**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, 2018**

**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**

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

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. TRƯƠNG ĐÌNH CHÂU**

**TP. HỒ CHÍ MINH, 2018**

|  |  |
| --- | --- |
| 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 12 năm 2018* |

**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 |
|  |  |  |
| **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 12 năm 2018.* |

**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** | |
| **Thời gian thực hiện:** Từ ngày 11/02/2018 đến ngày 01/06/2018 | |
| **Sinh viên thực hiện:**  **TRƯƠNG HỒNG PHÚC – 1770039** | |
| **Nội dung đề tài:**  Mục tiêu:  Phạm vi:  Đối tượng:  Phương pháp thực hiện:  Kết quả mong đợi: Kết nối các nhà máy dựa vào điện toán đám mây | |
| **Xác nhận của Cán bộ hướng dẫn** | TP. HCM, ngày 21 tháng 12 năm 2018  **Sinh viên** |

MỤC LỤC

[Chương 1. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc532944204)

[Chương 2. GIAO THỨC MQTT 6](#_Toc532944205)

[2.1. Cơ bản về giao thức MQTT 6](#_Toc532944206)

[2.1.1. Các thuật ngữ 6](#_Toc532944207)

[2.1.2. Tiêu chuẩn và kiến trúc 7](#_Toc532944208)

[2.1.3. Tính năng 8](#_Toc532944209)

[2.1.4. Gói điều khiển MQTT 9](#_Toc532944210)

[2.2. Thư viện MQTT Client 13](#_Toc532944211)

[2.2.1. Thư viện MQTT Client 13](#_Toc532944212)

[2.2.1.1. Các kiểu dữ liệu mới 13](#_Toc532944213)

[2.2.1.2. Giao diện của khối FB 13](#_Toc532944214)

[2.2.2. Thư viện MQTT Client Security 13](#_Toc532944215)

[2.2.3. FB 13](#_Toc532944216)

[Chương 3. XÂY DỰNG MQTT BROKER, WEBSERVICE TRÊN VPS 15](#_Toc532944217)

[3.1. PLC (Programmable Logic Controller) 15](#_Toc532944218)

[3.1.1. Giới thiệu 15](#_Toc532944219)

[3.1.2. Phần cứng PLC (PLC S7 1200 CPU 1214C DC/DC/DC) 15](#_Toc532944220)

[3.2. Mạch điều khiển động cơ DC 18](#_Toc532944221)

[Chương 4. TRUYỀN THÔNG S7 1500 VỚI S7 1200 19](#_Toc532944222)

[4.1. Phần mềm TIA Portal v14 19](#_Toc532944223)

[4.1.1. Giới thiệu 19](#_Toc532944224)

[4.1.2. Tag PLC 19](#_Toc532944225)

[4.2. Khối tổ chức OB (Oganization Blocks) 20](#_Toc532944226)

[4.3. Hàm chức năng (Function) 20](#_Toc532944227)

[4.4. Cấu trúc lập trình 21](#_Toc532944228)

[4.5. Giới thiệu tập lệnh của PLC sử dụng trong luận văn 22](#_Toc532944229)

[4.5.1. Bit logic 22](#_Toc532944230)

[4.5.2. Timer 23](#_Toc532944231)

[4.5.3. Bit logic 23](#_Toc532944232)

[4.6. Lập trình TIA Portal cho PLC S7-1200 24](#_Toc532944233)

[4.6.1. Tạo project mới 24](#_Toc532944234)

[4.6.2. Cấu hình cho PLC 25](#_Toc532944235)

[4.6.3. Khai báo các biến, các khối và lập trình các module analog 26](#_Toc532944236)

[4.6.3.1. Khai báo các biến 26](#_Toc532944237)

[4.6.3.2. Module Analog và viết chương trình đọc Analog 26](#_Toc532944238)

[4.6.4. Lập trình PID 27](#_Toc532944239)

[4.6.4.1. Giới thiệu khối PID Compact v2.2 27](#_Toc532944240)

[4.6.4.1. Cách cấu hình và sử dụng bộ PID\_Compact 29](#_Toc532944241)

[Chương 5. XÂY DỰNG MÔ HÌNH ẢO BẰNG PHẦN MỀM LABVIEW 30](#_Toc532944242)

[5.1. Tổng quan hệ thống SCADA 30](#_Toc532944243)

[5.1.1. Giới thiệu 30](#_Toc532944244)

[5.1.2. Cấu trúc cơ bản của một hệ thống SCADA 30](#_Toc532944245)

[5.2. Tổng quan hệ thống SCADA 31](#_Toc532944246)

[5.3. Thiết kế hệ thống SCADA với WinCC 31](#_Toc532944247)

[5.3.1. Tạo Project với WinCC v13 31](#_Toc532944248)

[5.3.2. Kết quả 32](#_Toc532944249)

[5.4. Thiết kế hệ thống SCADA với Webserver 32](#_Toc532944250)

[5.4.1. Tổng quan về web 32](#_Toc532944251)

[5.4.1.1. Giới thiệu về HTML 32](#_Toc532944252)

[5.4.1.2. Giới thiệu về Java Script 32](#_Toc532944253)

[5.4.1.3. Hiển thị biến trên CPU từ trang web 32](#_Toc532944254)

[5.4.2. Thiết kế hệ thống SCADA với Webserver 33](#_Toc532944255)

[5.4.2.1. Viết chương trình bằng ngôn ngữ HTML 33](#_Toc532944256)

[Chương 6. PHÁT TRIỂN ỨNG DỤNG PHÍA CLIENT 34](#_Toc532944257)

