NEXCOBOT

Intelligent Solutions for IoT Automation

User's Manual for miniBOT

Ver.: 1.4

June 21 2018

NEXCOBOT Inc.
www.nexcobot.com

Copyright Statement and Disclaimer

The contents contained in this document are the proprietary property of NexCOBOT Inc. (NexCOBOT hereafter) and is subject to the protection of intellectual property law (including, but not limited to the Copyright Act). The use of any material in relation to this document without the prior authorization of NexCOBOT is considered infringement. Without the written approval of NexCOBOT in advance, this document or any part of it shall not be photocopied, sold, distributed, modified, published, stored or otherwise used.

To keep this document and its contents correct and complete, NexCOBOT reserves the right to change or revise the document at any time without further notification.

Operating machine or equipment has a certain level of danger. It is the user’s responsibility to pay special attention and have safety protection in place before operating any machine or equipment. NexCOBOT shall not be held for any and all direct or indirect damage or loss to the equipment mentioned in this document due to the use for a purpose other than the intended.

Revision History

Rev.	Description
1.0	First released.
1.1	1.Add Ch2.5 Power on sequence 2.Add more description for Ch1.3 Payload 3.Ch1.3 Modify picture 4.Ch1.2 Modify spec of control box, 100V~240V,DI 15 channels 5.Add Ch3.1.4 I/O configuration
1.2	1. Remove empty pages.
1.3	1. Add control box pictures.
1.4	1. Modify some typo words

Contents

1. Introduction

1.1. Specifications of the robotic arm

Item			Specifications
Degree of freedom			6
Payload			rate : 1 kg
Drive system			Step motor
Position detection method			Incremental encoder
Joint equipment	J1	Motor	PBM604DXK50
		Reducer	VRGF-33C090-14BP10
		Encoder	Encoder With Motor
		Driver	PB4D003E440
		Brake	None
	J2	Motor	PBM604DCK50
		Reducer	VRGF-33C090-14BP10
		Encoder	Encoder With Motor
		Driver	PB4D003E440
		Brake	Yes
	J3	Motor	PBM603DCK50
		Reducer	VRGF-33B060P-14BP10
		Encoder	Encoder With Motor
		Driver	PB4D003E440
		Brake	Yes
	J4	Motor	PBM423DCK50
		Reducer	VRT-042-25-F5-S8ZM6
		Encoder	Encoder With Motor
		Driver	PB4D003E440
		Brake	Yes
	J5	Motor	PBM423DXK50
		Reducer	VRT-042-20-F5-S8ZM6
		Encoder	Encoder With Motor
		Driver	PB4D003E440
		Brake	None
	J6	Motor	PBM423DXK50
		Reducer	VRT-042-10-F3-S8ZM6
		Encoder	Encoder With Motor
		Driver	PB4D003E440
		Brake	None
Motion range (degree)	J1		350° (+175° ~ -175°)
	J2		212° (+110° ~ -92°)
	J3		216° (+63° ~ -153°)
	J4		350° (+175° ~ -175°)
	J5		184° (+92° ~ -92°)
	J6		360° (180° ~ -180°)
Reduction ratio	J1		33
	J2		33
	J3		33
	J4		25
	J5		20
	J6		10
Motor Resolution	J1		10000
	J2		10000

	J3	10000
	J4	10000
	J5	10000
	J6	10000
Encoder Resolution	J1	10000
	J2	10000
	J3	10000
	J4	10000
	J5	10000
	J6	10000
Cycle time (*1)		TBD

(*1) Test value for cycle time:

The path of travel for pickup/drop-off is defined as follows. The cycle time is the time needed for the arm to travel along this path to the end and back to the origin.

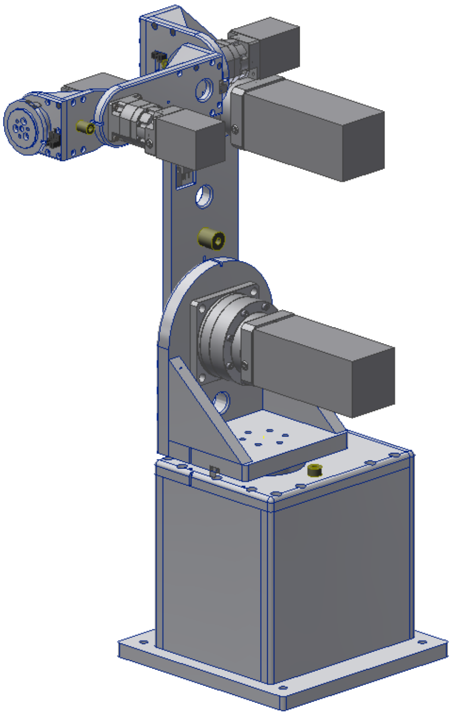
1.2. Electric Control Box Specifications

Item		Specifications
No. of control axes		6
Way of control		PTP (point-to-point control) CP (continuous path control)
Control system		DC step control
Communication interface		EtherCAT
External digital I/O		Input: 15 points Output: 16 points
Power supply	Voltage input rage	100~240 VAC
	Frequency	50/60Hz
Dimensions		440W*285D*450H
Weight		24kg

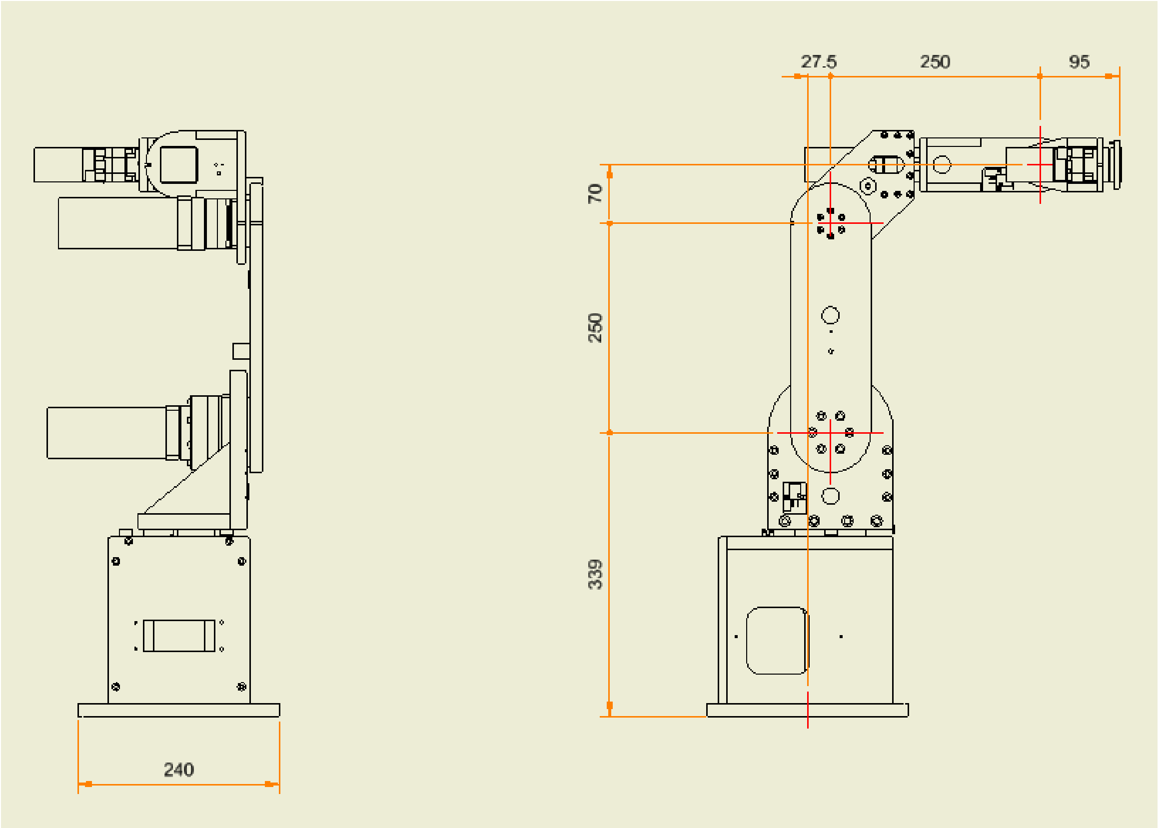
1.3. The Robotic Arm and Its Range of Motion

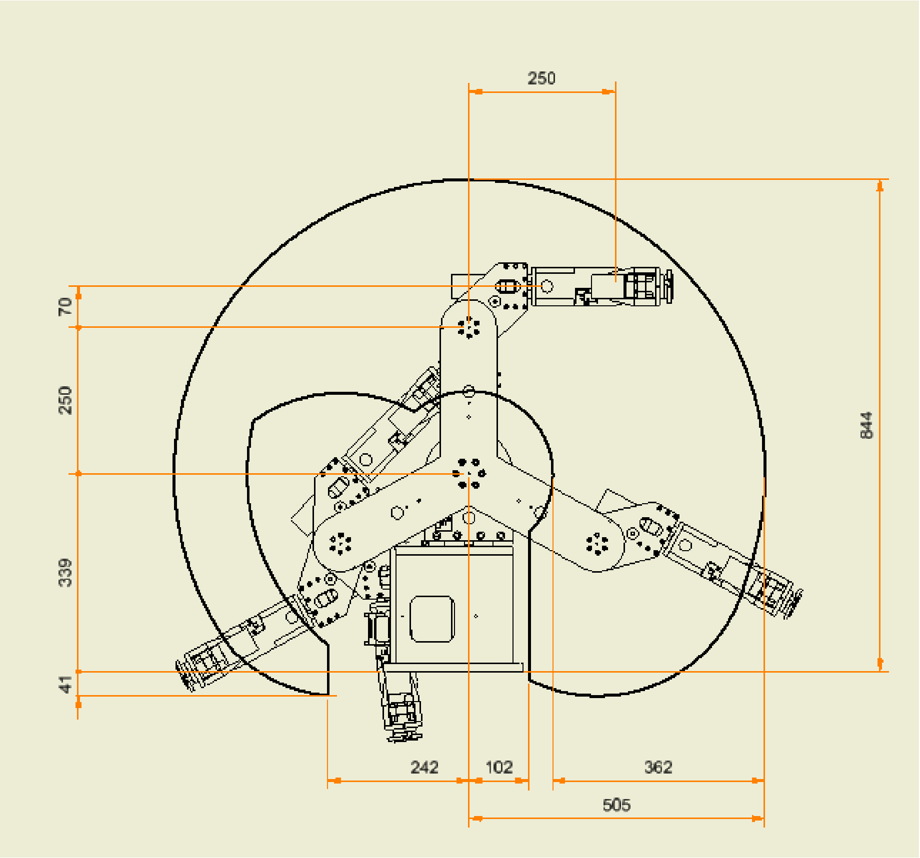
The body of the robotic arm is shown below:

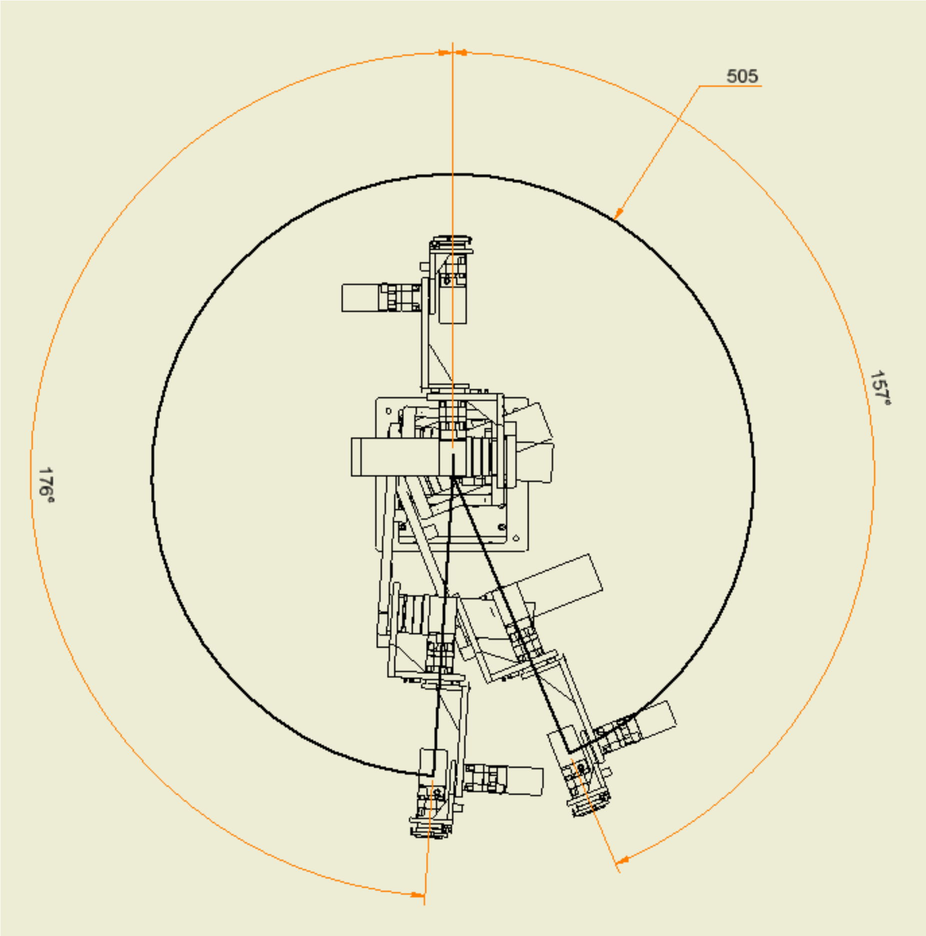
The Exteriors



3D Mechanism Schematics

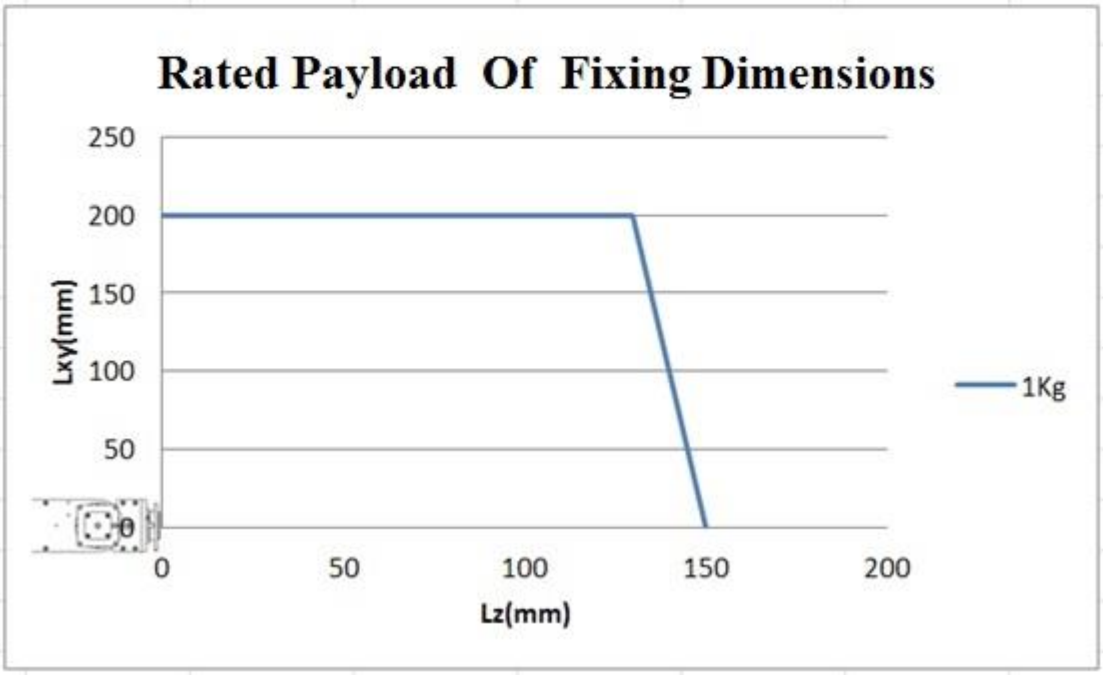

27.5
250
95
70
250
339
240


250
70
250
339
41
844
242
102
362
505


505
157°
176°

1.4. Rated Payload at the End of Robotic Arm

This robotic arm is designed with an effective payload of 1 kg. The figure below shows the location of center of gravity when the 1kg payload and the distance limits and range of motion for the tip of the arm. The horizontal axis indicates the distance in X-direction (or Y-direction) parallel to the flange surface at the tip of the arm, whereas the vertical axis indicates the distance in Z-direction perpendicular to the flange surface at the tip of the arm, both in mm.



Attention is also needed that the range of payload for the arm is related to the current posture of the arm as well as its velocity and acceleration when it moves around. For overloading during a motion (the driver overload alarm will occur in this case), it can be dealt with by adjusting the arm's posture and/or reducing acceleration or deceleration.

1.5. External Dimensions of Electric Control Box

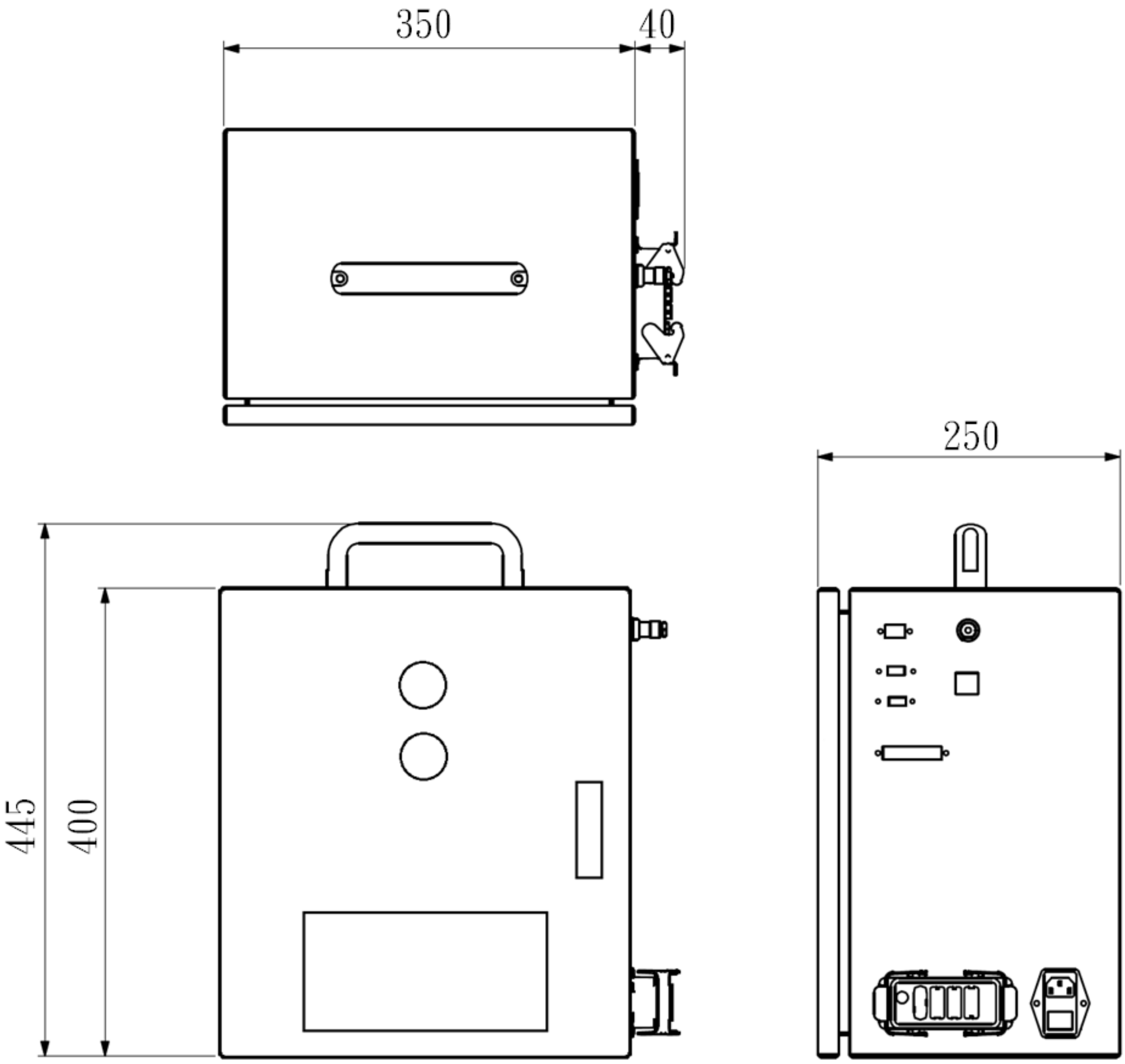
The electric control box of miniBOT looks like this:

Front

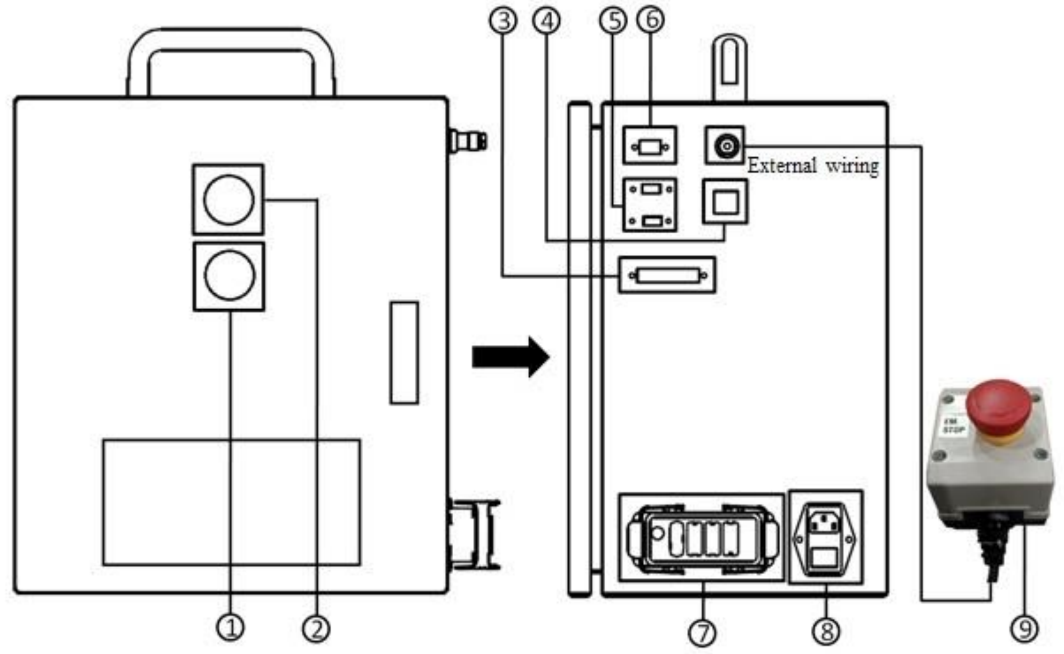
Right side

3D view

The dimensions of the electric control box in mm are shown below:



1.6. Functions Located on the Electric Control Box

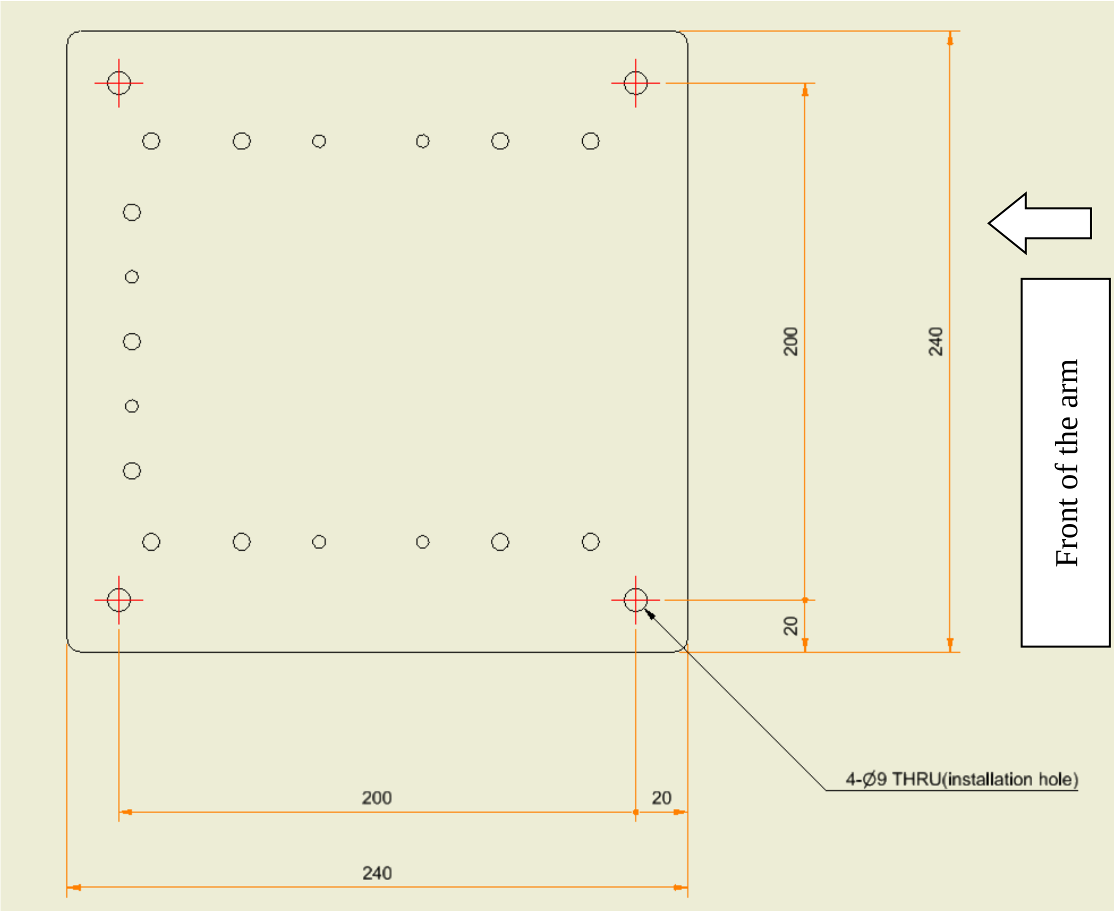


No.	Name	Function
1	Power switch for drive motor	Turn on/off the power to step drive motor
2	Emergency stop button	Push to stop the arm in emergency
3	External digital I/O port	Insert the external digital I/O cable here
4	Network cable port	Insert the network cable here for the control computer
5	USB port	USB port for the control computer
6	Monitor cable port	Insert the monitor cable here for the control computer
7	Robotic arm port	Insert the connection cable between the arm and electric control box here
8	Power and power switch	Utility power, single-phase 100~240VAC, 50/60Hz
9	External emergency stop button	Push to stop the arm in emergency

2. Installation

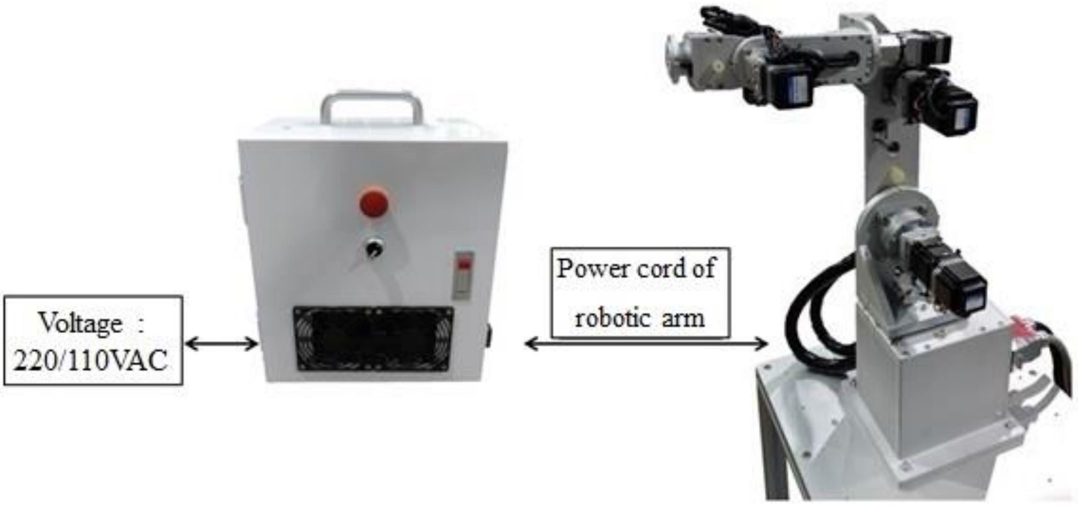
2.1. Installation of the Arm and Dimensions of Base

The installation base is shown below.



2.2. Install the Controller

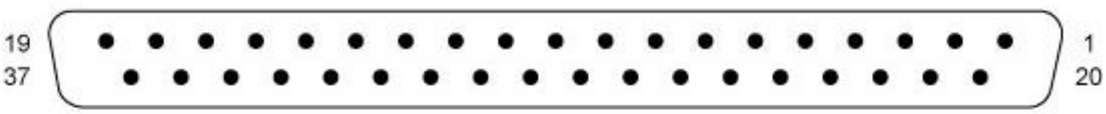
The robotic arm is installed as shown below:



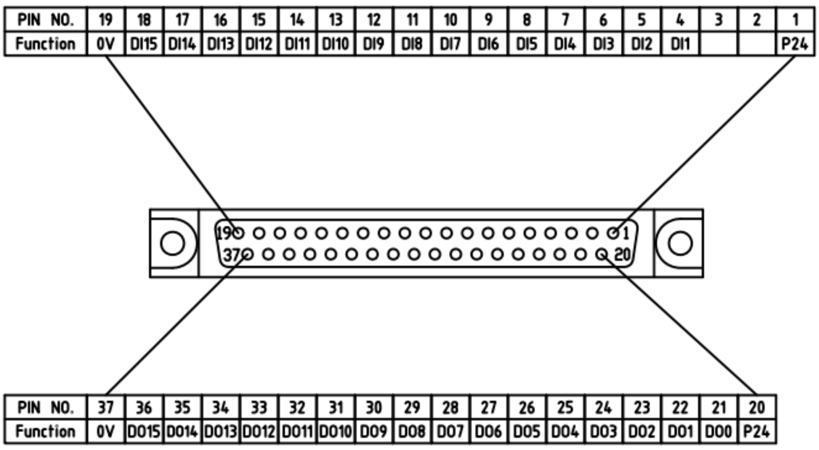
2.3. Pins and Connection for the Digital I/O

The electric control box is provided with 15 DI points and 16 DO points. The expansion I/O connector is located on the right side of the box, as shown below:

The I/O port is described as follows:



DC-37S (Female Socket Front View)



The pins are defined as follows:

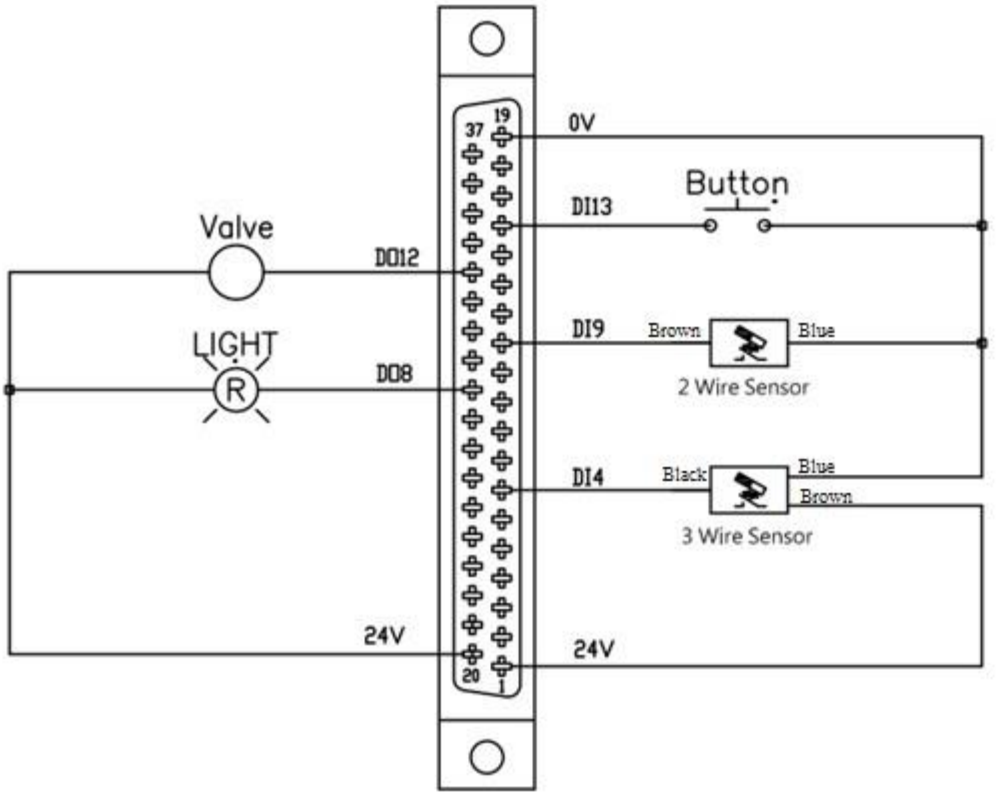
Pin	Parameter	Pin	Parameter
1	P24 (+24V)	20	P24 (+24V)
2	Reserved	21	DO[0]

3	Reserved	22	DO[1]
4	DI[1]	23	DO[2]
5	DI[2]	24	DO[3]
6	DI[3]	25	DO[4]
7	DI[4]	26	DO[5]
8	DI[5]	27	DO[6]
9	DI[6]	28	DO[7]
10	DI[7]	29	DO[8]
11	DI[8]	30	DO[9]
12	DI[9]	31	DO[10]
13	DI[10]	32	DO[11]
14	DI[11]	33	DO[12]
15	DI[12]	34	DO[13]
16	DI[13]	35	DO[14]
17	DI[14]	36	DO[15]
18	DI[15]	37	0V
19	0V		

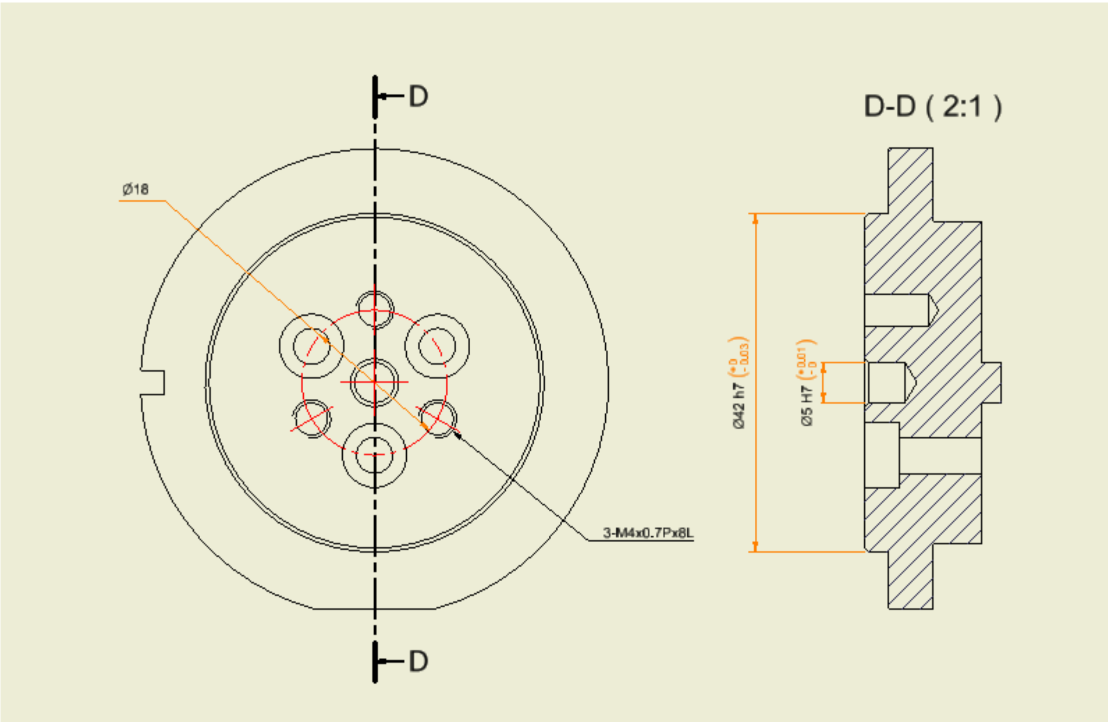
Note:

DI[0]: is predefined for the input for the detection of emergency stop button activation.

DI/DO wiring example:



2.4. Installation of Flange



2.5. Starting Sequence for the Electric Control Box

The following are the steps to start the electric control box:

Step 1:

Flip the general power switch located on the side of the box to the right to turn on and left to turn off.

Step 2:

Flip the power switch of the driver on the side of box to the right to turn on and left to turn off.

Step 3:

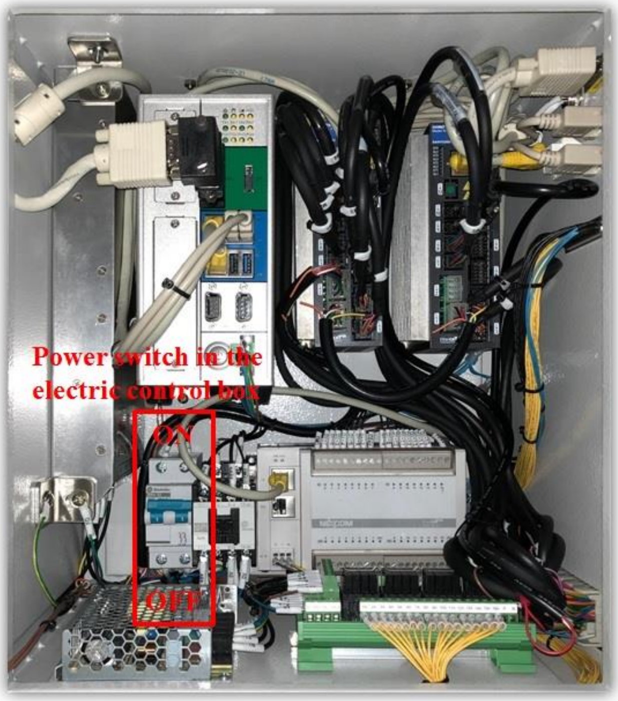
Push both the emergency stop button on the outside of the box and the external emergency stop button to activate and release to deactivate.

Step 4:

Use the program to control the robotic arm.

Reverse the steps mentioned above to turn the electric control box off.

In addition, if it is impossible to turn the power on, check if the no-fuse circuit breaker in the control box is ON.
This breaker is preset at normally ON at factory.



Flip the power switch in the electric control box up to turn it on and down to turn it off.

3. Program Operations

3.1. Working with the NexMotion Studio

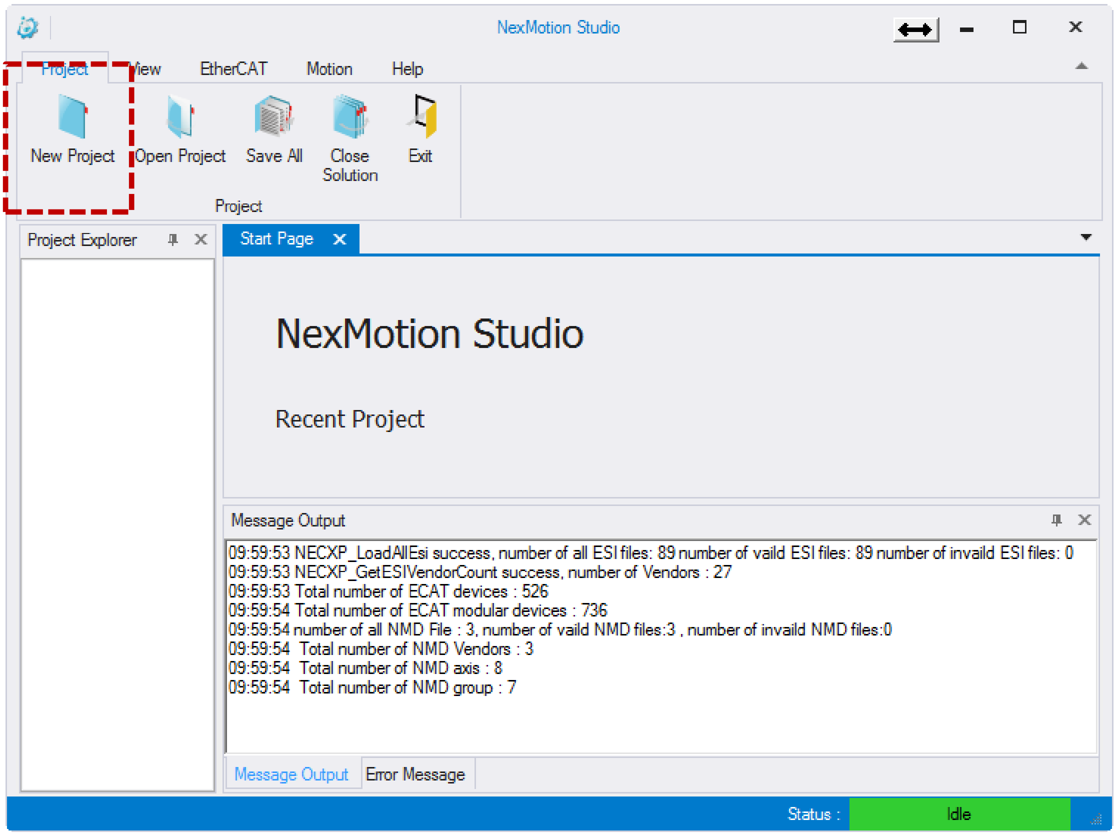
With the miniBOT installed, the utility program of NexMotion Studio is used for the following tests:

1. Check if it is possible to communicate with the motor drive through EtherCAT.
2. Homing of axes
3. Motion test for each axis
4. Operation test for the motions of miniBot

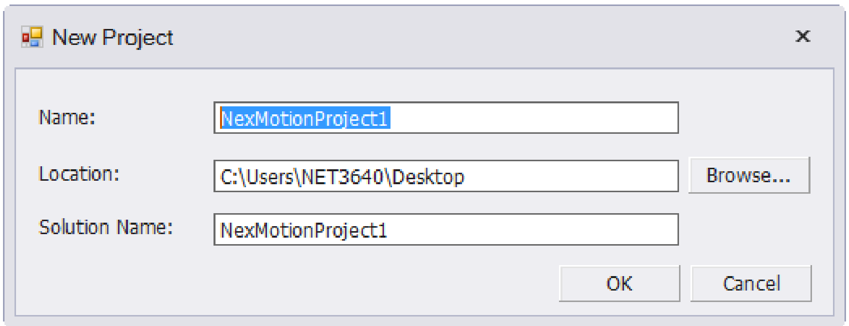
3.1.1. Establish a miniBOT project process

This section describes how to establish a project using the NexMotion Studio and how to test that the controller and motor drive are operating correctly. The following are the steps:

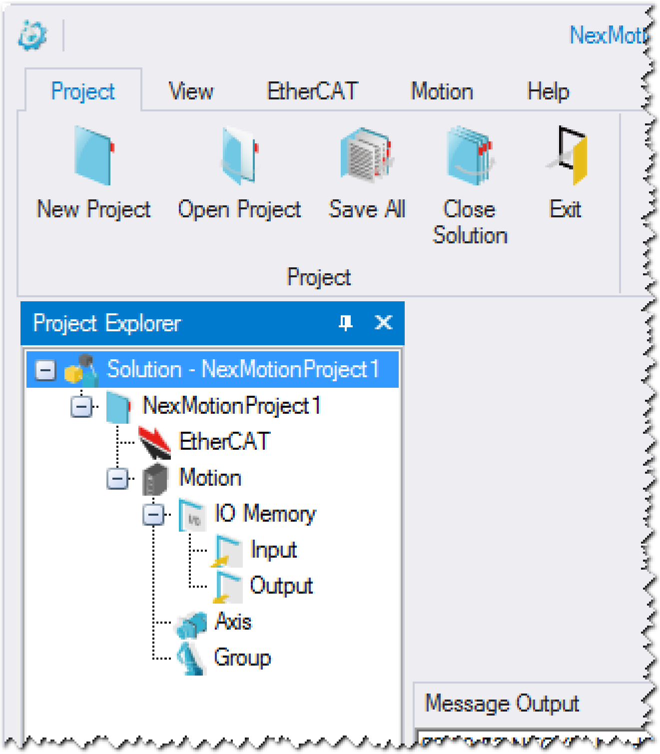
- Start the NexMotion Studio, and the homepage comes up as follows. Click on "New Project" to add a new project for controller settings.



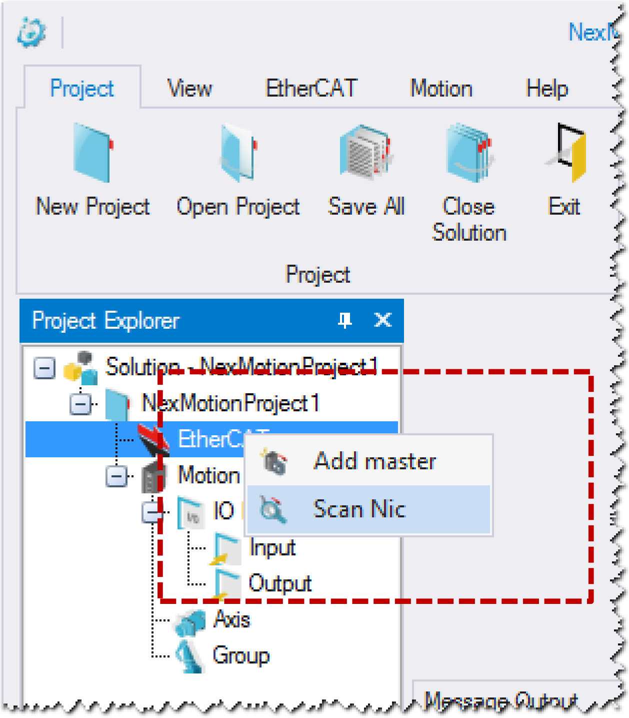
- To add a new project, you need to define its name, location and solution name. If no additional setting is made, the default name will be NexMotionProject1 and it is placed on the desktop as shown below.



