



**TRƯỜNG ĐẠI HỌC CÀN THƠ  
KHOA CÔNG NGHỆ  
BỘ MÔN TỰ ĐỘNG HÓA**  
\*\*\*\*\*\*

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**LẬP TRÌNH ỨNG DỤNG S7-1200**

Sinh viên thực hiện:

**Nguyễn Thị Hồng Cẩm**    **1080901**  
**Phạm Tiến Huy**                **1080974**

Cán bộ hướng dẫn:

**ThS. Nguyễn Hoàng Dũng**

**Cà Mau, 12 – 2011**



**TRƯỜNG ĐẠI HỌC CẦN THƠ**  
**KHOA CÔNG NGHỆ**  
**BỘ MÔN TỰ ĐỘNG HÓA**  
\*\*\*\*\*\*

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**LẬP TRÌNH ỨNG DỤNG S7-1200**

Sinh viên thực hiện:

**Nguyễn Thị Hồng Cẩm**    **1080901**  
**Phạm Tiến Huy**                **1080974**

Cán bộ hướng dẫn:

**ThS. Nguyễn Hoàng Dũng**

Cán bộ phản biện:

**Ks. Nguyễn Văn Khanh**  
**TS. Trần Thanh Hùng**

Luận văn được bảo vệ tại: Hội đồng chấm luận văn tốt nghiệp Bộ môn Tự Động Hóa, Khoa Công Nghệ, Trường Đại học Cần Thơ vào ngày 10 tháng 12 năm 2011

**Cần Thơ, 12 – 2011**

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

ອະນຸມາດ

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Cần Thơ, ngày ... tháng ... năm 2011

## NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 1

ອະນຸມາດ

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Cần Thơ, ngày ... tháng ... năm 2011

## NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 2

ອະນາກົດ

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Cần Thơ, ngày ... tháng ... năm 2011

## LỜI CAM ĐOAN

Lập trình ứng dụng S7-1200 là đề tài sử dụng dòng PLC mới nhất của hãng Siemens để làm ứng dụng thực tế. Đề tài hướng đến những ứng dụng có thể áp dụng thực tế là điều khiển tốc độ động cơ và điều khiển nhiệt độ. Đây là cơ hội cho chúng em tiếp cận được với dòng sản phẩm PLC mới và làm được ứng dụng thực tế. Vì vậy chúng em chọn đề tài này để làm luận văn tốt nghiệp cho mình.

Trong quá trình thực hiện đề tài, có thể còn nhiều thiếu sót do kiến thức hạn chế nhưng những nội dung trình bày trong quyển báo cáo này là những hiểu biết và thành quả của chúng em đạt được dưới sự hướng dẫn của thầy **Nguyễn Hoàng Dũng**.

Chúng em xin cam đoan rằng: những nội dung trình bày trong quyển báo cáo luận văn tốt nghiệp này không phải là bản sao chép từ bất kỳ công trình đã có trước nào. Nếu không đúng sự thật, chúng em xin chịu mọi trách nhiệm trước nhà trường.

*Cần Thơ, ngày 10 tháng 12 năm 2011*

Nhóm sinh viên thực hiện

**Nguyễn Thị Hồng Cẩm    Phạm Tiến Huy**

## **LỜI CẢM ƠN**

Với tấm lòng tôn sư trọng đạo, chúng em xin chân thành cảm ơn quý thầy cô đã tận tình dạy bảo chúng em, và đã truyền đạt cho chúng em những kiến thức quý báu để làm hành trang cho chúng em bước vào đời.

Xin chân thành cảm ơn các thầy cô trường Đại học Cần Thơ đã tận tình chỉ cho chúng em trong suốt khóa học vừa qua.

Xin cảm ơn thầy Nguyễn Hoàng Dũng đã tận tình hướng dẫn và cung cấp cho chúng em tài liệu, thiết bị để hoàn thành luận văn này, cũng như truyền thụ những kiến kinh nghiệm quý báu trong suốt thời gian thực hiện đề tài.

Xin ghi nhớ công ơn của cha mẹ đã không quản mọi gian lao khổ nhọc và cả sự hy sinh cao cả để con được như ngày hôm nay.

Một lần nữa xin gửi đến những người thân yêu, các bạn, các anh chị...Đã đóng góp ý kiến giúp đỡ về mặt tinh thần, kinh nghiệm, kiến thức...một lời biết ơn sâu sắc nhất.

Xin chân thành cảm ơn

**Nguyễn Thị Hồng Cẩm  
Phạm Tiến Huy**

## MỤC LỤC

<b>TỔNG QUAN VỀ ĐỀ TÀI.....</b>	<b>1</b>
I. ĐẶT VẤN ĐỀ .....	1
II. MỤC TIÊU NGHIÊN CỨU .....	1
III. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ .....	1
IV. PHẠM VI NGHIÊN CỨU .....	2
V. HƯỚNG GIẢI QUYẾT VẤN ĐỀ .....	2
VI. NỘI DUNG LUẬN VĂN .....	2
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....</b>	<b>3</b>
1.1 GIỚI THIỆU PHẦN MỀM TOTALLY INTEGRATED AUTOMATION PORTAL .....	3
1.2 SIMATIC STEP 7 BASIC.....	3
1.2.1 Cách tạo và quản lý project .....	3
1.2.2 Cấu hình cho thiết bị .....	8
1.2.3 Thiết lập truyền thông giữa PLC và các thiết bị ngoài .....	9
1.3 SIMATIC WINCC BASIC V10.5 .....	18
1.3.1 Giới thiệu.....	18
1.3.2 Các thành phần soạn thảo .....	18
1.3.3 Các chức năng cơ bản của WinCC v10.5.....	20
1.3.4 Thiết kế giao diện cho màn hình HMI .....	21
1.4 PC ACCESS .....	29
1.4.1 Giới Thiệu .....	29
1.4.2 Kết Nối Giữa PC Access Và PLC S7-1200 .....	29
<b>CHƯƠNG 2 CÁC PHƯƠNG PHÁP LẬP TRÌNH PLC.....</b>	<b>34</b>
2.1 LẬP TRÌNH .....	34
2.2 CÁC PHƯƠNG PHÁP LẬP TRÌNH PLC.....	34
2.2.1 Lập trình tuyến tính .....	34
2.2.2 Lập trình cấu trúc.....	36
2.2.3 Lập trình ngẫu nhiên .....	38
<b>CHƯƠNG 3 TỔNG QUAN VỀ PLC VÀ BỘ PLC S7-1200 .....</b>	<b>41</b>
3.1 TỔNG QUAN VỀ PLC .....	41
3.1.1 Giới Thiệu Về PLC .....	41
3.1.2 Ngôn Ngữ Lập Trình PLC .....	43
3.2 TỔNG QUAN VỀ HỘ PLC S7-1200 .....	44
3.2.1 Giới thiệu.....	44
3.2.2 Ứng dụng của PLC S7-1200 .....	45
3.2.3 Cấu Trúc Phần Cứng .....	46
3.2.4 Các kiểu dữ liệu và truy xuất dữ liệu trong S7-1200 .....	69
3.3 TẬP LỆNH CỦA PLC S7-1200 .....	73
3.3.1 Nhóm lệnh về bit .....	73
3.3.2 Nhóm lệnh định thời .....	86
3.3.3 Nhóm lệnh đếm .....	90
3.3.4 Nhóm lệnh so sánh.....	96
3.3.5 Nhóm lệnh toán học .....	101
3.3.6 Nhóm lệnh di chuyển .....	112
3.3.7 Nhóm lệnh chuyển đổi .....	120
3.3.8 Nhóm lệnh điều khiển chương trình .....	126
3.3.9 Nhóm lệnh toán học .....	127
3.3.10 Nhóm lệnh dịch và quay .....	136
3.3.11 Lệnh vòng lặp PID.....	139
3.3.12 Lệnh đếm xung tốc độ cao.....	149

3.3.13	Lệnh Ngắt.....	156
<b>CHƯƠNG 4: LẬP TRÌNH ỨNG DỤNG PLC S7-1200 .....</b>	<b>160</b>	
4.1	GIỚI THIỆU.....	160
4.2	LẬP TRÌNH ỨNG DỤNG ĐIỀU KHIỂN NHIỆT ĐỘ .....	161
4.2.1	Giới Thiệu .....	161
4.2.2	Sơ đồ khối.....	161
4.2.3	Kết nối phần cứng .....	161
4.2.4	Sơ đồ đấu dây .....	163
4.2.5	Lưu đồ giải thuật và chương trình.....	164
4.2.6	Kết quả.....	164
4.3	LẬP TRÌNH ỨNG DỤNG ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ AC .....	164
4.3.1	Giới thiệu.....	164
4.3.2	Sơ đồ khối.....	165
4.3.3	Kết nối phần cứng .....	165
4.3.4	Mô hình phần cứng .....	170
4.3.5	Lưu đồ giải thuật và chương trình.....	171
4.3.6	Kết quả.....	171
<b>KẾT LUẬN VÀ ĐỀ NGHỊ .....</b>	<b>172</b>	
I.	Kết quả.....	172
II.	Hạn chế .....	172
III.	Hướng phát triển.....	172
<b>PHỤ LỤC .....</b>	<b>174</b>	
<b>PHỤ LỤC 1 .....</b>	<b>174</b>	
<b>CHƯƠNG TRÌNH ĐIỀU KHIỂN TRÊN PLC .....</b>	<b>174</b>	
<b>PHỤ LỤC 2 .....</b>	<b>183</b>	
<b>CHƯƠNG TRÌNH ĐIỀU KHIỂN VÀ GIÁM SÁT .....</b>	<b>183</b>	
<b>PHỤ LỤC 3 .....</b>	<b>188</b>	
<b>GIÁM SÁT VÀ ĐIỀU KHIỂN HỆ THỐNG .....</b>	<b>188</b>	
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>193</b>	

## DANH MỤC BIỂU HÌNH

Hình 1.1: Giao diện phần mềm lập trình TIA Portal .....	4
Hình 1.2: Cửa sổ chọn lập trình.....	5
Hình 1.3: Add thiết bị màn hình HMI .....	5
Hình 1.4: Add thiết bị màn hình HMI .....	6
Hình 1.5: Cửa sổ làm việc của phần mềm TIA Portal .....	7
Hình 1.6: Chọn khối Main để vào cửa sổ làm việc .....	8
Hình 1.7: Chọn PLC để cấu hình .....	8
Hình 1.8: Giao diện cấu hình phần cứng PLC .....	9
Hình 1.9: Kết nối trực tiếp: HMI kết nối với CPU S7-1200.....	10
Hình 1.10: Các thiết bị được kết nối với nhau qua cổng PROFINET .....	10
Hình 1.11: Truy xuất các thiết bị đã kết nối với máy tính .....	11
Hình 1.12: Chọn chuẩn kết nối .....	11
Hình 1.13: Kết nối thành công.....	12
Hình 1.14: Cho phép đặt địa chỉ IP .....	12
Hình 1.15: Chọn download chương trình xuống PLC .....	13
Hình 1.16: Chọn chế độ Stop để download chương trình xuống PLC .....	13
Hình 1.17: Download chương trình xuống PLC .....	14
Hình 1.18: Chương trình đang cấu hình PLC .....	14
Hình 1.19: Hoàn thành download chương trình .....	15
Hình 1.20: Nạp chương trình và cấu hình thiết bị thành công.....	16
Hình 1.21: Giao diện hiển thị lỗi cấu hình.....	17
Hình 1.22: Reset địa chỉ IP .....	17
Hình 1.23: Add thiết bị màn hình HMI .....	21
Hình 1.24: Đặt tên và chọn thiết bị HMI .....	22
Hình 1.25: Cấu hình kết nối HMI và PLC .....	23
Hình 1.26: Chọn đặc tính cho màn hình HMI .....	23
Hình 1.27: Tạo một đồng hồ trên màn hình HMI .....	24
Hình 1.28: Đặt các đặc tính cho màn hình hiển thị .....	24
Hình 1.29: Cài đặt một hệ thống màn hình HMI .....	25
Hình 1.30: Chọn nút nhấn.....	26
Hình 1.31: Giao diện lập trình cho thiết bị HMI .....	27
Hình 1.32 Mô phỏng thời gian chạy .....	28
Hình 1.33: Tạo một project mới cho PC Access .....	30
Hình 1.34: Giao diện PC Access .....	30
Hình 1.35: Tạo liên kết giữa PLC và PC Access .....	31
Hình 1.36: Thiết lập các thuộc tính cho PC Access.....	32
Hình 1.37 Kiểm tra Item thông qua Test Client .....	33
Hình 2.1 Lưu đồ giải thuật chương trình rửa xe tự động .....	36
Hình 2.2 Lưu đồ giải thuật chương trình giữ xe tự động .....	38
Hình 2.3 Sơ đồ các trạng thái điều khiển bóng đèn bằng nút nhấn .....	39
Hình 3.1 PLC Siemens.....	42
Hình 3.2 PLC loại Omron.....	43
Hình 3.3 PLC loại Panasonic .....	43
Hình 3.4: Hình dạng của PLC S7-1200 và các module mở rộng .....	44
Hình 3.5: Hình dạng bên ngoài của PLC S7-1200 (CPU 1212C) .....	47
Hình 3.6: Cấu trúc bên trong PLC .....	49
Hình 3.7: Sơ đồ đấu dây PLC S7-1200 / CPU 1212C .....	51
Hình 3.8: Sơ đồ đấu dây PLC S7-1200 / CPU 1212C .....	52
Hình 3.9: Sơ đồ đấu dây SB 1223 DC/DC .....	54

Hình 3.10: Sơ đồ đấu dây SB 1232AQ .....	56
Hình 3.11: Sơ đồ đấu dây SB 1232AQ .....	56
Hình 3.12: Sơ đồ đấu dây SM 1221 .....	58
Hình 3.13: Sơ đồ đấu dây SM 1222 .....	61
Hình 3.14: Sơ đồ đấu dây SM 1223 .....	63
Hình 3.15: Sơ đồ đấu dây của SM 1231, SM 1232, SM 1234 .....	65
Hình 3.16: Module Communication .....	66
Hình 3.17: Thẻ nhớ .....	68
Hình 3.18: Màn hình HMI .....	69
Hình 3.19: Lệnh tiếp điểm thường mở NO .....	73
Hình 3.20: Ví dụ về lệnh tiếp điểm thường mở .....	73
Hình 3.21: Lệnh tiếp điểm thường đóng NC .....	74
Hình 3.22: Ví dụ về lệnh tiếp điểm thường đóng .....	74
Hình 3.23: Lệnh đảo bit .....	74
Hình 3.24: Lệnh đảo bit trong ngôn ngữ FBD .....	75
Hình 3.25: Ví dụ về lệnh tiếp điểm thường đóng .....	75
Hình 3.26: Lệnh AND, OR, XOR bit .....	75
Hình 3.27: Lệnh AND, OR, XOR bit trong ngôn ngữ LAD .....	76
Hình 3.28: Lệnh ngõ ra ngôn ngữ LAD .....	76
Hình 3.29: Lệnh ngõ ra đảo ngôn ngữ LAD .....	77
Hình 3.30: Ví dụ về lệnh ngõ ra đảo ngôn ngữ LAD .....	77
Hình 3.31: Lệnh SET và RESET bit .....	77
Hình 3.32: Ví dụ về lệnh SET và RESET bit .....	78
Hình 3.33: Lệnh SET và RESET một vùng bit .....	78
Hình 3.34: Ví dụ về lệnh SET và RESET một vùng bit .....	79
Hình 3.35: Lệnh ưu tiên SET và ưu tiên RESET .....	79
Hình 3.36: Ví dụ lệnh ưu tiên SET .....	80
Hình 3.37: Ví dụ lệnh ưu tiên RESET .....	80
Hình 3.38: Lệnh cạnh lên .....	81
Hình 3.39: Ví dụ về lệnh cạnh lên .....	81
Hình 3.40: Lệnh cạnh xuống .....	82
Hình 3.41: Ví dụ về lệnh cạnh xuống .....	82
Hình 3.42: Lệnh cạnh lên ngõ ra .....	82
Hình 3.43: Ví dụ về lệnh cạnh lên ngõ ra .....	83
Hình 3.45: Ví dụ về lệnh cạnh xuống ngõ ra .....	84
Hình 3.46: Lệnh cạnh lên xung CLK .....	85
Hình 3.47: Ví dụ về lệnh cạnh lên xung CLK .....	85
Hình 3.48: Lệnh cạnh xuống xung CLK .....	85
Hình 3.49: Ví dụ về lệnh cạnh xuống xung CLK .....	85
Hình 3.50: Lệnh định thời TP .....	86
Hình 3.51: Biểu đồ thời gian của lệnh TP .....	87
Hình 3.52: Lệnh định thời TON .....	87
Hình 3.53: Biểu đồ thời gian của Timer On .....	87
Hình 3.54: Lệnh định thời TOF .....	88
Hình 3.55: Biểu đồ thời gian của Timer Off .....	88
Hình 3.56: Lệnh định thời TONR .....	89
Hình 3.57: Biểu đồ thời gian của Timer ON có nhớ .....	89
Hình 3.58: Lệnh đếm lên .....	90
Hình 3.59: Biểu đồ thời gian của lệnh đếm lên (Counter Up) .....	91
Hình 3.60: Ví dụ về lệnh đếm lên .....	91
Hình 3.61: Lệnh đếm xuống .....	92
Hình 3.62: Biểu đồ thời gian của lệnh đếm xuống .....	92

Hình 3.63: Ví dụ về lệnh đếm xuống.....	93
Hình 3.64: Lệnh đếm lên-xuống .....	93
Hình 3.65: Lệnh đếm lên-xuống .....	94
Hình 3.66: Ví dụ về lệnh đếm lên-xuống.....	95
Hình 3.67: Lệnh so sánh bằng kiểu Byte .....	96
Hình 3.68: Ví dụ lệnh so sánh bằng_ kiểu Byte .....	97
Hình 3.69: Lệnh In-range.....	97
Hình 3.70: Ví dụ về lệnh In-range .....	98
Hình 3.71: Lệnh Out-of-range .....	99
Hình 3.72: Ví dụ về lệnh Out-of-range.....	99
Hình 3.73: Lệnh OK và NOT_OK.....	100
Hình 3.74: Lệnh cộng .....	101
Hình 3.75: Ví dụ về lệnh cộng.....	102
Hình 3.76: Lệnh trừ .....	102
Hình 3.77: Ví dụ về lệnh trừ .....	103
Hình 3.78: Lệnh nhân .....	103
Hình 3.79: Ví dụ về lệnh nhân.....	104
Hình 3.80: Lệnh chia .....	104
Hình 3.81: Ví dụ về lệnh chia .....	105
Hình 3.82: Lệnh chia có dư.....	105
Hình 3.83: Lệnh phủ định .....	106
Hình 3.84: Ví dụ về lệnh phủ định.....	107
Hình 3.85: Lệnh tăng và giảm .....	107
Hình 3.86: Ví dụ về lệnh tăng và giảm .....	108
Hình 3.87: Lệnh trị tuyệt đối .....	108
Hình 3.88: Lệnh giá trị lớn nhất và giá trị nhỏ nhất .....	109
Hình 3.89: Ví dụ về lệnh Max .....	110
Hình 3.90: Lệnh giới hạn .....	111
Hình 3.91: Lệnh di chuyển MOVE .....	112
Hình 3.92: Ví dụ về lệnh di chuyển MOVE .....	113
Hình 3.93: Lệnh move_block .....	113
Hình 3.94: Lệnh umove_blk .....	114
Hình 3.95: lệnh FILL_BLK .....	116
Hình 3.96: Ví dụ về lệnh FILL_BLK .....	116
Hình 3.97: Lệnh UFILL_BLK.....	117
Hình 3.98: Lệnh hoán đổi Swap .....	118
Hình 3.99: Ví dụ về lệnh hoán đổi Swap .....	119
Hình 3.100: Lệnh CONV .....	120
Hình 3.101: Ví dụ về lệnh CONV .....	121
Hình 3.102: Lệnh Round và Truncate .....	121
Hình 3.103: Lệnh CEIL và FLOOR .....	122
Hình 3.104: Lệnh SCARE_X và NORM_X .....	123
Hình 3.106: Ví dụ về lệnh NORM_X .....	125
Hình 3.107: Lệnh nhảy .....	126
Hình 3.108: Ví dụ về lệnh nhảy .....	127
Hình 3.109: Lệnh RET .....	127
Hình 3.110: Lệnh AND .....	127
Hình 3.111: Lệnh AND .....	129
Hình 3.112: Lệnh đảo .....	129
Hình 3.114: Lệnh giải mã .....	131
Hình 3.115: Ví dụ về lệnh giải mã .....	132
Hình 3.116 Lệnh chọn .....	133

Hình 3.117: Lệnh đa hợp .....	134
Hình 3.120 Lệnh quay .....	138
Hình 3.121: Mô hình một hệ thống điều khiển bằng PID.....	140
Hình 3.122 Sơ đồ khối bộ điều khiển PID.....	140
Hình 3.123: Lệnh PID_Compact .....	142
Hình 3.124: Tạo lệnh vòng lặp PID .....	145
Hình 3.125: Cấu hình cho bộ PID.....	146
Hình 3.126: Cài đặt chế độ hoạt động cho bộ điều khiển.....	147
Hình 3.127: Khối CTRL_HSC .....	149
Hình 3.128: Thêm bộ CTRL_HSC vào chương trình.....	151
Hình 3.129: Đặt tên cho bộ CTRL_HSC .....	151
Hình 3.130: Thiết lập các thông số cho khói CTRL_HSC .....	152
Hình 3.131: Chọn địa chỉ phần cứng cho khói CTRL_HSC .....	153
Hình 3.132: Cấu hình phần cứng HSC .....	154
Hình 3.133: Lệnh ATTACH và DETACH .....	156
Hình 3.134: Tạo khói OB cho lệnh ngắn .....	157
Hình 3.135: Lệnh STR_DINT và CAN_DINT .....	158
Hình 3.136: Lệnh STR_DINT và CAN_DINT .....	159
Hình 4.1: Mô hình thực tế của hệ thống .....	160
Hình 4.2: Sơ đồ hoạt động của bộ điều khiển .....	161
Hình 4.3: Cảm biến nhiệt độ LM 35 .....	162
Hình 4.4: Thiết bị gia nhiệt .....	162
Hình 4.5: Sơ đồ đấu dây các thiết bị vào PLC .....	163
Hình 4.6: Lưu đồ chương trình đo và điều khiển nhiệt độ.....	164
Hình 4.7: Nguyên lý hoạt động của bộ điều khiển tốc độ động cơ.....	165
Hình 4.8: Lắp module tương tự vào PLC .....	166
Hình 4.9: Sơ đồ nguyên lý của biến tần .....	167
Hình 4.10: Motor AC ba pha .....	168
Hình 4.11: Sơ đồ đấu dây motor AC ba pha với biến tần .....	169
Hình 4.12: Hình ảnh encoder thực .....	169
Hình 4.13: Sơ đồ mạch cách ly hai điện thế .....	170
Hình 4.14: Sơ đồ mạch cách ly hai điện thế .....	170
Hình 4.15: Sơ đồ mạch cách ly hai điện thế .....	171
Hình 1: Chương trình giám sát và điều khiển trên HMI.....	183
Hình 2: Giao diện mặc định sau khởi động .....	184
Hình 3: Giao diện chính .....	185
Hình 4: Giao diện PID giám sát và điều khiển nhiệt độ .....	186
Hình 5: Giao diện PID giám sát và điều khiển tốc độ động cơ .....	187
Hình 6: Điều khiển giám sát nhiệt độ .....	188
Hình 7: Giá trị nhiệt độ xác lập .....	189
Hình 8: Giá trị nhiệt độ ổn định ở nhiệt độ đặt .....	189
Hình 9: Điều khiển giám sát tốc độ động cơ .....	190
Hình 10: Giá trị tốc độ xác lập .....	191
Hình 11: Giá trị nhiệt độ ổn định ở nhiệt độ đặt .....	191
Hình 12: Thay đổi giá trị đặt .....	192

## DANH MỤC BIỂU BẢNG

Bảng 2.1: Bảng thông kê trạng thái điều khiển bóng đèn.....	40
Bảng 3.1: Thông số kỹ thuật cơ bản của PLC .....	46
Bảng 3.2: Thông số kỹ thuật của các module mở rộng .....	52
Bảng 3.3: Các thông số kỹ thuật của board tín hiệu Digital .....	53
Bảng 3.4: Các thông số kỹ thuật của board tín hiệu Analog .....	54
Bảng 3.5: Các thông số của SM 1221 ngõ vào Digital.....	57
Bảng 3.6: Các thông số kỹ thuật của SM 1222 ngõ ra Digital .....	58
Bảng 3.7: Thông số kỹ thuật của SM 1223 ngõ vào/ngõ ra digital .....	61
Bảng 3.8: Thông số kỹ thuật của SM 1231, SM 1232, SM 1234.....	63
Bảng 3.9: Thông số kỹ thuật của CM 1241 RS 485 .....	66
Bảng 3.10: Chức năng của các chân RS485 .....	67
Bảng 3.11: Thông số kỹ thuật của CM 1241 RS 232 .....	67
Bảng 3.12: Thông số kỹ thuật của màn hình HMI.....	69
Bảng 3.13: Các vùng nhớ trong PLC S7-1200 .....	70
Bảng 3.14: Bảng thông số kiểu dữ liệu Bool, Byte, Word, DWord .....	71
Bảng 3.15: Bảng thông số kiểu dữ liệu Integer .....	72
Bảng 3.16: Bảng thông số kiểu dữ liệu Floating-point real .....	72
Bảng 3.18: Các tham số của lệnh cạnh lên .....	81
Bảng 3.19: Các tham số của lệnh cạnh lên ngõ ra .....	83
Bảng 3.20: Các tham số của lệnh cạnh lên xung CLK .....	84
Bảng 3.21: Các thông số của bộ định thời PT .....	86
Bảng 3.22: Các thông số của bộ định thời TONR .....	89
Bảng 3.23: Các thông số của lệnh đếm lên (Counter Up) .....	90
Bảng 3.24: Thông số của lệnh đếm xuống (Counter Down) .....	92
Bảng 3.25: Thông số của lệnh đếm lên-xuống .....	94
Bảng 3.26: Thông số của lệnh so sánh kiểu Byte .....	96
Bảng 3.27: Một số lệnh so sánh kiểu Byte khác .....	97
Bảng 3.28: Thông số của lệnh so sánh kiểu Byte .....	98
Bảng 3.30: Thông số của lệnh cộng.....	101
Bảng 3.31: Thông số của lệnh phủ định .....	106
Bảng 3.32. Một vài trường hợp có thể xảy ra trong phép tính.....	106
Bảng 3.33: Thông số của lệnh .....	107
Bảng 3.34: Thông số của lệnh trị tuyệt đối .....	108
Bảng 3.35: Thông số của lệnh giá trị lớn nhất và nhỏ nhất .....	109
Bảng 3.36: Một vài trường hợp có thể xảy ra trong phép tính .....	110
Bảng 3.37: Thông số của lệnh giới hạn .....	111
Bảng 3.38: Thông số của lệnh .....	112
Bảng 3.39: Thông số của lệnh Move_block .....	113
Bảng 3.40: Trạng thái của ENO sau của lệnh MOVE. ....	115
Bảng 3.41: Thông số của lệnh Fill .....	116
Bảng 3.43 Thông số của lệnh hoán đổi.....	119
Bảng 3.44: Thông số của lệnh .....	120
Bảng 3.45: Một vài trường hợp có thể xảy ra trong phép tính .....	121
Bảng 3.46: Thông số của lệnh Round và lệnh Truncate .....	122
Bảng 3.47: Một vài trường hợp có thể xảy ra trong phép tính .....	122
Bảng 3.48: Thông số của lệnh SCARE và NORM.....	123
Bảng 3.49: Bảng trạng thái của ENO .....	124
Bảng 3.50: Thông số của lệnh AND.....	128
Bảng 3.51: Thông số của lệnh .....	129

Bảng 3.52: Thông số của lệnh mã hóa.....	130
Bảng 3.53: Một vài trường hợp có thể xảy ra khi thực hiện lệnh DECO .....	131
Bảng 3.54: Ví dụ về lệnh DECO .....	132
Bảng 3.55: Thông số của lệnh SEL .....	133
Bảng 3.56: Thông số của lệnh đa hợp.....	134
Bảng 3.57: Thông số của lệnh dịch.....	136
Bảng 3.58: Ví dụ về lệnh dịch .....	138
Bảng 3.59: Thông số của lệnh quay.....	138
Bảng 3.60: Ví dụ về lệnh quay .....	139
Bảng 3.61: Sự ảnh hưởng của các thông số Kp, Kd, Ki lên hệ thống .....	141
Bảng 3.62: Các thông số PID theo phương pháp Ziegler–Nichols.....	142
Bảng 3.63: Các thông số của bộ PID_Compact.....	143
Bảng 3.64: Chế độ hoạt động của PID_Compact.....	147
Bảng 3.65: High-speed counter trong các CPU .....	149
Bảng 3.66: Thông số của khối CTRL_HSC .....	149
Bảng 3.67: Mã lỗi của bộ CTRL_HSC.....	152
Bảng 3.68: Các thông số của bộ đếm.....	154
Bảng 3.69: Thông số của lệnh ATTACH và DETACH .....	156
Bảng 3.70: Thông số của lệnh SRT_DINT.....	158
Bảng 3.71: Thông số của lệnh CAL_DINT .....	158
Bảng 3.72: Thông số của lệnh CAL_DINT .....	159

## **KÝ HIỆU VÀ VIẾT TẮT**

PLC: Programmable Logic Controller.

CPU: Central Processing Unit.

HMI: Human Machine Interface.

WinCC: Windows Control Center.

SCADA: Supervisory Control And Data Acquisition.

SB: Signal Board

SM: Signal Module

CM: Communication Module

LAD: Ladder Logic

FBD: Function Block Diagram

OB: Organization Block

DB: Data Block

I/O: Input/Output

DI/DO: Digital Input/Digital Output

## **Tóm Tắt**

Đề tài lập trình ứng dụng S7-1200 nhằm khai thác các tính năng của dòng PLC mới này. Chúng em đã tìm hiểu các chức năng của dòng PLC này và làm hai ứng dụng “ Điều khiển nhiệt độ và điều khiển tốc độ động cơ AC”. Dữ liệu từ cảm biến sẽ được đưa vào các ngõ vào của PLC S7-1200 để tiến hành xử lý, qua bộ điều khiển PID của PLC ta sẽ điều khiển được giá trị nhiệt độ và tốc độ động cơ AC. Trên máy tính, ta xây dựng một giao diện người máy (HMI) dựa trên phần mềm Totally Integrated Automation Portal v10.5 (bao gồm SIMATIC WinCC Basic v10.5) để điều khiển và giám sát sự hoạt động của hệ thống.

## **Abstract**

Programming S7-1200 to exploit the features of this new PLC. We learn about its functions and make the applications "Control temperature and engine speed." Data from sensors will be fed into the input of the PLC S7-1200 to the process, the PID controller of the PLC will get the value of temperature and AC motor speed. We built a human-machine interface (HMI or Human Machine Interface) based software Totally Integrated Automation Portal v10.5 (including SIMATIC WinCC Basic v10.5) to control and monitor the activities of the system.

## LỜI MỞ ĐẦU

Ngày nay, các hoạt động sản xuất trên thế giới đang dần được tự động hóa. Lĩnh vực tự động hóa cũng đang phát triển mạnh mẽ ở Việt Nam. Việc ứng dụng các công nghệ mới vào trong sản xuất tự động là điều hết sức quan trọng. Trong hướng đó, chúng em đã chọn đề tài lập trình ứng dụng S7-1200 để thực hiện luận văn tốt nghiệp của mình. PLC S7-1200 là dòng mới nhất của hãng Siemens. Do đó việc nghiên cứu và ứng dụng dòng PLC này là rất cần thiết.

Chúng em đã chọn lập trình ứng dụng điều khiển nhiệt độ và lập trình ứng dụng điều khiển tốc độ động cơ trên dòng PLC S7-1200. Đây là hai ứng dụng quan trọng và phổ biến nhất trong hoạt động sản xuất.

Nội dung đề tài được tóm gọn trong bốn chương. Chương 1: Cơ sở lý thuyết. Chương 2: Các phương pháp lập trình PLC. Chương 3: Tổng quan về PLC và bộ PLC S7-1200. Chương 4: Lập trình ứng dụng PLC S7-1200.

Việc thực hiện xong luận văn này là cơ hội để vận dụng những kiến thức đã học, học hỏi những kiến thức mới và tích lũy thêm nhiều kinh nghiệm cho bản thân. Tuy nhiên, do kiến thức có hạn nên không thể tránh khỏi những sai sót trong quá trình thực hiện. Nên rất mong nhận được sự góp ý của Thầy cô và các bạn.

# **TỔNG QUAN VỀ ĐỀ TÀI**

## **I. ĐẶT VẤN ĐỀ**

Ngày nay, cùng với sự phát triển của công nghiệp thì việc chuyển đổi sang cơ chế sản xuất tự động và bán tự động trong các nhà máy xí nghiệp cũng là một điều tất yếu, việc điều khiển và vận hành trở nên đơn giản đi rất nhiều. Người vận hành chỉ việc điều khiển, giám sát và thu thập dữ liệu trực tiếp trên giao diện máy tính đã được lập trình trước. Điều đó mang lại hiệu quả cao trong sản xuất, và giảm đi nhiều chi phí và công sức.

Tập đoàn Siemens của Đức đã cho ra dòng PLC mới S7-1200 với thiết kế nhỏ gọn, nhiều tính năng ưu việt, đã được tích hợp nhiều module và tập lệnh mạnh so với PLC trước đây sẽ thúc đẩy sự phát triển của nhiều ngành công nghiệp, mà đặc biệt là điện tử - tự động.

Hệ thống điều khiển nhiệt độ và tốc độ động cơ AC là một ứng dụng nhỏ nhưng cũng thể hiện được những ưu điểm mà S7-1200 mang lại. Ta có thể điều khiển và giám sát hệ thống dễ dàng thông qua máy tính.

## **II. MỤC TIÊU NGHIÊN CỨU**

Mục tiêu chung:

- Ứng dụng PLC S7-1200 vào trong các hệ thống tự động.

Mục tiêu cụ thể:

- Khảo sát tập lệnh và tính năng của PLC S7-1200 để điều khiển đối tượng.
- Ứng dụng công cụ PID của PLC S7-1200 điều khiển nhiệt độ và tốc độ động cơ AC ba pha.

## **III. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ**

Hiện nay, các hệ thống điều khiển tự động sử dụng PLC rất phổ biến. Tuy nhiên, các hệ thống sử dụng dòng sản phẩm mới PLC S7-1200 là rất ít. Tháng 5, năm

2011 sinh viên Thái Văn Danh, lớp Cơ Điện Tử K32 khoa Công Nghệ, trường Đại học Cần Thơ đã thực hiện đề tài luận văn: “Điều Khiển PID Sử Dụng PLC S7-1200”. Luận văn này vẫn ứng dụng giải thuật PID vào trong điều khiển nhiệt độ và tốc độ động cơ. Điều này đã tạo điều kiện không nhỏ để chúng tôi có thể hoàn thành tốt luận văn của mình.

#### **IV. PHẠM VI NGHIÊN CỨU**

Khảo sát tập lệnh và tính năng của PLC S7-1200.

Sử dụng ngõ vào ra analog để đọc tín hiệu tương tự và điều khiển biến tần.

Làm mô hình hệ thống điều khiển nhiệt độ và mô hình điều khiển tốc độ động cơ AC ba pha.

Thiết kế giao diện HMI điều khiển giám sát và thu thập dữ liệu hai ứng dụng trên

#### **V. HƯỚNG GIẢI QUYẾT VẤN ĐỀ**

Tìm hiểu phần mềm lập trình Totally Integrated Automation Portal.

Khảo sát PLC S7-1200, cảm biến nhiệt độ LM35D, biến tần G110.

Thiết kế mô hình hệ thống điều khiển nhiệt độ và tốc độ động cơ AC.

Lập trình cho PLC và xây dựng giao diện điều khiển.

Điều khiển nhiệt độ: đối tượng điều khiển là thiết bị gia nhiệt, cảm biến nhiệt LM35D dùng để hồi tiếp nhiệt độ về cho bộ điều khiển, công cụ PID của S7-1200 là bộ điều khiển.

Điều khiển động cơ AC ba pha: Đối tượng điều khiển là động cơ AC ba pha được điều khiển gián tiếp qua biến tần, encoder là cảm biến nhằm hồi tiếp tốc độ về cho bộ điều khiển, công cụ PID của S7-1200 là bộ điều khiển.

Wincc được dùng để thiết kế giao diện HMI nhằm điều khiển giám sát và thu thập dữ liệu của hai ứng dụng trên.

## **VI. NỘI DUNG LUẬN VĂN**

Chương 1: Cơ Sở Lý Thuyết

Chương 2: Các Phương Pháp Lập Trình PLC

Chương 3: Tổng Quan Về PLC Và Bộ PLC S7-1200

Chương 4: Lập Trình Ứng Dụng PLC S7-1200

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1 GIỚI THIỆU PHẦN MỀM TOTALLY INTEGRATED AUTOMATION PORTAL

Totally Integrated Automation Portal bao gồm Simatic Step 7 Basic và Simatic WinCC Basic. Phần mềm này hỗ trợ lập trình cho bộ điều khiển PLC và màn hình điều khiển HMI.

### 1.2 SIMATIC STEP 7 BASIC

Simatic Step 7 Basic cung cấp một môi trường thân thiện với người dùng, từ hiệu chỉnh, thư viện, và bộ điều chỉnh logic cần thiết đến ứng dụng điều khiển.

Simatic Step 7 Basic cung cấp công cụ cho quản lý và cấu hình tất cả thiết bị trong project.

Simatic Step 7 Basic cung cấp hai ngôn ngữ lập trình (LAD và FBD), thích hợp và hiệu quả trong cải tiến lập trình điều khiển trong ứng dụng. Ngoài ra Simatic Step 7 Basic cung cấp công cụ tạo và cấu hình thiết bị HMI.

Simatic Tia Portal Step 7 Basic cung cấp một hệ thống trợ giúp trực tuyến và cung cấp 2 chế độ hiển thị khác nhau: a project-oriented view và a task-oriented set of portals.

Phần mềm Simatic Step7 Basic chạy trên hệ điều hành Windows, làm nhiệm vụ trung gian giữa người lập trình và PLC.

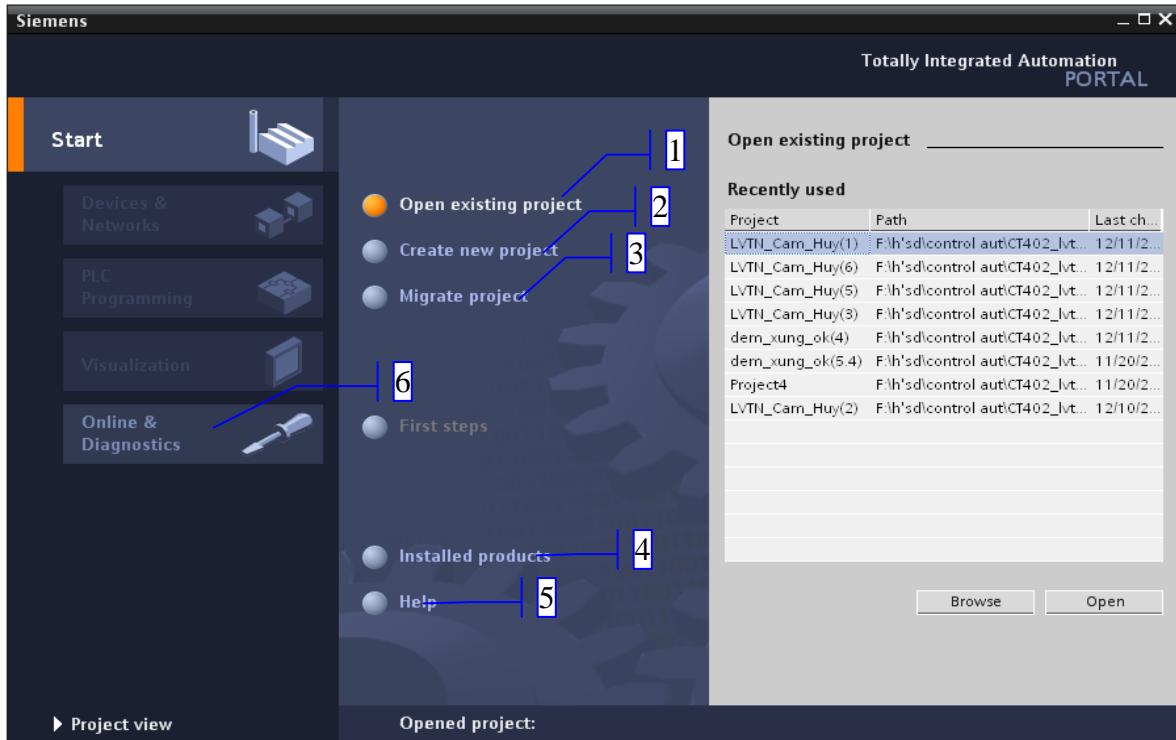
#### 1.2.1 Cách tạo và quản lý project

Để tạo một project mới ta thực hiện các bước sau:

- Mở phần mềm lập trình Totally Integrated Automation Portal bằng cách

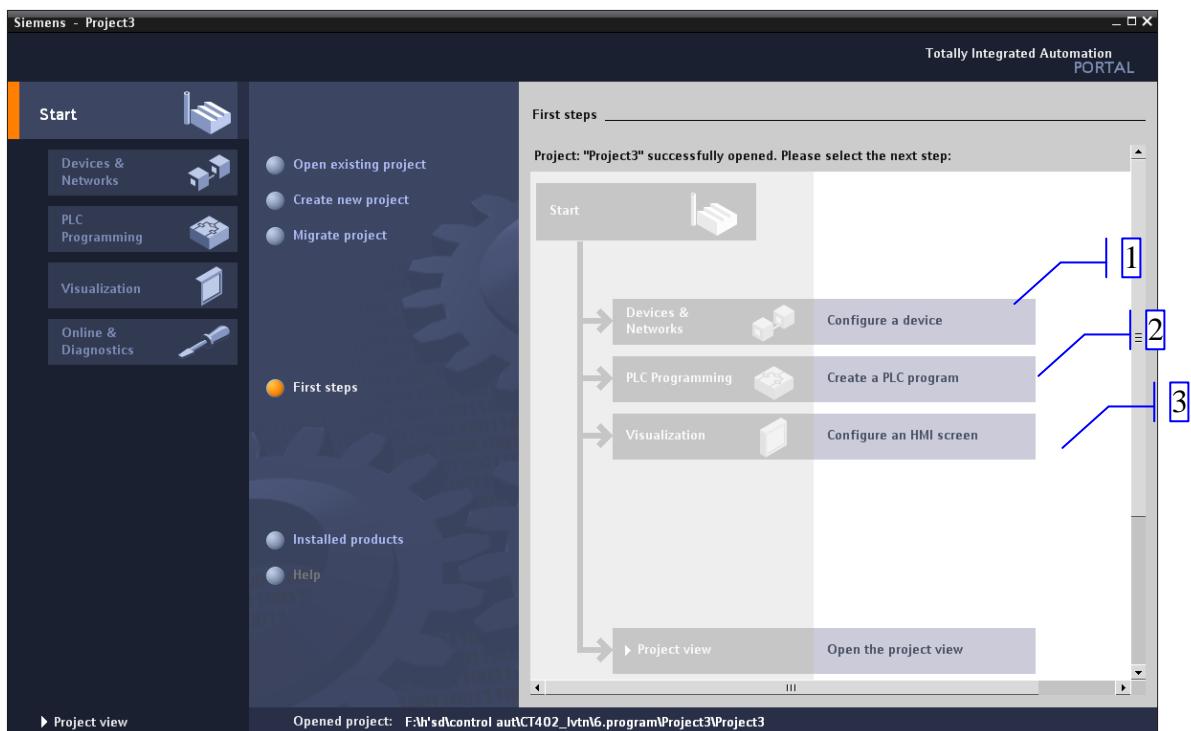
click đúp chuột vào biểu tượng  trên desktop hoặc vào Start/All Programs/Simens Automation/Integrated Automation Portal.

- Giao diện phần mềm lập trình như hình dưới:



**Hình 1.1: Giao diện phần mềm lập trình TIA Portal**

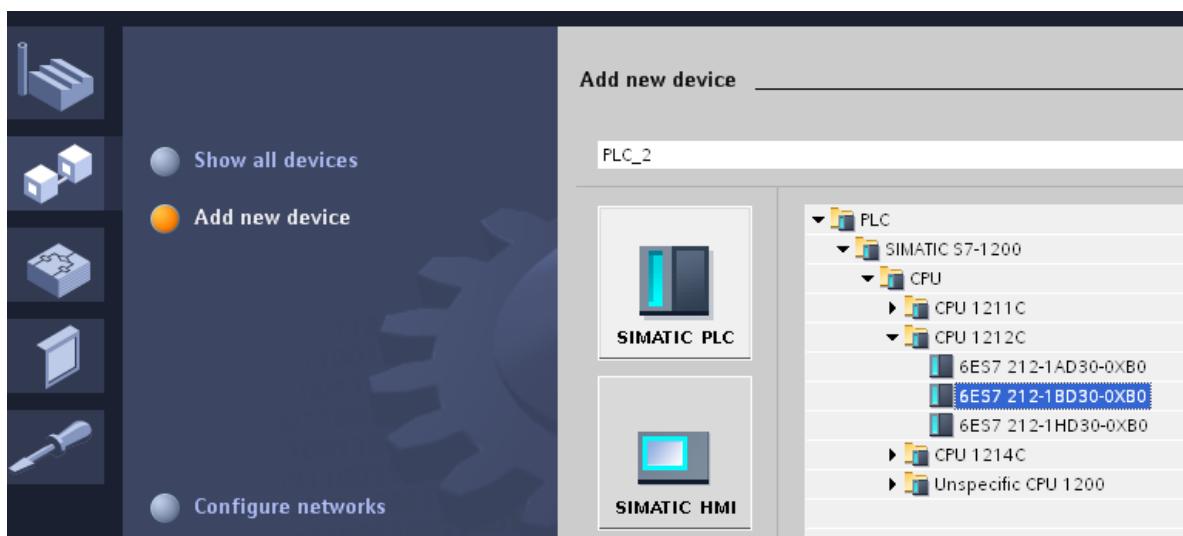
- Trên cửa sổ này ta có thể:
  1. Mở một Project đã có (Open existing project).
  2. Tạo một project mới (Create new project).
  3. Di chuyển một project ( Migrate project).
  4. Xem các phần mềm đã cài đặt (Installed products).
  5. Xem trợ giúp (Help).
  6. Kết nối trực tuyến chương trình với thiết bị (Online & Diagnostics/Acessible devices).
- Để tạo một project mới, ta chọn Start/ Create new project. Đặt tên và đường dẫn lưu project và chọn Create thì một giao diện mới hiện lên.



**Hình 1.2: Cửa sổ chọn lập trình**

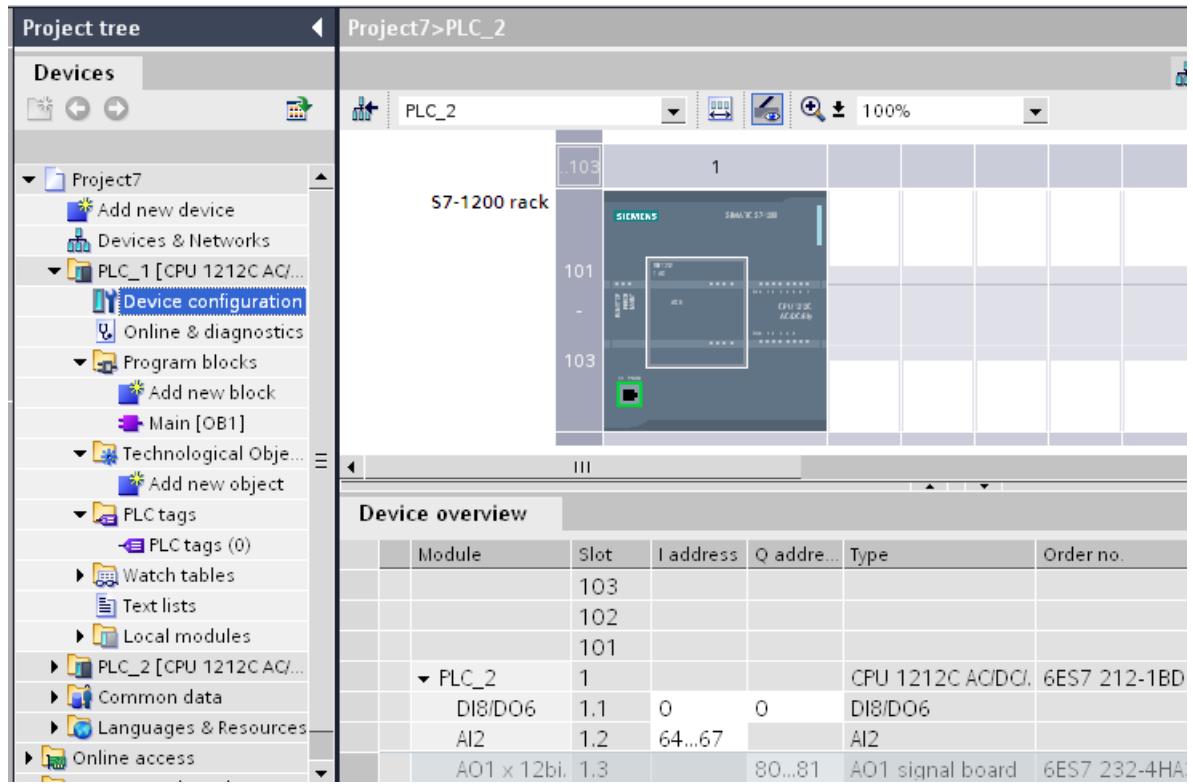
- Từ cửa sổ này, có hai cách tạo một chương trình để lập trình.
  - a. Cách 1: Cấu hình thiết bị trước và viết chương trình sau**

Chọn **Configure a device/Add device/PLC**



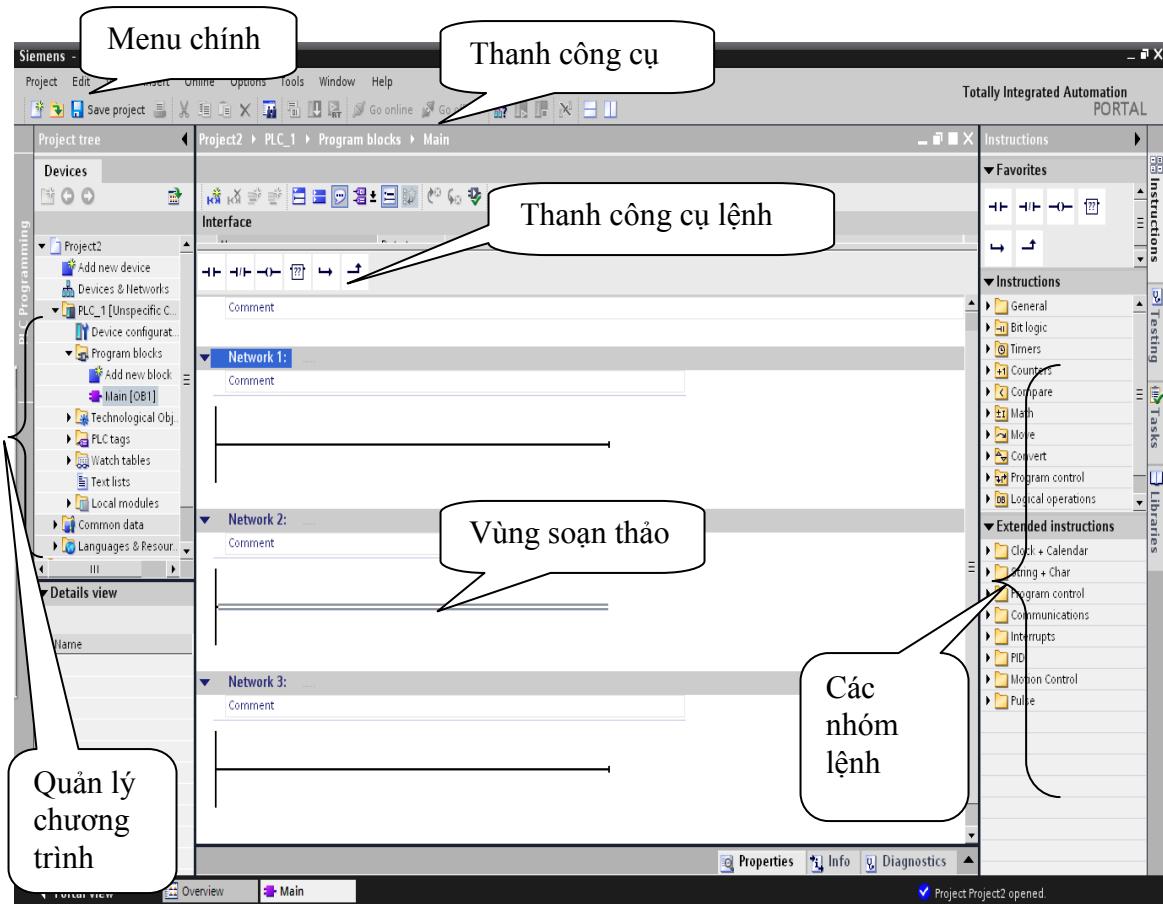
**Hình 1.3: Add thiết bị màn hình HMI**

Sau đó ta cấu hình cho thiết bị:



**Hình 1.4: Add thiết bị màn hình HMI**

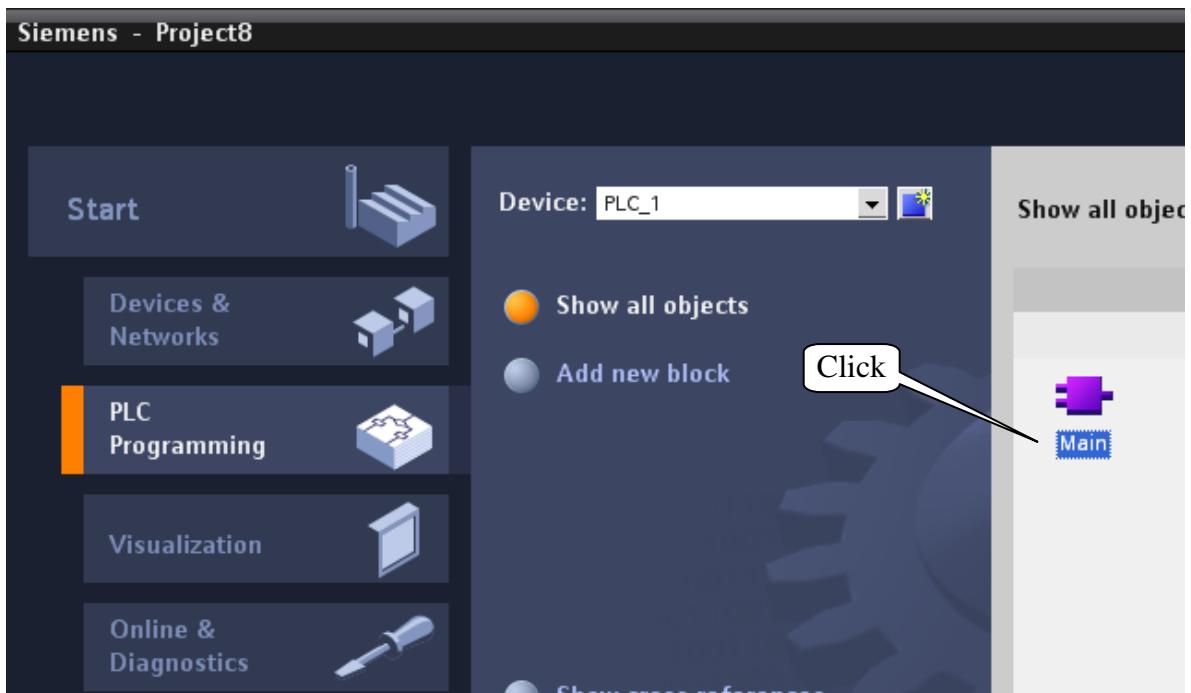
Sau khi cấu hình xong, ta viết chương trình phần mềm trên Main và các khối OB.



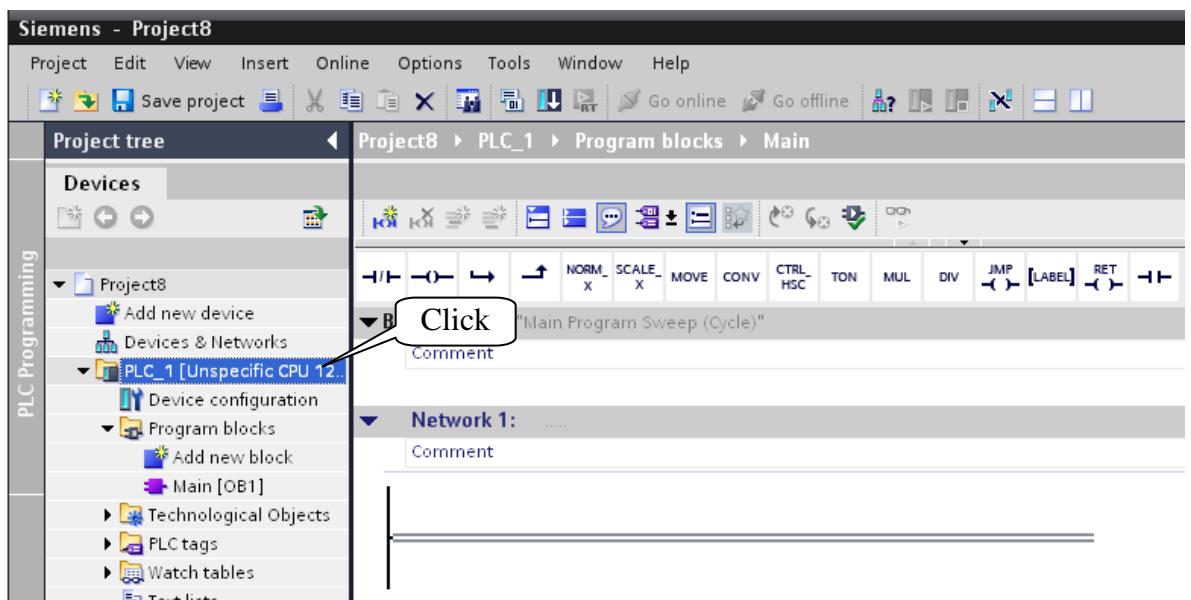
Hình 1.5: Cửa sổ làm việc của phần mềm TIA Portal

**b. Cách 2: Tạo và viết chương trình trước sau đó cấu hình các thiết bị sau:**

Chọn Create a Program PLC



**Hình 1.6: Chọn khối Main để vào cửa sổ làm việc  
Click vào khối Main để soạn thảo chương trình.**



**Hình 1.7: Chọn PLC để cấu hình**

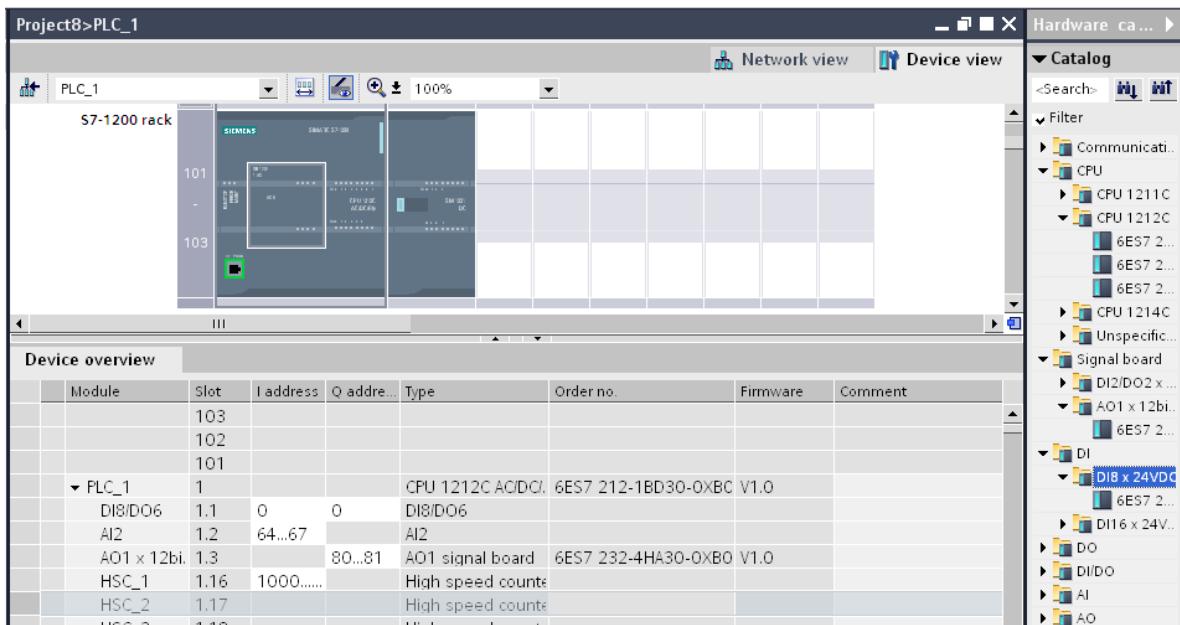
Sau đó vào Device configuration để cấu hình thiết bị.

### 1.2.2 Cấu hình cho thiết bị

Cấu hình cho thiết bị rất quan trọng. Nó định nghĩa phần cứng được sử dụng trong lập trình để tương thích với các lệnh đã lập trình. Mỗi thiết bị được đặt một tên

và một IP khác nhau. Ta add thêm các thiết bị cần sử dụng và kích hoạt nó. Nếu các thiết bị không được cấu hình hoặc cấu hình sai thì chương trình chúng ta không chạy được hoặc chạy sai.

Để cấu hình, ta có thể cấu hình trước khi viết chương trình hoặc sau khi soạn chương trình.



Hình 1.8: Giao diện cấu hình phần cứng PLC

### 1.2.3 Thiết lập truyền thông giữa PLC và các thiết bị ngoài

Các CPU S7-1200 có tích hợp một cổng PROFINET để hỗ trợ cả Ethernet và các tiêu chuẩn truyền thông TCP/IP chuẩn. PLC S7-1200 hỗ trợ các giao thức ứng dụng sau:

- Giao thức kết nối chuyển tải (TCP).
- ISO chuyển tải qua TCP (RFC 1006).

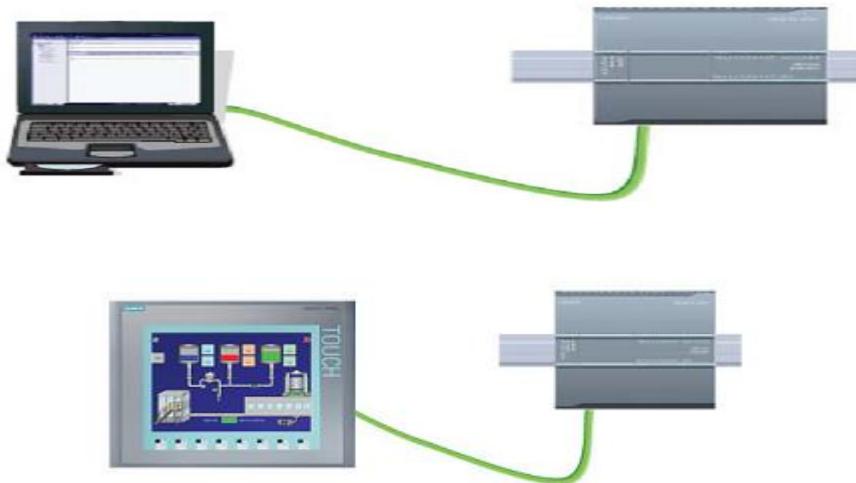
Các PLC S7-1200 có thể giao tiếp với các thiết bị sau:

- Các PLC S7-1200 khác.
- Thiết bị lập trình Step 7 Basic.
- Màn hình HMI.

- Các thiết bị không phải của Siemens có sử dụng chuẩn giao thức truyền thông TCP (khối truyền dẫn (T-block)).

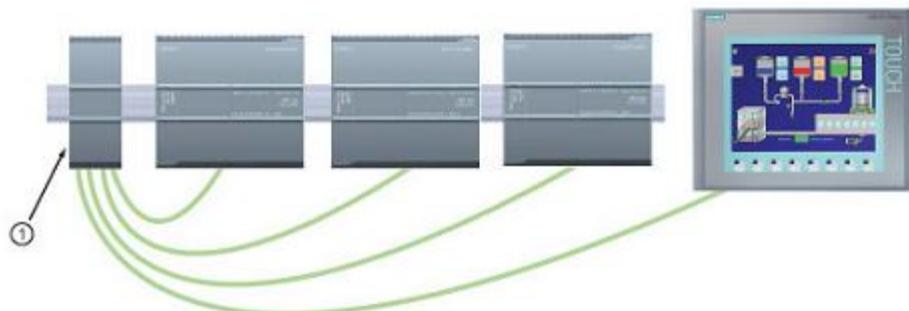
Có 2 cách để giao tiếp bằng cách sử dụng PROFIBUS:

- Kết nối trực tiếp: Sử dụng giao tiếp trực tiếp khi sử dụng thiết bị lập trình, HMI hoặc một PLC khác kết nối với một PLC đơn.



**Hình 1.9: Kết nối trực tiếp: HMI kết nối với CPU S7-1200**

- Kết nối qua mạng: Sử dụng giao tiếp mạng khi kết nối nhiều hơn hai thiết bị.



**Hình 1.10: Các thiết bị được kết nối với nhau qua cổng PROFINET**

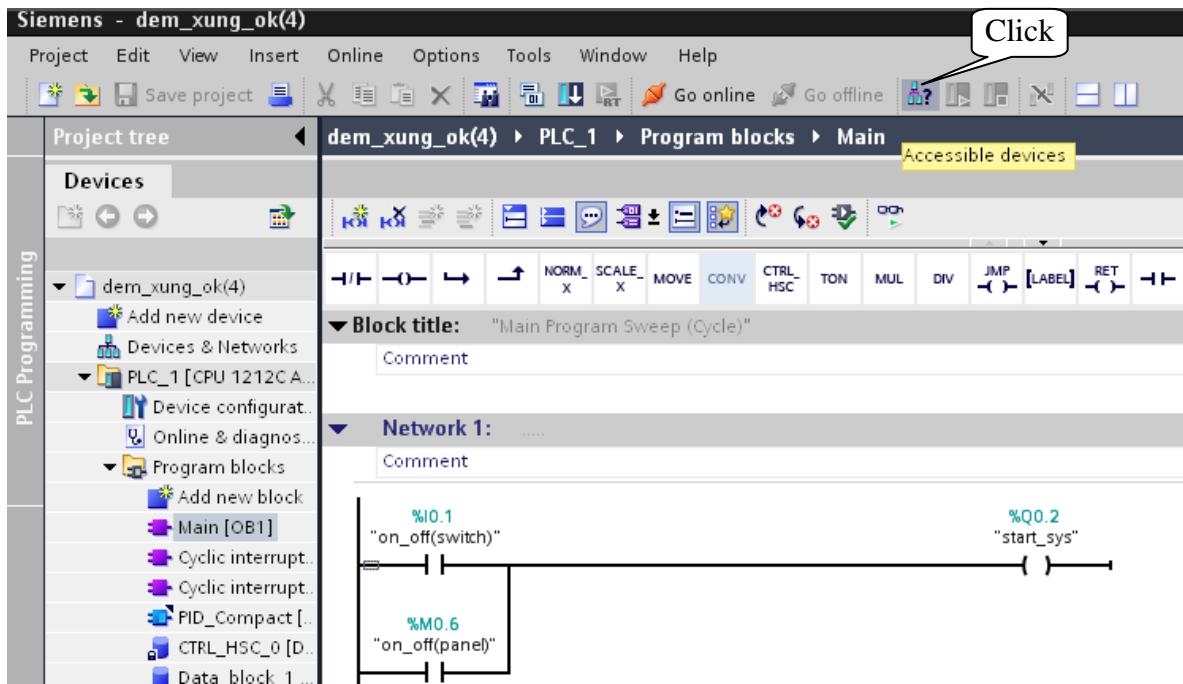
Trong đó: ① là thiết bị chuyển tiếp Ethernet CSM1277

#### **\* Kết nối CPU với máy tính:**

Sau khi soạn thảo chương trình xong, để download chương trình xuống máy tính, ta cần kết nối máy tính với PLC.

