**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**TRƯƠNG HỒNG PHÚC**

**ĐỀ CƯƠNG LUẬN VĂN THẠC SĨ**

**KẾT NỐI NHÀ MÁY VỚI ĐIỆN TOÁN ĐÁM MÂY**

**CONNECTING THE FACTORY WITH CLOUD COMPUTING**

**THẠC SĨ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**TP. HỒ CHÍ MINH, 2019**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**TRƯƠNG HỒNG PHÚC – 1770039**

**LUẬN VĂN TỐT NGHIỆP**

**KẾT NỐI NHÀ MÁY VỚI ĐIỆN TOÁN ĐÁM MÂY**

**(THIẾT KẾ HỆ THỐNG QUAN TRẮC TỰ ĐỘNG MÔI TRƯỜNG NƯỚC NUÔI TÔM)**

**THẠC SĨ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. TRƯƠNG ĐÌNH CHÂU**

**PGS.TS. PHẠM NGỌC TUẤN**

**TP. HỒ CHÍ MINH, 2019**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  **KHOA ĐIỆN – ĐIỆN TỬ**  **BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG** | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày….tháng 01 năm 2019* |

**NHẬN XÉT ĐỀ CƯƠNG LUẬN VĂN**

**CỦA CÁN BỘ HƯỚNG DẪN**

|  |  |  |
| --- | --- | --- |
| **Tên luận văn:** | | |
| **KẾT NỐI NHÀ MÁY VỚI ĐIỆN TOÁN ĐÁM MÂY** | | |
| **Nhóm Sinh viên thực hiện:** | | **Cán bộ hướng dẫn:** |
| TRƯƠNG HỒNG PHÚC | 1770039 | TS. TRƯƠNG ĐÌNH CHÂU  PGS.TS PHẠM NGỌC TUẤN |
|  |  |  |
| **Đánh giá Luận văn**   1. Về cuốn báo cáo:   Số trang Số chương  Số bảng số liệu Số hình vẽ  Số tài liệu tham khảo Sản phẩm  Một số nhận xét về hình thức cuốn báo cáo:   1. Về nội dung luận văn: 2. Về tính ứng dụng: 3. Về thái độ làm việc của sinh viên:   **Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình  **Điểm từng sinh viên:**  TRƯƠNG HỒNG PHÚC**: ………../10** | | |
|  | | |

|  |  |
| --- | --- |
|  | **Người nhận xét**  (Ký tên và ghi rõ họ tên) |

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  **KHOA ĐIỆN – ĐIỆN TỬ**  **BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG** | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày.….tháng 01 năm 2019.* |

**NHẬN XÉT ĐỀ CƯƠNG LUẬN VĂN**

**CỦA CÁN BỘ PHẢN BIỆN**

|  |  |  |
| --- | --- | --- |
| **Tên đề cương luận văn:** | | |
| **KẾT NỐI NHÀ MÁY VỚI ĐIỆN TOÁN ĐÁM MÂY** | | |
| **Nhóm Sinh viên thực hiện:** | | **Cán bộ phản biện:** |
| TRƯƠNG HỒNG PHÚC | 1770039 | ............................................. |
|  |  |  |
| **Đánh giá Đề cương luận văn**   1. Về cuốn báo cáo:   Số trang Số chương  Số bảng số liệu Số hình vẽ  Số tài liệu tham khảo Sản phẩm  Một số nhận xét về hình thức cuốn báo cáo:   1. Về nội dung luận văn: 2. Về tính ứng dụng: 3. Về thái độ làm việc của sinh viên:   **Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình  **Điểm từng sinh viên:**  TRƯƠNG HỒNG PHÚC**: ………../10** | | |

|  |  |
| --- | --- |
|  | **Người nhận xét**  (Ký tên và ghi rõ họ tên) |

**ĐỀ CƯƠNG LUẬN VĂN CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN ĐỀ TÀI:**  **KẾT NỐI NHÀ MÁY VỚI ĐIỆN TOÁN ĐÁM MÂY** | |
| **Cán bộ hướng dẫn: TS. TRƯƠNG ĐÌNH CHÂU**  **PGS.TS PHẠM NGỌC TUẤN** | |
| **Thời gian thực hiện:** Từ ngày 11/02/2019 đến ngày 01/06/2019 | |
| **Sinh viên thực hiện:**  **TRƯƠNG HỒNG PHÚC – 1770039** | |
| **Nội dung đề tài:**  Mục tiêu: Điều khiển, giám sát thông qua internet các nhà máy ở các vị trí khác nhau. Giúp các nhà máy có thể liên lạc, nói chuyện được với nhau.  Phạm vi: Cho các nhà máy nhiều cơ sở cần quản lí tập trung.  Đối tượng: Mô hình các nhà máy thật, nhà máy ảo  Phương pháp thực hiện: Theo chuẩn truyền thông MQTT và công nghệ điện toán đám mây  Kết quả mong đợi: SCADA các nhà máy dựa trên công nghệ điện toán đám mây. | |
| **Xác nhận của Cán bộ hướng dẫn** | TP. HCM, ngày 04 tháng 01 năm 2019  **Sinh viên** |

MỤC LỤC

[Chương 1. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc533027043)

[Chương 2. GIAO THỨC MQTT 6](#_Toc533027044)

[2.1. Cơ bản về giao thức MQTT 6](#_Toc533027045)

[2.1.1. Các thuật ngữ 6](#_Toc533027046)

[2.1.2. Tiêu chuẩn và kiến trúc 7](#_Toc533027047)

[2.1.3. Tính năng 8](#_Toc533027048)

[2.1.4. Gói điều khiển MQTT 9](#_Toc533027049)

[2.2. Thư viện MQTT Client 13](#_Toc533027050)

[2.2.1. Thư viện MQTT Client 13](#_Toc533027051)

[2.2.1.1. Các kiểu dữ liệu mới 13](#_Toc533027052)

[2.2.1.2. Giao diện của khối FB 16](#_Toc533027053)

[2.2.2. Thư viện MQTT\_Publish\_Secure 18](#_Toc533027054)

[2.2.2.1. Các kiểu dữ liệu mới 18](#_Toc533027055)

[2.2.2.1. Giao diện của khối FB 18](#_Toc533027056)

[Chương 3. XÂY DỰNG MQTT BROKER, WEBSERVICE TRÊN VPS 19](#_Toc533027057)

[3.1. VPS 19](#_Toc533027058)

[3.1.1. Giới thiệu 19](#_Toc533027059)

[3.1.2. Hệ điều hành VPS 20](#_Toc533027060)

[3.2. MQTT Broker 20](#_Toc533027061)

[3.3. Xây dựng Server Nodejs, Webservice 20](#_Toc533027062)

[Chương 4. TRUYỀN THÔNG S7 1500 VỚI S7 1200 22](#_Toc533027063)

[4.1. Phần mềm TIA Portal v14 22](#_Toc533027064)

[4.1.1. Giới thiệu 22](#_Toc533027065)

[4.1.2. Tag PLC 22](#_Toc533027066)

[4.1.3. Khối tổ chức OB (Oganization Blocks) 23](#_Toc533027067)

[4.1.4. Hàm chức năng (Function) 23](#_Toc533027068)

[4.1.5. Cấu trúc lập trình 24](#_Toc533027069)

[4.1.6. Giới thiệu tập lệnh của PLC sử dụng trong luận văn 25](#_Toc533027070)

[4.1.6.1. Bit logic 25](#_Toc533027071)

[4.1.6.2. Timer 26](#_Toc533027072)

[4.1.6.3. Bit logic 26](#_Toc533027073)

[4.2. PLC S7-1500/PLC S7-1200 26](#_Toc533027074)

[4.2.1. PLC S7-1500 26](#_Toc533027075)

[4.2.2. PLC S7-1200 27](#_Toc533027076)

[4.2.3. Truyền thông S7 1500 với S7 1200 29](#_Toc533027077)

[4.3. Lập trình TIA Portal cho PLC S7-1500/S7-1200 30](#_Toc533027078)

[4.3.1. Tạo project mới 30](#_Toc533027079)

[4.3.2. Cấu hình cho PLC 31](#_Toc533027080)

[4.3.3. Khai báo các biến, các khối và lập trình 32](#_Toc533027081)

[4.3.3.1. Khai báo các biến 32](#_Toc533027082)

[Chương 5. XÂY DỰNG MÔ HÌNH ẢO BẰNG PHẦN MỀM LABVIEW 33](#_Toc533027083)

[5.1. Labview 33](#_Toc533027084)

[5.1.1. Giới thiệu 33](#_Toc533027085)

[5.1.2. Giao tiếp Labview PLC 33](#_Toc533027086)

[5.2. PLC SIM Advance 2.0 33](#_Toc533027087)

[Chương 6. PHÁT TRIỂN ỨNG DỤNG PHÍA CLIENT 35](#_Toc533027088)

[6.1. Phát triển ứng dụng web 35](#_Toc533027089)

[6.1.1. Giới thiệu 35](#_Toc533027090)

[6.2. Phát triển ứng dụng Androi 35](#_Toc533027091)

[6.3. Phát triển ứng dụng IOS 36](#_Toc533027092)

