**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Cao Sỹ Trung**

**MSV: 14020488**

**Thiết kế hệ thống IoT**

**đo đạc thông tin môi trường nhà trồng nấm**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**HÀ NỘI - 2018**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Cao Sỹ Trung**

**Thiết kế hệ thống IoT**

**đo đạc thông tin môi trường nhà trồng nấm**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Cán bộ hướng dẫn: TS. Dương Lê Minh**

**HÀ NỘI - 2018**

**LỜI CÁM ƠN**

Lời đầu tiên, em xin bày tỏ lòng biết ơn sâu sắc nhất tới TS. Dương Lê Minh, người đã tận tình chỉ bảo, hướng dẫn, động viên và giúp đỡ em trong suốt quá trình thực hiện luận luận tốt nghiệp. Em cũng xin chân thành cảm ơn các thầy cô giáo trong khoa Công nghệ thông tin nói riêng và các thầy cô giáo Trường đại học Công Nghệ - Đại học Quốc gia Hà Nội đã dạy dỗ, truyền đạt cho em không chỉ những kiến thức chuyên môn mà còn chia sẻ kinh nghiệm về công việc và đời sống. Nhờ những kiến thức và kĩ năng được thầy cô truyền đạt trong suốt quá trình học tập mà em đã có thể tìm hiểu và thực hiện khóa luận một cách suôn sẻ và tốt nhất có thể.

Em xin bày tỏ lòng biết ơn vô hạn tới bố mẹ, gia đình em đã luôn động viên tạo mọi điều kiện tốt nhất trong suốt quá trình em thực hiện khóa luận tốt nghiệp. Cuối cùng em xin gửi lời cảm ơn tới các bạn trong tập thể lớp K59 – CD đã đồng hành, giúp đỡ em trong suốt bốn năm học tập tại trường.

Em xin chân thành cảm ơn.

**LỜI CAM ĐOAN**

Tôi xin cam đoan những kiên thức được trình bày trong luận án này là do tôi tìm hiểu và triển khai dưới sự hướng dẫn của TS. Dương Lê Minh.

Tất cả các tham khảo từ các nghiên cứ liên quan điều được tôi nêu ra một cách rõ ràng từ danh mục tài liệu tham khoả trong luận án này. Trong khoá luận khống có việc sao chép tài liệu, công trình nghiên cứu của người khác mà không chỉ rõ về nguồn gốc của tài liệu.

Hà Nội, ngày tháng năm 2017

Sinh viên

Cao Sỹ Trung

**MỤC LỤC**

[**LỜI CÁM ƠN** 3](#_Toc512269330)

[**LỜI CAM ĐOAN** 4](#_Toc512269331)

[**MỤC LỤC** 5](#_Toc512269332)

[**BẢNG KÝ HIỆU VIẾT TẮT** 7](#_Toc512269333)

[**MỤC LỤC BẢNG BIỂU** 8](#_Toc512269334)

[**MỤC LỤC HÌNH VẼ** 9](#_Toc512269335)

[**MỞ ĐẦU** 10](#_Toc512269336)

[**Bối cảnh** 10](#_Toc512269337)

[**Lý do chọn đề tài** 10](#_Toc512269338)

[**Nội dung đề tài** 11](#_Toc512269339)

[**Tóm tắt khóa luận tốt nghiệp:** 12](#_Toc512269340)

[**Kết cấu của khóa luận** 13](#_Toc512269341)

[**CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN** 14](#_Toc512269342)

[**1.1.** **Giới thiệu tổng quan về IoT** 14](#_Toc512269343)

[**1.1.1.** **Internet of Thing** 14](#_Toc512269344)

[**1.1.2.** **Lịch sử phát triển IoT** 15](#_Toc512269345)

[**1.1.3.** **Xu hướng phát triển của IoT** 16](#_Toc512269346)

[**1.2.** **Tình trạng IoT trong lĩnh vực trồng nấm** 18](#_Toc512269347)

[**1.2.1.** **Giới thiệu chung về tình hình trồng nấm tại Việt Nam** 18](#_Toc512269348)

[**1.2.2.** **Ứng dụng IoT trong trồng nấm** 18](#_Toc512269349)

[**CHƯƠNG 2: GIẢI PHÁP HỖ TRỢ GIÁM SÁT TRỒNG NẤM THÔNG MINH** 18](#_Toc512269350)

[**2.1.** **Đặc tả yêu cầu đối với bài toán thực tế** 18](#_Toc512269351)

[**2.2.** **Quy trình trồng nấm** 18](#_Toc512269352)

[**2.3.** **Thiết kế hệ thống** 18](#_Toc512269353)

[**2.3.1.** **Sơ đồ tổng quát hệ thống** 18](#_Toc512269354)

[**2.3.2.** **Hệ thống phần cứng** 18](#_Toc512269355)

[**2.3.3.** **Hệ thống phần mềm** 18](#_Toc512269356)

[**2.4.** **Phương pháp thu thập dữ liệu sensor** 18](#_Toc512269357)

[**2.5.** **Phương pháp điều khiển** 18](#_Toc512269358)

[**2.6.** **Phương pháp phát hiện bất thường** 18](#_Toc512269359)

[**CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG THỰC TẾ VÀ ĐÁNH GIÁ** 18](#_Toc512269360)

[**3.1.** **Mô hình triển khai** 18](#_Toc512269361)

[**3.1.1.** **Sơ đồ cài đặt thực tế phần cứng** 18](#_Toc512269362)

[**3.1.2.** **Cấu hình cài đặt phần mềm** 18](#_Toc512269363)

[**3.2.** **Các công nghệ sử dụng** 18](#_Toc512269364)

[**3.2.1.** **TimeSeries Database và InfluxDB** 18](#_Toc512269365)

[**3.2.2.** **Microservices** 18](#_Toc512269366)

[**3.2.3.** **MQTT và EMQ** 18](#_Toc512269367)

[**3.2.4.** **Bộ điều khiển Proportional Integral Derivative (PID)** 18](#_Toc512269368)

[**3.2.5.** **Docker** 18](#_Toc512269369)

[**3.2.6.** **Flask** 18](#_Toc512269370)

[**CHƯƠNG 4: TỔNG KẾT** 18](#_Toc512269371)

**BẢNG KÝ HIỆU VIẾT TẮT**

|  |  |
| --- | --- |
| IoT | Internet of Things |
| PID | Proportional Integral Derivative |
| TSDB | Time Series Database |

**MỞ ĐẦU**

**Bối cảnh**

Trong những năm gần đây với sự phát triển vượt bậc của công nghệ kết nối không dây và kỹ thuật điện tử cùng với các nhu cầu ý tưởng tự động hóa công việc cũng như các thiết bị thông minh đã thúc đẩy sự phát triển mạnh mẽ của Internet of Things (IoT) – xu hướng công nghệ của tương lai. IoT thực sự không còn là một khái niệm quá xa lạ nhưng những tiềm năng của nó là vô cùng to lớn và vẫn đang được phát triển từng giờ. Các đối tượng trong hệ sinh thái IoT có thể liên kết, trao đổi dữ liệu và vận hành một cách tự động theo một kịch bản được tạo sẵn mà không cần sự can thiệp trực tiếp từ con người. Đây chính là đặc điểm nổi trội của IoT

Nhắc đến IoT người ta sẽ nhắc đến Smart Home một ứng dụng thực tiễn của IoT đang thu hút sự đầu tư, nghiên cứu nhiều bậc nhất so với các lĩnh vực khác bởi tính thực tiễn của nó. Một căn nhà thông minh sẽ giúp bạn thực hiện một số thao tác, công việc thủ công, mang lại sự tiện lợi chuyên nghiệp, sẽ còn gì tuyệt vời hơn là một căn nhà có thể hiểu và thực hiện được ý muốn của chủ nhân của nó

IoT thực sự đã có tác động đến từng ngành, từng lĩnh vực trong đó có nông nghiệp. Những tác động đó đã đặt ngành nông nghiệp trước thách thức buộc phải triển khai các giải pháp nông nghiệp thông minh nếu không muốn đánh mất thị trường.

**Lý do chọn đề tài**

Trong thực tiễn cuộc sống, thị trường Nấm nói chung được phát triển rất tốt nhưng tình hình Nấm trồng tại Việt Nam lại không được tốt lắm. Nấm tại Việt Nam được người dân tiến hành trồng một cách thủ công và chủ yếu dùng kinh nghiệm là chủ yếu. Mặt khác trong quá trình trồng nấm nói chung thì phải trải qua rất nhiều các quy trình và giai đoạn ứng với các điều kiện khác nhau. Mỗi quá trình đều phải đòi hỏi thông số môi trường xác định để cây nấm có thể phát triển tốt nhất.Chính vì vậy hiệu suất trồng nấm của nông dân thì không cao và thiếu tính ổn định. Chưa kể một số giống nấm vốn không thích hợp với điều kiện trồng tại Việt Nam, nên việc trồng thủ công, đơn giản không thể đáp ứng được nhu cầu của thị trường Nấm tại Việt Nam.

Nhận thấy trên thị trường cũng có rất nhiều các loại sensor với giá thành phải chăng, và không quá phức tạp để có thể sử dụng, độ sai số ở ngưỡng chấp nhận được có thể dùng để thu thập các thông số về môi trường như nhiệt độ, độ ẩm, cường độ ánh sáng, nồng độ CO2… Chính vì vậy ta có thể sử dụng những thiết bị này để xây dựng một hệ thống hỗ trợ trồng nấm thông minh, bù đắp lại những thiếu sót khó khăn của quy trình trồng nấm trong thực tế nhằm nâng cao năng suất trồng nấm và đưa một số giống nấm trở nên phổ biến hơn trên thị trường Việt.

**Nội dung đề tài**

Tại đề tài này, chúng tôi xây dựng một hệ thống bao gồm việc sử dụng các sensor để tiến hành thu thập dữ liệu về các thông số như nhiệt độ môi trường, độ ẩm không khí, nồng độ Co2/O2, cường độ ánh sáng… rồi gửi dữ liệu đó qua giao thức MQTT, lưu trữ dữ liệu trên server và tiến hành hiển thị các thông số đo được trên web, hỗ trợ người dùng theo dõi được quá trình trồng nấm của mình một cách chi tiết. Việc điều chỉnh các thông số điều kiện môi trường đảm bảo điều kiện sinh trưởng và phát triển của nấm được tiến hành xử lý một cách tự động. Bất kỳ trục trặc hay sự cố nào đều được thông báo trực tiếp đến người dùng.

**ỨNG DỤNG IoT VÀO NHÀ TRỒNG NẤM THÔNG MINH**

**Cao Sỹ Trung**

K59 Công nghệ thông tin

**Kết cấu của khóa luận**

Ngoài phần giới thiệu, kết luận và tài liệu tham khảo, nội dung của khóa luận bao gồm 4 chương

Chương 1: Giới thiệu tổng quan về IoT và việc áp dụng IoT trong trồng nấm.

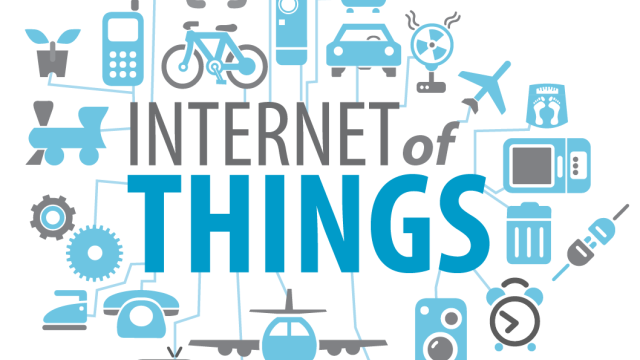
Chương 2: Giải pháp thiết kế mô hình hệ thống để theo dõi và điều khiển nhà trồng nấm.

Chương 3: Triển khai hệ thống và đánh giá kết quả: Sơ đồ lắp đặt, đánh giá.

Chương 4: Tổng kết: Kết quả nghiên cứu, hạn chế, đề xuất.

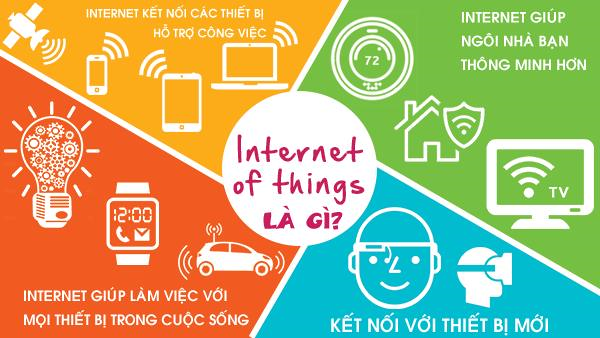
**CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN**

* 1. **Giới thiệu tổng quan về IoT**
     1. **Internet of Thing**



Hình 1.1 Internet of Things

Internet of things( IoT) là một mạng các thiết bị vật lý, phương tiện, đồ gia dụng và các thiết bị khác được nhúng với các thiết bị điện, cảm biến, phần mềm, các thiết bị truyền động và kết nối cho phép các đối tượng kết nối và trao đổi dữ liệu.



*Hình 1.2 Ứng dụng của IoT*

Hay hiểu một cách đơn giản thì IoT là một mạng lưới các vật kết nối với nhau, là một kịch bản của thế giới mà ở đó mỗi vật, người được cung cấp một định danh cho riêng mình. Các thiết bị có thể tự có khả năng truyền tải, trao đổi thông tin và dữ liệu mà không cần có sự tương tác trực tiếp giữa con người với con người hoặc con người với máy tính. Sự kết nối giữa các thiết bị cũng có thể sử dụng kết nối qua wifi, bluetooth, hồng ngoại, Zigbee. Các chuyên gia đã đưa ra các con số chỉ ra rằng các thiết bị có khả năng online tăng 31% từ năm 2016 đến 8,1 tỷ thiết bị vào năm 2017. Các thiết bị có khả năng cảm nhận hoặc được điều khiển từ xa qua một nền tảng network được thiết lập. Các thiết bị có thể là các thiết bị của nhà thông minh, máy giặt, ổ điện, điều hòa, tủ lạnh, … Một mạng lưới IoT có thể có từ 50 đến 100 nghìn tỷ đối tượng cùng kết nối.

