

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP
Kỹ sư ngành Kỹ Thuật Máy Tính

**Hệ thống kiểm tra độ mặn nguồn nước
thủy lợi kết hợp IoT và
điện toán đám mây**
(Salinity Monitor)

Hội đồng:

Kỹ Thuật Máy Tính

Giảng viên hướng dẫn:

TS. Phạm Quốc Cường

Giảng viên phản biện:

TS. Lê Trọng Nhân

Sinh viên thực hiện:

Võ Hùng Cường
Trần Đức Hưng

– 1410483
– 1411631

Tháng 6 năm 2018

Lời cảm ơn

Trong suốt thời gian hiện thực luận văn từ giai đoạn đề cương cho đến khi kết thúc, chúng tôi đã gặp khá nhiều khó khăn cũng như trực trặc trong quá trình nghiên cứu và phát triển đề tài, nhưng với sự ủng hộ, động viên và giúp đỡ của nhiều người. Chúng tôi đã vượt qua, hoàn thành và đạt được nhiều kết quả mong muốn.

Đầu tiên, chúng tôi xin chân thành cảm ơn Tiến sĩ Phạm Quốc Cường, người đã định hướng đề tài, đã cùng đồng hành, hỗ trợ, góp ý cho chúng tôi cả về mặt kiến thức, tinh thần lẫn kinh phí trong quá trình nghiên cứu và tiến hành hiện thực luận văn.

Chúng tôi xin gửi lời cảm ơn chân thành đến Tiến sĩ Lê Trọng Nhân, đã luôn trao đổi góp ý, hỗ trợ và hướng dẫn chúng tôi mỗi khi có thắc mắc về những vấn đề kĩ thuật.

Xin chân thành gửi lời cảm ơn đến các thầy cô giảng dạy tại trường Đại học Bách Khoa thành phố Hồ Chí Minh, đặc biệt tại Khoa Khoa học & Kỹ thuật Máy Tính đã giảng dạy, truyền đạt những kiến thức bổ ích, là nền tảng cho việc hoàn thành luận văn của ngày hôm nay.

Cuối cùng, chúng tôi gửi lời cảm ơn đến tất cả mọi người đã ủng hộ chúng tôi, sự giúp đỡ của những người bạn trong lớp học MTKTTN14, bạn bè, người thân và gia đình.

Cam đoan

Luận văn của chúng tôi có tham khảo các tài liệu, bài báo, trang web được trình bày ở **CHƯƠNG 6 – Tham khảo**, và ở mỗi tham khảo đều trích dẫn nguồn gốc của nó. Chúng tôi xin cam đoan rằng ngoài trích dẫn từ các tham khảo trên toàn bộ nội dung trong luận văn là do chúng tôi tự tạo ra từ kết quả thực hiện của riêng chúng tôi, không sao chép từ bất kì tài liệu nào khác.

Chúng tôi sẽ hoàn toàn chịu xử lý theo quy định nếu có bất cứ sai phạm nào so với lời cam kết.

MỤC LỤC

Chương 1	10
Giới thiệu tổng quan	10
1.1 Nhu cầu cấp thiết và ý tưởng của đề tài	10
1.2 Mục tiêu tổng quát	12
Chương 2	14
Cơ sở lý thuyết	14
2.1 Internet of Things (IoT).....	14
2.2 Các hệ thống (công trình) tương tự đã được phát triển.....	14
2.3 Mô hình hệ thống tổng quát kết hợp IoT.....	16
Chương 3	23
Phân tích và thiết kế	23
3.1 Mô hình chi tiết, các lý thuyết liên quan	23
3.1.1 Sơ đồ khối thiết kế.....	23
3.1.2 Kỹ thuật, thư viện, phần cứng, mô hình và các lý thuyết liên quan	23
3.1.2.1 Độ mặn.....	23
3.1.2.2 Node cảm biến.....	24
3.1.2.3 Công nghệ LoRa, giao tiếp - truyền dữ liệu giữa các node thu thập và node trung tâm	37
3.1.2.4 Node trung tâm - GateWay	42
3.1.2.5 Server.....	44
3.1.2.6 Web Client	56
3.2 Thiết kế chi tiết	60
3.2.1 Node cảm biến	60
3.2.1.1 Flow chart	60
3.2.1.2 Schematic và board mạch.....	62

3.2.1.3	Prototype data truyền và nhận tại node cảm biến	64
3.2.2	Node trung tâm	66
3.2.2.1	Flow chart thiết kế ở Node trung tâm – GateWay	66
3.2.2.2	Schematic và board mạch thiết kế tại node trung tâm – GateWay	68
3.2.2.3	Prototype data truyền và nhận tại GateWay	70
3.2.2.4	Cấu trúc tin nhắn SMS đến GateWay	71
3.2.3	Server	72
3.2.4.1	Thiết kế cơ sở dữ liệu	72
3.2.4.2	Cấu trúc của Server	77
3.2.4	Web Client	81
Chương 4		82
Hiện thực hệ thống		82
4.1	Node cảm biến	82
4.2	GateWay	84
4.3	Giao diện web cho người dùng	86
Chương 5		91
Tổng kết đánh giá các kết quả hiện thực		91
5.1	Hoàn thành và kết quả đạt được	91
5.2	Hạn chế	92
5.3	Kế hoạch và hướng phát triển trong tương lai	92
Chương 6		93
Tham khảo		93

DANH SÁCH HÌNH VẼ

Hình 1.1. Bản đồ xâm nhập mặn vùng Đồng bằng sông Cửu Long ^[2]	11
Hình 1.2. Hàng chục nghìn ha lúa ở miền Tây chết do nước mặn xâm nhập ^[3]	11
Hình 1.3. Một cống chống xâm ngập mặn được xây dựng tại Kiên Giang - ảnh Nguyễn Tuyễn ^[4]	12
Hình 2.1. Ứng dụng Web SmartBKTrffic ^[6]	15
Hình 2.2. SkyNet - hệ thống quan trắc khí thải ^[7]	15
Hình 2.3. Semtech's LoRa Technology Smart Parking ^[8]	16
Hình 2.4. Mô hình tổng quát 1 (không có node trung tâm)	17
Hình 2.5. Mô hình tổng quát 2 (Node trung tâm đóng vai trò là Gateway).....	19
Hình 3.1. Sơ đồ khối chi tiết cho thiết kế hệ thống	23
Hình 3.2. Digital C4E Sensor ^[9]	25
Hình 3.3. Sensor đo độ mặn của hãng Vernier ^[14]	26
Hình 3.4. Các chân ra của sensor ^[15]	27
Hình 3.5. P-NUCLEO-LRWAN1.....	28
Hình 3.6. SX1272MB2xAS LoRa [®]	30
Hình 3.7. Arduino Uno R3 ^[16]	31
Hình 3.8. Arduino Nano ^[17]	32
Hình 3.9. Atmega328P ^[35]	33
Hình 3.10. Measurement Node truyền dữ liệu đến Center Node sử dụng công nghệ Lora..	37
Hình 3.11. Năng lượng tiêu thụ định kỳ (màu xanh) và năng lượng tiêu thụ trung bình của một modem LoRa ^[10]	38
Hình 3.12. Dòng điện tiêu thụ của module SIM800 ở sleep mode ^[11]	39
Hình 3.13. Module E32-TTL-100 ^[18]	40
Hình 3.14. Một ví dụ lấy mẫu Long-Preamble sử dụng chu kỳ cho quá trình nhận	42

Hình 3.15. Mô hình ở node trung tâm.....	43
Hình 3.16. Arduino Mega 2560 Rev3 [20].....	43
Hình 3.17. Mô hình ở Server cùng với Database và User	44
Hình 3.18. Tổng quát xử lý tại server	46
Hình 3.19. Kiến trúc MVC	49
Hình 3.20. Request/Repose trong Kiến trúc MVC	49
Hình 3.21. Mô hình kết nối và truyền dữ liệu của một WebSocket.[29]	51
Hình 3.22. EJS tạo một khuôn mẫu cho HTML.....	56
Hình 3.23. Flow chart thiết kế ở các node cảm biến.....	60
Hình 3.24. Schematic thiết kế tại node cảm biến.....	62
Hình 3.25. Board mạch và mô hình 3D thiết kế tại node cảm biến.....	63
Hình 3.26. Flow chart thiết kế ở GateWay.....	66
Hình 3.27. Schematic thiết kết tại node cảm biến	68
Hình 3.28. Board mạch và mô hình 3D thiết kế tại GateWay	69
Hình 3.29. Sự ràng buộc các dữ liệu gây nên bởi application	76
Hình 3.30. Mô hình hoạt động của server.....	77
Hình 3.31. Biểu đồ High Lever Use Case của Web server	80
Hình 3.32. Cấu trúc file tương ứng tại browser của web user.....	81
Hình 4.1. Breadboard đầu tiên được sử dụng để testing tại node cảm biến.....	82
Hình 4.2. Board mạch của node cảm biến	82
Hình 4.3. Node cảm biến sau khi hoàn thiện	83
Hình 4.4. Testing tại GateWay	84
Hình 4.5. Board mạch của node trung tâm - GateWay.....	84
Hình 4.6. Node trung tâm – GateWay sau khi hoàn thiện	85
Hình 4.7. Giao diện trang Dashboard.....	86
Hình 4.8. Giao diện trang profile của user.....	87

Hình 4.9. Giao diện trang User Management	88
Hình 4.10. Giao diện trang Data Utilization	89
Hình 4.11. Giao diện trang Node Management.....	90

DANH SÁCH BẢNG

Bảng 3.1. Bảng phân loại thể hiện độ mặn của nước [13]	24
Bảng 3.2. Hàm lượng giới hạn hữu ích của độ mặn[12]	24
Bảng 3.3. Lược đồ cấu trúc dữ liệu của user	73
Bảng 3.4. Lược đồ cấu trúc dữ liệu của deleted _ user _ numbers.....	73
Bảng 3.5. Lược đồ cấu trúc dữ liệu của data.....	73
Bảng 3.6. Lược đồ cấu trúc dữ liệu của data _ latest.....	74
Bảng 3.7. Lược đồ cấu trúc dữ liệu của node _ data.....	75
Bảng 3.8. Các API cung cấp cho người dùng và Gateway	78
Bảng 3.9. Mô tả các API của server	79

CÁC TỪ VIỆT TẮT

ADC	Analog-to-digital converter
DBaaS	Database as a Service
DBAs	Database Administrators
DBMS	Database Management System
EJS	Embedded JavaScript templating
HTML	Hypertext Markup Language
I2C	Inter-Integrated Circuit
IoT	Internet of Thing
NoSQL	"non SQL" or "non relational"
ppt	Parts per thousand
SPI	Serial Peripheral Interface
SQL	Structured Query Language
UART	Universal asynchronous receiver-transmitter
WDT	WatchDog Timer

CHƯƠNG 1

GIỚI THIỆU TỔNG QUAN

1.1 Nhu cầu cấp thiết và ý tưởng của đề tài

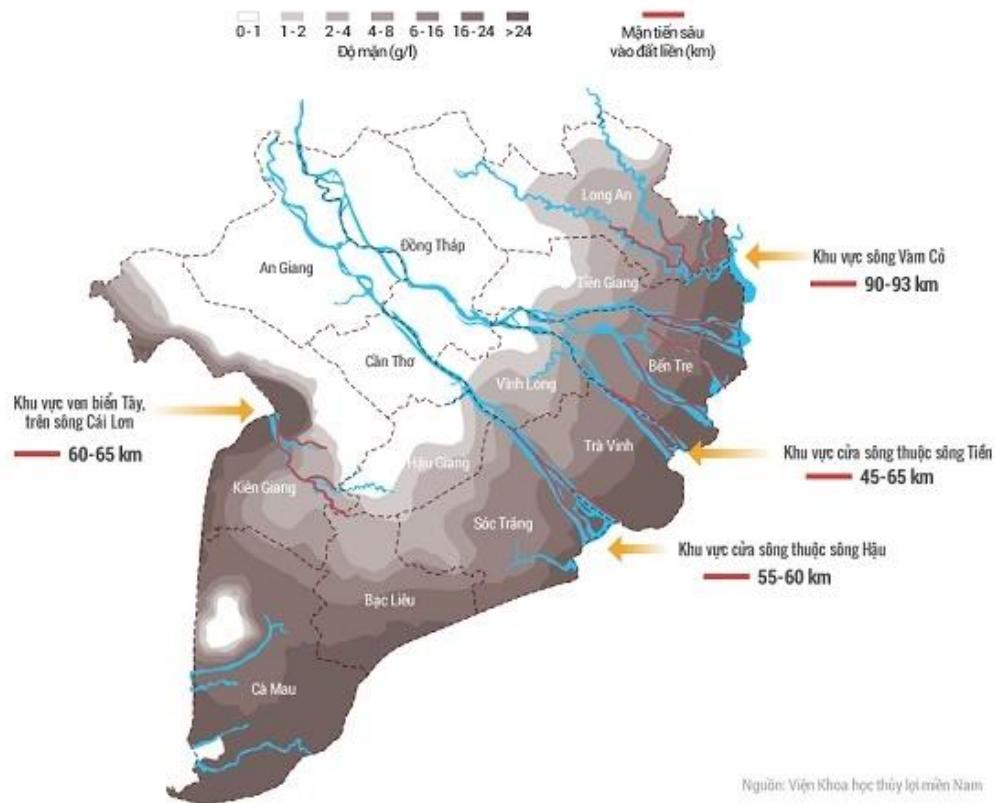
Biến đổi khí hậu đang làm cho các đại dương ấm lên. Từ năm 1961 đến năm 2003 nhiệt độ đại dương toàn cầu tăng bình quân $0,10^{\circ}\text{C}$. Nhiệt độ tăng làm cho tăng dung tích nước của các đại dương đồng thời làm cho băng tan từ các vùng cực Bắc và Nam cực, từ các khối băng tiềm tàng trên các núi cao. Hệ quả của các hiện tượng này là quá trình nước biển dâng. Từ năm 1961 đến 2003 tốc độ bình quân mực nước trung bình của các đại dương nâng lên khoảng $1,8 \pm 0,5 \text{ mm/năm}$. Nước biển dâng sẽ mở rộng vùng xâm nhập mặn, thu hẹp diện tích vùng nước ngọt.^[1]

Xâm nhập mặn ảnh hưởng đến 20% diện tích đất nông nghiệp và 50% của tất cả các vùng đất được tưới tiêu và là vấn đề bảo tồn đất lớn thứ hai.^[1]

Từ cuối năm 2014, nền nhiệt độ tăng cao, thiếu hụt lượng mưa, là nguyên nhân gây ra tình trạng hạn hán, xâm nhập mặn, đã gây thiệt hại nặng nề và tiếp tục đe dọa nghiêm trọng đến sản xuất và dân sinh. Các khu vực bị ảnh hưởng nặng là Nam Trung Bộ, Tây Nguyên, Đông Nam Bộ và Đồng bằng sông Cửu Long. Tính riêng năm 2015, ở khu vực Nam Trung Bộ và Tây Nguyên, đã có gần 40.000 ha lúa phải dừng sản xuất do thiếu nước, 122.000 ha cây trồng bị hạn hán, thiếu nước và hàng chục ngàn hộ dân bị thiếu nước sinh hoạt.^[2]

Đồng bằng sông Cửu Long là khu vực hiện nay đang bị ảnh hưởng nặng nhất của hạn hán, xâm nhập mặn và còn tiếp tục bị ảnh hưởng trong thời gian tới.^[2]

Xâm nhập mặn tại đồng bằng sông Cửu Long



Hình 1.1. Bản đồ xâm nhập mặn vùng Đồng bằng sông Cửu Long^[2]



Hình 1.2. Hàng chục nghìn ha lúa ở miền Tây chết do nước mặn xâm nhập^[3]



Hình 1.3. Một cổng chống xâm ngập mặn được xây dựng tại Kiên Giang - ảnh Nguyễn Tuyễn^[4]

Để giảm thiểu tác hại của việc xâm nhập mặn, cổng chống xâm ngập mặn được xây dựng và lắp đặt. Sau khi hoàn thành, công trình sẽ đảm bảo nguồn nước có nồng độ mặn thấp phục vụ tưới cho cây trồng, làm tăng năng suất nông nghiệp, giúp thích ứng với biến đổi khí hậu, bảo vệ và phát triển sản xuất, góp phần nâng cao đời sống của người dân.^[4]

Mặc dù cổng đã được xây dựng, nhưng việc đóng mở cổng phù hợp và đúng lúc để điều tiết độ mặn của nước ở cửa sông theo việc triều cường lên và xuống là hoàn toàn thủ công, do người quản lý cổng điều khiển bằng việc đo độ mặn nước thủy lợi, sau đó đưa ra quyết định đóng và mở cổng theo giá trị đo được. Công việc thủ công này khiến việc đóng mở cổng trở nên bất tiện, đôi khi không đúng lúc và kịp thời.

Vì yếu tố trên, dự án “Hệ thống kiểm tra độ mặn nguồn nước thủy lợi kết hợp IoT và điện toán đám mây” – Salinity Monitor được hình thành và phát triển với mục đích thu thập giá trị đo đặc độ mặn từ các khu vực riêng biệt của nguồn nước thủy lợi hoặc một dòng sông. Từ các dữ liệu theo thời gian thực này hệ thống sẽ thông báo đến người quản lý cổng thủy lợi khi nồng độ mặn vượt ngưỡng nhất định, cũng như có được những dữ liệu quan trọng dành cho việc thống kê và tính toán để đề ra giải pháp cho những vấn đề về môi trường. Bên cạnh đó việc tiếp cận và phát triển những công nghệ mới trong dự án này có thể hỗ trợ cho các dự án khác trong tương lai. Đề tài mang tính ứng dụng và thiết thực cao đã được nhóm chúng tôi lựa chọn làm Luận văn tốt nghiệp.

1.2 Mục tiêu tổng quát

- Tìm hiểu các thông số về độ mặn của nước sử dụng trong nông nghiệp.
- Tìm hiểu về cảm biến, máy đo độ mặn của nước trong nông nghiệp, hiện thực mạch đo độ mặn.

- Hiện thực giao tiếp ở mức năng lượng thấp giữa nhiều mạch đo độ mặn với một mạch thu thập dữ liệu trung tâm (có thể xem là Gateway) để đưa dữ liệu lên Internet, board mạch trung tâm này còn đóng vai trò là trung tâm thông báo, nhắn tin tới các cá nhân liên quan.
- Thiết kế API để đưa dữ liệu từ board mạch trung tâm lên Cloud Database hoặc Server.
- Hoàn thiện Web Server cho phép người dùng theo dõi thông tin, giám sát sử dụng dữ liệu để thống kê, phân tích.
- Đề xuất các mô hình khác có thể sử dụng được kết quả hiện thực hoặc công nghệ sử dụng trong hệ thống này. Khả năng kết hợp với các hệ thống khác.

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1 Internet of Things (IoT)

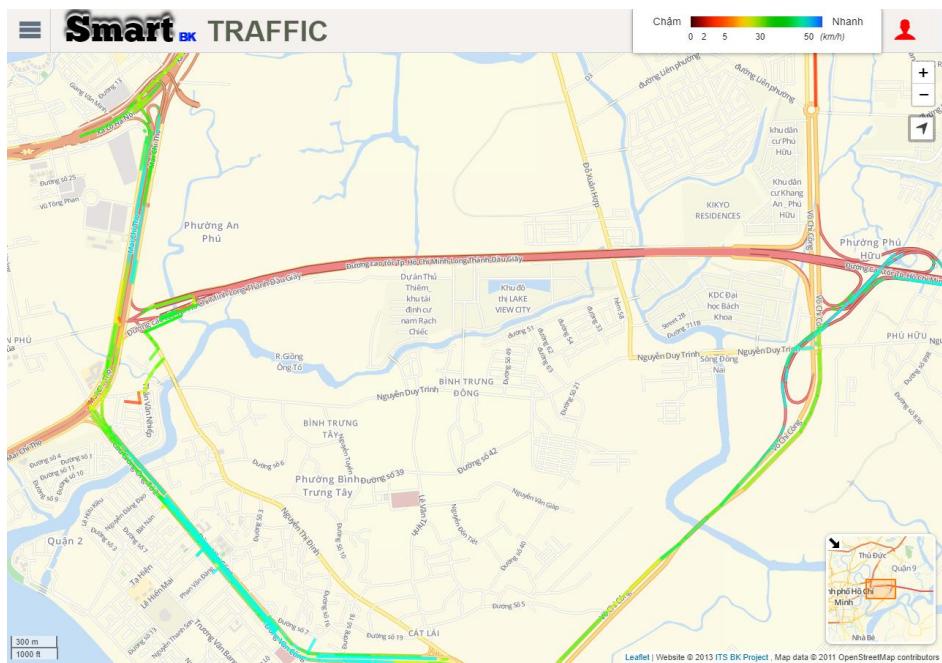
Internet of Things (IoT) là một liên mạng, trong đó các thiết bị điện, điện tử, phương tiện vận tải (được gọi là “thiết bị kết nối” và “thiết bị thông minh”), phòng ốc và các trang thiết bị khác được nhúng với các bộ phận điện tử, phần mềm, cảm biến, cơ cấu chấp hành cùng với khả năng kết nối mạng máy tính giúp cho các thiết bị này có thể thu thập và truyền tải dữ liệu.^[5]

Đo độ mặn của nước không phải là một vấn đề đặc biệt, nhưng để có được những dữ liệu độ mặn cập nhật thường xuyên liên tục dành cho việc phân tích, chính xác theo thời gian thực để cảnh báo kịp thời thì không thể sử dụng một cách đo và ghi chép dữ liệu thủ công như từ trước đến nay. Dựa vào những đặc điểm kỹ thuật đã nêu trên của IoT, một mô hình tự động và liên tục sẽ được hiện thực để áp ứng nhu cầu cấp thiết này.

2.2 Các hệ thống (công trình) tương tự đã được phát triển

BK Traffic

Bktraffic <http://traffic.hcmut.edu.vn> kết hợp hệ thống xe bus cùng định vị GPS, gửi dữ liệu qua hạ tầng mạng 3G để theo dõi vị trí xe và tính toán tốc độ lưu thông trên đoạn đường xe đang di chuyển.



Hình 2.1. Ứng dụng Web SmartBKTrffic^[6]

SkyNet^[7]

Hệ thống quan trắc khí thải của các phương tiện giao thông, thu thập dữ liệu các yếu tố môi trường qua mạng 3G dành cho việc phân tích và đánh giá môi trường tại khu vực khảo sát.

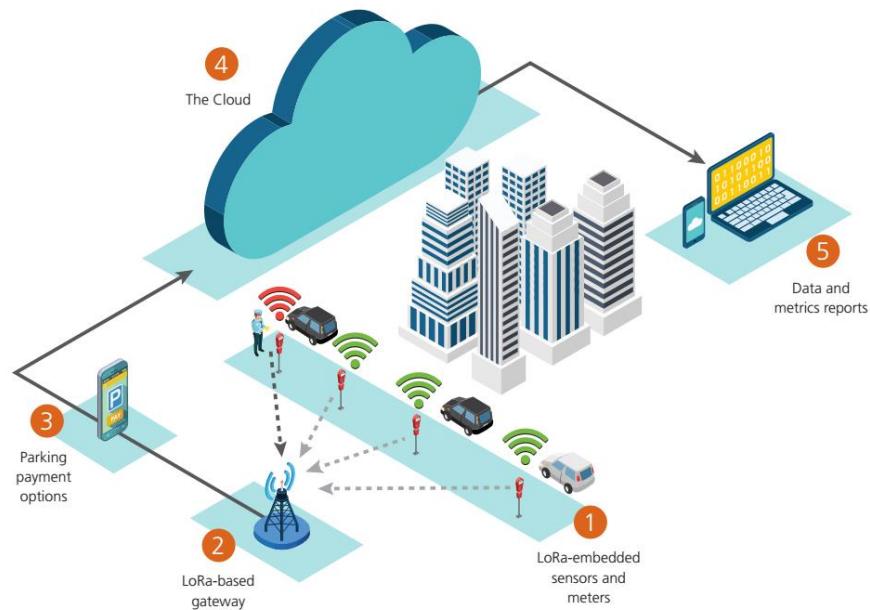


Hình 2.2. SkyNet - hệ thống quan trắc khí thải^[7]

Smart Parking^[8]

Một ứng dụng của công nghệ LoRa do Semtech phát triển, đỗ xe thông minh, các cảm biến tại các bãi đỗ xe có nhiệm vụ theo dõi những điểm đậu xe, thu thập và gửi dữ liệu về cho nhà cung cấp. Nhà cung cấp sử dụng những dữ liệu này, cấu hình và gửi xuống cho khách hàng, khách hàng có thể biết được những điểm đậu xe còn trống, đồng thời thông qua đó có

thể thanh toán phí dịch vụ sử dụng bãi đậu tương đương với thời gian đậu xe một cách đơn giản và nhanh chóng.