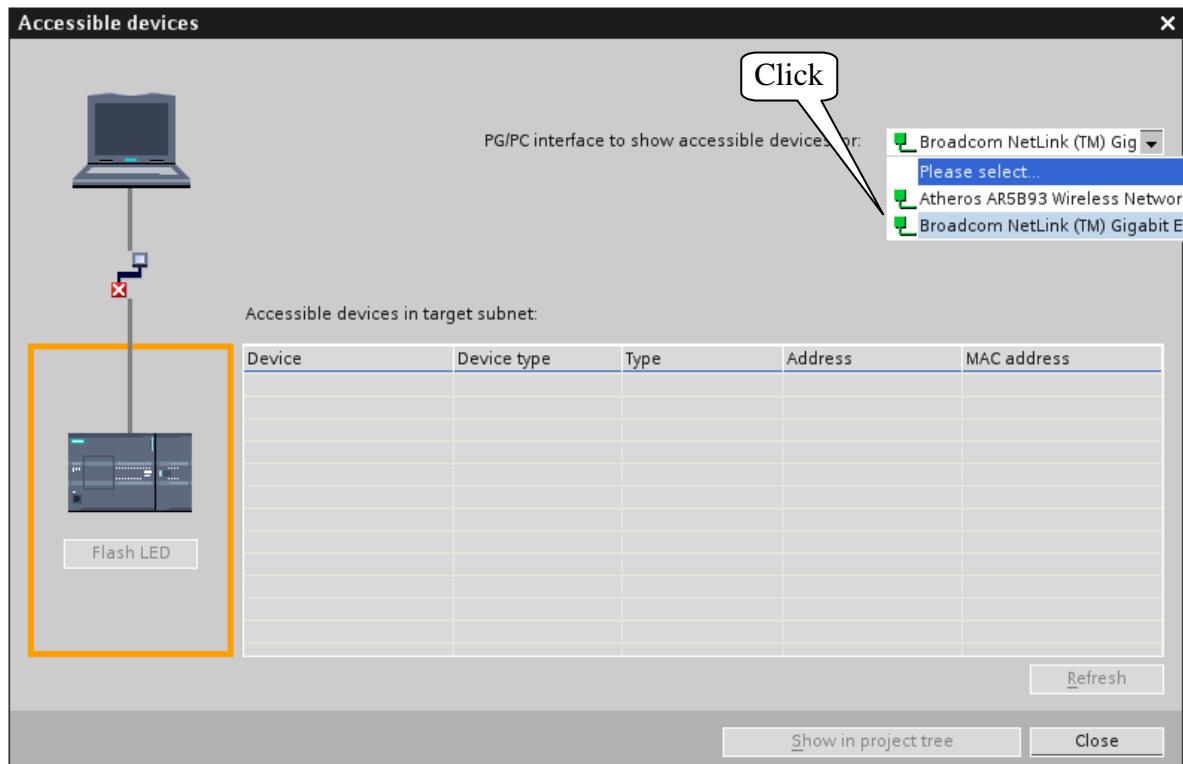
Trước tiên, kết nối cáp giữa PLC và máy tính như hình 1.9

Sau đó, ta chọn Accessible devices:



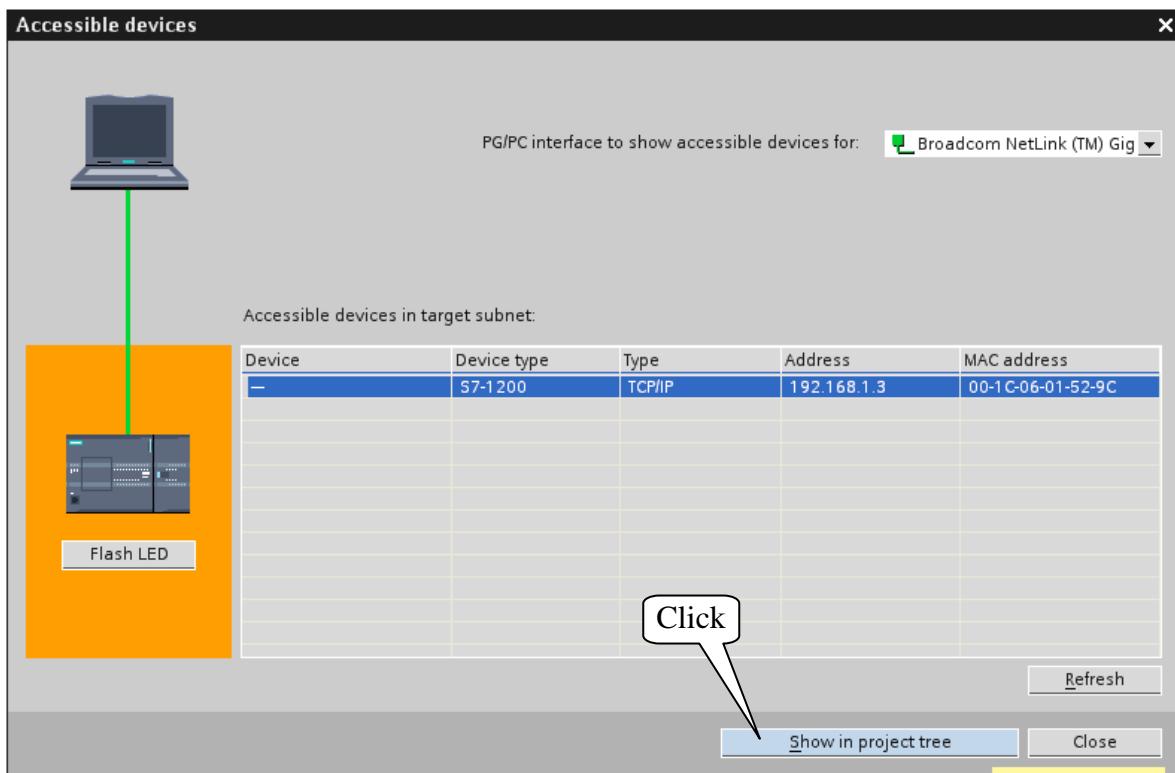
Hình 1.11: Truy xuất các thiết bị đã kết nối với máy tính

Chọn mạng giao tiếp:



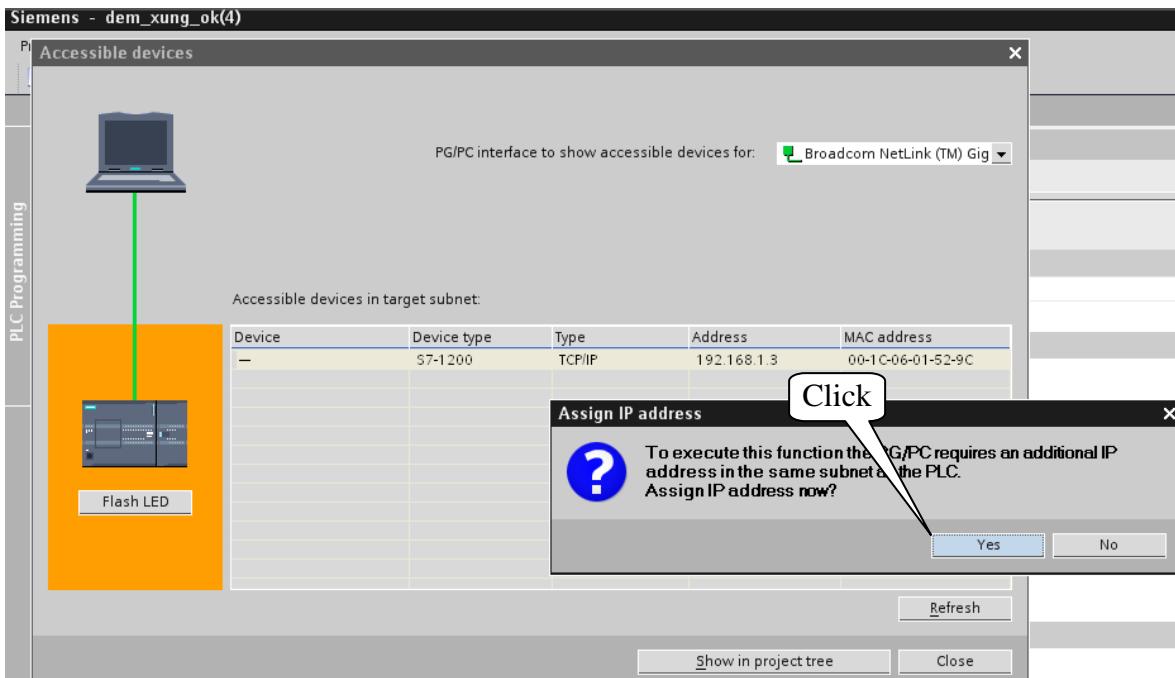
Hình 1.12: Chọn chuẩn kết nối

Nếu kết nối thành công, biểu tượng kết nối như hình bên dưới:



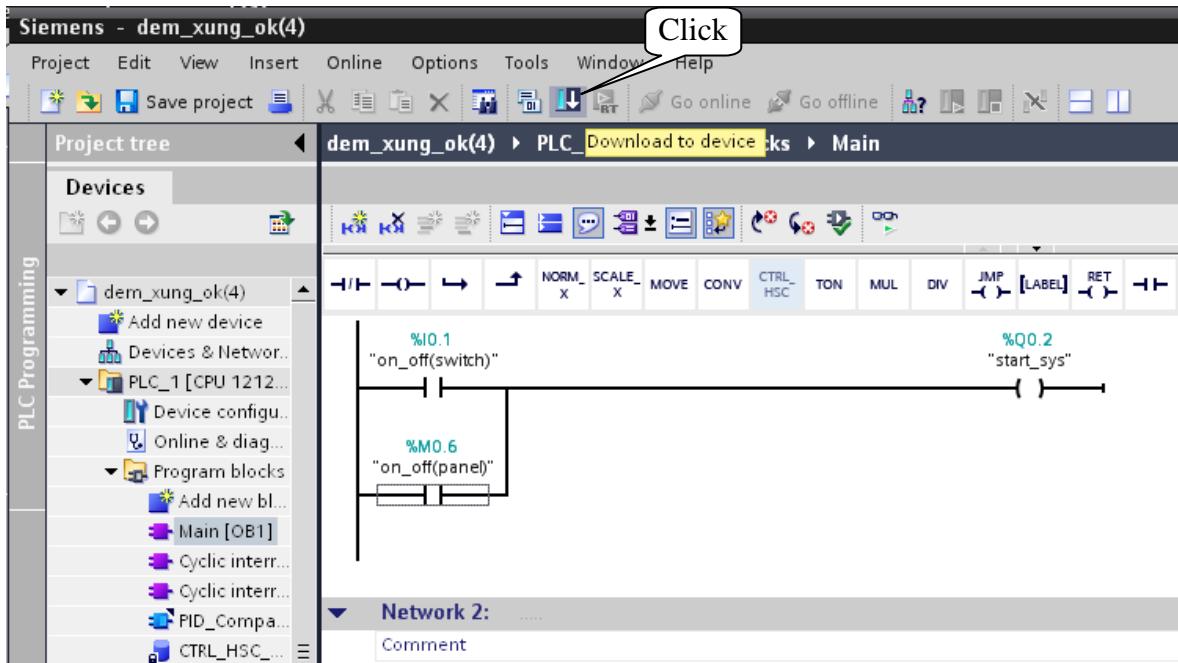
**Hình 1.13: Kết nối thành công**

Sau khi kết nối thành công, ta gán địa chỉ IP cho CPU:



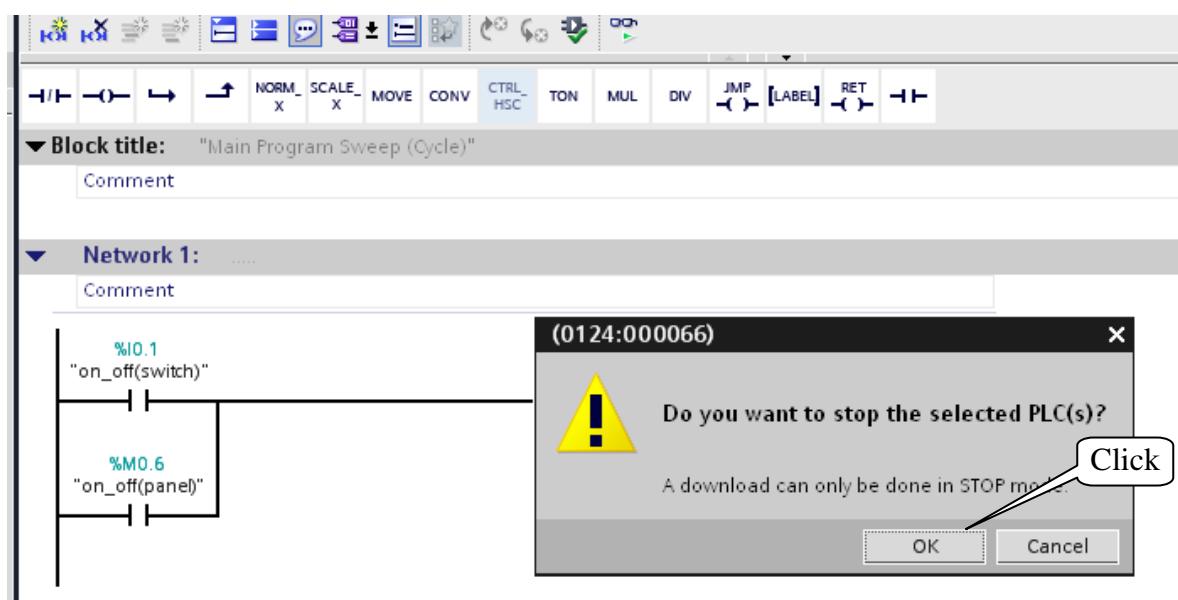
**Hình 1.14: Cho phép đặt địa chỉ IP**

Sau khi IP được gán, chương trình sẽ quay lại giao diện lập trình chính. Để download chương trình xuống CPU, ta click vào biểu tượng download hoặc vào Online/Download to device:



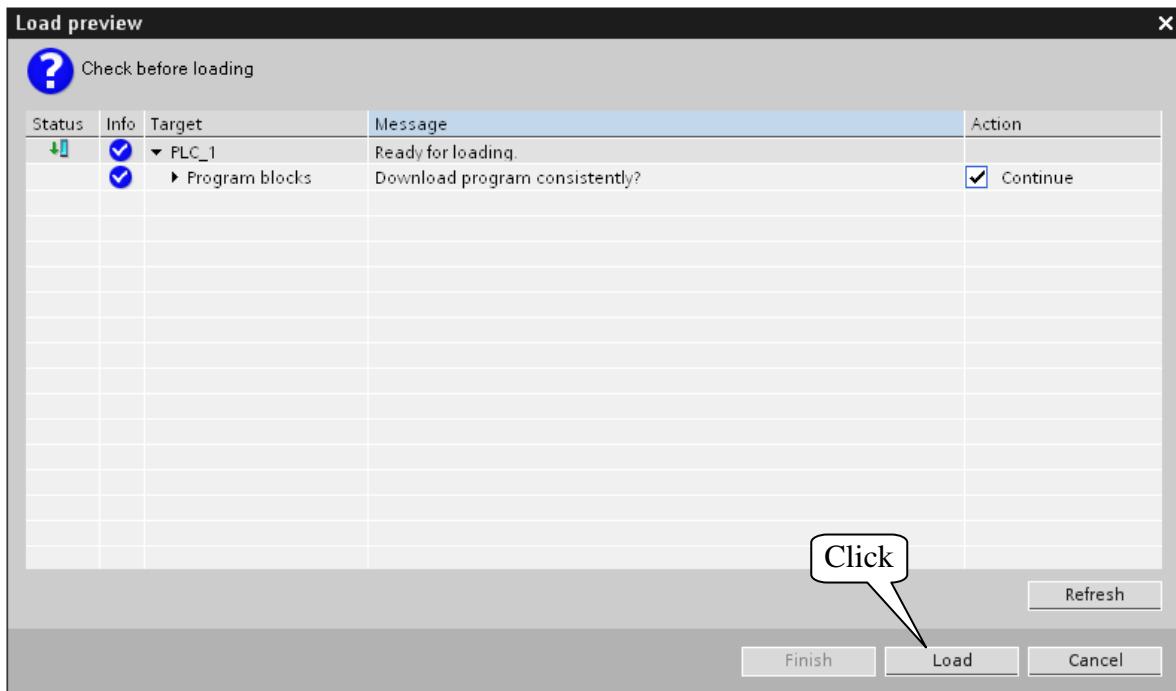
Hình 1.15: Chọn download chương trình xuống PLC

Nếu CPU đang ở chế độ RUN thì sẽ có thông báo yêu cầu stop CPU trước khi download:



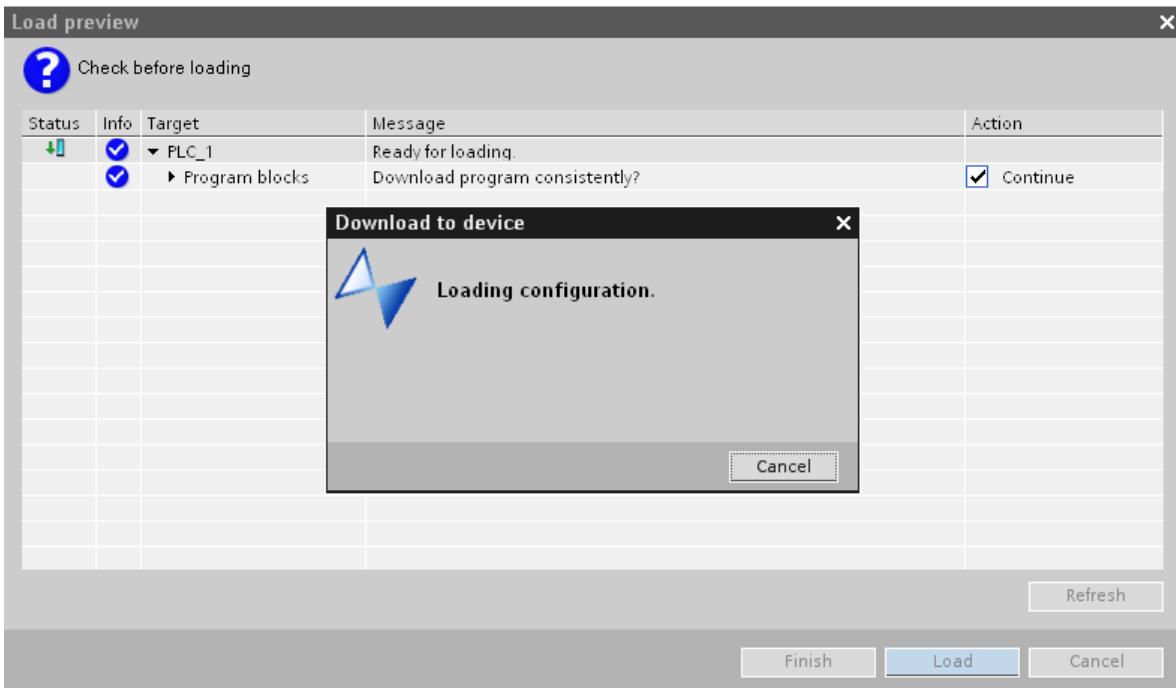
Hình 1.16: Chọn chế độ Stop để download chương trình xuống PLC

Chọn Load để download chương trình xuống PLC:



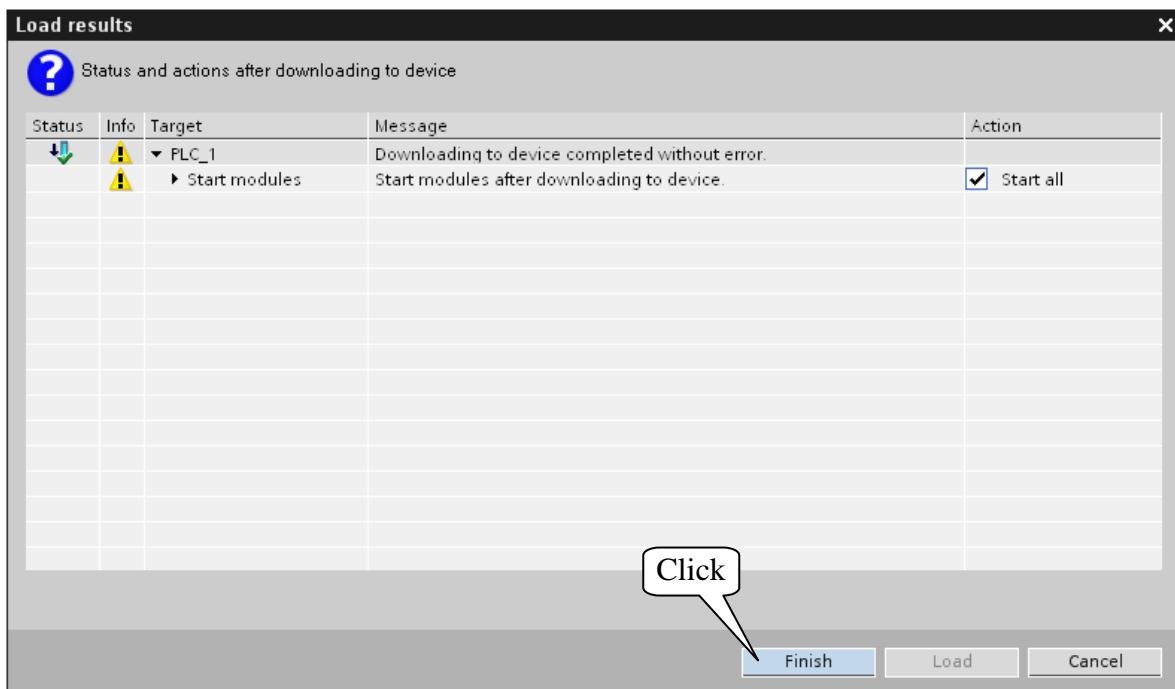
**Hình 1.17: Download chương trình xuống PLC**

Chương trình đang được download



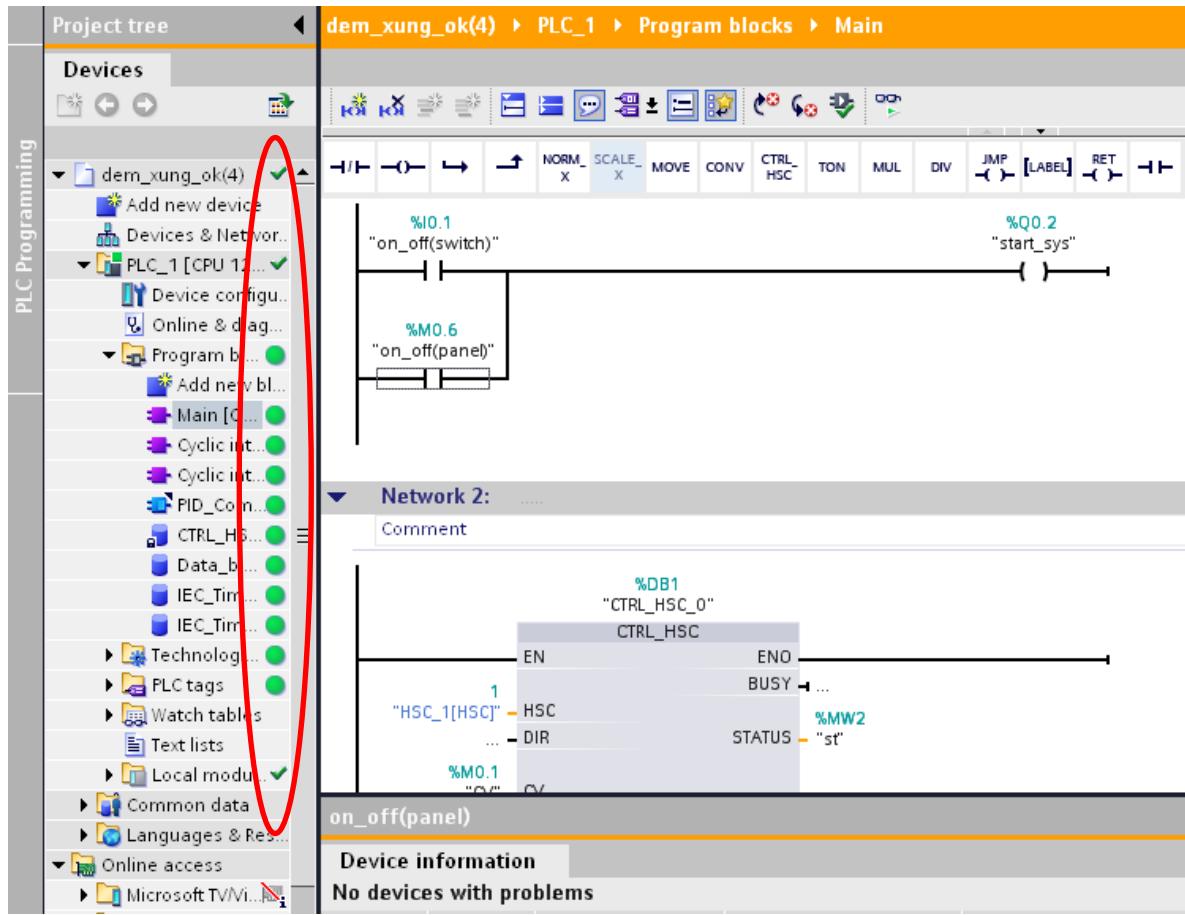
**Hình 1.18: Chương trình đang cấu hình PLC**

Quá trình download chương trình hoàn thành. Ta chọn finish.



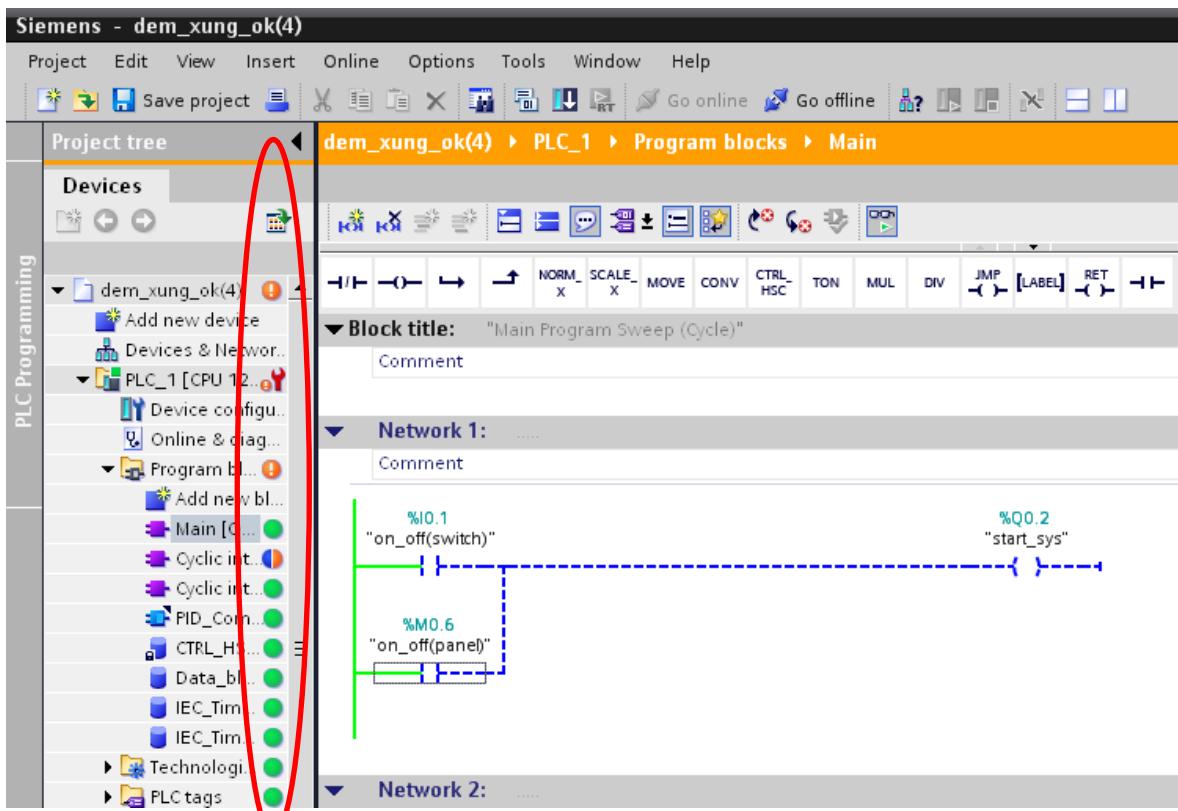
**Hình 1.19: Hoàn thành download chương trình**

Sau khi download chương trình thành công, ta kiểm tra lại cấu hình có lỗi hay không. Nếu không có lỗi, chương trình báo bằng các dấu check và dấu tròn màu xanh như hình 1.20.



**Hình 1.20: Nạp chương trình và cấu hình thiết bị thành công**

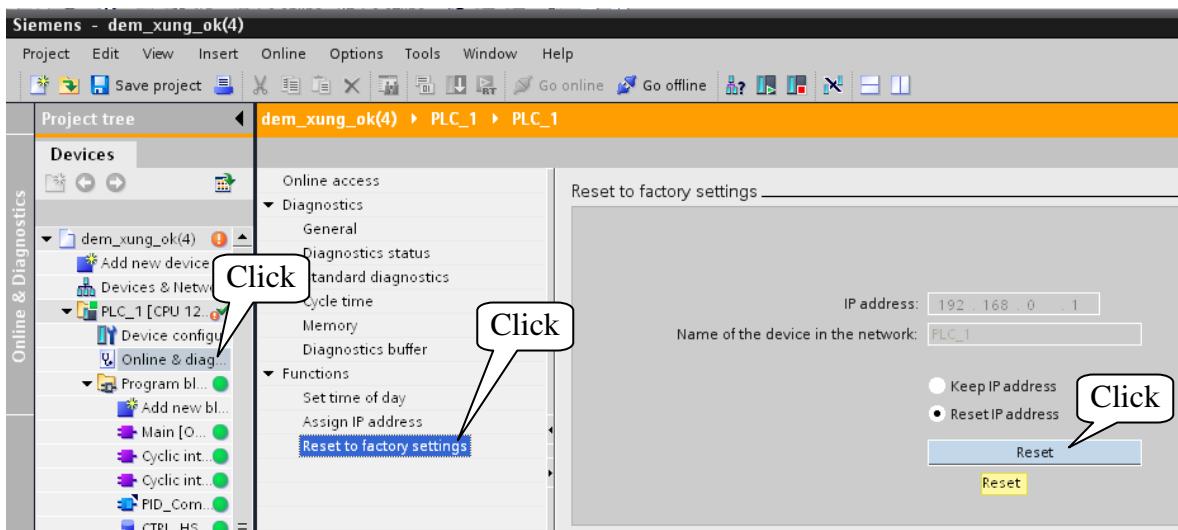
Nếu có lỗi xảy ra, chương trình sẽ báo lỗi bằng các dấu như hình dưới:



Hình 1.21: Giao diện hiển thị lỗi câu lệnh

Các lỗi này thường xảy ra khi ta chưa cấu hình hết các thiết bị được sử dụng hoặc cấu hình sai các thiết bị được lập trình. Hoặc có thể xảy ra trong khi mô phỏng. Để sửa các lỗi này, ta chỉ cần reset lại IP thì các thiết bị sẽ tự động được cấu hình.

Chọn Online & diagnostics/Reset to factory settings/Reset IP address/Reset.



Hình 1.22: Reset địa chỉ IP

Sau khi cài đặt và download chương trình thành công. Ta có thể mở phông chương trình.

## 1.3 SIMATIC WINCC BASIC V10.5

### 1.3.1 Giới thiệu

Ngoài SIMATIC STEP 7 Basic, trong Totally Integrated Automation Portal còn hỗ trợ thêm phần mềm SIMATIC WinCC Basic.

WinCC là một trong các chương trình ứng dụng Scada trong lĩnh vực dân dụng cũng như công nghiệp. WinCC được dùng để điều hành các màn hình hiển thị và hệ thống điều khiển trong tự động hóa sản xuất và quá trình.

WinCC viết tắt của Windows Control Center, là một phần mềm của hãng Siemens dùng để giám sát, điều khiển và thu thập dữ liệu trong quá trình sản xuất. Theo nghĩa hẹp, WinCC là chương trình hỗ trợ cho người lập trình thiết kế các giao diện Người và Máy (HMI) trong hệ thống SCADA (Supervisory Control And Data Acquisition), với chức năng chính là thu thập số liệu, giám sát và điều khiển quá trình sản xuất. Những thành phần có trong WinCC dễ sử dụng, giúp người dùng tích hợp những ứng dụng mới hoặc có sẵn mà không gặp bất kỳ trở ngại nào.

WinCC cung cấp nhiều module chức năng thường dùng trong công nghiệp. Giao diện điều khiển mạnh, việc truy cập ảnh nhanh chóng và chức năng lưu trữ an toàn (bảo mật) của nó đảm bảo tính hữu dụng cao.

### 1.3.2 Các thành phần soạn thảo

Trong giao diện soạn thảo có hỗ trợ các thẻ sau:

#### a. Tool (công cụ)

Thẻ “Tool” chứa các đối tượng trong các ô khác nhau.

- + Basic objects: Các đối tượng cơ bản.
- + Elements: Các yếu tố trong lập trình giao diện (button, I / O field, bar...).
- + Controls: Bao gồm Alarm view, Trend view, Recipe view, User view.

- + User control (tùy chọn).
- + Graphics: thư viện đồ họa.
- + Đưa đối tượng từ các ô vào màn hình soạn thảo bằng cách kéo và thả chúng. Các đối tượng có sẵn để lựa chọn được xác định bởi các tính năng của thiết bị HMI bạn đang cấu hình.

#### **b. Animations (hiệu ứng động)**

Thẻ “Animations” bao gồm các thẻ chứa các mẫu cho hiệu ứng chuyên động.

- + Movements (các hiệu ứng di chuyển): Diagonal (di chuyển chéo), direct (di chuyển theo các hướng được cài đặt), horizontal (di chuyển ngang) , vertical (di chuyển thẳng đứng).

- + Display (hiển thị): Appearance, visibility.
- + Miscellaneous (thành phần khác): Operability (khả năng hoạt động).

Sử dụng phương pháp kéo và thả để thêm các hình ảnh động cho một đối tượng màn hình từ các ô “Movements”, “Display” và “Miscellaneous”.

#### **c. Layout (giao diện)**

Thẻ “Layout” chứa các ô để hiển thị các đối tượng và các yếu tố:

- + Zoom: Để chọn xem chi tiết.
- + Layers: Quản lý các đối tượng của màn hình. Chúng chứa đựng những thông tin về khả năng hiển thị và lớp hoạt động.
- + Grid: Xác định rõ đối tượng được liên kết với lưới và thiết lập kích thước cho một lưới .
- + Objects out of range: Các đối tượng nằm ngoài khu vực có thể nhìn thấy được hiển thị với vị trí, tên và loại.

#### **d. Libraries (thư viện)**

Thẻ “Libraries” hiển thị các thư viện sau trong các khung khác nhau:

- + Project libraries (thư viện dự án): Thư viện dự án được lưu trữ cùng với dự án.

- + Global library (tổng thư viện): Thư viện được lưu trữ trong một tập tin riêng biệt trong đường dẫn chỉ định trên cấu hình máy tính của bạn.

### 1.3.3 Các chức năng cơ bản của WinCC v10.5

#### a. Alarms

Alarm là chức năng quan trọng trong quá trình sản xuất. Nó cho phép thu thập, lưu trữ, hiển thị thông báo trong hiện tại lẫn quá khứ. Chức năng này giúp người vận hành có thể tìm ra lỗi để khắc phục một cách nhanh chóng. Ta có thể cài đặt các thuộc tính cảnh báo phụ thuộc vào các biến trạng thái.

Các kiểu báo động trong WinCC

- + Analog alarms: Cảnh báo tương tự được dùng để giám sát việc vượt quá giới hạn cho phép trong hệ thống.
- + Discrete alarms: Dùng để biểu thị trạng thái của hệ thống trong quá trình vận hành.
- + System alarms: Cảnh báo cho thấy tình trạng của hệ thống, cộng với các lỗi giao tiếp giữa các thiết bị và hệ thống HMI...

#### b. Trend view

Công cụ này cho phép ta liên kết với các biến trạng thái của PLC để hiển thị các giá trị của biến trạng thái đó trên đồ thị. Giao diện của đồ thị sẽ phụ thuộc vào cấu hình của bạn cho loại đồ thị mà bạn lựa chọn.

#### c. User administrations

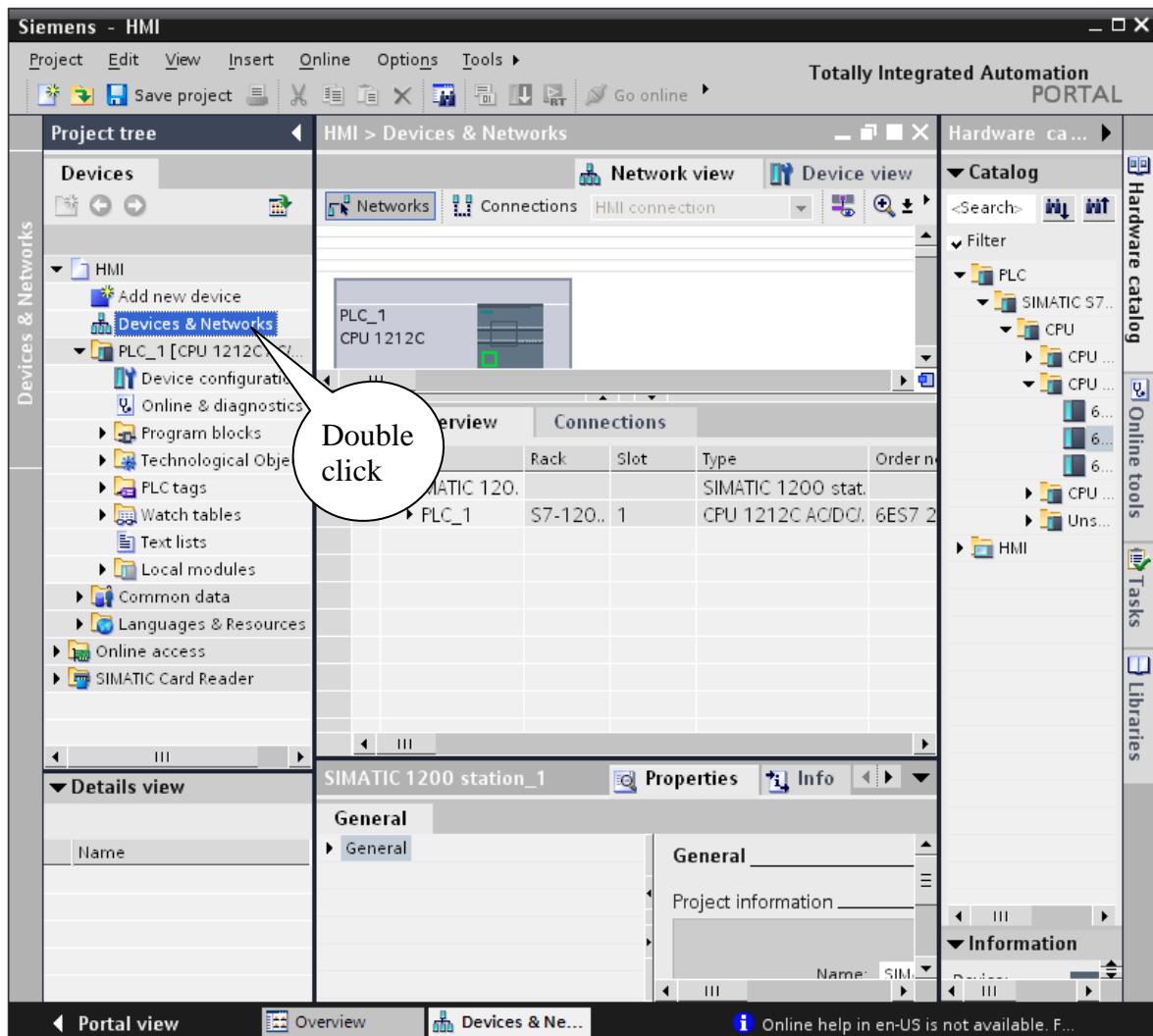
Đây là tính năng quản trị người dùng, một trong những tính năng quan trọng của WinCC. WinCC cung cấp chức năng này để giới hạn truy cập dữ liệu, điều khiển và các chức năng khác trong Runtime. Các tính năng này bảo vệ các ứng dụng của bạn chống lại hoạt động trái phép, hoạt động liên quan đến an toàn đã được giới hạn cho các nhóm người dùng cụ thể khi dự án được tạo ra. Để thực hiện điều này ta thiết lập người dùng và nhóm người dùng được thiết lập các quyền truy cập đặc trưng, được gọi là phân quyền. Sau đó bạn cấu hình các quyền cần thiết cho các đối tượng liên quan. Người truy cập chỉ được thực hiện trong các quyền giới hạn.

### 1.3.4 Thiết kế giao diện cho màn hình HMI

Để mở một cửa sổ thiết kế giao diện cho màn hình HMI, ta thực hiện các bước sau:

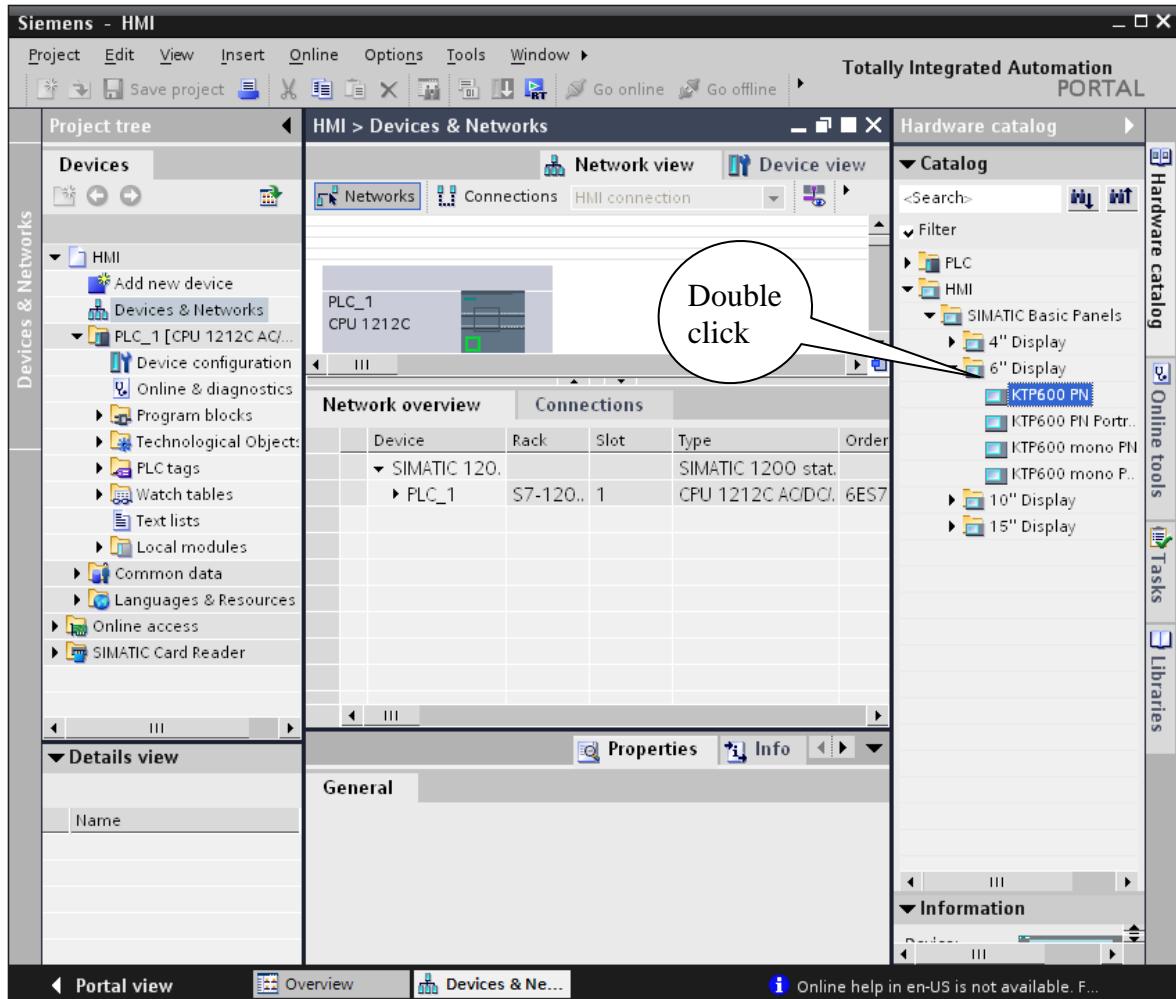
#### a. Thêm thiết bị HMI

Trong cửa sổ Project tree nhấn đúp chuột vào Add new device:



Hình 1.23: Add thiết bị màn hình HMI

Ta tiến hành đặt tên và chọn thiết bị HMI:

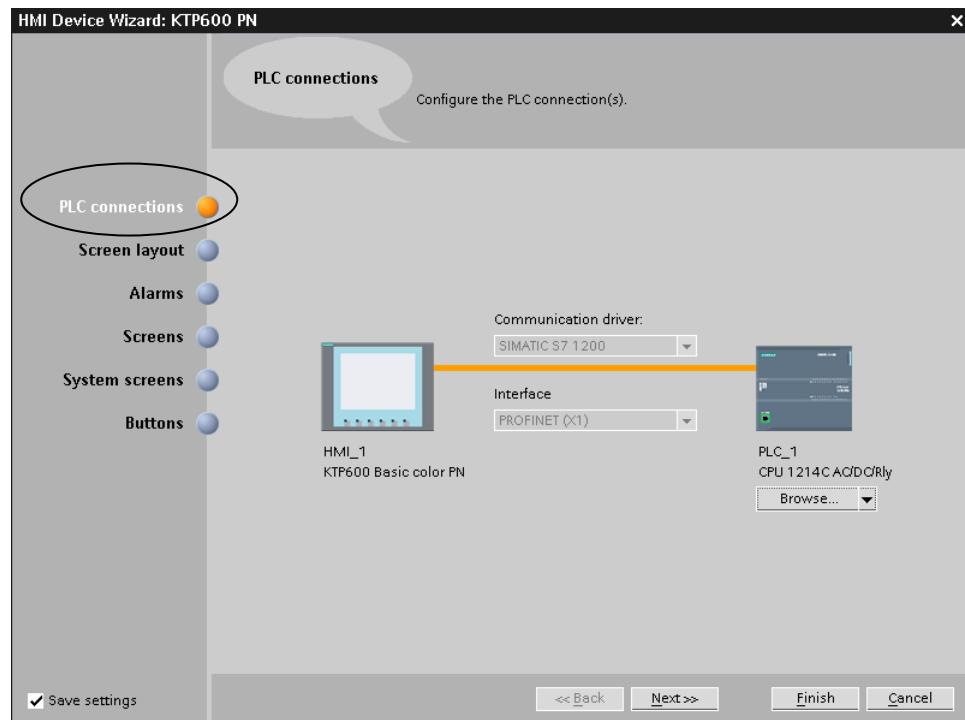


Hình 1.24: Đặt tên và chọn thiết bị HMI

### b. Tạo một mẫu cho thiết bị HMI

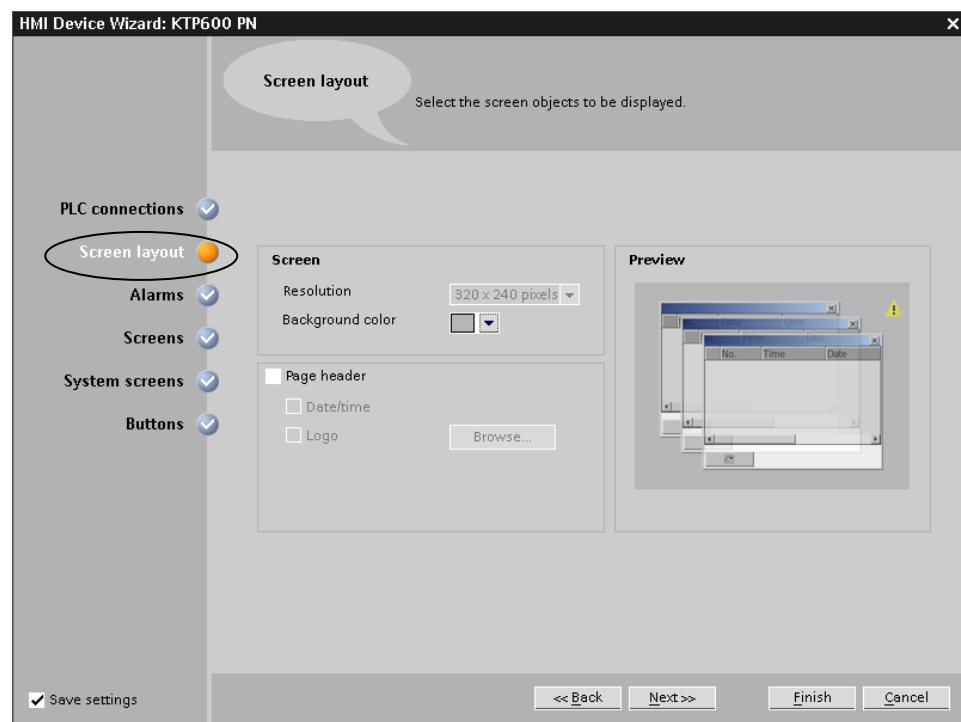
Để tạo một mẫu cho thiết bị HMI ta thực hiện các bước:

- + Cấu hình việc kết nối đến PLC



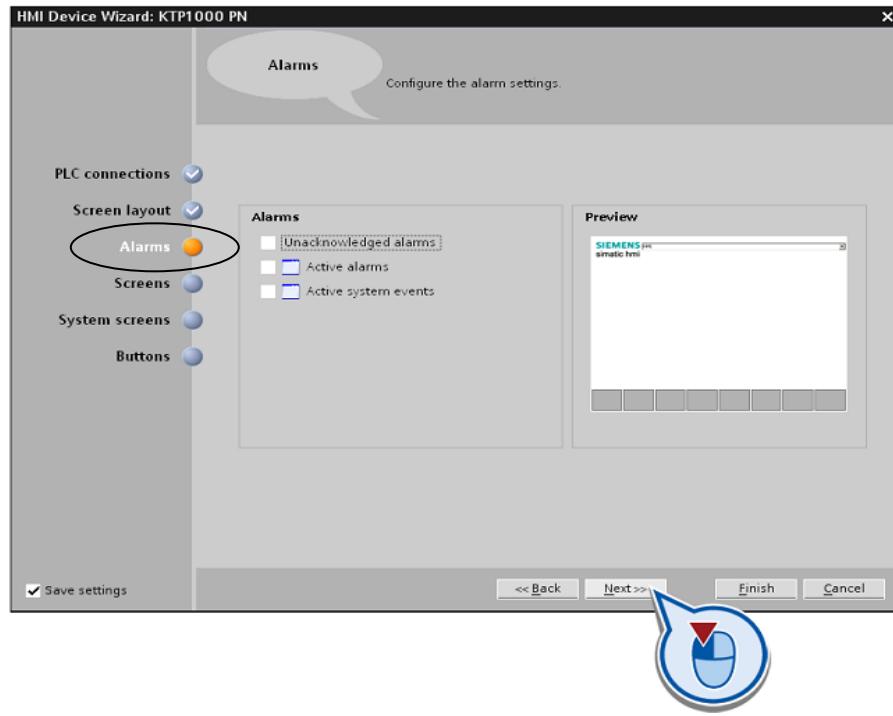
**Hình 1.25: Cấu hình kết nối HMI và PLC**

- + Chọn màu nền và các yếu tố cho tiêu đề



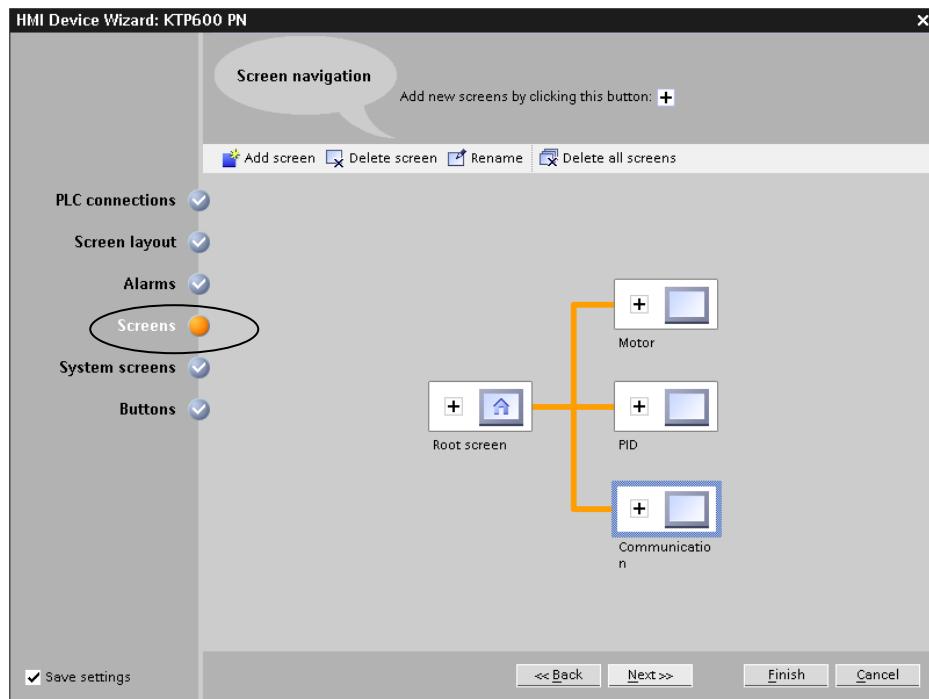
**Hình 1.26: Chọn đặc tính cho màn hình HMI.**

- + Kích hoạt Alarms hoặc bỏ chọn Alarms (ta cũng có thể thiết kế một Alarms trong khi soạn thảo giao diện).



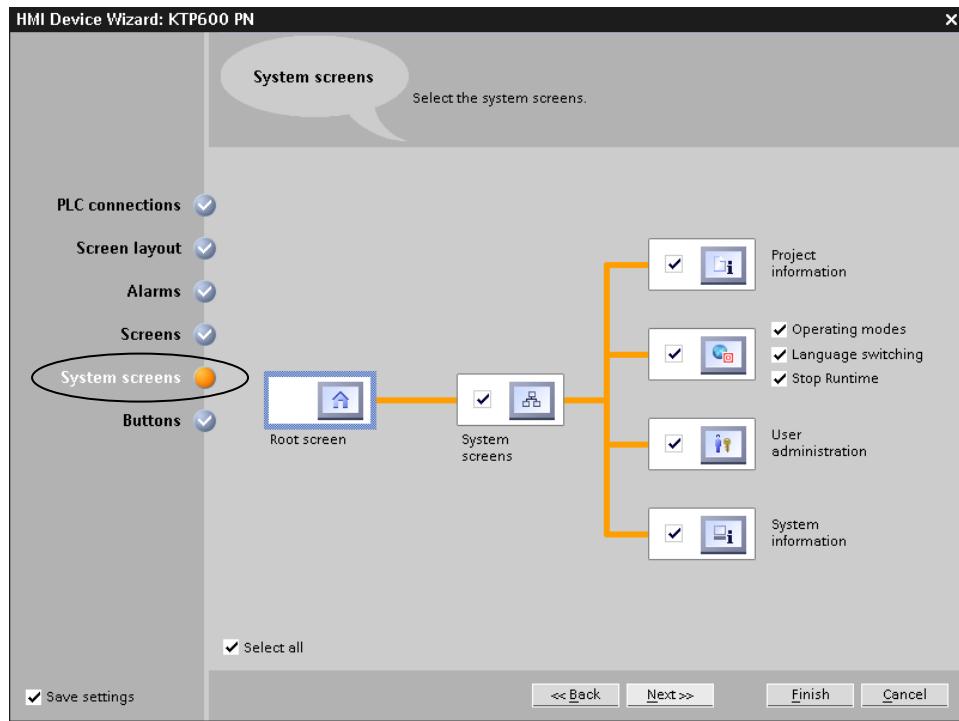
**Hình 1.27: Tạo một đồng hồ trên màn hình HMI**

- + Đổi tên màn hình HMI nếu cần thiết



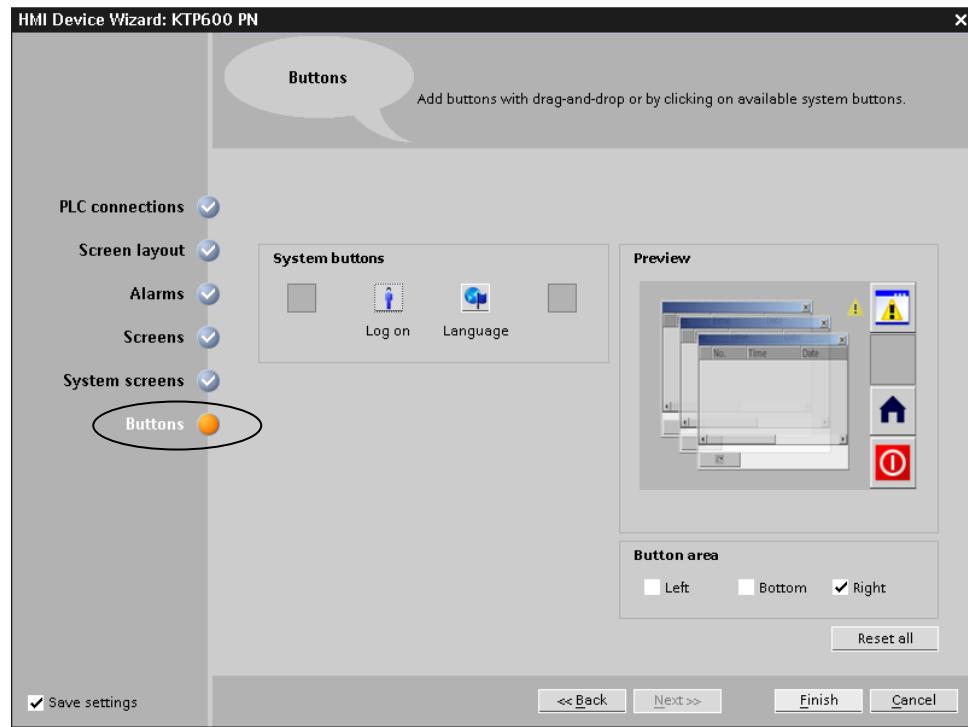
**Hình 1.28: Đặt các đặc tính cho màn hình hiển thị.**

- + Cài đặt hệ thống màn hình (nếu cần):



**Hình 1.29: Cài đặt một hệ thống màn hình HMI.**

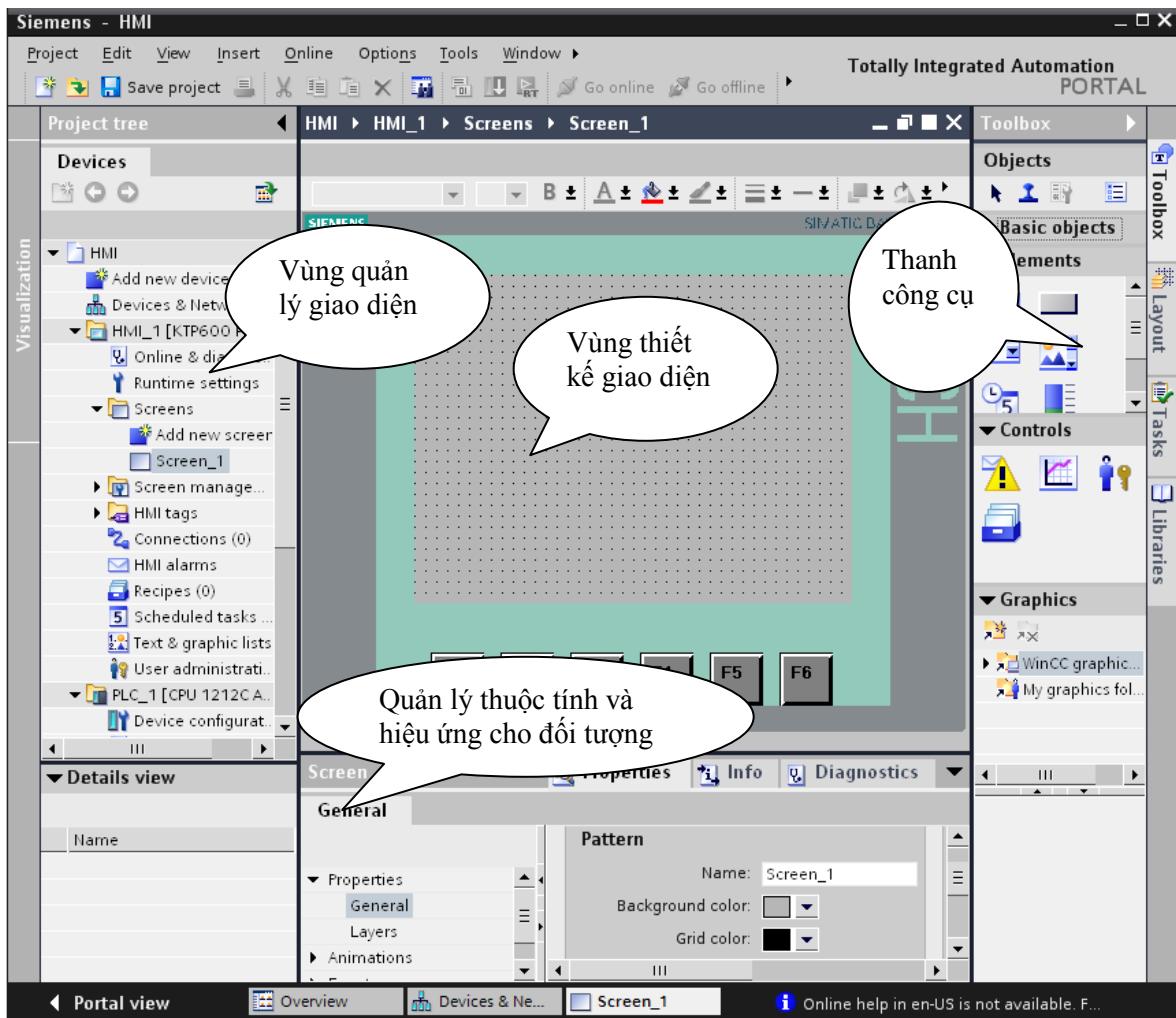
- + Chọn các nút nhấn cần thiết cho màn hình HMI, có thể chọn nút “EXIT”. Chế độ Runtime sẽ được chấm dứt nếu ta ấn nút “EXIT”.



**Hình 1.30: Chọn nút nhấn.**

Sau khi hoàn thành các thao tác trên ta ấn “Finish” để hoàn tất. Giao diện lập trình sẽ hiện ra (hình 1.31), sau đó ta sẽ tiến hành lập trình giao diện cho hệ thống.

### c. Lập trình giao diện cho HMI



Hình 1.31: Giao diện lập trình cho thiết bị HMI

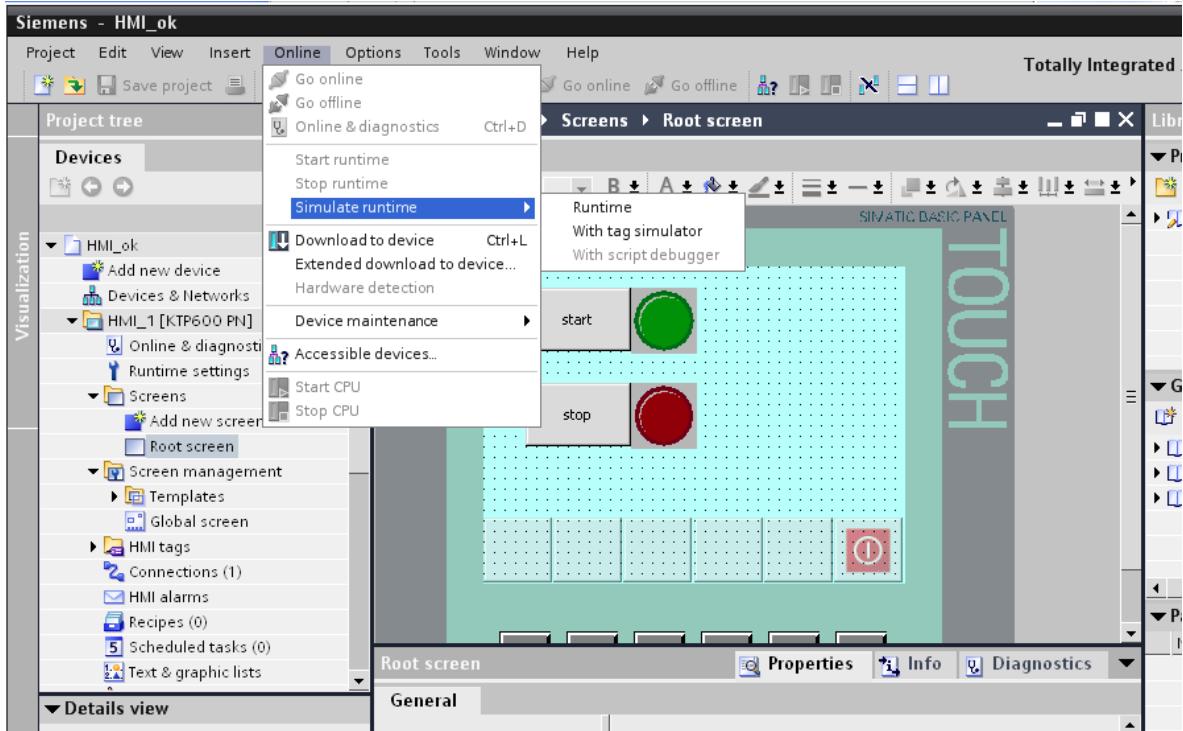
Tùy theo cấu trúc mỗi hệ thống người lập trình sẽ sử dụng các công cụ và thư viện để thiết kế giao diện HMI khác nhau.

### d. Mô phỏng thời gian chạy cho HMI

Một project sau khi được soạn thảo có thể mô phỏng thời gian chạy một cách dễ dàng bằng một trong hai chế độ sau: With tag simulator và Runtime.

Để bắt đầu mô phỏng thời gian chạy thông qua thanh menu. Các cửa sổ thiết bị HMI phải được hoạt động. Nếu menu không hoạt động, click chuột vào một vùng bất kỳ trong phạm vi màn hình HMI.

Vào menu Online/ Simulate Runtime. Sau đó chọn chế độ Runtime hoặc With tag simulator.



**Hình 1.32 Mô phỏng thời gian chạy**

With tag simulator: đây là chế độ mô phỏng giao diện với các thẻ độc lập của chương trình PLC, sử dụng khi không kết nối với PLC thực. Ta sử dụng bảng của Simulator Runtime để chọn các thẻ PLC phù hợp và sửa đổi giá trị của chúng. Các đối tượng trong màn hình HMI sẽ hoạt động như thẻ các thẻ đã được thiết lập bởi các chương trình của PLC trong thời gian chạy.

Trong chế độ này các nút nhấn cũng như các thành phần điều khiển khác sẽ không được kích hoạt.

Runtime: có thẻ chạy chế độ này khi giao diện HMI của bạn đã kết nối với PLC thực. Lúc này các đối tượng trên giao diện HMI sẽ hoạt động với các thẻ đã được thiết lập. Có thẻ điều khiển các đối tượng trên giao diện một cách dễ dàng.

Ngoài ra, bạn có thể chạy chế độ Runtime bằng cách click vào biểu tượng



trên thanh công cụ.

**Chú ý:** [3] (trang 62)

*Trong chế độ Runime, nếu giao diện HMI không thể kết nối được với PLC thì ta cần kiểm tra việc kết nối thông qua “Setting PG/PC Interface”:*

- + Nhấn đúp chuột vào “Setting PG/PC Interface” trong Control Panel. Một hộp thoại mở ra.
- + Chọn “S7ONLINE” trong “Access point of application”.
- + Chọn giao diện trong khu vực “Interface Parameter Assignment Used”.
- + Chọn “OK” để hoàn tất.

## 1.4 PC ACCESS

### 1.4.1 Giới Thiệu

PC Access là chương trình làm nhiệm vụ trung gian giữa PLC và WinCC. Trong đó PC Access được xem như là một OPC Client. PC Access liên kết trực tiếp với PLC thông qua các tag để cập nhật giá trị của các biến trạng thái.

