chế độ hoạt động, thay đổi ngưỡng cảnh báo hoặc điều khiển thiết bị thủ công mà không cần truy cập internet.

Tất cả tín hiệu từ khối ngoại vi được đưa về ESP32 để xử lý. Do đó, đây là khối tiếp nhận và truyền thông tin đầu vào quan trọng nhất, giúp hệ thống phản ánh chính xác tình trạng môi trường và nhu cầu người dùng

3.3.3. Khối giao diện tương tác



Hình 3-4 Sơ đồ kết nối esp với khối giao diện tương tác

Khối này đảm nhiệm vai trò trung tâm trong việc kết nối hệ thống với người dùng, cả từ xa qua internet và tại chỗ thông qua giao diện trực quan.

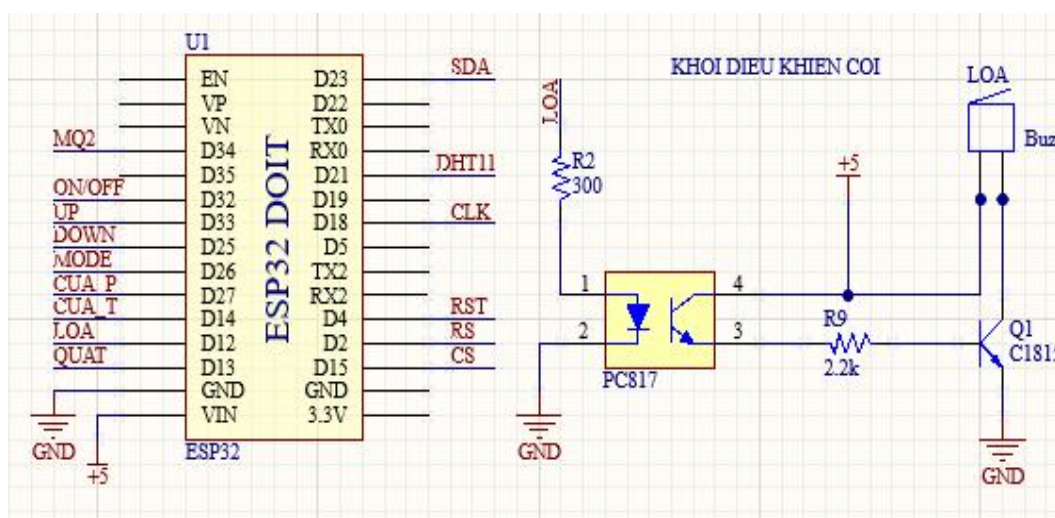
Giao tiếp từ xa qua Firebase: Nhờ tích hợp Wi-Fi trên vi điều khiển ESP32, hệ thống có khả năng kết nối và giao tiếp ổn định với Firebase Realtime Database. Chức năng này bao gồm:

- Cập nhật dữ liệu cảm biến: Các thông số môi trường từ cảm biến (nhiệt độ, độ ẩm, nồng độ khí gas) được gửi liên tục lên Firebase.
- Nhận lệnh điều khiển: Hệ thống tiếp nhận các lệnh điều khiển từ người dùng thông qua Web Dashboard (ứng dụng web trên máy tính hoặc điện thoại).
- Phản hồi điều khiển: Các lệnh điều khiển này được truyền ngay lập tức xuống vi điều khiển ESP32 để thực hiện các hành động điều khiển thực tế trên quạt, còi, và cửa. Sự tích hợp này biến Firebase thành cầu nối mạnh mẽ, cho phép người dùng giám sát và điều khiển hệ thống từ bất kỳ đâu, bất kỳ lúc nào, mang lại sự linh hoạt và tiện lợi tối đa.

Giao diện hiển thị (Màn hình TFT): Khối này sử dụng màn hình TFT 2.2 inch ILI9225, kết nối với ESP32 thông qua chuẩn giao tiếp SPI (các chân SCK, SDA, CS, RS, RST). Màn hình này cung cấp một giao diện trực quan tại chỗ, cho phép người dùng theo dõi tình trạng hệ thống mà không cần thiết bị phụ trợ như máy tính hoặc điện thoại. Cụ thể, màn hình hiển thị:

- Giá trị cảm biến: Hiển thị nhiệt độ, độ ẩm, và nồng độ khí gas hiện tại.
- Trạng thái thiết bị: Thể hiện rõ trạng thái hoạt động của quạt, còi và cửa.
- Chế độ hoạt động và ngưỡng cảnh báo: Cho biết hệ thống đang ở chế độ Tự động (Auto) hay Thủ công (Manual), cùng với các ngưỡng cảnh báo đã thiết lập. Giao diện này đảm bảo người dùng có thể nắm bắt nhanh chóng và trực quan toàn bộ hoạt động của hệ thống ngay tại vị trí lắp đặt.

3.3.4. Khối cảnh báo



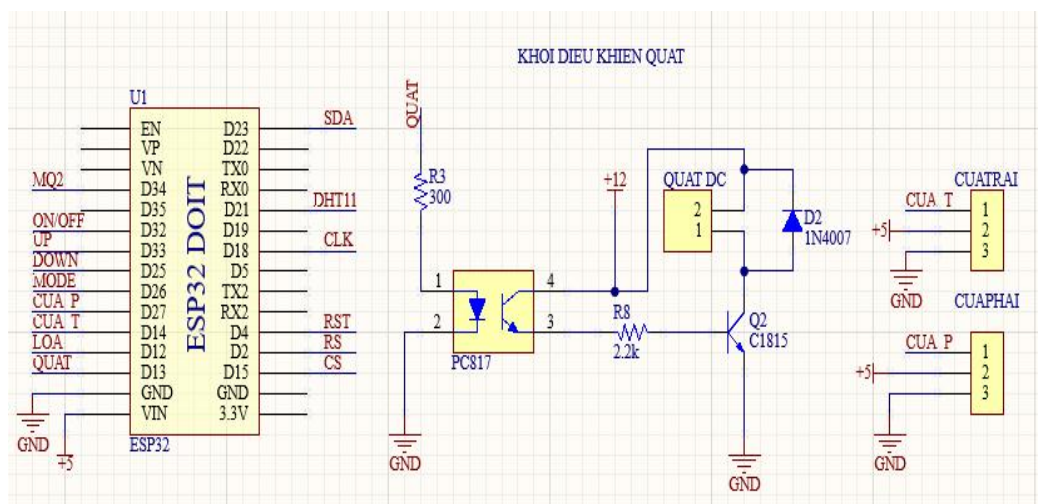
Hình 3-5 Sơ đồ kết nối giữa esp với khối cảnh báo

Khối cảnh báo có nhiệm vụ phát tín hiệu cảnh báo khi hệ thống phát hiện tình huống nguy hiểm. Tín hiệu được kích hoạt khi nhiệt độ hoặc nồng độ khí gas vượt ngưỡng cho phép, thông qua còi báo động.

Còi sẽ được kích hoạt bởi ESP32 thông qua tín hiệu điều khiển mức cao/thấp tại chân digital. Tùy theo mức độ nghiêm trọng của tín hiệu đầu vào, hệ thống có thể bật còi liên tục hoặc theo kiểu nhấp nháy cảnh báo. Đây là khối

quan trọng trong việc đảm bảo an toàn thực tế cho người dùng, giúp cảnh báo sớm các rủi ro xảy ra.

3.3.5. Khối xử lý và chấp hành



Hình 3-6 Sơ đồ kết nối esp với khối xử lý và chấp hành

Khối xử lý và chấp hành có vai trò thực hiện các hành động phản hồi của hệ thống khi có sự cố được phát hiện từ cảm biến, hoặc khi nhận được lệnh điều khiển từ người dùng thông qua Web Dashboard hoặc nút nhấn vật lý. Đây là khối quyết định đầu ra cuối cùng của hệ thống, đảm bảo các thiết bị thực thi hoạt động đúng theo yêu cầu nhằm kịp thời ứng phó với tình huống nguy hiểm.

Trong sơ đồ nguyên lý, khối xử lý và chấp hành bao gồm các thành phần chính như: quạt thông gió, servo điều khiển cửa và còi báo động (liên kết với khối cảnh báo). Tất cả các thiết bị này đều được điều khiển gián tiếp thông qua vi điều khiển ESP32, với sự hỗ trợ của các mạch khuếch đại và cách ly tín hiệu.

Cụ thể, quạt DC được điều khiển thông qua mạch relay có sự hỗ trợ của opto cách ly PC817 và transistor C1815. Điều này giúp cách ly hoàn toàn phần điều khiển mức logic 3.3V của ESP32 với phần tải công suất lớn (12V) của quạt, đồng thời tăng độ an toàn khi vận hành. Khi phát hiện nhiệt độ hoặc khí gas vượt ngưỡng cho phép, ESP32 sẽ kích hoạt transistor làm relay đóng, cho phép dòng điện 12V cấp cho quạt hoạt động. Quạt sẽ giúp làm loãng khí độc hoặc giảm nhiệt độ trong khu vực nguy hiểm.

Với servo điều khiển cửa, ESP32 sẽ xuất tín hiệu PWM (xung điều biến

độ rộng) để điều chỉnh góc quay của các động cơ servo. Thông qua đó, cửa sẽ được mở ra khi có cảnh báo, tạo điều kiện thoát khí hoặc nhiệt ra ngoài. Khi điều kiện môi trường trở lại bình thường, hệ thống sẽ tự động đóng cửa lại.