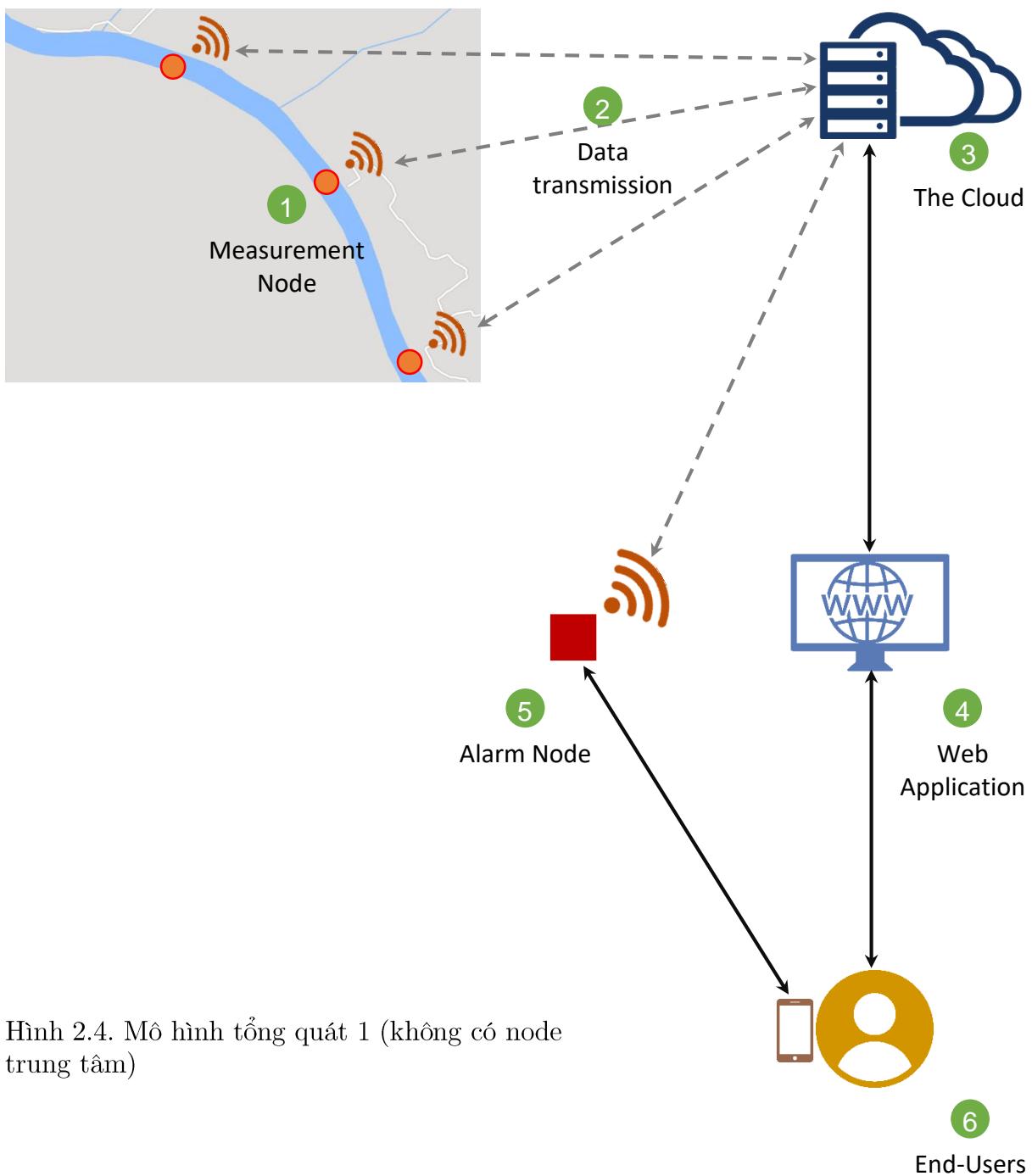


Hình 2.3. Semtech's LoRa Technology Smart Parking^[8]

2.3 Mô hình hệ thống tổng quát kết hợp IoT

Đề xuất 2 mô hình hệ thống tổng quát kết hợp IoT và The Cloud

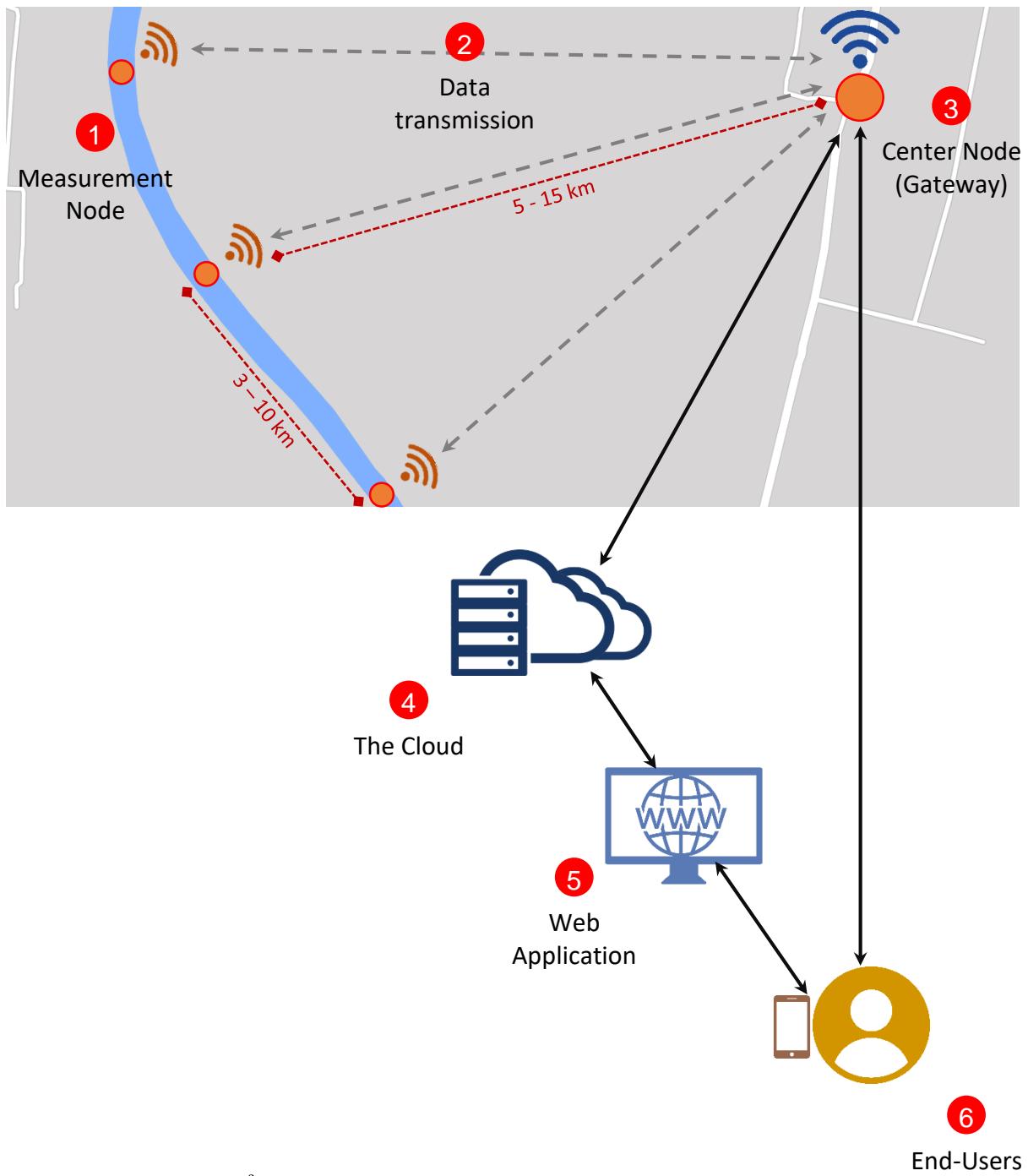
Độ mặn của nước sông và nguồn nước thủy lợi là không giống nhau ở các vị trí khác nhau trên một dòng chảy. Để tăng độ chính xác phục vụ cho hệ thống cảnh báo đồng thời cung cấp dữ liệu độ mặn tại nhiều vị trí khác nhau cho quá trình giám sát và phân tích tình hình nhiễm mặn tại các lưu vực sông thì hệ thống phải tiến hành đo độ mặn tại nhiều địa điểm khác nhau trên một khu vực hoặc một dòng chảy nhất định. Vì vậy mô hình tổng quát mà nhóm đưa ra gồm 2 mô hình sau:



Hình 2.4. Mô hình tổng quát 1 (không có node trung tâm)

Miêu tả cách thức hoạt động của mô hình tổng quát 1:

- 1 Các Measurement Node có chức năng đo giá trị độ mặn của nước từ sensor tại vị trí đặt node.
- 2 Quá trình truyền tải thông tin, sử dụng công nghệ di động viễn thông (GSM, 3G,...), mỗi Measurement Node gửi dữ liệu trực tiếp từ sensor lên Cloud thông qua Internet.
- 3 The Cloud, cung cấp API cho việc gửi dữ liệu lên từ các Measurement Node, sau đó lưu trữ dữ liệu dưới một dạng database phù hợp. Cung cấp API cho việc gửi yêu cầu xuống các Measurement Node hoặc Alarm Node dành cho việc cảnh báo với người dùng cuối, cung cấp API cho việc truy vấn dữ liệu từ Web Application.
- 4 Ứng dụng web dành cho việc giám sát, truy xuất dữ liệu dành cho phân tích và thống kê.
- 5 Alarm Node, dành cho việc cảnh báo bằng tín hiệu như đèn báo, còi hú, gửi tin nhắn đến End-User. Đồng thời sẽ là nơi nhận tin nhắn từ End-User để truy xuất dữ liệu gần nhất từ Cloud, hoặc dữ liệu đo đạt trực tiếp tức thời từ các Measurement Node và trả giá trị về cho End-User.
- 6 End-User, thường là người quản lý, giám sát hệ thống thủy lợi, cống chống ngập mặn. Là nhà khoa học, nghiên cứu muốn sử dụng dữ liệu từ hệ thống để phân tích cho các mục đích khoa học hoặc các nhà phát triển muốn cải tiến và ứng dụng công nghệ của hệ thống.



Hình 2.5. Mô hình tổng quát 2 (Node trung tâm đóng vai trò là Gateway)

Miêu tả cách thức hoạt động của mô hình tổng quát 2:

- 1 Các Measurement Node có chức năng đo giá trị độ mặn của nước từ sensor tại vị trí đặt node. Các node này cách nhau từ vài km cho đến hơn 10 km.
- 2 Quá trình truyền nhận thông tin từ các Measurement Node đến Center Node, sử dụng sóng vô tuyến để truyền dữ liệu với khoảng cách lên đến hơn 10km.
- 3 Center Node, đóng vai trò là Gateway quản lý thông tin nhận được từ các Measurement Node, gửi dữ liệu đó lên Cloud. Đồng thời Center còn đóng vai trò Alarm Node như mô hình tổng quát 1, dành cho việc cảnh báo bằng tín hiệu đến End-User. Đồng thời sẽ là nơi nhận tin nhắn từ End-User qua mạng di động viễn thông(GSM, 3G,...) trả giá trị dữ liệu trở về cho End-User.
- 4 The Cloud có vai trò tương tự với mô hình tổng quát 1. Cung cấp các API cần thiết cho Center Node và Web Application.
- 5 Ứng dụng web có vai trò giống với mô hình tổng quát 1, dành cho việc giám sát, truy xuất dữ liệu dành cho phân tích và thống kê.
- 6 End-User từ mô hình tổng quát 1.

So sánh 2 mô hình đề xuống, quyết định lựa chọn mô hình hiện thực

Mô hình tổng quát 1 (Không có node trung tâm)

- Ưu điểm:
 - Hiện thực các node đơn giản, độc lập.
 - Không phụ thuộc vào khoảng cách vật lý giữa các node con, cũng như giữa các node con đến node trung tâm.
 - Số lượng các node không ảnh hưởng nhiều đến hiệu suất của hệ thống.
- Nhược điểm:
 - Lê thuộc quá nhiều vào mạng di động (GSM, 3G,...).
 - Việc quản lý và điều khiển các node trở nên khó khăn khi phát triển và mở rộng hệ thống, số lượng node ngày càng lớn.
 - Các node xử lý hoàn toàn độc lập gây lãng phí lớn nguồn tài nguyên thông tin.
 - Cần chi phí duy trì cho việc sử dụng mạng di động.

Mô hình tổng quát 2 (Có node trung tâm đóng vai trò là Gateway). Đối nghịch với mô hình 1. Ưu điểm của mô hình 1 là nhược điểm của mô hình 2 và ngược lại.

- Ưu điểm:
 - Không lê thuộc quá nhiều vào mạng di động (GSM, 3G,...).
 - Chỉ tốn chi phí đầu tư ban đầu và bảo trì nếu cần thiết.
 - Việc quản lý và điều khiển các node trở nên đơn giản khi phát triển và mở rộng hệ thống, số lượng node ngày càng lớn.
 - Dễ dàng xây dựng mạng liên kết trao đổi thông tin giữa các thiết bị với nhau, tận dụng triệt để nguồn tài nguyên thông tin.
- Nhược điểm:
 - Hiện thực hệ thống trở nên khá phức tạp.
 - Khoảng cách vật lý giữa các node con, cũng như giữa các node con đến node trung tâm là một vấn đề lớn cần quan tâm trong mô hình này.
 - Khi số lượng các node trong một khu vực lớn, lượng node gửi dữ liệu đến một gateway tăng, hiệu suất của gateway bị giảm xuống, vì vậy cần cài đặt thêm

gateway để đảm bảo hệ thống vận hành tốt. Hiện thực giao tiếp giữa các gateway và các node con trở nên phức tạp hơn, cần phải có một sự chuẩn hóa nhất định.

Với hệ thống đo độ mặn cho nguồn nước thủy lợi trong đê tài, ngoài việc thu thập dữ liệu dành cho nghiên cứu, khảo sát và phân tích thì công việc cảnh báo cho quản lý công thủy lợi đóng vai trò quan trọng, vì vậy việc quá phụ thuộc vào mạng di động là không tốt, nó sẽ ảnh hưởng đến giai đoạn thu thập dữ liệu theo thời gian thực, gây gián đoạn nếu mạng di động gặp vấn đề.

Bên cạnh đó, dữ liệu độ mặn được phân chia theo khu vực, đồng thời với việc triển khai ứng dụng cho cả những khu vực lớn như Đồng bằng sông Cửu Long, thì việc quản lý bằng mô hình Gateway trở nên dễ dàng và phù hợp hơn khi tạo được một mạng liên kết lớn giữa các Gateway và các node, ứng dụng hoàn hảo để phát triển và thiết kế nền mô hình “Môi trường thông minh” (Smart Environment).

Ngoài ra, một mô hình giám sát và thu thập dữ liệu thì vẫn đề nâng lượng cần tập trung hàng đầu. Các node con đo độ mặn cần phải duy trì hoạt động trong một khoảng thời gian dài lên đến hàng tháng, hàng năm. Mặc dù việc giao tiếp thông qua mạng di động với những module riêng biệt có khả năng làm giảm năng lượng tiêu thụ nhưng so sánh với việc truyền thông tin bằng sóng vô tuyến với công nghệ LoRa (được sử dụng trong đê tài, sẽ được đề cập sau) sẽ tiết kiệm năng lượng hơn rất nhiều.

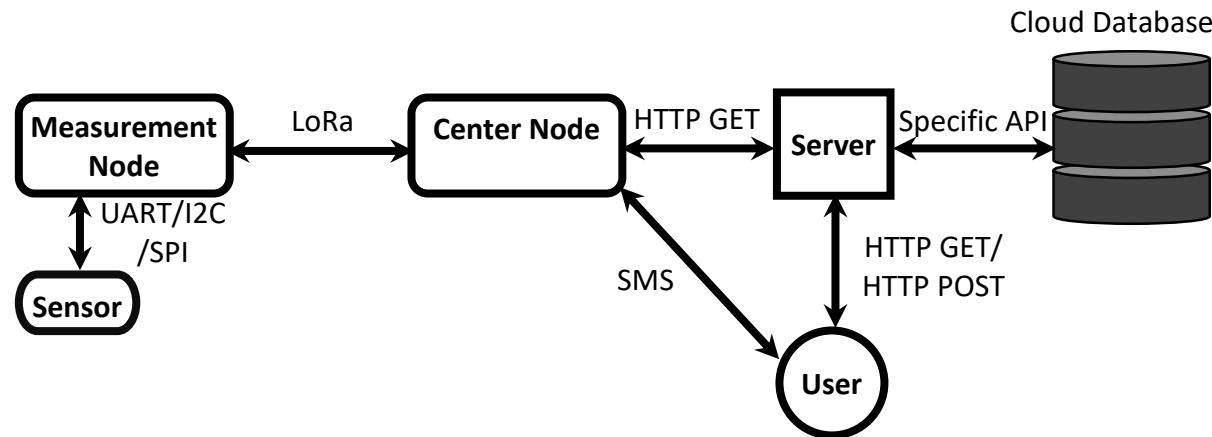
CHƯƠNG 3

PHÂN TÍCH VÀ THIẾT KẾ

3.1 Mô hình chi tiết, các lý thuyết liên quan

3.1.1 Sơ đồ khái thiết

Như đã giới thiệu mô hình tổng quát ở Mục 2.3, mô hình chi tiết sẽ đi sâu vào thiết kế kỹ thuật của hệ thống bao gồm các thành phần chính: Các node cảm biến thu thập dữ liệu từ các sensor, máy đo độ mặn, giao tiếp với board mạch chính bằng các giao thức truyền dữ liệu riêng biệt như UART, SPI, I2C,... dữ liệu truyền về node trung tâm qua sóng RF sử dụng công nghệ LoRa, node trung tâm up dữ liệu lên server thông qua giao thức truyền tải web mà cụ thể là HTTP GET, server có nhiệm vụ định dạng lại dữ liệu, gắn thời gian thực vào đuôi dữ liệu, sau đó chuyển dữ liệu vào database thông qua những api riêng biệt dành cho mỗi loại database khác nhau. Ngoài ra server còn có chức năng đóng vai trò là một web server phục vụ cho nhu cầu truy cập của các end user bằng các trình duyệt trên mọi thiết bị có kết nối internet.



Hình 3.1. Sơ đồ khái thiết cho thiết kế hệ thống

3.1.2 Kỹ thuật, thư viện, phần cứng, mô hình và các lý thuyết liên quan

3.1.2.1 Độ mặn

Xem xét về môi trường, **độ mặn** hay độ muối được ký hiệu S‰ (S viết tắt từ chữ salinity - độ mặn) là tổng lượng (tính theo gram) các chất hòa tan chứa trong 1 kg nước. Trong hải

dương học, người ta sử dụng độ muối (salinity) để đặc trưng cho độ khoáng của nước biển, nó được hiểu như tổng lượng tính bằng gam của tất cả các chất khoáng rắn hòa tan có trong 1 kg nước biển.

Độ mặn là lượng muối hòa tan trong nước, thường được đo bằng $\frac{g \text{ (muối)}}{kg \text{ (nước biển)}}$. Độ mặn là một yếu tố quan trọng trong việc xác định nhiều khía cạnh hóa học của nước tự nhiên và các quá trình sinh học bên trong nó, cùng với nhiệt độ và áp suất, chi phối các đặc tính vật lý như mật độ và khả năng nhiệt của nước. [13]

Độ mặn			
Nước ngọt	Nước lợ	Nước mặn	Nước muối
< 0.05 %	0.05 – 3 %	3 – 5 %	> 5 %
< 0.5 %	0.5 – 30 %	30 – 50 %	> 50 %

Bảng 3.1. Bảng phân loại thể hiện độ mặn của nước [13]

Độ mặn là một yếu tố sinh thái có tầm quan trọng đáng kể, ảnh hưởng đến các loại sinh vật sống trong một vùng nước. Cũng như vậy, độ mặn ảnh hưởng đến các loại cây trồng sẽ tăng trưởng trong cả nước, hoặc trên đất bị nuôi bởi nước (hoặc nước ngầm).

Bảng 3.2 liệt kê một số hàm lượng giới hạn hữu ích của nước mặn:

Kiểu nước mặn	Hàm lượng (ppt)
Nước biển	35
Độ mặn uống được tối đa cho người	3
Độ mặn thích hợp cho người	0.5 tới 0.75
Sinh vật trong sa mạc	Nhỏ hơn 15, tối đa 25
Nước tưới (đối với tưới tiêu và các điều kiện đất đai tối ưu)	<ul style="list-style-type: none"> • Nhỏ hơn 0,75: không có rủi ro mặn hóa • 0,75 - 1,5: giảm năng suất các cây trồng nhẹ cảm mặn • 1,5 - 3,5: giảm năng suất nhiều loại các cây trồng • 3,5 - 6,5: chỉ có cây trồng chịu mặn • 6,5 - 8,0: giảm năng suất các cây trồng chịu mặn

Bảng 3.2. Hàm lượng giới hạn hữu ích của độ mặn^[12]

3.1.2.2 Node cảm biến

Tại node thu thập dữ liệu ta có thể chia làm 2 phần: cảm biến và board mạch.

Phần 1: Cảm biến

Trên thị trường hiện có rất nhiều máy đo độ mặn với những mức giá và độ chính xác khác nhau, tuy nhiên các máy đo trên chủ yếu đã được đóng gói và phù hợp cho việc đo độ mặn thủ công, việc tìm kiếm một cảm biến đo độ mặn phù hợp với đề tài có mặt trên thị trường Việt Nam là rất khó khăn. Tuy nhiên, sau một khoản thời gian dài tìm kiếm và khảo sát, nhóm đã tìm được các loại cảm biến khá phù hợp với dự án và được mô tả cụ thể dưới đây.

Digital C4E Sensor

Cảm biến “thông minh” C4E lưu trữ dữ liệu đã hiệu chuẩn và lịch sử dữ liệu trong bộ cảm biến. Điều này cho phép ta có một hệ thống “plug and play” mà không cần hiệu chỉnh lại.^[9]

Nhờ có giao thức Universal Modbus RS485, cảm biến C4E có thể kết nối với tất cả các thiết bị thường được sử dụng (Dataloger, Controller, ...).^[9]



Hình 3.1. Digital C4E Sensor^[9]

Đặc điểm kỹ thuật:

- Nguyên lý đo: Cảm biến độ dãy với 4 điện cực (2 graphic, 2 platinum)
- Phân giải: 0,01 đến 1 phụ thuộc vào vùng đo
- Độ chính xác: +/- 1 % của toàn bộ phạm vi đo
- Khoảng đo độ mặn: 5-60 g/Kg
- Thời gian đáp ứng: < 5 s
- Nhiệt độ làm việc: 0°C to 50°C

- Signal interface: Modbus RS-485 (hoặc SDI-12)
- Maximum refreshing time: Max < 1 s
- Sensor power-supply 5 đến 12 volts
- Tiêu thụ điện:

Standby : 25 μ A

Average RS485 (1 measure/seconde) : 6,3 mA

Average SDI12 (1 measure/seconde) : 9,2 mA

Current pulse : 500 mA

- Sensor Dimensions Diameter: 27 mm ; Length : 177 mm Weight 350g (sensor + 3 m cable)
- Material: PVC, DELRIN, stainless steel
- Chống nước: IP68

Mức giá: dao động từ 100-200 USD.

Tất cả tài liệu được tìm thấy từ cảm biến này chỉ dừng lại ở mức user manual mà không có đặc tả kỹ thuật cho việc giao tiếp, cũng như những biểu đồ hoặc thông số đặc biệt cho cảm biến. Vì vậy việc triển khai thực hiện cảm biến này là chưa khả quan.

Salinity Sensor Vernier (SAL-BTA)



Hình 3.2. Sensor đo độ mặn của hãng Vernier [14]

Đặc điểm kỹ thuật:

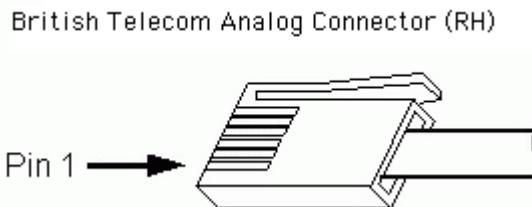
- Range of Salinity Sensor: 0 to 50 ppt (0 to 50,000 ppm)
- Accuracy using factory calibration: $\pm 3\%$ of full-scale (calibrated at 17.5 and 35 ppt)

- Accuracy using custom calibration: $\pm 1\%$ of full-scale (user calibrated)
- Response time: 90% of full-scale reading in 10 s
- Temperature compensation automatic from: 5 to 35°C
- Temperature range (can be placed in): 0 to 80°C
- Cell constant: 10 cm⁻¹
- Description: dip type, epoxy body, parallel platinum electrodes
- Dimensions: 12 mm OD and 150 mm length
- Calibration values: slope 16.3 ppt/V, intercept

Cách cảm biến đo độ mặn của nước: Cảm biến độ mặn Vernier đo khả năng hòa tan để dẫn được dòng điện giữa hai điện cực. Trong dung dịch, dòng điện dịch chuyển bằng cách vận chuyển các ion; do đó, nồng độ ion ngày càng tăng trong dung dịch sẽ dẫn đến giá trị dẫn điện cao hơn.

Cảm biến độ mặn thực chất là đo độ điện dẫn, được định nghĩa là nghịch đảo của điện trở.

Hiện nay tại Việt Nam các đại lý của các Vernier đã có và việc tìm kiếm sensor này để tiến hành project là không khó khăn.



Hình 3.3. Các chân ra của sensor [15]

- Pin 1 = Sensor output (+/-10V)
- Pin 2 = GND
- Pin 3 = Vres (resistance reference)
- Pin 4 = AutoIDENT (not supported on all sensors)
- Pin 5 = Power (+5VDC)
- Pin 6 = Sensor output (0-5V)

Việc lấy dữ liệu của sensor không sử dụng một kỹ thuật truyền đặc biệt nào, chỉ dựa vào giá trị output của sensor với mức điện áp tương ứng với độ mặn để có thể tính toán dựa vào slope và intercept. Một bộ ADC từ vi xử lý có thể đọc được giá trị này một cách nhanh chóng.

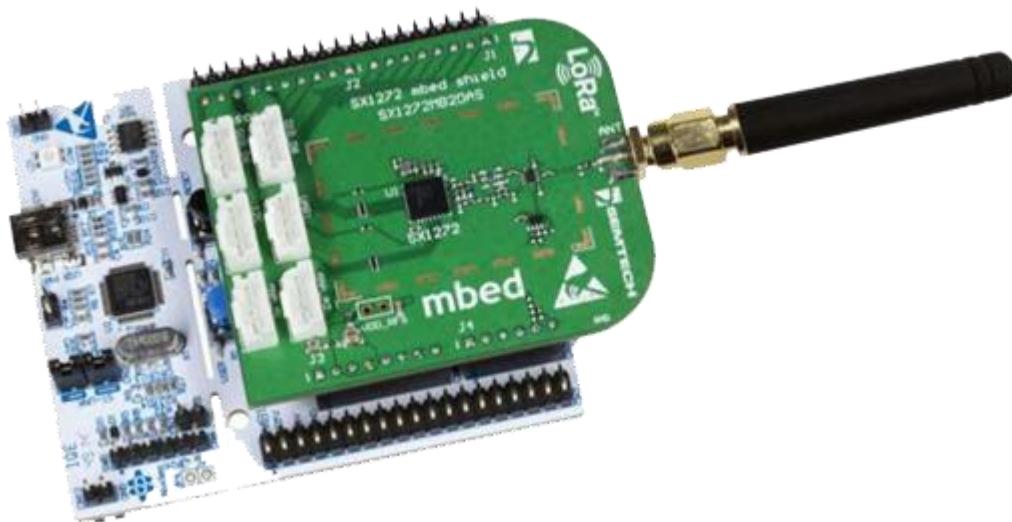
Nhược điểm của sensor này cũng như khuyến cáo khi dùng trong dự án: sản phẩm Vernier được thiết kế để sử dụng cho mục đích giáo dục, không được thiết kế và cũng không được đề xuất cho bất kỳ quy trình công nghiệp, y tế hoặc thương mại nào. Vì thế việc sử dụng sensor này chỉ dành trong mục đích xây dựng và thiết kế hệ thống. Để triển khai hệ thống trong công nghiệp lâu dài, một loại sensor khác dành riêng cho nó là cần thiết.

Phần 2: Board mạch vi xử lý

Giai đoạn đề cương luận văn:

Các cảm biến được nối với các chân trong vi xử lý để đọc giá trị. Trong mô hình này, với giai đoạn luận cương nhóm sử dụng bộ kit P-NUCLEO_LRWAN1

Gói STM32 Nucleo cho modem P-NUCLEO-LRWAN1 truyền nhận hiệu suất cao dùng công nghệ LoRa® và FSK/OOK RF là một công cụ phát triển để tìm hiểu và phát triển những giải pháp dựa trên công nghệ LoRa® và FSK/OOK.



Hình 3.4. P-NUCLEO-LRWAN1

Đặc điểm kỹ thuật:

- Ultra-low-power STM32L0 Series MCU, ARM® Cortex® -M0+ based with 192 Kbytes of Flash memory, 20 Kbytes of RAM, 6 Kbytes of EEPROM, LCD, crystal-less USB, T-RNG, PCROP
- 157 dB maximum link budget

- +20 dBm, 100 mW constant RF output versus V_{supply}
- +14 dBm high efficiency PA
- Programmable bit rate up to 300 kbps
- High sensitivity: down to -137 dBm
- Bullet-proof front end: IIP3 = -12.5 dBm
- 89 dB blocking immunity
- Low RX current of 10 mA, 200 nA register retention
- Fully integrated synthesizer with a resolution of 61 Hz
- FSK, GFSK, MSK, GMSK, LoRa® and OOK modulations
- Built-in bit synchronizer for clock recovery
- Sync word recognition
- Preamble detection
- 127 dB+ dynamic range RSSI
- Built-in temperature sensor and low-battery indicator 1.65 V to 3.6 V power supply

P-NUCLEO-LRWAN1 bao gồm hai khối chính:

- Board mở rộng SX1272MB2xAS LoRa® có một modem LoRa truyền nhận RF năng lượng thấp.



Hình 3.5. SX1272MB2xAS LoRa®

- Khối điều khiển: board NUCLEO-L073RZ MCU, vi xử lý 32-bit công suất cực thấp dành cho việc giao tiếp với các cảm biến cùng và với modem lora trên.



Hình 3.6. NUCLEO-L073RZ

Giai đoạn hiện thực luận văn:

Việc sử dụng các board mạch sẵn có trên trong giai đoạn hiện thực có nhiều bất cập vì các lý do như:

- Giá thành khá cao so với mặt bằng.
- Nhà cung ứng không có sẵn tại Việt Nam.

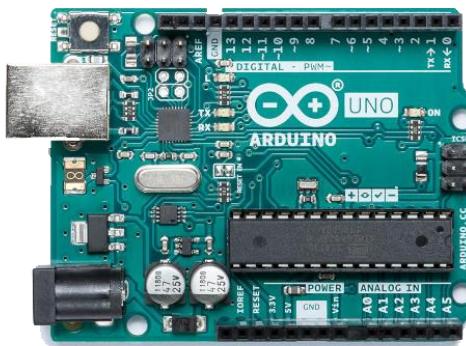
- Kích thước khá lớn so với quy định.
- Dư thừa nhiều tính năng không cần thiết được phát triển trên board mạch, dẫn đến hao tốn điện năng.