Từ các giá trị các biến trạng thái trong PC Access ta có thể truy xuất dữ liệu của WinCC sang một file Excel một cách dễ dàng bằng cách sử dụng chức năng “Add Ins” trên Excel.

### 1.4.2 Kết Nối Giữa PC Access Và PLC S7-1200

#### a. Khởi động PC Access:

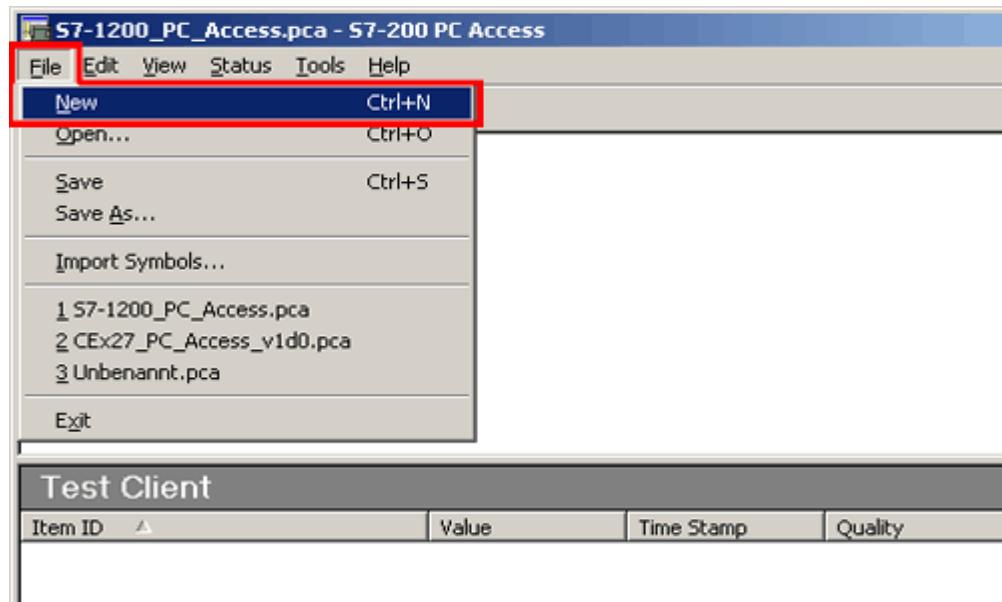
Start-> programs->SIMATIC->S7-200 PC Access PC Access V1.0 SP4.



Click vào biểu tượng **V1.0 PC Access SP4.lnk** trên desktop.

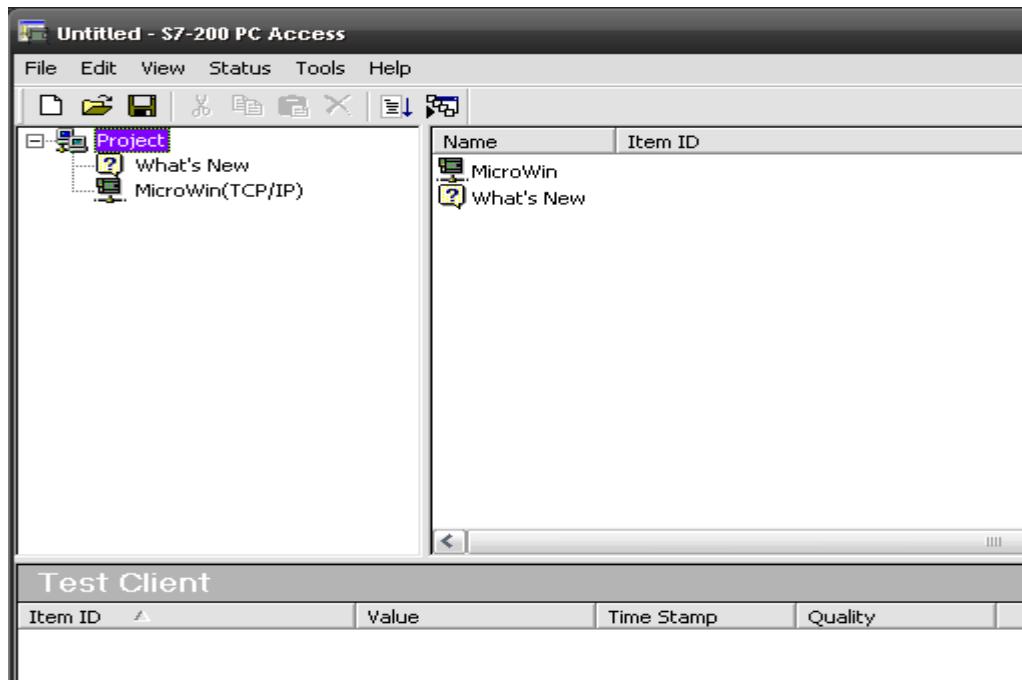
#### b. Tạo một Project mới

Từ thanh Menu chọn “File / New”. Một Project mới sẽ được tạo



**Hình 1.33: Tạo một project mới cho PC Access**

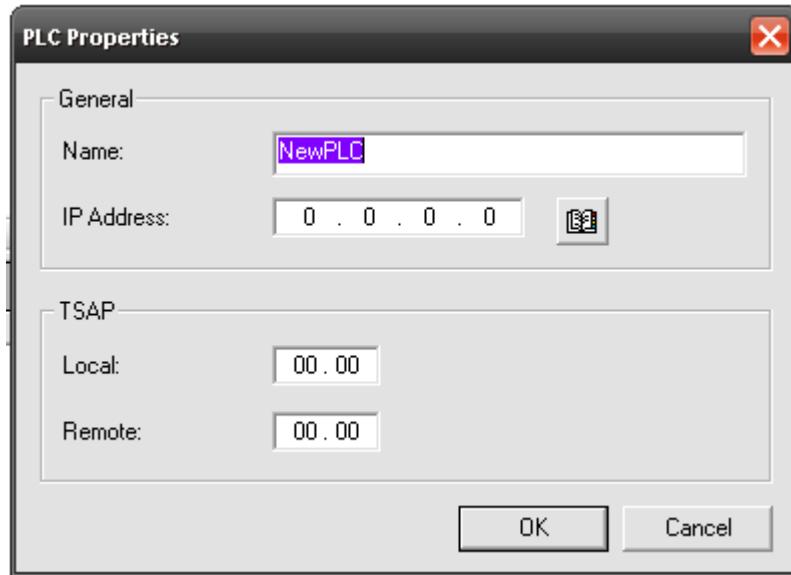
Lúc này màn hình xuất hiện giao diện của PC Access.



**Hình 1.34: Giao diện PC Access**

### c. Thêm một PLC mới

Từ thư mục “MicroWin(TCP/IP)”, ta click phải chuột chọn “New PLC” để tạo liên kết giữa PLC với PC Access. Hộp thoại "PLC properties" hiện ra.

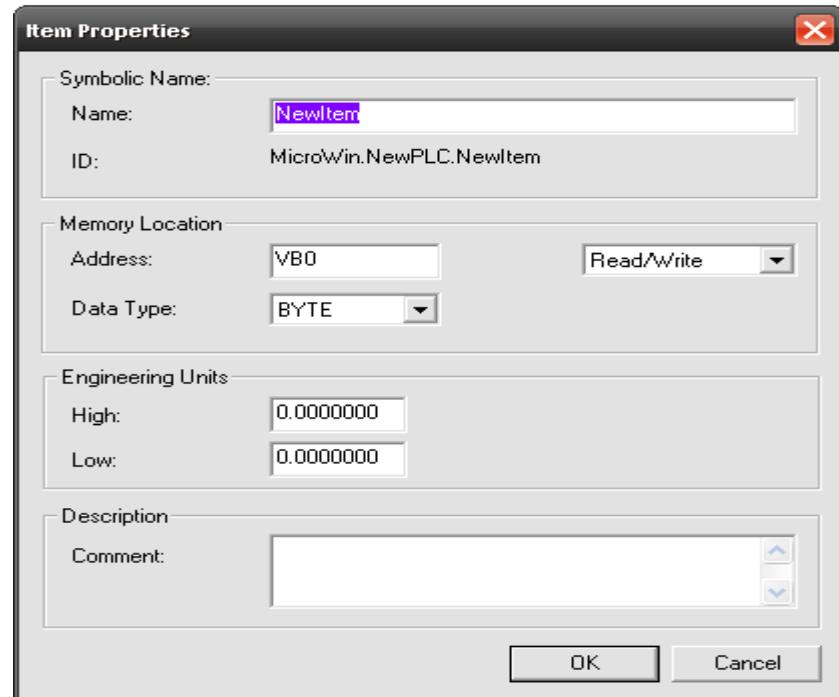


Hình 1.35: Tạo liên kết giữa PLC và PC Access

Trong đó:

- + Name: Tên đối với liên kết giữa PLC và PC Access.
- + IP Address: Địa chỉ IP của PLC S7-1200.
- + TSAP Local: "10.00" (TSAP của PLC S7-1200) .
- + TSAP Remote: "03.01" (TSAP của PC Access).

Sau khi đặt tên cho liên kết, tên này sẽ là thư mục con của “MicroWin(TCP/IP)”. Từ thư mục con này, ta click phải chuột chọn “New/Item...”. Sau khi chọn xong cửa sổ thuộc tính Item sẽ hiện ra như hình.

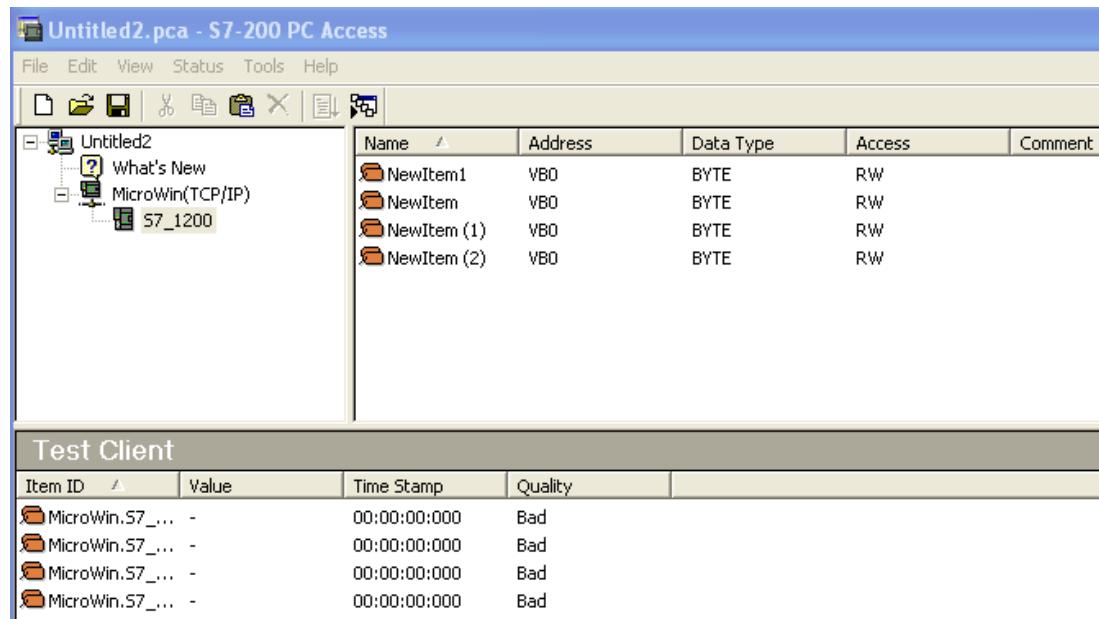


**Hình 1.36: Thiết lập các thuộc tính cho PC Access**

- + Name: Tên tag trên PLC Access liên kết với biến trạng thái của PLC.
- + Address: Địa chỉ hay biến trạng thái từ PLC.
- + Data Type: Kiểu dữ liệu của address.
- + Read/Write: Là kiểu truy xuất dữ liệu từ PC Access đối với address.
- + High: Giá trị cao nhất của address.
- + Low: Giá trị thấp nhất của address.
- + Comment: Ghi chú thông tin từ biến trạng thái này.

#### d. Kiểm tra Item thông qua Test Client.

Để kiểm tra Item thông qua Test Client, trước tiên ta add các Item vào Test Client.



**Hình 1.37 Kiểm tra Item thông qua Test Client**

Sau đó, vào “Status / Start Test Client” để chạy kiểm tra.

Nếu PC Access đã kết nối được với PLC S7-1200, trong cột “Value” ta sẽ thấy giá trị của các Item và trong cột “Quality” sẽ hiển thị là “Good”.

**Chú ý:**

*Để có thể liên kết giữa PLC S7-1200 và PC Access thì đối với PC Access phải sử dụng phiên bản PC Access V1.0.4.10 (SP4) hoặc cao hơn.*

## **CHƯƠNG 2 CÁC PHƯƠNG PHÁP LẬP TRÌNH PLC**

### **2.1 LẬP TRÌNH**

Lập trình dùng để chỉ thao tác của con người nhằm kiến tạo nên các chương trình máy tính thông qua các ngôn ngữ lập trình. Người ta còn gọi quá trình lập trình đó là quá trình mã hoá thông tin tự nhiên thành ngôn ngữ máy. Trong các trường hợp xác định thì chữ lập trình còn được viết là "viết mã" (cho chương trình máy tính).

### **2.2 CÁC PHƯƠNG PHÁP LẬP TRÌNH PLC**

Phương pháp lập trình là cách thức để người lập trình viết được chương trình từ một yêu cầu dựa trên ngôn ngữ lập trình.

Có nhiều phương pháp lập trình khác nhau. Mỗi phương pháp có một ưu điểm và khuyết điểm riêng.

Đối với lập trình PLC thì tùy từng bài tập cụ thể mà ta có những phương pháp cụ thể. Bên cạnh đó người lập trình cũng có thể kết hợp một vài phương pháp trong một chương trình. Sau đây là một số phương pháp lập trình PLC thông dụng:

#### **2.2.1 Lập trình tuyến tính**

Trong thực tế để giải quyết các bài toán đơn giản người ta thường dùng phương pháp lập trình tuyến tính.

Đặc trưng cơ bản của lập trình tuyến tính là tư duy theo lối tuần tự. Chương trình sẽ được thực hiện theo thứ tự từ đầu đến cuối, lệnh này kế tiếp lệnh kia cho đến khi kết thúc chương trình.

Lập trình tuyến tính có hai đặc trưng:

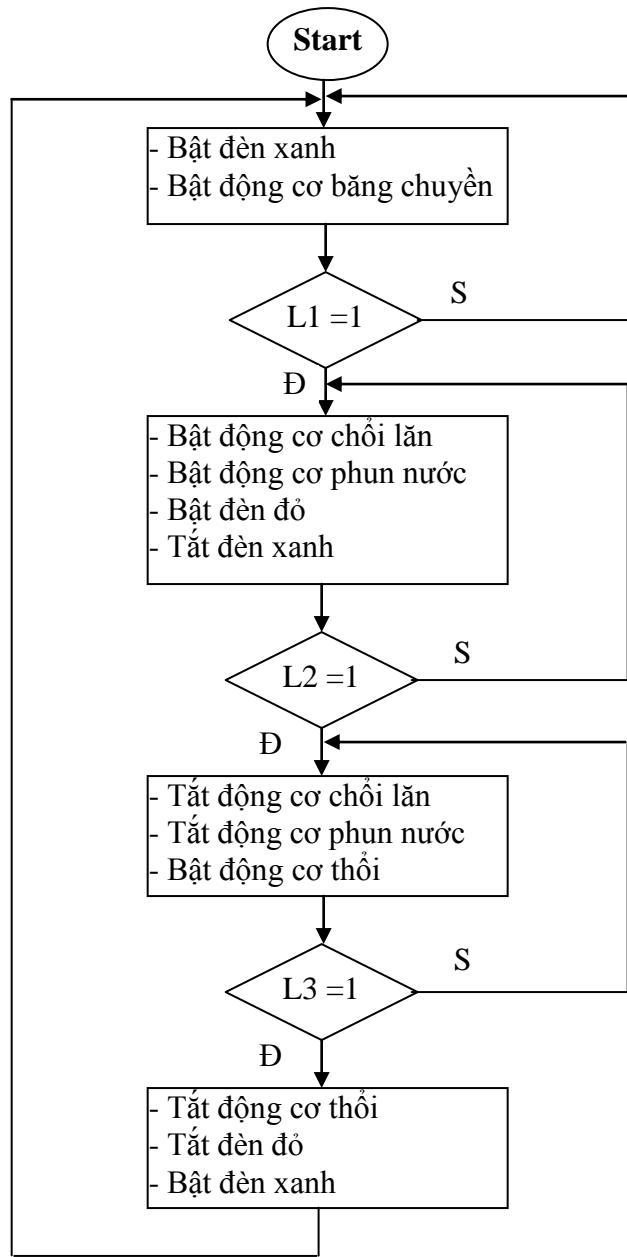
- Đơn giản: chương trình được tiến hành đơn giản theo lối tuần tự, không phức tạp.
- Đơn luồng: chỉ có một luồng công việc duy nhất, và các công việc được thực hiện tuần tự trong luồng đó.

Tính chất:

- **Ưu điểm:** Do tính đơn giản, lập trình tuyến tính được ứng dụng cho các chương trình đơn giản và có ưu điểm dễ hiểu.
- **Nhược điểm:** Với các ứng dụng phức tạp, người ta không thể dùng lập trình tuyến tính để giải quyết.

#### **Ví dụ: Hệ thống rửa xe ô tô tự động.**

- Nguyên lý hoạt động: Khởi động hệ thống bằng công tắc Sw (on/off). Khi hệ thống được khởi động thì động cơ băng chuyên chạy và bật đèn xanh. Nhận biết có xe vào hay không băng cảm biến L1. Khi có xe vào thì đèn xanh tắt, bật đèn đỏ, bật động cơ chổi lăn và bật động cơ phun nước. Khi xe đến vị trí cảm biến L2 thì hết việc rửa. Bắt đầu việc làm khô bằng cách tắt động cơ chổi lăn, tắt động cơ phun nước và bật động cơ thổi. Khi xe đến vị trí cảm biến L3 thì xong quá trình rửa xe. Lúc này tắt động cơ thổi hơi, tắt đèn đỏ và bật đèn xanh.
- Lưu đồ giải thuật:



**Hình 2.1 Lưu đồ giải thuật chương trình rửa xe tự động**

## 2.2.2 Lập trình cấu trúc

Trong một chương trình máy tính, các khái niệm năng có thể được thực hiện không chỉ theo trình tự mà còn có thể theo các tình huống và lặp lại nhiều lần. Một chương trình máy tính có thể được viết dựa trên ba cấu trúc: trình tự, quyết định và vòng lặp.

- Trình tự nghĩa là các câu lệnh được thực hiện theo trình tự nhất định: trên xuống.

– Quyết định là sự quy định sẽ thực hiện chương trình như thế nào phụ thuộc vào sự thỏa mãn các điều kiện nhất định.

– Vòng lặp thể hiện sự thực hiện có tính lặp một số đoạn lệnh của chương trình khi các điều kiện nào đó vẫn được thỏa mãn.

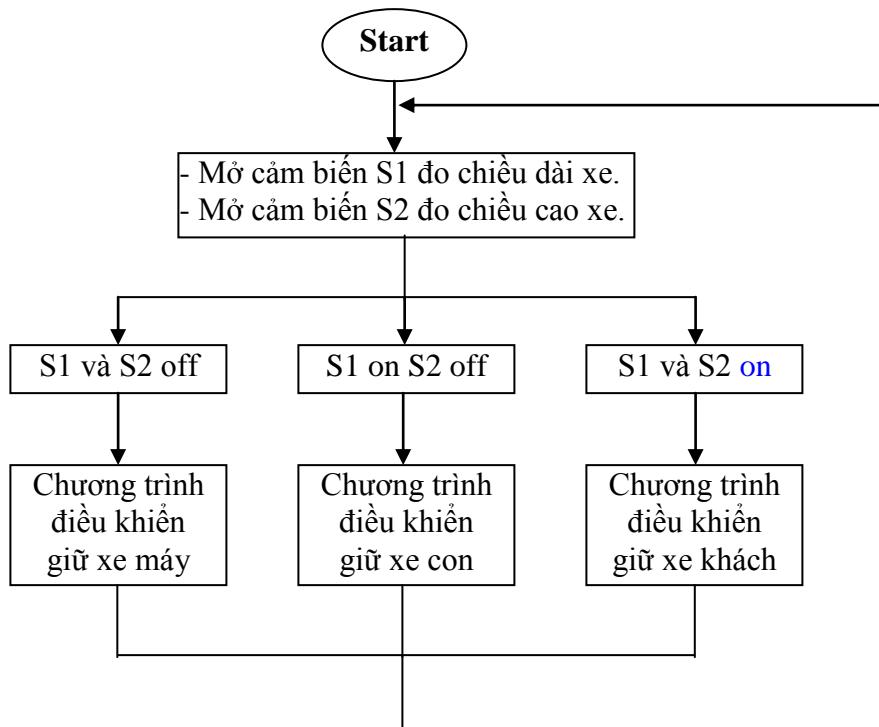
### Ưu điểm

- Chương trình rõ ràng, dễ hiểu, dễ theo dõi.
- Tự duy giải thuật rõ ràng.

Vấn đề cơ bản của lập trình cấu trúc là bằng cách nào để phân chia chương trình chính thành các chương trình con cho phù hợp với yêu cầu, chức năng và mục đích của mỗi bài toán. Thông thường, để phân rã bài toán trong lập trình cấu trúc, người ta sử dụng phương pháp thiết kế trên xuống (top-down). Theo phương pháp này, người thiết kế hệ thống chia các chức năng (hàm) chính của hệ thống thành các chức năng nhỏ hơn, đến lượt mình, các chức năng nhỏ này lại được chia tiếp thành các chức năng nhỏ hơn nữa cho đến khi được các khối (hàm) chương trình đủ nhỏ. Việc phân tích này được thể hiện trực quan theo sơ đồ khối.

### Ví dụ: Hệ thống giữ xe tự động.

– Nguyên lý: Hệ thống giữ xe với ba loại xe (theo kích thước): xe máy, xe con, xe khách. Khi khởi động hệ thống giữ xe thì bằng chuyền hoạt động và đèn xanh bật. Khi có xe vào thì nhận dạng loại xe (theo kích thước) bằng cảm biến. Nếu là xe máy thì đưa vào chương trình con giữ xe máy, nếu là xe con thì đưa vào chương trình con giữ xe con và xe khách thì đưa vào chương trình giữ xe khách.



Hình 2.2 Lưu đồ giải thuật chương trình giữ xe tự động

### 2.2.3 Lập trình ngẫu nhiên

Phương pháp lập trình tuyến tính và phương pháp lập trình cấu trúc thích hợp với các bài toán có ngõ vào ít trạng thái và các trạng thái là đơn giản. Tuy nhiên, đối với các bài toán với ngõ vào gồm nhiều biến trạng thái thì việc lập trình rất khó khăn.

Để giải quyết vấn đề này người ta đưa ra phương pháp lập trình thích nghi (ngẫu nhiên). Lập trình ngẫu nhiên là các trạng thái xảy ra một cách ngẫu nhiên không theo một trình tự nào hết. Chương trình phải đưa ra tất cả các trường hợp có thể xảy ra. Và xử lý đúng chức năng đối với mỗi trường hợp này

Để hiểu hơn vấn đề này, xét một ví dụ đơn giản bên dưới. Các trường hợp phức tạp với ngõ vào nhiều trạng thái các bước thực hiện tương tự.

#### Ví dụ: Mạch điều khiển tắt mở đèn bằng nút nhấn.

Nguyên lý: Khi nhấn nút thì đèn sáng, đèn vẫn sáng khi nút nhả. Khi nhấn lần kế tiếp thì đèn tắt, khi nhả nút thì đèn vẫn tắt.

Các bước thực hiện:

– Đặt giá  $x$  là biến trạng thái biểu diễn cho công tắc và  $y$  là biến trạng thái biểu diễn cho đèn. Khi công tắc được nhấn tương ứng  $x = 1$  khi không được nhấn  $x = 0$ . Khi đèn sáng tương ứng  $y = 1$  và đèn tắt tương ứng  $y = 0$ .

– Lập sơ đồ các trường hợp có thể xảy ra khi nút được nhấn.

Ta có: Các trạng thái lần lượt như sau:

+ (1) Khi nút không được nhấn và đèn tắt ta có:  $x = 0, y = 0$ .

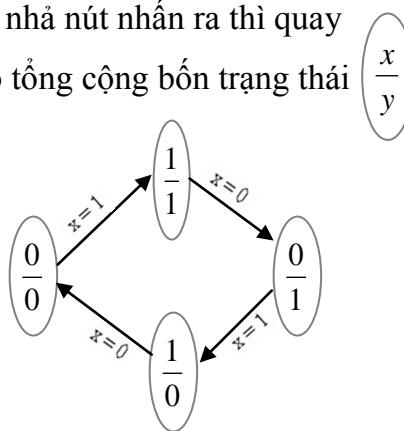
+ (2) Nhấn nút thì đèn sáng ( $x = 1$  và  $y = 1$ ).

+ (3) Khi nhả nút ra thì đèn vẫn sáng ( $x = 0$  và  $y = 1$ ).

+ (4) Khi nhấn nút lần nữa thì đèn tắt ( $x = 1$  và  $y = 0$ ).

+ Sau đó nhả nút nhấn ra thì quay  $\left(\frac{x}{y}\right)$  về trạng thái ban đầu.

Như vậy có tổng cộng bốn trạng thái  $\left(\frac{x}{y}\right)$  như sau:



**Hình 2.3 Sơ đồ các trạng thái điều khiển bóng đèn bằng nút nhấn**

– Lập bảng thống kê các trạng thái này:

Ở trạng thái thứ nhất ta có  $x = 0$ , điền 1 vào ô tương ứng với cột  $x = 0$ .

Trạng thái liền kề nó là 2, ta điền 2 vào ô kế bên. Các trạng thái khác tương tự.

Tiếp theo ta ghi trạng thái của đèn ( $y$ ). Với  $y = 0$  ở các trạng thái 1, 4 và  $y = 1$  ở các trạng thái 2, 3.

**Bảng 2.1: Bảng thống kê trạng thái điều khiển bóng đèn**

Trạng thái	x		y	
	0	1		
1	1	2	0	$Y_1$
2	3	2	1	$Y_2$

3	<b>3</b>	4	1	Y <sub>3</sub>
4	1	<b>4</b>	0	Y <sub>4</sub>

- Viết các điều kiện Set (S) và Reset (R) cho các quá trình:

S<sub>1</sub> là set Y<sub>1</sub>, R<sub>1</sub> là reset Y<sub>1</sub>. Các trạng thái kia tương tự.

Thiết lập phương trình S<sub>1</sub> bằng cách xét trạng thái được in đậm. Ta có trạng thái 1 được in đậm trong cột x = 0. Và các trạng thái cần set Y<sub>1</sub> là trạng thái đảo của Y<sub>2</sub> và đảo của Y<sub>3</sub>. Ta có:

$$S_1 = \bar{x} \cdot \bar{y}_2 \cdot \bar{y}_3$$

Thiết lập phương trình R<sub>1</sub> bằng cách xét các trạng thái không được in đậm. Ta có trạng thái 2 không được in đậm trong cột x = 1. Và trạng thái kế tiếp sau khi reset là Y<sub>2</sub>. Ta có:

$$R_1 = x \cdot y_2$$

Tương tự cho các trường hợp còn lại:

$$S_2 = x \cdot y_1$$

$$R_2 = \bar{x} \cdot y_3$$

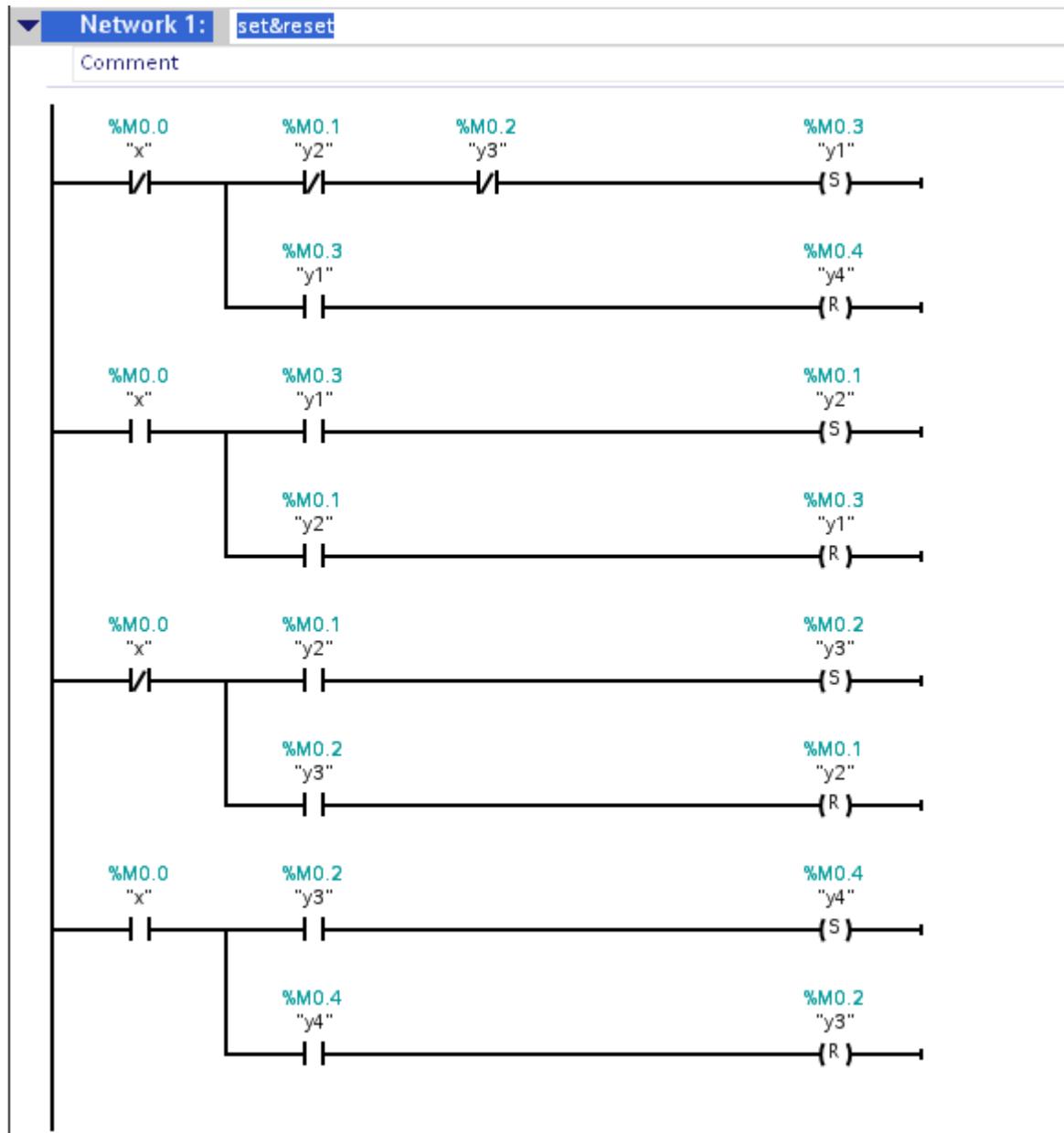
$$S_3 = \bar{x} \cdot y_2$$

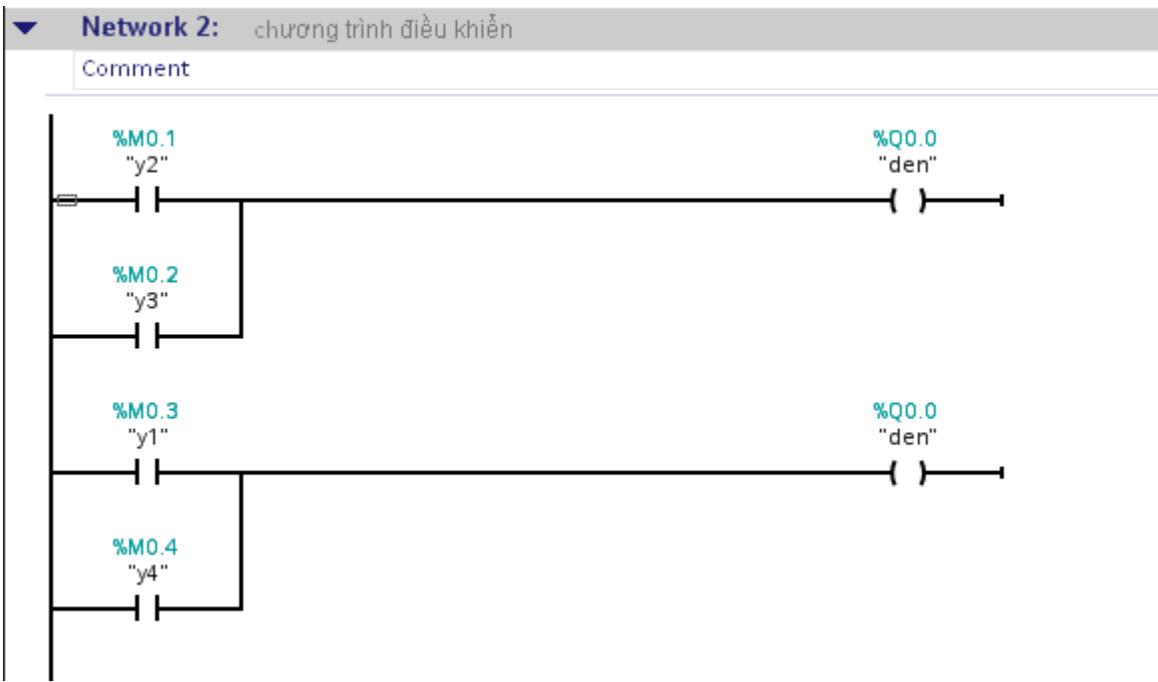
$$R_3 = x \cdot y_4$$

$$S_4 = x \cdot y_3$$

$$R_4 = \bar{x} \cdot y_1$$

- Cuối cùng, viết chương trình điều khiển dựa trên các điều kiện Set và Reset:





Hình 2.4 Chương trình điều khiển bóng đèn bằng nút nhấn

## **CHƯƠNG 3 TỔNG QUAN VỀ PLC VÀ BỘ PLC S7-1200**

### **3.1 TỔNG QUAN VỀ PLC**

#### **3.1.1 Giới Thiệu Về PLC**

##### **a. Giới thiệu**

PLC (Programmable Logic Controller) là dạng thiết bị điều khiển đặc biệt dựa trên bộ vi xử lý. PLC sử dụng bộ nhớ lập trình được để lưu trữ các lệnh và thực hiện các chức năng. Chẳng hạn cho phép tính logic, lập chuỗi, định giờ, đếm, và các thuật toán để điều khiển máy và các quá trình công nghệ.

PLC được thiết kế cho các kỹ sư, không yêu cầu cao về kiến thức máy tính và ngôn ngữ máy tính, có thể vận hành. Chúng được thiết kế cho các nhà kỹ thuật có thể cài đặt hoặc thay đổi chương trình. Vì vậy, các nhà thiết kế PLC phải lập trình sẵn sao cho chương trình điều khiển có thể nhập bằng cách sử dụng ngôn ngữ đơn giản (ngôn ngữ điều khiển).

Thuật ngữ logic được sử dụng vì việc lập trình chủ yếu liên quan đến các hoạt động logic, ví dụ nếu có các điều kiện A và B thì C làm việc... Người vận hành nhập chương trình (chuỗi lệnh) vào bộ nhớ PLC. Thiết bị điều khiển PLC sẽ giám sát các tín hiệu vào và các tín hiệu ra theo chương trình này và thực hiện các quy tắc điều khiển đã được lập trình.

PLC được chuyên biệt cho các tác vụ điều khiển và môi trường công nghiệp. Vì vậy các PLC:

- + Được thiết kế bền để chịu được rung động, nhiệt, ẩm và tiếng ồn,
- + Có sẵn giao diện cho các thiết bị vào ra,
- + Được lập trình dễ dàng với ngôn ngữ điều khiển dễ hiểu, chủ yếu giải quyết các phép toán logic và chuyển mạch.

Về cơ bản chức năng của bộ điều khiển logic PLC cũng giống như chức năng của bộ điều khiển thiết kế trên cơ sở các role hoặc các khối điện tử đó là:

- + Thu thập các tín hiệu vào và các tín hiệu phản hồi từ các cảm biến.

- + Liên kết, ghép nối các tín hiệu theo yêu cầu điều khiển và thực hiện đóng mở các mạch phù hợp với công nghệ.
- + Tính toán và soạn thảo các lệnh điều khiển trên cơ sở so sánh các thông tin thu thập được.
- + Phân phát các lệnh điều khiển đến các địa chỉ thích hợp.

### b. Ứng dụng

PLC được ứng dụng rất nhiều trong thực tế: trong các ngành công nghiệp cơ khí, đóng gói, chế tạo vật liệu, lắp ráp tự động, và rất nhiều ngành công nghiệp khác....

### c. Phân loại

PLC có các loại: Mitsubishi, Omron, Siemen, Nais.... Mỗi nhãn PLC lại có nhiều model khác nhau. Sự khác biệt giữa chúng chủ yếu là về khả năng hỗ trợ của CPU, số Port I/O, dung lượng nhớ.

- + PLC Siemens



**Hình 3.1 PLC Siemens**

- + PLC Omron



Hình 3.2 PLC loại Omron

- + PLC Panasonic



Hình 3.3 PLC loại Panasonic

### 3.1.2 Ngôn Ngữ Lập Trình PLC

#### a. Các loại ngôn ngữ lập trình cho PLC:

Có 3 loại cơ bản sau:

- + Hình thang(Ladder logic \_ LAD)
- + Khối hàm (Function Block Diagram \_ FBD).
- + Liệt kê câu lệnh (Statement List \_ STL).

#### b. Phần mềm hỗ trợ lập trình PLC

Mỗi loại PLC, tùy vào họ PLC và hãng sản xuất sẽ có phần mềm hỗ trợ lập trình riêng. Dưới đây là một tên một số phần mềm thông dụng:

Hãng Siemens :

- + Phần mềm lập trình Step 7 Micro, SIMATIC STEP 7.
- + Phần mềm SCADA : WinCC, WinCC Flexible.

- + Siemens OPC Server kết nối SCADA

Hãng Omron :

- + Phần mềm lập trình CX-Programmer PLC Omron.

Hãng LG Electronics :

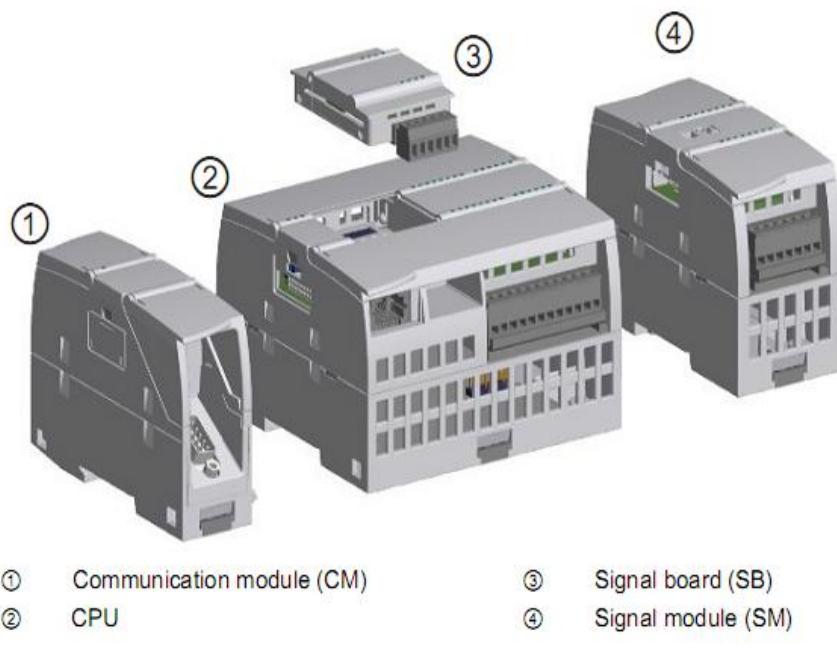
- + Phần mềm GMWIN lập trình PLC LG : Master-K, XGT Series

- + Phần mềm lập trình màn hình HMI LG : XP10, PMU 30, XGT

Series

## 3.2 TỔNG QUAN VỀ HỆ PLC S7-1200

### 3.2.1 Giới thiệu



**Hình 3.4: Hình dạng của PLC S7-1200 và các module mở rộng**

Hãng Siemens của Đức đã ra mắt một dòng PLC mới với những tính năng ưu việt hơn so với các PLC cùng loại trước đây. Đó là PLC S7-1200.

S7-1200 là một dòng PLC có thể điều khiển nhiều ứng dụng tự động hóa. CPU này tích hợp một bộ vi xử lý, một bộ nguồn, các mạch đầu vào, và mạch đầu ra nhỏ gọn.

Có thể chọn loại CPU được cấp điện áp nguồn là VAC hay VDC. Ngoài ra, S7-1200 cho phép mở rộng I/O bằng cách chọn các module cần thiết. S7-1200 bao gồm 9 module mở rộng. Các module tín hiệu mở rộng và các module giao tiếp dễ dàng kết nối với các mặt của bộ điều khiển. Tất cả các phần cứng Simatic S7-1200 có thể được gắn trên DIN rail tiêu chuẩn hay trực tiếp trên bảng điều khiển, giảm được không gian và chi phí đi kèm.

Các module tín hiệu có trong các model đầu vào, đầu ra và kết hợp loại 8, 16, và 32 điểm hỗ trợ các tín hiệu I/O DC, relay và analog. Bên cạnh đó, bảng tín hiệu tiên tiến có trong I/O số 4 kênh hay I/O analog 1 kênh gắn sẵn trước bộ điều khiển S7-1200 cho phép nâng cấp I/O mà không cần thêm không gian. Mỗi CPU đều có password bảo vệ cho phép truy cập vào các chức năng của CPU. Có thể sử dụng khôi phục đặc biệt để hiển thị hoặc ẩn chức năng " know-how protection".

Các đặc điểm khác: Bộ nhớ 50 KB với giới hạn giữa dữ liệu người sử dụng và dữ liệu chương trình, một đồng hồ thời gian thực, 16 vòng lặp PID với khả năng điều chỉnh tự động, cho phép bộ điều khiển xác định thông số vòng lặp gần tối ưu cho hầu hết các ứng dụng điều khiển quá trình thông dụng. Simatic S7-1200 cũng có một cổng giao tiếp Ethernet 10/100Mbit tích hợp với hỗ trợ giao thức Profinet cho lập trình, kết nối HMI /SCADA hay nối mạng PLC với PLC.

### **3.2.2 Ứng dụng của PLC S7-1200**

PLC S7-1200 được ứng dụng trong lĩnh vực điều khiển tự động như:

Điều khiển cửa tự động.

Hệ thống băng tải.

Đóng gói bao bì.

Quạt thông gió cho các nhà máy, xí nghiệp.

Trạm trộn xi măng và vữa.

Máy gia công gỗ.

Hệ thống nâng thủy lực.

Công nghiệp thực phẩm.

Phòng thí nghiệm.

### 3.2.3 Cấu Trúc Phần Cứng

#### ❖ Cấu trúc CPU

##### a. Phân loại:

Việc phân loại PLC S7-1200 dựa vào loại CPU mà nó được trang bị. Các loại CPU thông dụng: CPU 1211C, CPU 1212C, CPU 1214C.

Thông thường S7-1200 được phân ra thành hai loại chính:

+ Loại cấp điện 220VAC:

- Ngõ vào: Kích hoạt mức 1 = +24VDC (từ 15VDC – 30VDC)
- Ngõ ra: Relay
  - Ưu điểm của PLC loại này, là dùng ngõ ra relay. Do đó có thể sử dụng ngõ ra ở nhiều cấp điện áp khác nhau.
  - Tuy nhiên, do ngõ ra relay nên thời gian đáp ứng không nhanh cho các ứng dụng biến điệu độ rộng xung (PWM), hoặc output tốc độ cao...

+ Loại cấp điện áp 24VDC:

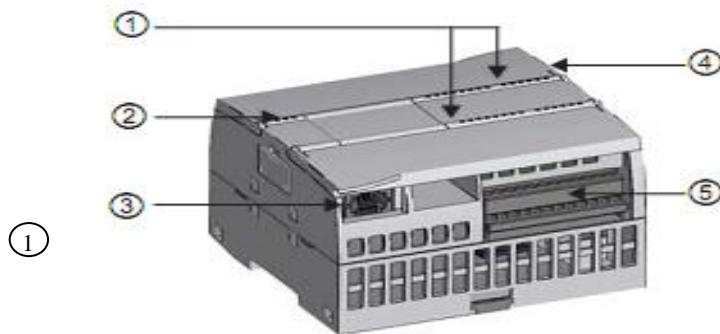
- Ngõ vào: Kích hoạt mức 1= +24VDC (từ 15VDC – 30VDC).
- Ngõ ra: Transistor
  - Ưu điểm: PLC loại này là dùng ngõ ra transistor. Do đó có thể sử dụng ngõ ra này để biến điệu độ rộng xung (PWM), output tốc độ cao...

**Bảng 3.1: Thông số kỹ thuật cơ bản của PLC**

Đặc điểm	CPU 1211C	CPU 1212C	CPU 1214C
Kích thước(mm)	90 x 100 x 75		110 x 100 x 75
Bộ nhớ:			
• Bộ nhớ làm việc • Bộ nhớ tải • Bộ nhớ lưu trữ	• 25 Kbytes • 1 Mbytes • 2 Kbytes		• 50 Kbytes • 2 Mbytes • 2 Kbytes
Số cổng logic vào/ra			

• Digital	• 6 ngõ vào/4 ngõ ra	• 8 ngõ vào/6 ngõ ra	• 14 ngõ vào/10 ngõ ra
• Analog	• 2 ngõ vào	• 2 ngõ vào	• 2 ngõ vào
Kích thước ảnh xử lý	1024 bytes (ngõ vào) và 1024 bytes (ngõ ra)		
Module tín hiệu mở rộng	Không	2	8
Board tín hiệu	1		
Module truyền thông	3 (mở rộng bên trái)		
Bộ đếm tốc độ cao	3	4	6
• Pha đơn	• 3 at 100kHz	• 3 at 100kHz 1 at 30kHz	• 3 at 100kHz 3 at 30kHz
• Pha vuông góc	• 3 at 80kHz	• 3 at 80kHz 1 at 20kHz	• 3 at 80kHz 3 at 20kHz
Ngõ ra xung	2		
Thẻ nhớ	Thẻ nhớ SIMATIC (tùy ý)		
Đồng hồ lưu trữ thời gian thực	10 ngày ở điều kiện thường/tối thiểu là 6 ngày ở 40 độ		
PROFINET	1 cổng truyền thông Ethernet		
Tốc độ thực thi lệnh	18 $\mu$ s /lệnh		
dấu chấm động			
Tốc độ thực thi lệnh bit	0.1 $\mu$ s /lệnh		

**b. Hình dạng bên ngoài:**



Hình 3.5: Hình dạng bên ngoài của PLC S7-1200 (CPU 1212C)

Trong đó:

- 1 - Trạng thái hoạt động của các ngõ I/O
- 2 - Chế độ hoạt động của PLC
- 3 - Cổng kết nối Profinet
- 4 - Khe cắm thẻ nhớ
- 5 - Nơi gắn dây nối

**c. Các đèn báo trên CPU 1200:**

STOP/RUN (cam/xanh): CPU ngừng / đang thực hiện chương trình đã nạp vào bộ nhớ.

ERROR (màu đỏ): Màu đỏ ERROR báo hiệu việc thực hiện chương trình đã xảy ra lỗi.

MAINT (maintenance): Led sáng báo hiệu việc có thẻ nhớ được gắn vào hay không.

LINK: Màu xanh báo hiệu việc kết nối với máy tính thành công.

Rx / Tx: Đèn vàng nhấp nháy báo hiệu tín hiệu được truyền.

Đèn cổng vào ra:

Ix.x (đèn xanh): Đèn xanh ở cổng vào báo hiệu trạng thái tức thời của cổng Ix.x. Đèn này báo hiệu trạng thái của tín hiệu theo giá trị của công tắc.

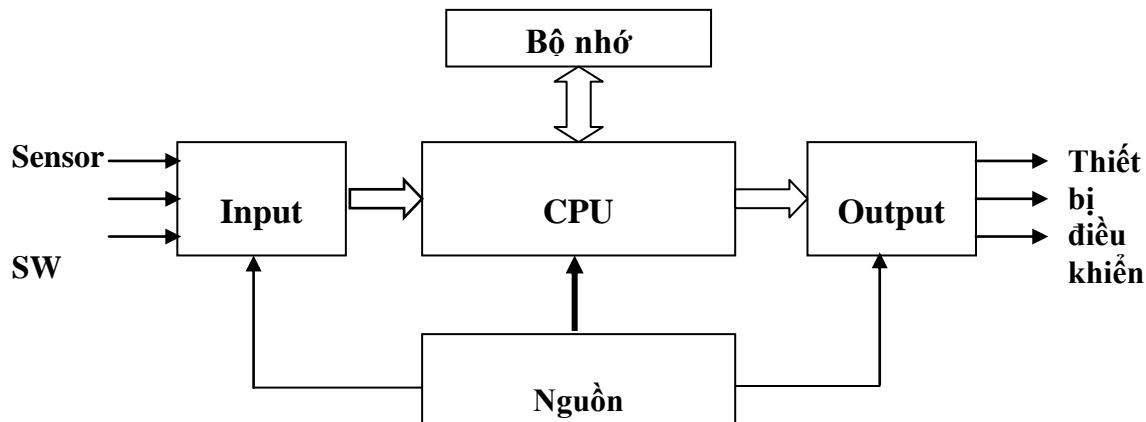
Qx.x (đèn xanh): Đèn xanh ở cổng ra báo hiệu trạng thái tức thời của cổng Qx.x. Đèn này báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.

**d. Cấu trúc bên trong:**

Cũng như các PLC cùng họ khác, PLC S7-1200 gồm 4 bộ phận cơ bản: bộ xử lý, bộ nhớ, bộ nguồn, giao tiếp xuất/nhập.

+ Bộ xử lý còn được gọi là bộ xử lý trung tâm(CPU), chứa bộ vi xử lý, biên dịch các tín hiệu nhập và thực hiện hoạt động điều khiển theo chương trình được lưu trong bộ nhớ của PLC. Truyền các quyết định dưới dạng tín hiệu hoạt động đến các thiết bị xuất.

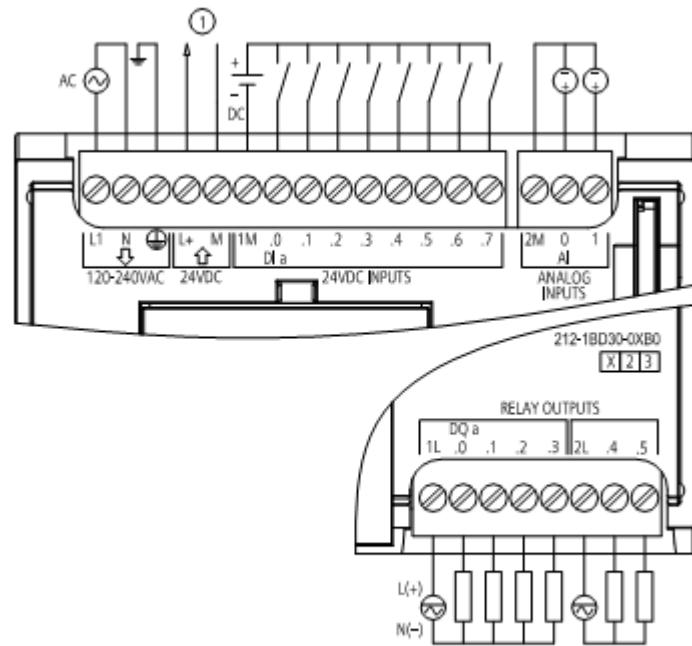
- + Bộ nguồn có nhiệm vụ chuyển đổi điện áp AC thành điện áp DC (24V) cần thiết cho bộ xử lý và các mạch điện trong các module giao tiếp nhập và xuất hoạt động.
- + Bộ nhớ là nơi lưu trữ chương trình được sử dụng cho các hoạt động điều khiển dưới sự kiểm soát của bộ vi xử lý.
- + Các thành phần nhập và xuất là nơi bộ nhớ nhận thông tin từ các thiết bị ngoại vi và truyền thông tin đến các thiết bị điều khiển. Tín hiệu nhập có thể từ công tắc, các bộ cảm biến... Các thiết bị xuất có thể là các cuộn dây của bộ khởi động động cơ, các val solenoid...
- + Chương trình điều khiển được nạp vào bộ nhớ nhờ sự trợ giúp của bộ lập trình hay bằng máy vi tính.



**Hình 3.6: Cấu trúc bên trong PLC**

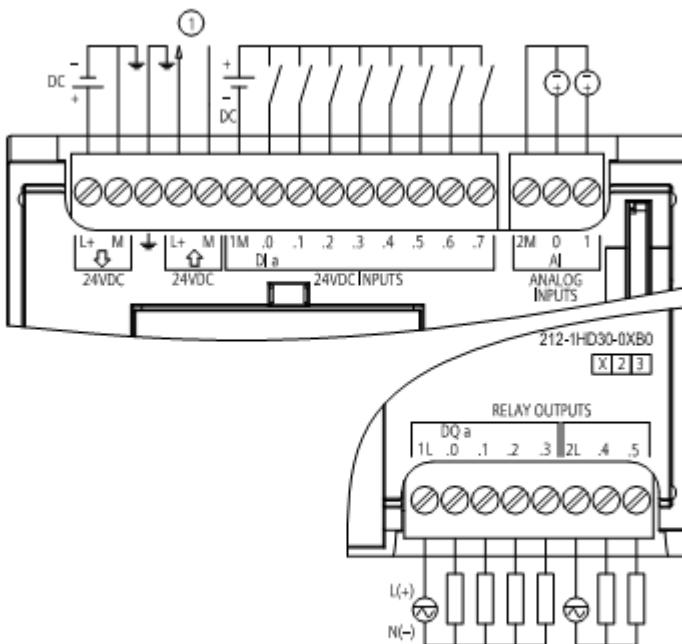
#### e. Sơ đồ đấu dây:

Chúng ta có thể cung cấp nguồn 24VDC hay 100 ~ 230 VAC cho PLC.



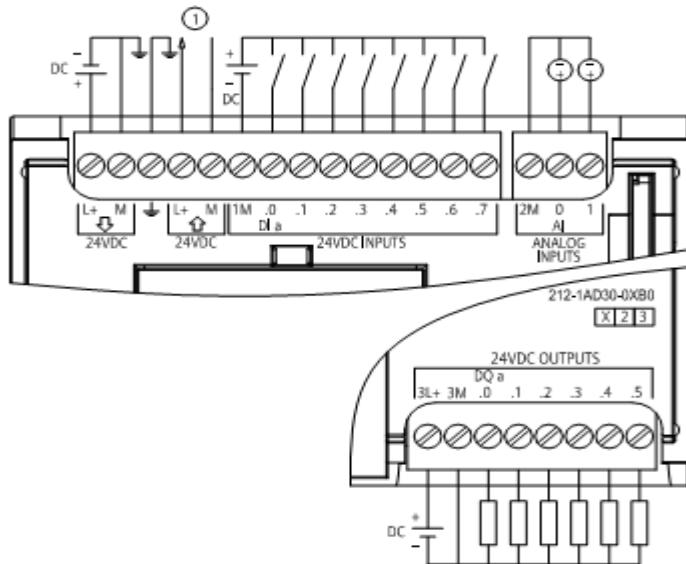
① 24 VDC Sensor Power Out

Figure A-4 CPU 1212C AC/DC Relay (6ES7 212-1BD30-0XB0)



① 24 VDC Sensor Power Out

Figure A-5 CPU 1212C DC/DC/Relay (6ES7 212-1HD30-0XB0)



① 24 VDC Sensor Power Out

Figure A-6 CPU 1212C DC/DC/DC (6ES7 212-1AD30-0XB0)

### Hình 3.7: Sơ đồ đấu dây PLC S7-1200 / CPU 1212C

Nguồn cung cấp cho PLC là 100 ~ 230 VAC với tần số từ 47Hz ~ 63Hz. Điện áp có thể thay đổi trong khoảng từ 85V ~ 264V. **Ở 240V dòng điện tiêu thụ là 120mA.**

Nguồn cung cấp là 24VDC điện áp có thể thay đổi trong khoảng từ 20.4V ~ 28.8V. **Ở 24V dòng điện tiêu thụ là 1.2A.**

Các ngõ vào được tác động ở mức điện thế tiêu biểu là 24VDC. Các ngõ ra của PLC ở mức 0 khi công tắc mở hay điện áp  $\leq 5$ VDC. Ngõ vào ở mức 1 khi công tắc đóng hay điện áp  $\geq 15$ VDC. Thời gian thay đổi trạng thái từ “0” lên “1” và từ “1” xuống “0” tối thiểu là **0.2ms** để PLC nhận biết được.

Các ngõ ra có thể là 5VDC – 30VDC hay 5VAC – 250VAC. Tuỳ theo yêu cầu thực tế mà ta có thể nối nguồn khác nhau để phù hợp với ứng dụng của nó

#### ❖ Module mở rộng:

Hệ PLC này cung cấp nhiều nhất 8 module tín hiệu và 1 mạch tín hiệu cho bộ xử lý có khả năng mở rộng. Ngoài ra, ta cũng có thể cài đặt thêm 3 module giao tiếp nhờ vào các giao thức truyền thông.

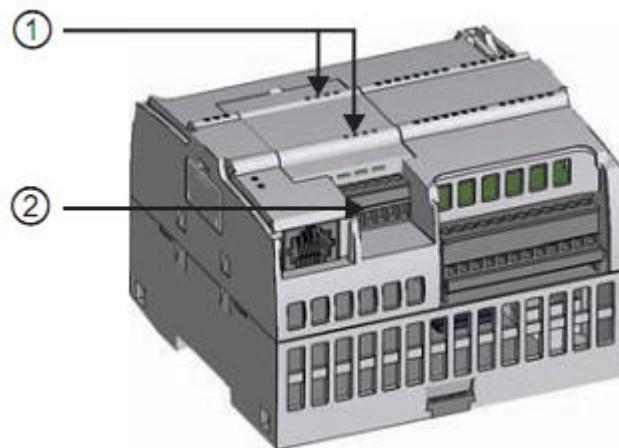
**Bảng 3.2: Thông số kỹ thuật của các module mở rộng**

Module		Ngõ vào	Ngõ ra	Kết hợp ngõ vào/ngõ ra
Module tín hiệu (SM)	Digital	8 x DC In	8 x DC Out 8 x Relay Out	8 x DC In/8 x DC Out 8 x DC In/8 x Relay Out
	Analog	4xAnalog In 8xAnalog In	2xAnalog Out 4xAnalog Out	4xAnalog In/2xAnalog Out
Board tín hiệu (SB)	Digital	–	–	2 x DC In/2 x DC Out
	Analog	–	1 x Analog Out	–
Module truyền thông (CM)				
<ul style="list-style-type: none"> <li>• RS485</li> <li>• RS232</li> </ul>				

### a. Signal Board (SB)

Signal Board cho phép thêm các ngõ vào/ ra digital hoặc analog cho CPU. SB được kết nối trên mặt trước của CPU.

- + SB có 4 ngõ vào/ra digital(2 ngõ vào DC và 2 ngõ ra DC).
- + SB có 1 ngõ ra analog.



**Hình 3.8: Sơ đồ đấu dây PLC S7-1200 / CPU 1212C**

- 1 - Đèn báo tình trạng trên SB
- 2 - Đầu nối dây có thể tháo mở được

Thông số kỹ thuật:

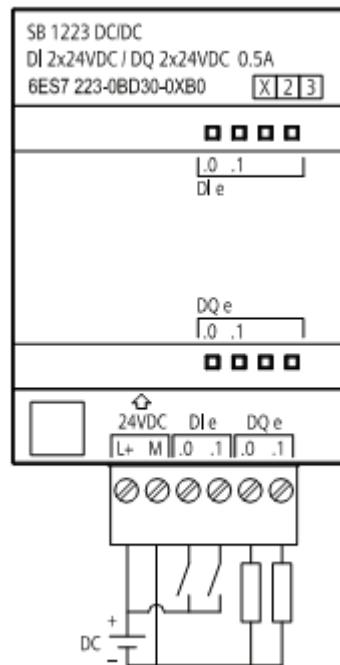
+ **SB 1223 với 2 ngõ vào 24VDC/2 ngõ ra 24VDC**

**Bảng 3.3: Các thông số kỹ thuật của board tín hiệu Digital**

Thông số kỹ thuật	
Model	SB 1223 DI 2x24VDC, DQ 2x24VDC
Số hiệu	6ES 223-0BD30-0XB0
Tổng quát	
Kích thước W x H x D(mm)	38 x 62 x21
Trọng lượng	40 grams
Công suất tiêu thụ	1.0 W
Dòng tiêu thụ(SM Bus)	50 mA
Dòng tiêu thụ(24 VDC)	4mA/đầu vào
Ngõ vào Digital	
Số ngõ vào	2
Loại	IEC Type 1 sink
Điện áp định mức	24 VDC ở 4mA, danh nghĩa
Điện áp cho phép liên tục	Tối đa 30 VDC
Điện áp xung	35 VDC trong 0,5s
Tín hiệu mức logic 1(tối thiểu)	15 VDC ở 2,5 mA
Tín hiệu mức logic 0(tối đa)	5 VDC ở 1mA
Mức tạo xung đầu vào HSC(tối đa)	20kHz(15 đến 30 VDC) 30kHz(15 đến 26 VDC)
Số ngõ vào đồng thời	2
Chiều dài cáp(mét)	Được bảo vệ là 500, không được bảo vệ là 300
Ngõ ra Digital	
Số ngõ ra	2
Loại	Solid state – MOSFET
Khoảng điện áp	20.4VDC đến 28.8 VDC
Tín hiệu mức logic 1	Tối thiểu 20 VDC

Tín hiệu mức logic 0 với tải 10k $\Omega$	Tối đa 0.1 VDC
Cường độ dòng điện (tối đa)	0.5A
Dòng điện rò trên diode	Tối đa 10uA
Mức xung đầu ra	Tối đa 20 kHz, tối thiểu 2Hz
Dòng điện xung	Tối đa 5A ở 100ms
Bảo vệ quá tải	Không
Độ trễ chuyển mạch	Tối đa 2 us để từ off sang on Tối đa 10 us để từ on sang off
Số ngõ ra đồng thời	2
Chiều dài cáp(mét)	Được bảo vệ là 500, không được bảo vệ là 150

Sơ đồ đấu dây:



Hình 3.9: Sơ đồ đấu dây SB 1223 DC/DC

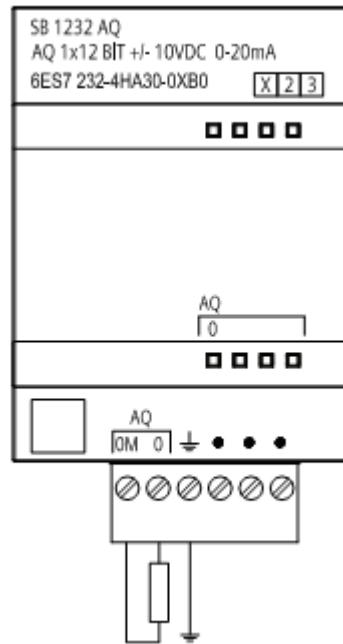
+ SB 1232 một ngõ ra Analog.

Bảng 3.4: Các thông số kỹ thuật của board tín hiệu Analog

Thông số kỹ thuật	
Model	SB 1223 AQ 1x12bit

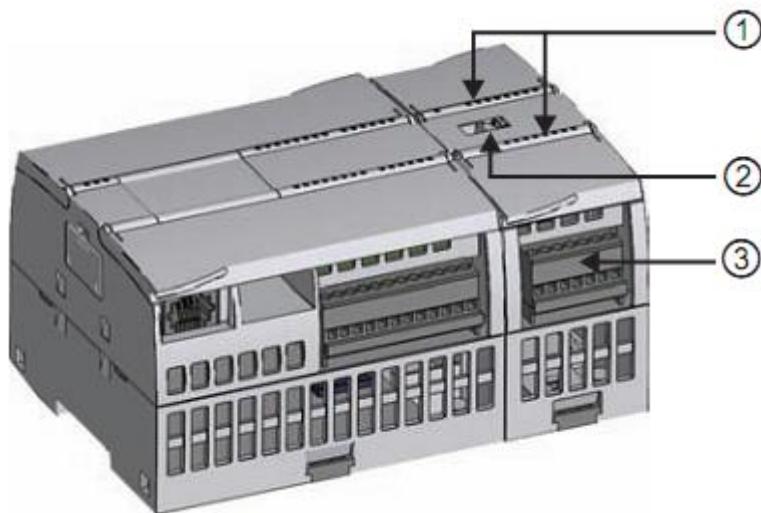
Số hiệu	6ES7 232-4HA30-0XB0
Tổng quát	
Kích thước W x H x D(mm)	38 x 62 x 21
Trọng lượng	40 grams
Công suất tiêu thụ	1.5 W
Dòng tiêu thụ(SM Bus)	15 mA
Dòng tiêu thụ(24 VDC)	40 mA(không tải)
Ngõ ra Analog	
Số ngõ ra	1
Loại	điện áp hoặc dòng điện
Khoảng giá trị	$\pm 10\text{V}$ hoặc 0 mA – 20 mA
Độ phân giải	Điện áp: 12 bits Dòng điện: 11 bits
Khoảng tỷ lệ tối đa(data word)	Điện áp: từ -27648 đến 27648 Dòng điện từ 0 đến 27648
Độ chính xác( $25^{\circ}\text{C}/0$ đến $55^{\circ}\text{C}$ )	$\pm 0.5\% / \pm 1\%$ của tỷ lệ tối đa
Thời gian thiết lập(95% giá trị mới)	Điện thế: $300\text{ }\mu\text{S (R)}$ , $750\text{ }\mu\text{S (1 uF)}$ Dòng điện: $600\text{ }\mu\text{S (1 mH)}$ , 2 ms (10 mH)
Tổng trở tải	Điện thế: $\geq 1000\text{ }\Omega$ Dòng điện: $\leq 600\text{ }\Omega$
Hoạt động từ RUN sang STOP	Giá trị sau cùng hoặc giá trị thay thế (giá trị mặc định là 0)
Chiều dài cáp(mét)	10 mét, xoắn và được bảo vệ
Hệ chẩn đoán	
Overflow/underflow	Có
Nối đất	Có
Wire break (chỉ ở chế độ dòng điện)	Có

Sơ đồ đấu dây:



Hình 3.10:Sơ đồ đấu dây SB 1232AQ

b. Signal modules(SM)



Hình 3.11:Sơ đồ đấu dây SM 1232

- 1 - Đèn trạng thái I/O của module tín hiệu
- 2 - Đầu nối Bus
- 3 - Đầu nối dây có thể tháo mở được

Module tín hiệu được sử dụng để bổ xung thêm chức năng cho CPU.  
Module tín hiệu kết nối phía bên phải của CPU.

