**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**----🙟🙝🕮🙟🙝----**

A blue and white circle with text

Description automatically generated

BÁO CÁO TỔNG KẾT MÔN HỌC IOT VÀ ỨNG DỤNG

**DỰ ÁN NHÀ THÔNG MINH HOME-X**

Sinh viên thực hiện: Nguyễn Thành Vinh 22027166

Nghiêm Đình Dương 22022104

Phạm Ngọc Khánh 22022112

Lê Anh Tuấn 22029022

Giảng viên phụ trách: PGT. TS Đinh Thị Thái Mai

Mã lớp học phần: ELT3244 65

**ĐÁNH GIÁ THÀNH VIÊN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Vai trò** | **MSV** | **Điểm** |
| Nguyễn Thành Vinh | Trưởng nhóm | 22027166 | 10/10 |
| Phạm Ngọc Khánh | Thành viên | 22029028 | 10/10 |
| Nghiêm Đình Dương | Thành viên | 22029036 | 10/10 |
| Lê Anh Tuấn | Thành viên | 22029082 | 3/10 |

**LỜI CẢM ƠN**

Nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến PGS.TS Đinh Thị Thái Mai – giảng viên phần lý thuyết và thầy Lưu Bách Hưng – giảng viên phần thực hành của bộ môn. Các thầy đã luôn tận tình giảng dạy, hướng dẫn và luôn đồng hành cùng sinh viên trong suốt quá trình học tập và thực hiện dự án cuối khóa này.

Nhờ sự hướng dẫn chu đáo và tâm huyết từ các thầy, nhóm đã có cơ hội áp dụng kiến thức lý thuyết vào thực tiễn, qua đó hiểu rõ hơn về quy trình xây dựng một hệ thống kỹ thuật – từ việc thu thập dữ liệu qua cảm biến, truyền tải, xử lý đến hiển thị thông tin theo thời gian thực. Đây không chỉ là một hoạt động học tập thiết thực mà còn là dịp để nhóm nâng cao kỹ năng làm việc nhóm, rèn luyện tư duy hệ thống và khả năng giải quyết vấn đề trong môi trường kỹ thuật.

Mặc dù nhóm em đã luôn nỗ lực để hoàn thành bài tập nghiêm túc và trách nhiệm để đạt kết quả cao, nhưng trong quá trình không thể tránh khỏi thiếu sót và sai lầm. Nhóm chúng em rất mong nhận được những lời nhận xét và góp ý quý báu từ các thầy để có thể học hỏi, cải thiện và phát triển tốt hơn trong các môn học và dự án trong tương lai.

Nhóm chúng em xin chân thành cảm ơn thầy, cô!

**MỤC LỤC**

[MỞ ĐẦU 5](#_Toc217253770)

[**1.** **Giới thiệu đề tài** 5](#_Toc217253771)

[**2.** **Mục tiêu nghiên cứu** 5](#_Toc217253772)

[**3.** **Phạm vi nghiên cứu** 6](#_Toc217253773)

[**4.** **Phương pháp nghiên cứu** 6](#_Toc217253774)

[**5.** **Ý nghĩa nghiên cứu** 8](#_Toc217253775)

[CHƯƠNG 1: TỔNG QUAN VỀ IOT VÀ DỰ ÁN 9](#_Toc217253776)

[**1.** **Giới thiệu về IoT** 9](#_Toc217253777)

[**2.** **Kiến trúc điển hình của 1 hệ thống IoT** 10](#_Toc217253778)

[**3.** **Các mô hình nhà thông minh phổ biến trên thế giới** 11](#_Toc217253779)

[**4.** **Giới thiệu tổng quan về mô hình Home-X** 13](#_Toc217253780)

[**5.** **Kết luận** 14](#_Toc217253781)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 15](#_Toc217253782)

[**1.** **Nguyên lý hoạt động của hệ thống và những features nổi bật của hệ thống** 15](#_Toc217253783)

[**1.1 Nguyên lý hoạt động của hệ thống nhà thông minh HomeX** 15](#_Toc217253784)

[**1.2** **Các tính năng nổi bật của hệ thống** 16](#_Toc217253785)

[**1.2.1** **Wi-Fi Commissioning** 17](#_Toc217253786)

[**1.2.1** **OTA gateway firmware version** 19](#_Toc217253787)

[**1.2.2** **Điều khiển qua Web UI Dashboard** 20](#_Toc217253788)

[**1.2.3** **Cơ chế tự động kết nối lại khi gián đoạn mạng** 21](#_Toc217253789)

[**1.2.4** **Giao thức UDP** 22](#_Toc217253790)

[**2.** **Môi trường lập trình và Google Firebase** 24](#_Toc217253791)

[**2.1.** **Arduino IDE** 24](#_Toc217253792)

[**2.2.** **Google Firebase** 25](#_Toc217253793)

[**3.** **Phát triển phần mềm** 26](#_Toc217253794)

[**3.1.** **Xử lý analog (ESP32)** 26](#_Toc217253795)

[**3.2.** **Wi-Fi (ESP WROOM 32)** 27](#_Toc217253796)

[**3.3.** **Repository** 28](#_Toc217253797)

[**4.** **Lưu đồ thuật toán** 28](#_Toc217253798)

[**4.1** **Luồng hoạt động tổng quát** 29](#_Toc217253799)

[**4.2** **Lưu đồ xử lý tín hiệu cảm biến và truyền thông IoT** 30](#_Toc217253800)

[**4.3** **Lưu đồ giải thuật điều khiển Servo và cơ chế chống lũ** 31](#_Toc217253801)

[CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM 32](#_Toc217253802)

[**1.** **User flow – Trải nghiệm người dùng** 32](#_Toc217253803)

[**2.** **Kết quả vận hành hệ thống** 32](#_Toc217253804)

[**a.** **Linh kiện sử dụng** 32](#_Toc217253805)

[**b.** **Video thực nghiệm** 34](#_Toc217253806)

[**3.** **Những đề xuất nâng cấp hệ thống** 34](#_Toc217253807)

[**4.** **Kết luận** 34](#_Toc217253808)

# **MỞ ĐẦU**

1. **Giới thiệu đề tài**

Trong bối cảnh đô thị hóa ngày càng phát triển, nhu cầu nâng cao mức độ an toàn, tiện nghi và tự động hóa trong không gian sống đang trở thành xu hướng tất yếu. Các khu vực chức năng trong nhà như bếp, phòng ngủ hay garage tiềm ẩn nhiều rủi ro khác nhau như rò rỉ khí gas, môi trường sinh hoạt không phù hợp hoặc nguy cơ ngập nước, nếu không được giám sát và xử lý kịp thời có thể gây thiệt hại về tài sản và ảnh hưởng đến an toàn của người sử dụng.

Các hệ thống giám sát truyền thống thường hoạt động độc lập, thiếu khả năng kết nối và phản hồi tự động, dẫn đến hạn chế trong việc cảnh báo sớm và xử lý sự cố. Trước sự phát triển mạnh mẽ của công nghệ Internet of Things (IoT), việc ứng dụng các cảm biến thông minh kết hợp với vi điều khiển và kết nối Internet đã mở ra hướng tiếp cận hiệu quả cho việc xây dựng hệ thống nhà thông minh có khả năng giám sát, cảnh báo và phản ứng tự động theo thời gian thực.

Xuất phát từ thực tế đó, đề tài “Thiết kế và triển khai hệ thống nhà thông minh ứng dụng IoT” được thực hiện nhằm nghiên cứu và xây dựng một mô hình nhà thông minh có khả năng giám sát an toàn và môi trường sống tại nhiều khu vực khác nhau, đồng thời tự động đưa ra các hành động xử lý và cảnh báo từ xa thông qua các nền tảng kết nối Internet. Hệ thống hướng tới giải pháp có chi phí hợp lý, dễ triển khai và có khả năng mở rộng, phù hợp với mô hình nhà ở hiện đại.

1. **Mục tiêu nghiên cứu**

Xây dựng một hệ thống có khả năng giám sát môi trường và an toàn theo thời gian thực, hoạt động ổn định, dễ sử dụng, dễ lắp đặt và có thể tự động xử lý cũng như cảnh báo từ xa khi phát hiện các tình huống bất thường.

Từ yêu cầu trên, các mục tiêu nghiên cứu cụ thể bao gồm:

* Nghiên cứu và lựa chọn các loại cảm biến phù hợp cho từng khu vực trong nhà, bao gồm: cảm biến khí gas, cảm biến nhiệt độ – độ ẩm, cảm biến mực nước, …
* Thiết kế và xây dựng hệ thống phần cứng – phần mềm tích hợp dựa trên mạch điều khiển (ESP WROOM32/ESP32) và kết nối không dây Wi-Fi.
* Xây dựng các chức năng thông minh cho từng khu vực:
  + **Khu vực bếp**: giám sát rò rỉ khí gas, tự động kích hoạt quạt hút và cảnh báo bằng buzzer.
  + **Khu vực phòng ngủ**: giám sát nhiệt độ và độ ẩm nhằm theo dõi điều kiện môi trường sinh hoạt.
  + **Khu vực garage**: phát hiện mực nước dâng cao và tự động đóng cửa chống ngập.
* Triển khai chức năng cảnh báo từ xa thông qua Web UI, cho phép người dùng theo dõi trạng thái hệ thống mọi lúc, mọi nơi.
* Thử nghiệm và đánh giá hiệu quả hoạt động của hệ thống trong điều kiện thực tế.

1. **Phạm vi nghiên cứu**

Dự án tập trung nghiên cứu và xây dựng mô hình hệ thống nhà thông minh ở quy mô nhỏ, phù hợp áp dụng cho hộ gia đình, không gian sinh hoạt – làm việc nhỏ. Hệ thống bao gồm các thành phần chính như vi mạch điều khiển, các cảm biến giám sát môi trường và an toàn, cùng phần mềm giám sát và cảnh báo thông qua Web UI.

Trong phạm vi đề tài, nhóm tập trung vào việc giám sát, cảnh báo và tự động xử lý các tình huống cơ bản tại ba khu vực tiêu biểu là bếp, phòng ngủ và garage, đồng thời tạo nền tảng để có thể mở rộng thêm các tính năng nhà thông minh khác trong tương lai.

1. **Phương pháp nghiên cứu**

Trong quá trình thực hiện đề tài, nhóm nghiên cứu đã áp dụng nhiều phương pháp khác nhau nhằm đảm bảo tính khoa học, logic và hiệu quả trong việc xây dựng hệ thống nhà thông minh ứng dụng IoT. Các phương pháp chính được sử dụng bao gồm: nghiên cứu lý thuyết, thiết kế hệ thống, thực nghiệm và đánh giá tổng thể.

Giai đoạn đầu tiên là nghiên cứu lý thuyết, tập trung thu thập các kiến thức nền tảng liên quan đến mô hình nhà thông minh và công nghệ IoT. Nhóm tiến hành tìm hiểu nguyên lý hoạt động của các loại cảm biến môi trường và an toàn như cảm biến khí gas, cảm biến nhiệt độ – độ ẩm, cảm biến mực nước; cấu trúc và cơ chế hoạt động của mạch điều khiển ESP8266/ESP32; cũng như các phương thức kết nối, giám sát và cảnh báo từ xa thông qua Internet.

Tiếp theo, nhóm thực hiện phân tích yêu cầu và thiết kế hệ thống. Dựa trên các khu vực chức năng trong nhà, nhóm xây dựng mô hình hệ thống gồm ba khu vực chính: bếp, phòng ngủ và garage. Trên cơ sở đó, nhóm lựa chọn các linh kiện phần cứng phù hợp như: cảm biến khí gas MQ2, cảm biến nhiệt độ – độ ẩm DHT11, cảm biến mực nước, vi mạch điều khiển ESP8266/ESP32, relay, quạt hút, cơ cấu đóng/mở cửa và nguồn cấp điện ổn định. Về phần mềm, nhóm xây dựng chương trình điều khiển để thu thập dữ liệu cảm biến, xử lý ngưỡng an toàn, kích hoạt các thiết bị chấp hành và gửi dữ liệu giám sát, cảnh báo đến người dùng thông Web UI.

Giai đoạn thực nghiệm được tiến hành bằng cách lắp ráp hệ thống trên breadboard và lập trình vi điều khiển bằng ngôn ngữ C/C++. Nhóm tiến hành kiểm tra từng chức năng riêng lẻ (module) và toàn hệ thống, bao gồm: phát hiện rò rỉ khí gas và kích hoạt quạt hút tại khu vực bếp, đo nhiệt độ – độ ẩm tại phòng ngủ, phát hiện mực nước dâng và điều khiển đóng cửa tại garage, cũng như khả năng hiển thị dữ liệu và gửi cảnh báo theo thời gian thực.

