



Giới thiệu bộ điều khiển Compact Logix 1769-L32E

plc-programmable logic controllers (Trường Đại học Sư phạm Kỹ Thuật Thành phố Hồ Chí Minh)



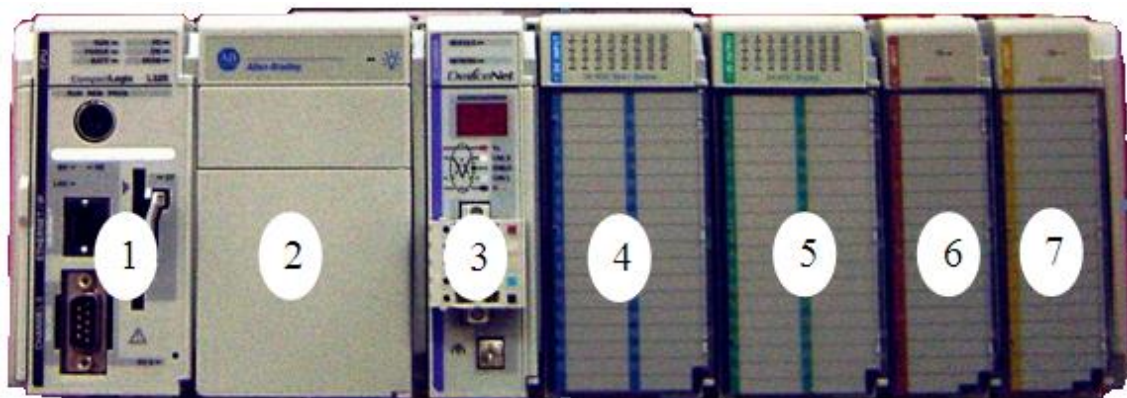
Scan to open on Studeersnel

Giới thiệu bộ điều khiển CompactLogix

Bộ điều khiển CompactLogix 1769-L32E

CompactLogix 1769-L32E là một sản phẩm trong dòng sản phẩm 1769-L3x của hãng Allen Bradley, cung cấp một giải pháp Logix cho các ứng dụng từ nhỏ đến trung bình. Thông thường những ứng dụng này yêu cầu hạn chế số lượng I/O và khả năng giao tiếp, các Module được gắn tiếp theo bên phải, Module gắn vào đầu tiên có tên là Local 1 và tăng dần lên.

CompactLogix 1769-L32E cung cấp bộ điều khiển tích hợp cổng giao tiếp Ethernet IP, RS 232, module 1769 SDN DeviceNet dùng để giao tiếp với mạng DeviceNet, module I/O Digital, Analog.

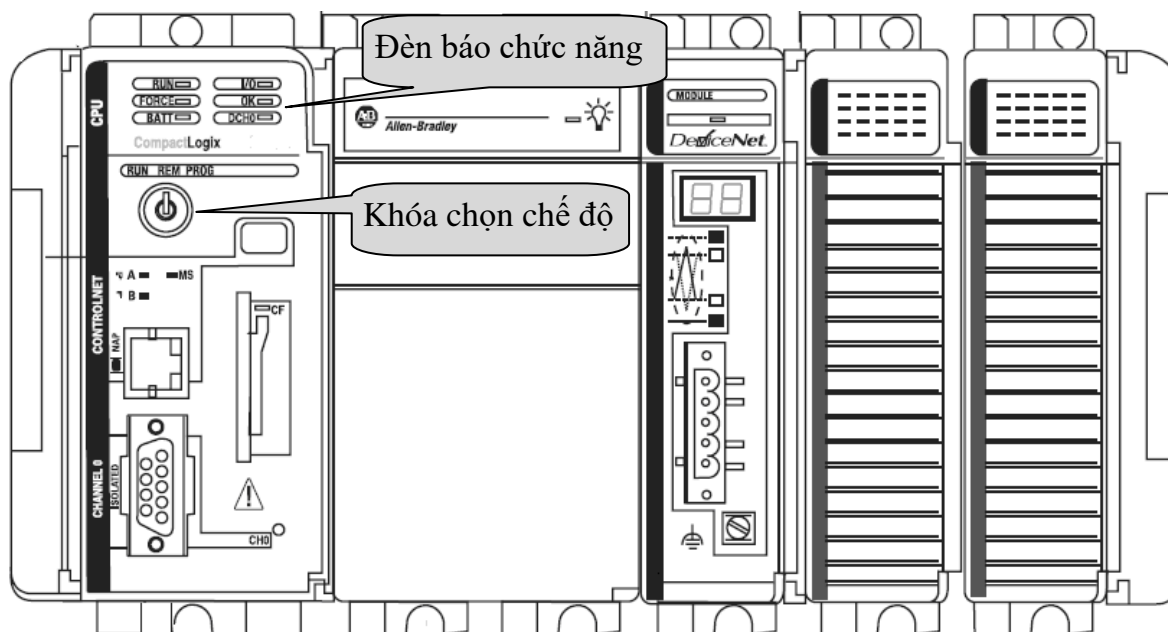


Cấu trúc một bộ CompactLogix 1769-L32E.

Chú thích các Module trong CompactLogix 1769-L32E

- | | |
|---|--|
| 1. 1769 L32E Controller (Compact Bus). | 5. 1769-OB32 Current Sourcing 24V DC Output (Local 3). |
| 2. Source. | |
| 3. DeviceNet (Local 1). | 6. 1769-IF4 Analog Input (Local 4). |
| 4. 1769-IQ32 Sinking/Sourcing 24V DC Input (Local 2). | 7. 1769-OF2 Analog Output (Local 5). |

1. Các chế độ và tín hiệu đèn trên PLC CompactLogix 1769-L32E.



Vị trí các đèn báo và khóa chọn chế độ.

Các chế độ của PLC Compact Logix 1769 L32E.

Vị trí của khóa	Chức năng
Run	<p>Tải project hiện tại trên PLC về máy tính.</p> <p>Chương trình đang chạy và cho phép các ngõ ra của PLC hoạt động</p> <p>Không thể tạo mới hay xóa các Tag, Bit, hay một chương trình.</p> <p>Không thể chuyển về các chế độ Run, Program, Test trong chương trình.</p>
Prog	<p>Các ngõ ra của PLC không hoạt động.</p> <p>Tải chương trình lên PLC hay tải chương trình từ PLC về máy tính.</p> <p>Có thể tạo, xóa, sửa đổi các Tag, Bit, hay một đoạn chương trình.</p> <p>Không thể chuyển về các chế độ Run, Program, Test trong chương trình.</p>

Rem	Tải chương trình lên PLC hay tải chương trình từ PLC về máy tính. Có thể chuyển về các chế độ Run, Program, Test trong chương trình RSLogix 5000.
------------	--

Trạng thái của các đèn chức năng.

Đèn chỉ báo	Trạng thái đèn	Mô tả
RUN	Tắt.	PLC đang ở chế độ Program hay Test
	Sáng màu xanh.	PLC đang ở chế độ RUN.
FORCE	Tắt.	Không có Tag nào bị ép buộc hoạt động. I/O force bị tắt.
	Sáng màu vàng.	I/O force đang kích hoạt.
	Nhấp nháy màu vàng.	Một hay nhiều Bit ở ngõ vào, ra đang bị ép buộc hoạt động.
BAT	Tắt.	Đang hoạt động tốt.
	Sáng màu đỏ.	Không có Pin. Pin còn khoảng 5% và cần được thay thế.
I/O	Tắt.	Không có thiết bị phần cứng nào được định cấu hình trong I/O. PLC không có chương trình.
	Sáng màu xanh.	PLC đang giao tiếp với các thiết bị phần cứng được định cấu hình trong I/O.
	Nhấp nháy màu xanh.	Một hay nhiều thiết bị trong I/O không đáp ứng với PLC.
	Nhấp nháy màu đỏ.	PLC không giao tiếp với bất kỳ thiết bị nào.

		PLC đang lỗi.
OK	Tắt.	Không có nguồn.
	Nhấp nháy màu đỏ.	Nếu lần là lần đầu sử dụng PLC thì PLC cần Update Firmware. Nếu không phải lần đầu thì PLC bị lỗi, để xóa lỗi thì: chuyển khóa từ Prog ⇒ Run ⇒ Prog và chọn Go Online trong chương trình RSLogix 5000.
	Sáng màu đỏ.	Bộ điều khiển phát hiện lỗi, cần thực hiện các bước sau để xóa lỗi từ bộ nhớ. Tắt, mở lại nguồn điện. Đổ chương trình vào PLC. Chuyển khóa về chế độ Run.
	Sáng màu xanh.	PLC đang hoạt động tốt.
	Nhấp nháy màu xanh.	PLC đang lưu trữ hay tải một chương trình từ bộ nhớ.

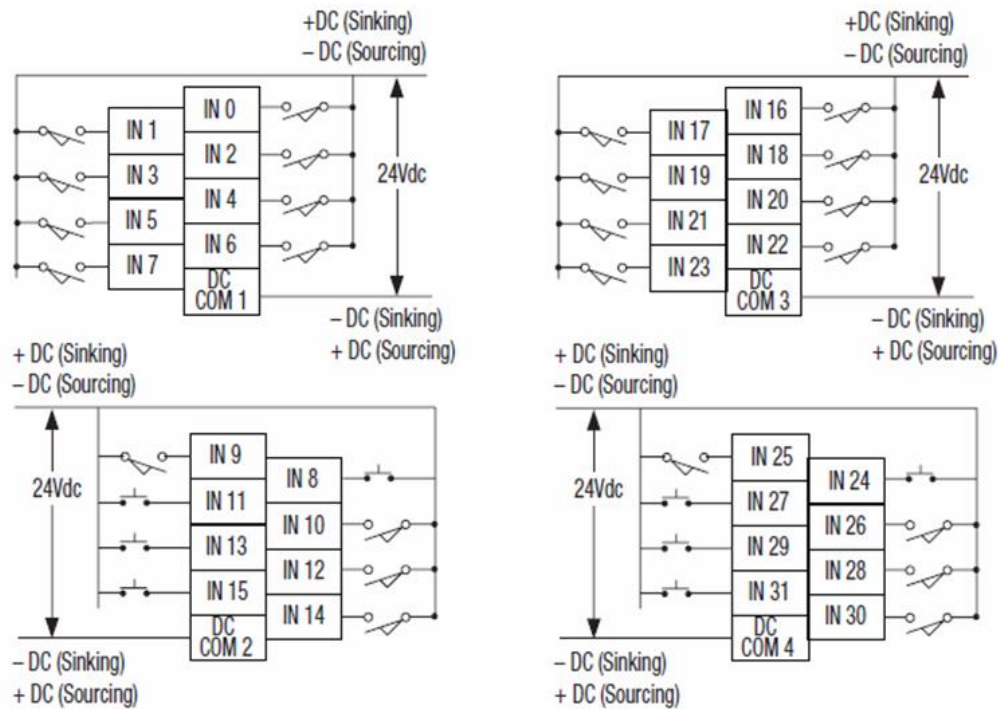
Trong bộ điều khiển CompactLogix để thực hiện việc giao tiếp với các thiết bị ngoại vi ta phải gắn vào các modul mở rộng bao gồm:

2. Modul 1769-IQ32 Sinking/Sourcing 24V DC Input:

Điện áp nhỏ nhất: 10V, điện áp lớn nhất: 30V.

Số lượng ngõ vào: 32, thời gian chuyển trạng thái ON và OFF: 8ms.

Giá trị dòng điện: 1,5mA khi ở trạng thái OFF, và 2mA ở trạng thái ON.



Cấu trúc Module 1769-IQ32.

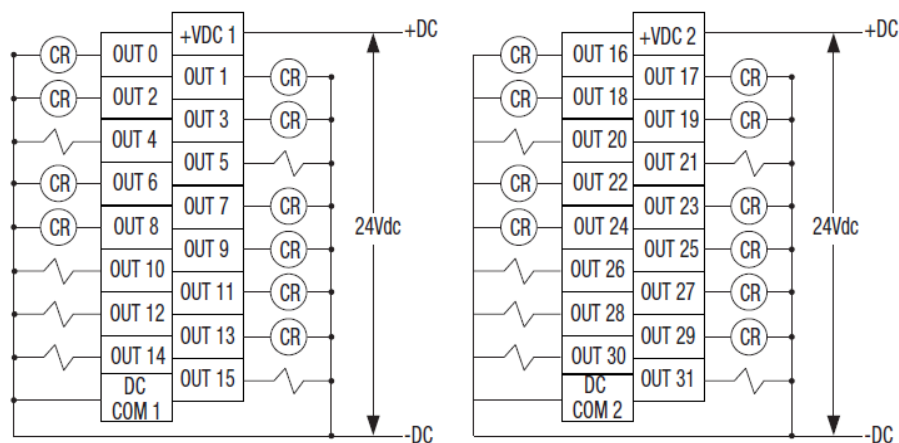
3. Modul 1769-OB32 Current Sourcing 24V DC Output:

Điện áp ngõ ra nhỏ nhất: 20,4V DC và lớn nhất: 26,4V DC.

