

## BÀI THÍ NGHIỆM 3

### ĐIỀU KHIỂN SERVO DÙNG MACHINE CONTROLLER

#### 1. MỤC ĐÍCH THÍ NGHIỆM

Trong bài thí nghiệm này, sinh viên sẽ học cách lập trình machine controller để điều khiển chuyển động các động cơ servo, kết hợp với robot tạo thành một hệ thống tự động hoàn chỉnh.

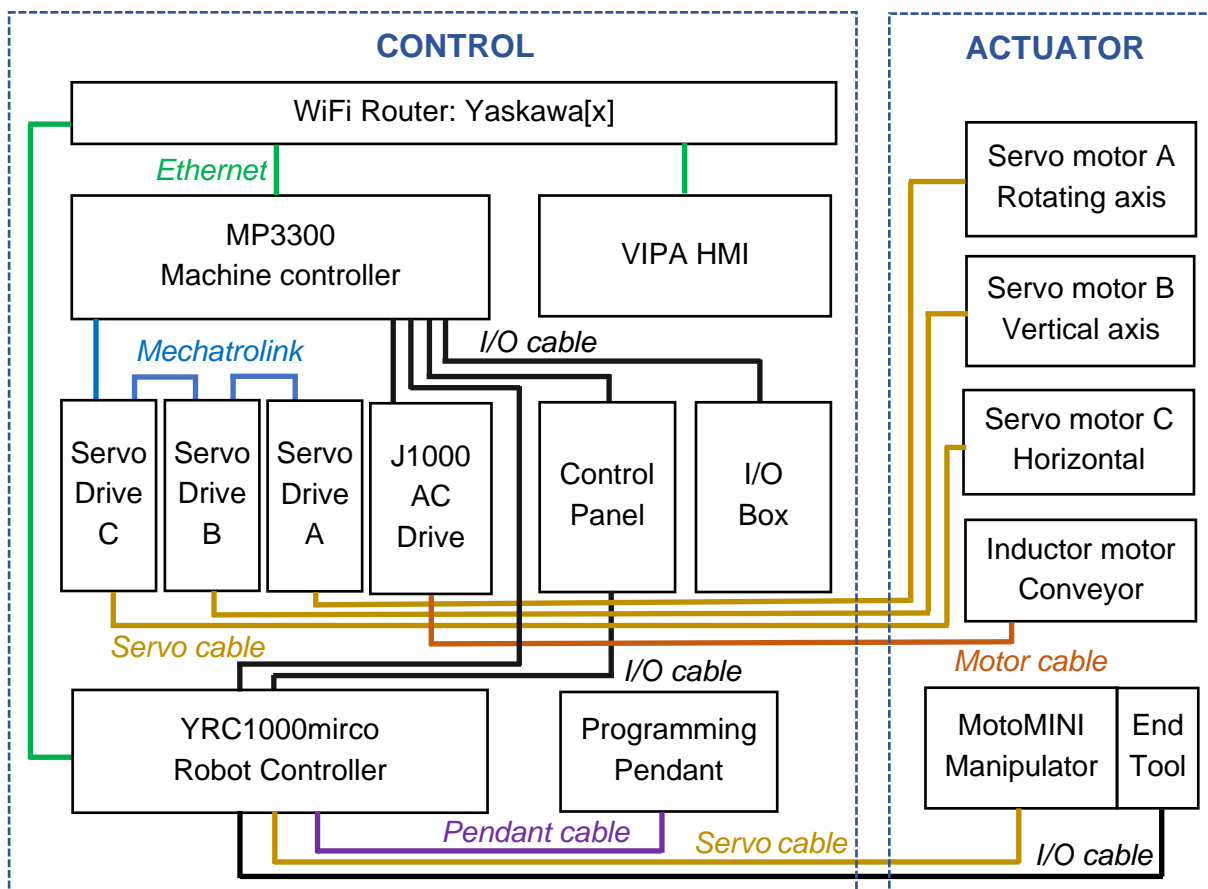
Sau khi thực hiện bài thí nghiệm, sinh viên có thể:

- Biết lập trình chương trình Motion sử dụng phần mềm MPE720.
- Biết lập trình machine controller MP3300 điều khiển chuyển động các động cơ servo.