Cuối cùng, hệ thống được đánh giá trong các điều kiện hoạt động khác nhau, nhằm kiểm tra độ chính xác của cảm biến, độ trễ phản hồi của hệ thống, tính ổn định của kết nối Wi-Fi và độ tin cậy của chức năng cảnh báo. Dựa trên kết quả thử nghiệm, nhóm đề xuất các hướng cải tiến và mở rộng trong tương lai như nâng cấp cảm biến, bổ sung các thiết bị chấp hành thông minh và tích hợp thêm các kịch bản tự động hóa nâng cao cho hệ thống nhà thông minh.

1. **Ý nghĩa nghiên cứu**

Đề tài “Thiết kế và triển khai hệ thống nhà thông minh ứng dụng IoT” mang lại nhiều ý nghĩa thiết thực cả về mặt khoa học lẫn thực tiễn. Trước hết, dự án góp phần nâng cao mức độ an toàn và tiện nghi trong không gian sống thông qua việc giám sát liên tục các khu vực quan trọng trong nhà và tự động xử lý khi phát hiện các tình huống bất thường như rò rỉ khí gas hay nguy cơ ngập nước. Điều này giúp giảm thiểu rủi ro, bảo vệ tài sản và đảm bảo an toàn cho người sử dụng.

Về mặt ứng dụng, hệ thống được thiết kế với chi phí hợp lý, cấu trúc đơn giản và dễ triển khai, phù hợp với các hộ gia đình, nhà trọ hoặc không gian sinh hoạt – làm việc nhỏ. Việc sử dụng nền tảng IoT và ứng dụng giám sát từ xa giúp người dùng có thể theo dõi trạng thái ngôi nhà mọi lúc, mọi nơi, từ đó nâng cao trải nghiệm sử dụng và hiệu quả quản lý.

Về mặt học thuật và công nghệ, đề tài giúp sinh viên vận dụng kiến thức về vi điều khiển, cảm biến, lập trình nhúng và truyền thông không dây vào một hệ thống thực tế hoàn chỉnh. Đồng thời, đây cũng là nền tảng để mở rộng và phát triển các giải pháp nhà thông minh khác trong tương lai, góp phần thúc đẩy việc ứng dụng công nghệ IoT vào đời sống, phù hợp với xu hướng chuyển đổi số và phát triển đô thị thông minh hiện nay.

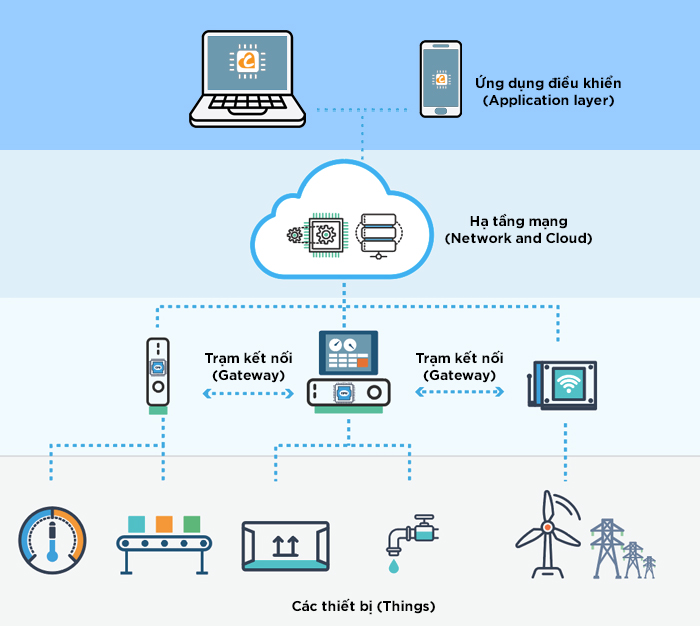
# **CHƯƠNG 1: TỔNG QUAN VỀ IOT VÀ DỰ ÁN**

1. **Giới thiệu về IoT**

Khái niệm "Cách mạng công nghiệp 4.0" lần đầu tiên xuất hiện trong một báo cáo của chính phủ Đức năm 2013 và nhanh chóng ảnh hưởng rộng rãi trên toàn thế giới, trong đó có Việt Nam.

Klaus Schwab, người sáng lập và chủ tịch điều hành diễn đàn kinh tế thế giới đã mang đến cái nhìn đơn giản hơn về cách mạng công nghiệp 4.0 như sau: “Cách mạng công nghiệp đầu tiên sử dụng năng lượng nước và hơi nước để cơ giới hóa sản xuất. Cuộc cách mạng lần thứ 2 diễn ra nhờ ứng dụng điện năng để sản xuất hàng loạt. Cuộc cách mạng lần thứ 3 sử dụng điện tử và công nghệ thông tin để tự động hóa sản xuất. Bây giờ, cuộc cách mạng công nghiệp thứ 4 đang này nở từ cuộc cách mạng lần thứ 3, nó kết hợp các công nghệ lại với nhau, làm mờ ranh giới giữa vật lý, kỹ thuật số và sinh học.”

Cách mạng công nghiệp 4.0 mang lại cả cơ hội lẫn thách thức lớn cho nước ta, đặc biệt nhấn mạnh vào việc kết nối các công nghệ tiên tiến như trí tuệ nhân tạo, dữ liệu lớn và đặc biệt là IoT.



IoT hay còn gọi là mạng lưới vạn vật kết nối internet, là khái niệm để chỉ mạng lưới các thiết bị được kết nối với internet, cho phép chúng thu thập và trao đổi dữ liệu với nhau mà không cần sự can thiệp trực tiếp của con người.

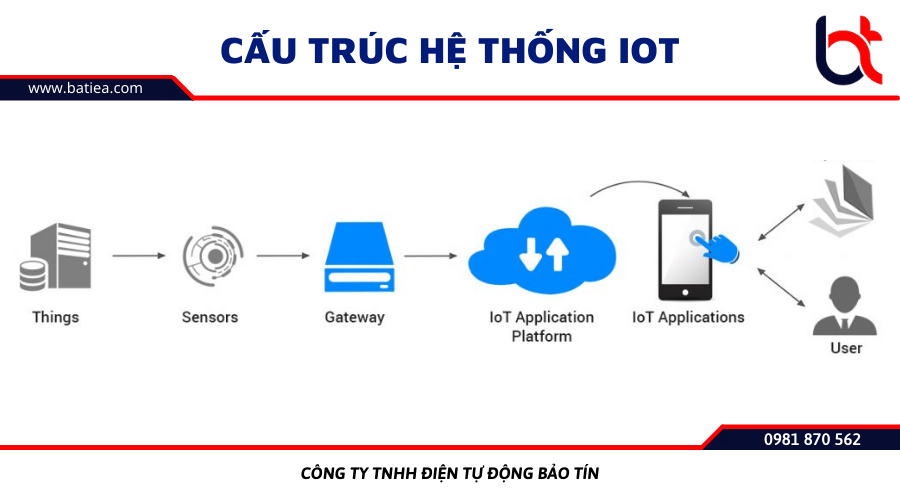
Các thiết bị trong hệ thống IoT có thể rất đa dạng, từ điện thoại thông minh, tivi, máy giặt, máy pha cà phê đến bóng đèn, cảm biến nhiệt độ, camera an ninh và hàng loạt thiết bị khác. Chúng có thể kết nối thông qua nhiều phương thức như WiFi, Bluetooth, mạng di động (4G/5G), ZigBee, hồng ngoại, ….

Điểm nổi bật của IoT là khả năng kết nối mọi các thiết bị vào một hệ sinh thái thông minh, giúp các thiết bị tự động tương tác, phản hồi theo thời gian thực và hỗ trợ ra quyết định chính xác. Với khả năng trên, IoT giúp nâng cao hiệu quả trong sản xuất và quản lý, mang đến nhiều tiện ích cho đời sống con người, từ việc điều khiển nhà thông minh đến quản lý giao thông, y tế, môi trường và nhiều ứng dụng khác.

Theo dự báo của Cisco, một trong những nhà cung cấp giải pháp mạng hàng đầu thì đến năm 2020 đã có khoảng 50 tỷ thiết bị kết nối vào internet. Sự phát triển của IoT ngày càng nhanh chóng, làm mờ ranh giới giữa vật lý, kỹ thuật số và sinh học, đóng vai trò là mạng lưới khổng lồ kết nối tất cả mọi thứ, bao gồm cả con người.

1. **Kiến trúc điển hình của 1 hệ thống IoT**

Hệ thống IoT đã trở thành nền tảng công nghệ then chốt trong việc xây dựng các giải pháp thông minh, bao gồm cả lĩnh vực phòng cháy chữa cháy. Khi áp dụng vào hệ thống phòng cháy, chữa cháy thông minh, mô hình này không chỉ giúp phát hiện sớm các nguy cơ cháy mà còn tối ưu hóa quy trình xử lý sự cố, giảm thiểu thiệt hại về người và tài sản



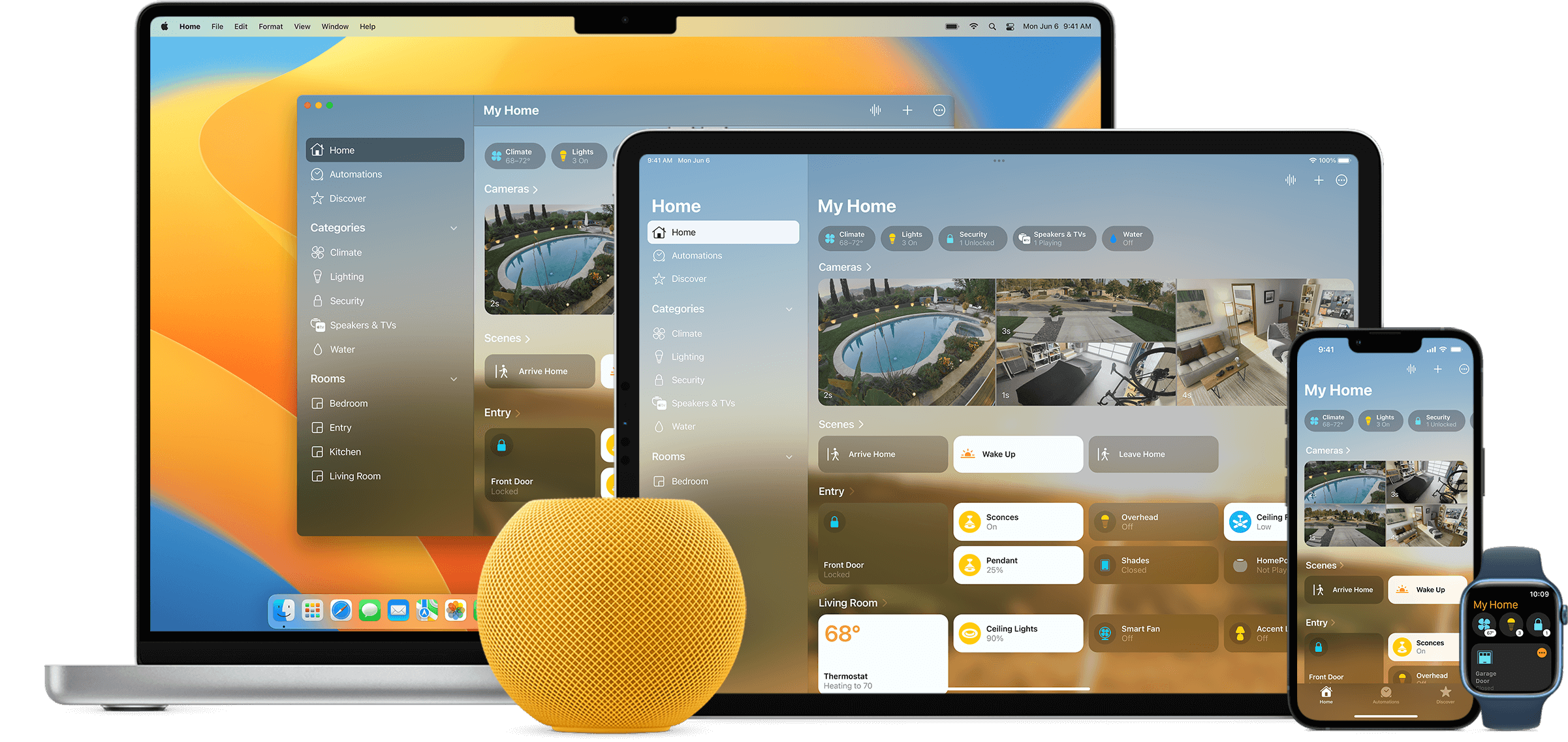
Cấu trúc của một hệ thống IoT thường được phân thành nhiều lớp, mỗi lớp đảm nhận vai trò cụ thể trong quá trình thu thập, truyền tải, xử lý dữ liệu và triển khai ứng dụng. Theo mô hình cấu trúc IoT theo chiều ngang, một hệ thống sẽ bao gồm 4 lớp: lớp thiết bị; lớp mạng; lớp nền tảng và lớp ứng dụng.