Chúng tôi đã hướng đến những sản phẩm khác có giá thành rẻ hơn, kích thước nhỏ hơn, cũng như tối ưu về mặt phần cứng giảm việc tiêu thụ năng lượng cho nó.

Arduino Uno R3:

Giai đoạn đầu của việc hiện thực, Aruino Uno R3 được lựa chọn sử dụng vì tính tiện lợi, nhanh gọn và rẻ tiền.

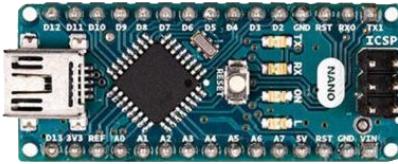
Arduino Uno là board mạch được nhắc tới đầu tiên khi nhắc tới Arduino. Là một board vi điều khiển dựa trên ATmega328P. Nó có 14 chân đầu vào/đầu ra digital (trong đó 6 có thể được sử dụng làm đầu ra PWM), 6 đầu vào analog, thạch anh 16 MHz, kết nối USB, giắc cắm nguồn, header ICSP và nút reset. Nó chứa mọi thứ cần thiết để hỗ trợ vi điều khiển; chỉ cần kết nối nó với máy tính bằng cáp USB hoặc cấp nguồn với bộ chuyển đổi AC hoặc DC để bắt đầu.



Hình 3.7. Arduino Uno R3 [16]

Arduino Nano:

Với việc cùng sử dụng một loại vi điều khiển với Arduino Uno, nhưng với thiết kế board mạch nhỏ hơn, hoàn thiện và tối ưu hơn, Arduino Nano được lựa chọn làm board mạch phát triển của đề tài.



Hình 3.8. Arduino Nano [17]

Thông số kỹ thuật:

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22 (6 of which are PWM)
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g

Tiến tới giai đoạn hiện thực cuối cùng, việc sử dụng Arduino Nano tuy đã đạt được một mức tối ưu nhất định, nhưng trên Arduino Nano lại chứa những phần tử điện tử đặc biệt tiêu hao năng lượng như ic ổn áp 5v Lm1117, USB, Led (đặc biệt là Led báo nguồn), vì thế, để đạt được mức tối ưu cuối cùng, loại bỏ các phần tử không ảnh hưởng đến board mạch tại node cảm biến, chỉ giữ lại nguyên vẹn phần xương sống cho board mạch (bare-bones board) chính là vi xử lý ATmega328P được sử dụng trên cả Arduino Uno lẫn Arduino Nano được nêu ra ở trên.

ATmega328P và các chế độ Sleep:



Hình 3.9. Atmega328P^[35]

Thông số kỹ thuật:

Bộ nhớ:

- 32KBytes Flash.
- 1KBytes EEPROM
- 2KBytes Internal SRAM
- Lưu giữ dữ liệu: 20 năm ở 85 ° C / 100 năm ở 25 °

Tính năng ngoại vi

- Hai bộ 8-bit Timer/Counters với Separate Prescaler và Compare Mode.
- Một 6-bit Timer/Counter với Separate Prescaler, Compare Mode và Capture Mode.
- Bộ đếm thời gian thực với bộ dao động riêng.
- 6 kênh PWM
- 6 bộ ADC 10-bit trong gói
- 2 Master / Slave SPI Serial Interface
- Một Serial USART có thể lập trình
- WatchDog Timer (WDT) với riêng biệt On-chip Oscillator
- Interrupt và Wake-up khi pin thay đổi

Tính năng vi điều khiển đặc biệt

- Power-on Reset và Programmable Brown-out Detection
- Ngắt nội và ngắt ngoại.

- Sáu chế độ ngủ: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby.

I / O và Gói

- 23 dòng I / O có thể lập trình
- PDIP 28 chân

Điện áp hoạt động:

- 1,8 - 5,5V

Phạm vi nhiệt độ:

- -40 ° C đến 105 ° C

Các lớp clock:

- 0 - 4MHz @ 1.8 - 5.5V
- 0 - 10MHz @ 2.7 - 5.5V
- 0 - 20MHz @ 4.5 - 5.5V

Công suất tiêu thụ ở 1MHz, 1.8V, 25 ° C

- Active Mode: 0.2mA
- Power-down Mode: 0.1µA
- Power-save Mode: 0,75µA (Bao gồm RTC 32kHz)

[36] Bộ vi điều khiển 8 bit của AVR bao gồm một số chế độ ngủ cho phép ứng dụng tắt các module không sử dụng trong MCU, nhờ đó tiết kiệm năng lượng. Thiết bị AVR cung cấp các chế độ ngủ khác nhau cho phép bạn điều chỉnh mức tiêu thụ điện năng phù hợp cho các yêu cầu ứng dụng.

Có năm chế độ ngủ để lựa chọn:

- Idle Mode
- Power Down
- Power Save
- Standby
- Extended Standby

Interrupt

Nếu một Interrupt (đã được kích hoạt) xảy ra trong lúc MCU đang ở chế độ ngủ, MCU sẽ tỉnh dậy. MCU sau đó được tạm dừng trong bốn cycles để tạo thời gian khởi động lại từ lúc ngủ, thực hiện interrupt routine, và tiếp tục thực hiện từ lệnh sau SLEEP. Nội dung của các thanh ghi và SRAM vẫn giữ nguyên, không bị thay đổi khi thiết bị thức dậy từ chế độ ngủ.

Reset

Nếu Reset xảy ra trong chế độ ngủ, MCU sẽ thức dậy và thực thi từ Reset Vector.

Idle Mode

Lệnh SLEEP làm MCU vào chế độ Idle, dừng CPU nhưng cho phép Serial Peripheral Interface (SPI), Universal Synchronous/Asynchronous Receiver/Transmitter (USART), Analog Comparator, 2-wire Serial Interface, Timer/Counters, WatchDog Timer (WDT), và hệ thống interrupt tiếp tục hoạt động. Chế độ ngủ này về cơ bản tạm dừng clock CPU (clkCPU) và clock Flash (clkFLASH), trong khi cho phép các clock khác chạy.

Idle Mode cho phép MCU thức dậy từ các ngắt ngoài cũng như các bộ phận bên trong như ngắt Timer Overflow và ngắt USART truyền hoàn toàn. Nếu không yêu cầu đánh thức từ ngắt Analog Comparator, bộ Analog Comparator có thể được cấp tắt. Điều này sẽ giảm mức tiêu thụ điện năng ở Idle Mode.

Power-Down Mode

Lệnh SLEEP làm cho MCU vào chế độ Power-Down. Trong chế độ này, bộ dao động bên ngoài được dừng lại, trong khi các ngắt ngoài, 2-wire Serial Interface address watch và Watchdog tiếp tục hoạt động (nếu được bật). Chỉ một trong những sự kiện này có thể đánh thức MCU:

- Reset bên ngoài
- Watchdog System Reset
- Watchdog Interrupt
- Brown-out Reset
- 2-wire Serial Interface address match
- External level interrupt trên INT
- Ngắt thay đổi pin

Chế độ ngủ này về cơ bản tạm dừng tất cả các clock được tạo ra, chỉ cho phép các module không đồng bộ hoạt động.

Power-save Mode

Lệnh SLEEP làm cho MCU vào Power-save Mode. Chế độ này giống với Power-down, với một ngoại lệ: Nếu Timer / Counter2 được kích hoạt, nó sẽ tiếp tục chạy trong khi ngủ.

Thiết bị có thể thức dậy từ sự kiện Timer Overflow hoặc Output Compare.

Nếu Timer / Counter2 không hoạt động, chế độ Power-down được khuyến nghị thay vì chế độ Power-save.

Standby Mode

Một clock bên ngoài được chọn, lệnh SLEEP làm cho MCU vào Standby Mode. Chế độ này giống với Power-Down với ngoại lệ là Oscillator vẫn tiếp tục chạy. Từ Standby Mode, thiết bị sẽ thức dậy trong sáu chu kỳ đồng hồ.

Extended Standby Mode

Một clock bên ngoài được chọn, lệnh SLEEP làm cho MCU vào chế độ Extended Standby Mode. Chế độ này giống với chế độ Power-Save với ngoại lệ là Oscillator vẫn tiếp tục chạy. Từ Extended Standby Mode, thiết bị sẽ thức dậy trong sáu chu kỳ đồng hồ.

Analog to Digital Converter

Nếu được bật, ADC sẽ được bật ở tất cả các chế độ ngủ. Để tiết kiệm điện, ADC phải được tắt trước khi vào chế độ ngủ bất kỳ. Khi ADC được tắt và bật lại, chuyển đổi tiếp theo sẽ là một chuyển đổi mở rộng.

Brown-Out Detector

Nếu ứng dụng không yêu cầu Brown-Out Detector (BOD), module này nên bị tắt. Nếu BOD được kích hoạt bởi BODLEVEL Fuses, nó sẽ được kích hoạt trong tất cả các chế độ ngủ, và do đó, luôn luôn tiêu thụ điện năng. Trong chế độ ngủ sâu hơn, điều này sẽ đóng góp đáng kể vào tổng mức tiêu thụ hiện tại.

WatchDog Timer (WDT)

Nếu WDT không cần thiết trong ứng dụng, thì module nên được tắt. Nếu WDT được kích hoạt, nó sẽ được kích hoạt trong tất cả các chế độ ngủ và do đó luôn luôn tiêu thụ điện năng. Trong chế độ ngủ sâu hơn, điều này sẽ đóng góp đáng kể vào tổng mức tiêu thụ hiện tại.

Port Pins

Khi vào chế độ ngủ, tất cả các port pins phải được cấu hình để sử dụng công suất tối thiểu. Điều quan trọng nhất là sau đó để đảm bảo rằng không có chân tải điện trở. Ở chế độ ngủ, cả clock I/O (clkI/O) và clock ADC (clkADC) đều bị dừng, input buffers của thiết bị sẽ bị tắt. Điều này đảm bảo rằng không có điện năng được tiêu thụ bởi logic đầu vào khi không

cần thiết. Trong một số trường hợp, logic đầu vào là cần thiết để phát hiện các điều kiện đánh thức và sau đó nó sẽ được kích hoạt.

Đối với các chân đầu Analog, digital input buffers sẽ bị vô hiệu hóa mọi lúc.

Việc tận dụng các chế độ ngủ tiết kiệm năng lượng cùng kết hợp với việc vô hiệu hóa các module không cần thiết trong ứng dụng là một trong những thủ thuật vô cùng quan trọng cũng như cần thiết giúp tiết kiệm năng lượng nhiều nhất của một mạch sử dụng vi điều khiển.

3.1.2.3 Công nghệ LoRa, giao tiếp - truyền dữ liệu giữa các node thu thập và node trung tâm

Lora là một công nghệ giao tiếp không dây được phát triển để tạo ra mạng WAN công suất thấp (low-power, wide-area networks – LPWANs) được ứng dụng cho các ứng dụng machine-to-machine (M2M) và IoT.

Một số đặc điểm chính của công nghệ LoRa:

- Tầm xa: >15 km / 9 mi
- Năng lượng thấp: tuổi thọ pin dự kiến từ 5-10 năm
- Chi phí thấp: Từ chi phí cảm biến ở end-node đến chi phí đầu tư cơ sở hạ tầng ban đầu
- Vị trí địa lý: cho phép theo dõi trong nhà / ngoài trời mà không có GPS
- Để đảm bảo cho việc truyền xa và năng lượng thấp thì cần phải sử dụng sóng có dải tần thấp (Sub-GHz).



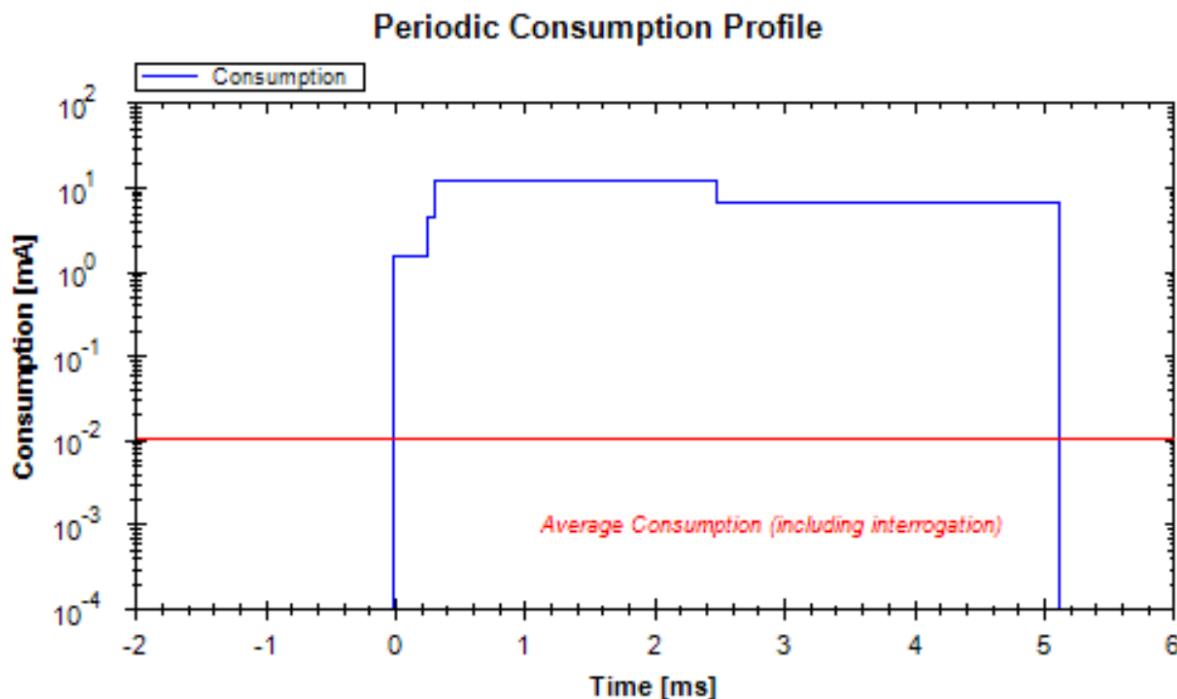
Hình 3.104. Measurement Node truyền dữ liệu đến Center Node sử dụng công nghệ Lora

LoRaWAN™ là một kỹ thuật LPWAN toàn cầu do LoRa Alliance™ tạo nên để phát triển một tiêu chuẩn duy nhất cho khả năng tương tác liền mạch trong cả công nghệ này. Công nghệ LoRa là lý tưởng cho các ứng dụng:

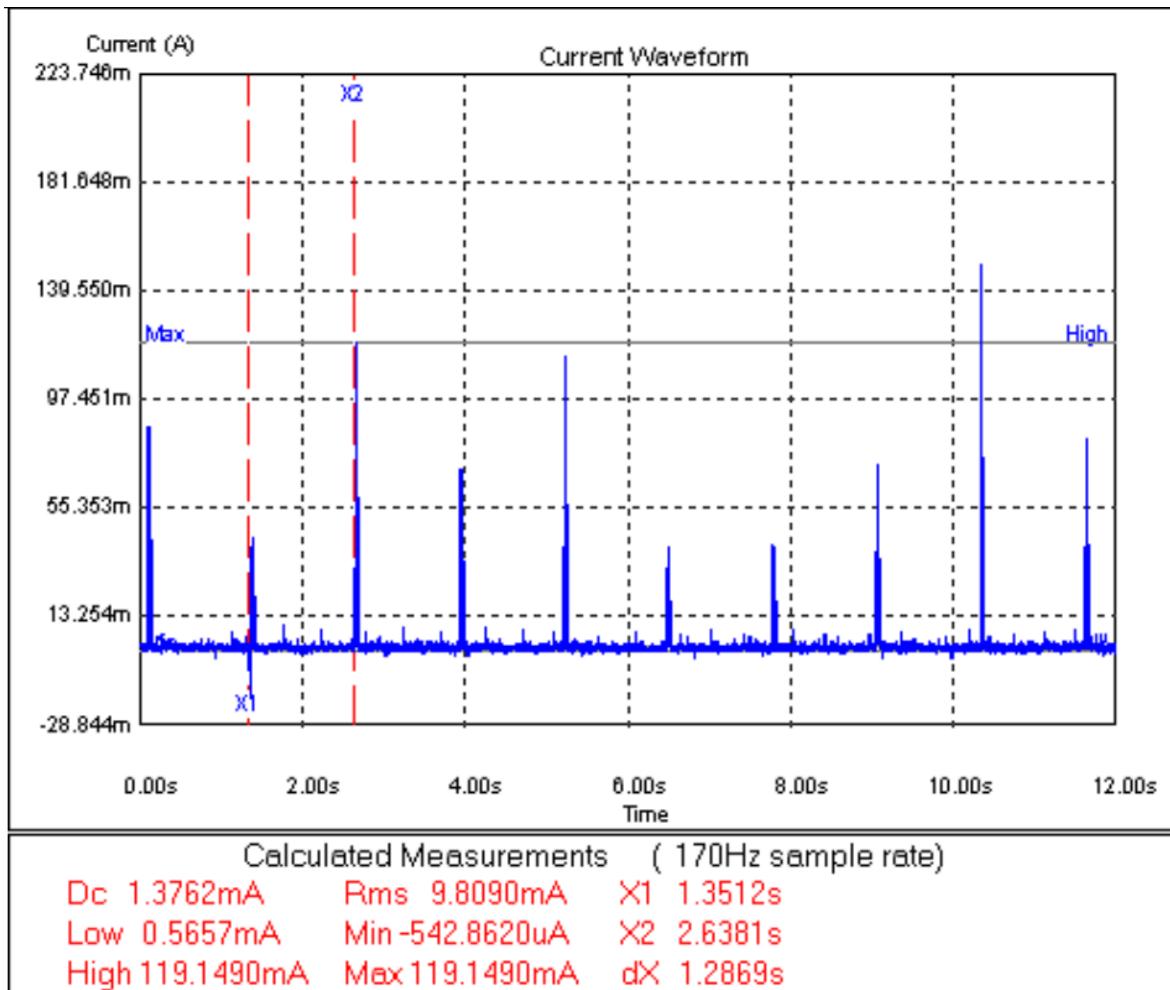
- Smart Cities

- Smart Home and Building
- Smart Environment
- Smart Metering
- Smart Agriculture
- Smart Industrial
- Retail and Logistics

So sánh năng lượng tiêu thụ khi sử dụng SX1272/3/6/7/8: LoRa Modem và Module SIM800_V1.08



Hình 3.11. Năng lượng tiêu thụ định kỳ (màu xanh) và năng lượng tiêu thụ trung bình của một modem LoRa^[10]



Hình 3.12. Dòng điện tiêu thụ của module SIM800 ở sleep mode^[11]

Dựa trên biểu đồ ở trên ta có thể dễ dàng nhận thấy dòng điện tiêu thụ trung bình của modem LoRa chỉ nằm trong khoảng 10^{-2} mA, dòng điện cao nhất chỉ nằm trong khoảng 10mA còn module SIM800 là 119mA, dòng điện hiệu dụng là 10mA.

Vì vậy quyết định lựa chọn công nghệ LoRa là một lựa chọn đúng đắn cho đề tài.

Module E32-TTL-100 (433T20D)

E32-TTL-100 (433T20D) là module thu phát không dây dựa trên SX1278 từ SEMTECH (có thể phát và nhận cùng lúc). Nó sử dụng công nghệ phổ rộng LoRa, với công nghệ này khoảng

cách truyền và khả năng xuyên thấu của module được cải thiện hơn 1 lần so với FSK truyền thống.



Hình 3.13. Module E32-TTL-100^[18]

Việc quyết định lựa chọn module E32-TTL-100 vì giá thành rất thấp so với module LoRa từ mBed được nêu ra ở trên và tính sẵn có của nó tại Việt Nam.

Định nghĩa các chân của module: ^[19]

Pin No.	Pin item	Pin direction	Pin application
1	M0	Input	Hoạt động kèm với M1 & quyết định 4 mode hoạt động của module.
2	M1	Input	Hoạt động kèm với M0 & quyết định 4 mode hoạt động của module.
3	RXD	Input	TTL UART inputs, kết nối với chân TXD của MCU, PC output.
4	TXD	Output	TTL UART outputs, kết nối với chân RXD MCU, PC input.
5	AUX	Output	Để chỉ ra trạng thái làm việc của module và đánh thức external MCU. Trong quá trình khởi tạo tự kiểm tra, pin có đầu ra mức thấp.
6	VCC	Power supply 2.3V~5.5V DC	
7	GND	Ground	

Các điểm đáng chú ý của chân AUX

No.	Notes for AUX
1	Khi AUX xuất ra mức thấp, điều đó có nghĩa là module đang bận và không thể tiến hành kiểm tra mode vận hành của module. Sau khi AUX xuất ra mức cao 1ms sau đó, nó sẽ hoàn thành nhiệm vụ chuyển đổi mode.
2	Sau khi chuyển sang chế độ hoạt động mới, nó sẽ không hoạt động ở chế độ mới ngay lập tức cho đến khi AUX tích cạn lên 2ms sau đó. Nếu AUX ở mức cao, mode vận hành mới có thể có hiệu lực ngay lập tức.
3	Khi người dùng chuyển sang mode hoạt động khác từ chế độ 3 (chế độ ngủ - sleep) hoặc vẫn đang trong quá trình reset, module

	sẽ đặt lại thông số người dùng, trong đó AUX có đầu ra ở mức thấp.
--	--

Các mode vận hành của module E32-TTL-100 được mô tả chi tiết dưới đây:

Mode thông thường (Mode 0)

Khi M1 = 0 & M0 = 0, module hoạt động ở mode 0	
Truyền	Gói dữ liệu được truyền từ module hoạt động ở chế độ 0 chỉ có thể được nhận bởi module hoạt động ở chế độ 0 hoặc 1.
Nhận	Module giữ chức năng nhận không dây luôn mở, nó có thể nhận gói dữ liệu được truyền từ module hoạt động ở mode 0 & mode 1. Sau khi nhận được gói dữ liệu, AUX xuất ra mức thấp, 5ms sau đó module bắt đầu truyền dữ liệu không dây qua serial port TXD pin

Mode Wake-up (Mode 1)

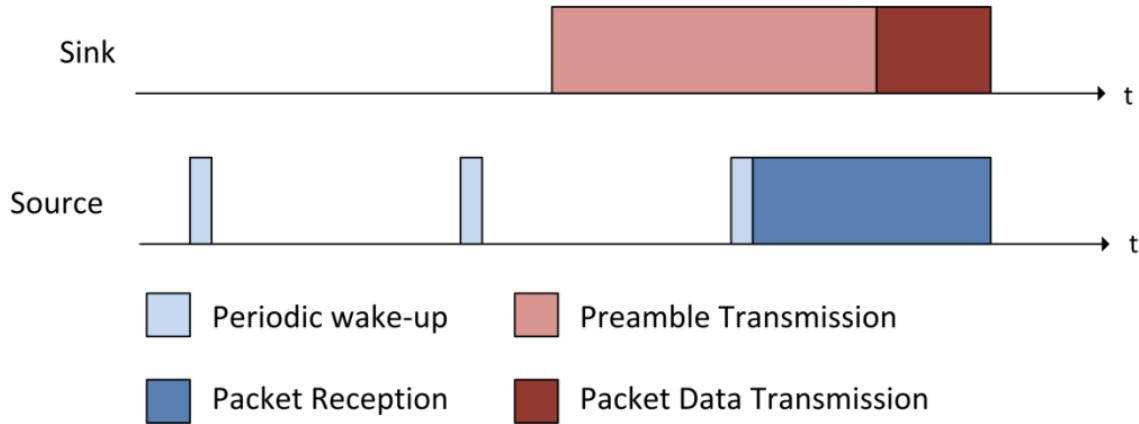
Khi M1 = 0 & M0 = 1, module hoạt động ở mode 1	
Truyền	Điều kiện truyền gói dữ liệu & chức năng chân AUX giống với chế độ 0. Sự khác biệt duy nhất là môđun sẽ thêm mã mở đầu (preamble) trước mỗi gói dữ liệu tự động. Độ dài mã mở đầu phụ thuộc vào thời gian đánh thức được thiết lập trong thông số người dùng. Mục đích của đoạn mã mở đầu là đánh thức module nhận được hoạt động ở mode 2. Do đó, gói dữ liệu được truyền từ mode 1 có thể nhận được theo mode 0, mode 1 và mode 2.
Nhận	Có chức năng giống như mode 0

Mode tiết kiệm năng lượng (Mode 2)

Khi M1 = 1 & M0 = 0, module hoạt động ở mode 2	
Truyền	UART bị đóng, module không thể nhận bất kỳ dữ liệu từ串行 port nào từ bên ngoài MCU. Do đó module hoạt động ở chế độ này không có chức năng truyền dẫn không dây
Nhận	Ở chế độ 2, yêu cầu bộ phát hoạt động ở mode 1. Module không day giám sát mã mở đầu, nó sẽ vẫn là trạng thái nhận và đợi cho đến khi hoàn thành bộ gói truyền. Sau đó, module cho phép AUX xuất ra mức thấp, 5ms sau đó sẽ mở cổng UART để truyền dữ liệu không dây đã nhận qua TXD. Cuối cùng AUX xuất đầu ra mức cao sau khi quá trình hoàn thành. Module không dây vẫn ở trạng thái làm việc “tiết kiệm năng lượng - giám sát” (polling). Bằng cách đặt thời gian đánh thức

	khác nhau, mô-đun có thể có độ trễ phản hồi nhận khác nhau (tối đa 2 giây) và mức tiêu thụ điện năng trung bình (tối thiểu 30uA). Người dùng cần đạt được sự cân bằng giữa thời gian trễ giao tiếp và công suất trung bình tiêu thụ.
--	--

Như đã đề cập ở trên, để giảm thiểu mức tiêu thụ, module được vận hành theo chu kỳ hạn chế. Hình 3.14 minh họa chu kỳ đó. Ở đây module nhận thức giấc theo chu kỳ với khoảng thời gian được xác định bởi độ dài của preamble được truyền bởi Sink. Khi tìm kiếm, module nhận và phát hiện phần preamble được truyền, nó được ‘đánh thức’ vào mode nhận và xử lý thông tin nhận được. Thiết bị này dành phần lớn thời gian của nó để thực hiện chu kỳ từ mode ngủ đến mode nhận và quay trở lại mode ngủ càng nhanh càng tốt để giảm thiểu mức tiêu thụ năng lượng.



Hình 3.14. Một ví dụ lấy mẫu Long-Preamble sử dụng chu kỳ cho quá trình nhận

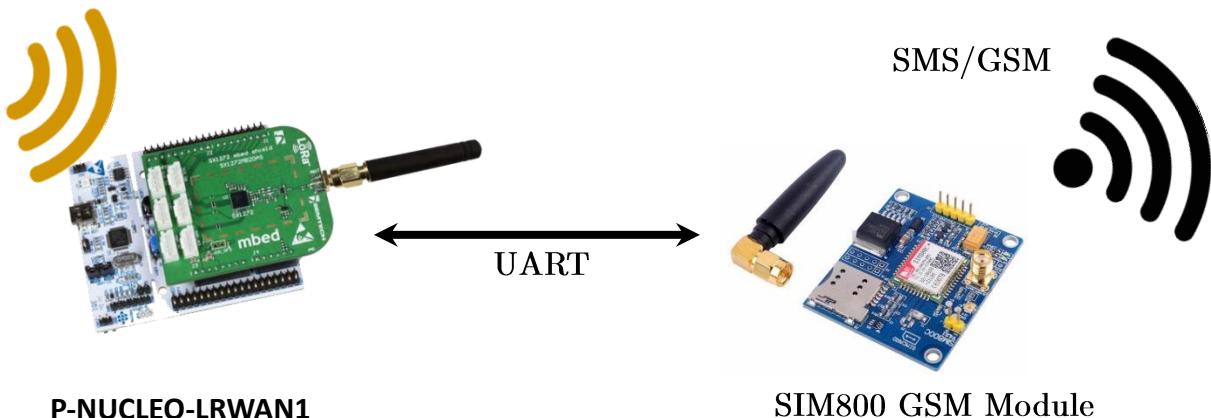
Mode Sleep (Mode 3). Được sử dụng cho việc cài đặt thông số cho module.