Dòng điện ngõ ra nhỏ nhất: 1mA và lớn nhất là 1A.

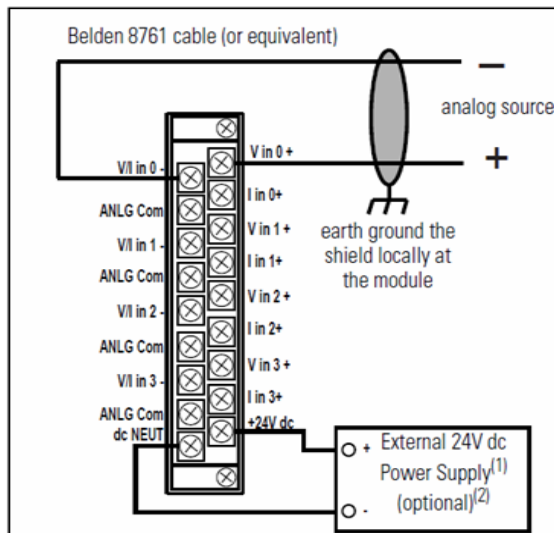
Thời gian chuyển trạng thái từ OFF sang ON là 0,1ms, ON sang OFF là 1ms.

Số lượng ngõ ra: 32

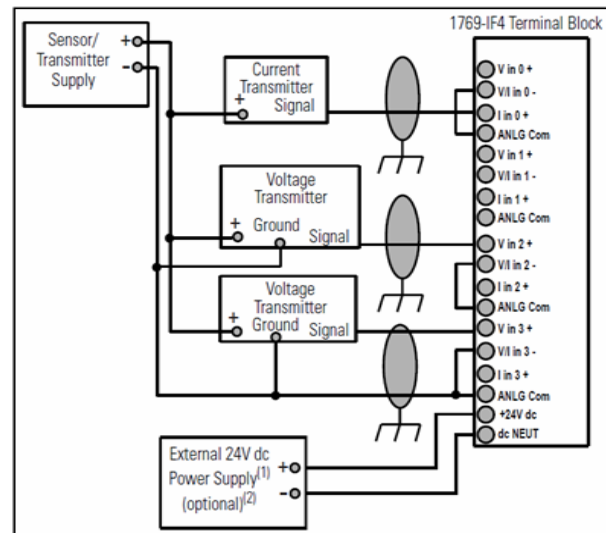


Cấu trúc Module 1769-OB32.

4. Modul 1769-IF4 Analog Input:



a. Kết nối vi sai ngõ vào.



b. Kết nối cảm biến và bộ khuếch đại tín hiệu.

Số ngõ vào: 4 tín hiệu điện áp, 4 tín hiệu dòng điện.

Có 4 kênh lựa chọn mức điện áp: -10V...10V DC, 0...10V DC, 0...5V DC, 1...5V DC.
Giá trị dòng điện tại 5V là 105mA.

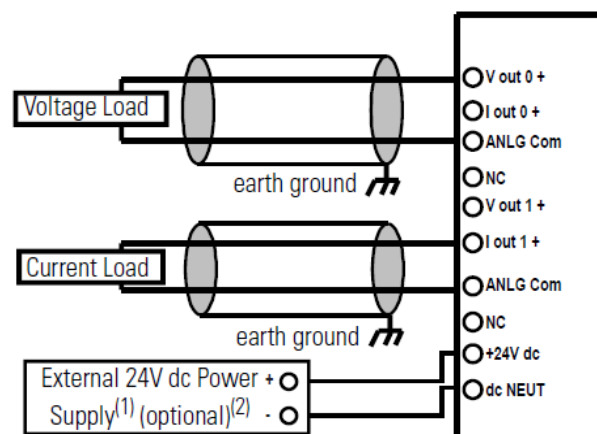
Có 2 kênh lựa chọn mức dòng điện: 0...20 mA, 4...20mA.

5. Modul 1769-OF2 Analog Output:

Số lượng ngõ ra: 2 ngõ ra điện áp, 2 ngõ ra dòng điện.

Có 4 kênh lựa chọn mức điện áp: -10V...10V DC, 0...10V DC, 0...5V DC, 1...5V DC.
Giá trị dòng điện tại 5V là 120mA.

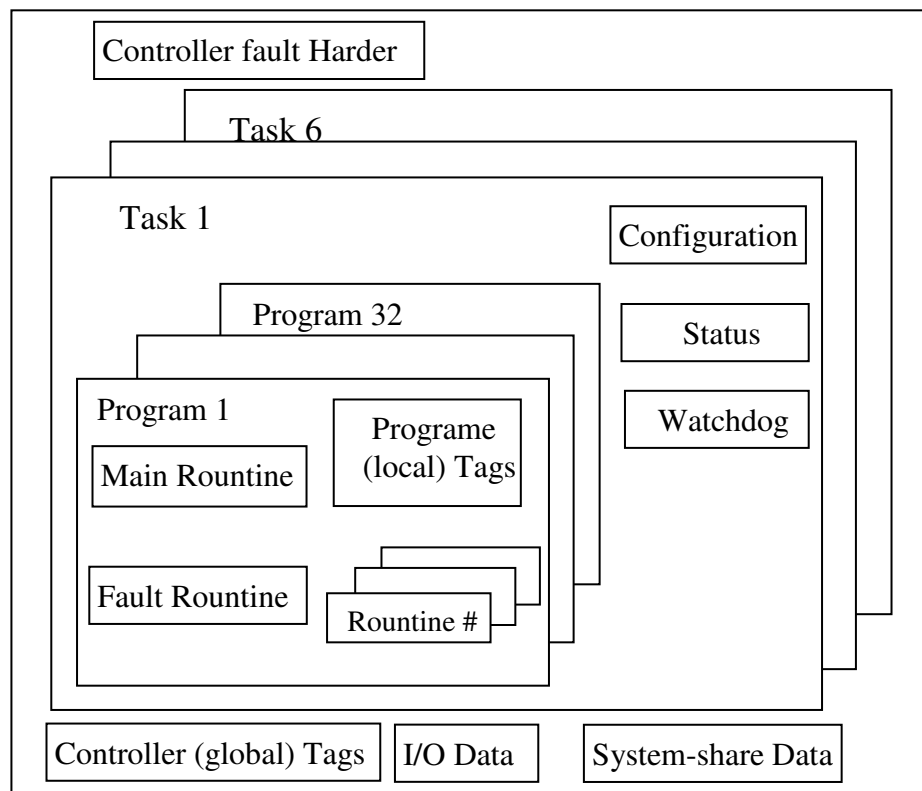
Có 2 kênh lựa chọn mức dòng điện: 0...20 mA, 4...20mA.



Sơ đồ và cách kết nối ngõ ra Analog.

6. Sơ đồ tổ chức một chương trình của CompactLogix L32E:

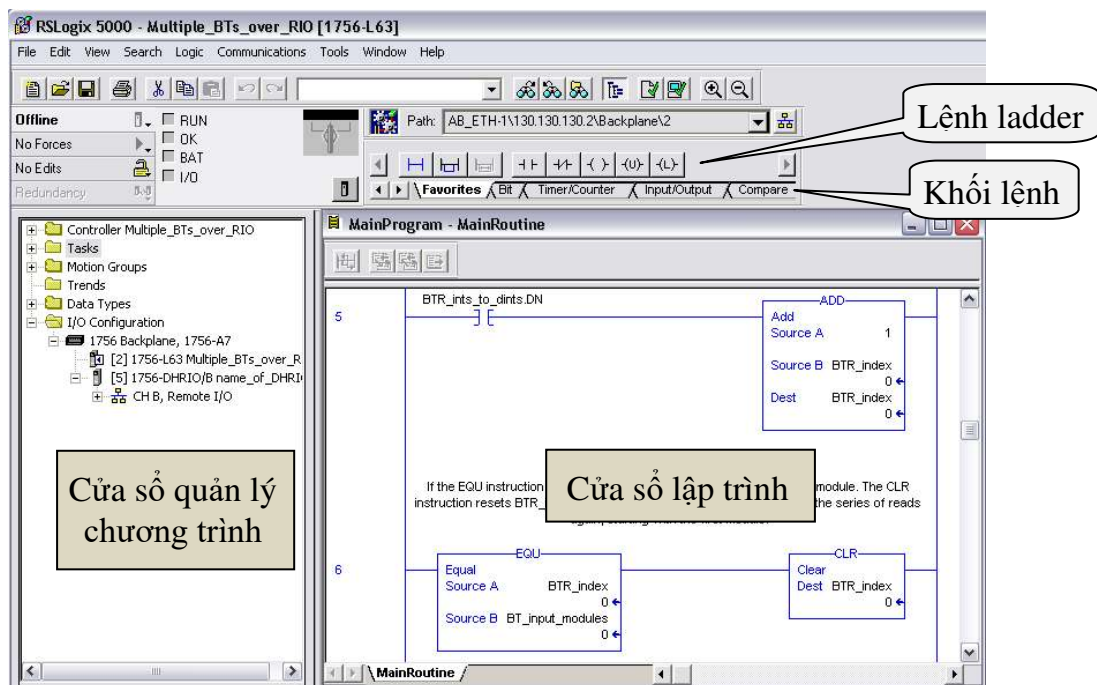
Cách quản lý của Compact Logix L32E nằm trong cửa sổ quản lý chương trình của phần mềm RSLogix 5000. Mỗi Project có 6 Task (nhiệm vụ), mỗi Task có 32 chương trình. Trong mỗi Project có Controller tags (Tags chung), mỗi chương trình cũng có Program tags (Tags riêng). Chi tiết về Task, Tags sẽ được trình bày trong phần giới thiệu về phần mềm RSLogix 5000 Enterprise.



Tổ chức chương trình của CompactLogix 1769-L32E.

Giới thiệu phần mềm RSLogix 5000 Enterprise

Phần mềm RSLogix 5000 Enterprise Series được thiết kế để làm việc với các nền tảng bộ điều khiển Logix 5000. Đây là phần mềm nhỏ gọn tuân thủ theo tiêu chuẩn IEC 61131-3 với phương pháp lập trình dạng bậc thang (ladder), văn bản có cấu trúc (structured text), sơ đồ khối chức năng (function block diagram) và trình bày chức năng một cách tuần tự để phát triển trong các chương trình ứng dụng.

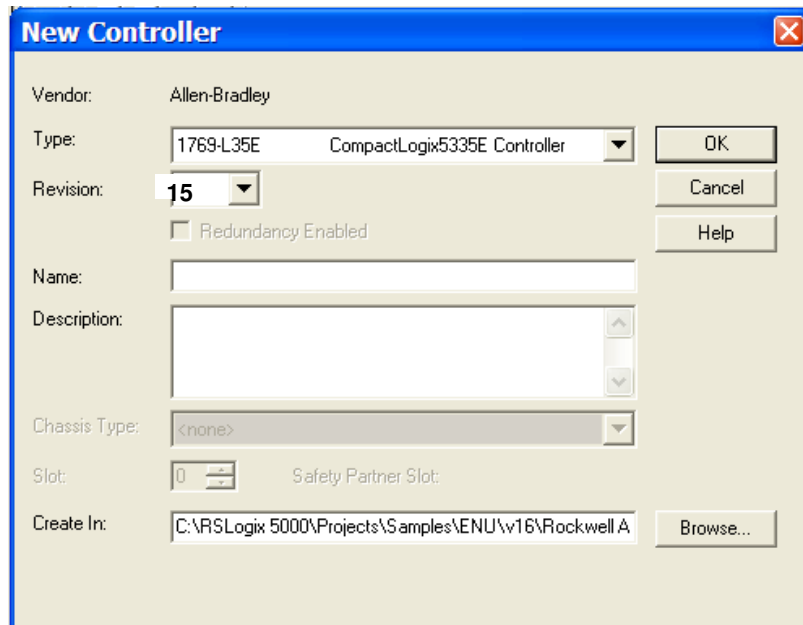


Giao diện phần mềm RSLogix 5000 Enterprise.

