**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN ĐIỆN TỬ**

---------------o0o---------------

****

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**THIẾT KẾ VÀ XÂY DỰNG NỀN TẢNG IOT   
ĐA GIAO THỨC**

**GVHD: Bùi Quốc Bảo**

**SVTH: Nguyễn Thanh Tâm**

**MSSV: 1613058**

**TP. HỒ CHÍ MINH, THÁNG 7 NĂM 2020**

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC BÁCH KHOA Độc lập – Tự do – Hạnh phúc.

-----✩----- -----✩-----

Số: \_\_\_\_\_\_ /BKĐT

Khoa: **Điện – Điện tử**

Bộ Môn: **Điện Tử**

N**HIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

1. HỌ VÀ TÊN : Nguyễn Thanh Tâm MSSV: 1613058

1. NGÀNH: **ĐIỆN TỬ - VIỄN THÔNG** LỚP : DD16VT03
2. Đề tài: Hệ
3. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):

* Tìm hiểu các giao thức truyền tai thông dụng trong các giải pháp IoT
* Thiết kế phần cứng và thi công phần cứng IoT Gateway.
* Thiết kế phần cứng và thi công các thiết bị đầu cuối.
* Lập trình kết nối IoT Gateway và các thiết bị đầu cuối.
* Lập trình kết nối IoT Gateway vào IoT Platform trên Server.

1. Ngày giao nhiệm vụ luận văn: ...............................
2. Ngày hoàn thành nhiệm vụ: ...................................
3. Họ và tên người hướng dẫn: Phần hướng dẫn

Thầy Bùi Quốc Bảo .....................................

................................................................. .....................................

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

*Tp.HCM, ngày…... tháng….. năm 20*

**CHỦ NHIỆM BỘ MÔN NGƯỜI HƯỚNG DẪN CHÍNH**

**PHẦN DÀNH CHO KHOA, BỘ MÔN:**

Người duyệt (chấm sơ bộ):.......................

Đơn vị:......................................................

Ngày bảo vệ : ...........................................

Điểm tổng kết: .........................................

Nơi lưu trữ luận văn: ...............................

***LỜI CẢM ƠN***

“Đi qua những năm tháng Bách Khoa, ta mới biết tuổi trẻ đáng trân trọng như thế nào. Trân trọng, không hẳn là vì có những lúc khó khăn tưởng chừng như gục ngã, không hẳn là vì ta biết mình trưởng thành đến đâu mà đơn giản là vì ta đã làm tất cả những điều đó cùng ai”

Cảm ơn cha mẹ

Cảm ơn thầy hướng dẫn….

Cảm ơn các bạn trong phòng thí nghiệm 209B3 đã tận tình giúp đỡ mình….

Cảm ơn CLB….

Cảm ơn Đảng và nhà nước….

Cảm ơn Bách Khoa, suốt khoảng thời gian 4 năm đã làm ta bao lần vấp ngã, làm bao….

*Tp. Hồ Chí Minh, ngày tháng năm .*

**Sinh viên**

**TÓM TẮT LUẬN VĂN**

Luận văn này trình bày về việc xây dựng hệ thống IoT để thu thập dữ liệu và điều khiển thiết bị thông qua các giao thức: Z-wave, LoraWan, Modbus Serial. Luận văn gồm ba phần chính:

Hardware Gateway và các thiết bị đầu cuối:

* Thiết kế và thi công các thiết bị đầu cuối có hỗ trợ các giao thức kết nối ở trên, bao gồm: Van nước, Công tắc điện không dây, Node thu thập dữ liệu cảm biến, //todo: add more device and information. MCU
* Thiết kế và thi công các gateway cho từng giao thức.
* Tìm hiểu, thiết kế, kiểm thử các thông số công suất, độ lợi, nhiễu trên các loại anten.

Firmware Gateway và các thiết bị đầu cuối

* Lập trình các interface cho các giao thức dựa trên nền tảng FreeRTOS cho các thiết bị chạy vi điều khiển.
* Tìm hiểu, triển khai trình điều khiển cho các thiết bị chạy trên nền tảng máy tính tính nhúng.
* Tìm hiểu, lập trình các giao thức (TCP, MQTT,..) và các chuẩn đóng gói dữ liệu (json,…) để kết nối đến IoT Platform (Thongsboard).

Software trên máy tính

* Lập trình giao diện Windows Form trên máy tính để cài đặt và kiểm tra trạng thái của gateway và các thiết bị đầu cuối.
* Triển khai LoraWan Network Server ở mạng cục bộ

**MỤC LỤC**

[1. GIỚI THIỆU 1](#_Toc44451632)

[1.1 Tổng quan 1](#_Toc44451633)

[1.2 Tình hình nghiên cứu trong và ngoài nước 2](#_Toc44451634)

[1.3 Nhiệm vụ luận văn 3](#_Toc44451635)

[2. LÝ THUYẾT 4](#_Toc44451636)

[2.1. Tìm hiểu về các kết nối trong hệ thống IoT thực tế 4](#_Toc44451637)

[3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 4](#_Toc44451638)

[4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM (NẾU CÓ) 5](#_Toc44451639)

[5. KẾT QUẢ THỰC HIỆN 5](#_Toc44451640)

[6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc44451641)

[6.1 Kết luận 6](#_Toc44451642)

[6.2 Hướng phát triển 6](#_Toc44451643)

[7. TÀI LIỆU THAM KHẢO 6](#_Toc44451644)

[8. PHỤ LỤC 7](#_Toc44451645)

DANH SÁCH HÌNH MINH HỌA

[Hình 5‑1 Kết quả thi công 3](#_Toc310380287)

[Hình 5‑2 Kết quả mô phỏng 3](#_Toc310380288)

**DANH SÁCH BẢNG SỐ LIỆU**

[Bảng 1 Thông số hệ thống 3](#_Toc310380293)

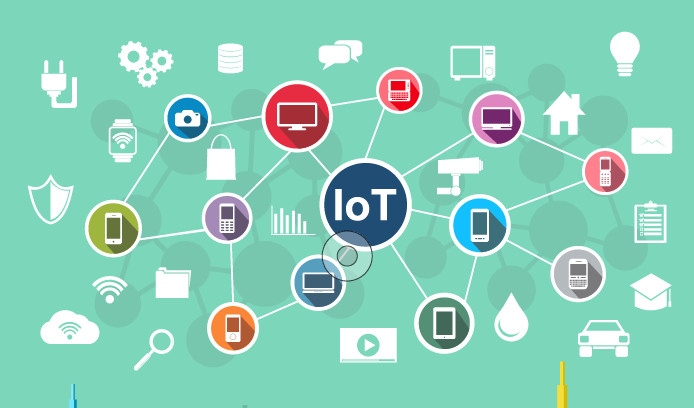
# GIỚI THIỆU

## Tổng quan

|  |
| --- |
|  |

IoT- Internet Vạn Vật đã và đang là động lực phát triển của mọi lĩnh vực công nghệ. Trong năm 2018: theo Stantista, thị trường nhà thông minh toàn cầu đạt gần 60 tỷ USD; theo Gartner, chi tiêu cho an ninh IoT đạt 547 triệu USD; theo IDC, chi tiêu trên toàn thế giới về IoT đạt 772,5 tỷ USD. McKinsey ước tính có 127 thiết bị mới kết nối với Internet mỗi ngày, IoT có tác động kinh tế hàng năm từ 3.9 nghìn tỷ USD đến 11.1 nghìn tỷ USD vào năm 2025. Trong công nghiệp, Accenture ước tính Internet Công nghiệp có thể mang đến 14.2 nghìn tỷ USD cho nền kinh tế toàn cầu vào năm 2030. Trong vận tải, Business Insider ước tính rằng các nhà sản xuất toàn cầu sẽ đầu tư 70 tỷ USD vào năm 2020. Từ đó, ta có thể thấy rõ tầm ảnh hưởng lớn của IoT, cùng với những lợi ích và sự phát triển không ngừng nghỉ của nó.

