Machine Automation Controller NJ-series

Startup Guide for Simulink® & Sysmac Studio

SYSMAC-SE20□□

NJ501-□□□□

NJ301-□□□□

R88D-KN□-ECT

GX-AD0471/DA0271

Startup Guide

always in control

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Introduction

The *NJ-series Startup Guide for Simulink® and Sysmac Studio* (hereinafter, may be referred to as "this Guide") describes the startup procedures that are required to use a combination of Simulink® from The MathWorks® Inc. and NJ-series CPU Unit for the first time and the basic operating instructions for the Sysmac Studio. A simple single-axis positioning example is used for the discussion. You can perform the procedures that are presented in this Guide to quickly gain a basic understanding of the combination of Simulink® and NJ-series CPU Unit.

This Guide does not contain safety information and other details that are required for actual use. Thoroughly read and understand the manuals for all of the devices that are used in this Guide to ensure that the system is used safely. Review the entire contents of these materials, including all safety precautions, precautions for safe use, and precautions for correct use.

Intended Audience

This guide is intended for the following personnel.

- Personnel in charge of introducing FA systems
- Personnel in charge of designing FA systems

The personnel must also have the following knowledge.

- Knowledge of electrical systems (an electrical engineer or the equivalent)
- Knowledge of MATLAB®/Simulink® from The MathWorks® Inc.
- Knowledge of NJ-series CPU Units
- Knowledge of operation procedure of Sysmac Studio

Applicable Products

This guide covers the following products.

- CPU Units of NJ-series Machine Automation Controllers
- Sysmac Studio Automation Software
- MATLAB®/Simulink® from The MathWorks® Inc.
- Simulink® PLC Coder™ from The MathWorks® Inc.

Special Information

The icons that are used in this Guide are described below.

Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

Additional information to read as required.
This information is provided to increase understanding or make operation easier.

Terms and Conditions Agreement

CPU Units of NJ-series Machine Automation Controllers

Warranties

(a) Exclusive Warranty. Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

(b) Limitations. OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE

PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right. (c) Buyer Remedy. Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer’s application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user’s programming of a programmable Product, or any consequence thereof.

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron’s test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron’s Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron’s representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Sysmac Studio Automation Software

1. WARRANTY

(1) The warranty period for the Software is one year from the date of purchase, unless otherwise specifically agreed.

(2) If the User discovers defect of the Software (substantial non-conformity with the manual), and return it to OMRON within the above warranty period, OMRON will replace the Software without charge by offering media or download from OMRON’s website. And if the User discovers defect of media which is attributable to OMRON and return it to OMRON within the above warranty period, OMRON will replace defective media without charge. If OMRON is unable to replace defective media or correct the Software, the liability of OMRON and the User’s remedy shall be limited to the refund of the license fee paid to OMRON for the Software.

2. LIMITATION OF LIABILITY

(1) THE ABOVE WARRANTY SHALL CONSTITUTE THE USER’S SOLE AND EXCLUSIVE REMEDIES AGAINST OMRON AND THERE ARE NO OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTY OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE. IN NO EVENT, OMRON WILL BE LIABLE FOR ANY LOST PROFITS OR OTHER INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE SOFTWARE.

(2)OMRON SHALL HAVE NO LIABILITY FOR DEFECT OF THE SOFTWARE BASED ON MODIFICATION OR ALTERNATION TO THE SOFTWARE BY THE USER OR ANY THIRD PARTY.

(3)OMRON SHALL HAVE NO LIABILITY FOR SOFTWARE DEVELOPED BY THE USER OR ANY THIRD PARTY BASED ON THE SOFTWARE OR ANY CONSEQUENCE THEREOF.

3. APPLICABLE CONDITIONS

USER SHALL NOT USE THE SOFTWARE FOR THE PURPOSE THAT IS NOT PROVIDED IN THE ATTACHED USER MANUAL.

4. CHANGE IN SPECIFICATION

The software specifications and accessories may be changed at any time based on improvements and other reasons.

5. ERRORS AND OMISSIONS

The information in this manual has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

Precautions

- When building a system, check the specifications for all devices and equipment that will make up the system and make sure that the OMRON products are used well within their rated specifications and performances. Safety measures, such as safety circuits, must be implemented in order to minimize the risks in the event of a malfunction.
- Thoroughly read and understand the manuals for all devices and equipment that will make up the system to ensure that the system is used safely. Review the entire contents of these manuals, including all safety precautions, precautions for safe use, and precautions for correct use.
- Confirm all regulations, standards, and restrictions that the system must adhere to.
- Contact The MathWorks® Inc. for the codes that were outputted from Simulink® PLC Coder™.
- Applicability of codes that were outputted from Simulink® PLC Coder™ must be judged by the customer.
- Check the user program for proper execution before you use it for actual operation.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- MATLAB® and Simulink® are registered trademarks of The MathWorks® Inc.
- Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.

Other company names and product names in this Guide are the trademarks or registered trademarks of their respective companies.

Software Licenses and Copyrights

The NJ-series CPU Units and Sysmac Studio incorporate certain third party software. The license and copyright information associated with this software is available at http://www.fa.omron.co.jp/nj_info_e/.

Related Manuals

The following manuals are related to the NJ-series Controllers. Use these manuals for reference.

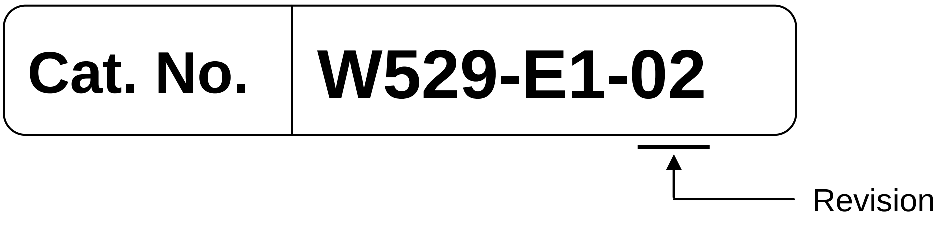
Manual name	Cat. No.	Model numbers	Application	Description
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC-SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	The operating procedures of the Sysmac Studio are described.
NJ-series CPU Unit Hardware User’s Manual	W500	NJ501-□□□□ NJ301-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NJ-series system is provided along with the following information on a Controller built with an NJ501 CPU Unit. ・Features and system configuration ・Introduction ・Part names and functions ・General specifications ・Installation and wiring ・Maintenance and inspection Use this manual together with the NJ-series CPU Unit Software User’s Manual (Cat. No. W501).
NJ-series CPU Unit Software User’s Manual	W501	NJ501-□□□□ NJ301-□□□□	Learning how to program and set up an NJ-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ-series CPU Unit. ・CPU Unit operation ・CPU Unit features ・Initial settings ・Programming based on IEC 61131-3 language specifications Use this manual together with the NJ-series CPU Unit Hardware User’s Manual (Cat. No.W500).
NJ-series CPU Unit Motion Control USER'S MANUAL	W507	NJ501-□□□□ NJ301-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described. Use this manual together with the NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500) and NJ-series CPU Unit Software User’s Manual (Cat. No. W501).

Manual name	Cat. No.	Model numbers	Application	Description
NJ-series Instructions Reference Manual	W502	NJ501-□□□□ NJ301-□□□□	Learning detailed specifications on the basic instructions of an NJ-series CPU Unit.	The instructions in the instruction set (IEC61131-3 specifications) are described. When programming, use this manual together with the NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500) and NJ-series CPU Unit Software User’s Manual (Cat. No. W501).
NJ-series Motion Control Instructions Reference Manual	W508	NJ501-□□□□ NJ301-□□□□	Learning about the specifications of the motion control instructions that are provided by OMRON.	The motion control instructions are described. When programming, use this manual together with the NJ-series CPU Unit Hardware User’s Manual (Cat. No. W500), NJ-series CPU Unit Software User’s Manual (Cat. No. W501) and NJ-series CPU Unit Motion Control User’s Manual (Cat. No. W507).
NJ-series Troubleshooting Manual	W503	NJ501-□□□□ NJ301-□□□□	Learning about the errors that may be detected in an NJ-series Controller.	Concepts on managing errors that may be detected in an NJ-series Controller and information on individual errors are described. Use this manual together with the NJ-series CPU Unit Hardware User’s Manual (Cat. No.W500) and NJ-series CPU Unit Software User’s Manual (Cat. No. W501).
AC Servomotors/Servo Drives (Built-in EtherCAT Communications) User's Manual	I576	R88D-KN□-ECT/ R88M-K	Learning detailed specifications of a G5-series Servo Drive.	This manual explains how to install and wire the G5 Series Servo Drive, set parameters needed to operate the G5 Series Servo Drive, and remedies to be taken and inspection methods to be used in case that problems occur.
AC Servomotors/Servo Drives EtherCAT Communications Linear Motor Type User's Manual	I577	R88D-KN□-ECT-L/R88L-EC	Learning detailed specifications of a G5-series Servo Drive.	This manual explains how to install and wire the G5 Series Servo Drive, set parameters needed to operate the G5 Series Servo Drive, and remedies to be taken and inspection methods to be used in case that problems occur.
EtherCAT Slave Units User's Manual	W488	GX-□□□□□□□	Learning detailed specifications of a GX-series EtherCAT Slave Unit.	This manual contains information you need to know to use the GX-series EtherCAT Slave Unit.

Manual name	Cat. No.	Model numbers	Application	Description
NS-series Programmable Terminals Setup Manual	V072	NS15/NS12/NS10/NS8/NS5	Learning detailed specifications of NS-series.	This manual describes installation and connection procedures, general specifications, and other hardware information for NS-series PTs. Use this manual together with the NS5, NS8, NS10, NS12, NS15 Programmable Terminals Programming Manual (Cat. No. V073).
NS-Series Programmable Terminals Programming Manual	V073	NS15/NS12/NS10/NS8/NS5	Learning about the operation of the PT or the setting methods.	This manual describes using NS-series PT functions and application methods. It also provides troubleshooting methods in the event that problems occur with the PT.
CX-Designer User’s Manual	V099	NS-CXDC1-V3	Learning about the functions and performance of the CX-Designer.	The CX-Designer is software to create screen data for NS-series Programmable Terminals (PTs). This manual describes how to install the CX-Designer and the user interface. It also describes characteristic functions and application methods.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised content
01	June 2013	Original production
02	January 2014	Revisions for adding the SILS (Software In the Loop Simulation) function.

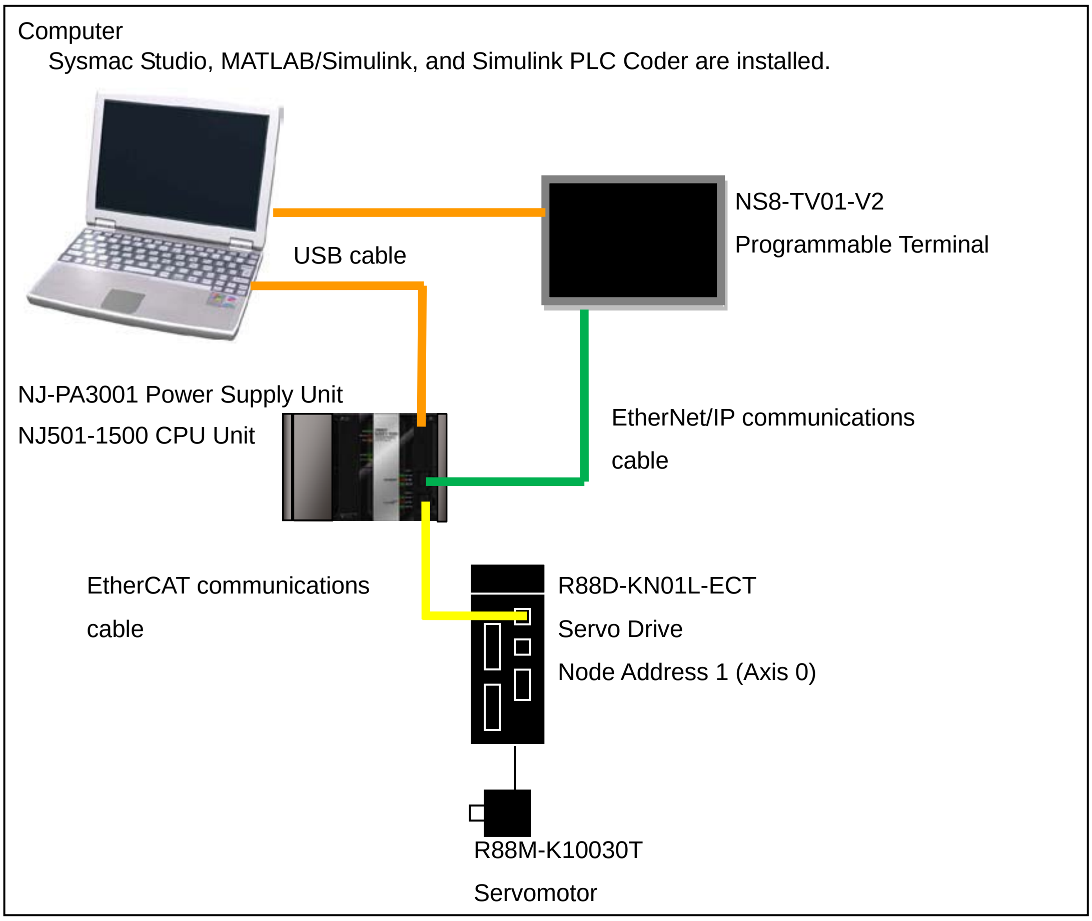
CONTENTS

1. System to Construct and Configuration Devices

1.1. System Configuration and Configuration Devices

This section describes the system configuration and configuration devices used in this Guide.

The following figure represents the system configuration.



The models of the devices that are described in this Guide are given in the following table. When selecting devices for an actual application, refer to the device manuals.

Device name	Model	Manual name
NJ-series CPU Unit	NJ501-1500 (Unit version 1.07)	NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)
NJ-series Power Supply Unit	NJ-PA3001	
EtherCAT communications cables EtherNet/IP communications cables	XS5W-T421-CMD-K	
Programmable Terminal	NS8-TV01-V2	NS-Series Programmable Terminals Programming Manual (Cat. No. V073)
AC Servo Drives	R88D-KN01L-ECT (version 2.10)	AC Servomotors/Servo Drives (Built-in EtherCAT Communications) User's Manual (Cat. No. I576)
AC Servomotors	R88M-K10030L	
Motor Power Cables (for the AC Servo Drives)	R88A-CAKA003S	
Encoder Cables (for the AC Servo Drives)	R88A-CRKA003C	
USB cable	Commercially available USB cable[*1]	---

[*1]. Use a USB2.0 (or 1.1) cable (A connector - B connector), 5.0 m max.

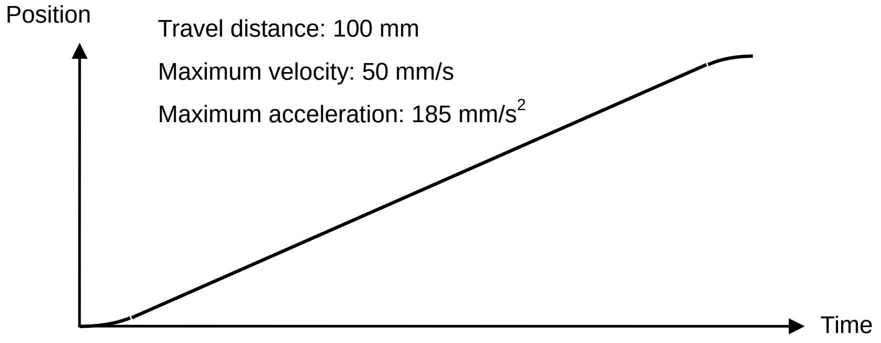
The names and versions of the software that are used in this Guide are given below. Install the following software to a computer (OS: Windows 7 64bit).

Manufacturer	Name	Version
OMRON Corporation	Sysmac Studio	Version 1.09
OMRON Corporation	CX-Designer	Version 3.54
The MathWorks Inc.	MATLAB/Simulink	R2013b
The MathWorks Inc.	Simulink PLC Coder	R2013b

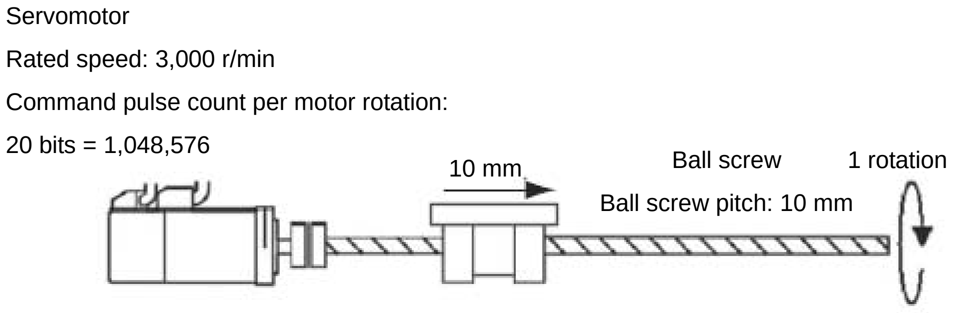
1.2. The Servo System Constructed in this Guide

This guide describes the procedure to start up the system for single-axis positioning with a Servo Drive and Servomotor for one axis. The operations from creating the control algorithm using the Simulink® from the MathWorks® Inc. to operation check using the actual devices are given as the startup procedure.

The single-axis Servo system that is set up in this Guide performs the single-axis positioning operation on the following path.



The mechanical configuration is as shown below.



2. Before You Begin

2.1. Wiring the Devices and Installing the Software

You wire the devices and install the software on the computer as described in *1.1. System Configuration and Configuration Devices*.

Additional Information

Refer to the manuals for the devices that are used in the system for wiring of the devices.

Additional Information

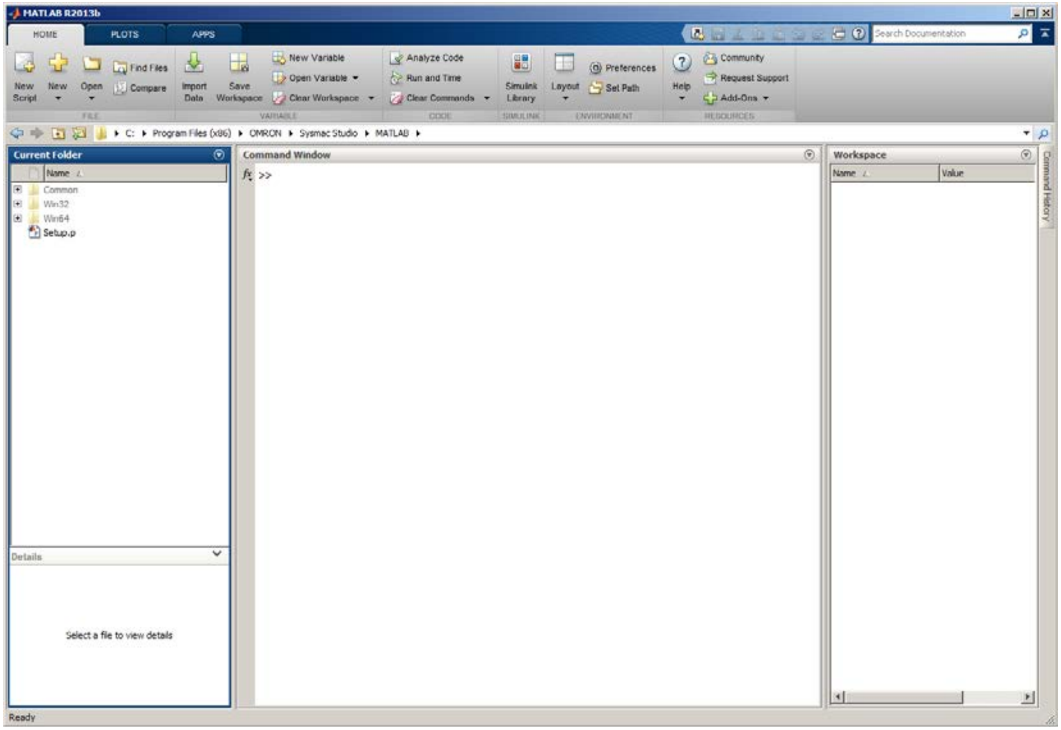
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for installation of the Sysmac Studio.

Additional Information

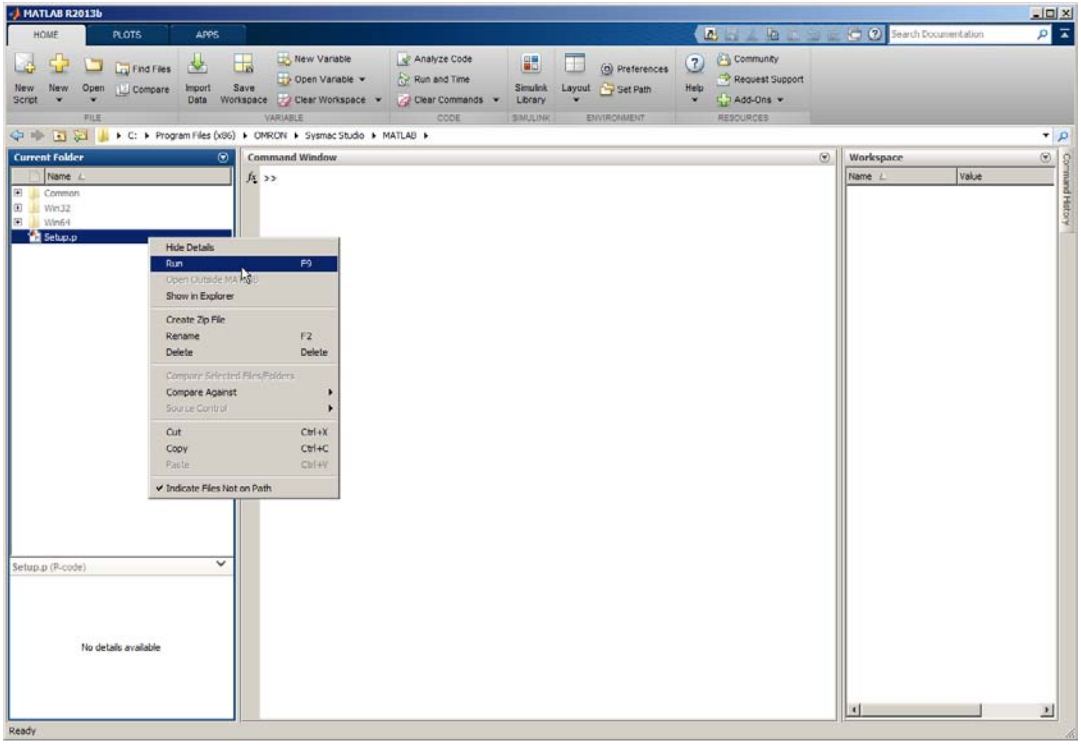
Access the website of The MathWorks Inc. or refer to the *MATLAB & Simulink Installation Guide* that is provided by The MathWorks Inc. for installation of MATLAB/Simulink and Simulink PLC Coder.