Để sử dụng phần mềm này một cách hiệu quả ta lần lượt thực hiện theo trình tự sau:

1. Khởi tạo 1 dự án mới

Để bắt đầu lập trình cho PLC ta vào biểu tượng RSLogix 5000 Enterprise, sau đó chọn File => New ta sẽ có được giao diện như hình sau:

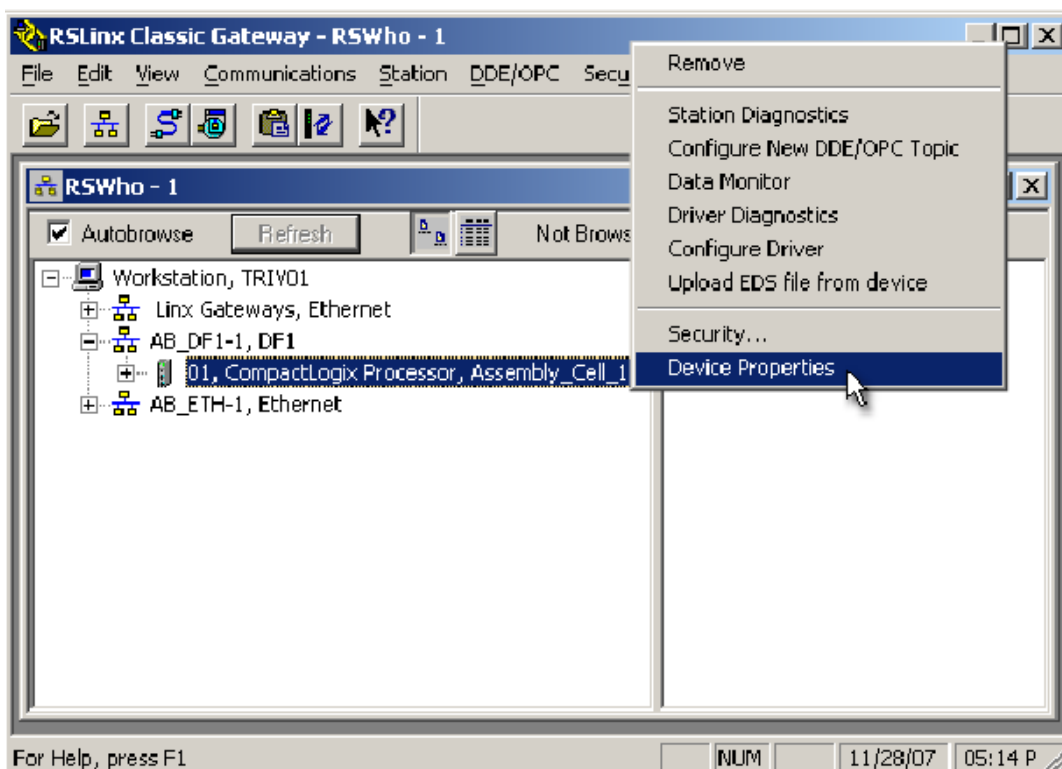


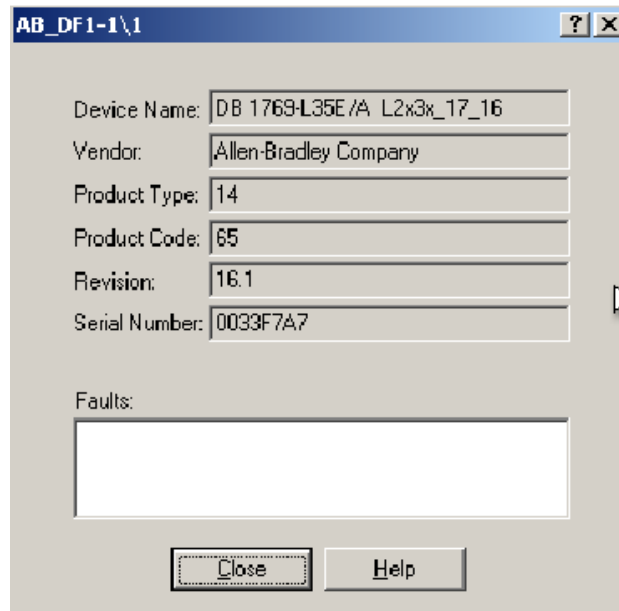
Tiếp theo ta đi cấu hình cho bộ điều khiển bằng cách chọn loại PLC trong ô **Type**. Với loại CompactLogix hiện hành ta chọn 1769- L32E.

Chọn phiên bản của phần mềm đang sử dụng trong ô **Revision** là 15, đặt tên cho Project trong **Name** hay viết những mô tả về chương trình trong **Description** và chọn đường dẫn để lưu chương trình trong phần **Creat in**.

Click **Ok** để hoàn thành khai báo cho bộ điều khiển.

Lưu ý: Muốn biết phiên bản firmware của PLC đang sử dụng làm như sau:

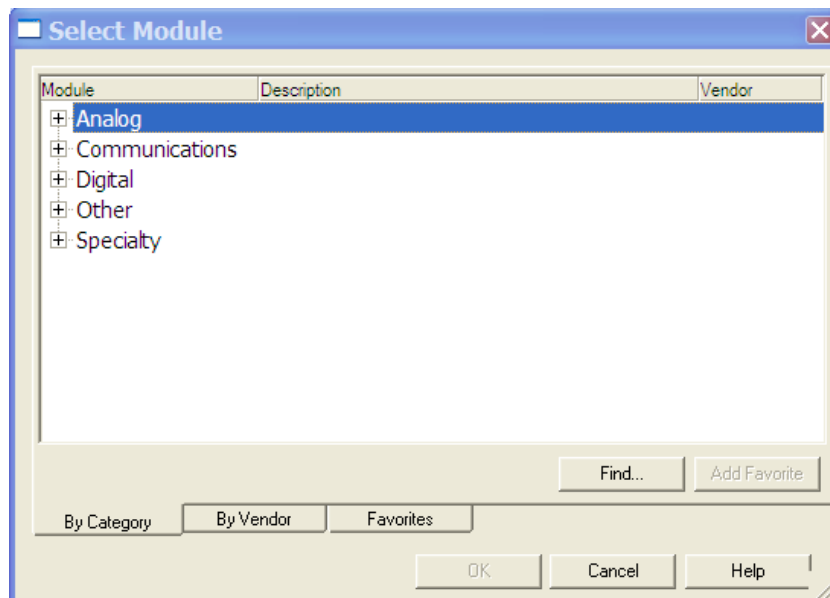




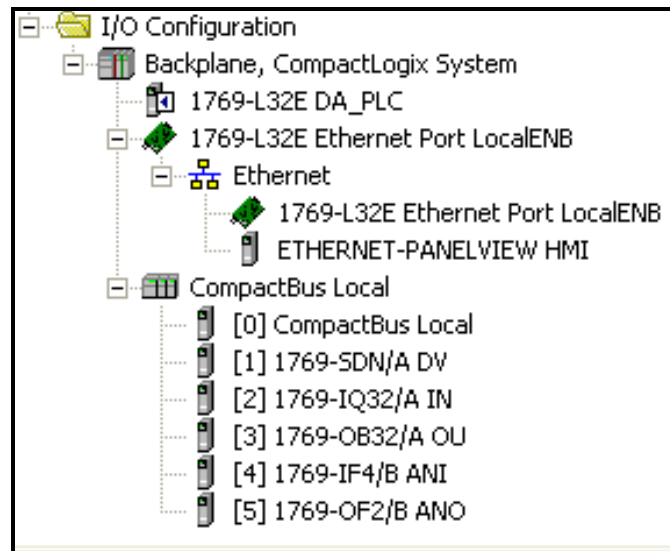
Tiếp theo ta đi khai báo cho các modul kết nối với bộ điều khiển.

Trong Folder **I/O Configuration** ở phần **Project Tree**, click phải chuột vào **Compact Bus Local** trong **Backplane, CompactLogix System** chọn **Insert Modul...**

Các modul được phân loại thành các nhóm như hình sau:



Sau khi chọn loại Modul đúng với mã hiệu trên modul thực tế, ta đặt tên Modul trong ô **Name**, Mô tả modul trong **Description** chọn **Slot** tương ứng trên Chassis, cuối cùng ta có đầy đủ các modul trong hệ thống như hình sau:

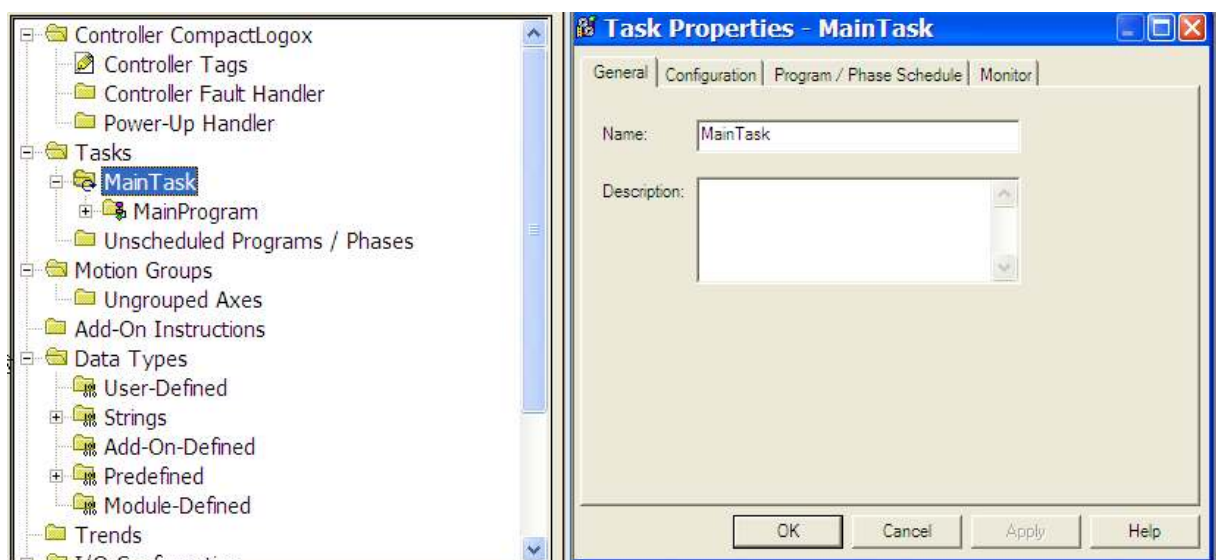


Lưu ý: Một ưu điểm lớn của PLC AB là mặc dù những modul đã kết nối với PLC nhưng nếu không sử dụng thì ta không cần phải khai báo.

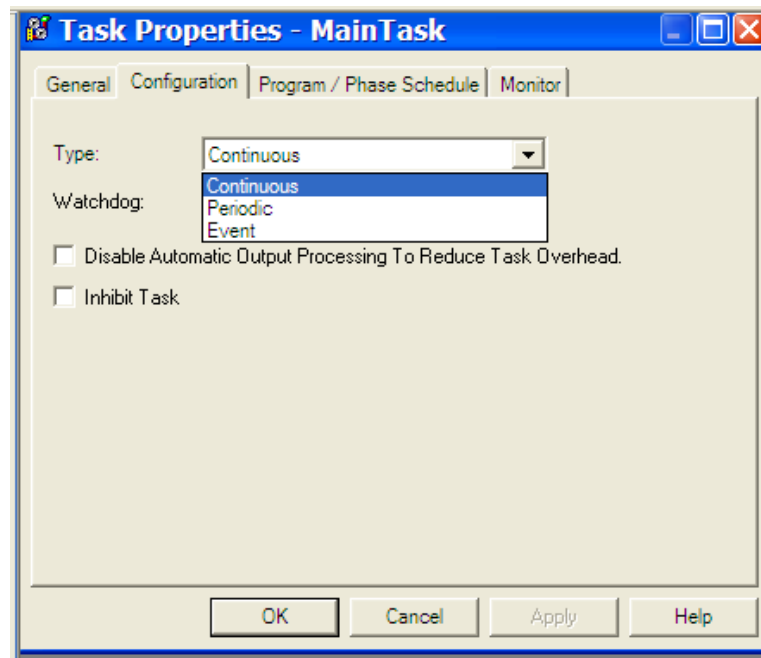
2. Thiết lập Task trong dự án

Khi khởi tạo một Project ta cần khai báo các Tasks dùng trong Project.

Double Click vào **Main Task** để đặt tên trong ô **Name** và mô tả cho Task trong ô **Description**.



Click vào thẻ **Configuration** để chọn loại Task trong ô **Type** và chọn thời gian tương ứng trong ô **Watchdog**.



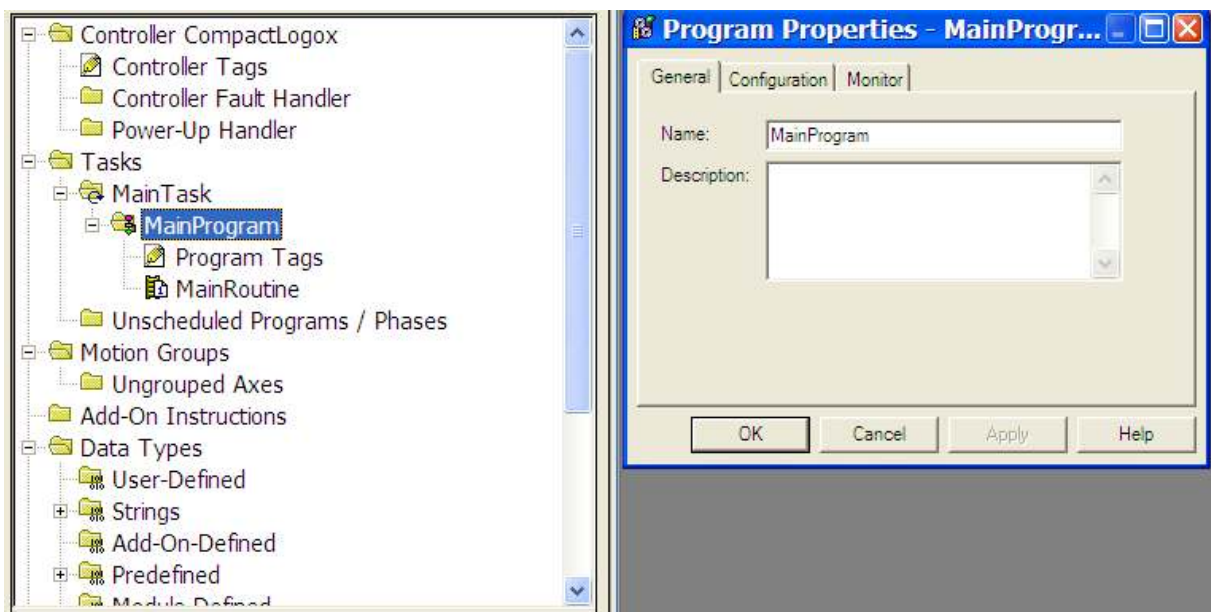
Nhấn **OK** khi hoàn tất công việc cài đặt.