[6.1. Tổng quan hệ thống SCADA 34](#_Toc532944258)

[6.1.1. Giới thiệu 34](#_Toc532944259)

[6.1.2. Cấu trúc cơ bản của một hệ thống SCADA 34](#_Toc532944260)

[6.2. Tổng quan hệ thống SCADA 35](#_Toc532944261)

[6.3. Thiết kế hệ thống SCADA với WinCC 35](#_Toc532944262)

[6.3.1. Tạo Project với WinCC v13 35](#_Toc532944263)

[6.3.2. Kết quả 36](#_Toc532944264)

[6.4. Thiết kế hệ thống SCADA với Webserver 37](#_Toc532944265)

[6.4.1. Tổng quan về web 37](#_Toc532944266)

[6.4.1.1. Giới thiệu về HTML 37](#_Toc532944267)

[6.4.1.2. Giới thiệu về Java Script 37](#_Toc532944268)

[6.4.1.3. Hiển thị biến trên CPU từ trang web 37](#_Toc532944269)

[6.4.2. Thiết kế hệ thống SCADA với Webserver 38](#_Toc532944270)

[6.4.2.1. Viết chương trình bằng ngôn ngữ HTML 38](#_Toc532944271)

[Chương 7. KẾT QUẢ THỰC HIỆN - HƯỚNG PHÁT TRIỂN 39](#_Toc532944272)

[7.1. Kết quả thực hiện 39](#_Toc532944273)

[7.1.1. Mô hình thực hiện 39](#_Toc532944274)

[7.1.2. Lưu đồ giải thuật của hệ thống 39](#_Toc532944275)

[7.1.3. Điều khiển PID 39](#_Toc532944276)

[7.1.4. Giám sát qua SCADA 39](#_Toc532944277)

[7.1.4.1. Giám sát bồn nước qua phần giao diện WinCC 39](#_Toc532944278)

[7.1.4.2. Giám sát bồn nước qua Webserver 39](#_Toc532944279)

[7.2. Đánh giá 39](#_Toc532944280)

[7.2.1. Kết quả 39](#_Toc532944281)

[7.2.2. Ưu điểm 39](#_Toc532944282)

[7.2.3. Nhược điểm 39](#_Toc532944283)

[7.3. Ứng dụng 39](#_Toc532944284)

[7.4. Hướng phát triển 40](#_Toc532944285)

DANH MỤC HÌNH VẼ

[Hình 1.1 Nội dung của luận văn 1](#_Toc532937287)

[Hình 2.3 Sơ đồ bộ điều khiển PID 2](#_Toc532937288)

[Hình 4.1 PLC S7-1200 2](#_Toc532937289)

[Hình 5.1 PLC Tags trong TIA PORTAL 3](#_Toc532937290)

[Hình 5.2 Cấu trúc lập trình 5](#_Toc532937291)

[Hình 5.3 Biểu tượng TIA Portal trên Desktop 8](#_Toc532937292)

[Hình 5.4 Giao diện tạo dự án mới của TIA Portal 8](#_Toc532937293)

[Hình 5.5 Điền cáo thông tin của project 9](#_Toc532937294)

[Hình 5.6 Mở project Tia Portal 9](#_Toc532937295)

[Hình 5.7 Mở dự án Tia Portal 10](#_Toc532937296)

[Hình 5.8 Thêm PLC vào dự án 11](#_Toc532937297)

[Hình 5.9 Set IP cho S7-1200 12](#_Toc532937298)

[Hình 5.10 Set IP cho PC 12](#_Toc532937299)

[Hình 5.11 Các biến sử dụng trong Project 13](#_Toc532937300)

[Hình 5.16 Khối hàm PID\_Compact 14](#_Toc532937301)

[Hình 6.2 Devices & Networks 18](#_Toc532937302)

[Hình 6.4 Add new screen 18](#_Toc532937303)

[Hình 6.5 Các đối tượng cơ bản trong WinCC 18](#_Toc532937304)

[Hình 6.6 HMI Tags 18](#_Toc532937305)

[Hình 6.13 Thư mục chứa các trang web 18](#_Toc532937306)

[Hình 7.6 Kết quả chạy thực tế trên WinCC 18](#_Toc532937307)

[Hình 7.7 Kết quả chạy thực tế của Trend trên web 18](#_Toc532937308)

[Hình 7.8 Kết quả chạy thực tế của Report trên web 18](#_Toc532937309)

DANH MỤC BẢNG

[*Bảng 2.1 Ảnh hưởng của các hệ số trên đáp ứng hệ kín* 7](#_Toc532859332)

[*Bảng 4.1 Tính năng các loại CPU* 9](#_Toc532859333)

[*Bảng 5.1 Các thuộc tính trong khối hàm PID\_Compact* 23](#_Toc532859334)