Khi M1 = 1 & M0 = 1 module hoạt động ở mode 3	
Truyền	N/A
Nhận	N/A

3.1.2.4 Node trung tâm - GateWay

Giai đoạn đề cương

Sau khi nhận dữ liệu từ các Measurement Node, Node trung tâm sử dụng giao tiếp uart truyền dữ liệu nhận được đến module SIM800 GSM để upload dữ liệu lên server thông qua GSM. Tại đây, nếu board mạch nhận được tín hiệu phát hiện độ mặn vượt quá hoặc thấp hơn một giá trị mặc định nào đó thì hệ thống sẽ gửi tin nhắn SMS cho một số số điện thoại nhất định.



Hình 3.15. Mô hình ở node trung tâm

Giao thức gửi dữ liệu lên server được sử dụng là HTTP GET.

Ngoài ra, GateWay còn nhận trách nhiệm nhận lệnh từ tin nhắn SMS như: yêu cầu thay đổi node id, thay đổi khoảng thời gian chu kỳ gửi dữ liệu của một node, yêu cầu giá trị tức thời của một node mà không cần phải đợi một chu kỳ, thay đổi giá trị ngưỡng của các node để báo động khi độ mặn vượt quá ngưỡng thiết lập.

Giai đoạn hiện thực luận văn:

Với những lý do tương tự được đề cập trong Mục 3.1.2.2 – phần 2, Arduino Mega đã được sử dụng thay thế cho gói P-NUCLEO-LRWAN1.



Hình 3.16. Arduino Mega 2560 Rev3 [20]

Arduino Mega 2560 R3 là phiên bản nâng cấp của Arduino Uno R3 với số chân giao tiếp, ngoại vi và bộ nhớ nhiều hơn, phù hợp cho các ứng dụng cần nhiều bộ nhớ hoặc nhiều chân, cổng giao tiếp hơn so với Arduino Uno.

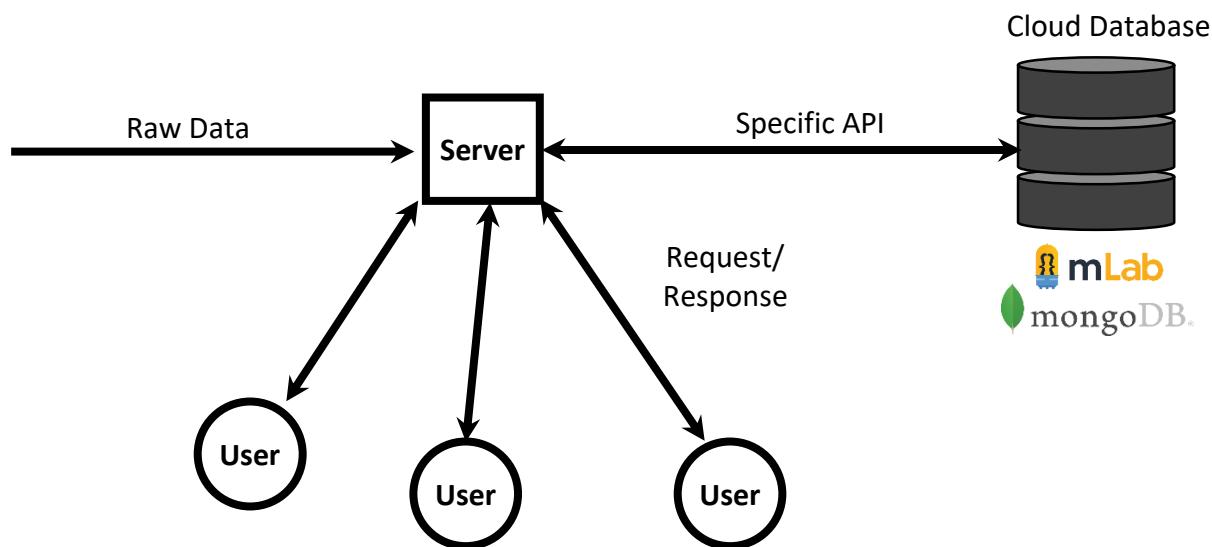
Thông số kỹ thuật

Microcontroller	Atmega2560
-----------------	------------

Operating Voltage	5 V
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
Clock Speed	16 MHz
Analog IN Pins	16
EEPROM	4 KB
DC Current per I/O Pins	20 mA
Input Voltage	7-12 V
Input Voltage (limit)	6 – 20 V
Digital I/O Pins	54 (of which 15 provide PWM output)
Weight	37 g

3.1.2.5 Server

Dữ liệu thô từ node trung tâm được truyền đến server qua internet. Tại đây dữ liệu được định dạng lại, gắn thêm thời gian thực, đóng gói và lưu trữ xuống database.



Hình 3.17. Mô hình ở Server cùng với Database và User

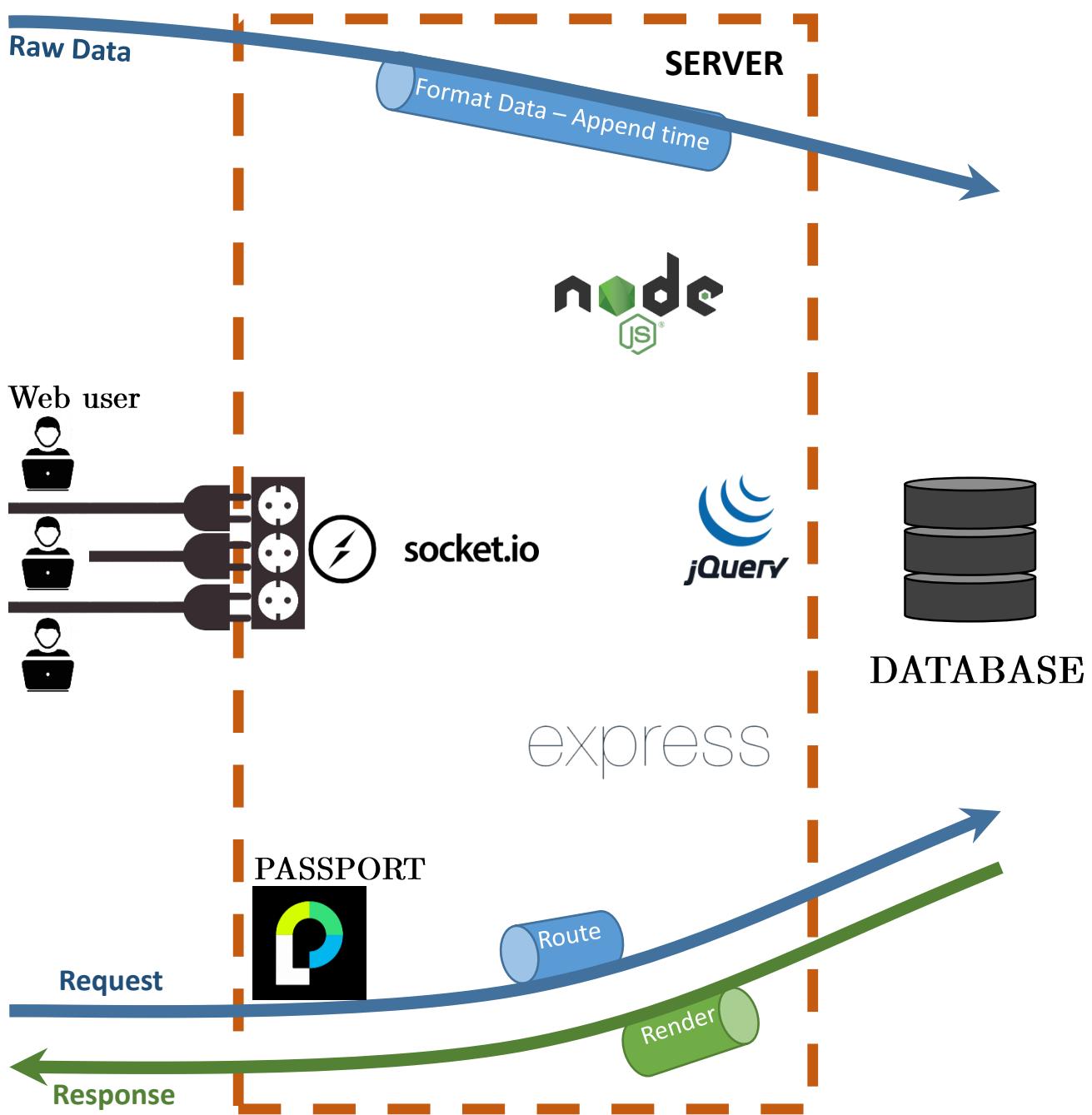
Database của hệ thống có thể sử dụng các DBMS thuộc SQL như Oracle, MySQL,... hay NoSQL như MongoDB, CouchDB,... Nhưng vì dữ liệu của các sensor được lưu dưới dạng datalog, nên việc sử dụng NoSQL sẽ có hiệu quả và đem lại hiệu suất cao hơn.

Việc sử dụng Database có thể được trực tiếp quản lý dưới local (máy chủ vật lý) nhằm đảm bảo tính bảo mật nhiều hơn nhưng đòi hỏi phải tốn chi phí lưu trữ và quản lý dữ liệu, hoặc ngoài ra, hệ thống có thể sử dụng một cloud database (Database-as-a-Service) để lưu trữ dữ liệu tại một cloud không nằm dưới local. Trong mô hình này, hệ thống sử dụng một cloud database với một gói miễn phí cho phép, sử dụng cho việc testing được cung cấp từ mLab^[34], mọi dữ liệu được xử lý dưới cơ sở dữ liệu của MongoDB^[31].

Khi một request (yêu cầu) được gửi đến server để truy cập vào trang web hoặc để lấy một dữ liệu bất kỳ. Server sẽ truy vấn dữ liệu từ database, sau đó response (phản hồi) trả về một trang web hoàn chỉnh (với phương thức “Render”) hoặc một dữ liệu dưới định dạng tùy thuộc vào mỗi yêu cầu của các user. Ngoài ra, nhằm đảm bảo tính bảo mật cũng như dễ dàng nhận biết để phân quyền cho mỗi user, server sử dụng một thư viện mang tên PassportJS, đúng như tên gọi của nó, yêu cầu user xuất trình “passport” của mình để xác thực quyền trước khi server bắt đầu phục vụ yêu cầu của user.

Server được viết bằng ngôn ngữ Javascript trên môi trường server NodeJS với framework ExpressJS, sử dụng phương pháp lập trình bất đồng bộ, đáp ứng nhanh đối với cùng lúc nhiều yêu cầu truy cập. Kết hợp với Socket.IO là một công cụ khiến cho việc đáp ứng trở nên nhanh hơn dựa trên một web socket được thiết lập mỗi khi web user kết nối đến server.

Khi một request (yêu cầu) được gửi đến server để truy cập vào trang web hoặc để lấy một dữ liệu bất kỳ. Server sẽ truy vấn dữ liệu từ database, sau đó response (phản hồi) trả về một trang web hoàn chỉnh (với phương thức “Render”) hoặc một dữ liệu dưới định dạng tùy thuộc vào mỗi yêu cầu của các user. Ngoài ra, nhằm đảm bảo tính bảo mật cũng như dễ dàng nhận biết để phân quyền cho mỗi user, server sử dụng một thư viện mang tên PassportJS, đúng như tên gọi của nó, yêu cầu user xuất trình “passport” của mình để xác thực quyền trước khi server bắt đầu phục vụ yêu cầu của user.



Hình 3.18. Tóm quát xử lý tại server

Node.js

Node.js là một nền tảng Server side được xây dựng dựa trên Javascript Engine (V8 Engine). Node.js sử dụng mô hình theo hướng sự kiện, non-blocking I/O, làm cho việc xử lý gọn nhẹ và hiệu quả hơn. Hệ sinh thái gói của Node.js, npm, là một hệ sinh thái lớn nhất của các thư viện mã nguồn mở trên thế giới.^{[21][22]}

Một cách đơn giản hơn, Node.js là^[23]:

- Một nền tảng cho phép chúng ta chạy chương trình Javascript trên một máy tính hoặc một máy chủ.
- Cho phép ta đọc, xóa, và chỉnh sửa tập tin.
- Dễ dàng trong việc giao tiếp với cơ sở dữ liệu.

Tại sao Node.js được sử dụng phổ biến:

- Sử dụng ngôn ngữ lập trình Javascript (là ngôn ngữ lập trình không thể thiếu dành cho hiện thực giao diện người dùng website – front-end), vì vậy lập trình viên có thể dễ dàng tận dụng nó cho server-side.
- Xử lý nhanh (chạy trên công vụ V8^[24], và sử dụng non-blocking code).
- Hệ sinh gói mã nguồn mở khổng lồ (npm).
- Thích hợp cho những dịch vụ thời gian thực (như ứng dụng chat), bởi vì ta có thể sử dụng WebSocket^[25] (sẽ được đề cập ở phần sau).

Điểm đáng chú ý của Node.js là việc sử dụng **phương pháp lập trình bất đồng bộ**.

Tác vụ khá phổ biến của một web server là đọc một file và trả về nội dung cho khách hàng.

Dưới đây là cách mà PHP hoặc ASP xử lý yêu cầu trên^[26]:

1. Gửi tác vụ đến hệ thống máy tính.
2. Chờ trong khi hệ thống mở và đọc tệp tin.
3. Trả về nội dung cho khách hàng.
4. Sẵn sàng cho việc xử lý yêu cầu tiếp theo.

Cách mà Node.js xử lý yêu cầu trên:

1. Gửi tác vụ đến hệ thống máy tính.
2. Sẵn sàng cho việc xử lý yêu cầu tiếp theo.
3. Khi hệ thống đã mở và đọc tệp tin, máy chủ trả về nội dung cho khách hàng.

Node.js loại bỏ sự chờ đợi trong quá trình xử lý và đơn giản là tiếp tục với yêu cầu tiếp theo. Đơn giản hơn, như một người bồi bàn sẵn sàng nhận đặt món của khách hàng tiếp theo thay vì chờ đợi món ăn của khách hàng hiện tại để phục vụ họ sau khi món ăn được nấu xong, sự chờ đợi này là lãng phí.

Express.js và mô hình MVC

Express.js, hoặc đơn giản là Express, là một framework ứng dụng web cho Node.js, được phát hành dưới dạng phần mềm miễn phí và mã nguồn mở. Nó được thiết kế để xây dựng các ứng dụng web và API. Express được gọi là framework máy chủ tiêu chuẩn thực tế cho Node.js.

Express là Node.js web framework phổ biến nhất và là thư viện cơ bản cho một số Node.js web framework khác. Nó cung cấp các cơ chế cho việc^[27]:

- Viết handler (trình xử lý) cho các requests (yêu cầu) với các phương thức HTTP khác nhau tại các đường dẫn URL khác nhau (route – định tuyến).
- Kết hợp với công cụ hiển thị “view” để tạo những phản hồi cho yêu cầu bằng cách chèn thêm data vào những template có sẵn.
- Có thể tùy chỉnh được các cài đặt phổ biến của một ứng dụng web như port sử dụng cho việc kết nối, thiết lập vị trí của các templates sử dụng cho việc phản hồi các request.
- Có thể thêm “middleware” bổ sung để xử lý các request trong chuỗi xử lý request này.

Tạo một máy chủ và lắng nghe bất kỳ loại HTTP request nào trên URL <http://localhost:3000/> trở nên vô cùng đơn giản với đoạn code dưới đây. Khi một yêu cầu được nhận, server sẽ gửi một phản hồi plain-text là “Hello World”.

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!');
});

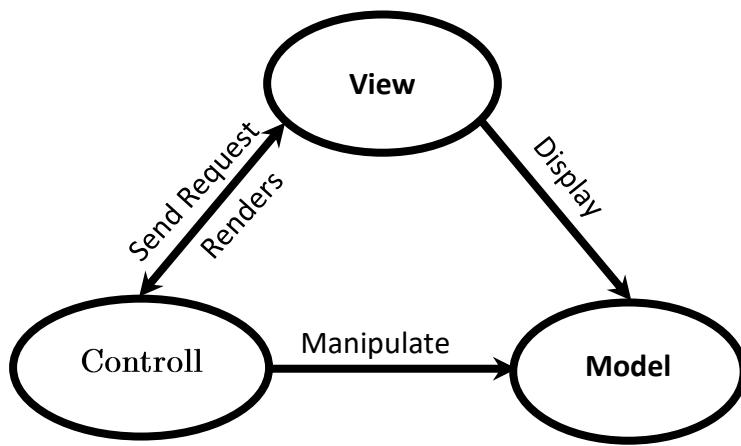
app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

Kiến trúc mô hình MVC (Model – View – Controller)^[28] tách ứng dụng thành 3 thành phần như tên gọi của nó.

Model: Model đại diện cho hình dạng của dữ liệu và logic nghiệp vụ. Nó duy trì và cung cấp dữ liệu cho ứng dụng. Những đối tượng (object) model truy xuất và lưu trữ trạng thái của Model trong một database.

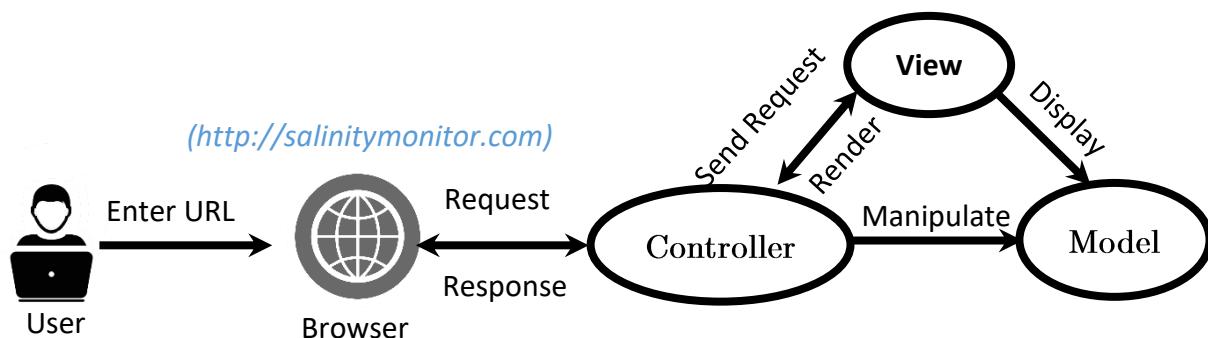
View: View là một giao diện người dùng. View hiển thị dữ liệu cho user và cũng cho phép họ chỉnh sửa dữ liệu thông qua Model.

Controller: Controller xử lý yêu cầu của người dùng. Điện hình, user tương tác với View, trong đó đưa lên những URL request thích hợp, những request này sẽ được xử lý bởi một controller. Controller hiển chỉ một view tương ứng kết hợp với dữ liệu của model dưới dạng một response (phản hồi).



Hình 3.19. Kiến trúc MVC

Hình dưới đây minh họa một luồng xử lý của một request trong mô hình MVC:



Hình 3.20. Request/Repose trong Kiến trúc MVC

Theo hình 3.20, khi người dùng nhập URL vào trình duyệt, nó sẽ đi đến server và gọi bộ Controller thích hợp. Sau đó, bộ Controller này sử dụng View và Model tương ứng để tạo ra những phản hồi đáp ứng đúng nhu cầu của người dùng và gửi lại nó cho họ.

Mô hình Web Server của hệ thống được xây dựng dựa trên kiến trúc MVC để dễ dàng quản lý và xử lý yêu cầu của user.

WebSocket – Socket.IO

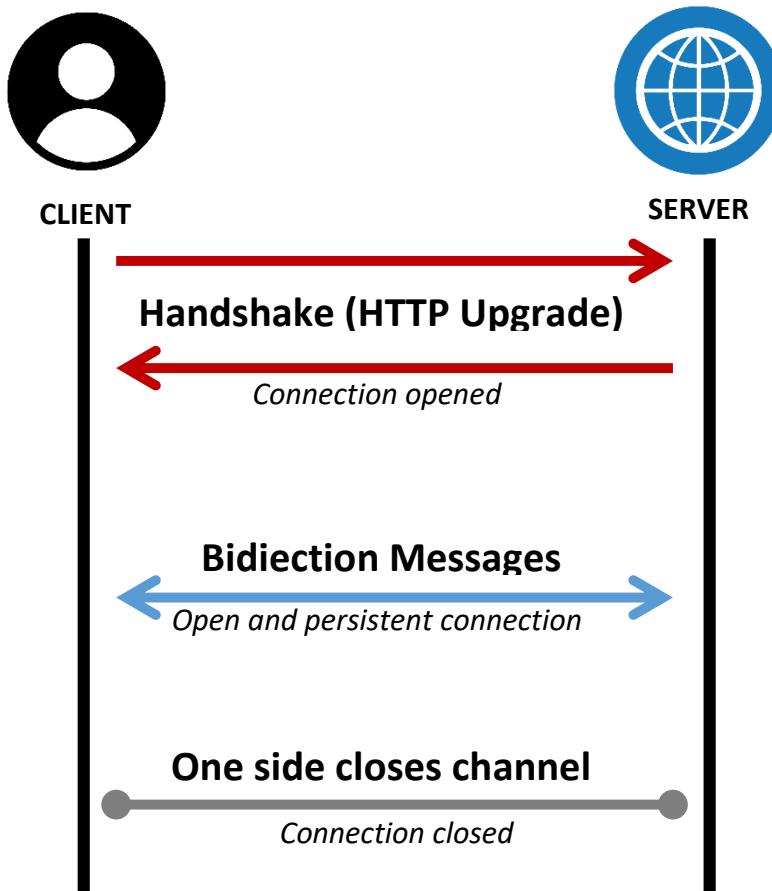
WebSocket là một giao thức truyền thông máy tính. Không giống như HTTP, WebSocket cung cấp giao tiếp full-duplex (song công toàn phần – có thể đồng thời truyền và nhận dữ liệu cùng một lúc). Ngoài ra WebSocket cho phép các luồng messages (thông điệp) được kèm trên TCP. TCP đơn lẻ làm việc với các luồng bytes vốn nó không phải là khái niệm của một message. Với lý do chi phí cho việc handshake (bắt tay giao tiếp) của TCP và phần header của HTTP là quá lớn, gây nên sự kém hiệu quả trong đối với các messages nhỏ. Giao thức WebSocket hướng đến việc giải quyết những vấn đề này mà không ảnh hưởng đến các giả định về sự bảo mật của web.^[25]

Bởi vì một kết nối WebSocket được giữ luôn mở trong suốt thời gian của một session (phiên làm việc), các messages có thể truyền qua lại giữa các bên tham gia vào socket với rất ít chi phí và độ trễ thấp. Điều này giúp loại bỏ đi việc polling (thăm dò) của cả client lẫn server. Và đặc biệt hữu ích cho bất kỳ ứng dụng nào dựa trên nền JavaScript mà xử lý dữ liệu một cách thường xuyên, bất đồng bộ cũng như các ứng dụng real time (thời gian thực).^[29]

Việc theo dõi dữ liệu độ mặn của hệ thống thủy lợi không đòi hỏi thời gian xử lý phải chính xác ở mức cao nhất nhưng việc có nhiều node dữ liệu cùng gửi lên Server đòi hỏi việc truyền dữ liệu có độ trễ tương đối thấp. Việc tận dụng WebSocket cũng làm giao thức kết nối giữa các user là khá hiệu quả.

Cách thức hoạt động của một WebSocket:

Để khởi tạo một kết nối WebSocket, máy khách và máy chủ phải trao đổi một handshake từ HTTP, vừ từ đó trở đi, các quy tắc bất đồng bộ WebSocket được áp dụng. Cụ thể, những headers song kềnh của HTTP được thay thế bằng khung messages chỉ có kích thước vài bytes, và cả Server lẫn Client đều có thể gửi dữ liệu mới một cách đồng thời mà không cần phải chờ yêu cầu từ bên kia.



Hình 3.21. Mô hình kết nối và truyền dữ liệu của một WebSocket.^[29]

Socket.IO – là một thư viện JavaScript cho các ứng dụng web thời gian thực. Nó cho phép Server và Client giao tiếp 2 chiều trên thời gian thực. Socket.IO được chia là 2 phần: một thư viện phía client chạy trong trình duyệt, và một thư viện phía server cho Node.js. Cả hai thành phần đều có API gần giống nhau vì đều là thư viện JavaScript. Như Node.js, Socket.IO là thư viện theo hướng sự kiện.

Với 2 snippet dưới đây tương ứng với 2 phần trong Socket.IO^[30]

Server (app.js)

```
var io = require('socket.io')(app);

io.on('connection', function (socket) {
    socket.emit('news', { hello: 'world' });
    socket.on('my other event', function (data) {
        console.log(data);
    });
});
```

Client (index.html)

```
<script src="/socket.io/socket.io.js"></script>
<script>
    var socket = io.connect('http://localhost');
    socket.on('news', function (data) {
        console.log(data);
        socket.emit('my other event', { my: 'data' });
    });
</script>
```

Một socket sẽ tạo ra giữa Web Client và Server, lúc này Server sẽ gửi một gói tin có tiêu đề là “news” đồng thời cũng lắng nghe khi có một gói tin với tiêu đề là “my other event”, khi Client nhận được và in ra tại console giá trị của data từ gói tin “news” là { hello: 'world' } , Client sẽ gửi một thông điệp có tiêu đề là “my other event” đến server với chính chính dữ liệu mà Server gửi đến và Server sẽ nhận lại dữ liệu sau đó in ra console.

Socket.IO chủ yếu sử dụng giao thức WebSocket với việc polling là một tùy chọn dự phòng. Mặc dù sử dụng Socket.IO đơn giản chỉ như một vỏ bọc của WebSocket nhưng thực chất nó cung cấp nhiều tính năng hơn, bao gồm broadcast tới cùng một lúc nhiều socket, lưu trữ dữ liệu riêng đối với mỗi client.

Việc sử dụng Socket.IO trên nền Node.js là một lựa chọn phù hợp và tương đối phổ biến hiện nay.

MongoDB

MongoDB là một cơ sở dữ liệu định hướng document (tài liệu), cross-platform (đa nền tảng và mã nguồn mở, một loại cơ sở dữ liệu NoSQL (một loại dữ liệu không bắt buộc phải tuân theo một lược đồ quan hệ đã được thiết lập trước như MySQL). Là một cơ sở dữ liệu NoSQL, MongoDB loại bỏ cấu trúc table (bảng) của cơ sở dữ liệu quan hệ để thích ứng với các document kiểu JSON, có các lược đồ động mà nó gọi là BSON.

Điều này giúp tập hợp dữ liệu cho một số loại ứng dụng nhất định nhanh hơn và dễ dàng hơn. MongoDB được xây dựng cho khả năng mở rộng, tính sẵn sàng cao và hiệu suất triển khai một máy chủ đơn lẻ đến cơ sở hà tầng lớn và phức tạp.^[31]

MongoDB đã trở thành một trong những cơ sở dữ liệu NoSQL phổ biến nhất, được sử dụng làm backend cho nhiều trang web chuyên nghiệp lớn như eBay, Craigslist, SourceForge và The New York Times.^[31]

Một số tính năng của MongoDB:

- Ad hoc queries (truy vấn ngẫu nhiên), hỗ trợ tìm kiếm theo field (trường), tìm kiếm regular expression (biểu thức chính quy), và truy vấn theo range (phạm vi).
- Indexing (chỉ mục) – bất kỳ field nào trong BSON document đều có thể được lập chỉ mục.
- Nhân rộng – Cung cấp tính sẵn sàng cao thông qua các bộ replica (bản sao) bao gồm 2 hay nhiều bản chép của dữ liệu gốc.
- Load balancing (Cần bằng tải) – sharding (một phương pháp phân phối dữ liệu trên nhiều máy vật lý) cho phép MongoDB mở rộng theo chiều ngang, có nghĩa là dữ liệu sẽ được phân phối và chia thành các ranges và rồi lưu trữ trong các phân đoạn khác nhau mà có thể được đặt trong các Server khác nhau.
- Aggregation (phương pháp truy vấn một lúc lớn nhiều dữ liệu) cung cấp nhiều phương pháp mạnh mẽ để truy cập và phân tích dữ liệu.
- Mô hình document của MongoDB ánh xạ tới các đối tượng trong code ứng dụng, khiến cho các lập trình viên dễ dàng làm việc với dữ liệu hơn.

```
var data_schema = new Schema({  
    node_id: String,  
    salinity: Number,  
    date: Date  
})
```

Snippet trên đưa ra một ví dụ về việc ánh xạ một schema (lược đồ) tới một collection (nhóm) và định nghĩa các field của các document trong collection đó. Collection “Data” gồm 3 field “node_id”, “salinity”, “date” lần lượt có giá trị thuộc kiểu “String”, “Number”, “Date”.