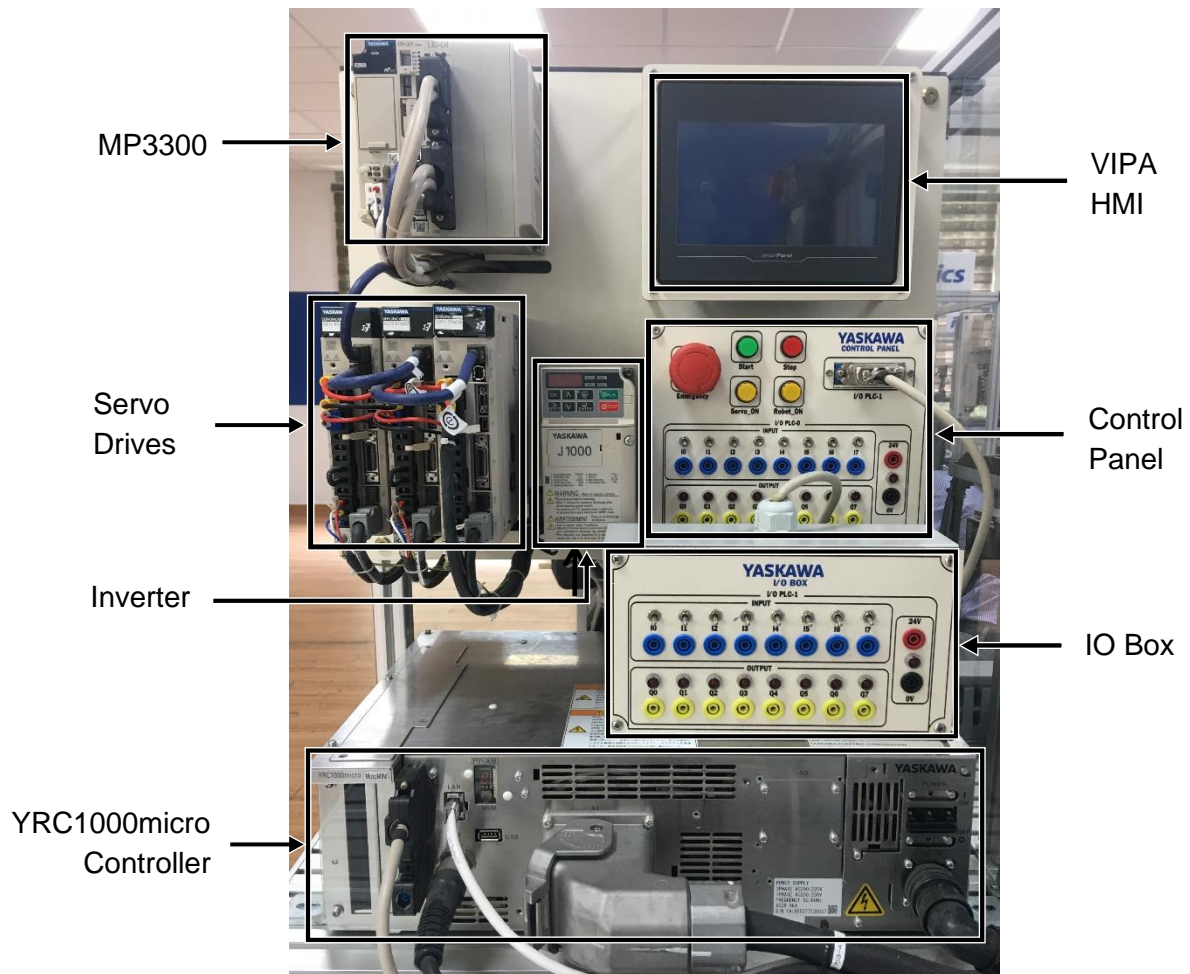
#### 2. MÔ TẢ THIẾT BỊ

##### 2.1. Tổng quan hệ thống

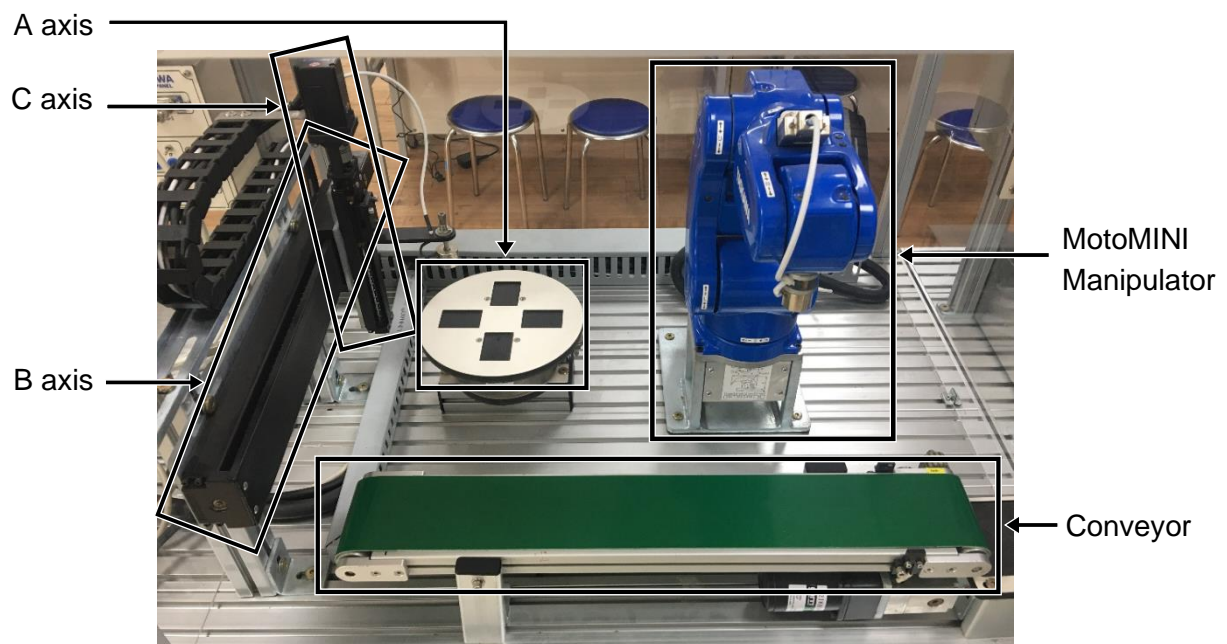
Sơ đồ tổng quan và hình ảnh thực tế các thiết bị trong bài thí nghiệm được thể hiện ở Hình 1, Hình 2 và Hình 3. Machine controller là trung tâm điều khiển và liên kết các thiết bị trong hệ thống như: các servo drive điều khiển các động cơ servo trên trục A, B và C; AC drive điều khiển động cơ trên băng tải; robot controller điều khiển cánh tay robot; Control Panel, I/O Box và HMI tương tác với người dùng. Machine controller được kết nối với các servo drive thông qua giao thức Mechatrolink, dùng để điều khiển các servo motor trong bài thí nghiệm này.



Hình 1. Sơ đồ tổng quan bộ thí nghiệm



Hình 2. Các thiết bị điều khiển



Hình 3. Các thiết bị chấp hành

Như Hình 3, động cơ servo trục A kéo mâm xoay, trục B kéo thanh trượt ngang, và trục C kéo thanh trượt dọc.

## 2.2. Kết nối I/O

Machine Controller (MC) có kết nối I/O với các thiết bị trong hệ thống như Bảng 1.

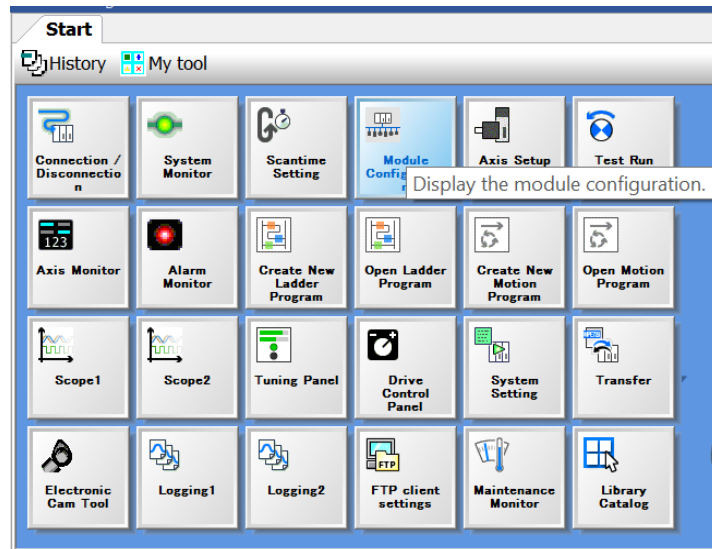
*Bảng 1. Kết nối I/O của Machine Controller*