Ngoài ra, còi báo động được kích hoạt từ tín hiệu điều khiển của ESP32 như đã trình bày ở mục khối cảnh báo. Còi sẽ phát âm thanh cảnh báo rõ ràng giúp người dùng nhanh chóng nhận biết tình huống bất thường.

3.3.6. Sơ đồ nguyên lý & sơ đồ mạch in

Sau khi hoàn thiện việc phân tích chức năng từng khối trong hệ thống, việc thiết kế sơ đồ nguyên lý và mạch in PCB được em thực hiện bằng phần mềm Altium Designer, một công cụ chuyên nghiệp trong thiết kế mạch điện tử. Quá trình này giúp thể hiện rõ ràng cách kết nối giữa các linh kiện với nhau, đồng thời tối ưu bố cục cho việc triển khai mạch thực tế.

❖ Sơ đồ nguyên lý

Sơ đồ nguyên lý (schematic) được xây dựng dựa trên các khối chức năng đã phân tích trước đó. Trong sơ đồ, các linh kiện được chia thành từng khối riêng biệt gồm:

Khối nguồn: chuyển đổi nguồn 12V DC đầu vào xuống 5V ổn định sử dụng IC 7805, kết hợp với tụ lọc và LED báo nguồn.

Khối xử lý trung tâm: vi điều khiển ESP32 DevKit đóng vai trò trung tâm, kết nối với các cảm biến, thiết bị điều khiển và màn hình hiển thị.

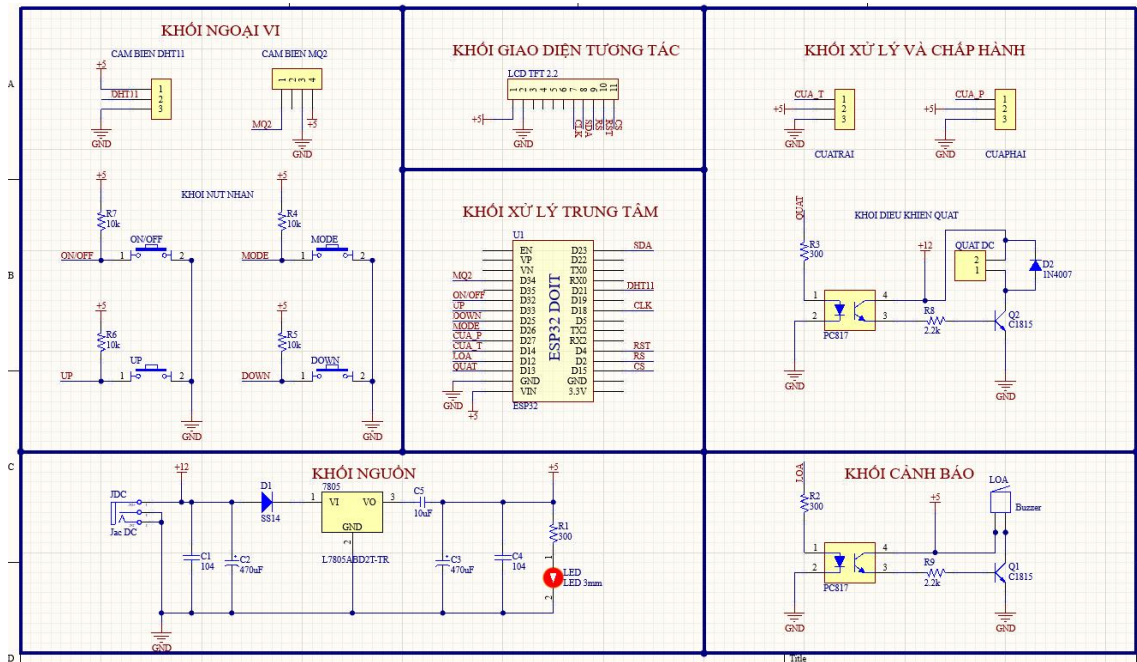
Khối ngoại vi: gồm cảm biến MQ2, DHT11 và các nút nhấn vật lý (MODE, ON/OFF, UP, DOWN).

Khối giao diện tương tác: bao gồm màn hình TFT ILI9225 kết nối SPI với ESP32.

Khối cảnh báo: sử dụng buzzer và transistor điều khiển để phát cảnh báo âm thanh khi có sự cố.

Khối chấp hành: điều khiển quạt và servo cửa, dùng relay và transistor để cách ly tải.

Tổng thể sơ đồ nguyên lý đóng vai trò như bản thiết kế logic của hệ thống, là cơ sở để triển khai sơ đồ mạch in một cách chính xác và khoa học.



Hình 3-7 Sơ đồ nguyên lý toàn hệ thống

❖ Sơ đồ mạch in (PCB)

Sau khi sơ đồ nguyên lý hoàn tất và được kiểm tra logic, quá trình thiết kế mạch in (PCB layout) được thực hiện. Các linh kiện được bố trí hợp lý trên board mạch nhằm đảm bảo:

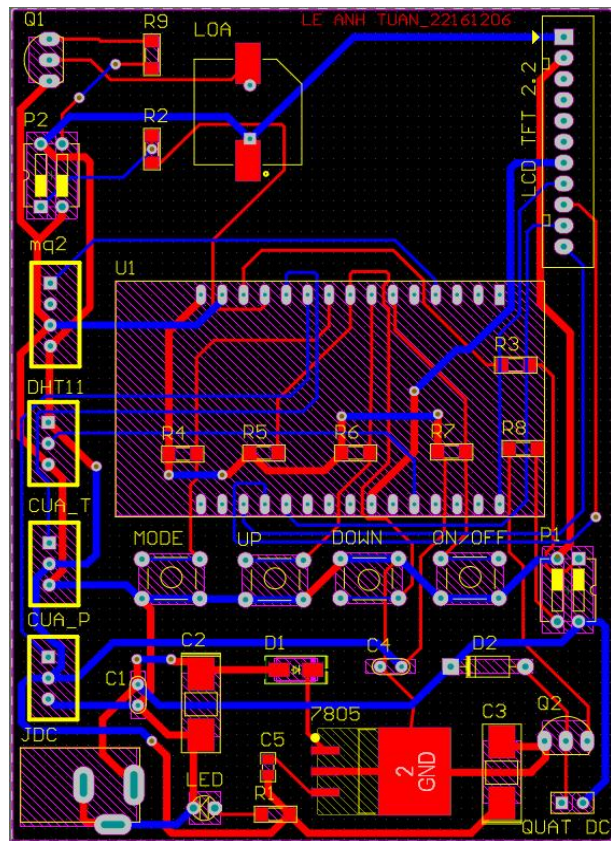
Đường mạch ngắn gọn, giảm thiểu nhiễu tín hiệu và tổn thất điện áp.

Phân tách rõ ràng giữa khối công suất (quạt, relay, còi) và khối tín hiệu điều khiển (ESP32, cảm biến, màn hình).

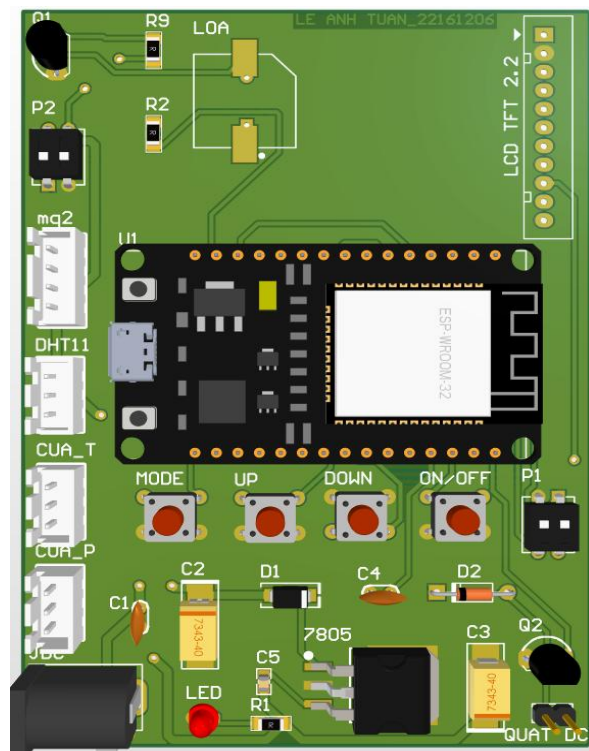
Tối ưu diện tích PCB, đảm bảo mạch gọn gàng và tiết kiệm không gian.

Chân nguồn và mass được đi riêng biệt, đảm bảo dòng ổn định và giảm hiện tượng sụt áp cục bộ.

Mạch in được thiết kế dưới dạng hai lớp (2-layer PCB). Lớp trên (Top Layer) thường được bố trí cho tín hiệu điều khiển, trong khi lớp dưới (Bottom Layer) dành cho các đường nguồn và mass. Các đường tín hiệu công suất như quạt và còi sử dụng đường mạch rộng, trong khi tín hiệu logic sử dụng đường mảnh hơn để đảm bảo tiết kiệm không gian nhưng vẫn đáp ứng tiêu chuẩn kỹ thuật.