* + 1. **Lịch sử phát triển IoT**

Ngày trước khái niệm nhà thông minh với khả năng điều khiển tự động từ xa hoặc điều khiển qua các nút bấm được xuất hiện trên phim ảnh nhưng hiện tại với sự phát triển của khoa học và công nghệ những điều tưởng chừng như viễn tưởng ấy đã xuất hiện trong cuộc sống của chúng ta. Vào những năm 1915 sự xuất hiện của kỹ thuật điện tử cùng phiên bản đầu tiên của mạch bán dẫn đã thay đổi chiều hướng kỹ thuật của con người. Vào những năm 1930 khi các thiết bị dần trở nên phổ biến hơn, người ta đã hình dung những ý tưởng về việc một ngôi nhà có thể hiểu được những ý muốn của chủ nhân nó, nhưng phải đến tận năm 1984 thì khái niệm ‘Smart Home’ mới được hình thành và định nghĩa một cách chính xác. Cùng với đó là sự phát triển vượt bậc của công nghệ truyền tải không dây kết hợp với các kỹ thuật điện hiện đại đã như ánh sáng soi đường cho việc tự động hóa nhà ở. Vào năm 1999,  IoT hay “Internet of Things” được đưa ra bởi Kevin Ashton. Ông là nhà khoa học đã sáng lập ra trung tâm Auto-ID ở Đại học MIT, nơi thiết lập các quy chuẩn toàn cầu cho RFID (một phương thức giao tiếp không dây dùng sóng Radio) cũng như một số loại cảm biến khác. IoT sau đó được dùng nhiều trong các ấn phẩm đến từ các hãng và nhà phân tích.

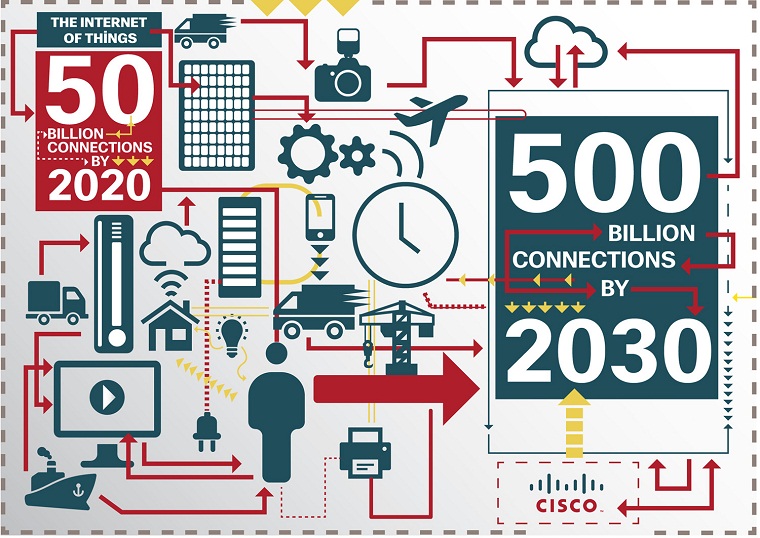
* + 1. **Xu hướng phát triển của IoT**

Một hệ sinh thái IoT thực sự là một thế giới mà trong đó mọi thiết bị đều kết nối được với nhau. Vậy điều gì đã làm nên sự khác biệt của IoT so với các xu hướng công nghệ trước đây? Câu trả lời cho câu hỏi này chính là cảm biến. Mọi thiết bị IoT sẽ đều được tích hợp các cảm biến để xác định các thay đổi về nhiệt độ, độ ẩm, ánh sáng, áp lực, âm thanh, vị trí, … Chúng sẽ trở thành con mắt thành đôi tai điện tử cho những người sử dụng chúng. Các cảm biến này sẽ ghi lại mọi thay đổi về thế giới xung quanh. Sự tương tác giữa các thiết bị với nhau, sử dụng thông tin thu thập về thế giới xung quanh sẽ được tiến hành theo một kịch bản có trước để xử lý mà không cần có sự can thiệp trực tiếp của con người. Với những ý tưởng và nền tảng của IoT hiện nay trên thế giới cũng đã phát triển được các ứng dụng IoT trên nhiều lĩnh vực. Nhà thông minh là từ khóa được tìm kiếm hàng đầu, số công ty hoạt động nghiên cứu về nhà thông minh nhiều hơn bất kỳ lĩnh vực nào khác. Có thể kể đến một số công ty lớn như: Nest, AlertMe hoặc một số công ty đa quốc gia như Philips, Haier, Belkin,…



*Hình 1.3 Hội thảo về chủ đề phát triển Smart Home*

Tiếp theo phải kể đến các thiết bị có thể mang theo (Wearables) cũng đang là một chủ đề nóng với các sản phẩm như đồng hồ thông minh của Apple, máy huấn luyện thông minh của Sony. Ngoài ra ở các lĩnh vực khác cũng phát triển khá mạnh mẽ như lưới điện thông minh (Smart grid) hứa hẹn nâng cao hiệu quả, độ tin cậy và kinh tế của việc sử dụng điện thông qua của người sử dụng điện và cung cấp điện. Internet công nghiệp (Industrial Internet), ô tô kết nối (Connected car), y tế kết nối (Connected health), bán lẻ thông minh (Smart retail), chuỗi cung ứng thông minh (Smart supply chain), trang trại thông minh (Smart farming) cũng đang là các lĩnh vực phát triển tiềm năng, đầy hứa hẹn.



*Hình 1.4 Internet of Things trong tương lai*

Với những nền tảng công nghệ vững chắc cùng những tiềm năng vô cùng to lớn của IoT trong vô vàn lĩnh vực của đời sống mà IoT sẽ còn phát triển vô cùng mạnh mẽ trong tương lai. Dự tính đến năm 2020 sẽ có khoảng 4 tỷ người kết nối với nhau thông qua IoT, 4 ngàn tỷ USD doanh thu từ IoT với hơn 25 triệu ứng dụng, 25 tỷ hệ thống thông minh và hệ thống nhúng, 50 ngàn tỷ gigabytes dữ liệu.

* 1. **Tình trạng IoT trong lĩnh vực trồng nấm**
     1. **Giới thiệu chung về tình hình trồng nấm tại Việt Nam**

Thị trường nấm của Việt Nam có thể đánh giá là một thị trường tiềm năng. Tuy nhiên, đi liền với sự tiềm năng này đang có rất nhiều vấn đề đáng được lưu ý tới. Nhiều người khi mới bắt đầu vào nghề trồng nấm thường nghĩ trồng nấm cũng đơn giản như trồng rau. Nhưng thực tế đã chứng minh đây hoàn toàn là điều không đúng. Phải khẳng định trồng nấm là nghề đòi hỏi trình độ kỹ thuật, công nghệ tay nghề cao. Do đó muốn trồng được nấm người dân bắt buộc phải trải qua các khóa học ngắn hạn hoặc dài hạn tùy theo loại nấm muốn trồng. Nếu đã thiếu công nghệ mà kỹ thuật không có thì không chỉ với cây nấm mà với tất cả các cây trồng khác hiệu quả thu được sẽ không cao.

Trồng nấm tại Việt Nam cũng có nhiều thuận lợi và khó khăn. Về thuận lợi, trong nhiều năm gần đây, các Bộ các Ngành đã có những văn bản chính sách khuyến khích cho nghề trồng nấm phát triển. Mặt khác nước ta là nước nông nghiệp giàu tiềm năng về lâm nghiệp nguồn phế phụ liệu sẵn có khắp nơi dồi dào, rẻ tiền như: rơm rạ, mùn cưa, bã mía, thân cây, lõi ngô … và trong quy trình sản xuất hầu như không có nguyên liệu nào phải nhập ngoại hay khó tìm. Về khí hậu, tuy Việt Nam nằm trọn trong vùng nhiệt đới nhưng khí hậu Việt Nam phân bố thành 2 vùng khí hậu riêng biệt. Miền Bắc mang khí hậu nhiệt đới gió mùa, miền Nam mang khí hậu nhiệt đới. Điều kiện tự nhiên của nước ta (nhiệt độ, độ ẩm, …) rất thích hợp cho nấm phát triển. Các nhóm nấm ưa nhiệt cao, ưa nhiệt trung bình, ưa nhiệt thấp đều có thể trồng được. Về nguồn nhân lực, lực lượng lao động của nước ta rất dồi dào, giá công lao động rẻ. Tính trung bình mọi lao động nông nghiệp mới chỉ dùng 30-40% quỹ thời gian. Mọi lao động phụ đều có thể tham gia vào các công đoạn nuôi trồng nấm. Về thị trường tiêu thụ thì thị trường tiêu thụ các loại nấm ăn và nấm dược liệu ngày càng được mở rộng. Giá bán nấm tươi ở các tỉnh thành phố lớn thường cao gấp 2-3 lần so với giá sản xuất. Nhu cầu ăn nấm của nhân dân trong nước ngày càng tăng do nhiều người đã hiểu được giá trị dinh dưỡng và làm thuốc của nấm.

Tuy nhiên so với nhiều loại thực phẩm khác như thịt, rau thì chất lượng và giá cả của nấm vẫn còn chưa đạt yêu cầu. Chất lượng chưa được ổn định và giá nấm vẫn còn cao so với một số thực phẩm khác và cao hơn sản phẩm nhập khẩu. Chưa kể đó trồng nấm tại Việt Nam cũng có nhiều khó khăn. Công nghệ trồng nấm ở nước ta hiện nay như thế nào? Công nghệ trồng nấm ở nước ta hiện nay còn đang rất đơn giản, chủ yếu là sản xuất nấm thủ công, dùng sức người là chính. Có thể nói là công nghệ thấp cách xa các nước trồng nấm tiên tiến 40-50 năm. Chúng ta chưa có ngành chế tạo máy móc, thiết bị chuyên ngành phụ trợ cho ngành trồng nấm nên mức độ và tốc độ nâng cao công nghệ rất chậm và chưa đạt được kết quả như mong muốn.

Nấm rất nhạy cảm với các yếu tố môi trường như: Nhiệt độ, độ ẩm, ánh sáng, độ thông thoáng, sâu bệnh, nguyên liệu, nguồn nước, v.v... Trồng nấm phải coi là một nghề mà nghề này đòi hỏi phải có sự đầu tư đồng bộ về tri thức, kinh tế và quyết tâm cao mới phát triển bền vững được. Bên cạnh đó, nhận thức của người dân cũng như việc tiếp nhận kỹ thuật nhân giống, nuôi dưỡng, bảo quản, chế biến và tiêu dùng nấm còn nhiều hạn chế.

Khi phải sản xuất nấm công nghiệp chúng ta phải đối mặt với những khó khăn gì? Đó chính là vấn đề về thiếu thốn nguồn nhân lực, thiếu thốn nhà xưởng, thiết bị, công cụ sản xuất nấm…

* + 1. **Ứng dụng IoT trong trồng nấm**

Tất cả những vấn đề được nêu ở trên đã nói lên được thực trạng về ngành trồng nấm ở Việt Nam hiện nay. Sở hữu những thuận lợi nhưng tình hình ngành trồng nấm ở Việt Nam chỉ dừng lại ở mức độ tiềm năng và có thể phát triển được. Nhận thấy những ưu điểm của IoT và độ phát triển của nó tại Việt Nam ta sẽ đặt ra câu hỏi: Tại sao không ứng dụng IoT vào ngành trồng nấm tại Việt Nam. IoT là công nghệ sẽ giải quyết tốt các khó khăn của ngành trồng nấm nói riêng và của nền nông nghiệp Việt Nam nói chung. Hiện nay IoT tại Việt Nam đang rất phát triển, cộng đồng hỗ trợ lớn.

So sánh về tình hình trồng nấm ở các hộ cá thể. Dễ dàng nhận thấy được những ưu điểm của một hệ thống trồng nấm có ứng dụng IoT so với một hệ thống trồng nấm thủ công của những hộ trồng nấm riêng lẻ. Do chỉ là hộ riêng lẻ nên việc áp dụng những công nghệ tiên tiến nhất vào quy trình trồng là điều khó đạt được. Vì thế nên quy trình trồng nấm của những hộ này thường là thủ công, áp dụng kinh nghiệm, kiến thức cá nhân để cảm nhận, đánh giá về quy trình trồng nấm của chính mình. Việc áp dụng hệ thống trồng nấm ứng dụng IoT với những thông số được thiết lặp sẵn trong hệ thống, sự giám sát các thông số môi trường thời gian thực sẽ là ưu điểm, giúp người trồng nấm có thể nắm bắt tình hình hộ trồng nấm của mình hiện tại đang như thế nào một cách chính xác. Những con số được cung cấp là căn cứ để người dùng đưa ra được những quyết định của mình, không còn phải lo lắng về cảm tính của mình đúng hay sai. Quyết định sẽ được đưa ra một các thật khoa học.

Đối với trồng nấm công nghiệp, khi ứng dụng IoT ta sẽ có câu trả lời cho bài toán về nguồn nhân lực. Một hệ thống tự động giám sát và điều khiển về quy trình sẽ giảm thiểu nguồn nhân lực tối đa mà vẫn đảm bảo được về kỹ thuật một cách đồng bộ. Việc đồng bộ giám sát và điều khiển với các thông số chính xác về môi trường đảm bảo sự nhạy cảm của nấm sẽ được giải quyết, năng suất trồng nấm sẽ đồng đều và nâng cao rõ rệt. Bài toán kinh tế cũng sẽ được giải quyết khi cung cấp đủ nấm, giá thành nấm trong nước giảm sẽ có thể đáp ứng được nhu cầu về nấm ăn và nấm dược liệu của nhân dân.

Chưa kể đối với một số giống nấm mới, chưa rõ điều kiện và thông số môi trường để phát triển một cách tốt nhất, hệ thống giám sát và điều khiển tự động còn giúp người trồng thực hiện tính năng trồng thử nghiệm để thí nghiệm tìm ra được những điều kiện để giống nấm được sinh trưởng và phát triển tốt nhất.

Nắm rõ những ưu điểm của việc ứng dụng IoT vào trong nuôi trồng nấm, khóa luận này được thực hiện theo đơn hàng của viện ứng dụng Công nghệ để phát triển một hệ thống ứng dụng IoT giám sát và tự động điều khiển trong quy trình trồng nấm, giúp hạn chế sức lao động và tự động hóa quy trình trồng nấm mà vẫn đảm bảo được chất lượng của cây nấm trồng ra. Đây chính là một dự án nhằm mục đích thúc đẩy sự phát triển của ngành trồng nấm tại Việt Nam.