You make the MATLAB environment settings for performing the SILS (Software In the Loop Simulation) using Simulink and Sysmac Studio according to the following procedure.

1 Start the MATLAB and select the MATLAB folder in the directory where the Sysmac Studio is installed as the Current Folder. (The default installation folder is *C:\Program Files (x86)\OMRON\Sysmac Studio\MATLAB*.)

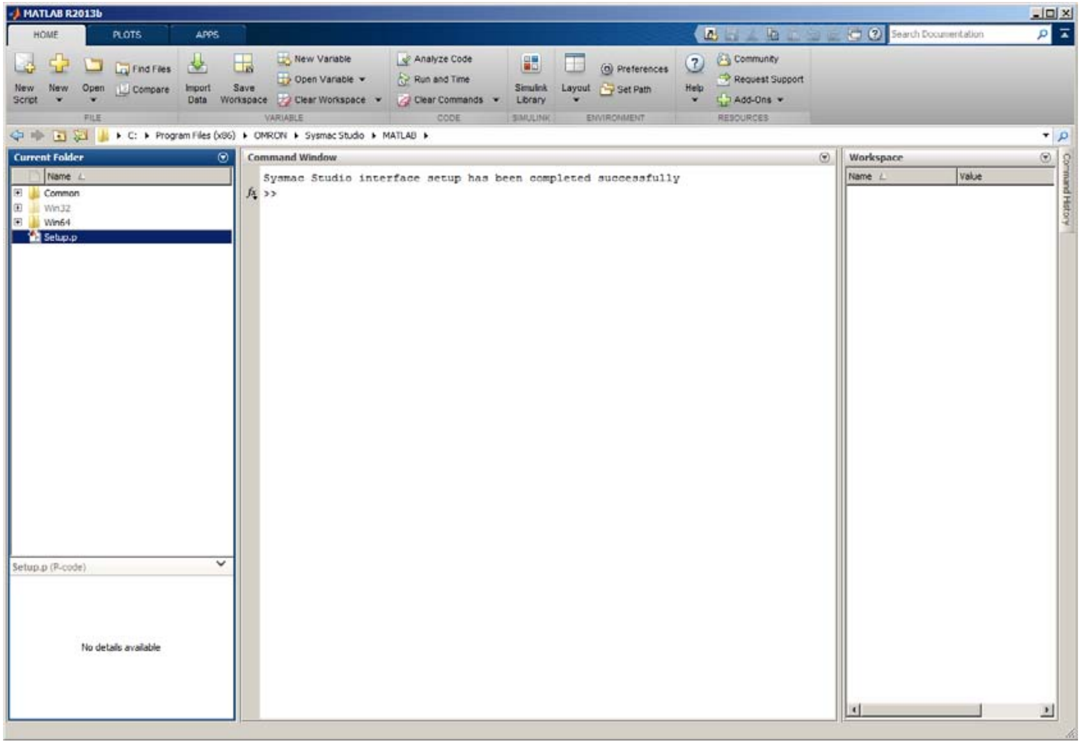


2 Right-click *Setup.p* and select ***Run*** from the menu.



```
fx >>
```

3 Confirm that *Sysmac Studio interface setup has been completed successfully* is displayed in the Command Window.



```
  Sysmac Studio interface setup has been completed successfully
fx >>
```

2.2. Designing the Control Algorithm

You build a model for the Controller and controlled system using the Simulink. The code is created for the Controller by the Simulink PLC Coder. Therefore, you need to build the model using a block supported by the Simulink PLC Coder.

Additional Information

Access the website of The MathWorks Inc. or refer to the *Simulink User Guide* that is provided by The MathWorks Inc. for how to use the Simulink.

Additional Information

Access the website of The MathWorks Inc. or refer to the *Simulink PLC Coder User's Guide* that is provided by The MathWorks Inc. for the blocks supported by the Simulink PLC Coder.

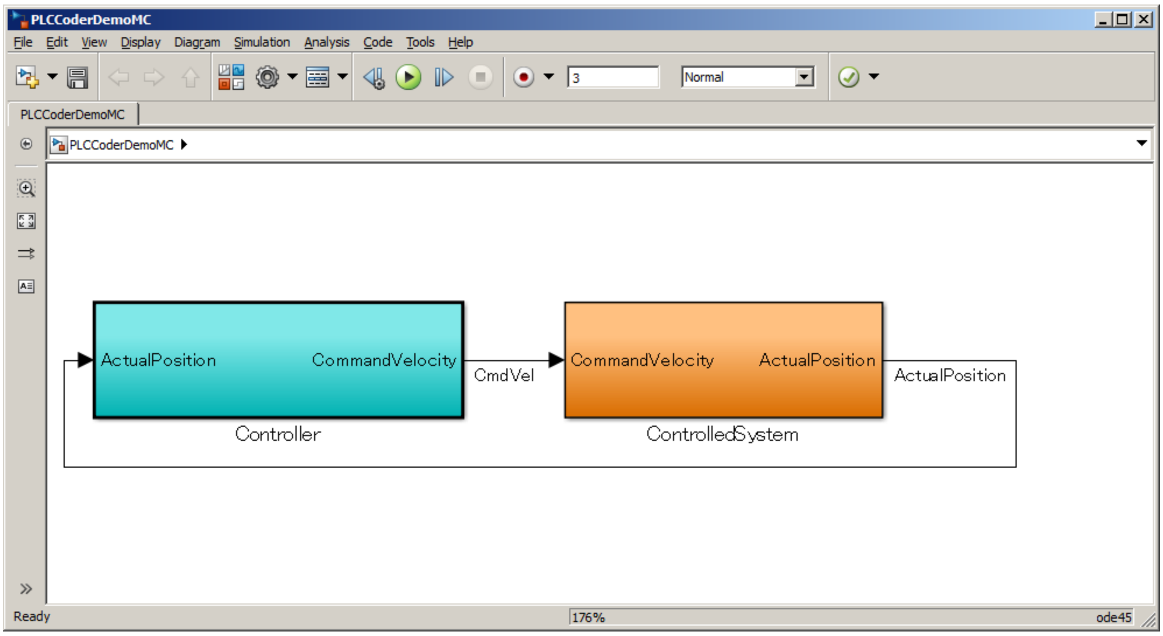
This Guide gives an example for designing the control algorithm so that an NJ-series CPU Unit controls the position and a Servo Drive controls the velocity.

In the *Sample File No. 1 PLCCoderDemoMC.mdl* that is provided separately, a model is created for the Controller (Controller block) and controlled system (ControlledSystem block) by the Simulink as shown in the following figure.

The sampling time of the Controller is set to 1 ms in the sample.

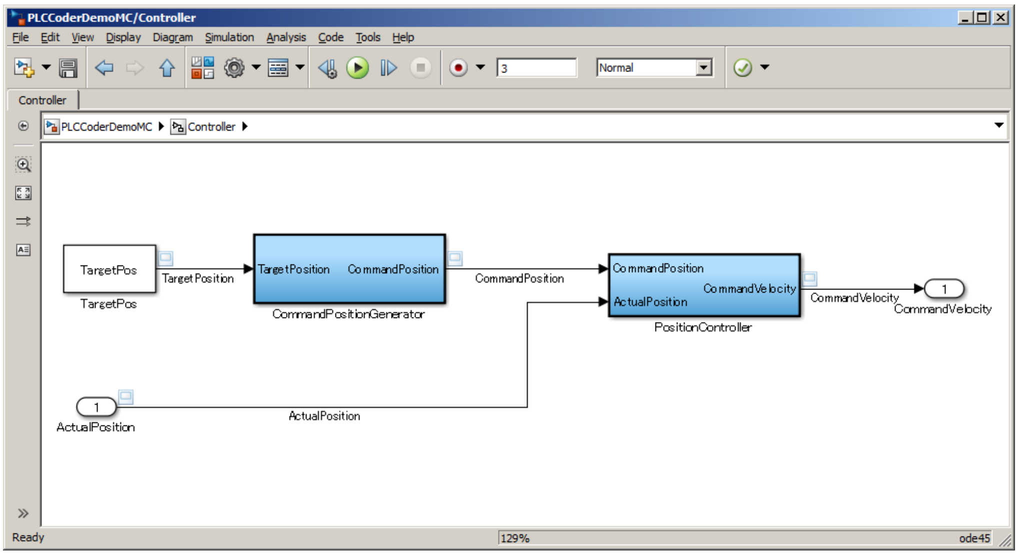
Additional Information

Set the sampling time of the Controller so that it matches the task period of the Sysmac Studio. (Primary periodic task period on the Sysmac Studio: 500 μs, 1 ms, 2 ms, or 4 ms)

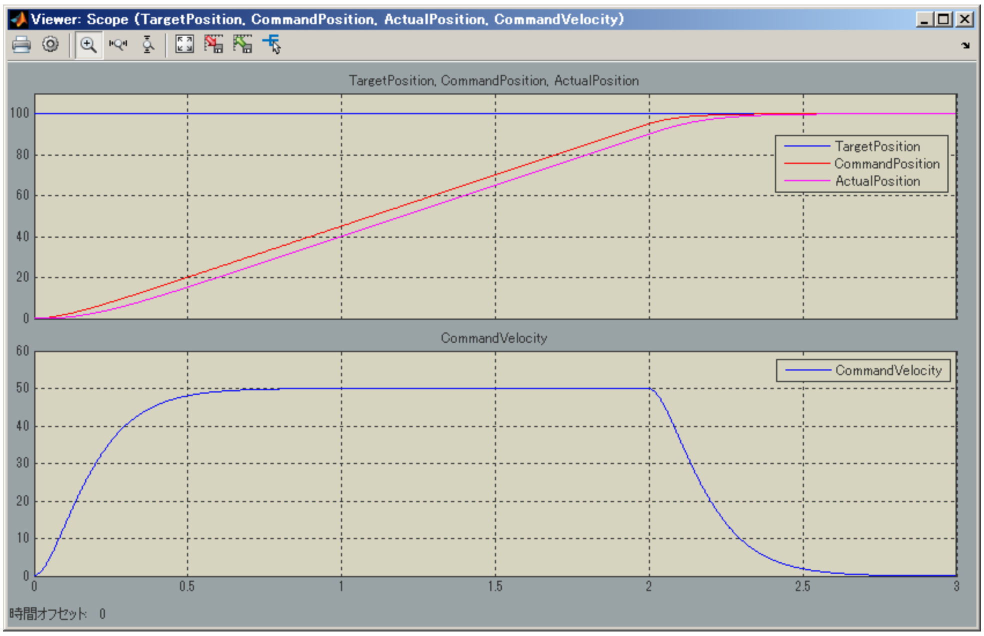


The following figure shows the inside of the Controller block.

The Controller block is composed of two blocks; the CommandPositionGenerator block for creating position command values and the PositionController block for position control.



You will get the simulation execution results as shown below.



3. Setting up the System

3.1. System Setup Procedures

The operation procedure of Simulink and Sysmac Studio is given below.

3.2.1	Outputting the Code using the Simulink PLC Coder	You make a setting for outputting the code for the Sysmac Studio and output the code with test code.
	▼	
3.2.2	Importing the Code into the Sysmac Studio	You import the code outputted by the Simulink PLC Coder into the Sysmac Studio.
	▼	
3.2.3	Checking the Calculation Accuracy	You confirm that the code has the same calculation accuracy as the Simulink (within the acceptable error range) by a simulation.
	▼	
3.2.4	Creating the EtherCAT Network Configuration	You register a R88D-KN01L-ECT Servo Drive that operates as axis 0 on the EtherCAT network configuration.
	▼	
3.2.5	Setting the Axis	You add an axis to control the Servo Drive, assign the Servo Drive to the axis, and make the axis parameter settings.
	▼	
3.2.6	Creating Programs	You create a program for calling the function blocks whose code was outputted by the Simulink PLC Coder and a program for outputting command values to the Servo Drive.
	▼	
3.2.7	Creating the Programming Terminal Screen	You create a Programmable Terminal screen with the CX-Designer.
	▼	
3.2.8	Preparing the SILS (Software In the Loop Simulation)	You add the Sysmac Controller Interface block to the Simulink model and make the setting for data exchange between Simulink and Sysmac Studio.
	▼	
3.2.9	Debugging by Simulation	You debug the programs and screens that you created by the SILS (Software In the Loop Simulation).
	▼	
3.2.10	Transferring the Programs to the CPU Unit	You transfer the programs and parameter settings to the physical CPU Unit.
	▼	

3.2.11	Transferring Screen Data to Programmable Terminal	You transfer the screen data created with the CX-Designer to the physical Programmable Terminal.
	▼	
3.2.12	System Operation Check	You execute the operation according to the programs transferred to the physical CPU Unit and check the operation using the data trace function.

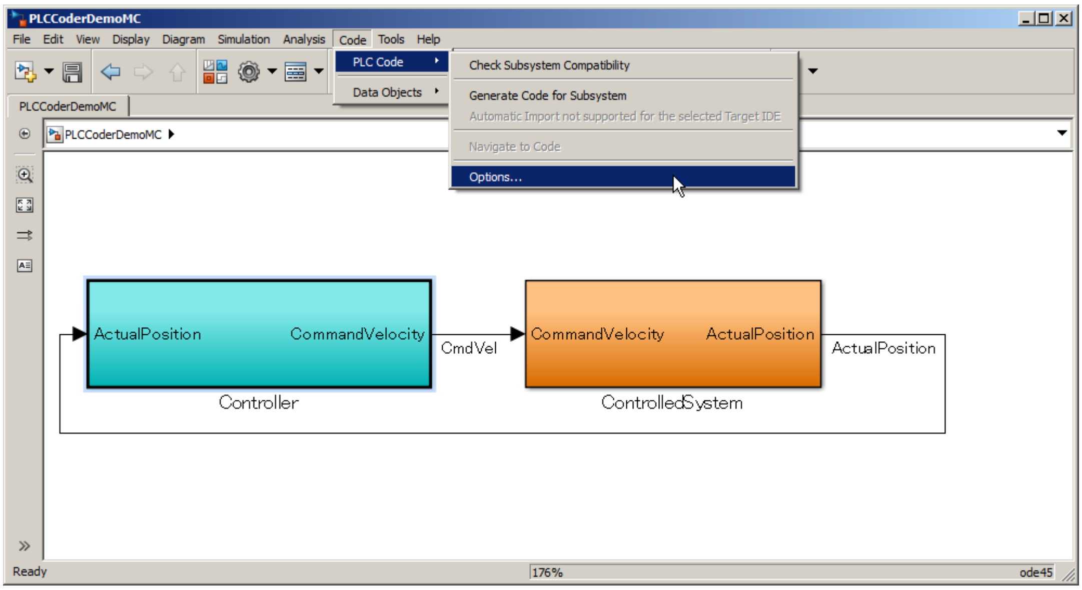
3.2. Simulink PLC Coder & Sysmac Studio Operation Procedure

3.2.1. Outputting the Code using the Simulink PLC Coder

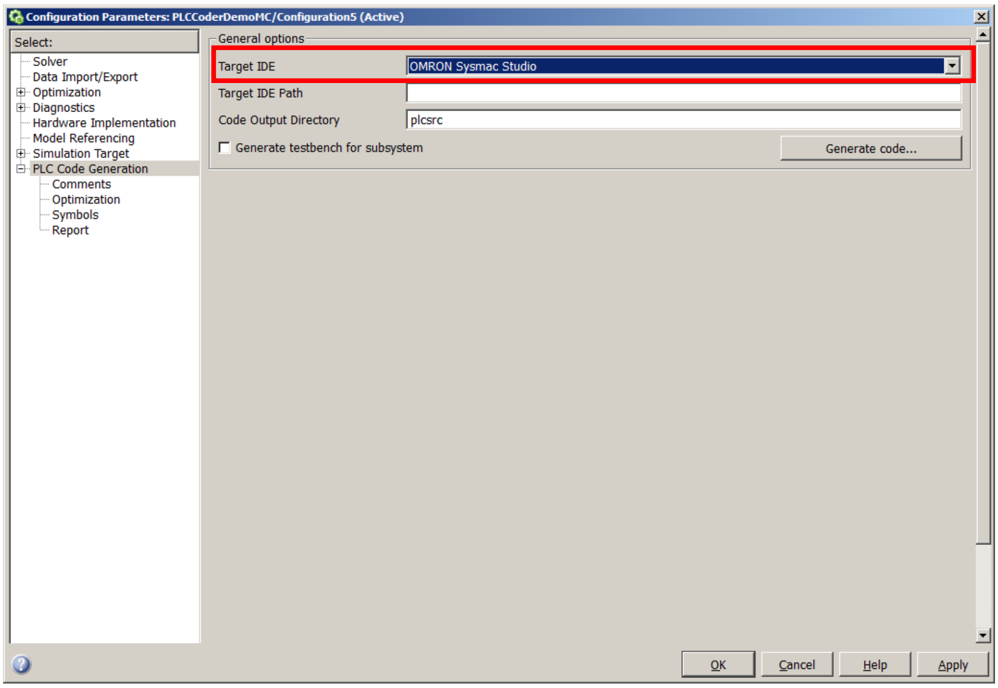
You make a setting for outputting the code for the Sysmac Studio and output the code with test code from the Simulink.

1 Open the *Sample File No. 1 PLCCoderDemoMC.mdl* that is provided separately on the Simulink.

2 Click the Controller block to output the code and select ***PLC Code*** - ***Options*** from the **Code** Menu.



3 Select *PLC Code Generation*, and then select *OMRON Sysmac Studio* for *Target IDE*.

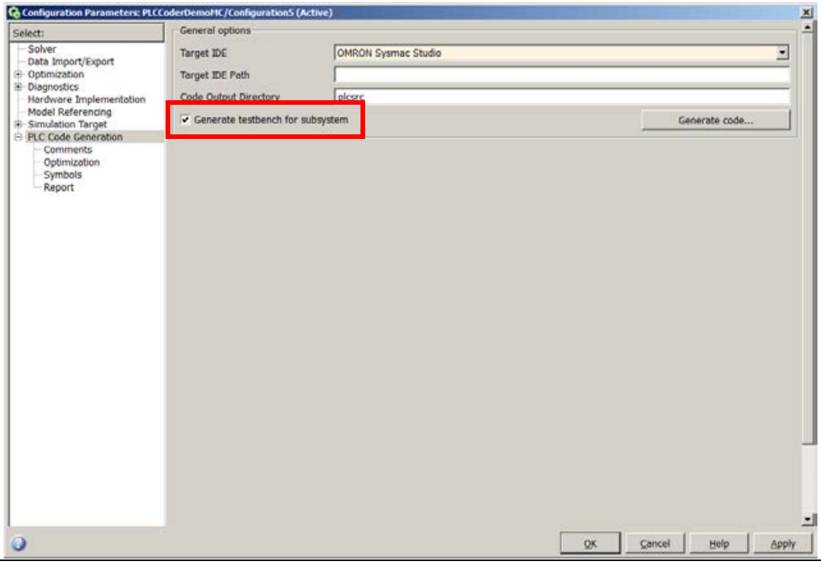
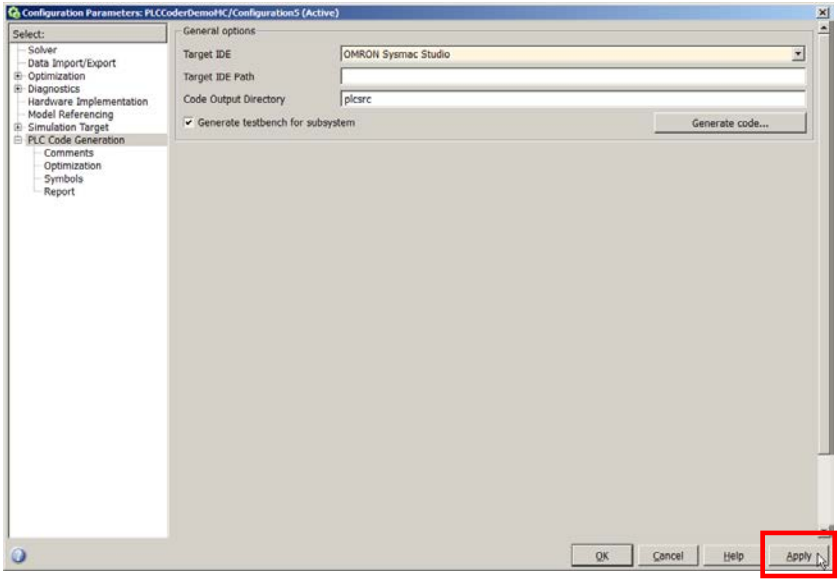
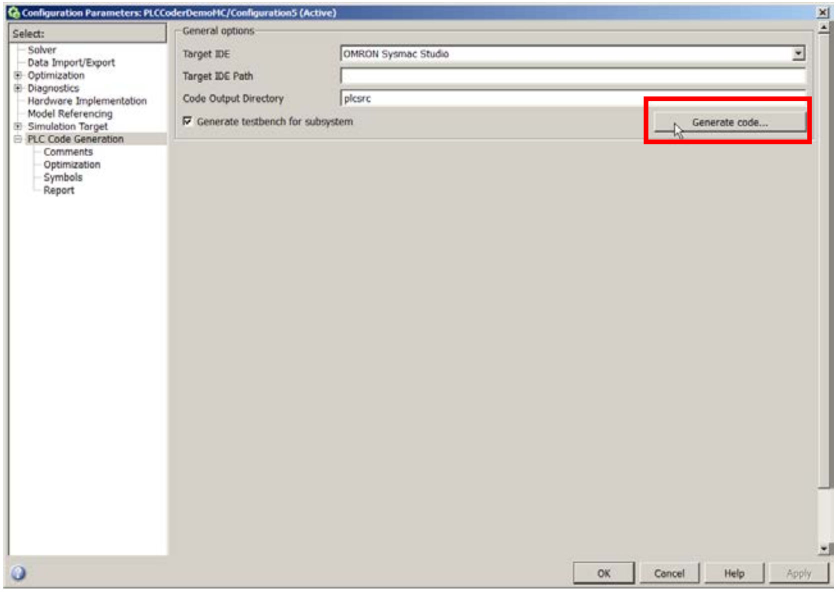


