

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

**HỆ THỐNG GIÁM SÁT MÔI TRƯỜNG VÀ ĐIỀU
KHIỂN THIẾT BỊ ỨNG DỤNG LORA VÀ
THINGSBOARD IOT PLATFORM**

GVHD: ThS. Trần Văn Hoàng

SVTH: Huỳnh Tấn Đạt

MSSV: 1410783

TP. HỒ CHÍ MINH THÁNG 6, NĂM 2018

-----☆-----
Số: _____/BKĐT
Khoa: **Điện – Điện tử**
Bộ Môn: **Điện Tử**

-----☆-----

NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP

1. HỌ VÀ TÊN: HUỖNH TẤN ĐẠT MSSV: 1410783
2. NGÀNH: **ĐIỆN TỬ - VIỄN THÔNG** LỚP: DD14DV2
3. Đề tài: **HỆ THỐNG GIÁM SÁT MÔI TRƯỜNG VÀ ĐIỀU KHIỂN THIẾT BỊ ỨNG DỤNG LORA VÀ THINGSBOARD IOT PLATFORM**
4. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):
 - ✓ Xây dựng mạng không dây Lora, gồm 1 node master và 2 node slave
 - ✓ Thu thập dữ liệu môi trường từ các slave và gửi đến master để đưa lên server
 - ✓ Điều khiển tự động hoặc thủ công các thiết bị điện tại các node thông qua giao diện web
 - ✓ Đưa ra các cảnh báo khi xảy ra sự cố
5. Ngày giao nhiệm vụ luận văn: 1/2018
6. Ngày hoàn thành nhiệm vụ: 14/5/2018
7. Họ và tên người hướng dẫn: Trần Văn Hoàng

Phản hướng dẫn

.....
.....

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

Trong cuộc sống, thành công là điều không dễ dàng. Bên cạnh các yếu tố cá nhân không thể thiếu sự giúp đỡ từ mọi người xung quanh.

Đầu tiên em xin gửi lời cảm ơn chân thành đến thầy ThS. Trần Văn Hoàng, người luôn tâm huyết với ngành, truyền đạt những kiến thức, kinh nghiệm quý báu cho em trong suốt thời gian qua. Thầy là người hướng dẫn, định hướng, cũng là người tạo động lực và tạo mọi điều kiện tốt nhất để em hoàn thành khóa luận này. Đồng thời, em cũng chân thành gửi lời cảm ơn tới các thầy cô đang giảng dạy tại khoa Điện - Điện tử, trường đại học Bách Khoa thành phố Hồ Chí Minh. Nhờ các thầy cô, em mới được tiếp cận với những kiến thức mới hết sức bổ ích và những ứng dụng thực tiễn thú vị. Ngoài ra, còn có bạn bè luôn sẵn sàng đồng hành giúp đỡ em khi khó khăn về học tập và cuộc sống, những người cùng nhau chia sẻ, niềm vui, nỗi buồn những ngày qua, đó thật sự là những kỉ niệm đẹp đẽ dưới mái trường Bách Khoa này.

Lời cuối cùng, con xin gửi lời cảm ơn chân thành tới ba mẹ. Người đã sinh ra, nuôi dưỡng và dạy dỗ con nên người. Con cảm ơn gia đình đã luôn yêu thương, ủng hộ và tạo động lực để con cố gắng trong cuộc sống.

Tp. Hồ Chí Minh, ngày 12, tháng 6, năm 2018.

Sinh viên

Huỳnh Tấn Đạt

TÓM TẮT LUẬN VĂN

Luận văn “HỆ THỐNG GIÁM SÁT MÔI TRƯỜNG VÀ ĐIỀU KHIỂN THIẾT BỊ DỰA TRÊN LORA VÀ THINGSBOARD IOT PLATFORM” trình bày về mô hình giám sát các yếu tố môi trường và điều khiển tự động các thiết bị máy bơm trong nông nghiệp. Mục tiêu của luận văn này là xây dựng một hệ thống thu thập các yếu tố môi trường tại nhiều nơi cách xa nhau và điều khiển cách thiết bị tại đó thông qua smartphone hoặc laptop. Hướng ứng dụng của luận văn: giúp người nông dân dễ dàng giám sát ruộng vườn, thuận tiện trong việc điều khiển tưới tiêu từ xa, cũng như dựa vào các thông số thu thập được để có thể phân tích, tìm ra cách trồng trọt tối ưu.

MỤC LỤC

LỜI CẢM ƠN	1
TÓM TẮT LUẬN VĂN	2
DANH SÁCH HÌNH MINH HỌA	6
DANH SÁCH BẢNG SỐ LIỆU	8
1. GIỚI THIỆU	9
1.1 Tổng quan	9
1.2 Tình hình nghiên cứu trong và ngoài nước:	9
1.3 Nhiệm vụ luận văn	10
1.3.1 Yêu cầu đề tài	10
1.3.2 Mục tiêu đề tài	11
1.3.3 Giới hạn đề tài	12
2. LÝ THUYẾT	13
2.1 Tìm hiểu về công nghệ LoRa	13
2.1.1 Giới thiệu	13
2.1.2 Đặc tính công nghệ LoRa	15
2.1.3 Các loại module LoRa và chuẩn giao tiếp với vi điều khiển	15
2.1.4 Giới thiệu module LoRa E32-TTL-100	16
2.1.5 Thông số	17
2.1.6 Các chế độ hoạt động	18
2.1.7 Phần mềm cấu hình	18
2.2 Tìm hiểu về Raspberry Pi	19
2.2.1 Giới thiệu về Raspberry	19
2.2.2 Cài đặt một số thư viện cần thiết	20
2.3 Tìm hiểu về kit Tiva C Launchpad và các ngoại vi	21
2.3.1 Sử dụng UART	22

2.3.2	Sử dụng ADC.....	22
2.4	Sử dụng cảm biến DHT11 và ACS712.....	23
2.4.1	Cảm biến DHT11.....	23
2.4.2	Cảm biến dòng ACS712.....	24
2.5	Phương pháp checksum CRC16.....	25
2.6	Tìm hiểu về Thingsboard platform và MQTT.....	26
2.6.1	Giới thiệu về Thingsboard.....	26
2.6.2	Các ưu điểm của Thingsboard.....	27
2.6.3	Cấu trúc Thingsboard Server.....	28
2.6.4	Giới thiệu về MQTT.....	28
2.7	Cơ chế truyền dữ liệu từ các node slave đến node master.....	30
2.8	Cơ chế kiểm soát lỗi.....	30
2.9	Cơ chế đưa dữ liệu lên server và điều khiển từ server.....	31
3.	THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG.....	34
3.1	Yêu cầu thiết kế cụ thể.....	34
3.2	Lựa chọn thiết bị và chi tiết kỹ thuật cụ thể.....	34
3.3	Phân tích thiết kế.....	35
3.4	Sơ đồ khối tổng thể.....	36
3.5	Sơ đồ khối mạch nguyên lý chi tiết.....	37
4.	THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM.....	38
4.1	Yêu cầu đặt ra cho phần mềm.....	38
4.2	Nguyên lý hoạt động.....	38
4.3	Sơ đồ giải thuật:.....	38
5.	KẾT QUẢ THỰC HIỆN.....	44
5.1	Kết quả thi công mạch.....	44
6.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	47

6.1	Kết luận.....	47
6.2	Hướng phát triển.....	47
7.	<i>TÀI LIỆU THAM KHẢO</i>	48
8.	<i>PHỤ LỤC</i>	50

DANH SÁCH HÌNH MINH HỌA

Hình 2 1 Mạng không dây LoRa.....	13
Hình 2 2 So sánh băng thông và khoảng cách truyền các chuẩn không dây.....	14
Hình 2 3 Các loại module LoRa.....	15
Hình 2 4 Module LoRa E32-TTL-100	16
Hình 2 5 Thiết bị cấu hình module LoRa.....	17
Hình 2 6 Phần mềm cấu hình	19
Hình 2 7 Raspberry Pi 3.....	20
Hình 2 8 Kit Tiva C LaunchPad.....	21
Hình 2 9 Cảm biến DHT11	23
Hình 2 10 Khởi động DHT11	23
Hình 2 11 DHT11 chuẩn bị trả tín hiệu về.....	23
Hình 2 12 Sơ đồ timing tổng quát.....	24
Hình 2 13 Cảm biến dòng ACS712	24
Hình 2 14 Dòng điện AC	25
Hình 2 15 Mô hình Thingsboard.....	26
Hình 2 16 Cấu trúc server Thingsboard	28
Hình 2 17 Mô hình MQTT.....	29
Hình 2 18 Cơ chế lọc các rule ở server Thingsboard.....	31
Hình 2 19 Dữ liệu sau khi đưa server.....	32
Hình 2 20 Mô hình thực hiện RPC.....	32
Hình 2 21 Cài đặt thông số widget điều khiển.....	33
 Hình 3 1 Mô hình smart farming.....	 34
Hình 3 2 Module RF	35

Hình 3 3 Module LoRa E32-TTL-100	35
Hình 3 4 Sơ đồ khối tổng thể của hệ thống.....	36
Hình 3 5 Sơ đồ mạch tại Master.....	37
Hình 3 6 Sơ đồ mạch tại Slave.....	37
Hình 5 1 Hình ảnh thực tế tại mỗi slave.....	44
Hình 5 2 Hình ảnh thực tế tại master	44
Hình 5 3 Giao diện hoạt động của hệ thống.....	45
Hình 5 4 Giao diện cụ thể tại một slave	46

DANH SÁCH BẢNG SỐ LIỆU

Bảng 1 So sánh đặc điểm các chuẩn truyền không dây	10
Bảng 2 So sánh các loại vi điều khiển	11
Bảng 3 Đặc tính công nghệ LoRa	15
Bảng 4 Các chế độ hoạt động của module LoRa	18
Bảng 5 Khung dữ liệu gửi từ Slave	30
Bảng 6 Các chức năng thông qua số mã hóa	30
Bảng 7 Lựa chọn module RF tối ưu	36

1. GIỚI THIỆU

1.1 Tổng quan

Cuộc sống ngày càng hiện đại, cùng sự phát triển không ngừng của khoa học kỹ thuật dẫn đến nhu cầu về mọi vật kết nối với nhau, cung cấp thông tin cho người dùng, giúp cho cuộc sống giản đơn và tiện ích hơn.

Trong công nghiệp, một trong những vấn đề thiết yếu đặt ra là nhu cầu giám sát thiết bị máy móc, động cơ, bồn chứa đặt ở nhiều nơi khác nhau. Các nhân viên bảo trì, vận hành luôn muốn biết chính xác các thông số hoạt động của thiết bị, để đảm bảo nó luôn ổn định cũng như sớm nhận ra sự thay đổi để phát hiện các nguy cơ hỏng hóc tiềm ẩn.

Trong nông nghiệp, các yếu tố như nhiệt độ, độ ẩm không khí rất quan trọng. Dựa vào các yếu tố này mà người trồng có thể quyết định thời lượng tưới tiêu và bón phân sao cho hợp lí. Đối với những người có nhiều đất canh tác mà cách xa nhau thì việc giám sát các yếu tố môi trường cũng như điều khiển bật tắt máy bơm sẽ rất tốn thời gian di chuyển và phải nhớ luôn thời gian tắt máy. Để làm việc này trở nên tiện hơn, cần có một phương pháp hay công nghệ nào đó giúp người nông dân có thể quản lý, điều khiển mọi lúc, mọi nơi.

1.2 Tình hình nghiên cứu trong và ngoài nước:

Việc thu thập các dữ liệu, thông số cơ bản như nhiệt độ, độ ẩm, thông số các thiết bị trong nhà máy, nông nghiệp, gia đình... là nhu cầu tất yếu. Thực tế đã có nhiều hệ thống giám sát với các platform có sẵn như:

- Trong công nghiệp, dùng hệ thống cảm biến kết nối thiết bị PLC, giám sát qua SCADA. Ưu điểm: hoạt động tốt trong môi trường công nghiệp, dễ điều khiển qua hệ thống SCADA, ít bị nhiễu. Hạn chế: chi phí đầu tư cho PLC và phần mềm SCADA rất đắt tiền. Kích thước của các thiết bị PLC lớn, nhiều kết nối dây, khả năng kết nối internet vẫn còn hạn chế.
- Trong nông nghiệp: dùng hệ thống mạng cảm biến không dây nhiều slave gửi dữ liệu đến một master có kết nối internet để đưa dữ liệu lên server. Có nhiều loại module như Lora, Zigbee, ESP8266, BLE và nhiều platform hỗ trợ việc giám sát,

điều khiển như Thingsboard, PubNub, AWS IoT. Ưu điểm: các module kích thước nhỏ gọn, truyền được khoảng cách xa, ít tốn năng lượng, chi phí đầu tư phần cứng rẻ hơn PLC. Nhiều sản phẩm đã được hoàn thiện và bán ra thị trường như giải pháp Internet of Farm (Luda.Farm), Telit, Libelium-Sparkster Smart Agriculture Solution Kit....

1.3 Nhiệm vụ luận văn

1.3.1 Yêu cầu đề tài

Yêu cầu đặt ra của luận văn: xây dựng hệ thống giám sát môi trường và điều khiển thiết bị trong nông nghiệp. Hệ thống mạng không dây truyền nhận dữ liệu ở khoảng cách từ vài trăm cho đến vài kilomet, cụ thể là mạng hình sao, gồm một node master và 2 node slave. Master có thể nhận dữ liệu và gửi yêu cầu đến 2 slave nhưng 2 slave không thể giao tiếp với nhau. Mỗi slave có vi điều khiển kết nối với module không dây. Master đóng vai trò như một gateway, nhận dữ liệu từ 2 slave và đưa lên server. Từ yêu cầu đặt ra, ta cần phải thực hiện 2 lựa chọn:

- Lựa chọn module không dây
- Lựa chọn vi điều khiển ở slave

Lựa chọn module không dây

Technique	Bluetooth	Zigbee	GPRS	LoRa
Range	5-10m	10-75m	GSM system covered area	Up to 5km
Data rate	1Mbps (v1) 3Mbps (v2)	20Kbps to 250Kbps	Up to 114Kbps	2,4 Kbps
Area	Local area	Local area	Wire area	Local area
Point to point	Point to multi point	Point to multi point	Point to multi point	Point to multi point
Power profile	Relatively high	Low	Low	Low
Licensing	Not needed	Need in commercial purpose	Needs operator SIM card	Not need

Bảng 1 So sánh đặc điểm các chuẩn truyền không dây[11]

Lựa chọn vi điều khiển ở các node

Kit	Adruino Uno	8051	Tiva C Launchpad
Ưu điểm	Giá thành rẻ, thư viện hỗ trợ phong phú.	Giá thành rẻ	Ngoại vi đa dạng: 5 timer, 12 analog input, nhiều chế độ ngắt có hỗ trợ hệ điều hành thời gian thực, dễ debug lỗi
Hạn chế	Ngoại vi hạn chế 2 timer, 1 ADC	Không thể giao tiếp ADC, UART trực tiếp mà phải thông qua IC chuyển đổi	Giá cao hơn 2 vi điều khiển còn lại

Bảng 2 So sánh các loại vi điều khiển

Dựa vào yêu cầu đặt ra là giám sát và điều khiển từ xa với khoảng cách lên đến vài kilomet thì lựa chọn module LoRa là hợp lý nhất, còn phần vi điều khiển ở mỗi node thì chọn kit Tiva C. Tuy giá mắc hơn 2 vi điều khiển kia nhưng ngoại vi nhiều cũng như hiệu suất cao, sau này có thể dễ dàng phát triển thêm.