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 tập trung

**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

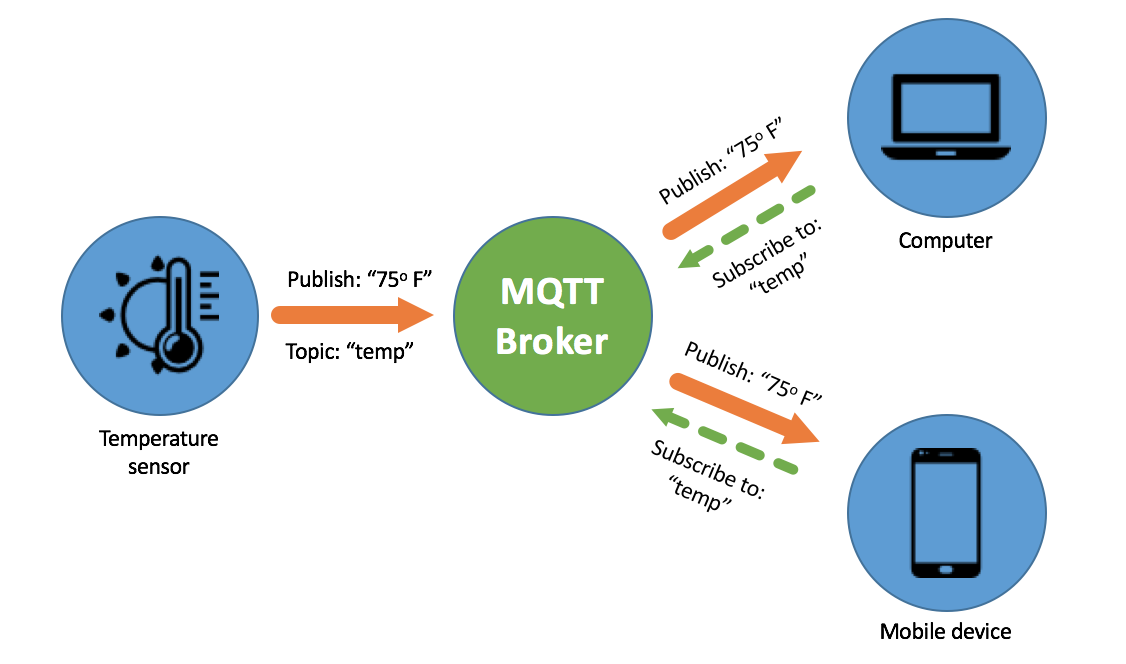
* **Tiêu chuẩn ISO**

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 2.2 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 Ảnh hưởng của các hệ số trên đáp ứng hệ kín*

|  |  |  |
| --- | --- | --- |
| 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 |

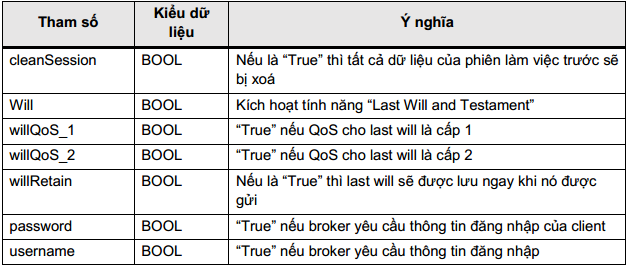
**Kiểu dữ liệu “typeMqttParam”**

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

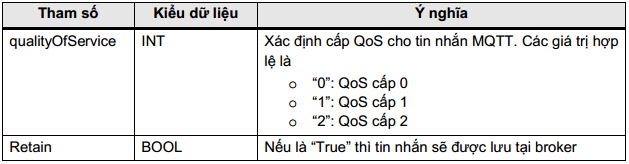
* Các cờ để kết nối.
* Các cờ để gửi tin nhắn.
* Thông tin đăng nhập broker.
* Chủ đề.
* Nội dụng tin nhắn.

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

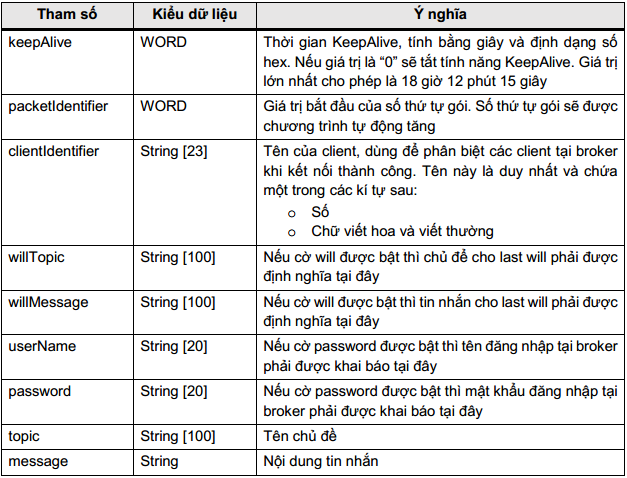
“typeMqttConnectFlags” chứa các cờ để kết nối với broker.



“typeMqttPublishFlags” chứa các cờ cho tin nhắn MQTT



Các tham số khác của kiểu dữ liệu “typeMqttParam”



#### Giao diện của khối FB

### Thư viện MQTT Client Security

### FB

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

FB

# XÂY DỰNG MQTT BROKER, WEBSERVICE TRÊN VPS

## PLC (Programmable Logic Controller)

### Giới thiệu

PLC là thiết bị điều khiển có khả năng lập trình được, PLC được sử dụng ngày càng rộng rãi trong công nghiệp cũng như trong nhiều mục đích khác nhau do yêu cầu về tự động hoá ngày càng cao.

Cùng với sự phát triển của công nghệ máy tính, hiện nay PLC càng chiếm ưu thế trong ứng dụng tự động hoá công nghiệp nói chung do các yếu tố sau:

* Cho phép nhanh chóng thay đổi chương trình điều khiển.
* Có chức năng liên kết, truyền thông, nối mạng ở nhiều cấp độ nhằm đáp  
  ứng kịp thời nhu cầu điều khiển và giám sát hệ thống làm việc.
* Độ tin cậy cao trong môi trường công nghiệp.
* Dễ thay đổi, dễ nâng cấp, dễ mở rộng, bảo trì và sửa chữa nhanh chóng.

### Phần cứng PLC (PLC S7 1200 CPU 1214C DC/DC/DC)

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 3.1 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.