Target IDE: OMRON Sysmac Studio

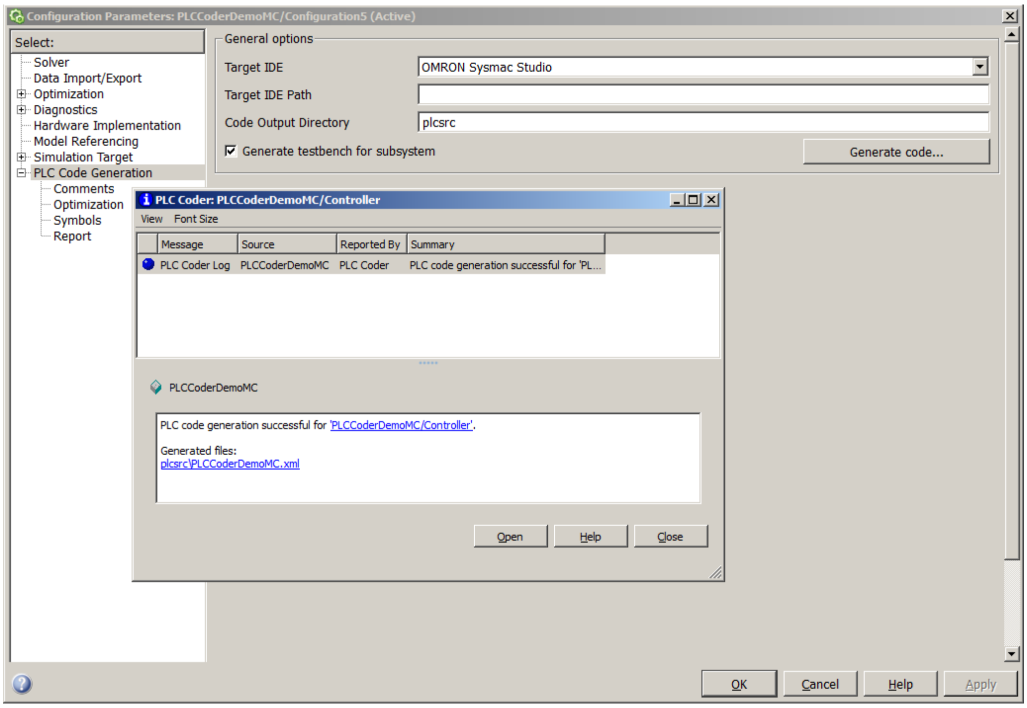
Target IDE Path:

Code Output Directory: plcsrc

☐ Generate testbench for subsystem

4	Select the *Generate testbench for subsystem* check box.  Configuration Parameters: PLCCoderDemoMC/Configuration5 (Active) General options Target IDE: OMRON Sysmac Studio Target IDE Path: Code Output Directory: plcsrc ☑ Generate testbench for subsystem Select: Solver Data Import/Export Optimization Diagnostics Hardware Implementation Model Referencing Simulation Target PLC Code Generation Comments Optimization Symbols Report Generate code... OK Cancel Help Apply
5	Click the **Apply** Button.  Configuration Parameters: PLCCoderDemoMC/Configuration5 (Active) General options Target IDE: OMRON Sysmac Studio Target IDE Path: Code Output Directory: plcsrc ☑ Generate testbench for subsystem Select: Solver Data Import/Export Optimization Diagnostics Hardware Implementation Model Referencing Simulation Target PLC Code Generation Comments Optimization Symbols Report Generate code... OK Cancel Help Apply
6	Click the **Generate Code** Button.  Configuration Parameters: PLCCoderDemoMC/Configuration5 (Active) General options Target IDE: OMRON Sysmac Studio Target IDE Path: Code Output Directory: plcsrc ☑ Generate testbench for subsystem Select: Solver Data Import/Export Optimization Diagnostics Hardware Implementation Model Referencing Simulation Target PLC Code Generation Comments Optimization Symbols Report Generate code... OK Cancel Help Apply

7 The PLCCoderDemoMC.xml file is saved into the *plcsrc* folder specified in *Code Output Directory*.



Configuration Parameters: PLCCoderDemoMC/Configuration5 (Active)

Select:
- Solver
- Data Import/Export
- Optimization
- Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target
- PLC Code Generation
  - Comments
  - Optimization
  - Symbols
  - Report

General options

Target IDE: OMRON Sysmac Studio

Target IDE Path:

Code Output Directory: plcsrc

☑ Generate testbench for subsystem

PLC Coder: PLCCoderDemoMC/Controller

Message	Source	Reported By	Summary
PLC Coder Log	PLCCoderDemoMC	PLC Coder	PLC code generation successful for 'PL...

PLCCoderDemoMC

PLC code generation successful for 'PLCCoderDemoMC/Controller'.

Generated files:
plcsrc\PLCCoderDemoMC.xml

Additional Information

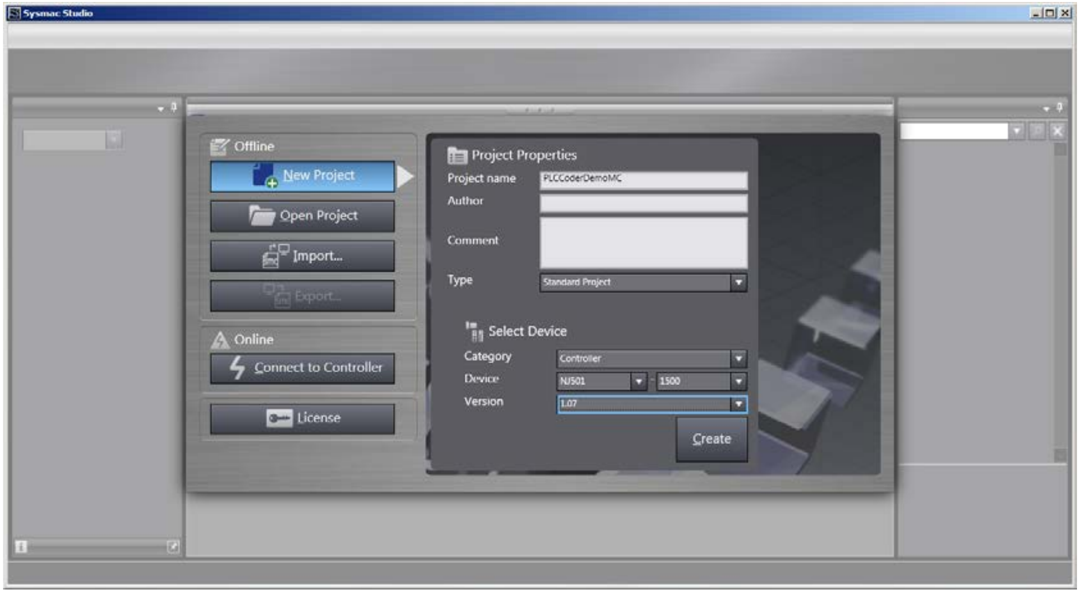
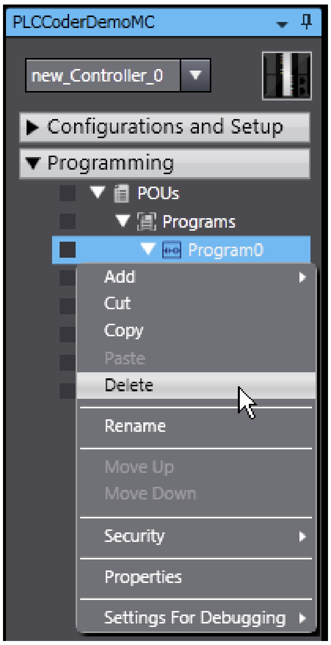
When you adjust the parameters after code generation, you generate the code as a variable, not a constant (literal). Access the website of The MathWorks Inc. or refer to the *Simulink PLC Coder User’s Guide* that is provided by The MathWorks Inc. for the setting procedure.

3.2.2. Importing the Code into the Sysmac Studio

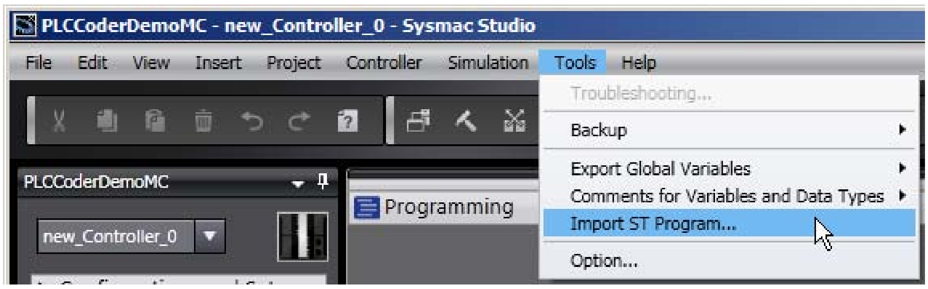
You import the code outputted by the Simulink PLC Coder into the Sysmac Studio.

Additional Information

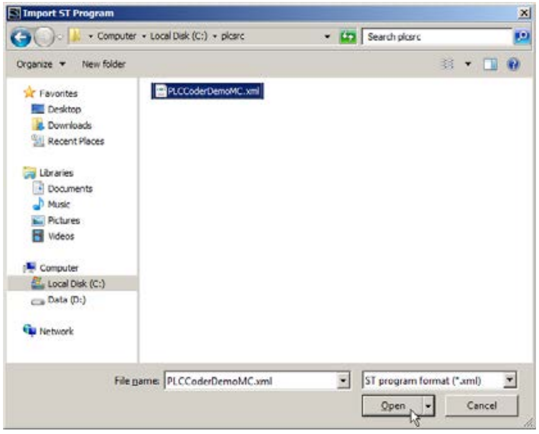
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for how to use the Sysmac Studio.

1	Start the Sysmac Studio and create a new project. Set the Select Device Area as shown below. Category: Controller Device: NJ501-1500 Version: 1.07  Project Properties Project name: PLCCoderDemoMC Author: Comment: Type: Standard Project Select Device Category: Controller Device: NJ501 - 1500 Version: 1.07 Sysmac Studio Offline New Project Open Project Import... Export... Online Connect to Controller License Create
2	Delete the **Program0** that is automatically created when a new project is created because it is not used in this Guide.  PLCCoderDemoMC new_Controller_0 Configurations and Setup Programming POUs Programs Program0 Add Cut Copy Paste Delete Rename Move Up Move Down Security Properties Settings For Debugging

3 Select ***Import ST Program*** from the **Tools** Menu.


PLCCoderDemoMC - new_Controller_0 - Sysmac Studio
File Edit View Insert Project Controller Simulation Tools Help
Troubleshooting...
Backup
Export Global Variables
Comments for Variables and Data Types
Import ST Program...
Option...
PLCCoderDemoMC
new_Controller_0
Programming

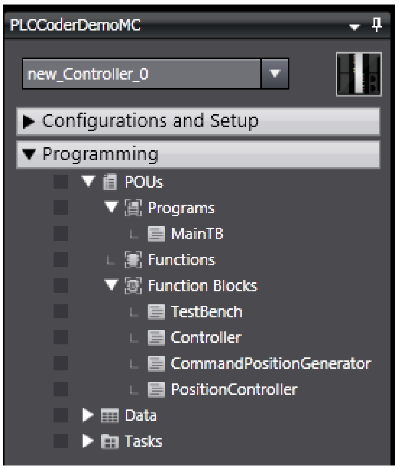
4 Select the PLCCoderDemoMC.xml file that was outputted in the previous section in the Import ST Program Dialog Box.



File name: PLCCoderDemoMC.xml

ST program format (*.xml)

The data is imported and the programs, functions, function blocks, data types, and global variables in the XML file are added to the project of Sysmac Studio.



The **Controller** block whose code is outputted by the Simulink PLC Coder and its internal blocks **CommandPositionGenerator** and **PositionController** are imported as function blocks of Sysmac Studio.

TestBench is a function block for a test to call the Controller function block.

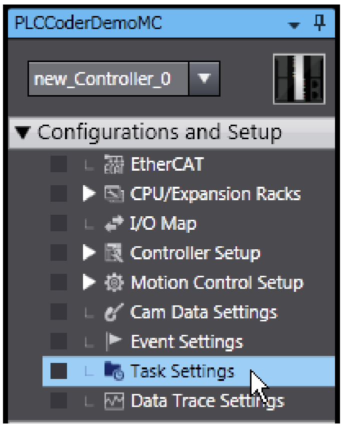
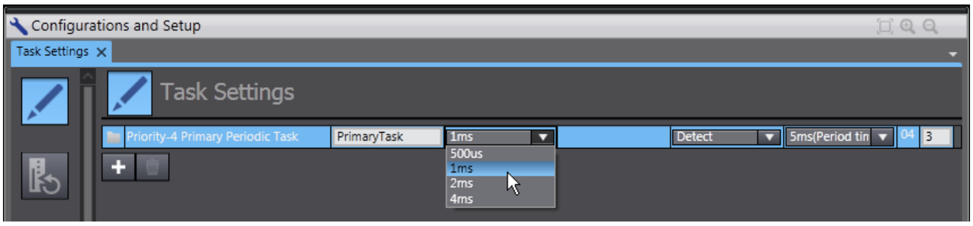
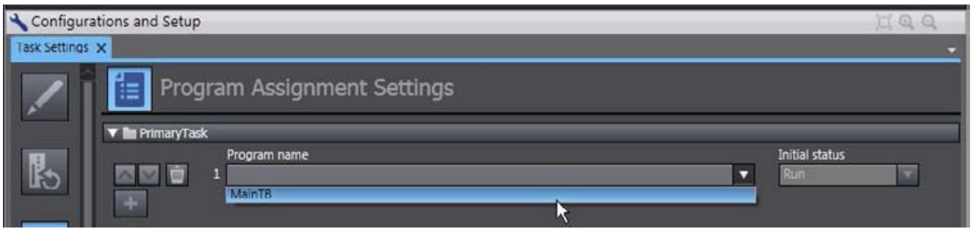
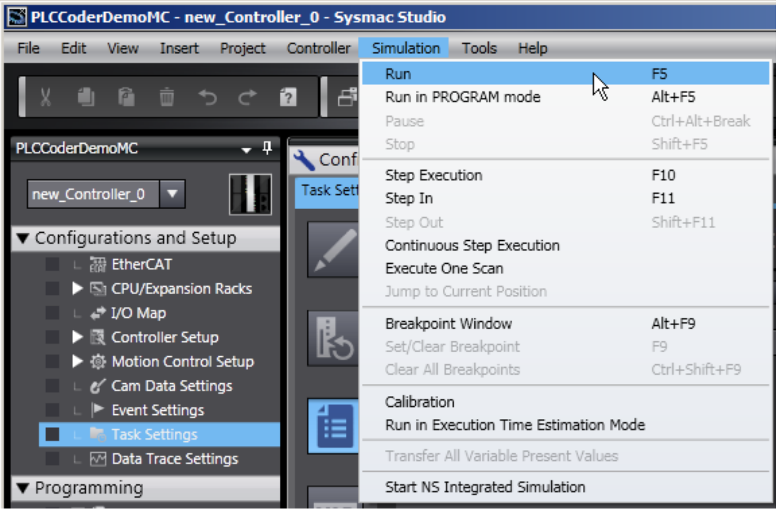
MainTB is a program to call the TestBench function block.

Additional Information

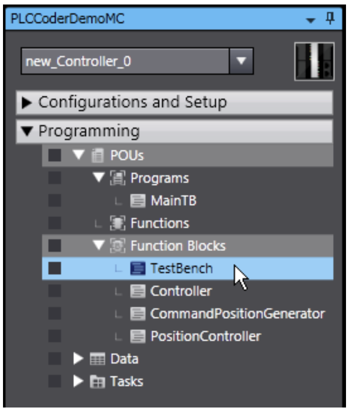
The TestBench function block and the MainTB program are outputted when the *Generate testbench for subsystem* check box is selected in Step 4 of *3.2.1 Outputting the Code using the Simulink PLC Coder*.

3.2.3. Checking the Calculation Accuracy

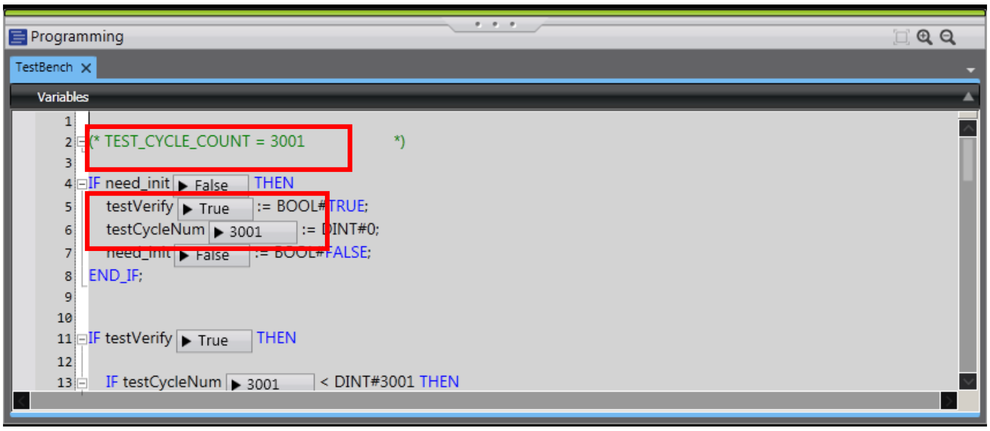
You confirm that the code has the same calculation accuracy as the Simulink (within the acceptable error range) by a simulation.

1	Double-click **Task Settings** in the Multiview Explorer to display the Task Settings Tab Page.  PLCCoderDemoMC new_Controller_0 Configurations and Setup EtherCAT CPU/Expansion Racks I/O Map Controller Setup Motion Control Setup Cam Data Settings Event Settings Task Settings Data Trace Settings
2	Set the task period to 1 ms in the Task Settings View on the Sysmac Studio so that the period matches the sampling time of the Controller on the Simulink.  Configurations and Setup Task Settings Task Settings Priority-4 Primary Periodic Task PrimaryTask 1ms 500us 1ms 2ms 4ms Detect 5ms(Period tin 04 3
3	Select the MainTB program in the Program Assignment Settings View.  Configurations and Setup Task Settings Program Assignment Settings PrimaryTask Program name 1 MainTB Initial status Run
4	Select ***Run*** from the **Simulation** Menu of the Sysmac Studio.  PLCCoderDemoMC - new_Controller_0 - Sysmac Studio File Edit View Insert Project Controller Simulation Tools Help Run F5 Run in PROGRAM mode Alt+F5 Pause Ctrl+Alt+Break Stop Shift+F5 Step Execution F10 Step In F11 Step Out Shift+F11 Continuous Step Execution Execute One Scan Jump to Current Position Breakpoint Window Alt+F9 Set/Clear Breakpoint F9 Clear All Breakpoints Ctrl+Shift+F9 Calibration Run in Execution Time Estimation Mode Transfer All Variable Present Values Start NS Integrated Simulation PLCCoderDemoMC new_Controller_0 Configurations and Setup EtherCAT CPU/Expansion Racks I/O Map Controller Setup Motion Control Setup Cam Data Settings Event Settings Task Settings Data Trace Settings Programming Conf Task Sett

5 Double-click **TestBench** in the Multiview Explorer to display the program.



6 Confirm that *testVerify* is True and *testCycleNum* is the value of *TEST_CYCLE_COUNT* written in the comment.



```
(* TEST_CYCLE_COUNT = 3001                 *)
IF need_init ▶ False  THEN
    testVerify ▶ True  := BOOL#TRUE;
    testCycleNum ▶ 3001  := DINT#0;
    need_init ▶ False  := BOOL#FALSE;
END_IF;


IF testVerify ▶ True  THEN

    IF testCycleNum ▶ 3001  < DINT#3001 THEN
```

You can confirm that calculation accuracy of the output data is the same level as the Simulink (within the acceptable error range) if *testVerify* is True.
You can also confirm that the simulation has been completed if *testCycleNum* is the value of *TEST_CYCLE_COUNT* written in the comment.

Additional Information

The initial value of the acceptable error depends on the data type as shown below. Set an appropriate value according to the actual application.