Còn đối với phần server điều khiển, chọn Raspberry Pi 3 (vừa là server vừa là gateway cho LoRa) vì có thể chạy trực tiếp server Thingsboard, đồng thời hỗ trợ thư viện MQTT viết bằng Python để truyền dữ liệu đến server.

1.3.2 Mục tiêu đề tài

- Thiết lập mạng Lora và khung truyền để trao đổi dữ liệu giữa các nơi với nhau, mạng hình sao: một master và hai slave. Chỉ có slave giao tiếp được với master, giữa các slave không thể giao tiếp với nhau
- Giám sát nhiệt độ, độ ẩm và dòng điện trên các thiết bị tại hai nơi, khoảng cách có thể lên đến 3km
- Xây dựng dashboard hiển thị các thông số dạng đồ thị, hiển thị cảnh báo và giao diện điều khiển thiết bị cho cả hai nơi
- Có khả năng hẹn giờ bật tắt máy bơm, đồng thời dựa vào dự báo thời tiết và dựa trên độ ẩm không khí thu được để phát hiện mưa.
- Cảnh báo đến smartphone khi có các dấu hiệu bất thường nằm ngoài điều kiện cho phép

1.3.3 Giới hạn đề tài

Trong giới hạn thời gian cho phép để hoàn thành luận văn, kết hợp với những kiến thức tích lũy được trong quá trình học tập, do đó em chỉ tập trung nghiên cứu về:

- Tạo một giao diện web để hiển thị dữ liệu và điều khiển thiết bị dựa trên các widget có sẵn của Thingsboard.
- Thiết lập Raspberry Pi như một gateway bằng ngôn ngữ Python để xác định dữ liệu đến từ node nào, sau đó đưa dữ liệu lên dashboard phù hợp.
- Sử dụng vi điều khiển Tiva C (làm trạm thu dữ liệu):
 - ✓ Đo nhiệt độ.
 - ✓ Đo độ ẩm.
 - ✓ Dòng điện tiêu thụ trên thiết bị.
- Thiết kế, thi công mô hình và hoàn thiện sản phẩm, chạy ổn định.
- Hoạt động của hệ thống:
 - ✓ Thu thập dữ liệu từ 2 trạm và gửi dữ liệu lên Raspberry Pi.
 - ✓ Vẽ đồ thị nhiệt độ, độ ẩm và hiển thị dữ liệu.
 - ✓ Thời gian cập nhật dữ liệu là mỗi 10 giây.
 - ✓ Lưu lại dữ liệu trên database
 - ✓ Có chế độ tự động và thủ công hẹn giờ bật tắt thiết bị
 - ✓ Cảnh báo và tắt thiết bị khi các yếu tố nằm ngoài điều kiện cho phép

2. LÝ THUYẾT

2.1 Tìm hiểu về công nghệ LoRa

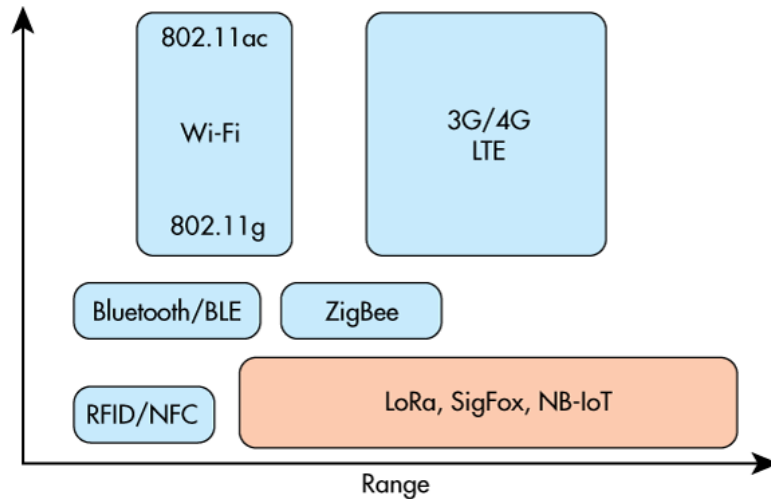
2.1.1 Giới thiệu



Hình 2 1 Mạng không dây LoRa [13]

Cảm biến không dây thường được dùng trong hệ sinh thái Internet of Things, bao gồm các giải pháp tốc độ độ cao như công nghệ Wifi hoặc mạng mesh như Zigbee. Khi nói đến công nghệ không dây, ta thường nghĩ đến các tiêu chuẩn như: công suất thấp, khoảng cách xa hoặc tốc độ truyền cao. Công suất thấp trong trường hợp này sẽ là một thiết bị có thể hoạt động liên tục trong nhiều tháng cho đến cả năm và khoảng cách truyền nhận dữ liệu lên đến hàng

kilometer. Chi phí thấp cũng là một điểm đáng lưu tâm. Công nghệ LoRa là giải pháp sáng giá đáp ứng được cả nhu cầu trên.



Hình 2.2 So sánh băng thông và khoảng cách truyền các chuẩn không dây [14]

LoRa Technology là công nghệ được phát triển bởi công ty SEMTECH, kết hợp 3 yếu tố: khoảng cách xa, công suất thấp và an toàn dữ liệu. LoRa viết tắt của Long Range Radio và chủ yếu dùng cho các mạng M2M và IoT. LoRa có thuật toán tối ưu data rate giúp tối đa thời gian dùng pin và dung lượng mạng tại các node. Giao thức LoRa bao gồm một số lớp khác nhau: mã hóa ở cấp độ mạng, ứng dụng và thiết bị để liên lạc an toàn.

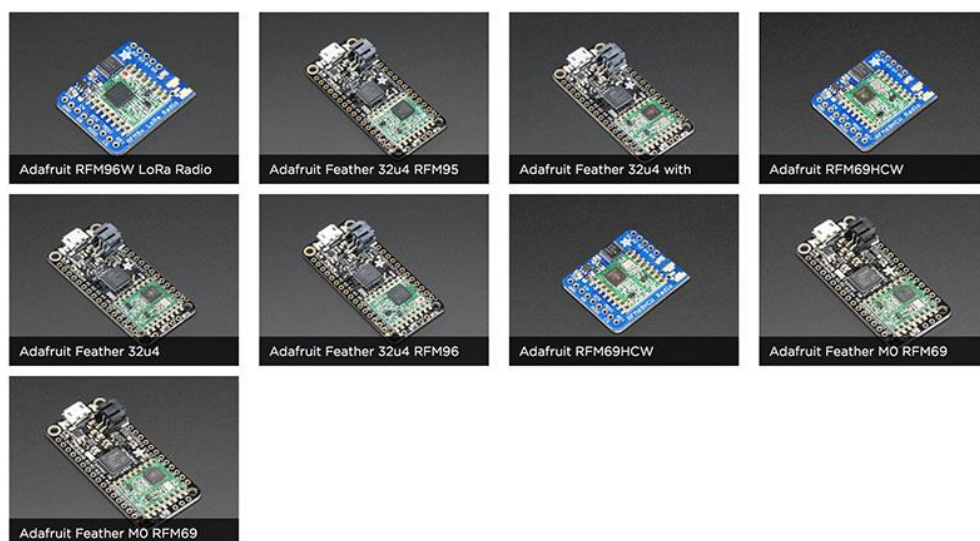
2.1.2 Đặc tính công nghệ LoRa

Specification	Lora Features
Range	2-5km in urban 15km suburban
Frequency	433/868/915 MHz
Standard	IEEE 802.15.5g
Modulation	Spread spectrum modulation type based on FM pulses which vary
Capacity	One LoRa gateway can take thousands of nodes
Battery	Long battery life
Lora Physical layer	Frequency, power, modulation and signaling between nodes and gateway

Bảng 3 Đặc tính công nghệ LoRa [12]

2.1.3 Các loại module LoRa và chuẩn giao tiếp với vi điều khiển

Công ty SEMTECH đi đầu trong lĩnh vực phát triển công nghệ không dây LoRa và đã giới thiệu ra thị trường một số module LoRa. Đặc biệt, dòng sản phẩm thu phát RF SX127x cho thị trường IoT / M2M. Các mô đun này hoạt động trong băng tần 860-1000 MHz và 137-960MHz. Adafruit gần đây đã giới thiệu một số bo mạch phát triển dựa trên các chip SEMTECH RFM69 và RFM95 ở nhiều tần số khác nhau như 433, 868 và 960 MHz.



Hình 2 3 Các loại module LoRa [15]

2.1.4 Giới thiệu module LoRa E32-TTL-100

Đây là module LoRa được dùng trong luận văn này, giao tiếp với vi điều khiển qua UART, khoảng cách truyền lí thuyết 3000m, dùng chip SX1278 của SEMTECH. Bên trong module có một chip xử lí và chip RF. Chip SX1278 trong module này có nhiều ưu điểm.



Hình 2 4 Module LoRa E32-TTL-100 [16]

Ultra low power: chip hỗ trợ WOR để giảm công suất tiêu thụ tổng thể. Ở chế độ tiết kiệm năng lượng mode 2, nó cân chỉnh công suất tiêu thụ tổng thể bằng cách thiết lập độ trễ khi nhận dữ liệu. Khi cấu hình độ trễ nhận là 2000ms thì dòng tiêu thụ chỉ 30uA.

Fixed-transmission (truyền cho một module cụ thể): module có thể kết nối với các module khác hoạt động khác kênh truyền và khác địa chỉ. Ví dụ, module A truyền AA BB CC tới module B (địa chỉ 0x0001, kênh truyền 0x80), ta thực hiện như sau: 00 01 80 AA BB CC, module B sẽ nhận được AA BB CC.

Broadcast transmission: đặt đại chỉ 0xFFFF, module có thể trao đổi dữ liệu với bất kì module nào trong cùng kênh truyền.

FEC (Forward Error Correction): khả năng mã hóa và sửa lỗi mạnh mẽ. Trong trường hợp bị can thiệp đột ngột, nó có thể sửa chính xác gói dữ liệu bị can thiệp, do

đó độ tin cậy và khả năng truyền xa được cải thiện tương ứng. Nếu không có FEC, gói dữ liệu sẽ bị mất.

Watchdog: một watchdog được tích hợp bên trong chip, khi xảy ra gián đoạn, nó có thể reset module trong 0.107s và hoạt động trở lại với các thông số cài đặt từ trước.

2.1.5 Thông số

- Băng thông hoạt động: 410 – 441MHz, mặc định 433MHz
- Nguồn cấp 2.3 – 5.5 V
- Định địa chỉ: 0000 – FFFF
- Kênh truyền: 0-23
- Tốc độ truyền trong không khí: 0.3, 1.2, 2.4, 4.8, 9.6, 19.2 kbps
- Dạng khung dữ liệu: 8N1, 8E1, 801
- Buffer bộ truyền: 512 bytes, mỗi khung có thể truyền tối đa 58 bytes
- Công suất truyền: 20dBm, dòng 110mA
- Dòng điện của module khi nhận dữ liệu: 14,5mA

Thiết bị cấu hình, cách hoạt động: Để nguyên 2 jumper, bộ thu tín hiệu từ các module khác. Gỡ 2 jumper, cài đặt thông số.



Hình 2 5 Thiết bị cấu hình module LoRa [16]

2.1.6 Các chế độ hoạt động

Mode	M1	M0	Chi tiết
Mode 0 (Nomal)	0	0	UART và kênh truyền không dây được mở. Bật chế độ truyền.
Mode 1 (Wake – up)	0	1	UART và kênh truyền không dây được mở. Điểm khác biệt so với mode 0 là module sẽ tự thêm vào kí tự trước chuỗi dữ liệu để khởi động bộ thu (khi nó ở chế độ tiết kiệm năng lượng)
Mode 2 (Power saving)	1	0	Vô hiệu hóa UART. UART chỉ bật và truyền khi nhận được dữ liệu không dây.
Mode 3 (Sleep)	1	1	Cài đặt thông số

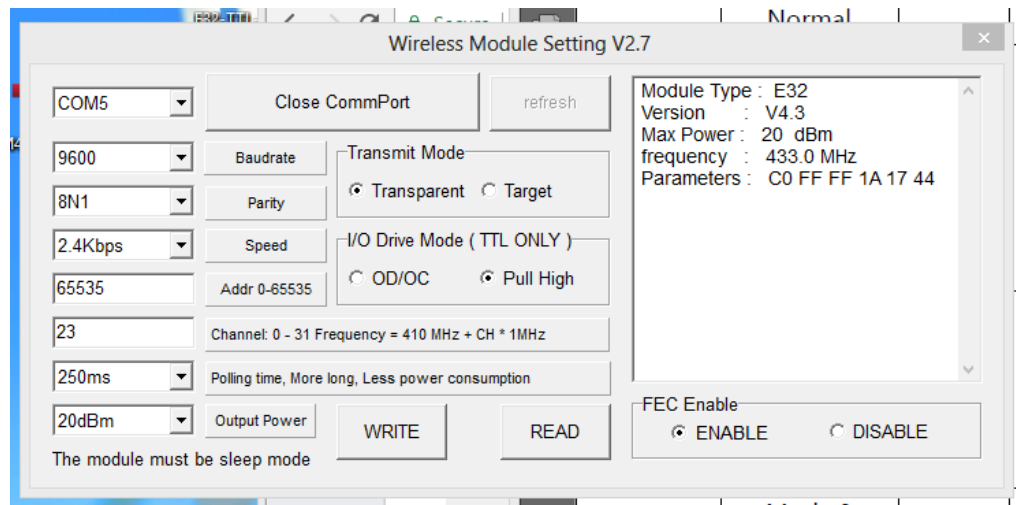
Bảng 4 Các chế độ hoạt động của module LoRa

2.1.7 Phần mềm cấu hình

Có 2 mode truyền:

- Transparent: truyền point to point
- Target: truyền khi có module trung gian làm bộ repeater.

