ECATSEMUG

eCAT Servo Motor Example User Guide

Rev. 2 — 10 January 2024

Application note
COMPANY PUBLIC

Document information

Information	Content
Keywords	eCAT, Ethernet for Control Automation Technology, EtherCAT, SSC, EtherCAT Controller, COE, Servo Motor
Abstract	The servo motor example uses TwinCAT software as EtherCAT Master, i.MXRT1180 as EtherCAT Slave, which supports CiA402 profile to drive the motor.

1 Introduction

The servo motor example uses TwinCAT software as EtherCAT Master, i.MXRT1180 as EtherCAT Slave, which supports CiA402 profile to drive the motor.

The EtherCAT Slave Controller (ESC) integrated on i.MXRT1180 takes care of the EtherCAT communication as an interface between the EtherCAT fieldbus and slave application.

The EtherCAT Slave Stack Code (SSC) from Beckhoff Automation is a tool to generate the EtherCAT slave stack code in ANSI C. Based on this stack, this servo motor example currently supports Cyclic Synchronous Position(CSP) and Cyclic Synchronous Velocity (CSV) control modes. The Object Dictionary Tool (OD-Tool) has been integrated into the SSC OD configuration tool, supporting a consistent definition of the offline and online object dictionary.

2 Hardware setup

- **Hardware requirements**
  - MIMXRT1180-EVK REVC
  - FRDM-MC-LVPMSM
  - Teknic 2311P Motor
  - RJ45 Network cable
  - Mini/micro universal serial bus (USB) cable
  - Personal Computer on which the TwinCAT3 has been installed
- **Board settings**

Table 1. Jumper settings

Jumper	Setting	Jumper	Setting	Jumper	Setting
JP1	1-2	J58	2-3	J78	1-2
J4	1-2	J59	2-3	J79	1-2
J6	1-2	J63	1-2	J90	1-2
JP6	1-2	J65	1-2	J91	1-2
J9	1-2	J72	1-2	J93	1-2
J11	2-3	J73	1-2	J97	1-2
J12	1-2	J75	2-3	J98	1-2
J14	1-2	J76	1-2	J99	1-2
J57	2-3	J77	1-2	J100	1-2

For a correct connection, the servo motor application requires you to remove and solder some zero-ohm resistors. Remove and solder zero-ohm resistors according to Table 2.

Table 2. Remove and solder zero-ohm resistors

Add resistors		Remove resistors	
R747	R767	R104	R146
R749	R1049	R105	R148
R760	R1051	R107	R1089
R763		R126	R1090
		R135	

2.1 Hardware assembly

1. Connect the FRDM-MC-LVPMSM shield on top of the MIMXRT1180-EVK board.
   Note: *There is only one possible option.*
2. Connect the 3-phase motor wires to the screw terminals (J7) on the Freedom permanent magnet synchronous motor (PMSM) power stage.
3. Plug the 24 V direct current (DC) power supply to the DC power connector on the Freedom PMSM power stage.

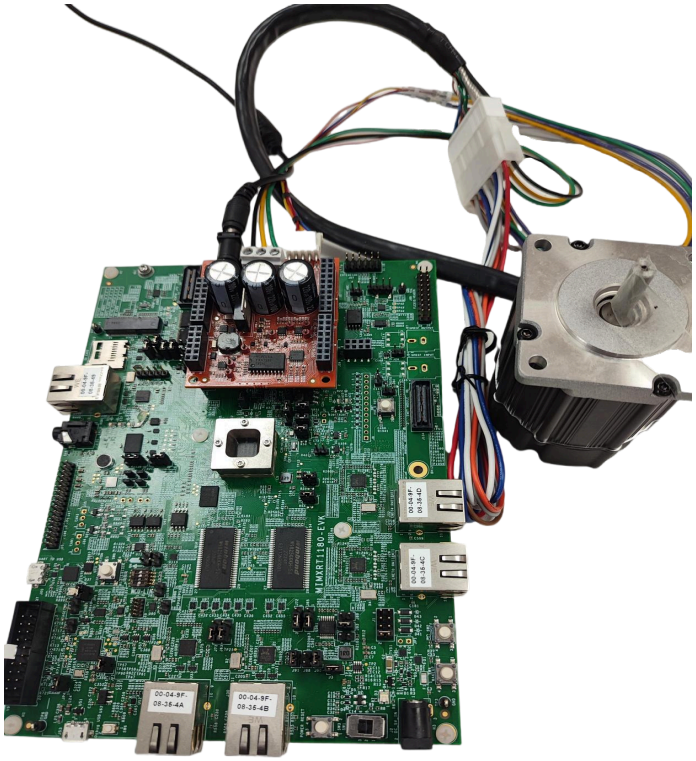


Figure 1. Hardware assembly

2.2 Connection

1. Connect EtherCAT Port 0 (J28 RJ45 Interface) on the MIMXRT1180-EVK board with TwinCAT3 via the Ethernet cable.
2. Connect a USB cable between the host PC and the OpenSDA USB port on the MIMXRT1180-EVK board.
3. Open a serial terminal with the following settings.
   - 115,200 baud rate
   - 8 data bits
   - No parity
   - One stop bit
   - No flow control

3 MCUXpresso SDK download

To configure and download the MCUXpresso SDK for the servo motor example building, perform the following steps.

1. Click the link https://mcuxpresso.nxp.com/en/welcome to go to the MCUXpresso main page.
2. Click the **Select Development Board** button.