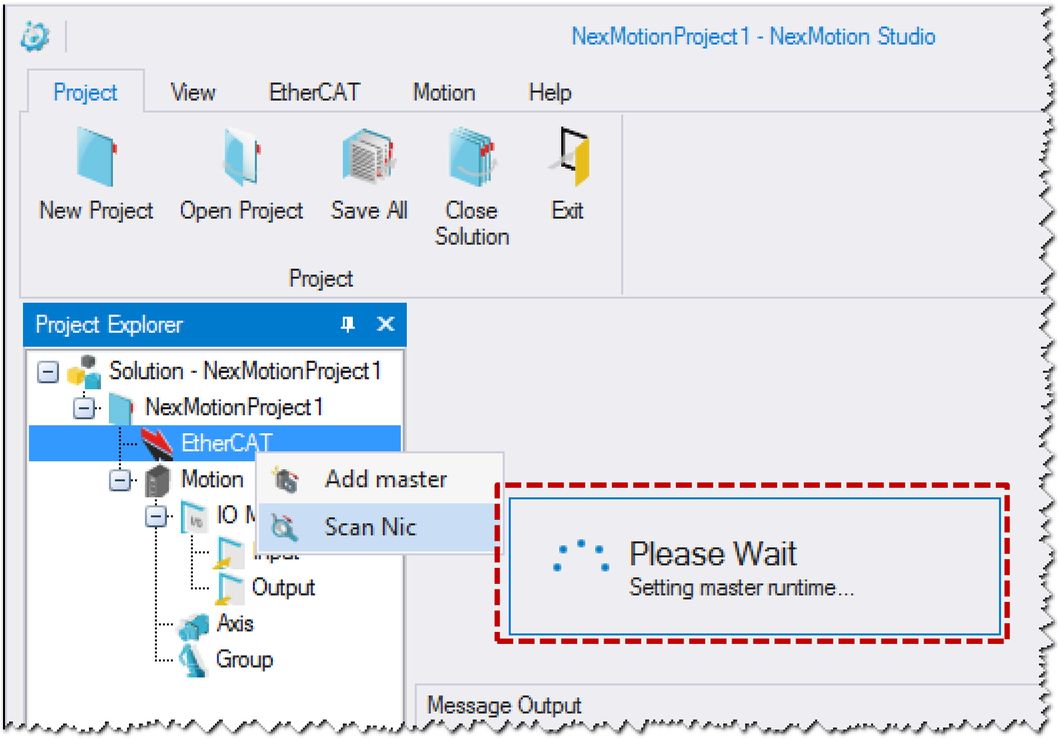
- As a new project is added, the Project Explorer will display the directory tress on screen as follows. It provides the functions of EtherCAT and Motion.



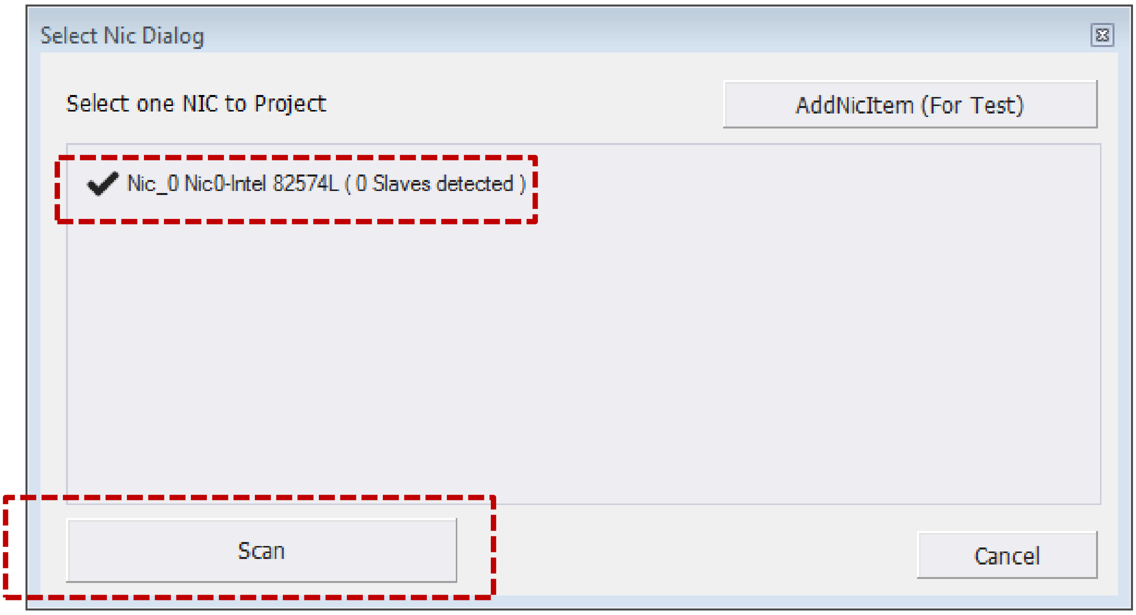
- First scan the equipment connected to the EtherCAT card. Right-click on “EtherCAT” and select “Scan Nic” as shown below.



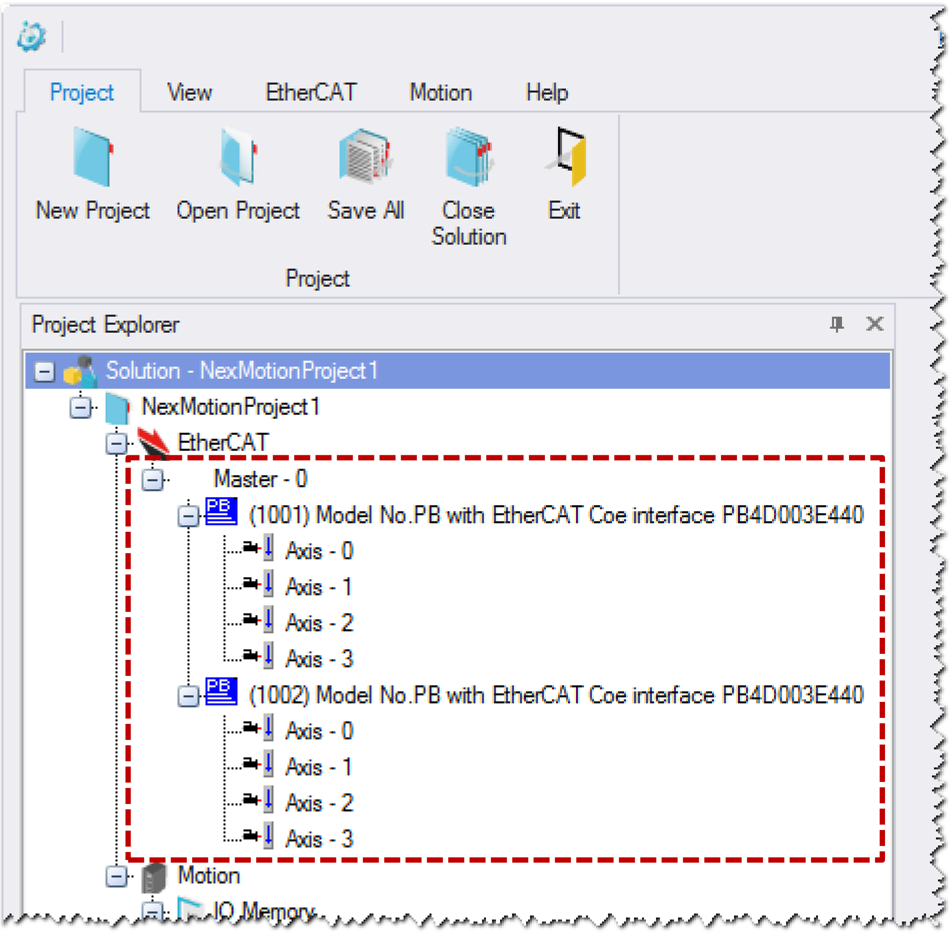
● As the controller's EtherCAT communication is activated, the following "Please Wait" message will show up.



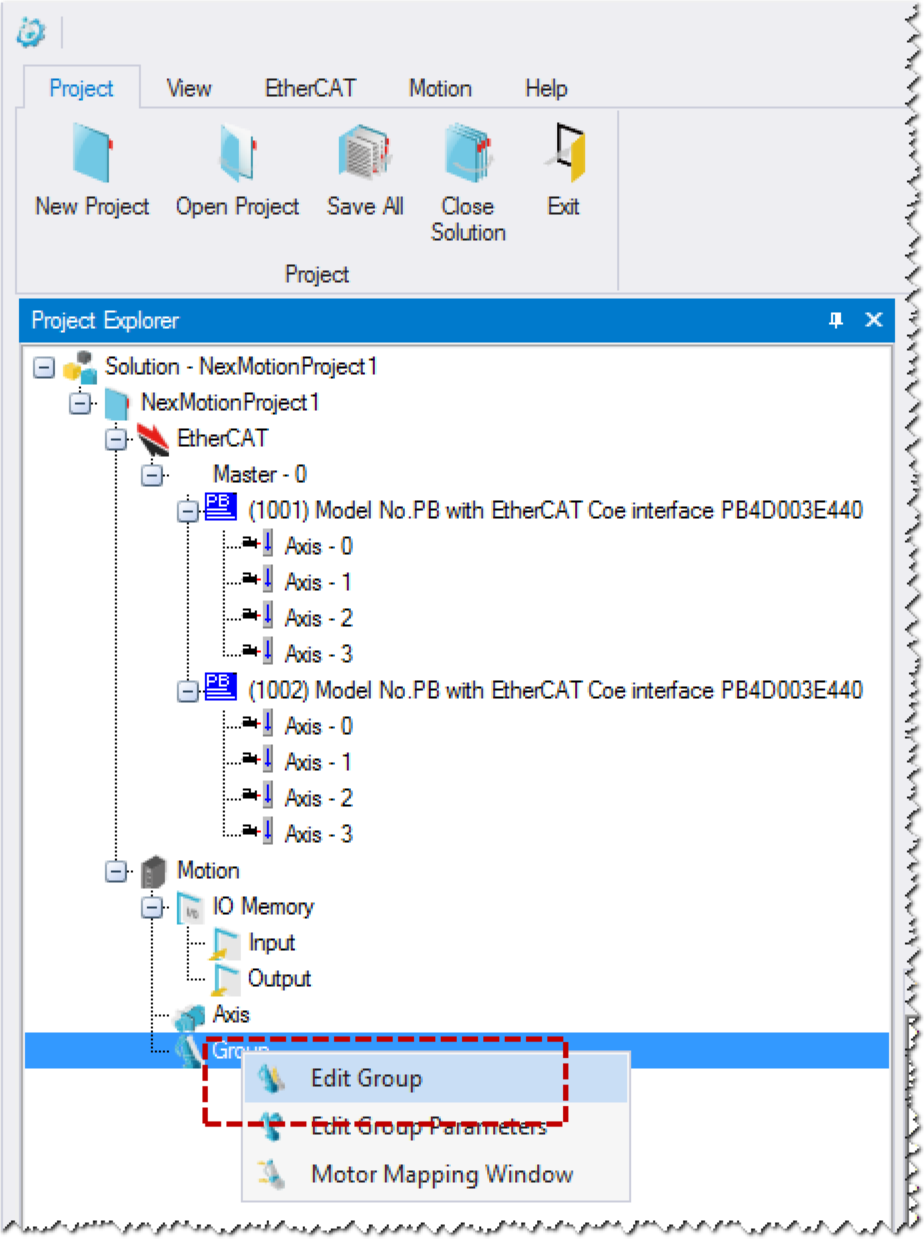
● As it is activated, a dialog box will pop up showing a list of available EtherCAT cards. If there is only one card, that card will be selected as default as shown below. If there are multiple cards, then select as needed. Click “Scan” after the selection is made.



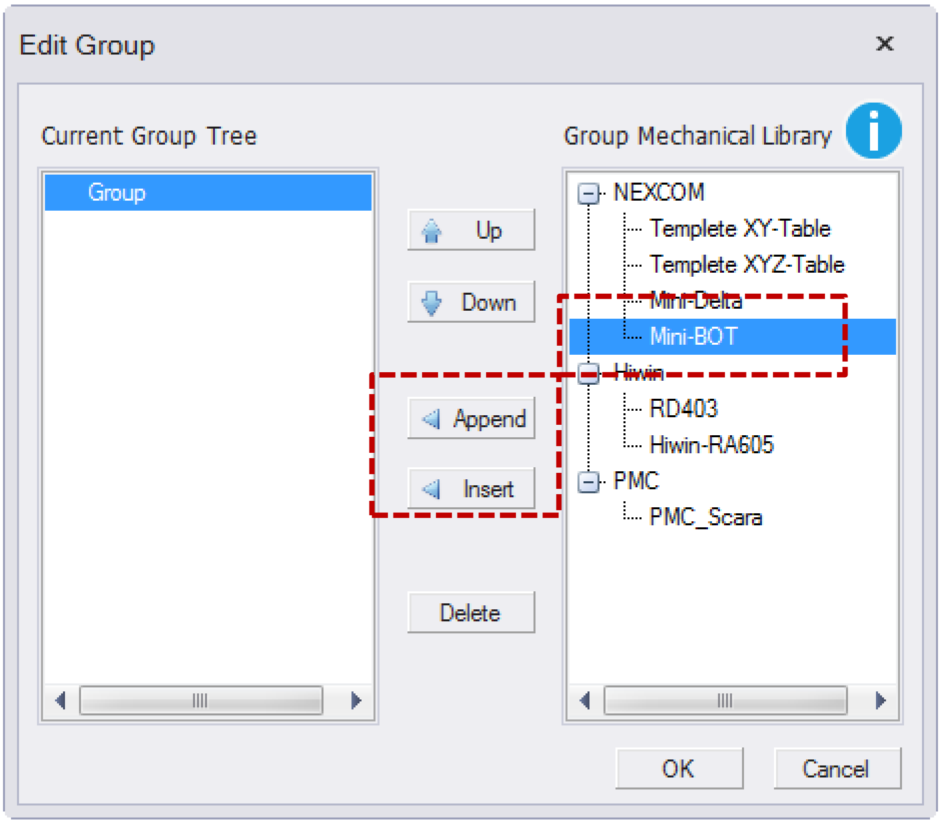
- Once scanned, the equipment connected is shown below “EtherCAT” in the directory tree. We can see that there are two drivers for the miniBot, as shown below.



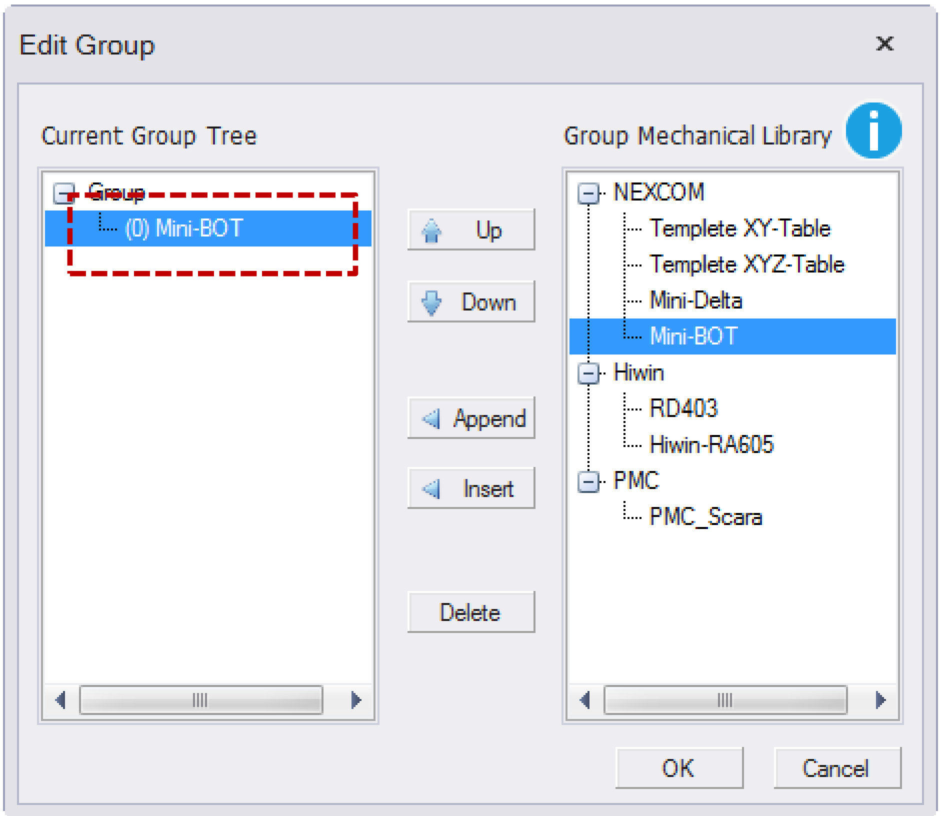
- As the equipment is scanned, right-click "Group" in the directory tree since the robotic arm is a motion configuration of group relationship. Select "Edit Group" to add the robotic arm module, as shown below.



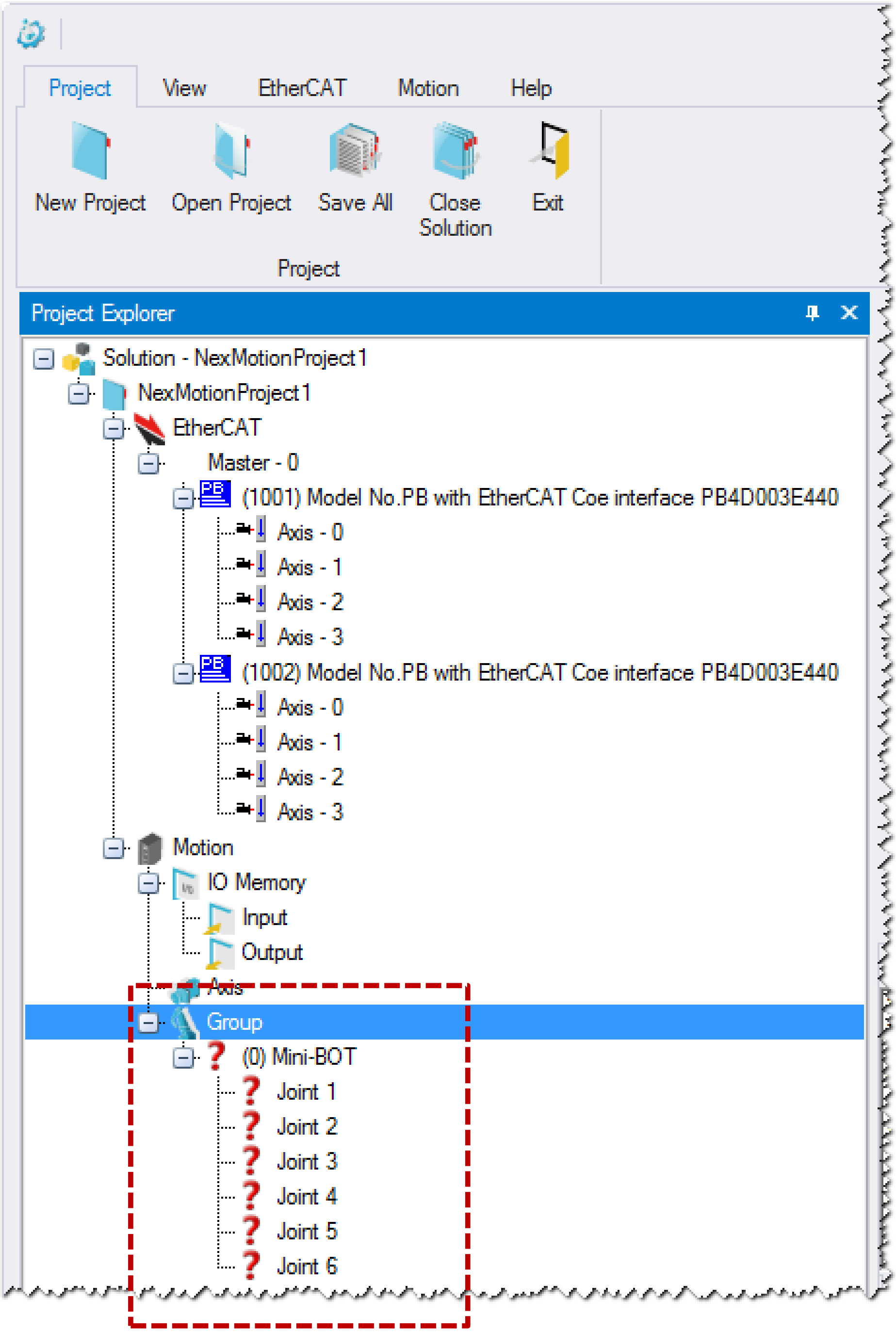
- The dialog box showing the list of the robotic arm modules supported by the controller pops up as follows. Select “Mini-BOT” and then “Append” or “Insert” to add it to the Group.