**CHƯƠNG 2: GIẢI PHÁP HỖ TRỢ GIÁM SÁT TRỒNG NẤM THÔNG MINH**

* 1. **Đặc tả yêu cầu đối với bài toán thực tế**

Xuất phát từ bài toán thực tế về nhu cầu xây dựng một hệ thống hỗ trợ, giám sát và điều khiển tự động quy trình trồng nấm. Thông tin đầu vào của bài toán sẽ là một quy trình đầy đủ khi tiến hành trồng 1 loại nấm xác định. Hệ thống sẽ tiến hành thu thập các thông số về môi trường kể từ khi bắt đầu trồng nấm đó, hiển thị và cảnh báo đến người dùng khi có bất kỳ ngoại lệ nào không đúng với quy trình được thiết lập sẵn. Hệ thống thực hiện 2 chức năng chính. Một là hỗ trợ giám sát các thông số của nhà trồng nấm và tiến hành điều khiển tự động theo quy trình cho trước. Hai là hệ thống hoàn toàn có thể hỗ trợ quy trình trồng nấm thử nghiệm. Đối với những giống nấm mới, khi tiến hành trồng thử cần thử ở nhiều mức khác nhau đối với một thông số, từ kết quả thu được đánh giá và chọn ra được giá trị phù hợp nhất cho thông số đó. Từ những số liệu được cung cấp trong quy trình trồng nấm và những số liệu thu thập được qua các sensor hệ thống sẽ giúp cho người dùng có được báo cáo, thống kê, và lưu lại kết quả cho những đợt trồng tiếp theo. Trong thực tế cũng đã có những sản phẩm hỗ trợ ứng dụng IoT trong nông nghiệp nhưng hỗ trợ trồng nấm trong giai đoạn thử nghiệm thì chưa có. Có thể nói đây là chức năng đáng chú ý của hệ thống. Ngoài ra hệ thống cũng sẽ đáp ứng một số yêu cầu phi chức năng như tính dễ dùng, sử dụng ít tài nguyên mà vẫn đáp ứng được ràng buộc về thời gian và hiệu suất.

* 1. **Quy trình trồng nấm**

Nấm (tên khoa học là Fungi) bao gồm những sinh vật nhân chuẩn dị dưỡng có thành tế bào bằng kitin (chitin). Phần lớn nấm phát triển dưới dạng các sợi đa bào được gọi là sợi nấm (hyphae) tạo nên hệ sợi (mycelium), một số nấm khác lại phát triển dưới dạng đơn bào. Quá trình sinh sản (hữu tính hoặc vô tính) của nấm thường là qua bào tử, được tạo ra trên những cấu trúc đặc biệt hay quả thể. Một số loài mất khả năng tạo nên những cấu trúc sinh sản đặc biệt và nhân lên qua hình thức sinh sản sinh dưỡng. Các điều kiện môi trường cần để nấm sinh trưởng và phát triển thường được quan tâm đến như là nhiệt độ môi trường, độ ẩm không khí, ánh sáng, nồng độ CO2/O2.

Về một quy trình trồng nấm tổng quát ta phải trải qua lần lượt các bước. Mỗi bước đều phải thực hiện với một độ chính xác cao để đảm bảo thành công trong quy trình trồng nấm.

Ta sẽ tiến hành trồng nấm từ những bước đầu tiên. Người trồng tiến hành đi mua giống. Các giống nấm thường được bán theo bịch tính trên đơn vị kilogam. Khi chọn giống nấm người trồng cần chú ý lựa chọn những bịch giống có màu trắng, mùi thơm và phải đủ ngày. Yếu tố đủ ngày thường là do cơ quan cung cấp giống cung cấp thông tin về giống nấm. Về phần nguyên liệu trồng nấm thường là rơm rạ, người trồng cần phải chuẩn bị nguyên liệu bằng cách lấy rơm rạ cắt khúc nhỏ (khoảng 1cm) sau đó tạo ẩm bằng nước rồi trộn thêm chất phụ gia. Chất phụ gia thường là cám gạo hoặc cám ngô theo tỷ lệ từ 10% – 20%. Sau khi trộn hỗn hợp nguyên liệu xong, người trồng tiến hành đóng nguyên liệu thành từng bịch. Thực tế người dùng thường đóng nguyên liệu vào bịch với trọng lượng 1kg/bịch. Sau đó đem tất cả các bịch nguyên liệu này đem đi hấp thanh trùng. Về yêu cầu đối với các bịch trồng phải đảm bảo bịch thật kín để sau khi hấp thanh trùng độ ẩm của bịch sẽ không giảm nhanh. Cũng vào quá trình hấp thanh trùng này bịch nấm được cố định độ ẩm cơ chất có trong từng bịch và trong quá trình trồng nấm tiếp theo chúng ta sẽ không thể can thiệp vào bịch trồng này nữa mà chỉ cần đảm bảo các yếu tố bên ngoài bịch. Tiếp theo quy trình trồng nấm, sau khi hấp thanh trùng ta đem các bịch nấm này để nguội. Lúc này cơ chất bên trong bịch là hoàn toàn sạch sẽ và cách ly với môi trường bên ngoài. Ta tiến hành cấu giống nấm vào bịch với tỷ lệ nhất định. Trong thực tế tỷ lệ này thường là 5% - 10% đối với từng giống nấm cụ thể. Kết thúc giai đoạn đầu tiên của việc trồng nấm.

Sau khi tiến hành trồng nấm vào các bịch. Ta sẽ để nguyên các bịch đó trên giá trồng nấm của nhà trồng nấm để chuyển đến giai đoạn ủ sợi. Giai đoạn này ta sẽ không phải tác động gì đến bịch nấm vì môi trường trong bịch và môi trường ngoài bịch là cách ly hoàn toàn với nhau. Ta sẽ chỉ tiến hành chăm sóc bằng cách đảm bảo các yếu tố về nhiệt độ môi trường, độ ẩm không khí, cường độ ánh sáng, nồng độ CO2/O2 tác động vào bịch. Giai đoạn này thường mất 1 khoảng thời gian khác nhau với từng giống nấm khác nhau. Thời gian cuối của giai đoạn này ta sẽ thấy bịch nấm ra sợi. Đây chính là dấu hiệu để biết giai đoạn ủ sợi đã kết thúc. Nếu bịch nấm phủ kín sợi thì giai đoạn này được xem như là thành công. Ta sẽ chuyển đến giai đoạn tiếp theo – giai đoạn rạch bịch.

Lúc này khi bịch nấm đã lên đều sợi ta sẽ tiến hành rạch bịch nấm đó và lại tiến hành chăm sóc. Chờ đến khi nấm ra mầm – mầm này mọc ra từ các vết rạch. Các mầm này sau sẽ phát triển thành các cây nấm và chúng ta sẽ tiến hành thu hoạch chúng. Kể từ lúc rạch bịch xong giai đoạn ra quả thể đã được bắt đầu.

Đối với quy trình trồng nấm nói chung. Sau giai đoạn rạch thì chúng ta sẽ có 3 giai đoạn ra quả thể xen kẽ 2 giai đoạn nghỉ. Các giai đoạn ra quả thể là giống nhau với cùng một giống nấm. Tương tự với các giai đoạn nghỉ.

*Hình 2.2 Giai đoạn trồng nấm tổng quát*

Nấm không giống với thực vật thông thường. Khi tiến hành trồng nấm trong suốt cả giai đoạn chỉ có trồng được hoặc không trồng được. Khi thấy xuất hiện những bịch nấm trồng không đạt yêu cầu theo từng giai đoạn tức là bịch nấm đó thất bại. Không có trường hợp chữa bịch nấm đã hỏng. Đối với các bịch nấm bị hỏng cần được đem ra xa nhà trồng nấm, tiến hành tiêu hủy bằng cách đem chôn hoặc đốt để tránh ảnh hưởng đến các bịch nấm còn lại.

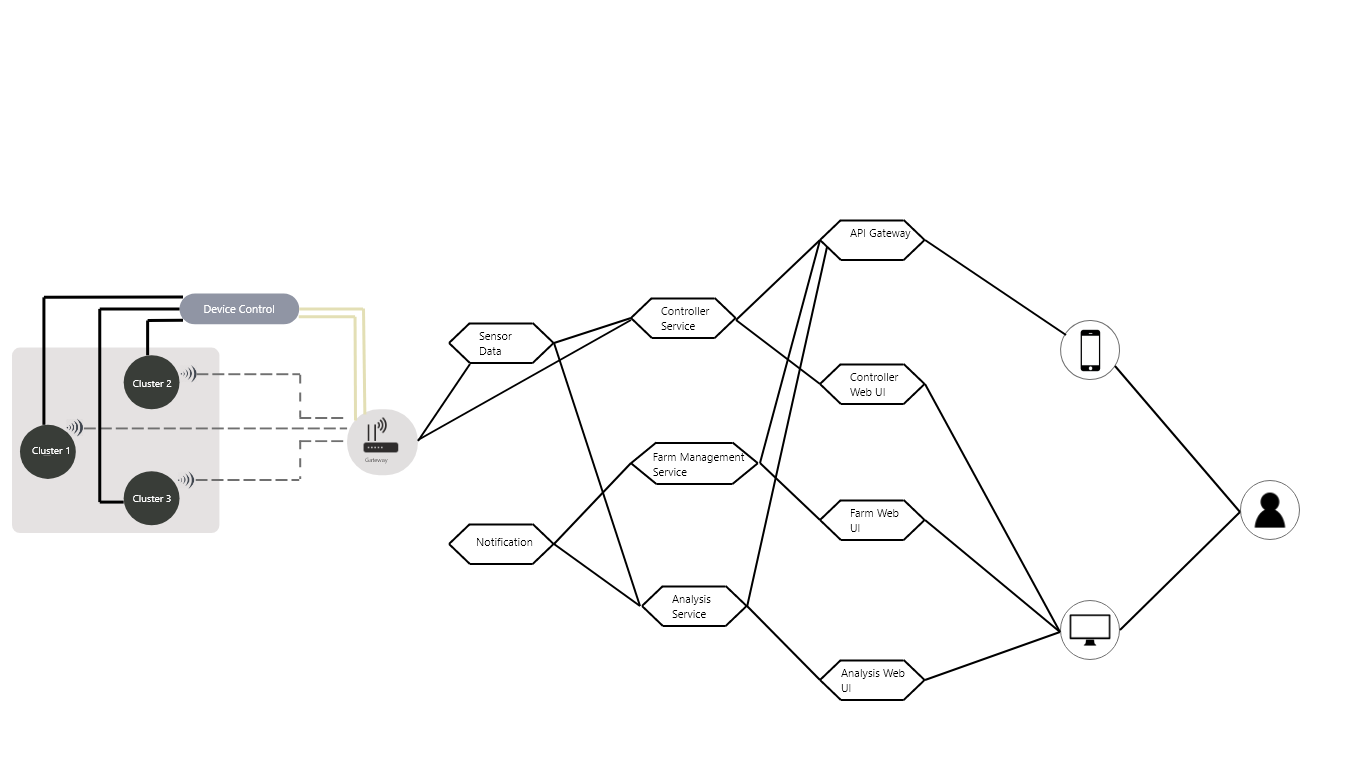
Sau khi tiến hành thu hoạch toàn bộ nấm ở giai đoạn ra quả thể 3. Ta sẽ tiến hành bỏ tất cả các bịch nấm đó đi và tiến hành trồng đợt mới.

Dưới đây là thông tin cơ bản về 4 giống nấm phổ biến thường được trồng trong các nhà nấm ở Việt Nam bao gồm: Nấm sò, nấm rơm, nấm mộc nhĩ, nấm linh chi.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Giống nấm  Thông tin | | Nấm Sò | Nấm Rơm | Nấm Mộc Nhĩ | Nấm Linh Chi |
| Tên khoa học | | *Pleurotus ostreatus* | *Volvariella volvacea* | *Auricularia auricula* | *Ganoderma lucidum* |
| Phân loại | | Giới: Nấm (Fungi).  Ngành: Nấm đảm (Basidiomycota)  Lớp: Nấm tán (Agaricomycetes).  Bộ: Nấm tán (Agaricales).  Họ: Nấm lớn (Pluteaceae)  Chi: Nấm bào ngư ([Pleurotus](http://vi.wikipedia.org/wiki/Pleurotus))  Loài: *Pleurotus ostreatus* | Giới: Nấm (Fungi).  Ngành: Nấm đảm (Basidiomycota).  Lớp: Nấm tán (Agaricomycetes).  Bộ: Nấm tán (Agaricales).  Họ: Nấm lớn (Pluteaceae).  Chi: Nấm rơm (*Volvariella*).  Loài: *Volvariella volvacea.* | Giới: Nấm (Fungi).  Ngành: Nấm đảm (Basidiomycota).  Lớp: Nấm tán (Agaricomycetes).  Bộ: Nấm tán (Agaricales).  Họ: Mộc nhĩ (Auricularaceae).  Chi: Mộc nhĩ (*Auricularia*).  Loài: Nấm mèo (*Auricularia auricula*). | Giới: Nấm (Fungi).  Ngành: Nấm đảm (Basidiomycota).  Lớp: Nấm tán (Agaricomycetes).  Bộ: Nấm đa tầng (Polyporales).  Họ: Nấm lim (Ganodermataceae).  Chi: Nấm Linh chi (*Ganoderma*).  Loài:Nấm linh chi đỏ -*Ganoderma lucidum* |
| Cơ chất trồng | | Mùn cưa/rơm rạ/bông | Rơm | Mùn cưa | Mùn cưa |
| Kiểu bịch | | Bịch | Mô rơm | Bịch/khúc gỗ | Bịch/khúc gỗ |
| Nhiệt độ nuôi trồng | | Nhóm chịu lạnh: 13 - 200C  Nhóm chịu nhiệt: 24 - 280C | 30 – 37 | 25 - 32 | 25 - 30 |
| Độ ẩm (%) | Cơ chất | 60-6 | 65 – 75 | 65 - 70 | 60 - 65 |
| Không khí | 80% | 85% | 85% | 80% |
| Thời gian ủ sợi (ngày) | | 20-25 | 25 - 30 | 30-35 | 50-60 |