## Mạch điều khiển động cơ DC

Mạch công suất dùng để khuếch đại công suất tín hiệu từ ngõ ra PWM của PLC cấp điều khiển động cơ DC, và qua hai điện trở 2.2k để phân áp

cơ DC thực tế

Sơ đồ nối chân:

* Nguồn dương (+) 12V được nối với chân 2 của 12V, nguồn âm (-) 12v được nối với chân 1 của 12V
* Nguồn dương (+) của động cơ được nối với chân 1 của DONGCO1, nguồn âm (-) được nối với chân 2 của DONGCO1
* Chân Q0.0 và Q0.2 của PLC được nối với chân 1,2 của PLC, Chân M của PLC được nối với chân 3 của PLC
* IC IRF540 là IC dùng để kích xung
* Khi PLC cấp xung dương, điện áp qua điện trở R = 2.2k sẽ kích cho IRF 540 hoạt động, IRF hoạt động thì từ nguồn dương VCC qua động cơ, đi qua FET và về Mass, làm cho động cơ hoạt động. Khi ta không cấp xung cho PLC thì FET ngắt và làm cho động cơ dừng.
* Diode D1 dùng để bảo vệ FET, chống dòng điện dội ngược về khi đóng ngắt.

# TRUYỀN THÔNG S7 1500 VỚI S7 1200

## Phần mềm TIA Portal v14

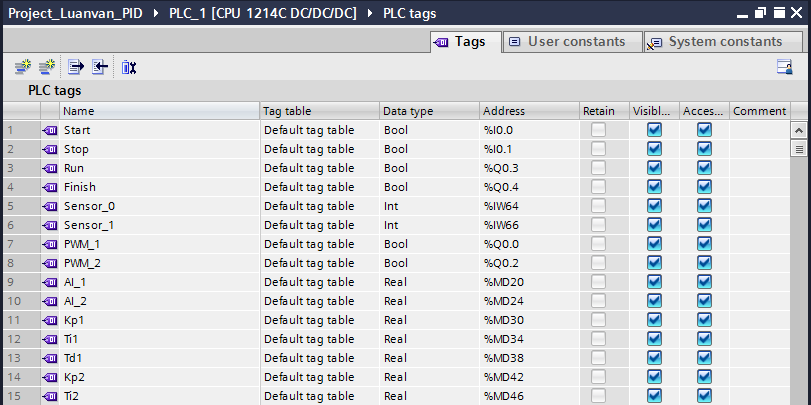
### Giới thiệu

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 5.1 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

Start Information: Khi một số OB được bắt đầu, hệ điều hành đọc ra thông tin được thẩm định trong chương trình người dùng, điều này rất hữu ích cho việc chẩn đoán lỗi, cho dù thông tin được đọc ra được cung cấp trong các mô tả của các khối OB.

## 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.

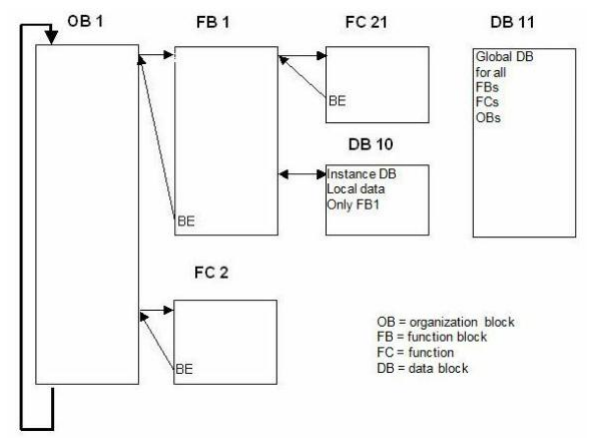
Functions có thể được sử dụng với mục đích:

* + 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 5.2 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 |