[Chương 7. KẾT QUẢ DỰ KIẾN KẾ HOẠCH THỰC HIỆN 37](#_Toc533027093)

[7.1. Kết quả thực hiện 37](#_Toc533027094)

[7.1.1. Mô hình thực hiện 37](#_Toc533027095)

[7.1.2. Kết quả mong muốn 37](#_Toc533027096)

[7.2. Nội dung luận văn 37](#_Toc533027097)

[7.3. Thời gian thực hiện 38](#_Toc533027098)

DANH MỤC HÌNH VẼ

[Hình 1.1 Nội dung của luận văn 4](#_Toc533027177)

[Hình 2.1 Cấu trúc MQTT 8](#_Toc533027178)

[Hình 2.2 Cấu trúc gói SUBSCRIBE 12](#_Toc533027179)

[Hình 2.3 Cấu trúc gói SUBACK 12](#_Toc533027180)

[Hình 2.4 Cấu trúc gói UNSUBSCRIBE 13](#_Toc533027181)

[Hình 2.5 Khối chức năng LMqtt\_Publisher 16](#_Toc533027182)

[Hình 2.6 Khối chức năng LMqtt\_Publisher 18](#_Toc533027183)

[Hình 4.1 PLC Tags trong TIA PORTAL 22](#_Toc533027184)

[Hình 4.2 Cấu trúc lập trình 24](#_Toc533027185)

[Hình 4.3 PLC S7-1500 27](#_Toc533027186)

[Hình 4.4 PLC S7-1200 28](#_Toc533027187)

[Hình 4.5 Biểu tượng TIA Portal trên Desktop 30](#_Toc533027188)

[Hình 4.6 Giao diện tạo dự án mới của TIA Portal 30](#_Toc533027189)

[Hình 4.7 Điền cáo thông tin của project 31](#_Toc533027190)

[Hình 4.8 Mở dự án Tia Portal 31](#_Toc533027191)

[Hình 5.1 HMI Tags 34](#_Toc533027192)

DANH MỤC BẢNG

[*Bảng 2.1 Kiểu dữ liệu “typeTcpConnParam”* 14](#_Toc533027193)

[*Bảng 4.1 Tính năng các loại CPU* 28](#_Toc533027194)

DANH MỤC TỪ VIẾT TẮT

PLC Programble Logic Control

SCADA Supervisory Control and Data Acquisition

PWM Pulse Width Modulation

HTML HyperText Markup Language

OB Organization Blocks

WinCC Windows Control Center

HMI Human Machine Interface

VPS Virtual Private Server

MQTT Message Queuing Telemetry Transport

WWW World Wide Web

TÓM TẮT ĐỀ CƯƠNG LUẬN VĂN

KẾT NỐI NHÀ MÁY VỚI ĐIỆN TOÁN ĐÁM MÂY

**Yêu cầu**

Kết nối nhà máy dựa trên công nghệ điện toán đám mây.

Quản lí, giám sát, theo dõi dữ liệu các nhà máy tập trung dựa vào điện toán đám mây.

Quản lí, giám sát, các nhà máy thông qua internet (Web, app)

**Thời gian thực hiện đề tài (Từ ngày 11/02/2019 đến 01/06/2019)**

* Tuần 1: Tìm hiểu giao thức MQTT
* Tuần 2: Truyền thông S7 1500 với MQTT Broker
* Tuần 3: Tìm hiểu VPS, thiết lập máy tính ảo trên cloud
* Tuần 4: Xây dựng MQTT Broker, Webservice trên VPS
* Tuần 5: Truyền thông S7 1500 với S7 1200
* Tuần 6: Điều khiển thiết bị chấp hành (Motor, Pump,,… )
* Tuần 7: Xây dựng server trên private cloud (Nodejs)
* Tuần 8: Xây dựng ứng dụng cho Androi
* Tuần 9: Xây dựng ứng dụng cho IOS
* Tuần 10: Tìm hiểu Labview
* Tuần 11: Điều khiển mô hình bằng mô phỏng Labview
* Tuần 12: Viết báo cáo, và thuyết trình
* Tuần 13: Chỉnh sửa và báo cáo.

# GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

* 1. **Giới thiệu**

**Số hoá trong công nghiệp**

Số hoá trong công nghiệp là một quá trình tất yếu trong nền sản xuất hiện đại và có tác động to lớn đến kinh tế và đời sống xã hôi. “Internet of Things” hay “IoT” chính là điều kiện tiên quyết cho quá trình số hoá, gắn liền với một trong những xu hướng lớn nhất hiện nay trong lĩnh vực công nghiệp: Gia tăng các thiết bị, máy móc và sản phẩm gắn liền với tự động và mạng.

PLC là thiết bị điều khiển phổ biến trong các nhà máy hiện nay tại Việt Nam cũng như trên thế giới. Việc kết nối PLC với mạng internet, mà cụ thể là dịch vụ điện toán đám mây là con đường nhanh nhất và khả thi nhất cho quá trình số hoá diễn ra. Đối với các dòng PLC của hãng Siemens, có hai cách để kết nối với dịch vụ điện toán đám mây:

o Kết nối PLC S7-1500 với Mindsphere (nền tảng điện toán đám mây của Siemens, mới được ra mắt chính thức tại Việt Nam vào năm 2018).

o Kết nối PLC S7-300, S7-1200 và S7-1500 với dịch vụ điện toán đám mây thông qua giao thức “Message Queue Telemetry Transport” (viết tắt là “MQTT”).

MQTT là giao thức truyền thông được sử dụng phổ biến trong lĩnh vực IoT. Đây là giao thức hoàn toàn miễn phí, có cộng đồng hỗ trợ lớn mạnh và đang là xu hướng mới trong lĩnh vực tự động thế giới.

**Tổng quan về giao thức MQTT**

MQTT là giao thức đăng kí và gửi được tích hợp đơn giản ở cấp TCP/IP. Giao thức này phù hợp cho việc gửi và nhận tin nhắn giữa các thiết bị chức năng thấp và những hệ thống mạng không tin cậy, băng thông thấp, độ trễ cao. MQTT là giao thức chính được Facebook sử dụng trong Facebook Messenger. Với đặc tính trên, MQTT đã và đang đóng vai trò quan trong trong IoT và giao tiếp Machine- to- Machine (M2M).

**SCADA với điện toán đám mây**

Nhiều hệ thống nước và vệ sinh nông thôn nhỏ và nông thôn phải đối mặt với những khó khăn về ngân sách và những sự cắt giảm khiến khó có thể đầu tư ban đầu vào một hệ thống SCADA giúp họ điều hành hoạt động của mình hiệu quả hơn. Các cơ sở nhỏ hơn cũng không thể có đủ nhân viên kỹ thuật chuyên dụng để quản lý các hệ thống SCADA của họ.

Điều này làm cho SCADA trên đám mây trở thành một giải pháp rất hấp dẫn đối với ngành nước và nước thải. Nó sẽ hạn chế nhu cầu mua phần cứng, phần mềm và bảo trì.

Các cơ sở xử lý nước nhỏ hơn thường phải vật lộn với việc nâng cấp hệ thống tự động hóa.

SCADA dựa trên đám mây cũng có thể cung cấp sự nhanh nhẹn khi áp dụng các ứng dụng mới và nâng cấp, đồng thời tăng cường độ tin cậy thông qua kết nối Internet nhiều lần của nó theo cách hiệu quả hơn nhiều chi phí, thay vì dựa vào một Nhà cung cấp dịch vụ Internet (ISP).

**Xây dựng SCADA với điện toán đám mây và kết nối nhà máy**

SCADA dựa trên đám mây đã cho thấy làm giảm đáng kể chi phí liên quan đến một hệ thống SCADA truyền thống. Bằng cách di chuyển sang đám mây, các chi phí liên quan đến thay thế phần cứng đã lỗi thời như máy tính chạy Windows 7, windows 10 sẽ biến mất vì ứng dụng chạy trong môi trường ảo. Nhà cung cấp đám mây cập nhật phần cứng, và quá trình này là vô hình đối với người dùng.

Chi phí phần cứng liên quan cũng có thể được thanh toán hàng tháng thay vì một chi phí trả trước lớn. Người dùng trả tiền để tăng dung lượng lưu trữ khi họ cần. Ví dụ: họ không cần phải mua một máy chủ lớn với giấy phép SQL server 30.000 USD để có thể sử dụng lưu trữ dữ liệu trong 10 năm tới. Họ có thể thay thế mở rộng lưu trữ dữ liệu từng bước, mà không phải mua thêm phần cứng và phần mềm.

Việc sử dụng cơ sở hạ tầng điện toán đám mây trước đây thường cho phép triển khai và nâng cấp theo ngày hơn là hàng tháng. Các nguồn tài nguyên máy tính có thể được bổ sung nhanh chóng khi các dự án SCADA bổ sung được đưa lên mạng, hoặc dễ bị rơi nếu giải pháp không phù hợp hoặc không hoạt động. Chi phí cấm của phần cứng và thay đổi nền tảng phần cứng được loại bỏ, không còn khóa người dùng vào một giải pháp cụ thể.