*Bảng 2.1 Thông tin một số giống nấm phổ biến tại Việt Nam*

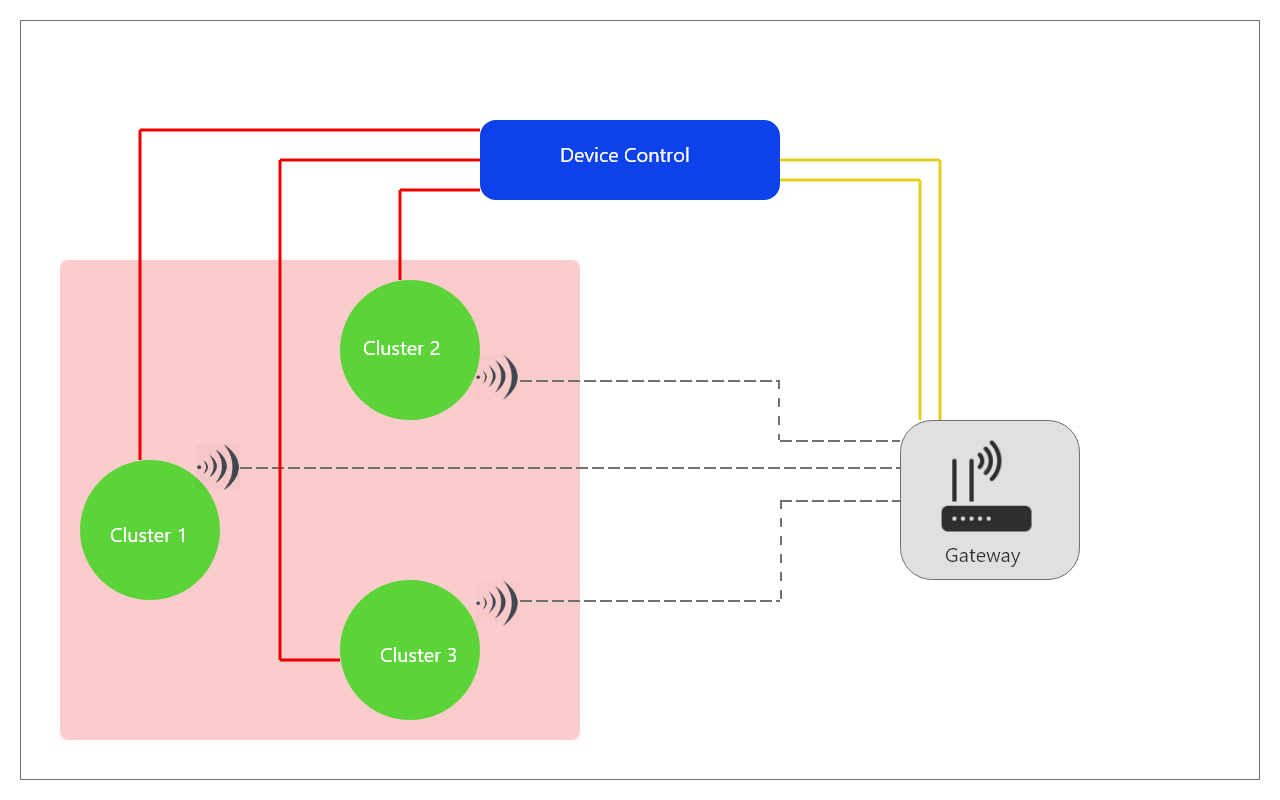
* 1. **Thiết kế hệ thống**
     1. **Sơ đồ tổng quát hệ thống**



*Hình 2.1. Sơ đồ tổng quát của hệ thống*

Ở trên là sơ đồ tổng quát của hệ thống( Hình 2.1) được chia làm hai phần chính đó là sơ đồ phần cứng và sơ đồ cài đặt trên server. Thông tin chi tiết về sơ đồ phần cứng được mô tả ở mục 2.3.2. Hệ thống server được thiết kế theo microservice được trình bày ở khóa luận của bạn Nguyễn Hoàng Biên (K59- CLC).

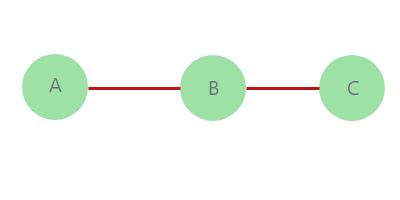
* + 1. **Hệ thống phần cứng**



*Hình 2.2: Sơ đồ hệ thống phần cứng*

Trong mô hình lắp đặt ở phần cứng được chia làm 3 phần chính đó là các Cluster, Device control và Gateway.

Phần đầu tiên là các cluster . Sau khi đã xem mô hình nhà trồng nấm đã triển khai ở thực tế và cơ chế điều khiển của các dự án nông nghiệp thông minh thì nhóm đã quyết định chia nhà nấm ra từng cụm( cluster) ra để quản lý về đo đạc và điều khiển môi trường. Mỗi cụm phải được phân bổ sao cho sự phân bổ về môi trường của các cụm là như nhau nghĩa là trong cùng một cụm thì các thông số về nhiệt độ, độ ẩm không khí… xung quanh cụm phải bằng nhau nên phạm vi mỗi cụm có thể là khác nhau. Lơi ích của việc phân cụm là giúp phần điều khiển và phân bố thiết bị theo dõi dễ dàng hơn so với việc không chia cụm. Giờ đây tại mỗi cụm thì người dùng chỉ cần lắp đặt một sensor cho mỗi loại điều kiện theo dõi vì sự phân bố môi trường trong cụm là như nhau. Vấn đề đặt ra là khi không chia ra các cụm thì việc điều khiển như tưới tiêu sẽ khó mà quản lý đều được. Ví dụ như việc lắp đặt 3 vòi phun như hình vẽ :



*Hình 2.3: Ví dụ chia khu*

Ở ví dụ hình vẽ trên khi độ ẩm ở xung quanh máy bơm A và máy bơm C bị giảm sút còn ở xung quanh máy bơm B vẫn đang bình thường thì server sẽ nhận được dữ liệu và cấp lệnh tưới cho máy bơm A à C khi đó độ ẩm ở khu B rất có thể sẽ tăng vì bị ảnh hưởng ở hai khu còn lại khi khoảng cách ở các khu không được đảm bảo nên việc chia ra các cụm đều nhau là để giải quyết các vấn đề này. Việc điều khiển không ảnh hưởng lẫn nhau giữa các cụm sẽ giúp tưới tiêu được đều hơn. Điểm yếu của việc phân cụm là đó là các nhà nấm phải phân bố nấm theo cụm và việc phải khoanh cụm để quản lý việc này phải được thực nghiệm thì mới tối ưu được việc chia cụm. Mỗi cụm sẽ gồm các bịch nấm, một sensor cho mỗi loại điều kiện môi trường, một thiết bị lấy dữ liệu dữ liệu từ các sensor về truyền lên Gateway và một module phát tín hiệu sóng.

Phần thứ hai là Gateway - thành phần trung gian giữa server và các thiết bị phần cứng, nhiệm vụ của gateway lấy dữ liệu thu thập được ở các cluster qua tín hiện sóng và publish dữ liệu nhận lên MQTT Broker thông qua giao thức MQTT, server backend sẽ subscribe kênh mà gateway publish và nhận dữ liệu và gateway gửi lên. Ngoài ra gateway còn nhận lệnh control các thiết bị điều khiển môi trường từ server cũng thông qua giao thức MQTT.

Phần cuối cùng là Device control. Device control bao gồm các thiết bị điều khiển môi trường một cách tự động hóa như là bóng đèn UV, máy phun sương, điều hòa… Các Device Cotrol đóng vai trò giúp nhà trồng nấm luôn ở điều kiện lý tưởng nhất ở tất cả các khu vực. Mỗi Device Control sẽ điều khiển môi trường một cụm riêng biệt vì vậy hệ thống sẽ đánh mã số của Device Control dựa theo mã số của cụm. Gateway sẽ nhận lệnh điều khiển từ server và từ đó điều chỉnh lại trạng thái của các Device Control.

Như vậy chúng ta đã có một sơ đồ hệ thống phần cứng hoàn chỉnh từ 3 thành phần nêu trên.

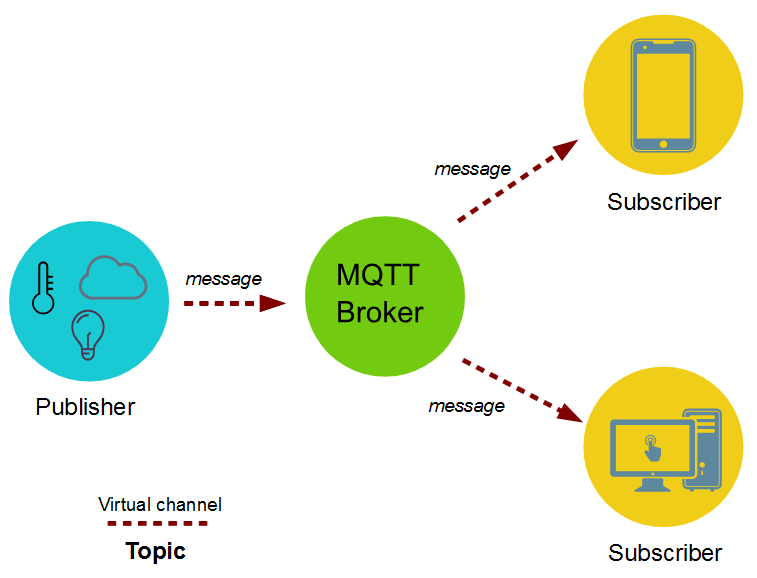
* 1. **Giao thức truyền tải dữ liệu trong IoT**

Dữ liệu từ gateway nhà trồng nấm cần được gửi lên server và cùng với đó dữ liệu điều khiển từ server cũng được gửi đến phía gateway nhà trồng nấm thông qua môi trường internet.Khi mà HTTP đang là giao thức truyền tải dữ liệu qua internet phổ biến hiện nay nhưng nó lại phát sinh một số vấn đề hạn chế trong trong việc truyền tải dữ liệu với các thiết bị IoT, cụ thể đó là thiết bị gateway đặt ở phía nhà trồng nấm. HTTP không thực sự là một giao thức lý tưởng khi mà cần phải đáp ứng những yêu cầu sau của hệ thống:

* Gửi thông tin từ một đến nhiều điểm đích
* Lắng nghe sự kiện khi chúng xảy ra
* Truyền phát một lượng lớn các gói tin nhỏ
* Có độ nhạy cảm cao với:
  + Dung lượng của dữ liệu
  + Điện năng tiêu thụ (đối với thiết bị sử dụng pin)
  + Thời gian phản hồi (khi thông tin gửi cần phải đáp ứng với thời gian thực)
* An ninh và tính riêng tư
* Khả năng mở rộng

Vì những khuyết điểm trên, nhiều giao thức truyền dữ liệu trong IoT đã được phát triển, 2 giao thức phổ biến nhất được áp dụng nhiều đó là MQTT và CoAP

**2.4.1 MQTT**

MQTT (Message Queueing Telemetry Transport) là giao thức được phát triển bởi IBM. MQTT sử dụng TCP, gọn nhẹ và có nhiều tính năng có lợi cho các thiết bị IoT. Giao thức sử dụng kiến trúc publish/subscribe khác với mô hình request/response của HTTP. Điểm giao tiếp trung tâm của mqtt là MQTT broker. Nó chịu trách nhiệm truyền phát các gói tin từ người gửi đến các người nhận. Mỗi client có thể publish, subscribe tới topic nào đó. Topic ở đây là thông tin định tuyến của broker. Khi một client publish một thông điệp trên một topic, broker sẽ gửi thông điệp đó tới tất cả client mà subscribe tới topic đó.

*Hinh 2.5. Truyền tin trong MQTT*

**Tầng ứng dụng QoS (Quality of service)**

Có 3 tùy chọn khi đăng kí kênh và gửi bản tin:

“Delivered at least once”: Gửi ít nhất một lần: Cần ít nhất 1 lần xác nhận từ đầu cuối tức là có thể có nhiều hơn một lần xác nhận đã nhận bản tin.

“Delivered exactly once”: Chỉ gửi một lần: Đảm bảo khi gửi bản tin, phía nhận chỉ nhận được đúng 1 lần, quá trình này cần qua nhiều bước bắt tay

“Fire and forget”: Gửi và quên: Broker/Client sẽ gửi dữ liệu đúng 1 lần, quá trình gửi được xác nhận bởi giao thức TCP/IP.

**LWT (Last Will and Testament)**

Các client MQTT có thể đăng ký một bản tin tùy chỉnh được gửi bởi broker nếu các client ngắt kết nối. Những bản tin này có thể được dùng để báo cho các Clients đã đăng ký Subscriber khi một thiết bị ngắt kết nối.

**Khả năng duy trì bản tin**

MQTT hỗ trợ lưu trữ các bản tin trong Broker để duy trì bản tin. Khi publish các bản tin, các Client có thể yêu cầu broker duy trì các bản tin. Chỉ có các bản tin mới nhất được lưu lại. Khi một Client đăng kí đến một kênh, bất kì bản tin nào đã được đăng kí lưu trữ sẽ được gửi đến Client.

**Tính bảo mật**

MQTT broker có thể yêu cầu tên người dùng và mật khẩu xác thực từ client để kết nối. Để đảm bảo tính bảo mật, kết nối TCP có thể được mã hóa với SSL/TLS (Transport Layer Security (TLS) protocol, Secure Sockets Layer (SSL) protocol).

* 1. **Phương pháp thu thập dữ liệu sensor**

Để một hệ thống nhà nấm hoạt động tốt đẹp và ổn định thì việc thu thập những dữ liệu điều kiện môi trường ảnh hưởng đến chất lượng của cây nấm là điều kiện tiên quyết. Việc cảnh báo người dùng hay điều khiển các device control thì tất cả dựa vào dữ liệu môi trường thực đã được đo đạc được ở nhà nấm.