* **Lớp thiết bị (Sensors/End devices):** là lớp đầu tiên đóng vai trò thu thập dữ liệu từ môi trường thực. Các thiết bị như cảm biến nhiệt độ, cảm biến khí gas sẽ ghi nhận các thông số quan trọng và chuyển chúng thành tín hiệu số, sau đó truyền tải vào vi điều khiển để tiến hành xử lý dựa trên các ngưỡng đã được thiết lập.
* **Lớp mạng (Network):** là lớp chịu trách nhiệm truyền dữ liệu từ lớp thiết bị đến các gateway hoặc trực tiếp lên đám mây. Việc truyền tải có thể thông qua các công nghệ như WiFi, ZigBee, LoRa, mạng di động (4G/5G), hoặc ethernet.
* **Lớp nền tảng** **(Cloud platform)**: là tầng trung gian trong mô hình IoT, chịu trách nhiệm xử lý, lưu trữ, quản lý và phân phối dữ liệu thu thập từ các thiết bị cảm biến thông qua lớp mạng. Đây là nơi tập trung sức mạnh tính toán và điều phối các hoạt động trong toàn hệ thống IoT. Nhờ có lớp này mà các hoạt động như theo dõi và xử lý theo thời gian thực; quản lý thiết bị từ xa và cảnh báo nguy hiểm có thể phối hợp chặt chẽ với nhau, nhất là đối với kiểu hệ thống cần sự chính xác cao.
* **Lớp ứng dụng (Applications):** là lớp cung cấp giao diện tương tác cho người dùng và tích hợp với các hệ thống khác. Trong hệ thống phòng cháy chữa cháy, ứng dụng di động hoặc Dashboard web hiển thị trạng thái các cảm biến, trạng thái hoạt động của các thiết bị. Hệ thống cũng có cơ chế tự động, thông minh dựa trên dữ liệu từ cảm biến, đồng thời gửi cảnh báo qua email và đẩy thông báo lên mobile app và Dashboard web.

1. **Các mô hình nhà thông minh phổ biến trên thế giới**

Trên thế giới hiện nay có nhiều nền tảng và mô hình nhà thông minh (smart home) do các tập đoàn công nghệ phát triển, giúp người dùng điều khiển và tự động hoá các thiết bị trong nhà thông qua ứng dụng hoặc lệnh giọng nói. Các hệ sinh thái này không chỉ tập trung vào việc kết nối các thiết bị như đèn, camera, cảm biến và ổ khoá mà còn hỗ trợ tạo các *routine* (kịch bản tự động hoá) để nâng cao tiện nghi và an toàn cho cư dân.

**Apple Home (HomeKit)** là hệ thống của Apple, được thiết kế để tích hợp sâu với các thiết bị iOS và macOS, cho phép kiểm soát nhà thông minh ngay trong ứng dụng *Home*. Nó nổi bật ở khả năng bảo mật và quyền riêng tư, đảm bảo dữ liệu điều khiển tại nhà được mã hoá chặt chẽ. [Transcend Home Theater](https://ththeater.com/apple-homekit-vs-google-home-vs-alexa-vs-smartthings/?utm_source=chatgpt.com)



**Google Home** (với trợ lý Google Assistant và các sản phẩm như Google Nest Hub, Nest Audio) là một trong những nền tảng phổ biến nhất, hỗ trợ điều khiển bằng giọng nói và tương thích với rất nhiều thiết bị từ nhiều hãng khác nhau. [Nhà thông minh Matter Việt Nam](https://mattervn.com/thiet-bi-nha-thong-minh-cua-google/?srsltid=AfmBOoplf3DfRS7OxQl5gp5yV7trEij-t8kKyNTXzck0ZbHMRB2cwglS&utm_source=chatgpt.com)



**Amazon Alexa** là hệ sinh thái smart home nhiều thiết bị hỗ trợ nhất hiện nay, với Amazon Echo, Echo Dot và Echo Show làm trung tâm điều khiển, cho phép người dùng tạo các routine tự động hoá đa dạng.



Dựa trên những mô hình đã được phát triển và thương mại hóa, nhóm đã có thêm nhiều ý tưởng và cách thức phát triển dự án.

1. **Giới thiệu tổng quan về mô hình Home-X**

**Hệ thống nhà thông minh** được xây dựng với mục tiêugiám sát an toàn và tự động hoá các khu vực quan trọng trong ngôi nhà, giúp nâng cao chất lượng sống, giảm thiểu rủi ro và hạn chế sự can thiệp thủ công của con người. Hệ thống sử dụng các cảm biến thông minh kết hợp với vi điều khiển để theo dõi môi trường theo thời gian thực, từ đó tự động đưa ra các hành động xử lý phù hợp khi phát hiện các dấu hiệu bất thường.



**Hình 1: System diagram**

**Tại khu vực bếp,** hệ thống tập trung vào yếu tố an toàn với chức năng giám sát rò rỉ khí gas. Khi nồng độ khí gas vượt ngưỡng an toàn, hệ thống sẽ tự động kích hoạt quạt hút để đẩy khí gas ra ngoài, đồng thời phát cảnh báo bằng buzzer nhằm thông báo kịp thời cho người dùng, giúp hạn chế nguy cơ cháy nổ và đảm bảo an toàn sinh hoạt.

Đốivới **khu vực phòng ngủ,** hệ thống thực hiệngiám sát nhiệt độ và độ ẩmtrong không gian phòng.Các thông số môi trường được thu thập liên tục, giúp người dùng theo dõi điều kiện vi khí hậu, từ đó có thể điều chỉnh các thiết bị hỗ trợ như quạt, điều hòa hoặc máy tạo ẩm để đảm bảo sự thoải mái và bảo vệ sức khỏe.

**Tại khu vực garage,** hệ thống được trang bịcơ chế phát hiện mực nước dâng caonhằm phòng chống ngập úng. Khi mực nước vượt quá ngưỡng cho phép, hệ thống sẽ tự động đóng cửa garage để ngăn nước tràn vào bên trong, bảo vệ phương tiện và tài sản.

Nhìn chung, hệ thống nhà thông minh này hoạt động dựa trên nguyên lý giám sát – phát hiện – phản ứng tự động, cho phép xử lý nhanh các tình huống nguy hiểm, nâng cao mức độ an toàn và tiện nghi. Với chi phí hợp lý, khả năng mở rộng linh hoạt và dễ dàng tích hợp thêm các tính năng khác, hệ thống có thể trở thành nền tảng cốt lõi cho một ngôi nhà thông minh hiện đại.

1. **Kết luận**

Chương 1 đã trình bày các cơ sở lý thuyết nền tảng phục vụ cho việc nghiên cứu và phát triển hệ thống nhà thông minh ứng dụng công nghệ IoT. Thông qua việc phân tích khái niệm, đặc điểm và vai trò của IoT trong bối cảnh cách mạng công nghiệp 4.0, chương này đã làm rõ cách thức mà mạng lưới các thiết bị thông minh, hệ thống cảm biến, vi điều khiển và nền tảng kết nối Internet có thể được tích hợp để hình thành nên một hệ thống nhà thông minh có khả năng giám sát, điều khiển và phản hồi tự động theo thời gian thực.

Bên cạnh đó, chương cũng đã trình bày các mô hình và thành phần cơ bản của hệ thống nhà thông minh hiện nay, bao gồm các khối cảm biến, khối xử lý trung tâm, thiết bị chấp hành và hệ thống giám sát – cảnh báo từ xa. Đặc biệt, điểm nổi bật của hệ thống nhà thông minh ứng dụng IoT nằm ở khả năng tự động phát hiện các trạng thái bất thường trong môi trường sống như rò rỉ khí gas, điều kiện nhiệt độ – độ ẩm không phù hợp hoặc nguy cơ ngập nước, từ đó kích hoạt các cơ chế xử lý và cảnh báo kịp thời nhằm đảm bảo an toàn và nâng cao tiện nghi cho người sử dụng.

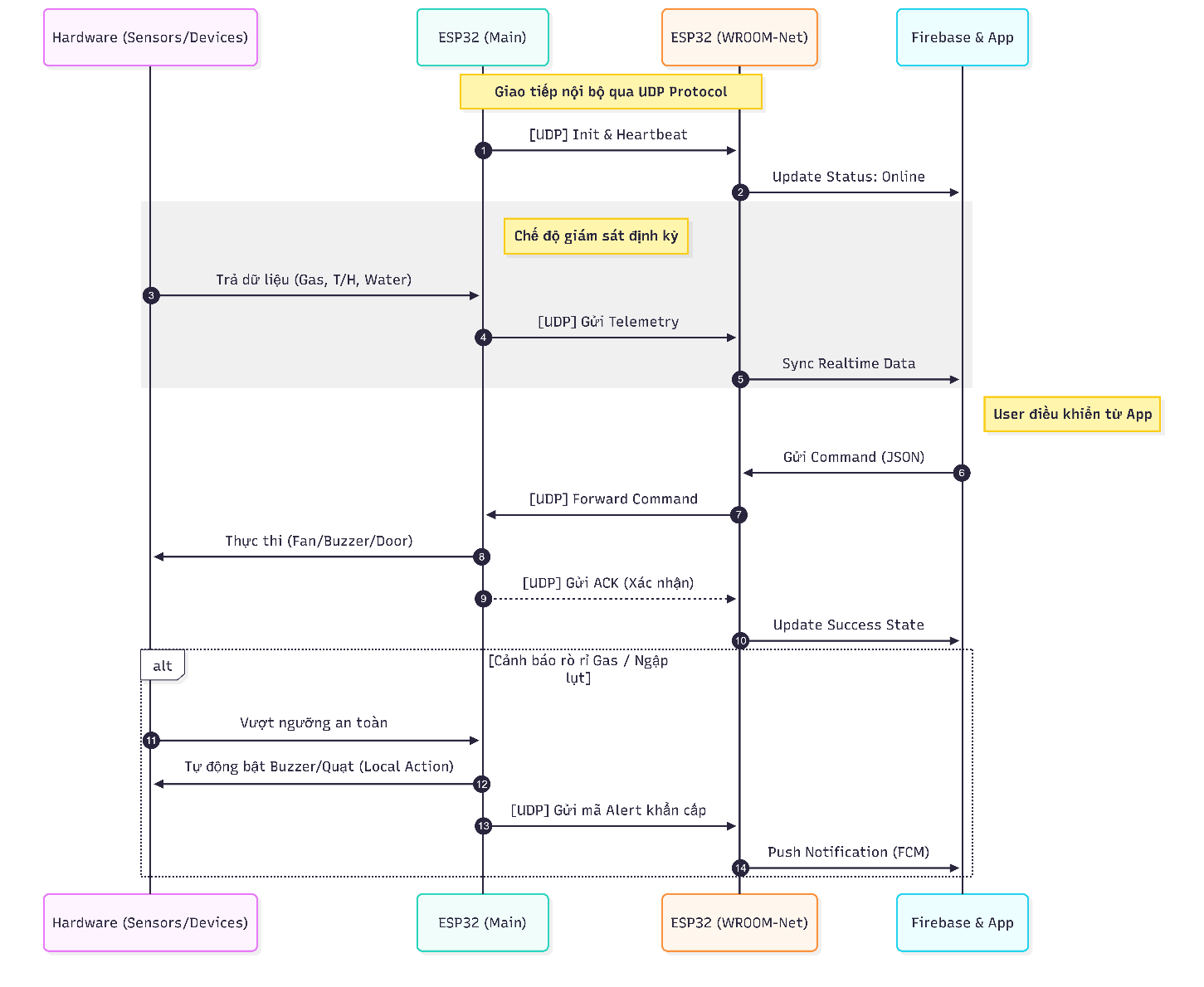
Với sự kết hợp giữa các cảm biến thông minh, mạch điều khiển, các thiết bị chấp hành và nền tảng kết nối Internet, hệ thống nhà thông minh không chỉ đáp ứng tốt các yêu cầu về độ chính xác, độ tin cậy và tính linh hoạt, mà còn cho phép mở rộng và tích hợp thêm nhiều chức năng tự động hóa khác trong tương lai.