Thông số kỹ thuật:

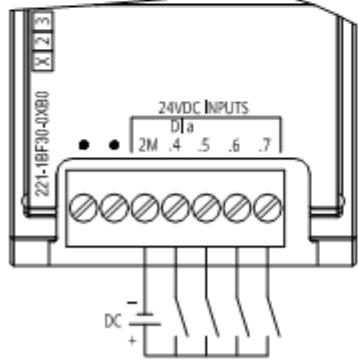
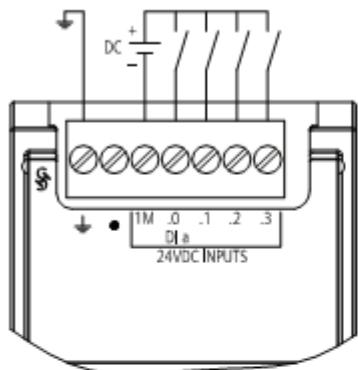
+ **SM 1221 ngõ vào Digital**

**Bảng 3.5: Các thông số của SM 1221 ngõ vào Digital**

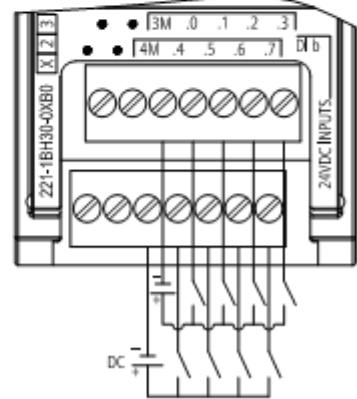
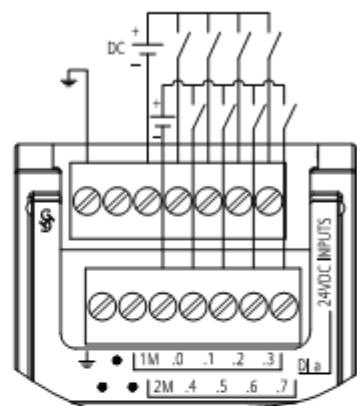
Thông số kỹ thuật		
Model	SM 1221 DI 8x24VDC	SM 1221 DI 8x24VDC
Số hiệu	6ES 221-1BF30-0XB0	6ES 221-1BH30-0XB0
Tổng quát		
Kích thước WxDxH(mm)	45x100x75	
Trọng lượng	170 grams	210 grams
Công suất tiêu thụ	1.5W	2.5W
Dòng tiêu thụ(SM Bus)	105 mA	130 mA
Dòng tiêu thụ(24 VDC)	4 mA/ ngõ vào	
Các ngõ vào Digital		
Số ngõ vào	8	16
Loại	Sink/Source(IEC Type 1 sink)	
Điện áp định mức	24 VDC ở 4mA	
Điện áp cho phép liên tục	Tối đa 30 VDC	
Điện áp xung	35 VDC trong 0.5s	
Tín hiệu mức logic 1	15 VDC ở 2.5 mA	
Tín hiệu mức logic 0	5 VDC ở 1 mA	
Số ngõ vào đồng thời	8	16
Chiều dài cáp (mét)	Được bảo vệ là 500, không được bảo vệ là 300	

Sơ đồ đấu dây:

**SM 1221 DI 8 x 24 VDC**



**SM 1221 DI 16 x 24 VDC**



**Hình 3.12: Sơ đồ đấu dây SM 1221**

+ **SM 1222 ngõ ra Digital**

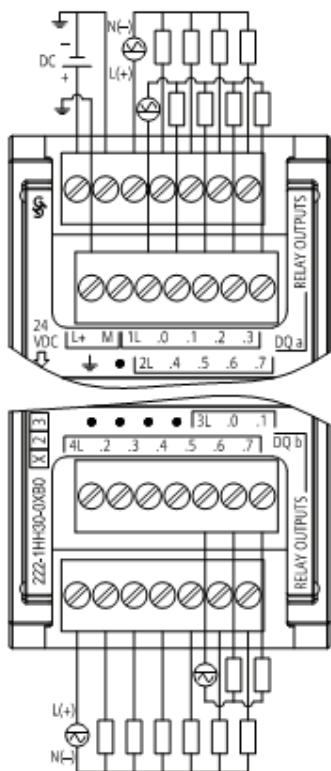
**Bảng 3.6: Các thông số kỹ thuật của SM 1222 ngõ ra Digital**

Thông số kỹ thuật				
Model	SM 1222 DQ 8xRelay	SM 1222 DQ 16xRelay	SM 1222 DQ 8x24 VDC	SM 1222 DQ 16x24 VDC
Số hiệu	6ES7 222- 1HF30-0XB0	6ES7 222- 1HH30-0XB0	6ES7 222- 1BF30-0XB0	6ES7 222- 1BH30-0XB0
Tổng quát				
Kích thước WxHxD(mm)	45x100x75			
Trọng lượng	190 grams	260 grams	180 grams	220 grams
Công suất tiêu thụ	4.5 W	8.5W	1.5W	2.5W
Dòng tiêu thụ(SM Bus)	120 mA	135 mA	120 mA	140 mA
Dòng tiêu thụ (24VDC)	11mA/ ngõ ra Rờ le			
Ngõ vào digital				
Số ngõ vào	8	16	8	16
Loại	Sink/Source(IEC Type 1 sink)			
Điện áp định mức	24 VDC ở 4mA			
Điện áp cho phép liên tục	Tối đa 30 VDC			
Điện áp xung	35 VDC trong 0.5s			
Tín hiệu mức logic 1	15 VDC ở 2.5 mA			
Tín hiệu mức logic 0	5 VDC ở 1 mA			
Cách điện	500 VAC trong 1 phút			
Nhóm cách điện	2			
Số ngõ vào đồng thời	8	16	8	16
Chiều dài cáp(mét)	Được bảo vệ là 500, không được bảo vệ là 150			
Ngõ ra Digital				
Số ngõ ra	8	16	8	16
Loại	Rờ le,tiếp điểm		Solid state-MOSFET	
Mức điện áp	5VDC đến 30VDC hoặc 5VAC đến 250VAC		20.4 đến 28.8	
Tín hiệu mức logic 1 ở dòng tối đa				Tối thiểu 20 VDC
Tín hiệu mức logic 0 với tải <b>10K Ω</b>				Tối đa 0.1 VDC

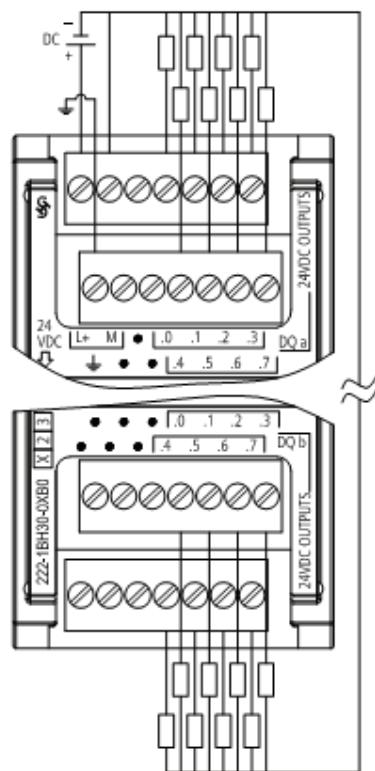
Dòng điện (tối đa)	2.0A	0.5A		
Đèn tải	30W DC/200W AC	5W		
Điện trở tiếp xúc ở trạng thái đóng	Tối đa <b>0.2 Ω</b>	Tối đa <b>0.6 Ω</b>		
Dòng điện rò trên điểm		Tối đa <b>10 μA</b>		
Dòng điện xung	7A	Tối đa 8A trong 10ms		
Bảo vệ quá tải	Không			
Cách điện	1500 VAC trong 1 phút (ngõ ra tiếp điểm) Không cách điện (ngõ ra logic)	500 VAC trong 1 phút		
Điện trở cách điện	Tối thiểu <b>100 MΩ</b>			
Độ trễ chuyển mạch	Tối đa 10 ms	Tối đa <b>50 μs</b> từ off sang on Tối đa <b>200 μs</b> từ on sang off		
Hoạt động từ RUN sang STOP	Giá trị sau cùng hoặc giá trị thay thế (giá trị mặc định là 0)			
Số ngõ ra đồng thời	8	16	8	16
Chiều dài cáp (mét)	Được bảo vệ là 500, không được bảo vệ là 150			

Sơ đồ đấu dây:

**SM 1222 DQ 16 x Relay**



**SM 1222 DQ 16 x 24 VDC**



**Hình 3.13: Sơ đồ đấu dây SM 1222**

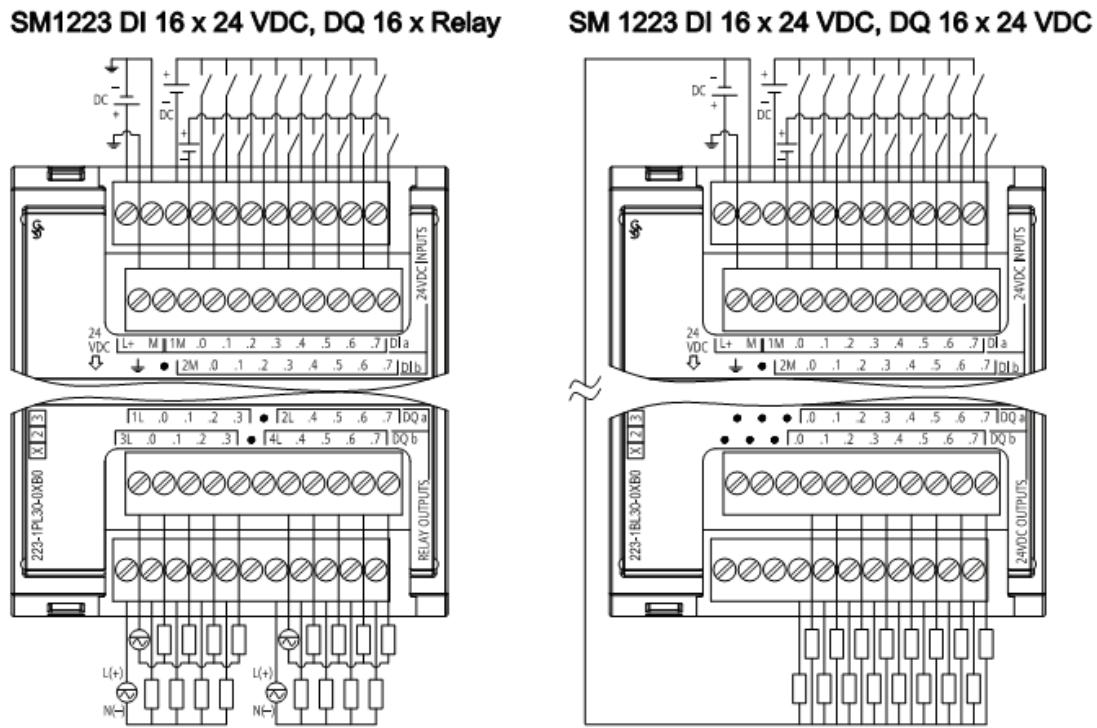
+ **SM 1223 ngõ vào/ngõ ra Digital**

**Bảng 3.7: Thông số kỹ thuật của SM 1223 ngõ vào/ngõ ra digital**

Thông số kỹ thuật				
Model	SM 1223 DI 8x24 VDC, DQ 8xRelay	SM 1223 DI 16x24 VDC, DQ	SM 1222 DI 8x24 VDC, DQ 8x24 VDC 16xRelay	SM 1222 DI 16x24 VDC , DQ 16x24 VDC
Số hiệu	6ES7 223- 1PH30-0XB0	6ES7 223-1PL30- 0XB0	6ES7 223- 1BH30-0XB0	6ES7 222- 1BL30-0XB0
Tổng quát				
Kích thước WxHxD(mm)	45x100x75	70x100x75	45x100x75	70x100x75
Trọng lượng	230 grams	350 grams	210 grams	310 grams

Công suất tiêu thụ	5.5 W	10W	2.5W	4.5W
Dòng tiêu thụ(SM Bus)	145 mA	180 mA	145 mA	180 mA
Dòng tiêu thụ (24VDC)	11mA/ ngõ ra Rờ le 4 mA/ngõ ra			
Ngõ ra Digital				
Số ngõ ra	8	16	8	16
Loại	Rờ le, tiếp điểm		Solid state-MOSFET	
Mức điện áp	5VDC đến 30VDC hoặc 5VAC đến 250VAC		20.4 đến 28.8	
Tín hiệu mức logic 1 ở dòng tối đa			Tối thiểu 20 VDC	
Tín hiệu mức logic 0 với tải <b>10K Ω</b>			Tối đa 0.1 VDC	
Dòng điện (tối đa)	2.0A		0.5A	
Đèn tải	30W DC/200W AC		5W	
Điện trở tiếp xúc ở trạng thái đóng	Tối đa <b>0.2 Ω</b>		Tối đa <b>0.6 Ω</b>	
Dòng điện rò trên điểm			Tối đa <b>10 μA</b>	
Dòng điện xung	7A		Tối đa 8A trong 10ms	
Bảo vệ quá tải	Không			
Cách điện	1500 VAC trong 1 phút (ngõ ra tiếp điểm) Không cách điện (ngõ ra logic)		500 VAC trong 1 phút	
Điện trở cách điện	Tối thiểu <b>100 MΩ</b>			
Độ trễ chuyển mạch	Tối đa 10 ms		Tối đa <b>50 μs</b> từ off sang on Tối đa <b>200 μs</b> từ on sang off	
Hoạt động từ RUN sang STOP	Giá trị sau cùng hoặc giá trị thay thế(giá trị mặc định là 0)			
Số ngõ ra đồng thời	8	16	8	16
Chiều dài cáp(mét)	Được bảo vệ là 500, không được bảo vệ là 150			

Sơ đồ đấu dây:



Hình 3.14: Sơ đồ đấu dây SM 1223

### + Module tin hiệu analog

Bảng 3.8: Thông số kỹ thuật của SM 1231, SM 1232, SM 1234

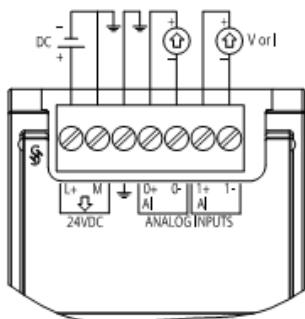
Thông số kỹ thuật			
Model	SM 1231 AI 4x13bits	SM 1234 AI 4x13 bit AQ 2x14bit	SM 1232 AQ 2x14bit
Số hiệu	6ES7 231-4HD30-0XB0	6ES7 234- 4HE30-0XB0	6ES7 231-4HB30- 0XB0
Tổng quát			
Kích thước WxHxD(mm)	45x100x75		
Trọng lượng	180grams	220 grams	180 grams
Công suất tiêu thụ	1.5W	2.0W	1.5W

Dòng tiêu thụ(SM Bus)	80 mA		
Dòng tiêu thụ (24VDC)	45 mA	60 mA(không tải)	45 mA(không tải)
Ngõ vào digital			
Số ngõ vào	4		0
Loại	Điện áp hoặc dòng điện		
Khoảng hoạt động	$\pm 10\text{ V}$ , $\pm 5\text{ V}$ , $\pm 2.5\text{ V}$ , or 0 to 20 mA		
Khoảng tỷ lệ tối đa	-27648 đến 27648		
Độ phân giải	12 bits		
Dòng điện/điện áp tối đa	$\pm 35\text{ V}$ / $\pm 40\text{ mA}$		
Loại bù tiếng ồn	400, 60, 50, hoặc 10 Hz		
Trở kháng	$\geq 9\text{ M}\Omega$ (điện áp) / $250\text{ }\Omega$ (dòng điện)		
Cách điện	Không		
Độ chính xác ( $25^\circ\text{C}$ / 0 đến $55^\circ\text{C}$ )	$\pm 0.1\%$ / $\pm 0.2\%$ của tỷ lệ tối đa		
Thời gian chuyển đổi từ analog sang digital	$625\text{ }\mu\text{s}$		
Chiều dài cáp(mét)	10 mét		
Ngõ ra Digital			
Số ngõ ra	0	2	
Loại	-	Điện áp hoặc dòng điện	
Khoảng hoạt động	-	$\pm 10\text{ V}$ hoặc từ 0 mA đến 20 mA	
Độ phân giải	-	Điện áp: 14 bits Dòng điện: 13bits	
Khoảng tỷ lệ tối đa	-	Điện áp từ: -27468 đến 27468 Dòng điện từ: 0 đến 27468	
Độ chính xác ( $25^\circ\text{C}$ / 0 đến $55^\circ\text{C}$ )	-	$\pm 0.3\%$ / $\pm 0.6\%$ của tỷ lệ tối đa	
Thời gian thiết lập	-	Điện áp: $300\text{ }\mu\text{s (R)}$ , $750\text{ }\mu\text{s (1 uF)}$ Dòng điện:	

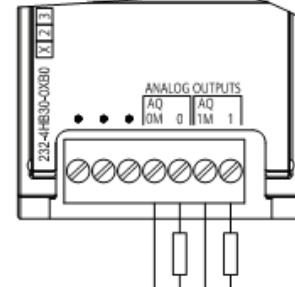
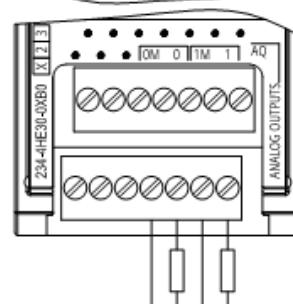
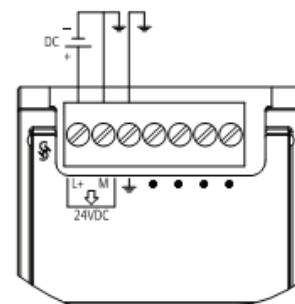
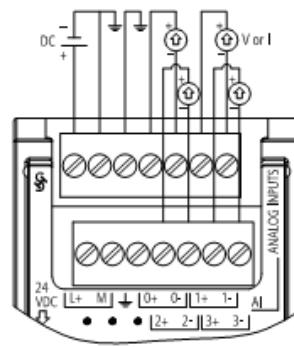
		600 $\mu$ S (1 mH), 2 ms (10 mH)
Tổng trờ tải	–	Điện áp: $\geq 1000 \Omega$ Dòng điện: $\leq 600 \Omega$
Hoạt động từ RUN sang STOP	–	Giá trị sau cùng hoặc giá trị thay thế (giá trị mặc định là 0)
Cách điện	–	Không
Chiều dài cáp	–	10 mét
Hệ chuẩn đoán		
Overflow/underflow	Có	Không áp dụng
Nối đất (chỉ ở chế độ điện áp)	Không	Có ở ngõ ra
Wire break (chỉ ở chế độ dòng điện)	Không	Có ở ngõ ra
Điện áp thấp	Có	

Sơ đồ đấu dây:

**SM 1231 AI 4 x 13 Bit**



**SM 1234 AI 4x13bit AQ 2x14bit    SM 1232 AQ 2 x 14 bit**



**Hình 3.15: Sơ đồ đấu dây của SM 1231, SM 1232, SM 1234**

### c. Modules truyền thông (CM)

Hệ S7-1200 cung cấp các modules truyền thông (CM) để bổ sung thêm chức năng cho hệ thống. Có 2 module truyền thông: RS232 và RS485.

- + CPU hỗ trợ tối đa 3 module truyền thông.
- + Mỗi CM kết nối vào bên trái của CPU (hoặc bên trái của CM khác kết nối với CPU).



**Hình 3.16: Module Comunication**

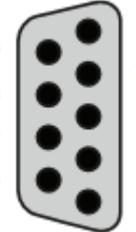
Thông số kỹ thuật:

**Bảng 3.9: Thông số kỹ thuật của CM 1241 RS 485**

Số hiệu	6ES 241-1CH30-0XB0
<b>Kích thước và trọng lượng</b>	
Kích thước (mm)	30X100X75
Trọng lượng	150 GRAM
<b>Truyền và nhận</b>	
Khoảng điện áp	-7V đến +12V, 1 giây, 3 giây VRMS
Chênh lệch điện áp	Tối thiểu 2V ở $R_L=100 \Omega$ Tối thiểu 1.5V ở $R_L=54 \Omega$
Trở kháng ngõ vào	Tối thiểu 5.4 KΩ
Nguồng nhận/độ nhạy	Tối thiểu +/- 0.2 V, 60mA trễ điển hình
Cách điện	

Tín hiệu RS485 đến dây đất	500 VAC trong 1 phút
Tín hiệu RS485 đến CPU logic chung	
Chiều dài cáp có vỏ bọc	Tối đa 1000 mét
<b>Đặc điểm kỹ thuật bộ nguồn</b>	
Công suất tiêu hao	1.1 W
Từ +5 VDC	220 mA

**Bảng 3.10: Chức năng của các chân RS485**

Chân	Chức năng	Đầu nối	Chân	Chức năng
1 GND	Nối mass		6	+5V với điện trở 100Ω
2	Không kết nối		7	Không kết nối
3 TxD+	Tín hiệu B (RxD/TxD+): Ngõ vào/ngõ ra		8	Tín hiệu A (RxD/TxD-): Ngõ vào/ngõ ra
4 RTS	RTS(theo mức TTL)		9	Không kết nối
5 GND	Nối mass		Vỏ	Chassis ground

**Bảng 3.11: Thông số kỹ thuật của CM 1241 RS 232**

Thông số kỹ thuật	
Số hiệu	6ES7 241-1AH-0XB0
Kích thước và trọng lượng	
Kích thước (mm)	30x100x75
Trọng lượng	150 grams
Truyền và nhận	
Điện áp đầu ra bộ truyền	Tối thiểu +/-5V ở $R_L=3k\Omega$
Điện áp đầu ra truyền tải	Tối đa +/-15V VDC
Trở kháng đầu vào	Tối thiểu 3kΩ
Nguồn nhận/độ nhạy	Thấp nhất 0.8V cao nhất 2.4V
Điện áp đầu vào bộ nhận	Tối đa +/- 30 VDC

Cách điện	500 VAC trong 1 phút
Tín hiệu RS232 đến dây đất	
Tín hiệu RS232 đến CPU logic chung	
Chiều dài cáp có vỏ bọc	Tối đa 10 mét
Đặc điểm kỹ thuật bộ nguồn	
Công suất tiêu hao	1.1W
Từ +5 VDC	220 mA

#### d. Thẻ nhớ

Một thẻ nhớ tùy ý có thể được sử dụng như một thẻ chương trình hoặc một thẻ lệnh. Thẻ nhớ do siement định dạng trước:



**Hình 3.17: Thẻ nhớ**

- + Thẻ chương trình: thẻ nhớ được đưa vào vị trí bộ nhớ của CPU, tất cả các chức năng của CPU được điều khiển bằng thẻ. Yêu cầu thẻ nhớ phải được đặt trong CPU.
- + Thẻ chuyển lệnh: Thẻ nhớ được sử dụng để chuyển dữ liệu dự án được lưu trữ trong thẻ đến CPU, sau đó thẻ được tháo ra. Bằng cách này, ta có thể chuyển dữ liệu dự án đến nhiều CPU mà không cần Step 7 Basic.

Thẻ nhớ có 2 loại: loại 2MB và loại 24MB. Để lắp thẻ nhớ, ta mở nắp phía trên của CPU và lắp thẻ nhớ vào khe.

#### e. Màn hình HMI

Sự hiển thị trở thành một thành phần tiêu chuẩn cho hầu hết các thiết kế máy, bảng điều khiển SIMATIC HMI Basic cung cấp các thiết bị màn hình cảm ứng để điều khiển và giám sát công việc.



**Hình 3.18: Màn hình HMI**

**Bảng 3.12: Thông số kỹ thuật của màn hình HMI**

Đặc trưng	KTP600 Basic color	TP1500 Basic color
Màn hình +Kích thước + Độ phân giải	TFT, 256 colors +5.7" +320x240	TFT, 256 colors 15.0" 1024x768
Phần tử điều khiển	Màn hình cảm ứng + 8 phím	Màn hình cảm ứng
Đánh giá sự bảo vệ	IP65	IP65
Giao diện	PROFINET	PROFINET
Chức năng +Tag +Process screens +Alarms +Trend curves	256 50 200 25	256 50 200 25
Kích thước +Mặt trước vỏ (rộng x cao x sâu) +Mounting cut-out (Rộng x cao)	196 x 140 x 60 214 x 158	400 x 310 x 60 367 x 289

### 3.2.4 Các kiểu dữ liệu và truy xuất dữ liệu trong S7-1200

#### a. Truy xuất dữ liệu trong S7-1200:

Khi thực thi lệnh, PLC sẽ cập nhật giá trị tại các ô nhớ có địa chỉ yêu cầu. Bộ nhớ của PLC được chia làm ba vùng nhớ theo chức năng:

- + Vùng nhớ toàn cục: Bao gồm các ô nhớ input (I), output (Q) và bit nhớ (M). Dữ liệu sẽ được thay đổi liên tục bởi các khối chức năng. Và dữ liệu sẽ mất khi khởi động lại PLC.
- + Khối dữ liệu (DB): Vùng nhớ này sẽ lưu code chương trình. Dữ liệu sẽ được bảo toàn khi thực hiện các khối chức năng hoặc khởi động lại PLC.
- + Vùng nhớ tạm (L): Là vùng nhớ để PLC sao chép code để thực thi các khối chức năng.

Mỗi một vùng nhớ được đặt tên địa chỉ khác nhau và cách tác động khác nhau:

**Bảng 3.13: Các vùng nhớ trong PLC S7-1200**

Vùng nhớ	Mô tả	Tác động ngoài	Tác động trong chương trình
<b>I</b> Image input	Copy các giá trị từ ngõ vào vật lý để bắt đầu chu kỳ quét	Không	Không
<b>I_P</b> Physical input	Đọc giá trị từ các ngõ vào của CPU	Được	Không
<b>Q</b> Image output	Copy các giá trị ra ngõ ra vật lý để bắt đầu chu kỳ quét	Không	Không
<b>Q_P</b> Physical output	Ghi giá trị ra các ngõ ra của CPU	Được	Không
<b>M</b> Bit memory	Lưu giá trị	Không	Được
<b>L</b> Temp memory	Dữ liệu tạm thời	Không	Không
<b>DB</b> Data block	Dữ liệu và tham số nhớ cho FBs	Không	Được

### b. Các kiểu dữ liệu trong S7-1200:

Khi viết chương trình ta sử dụng các lệnh. Mỗi lệnh gồm một hoặc nhiều ngõ vào và ngõ ra. Mỗi ngõ vào hoặc ngõ ra được định nghĩa bởi một tag. Mỗi tag có một tên và một ô nhớ riêng. Bên cạnh đó, mỗi tag cũng được định nghĩa một kiểu dữ liệu phù hợp với mục đích sử dụng.

Để viết được chương trình ta phải hiểu rõ các kiểu dữ liệu. Dưới đây là một số kiểu dữ liệu [5] (trang 53 đến trang 60) mà S7-1200 có hỗ trợ:

#### Dữ liệu kiểu Bool, Byte, Word, and DWord

Bảng 3.14: Bảng thông số kiểu dữ liệu Bool, Byte, Word, DWord

Kiểu dữ liệu	Số bit	Kiểu số	Giai	Ví dụ giá trị	Ví dụ địa chỉ
Bool	1	Boolean	FALSE or TRUE	TRUE, 1,	I1.0 Q0.1 M50.7 DB1.DBX2.3 Tag_name
		Binary	0 or 1	0, 2#0	
		Octal	8#0 or 8#1	8#1	
		Hexadecimal	16#0 or 16#1	16#1	
Byte	8	Binary	2#0 to 2#11111111	2#00001111	IB2 MB10 DB1.DBB4 Tag_name
		Unsigned integer	0 to 255	15	
		Octal	8#0 to 8#377	8#17	
		Hexadecimal	B#16#0 to B#16#FF	B#16#F, 16#F	
Word	16	Binary	2#0 to 2#1111111111111111	2#11110000111 10000	MW10 DB1.DBW2 Tag_name
		Unsigned integer	0 to 65535	61680	
		Octal	8#0 to 8#177777	8#170360	
		Hexadecimal	W#16#0 to W#16#FFFF, 16#0 to 16#FFFF	W#16#F0F0, 16#F0F0	
DWord	32	Binary	2#0 to 2#1111111111111111 1111111 11111111	2#11110000111 1111100 001111	MD10 DB1.DBD8 Tag_name
		Unsigned integer 0	0 to 4294967295	15793935	
		Octal	8#0 to 8#3777777777	8#74177417	
		Hexadecimal	DW#16#0000_0000 to DW#16#FFFF_FFFF, 16#0000_0000 to 16#FFFF_FFFF	DW#16#F0FF0 F, 16#F0FF0F	

## Dữ liệu kiểu Integer

Bảng 3.15: Bảng thông số kiểu dữ liệu Integer

Kiểu dữ liệu	Số bit	Giai	Ví dụ giá trị	Ví dụ địa chỉ
USInt	8	0 to 255	78, 2#01001110	MB0,
SInt	8	-128 to 127	+50, 16#50	DB1.DBB4, Tag_name
UInt	16	0 to 65,535	65295, 0	MW2,
Int	16	-32,768 to 32,767	30000, +30000	DB1.DBW2, Tag_name
UDInt	32	0 to 4,294,967,295	4042322160	MD6,
DInt	32	-2,147,483,648 to 2,147,483,647	-2131754992	DB1.DBD8, Tag_name

## Dữ liệu kiểu Floating-point real

Bảng 3.16: Bảng thông số kiểu dữ liệu Floating-point real

Kiểu dữ liệu	Số bit	Giai	Ví dụ giá trị	Ví dụ địa chỉ
Real	32	-3.402823e+38 to -1.175 495e-38, ±0, +1.175 495e-38 to +3.402823e+38	123.456, -3.4, 1.0e-5	MD100, DB1.DBD8, Tag_name
LReal	64	-1.7976931348623158e+308 to -2.2250738585072014e-308, ±0,	12345.123456 789e40, 1.2E+40	DB_name.var_name Quy tắc: - Không hỗ trợ địa chỉ trực tiếp - Có thể đặt các khối OB, FB, FC - Có thể gán các trường

		+2.2250738585072014 e-308 to +1.7976931348623158 e+308		hợp biên toàn cục,
--	--	---	--	--------------------

Ngoài ra S7-1200 còn hỗ trợ thêm các kiểu dữ liệu Time, Date, String và Arrays

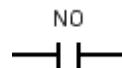
### 3.3 TẬP LỆNH CỦA PLC S7-1200

Mỗi loại PLC sẽ có một tập lệnh khác nhau. PLC S7-1200 là dòng PLC mới nhất hiện nay. Nó hỗ trợ tập lệnh đa dạng và mạnh mẽ. Do đó, để sử dụng hiệu quả và khai thác hết khả năng của PLC S7-1200 cần phải nắm thật rõ các lệnh của nó. Dưới đây là các lệnh của PLC S7-1200 [8] (trang 667 đến trang 1000).

#### 3.3.1 Nhóm lệnh về bit

##### a. Tiếp điểm thường mở:

Ký hiệu:



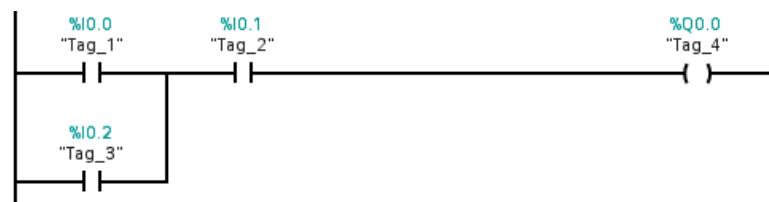
Hình 3.19: Lệnh tiếp điểm thường mở NO

Mô tả:

Tiếp điểm thường mở (Normally Open) viết tắt là NO. Đối với PLC, mỗi tiếp điểm đại diện cho trạng thái một bit trong bộ nhớ dữ liệu hay vùng ảnh của các đầu vào, ra. Tiếp điểm thường mở sẽ được đóng khi bit bằng 1.

Vị trí đặt lệnh: Ở vị trí bất kỳ trong network.

Ví dụ:



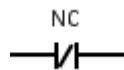
Hình 3.20: Ví dụ về lệnh tiếp điểm thường mở

Ngõ ra Q0.0 on khi có một trong các điều kiện sau:

- I0.0 và I0.1 cùng on.
- I0.2 và I0.1 cùng on.

### b. Tiếp điểm thường đóng

Ký hiệu:



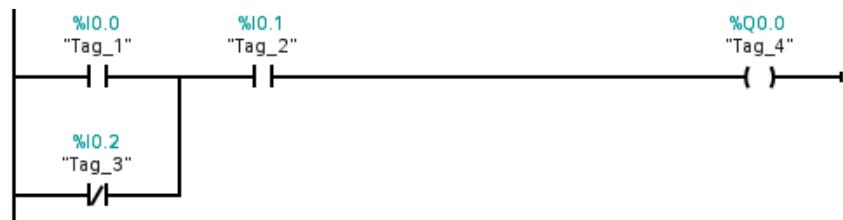
Hình 3.21: Lệnh tiếp điểm thường đóng NC

Mô tả:

Tiếp điểm thường đóng (Normally Closed) viết tắt là NC. Đối với PLC, mỗi tiếp điểm đại diện cho trạng thái một bit trong bộ nhớ dữ liệu hay vùng ảnh của các đầu vào, ra. Tiếp điểm thường đóng sẽ được mở khi bit **đang 0**.

Vị trí đặt lệnh: Ở vị trí bất kỳ trong network

Ví dụ:



Hình 3.22. Ví dụ về lệnh tiếp điểm thường đóng

Ngõ ra Q0.0 on khi có một trong các điều kiện sau:

- I0.0 và I0.1 cùng on.
- I2 off và I0.1 on.

### c. Lệnh đảo bit

Ký hiệu:



Hình 3.23: Lệnh đảo bit

Mô tả:

Lệnh đảo bit sẽ đảo giá trị logic của bit. Nếu phía trước lệnh này có một nguồn điện thì đầu ra của lệnh sẽ không có nguồn. Ngược lại, nếu phía trước lệnh không có nguồn thì đầu ra của lệnh sẽ có nguồn.

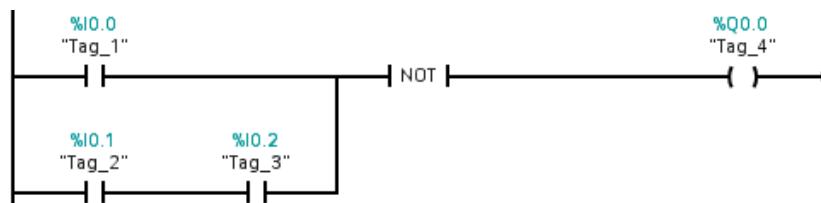
Vị trí đặt lệnh: Ở vị trí bất kỳ trong network

Trong FBD, lệnh đảo không có biểu tượng riêng. Nó được tích hợp như là đầu vào đảo của những khối chức năng khác (với vòng tròn nhỏ ở đầu vào của các khối chức năng đó).



**Hình 3.24: Lệnh đảo bit trong ngôn ngữ FBD**

Ví dụ:



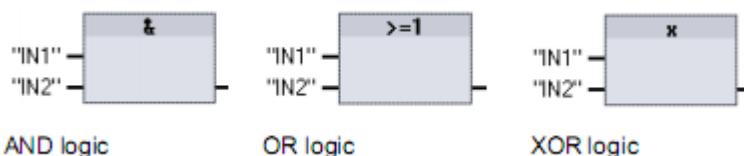
**Hình 3.25. Ví dụ về lệnh tiếp điểm thường đóng**

Ngõ ra Q0.0 off khi có một trong các điều kiện sau:

- I0.0 on.
- I0.1 và I0.2 cùng on.

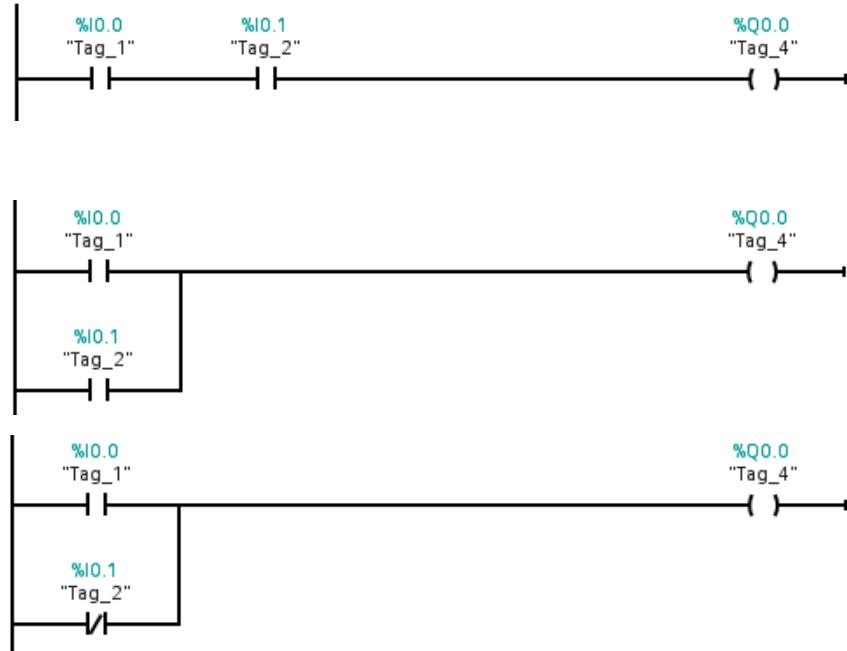
#### d. Lệnh AND, OR, XOR bit

Ký hiệu:



**Hình 3.26: Lệnh AND, OR, XOR bit**

Các lệnh này ở dạng ngôn ngữ FBD. Các lệnh này có thể tạo ra dưới dạng ngôn ngữ LAD bằng cách kết nối các tiếp điểm thường đóng, thường hở lại với nhau.



**Hình 3.27: Lệnh AND, OR, XOR bit trong ngôn ngữ LAD**

Mô tả:

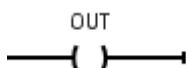
Ngõ ra của lệnh AND chỉ tích cực khi tất cả các ngõ vào được tích cực.

Ngõ ra của lệnh OR sẽ tích cực nếu một trong các ngõ vào được tích cực.

Ngõ ra của lệnh XOR sẽ tích cực nếu hai ngõ vào có trạng thái bit giống nhau.

#### e. Lệnh ngõ ra

Ký hiệu:

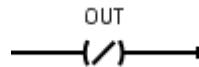


**Hình 3.28: Lệnh ngõ ra ngôn ngữ LAD**

Mô tả:

Mỗi lệnh sẽ được gán một địa chỉ bit đầu ra. Lệnh ngõ ra sẽ ghi giá trị cho bit đầu ra. Khi có nguồn đi qua thì cuộn dây ở trạng thái ON.

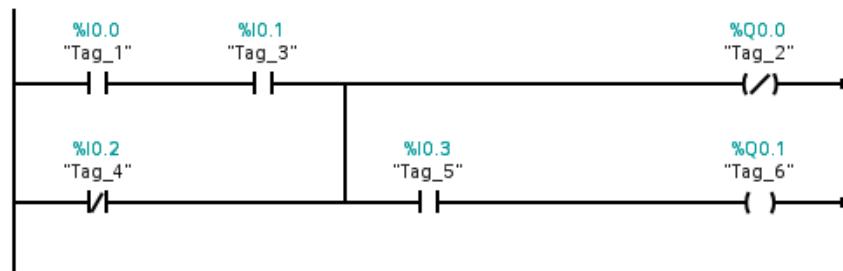
Ngoài ra, còn có lệnh ngõ ra đảo. Khi không có nguồn qua cuộn dây thì ngõ ra ON.



Hình 3.29: Lệnh ngõ ra đảo ngôn ngữ LAD.

Vị trí đặt lệnh: ở vị trí bất kỳ trong network

Ví dụ:



Hình 3.30: Ví dụ về lệnh ngõ ra đảo ngôn ngữ LAD.

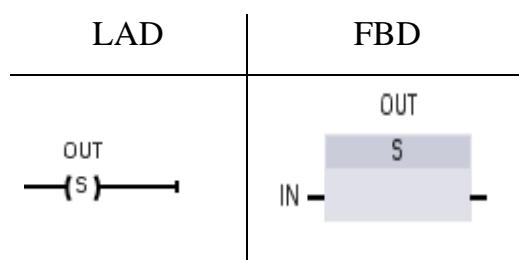
Ngõ ra Q0.0 off khi có một trong các điều kiện sau:

- I0.0 và I0.1 cùng on.
- I0.2 off.
- Ngõ ra Q0.1 on khi có một trong các điều kiện sau:
- I0.0, I0.1 và I0.3 cùng on.
- I0.2 off và I0.3 on.

#### f. Lệnh SET và RESET

##### SET và RESET bit

Ký hiệu:



Hình 3.31: Lệnh SET và RESET bit

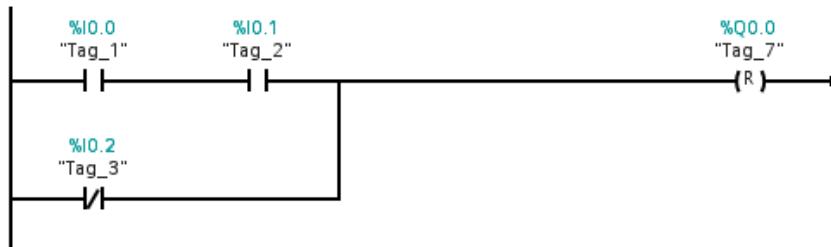
Mô tả:

Khi lệnh SET được tích cực thì ngõ ra sẽ lên mức 1. Trạng thái này sẽ giữ nguyên khi ngõ ra không còn được tích cực.

Khi lệnh RESET được tích cực thì ngõ ra sẽ xuống mức 0. Trạng thái này sẽ giữ nguyên khi ngõ ra không còn được tích cực.

Vị trí đặt lệnh: ở vị trí bất kỳ trong network

Ví dụ:



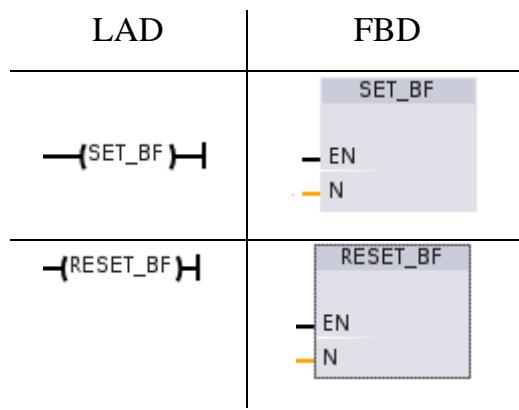
**Hình 3.32: Ví dụ về lệnh SET và RESET bit**

Ngõ ra Q0.0 off khi có một trong các điều kiện sau:

- I0.0 và I0.1 cùng on.
- I0.2 off.

### **SET và RESET một vùng bit**

Ký hiệu:



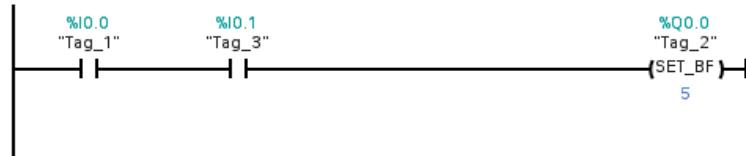
**Hình 3.33: Lệnh SET và RESET một vùng bit.**

Mô tả:

Khi lệnh SET\_BF được tích cực thì ngõ ra sẽ lên mức 1 với số bit đặt trước. Trạng thái này sẽ giữ nguyên khi ngõ ra không còn được tích cực. Ngược lại, khi lệnh RESET\_BF được tích cực thì ngõ ra sẽ xuống mức 0 với số bit đặt trước. Trạng thái này sẽ giữ nguyên khi ngõ ra không còn được tích cực.

Vị trí đặt lệnh: lệnh này chỉ được đặt bên phải cùng của network.

Ví dụ:

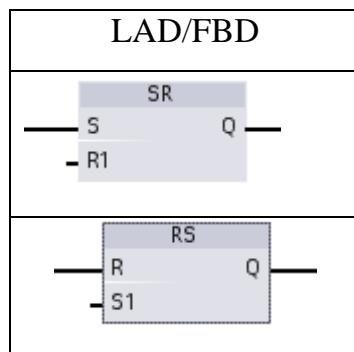


Hình 3.34: Ví dụ về lệnh SET và RESET một vùng bit.

Các bit Q0.0, Q0.1, Q0.2, Q0.3 và Q0.4 đều được đặt lên 1 khi I0.0 và I0.1 on.

### Lệnh ưu tiên SET và RESET

Ký hiệu:



Hình 3.35: Lệnh ưu tiên SET và ưu tiên RESET.

Các tham số của lệnh:

Bảng 3.17: Các tham số của lệnh ưu tiên SET và ưu tiên RESET

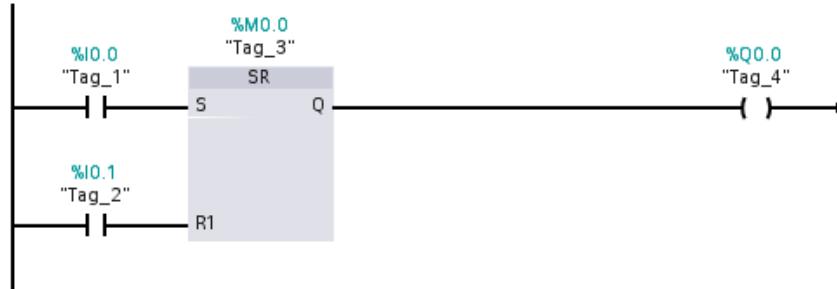
Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
Toán hạng	BOOL	I,Q,M,L,D	Toán hạng cụ thể
S	BOOL	I,Q,M,L,D	Cho phép Set
R1	BOOL	I,Q,M,L,D	Cho phép Reset
Q	BOOL	I,Q,M,L,D	Trạng thái tín hiệu cụ thể

Mô tả:

Lệnh RS là ưu tiên SET. Khi cả SET và RESET đều được gọi thì ngõ ra sẽ là mức 1. Ngược lại, lệnh SR là ưu tiên RESET. Khi cả SET và RESET đều được gọi thì ngõ ra sẽ là mức 0.

Vị trí đặt lệnh: có thể đặt ở bên phải của network.

Ví dụ:



**Hình 3.36: Ví dụ lệnh ưu tiên SET**

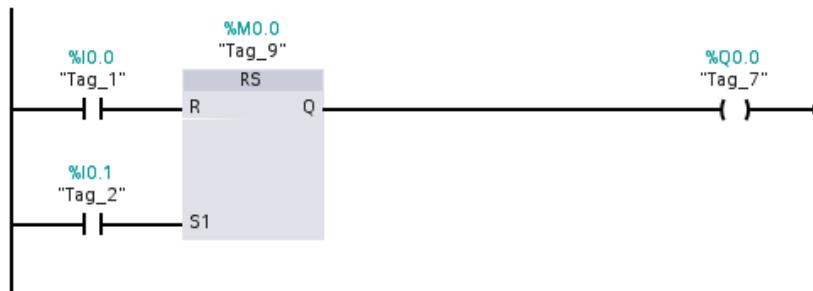
Bit nhớ M0.0 và ngõ ra Q0.0 được đặt lệnh 1 khi có một trong các điều kiện sau :

- I0.0 on.
- I0.1 off.

Bit nhớ M0.0 và ngõ ra Q0.0 xuống 0 khi có một trong các điều kiện sau :

- I0.0 off và I0.1 on.
- Cả I0.0 và I0.1 on.

Ví dụ:



**Hình 3.37: Ví dụ lệnh ưu tiên RESET**

Bit nhớ M0.0 và ngõ ra Q0.0 được đặt lệnh 1 khi có một trong các điều kiện sau:

- I0.0 off và I0.1 on.
- Cả I0.0 và I0.1 on.

Bit nhớ M0.0 và ngõ ra Q0.0 xuống 0 khi có một trong các điều kiện sau :

- I0.0 on.
- I0.1 off.

### g. Lệnh cạnh lên và cạnh xuống

#### Lệnh cạnh lên:

Ký hiệu:



Hình 3.38: Lệnh cạnh lên

Các tham số của lệnh:

Bảng 3.18: Các tham số của lệnh cạnh lên

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
Toán hạng 1	BOOL	I,Q,M,L,D	Tín hiệu được truy vấn
Toán hạng 2	BOOL	I,Q,M,L,D	Bit nhớ lưu trạng thái tín hiệu được truy vấn trước đó .

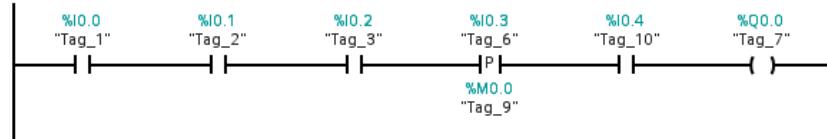
Mô tả:

P: Tiếp điểm sẽ đóng khi phát hiện tín hiệu chuyển từ mức thấp lên mức cao.

Vị trí đặt lệnh:

Nếu sử dụng lệnh này dưới dạng ngôn ngữ LAD thì ta có thể đặt lệnh này ở bất kỳ vị trí nào trong network ngoại trừ cuối nhánh. Nếu sử dụng lệnh này dưới dạng ngôn ngữ FBD thì ta chỉ có thể đặt lệnh này ở đầu của nhánh lệnh.

Ví dụ:



Hình 3.39: Ví dụ về lệnh cạnh lên

Ngõ ra Q0.0 được đặt lên 1 khi có các điều kiện sau:

- I0.0, I0.1 và I0.2 on.
- Có xung cạnh lên tại I0.3

- I0.4 on.

### Lệnh cạnh xuống:

Ký hiệu:



Hình 3.40: Lệnh cạnh xuống

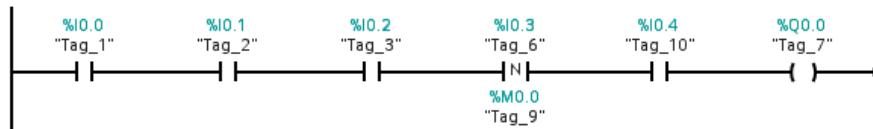
Các tham số của lệnh: (tương tự lệnh cạnh lên)

Mô tả:

N: Tiếp điểm sẽ đóng khi phát hiện tín hiệu chuyển từ mức cao xuống mức thấp.

Vị trí đặt lệnh: (tương tự lệnh cạnh xuống)

Ví dụ:



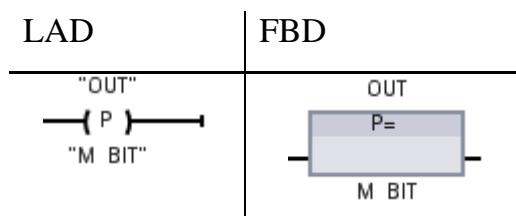
Hình 3.41: Ví dụ về lệnh cạnh xuống

Ngõ ra Q0.0 được đặt lệnh 1 khi có các điều kiện sau:

- I0.0, I0.1 và I0.2 on.
- Có xung cạnh xuống tại I0.3
- I0.4 on.

### Lệnh cạnh lên ngõ ra:

Ký hiệu:



Hình 3.42: Lệnh cạnh lên ngõ ra

Các tham số của lệnh:

**Bảng 3.19: Các tham số của lệnh cạnh lên ngõ ra**

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
Toán hạng 1	BOOL	I,Q,M,L,D	Bit ngõ ra được tác động.
Toán hạng 2	BOOL	I,Q,M,L,D	Bit nhớ cạnh.

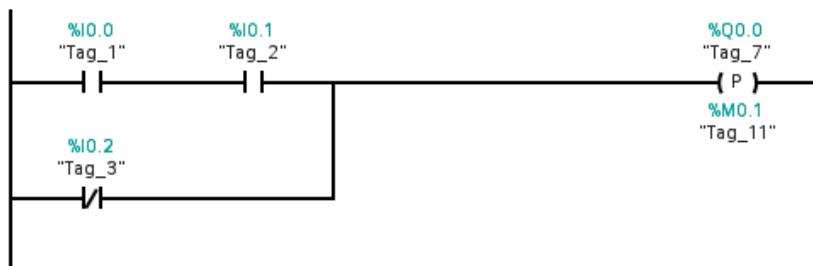
Mô tả:

P=: Ngõ ra sẽ ON khi phát hiện tín hiệu chuyển từ mức thấp lên mức cao.

Vị trí đặt lệnh:

Nếu sử dụng lệnh này dưới dạng ngôn ngữ LAD thì ta có thể đặt lệnh này ở bất kỳ vị trí nào trong network. Nếu sử dụng lệnh này dưới dạng ngôn ngữ FBD thì ta chỉ có thể đặt lệnh này ở vị trí bất kỳ của nhánh lệnh.

Ví dụ:

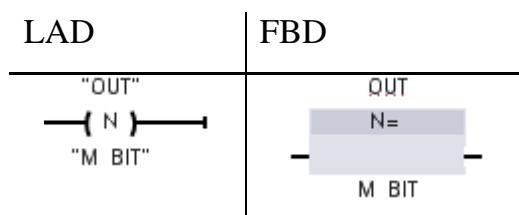


**Hình 3.43: Ví dụ về lệnh cạnh lên ngõ ra**

Ngõ ra Q0.0 được đặt lên 1 khi trạng thái tín hiệu ở đầu vào của cuộn dây chuyển mạch từ "0" lên "1" (cạnh lên). Trong tất cả các trường hợp khác, Q.0 có trạng thái "0".

### Lệnh cạnh xuống ngõ ra:

Ký hiệu:



**Hình 3.44: Lệnh cạnh xuống ngõ ra**

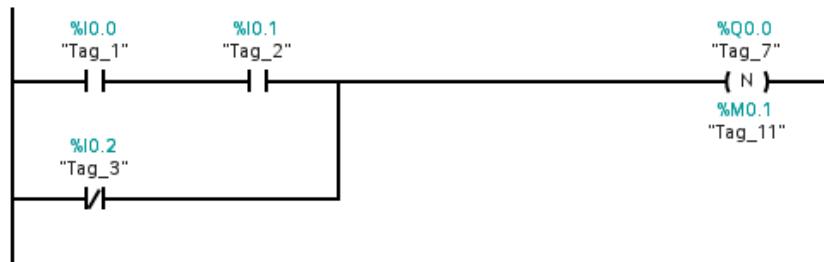
Các tham số của lệnh: (tương tự lệnh cạnh lên ngõ ra)

Mô tả:

N=: Ngõ ra sẽ ON khi phát hiện tín hiệu chuyển từ mức cao xuống mức thấp.

Vị trí đặt lệnh: (tương tự như lệnh ngõ ra cạnh xuống)

Ví dụ:

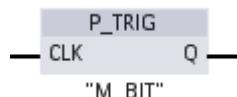


Hình 3.45: Ví dụ về lệnh cạnh xuống ngõ ra

Ngõ ra Q0.0 được đặt lên 1 khi trạng thái tín hiệu ở đầu vào của cuộn dây chuyển mạch từ “1” xuống “0” (cạnh xuống). Trong tất cả các trường hợp khác, Q.0 có trạng thái "0".

### Lệnh cạnh lén xung CLK:

Ký hiệu:



Hình 3.46: Lệnh cạnh lén xung CLK

Các tham số của lệnh:

Bảng 3.20: Các tham số của lệnh cạnh lén xung CLK

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
Toán hạng	BOOL	I,Q,M,L,D	Bit nhớ lưu trạng thái cạnh trước đó
CLK	BOOL		Ngõ vào cạnh
Q	BOOL		Ngõ ra

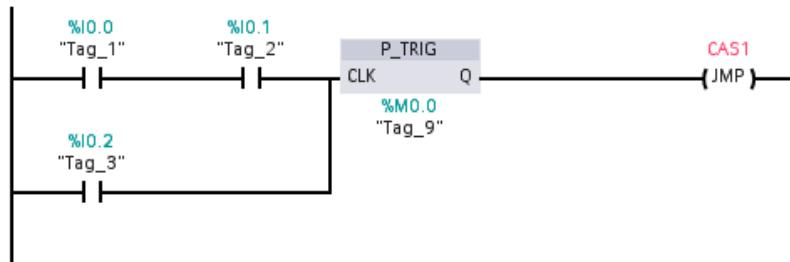
Mô tả:

P\_TRIG: Ngõ ra sẽ ON khi phát hiện tín hiệu chuyển từ mức thấp lên mức cao của xung CLK.

Vị trí đặt lệnh:

Nếu sử dụng lệnh này dưới dạng ngôn ngữ LAD thì ta không được đặt lệnh này ở đầu hoặc cuối network. Nếu sử dụng lệnh này dưới dạng ngôn ngữ FBD thì ta chỉ có thể đặt lệnh ở vị trí bất kỳ của nhánh lệnh ngoại trừ cuối nhánh.

Ví dụ:



**Hình 3.47: Ví dụ về lệnh cạnh lên xung CLK**

Kết quả của các phép toán logic sẽ được lưu trong bit nhớ M0.0. Nếu có thay đổi trạng thái tín hiệu từ "1" xuống "0", chương trình nhảy đến nhãn CAS1.

### Lệnh cạnh xuống xung CLK:

Ký hiệu:



**Hình 3.48: Lệnh cạnh xuống xung CLK.**

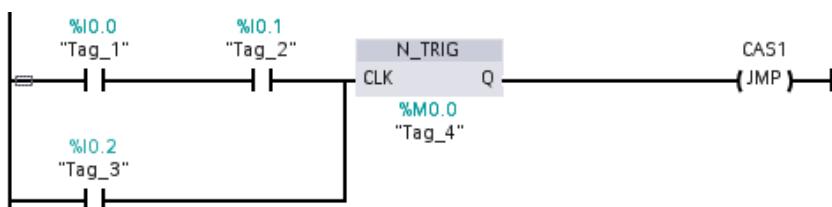
Các tham số của lệnh: (tương tự như lệnh cạnh lên xung CLK)

Mô tả:

N\_TRIG: Ngõ ra sẽ ON khi phát hiện tín hiệu chuyển từ mức cao xuống mức thấp của xung CLK.

Vị trí đặt lệnh: (tương tự như lệnh cạnh lên xung CLK)

Ví dụ:



**Hình 3.49: Ví dụ về lệnh cạnh xuống xung CLK.**

Kết quả của các phép toán logic sẽ được lưu trong bit nhớ M 0.0. Nếu có thay đổi trạng thái tín hiệu từ "0 lên "1", chương trình nhảy để nhảy nhãn là CAS1.

### 3.3.2 Nhóm lệnh định thời

Lệnh định thời được sử dụng trong chương trình để tạo một khoảng thời gian trễ. Thời gian trễ sẽ được quy định cho mỗi bộ định thời. Số bộ định thời sẽ phụ thuộc và mỗi loại CPU mà ta sử dụng. Mỗi timer sử dụng một dữ liệu IEC\_Timer 16 byte kiểu dữ liệu DB để lưu trữ dữ liệu thời gian.

#### c. TP

Bộ định thời TP tạo ra một xung với độ rộng về thời gian được đặt trước.

Ký hiệu:



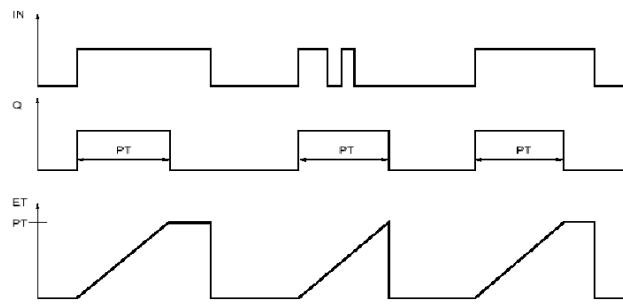
**Hình 3.50: Lệnh định thời TP**

Các thông số của bộ định thời PT:

**Bảng 3.21: Các thông số của bộ định thời PT**

Thông số	Dữ liệu	Vùng nhớ	Mô tả
IN	BOOL	I, Q, M, D, L	Đầu vào cho phép Timer
PT	TIMER	I, Q, M, D, L, hằng số	Giá trị đặt trước cho Timer
Q	BOOL	I, Q, M, D, L	Đầu ra Timer
ET	TIMER	I, Q, M, D, L	Giá trị thời gian trôi qua ở đầu ra
Timer data block	DB	I, Q, M, D, L	Xác định bộ định thời để reset lại khi RT cho phép

Biểu đồ thời gian:



Hình 3.51: Biểu đồ thời gian của lệnh TP

Mô tả:

Khi ngõ vào IN lên 1 thì ngõ ra Q sẽ ON, thời gian Timer bắt đầu tính, đủ thời gian đặt (PT) Q sẽ OFF. Khi IN xuống 0 chưa đủ thời gian đặt mà đã lên 1 thì thời gian đặt (PT) vẫn được tính tiếp.

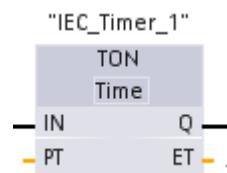
Vị trí đặt lệnh:

Có thể được đặt bên trong hoặc ở cuối của network.

### c. TON

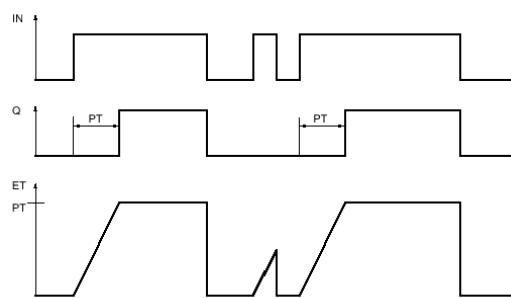
Lệnh định thời TON sẽ trì hoãn đặt ngõ ra ON sau một khoảng thời gian đặt trước.

Ký hiệu:



Hình 3.52: Lệnh định thời TON

Các thông số của bộ định thời TON (tương tự như lệnh TP)



Hình 3.53: Biểu đồ thời gian của Timer On

Mô tả:

Khi ngõ vào IN lên 1 thời gian cho Timer bắt đầu tính, khi ET  $\geq$  PT thì ngõ ra Q ON. Nếu IN lên 1 trong khoảng thời gian chưa đủ thời gian đặt PT mà IN xuống 0 thì ngõ ra Q vẫn giữ nguyên trạng thái (OFF). Khi Q đang ON, ngõ vào IN xuống 0 thì Q sẽ OFF.

Vị trí đặt lệnh:

Có thể được đặt bên trong hoặc ở cuối của network.

### c. TOF

Lệnh định thời TOF sẽ trì hoãn reset ngõ ra về OFF sau một khoảng thời gian đặt trước.

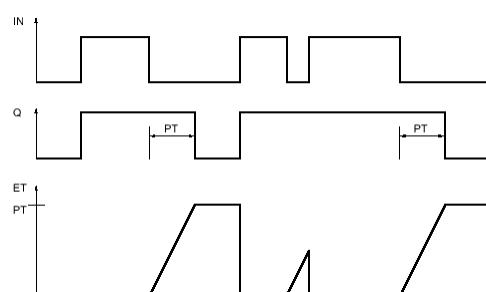
Ký hiệu:



Hình 3.54: Lệnh định thời TOF

Các thông số của bộ định thời TOF (tương tự như lệnh TP)

Biểu đồ thời gian



Hình 3.55: Biểu đồ thời gian của Timer Off

Mô tả:

Khi ngõ vào IN lên 1 thì ngõ ra Q sẽ ON. Khi IN xuống 0, thời gian Timer bắt đầu tính, khi đủ thời gian đặt trước (TP) thì ngõ ra sẽ OFF. Khi IN xuống 0 và lại lên 1 khi chưa đủ thời gian đặt ET<TP thì bit ET vẫn giữ nguyên trạng thái, ngõ ra Q vẫn ON.

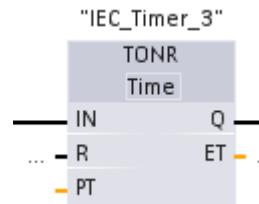
Vị trí đặt lệnh:

Có thể được đặt bên trong hoặc ở cuối của network.

### c. TONR

Bộ định thời TONR sẽ trì hoãn đặt ngõ ra ON sau một khoảng thời gian đặt trước. Và ngõ ra chỉ về OFF khi có tín hiệu reset từ chân R.

Ký hiệu



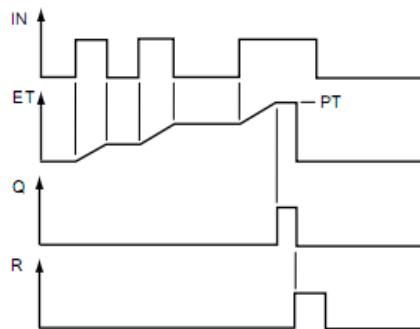
Hình 3.56: Lệnh định thời TONR

Các thông số của bộ định thời TONR

Bảng 3.22: Các thông số của bộ định thời TONR

Thông số	Dữ liệu	Mô tả
IN	BOOL	Đầu vào cho phép Timer
R	BOOL	Thiết lập lại TONR.
PT	TIMER	Giá trị đặt trước cho Timer
Q	BOOL	Đầu ra Timer
ET	TIMER	Giá trị thời gian trôi qua ở đầu ra
0Timer data block	DB	Xác định bộ định thời để reset lại khi RT cho phép

Biểu đồ thời gian:



Hình 3.57: Biểu đồ thời gian của Timer ON có nhớ

Mô tả:

Ngõ vào IN có tác dụng kích thời gian cho Timer, Khi ngõ vào IN = 1 thời gian Timer được tính. Khi IN=0 thời gian đã tính được giữ nguyên. Khi IN = 1 lần kế tiếp thời gian Timer sẽ được tính tiếp. Khi thời gian ET $\geq$ TP thì ngõ ra sẽ ON. Khi có tín hiệu xung reset ở chân R, ngõ ra sẽ OFF và thời gian đặt được tính lại từ 0.

Vị trí đặt lệnh:

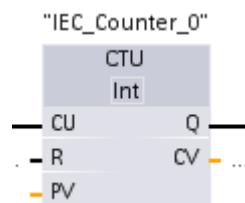
Có thể được đặt bên trong hoặc ở cuối của network.

### 3.3.3 Nhóm lệnh đếm

Các lệnh đếm được sử dụng để đếm các sự kiện bên trong chương trình hoặc các sự kiện của quá trình bên ngoài.

#### a. Lệnh đếm lên

Ký hiệu:

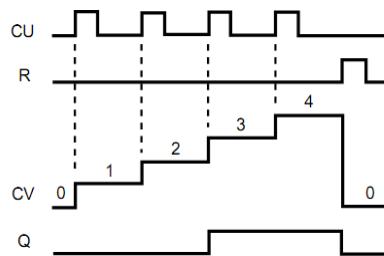


Hình 3.58: Lệnh đếm lên

Các thông số của lệnh đếm lên (Counter Up)

Bảng 3.23: Các thông số của lệnh đếm lên (Counter Up)

Thông số	Dữ liệu	Mô tả
Counter name		Số hiệu Counter
CU	BOOL	Kích đếm lên
R	BOOL	Reset
PV	SINT, INT, DINT, USINT, UINT, UDINT	Giá trị đặt trước cho Counter
Q	BOOL	Đúng nếu CV $\geq$ PV
CV	SINT, INT, DINT, USINT, UINT, UDINT	Giá trị hiện tại



Hình 3.59: Biểu đồ thời gian của lệnh đếm lên (Counter Up)

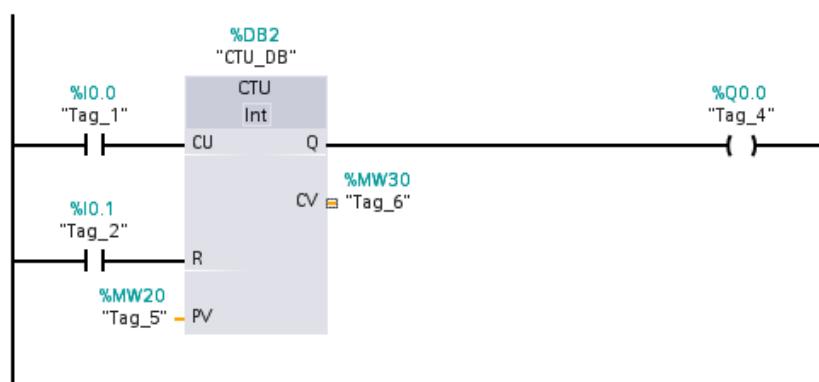
Mô tả:

Khi có một sườn cạnh lên ở CU, giá trị bộ đếm được tăng lên 1. Khi giá trị hiện tại (CV) lớn hơn hoặc bằng giá trị đặt (PV), ngõ ra sẽ được bật lên ON. Khi chân Reset được kích giá trị bộ đếm hiện tại và ngõ ra Q trở về 0. Bộ đếm ngưng đếm khi giá trị bộ đếm đạt giá trị tối đa là 32767.

Vị trí đặt lệnh:

Có thể được đặt bên trong hoặc ở cuối của network.

Ví dụ:

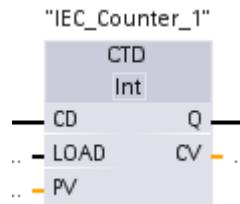


Hình 3.60: Ví dụ về lệnh đếm lên

Khi I0.0 thay đổi từ "0" lên "1", Bộ đếm lên hoạt động và đầu ra MW30 tăng một đơn vị. Với mỗi cạnh lên, giá trị đếm được tăng lên cho đến giá trị giới hạn của các kiểu dữ liệu quy định là 32767. Giá trị tham số MW20 là giới hạn để xác định ngõ ra Q0.0. Đầu ra Q0.0 có trạng thái tín hiệu "1" tức là số hiện nay là lớn hơn hoặc bằng giá trị tham số MW20. Trong tất cả các trường hợp khác, đầu ra Q0.0 có tín hiệu trạng thái "0".