MC DI	Target Device	Signal
DI_00	Control Pannel	START button
DI_01	Control Pannel	STOP button
DI_02	Control Pannel	Servo_ON button
DI_03	Control Pannel	Robot_ON button
DI_04	B axis	Sensor POT_B
DI_05	B axis	Sensor NOT_B
DI_06	B axis	Sensor HOME_B
DI_07	C axis	Sensor POT_C
DI_08	C axis	Sensor NOT_C
DI_09	C axis	Sensor HOME_C
DI_10	A axis	Sensor HOME_A
DI_11	Conveyor	Sensor Conveyor
DI_12	Robot	OUT#3
DI_16..23	Control Panel	I0..7 switches
DI_24..31	I/O Box	I0..7 switches
MC DO	Target Device	Signal
DO_00	Inverter	S1
DO_01	Robot	IN#1
DO_02	Robot	IN#2
DO_03	Robot	IN#3
DO_04	Robot	IN#4
DO_05	Magnet on C axis	MAG2
DO_16..23	Control Panel	Q0..7 lamps
DO_24..31	I/O Box	Q0..7 lamps

## 3. CHUẨN BỊ THÍ NGHIỆM

### 3.1. Cấu hình cho Machine Controller

Tại tab **Start**, chọn vào **Module Configuration** để cấu hình bộ điều khiển cũng như các thông số cho các servo:



Ta được một cửa sổ mới hiện ra có dạng như sau:

File Save to project Edit Setting Online Read Write Self Configuration All modules specified module Snap Save in Excel File								
Edit Edit Status Version	Module	Function Module/Slave	Status	Circuit No/AxisAddress		Motion Register	Register(Input/Output)	
				Start	cupied circ		Disabled	Start - End
	01 [CPU-301(16axes)] : --							
		01 CPU	----	----	----	----		----
		02 218IFD	----	Circuit No1	1	----	<input type="checkbox"/> Input	000 - 07FF[H]
		03 SVC	----	Circuit No1	1	8000 - 87FF[H]	<input type="checkbox"/> OutPut	800 - 0BFF[H]
	00 CPU301(16)	04 SVR	----	Circuit No2	1	8800 - 8FFF[H]		----
	01 MBU-303	05 M-EXECUTOR	----	----	----	----		000 - 0C3F[H]
		06 -- UNDEFINED --						
		07 -- UNDEFINED --						
	01 -- UNDEFINED --							

Trong đó, **CPU301(16)** là CPU chính của bộ điều khiển, gồm có các module nhỏ như **CPU**, **218IFD** (Để cấu hình địa chỉ IP và giao tiếp cho Machine Controller), **SVC** (Để cấu hình cho các động cơ Servo – trực thực), **SVR** (Để cấu hình cho các trục ảo) ...

#### Cấu hình các thông số cho các động cơ servo:

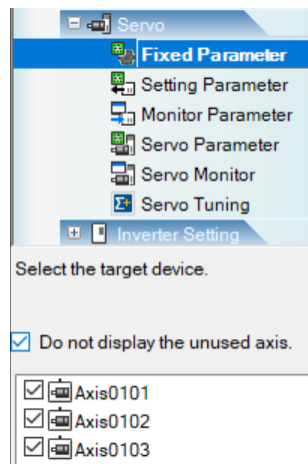
03 SVC	----	Circuit No1	1	8000 - 87FF[H]	<input type="checkbox"/> Input	000 - 0BFF[H]
01 -- UNDEFINED --	----				<input type="checkbox"/> OutPut	
02 -- UNDEFINED --	----					
03 -- UNDEFINED --	----					

Mở rộng Module **SVC**, double click vào **01--UNDEFINED--**, và chọn loại động cơ **SGD7S-\*\*\*\*20\***. Vì phần cứng của lab bao gồm 3 động cơ servo được điều khiển bằng Machine Controller nên ta thực hiện tương tự với 2 động cơ còn lại, được kết quả như sau:

03 SVC		Circuit No1	1	8000 - 87FF[H]	<input type="checkbox"/> Input <input type="checkbox"/> OutPut	00 - 0BFF
01 SGD7S-****20*		03[H] (00[H])		8000 - 807F[H]	<input type="checkbox"/> Input <input type="checkbox"/> OutPut	
02 SGD7S-****20*		04[H] (00[H])		8080 - 80FF[H]	<input type="checkbox"/> Input <input type="checkbox"/> OutPut	
03 SGD7S-****20*		05[H] (00[H])		8100 - 817F[H]	<input type="checkbox"/> Input <input type="checkbox"/> OutPut	

Trong đó, **Circuit No1** là địa chỉ vật lý của servopack (được chỉnh bằng cách vặn núm xoay trên servopack). Trước khi thực hiện nạp xuống Machine Controller nên kiểm tra chính xác địa chỉ của các servopack phải lần lượt là 03H, 04H và 05H.

Sau khi thực hiện xong các thao tác trên, chọn **Save to Project** để lưu các cấu hình lại.



Ở phía bên trái của cửa **Module Configuration** (như hình trên), tích chọn vào các **Axis0101**, **Axis0102** và **Axis0103**, và double click vào **Fixed Parameter** để cấu hình các thông số sau:

	Axis0101 Circuit#01 Axis#01 SGD7S-****20*	Axis0102 Circuit#01 Axis#02 SGD7S-****20*	Axis0103 Circuit#01 Axis#03 SGD7S-****20*
0 : Selection of operation modes	0 : Normal operation mode	0 : Normal operation mode	0 : Normal operation m...
1 : Function selection flag 1	0000[H]	0000[H]	0000[H]
2 : Function selection flag 2	0000[H]	0000[H]	0000[H]
4 : Reference unit selection	2 : deg	1 : mm	1 : mm
5 : Number of digits below decimal point	1 : 0.1	1 : 0.1	1 : 0.1
6 : Travel distance per machine rotation	360.0[deg]	75.0[mm]	125.0[mm]
8 : Servo motor gear ratio	225[rev]	1[rev]	1[rev]
9 : Machine gear ratio	3600[rev]	1[rev]	1[rev]
10 : Infinite length axis reset position(P...	360.0[deg]	36000.0[mm]	36000.0[mm]
12 : Positive software limit value	214748364.7[deg]	214748364.7[mm]	214748364.7[mm]
14 : Negative software limit value	-214748364.8[deg]	-214748364.8[mm]	-214748364.8[mm]
30 : Encoder selection	0 : Incremental encoder	0 : Incremental encoder	0 : Incremental encoder
34 : Rated motor speed	3000[ $\text{min}^{-1}$ ]	3000[ $\text{min}^{-1}$ ]	3000[ $\text{min}^{-1}$ ]
36 : Number of pulses per motor rotati...	16777216 : 24Bit[pulse/rev]	16777216 : 24Bit[pulse/rev]	16777216 : 24Bit[pulse/rev]
38 : Maximum number of absolute enc...	65534[rev]	65534[rev]	65534[rev]
42 : Feedback speed movement avera...	10[ms]	10[ms]	10[ms]
44 : User Select Servo Driver User Co...	0000[H]	0000[H]	0000[H]
45 : User Select Servo Driver User Co...	1[word]	1[word]	1[word]