Những nội dung được trình bày trong Chương 1 đã tạo nên nền tảng lý thuyết vững chắc cho việc triển khai đề tài. Đây là cơ sở quan trọng để chuyển sang các chương tiếp theo, trong đó sẽ tập trung vào thiết kế chi tiết hệ thống phần cứng và phần mềm, quy trình lắp đặt, thử nghiệm và đánh giá hiệu năng của mô hình nhà thông minh trong điều kiện thực tế.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

1. **Nguyên lý hoạt động của hệ thống và những features nổi bật của hệ thống**

**1.1 Nguyên lý hoạt động của hệ thống nhà thông minh HomeX**



**Hình 2: Sequence Diagram**

Sequence diagram: Là một loại sơ đồ tập trung vào tương tác giữa các đối tượng theo thời gian. Thể hiện thứ tự các thông điệp (messages) được gửi qua lại giữa các thành phần (user, hệ thống, module). Dùng để mô tả cách các đối tượng phối hợp để thực hiện một chức năng cụ thể.

Khi hệ thống được cấp nguồn, bộ điều khiển trung tâm (ESP WROOM32/ESP32) tiến hành khởi tạo các ngoại vi và kết nối vào mạng Wi-Fi, đồng thời thiết lập kết nối với nền tảng giám sát thông qua Goolgle Firebase. Sau khi kết nối thành công, trạng thái hoạt động của hệ thống được cập nhật và hiển thị cho người dùng theo thời gian thực (real-time) thông qua giao diện HomeX Dashboard do nhóm thiết kế.

Trong quá trình vận hành, hệ thống hoạt động theo cơ chế giám sát định kỳ và liên tục. Bộ điều khiển trung tâm lần lượt thu thập dữ liệu từ các cảm biến được lắp đặt tại các khu vực khác nhau trong nhà, bao gồm cảm biến khí gas tại khu vực bếp, cảm biến nhiệt độ – độ ẩm tại phòng ngủ và cảm biến mực nước tại garage. Dữ liệu cảm biến sau khi được thu thập sẽ được đưa vào bộ xử lý để so sánh với các ngưỡng an toàn đã được thiết lập trước.

**Tại khu vực bếp** khi phát hiện rò rỉ khí gas tại, tức là nồng độ khí gas vượt quá ngưỡng cho phép, hệ thống sẽ lập tức kích hoạt các thiết bị chấp hành tương ứng. Cụ thể, quạt hút khí gas được bật để đẩy khí gas ra ngoài, đồng thời buzzer phát tín hiệu cảnh báo tại chỗ. Song song với đó, hệ thống gửi thông báo cảnh báo đến người dùng thông qua Web UI hoặc email, giúp người dùng kịp thời nắm bắt tình trạng nguy hiểm.

**Tại khu vực garage** trong trường hợp mực nước dâng cao vượt ngưỡng an toàn, hệ thống sẽ tự động điều khiển cơ cấu đóng cửa garage nhằm ngăn nước tràn vào bên trong. Đồng thời, một thông báo cảnh báo cũng được gửi đến người dùng để thông tin về nguy cơ ngập và trạng thái xử lý của hệ thống.

Nếu các thông số cảm biến nằm trong ngưỡng an toàn, hệ thống không kích hoạt các thiết bị chấp hành mà chỉ thực hiện chức năng giám sát. Các giá trị nhiệt độ và độ ẩm tại phòng ngủ, cùng với trạng thái an toàn của hệ thống, sẽ được cập nhật liên tục lên Firebase – Web UI.

Nhờ cơ chế giám sát – phát hiện – xử lý – cảnh báo được thực hiện tự động, hệ thống nhà thông minh đảm bảo phản ứng nhanh trước các tình huống bất thường, đồng thời mang lại sự an toàn và tiện nghi cho người sử dụng.

* 1. **Các tính năng nổi bật của hệ thống**

Bên cạnh các chức năng giám sát và cảnh báo cơ bản, hệ thống được nhóm tập trung thiết kế theo hướng ổn định, linh hoạt và phù hợp với triển khai thực tế.

Một trong những điểm nổi bật của hệ thống là khả năng cập nhật firmware từ xa **(Over-The-Air – OTA) cho gateway**, cho phép nâng cấp, sửa lỗi và bổ sung tính năng mà không cần can thiệp trực tiếp vào phần cứng. Điều này giúp hệ thống duy trì tính ổn định trong quá trình vận hành dài hạn và phù hợp với các kịch bản sử dụng thực tế.

Bên cạnh đó, hệ thống hỗ trợ **Wi-Fi commissioning**, cho phép người dùng cấu hình và thay đổi mạng Wi-Fi trong quá trình sử dụng mà không cần nạp lại chương trình. Khác với nhiều mô hình truyền thống thường cấu hình cố định thông tin mạng ngay trong mã nguồn, giải pháp này giúp tăng tính linh hoạt, dễ lắp đặt và dễ mở rộng hệ thống sang các môi trường sử dụng khác nhau.

Ngoài ra, hệ thống được thiết kế theo hướng hoạt động độc lập và an toàn, trong đó các chức năng giám sát và xử lý sự cố vẫn được đảm bảo ngay cả khi mất kết nối Internet. Kiến trúc này giúp giảm sự phụ thuộc vào nền tảng đám mây, nâng cao độ tin cậy và đảm bảo các tình huống nguy hiểm như rò rỉ khí gas hay ngập nước vẫn được xử lý kịp thời.

Những đặc điểm trên cho thấy hệ thống không chỉ dừng lại ở mức mô phỏng học thuật mà còn hướng tới một giải pháp có khả năng vận hành ổn định, dễ bảo trì và phù hợp với ứng dụng thực tế, qua đó tạo ra sự khác biệt rõ rệt so với các mô hình nhà thông minh cơ bản khác.

#### **Wi-Fi Commissioning**

**Khái niệm**

Wi-Fi Commissioning, hay còn gọi là quy trình cấu hình mạng ban đầu, là quá trình liên kết một thiết bị IoT mới vào mạng Wi-Fi của người dùng.

Đây được coi là bước thiết lập đầu tiên và quan trọng nhất để thiết bị có thể truy cập Internet hoặc mạng cục bộ nhằm trao đổi dữ liệu.

**Tính cần thiết**

Khác với máy tính xách tay hay điện thoại thông minh, các thiết bị IoT như node cảm biến hay công tắc thông minh thường là các thiết bị không có màn hình hiển thị và bàn phím vật lý (headless devices).

Do đặc thù phần cứng này, người dùng không thể trực tiếp nhập tên mạng (SSID) và mật khẩu vào thiết bị theo cách thông thường.

Wi-Fi Commissioning sinh ra để giải quyết vấn đề giao diện nhập liệu, cho phép chuyển thông tin đăng nhập mạng từ điện thoại hoặc máy tính sang thiết bị IoT một cách an toàn và thuận tiện.

**Nguyên lý hoạt động**

Bước đầu tiên, thiết bị IoT khởi động ở chế độ chờ cấu hình do chưa có thông tin về mạng Wi-Fi gia đình.

Thiết bị sau đó tạo ra một kênh giao tiếp tạm thời với điện thoại của người dùng, kênh này có thể là sóng Wi-Fi (chế độ Access Point) hoặc Bluetooth (BLE).

Người dùng sử dụng ứng dụng trên điện thoại để nhập SSID và mật khẩu của mạng Wi-Fi cần kết nối.

Ứng dụng sẽ gửi thông tin xác thực này qua kênh giao tiếp tạm thời đến thiết bị IoT.

Cuối cùng, thiết bị IoT nhận thông tin, lưu vào bộ nhớ, ngắt kết nối tạm thời và chuyển sang chế độ Station (STA) để kết nối chính thức vào mạng Wi-Fi của người dùng.

**Phương pháp thực hiện phổ biến**

Phương pháp SoftAP (Software Access Point): Thiết bị IoT tự phát ra một mạng Wi-Fi riêng để người dùng kết nối điện thoại vào và truy cập một trang web cục bộ để điền thông tin mạng. Đây là phương pháp có tính tương thích cao nhất.

BLE Provisioning: Thiết bị sử dụng sóng Bluetooth để nhận thông tin cấu hình từ điện thoại, mang lại trải nghiệm tốt hơn và tiêu tốn ít năng lượng hơn, nhưng đòi hỏi phần cứng hỗ trợ.

Phương pháp SmartConfig (ESP-Touch): Điện thoại phát các gói tin UDP quảng bá chứa mã hóa SSID và mật khẩu vào không khí, thiết bị IoT sẽ lắng nghe và giải mã để tự kết nối.

Trong dự án nhà thông minh Home-X, nhóm đã sủ dụng thư viện **Wi-Fi Manager** để thực hiện Commission cho hệ thống.

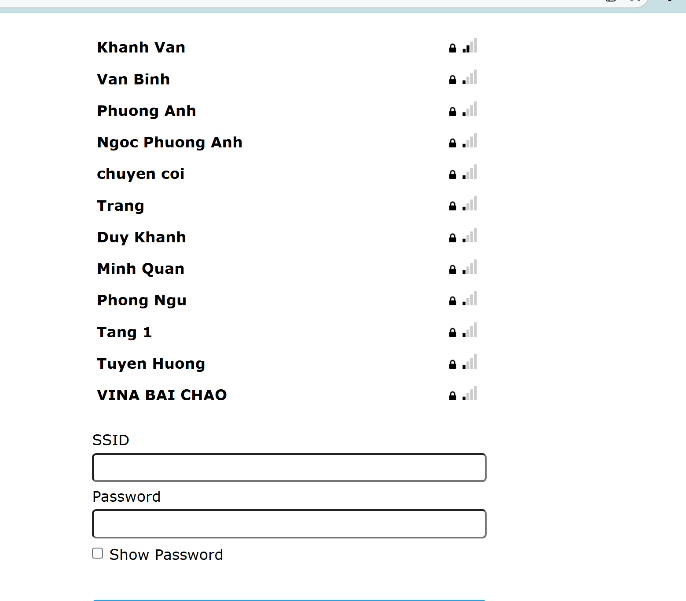
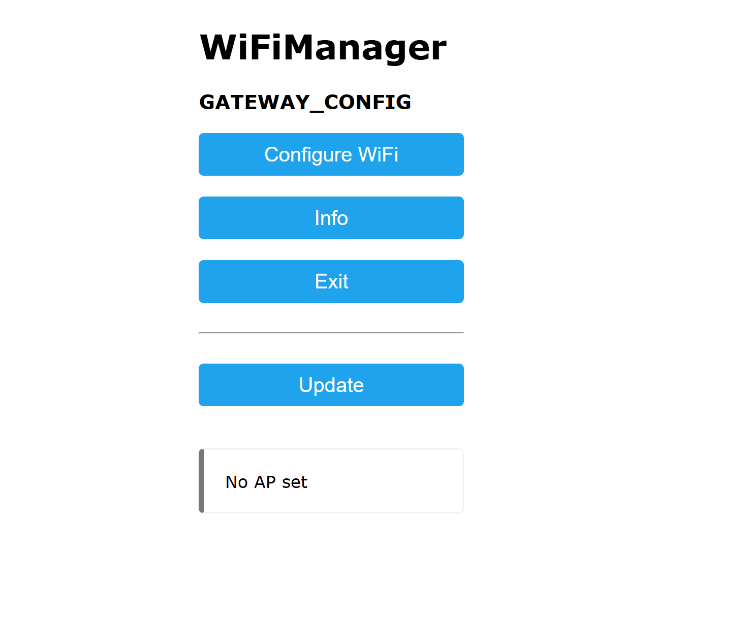
**Vai trò trong hệ thống nhà thông minh Home-X**

Việc tích hợp Wi-Fi Commissioning giúp sản phẩm IoT trở nên thương mại hóa và thân thiện với người dùng phổ thông.

Quy trình này loại bỏ việc phải nạp cứng (hardcode) tên và mật khẩu Wi-Fi trong mã nguồn lập trình.

Nhờ đó, thiết bị có thể hoạt động linh hoạt ở bất kỳ môi trường mạng nào mà không cần phải nạp lại chương trình.

**Hình 3: Giao diện Portal cho Wi-Fi Config**



#### **OTA gateway firmware version**

Bên cạnh Wi-Fi Commission, nhóm đã tìm hiểu và áp dụng quy trình OTA và nhận thấy rắng quy trình này là cần thiết để quản lý vòng đời của firmware, tối ưu trải nghiệm của users.