Figure 2. Select Development Board

3. Type 1180 in the **Search for Hardware** field.
4. Select *MIMXRT1180-EVK* from the drop-down list. See, Figure 4.

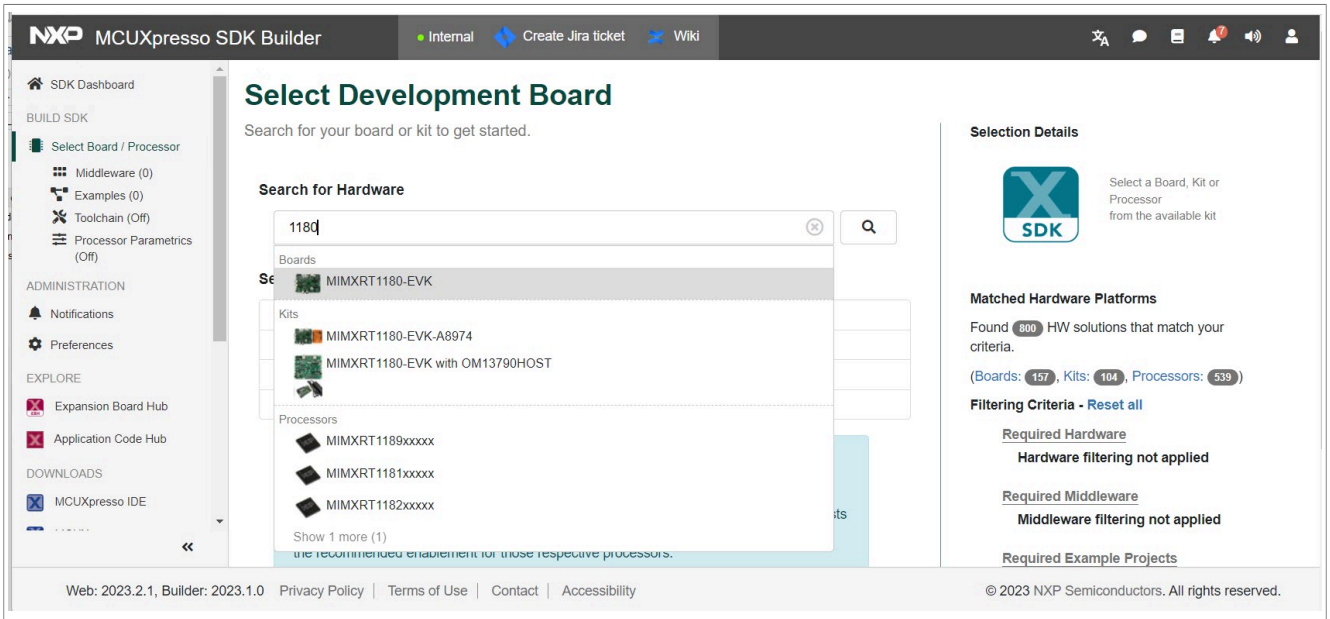


Figure 3. Select a board

5. Select the SDK version from which the SDK will be generated.
6. Click the **Build MCUXpresso SDK** button.