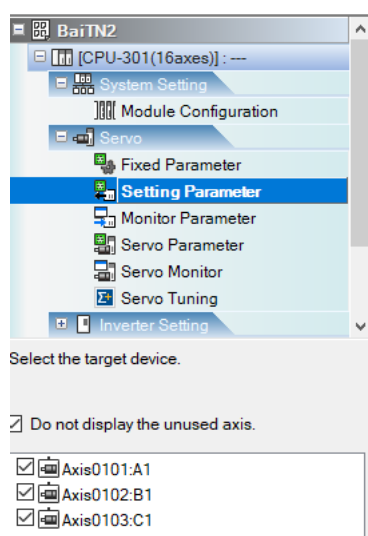


Trong đó cần lưu ý các thông số:

<b>Reference unit selection</b>	Chọn đơn vị. Ví dụ: deg, mm...
<b>Number of digits below decimal point</b>	Chọn số chữ số thập phân sau dấu phẩy của đơn vị. Ví dụ, nếu chọn thông số này bằng 1 thì đơn vị là 0.1 deg/mm/...
<b>Travel distance per machine rotation</b>	Khoảng cách đi được nếu động cơ quay được 1 vòng.
<b>Servo motor gear ratio</b>	Thông số của bộ truyền động gắn với động cơ.
<b>Machine gear ratio</b>	
<b>Positive software limit value</b>	Giá trị giới hạn trên.
<b>Negative software limit value</b>	Giá trị giới hạn dưới.
<b>Encoder selection</b>	Chọn loại encoder (absolute hoặc incremental)
<b>Rated motor speed</b>	Tốc độ định mức của động cơ.

Có thể tự nhập bằng tay các thông số này, hoặc phần mềm MPE720 có cho phép người dùng import từ file excel có sẵn như sau: Đầu tiên chọn **Import** → **Import all together**, chọn đường dẫn đến thư mục lưu các thông số này chọn **OK**.

Sau khi hoàn tất, chọn **Save to Project**.



Ở phía bên trái của cửa **Module Configuration** (như hình trên), tích chọn vào các **Axis0101**, **Axis0102** và **Axis0103**, và double click vào **Setting Parameter** để cấu hình các thông số sau:

3 : Function setting 1	0W8103	0000[H]	0000[H]	0000[H]
[Bit0-3]Speed unit selection	0B81030	0 : reference unit/sec	0 : reference unit/sec	0 : reference unit/sec
[Bit4-7]Acceleration/Deceleration u...	0B81034	0 : Reference unit/s^2	0 : Reference unit/s^2	0 : Reference unit/s^2
[Bit8-B]Filter type selection	0B81038	0 : Filter none	0 : Filter none	0 : Filter none
[BitC-F]Torque unit selection	0B8103C	0 : 0.01%	0 : 0.01%	0 : 0.01%

Trong đó, chú ý các thông số sau:

<b>Speed unit selection</b>	Chọn đơn vị của tốc độ. Trong trường hợp này chọn đơn vị là reference unit/sec (tức là 0.1 deg/s hoặc 0.1 mm/s)
-----------------------------	---

<b>Acceleration/ Deceleration unit selection</b>	Chọn đơn vị của gia tốc. Trong trường này chọn đơn vị là reference unit/sec <sup>2</sup> (tức là 0.1 deg/s <sup>2</sup> hoặc 0.1 mm/s <sup>2</sup> )
--	--

Sau khi hoàn tất, chọn **Save to Project**.

### 3.2. Tìm hiểu các hàm motion của machine controller MP3300

**Tìm trong tài liệu [2] các hàm sau và nêu chức năng của các hàm đó:**

Hàm	Ví dụ	Chức năng
<b>ZRN</b>	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)
<b>WHILE WEND</b>	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.
<b>IOW</b>	IOW IB0C40C==1;	Chương trình chờ cho đến khi IB0C40C bằng 1 mới thực hiện lệnh kế tiếp.
<b>ABS</b>	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.
<b>INC</b>	INC;	INC = INCREMENTAL. Nếu khai báo INC thì từ đó trở về sau, các hàm motion sẽ theo mode này.
<b>MOV</b>	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).
<b>MVS</b>	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).
<b>TIM1MS</b>	TIM1MS T250;	Chương trình delay một khoảng thời gian 250 ms.

### 3.3. Đọc tài liệu machine controller về Software Limits

Đọc các tài liệu *Motion Control, Machine Controller MP3000 Series, USER'S MANUAL SVC/SVR, SVC32/SVR32*.

Sinh viên xem tài liệu phần 6.3. *Software Limits*, và trả lời các câu hỏi sau:

a. Tại sao cần phải có Software Limit?

→ Để cài đặt giới hạn trên và giới hạn dưới cho tầm di chuyển của máy, tránh hư hỏng cho máy.

b. Cần cài đặt các thông số nào liên quan đến Software Limit?