Sử dụng schema trên khiến cho việc định dạng dữ liệu, lấy dữ liệu và làm việc với nó trở nên dễ dàng hơn.

Các ứng dụng đang lưu trữ ngày càng nhiều dữ liệu, và tỷ lệ truy cập nó cũng ngày càng tăng cao. Nếu dữ liệu chỉ được lưu trữ lại một server đơn lẻ thì đến một lúc sẽ đạt đến giới hạn và ta cần phải mở rộng. MongoDB mở rộng quy mô bằng cách thêm vào nhiều server với nhiều dung lượng hơn mỗi khi cần thiết. *Dữ liệu độ mặn của hệ thống là liên tục và luôn được lưu trữ lại, việc dễ dàng linh động trong việc mở rộng và tính sẵn sàng cao kết hợp với modeling (mô hình hóa) dữ liệu khiến cho ứng dụng của hệ thống tăng tốc hơn, đáp ứng được với nhu cầu ngày càng cao hiện nay. Đồng thời aggregation của MongoDB tập hợp các nhóm giá trị từ các document với nhau, và có thể thực hiện hàng loạt các phép toán trên các dữ liệu đã nhóm để trả về một kết quả duy nhất cho phép các ứng dụng có thể phân*

tích dữ liệu và đưa cho người dùng những góc nhìn bao quát về các dữ liệu ứng dụng đã lưu trữ.

mLab

mLab là một dịch vụ quản lý hoàn toàn cơ sở dữ liệu đám mây, được dùng để hosts cơ sở dữ liệu MongoDB. mLab chạy trên các nhà cung cấp điện toán đám mây như Amazon, Google và Microsoft Azure và hiện đang hợp tác với các nhà cung cấp platform-as-a-service.

Hiện tại mLab cho phép sử dụng miễn phí để sử dụng với các ứng dụng nhỏ và dùng thử có dung lượng lưu trữ lên đến 0.5 GB đủ và đảm bảo cho thực hiện dự án.

Việc sử dụng hệ quản trị cơ sở dữ liệu từ mLab như một Database as a service (DBaaS – cơ sở dữ liệu dưới dạng dịch vụ). Là một mô hình dịch vụ điện toán đám mây cung cấp cho người dùng một dạng truy cập cơ sở dữ liệu mà không cần thiết lập phần cứng vật lý, cài đặt phần mềm, hoặc cấu hình cho hiệu suất. Tất cả các nhiệm vụ quản trị và bảo trì đều được nhà cung cấp dịch vụ quản lý, vậy nên tất cả việc người dùng hoặc chủ sở hữu ứng dụng cần làm là sử dụng cơ sở dữ liệu. Tất nhiên, nếu khách hàng muốn chọn cách để kiểm soát nhiều hơn cơ sở dữ liệu, tùy chọn này có sẵn và có thể khác nhau tùy thuộc vào nhà cung cấp. [32]

Các lợi ích của việc sử dụng DBaaS:

- DBaaS giúp người dùng tiết kiệm chi phí. Việc sử dụng DBaaS ít tốn kém hơn nhiều so với các dịch vụ nền tảng đám mây khác, vì việc mua phần mềm và phần cứng đắt tiền có thể là một chi phí đáng lo ngại cho các công ty thuộc mọi quy mô, các ứng dụng mọi kích thước. Chi phí bỏ ra thu lại các chức năng cần thiết thông qua DBaaS cho phép các công ty đầu tư vào các nguồn lực mà họ thực sự cần, trong khi không phải lo lắng về việc bảo trì cơ sở dữ liệu tại chỗ.
- Cung ứng nhanh chóng. Việc cung cấp cơ sở dữ liệu theo yêu cầu mất một khoảng thời gian ngắn hơn tính bằng hàng ngày hoặc hàng tuần, có thể mất để có một cơ sở dữ liệu vật lý xử lý cùng một thông tin. Điều này không chỉ làm giảm thời gian ra thị trường của ứng dụng, mà còn giúp tiết kiệm chi phí hơn.
- Outsourcing. Tận dụng toàn bộ lợi thế của DBaaS cũng có nghĩa là outsourcing việc quản trị và giám sát cơ sở dữ liệu. Sao lưu, phục hồi, điều chỉnh, tối ưu hóa, vá lỗi và nâng cấp, tất cả có thể được tự động hoặc xử lý bởi các chuyên gia.
- Nâng cao tính bảo mật. Việc duy trì từ xa cơ sở dữ liệu từ các máy chủ khác nhau mặc định tạo ra cơ chế bảo mật để ngăn chặn vi phạm an ninh. Cơ sở dữ liệu nằm ngoài trang web đồng nghĩa với việc ngăn chặn được các vi phạm tại chỗ.
- Theo dõi. Việc sử dụng cơ sở dữ liệu có thể dễ dàng được theo dõi khi sử dụng DBaaS. Tính toán chi tiết về việc sử dụng cơ sở dữ liệu có thể theo dõi thời gian sử dụng, không gian trống, đảm bảo tính khả dụng và mức tiêu thụ tài nguyên. Hơn nữa, nó có thể cung cấp chế độ xem theo kiểu dashboard của dữ liệu, cho phép tóm tắt ngắn gọn cho mỗi người dùng.

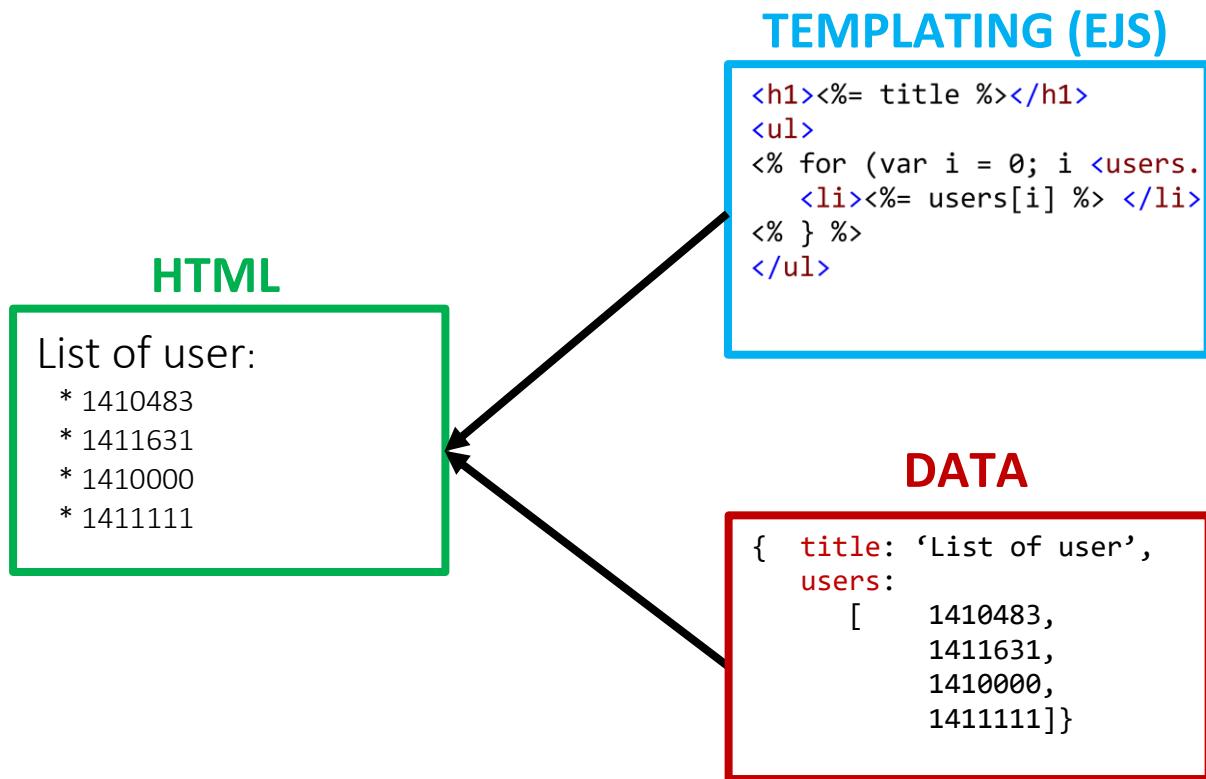
- Tiết kiệm nhân lực. Giải phóng khỏi lương công việc cho các lập trình viên và nhà phát triển của ứng dụng là một trong những lợi ích lớn nhất của DBaaS. Với một cơ sở dữ liệu tại chỗ, các Database Administrator (DBAs – người quản trị cơ sở dữ liệu) phải tập trung vào việc quản trị cơ sở dữ liệu thực tế. Với DBaaS, các lập trình viên và nhà phát triển được tự do tập trung vào dữ liệu ứng dụng và phát triển cơ sở dữ liệu, vì DBaaS cung cấp các dịch vụ quản trị cơ sở dữ liệu hoàn toàn ngoài một môi trường hoạt động của ứng dụng.
- Không gian máy chủ. Outsourcing cơ sở dữ thông qua DBaaS giải phóng rất nhiều không gian máy chủ. Tuy nhiên, điều này không có nghĩa là bạn không thể hoặc không nên sử dụng các máy chủ cũ. DBaaS cung cấp dịch vụ sao lưu dữ liệu được lưu trữ tại local, trong những trường hợp khẩn cấp.
- Khả năng mở rộng. DBaaS cho phép khả năng mở rộng theo nhu cầu, có nghĩa là khi một ứng dụng hoặc tổ chức sử dụng DBaaS tăng dung lượng cơ sở dữ liệu của họ, việc mở rộng chỉ cần đặt hàng với bên quản lý DBaaS. Nhưng với một cơ sở dữ liệu tại chỗ, các nhóm lập trình viên và cơ sở dữ liệu phải bổ sung thêm phần cứng vào cơ sở dữ liệu, mua và cài đặt thiết bị mới, tích hợp phần cứng đã cập. Không phải là một công việc đơn giản!

3.1.2.6 Web Client

EJS – Embedded JavaScript templating

EJS là một thư viện templating (mẫu) JavaScript. Nó thường được sử dụng để xây dựng chuỗi HTML từ những dữ liệu JSON.

Cách EJS hoạt động :



Hình 3.22. EJS tạo một khuôn mẫu cho HTML

EJS được xây dựng dựa trên cú pháp ngôn ngữ thuần JavaScript, với những cú pháp đơn giản bằng việc nhúng code Javascript vào trong các thẻ scriptlet. Sử dụng EJS khiến cho công việc đưa dữ liệu tới người dùng trở nên đơn giản, linh động và nhanh gọn hơn.

Các plugin và thư viện cho JavaScript

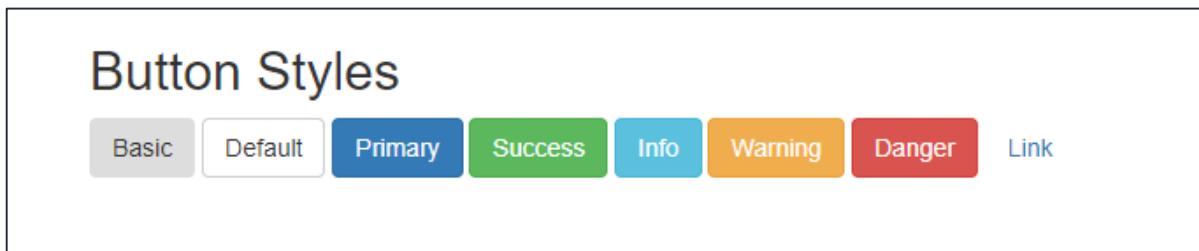
Bootstrap

Bootstrap là một bộ công cụ mã nguồn mở để phát triển với HTML, CSS và JavaScript.

Bootstrap là framework HTML, CSS và JavaScript phổ biến nhất để phát triển các trang web responsive, mobile-first.

```
<div class="container">
  <h2>Button Styles</h2>
  <button type="button" class="btn">Basic</button>
  <button type="button" class="btn btn-default">Default</button>
  <button type="button" class="btn btn-primary">Primary</button>
  <button type="button" class="btn btn-success">Success</button>
  <button type="button" class="btn btn-info">Info</button>
  <button type="button" class="btn btn-warning">Warning</button>
  <button type="button" class="btn btn-danger">Danger</button>
</div>
```

Với đoạn snippet đơn giản trên, chỉ bằng việc sử dụng framework bootstrap, trang HTML có thể tạo nên được như sau:



Jquery

Jquery là một thư viện JavaScript nhanh, nhỏ và giàu tính năng. Nó làm cho những thứ như truy vấn những dữ liệu từ trang HTML hoặc ngược lại, thao tác, xử lý sự kiện, animation (hoạt ảnh), và Ajax đơn giản hơn nhiều với một API dễ sử dụng, có thể hoạt động trên vô số các trình duyệt web. Kết hợp của tính linh hoạt và khả năng mở rộng, JQuery đã thay đổi hàng triệu lập trình viên sử dụng JavaScript. [3]

```
<button id="show">Show</button>
<script>
  $(document).ready(function () {
    $("#show").click(function () {
      alert("You've just clicked");
    });
  });
</script>
```

Với một đoạn snippet trên, khi người dùng click vào nút show, dòng chữ "You've just clicked" sẽ hiện thị trên browser và thông báo cho họ.

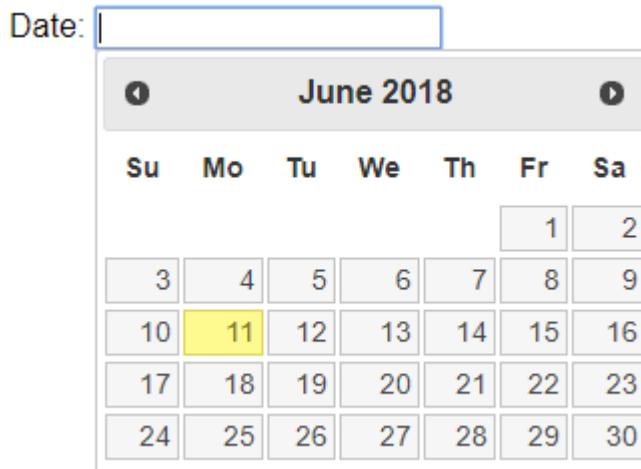


Datepicker – jQuery UI

jQuery UI là tập hợp các giao diện, hiệu ứng, tiện ích và chủ đề được xây dựng trên nền thư viện jQuery JavaScript. Việc xây dựng các ứng dụng web tương tác cao hay chỉ cần thêm một bộ datepicker (bộ chọn ngày) vào một biểu mẫu, jQuery UI sẽ dễ dàng thực hiện nó.

```
<p>Date: <input type="text" id="datepicker"></p>
<script>
    $(function () {
        $("#datepicker").datepicker();
    });
</script>
```

Với đoạn snippet trên, việc tạo một biểu mẫu để thuận tiện hơn cho việc chọn ngày tháng sẽ có giao diện như sau.



gmaps.js

gmaps.js là một thư viện JavaScript sử dụng những API từ Google Maps một cách đơn giản. Việc tạo một bản đồ để nhúng vào trang web:

chỉ đơn giản với việc gọi hàm:

```
new GMaps({  
  div: '#map',  
  lat: -12.043333,  
  lng: -77.028333  
});
```

Sử dụng Google Maps khiến cho vị trí của các Node trở nên trực quan hơn đối với người dùng.



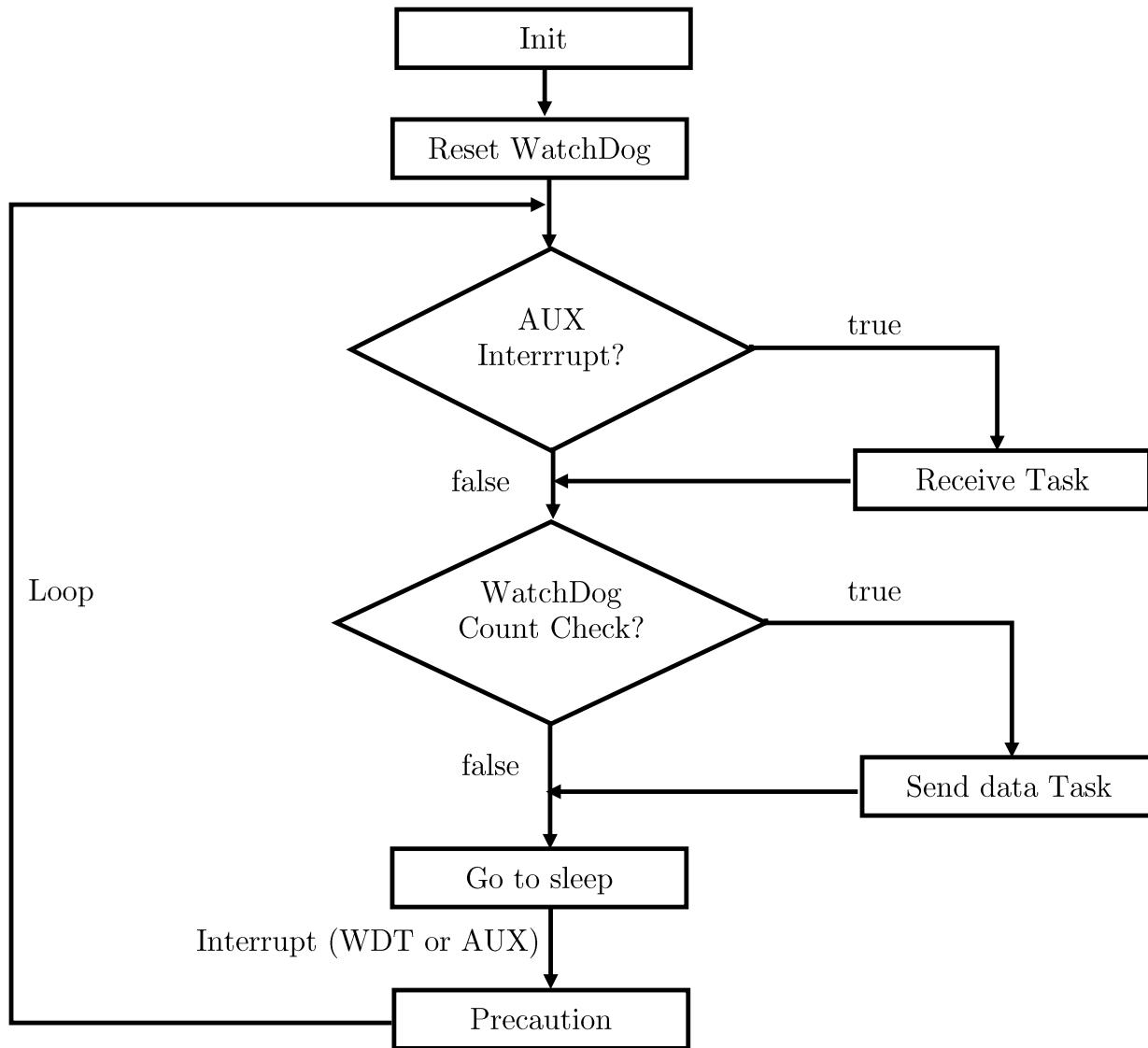
Chart.js

Để biểu diễn biểu đồ, đồ thị đặc trưng cho giá trị độ mặn theo thời gian thực, sử dụng thư viện mã nguồn mở giúp việc vẽ biểu đồ trở nên đơn giản, thuận tiện hơn. Chart.js tuy chỉ thực hiện những biểu đồ đơn giản đòi hỏi các nhà thiết kế và phát triển nhưng vô cùng linh hoạt, là một lựa chọn phù hợp trong dự án.

3.2 Thiết kế chi tiết

3.2.1 Node cảm biến

3.2.1.1 Flow chart



Hình 3.23. Flow chart thiết kế ở các node cảm biến

Theo flow chart trên, khi chương trình bắt đầu chạy, các dòng xử lý nhỏ được miêu tả rõ hơn dưới đây:

Init (initialization): các biến toàn cục được khởi tạo, đặc biệt <node_id> và <interval> là 2 biến cần phải được giữ lại khi không cấp nguồn cho MCU, vì vậy, vùng nhớ của EEPROM (hỗ trợ bởi ATMega328P) được sử dụng để lưu trữ hai biến này, xác nhận thành

công giao tiếp giữa module LoRa và vi điều khiển, cấu hình các thanh ghi cần thiết và các port pins của vi xử lý. Sau khi khởi tạo thành công, để báo hiệu cho người dùng biết mạch hoạt động thành công, một led được phát sáng trong 2 giây.

Reset WatchDog: Việc sử dụng chế độ Sleep power-down của MCU sẽ disable tất cả các timer nên việc tính toán chu kỳ cho việc gửi dữ liệu thường xuyên phải sử dụng một timer riêng là WatchDog (có thể sử dụng được khi ở chế độ Sleep power-down của MCU).

Receive Task: Khi nhận được tín hiệu LOW từ chân AUX (tham khảo tại phần 3.1.2.3). Điều này cho thấy việc module LoRa vừa nhận được một gói tin nào đó và muốn gửi nó qua UART cho vi điều khiển. Tại đây, vi điều khiển sẽ bị đánh thức bởi interrupt, interrupt routine được sử dụng, một biến global dành cho chân AUX được set thành giá trị boolean true. Trong vòng lặp loop, khi biến global dành cho chân AUX mang giá trị true, MCU bắt đầu quá trình đọc gói tin nhận được từ module LoRa, kiểm tra nội dung của gói tin (định dạng, nội dung, đích đến có đúng với prototype hoặc gói tin có yêu cầu node đang chạy hay không), sau khi thực hiện các tác vụ nhận, nếu gói tin là request từ gateway tới node nhận được, node sẽ thực hiện response đúng theo yêu cầu của request đó như gửi ACK, hoặc gửi dữ liệu độ mặn về cho gateway. Kết thúc Receive Task, biến global dành cho chân AUX được set lại thành giá trị false.

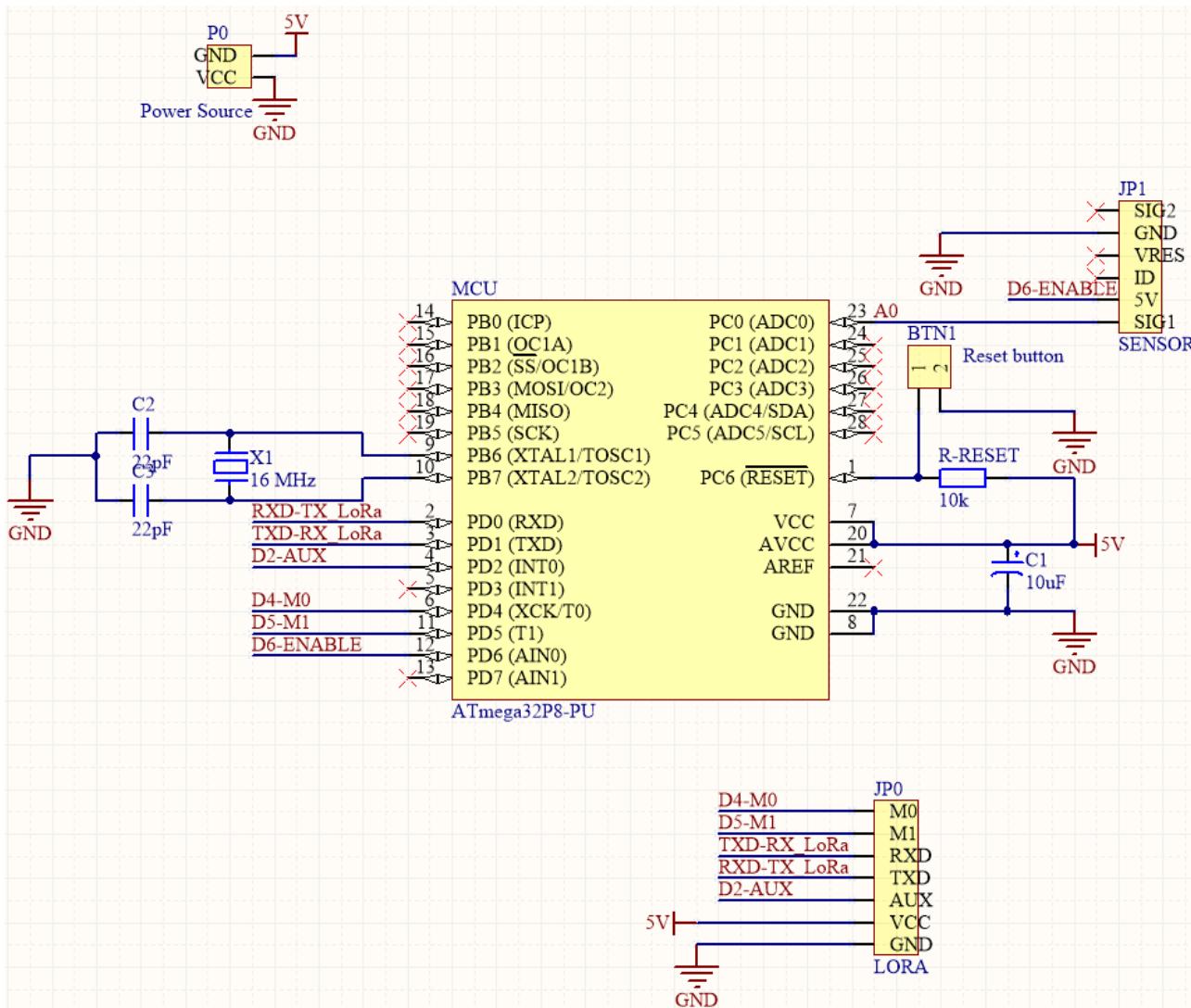
Send Data Task: Mỗi khi WDT tạo nên một interrupt theo một chu kỳ được cấu hình trước, một biến global được khởi tạo ban đầu dành riêng cho WatchDog sẽ được tăng thêm 1 đơn vị, khi giá trị của biến này nhân với thời gian một chu kỳ ngắn của WatchDog lớn hơn hoặc bằng thời gian interval giữa hai lần gửi dữ liệu lên GateWay, Send Data Task được thực hiện, đồng thời biến global dành riêng cho WatchDog sẽ được reset về 0. Trong quá trình đọc dữ liệu, vì sensor độ mặn Salinity Vernior đề cập tại phần 3.1.2.2 chỉ có thể đọc đúng giá trị sau khi enable trong 10 giây, vì vậy trong lúc này, MCU sẽ tiếp tục Sleep Power-down trong khi vẫn để chân enable của sensor được tích cực.

Go to Sleep: MCU quay trở lại với chế độ Sleep Power-down, nhưng để tối ưu về mặt sử dụng năng lượng vì hầu như tất cả thời gian hoạt động của MCU đều nằm trong khoảng thời gian này, các module ngoại vi, bắt đồng bộ sẽ bị disable trong lúc ngủ như: SPI, I2C, ADC, ..., đồng thời interrupt sẽ được bật enable, interrupt dành cho chân kết nối với AUX của LoRa sẽ được cấu hình để đánh thức MCU khi có một gói tin gửi đến.

Precaution: Sau khi được đánh thức dậy, MCU cần phải thực hiện những tác vụ đặc biệt để nó có thể quay lại hoạt động bình thường mà không bị gián đoạn như enable những module bắt đồng bộ, ngoại vi cần được sử dụng, tắt bỏ interrupt, set các biến global,... Những công việc này được thực hiện tại Precaution, ngay sau khi MCU thực hiện xong interrupt routine.

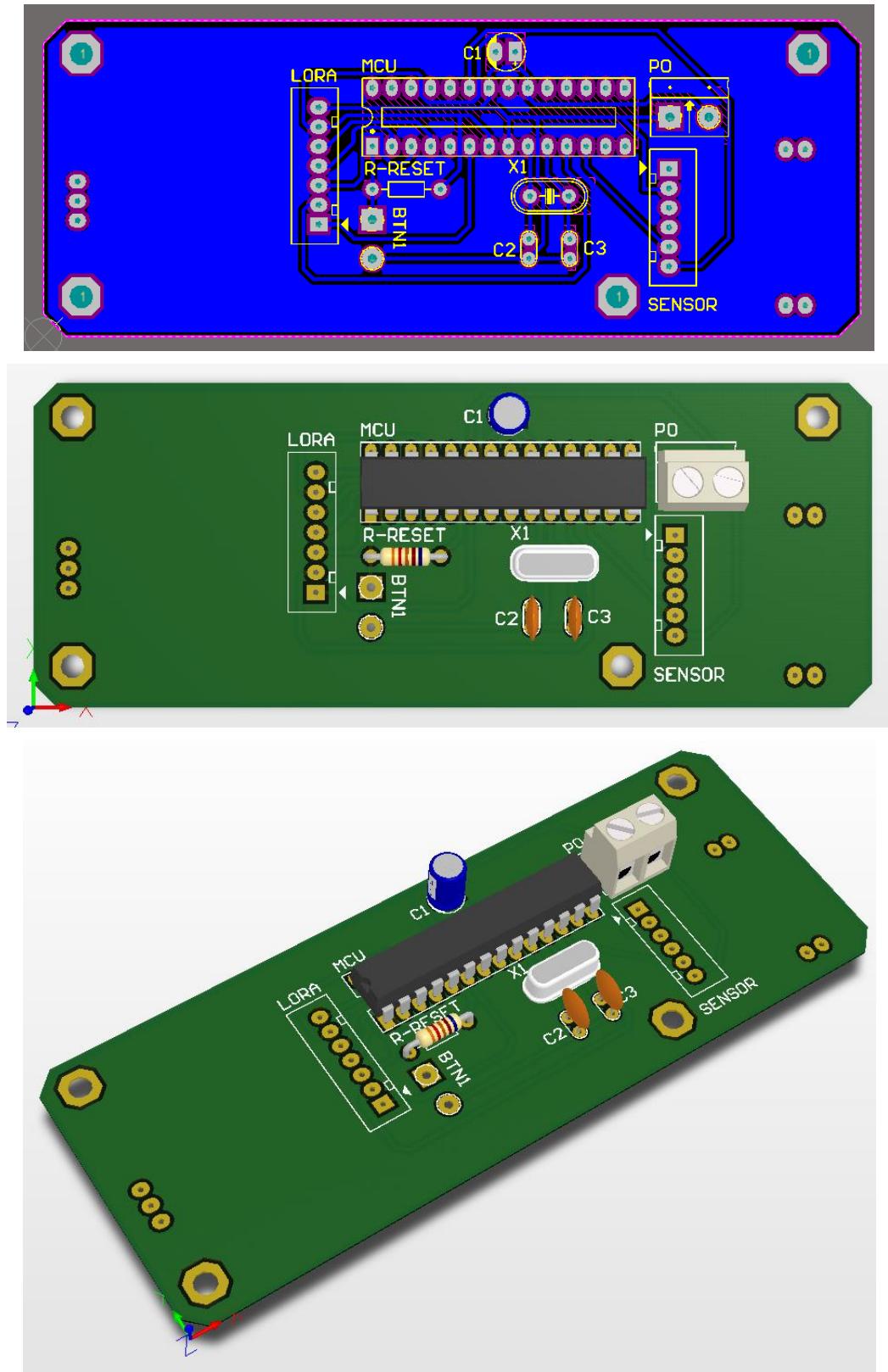
Ngoài việc sử dụng WDT làm timer để định kỳ thời gian gửi dữ liệu, WDT còn có tác dụng giám sát treo, nếu biến global của WDT vì một lý do nào đó không reset về 0 sau một khoảng thời gian xác định trước, MCU sẽ được reset và chạy lại từ đầu.

3.2.1.2 Schematic và board mạch



Hình 3.24. Schematic thiết kế tại node cảm biến

Trong sơ đồ mạch điện được thể hiện ở trên, mọi linh kiện điện tử không cần thiết được sử dụng trong Arduino Nano được loại bỏ thay vào đó chỉ là những phần tử cốt lõi là vi xử lý ATMega328P như thạch anh, các tụ điện, điện trở nối nguồn cho chân Reset, nút nhấn nối đất cho chân Reset để giảm lượng tiêu thụ điện năng đến mức tối thiểu. Các chân được nối tương ứng giữa MCU với module LoRa E32-TTL-100 và chân của sensor đo độ mặn. Chân AUX của module LoRa được nối với chân ngắt ngoài INT0 (D2 tương ứng với Arduino), chân cấp nguồn 5v cho sensor được nối thẳng với một chân digital (D6) của MCU với đầu ra là high khi enable.



Hình 3.25. Board mạch và mô hình 3D thiết kế tại node cảm biến

Công việc hàn mạch được tiến hành thủ công bằng tay để dễ kiểm tra và kiểm soát lỗi trong giai đoạn hiện thực nên các linh kiện cắm được sử dụng thay vì các linh kiện dán để việc làm bằng tay và chỉnh sửa trở nên dễ dàng hơn. Vì vậy kích thước board mạch trở nên lớn hơn rất nhiều, chiều dài: 106,68 mm, chiều rộng: 41,148 mm. Ngoài ra còn có các lỗ cắm đặc biệt để cố định những chân để giữ module LoRa, Sensor và bắt ốc vít cho board đính với vỏ hộp bên ngoài.

3.2.1.3 Prototype data truyền và nhận tại node cảm biến

Cấu trúc gói tin được truyền qua Lora tại node cảm biến được thiết kế như sau:

[TYPE;NODEID;DATA]

Trong đó:

TYPE ∈ [DATA; ACK; CHANGEID; CHANGEINTERVAL]

NODEID: Node Id

DATA: Dữ liệu đi kèm với gói tin

Ví dụ: [DATA;6;43.590]

- Khi Node cảm biến nhận gói tin:

TYPE: DATA

NODEID: <node_id> của node được yêu cầu gửi giá trị.

DATA: “True”.

Mô tả: Gói tin từ GateWay yêu cầu đọc giá trị sensor và trả về ngay lập tức.

TYPE: CHANGEID

NODEID: <node_id> của node được yêu cầu thay đổi id.

DATA: <node_id> mới yêu cầu thay đổi.

Mô tả: Gói tin từ GateWay yêu cầu thay đổi id của node.

TYPE: CHANGEINTERVAL

NODEID: <node_id> của node được yêu cầu thay đổi id.

DATA: <interval> chu kỳ gửi dữ liệu tới GateWay của Sensor Node.

Mô tả: Gói tin từ GateWay yêu cầu thay đổi chu kỳ gửi dữ liệu của Sensor Node.

Gói tin nhận được không đúng format hoặc không phải gói tin yêu cầu từ mình thì node cảm biến sẽ bỏ qua và không xử lý nó.

- Khi Node cảm biến gửi gói tin:

TYPE: ACK

NODEID: <node_id>

DATA: “True”

Mô tả: Gói tin từ Sensor Node phản hồi cho GateWay về việc tác vụ từ GateWay yêu cầu thay đổi node_id đã thành công.

TYPE: ACK

NODEID: <node_id>

DATA: <interval>

Mô tả: Gói tin từ Sensor Node phản hồi cho GateWay về việc tác vụ từ GateWay yêu cầu thay đổi chu kỳ gửi đã thành công.

TYPE: DATA

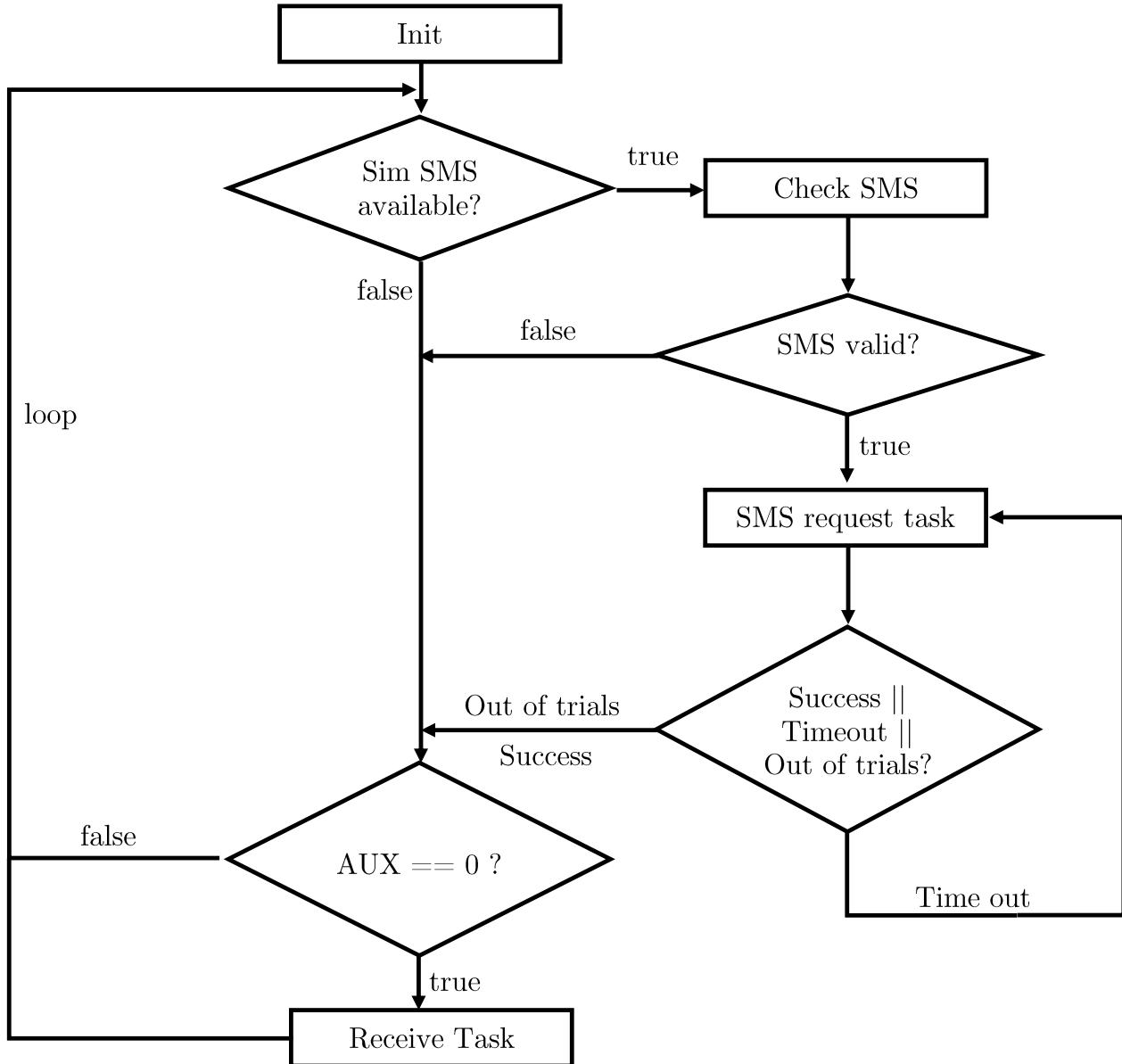
NODEID: <node_id>

DATA: Giá trị độ mặn

Mô tả: Gói tin từ Node cảm biến gửi giá trị độ mặn của node tới GateWay.

3.2.2 Node trung tâm

3.2.2.1 Flow chart thiết kế ở Node trung tâm – GateWay



Hình 3.26. Flow chart thiết kế ở GateWay

Theo Flow chart trên, khi chương trình bắt đầu chạy, các dòng xử lý nhỏ được miêu tả rõ hơn dưới đây:

Init (initialization): các biến toàn cục được khởi tạo, xác nhận thành công giao tiếp giữa module LoRa và module Sim800 với vi điều khiển, cấu hình các thanh ghi cần thiết và các port pins của vi xử lý, cấu hình cho module Sim. Sau khi khởi tạo thành công, để báo hiệu

cho người dùng biết mạch hoạt động thành công, một cuộc gọi sẽ được thực hiện đến số điện thoại đã được cài đặt mặc định từ trước.

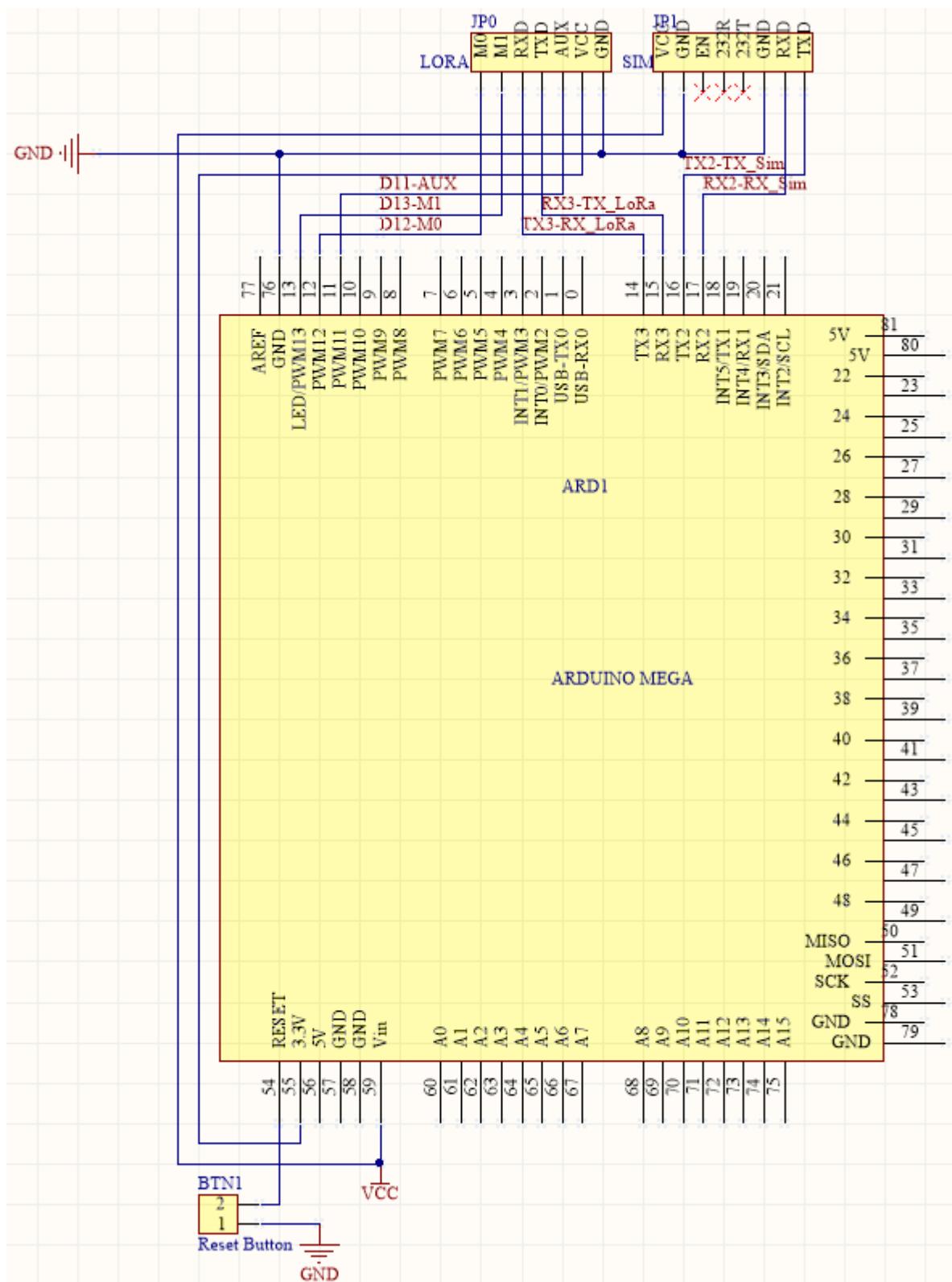
Check SMS: Khi một tin nhắn SMS được gửi tới, module Sim phát một tín hiệu thông báo đến MCU qua giao tiếp UART. MCU sẽ thực hiện việc đọc tin nhắn mới đến và kiểm tra format của nội dung, nếu nó mang một định dạng đúng theo mô tả tại phần 3.2.2.4 thì yêu cầu từ tin nhắn sẽ được thực hiện tại SMS request task. Ngược lại, nội dung của SMS không đúng định dạng thì flow sẽ quay lại vòng lặp.

SMS Request Task: Khi tin nhắn SMS đọc được đúng định dạng, dựa theo loại yêu cầu mà GateWay thực hiện tác vụ khác nhau bao gồm: yêu cầu nhận dữ liệu độ mặn thời điểm hiện tại của node, yêu cầu thay đổi id của các node, yêu cầu thay đổi giá trị ngưỡng báo động tại các node, yêu cầu thay đổi interval của các node. Các request được GateWay gửi tới các node cảm biến và bắt đầu chờ đáp trả từ node hoặc server, trong thời gian này, GateWay sẽ thực hiện các tác vụ khác, sau một khoảng thời gian cài đặt trước mà không nhận được phản hồi từ node hoặc server (timeout xảy ra), GateWay lại tiếp tục gửi thêm một request nữa, số lần gửi thêm được giới hạn bởi cài đặt của người dùng. Ngay cả khi request thực hiện có thành công hay không, GateWay sẽ thông báo lại cho chủ nhân tin nhắn về kết quả của thực hiện của yêu cầu này.

Receive Task: Khi nhận được tín hiệu LOW từ chân AUX (tham khảo tại phần 3.1.2.3). Điều này cho thấy việc module LoRa vừa nhận được một gói tin nào đó và muốn gửi nó qua UART cho vi điều khiển. Tại đây, vi điều khiển của GateWay không nằm trong trạng thái ngủ như các node cảm biến, vì vậy việc đọc giá trị gửi từ module LoRa được phần cứng thực hiện và lưu vào buffer, vi điều khiển chỉ cần thực hiện việc đọc dữ liệu có trong buffer đó và xử lý, kiểm tra nội dung của gói tin (định dạng, nội dung, đích đến có đúng với prototype hoặc gói tin có yêu cầu node đang chạy hay không), sau khi thực hiện các tác vụ nhận, nếu gói tin là các đáp ứng, dữ liệu gửi từ node nhận, GateWay sẽ đưa dữ liệu đó lên server hoặc đáp ứng lại bằng SMS.

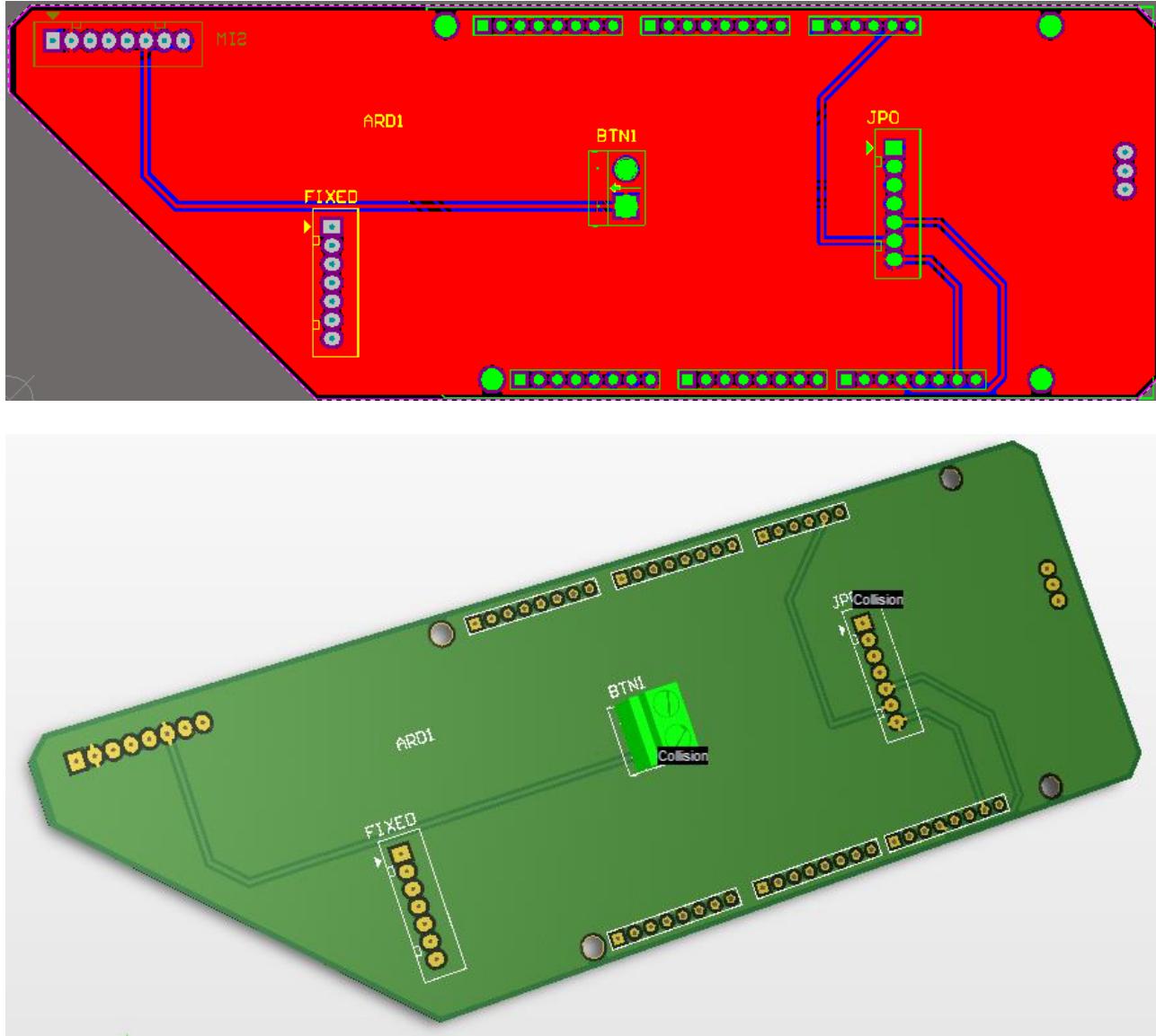
WDT cũng được GateWay sử dụng trong việc giám sát treo hoặc MCU bị rơi vào trạng thái vòng lặp vô tận. Nếu biến global của WDT vì một lý do nào đó không reset về 0 sau một khoảng thời gian xác định trước, MCU sẽ được reset và chạy lại từ đầu.

3.2.2.2 Schematic và board mạch thiết kế tại node trung tâm – GateWay



Hình 3.27. Schematic thiết kế tại node cảm biến

Trong sơ đồ được thể hiện ở trên, mạch điện được vẽ đóng vai trò như một interface kết nối giữa Arduino Mega, module LoRa E32-TTL-100, và module SIM800 kết hợp thêm một nút nhấn nối đất để reset khi cần thiết. Các chân được nối tương ứng giữa MCU với module LoRa E32-TTL-100 và Sim800. Arduino Mega hỗ trợ đến 3 bộ hardware UART, vì thế việc sử dụng module LoRa lẫn Sim800 kết hợp vẫn được xử lý dễ dàng. Điều đặc biệt đáng chú ý là 2 chân RX và TX bởi bộ UART của Sim800 được thiết kế ngược nên TxD của Sim sẽ được nối với TxD của Arduino Mega, tương tự với chân RXD.



Hình 3.28. Board mạch và mô hình 3D thiết kế tại GateWay

Kích thước board mạch, chiều dài: 156,845 mm, chiều rộng: 53,594 mm. Vì GateWay sử dụng nguồn được cung cấp bên ngoài là 9V và thông thường được đặt trong nhà, với nguồn

điện áp luôn có sẵn, do đó việc kích thước lớn không phải là vấn đề lớn đối với GateWay. Ngoài ra còn có các lỗ cắm đặc biệt để cố định những chân để giữ module LoRa, Sim và bắt ốc vít cho board đính với vỏ hộp bên ngoài.

3.2.2.3 Prototype data truyền và nhận tại GateWay

Cấu trúc gói tin được truyền qua Lora tại GateWay được thiết kế như sau:

[TYPE;NODEID;DATA]

Trong đó:

TYPE ∈ [DATA; ACK; CHANGEID; CHANGINTERVAL]

NODEID: Node Id

DATA: Dữ liệu đi kèm với gói tin

Ví dụ: [DATA;5;true]

- Khi GateWay nhận gói tin:

TYPE: DATA

NODEID: <node_id> của node gửi.

DATA: giá trị độ mặn.

Mô tả: Gói tin giá trị độ mặn nhận được từ Node cảm biến

TYPE: ACK

NODEID: <node_id> của node được yêu cầu thực hiện tác vụ.

DATA: “true”

Mô tả: Gói tin từ Sensor Node phản hồi cho GateWay về việc tác vụ từ GateWay yêu cầu thay đổi node_id đã thành công.

TYPE: ACK

NODEID: <node_id>

DATA: <interval>

Mô tả: Gói tin từ Sensor Node phản hồi cho GateWay về việc tác vụ từ GateWay yêu cầu thay đổi chu kỳ gửi đã thành công.

Gói tin nhận được không đúng format GateWay sẽ bỏ qua và không xử lý nó.

- Khi GateWay gửi gói tin:

TYPE: DATA

NODEID: <node_id>

DATA: “True”

Mô tả: Gói tin yêu cầu gửi giá trị độ mặn ngay lập tức tới node cảm biến.

TYPE: CHANGEID

NODEID: <node_id> của node nhận

DATA: <node_id> mới mà GateWay muốn thay đổi

Mô tả: Gói tin từ GateWay yêu cầu thay đổi ID của node cảm biến.

TYPE: CHANGEINTERVAL

NODEID: <node_id> của node nhận

DATA: <interval> chu kỳ gửi dữ liệu tới GateWay của Sensor Node.

Mô tả: Gói tin từ GateWay yêu cầu thay đổi chu kỳ gửi dữ liệu của Sensor Node.

3.2.2.4 Cấu trúc tin nhắn SMS đến GateWay

No.	Cấu trúc	Mô tả
1	SAL <node_id>	Yêu cầu nhận dữ liệu độ mặn thời điểm hiện tại của node.
2	CHD <old_id> <new_id>	Yêu cầu thay đổi id của các node.
3	VAL <threshold> <option>	Yêu cầu thay đổi giá trị ngưỡng báo động tại các node. <option> có thể là <node_id> muốn được thay đổi hoặc “ALL” nếu muốn thay đổi giá trị ngưỡng tại tất cả các node.
4	CHI <node_id> <interval>	Yêu cầu thay đổi interval của các node.

3.2.3 Server

Như đã đề cập tại phần 3.1.2.4, Server đóng một vai trò quan trọng trong hệ thống, nó cung cấp API cho các node trung tâm (Gateway) đưa dữ liệu lên cơ sở dữ liệu, cung cấp các tính năng khác như yêu cầu dữ, cập nhật dữ liệu. Đồng thời Server đóng vai trò là một Web Server, cung cấp website giúp cho người dùng, các quản trị viên có thể xem dữ liệu một cách thực tế, trực quan giúp họ đánh giá được một cách tổng quát giá trị độ mặn tại các node. Lấy dữ liệu để phân tích cho các ứng dụng sau.

Với framework Express.js và mô hình MVC được giới thiệu ở phần trên, mô hình thiết kế của một Server được mô tả chi tiết dưới đây.

3.2.4.1 Thiết kế cơ sở dữ liệu

Dữ liệu được lưu trữ bằng MongoDB, vậy nên mỗi document đều có một field mặc định là ‘_id’ được thêm vào tự động mỗi khi có document mới được thêm vào. Mỗi ‘_id’ này là duy nhất và khác nhau. Vì thế, ta có thể sử dụng nó để làm khóa cho mỗi document.

Mỗi lược đồ dưới đây gắn liền với một model tương ứng trong mô hình MVC tạo thuận lợi cho việc sử dụng database.

Cơ sở dữ liệu được thiết kế thành các bảng 3.3, 3.4, 3.5, 3.6, 3.7.

Name	user	Lưu trữ thông tin của user	
Field	username	String	-
	pass	String	SHA256 hash code
	role	String	[admin, user, watiing]
	fullname	String	-
	email	String	Email format
	phone	String	Phone format
	usernumber	Number	-
Method	getAllUser	Lấy thông tin của tất cả user	
	getAllUserSortedByUsernumber	Lấy thông tin của tất cả user sắp xếp theo usernumber	
	findById	Lấy thông tin của user bằng Id	
	findByUsername	Lấy thông tin user bằng username	
	UpdateProfileUser	Cập nhật thông tin của user	
	ValidateUsernameAndEmail	Kiểm tra format của username và email	
	GetMaxUsernumber	Lấy giá trị max của usernumber	
	InsertNewUser	Thêm user	

	deleteUserByID	Xóa user dựa trên ID
	updateRoleByUserID	Cập nhật role của user bởi ID

Bảng 3.3. Lược đồ cấu trúc dữ liệu của user

Name	deleted_user_numbers	Lưu trữ các usernumber đã bị xóa	
Field	usernumber	Number	-
Method	count	Trả về số lượng usernumber đã bị xóa hiện có mặt trong lược đồ	
	insertDeletedUsernumber	Thêm một usernumber đã bị xóa	
	cutFirstUsernumber	Trả về usernumber đầu tiên và xóa nó ra khỏi collection	

Bảng 3.4. Lược đồ cấu trúc dữ liệu của deleted_user_numbers

Name	data	Dữ liệu được độ mặn được lưu trữ	
Field	usernumber	Number	usernumber đã bị xóa
	node_id	String	-
	salinity	Number	0..50
	date	Date	Date format
Method	getAllData	Lấy tất cả data hiện có trong collection	
	findById	Tìm data bởi id	
	InsertNewData	Thêm một data mới	
	GetDataByDate	Aggregation data theo ngày	
	GetDataByHour	Aggregation data theo giờ	
	GetDataBySalinity	Aggregation theo giá trị độ mặn (salinity)	

Bảng 3.4. Lược đồ cấu trúc dữ liệu của data

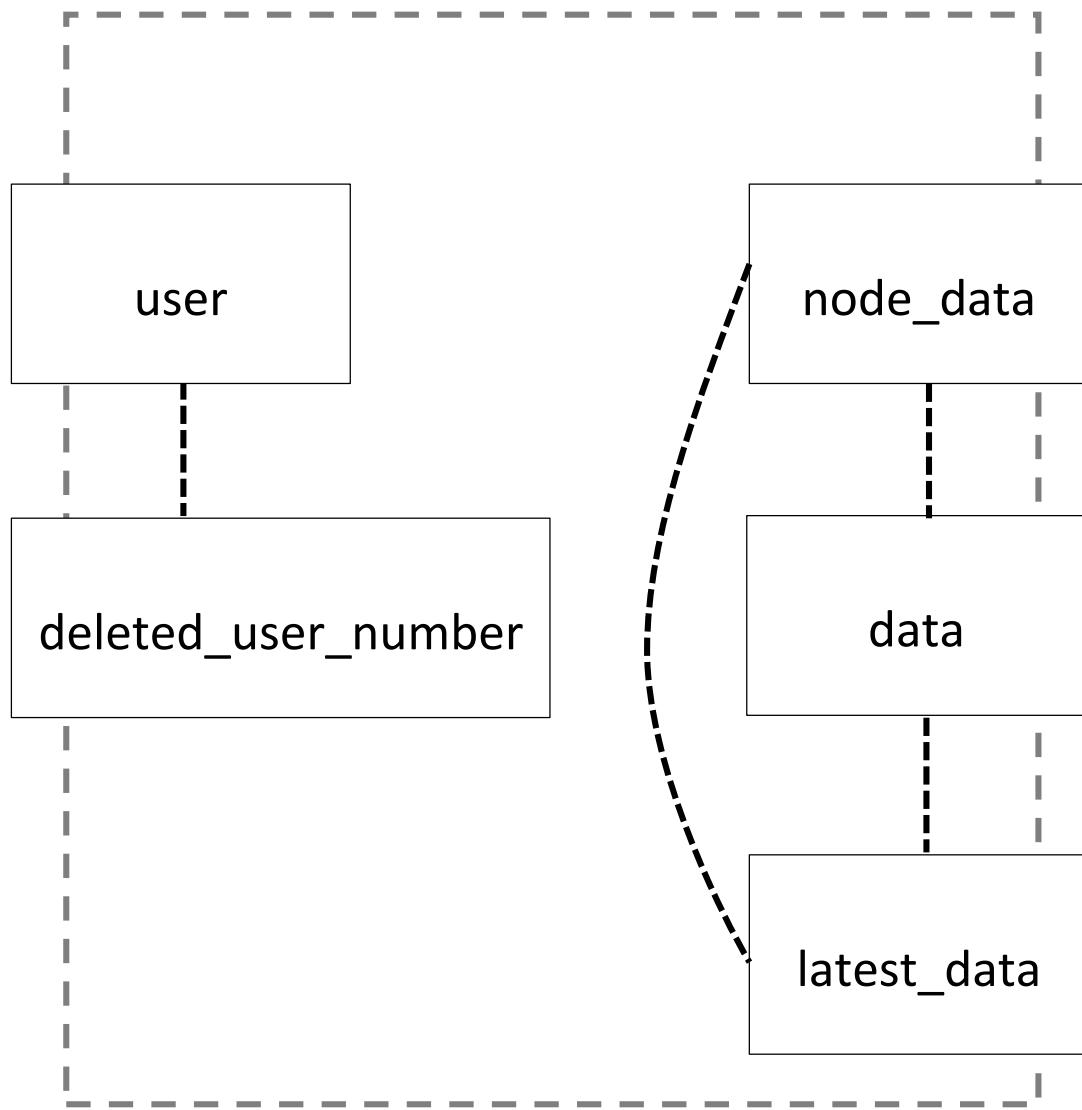
Name	data_latest	Dữ liệu được độ mặn mới nhất được lưu trữ để truy xuất nhanh chóng (chỉ 10 document tồn tại cho một node_id)	
Field	usernumber	Number	usernumber đã bị xóa
	node_id	String	-
	salinity	Number	0..50
	date	Date	Date format
Method	getAllData	Lấy tất cả data hiện có trong collection	
	findById	Tìm data bởi id	
	getAllByNodeId	Lấy tất cả data hiện có của một node_id	
	InsertNewData	Thêm một data mới	
	DeleteNodeLatestData	Xóa tất cả dữ liệu hiện có trong collection của node tương ứng	
	GetLatestValueOfNode	Lấy giá trị độ mặn mới nhất của một node	

Bảng 3.5. Lược đồ cấu trúc dữ liệu của data_latest

Name	node_data	Dữ liệu được độ mặn mới nhất được lưu trữ để truy xuất nhanh chóng (chỉ 10 document tồn tại cho một node_id)	
Field	node_id	Number	usernumber đã bị xóa
	information	String	-
	y	Number	-90..90
	x	Number	-180..180
	threshold	Number	0..50
	start_date	Date	Date format
Method	getAllNodeData	Lấy tất cả node data hiện có trong collection	
	findById	Tìm node data bởi id	
	UpdateNode	Cập nhật thông tin của node	
	GetMaxNodeId	Trả về giá trị max của node_id hiện có	
	AddNode	Thêm một node	
	DeleteNode	Xóa một node	
	UpdateAllNodeThreshold	Cập nhật giá trị ngưỡng của tất cả các node	
	UpdateNodeThreshold	Cập nhật giá trị ngưỡng của một node	
	ChangeNodeId	Thay đổi NodId	

Bảng 3.6. Lược đồ cấu trúc dữ liệu của node_data

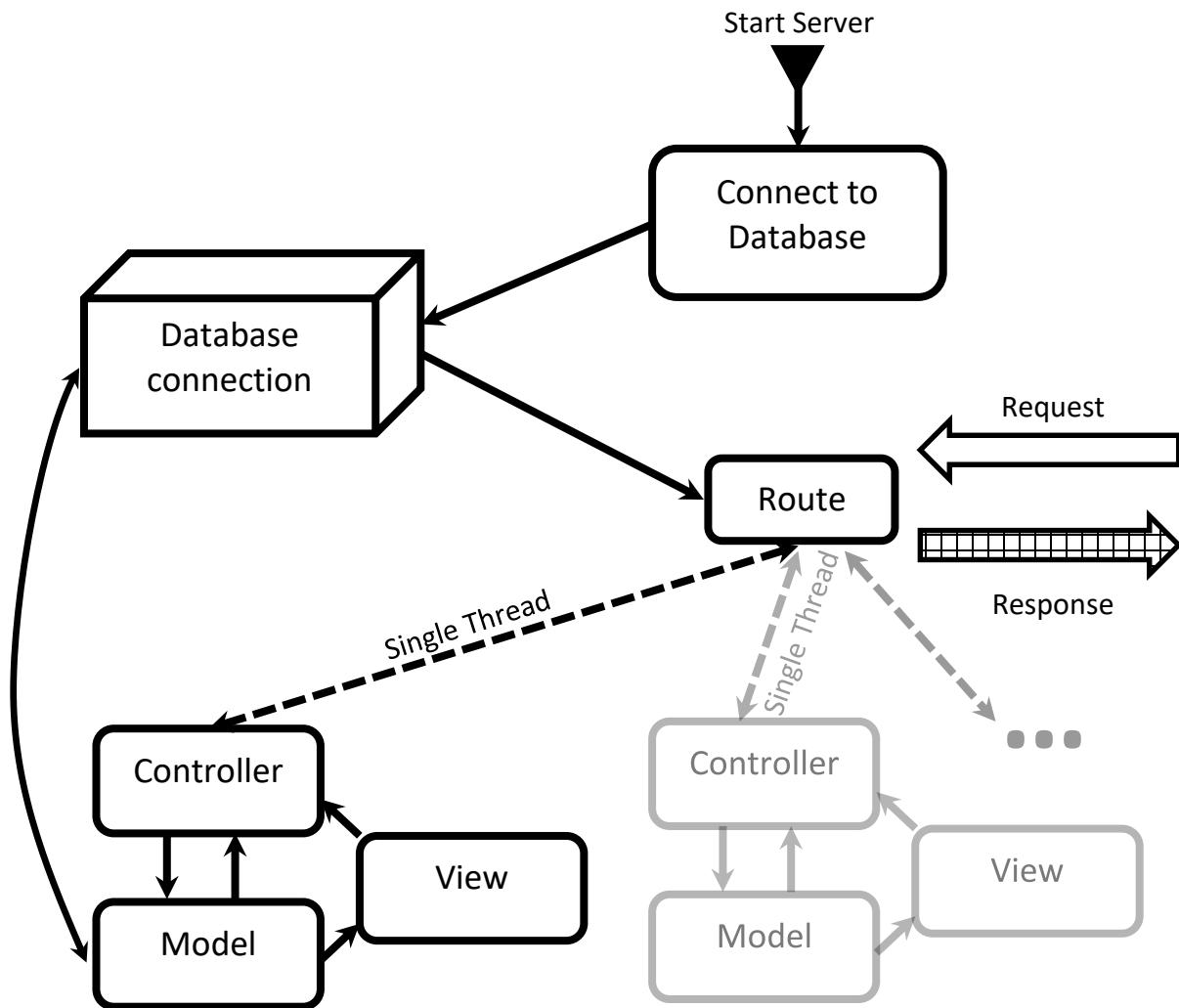
Sử dụng kiểu cơ sở dữ liệu NoSQL nên các schema không có ràng buộc với nhau, nhưng việc ràng buộc được thực hiện bởi người sử dụng database thông qua các thuật toán trước khi hành vi của ứng dụng tác động đến database được thể hiện bởi hình 3.29.



Hình 3.29. Sự ràng buộc các dữ liệu gây nên bởi application

3.2.4.2 Cấu trúc của Server

Mô hình hoạt động của Server



Hình 3.30. Mô hình hoạt động của server

Hình 3.30 mô tả cách thức hoạt động của một server. Khi server được khởi động, việc đầu tiên để nó thực hiện là kết nối với database và duy trì kết nối đó mãi về sau. Sau khi cơ sở dữ liệu được kết nối thành công, quá trình chờ nhận request từ client bắt đầu. Khi một request được gửi đến server, việc đầu tiên là nó sẽ điều hướng (route) xác định request này là gì, bước tiếp theo là xác nhận bộ controller tương ứng với request đó. Sau khi bộ controller được xác định, server tạo một thread riêng biệt cho việc xử lý yêu cầu từ server theo kiến trúc mô hình MVC được miêu tả tại phần 3.1.2.5 và quay trở lại chờ một request khác, kết nối được tạo ra ban đầu với cơ sở dữ liệu được sử dụng trong model để truy xuất, cập nhật cũng như làm các tác vụ đặc biệt khác. Sau khi kết thúc việc xử lý, một callback được gọi để thông báo với server. Server nhận được callback cùng với dữ liệu trả về của thread, cuối

cùng đáp ứng kết quả về cho client. Đây chính là một ưu điểm cực lớn của việc lập trình bắt đồng bộ, khiến cho server đáp ứng và xử lý nhanh hơn một cách đáng kể.

Các API cung cấp:

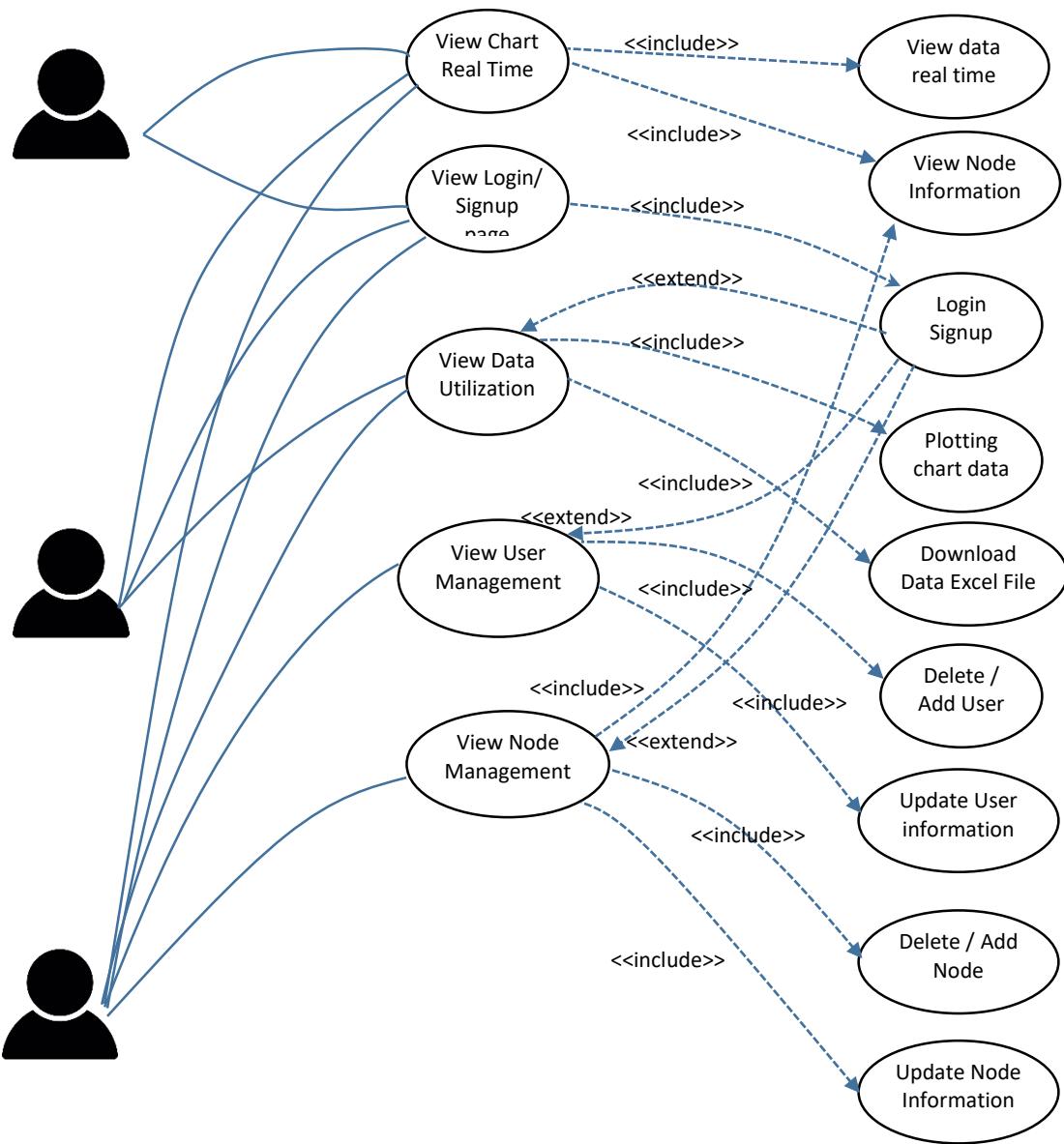
No.	Method	URL	Role Authentication
1	GET	/	All
2	GET	/index/	All
3	GET	/profile/	All
4	GET	/get_all_nodes/	All
5	GET	/get_latest_data/:node_id/	All
6	GET	/users/	User
7	POST	/users/login/	All
8	POST	/users/sign_up/	All
9	POST	/users/update_profile_user/	User
10	GET	/users/signout/	User
11	GET	/user_management/	Admin
12	GET	/user_management/delete_user/	Admin
13	POST	/user_management/add_an_user/	Admin
14	POST	/user_management/update_role_user/	Admin
15	GET	/data_utilization/	User
16	POST	/data_utilization/get_data_by_date/	User
17	POST	/data_utilization/get_data_by_hour/	User
18	POST	/data_utilization/get_data_by_salinity/	User
19	POST	/data_utilization/export_data/	User
20	GET	/data_utilization/get_data_file/:node_id/	User
21	GET	/data_utilization/import/:node_id/:salinity/	All
22	GET	/data_utilization/export/:node_id/	All
23	GET	/node_management/	Admin
24	GET	/node_management/get_node/:node_id/	All
25	GET	/node_management/update_node_threshold/all/:threshold	All
26	GET	/node_management/update_node_threshold/:node_id/:threshold	All
27	GET	/node_management/change_node_id/:old_node_id/:new_node_id	All
28	POST	/node_management/update_node	Admin
29	POST	/node_management/add_node	Admin
30	POST	/node_management/delete_node	Admin

Bảng 3.7. Các API cung cấp cho người dùng và Gateway

No	Mô tả
1	Trả về trang chủ dashboard
2	Trả về trang chủ dashboard
3	Trả về trang login/sign up hoặc trang thông tin của user nếu user đã đăng nhập
4	Lấy thông tin của tất cả các node
5	Lấy dữ liệu mới nhất của node
6	Trả về trang login/sign up hoặc trang thông tin của user nếu user đã đăng nhập
7	Đăng nhập
8	Đăng ký user
9	Cập nhật thông tin user
10	Đăng xuất
11	Trả về trang quản lý user
12	Xóa một user
13	Thêm một user
14	Cập nhật quyền của user
15	Trả về trang Data Utilization
16	Trả về data được phân tích theo ngày
17	Trả về data được phân tích theo giờ
18	Trả về data được phân tích theo độ mặn
19	Tạo file excel chứa thông tin data phân tích
20	Trả về file excel chứa thông tin data phân tích
21	Thêm data từ gateway
22	Lấy data mới nhất từ của node
23	Trả về trang quản lý node
24	Lấy thông tin của một node
25	Cập nhật giá trị báo động của tất cả các node
26	Cập nhật giá trị báo động của một node
27	Thay đổi node_id của node
28	Cập nhật thông tin node
29	Thêm một node
30	Xóa một node

Bảng 3.8. Mô tả các API của server

Biểu đồ High Level Use Case của Web server

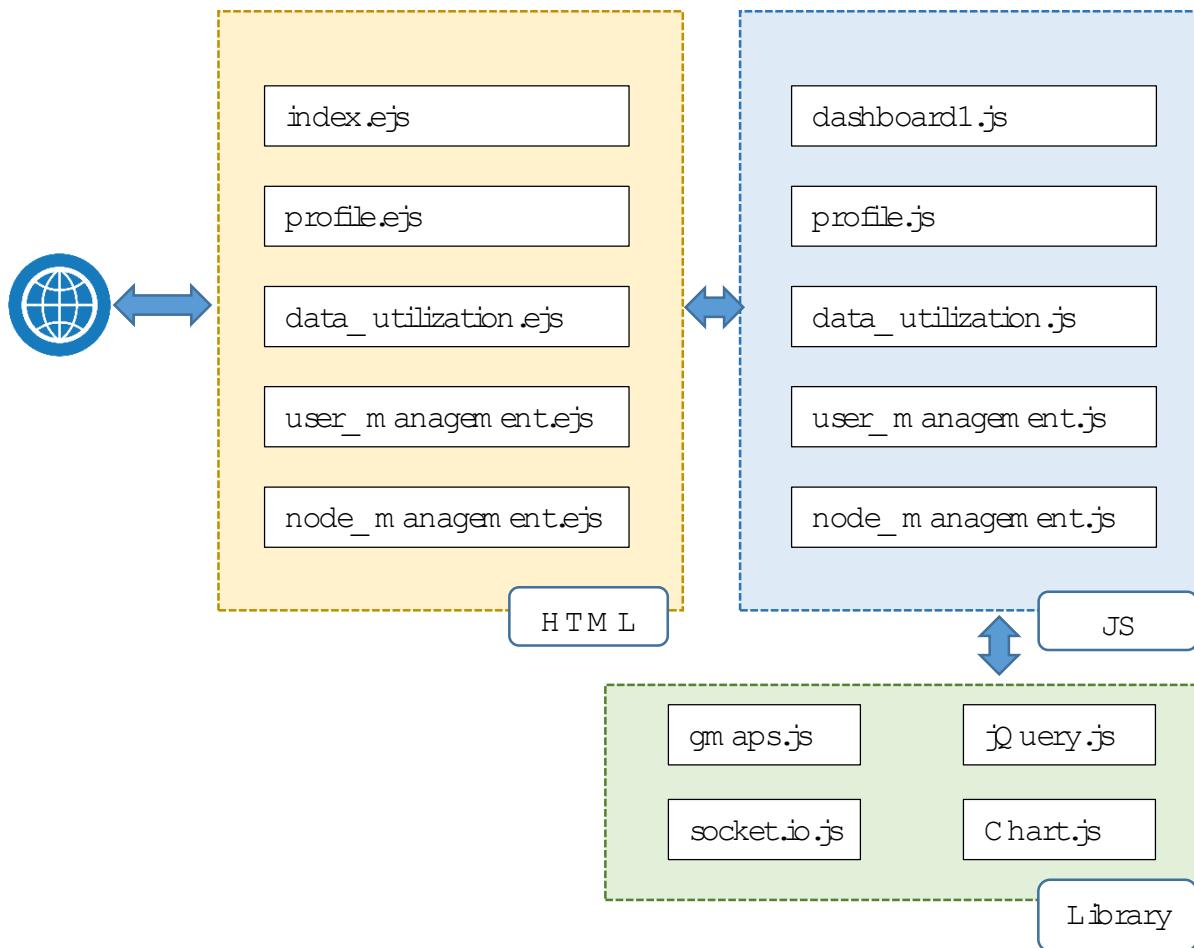


Hình 3.31. Biểu đồ High Lever Use Case của Web server

3.2.4 Web Client

Sau khi nhận kết quả trả về từ server là tất cả các file bao gồm: HTML, Javascript, hình ảnh,... browser sử dụng kết hợp tất cả các file trên, tạo nên một ứng dụng web với các giao diện và tính năng đầy đủ cho người dùng.

Mỗi View trong mô hình MVC tương ứng với một file HTML được tạo nên bởi khuôn mẫu EJS cùng với dữ liệu được truy vấn tại Model. Cấu trúc của toàn bộ file user nhận được tại browser được mô tả dưới đây.

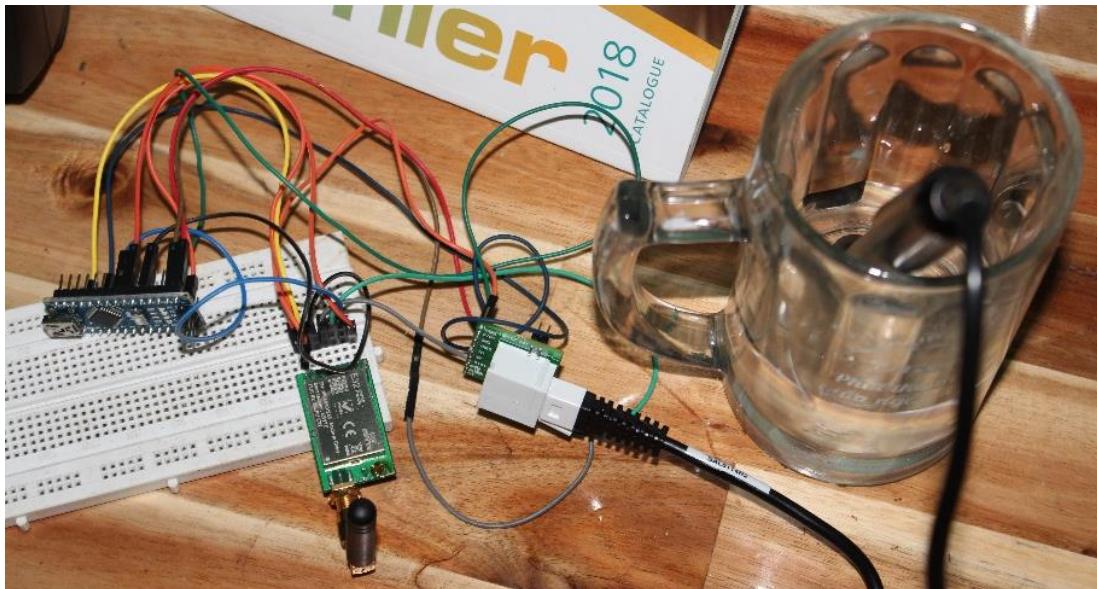


Hình 3.32. Cấu trúc file tương ứng tại browser của web user

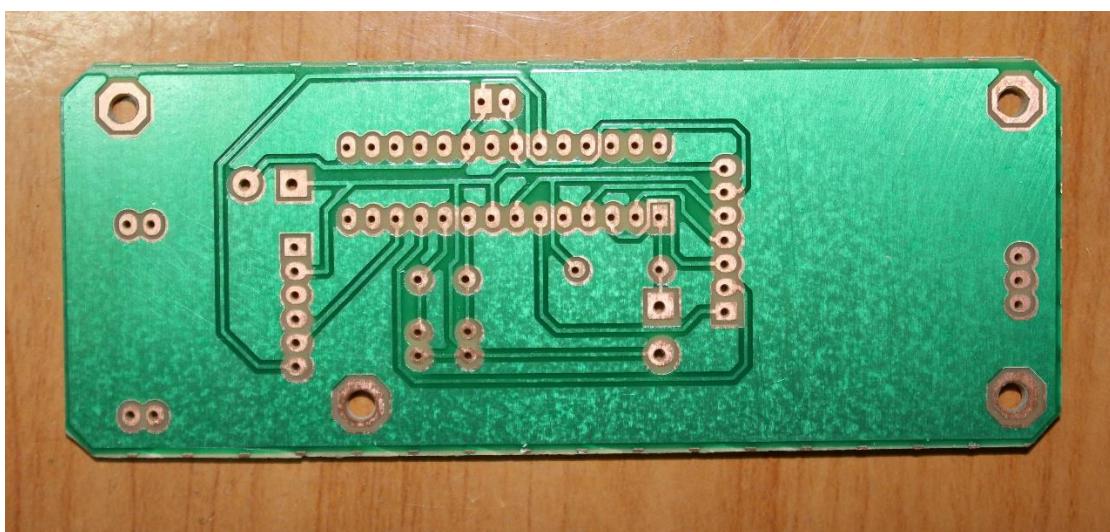
CHƯƠNG 4

HIỆN THỰC HỆ THỐNG

4.1 Node cảm biến



Hình 4.1. Breadboard đầu tiên được sử dụng để testing tại node cảm biến

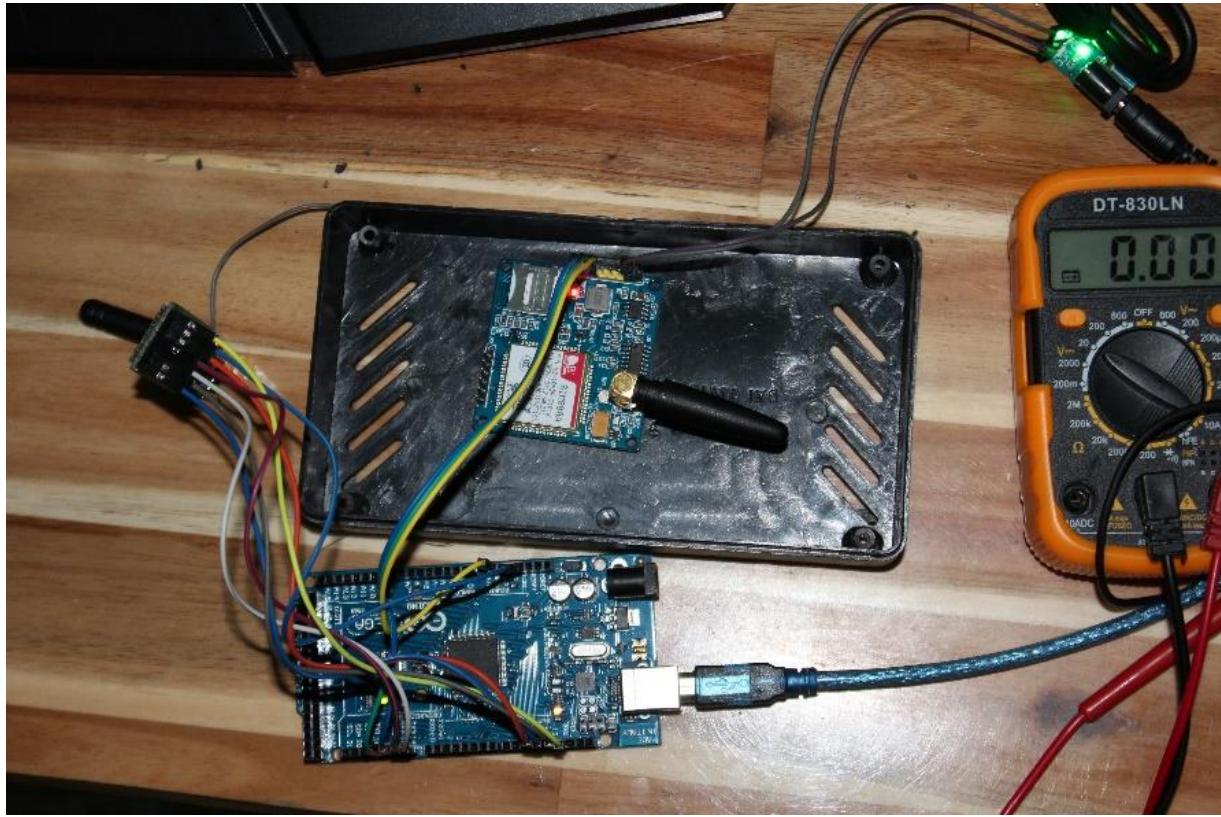


Hình 5. Board mạch của node cảm biến

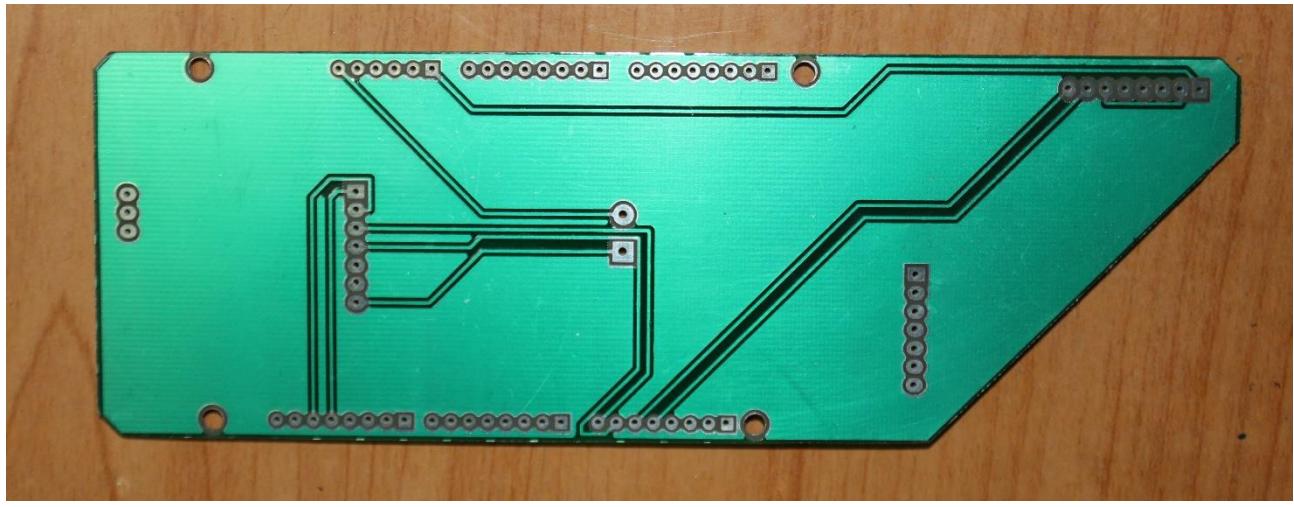


Hình 6. Node cảm biến sau khi hoàn thiện

4.2 GateWay



Hình 7. Testing tại GateWay

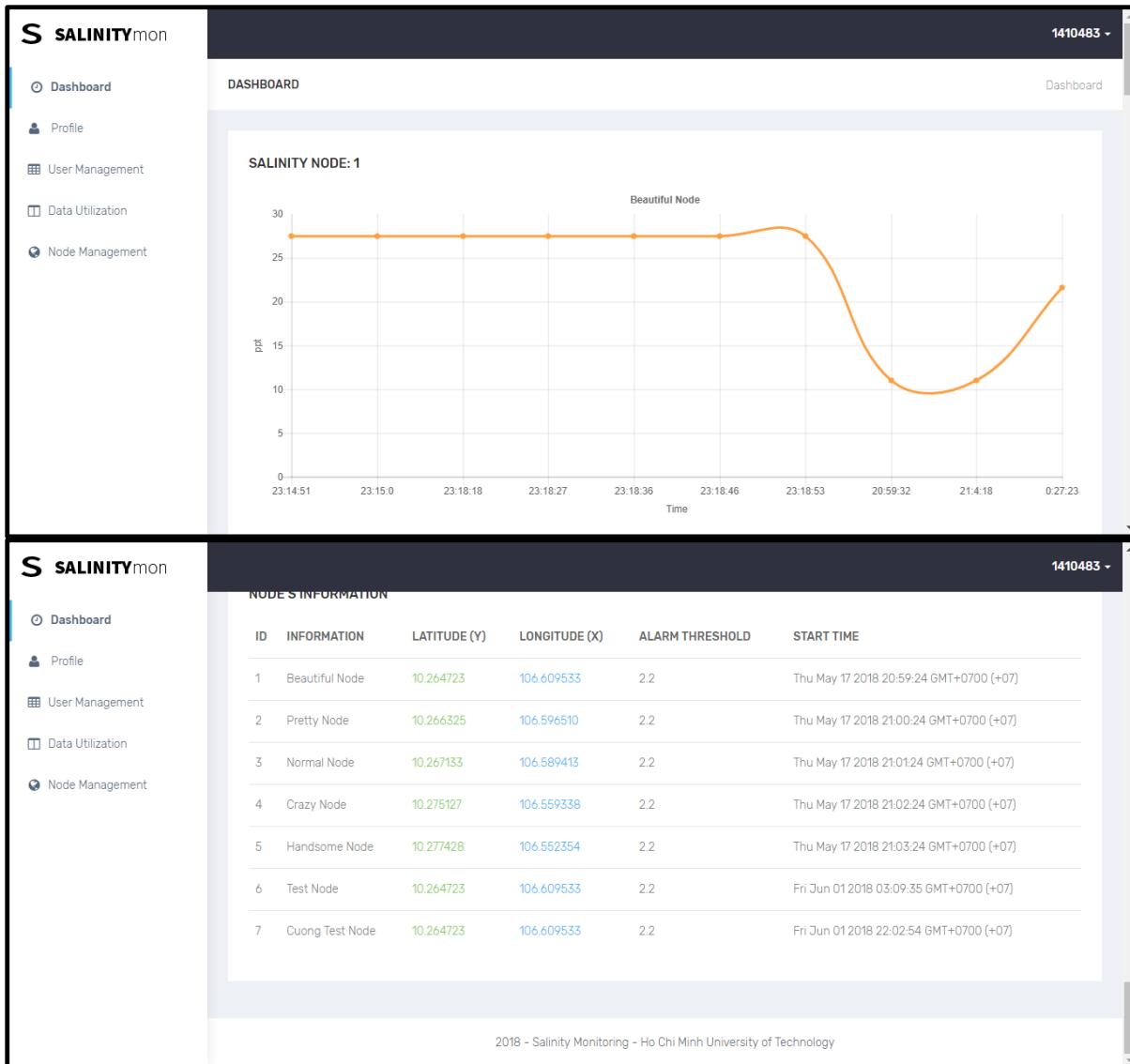


Hình 8. Board mạch của node trung tâm - GateWay



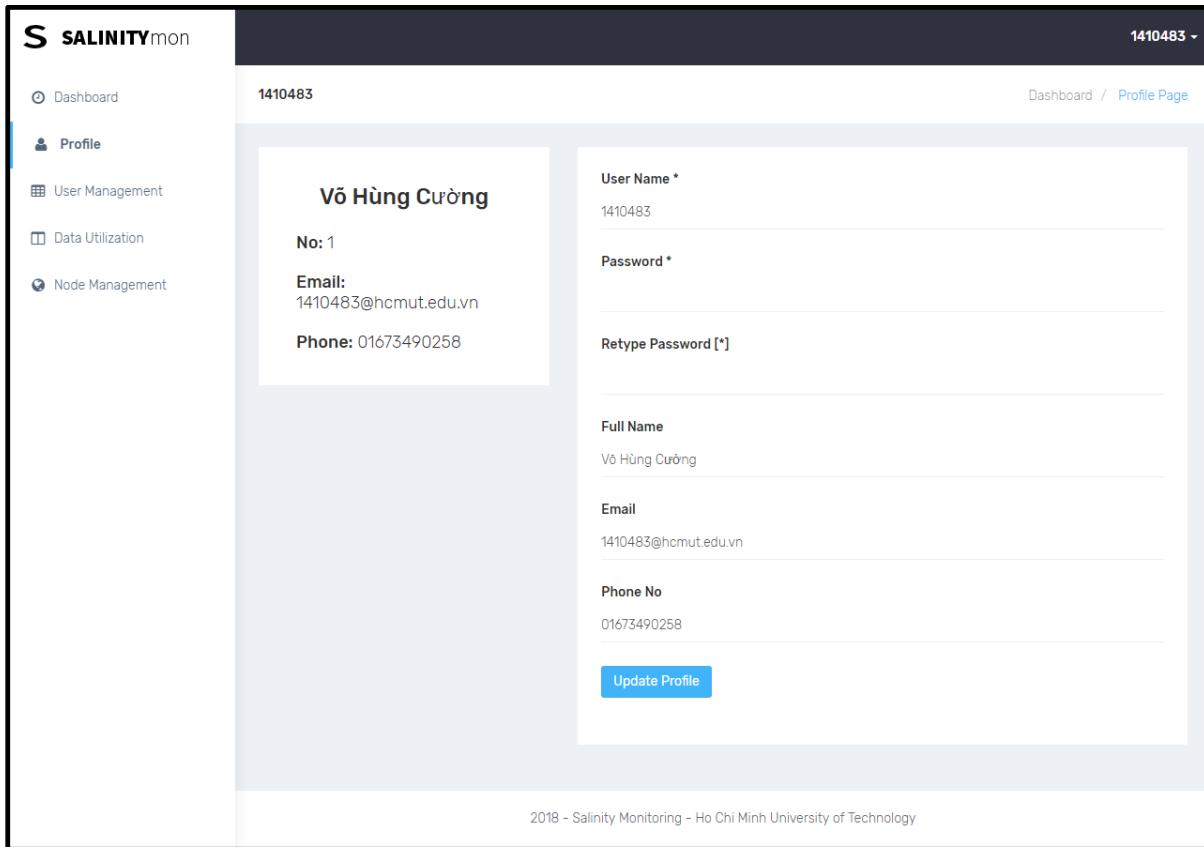
Hình 9. Node trung tâm – GateWay sau khi hoàn thiện

4.3 Giao diện web cho người dùng



Hình 10. Giao diện trang Dashboard

Trang Dashboard cho phép tất cả mọi người thấy được giá trị độ mặn của các node tại thời điểm hiện tại và thông tin các node hiện đang có trên hệ thống.



Hình 11. Giao diện trang profile của user

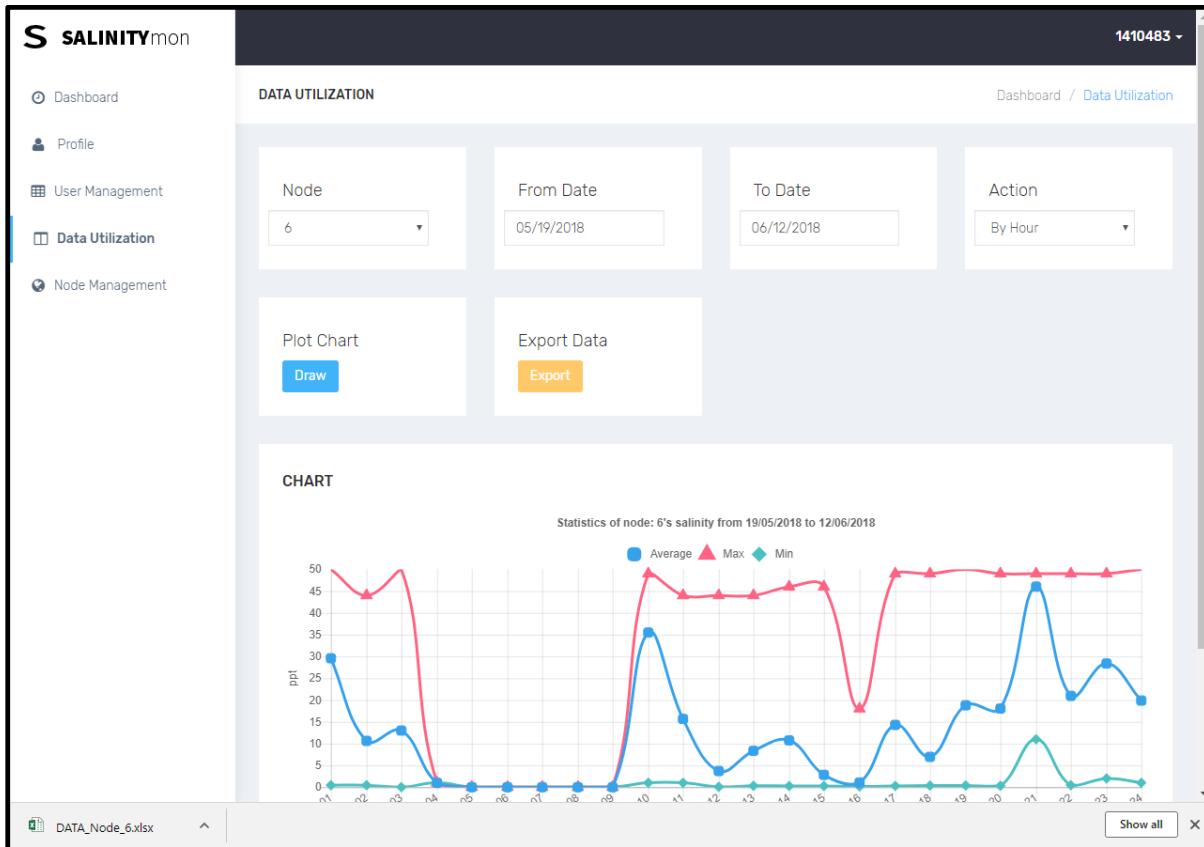
Tại đây user có thể cập nhật thông tin và thay đổi password của mình.

The screenshot shows the 'USER MANAGEMENT' section of the SALINITYmon application. On the left sidebar, 'User Management' is selected. The main area displays three tables:

- ADMIN USER**: Shows two entries. Number 1: Võ Hüng Cường, Email: 1410483@hcmut.edu.vn, Username: 1410483, Role: admin. Number 2: Tran Duc hung, Email: 1411631@hcmut.edu.vn, Username: 1411631, Role: admin.
- COMMON USER**: An empty table with columns: Number, Full Name, Email, Username, Role, Action.
- WAITTING USER**: Shows one entry. Number 3: Võ Hüng Cường, Email: cuongphevip@gmail.com, Username: cuongphevip, Role: WAITTING. Action buttons: Update, Delete.

Hình 12. Giao diện trang User Management

Trang User Management giành cho những user có quyền Admin có thể chỉnh sửa quyền của các user khác cũng như thêm và xóa tài khoản của user.



Hình 4.10. Giao diện trang Data Utilization

Trang Data Utilization dành cho các user đã đăng nhập, tại đây họ có thể vẽ ra những biểu đồ, đồ thị theo những trường khác nhau một cách trực quan, đồng thời có thể lấy dữ liệu về dưới dạng file excel để sử dụng với những mục đích khác như phân tích, đánh giá.

The figure consists of two screenshots of a web-based application interface for managing nodes in a salinity monitoring system.

Screenshot 1: NODE MANAGEMENT

- Left Sidebar:** Includes links for Dashboard, Profile, User Management, Data Utilization, and Node Management (highlighted).
- Header:** Shows the user ID 1410483 and a navigation bar with Dashboard / Node Management.
- Content Area:**
 - Node Selection:** A dropdown menu labeled "Node" with value "1".
 - Node Information:** Displays "Beautiful Node".
 - Coordinate:** Shows Latitude (y) 10.264723 and Longitude (x) 106.609533.
 - Alarm Threshold:** Set to 2.2.
 - Start Time:** Set to Thu May 17 2018 20:59:24 GMT+0700 (+07).
 - Action Buttons:** Update Node (blue), Add Node (yellow), Delete Node (red).

Screenshot 2: NODE POSITION

- Left Sidebar:** Same as the first screenshot.
- Header:** Shows the user ID 1410483.
- Content Area:**
 - Map:** A Google Map showing the Mekong River area. A red marker indicates the position of the "Beautiful Node". A callout box points to the marker with the text "Beautiful Node".
 - Map Controls:** Includes zoom controls (+, -), a map type switcher (Bản đồ, Vẽ tinh), and a scale bar.
 - Attribution:** Includes "Google" and copyright information: "Đữ liệu bản đồ ©2018 Google | 50 m | Điều khoản sử dụng | Báo cáo một lỗi bản đồ".
 - Page Footer:** "2018 - Salinity Monitoring - Ho Chi Minh University of Technology"

Hình 4.11. Giao diện trang Node Management

Trang Node Management dành cho các user có quyền Admin, hiển thị thông tin node, cũng như cập nhật, thêm và xóa các node. Bên cạnh đó còn thể hiện vị trí của các node qua bản đồ Google Maps.

CHƯƠNG 5

TỔNG KẾT ĐÁNH GIÁ CÁC KẾT QUẢ HIỆN THỰC

Mọi thông tin chi tiết về Source code, thiết kế, đến hiện thực được chúng tôi public tại: <https://github.com/cuongphevip/SalinityMonitoring>

5.1 Hoàn thành và kết quả đạt được

Node cảm biến

- Hiện thực thành công protocol giao tiếp giữa các node cảm biến và GateWay.
- Giá trị độ mặn có ảnh hưởng lớn đến hệ thống nông lâm ngư nghiệp của cả nước nói chung và Đồng bằng Sông Cửu Long nói riêng, việc đọc giá trị độ mặn và lưu trữ dữ liệu lên server là một điều vào quan trọng trong việc phân tích và đánh giá sau này của các nhà nghiên cứu.
- Với chế độ Sleep Power Down của vi điều khiển, năng lượng làm việc của vi điều khiển là rất thấp, việc tận dụng triệt để chế độ này giúp cho các board mạch tại node cảm biến có khả năng sử dụng lâu dài hơn mà không cần phải thay nguồn năng lượng mới.

Công nghệ giao tiếp không dây LoRa

Công nghệ LoRa được áp dụng trong đề tài là một công nghệ mới nhưng mang tính thực tiễn và có khả năng phát triển rộng trong tương lai với khả năng truyền và nhận không dây trong một phạm vi rộng với mức sử dụng năng lượng thấp.

GateWay

Quản lý về việc vừa sử dụng công nghệ truyền và nhận LoRa đồng thời với GPRS tích hợp bằng module SIM tạo điều kiện thuận lợi cho việc người quản lý sử dụng tất cả mọi nguồn tài nguyên từ điện thoại đến Internet để kiểm soát, quản lý các node cảm biến độ mặn. Khiến cho mô hình có tính triển khai mở rộng cao hơn.

Server

Triển khai server thành công, là nơi lưu trữ dữ liệu đồng thời cũng là nơi cung cấp các tài nguyên cần thiết cho việc phân tích đánh giá sau này. Việc quản lý dữ liệu một cách riêng biệt mà không cần nhờ một bên thứ 3 nào khiến cho tính bảo mật thông tin được nâng cao hơn.

Bên cạnh đó, các kiến thức về việc xây dựng một hệ thống từ khâu đặt mục tiêu, thiết kế, quản lý đến hiện thực đã giúp nhiều cho chúng tôi trong việc thực hiện các đề tài khác trong tương lai.

5.2 Hạn chế

Giải thuật vẫn chưa được tối ưu hóa hoàn toàn khiến cho việc truyền nhận giữa GateWay và các node đôi khi xảy ra lỗi.

Các board mạch và phần cứng tuy đã hoạt động tốt nhưng chưa ổn định.

Việc kiểm định trong một khoảng thời gian dài của các board mạch, server chưa được tiến hành, vì vậy từ một sản phẩm mang tính thí nghiệm để đưa đến bước hiện thực ngoài thực tế còn phải trải qua nhiều giai đoạn lớn.

Về tính an toàn của sản phẩm vẫn chưa được hiện thực. Trong giao tiếp giữa Node cảm biến và GateWay, mọi bên thứ 3 đều có thể có thông tin từ các gói tin truyền nhận giữa chúng, ngoài ra, với những tác động bằng các gói tin rác, hệ thống Node và GateWay có thể bị treo khi phải xử lý quá nhiều gói tin cùng một lúc. Bên cạnh đó server không thể phòng ngừa những tấn công đặc biệt từ bên ngoài bởi vì việc triển khai hệ thống an ninh mạng hiện chưa được phát triển.

5.3 Kế hoạch và hướng phát triển trong tương lai

- Kết hợp với các nguồn năng lượng khác như năng lượng mặt trời, gió, mưa để nâng cao tuổi thọ pin tại các node cảm biến.
- Viết ứng dụng di động dành riêng cho người dùng của hệ thống.
- Nâng cao tính bảo mật của hệ thống thông qua nâng cao tính bảo mật khi truyền nhận qua công nghệ LoRa đồng thời xây dựng hệ thống an ninh cho Server.
- Tìm kiếm, hợp tác để có được một sensor độ mặn phù hợp để có thể sử dụng lâu dài trong môi trường thiên nhiên, công nghiệp phù hợp cho tính triển khai thực tế.
- Kết hợp OS vào GateWay để tốc độ xử lý được cải thiện, tiến trình được quản lý dễ dàng hơn.

CHƯƠNG 6

THAM KHẢO

- [1] “Xâm Nhập Mặn” - Bách khoa toàn thư mở Wikipedia.
https://vi.wikipedia.org/wiki/Xâm_nhập_mặn
- [2] “Xâm nhập mặn vùng Đồng bằng sông Cửu Long (2015 – 2016), hạn hán ở Miền Trung, Tây Nguyên và giải pháp khắc phục” – Tổng cục thủy lợi.
<http://www.tongcucthuylieu.gov.vn/>
- [3] “Miền Tây hạn, mặn nghiêm trọng nhất 100 năm” – VnExpress.
<https://vnexpress.net/tin-tuc/thoi-su/mien-tay-han-man-nghiem-trong-nhat-100-nam-3356338.html>
- [4] “Việt Nam vay Nhật gần 5.000 tỷ đồng xây cống ngăn mặn ở Bến Tre” – DanTri
<http://dantri.com.vn>
- [5] “Internet Vạn Vật” - Bách khoa toàn thư mở Wikipedia.
https://vi.wikipedia.org/wiki/Internet_Vạn_Vật
- [6] “ITS HCMUT” – Đại học Bách Khoa thành phố Hồ Chí Minh.
<https://traffic.hcmut.edu.vn/>
- [7] Luận văn tốt nghiệp Đại học Bách Khoa thành phố Hồ Chí Minh, ngành Kỹ thuật máy tính – Nghiên cứu và hiện thực hệ thống quan trắc khí thải của các phương tiện giao thông. Thực hiện: Nguyễn Mạnh Cường, Huỳnh Phạm So Ny, Võ Tấn Tùng.
- [8] “IoT-Enabled Smart Parking Management” – SEMTECH
- [9] “4-ELECTRODE CONDUCTIVITY SENSOR / DIGITAL / FOR WASTEWATER / FOR CLEAN WATER” – AQULABO
<http://www.directindustry.com/prod/aqualabo/product-54155-1949558.html>
- [10] “SX1272/3/6/7/8: LoRa Modem - Low Energy Consumption Design” – SEMTECH
http://www.semtech.com/images/datasheet/LoraLowEnergyDesign_STD.pdf
- [11] “SIM800 Series_EMBEDDED - AT_Sleep_Application_Note_V1.01”

https://cdn-shop.adafruit.com/product-files/2637/SIM800+Series+Embedded+AT+Sleep+Application+Note_V1.01.pdf

- [12] “Nước mặn” - Bách khoa toàn thư mở Wikipedia.
https://vi.wikipedia.org/wiki/N%C6%B0%E1%BB%9Bc_m%E1%BA%B7n

- [13] “Độ mặn” - Bách khoa toàn thư mở Wikipedia.
https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99_m%E1%BA%B7n

- [14] “Salinity Sensor” – Vernier
<https://www.vernier.com/products/sensors/sal-bta/>

- [15] “Sensor Pinouts” – Vernier
<https://www.vernier.com/support/sensor-pinouts/>

- [16] “ARDUINO UNO REV3” – Arduino
<https://store.arduino.cc/usa/arduino-uno-rev3>

- [17] “ARDUINO NANO” – Arduino
<https://store.arduino.cc/usa/arduino-nano>

- [18] “E32-433T20DT” – EBYTE
<http://www.cdebyte.com/en/product-view-news.aspx?id=130>

- [19] “E32-TTL-100 Datasheet v1.2” – EBYTE

- [20] “ARDUINO MEGA 2560 REV3” – Arduino
<https://store.arduino.cc/usa/arduino-mega-2560-rev3>

- [21] “Node.js là gì ?” – VietJack
http://vietjack.com/nodejs/nodejs_la_gi.jsp

- [22] <https://nodejs.org/en/>

- [23] “Node JS Tutorial for Beginners #1 – Introduction” – The Net Ninja
<https://www.youtube.com/watch?v=w-7RQ46RgxU&list=PL4cUxeGkcC9gcy9lrvMJ75z9maRw4byYp>

- [24] “Chrome V8” – Wikipedia The Free Encyclopedia

https://en.wikipedia.org/wiki/Chrome_V8

- [25] “WebSocket” - Wikipedia The Free Encyclopedi

<https://en.wikipedia.org/wiki/WebSocket>

- [26] “Node.js Introduction” – w3schools.com

https://www.w3schools.com/nodejs/nodejs_intro.asp

- [27] “Express/Node introduction” – MDN web docs

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

- [28] “MVC Architecture” – TutorialsTeacher

Ht <http://www.tutorialsteacher.com/mvc/mvc-architecture>

- [29] “What is WebSocket?” – PubNub

<https://www.pubnub.com/learn/glossary/what-is-websocket/>

- [30] “socket.io” – socket.io

<https://socket.io/docs/>

- [31] “MongoDB” – techopedia.

<https://www.techopedia.com/definition/30340/mongodb>

- [32] “Database as a Service (DBaaS)” – techopedia.

<https://www.techopedia.com/definition/29431/database-as-a-service-dbaas>

- [33] “jQuery” – jQuery

<https://jquery.com/>

- [34] “mLab” – mLab

<https://mlab.com/>

- [35] “Atmega328P” – Microchip

<https://www.microchip.com/wwwproducts/en/ATmega328P>

- [36] “AVR® Low Power Sleep Modes” – MicroChip Developer Help

<http://microchipdeveloper.com/8avr:avrsleep>