Tốc độ truyền trong không khí (Speed) càng nhỏ, công suất truyền càng lớn thì truyền càng xa. Tần số hoạt động của module: 410 + kênh truyền, giới hạn kênh truyền từ 0-31.



Hình 2 6 Phần mềm cấu hình [16]

2.2 Tìm hiểu về Raspberry Pi

2.2.1 Giới thiệu về Raspberry

Raspberry Pi là một chiếc máy tính nhúng tí hon giá 35\$ chạy được nhiều hệ điều hành: Linux, Ubuntu, Window IoT. Nó được ra mắt vào tháng 2 năm 2012. Ban đầu Raspberry Pi được phát triển dựa trên ý tưởng tiến sĩ Eben Upton tại đại học Cambridge muốn tạo ra một chiếc máy tính giá rẻ để học sinh có thể dễ dàng tiếp cận và khám phá thế giới tin học. Dự định khiêm tốn của ông đến cuối đời là có thể bán được tổng cộng 1000 bo mạch cho các trường học. Nhưng nó đã thành công ngoài sức tưởng tượng khi đã bán được hơn một triệu bo mạch chỉ trong vòng chưa đầy một năm.



Hình 2 7 Raspberry Pi 3 [17]

Raspberry Pi có thể đáp được các yêu cầu cơ bản hằng ngày như lướt web, học lập trình, xem phim HD đến những ý tưởng không ngờ đến như điều khiển robot, nhà thông minh...

Cộng đồng Raspberry Pi phát triển rất mạnh: hầu hết những thắc mắc của người dùng đều được giải đáp rất nhanh và còn hơn thế nữa, người dùng có thể tìm thấy hàng ngàn dự án đã thực hiện và vô số ý tưởng độc đáo.

Với những ưu điểm độc đáo trên, Raspberry Pi đã vượt ra khỏi biên giới của trường học và trở thành thiết bị ưa thích của rất nhiều người đam mê điện tử và lập trình. Sự thành công của nó đã mở ra một bước phát triển mới cho tin học: đem máy tính và cảm hứng lập trình đến gần mọi người hơn bao giờ hết.

2.2.2 Cài đặt một số thư viện cần thiết

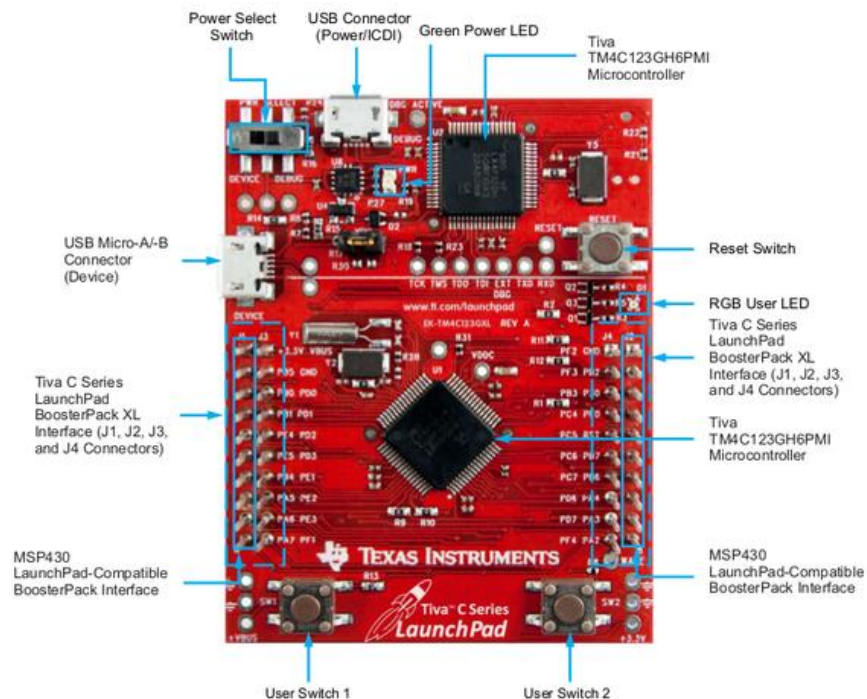
Để có thể thực hiện hoàn chỉnh luận văn, cần cài thêm một số thư viện và phần mềm hỗ trợ như:

- Thingsboard IoT platform: thingsboard.io/docs/user-guide/install/rpi/
- Thư viện serial, PyCRC, paho MQTT, python-forecast

Cấu hình UART trên raspberry:

- `sudo nano /boot/cmdline.txt` xóa dòng `"console=ttyAMA0,115200"`
- disable Serial Console
- `sudo nano /boot/config.txt` =>> thêm vào `"enable_uart=1"`
- `sudo reboot`

2.3 Tìm hiểu về kit Tiva C Launchpad và các ngoại vi



Hình 2 8 Kit Tiva C LaunchPad [18]

Kit Tiva C Launchpad dùng chip TM4C123GH6PM của hãng Texas Instrument, có hiệu năng cao và ngoại vi đa dạng:

- 80MHz 32-bit [ARM Cortex-M4-based microcontrollers](#) CPU
- 256KB Flash, 32KB SRAM, 2KB EEPROM

- Two Controller Area Network (CAN) modules
- USB 2.0 Host/Device/OTG + PHY
- Dual 12-bit 2MSPS ADCs, điều xung PWMs
- 8 UART, 6 I2C, 4 SPI

Các ngoại vi chính dùng trong luận văn: UART, GPIO, ADC, Timer, SysTick và các ngắt. Một số chân GPIO bị lock như: PC0-PC3, PD7, PF0, PE7. Để thực hiện hoàn chỉnh, có một số lưu ý về cấu hình và sử dụng ngoại vi của vi điều khiển.

2.3.1 Sử dụng UART

Để đảm bảo không bị mất dữ liệu từ module gửi đến qua UART, cần phải cài đặt ngắt nhận data và ưu tiên ngắt cao nhất (ở đây độ ưu tiên 0 là cao nhất). Các dữ liệu sau khi nhận được sẽ được đưa vào một ring buffer, từ buffer đó chương trình chính có thể lần lượt lấy ra từng dữ liệu và xử lý từng yêu cầu mà không bị thiếu hay mất dữ liệu khi gửi liệu gửi đến quá nhiều trong khoảng thời gian ngắn.

2.3.2 Sử dụng ADC

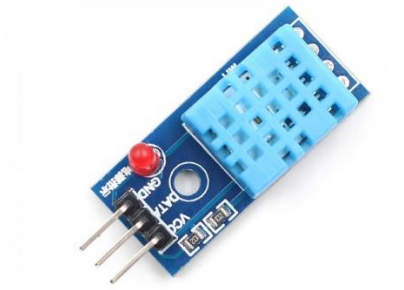
Việc đọc dữ liệu từ ADC không thật sự ổn định, các giá trị ADC bị sai lệch rất nhiều. Để hạn chế việc này, cần phải lấy mẫu ADC bằng phần cứng lần phần mềm bằng 2 phương pháp oversampling và averaging.

Oversampling: tập hợp thêm nhiều tín hiệu chuyển đổi đầu vào bằng cách dùng tần số lấy mẫu cao hơn tần số lấy mẫu Nyquist và khi kết hợp với kỹ thuật Averaging sẽ cải thiện ENOB. Điều này là có thể vì nó đồng thời cải thiện tín hiệu nhiễu mã hóa, nhiễu SNR (ảnh hưởng trực tiếp đến ENOB). Tần số Nyquist: $f_S = 2 * f_H$. Tần số lấy mẫu bằng phương pháp Oversampling: $f_{OS} = 4^x * f_S$ (x là hệ số cải thiện mong muốn trong ENOB).

Averaging: đóng vai trò như bộ lọc thông thấp đối với tín hiệu đầu vào, bằng cách lấy n mẫu cộng lại với nhau, rồi chia kết quả cho n sau đó kết hợp với Oversampling để lựa chọn kết quả.

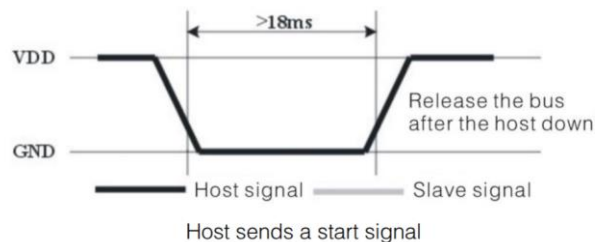
2.4 Sử dụng cảm biến DHT11 và ACS712

2.4.1 Cảm biến DHT11



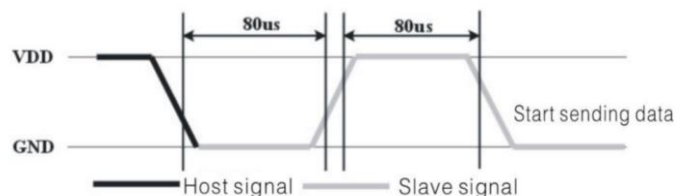
Hình 2 9 Cảm biến DHT11 [19]

DHT11 là cảm biến digital, dùng để xác định nhiệt độ và độ ẩm môi trường. Nó chỉ có một bus, bus này thực hiện chức năng vừa truyền vừa nhận. Để thực hiện lấy dữ liệu từ DHT11, ta phải thực hiện các bước sau: cho chân I/O Tiva ở chế độ output và xuất mức cao. Sau đó chuyển sang xuất mức thấp trong khoảng thời gian ít nhất là 18ms, rồi đưa chân I/O về chế độ input.



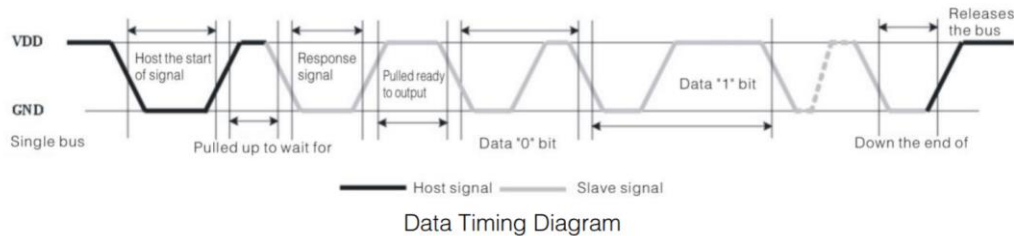
Hình 2 10 Khởi động DHT11 [19]

Sau khi DHT11 nhận được tín hiệu, nó xuất lại xung mức thấp 80us (báo hiệu đã nhận được yêu cầu từ Tiva), sau đó chuyển qua xung mức cao 80us (sẵn sàng gửi lại dữ liệu về cho Tiva).



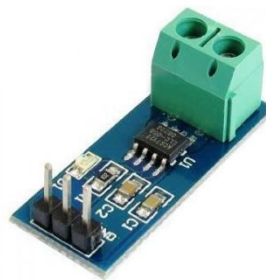
Hình 2 11 DHT11 chuẩn bị trả tín hiệu về [19]

Tiếp theo là chuỗi 40 bits của DHT11 gửi về, 8 bits và 8 bits thấp nhiệt độ, 8 bits cao và 8 bits thấp độ ẩm, và 8 bits cuối checksum. Xác định bit 0: 50us mức thấp, 26-28us mức cao. Xác định mức 1: 50us mức thấp, 70us mức cao.



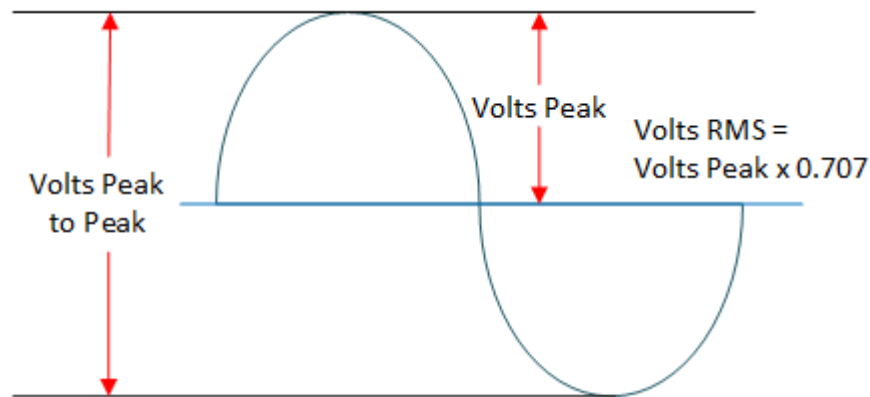
Hình 2 12 Sơ đồ timing tổng quát [19]

2.4.2 Cảm biến dòng ACS712



Hình 2 13 Cảm biến dòng ACS712 [20]

Cảm biến dòng ACS712 hoạt động dựa trên hiệu ứng Hall, dòng điện tối đa nó có thể đo được là 5A. Đối với dòng điện AC đưa vào, mức offset của chip là 2.5V, giá trị ngõ ra sẽ thay đổi trong khoảng 0-5V, với 2.5V là 0A. Chip trả về giá trị voltage ở đầu ra, dựa vào đó để xác định giá trị dòng điện hiệu dụng (RMS) và $I_{RMS} = 0.707 * I_{peak-peak}$.