Lưu ý: Trong 1 Project chỉ có 1 Continuous Task mà thôi.

3. Thiết lập Program trong Task

Để lập trình cho PLC, ban đầu ta cần khai báo chương trình (Program) như sau:

Double Click vào **MainProgram** để đặt tên cho chương trình trong ô **Name** và mô tả cho chương trình trong ô **Description**.

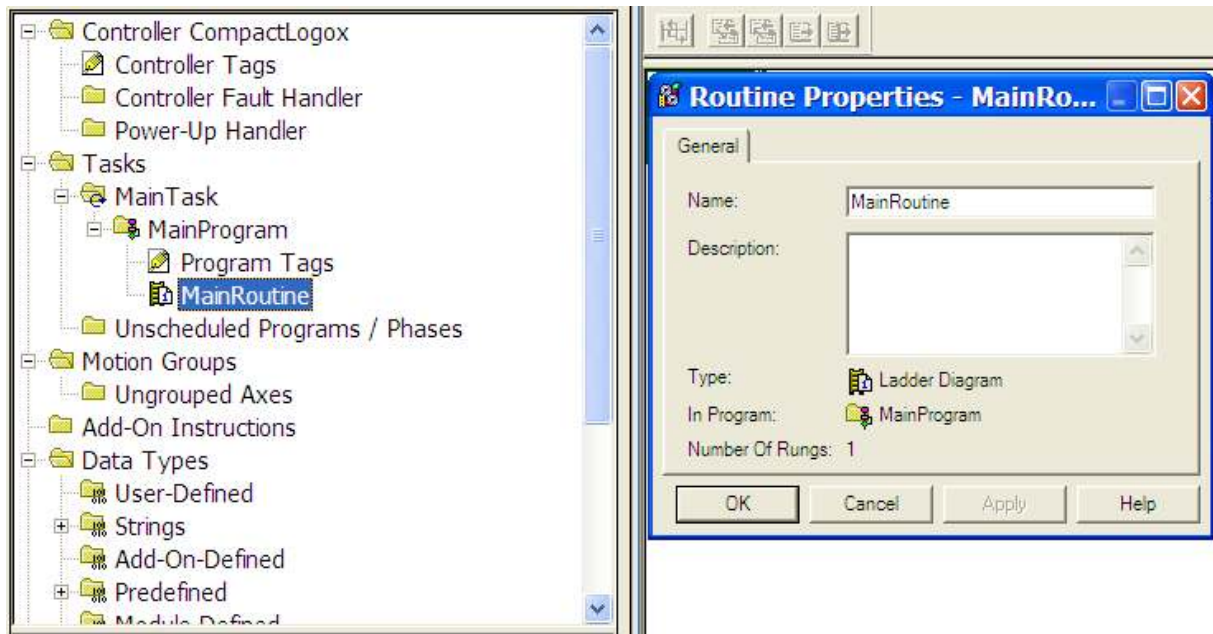


Nhấn **OK** khi hoàn tất công việc cài đặt.

4. Thiết lập Routine trong Program

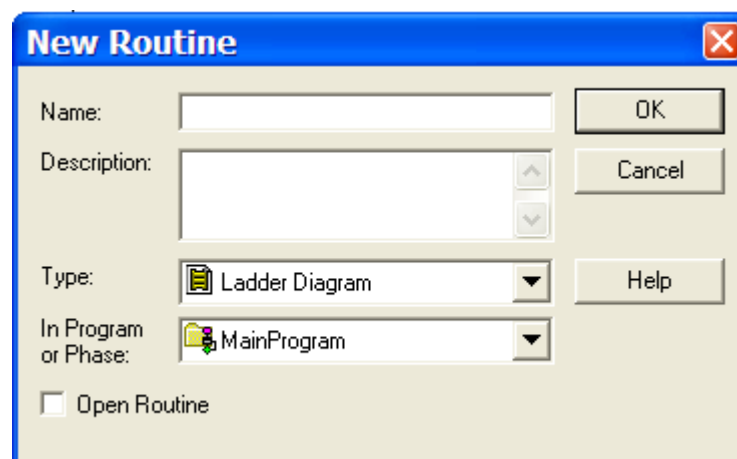
Chương trình con (Routine) là nơi viết chương trình cho bộ điều khiển, ta có thể khai báo các chương trình này hoạt động theo các kiểu sau:

Double Click vào **MainRoutine** để đặt tên cho chương trình trong ô **Name** và mô tả cho chương trình trong ô **Description**.



Trong **Routine Properties** ta có thể biết các thông tin của chương trình như Dạng ngôn ngữ lập trình, Routine nằm ở chương trình nào, hay số hàng trong chương trình.

Muốn thêm Routine ta click phải vào **Program**, sau đó chọn **New Routine...** hộp thoại sau xuất hiện cho phép đặt tên cho chương trình con trong ô **Name** và mô tả cho chương trình con trong ô **Description**.



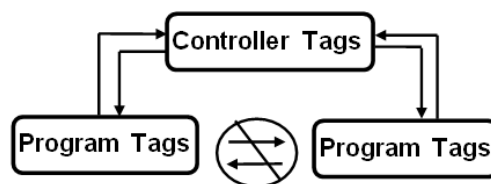
Nhấn **OK** khi hoàn tất công việc cài đặt.

5. Thiết lập Tags trong chương trình

Tags là tên của một vùng nhớ lưu trữ dữ liệu trong bộ điều khiển. Bộ điều khiển sử dụng các tags để điều khiển mà không cần truy cập trực tiếp tới địa chỉ vật lý của các module.

Có 2 loại Tags là Controller Tags và Program Tags

- ❖ Controller Tags là các Tags chứa các dữ liệu, các chương trình con trong toàn bộ chương trình có thể truy cập được.
- ❖ Program Tags là các Tags chứa dữ liệu mà chỉ có duy nhất chương trình con của chương trình đó mới truy cập được.



Có 4 kiểu Tags như sau:

- **Base Tag:** Là tag xác định vùng nhớ nơi dữ liệu được lưu trữ. Bộ điều khiển sử dụng 4 byte (DWORD) để lưu trữ dữ liệu.



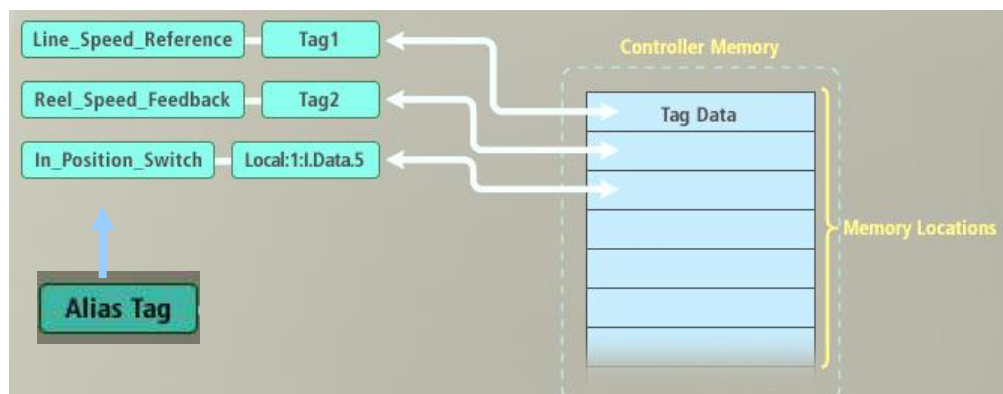
Location: Xác định vị trí cho một module.

Slot: Vị trí của module được gắn.

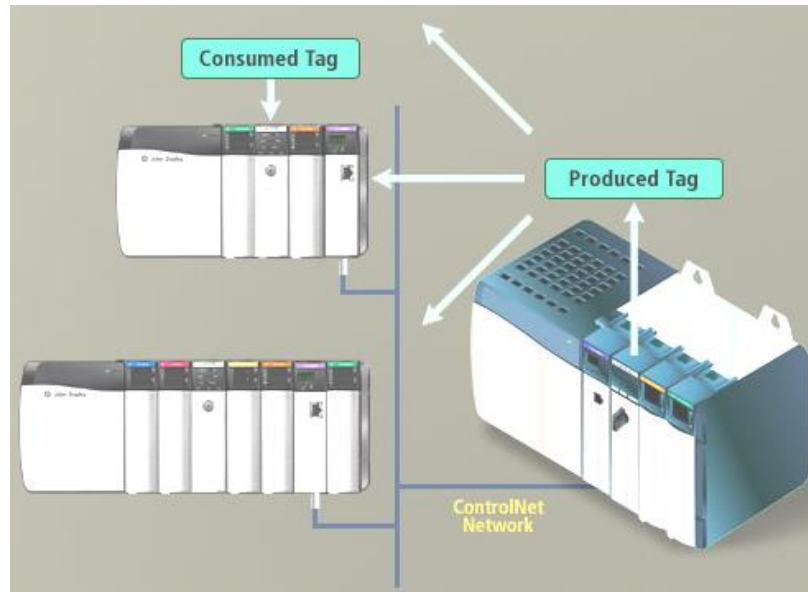
Type: Kiểu dữ liệu Input hay Output

Member: Dữ liệu.

- **Alias tag:** Là một tên gọi khác của một vùng nhớ đã được đặt tên. Alias tag có liên hệ với Base tag, Consumer tag và Producer tag.



- **Producer tag:** Là tag mà bộ điều khiển tạo ra cho một hay nhiều bộ điều khiển khác có thể sử dụng.
- **Consumed tag:** Là tag mà giá trị của nó được nhận từ điều khiển khác.



Các kiểu dữ liệu của Tags bao gồm: BOOL, SINT, INT, DINT và REAL. Đây là vùng nhớ lưu trữ dữ liệu của Tag.

Để tạo các Tags cho chương trình ta click phải chuột vào **Controller Tags** hoặc **Program Tags**, chọn **New Tag...**

Đặt tên cho Tags trong ô **Name** và mô tả Tags trong ô **Description**.

Chọn kiểu Tag trong ô **Type**.

Kiểu dữ liệu trong **Data Type**.

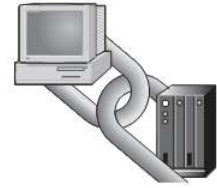
Kiểu hiển thị giá trị trong **Style**.

Nhấn **OK** khi hoàn tất công việc cài đặt.

Giới thiệu phần mềm RSLinx

1. Giới thiệu tổng quan về phần mềm RSLinx.

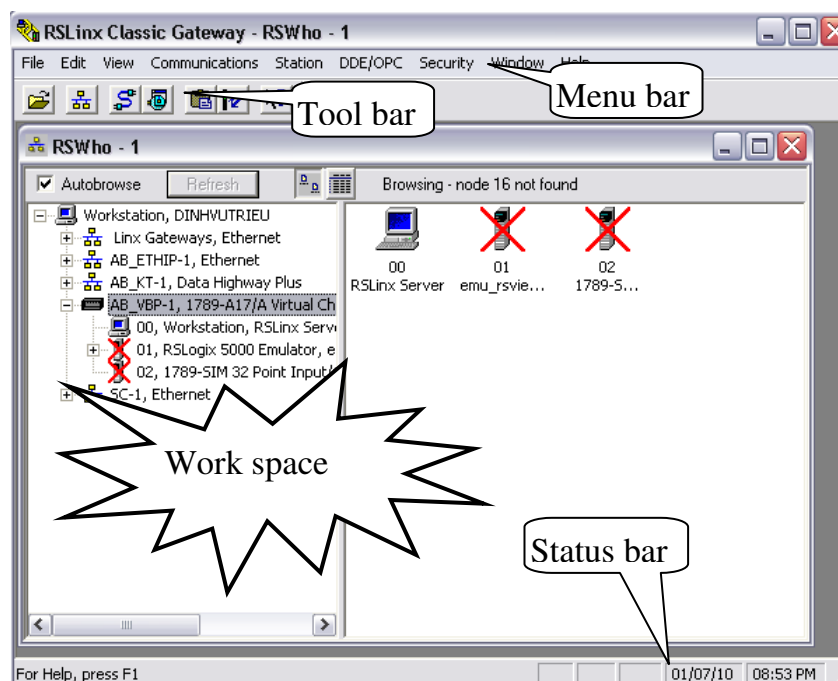
RSLinx là phần mềm do hãng Allen Bradley phát triển để giao tiếp với các thiết bị mà hãng sản xuất thông qua các phần mềm như: RS LogixTM 5/500/5000, RSView 32, FactoryTalk..., ngoài ra phần mềm này còn dùng để giao tiếp với các phần mềm của hãng thứ ba.