Việc thu thập và lưu trữ dữ liệu cảm biến sẽ được thực hiện ở hai phase chính. Phase thứ nhất là thu thập dữ liệu từ cảm biến và gửi lên gateway sẽ đẩy dữ liệu lên server để lưu trữ vào cơ sở dữ liệu. Chi tiết về các phase được mô tả sau đây.

Trong phase đầu tiên sau khi đã chia nhà trồng nấm thành các cluster để quản lý thì mỗi cluster sẽ được đánh mã số riêng được quy định bởi người dùng. Vì việc quy định phạm vi cụm ở các nấm khác nhau là khác nhau nên việc đánh mã số cụm được thực hiện thủ công. Việc thu thập và truyền dữ liệu lên gateway sẽ được thao tác riêng biệt ở các cluster và mỗi lần truyền dữ liệu sẽ lấy tất cả các thông số ở các cảm biến trong một cluster gửi lên cho server. Nghĩa là trước khi đẩy dữ liệu sensor lên gateway thì việc đầu tiên phải thu thập được đầy đủ dữ liệu các sensor trong 1 cluster và gửi lên đồng bộ. Thời gian giữa 2 lần gửi dữ liệu sẽ được cài đặt trước, người dùng sau này vẫn có thể sửa đổi tốc độ gửi dữ liệu theo ý muốn của mình. Có hai phương pháp để truyền dữ liệu từ các cảm biến lên gateway:

Phương pháp thứ nhất là sử dụng sensor không dây và truyền dữ liệu qua giao thức sóng không dây đẫ được cài đặt sẵn bên trong sensor. Công việc của gateway là cài đặt giao thức truyền sóng để nhận dữ liệu. Ưu điểm của phương pháp này là việc lắp đặt rất đơn giản về mặt lắp đặt cảm biến vì không cần dây để kết nối đến sensor để lấy dữ liệu sensor. Nhược điểm của phương pháp này là những sensor không dây thường có chi phí khá là đắt đỏ so với sensor thông thường nên những loại sensor này thường dùng trong những hệ thống lớn có quy mô đầu tư rộng. Điểm bất lợi nữa là khó mà phân biệt được dữ liệu các sensor ở các cụm với nhau.

Phương pháp thứ hai và là cũng là phương pháp đang được cài đặt ở hệ thống này là ở mỗi cluster sẽ sử dụng một thiết bị trung gian để thu thập đồng bộ tất cả dữ liệu trong một cụm và gửi dữ liệu lên server qua một giao thức không dây nào đó. Ưu điểm của phương pháp này là không cần nhiều chi phí để mua sensor không dây và việc đồng bộ gửi dữ liệu các sensor khác nhau trong cùng một cluster cũng đơn giản hơn. Với việc đánh mã số thiết bị này theo mã số của cluster nên thông tin các sensor giữa các cụm khác nhau sẽ phân biệt bằng chính ID của thiết bị. Các giao thức truyền dữ liệu giữa thiết bị trung gian và gateway không bị bó buộc vào một loại giao thức mà sensor quy định trước như phương pháp thứ nhất. Nhược điểm của phương pháp này là việc lắp đặt dây sẽ lằng nhằng hơn vì việc phải nối dây giữa sensor và thiết bị trung gian.

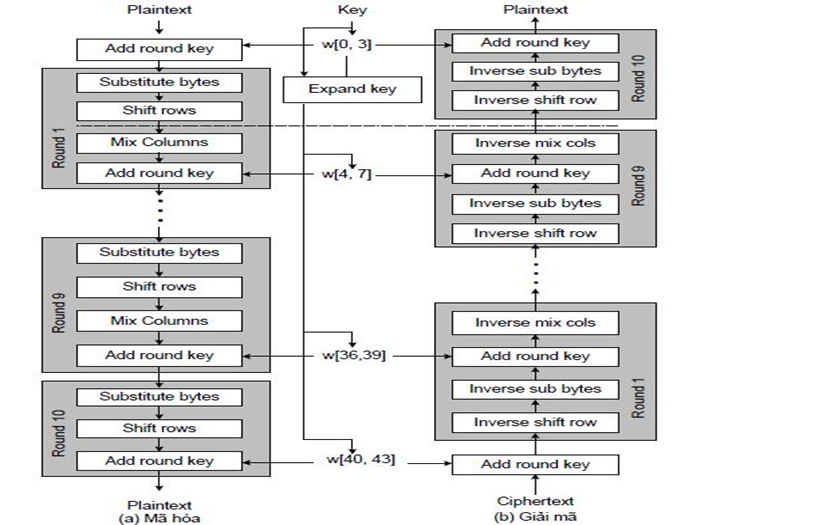
Phase thứ hai là truyền dữ liệu sensor từ gateway lên server. Hiện tại có nhiều giao thức để hỗ trợ cho việc này như MQTT, Websocket, SOAP,HTTP… Hiện tại MQTT đang là giao thức được sử dụng phổ biến hơn cả cho IoT platform vì có nhiều ưu điểm nổi trội của như là tính bảo mật cao và cơ chế publish/subscribe ( Chi tiết về MQTT được mô tả ở Chương 3). Sau khi phase một được thực hiện thành công, gateway lấy dữ liệu sensor thì lập tức sẽ publish dữ liệu lên channel MQTT mà server đã subscribe từ trước. Server sẽ tiến hành phân tích để loại bỏ dữ liệu nhiễu và lưu trữ vào cơ sở dữ liệu. Như vậy việc thu thập cơ sở dữ liệu đã hoàn tất và công việc này được lặp đi lặp lại từ đầu vụ mùa cho đến khi kết thúc một vụ trồng nấm.

* 1. **Phương pháp mã hóa dữ liệu truyền**

Để một nhà nấm hoạt động một cách an toàn thì việc bảo mật dữ liệu truyền là rất quan trọng. Việc cung cấp mã hóa dữ liệu trong quá trình truyền đi là rất quan trọng, hiện nay có rất nhiều thuật toán mã hóa được sử dụng trên thế giới nhưng với bộ xử lý nhỏ của các thiết bị IoT thì việc cân bằng giữa độ an toàn và hiệu năng của hệ thống cần được quan tâm. Với những thuật toán mã hóa an toàn thì có độ phức tạp và cần nhiều bộ nhớ để xử lý thì khó có thể áp dụng ở các thiết bị IoT. Sau khi tham khảo và tìm hiểu hai loại mã hóa là mã hóa đối xứng AES và mã hóa không đổi xứng RSA thì tôi thấy AES thích hợp hơn với các thiết bị IoT

RSA  là một hệ mã hóa không đối xứng được xây dựng bởi Ron Rivest, Adi Shamir và Leonard Adleman ( tên được đặt theo tên của 3 tác giả) và được sử dụng rộng trong thao tác mã hoá và công nghệ chữ ký điện tử. Ở hệ mã hóa này, public key( kháo công khai ) có thể chia sẻ cho tất cả mọi người. Hoạt động của RSA dựa trên 4 bước chính: sinh khóa, chia sẻ key, mã hóa và giải mã. Vấn đề của RSA là việc sinh khóa liên tục thì việc truyền khóa từ nơi nhận đến nơi gửi là vấn đề khá là khó khăn. Để giữ an toàn cho private key( khóa bí mật) thì việc đầu tiên phải giữ cho 2 số nguyên tố sinh khóa được bí mật việc giữa bí mật thì yêu cầu phải dùng 1028 bit để lưu giá trị cho mỗi số nguyên tố, việc này gần như là bất khả thi với các thiết bị IoT có bộ nhớ xử lý nhỏ.

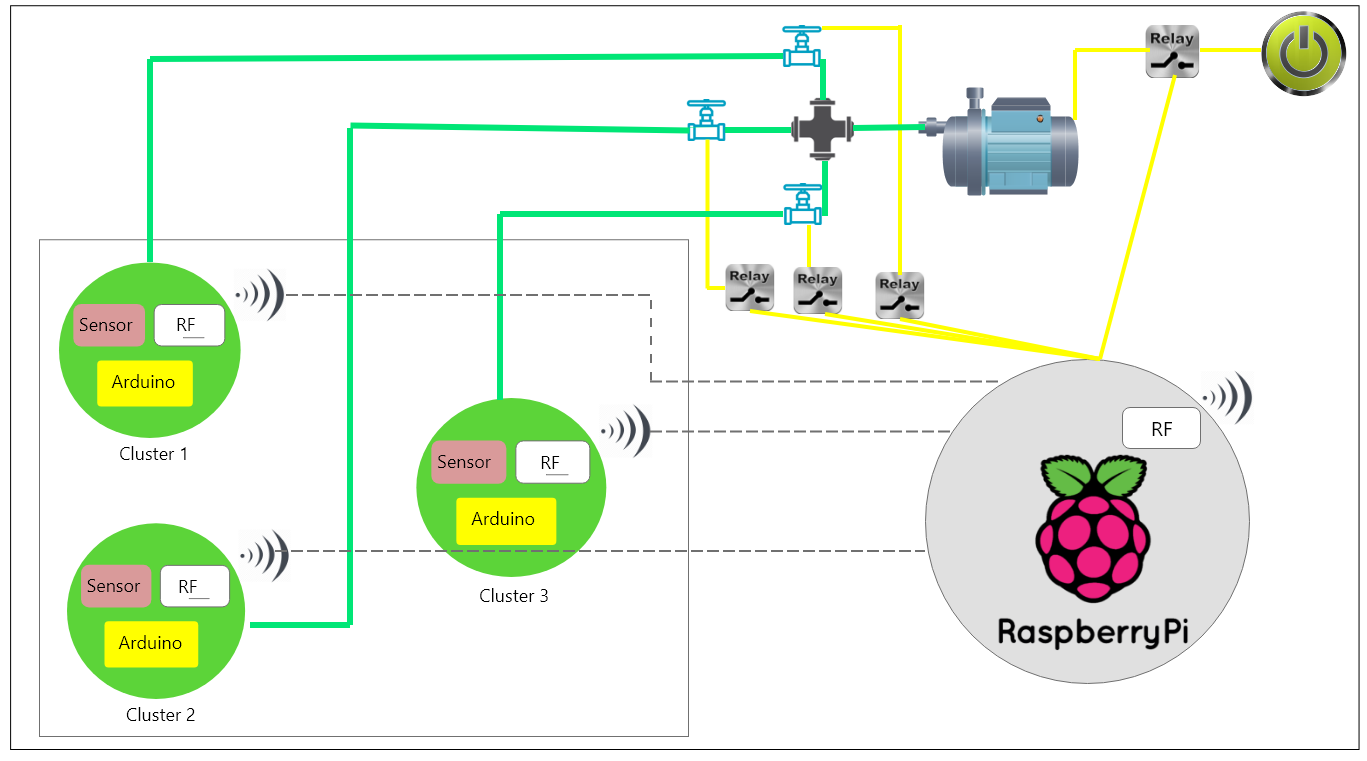
AES ( Advanced Encryption Standard - tiêu chuẩn mã hóa nâng cao) là một thuật toán mã hóa khối được chính phủ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa. Thuật toán được xây dựng dựa trên Rijndael Cipher phát triển bởi 2 nhà mật mã học người Bỉ: Joan Daemen và Vincent Rijmen. AES làm việc với các khối dữ liệu 128bit và độ dài khóa 128bit, 192bit hoặc 256bit. Các khóa mở rộng sử dụng trong chu trình được tạo ra bởi thủ tục sinh khóa Rijndael. Hầu hết các phép toán trong thuật toán AES đều thực hiện trong một trường hữu hạn của các byte. Mỗi khối dữ liệu đầu vào 128bit được chia thành 16byte, có thể xếp thành 4 cột, mỗi cột 4 phần tử hay một ma trận 4x4 của các byte, nó gọi là ma trận trạng thái. Tùy thuộc vào độ dài của khóa khi sử dụng 128bit, 192bit hay 256bit mà thuật toán được thực hiện với số lần lặp khác nhau. Các điểm yếu của AES là có cấu trúc đại số khá đơn giản, mỗi khối luôn được mã háo theo một chiều. Nhưng AES lại có các ưu điểm là với các độ dài khóa từ 128, 192, 256 bit để mã hóa thì nó làm cho AES rất mạnh mẽ trong việc chống xâm nhập. Giao thức đang rất được phổ biến trên thế giới nên tồn tại rất nhiều mã nguồn miễn phí để sử dụng , ngoài ra vì việc mã hóa ở từng khối dữ liệu nhỏ thì với số lượng dữ liệu truyền đi ít trong hệ thống IoT sẽ giải quyết vấn đề về hiệu năng hệ thống.



*Hình 2.6 Sơ đồ tổng quát mã hóa và giải mã của AES*

**CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG THỰC TẾ VÀ ĐÁNH GIÁ**

* 1. **Mô hình triển khai**
     1. **Sơ đồ cài đặt thực tế phần cứng**

****

*Hình 3.1: Sơ đồ cài đặt thực tế phần cứng*

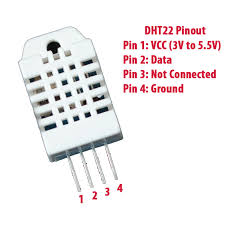
Như giới thiệu ở mục 2.3.2 thì ở hệ thống lắp đặt phần cứng sẽ chia làm 3 phần chính đó là Cluster, Gateway và Control Device

**3.1.1.1 Cluster**

Sau khi thống nhất chia nhà trồng nấm thành các cluster để quản lý và điều khiển thì trong mỗi cluster sẽ được cài đặt

* Một modules nhiệt độ độ ẩm DHT22
* Một board Arduino Uno R3
* Một Module phát sóng nRF24L01

**Sensor nhiệt độ, độ ẩm DHT22**

****

*Hình 3.2 Sensor nhiệt độ, độ ẩm DHT22*

Cảm biến nhiệt độ, độ ẩm DHT22 là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp chuẩn one-wire (giao tiếp qua 1 dây truyền digital) có độ chính xác cao và khoảng đo rộng so với các cảm biến cùng loại là như là DHT11.

Các thông số ký thuật của DHT22:

* Nguồn cấp : 3 -> 5 VDC.
* Dòng sử dụng: 2.5mA tối đa (khi truyền dữ liệu).
* Dòng sử dụng: 2.5mA tối đa (khi truyền dữ liệu).
* Đo tốt ở độ ẩm 0-100%RH với sai số 2-5%.
* Tần số lấy mẫu tối đa là : 0.5Hz (2 giây 1 lần).
* Kích thước 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53").