Hình 2 14 Dòng điện AC [20]

Với cảm biến ACS712, ngõ ra là tín hiệu điện áp từ 0-5V. Để lấy được giá trị dòng điện hiệu dụng, ta cần thực hiện các bước sau:

- Tìm điện áp đỉnh-đỉnh, thực hiện bằng cách so sánh 1000 mẫu lấy về trong 1s. Để lưu giá trị ADC max và min, ta tạo hai biến $maxValue = 0$, $minValue = 4096$, $readValue$ là biến lưu giá trị ADC đọc về ở thời điểm hiện tại. Ta sẽ so sánh giá trị ADC đọc về với 2 biến trên, nếu $readValue > maxValue$, $maxValue = readValue$ và nếu $readValue < minValue$, $minValue = readValue$. Sau đúng 1000 lần thì dễ dàng xác định được giá trị đỉnh đỉnh của dòng điện AC, $I_{pp} = maxValue - minValue$.
- Chia dòng điện đỉnh đỉnh với 2, $I_p = I_{pp}/2$
- Nhân điện áp đỉnh với 0.707 để có giá trị hiệu dụng, $I_{RMS} = 0.707 * I_p$

Lưu ý: do điện áp ngõ ra của ACS712 là 0-5V, còn điện áp ngưỡng ADC của kit Tiva C là 3.3V, nên ở ngõ ra của ACS712 cần ghép thêm điện trở phân áp để scale điện áp xuống ngưỡng 0-3.3V.

2.5 Phương pháp checksum CRC16

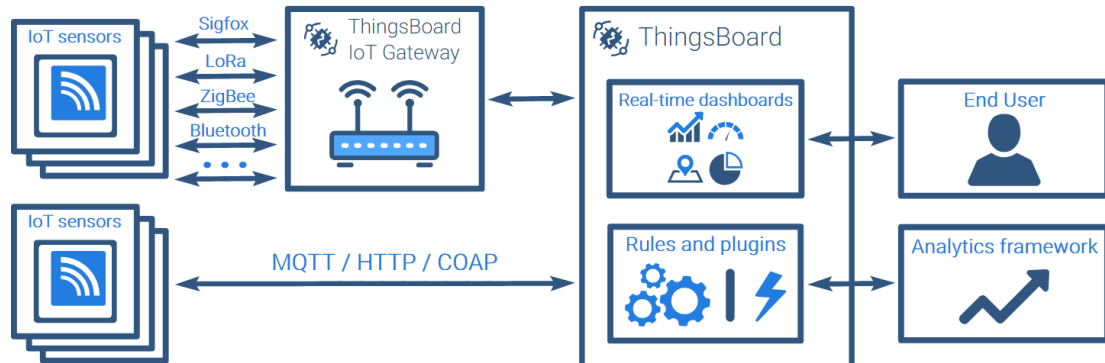
Cyclic Redundancy Check thường viết tắt là CRC, là phương pháp kiểm tra và phát hiện lỗi, được sử dụng trong các mạng số và thiết bị lưu trữ để phát hiện sự thay đổi tình cờ đối với dữ liệu được truyền đi hay lưu trữ.

CRC là một loại hàm băm dùng để phát sinh giá trị kiểm thử cho chuỗi bit, các gói tin vận chuyển qua mạng hay một khối nhỏ của tệp dữ liệu. Giá trị của CRC được tính toán và đính kèm vào dữ liệu trước khi dữ liệu được truyền đi hay lưu trữ. Khi dữ liệu được sử dụng, nó sẽ được kiểm thử bằng cách sinh ra mã CRC và so khớp với mã CRC trong dữ liệu. CRC rất phổ biến, vì nó rất đơn giản để thêm vào trong các máy tính sử dụng hệ cơ sở nhị phân, dễ dàng phân tích tính đúng, và rất phù hợp để dò các lỗi gây ra bởi nhiễu trong khi truyền dữ liệu [10].

Đa thức chia trong CRC16: $P(x) = x^{16} + x^{15} + x^2 + 1$. Các bytes sẽ được chia cho đa thức $P(x)$, phần dư của phép chia là CRC và được chèn vào byte data. Phía nhận (gồm CRC đã tính) sẽ kiểm lại CRC cũng bằng cách chia cho đa thức $P(x)$, nếu kết quả là 0, thì truyền thành công, ngược lại đã xảy ra lỗi trong quá trình truyền.

2.6 Tìm hiểu về Thingsboard platform và MQTT

2.6.1 Giới thiệu về Thingsboard



Hình 2 15 Mô hình Thingsboard [21]

ThingsBoard là một IoT platform mã nguồn mở cho phép dễ dàng tùy biến, quản lý và mở rộng các dự án IoT. Với ThingsBoard, bạn có thể:

- Kết nối và điều khiển thiết bị.
- Thu thập và trực quan hóa dữ liệu từ các thiết bị.
- Phân tích dữ liệu thiết bị và kích hoạt cảnh báo.
- Cung cấp dữ liệu thiết bị cho các hệ thống khác.
- Có thể tạo ra các tính năng cụ thể bằng cách thiết lập các rules và plugin.

Thingsboard hoạt động theo mô hình server-client. Mỗi client sẽ đc cấp một access token để kết nối với server. Platform này hỗ trợ nhiều phương thức kết client kết nối với server như: MQTT, CoAP, HTTP.

Thingsboard cung cấp các widget cho phép xây dựng dashboard một cách trực quan, sinh động và dễ dàng tùy biến theo ý mình, như điều khiển, vẽ đồ thị đối với dữ liệu thời gian thực.

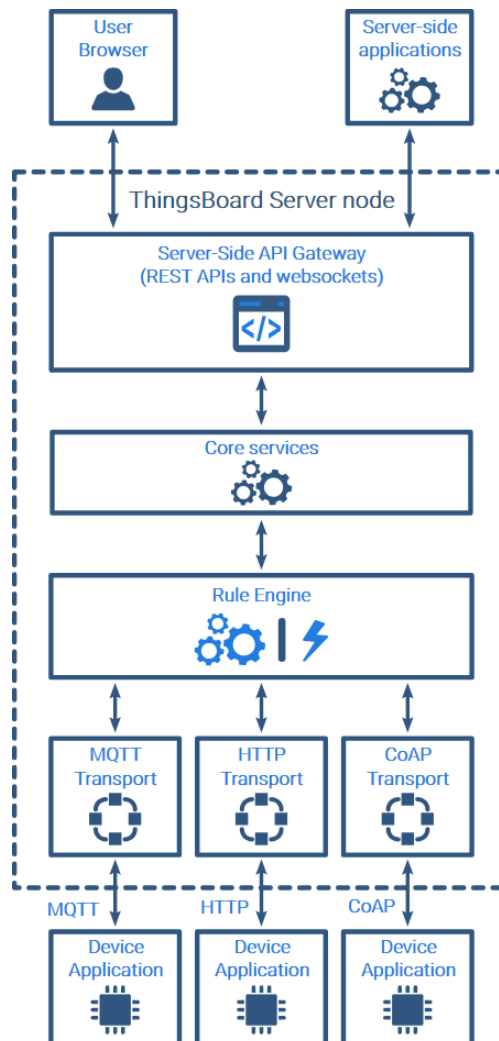
Ngoài ra, nó còn hỗ trợ nhiều plugin khác như: cảnh báo Telegram Bot, gửi mail đến người dùng qua SMTP, phân tích dữ liệu với Apache Spark.

2.6.2 Các ưu điểm của Thingsboard

Thingsboard có nhiều ưu điểm như:

- **Scalable (có thể mở rộng):** có thể mở rộng và thêm nhiều plugin dựa vào các open source.
- **fault-tolerant (khả năng chịu đựng lỗi):** mỗi node trong cluster đều như nhau.
- **robust and efficient (mạnh mẽ và hiệu quả):** một server có thể xử lý từ hàng chục đến hàng trăm thiết bị tùy theo tình huống xử dụng. ThingsBoard cluster có thể xử lý hàng triệu thiết bị.
- **customizable (tùy chỉnh):** cho phép tùy chỉnh giao diện cũng như thêm vào các luật để lọc dữ liệu.
- **durable (ổn định):** không bị mất dữ liệu.

2.6.3 Cấu trúc Thingsboard Server

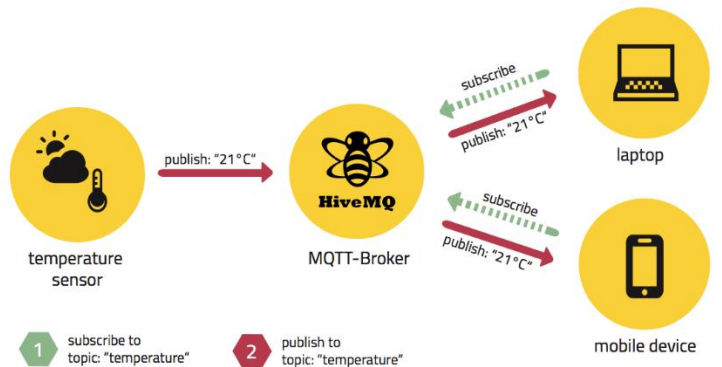


Hình 2 16 Cấu trúc server Thingsboard [21]

2.6.4 Giới thiệu về MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền dữ liệu mã nguồn mở dựa trên giao thức TCP/IP, phổ biến trong Internet of Things. MQTT hỗ trợ rất nhiều ngôn ngữ lập trình như: Java, Python, GoLang, C/C++, Rust, C#, embedded C/C++. Nhiều ứng dụng dùng MQTT như Facebook Messenger,

Amazon Web Services, Microsoft Azure, McAfee OpenDXL, Pimatic home automation...



Hình 2 17 Mô hình MQTT [22]

MQTT hoạt động theo mô hình broker-client. Message trong MQTT gồm: topic + payload. Client có thể publish một message hoặc subscribe một topic từ client khác. Các message sau khi đã được publish sẽ đến broker, broker tiến hành kiểm tra các client nào đã subscribe topic này, sau đó sẽ gửi message đến đúng client đã subscribe topic đó. Để đảm bảo message luôn được truyền đi cũng như tính ổn định, các message được chia làm 3 QoS (Quality of Server): 0,1,2.

- At most once (0): broker/client sẽ gửi dữ liệu đúng 1 lần, quá trình gửi được xác nhận chỉ bởi giao thức TCP/IP.
- At least once (1): broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
- Exactly once (2): broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay.

Ví dụ, publish và subscribe một message bằng ngôn ngữ python:

```

client.publish('v1/devices/me/attributes', json.dumps({"Device 1": "Not Connect", "Device 2": "Not Connect"}))

client.subscribe('v1/devices/me/rpc/request/+')
    
```

2.7 Cơ chế truyền dữ liệu từ các node slave đến node master

Dữ liệu gửi đi từ các node slave phải theo khung dữ liệu chuẩn MODBUS, giúp phân tách dữ liệu dễ và kiểm tra tính chính xác khung dữ liệu. Mỗi lần node gửi dữ liệu về slave sẽ gồm 6 bytes:

Địa chỉ của node gửi	Chức năng muốn gửi	Byte cao Data	Byte thấp Data	CRC16 byte thấp	CRC16 byte cao
----------------------	--------------------	---------------	----------------	-----------------	----------------

Bảng 5 Khung dữ liệu gửi từ Slave

Địa chỉ của hai node: 0x10 và 0x30. Địa chỉ master là 0xFF. Mỗi địa chỉ phải là duy nhất.

Chức năng:

0x21	Bật tắt đèn 1
0x22	Bật tắt đèn 2
0x23	Dòng điện AC qua đèn 1
0x24	Nhiệt độ
0x25	Độ ẩm
0x26	Dòng điện AC qua đèn 2
0x27	Duy trì kết nối với master, sau mỗi 5 phút
0x28	Kết nối lần đầu
0x29	Báo lại với node gửi, dữ liệu gửi bị sai

Bảng 6 Các chức năng thông qua số mã hóa

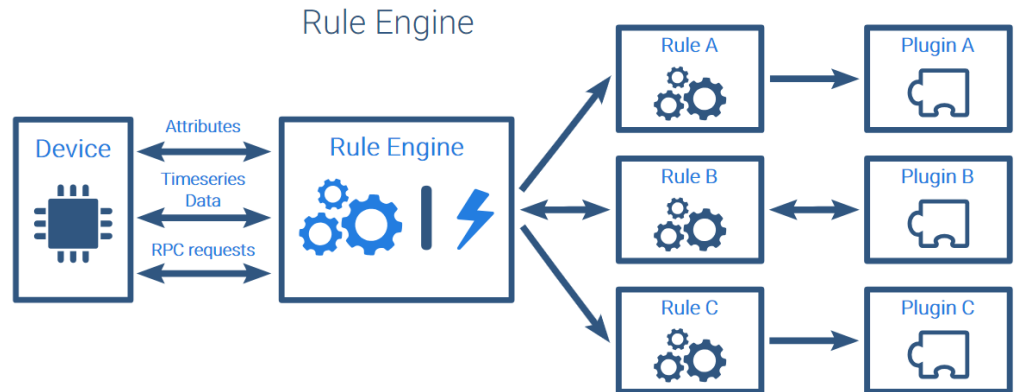
Do dữ liệu từ master gửi đến slave chỉ là các lệnh yêu cầu kết nối và bật tắt đèn, nên khung dữ liệu chỉ gồm 5 bytes vì chỉ có 1byte data.

Bên nhận dữ liệu, đầu tiên sẽ tiến hành checksum để kiểm tra data có bị mất hay thiếu sót không. Nếu kết quả CRC = 0, data truyền đúng và sẽ lọc các thành phần trong khung dữ liệu và tiến hành xử lý. Nếu kết quả CRC != 0, khung dữ liệu bị sai, bên nhận sẽ gửi lệnh yêu cầu gửi lại.

2.8 Cơ chế kiểm soát lỗi

Điều kiện cho phép của môi trường ở Việt Nam: nhiệt độ ở mức 18-40 độ, độ ẩm ở mức từ 50-85%, trên 95% nếu chuẩn bị mưa. Đối với dòng điện, tùy theo thiết

bị gắn vào relay mà có dòng điện cho phép khác nhau. Dựa vào điều kiện này ta thiết lập các rules ở server Thingsboard và cơ chế kiểm tra lỗi ở master.