**Khái niệm OTA là gì?**

OTA (Over-The-Air) là quá trình cập nhật phần mềm (firmware), cấu hình hoặc dữ liệu cho các thiết bị IoT thông qua kết nối không dây (Wi-Fi, Bluetooth, 4G/5G, LoRa).

Thay vì phải tháo thiết bị ra và cắm cáp vào máy tính để nạp code, chỉ cần gửi tệp tin cập nhật (file .bin) từ server, thiết bị sẽ tự động tải về và cài đặt.

**Tầm quan trọng của OTA trong IoT**

Đối với một hệ thống Smart Home, OTA đóng vai trò sống còn vì:

Sửa lỗi và bảo mật: Khi phát hiện lỗ hổng bảo mật hoặc lỗi logic (bug) sau khi lắp đặt, có thể vá lỗi từ xa cho hàng trăm thiết bị cùng lúc.

Cập nhật tính năng mới: Có thể thêm các kịch bản tự động hóa mới mà không cần thu hồi thiết bị từ nhà khách hàng.

Tiết kiệm chi phí: Giảm thiểu chi phí bảo trì tận nơi.

Khả năng mở rộng (Scalability): Giúp quản lý **vòng đời** của hàng nghìn thiết bị một cách đồng bộ.

**Các phương pháp OTA phổ biến**

**A. Pull Strategy (Thiết bị chủ động)**

Thiết bị thường xuyên kiểm tra trên Server (Firebase hoặc HTTP Server) xem có phiên bản mới không.

Ưu điểm: Đơn giản, an toàn cho thiết bị vì nó chỉ tải khi sẵn sàng.

**B. Push Strategy (Server chủ động)**

Server gửi một tín hiệu (thông qua MQTT hoặc Firebase Cloud Messaging) ra lệnh cho thiết bị bắt đầu cập nhật.

Ưu điểm: Cập nhật ngay lập tức cho toàn bộ hệ thống khi có biến cố khẩn cấp.

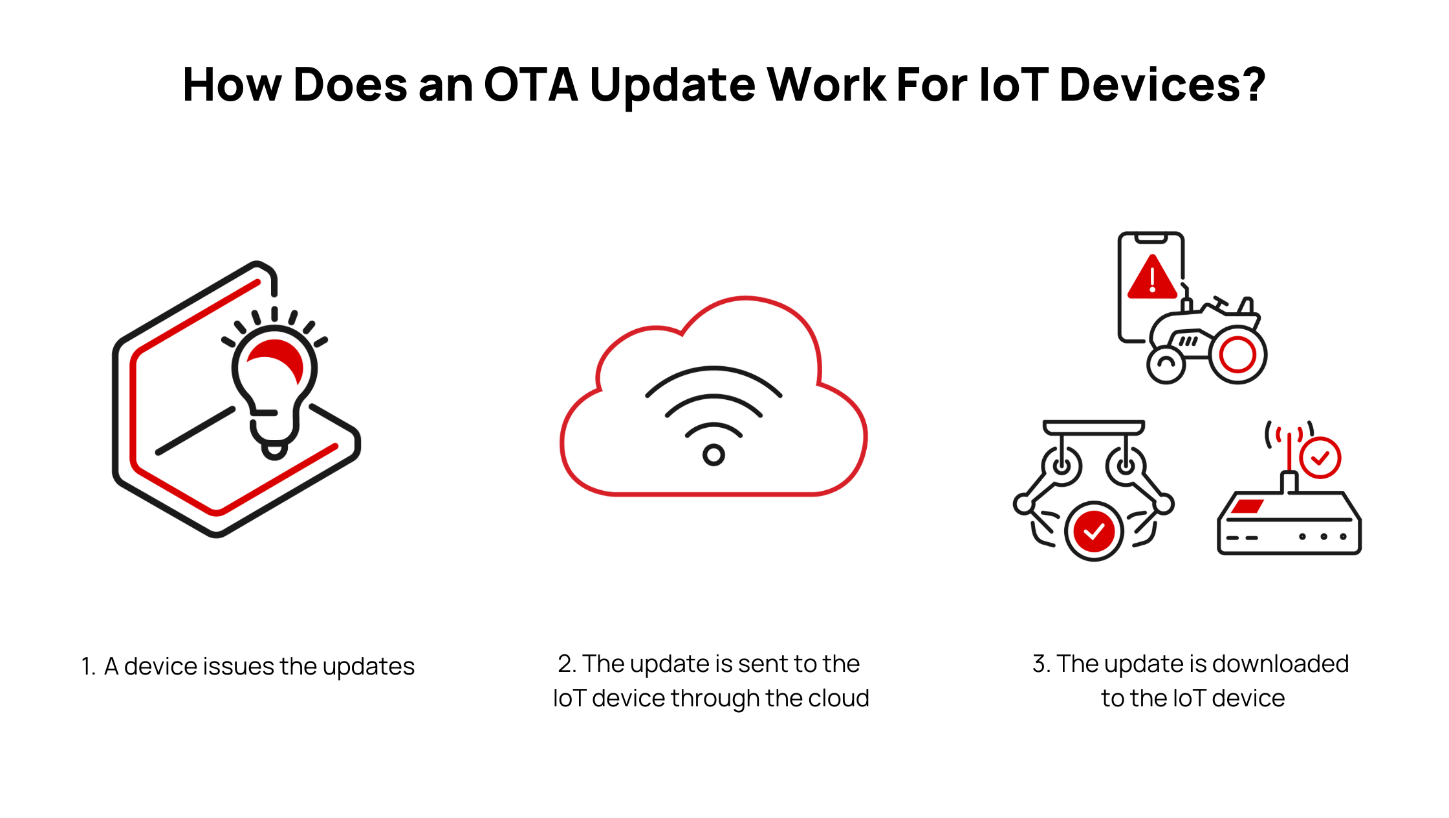
**Cơ chế hoạt động an toàn (Dual Bank Partition)**

Để tránh tình trạng thiết bị biến thành "cục gạch" (brick) nếu đang cập nhật mà mất điện, các dòng ESP32 sử dụng cơ chế Partition Switching:

Bộ nhớ Flash được chia thành 2 vùng: OTA\_0 và OTA\_1.

Nếu firmware hiện tại chạy ở OTA\_0, bản mới sẽ được tải vào OTA\_1.

Sau khi tải xong và kiểm tra mã lỗi (Checksum), thiết bị mới đổi cờ (Boot flag) để khởi động từ OTA\_1. Nếu lỗi, nó vẫn chạy bản cũ ở OTA\_0.



**Hình 4: Quy trình OTA cho các thiết bị IoT**

#### **Điều khiển qua Web UI Dashboard**

Nhóm đã thực hiện setup cho ESP WROOM32 hoạt động hết tiềm năng khi vừa ở trong chế độ AP và STA song song để thực hiện phát mạng local và đồng thời connect với mạng Wi-Fi LAN, bên cạnh đó nhóm đã biến thiết bị trở thành 1 Web Server và deploy trang Web UI này, giúp users dễ dàng điều khiển hệ thống.

**Cơ chế cập nhật dữ liệu thời gian thực (Real-time Update):** Giao diện Web không sử dụng phương pháp tải lại trang (reload) truyền thống gây giật lag. Thay vào đó, mã Javascript được nhúng trong Web sử dụng thư viện Firebase SDK (import { getDatabase... }) để lắng nghe sự thay đổi dữ liệu (onValue) trực tiếp từ Database. Khi có dữ liệu mới từ cảm biến, các thẻ HTML (như nhiệt độ, độ ẩm) sẽ tự động cập nhật số liệu ngay lập tức mà không làm gián đoạn trải nghiệm người dùng



**Hình 5: Web UI điều khiển hệ thống qua mạng local**

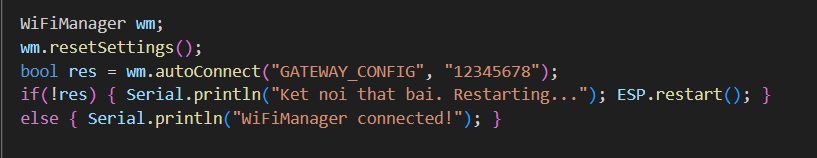
#### **Cơ chế tự động kết nối lại khi gián đoạn mạng**

**Lý do chọn tính năng này**

* Trong các hệ thống IoT giám sát 24/7, sự ổn định của kết nối mạng là yếu tố sống còn. Mạng Wi-Fi trong thực tế có thể bị gián đoạn do nhiều nguyên nhân như: Router bị khởi động lại, nhiễu sóng, hoặc mất điện cục bộ.
* Nếu không có cơ chế tự động kết nối lại, khi xảy ra sự cố mạng, thiết bị sẽ bị treo hoặc mất liên lạc vĩnh viễn cho đến khi người dùng phải can thiệp thủ công (nhấn nút Reset). Điều này gây bất tiện lớn và làm gián đoạn quá trình thu thập dữ liệu.
* Do đó, hệ thống được thiết kế với khả năng "tự phục hồi" kết nối ngay khi mạng Internet có trở lại.

**Cơ chế thực hiện trên Node Gateway (Trung tâm)**

* Gateway sử dụng tính năng quản lý kết nối chuyên dụng được tích hợp trong thư viện Firebase ESP Client và WiFiManager.
* Ngay từ khi khởi động, Gateway sử dụng WiFiManager để tìm kiếm và kết nối với mạng Wi-Fi đã được lưu trong bộ nhớ EEPROM/Flash. Nếu kết nối thất bại, nó sẽ tự động thử lại thay vì dừng hoạt động.
* Trong quá trình vận hành, hệ thống kích hoạt chế độ "Firebase.reconnectWiFi(true)". Đây là một cờ cấu hình quan trọng, cho phép thư viện chạy ngầm một tiến trình giám sát trạng thái kết nối.
* Khi phát hiện tín hiệu Wi-Fi bị mất, thư viện sẽ tự động thực hiện quy trình bắt tay lại (re-handshake) với Router và thiết lập lại phiên làm việc SSL với máy chủ Firebase mà không cần viết thêm mã lệnh phức tạp trong vòng lặp chính.



**Hiệu quả đạt được**

* Loại bỏ hoàn toàn sự phụ thuộc vào thao tác khởi động lại thủ công của con người khi mạng chập chờn.
* Đảm bảo tính liên tục của dữ liệu gửi lên máy chủ, giảm thiểu các khoảng trống dữ liệu trong biểu đồ giám sát.
* Tăng độ tin cậy của hệ thống cảnh báo từ xa, đảm bảo các email cảnh báo lũ lụt hoặc rò rỉ khí gas luôn được gửi đi ngay khi có kết nối mạng.

#### **Giao thức UDP**

**Khái niệm:** UDP (User Datagram Protocol - Giao thức dữ liệu người dùng) là một giao thức cốt lõi nằm trong tầng Giao vận (Transport Layer) của bộ giao thức Internet (IP). Không giống như TCP (Transmission Control Protocol), UDP hoạt động theo cơ chế "phi kết nối"

**Đặc điểm kỹ thuật nổi bật:**

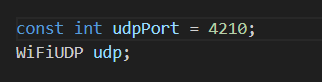
* **Không thiết lập kết nối :** UDP không yêu cầu quá trình "bắt tay 3 bước" (handshake) giữa bên gửi và bên nhận trước khi truyền dữ liệu. Nó chỉ đơn giản là gửi các gói tin (datagram) đi.
* **Không đảm bảo độ tin cậy :** UDP không có cơ chế xác nhận (ACK) để biết gói tin đã đến nơi hay chưa, cũng không tự động gửi lại (retransmit) khi gói tin bị mất, và không đảm bảo thứ tự đến của các gói tin.
* **Tốc độ cao và độ trễ thấp:** Do loại bỏ các thủ tục kiểm soát lỗi và thiết lập kết nối phức tạp, header của UDP rất nhỏ (chỉ 8 bytes so với 20 bytes của TCP), giúp tiết kiệm băng thông và giảm độ trễ xử lý.
* **Hỗ trợ Broadcast và Multicast:** UDP cho phép gửi tin nhắn đến nhiều thiết bị trong mạng cùng lúc, điều mà TCP không hỗ trợ trực tiếp.

**Tại sao chọn UDP cho IoT?** Trong các ứng dụng IoT thời gian thực (Real-time), việc nhận dữ liệu mới nhất (ví dụ: nhiệt độ hiện tại) quan trọng hơn là việc nhận lại dữ liệu cũ bị mất. Nếu một gói tin nhiệt độ bị mất, hệ thống chỉ cần đợi gói tiếp theo sau vài mili-giây thay vì dừng lại để yêu cầu gửi lại gói cũ.