Để đạt được tối ưu giá trị từ IoT, việc làm chủ một nền tảng IoT (IoT platform) là cần thiết. Nền tảng IoT là tập hợp những ứng dụng thực hiện việc liên kết giữa các thiết bị IoT và trung tâm dữ liệu từ đó tạo ra một hệ thống cung cấp đầy đủ thông tin được thu thập từ các thiết bị IoT (cảm biến, thiết điện tử tiêu dùng, thiết bị di động…) khác nhau để tạo và quản lý ứng dụng; lưu trữ và phân tích dữ liệu cũng như điều khiển từ xa các thiết bị IoT



## Tình hình nghiên cứu trong và ngoài nước

Xem xét nghiên cứu nhiều nguồn tài liệu

Nhận xét cách làm và kết quả của các nghiên cứu đã có, những tài liệu luận văn mà sinh viên đã tham khảo.

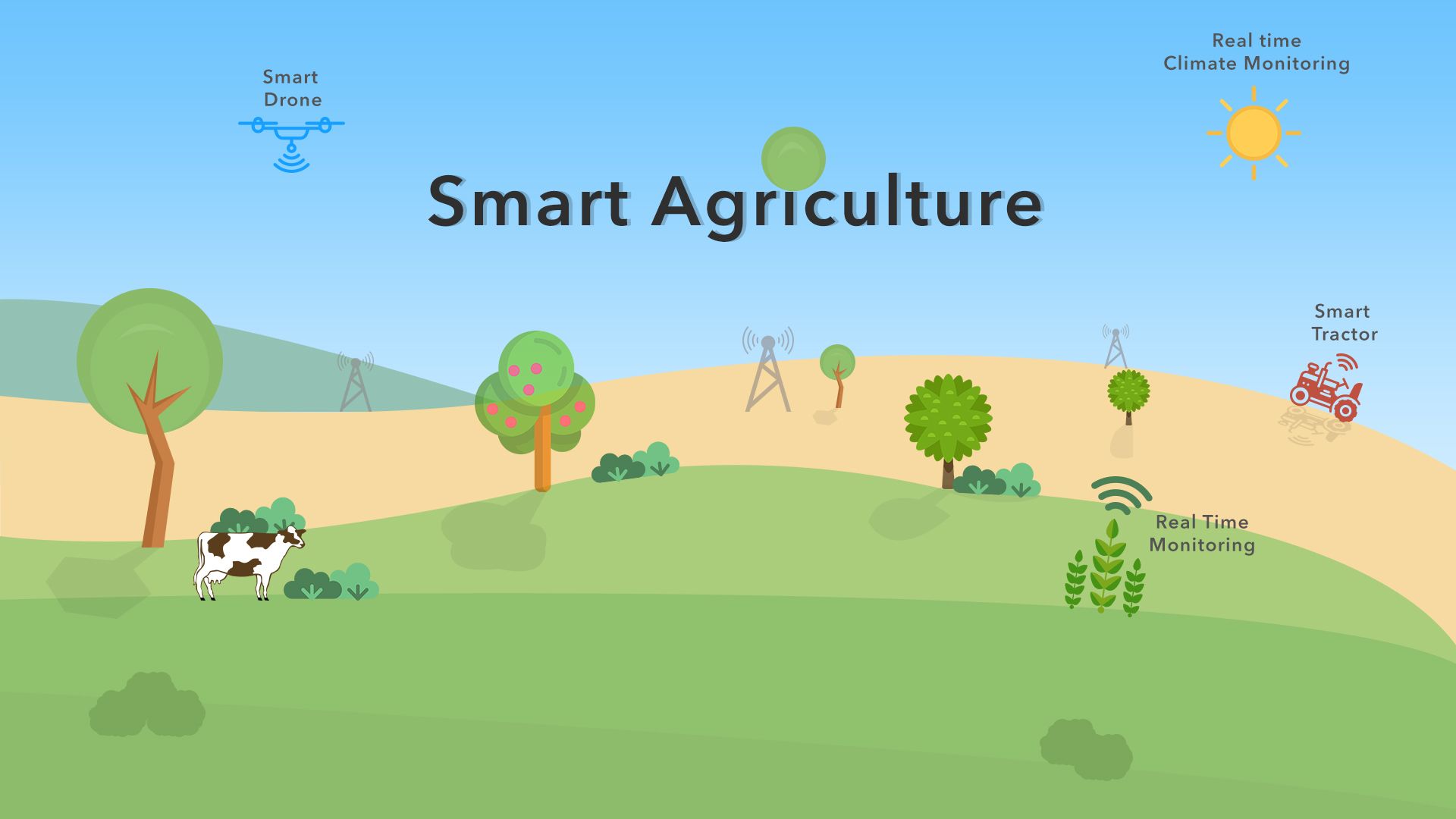
IoT hiện đang là động lực phát triển của hầu hết các lĩnh vực công nghệ. Các giải pháp IoT cho phép các doanh nghiệp cải thiến các hệ thống hiện có giúp nâng cao kết nối với khách hàng và đối tác. Một số ứng dụng IoT dẫn đầu xu hướng công nghệ hiện nay có thể kể đến như:

* Công nghiệp thông minh (Industrial IoT): Một số máy móc công nghiệp có thể hư hỏng hoặc quá hạn sử dụng, cần sự can thiệp từ các kĩ sư. Việc tích hợp các cảm biến và các kết nối không dây cho phép các máy móc có thể tự động phản hồi thông tin kịp thời về cho người quản lý.



Hình 1.2.1-1: Minh họa công nghiệp thông minh

* Nông nghiệp thông minh (Farming IoT): Với việc tích hợp các hệ thống cảm biến trên các cánh đồng, nông dân có thể nhận được các dữ liệu cần thiết để dự đoán về mùa màng, hoặc các hệ thống tự quản lý việc tưới tiêu, cung cấp ánh sáng.



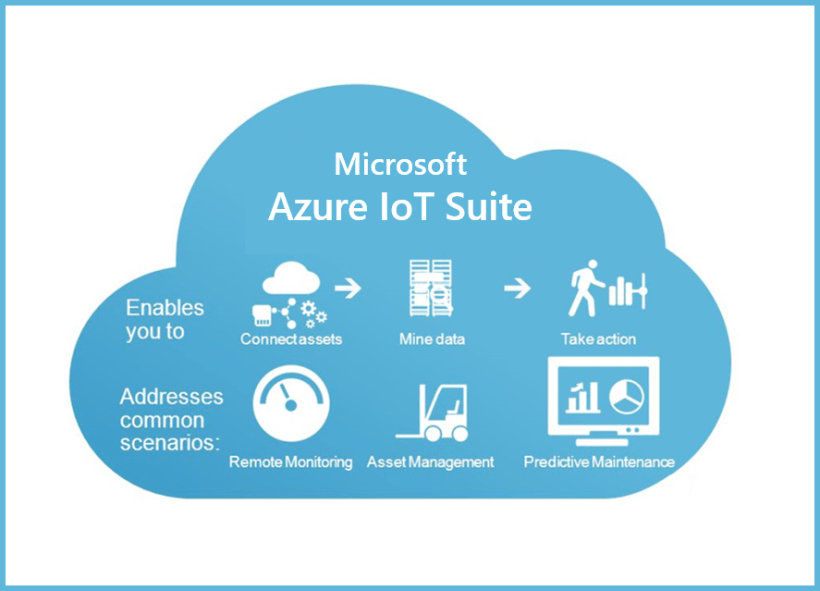
Hình 1.2.1-2: Minh họa nông nghiệp thông minh

* Vận chuyển thông minh (Logistics and transportation IoT): một trong những ứng dụng phổ biến là gán nhãn cho các xe container bằng cách sử dụng RFID. Điều này cho phép các công ty theo dõi quá trình vận chuyển hàng hóa, chọn ra đường đi tốt nhất.