→ Cài Fixed Parameters và/hoặc Setting Parameters:

◆ Fixed Parameters

No.	Name	Setting Range	Setting Unit	Default
No.1 Bit 0	Axis Selection	0: Finite-length axis 1: Infinite-length axis	–	0: Finite-length axis
No.1 Bit 1	Enable Positive Software Limit	0: Disabled 1: Enabled	–	0: Disabled
No.1 Bit 2	Enable Negative Software Limit	0: Disabled 1: Enabled	–	0: Disabled
No.1 Bit C	Software Limit Parameter Selection	0: Fixed parameters 1: Setting parameters	–	0: Fixed parameters
No.12	Positive Software Limit	$-2^{31}$ to $2^{31}-1$	Reference units	$2^{31}-1$
No.14	Negative Software Limit	$-2^{31}$ to $2^{31}-1$	Reference units	$-2^{31}$

◆ Setting Parameters

Register Address	Name	Setting Range	Setting Unit	Default
OL□□□66	Positive Software Limit	0: Finite-length axis 1: Infinite-length axis	Reference units	0: Finite-length axis
OL□□□68	Negative Software Limit	0: Disabled 1: Enabled	Reference units	0: Disabled

c. Khi dùng hàm MOV (Lệnh POSING) thì ảnh hưởng của Software Limit như thế nào đối với chuyển động của servo?

→ Axis bắt đầu giảm tốc khi tiến lại vị trí Software Limit và dừng ngay tại giá trị Limit đó.

## 4. THỰC HIỆN THÍ NGHIỆM

### 4.1. Điều khiển mâm xoay

#### 4.1.1. Viết chương trình về điểm zero của mâm xoay

Đối với loại incremental encoder như trong LAB, khi mới bật điện servopack lên, servo mặc định cho rằng vị trí ngay tại điểm đó là vị trí zero (tức là vị trí có tọa độ 0). Tuy nhiên, ta không muốn vị trí zero nằm ngẫu nhiên như vậy, mà phải cố định. Vì thế, mỗi lần sau khi ngắt điện và bật lại, ta cần phải dò zero cho servo.

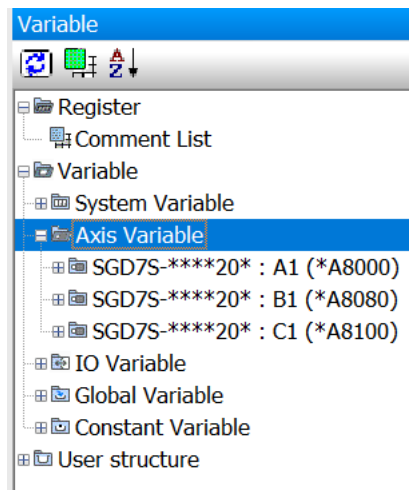
Có tới 13 cách về điểm zero như trong tài liệu [2] trình bày, tuy nhiên với cách đấu nối dây cảm biến hiện tại của LAB, ta chỉ có thể chọn cách về zero số 19 (Chỉ dùng tín hiệu zero input).

**Yêu cầu 1.1:** Viết chương trình Motion MP0001 thực hiện về zero cho mâm xoay sao cho điểm zero của mâm xoay là điểm mà cảm biến tiệm cận trên mâm (DI-10) được bật lên.

#### Hướng dẫn:

Sau khi thực hiện cấu hình cho machine controller như phần chuẩn bị, trong mục **Axis Variable** xuất hiện các biến A1, B1, C1 quản lý các servo drive trục A, B và C như sau:





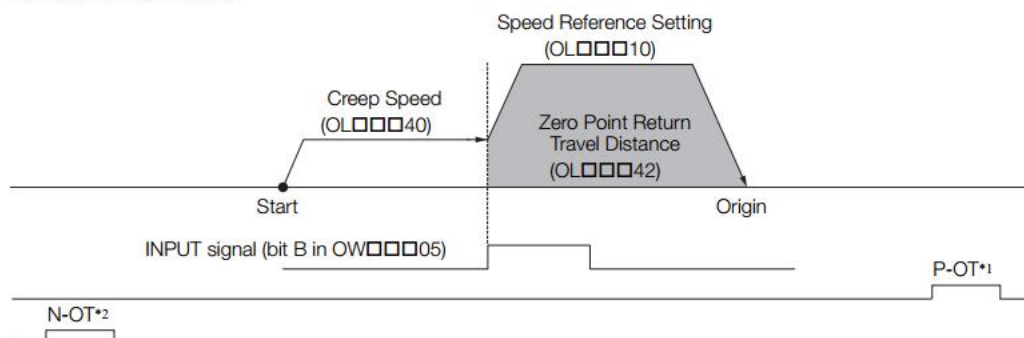
Để thử bật servo A1, ta cho biến **A1.ServoOn** (OB80000) = 1.

Để giám sát trạng thái của servo A1 đã được bật hay chưa, ta đọc giá trị của biến **A1.Running** (IB80001): Nếu ON tức là servo được bật và OFF tức là servo tắt.

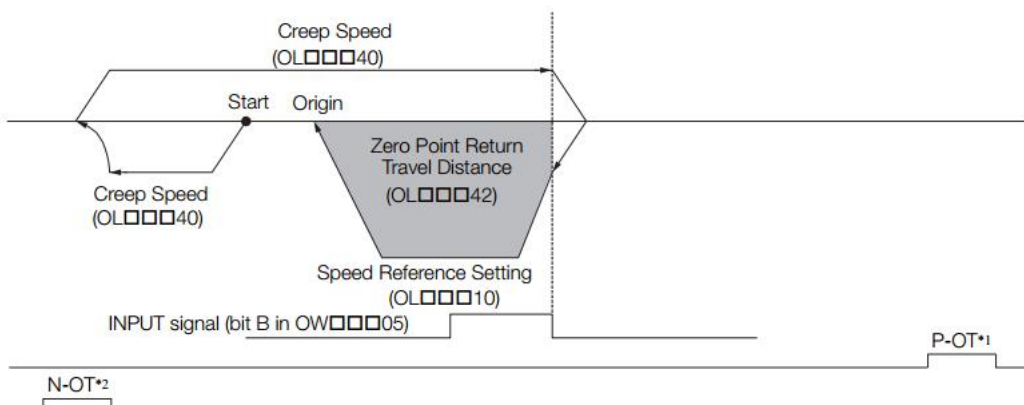
Chọn cách về zero thứ 19 (Chỉ dùng tín hiệu input), lúc này cần phải khai báo tín hiệu A1.ZeroPointReturn.InputSignal = DI\_10 bên chương trình ladder.

Quá trình hoạt động của cách về zero như được mô tả như sau:

- Normal Execution



- Overtravel Signal Detected during Travel at the Creep Speed



\*1. SERVOPACK P-OT signal.

\*2. SERVOPACK N-OT signal.