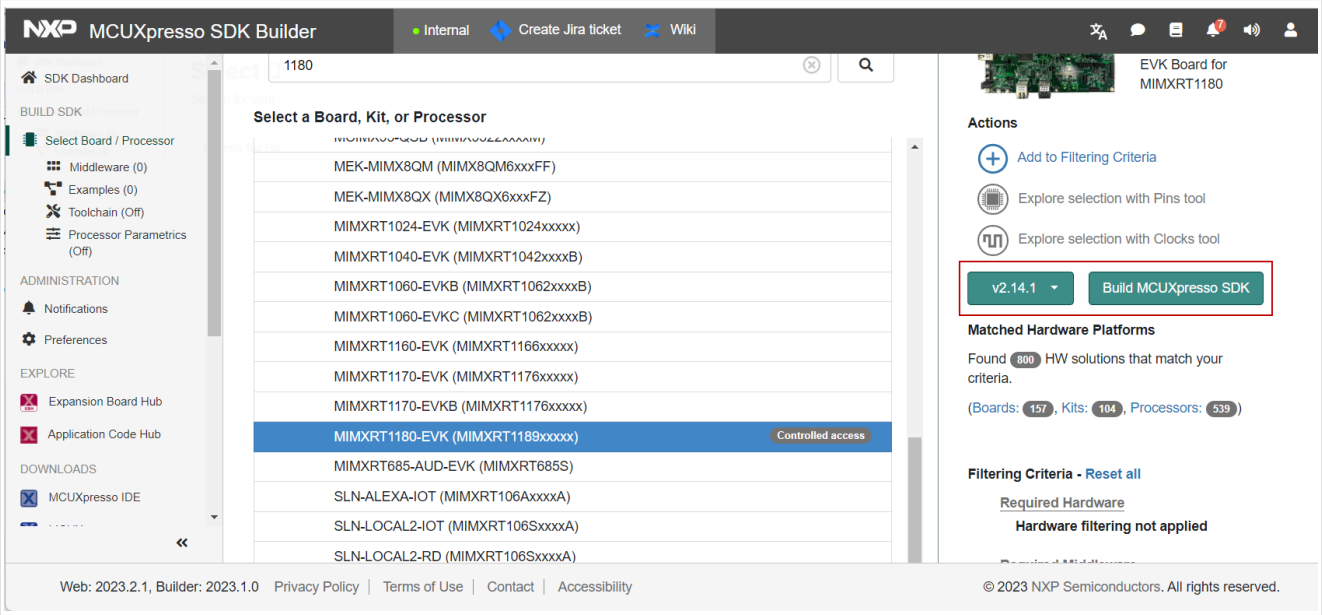


Figure 4. Select the SDK version and build MCUXpresso SDK

7. Select the host operating system.

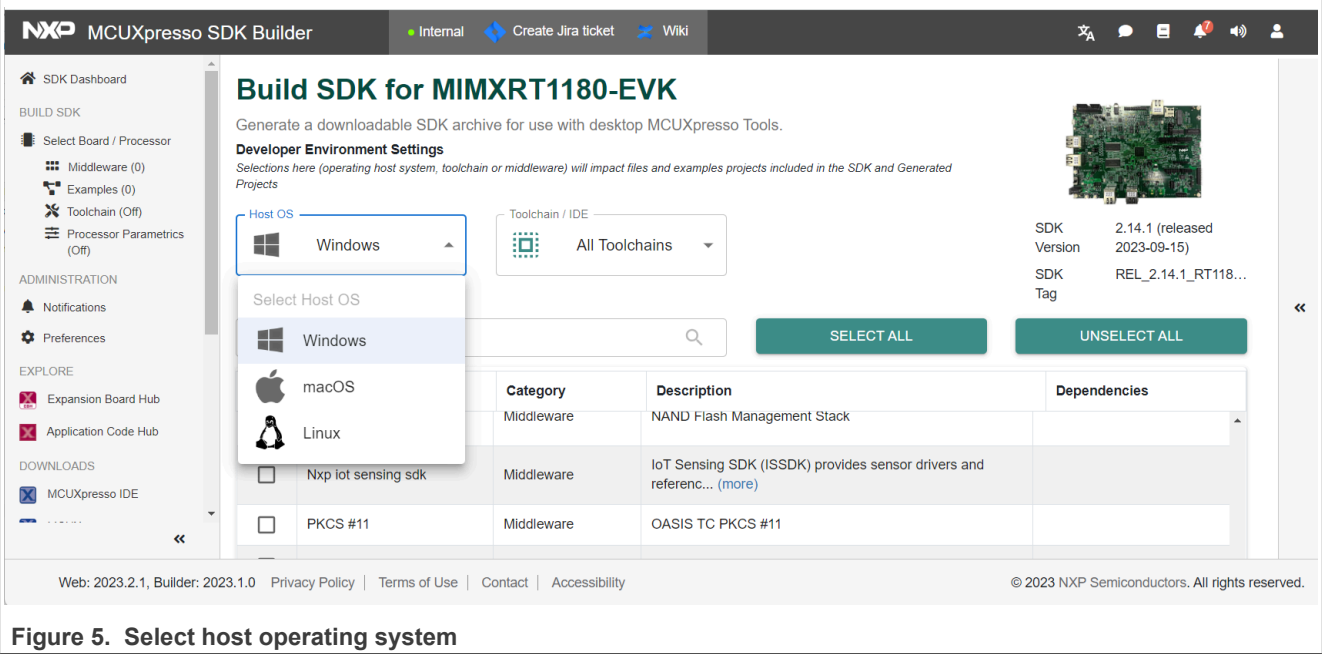


Figure 5. Select host operating system

8. Select the Toolchain/IDE.

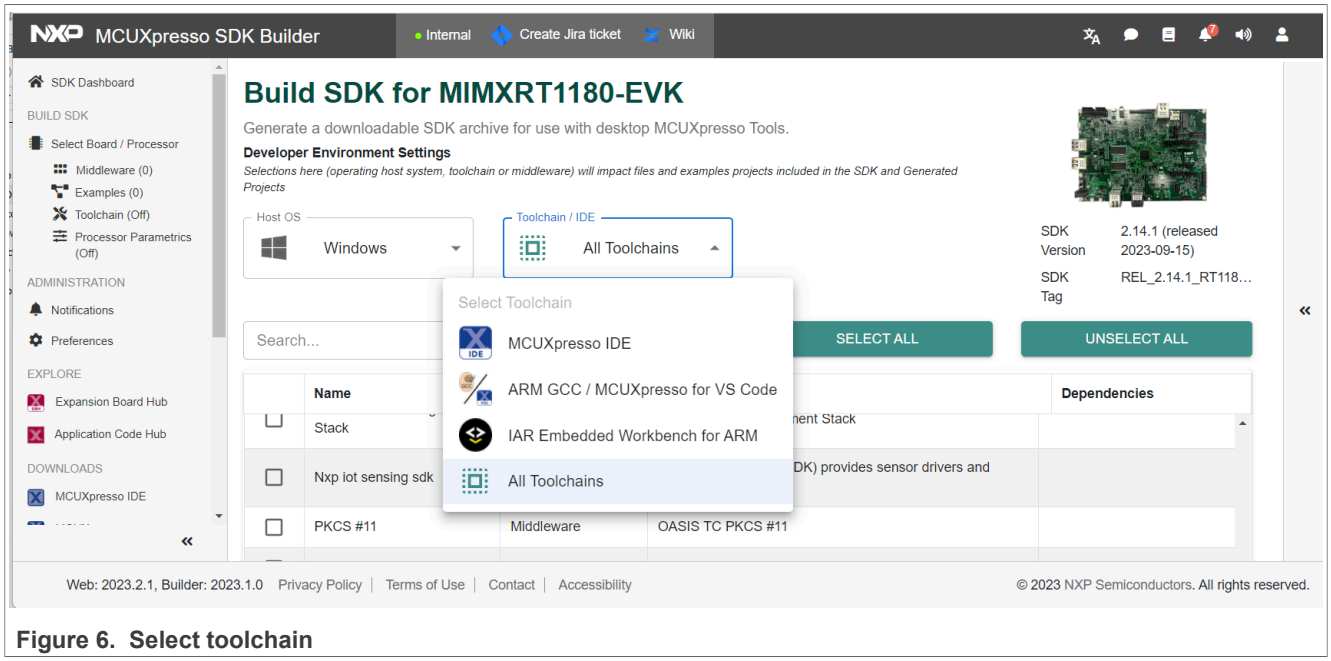


Figure 6. Select toolchain