■ Integer data: 0 (Match)

```
IF testVerify AND (out_Out1 <> cycle_Out1) THEN
    testVerify := BOOL#FALSE;
END_IF;
```

■ REAL data: 0.0001

```
IF testVerify AND (ABS(out_Out1 - cycle_Out1) > REAL#0.0001) THEN
    testVerify := BOOL#FALSE;
END_IF;
```

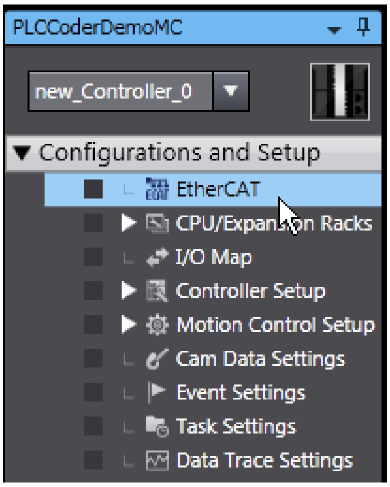
■ LREAL data: 1.0E-5

```
IF testVerify AND (ABS(out_Out1 - cycle_Out1) > LREAL#1.0E-5) THEN
    testVerify := BOOL#FALSE;
END_IF;
```

3.2.4. Creating the EtherCAT Network Configuration

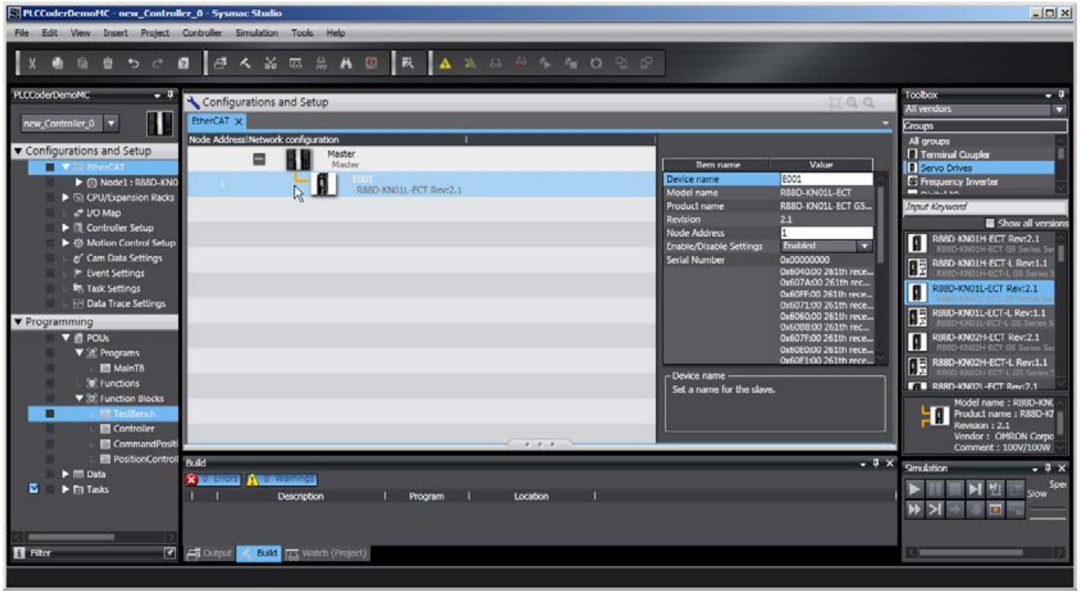
You register a R88D-KN01L-ECT Servo Drive that operates as axis 0 on the EtherCAT network configuration.

1 Double-click **EtherCAT** in the Multiview Explorer to display the EtherCAT Tab Page where you edit the EtherCAT network configuration.



2 Drag the R88D-KN01L-ECT from the Toolbox to the master.

The Servo Drive is added under the master with a node address of 1.



Item name	Value
Device name	E001
Model name	R88D-KN01L-ECT
Product name	R88D-KN01L-ECT G5...
Revision	2.1
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000

Device name
Set a name for the slave.

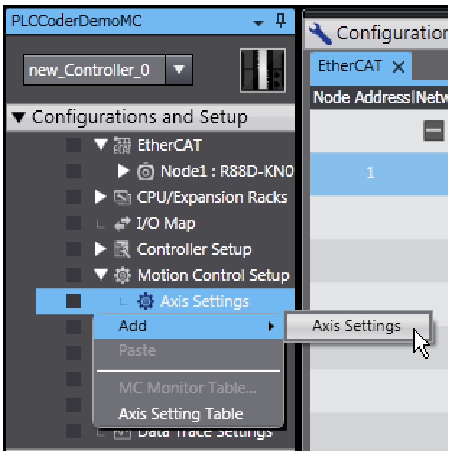
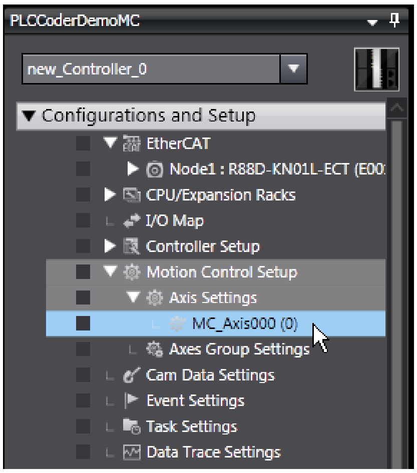
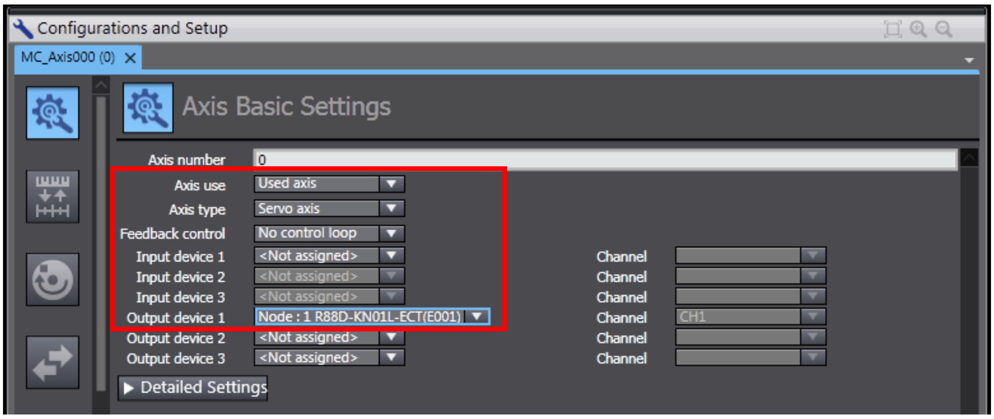
Additional Information

To use digital I/O devices, analog I/O devices, and encoder input devices, add the devices using the same procedure. For data access to the devices that you added, register the device variables in the I/O Map.

The examples for using GX-AD0471 Analog Input Terminal and GX-DA0271 Analog Output Terminal are provided as samples. Refer to the *Sample File No. 4 PLCCoderDemoMC_ADDA.mdl* and *No. 5 PLCCoderDemoMC_ADDA.smc2* that are provided separately.

3.2.5. Setting the Axis

You add an axis to control the Servo Drive, assign the Servo Drive to the axis, and make the axis parameter settings.

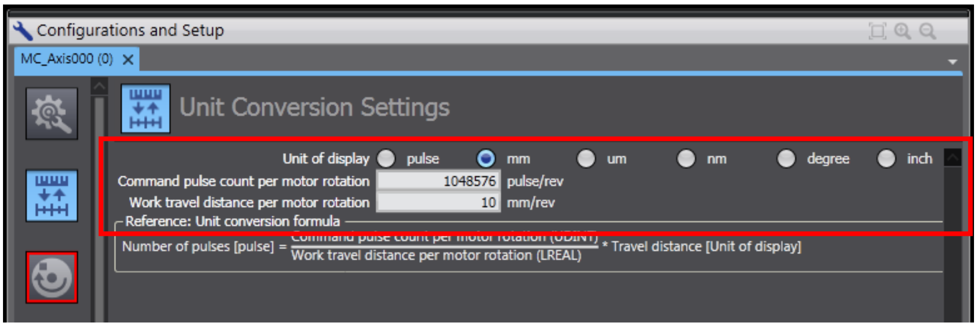
1	Double-click **Motion Control Setup** in the Multiview Explorer and right-click **Axis Settings** and select ***Add*** - ***Axis Settings*** from the menu.  PLCCoderDemoMC new_Controller_0 Configurations and Setup EtherCAT Node1 : R88D-KN0 CPU/Expansion Racks I/O Map Controller Setup Motion Control Setup Axis Settings Add Paste MC Monitor Table... Axis Setting Table Data Trace Settings Axis Settings Configuration EtherCAT Node Address Netw 1
2	Double-click **MC_Axis000(0)** (Axis 0) that was added under **Motion Control Setup** - **Axis Settings** in the Multiview Explorer to display the axis parameter setting view.  PLCCoderDemoMC new_Controller_0 Configurations and Setup EtherCAT Node1 : R88D-KN01L-ECT (E00 CPU/Expansion Racks I/O Map Controller Setup Motion Control Setup Axis Settings MC_Axis000 (0) Axes Group Settings Cam Data Settings Event Settings Task Settings Data Trace Settings
3	Make the Axis Basic Settings as shown below to assign the Servo Drive to the axis. Axis type: Servo axis Output device 1: Node: 1 R88D-KN01L-ETC(E001)  Configurations and Setup MC_Axis000 (0) Axis Basic Settings Detailed Settings Axis number: 0 Axis use: Used axis Axis type: Servo axis Feedback control: No control loop Input device 1: <Not assigned> Channel Input device 2: <Not assigned> Channel Input device 3: <Not assigned> Channel Output device 1: Node : 1 R88D-KN01L-ECT(E001) Channel: CH1 Output device 2: <Not assigned> Channel Output device 3: <Not assigned> Channel

4 Make the Unit Conversion Settings according to the mechanical configuration.

Unit of display: mm

Command pulse count per motor rotation: 1048576 pulse/rev

Work travel distance per motor rotation: 10 mm/rev



Unit Conversion Settings

Unit of display ○ pulse ● mm ○ um ○ nm ○ degree ○ inch

Command pulse count per motor rotation 1048576 pulse/rev

Work travel distance per motor rotation 10 mm/rev

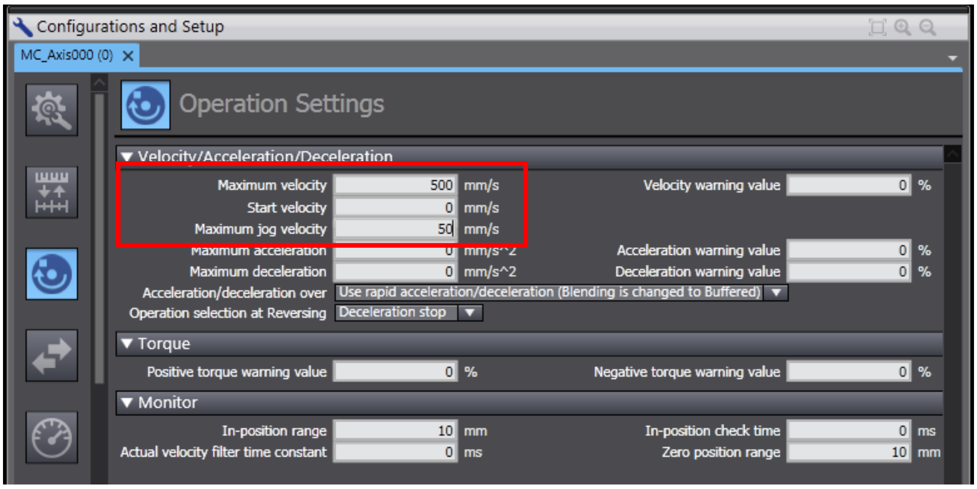
Reference: Unit conversion formula

Number of pulses [pulse] = Command pulse count per motor rotation (UDINT) / Work travel distance per motor rotation (LREAL) * Travel distance [Unit of display]

5 Make the Operation Settings according to the mechanical configuration.

Maximum velocity: 500 mm/s

Maximum jog velocity: 50 mm/s



Operation Settings

▼ Velocity/Acceleration/Deceleration

Maximum velocity	500 mm/s	Velocity warning value	0 %
Start velocity	0 mm/s		
Maximum jog velocity	50 mm/s		
Maximum acceleration	0 mm/s^2	Acceleration warning value	0 %
Maximum deceleration	0 mm/s^2	Deceleration warning value	0 %

Acceleration/deceleration over Use rapid acceleration/deceleration (Blending is changed to Buffered) ▼

Operation selection at Reversing Deceleration stop ▼

▼ Torque

Positive torque warning value	0 %	Negative torque warning value	0 %

▼ Monitor

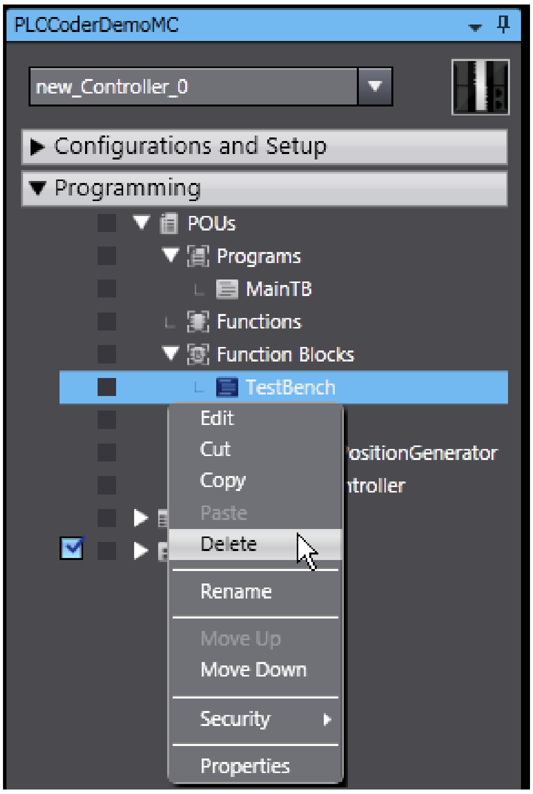
In-position range	10 mm	In-position check time	0 ms
Actual velocity filter time constant	0 ms	Zero position range	10 mm

3.2.6. Creating Programs

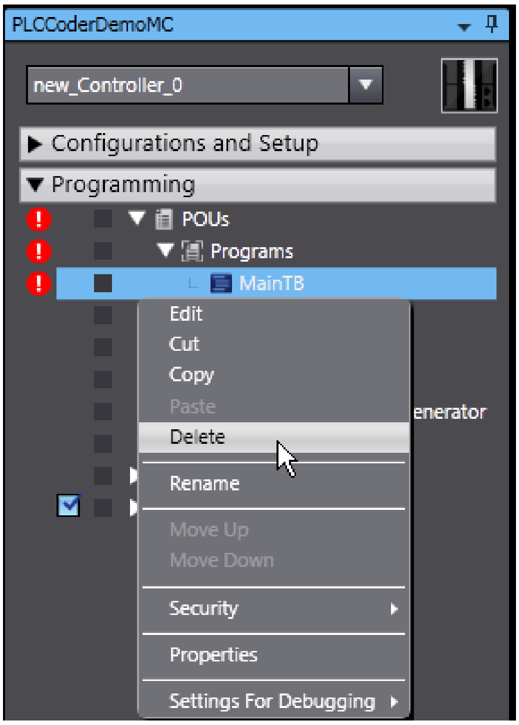
You create a program for calling the function blocks whose code was outputted by the Simulink PLC Coder and a program for outputting command values to the Servo Drive.

1 Delete *TestBench* and *MainTB* because they are used for the test to check the calculation accuracy.

Right-click **TestBench** in the Multiview Explorer and select ***Delete*** from the menu.

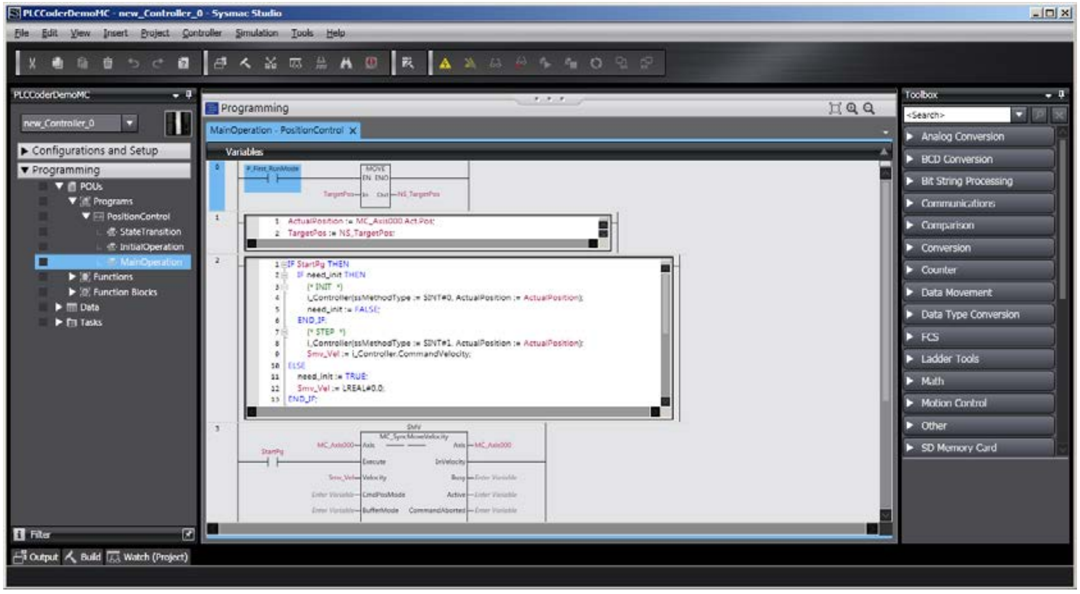


Right-click **MainTB** in the Multiview Explorer and select ***Delete*** from the menu.



2 Create the PositionControl program for the following processing.

- Servo ON (by executing the MC_Power instruction)
- Homing (by executing the MC_Home and MC_MoveZeroPosition instructions)
- Calculation of velocity command values by the Controller function block whose code was outputted by the Simulink PLC Coder
- Output of velocity command values to the Servo Drive (by executing the MC_SyncMoveVelocity instruction)
- Execution of the above operations by the signals from the Programmable Terminal
- Output of the execution status of the above operations to the Programmable Terminal



```
ActualPosition := MC_Axis000.Act.Pos;
TargetPos := NS_TargetPos;
```

```
IF StartPg THEN
  IF need_init THEN
    (* INIT *)
    i_Controller(ssMethodType := SINT#0, ActualPosition := ActualPosition);
    need_init := FALSE;
  END_IF;
    (* STEP *)
    i_Controller(ssMethodType := SINT#1, ActualPosition := ActualPosition);
    Smv_Vel := i_Controller.CommandVelocity;
ELSE
  need_init := TRUE;
  Smv_Vel := LREAL#0.0;
END_IF;
```

Precautions for Correct Use

The sample programming that is provided in this Guide includes only the programming that is required to operate the Servomotors. When programming actual applications, also program EtherCAT communications, device interlocks, I/O with other devices, and other control procedures.

Additional Information

Refer to the *Sample File No. 2 PLCCoderDemoMC.smc2* that is provided separately for the above program.

Additional Information

Refer to *4.1. Programming in Ladder Diagram Language* for programming in ladder diagram language.

Additional Information

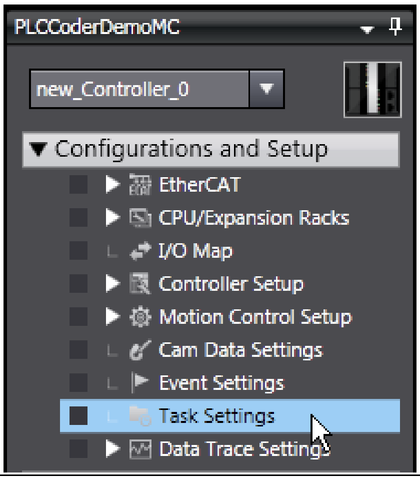
The instruction to use differs by the command given to the Servo Drive. Use the following instructions according to the command type.

Position command: MC_SyncMoveAbsolute
Velocity command: MC_SyncMoveVelocity
Torque command: MC_TorqueControl

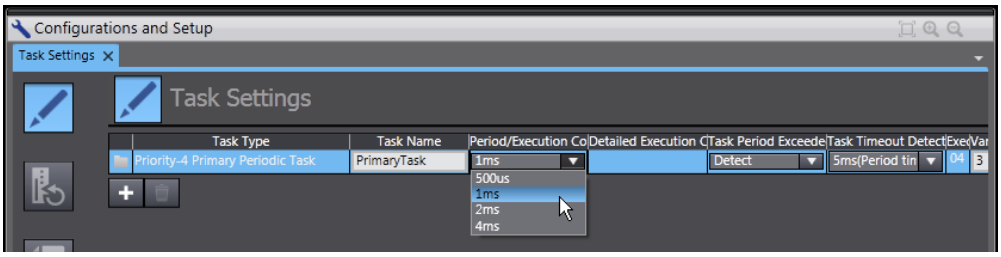
If you use a MC_TorqueControl instruction, the command values are not outputted cyclically. You need to write the program so that the command values are outputted cyclically. Refer to the *MC_mySyncTorqueControl* of the *Sample File No. 3 PLCCoderDemoMC_Torque.smc2* that is provided separately for the program.

Assigning the PositionControl program that you created to a task.