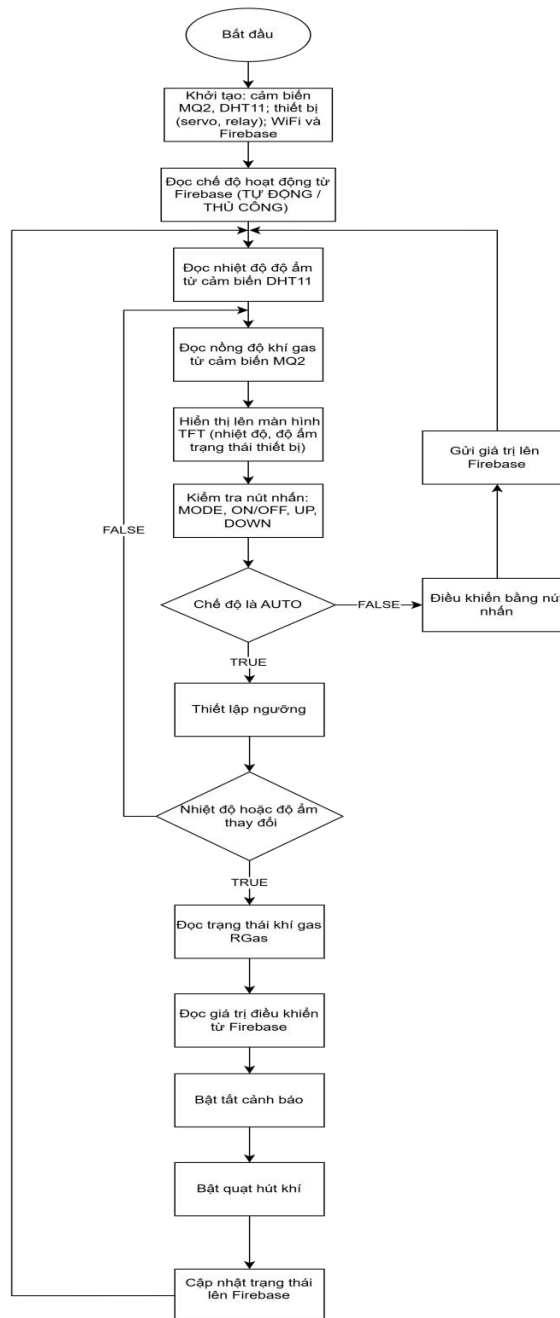
Hình 3-8 Board mạch PCB hoàn thiện



Hình 3-9 Hình 3d mạch PCB hoàn thiện

3.4. Thiết kế phần mềm

3.4.1. Lưu đồ giải thuật chương trình chính



Hình 3-10 Lưu đồ chương trình chính

Giải thích lưu đồ:

Khi mới cấp nguồn, chương trình sẽ thực hiện việc khởi tạo các cổng (port) kết nối với các thiết bị cảm biến và điều khiển (như cảm biến MQ2,

DHT11, servo, relay). Tiếp theo, hệ thống kết nối đến mạng WiFi và Firebase bằng đường dẫn và mật khẩu đã được cấu hình sẵn.

Sau đó, chương trình bắt đầu đọc dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT11. Nếu có sự thay đổi về giá trị nhiệt độ hoặc độ ẩm so với lần đọc trước, chương trình sẽ gửi dữ liệu cập nhật lên Firebase để lưu trữ và theo dõi.

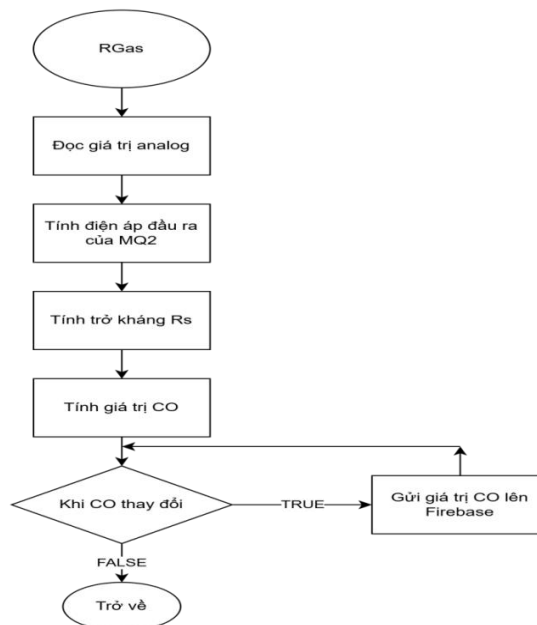
Song song với đó, chương trình liên tục đọc nồng độ khí gas từ cảm biến MQ2. Dựa vào mức khí gas đo được, chương trình sẽ lấy các giá trị điều khiển từ Firebase để điều khiển các thiết bị như còi báo động, quạt hút khí, cửa servo theo các mức cảnh báo được thiết lập.

Đặc biệt, quạt hút khí sẽ được bật và điều chỉnh tốc độ dựa trên nồng độ khí gas hiện tại trong không khí để xử lý rò rỉ hiệu quả hơn. Nếu nồng độ khí gas vượt mức ngưỡng an toàn, chương trình sẽ gửi email cảnh báo khẩn cấp đến người dùng để kịp thời xử lý.

Chương trình sẽ chạy liên tục trong vòng lặp, đọc cảm biến, cập nhật dữ liệu và điều khiển thiết bị một cách tự động hoặc theo lệnh thủ công từ Firebase.

3.4.2. Lưu đồ chương trình con

❖ Chương trình đọc khí gas

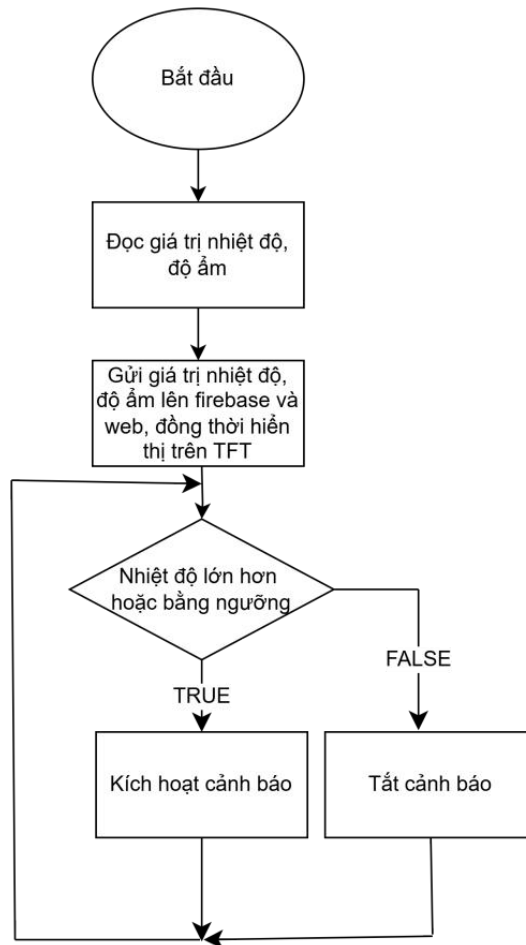


Hình 3-11 Lưu đồ chương trình đọc khí gas

Giải thích lưu đồ:

Chương trình sẽ tiến hành đọc giá trị analog từ chân A0 sau đó tính mức điện áp đầu ra của cảm biến. Tính toán giá trị của trở kháng thay đổi từ đó xác định được mức khí gas trong môi trường hiện tại. Đồng thời gửi giá trị lên firebase khi có sự thay đổi về lượng khí gas.