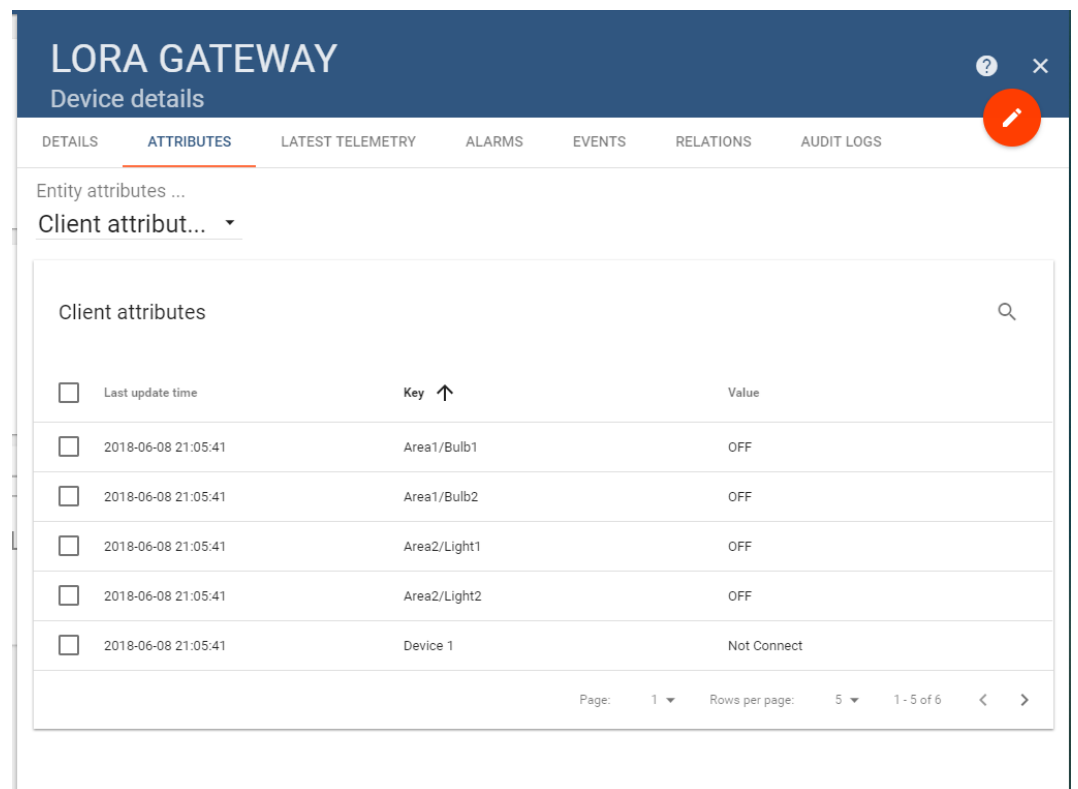


Hình 2 18 Cơ chế lọc các rule ở server Thingsboard [21]

Cơ chế kiểm soát lỗi ở master: mỗi lần dữ liệu nhiệt độ hoặc độ ẩm nằm ngoài ngưỡng cho phép, bộ đếm lỗi sẽ tăng lên 1, nếu dữ liệu tiếp theo nằm ở ngưỡng cho phép thì bộ đếm lỗi giảm 1, giảm tới 0 thì dừng. Bộ đếm đạt giá trị 5 sẽ gửi cảnh báo đến người dùng.

2.9 Cơ chế đưa dữ liệu lên server và điều khiển từ server

Đưa dữ liệu lên server: dựa vào dạng khung dữ liệu từ slave, master tách ra thành 2 luồng dữ liệu: từ slave 1 và slave 2. Dữ liệu từ mỗi slave được mã hóa dạng JSON, có dạng như sau `{"data": "value"}` và đưa lên server thông qua MQTT với câu lệnh: `client.publish('v1/devices/me/telemetry', json.dumps({"data": "value"}))`. Server sẽ lọc từng kiểu data tương ứng để hiển thị.



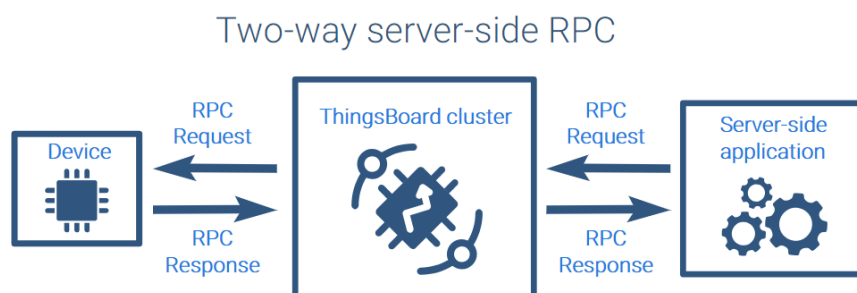
The screenshot shows the 'LORA GATEWAY' interface with the 'Device details' tab selected. Under 'Entity attributes', 'Client attributes' is chosen. A table lists attributes with checkboxes, timestamps, keys, and values.

<input type="checkbox"/>	Last update time	Key ↑	Value
<input type="checkbox"/>	2018-06-08 21:05:41	Area1/Bulb1	OFF
<input type="checkbox"/>	2018-06-08 21:05:41	Area1/Bulb2	OFF
<input type="checkbox"/>	2018-06-08 21:05:41	Area2/Light1	OFF
<input type="checkbox"/>	2018-06-08 21:05:41	Area2/Light2	OFF
<input type="checkbox"/>	2018-06-08 21:05:41	Device 1	Not Connect

Page: 1 Rows per page: 5 1 - 5 of 6

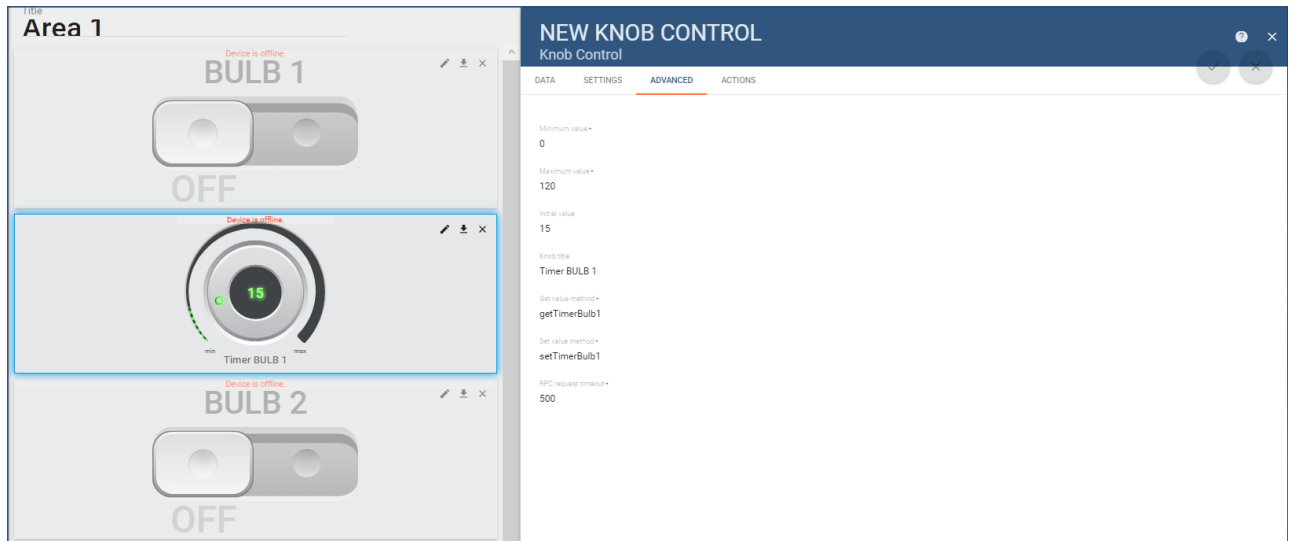
Hình 2 19 Dữ liệu sau khi đưa server

Thingsboard cho phép gửi các lệnh RPC (remote procedure calls) từ phía server đến thiết bị và ngược lại. Về cơ bản, tính năng này cho phép bạn gửi lệnh tới thiết bị và nhận kết quả thực hiện lệnh. Tương tự, bạn có thể thực hiện yêu cầu từ thiết bị, áp dụng một số tính toán hoặc logic phía máy chủ khác và phản hồi trở lại thiết bị.



Hình 2 20 Mô hình thực hiện RPC [21]

Cài đặt một giao diện điều khiển trên Dashboard



Hình 2 21 Cài đặt thông số widget điều khiển

Sau khi thực hiện điều chỉnh các thông số trên dashboard, server sẽ gửi các thông số này xuống cho client (ở đây là master-raspberry). Client sẽ kiểm tra các lệnh này trong hàm `on_message`, sau đó gửi khung data tương ứng đến slave.

Ví dụ, kiểm tra lệnh yêu cầu bật đèn 1 ở khu vực 1 như sau:

```
if data['method'] == 'setArea1Bulb1':

    if data['params'] == True:

        print ("Area 1 Bulb 1 ON")

        #send data via uart to lora module,

        #then lora send data to

        #slave to turn on led

        node_crc(0x10,0x21,0x11)

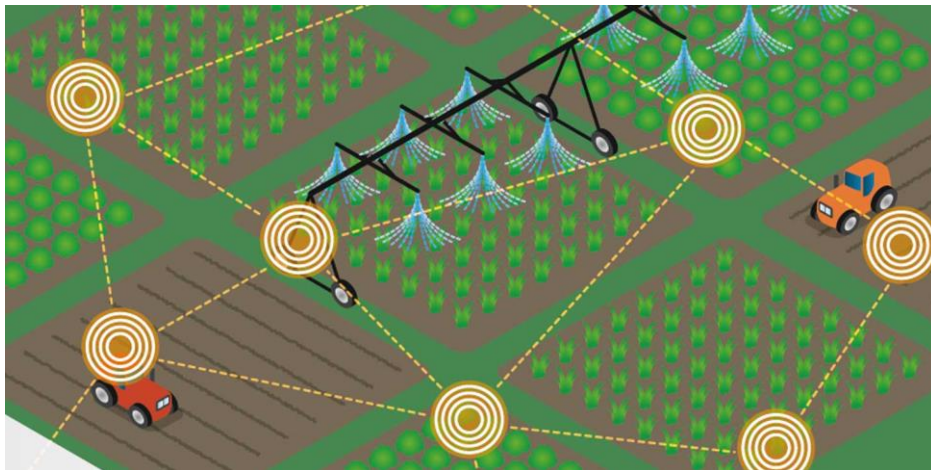
        bulb1 = True

        client.publish('v1/devices/me/rpc/response/'+requestId,
            json.dumps(bulb1), 1)
```

3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

3.1 Yêu cầu thiết kế cụ thể

Hệ thống sẽ gồm có 2 trạm thu thập dữ liệu và 1 trung tâm ghi nhận dữ liệu. Mỗi trạm sẽ thu thập các thông số từ cảm biến như: nhiệt độ, độ ẩm và dòng điện trên thiết bị.



Hình 3 1 Mô hình smart farming [23]

Trạm điều khiển và trạm lấy mẫu phân biệt nhau bằng cách định địa chỉ khác nhau, để có thể dễ dàng thêm nhiều trạm trong tương lai. Hệ thống sẽ truyền dữ liệu không dây giữa các trạm với trạm trung tâm thông qua module LoRa.

Trạm điều khiển sẽ hiển thị các thông tin dưới dạng đồ thị, điều khiển thiết bị ở các node, gửi cảnh báo khi các thông số nằm ngoài điều kiện cho phép.

3.2 Lựa chọn thiết bị và chi tiết kĩ thuật cụ thể

- Hai trạm thu thập dữ liệu, gồm các thiết bị và module như sau:
 - Kit Tiva C LaunchPad, đóng vai trò là trung tâm xử lí.
 - Relay 5V có opto cách ly
 - Các module cảm biến: nhiệt độ và độ ẩm DHT11, cảm biến dòng ACS712 và module Lora SX1278

➤ Tại trung tâm:

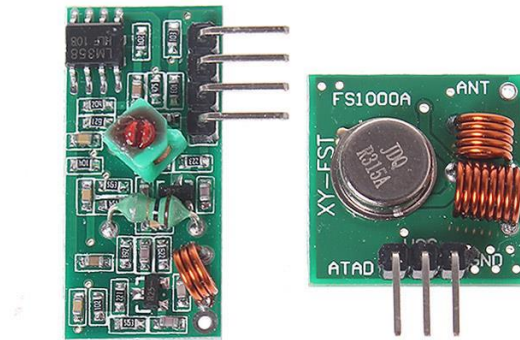
- Raspberry Pi 3 nhận dữ liệu từ module LoRa SX1278 UART 3000m qua UART, sau đó kiểm tra, so sánh với ngưỡng đặt ra rồi gửi lên server.
- Nguồn cấp Raspberry Pi 3 là nguồn 5.1V – 2.5A, nguồn cấp cho module LoRa từ nguồn 5V trên GPIO của Raspberry.

3.3 Phân tích thiết kế

Lựa chọn module RF truyền nhận tầm xa:

Yêu cầu đặt ra cho module RF truyền nhận là có thể truyền nhận từ xa, có thể sửa lỗi và ít tiêu tốn năng lượng.

