PHỤ LỤC

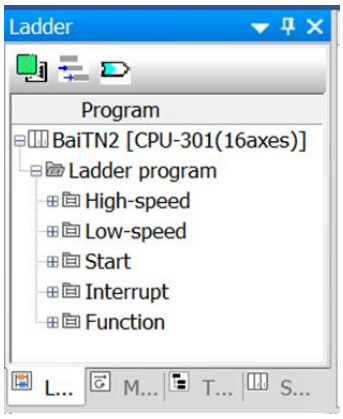
HƯỚNG DẪN LẬP TRÌNH MACHINE CONTROLLER TRÊN PHẦN MỀM MPE720

1. GIỚI THIỆU

MPE720 là phần mềm chuyên dụng để viết, biên dịch và nạp chương trình cho Controller (Trong phòng thí nghiệm Yaskawa Mechatro là Machine Controller MP3000 16 axes).

Có hai ngôn ngữ để viết chương trình: Ladder và Motion. Trong đó, viết Ladder cũng tương tự giống với các viết chương trình cho PLC của các hãng; còn ngôn ngữ Motion thì có các tập lệnh đặc biệt của Yaskawa, cho phép người dùng dễ dàng điều khiển chuyển động cho các động cơ servo.

2. HƯỚNG DẪN VIẾT CHƯƠNG TRÌNH LADDER



Trong tab ***Ladder***:

+ ***High-speed***: Chương trình thực thi sau mỗi lần quét tốc độ cao.

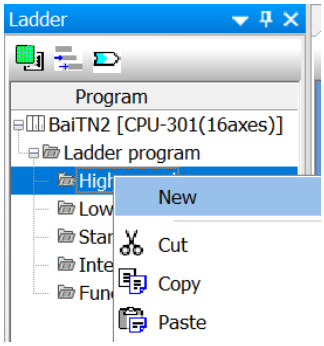
+ ***Low-speed***: Chương trình thực thi sau mỗi lần quét tốc độ thấp.

+ ***Start***: Chương trình thực thi sau mỗi lần bật điện controller.

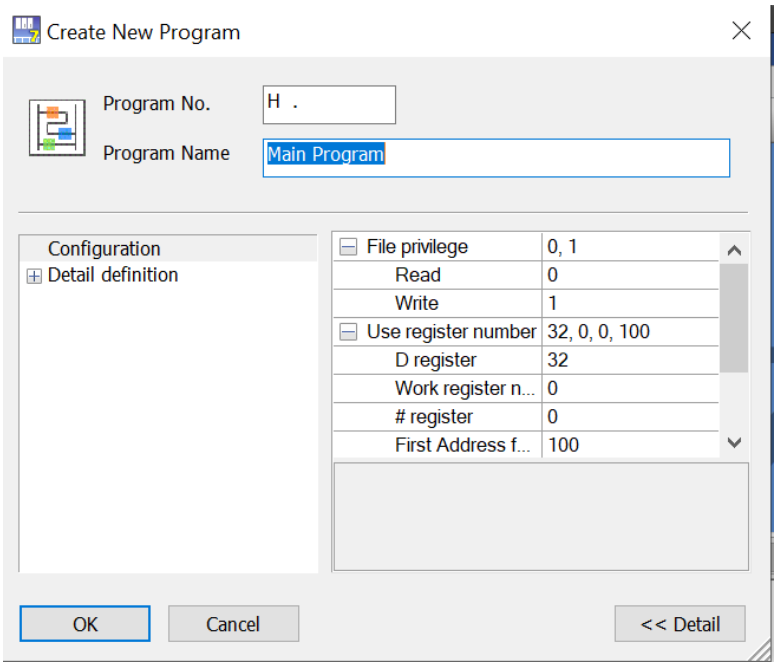
+ ***Interrupt:*** Chương trình thực thi sau mỗi lần có tín hiệu interrupt.

+ ***Function***: Nơi viết các hàm. Để chạy các hàm này, gọi nó bằng hàm SEE trong các nơi chương trình được viết ở phần trên.