Đầu tiên, servo di chuyển tốc độ tối đa Creep Speed. Đến khi có cạnh lên của tín hiệu Zero Input, servo di chuyển với tốc độ tối đa Speed Reference và với khoảng cách là Travel Distance, vị trí dừng sau quá trình này là vị trí zero. Cài đặt biến A1 như sau:

Axis	A1
Creep Speed	50 deg/s
Speed Reference	50 deg/s
Travel Distance	360 deg
Input Signal	DI-10

#### 4.1.2. Viết chương trình di chuyển mâm xoay

**Yêu cầu 1.2:** Viết chương trình Motion MP0002, sao cho mỗi khi tín hiệu từ robot (DI-12) được bật thì mâm xoay xoay 90 deg.

**Gợi ý:** Có thể dùng bit SB1 (AlwaysOn) để cho chương trình luôn lặp lại.

#### 4.1.3. Viết chương trình Ladder

**Yêu cầu 1.3:** Viết chương trình ladder thực hiện các yêu cầu sau:

- Nhấn nút Servo On (DI-02) lần thứ nhất, tất cả servo đều được bật, nhấn Servo On lần nữa, tất cả các servo đều được tắt. Trạng thái của các servo được thể hiện qua đèn DO-16, DO-17 và DO-18.
- Khi nhấn nút Start (DI-0), băng tải (DO-0) hoạt động, nhấn nút Stop (DI-01) băng tải dừng. Đồng thời khi băng tải đang hoạt động và có vật chắn cảm biến trên băng tải (DI-11), băng tải dừng và tín hiệu nối qua I/O của robot (DO-01) được bật, cho đến khi vật không chắn cảm biến nữa, băng tải tiếp tục hoạt động và tín hiệu DO-01 tắt.
- Nhấn nút **Start**, chương trình motion thực thi, nhấn nút **Stop** chương trình motion dừng. Đồng thời nếu có Error, Alarm hoặc Warning của CPU, Servo hoặc của chương trình Motion, phải hiển thị ra đèn DO-19 (Nhấp nháy). Khi nhấn nút **Robot On** (DI-03) thì các Alarm hoặc Warning phải được reset.

**Gợi ý:** Các biến thể hiện trạng thái Error, Alarm hoặc Warning:

Tên biến	Nội dung	Giá trị khi Alarm
CPU.Status.Alarm (SB402)	Trạng thái Alarm của CPU	1
CPU.Status.Error (SB403)	Trạng thái Error của CPU	1
[CallMotion].Status.Alarm Trong đó: CallMotion là tên của struct dùng để gọi và giám sát chương trình motion.	Trạng thái Alarm của chương trình Motion. Ví dụ: Chưa bật Servo mà thực hiện hàm MOV...	1
[CallMotion].Status.SystemPause	Trạng thái buộc dừng của chương trình motion, xảy ra khi chương trình đang thực thi mà bị buộc dừng (có cạnh lên của biến [CallMotion].Status.Stop)	1
[A1].Alarm Trong đó: A1 là tên của servo hay axis do ta khai báo.	Trạng thái Alarm của Servo. Giá trị Alarm này nếu khác 0 sẽ là mã lỗi của servo.	Khác 0.

Cách để reset lỗi:

Tên biến tác động	Cách tác động	Công dụng
[A1].AlarmClear	Xung cạnh lên	Reset một số lỗi của servo.
[CallMotion].Control.AlarmReset	Xung cạnh lên	Reset chương trình Motion (Lỗi, trạng thái buộc dừng...)

## 4.2. Điều khiển hệ thanh trượt

### 4.2.1. Cài đặt giới hạn mềm của hệ thanh trượt

Đối với mâm xoay, ta không cần giới hạn tọa độ di chuyển của nó. Nhưng đối với hệ thanh trượt, ta cần phải có các giới hạn này, có hai cách thực hiện điều này (có thể thực hiện một trong hai hoặc tốt nhất là kết hợp cả hai cách):

Cách 1: Dùng cảm biến P-OT (Positive Overtravel) và cảm biến N-OT (Negative Overtravel) nối với Servopack. Khi đang di chuyển theo chiều dương mà gặp cảm biến P-OT, servo tự động giảm tốc về 0 và tương tự đối với việc di chuyển theo chiều âm.

Cách 2: Dùng giới hạn mềm (Software Limit). Các giá trị giới hạn này gồm Positive SoftLimit và Negative SoftLimit, được cài trong chương trình thông qua các thông số. Khi dùng kết hợp cả hai cách, giá trị tuyệt đối của tọa độ của Softlimit luôn nhỏ hơn giá trị tuyệt đối của tọa độ của cảm biến Overtravel tương ứng.

Trong bài thí nghiệm này, bởi vì cảm biến P-OT và N-OT không nối với Servopack mà nối với Machine Controller, nên ta chỉ có thể thực hiện theo cách thứ hai.

**Yêu cầu 2.1:** Cài đặt các thông số giới hạn mềm sau:

Axis	B1	C1
Positive Software Limit	1 mm	1 mm
Negative Software Limit	-280 mm	-60 mm

**Lưu ý:** Sau khi cài xong các thông số này tại Fixed Parameter. Chọn Save to Flash, chờ hoàn tất rồi mới ngắt điện Machine Controller. Sau đó, mở nguồn điện lại để cho các thông số vừa ghi có hiệu lực.

### 4.2.2. Viết chương trình về điểm zero của hệ thanh trượt

**Yêu cầu 2.2:** Viết thêm vào chương trình motion MPM001 phần về điểm zero cho trục B và C, sử dụng cách về zero thứ 19 (Chỉ dùng tín hiệu input). Trạng thái về điểm zero hoàn tất được lần lượt hiển thị ra các đèn DO-25 và DO-26.