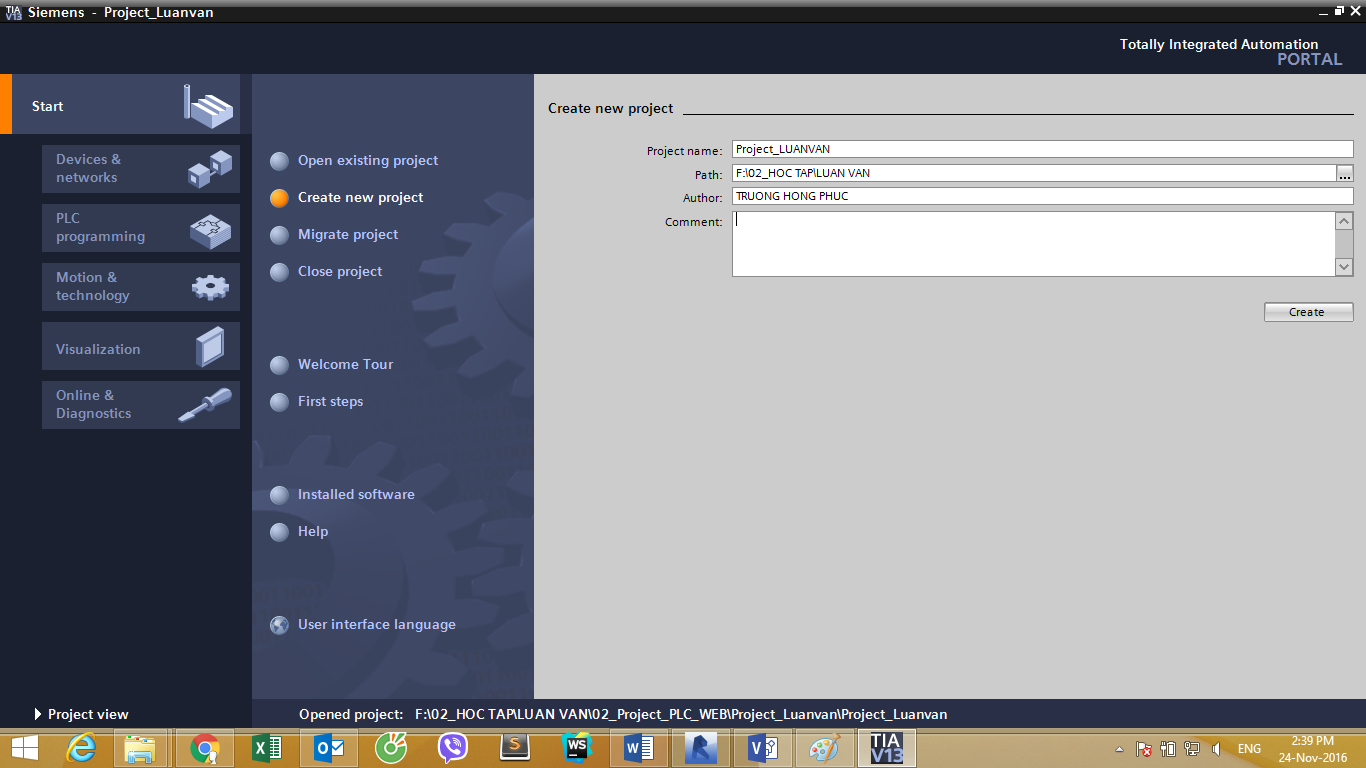
## Lập trình TIA Portal cho PLC S7-1200

### Tạo project mới

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

Hình 5.3 Biểu tượng TIA Portal trên Desktop

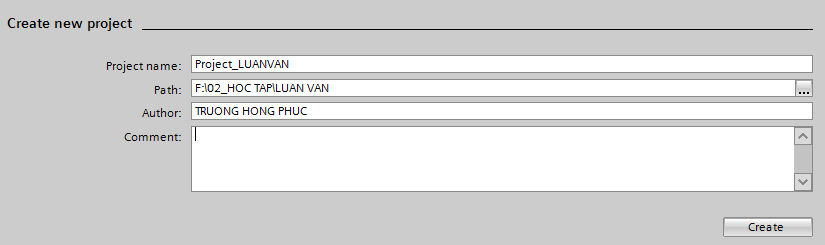
Sau khi khởi động TIA Portal v13. Chọn **Start** 🡪 **Create new project**



Hình 5.4 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



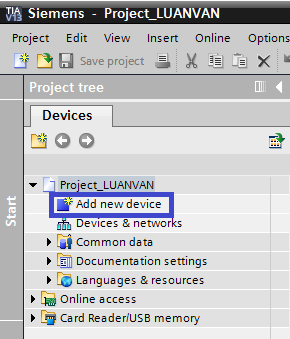
Hình 5.5 Điền cáo thông tin của project

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 5.7 Mở dự án Tia Portal

Trong luận văn sử dụng PLC S7 1200 (CPU 1214 DC/DC/DC). Thêm CPU phù hợp với PLC thực tế.

s

Hình 5.8 Thêm PLC vào dự án

Để PC và PLC có thể giao tiếp được với nhau thì PC và PLC phải cùng lớp mạng. Cần đặt địa chỉ IP tĩnh cho PC và PLC.

Đặt địa chỉ IP cho PLC S7-1200 trong IP protocol với thông số:

* IP address: **192.168.1.3**
* Subnet mask: **255.255.255.0**

Hình 5.9 Set IP cho S7-1200

Đặt địa chỉ IP cho PC (Laptop) trong **Internet Protocol Version 4** với thông số:

* IP address: **192.168.1.2**
* Subnet mask: **255.255.255.0**

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

#### 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

Các biến sử dụng trong project:

Hình 5.11 Các biến sử dụng trong Project

#### Module Analog và viết chương trình đọc Analog

* Module Analog Input

Khởi tạo chương trình đọc analog để đọc về giá trị của cảm biến trên PLC bằng cách tạo hàm FC để đọc và chuyển đổi giá trị về số thực Real

Trong đó:

FLOAT(IN) Giá trị Analog đầu vào

OUT Giá trị đầu ra (2cm đến 510cm)

các thông số cần chọn vào

### Lập trình PID

#### Giới thiệu khối PID Compact v2.2

Sử dụng khối hàm PID\_Compact v2.2 trong TIA Portal để sử dụng cho luận văn

Hình 5.16 Khối hàm PID\_Compact

PID\_Compact cung cấp 1 bộ điều khiển PID với chức năng tự điều chỉnh cho chế độ tự động hoặc bằng tay.