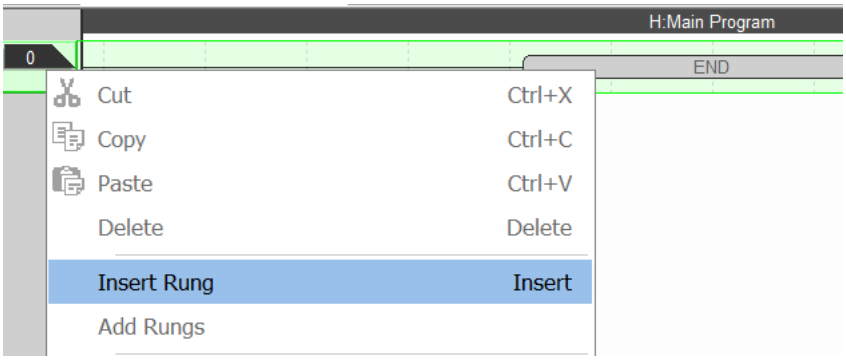
Nhấp chuột phải vào phần muốn viết chương trình, chọn ***New.***



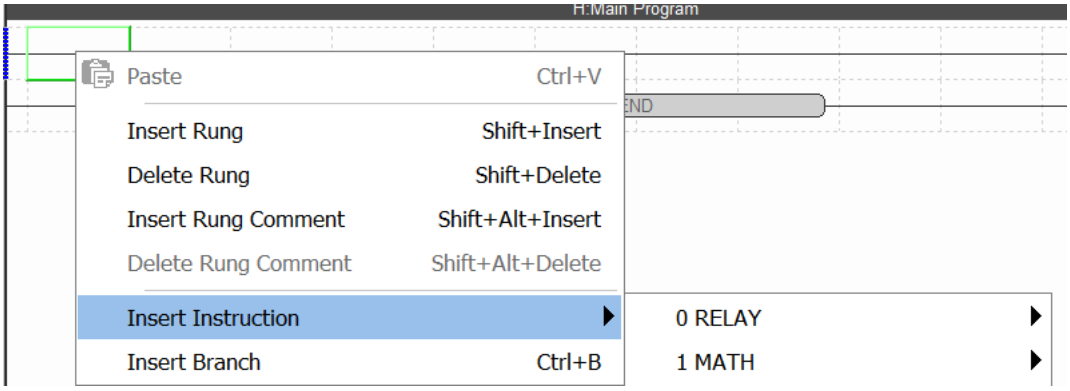
Sau đó đặt tên cho chương trình và chọn ***OK***.



Nhấp chuột phải vào Rung số 0, chọn ***Insert Rung*** để có thể bắt đầu viết chương trình Ladder.

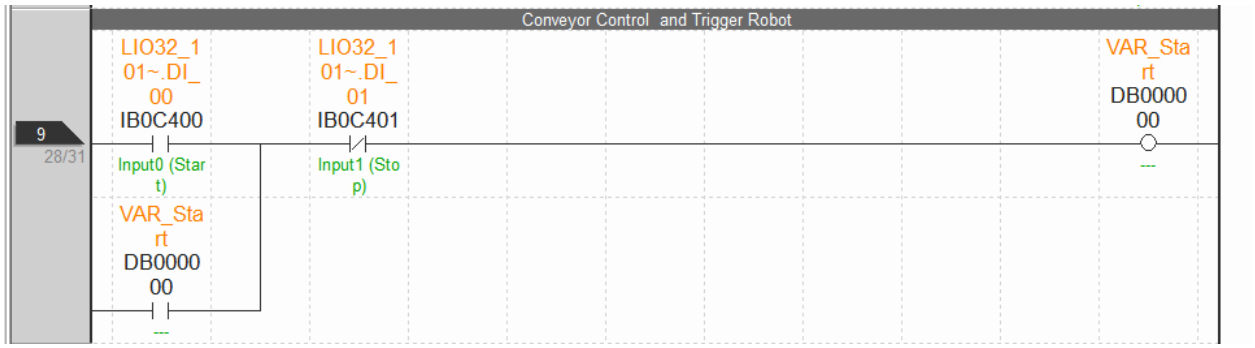


Một Rung mới được tạo, chuột phải vào giữa Rung, chọn ***Insert Instruction***, từ đó chọn các lệnh mong muốn.