**Board Arduino Uno R3**

****

*Hình 3.3: Boar Arduino Uno R3*

Arduino là một board mã nguồn mở với rất nhiều thư viện để giao tiếp từ vi điều khiển của nó với các thành phần bên ngoài như đèn LED, motor, LCD, bàn phím, module Bluetooth, GMS và rất nhiều thứ khác muốn giao tiếp với arduino. Arduino cơ bản được tạo bởi vi điều khiển nhưng nó có đủ các ổ cắm ngoài để kết nối với các thiết bị khác và cũng build các project bởi lập trình viên trên máy tính thông qua cổng USB. Có rất nhiều board Arduino được sản xuất để phục vụ lập trình.

Arduino làm một trong những thiết bị điện tử tốt nhất ở thế kỷ 21 đi kèm với mã nguồn mở mà bất kỳ ai cũng có thể sử dụng bởi vì nó thân thiện và dễ dàng để sử dụng. Nó không chỉ thiết kế và phát triển mà nó còn có thể test prototyped và phần cứng vô thời hạn. Ngôn ngữ lập trình thân thiện Arduino là một phiên bản của C++ nên dễ dàng với mọi người để xây dựng và phát triển những ý tưởng của họ.

Mục đích của Arduino là được phát triển để cho mọi người có thể xây dựng, mày mò một ứng dụng có phần cứng hoàn chỉnh mà không cần hiểu biết nhiều về ngôn ngữ lập trình và giải thuật. Ở trong khoảng thời gian mã nguồn mở cho lập trình cho phần cứng đang rất được thịnh hành trên internet thì chi phí để phát triển arduino rất nhỏ so với các vi điều khiển còn lại.

Bộ board Arduino được sử dụng trong dự án là Arduini Uno R3 là vi điều khiển ATmega328P có 14 chân Digital và 6 chân analog.

Các thông số kỹ thuật của board Aruino Uno R3

* Điện áp nguồn: 5V
* Điện áp đầu vào(Recommended): 7-12V
* Điện áp đầu vào(limit): 6-20V
* Số chân I/O Digital: 14 (trong đó có 6 chân PWM đầu ra)
* Số chân PWM Digital: 6
* Số chân Analog: 6
* DC Current per I/O pin: 20mA
* Cường độ cung cấp cho 3.3V : 50mA
* Bộ nhớ Flash: 32Kb( ATmega328P) trong đó sử dụng 0.5Kb cho bootloader
* Bộ nhớ SRAM: 2Kb( ATmega328P)
* EEPROM: 1Kb( ATmega328P)
* Chiều dài: 68.6 mm
* Chiều rộng: 53.4 mm
* Trọng lượng: 25g

**Module nRF24L01(8 chân)**

NRF24L01 là module thu phát sóng đơn tuyến với tần số 2.4-2.5 GHz ISM với độ tích hợp cao . Nguồn ra, kênh tần số ,cài đặt giao thức được lập trình dễ dàng thông qua SPI interface. Phạm vi truyền nhận của sóng trong vòng 1Km thích hợp với điều khiển trong nhà, phạm vi nhỏ. Ưu điểm của nRF24L01 là một trong những IC có giá thành rẻ nhất phù hợp với các đối tượng học sinh, sinh viên hoặc những người làm dự án thử nghiệm. Nguồn điện tiêu thụ để hoạt động rất thấp , chỉ đến 9.0mA ở nguồn ra -6dBm và 12.3mA trong chế độ Rx.Nhược điểm là sóng chỉ truyền trong nội bộ khi lắp thêm ăng-ten thì phạm vi tối đa cũng chỉ đến vài Km và chế độ nhận dữ liệu chỉ xác định được tối đa 6 địa chỉ khác nhau ở bên gửi dữ liệu.

Các tính năng chính:

1. Sóng

* Sử dụng tần số 2.4GHz
* Bao gồm 126 kênh RF để lựa chọn phát sóng.
* Chân Tx và Rx cơ bản
* Tốc độ truyền dữ liệu 1Mbps và 2Mbps

1. Bộ phần truyền dữ liệu

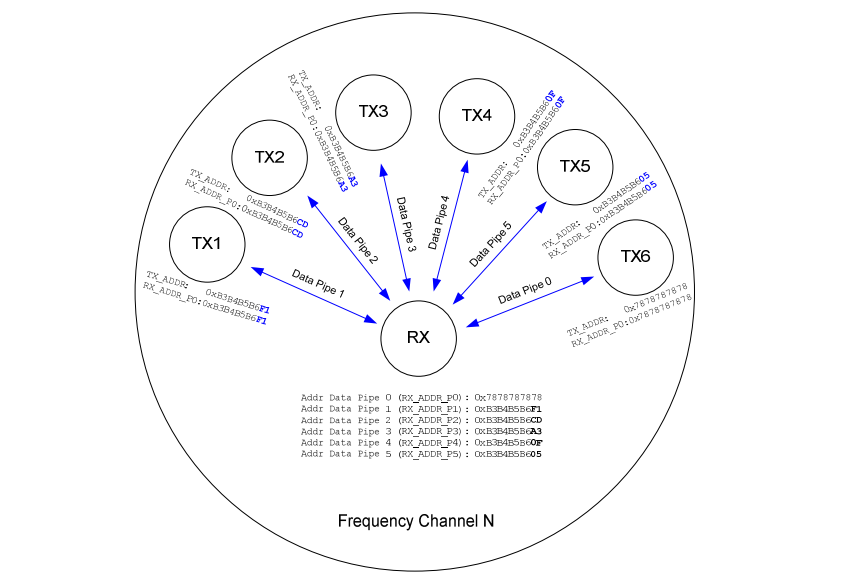
* Có thể lập trình được nguồn ra từ : 0, -6, -12 hoặc -18dBm
* 11.3mA tại 0dBm output

1. Bộ phận nhận dữ liệu

* Tích hợp bộ lọc kênh
* 12.3mA tại 2Mpbs

1. Giao thức Enhanced ShockBurst

* Là một phương pháp xử lý gói tin làm liên kết giữa 2 bên nhận và gửi gói tin dễ dàng và hiệu quả hơn. Trong một hệ thống 2 chiều điển hình, chiều thứ nhất sẽ là Primary Receiver hay còn gọi là PRX chiều còn lại là Primary Transmitter gọi là PTX. Một lần gửi dữ liệu được coi là thành công khi PTX nhận được gói xác nhận( Acknowledgment packet) từ PRX . Khi không nhận được Acknowledgment packer trong một khoảng thời gian nào đó thì coi như việc truyền tin bị hỏng và sẽ tự động truyền lại gói tin.
* Dung lượng tối đa của dữ liệu truyền đi là 32Kb
* Tự động xử lý gói tin
* Tự động xử lý giao dịch gói tin( Tự động báo nhận , tự động truyền lại gói tin bị mất)
* Nhận tối đa 6 địa chỉ luồng đến( data pipes) khác nhau. Mỗi data pipe được đánh địa chỉ khác nhau nhưng có chung 1 kênh tần số. 8 bit được dùng để đánh địa chỉ cho mỗi pipe. NRF24 sẽ sử dụng địa chỉ của pipe để báo nhận gói tin. Có nghĩa là NRF24 sẽ truyền Ack với cùng địa chỉ với địa chỉ tại nơi nó nhận được dữ liệu.



*Hình 3.4: Quy định về data pipes trên một channel*

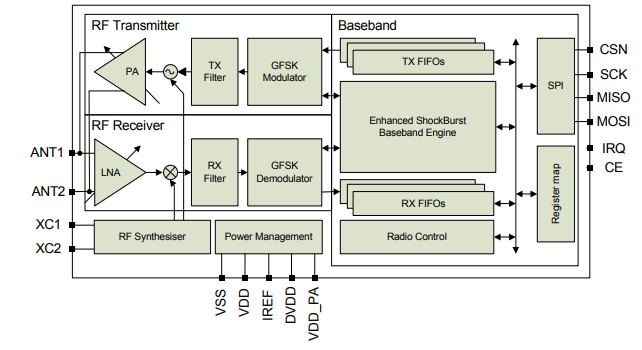
1. Quản lý năng lượng

* Được tích hợp bộ điều chỉnh điện áp
* Giới hạn năng lượng từ 1.9V đến 3.6V

1. SPI interface

* Giao tiếp với vi điều khiển bằng bộ SPI interface kết nối qua 4 pin
* Tốc độ truyền chuẩn của SPI interface 8Mbps.

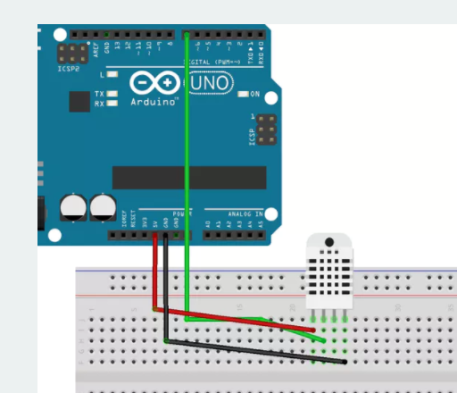
1. Sơ đồ mạch



*Hình 3.5: Sơ đồ mạch của module nRF24l01*

* Chân CE là input quan trọng với RF24. Chân này được sử dụng để kiểm soát truyền và nhận dữ liệu ở các mode TX và RX ( TX mode là mdoe được sử dụng để nhận dữ liệu đến, RX mode là mode được sử dụng để truyền dữ liệu đi).
* 4 chân CSN,MOSI,MISO,SCK là các chân làm việc với SPI Interface.

**Sơ đồ kết nối giữa Arduino Uno và DHT22**

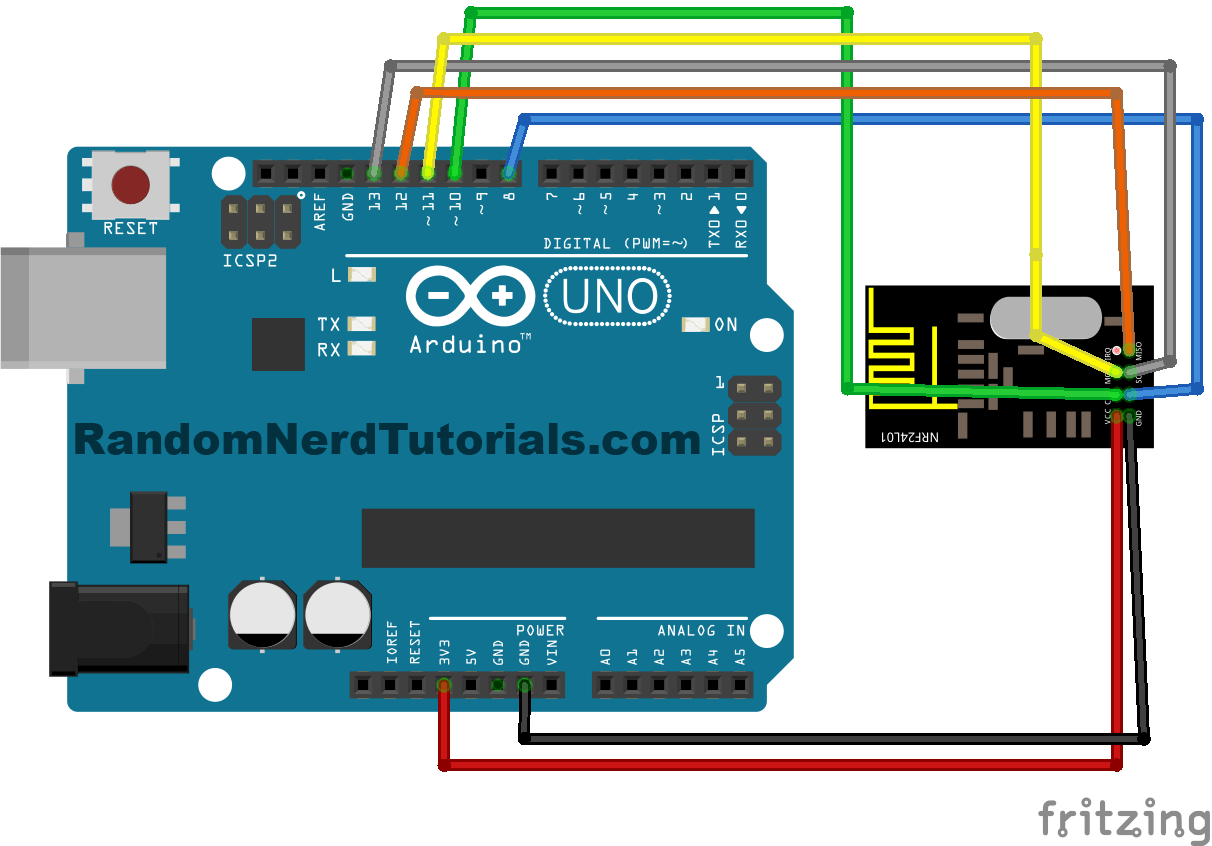


*Hình 3.6: Sơ đồ kết nối giữa Arduino và cảm biến DHT22*

Hướng đẫn nối chân:

* Chân đầu tiên của DHT22 là chân dương được nối với chân 5V của Arduino Uno( dây đỏ)
* Chân thứ 2 từ bên trái sang của DHT 22 là chân đọc dữ liệu được nối với chân Digital thứ 7 của Arduino Uno( dây xanh lá cây)
* Chân thứ 3 từ trái sang của DHT22 không sử dụng.
* Chân thứ 4 từ bên trái là chân âm của DHT22 được nối với chân GND của Arduino Uno( dây màu đen)

**Sơ đồ kết nối giữa Arduino Uno và nRF24L01**

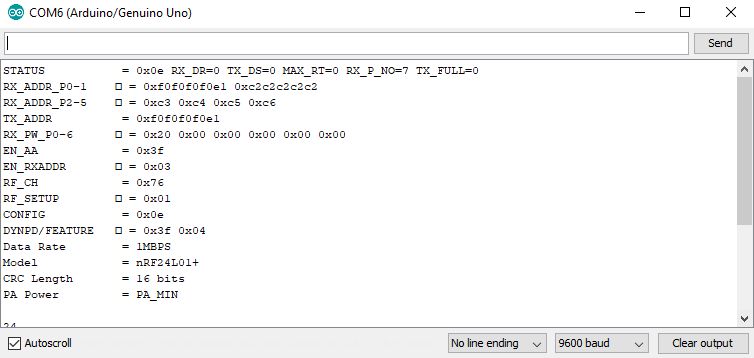


*Hình 3.7: Sơ đồ nối chân giữa Arduino Uno và module nRF24L01*

Hướng dẫn nối chân

* Chân thứ nhất của nRF24lL01( đấu với dây đen) là chân âm được nói với chân GND của Arduino Uno.
* Chân thứ 2 của nRF24L01( đấu với dây màu đỏ) là chân dương được đấu với chân 3.3V của Arduino Uno.
* Chân thứ 3 của nRF24L01( đấu với dây màu xanh dương) là chân CE được nói với chân digital 8 của Arduino Uno.
* Chân thứ 4 của nRF24L01( đấu với dây xanh lá cây) là chân CNS được nối với chân digital 10 của Arduino Uno.
* Chân thứ 5 của nRF24L01( đấu với dây xám) là chân SCK được với chân digital 13 của Arduino Uno
* Chân thứ 6 của nRF24L01( đấu với dây vàng) là chân MOSI được nối với chân digital 11 của Arduino Uno
* Chân thứ 7 của nRF24L01( đấu với dây cam) là chân MISO được nối với chân digital 12 của Arduino Uno.