Nhiều ứng dụng SCADA của nước và nước thải dựa vào việc giám sát từ xa các thiết bị quan trọng và xử lý thông qua trình duyệt web, điện thoại thông minh và các thiết bị di động khác. Sử dụng một giải pháp lưu trữ tự do với quyền truy cập được cung cấp bởi một ISP duy nhất có thể gây ra vấn đề nếu nhà cung cấp dịch vụ gặp phải sự gián đoạn. Điện toán đám mây cung cấp nhiều kết nối Internet, cung cấp độ tin cậy cao hơn và làm như vậy một cách hiệu quả về chi phí.

* 1. **Yêu cầu và nội dung**
     1. **Yêu cầu**
* Xây dựng mô hình nhà máy bằng thiết bị thật PLC S7 1500, S7 1200.
* Kết nối được các mô hình nhà máy dựa trên giao thức truyền thông MQTT
* Mô phỏng mô hình nhà máy (PLC, thiết bị chấp hành) bằng Labview.
* Điều khiển và giám sát nhà máy thông qua mạng internet (PC, Laptop).
* Điều khiển và giám sát nhà máy thông qua điện thoại (Androi, IOS).
  + 1. **Nội dung của đề cương luận văn**



Hình 1.1 Nội dung của luận văn

Mô hình gồm 3 nhà máy ở 3 vị trí khác nhau. Trong đó có 2 mô hình nhà máy thật, 1 mô hình nhà máy mô phỏng.

Các nhà máy này có thể giao tiếp, nói chuyện được với nhau sử dụng giao thức MQTT.

Tất cả các dữ liệu thu thập được từ các nhà máy được tập trung bởi một server trên cloud. Các dữ liệu đó được lưu vào database.

Các client (smartphone hệ điều hành IOS, Androi, Laptop, PC) điều khiển và giám sát (SCADA) được các nhà máy này thông qua các phần mềm ở phía client phù hợp với từng thiết bị.

Các thành phần sử dụng trong phần mềm này gồm:

* Thiết bị PLC S7 1500, S7 1200, Motor, sensor, smartphone …
* Phần mềm sử dụng: TIA Portal v14 SP1, Labview
* Phần mềm server: Nodejs
* Giao thức: MQTT, Modbus TCP/IP

# GIAO THỨC MQTT

## Cơ bản về giao thức MQTT

### Các thuật ngữ

* **MQTT message**

Một tin nhắn với giao thức MQTT bao gồm các phần sau:

* + Chủ đề xác định (“Topic”).
  + Cấp độ chất lượng dịch vụ (“Quality of Service”).
  + Nội dung tin nhắn.
* **MQTT client**

MQTT client là một chương trình hoặc một thiết bị sử dụng giao thức MQTT. Client luôn chủ động thiết lập kết nối đến broker. Một client có thể thực hiện các chức năng sau:

* + Gửi tin nhắn với chủ đề xác định (“Topic”) đến broker (cơ chế gửi – Publish).
  + Đăng kí tin nhắn của một chủ đề xác định tại broker (cơ chế đăng kí – Subscriber).
  + Tự huỷ đăng kí một chủ đề đã đăng kí trước đó.
  + Ngắt kết nối với broker.
* **MQTT broker**

MQTT broker là thành phần trung tâm của giao thức MQTT, có thể là một chương trình hoặc thiết bị. Broker đóng với trò là người trung gian giữa MQTT client gửi và MQTT client đăng kí. MQTT broker quản lí những chủ đề (“Topic”) bao gồm các tin nhắn chứa trong đó và điều chỉnh quyền truy cập chủ đề. Những chức năng cơ bản của broker như sau:

o Chấp nhận hoặc từ chối kết nối từ client.

o Nhận tin nhắn từ MQTT client.

o Chỉnh sửa yêu cầu đăng kí từ MQTT client.

o Chuyển tiếp tin nhắn đến các client đã đăng kí trước đó.

* **Chủ đề (“Topic”)**

Các tin nhắn MQTT được tổ chức theo các chủ đề. Muốn nhận được tin nhắn, các client phải đăng kí đúng chủ đề (cơ chế subscriber). Người gửi tin nhắn (cơ chế publisher) có trách nhiệm định nghĩa tên chủ đề và nội dung khi gửi tin nhắn. Broker có trách nhiệm phân phát tin nhắn đến các subscriber theo đúng chủ đề đã đăng kí.

### Tiêu chuẩn và kiến trúc

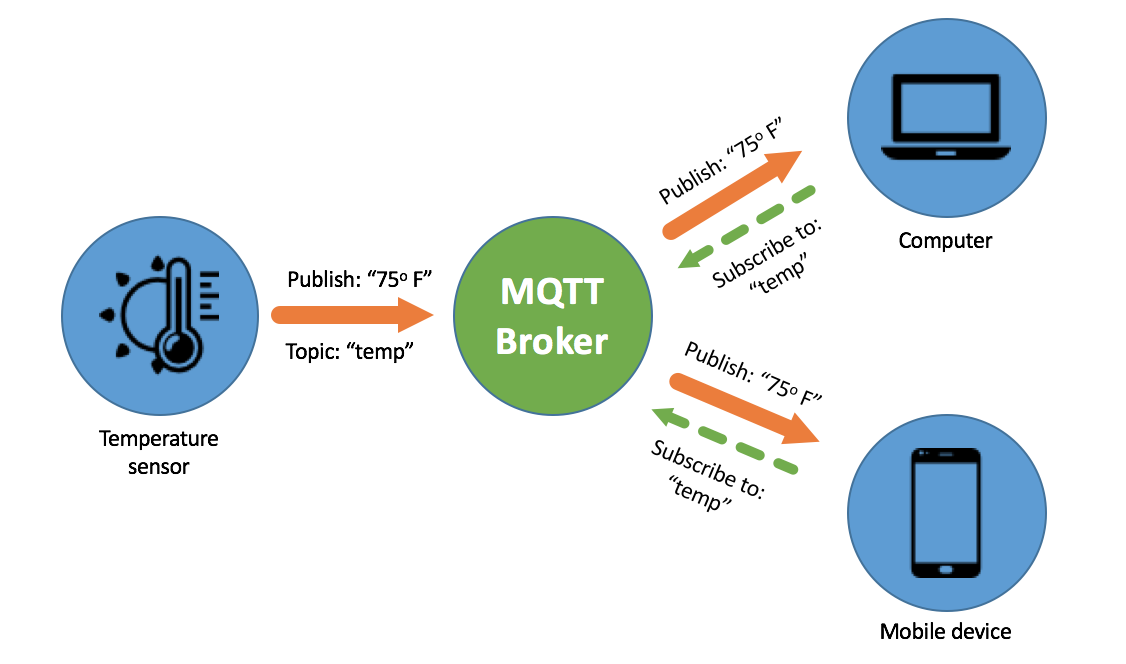
* **Tiêu chuẩn ISO**

MQTT vận hành trên các cổng (port) truy cập khác nhau, phụ thuộc vào giao thức bảo mật được sử dụng. Các cổng được cung cấp là:

* 1883: MQTT, không mã hoá.
* 8883: MQTT, mã hoá.
* 8884: MQTT, mã hoá, yêu cầu chứng chỉ client.
* 8080: MQTT thông qua WebSockets, không mã hoá.
* 8081: MQTT thông qua WebSockets, mã hoá.
* **Kiến trúc**

MQTT là giao thức gửi và đăng kí. Giao thức này tách riêng client gửi tin nhắn (publisher) và client nhận tin nhắn (subscriber). Điều này có nghĩa là “publisher” sẽ không biết gì về sự tồn tại của “subscriber” và ngược lại.

Thành phần thứ ba trong kiến trúc MQTT là MQTT broker. MQTT broker đóng vai trò là người trung gian giữa “publisher” và “subscriber”. Broker điều khiển việc kết nối của của client.



Hình 2.1 Cấu trúc MQTT

### Tính năng

MQTT cung cấp các tính năng hữu ích sau:

* **Quality of Service (QoS)**

MQTT cung cấp ba cấp độ đảm bảo chất lượng truyền tin nhắn:

* QoS “0”: Cấp độ thấp nhất là 0, không đảm bảo tin nhắn sẽ đến.
* QoS “1”: Cấp độ 1 đảm bảo rằng tin nhắn kết thúc nằm trong hàng đợi chủ đề ít nhất một lần. Broker xác nhận đã nhận được tin nhắn.
* QoS “2”: Cấp độ cao nhất, bằng cách bắt tay nhiều lần với client, broker đảm bảo rằng tin nhắn được gửi chính xác một lần.
* **Last will**

MQTT hỗ trợ tính năng “Last Will and Testament”. Tính năng này được sử dụng để thông báo những client khác nếu kết nối đến một client bị ngắt ngẫu nhiên.

Mỗi client có thể chỉ định “last will” của riêng mình trong khi kết nối đến broker. “Last will” có cấu trúc giống như một tin nhắn MQTT thông thường, bao gồm chủ đề, QoS và nội dung. Broker lưu trữ các “last will”. Ngay khi broker nhận ra kết nối với client bị ngắt đột ngột, broker sẽ gửi “last will” như một tin nhắn MQTT đến tất cả subscriber đã đăng kí chủ đề đó để các subscriber được biết.