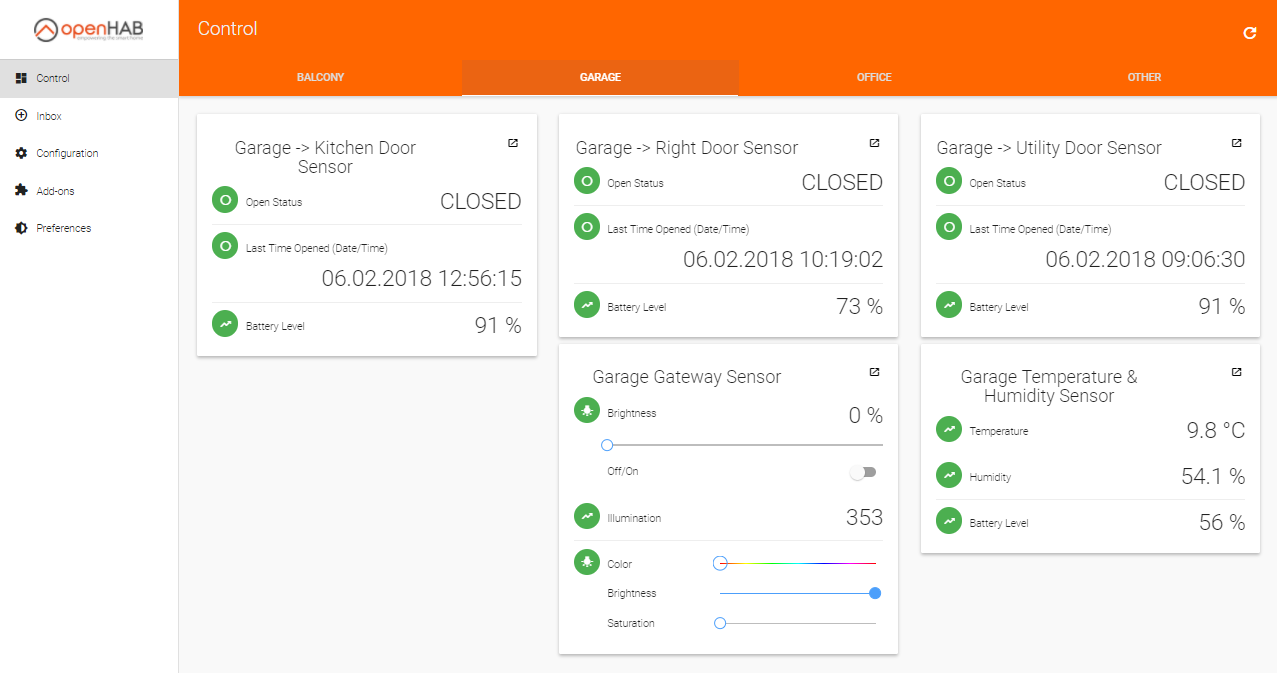
Hình 1.2.1-3: Minh họa vận chuyển thông minh

Các công ty dẫn đầu công nghiệp IoT trên thế giới hiện nay có thể kể đến như Google, Microsoft, Amazon, IBM, Oracle,... Trong đó, Google, Amazon và Microsoft chủ yếu cung cấp các nền tảng Cloud hỗ trợ lưu trữ dữ liệu, IBM và Oracle thiên về các thiết bị Gateway và phần cứng hoặc ứng dụng công nghệ AI, Block Chain...



Hình 1.2.2-1: Microsoft Azure IoT

Một số nền tảng IoT mã nguồn mở như Kaa, ThingsBoard, OpenHAB thì không phải một nền tảng IoT hoàn chỉnh, mà chỉ tập trung vào giải pháp phần mềm hỗ trợ cho việc kết nối các thiết bị phần cứng.



Hình 1.2.2-2: Giao diện OpenHAB dashboard

Tại Việt Nam, nhu cầu về những giải pháp IoT đang rất lớn, để hỗ trợ cho các hệ thống IoT công-nông nghiệp như Nông nghiệp thông minh, Vận tải thông minh và Thành phố thông minh; tuy nhiên, những thương hiệu IoT lớn ở Việt Nam như Lumi, SmartHome by BKAV,… chỉ mới tập trung vào thiết kế và lắp đặt các thiết bị Nhà ở thông minh với những gói dịch vụ và sản phẩm đầu cuối nhất định, gây bất tiện cho người dùng trong việc nâng cấp và thiếu đa dạng trong việc chọn lựa. Bên cạnh đó, một vài cái tên trẻ như Inut, IoTmaker,.. đã xây dựng và phát triển nền tảng IoT của riêng họ với các giải pháp IoT cùng phần cứng. Tuy nhiên, các nền tảng này chỉ mới tập trung vào một giao thức truyền là Wifi với đặc trưng là công suất tiêu thụ lớn, nên khả năng ứng dụng vào các ứng dụng đòi hỏi năng lượng thấp, đặc biệt là nông nghiệp thông minh là không cao.



Hình 1.2.2-3: Smarthome by BKAV

Như vậy, việc tự xây dựng một nền tảng IoT hoàn chỉnh, hỗ trợ từ phần cứng đến phần mềm là mục tiêu cần thiết trong việc phát triển công nghệ IoT hiện nay.

## Nhiệm vụ luận văn

Mô tả các nhiệm vụ của đề tài bao gồm yêu cầu, kết quả cần đạt và giới hạn đề tài. Trong từng nội dung sinh viên cũng cần trình bày thêm cách tiếp cận cũng như ý tưởng thực hiện.

(Ví dụ)

Nội dung 1: Tìm hiểu lý thuyết về …

Nội dung 2: Tìm hiểu về Kit …

Nội dung 3: Thiết kế bộ điều khiển … sử dụng giải thuật …

Các nhiệm vụ chính của luận văn:

* Thiết kế và thi công phần cứng của IoT Gateway và các thiết bị đầu cuối.
* Lập trình cho IoT Gateway giao tiếp và xử lý đa giao thức các dữ liệu từ thiết bị đầu cuối.

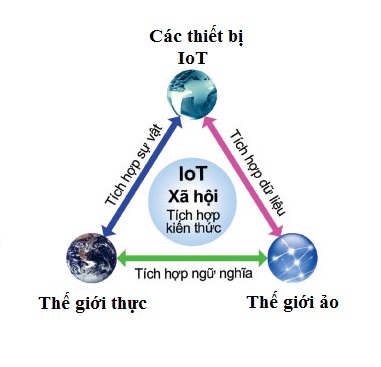
Từ các nhiệm vụ chính đó, cần phải thực hiện những nội dung:

* Nội dung 1: Tìm hiểu lý thuyết về các chuẩn giao tiếp, cách hoạt động, nghi thức quản lý, các thông số, ưu nhược điểm khi triển khai trên các hệ thống IoT cụ thể
* Nội dung 2: Đề xuất, lựa chọn các giải pháp trong từng mô hình cụ thể.
* Nội dung 3: Phân tích, lựa chọn các thiết bị, linh kiện, cảm biến phù hợp cho hệ thống.
* Nội dung 4: Tìm hiểu, tham khảo các thiết kế có sẵn. Từ đó đưa ra sơ đồ chi tiết cho hệ thống.
* Nội dung 5: Tìm hiểu các quy tắc placement, layout mạch cho những thành phần đã đặt ra trên board. Sau đó thực hiện thi công mạch.
* Nội dung 6: Tìm hiểu về hệ điều hành FreeRTOS và cách triển khai hệ thống multi-tasking trên vi điều khiển. Từ đó phát triển firmware cho các thiết bị trên nền tảng os này.
* Nội dung 7: Tìm hiểu cách giao tiếp, điều khiển và trao đổi dữ liệu giữa các modules (LoRa, Zware, Modbus, Ethernet, Wifi,…) từ đó viết trình điều khiển cho từng module trên vi điều khiển và máy tính nhúng.
* Nội dung 8: Thống nhất và chuẩn hoá các loại dữ liệu thu được từ các gateway để giao tiếp với IoT Platform trên server.
* Nội dung 9: Kiểm thử, đo đạc các thông số của hệ thống. Sửa lỗi phần cứng và phần mềm.