9. Select the **Motor Control Software** checkbox.

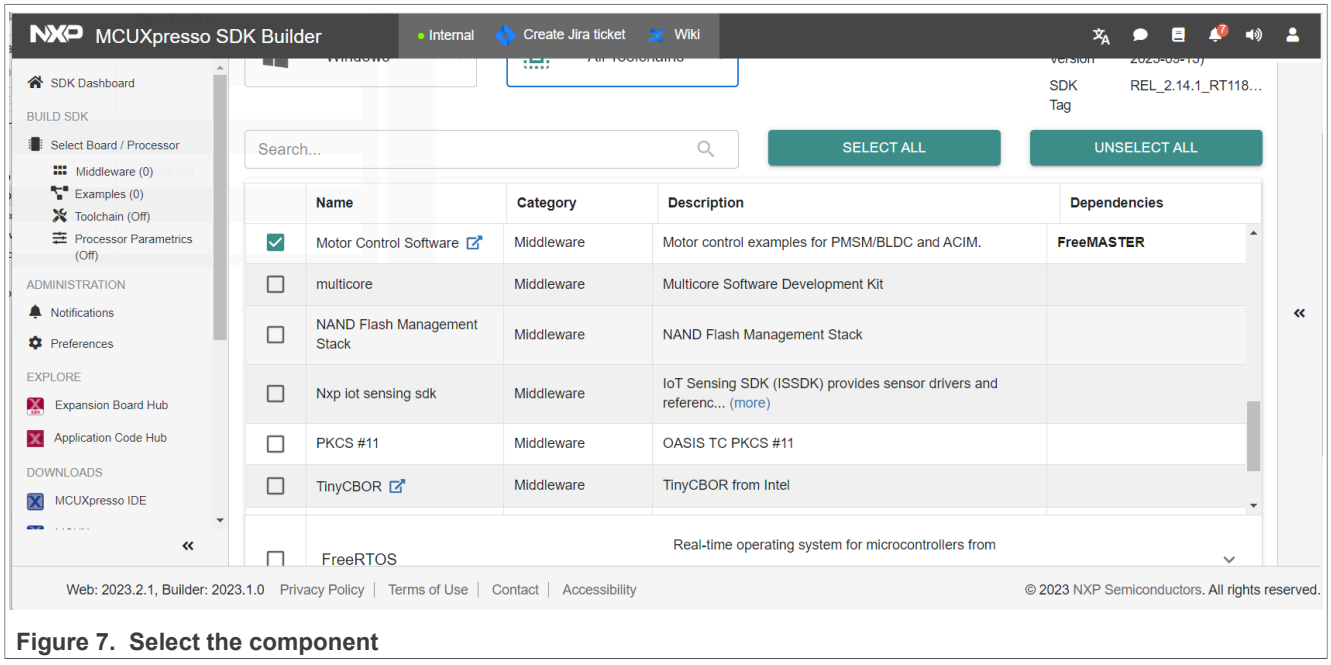


Figure 7. Select the component

10. To download the SDK, click the **Download SDK** button.

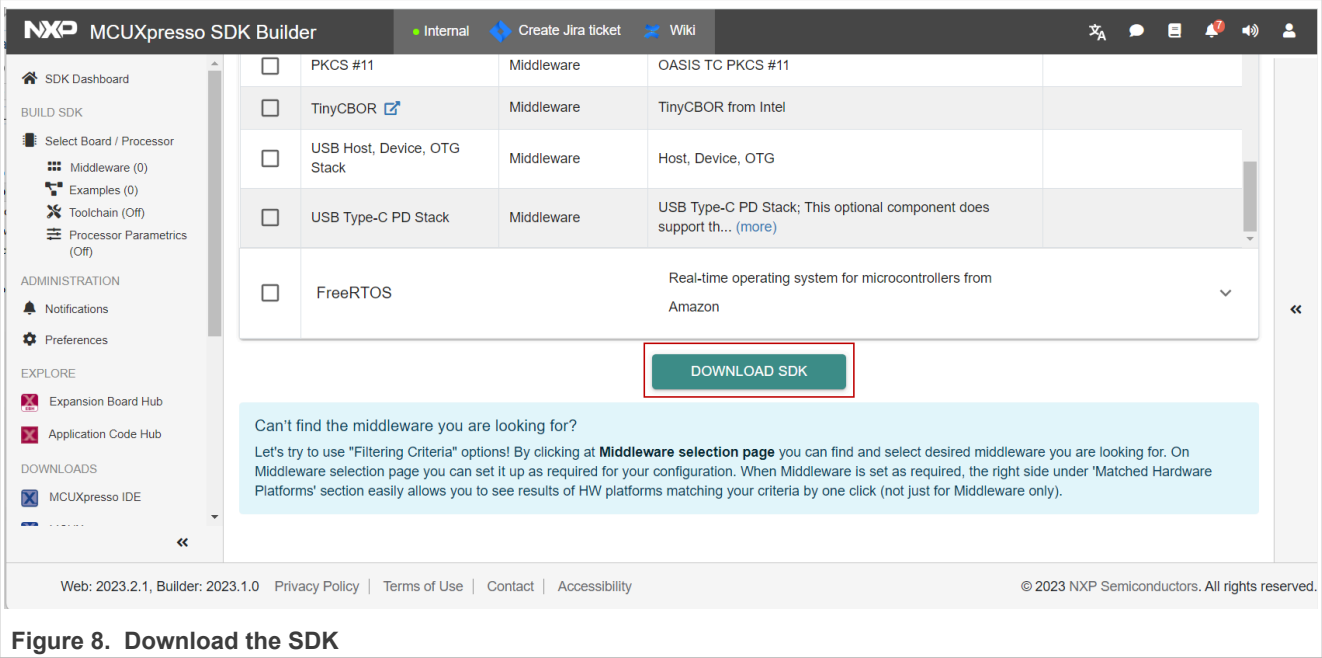


Figure 8. Download the SDK

You can now access, download, and share your requested SDK builds from the **MCUXpresso Dashboard**.