## b. Lệnh đếm xuống

Ký hiệu:



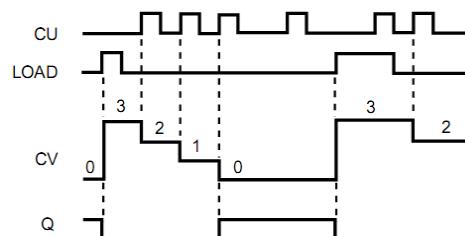
Hình 3.61: Lệnh đếm xuống.

Các thông số của lệnh đếm xuống (Counter Down)

Bảng 3.24: Thông số của lệnh đếm xuống (Counter Down)

Thông số	Dữ liệu	Mô tả
Counter name		Số hiệu Counter
CD	BOOL	Kích đếm xuống
R	BOOL	load
PV	SINT, INT, DINT, USINT, UINT, UDINT	Giá trị đặt trước cho Counter
Q	BOOL	Đúng nếu CV<=0
CV	SINT, INT, DINT, USINT, UINT, UDINT	Giá trị hiện tại

Biểu đồ thời gian



Hình 3.62: Biểu đồ thời gian của lệnh đếm xuống

Mô tả:

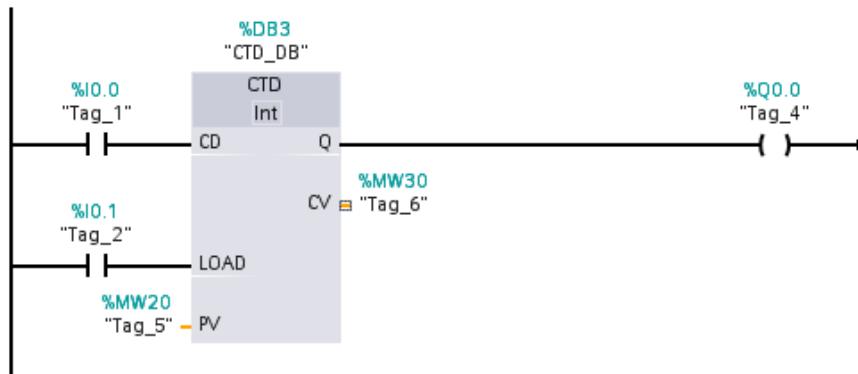
Khi chân Load được kích (sườn lên) thì giá trị PV được nạp cho bộ đếm. Mỗi lần có sườn cạnh lên ở chân CD, giá trị bộ đếm được giảm xuống 1. Khi giá trị hiện tại (CV) của bộ đếm nhỏ hơn hoặc bằng 0, ngõ ra sẽ được bật lên ON.

Bộ đếm ngưng đếm khi giá trị bộ đếm đạt tối thiểu là -32767.

Vị trí đặt lệnh:

Có thể được đặt bên trong hoặc ở cuối của network.

Ví dụ:

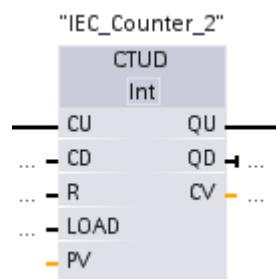


Hình 3.63: Ví dụ về lệnh đếm xuống

Khi I0.0 thay đổi từ "0" lên "1", Bộ đếm xuống hoạt động và đầu ra MW30 giảm một đơn vị. Với mỗi cạnh lên, giá trị đếm được giảm đi cho đến giá trị giới hạn của các kiểu dữ liệu quy định là -32767. Giá trị tham số MW20 là giới hạn để xác định ngoặt ra Q0.0. Đầu ra Q0.0 có trạng thái tín hiệu "0" tức là số hiện nay là nhỏ hơn hoặc bằng giá trị tham số MW20. Trong tất cả các trường hợp khác, đầu ra Q0.0 có tín hiệu trạng thái "0".

### c. Lệnh đếm lên xuống

Ký hiệu:



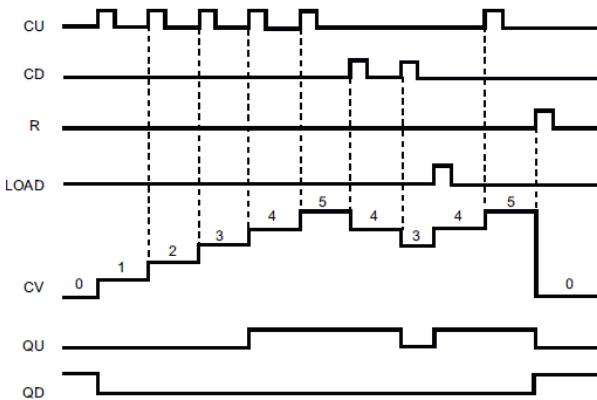
Hình 3.64: Lệnh đếm lên-xuống.

Các thông số của lệnh đếm lên xuống (Counter Up/Down):

Bảng 3.25: Thông số của lệnh đếm lên-xuống

Thông số	Dữ liệu	Mô tả
Counter name		Số hiệu Counter
CU, CD	BOOL	Kích đếm lên/xuống
R	BOOL	load
PV	SINT, INT, DINT, USINT, UINT, UDINT	Giá trị đặt trước cho Counter
QU	BOOL	Đúng nếu CV $\geq 0$
QD	BOOL	Đúng nếu CV $\leq 0$
CV	SINT, INT, DINT, USINT, UINT, UDINT	Giá trị hiện tại

Biểu đồ thời gian:



Hình 3.65: Lệnh đếm lên-xuống.

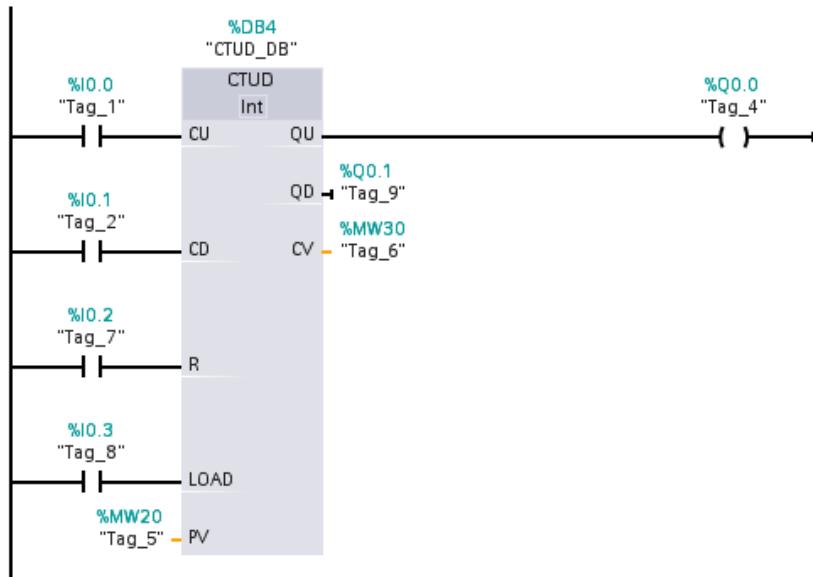
Mô tả:

Mỗi lần có một sườn cạnh lên ở chân CU, giá trị bộ đếm được tăng lên 1. Mỗi lần có một sườn cạnh xuống ở chân CD, giá trị bộ đếm được giảm xuống 1. Khi giá trị hiện tại CV $\geq$ PV, ngõ ra QU sẽ được bật lên ON. Khi giá trị hiện tại CV $\leq 0$ , ngõ ra QD sẽ được bật lên ON. Khi chân R được kích (cạnh lên) giá trị bộ đếm và ngõ ra Q sẽ được reset về 0. Khi chân LOAD được kích (cạnh lên) giá trị PV được đặt lại.

Vị trí đặt lệnh:

Có thể được đặt bên trong hoặc ở cuối của network.

Ví dụ:



Hình 3.66: Ví dụ về lệnh đếm lên-xuống.

Nếu I0.0 hoặc I0.1 thay đổi từ "0" đến "1" (tích cực cạnh), bộ đếm sẽ hoạt động. Khi có một cạnh lên ở đầu vào I0.0, giá trị hiện tại tăng một đơn vị và lưu vào đầu ra MW30. Khi có cạnh lên ở đầu vào I0.1, giá trị hiện tại giảm một đơn vị và lưu vào đầu ra MW30.

Khi có một cạnh lên ở đầu vào CU, giá trị đếm được tăng lên cho đến khi nó đạt đến cao giới hạn là 32 767. Khi có một cạnh lên ở I0.1, giá trị giảm đi cho đến khi nó đạt đến giá trị giới hạn thấp -32 768.

Ngõ ra Q0.0 có trạng thái tín hiệu "1" tức là giá trị hiện tại là lớn hơn hoặc bằng giá trị đầu vào MW20. Trong tất cả các trường hợp khác, Q0.0 có trạng thái "0". Ngõ ra Q0.1 có trạng thái tín hiệu "1" tức là giá trị hiện tại là nhỏ hơn hoặc bằng giá trị đầu vào MW20. Trong tất cả các trường hợp khác, Q0.1 có trạng thái "0".

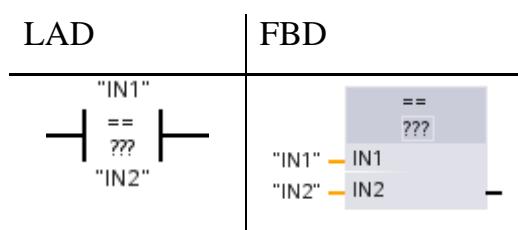
### 3.3.4 Nhóm lệnh so sánh

#### a. Lệnh so sánh Byte

Lệnh so sánh Byte dùng để so sánh hai giá trị dạng Byte được chỉ định bởi hai toán hạng ở đầu vào của lệnh: IN1 và IN2. So sánh kiểu Byte là kiểu so sánh không dấu. Hai giá trị đem so sánh phải có cùng kiểu dữ liệu. Ngõ ra sẽ có kết quả là 1 nếu hai giá trị đem so sánh giống nhau. Ngược lại thì ngõ ra là 0.

Lệnh so sánh bằng kiểu Byte:

Ký hiệu:



Hình 3.67: Lệnh so sánh bằng kiểu Byte

Các thông số của lệnh:

Bảng 3.26: Thông số của lệnh so sánh kiểu Byte

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
Toán hạng 1	USINT,UINT,UDINT,SINT, INT,DINT,REAL,CHAR, STRING,TIME,DTL	I,Q,M,L,D, hằng số	Giá trị so sánh thứ nhất.
Toán hạng 2	USINT,UINT,UDINT,SINT, INT,DINT,REAL,CHAR, STRING,TIME,DTL	I,Q,M,L,D, hằng số	Giá trị so sánh thứ hai.

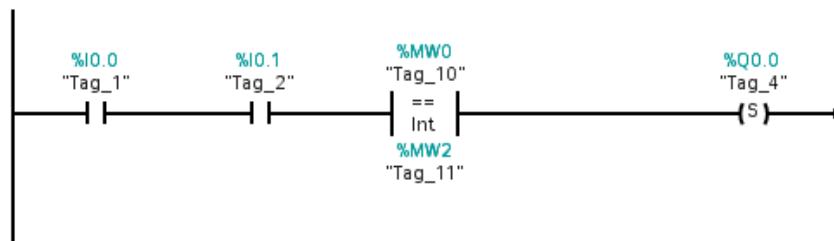
Mô tả:

Ngõ ra 1 khi hai giá trị đem so sánh là bằng nhau.

Vị trí đặt lệnh:

Đặt ở bên trái cạnh hoặc bên trong network.

Ví dụ:



**Hình 3.68: Ví dụ lệnh so sánh bằng\_kiểu Byte**

- Q 0.0 lên 1 khi có một trong các điều kiện sau:
- I 0.0, I0.1 on và giá trị MW0 = MW2.

Một số lệnh so sánh khác:

Để chọn kiểu so sánh khác, click chuột vào dấu “==” để chọn kiểu so sánh.

**Bảng 3.27: Một số lệnh so sánh kiểu Byte khác**

Phép toán	So sánh là đúng nếu:
=	IN1 bằng IN2
<>	IN1 không bằng IN2
>=	IN1 lớn hơn hoặc bằng IN2
<=	IN1 nhỏ hơn hoặc bằng IN2
>	IN1 lớn hơn IN2
<	IN1 nhỏ hơn IN2

## b. Lệnh In-range and Out-of-range

### Lệnh In-range:

Ký hiệu:



**Hình 3.69: Lệnh In-range**

Các thông số của lệnh:

**Bảng 3.28: Thông số của lệnh so sánh kiểu Byte**

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
Ngõ vào	BOOL	I,Q,M,L,D	Kết quả logic trước
Ngõ ra	BOOL	I,Q,M,L,D	Kết quả so sánh
MIN	SINT, INT, DINT, USINT UINT, UDINT, REAL	I,Q,M,L,D, hàng số	Giới hạn dưới của thang giá trị
Giá trị	SINT, INT, DINT, USINT UINT, UDINT, REAL	I,Q,M,L,D, hàng số	Giá trị so sánh
MAX	SINT, INT, DINT, USINT UINT, UDINT, REAL	I,Q,M,L,D, hàng số	Giới hạn trên của thang giá trị

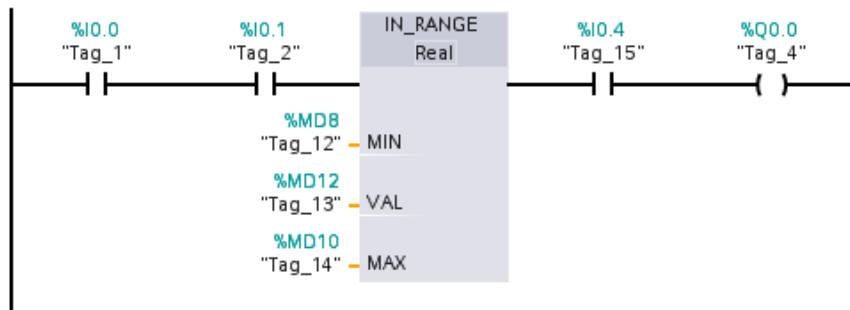
Mô tả:

Lệnh IN\_RANGE dùng để kiểm tra một giá trị có nằm trong phạm vi giá trị chỉ định hay không. Nếu MIN  $\leq$  VAL  $\leq$  MAX thì ngõ ra sẽ lên 1. Ngược lại thì ngõ ra ở mức 0.

Vị trí đặt lệnh:

Đặt ở bên trái cạnh hoặc bên trong network.

Ví dụ:



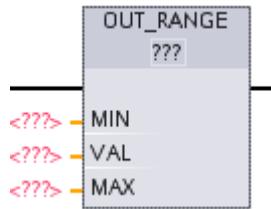
**Hình 3.70: Ví dụ về lệnh In-range**

Q0.0 on khi có tất cả các điều kiện sau:

- I0.0 và I0.1 là "1".
- Giá trị ở đầu vào MD12 trong phạm vi giá trị được quy định bởi các giá trị hiện tại ở MD8 và MD10 (MIN  $\leq$  VAL  $\leq$  MAX).
- Đầu vào I0.4 có trạng thái tín hiệu "1".

## Lệnh Out-of-range:

Ký hiệu:



Hình 3.71: Lệnh Out-of-range

Các thông số của lệnh (tương tự lệnh In-Range)

Có thể chọn kiểu dữ liệu cần so sánh bằng cách click vào “???” và chọn kiểu dữ liệu từ danh sách.

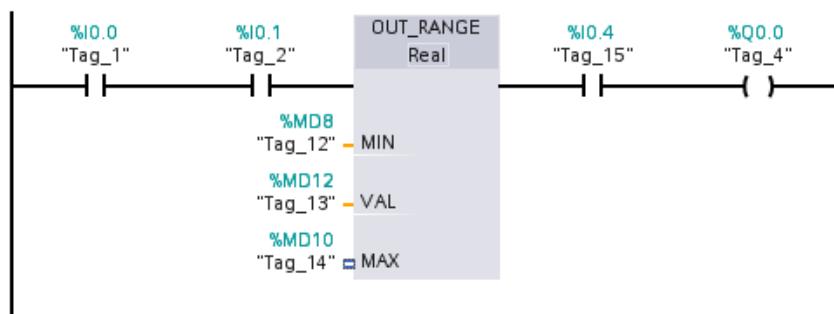
Mô tả:

Lệnh OUT\_RANGE dùng để kiểm tra một giá trị có nằm ngoài phạm vi giá trị chỉ định hay không. Nếu VAL < MIN hoặc VAR > MAX thì ngõ ra sẽ lên 1. Ngược lại thì ngõ ra ở mức 0.

Vị trí đặt lệnh:

Đặt ở bên trái cạnh hoặc bên trong network.

Ví dụ:



Hình 3.72: Ví dụ về lệnh Out-of-range

Q0.0 on khi có tất cả các điều kiện sau:

- I0.0 và I0.1 là "1".
- Giá trị ở đầu vào MD12 ngoài phạm vi giá trị được quy định bởi các giá trị hiện tại ở MD8 và MD10 (VAL < MIN hoặc VAR > MAX).
- Đầu vào I0.4 có trạng thái tín hiệu "1".

### c. Lệnh OK and Not OK

Ký hiệu:



Hình 3.73: Lệnh OK và NOT\_OK

Mô tả:

Kiểm tra xem có một dữ liệu tham khảo là một số thực hợp lệ theo IEEE đặc điểm kỹ thuật 754 hay không.

Một giá trị Real hoặc LReal là không hợp lệ nếu nó là + / - INF (vô cực), NaN (không phải là một số), hoặc nếu nó là một giá trị không bình thường. Một giá trị không bình thường là một số lượng xấp xỉ bằng không. CPU thay thế một số không cho một giá trị không bình thường trong tính toán.

Lệnh OK thì khi dữ liệu được kiểm là đúng thì ngõ ra tích cực. Ngược lại thì ngõ ra không tích cực.

Lệnh NOT\_OK thì khi dữ liệu được kiểm là sai thì ngõ ra tích cực. Ngược lại thì ngõ ra không tích cực.

Kiểu dữ liệu so sánh:

Bảng 3.29: Kiểu dữ liệu so sánh của lệnh OK và NOT\_OK.

Thông số	Dữ liệu	Mô tả
IN	Real, LReal	Dữ liệu ngõ vào

### 3.3.5 Nhóm lệnh toán học

#### a. Lệnh cộng, trừ, nhân, chia

##### Lệnh cộng:

Ký hiệu:



Hình 3.74: Lệnh cộng

Các thông số của lệnh:

Bảng 3.30: Thông số của lệnh cộng

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN1	SInt,Int,Dint,USInt, UInt,UDint Real	I,Q,M,L,D, hàng số	Giá trị số thứ nhất
IN2	SInt,Int,Dint,USInt, UInt,UDint Real	I,Q,M,L,D, hàng số	Giá trị số thứ hai
OUT	SInt,Int,Dint,USInt, UInt,UDint Real	I,Q,M,L,D	Kết quả của phép tính

Click vào “???” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Để thêm một đầu vào cho lệnh, bấm vào biểu tượng tạo "Create"



hoặc click chuột phải trên một đầu vào khác và chọn lệnh chèn đầu vào "Insert Input".

Để xóa một đầu vào, click phải chuột vào một đầu vào và chọn lệnh "Delete".

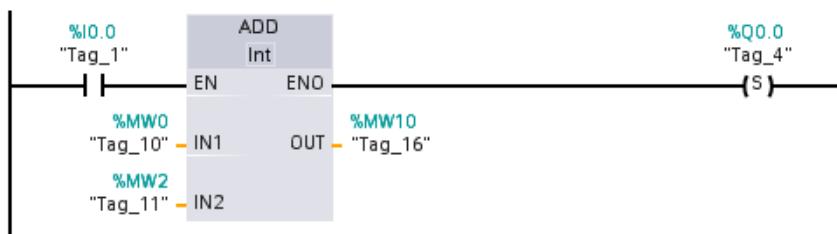
Mô tả:

Khi chân EN được kích hoạt (EN = 1), lệnh sẽ thực hiện phép toán cộng các giá trị (IN1 và IN2) và lưu trữ các kết quả trong các địa chỉ bộ nhớ theo quy định của đầu ra (OUT). Sau khi hoàn thành phép tính, chân ENO sẽ lên 1.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

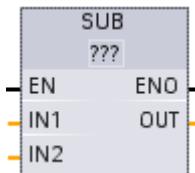


Hình 3.75: Ví dụ về lệnh cộng

Nếu I0.0 = 1, thì lệnh sẽ cộng giá trị tại MW0 và giá trị MW2. Kết quả được lưu tại MW10. Nếu không có lỗi xảy ra trong quá trình thực hiện, ENO = "1" và Q0.0 on.

### Lệnh trừ:

Ký hiệu:



Hình 3.76: Lệnh trừ

Các thông số của lệnh (tương tự tham số của lệnh cộng)

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Khi chân EN được kích hoạt (EN = 1), lệnh sẽ thực hiện phép toán trừ giá trị IN1 cho IN2 và lưu trữ các kết quả trong các địa chỉ bộ nhớ

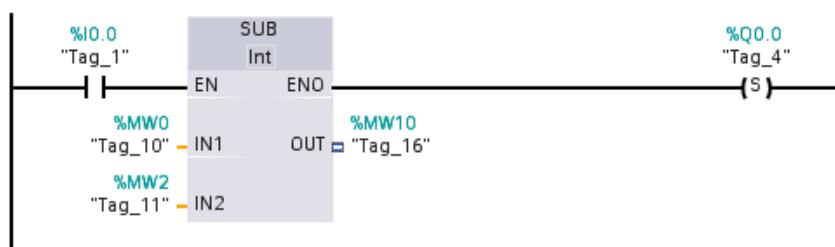
theo quy định của đầu ra (OUT). Sau khi hoàn thành phép tính, chân ENO sẽ lên 1.

Một vài trường hợp có thể xảy ra trong phép tính thông qua trạng thái của ENO:

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

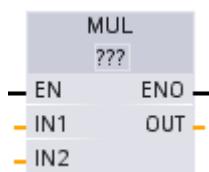


Hình 3.77: Ví dụ về lệnh trừ

Nếu I0.0 = 1, thì lệnh sẽ trừ giá trị tại MW0 cho giá trị MW2. Kết quả được lưu tại MW10. Nếu không có lỗi xảy ra trong quá trình thực hiện, ENO = "1" và Q0.0 on.

### Lệnh nhân:

Ký hiệu:



Hình 3.78: Lệnh nhân

Các thông số của lệnh (tương tự tham số của lệnh cộng)

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

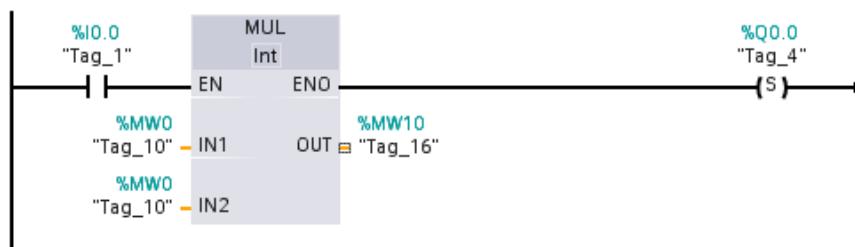
Khi chân EN được kích hoạt (EN = 1), lệnh sẽ thực hiện phép toán nhân giá trị IN1 cho IN2 và lưu trữ các kết quả trong các địa chỉ bộ

nhớ theo quy định của đầu ra (OUT). Sau khi hoàn thành phép tính, chân ENO sẽ lên 1.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

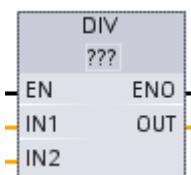


**Hình 3.79: Ví dụ về lệnh nhân**

Nếu I0.0 = 1, thì lệnh sẽ nhân giá trị tại MW0 cho giá trị MW2. Kết quả được lưu tại MW10. Nếu không có lỗi xảy ra trong quá trình thực hiện, ENO = "1" và Q0.0 on.

### Lệnh chia:

Ký hiệu:



**Hình 3.80: Lệnh chia**

Các thông số của lệnh (tương tự tham số của lệnh cộng)

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Khi chân EN được kích hoạt (EN = 1), lệnh sẽ thực hiện phép toán chia giá trị IN1 cho IN2 và lưu trữ các kết quả trong các địa chỉ bộ nhớ theo quy định của đầu ra (OUT). Sau khi hoàn thành phép tính, chân ENO sẽ lên 1.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



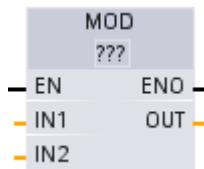
Hình 3.81: Ví dụ về lệnh chia

Nếu I0.0 = 1, thì lệnh sẽ chia giá trị tại MW0 cho giá trị MW2.

Kết quả được lưu tại MW10. Nếu không có lỗi xảy ra trong quá trình thực hiện, ENO = "1" và Q0.0 on.

### b. Lệnh chia có dư

Ký hiệu:



Hình 3.82: Lệnh chia có dư

Các thông số của lệnh (tương tự tham số của lệnh cộng)

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Lệnh MOD được sử dụng để trả lại phần dư trong phép chia số nguyên.

Giá trị tại đầu vào IN1 được chia cho giá trị ở đầu vào IN2 và phần dư được đưa ra OUT.

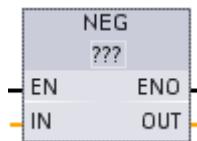
Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

### c. Lệnh phủ định

Ký hiệu:



**Hình 3.83: Lệnh phủ định**

Các thông số của lệnh:

**Bảng 3.31: Thông số của lệnh phủ định**

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	SInt,Int,Dint,Real	I,Q,M,L,D, hằng số	Giá trị ngõ vào
OUT	SInt,Int,Dint,Real	I,Q,M,L,D	Kết quả của phép tính

Click vào “???” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Lệnh NEG đảo dấu của giá trị tham số IN và lưu trữ kết quả trong tham số OUT.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Một vài trường hợp có thể xảy ra trong phép tính thông qua trạng thái của ENO:

**Bảng 3.32. Một vài trường hợp có thể xảy ra trong phép tính**

ENO	Mô tả
1	Không có lỗi
0	Giá trị của kết quả nằm ngoài khoảng giá trị của dữ liệu

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

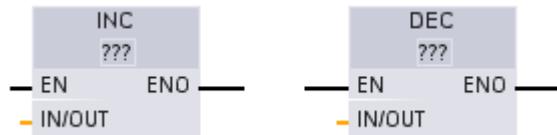


Hình 3.84: Ví dụ về lệnh phủ định

Nếu I0.0 = "1", giá trị đầu vào MD8 sẽ bị đảo dấu và kết quả được lưu ở đầu ra MD12. Nếu không có lỗi xảy ra trong quá trình thực hiện ENO = "1" và Q0.0 on.

#### d. Lệnh tăng và giảm

Ký hiệu:



Hình 3.85: Lệnh tăng và giảm

Các thông số của lệnh:

Bảng 3.33: Thông số của lệnh

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN/OUT	SInt,Int,Dint, USInt,UInt,UDint	I,Q,M,L,D	Giá trị cần tác động

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Lệnh tăng dùng để tăng một số nguyên có hoặc không có dấu.

$$\text{IN\_OUT value} + 1 = \text{IN\_OUT value}$$

Lệnh giảm dùng để giảm một số nguyên có hoặc không có dấu.

$$\text{IN\_OUT value} - 1 = \text{IN\_OUT value}$$

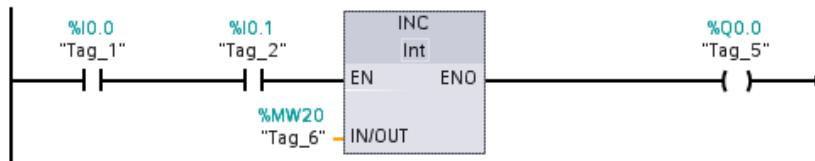
Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Một vài trường hợp có thể xảy ra trong phép tính (tương tự lệnh đảo).

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

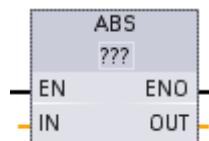


Hình 3.86: Ví dụ về lệnh tăng và giảm

Nếu I0.0 và I0.1 on thì giá trị MW20 tăng một đơn vị và Q0.0 on.

#### e. Lệnh trị tuyệt đối

Ký hiệu:



Hình 3.87: Lệnh trị tuyệt đối

Các thông số của lệnh

Bảng 3.34: Thông số của lệnh trị tuyệt đối

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	SInt,Int,Dint,R EAL	I,Q,M,L,D, hằng số	Giá trị vào
OUT	SInt,Int,Dint,R EAL	I,Q,M,L,D	Giá trị được lấy trị tuyệt đối

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Lệnh trị tuyệt đối sẽ lấy trị tuyệt đối của một số nguyên có dấu hoặc một số thực ở tham số IN và lưu trữ kết quả trong tham số OUT.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Một vài trường hợp có thể xảy ra trong phép tính (tương tự lệnh đảo).

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

#### f. Lệnh giá trị lớn nhất và nhỏ nhất

Ký hiệu:



Hình 3.88: Lệnh giá trị lớn nhất và giá trị nhỏ nhất.

Các thông số của lệnh

Bảng 3.35: Thông số của lệnh giá trị lớn nhất và nhỏ nhất

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN1	SInt,Int,Dint,USInt, UIInt,UDint Real	I,Q,M,L,D, hàng số	Giá trị số thứ nhất
IN2	SInt,Int,Dint,USInt, UIInt,UDint Real	I,Q,M,L,D, hàng số	Giá trị số thứ hai
OUT	SInt,Int,Dint,USInt, UIInt,UDint Real	I,Q,M,L,D	Kết quả của phép tính

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Lệnh giá trị nhỏ nhất dùng để so sánh giá trị của hai tham số vào IN1 và IN2 và lấy giá trị nhỏ nhất đưa vào tham số OUT.

Lệnh giá trị lớn nhất dùng để so sánh giá trị của hai tham số vào IN1 và IN2 và lấy giá trị lớn nhất đưa vào tham số OUT.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Một vài trường hợp có thể xảy ra trong phép tính thông qua trạng thái của ENO:

Bảng 3.36: Một vài trường hợp có thể xảy ra trong phép tính

ENO	Mô tả
1	Không có lỗi
0	Một hoặc nhiều ngõ vào không phải kiểu số thực. Kết quả không phải là số nguyên.

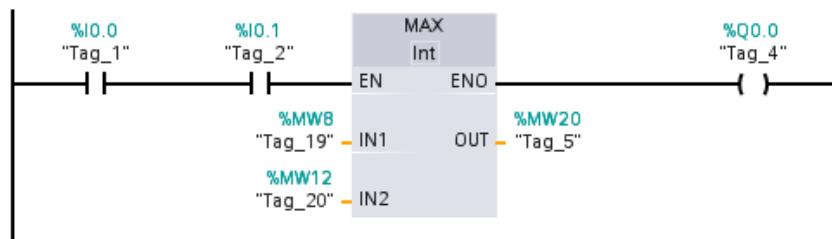
Để thêm vào một ngõ vào, click vào biểu tượng tạo chuột phải vào tham số IN và chọn lệnh chèn ngõ vào “Insert input”

Để xóa một ngõ vào, click chuột phải vào ngõ vào IN và chọn lệnh xóa “Delete”.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



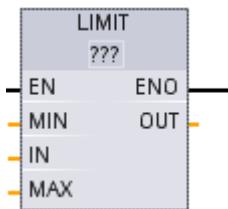
Hình 3.89: Ví dụ về lệnh Max

Trong đó:

- MW8 = 12666
- MW12 = 144444
- Nếu I0.0 và I0.1 on, lệnh lựa chọn giá trị lớn hơn (MW12). Giá trị này được sao chép đến đầu ra MW20. Nếu không có lỗi xảy ra, Q0.0 on.

### g. Lệnh giới hạn

Ký hiệu:



Hình 3.90: Lệnh giới hạn

Các thông số của lệnh

Bảng 3.37: Thông số của lệnh giới hạn

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
MIN	SInt,Int,Dint,USInt, UInt,UDint Real	I,Q,M,L,D, hằng số	Giới hạn nhỏ nhất
IN	SInt,Int,Dint,USInt, UInt,UDint Real	I,Q,M,L,D, hằng số	Giá trị vào
MAX	SInt,Int,Dint,USInt, UInt,UDint Real	I,Q,M,L,D, hằng số	Giới hạn lớn nhất
OUT	SInt,Int,Dint,USInt, UInt,UDint Real	I,Q,M,L,D	Kết quả của phép tính

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Tham số đầu vào IN1 và IN2 phải có cùng kiểu dữ liệu.

Mô tả:

Lệnh giới hạn kiểm tra giá trị của tham số vào IN có nằm trong khoảng giá trị MIN và MAX không.

Nếu giá trị của tham số IN nằm trong phạm vi quy định MIN và MAX, thì giá trị của IN được lưu trữ trong tham số OUT.

Nếu giá trị của tham số IN nằm ngoài phạm vi quy định MIN và MAX, thì giá trị đưa ra OUT là giá trị của tham số MIN (nếu giá trị

IN nhỏ hơn giá trị MIN) hoặc giá trị của tham số MAX (nếu giá trị IN lớn hơn giá trị MAX).

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Vị trí đặt lệnh:

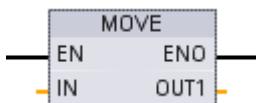
Lệnh có thể đặt ở vị trí bất kỳ trong network.

### 3.3.6 Nhóm lệnh di chuyển

Lệnh di chuyển được sử dụng để sao chép dữ liệu đến một địa chỉ bộ nhớ mới và chuyển đổi từ một kiểu dữ liệu này sang kiểu khác. Các nguồn dữ liệu là không thay đổi bởi quá trình di chuyển.

#### a. Lệnh Move

Ký hiệu:



Hình 3.91: Lệnh di chuyển MOVE

Các thông số của lệnh di chuyển:

Bảng 3.38: Thông số của lệnh

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	Các kiểu dữ liệu cơ bản và STRUCT, DTL,ARRAY	I,Q,M,L,D, hàng số	Giá trị nguồn
OUT	Các kiểu dữ liệu cơ bản và STRUCT, DTL,ARRAY	I,Q,M,L,D	Địa chỉ đích

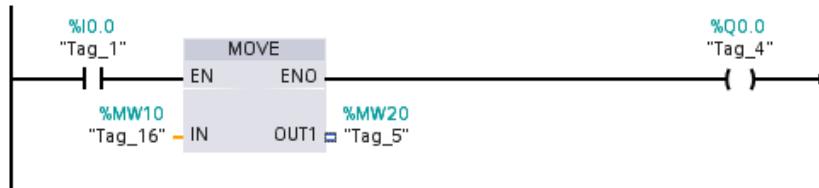
Mô tả:

Lệnh MOVE sao chép một phần tử dữ liệu từ địa chỉ nguồn chỉ định bởi tham số IN đến địa chỉ mới được chỉ định bởi tham số OUT.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



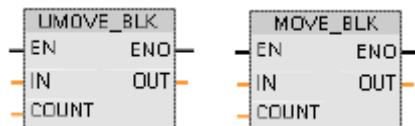
**Hình 3.92: Ví dụ về lệnh di chuyển MOVE**

Trong đó:

- MW10 = 001111110101111
- Nếu I0.0 = "1", lệnh sẽ thực hiện. Nội dung của toán hạng đầu vào (MW10) được sao chép qua MW20 và Q0.0 on.

### b. Lệnh Move Block

Ký hiệu:



**Hình 3.93: Lệnh move\_block**

Các thông số của lệnh:

**Bảng 3.39: Thông số của lệnh Move\_block**

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	ARRAY	L,D	Phần tử thứ nhất của nguồn được sao chép
COUNT	UINT	I,Q,M,L,D, hàng số	Số phần tử được sao chép
OUT	ARRAY	L,D	Phần tử thứ nhất của vùng đích được chép

Mô tả:

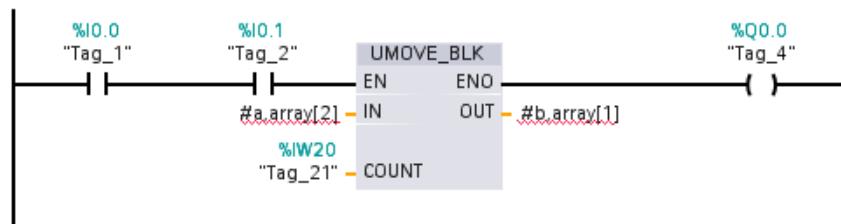
Lệnh MOVE\_BLOCK và UMOVE\_BLOCK sao chép một khối dữ liệu đến một địa chỉ mới. Hai lệnh này có thêm tham số COUNT. Tham số COUNT cho biết có bao nhiêu phần tử được sao chép. Số lượng các byte mỗi phần tử sao chép phụ thuộc vào loại dữ liệu được đặt.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



Hình 3.94: Lệnh umove\_blk

Với IW20 = 3

Nếu I0.0 và I0.1 on, lệnh sẽ thực thi. Nội dung 3 bit [2 .. 4] trong mảng "a\_array" được sao chép qua vị trí 3 bit [1 .. 3] trong mảng "b\_array". Khi lệnh thực thi sẽ không bị ngắt bởi các lệnh khác. Nếu không có lỗi xảy ra trong quá trình thực hiện, ENO sẽ = "1" và Q0.0 on.

Để thêm ngõ ra cho lệnh MOVE, click vào biểu tượng tạo ngõ ra

“Create”  hoặc click chuột phải vào ngõ ra OUT chọn lệnh chèn “Insert output”.

Để xóa một ngõ ra, click chuột phải vào ngõ ra cần xóa và chọn lệnh “Delete”.

Quy tắc cho các hoạt động sao chép dữ liệu

Để sao chép kiểu dữ liệu Bool, sử dụng SET\_BF, RESET\_BF, R, S, hoặc đầu ra (LAD).

Để sao chép một kiểu dữ liệu cơ bản, sử dụng MOVE

Để sao chép một mảng của một kiểu dữ liệu cơ bản, sử dụng MOVE\_BLK hoặc UMOVE\_BLK.

Để sao chép một cấu trúc, sử dụng MOVE.

Để sao chép một chuỗi, sử dụng S\_MOVE (giới thiệu trong phần lệnh về chuỗi).

Để sao chép một ký tự duy nhất trong một chuỗi, sử dụng MOVE.

Lệnh MOVE\_BLK và UMOVE\_BLK có thể được sử dụng để sao chép các mảng hoặc cấu trúc đến các vùng nhớ I, Q, M.

Lệnh MOVE\_BLK và UMOVE\_BLK khác nhau trong xử lý ngắt như thế nào:

sự kiện ngắt được xếp hàng đợi và xử lý trong quá trình thực hiện MOVE\_BLK. Sử dụng lệnh MOVE\_BLK khi dữ liệu ở địa chỉ đích di chuyển không được sử dụng trong chương trình con ngắt ngoài OB, hoặc nếu được sử dụng, dữ liệu đích không được nhất quán. Nếu một toán tử MOVE\_BLK bị ngắt, sau đó phần tử dữ liệu cuối cùng di chuyển đầy đủ và phù hợp tại địa chỉ đích.

Toán tử MOVE\_BLK trở lại sau khi thực hiện ngắt OB hoàn tất.

Sự kiện ngắt được xếp hàng đợi nhưng không được xử lý cho đến khi MOVE\_BLK thực hiện là hoàn tất. Sử dụng lệnh UMOVE\_BLK khi toán tử di chuyển phải được hoàn thành và điểm đến dữ liệu phù hợp, trước khi thực hiện một chương trình con ngắt OB.

Trạng thái của ENO sau của lệnh MOVE.

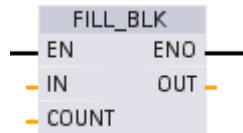
**Bảng 3.40: Trạng thái của ENO sau của lệnh MOVE.**

ENO	Điều kiện	Kết quả
1	Không có lỗi	Tất cả COUN sao chép thành công
0	Giá trị nguồn hoặc đích vượt quá giới hạn của vùng nhớ có sẵn.	Không có phần nào của phần tử được sao chép

### c. Lệnh Fill

#### Lệnh FILL\_BLK

Ký hiệu:



Hình 3.95: lệnh FILL\_BLK

Các thông số của lệnh:

Bảng 3.41: Thông số của lệnh Fill

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	ARRAY	L,D, hằng số	Phần tử dùng để điền vào vùng đích
COUNT	UINT	I,Q,M,L,D, hằng số	Số phần tử sao chép lặp lại
OUT	ARRAY	L,D	Địa chỉ vùng đích bắt đầu điền vào.

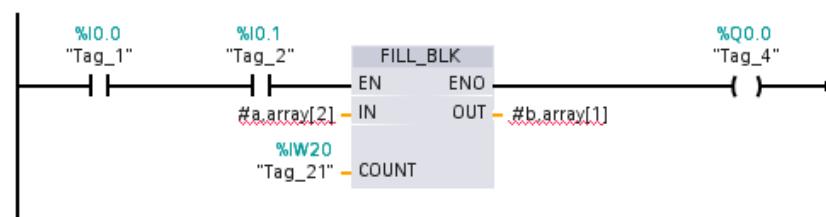
Mô tả:

FILL\_BLK: lệnh điền ngắn sẽ điền vào một vùng địa chỉ bắn sao của một phần tử dữ liệu chỉ định.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



Hình 3.96: Ví dụ về lệnh FILL\_BLK

Với IW20 = 3

Nếu ngõ vào I0.0 và I0.1 on, lệnh sẽ thực thi. Khi đó bit thứ hai [2] của mảng "a\_array" sẽ được sao chép đến 3 bit [1 .. 3] ở mảng "b\_array". Nếu không có lỗi xảy ra trong quá trình thực hiện ENO =1 và Q0.0 on.

### **Lệnh UFILL\_BLK**

Ký hiệu:



**Hình 3.97: Lệnh UFILL\_BLK**

Các thông số của lệnh (tương tự như lệnh FILL\_BLK)

Mô tả:

UFILL\_BLK: lệnh điền không ngắt sẽ điền vào một vùng địa chỉ bản sao của một phần tử dữ liệu chỉ định.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Lưu ý:

Quy tắc cho phép toán điền dữ liệu

Để điền vào với loại dữ liệu BOOL, sử dụng SET\_BF, RESET\_BF, R, S, hoặc cuộn dây đầu ra (LAD).

Để điền vào tất cả các bit, sử dụng MOVE

Để điền vào một mảng với một kiểu dữ liệu cơ bản, sử dụng FILL\_BLK hoặc UFILL\_BLK.

Để điền vào một ký tự duy nhất trong một chuỗi, sử dụng MOVE

Lệnh FILL\_BLK và UFILL\_BLK không được sử dụng để điền vào các mảng trong I, Q, hoặc M các vùng nhớ.

Lệnh FILL\_BLK và UFILL\_BLK sao chép các phần tử dữ liệu nguồn IN đến địa chỉ ban đầu được xác định bởi OUT. Quá trình sao chép lặp đi lặp lại và ngăn chặn các địa chỉ liền kề là đầy cho đến khi

số lượng bản sao bằng với tham số COUNT. FILL\_BLK và UFILL\_BLK khác nhau trong các xử lý ngắt như thế nào.

Sự kiện ngắt được xếp hàng đợi và xử lý trong quá trình thực hiện FILL\_BLK. Sử dụng FILL\_BLK khi các dữ liệu ở địa chỉ đích di chuyển không được sử dụng trong một chương trình con ngắt ngoài OB, nếu được sử dụng, dữ liệu điểm đến không được trùng.

Sự kiện ngắt được xếp hàng đợi nhưng không được xử lý cho đến khi FILL\_BLK thực hiện là hoàn tất.

Sử dụng lệnh UFILL\_BLK khi hoạt động di chuyển phải được hoàn thành và điểm đến dữ liệu phù hợp, trước khi thực hiện một chương trình con ngắt OB.

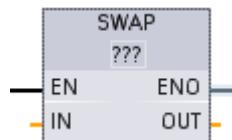
Trạng thái của ENO sau của lệnh

**Bảng 3.42: Trạng thái của ENO sau của lệnh**

ENO	Điều kiện	Kết quả
1	Không có lỗi	Tất cả COUNT sao chép thành công
0	Giá trị đích vượt quá giới hạn của vùng nhớ có sẵn.	Không có phần nào của phần tử được sao chép

#### d. Lệnh hoán đổi Byte Swap

Ký hiệu:



**Hình 3.98: Lệnh hoán đổi Swap**

Các thông số của lệnh:

**Bảng 3.43 Thông số của lệnh hoán đổi**

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào

ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	Word,DWork	I,Q,M,L,D, hằng số	Toán tử byte được hoán đổi
OUT	Word,DWork	I,Q,M,L,D	Kết quả

Mô tả:

Lệnh Swap được sử dụng để hoán đổi dữ liệu hai hay bốn byte.  
Không có thay đổi thứ tự bit trong mỗi byte.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



**Hình 3.99: Ví dụ về lệnh hoán đổi Swap**

Với:

$$IW10 = 0000\ 1111\ 0101\ 0101$$

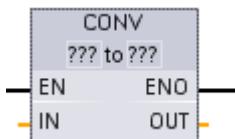
$$QW20 = 0101\ 0101\ 0000\ 1111$$

Nếu I0.1 = "1", lệnh sẽ thực thi. Các byte được thay đổi và được lưu trữ trong QW20. Nếu không có lỗi xảy ra trong quá trình thực hiện ENO = "1" và Q0.0 on.

### 3.3.7 Nhóm lệnh chuyển đổi

#### a. Lệnh CONV

Ký hiệu:



**Hình 3.100: Lệnh CONV**

## Các thông số của lệnh

**Bảng 3.44: Thông số của lệnh**

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	Byte,Word,DWork SInt,Int,Dint,USInt UInt,UDInt,BCD16 BCD32,Real	I,Q,M,L,D, hằng số	Giá trị được chuyển đổi
OUT	Byte,Word,DWork SInt,Int,Dint,USInt UInt,UDInt,BCD16 BCD32,Real	I,Q,M,L,D	Kết quả chuyển đổi

Mô tả:

Sử dụng lệnh CONVERT để chuyển đổi một phần tử dữ liệu từ một loại dữ liệu này sang một loại dữ liệu khác

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Click vào “????” và chọn kiểu dữ liệu cần thiết. Các kiểu dữ liệu có hỗ trợ:

Một vài trường hợp có thể xảy ra trong phép tính thông qua trạng thái của ENO:

**Bảng 3.45: Một vài trường hợp có thể xảy ra trong phép tính**

ENO	Điều kiện	Kết quả
1	Không có lỗi	Giá trị hợp lệ
0	IN là +/- INF hoặc +/- NaN	IN là +/- INF hoặc +/- NaN
0	Kết quả vượt quá thang đo kiểu dữ liệu ngõ ra	OUT được đặt không đáng kể byte của IN

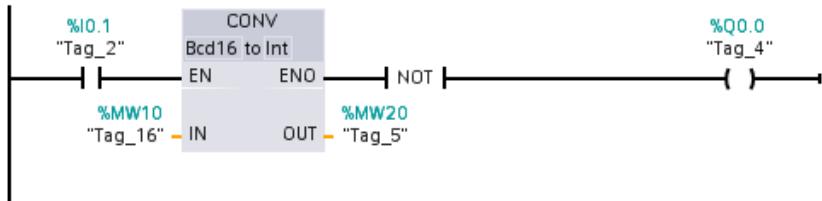
Chú ý:

Chuyển đổi từ BCD16 được giới hạn các kiểu dữ liệu Int. Chuyển đổi từ BCD32 được giới hạn các kiểu dữ liệu dint.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

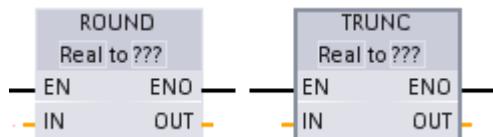


Hình 3.101: Ví dụ về lệnh CONV

Nếu I0.0 = 1, nội dung của MW10 là số BCD mã ba chữ số và chuyển đổi sang một số nguyên (16 bit). Kết quả được lưu trữ trong MW12. Q0.0 = "1", nếu chuyển đổi không được thực hiện thì ENO = EN = 0.

### b. Lệnh Round and truncate

Ký hiệu:



Hình 3.102: Lệnh Round và Truncate.

Các thông số của lệnh:

Bảng 3.46: Thông số của lệnh Round và lệnh Truncate

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
IN	Real	I,Q,M,L,D, hàng số	Giá trị ngõ vào
OUT	SInt,Int,Dint,USInt UInt,UDInt, Real	I,Q,M,L,D	Kết quả

Mô tả:

Lệnh ROUND chuyển một số thực sang số nguyên. Phần nhỏ số thực sẽ được làm tròn với giá trị số nguyên gần nhất. Ví dụ, nếu làm tròn số 10.5 thì kết quả là 10 và 11 đều đúng.

Lệnh TRUNCATE chuyển một số thực sang số nguyên. Phần phân số của số thực được cắt bỏ.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Click vào “???” và chọn kiểu dữ liệu cần thiết. Các kiểu dữ liệu có hỗ trợ.

Một vài trường hợp có thể xảy ra trong phép tính thông qua trạng thái của ENO:

**Bảng 3.47: Một vài trường hợp có thể xảy ra trong phép tính**

ENO	Điều kiện	Kết quả
1	Không có lỗi	Giá trị hợp lệ
0	IN là +/- INF hoặc +/- NaN	IN là +/- INF hoặc +/- NaN

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

### c. Lệnh Ceiling and floor

Ký hiệu:



**Hình 3.103: Lệnh CEIL và FLOOR.**

Các thông số của lệnh (tương tự lệnh ROUND)

Mô tả:

Lệnh CELL chuyển đổi một số thực (Real hoặc LReal) sang số nguyên gần nhất lớn hơn hoặc bằng số thực tế được chọn (IEEE "tròn + vô cực").

Lệnh FLOOR chuyển đổi một số thực (Real hoặc LReal) sang số nguyên gần nhất nhỏ hơn hoặc bằng số thực tế được chọn (IEEE "tròn - vô cực")..

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Click vào “???” và chọn kiểu dữ liệu cần thiết. Các kiểu dữ liệu có hỗ trợ:

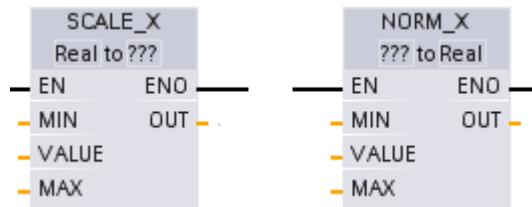
Một vài trường hợp có thể xảy ra trong phép tính (tương tự lệnh ROUND)

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

#### d. Lệnh Scale and normalize

Ký hiệu:



Hình 3.104: Lệnh SCARE\_X và NORM\_X.

Các thông số của lệnh

Bảng 3.48: Thông số của lệnh SCARE và NORM

Tham số	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Cho phép ngõ vào
ENO	BOOL	I,Q,M,L,D	Cho phép ngõ ra
MIN	SInt,Int,Dint,USInt UInt,UDInt, Real	I,Q,M,L,D, hàng số	Giới hạn dưới của thang giá trị
MAX	SInt,Int,Dint,USInt UInt,UDInt, Real	I,Q,M,L,D, hàng số	Giới hạn trên của thang giá trị
OUT	SInt,Int,Dint,USInt UInt,UDInt, Real	I,Q,M,L,D	Kết quả

Mô tả:

Lệnh SCARE\_X: quy mô giá trị tham số thực sự với ( $0.0 \leq$  giá trị  $\leq 1.0$ ) trong các loại dữ liệu và giá trị phạm vi quy định bởi MIN và MAX thông số:

$$OUT = \frac{\text{giá trị} - \text{MIN}}{\text{MAX} - \text{MIN}} + \text{MIN}$$

Lệnh NORM\_X: bình thường hóa các giá trị tham số trong phạm vi giá trị quy định bởi MIN và MAX thông số:

$$\text{OUT} = (\text{giá trị} - \text{MIN}) / (\text{MAX} - \text{MIN}), \text{ với } (0.0 \leq \text{OUT} \leq 1.0)$$

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Click vào “????” và chọn kiểu dữ liệu cần thiết. Các kiểu dữ liệu có hỗ trợ.

Đối với SCALE\_X: Tham số MIN, MAX, và OUT phải cùng loại dữ liệu.

Đối với NORM\_X: Tham số MIN, giá trị, và MAX phải cùng loại dữ liệu.

Một vài trường hợp có thể xảy ra trong phép tính thông qua trạng thái của ENO:

**Bảng 3.49: Bảng trạng thái của ENO**

ENO	Điều kiện	Kết quả
1	Không có lỗi	Giá trị hợp lệ
0	Kết quả vượt quá thang đo OUT	Một phần số thực đưa ra OUT
0	Tham số MAX $\leq$ MIN	SCALE_X: một phần số thực NORM_X: giá trị được điền vào độ lớn từ word
0	Value = +/- INF hoặc +/- NaN	Giá trị ghi ra OUT

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ về lệnh SCALE\_X:



**Hình 3.105: Ví dụ về lệnh SCARE\_X.**

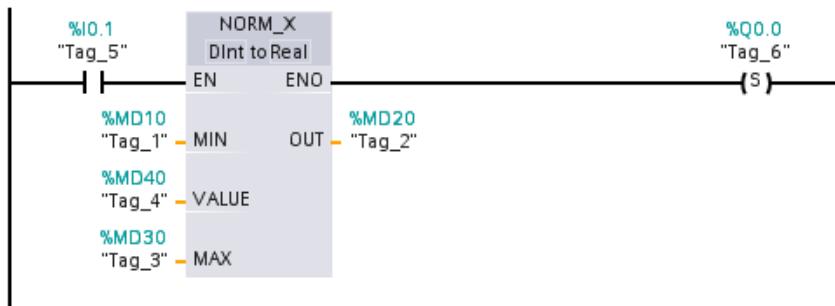
Với:

$$MD10 = 10, MD20 = 0.5, MD30 = 30$$

Nếu quá trình thi hành lệnh thành công thì:

$$MD40 = 20, ENO = 1, Q0.0 = 1$$

Ví dụ về lệnh NORM\_X:



Hình 3.106: Ví dụ về lệnh NORM\_X.

$$MD10 = 10, MD20 = 20, MD30 = 30$$

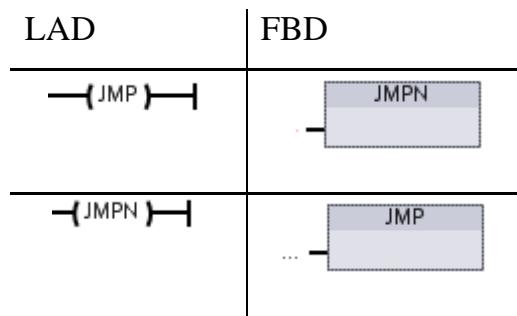
Nếu quá trình thi hành lệnh thành công thì:

$$MD40 = 0.5, ENO = 1, Q0.0 = 1$$

### 3.3.8 Nhóm lệnh điều khiển chương trình

#### a. Lệnh nhảy

Ký hiệu:



Hình 3.107: Lệnh nhảy

Mô tả:

Lệnh nhảy rẽ nhánh chương trình đến một đoạn lệnh được đánh dấu bằng một nhãn. Nhãn dùng để đánh dấu vị trí cho các lệnh nhảy.

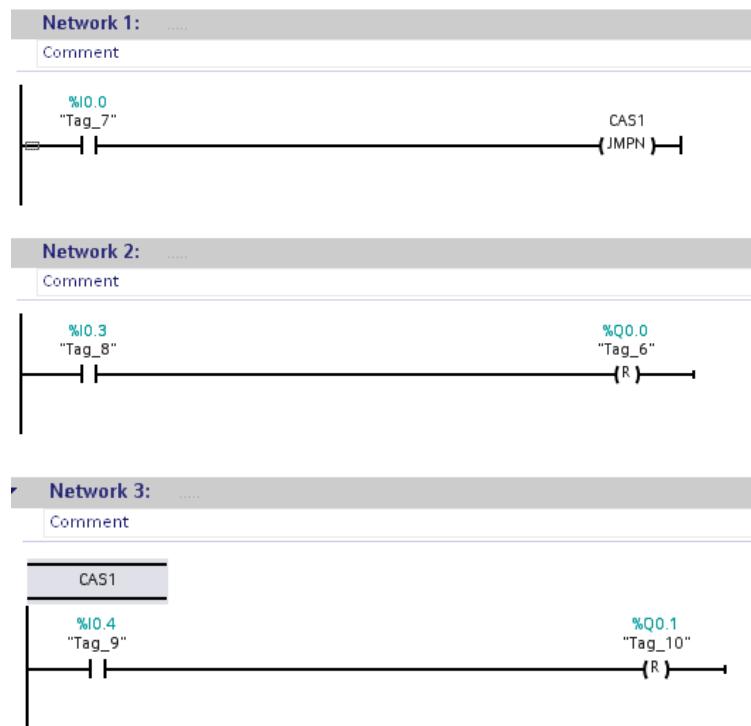
Label\_name: vị trí cần nhảy đến. - Lệnh nhảy JMP sẽ thực thi khi ngõ vào được tích cực mức 1. Lệnh sẽ nhảy đến vị trí lệnh đầu tiên của nhãn và tiếp tục thi hành lệnh.

Lệnh nhảy JMPN sẽ thực thi khi ngõ vào ở mức 0. Lệnh sẽ nhảy đến vị trí lệnh đầu tiên của nhãn và tiếp tục thi hành lệnh.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bên phải một lệnh cạnh.

Ví dụ:

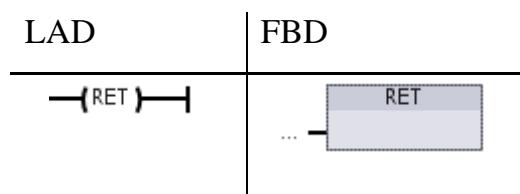


Hình 3.108: Ví dụ về lệnh nhảy

Nếu đầu vào I0.0 = "0", lệnh sẽ thực thi. Chương trình sẽ tạm dừng tại lệnh JMPN để nhảy đến nhãn CAS1 mà lệnh JMPN chỉ định để thi hành lệnh trong nhãn này trước sau đó mới trở về thực thi các lệnh đặt sau lệnh JMPN. Tại nhãn CAS1 trong network 3, nếu đầu vào I0.4 = 1 thì Q0.1 on.

### b. Lệnh RET

Ký hiệu:



### Hình 3.109: Lệnh RET.

Mô tả:

Lệnh RET được sử dụng để kết thúc một đoạn chương trình được gọi và quay trở lại ngay vị trí lệnh gọi chương trình con để thực hiện tiếp các lệnh kế tiếp.

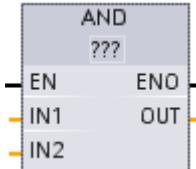
Vị trí đặt lệnh:

Có thể được đặt ở đầu hoặc cuối của một đoạn lệnh.

### 3.3.9 Nhóm lệnh toán học

#### a. Lệnh AND, OR và XOR

Ký hiệu:



Hình 3.110: Lệnh AND

Các thông số của lệnh

Bảng 3.50: Thông số của lệnh AND

Tham số	Kiểu dữ liệu	Vùng nhớ	Diễn tả
EN	BOOL	I, Q, M, L, D	Cho phép ngõ vào
ENO	BOOL	I, Q, M, L, D	Cho phép ngõ ra
IN1	BYTE, WORD DWORD	I, Q, M, L, D hoặc hằng số	Giá trị đầu tiên cho toán hạng logic
IN2	BYTE, WORD DWORD	I, Q, M, L, D hoặc hằng số	Giá trị thứ 2 cho toán hạng logic
OUT	BYTE, WORD DWORD	I, Q, M, L, D	Kết quả của toán hạng

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Lệnh AND, OR và XOR đều là các lệnh thực hiện phép toán logic theo đúng chức năng và tên gọi của chúng. Kết quả là giá trị của phép tính logic từng bit tương ứng của các ngõ vào IN. Kết quả sẽ đưa ra ngõ ra OUT. ENO luôn luôn là TRUE sau khi thực hiện phép tính.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Chú ý:

Có thể thêm vào các ngõ vào IN bằng cách click chuột vào biểu

tượng tạo hoặc click chuột phải vào ngõ vào IN và chọn lệnh chèn ngõ vào “Insert input”

Để xóa một ngõ vào click chuột phải và chọn lệnh xóa ngõ vào “Delete”.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



**Hình 3.111: Lệnh AND**

Với:

$$MW0 = 01010101 \ 01010101, MW2 = 00000000 \ 00001111$$

Nếu đầu vào I0.0 = 1, lệnh được thực thi. Giá trị tại đầu vào MW0 được and từng bit với giá trị đầu vào MW2. Kết quả đưa đến đầu ra MW10 và MW = 00000000 00000101. Khi đó ENO và Q0.0 on.

## b. Lệnh đảo

Ký hiệu:



**Hình 3.112: Lệnh đảo**

Các thông số của lệnh

Bảng 3.51: Thông số của lệnh

Tham số	Kiểu dữ liệu	Vùng nhớ	Điễn tả
EN	BOOL	I, Q, M, D, L	Cho phép ngõ vào
ENO	BOOL	I, Q, M, D, L	Cho phép ngõ vào
IN	BYTE, WORD, DWORD, USINT, UINT, UDINT, SINT, INT, DINT	I, Q, M, D, L hoặc hằng số	Giá trị vào
OUT	BYTE, WORD, DWORD, USINT, UINT, UDINT, SINT, INT, DINT	I, Q, M, D, L	Phản đảo của giá trị ở ngõ vào IN

Click vào “???” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Lệnh đảo là phép toán nhị phân đảo tham số vào IN. Lệnh đảo sẽ thực hiện đảo từng bit nhị phân tham số IN. Kết quả của phép tính sẽ được đưa ra ngõ ra OUT. Nếu IN = 1 thì sau khi thi hành lệnh ngõ ra OUT = 0

Nếu IN = 0 thì sau khi thi hành lệnh ngõ ra OUT = 1

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

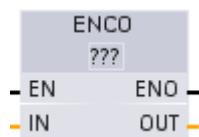
Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

### c. Lệnh mã hóa và giải mã

#### Lệnh mã hóa (ENCO):

Ký hiệu:



Hình 3.113 Lệnh mã hóa

Các thông số của lệnh:

Bảng 3.52: Thông số của lệnh mã hóa

Tham số	Kiểu dữ liệu	Vùng nhớ	Diễn tả
EN	BOOL	I, Q, M, D, L	Cho phép ngõ vào
ENO	BOOL	I, Q, M, D, L	Cho phép ngõ vào
IN	BYTE, WORD, DWORD	I, Q, M, D, L hoặc hằng số	Giá trị vào
OUT	INT	I, Q, M, D, L	Giá trị ngõ ra

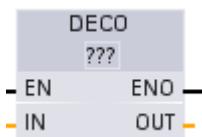
Click vào “???” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Lệnh ENCO sẽ mã hóa một mẫu bit cho một số nhị phân. Lệnh này sẽ mã hóa số nhị phân ở ngõ vào IN và trả kết quả tại OUT. Kết quả là số tương ứng với vị trí bit 1 của giá trị tham số vào IN.

### Lệnh giải mã (DECD):

Ký hiệu:



Hình 3.114: Lệnh giải mã

Các thông số của lệnh (tương tự như lệnh ENCO)

Click vào “???” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Lệnh DECO sẽ giải mã một mẫu bit. Lệnh này sẽ giải mã tham số vào IN và cho ra số nhị phân tương ứng. Số nhị phân ở ngõ ra sẽ được đặt một bit lên 1 (các bit còn lại = 0) tại vị trí được chỉ định từ tham số đầu vào.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Một vài trường hợp có thể xảy ra trong phép tính thông qua trạng thái của ENO.

**Bảng 3.53: Một vài trường hợp có thể xảy ra khi thực hiện lệnh DECO**

ENO	Điều kiện	Kết quả
1	Không có lỗi	Thành phần bit hợp lý
0	IN là 0	Ngõ ra được đặt lên 0

Tham số OUT của lệnh DECO sẽ có kiểu dữ liệu của một Byte, Word, hoặc Dword. Nếu giá trị của tham số IN vượt quá giao có giá trị thì phép toán modulo được thực hiện để lược bỏ các bit dấu.

Tham số IN của lệnh DECO:

- 3 bits (giá trị từ 0-7) IN thì đặt 1 bit trong Byte ngõ ra.
- 4-bits (giá trị từ 0-15) IN thì đặt 1 bit trong Word ngõ ra.
- 5 bits (giá trị từ 0-31) IN thì đặt 1 bit trong DWord ngõ ra.

Ví dụ:

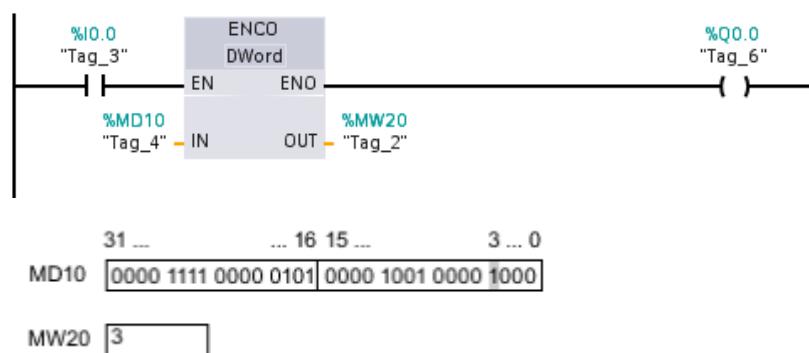
**Bảng 3.54: Ví dụ về lệnh DECO**

Ngõ vào DECO			Ngõ ra DECO
Ngõ ra là byte	Ngõ vào Min	0	00000001
	Ngõ vào Max	7	10000000
Ngõ ra là byte	Ngõ vào Min	0	0000000000000001
	Ngõ vào Max	15	1000000000000000
Ngõ ra là byte	Ngõ vào Min	0	0001
	Ngõ vào Max	31	100

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:





Giá trị ngõ vào và ra:

MW10	<input type="text" value="3"/>
	31 ...                    ... 16 15 ...                    3 ... 0
MD20	<input type="text" value="0000 0000 0000 0000"/> <input type="text" value="0000 0000 0000 1000"/>

Hình 3.115: Ví dụ về lệnh giải mã

#### d. Lệnh chọn, đa hợp và giải đa hợp

##### Lệnh chọn (SEL)

Ký hiệu:



Hình 3.116 Lệnh chọn.

Các thông số của lệnh

Bảng 3.55: Thông số của lệnh SEL

Tham số	Kiểu dữ liệu	Vùng nhớ	Diễn tả
EN	BOOL	I, Q, M, D, L	Cho phép ngõ vào
ENO	BOOL	I, Q, M, D, L	Cho phép ngõ ra
G	BOOL	I, Q, M, D, L	Selection switch
IN0	BYTE,WORD,DWORD, USINT,UINT,UDINT, SINT,INT,DINT, REAL, TIME, CHAR	I, Q, M, D, L hoặc hằng số	Giá trị ngõ vào thứ nhất
IN1	BYTE,WORD,DWORD, USINT,UINT,UDINT,	I, Q, M, D, L hoặc hằng số	Giá trị ngõ vào thứ hai

	SINT, INT, DINT, REAL, TIME, CHAR		
OUT	BYTE, WORD, DWORD, USINT, UINT, UDINT, SINT, INT, DINT, REAL, TIME, CHAR	I, Q, M, D, L	Giá trị của ngõ vào được chọn

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Lệnh chọn sẽ chọn một giá trị từ các ngõ vào và xuất kết quả ra ngõ ra OUT, kết quả này phụ thuộc vào tham số G. Nếu G = 0, giá trị tại đầu vào IN0 được sao chép. Khi G = 1, giá trị đầu vào IN1 được sao chép đến OUT.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Tham số của ngõ vào và ra phải giống nhau.