- Phương án 1: Dùng module phát sóng RF 433MHz



Hình 3 2 Module RF

- Phương án 2: Dùng module truyền LoRa SX1278 UART



Hình 3 3 Module LoRa E32-TTI-100 [16]

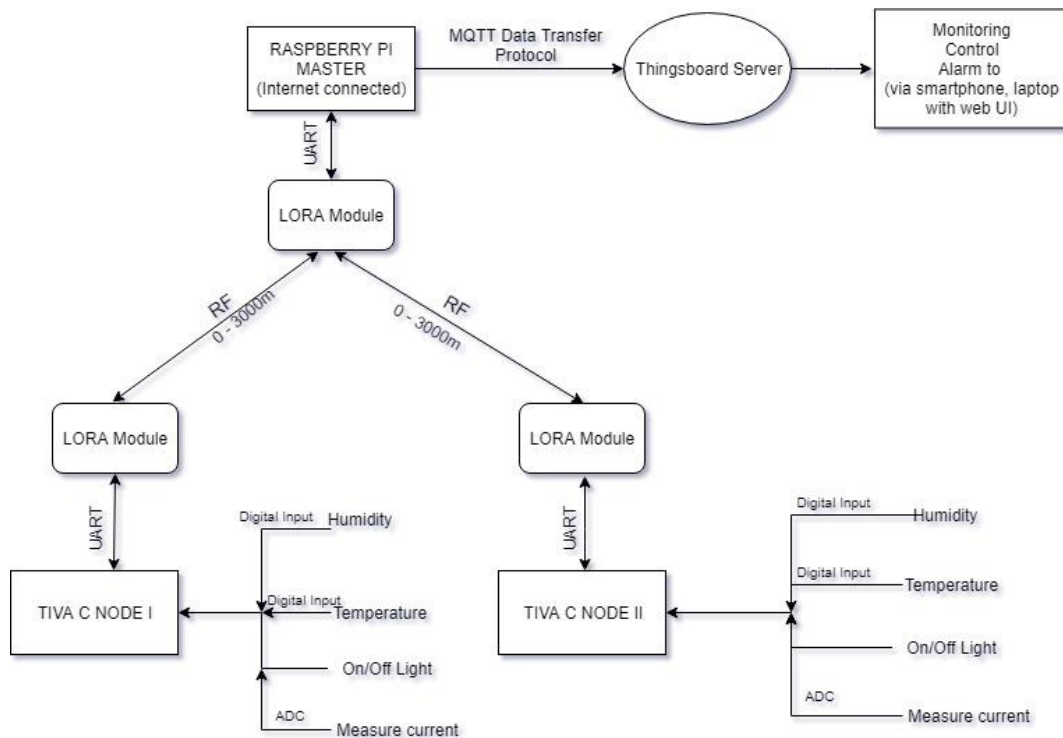
Phân tích ưu điểm và nhược điểm của các module truyền RF

	Phương án 1	Phương án 2
Ưu điểm	Module RF 433Mhz có giá tương đối rẻ	Có nhiều dạng cấu hình, giao tiếp qua UART đơn giản, có cơ chế sửa lỗi FEC. Có thể hoạt động ở chế độ repeater hoặc broadcast. Ít tiêu tốn năng lượng
Nhược điểm	Cần phải tạo thư viện truyền nhận, không có cơ chế tự sửa lỗi. Phải cần đến cả hai module truyền và nhận	Giá mắc

Bảng 7 Lựa chọn module RF tối ưu

Từ ưu và nhược điểm trên cho thấy ta ưu tiên dùng module LoRa.

3.4 Sơ đồ khối tổng thể



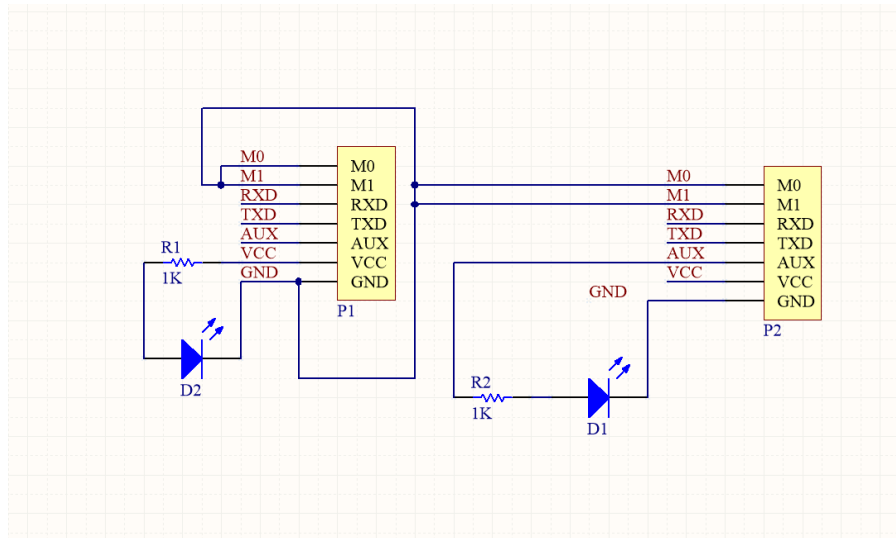
Hình 3 4 Sơ đồ khối tổng thể của hệ thống

Hệ thống gồm 1 module LoRa làm Master tại Raspberry Pi, hoạt động ở chế độ broadcast (gửi dữ liệu cho mọi node có địa chỉ khác nhưng cùng kênh truyền) và 2 module LoRa hoạt động chế độ point-to-point. Tiva C thu thập và gửi dữ liệu đến

Raspberry, nó sẽ phân tích các gói tin để biết dữ liệu đến từ đâu, sau đó đưa lên server.

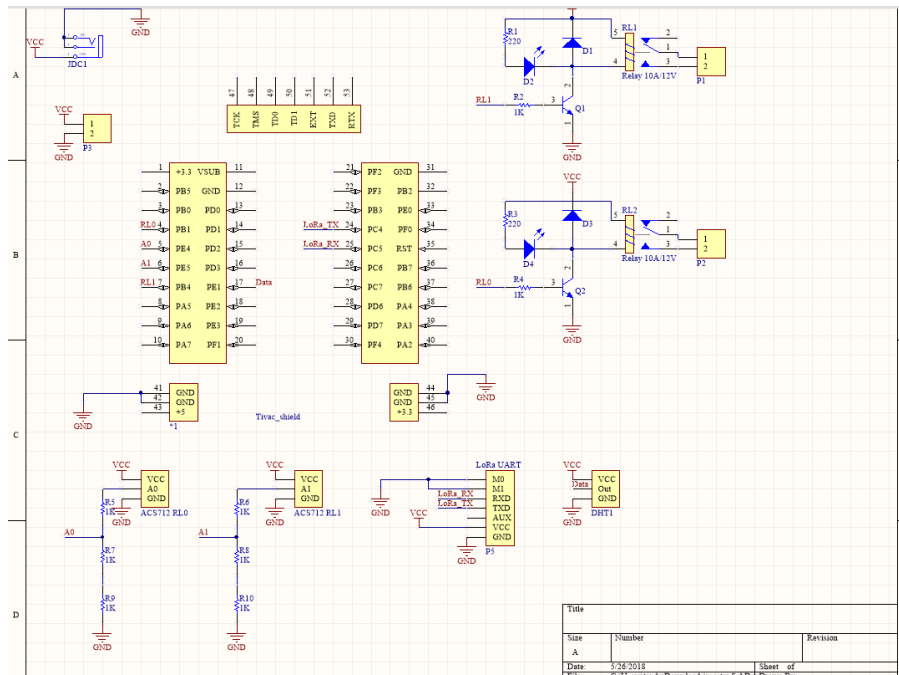
3.5 Sơ đồ khối mạch nguyên lý chi tiết

Sơ đồ mạch tại Master:



Hình 3 5 Sơ đồ mạch tại Master

Sơ đồ mạch tại Slave



Hình 3 6 Sơ đồ mạch tại Slave

4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

4.1 Yêu cầu đặt ra cho phần mềm

Hệ thống phải hoạt động chuẩn xác, có chế độ điều khiển tự động và tử công, chế độ hẹn giờ, dựa vào dự báo thời tiết và độ ẩm của cảm biến để quyết định có nên tưới cây hay không và gửi cảnh báo đến smartphone người dùng.

4.2 Nguyên lý hoạt động

Sau khi được cấp nguồn, kit Tiva C sẽ gửi yêu cầu kết nối đến Raspberry mỗi 3 giây, đồng thời nhấp nháy led đỏ báo hiệu đang cố gắng kết nối. Nếu Raspberry nhận được yêu cầu kết nối, nó sẽ gửi lệnh xác nhận kết nối. Sau khi đã kit Tiva đã được kết nối, nó vào trạng thái thu thập dữ liệu từ các cảm biến và nhận lệnh điều khiển từ Tiva.

Sau khi đã kết nối thành công, mỗi 5 phút Raspberry sẽ gửi tín hiệu kiểm tra kết nối đến Tiva. Nếu gửi tín hiệu mà không nhận được trả lời, Raspberry sẽ thông báo mất kết nối. Đối với Tiva, nếu không nhận được yêu cầu duy trì kết nối từ Raspberry, nó quay lại trạng thái kết nối như lúc mới cấp nguồn.

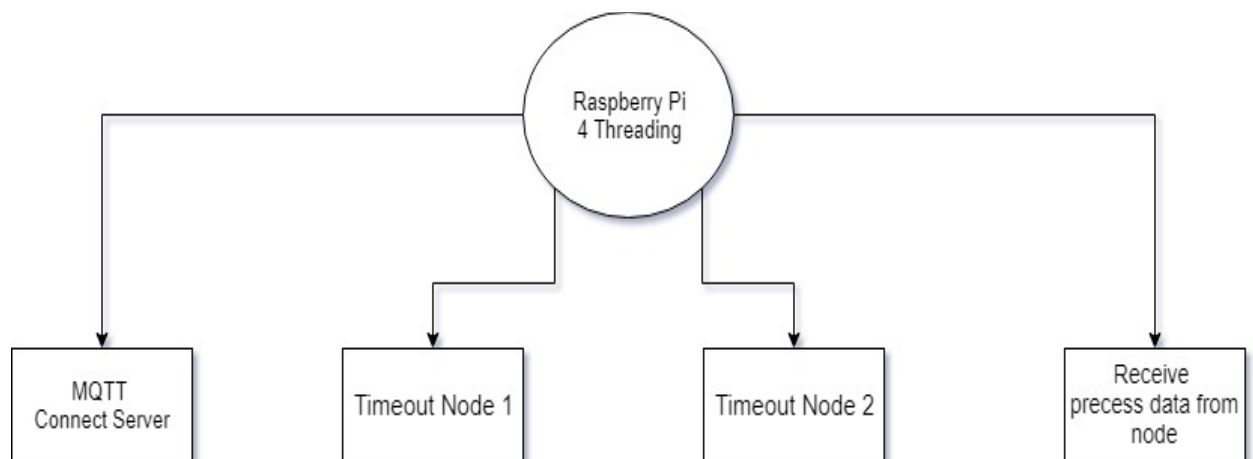
Raspberry sau khi đã nhận được dữ liệu từ các Tiva, sẽ phân tích theo khu vực, chuyển thành dạng JSON rồi mới đưa lên server thông qua MQTT. Server sẽ chạy các đồ thị để có cái nhìn trực quan nhất có thể.

4.3 Sơ đồ giải thuật:

❖ Tại Raspberry:

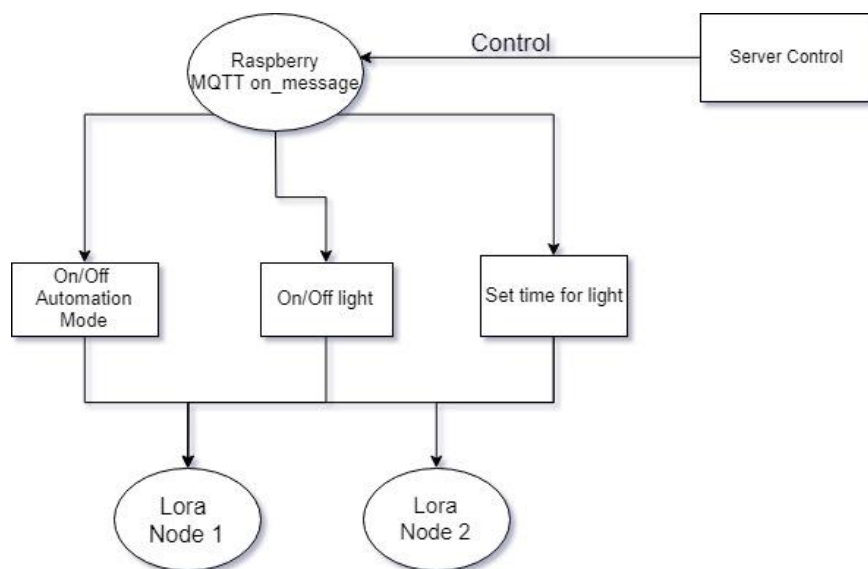
Chúng ta sẽ sử dụng 4 thread, mỗi thread sẽ làm những chức năng riêng:

- Duy trì kết nối với server và nhận lệnh điều khiển từ server
- Nhận và xử lý data từ các 2 node gửi đến, sau đó đưa lên dashboard
- Kiểm tra kết nối node 1
- Kiểm tra kết nối node 2



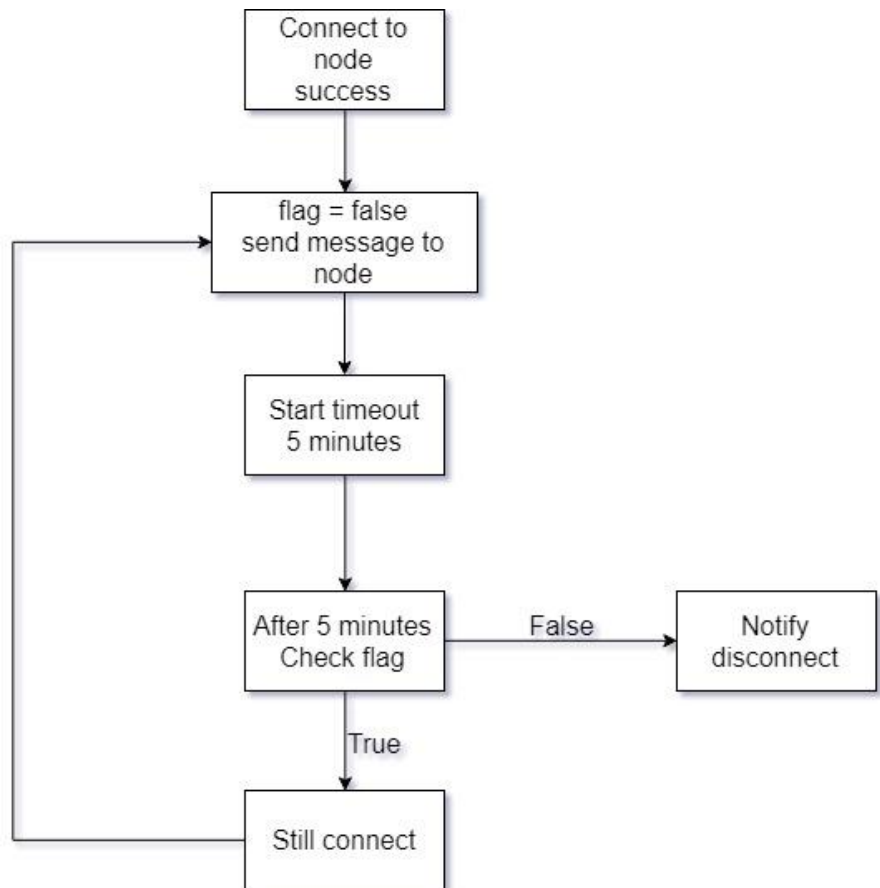
Hình 4 1 Các task thực hiện tại Raspberry

Cụ thể từng task, task nhận lệnh điều khiển từ server sẽ kiểm tra các thông số trong hàm on_connect của MQTT, sau khi kiểm tra phù hợp Raspberry sẽ gửi dữ liệu đến slave tương ứng.