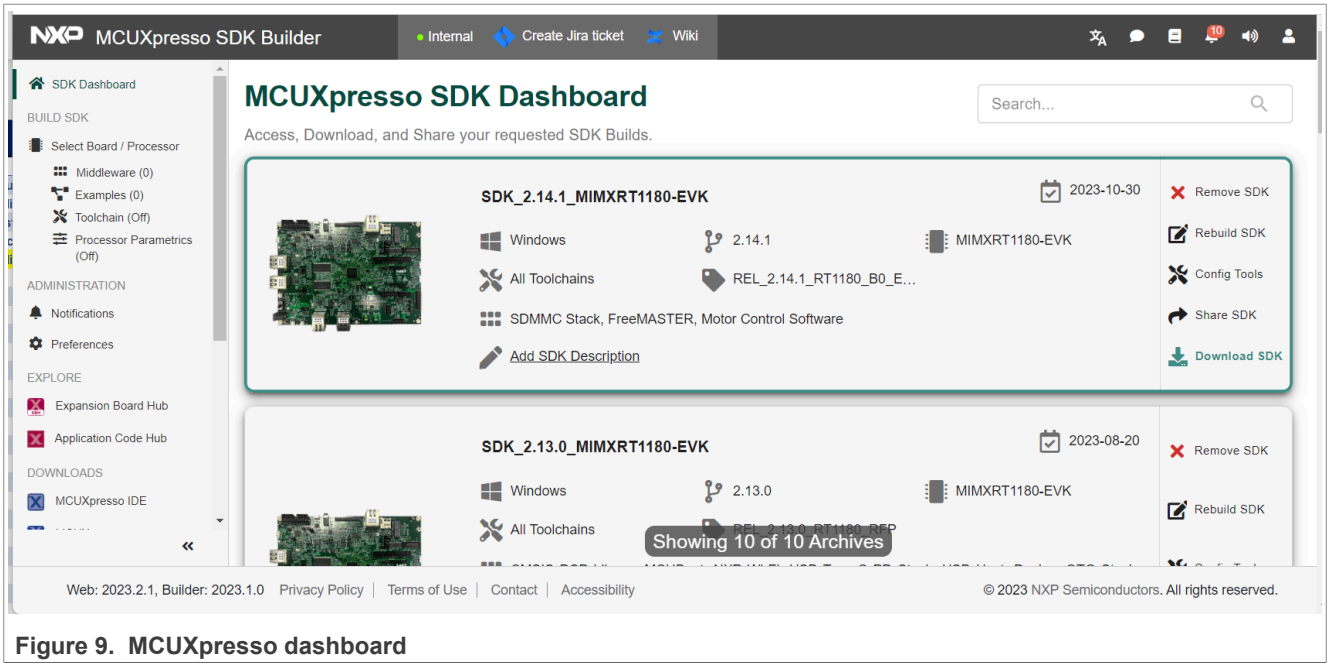


Figure 9. MCUXpresso dashboard

11. Click the **Download SDK** button.

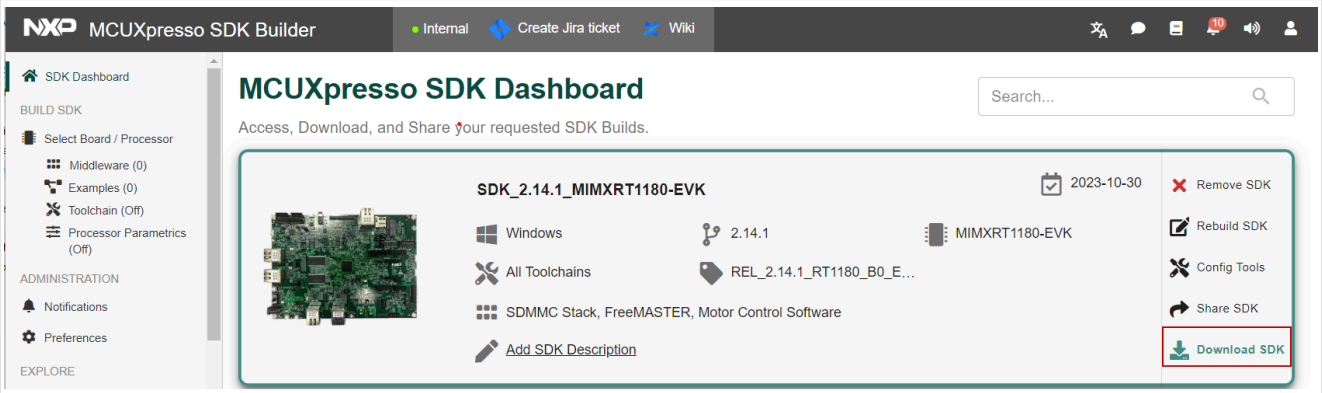


Figure 10. Download SDk

12. Click the **Download SDK archive including documentation** link.

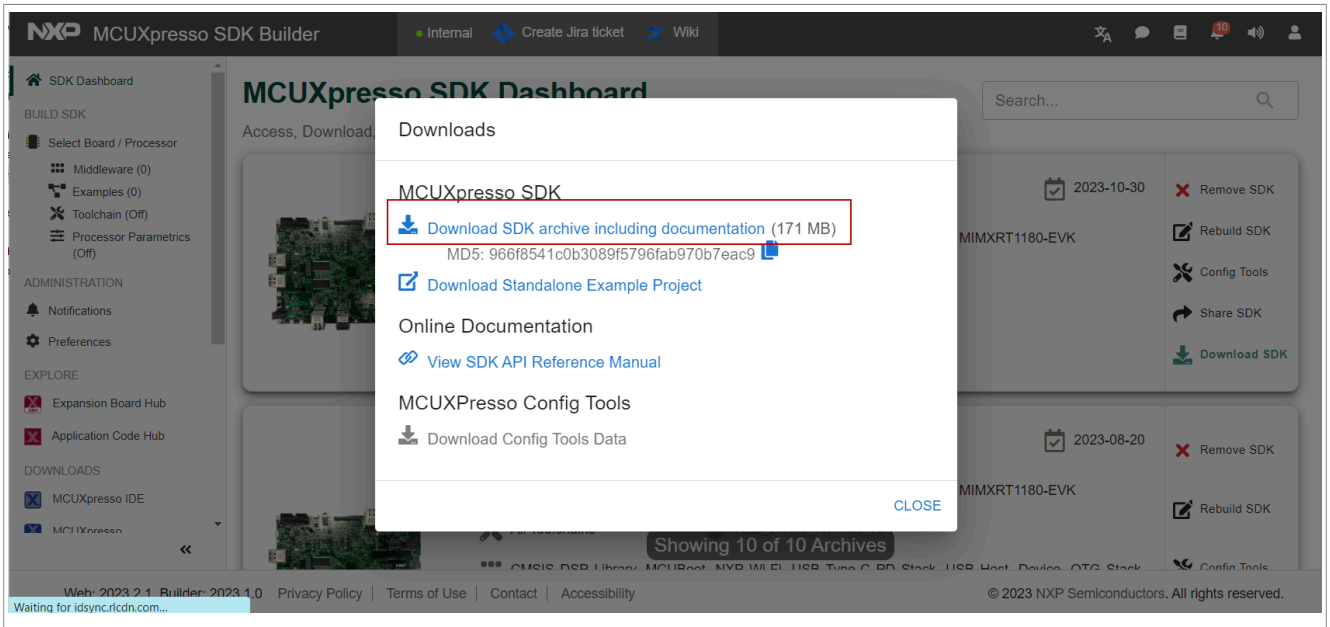


Figure 11. Download the SDK archive

13. Read the end user license agreement (EULA).
14. Click the **Agree** link.

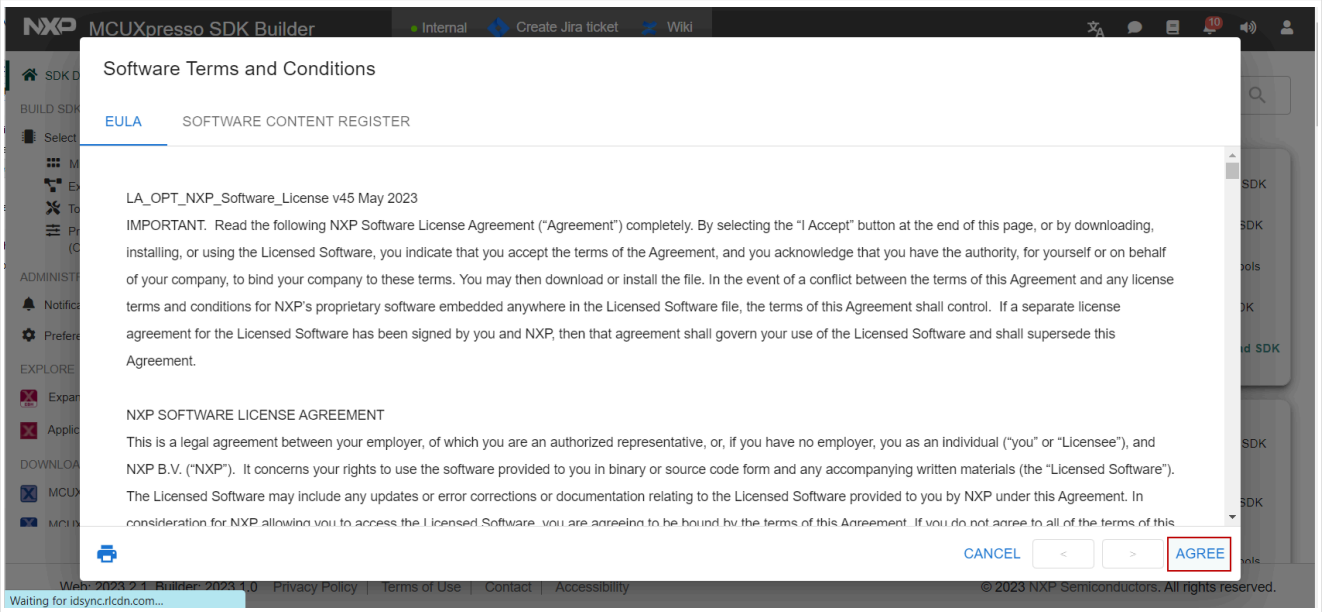


Figure 12. Accept the EULA

Wait until the build finishes and the download the SDK zip file.

4 Generate SSC code for servo motor

Due to the license issue, the MCUXpresso SDK does not include the EtherCAT Slave Stack Code (SSC). The SSC Tool must generate the slave stack code first. This section lists the steps to download the SSC tool and generate the slave stack code for the servo motor example via the SSC tool.

Note: *You must have a Backoff account to download the EtherCAT SSC tool.*

1. Download and install the SSC tool.
   a. Click the link https://www.ethercat.org/en/downloads.html.
   b. Type *SSC* in the **Text Filter** field.
   c. Select the *EtherCAT Slave Stack Code (SSC)* link under the **Description** column.
   Note: *To proceed with the download, you must provide your account credentials, if you are not already logged in.*