**Ứng dụng UDP trong hệ thống Home-X**

Trong dự án, giao thức UDP được triển khai làm kênh giao tiếp cốt lõi giữa Sensor Node và Gateway với các đặc điểm kỹ thuật sau:

* Thư viện và Cấu hình: Hệ thống sử dụng thư viện **WiFiUdp.h** thuộc bộ ESP32 Core và thiết lập cổng giao tiếp (Port) 4210 thống nhất trên cả hai thiết bị để đảm bảo khả năng nhận diện gói tin chính xác trong mạng nội bộ
* Kỹ thuật đóng gói dữ liệu (Struct Packing): Thay vì xử lý chuỗi ký tự (String) gây tốn tài nguyên, nhóm sử dụng cấu trúc dữ liệu struct\_message được định nghĩa đồng bộ ở cả hai đầu gửi và nhận. Cấu trúc này chứa toàn bộ các trường dữ liệu như giá trị gas (mq2\_val), nước (water1\_val), nhiệt độ, độ ẩm và góc Servo, giúp tối ưu hóa băng thông bằng cách gửi tất cả trong một gói tin duy nhất
* Cơ chế truyền dữ liệu lên (Uplink): Tại Sensor Node, hàm sendDataUDP thực hiện đọc cảm biến, đóng gói struct và gửi nguyên khối dữ liệu (udp.write) đến địa chỉ IP cố định của Gateway là 192.168.4.1. Phía Gateway sử dụng hàm udp.parsePacket() trong vòng lặp chính để liên tục kiểm tra và cập nhật dữ liệu lên Firebase ngay khi nhận được gói tin hợp lệ
* Cơ chế điều khiển xuống (Downlink): Khi nhận tín hiệu từ Web UI, Gateway đóng gói mã lệnh vào cấu trúc struct\_command (với các mã quy ước: 1-Nâng, 2-Hạ, 3-Reset...) và gửi ngược lại cho Sensor Node thông qua hàm sendCommandToSensor, đảm bảo phản hồi tức thời các tác vụ điều khiển
* Hiệu năng và Độ trễ: Chu kỳ gửi dữ liệu được thiết lập ở mức 300ms (intervalSend = 300). Tần suất cập nhật cao này giúp hệ thống đạt được trải nghiệm thời gian thực (Real-time) trên Web và LCD, đồng thời khắc phục nhược điểm về độ tin cậy của giao thức UDP (dữ liệu mới sẽ nhanh chóng thay thế dữ liệu cũ nếu có gói tin bị mất do nhiễu).

****

**Hình 6: Cổng giao tiếp UDP chung**

1. **Môi trường lập trình và Google Firebase**
2. **Arduino IDE**

Arduino IDE là một phần mềm, được sử dụng chủ yếu để viết và biên dịch mã vào các module Arduino, ESP32, ESP8266. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện.

A close-up of a logo

Description automatically generated

Khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và gửi đến bo mạch bằng cáp USB. Mỗi bo Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết. Chức năng của từng phần như sau:

* Nút kiểm tra chương trình: Dùng để kiểm tra xem chương trình được viết có lỗi không. Nếu chương trình bị lỗi thì phần mềm Arduino IDE sẽ hiển thị thông tin lỗi ở vùng thông báo thông tin.
* Nút nạp chương trình xuống bo Arduino: Dùng để nạp chương trình được viết xuống mạch Arduino. Trong quá trình nạp, chương trình sẽ được kiểm tra lỗi trước sau đó mới thực hiện nạp xuống mạch Arduino.
* Hiển thị màn hình giao tiếp với máy tính: Khi nhấp vào biểu tượng cái kính lúp thì phần giao tiếp với máy tính sẽ được mở ra. Phần này sẽ hiển thị các thông số mà người dùng muốn đưa lên màn hình. Muốn đưa lên màn hình phải có lệnh Serial.print() mới có thể đưa thông số cần hiển thị lên màn hình.
* Vùng lập trình: Vùng này để người lập trình thực hiện việc lập trình cho chương trình của mình.
* Vùng thông báo thông tin: Có chức năng thông báo các thông tin lỗi của chương trình hoặc các vấn đề liên quan đến chương trình được lập.
* Sử dụng một số menu thông dụng trên phần mềm Arduino IDE: Có vài menu trong phần mềm IDE, tuy nhiên thông dụng nhất vẫn là menu File, ngoài những tính năng như mở một file mới hay lưu một file, phần menu này có một mục đáng chú ý là Example. Phần Example (ví dụ) đưa ra các ví dụ sẵn để người lập trình có thể tham khảo, giảm bớt thời gian lập trình. Hình bên dưới thể hiện việc chọn một ví dụ cho led chớp tắt (blink) để nạp cho mạch Arduino. Ví dụ về led chớp tắt này thường được dùng để kiểm tra bo khi mới mua về.

1. **Google Firebase**

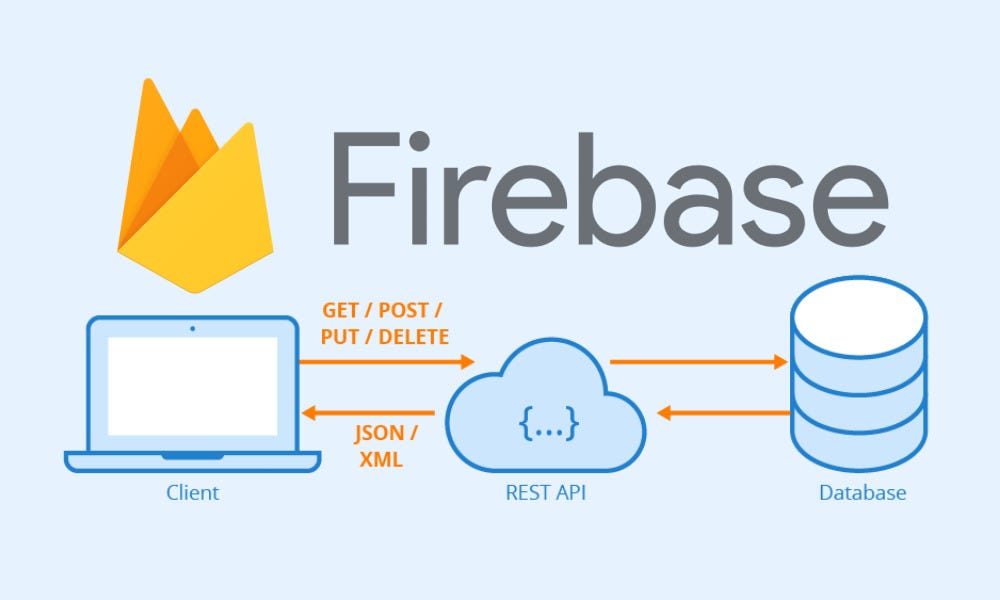
Firebase là nền tảng do Google phát triển, hỗ trợ xây dựng và quản lý ứng dụng web cũng như mobile một cách nhanh chóng. Nó cung cấp hạ tầng backend dưới dạng dịch vụ, giúp lập trình viên tập trung vào phát triển tính năng thay vì lo về máy chủ.

**Firebase là gì?**

Một nền tảng phát triển ứng dụng với nhiều dịch vụ tích hợp như cơ sở dữ liệu, xác thực, lưu trữ, phân tích.

**Firebase hoạt động như thế nào?**

Firebase hoạt động theo mô hình Backend-as-a-Service (BaaS). Nhà phát triển chỉ cần kết nối ứng dụng với SDK của Firebase, dữ liệu và các chức năng sẽ được xử lý trên hạ tầng của Google.

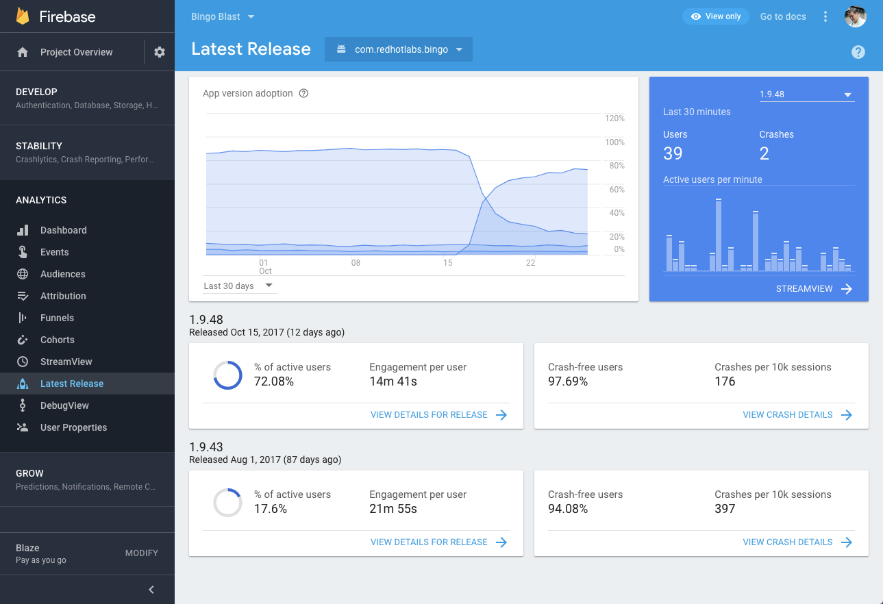


**Tính năng, đặc điểm**

* Realtime Database và Firestore để lưu trữ dữ liệu thời gian thực.
  + Authentication hỗ trợ đăng nhập bằng email, số điện thoại, mạng xã hội.
  + Cloud Storage để lưu trữ file.
  + Hosting cho web app.
  + Analytics và Crashlytics để theo dõi hành vi người dùng, lỗi ứng dụng.

**Các tiện ích trên giao diện được nhà cung cấp dễ sử dụng**

Firebase Console có giao diện trực quan, dễ thao tác. Người dùng có thể quản lý dữ liệu, theo dõi log, cấu hình dịch vụ mà không cần viết nhiều lệnh phức tạp.



1. **Phát triển phần mềm**
2. **Xử lý analog (ESP32)**

**1. Thu thập dữ liệu cảm biến (Data Acquisition)**

Đây là chức năng cơ bản nhất để hiểu trạng thái ngôi nhà:

* Đo nhiệt độ & độ ẩm (readDHTAndDisplay): Sử dụng cảm biến DHT11 để lấy thông số môi trường. Dữ liệu này được cập nhật mỗi 2 giây lên màn hình LCD tại chỗ.
* Phát hiện rò rỉ Gas (readMQ2AndAlarm): Đọc giá trị từ cảm biến MQ2. Nếu vượt ngưỡng (3000), nó sẽ kích hoạt còi báo (Buzzer) ngay lập tức để cảnh báo tại chỗ.
* Phát hiện ngập lụt (chong\_lu): Cảm biến nước giám sát mực nước. Nếu phát hiện nước dâng cao vượt ngưỡng, hệ thống sẽ tự động chuyển sang chế độ bảo vệ.
* Phát hiện chuyển động (movement\_detected): Sử dụng cảm biến PIR (Move Sensor) để tự động bật/tắt đèn hành lang.

**2. Xử lý logic tự động (Local Automation)**

Node sensor này có khả năng tự ra quyết định mà không cần chờ lệnh từ Gateway:

* Phản ứng lũ lụt: Khi water\_level > 3000, nó tự động điều khiển Servo quay đến góc 150 độ (đóng cửa/nâng rào chắn) một cách mượt mà thông qua hàm changeServoSmoothly.
* Chiếu sáng thông minh: Tự động bật LED khi có người đi qua và tắt khi không còn chuyển động (nếu không ở chế độ điều khiển tay).

**3. Giao tiếp không dây nội bộ (UDP Communication)**

Đây là phần quan trọng nhất để kết nối với bộ điều khiển trung tâm:

* Gửi dữ liệu (sendDataUDP): Đóng gói toàn bộ thông số (nhiệt độ, gas, góc quay servo, trạng thái đèn...) vào một struct\_message và gửi tới IP của Gateway qua giao thức UDP mỗi 0.3 giây.
* Nhận lệnh (receiveCommandUDP): Lắng nghe các gói tin điều khiển từ Gateway. Các lệnh bao gồm:

1: Nâng rào chắn.

2: Hạ rào chắn.

3: Reset hệ thống.

5: Bật đèn cưỡng bách (Manual Mode).