RSLinx Classic có thể hỗ trợ nhiều ứng dụng đồng thời, giao tiếp đến một loạt thiết bị trên các mạng khác nhau.

2. Giao diện và các nút chức năng của RSLinx.

Vì đây là phần mềm hỗ trợ kết nối nên có giao diện tương đối đơn giản, khi click vào biểu tượng của **RSWho** thì một giao diện có dạng 2 cửa sổ mở ra. Ở khung bên trái hiển thị hệ thống phân cấp của mạng lưới, ở khung bên phải hiển thị biểu tượng của thiết bị cùng với tình trạng của thiết bị.



Giao diện của phần mềm RSLinx Classic.

Menu bar:

Tập hợp hầu hết các tính năng của phần mềm RSLinx, gồm các tính năng sau:






Mô tả chức năng trong menu RSLinx Classic.



Tên chức năng trên Menu	Mô tả chức năng.
File	Tạo và mở một dự án RSLinx Classic
Edit	Copy liên kết của DDE và OPC vào Clipboard
View	Thiết lập và thay đổi cách hiển thị của chương trình.
Communications	Cấu hình điều khiển, các chủ đề, và các tùy chọn khác.
Station	Thực hiện việc giám sát các hành động.
DDE/OPC	Cấu hình DDE/OPC và chuẩn đoán thông tin.
Security	Thiết lập bảo mật của người dùng và quyền truy cập.
Window	Sắp xếp các cửa sổ của RSLinx.

Tool bar:

Truy cập nhanh đến các chức năng, bao gồm các biểu tượng cơ bản sau:

Chú thích các biểu tượng trong Toolbar RSLinx.

Biểu tượng	Truy cập từ Menu	Mô tả
	File > Open Project	Mở một project
	Communications → RSWho	Mở RS Who để xem các kết nối.
	Communications → Configure Drivers	Hiển thị cấu hình hiện tại của RSLinx, cho phép thêm cấu hình mới.
	Communications → Driver Diagnostics	Cung cấp thông tin chi tiết về trình điều khiển.
	Edit → Copy DDE/OPC Link	Copy liên kết DDE/OPC

	DDE/OPC → Topic Configuration	Cho phép tạo mới hay sửa đổi một DDE/OPC
	Help → What's This?	Hiển thị giúp đỡ khi đưa con trỏ đến vị trí bất kỳ.

Work space:

Hiển thị các cửa sổ con như RS Who hay các hộp thoại thông báo.

Startus bar:

Cung cấp thông tin về trạng thái làm việc của hệ thống.

Các chuẩn giao tiếp và phần mềm mà RSLinx hỗ trợ.

Ethernet to PLC-5 / SLC 5 / 5820-EI / 1756-ENET

1784-KTX(D) / PKTX(D) / PCMK / KT / KT2 to DH+

1784-KTX(D) / PKTX(D) / PCMK to DH-485

1784-KTCX to ControlNet

1784-PCC to ControlNet

1784-PCIC(S) to ControlNet

1784-PCD / PCIDS to DeviceNet (Windows NT)

RS-232 Serial (DF1) to 1770-KF2 / 1785-KE to DH+

RS-232 Serial (DF1) to 1770-KF3 / 1747-KE / 1747-PIC to DH-485

RS-232 Serial (DF1) to PLC-5, SLC-5, MicroLogix, PanelView

RS-232 Serial (DF1) to 1770-KFC to ControlNet

RS-232 Serial (DF1) to 1770-KFD to DeviceNet

RS-232 Serial (DF1) to ControlLogix

DF1 Polling Master (RS-232 serial)

DF1 Slave (RS-232 Serial)

SoftLogix 5 Controller

SoftLogix 5800 Controller.

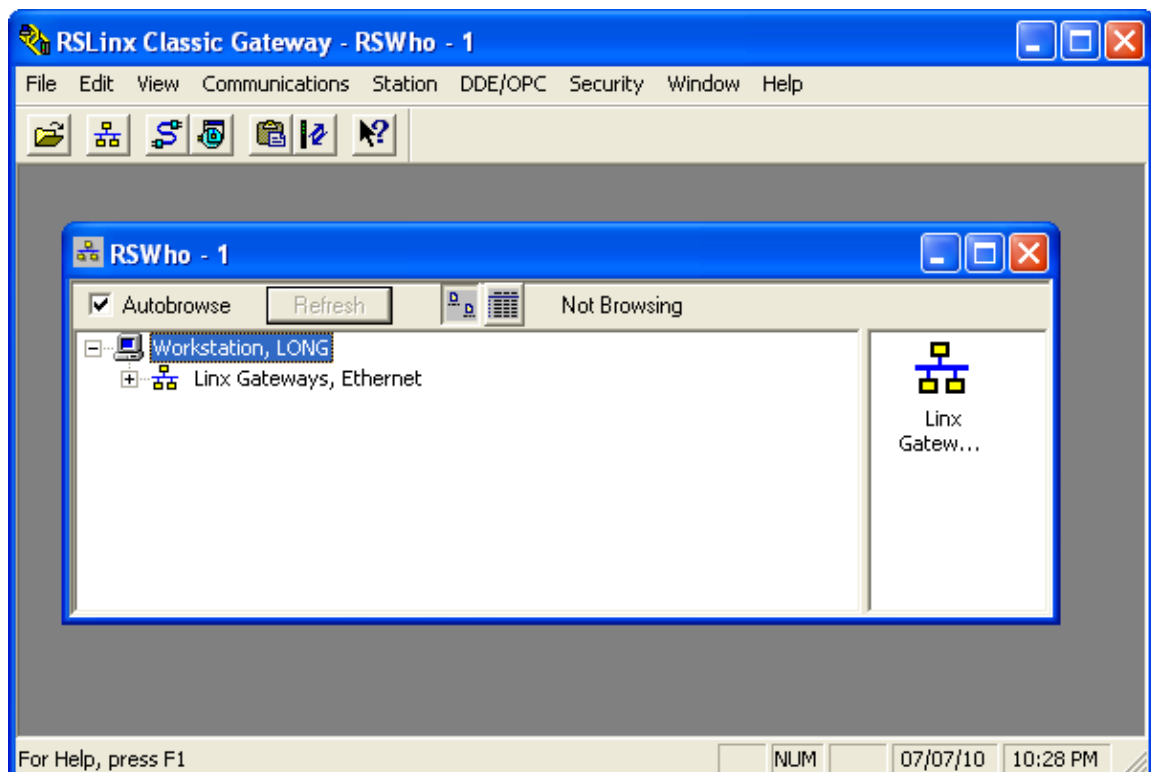
Các bài thực hành

Bài mở đầu:

Tạo và kiểm tra chương trình trong RSLogix 5000

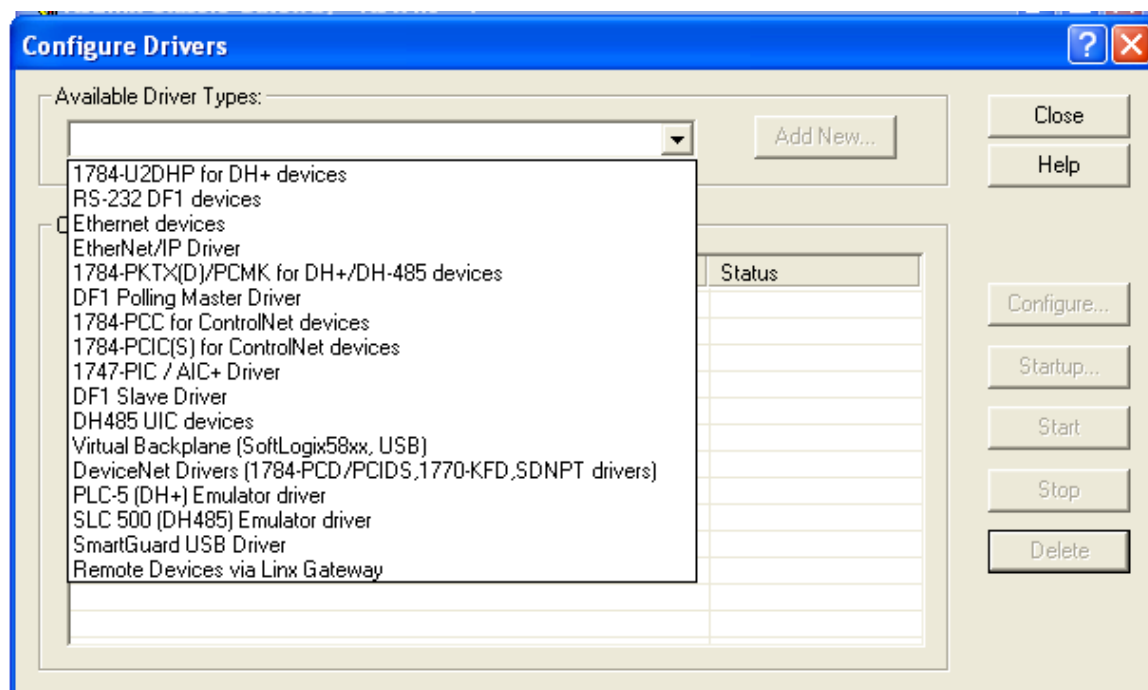
Bước 1: Khởi động phần mềm RSLinx.

- Vào “Start → Program → RockwellSoftware → RSLinx → RSLinx Classic. Phần mềm RSLinx là phần mềm dùng để giao tiếp giữa máy tính với bộ điều khiển.
- Từ thanh menu chính của RSLinx, chọn “Communications → RSWho”. Ta có được giao diện sau.



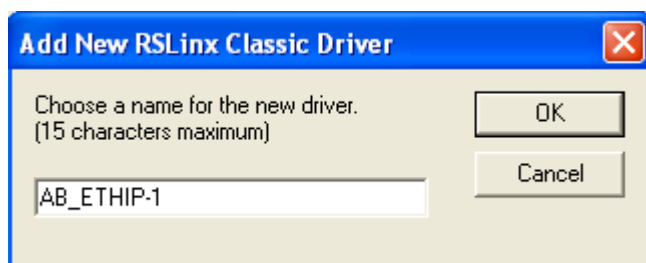
- Để liên kết giữa CompactLogix và máy tính ta làm như sau:

- 1_ Từ thanh menu, chọn “Communications → Configure Drivers”. Hộp thoại configure Drivers xuất hiện.
- 2_ Click vào mũi tên trong hộp thoại “Available Driver Types” để hiển thị danh sách các Driver giao tiếp.

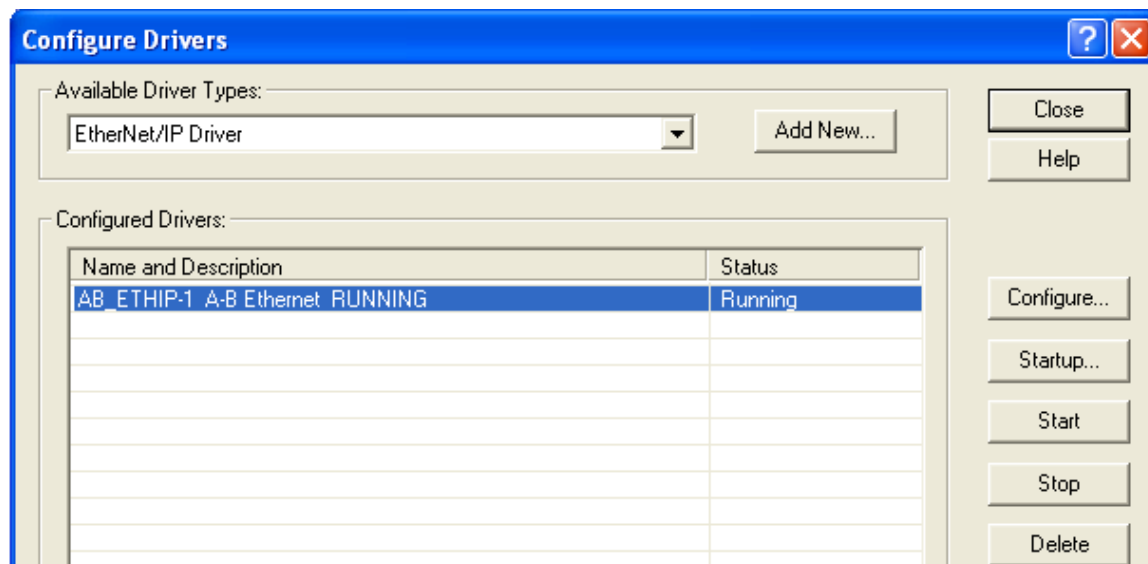


3_Chọn “EtherNet/IP Driver”

4_Click “Add New...”