*Bảng 5.1 Các thuộc tính trong khối hàm PID\_Compact*

|  |  |  |  |
| --- | --- | --- | --- |
| Setpoint | IN | Real | Giá trị đặt của bộ điều khiển PID trong chế độ tự động. Giá trị mặc định: 0.0 |
| Input | IN | Real | Process value. Giá trị mặc định: 0.0  Để sử dụng chế độ này phải cài đặt  Config.InputPerOn = FALSE. |
| Input\_PER | IN | Word | Giá trị xử lý analog (tùy chọn).Giá trị mặc định: W#16#0.  Để sử dụng chế độ này phải cài đặt  Config.InputPerOn = TRUE. |
| ManualEnable | IN | Bool | Cho phép hoặc không cho phép chế độ vận hành bằng tay. Giá trị mặc định: FALSE.   * Trên cạnh của sự chuyển đổi từ FALSE sang TRUE,bộ điều khiển PID chuyển sang chứ độ bằng tay, State=4 và Mode vẫn không đổi * Trên cạnh của sự thay đổi từ TRUE sang FALSE,bộ điều khiển PID chuyển tới chế độ vận hành cuối cùng và State = Mode |
| ManualValue | IN | Real | Giá trị xử lí cho việc vận hành bằng tay.  Giá trị mặc định: 0.0 |
| Reset | IN | Bool | Khởi động lại bộ điều khiển. Mặc định FALSE  Nếu Reset=TRUE,những điều sau đây được áp dụng:   * Chế độ vận hành không hoạt động * Input value = 0 * Integral part of the process value = 0 * Giá trị trung gian của hệ thống được reset các thông số PID được duy trì. |
| ScaledInput | OUT | Real | Scaled process value. Mặc định: 0.0 |
|  |  |  |  |
| SetpointLimit\_L | OUT | Bool | Giới hạn dưới của SP.Default value: FALSE  Nếu SetpointLimit\_H=TRUE,đạt đến giới hạn dưới tuyệt đối của SP. Mặc định: FALSE |
| InputWarning\_H | OUT | Bool | Nếu InputWarning\_H = TRUE,giá trị xử lí(PV) đạt đến hay vượt mức giới hạn trên Mặc định: FALSE |
| InputWarning\_L | OUT | Bool | Nếu InputWarning\_H = TRUE, giá trị xử lí (PV) đạt đến hay vượt mức giới hạn dưới. Mặc định: FALSE |
| Error | OUT | DWord | Thông báo lỗi. Giá trị mặc định: DW#16#0000 (Không lỗi) |

#### Cách cấu hình và sử dụng bộ PID\_Compact

Đầu tiên phải tạo một khối hàm ngắt chu kỳ OB30 đối với CPU 12124 vì bộ PID cần thời gian để thực thi.

# XÂY DỰNG MÔ HÌNH ẢO BẰNG PHẦN MỀM LABVIEW

## Tổng quan hệ thống SCADA

### Giới thiệu

SCADA (Supervisory Control And Data Acquisition) là một hệ thống thu thập dữ liệu, giám sát và điều khiển các quá trình từ xa. Người vận hành có thể nhận biết và điều khiển hoạt động các thiết bị thông qua máy tính và mạng truyền thông. Nói cách khác, SCADA thường được dùng để chỉ tất cả các hệ thống máy tính được thiết kế để thực hiện các chức năng sau:

* Thu
* tự động hoặc bằng tay một cách kịp thời và chính xác

### Cấu trúc cơ bản của một hệ thống SCADA

Một hệ thống SCADA cơ bản có các thành phần chính là: MTU, RTU và thành phần truyền thông

* **MTU ( Master Terminal Unit)**

MTU là trung tâm của một hệ thống SCADA, trong thực tế nó thường là một hệ máy tính công nghiệp. MTU giao tiếp với người điều hành và RTU thông qua khối truyền thông. Ngoài ra MTU còn được kết nối với các thiết bị ngoại vi như monitor, máy in và có thể kết nối với mạng truyền thông. Nhiệm vụ của MTU bao gồm:

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

thống không dây wireless), các trạm lặp trong trường hợp truyền đi xa.

Phần mềm: đó là các giao thức truyền thông (protocol), các ngôn ngữ lập trình được dùng để các thiết bị có thể giao tiếp với nhau.

## Tổng quan hệ thống SCADA

WinCC là một hệ thống điều khiển trung lập có tính công nghiệp và tính kĩ thuật, hệ thống màn hình hiển thị đồ họa và điều khiển nhiệm vụ trong sản xuất và tự động hóa quá trình. Hệ thống này đưa ra nhưng module chức năng tích hợp công nghiệp cho hiển thị đồ họa, những thông báo, lưu trữ, báo cáo.

## Thiết kế hệ thống SCADA với WinCC

### Tạo Project với WinCC v13

WinCC đã được tích hợp luôn trong bộ phần mềm TIA v13 Portal nên từ phần viết chương trình ta thêm phần SCADA bằng cách:

Trong Project chọn **Add new device 🡪 PC systems 🡪 SIMATIC HMI**

Hình 6.2 Devices & Networks

Sau đó liên kết 2 thiết bị với nhau. Lưu ý là 2 thiết bị này phải cùng thuộc lớp mạng

Địa chỉ của PLC: 192.168.1.3

Địa chỉ của PC-System là: 192.168.1.10

SCADA cho luận văn.

Hình 6.5 Các đối tượng cơ bản trong WinCC

Dùng các đối tượng này để tạo rat rang giao hiện phù hợp với luận văn. Để thiết lập các giá trị Tags ta vào HMI tags. Thường thì đặt tên HMI tags trùng với PLC tag để dễ quản lí. Để chương trình cập nhật nhanh hơn, trong Acquisition cycle chọn 100ms.

Hình 6.6 HMI Tags

### Kết quả

Kết quả sau khi dùng những đối tượng cơ bản tạo giao diện. Ta được Trang chủ Hom

## Thiết kế hệ thống SCADA với Webserver

### Tổng quan về web

#### Giới thiệu về HTML

HTML (viết tắt cho HyperText Markup Language, tức là “Ngôn ngữ Đánh

giản trong file HTML.

#### Giới thiệu về Java Script

JavaScript là một ngôn ngữ lập trình được tạo ra cho mục đích tối ưu hoá Như là thông số của 1 HTML tag.

* Là 1 file JS ngoài.

chỉ vẫn nhập mã vào một lần và có thể tham chiếu nó vào nhiều file HTML khác nhau.