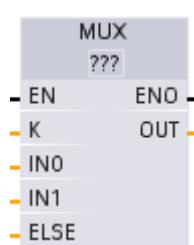
Ngõ ra ENO luôn TRUE sau khi SEL thi hành lệnh.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

### Lệnh đa hợp:

Ký hiệu:



Hình 3.117: Lệnh đa hợp

Các thông số của lệnh:

Bảng 3.56: Thông số của lệnh đa hợp

Tham số	Kiểu dữ liệu	Vùng nhớ	Diễn tả
EN	BOOL	I, Q, M, D, L	Cho phép ngõ vào
ENO	BOOL	I, Q, M, D, L	Cho phép ngõ ra

K	UINT	I, Q, M, D, L	Xác định các đầu vào có nội dung được sao chép
IN0	BYTE,WORD,DWORD, USINT, UINT, UDINT, SINT, INT, DINT, REAL, TIME, CHAR	I, Q, M, D, L hoặc hằng số	Ngõ vào thứ nhất có thể dùng
IN	BYTE,WORD,DWORD, USINT, UINT, UDINT, SINT, INT, DINT, REAL, TIME, CHAR	I, Q, M, D, L hoặc hằng số	Ngõ vào có thể dùng
ELSE	BYTE,WORD,DWORD, USINT, UINT, UDINT, SINT, INT, DINT, REAL, TIME, CHAR	I, Q, M, D, L hoặc hằng số	Xác định giá trị được sao chép khi $K > n$
OUT	BYTE,WORD,DWORD, USINT, UINT, UDINT, SINT, INT, DINT, REAL, TIME, CHAR	I, Q, M, D, L	Giá trị được sao chép vào ngõ ra

Click vào “???” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Lệnh đa hợp sẽ sao chép giá trị của một tham số vào cho ngõ ra OUT, phụ thuộc vào tham số K. Nếu vượt quá giá trị tham số K, giá trị ELSE sẽ được đưa ra OUT.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Tham số của ngõ vào và ra phải giống nhau.

Chú ý:

Có thẻ thêm vào các ngõ vào IN bằng cách click chuột vào biểu

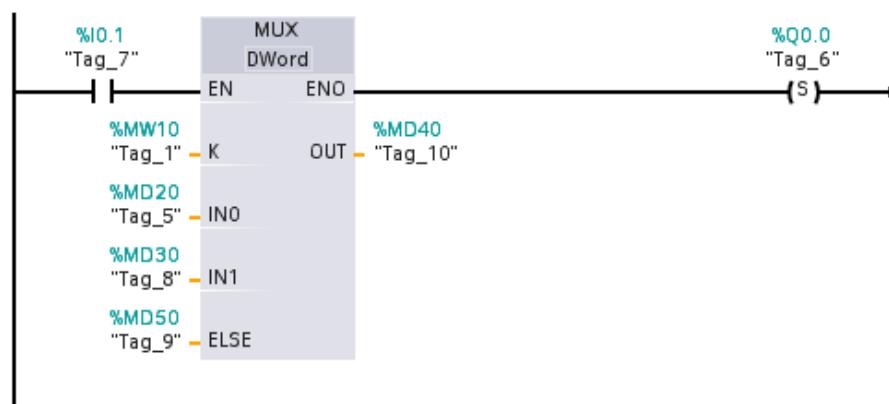
tượng tạo  hoặc click chuột phải vào ngõ vào IN và chọn lệnh chèn ngõ vào “Insert input”

Để xóa một ngõ vào click chuột phải và chọn lệnh xóa ngõ vào “Delete”.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:



Hình 3.118: Ví dụ về lệnh đa hợp

Trong đó:

- K MW10 = 1
- IN0 MD20 = DW#16#00000000
- IN1 MD30 = DW#16#FFFFFF
- ELSE MD50 = DW#16#FFFF0000

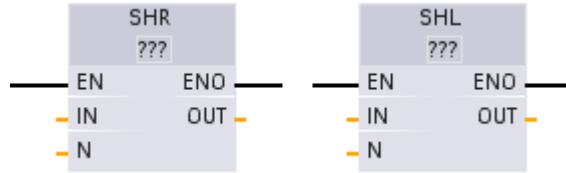
Thì:

- OUT MD40 = DW#16#FFFFFF

### 3.3.10 Nhóm lệnh dịch và quay

#### a. Lệnh dịch

Ký hiệu:



**Hình 3.119: Lệnh dịch**

Các thông số của lệnh

**Bảng 3.57: Thông số của lệnh dịch**

Tham số	Kiểu dữ liệu	Vùng nhớ	Diễn tả
EN	BOOL	I, Q, M, L, D	Cho phép ngõ vào
ENO	BOOL	I, Q, M, L, D	Cho phép ngõ ra
IN	BYTE,WORD,WORD, USINT,INT,UDINT, SINT, INT, DINT,	I, Q, M, L, D hoặc hằng số	Giá trị được dịch
IN	UINT	I, Q, M, L, D hoặc hằng số	Số bit được dịch
OUT	BYTE,WORD,DWORD ,USINT,UINT,UDINT, SINT, INT, DINT	I, Q, M, L, D	Kết quả của dịch toán tử

Click vào “???” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Lệnh ghi dịch (SHL và SHR) được sử dụng để dịch một nhóm bit của tham số vào IN. Kết quả được đưa ra tham số OUT. Tham số N sẽ quy định có bao nhiêu bit sẽ được dịch.

SHL: Dịch trái.

SHR: Dịch phải.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Chú ý:

Với N=0 thì không có bit nào được dịch và giá trị ngõ vào IN bằng ngõ ra.

Số 0 sau khi dịch thì kết quả không đổi.

Nếu số lượng vị trí dịch (N) vượt quá số lượng các bit giá trị cần dịch (8 Byte, 16 cho Word, 32 cho Dword), thì tất cả các giá trị bit ban đầu sẽ được chuyển ra ngoài và thay thế bằng số không (không được chỉ định để OUT).

ENO luôn là TRUE sau khi lệnh thực thi.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

**Bảng 3.58: Ví dụ về lệnh dịch**

Dịch các bit của 1 word sang trái bằng việc chèn các số 0 từ bên phải(N=1)			
IN	1110 0010 1010 1101	Giá trị ngõ ra trước khi dịch:	1110 0010 1010 1101
		Sau khi dịch trái lần thứ nhất:	1100 0101 0101 1010
		Sau khi dịch trái lần thứ hai:	1000 1010 1011 0100
		Sau khi dịch trái lần thứ ba:	0001 0101 0110 1000

### b. Lệnh quay

Ký hiệu:



**Hình 3.120 Lệnh quay**

Các thông số của lệnh:

**Bảng 3.59: Thông số của lệnh quay**

Tham số	Kiểu dữ liệu	Vùng nhớ	Diễn tả
EN	BOOL	I, Q, M, L, D	Cho phép ngõ vào
ENO	BOOL	I, Q, M, L, D	Cho phép ngõ ra
IN	BYTE,WORD,D,DWORD	I, Q, M, L, D hoặc hằng số	Giá trị được quay
N	UINT	I, Q, M, L, D	Số bit được quay

		hoặc hằng số	
OUT	BYTE,WORD,D,DWORD	I, Q, M, L, D	Kết quả của toán tử

Click vào “????” và chọn kiểu dữ liệu cần thiết.

Mô tả:

Sử dụng lệnh quay trái (ROR) và quay phải (ROL) để quay mẫu bit của tham số IN. Kết quả được trả về tham số OUT. Thông số N xác định số lượng vị trí bit sẽ quay.

Khi ngõ vào cho phép EN lên 1 thì lệnh sẽ thực thi.

Chú ý:

Đối với N = 0, lệnh quay không xảy ra. Giá trị IN = OUT.

Các bit dữ liệu sẽ dịch theo một chiều theo vòng. Bit có vị trí 0 sẽ chuyển lên vị trí 1 và bit có vị trí 1 sẽ dịch lên vị trí 2 và tương tự cho các bit còn lại. Bit ở vị trí cuối cùng sẽ đặt ở vị trí 0. Do đó giá trị của các bit không bị mất đi.

Nếu số lượng vị trí bit để quay (N) vượt quá số lượng bit của giá trị (8 Byte, 16 cho Word, 32 cho Dword), thì lệnh quay vẫn được thực hiện.

ENO luôn TRUE sau khi lệnh quay kết thúc.

Vị trí đặt lệnh:

Lệnh có thể đặt ở vị trí bất kỳ trong network.

Ví dụ:

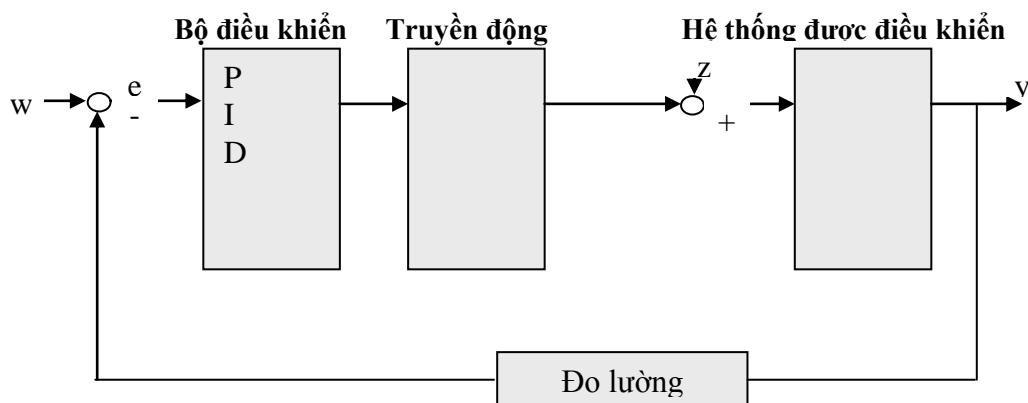
**Bảng 3.60: Ví dụ về lệnh quay**

Xoay các bit sang phải			
IN	0100 0000 0000 0001	Giá trị ra trước khi quay :	0100 0000 0000 0001
		Sau khi quay phải lần thứ nhất :	1010 0000 0000 0000
		Sau khi quay phải lần thứ hai :	0101 0000 0000 0000

### 3.3.11 Lệnh vòng lặp PID

#### a. Bộ điều khiển PID

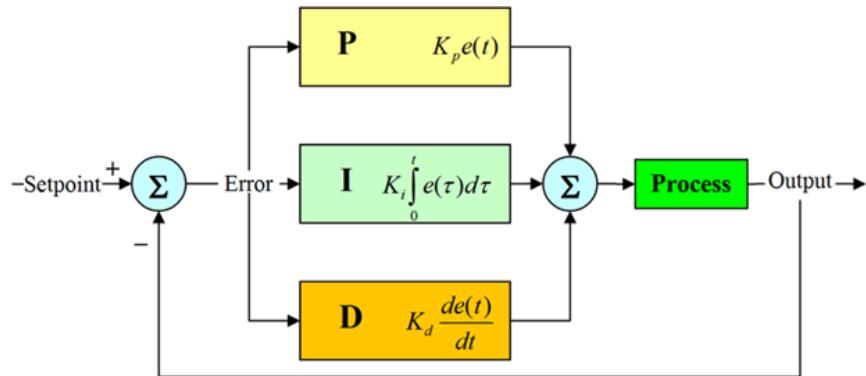
Bộ điều khiển vi tích phân tỉ lệ (bộ điều khiển PID) hoạt động theo cơ chế vòng lặp được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp. Bộ điều khiển PID được sử dụng phổ biến nhất trong số các bộ điều khiển hồi tiếp.



Hình 3.121: Mô hình một hệ thống điều khiển bằng PID

Bộ điều khiển PID tính toán giá trị "sai số" là hiệu số giữa giá trị đo được và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào. Để đạt được kết quả tốt nhất, các thông số PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống. Trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống.

Giải thuật tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt, do đó đôi khi nó còn được gọi là điều khiển ba khâu: khâu tỉ lệ, tích phân và vi phân viết tắt là P, I, và D. Giá trị tỉ lệ xác định tác động của sai số hiện tại, giá trị tích phân xác định tác động của tổng các sai số quá khứ, và giá trị vi phân xác định tác động của tốc độ biến đổi sai số. Tổng chập của ba tác động này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển như vị trí của van điều khiển hay bộ nguồn của phần tử gia nhiệt...



Hình 3.122 Sơ đồ khái bô điều khiển PID

Hàm truyền của khâu PID [2] (trang 180):

$$G(s) = K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s}$$

Trong đó:

- $K_P$  là độ lợi khâu tỷ lệ.
- $K_I$  là độ lợi khâu tích phân.
- $K_D$  là độ lợi khâu vi phân.

Ảnh hưởng của các thông số  $K_P$ ,  $K_I$ ,  $K_D$ , lên hệ thống cho bởi bảng sau:

Bảng 3.61: Sự ảnh hưởng của các thông số  $K_P$ ,  $K_D$ ,  $K_I$  lên hệ thống

Thông số	Thời gian khởi động	Quá độ	Thời gian xác lập	Sai số xác lập	Độ ổn định
$K_P$	Giảm	Tăng	Thay đổi nhỏ	Giảm	Giảm cấp
$K_I$	Giảm	Tăng	Tăng	Giảm đáng kể	Giảm cấp
$K_D$	Giảm	Giảm ít	Giảm ít	Về lý thuyết không đổi	Cải thiện nếu $K_D$ nhỏ

### b. Phương pháp điều chỉnh PID

Phương pháp thủ công:

Đặt giá trị đầu tiên của  $K_I$  và  $K_D$  bằng không. Tăng dần  $K_P$  cho đến khi đầu ra của vòng điều khiển dao động, sau đó  $K_P$  có thể được đặt tối

xấp xỉ một nữa giá trị đó để đạt được đáp ứng "1/4 giá trị suy giảm biên độ". Sau đó tăng K<sub>i</sub> đến giá trị phù hợp sao cho đủ thời gian xử lý.

Tuy nhiên, K<sub>i</sub> quá lớn sẽ gây mất ổn định. Cuối cùng, tăng K<sub>d</sub>, nếu cần thiết, cho đến khi vòng điều khiển nhanh có thể chấp nhận được nhanh chóng lấy lại được giá trị đặt sau khi bị nhiễu.

#### Phương pháp Ziegler–Nichols:

Một phương pháp điều chỉnh theo kinh nghiệm khác là phương pháp Ziegler–Nichols, được đưa ra bởi John G. Ziegler và Nathaniel B. Nichols vào những năm 1940. Giống phương pháp trên, độ lợi K<sub>i</sub> và K<sub>d</sub> lúc đầu được gán bằng không. Độ lợi P được tăng cho đến khi nó tiến tới độ lợi tối hạn, K<sub>u</sub>, ở đầu ra của vòng điều khiển bắt đầu dao động. K<sub>u</sub> và thời gian giao động P<sub>u</sub> được dùng để gán độ lợi như sau:

Bảng 3.62: Các thông số PID theo phương pháp Ziegler–Nichols

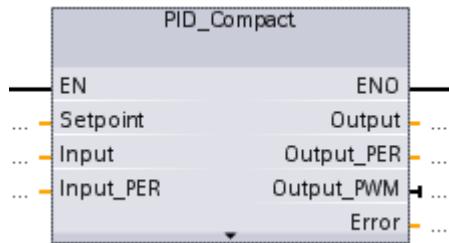
Phương pháp Ziegler–Nichols			
Dạng điều khiển	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
P	0.50K <sub>u</sub>	-	-
PI	0.45K <sub>u</sub>	1.2K <sub>p</sub> / P <sub>u</sub>	-
PID	0.60K <sub>u</sub>	2K <sub>p</sub> / P <sub>u</sub>	K <sub>p</sub> P <sub>u</sub> / 8

#### Phần mềm điều chỉnh PID:

Hầu hết các ứng dụng công nghiệp hiện đại không còn điều chỉnh vòng điều khiển sử dụng các phương pháp tính toán thủ công như trên nữa. Thay vào đó, phần mềm điều chỉnh PID và tối ưu hóa vòng lặp được dùng để đảm bảo kết quả chắc chắn. Những gói phần mềm này sẽ tập hợp dữ liệu, phát triển các mô hình xử lý, và đề xuất phương pháp điều chỉnh tối ưu.

#### c. PID trong PLC S7-1200

Simatic S7-1200 cung cấp 16 vòng lặp PID với khả năng điều chỉnh tự động, cho phép bộ điều khiển xác định thông số vòng lặp gần tối ưu hóa cho hầu hết các ứng dụng điều khiển quá trình thông dụng.



**Hình 3.123: Lệnh PID\_Compact**

PID\_Compact cung cấp bộ điều khiển tự điều chỉnh PID cho chế độ tự động và chế độ bằng tay.

Bộ điều khiển PID sử dụng công thức sau đây để tính toán giá trị đầu ra cho lệnh PID\_Compact [7] (trang 141).

$$y = K_p \left[ (b - w - x) + \frac{1}{T_1 - s} (w - x) + \frac{T_0 - s}{a - T_0 - s - 1} (c - w - x) \right]$$

Trong đó:

x      giá trị xử lý

y      giá trị ngõ ra

z      giá trị đặt

K<sub>p</sub>    độ lợi khâu tỷ lệ

T<sub>I</sub>    thời gian khâu tích phân

T<sub>D</sub>    thời gian khâu tích phân

s      biến số trong biến đổi Laplace

a      hệ số khâu tỷ lệ

b      hệ số khâu tích phân

c      hệ số khâu vi phân

Lệnh PID\_Compact được sử dụng để kiểm soát các quy trình kỹ thuật với đầu vào và đầu ra liên tục.

Lệnh PID\_Compact có thể tính toán các thành phần P, I-, và D trong quá trình khởi động (nếu cấu hình để trước khi điều chỉnh "pretuning"). Cũng có thể cấu hình lệnh để điều chỉnh "tuning" để cho phép tối ưu hóa các thông số. Và để không cần phải tự xác định các thông số.

Các tham số của lệnh PID\_Compact:

**Bảng 3.63: Các thông số của bộ PID\_Compact**

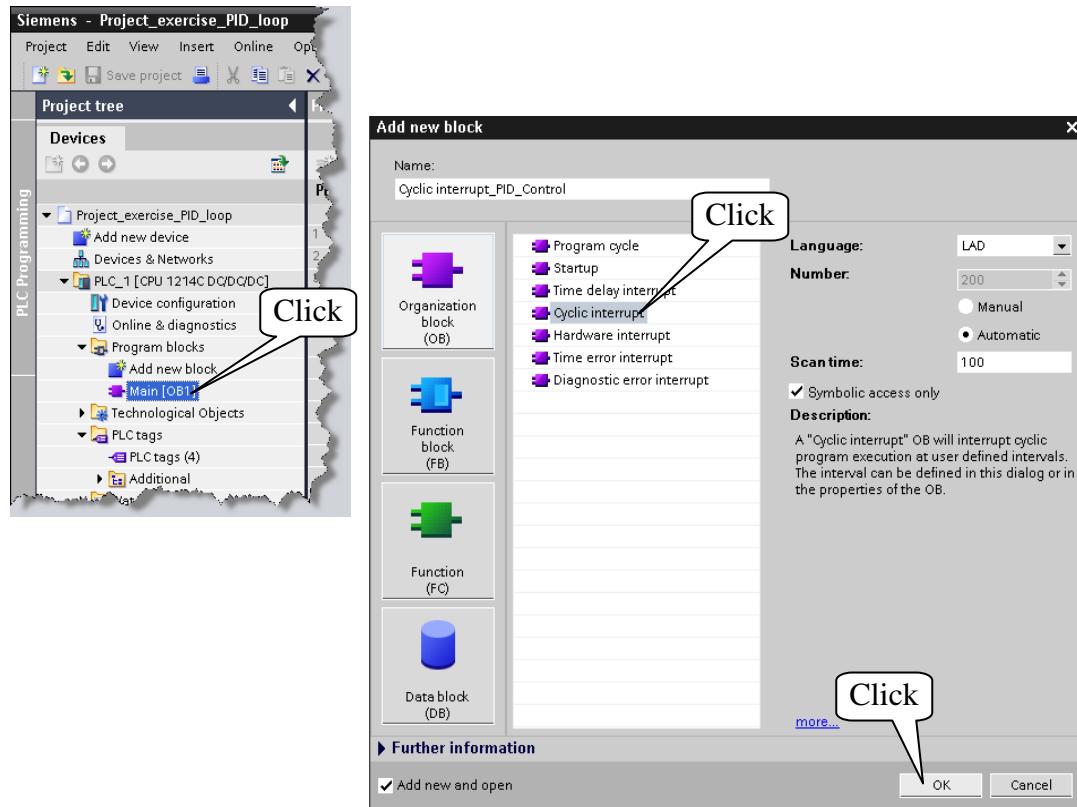
Tham số	Khai báo	Kiểu dữ liệu	Giá trị đầu	Mô tả
Setpoint	INPUT	REAL	0.0	Giá trị đặt cho bộ điều khiển PID. Mặc định là 0.0
Input	INPUT	REAL	0.0	Giá trị để xử lý
Input_PER	INPUT	WORD	W#16#0	Giá trị tương tự
ManualEnable	INPUT	BOOL	FALSE	Cho phép và không cho phép chế độ điều khiển bằng tay.
ManualValue	INPUT	REAL	0.0	Giá trị vào ở chế độ điều khiển bằng tay
Reset	INPUT	BOOL	FALSE	Khởi động lại bộ điều khiển. Sau khi reset giá trị sẽ là TRUE
ScaledInput	OUTPUT	REAL	0.0	Quy đổi giá trị vào
Output	OUTPUT	REAL	0.0	Giá trị ngõ ra
Output_PER	OUTPUT	WORD	W#16#0	Giá trị ra tương tự
Output_PWM	OUTPUT	BOOL	FALSE	Giá trị ra dạng xung
SetpointLimit_H	OUTPUT	BOOL	FALSE	Điểm đặt trên
SetpointLimit_L	OUTPUT	BOOL	FALSE	Điểm đặt dưới
InputWarning_H	OUTPUT	BOOL	FALSE	Mức trên cảnh báo của giá trị vào
InputWarning_L	OUTPUT	BOOL	FALSE	Mức dưới cảnh báo của giá trị vào

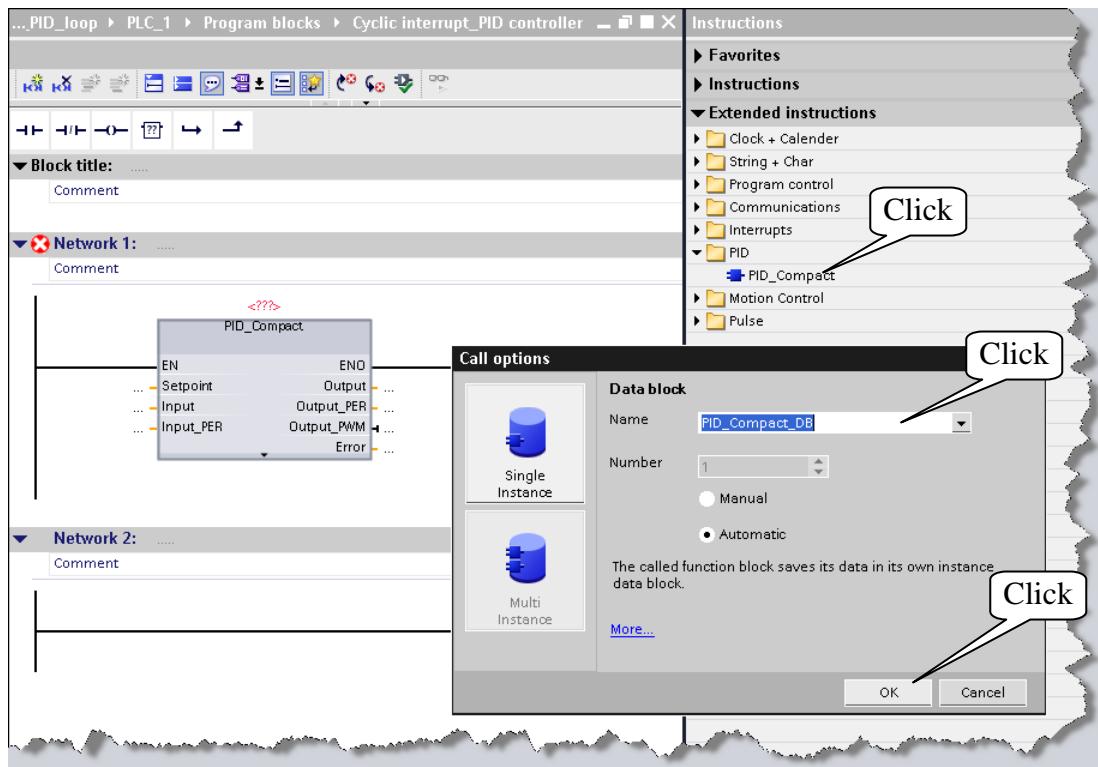
State	OUTPUT	INT	0	Trạng thái hoạt động
Error	OUTPUT	DWORD	W#32#0	Báo lỗi

Chèn một lệnh PID vào network

Để thêm một lệnh PID vào network ta làm như sau:

Bước 1: Chọn vòng lặp và chèn lệnh

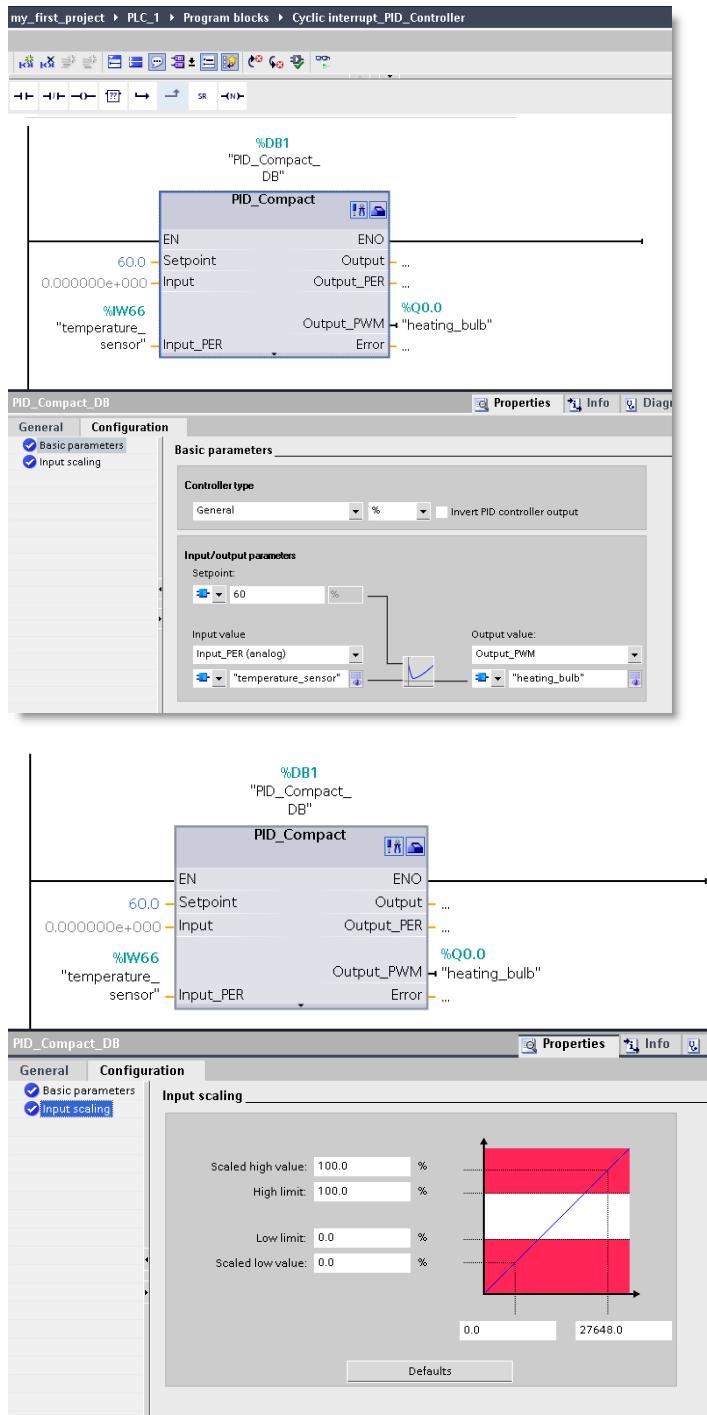




Hình 3.124: Tạo lệnh vòng lặp PID

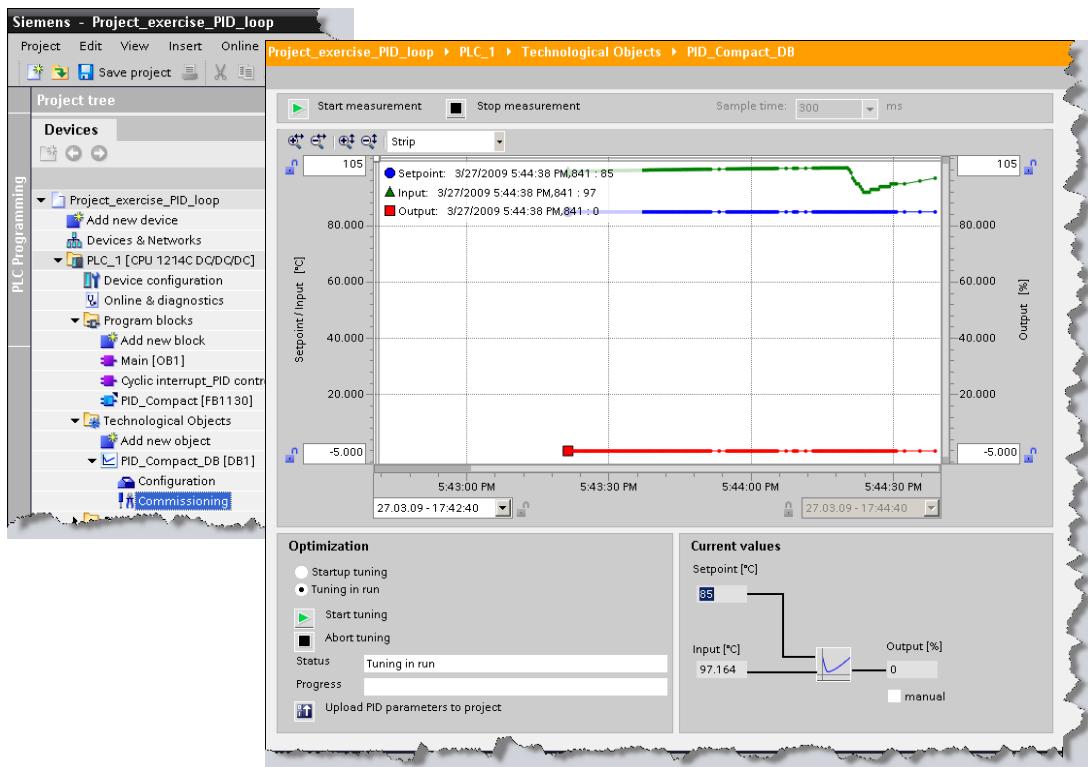
Khi chèn một lệnh PID vào chương trình, STEP 7 sẽ tự động tạo ra một đối tượng và một DB. DB chứa tất cả các tham số của lệnh PID. Mỗi lệnh PID phải có một DB riêng biệt. Do đó phải cấu hình các thông số cho đối tượng.

#### Bước 2: Cấu hình bộ điều khiển PID



**Hình 3.125: Cấu hình cho bộ PID**

Bước 3. Cài đặt chế độ hoạt động cho bộ điều khiển PID



**Hình 3.126: Cài đặt chế độ hoạt động cho bộ điều khiển**

Giải thích các chế độ hoạt động của khối “PID\_compact”:

**Bảng 3.64: Chế độ hoạt động của PID\_Compact**

Chế độ hoạt động	Mô tả
Không hoạt động (Inactive)	<ul style="list-style-type: none"> <li>- Khi chương trình được nạp xuống CPU lần đầu tiên, thì bộ điều khiển PID vẫn ở trạng thái “Inactive”. Trong trường hợp này ta tiến hành chế độ “Self tuning during initial start” trên cửa sổ vận hành.</li> <li>- Trong quá trình điều khiển PID ta cần chuyển sang chế độ “Inactive” khi có lỗi xảy ra hoặc khi ta “Dừng điều khiển”.</li> <li>- “Active errors” được xác định là lỗi khi ta thay đổi chế độ hoạt động.</li> </ul>
Tự điều chỉnh trong quá trình khởi động ban đầu / Tự điều chỉnh trong thời	<ul style="list-style-type: none"> <li>- Chế độ “Self tuning during initial start / Self tuning in the operating point” xảy ra khi được gọi lên từ cửa sổ vận hành.</li> </ul>

điểm hoạt động (Self tuning during initial start / Self tuning in the operating point)	
Điều chỉnh tự động (Automatic mode)	<ul style="list-style-type: none"> <li>- Trong chế độ “Automatic mode”, hàm PID_Compact sẽ tự hiệu chỉnh vòng điều khiển theo các thông số xác định. Bộ điều khiển sẽ chuyển sang chế độ tự động, nếu các điều kiện sau: <ul style="list-style-type: none"> <li>+ Chế độ “Self tuning during initial start” đã hoàn thành.</li> <li>+ Chế độ “Self tuning in the operating point” đã hoàn thành.</li> <li>+ Kiểm tra khôi PID_Compact, và gán thuộc tính “Sate” bằng 3.</li> </ul> </li> </ul>
Điều chỉnh bằng tay (Manual mode)	<ul style="list-style-type: none"> <li>- Các biến số có thể được thiết lập bằng tay nếu bộ điều khiển PID hoạt động ở chế độ “Manual mode”. Chế độ “Manual mode” được chọn như sau: <ul style="list-style-type: none"> <li>+ Giá trị “TRUE” ở thông số “ManualEnable”</li> <li>+ Lựa chọn “Manual manipulated variable”, kiểm tra ở cửa sổ vận hành.</li> </ul> </li> </ul>

**Chú ý:**

*Ngõ ra Output, Output\_PER và Output\_PWM có thể được sử dụng song song.*

*Sau khi thiết lập được chế độ Automatic mode, ta phải cập nhật và lưu lại thông số PID. Ta nhấp vào nút  Upload PID parameters to project. Khi này, thông số PID được lưu trong chương trình trên PLC. PLC mới thực sự hoạt động ở chế độ Automatic mode trong các lần hoạt động tiếp theo.*

### 3.3.12 Lệnh đếm xung tốc độ cao

#### a. Giới thiệu

Bộ đếm tốc độ cao thường được sử dụng để đếm những sự kiện xảy ra với tần số lớn mà các bộ đếm thông thường trong PLC không đếm được. VD: Tín hiệu xung từ encoder...

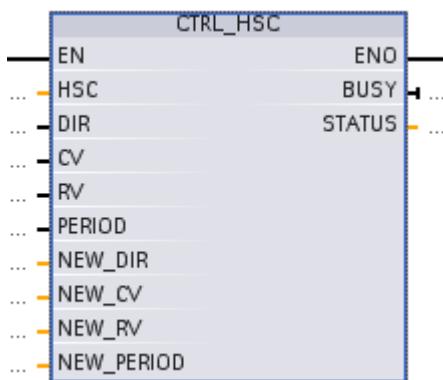
Các CPU S7-1200 khác nhau, thì có số High-speed counters khác nhau:

**Bảng 3.65: High-speed counter trong các CPU**

CPU S7-1200	Số HSC	Tên gọi HSC
CPU 1211C	3	HSC1...3
CPU 1212C	4	HSC1...4
CPU 1214C	6	HSC1...6

#### b. Khối CTRL\_HSC

Ký hiệu:



**Hình 3.127: Khối CTRL\_HSC**

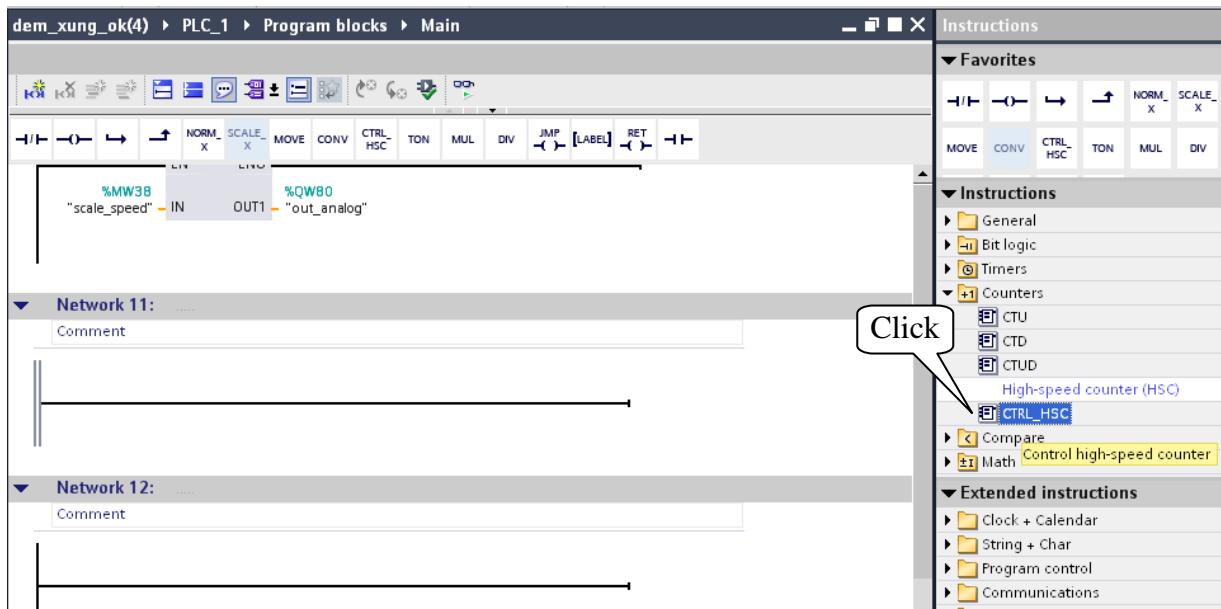
**Bảng 3.66: Thông số của khối CTRL\_HSC**

Thuộc tính	Kiểu dữ liệu	Vùng nhớ	Mô tả
EN	BOOL	I,Q,M,L,D	Kích hoạt bộ đếm
ENO	BOOL	I,Q,M,L,D	Đầu ra

HSC	HW_HSC	L,D, hằng số	Địa chỉ phần cứng bộ đếm tốc độ cao (HW-ID)
DIR	BOOL	I,Q,M,L,D	Cho phép hướng đếm mới
CV	BOOL	I,Q,M,L,D	Cho phép đếm giá trị mới
RV	BOOL	I,Q,M,L,D	Cho phép giá trị tham chiếu mới
PERIOD	BOOL	I,Q,M,L,D	Cho phép chu kỳ mới
NEW_DIR	INT	I,Q,M,L,D	Hướng đếm mới sẽ tải xuống khi DIR =TRUE. 1 = đếm lên; -1 = đếm xuống
NEW_CV	DINT	I,Q,M,L,D	Giá trị đếm mới sẽ tải xuống khi CV = TRUE. Giá trị này từ 2147483648 đến 2147483647
NEW_RV	DINT	I,Q,M,L,D	Giá trị tham chiếu mới sẽ tải xuống khi RV = TRUE
NEW_PERIOD	INT	I,Q,M,L,D	Khoảng thời gian của tần số đo, được nạp xuống khi PEROD = TRUE. Giá trị: 0.01s, 0.1s, 1s
BUSY	BOOL	I,Q,M,L,D	Tình trạng bộ đếm
STATUS	WORD	I,Q,M,L,D	Tình trạng hoạt động

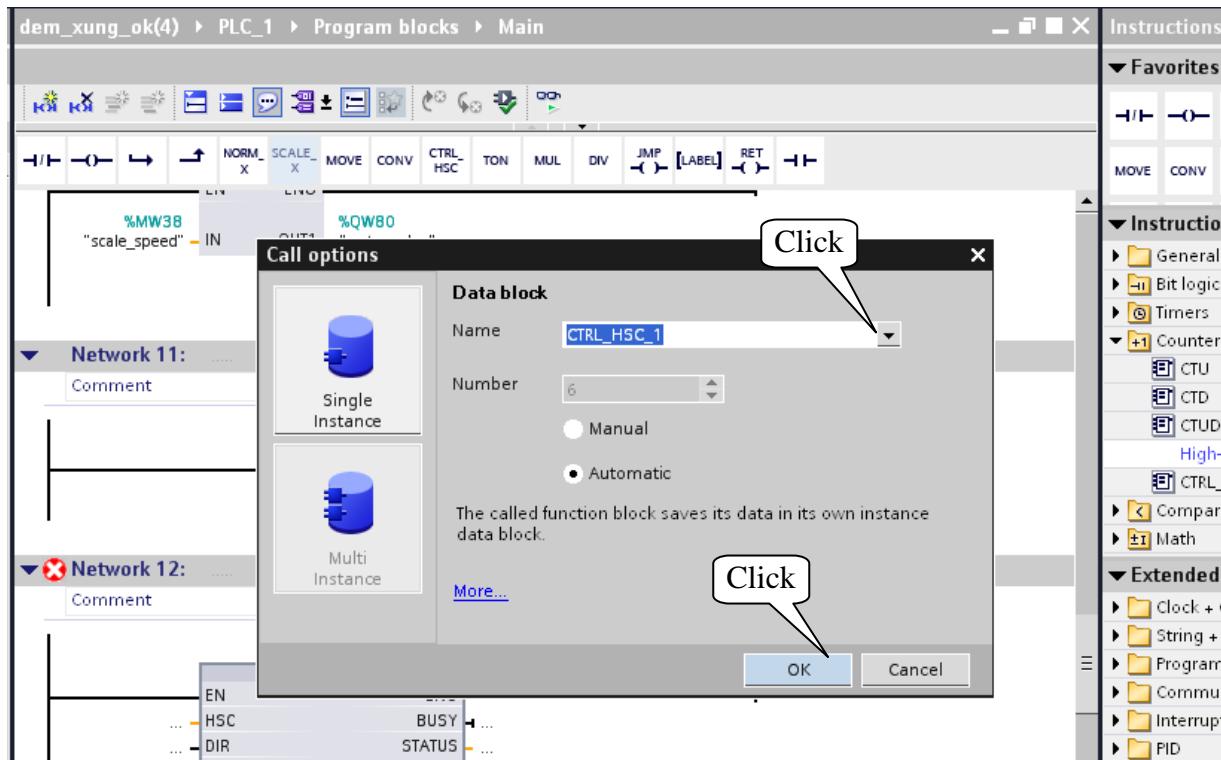
Để sử dụng bộ đếm tốc độ cao, ta thực hiện các bước như sau:

- Bước 1: Thêm bộ CTRL\_HSC



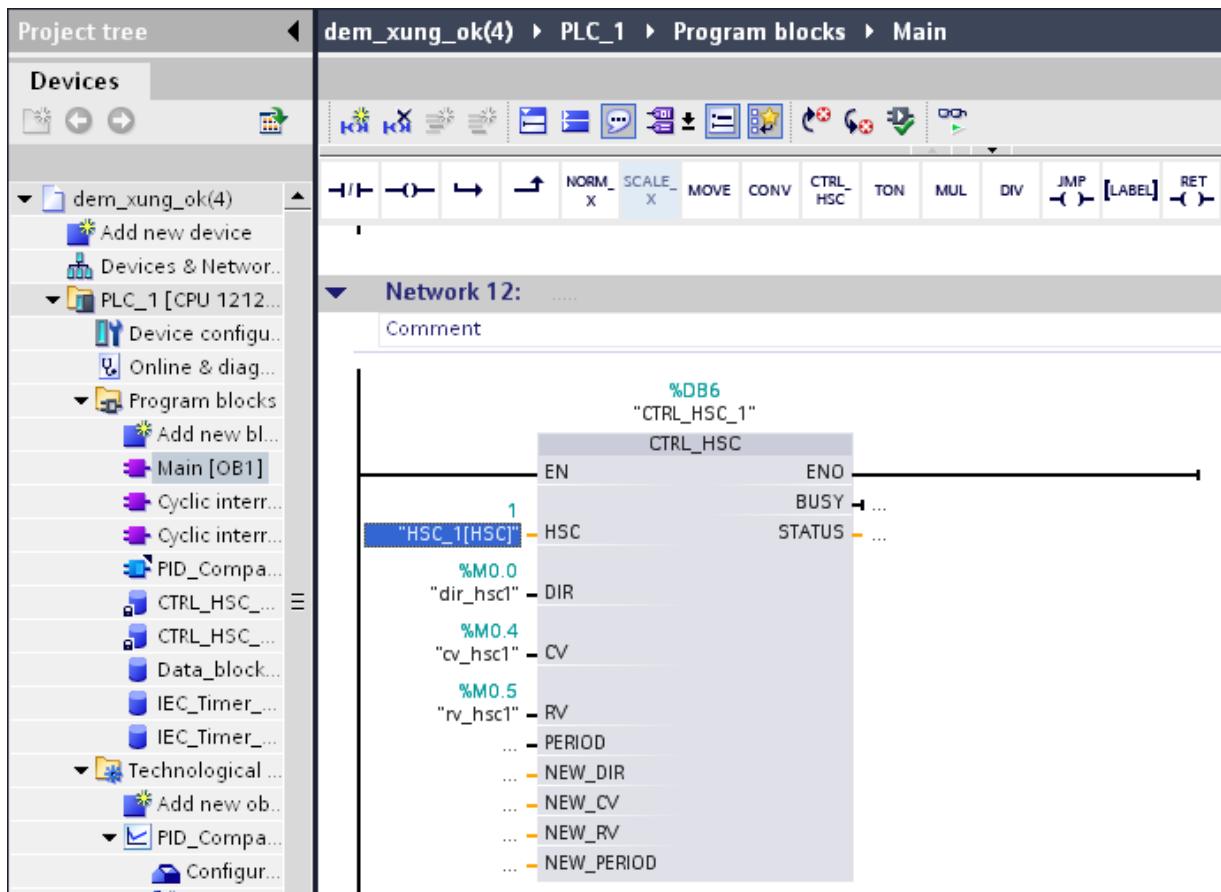
Hình 3.128: Thêm bộ CTRL\_HSC vào chương trình

- Bước 2: Đặt tên cho bộ CTRL\_HSC



Hình 3.129: Đặt tên cho bộ CTRL\_HSC

- Bước 3: Cài đặt các thông số cho CTRL\_HSC



Hình 3.130: Thiết lập các thông số cho khối CTRL\_HSC

### c. Cách thức hoạt động của bộ đếm:

Đầu tiên, các giá trị đặt trước sẽ được nạp vào bộ HSC, và ngõ ra được kích hoạt.

Nếu giá trị hiện tại bằng với giá trị mặc định thì xảy ra ngắn, giá trị mặc định sẽ được tải xuống và trạng thái tín hiệu tiếp theo sẽ đưa ra đầu ra. Nếu một sự kiện ngắn xảy ra vì counter bị reset, thì giá trị mặc định đầu tiên và trạng thái tín hiệu đầu tiên của đầu ra sẽ được thiết lập và vòng lặp sẽ lập lại.

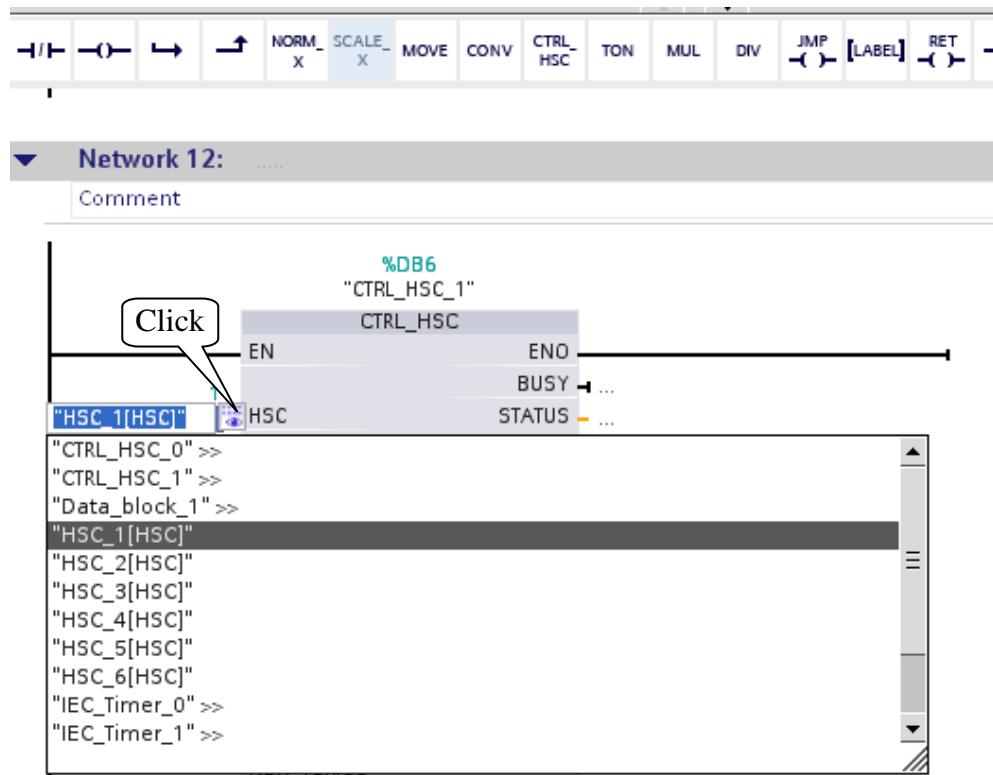
Bảng 3.67: Mã lỗi của bộ CTRL\_HSC

Mã lỗi (Hệ thập lục phân)	Mô tả
0	Không có lỗi
80A1	Địa chỉ phần cứng của bộ đếm tốc độ cao

	không có giá trị
80B1	Đếm định hướng (NEW_DIR) không có giá trị
80B2	Giá trị đếm (NEW_CV) không có giá trị
80B3	Giá trị tham chiếu (NEW_RV) không có giá trị
80B4	Khoảng thời gian tàn số đo (NEW_PERIOD) không có giá trị
80C0	Có quá nhiều truy nhập vào bộ đếm tốc độ cao

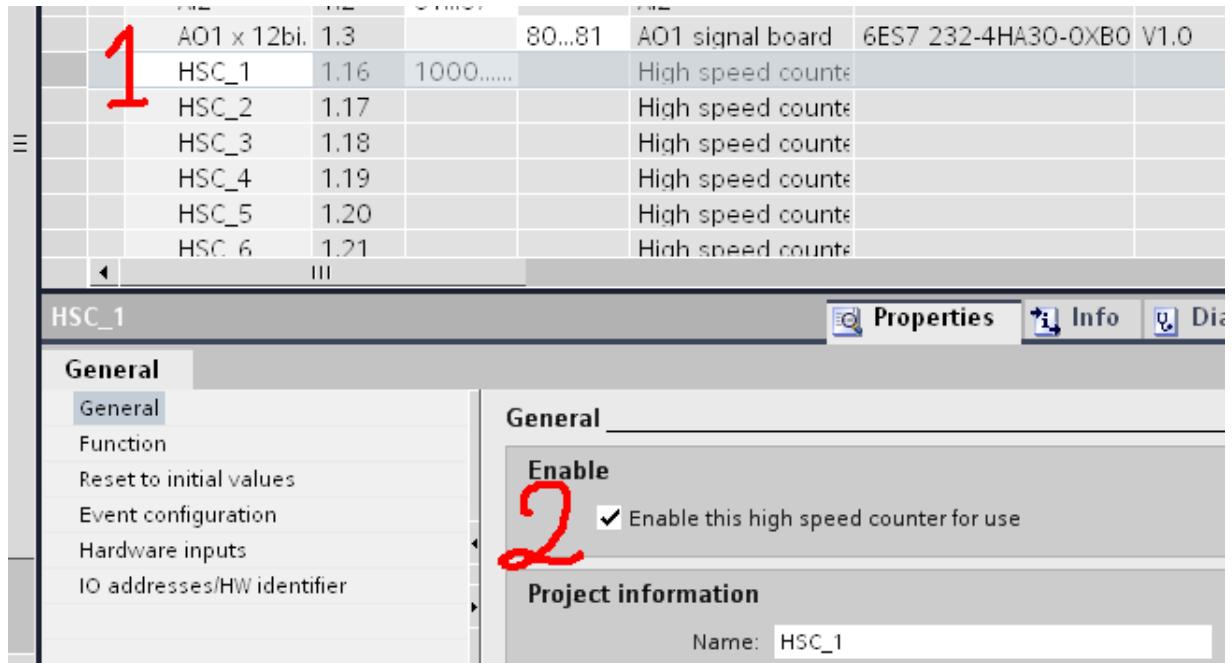
#### d. Cấu hình phần cứng của bộ đếm

Trong khối CTRL\_HSC có ngõ vào HSC để khai báo địa chỉ phần cứng. Ta sẽ chọn bộ đếm HSC được sử dụng:



Hình 3.131: Chọn địa chỉ phần cứng cho khối CTRL\_HSC

Sau khi chọn bộ đếm, ta phải vào mục cấu hình PLC để kích hoạt bộ đếm đã chọn:



Hình 3.132: Cấu hình phần cứng HSC

Quy định đầu vào cho bộ đếm:

Bảng 3.68: Các thông số của bộ đếm

Chế độ đếm	Mô tả	Ngõ vào			
	HSC1	I0.0 (CPU) I4.0 (signal board)	I0.1 (CPU) I4.1 (signal board)	I0.3 (CPU) I4.3 (signal board)	
	HSC2	I0.2 (CPU) I4.2 (signal board)	I0.3 (CPU) I4.3 (signal board)	I0.1 (CPU) I4.1 (signal board)	
	HSC3	I0.4 (CPU)	I0.5 (CPU)	I0.7 (CPU)	
	HSC4	I0.6 (CPU)	I0.7 (CPU)	I0.5 (CPU)	
	HSC5	I1.0 (CPU) I4.0 (Signal board)	I1.1 (CPU) I4.0 (Signal board)	I1.2 (CPU) I4.3 (CPU)	
	HSC6	I1.3 (CPU)	I1.4 (CPU)	I1.5 (CPU)	

Counting/ Frequency	Đếm đơn pha với điều khiển nội	Xung đồng hồ Bộ tạo xung		
Counting		Xung đồng hồ Bộ tạo xung		Reset
Counting/ Frequency	Đếm đơn pha với điều khiển ngoại	Xung đồng hồ Bộ tạo xung	Trực tiếp	
Counting		Xung đồng hồ Bộ tạo xung	Trực tiếp	Reset
Counting/ Frequency	Đếm 2-pha với 2 xung ngõ vào	Xung đồng hồ Bộ tạo xung	Xung đồng hồ Bộ tạo xung	
Counting		Xung đồng hồ Bộ tạo xung	Xung đồng hồ Bộ tạo xung	Reset
Counting/ Frequency	A/B counter	Xung đồng hồ Bộ tạo xung	Xung đồng hồ Bộ tạo xung	
Counting		Xung đồng hồ Bộ tạo xung	Xung đồng hồ Bộ tạo xung	Reset
Trục chuyển động	Bộ tạo xung PWM/PTO	HSC1 và HSC2 hỗ trợ chế độ đếm với bộ tạo xung PTO1 và PTO2: - Với PTO1, HSC1 định Q0.0 cho ngõ ra ra xung - Với PTO2, HSC2 định Q0.2 cho ngõ ra ra xung Q0.1 được sử dụng làm ngõ ra cho chuyển động trực tiếp		

**Chú ý:**

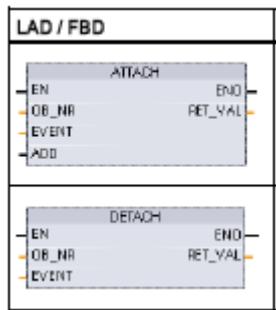
Sau khi kích hoạt (Enable) phần cứng HSC, ta cần kiểm tra cấu hình thành công chưa. Nếu không thành công, ta có thể reset IP để PLC tự nhận phần cứng HSC đã kích hoạt (xem mục 1.2.3).

### 3.3.13 Lệnh Ngắt

Lệnh ngắt cho phép PLC tạm dừng thi hành lệnh tại vị trí xảy ra ngắt để thực hiện một lệnh nào đó gọi là xử lý ngắt. Sau khi chương trình xử lý ngắt hoàn tất, chương trình sẽ quay lại vị trí tạm dừng để tiếp tục thi hành lệnh.

#### a. Lệnh ATTACH và DETACH

Ký hiệu:



Hình 3.133: Lệnh ATTACH và DETACH

Lệnh ATTACH và DETACH dùng để kích hoạt và vô hiệu hóa chương trình con của ngắt.

ATTACH kích hoạt chương trình con OB của ngắt cho phần cứng phục vụ ngắt.

DETACH sẽ vô hiệu hóa chương trình con OB của ngắt cho phần cứng phục vụ ngắt.

Bảng 3.69: Thông số của lệnh ATTACH và DETACH

Tham số và kiểu	Kiểu dữ liệu	Mô tả
OB_NR	IN	OB_ATT Tạo một khối OBs mới để sử dụng bằng cách chọn “Add new block”. Click đúp vào khối OBs và chọn kiểu OBs cần thiết
EVENT	IN	EVENT_ATT Định nghĩa sự kiện ngắt: chọn phần cứng ngắt và click đúp vào vùng tham số để kích hoạt sự kiện ngắt.
ADD (chỉ với ATTACH)	IN	Bool ADD = 0 (mặc định) bằng 0 thì sự kiện ngắt thay thế các ngắt trước ADD = 1 thì ngắt được thêm vào với các ngắt trước trong OBs

RET_VAL	OUT	Int	Mã điều kiện thực thi lệnh
---------	-----	-----	----------------------------

Phần cứng phục vụ ngắn:

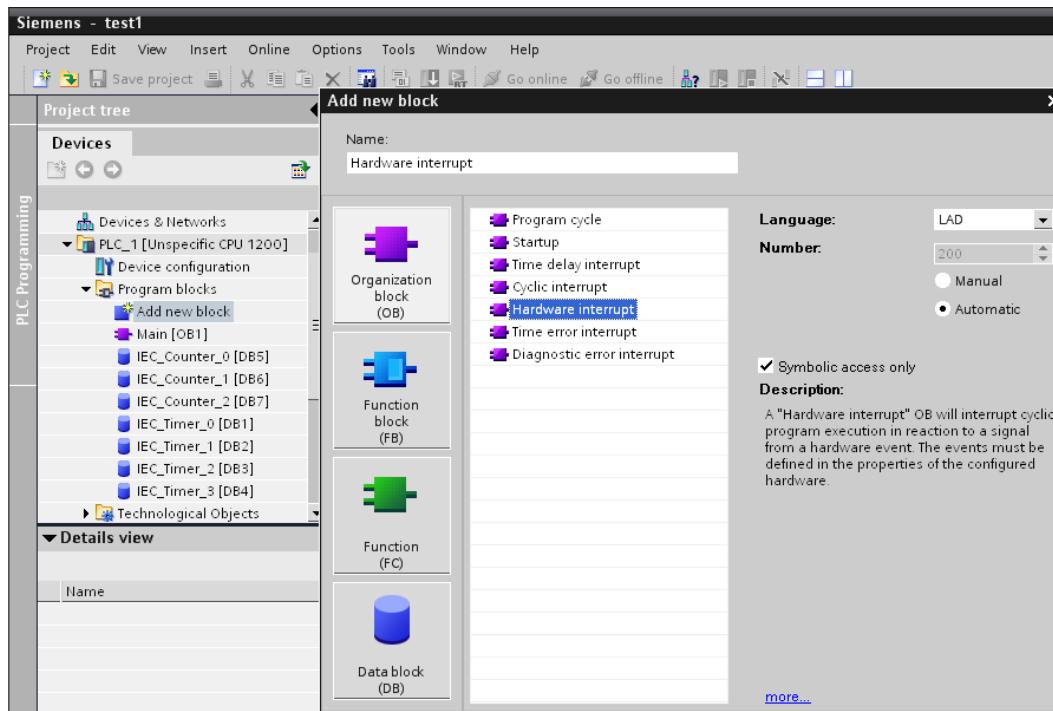
Ngắn cạnh lên: cạnh lên xảy ra khi ngõ vào số chuyển từ OFF lên ON.

Ngắn cạnh xuống: cạnh xuống xảy ra khi ngõ vào số chuyển từ ON xuống OFF.

Thêm phần cứng OB cho chương trình ngắn:

Trên cửa sổ project chọn "Program blocks":

Double-click thêm khối mới “Add new block”, chọn sắp xếp khối “Organization block (OB)” và chọn “Hardware interrupt”.



Hình 3.134: Tạo khối OB cho lệnh ngắn

Có thể đổi tên OB, chọn ngôn ngữ lập trình (LAD hoặc FBD).

Có thể gọi FCS và FBS từ OB này.

Tất cả các phần cứng hiện trong OB ngắn xuất hiện trong cấu hình thiết bị HW

## b. Bắt đầu và hủy bỏ lệnh ngắn thời gian



**Hình 3.135: Lệnh STR\_DINT và CAN\_DINT**

Có thể bắt đầu và hủy bỏ ngắt thời gian với lệnh SRT\_DINT và CAN\_DINT. Mỗi ngắt được đặt một khoảng thời gian.

Nếu các sự kiện ngắt bị hủy trước khi ngắt thực thi thì chương trình ngắt không xảy ra.

SRT\_DINT bắt đầu ngắt thời gian. Với thời gian trì hoãn quy định trong tham số DTIME.

**Bảng 3.70: Thông số của lệnh SRT\_DINT**

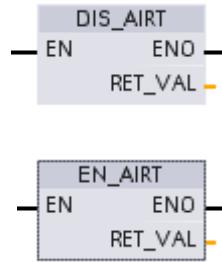
Tham số	Kiểu tham số	Kiểu dữ liệu	Mô tả
OB_NR	IN	Int	Khôi OB dùng để tạo thời gian trễ: chọn thời gian trễ từ OB sau khi tạo khôi OB.
DTIME	IN	Time	Thời gian trễ cho phép từ 1 đến 60000ms Có thể tạo thời gian trễ lớn hơn bằng cách cho bộ đến bên trong thời gian trễ trong OB.
SIGN	IN	Word	Không sử dụng trong S7-1200
RET_VAL	OUT	Int	Mã điều kiện thực thi

CAN\_DINT hủy bỏ một ngắt đã bắt đầu.

**Bảng 3.71: Thông số của lệnh CAL\_DINT**

Tham số	Kiểu tham số	Kiểu dữ liệu	Mô tả
OB_NR	IN	Int	Thời gian trễ định nghĩa trong OB.
RET_VAL	OUT	Int	Mã điều kiện thực thi

### c. Cho phép và không cho phép báo ngắt



Hình 3.136: Lệnh STR\_DINT và CAN\_DINT

Sử dụng các lệnh DIS\_AIRT và EN\_AIRT để vô hiệu hóa và cho phép báo ngắt

DIS\_AIRT trì hoãn việc xử lý các sự kiện ngắt. Có thể thực hiện DIS\_AIRT nhiều hơn một lần trong một OB.

EN\_AIRT cho phép xử lý các sự kiện gián đoạn trước đó đã vô hiệu hóa bằng lệnh DIS\_AIRT. Mỗi thao tác DIS\_AIRT phải bị hủy bỏ bởi một thực hiện EN\_AIRT.

Thông số RET\_VAL cho biết số lần xử lý ngắt đã bị vô hiệu.

Bảng 3.72: Thông số của lệnh CAL\_DINT

Tham số	Kiểu tham số	Kiểu dữ liệu	Mô tả
RET_VAL	OUT	Int	Số bộ trẽ bằng số DIS_AIRT

## CHƯƠNG 4: LẬP TRÌNH ỨNG DỤNG PLC S7-1200

### 4.1 GIỚI THIỆU

PLC S7-1200 là dòng PLC mới của hãng Siemens. Dòng sản phẩm này dần dần sẽ thay thế dòng sản phẩm trước đó là S7-200. Trên thế giới đã ứng dụng loại PLC này vào sản xuất. Tuy nhiên ở Việt Nam, dòng sản phẩm này còn khá mới lạ. Được sự hỗ trợ của Siemens về thiết bị, chúng tôi sử dụng PLC S7-1200 để làm hai ứng dụng là điều khiển giám sát tốc độ động cơ AC và giám sát điều khiển nhiệt độ.

Điều khiển giám sát tốc độ động cơ AC và giám sát điều khiển nhiệt độ là hai ứng dụng riêng biệt. Ở đây, nhóm chúng tôi đã làm trên một mô hình và lập trình chung trên một PLC. Các thiết bị sử dụng trong mạch gồm: PLC S7-1200 dùng để điều khiển, một cảm biến LM35D để đo nhiệt độ của môi trường, máy sấy dùng để làm thiết bị gia nhiệt, màn hình HMI để điều khiển và mô phỏng quá trình hoạt động của mô hình, một module analog SB1232 để cung cấp một ngõ ra analog cho S7-1200, một biến tần dùng để điều khiển động cơ, một motor AC và cuối cùng là encoder để đo tốc độ.



Hình 4.1: Mô hình thực tế của hệ thống

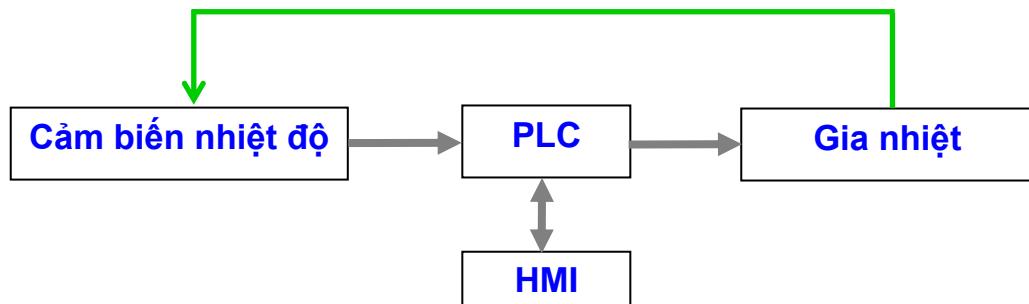
## 4.2 LẬP TRÌNH ỨNG DỤNG ĐIỀU KHIỂN NHIỆT ĐỘ

### 4.2.1 Giới Thiệu

Trong các thiết bị sử dụng nhiệt, việc điều khiển chính xác nhiệt độ nhất định là một việc khá súc phức tạp. Trong điều khiển hiện đại, có nhiều phương pháp điều khiển để đạt được độ chính xác cao. Một trong các phương pháp điều khiển chính xác nhất là điều khiển PID. PLC S7-1200 có hỗ trợ 16 vòng lặp PID cho phép tự động điều chỉnh. Mạch đo và điều khiển nhiệt độ được ứng dụng trong: nhà kho, lò nung, lò áp trứng, vườn ươm thông minh...

### 4.2.2 Sơ đồ khái quát

Hệ thống đo và điều khiển nhiệt độ được chia làm bốn khối như hình 4.2 bên dưới:



Hình 4.2: Sơ đồ hoạt động của bộ điều khiển

Giải thích:

- Tín hiệu điều khiển từ HMI được truyền xuống PLC qua dây cáp mạng RJ45 để điều khiển và giám sát nhiệt độ.
- Ngõ ra của cảm biến nhiệt độ là ngõ ra analog (điện thế). Giá trị điện thế sẽ thay đổi theo nhiệt độ. Giá trị này được đưa vào PLC qua ngõ vào analog của PLC và được chuyển đổi thành giá trị nhiệt độ. PLC sẽ so sánh nhiệt độ đo được với nhiệt độ đặt trước để điều khiển thiết bị gia nhiệt.

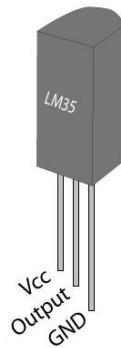
### 4.2.3 Kết nối phần cứng

#### a. Kết nối cảm biến nhiệt độ với PLC

Một vài thông số của cảm biến nhiệt độ LM35:

- + Dải nhiệt độ biến đổi:  $2^{\circ}\text{C}$  đến  $150^{\circ}\text{C}$ .

- + Tương ứng  $10\text{mV}/{}^{\circ}\text{C}$
- + Nhiệt độ ra thẳng thang đo Celcius nghĩa là ở  $25^{\circ}\text{C}$  điện áp ra là  $0.25\text{V}$
- + Đảm bảo độ chính xác  $0.5$  độ C tại nhiệt độ  $25$  độ C.
- + Sai số cực đại  $2^{\circ}\text{C}$  khi nhiệt độ lớn hơn  $100^{\circ}\text{C}$ .
- + Làm việc với nguồn nuôi  $4\text{V}$  đến  $20\text{V}$ .
- + Trở kháng ra thấp  $0.1$  ohm với tải  $1\text{mA..}$
- + Khả năng tự làm nóng thấp,  $0.08$  độ C trong không khí.



**Hình 4.3: Cảm biến nhiệt độ LM 35**

Cách mắc:

Ta nối chân Vcc với nguồn  $5\text{v}$  và chân GND với mass  $0\text{v}$ , chân OUTPUT nối với chân Analog input của PLC.