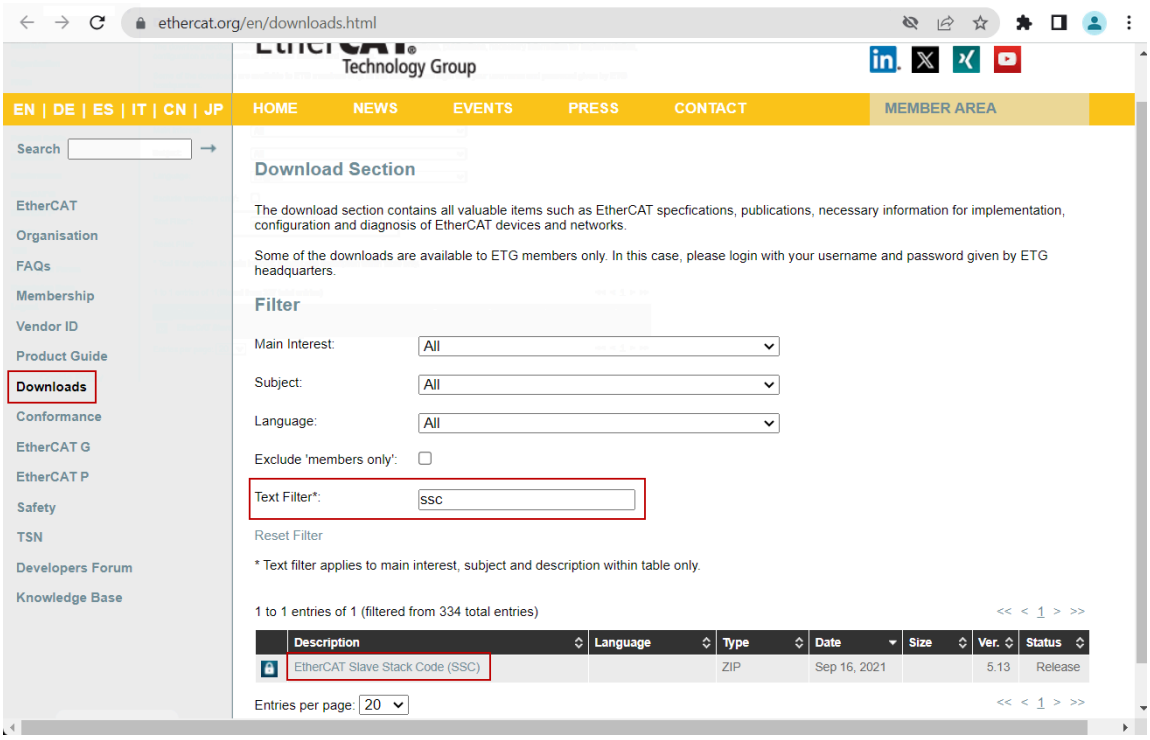
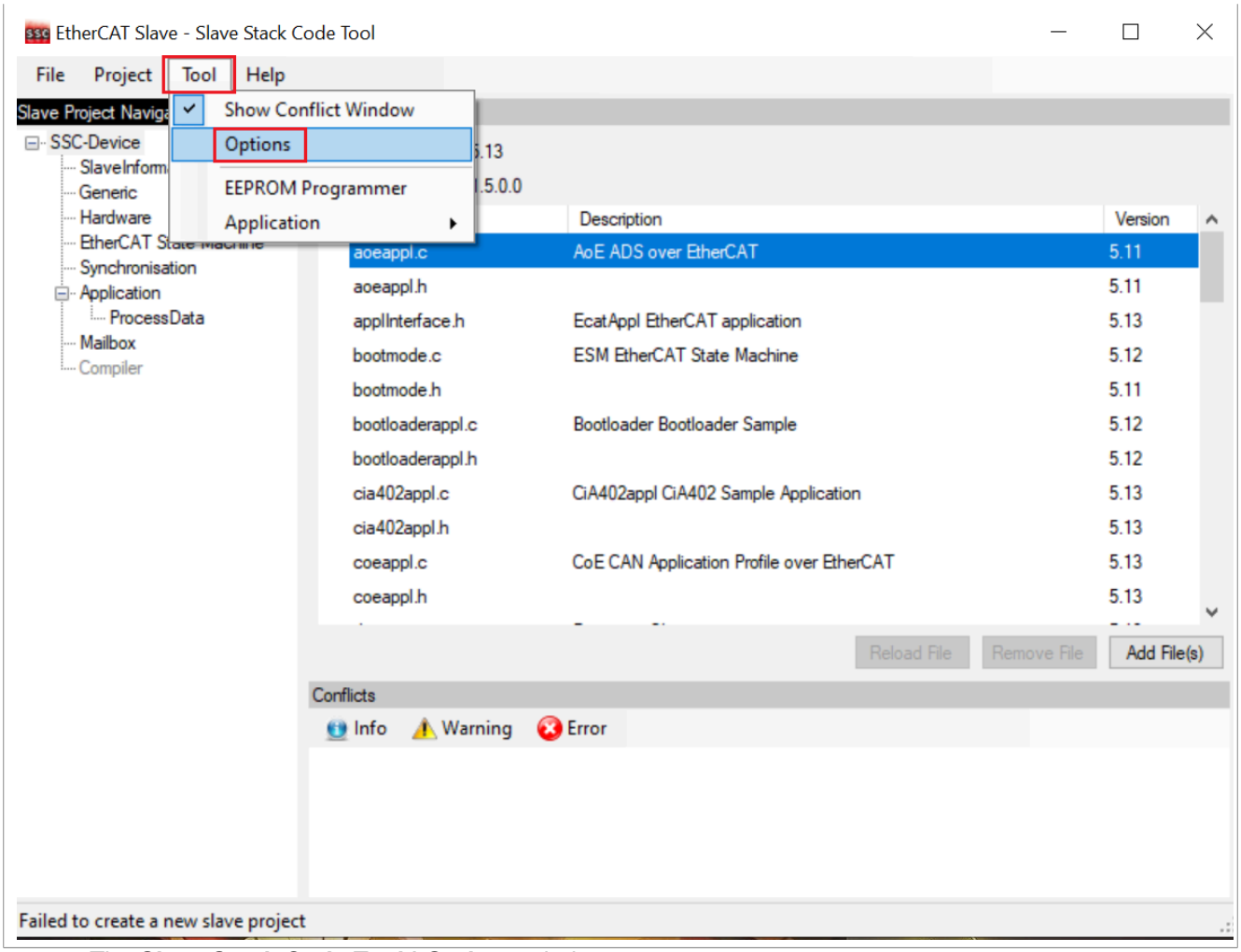


Figure 13. Search and select EtherCAT SSC

d. Install the downloaded SSC tool.
e. Launch the SSC tool.

2. Import the configuration file for the servo motor example.
   a. Select **Tool > Options**.

Figure 14. Select tool options



The **Slave Stack Code Tool | Options** window appears.

b. Click the **Configurations** tab.

c. To open the **Import Configuration file(s)** dialog box, click the + icon.

d. Navigate to the *\boards\evkmimxrt1180\ecat_examples\servo_motor\cm7\SSC* folder.

e. Select the *servo_motor.xml* file.