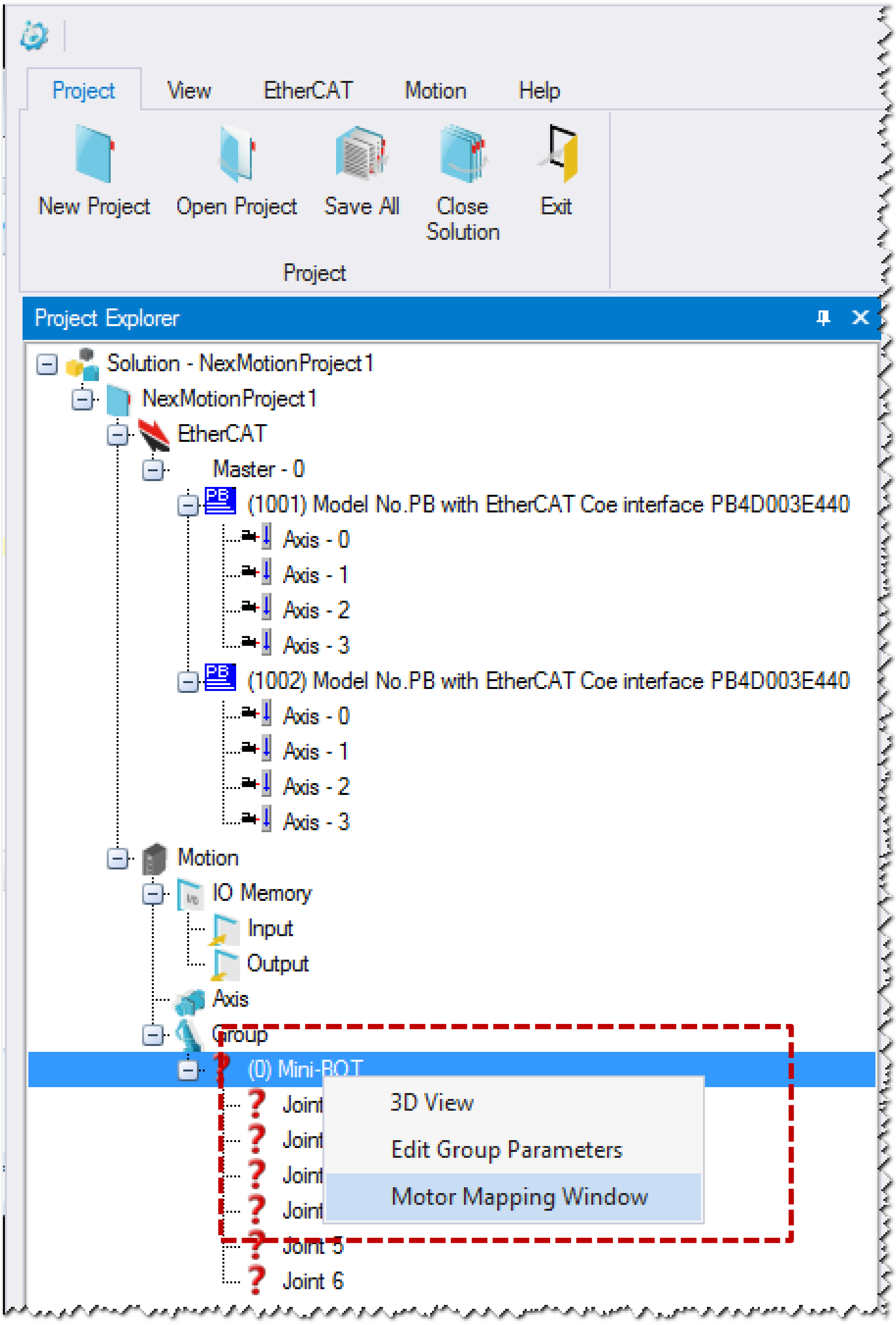
- The "Mini-BOT" is added to the Group as shown below. Click on "OK" to close this dialog box.



- Mini-BOT is shown under the Group in the directory tree, and it comes with 6 joints marked with question mark, as shown below.



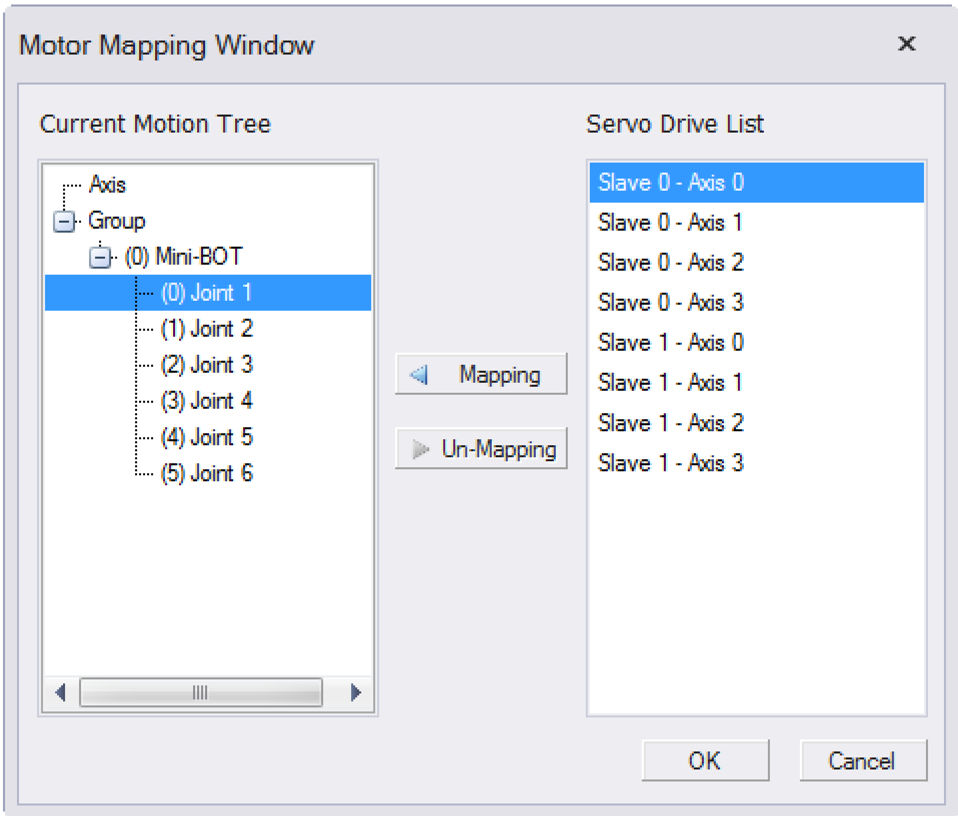
- Now it is time to define the relationship between the joints of the robotic arm and the motors. Right-click "Mini-BOT" in the directory tree and select "Motor Mapping Window" as shown below.



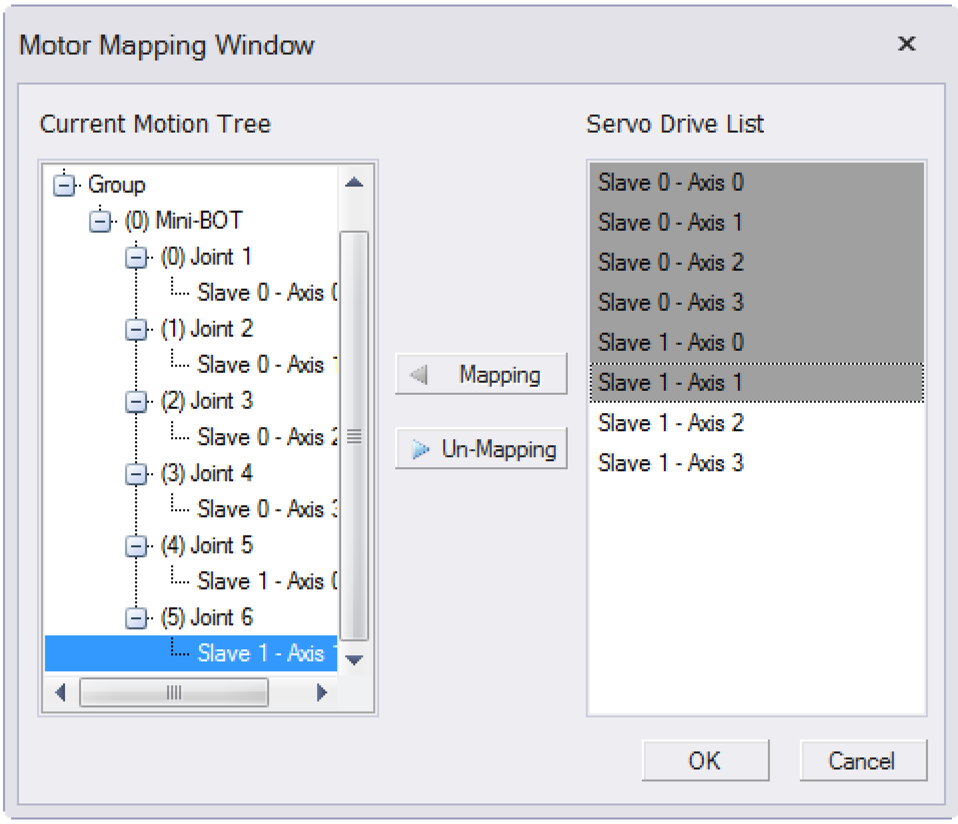
- The dialog box showing the relationship between the robotic arm joints and motors pops up. The actual wiring between the arm and motors is provided in the table below. Match the motors on the right to the joints on the left using "Mapping," as shown below.

Joint	Drive	Motor
1	Drive 1	Axis 1
2		Axis 2

3		Axis 3
4		Axis 4
5	Drive 2	Axis 1
6		Axis 2
Not in use		Axis 1
Not in use		Axis 2



- The relationship between the robotic arm joints and motors is established after the mapping, as shown below. Click on "OK" to close this dialog box.


Motor Mapping Window
Current Motion Tree
Group
(0) Mini-BOT
(0) Joint 1
(1) Joint 2
(2) Joint 3
(3) Joint 4
(4) Joint 5
(5) Joint 6
Mapping
Un-Mapping
Servo Drive List
Slave 0 - Axis 0
Slave 0 - Axis 1
Slave 0 - Axis 2
Slave 0 - Axis 3
Slave 1 - Axis 0
Slave 1 - Axis 1
Slave 1 - Axis 2
Slave 1 - Axis 3
OK
Cancel

3.1.2. Homing process

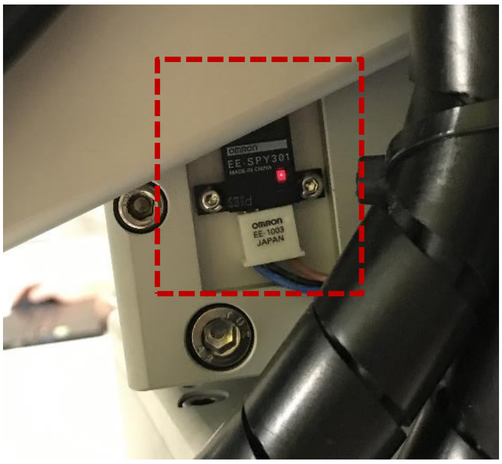
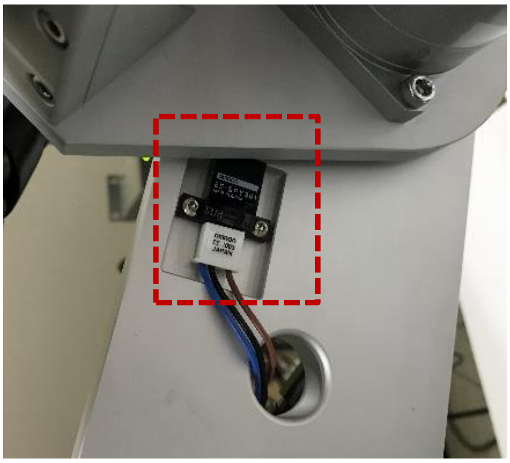
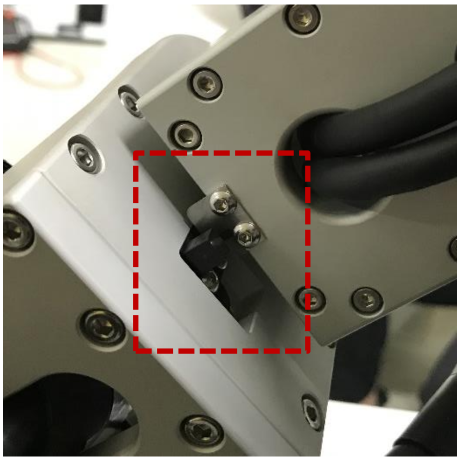
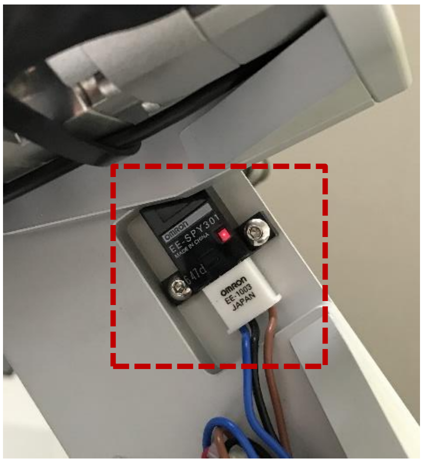
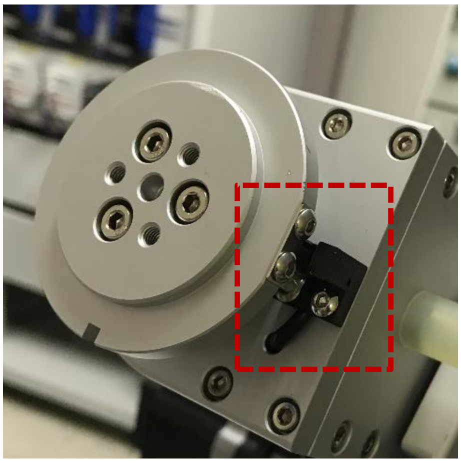
As the project settings are done in the previous section, this section talks about homing the robotic arm. Before the homing starts:

1. Check that all homing parameters are properly defined;
2. Activate the controller (Operation)
3. Group Drive Enable
4. Activate homing

Assuming that the Mini-BOT is any posture as the controller is powered up, one example is shown in the left picture below. However, the actual angles obtained from the joints are all zero. The homing is performed to ensure that all joints are at the correct angles. The robotic arm becomes what you see in the right picture shown below as the homing is done.

Before homing	After homing
	

As the homing is done, check that the homing sensors at the joints are in position, as shown below.

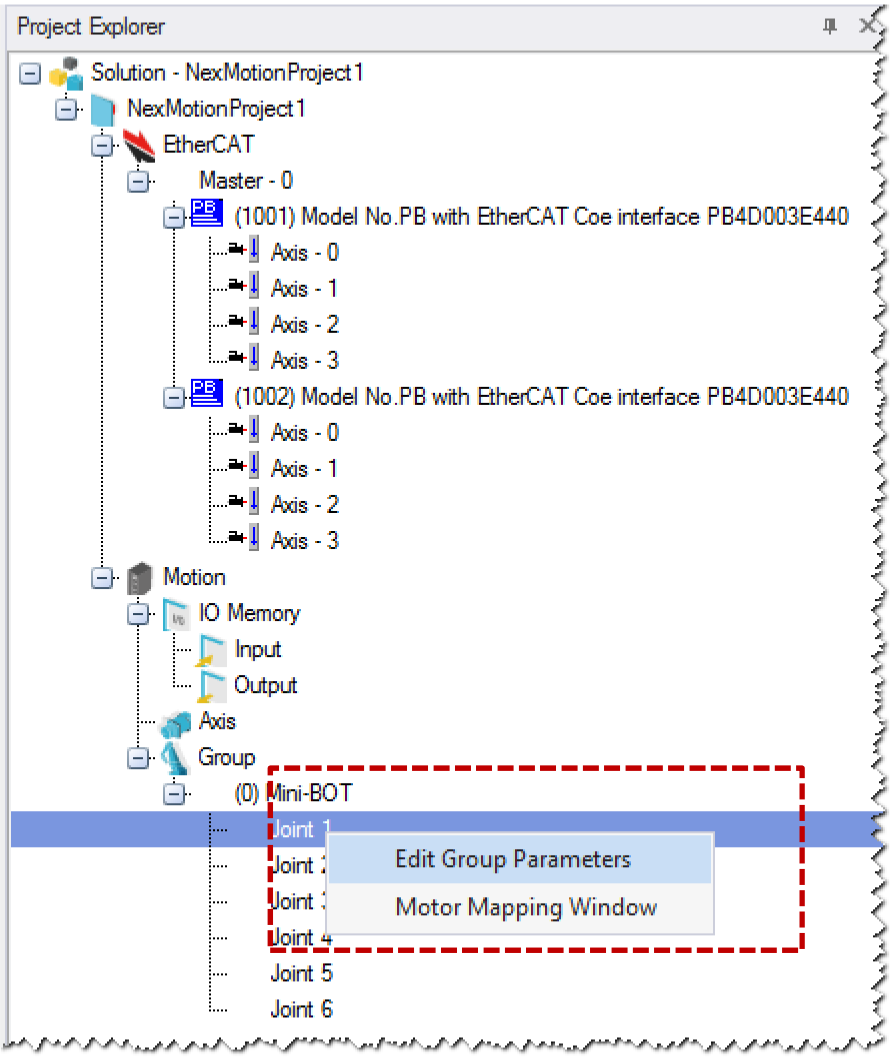
Axis 1	Axis 2
	
Axis 3	Axis 4
	
Axis 5	Axis 6
	

- Next, define the homing motion parameters (parameters of axes in the group) for the joints. The table below provides example parameters values for homing.

Users are allowed to make appropriate adjustments according to the machine or applications.

Parameters:	0x80	0x81	0x82	0x83	0x84
Joint 1	20	30	5	60	172
Joint 2	20	30	5	60	196.69
Joint 3	22	30	5	60	-60
Joint 4	20	30	5	60	175
Joint 5	22	30	5	60	-100.44
Joint 6	20	30	25	60	0

- Right-click a joint, for example "Joint 1," in the directory tree and select "Edit Group Parameters," as shown below.



- The parameter page for Group and joints shows up after the selection is made. Set the joint 1 parameter 0x80 at 20 and 0x84 at 172 as shown below.

Group | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6

Num	Sub	Type	Value	Description
0x28	0x0	F64_T	0	Base velocity
0x30	0x0	I32_T	0	Absolute or relative programming
0x31	0x0	I32_T	0	Profile type
0x32	0x0	F64_T	2	Maximum velocity
0x33	0x0	F64_T	20	Acceleration
0x34	0x0	F64_T	20	deceleration
0x35	0x0	F64_T	500	Jerk
0x36	0x0	I32_T	0	Buffer mode
0x80	0x0	I32_T	20	EtherCAT CiA HOME method
0x81	0x0	F64_T	30	EtherCAT CiA HOME speed search switch
0x82	0x0	F64_T	5	EtherCAT CiA HOME speed search zero
0x83	0x0	F64_T	60	EtherCAT CiA HOME acceleration
0x84	0x0	F64_T	172	EtherCAT CiA HOME offset

- Set the joint 2 parameter 0x80 at 20 and 0x84 at 196.69 as shown below.

Group | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6

Num	Sub	Type	Value	Description
0x28	0x0	F64_T	0	Base velocity
0x30	0x0	I32_T	0	Absolute or relative programming
0x31	0x0	I32_T	0	Profile type
0x32	0x0	F64_T	2	Maximum velocity
0x33	0x0	F64_T	20	Acceleration
0x34	0x0	F64_T	20	deceleration
0x35	0x0	F64_T	500	Jerk
0x36	0x0	I32_T	0	Buffer mode
0x80	0x0	I32_T	20	EtherCAT CiA HOME method
0x81	0x0	F64_T	30	EtherCAT CiA HOME speed search switch
0x82	0x0	F64_T	5	EtherCAT CiA HOME speed search zero
0x83	0x0	F64_T	60	EtherCAT CiA HOME acceleration
0x84	0x0	F64_T	196.69	EtherCAT CiA HOME offset

- Set the joint 3 parameter 0x80 at 22 and 0x84 at -60 as shown below.

Group | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6

Num	Sub	Type	Value	Description
0x28	0x0	F64_T	0	Base velocity
0x30	0x0	I32_T	0	Absolute or relative programming
0x31	0x0	I32_T	0	Profile type
0x32	0x0	F64_T	2	Maximum velocity
0x33	0x0	F64_T	20	Acceleration
0x34	0x0	F64_T	20	deceleration
0x35	0x0	F64_T	500	Jerk
0x36	0x0	I32_T	0	Buffer mode
0x80	0x0	I32_T	22	EtherCAT CiA HOME method
0x81	0x0	F64_T	30	EtherCAT CiA HOME speed search switch
0x82	0x0	F64_T	5	EtherCAT CiA HOME speed search zero
0x83	0x0	F64_T	60	EtherCAT CiA HOME acceleration
0x84	0x0	F64_T	-60	EtherCAT CiA HOME offset

- Set the joint 4 parameter 0x80at 20 and 0x84 at 175 as shown below.

Group | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6

Num	Sub	Type	Value	Description
0x28	0x0	F64_T	0	Base velocity
0x30	0x0	I32_T	0	Absolute or relative programming
0x31	0x0	I32_T	0	Profile type
0x32	0x0	F64_T	2	Maximum velocity
0x33	0x0	F64_T	20	Acceleration
0x34	0x0	F64_T	20	deceleration
0x35	0x0	F64_T	500	Jerk
0x36	0x0	I32_T	0	Buffer mode
0x80	0x0	I32_T	20	EtherCAT CiA HOME method
0x81	0x0	F64_T	30	EtherCAT CiA HOME speed search switch
0x82	0x0	F64_T	5	EtherCAT CiA HOME speed search zero
0x83	0x0	F64_T	60	EtherCAT CiA HOME acceleration
0x84	0x0	F64_T	175	EtherCAT CiA HOME offset

- Set the joint 5 parameter 0x80 at 22 and 0x84 at -100.44 as shown below.

Group | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6

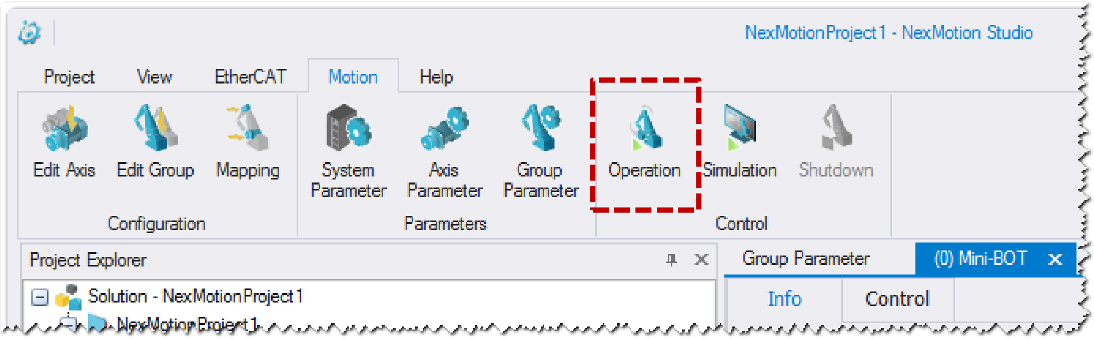
Num	Sub	Type	Value	Description
0x28	0x0	F64_T	0	Base velocity
0x30	0x0	I32_T	0	Absolute or relative programming
0x31	0x0	I32_T	0	Profile type
0x32	0x0	F64_T	50	Maximum velocity
0x33	0x0	F64_T	50	Acceleration
0x34	0x0	F64_T	50	deceleration
0x35	0x0	F64_T	500	Jerk
0x36	0x0	I32_T	0	Buffer mode
0x80	0x0	I32_T	22	EtherCAT CiA HOME method
0x81	0x0	F64_T	30	EtherCAT CiA HOME speed search switch
0x82	0x0	F64_T	5	EtherCAT CiA HOME speed search zero
0x83	0x0	F64_T	60	EtherCAT CiA HOME acceleration
0x84	0x0	F64_T	-100.44	EtherCAT CiA HOME offset

- Set the joint 6 parameter 0x80 at 20 and 0x84 at 0 as shown below.