# LÝ THUYẾT

## 2.1. Tìm hiểu tổng quan về IoT và các kết nối trong một hệ thống IoT thực tế

### 2.1.1. Tổng quan về IoT

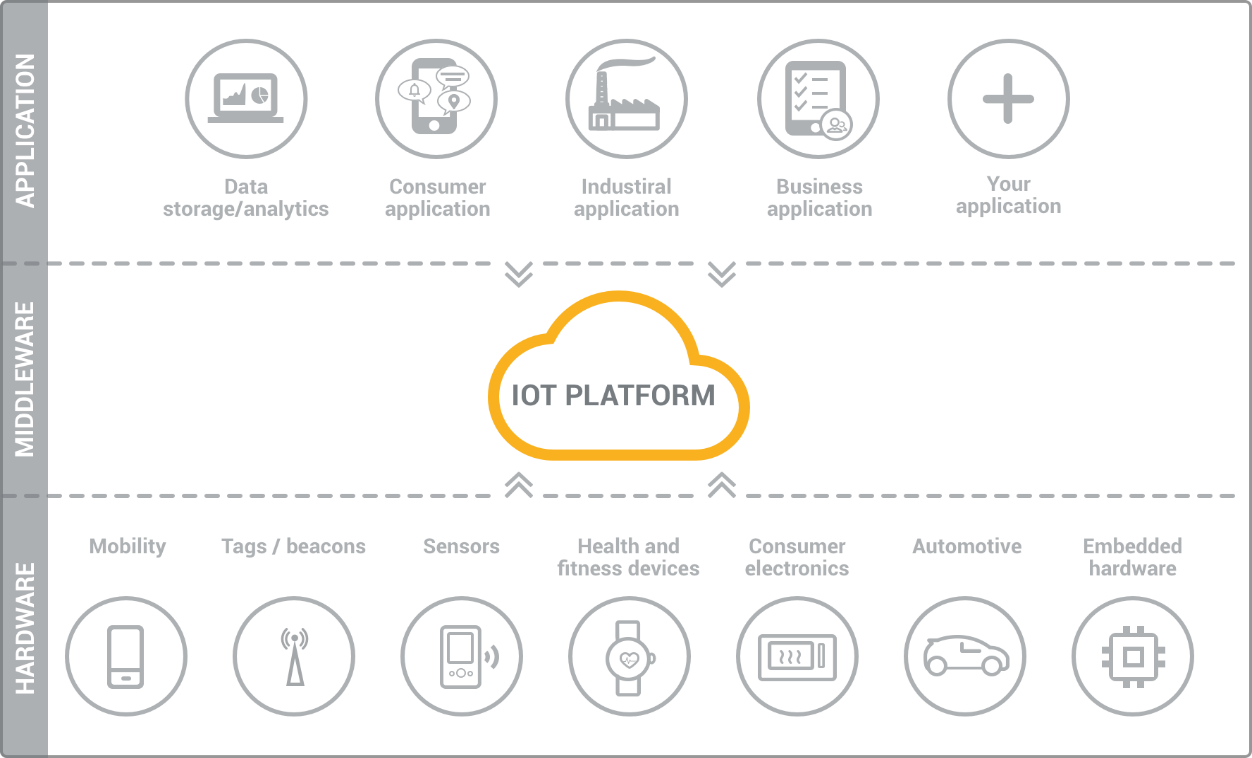
Internet Vạn Vật, hay cụ thể hơn là Mạng lưới vạn vật kết nối Internet hoặc là Mạng lưới thiết bị kết nối Internet (tiếng Anh: Internet of Things, viết tắt IoT) là một liên mạng, trong đó các thiết bị, phương tiện vận tải (được gọi là "thiết bị kết nối" và "thiết bị thông minh"), phòng ốc và các trang thiết bị khác được nhúng với các bộ phận điện tử, phần mềm, cảm biến, cơ cấu chấp hành cùng với khả năng kết nối mạng máy tính giúp cho các thiết bị này có thể thu thập và truyền tải dữ liệu. Hệ thống IoT cho phép vật được cảm nhận hoặc được điều khiển từ xa thông qua hạ tầng mạng hiện hữu, tạo cơ hội cho thế giới thực được tích hợp trực tiếp hơn vào hệ thống điện toán, hệ quả là hiệu năng, độ tin cậy và lợi ích kinh tế được tăng cường bên cạnh việc giảm thiểu sự can dự của con người. Khi IoT được gia tố cảm biến và cơ cấu chấp hành, công nghệ này trở thành một dạng thức của hệ thống ảo-thực với tính tổng quát cao hơn, bao gồm luôn cả những công nghệ như điện lưới thông minh, nhà máy điện ảo, nhà thông minh, vận tải thông minh và thành phố thông minh. Mỗi vật được nhận dạng riêng biệt trong hệ thống điện toán nhúng và có khả năng phối hợp với nhau trong cùng hạ tầng Internet hiện hữu.

// Khúc này chua đủ thì copy thêm giới thiệu tổng quan trên wiki

Sự phát triển IoT giúp tạo ra một hệ sinh thái giúp kết nối nhiều hệ thống với nhau, giúp tăng hiệu suất cho các tác vụ. Các thiết bị có thể chia sẻ dữ liệu với nhau, phân tích các dữ liệu thu thập được, và sử dụng các dữ liệu đó để tự động thực hiện các yêu cầu đặt ra mà không cần sự can thiệp vật lý từ con người. Hiện nay, có rất nhiều công ty như Eddie Stobart Transport and Logistics, Amazon, Dell, Aviva,... phát triển nhanh chóng dựa vào việc tối ưu công nghệ IoT.

Về cơ bản, một nền tảng IoT là hệ thống hoàn chỉnh, kết hợp các thiết bị phần cứng, các kết nối, phần mềm, các ứng dụng thông minh để tạo ra một giải pháp hiệu quả cho việc quản lý, cấu hình các thiết bị, thu thập và phân tích dữ liệu, kết nối đến các hệ thống cloud và server.

Các thành phần cấu tạo nên một hệ thống IoT bao gồm:

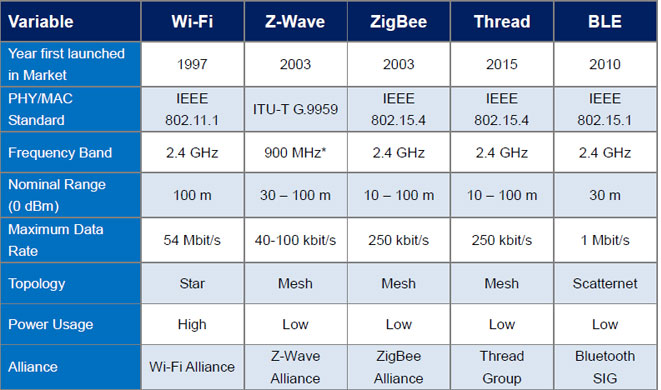
* Các thiết bị kết nối: là các máy móc, cảm biến, động cơ nhằm thực hiện các tác vụ được yêu cầu, ngoài ra còn có thể thu thập dữ liệu từ môi trường, kết nối, tương tác với các thiết bị khác, gửi dữ liệu về server,...
* Các kết nối không dây: đóng vai trò như một kênh giúp các thiết bị có thể giao tiếp và truyền dữ liệu qua lại, nhận lệnh từ server. Các yêu cầu đặt ra cho các chuẩn kết nối trong một nền tảng IoT là độ tin cậy, bảo mật, tốc độ và độ tiêu hao năng lượng.
* Phần mềm xử lý dữ liệu: có thể được chạy ở server hoặc cloud, có nhiệm vụ phân tích các dữ liệu thu thập được, đưa ra các yêu cầu cần thực hiện bên trong hệ thống IoT
* Giao diện người dùng: cung cấp cho người dùng cái nhìn tổng thể về hoạt động của toàn bộ hệ thống, hỗ trợ người dùng tương tác với các thiết bị, quản lý hệ thống, hiển thị các thông tin dữ liệu một cách thân thiện với người dùng
* 

### 2.1.2. Các kết nối trong hệ thống IoT thực tế

Kết nối IoT – kết nối vạn vật, trong đó trọng điểm là các giao thức kết nối để liên kết các thiết bị. Các ứng dụng IoT đều có các ràng buộc rõ ràng về phạm vi kết nối (không dây, có dây), điện năng tiêu thụ, tốc độ truyền tai dữ liệu, độ tin cậy và bảo mật. Giao thức truyền thông tin có rất nhiều với nhiều chủng loại thiết bị khác nhau dành cho các nhu cầu khác nhau. Những giao thức phổ biến trong các hệ thống IoT hiện nay:



Những giao thức phổ biến trong các hệ thống IoT hiện nay:

* Bluetooth: Bluetooth xuất hiện hầu hết ở các thiết bị như máy tính, điện thoại… Bluetooth hỗ trợ tốc độ truyền tải dữ liệu lên tới 70 Kbps trong phạm vi 10 – 100m và có công suất tiêu thụ lên đến 1W. Tuy nhiên, các chuẩn công nghệ Bluetooth mới hơn có hỗ trợ BLE – Bluetooth Low Energy giúp giảm công suất đáng kể (0.01 – 0.5W).
* LoRa: LoRa là công nghệ giao tiếp ở khoảng cách xa, có thể lên đến hàng kilomet mà không cần đến bộ khuếch đại. Hơn thế nữa, công suất tiêu thụ của các thiết bị sử dụng công nghệ LoRa là thấp, khoảng 0.1W. Công nghệ LoRa dễ bắt gặp trong các ứng dụng Sensor Network. Tuy nhiên, Tốc độ truyền thấp khoảng 0.3 – 50Kbps.
* Wifi: Wifi là giao thức thường được sử dụng để truyền hình ảnh, âm thanh nhờ băng thông lớn. Tuy nhiên, Wifi có phạm vi truyền sóng nhỏ và công suất tiêu thụ lớn.
* RS485: Có tốc độ truyền tương đối cao và ổn định, tuy nhiên, RS485 là giao thức có dây nên chỉ phù hợp để kết nối các trạm trong mạng cục bộ.
* Ethernet: là công nghệ truyền thông dùng để kết nối các mạng LAN cục bộ, có tốc độ cao hơn RS485, ít bị gián đoạn,độ bảo mật cao, tuy nhiên, giao thức này là giao thức có dây.
* Zigbee: Zigbee là một giao thức mở, sử dụng mạng lưới (mesh network) để mở rộng hệ thống. Zigbee phù hợp với hệ thống cần nhiều nút mạng. Tốc độ truyền khoảng 40-250Kbps. Cũng như các giao thức trên, công nghệ này cũng tiêu hao ít năng lượng. Nhược điểm của giao thức này là khoảng cách giữa các nút còn hạn chế, khó xuyên tường, đường truyền có độ tin cậy thấp, chưa thực sự thích hợp cho các dự án IoT thông minh ở Việt Nam.
* Z-Wave: Tương tự như Zigbee, Z-Wave cũng sử dụng mạng lưới để mở rộng phạm vi kết nối. Ngoài ra, Z-Wave có hầu hết các ưu điểm của Zigbee và cải thiện được khoảng cách truyền sóng giữa các nút (lên đến 100m), Z-Wave sử dụng tần số 900MHz – dải tần ít được các giao thức sử dụng nên tránh được các xung đột. Vì thế, giao thức Z-Wave có độ tin cậy cao hơn. Đây là công nghệ đang được chú ý và được các nhà sản xuát tập trung nhiều hơn.
* 

// Nếu thiếu sẽ extend khúc này ra + hình ảnh

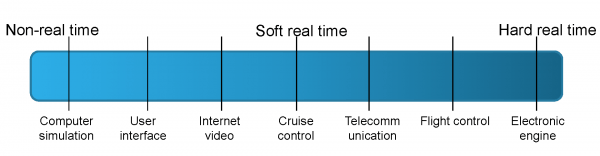
// Khúc này copy lý thuyết lora, lorwan Modbus. Zwave đông nghẹt vào,

// RTOS các kiểu

Hệ điều hành thời gian thực được thiết kế ra cho các nhiệm vụ đặc biệt. Các ứng dụng cần được thực thi với thời gian thật chính xác, các lỗi phát sinh cần được cô lập và xử lý nhanh chóng. Mọi sự chậm trễ, lỗi phát sinh không lường trước có thể khiến hệ thống bị đổ vỡ. Vì vậy, RTOS sử dụng trong những ứng dụng yêu cầu thời gian đáp ứng nhanh, chính xác về thời gian. Vi điều khiển có tài nguyên rất giới hạn. Do đó, hệ điều hành này chỉ tập trung vào một số ít các tính năng. Chúng cố gắng tối ưu tối đa số luồng, bộ lập lịch và các tác vụ (task) trên một hệ thống cỡ nhỏ.



FreeRTOS là một hệ điều hành nhúng thời gian thực (Real Time Operating System) mã nguồn mở được phát triển bởi Real Time Engineers Ltd, sáng lập và sở hữu bởi Richard Barry. FreeRTOS được thiết kế phù hợp cho nhiều hệ nhúng nhỏ gọn vì nó chỉ triển khai rất ít các chức năng như: cơ chế quản lý bộ nhớ và tác vụ cơ bản, các hàm API quan trọng cho cơ chế đồng bộ. Nó không cung cấp sẵn các giao tiếp mạng, drivers, hay hệ thống quản lý tệp (file system) như những hệ điều hành nhúng cao cấp khác. Tuy vậy, FreeRTOS có nhiều ưu điểm, hỗ trợ nhiều kiến trúc vi điều khiển khác nhau, kích thước nhỏ gọn (4.3 Kbytes sau khi biên dịch trên Arduino), được viết bằng ngôn ngữ C và có thể sử dụng, phát triển với nhiều trình biên dịch C khác nhau (GCC, OpenWatcom, Keil, IAR, Eclipse, …), cho phép không giới hạn các tác vụ chạy đồng thời, không hạn chế quyền ưu tiên thực thi, khả năng khai thác phần cứng. Ngoài ra, nó cũng cho phép triển khai các cơ chế điều độ giữa các tiến trình như: queues, counting semaphore, mutexes.

.

(Tùy theo nội dung nghiên cứu mà sinh viên chọn tiêu đề cho phù hợp)

Đối với nội dung tìm hiểu lý thuyết, sinh viên cần trình bày:

* Ngắn gọn và liên quan trực tiếp đến đề tài
* Mỗi chương liên quan đến một vấn đề
* Với phần lý thuyết không quan trọng, sinh viên có thể đưa vào mục tài liệu tham khảo.

# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

## Yêu cầu thiết kế

Từ các mục tiêu đặt ra của đề tài, phần cứng của hệ thống cần thực hiện bao gồm:

**Gateway Modbus Serial**:

* Có các cổng RS485, RS232, UART,… để giao tiếp với các thiết bị chuẩn Modbus có sẵn trên thị trường cũng như dễ dàng cài đặt, hiệu chỉnh.
* Có cổng Ethernet để triển khai mô hình Modbus TCP hoặc các dịch vụ trên lớp mạng để giao tiếp với IoT Platform trên server.
* Phần cứng vi điều khiển đủ mạnh để đảm bảo chạy các ứng dụng một cách ổn định.
* Đèn báo trạng thái, buzzer để thông báo cho người dùng.
* Hỗ trợ tối đa 4 cổng RS485 để quản lý được nhiều thiết bị.

**Gateway LoraWAN**:

* Tương thích với phần cứng máy tính nhúng.
* Khối nguồn để đảm bảo khối concentrator hoạt động ổn định
* Kích thước nhỏ gọn, dễ lắp đặt.

**Gateway Z-wave:**

* Do Z-wave là mạng kết nối theo kiểu mess nên gateway khá đơn giản (phần cứng như một node) nên có thể tích hợp chung với Gateway LoraWAN

**Các thiết bị đầu cuối:**

* Trong phạm vi đề tài sẽ thực hiện các bản thử nghiệm của một số thiết bị iot thông dụng, trong đó gồm: Van nước điều khiển qua Z-wave, Z-wave node có sẵn các ngoại vi để tích hợp cho các thiết bị có sẵn (relay, cảm biến chuyển động), LoraWan node và Modbus node dùng thu thập dữ liệu cảm biến thời tiết (ánh sáng, nhiệt độ, độ ẩm,…). Ngoài ra, đề tài cũng sử dụng thiết bị công nghiệp có sẵn (biến tần, plc,…) để thử nghiệm độ tương thích.
* Đảm bảo hỗ trợ ít nhất một trong ba giao thức của gateway để giao tiếp truyền tải dữ liệu
* Hỗ trợ các ngoại vi cơ bản nhất (GPIO, OneWire, SPI, I2C,…) để điều khiển cũng như giao tiếp với các cảm biến thông dụng.
* Hỗ trợ các cổng debug để dễ dàng kiểm tra, hiệu chỉnh, sửa lỗi. Đèn led, nút nhấn để điều khiển bằng tay.
* Thiết kế khối nguồn hợp lý, linh kiện để đem lại mức tiêu thụ năng lượng thấp cho các thiết bị đầu cuối không dây, các thiết bị dùng nguồn pin.
* Nắm vững các quy tắc layout với các khối đặc thù như nguồn, vi điều khiển, cổng ethenet, chuẩn RS485. Các quy tắc routing về đường dây, ăn-ten cho các thiết bị không dây.
* Sắp xếp linh kiện một cách hợp lý. Dễ dàng thay thế hoặc đo đạc trong quá trình kiểm tra hoạt động.