3 Double-click **Task Settings** in the Multiview Explorer to display the Task Settings Tab Page.

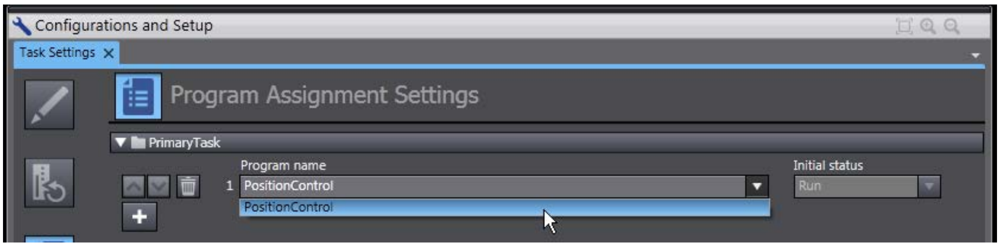


4 Set the task period to 1ms in the Task Settings View on the Sysmac Studio so that the period matches the sampling time of the Controller on the Simulink.



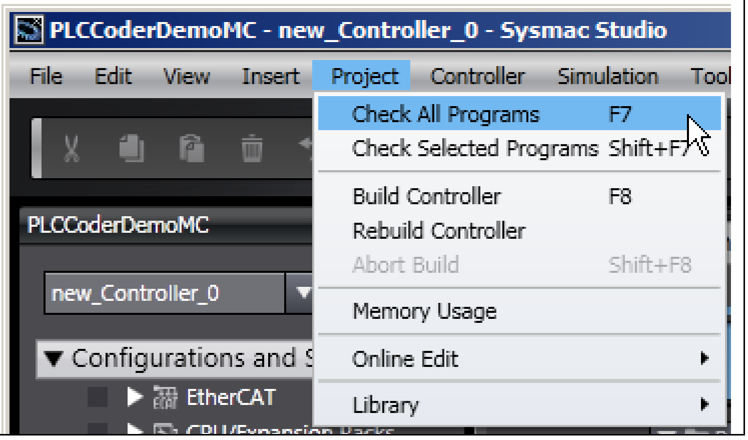
Task Type	Task Name	Period/Execution Co	Detailed Execution C	Task Period Exceede	Task Timeout Detect	Exe	Var
Priority-4 Primary Periodic Task	PrimaryTask	1ms		Detect	5ms(Period tin	04	3

5 In the Program Assignment Settings View, select the PositionControl program that you created.



Program name
1 PositionControl

Initial status
Run

6	Check the program that you created. Select ***Check All Programs*** from the **Project** Menu. 

3.2.7. Creating the Programming Terminal Screen

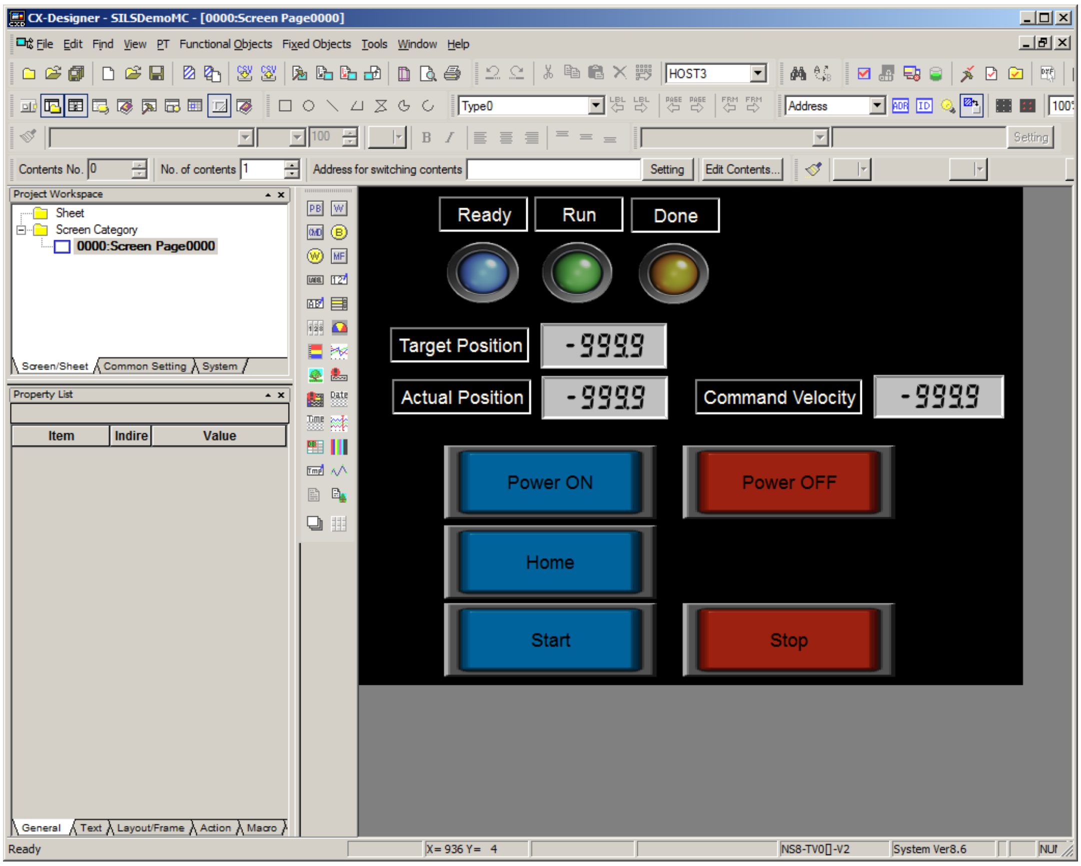
You create a Programmable Terminal screen with the CX-Designer.

Additional Information

Refer to the *CX-Designer User's Manual* (Cat. No. V099) or online help of the CX-Designer for how to use the CX-Designer.

1 Create a Programmable Terminal screen for the following processing.

- Servo ON/OFF operation and status display
- Homing operation
- Start/stop of movement to the command position and status display
- Status display of completion of movement to the command position
- Command position setting and display
- Current position display
- Velocity command value display



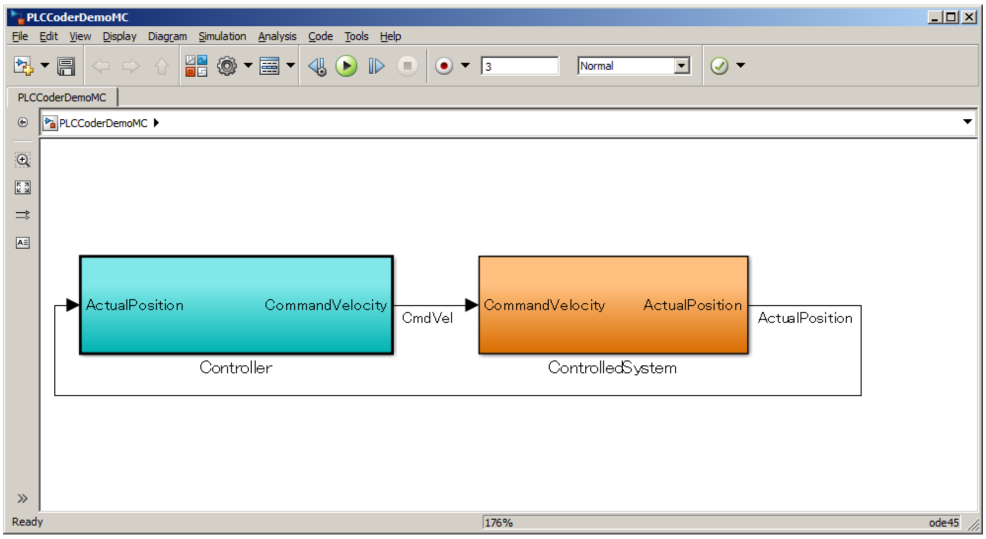
Additional Information

Refer to the *Sample File No. 8 SILSDemoMC.zip* that is provided separately for the above program.

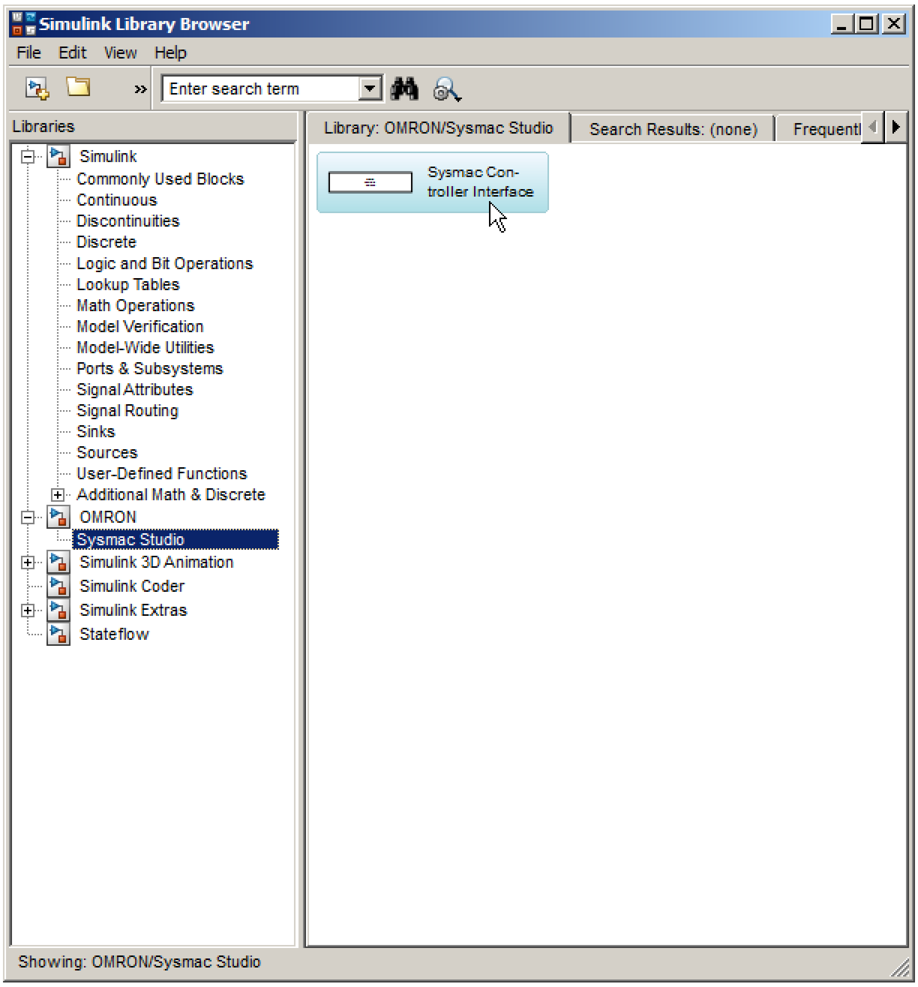
3.2.8. Preparing the SILS (Software In the Loop Simulation)

You add the Sysmac Controller Interface block to the Simulink model and make the setting for data exchange between Simulink and Sysmac Studio.

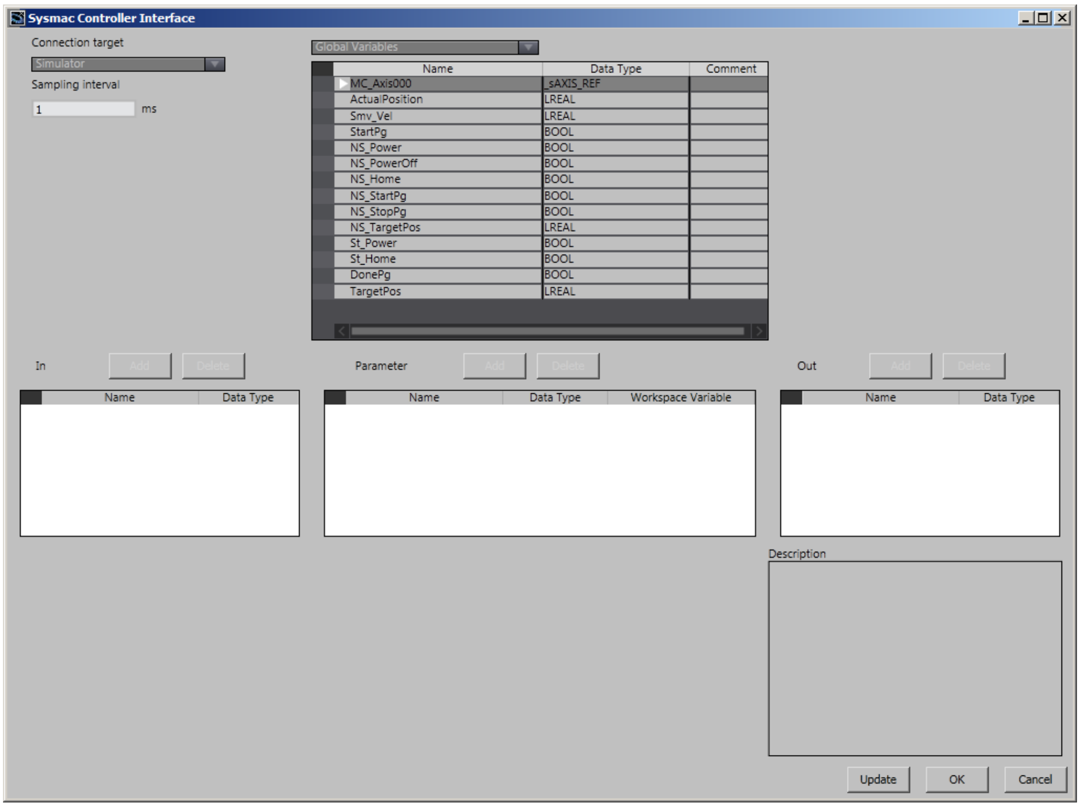
1 Open the *Sample File No. 1 PLCCoderDemoMC.mdl* that is provided separately on the Simulink.



2 Select the Sysmac Controller Interface block from the Simulink Library Browser and add it to the Simulink model that you opened in Step 1.



3 Double-click the Sysmac Controller Interface block that you added in Step 2 and display the dialog box where to make the setting for data exchange between Simulink and Sysmac Studio's Simulator.



Sysmac Controller Interface

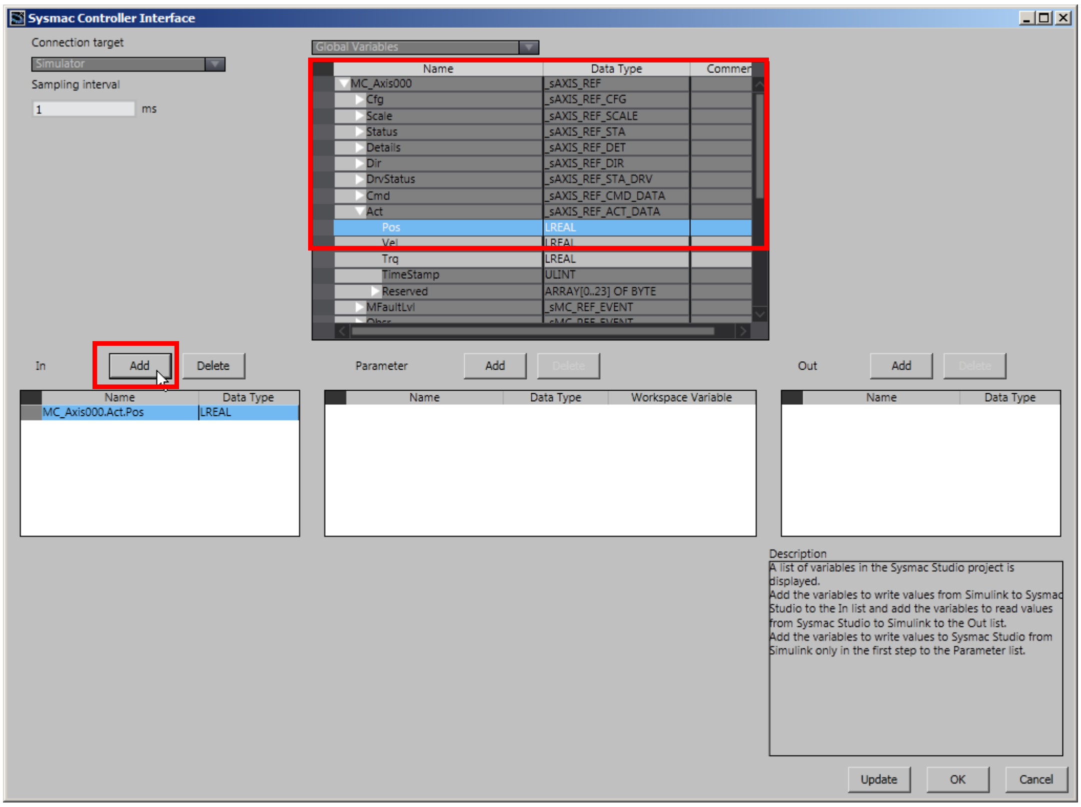
Connection target: Simulator

Sampling interval: 1 ms

Global Variables

Name	Data Type	Comment
MC_Axis000	_sAXIS_REF	
ActualPosition	LREAL	
Smv_Vel	LREAL	
StartPg	BOOL	
NS_Power	BOOL	
NS_PowerOff	BOOL	
NS_Home	BOOL	
NS_StartPg	BOOL	
NS_StopPg	BOOL	
NS_TargetPos	LREAL	
St_Power	BOOL	
St_Home	BOOL	
DonePg	BOOL	
TargetPos	LREAL	

4 Select *MC_Axis000.Act.Pos* from the list of variables in the Sysmac Studio project and click the **Add** Button for the In list to pass the actual current position calculated by the Simulink to the Sysmac Studio's Simulator.



Sysmac Controller Interface

Connection target: Simulator

Sampling interval: 1 ms

Global Variables

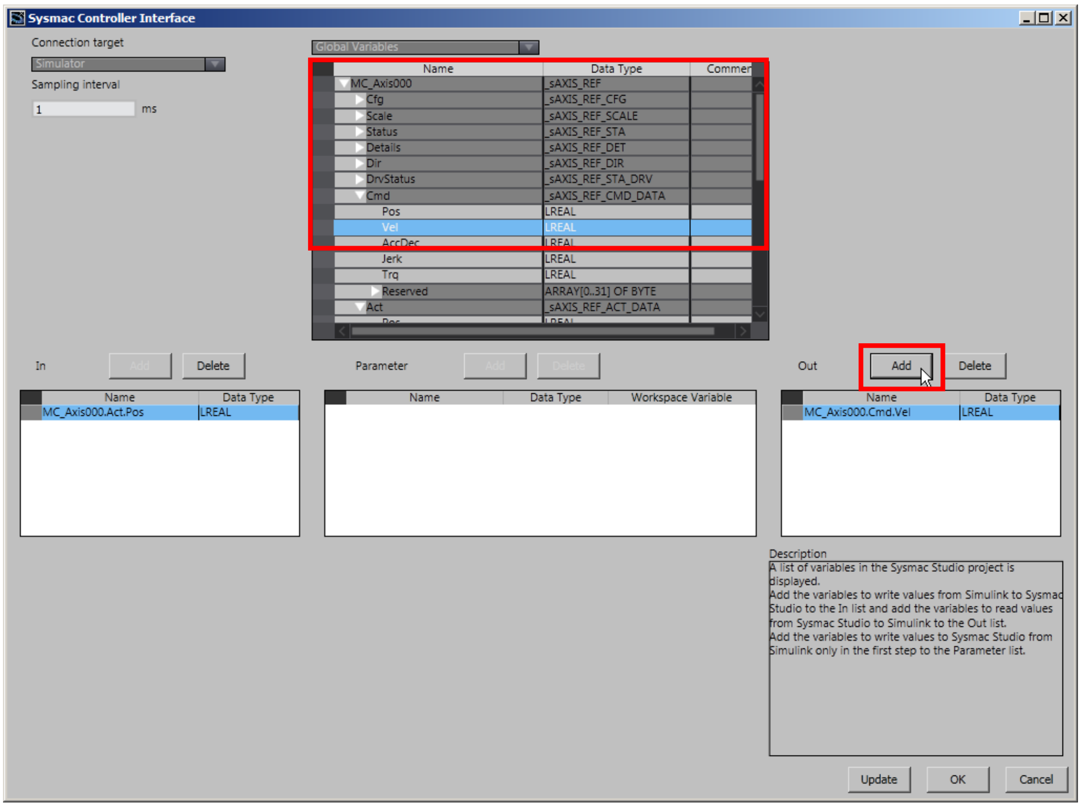
Name	Data Type	Commen
MC_Axis000	_sAXIS_REF	
Cfg	_sAXIS_REF_CFG	
Scale	_sAXIS_REF_SCALE	
Status	_sAXIS_REF_STA	
Details	_sAXIS_REF_DET	
Dir	_sAXIS_REF_DIR	
DrvStatus	_sAXIS_REF_STA_DRV	
Cmd	_sAXIS_REF_CMD_DATA	
Act	_sAXIS_REF_ACT_DATA	
Pos	LREAL	
Vel	LREAL	
Trq	LREAL	
TimeStamp	ULINT	
Reserved	ARRAY[0..23] OF BYTE	
MFaultLvl	_sMC_REF_EVENT	

In

Name	Data Type
MC_Axis000.Act.Pos	LREAL

Description:
A list of variables in the Sysmac Studio project is displayed.
Add the variables to write values from Simulink to Sysmac Studio to the In list and add the variables to read values from Sysmac Studio to Simulink to the Out list.
Add the variables to write values to Sysmac Studio from Simulink only in the first step to the Parameter list.

5 Select *MC_Axis000.Cmd.Vel* from the list of variables in the Sysmac Studio project and click the **Add** Button for the Out list to pass the velocity command value for the Servo Drive calculated by the Sysmac Studio's Simulator to the Simulink.