5_Chọn OK. Màn hình sau xuất hiện



6_Trạng thái của Driver phải là “Running”

Lưu ý: “Configure Drivers” thường có các trạng thái sau:

ERROR – lỗi giao tiếp không được thực hiện do sai địa chỉ hoặc cấu hình sai.

STOPPED – lỗi kết nối phần cứng.

RUNNING – kết nối thành công.

Trước khi đóng cửa sổ “Configure driver” phải chắc chắn không có lỗi gì xảy ra. Click “Close” để tắt.

Tại đây ta có thêm vào các driver giao tiếp Ethernet/IP khác bằng cách chọn “Communication → Configure Driver” từ thanh menu.

Trên cửa sổ giao diện “Available Driver Types” sổ danh sách các driver, chọn Ethernet Device và Add New...

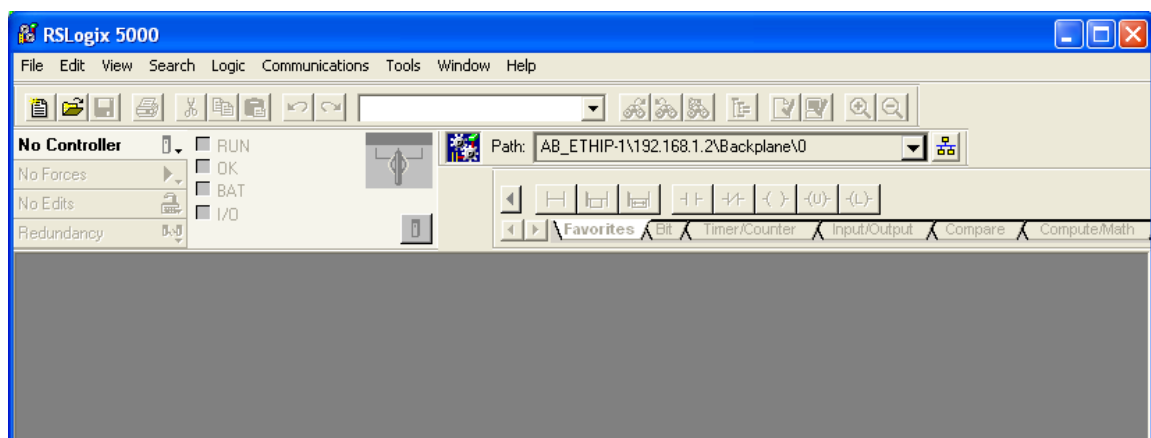
Chọn tên mặc định của driver bằng cách click “OK”.

Bước 2: Sử dụng phần mềm RSLogix 5000 để tạo mới một dự án.

Để trực quan hơn trong quá trình thực hành ta dựa vào yêu cầu cụ thể sau:

“Viết chương trình điều khiển khởi động trực tiếp động cơ KĐB 3 pha bằng 2 nút nhấn On và Off, bảo vệ động cơ bằng relay nhiệt. Có đèn báo chế độ làm việc và sự cố.”

- Để bắt đầu ta tiến hành khởi động phần mềm RSLogix 5000 bằng cách “Start → Program → RockwellSoftware → RSLogix5000” sẽ xuất hiện màn hình sau:



- Vào “File” từ thanh menu chính, chọn “New” tiếp theo thực hiện các bước sau:

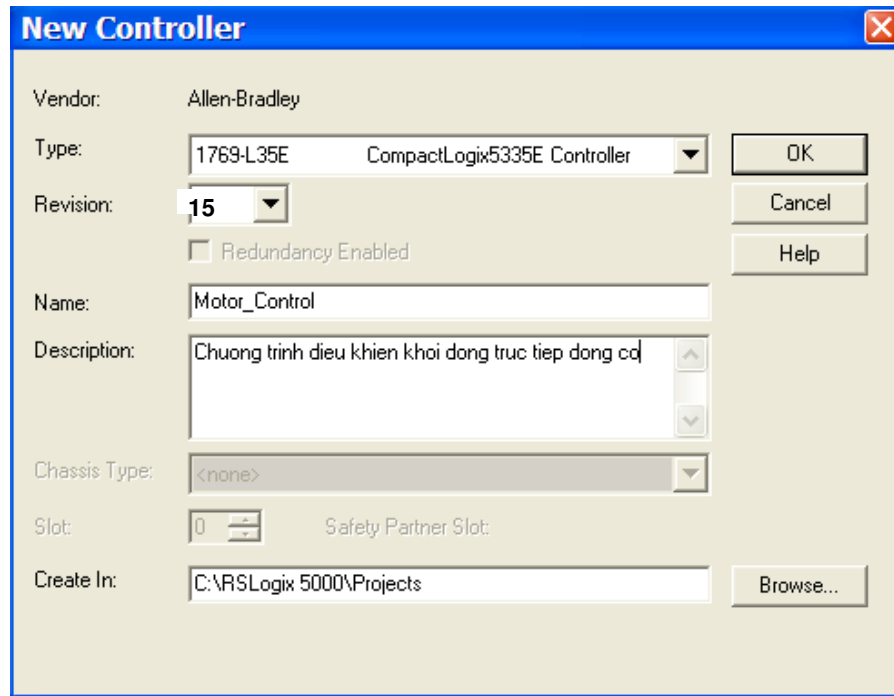
1_ Chọn tên bộ điều khiển 1769-L35E.

2_ Trong hộp thoại revision, ta chọn 15.

3_ Đặt tên cho project là “Motor Control” và có thể thêm vào chú thích cho project của bạn ở bên dưới.

4_ Trong hộp thoại “Create in”, nhấn “Browse để chọn địa chỉ lưu project.

Ta được như hình sau:



New Controller

Vendor: Allen-Bradley

Type: 1769-L35E CompactLogix5335E Controller

Revision: 15

☐ Redundancy Enabled

Name: Motor_Control

Description: Chương trình điều khiển khởi động trực tiếp động cơ

Chassis Type: <none>

Slot: 0 Safety Partner Slot:

Create In: C:\RSLogix 5000\Projects

Buttons: OK, Cancel, Help, Browse...

Nhấn OK để hoàn tất việc khai báo cho bộ điều khiển.

- Tiếp theo ta sẽ xác định địa chỉ của các module I/O trong một chassis mà ta đang dùng. Trong Folder **I/O Configuration** ở phần **Project Tree**, click phải chuột vào **Compact Bus Local trong Backplane, CompactLogix System** chọn **Insert Modul...**

Lần lượt Insert vào các modul như sau:

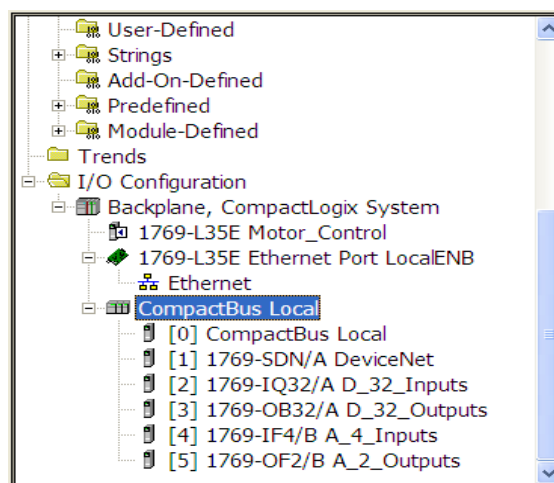
Modul DeviceNet 1769-SDN/A

Modul ra số 1769-IQ32/A

Modul vào tương tự 1769-IF4/B

Modul ra tương tự 1769-OF2/B

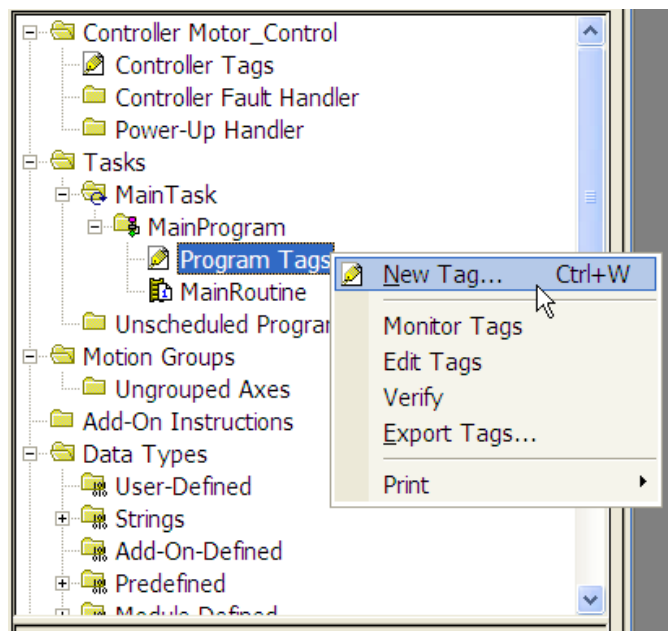
Sau khi chọn loại Modul đúng với mã hiệu trên modul thực tế, ta đặt tên Modul trong ô Name, Mô tả modul trong Description chọn Slot tương ứng trên Chassis, cuối cùng ta có đầy đủ các modul trong hệ thống như hình sau:



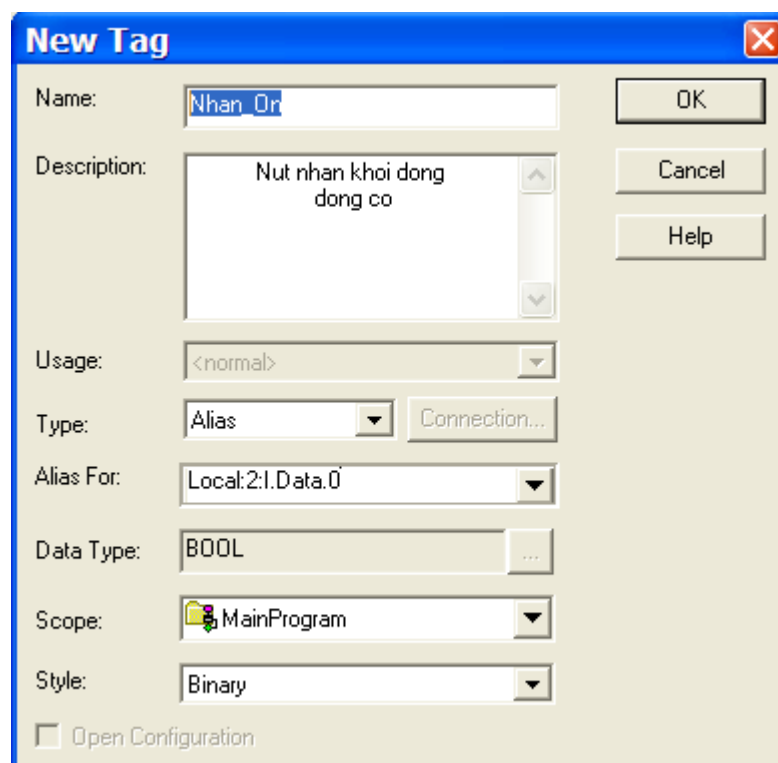
Lưu ý: Một ưu điểm lớn của PLC AB là mặc dù những modul đã kết nối với PLC nhưng nếu không sử dụng thì ta không cần phải khai báo.

- Tiếp theo ta khai báo Tags cho chương trình.

Trong mục Project Tree ta click phải vào Program Tag, chọn New tag... như hình sau:



Khai báo Tag có tên là **Nhan_On** như hình sau. Trong đó Local:2:I.Data.0 là ngõ vào thứ nhất của modul số 32 ngõ vào.



Tương tự cho nút Nhan_Off, Relay_Nhiet, Motor, Den_Run và Den_Su_Co như bảng sau:

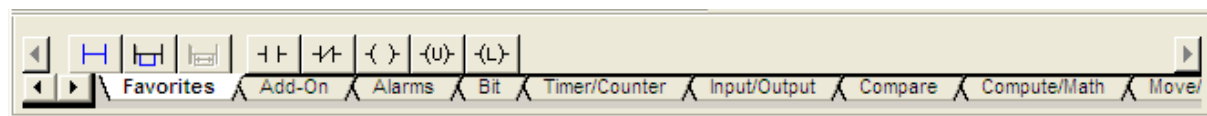
Name	Alias For	Base Tag	Data Typ	Style	Description
Nhan_Off	Local:2:I.Data.1(C)	Local:2:I.Data.1(C)	BOOL	Binary	Nut nhan dung dong co
Nhan_On	Local:2:I.Data.0(C)	Local:2:I.Data.0(C)	BOOL	Binary	Nut nhan khai dong dong co
Relay_Nhiet	Local:2:I.Data.2(C)	Local:2:I.Data.2(C)	BOOL	Binary	Tiep diem role nhiet
Motor	Local:3:O.Data.0(C)	Local:3:O.Data.0(C)	BOOL	Binary	Ngo ra dieu khien dong co
Den_Run	Local:3:O.Data.1(C)	Local:3:O.Data.1(C)	BOOL	Binary	Den bao DC chay
Den_Su_co	Local:3:O.Data.2(C)	Local:3:O.Data.2(C)	BOOL	Binary	Den bao khi bi su co qua tai

Nhấn **Save** để lưu lại các Tags đã khai báo.

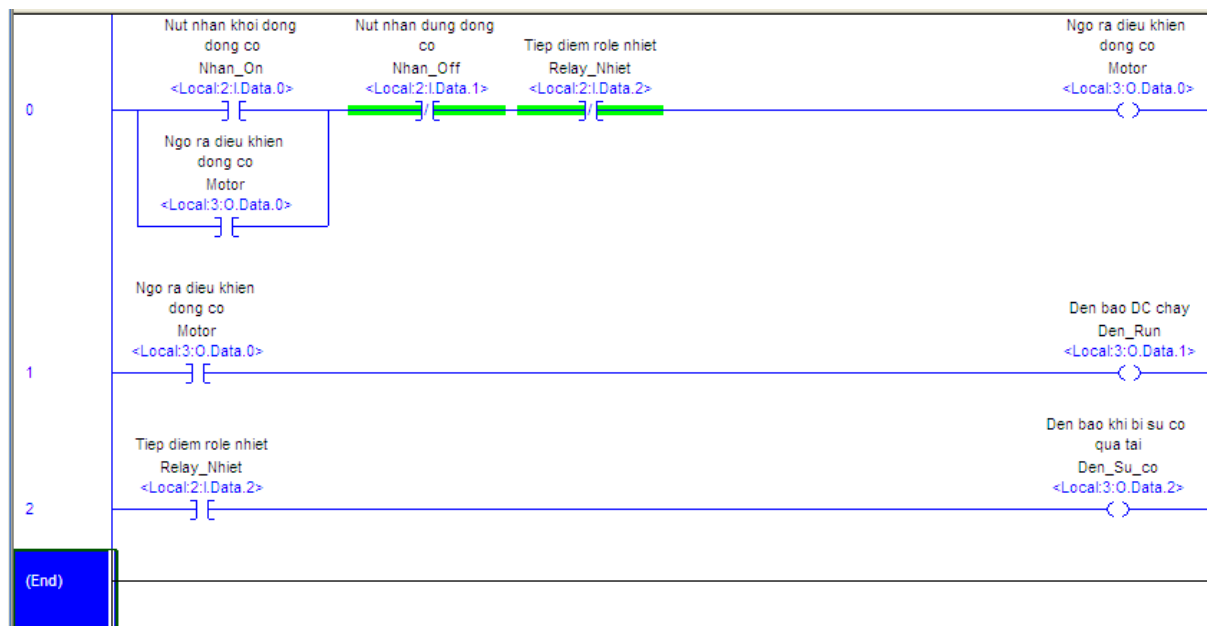
- Viết chương trình theo yêu cầu trên.


Click chuột vào **MainRoutine** để vào màn hình lập trình.

Tập lệnh của RSLogix 5000 được cho ở các nhóm lệnh. Ta có thể lấy các lệnh vào/ra cơ bản trong thẻ **Favorites** hoặc **Bit** như hình sau.



Thực hiện việc kéo – thả các tiếp điểm cần thiết vào màn hình lập trình và tiến hành gán giá trị các tiếp điểm cho các Tags vừa khai báo ta được chương trình như sau:



- Để rẽ nhánh chương trình ta chọn biểu tượng  rồi kéo vào vị trí cần chèn.

- Để thêm 1 Rung (hàng) ta đặt chuột vài vị trí hàng cần chèn rồi bấm vào biểu tượng



- Để chọn Tag gán cho tiếp điểm ta Click đúp chuột vào khung màu xanh của tiếp điểm rồi Click vào mũi tên để hiện danh sách các Tags, sau đó chọn Tag cần gán vào. Hoặc gõ tên Tag cần gán vào.

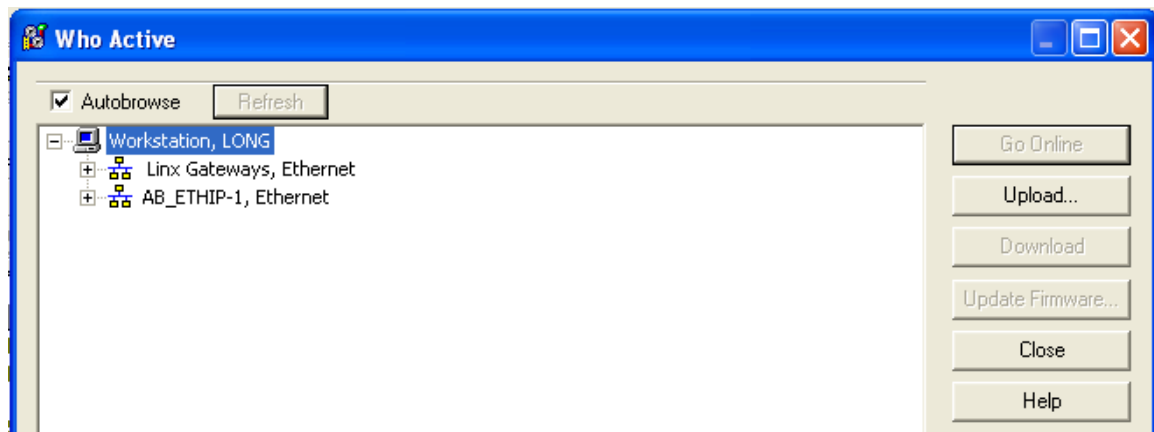
- Để kiểm tra lỗi ta Click vào biểu tượng **Verify Routine** hoặc vào **Logic => Verify => Routine**.

Bước 3: Download chương trình

Sau khi đã hoàn tất chương trình ta tiến hành download xuống bộ điều khiển.

1_Click vào “Communications” từ menu chính.

2_Chọn “Who Active” màn hình sau xuất hiện:



- Click vào “+” trước “AB_ETHIP-1, EtherNet” để chọn driver giao tiếp và bộ điều khiển.

- Click “+” trước địa chỉ của bộ điều khiển để mở rộng chúng. Ví dụ: “192.168.1.1...”

- Click “+” trước Backplane để mở rộng thư mục.

- Chúng ta đang chọn mạng điều khiển giao tiếp là EtherNet thông qua module 1769-L35E. Cuối cùng ta chọn bộ điều khiển trong Chassis để download chương trình.

- Nhấn Download, nhấn OK.

- Khi download thành công ,chương trình sẽ chuyển sang chế độ online.

Nhìn vào biểu tượng  để biết trạng thái đang online.

Bước 4: Thực hiện kiểm tra chương trình

Để kiểm tra chương trình ta có 2 cách:

- Thay đổi trạng thái của ngõ vào bằng cách tác động các công tắc trên Kit thực hành để kiểm tra trạng thái ngõ ra trên các đèn.



- Dùng chức năng Force trong chương trình. Trường hợp này gọi là kích mềm.

Click vào mũi tên “No Force” → I/O Forcing → Enable All... → chọn Yes.

Click phải lên tiếp điểm cần Force → chọn Force On hay Force Off.

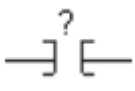
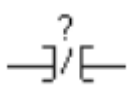
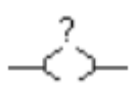
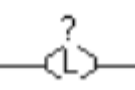
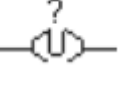
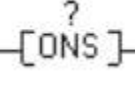
Để tắt chế độ Force ta Click vào mũi tên “No Force” → I/O Forcing → Remove...



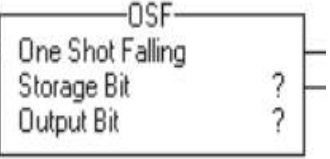
Ta vừa thực hiện hoàn chỉnh 1 chương trình trên CompactLogix. Tiếp theo đây sẽ là những bài thực hành áp dụng các lệnh khác của RSLogix.

Bài thực hành 1:

Sử dụng các lệnh vào/ra trong RSLogix 5000

Các lệnh vào/ra cơ bản.

Tên lệnh	Chức năng	Ngôn ngữ lập trình
XIC 	Tiếp điểm thường mở sẽ đóng khi bit được set.	Relay ladder Structured Text
XIO 	Tiếp điểm thường đóng sẽ mở khi bit được set.	Relay ladder Structured Text
OTE 	Cuộn dây ngõ ra.	Relay ladder Structured Text
OTL 	Lệnh set ngõ ra.	Relay ladder Structured Text
OUT 	Lệnh reset ngõ ra	Relay ladder Structured Text
ONS 	Ngõ ra được kích hoạt tại một lần quét khi giá trị của hàng đó (Ladder) là đúng.	Relay ladder Structured Text

	 <p>Tác động cạnh lên.</p>	Relay ladder
<p>OSF</p> 	<p>Tác cạnh xuống.</p>	Relay ladder Function Block

Bài tập 1:

Viết chương trình điều khiển đảo chiều quay trực tiếp động cơ AC bằng 3 nút nhấn theo yêu cầu sau bằng 2 cách:

Nhấn nút T : Động cơ chạy thuận.

Nhấn nút N : Động cơ chạy nghịch.

Nhấn nút D : Dừng động cơ.

Đèn RUN : Báo động cơ đang chạy.

Đèn ALARM : Báo xảy ra sự cố.

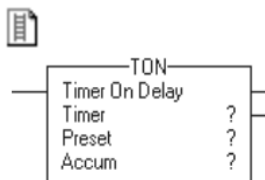
Cách 1: Sử dụng lệnh cuộn dây ngõ ra.

Cách 2: Sử dụng lệnh Set ,Reset.

Bài thực hành 2:

Sử dụng các lệnh Timer trong RSLogix 5000

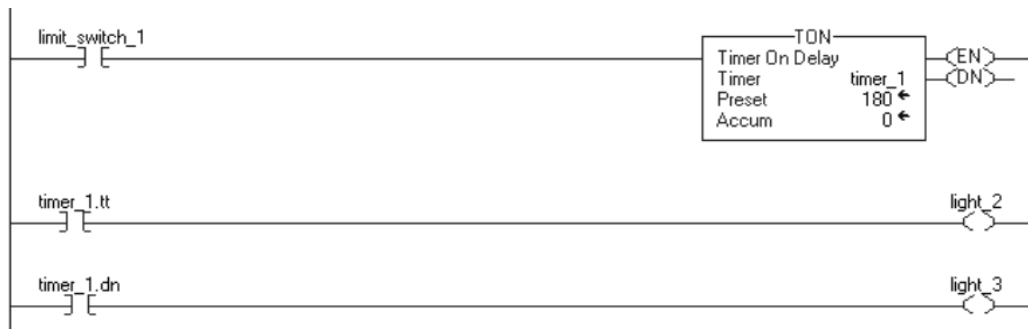
Có 6 loại Timer là TON, TOF, RTO, TONR, TOFR, RTOR và 4 loại Counter và RES dùng để reset Timer. Tất cả các loại Timer đều sử dụng ngôn ngữ ladder. Ngoài ra còn có phần gợi nhớ để thực hiện các chức năng của nó.

Cấu tạo Timer (ladder)				
	Toán hạng	Kiểu dữ liệu	Định dạng	Mô tả
	Timer	TIMER	Tag	Cấu trúc của bộ đếm
	Preset	DINT	Immediate	Khoảng thời gian trễ
	Accum	DINT	Immediate	Tổng số m s mà Timer đã đếm

Phần gợi nhớ của Timer.

Phần gợi nhớ	Kiểu dữ liệu	Mô tả
.EN	BOOL	Bit sẽ được set khi Timer có điện
.TT	BOOL	Bit sẽ set đúng thời gian đã định trước khi Timer có điện
.DN	BOOL	Giá trị của tiếp điểm sẽ đổi trạng thái khi Timer bắt đầu đếm
.PRE	DINT	Giá trị thời gian được đặt trước vốn đặt trước giá trị của bit .DN
.ACC	DINT	Lưu trữ số giây quét qua Timer

Ví dụ lệnh Timer (TON).



Khi bit limit_switch_1 đóng, ngõ ra light_2 bắt đầu sáng 180ms (timer_1 bắt đầu đếm). Khi timer_1.acc đếm đến 180ms thì light_2 sẽ tắt, light_3 sáng.

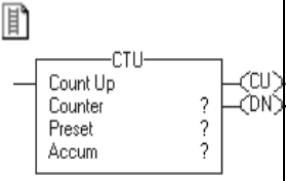
Bài tập: Viết chương trình điều khiển đèn giao thông ưu tiên cho người đi bộ như sau:

- Bình thường Đèn Xanh 1 và đèn Đỏ 2 sáng.
- Nếu có người muốn qua đường thì nhấn GO đèn Xanh 1 tắt chuyển sang đèn Vàng 1 sáng 5s, sau đó sáng đèn Đỏ 1 và đèn Xanh 2 sáng cho phép người đi qua đường trong thời gian 30s rồi chuyển sang sáng đèn Xanh 1 và đèn Đỏ 2. Chu trình lặp lại cho đến khi có nhấn tiếp nút GO.