## Phân tích

Sau đây là các phương án cho từng thiết bị

* Gateway Modbus Serial

|  |  |  |  |
| --- | --- | --- | --- |
| Phương án | Mô tả hoạt động | Ưu điểm | Nhược điểm |
| Sử dụng STM32F407 làm vi điều khiển chính cho Gateway | Triển khai Modbus Protocol ở chế độ master.  Kết nối mạng nội bộ LAN để có thể gửi nhận dữ liệu tổng hợp được đến IoT Platform trên server  Các ngoại vi cơ bản để thông báo chức năng. | Cấu hình và ngoại vi cơ bản đáp ứng được yêu cầu.  Được dùng phố biến nên có sẵn các thư viện cơ bản, cộng đồng hỗ trợ động đảo. Dễ tìm thấy và mua ở thị trường trong nước Giá thành vừa phải khi so cùng cấu hình so với một số nhà sản xuất khác | Giá thành cao |
| Sử dụng LAN8720A IC giao tiếp giữa vi điều khiển chính và Ethernet port | IC giao tiếp Ethernet, hỗ trợ tốc độ cao 10/100 Mbps. | Rẻ, nhỏ gọn, giao tiếp RMII dễ sử dụng.  Thư viện có sẵn cho STM32.  Đơn giản hoá trong việc xử lý các gói ở lớp PHY | Cần vi điều khiển hỗ rợ, đủ mạnh để chạy các kết nối mạng. |
| MAX485, MAX3232 | IC chuyển tiếp UART-TTL sang RS485/RS232. | Rẻ. Chuẩn giao tiếp thích hợp cho truyền xa hơn TTL | Một số ic không chính hãng chạy không ôn đỉnh. |
|  |  | Rẻ | Tín hiệu nhận được sẽ sai khi đi lệch ô. |
| RAK833 | Lora Concentrator hỗ trợ nhận tín hiệu |  |  |
| RAK811 | Lora Node |  |  |
| RAK4600 | Lora Node + BLE |  |  |
| ZM5202 | Z-wave device |  |  |
| IC AP3211 | IC AP3211 có kích thước nhỏ gọn, dòng điện đầu ra tối đa 1.5 ampe. |  |  |

- Gateway Lora và Gateway Z-wave

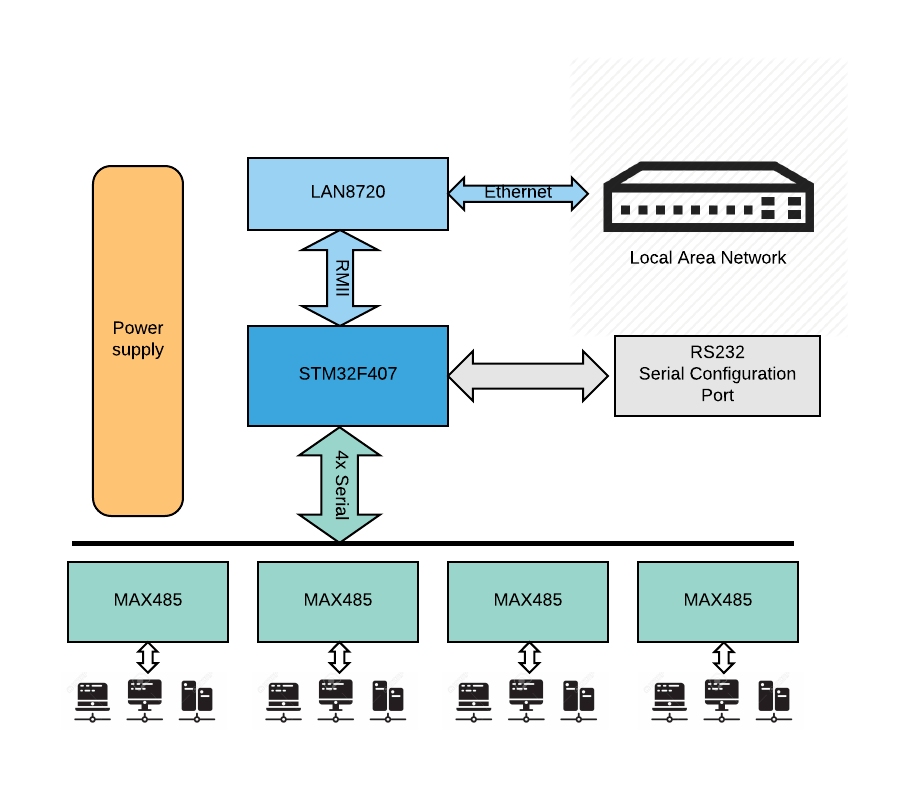
|  |  |  |  |
| --- | --- | --- | --- |
| Sử dụng máy tính nhúng Rasberry Pi 3B+ làm khối điều khiển chính của gateway | Raspberry máy tính nhúng nhỏ được tích hợp nhiều phần cứng mạnh mẽ đủ khả năng chạy hệ điều hành và cài đặt được nhiều ứng dụng trên nó. Phiển bản Rasberry Pi 3+ sử dụng chip Broadcom BCM2837B0, quad-core @1.4Ghz, 1 GB RAM, 4xUSB 2.0, 10/100mb Ethernet, Wifi 802.11n và 40 GPIO pinout mở rộng. | Rẻ, cấu hình mạnh, dễ tìm và dễ mua ở thị trường trong nước. Được sử dụng phổ biến, cộng đồng hỗ trợ mạnh, nhiều thư viện, chạy được nhiều nền tảng hệ điều hành thông dụng như Linux, Raspbian nên dễ dàng trong việc triển khai các ứng dụng. | Không đủ nhiều ngoại vi như |
| Sử dụng RAK833, RAK831 Lora Concentrator làm thiết thu phát dữ liệu đến các thiết bị đầu cuối. | Lora Concentrator hỗ trợ nhận tín hiệu với 8 kênh đồng thời thu nhận tín hiệu. Do đó nó có thể làm một đảm bao thu nhận cho khoảng 500 thiết bị đầu cuối (theo nhà sản xuất). Hỗ trợ LoraWAN Stack 1.0.2 Hỗ trợ toàn bộ các băng tần của LoraWAN: 433MHz, 470MHz, 865MHz, 868MHz, 915MHz, 920MHz, 923MHz | Nhỏ gọn, giao tiếp SPI đơn giản, dễ lắp đặt.  Có sẵn thư viện từ Semtech, tương thích trên các nền tảng hệ điều hành phổ biến (OpenWRT, Raspbian,…) | Giá thành tương đối đắt và chưa phổ biến trong nước Cần Antenna nhạy để thu truyền được khoảng cách xa. |
| Sử dụng nguồn xung với IC AP3211. | IC AP3211 có kích thước nhỏ gọn, dòng điện đầu ra tối đa 1.5 ampe. Mạch thiết kế đơn giản. | Có chế độ bảo vệ quá dòng, bảo vệ quá áp, bảo vệ quá nhiệt độ trên IC AP3211(nhiệt độ vượt quá 1600C). Ngoài ra, chip tự động tắt hoàn toàn nếu điện áp vào dưới 3,8 volt và IC hoạt động nếu điện áp đầu vào trên 3.6 volt |  |
| Tích hợp ZM5202 Z-wave module | Z-wave module hỗ trợ các để làm Gateway cho giao thức Zwave |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

-IOT node

|  |  |  |  |
| --- | --- | --- | --- |
| Phương án | Mô tả hoạt động | Ưu điểm | Nhược điểm |
| Sử dụng STM32F103 làm vi điều khiển chính cho node | Thu thập dữ liệu từ 6 port IO có sẵn, bao gồm 3 Input coupler, 2 Analog và 1 Onewire pot.  Triển khai Modbus Protocol ở chế độ slave.  Giao tiếp với module RAK để kết nối vào mạng LoraWAN như 1 node  Các ngoại vi điều khiển bằng tay | Cấu hình và ngoại vi cơ bản đáp ứng được yêu cầu.  Được dùng phố biến nên có sẵn các thư viện cơ bản, cộng đồng hỗ trợ động đảo. Dễ tìm thấy và mua ở thị trường trong nước Giá thành vừa phải khi so cùng cấu hình so với một số nhà sản xuất khác | Giá thành còn rẻ nhưng mua nhiều nó mắc vcl |
| Sử dụng EEPROM AT24C4 để lưu các lại các thông số tuỳ chỉnh | Lưu lại các giá trị tuỳ chỉnh, bao gồm các cài đặt liên quan đến từng port IO, các thông số của cổng Serial và các thông số của LoraWAN stack | Giá thành rẻ, giao tiếp I2C đơn giản dễ sử dụng.  Số lần ghi xoá nhiều cũng như ổn định hơn so với việc dùng bộ nhớ Flash tích hợp sẵn trên vi điều khiển để lưu lại các giá trị |  |
| Sử dụng RAK811/ RAK4600 | Module LoraWAN cho phép kết nối vào mạng LoraWAN để truyền nhận dữ liệu. Trên module đã có sẵn LoraWAN stack 1.0.2. Trong đề tài sử dụng 2 loại module từ RAK, trong đó RAK4600 sẽ hỗ trợ thêm Bluetooth Low Energy | Hỗ trợ AT Command thông qua serial để giao tiếp với vi điều khiển góp phần làm đơn giản hoá việc triển khai LoraWAN stack trên các Endnode device. | Firmware hoạt động chưa thực sự ổn định. Một vài tính năng liên quan đến LoraWAN vẫn còn đang phát triển. Một vài thông số liên quan đến năng lượng phát, thời gian truyền sóng, kích thước gói tin vẫn còn bị giới hạn và không thể mở khoá.  Tương tự Gateway, các thiết bị đầu cuối cần antenna |