Sysmac Controller Interface

Connection target: Simulator

Sampling interval: 1 ms

Global Variables

Name	Data Type	Comment
MC_Axis000	_sAXIS_REF	
Cfg	_sAXIS_REF_CFG	
Scale	_sAXIS_REF_SCALE	
Status	_sAXIS_REF_STA	
Details	_sAXIS_REF_DET	
Dir	_sAXIS_REF_DIR	
DrvStatus	_sAXIS_REF_STA_DRV	
Cmd	_sAXIS_REF_CMD_DATA	
Pos	LREAL	
Vel	LREAL	
AccDec	LREAL	
Jerk	LREAL	
Trq	LREAL	
Reserved	ARRAY[0..31] OF BYTE	
Act	_sAXIS_REF_ACT_DATA	
Pos	LREAL	

In

Name	Data Type
MC_Axis000.Act.Pos	LREAL

Parameter

Name	Data Type	Workspace Variable

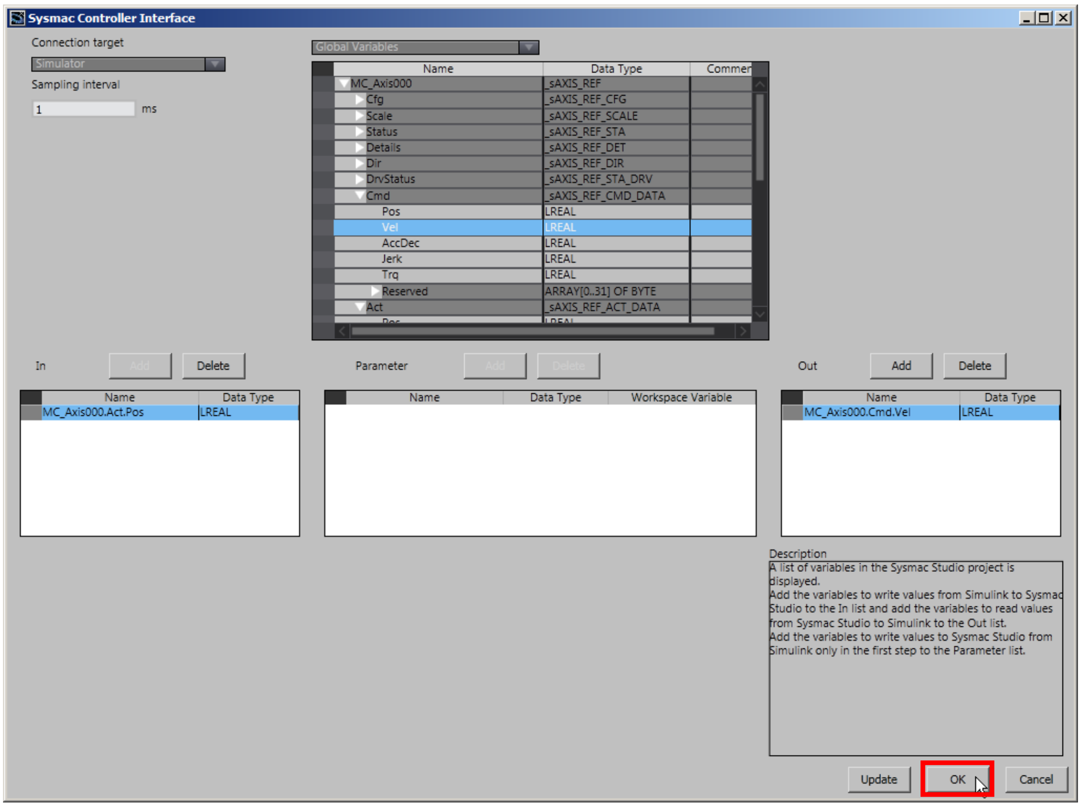
Out

Name	Data Type
MC_Axis000.Cmd.Vel	LREAL

Description

A list of variables in the Sysmac Studio project is displayed.
Add the variables to write values from Simulink to Sysmac Studio to the In list and add the variables to read values from Sysmac Studio to Simulink to the Out list.
Add the variables to write values to Sysmac Studio from Simulink only in the first step to the Parameter list.

6 Click the **OK** Button to close the dialog box.



Sysmac Controller Interface

Connection target: Simulator

Sampling interval: 1 ms

Global Variables

Name	Data Type	Comment
MC_Axis000	_sAXIS_REF	
Cfg	_sAXIS_REF_CFG	
Scale	_sAXIS_REF_SCALE	
Status	_sAXIS_REF_STA	
Details	_sAXIS_REF_DET	
Dir	_sAXIS_REF_DIR	
DrvStatus	_sAXIS_REF_STA_DRV	
Cmd	_sAXIS_REF_CMD_DATA	
Pos	LREAL	
Vel	LREAL	
AccDec	LREAL	
Jerk	LREAL	
Trq	LREAL	
Reserved	ARRAY[0..31] OF BYTE	
Act	_sAXIS_REF_ACT_DATA	
Pos	LREAL	

In

Name	Data Type
MC_Axis000.Act.Pos	LREAL

Parameter

Name	Data Type	Workspace Variable

Out

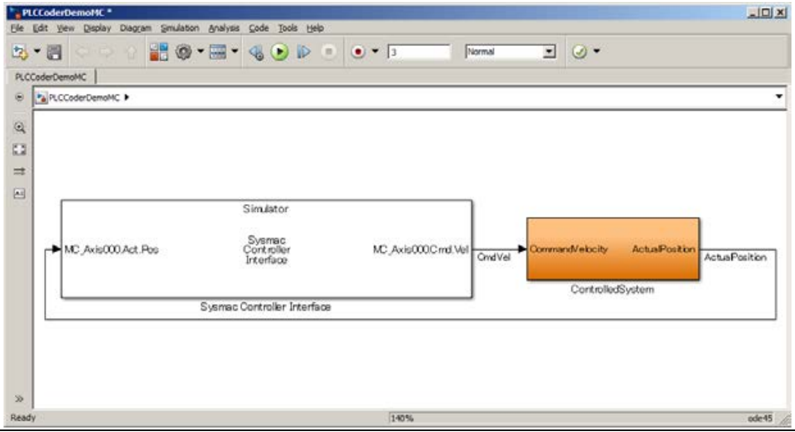
Name	Data Type
MC_Axis000.Cmd.Vel	LREAL

Description

A list of variables in the Sysmac Studio project is displayed.
Add the variables to write values from Simulink to Sysmac Studio to the In list and add the variables to read values from Sysmac Studio to Simulink to the Out list.
Add the variables to write values to Sysmac Studio from Simulink only in the first step to the Parameter list.

7 Delete the Controller block and replace it with the Sysmac Controller Interface block that you added.

Connect the input signal line and output signal line of the Sysmac Controller Interface block.



Additional Information

Refer to the *Sample File No. 9 SILSDemoMC.mdl* for the Simulink model that you created by the above operation.

Additional Information

You can use the Variant Subsystem to make the same Simulink model for the simulation using the Simulink only (MILS: Model In the Loop Simulation) and the simulation using Simulink and Sysmac Studio (SILS: Software In the Loop Simulation).

Refer to the *Sample File No. 10 MILS_SILS_DemoMC.mdl*.

Additional Information

You can add the following axis variable members to the In list.

Variable name (Member)	Name
Act.Pos	Actual current position
Act.Vel	Actual current velocity
Act.Trq	Actual current torque

However, you can add only the axes whose *Axis use* parameter is set to *Unused axis (changeable to used axis)* or *Used axis* and whose *Axis type* parameter is set to Servo axis or *Encoder axis*. Like the actual access from Servo Drive or encoder to Controller, these variables are converted to the data type for the PDO communications (*Act.Pos* and *Act.Vel* are converted to DINT data and *Act.Trq* is converted to INT data) for unit conversion of axis variables (i.e., calculation based on the electronic gear ratio setting) using the command pulse count per motor rotation and work travel distance per motor rotation.

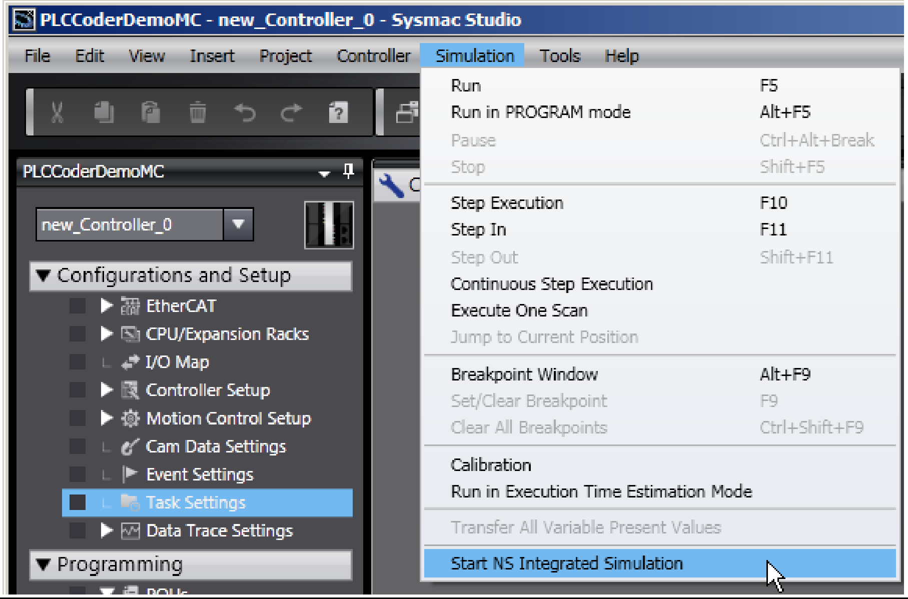
Additional Information

Add the variable whose value is passed from Simulink to Sysmac Studio only in the first step to the Parameter list.

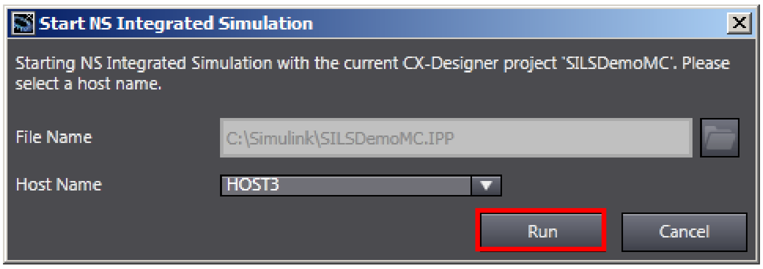
3.2.9. Debugging by Simulation

You debug the programs and screens that you created by the SILS (Software In the Loop Simulation).

1 Select ***Start NS Integrated Simulation*** from the **Simulation** Menu of the Sysmac Studio.



2 Select the CX-Designer file that you created in *3.2.7 Creating the Programming Terminal Screen* and the host name in the Start NS Integrated Simulation Dialog Box and click the **Run** Button.

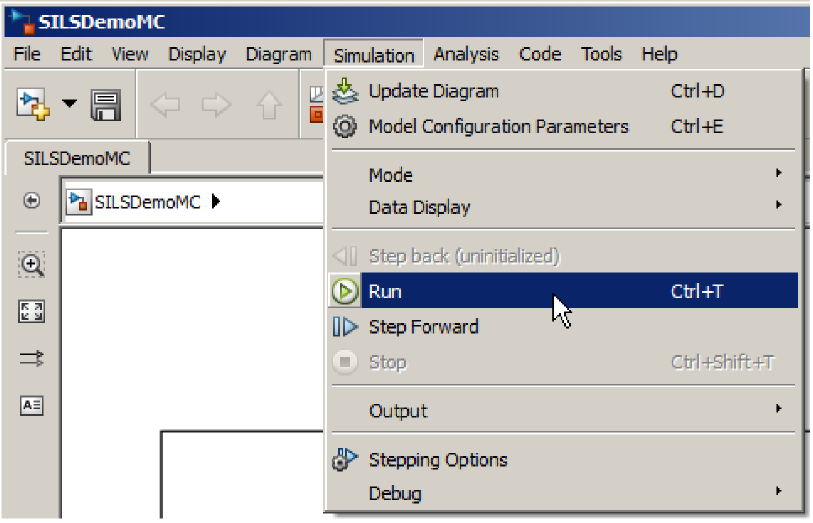

Start NS Integrated Simulation

Starting NS Integrated Simulation with the current CX-Designer project 'SILSDemoMC'. Please select a host name.

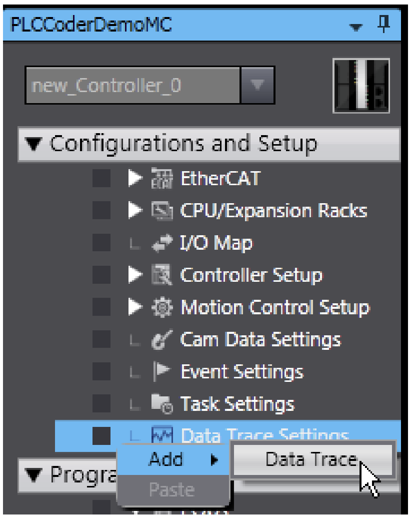
File Name: C:\Simulink\SILSDemoMC.IPP

Host Name: HOST3

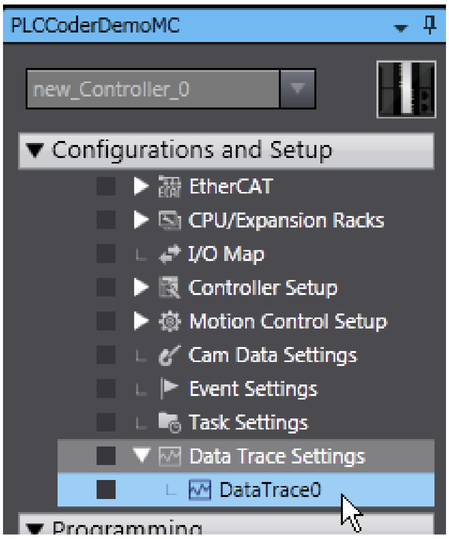
3 Select ***Run*** from the Simulation Menu of the Simulink.



4 Right-click **Data Trace Settings** in the Multiview Explorer and select ***Add* - *Data Trace*** from the menu to add DataTrace0.



5 Double-click **DataTrace0** that you added.



6 Make the trace settings as shown below.

Trigger condition: Rising edge of PositionControl.StartPg

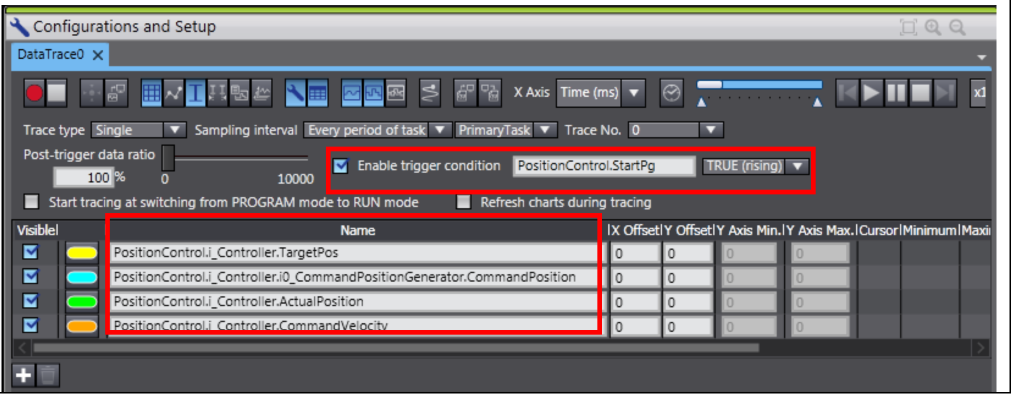
Trace target variables:

PositionControl.i_Controller.TargetPos

PositionControl.i_Controller.i0_CommandPositionGenerator.CommandPosition

PositionControl.i_Controller.ActualPosition

PositionControl.i_Controller.CommandVelocity



Trace type Single Sampling interval Every period of task PrimaryTask Trace No. 0

Post-trigger data ratio 100 % 0 10000

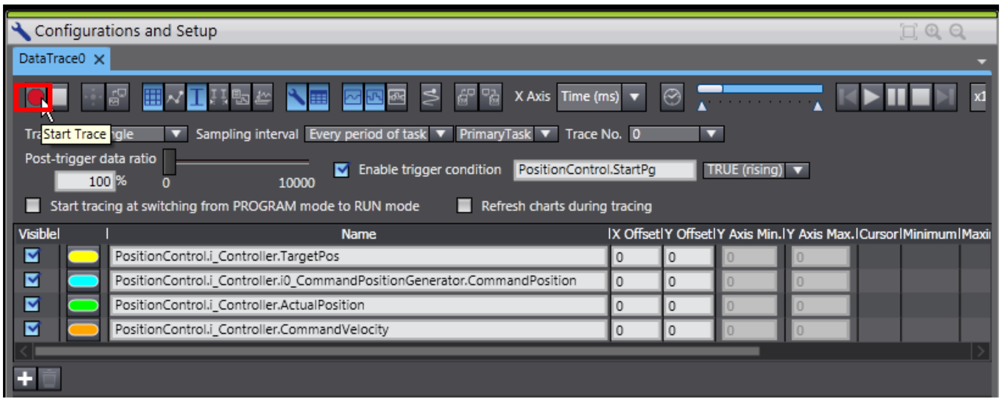
☑ Enable trigger condition PositionControl.StartPg TRUE (rising)

☐ Start tracing at switching from PROGRAM mode to RUN mode ☐ Refresh charts during tracing

Visible		Name	X Offset	Y Offset	Y Axis Min.	Y Axis Max.	Cursor	Minimum	Maxi
☑	🟨	PositionControl.i_Controller.TargetPos	0	0	0	0			
☑	🟦	PositionControl.i_Controller.i0_CommandPositionGenerator.CommandPosition	0	0	0	0			
☑	🟩	PositionControl.i_Controller.ActualPosition	0	0	0	0			
☑	🟧	PositionControl.i_Controller.CommandVelocity	0	0	0	0			

7 Click the **Start Trace** Button (with red filled circle icon) on the upper left part of the window to start data tracing.

The data trace function is started and waits for the trigger.



X Axis: Time (ms)

Tra[Start Trace]ngle ▼ Sampling interval: Every period of task ▼ PrimaryTask ▼ Trace No. 0 ▼

Post-trigger data ratio: 100 % (0 – 10000)

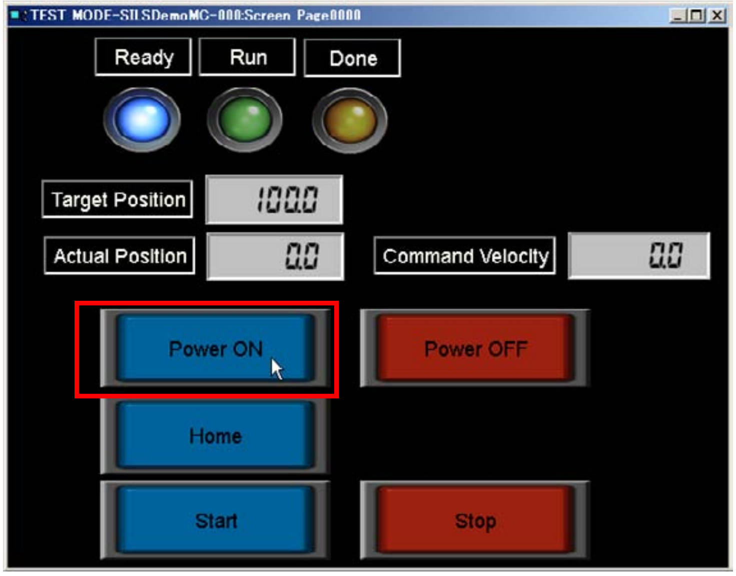
☑ Enable trigger condition: PositionControl.StartPg TRUE (rising) ▼

□ Start tracing at switching from PROGRAM mode to RUN mode □ Refresh charts during tracing

Visible		Name	X Offset	Y Offset	Y Axis Min.	Y Axis Max.	Cursor	Minimum	Maxi
☑	🟨	PositionControl.i_Controller.TargetPos	0	0	0	0			
☑	🟦	PositionControl.i_Controller.i0_CommandPositionGenerator.CommandPosition	0	0	0	0			
☑	🟩	PositionControl.i_Controller.ActualPosition	0	0	0	0			
☑	🟧	PositionControl.i_Controller.CommandVelocity	0	0	0	0			

8 Click the **Power ON** Button on the Test Window for the Programmable Terminal.

The Servo is turned ON and the **Ready** Lamp is lit.



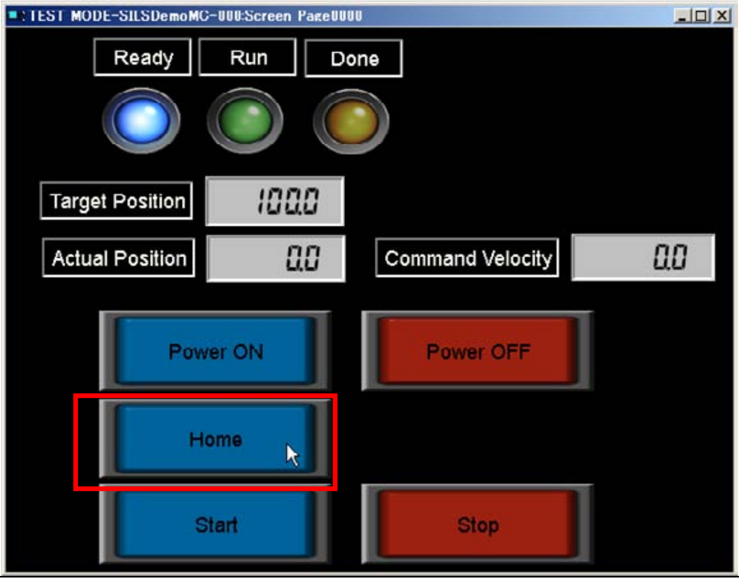
Target Position 100.0

Actual Position 0.0

Command Velocity 0.0

9 Click the **Home** Button on the Test Window for the Programmable Terminal.

The axis is returned to the home.