Sau khi cài cắm dây thành công sẽ tiến hành chạy thử sample có sẵn trong thư viện RF24 ở đường dẫn <https://yadi.sk/d/CXjuENmaEEwun> .Bật Serial Monitor của Arduino IE để kiểm tra cấu hình cài đặt truyền sóng RF gửi dữ liệu đi từ Arduino.



*Hình 3.7: Cấu hình sóng RF sau khi đã kết nối với Arduino*

Ở đây Serial Monitor hiển thị cho chúng ta biết các thông số quan trọng là data pipe của sóng đi đang được đánh là “F0F0F0F0E1” tốc độ truyền dữ liệu ở SPI Interface là 1 MBPS, kênh được truyền là kênh thứ 76 và đang sử dụng nguồn ở cấp độ thấp

Sau khi thực hiện cài cắm thành công thì đã có một Cluster hoàn chỉnh để gửi dữ liệu sensor lên Gateway.

**3.1.1.2 Gateway**

Gateway là bộ phận trung gian giữa phần cứng và server. Nhiệm vụ của gateway là nhận dữ liệu sóng từ arduino rồi gửi dữ liệu sensor lên server, ngoài ra gateway còn có nhiệm vụ là điều khiển các device control từ server gửi về.

Sau khi thống nhất thì Raspberry Pi 3 Model B được sử dụng để cài đặt Gateway, ở Gateway cũng có một module nRf24L01 có nhiệm vụ nhận dữ liệu từ các tín hiệu truyền đi ở Arduino và cuối cùng là Module Relay để điều khiển bật tắt tự động máy bơm.

**Raspberry Pi 3 Model B**



*Hình 3.8 Module Raspberry Pi 3Model B*

Raspberry là máy tính nhỏ gọn, kích cỡ chỉ bằng thẻ tín dụng có giá thành rẻ dao động từ 5$- 35$ được bán phổ biến trên toàn thế giới và có thể hoạt động như một desktop computer hoặc được xây dựng các thiết bị thông minh được phát triển bởi Raspberry Pi Foundation . Raspberry Pi được phát minh bởi Eben Upto vào năm 2006 người đã nhận ra sự thiếu hụt các kỹ năng về khoa học máy tính ở các ứng viên của đại học Cambridge khi ông làm việc và giảng dạy ở đây. Vì vậy Raspberry Pi đã ra đời nhằm thúc đẩy việc giảng dạy về khoa học máy tính ở các trường học và các nước phát triển. Nó mở ra cho mọi người ở tất cả lứa tuổi cơ hội để khám phá máy tính, tìm hiểu và lập trình bằng nhiều ngôn ngữ khác nhau, chẳng hạn như Scratch, Python, PHP… Nó có khả năng làm được mọi thứ như một máy tính để bàn, từ trình duyệt web, xem video độ nét cao, xử lý bảng tính (spreadsheets), xử lý văn bản (word processing) và thậm chí là chơi game.

Raspberry Pi không thay thế hoàn toàn máy tính để bàn hoặc máy xách tay . Bạn không thể chạy hệ điều hành Windows trên đó vì BCM2835 dựa trên kiến trúc ARM nên không hỗ trợ loại mã x86/x64 , nhưng vẫn có thể chạy bằng các hệ điều hành nhân Linux với các tiện ích như lướt web , môi trường giao diện Desktop và các tác vụ khác . Tuy nhiên Raspberry Pi là một thiết bị đa năng rất đáng ngạc nhiên với nhiều phần cứng có giá thành rẻ nhưng rất hoàn hảo cho những hệ thống điện tử , những dự án điện tử ở quy mô nhỏ, thiết lập hệ thống tính toán rẻ tiền cho những bài học trải nghiệm môi trường lập trình.

Vì là một micro computer nên Raspberry Pi cũng có những hạn chế như:

* CPU xử lý chậm nên tốn nhiều thời gian để tải và cài đặt các ứng dụng ngoài và không thể xử lý các tác vụ kiểu multitasking.
* Không tương thích với các hệ điều hành khác như Windows OS…
* Để sử dụng Raspberry Pi chúng ta sẽ phải bỏ số tiền nhiều hơn số tiền để mua Board Pi vì để cài đặt được raspberry Pi thì cần phải mua các phụ kiện như thẻ SD, Sạc nguồn USB, chuột, bàn phím …
* Không phù hợp với các các dự án lớn yêu cầu xử lý nhiều

Các thông số cơ bản của Raspberry Pi 3 Model B:

* Boardcom BCM2873 64bit ARMv8 Quad Core đơn board chạy tại 1.2GHz
* 1GB RAM
* BCM43143
* Bluetooth Low Enery
* 40 chân GPIO
* 4 Cổng USB
* 4 cổng đầu ra Stereo và Composite video
* Full size HDMI
* Cổng CSI Camera để kết nối với Raspberry Pi Camera
* Cổng hiển thị DSI để kết nối với Raspberry touch screen
* Cổng micro SD để load hệ điều hành và lưu trữ dữ liệu.

**Module 4 Relay 5V**



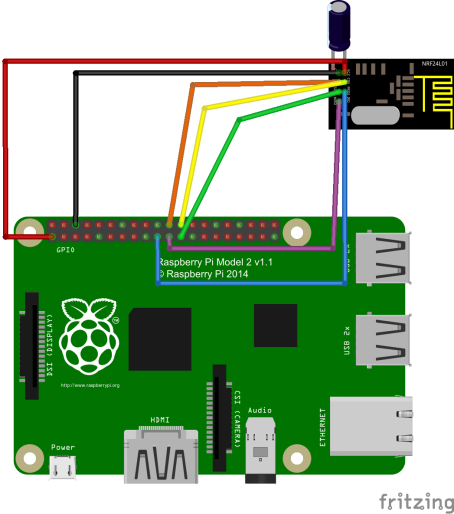
*Hình 3.9: Module Relay 4 kênh 5V*

Module 4 relay 5V phù hợp cho các ứng dụng đóng ngắt điện thế cao AC hoặc DC, các thiết bị được tiêu thụ dòng lớn, module thiết kế nhỏ gọn, có opto và transistor cách ly, kích đóng bằng mức thấp (0V) phù hợp với mọi loại MCU và thiết kế có thể sử dụng nguồn ngoài giúp cho việc sử dụng trở nên linh động và dễ dàng hơn.

Các thông số kỹ thuật

* Sử dụng điện áp nuôi 5VD
* 4 Relay đóng ngắt ở điện thế kích bằng 0V nên có thể sử dụng cho cả tín hiệu 5V hay 3v3 (cần cấp nguồn ngoài), mỗi Relay tiêu thụ dòng khoảng 80mA
* Điện thế đóng ngắt tối đa: AC250V - 10A hoặc DC30V - 10A
* Có đèn báo đóng ngắt trên mỗi Relay.

**Sơ đồ kết nối giữa Raspberry Pi 3 và Module nRF24L01**

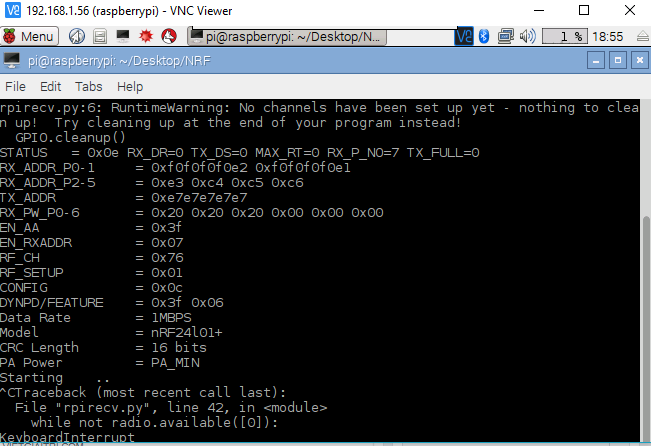
****

*Hình 3.10: Sơ đồ kết nối chân giửa Raspberry pi 3 và nRF24L01*

Sở đồ nối chân

* Chân thứ nhất của nRF24lL01( đấu với dây đen) là chân âm được nói với chân GND của Raspberry Pi 3.
* Chân thứ 2 của nRF24L01( đấu với dây màu đỏ) là chân dương được đấu với chân 3.3V của Raspberry Pi 3.
* Chân thứ 3 của nRF24L01( đấu với dây màu cam) là chân CE được nói với chân 22 Raspberry Pi 3
* Chân thứ 4 của nRF24L01( đấu với dây vàng) là chân CNS được nối với chân 24 của Raspberry Pi 3
* Chân thứ 5 của nRF24L01( đấu với dây xanh lá cây) là chân SCK được với 23 của Raspberry Pi 3
* Chân thứ 6 của nRF24L01( đấu với dây xanh dương) là chân MOSI được nối với chân 19 của Raspberry Pi 3
* Chân thứ 7 của nRF24L01( đấu với dây màu tím) là chân MISO được nối với chân 21 của Raspberry Pi 3.

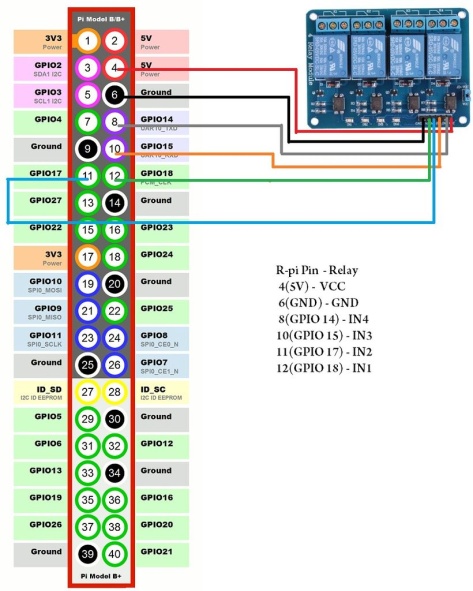
Sau khi lắp đặt xong thì để kiểm tra nối dây tương tự như là giữa Arduino và nRF24L01. Tại đây chạy thử sample từ thư viện <https://github.com/Blavery/lib_nrf24> và hiển thị dữ liệu cài đặt của sóng RF trên terminal ( Lưu ý trước khi chạy thử sample thì phải mở SPI interface ở raspberry pi và cài đặt thư viện spidev tại <https://github.com/Gadgetoid/py-spidev/archive/master.zip> để có thể giao tiếp SPI giữa nRF24L01 và raspberry)



*Hình 3.11: Cấu hình RF sau khi kết nối với Raspberry*

Ở đây Terminal hiển thị cho chúng ta biết các thông số quan trọng là data pipe của sóng nhận ở Rx0 đang được đánh là “F0F0F0F0E2”, tại Rx1 là “F0F0F0E1” , còn từ Rx2 đến Rx5 đang được hiển thị ẩn, tốc độ truyền dữ liệu ở SPI Interface là 1 MBPS, kênh được truyền là kênh thứ 76 và đang sử dụng nguồn ở cấp độ thấp

**Kết nối giữa Raspberry Pi và Module 4 Relay 5V**

******

*Hình 3.12: Sơ đồ nối chân giữa Raspberry pi 3 và Module 4 Relay*

Sơ đồ nối dây chi tiết:

* Chân thứ nhất của 4 Relay( dây màu đen) là chân âm được nối với chân thứ 4 của Raspberry Pi.
* Chân thứ 2 của 4 Relay( dây xanh lá cây) là chân điều khiển relay đầu tiên được nối với chân thứ 12 của Raspberry Pi.
* Chân thứ 3 của 4 Relay( dây màu xanh dương) là chân điều khiển relay thứ hai được nối với chân 11 của Raspberry Pi.
* Chân thứ 4 của 4 Relay( dây màu cam) là chân điều khiển relay thứ 3 được nối với chân thứ 10 của Raspberry Pi.
* Chân thứ 5 của 4 Relay( dây xám) là chân điều khiển relay cuối và được nối với chân 8 của Raspberry Pi.
* Chân cuối của Relay( dây đỏ) là chân dương được nối với nguồn 5V của Raspberry Pi.

Sau khi lắp đăjt xong giữa Raspberry pi – nRF24L01 và Raspberry Pi – 4 Relay thì chúng ta đã có một bộ Gateway hoàn chỉnh. Hiện tại hệ thống đang dùng 1 relay 4 kênh để điều khiển cho 3 cluster thử nghiệm nên sau này tùy thuộc vào số lượng cluster mà số lượng relay sẽ tăng theo.

**3.1.1.3 Device Control**

Các Device Control có nhiệm vụ giữ cho môi trường nhà nấm luôn ở tình trạng ổn định để nấm có thể phát triển trong môi trường tốt nhất. Hiện tại hệ thống đang sử dụng máy phun sương để thực hiện điều khiển độ ẩm không khí trong nhà trồng nấm. Dự định ban đầu cho hệ thống phun sương là lắp một máy bơm tổng cho cả nhà nấm và chia vòi để điều khiển từng cụm một. Để điều khiển theo cơ chế này thì phải có van từ để bật tắt tự động từng cụm một nhưng do van từ hiện tại khá là khó tìm được và chi phí khá là cao nên nhóm đã quyết định sử dụng một máy phun sương DC 12V nhỏ cho mỗi một cluster.