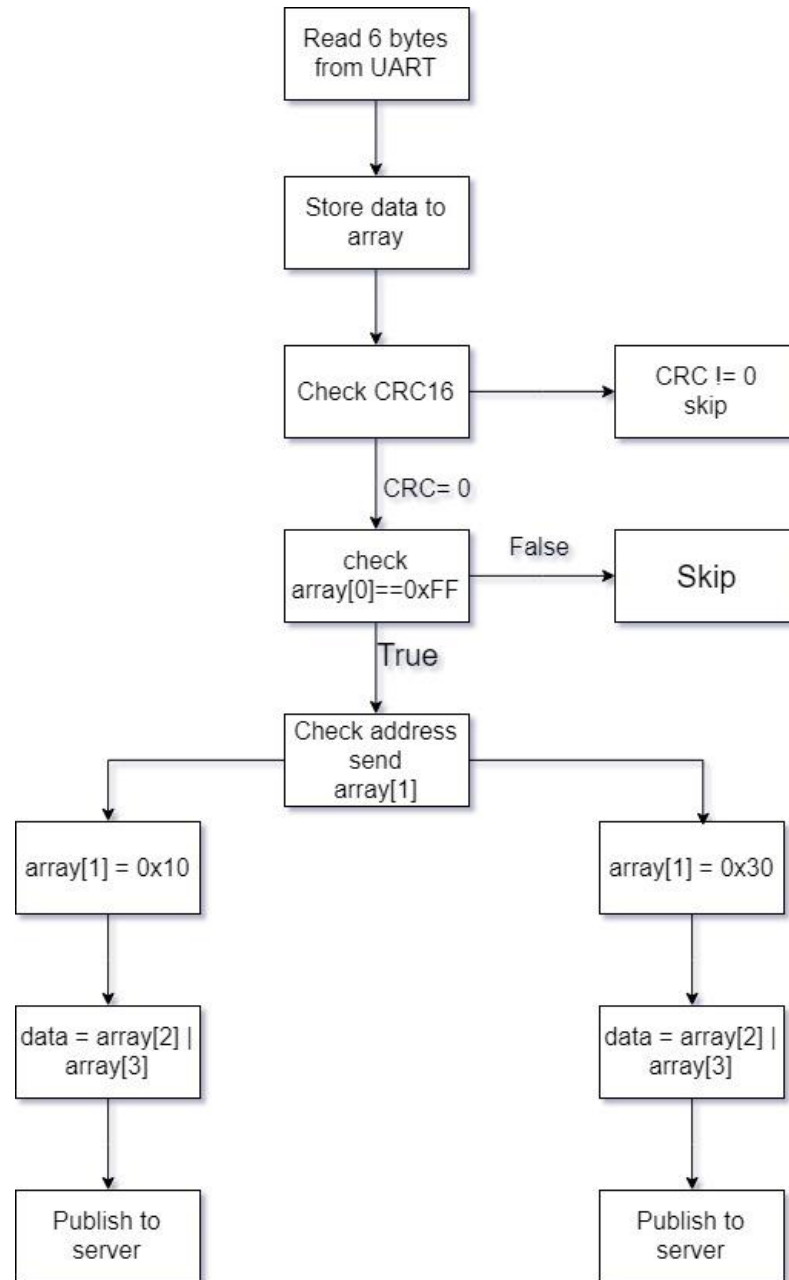
Hình 4 2 Kết nối với server thông qua giao thức MQTT

Task kiểm tra kết nối giữa Tiva C và Raspberry, sau khi kết nối thành công. Mỗi 5 phút Raspberry sẽ gửi một lệnh kiểm tra kết nối, sau 5 phút nó sẽ kiểm tra lại. Nếu có lệnh trả về từ Tiva, cho thấy thiết bị vẫn còn kết nối, ngược lại sẽ thông báo với người dùng lỗi mất kết nối.



Hình 4 3 Duy trì kết nối giữa Raspberry và Tiva

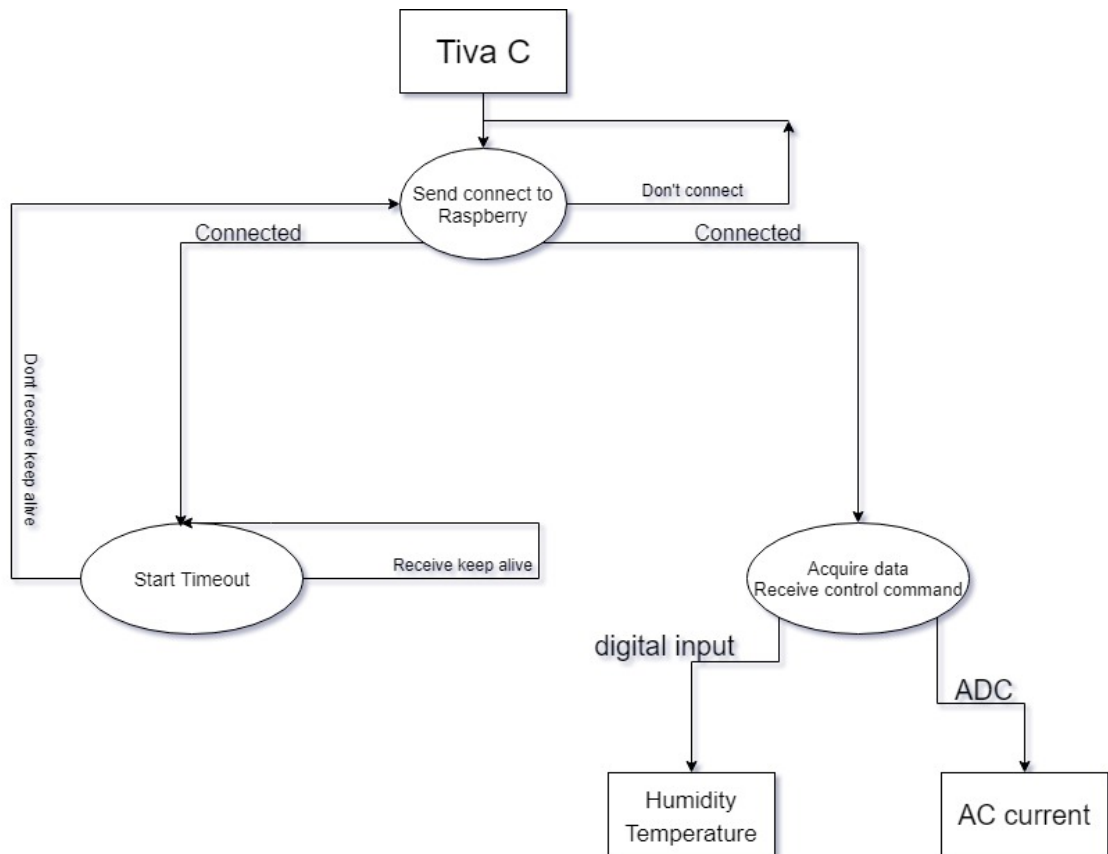
Task kiểm tra dữ liệu nhận được từ Tiva, Raspberry đọc 6 bytes UART một lần, sau đó chuyển data qua dạng hex và lưu vào một mảng mới. Kiểm tra CRC, nếu CRC =0, và byte đầu của mảng có địa chỉ 0xFF mới tiến hành lọc dữ liệu, ngược lại giá trị CRC và địa chỉ không đúng thì bỏ qua.



Hình 4 4 Giải thuật xử lý dữ liệu ở Raspberry

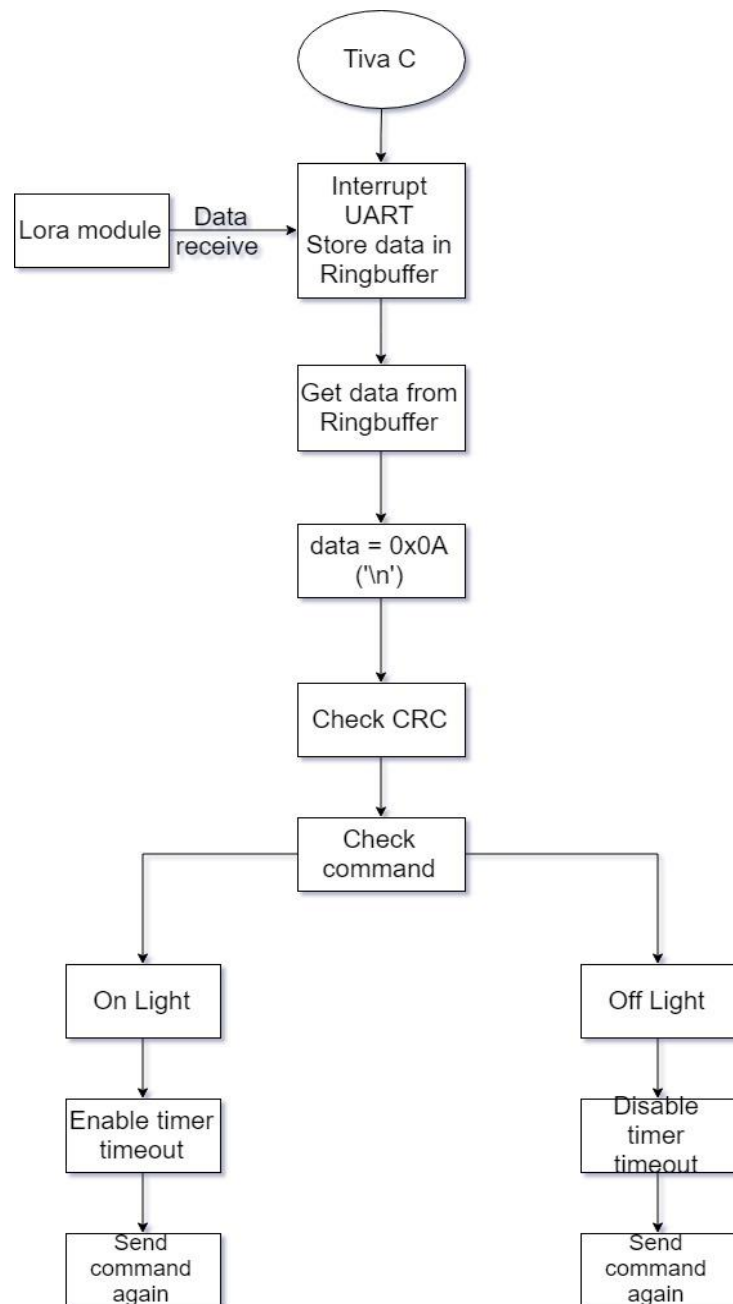
❖ Sơ đồ giải thuật tại Tiva

Conect và gửi dữ liệu đến Raspberry: khi Tiva được cấp nguồn, sẽ gửi kết nối đến Raspberry, việc gửi này sẽ lặp đi lặp lại liên tục cho đến khi kết nối thành công. Sau khi kết nối thành công, nó bắt đầu thu tập dữ liệu và mặt khác bắt đầu bật chế độ timeout, để kiểm tra duy trì kết nối với Raspberry.



Hình 4 5 Kết nối và duy trì kết nối ở Raspberry

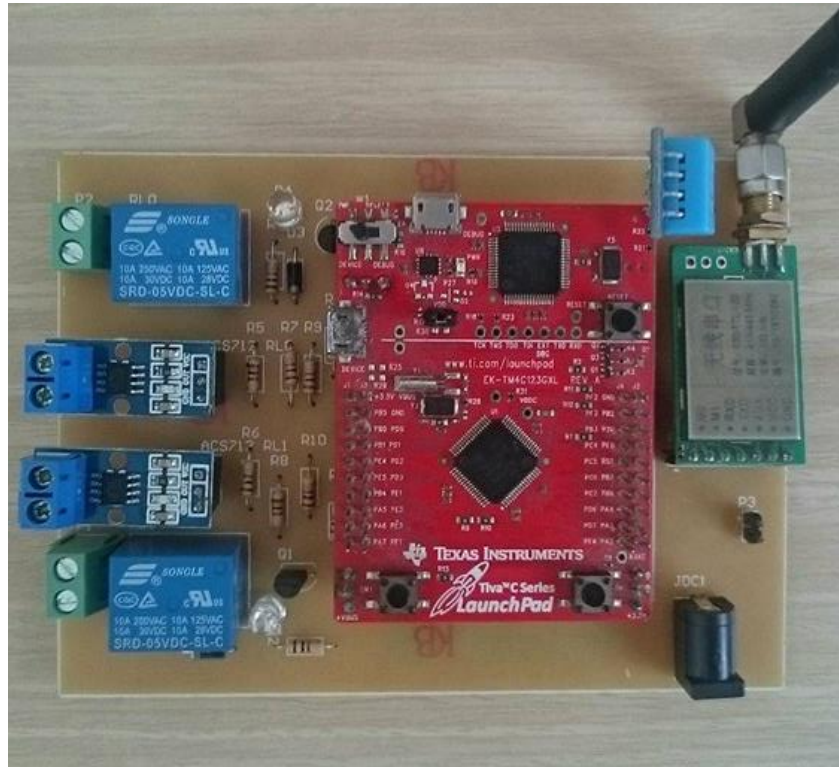
Nhận và phân tích lệnh từ raspberry gửi xuống, khi có dữ liệu đưa xuống, Tiva C sẽ vào ngắt UART và đưa các data này vào buffer. Trong chương trình chính sẽ lấy dữ liệu trong buffer ra, gặp 0x0A thì ngừng và bắt đầu kiểm tra CRC. Nếu CRC=0, sẽ kiểm tra lệnh cụ thể và thực hiện. Sau khi thực hiện xong sẽ gửi kết quả về cho Raspberry. Ngược lại nếu CRC != 0, sẽ gửi lệnh báo cáo data trong quá trình truyền bị sai để Raspberry biết và truyền lại.