Cú pháp như sau:

<script src="<Script>.js" type="text/javascript"></script>

#### Hiển thị biến trên CPU từ trang web

Các biến phải được khai báo thông qua lệnh AWP:

Cú pháp: **:=<Varname>:**

Trong đó:

* + Các biến phải được chuyển giao tới CPU (ví dụ phương pháp POST trong file HTML).

Ví dụ: Ghi dữ liệu vào PLC thông qua Input trong HTML

<!-- AWP\_In\_Variable Name='"Target\_Level"' -->

<form method="post">

<p>Input Target Level: <input name='"Target\_Level"' type="text">

</p>

</form>

### Thiết kế hệ thống SCADA với Webserver

#### Viết chương trình bằng ngôn ngữ HTML

Tạo các trang web và hình ảnh trong 1 thư mục.

# PHÁT TRIỂN ỨNG DỤNG PHÍA CLIENT

## Tổng quan hệ thống SCADA

### Giới thiệu

SCADA (Supervisory Control And Data Acquisition) là một hệ thống thu thập dữ liệu, giám sát và điều khiển các quá trình từ xa. Người vận hành có thể nhận biết và điều khiển hoạt động các thiết bị thông qua máy tính và mạng truyền thông. Nói cách khác, SCADA thường được dùng để chỉ tất cả các hệ thống máy tính được thiết kế để thực hiện các chức năng sau:

* Thu
* tự động hoặc bằng tay một cách kịp thời và chính xác

### Cấu trúc cơ bản của một hệ thống SCADA

Một hệ thống SCADA cơ bản có các thành phần chính là: MTU, RTU và thành phần truyền thông

* **MTU ( Master Terminal Unit)**

MTU là trung tâm của một hệ thống SCADA, trong thực tế nó thường là một hệ máy tính công nghiệp. MTU giao tiếp với người điều hành và RTU thông qua khối truyền thông. Ngoài ra MTU còn được kết nối với các thiết bị ngoại vi như monitor, máy in và có thể kết nối với mạng truyền thông. Nhiệm vụ của MTU bao gồm:

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

thống không dây wireless), các trạm lặp trong trường hợp truyền đi xa.

Phần mềm: đó là các giao thức truyền thông (protocol), các ngôn ngữ lập trình được dùng để các thiết bị có thể giao tiếp với nhau.

## Tổng quan hệ thống SCADA

WinCC là một hệ thống điều khiển trung lập có tính công nghiệp và tính kĩ thuật, hệ thống màn hình hiển thị đồ họa và điều khiển nhiệm vụ trong sản xuất và tự động hóa quá trình. Hệ thống này đưa ra nhưng module chức năng tích hợp công nghiệp cho hiển thị đồ họa, những thông báo, lưu trữ, báo cáo.

## Thiết kế hệ thống SCADA với WinCC

### Tạo Project với WinCC v13

WinCC đã được tích hợp luôn trong bộ phần mềm TIA v13 Portal nên từ phần viết chương trình ta thêm phần SCADA bằng cách:

Trong Project chọn **Add new device 🡪 PC systems 🡪 SIMATIC HMI**

Hình 6.2 Devices & Networks

Sau đó liên kết 2 thiết bị với nhau. Lưu ý là 2 thiết bị này phải cùng thuộc lớp mạng

Địa chỉ của PLC: 192.168.1.3

Địa chỉ của PC-System là: 192.168.1.10

Để tạo giao diện lập trình WinCC ta phải tạo một trang Screen để làm việc.

Chọn **HMI\_RT\_1 [WinCC RT Advanced]** 🡪 **Screens** 🡪 **Add new screen**

Hình 6.4 Add new screen

Sau khi đã tạo ra một trang Screen để làm việc sử dụng các đối tượng trong khối tùy chọn **Options** (Basic objects, Elements, Controls, Graphich…) để thiết kế SCADA cho luận văn.

Hình 6.5 Các đối tượng cơ bản trong WinCC

Dùng các đối tượng này để tạo rat rang giao hiện phù hợp với luận văn. Để thiết lập các giá trị Tags ta vào HMI tags. Thường thì đặt tên HMI tags trùng với PLC tag để dễ quản lí. Để chương trình cập nhật nhanh hơn, trong Acquisition cycle chọn 100ms.

Hình 6.6 HMI Tags

### Kết quả

Kết quả sau khi dùng những đối tượng cơ bản tạo giao diện. Ta được Trang chủ Home

## Thiết kế hệ thống SCADA với Webserver

### Tổng quan về web

#### Giới thiệu về HTML

HTML (viết tắt cho HyperText Markup Language, tức là “Ngôn ngữ Đánh dấu Siêu văn bản”) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web, nghĩa là các mẫu thông tin được trình bày trên World Wide Web.

giản trong file HTML.

#### Giới thiệu về Java Script

JavaScript là một ngôn ngữ lập trình được tạo ra cho mục đích tối ưu hoá Như là thông số của 1 HTML tag.

* Là 1 file JS ngoài.

Nếu muốn sử dụng cùng chức năng JavaScript đó với nhiều file HTML. Ta chỉ vẫn nhập mã vào một lần và có thể tham chiếu nó vào nhiều file HTML khác nhau.

Cú pháp như sau:

<script src="<Script>.js" type="text/javascript"></script>

#### Hiển thị biến trên CPU từ trang web

Các biến phải được khai báo thông qua lệnh AWP:

Cú pháp: **:=<Varname>:**

Trong đó:

* + Các biến phải được chuyển giao tới CPU (ví dụ phương pháp POST trong file HTML).