3.3 Sơ đồ khối tổng quát

Gateway Modbus Serial:

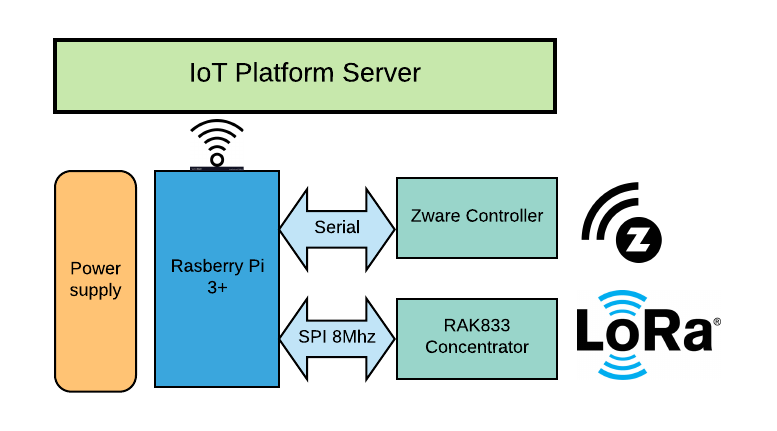


Giải thích:

Khối vi điều khiển STM32F407VET6 đóng vai trò làm trung tâm xử lý dữ liệu, bao gồm các nhiệm vụ chính:

* Chạy lớp giao thức Modbus ở chế độ Master, bao gồm tổng cộng 4 Master riêng biệt dưới giao diện serial thông qua RS485.
* Chạy các stack mạng thông qua giao diện RMII với LAN8720.

Gateway Zwave and Lora



Giải thích:

Khối trung tâm: máy tính nhúng Rasberry Pi 3+ có vài trò là trung tâm trung chuyển dữ liệu. Máy tính nhúng sẽ chạy các driver để tương tác với các khối tập trung dữ liệu (bao gồm cả 2 giao thức Lora và Zwave) và các API cung cấp bởi IoT Platform Server để thực hiện các luồng dữ liệu từ các khối tập trung của từng giao thức lên IoT Server và ngược lại.

Zware Controller: là một module Zwave, có nhiệm vụ control các node trong mạng Zwave, kết nối với máy tính nhúng qua đường UART Serial.

RAK833 Concentrator: bộ tập trung dữ liệu của LoraWAN, có nhiệm vụ lắng nghe các gói tin Lora ở các dãy tần số và độ trải phổ cài đặt trước để gửi sang máy tính nhúng, đồng thời nhận gói tin từ máy tính nhúng để phát đến các node trong mạng.

IoT Platform Server: // khúc này Thông viết

Khối nguồn cung cấp: Cung cấp đúng mức điện áp và đảm bảo đủ dòng cần thiết cho các module, đặc biệt là khối RAK833 Concentrator.

//theo yêu cầu từ nhà sản xuất, LoraWAN concentrator cần tối thiểu một bộ nguồn có thể tải điện áp 3V3 ở 1A. Do đó IC AP3211 với ưu điểm nhỏ gọn với dòng chịu tải tối đa 1.5A

Sơ đồ mạch chi tiết

Lorawan and Zwave gateway

Khối nguồn cung cấp:

Trong thiết kế đặt ra, board được sử dụng như một pack mở rộng để tương thích với khối GPIO của Rasperry. Do đó, việc sử dụng chung nguồn với Rasberry là một cách đơn giản và tối ưu cho thiết kế. Tuy nhiên cần phải xem xét các thông số:

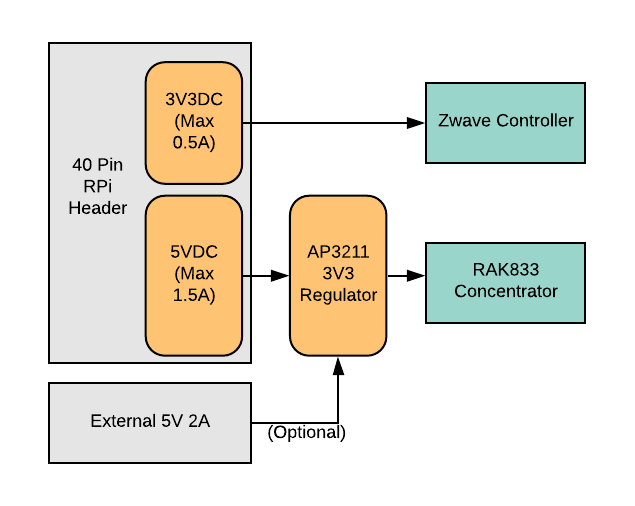
Đối với Rasberry:

* Pin 5V được nối trực tiếp với nguồn điện cung cấp chính của Rasberry và được bảo vệ bằng cầu chì tự phục hồi, dòng mong muốn có thể đạt đến 1.5A trong trường sử dụng nguồn đầu vào chuẩn của Rasberrypi.org (5V 2.5A).
* Pin 3V3 được hạ áp thông qua nguồn xung tích hợp trên board, có thể lên 0.5A. Tuy nhiên theo nhà sản xuất, Pin này không nên khai thác như một nguồn 3V3 chính cho các ứng dụng bên ngoài, thay vào đó các ứng dụng dùng 3V3 nên được hạ áp từ pin 3V3 bởi một mạch hạ áp ngoài.

Đối với RAK833 Concentrator: theo tài liệu tham khảo từ nhà sản xuất, cần tối thiểu một bộ nguồn 3V3/1A DC.

Đối với Zwave ZM5202: Dòng tối đa khi module thu phát tín hiệu là 48mA.

Sau khi xem xét các thông số và tham khảo thêm nguồn thông tin từ các thiết kế có sẵn, phần nguồn được thiết kế:



* RAK833 Concentrator được cung cấp điện áp 3V3 từ nguồn xung của IC AP3211. Nguồn này sẽ được thiết kế để chịu tải tối đa 1,5A, được lấy đầu vào từ Pin 5V của Rasberry.
* Zwave Controller có dòng sử dụng thấp nên có thể sử dụng nguồn 3V3 cung cấp bởi Rasberry, phương án này góp phần đảm bảo nguồn của RAK833 không bị ảnh hưởng.
* Yêu cầu thiết kế
  + Liệt kê các yêu cầu đặt ra
  + Ghi cụ thể (có tính định lượng) các yêu cầu, chi tiết kỹ thuật.
* **Phân tích**
  + Phân tích rõ cách thức dẫn đến phương pháp thiết kế từ yêu cầu đã đặt ra
  + Nêu rõ ưu điểm và khuyết điểm của từng phương pháp, từ đó lựa chọn phương pháp phù hợp
* Vẽ sơ đồ khối tổng quát và **giải thích** (nếu mạch đơn giản thì lược bỏ phần này)
  + Phải giải thích rõ nhiệm vụ, chức năng từng khối
* Vẽ sơ đồ khối chi tiết và **giải thích**
  + Phải giải thích rõ nhiệm vụ, chức năng từng khối
* Tính toán và vẽ sơ đồ mạch chi tiết
  + Thiết kế, vẽ sơ đồ mạch chi tiết và tính toán từng khối đã nêu trong phần trên

# THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM (NẾU CÓ)

## Yêu cầu

Modbus Gateway:

* Gateway phải chạy được Modbus Stack ở chế độ Master để thu thập, thay đổi, quản lý dữ liệu cho các thiết bị đầu cuối. Đồng thời để nâng cao tối đa số lượng thiết bị trong mạng, 4 cổng RS485 trên board đều phải được khai thác, có thể xem như Gateway quản lý 4 mạng Modbus riêng biệt.
* Gateway được kết nối đến IoT Platform thông qua MQTT, trong đó mỗi gateway sẽ có địa chỉ ID định danh riêng trong mạng, đồng thời đảm bảo 2 luồng dữ liệu chính là downlink (gói tin yêu cầu từ IoT Platform) và uplink (gói tin phản hồi yêu cầu và gói tin đồng bộ)
* Gateway phải có chế độ tuỳ chỉnh, bao gồm: các thông số liên quan đến địa chỉ IP của Gateway trong mạng (Static IP, Netmask, Default Gateway), Gateway ID, MQTT Server IP và các thông số để điều chỉnh cho mạng Modbus: Baudrate, Stopbit, Parity, ...
* Các cơ chế thông báo lỗi, đèn LED và Buzzer thông báo lưu lương và lỗi trong quá trình chạy thực nghiệm

LoraWan Gateway và Zwave Gateway

* Gateway LoraWAN làm trạm trung chuyển các gói tin LoraWAN từ các LoraWAN node đến LoraWAN Server.
* Gateway Zwave giao tiếp với máu tính nhúng thông qua serial, qua đó luồng dữ liệu của Zwave

IoT Node:

* Xây dựng node thu thập dữ liệu môi trường, node phải hỗ trợ các kết nối ngõ vào cho các cảm biến thu thập thông dụng như GPIO Input, Analog Converter, Onewire,...
* Trong nhiều trường hợp, các cổng kết nối sẽ thu thập dữ liệu từ các cảm biến khác nhau, do đó phải có các profile riêng và cơ chế cài đặt cho từng cảm biến.
* Hỗ trợ Modbus Serial ở chế độ Slave, chuẩn hoá các giá trị thu thập được trên những thanh ghi đã được định danh trước
* Hỗ trợ các Command để giao tiếp với module RAK, bao gồm cài đặt, kết nối và gửi nhận dữ liệu vào mạng LoraWAN.
* Hỗ trợ các command để kết nối với App trên máy tính để dễ dàng thay đổi chính sửa cũng như quản lý node.

Valve Controller

PHân tích

* Yêu cầu đặt ra cho phần mềm
  + Liệt kê các yêu cầu đặt ra
  + Ghi cụ thể (có tính định lượng) các yêu cầu, chi tiết kỹ thuật.
* **Phân tích**
  + Phân tích các yêu cầu để đưa ra phương pháp thực hiện chương trình
* Vẽ lưu đồ giải thuật tổng quát và **giải thích** (nếu giải thuật đơn giản thì lược bỏ phần này)
  + Phải giải thích rõ nhiệm vụ, chức năng từng phần
* Vẽ lưu đồ giải thuật chi tiết và **giải thích**
  + Phải giải thích rõ nhiệm vụ, chức năng từng phần

# KẾT QUẢ THỰC HIỆN

## Phần cứng

* + 1. LoraWan và Zwave Gateway

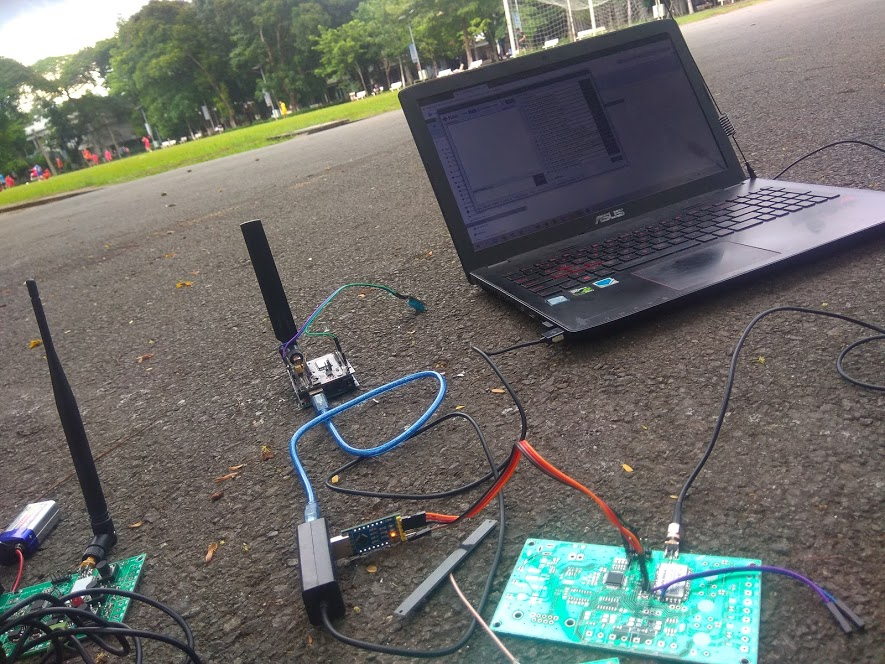


// Đo nguồn

//

* + 1. IoT Node

Đo đạt khoảng cách



* + 1. Zwave node
    2. Valve nước

Trong phần này, sinh viên mô tả:

* Trình bày **cách thức đo đạc, thử nghiệm** 
  + Ghi rõ các thiết bị sử dụng và sơ đồ kết nối trong việc thử nghiệm
  + Ghi rõ các phần mềm sử dụng trong việc viết và thực thi chương trình
  + Ghi rõ cách bước tiến hành thử nghiệm (phần cứng và phần mềm)
* Trình bày số liệu đo đạc
  + Thực hiện thu thập số liệu trong nhiều trường hợp
  + Ghi rõ số liệu đo đạc thu được dưới hình thức bảng biểu, đồ thị …
* **Giải thích và phân tích về kết quả thu được**
  + Cần giải thích rõ ràng số liệu thu được trên các bảng biểu, đồ thị, dạng sóng …
  + Phân tích các số liệu để biết kết quả đã thực hiện là phù hợp, đạt yêu cầu

Nếu những bảng số liệu và kết quả mô phỏng quá nhiều, sinh viên có thể trình bày đưa vào phần Phụ Lục.

Ví dụ về hình minh họa: (dùng chức năng **Insert Caption** để tạo liên kết cho Danh sách hình minh họa)



Hình 5‑1 Kết quả thi công



Hình 5‑2 Kết quả mô phỏng

Ví dụ về Bảng số liệu

Bảng 1 Thông số hệ thống

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số 1 | Thông số 2 | Thông số 3 | Thông số 4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Sinh viên tóm tắt những điều rút ra được từ kết quả đề tài, những kinh nghiệm có được sau khi thực hiện đề tài. **Ưu và khuyết điểm** của kết quả nghiên cứu đề tài cũng được trình bày trong mục này. Sinh viên cần so sánh với mục tiêu đặt ra trong chương 1.

## Hướng phát triển

Sinh viên trình bày hướng phát triển và khả năng ứng dụng của đề tài

# TÀI LIỆU THAM KHẢO

Trong mục này, sinh viên liệt kê những tài liệu đã tham khảo khi thực hiện đề tài luận văn. Những nội dung trình bày ở mục trên có tham khảo tài liệu thì sinh viên cần ghi chú bằng chỉ số (ví dụ [1], [2]). Chỉ số này cần tương ứng danh mục tài liệu tham khảo. Sinh viên xem thêm hướng dẫn cách viết trích dẫn kiểu IEEE.

Ví dụ:

1. Tống Văn On, “Thiết kế mạch số với VHDL & Verilog”, Nhà xuất bản Lao động Xã Hội, 2007.
2. Altera Corp., “SDRAM Controller for Altera’s DE2/ DE1 boards”, [www.altera.com](http://www.altera.com)

# PHỤ LỤC

Trong phần này, sinh viên có thể trình bày:

* Những kết quả nghiên cứu bổ sung mà trong phần Kết quả luận văn chưa trình bày hết.
* Phần mã nguồn chương trình, sinh viên cũng có thể trình bày trong mục này. Để ngắn gọn, sinh viên chỉ đưa những mã nguồn chính vào phần Phụ lục.
* Sơ đồ toàn mạch chi tiết