#### b. Kết nối thiết bị gia nhiệt với PLC

Ở đây ta sử dụng thiết bị gia nhiệt là máy sấy tóc, công suất là  $850\text{W}$ , điện áp là  $220\text{V}$ , tần số  $50\text{Hz}$ , nhiệt độ tối đa là  $60^{\circ}$ .



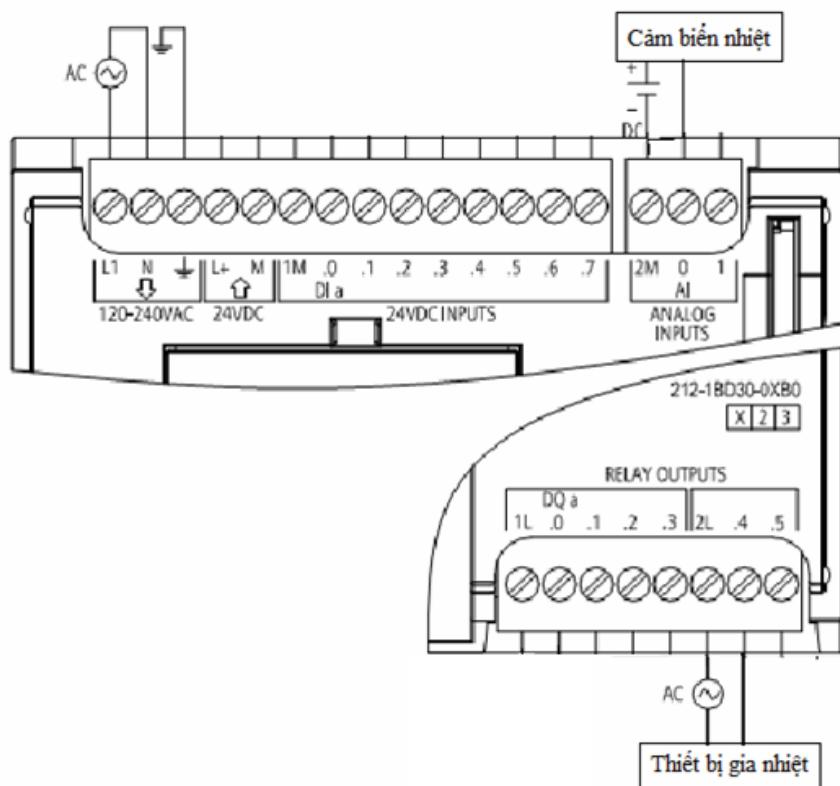
**Hình 4.4: Thiết bị gia nhiệt**

Máy sấy được nối với PLC qua ngõ ra Q0.4 để điều khiển đóng, mở nguồn cho máy sấy.

### c. Kết nối HMI với PLC

HMI nối với PLC qua cổng mạng RJ45.

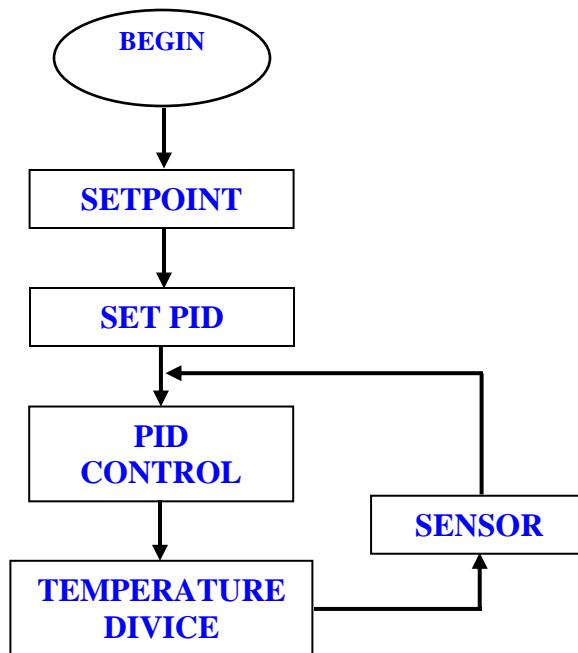
#### 4.2.4 Sơ đồ đấu dây



Hình 4.5: Sơ đồ đấu dây các thiết bị vào PLC

#### 4.2.5 Lưu đồ giải thuật và chương trình

##### a. Lưu đồ giải thuật:



Hình 4.6: Lưu đồ chương trình đo và điều khiển nhiệt độ

Chương trình:

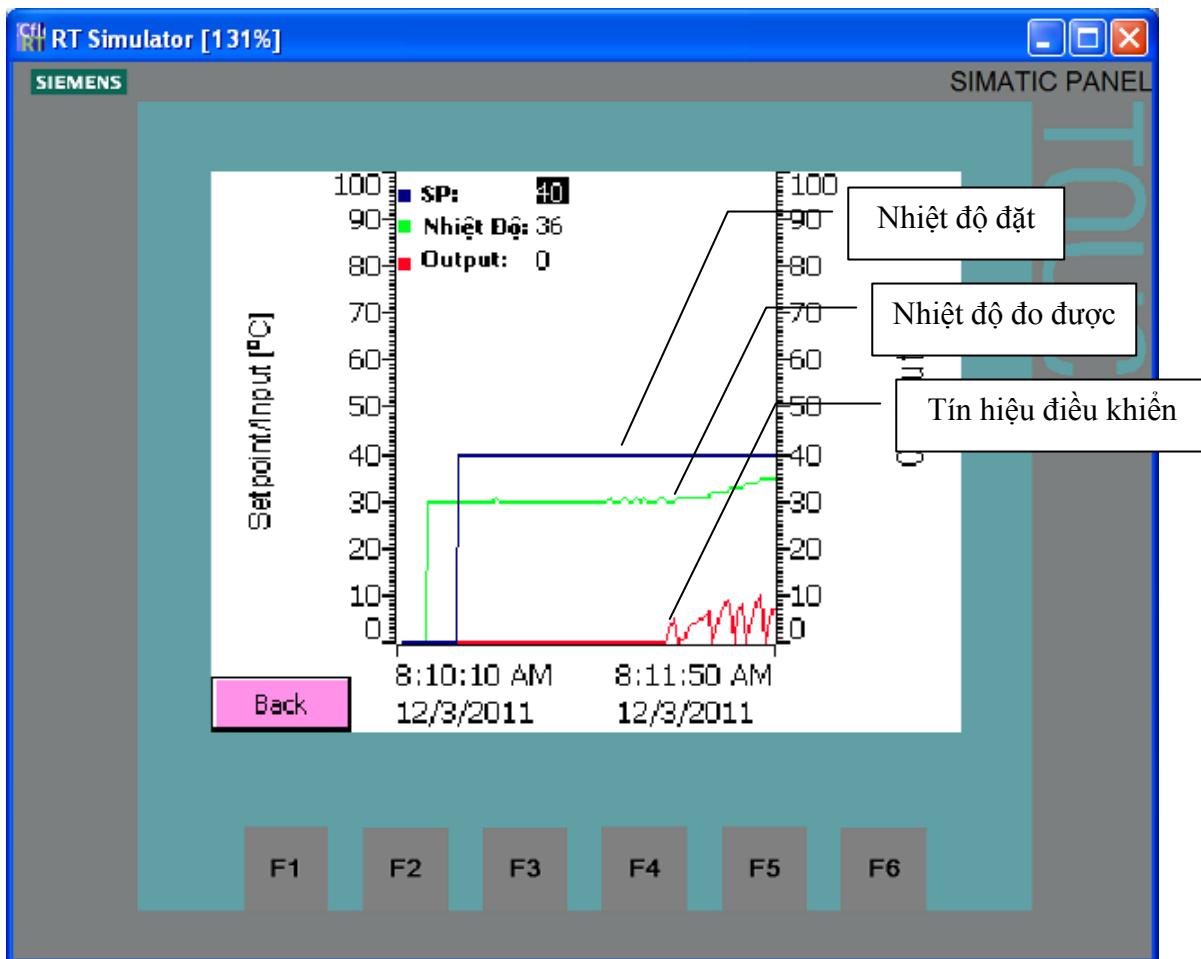
(Xem phần phụ lục)

#### 4.2.6 Kết quả

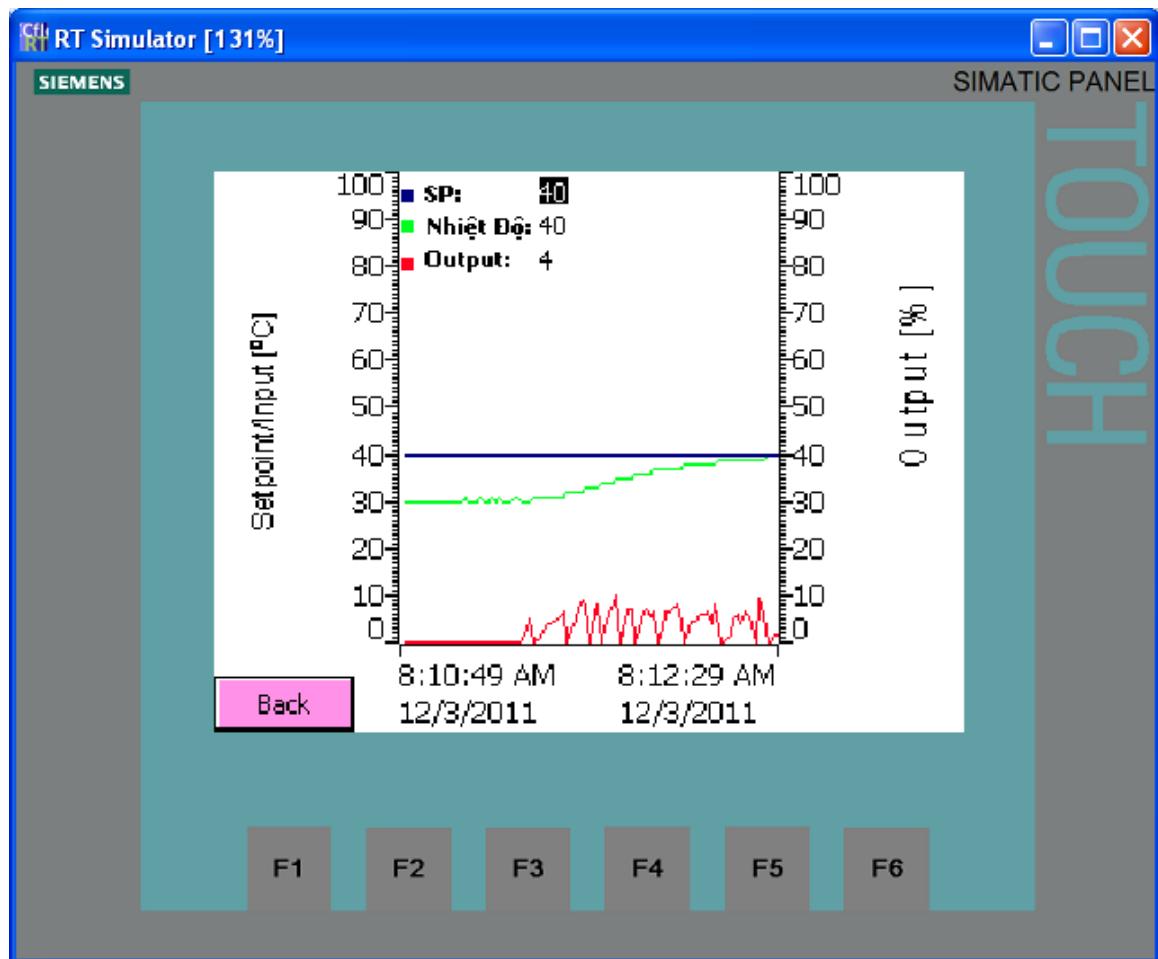
Hệ thống dễ giám sát và điều khiển qua giao diện HMI.

Điều khiển chính xác nhiệt độ đặt.

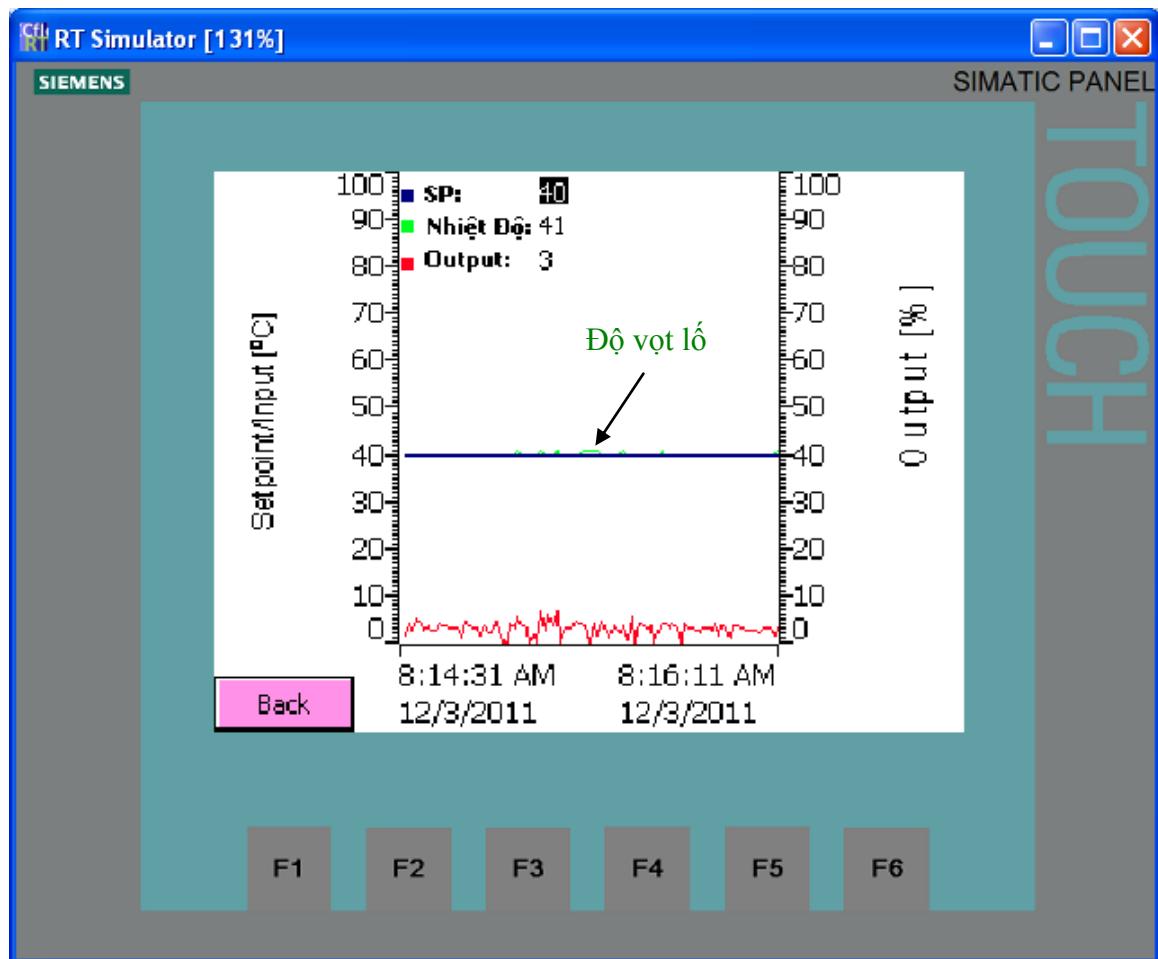
- Nhiệt độ đặt là 40°C



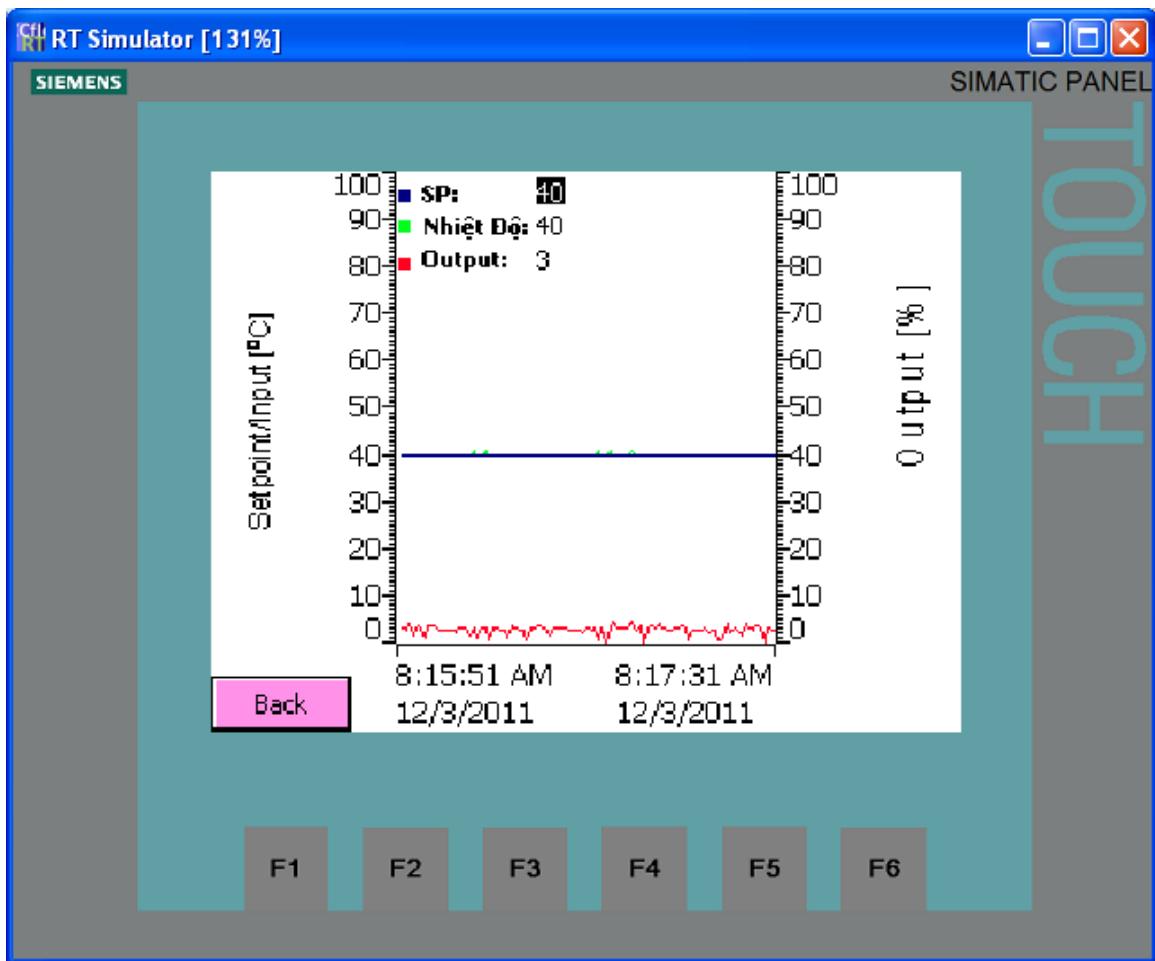
Hình 4.7: Đặt nhiệt độ điều khiển



Hình 4.8: Đáp ứng của bộ điều khiển nhiệt độ



Hình 4.9: Độ vọt lô



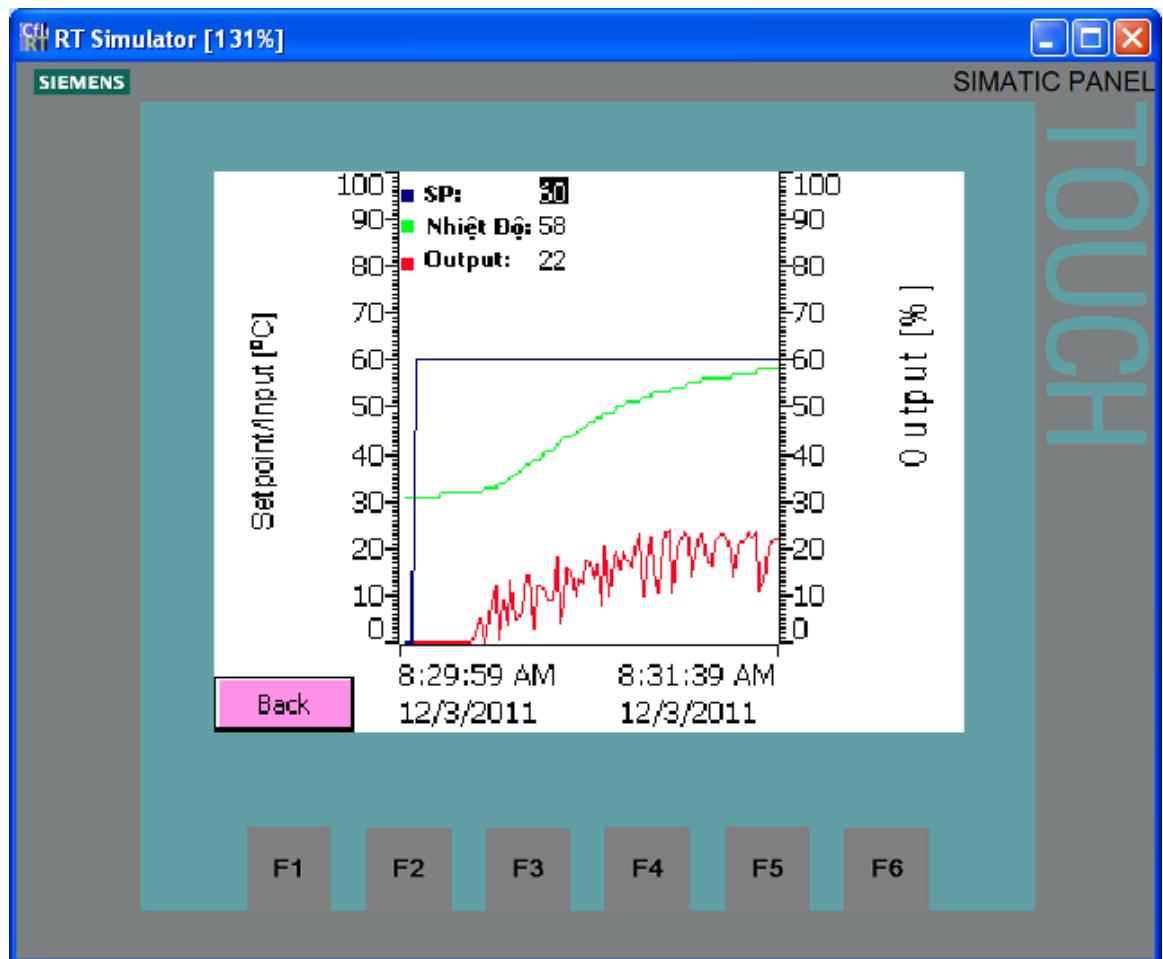
Hình 4.10: Đáp ứng ổn định

- Từ đồ thị ta thấy
  - + Độ vọt lô:

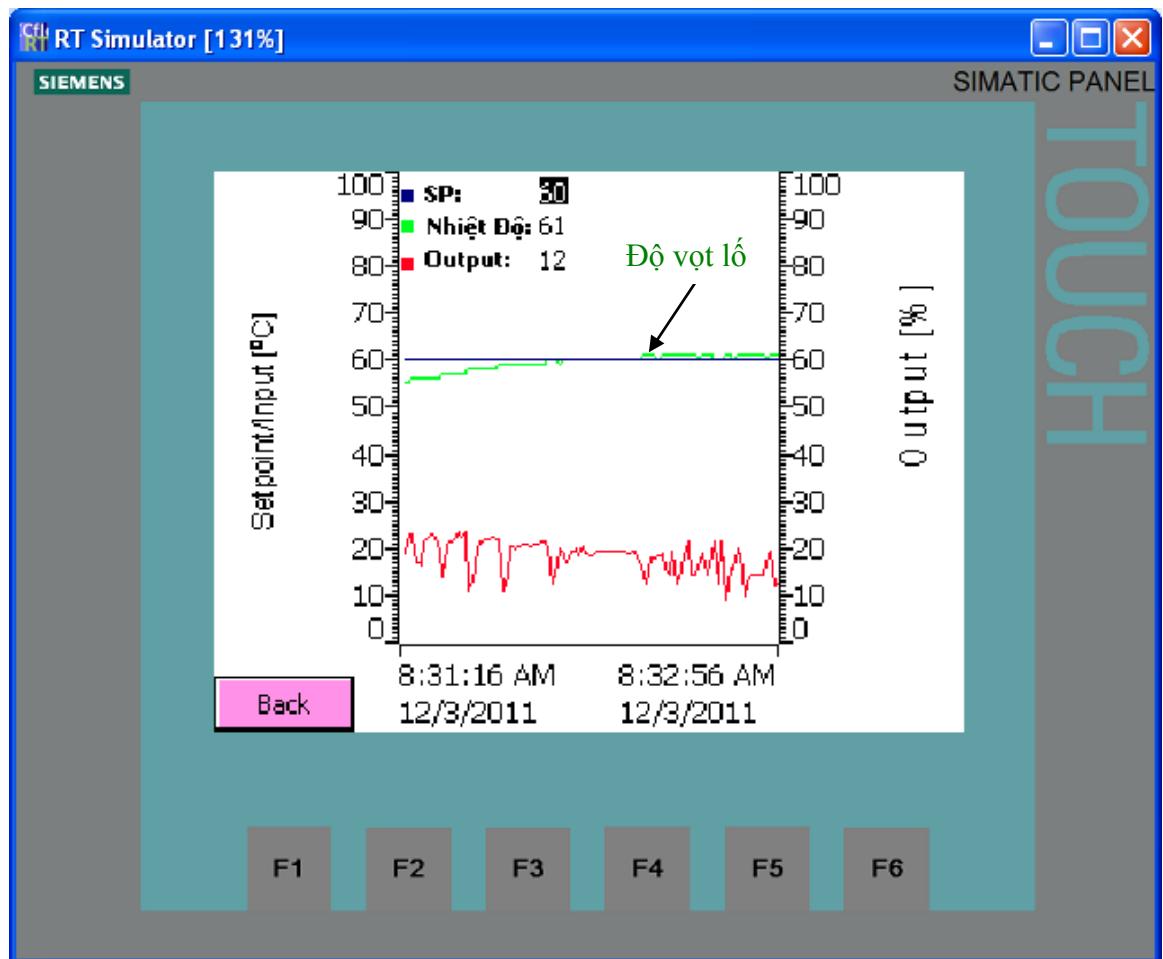
$$POT = \frac{Output_{max} - Setpoint}{Setpoint} \times 100 = \frac{41 - 40}{40} \times 100 = 2.5\%$$

- + Sai số xác lập gần bằng 0.
- + Thời gian tăng xấp xỉ 20 giây.

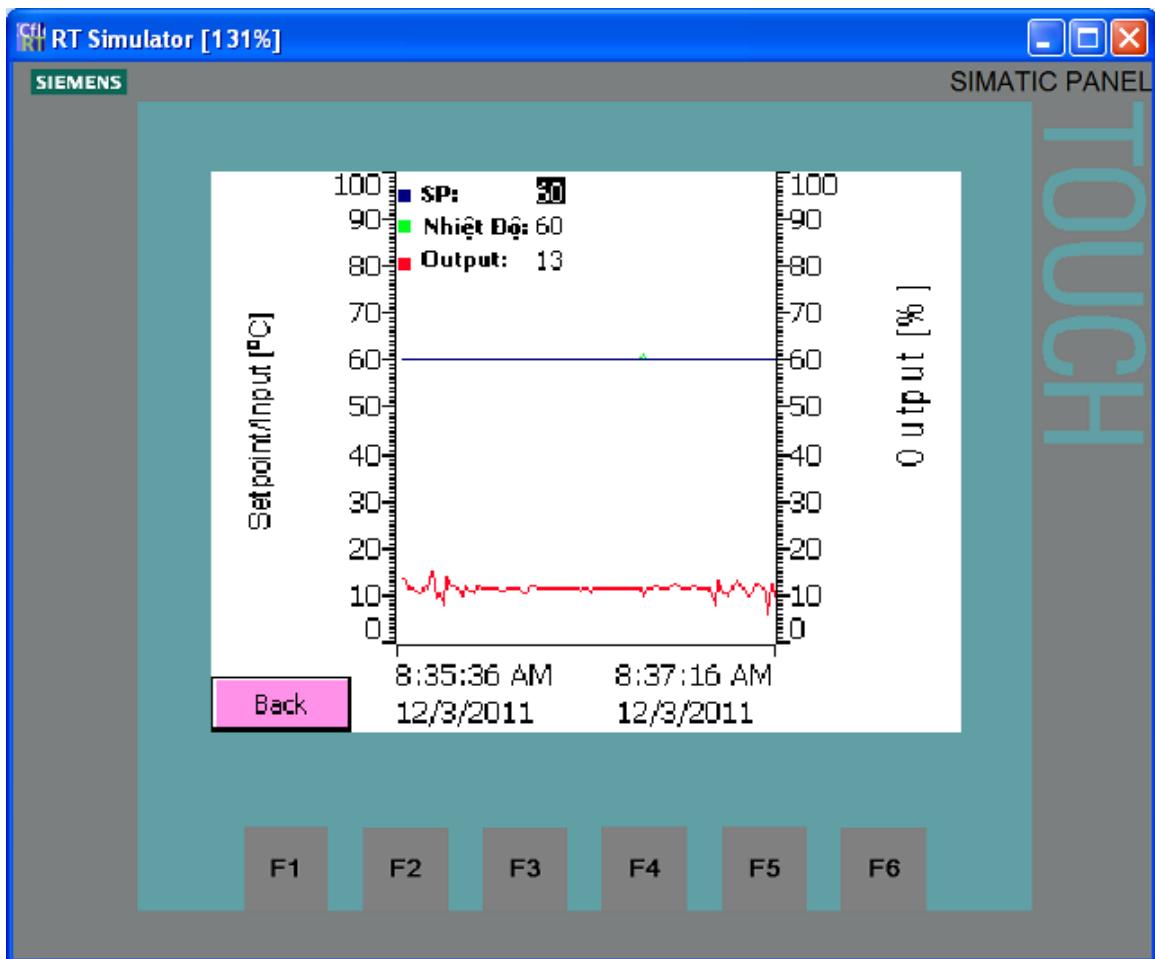
- Nhiệt độ đặt là 60°C



Hình 4.11: Đặt nhiệt độ điều khiển



Hình 4.12: Độ vọt lồ



Hình 4.13: Đáp ứng ổn định

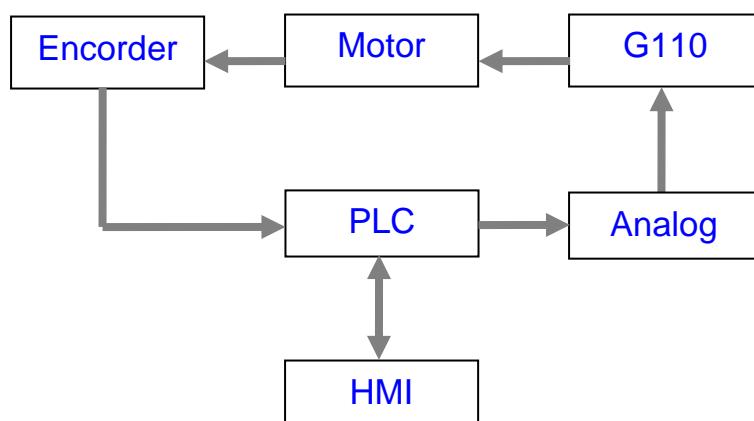
- Từ đồ thị ta thấy
    - + Độ vọt lồ:
- $$POT = \frac{Output_{max} - Setpoint}{Setpoint} \times 100 = \frac{61 - 60}{60} \times 100 = 1.67$$
- + Sai số xác lập gần bằng 0.
  - + Thời gian tăng xấp xỉ 20 giây.
- (Xem thêm trong phần phụ lục 3)

## 4.3 LẬP TRÌNH ỨNG DỤNG ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ AC

### 4.3.1 Giới thiệu

Ngoài điều khiển nhiệt độ, thì điều khiển động cơ là một trong những hình thức điều khiển nhiều trong thực tế. PLC S7-1200 cũng hỗ trợ rất tốt khả năng điều khiển tốc độ. Để điều khiển tốc độ động cơ ta dùng cũng dùng giải thuật PID. Các thiết bị được sử dụng: PLC S7-1200, module analog SB1232, biến tần Sinamic G110, động cơ AC, và encoder.

### 4.3.2 Sơ đồ khái



Hình 4.14: Nguyên lý hoạt động của bộ điều khiển tốc độ động cơ

Giải thích:

- Tín hiệu điều khiển từ HMI được truyền xuống PLC qua dây cáp mạng RJ45 để điều khiển và giám sát tốc độ motor AC.
- PLC thông qua module analog (ngõ ra áp 0 đến 10v) đưa điện thế vào ngõ vào analog của biến tần để điều khiển biến tần.
- Biến tần sẽ thay đổi tần số để điều khiển động cơ AC tùy vào điện thế ngõ vào analog.
- Tín hiệu từ encoder được đưa vào bộ đếm xung tốc độ cao (HSC) của PLC làm tín hiệu hồi tiếp cho vòng điều khiển PID.

### 4.3.3 Kết nối phần cứng

#### a. Kết nối module analog SB 1232 với PLC S7-1200

Module này được kết nối với PLC bằng cách ráp module vào phía trên PLC. Chỉ cần cắm module này vào khe cắm như hình bên dưới:

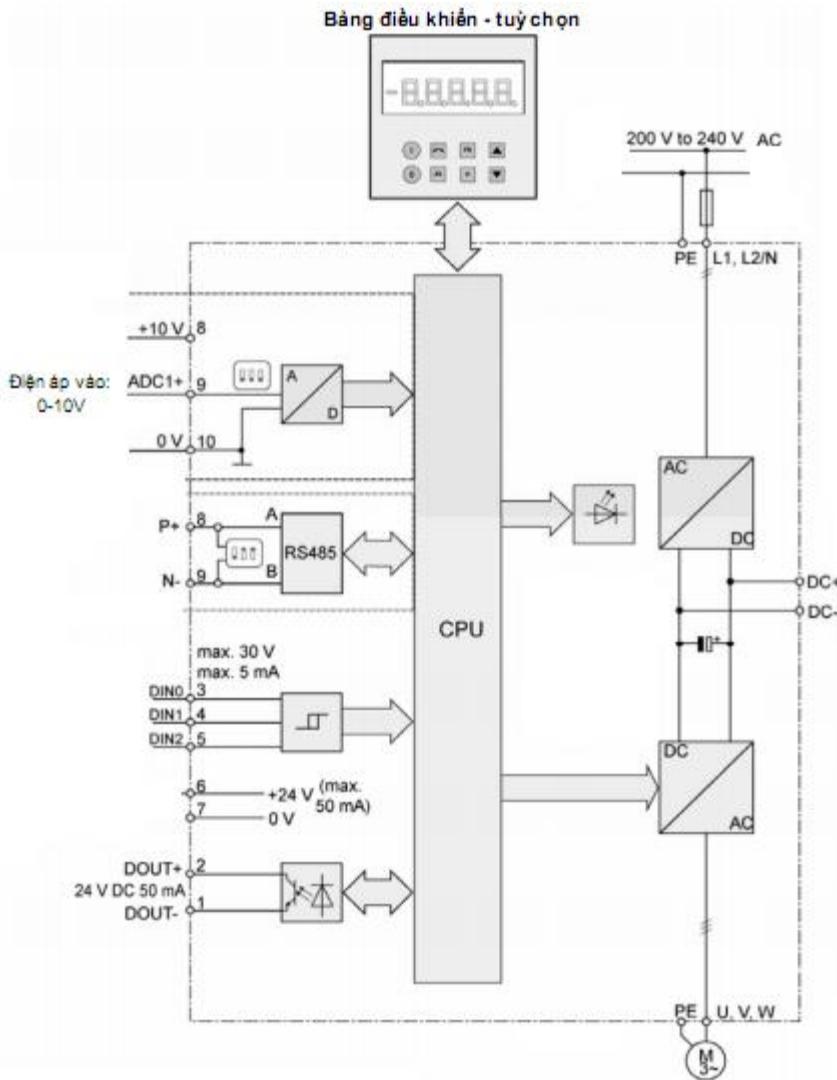


**Hình 4.15: Lắp module tương tự vào PLC**

### b. Biến tần G110

Sinamics G110 là biến tần cho phép thay đổi tần số như mong muốn để điều khiển động cơ AC 3 pha. Có nhiều loại model với công suất từ 120W đến 3KW . Chúng tôi sử dụng model có công suất 2.2kW

G110 sử dụng phương pháp biến điều độ rộng xung đặc biệt, với tần số xung có thể lựa chọn cho phép motor hoạt động thật yên tĩnh và nhẹ nhàng. Bên cạnh đó, với các tính năng bảo vệ toàn diện giúp bảo vệ an toàn cho chính biến tần G110 và motor kết nối khi sử dụng.



Hình 4.16: Sơ đồ nguyên lý của biến tần

Như vậy, để sử dụng biến tần ta nối như sau:

- + Cáp nguồn từ 200V đến 240V cho biến tần.
- + Chân số 3 dùng để điều khiển bật motor, chân 4 dùng để đảo chiều động cơ. Hai chân này sử dụng điện thế 24V DC để kích.
- + Chân số 10 nối mass và chân số 9 nối với ngõ ra của module analog SB 1232.
- + Các chân PE, U, V, W ta nối với động cơ AC.

Để biến tần G110 hoạt động ở chế độ điều khiển Analog ta cần cài đặt các thông số như sau:

- + Cấp độ truy cập các thông số ở chế độ mở rộng: P0003 = 2
- + Cài đặt tần số mặc định chuẩn Châu Âu: P0100 = 50Hz.
- + Điện áp định mức của động cơ là 230V: P0304 = 230
- + Chọn nguồn lệnh là dạng đầu nối: P0700 = 2.
- + Lựa chọn điểm đặt tần số là tương tự: P1000 = 2.
- + Đặt tần số nhỏ nhất: P1080 = 0.0Hz.
- + Đặt tần số lớn nhất: P1082 = 0.0Hz.

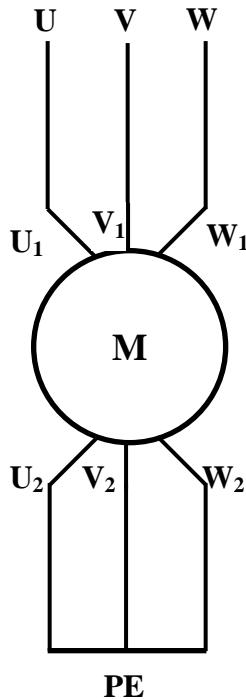
**c. Động cơ ba pha:**

- + Động cơ ba pha được sử dụng là motor AC ba pha.
- + Điện áp nguồn 220v
- + 1390 vòng /phút
- + Công suất 2.2 KW



**Hình 4.17: Motor AC ba pha.**

Do sử dụng loại motor có công suất nhỏ 2.2 KW nên có thể đấu theo kiểu hình sao (đối với động cơ dưới 5 KW. Cách nối motor với biến tần như hình bên dưới:



**Hình 4.18: Sơ đồ đấu dây motor AC ba pha với biến tần.**

#### d. Encoder



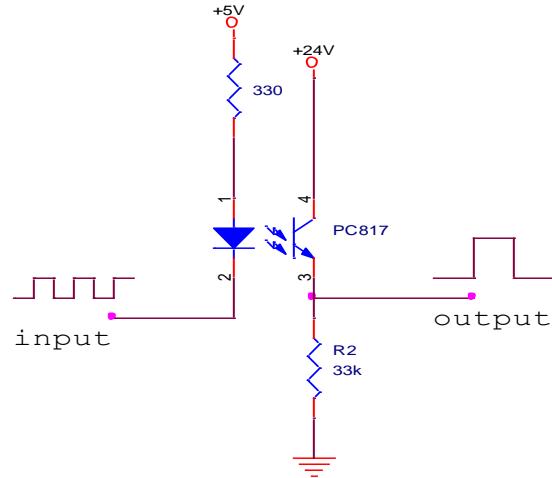
**Hình 4.19: Hình ảnh encoder thực.**

Encoder được sử dụng ở đây là loại encoder quang tương đối có độ phân giải 100 xung trên vòng quay.

Để số vòng quay của encoder bằng số vòng quay của motor, ta dùng các bánh răng đồng trực.

Do đầu vào của PLC là 24V, mà điện áp ngõ ra của Encoder là 5V, nên ta phải sử dụng mạch cách ly 2 điện áp (xem hình 4.20). Ở đây ta sử dụng Opto để cách ly.

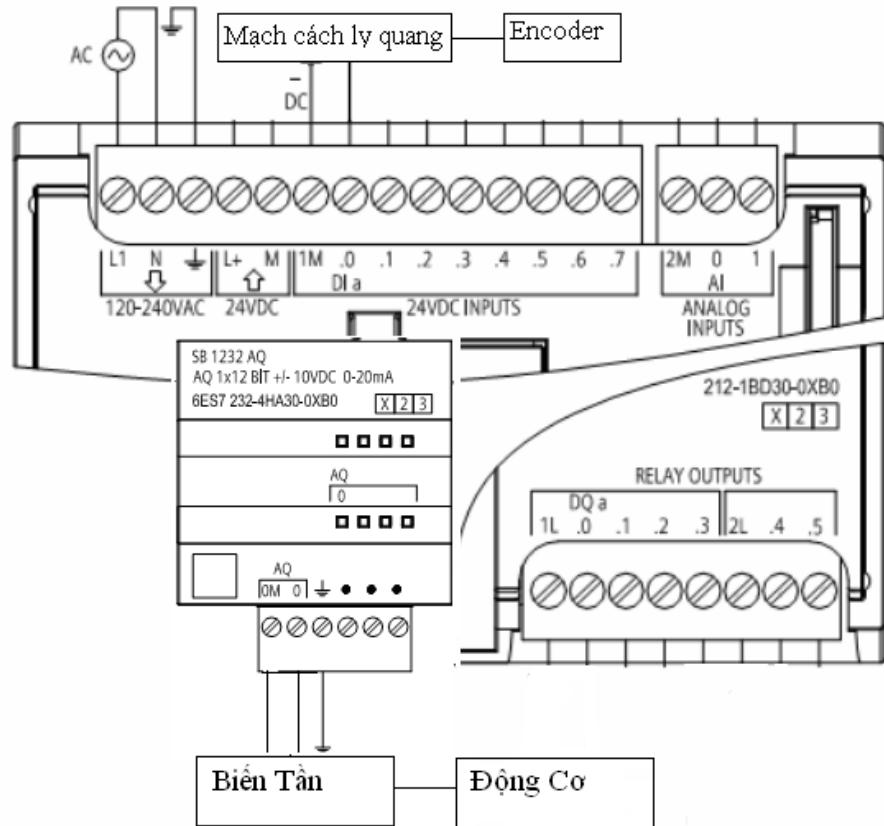
Sơ đồ mạch:



Hình 4.20: Sơ đồ mạch cách ly hai điện thế

#### 4.3.4 Mô hình phần cứng

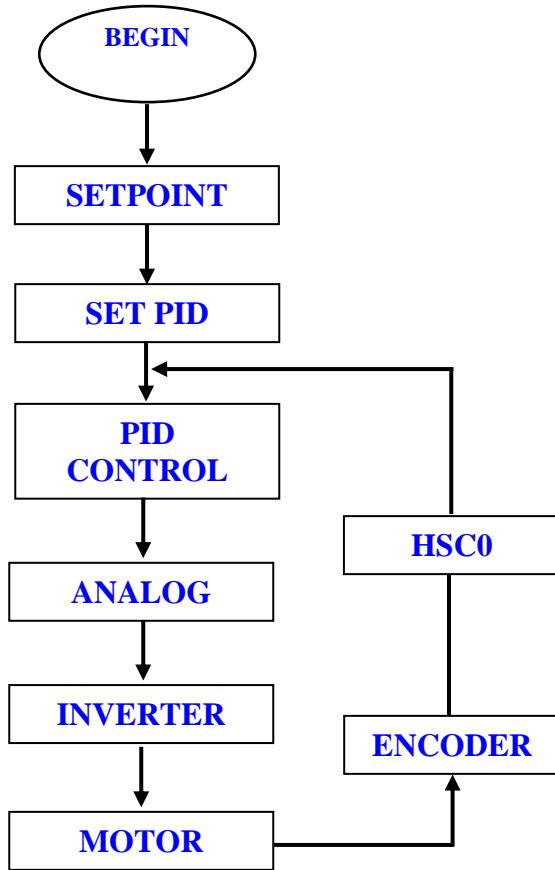
##### a. Sơ đồ đấu dây các thiết bị vào PLC



Hình 4.21: Sơ đồ mạch cách ly hai điện thế

#### 4.3.5 Lưu đồ giải thuật và chương trình

a. Lưu đồ giải thuật:



Hình 4.22: Sơ đồ mạch cách ly hai điện thế

b. Chương trình:

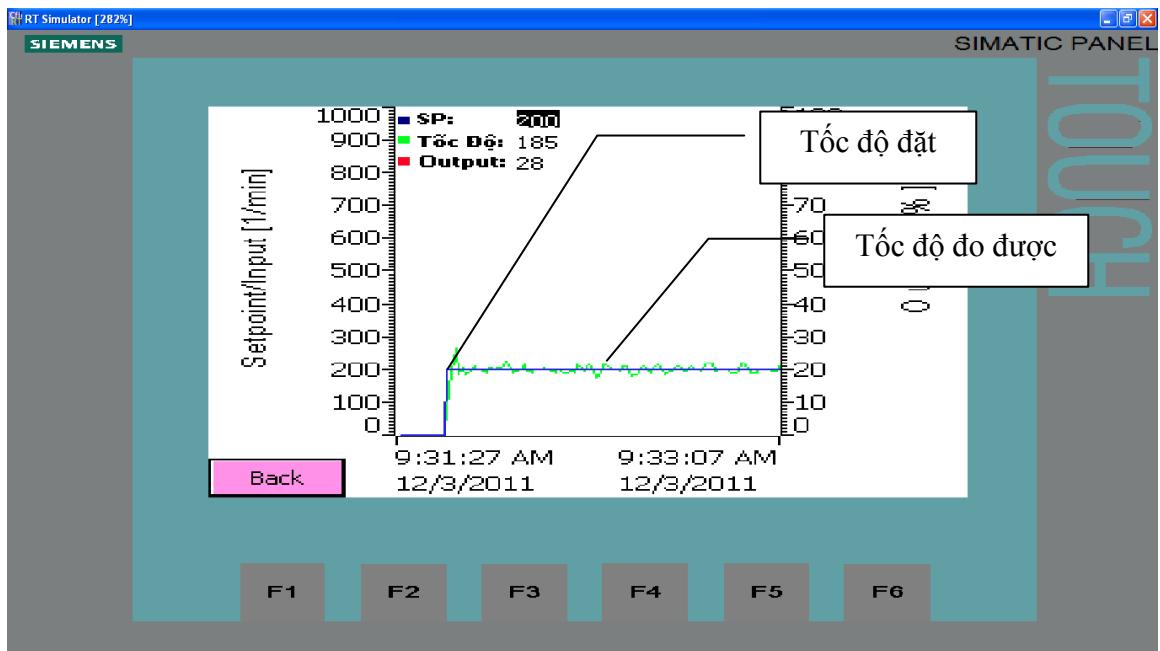
(Xem phần phụ lục)

#### 4.3.6 Kết quả

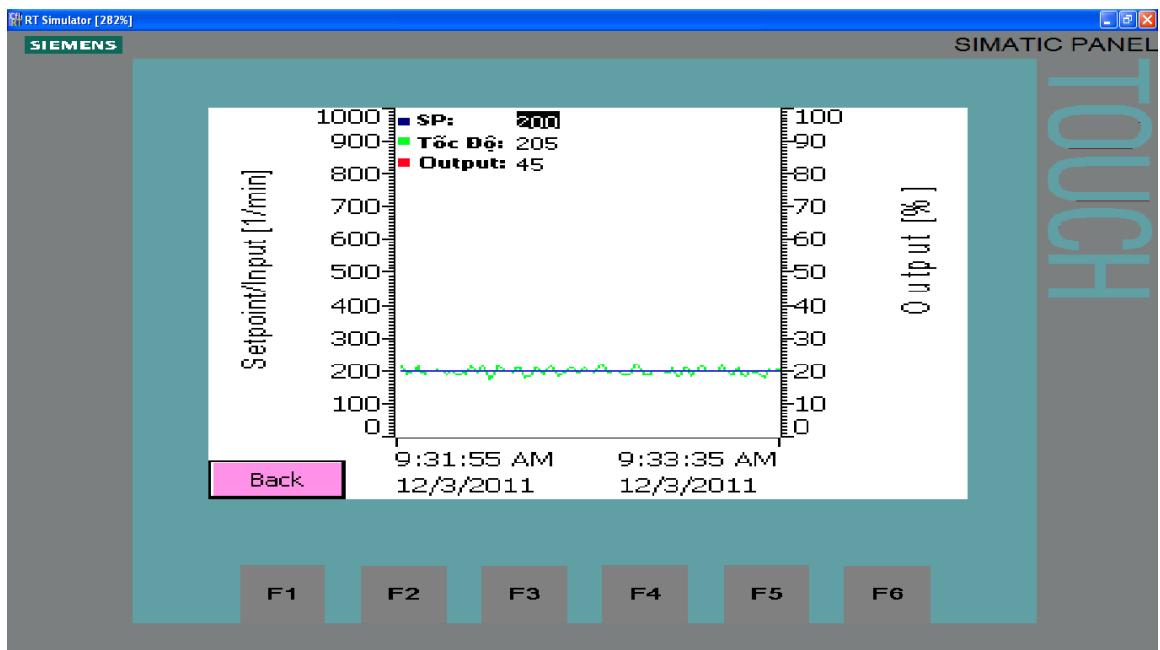
Hệ thống dễ giám sát và điều khiển qua giao diện HMI.

Điều khiển chính xác tốc độ đặt

- Tốc độ đặt 200 vòng/phút:



Hình 4.23: Đặt tốc độ điều khiển



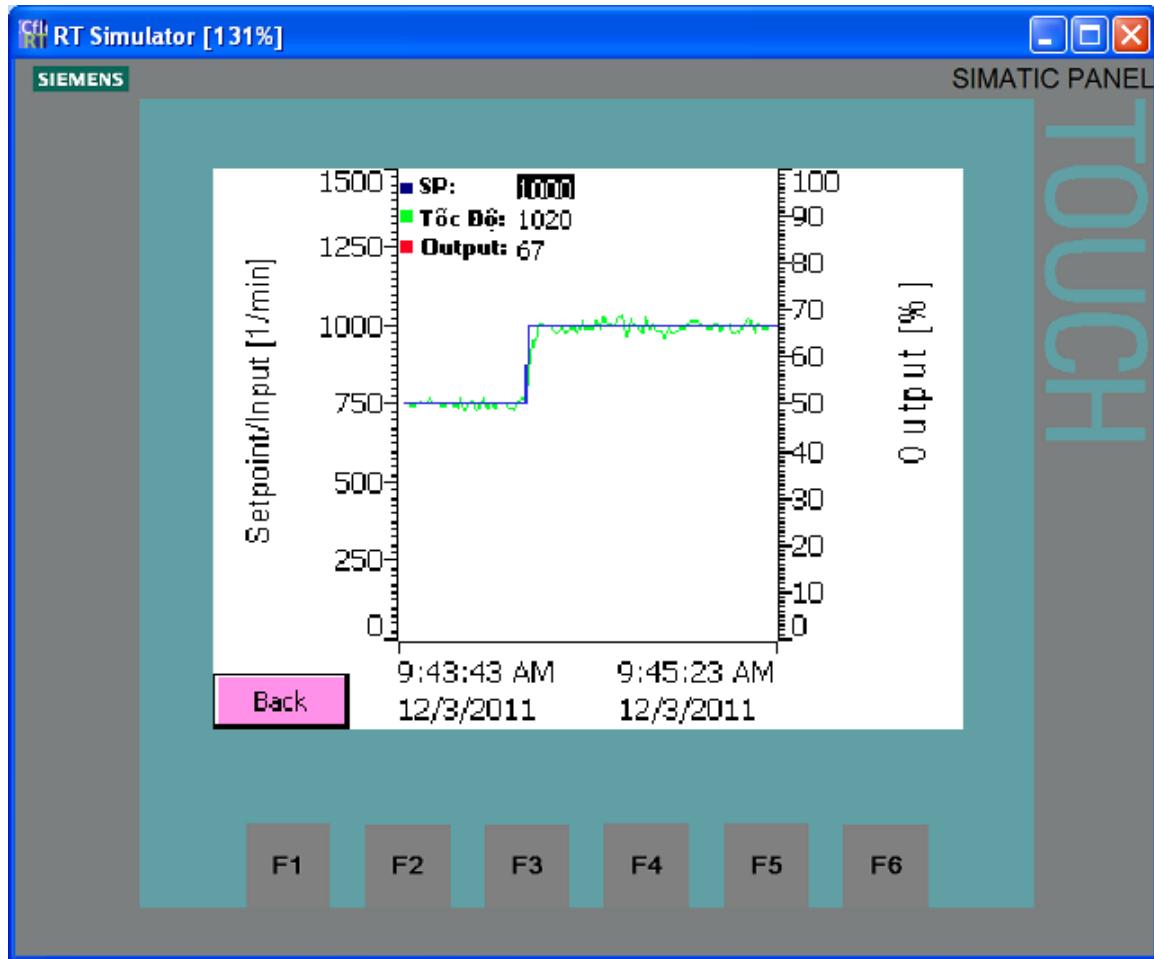
Hình 4.24: Tốc độ đặt điều khiển

- Từ đồ thị ta thấy
  - + Độ vọt lô:

$$POT = \frac{Output_{max} - Setpoint}{Setpoint} \times 100 = \frac{222 - 200}{200} \cdot 100 = 11\%$$

+ Sai số xác lập khoảng 10 vòng phút:

- + Thời gian tăng xấp xỉ 1 giây.
- Tốc độ đặt 1000 vòng/phút:



Hình 4.25: Đáp ứng của bộ điều khiển

- Từ đồ thị ta thấy
    - + Độ vọt lô:
- $$POT = \frac{Output_{max} - Setpoint}{Setpoint} \times 100 = \frac{1050 - 1000}{1000} \times 100 = 5\%$$
- + Sai số xác lập khoảng 10 vòng phút:
  - + Thời gian tăng xấp xỉ 1 giây.
- (Xem thêm trong phần phụ lục 3)

## KẾT LUẬN VÀ ĐỀ NGHỊ

### I. Kết quả

Tìm hiểu được PLC S7-1200 và các module mở rộng.

Khảo sát và cho ví dụ lồng ghép vào các lệnh.

Lập trình ứng dụng trên PLC S7-1200, HMI KTP600, module mở rộng. Làm mô hình ứng dụng thực tế với các đặc điểm:

- Chương trình điều khiển hoạt động tốt.
- Giao diện HMI thân thiện với người dùng.
- Nhiệt độ đo và điều khiển dễ dàng và chính xác.
- Điều khiển tốc độ động cơ đáp ứng nhanh và đáp ứng được yêu cầu.
- Nhiệt độ có thể điều khiển được: từ nhiệt độ môi trường đến 80°C.
- Tốc độ động cơ điều khiển được từ 0 đến 1390 vòng/phút.

### II. Hạn chế

Dù đã hoàn thành những mục tiêu đề ra, nhưng do kiến thức và thời gian hạn chế nên ứng dụng còn có nhiều khuyết điểm:

- Phần cứng hệ thống với các thiết bị đặt quá gần nhau, đồng thời không thể nối mass thiết bị nên ngõ vào analog bị nhiễu khi điều khiển động cơ. Điều này dẫn đến giá trị nhiệt độ đo được sẽ bị sai số nếu chạy song song với bộ điều khiển động cơ.
- Thời gian đáp ứng của bộ điều khiển nhiệt độ khá chậm (khoảng 20 đến 30 giây).
- Sai số của bộ điều khiển tốc độ động cơ còn lớn (khoảng 5 đến 10 vòng/phút)

### III. Hướng phát triển

Để hệ thống có thể hoàn thiện và hoạt động tốt hơn cần phải thực hiện một số vấn đề sau:

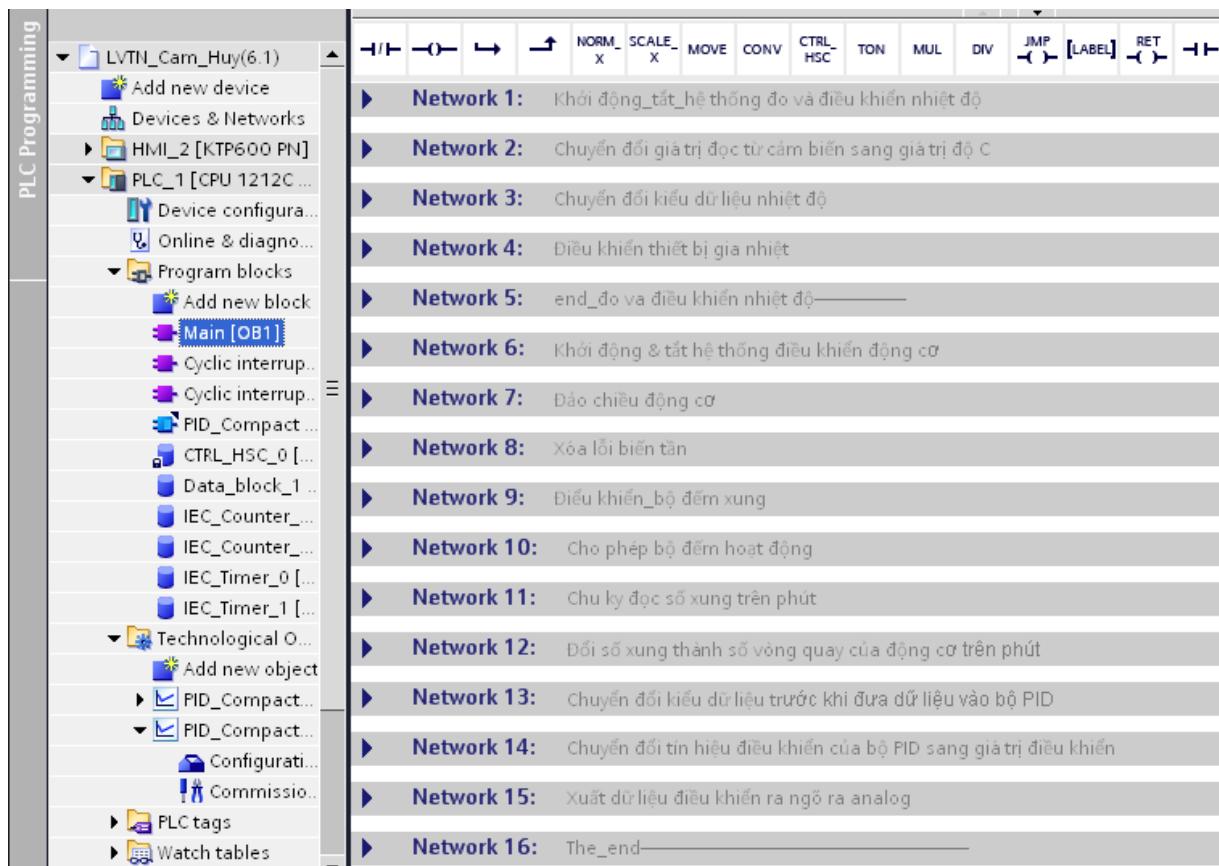
- Thêm thiết bị chống nhiễu, nối mass thiết hoặc lắp đặt các thiết bị với khoảng cách xa hơn.
- Sử dụng các thiết bị cảm biến có độ nhạy cao và thiết bị tạo nhiệt công suất lớn để đáp ứng điều khiển nhiệt độ nhanh hơn.
- Sử dụng encoder có độ phân giải cao (khoảng 1000 xung/phút) để giảm sai số
- Sử dụng thêm thiết bị làm lạnh để có thể giảm nhiệt độ xuống nhỏ hơn nhiệt độ môi trường.
- Sử dụng thêm module CSM1277 để kết nối và lưu trữ dữ liệu trên máy tính.
- Điều khiển hệ thống từ xa bằng Web Navigator.

## PHỤ LỤC

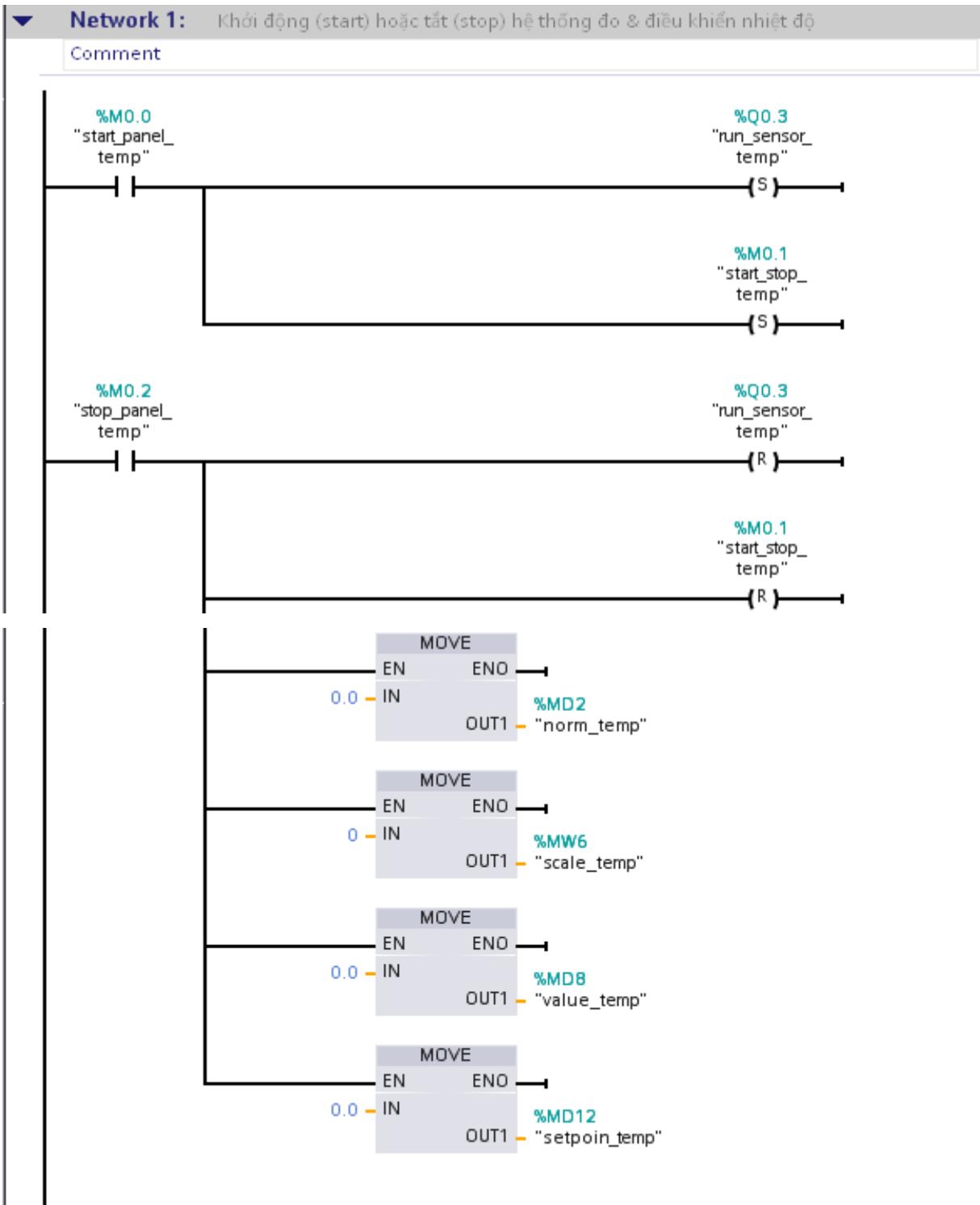
### PHỤ LỤC 1 CHƯƠNG TRÌNH ĐIỀU KHIỂN TRÊN PLC

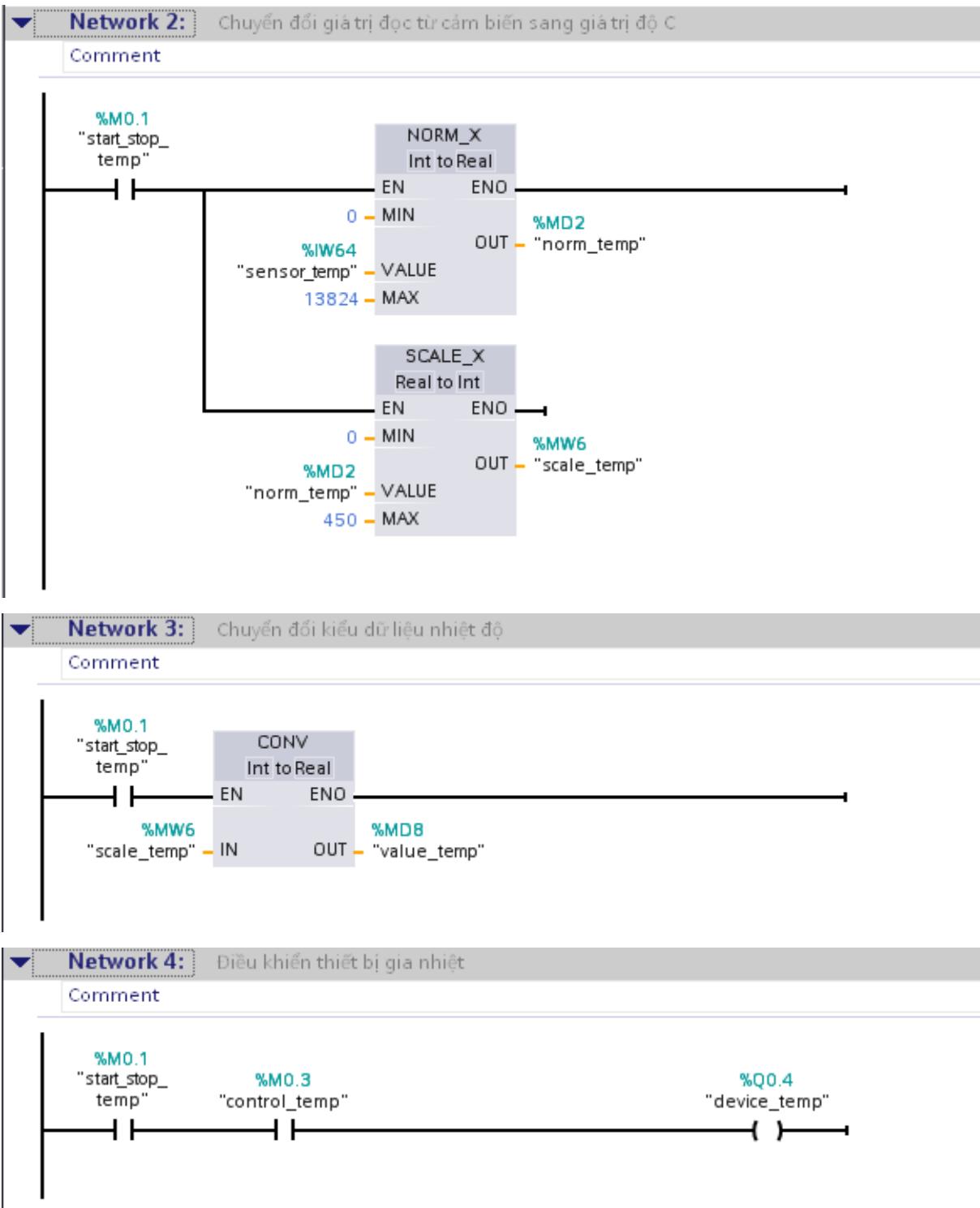
Lợi dụng đặc tính hỗ trợ mạnh về tập lệnh (hỗ trợ nhiều vòng lặp ngắt, PID...) nên nhóm đã viết chương trình điều khiển và giám sát trên cùng một PLC. Điều này giúp tiết kiệm được chi phí và dễ dàng điều khiển và giám sát hơn.