#### Hướng dẫn:

Cài đặt biến B1 và C1 như sau:

Axis	B1	C1
Creep Speed	20 mm/s	20 mm/s
Speed Reference	20 mm/s	20 mm/s
Travel Distance	0 mm	0
Input Signal	DI-08	DI-04

**Lưu ý:** Bắt buộc về điểm zero của trục phương thẳng đứng C1 trước, rồi mới về điểm zero của trục ngang B1.

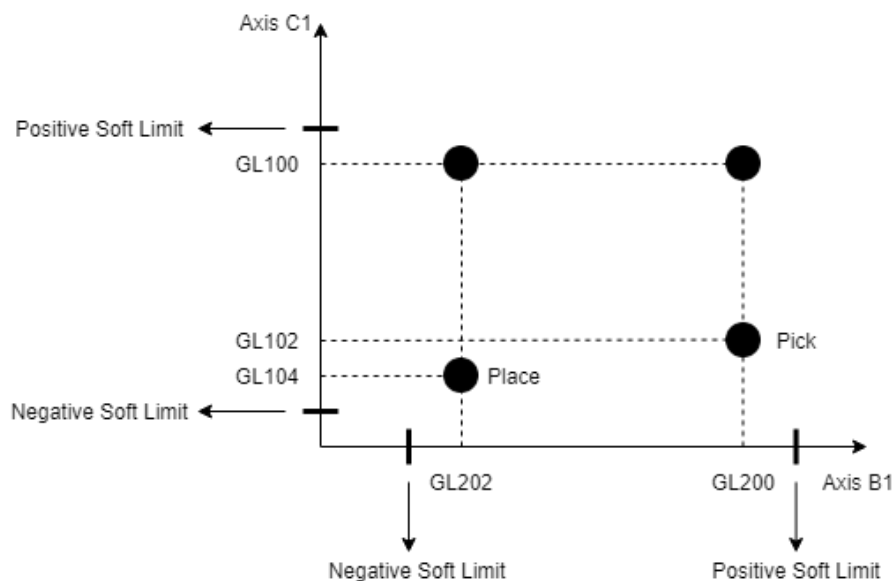
### 4.2.3. Viết chương trình di chuyển phối hợp mâm xoay và thanh trượt

#### Yêu cầu 2.3:

- a. Viết thêm vào chương trình motion MPM002, sao cho chương trình thực hiện theo trình tự sau:
  - 1) Trục C1 di chuyển tới vị trí có tọa độ GL100, sau đó trục B1 mới được di chuyển tới vị trí có tọa độ GL200.
  - 2) Chờ tín hiệu từ robot (DI-12) được bật.
  - 3) Mâm xoay 90 deg.
  - 4) Trục C1 di chuyển tới vị trí có tọa độ GL102, bật DO- , rồi di chuyển về vị trí có tọa độ GL100.
  - 5) Trục B1 di chuyển tới vị trí có tọa độ GL202.
  - 6) Trục C1 di chuyển tới vị trí có tọa độ GL104, tắt DO- , sau đó di chuyển về vị trí có tọa độ GL100.
  - 7) Trục B1 di chuyển về vị trí có tọa độ GL200.
  - 8) Quay lại bước 2.
- b. Viết chương trình Ladder gọi chương trình motion MPM001, sao cho khi bật switch DI-24, các servo về điểm zero.
- c. Viết chương trình Ladder gọi chương trình motion MPM002, sao cho khi bật switch DI-25 chương trình này bắt đầu thực thi, tắt switch này chương trình dừng; bật switch DI-26 chương trình tạm dừng và tắt switch DI-26 chương trình tiếp tục thực thi. Đồng thời nếu có Error, Alarm hoặc Warning của CPU, Servo hoặc của chương trình Motion, phải hiển thị ra đèn DO-19 (Nhấp nháy). Khi có cảnh lên từ switch DI-31 thì các Alarm hoặc Warning phải được reset.

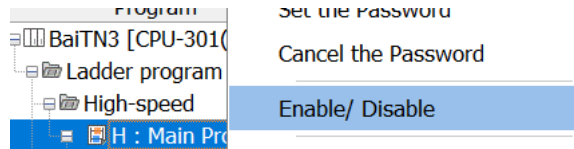
### 4.2.4. Tìm các tọa độ di chuyển của hệ thanh trượt

**Yêu cầu 2.4:** Ở mục 4.2, ta chỉ gán tọa độ di chuyển của hệ ball screw này thông qua các biến, bây giờ ta gán giá trị cho các biến này như sau:

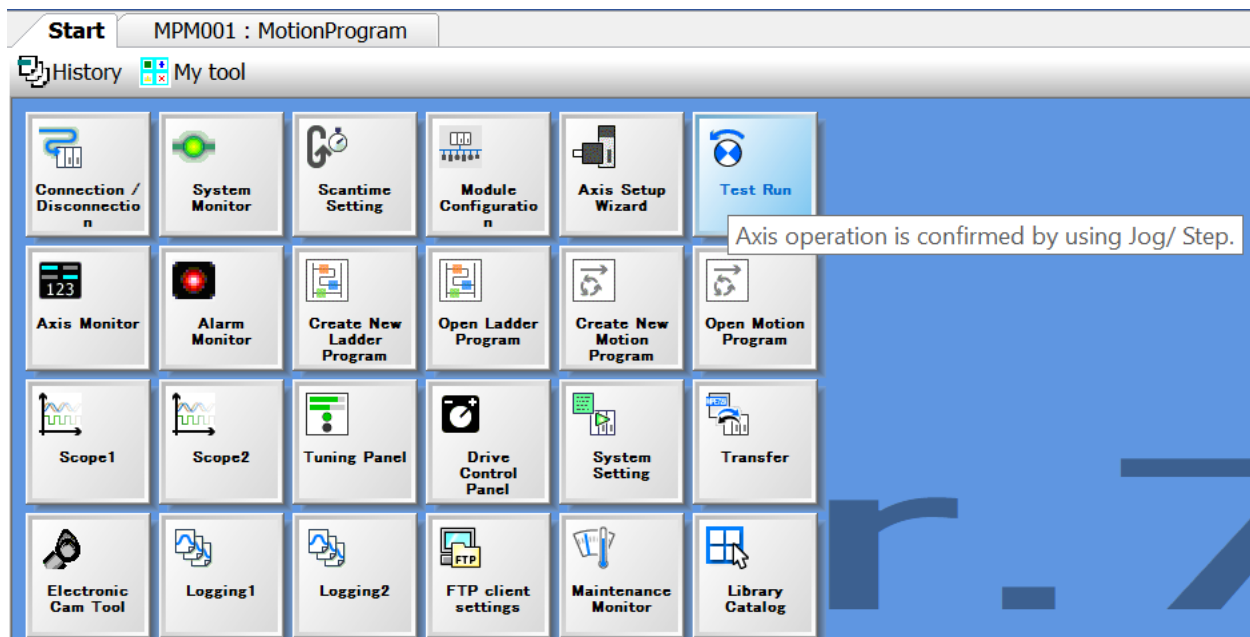


#### Hướng dẫn:

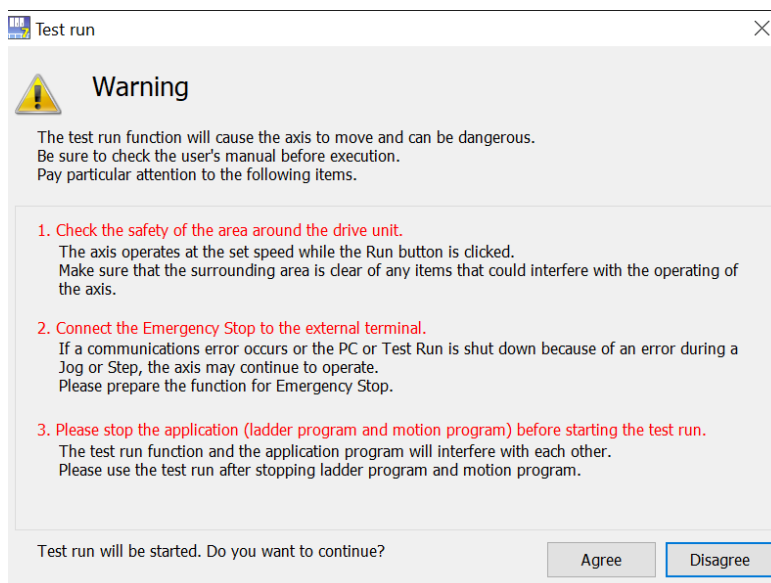
- 1) Bật tắt cả các Servo, bật switch DI-24 để tắt cả về điểm zero, chờ các tín hiệu DO-24, DO-25 và DO-26 đều bật lên (về điểm zero hoàn tất).
- 2) Tắt tắt cả các servo, tắt switch DI-24.
- 3) Vô hiệu hóa tất cả chương trình Ladder bằng cách nhấp chuột phải vào chương trình Ladder và chọn Enable/Disable.



- 4) Về lại cửa sổ Start và chọn Test Run.

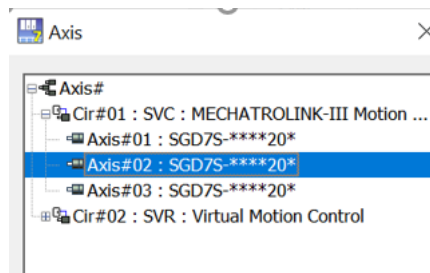


Đọc và kiểm tra lại tất cả các Warning được pop-up ra như hình dưới, sau đó chọn Agree:

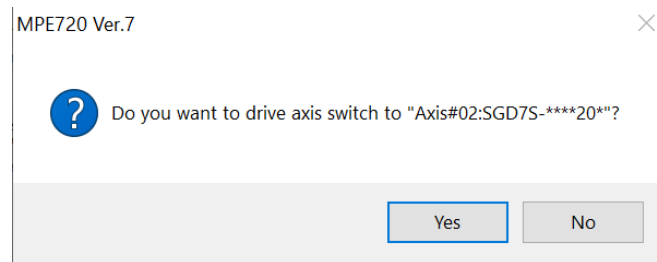


Tiếp tục chọn Axis B1 hoặc C1 là các trục cần JOG và chọn OK.

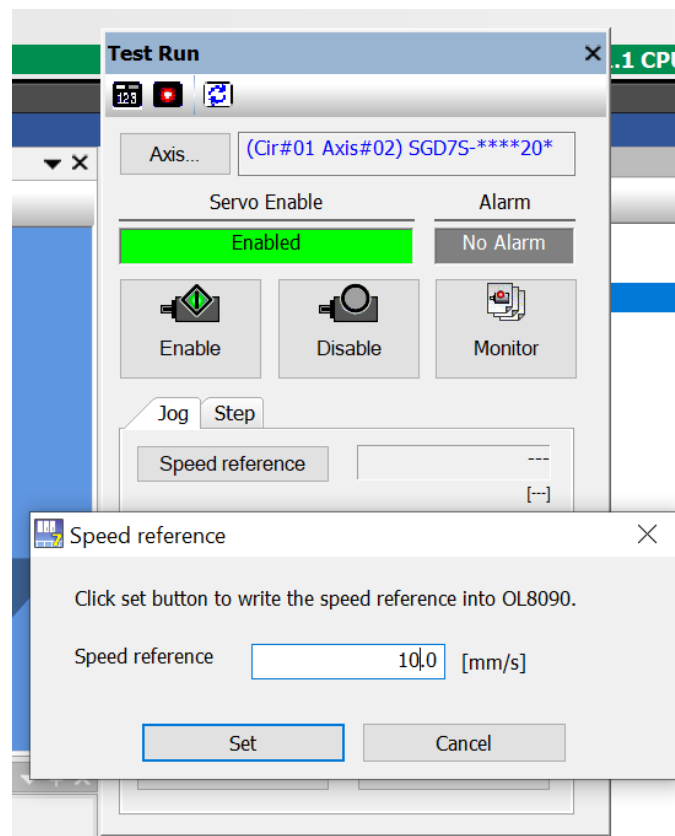




Xác nhận trực muốn Jog bằng cách chọn Yes.



Một cửa sổ mới hiện ra như sau:



Chọn Enable để bật Servo. Sau đó chọn Tab Jog và cài đặt tốc độ Jog (Không được lớn hơn 10 mm/s) và chọn Set.

Bấm vào các nút Forward và Backward để servo di chuyển lần lượt theo chiều thuận và chiều ngược.

Lưu ý: Luôn để trục C1 (trục đứng) phải có vị trí đảm bảo không va chạm mới được di chuyển trục B1 (trục ngang).

5) Thực hiện jog cả hai trục và điền vào bảng sau giá trị các tọa độ:

Biến	GL100	GL102	GL104	GL200	GL202
Giá trị					

6) Viết chương trình gán giá trị vào biến.

#### 4.3. Lập trình robot

**Yêu cầu 3:** Viết chương trình robot như bài thí nghiệm 2.

### 5. BÁO CÁO

Trình bày việc thực hiện các yêu cầu trong bài: chương trình robot, chương trình ladder và motion của machine controller, hình ảnh/video kết quả, trả lời câu hỏi, nhận xét, kết luận.

### TÀI LIỆU THAM KHẢO

- [1] Machine Controller MP3000 Ladder Program Manual, Yaskawa Electric Corp., 2015.
- [2] Machine Controller MP3000 Motion Program Manual, Yaskawa Electric Corp., 2015.