❖ Chương trình đọc nhiệt độ, độ ẩm

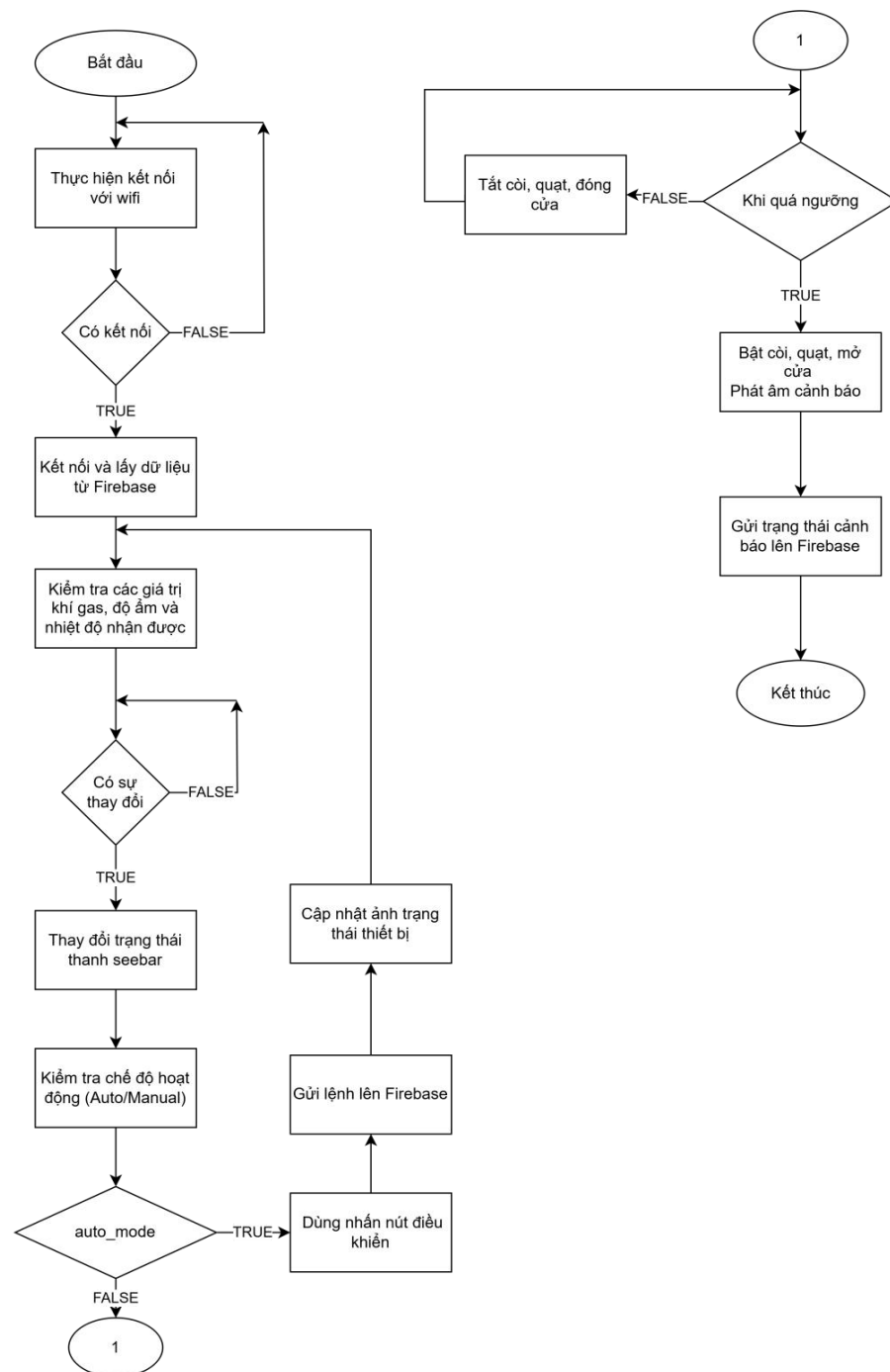


Hình 3-12 Lưu đồ chương trình đọc nhiệt độ, độ ẩm

Giải thích lưu đồ:

Chương trình tiến hành đọc giá trị nhiệt độ và độ ẩm từ cảm biến DHT11. Sau đó, dữ liệu được gửi lên Firebase và hiển thị trên màn hình TFT. Hệ thống so sánh nhiệt độ hiện tại với ngưỡng đã cài đặt: nếu nhiệt độ vượt ngưỡng, sẽ kích hoạt cảnh báo; ngược lại, cảnh báo sẽ được tắt.

❖ Lưu đồ giải thuật của Web Dashboard



Hình 3-13 Lưu đồ chương trình Web Dashboard

Giải thích lưu đồ:

Đầu tiên hệ thống sẽ tiến hành kết nối với Wifi. Nếu kết nối thành công, hệ thống sẽ kết nối đến Firebase và tiến hành lấy dữ liệu ngưỡng cảnh báo (bao gồm ngưỡng khí gas, độ ẩm và nhiệt độ) được cài đặt sẵn từ Firebase.

Tiếp theo, hệ thống đọc các giá trị cảm biến gồm: khí gas, nhiệt độ và độ ẩm. Những giá trị này sẽ được so sánh với ngưỡng đã cài đặt để xác định có xảy ra tình huống cảnh báo hay không. Nếu có sự thay đổi trong giá trị cảm biến, hệ thống sẽ cập nhật thanh trạng thái seebar và thay đổi ảnh trạng thái thiết bị trên giao diện web nhằm phản ánh đúng tình hình thực tế.

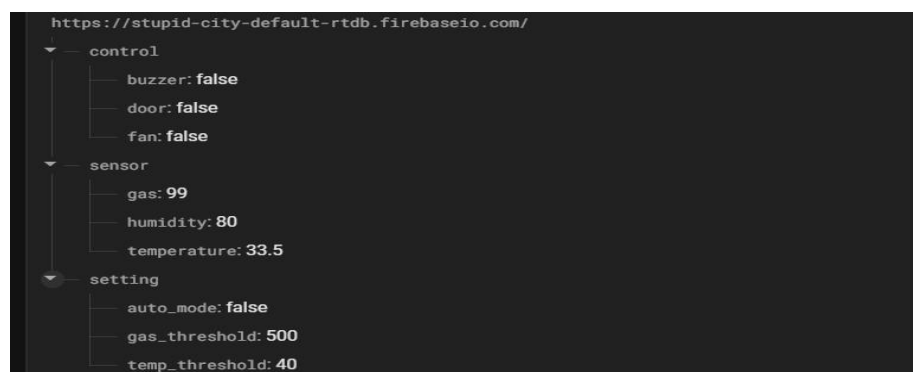
Sau đó, hệ thống sẽ kiểm tra chế độ hoạt động của thiết bị là tự động (Auto) hay thủ công (Manual). Nếu ở chế độ thủ công, người dùng sẽ sử dụng nút điều khiển để gửi lệnh lên Firebase, từ đó điều khiển các thiết bị như còi, quạt, cửa... và trạng thái tương ứng cũng sẽ được cập nhật lại trên Firebase.

Ngược lại, nếu hệ thống đang ở chế độ tự động, nó sẽ tự động kiểm tra giá trị cảm biến có vượt ngưỡng cảnh báo hay không. Nếu không vượt ngưỡng, hệ thống sẽ tắt còi, tắt quạt, đóng cửa. Ngược lại, nếu vượt ngưỡng, hệ thống sẽ bật còi, bật quạt, mở cửa và phát âm thanh cảnh báo, đồng thời gửi trạng thái cảnh báo lên Firebase.

3.4.3. Cấu trúc dữ liệu Firebase

Hệ thống sử dụng Firebase Realtime Database để thực hiện việc lưu trữ dữ liệu cảm biến, trạng thái thiết bị và các thông số cấu hình. Việc tổ chức cấu trúc dữ liệu một cách hợp lý và phân tầng rõ ràng giúp hệ thống dễ dàng mở rộng, truy xuất nhanh chóng và đồng bộ hiệu quả giữa vi điều khiển ESP32 và giao diện Web Dashboard.

Cây dữ liệu trong Firebase được tổ chức theo cấu trúc phân cấp như sau:



Hình 3-14 Cây dữ liệu Firebase

Giải thích chức năng từng nhánh:

- */sensor/*: Chứa các dữ liệu cảm biến được cập nhật liên tục từ ESP32. Dữ liệu tại đây phục vụ cho việc hiển thị trên Web Dashboard và lưu vết lịch sử hoạt động.
 - *temperature*: nhiệt độ đo được từ DHT11.
 - *humidity*: độ ẩm đo từ DHT11.
 - *gas*: nồng độ khí gas tính bằng ppm từ MQ2.
- */control/*: Lưu trạng thái hiện tại của các thiết bị điều khiển. Dữ liệu ở nhánh này có thể được cập nhật từ Web Dashboard hoặc từ chế độ Auto của ESP32, và cũng có thể được đọc lại bởi ESP32 để điều khiển thiết bị.
 - *fan*: trạng thái quạt thông gió.
 - *buzzer*: trạng thái còi báo.
 - *door*: trạng thái cửa servo.
- */setting/*: Lưu các cấu hình quan trọng ảnh hưởng đến logic điều khiển của hệ thống. Người dùng có thể thay đổi các giá trị này từ Web Dashboard hoặc trực tiếp bằng nút nhấn vật lý.
 - *auto_mode*: xác định hệ thống đang ở chế độ Auto (true) hay Manual (false).
 - *gas_threshold*: ngưỡng nồng độ khí gas cho phép trước khi hệ thống kích hoạt cảnh báo.
 - *temp_threshold*: ngưỡng nhiệt độ cảnh báo.

Tính năng đồng bộ thời gian thực

Firestore Realtime Database hỗ trợ cập nhật dữ liệu theo thời gian thực, giúp:

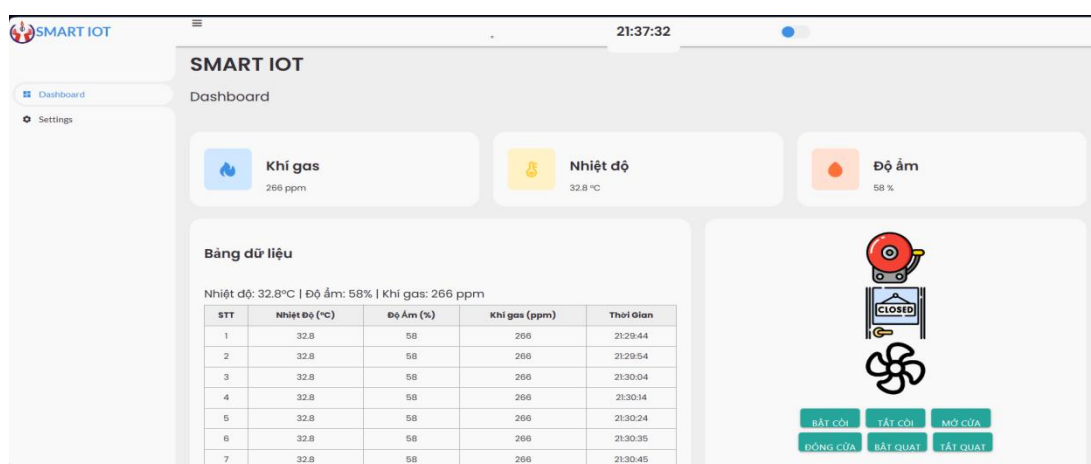
- Web Dashboard nhận được thông tin mới ngay khi ESP32 ghi dữ liệu.
- ESP32 cập nhật trạng thái thiết bị lập tức khi người dùng thao tác từ xa.
- Mọi thay đổi đều được ghi nhận ngay lập tức trên cả hai phía: thiết bị và người dùng.