Sau đây là chương trình:

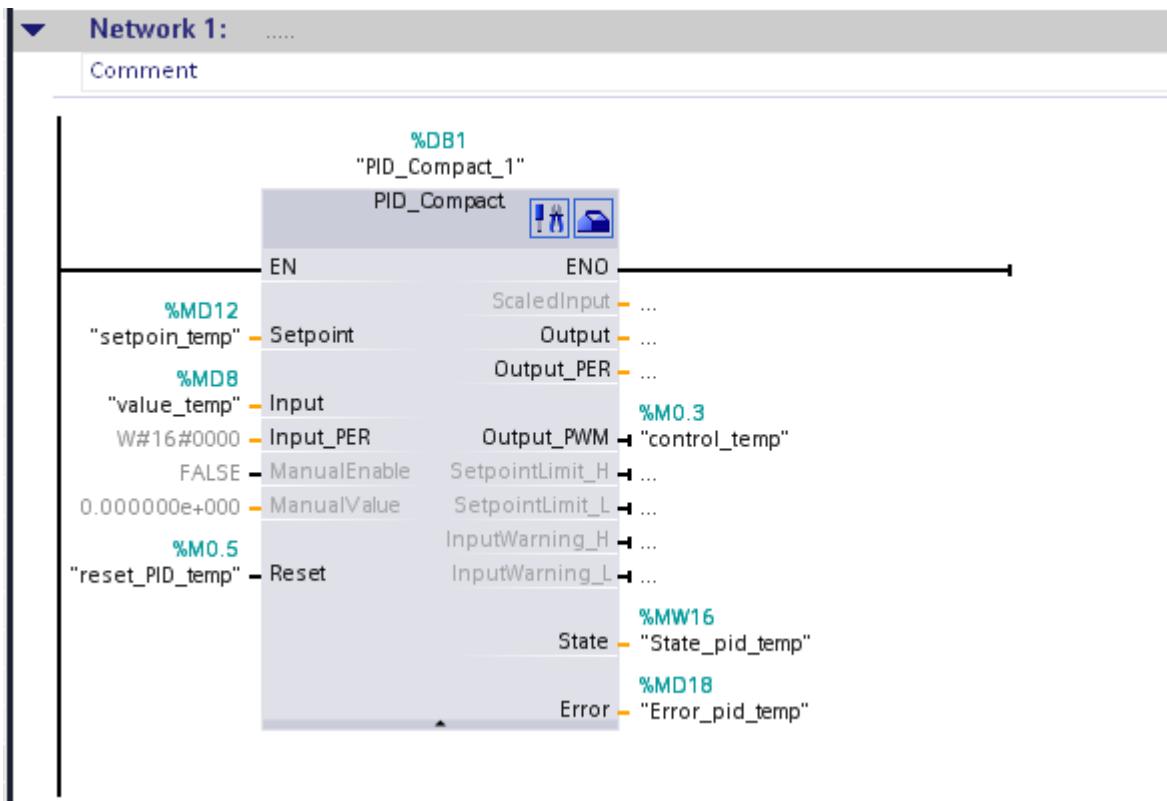


- Chương trình chính giám sát và điều khiển nhiệt độ:

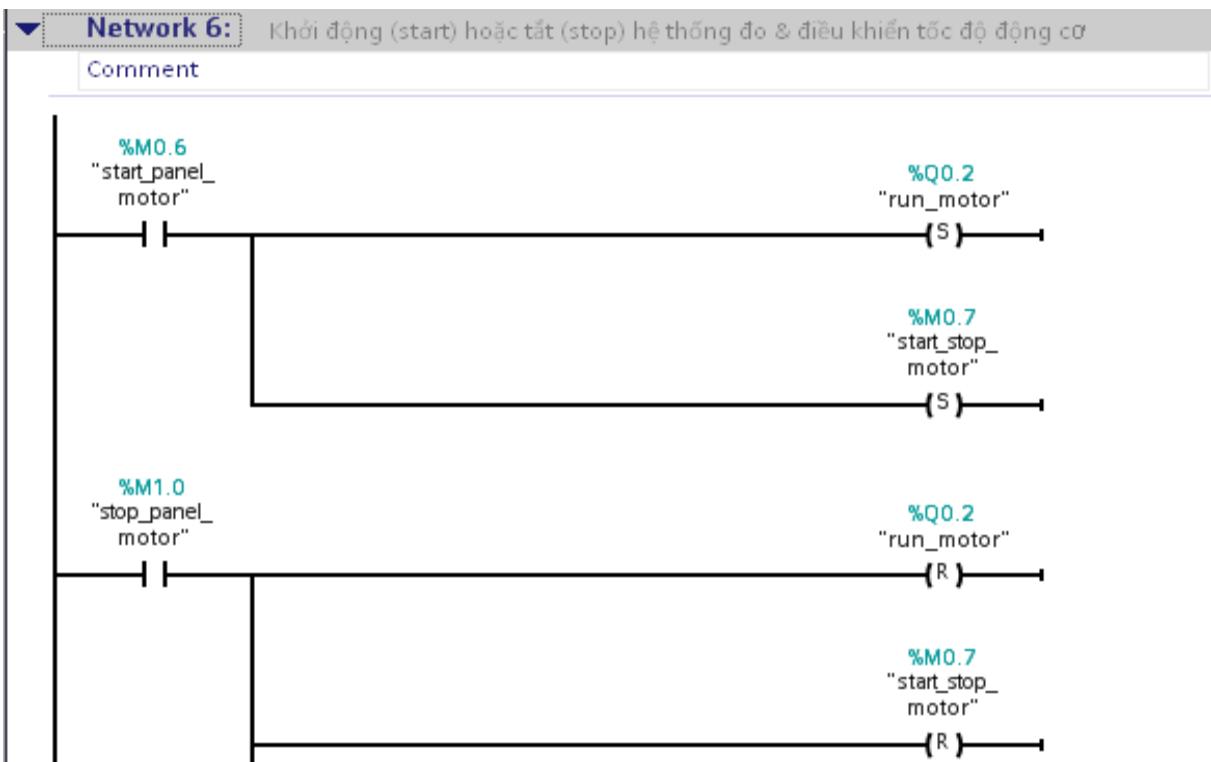


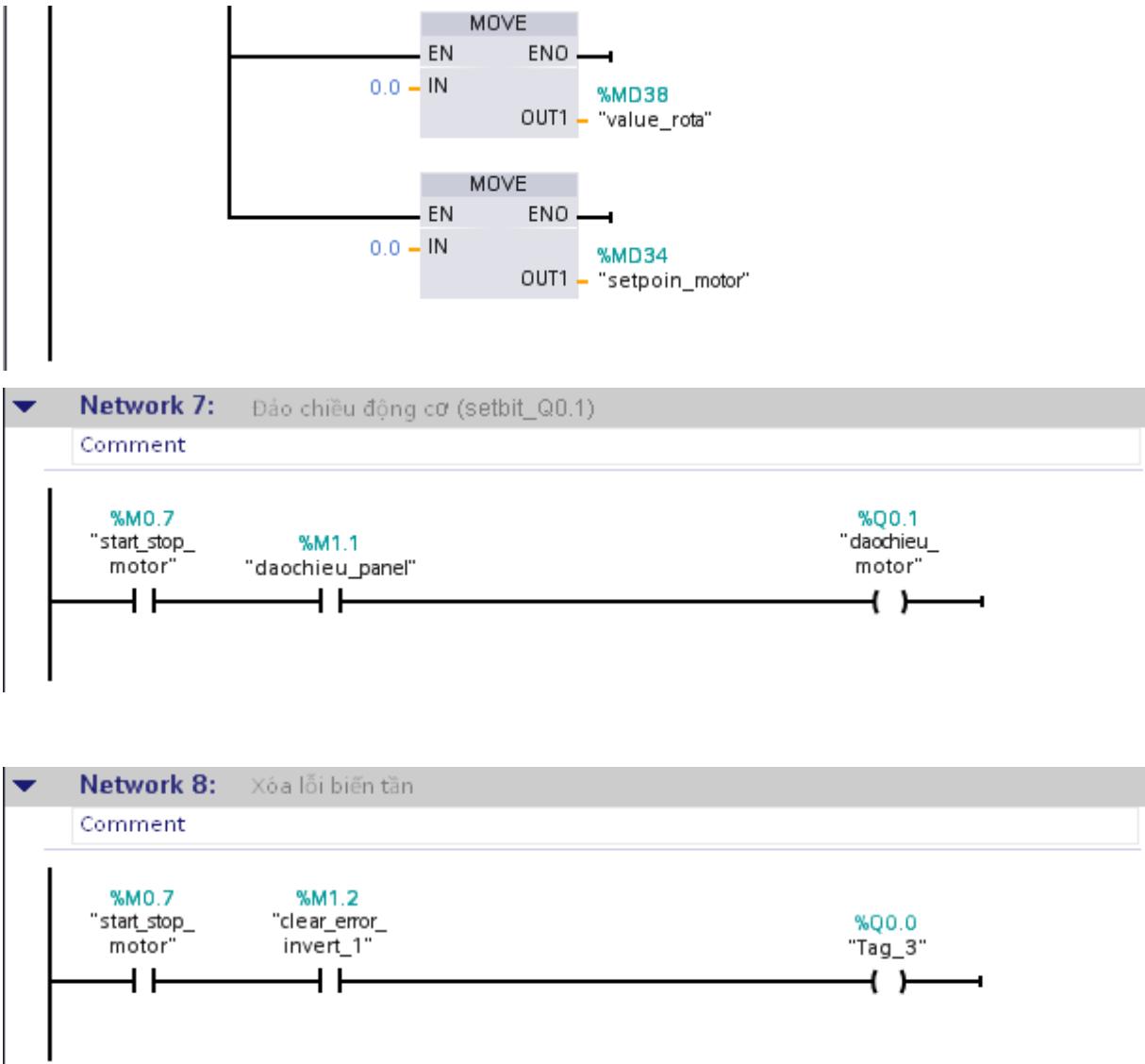


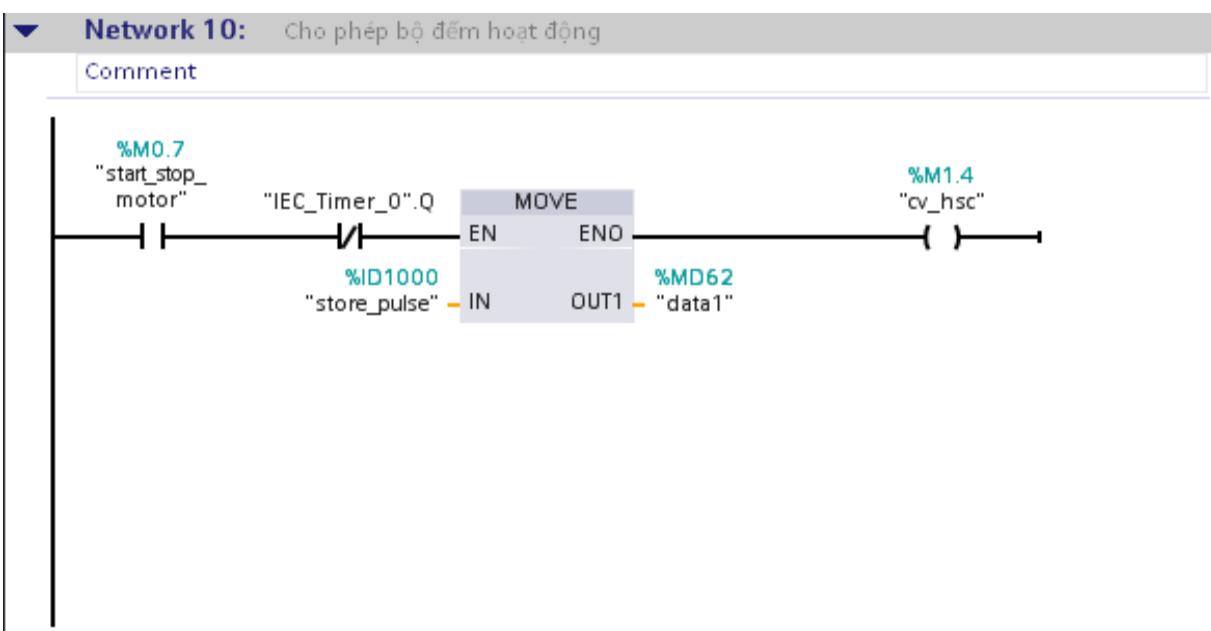
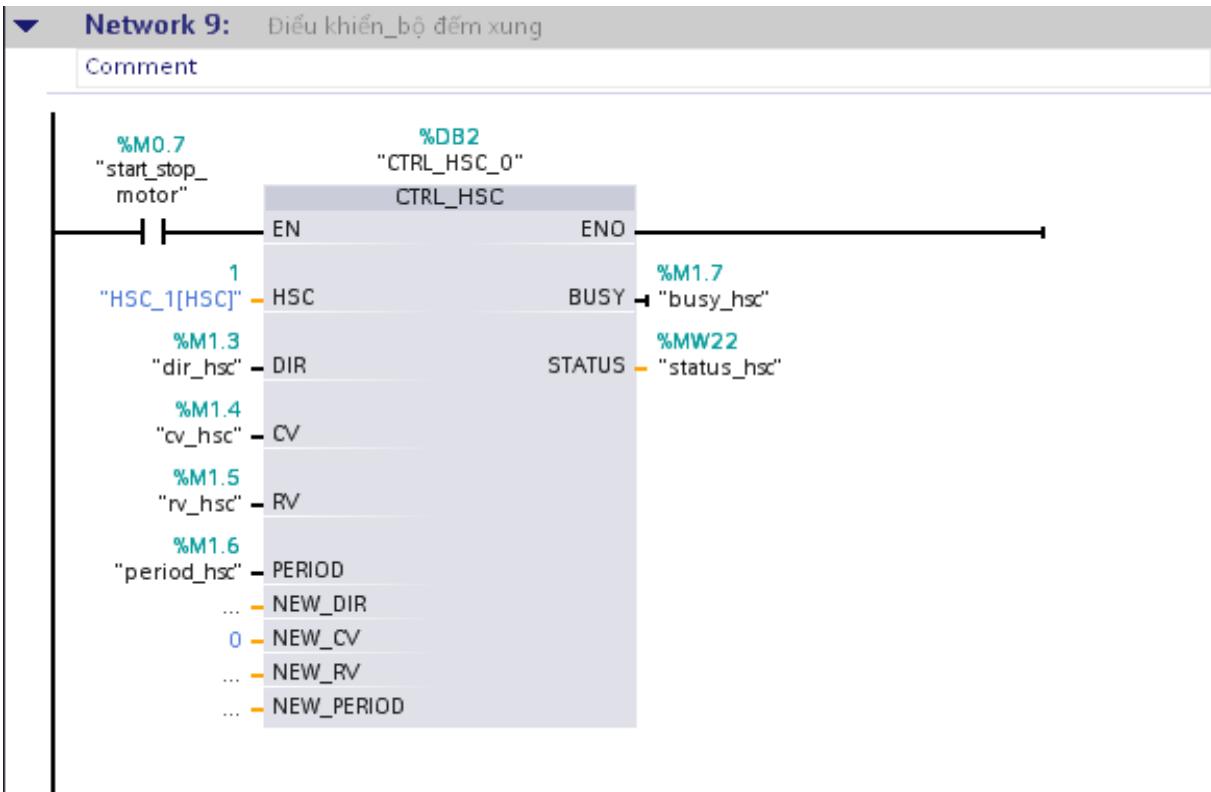
### PID điều khiển nhiệt độ:

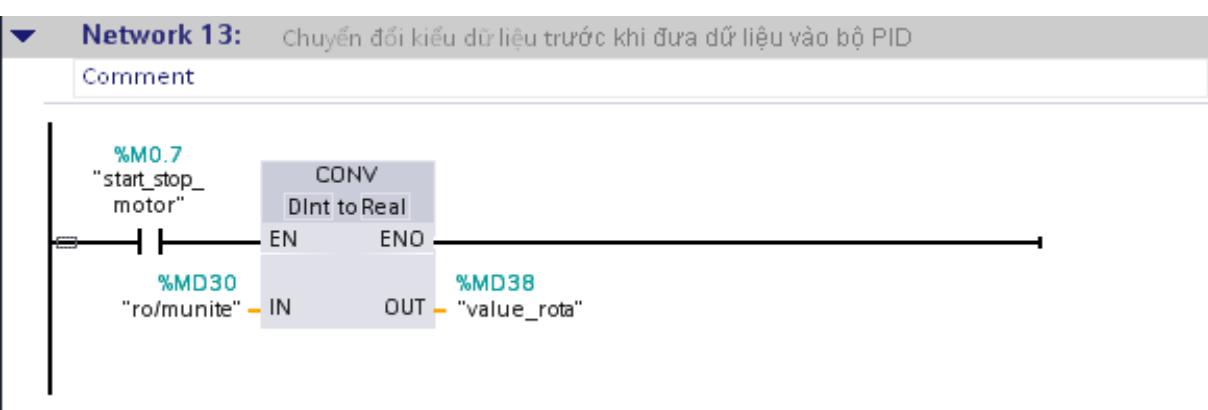
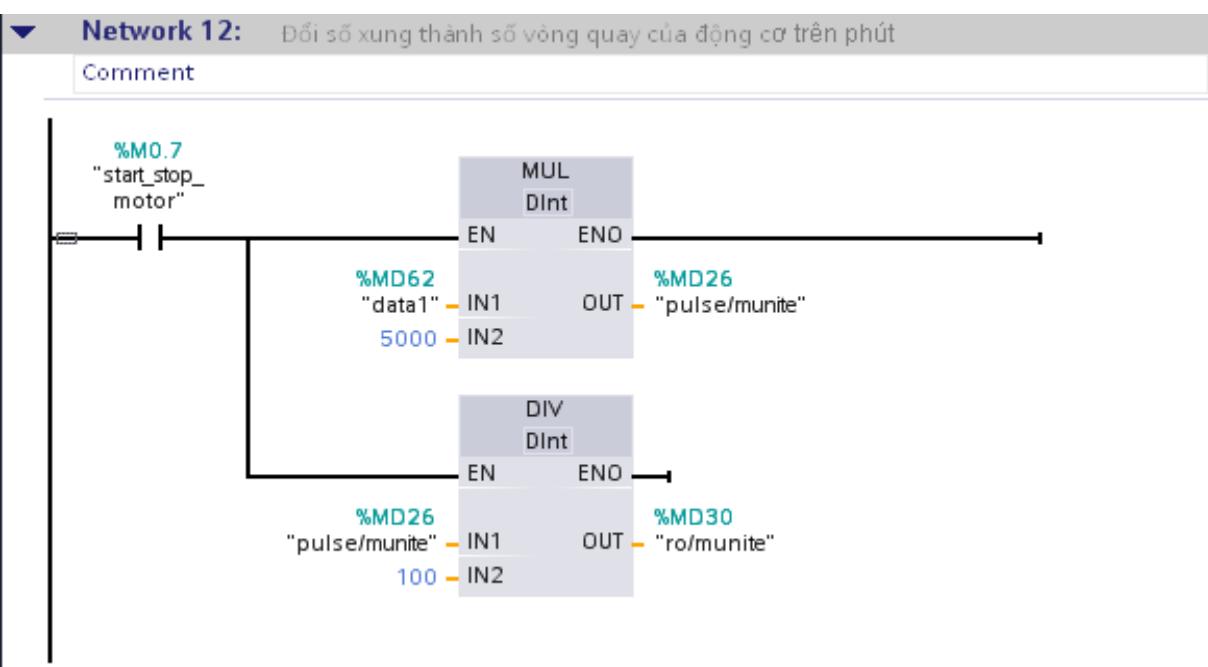
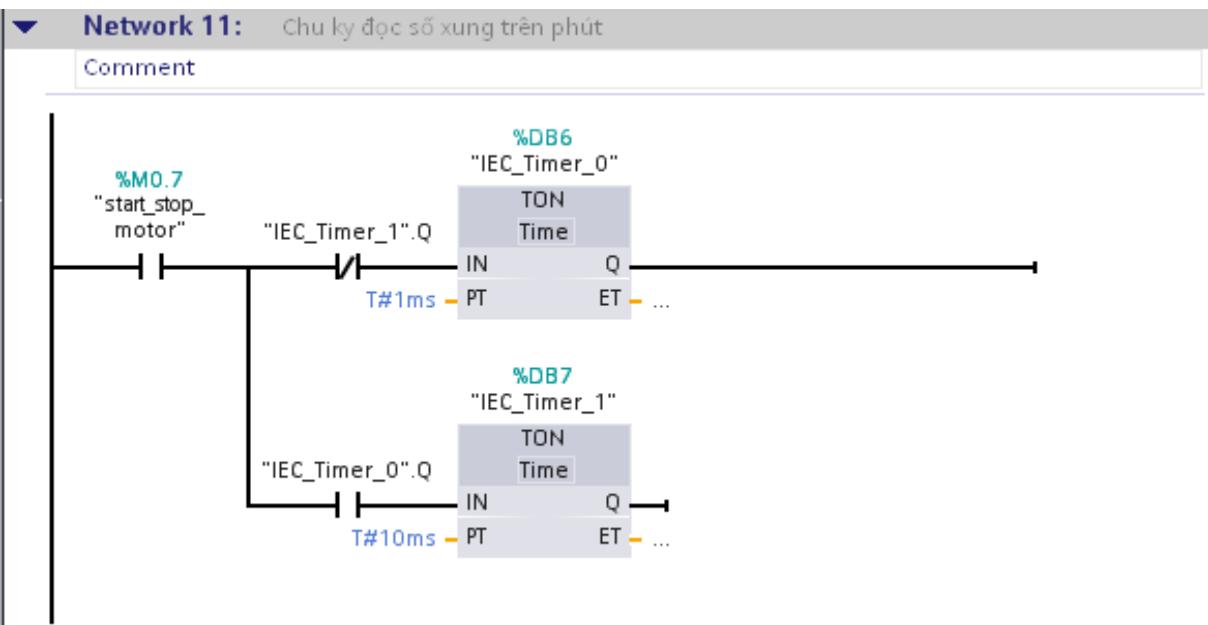


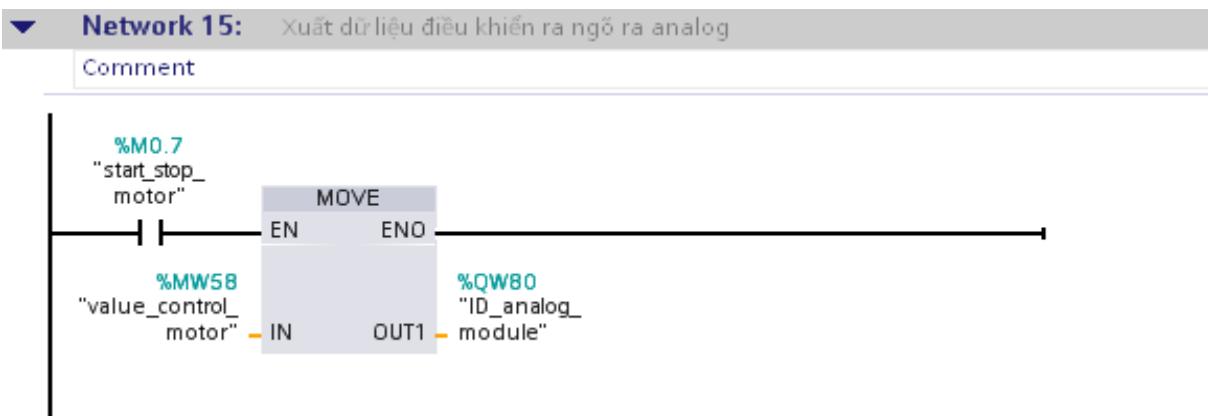
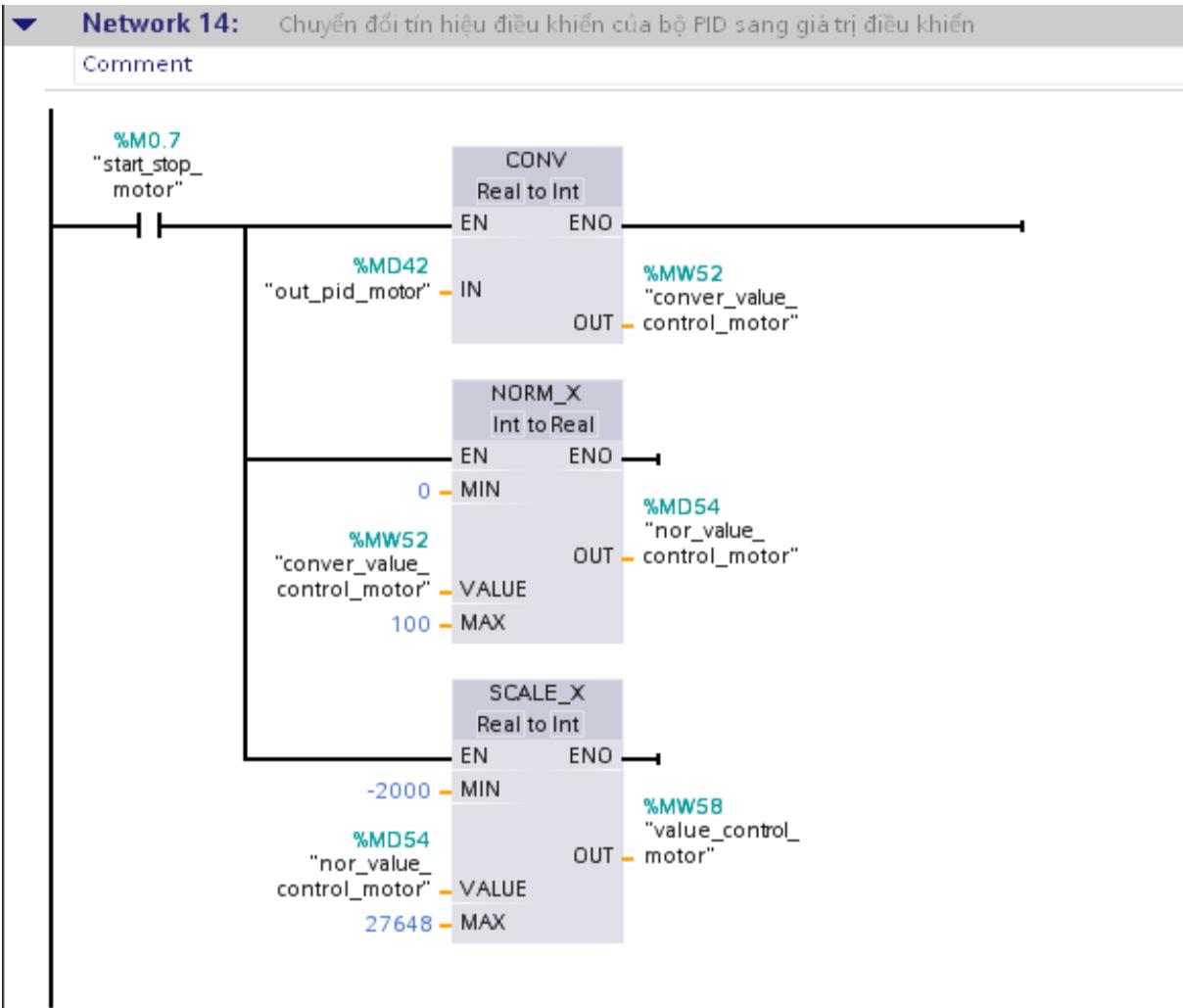
- Chương trình chính giám sát và điều khiển tốc độ động cơ:



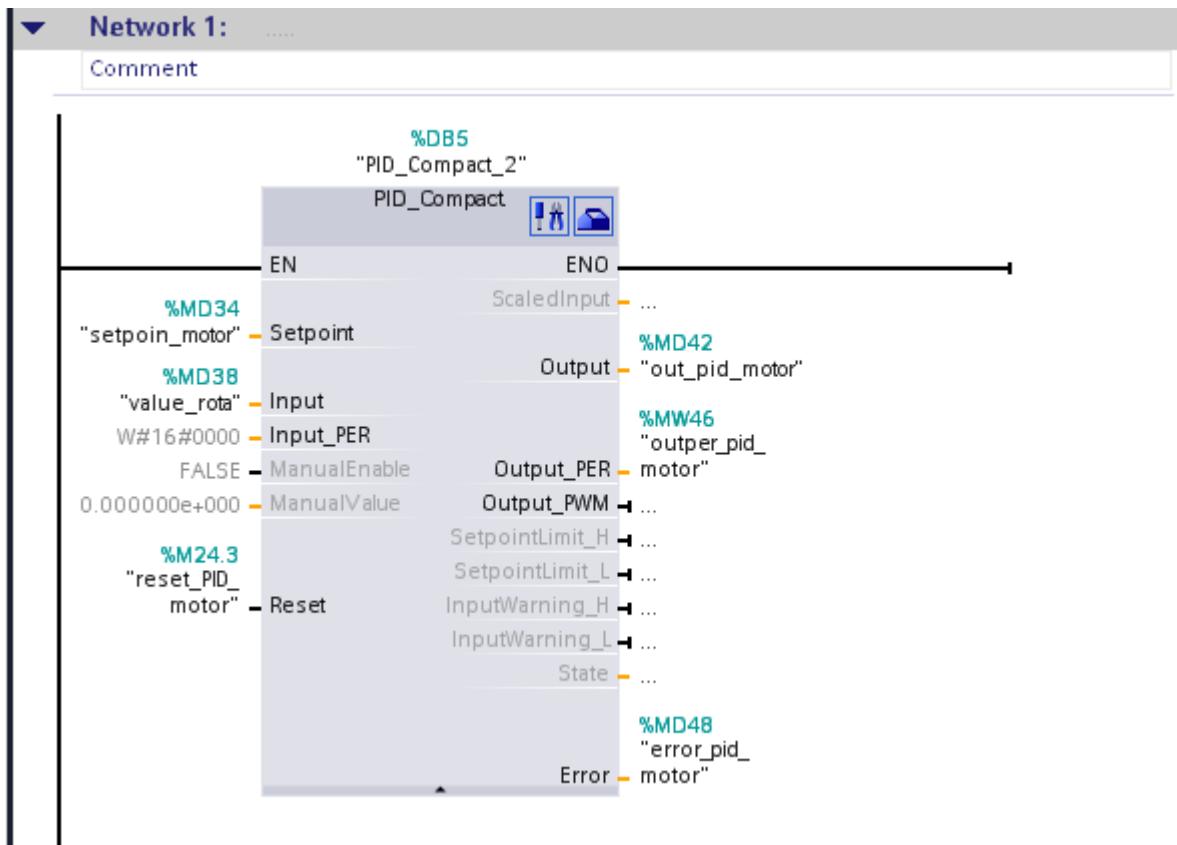








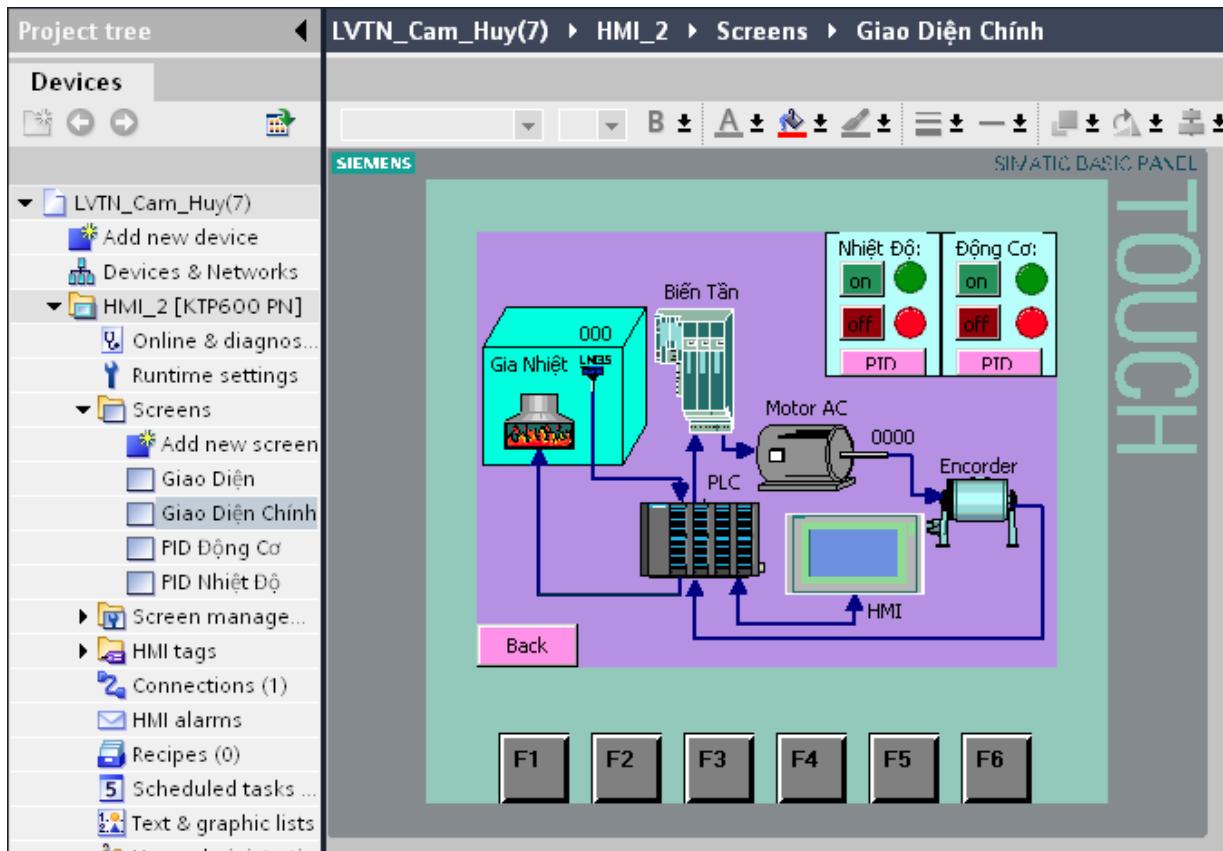
## PID điều khiển tốc độ động cơ:



## PHỤ LỤC 2

### CHƯƠNG TRÌNH ĐIỀU KHIỂN VÀ GIÁM SÁT

Chương trình được viết trên cùng một giao diện điều khiển như sau:



Hình 1: Chương trình giám sát và điều khiển trên HMI

Liên kết các tag:

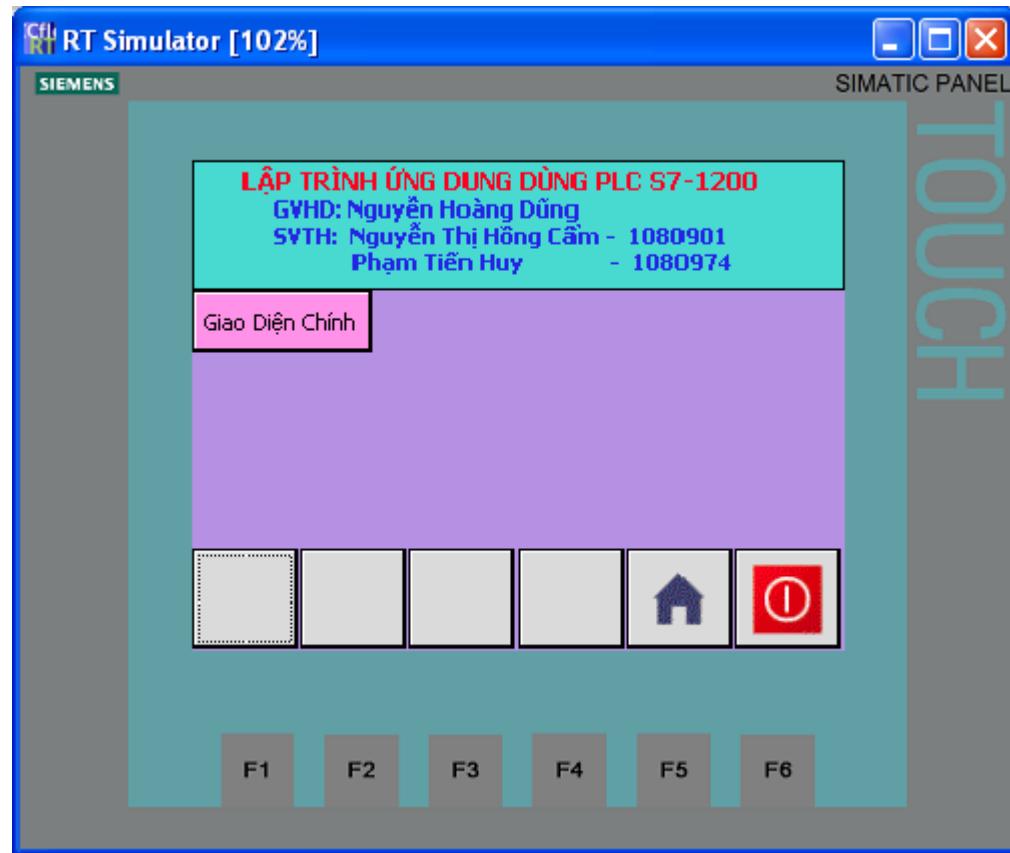
**HMI tags**

	Name ▲	Connection	Data type	PLC tag
	daochieu_motor	HMI_connection_2 ...	Bool	daochieu_motor ...
	daochieu_panel	HMI_connection_2	Bool	daochieu_panel
	daochieu_sw	HMI_connection_2	Bool	daochieu_sw
	device_temp	HMI_connection_2	Bool	device_temp
	PID_Compact_1_Output	HMI_connection_2	Real	PID_Compact_1.Output
	PID_Compact_1_Output_PER	HMI_connection_2	Word	PID_Compact_1.Output...
	PID_Compact_1_Output_PWM	HMI_connection_2	Bool	PID_Compact_1.Output...
	PID_Compact_2_Output	HMI_connection_2	Real	PID_Compact_2.Output
	PID_Compact_2_Setpoint	HMI_connection_2	Real	PID_Compact_2.Setpoint
	run_sensor_temp	HMI_connection_2	Bool	run_sensor_temp
	scale_temp	HMI_connection_2	Int	scale_temp
	set_poin_motor	HMI_connection_2	Real	setpoint_motor
	setpoint_temp	HMI_connection_2	Real	setpoint_temp
	start_panel_motor	HMI_connection_2	Bool	start_panel_motor
	start_panel_temp	HMI_connection_2	Bool	start_panel_temp
	start_stop_motor	HMI_connection_2	Bool	start_stop_motor
	start_stop_temp	HMI_connection_2	Bool	start_stop_temp
	start_sw_temp	HMI_connection_2	Bool	start_sw_temp
	stop_panel_motor	HMI_connection_2	Bool	stop_panel_motor
	stop_panel_temp	HMI_connection_2	Bool	stop_panel_temp
	stop_sw_temp	HMI_connection_2	Bool	stop_sw_temp
	value_rota	HMI_connection_2	Real	value_rota
	value_temp	HMI_connection_2	Real	value_temp
	<Add new>			

Hình 2: Các tag liên kết HMI với PLC

Sau đây là từng giao diện điều khiển:

- Giao diện mặc định khi khởi động HMI:

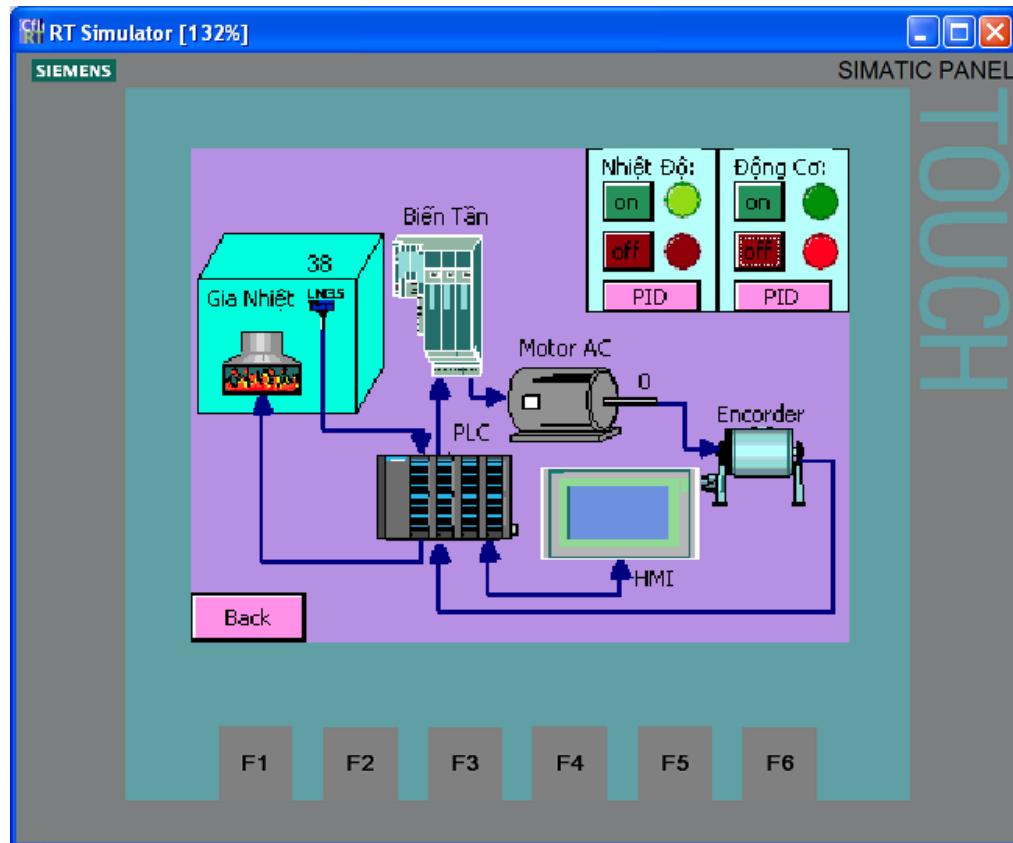


Hình 3: Giao diện mặc định sau khởi động

Giải thích hoạt động:

Khi khởi động HMI, giao diện này sẽ được hiển thị. Để vào giao diện điều khiển, ta nhấp vào nút “Giao diện chính”. Để tắt HMI, ta chọn biểu tượng shutdown: | |

- Giao diện chính:

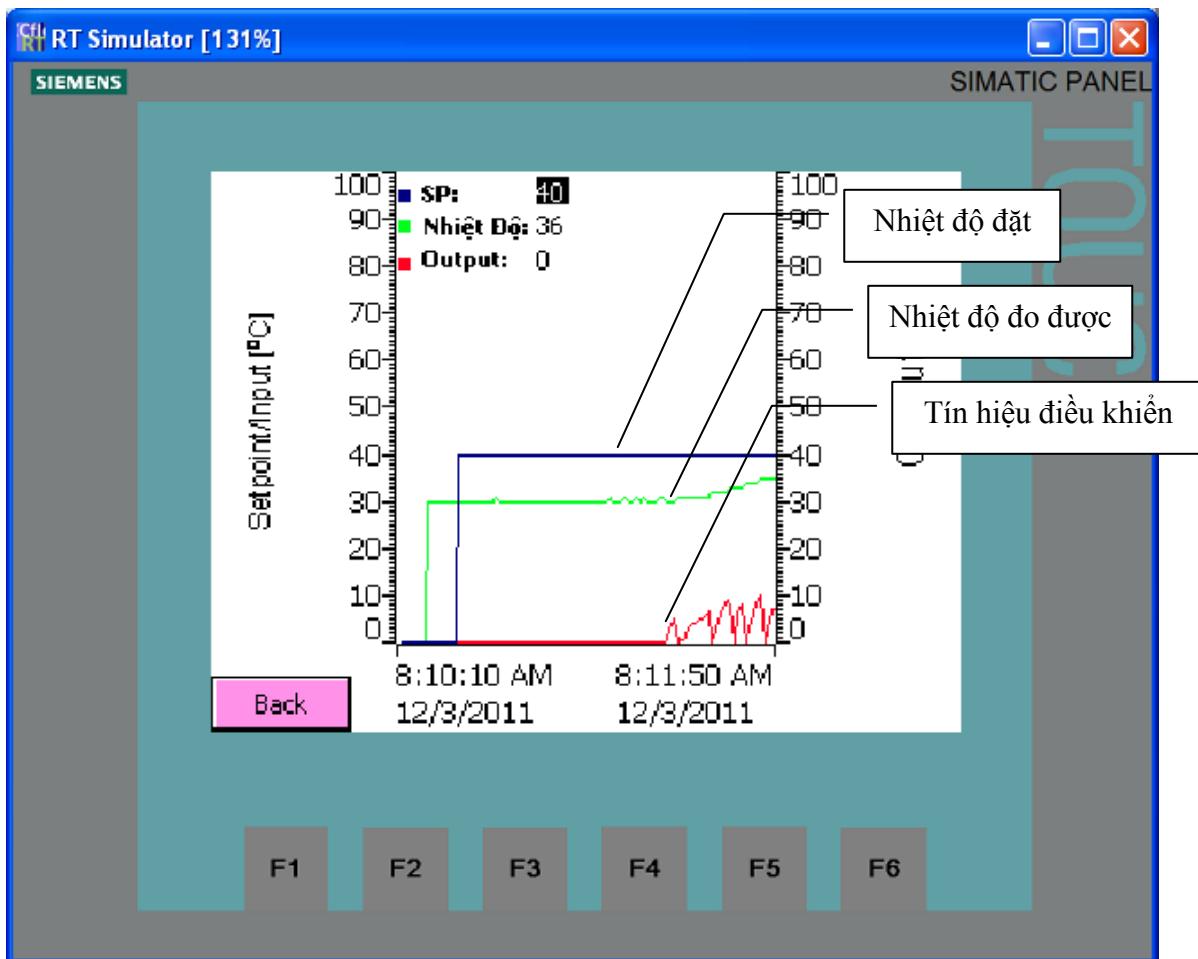


**Hình 4: Giao diện chính**

Giải thích hoạt động:

Giao diện chính sẽ hiển thị bảng điều khiển tắt mở (on/off) hệ thống điều khiển. Hai hệ thống có thể hoạt động độc lập hoặc cùng lúc. Màn hình cũng hiển thị trạng thái làm việc của hệ thống qua trạng thái các dây dẫn. Trên màn hình chính cũng hiển thị giá trị nhiệt độ và tốc độ động cơ. Để xem biểu đồ PID hoặc thay đổi giá trị đặt ta nhấp vào nút PID.

- Giao diện PID nhiệt độ:

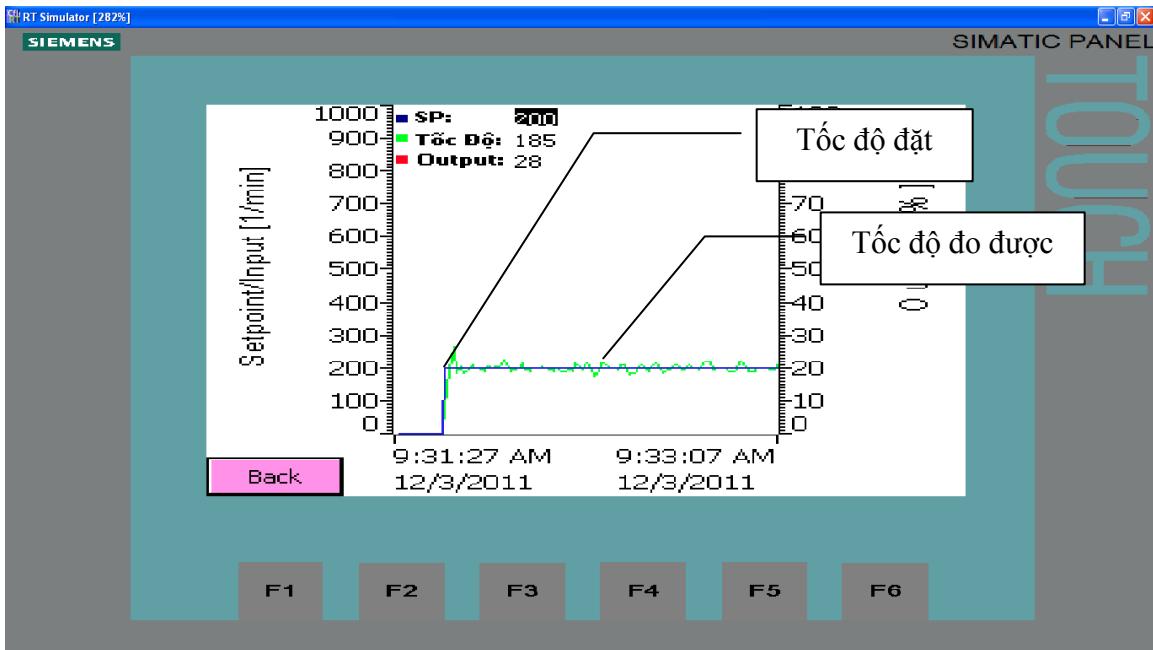


**Hình 5: Giao diện PID giám sát và điều khiển nhiệt độ**

Giải thích hoạt động:

Giao diện này hiển thị biểu đồ trạng thái tín hiệu đặt (nhiệt độ đặt), tín hiệu vào (nhiệt độ đo được) và tín hiệu điều khiển (công suất ra). Trên giao diện này ta có thể thay đổi giá trị đặt tùy ý.

- Giao diện PID động cơ:



**Hình 6: Giao diện PID giám sát và điều khiển tốc độ động cơ**

Giải thích hoạt động:

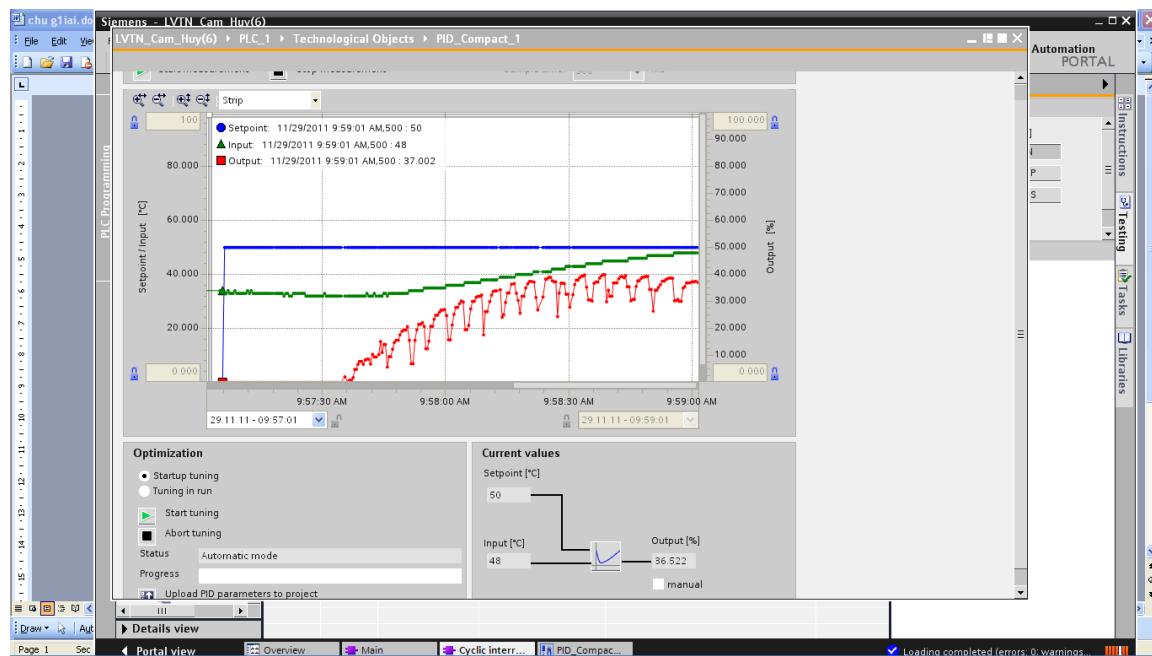
Tương tự như giao diện PID nhiệt độ, trên giao diện này cũng hiển thị trạng thái tín hiệu đặt (tốc độ đặt), tín hiệu vào (tốc độ đo được) và tín hiệu điều khiển (công suất ra). Trên giao diện này ta có thể thay đổi giá trị đặt tùy ý.

## PHỤ LỤC 3

### GIÁM SÁT VÀ ĐIỀU KHIỂN HỆ THỐNG

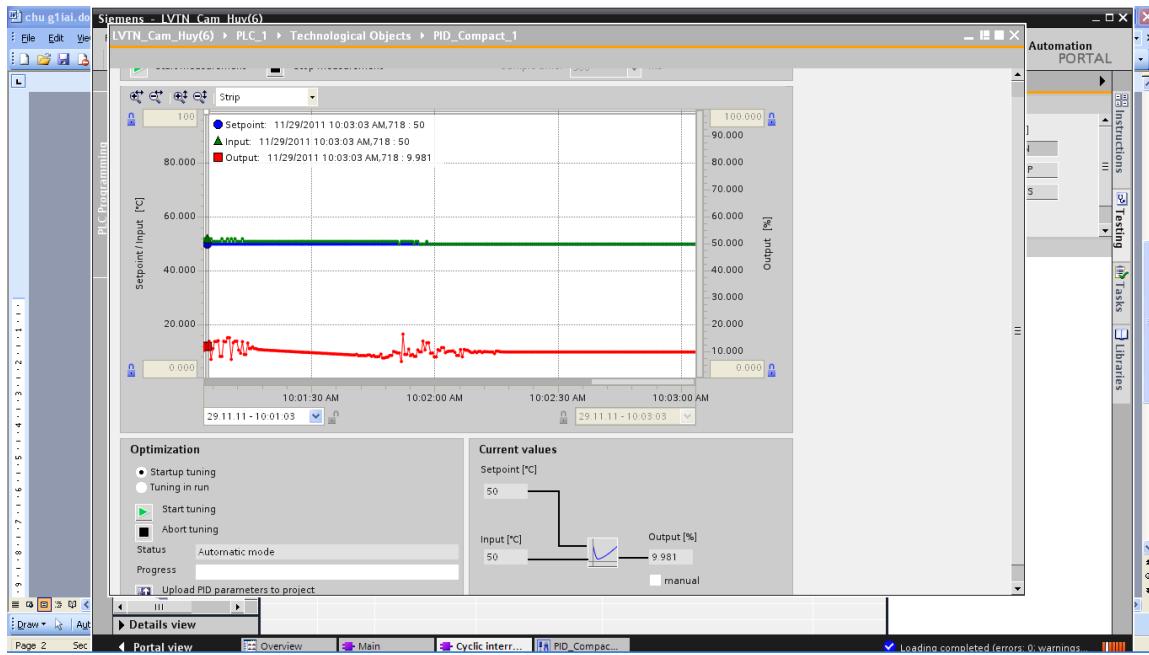
#### Điều khiển và giám sát nhiệt độ:

- Khởi động hệ thống điều khiển và giám sát nhiệt độ. Đặt giá trị nhiệt độ mong muốn. Hệ thống sẽ tự động điều chỉnh nhiệt độ như hình bên dưới:



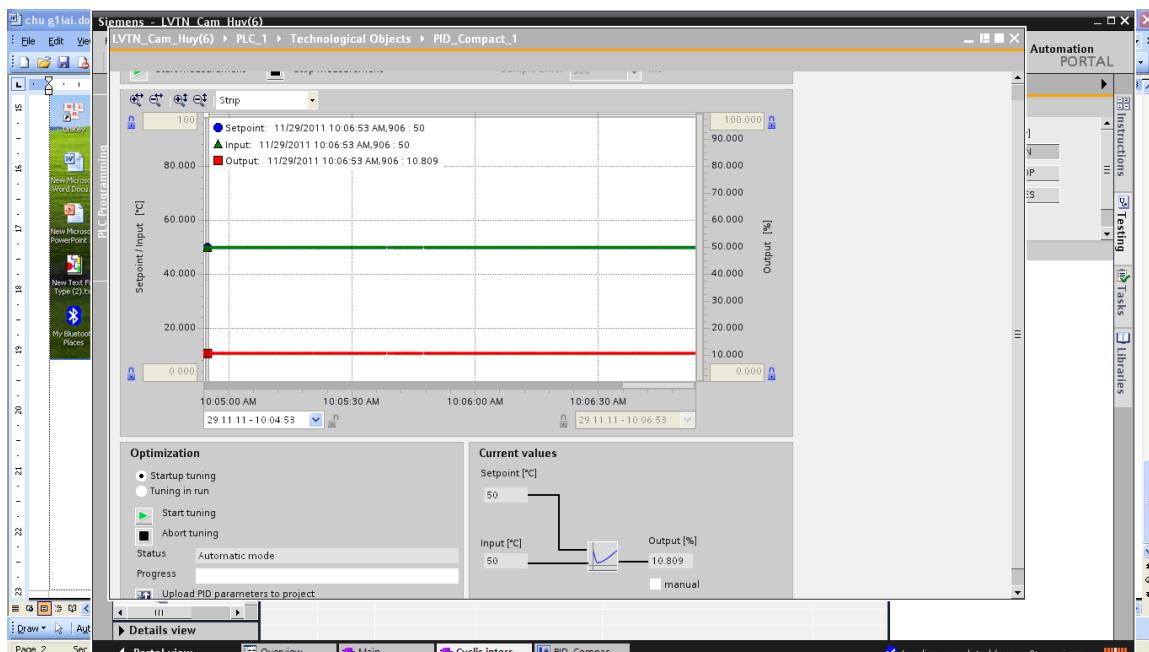
Hình 7: Điều khiển giám sát nhiệt độ

- Bộ điều khiển sẽ tiếp tục điều chỉnh cho sai số xác lập nhỏ nhất:



**Hình 8: Giá trị nhiệt độ xác lập**

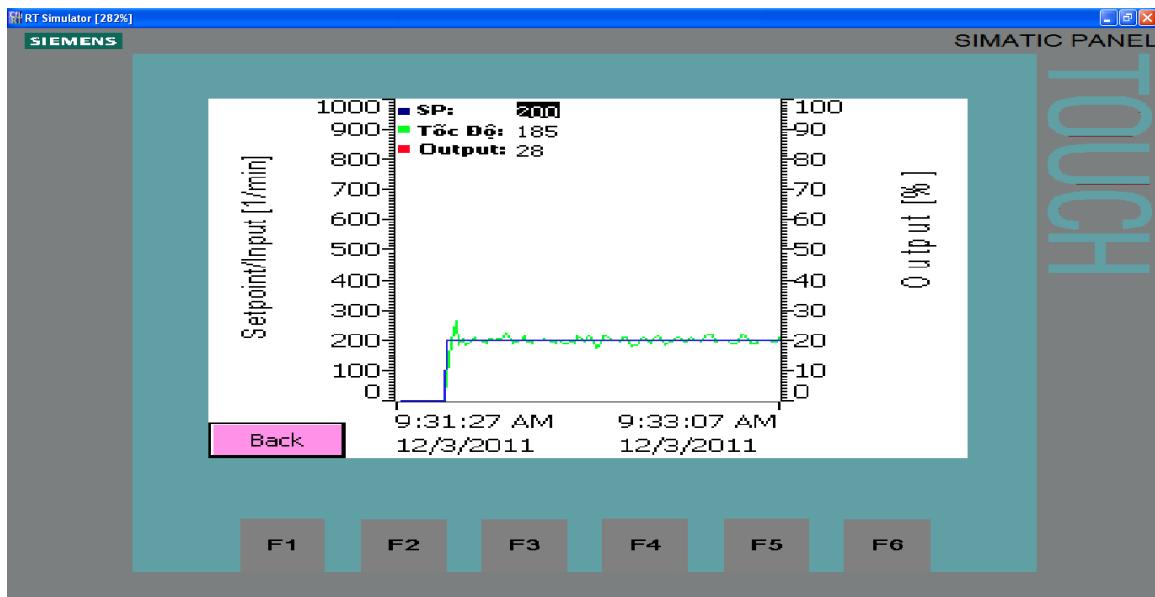
- Giá trị nhiệt độ thực tế bằng và ổn định ở giá trị nhiệt độ đặt:



**Hình 9: Giá trị nhiệt độ ổn định ở nhiệt độ đặt**

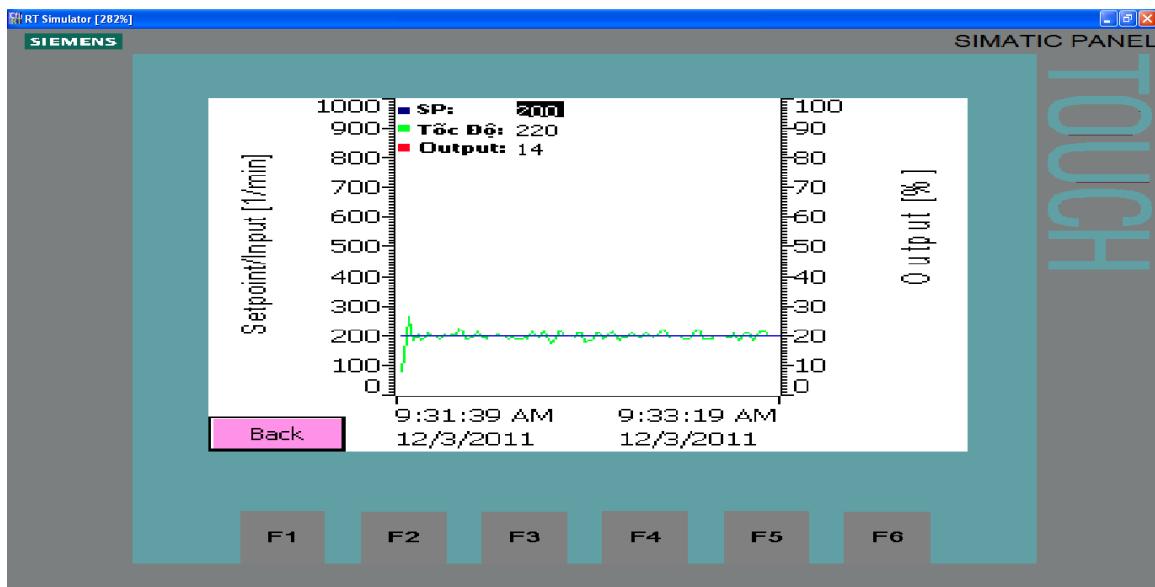
## Điều khiển và giám sát tốc độ động cơ:

- Khởi động hệ thống điều khiển và giám sát tốc độ động cơ. Đặt giá trị tốc độ mong muốn. Hệ thống sẽ tự động điều chỉnh tốc độ như hình bên dưới:



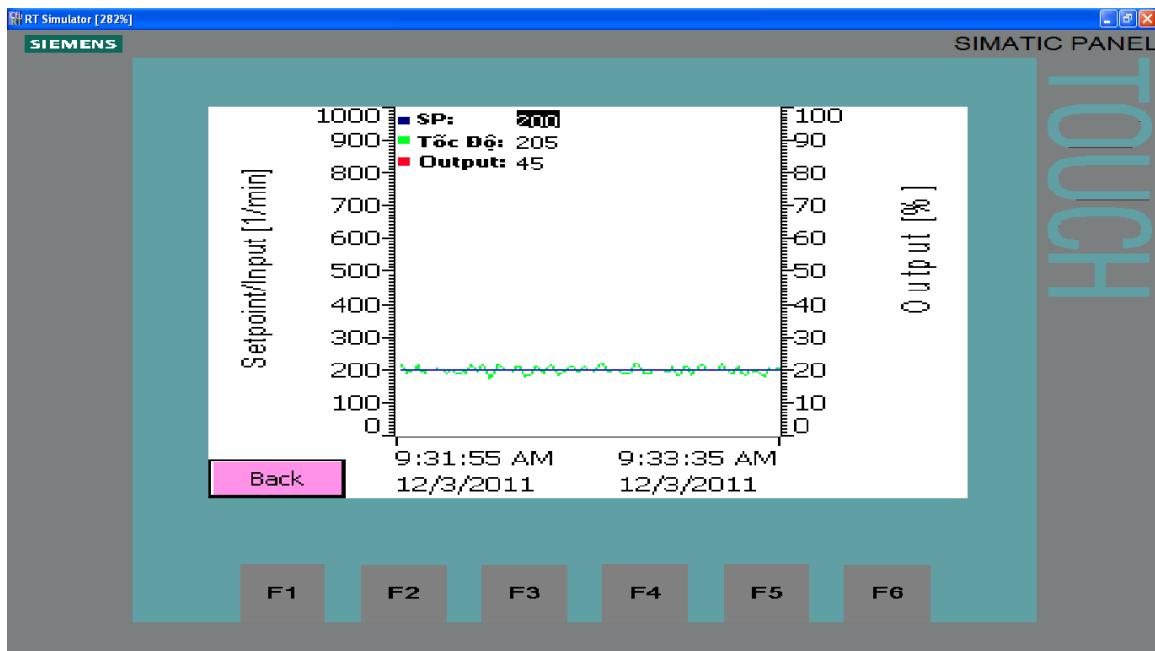
Hình 10: Điều khiển giám sát tốc độ động cơ

- Bộ điều khiển sẽ tiếp tục điều chỉnh cho sai số xác lập nhỏ nhất:



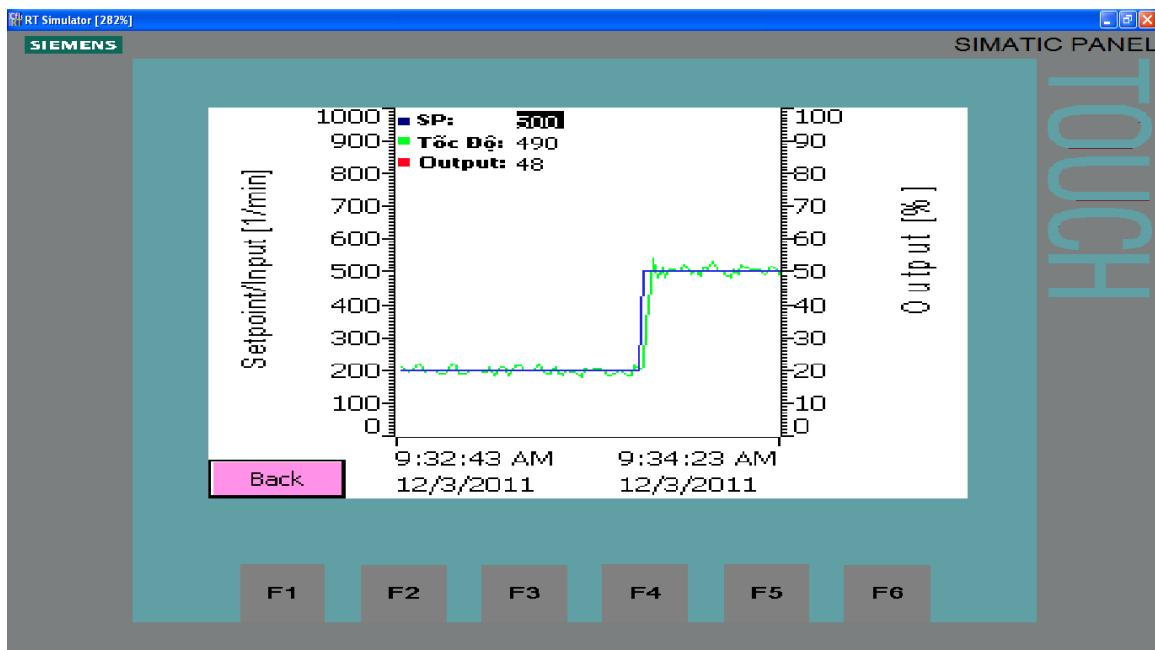
Hình 11: Giá trị tốc độ xác lập

- Giá trị tốc độ thực tế bằng và ổn định ở giá trị tốc độ đặt:



Hình 12: Giá trị nhiệt độ ổn định ở nhiệt độ đặt

Ta có thể đặt lại giá trị mong muốn điều khiển. Bộ điều khiển vẫn hoạt động đúng chức năng:



Hình 12: Thay đổi giá trị đặt

## TÀI LIỆU THAM KHẢO

- [1] **Nguyễn Hữu Đông, Nguyễn Ngọc Phú, Hệ thống phân loại sản phẩm Dài, Trung bình, Ngắn dùng S7-200 và WinCC**, LVTN chuyên ngành cơ điện tử, Đại học Cần Thơ, 2011, trang 119 đến 124.
- [2] **Nguyễn Thị Phương Hà, Huỳnh Thái Hoàn, Lý thuyết điều khiển tự động**, Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh, chương 6.
- [3] **Lâm Văn Khởi, Nguyễn Quốc Thái, Điều khiển giám sát hệ thống phân loại sản phẩm màu**, LVTN chuyên ngành Kỹ thuật Điều khiển, Đại học Cần Thơ, 2011, chương 2.
- [4] **Thái Văn Danh, Điều khiển PID sử dụng S7-1200**, LVTN chuyên ngành cơ điện tử, Đại học Cần Thơ, 2011, chương 3 và 4.
- [5] **Siemens, SIMATIC S7-1200 Programmable controller System Manual**, 11/2009.
- [6] **Siemens, SIMATIC S7-1200 and STEP 7 Basic V10.5**, 11/2009
- [7] **Siemens, SIMATIC S71200 Easy book en-US en-US**, 04/2011.
- [8] **Siemens, SIMATIC S7-1200 STEP 7 V10.5 Help**, 02/2010.