Bài thực hành 3:

Sử dụng các lệnh Counter trong RSLogix 5000

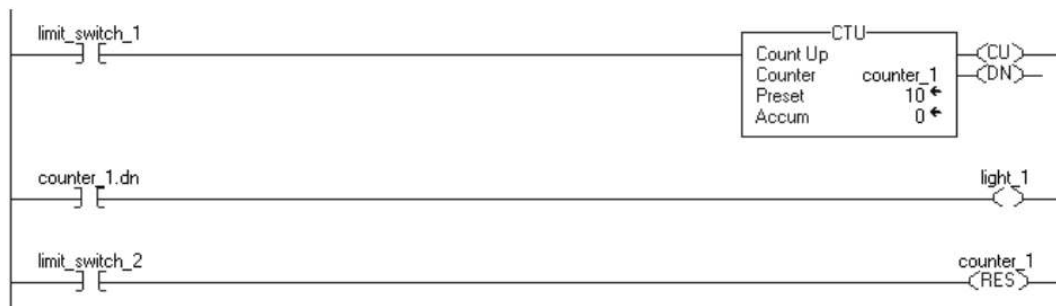
Cấu tạo Counter (ladder)

	Toán hạng	Kiểu dữ liệu	Định dạng	Mô tả
	Counter	COUNTER	Tag	Cấu trúc của bộ đếm
	Preset	DINT	Immediate	Giá trị cần đếm
	Accum	DINT	Immediate	Giá trị theo thời gian của bộ đếm.

Phần gợi nhớ của Counter.

Phần gợi nhớ	Kiểu dữ liệu	Mô tả
.CU	BOOL	Bit được đếm lên khi Counter có điện.
.DN	BOOL	Bit hoàn thành khi .ACC > .PRE
.OV	BOOL	Giá trị của Counter vượt qua ngưỡng trên của 2, 147, 483, 647. Sau đó Counter sẽ bắt đầu đếm lại.
.UN	BOOL	Giá trị của Counter ở dưới mức -2, 147, 483, 648. Sau khi vượt qua 2, 147, 483, 647 Counter sẽ bắt đầu đếm lại
.PRE	DINT	Định rõ giá trị đặt trước, giá trị này phải được lưu trước khi set giá trị của bit .DN
.ACC	DINT	Ghi lại giá trị đếm của Counter.

Ví dụ lệnh Counter đếm lên (CTU).



Sau khi bit limit_switch_1 đếm được 10 lần, phần bit mở rộng .DN sẽ được bật (counter_1.dn được set lên 1), light_1 sáng cho đến khi limit_switch_2 đóng thì counter_1 bị reset. Quá trình sẽ bắt đầu lại từ đầu khi limit_switch_1 thực hiện việc đóng mở 10 lần.

Bài tập: Viết chương trình điều khiển băng tải theo yêu cầu sau:

- Nhấn On: Nếu cảm biến 1 phát hiện có sản phẩm vào thì cho băng tải chạy. Khi đến vị trí cảm biến 2 phát hiện sản phẩm thì cho băng tải dừng sau 1s nếu cảm biến 3 phát hiện thì cho kích van tác động loại sản phẩm, nếu cảm biến 3 không phát hiện thì cho băng tải chạy tiếp. đếm số lượng mỗi loại sản phẩm.
- Nhấn Off: Dừng băng tải.

Bài thực hành 4:

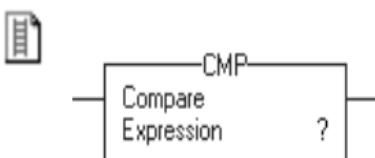
Sử dụng các lệnh so sánh trong RSLogix 5000

Lệnh so sánh cho phép so sánh 1 giá trị bằng cách sử dụng 1 biểu thức hay 1 lệnh so sánh đặc trưng riêng.

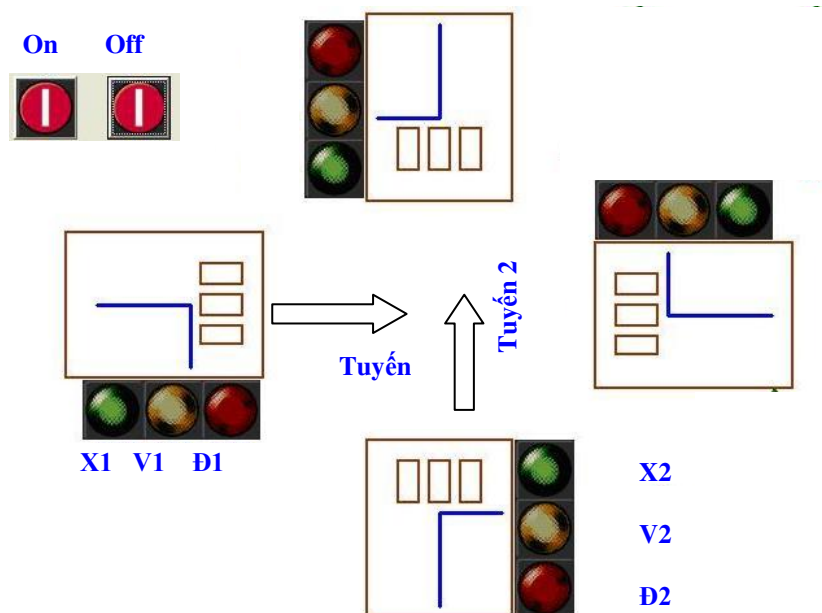
Các lệnh so sánh.

Tên lệnh	Chức năng	Ngôn ngữ lập trình.
CMP (compare)	So sánh giá trị dựa vào một biểu thức	Relay ladder Structured Text
EQU (Equal)	So sánh hai giá trị bằng nhau	Relay ladder Structured Text Function Block
GEQ (Greater Than or Equal)	So sánh 1 giá trị lớn hơn hay bằng 1 giá trị khác	Relay ladder Structured Text Function Block
GRT (Greater than)	So sánh 1 giá trị lớn hơn 1 giá trị khác	Relay ladder Structured Text
LEQ (Less Than or Equal)	So sánh 1 giá trị nhỏ hơn hay bằng 1 giá trị khác.	Relay ladder Structured Text Function Block
LES (Less Than)	So sánh 1 giá trị nhỏ hơn 1 giá trị khác.	Relay ladder Structured Text
LIM (Limit Test)	So sánh 1 giá trị nằm giữa 2 giá trị.	Relay ladder Structured Text Function Block
NEQ (Not Equal)	So sánh 1 giá trị không bằng 1 giá trị khác.	Relay ladder Structured Text

Lệnh CMP (Compare).

	Toán hạng	Kiểu dữ liệu	Định dạng
	Biểu thức	SINT INT, DINT REAL, String ASINT hay INT chuyển sang DINT	Immediate Tag

Bài tập: Viết chương trình điều khiển đèn giao thông ở ngã tư 3 với thời gian sáng mỗi đèn như sau: Đèn Xanh sáng 8s, Đèn Vàng sáng 2s, Đèn Đỏ sáng 10s.



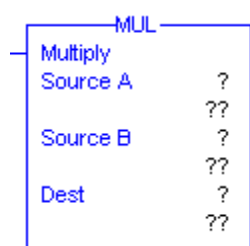
Bài thực hành 5:

Sử dụng các lệnh toán học trong RSLogix 5000

Các dạng toán hạng.

Toán hạng	Mô tả	Kiểu dữ liệu	Toán hạng	Mô tả	Kiểu dữ liệu
+, -, *, /		DINT, REAL	FRD	BCD sang Integer	DINT
=, <, >		DINT, REAL	OR	OR	DINT
**	X mũ Y	DINT, REAL	TRN	Cắt bỏ	DINT, REAL
<=, >=		DINT, REAL	SQR	Căn bậc 2	DINT, REAL
ABS	Trị tuyệt đối	DINT, REAL	AND	AND	DINT, REAL
ADD, SUB	Cộng, trừ	DINT, REAL	ACS	ArcCosin	REAL
MUL, DIV	Nhân, chia	DINT, REAL	:	:	:
AND	AND	DINT, REAL			
ACS	Arc Cosin	REAL			

Ví dụ Lệnh MUL:



Source A, Source B: là số nhân, định dạng số (giá trị cần nhân) hoặc Tag.

Dest: kết quả của phép nhân, định dạng Tag.

Dấu ?? sẽ hiển thị kết quả của phép nhân.

Bài thực hành 6:

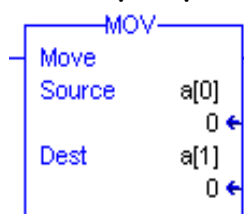
Sử dụng các lệnh di chuyển dữ liệu trong RSLogix 5000

Lệnh di chuyển có thể tổng hợp dữ liệu nhưng có thể làm mất độ chính xác của dữ liệu, lỗi có thể xảy ra và các lệnh phải mất nhiều thời gian hơn để thực hiện. Khi sử dụng dữ liệu trong kiểu lập trình ladder, các lệnh nên dùng kiểu Bool, các lệnh sẽ thực hiện nhanh hơn và chiếm ít bộ nhớ hơn kiểu DINT, REAL.

Các lệnh di chuyển dữ liệu.

Tên lệnh	Chức năng	Ngôn ngữ lập trình
MOV	Di chuyển 1 đại lượng	Relay ladder Structured Text
MVM	Di chuyển 1 số nguyên	Relay ladder
MVMT	Di chuyển 1 số nguyên trong sơ đồ khối	Relay ladder Function Block
BTB	Di chuyển bit trong 1 số nguyên hay giữa 2 số nguyên	Relay ladder
BTBT	Di chuyển bit trong 1 số nguyên hay giữa 2 số nguyên trong sơ đồ khối	Relay ladder Function Block
CLR	Xóa giá trị	Relay ladder
SWPB	Sắp xếp lại giá trị của Byte	Relay ladder, Structure

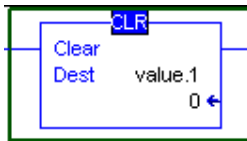
Ví dụ: Lệnh MOV (move):



Cho phép di chuyển dữ liệu từ nguồn đến đích, nguồn trong lệnh này không thay đổi.

Source: Kiểu dữ liệu DINT, REAL, định dạng số (cần di chuyển) hoặc Tag.

Dest: Địa chỉ của Tag cần lưu dữ liệu, kiểu DINT, REAL, định dạng Tag.



Lệnh CLR (Clear):

Lệnh CLR xóa dữ liệu trong địa chỉ của Dest.

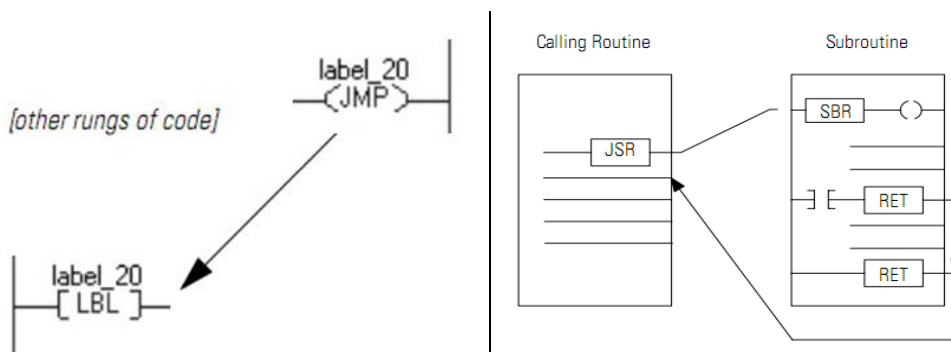
Bài thực hành 7:

Sử dụng các lệnh nhảy trong RSLogix 5000

Lệnh nhảy tới nhãn JMP.

Khi kích hoạt lệnh JMP, chương trình sẽ nhảy tới nhãn LBL và tiếp tục thực hiện chương trình tại vị trí này. Khi tắt lệnh này thì nó không còn ảnh hưởng đến chương trình nữa. Lệnh JMP có thể di chuyển sự thi hành của các lệnh đến trước hoặc sau. Nhảy về phía trước của nhãn là bỏ qua 1 vài lần quét qua chương trình, còn nhảy về phía sau là sự cho phép chương trình lặp lại các thuật toán logic. Nên thận trọng khi sử dụng lệnh nhảy về phía sau vì sẽ xảy ra tình trạng là số lần đếm sẽ nhiều, bộ định thời gian sẽ bị tràn vì chương trình sẽ không bao giờ kết thúc.

Ví dụ: Khi lệnh JMP được kích hoạt, lệnh này sẽ thực hiện việc nhảy đến vị trí của lệnh LBL với tên nhãn được đặt trong ví dụ này là label_20:



Lệnh nhảy tới nhãn và nhảy tới chương trình con.

Lệnh JSR thực hiện việc nhảy đến 1 chương trình con. Lệnh SBR và lệnh trở lại RET là những lệnh không bắt buộc mà dẫn đến việc thay đổi dữ liệu.

Bài tập: Viết chương trình điều khiển động cơ hoạt động theo 2 chế độ dùng 2 chương trình con theo yêu cầu sau: Nhấn PB1: Động cơ chạy sau 20s thì dừng.

Nhấn PB2 : Động cơ chạy sau 30s thì dừng. Nhấn STOP: Động cơ dừng.