******

*Hình 3.13: Máy phun sương DC12V*

Mỗi máy phun sương sẽ được kết nối đến một Relay và được điều khiển bật tắt thông qua relay. Giờ đây việc thực hiện tự động điều khiển độ ẩm ở mỗi cluster đã hoàn tất.

**3.1.1.4 Quy trình hoạt động hệ thống phần cứng**

**Arduino đọc dữ liệu sensor và gửi dữ liệu lên cho Raspberry Pi**

Mỗi Aruino sẽ được đánh mã số theo Cluster và có nhiệm vụ thu thập dữ liệu từ sensor DHT22 lên cho Raspberry Pi. Vì dung lượng tối đa của một gói dữ liệu qua sóng RF là 32 byte nên dữ liệu truyền lên sẽ có định dạng mảng 32 byte. Dữ sẽ được mã hóa trước khi truyền đi với cơ chế mã hóa AES.

|  |  |
| --- | --- |
| Byte | Mô tả |
| 1 | Giá trị cluster Id |
| 2 | Giá trị farm Id |
| 3 5 | Giá trị nhiệt độ |
| 6 8 | Giá trị độ ẩm |
| 9 11 | Giá trị nồng độ Co2 |
| 12 14 | Giá trị cường độ ánh áng |
| 15 32 | Không sử dụng |

Cluster Id sẽ được sử dụng 1 byte để truyền dữ liệu và là byte đầu tiên của mảng,1 byte để lưu trữ mã số của nhà nấm để tránh việc hai nhà nấm gần nhau lưu trữ giá trị nấm không chính xác, 12 byte tiếp theo dùng để lưu trữ dữ liệu về các giá trị của nhiệt độ, độ ẩm, nồng độ Co2 và cường độ ánh sáng. Từ byte thứ 15 đến 32 không được sử dụng. Lưu ý rằng 12 byte lưu giá trị của 4 giá trị dữ liệu nghĩa là 3 byte cho mỗi giá trị. Giá trị dữ liệu sẽ là số thập phân nên cần chuyển đổi sang dạng byte để truyền đi. Trong 3 byte đó thì giá trị byte đầu tiên sẽ để kiểm tra giá trị của nó có sử dụng hay không, nếu byte đầu có giá trị là 0 thì server sẽ không đọc dữ liệu từ 2 byte tiếp theo, còn nếu byte đầu có giá trị là 2 byte dữ liệu tiếp theo sẽ được đọc và lưu trữ vào cơ sở dữ liệu. Trong 2 byte dữ liệu thì byte thứ nhất sẽ lưu giá trị trước dấu phẩy của số thập phân và byte tiếp theo để lưu giá trị sau dấu thập phân( Lưu ý rằng vì chỉ sử dụng 1 byte cho việc lưu giá trị sau dấu thập phân nên chỉ lấy đến số thứ 2 đằng sau dấu phẩy). sau đây là ví dụ 1 mảng dữ liệu được truyền đi

[02,24, 01, 32, 45, 01, 72, 70, 00, 00,0 0, 01, 12, 02, 00 , 00 ……]

* Cluster Id: 02
* Nhiệt độ: 32.45
* Độ ẩm: 72.70
* Nồng độ Co2: không sử dụng
* Ánh sáng: 12.02

Sau khi đã định dạng được gói tin thì nhiệm vụ truyền tin sẽ do sóng RF làm việc. Muốn gửi được dữ liệu lên gateway thì phải phải cài đặt kênh truyền của nRF24L01 tại arduino sao cho phải cùng một kênh tần số, để ở chế độ Tx Mode và địa chỉ tín hiệu đến(data pipe) phải trùng với địa chỉ của tín hiệu nhận dữ liệu ở gateway. Tại gateway có thể nhận 6 data pipes nên có thể phân biệt được 6 địa chỉ đến khác nhau. Nhiều cluster cũng có thể đánh cùng một địa chỉ đến vì trong trường hợp 2 cluster cùng gửi dữ liệu lên cùng một lúc thì giao thức Enhanced ShockBurst sẽ tự động xử lý sao cho gateway đều có thể nhận được tất cả dữ liệu truyền lên. Nhiều dữ liệu cùng truyền lên cùng lúc sẽ được đưa vào hàng đợt Rx và sẽ được xử lý đồng bộ ( Lưu ý thời gian giữa hai lần gửi từ Arduino lên Raspberry Pi không nên quá nhanh tránh để quá tải kênh nhận và server lưu trữ quá nhiều dữ liệu

Rx Mode được sử dụng cho nRF24L01 vì nó chỉ có nhiêm vụ nhận dữ liệu từ các Cluster truyền lên mà không cần truyền lại. Sau khi nhận được dữ liệu thì Raspberry Pi sẽ đọc dữ liệu mảng và đưa vào định dạng Json và Publish lên MQTT Broker. Raspberry Pi phải đăng ký chanel trước khi nhận dữ liệu từ các Cluter( channel mà raspberry Pi đăng ký trên Broker phải trùng với channel mà server subscribe). Và có một điểm quan trọng nữa là để phân biệt các nhà nấm khác nhau thì mỗi gateway sẽ được đánh mã số theo mã số của nhà nấm. Mã số của nhà nấm sẽ tự động được sinh ra bằng cơ chế tự động sinh ID của cơ sở dữ liệu. Ví dụ định dạng Json được truyền lên có cấu trúc như sau :

{

“cluster\_id”: “02”,

“farm\_id”:  “03”,

“temperature”: 32.33,

“humidity”:  80,16,

“illuminance”:  10.00,

“co2”:  80.00,

“timestamp””:  1524657060000

}

Trong ví dụ này dữ liệu được truyền lên từ nhà nấm có số thứ tự thứ 03( farm\_id). Trong nhà nấm này tồn tại cụm có mã số là 02( cluster\_id) . Dữ liệu môi trường truyền lên bao gồm nhiệt độ( tempetature), độ ẩm( humidity) , cường độ ánh sáng( illuminance), nồng độ Co2( co2) và cuối cùng là thời gian truyền dữ liệu lên ở dạng timestamp. Dữ liệu thời gian truyền lên đang để ở dạng long. Mục đích của trường timestamp trường hợp server mất kết nối và không thể nhận được dữ liệu ngay khi raspberry pi publish lên broker thì vẫn có thể lưu vào time seriel database.

**Điều khiển tự động**

Để một nhà nấm có thể gọi là nhà nấm thông minh thì việc điều khiển tự động môi trường là một thành phần cốt lõi. Sau khi đã thu thập được dữ liệu từ các cluster từ phía nhà nấm, server sẽ tiến hành loại bỏ các dữ liệu nhiễu rồi phân tích dữ liệu.Khi sensor trả về các thông số môi trường không nằm trong khoảng chuẩn mà nằm ngoài phạm vi cho phép thì server sẽ lập tức gửi lệnh điều khiển về cho Gateway để Gateway đổi trạng thái của các Deveice control. Lệnh điều khiển sẽ được dựa vào cơ chế Proportional Integral Derivative ( PID – đã được giới thiệu ở chương trước) dữ liệu điều khiển cũng để ở dạng JSON. Server sẽ publish lệnh điều khiển lên topic {farm\_id}/control/humidity trong đó farm\_id là mã số của nhà nấm cần được điều khiển để điều khiển về độ ẩm nhà trồng nấm và topic {farm\_id}/control/temp để điều khiển nhiệt độ nhà trồng nấm.Theo mong muốn thì nhà trồng nấm sẽ sử dụng điều hòa để có thể điều chỉnh nhiệt độ phòng nên việc điều khiển sẽ không chia theo cụm mà cho toàn bộ nhà nấm và việc sử dụng điều hòa cũng giúp nhiệt độ nhà nấm luôn ổn định, việc điều chỉnh nhiệt độ hầu như chỉ xảy ra ở giai đoạn chuyển giao giai đoạn sinh trưởng của nấm.

Sau khi server publish dữ liệu lên topic {farm\_id}/control/humidity thì nhà nấm nào có cùng farm\_id sẽ nhận được dữ liệu mà topic trả về. Mục đích để số thứ tự của nhà nấm vào đường dẫn topic là để tránh việc gateway phải kiểm tra lệnh điều khiển trả về không phải của nhà mình. Gateway chỉ subscribe topic có chứ mã số nhà nấm chứa nó. Ví dụ về lệnh điều khiển độ ẩm:

{

“relay\_id”: “02”,

“command”: “ON”,

“duration”: 10,

“off\_duration”: 5

}

Phân tích lệnh điều khiển này thì “relay\_id” chính là mã số của cluster mà máy bơm cần bơm, để việc xử lý được đồng bộ thì mã số của cluster và relay cùng đánh giống nhau. Relay sẽ được nối trực tiếp với máy bơm để tự động bật tắt máy bơm. “command” là lệnh điều khiển trạng thái máy bơm. “ON” tương ứng với việc bật máy và “OFF” tương ứng với việc tắt máy bơm.Trường thứ 3 là “duration” là thời gian bật máy bơm, duration sẽ có giá trị bằng 0 khi lệnh điều khiển là “OFF”. Trường cuối cùng là “off\_duration” là thời gian tối thiểu thay đổi trạng thái ở hai lần liên tiếp việc này giúp máy bơm không bị bật tắt quá nhiều lần liên tiếp khi độ ẩm xấp xỉ ở ngưỡng giới hạn.Ví dụ khi độ ẩm dưới 80% thì máy bơm bật và khi quá 80% thì tắt máy bơm và khi đó độ ẩm đặt ở ngưỡng phạm vi 78 - 82 % thì việc bật tắt máy bơm liên tục sẽ khiến cho máy bơm nhanh hư và hệ thống không được tối ưu.

* + 1. **Cấu hình cài đặt phần mềm**
       1. **Cài đặt giao thức giao tiếp giữa nhà trồng nấm và server**

Như đã trình bày ở trên để bổ sung giao thức MQTT thì cần có một thành phần trung gian là MQTT Broker, thành phần này chịu trách nhiệm nhận các gói tin MQTT, lọc chúng, quyết định xem ai là người muốn nhận nó và gửi gói tin đó cho tất cả những client đã subscribe. Một chức năng khác của MQTT broker là xác thực và phân quyền client.Khi có yêu cầu subcribe hay publish tới topic nào đó thì MQTT Broker sẽ dựa vào thông tin tài khoản đang yêu cầu có quyền hay không thì mới cho phép thực hiện.

Hiện tại trên thế giới có nhiều Broker opensource phổ biến, có ưu và nhược điểm như sau:

* [Mosquitto](https://mosquitto.org/): Viết bằng C, nhanh, ổn định, dùng nhiều. Thích hợp cho các hệ thống nhỏ chạy trên local như Raspberry Pi, hỗ trợ TLS/SSL. Khó scale up, ít phương thức xác thực thiết bị.
* [EMQ (EMQTTD)](http://www.emqtt.io/): Viết bằng Erlang, ổn định, thích hợp hệ thống lớn, dễ scale up, dễ cài đặt, hỗ trợ nhiều phương thức xác thực người dùng, có nhiều plugin. Khó cho người mới
* [Vernemq](https://vernemq.com/) tương tự như EMQTTD nhưng ít phương thức xác thực, ít tính năng hơn
* Khác: [RabbitMQ](https://www.rabbitmq.com/), [HiveMQ](http://www.hivemq.com/), [Kafka](https://kafka.apache.org/), [Apache ActiveMQ](https://activemq.apache.org/) ... nhưng phù hợp hơn với các hệ thống sử dụng microservice

Ở đây tôi chọn EMQ làm MQTT broker cho hệ thống theo dõi và điều khiển nhà trồng nấm bởi nó có nhiều ưu điểm hơn so với những cái khác.Chi tiết về EMQ được giới thiệu ở khóa luận của bạn Nguyễn Hoàng Biên( K59 –CLC).

* 1. **Đánh giá kết quả**

Sau khi lắp đặt và chạy thực nghiệm thì hệ thống đã chạy ổn định và tốn ít hiệu năng. Dữ liệu từ sensor từ các cụm nắm được đẩy lên gateway liên tục và có độ trễ không đáng kể. Việc điều khiển máy bơm đã hoàn toàn được tự động hóa dựa vào dữ liệu môi trường. Tuy nhiên hệ thống vẫn còn các nhược điểm như sau

* Dữ liệu từ các sensor đo được vẫn đang sai số so với thực tế
* Chưa lắp đặt được ở nhà nấm thật.
* Chưa giải quyết các vấn đề khi mạng internet không ổng định

Sau đây là dữ liệu nhà nấm được hiển thị trên visualize



Hình 3.14 Dữ liệu sensor

**CHƯƠNG 4: TỔNG KẾT**

**4.1 Đánh giá**

Hệ thống đã hoàn thiện các chức năng đo đạc thông số môi trường nhà nấm và điều khiển các device control.Việc điều khiển đã tách biệt được các cụm khác nhau trong một nhà nấm. Gửi dữ liệu từ các cluster lên gateway đã được mã hóa an toàn tuy nhiên dữ liệu truyền từ gateway lên MQTT Broker chưa được mã hóa mà mới chỉ sử dụng cơ chế xác thực người dùng.

**4.2 Hướng phát triển**

Mã hóa dữ liệu từ gateway lên MQTT broker và lắp đặt các thiết bị sensor chất lượng tốt hơn để việc đo đạc thông số môi trường được chính xác hơn. Lắp đặt hệ thống ở nhà nấm thực để có thể có thông tin thực tế

***Tài liệu tham khảo***

- nRF24L01 Single Chip 2.4GHz Transceiver Product Specification – Semiconducter

- Step-by-step procedure to connect the NRF24L01+ to the GPIO pins and use the Raspberry as a Serial Gateway (MySensors 1.x)

- Temperature Monitoring With DHT22 & Arduino - <https://create.arduino.cc/projecthub/attari/temperature-monitoring-with-dht22-arduino-15b013>

- Tìm hiểu thuật toán mã hóa khóa đối xứng AES - https://viblo.asia/p/tim-hieu-thuat-toan-ma-hoa-khoa-doi-xung-aes-gAm5yxOqldb