Việc thiết kế cấu trúc cây như trên vừa đảm bảo sự phân tách chức năng rõ ràng, vừa giúp truy xuất dữ liệu nhanh, dễ quản lý và thuận tiện khi mở rộng hệ thống.

3.4.4. Giao diện phần mềm Web Dashboard

Bên cạnh giao tiếp tại chỗ qua màn hình TFT và nút nhấn vật lý, hệ thống còn được trang bị một giao diện Web Dashboard giúp người dùng có thể giám sát và điều khiển thiết bị từ xa một cách trực quan, tiện lợi và theo thời gian thực. Giao diện Web này được xây dựng bằng các công nghệ phổ biến như HTML, CSS và JavaScript, sử dụng thư viện Firebase SDK để giao tiếp trực tiếp với Realtime Database.

❖ Giao diện và bố cục



Hình 3-15 Giao diện Web Dashboard

Giao diện Website được mô tả như hình. Giao diện đầu tiên là giao diện giám sát với các khung hiển thị giá trị đẹp mắt. Thanh Taskbar được bố trí bên góc trái giúp chọn lựa các chế độ được nhanh chóng và thuận tiện hơn.

Ngoài ra, phần giao diện này còn hiển thị đồng hồ thời gian thực ở vị trí

trung tâm trên thanh điều hướng, hỗ trợ người dùng theo dõi thời gian hệ thống hoạt động.

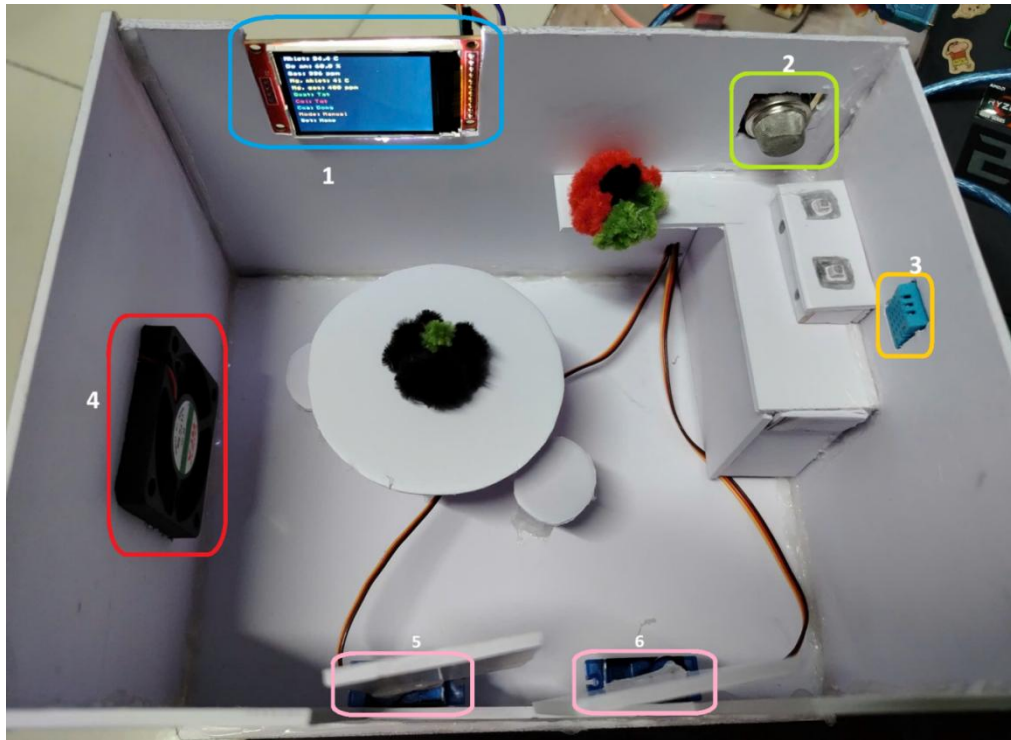
Giao diện Dashboard SMART IOT được thiết kế trực quan với tông màu sáng, bao gồm thanh điều hướng bên trái chứa các mục Dashboard và Settings. Phần chính giữa hiển thị tiêu đề SMART IOT cùng đồng hồ thời gian thực. Ba thẻ hiển thị trạng thái cảm biến cập nhật liên tục các giá trị môi trường gồm: khí gas (ppm), nhiệt độ (°C) và độ ẩm (%). Dưới đó là một bảng dữ liệu lớn, ghi lại các giá trị cảm biến theo từng thời điểm, giúp người dùng theo dõi và đánh giá diễn biến môi trường.

❖ Chức năng chính của Web Dashboard

- Theo dõi dữ liệu cảm biến theo thời gian thực:
 - ❖ Người dùng có thể giám sát các thông số gồm:
 - ❖ Nồng độ khí gas (ppm)
 - ❖ Nhiệt độ môi trường (°C)
 - ❖ Độ ẩm không khí (%)
- Trạng thái của các thiết bị: hiển thị tình trạng hoạt động của quạt, còi báo động, cửa, đèn.
- Cảnh báo hệ thống (âm thanh/hình ảnh): Cảnh báo bằng âm thanh và/hoặc hình ảnh được kích hoạt khi giá trị cảm biến vượt quá ngưỡng an toàn.
- Lựa chọn chế độ điều khiển:
 - ❖ *Chế độ Tự động (Auto Mode)*: Hệ thống tự động điều chỉnh thiết bị dựa trên giá trị cảm biến khi vượt ngưỡng.
 - ❖ *Chế độ Thủ công (Manual Mode)*: Người dùng có thể chủ động bật/tắt thiết bị bằng nút điều khiển trên Dashboard.
- Thay đổi ngưỡng cảnh báo: người dùng nhập giá trị mới cho nồng độ khí gas, nhiệt độ hoặc độ ẩm. Hệ thống đẩy dữ liệu lên Firebase để cập nhật tức thời.

- Gửi lệnh điều khiển thiết bị: khi người dùng nhấn nút bật/tắt trên Dashboard, lệnh sẽ được gửi đến Firebase. Vi điều khiển ESP32 đọc lệnh và thực hiện thao tác tương ứng.

Đóng gói mô hình hệ thống: nhóm sử dụng chất liệu giấy form để hoàn thiện mô hình phần cứng của hệ thống. Đây là loại vật liệu có tính chất xốp nhưng không kém phần cứng cáp, dễ dàng cho việc cắt ghép và tạo hình, giúp mô phỏng bố trí các linh kiện một cách trực quan, gọn gàng và dễ lắp ráp.



Hình 4-2 Mô hình hệ thống

Các vị trí linh kiện trên mô hình:

Số 1: Vị trí của màn hình hiển thị TFT ILI9225, dùng để hiển thị thông số cảm biến và trạng thái thiết bị.

Số 2: Vị trí của cảm biến MQ-2 – phát hiện nồng độ khí gas dễ cháy.

Số 3: Vị trí của cảm biến DHT11 – đo nhiệt độ và độ ẩm môi trường.

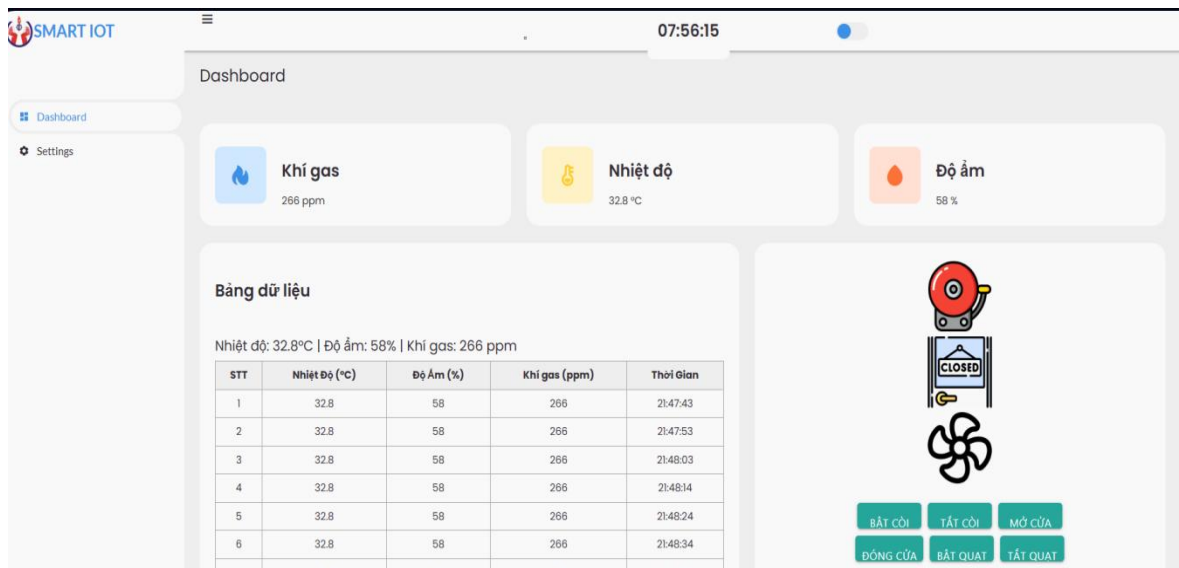
Số 4: Vị trí quạt – thiết bị giúp làm loãng khí gas khi phát hiện rò rỉ.

Số 5: Cửa phải – mô phỏng đóng/mở bằng servo.

Số 6: Cửa trái – đồng bộ hoạt động cùng cửa phải để tăng tính thực tiễn.

4.2. KẾT QUẢ GIAO DIỆN WEB

4.2.1. Giao diện trang trình

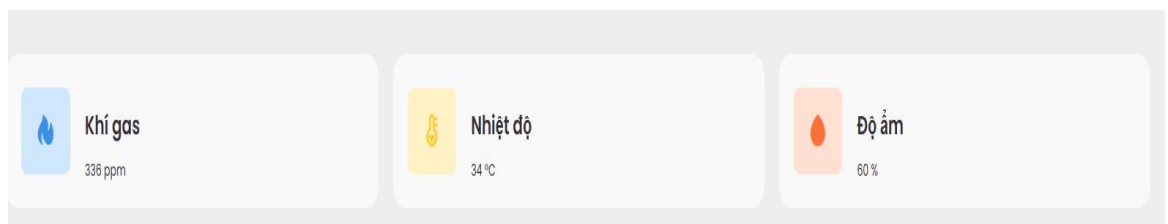


Hình 4-3 Giao diện web hiển thị

Giao diện trang chính (Dashboard) là nơi người dùng có thể giám sát thời gian thực các thông số môi trường và trạng thái hoạt động của thiết bị. Giao diện được xây dựng bằng HTML, CSS, JavaScript và kết nối với Firebase để hiển thị dữ liệu mới nhất.

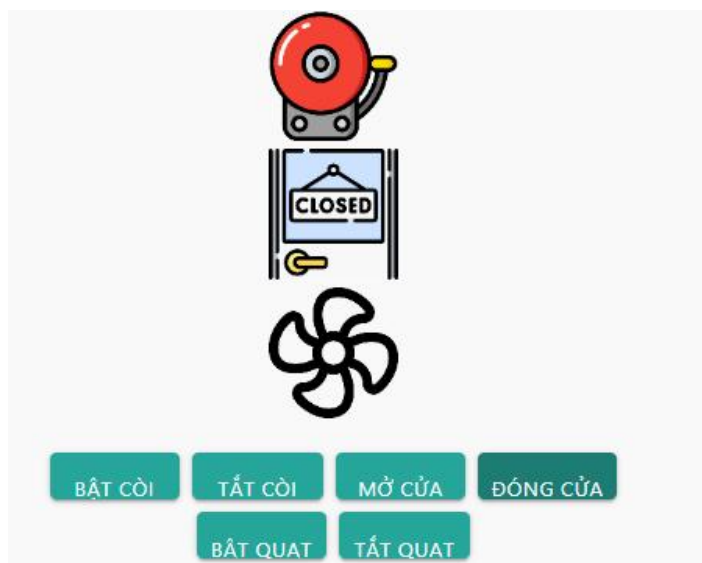
Các thành phần chính bao gồm:

- Ba thẻ thông tin cảm biến: hiển thị liên tục giá trị nồng độ khí gas (ppm), nhiệt độ (°C), độ ẩm (%).



Hình 4-4 Bảng theo dõi dữ liệu

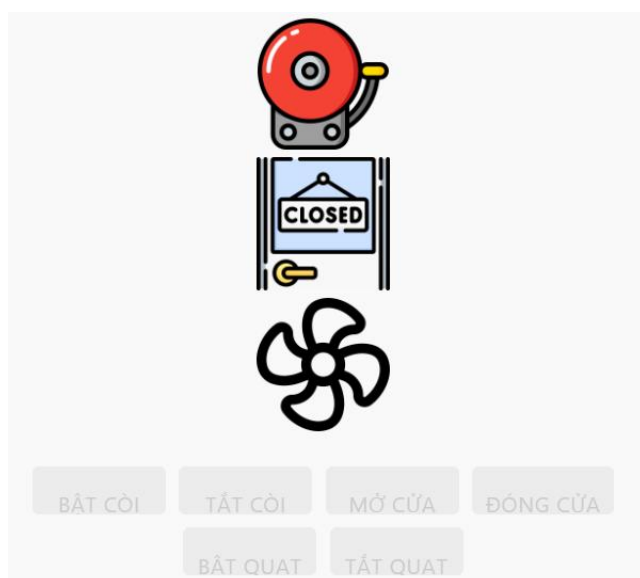
- Bảng trạng thái thiết bị: hiển thị tình trạng bật/tắt của quạt, còi báo động, cửa.



Hình 4-5 Bảng trạng thái các thiết bị khi ở chế độ manual

Hình ảnh thể hiện hệ thống đang hoạt động ở chế độ Manual, cho phép người dùng điều khiển thủ công thiết bị thông qua các nút .

Các nút lệnh được hiển thị màu xanh dương, cho thấy chúng đang sẵn sàng thao tác. Biểu tượng trực quan của còi, cửa và quạt giúp người dùng nhận biết nhanh chức năng tương ứng.



Hình 4-6 Bảng trạng thái các thiết bị khi ở chế độ auto

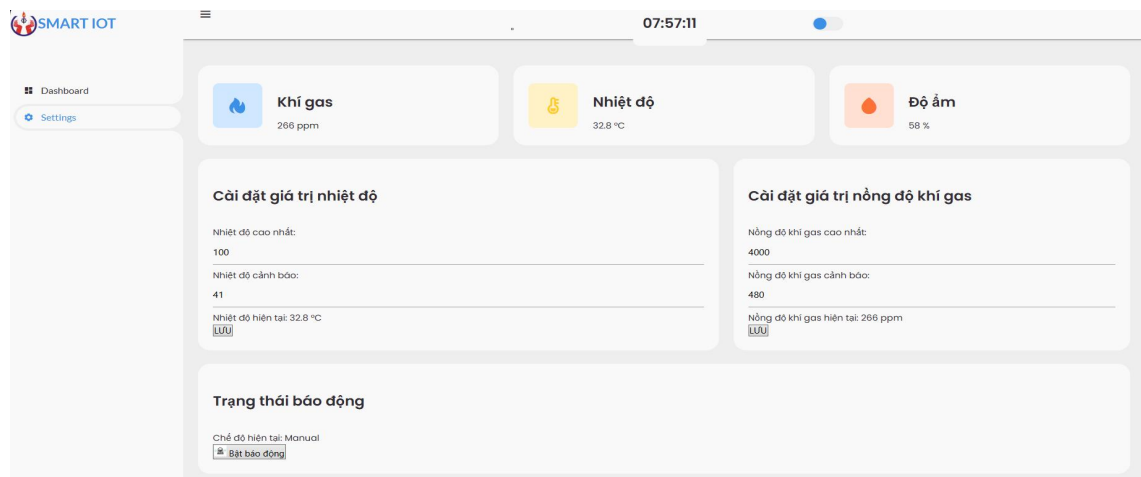
Hình ảnh cho thấy hệ thống đang ở chế độ Auto (tự động). Trong chế độ này:

- Các thiết bị sẽ tự động bật/tắt dựa trên giá trị cảm biến và ngưỡng cảnh báo đã cài đặt.
- Tất cả nút điều khiển thủ công đều bị mờ (disabled), nhằm tránh việc người dùng can thiệp khi hệ thống đang vận hành tự động.

Cách trình bày trực quan này giúp người dùng dễ dàng phân biệt và tránh thao tác sai chế độ điều khiển.

- Thanh điều hướng: chứa các mục chuyển đổi giữa Dashboard và Settings.
- Đồng hồ thời gian thực: hiển thị ở phần trung tâm để theo dõi thời gian hoạt động.
- Hệ thống phản hồi thời gian thực: mọi thay đổi (cảm biến, trạng thái thiết bị, chế độ điều khiển) đều đồng bộ ngay lập tức với Firebase và cập nhật lại trên giao diện

4.2.2. Giao diện trang cài đặt


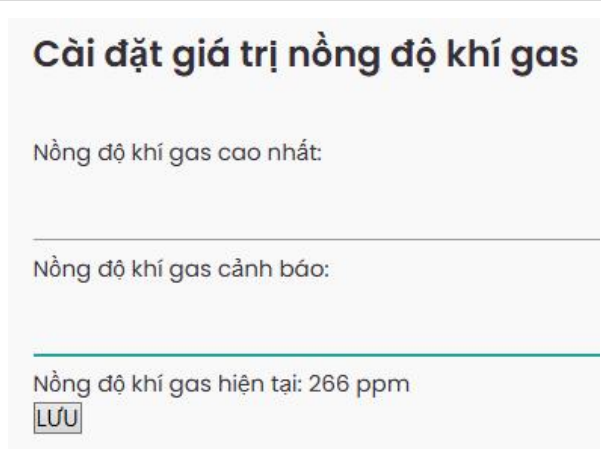


Hình 4-7 Giao diện web cài đặt

Giao diện trang Setting cho phép người dùng tùy chỉnh ngưỡng cảnh báo và giám sát chế độ hoạt động của hệ thống. Đây là nơi cấu hình quan trọng giúp hệ thống hoạt động linh hoạt theo môi trường thực tế.

❖ Thay đổi ngưỡng cảnh báo:



Người dùng có thể nhập giá trị mới cho nồng độ khí gas, nhiệt độ, hoặc độ ẩm thông qua các ô nhập liệu trực tiếp trên giao diện web. Sau khi nhập, hệ thống sẽ gửi giá trị mới lên Firebase Realtime Database để cập nhật ngay lập tức, đảm bảo các thiết bị hoạt động theo ngưỡng mới được thiết lập. Điều này giúp tăng tính linh hoạt, cho phép người dùng dễ dàng điều chỉnh theo từng môi trường cụ thể.

<p>Cài đặt giá trị nhiệt độ</p> <ul style="list-style-type: none"> ❖ Nhiệt độ cao nhất: --°C ❖ Ngưỡng cảnh báo: --°C ❖ Nhiệt độ hiện tại: --°C ❖ Có nút "CHỈNH SỬA" → cho phép người dùng thay đổi ngưỡng cảnh báo. 	<p>Cài đặt giá trị nồng độ khí gas</p> <ul style="list-style-type: none"> ❖ Nồng độ cao nhất: -- ppm ❖ Ngưỡng cảnh báo: -- ppm ❖ Nồng độ hiện tại: -- ppm ❖ Có nút "CHỈNH SỬA" → cho phép người dùng thay đổi ngưỡng cảnh báo.
	

BẢNG 4-1 Cài đặt ngưỡng cảnh báo

Sau khi xác nhận, hệ thống sẽ gửi dữ liệu lên Firebase Realtime Database để cập nhật ngay lập tức. Các thiết bị sẽ vận hành theo ngưỡng mới mà không cần reset lại hệ thống.

- ❖ **Chế độ điều khiển:** hiển thị và chuyển đổi giữa chế độ Auto (tự động điều khiển khi vượt ngưỡng) và Manual (người dùng điều khiển thủ công).

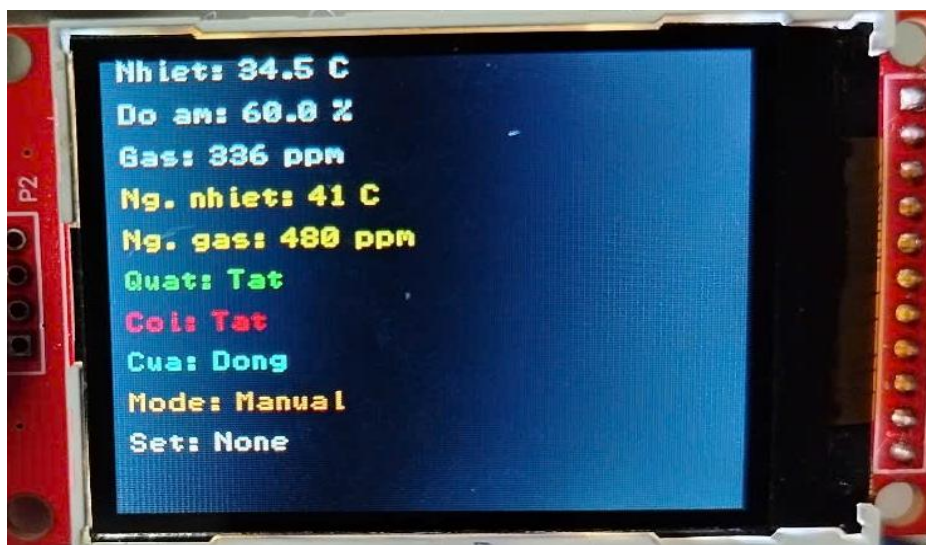
Auto (Tự động): Hệ thống tự động bật/tắt thiết bị dựa trên giá trị cảm biến.	Manual (Thủ công): Người dùng có thể điều khiển quạt, còi báo động, cửa (servo) bằng nút bấm trên giao diện.
<p>Trạng thái báo động</p> <p>Chế độ hiện tại: Auto</p> <p> Tắt báo động</p>	<p>Trạng thái báo động</p> <p>Chế độ hiện tại: Manual</p> <p> Bật báo động</p>

BẢNG 4-2 Chế độ điều khiển

4.3 KIỂM TRA VÀ ĐÁNH GIÁ CHỨC NĂNG CHÍNH

4.3.1. Đọc và hiển thị dữ liệu cảm biến trên màn hình

Hệ thống sử dụng cảm biến MQ-2 để đo nồng độ khí gas, và DHT11 để đo nhiệt độ, độ ẩm. Dữ liệu được đọc mỗi 1 giây (delay 1000 ms), xử lý và hiển thị ngay lập tức lên màn hình TFT ILI9225.



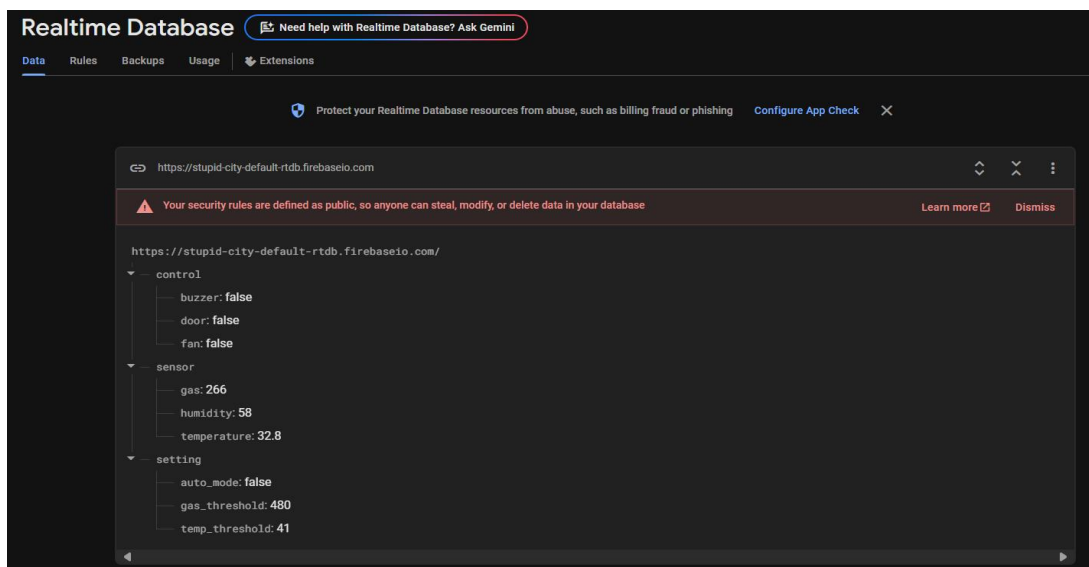
Hình 4-8 Kết quả hiển thị trên màn hình TFT

Trên màn hình TFT, các thông tin được trình bày rõ ràng:

- Giá trị cảm biến: nhiệt độ, độ ẩm, nồng độ khí gas (ppm).
- Ngưỡng cảnh báo: nhiệt độ/gas do người dùng thiết lập.
- Trạng thái thiết bị: quạt, còi, cửa (BẬT/TẮT - MỞ/ĐÓNG).
- Chế độ hoạt động: Auto / Manual.
- Trạng thái cài đặt: Setting hoặc Normal.

Việc cập nhật định kỳ và sử dụng màu sắc trực quan giúp người dùng dễ dàng theo dõi tình trạng hệ thống tại chỗ mà không cần truy cập internet.

4.3.2 Gửi dữ liệu lên Firebase



Hình 4-9 Giao diện firebase

Song song với việc hiển thị tại chỗ, hệ thống gửi dữ liệu cảm biến lên Firebase Realtime Database dưới dạng cây JSON. Cụ thể:

- /sensor/: chứa giá trị nhiệt độ, độ ẩm, khí gas.
- /setting/: chứa ngưỡng cảnh báo.
- /control/: chứa trạng thái thiết bị như quạt, còi, cửa.

Firebase cho phép đồng bộ dữ liệu theo thời gian thực, đảm bảo mọi thay đổi từ ESP32 hay từ giao diện Web đều được cập nhật và phản hồi tức thì. Dữ liệu được dùng để:

- Theo dõi từ xa qua Web Dashboard.
- Lưu vết hoạt động để kiểm tra sau.
- Điều kiện cho các tác vụ điều khiển tự động

4.3.3 Điều khiển thiết bị từ Firebase và nút nhấn

Hệ thống hỗ trợ điều khiển thiết bị thông qua:

Web Dashboard: người dùng nhấn các nút điều khiển, lệnh sẽ gửi lên Firebase. ESP32 đọc lệnh và thực hiện tương ứng như bật/tắt còi, quạt, mở/đóng cửa.

Nút nhấn vật lý: sử dụng các nút MODE, UP, DOWN, ON/OFF để điều khiển hệ thống kể cả khi không có mạng.

Ba chế độ hoạt động:

- Auto: hệ thống tự bật/tắt thiết bị khi giá trị cảm biến vượt ngưỡng.
- Manual: người dùng điều khiển chủ động, trạng thái được đồng bộ lại Firebase.
- Setting: cho phép chỉnh sửa ngưỡng cảnh báo ngay trên thiết bị.

Việc hỗ trợ song song cả điều khiển từ xa và trực tiếp giúp tăng độ tin cậy, linh hoạt và an toàn trong vận hành hệ thống

4.4. ĐÁNH GIÁ KẾT QUẢ

4.4.1. Độ chính xác của cảm biến

Đánh giá độ chính xác của các cảm biến được sử dụng là yếu tố then chốt để xác định độ tin cậy của dữ liệu đầu vào.

- **Cảm biến nhiệt độ và độ ẩm DHT11:**

Thông số kỹ thuật: Cảm biến DHT11 có dải đo nhiệt độ từ 0 – 50 °C với sai số lý thuyết là ± 2 °C, và dải đo độ ẩm từ 20 – 90% RH với sai số lý thuyết là $\pm 5\%$.

Đánh giá thực tế: Khi so sánh dữ liệu thu thập từ DHT11 với một thiết bị đo chuyên dụng (nhiệt ẩm kế điện tử), độ chênh lệch trung bình ghi nhận được là khoảng ± 1.5 °C đối với nhiệt độ và $\pm 3\%$ RH đối với độ ẩm. Mức sai số này được đánh giá là chấp nhận được và phù hợp cho các ứng dụng dân dụng và IoT phổ thông, nơi không yêu cầu độ chính xác quá cao chuẩn phòng thí nghiệm.

- **Cảm biến khí gas MQ-2:**

Đặc điểm: Cảm biến MQ-2 là một cảm biến analog, đặc trưng bởi độ chính xác phụ thuộc đáng kể vào thời gian làm nóng ban đầu và quá trình hiệu chuẩn chi tiết.

Đánh giá thực tế: Cảm biến MQ-2 yêu cầu một khoảng thời gian làm nóng từ 30 đến 60 giây để tín hiệu đầu ra ổn định. Cảm biến này thể hiện độ nhạy cao với các loại khí dễ cháy như LPG, methane và khói. Tuy nhiên, nó không cung cấp khả năng đo nồng độ tuyệt đối chính xác (giá trị ppm chỉ mang tính tương đối và ước lượng). Trong môi trường kiểm thử thực tế, khi đưa nguồn khí gần cảm biến, giá trị đọc được tăng lên tức thời và ổn định nhanh chóng, cho thấy khả năng phản ứng nhanh và ổn định của cảm biến trong việc phát hiện sự hiện diện của khí gas.

Hệ thống đạt yêu cầu về độ chính xác để phát hiện kịp thời các tình trạng môi trường bất thường và phát cảnh báo, hoàn toàn phù hợp với mục tiêu giám sát và điều khiển cơ bản của đề tài.

4.4.2. Độ trễ cập nhật dữ liệu trên Firebase

Khả năng đồng bộ dữ liệu theo thời gian thực là một yếu tố then chốt đối với hiệu quả của một hệ thống IoT, đặc biệt là trong các ứng dụng cảnh báo.

- **Cơ chế cập nhật dữ liệu:**

Vi điều khiển ESP32 được lập trình để thu thập dữ liệu từ các cảm biến mỗi 1 giây một lần.

Ngay sau khi thu thập, dữ liệu được gửi tức thời lên Firebase Realtime Database tại các nhánh tương ứng như */sensor/* (dữ liệu cảm biến), */control/* (trạng thái thiết bị điều khiển), và */setting/* (cài đặt ngưỡng). Firebase Realtime Database có cơ chế đồng bộ dữ liệu hai chiều mạnh mẽ, đảm bảo thông tin luôn được cập nhật liên tục và tức thời giữa các client kết nối.

- **Kiểm thử và đánh giá độ trễ:**

Độ trễ từ ESP32 đến Web Dashboard: Khi đo thời gian từ lúc dữ liệu được ghi từ ESP32 lên Firebase cho đến khi nó hiển thị trên giao diện Web Dashboard, kết quả trung bình nằm trong khoảng 1.5 – 2 giây. Trong các điều kiện mạng tối ưu, độ trễ có thể xuống dưới 1 giây (gần như tức thời). Tuy nhiên, trong những trường hợp mạng Wi-Fi yếu hoặc khi Firebase có thể bị nghẽn cục bộ, độ trễ tối đa ghi nhận được là 3.5 giây.

Độ trễ phản hồi lệnh điều khiển từ xa: Đối với các lệnh điều khiển được gửi từ Web Dashboard (qua Firebase) đến ESP32 và sau đó thiết bị phản hồi trạng thái ngược lại Web Dashboard, thời gian cập nhật trạng thái thiết bị trên Web chỉ mất khoảng 1 – 2 giây. Điều này chứng tỏ ESP32 phản hồi lệnh rất nhanh chóng nhờ vào cơ chế hỗ trợ stream dữ liệu của Firebase.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT LUẬN

Tổng kết lại, đề tài hiện tại mới chỉ dừng lại ở mức độ mô hình thử nghiệm, chưa đáp ứng đầy đủ các yêu cầu sử dụng thực tế. Mặc dù hệ thống đã có khả năng giám sát và điều khiển các thiết bị theo thông số môi trường, nhưng để có thể triển khai và ứng dụng hiệu quả trong thực tiễn, cần có những cải tiến cả về phần cứng lẫn phần mềm. Việc kết hợp ESP32 với Web Dashboard đã mở ra hướng tiếp cận hiện đại, giúp việc giám sát và điều khiển trở nên trực quan và tiện lợi hơn. Tuy nhiên, để mang lại giá trị thực sự cho người dùng, hệ thống cần được hoàn thiện về giao diện, độ bền và độ chính xác của phần cứng, cũng như mở rộng các chức năng phần mềm. Khi mô hình có thể được triển khai ổn định trong thực tế và đem lại hiệu quả rõ ràng, khi đó mới có thể khẳng định rằng đề tài đã đạt được mục tiêu đề ra.

5.2. HƯỚNG PHÁT TRIỂN

Mở rộng phạm vi ứng dụng: Trong tương lai, hệ thống không chỉ dừng lại ở việc giám sát khí gas mà có thể tích hợp thêm nhiều loại cảm biến khác như cảm biến bụi mịn, độ ồn, ánh sáng, áp suất không khí, v.v. nhằm cung cấp một giải pháp toàn diện về quản lý môi trường trong nhà hoặc ngoài trời.

Tích hợp công nghệ IoT và trí tuệ nhân tạo (AI): Việc ứng dụng AI để phân tích dữ liệu từ các cảm biến, phát hiện các xu hướng bất thường và tự động đưa ra cảnh báo hoặc điều chỉnh thiết bị sẽ giúp hệ thống trở nên thông minh và hiệu quả hơn. Đồng thời, nền tảng IoT có thể hỗ trợ hệ thống kết nối và đồng bộ dữ liệu theo thời gian thực, nâng cao khả năng điều khiển từ xa và tự động hóa.

Phát triển ứng dụng di động hoặc giao diện đám mây: Ngoài Web Dashboard hiện tại, có thể phát triển thêm ứng dụng di động hoặc giao diện đám mây, giúp người dùng có thể theo dõi và điều khiển thiết bị mọi lúc mọi nơi thông qua smartphone hoặc máy tính bảng. Việc này không chỉ tăng tính tiện lợi mà còn nâng cao trải nghiệm người dùng và tính thương mại hóa của sản phẩm.

TÀI LIỆU THAM KHẢO

1. <https://blynk.cloud/dashboard/398947/templates/edit/1213581/datastreams>
2. <https://asin.com.vn/may-do-nong-do-khi-gar-fgd-1.html>
3. <https://youtu.be/MsTFVRvquGU?si=CzI26W6j70363t4L>
4. <https://www.dienmayxanh.com/kinh-nghiem-hay/nhung-nguyen-nhan-khien-binh-gas>
5. <https://community.blynk.cc/t/mach-esp32-tu-k-t-n-i-l-i-m-ng-khi-b-m-t-m-ng-ho-c-m-t>
6. <https://mecsu.vn/ho-tro-ky-thuat/i2c-lcd-voi-esp32-va-esp8266-su-dung>
7. <https://www.iotzone.vn/esp32/huong-dan-dht11-dht22-esp32-tren-arduino-ide/>
8. <https://www.upesy.com/blogs/tutorials/esp32-dht22-am2302-humidity-temperature>
9. <https://esp32io.com/tutorials/esp32-temperature-humidity-sensor>
10. <https://mecsu.vn/ho-tro-ky-thuat/cam-bien-khi-mq2.61z>