Target Position 100.0

Actual Position 0.0

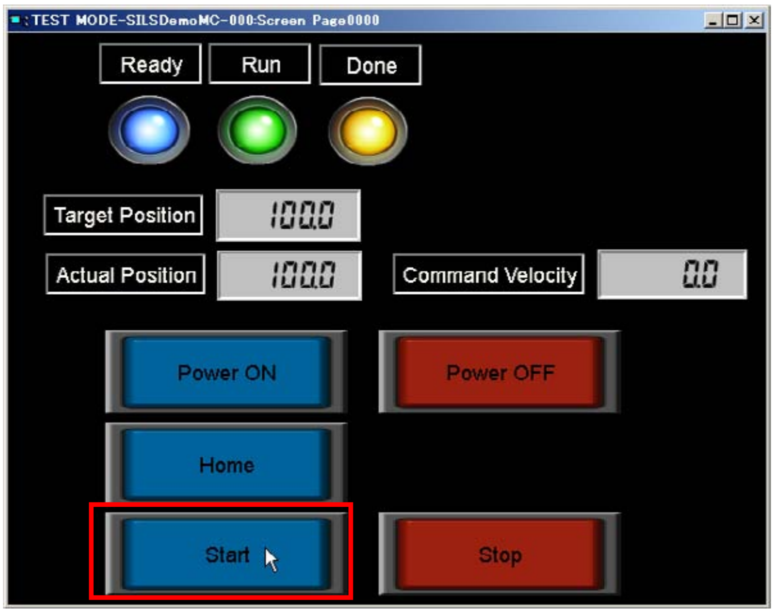
Command Velocity 0.0

10 Click the **Start** Button on the Test Window for the Programmable Terminal.

The axis starts moving to the Target Position and the **Run** Lamp is lit.

The Actual Position value and Command Velocity value change.

When the movement is completed, the **Done** Lamp is lit.



:TEST MODE-SILSDemoMC-000:Screen Page0000

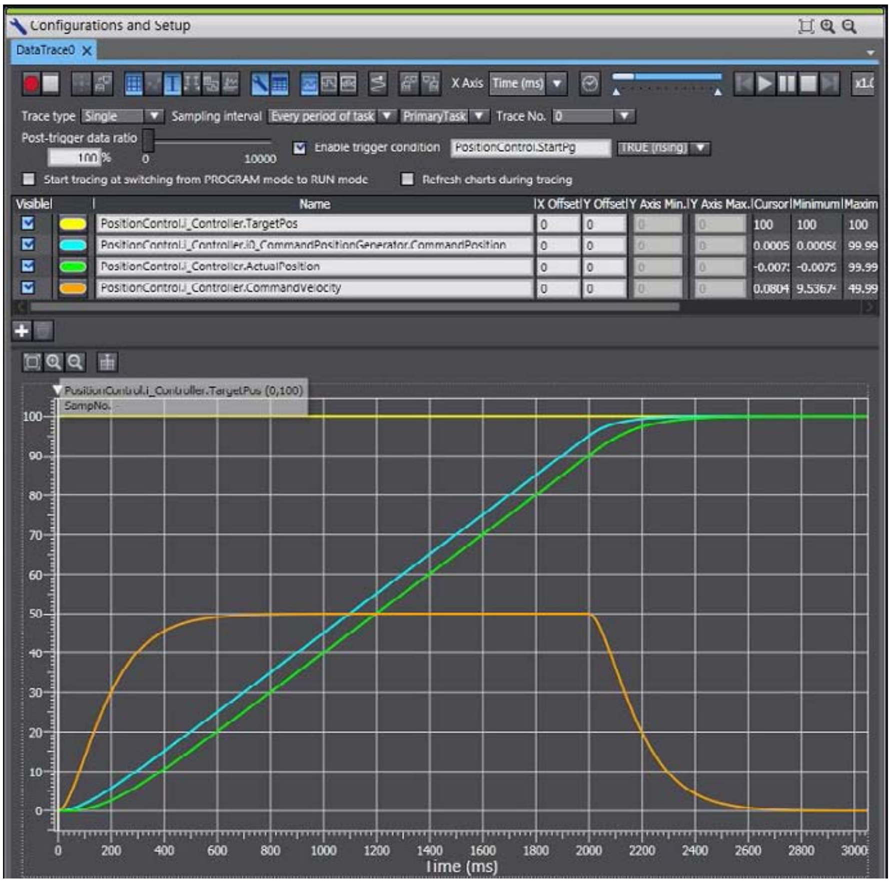
Target Position 100.0

Actual Position 100.0

Command Velocity 0.0

11 When you click the **Stop** Button (with write square icon) or the trace data becomes full, the data trace operation will stop and the results will be displayed.

You can confirm that you got the same trace results as the waveform shown in *1.2. The Servo System Constructed in this Guide* and *2.2. Designing the Control Algorithm*.



Configurations and Setup

DataTrace0

X Axis Time (ms)

Trace type Single Sampling interval Every period of task PrimaryTask Trace No. 0

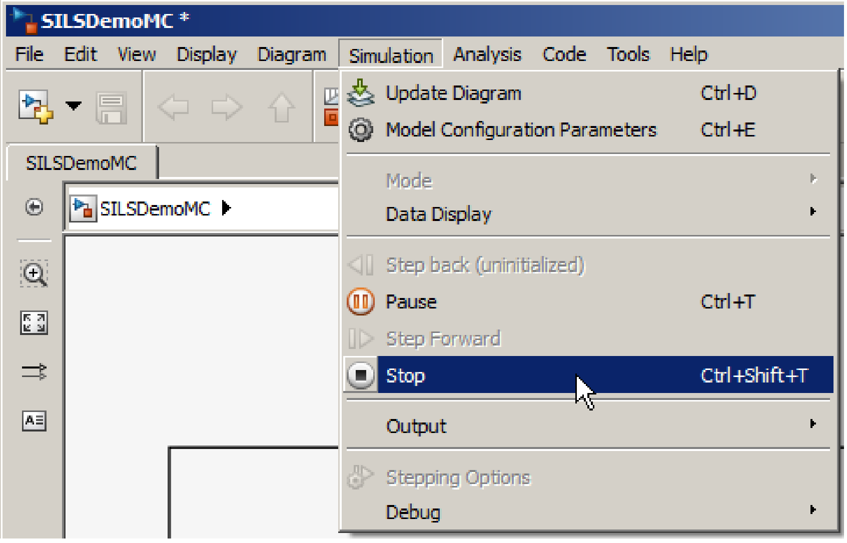
Post-trigger data ratio 100 % 0 10000

☑ Enable trigger condition PositionControl.StartPg TRUE (rising)

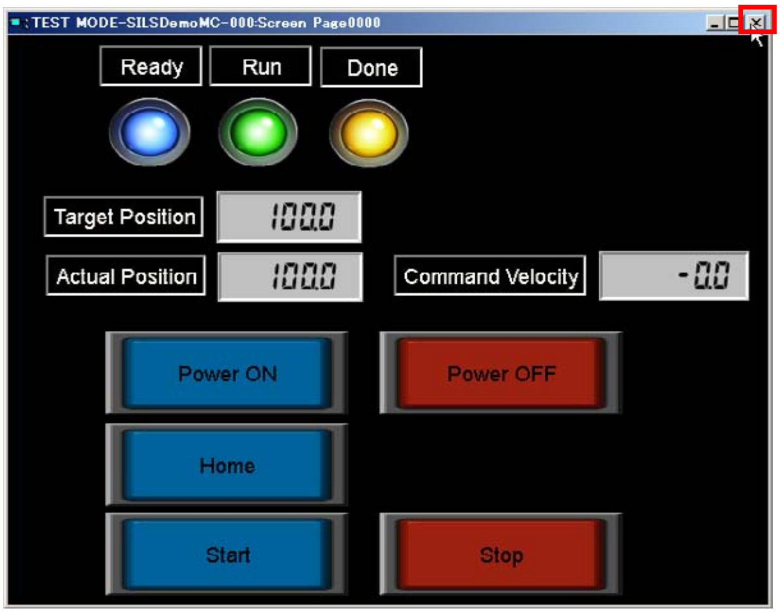
☐ Start tracing at switching from PROGRAM mode to RUN mode ☐ Refresh charts during tracing

Visible		Name	X Offset	Y Offset	Y Axis Min.	Y Axis Max.	Cursor	Minimum	Maxim
☑	🟨	PositionControl.i_Controller.TargetPos	0	0	0	0	100	100	100
☑	🟦	PositionControl.i_Controller.i0_CommandPositionGenerator.CommandPosition	0	0	0	0	0.0005	0.00050	99.99
☑	🟩	PositionControl.i_Controller.ActualPosition	0	0	0	0	-0.007:	-0.0075	99.99
☑	🟧	PositionControl.i_Controller.CommandVelocity	0	0	0	0	0.0804	9.5367	49.99

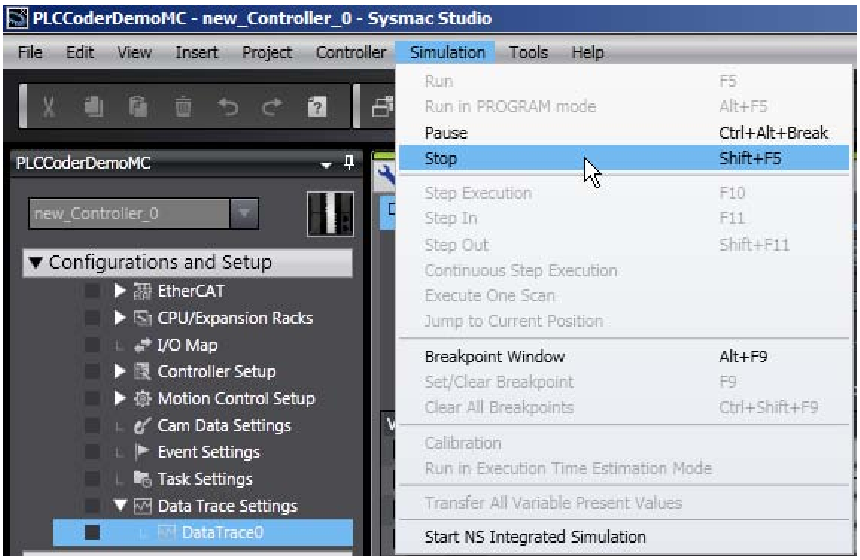
12 Select ***Stop*** from the Simulation Menu of the Simulink.


SILSDemoMC *
File Edit View Display Diagram Simulation Analysis Code Tools Help
SILSDemoMC
Update Diagram Ctrl+D
Model Configuration Parameters Ctrl+E
Mode
Data Display
Step back (uninitialized)
Pause Ctrl+T
Step Forward
Stop Ctrl+Shift+T
Output
Stepping Options
Debug

13 Click the Close Button ([x] button) on the Test Window for the Programmable Terminal to finish the test function.



14 Select ***Stop*** from the **Simulation** Menu of the Sysmac Studio.



Precautions for Correct Use

When the SIM_SetVelocity simulation instruction is used for the encoder axis, the Simulink cannot pass the value to the *Act.Vel* (current velocity) variable of the Sysmac Studio.
Do not use the SIM_SetVelocity simulation instruction to pass the value from the Simulink to the *Act.Vel* (current velocity) variable of the Sysmac Studio.

Precautions for Correct Use

Unit conversion of the axis variables (i.e., calculation based on the electronic gear ratio setting) uses the command pulse count per motor rotation and work travel distance per motor rotation at the simulation start of the Simulink. Therefore, if the command pulse count per motor rotation or work travel distance per motor rotation is changed by the MC_WriteAxisParameter instruction during the simulation, the Simulink cannot correctly write the values to the *Act.Pos* (actual current position) variable and the *Act.Vel* (actual current velocity) variable of the Sysmac Studio.
Do not change the command pulse count per motor rotation or work travel distance per motor rotation when the values are written from the Simulink to the *Act.Pos* (actual current position) variable and the *Act.Vel* (actual current velocity) variable of the Sysmac Studio.

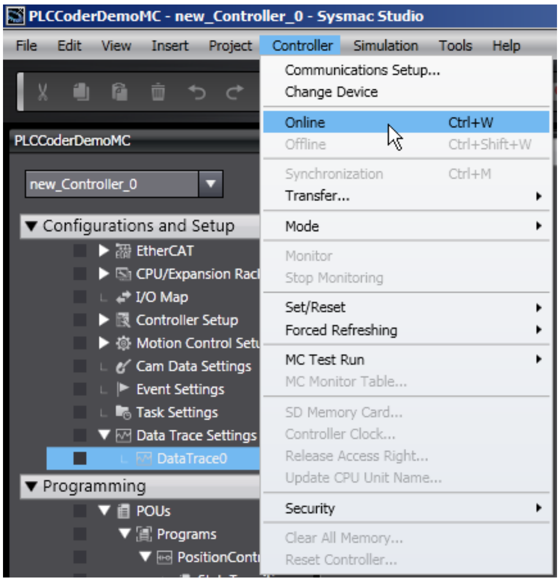
Additional Information

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the program debugging procedures.

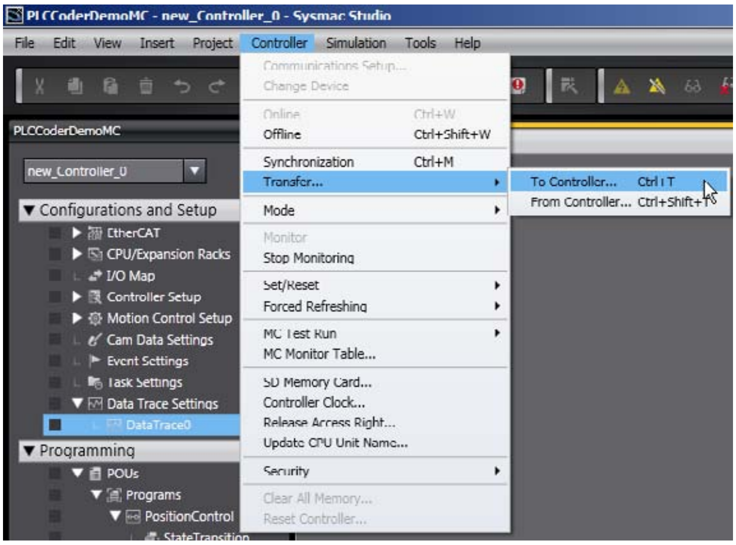
3.2.10. Transferring the Programs to the CPU Unit

You transfer the programs and parameter settings to the physical NJ-series CPU Unit.

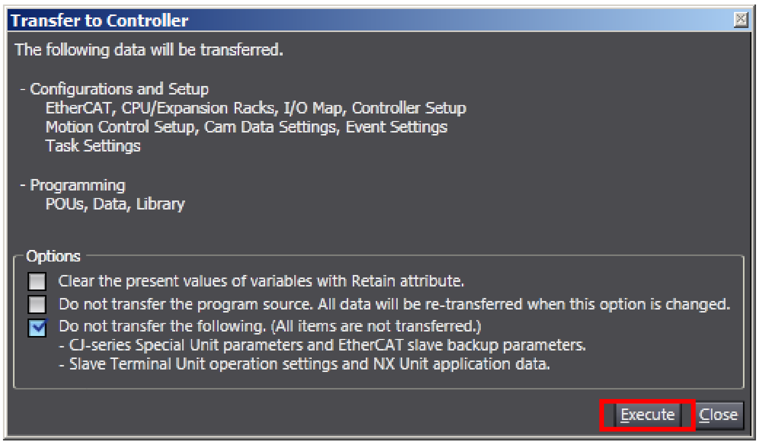
1 Select ***Online*** from the **Controller** Menu.



2 Select ***Transfer*** - ***To Controller*** from the **Controller** Menu.



3 Click the **Execute** Button.



Transfer to Controller

The following data will be transferred.

- Configurations and Setup
  EtherCAT, CPU/Expansion Racks, I/O Map, Controller Setup
  Motion Control Setup, Cam Data Settings, Event Settings
  Task Settings

- Programming
  POUs, Data, Library

Options

□ Clear the present values of variables with Retain attribute.

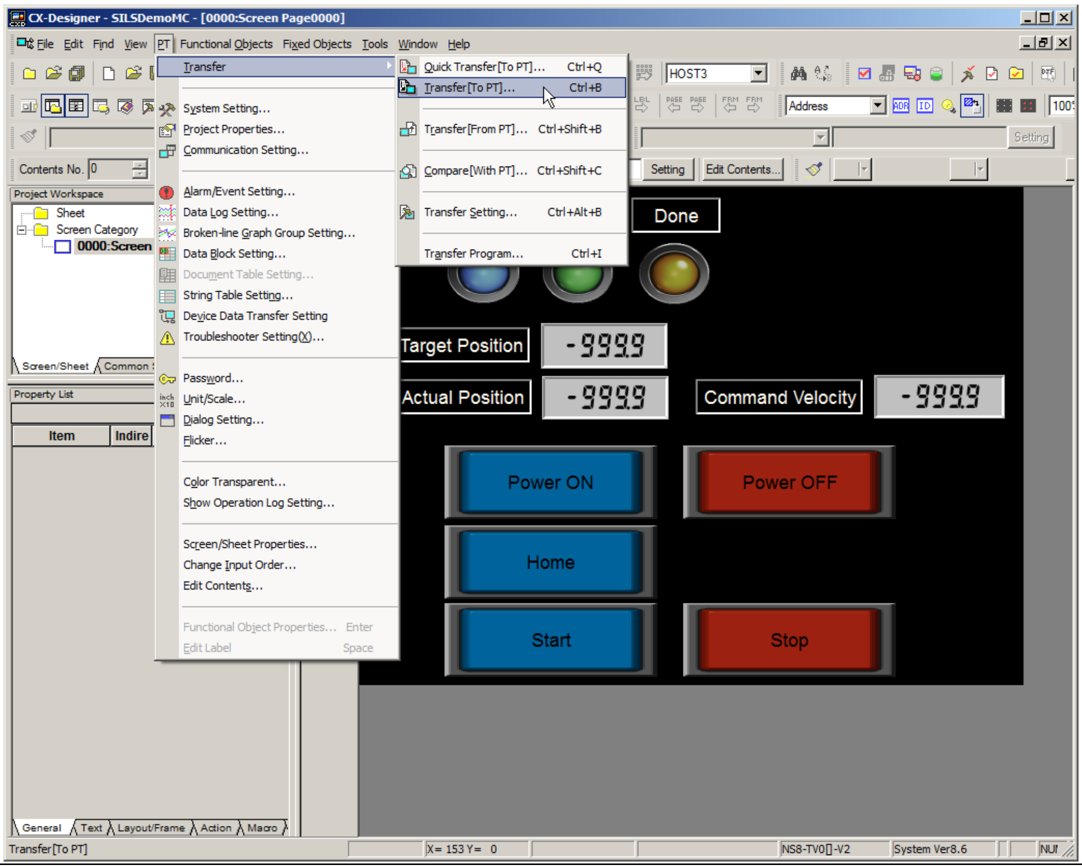
□ Do not transfer the program source. All data will be re-transferred when this option is changed.

☑ Do not transfer the following. (All items are not transferred.)
- CJ-series Special Unit parameters and EtherCAT slave backup parameters.
- Slave Terminal Unit operation settings and NX Unit application data.

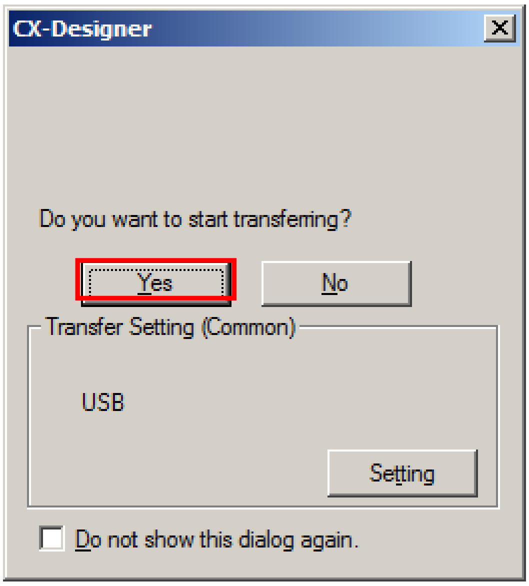
3.2.11. Transferring Screen Data to Programmable Terminal

You transfer the screen data created with the CX-Designer to the physical Programmable Terminal.

1 Select ***Transfer*** - ***Transfer[To PT]*** from the **PT** Menu of the CX-Designer.



2 Click the **Yes** Button.



Do you want to start transfering?

Transfer Setting (Common)

USB

☐ Do not show this dialog again.

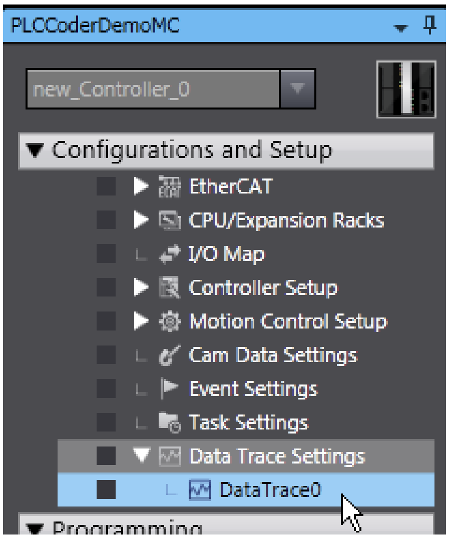
3.2.12. System Operation Check

You execute the operation according to the programs transferred to the physical device and check the operation using the data trace function.