* **KeepAlive:**

Tính năng KeepAlive đảm bảo rằng kết nối vẫn được mở và client được kết nối đến broker. Với KeepAlive, client định nghĩa một khoảng thời gian và thông báo với broker trong quá trình thiết lập kết nối. Đây là khoảng thời gian lớn nhất client và broker có thể duy trì kết nối mà không cần tương tác. Nếu vượt quá thời gian đó, broker sẽ tự ngắt kết nối với client.

Nếu client liên tục gửi tin nhắn đến broker với chu kì nhỏ hơn thời gian KeepAlive thì client không cần làm gì để duy trì kết nối. Tuy nhiên, nếu client không gửi bất kì tin nhắn nào trong khoảng thời gian KeepAlive thì phải ping đến broker trước khi thời gian hết hạn.

Khi một tin nhắn hoặc lệnh ping được gửi đến broker, đồng hồ đếm KeepAlive sẽ được đặt lại.

* **Message persistence:**

Nếu kết nối đến client bị gián đoạn, broker có thể lưu trữ các tin nhắn mới để phân phát cho client này sau.

* **Retained messages**

Lần đầu tiên một client đăng kí một topic, nó thường chỉ nhận được tin nhắn sau khi một client khác gửi tin nhắn đến chủ đề được đăng kí. Với chức năng “Retained messages”, subscriber sẽ lập tức nhận được giá trị cuối cùng được gửi đến chủ đề đó, ngay khi nó đăng kí chủ đề với broker.

### Gói điều khiển MQTT

Hầu hết các gói điều khiển MQTT làm việc theo thủ tục bắt tay. MQTT client luôn chủ động yêu cầu broker và broker xác nhận yêu cầu từ client.

Gói điều khiển MQTT có cấu trúc cố định. Cấu trúc của một gói tin được thể hiện trong sơ đồ

|  |
| --- |
| **Fixed header**  Bắt buộc cho tất cả các gói tin |
| **Variable header**  Bắt buộc cho một vài gói tin |
| **Payload**  Bắt buộc cho tất cả các gói tin |

Trường “Fixed header” luôn chứa các thành phần sau:

* Số nhận dạng cho loại gói điều khiển MQTT.
* Khu vực cho các cờ, nếu không có cờ nào được sử dụng thì các bit sẽ được đánh dấu là dự trữ.
* Số byte của các thành phần sau “Fixed header”.

Trường “Variable header” chỉ bắt buộc cho một vài gói điều khiển. Nội dung của trường này phụ thuộc vào loại gói điều khiển.

Trường “Payload” bắt buộc cho tất cả các gói điều khiển. Nội dung của trường này phụ thuộc vào loại gói điều khiển. Với mỗi loại gói điều khiển sẽ có các quy tắc về nội dung và thứ tự mà trường “Payload” được điền.

**Một số gói điều khiển thông dụng:**

* **MQTT Connection (gói kết nối MQTT)**

Gói này luôn được thực hiện giữa client và broker. MQTT không cho phép các client kết nối trực tiếp với nhau (client – client). Kết nối sẽ được khởi tạo bởi client ngay khi nó gửi gói “CONNECT” đến broker. Nếu broker đồng ý sẽ gửi gói “CONNACK” và mã trạng thái.

Broker sẽ lập tức đóng kết nối trong các trường hợp sau:

* Gói “CONNECT” bị lỗi.
* Cấu trúc của gói “CONNECT” không thoả yêu cầu.
* Kết nối quá lâu (time-out).

Gói “CONNECT” có chứa khu vực dành cho các cờ trong trường “Variable header”. Byte này chứa các tham số xác định hành vi của kết nối MQTT. Ngoài ra nó cũng cho thấy các thành phần tuỳ chọn có được sử dụng trong trường “Payload” hay không (tên đăng nhập, mật khẩu, last will,…).

Trong trường “Payload” có các thành phần bắt buộc như sau:

* “ClientID”: Được broker dùng để phân biệt các client với nhau. Nếu để trống trường này sẽ không thể kết nối vào broker được.
* “CleanSession”: Xác định loại kết nối.
* KeepAlive: Xác định thời gian mà client bắt buộc phải tương tác với broker (bằng cách gửi tin nhắn hoặc thực hiện lệnh PING). Nếu client không tương tác với broker trong khoảng thời gian này thì broker sẽ ngắt kết nối với client.
* **Cơ chế gửi**

Client có thể gửi tin nhắn đến broker ngay sau khi kết nối thành công vào broker. Việc gửi tin nhắn được thực hiện thông qua gói “PUBLISH”. Do MQTT quản lí các tin nhắn theo chủ đề nên trong gói “PUBLISH” bắt buộc phải có chủ đề. Chủ đề nằm ở trường “Variable header” và nội dung tin nhắn sẽ nằm ở trường “Payload”.

Cơ chế đẩy sẽ kết thúc hoặc sẽ gửi gói xác nhận tuỳ thuộc vào thiết lập QoS:

* Nếu QoS là “0”, quá trình gửi kết thúc mà không có bất cứ thông tin gì xác nhận.
* Nếu QoS là “1”, broker xác nhận gói “PUBLISH” bằng gói “PUBACK”.
* Nếu QoS là “2”, broker xác nhận gói “PUBLISH” bằng gói “PUBREC”. Client sẽ xác nhận lại bằng gói “PUBREL”. Sau đó broker hoàn tất quá trình bắt tay bằng cách gửi gói “PUBCOM”.
* **Cơ chế PING MQTT**

Nếu chức năng KeepAlive được kích hoạt (thời gian KeepAlive lớn hơn 0), client phải gửi đến broker ít nhất một tin nhắn trong khoảng thời gian này, nếu không broker sẽ ngắt kết nối với client.

Để tránh việc ngắt kết nối, client phải gửi lệnh “PING” đến broker trước khi thời gian hết hạn. Việc này được client thực hiện bằng gói “PINGREQ” và sẽ được broker xác nhận bằng gói “PINGRESP”.

Lưu ý: Với thư viện MQTT được sử dụng trong ví dụ này, nếu chức năng KeepAlive được bật thì thời gian KeepAlive phải lớn hơn 2 giây.

* **Cơ chế đăng kí**

Để nhận được tin nhắn ở chủ đề quan tâm, client gửi gói “SUBSCRIBE” đến broker. Cấu trúc của gói tin này bao gồm chỉ số gói và danh sách các chủ đề đăng kí.

|  |
| --- |
| PacketID  QoS1  Topic1  QoS2  Topic2  …. |

Hình 2.2 Cấu trúc gói SUBSCRIBE

Broker nhận được yêu cầu đăng kí sẽ xác nhận bằng gói “SUBACK”. Gói tin này bao gồm chỉ số gói (trùng với chỉ số của gói đăng kí” và danh sách các code trạng thái.

|  |
| --- |
| PacketID  returnCode 1  returnCode 2  … |

Hình 2.3 Cấu trúc gói SUBACK

* **Cơ chế huỷ đăng kí**

Để ngừng nhận tin nhắn từ một chủ đề đã đăng kí trước đó, client sẽ gửi gói “UNSUBSCRIBE” đến broker. Cấu trúc của gói tin này tương tự như gói “SUBSCRIBE”, cũng bao gồm chỉ số gói và danh sách các chủ đề muốn huỷ đăng kí.

|  |
| --- |
| PacketID  Topic1  Topic2  … |

Hình 2.4 Cấu trúc gói UNSUBSCRIBE

Để xác nhận yêu cầu huỷ đăng kí, broker gửi gói “UNSUBACK” đến client. Gói tin này chỉ có chỉ số gói (trùng với chỉ số của gói “UNSUBSCRIBE”). Sau khi nhận gói “UNSUBACK”, client sẽ xác nhận được rằng các gói đăng kí đã bị huỷ.

* **MQTT disconnection (ngắt kết nối MQTT)**

Client có thể chủ động ngắt kết nối đến broker bằng cách gửi gói “DISCONNECT”. Broker sau đó sẽ xoá tất cả thông tin “Last Will and Testament” và sẽ ngừng gửi tin nhắn của những chủ đề mà client đã đăng kí.

## Thư viện MQTT Client

### Thư viện MQTT Client

Đây là thư viện chính thức của Siemens, có chức năng gửi tin nhắn đến broker không bảo mật theo giao thức MQTT.

#### Các kiểu dữ liệu mới

Để cấu trúc dữ liệu một cách thuận tiện, các kiểu dữ liệu mới được tạo ra và sử dụng trong chương trình.

* “typeTcpConnParam”.
* “typeMqttParam”, được chia thành:
* “typeMqttConnectFlags”.
* “typeMqttPublishFlags”.

**Kiểu dữ liệu “typeTcpConnParam”**

Kiểu dữ liệu này lưu trữ tất cả thông tin cần thiết để thiết lập kết nối TCP.

Bảng bên dưới mô tả các tham số của SIMATIC S7-1500/ SIMATIC S7-1200:

*Bảng 2.1 Kiểu dữ liệu “typeTcpConnParam”*

|  |  |  |
| --- | --- | --- |
| Tham số | Kiểu dữ liệu | Ý nghĩa |
| hwIdentifier | HW\_ANY | HW ID của giao tiếp PROFINET của CPU |
| connectionID | CONN\_OUC | ID của kết nối TCP |
| ipAdressBroker | Array[0..3] of BYTE | Địa chỉ IP của broker. Ví dụ địa chỉ IP là  192.168.0.1  ipAddressBroker[0]: 192  ipAddressBroker[1]: 168  ipAddressBroker[2]: 0  ipAddressBroker[3]: 1 |
| localPort | UINT | Cổng nội bộ của CPU |
| mqttPort | UINT | Cổng kết nối của MQTT broker |

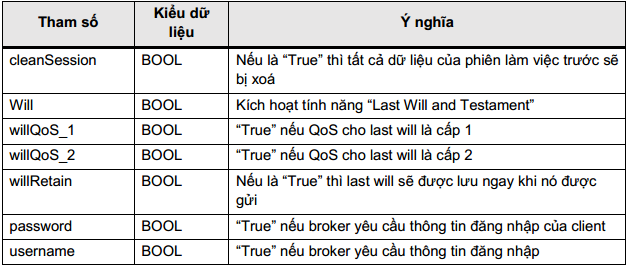
**Kiểu dữ liệu “typeMqttParam”**

Kiểu dữ liệu này bao gồm tất cả các thông tin về MQTT.

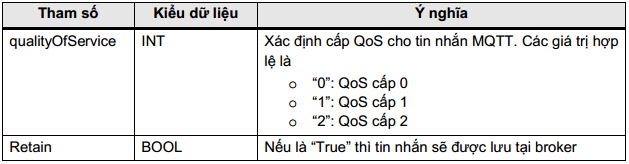
* Các cờ để kết nối.
* Các cờ để gửi tin nhắn.
* Thông tin đăng nhập broker.
* Chủ đề.
* Nội dụng tin nhắn.

Để thuận tiện cho việc quản lí, kiểu dữ liệu này được tách ra làm hai kiểu dữ liệu con:

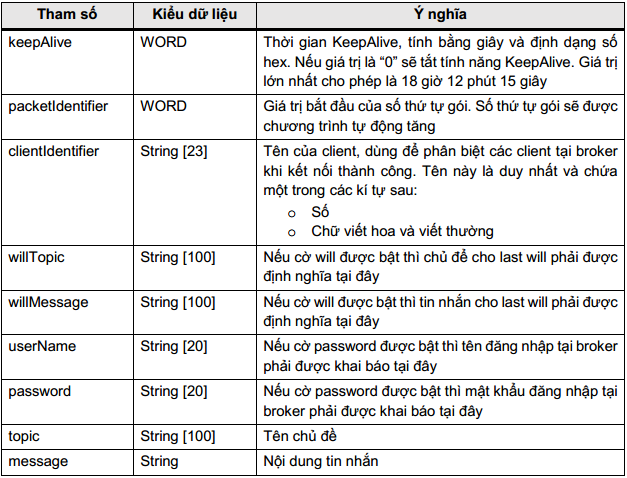
“typeMqttConnectFlags” chứa các cờ để kết nối với broker.



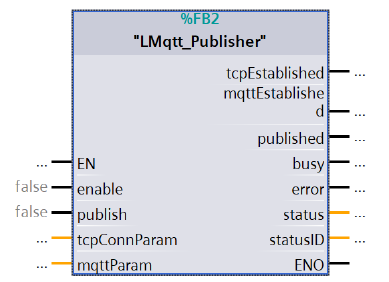
“typeMqttPublishFlags” chứa các cờ cho tin nhắn MQTT



Các tham số khác của kiểu dữ liệu “typeMqttParam”



#### Giao diện của khối FB



Hình 2.5 Khối chức năng LMqtt\_Publisher

Khối chức năng “LMqtt\_Publisher” được thiết kế cho “optimized block access”.

**Tham số đầu vào**

|  |  |  |
| --- | --- | --- |
| **Tham số** | **Kiểu dữ liệu** | **Chức năng** |
| enable | BOOL | Kích hoạt khối chức năng bằng cạnh lên. Khối chức năng sẽ duy trì trạng thái kích hoạt nếu enable giữ trạng thái “true”.  Cạnh xuống ở chân này sẽ ngắt kết nối TCP và MQTT. |
| publish | BOOL | Gửi tin nhắn đến broker khi có cạnh lên. |
| tcpConnParam | “typeTcpConnParam” | Chứa các thông tin kết nối TCP |
| mqttParam | “typeMqttParam” | Chứa các thông tin kết nối MQTT và thông tin tin nhắn. |

**Tham số ngõ ra**

|  |  |  |
| --- | --- | --- |
| **Tham số** | **Kiểu dữ liệu** | **Chức năng** |
| tcpConnected | BOOL | “True” nếu kết nối TCP được thiết lập |
| mqttConnected | BOOL | “True” nếu kết nối MQTT được thiết lập |
| publish | BOOL | “True” nếu tinh nhắn đến broker thành công và chỉ “True” trong một chu kì |
| busy | BOOL | “True” khi một tin nhắn hoặc lệnh ping đang được gửi đến broker |
| error | BOOL | “True” nếu có lỗi |
| statusID | INT | Trạng thái phát sinh lỗi |
| status | DWORD | Thông báo lỗi |

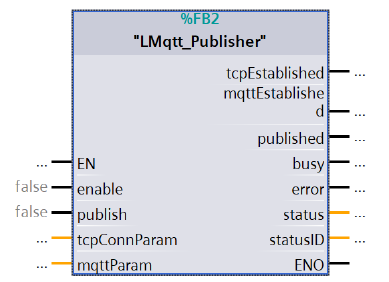
### Thư viện MQTT\_Publish\_Secure

#### Các kiểu dữ liệu mới

Hoàn toàn tương tự phần 2.2.1.1, riêng kiểu “typeTcpConnParam” có thêm một vài tham số mới

|  |  |  |
| --- | --- | --- |
| **Tham số** | **Kiểu dữ liệu** | **Ý nghĩa** |
| activateSecureConn | BOOL | “True” nếu giao tiếp được bảo mật thông qua TLS |
| validateSubjectAlternateNameOfServer | WORD | Bật bit 0 để TLC client xác thực tên thay thế của chứng chỉ. Bit 1 đến 15 dự trữ.  Chỉ liên quan nếu “activateSecureConn” được bật |
| IdTlsServerCertificate | UDINT | Chỉ số của chứng chỉ được dùng để xác thực TLS server. Nếu là “0” thì TLS client sẽ sử dụng tất cả chứng chỉ đang có trong bộ nhớ để xác thực server.  Chỉ liên quan nếu “activateSecureConn” được bật |
| idTlsClientCertificate | UDINT | ID của chứng chỉ để tự xác nhận client với server.  Chỉ liên quan nếu “activateSecureConn” được bật và TLS server yêu cầu xác thực client. |

#### Giao diện của khối FB



Hình 2.6 Khối chức năng LMqtt\_Publisher

# XÂY DỰNG MQTT BROKER, WEBSERVICE TRÊN VPS

## VPS

### Giới thiệu

VPS (Virtual Private Server) là dạng máy chủ ảo được tạo ra bằng phương pháp phân chia một máy chủ vật lý thành nhiều máy chủ khác nhau có tính năng tương tự như máy chủ riêng (dedicated server), chạy dưới dạng chia sẻ tài nguyên từ máy chủ vật lý ban đầu đó. Mỗi VPS là một hệ thống hoàn toàn riêng biệt, có một phần CPU riêng, dung lượng RAM riêng, dung lượng ổ HDD riêng, địa chỉ IP riêng và hệ điều hành riêng, người dùng có toàn quyền quản lý root và có thể restart lại hệ thống bất cứ lúc nào.

**Ưu điểm của VPS?**

* Dễ dàng tùy biến nguồn tài nguyên, trong mức giới hạn của máy chủ vật lý cho phép.
* Từ 1 máy chủ vật lý, có thể tạo ra nhiều VPS. Tiết kiệm được tiền đầu tư phần cứng, tiền điện vận hành máy chủ, không gian lắp đặt…
* Do nhiều VPS có thể nằm tập trung trên 1 hệ thống máy chủ. Việc kiểm tra vận hành sẽ dễ dàng hơn.

**Nhược điểm của VPS?**

* Hoạt động của VPS bị ảnh hưởng bởi hoạt động và độ ổn định của máy chủ vật lý tạo ra VPS.
* Việc sử dụng chung máy chủ vật lý khiến VPS của bạn bị phụ thuộc.
* Tốn thời gian và chi phí để nâng cấp tài nguyên và cũng không thể mở rộng nhiều.
* Cách thức vận hành và năng suất hoạt động của VPS không đạt được hiệu quả như mong muốn.

### Hệ điều hành VPS

Do chi phí hạn chế, trong luận văn sử dụng hệ điều hành Linux. Đây là hệ điều hành miễn phí và được sử dụng rộng rãi trên môi trường cloud, internet

## MQTT Broker

MQTT broker là thành phần trung tâm của giao thức MQTT, có thể là một chương trình hoặc thiết bị. Broker đóng với trò là người trung gian giữa MQTT client gửi và MQTT client đăng kí. MQTT broker quản lí những chủ đề (“Topic”) bao gồm các tin nhắn chứa trong đó và điều chỉnh quyền truy cập chủ đề. Những chức năng cơ bản của broker như sau:

o Chấp nhận hoặc từ chối kết nối từ client.

o Nhận tin nhắn từ MQTT client.

o Chỉnh sửa yêu cầu đăng kí từ MQTT client.

o Chuyển tiếp tin nhắn đến các client đã đăng kí trước đó.

Ở luận văn này sử dụng MQTT Broker Mosquitto trên VPS.

## Xây dựng Server Nodejs, Webservice

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Nodejs được xây dựng và phát triển từ năm 2009. Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực. Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

.

Những ứng dụng có thể và nên viết bằng Nodejs:

* Websocket server: Các máy chủ web socket
* Fast File Upload Client: là các chương trình upload file tốc độ cao.
* Ad Server: Các máy chủ quảng cáo.
* Cloud Services: Các dịch vụ đám mây.
* RESTful API: đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
* Any Real-time Data Application: bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.

# TRUYỀN THÔNG S7 1500 VỚI S7 1200

## Phần mềm TIA Portal v14

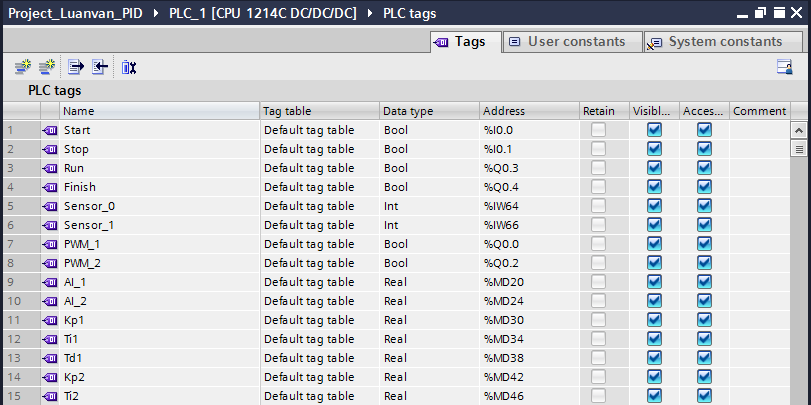
### Giới thiệu

TIA Portal là phần mềm lập trình điều khiển trực quan, hiệu quả và xác thực, giúp người sử dụng thiết kế toàn bộ chương trình tự động hóa một cách tối ưu chỉ trong một giao diện phần mềm duy nhất.

### Tag PLC

Tag các biến số bên trong hệ điều hành thiết bị, dùng để làm các biến trung gian cho quá trình tính toán, các biến số quá trình trong các thiết bị trên mạng điều khiển: trong PLC, trong thiết bị đo lường thông minh, trong các thiết bị nhúng nà controller khác.

Phạm vi ứng dụng: giá trị Tag có thể được sử dụng mọi khối chức năng trong PLC.



Hình 4.1 PLC Tags trong TIA PORTAL

Một số thuộc tính cơ bản của Tag trong PLC

* Name: chỉ được khai báo và sử dụng một lần trên CPU
* Data type: kiểu dữ liệu chỉ định cho các tag
* Address: địa chỉ của tag
* Retain: khai báo của tag sẽ được lưu trữ lại
* Comment: Ghi chú miêu tả của tag
* Nhóm tag: tạo nhóm tag bằng cách chọn add new tag table

### Khối tổ chức OB (Oganization Blocks)

OB (Organization Blocks) là giao diện giữa hoạt động hệ thống và chương trình người dùng. Chúng được gọi ra bởi hệ thống hoạt động, và điều khiển theo quá trình

* + Xử lý chương trình theo quá trình
  + Báo động – kiểm soát xử lý chương trình
  + Xử lý lỗi

Startup OB, Cycle OB, Timing Error OB và Diagnosis OB: có thể chèn và lập trình các khối này trong các project. Không cần phải gán các thông số cho chúng và cũng không cần gọi chúng trong chương trình chính.

Process Alarm OB và Time Interrupt OB: Các khối OB này phải được tham số hoá khi đưa vào chương trình. Ngoài ra, quá trình báo động OB có thể được gán cho một sự kiện tại thời gian thực hiện bằng cách sử dụng các lệnh ATTACH, hoặc tách biệt với lệnh DETACH

### Hàm chức năng (Function)

Funtions (FCs) là các khối mã không cần bộ nhớ. Dữ liệu của các biến tạm thời bị mất sau khi FC được xử lý. Các khối dữ liệu toàn cục có thể được sử dụng để lưu trữ dữ liệu FC.

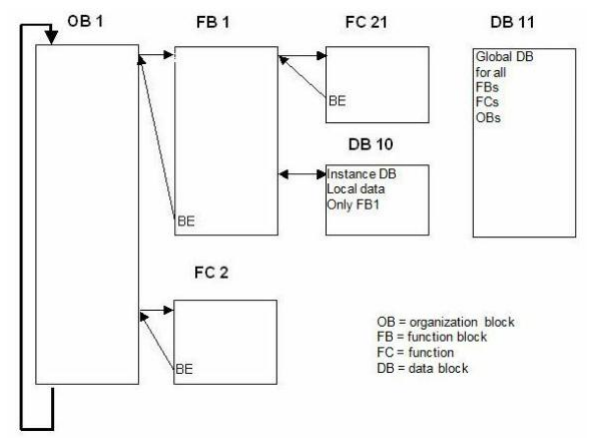
Functions có thể được sử dụng với mục đích:

* + Trả lại giá trị cho hàm chức năng được gọi
  + Thực hiện công nghệ chức năng, ví dụ: điều khiển riêng với các hoạt động nhị phân
  + Ngoài ra, FC có thể được gọi nhiều lần tại các thời điểm khác nhau trong một chương trình, tạo điều kiện cho lập trình chức năng lập đi lặp lại phức tạp.

FB (Function Block): đối với mỗi lần gọi, FB cần một khu vực nhớ. Khi một FB được gọi, một Data Block (DB) đƣợc gán với instance DB. Dữ liệu trong Instance DB sau đó truy cập vào các biến của FB. Các khu vực bộ nhớ khác nhau đã được gán cho một FB nếu nó được gọi ra nhiều lần.

DB (Data Block): DB thường để cung cấp bộ nhớ cho các biến dữ liệu. Có hai loại của khối dữ liệu DB: Global DBs vùng nhớ tất cả các OB, FB và FC có thể đọc được dữ liệu lưu trữ, hoặc có thể tự ghi dữ liệu vào DB, và mặc định DB được gán cho một FB nhất định.

### Cấu trúc lập trình



Hình 4.2 Cấu trúc lập trình

PLC thực hiện chương trình theo chu trình lặp. Mỗi vòng lặp được gọi là vòng quét. Mỗi vòng quét được bắt đầu bằng giai đoạn chuyển dữ liệu từ các cổng vào số tới vùng bộ đệm ảo I, tiếp theo là giai đoạn thực hiện chương trình. Trong từng vòng quét chương trình được thực hiện từ lệnh đầu tiên đến lệnh kết thúc của khối OB1.

### Giới thiệu tập lệnh của PLC sử dụng trong luận văn

#### Bit logic

* Tiếp điểm thường hở

|  |  |  |
| --- | --- | --- |
| LAD |  | Tiếp điểm thường hở sẽ đóng khi giá trị của bit có địa chỉ là n bằng 1 Vùng nhớ: I, Q, M, L,D |

* Tiếp điểm thường đóng

|  |  |  |
| --- | --- | --- |
| LAD |  | Tiếp điểm thường hở sẽ đóng khi giá trị của bit có địa chỉ là n bằng 0 Vùng nhớ: I, Q, M, L,D |

* Lệnh OUT

|  |  |  |
| --- | --- | --- |
| LAD |  | Giá trị của bit có địa chỉ là n sẽ bằng 1 khi đầu vào của lệnh này bằng 1 và ngược lại.  Vùng nhớ: Q, M, L, D |

* Lệnh SET

|  |  |  |
| --- | --- | --- |
| LAD |  | Giá trị của các bit có địa chỉ là n sẽ bằng 1 khi đầu vào của lệnh này bằng 1 Khi đầu vào của lệnh bằng 0 thì bit này vẫn giữ nguyên trạng thái.  Vùng nhớ: Q, M, L, D |

* Lệnh Reset

|  |  |  |
| --- | --- | --- |
| LAD |  | Giá trị của các bit có địa chỉ là n sẽ bằng 0 khi đầu vào của lệnh này bằng 1. Khi đầu vào của lệnh bằng 0 thì các bit này vẫn giữ nguyên trạng thái. Vùng nhớ: Q, M, L, D |

#### Timer

Sử dụng lệnh Timer để tạo một chương trình trễ định thời. Số lượng của  
Timer phụ thuộc vào người sử dụng và số lượng vùng nhớ của CPU. Mỗi timer sử dụng 16 byte IEC\_Timer dữ liệu kiểu cấu trúc DB. Step 7 tự động tạo khối DB khi lấy khối Timer

#### Bit logic

* Timer trễ không nhớ - TON

|  |  |  |
| --- | --- | --- |
| LAD |  | Khi ngõ vào IN ngừng tác động thì reset và dừng hoạt động Timer.  Thay đổi PT khi Timer vận hành không có ảnh hưởng gì |

* Timer trễ sườn xuống – TOF

|  |  |  |
| --- | --- | --- |
| LAD |  | Khi ngõ vào IN ngừng tác động thì reset và dừng hoạt động Timer.  Thay đổi PT khi Timer vận hành không có ảnh hưởng gì |

## PLC S7-1500/PLC S7-1200

### PLC S7-1500

Năm 2013, Siemens đã chính thức công bố dòng sản phẩm mang tên S7-1500. Đây là họ PLC mới của Siemens tích hợp một số chức năng hiện đại, cải tiến tính năng, tối ưu hóa tốc độ xử lý, tiết kiệm điện

Tích hợp các chức năng:

+ Màn hình hiển thị và xem được đặc tuyến

+ Profinet: tích hợp lên tới 3 cổng profinet

+ Web server: Tình trạng CPU được thực hiện trên một trình duyệt Internet tiêu chuẩn, độc lập, người dùng thuận lợi cho việc thu thập thông tin.



Hình 4.3 PLC S7-1500

### PLC S7-1200

S7-1200 là một loại PLC của hãng SIEMENS ngoài ra còn có các dòng khác như: S7-200, S7-300, S7-1500. Trong luận văn, sử dụng PLC SIEMENS S7-1200 1214C DC/DC/DC

CPU cung cấp một cổng PROFINET để giao tiếp qua một mạng PROFINET. Các *module* truyền thông là có sẵn dành cho việc giao tiếp qua các mạng RS232 hay RS485.

****

Hình 4.4 PLC S7-1200

Bộ phận kết nối nguồn.

② Các bộ phận kết nối nối dây của người dùng

③ Các LED trạng thái dành cho I/O tích hợp

④ Bộ phận kết nối PROFINET (phía trên của CPU).

Các kiểu CPU khác nhau cung cấp một sự đa dạng các tính năng và dung lượng giúp cho người dùng tạo ra các giải pháp có hiệu quả cho nhiều ứng dụng khác nhau.

*Bảng 4.1 Tính năng các loại CPU*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Chức năng** | **CPU 1211C** | **CPU 1212C** | | **CPU 1214C** | |
| Kích thước (mm) | 90 x 100 x 75 |  | | 110 x 100 x 75 | |
| Bộ nhớ người dùng:   * Bộ nhớ làm việc * Bộ nhớ nạp * Bộ nhớ giữ lại | * 25 kB * 1 MB * 2 kB |  | | * 50 kB * 2 MB * 2 kB | |
| I/O tích hợp cục bộ   * Kiểu số * Kiểu tương tự | * 6 ngõ vào / 4 ngõ ra * 2 ngõ ra | * 8 ngõ vào / 6 ngõ ra * 2 ngõ ra | | * 14 ngõ vào / 10 ngõ ra * 2 ngõ ra | |
| Kích thước ảnh tiến trình | 1024 *byte* ngõ vào (I) và 1024 *byte* ngõ ra (Q) | | | | |
| Bộ nhớ bit (M) | 4096 *byte* | | | 8192 *byte* | |
| Độ mở rộng các *module* tín hiệu | Không | **2** | | **8** | |
| Bảng tín hiệu | **1** | | | | |
| Các *module* truyền thông | 3 (mở rộng về bên trái) | | | | |
| Các bộ đếm tốc độ cao   * Đơn pha * Vuông pha | 3   * 3 tại 100 kHz * 3 tại 80 kHz | | 4   * 3 tại 100 kHz   1 tại 30 kHz   * 3 tại 80 kHz   1 tại 20 kHz | |  |
| Các ngõ ra xung | **2** | | | | |
| PROFINET | 1 cổng truyền thông Ethernet | | | | |
| Thẻ nhớ | Thẻ nhớ SIMATIC (tùy chọn) | | | | |

Họ S7-1200 cung cấp một số lượng lớn các *module* tín hiệu và bảng tín hiệu để mở rộng dung lượng của CPU. Người dùng còn có thể lắp đặt thêm các *module* truyền thông để hỗ trợ các giao thức truyền thông khác

### Truyền thông S7 1500 với S7 1200

S7 1500 và S7 1200 giao tiếp với nhau thông qua chuẩn Modbus TCP/IP

## Lập trình TIA Portal cho PLC S7-1500/S7-1200

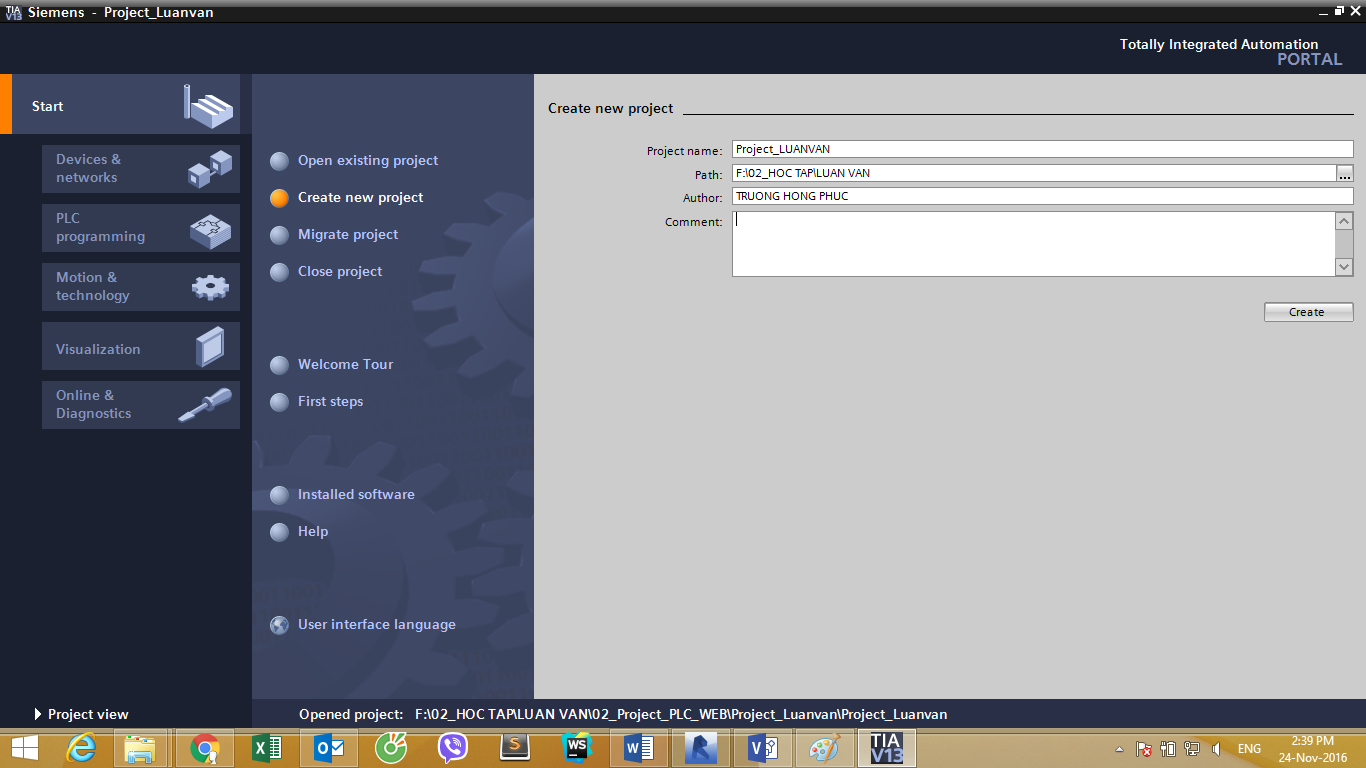
### Tạo project mới

Khởi động phần mềm TIA Portal v14, bằng cách click chuột vào biểu tượng phần mềm TIA Portal v13 trên desktop.



Hình 4.5 Biểu tượng TIA Portal trên Desktop

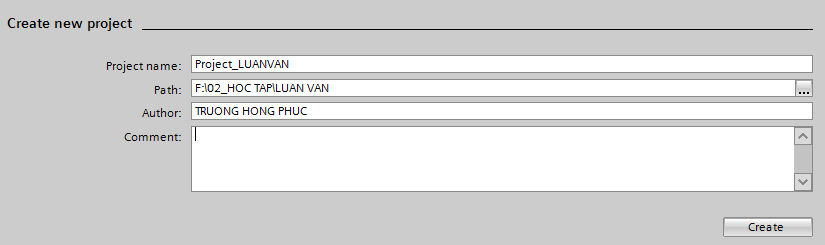
Sau khi khởi động TIA Portal v14. Chọn **Start** 🡪 **Create new project**



Hình 4.6 Giao diện tạo dự án mới của TIA Portal

Trong phần Create new project có các mục

* + Project name: Tên dự án
  + Parth: Đường dẫn đến dự án
  + Author: Tác giả
  + Comment: Thêm những ghi chú cho dự án



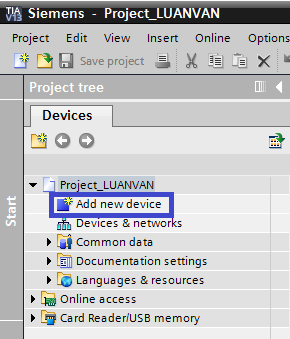
Hình 4.7 Điền cáo thông tin của project

Đặt tên như hình vẽ

* + Author: TRUONG HONG PHUC
  + Comment: Để trống

### Cấu hình cho PLC

Thêm thiết bị PLC S7-1200 vào dự án. Bằng cách click vào **Add new device**



Hình 4.8 Mở dự án Tia Portal

Thêm các PLC cần sử dụng và dự án để lập trình.

### Khai báo các biến, các khối và lập trình

#### Khai báo các biến

Vào **PLC tags** 🡪 **Double Click** 🡪 **Default tag table**

Trong phần **Default tag table** khai báo các thông tin sau

* Name Tên của biến
* Data type Kiểu của biến
* Address Địa chỉ của biến

# XÂY DỰNG MÔ HÌNH ẢO BẰNG PHẦN MỀM LABVIEW

## Labview

### Giới thiệu

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) là một phần mềm máy tính được phát triển bởi công ty National Instruments. LabVIEW dùng trong hầu hết các phòng thí nghiệm, lĩnh vực khoa học kỹ thuật như tự động hóa, điều khiển, điện tử, cơ điện tử, hàng không, hóa sinh, điện tử y sinh

Labview sử dụng rất rộng rãi trong khoa học – kỹ thuật – giáo dục nhằm nhanh chóng và dễ dàng tạo ra các ứng dụng giao tiếp máy tính, đo lường, mô phỏng hệ thống, kết nối thiết bị ngoại vi với máy tính theo thời gian thực. Lập trình đồ họa giống như các ngôn ngữ khác.

### Giao tiếp Labview PLC

Nội dung gồm 4 phần:

1. Cấu hình PC giao tiếp Ethernet.

2. Lập trình PLC S7-1500 với TIA PORTAL V14.

3. Cấu hình KEEPServerEx v5.7 để giao tiếp SCADA.

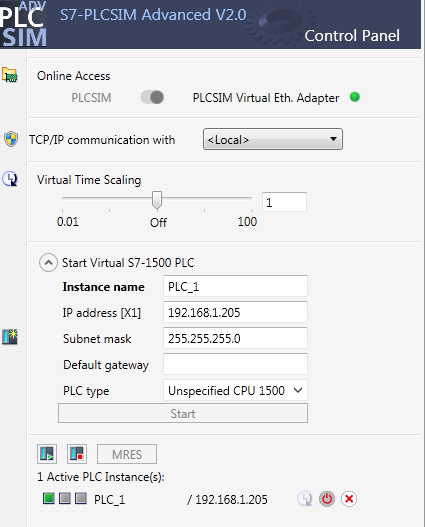
4. Kiểm tra hệ SCADA với LabVIEW truyền thông. Nhiệm vụ của MTU bao gồm:

* Cập nhật dữ liệu từ các thiết bị chấp hành và nhận lệnh từ người điều hành.
* Xuất dữ liệu đến các thiết bị chấp hành RTU.

## PLC SIM Advance 2.0

PLC Sim Advance là chương trình mô phỏng PLC S7 1500 được phát triển bởi hãng Siemens.

PLC Sim Advance cho phép sử dụng PLC như một PLC S7 1500 thật.

****

Hình 5.1 HMI Tags

# PHÁT TRIỂN ỨNG DỤNG PHÍA CLIENT

## Phát triển ứng dụng web

### Giới thiệu

Khái niệm application là ứng dụng - một loại chương trình có khả năng làm cho máy tính thực hiện trực tiếp một công việc nào đó người dùng muốn thực hiện

Ban đầu, các website chỉ bao gồm text, hình ảnh và video, liên kết với nhau thông qua các link. Tác dụng của website là lưu trữ và hiển thị thông tin. Người dùng chỉ có thể đọc, xem, click các link để di chuyển giữa các page.

Về sau, với sự ra đời của các ngôn ngữ server: Nodejs, … các website đã trở nên “động” hơn, có thể tương tác với người dùng. Từ đây, người dùng có thể dùng web để “thực hiện một công việc nào đó bằng máy tính“, do đó web app ra đời.

Web app là những ứng dụng chạy trên web. Thông qua web app, người dùng có thể thực hiện một số công việc: tính toán, chia sẻ hình ảnh, mua sắm … Tính tương tác của web app cao hơn website rất nhiều.

Ngôn ngữ để tạo ra giao diện web là HTML, CSS, Javascript.

## Phát triển ứng dụng Androi

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.

Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào năm 2008.

Android có mã nguồn mở và Google phát hành mã nguồn. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu.

Ngôn ngữ lập trình ứng dụng Androi là Java.

## Phát triển ứng dụng IOS

iOS (trước đây là iPhone OS) là hệ điều hành trên các thiết bị di động của Apple. Đây là hệ điều hành phổ biến thứ 2 trên toàn cầu, sau Android.

Ngôn ngữ lập trình IOS là Swift

# KẾT QUẢ DỰ KIẾN KẾ HOẠCH THỰC HIỆN

## Kết quả thực hiện

### Mô hình thực hiện

Xây dựng 2 mô hình nhà máy bằng thiết bị thật

Xây dựng 1 mô hình nhà máy bằng mô phỏng

### Kết quả mong muốn

Thu thập dữ liệu về cloud server

Điều khiển và giám sát thông qua phần mềm, web trên PC

Điều khiển và giám sát thông qua ứng dụng smartphone hệ điều hành Androi

Điều khiển và giám sát thông qua ứng dụng smartphone hệ điều hành IOS

## Nội dung luận văn

Luận văn gồm 7 chương

Chương 1. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

Chương 2. GIAO THỨC MQTT

2.1. Cơ bản về giao thức MQTT

2.2. Thư viện MQTT Client

Chương 3. XÂY DỰNG MQTT BROKER, WEBSERVICE TRÊN VPS

3.1. VPS

3.2. MQTT Broker

3.3. Xây dựng Server Nodejs, Webservice

Chương 4. TRUYỀN THÔNG S7 1500 VỚI S7 1200 22

4.1. Phần mềm TIA Portal v14

4.2. PLC S7-1500/PLC S7-1200

4.3. Lập trình TIA Portal cho PLC S7-1500/S7-1200

Chương 5. XÂY DỰNG MÔ HÌNH ẢO BẰNG PHẦN MỀM LABVIEW

5.1. Labview

5.2. PLC SIM Advance 2.0

Chương 6. PHÁT TRIỂN ỨNG DỤNG PHÍA CLIENT

6.1. Phát triển ứng dụng web 35

6.2. Phát triển ứng dụng Androi 35

6.3. Phát triển ứng dụng IOS 36

Chương 7. KẾT QUẢ DỰ KIẾN THỰC HIỆN - HƯỚNG PHÁT TRIỂN

7.1. Kết quả thực hiện

7.2. Kết quả, đánh giá

7.3. Hướng phát triển

## Thời gian thực hiện

**Thời gian thực hiện đề tài (Từ ngày 11/02/2019 đến 01/06/2019)**

* Tuần 1: Tìm hiểu giao thức MQTT
* Tuần 2: Truyền thông S7 1500 với MQTT Broker
* Tuần 3: Tìm hiểu VPS, thiết lập máy tính ảo trên cloud
* Tuần 4: Xây dựng MQTT Broker, Webservice trên VPS
* Tuần 5: Truyền thông S7 1500 với S7 1200
* Tuần 6: Điều khiển thiết bị chấp hành (Motor, Pump,,… )
* Tuần 7: Xây dựng server trên private cloud (Nodejs)
* Tuần 8: Xây dựng ứng dụng cho Androi
* Tuần 9: Xây dựng ứng dụng cho IOS
* Tuần 10: Tìm hiểu Labview
* Tuần 11: Điều khiển mô hình bằng mô phỏng Labview
* Tuần 12: Viết báo cáo, và thuyết trình
* Tuần 13: Chỉnh sửa và báo cáo.

TÀI LIỆU THAM KHẢO

[1] Phó GS. TS. Nguyễn Thị Phương Hà, TS. Huỳnh Thái Hoàng, Lý Thuyết  
Điều Khiển Tự Động.

[2] SIEMENS-S7-1200 Programmable Controller System Manual

[3] SIEMENS-S7-1200 PID Control with PID\_Compact

[4] SIEMENS-S7-1200 Basics on Creating HTMLs for SIMATIC CPUs

[5] SIEMENS-S7-1200 Creating User-defined Web Pages on S7-1200 / S7-1500

[6] Các tài liệu trên mạng Internet

* <http://www.w3schools.com/>
* <https://plcvietnam.com.vn/forum>
* <https://www.siemens.com/global/en/home.html>
* <http://www.opcfoundation.org>
* <http://www.ni.com>