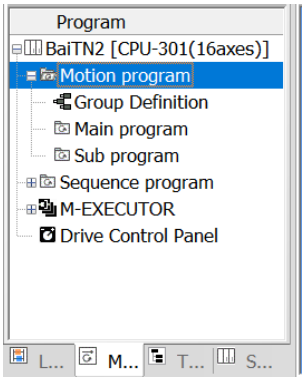
Hoặc có thể chọn nhanh các lệnh trên thanh công cụ:

Ví dụ: Chương trình kích I/O để bật/tắt động cơ băng tải:

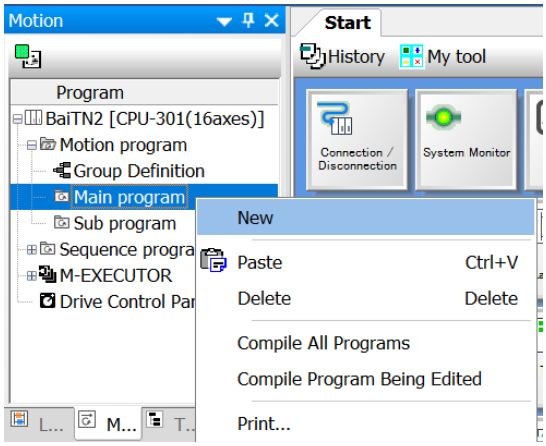


Sau khi viết xong chương trình, nhấn **F4** để complile.

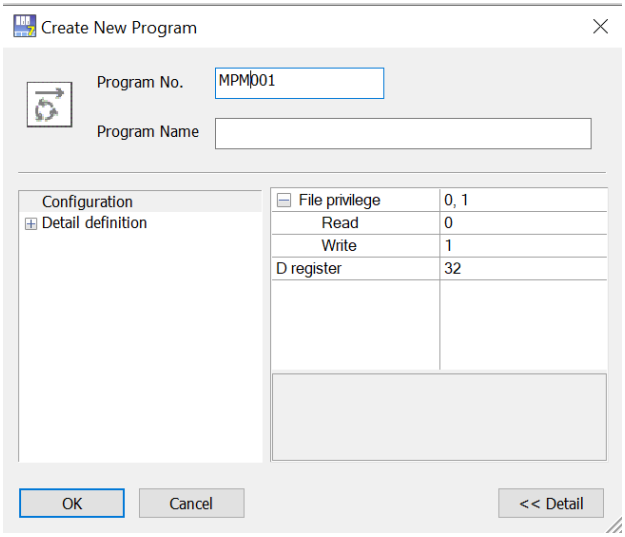
3. HƯỚNG DẪN VIẾT CHƯƠNG TRÌNH MOTION



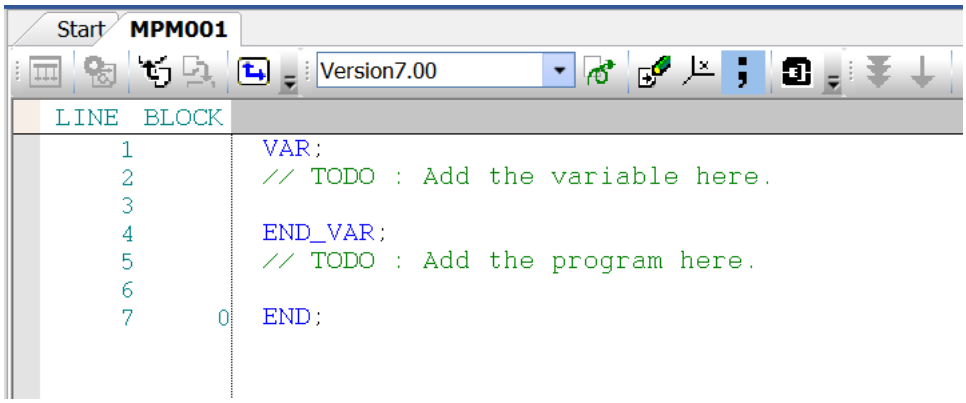
Tại Tab ***Motion Program***, nhấp chuột phải vào ***Main Program***, chọn ***New***. Đặt tên cho chương trình motion, và chọn ***OK***.



Lưu ý giá trị trong Program No. MPM**001**. Khi gọi chương trình motion (bằng hàm MSEE) trong ladder, con số 001 này được dùng để chỉ địa chỉ của chương trình motion cần gọi.



Một tab mới hiện ra để viết chương trình:



Trong đó:

+ Khoảng giữa *VAR;* và *END_VAR;* là để khai báo biến, gồm tên biến và địa chỉ của nó. Nếu không khai báo biến ở đây, thì trong chương trình chỉ được sử dụng địa chỉ tuyệt đối (như MW0110).

```
5   VAR;
6
7   //  LONG    MxxxSta     %GL3042;
8       LONG    MxxxSta     %ML02310;
9
10      LONG    MxxxNo      %GL03542;
11      LONG    MxxxAdr;
12
13      LONG    JT1Pos      %GL02020;
14      LONG    JT2Pos      %GL02022;
15      LONG    JT3Pos      %GL02024;
16      LONG    JT4Pos      %GL02026;
```

+ Comment được viết sau dấu "//"

+ Khoảng từ *END_VAR;* đến *END;* là để viết chương trình motion.

Ví dụ chương trình để trở về vị trí home của bộ mâm xoay và hệ ball crew như sau:

```
END_VAR;
    // HOME A
    OW803C=19;"Home method using zero input signal"
    OL8040=1000; "A Axis creep speed(0.1deg/s)"
    OL8010=3000;"A Axis speed reference(0.1deg/s)"
    OL8042=900;"A Axis final travel distance(0.1deg/s)"

    // HOME B
    OW80BC=19; "Home method using zero input signal"
    OL80C0=1000;"B Axis creep speed(0.1mm/s)"
    OL8090=1000;"B Axis speed reference(0.1mm/s)"
    OL80C2=0;"B Axis final travel distance(0.1mm)"

    // HOME C
    OW813C=19; "Home method using zero input signal"
    OL8140= 1000;"C Axis creep speed(0.1mm/s)"
    OL8110= 1000;"C Axis speed reference(0.1mm/s)"
    OL8142= 0; "C Axis final travel distance(0.1mm)"
    ZRN[A1]0[B1]0[C1]0;
    END;
```

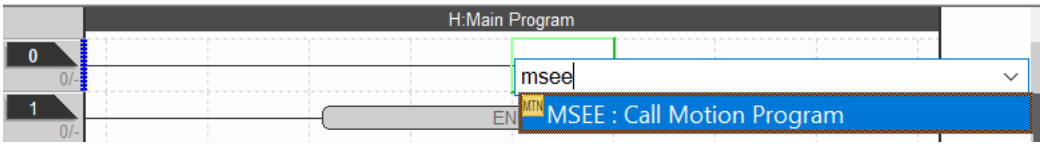
Sau khi viết xong chương trình, nhấn **F4** để complile.

Các hàm cơ bản:

Hàm	Ví dụ	Chức năng
ZRN	ZRN[A1]0[B1]0[C1]0; Lưu ý: Phải là số 0.	Về vị trí home cho các trục [A1], [B1] và [C1] (Thực hiện đồng thời)
WHILE WEND	WHILE SB1==1; //Program here WEND;	Chương trình được viết trong hàm này lặp lại với điều kiện SB1 bằng 1. Do SB1 là biến AlwaysOn nên chương trình viết trong đây sẽ được lặp lại mãi mãi.
IOW	IOW IB0C40C==1;	Chương trình chờ cho đến khi IB0C40C bằng 1 mới thực hiện lệnh kế tiếp.
ABS	ABS;	ABS = ABSOLUTE, đây là chế độ mặc định của các hàm motion. Nếu khai báo ABS thì từ đó trở về sau, các hàm motion sẽ theo mode này.
INC	INC;	INC = INCREMENTAL. Nếu khai báo INC thì từ đó trở về sau, các hàm motion sẽ theo mode này.
MOV	ABS; MOV[A1]900[B1]-100;	Di chuyển trục [A1] đến tọa độ 900 và trục [B1] đến tọa độ -100. Hàm này chỉ đảm bảo các trục [A1] và trục [B1] bắt đầu cùng lúc (Không đồng bộ).
	INC; MOV[A1]900[B1]-100;	Di chuyển trục [A1] thêm một khoảng cách 900 và trục [B1] thêm một khoảng cách -100 (di chuyển theo chiều âm của trục này). Hàm này chỉ đảm bảo các trục [A1] và trục [B1] bắt đầu cùng lúc (Không đồng bộ).
MVS	ABS; MVS[A1]900[B1]-100;	Di chuyển trục [A1] đến tọa độ 900 và trục [B1] đến tọa độ -100. Hàm này đảm bảo các trục [A1] và trục [B1] bắt đầu cùng lúc và kết thúc cùng lúc (Đồng bộ).
TIM1MS	TIM1MS T250;	Chương trình delay một khoảng thời gian 250 ms.

4. HƯỚNG DẪN GỌI CHƯƠNG TRÌNH MOTION BẰNG LADDER

Tại giữa rung, double click, gõ ***"msee"*** nhấn ***Enter:***

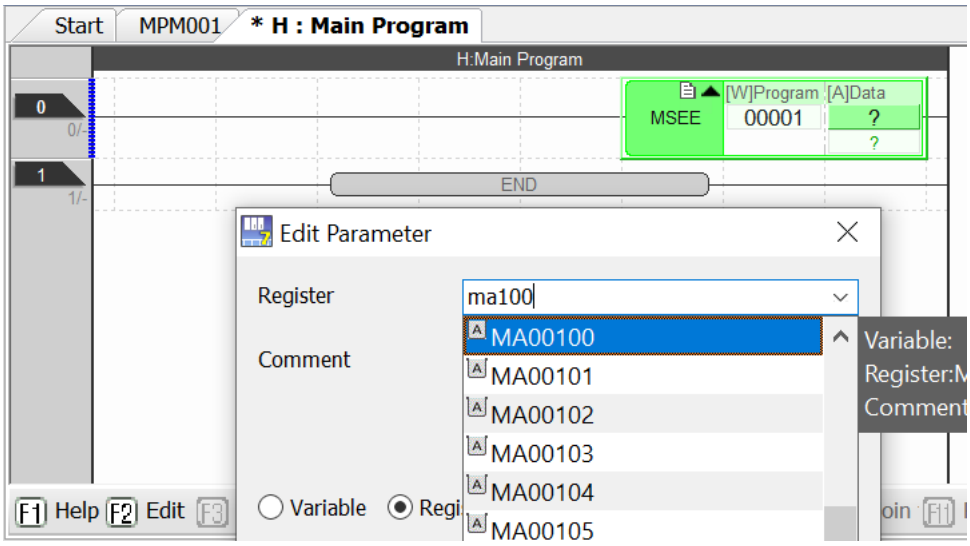


Trong hàm MSEE:

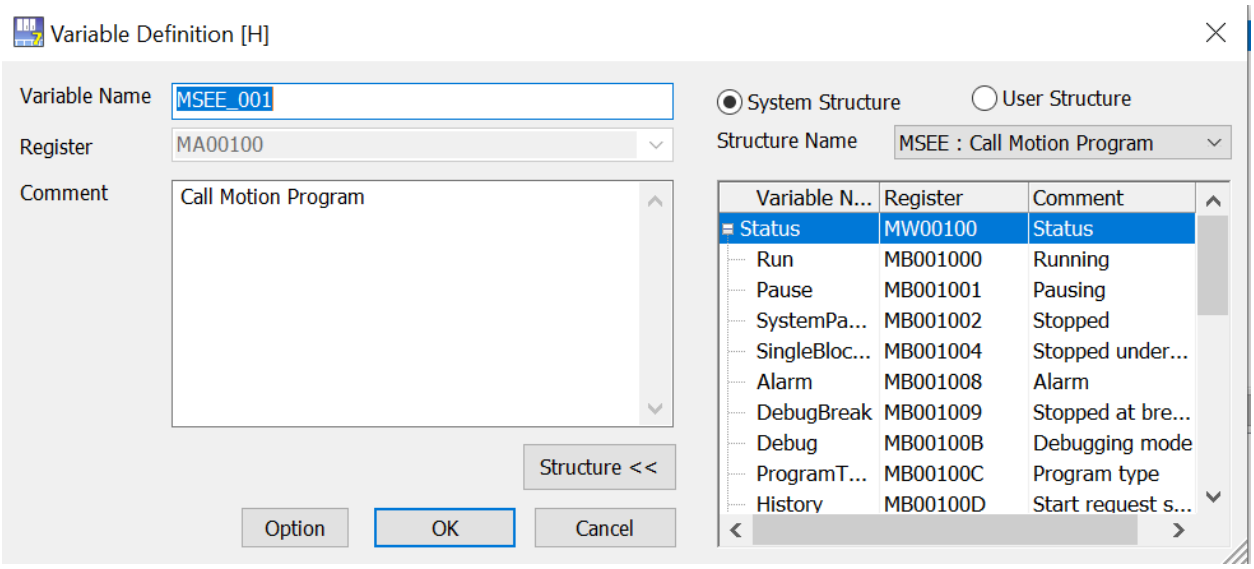
+ ***Program***: Khai báo địa chỉ chương trình motion muốn gọi.

+ ***Data***: Khai báo địa chỉ của một struct để điều khiển và quản lý chương trình này.

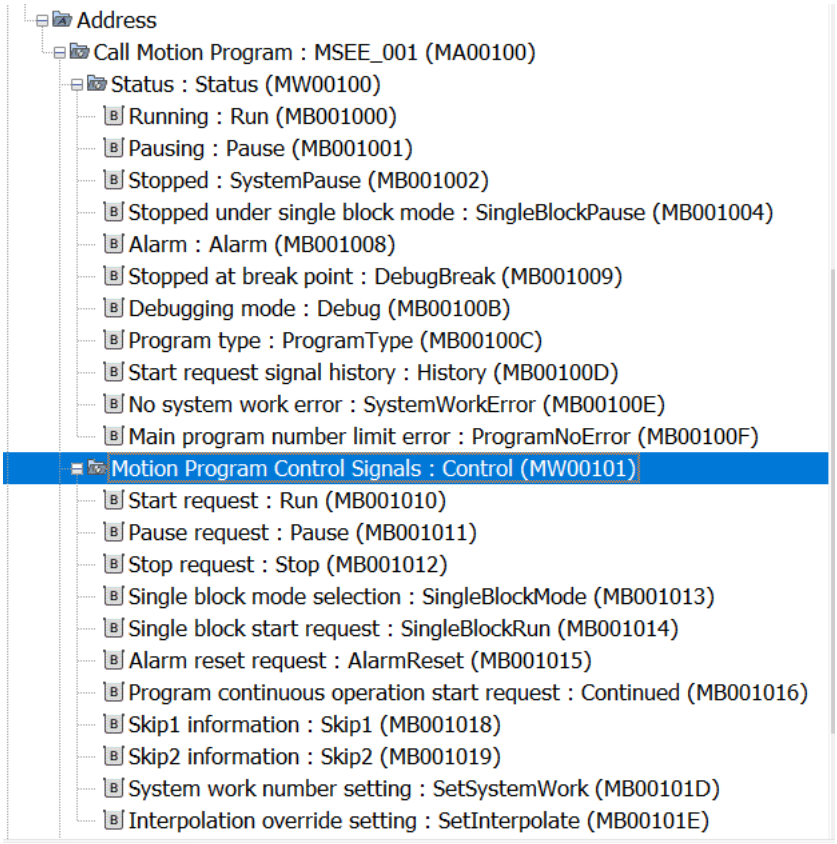
Ví dụ: Như struct này có địa chỉ từ MA00100:



Một cửa sổ mới hiện ra, ta nhập tên mong muốn của struct này và chọn **OK**.



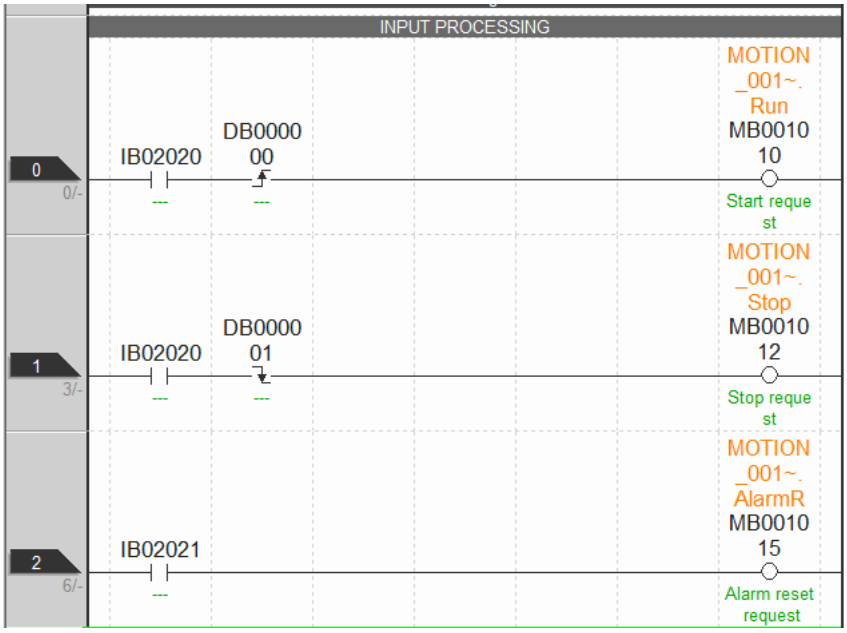
Trong struct này ta cần quan tâm đến các trường sau:

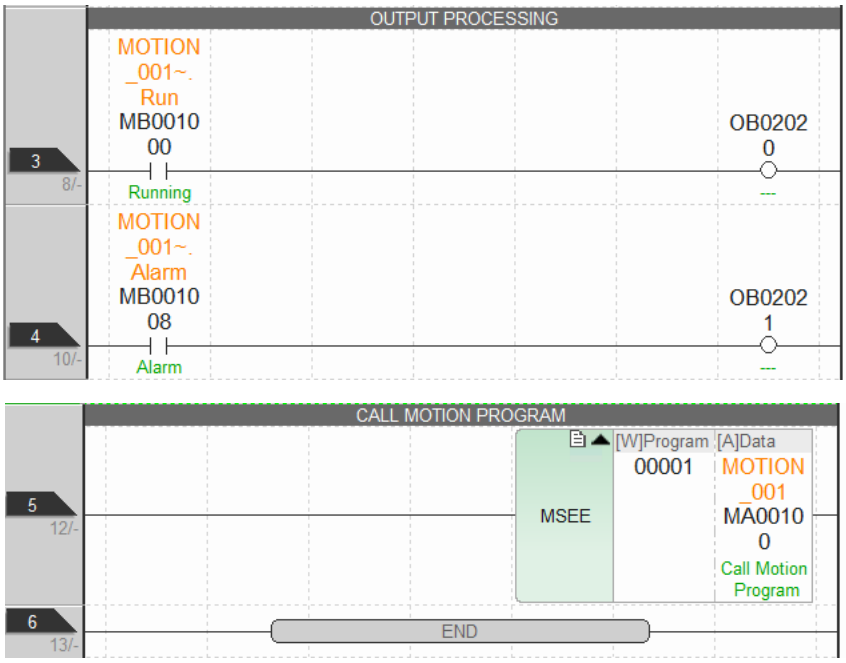


+ ***Status***: Trường này chỉ đọc, dùng để xem trạng thái của chương trình motion như Run, Pause, Alarm…

+ *Control*: Trường này để điều khiển chương trình như Run, Pause, Stop, AlarmReset (Các bit này tác động bằng cạnh lên).

Ví dụ: Chương trình bắt đầu khi switch có địa chỉ IB2020 bật, dừng khi switch này tắt. Nhấn nút có địa chỉ IB2021 để reset alarm. Đèn báo chương trình motion đang hoạt động có địa chỉ OB2020, đèn báo chương trình motion alarm có địa chỉ OB2021.




OUTPUT PROCESSING
MOTION _001~. Run
MB001000
Running
OB02020

MOTION _001~. Alarm
MB001008
Alarm
OB02021

CALL MOTION PROGRAM
[W]Program
00001
[A]Data
MOTION _001
MA00100
Call Motion Program
MSEE
END