Precautions for Correct Use

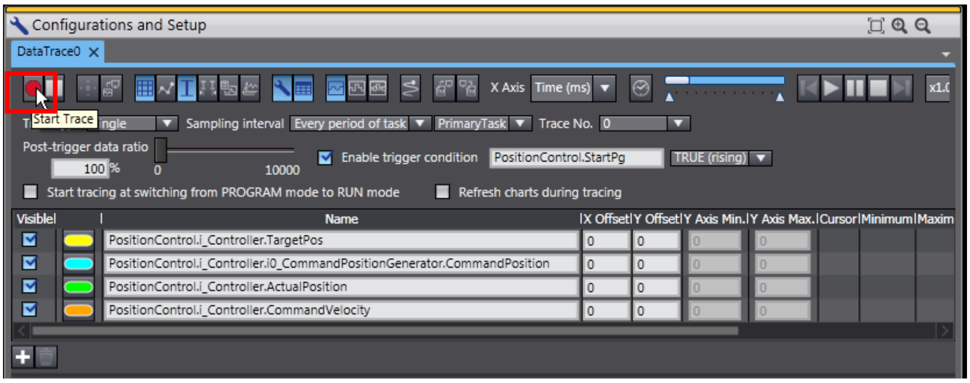
The physical motor will run. Thoroughly read and understand the manuals for all devices that make up the system to ensure that the system is used safely. Review the entire contents of these manuals, including all safety precautions, precautions for safe use, and precautions for correct use before the actual operation.

1 Double-click **DataTrace0**.



2 Click the **Start Trace** Button (with red filled circle icon) on the upper left part of the window to start data tracing.

The data trace function is started and waits for the trigger.



Time (ms)

T… ingle | Sampling interval: Every period of task | PrimaryTask | Trace No. 0

Post-trigger data ratio: 100 % (0 – 10000)

☑ Enable trigger condition: PositionControl.StartPg | TRUE (rising)

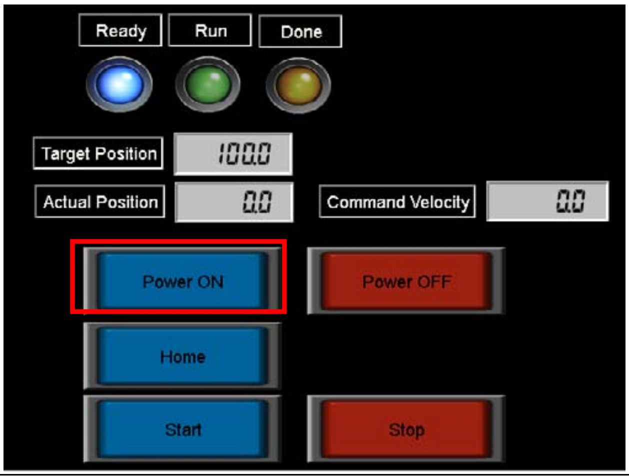
□ Start tracing at switching from PROGRAM mode to RUN mode

□ Refresh charts during tracing

Visible		Name	X Offset	Y Offset	Y Axis Min.	Y Axis Max.	Cursor	Minimum	Maxim
☑	🟨	PositionControl.i_Controller.TargetPos	0	0	0	0			
☑	🟦	PositionControl.i_Controller.i0_CommandPositionGenerator.CommandPosition	0	0	0	0			
☑	🟩	PositionControl.i_Controller.ActualPosition	0	0	0	0			
☑	🟧	PositionControl.i_Controller.CommandVelocity	0	0	0	0			

3 Press the **Power ON** Button on the physical Programmable Terminal.

The Servo is turned ON and the **Ready** Lamp is lit.



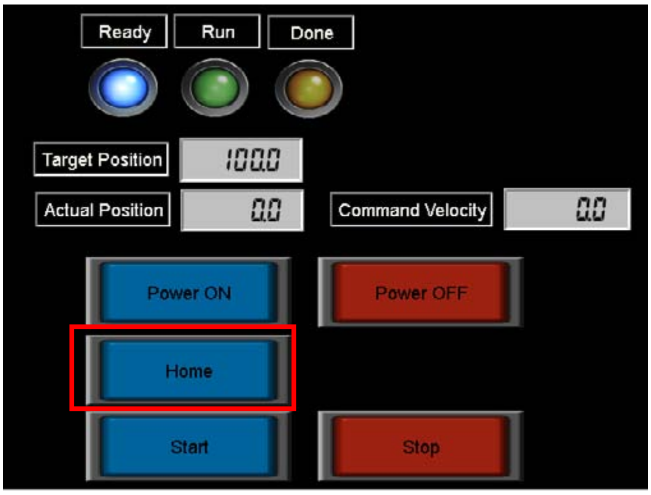
Target Position 100.0

Actual Position 0.0

Command Velocity 0.0

4 Press the **Home** Button on the physical Programmable Terminal.

The axis is returned to the home.



Target Position 100.0

Actual Position 0.0

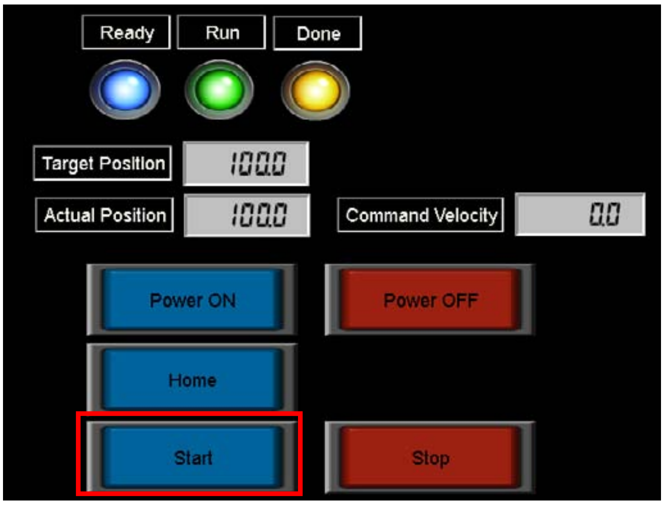
Command Velocity 0.0

5 Press the **Start** Button on the physical Programmable Terminal.

The axis starts moving to the Target Position and the **Run** Lamp is lit.

The Actual Position value and Command Velocity value change.

When the movement is completed, the **Done** Lamp is lit.



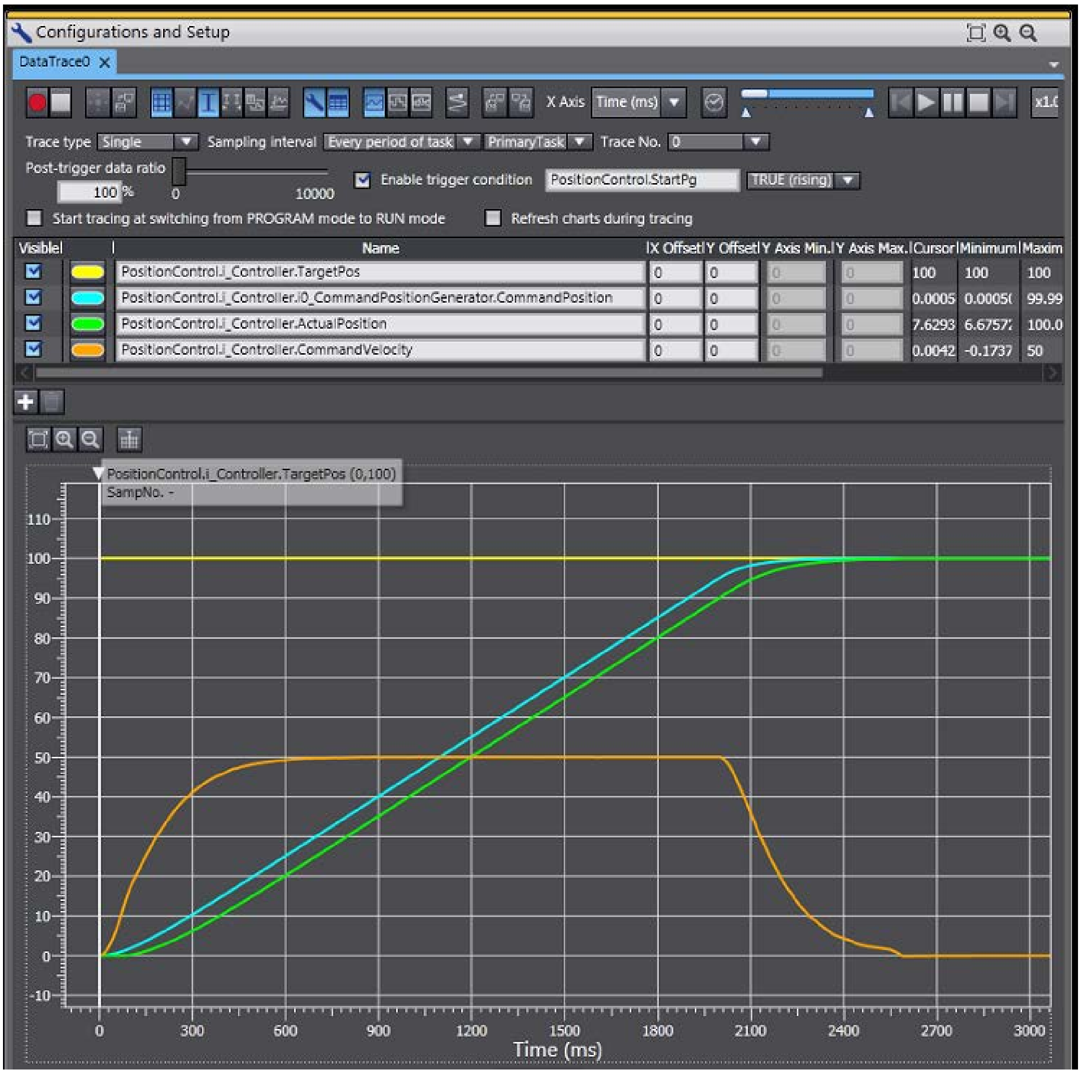
Target Position 100.0

Actual Position 100.0

Command Velocity 0.0

6 When you click the **Stop** Button (with write square icon) or the trace data becomes full, the data trace operation will stop and the results will be displayed.

You can confirm that you got the same trace results as the waveform shown in *1.2. The Servo System Constructed in this Guide* and *2.2. Designing the Control Algorithm*.



Configurations and Setup

DataTrace0

X Axis Time (ms)

Trace type Single Sampling interval Every period of task PrimaryTask Trace No. 0

Post-trigger data ratio 100 % 0 10000

☑ Enable trigger condition PositionControl.StartPg TRUE (rising)

□ Start tracing at switching from PROGRAM mode to RUN mode □ Refresh charts during tracing

Visible		Name	X Offset	Y Offset	Y Axis Min.	Y Axis Max.	Cursor	Minimum	Maxim
☑	🟨	PositionControl.i_Controller.TargetPos	0	0	0	0	100	100	100
☑	🟦	PositionControl.i_Controller.i0_CommandPositionGenerator.CommandPosition	0	0	0	0	0.0005	0.0005(	99.99
☑	🟩	PositionControl.i_Controller.ActualPosition	0	0	0	0	7.6293	6.6757:	100.0
☑	🟧	PositionControl.i_Controller.CommandVelocity	0	0	0	0	0.0042	-0.1737	50

PositionControl.i_Controller.TargetPos (0,100)
SampNo. -

4. Appendix

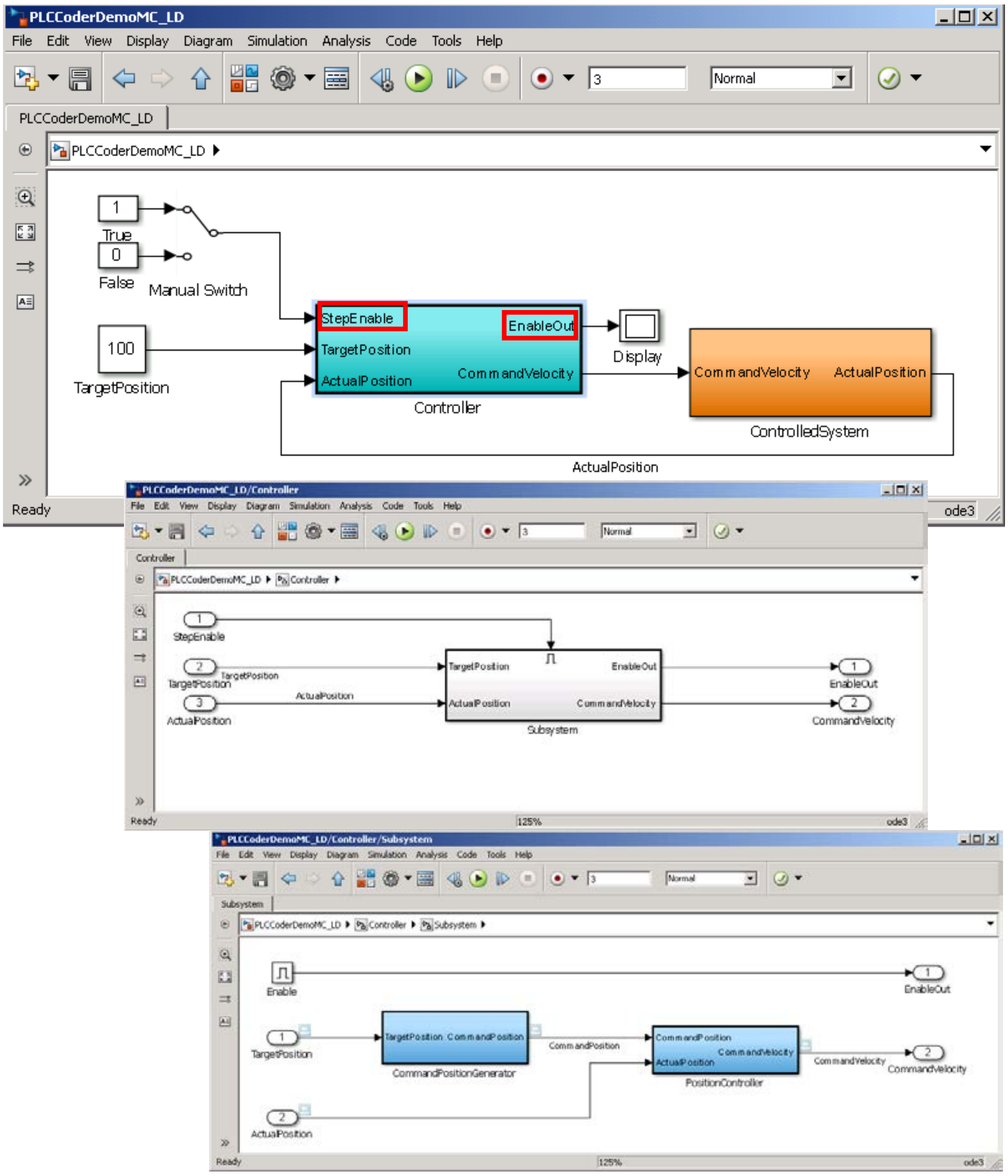
4.1. Programming in Ladder Diagram Language

To call a function block from a program written in the ladder diagram language, the function block must have at least one BOOL input variable and one BOOL output variable.
This section describes the procedure for adding boolean signals to the block on the Simulink.

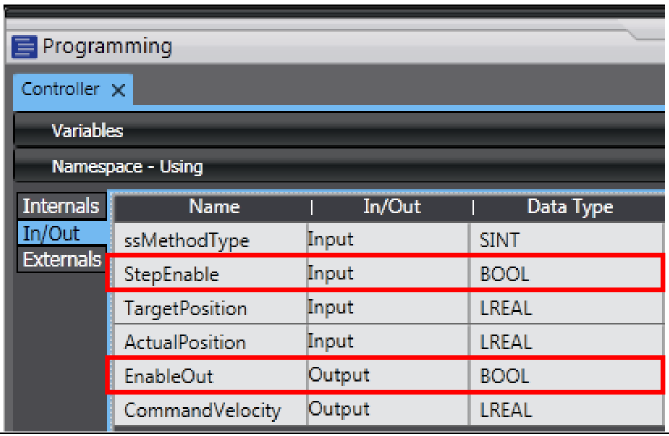
Additional Information

You also can add BOOL variables on the Sysmac Studio after importing the code without changing the block on the Simulink.

1 Add boolean signals to the Controller block on the Simulink.

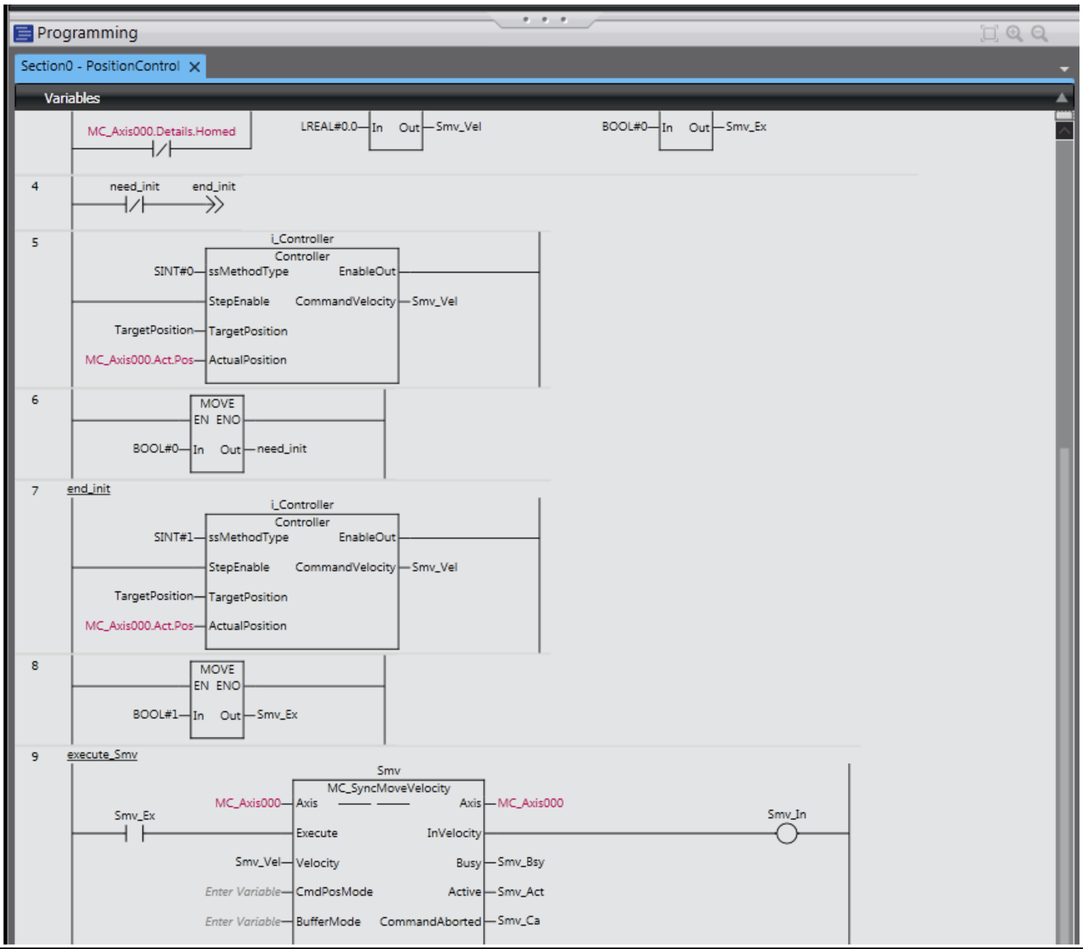


2 When the code is imported to the Sysmac Studio, the BOOL variables are added as shown below.



Name	In/Out	Data Type
ssMethodType	Input	SINT
StepEnable	Input	BOOL
TargetPosition	Input	LREAL
ActualPosition	Input	LREAL
EnableOut	Output	BOOL
CommandVelocity	Output	LREAL

3 The program to call the function block is written in the ladder diagram language as shown below.



Additional Information

Refer to the *Sample File No. 6 PLCCoderDemoMC_LD.mdl* that is provided separately for the Simulink model used in this section.

Refer to the *Sample File No. 7 PLCCoderDemoMC_LD.smc2* that is provided separately for the program used in this section.

4.2. Sample File List

The following sample files are related to this Guide.

We provide the sample files separately.

No.	File Name	Description
1	PLCCoderDemoMC.mdl	File that contains the Simulink model described in *2.2. Designing the Control Algorithm* of this Guide.
2	PLCCoderDemoMC.smc2	Sysmac Studio project file that contains Sysmac Studio programs described in *3.2.6 Creating Programs* of this Guide.
3	PLCCoderDemoMC_Torque.smc2	Sysmac Studio project file that contains the program to output torque commands cyclically.
4	PLCCoderDemoMC_ADDA.mdl	File that contains the Simulink model that shows the usage example of GX-AD0471 Analog Input Terminal and GX-DA0271 Analog Output Terminal.
5	PLCCoderDemoMC_ADDA.smc2	Sysmac Studio project file that shows the usage example of GX-AD0471 Analog Input Terminal and GX-DA0271 Analog Output Terminal.
6	PLCCoderDemoMC_LD.mdl	File that contains the Simulink model described in *4.1. Programming in Ladder Diagram Language* of this Guide.
7	PLCCoderDemoMC_LD.smc2	Sysmac Studio project file that contains Sysmac Studio programs described in *4.1. Programming in Ladder Diagram Language* of this Guide.
8	SILSDemoMC.zip	CX-Designer project file that is described in *3.2.7 Creating the Programming Terminal Screen* of this Guide.
9	SILSDemoMC.mdl	File that contains the Simulink model described in *3.2.8 Preparing the SILS (Software In the Loop Simulation)* of this Guide.
10	MILS_SILS_DemoMC.mdl	File that contains the Simulink model that shows the usage example of Variant Subsystem.

OMRON Corporation **Industrial Automation Company**
Tokyo, JAPAN

Contact: www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69-2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ELECTRONICS LLC
One Commerce Drive Schaumburg,
IL 60173-5302 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor: