

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

ĐỒ ÁN TỐT NGHIỆP
CAPSTONE PROJECT

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:

**XÂY DỰNG HỆ THỐNG GIÁM SÁT VÀ THU
THẬP DỮ LIỆU ĐIỆN NĂNG TRÊN NỀN TẢNG
THINGSBOARD**

Giảng viên hướng dẫn: **TS. TRẦN THÁI ANH ÂU**

Đồng hướng dẫn: **KS. NGUYỄN KHƯƠNG QUỲNH**

Nhóm sinh viên thực hiện:

1. ĐỖ BÁ HẢI – MSSV: 105200359 – LỚP: 20TDHCLC1

2. NGUYỄN SƠN TÙNG – MSSV: 105200394 – LỚP: 20TDHCLC1

Đà Nẵng, 6/2025

TÓM TẮT

Tên đề tài: Xây dựng hệ thống giám sát và thu thập dữ liệu điện năng trên nền tảng ThingsBoard.

Sinh viên thực hiện:

- | | | |
|--------------------|----------------------|----------------|
| 1. Đỗ Bá Hải | Số thẻ SV: 105200359 | Lớp: 20TDHCLC1 |
| 2. Nguyễn Sơn Tùng | Số thẻ SV: 105200394 | Lớp: 20TDHCLC1 |

Giám sát và thu thập dữ liệu là một trong những lĩnh vực đa dạng và phát triển rất nhanh trong ngành Kỹ thuật điều khiển và Tự động hóa. Đồ án tốt nghiệp với đề tài “Xây dựng hệ thống giám sát và thu thập dữ liệu điện năng trên nền tảng ThingsBoard” không chỉ cung cấp một giải pháp giám sát điện năng thông minh mà còn giúp cho người chủ đầu tư có cái nhìn tổng quan về việc quản lý năng lượng tiêu thụ và đặc biệt là tại Việt Nam khi mà Tập đoàn Điện lực Việt Nam (EVN) quy định giá cả năng lượng cho hoạt động sản xuất ở ba khung giờ khác nhau (khung giờ bình thường, khung giờ thấp điểm và khung giờ cao điểm). Điều này giúp cho người chủ đầu tư có thể phân bổ nhân công hay sắp xếp các hoạt động sản xuất công nghiệp để tối ưu hóa chi phí năng lượng trong quá trình sản xuất.

Hệ thống trong đề tài lần này có các chức năng chính như: Giám sát dữ liệu điện năng theo thời gian thực cùng với biểu đồ trực quan, phân quyền và quản lý người dùng, tạo cảnh báo và gửi về E-mail người dùng khi mà các thông số vượt ngưỡng quy định và đặc biệt là tính toán được công suất tiêu thụ ở từng khung giờ khác nhau và tính được tiền điện cho từng khung giờ tương ứng đó. Việc xây dựng hệ thống trên nền tảng IoT ThingsBoard là một giải pháp có tính linh hoạt, có khả năng mở rộng và hỗ trợ giao thức MQTT giúp cho việc xây dựng dễ dàng hơn so với việc phải lập trình Web.

Nội dung của đồ án này trình bày chi tiết quá trình thu thập dữ liệu thô của đồng hồ Selec từ PLC qua giao thức Modbus RTU sau đó kết nối và gửi dữ liệu từ PLC đến ThingsBoard qua MQTT Server. Tại ThingsBoard tiếp tục thực hiện việc xử lý dữ liệu thô đó từ đó xây dựng các chức năng cơ bản của một hệ giám sát.

Kết quả của đồ án là một hệ thống giám sát năng lượng hoàn chỉnh, đáp ứng các yêu cầu về giám sát hiệu quả và giao diện người dùng thân thiện. Hệ thống trong đề tài này là một giải pháp rất phù hợp cho nhu cầu giám sát năng lượng của các cơ sở sản xuất vừa cho tới các nhà máy hoạt động sản xuất công nghiệp lớn.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	Đỗ Bá Hải	105200359	20TDHCLC1	Kỹ thuật điều khiển và Tự động hóa
2	Nguyễn Sơn Tùng	105200394	20TDHCLC1	Kỹ thuật điều khiển và Tự động hóa

1. Tên đề tài đồ án:

Xây dựng hệ thống giám sát và thu thập dữ liệu điện năng trên nền tảng ThingsBoard.

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

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

4. Nội dung các phần thuyết minh và tính toán:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Đỗ Bá Hải	- Tìm hiểu tổng quan đề tài. - Xây dựng kiến trúc IoT phù hợp cho hệ thống. - Tìm hiểu và đầu nối phần cứng của hệ thống. - Xây dựng các thuật toán xử lý dữ liệu. - Thực hiện đánh giá kết quả của hệ thống. - Viết báo cáo thuyết minh.
2	Nguyễn Sơn Tùng	

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Đỗ Bá Hải	- Kết nối Modbus RTU để đọc dữ liệu từ PLC - Lập trình và cấu hình các thông số để kết nối PLC với MQTT Server. - Xây dựng các chức năng: Báo cáo, Gửi cảnh báo về Email. - Thiết kế giao diện trang Overview và Notify.

2	Nguyễn Sơn Tùng	- Tạo mô phỏng các thông số giả lập từ thông số thật của đồng hồ - Lập trình và cấu hình các thông số để kết nối MQTT Server với ThingsBoard. - Xây dựng các chức năng: Lưu dữ liệu vào MySQL, phân quyền và quản lý người dùng. - Thiết kế giao diện trang Electricity và Energy.
---	-----------------	---

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Đỗ Bá Hải	- Bản vẽ sơ đồ đấu nối dây của hệ thống.
2	Nguyễn Sơn Tùng	

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung

6. Họ tên người hướng dẫn:	Phần/ Nội dung:
TS. Trần Thái Anh Âu	- Hướng dẫn thuyết minh báo cáo đề tài.
KS. Nguyễn Khương Quỳnh	- Hướng đi của đề tài. - Giải pháp kết nối cho kiến trúc IoT. - Xây dựng giao diện giám sát năng lượng.

7. Ngày giao nhiệm vụ đồ án: 01/03/2025

8. Ngày hoàn thành đồ án: 01/06/2025

Đà Nẵng, ngày 15 tháng 6 năm 2025.

Trưởng Bộ môn Tự Động Hóa

Người hướng dẫn

Đồng hướng dẫn

TS. Giáp Quang Huy

TS. Trần Thái Anh Âu

KS. Nguyễn Khương Quỳnh

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Đỗ Bá Hải

Số thẻ SV: 105200359

Nguyễn Sơn Tùng

Số thẻ SV: 105200394

Tên đề tài ĐATN: Xây dựng hệ thống giám sát và thu thập dữ liệu điện năng trên nền tảng ThingsBoard.

Họ tên GVHD: TS. Trần Thái Anh Âu

Đơn vị: Khoa Điện, Trường ĐHBK

Họ tên KS đồng HD: KS. Nguyễn Khương Quỳnh

Đơn vị: Công ty AUE VIỆT NAM

Tuần	Ngày	Khối lượng		GVHD ký tên
		đã thực hiện (%)	tiếp tục thực hiện (%)	
1	01/03/2025	Nhận đề tài	Tìm hiểu tổng quan đề tài	
2	10/03/2025	Xác định tính cấp thiết, mục tiêu, phạm vi, đối tượng và phương pháp nguyên cứu.	Tìm hiểu các yêu cầu của đề tài.	
3	17/03/2025	Tìm hiểu tổng quan các giải pháp kết nối.	Viết báo cáo thuyết minh phần mở đầu và chương một.	
4	24/03/2025	Duyệt lần 1: Đánh giá khối lượng hoàn thành ____ % : Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐		
5	25/03/2025	Xây dựng kiến trúc IoT phù hợp với giải pháp đã chọn.	Tìm hiểu các thiết bị phần cứng.	
6	31/03/2025	Cài đặt, đấu nối thiết bị phần cứng.	Tìm hiểu nguyên lý của hệ thống.	
7	07/04/2025	Cài đặt, đấu nối thiết bị phần cứng.	Viết báo cáo thuyết minh chương hai.	
8	14/04/2025	Duyệt lần 2: Đánh giá khối lượng hoàn thành ____ % : Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐		

9	15/04/2025	Hoàn thành báo cáo thuyết minh chương hai.	Tìm hiểu về nền tảng IoT ThingsBoard.	
10	21/04/2025	Viết code để đọc dữ liệu về PLC bằng Modbus RTU.	Tạo dữ liệu mô phỏng từ đồng hồ thật.	
11	28/04/2025	Kết nối PLC với nền tảng ThingsBoard qua MQTT Server.	Viết báo cáo thuyết minh chương ba.	
12	5/05/2025	Duyệt lần 3: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐		
13	06/05/2025	Hoàn thiện báo cáo chương ba.	Xây dựng thuật toán xử lý dữ liệu trên ThingsBoard.	
14	15/05/2025	Xây dựng các chức năng của hệ giám sát trên ThingsBoard	Viết báo cáo chương bốn và chương năm.	
15	01/06/2025	Kiểm nghiệm và đánh giá kết quả.	Hoàn thiện thuyết minh ĐATN. Chuẩn bị slide và nội dung thuyết trình.	

LỜI NÓI ĐẦU VÀ CẢM ƠN

Trải qua một thời gian nghiên cứu và thực hành làm đồ án tốt nghiệp mặc dù vẫn còn nhiều hạn chế về kiến thức cũng như kỹ năng chuyên môn nhưng nhờ vào sự giúp đỡ của rất nhiều người trong cùng lĩnh vực đã động viên và giúp đỡ thì cuối cùng nhóm em cũng đã hoàn thành xong đồ án tốt nghiệp.

Lời đầu tiên xin cho phép nhóm em gửi lời cảm ơn sâu sắc đến TS. Trần Thái Anh Âu, là người thầy luôn quan tâm và giới thiệu giúp cho nhóm chúng em được làm đồ án tốt nghiệp tại Công ty TNHH AUE Việt Nam. Ngoài ra thầy còn tận tình dành thời gian đọc báo cáo thuyết minh và góp ý chi tiết để nhóm kịp thời sửa lỗi và hoàn thành được báo cáo thuyết minh.

Xin chân thành gửi lời cảm ơn đến anh Nguyễn Quang Vũ – là Tổng Giám Đốc Công ty TNHH AUE Việt Nam đã tạo điều kiện thuận lợi cho nhóm em được thực tập cũng như làm đồ án tốt nghiệp tại công ty mình.

Xin chân thành cảm ơn anh Nguyễn Khương Quỳnh – là Kỹ sư tại Công ty TNHH AUE Việt Nam đã cung cấp các tài liệu tham khảo của hãng Mitsubishi, truyền đạt kiến thức và kinh nghiệm cũng như định hướng hướng đi của đề tài và hỗ trợ nhóm em trong quá trình làm đồ án tại công ty mình.

Cuối cùng, nhóm em xin chân thành cảm ơn ban giám hiệu trường Đại học Bách Khoa – Đại học Đà Nẵng cũng như các quý Thầy Cô tại khoa Điện đã truyền đạt những bài học quý giá, những kinh nghiệm bổ ích và tạo điều kiện thuận lợi cho nhóm em hoàn thiện đồ án này.

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

“Nhóm em xin cam đoan đề tài đồ án tốt nghiệp: *“Xây dựng hệ thống giám sát và thu thập dữ liệu điện năng trên nền tảng ThingsBoard”* là một dự án nghiên cứu, tìm hiểu và thực hành độc lập của nhóm em dưới sự hướng dẫn của TS. Trần Thái Anh Âu và KS. Nguyễn Khương Quỳnh. Toàn bộ sản phẩm, cách thức thực hiện, kết quả và nội dung của thuyết minh là quá trình mà nhóm sinh viên đã nỗ lực nghiên cứu, thực hiện và hoàn toàn trung thực. Các tài liệu tham khảo, các khóa học đều đã được trích dẫn đầy đủ và ghi rõ nguồn gốc. Chúng em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường nếu như có xảy ra vấn đề về liêm chính học thuật”.

Nhóm sinh viên thực hiện

Đỗ Bá Hải

Nguyễn Sơn Tùng

MỤC LỤC

TÓM TẮT	ii
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	iii
PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP	v
LỜI NÓI ĐẦU VÀ CẢM ƠN	vii
LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT	viii
MỤC LỤC	ix
DANH MỤC HÌNH ẢNH	xv
DANH MỤC BẢNG BIỂU	xix
DANH MỤC CÁC TỪ VIẾT TẮT	xx
MỞ ĐẦU.....	1
1. MỤC ĐÍCH THỰC HIỆN ĐỀ TÀI	1
2. MỤC TIÊU CỦA ĐỀ TÀI	2
3. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	2
4. PHƯƠNG PHÁP NGHIÊN CỨU	3
CHƯƠNG 1: TỔNG QUAN CỦA ĐỀ TÀI.....	4
1.1. Yêu cầu của bài toán	4
1.1.1. Thu thập dữ liệu điện năng từ đồng hồ Selec MFM383A.....	4
1.1.2. Giám sát từ xa.....	4
1.1.3. Tính toán sản lượng tiêu thụ điện trong từng khung giờ	4
1.1.4. Lưu trữ dữ liệu và truy xuất dữ liệu.....	4
1.1.5. Tạo cảnh báo và gửi Email về người dùng.....	5
1.1.6. Xây dựng chức năng phân quyền và quản lý người dùng	5
1.2. Các thông số cần được theo dõi trong đề tài	5

1.3. Quy định về các khung giờ cao điểm, thấp điểm & bình thường của EVN.	5
1.4. Các mức cảnh báo trong hệ thống	6
1.5. Tổng quan về cách kết nối hệ thống thu thập dữ liệu trên ThingsBoard	7
1.5.1. Kết nối PLC hãng Siemens với ThingsBoard	7
1.5.2. Kết nối PLC các hãng khác với ThingsBoard	7
1.6. Tổng quan về các giải pháp kết nối PLC với IoT ThingsBoard	8
1.6.1. Giải pháp kết nối thông qua máy tính làm Server	8
1.6.2. Giải pháp kết nối thông qua bộ chuyển đổi Industrial Iot Gateway	9
1.6.3. Giải pháp kết nối trực tiếp từ PLC lên IoT ThingsBoard	9
1.7. Tổng quan về các IoT ThingsBoard Cloud Server	10
1.8. Kết luận chương 1	11
CHƯƠNG 2: KIẾN TRÚC IOT CỦA HỆ THỐNG VÀ CÁC THIẾT BỊ SỬ DỤNG	12
2.1. Thiết kế kiến trúc IoT cho hệ thống	12
2.2. Các thành phần chính trong hệ thống	13
2.2.1. Đồng hồ đo điện năng Selec MFM383A	13
2.2.2. PLC Siemens S7 – 1214C DC/DC/DC	14
2.2.3. Module truyền thông CB 1241 RS485	14
2.2.4. Biến dòng vuông Selec	15
2.2.5. Miniature Circuit Breaker (MCB)	16
2.2.6. Bộ nguồn tổ ong 24V	17
2.2.7. Rơ le trạng thái rắn	17
2.3. Tổng hợp các thiết bị phần cứng của hệ thống	18
2.4. Các phần mềm sử dụng trong đề tài	19
2.4.1. Node.js	19
2.4.2. Visual Studio Code	19

2.4.3.	TIA Portal V17.....	19
2.4.4.	MySQL	20
2.5.	Các ngôn ngữ lập trình được sử dụng trong đề tài	20
2.5.1.	JavaScript.....	20
2.5.2.	TBEL (ThingsBoard Expression Language)	20
2.6.	Sơ đồ bản vẽ đi dây của hệ thống.....	21
2.7.	Kết luận chương 2.....	21
CHƯƠNG 3:	CÁC CƠ SỞ KIẾN THỨC LIÊN QUAN VÀ KẾT NỐI MQTT	
SERVER		22
3.1.	Giới thiệu về ThingsBoard và các thành phần chính.....	22
3.1.1.	Giới thiệu về ThingsBoard	22
3.1.2.	Entities và Relations	22
3.1.2.1.	<i>Entities (Thực thể)</i>	22
3.1.2.2.	<i>Realations (Quan hệ)</i>	23
3.1.3.	Device và Device Profile	23
3.1.3.1.	<i>Device (Thiết bị)</i>	23
3.1.3.2.	<i>Device Profile</i>	24
3.1.4.	Asset và Asset Profile	24
3.1.4.1.	<i>Asset (Tài sản)</i>	24
3.1.4.2.	<i>Asset Profile (Hồ sơ tài sản)</i>	24
3.1.5.	Alarm	25
3.1.6.	Rule Node và Rule Chain	25
3.1.6.1.	<i>Rule Node</i>	25
3.1.6.2.	<i>Rule Chain</i>	26
3.2.	Các giao thức truyền thông sử dụng trong đề tài	27
3.2.1.	Giao thức MQTT	27
3.2.1.1.	<i>Giới thiệu về MQTT</i>	27
3.2.1.2.	<i>Ưu điểm và nhược điểm của MQTT</i>	27

3.2.2.	Giao thức S7 Communication	28
3.2.3.	Giao thức SMTP	28
3.2.3.1.	<i>Giới thiệu giao thức SMTP.....</i>	28
3.2.3.2.	<i>Cách thức hoạt động của SMTP Server</i>	28
3.2.4.	Giao thức Modbus RTU.....	29
3.2.4.1.	<i>Định nghĩa giao thức Modbus RTU</i>	29
3.2.4.2.	<i>Cách thức hoạt động của Modbus RTU</i>	29
3.2.4.3.	<i>Ưu điểm và nhược điểm của Modbus RTU</i>	30
3.3.	Tạo dữ liệu mô phỏng từ dữ liệu thật và kết nối Modbus RTU	30
3.3.1.	Ý tưởng thực hiện	30
3.3.2.	Tạo Data Type và tạo các khối Data Block	31
3.3.3.	Tạo dữ liệu mô phỏng từ dữ liệu của đồng hồ thật.....	32
3.3.4.	Đọc dữ liệu về bằng Modbus RTU	33
3.4.	Kết nối MQTT Server với PLC S7-1200	36
3.4.1.	Cài đặt phần cứng cho PLC	36
3.4.2.	Kết nối để đọc dữ liệu từ PLC lên MQTT Server	36
3.5.	Kết nối MQTT Server với ThingsBoard để đưa dữ liệu lên Platform theo chuẩn MQTT	39
3.5.1.	Khai báo MQTT Server trên nền tảng IoT ThingBoard	39
3.5.2.	Kết nối ThingsBoard Server	39
3.5.3.	Gửi dữ liệu lên ThingsBoard	39
3.6.	Lưu trữ các dữ liệu đọc được từ PLC vào MySQL.....	40
3.6.1.	Khởi tạo cơ sở dữ liệu trong MySQL	40
3.6.2.	Khởi tạo kết nối dữ liệu với MySQL.....	41
3.7.	Kết luận chương 3.....	42
	CHƯƠNG 4: XÂY DỰNG CÁC CHỨC NĂNG VÀ THUẬT TOÁN XỬ LÝ DỮ LIỆU TRÊN THINGSBOARD	43

4.1. Tổng quan về xây dựng hệ thống thu thập dữ liệu trên ThingsBoard	43
4.2. Xây dựng chức năng Alarm (cảnh báo) trên ThingsBoard	44
4.2.1. Xử lý trong ThingsBoard để tạo cảnh báo đến giao diện người dùng	45
4.2.2. Xử lý Rule Chain để gửi E-mail đến người dùng	46
4.3. Truy xuất dữ liệu điện năng theo thời gian thực cho từng khung giờ	48
PLC 4.3.1. Xử lý dữ liệu điện năng theo từng khung giờ bằng cách viết code	49
4.3.1.1. Lưu đồ thuật toán	49
4.3.1.2. Lấy mốc thời gian thực	49
4.3.1.3. Tính công suất tiêu thụ trong khung giờ bình thường	50
4.3.1.4. Tính công suất tiêu thụ trong khung giờ cao điểm	51
4.3.1.5. Tính công suất tiêu thụ trong khung giờ thấp điểm	52
4.3.1.6. Giá trị công suất tại thời điểm đầu ngày, đầu tháng, đầu năm	52
4.3.1.7. Tính giá trị công suất tiêu thụ trong ngày	53
4.3.1.8. Tính giá trị công suất tiêu thụ trong tháng	54
4.3.1.9. Tính giá trị công suất tiêu thụ trong năm	55
4.3.1.10. Ưu và nhược điểm của phương pháp xử lý dữ liệu điện năng theo từng khung giờ bằng cách viết code PLC	56
4.3.2. Xử lý dữ liệu điện năng theo từng khung giờ trên ThingsBoard	57
4.3.2.1. Lưu đồ thuật toán	57
4.3.2.2. Xử lý dữ liệu điện năng được gửi lên từ MQTT Server	58
4.3.2.3. Tạo Asset để lưu trữ dữ liệu điện năng đọc từ Device	58
4.3.2.4. Xử lý dữ liệu điện năng trong Rule Chain	59
4.3.2.5. Xử lý dữ liệu điện năng trong Dashboard	59
4.3.2.6. Ưu và nhược điểm của phương pháp xử lý dữ liệu điện năng theo từng khung giờ trực tiếp trên ThingsBoard	61

4.4.	Xây dựng chức năng phân quyền và quản lý người dùng	61
4.5.	Kết luận chương 4.....	63
CHƯƠNG 5: KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ.....		64
5.1.	Kết quả mô hình của phần cứng	64
5.2.	Kết quả mô phỏng giá trị đồng hồ trong PLC	64
5.3.	Kết quả lưu dữ liệu vào MySQL	65
5.4.	Kết quả các Dashboard giám sát trong ThingsBoard	65
5.4.1.	Trang OVERVIEW	65
5.4.2.	Trang NOTIFICATION	66
5.4.3.	Trang ELECTRICITY	66
5.4.4.	Trang ENERGY	67
5.5.	Những mặt hạn chế của đề tài	68
KẾT LUẬN		69
TÀI LIỆU THAM KHẢO.....		70

DANH MỤC HÌNH ẢNH

Hình 1.1: Cách kết nối giữa PLC Siemens với IoT ThingsBoard qua MQTT Server.	7
Hình 1.2: Cách kết nối giữa PLC các hãng khác với IoT ThingsBoard qua MQTT Server.	8
Hình 1.3: Kiến trúc giải pháp kết nối thông qua máy tính.	8
Hình 1.4: Kiến trúc giải pháp kết nối thông qua IoT Gateway.	9
Hình 1.5: Kiến trúc giải pháp kết nối trực tiếp từ PLC lên ThingsBoard.	10
Hình 2.1: Kiến trúc IoT của hệ thống.	12
Hình 2.2: Sơ đồ đấu nối kiểu 1 pha – 2 dây của đồng hồ Selec MFM383A.	13
Hình 2.3: PLC S7 – 1214C DC/DC/DC.	14
Hình 2.4 : Module truyền thông CB 1241 RS485.	14
Hình 2.5: Biến dòng vuông Selec SPCT62/22-30/5.	15
Hình 2.6: Cấu tạo của biến dòng.	16
Hình 2.7: Nguyên lý hoạt động của biến dòng.	16
Hình 2.8: MCB và Nguyên lý hoạt động của MCB.	16
Hình 2.9: Bộ nguồn tổ ong 24V.	17
Hình 2.10: Cấu tạo của rơ le trạng thái rắn (SSR).	18
Hình 2.11: Phần mềm Node.js.	19
Hình 2.12: Phần mềm Visual Studio Code.	19
Hình 2.13: Phần mềm TIA Portal V17	20
Hình 2.14: Phần mềm MySQL.	20
Hình 2.15: Sơ đồ bản vẽ đi dây của phần cứng.	21
Hình 3.1: Kiến trúc cơ bản của nền tảng IoT ThingsBoard.	22
Hình 3.2: Mối quan hệ cơ bản giữa các thực thể.	23
Hình 3.3: Các Alarm trong ThingsBoard khi có sự cố xảy ra.	25
Hình 3.4: Ví dụ về 3 Node được kết nối với nhau qua mối quan hệ “True” và “False”.	25

Hình 3.5: Hình ảnh về một Rule Chain	26
Hình 3.6: Một ví dụ về mối quan hệ giữa MQTT Borker và MQTT Client.	27
Hình 3.7: Cách thức SMTP hoạt động.	29
Hình 3.8: Thiết bị Master và thiết bị Slave	29
Hình 3.9: Cấu trúc bản tin của Modbus RTU	30
Hình 3.10: Data Type lưu các thông số đọc được từ đồng hồ Selec MFM383A.	31
Hình 3.11: Đoạn chương trình tạo dữ liệu mô phỏng về điện áp	32
Hình 3.12: Đoạn chương trình tạo dữ liệu mô phỏng các thông số cần giám sát.	32
Hình 3.13: Dữ liệu trong Data Block sau khi được mô phỏng.	33
Hình 3.14: Cách thêm CB 1241 vào PLC trong Tia Portal V17	33
Hình 3.15: Các bước cấu hình cho CB 1241 (RS485).	34
Hình 3.16: Cài đặt các thông số về truyền thông.	34
Hình 3.17: Cài đặt thông số Wait time.	35
Hình 3.18: Thiết lập hàm MB_COMM_LOAD.....	35
Hình 3.19: Dùng khối MB_MASTER để đọc các dữ liệu từ đồng hồ.	36
Hình 3.20: Bật chức năng Full access của PLC.	36
Hình 3.21: Bật chức năng PUT/GET của PLC.....	36
Hình 3.22: Thiết lập các thông số của PLC.....	37
Hình 3.23: Các biến về thông số điện năng cần được gửi đi.....	37
Hình 3.24: Kết quả sau khi đọc được giá trị Tag lên MQTT Server.	38
Hình 3.25: Đoạn code khai báo MQTT Server trên ThingsBoard.	39
Hình 3.26: Đoạn code kết nối ThingsBoard Server.....	39
Hình 2.27: Đoạn code gửi dữ liệu lên ThingsBoard.	39
Hình 3.28: Kết quả dữ liệu về công suất tác dụng được gửi lên ThingsBoard. .	40
Hình 3.29: Tạo cơ sở dữ liệu trong MySQL.	40
Hình 3.30: Tạo bảng để chứa các dữ liệu từ PLC.	41

Hình 3.31: Đoạn code khởi tạo SQL.....	41
Hình 3.32: Hàm xử lý thời gian tại múi giờ Việt Nam.....	41
Hình 3.33: Hàm gọi và ghi dữ liệu vào MySQL.....	42
Hình 3.34: Hàm gọi dữ liệu qua MySQL sau mỗi 4s.....	42
Hình 4.1: Lưu đồ thuật toán chung của đề tài.	43
Hình 4.2: Tổng hợp dữ liệu gọi lên ThingsBoard tại mục Device.....	44
Hình 4.3: Đoạn code JavaScript về tạo cảnh báo mức Major trong Node Script.	45
Hình 4.4: Root Rule Chain để xử lý dữ liệu và tạo cảnh báo.....	45
Hình 4.5: Kết quả sau khi xử lý dữ liệu Alarm.	46
Hình 4.6: Thiết lập thông tin trong Node To E-mail.	46
Hình 4.7: Thiết lập các thông số về E-mail Google SMTP.	47
Hình 4.8: Rule Chain xử lý việc tạo và gọi E-mail.....	47
Hình 4.9: Password được SMTP Gmail của Google cấp.	47
Hình 4.10: Kết quả gọi cảnh báo về E-mail khi có sự cố xảy ra về điện áp.	48
Hình 4.11: Lưu đồ thuật toán tính điện năng tiêu thụ theo từng khung giờ bằng PLC.....	49
Hình 4.12: Khối hàm lấy mốc thời gian thực.....	49
Hình 4.13: Tính công suất tiêu thụ từ 4h đến 9h30.....	50
Hình 4.14: Tính tổng công suất tiêu thụ trong khung giờ bình thường.....	50
Hình 4.15: Tính công suất tiêu thụ khung giờ bình thường vào ngày chủ nhật.	51
Hình 4.17: Tính tổng công suất tiêu thụ trong khung giờ cao điểm.....	52
Hình 4.18: Tính công suất tiêu thụ khung giờ thấp điểm.....	52
Hình 4.19: Giá trị công suất đầu ngày, đầu tháng lần đầu bật.....	53
Hình 4.20: Giá trị công suất đầu năm lần đầu bật.	53
Hình 4.21: Tính công suất tiêu thụ trong ngày.	54
Hình 4.22: Lấy giá trị công suất đầu ngày.....	54
Hình 4.23: Tính công suất tiêu thụ trong tháng.....	55

Hình 4.24: Lấy giá trị công suất tại thời điểm đầu tháng.....	55
Hình 4.25: Tính công suất tiêu thụ trong năm.....	56
Hình 4.26: Lấy giá trị công suất tại thời điểm đầu năm.....	56
Hình 4.27: Lưu đồ thuật toán để tính và truy xuất dữ liệu điện năng theo từng khung giờ trong ThingsBoard.....	57
Hình 4.28: Rule Chain và đoạn code để lọc ra giá trị điện năng tiêu thụ.	58
Hình 4.29: Rule Chain để chia dữ liệu điện năng vào Asset.....	58
Hình 4.30: Kết quả dữ liệu điện năng đọc lên từ Device sau khi được xử lý vào Asset.	59
Hình 4.31: Rule Chain để tính và truy xuất dữ liệu điện năng theo thời gian. ..	59
Hình 4.31: Xử lý trong Dashboard với từng khung giờ cụ thể.	60
Hình 4.32: Kết quả cuối cùng sau khi xử lý trong Rule Chain.	60
Hình 4.33: Tạo Customer để quản lý người dùng.....	61
Hình 4.34: Tạo mật khẩu và tài khoản cho User.	62
Hình 4.35: Màn hình khi User được phân quyền đăng nhập vào.....	62
Hình 4.36: Các Dashboard được chỉ định dành riêng cho User.....	63
Hình 5.1: Mô hình phần cứng thực tế của hệ thống.....	64
Hình 5.2: Các thông số điện năng ở đồng hồ 2 sau khi mô phỏng giả lập.....	64
Hình 5.3: Dữ liệu được lưu trữ trong MySQL theo dạng bảng.....	65
Hình 5.4: Giao diện màn hình trang Overview.	65
Hình 5.5: Giao diện màn hình trang Notification.....	66
Hình 5.6: Giao diện màn hình trang Electricity.....	66
Hình 5.7: Kết quả in báo cáo dưới dạng file Excel.	67
Hình 5.8: Giao diện màn hình trang ENERGY.	67
Hình 5.9: Truy xuất dữ liệu điện năng từng khung giờ trong tháng.....	68
Hình 5.10: Kết quả in báo cáo điện năng tiêu thụ từng khung giờ dưới dạng Excel.	68

DANH MỤC BẢNG BIỂU

Bảng 1.1: Quy định về giờ cao điểm, thấp điểm và giờ bình thường của EVN. .	6
Bảng 1.2: Bảng mô tả một số mức cảnh báo cho từng thông số.	7
Bảng 1.3: Các thiết bị trong giải giáp kết nối thông qua máy tính.	8
Bảng 1.4: Các thiết bị trong giải pháp kết nối thông qua IoT Gateway.	9
Bảng 1.5: Bảng tổng hợp các IoT ThingsBoard Cloud Server.	10
Bảng 2.1: Bảng tổng hợp các thiết bị phần cứng của hệ thống.	18
Bảng 3.1: Bảng tổng hợp các nhóm Rule Node và chức năng của chúng.	26
Bảng 3.2: Địa chỉ thanh ghi các thông số của đồng hồ Selec trên Datasheet. ...	35
Bảng 3.3: Các thông số cần được thiết lập của PLC trong MQTT Server.	37
Bảng 3.4: Các biến về thông số điện năng cần được gửi đi.	38

DANH MỤC CÁC TỪ VIẾT TẮT

PLC	Programmable Logic Controller (Bộ điều khiển logic lập trình)
RTU	Remote Terminal Unit (Thiết bị đầu cuối từ xa)
TCP/IP	Transmission Control Protocol/ Internet Protocol (Giao thức điều khiển truyền nhận/ Giao thức liên mạng)
IoT	Internet of Things (Internet vạn vật)
IIoT	Industrial Internet of Things (Internet vạn vật công nghiệp)
SQL	Structured Query Language (Ngôn ngữ truy vấn có cấu trúc)
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
TBEL	ThingsBoard Expression Language
HTTP	HyperText Transfer Protocol (Giao thức truyền tải siêu văn bản)
SMTP	Simple Mail Transfer Protocol (Giao thức truyền tải thư tin đơn giản hóa)
CoAP	Constrained Application Protocol (Giao thức ứng dụng bị ràng buộc)
MCB	Miniature Circuit Breaker (Aptomat tép)
CT	Current Transformer (Biến dòng)
PT	Potential Transformer (Biến điện áp)
SSR	Solid State Relay (Rơ le trạng thái rắn)

MỞ ĐẦU

1. MỤC ĐÍCH THỰC HIỆN ĐỀ TÀI

Hiện nay tại Việt Nam nhu cầu sử dụng năng lượng điện không ngừng tăng cao do các hoạt động sản xuất công nghiệp quy mô lớn của các công ty FDI ngày càng được đầu tư nhiều vào đất nước ta cộng với việc giá cả tiền điện do Tập đoàn Điện lực Việt Nam (EVN) công bố gần đây đã có mức tăng đáng kể theo các năm đặc biệt là vào mùa nắng nóng cao điểm. Hơn thế nữa, tại Việt Nam thì EVN có đưa ra quy định về giá cả bán điện cho các xí nghiệp hay các nhà máy có nhu cầu hoạt động sản xuất theo hình thức điện ba giá đưa ra các quy định về khung giờ bao gồm: Khung giờ bình thường, khung giờ cao điểm và khung giờ thấp điểm. Nếu sử dụng điện vào từng thời điểm khác nhau thì sẽ được bán giá điện với những mức giá khác nhau tương ứng với từng khung giờ. Vì vậy việc giám sát năng lượng cũng như công suất tiêu thụ tại từng khung giờ ở các xí nghiệp hay các nhà máy hoạt động sản xuất tại Việt Nam là điều vô cùng quan trọng và cấp thiết đối với các người chủ đầu tư. Việc này giúp cho người chủ đầu tư có thể có cái nhìn tổng quan về công suất tiêu thụ từ đó bố trí nhân công hay sắp xếp các hoạt động sản xuất ở thời gian phù hợp để tối ưu hóa về các chi phí sử dụng điện cho việc sản xuất.

Mặc khác thì các hệ thống giám sát truyền thống hiện nay thường thiếu tính linh hoạt và khó mở rộng và không đáp ứng được nhu cầu theo dõi dữ liệu thời gian thực cũng như phân tích chi tiết để tối ưu hóa việc sử dụng năng lượng. Hoặc để xây dựng một hệ giám sát để có thể theo dõi được dữ liệu thời gian thực từ xa thì mất nhiều công sức và kinh phí, đòi hỏi người xây dựng phải có kiến thức tốt về công nghệ thông tin.

Vì vậy nhóm sinh viên chúng em quyết định thực hiện đề tài “*Xây dựng hệ thống giám sát và thu thập dữ liệu điện năng trên nền tảng ThingsBoard*”. Đề tài này bên cạnh việc cung cấp một giải pháp giám sát điện năng toàn diện mà còn có thể tận dụng ưu điểm vượt trội của ThingsBoard – một nền tảng IoT mã nguồn mở, dễ dàng triển khai và mở rộng. Với nền tảng IoT ThingsBoard này thì hệ thống có khả năng hiển thị dữ liệu thời gian thực, vẽ biểu đồ và tính toán khối lượng sử dụng điện năng ở các khung giờ bình thường, khung giờ cao điểm và khung giờ thấp điểm trong ngày, phát hiện và cảnh báo kịp thời khi các giá trị về điện áp hay dòng điện vượt ngưỡng qua E-mail người quản lý, giúp chủ động hơn trong việc quản lý và tối ưu hóa việc tiêu thụ năng lượng.

Sử dụng ThingsBoard mang lại nhiều lợi ích như hỗ trợ được nhiều giao thức mạng như MQTT, HTTP, CoAP,... Giao diện trực quan, khả năng tích hợp hệ thống một cách dễ dàng và không cần phải xây dựng hệ thống lại từ đầu, tiết kiệm được chi phí cho chủ đầu tư. Điều này đáp ứng được xu hướng chuyển đổi số và quản lý thông minh trong lĩnh vực sử dụng năng lượng trong bối cảnh hiện nay.

2. MỤC TIÊU CỦA ĐỀ TÀI

Mục tiêu của đề tài “*Xây dựng hệ thống giám sát và thu thập dữ liệu điện năng trên nền tảng ThingsBoard*” này là xây dựng một mô hình hệ thống tiêu thụ điện năng, từ đó xây dựng một hệ thống giám sát và thu thập dữ liệu về các thông số từ đồng hồ đo điện Selec MFM383A của hãng Selec Controls và một số dữ liệu mô phỏng được tạo ra từ dữ liệu thật của đồng hồ Selec MFM383A từ PLC. Hệ thống cung cấp một cách nhìn trực quan về tình trạng tiêu thụ điện năng của mô hình, giúp người giám sát hoặc chủ đầu tư có cái nhìn tổng quát và đưa ra các giải pháp phân phối sử dụng điện sao cho tối ưu nhất cho các xí nghiệp hay các nhà máy.

Ngoài ra đề tài còn giúp cho sinh viên nắm vững, hiểu rõ các thức hoạt động của hệ thống giám sát và thu thập dữ liệu cũng như các giao thức truyền thông công nghiệp và giao thức truyền thông IoT. Từ đó xây dựng chương trình thu thập dữ liệu, mô phỏng dữ liệu, xây dựng các thuật toán xử lý dữ liệu và lập trình tính toán cho việc phân chia dữ liệu vào từng khung giờ.

3. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đối tượng nghiên cứu của đề tài này là nền tảng IoT ThingsBoard, MQTT Server, đồng hồ đo điện đa năng Selec MFM383A, PLC S7-1200, các thiết bị đo cùng với các giao thức truyền thông công nghiệp và giao thức truyền thông IoT. Trong đó chủ yếu tập trung chính và nền tảng IoT ThingsBoard và MQTT Server vì đây là thành phần chính để hoàn thiện đề tài.

Phạm vi nghiên cứu của đề tài này là xây dựng một Web giám sát và thu thập dữ liệu điện năng của đồng hồ Selec MFM383A từ PLC từ đó xử lý dữ liệu điện năng bằng việc ứng dụng những công nghệ tích hợp sẵn trên nền tảng IoT ThingsBoard. Bên cạnh đó hệ thống còn có các chức năng đầy đủ của một hệ giám sát như cảnh báo lỗi và gửi cảnh báo về E-mail người dùng, thu thập dữ liệu điện năng theo thời gian thực, truy xuất và lưu trữ dữ liệu dưới dạng file Excel hoặc dạng bảng, phân quyền và quản lý người dùng và cung cấp một giao diện người dùng trực quan.

4. PHƯƠNG PHÁP NGHIÊN CỨU

Phương pháp nghiên cứu của đề tài này bao gồm:

- Xây dựng một mô hình nhỏ để tích hợp tất cả các thiết bị cần sử dụng trong đề tài với tải tiêu thụ điện năng là một bóng đèn LED có công suất 40W.
- Sử dụng PLC S7-1214C DC/DC/DC để đọc dữ liệu điện năng thông qua giao thức truyền thông công nghiệp Modbus RTU bên cạnh đó còn sử dụng PLC cho việc tạo các thông số mô phỏng về dữ liệu dựa trên dữ liệu thật từ đồng hồ đo điện.
- Sử dụng phần mềm Visual Studio Code cho việc nghiên cứu và cấu hình cho MQTT Server để đọc dữ liệu từ PLC từ đó đưa dữ liệu lên nền tảng ThingsBoard thông qua giao thức truyền thông MQTT.
- Nghiên cứu và phát triển hệ thống từ đó xây dựng giao diện trên nền tảng ThingsBoard để hoàn thiện và đáp ứng được các yêu cầu, tính năng cơ bản cần thiết cho một hệ thống giám sát và thu thập dữ liệu.

5. CẤU TRÚC CỦA ĐỀ TÀI

Thuyết minh đề tài được phân thành cấu trúc bao gồm 5 chương chính:

Chương 1: Trình bày tổng quan về đề tài và các lý thuyết liên quan đến vấn đề sử dụng điện năng trong các hoạt động sản xuất, các giải pháp kết nối và các mức cảnh báo được nhóm sinh viên sử dụng trong đề tài.

Chương 2: Trình bày về kiến trúc IoT của hệ thống, sơ đồ đấu nối, các phần cứng và các phần mềm cũng như các ngôn ngữ lập trình tương ứng với các phần mềm được sử dụng trong đề tài.

Chương 3: Trình bày về các kiến thức liên quan đến nền tảng IoT ThingsBoard, các giao thức truyền thông công nghiệp được sử dụng trong đề tài và cách kết nối MQTT Server.

Chương 4: Trình bày quá trình thiết kế và xây dựng các chức năng của hệ thống giám sát và thuật toán xử lý dữ liệu trên nền tảng.

Chương 5: Trình bày kết quả đạt được và đưa ra đánh giá.

CHƯƠNG 1: TỔNG QUAN CỦA ĐỀ TÀI

1.1. Yêu cầu của bài toán

Bài toán được đặt ra là phải xây dựng một hệ thống có đầy đủ các chức năng của một hệ thống giám sát và thu thập dữ liệu cơ bản. Các yêu cầu được trình bày cụ thể như sau:

1.1.1. Thu thập dữ liệu điện năng từ đồng hồ Selec MFM383A

Tích hợp bộ điều khiển lập trình PLC để thu thập các dữ liệu thông số từ đồng hồ đo sau đó gửi lên nền tảng IoT ThingsBoard thông qua một máy tính làm Server trung gian. Các thông số này bao gồm: Điện áp, Dòng điện, Tần số, Công suất tác dụng, Công suất tiêu thụ tích lũy.

1.1.2. Giám sát từ xa

Tích hợp hệ thống giám sát từ xa cho phép theo dõi chi tiết các thông số đọc được từ đồng hồ dưới dạng bảng hoặc dưới dạng biểu đồ trực quan thông qua Internet.

Xây dựng giao diện người dùng (Dashboard) thân thiện, dễ sử dụng để giám sát và quản lý thông qua nền tảng IoT ThingsBoard hoặc các ứng dụng di động.

1.1.3. Tính toán sản lượng tiêu thụ điện trong từng khung giờ

Xây dựng thuật toán tính toán được công suất tiêu thụ điện năng theo từng khung giờ mong muốn (khung giờ bình thường, khung giờ cao điểm và khung giờ thấp điểm).

Tính toán được tiền điện tương ứng với từng khung giờ mong muốn.

1.1.4. Lưu trữ dữ liệu và truy xuất dữ liệu

Hệ thống khi vận hành phải lưu trữ được dữ liệu vào đám mây hay cơ sở dữ liệu của nền tảng IoT ThingsBoard để đảm bảo tính linh hoạt và khả năng mở rộng cũng như khả năng truy cập từ xa. Ngoài ra thì các dữ liệu phải được sao lưu định kỳ vào MySQL để phòng ngừa việc mất thông tin do sự cố hệ thống cũng như đảm bảo về mặt bảo mật của dữ liệu.

Hệ thống còn phải truy xuất được dữ liệu và in báo cáo về các dữ liệu thu thập được hay công suất tiêu thụ từng khung giờ dưới dạng file Excel hoặc file JSON.

1.1.5. Tạo cảnh báo và gửi Email về người dùng

Khi có sự cố về dòng điện hay điện áp vượt quá ngưỡng cho phép mà nhóm sinh viên đã đưa ra thì hệ thống sẽ ngay lập tức gửi cảnh báo lên giao diện người dùng Dashboard và đồng thời gửi cảnh báo về Email người dùng với chi tiết của cảnh báo như loại cảnh báo, mức độ nguy hiểm, trạng thái cảnh báo, thời gian cảnh báo,...

1.1.6. Xây dựng chức năng phân quyền và quản lý người dùng

Chức năng phân quyền giúp ngăn chặn việc truy cập trái phép vào các dữ liệu quan trọng cũng như giới hạn quyền truy cập theo vai trò của người dùng, chỉ cho phép thực hiện cách hành động phù hợp được chỉ định riêng cho từng người.

Việc phân quyền linh hoạt giúp quản lý người dùng tốt hơn đặc biệt là khi có nhiều người dùng thuộc các phòng ban khác nhau (kỹ thuật, vận hành, phân tích dữ liệu).

1.2. Các thông số cần được theo dõi trong đề tài

Trong đề tài này do có sự giới hạn về mặt phần cứng của hệ thống nên đồ án sử dụng tải tiêu thụ năng lượng là bóng đèn LED 40W. Do đó những thông số thu thập và theo dõi trong đề tài này chỉ mang tính chất đại diện, những thông số đó bao gồm:

- Điện áp pha (V)
- Dòng trên pha (A)
- Tần số hoạt động (Hz)
- Công suất tác dụng (kW)
- Công suất tiêu thụ tích lũy (kWh)

1.3. Quy định về các khung giờ cao điểm, thấp điểm & bình thường của EVN.

Khác với việc tính tiền điện theo giá bậc thang cho các nhu cầu sinh hoạt hoặc kinh doanh bình thường thì đối với các nhà máy, nhà xưởng, xí nghiệp hoặc các công ty liên quan đến các hoạt động sản xuất hàng hóa hay kinh doanh dịch vụ thì tổng công ty điện lực Việt Nam EVN có quy định rõ thêm các khoảng tiền khác nhau cho từng khung giờ khác nhau như sau:

Bảng 1.1: Quy định về giờ cao điểm, thấp điểm và giờ bình thường của EVN.

	Giờ bình thường	Giờ cao điểm	Giờ thấp điểm
Thứ 2 đến thứ 7	• Từ 4h00 đến 9h30 • Từ 11h30 đến 17h00 • Từ 20h00 đến 22h00	• Từ 9h30 đến 11h30 • Từ 17h00 đến 20h00	• Tất cả các ngày trong tuần, bắt đầu từ 22h00 đến 4h00 sáng hôm sau.
Ngày chủ nhật	• Từ 4h00 đến 22h00	• Không có giờ cao điểm	

Các đối tượng khách hàng mua điện theo hình thức ba giá này là:

- Những khách hàng sử dụng cho mục đích sản xuất kinh doanh, dịch vụ được cấp điện qua máy biến áp chuyên dùng từ 25kVA trở lên hoặc có sản lượng tiêu thụ điện trung bình 3 tháng liên tục từ 2000 kWh/ tháng trở lên.
- Đơn vị bán lẻ điện tại khu công nghiệp.
- Đơn vị mua điện để bán lẻ điện ngoài mục đích sinh hoạt tại tổ hợp thương mại - dịch vụ - sinh hoạt. [1]

1.4. Các mức cảnh báo trong hệ thống

Trong thực tế, các nguyên nhân gây ra sự cố quá dòng trong nhà máy thường bao gồm những nguyên nhân sau đây:

- Khởi động động cơ lớn: Khi khởi động trực tiếp các động cơ công suất lớn có thể làm dòng khởi động tăng gấp 5 – 7 lần dòng định mức trong thời gian ngắn.
- Ngắn mạch là nguyên nhân phổ biến nhất khi dòng điện tăng đột ngột do điện trở mạch thấp khi các dây pha chạm nhau.
- Quá tải thiết bị: Khi thiết bị tiêu thụ dòng lớn hơn định mức lâu dài (động cơ làm việc quá công suất) thì dòng tăng cao gây nóng dây dẫn và thiết bị.

Các nguyên nhân gây ra sự cố quá áp trong nhà máy thường bao gồm những nguyên nhân sau đây:

- Đứt dây trung tính trong hệ thống dẫn đến việc mất cân bằng điện áp giữa các pha, gây quá áp cục bộ trên một số thiết bị.
- Xung điện cao thế do sét đánh hoặc sự cố lưới điện truyền qua đường dây dẫn gây tăng áp đột ngột.

- Khi tải lớn đang hoạt động mà bị cắt ra đột ngột thì công suất phản kháng (đặc biệt từ cuộn dây) không tiêu tán hết sẽ gây tăng áp.

Trong đề tài lần này, việc tạo cảnh báo sẽ được giả lập bằng cách thay đổi các giá trị giám sát (Monitor Value) về điện áp pha và dòng trên pha qua phần mềm TIA Portal sao cho các giá trị này vượt các ngưỡng quy định của bảng dưới đây. Các mức cảnh báo trong đề tài sẽ chia thành 2 loại gồm: Major (mức cao) và Critical (mức nguy hiểm).

Bảng 1.2: Bảng mô tả một số mức cảnh báo cho từng thông số.

Loại dữ liệu	Major	Critical
Điện áp pha	250 V	300 V
Điện áp dây	50 A	100 A

1.5. Tổng quan về cách kết nối hệ thống thu thập dữ liệu trên ThingsBoard

1.5.1. Kết nối PLC hãng Siemens với ThingsBoard

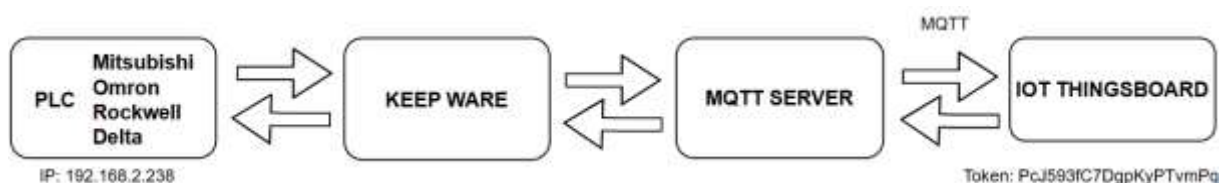
Các dữ liệu về điện năng được thu thập từ PLC Siemens sẽ được gửi đến MQTT Server qua giao thức truyền thông S7 Communication với địa chỉ IP của PLC. Sau đó dữ liệu tiếp tục được gửi lên nền tảng IoT ThingsBoard thông qua giao thức MQTT. Mỗi thiết bị tương ứng với tài khoản ThingsBoard sẽ có một mã Access Token riêng. Mã Access Token này rất quan trọng, dựa vào mã này ta có thể lập trình gửi dữ liệu lên nền tảng IoT ThingsBoard một cách chính xác.



Hình 1.1: Cách kết nối giữa PLC Siemens với IoT ThingsBoard qua MQTT Server.

1.5.2. Kết nối PLC các hãng khác với ThingsBoard

Đối với các dòng PLC của các hãng (ngoại trừ PLC Siemens) như PLC Mitsubishi, Omron, Rockwell hoặc Delta,... thì phải thông qua phần mềm OPC trung gian là KEPLWARE mà không thể kết nối trực tiếp từ PLC đến MQTT Server giống như của hãng Siemens được. Điều này cũng dễ hiểu vì PLC hãng Siemens có chuẩn truyền thông công nghiệp S7 Communication được hãng phát triển độc quyền dành cho các dòng PLC Siemens. Tận dụng ưu điểm này từ đó chọn PLC Siemens để nghiên cứu và sử dụng trong đề tài lần này.



Hình 1.2: Cách kết nối giữa PLC các hãng khác với IoT ThingsBoard qua MQTT Server.

1.6. Tổng quan về các giải pháp kết nối PLC với IoT ThingsBoard

Có nhiều giải pháp để kết nối từ PLC với nền tảng IoT ThingsBoard, mỗi giải pháp đều có ưu và nhược điểm riêng và mức độ phù hợp với từng yêu cầu của dự án được đặt ra.

1.6.1. Giải pháp kết nối thông qua máy tính làm Server

Máy tính sẽ làm điểm trung gian để kết nối với PLC, thu thập dữ liệu từ PLC thông qua MQTT Server.

Bảng 1.3: Các thiết bị trong giải pháp kết nối thông qua máy tính.

STT	Thiết bị	Số tiền
1	Máy tính (Máy bàn, laptop,...)	Tùy cấu hình
2	Cloud IoT ThingsBoard	Free nếu hệ thống nhỏ, 10\$/1 tháng cho hệ thống lớn

Ưu điểm:

- Tiết kiệm chi phí, tận dụng được máy tính có sẵn để làm Server trung gian.

Nhược điểm:

- Máy tính phải hoạt động và treo máy liên tục để có thể đảm bảo được quá trình truyền nhận dữ liệu từ các bên. Điều khiến máy nhanh bị nóng và có thể ảnh hưởng đến quá trình hoạt động của hệ thống.



Hình 1.3: Kiến trúc giải pháp kết nối thông qua máy tính.

1.6.2. Giải pháp kết nối thông qua bộ chuyển đổi Industrial Iot Gateway

Ở giải pháp này ta sử dụng một Iot Gateway làm điểm trung gian giúp gửi dữ liệu từ PLC lên IoT ThingsBoard bằng MQTT hoặc HTTP. Giải pháp này thường được sử dụng cho các hệ thống ở ngoài những nơi xa xôi như ngoài ruộng hoặc trên núi, những nơi mà PLC không thể kết nối được với máy tính thì đây là một giải pháp phù hợp.

Bảng 1.4: Các thiết bị trong giải pháp kết nối thông qua IoT Gateway.

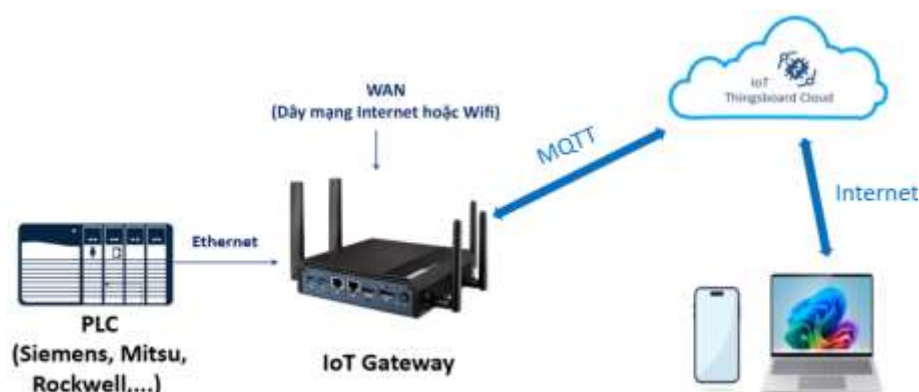
STT	Thiết bị	Số tiền
1	Bộ chuyển đổi Industrial IoT Gateway	Cao
2	Cloud IoT ThingsBoard	Free nếu hệ thống nhỏ, 10\$/1 tháng cho hệ thống lớn

Ưu điểm:

- Vì sử dụng một bộ chuyển đổi Industrial Iot Gateway nên hệ thống sẽ trở nên nhỏ gọn và linh hoạt hơn.
- Là một giải pháp toàn diện được sử dụng rộng rãi trong nhiều hệ thống phức tạp hoặc những nơi mà không có máy tính.
- Có thể sử dụng SIM 5G ở những nơi không có mạng.

Nhược điểm:

- Chi phí đầu tư ban đầu cao.



Hình 1.4: Kiến trúc giải pháp kết nối thông qua IoT Gateway.

1.6.3. Giải pháp kết nối trực tiếp từ PLC lên IoT ThingsBoard

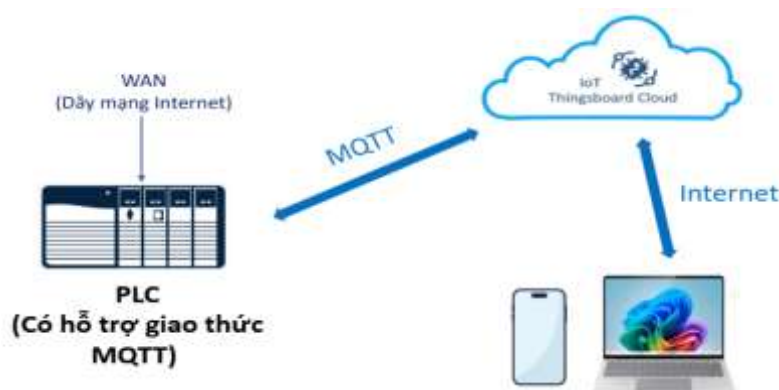
Ở giải pháp này ta sử dụng PLC có hỗ trợ các hàm bức điện MQTT có thể kết nối với ThingsBoard thông qua MQTT trực tiếp từ PLC.

Ưu điểm:

- Nhanh gọn, ít tốn các thiết bị làm trung gian việc truyền nhận dữ liệu.
- Chi phí thấp.

Nhược điểm:

- Giải pháp này được sử dụng tương đối ít vì thông thường PLC được dùng để xử lý các tín hiệu điều khiển hay thu thập dữ liệu từ các thiết bị ở cấp trường, vì vậy khi ta thêm chức năng MQTT sẽ làm cho bộ xử lý của PLC vận hành tương đối nặng, ảnh hưởng đến chương trình điều khiển.



Hình 1.5: Kiến trúc giải pháp kết nối trực tiếp từ PLC lên ThingsBoard.

1.7. Tổng quan về các IoT ThingsBoard Cloud Server

Bảng 1.5: Bảng tổng hợp các IoT ThingsBoard Cloud Server.

STT	Loại Cloud	Mô tả đặc điểm	Ưu điểm	Nhược điểm
1	Mạng nội bộ	Sử dụng một máy tính cài ThingsBoard làm Cloud Server. PLC kết nối với Server Cloud nội bộ thông qua dây mạng Ethernet hoặc Wifi.	Không mất chi phí mua Cloud.	Chỉ dùng được trong mạng nội bộ của công ty và không điều khiển được qua Internet.
2	Cloud của hãng ThingsBoard	Hãng ThingsBoard đã tạo sẵn Cloud, người dùng có thể kết nối PLC đến Cloud của hãng.	Điều khiển được qua Internet.	Mất chi phí mua Cloud hàng tháng. Nếu hệ thống nhỏ có thể sử dụng bản Demo miễn phí.

3	Cloud bên thứ ba	Người dùng phải mua Cloud của Google hoặc Amazon sau đó cài ThingsBoard trên Cloud đó.	Điều khiển được qua Internet.	Mất chi phí mua Cloud hàng tháng.
4	Cloud tự tạo qua IP Tĩnh	Dùng một máy tính làm Server và cài ThingsBoard trên đó. Sau đó mở cổng (NAT PORT) ra Internet.	Điều khiển được qua Internet.	Mất chi phí mua IP tĩnh.

Dựa vào các đặc điểm cũng như ưu và nhược điểm của bảng tổng hợp các IoT Cloud trên nên đề án này sẽ chọn Cloud của hãng ThingsBoard để sử dụng trong đề tài lần này vì nó phù hợp với yêu cầu của dự án lần này và chi phí để sử dụng Cloud của hãng hàng tháng thì cũng khá hợp lý.

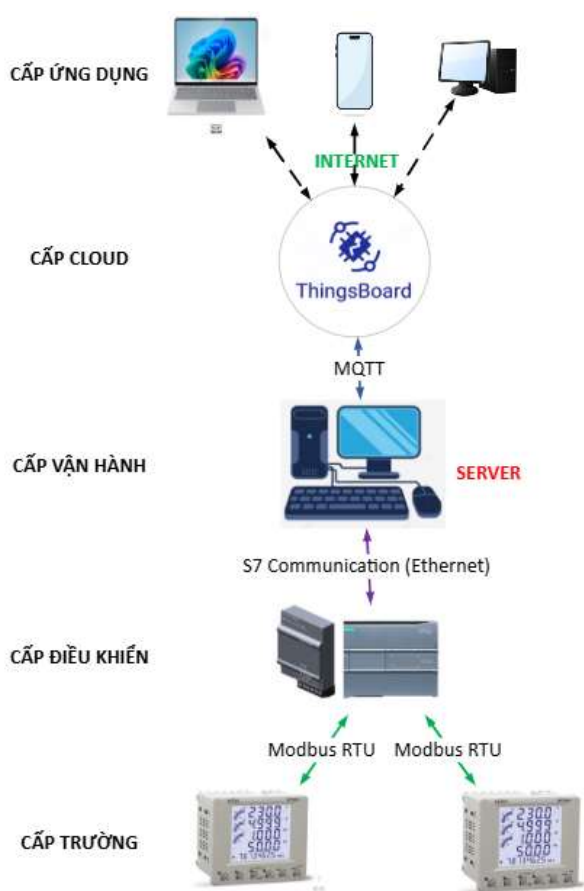
1.8. Kết luận chương 1

Trong chương 1 này đã trình bày tổng quan về yêu cầu của bài toán và các vấn đề sử dụng điện trong sản xuất theo quy định của EVN. Đồng thời cũng trình bày tổng quan các giải pháp kết nối của một hệ thống IoT và lựa chọn được IoT ThingsBoard Cloud Server phù hợp với kiến trúc cũng như yêu cầu của dự án.

CHƯƠNG 2: KIẾN TRÚC IOT CỦA HỆ THỐNG VÀ CÁC THIẾT BỊ SỬ DỤNG

2.1. Thiết kế kiến trúc IoT cho hệ thống

Từ việc nêu ra các giải pháp kết nối cho hệ thống và từ các yêu cầu kỹ thuật của dự án thì đồ án đã đưa ra lựa chọn và kiến trúc IoT phù hợp.



Hình 2.1: Kiến trúc IoT của hệ thống.

Cấp trường:

- Gồm các thiết bị: Đồng hồ Selec MFM838A.
- Chức năng: Dùng để đo các dữ liệu về thông số điện năng.

Cấp điều khiển:

- Gồm các thiết bị: PLC S7 1200 và Module truyền thông CB 1241 RS485.
- Chức năng: Kết nối và thu thập các dữ liệu từ đồng hồ đo thông qua giao thức truyền thông Modbus RTU với cổng RS-485.

Cấp vận hành:

- Gồm 1 máy tính (PC) hoạt động như 1 Server.
- Chức năng: Server kết nối với PLC thông qua giao thức S7 Communication ở cổng Ethernet, thu thập dữ liệu đã xử lý từ PLC, bên cạnh đó còn đóng vai trò là MQTT Client để gửi dữ liệu lên ThingsBoard.

Cấp Cloud:

- Nền tảng chính: Nền tảng IoT ThingsBoard hỗ trợ giao thức MQTT.
- Chức năng: Nhận dữ liệu từ Server thông qua giao thức MQTT.

Cấp ứng dụng:

- Gồm các ứng dụng: hệ thống giám sát trên máy tính hoặc điện thoại,...
- Chức năng: Đây là giao diện người dùng cuối và cung cấp giao diện trực quan để theo dõi, phân tích và đưa ra quyết định.

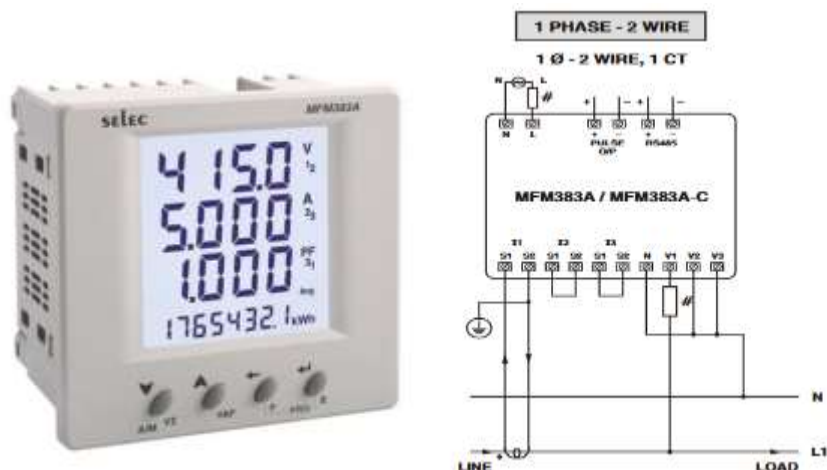
2.2. Các thành phần chính trong hệ thống

2.2.1. Đồng hồ đo điện năng Selec MFM383A

Đồng hồ đo điện Selec MFM383A là thiết bị đo lường và giám sát điện năng phổ biến trong các hệ thống quản lý năng lượng công nghiệp. Thiết bị còn tích hợp truyền thông Modbus RTU, cho phép dễ dàng giao tiếp với các thiết bị điều khiển như PLC.

Thông số kỹ thuật:

- Cài đặt hệ số CT: 1A/5A – 10.000A
- Mạng kết nối: 3 pha – 4 dây, 3 pha – 3 dây, 1 pha – 2 dây
- Cài đặt hệ số PT Sơ cấp/PT Thứ cấp: 100V – 10 KV/100 – 500V AC
- Nguồn cấp: 85 – 270V AC



Hình 2.2: Sơ đồ đấu nối kiểu 1 pha – 2 dây của đồng hồ Selec MFM383A.

2.2.2. PLC Siemens S7 – 1214C DC/DC/DC

PLC S7 – 1214C DC/DC/DC thuộc dòng PLC S7 – 1200 của Siemens, phù hợp cho các hệ thống tự động hoá, các hệ thống sản xuất trong công nghiệp. Ngoài ra còn có chức năng tích hợp các giao thức truyền thông công nghiệp như Profinet, Profibus, Modbus RTU,... Cho phép người dùng dễ dàng kết nối với các thiết bị ngoại vi.

Thông số kỹ thuật:

- Điện áp đầu vào: 20.4V đến 28.8V DC
- Số lượng đầu vào số (DI): 14 đầu vào 24V DC
- Số lượng đầu ra số (DO): 10 đầu ra 24V DC
- Số lượng đầu vào Analog (AI): 2 đầu vào 0 – 10V DC
- Khả năng kết nối: Truyền thông S7 Communication, Profinet hoặc Ethernet.



Hình 2.3: PLC S7 – 1214C DC/DC/DC.

2.2.3. Module truyền thông CB 1241 RS485

Module truyền thông CB 1241 RS485 là một phụ kiện đi kèm cho dòng S7-1200, có chức năng giúp PLC mở rộng khả năng truyền thông qua thức Modbus RTU với cổng RS485, giúp PLC dễ dàng kết nối với các thiết bị ngoại vi cấp trường trong công nghiệp như cảm biến, đồng hồ đo,...



Hình 2.4 : Module truyền thông CB 1241 RS485.

2.2.4. Biến dòng vuông Selec

Biến dòng là thiết bị sử dụng trong các hệ thống đo lường và bảo vệ. Có nhiệm vụ là chuyển đổi dòng điện lớn ở mạch sơ cấp (primary) thành dòng điện nhỏ ở mạch thứ cấp (secondary) theo một tỷ lệ nhất định.

- Tính chọn biến dòng dựa theo dòng của tải:

Tải tiêu thụ điện năng trong đề tài này là bóng đèn LED có công suất 40W vì vậy cần tính chọn biến dòng phù hợp với dòng của tải ở điện áp 1 pha – 2 dây.

$$\cos \theta = 0,8$$

$$P = UI \cos \theta \quad (2.1)$$

$$\Rightarrow I = \frac{P}{U \cdot \cos \theta} = \frac{40}{220 \cdot 0,8} = 0,23(A)$$

⇒ Chọn biến dòng có tỉ số 30/5A

Thông số kỹ thuật:

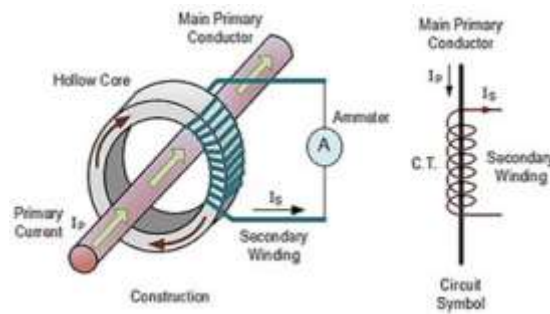
- Dòng điện sơ cấp 30A
- Dòng điện thứ cấp 5A
- Tần số hoạt động: 50 ~ 60Hz



Hình 2.5: Biến dòng vuông Selec SPCT62/22-30/5.

Biến dòng gồm các thành phần chính như sau:

- Primary Current: Dòng điện sơ cấp
- Secondary Winding: Cuộn dây thứ cấp
- Ammeter: Đồng hồ đo dòng
- Hollow Core: Lõi rỗng



Hình 2.6: Cấu tạo của biến dòng.

Nguyên lý hoạt động của biến dòng dựa trên hiện tượng cảm ứng điện từ. Khi có dòng điện xoay chiều chạy qua dây dẫn thì xung quanh nó sẽ xuất hiện một điện trường khi đó điện trường này sẽ cảm ứng lên cuộn dây và xuất hiện một dòng điện trong đó. Tỉ số biến dòng được dựa vào số vòng dây được quấn trong cuộn dây biến dòng.



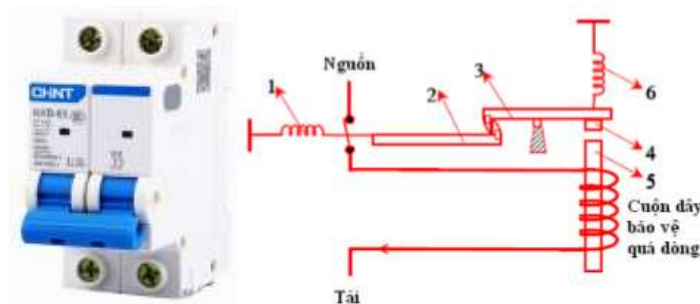
Hình 2.7: Nguyên lý hoạt động của biến dòng.

2.2.5. *Miniature Circuit Breaker (MCB)*

Dùng để đóng ngắt mạch điện, bảo vệ hệ thống điện và các khí cụ điện khác của mạch điện trong trường hợp quá tải, sụt áp hoặc ngắn mạch.

Thông số kỹ thuật:

- Điện áp định mức: 240/415V
- Dòng điện định mức: 10A
- Dòng cắt ngắn mạch: 6KA
- Số cực: 2



Hình 2.8: MCB và Nguyên lý hoạt động của MCB.

Khi đóng điện thì MCB sẽ được giữ ở trạng thái đóng tiếp điểm nhờ móc khớp 2 với móc 3 cùng chung tiếp điểm động. Sau khi bật MCB thì dòng định mức ở nam châm điện 5 và phần ứng 4 sẽ hút không khí lại. Khi quá tải hoặc ngắn mạch thì lực hút điện từ ở nam châm điện 5 sẽ hút phần ứng 4 làm móc 3 và móc 5 sẽ bật nhả ra khi đó lò xo 1 được thả lỏng và các tiếp điểm của MCB được mở ra làm cho mạch điện bị ngắt.

2.2.6. Bộ nguồn tổ ong 24V

Là một dạng bộ nguồn DC chuyển mạch để cung cấp điện áp 24V DC cho các hệ tự động hóa yêu cầu nguồn cấp DC.

Thông số kỹ thuật:

- Nguồn cấp: 100 ~ 240V AC, 50/60 Hz, 1 pha
- Điện áp ngõ ra: 24V DC. Có thể điều chỉnh -10% đến 10% bằng biến trở
- Dòng điện ngõ ra: 2.2A



Hình 2.9: Bộ nguồn tổ ong 24V.

Nguồn tổ ong hoạt động dựa trên nguyên tắc sử dụng các bộ điều chỉnh hay Transitor công suất để tạo ra điện áp mong muốn. Ban đầu khi ta cho nguồn xoay chiều 220V đi qua khi đó biến áp sơ cấp được điều khiển đóng cắt bằng sơ công suất tạo ra từ trường biến thiên. Cuộn biến áp thứ cấp sinh ra điện áp sau đó được chỉnh lưu và lọc bởi các cuộn lọc nhiễu. Cuối cùng nhờ vào IC quang và IC TL431 điều chỉnh dòng điện để duy trì ổn định điện áp cấp cho tải.

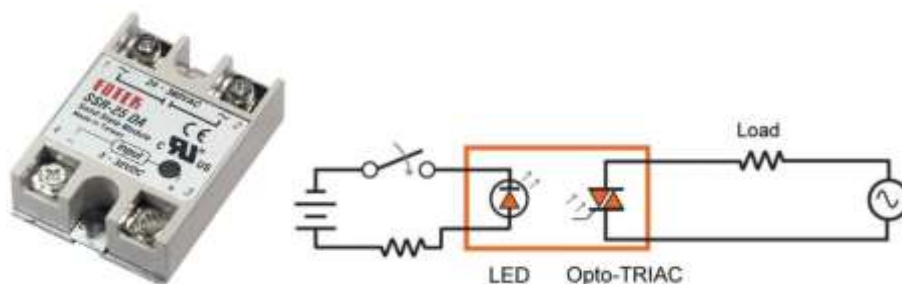
2.2.7. Rơ le trạng thái rắn

Rơ le trạng thái rắn (SSR) là thiết bị mà khi có điện áp bên ngoài (AC hoặc DC) được áp dụng trên các đầu cuối điều khiển của nó thì nó sẽ chuyển mạch điện tử từ bật sang tắt hoặc ngược lại.

Thông số kỹ thuật:

- Điện áp đầu vào: 3-32V DC
- Điện áp ngõ ra: 24-380V AC

- Dòng tối đa: 25A
- Dòng tiêu thụ: < 2mA



Hình 2.10: Cấu tạo của rơ le trạng thái rắn (SSR).

Rơ le trạng thái rắn SSR có cấu tạo gồm các Diot phát quang và bộ Tri-ac. Khác với rơ le truyền thống hay Contactor ở chỗ là chúng không có bộ phận chuyển động để đóng ngắt dòng điện nên nó có tuổi thọ dài hơn so với các loại rơ le truyền thống. Nó hoạt động theo nguyên lý điều khiển bằng tín hiệu điện sau đó chuyển đổi thành tín hiệu quang để kích các linh kiện bán dẫn từ đó giúp chuyển mạch tải mà không cần cơ cấu chuyển động vật lý như rơ le truyền thống.

2.3. Tổng hợp các thiết bị phần cứng của hệ thống

Bảng 2.1: Bảng tổng hợp các thiết bị phần cứng của hệ thống.

STT	Tên sản phẩm	SL	Mã sản phẩm/kích thước	Hãng sản xuất
1	Vỏ tủ điện	1	300x400x210mm	Việt Nam
2	PLC S7 1214C DC/DC/DC	1	6ES7 214-1AG-40-0XB0	Siemens
3	Module truyền thông CB 1241 RS485	1	6ES7 241-1CH30-1XB0	Siemens
4	Đồng hồ đo điện đa năng	1	MFM383A-C	Selec Controls
5	Biến dòng vuông	1	SPCT-62/22-30/5	Selec Controls
6	Bộ nguồn tổ ong	1	S8FS-C05024	Omron
7	Switch 2 vị trí ON/OFF	1	XA2ED21	Schneider
8	Đèn báo pha	1	XA2EVM4LC	Schneider
9	Rơ le trạng thái rắn	1	SSR-25 DA	Fotek
10	MCB	1	NXB-63 C10 2P	Chint

2.4. Các phần mềm sử dụng trong đề tài

2.4.1. Node.js

Node.js là một môi trường chạy JavaScript trên Server. Nó được xây dựng trên V8 JavaScript Engine của Google Chrome và giúp các lập trình viên sử dụng JavaScript để xây dựng các ứng dụng phía Server, thay vì chỉ chạy trong trình duyệt như hệ thống.



Hình 2.11: Phần mềm Node.js.

2.4.2. Visual Studio Code

(VS Code) là một trình soạn thảo mã nguồn mở miễn phí, đa nền tảng, nhẹ nhưng rất mạnh mẽ và được phát triển từ Microsoft. Nó hỗ trợ nhiều ngôn ngữ lập trình đặc biệt như là JavaScript và được sử dụng rộng rãi. Bên cạnh đó còn cho phép người dùng commit, push, ngay trong trình soạn thảo mà không cần mở Terminal.



Hình 2.12: Phần mềm Visual Studio Code.

Ở trong đề tài lần này sẽ sử dụng 2 phần mềm **Node.js** và **Visual Studio Code** để hỗ trợ cho quá trình viết code JavaScript cho MQTT Server.

2.4.3. TIA Portal V17

Trong đề tài lần này sẽ sử dụng phần mềm TIA Portal để lập trình việc kết nối truyền thông Modbus RTU giữa PLC và đồng hồ đo điện Selec MFM383A cũng như lập trình việc tính toán công suất tiêu thụ cho từng khung giờ theo thời gian thực.



Hình 2.13: Phần mềm TIA Portal V17

2.4.4. MySQL

MySQL là một hệ thống quản trị cơ sở dữ liệu quan hệ mã nguồn mở, được sử dụng để lưu trữ, quản lý và truy cập dữ liệu. MySQL hoạt động dựa trên cách thức vận hành chính đó là tạo ra các bảng để lưu trữ dữ liệu.



Hình 2.14: Phần mềm MySQL.

Trong đề tài lần này sẽ sử dụng phần mềm MySQL để tạo các bảng lưu trữ các dữ liệu đọc được từ PLC nhằm mục đích tăng tính bảo mật của dữ liệu cũng như tránh các trường hợp mất dữ liệu khi hệ thống gặp sự cố ngoài ý muốn.

2.5. Các ngôn ngữ lập trình được sử dụng trong đề tài

2.5.1. JavaScript

JavaScript hoạt động thông qua nền tảng **Node.js**. Chức năng chính của ngôn ngữ lập trình JavaScript trong đề tài lần này là được dùng để kết nối MQTT Server với nền tảng IoT ThingsBoard và xử lý các yêu cầu đặc thù về công việc tính toán công suất tiêu thụ trong từng khung giờ theo cũng như tính toán tiền điện tương ứng với từng khung giờ đó trong các Node của ThingsBoard.

2.5.2. TBEL (ThingsBoard Expression Language)

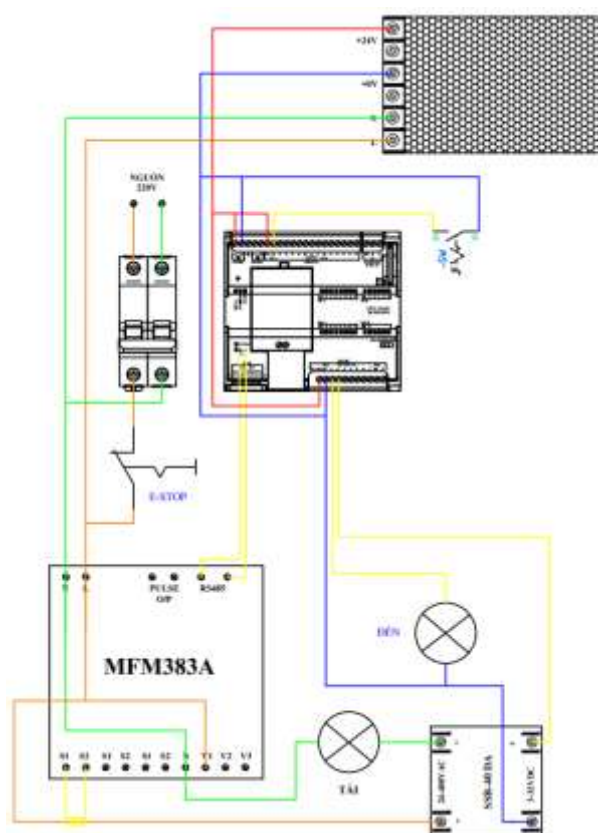
Dựa vào tên của ngôn ngữ lập trình này thì ta có thể thấy đây chính là ngôn ngữ lập trình được phát triển riêng của nền tảng IoT ThingsBoard. Vì đây là ngôn ngữ lập trình dành riêng cho nền tảng IoT ThingsBoard nên nó được thiết kế đặc thù cho các tác

vụ trong ThingsBoard, đặc biệt trong các Rule Chain. Điều này giúp người dùng dễ dàng thao tác việc xử lý dữ liệu.

Ứng dụng chính của TBEL trong ThingsBoard:

- Xử lý và lọc dữ liệu từ Telemetry và Metadata: TBEL có chức năng đọc và xử lý cũng như lọc dữ liệu được gửi lên từ thiết bị (Devices).
- Tạo và xử lý các cảnh báo (Alarms): TBEL có chức năng định nghĩa về các quy tắc để tạo ra các cảnh báo khi các giá trị giám sát (Monitor Value) vượt ngưỡng cho trước.

2.6. Sơ đồ bản vẽ đi dây của hệ thống



Hình 2.15: Sơ đồ bản vẽ đi dây của phần cứng.

2.7. Kết luận chương 2

Trong chương 2 này đã xây dựng được kiến trúc IoT của hệ thống. Bên cạnh đó cũng trình bày được các thiết bị phần cứng, sơ đồ đấu nối các thiết bị phần cứng, các thiết bị phần mềm cũng như ngôn ngữ lập trình tương ứng của từng phần mềm. Ở chương tiếp theo sẽ trình bày về các cơ sở kiến thức liên quan đến đề tài bao gồm các định nghĩa về các thành phần trong nền tảng IoT ThingsBoard, các chuẩn truyền thông công nghiệp và cách kết nối MQTT Server cho hệ thống.

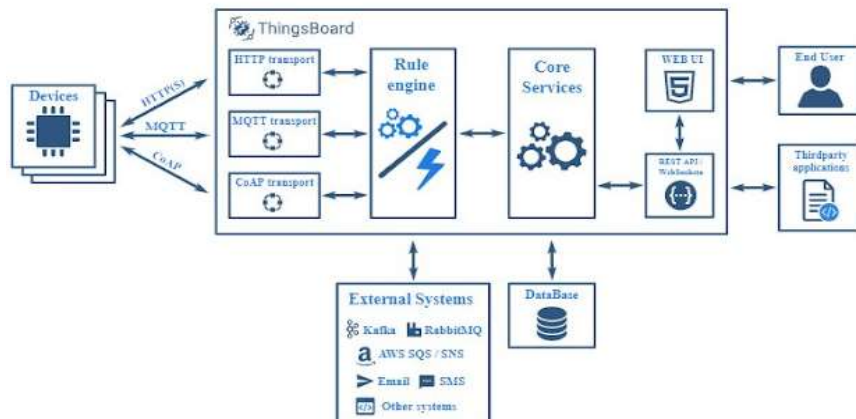
CHƯƠNG 3: CÁC CƠ SỞ KIẾN THỨC LIÊN QUAN VÀ KẾT NỐI MQTT SERVER

3.1. Giới thiệu về ThingsBoard và các thành phần chính

3.1.1. Giới thiệu về ThingsBoard

ThingsBoard là một nền tảng IoT mã nguồn mở. Nó cung cấp một giải pháp toàn diện cho việc quản lý và giám sát các thiết bị IoT và dữ liệu. ThingsBoard có khả năng kết nối, thu thập dữ liệu, xử lý dữ liệu và hiển thị thông tin từ các thiết bị IoT khác nhau thông qua một giao diện trực quan và dễ sử dụng.

ThingsBoard còn hỗ trợ các giao thức IoT tiêu chuẩn công nghiệp như MQTT, HTTP và CoAP. Kết hợp khả năng mở rộng, khả năng chịu tải và hiệu suất cao nhằm thu thập dữ liệu từ các thiết bị IoT để xử lý và giám sát. Ngoài ra còn cung cấp Gateway Server có thể giúp giao tiếp với các thiết bị IoT một cách dễ dàng.



Hình 3.1: Kiến trúc cơ bản của nền tảng IoT ThingsBoard.

3.1.2. Entities và Relations

3.1.2.1. Entities (Thực thể)

Các loại thực thể phổ biến trong ThingsBoard bao gồm:

- Device (Thiết bị): Đại diện cho các thiết bị IoT, như cảm biến, đồng hồ đo điện, hoặc máy móc công nghiệp (ví dụ: cảm biến, PLC, ESP32...).
- Asset (Tài sản): Đại diện cho các tài sản vật lý như tòa nhà, khu vực, hoặc bộ phận trong nhà máy.

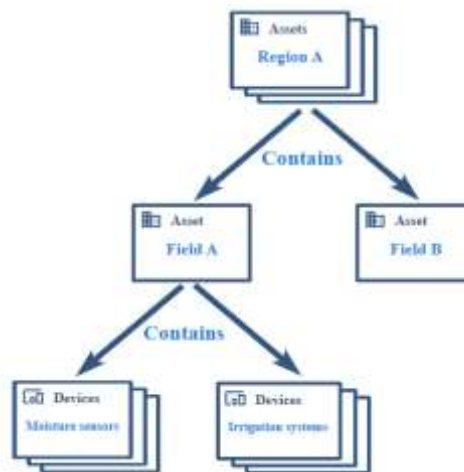
- Dashboard (Bảng điều khiển): Đại diện cho các giao diện trực quan hóa dữ liệu từ Entities được cấu hình để theo dõi thiết bị hoặc tài sản.
- Rule Chain/Rule Node (Chuỗi quy tắc): Đại diện cho các luồng xử lý dữ liệu (xử lý, lọc, phân tích) trong hệ thống dựa trên các nút quy tắc Rule Node.

Mỗi thực thể là một đơn vị dữ liệu có định danh riêng trong ThingsBoard như thuộc tính (attributes), giá trị đo lường từ xa (telemetry), trạng thái (state), và các thông tin định danh khác.

3.1.2.2. Relations (Quan hệ)

Relations là cách ThingsBoard thiết lập mối liên kết giữa các thực thể. Những mối quan hệ này giúp mô hình hóa các kết nối trong hệ thống IoT, chẳng hạn như:

- Một thiết bị (Device) thuộc về một tài sản (Asset).
- Một khách hàng (Customer) sở hữu một hoặc nhiều thiết bị (Device).
- Một tài sản (Asset) nằm trong phạm vi quản lý của một tổ chức (Tenant).



Hình 3.2: Mối quan hệ cơ bản giữa các thực thể.

3.1.3. Device và Device Profile

3.1.3.1. Device (Thiết bị)

Device trong ThingsBoard là đại diện cho một phần cứng thực tế, chẳng hạn như cảm biến, PLC, thiết bị IoT. Device là thành phần trung tâm của ThingsBoard, nơi nó gửi dữ liệu Telemetry (như nhiệt độ, độ ẩm, điện năng tiêu thụ,...).

Các thiết bị có thể được kết nối với ThingsBoard qua nhiều giao thức truyền thông khác nhau như MQTT, HTTP, hoặc CoAP. Mỗi thiết bị được định danh duy nhất trong hệ thống bằng một mã Device Token, đảm bảo tính bảo mật trong quá trình truyền dữ liệu.

Các tính năng của Device trong ThingsBoard cho phép người dùng có thể:

- Quản lý thiết bị: Thêm, xóa, và theo dõi trạng thái hoạt động của các thiết bị.
- Tùy chỉnh dữ liệu: Thu thập dữ liệu theo thời gian thực từ thiết bị và hiển thị trên các dashboard.

3.1.3.2. Device Profile

Device Profile là một khuôn mẫu (template) chứa các cấu hình và cài đặt được áp dụng cho một hoặc nhiều Device. Nó cho phép quản lý các thiết bị một cách hiệu quả bằng cách thiết lập các tham số chung, giúp giảm thiểu công sức và thời gian cấu hình từng thiết bị riêng lẻ.

Các thành phần chính của Device Profile gồm:

- Transport Configuration (Cấu hình giao tiếp): Quy định giao thức truyền thông (MQTT, CoAP, HTTP).
- Telemetry Settings (Cài đặt dữ liệu telemetry): Định nghĩa các loại dữ liệu mà thiết bị gửi lên ThingsBoard.
- Alarm Rules (Quy tắc cảnh báo): Cài đặt các quy tắc kích hoạt cảnh báo dựa trên dữ liệu ví dụ như khi điện áp hoặc dòng điện vượt ngưỡng cho phép.

3.1.4. Asset và Asset Profile

3.1.4.1. Asset (Tài sản)

Trong ThingsBoard thì Asset đại diện cho các đối tượng logic hoặc vật lý. Mặc dù không phải là thiết bị nhưng có vai trò quan trọng trong hệ thống IoT. Asset có thể là một nhà máy, tòa nhà, khu vực sản xuất, hoặc bất kỳ một thực thể nào liên quan đến việc quản lý hệ thống IoT. Asset giúp nhóm các thiết bị (Device) vào một cấu trúc quản lý logic để dễ dàng quản lý và vận hành.

Ví dụ về Asset: Một nhà máy thông minh hay trạm biến áp có thể là một Asset, trong đó bao gồm các thiết bị như cảm biến nhiệt độ, cảm biến độ ẩm,...

3.1.4.2. Asset Profile (Hồ sơ tài sản)

Asset Profile là một khuôn mẫu (template) được sử dụng để định nghĩa các thuộc tính và cấu hình chung cho các loại Asset tương tự. Thay vì thiết lập từng Asset riêng lẻ, người dùng có thể tạo một Asset Profile với các thông số chung, sau đó áp dụng cho nhiều Asset.

3.1.5. Alarm

Trong ThingsBoard, Alarm (Cảnh báo) là tính năng quan trọng dùng để theo dõi và xử lý các sự kiện bất thường trong hệ thống IoT.

☐	Created time ↓	Originator	Type	Severity	Assignee	Status	Details
☐	2025-05-08 11:03:09	S7 1200	Cảnh báo V1N_CLK2 mức nguy hiểm	Critical	Unassigned	Active Unacknowledged	***
☐	2025-05-02 19:39:26	S7 1200	Cảnh báo ITN mức nguy hiểm	Critical	Unassigned	Active Unacknowledged	***
☐	2025-05-02 19:37:35	S7 1200	Cảnh báo ITN mức cao	Major	Unassigned	Active Unacknowledged	***

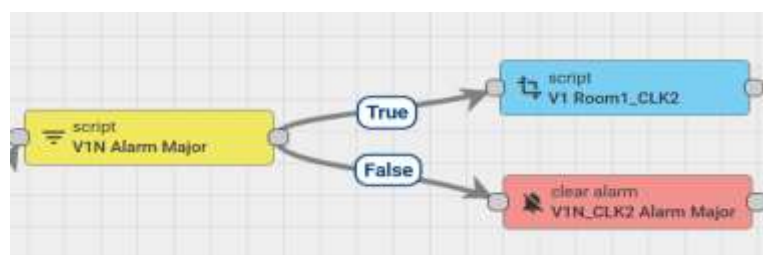
Hình 3.3: Các Alarm trong ThingsBoard khi có sự cố xảy ra.

- **Các thành phần cơ bản cấu thành nên một Alarm bao gồm:**
 - **Originator (Nguồn gốc):** Là thiết bị gây ra cảnh báo, ví dụ như thiết bị khi gửi dữ liệu dòng điện hay điện áp vượt ngưỡng cho phép.
 - **Type (Loại cảnh báo):** Phân loại cảnh báo giúp người dùng có thể xác định nguyên nhân gây ra cảnh báo là từ đâu.
 - **Severity (Mức độ nghiêm trọng):** Cảnh báo có thể có mức độ cao đến nguy hiểm, giúp người dùng đánh giá mức độ ưu tiên cần xử lý.
 - **Status (Trạng thái):** Cảnh báo có thể có các trạng thái như Active (Đang kích hoạt) hoặc Cleared (Đã giải quyết).

3.1.6. Rule Node và Rule Chain

3.1.6.1. Rule Node

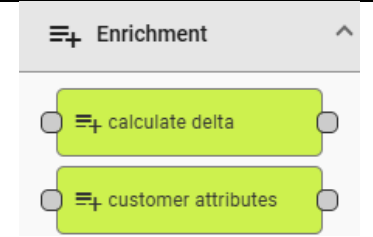
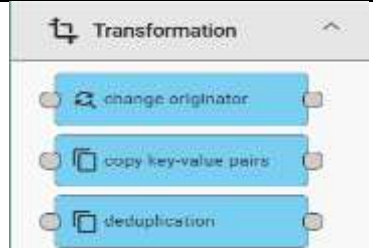
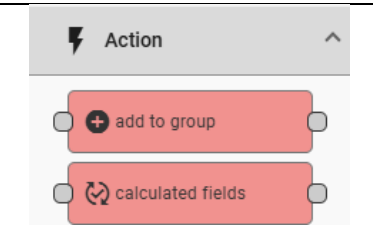
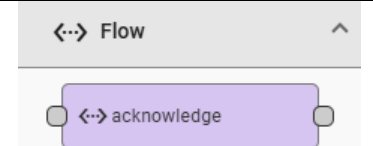
Trong nền tảng ThingsBoard thì Rule Node là thành phần cơ bản của Rule Engine. Các Rule Node có thể kết nối với nhau để tạo thành các quy trình xử lý dữ liệu.



Hình 3.4: Ví dụ về 3 Node được kết nối với nhau qua mối quan hệ “True” và “False”.

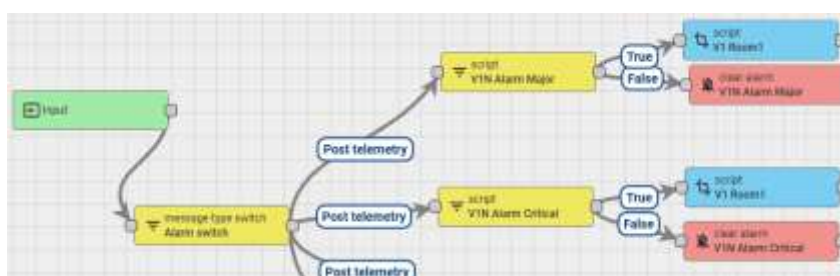
Các Rule Node trong ThingsBoard được chia thành các nhóm khác nhau, mỗi nhóm có một chức năng cụ thể trong việc xử lý dữ liệu. Các nhóm đó bao gồm như:

Bảng 3.1: Bảng tổng hợp các nhóm Rule Node và chức năng của chúng.

+ Filter Node (Node lọc): Có chức năng lọc dữ liệu đầu vào trước khi vào bước xử lý dữ liệu	
+ Enrichment Node (Node làm giàu dữ liệu): Có chức năng làm giàu dữ liệu bằng việc bổ sung thêm các thông tin từ các nguồn khác ví dụ như thông tin từ các thuộc tính của thiết bị hay tài sản.	
+ Transformation Node (Node chuyển đổi dữ liệu): Có chức năng chuyển đổi, tính toán hoặc thay đổi cấu trúc của dữ liệu đầu vào trước khi đưa vào bước lưu trữ hoặc thực hiện các bước tiếp theo.	
+ Action Node (Node hành động): Có chức năng dùng để tạo cảnh báo hoặc gửi thông báo qua E-mail người dùng khi hệ thống có sự cố.	
+ Flow Nodes (Node điều hướng): Có chức năng điều hướng dữ liệu trong Rule Chain.	

3.1.6.2. Rule Chain

Rule Chain là tập hợp của các Rule Node liên kết với nhau theo chuỗi, có chức năng xử lý dữ liệu theo một quy trình liên tục từ điểm đầu đến điểm cuối. Các Rule Chain giúp quản lý các quy tắc một cách dễ quản lý hơn để từ đó việc xử lý dữ liệu phức tạp được đơn giản hóa.



Hình 3.5: Hình ảnh về một Rule Chain.

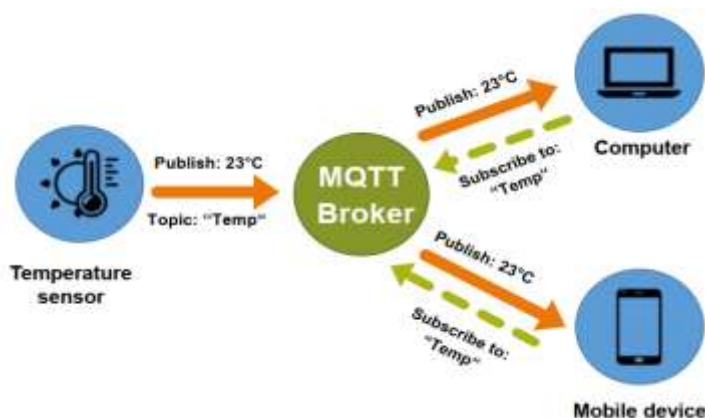
3.2. Các giao thức truyền thông sử dụng trong đề tài

3.2.1. Giao thức MQTT

3.2.1.1. Giới thiệu về MQTT

MQTT (Message Queueing Telemetry Transport) là một giao thức đơn giản ở cấp độ TCP/IP và cũng là giao thức nền. Để truyền tin nhắn giữa các thiết bị thì nó được hoạt động theo cơ chế Publish – Subscribe (xuất bản – đăng ký).

Khi một thiết bị gửi tin nhắn (Publish) đến một chủ đề nào đó thì Broker sẽ phân phối tin nhắn đó đến tất cả các thiết bị đã đăng ký (Subscribe) chủ đề đó. Điều này giúp giảm sự phụ thuộc giữa các thiết bị khi mà thiết bị được sử dụng ở những nơi mà giá cả mạng viễn thông đắt đỏ hoặc băng thông thấp hay thiếu độ tin cậy.



Hình 3.6: Một ví dụ về mối quan hệ giữa MQTT Borker và MQTT Client.

3.2.1.2. Ưu điểm và nhược điểm của MQTT

Ưu điểm:

- Có cơ chế Publish/Subscribe cho phép phân tán dữ liệu hiệu quả
- Đây cũng là giao thức sử dụng băng thông thấp trong môi trường có độ trễ cao nên dựa vào nhược điểm này thì nó có thể là một giao thức lý tưởng cho các ứng dụng M2M (Machine to Machine).

Nhược điểm:

- So với CoAP, MQTT có tốc độ truyền dữ liệu chậm hơn. Trong khi CoAP sử dụng hệ thống tài nguyên ổn định – tĩnh, MQTT lại hoạt động dựa trên cơ chế đăng ký động (subscribe).

3.2.2. Giao thức S7 Communication

S7 Communication là một giao thức truyền thông độc quyền do Siemens phát triển, được thiết kế và tối ưu hóa đặc biệt cho việc trao đổi dữ liệu và giao tiếp một cách hiệu quả và đơn giản giữa các thiết bị tự động hóa thuộc dòng SIMATIC S7 của hãng Siemens.

Khả năng giao tiếp S7 được tích hợp sẵn trên các CPU S7-1200 và sử dụng cổng Ethernet tiêu chuẩn để truyền tải dữ liệu. Cổng Ethernet tích hợp là cổng kết nối vật lý phổ biến nhất và được tích hợp trên tất cả các dòng CPU S7-1200. Loại cổng sử dụng là loại RJ45.

Các đặc điểm nổi bật của Giao thức S7 Communication:

- Là một giao thức độc quyền của Siemens phát triển nên được tích hợp sẵn trên các dòng PLC, biến tần,.. của Siemens.
- Đa nền tảng giao tiếp: Hỗ trợ nhiều giao thức kết nối vật lý như Ethernet (S7-Communication via Profinet), Profibus,...

3.2.3. Giao thức SMTP

3.2.3.1. Giới thiệu giao thức SMTP

SMTP (Simple Mail Transfer Protocol) là giao thức chuẩn để gửi email trên Internet. Đây là nền tảng để giao tiếp giữa các hệ thống mail server, giúp chuyển tiếp và gửi email từ người gửi đến người nhận thông qua các mạng. [4]

SMTP hoạt động dựa trên mô hình Client-Server:

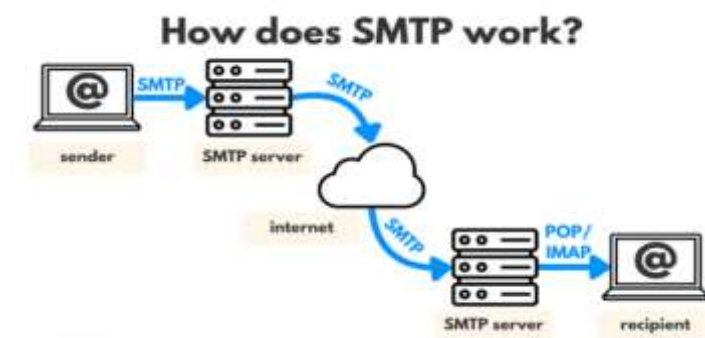
- SMTP Client: Thiết bị hoặc ứng dụng gửi email (ví dụ: Outlook, Gmail, hoặc hệ thống IoT).
- SMTP Server: Máy chủ chịu trách nhiệm xử lý, gửi và phân phối email đến server đích hoặc hộp thư người nhận.

3.2.3.2. Cách thức hoạt động của SMTP Server

Quá trình gửi E-mail qua SMTP Server sẽ được chia thành 3 bước chính:

- B1: Kết nối SMTP Client (ứng dụng gửi email) thiết lập để kết nối với SMTP Server thông qua cổng SMTP Gmail (TLS) là 587 hoặc cổng SMTP Gmail (SSL) là 465.
- B2: Trao đổi thông tin: Server nhận thông tin về người gửi, người nhận và nội dung e-mail.

- B3: Chuyển tiếp: Nếu địa chỉ e-mail đích thuộc cùng Server, e-mail sẽ được lưu trữ trực tiếp, nếu không thì e-mail sẽ được chuyển qua các Server trung gian.



Hình 3.7: Cách thức SMTP hoạt động.

3.2.4. Giao thức Modbus RTU

3.2.4.1. Định nghĩa giao thức Modbus RTU

Modbus RTU là một dạng phương thức truyền thông nối tiếp, các thiết bị có cùng đầu ra là Modbus RTU thì đều có thể mắc nối tiếp lại với nhau tạo thành một chuỗi các thiết bị kết nối với nhau. Modbus RTU hoạt động theo giao thức mở sử dụng đường truyền vật lý bằng RS232 hoặc RS485 và giao tiếp giữa chúng tuân thủ theo nguyên tắc Master và Slave (hay còn gọi là mối quan hệ Chủ – Tớ).



Hình 3.8: Thiết bị Master và thiết bị Slave.

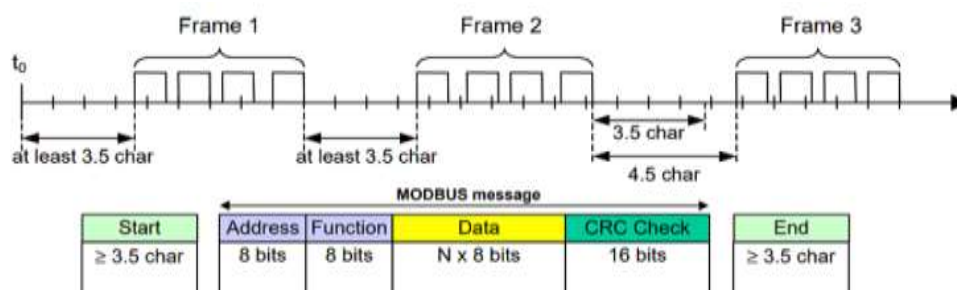
Trong đề tài này nhóm sinh viên sử dụng PLC S7-1200 để đọc dữ liệu từ đồng hồ đo điện Selec nên S7-1200 sẽ đóng vai trò làm Master và đồng hồ Selec sẽ đóng vai trò là Slave.

3.2.4.2. Cách thức hoạt động của Modbus RTU

Modbus RTU hoạt động bằng cách truyền tín hiệu điện dưới dạng mã nhị phân giữa Master và Slave với nhau. Đoạn mã này sẽ được truyền qua hai phương thức truyền

đó là Modbus RTU RS485 và Modbus RTU RS232 (RS422). Một hệ thống thì có thể có tối đa 247 thiết bị Slave và mỗi thiết bị có một địa chỉ duy nhất (từ 1-247).

Một cấu trúc bản tin của Modbus RTU bao gồm: 1 byte địa chỉ, 1 byte mã chức năng, n byte trường dữ liệu, 2 byte trường CRC như hình sau:



Hình 3.9: Cấu trúc bản tin của Modbus RTU.

Chức năng và vai trò của mỗi byte như sau: Byte địa chỉ: Xác định địa chỉ nào được nhận dữ liệu từ Master (trong trường hợp là Slave) và ngược lại là xác định nhận được từ địa chỉ của đối tượng Slave nào ở phía Master, địa chỉ này được quy định thông thường từ 0 – 254.

3.2.4.3. Ưu điểm và nhược điểm của Modbus RTU

Ưu điểm:

- Có khả năng truyền dữ liệu đi xa với khoảng cách lên đến 1200m (trong thực tế thì tầm 700m ~ 900m) với tốc độ nhanh.
- Các Slave có chuẩn truyền thông Modbus RTU có thể kết nối nối tiếp với nhau từ đó có thể giảm thiểu số lượng dây kết nối và chi phí cũng như không gian lắp đặt.
- Các thiết bị của nhiều hãng khác nhau nhưng có cùng chung chuẩn truyền thông Modbus RTU đều có thể kết nối và tích hợp với nhau.

Nhược điểm:

- Thiết bị Master phải luôn thăm dò từng thiết bị Slave và tìm kiếm khi mà có các thay đổi về mặt dữ liệu nên điều này sẽ gây tiêu hao băng thông.

3.3. Tạo dữ liệu mô phỏng từ dữ liệu thật và kết nối Modbus RTU

3.3.1. Ý tưởng thực hiện

Đầu tiên là phải đọc được các dữ liệu thông số từ đồng hồ Selec MFM383A về PLC thông qua Module truyền thông CB 1241 nhờ giao thức truyền thông Modbus RTU.

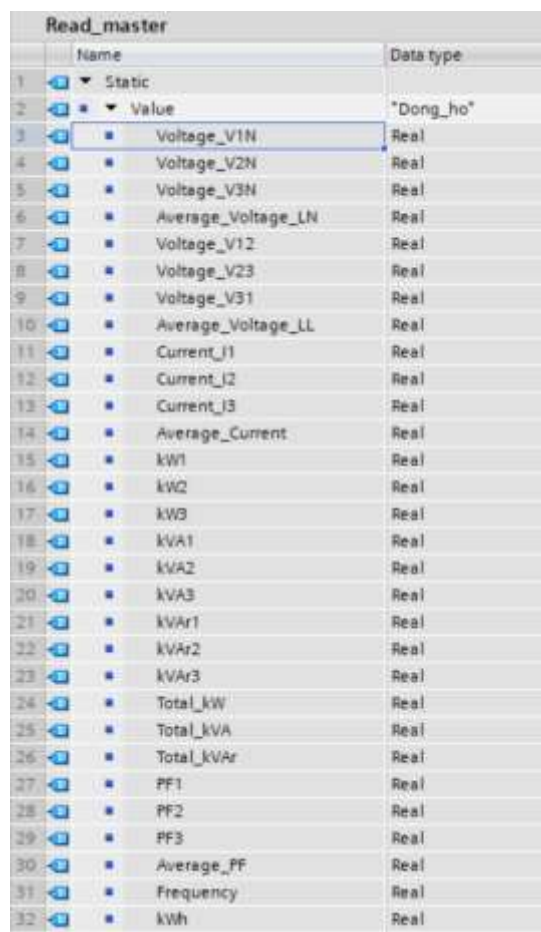
Sau đó dựa vào các giá trị dữ liệu thật tạo khối mô phỏng dữ liệu để nhân bản thêm một đồng hồ Selec ảo với từng giá trị (về điện áp, dòng điện, tần số, công suất tác dụng...) có một khoảng sai số cho phép với giá trị ban đầu của đồng hồ Selec thật.

Do việc có sự tương đồng của 2 đồng hồ Selec MFM383A này (đồng hồ thật và giá trị ảo) nên việc xây dựng 1 Data Type chung để tạo ra các biến chứa trong Data Block một cách nhanh chóng và đồng bộ là điều cần thiết.

Tiếp theo thực hiện việc gửi các giá trị từ PLC lên MQTT Server qua giao thức S7 Communication. Để thực hiện việc gửi giá trị này thì các biến phải được lưu trữ trong Data Block thay vì dạng Tag để dễ dàng truyền thông. Từ các yêu cầu nêu trên thì nhóm sinh viên tiến hành lập trình PLC.

3.3.2. Tạo Data Type và tạo các khối Data Block

Dựa vào Datasheet của hãng sản xuất đồng hồ Selec thì tiếp tục tiến hành tạo một Data Block và một Data Type với kiểu dữ liệu là Real và đồng thời chứa các thông số của đồng hồ theo hãng sản xuất.



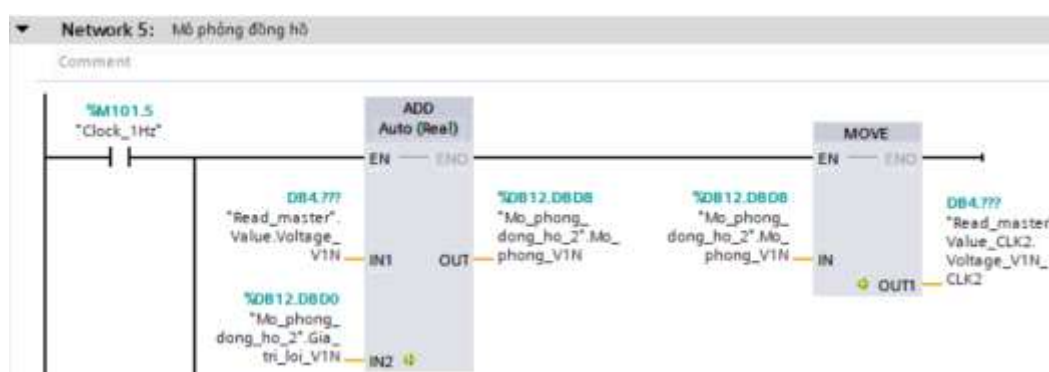
Read_master		
	Name	Data type
1	Static	
2	Value	"Dong_ho"
3	Voltage_V1N	Real
4	Voltage_V2N	Real
5	Voltage_V3N	Real
6	Average_Voltage_LN	Real
7	Voltage_V12	Real
8	Voltage_V23	Real
9	Voltage_V31	Real
10	Average_Voltage_LL	Real
11	Current_I1	Real
12	Current_I2	Real
13	Current_I3	Real
14	Average_Current	Real
15	kW1	Real
16	kW2	Real
17	kW3	Real
18	kVA1	Real
19	kVA2	Real
20	kVA3	Real
21	kVAr1	Real
22	kVAr2	Real
23	kVAr3	Real
24	Total_kW	Real
25	Total_kVA	Real
26	Total_kVAr	Real
27	PF1	Real
28	PF2	Real
29	PF3	Real
30	Average_PF	Real
31	Frequency	Real
32	kWh	Real

Hình 3.10: Data Type lưu các thông số đọc được từ đồng hồ Selec MFM383A.

Tiếp theo tiến hành tạo các biến trong Data Block từ Data Type đã tạo. Vì hạn chế về mặt phần cứng trong đề tài lần này nên cần tạo thêm một đồng hồ Selec ảo từ đồng hồ Selec thật với các biến có tên tương tự như các biến của đồng hồ thật.

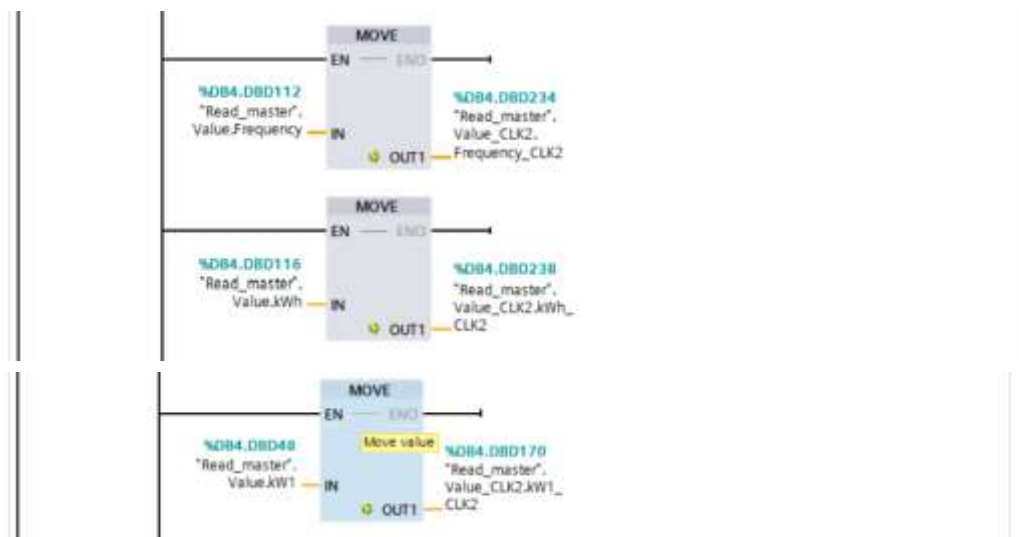
3.3.3. Tạo dữ liệu mô phỏng từ dữ liệu của đồng hồ thật

Để mô phỏng đồng hồ thứ 2 thì tiến hành lấy và cập nhật dữ liệu liên tục từ đồng hồ thật và cộng thêm vào các sai số để có thể tạo cảnh báo bên ThingsBoard. Giả dụ như dữ liệu Voltage_V1N của đồng hồ 2 sẽ lấy từ giá trị gốc Voltage_V1N đồng hồ thật được cộng thêm sai số.



Hình 3.11: Đoạn chương trình tạo dữ liệu mô phỏng về điện áp.

Các thông số khác như tần số, công suất tác dụng, công suất tiêu thụ tích lũy cũng sẽ được lấy trực tiếp từ đồng hồ thật.



Hình 3.12: Đoạn chương trình tạo dữ liệu mô phỏng các thông số cần giám sát.

Các biến ảo được tạo từ đồng hồ thật sẽ có đuôi dạng “_CLK2” để nhận biết đây là biến được mô phỏng từ dữ liệu thật.

Read_master		
	Name	Data type
34	Value_CLK2	"Đồng hồ_2"
35	Voltage_V1N_CLK2	Real
36	Voltage_V2N_CLK2	Real
37	Voltage_V3N_CLK2	Real
38	Average_Voltage_LN_CLK2	Real
39	Voltage_V12_CLK2	Real
40	Voltage_V23_CLK2	Real
41	Voltage_V31_CLK2	Real
42	Average_Voltage_LL_CLK2	Real
43	Current_I1_CLK2	Real
44	Current_I2_CLK2	Real
45	Current_I3_CLK2	Real
46	Average_Current_CLK2	Real
47	kW1_CLK2	Real
48	kW2_CLK2	Real
49	kW3_CLK2	Real
50	kVA1_CLK2	Real
51	kVA2_CLK2	Real
52	kVA3_CLK2	Real
53	kVar1_CLK2	Real
54	kVar2_CLK2	Real
55	kVar3_CLK2	Real
56	Total_kW_CLK2	Real
57	Total_kVA_CLK2	Real
58	Total_kVar_CLK2	Real
59	PF1_CLK2	Real
60	PF2_CLK2	Real
61	PF3_CLK2	Real
62	Average_PF_CLK2	Real
63	Frequency_CLK2	Real
64	kWh_CLK2	Real

Hình 3.13: Dữ liệu trong Data Block sau khi được mô phỏng.

3.3.4. Đọc dữ liệu về bằng Modbus RTU

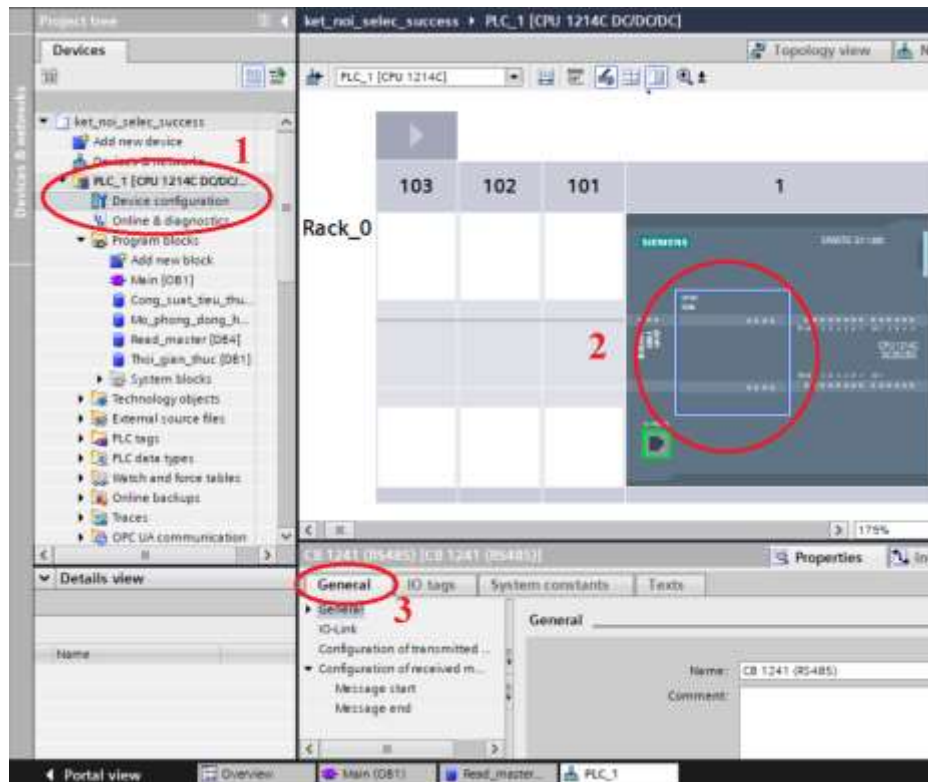
Đầu tiên phải thêm Module truyền thông CB 1241 (RS485) vào PLC để hoàn tất việc lắp đặt phần cứng trên PLC bằng cách tại mục Communication Board, tiếp theo chọn Point-to-Point sau đó kéo thả Module CB 1241 vào trong PLC như hình dưới đây.



Hình 3.14: Cách thêm CB 1241 vào PLC trong Tia Portal V17.

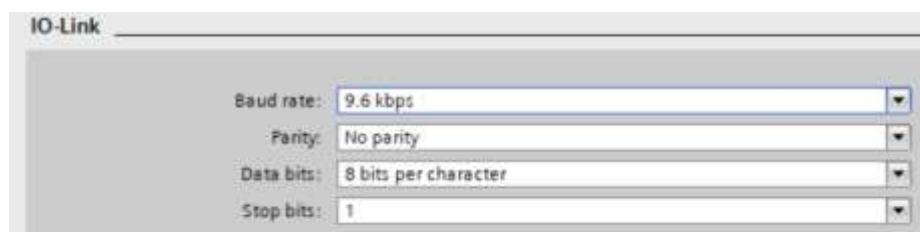
Tiếp theo là phải cấu hình các thông số cho Module truyền thông CB 1241 (RS485). Việc thực hiện được theo các bước như sau:

- B1: Vào Device Configuration.
- B2: Click chọn vào hình ảnh Module truyền thông CB 1241.
- B3: Click vào mục General, tại Tag General trong mục IO-Link, ta cần quan tâm tới các thông số về truyền thông.



Hình 3.15: Các bước cấu hình cho CB 1241 (RS485).

- B4: Tại mục IO-Link thì sẽ cấu hình các thông số về truyền thông như Baud rate, Parity, Data bits, Stop bits. Những thông số này phải đúng như trên thiết bị Slave mà nhà sản xuất đồng hồ Selec cung cấp.



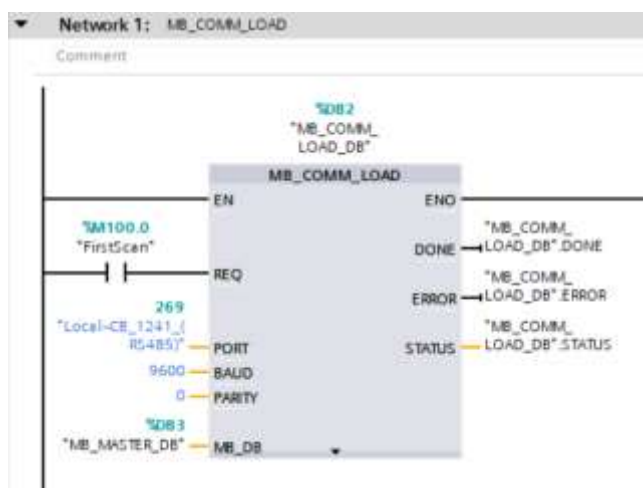
Hình 3.16: Cài đặt các thông số về truyền thông.

- B5: Tiếp theo là cài đặt thông số Wait time. Đây là thời gian chờ thiết lập lại, trong trường hợp khi việc kết nối thất bại thì trong khoảng thời gian Wait time này Module sẽ cố gắng kết nối trở lại.



Hình 3.17: Cài đặt thông số Wait time.

- B6: Cuối cùng là thiết lập hàm MB_COMM_LOAD để cấu hình cho việc kết nối Modbus RTU.



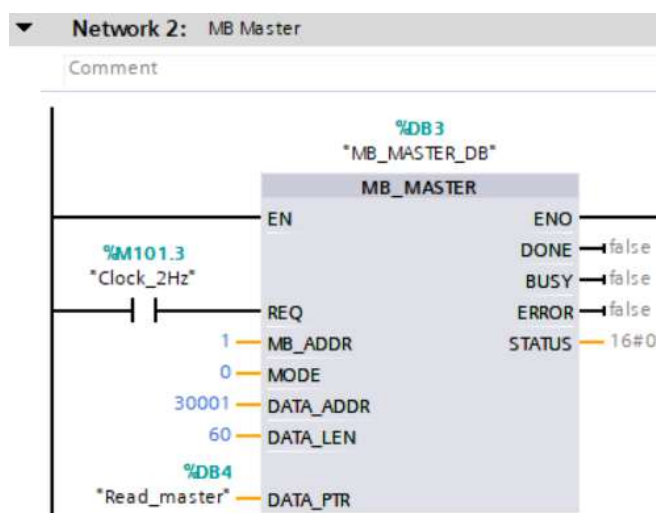
Hình 3.18: Thiết lập hàm MB_COMM_LOAD.

Sau khi thiết lập xong hàm MB_COMM_LOAD thì bắt đầu liệt kê các địa chỉ thanh ghi của từng dữ liệu mà ta cần truyền thông.

Bảng 3.2: Địa chỉ thanh ghi các thông số của đồng hồ Selec trên Datasheet.

Parameter	Địa chỉ thanh ghi	Unit	Data Type
Voltage V1N	30000	A	Float 32 (Real)
Current I1	30016	V	Float 32 (Real)
Frequency	30056	Hz	Float 32 (Real)
Active Power	30024	kW	Float 32 (Real)
Accumulated Energy	30058	kWh	Float 32 (Real)

Sau khi dựa vào bảng địa chỉ thanh ghi của các thông số đồng hồ trên Datasheet thì tiến hành lập trình khối MB_Master để gửi dữ liệu từ đồng hồ lên PLC. Theo chuẩn Modbus RTU của các dòng PLC S7-1200 của Siemens thì các Holding Register có địa chỉ thanh ghi bắt đầu từ 30001 nhưng địa chỉ thanh ghi bắt đầu trong Datasheet của đồng hồ lại bắt đầu từ 30000 nên giải pháp ở đây không thể khởi tạo khối MB_MASTER cho từng dữ liệu mà thay vào đó ta thực hiện việc truyền thông tất cả các dữ liệu trong Datasheet của nhà sản xuất với DATA_LEN là 60. Sau đó ta mới tiến hành lọc các dữ liệu cần thiết trong yêu cầu để gửi lên MQTT Server.

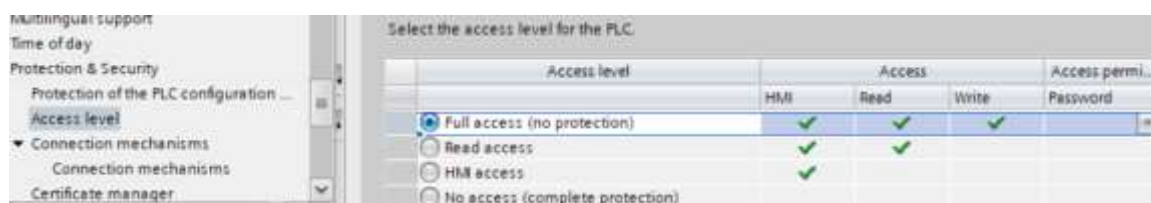


Hình 3.19: Dùng khối MB_MASTER để đọc các dữ liệu từ đồng hồ.

3.4. Kết nối MQTT Server với PLC S7-1200

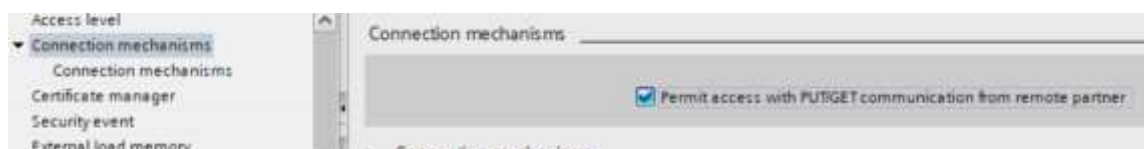
3.4.1. Cài đặt phần cứng cho PLC

PLC chỉ có thể giao tiếp với bên thứ 3 khi không đặt mật khẩu Access, do đó để kết nối PLC với MQTT Server thì trong mục Protection & Security ta phải bật chức năng “Full Access” cho PLC.



Hình 3.20: Bật chức năng Full access của PLC.

Để đọc hoặc ghi dữ liệu từ PLC lên MQTT Server thì ta cần phải bật chức năng PUT/GET trong phần cứng của PLC bằng cách tại mục Connection ta click chọn vào mục “Permit access with PUT/GET communication from remote partner”.



Hình 3.21: Bật chức năng PUT/GET của PLC.

3.4.2. Kết nối để đọc dữ liệu từ PLC lên MQTT Server

Trước tiên là tiến hành thiết lập các thông số liên quan đến PLC bao gồm địa chỉ IP của PLC, Slot (đối với dòng PLC S7-1200 và S7-1500 thì Slot = 1, đối với dòng PLC

S7-300 và S7-400 thì Slot = 2). Rack (đối với dòng PLC S7-1200 và S7-1500 thì Rack = 1, đối với dòng PLC S7-300 và S7-400 thì Rack = 2).

```

5 ///////////////////////////////////////////////////////////////////
6 // CONNECT REPORT SERVER TO PLC //
7 ///////////////////////////////////////////////////////////////////
8 // 1. PLC parameter setting
9 var PLC_IP      = '192.168.2.238';    // PLC IP address '192.168.2.238' '192.168.0.1'
10 var Slot_ID     = 1;                // S71200/1500 = 1, S7300/400 = 2
11 var scanTime    = 1;                // Scan time (second)
12 var plc_tag     = "";               // Variable save PLC tags value
13 var tagval_show = true;              // Show tag value (true = yes, false = no)
14

```

Hình 3.22: Thiết lập các thông số của PLC.

Bảng 3.3: Các thông số cần được thiết lập của PLC trong MQTT Server.

STT	Tên biến	Thông số	Chú thích
1	Var PLC_IP	192.168.2.238	Địa chỉ IP của PLC S7-1200
2	Var Slot_ID	1	Địa chỉ Slot của PLC S7-1200
3	Var ScanTime	1	Thời gian quét (giây)
4	Var PLC_tag	“ ”	Biến lưu các giá trị tag của PLC
5	Var Tagval_show	True	Hiển thị giá trị tag của PLC

Bước tiếp theo ta tiến hành tạo các biến về thông số điện năng liên quan đến đồng hồ Selec MFM383A trong khối DB để MQTT Server đọc được thông qua giao thức S7 Communication. Trong đó các biến gọi đi phải có kiểu dữ liệu là Real.

```

15 //2. PLC tag defined
16 var config_tag = {
17     Voltage_V1N:      'DB4,REAL0',
18     Current_I1:       'DB4,REAL32',
19     kW1:              'DB4,REAL48',
20     Frequency:        'DB4,REAL112',
21     kWh:              'DB4,REAL116',
22     kWh_mul:          'DB4,REAL242',
23     sql_insert_Trigger: 'DB4,X120.0',
24     Voltage_V1N_CLK2: 'DB4,REAL122',
25     Current_I1_CLK2:  'DB4,REAL154',
26     kW1_CLK2:         'DB4,REAL170',
27     Frequency_CLK2:   'DB4,REAL234',
28     kWh_CLK2:         'DB4,REAL238',

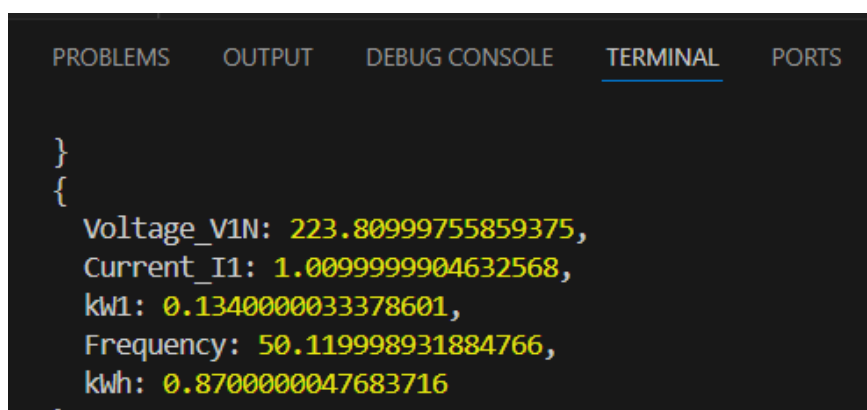
```

Hình 3.23: Các biến về thông số điện năng cần được gọi đi.

Bảng 3.4: Các biến về thông số điện năng cần được gửi đi.

STT	Tên biến	Địa chỉ & Kiểu dữ liệu	Chú thích
1	Voltage_V1N	DB4,REAL0	Điện áp pha ROOM 1
2	Current_I1	DB4,REAL32	Dòng trên pha ROOM 1
3	KW1	DB4,REAL48	Công suất tác dụng ROOM 1
4	Frequency	DB4,REAL112	Tần số ROOM 1
5	kWh	DB4,REAL116	Công suất tiêu thụ tích lũy ROOM 1
6	sql_insert_Trigger	DB4,X120.0	Biến Trigger để gửi cảnh báo
7	Voltage_V1N_CLK2	DB4,REAL122	Điện áp pha ROOM 2
8	Current_I1_CLK2	DB4,REAL154	Dòng trên pha ROOM 2
9	kW1_CLK2	DB4,REAL170	Công suất tác dụng ROOM 2
10	Frequency_CLK2	DB4,REAL234	Tần số ROOM 1
11	kWh_CLK2	DB4,REAL238	Công suất tiêu thụ tích lũy ROOM 1

Sau khi đã kết nối với PLC, xác định địa chỉ IP, ta chỉ cần nhập câu lệnh “**node PLC_MQTT_Server.js**” để chạy MQTT Server.



```
}  
{  
  Voltage_V1N: 223.80999755859375,  
  Current_I1: 1.0099999904632568,  
  kW1: 0.1340000033378601,  
  Frequency: 50.119998931884766,  
  kWh: 0.8700000047683716  
}
```

Hình 3.24: Kết quả sau khi đọc được giá trị Tag lên MQTT Server.

Với kết quả đọc được từ PLC này thì sẽ được lưu qua biến “Var PLC_Tag”. Ở bước tiếp theo ta sẽ dùng dữ liệu của biến này để gửi dữ liệu lên nền tảng IoT ThingsBoard.

3.5. Kết nối MQTT Server với ThingsBoard để đưa dữ liệu lên Platform theo chuẩn MQTT

3.5.1. Khai báo MQTT Server trên nền tảng IoT ThingsBoard

Trước hết ta sử dụng đường link của MQTT Server của ThingsBoard làm một Client MQTT Server và Username chính là Access Token. Mã Access Token này được lấy trực tiếp từ thông tin của Device trên ThingsBoard.

```
55 ////////////////////////////////////////////////////////////////////  
56 | | | | // CODE GỬI DỮ LIỆU MQTT LÊN THINGSBOARD  
57 ////////////////////////////////////////////////////////////////////  
58 // Khai báo MQTT Server trên IoT Thingsboard  
59 var mqtt = require('mqtt');  
60 var client = mqtt.connect('mqtt://eu.thingsboard.cloud'  
61 | | | | username: '819bw4ke2yq5h2znedak'
```

Hình 3.25: Đoạn code khai báo MQTT Server trên ThingsBoard.

3.5.2. Kết nối ThingsBoard Server

Khi Client bắt đầu ON thì sẽ thực hiện kết nối với ThingsBoard Server bằng câu lệnh này. Nếu đã thực hiện thành công thì kết quả sẽ trả về Terminal là “**đã kết nối MQTT thành công !**”.

```
63 // Kết nối Thingsboard Server  
64 client.on('connect', function () {  
65 | | | | console.log('đã kết nối MQTT thành công!');
```

Hình 3.26: Đoạn code kết nối ThingsBoard Server.

3.5.3. Gửi dữ liệu lên ThingsBoard

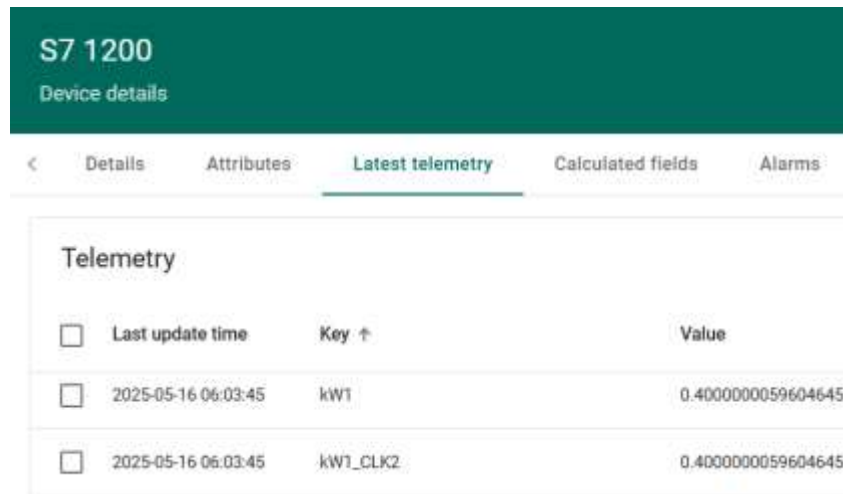
Đây là đoạn code có chức năng gửi dữ liệu lên nền tảng IoT ThingsBoard thông qua giao thức MQTT. Chức năng của các biến được trình bày như sau:

- Biến “**mqtt_topic**” dùng để gán chuỗi “**v1/devices/me/telemetry**”: Đây là Topic mặc định mà ThingsBoard sử dụng để nhận dữ liệu Telemetry (đo lường từ xa) như dữ liệu cảm biến, trạng thái hoạt động,...
- Biến “**client.publish(mqtt_topic , mqtt_data)**”: Dùng để gửi dữ liệu thông qua MQTT Client.

```
68 // Hàm gửi dữ liệu lên IoT Thingsboard  
69 function fn_mqttsend(){  
70 | | | | var mqtt_topic = 'v1/devices/me/telemetry'; // Topic nhận dữ liệu trên Thingsboard  
71 | | | | var mqtt_data = JSON.stringify(pic_tag); // Convert json to string  
72 | | | | client.publish(mqtt_topic , mqtt_data); // Send mqtt to Thingsboard  
73 | }  
74 }
```

Hình 2.27: Đoạn code gửi dữ liệu lên ThingsBoard.

Tiếp theo ta tiến hành chạy đoạn code bằng cách nhập câu lệnh “**node PLC_MQTT_Server.js**” tại Terminal sau đó truy cập vào nền tảng IoT ThingsBoard để có thể quan sát dữ liệu đã được gửi lên trên nền tảng.



The screenshot shows the 'S7 1200' device details page in ThingsBoard. The 'Latest telemetry' tab is selected, displaying a table of data points. Each row has a checkbox, a timestamp, a key, and a value.

	Last update time	Key	Value
☐	2025-05-16 06:03:45	kW1	0.4000000059604645
☐	2025-05-16 06:03:45	kW1_CLK2	0.4000000059604645

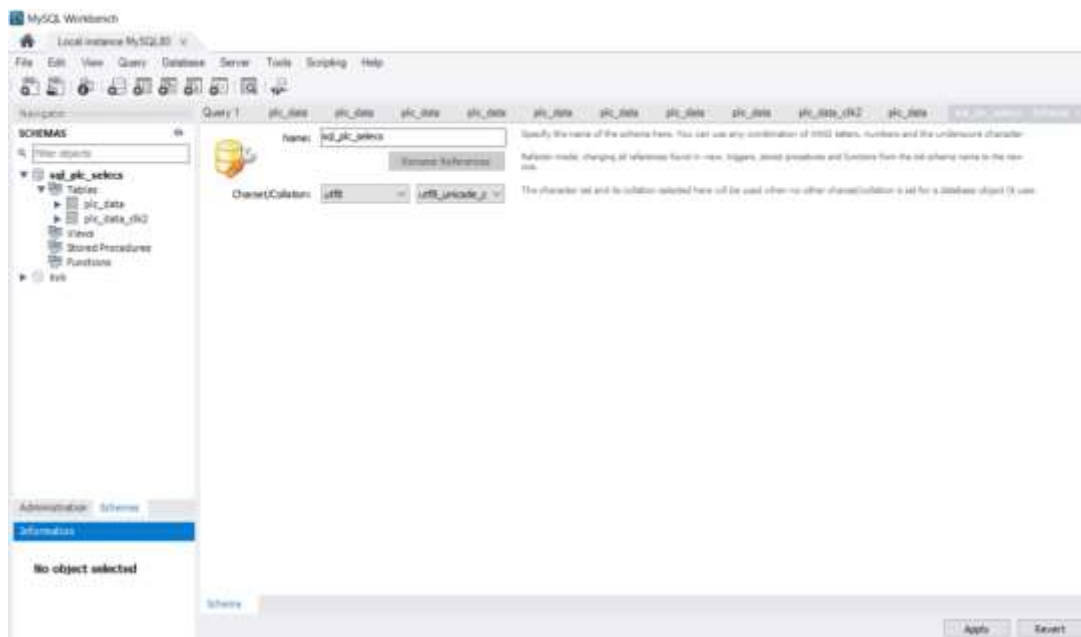
Hình 3.28: Kết quả dữ liệu về công suất tác dụng được gửi lên ThingsBoard.

3.6. Lưu trữ các dữ liệu đọc được từ PLC vào MySQL

Để dữ liệu có thể được bảo mật một cách an toàn thì việc lưu trữ dữ liệu vào MySQL là vô cùng cần thiết và quan trọng.

3.6.1. Khởi tạo cơ sở dữ liệu trong MySQL

Tạo cơ sở dữ liệu trong MySQL để lưu trữ các giá trị đọc được từ PLC.



Hình 3.29: Tạo cơ sở dữ liệu trong MySQL.

Tạo hàm gửi dữ liệu và ghi dữ liệu vào cơ sở dữ liệu, nếu thành công sẽ in “SQL - Ghi dữ liệu thành công” ra console log.

```
// Thực hiện ghi dữ liệu vào SQL
sqlcon.query(sql_write_str1, function (err, result) {
    if (err) {
        console.log(err);
    } else {
        console.log("SQL - Ghi dữ liệu thành công 1");
    }
});
```

Hình 3.33: Hàm gọi và ghi dữ liệu vào MySQL.

Dữ liệu được gửi lên liên tục sẽ gây ra việc tắc nghẽn và có độ trễ cho nên để tránh việc đó ta cần tạo hàm setInterval để gọi lại hàm gửi dữ liệu qua MySQL sau mỗi khoảng thời gian 4s.

```
var mqtt_send_timer = 4000;

setInterval(fn_sql_insert, mqtt_send_timer);
setInterval(fn_sql_insert_clk2, mqtt_send_timer);
```

Hình 3.34: Hàm gọi dữ liệu qua MySQL sau mỗi 4s.

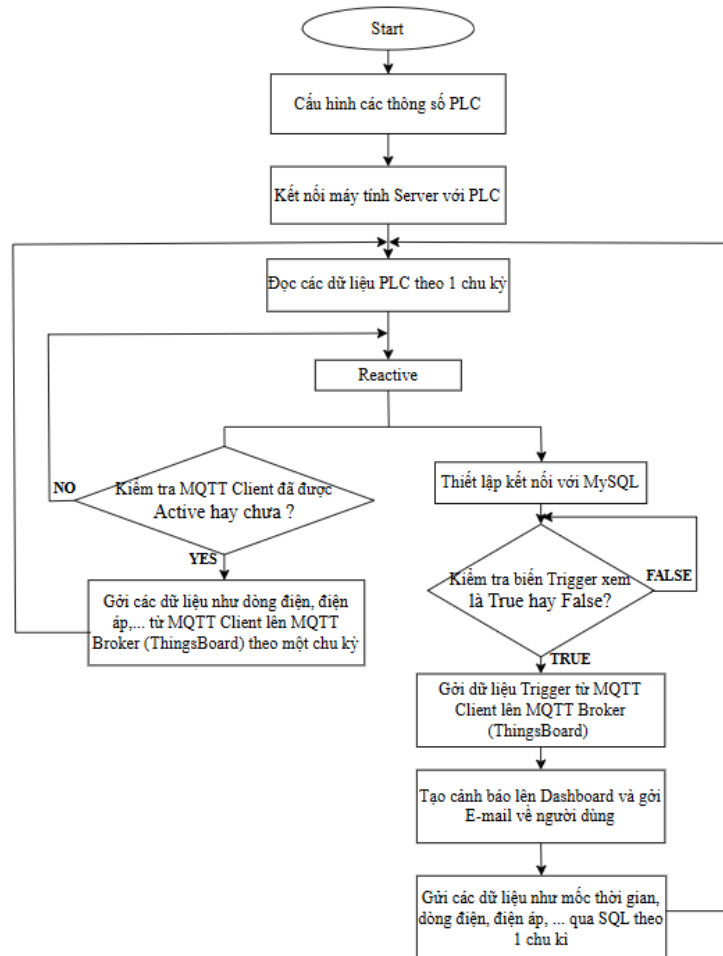
3.7. Kết luận chương 3

Trong chương 3 này đã trình bày các cơ sở kiến thức liên quan về các thành phần trong nền tảng IoT ThingsBoard và giao thức truyền thông công nghiệp được sử dụng trong đề tài. Bên cạnh đó là thực hiện việc tạo dữ liệu mô phỏng từ dữ liệu thật sau đó kết nối MQTT Server để đưa dữ liệu được thu thập từ PLC lên nền tảng IoT ThingsBoard.

CHƯƠNG 4: XÂY DỰNG CÁC CHỨC NĂNG VÀ THUẬT TOÁN XỬ LÝ DỮ LIỆU TRÊN THINGSBOARD

4.1. Tổng quan về xây dựng hệ thống thu thập dữ liệu trên ThingsBoard

Đầu tiên là về lưu đồ thuật toán chung của đề tài.

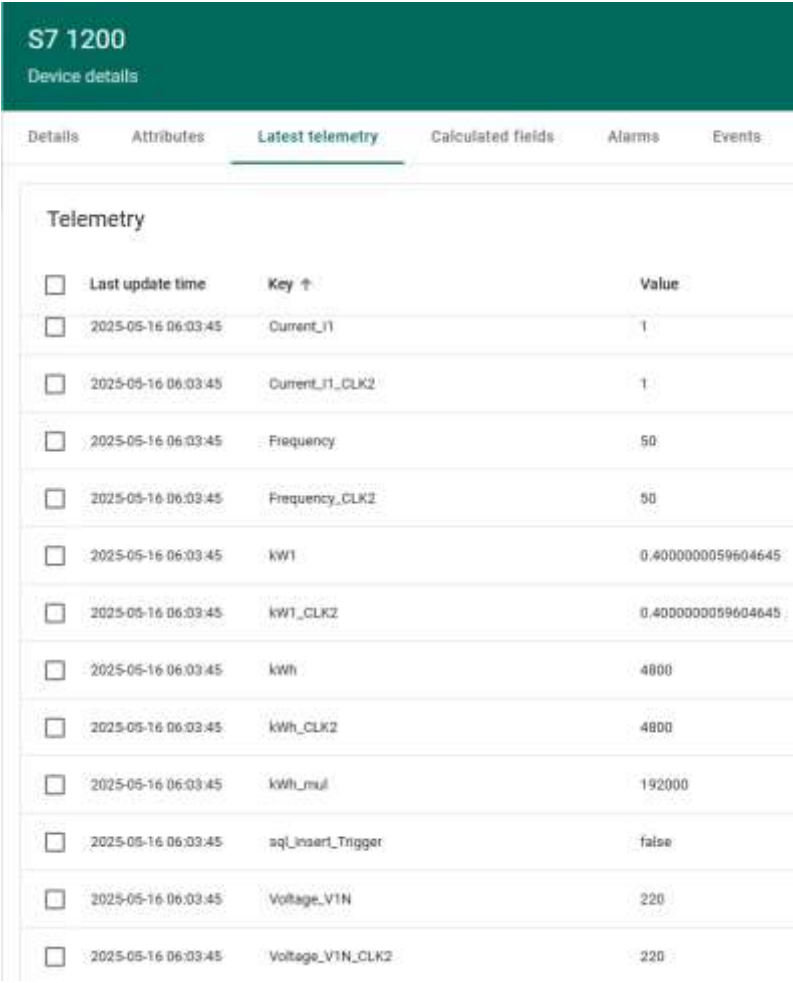


Hình 4.1: Lưu đồ thuật toán chung của đề tài.

Đầu tiên ta bắt đầu cấu hình các thông số cho PLC và kết nối máy tính làm Server trung gian với PLC, từ đó bắt đầu đọc các dữ liệu từ PLC theo một chu kỳ. Tiếp đến ở phía ThingsBoard (MQTT Broker) sẽ kiểm tra tình trạng Active hay Inactive của các MQTT Client (được gọi lên từ MQTT Server). Kiểm tra xem nếu MQTT Client đã Active thì ThingsBoard sẽ nhận toàn bộ các dữ liệu cảm biến (telemetry) được gọi lên định kỳ (hai giây một lần) qua MQTT Broker, nếu chưa Active thì sẽ quay lại trạng thái chờ. Cùng lúc đó sẽ tiến hành thiết lập kết nối đến MySQL, so sánh dữ liệu vừa thu thập

được so với ngưỡng cài đặt trước nếu như có tín hiệu Trigger Alarm (ví dụ: dòng điện vượt quá mức) được gửi lên từ MQTT Server thì sẽ cảnh báo lên cho ThingsBoard biết có sự cố xảy ra, lúc đó dữ liệu về sự cố đó sẽ được gửi lên ThingsBoard ngay lập tức.

Đối với dữ liệu gây cảnh báo khi gửi lên ThingsBoard sẽ được phân loại để biết dữ liệu này thuộc ROOM nào, trình trạng của cảnh báo là gì (Critical/Major) và sao đó sẽ được gửi lên giao diện Dashboard cũng như là gửi E-mail cảnh báo sự cố về cho người dùng.



S7 1200		
Device details		
Details	Attributes	Latest telemetry
Telemetry		
☐	Last update time	Key ↑ Value
☐	2025-05-16 06:03:45	Current_I1 1
☐	2025-05-16 06:03:45	Current_I1_CLK2 1
☐	2025-05-16 06:03:45	Frequency 50
☐	2025-05-16 06:03:45	Frequency_CLK2 50
☐	2025-05-16 06:03:45	kW1 0.4000000059604645
☐	2025-05-16 06:03:45	kW1_CLK2 0.4000000059604645
☐	2025-05-16 06:03:45	kWh 4800
☐	2025-05-16 06:03:45	kWh_CLK2 4800
☐	2025-05-16 06:03:45	kWh_mul 192000
☐	2025-05-16 06:03:45	sqlInsert_Trigger false
☐	2025-05-16 06:03:45	Voltage_V1N 220
☐	2025-05-16 06:03:45	Voltage_V1N_CLK2 220

Hình 4.2: Tổng hợp dữ liệu gửi lên ThingsBoard tại mục Device.

4.2. Xây dựng chức năng Alarm (cảnh báo) trên ThingsBoard

Trong đề tài này, việc tạo ra các cảnh báo giả lập sẽ được thực hiện bằng cách thay đổi các giá trị giám sát (Monitor Value) về điện áp pha và dòng trên pha qua phần mềm TIA Portal sao cho các giá trị này vượt các ngưỡng quy định cho phép.

Các mức cảnh báo trong đề tài sẽ chia thành 2 loại gồm: Major (mức cao) và Critical (mức nguy hiểm).

4.2.1. Xử lý trong ThingsBoard để tạo cảnh báo đến giao diện người dùng

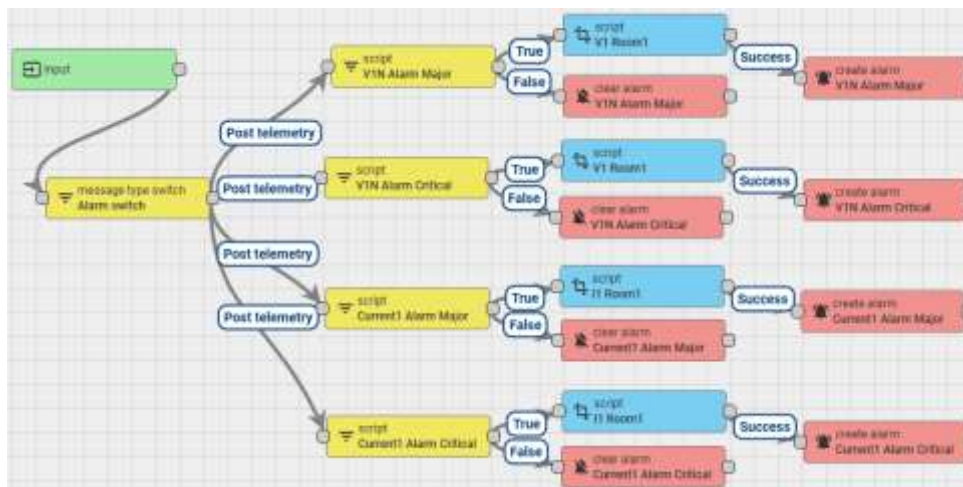
Việc xử lý dữ liệu cảnh báo trong ThingsBoard sẽ thông qua lần lượt các Node dưới đây:

- Node Input: Đây là điểm đầu vào của Rule Chain, nơi đây sẽ nhận dữ liệu Trigger Alarm được giả lập bằng cách thay đổi giá trị giám sát trên phần mềm TIA Portal.
- Node Message Type Switch: Node này có chức năng phân luồng dữ liệu sau đó tiếp tục chia dữ liệu theo các điều kiện khác nhau của các loại cảnh báo và được kết nối với các Node Script thông qua mối quan hệ “Post Telemetry”.
- Node Script: Sau khi dữ liệu được chia vào Node này rồi thì dựa trên yêu cầu về dữ liệu bên trong Node thì nó sẽ thực hiện việc tạo cảnh báo (thông qua mối quan hệ True) hoặc là xóa cảnh báo (thông qua mối quan hệ False).

```
function Filter(msg, metadata, msgType) {
```

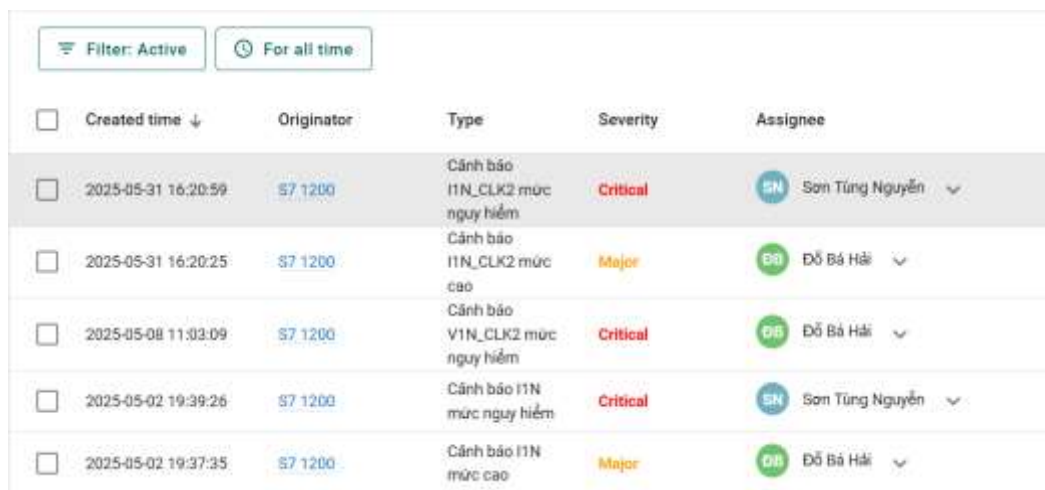
```
1 return msg.Voltage_V1N >= 250 && msg.Voltage_V1N < 300;
```

Hình 4.3: Đoạn code JavaScript về tạo cảnh báo mức Major trong Node Script.



Hình 4.4: Root Rule Chain để xử lý dữ liệu và tạo cảnh báo.

Sau khi xong việc tiến thành xử lý Rule Chain để phân chia từng loại dữ liệu tương ứng với các mức cảnh báo trong đề tài này thì kết quả về cảnh báo sẽ được hiển thị trên giao diện Dashboard người dùng.



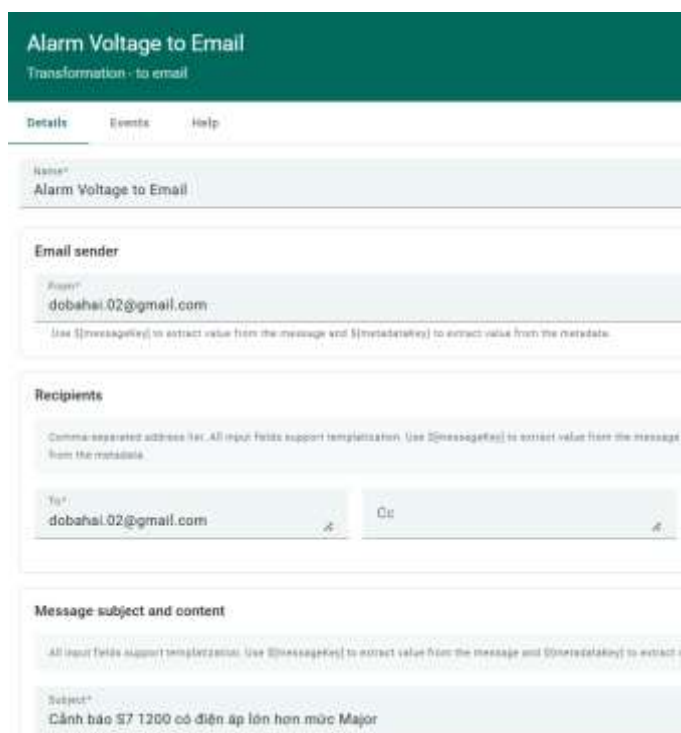
☐	Created time ↓	Originator	Type	Severity	Assignee
☐	2025-05-31 16:20:59	S7 1200	Cảnh báo I1N_CLK2 mức nguy hiểm	Critical	SN Sơn Tùng Nguyễn
☐	2025-05-31 16:20:25	S7 1200	Cảnh báo I1N_CLK2 mức cao	Major	DB Đỗ Bá Hải
☐	2025-05-08 11:03:09	S7 1200	Cảnh báo V1N_CLK2 mức nguy hiểm	Critical	DB Đỗ Bá Hải
☐	2025-05-02 19:39:26	S7 1200	Cảnh báo I1N mức nguy hiểm	Critical	SN Sơn Tùng Nguyễn
☐	2025-05-02 19:37:35	S7 1200	Cảnh báo I1N mức cao	Major	DB Đỗ Bá Hải

Hình 4.5: Kết quả sau khi xử lý dữ liệu Alarm.

4.2.2. Xử lý Rule Chain để gửi E-mail đến người dùng

Việc gửi E-mail cảnh báo đến người dùng sẽ được thực hiện qua giao thức SMTP và thiết lập các Node để xử lý trong Rule Chain. Các Node này gồm 2 Node chính đó là:

- Node To E-mail: Node này dùng để thiết lập các dữ liệu liên quan đến nội dung của E-mail như tiêu đề E-mail được gửi, nội dung chính của E-mail, thông tin của người gửi và người nhận, thông tin thiết bị, các giá trị đó được,...



Alarm Voltage to Email
Transformation - to email

Details Events Help

Name*
Alarm Voltage to Email

Email sender
From*
dobahai.02@gmail.com
Use \${messagekey} to extract value from the message and \${metakey} to extract value from the metadata.

Recipients
Comma-separated address list. All input fields support templating. Use \${messagekey} to extract value from the message or from the metadata.
To*
dobahai.02@gmail.com Cc

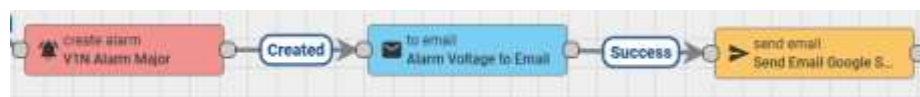
Message subject and content
All input fields support templating. Use \${messagekey} to extract value from the message and \${metakey} to extract value from the metadata.
Subject*
Cảnh báo S7 1200 có điện áp lớn hơn mức Major

Hình 4.6: Thiết lập thông tin trong Node To E-mail.

- Node Send E-mail: Node này có chức năng gửi E-mail cảnh báo bằng dịch vụ SMTP Server được cấu hình sẵn trước đó. Trong Node này, phần Username là tên đầy đủ của tài khoản Gmail được dùng để thiết lập SMTP Gmail, phần Password là mật khẩu của ứng dụng dành riêng cho thiết bị của mình.

Hình 4.7: Thiết lập các thông số về E-mail Google SMTP.

Sự kết hợp giữa 2 Node “To E-mail” và “Send E-mail” trong Transformation Node phải được kết nối qua mối quan hệ Success.



Hình 4.8: Rule Chain xử lý việc tạo và gửi E-mail.

Sau khi thiết lập cấu hình SMTP Gmail thành công thì Password sẽ được cấp dành riêng cho thiết bị của mình.



Hình 4.9: Password được SMTP Gmail của Google cấp.

Sau khi thiết lập, cảnh báo Alarm sẽ được gửi đến người dùng thông qua SMTP Server với những thông tin đã được cấu hình sẵn bên trong Node “To E-mail”.



Hình 4.10: Kết quả gửi cảnh báo về E-mail khi có sự cố xảy ra về điện áp.

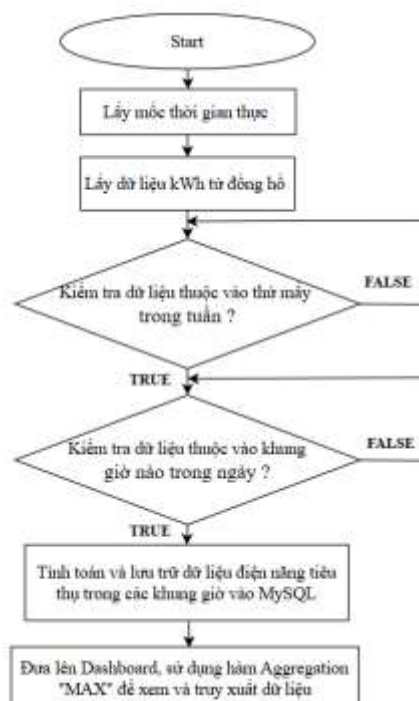
4.3. Truy xuất dữ liệu điện năng theo thời gian thực cho từng khung giờ

Đây cũng chính là một trong những yêu cầu quan trọng cũng như chức năng chính trong đề tài quản lý năng lượng lần này. Trong quá trình tìm hiểu và nghiên cứu thì đề án đã nghiên cứu được ra hai phương pháp xử lý dữ liệu điện năng tiêu thụ theo thời gian thực.

Cách đầu tiên là viết code PLC bằng ngôn ngữ lập trình Ladder từ đó đưa dữ liệu cuối cùng lên ThingsBoard, cách thứ hai là đưa dữ liệu điện năng tiêu thụ trực tiếp từ PLC lên ThingsBoard từ đó mới bắt đầu truy xuất dữ liệu điện năng tiêu thụ bằng cách xử lý dữ liệu trực tiếp trên nền tảng ThingsBoard bằng ngôn ngữ lập trình JavaScript và xử lý trong Rule Chain. Đề án sẽ trình bày cả hai cách từ đó nêu ra ưu và nhược điểm của từng cách.

4.3.1. Xử lý dữ liệu điện năng theo từng khung giờ bằng cách viết code PLC

4.3.1.1. Lưu đồ thuật toán

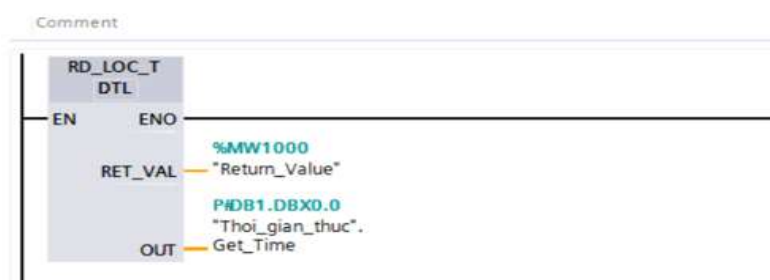


Hình 4.11: Lưu đồ thuật toán tính điện năng tiêu thụ theo từng khung giờ bằng PLC.

Ở phương pháp tính toán và xử lý dữ liệu điện năng bằng cách viết code PLC này thì ban đầu ta cần phải lấy mốc thời gian thực sau đó mới lấy dữ liệu kWh trực tiếp từ đồng hồ Selec. Tiếp đến sẽ tiến hành kiểm tra dữ liệu đó thuộc vào ngày nào trong tuần vì ở ngày chủ nhật sẽ có quy định về thời gian của các khung giờ (khung giờ bình thường, cao điểm và thấp điểm) khác nhau. Cuối cùng là kiểm tra vùng dữ liệu đó thuộc vào khung giờ nào của ngày nào và lưu vùng dữ liệu đó vào MySQL. Việc truy xuất dữ liệu sẽ được thực hiện qua hàm Aggregation “MAX” trong ThingsBoard.

4.3.1.2. Lấy mốc thời gian thực

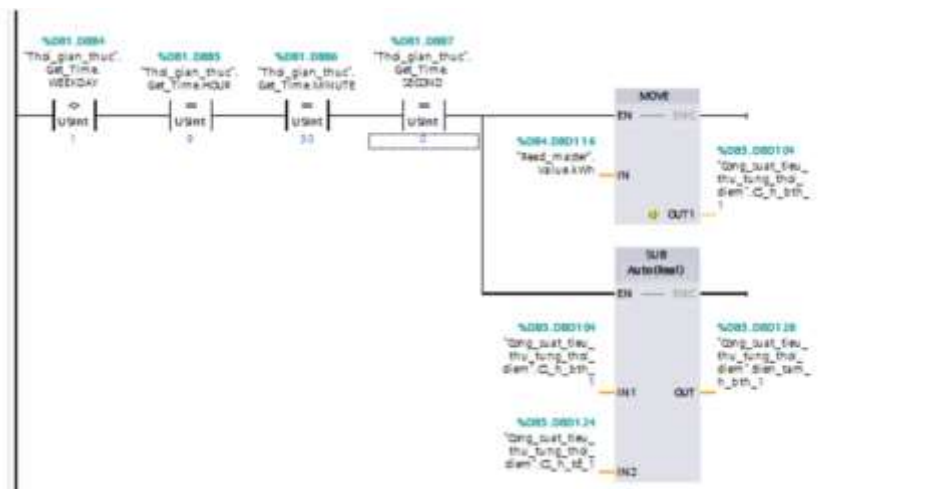
Để tính sản lượng công suất tiêu thụ trong từng khung giờ bình thường, cao điểm, thấp điểm hoặc trong ngày, trong tháng, trong năm, ta cần lấy các mốc thời gian thực.



Hình 4.12: Khối hàm lấy mốc thời gian thực.

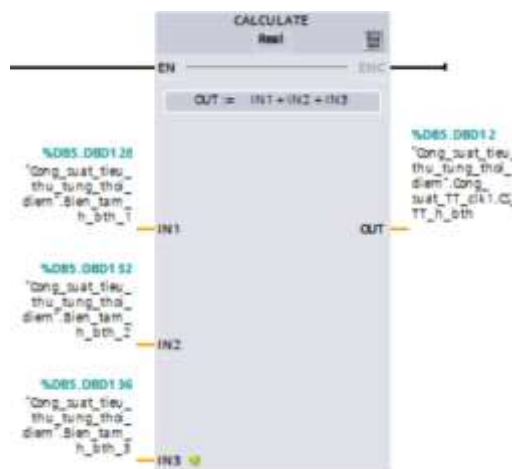
4.3.1.3. Tính công suất tiêu thụ trong khung giờ bình thường

Để tính được công suất tiêu thụ trong khung giờ bình thường, đầu tiên ta xác định xem ngày tính có phải chủ nhật hay không, sau đó ta lấy giá trị đọc được từ các mốc thời gian như 4h, 9h30, 11h30, 17h, 20h, 22h để chuẩn bị tính toán. Nếu như ngày tính không phải ngày chủ nhật, giờ bình thường sẽ được chia thành 3 khung giờ (4h -> 9h30, 11h30 -> 17h, 20h -> 22h) thì ta lấy mốc công suất tiêu thụ tại 9h30 trừ đi mốc công suất tiêu thụ tại 4h, mốc công suất tiêu thụ tại 17h00 trừ đi mốc công suất tiêu thụ tại 11h30, mốc công suất tiêu thụ tại 22h00 trừ đi mốc công suất tiêu thụ tại 20h00 và cộng lại để ra tổng công suất tiêu thụ trong khung giờ bình thường. Còn nếu ngày tính là chủ nhật thì ta lấy công suất tiêu thụ tại 22h00 trừ cho công suất tiêu thụ tại 4h00.



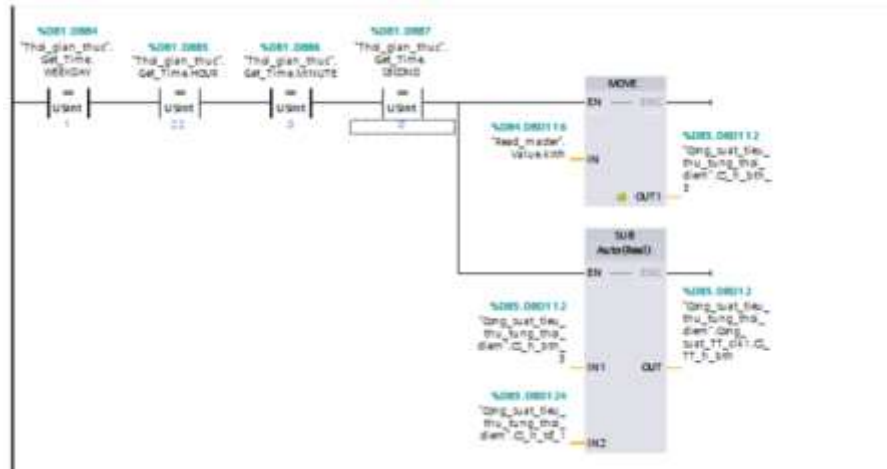
Hình 4.13: Tính công suất tiêu thụ từ 4h đến 9h30.

Các khoảng thời gian (từ 4h đến 9h30, 11h30 đến 17h00 và 20h00 đến 22h00) thuộc khung giờ bình thường cũng đều được tính tương tự như trên. Cuối cùng để cho ra khung giờ bình thường thì ta sẽ cộng tổng 3 khoảng thời gian trên.



Hình 4.14: Tính tổng công suất tiêu thụ trong khung giờ bình thường.

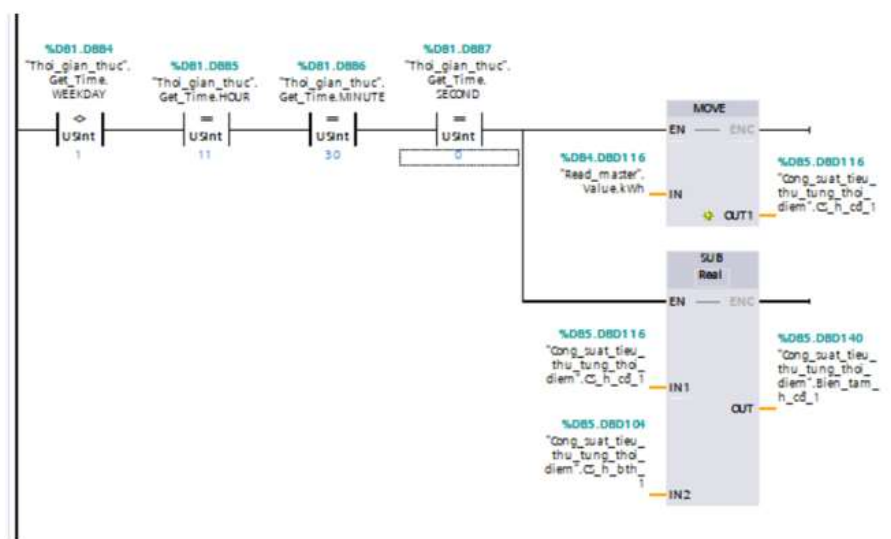
Vì khung giờ bình thường trong ngày chủ nhật sẽ được quy định khác nên ta phải tính riêng công suất tiêu thụ trong giờ bình thường của ngày chủ nhật.



Hình 4.15: Tính công suất tiêu thụ khung giờ bình thường vào ngày chủ nhật.

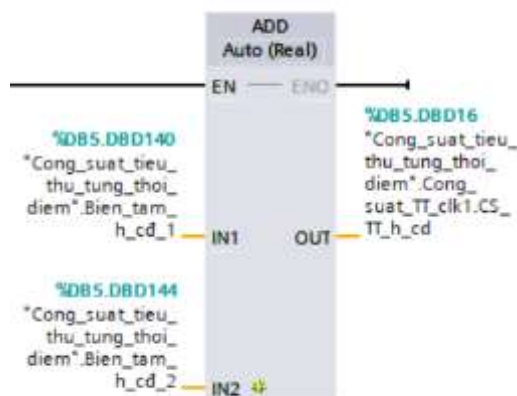
4.3.1.4. Tính công suất tiêu thụ trong khung giờ cao điểm.

Để tính được công suất tiêu thụ trong khung giờ cao điểm, đầu tiên ta xác định xem ngày tính có phải chủ nhật hay không, sau đó ta lấy giá trị đọc được từ các mốc thời gian như 9h30, 11h30, 17h, 20h để chuẩn bị tính toán. Nếu như ngày tính không phải ngày chủ nhật, giờ bình thường sẽ được chia thành 2 khung giờ (9h30 -> 11h30, 17h00 -> 20h00) thì ta lấy mốc công suất tiêu thụ tại 11h30 trừ đi mốc công suất tiêu thụ tại 9h30, mốc công suất tiêu thụ tại 20h00 trừ đi mốc công suất tiêu thụ tại 17h00, cộng lại để ra tổng công suất tiêu thụ trong khung giờ cao điểm. Chủ nhật sẽ không có giờ cao điểm.



Hình 4.16: Tính công suất tiêu thụ từ 9h30 đến 11h30.

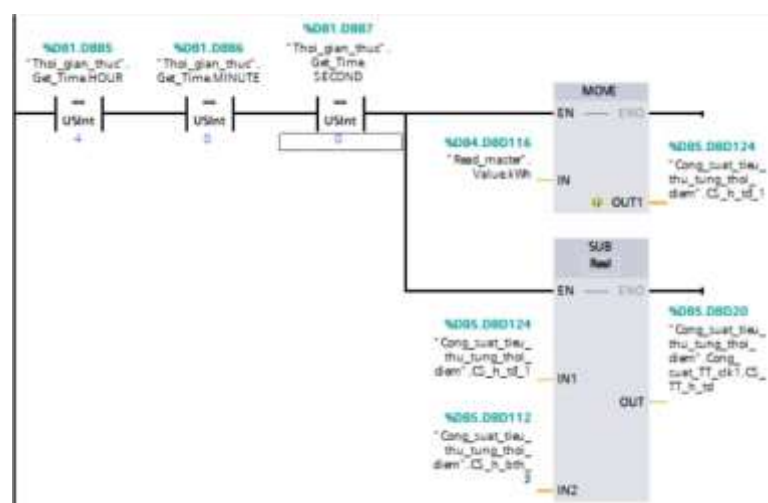
Các khoảng thời gian (từ 9h30 đến 11h30 và từ 17h00 đến 20h00) thuộc khung giờ cao điểm cũng đều được tính tương tự như trên. Để cho ra khung giờ cao điểm thì ta sẽ cộng tổng 2 khung giờ trên.



Hình 4.17: Tính tổng công suất tiêu thụ trong khung giờ cao điểm.

4.3.1.5. Tính công suất tiêu thụ trong khung giờ thấp điểm

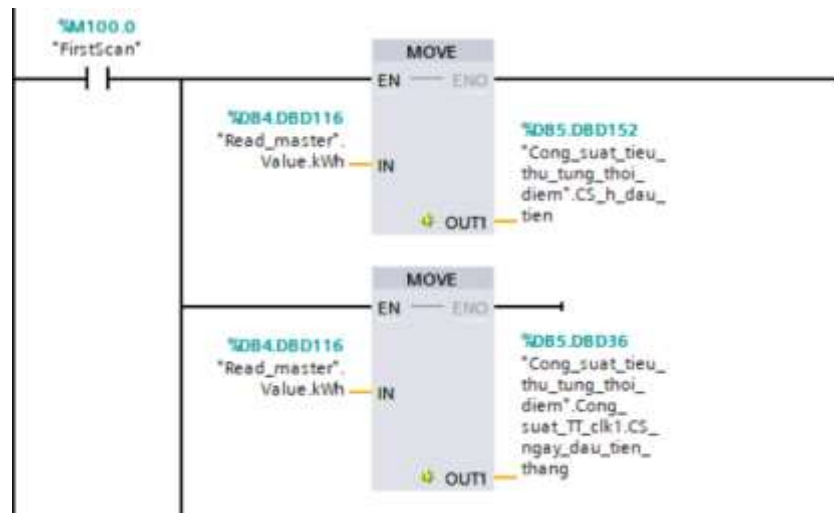
Để tính được công suất tiêu thụ trong khung giờ thấp điểm ta lấy giá trị đọc được từ các mốc thời gian như 22h00 đến 4h00 để chuẩn bị tính toán. Để tính lượng công suất tiêu thụ trong giờ thấp điểm thì ta lấy mốc công suất tiêu thụ tại 4h00 trừ đi mốc công suất tiêu thụ tại 22h00.



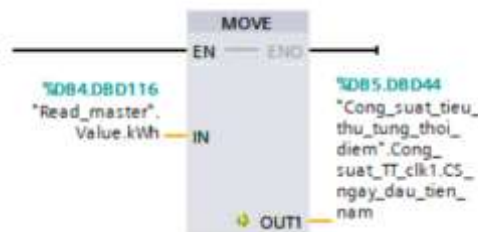
Hình 4.18: Tính công suất tiêu thụ khung giờ thấp điểm.

4.3.1.6. Giá trị công suất tại thời điểm đầu ngày, đầu tháng, đầu năm

Để tính giá trị công suất tiêu thụ trong ngày, tháng, năm, ta lấy mốc thời gian tại thời điểm đầu ngày, đầu tháng, đầu năm.



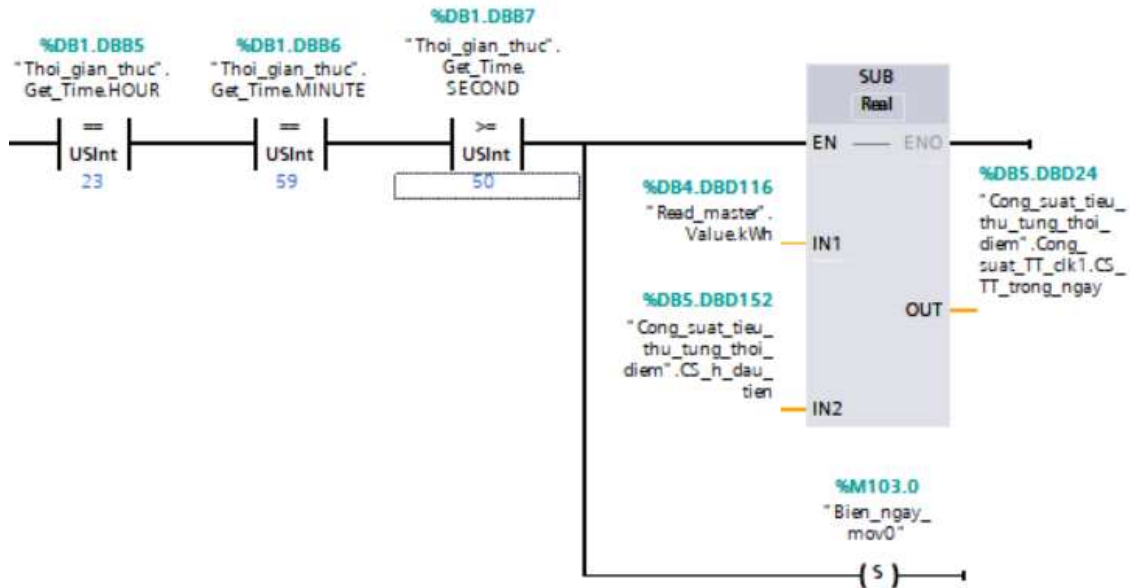
Hình 4.19: Giá trị công suất đầu ngày, đầu tháng lần đầu bật.



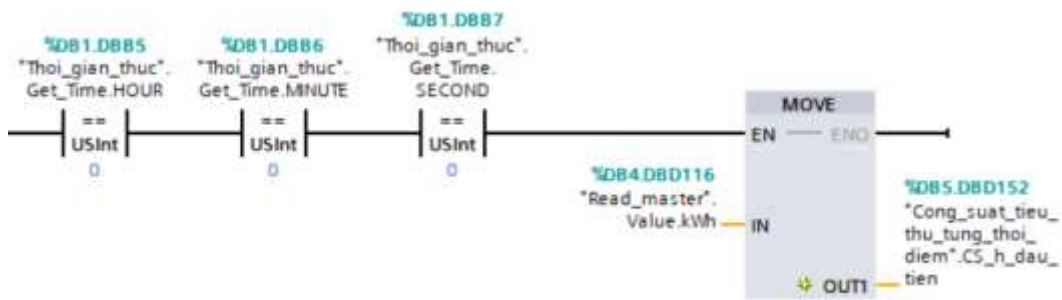
Hình 4.20: Giá trị công suất đầu năm lần đầu bật.

4.3.1.7. Tính giá trị công suất tiêu thụ trong ngày

Để tính công suất tiêu thụ trong ngày, ta sẽ lấy giá trị kWh từ lúc bật PLC lần đầu và bắt đầu lấy giá trị kWh tại mốc thời gian từ 23h59p50s (do khoảng 4s quét dữ liệu 1 lần) để tính lượng công suất tiêu thụ trong ngày, sau đó sẽ lấy giá trị kWh từ 00h00p00s (gọi là giá trị công suất đầu ngày) và tính giá trị công suất tiêu thụ trong ngày như bình thường. Sau khi tính xong ta chờ 1 khoảng thời gian để reset giá trị về 0 để chuẩn bị cho lần tính tiếp theo.



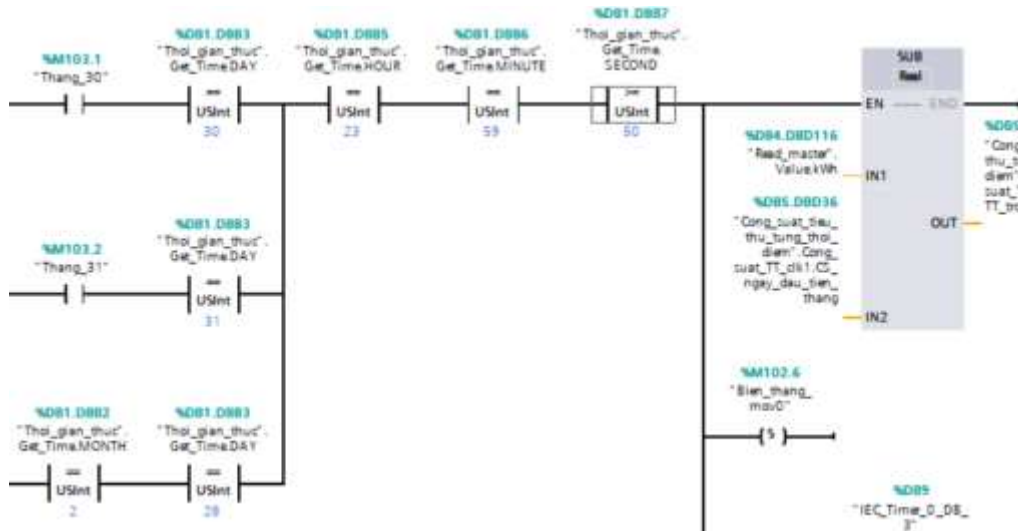
Hình 4.21: Tính công suất tiêu thụ trong ngày.



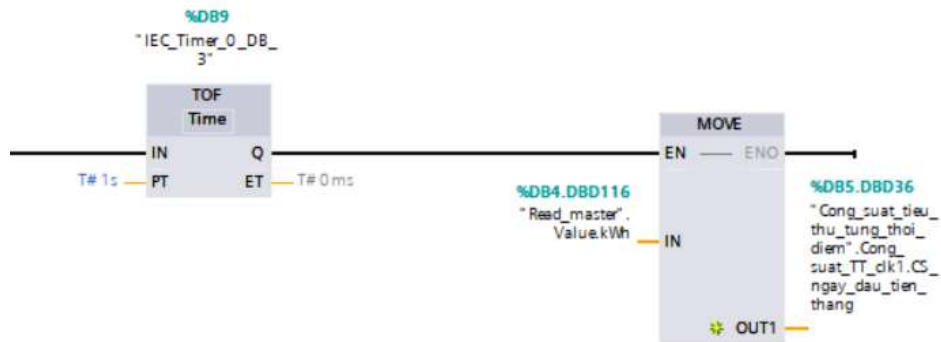
Hình 4.22: Lấy giá trị công suất đầu ngày.

4.3.1.8. Tính giá trị công suất tiêu thụ trong tháng

Để tính giá trị công suất tiêu thụ trong tháng, ta cần xác định xem tháng đó có 30 ngày hay 31 ngày hoặc có phải tháng 2 hay không. Nếu là tháng có 30 ngày thì sẽ lấy giá trị công suất cuối tháng tại ngày 30, nếu tháng có 31 ngày thì sẽ lấy giá trị công suất cuối tháng tại ngày 31, nếu là tháng 2 thì sẽ lấy giá trị công suất cuối tháng tại ngày 28. Lấy giá trị kWh từ lúc bật PLC lần đầu và giá trị kWh tại mốc thời gian từ 23h59p50s 23h59p50s (do khoảng 4s quét dữ liệu 1 lần) vào cuối tháng để tính lượng công suất tiêu thụ trong tháng, sau đó sẽ lấy giá trị kWh từ ngày đầu tháng tại ngày đầu tháng vào lúc 00h00p00s (gọi là giá trị công suất đầu tháng) và tính giá trị công suất tiêu thụ trong tháng như bình thường. Sau khi tính xong ta chờ 1 khoảng thời gian để reset giá trị về 0 để chuẩn bị cho lần tính tiếp theo.



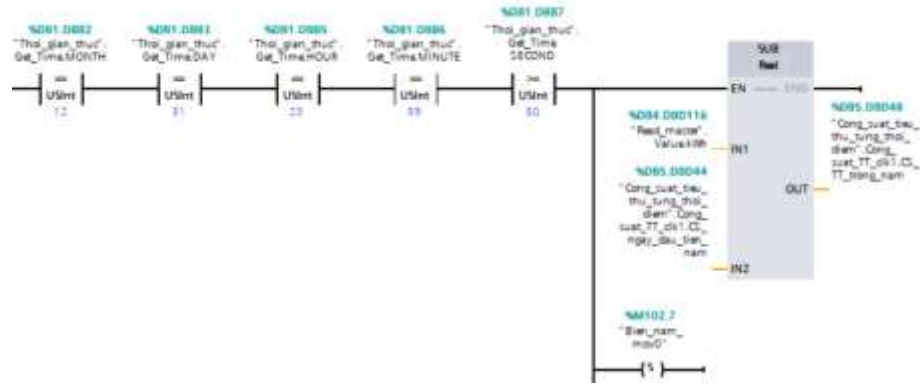
Hình 4.23: Tính công suất tiêu thụ trong tháng.



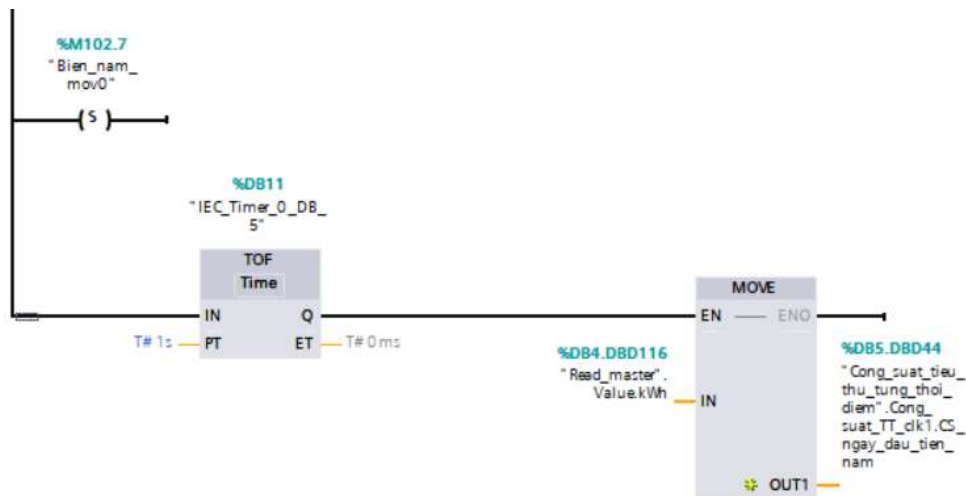
Hình 4.24: Lấy giá trị công suất tại thời điểm đầu tháng.

4.3.1.9. Tính giá trị công suất tiêu thụ trong năm

Để tính giá trị công suất tiêu thụ trong năm, lấy giá trị kWh từ lúc bật PLC lần đầu và giá trị kWh tại mốc thời gian từ ngày 31 tháng 12 23h59p50s (do khoảng 4s quét dữ liệu 1 lần) để tính lượng công suất tiêu thụ trong năm, sau đó sẽ lấy giá trị kWh từ ngày đầu năm vào lúc 00h00p00s (gọi là giá trị công suất đầu năm) và tính giá trị công suất tiêu thụ trong năm như bình thường. Sau khi tính xong ta chờ 1 khoảng thời gian để reset giá trị về 0 để chuẩn bị cho lần tính tiếp theo.



Hình 4.25: Tính công suất tiêu thụ trong năm.



Hình 4.26: Lấy giá trị công suất tại thời điểm đầu năm.

4.3.1.10. Ưu và nhược điểm của phương pháp xử lý dữ liệu điện năng theo từng khung giờ bằng cách viết code PLC.

Ưu điểm:

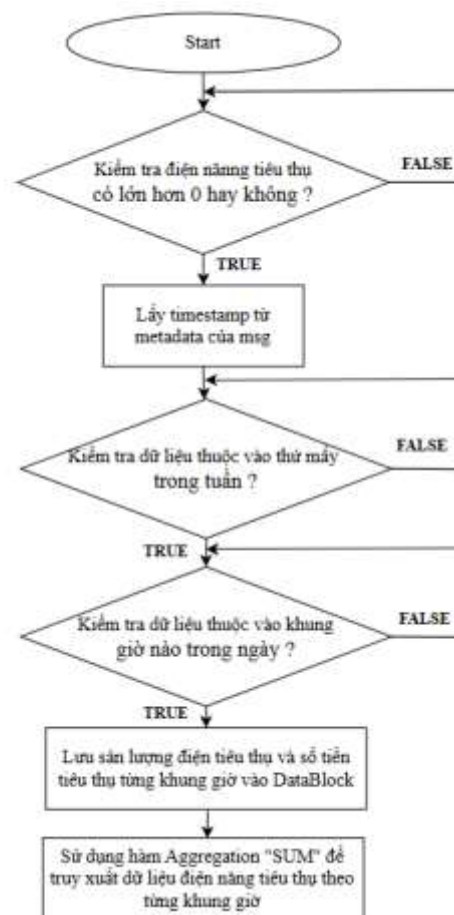
- Việc tính toán dữ liệu điện năng theo từng khung giờ bằng cách viết Code PLC sẽ dễ dàng thực hiện việc tính toán hơn dựa vào ngôn ngữ lập trình Ladder, người dùng có thể dễ dàng hiểu được cách tính toán này nếu có kiến thức về lập trình PLC.
- Phương pháp tính toán này sẽ đưa ra dữ liệu trực tiếp về công suất tiêu thụ tại các khung giờ ở chương trình PLC. Từ đó ta có thể dễ dàng đưa trực tiếp các dữ liệu về công suất tiêu thụ tại các khung giờ này lên nền tảng IoT ThingsBoard theo dạng “Key Value” thông qua MQTT Server như các loại dữ liệu khác mà không cần phải xử lý dữ liệu trực tiếp trong Rule Chain.
- Dữ liệu sẽ được lưu trực tiếp vào MySQL, việc này giúp tăng tính bảo mật của dữ liệu.

Nhược điểm:

- Vì dữ liệu được tính toán và phân chia ra quá nhiều ở chương trình PLC nên việc gửi các dữ liệu từ PLC theo dạng lên nền tảng theo dạng “Key Value” sẽ gây ra hiện tượng quá tải dữ liệu, chưa tối ưu hóa được lượng Telemetry data gửi lên ThingsBoard nên gây ra tình trạng mất một số dữ liệu cũng như gây tổn hao tài nguyên trên ThingsBoard.

4.3.2. Xử lý dữ liệu điện năng theo từng khung giờ trên ThingsBoard.

4.3.2.1. Lưu đồ thuật toán



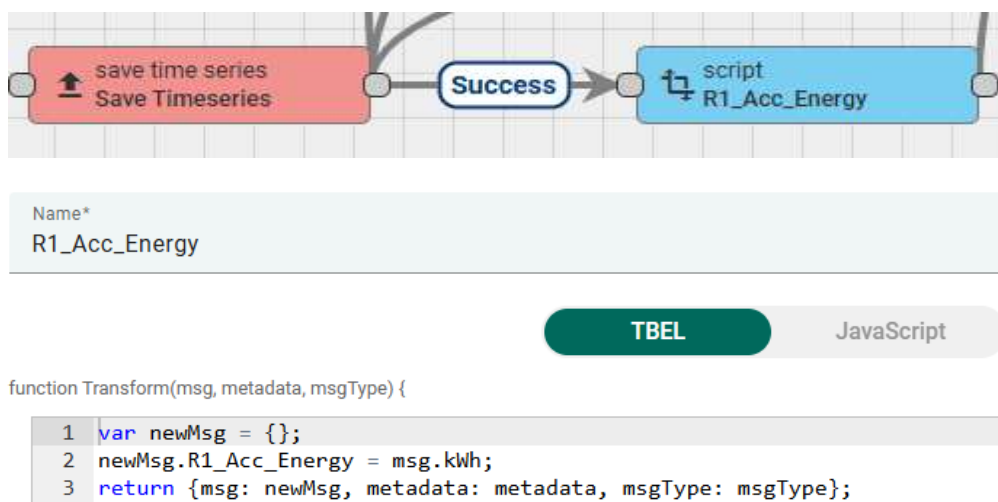
Hình 4.27: Lưu đồ thuật toán để tính và truy xuất dữ liệu điện năng theo từng khung giờ trong ThingsBoard.

Đầu tiên thuật toán sẽ thực hiện việc kiểm tra về dữ liệu điện năng tiêu thụ của đồng hồ Selec. Khi hệ thống hoạt động thì dữ liệu điện năng tiêu thụ sẽ tăng theo lũy tiến khi đó bắt đầu lấy dữ liệu từ Metadata trong **msg** của ThingsBoard. Tiếp đến sẽ tiến hành việc kiểm tra vùng dữ liệu đó thuộc vào ngày nào trong tuần vì ở ngày chủ nhật sẽ có quy định về thời gian của các khung giờ (khung giờ bình thường, cao điểm và thấp điểm) khác nhau. Cuối cùng là kiểm tra vùng dữ liệu đó thuộc vào khung giờ nào của

ngày nào và lưu vùng dữ liệu đó vào DataBlock. Việc truy xuất dữ liệu sẽ được thực hiện qua hàm Aggregation “SUM” trong ThingsBoard.

4.3.2.2. Xử lý dữ liệu điện năng được gửi lên từ MQTT Server

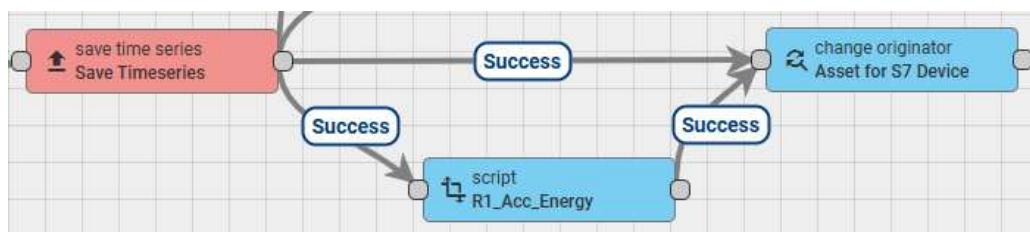
Các biến dữ liệu (dòng điện, điện áp, tần số, công suất tác dụng, điện năng tiêu thụ) được PLC gửi lên ThingsBoard thông qua MQTT Server sẽ bao gồm rất nhiều biến dữ liệu khác nhau nhưng trong phần này ta chỉ cần lấy dữ liệu về điện năng tiêu thụ của tải. Do đó ta cần phải thực hiện việc lọc dữ liệu để kết quả trả về là dữ liệu điện năng tiêu thụ duy nhất bằng cách thông qua một bước xử lý trong Rule Chain.



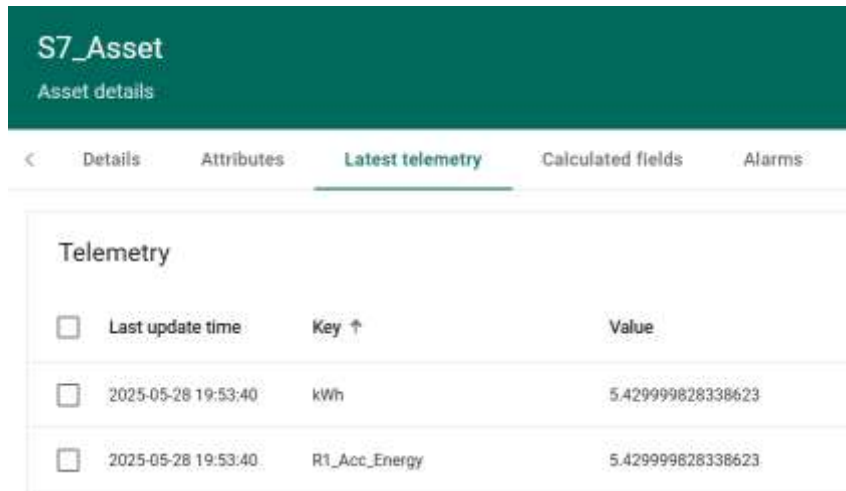
Hình 4.28: Rule Chain và đoạn code để lọc ra giá trị điện năng tiêu thụ.

4.3.2.3. Tạo Asset để lưu trữ dữ liệu điện năng đọc từ Device

Sau khi dữ liệu điện năng tiêu thụ được lọc ra từ hàng loạt các biến dữ liệu được gửi lên từ MQTT Server thì ta cần phải tạo một Asset để lưu giá trị điện năng tiêu thụ vào trong đó nhằm tăng tính bảo mật của dữ liệu cũng như là chia nhỏ dữ liệu ra để có thể quản lý dữ liệu một cách tốt nhất.



Hình 4.29: Rule Chain để chia dữ liệu điện năng vào Asset.



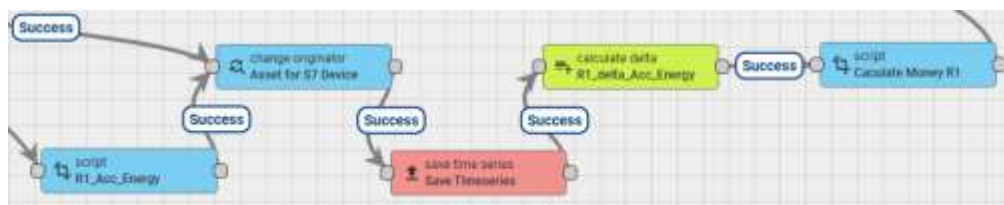
S7_Asset			
Asset details			
Details Attributes Latest telemetry Calculated fields Alarms			
Telemetry			
☐	Last update time	Key ↑	Value
☐	2025-05-28 19:53:40	kWh	5.429999828338623
☐	2025-05-28 19:53:40	R1_Acc_Energy	5.429999828338623

Hình 4.30: Kết quả dữ liệu điện năng đọc lên từ Device sau khi được xử lý vào Asset.

4.3.2.4. Xử lý dữ liệu điện năng trong Rule Chain

Vì thông số điện năng tiêu thụ tích lũy ở đồng hồ Selec MFM383A chỉ có thể tăng lên chứ không giảm xuống nên điện năng tiêu thụ chỉ có thể lớn hơn hoặc bằng 0 theo thời gian. Do đó nên ta chỉ cập nhật dữ liệu khi có điện năng tiêu thụ ở tải có nghĩa là thông số điện năng tiêu thụ tích lũy tăng lên so với lần cập nhật trước đó thì ta mới cập nhật dữ liệu. Để thực hiện được điều này ta cần thực hiện qua 2 Node:

- Node **“enrichment – calculate delta”**: Có chức năng tính toán xem độ chênh lệch điện năng tiêu thụ tích lũy có lớn hơn 0 hay không.
- Node **“transformation – script”**: Được lập trình để tính tiền điện tương ứng với điện năng tiêu thụ theo từng khung giờ với chức năng kiểm tra Timestamp và trả về các Key Value cần thiết để thực hiện việc hiển thị trên Dashboard.

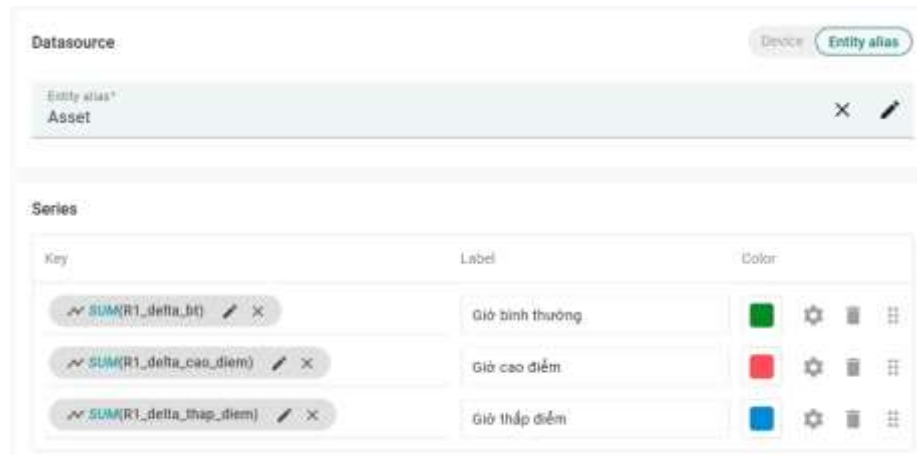


Hình 4.31: Rule Chain để tính và truy xuất dữ liệu điện năng theo thời gian.

4.3.2.5. Xử lý dữ liệu điện năng trong Dashboard

Từ các dữ liệu đã được tính toán trong Node **“transformation – script”** và lưu trữ theo Timestamp dưới dạng Key Value cụ thể thì trong ThingsBoard cung cấp cho người dùng khả năng truy xuất các **“Key Value”** với hàm Aggregation **“SUM”** và TimeWindow (cửa sổ thời gian: cho phép người dùng chọn khoảng thời gian cần truy

xuất dữ liệu). Những lợi thế trên của ThingsBoard giúp giảm thiểu việc người dùng phải lập trình để xuất dữ liệu thuộc ngày nào hay tháng nào.



Hình 4.31: Xử lý trong Dashboard với từng khung giờ cụ thể.

Sau khi kết hợp giữa các tính năng có sẵn trong ThingsBoard và cùng với việc lập trình xử lý dữ liệu từng bước trong Rule Chain thì cuối cùng ta có thể dễ dàng truy xuất dữ liệu điện năng tiêu thụ trong từng khoảng thời gian cụ thể cũng như tiền điện tương ứng trong từng khoảng thời gian đó.

Kết quả cuối cùng cho ta thấy hình dưới là truy xuất tổng sản lượng điện năng tiêu thụ trong từng khung giờ kèm và tiền điện tương ứng với từng khung giờ đó, sử dụng hàm Aggregation SUM và trong TimeWindow ta chọn Current day (trong ngày hôm nay) ta sẽ được kết quả như hình dưới.



Hình 4.32: Kết quả cuối cùng sau khi xử lý trong Rule Chain.

4.3.2.6. Ưu và nhược điểm của phương pháp xử lý dữ liệu điện năng theo từng khung giờ trực tiếp trên ThingsBoard

Ưu điểm:

- Phương pháp này giúp tối ưu hóa dữ liệu đầu vào khi gửi lên nền tảng IoT ThingsBoard vì ta chỉ cần gửi duy nhất dữ liệu về điện năng tiêu thụ từ PLC lên ThingsBoard sau đó nhờ vào các công cụ hỗ trợ trong nền tảng và xử lý trực tiếp trong Rule Chain ta có thể xử lý được dữ liệu điện năng theo từng khung giờ (giờ cao điểm, giờ thấp điểm, giờ bình thường) một cách dễ dàng.
- Dữ liệu về điện năng tiêu thụ sẽ được lưu trữ trong cơ sở dữ liệu riêng của ThingsBoard nên người dùng có thể truy xuất dữ liệu một cách dễ dàng.

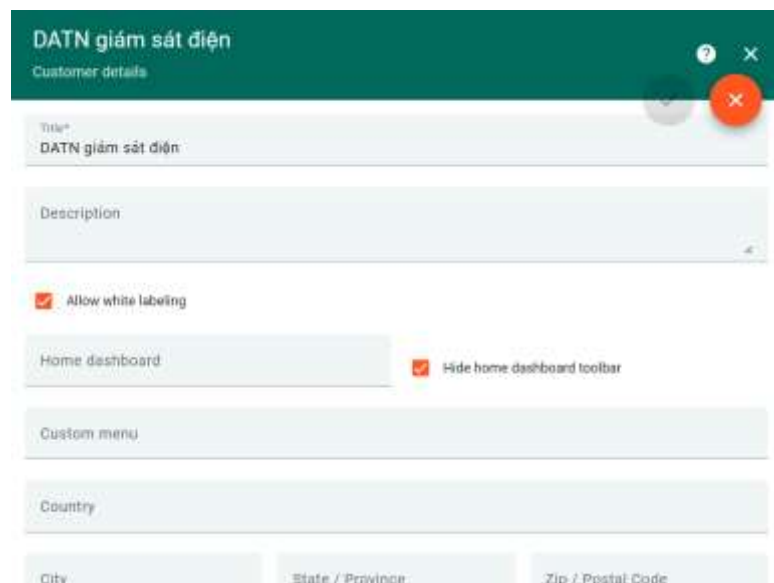
Nhược điểm:

- Vì dữ liệu được lưu trữ trong cơ sở dữ liệu riêng của ThingsBoard nên việc bảo mật dữ liệu không được cao.
- Phương pháp này đòi hỏi người lập trình phải am hiểu về các công cụ hỗ trợ cũng như việc xử lý dữ liệu trực tiếp trong Rule Chain.

4.4. Xây dựng chức năng phân quyền và quản lý người dùng

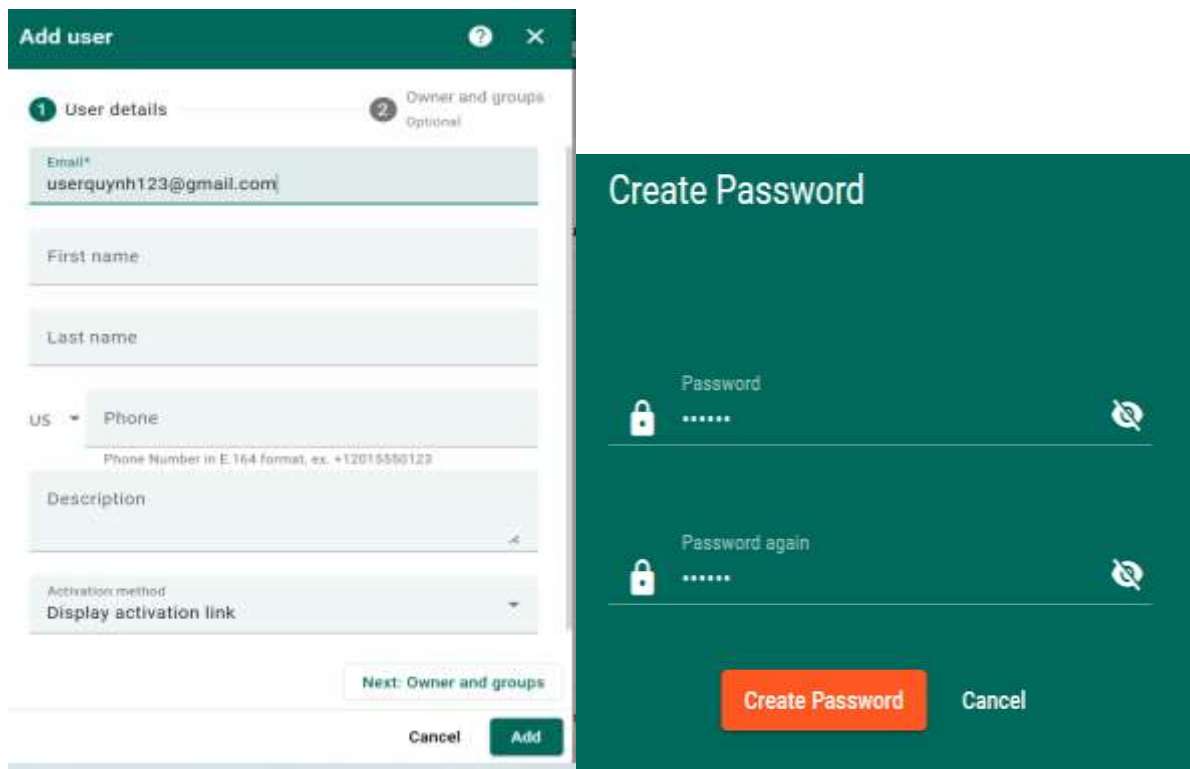
Để có thể quản lý và giám sát các thông số một cách hiệu quả, cần phân quyền cho một số cá nhân để quản lý từng khu vực riêng và không can thiệp lẫn nhau, chỉ để người có quyền cao nhất quản lý tất cả.

Đầu tiên, vào ThingsBoard chọn Customer và đặt tên là “DATN giám sát điện”.

The image shows a screenshot of the ThingsBoard web interface for creating a new customer. The title bar at the top is green and says "DATN giám sát điện" with a subtitle "Customer details". Below the title bar, there are several input fields: "Title" (containing "DATN giám sát điện"), "Description", "Allow white labeling" (checked), "Home dashboard" (with a dropdown menu), "Hide home dashboard toolbar" (checked), "Custom menu", "Country", "City", "State / Province", and "Zip / Postal Code". There are also some UI elements like a red close button and a question mark icon in the top right corner.

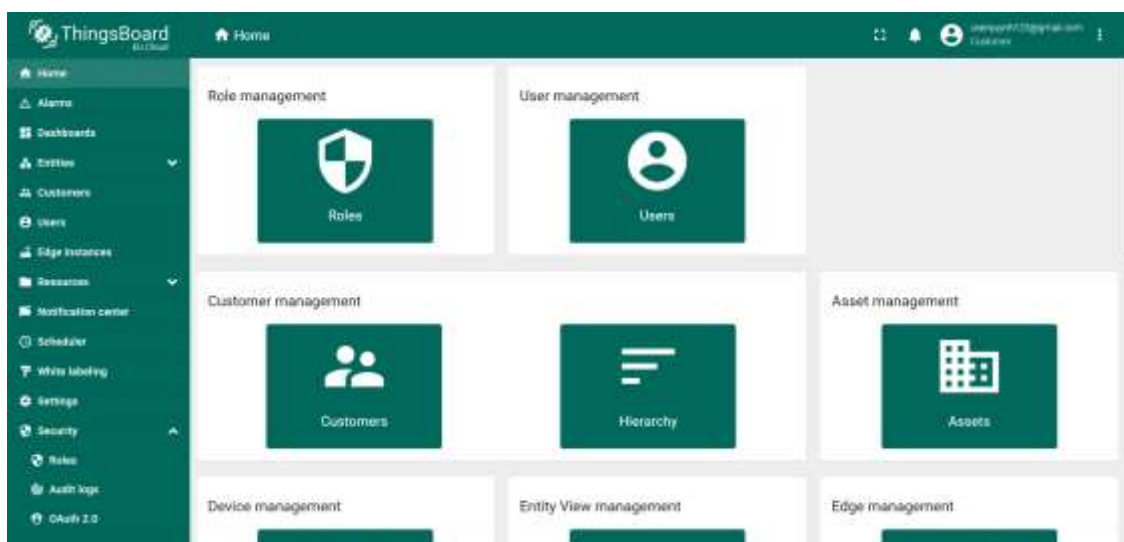
Hình 4.33: Tạo Customer để quản lý người dùng.

Sau khi đã tạo Customer xong, vào “Manage Customer User” để thêm các User vào, mục đích để cấp quyền cho người vận hành được phép giám sát một số Dashboard được chỉ định. Sau đó tạo mật khẩu cho tài khoản vừa tạo.

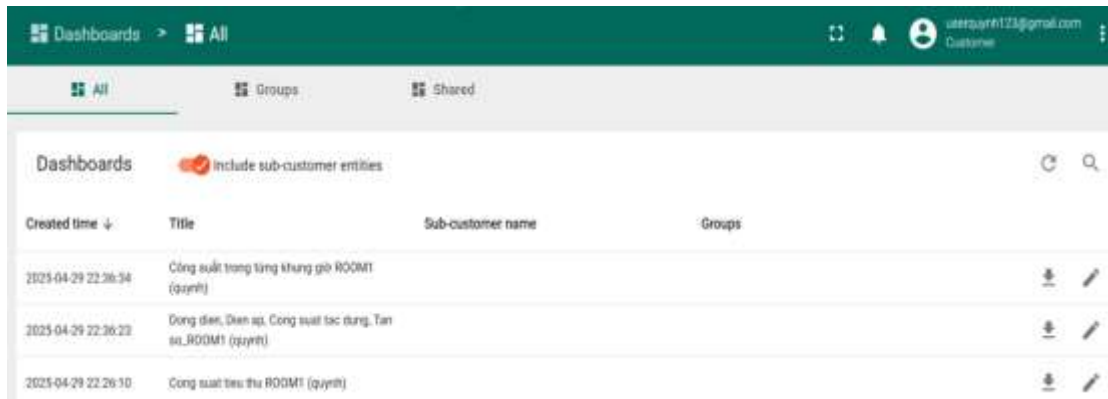


Hình 4.34: Tạo mật khẩu và tài khoản cho User.

Sau khi đã tạo tài khoản và mật khẩu, vào Dashboard chọn “Manage owner and groups” ủy quyền các Dashboard cho User vừa tạo. User chỉ được truy cập vào các Dashboard mà được chỉ định trước đó.



Hình 4.35: Màn hình khi User được phân quyền đăng nhập vào.



Hình 4.36: Các Dashboard được chỉ định dành riêng cho User.

4.5. Kết luận chương 4

Trong chương 4 này đã trình bày cụ thể quá trình thu thập dữ liệu và xử lý dữ liệu thô từ PLC lên nền tảng IoT ThingsBoard. Từ đó thiết kế và xây dựng các chức năng cần thiết của một hệ thống giám sát năng lượng. Đặc biệt đề án cũng đã thực hiện được chức năng tính toán công suất tiêu thụ trong từng khung giờ (giờ cao điểm, giờ thấp điểm, giờ bình thường) bằng hai phương pháp xử lý dữ liệu khác nhau và từ đó nêu ra được ưu điểm và nhược điểm của từng phương pháp đó. Điều này giúp cho người chủ đầu tư có nhiều lựa chọn hơn về giải pháp quản lý năng lượng cũng như lựa chọn được phương pháp phù hợp với hệ thống đang có sẵn tại phân xưởng, nhà máy.

Trong chương 5 tiếp theo sẽ trình bày các kết quả đạt được sau khi xây dựng toàn bộ hệ thống và tiến hành đánh giá kết quả từ đó đưa ra các mặt hạn chế cũng như hướng phát triển của đề tài lần này.

CHƯƠNG 5: KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ

5.1. Kết quả mô hình của phần cứng

Mô hình phần cứng thực tế của hệ thống trong đề tài lần này được xây dựng với tải tiêu thụ điện năng là bóng đèn LED 40W. Các dữ liệu điện năng của tải sẽ được đọc từ đồng hồ Selec MFM383A và gửi lên PLC để tổng hợp. Dùng một máy tính làm Server trung gian để nhận các dữ liệu từ PLC và gửi lên nền tảng IoT ThingsBoard thông qua MQTT Server.



Hình 5.1: Mô hình phần cứng thực tế của hệ thống.

5.2. Kết quả mô phỏng giá trị đồng hồ trong PLC

Kết quả được hiển thị như hình bên dưới. Các giá trị có đuôi ở tên là “CLK2” là giá trị được mô phỏng dựa trên giá trị thật của đồng hồ Selec MFM383A.

Read_master					
	Name	Data type	Offset	Start value	Monitor value
[-]	Value_CLK2	"Dong_ho_2"	122.0		
[-]	Voltage_V1N_CLK2	Real	122.0	0.0	224.6
[-]	Current_I1_CLK2	Real	154.0	0.0	0.98
[-]	kW1_CLK2	Real	170.0	0.0	0.137
[-]	Frequency_CLK2	Real	234.0	0.0	50.14
[-]	kWh_CLK2	Real	238.0	0.0	1.34

Hình 5.2: Các thông số điện năng ở đồng hồ 2 sau khi mô phỏng giả lập.

Ta có thể thấy các thông số được mô phỏng thì có giá trị gần tương đương với giá trị của thông số thật.

5.3. Kết quả lưu dữ liệu vào MySQL

Các dữ liệu cần được giám sát như điện áp, dòng điện, tần số, công suất tiêu thụ trong từng khung giờ,... của đồng hồ và các dữ liệu được mô phỏng trong PLC cũng sẽ được lưu vào MySQL theo thời gian thực ở dạng bảng dưới đây.

date_time	Voltage_VIN	Current_I1	Frequency	CS_TT_h_bth	CS_TT_h_cd	CS_TT_h_td	CS_TT_trong_ngay	CS_TT_trong_thang	CS_TT_trong_n
2025-05-12 11:53:01	220	1	50	100	20	0	120	0	0
2025-05-12 11:53:03	220	1	50	100	20	30	150	0	0
2025-05-12 11:53:05	220	1	50	100	20	30	150	0	0
2025-05-12 11:53:07	220	1	50	100	20	30	0	30	0
2025-05-12 11:53:09	220	1	50	100	20	30	0	0	0
2025-05-12 11:53:11	220	1	50	100	20	30	0	0	0
2025-05-12 11:53:13	220	1	50	100	20	30	0	0	0
2025-05-12 11:53:15	220	1	50	100	20	30	0	0	0
2025-05-12 11:53:17	220	1	50	100	20	30	0	0	30
2025-05-12 11:53:19	220	1	50	100	20	30	0	0	30
2025-05-12 11:53:21	220	1	50	100	20	30	0	0	0
2025-05-12 11:53:23	220	1	50	0	20	30	0	0	0

Hình 5.3: Dữ liệu được lưu trữ trong MySQL theo dạng bảng.

Kết quả các dữ liệu được lưu trong dạng bảng, người dùng có thể truy xuất và có một cái nhìn tổng quan về dữ liệu theo thời gian thực.

5.4. Kết quả các Dashboard giám sát trong ThingsBoard

5.4.1. Trang OVERVIEW

Trang OVERVIEW cho ta cái nhìn tổng quan về các dữ liệu được hiển thị theo thời gian thực của hệ thống và có Chart để người dùng có thể quan sát được sự thay đổi của các dữ liệu.

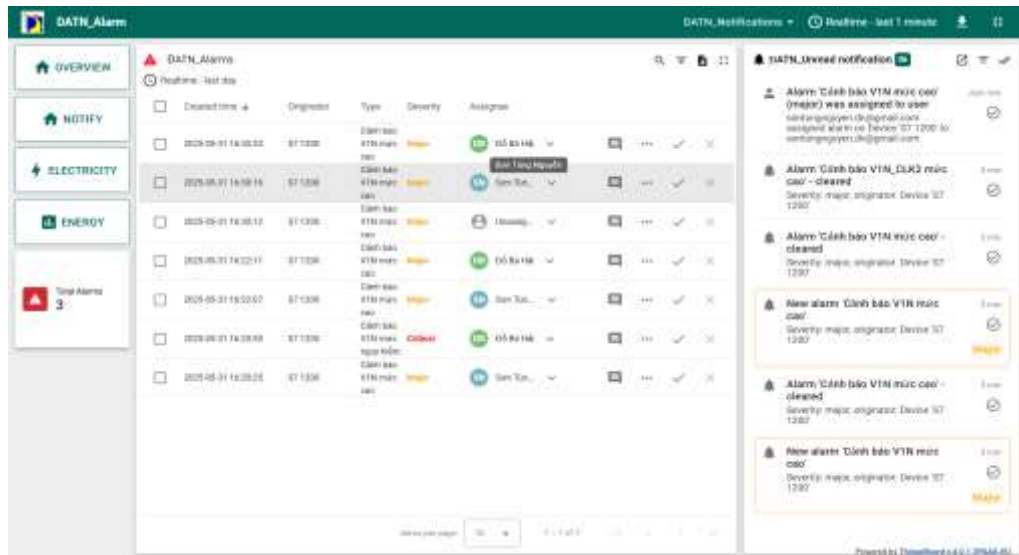


Hình 5.4: Giao diện màn hình trang Overview.

Kết quả hiển thị về các thông số thật của đồng hồ cũng như thông số sau khi được giả lập. Ta có thể thấy các thông số sau khi được giả lập và thông số thật có giá trị gần tương đương với nhau. Bên cạnh đó người dùng có thể biết được tình trạng Alarm cũng như số lượng Alarm của hệ thống.

5.4.2. Trang NOTIFICATION

Hiển thị số lượng các thông báo chưa được đọc, số cảnh báo chưa được xóa và chi tiết của cảnh báo như tình trạng, nguồn gốc, loại cảnh báo, mức cảnh báo. Người dùng có thể in theo dõi hoặc in báo cáo dưới dạng file Excel.

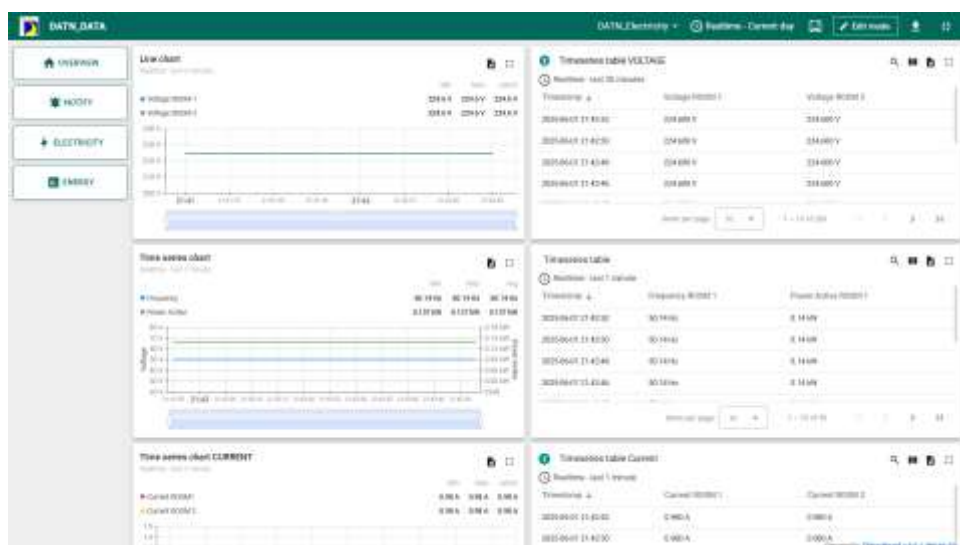


Hình 5.5: Giao diện màn hình trang Notification.

Các cảnh báo gửi về được hiển thị một cách trực quan theo từng giao diện thông báo. Ngoài ra cảnh báo còn được gửi về E-mail người dùng một cách nhanh chóng.

5.4.3. Trang ELECTRICITY

Ở trang này ta có thể xem được chi tiết của từng dữ liệu thống số điện năng bên cạnh đó dữ liệu được hiển thị theo thời gian thực dưới dạng đồ thị và dạng bảng.



Hình 5.6: Giao diện màn hình trang Electricity.

Ngoài ra người dùng còn có thể in kết quả báo cáo dưới dạng file Excel.

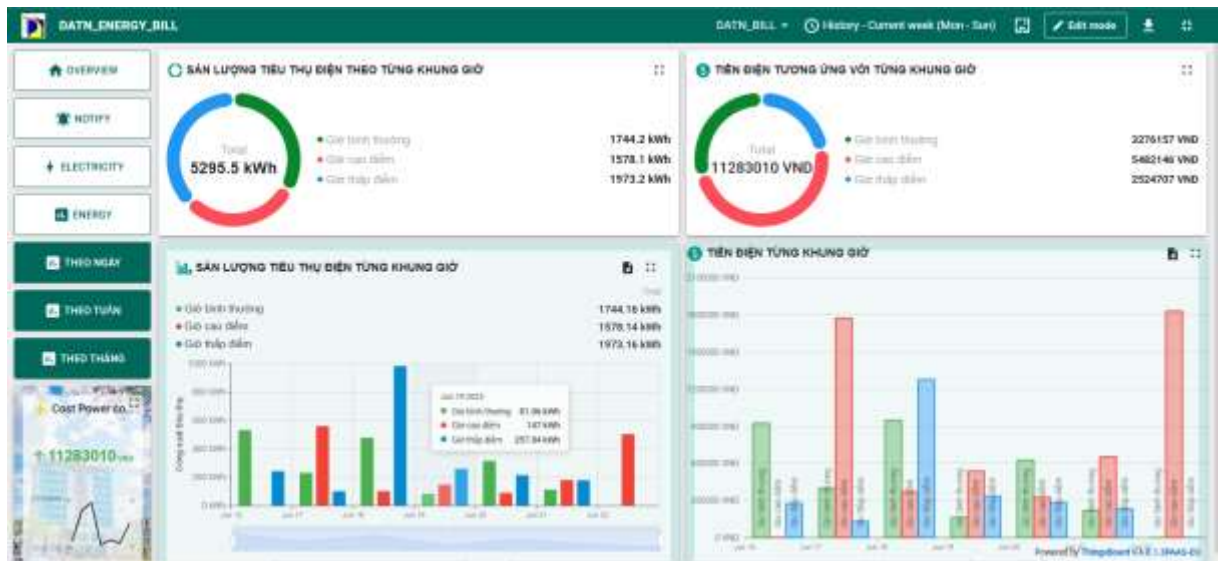
Timestamp	Frequency ROOM 1	Power Active ROOM 1
2025-06-01 21:36:59	50.13999939	0.136999995
2025-06-01 21:37:01	50.13999939	0.136999995
2025-06-01 21:37:03	50.13999939	0.136999995
2025-06-01 21:37:05	50.13999939	0.136999995
2025-06-01 21:37:07	50.13999939	0.136999995
2025-06-01 21:37:09	50.13999939	0.136999995
2025-06-01 21:37:11	50.13999939	0.136999995
2025-06-01 21:37:13	50.13999939	0.136999995
2025-06-01 21:37:15	50.13999939	0.136999995
2025-06-01 21:37:17	50.13999939	0.136999995

Hình 5.7: Kết quả in báo cáo dưới dạng file Excel.

Kết quả truy xuất dữ liệu cho ta thấy dữ liệu được gửi lên nền tảng mỗi 2 giây một lần, điều này đôi khi gây ra sự tắc nghẽn và gây quá tải dữ liệu và không thật sự cần thiết. Để khắc phục điều này ta nên gửi dữ liệu lên nền tảng mỗi 2 phút một lần.

5.4.4. Trang ENERGY

Hiện thị dữ liệu sản lượng điện tiêu thụ trong từng khung giờ ở dạng biểu đồ cột và biểu đồ tròn một cách trực quan.



Hình 5.8: Giao diện màn hình trang ENERGY.

Người dùng có thể xem lịch sử dữ liệu sản lượng tiêu thụ điện từng khung giờ trong ngày, trong tuần, trong tháng dựa vào TimeWindow.



Hình 5.9: Truy xuất dữ liệu điện năng từng khung giờ trong tháng.

Timestamp	Giờ bình thường (kWh)	Giờ cao điểm (kWh)	Giờ thấp điểm (kWh)
16/06/2025 12:00	531		241.92
17/06/2025 12:00	230	559.24	100
18/06/2025 12:00	477.1	101.4	982
19/06/2025 12:00	81.06	147	257.84
20/06/2025 12:00	314.1	88.9	213.9
21/06/2025 12:00	110.9	179.2	177.5
22/06/2025 12:00			
23/06/2025 12:00			106.03

Hình 5.10: Kết quả in báo cáo điện năng tiêu thụ từng khung giờ dưới dạng Excel.

5.5. Những mặt hạn chế của đề tài

Đề án vẫn còn một số điểm hạn chế như sau:

- Vì hạn chế về mặt phần cứng nên trong phần kiến trúc IoT đã lựa chọn một máy tính làm Server trung gian để đưa dữ liệu từ PLC lên nền tảng. Điều này làm cho việc demo quá trình thu thập và xử lý dữ liệu điện năng theo thời gian gặp nhiều khó khăn vì phải treo máy tính liên tục theo thời gian thực để có được dữ liệu nên đôi khi gặp nhiều sự cố trong quá trình demo.
- Việc triển khai dự án và phải trả phí sử dụng trên phiên bản Cloud của ThingsBoard mới chỉ là cơ sở để tiếp cận các công cụ có sẵn trên nền tảng, chưa thể xây dựng được cho mình một Server riêng.
- Vì đề tài này sử dụng tải tiêu thụ ở nguồn điện 1 pha – 2 dây với công suất nhỏ nên việc lấy dữ liệu công suất tiêu thụ tích lũy theo thời gian mất rất nhiều thời gian để có được dữ liệu do đó cần phải tải thêm phần mềm mô phỏng để ta có thể giả lập thông số điện năng tăng nhanh theo thời gian.
- Nếu có sự cố thì cảnh báo sẽ được gửi liên tục mà không có khoảng thời gian Wait time, gây phiền toái hoặc gây sớm tới giới hạn cảnh báo cho phép của nền tảng ThingsBoard.

KẾT LUẬN

- Đồ án nhìn chung đã thực hiện được các yêu cầu của bài toán được đặt ra:
 - Trang Web cung cấp giao diện để người dùng có thể giám sát các thông số dưới dạng bảng hoặc biểu đồ một cách trực quan.
 - Đã tính toán được công suất tiêu thụ từng khung giờ và tiền điện tương ứng với từng khung giờ đó giúp cho người chủ đầu tư có thể dễ dàng quản lý năng lượng tốt hơn cũng như bố trí nhân công hay sắp xếp thời gian các hoạt động sản xuất trong nhà máy để tối ưu hóa chi phí năng lượng.
 - Người dùng có thể dễ dàng truy xuất và in báo cáo dữ liệu dưới dạng Excel với thời gian cụ thể.
 - Tạo được các cảnh báo và gửi E-mail về người dùng để kịp thời xử lý sự cố.
 - Phân quyền và quản lý được người dùng.
- Các hướng phát triển của đề tài được đề xuất như sau:
 - Ở phần kiến trúc của hệ thống ta có thể sử dụng một Module chuyển đổi Modbus RTU sang Modbus TCP/IP sau đó đưa dữ liệu trực tiếp lên nền tảng mà không cần phải qua bước thu thập dữ liệu tại PLC.
 - Hệ thống sẽ được cải thiện tốt hơn nếu ta chọn giải pháp kết nối thông qua bộ chuyển đổi Industrial IoT Gateway thay vì dùng máy tính làm một Server trung gian, điều này giúp cho hệ thống hoạt động tốt ở yêu cầu hoạt động liên tục nhiều ngày hoặc ở những nơi không có mạng Internet hoặc Wifi vì có thể sử dụng Sim 5G.
 - Phát triển xây dựng hệ thống với tải lớn hơn (VD: Động cơ, Lò sấy nhiệt,...) ở nguồn điện 3 pha – 4 dây để có thể thu thập được nhiều dữ liệu về thông số điện năng hơn.
 - Nếu có kiến thức chuyên môn về lập trình phần mềm như Front-end và Back-end thì có thể sử dụng ThingsBoard bản Community Edition để xây dựng một hệ thống trên Server của riêng mình.
 - Ngoài việc giám sát đơn thuần thì ta có thể ứng dụng thêm các cuộc gọi thủ tục từ xa (RPC – Remote Procedure Calls) để điều khiển các thiết bị hoặc có thể kết nối ThingsBoard với một bên thứ ba.
 - Có thể ứng dụng AI vào để phân tích và dự đoán xu hướng tiêu thụ năng lượng.

TÀI LIỆU THAM KHẢO

- [1] EVN, “Biểu giá bán lẻ điện (Theo Quyết định số 1279/QĐ - BCT ngày 09/05/2025 của Bộ Công Thương,)” *Trang thông tin điện tử - Tập đoàn Điện lực Việt Nam*, 2025 [Online]. Xem tại: <https://www.evn.com.vn/d/vi-VN/news/Bieu-gia-ban-le-dien-theo-Quyết-dinh-so-1279QD-BCT-ngay-0952025-cua-Bo-Cong-Thuong-60-28-502668>. [Ngày truy cập: 13/05/2025].
- [2] “Catalog MFM383A / MFM383A-C,” Selec Controls Pvt. Ltd., Mahape, Navi Mumbai, India.
- [3] Minh Quang, “Truyền thông Modbus RTU với PLC Siemens S7 – 1200,” *Minh Quang Electric*, [Online]. Xem tại: <https://dienminhquang.com/s7-1200-modbus-rtu/>. [Ngày truy cập: 08/03/2025].
- [4] Rukham Khan, “SMTP Settings: How to Configure an SMTP Server,” *Email Vendor Selection*, [Online]. Xem tại: <https://www.emailvendorselection.com/smtp-settings/>. [Ngày truy cập: 02/04/2025].
- [5] ThingsBoard, “ThingsBoard Community Edition,” *ThingsBoard Cloud*, [Online]. Xem tại: <https://thingsboard.io/docs/>. [Ngày truy cập: 01/03/2025].
- [6] Ngọc Automation, “Khóa học IoT cho các dòng PLC,” *Youtube*, 26/08/2023, [Video file] Xem tại: <https://hoctudonghoa.com/courses/khoa-hoc-iot-cho-plc/9267/>. [Ngày truy cập: 01/03/2025].
- [7] Mitsubishi Electric, “Search for Software for Energy Saving Supporting Devices Energy Saving Supporting Devices,” *Energy Saving Supporting Devices*, [Online]. Xem tại: <https://www.mitsubishielectric.com/fa/download/software/detailsearch.page?mode=software&kisyu=/ems&shiryoid=1000000186>. [Ngày truy cập: 02/05/2025].
- [8] Schneider Electric, “What's New in the Power Monitoring Expert Demo,” *Power Monitoring Expert*, [Online]. Xem tại: https://www.pmedemo.biz/web/#WhatsNew_EN. [Ngày truy cập: 02/05/2025].