**4. Tương tác tại chỗ (Human-Machine Interface - HMI)**

Màn hình LCD 16x2: Hiển thị trạng thái kết nối Wi-Fi và các thông số nhiệt độ, độ ẩm thời gian thực.

Nút bấm vật lý:

**up\_servo/down\_servo:** Cho phép người dùng điều khiển rào chắn bằng tay tại chỗ.

**reset\_button:** Khôi phục hệ thống về trạng thái an toàn ban đầu.

**turn\_off\_system:** Nút nguồn mềm để bật/tắt toàn bộ hoạt động của node.

**5. Quản lý trạng thái hệ thống (System Management)**

Chế độ Manual vs Auto: Khi người dùng can thiệp bằng nút bấm hoặc lệnh từ App, biến servo\_hand sẽ chuyển về 0 để ưu tiên lệnh điều khiển tay, tạm dừng chế độ tự động chống lụt cho đến khi được Reset.

Chuyển động mượt (Smooth Motion): Hàm changeServoSmoothly giúp Servo không quay quá nhanh, tránh gây hỏng hóc cơ khí cho rào chắn hoặc cửa.

1. **Wi-Fi (ESP WROOM 32)**

**1. Quản lý kết nối đa tầng (Connectivity Manager)**

Gateway này hoạt động ở chế độ kép (WIFI\_AP\_STA):

* Chế độ Station (STA): Kết nối với Wi-Fi thông qua WiFiManager. Đây là đường dẫn để nó ra Internet nói chuyện với Firebase và Google Script (Email).
* Chế độ Access Point (AP): Nó tự phát ra một mạng Wi-Fi riêng (ESP32\_GATEWAY) để con Node Sensor kết nối vào. Điều này giúp hệ thống hoạt động độc lập, không phụ thuộc hoàn toàn vào router.

**2. Trạm trung chuyển dữ liệu (Data Bridge - UDP to Cloud)**

* Lắng nghe UDP: Nó chờ đợi gói tin từ Node Sensor. Khi nhận được, nó dùng hàm udp.remoteIP() để ghi nhớ địa chỉ của con Sensor (để sau này còn gửi lệnh ngược lại).
* Đẩy dữ liệu lên Firebase: Sau khi nhận gói dữ liệu từ Sensor, nó ngay lập tức đóng gói thành định dạng JSON và đẩy lên Firebase Realtime Database qua hàm pushDataToFirebase().

**3. Trung tâm xử lý cảnh báo từ xa (Alerting System)**

Trong khi Sensor Node chỉ báo động bằng còi tại chỗ, thì Gateway lo việc báo động cho dù đang ở bất cứ đâu:

* Gửi Email: Qua hàm sendEmail(), nó gọi một Google Apps Script để gửi thư thông báo khi có rò rỉ Gas hoặc ngập lụt.
* Quản lý thời gian gửi: Nó có biến emailInterval (5 phút) để tránh việc gửi hàng trăm email rác liên tục khi sự cố chưa được khắc phục.

**4. Điều phối lệnh điều khiển (Command Dispatcher)**

Khi nhấn nút trên Web UI:

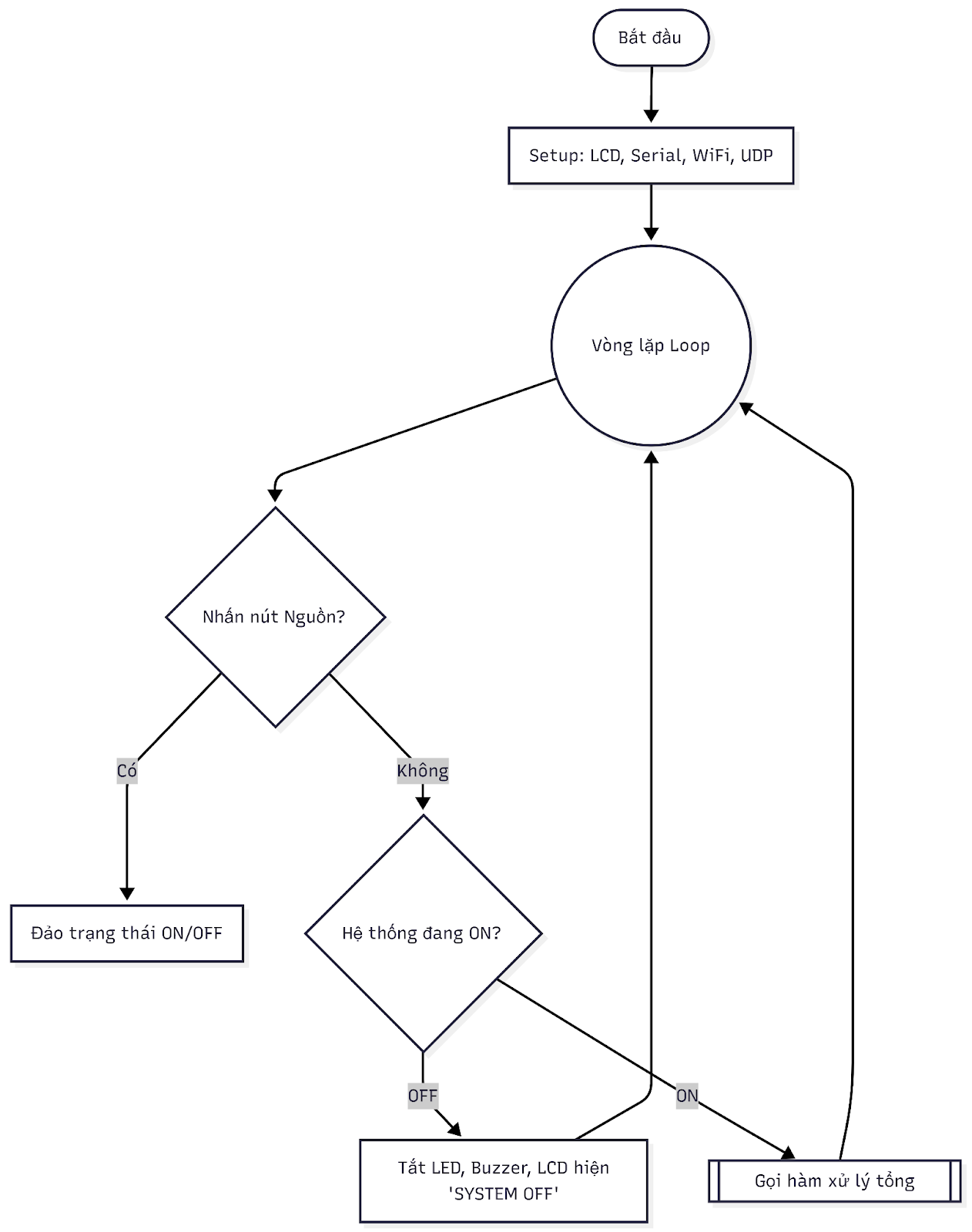
* Lệnh được ghi vào Firebase (/control/cmd).
* Gateway liên tục kiểm tra (checkFirebaseCommand) mỗi 0.5 giây.
* Nếu có lệnh mới, nó dùng sendCommandToSensor() để bắn lệnh đó qua giao thức UDP tới đích danh con Sensor đang kết nối.

1. **Repository**

Repo github: <https://github.com/ngthvinh08/IoT-Final-Project.git>

1. **Lưu đồ thuật toán**

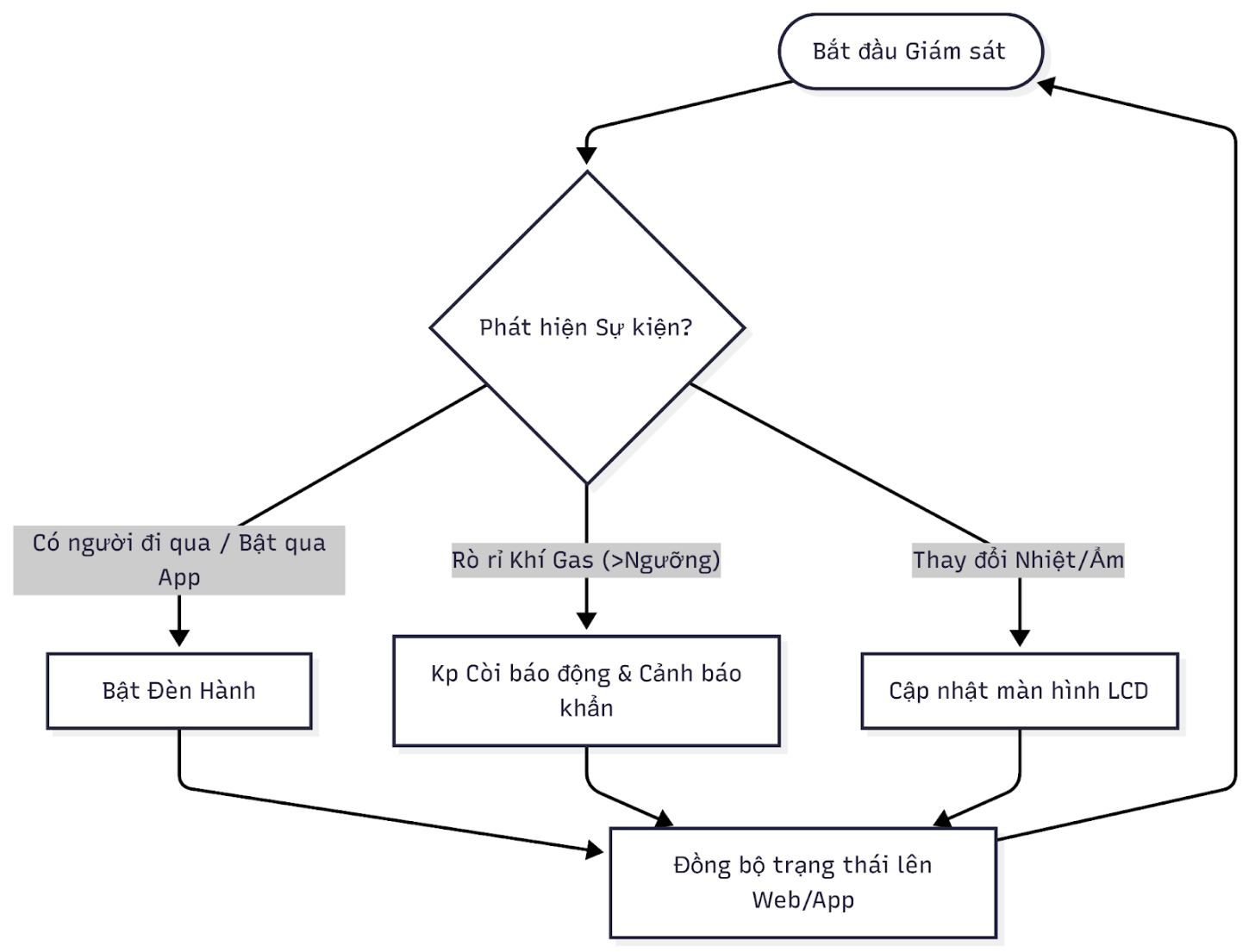
### **Luồng hoạt động tổng quát**



Lưu đồ này mô tả quá trình khởi động và vòng lặp chính kiểm soát trạng thái hoạt động của toàn bộ hệ thống. Quá trình hoạt động được chia thành các bước sau:

* Bước 1: Khởi tạo hệ thống (Initialization). Ngay khi được cấp nguồn, vi điều khiển ESP32 thực hiện thiết lập các thông số ban đầu bao gồm giao tiếp Serial, khởi động màn hình LCD, thiết lập chế độ cho các chân GPIO (Input/Output), đưa Servo về góc mặc định 15 độ và khởi động cảm biến nhiệt độ - độ ẩm DHT.
* Bước 2: Thiết lập kết nối mạng. Hệ thống tiến hành kết nối tới WiFi Gateway. Nếu kết nối thất bại, chương trình sẽ tạm dừng 500ms và thử lại liên tục cho đến khi thành công. Sau khi có kết nối WiFi, hệ thống sẽ khởi tạo kết nối UDP Server để sẵn sàng truyền nhận dữ liệu.
* Bước 3: Vòng lặp chính kiểm tra nút nguồn. Tại vòng lặp vô hạn, hệ thống liên tục kiểm tra trạng thái của nút nhấn nguồn (Pin 32). Nếu phát hiện nút được nhấn, biến trạng thái hoạt động sẽ được đảo ngược giá trị (từ đang bật sang tắt hoặc ngược lại).
* Bước 4: Điều khiển trạng thái hoạt động. Dựa vào biến trạng thái, hệ thống chia làm hai trường hợp. Trường hợp hệ thống tắt, vi điều khiển ngắt toàn bộ các thiết bị đầu ra như đèn LED, còi báo động, hiển thị thông báo trạng thái tắt lên màn hình LCD và quay lại đầu vòng lặp. Trường hợp hệ thống bật, vi điều khiển gọi hàm xử lý trung tâm để thực hiện các chức năng đo đạc và điều khiển.