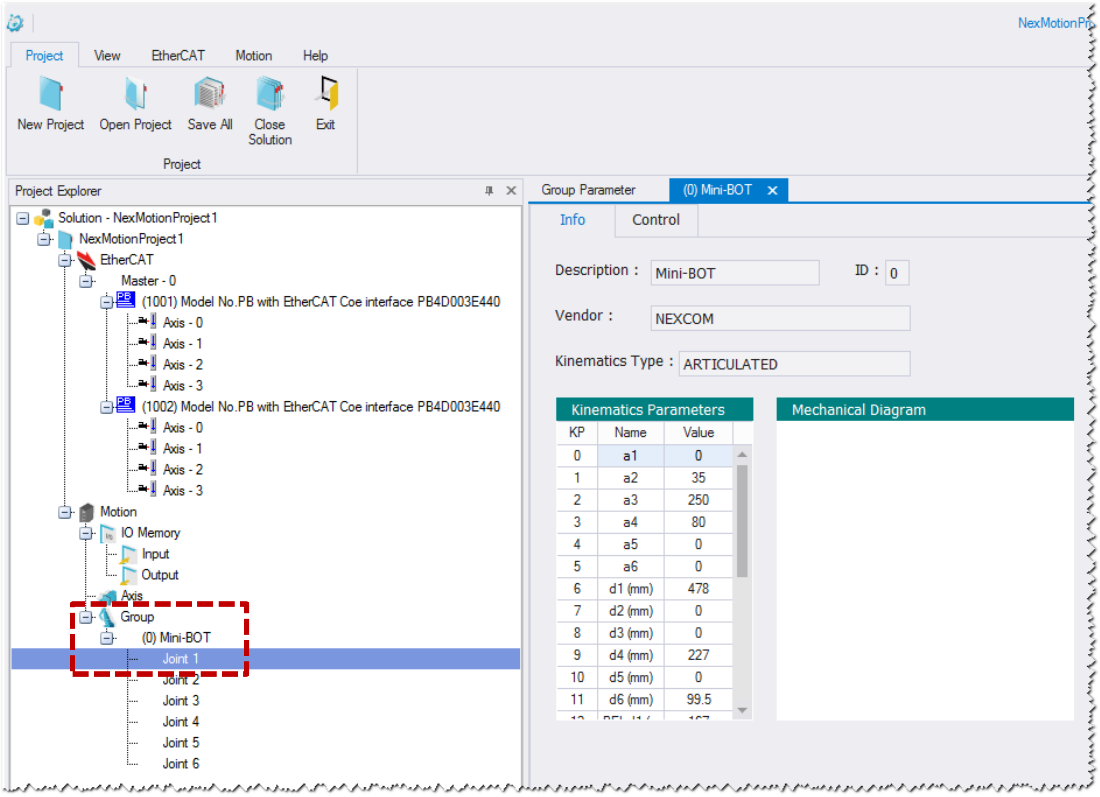
Group | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6

Num	Sub	Type	Value	Description
0x28	0x0	F64_T	0	Base velocity
0x30	0x0	I32_T	0	Absolute or relative programming
0x31	0x0	I32_T	0	Profile type
0x32	0x0	F64_T	60	Maximum velocity
0x33	0x0	F64_T	70	Acceleration
0x34	0x0	F64_T	70	deceleration
0x35	0x0	F64_T	500	Jerk
0x36	0x0	I32_T	0	Buffer mode
0x80	0x0	I32_T	20	EtherCAT CiA HOME method
0x81	0x0	F64_T	30	EtherCAT CiA HOME speed search switch
0x82	0x0	F64_T	25	EtherCAT CiA HOME speed search zero
0x83	0x0	F64_T	60	EtherCAT CiA HOME acceleration
0x84	0x0	F64_T	0	EtherCAT CiA HOME offset

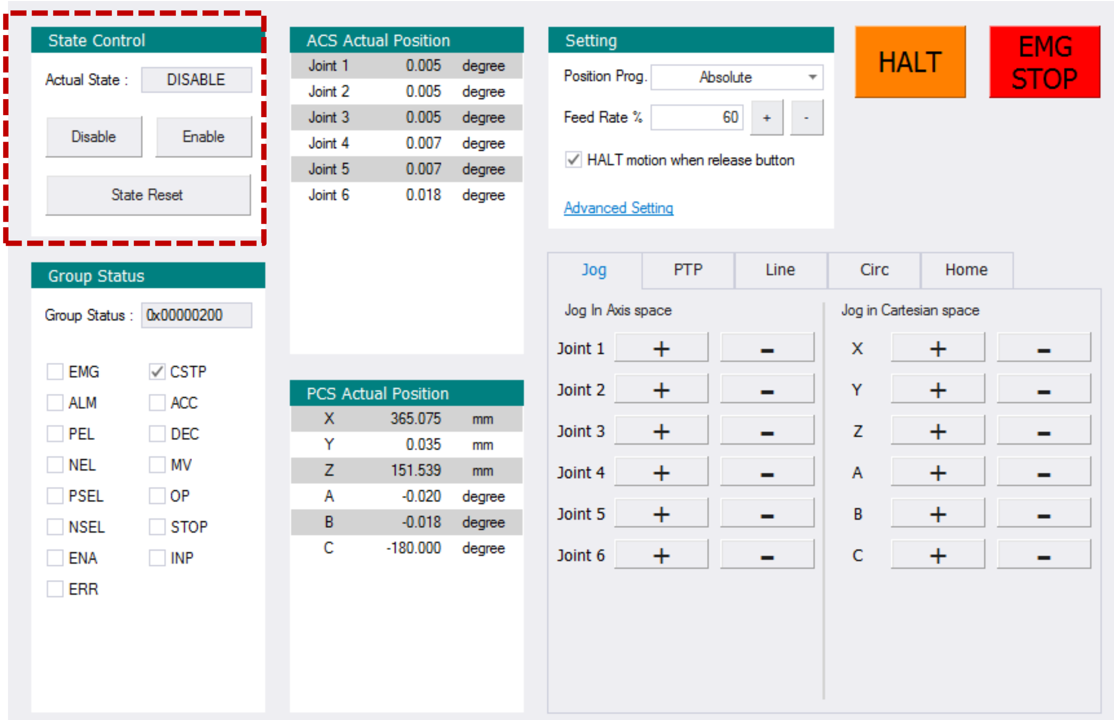
- As the miniBOT project settings are done, click on “Operation” in the “Motion” tab to activate the real-time motion control system of the controller as shown below.



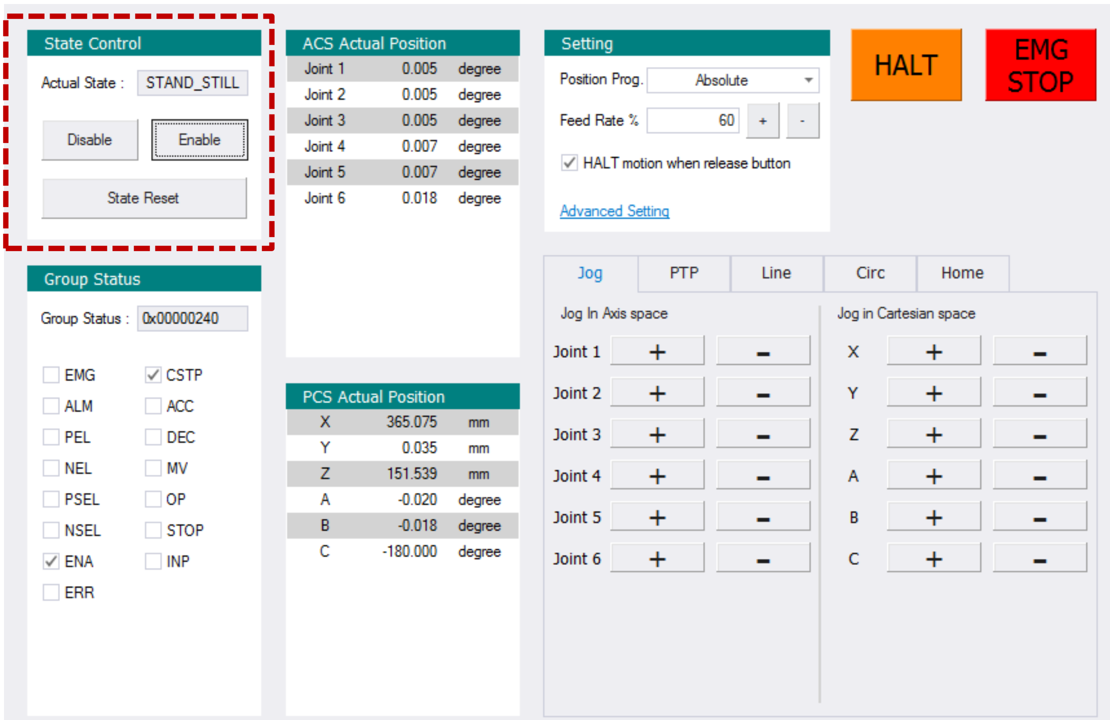
- After activation, double-click on “Mini-BOT” in the directory tree to open the control window of the robotic arm as shown below. The Mini-BOT page shows up on the right and it contains the Info and Control tabs. The Info provides kinematic parameters, whereas the Control controls the motions of the arm.



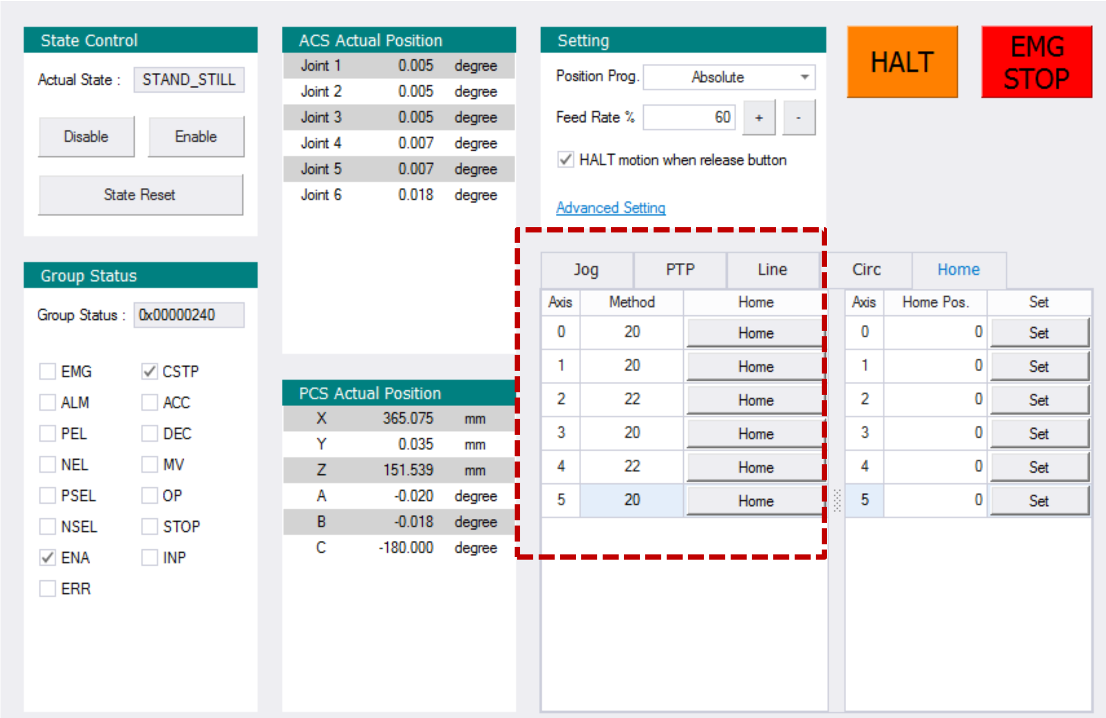
- Select the Control tab and the motion control page shows up as follows. If the drive does not produce any alarm, the actual state will e DISABLE; if it does, the actual state will be ERROR STOP. Now click on “State Reset” to reset the alarm. The state changes to DISABLE as the alarm is reset.



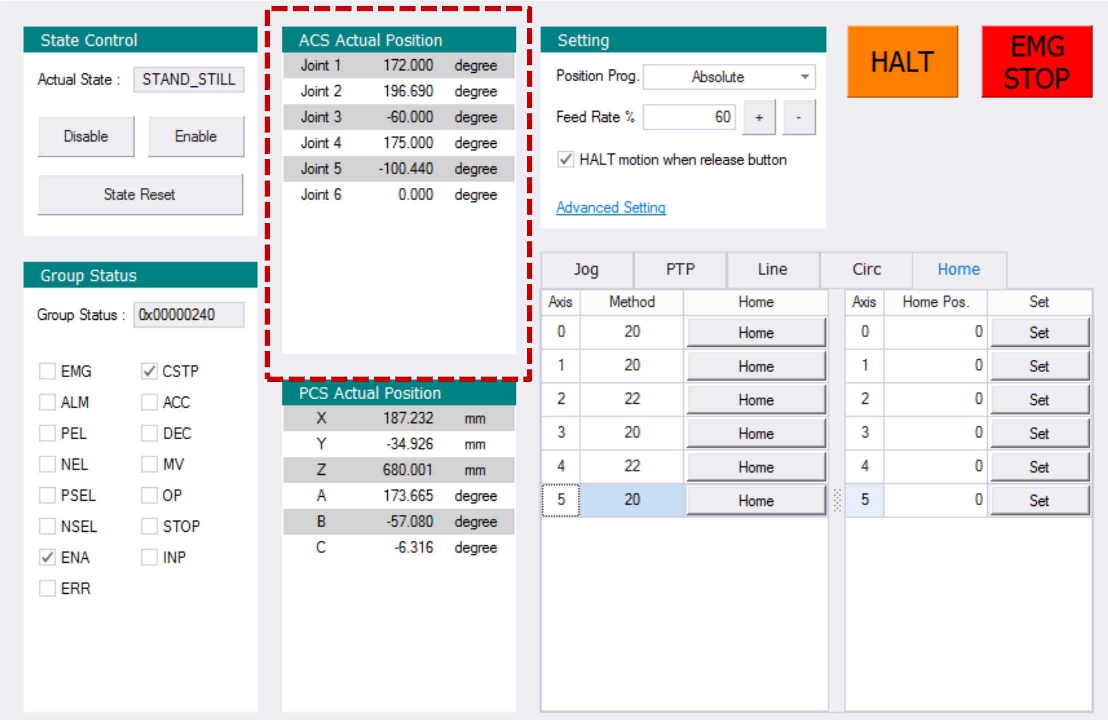
- For any motion control you wish to perform, click on “Enable” first to excite the motors, and the actual state changes to STAND STILL, and the ENA bit of the group state is ON, as shown below.



- Incremental coder is used for the robotic arm, which means that the actual positions are all zero when the power is turned on. That is why homing is required. The homing settings have been defined for the joints above (0x80). Therefore, click on the "Home" tab and the corresponding values will show up in the Method column as shown below.



- Click on the "Home" button for each of the joints for homing. The actual state changes to HOMING. At the end of homing, the ACS actual position of the joint will become the pre-set value (0x84) as shown below.

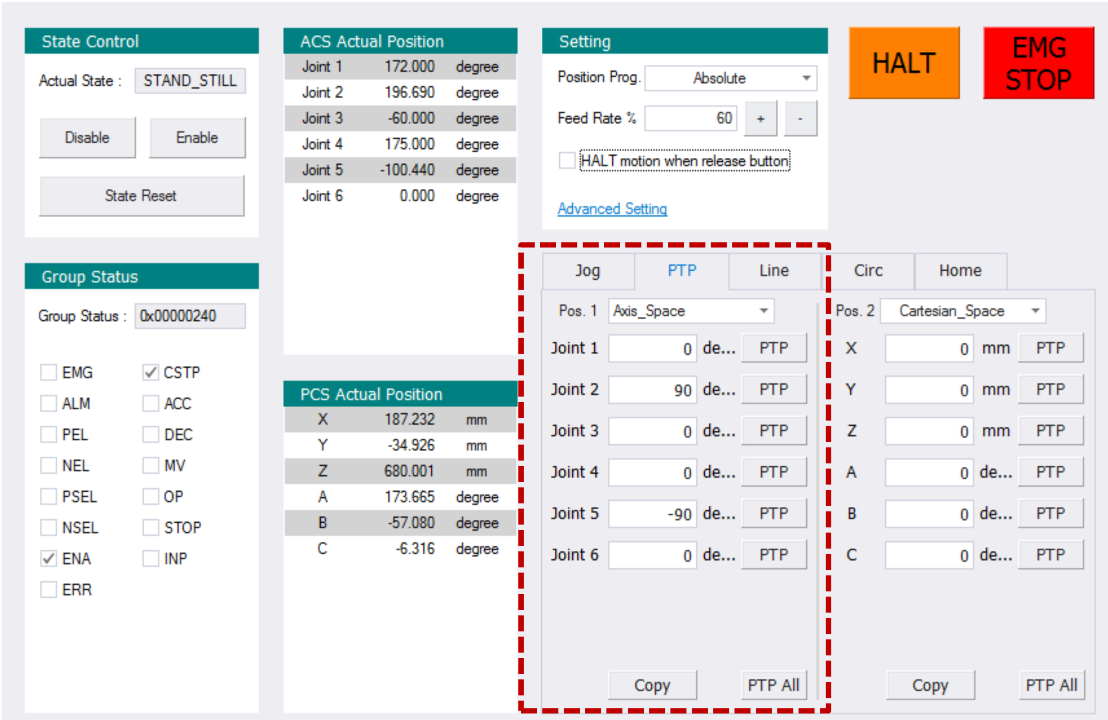


3.1.3. Motion operation process

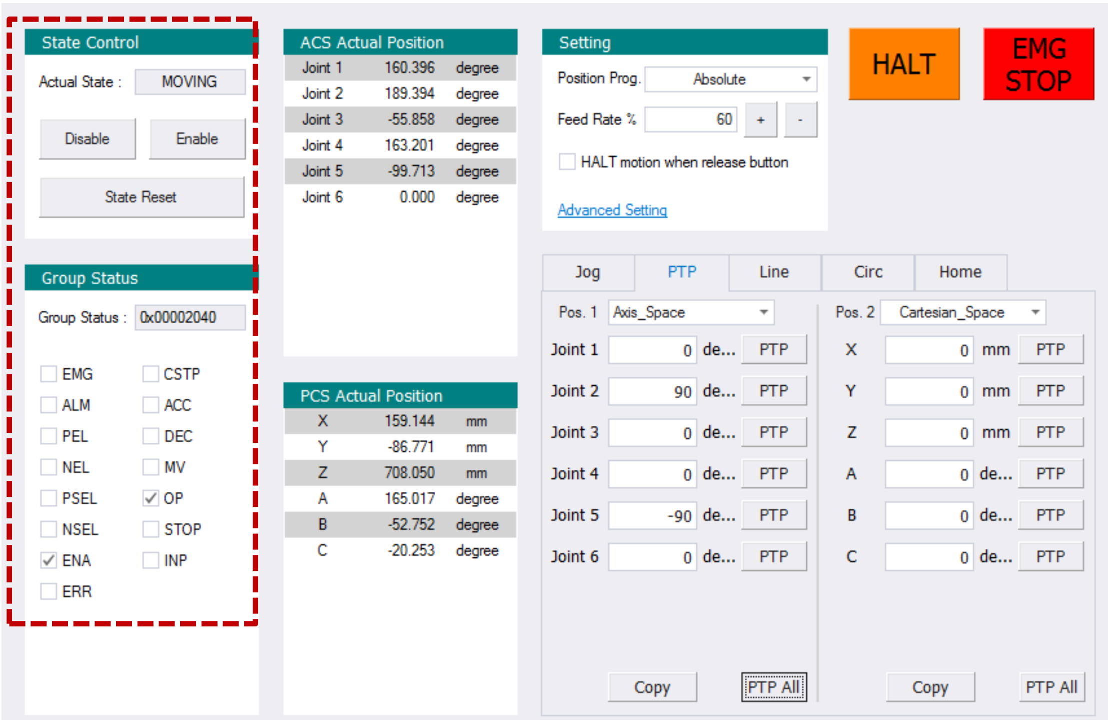
This section explains the motion operations of miniBot through NexMotion Studio, including:

1. Point-to-point (PTP) motion in axis space;
2. Line motion in Cartesian space;
3. Circle motion in Cartesian space.

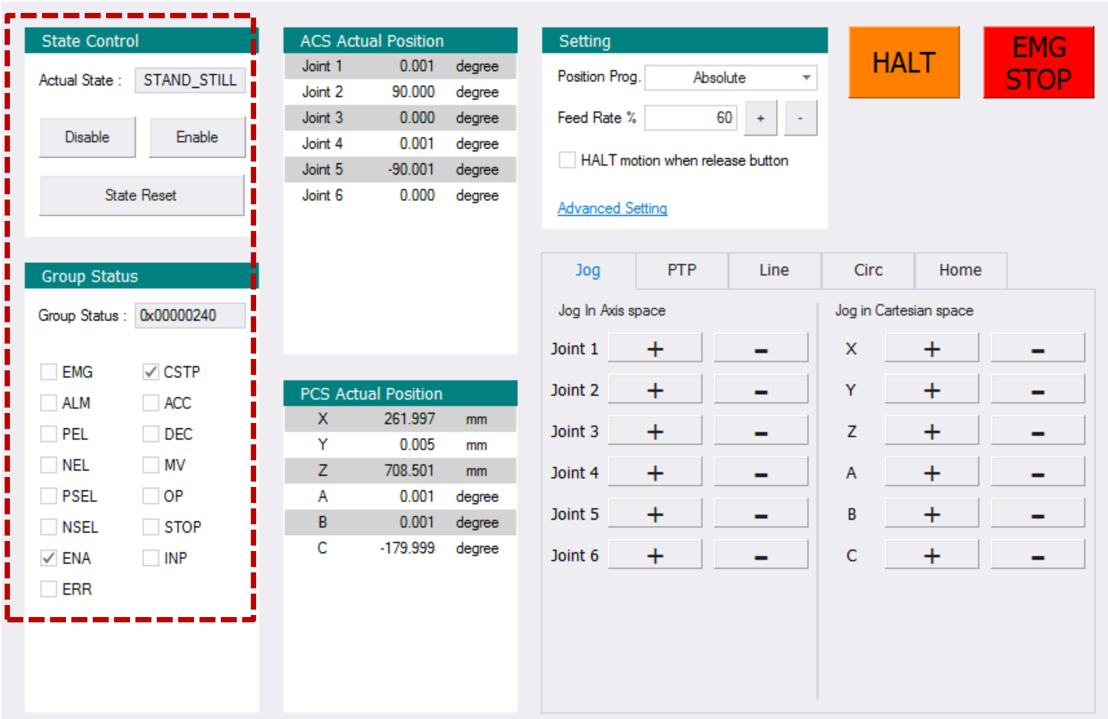
- Move the robotic arm to the required posture using PTP. Click on the “PTP” tab. Enter 90 for joint 2, -90 for joint 5 and 0 for the rest. Click on “PTP All” to execute as shown below.



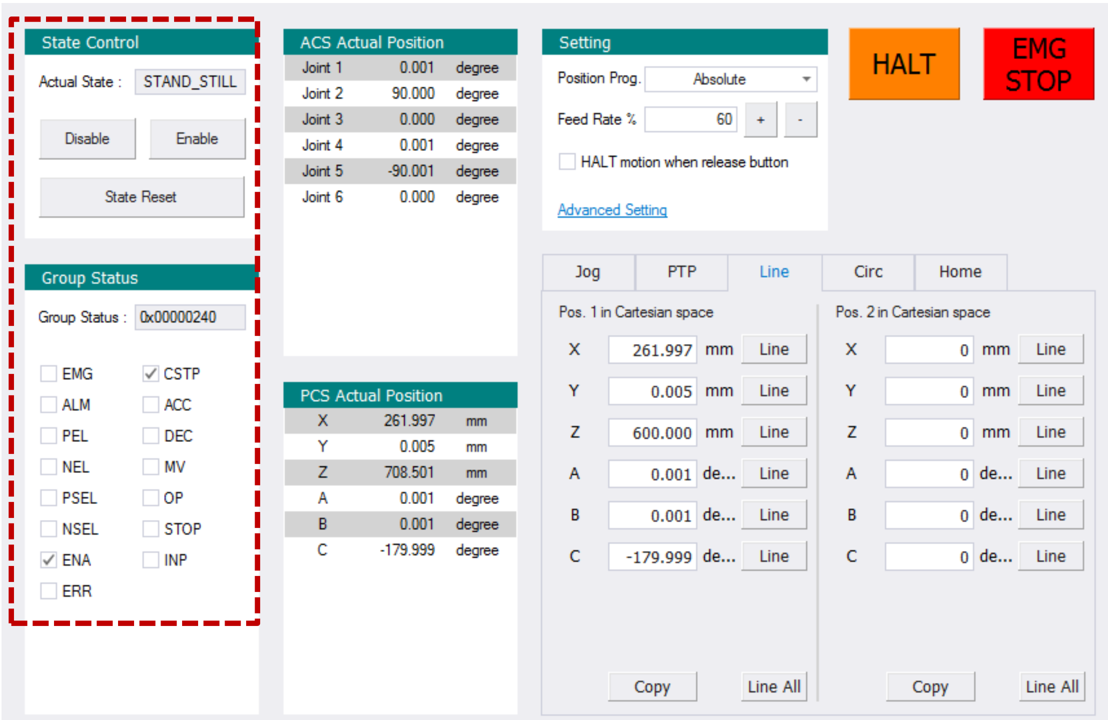
- The actual state turns into MOVING and the OP bit of Group Status into ON. The joints move to the required position as shown below.



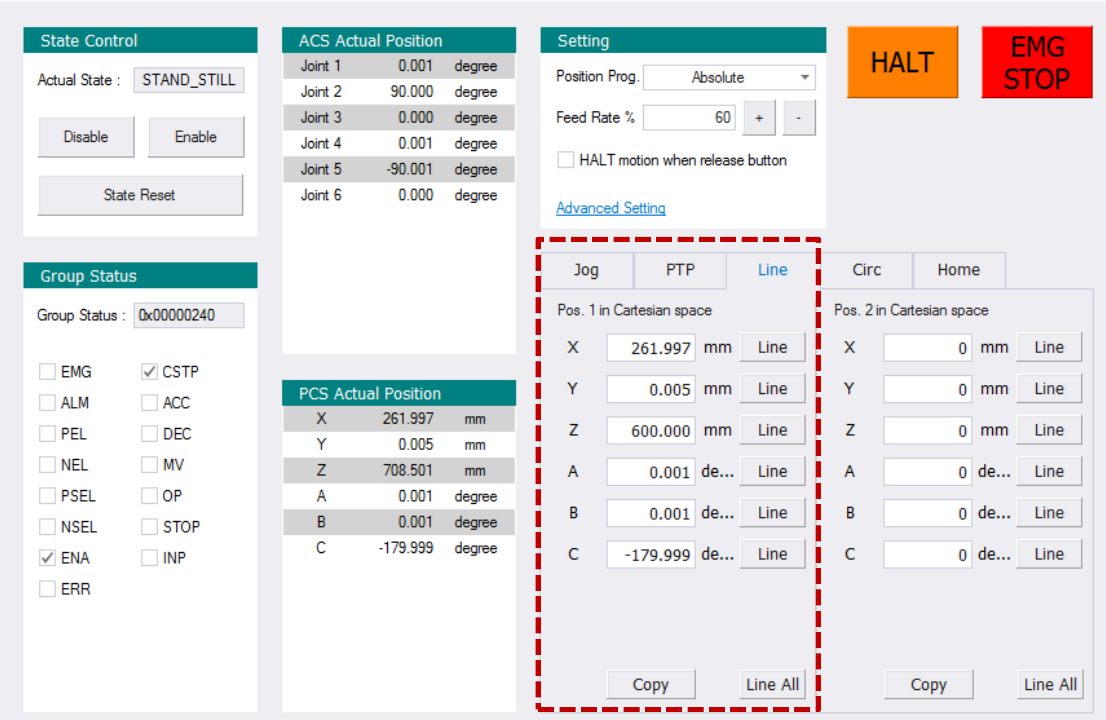
- As the required position is reached, the actual state changes to STAND STILL and the OP bit of Group Status to OFF as shown below.



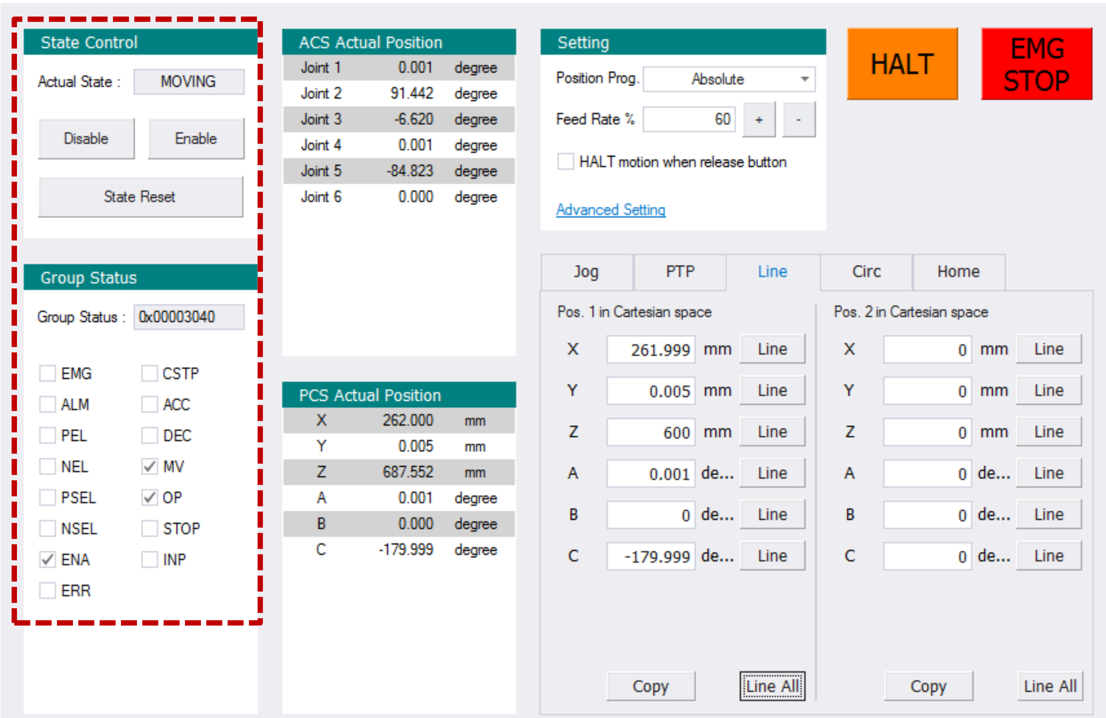
- As the required position is reached, the actual state changes to STAND STILL and the OP bit of Group Status to OFF as shown below.



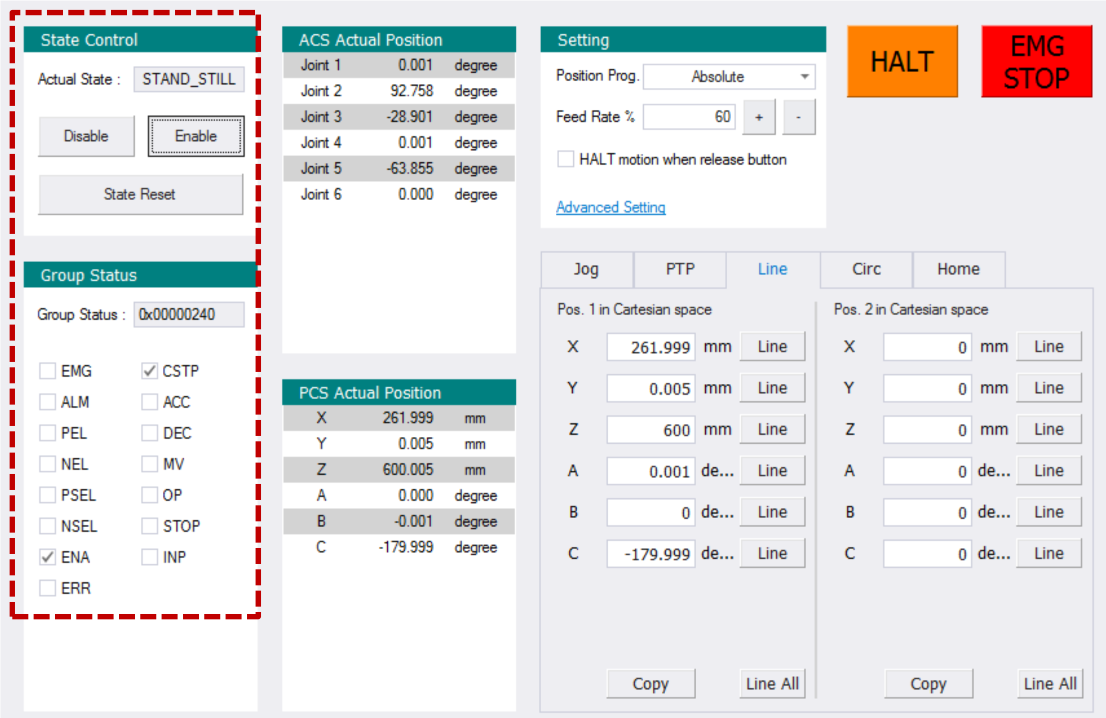
- Move the robotic arm to the required posture using Line. Click on the "Line" tab. Click on "Copy" to copy the current PCS actual position into the fields. Enter 600 in the Z field and keep the rest unchanged. Then click on "Line All" to execute as shown below.



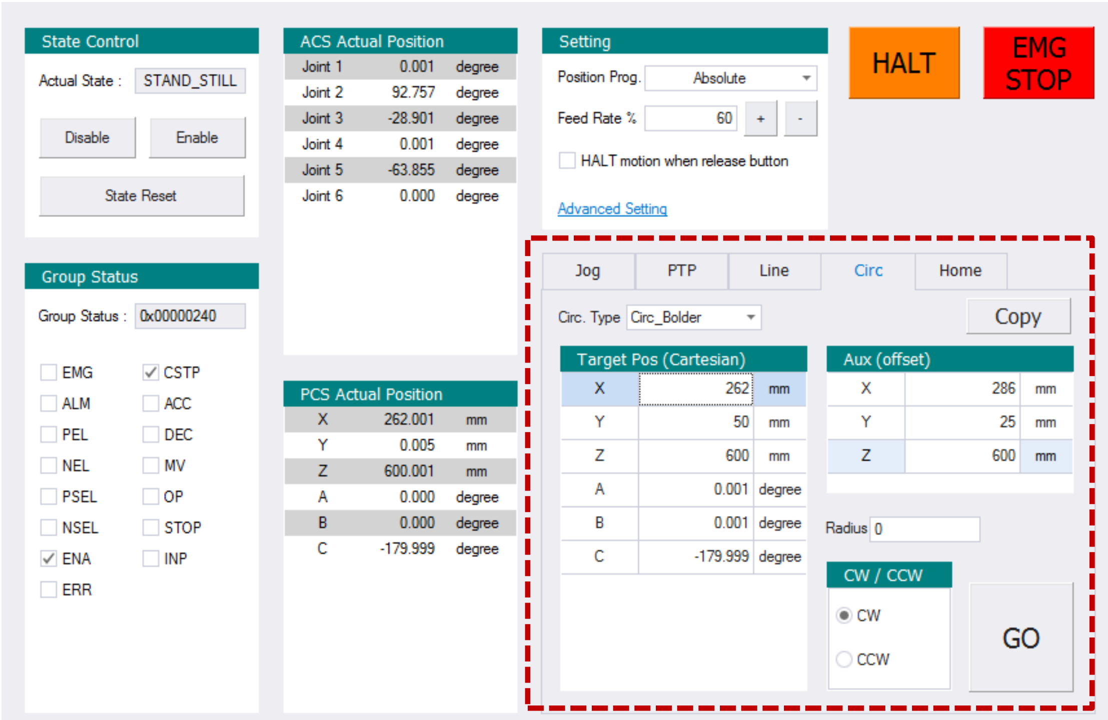
- As the "Line All" is clicked, the actual state turns into MOVING and the ACC, MV and DEC bit of the Group Status into ON. The arm moves to the required position as shown below.



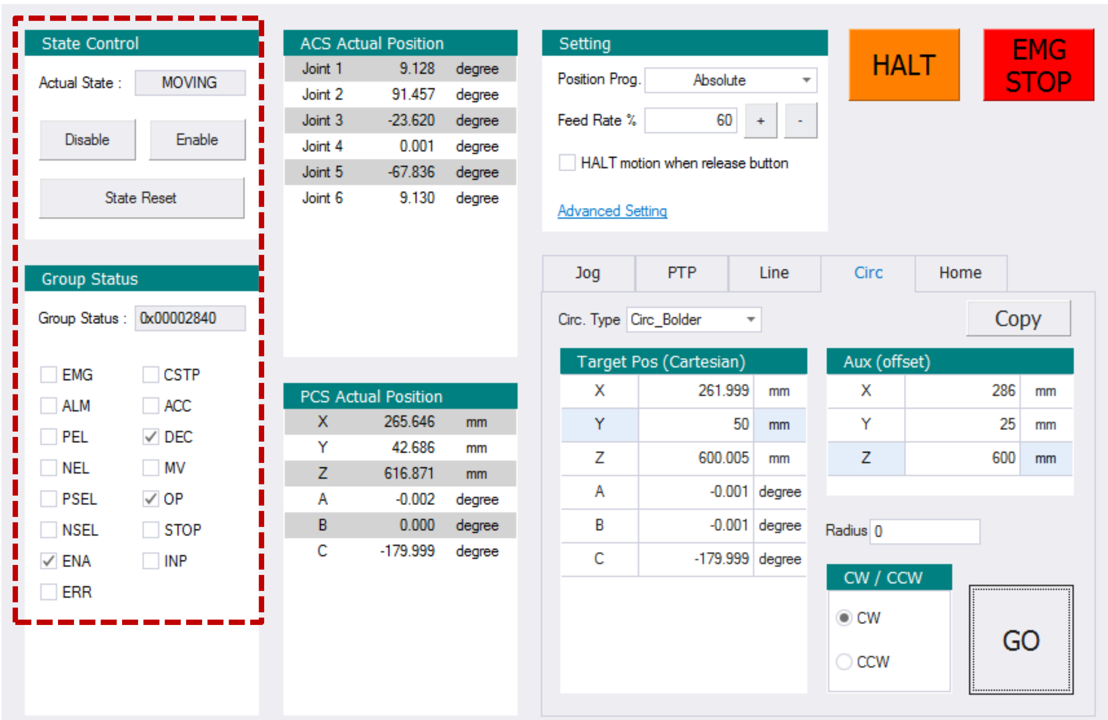
- As the required position is reached, the actual state turns into STAND STILL and the OP bit of Group Status into OFF as shown below.



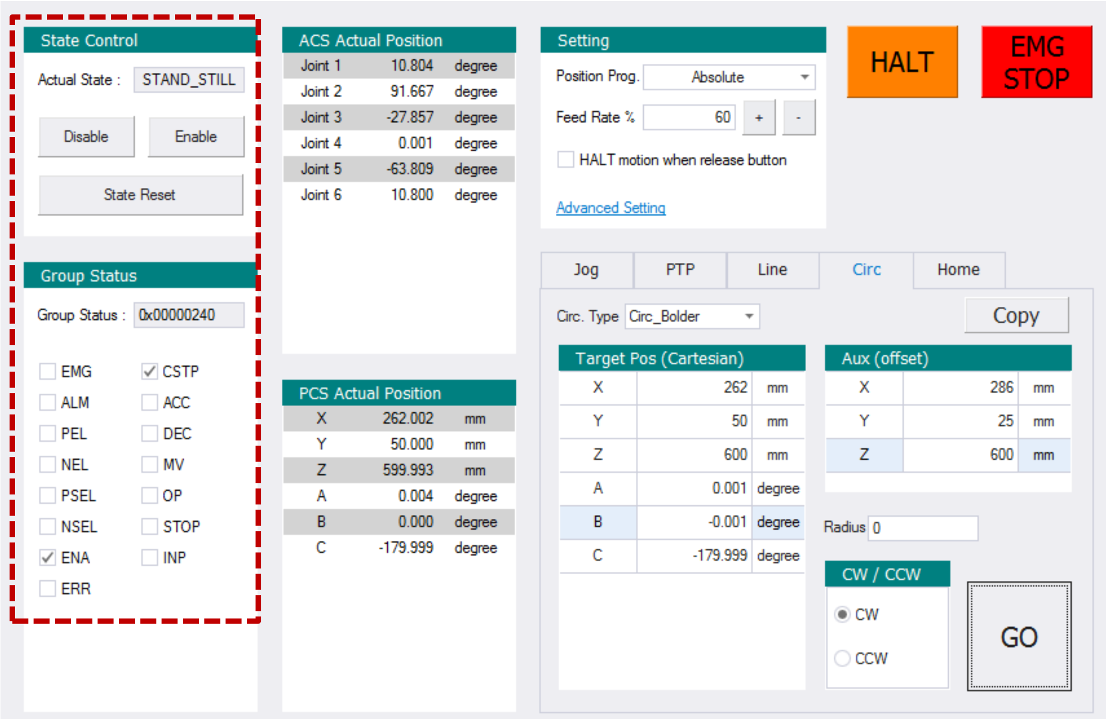
- Move the robotic arm to the required posture using Circle. Click on the "Circ" tab. Click on "Copy" to copy the current PCS actual position into the fields. Enter 262 in the X field of Target Pos, 50 in the Y field, and keep the rest unchanged. Enter 286 in the X field of Aux, 25 in the Y field and keep the rest unchanged. Click on "GO" to execute as shown below.



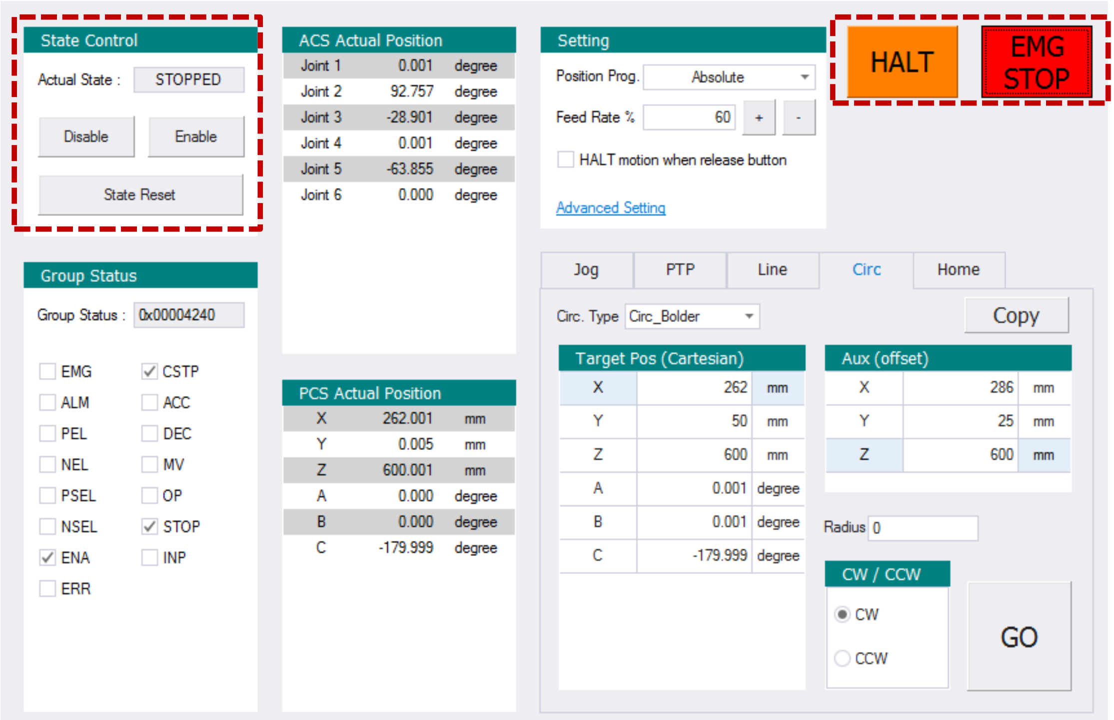
- After clicking, the actual state changes to MOVING and the ACC, MV and DEC bit of Group Status to ON. The arm moves to the required position as shown below.



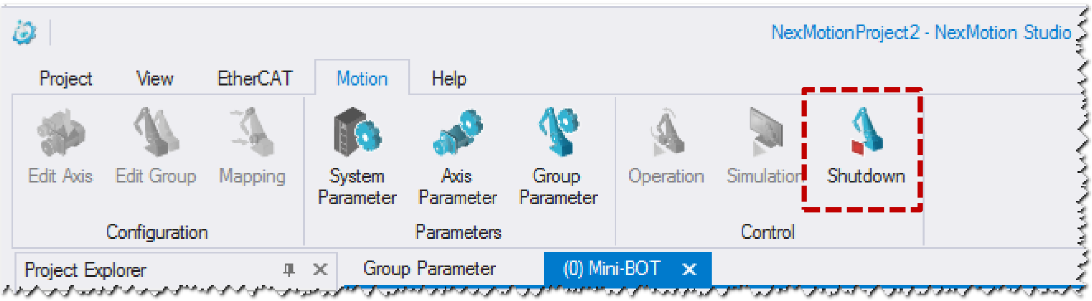
- As the required position is reached, the actual state changes to STAND STILL and the OP bit of Group Status to OFF as shown below.



- There are two ways to stop the ongoing motion: HALT and EMG STOP.
  - HALT: decelerate to 0 using the deceleration of parameter 0x34, and the actual state turns from MOVING → STOPPING → STAND STILL.
  - EMG STOP: decelerate to 0 using the deceleration of parameter 0x21, and the actual state turns from MOVING → STOPPING → STOPPED. At this moment, no more motion command will be executed until the actual state is changed back to STAND STILL, which is done by clicking on “State Reset” as shown below.

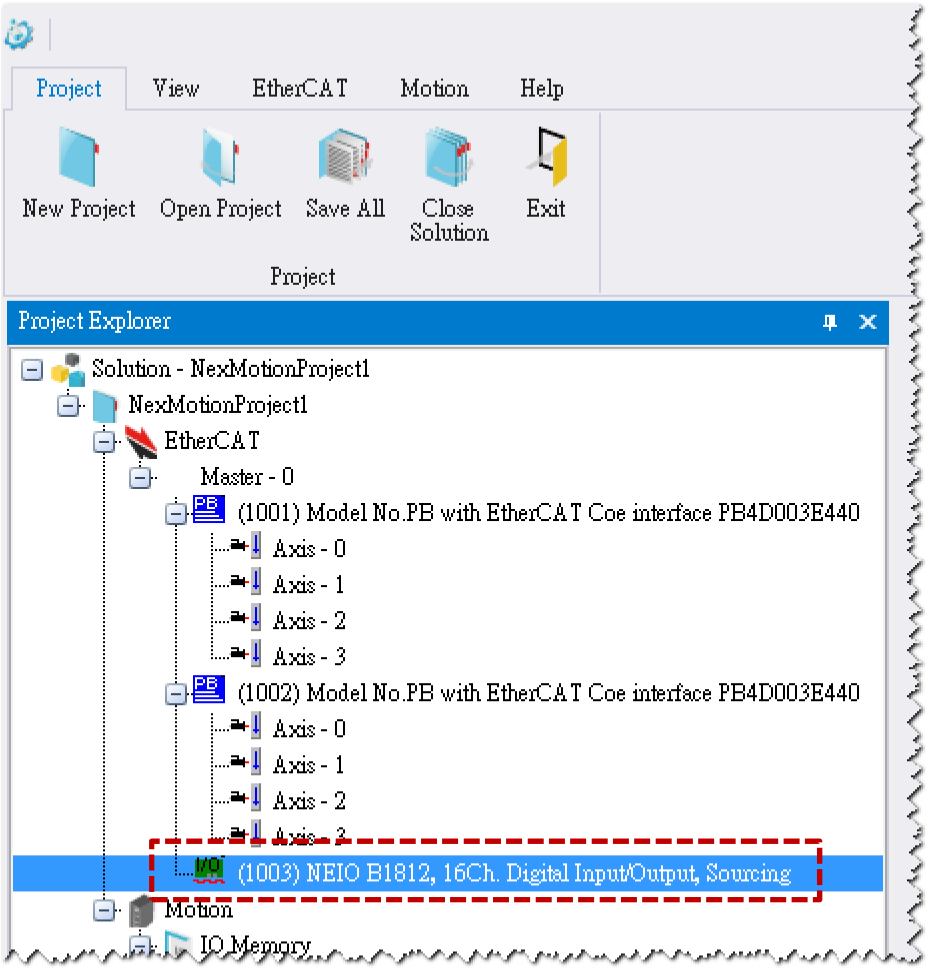


- Click on "Shutdown" in the "Motion" tab to shut the real-time motion control system in the controller down.

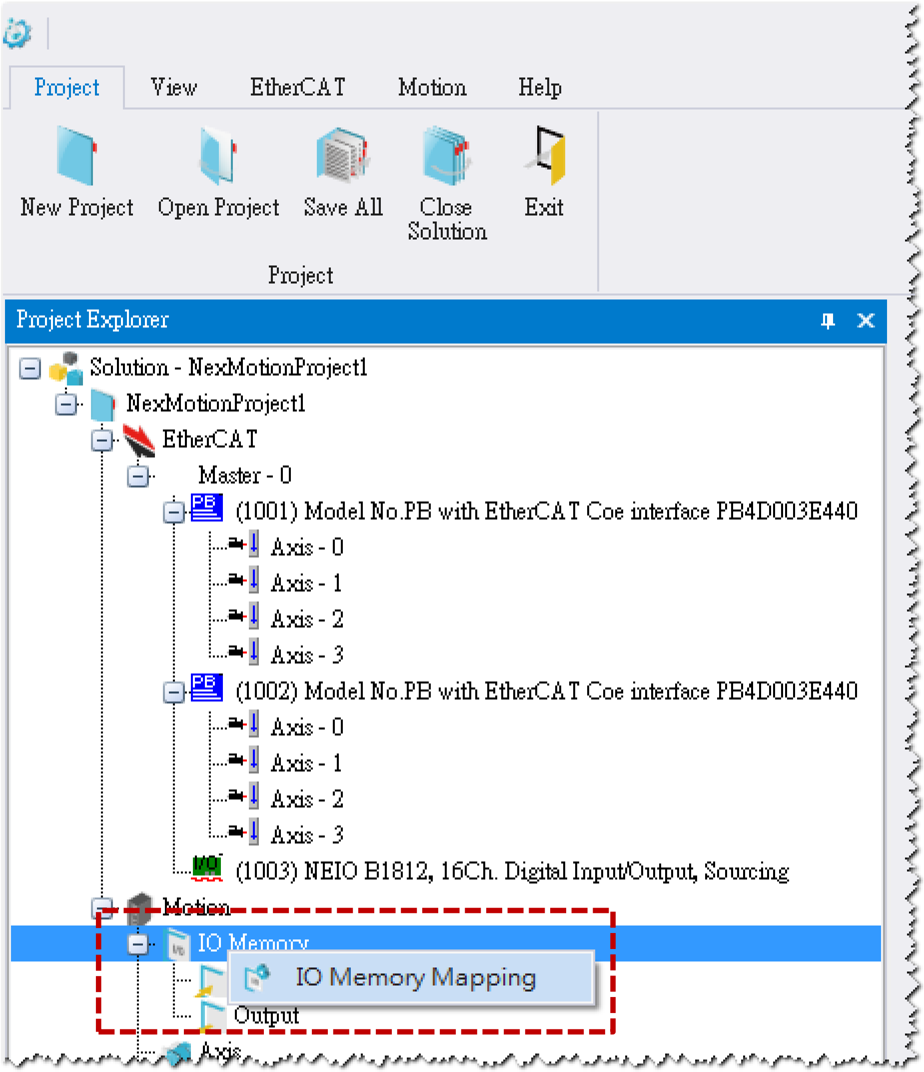


3.1.4. I/O program settings and operations

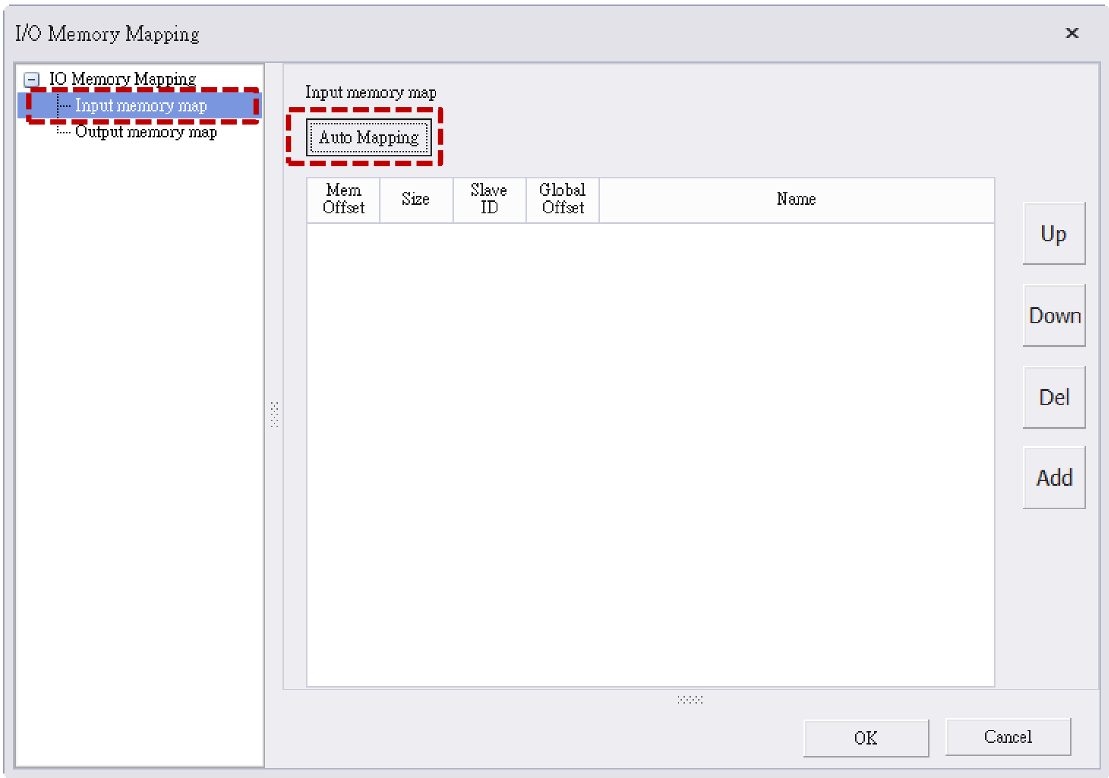
When scanning equipment that is currently connected to EtherCAT, an I/O module called NEIO appears under the two drives as shown below. This section describes how to map the inputs and outputs of this module to the correct motion control I/O memories.



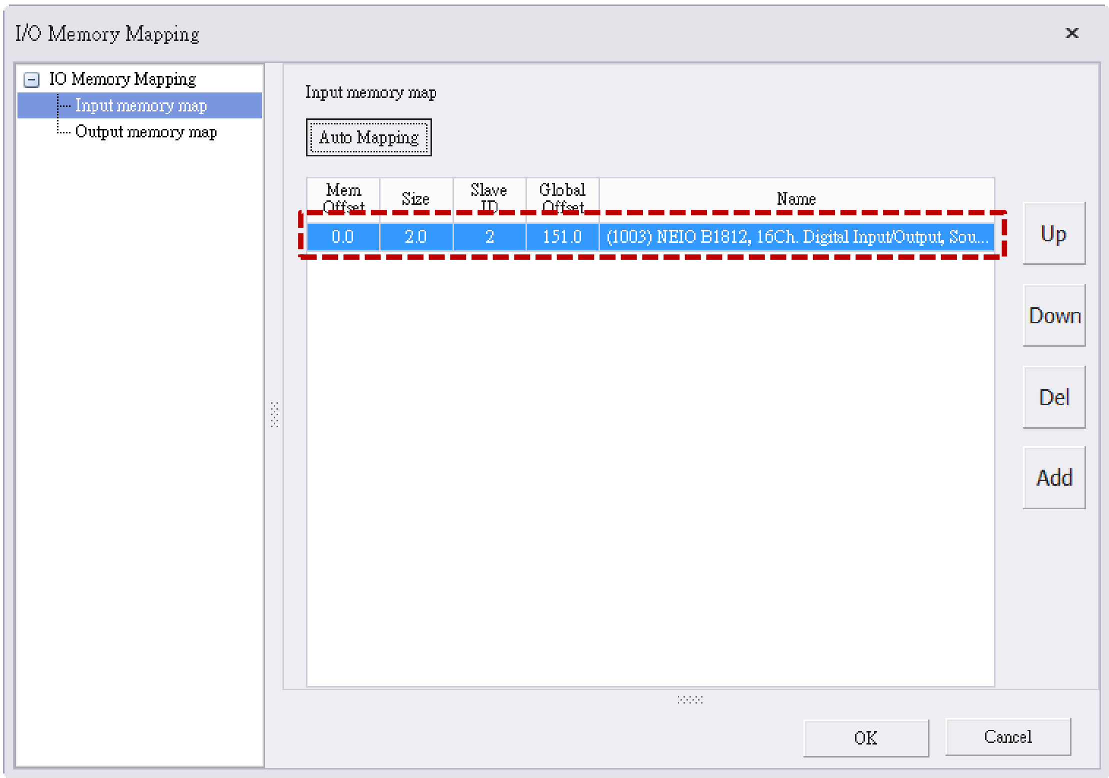
- Right-click the "IO Memory" in the directory tree. Select "IO Memory Mapping" to map the I/O memory positions as shown below.



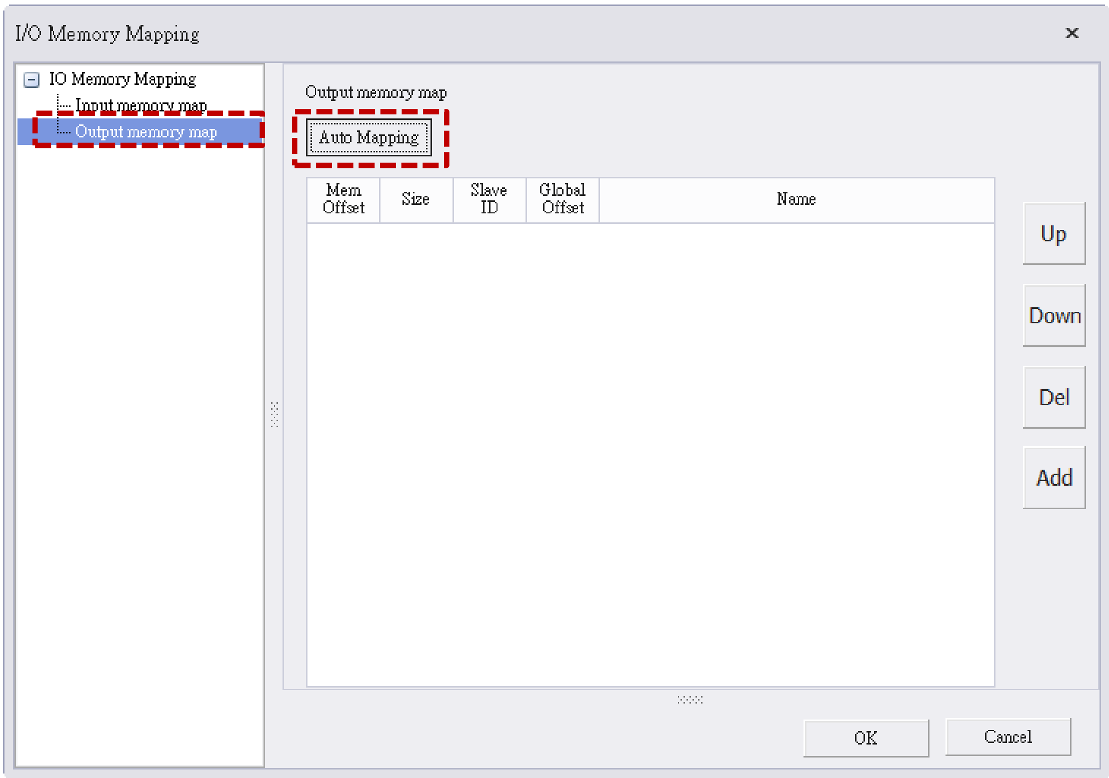
- The IO Memory Mapping dialog box shows as the selection is made. On the upper left corner there is the memory mapping list for input and output. First click on “Input memory map” and then “Auto mapping” as shown below.



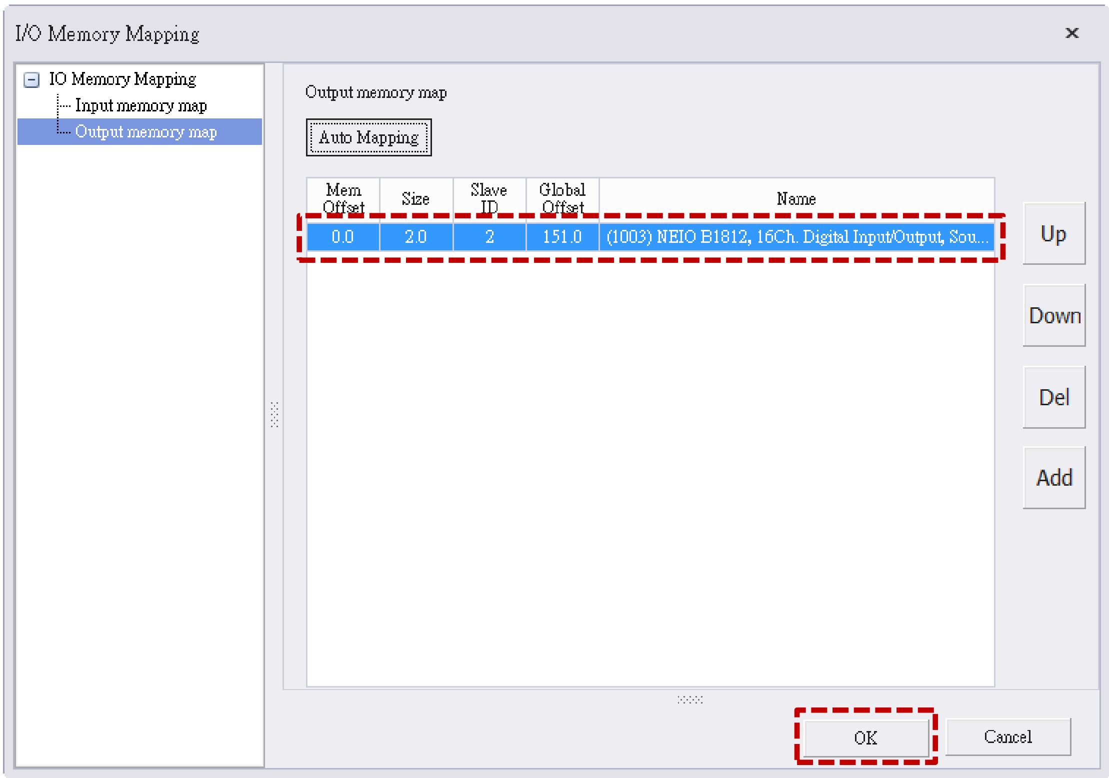
- After clicking, the program automatically maps the memory location and quantity available for inputs to the controller as shown below.



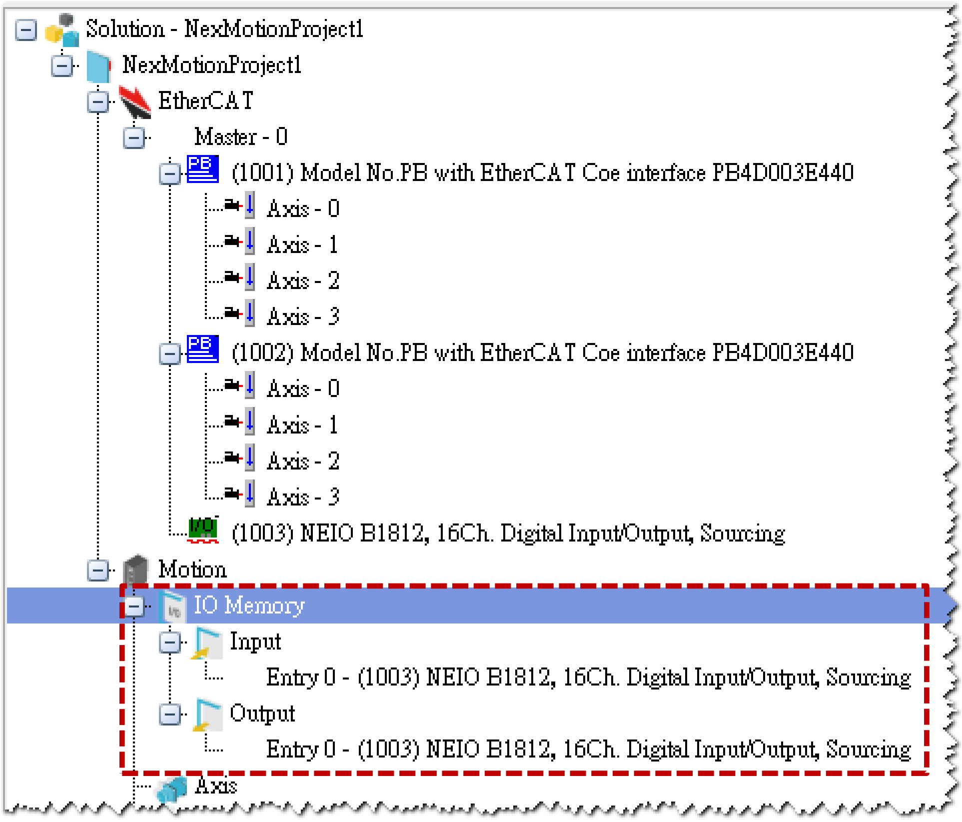
- Then click on “Output memory map” and then “Auto mapping” as shown below.



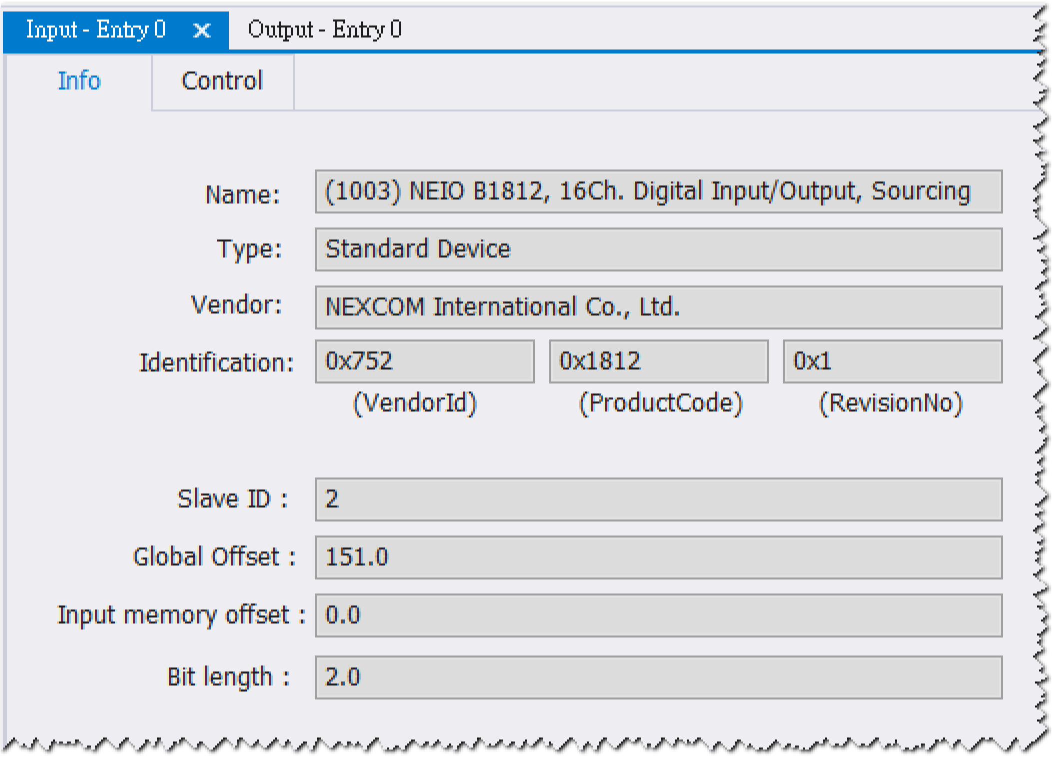
- After clicking, the program automatically maps the memory location and quantity available for outputs to the controller as shown below. Click on “OK” to exit.



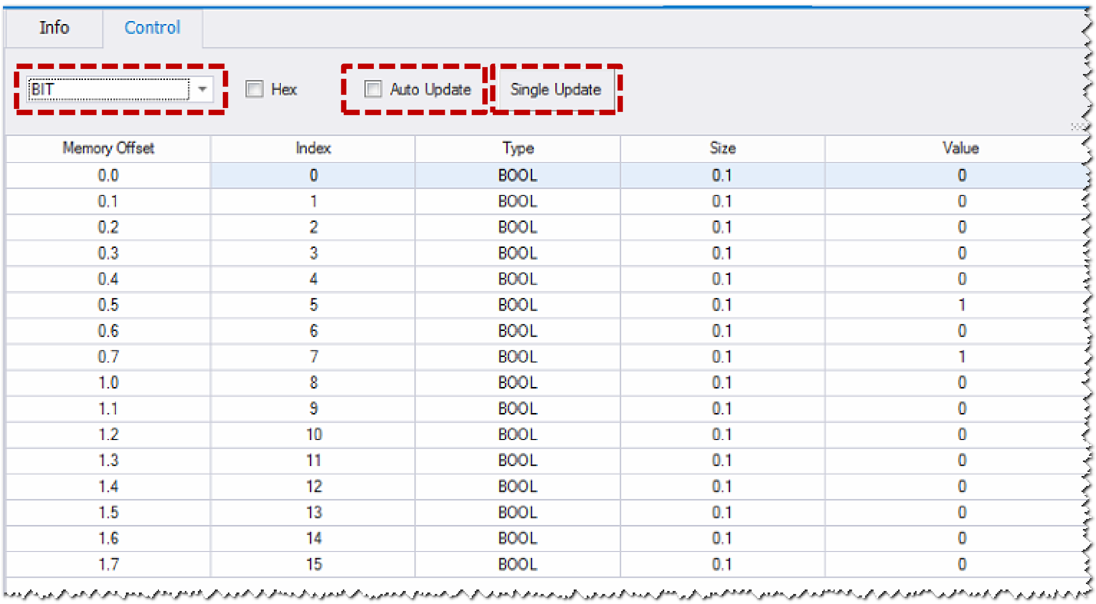
- After clicking OK, the corresponding module numbers are shown under Input and Output in the “IO Memory” of the directory tree, as shown below.



- When the “Operation” is done in the controller, click on “Entry 0 – (1003) NEIO B1812, 16CH…” under the “Input” in the directory tree, and the module information is shown as follows.

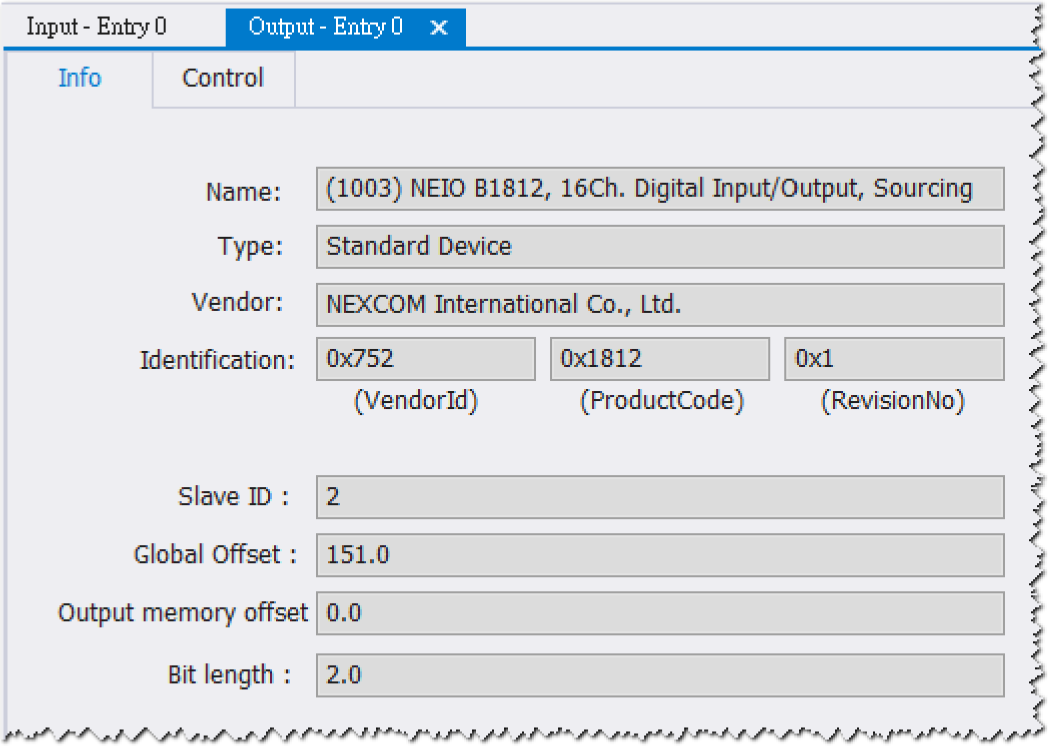


- Click on "Control" tab and the information of the pins at the input end that are triggered. Select "BIT" to bring out the ON/OFF state of every input point as shown below. Check "Auto Update" if automatic updates are required at a fixed interval (200ms) or "Single Update" if only a single update is required.

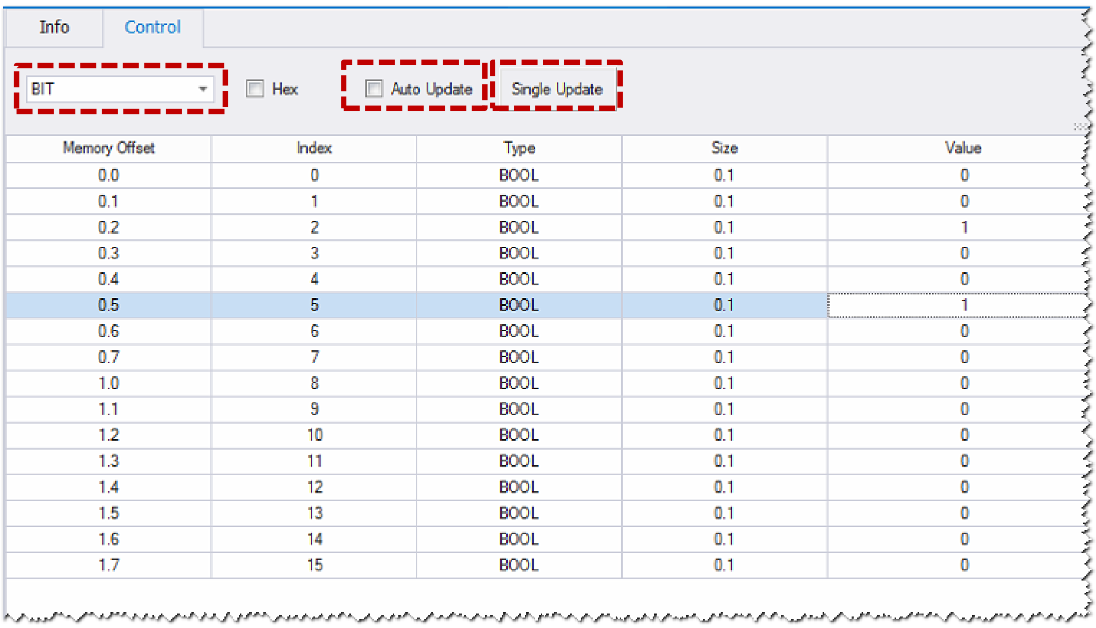


Memory Offset	Index	Type	Size	Value
0.0	0	BOOL	0.1	0
0.1	1	BOOL	0.1	0
0.2	2	BOOL	0.1	0
0.3	3	BOOL	0.1	0
0.4	4	BOOL	0.1	0
0.5	5	BOOL	0.1	1
0.6	6	BOOL	0.1	0
0.7	7	BOOL	0.1	1
1.0	8	BOOL	0.1	0
1.1	9	BOOL	0.1	0
1.2	10	BOOL	0.1	0
1.3	11	BOOL	0.1	0
1.4	12	BOOL	0.1	0
1.5	13	BOOL	0.1	0
1.6	14	BOOL	0.1	0
1.7	15	BOOL	0.1	0

- Click on "Entry 0 – (1003) NEIO B1812, 16CH…" under the "Output" in the directory tree and the module information is shown as follows.



- Click on "Control" tab and the information of the pins at the output end that are triggered. Select the pull-down menu of "PDO" and click on "BIT" to bring out the ON/OFF state of every input point as shown below. Say you wish to turn DO[2] and DO[5] ON, set the value of both Memory Offset 0.2 and 0.5 at 1 as shown below. Check "Auto Update" if automatic updates are required at a fixed interval (200ms) or "Single Update" if only a single update is required.



Memory Offset	Index	Type	Size	Value
0.0	0	BOOL	0.1	0
0.1	1	BOOL	0.1	0
0.2	2	BOOL	0.1	1
0.3	3	BOOL	0.1	0
0.4	4	BOOL	0.1	0
0.5	5	BOOL	0.1	1
0.6	6	BOOL	0.1	0
0.7	7	BOOL	0.1	0
1.0	8	BOOL	0.1	0
1.1	9	BOOL	0.1	0
1.2	10	BOOL	0.1	0
1.3	11	BOOL	0.1	0
1.4	12	BOOL	0.1	0
1.5	13	BOOL	0.1	0
1.6	14	BOOL	0.1	0
1.7	15	BOOL	0.1	0

3.2. Introduction to the miniBOT example program

The MiniBOT example program is an application example developed by using C#. A motion control procedure is programmed for miniBot using the APIs provided in the NexMotion library. The example program works similarly to the NexMotion Studio except that the source code of the example program is provided to allow users to build their own UIs using the APIs provided in the NexMotion library and understand the motion control procedures corresponding to individual APIs.

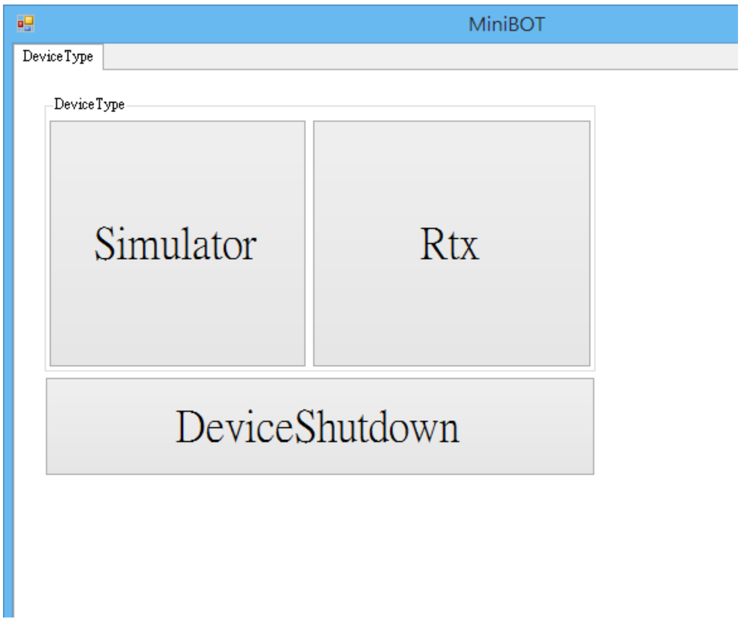
The functions and coding are described in the following section:

3.2.1. Activate/deactivate motion mode

The MiniBOT example program provides two modes:

1. Simulation mode; and
2. Operation mode.

The NCF file generated by NexMotion Studio will be accessed when either mode is activated. The NexMotion motion core is opened according to the parameters defined in the NexMotion Studio.

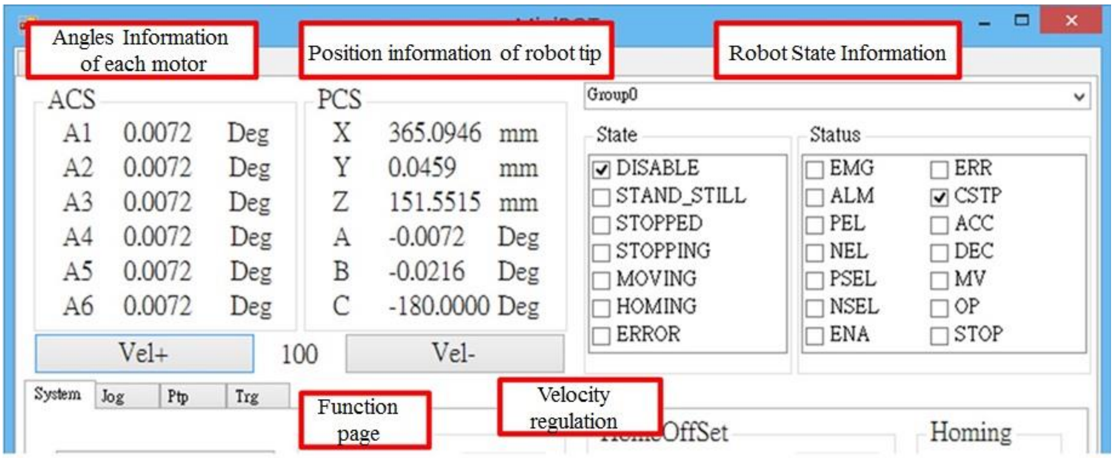


- The Simulation mode: open the NexMotion motion core and simulate the motor operations of MiniBOT. Motion status can be simulated without actually connecting the motors.

- The Operation mode: open the NexMotion motion mode and EtherCAT communications. The hardware has to be connected in order for this mode to work.

- Device Shutdown: this turns both the NexMotion motion core and EtherCAT communications off.

The interface looks like this as the motion mode is activated:



- Motor angle: it shows the angles of each motor simulated or actually obtained.
- Robot tip: the position of robot tip is determined based on the motor angles and the mechanism type defined.
- Robot states: it shows the states and Statuses of a group.
- Velocity regulation: it modulates the motion speed from 10% to 100%.

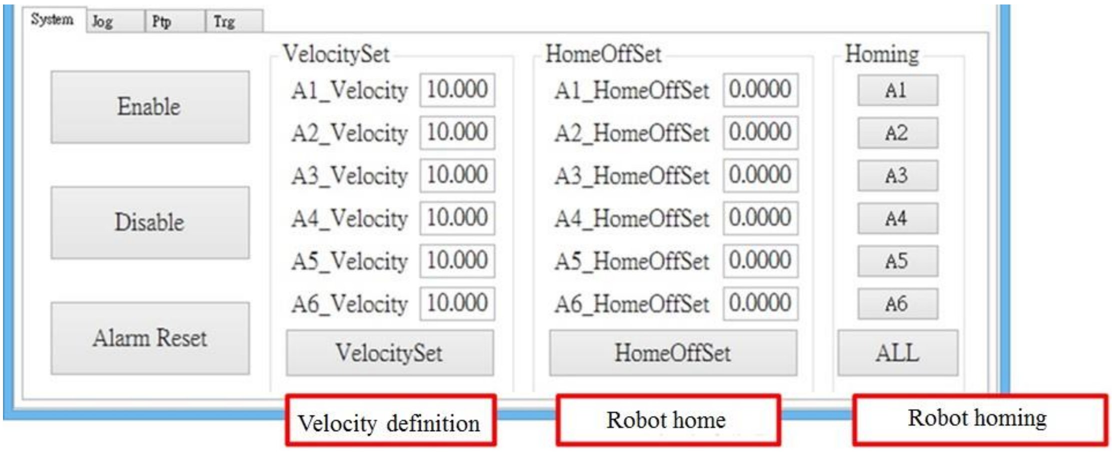
Function page: contains System, Jog, Ptp and Line and allows for switching among them.

3.2.2. System functions

- Function buttons: there are Enable, Disable and Alarm Reset. The Robot states are shown in the information display.
- Velocity definition: it defines the maximum velocity for each axis. The actual operating velocity is configured based the maximum velocity and the percentage of regulation.
- Robot home: the home offset is defined based on the motor angles.

- Robot homing: the homing is performed according to the homing settings in the MiniBOT.

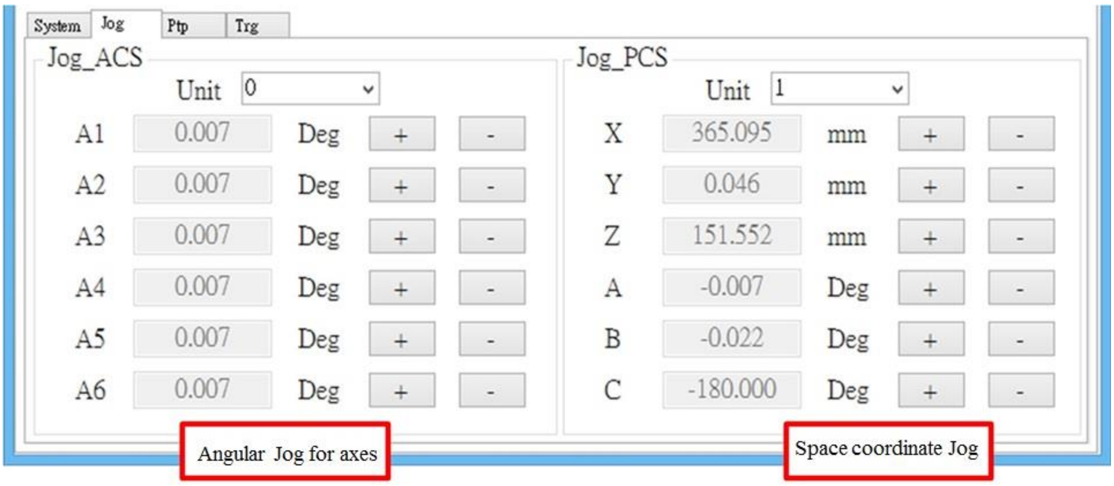
See Homing Proecss for parameter settings and process.



3.2.3. Jog

Push the Jog button to start the jogging motion and release to stop. For jog with distance limit, the distance limit will be recalculated after stopping.

- Angular Jog for axes: this performs the angular jog for the axes. The jog distance depends on Unit which comes with 0, 1, 5 and 10, where 0 means pure Jog motion and the rest means jog motions with distance limit.
- Space coordinate Jog: this perform Jog motion for space coordinates. The space coordinate Jog is a jog motion with distance limit. The distance Unit consists of 1, 5 and 10.

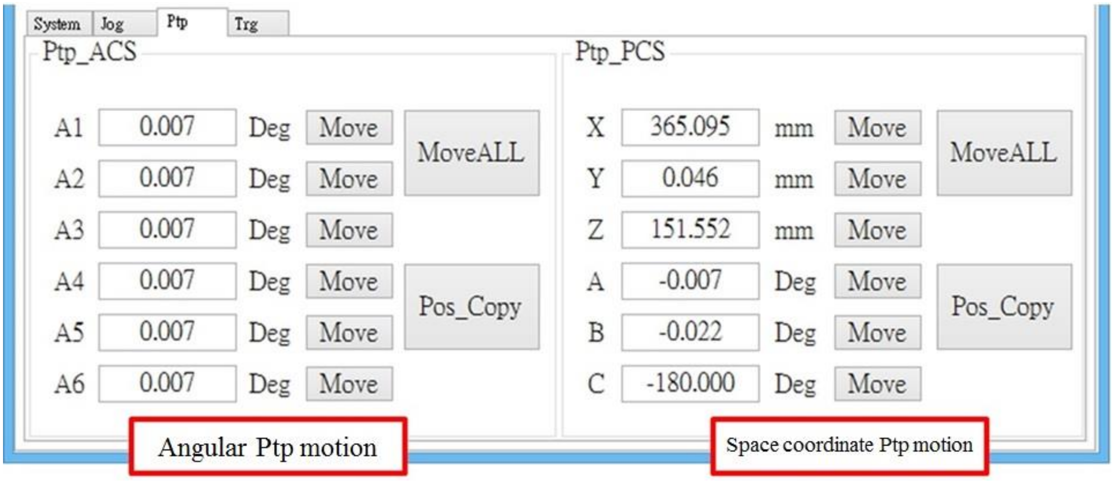


3.2.4. Ptp motion

Push the Ptp button to start a Ptp motion and release it to stop. The Ptp motion is performed according to the commands for the tip of each axis without considering the trajectory of tip.

The Pos_Copy button is used to replicate the information of current position to the command box.

- Angular Ptp motion: the Ptp motion is performed according to the angle command for the axes. There are two types of motion, single-axis Ptp motion and MoveAll Ptp motion, where velocity is regulated on all axes according to the given command for the MoveAll Ptp motion to reach the target positions all at the same time.
- Space coordinate Ptp motion: the Ptp motion is performed according to the angular commands given for the space coordinates of tip. There are two types of motion, space coordinate parameter Ptp motion and MoveAll space coordinate Ptp motion, where velocity is regulated on all axes according to the given command for the MoveAll Ptp motion to reach the target positions all at the same time.



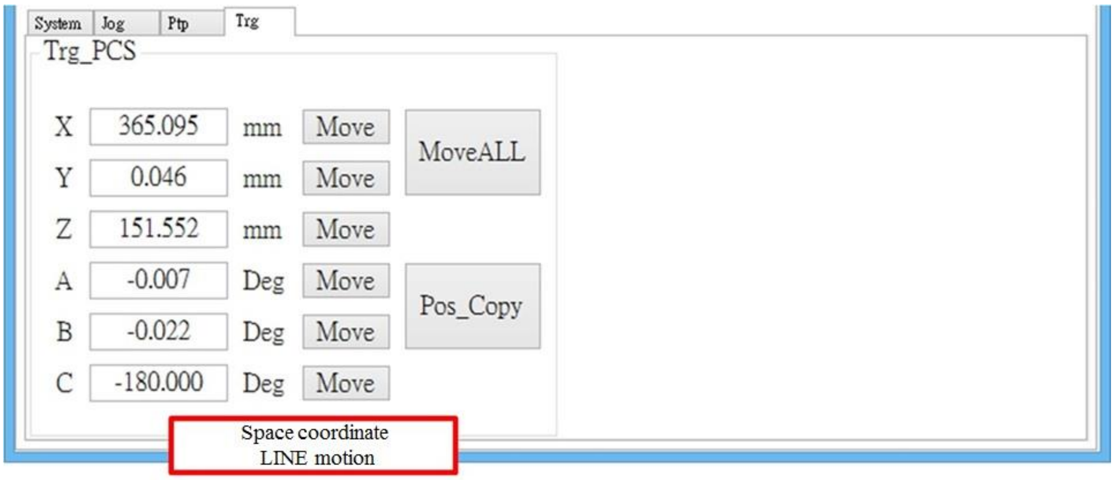
3.2.5. LINE motion

Push the LINE button to start the motion and release to stop.

The LINE motion is performed based on space coordinate commands, and therefore, the tip motion trajectory is linear.

The Pos_Copy button is used to replicate the information of current position to the command box.

- Space coordinate LINE motion: the LINE motion is performed according to space coordinate commands. There are two types of motion; space coordinate parameter LINE motion and MoveAll space coordinate LINE motion.


System
Jog
Ptp
Trg
Trg_PCS
X
365.095
mm
Move
Y
0.046
mm
Move
Z
151.552
mm
Move
A
-0.007
Deg
Move
B
-0.022
Deg
Move
C
-180.000
Deg
Move
MoveALL
Pos_Copy
Space coordinate LINE motion