Ví dụ: Ghi dữ liệu vào PLC thông qua Input trong HTML

<!-- AWP\_In\_Variable Name='"Target\_Level"' -->

<form method="post">

<p>Input Target Level: <input name='"Target\_Level"' type="text">

</p>

</form>

### Thiết kế hệ thống SCADA với Webserver

#### Viết chương trình bằng ngôn ngữ HTML

Tạo các trang web và hình ảnh trong 1 thư mục.

Trong luận văn, dùng các trang chính

Hình 6.13 Thư mục chứa các trang web

Lập trình web bằng ngôn ngữ HTML, Gồm có 2 phần chính là <head> </head> và <body> </body>

Thẻ <head> là nơi chứa tất cả các phần tử đầu vào của văn bản HTML. Phần tử đầu vào có thể là các script, các đoạn mã hướng dẫn trình duyệt để tìm các tập tin quy định style cho trang web (CSS), cung cấp các thông tin Meta, các mô tả, từ khoá...

Thẻ <body> là thành phần quan trọng nhất, nó chứa nhưng đoạn mã HTML dùng để hiển thị trên website.

Trong luận văn, Trang index.html lập trình như sau

# KẾT QUẢ THỰC HIỆN - HƯỚNG PHÁT TRIỂN

## Kết quả thực hiện

### Mô hình thực hiện

Đã hoàn thiện xong mô hình gồm phần cứng hệ bồn nước đôi, hai bồn nước trong suốt, tiện cho việc quan sát. Mô hình sử dụng 2 cảm biến siêu âm SRF-06

### Lưu đồ giải thuật của hệ thống

### Điều khiển PID

Dùng công cụ của phần mềm dò được bộ thông số của cả bộ PID\_Compact.

### Giám sát qua SCADA

#### Giám sát bồn nước qua phần giao diện WinCC

Kết quả chạy thực tế trên giao diện, sử dụng phần mềm WinCC v13

Hình 7.6 Kết quả chạy thực tế trên WinCC

#### Giám sát bồn nước qua Webserver

Hình 7.7 Kết quả chạy thực tế của Trend trên web

## Đánh giá

### Kết quả

* Đã dò được thông số PID, hệ thống đáp ứng đúng với mong muốn, yêu cầu đã đặt ra.
* Khi đặt 2 mức nước khác nhau ở hai bồn thì hệ thống thực hiện đúng với giá trị đã đặt, tuy nhiên đáp ứng vẫn còn chậm so với mong muốn. Nhưng vì trong điều khiển trong các nhà máy công nghiệp thì không yêu cầu độ chính xác quá cao và đáp ứng quá nhanh nên có thể đánh giá hệ thống hoạt động tương đối tốt.
* Đã kết nối PLC ra Internet bằng cách cho PLC kết nối với Router và mở port 80, đăng kí tên miền hongphucbkhcm.ddns.net. Tuy nhiên thời gian truyền từ web xuống PLC còn rất chậm nên Web chỉ mới đáp ứng được cho việc giám sát, còn việc điều khiển thì WinCC hay điều khiển trực tiếp sẽ cho đáp ứng tốt hơn.

### Ưu điểm

* Hệ thống hoạt động tốt. Hệ bồn nước đôi hoạt động theo đúng yêu cầu đã đặt ra và đúng giá trị setpoint.
* Thiết kế được giao diện điều khiển qua WinCC một cách trực quan, dễ quan sát, và điều khiển, đáp ứng nhanh.
* Thiết kế được giao diện web bằng ngôn ngữ HTML, trang web sinh động, dễ nhìn, có gần như đầy đủ các chức năng giống như WinCC

### Nhược điểm

* Kết quả điều khiển PID vẫn còn sai số so với giá trị đặt, do ảnh hưởng bởi độ chính xác của cảm biến, tính trễ của motor bơm nước.
* Cảm biến cho dòng ra là 4mA đến 25mA, dùng điện trở 250 Ohm để biến dòng thành áp. Nhưng điện trở không đúng chính xác đến 250Ohm áp trả về ở 2 mức nước là khác nhau mặc dù cùng chung mực nước.
* Điều khiển và giám sát qua web còn đáp ứng chậm do sử dụng nhiều biến dẫn đến thời gian truyền dữ liệu từ web về PLC bị ảnh hưởng.
* Việc kết nối ra Internet còn chậm nên web chỉ mới đáp ứng cho việc giám sát.

## Ứng dụng

Điều khiển mức nước trong đề tài có thể ứng dụng rộng rãi trong các quá trình sản xuất ở các nhà máy như nhà máy sản xuất bia, rượu, hóa chất….

## Hướng phát triển

Có thể dùng nhiều thuật điều khiển hiện đại hơn giúp mô hình chính xác hơn như điều khiển PID Fuzy, phân bố cực

Phần cứng: Dùng biến tần để điều khiển điều khiển tốt hơn.

Phát triển đề tài Webserver không chỉ thực hiện local mà còn đưa webserver lên Internet. Khi đó việc điều khiển sẽ không phụ thuộc vào vị trí. Và chúng ta sẽ giám sát được ở bất kì nơi nào có kết nối Internet. Nhưng điều đó sẽ làm cho webserver có tính bảo mật không cao, nguy cơ mất kiểm soát quyền điều khiển. Và khi sử dụng Internet cũng sẽ đối mặt với vấn đề về tốc độ, độ ổn định của mạng Internet đang sử dụng

Trong thực tế thì việc điều khiển qua Web cũng được sử dụng rất ít do tính bảo mật. Trong tương lai nếu vấn đề này được giải quyết thì việc điều khiển và giám sát qua Internet là tất yếu.

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>