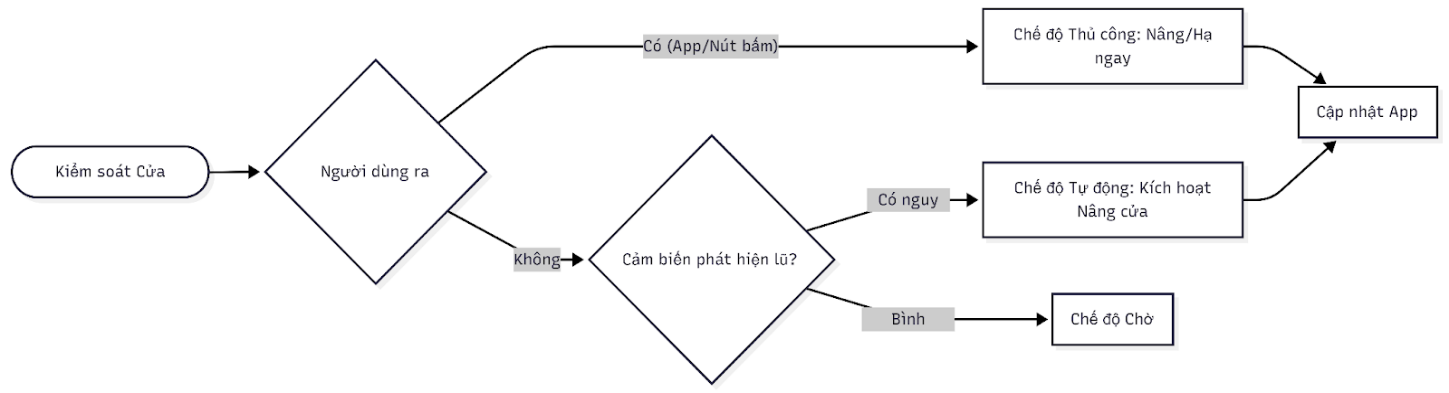
### **Lưu đồ xử lý tín hiệu cảm biến và truyền thông IoT**



Lưu đồ này mô tả các tác vụ giám sát môi trường và truyền dữ liệu chạy nền bên trong hàm xử lý trung tâm.

* Xử lý Reset hệ thống: Hệ thống kiểm tra nút nhấn Reset cứng. Khi nút này được nhấn, toàn bộ các biến trạng thái quan trọng như chế độ tự động, cờ báo lũ và các hiển thị trên LCD sẽ được đặt lại về giá trị mặc định ban đầu.
* Giám sát an ninh và chiếu sáng: Hệ thống đọc tín hiệu từ cảm biến chuyển động PIR và trạng thái điều khiển đèn từ ứng dụng. Đèn LED sẽ được bật sáng nếu phát hiện có chuyển động hoặc có lệnh bật đèn thủ công. Ngược lại, đèn sẽ tắt để tiết kiệm năng lượng.
* Giám sát khí gas: Vi điều khiển đọc giá trị analog từ cảm biến MQ2. Nếu nồng độ khí vượt quá ngưỡng an toàn cho phép, còi báo động sẽ được kích hoạt liên tục để cảnh báo nguy hiểm.
* Định thời và truyền thông đa nhiệm: Để đảm bảo hệ thống hoạt động mượt mà, thuật toán sử dụng bộ định thời gian thực mà không gây ngắt quãng chương trình. Cứ mỗi 2 giây, hệ thống đọc dữ liệu nhiệt độ, độ ẩm từ cảm biến DHT11 và cập nhật lên màn hình LCD. Song song đó, cứ mỗi 300 mili-giây, hệ thống thu thập toàn bộ dữ liệu cảm biến và trạng thái thiết bị, đóng gói thành cấu trúc tin và gửi về Gateway thông qua giao thức UDP.

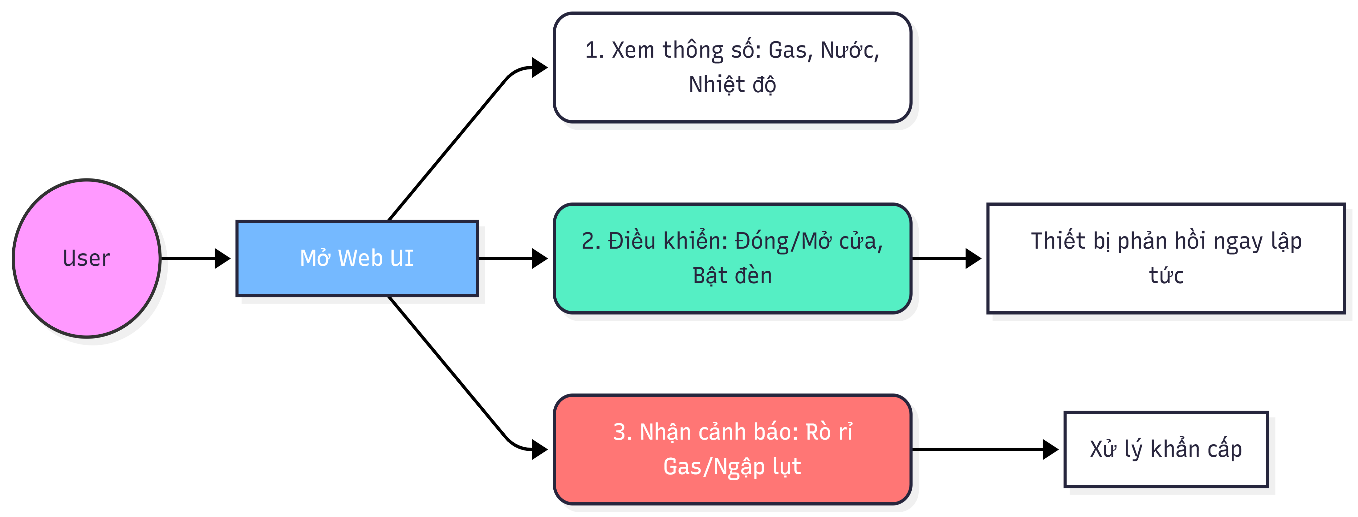
### **Lưu đồ giải thuật điều khiển Servo và cơ chế chống lũ**



Thuật toán điều khiển Servo đóng vai trò quan trọng trong việc vận hành cửa đập, hoạt động dựa trên cơ chế ưu tiên theo thứ tự: Lệnh điều khiển từ xa, Nút nhấn vật lý và Chế độ tự động.

* Ưu tiên 1 (Lệnh UDP từ xa): Hệ thống kiểm tra gói tin nhận được từ Gateway. Nếu nhận được lệnh nâng hoặc hạ, hệ thống lập tức chuyển sang chế độ điều khiển thủ công và thực hiện quay Servo theo yêu cầu.
* Ưu tiên 2 (Nút nhấn vật lý): Trong trường hợp không có lệnh từ mạng, hệ thống kiểm tra các nút nhấn tại trạm. Nếu người dùng nhấn nút nâng hoặc hạ, hệ thống cũng sẽ chuyển sang chế độ thủ công và điều khiển động cơ Servo tương ứng với nút nhấn.
* Ưu tiên 3 (Chế độ tự động): Nếu không có bất kỳ tác động nào từ con người (không có lệnh mạng và không có nút nhấn), hệ thống sẽ hoạt động ở chế độ tự động. Vi điều khiển đọc giá trị từ cảm biến mực nước. Khi giá trị cảm biến vượt ngưỡng cài đặt (mức 3000) và chưa có trạng thái phát hiện lũ trước đó, hệ thống sẽ tự động kích hoạt cơ chế chống lũ bằng cách nâng Servo lên góc 150 độ để ngăn nước.

# **CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM**

1. **User flow – Trải nghiệm người dùng** 

* Xem (Monitor): Người dùng vào Web UI đề thấy các chỉ số "Real-time" từ cảm biến gửi lên.
* Điều khiển (Control): Người dùng bấm nút trên giao diện -> Cửa đóng/mở hoặc Đèn bật/tắt.
* Cảnh báo (Alert): Nếu có nguy hiểm, Web sẽ chủ động "nhảy" thông báo (Push Notification) để người dùng biết.

1. **Kết quả vận hành hệ thống**
   1. **Linh kiện sử dụng**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên linh kiện | Chức năng | Hình ảnh thực tế |
| 1 | Vi điều khiển **ESP WROOM 32** | Vi điều khiển chính, xử lý dữ liệu từ cảm biến và điều khiển các thiết bị ngoại vi, hỗ trợ kết nối Wi-Fi. |  |
| 2 | Cảm biến gas MQ2 | Phát hiện khí gas như LPG, butan, methane, khói, rượu... | A close-up of a blue and silver device  Description automatically generated |
| 3 | Cảm biến chuyển động | Cảm biến phát hiện chuyển động của vật thể |  |
| 4 | Cảm biến nhiệt độ độ ẩm DHT11 | Đo nhiệt độ (°C) và độ ẩm (%RH) của môi trường | A blue electronic device with a red button  Description automatically generated |
| 5 | Relay 2 kênh | Dùng để điều khiển thiết bị như quạt và máy bơm | A close-up of a relay module  Description automatically generated |
| 6 | Quạt | Hút khí gas và thông gió giảm nhiệt độ phòng | A black square fan with red and black wires  Description automatically generated |
| 7 | LCD | LCD 16x2 Hiển thị thông số cảm biến | Display LCD 16x2 con Fondo Azul | Tienda en Linea – Componentes ... |
| 8 | Động cơ servo | Đóng/Mở cửa chống ngập | Using the SG90 Servo Motor With an Arduino - Electronics-Lab |
| 9 | Buzzer | Cảnh báo khi khí gas rò rỉ vượt ngưỡng | Piezo Buzzer - Passive Buzzer Latest Price, Manufacturers & Suppliers |
| 10 | Cảm biến mực nước | Cảnh báo mực nước dâng cao |  |

* 1. **Video thực nghiệm**

Link drive: [Demo\_IoT\_HomeX\_Project](https://drive.google.com/drive/folders/1d8RKhAwyyxSs4hjf4Zv9_2kgq0v-kmBo?usp=sharing)

1. **Những đề xuất nâng cấp hệ thống**

Để hệ thống hoàn thiện hơn và có khả năng thương mại hóa, nhóm đề xuất các hướng nâng cấp sau:

* **Tích hợp AI:** Sử dụng dữ liệu lịch sử trên Firebase để huấn luyện mô hình máy học (Machine Learning) đơn giản, giúp dự đoán xu hướng nhiệt độ hoặc nguy cơ ngập lụt trước khi nó xảy ra.
* **Điều khiển bằng giọng nói:** Tích hợp thêm Google Assistant hoặc Amazon Alexa để điều khiển các thiết bị trong nhà thông minh bằng giọng nói, tăng tính tiện dụng.
* **Dự phòng năng lượng:** Bổ sung module pin dự phòng và mạch sạc năng lượng mặt trời để hệ thống cảnh báo lũ vẫn hoạt động được khi bị cắt điện do bão lũ.

1. **Kết luận**

Hệ thống Home-X được xây dựng đã hoàn thành tốt các mục tiêu đề ra, thiết lập được một hạ tầng IoT ổn định dựa trên sự phối hợp chặt chẽ giữa các tầng thiết bị. Việc triển khai kiến trúc Dual-MCU với giao thức UDP không dây giúp tối ưu hóa hiệu suất xử lý, cho phép hệ thống duy trì khả năng tự động hóa cục bộ cực nhanh trong các tình huống khẩn cấp như rò rỉ gas hay ngập lụt, ngay cả khi mất kết nối Internet. Đồng thời, việc tích hợp thành công nền tảng Cloud Firebase cùng tính năng cập nhật OTA đã tạo nên một giải pháp giám sát từ xa hiện đại, tiện lợi và có khả năng bảo trì linh hoạt. Mặc dù vẫn còn dư địa để phát triển về mặt bảo mật đường truyền và tích hợp trí tuệ nhân tạo, nhưng về cơ bản, dự án đã khẳng định được tính khả thi và tiềm năng ứng dụng cao trong việc nâng cao sự an toàn cũng như chất lượng sống cho các hộ gia đình hiện nay.