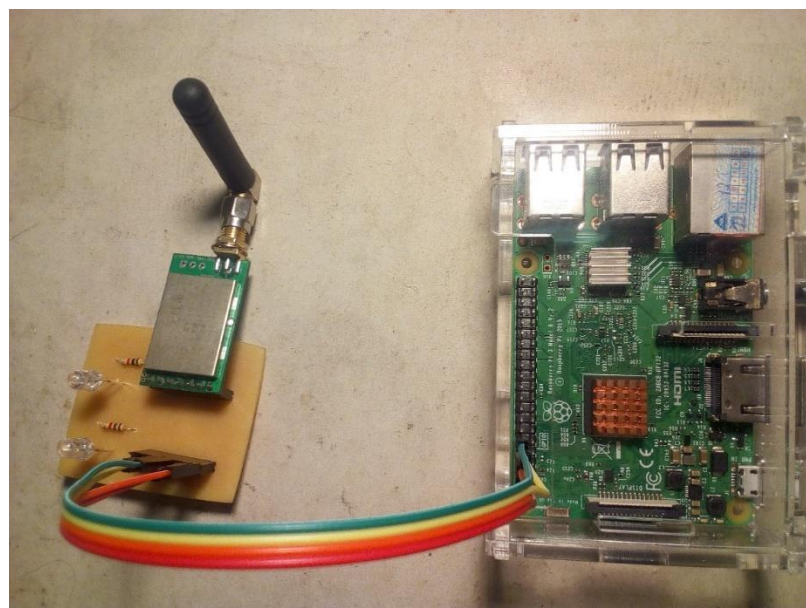
Hình 4 6 Giải thuật nhận dữ liệu và xử lý ở Tiva

5. KẾT QUẢ THỰC HIỆN

5.1 Kết quả thi công mạch

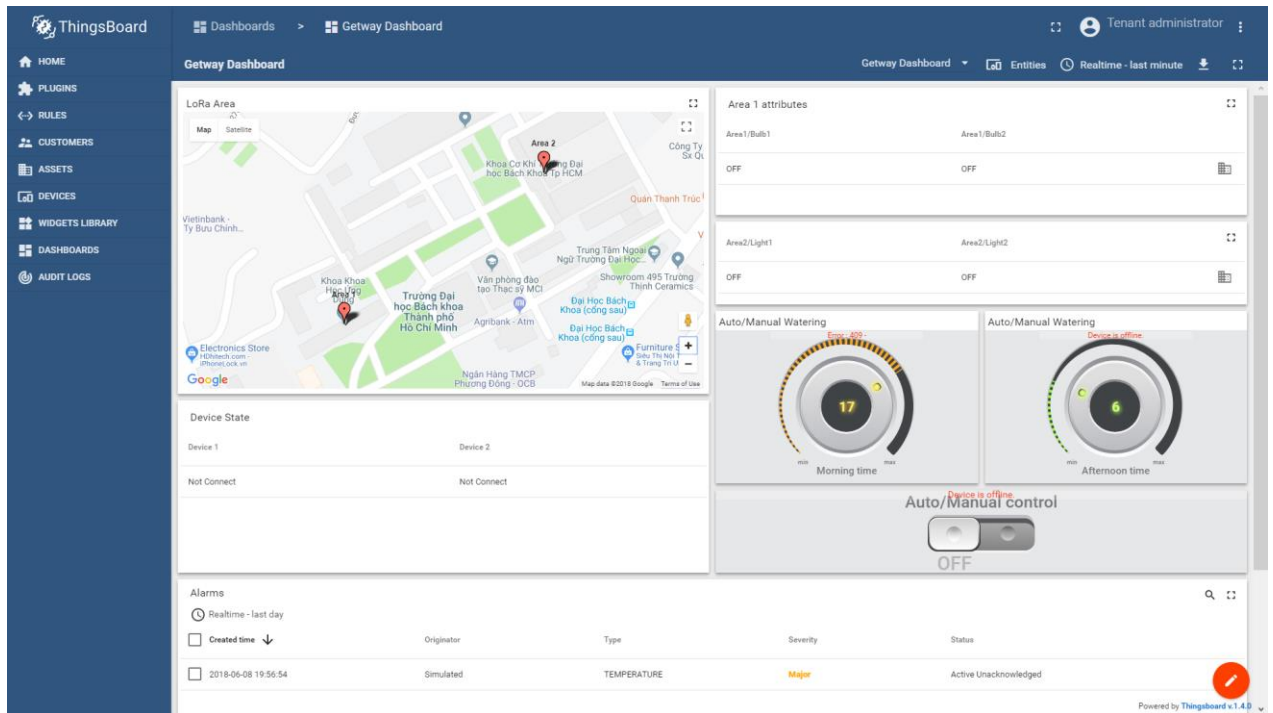


Hình 5 1 Hình ảnh thực tế tại mỗi slave



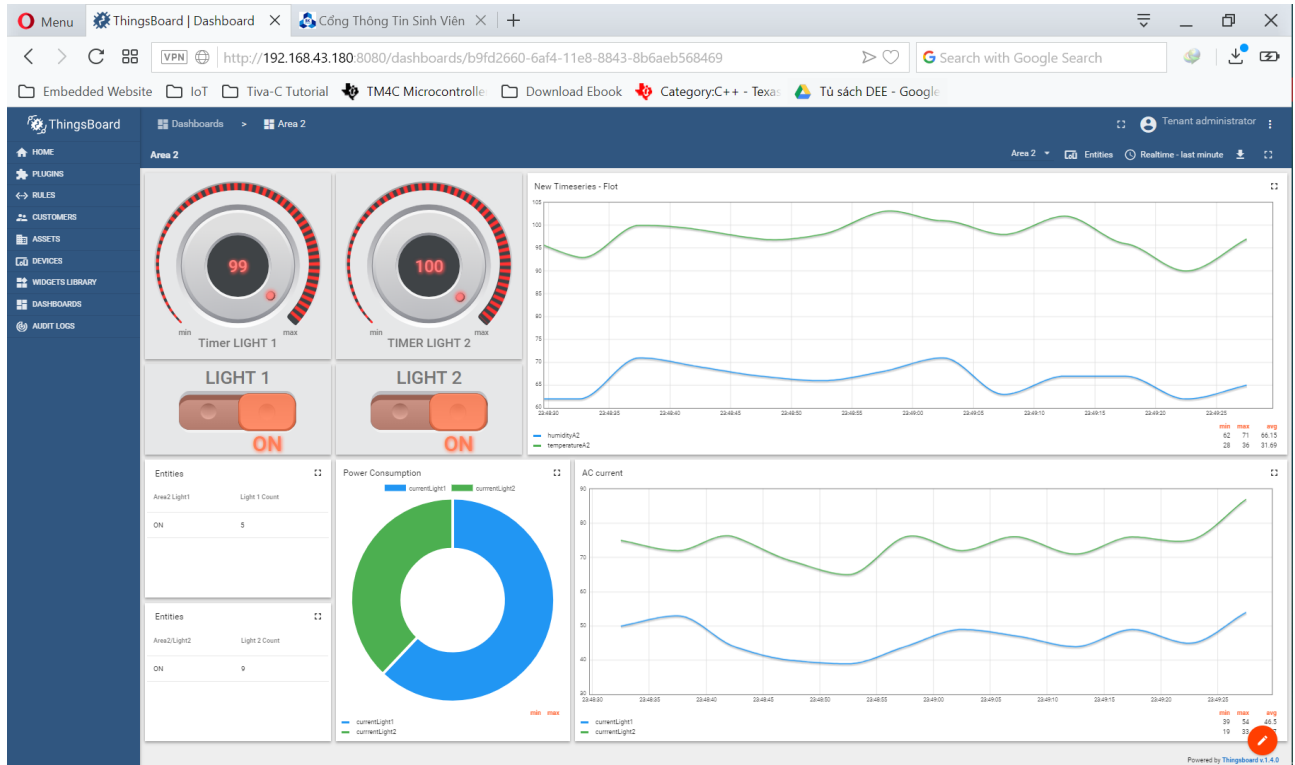
Hình 5 2 Hình ảnh thực tế tại master

Dashboard ở gateway, tổng quan về toàn bộ hoạt động của hệ thống như: vị trí đặt module LoRa, tình trạng bật tắt đèn ở từng khu vực, tình trạng kết nối của thiết bị với gateway, thông báo các cảnh báo, chế độ tự động bật tắt đèn theo thời gian định trước.



Hình 5 3 Giao diện hoạt động của hệ thống

Dashboard hiển thị ở một node cho thấy các giá trị nhiệt độ, độ ẩm, dòng điện của đèn 1 và dòng điện của đèn 2, bật tắt đèn thủ công và cài đặt thời gian bật tắt đèn bao lâu (giới hạn từ 0-120 phút).



Hình 5 4 Giao diện cụ thể tại một salve

Nhận xét:

- Trang web đơn giản, dễ quan sát.
- Tốc độ tải trang web nhanh.
- Tự động refresh giúp cho việc theo dõi dữ liệu liên tục và tiện lợi.
- File Excel có dung lượng nhẹ, định dạng dễ quan sát theo từng mốc thời gian.
- Cảnh báo đến smartphone giúp đưa ra quyết định xử lý kịp thời.

6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Sau khi nghiên cứu và thực hiện đề tài “*Hệ thống giám sát môi trường và điều khiển thiết bị ứng dụng LoRa và Thingsboard IoT platform*”, em đã thực hiện được những yêu cầu ở đề tài này như sau:

- Tìm hiểu về máy tính Raspberry.
- Làm quen với hệ điều hành linux (Raspbian).
- Kỹ năng lập trình ngôn ngữ lập trình mới như Python, ngoài ngôn ngữ được học ở trường là C.
- Cách sử dụng Tiva C và thư viện hàm API của TivaWare.
- Hiểu được cách sử dụng module LoRa cơ bản, cách thiết lập một mạng nhỏ gồm 1 master và 2 slave.
- Sử dụng một IoT platform, ở đây là Thingsboard
- Lấy được dữ liệu thu thập về máy tính người sử dụng.
- Lắp đặt và thiết kế mô hình sản phẩm.
- Sản phẩm hoạt động ổn định.

Trong thời gian cho phép đề tài đã thực hiện xong các yêu cầu đặt ra. Hệ thống hoạt động nhịp nhàng ổn định. Người dùng có thể truy cập từ nhiều thiết bị có kết nối internet như máy tính bàn, laptop, máy tính bảng, điện thoại thông minh...vv. Đa số các thiết bị trên cho khả năng hiển thị dữ liệu tốt.

6.2 Hướng phát triển

Ngoài những kết quả đã đạt được trong đề tài này, hướng phát triển trong tương lai nhằm khắc phục những thiếu sót, cải tiến sản phẩm dựa trên sự hỗ trợ vô cùng lớn từ máy tính nhỏ Raspberry và cộng đồng phát triển. Hướng phát triển trong tương lai:

- Điều khiển bật tắt các thiết bị ổn định.
- Mở rộng điều khiển các thiết bị từ xa qua web như đèn, quạt, thiết bị làm mát và điều chỉnh độ ẩm cho thích hợp trong môi trường sản xuất...

- Phân tích dữ liệu thu thập được từ các cảm biến, từ đó đưa ra các giải pháp tối ưu như: thời gian tới bao lâu là hợp lí, số lượng phân bố cần dùng, ghi chú, nhắc nhở. Phân tích dữ liệu bằng các tool Big Data như: Edge computing, KNIME, OpenRefine ...
- Đưa vào hệ thống tích hợp vào các platform lớn, nhiều tính năng như: IBM Watson, Amazon AWS IoT...

7. TÀI LIỆU THAM KHẢO

- [1] Getting Started with the Tiva™ TM4C123G LaunchPad Workshop
- [2] Datasheet LoRa UART E32-TTL-100
- [3] Datasheet DHT11
- [4] DHT11 temperature & humidity sensor code snippet,
<https://forum.43oh.com/topic/7458-dht11-temperature-humidity-sensor-code-snippet/>
- [5] Tiva C Tutorial,
<https://sites.google.com/site/luiselectronicprojects/tutorials/tiva-tutorials>
- [6] ACS712 Current Sensor User Manual, <http://henrysbench.capnfatz.com/>
- [7] Thingsboard documentation, <https://thingsboard.io/>
- [8] ADC Oversampling Techniques for Stellaris® Family Microcontrollers, Texas Instrument
- [9] CRC16, <https://www.lammertbies.nl/comm/info/crc-calculation.html>
- [10] Cyclic Redundancy Check,
https://vi.wikipedia.org/wiki/Cyclic_Redundancy_Check
- [11] Bảng 1 So sánh đặc điểm các chuẩn truyền không dây, slide 16, Wireless weather station, Ahmad A. Kashef, Anas W. Hussain
- [12] Bảng 3 Đặc tính công nghệ LoRa,
<http://www.electronicdesign.com/industrial-automation/choices-abound-long-range-wireless-iot>

- [13] Hình 2.1 Mạng không dây LoRa, <https://intelilight.eu/gsm-operators-promise-internet-things-revolution-nb-iot-communications/>
- [14] Hình 2.2 So sánh băng thông và khoảng cách truyền các chuẩn không dây, <http://www.electronicdesign.com/industrial-automation/choices-abound-long-range-wireless-iot>
- [15] Hình 2.3 Các loại module LoRa, <http://www.electronicdesign.com/industrial-automation/choices-abound-long-range-wireless-iot>
- [16] Hình 2.4, 2.5 và 2.6, các thông tin về module Lora, tham khảo datasheet
- [17] Hình 2.7 Raspberry Pi 3, <https://www.amazon.co.uk/Raspberry-Pi-Model-Quad-Motherboard/dp/B01CD5VC92>
- [18] Hình 2.8 Kit Tiva C launchpad, <https://www.element14.com/community/docs/DOC-67569/1/tiva-c-series-launchpad-low-cost-evaluation-board-for-arm-cortex-m4f-based-microcontrollers>
- [19] Hình 2.9, 2.10, 2.11, các thông tin về cảm biến DHT11, <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>
- [20] Hình 2.13 module ACS712 5A, <http://www.robotpark.com/ACS712-20A-Current-Sensor>
- [21] Hình 2.15, 2.16, 2.18, 2.20, các thông tin về Thingsboard platform, <https://thingsboard.io/docs/>
- [22] Hình 2.17 Mô hình MQTT, <https://www.hivemq.com/blog/how-to-get-started-with-mqtt>
- [23] Hình 3.1 Mô hình smart farming, <https://www.richardvanhooijdonk.com/en/smart-farming-new-benchmark/>
- [24] Modbus serial, http://www.modbus.org/docs/Modbus_over_serial_line_V1.pdf
- [25] Hình 3.2 Module RF, <http://www.electrorobotech.electrogroups.org/shop/modules/315-mhz-rf-module/>

8. PHỤ LỤC