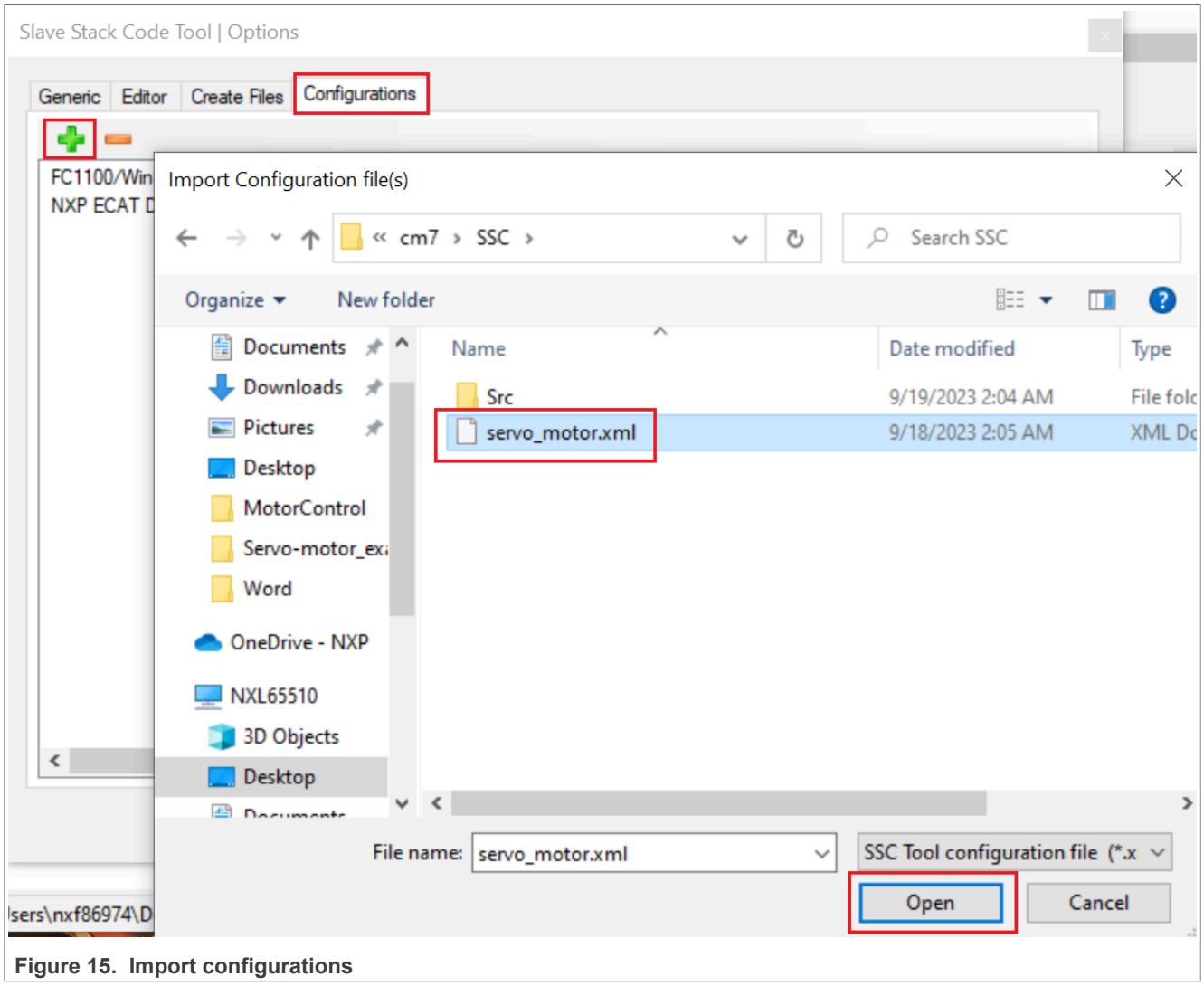


Figure 15. Import configurations

f. Click the **Open** button.
The configuration is imported.

3. Create a new project with the imported configuration.
   a. Select **File > New**.

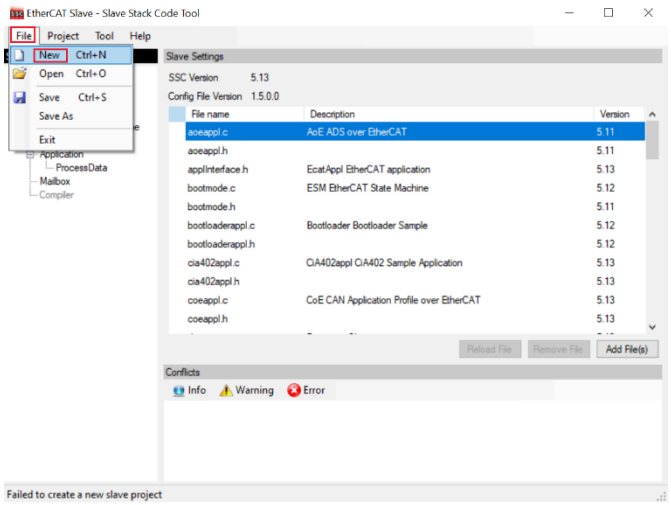


Figure 16. Create new project

The **New Project** dialog box appears.

b. Select the **Custom** option.

c. Select *NXP ECAT Servo Motor modular <NXP>* from the drop-down list.

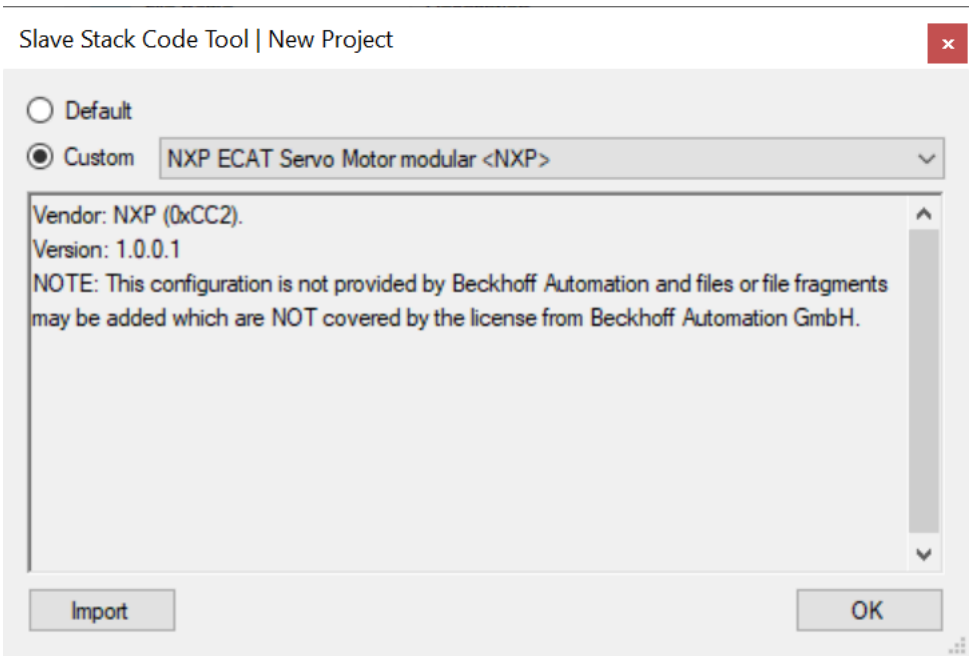


Figure 17. Select the custom project

4. Save this new project as *ECAT-SERVO-MOTOR.esp* to any folder.

a. Select **File > Save**.

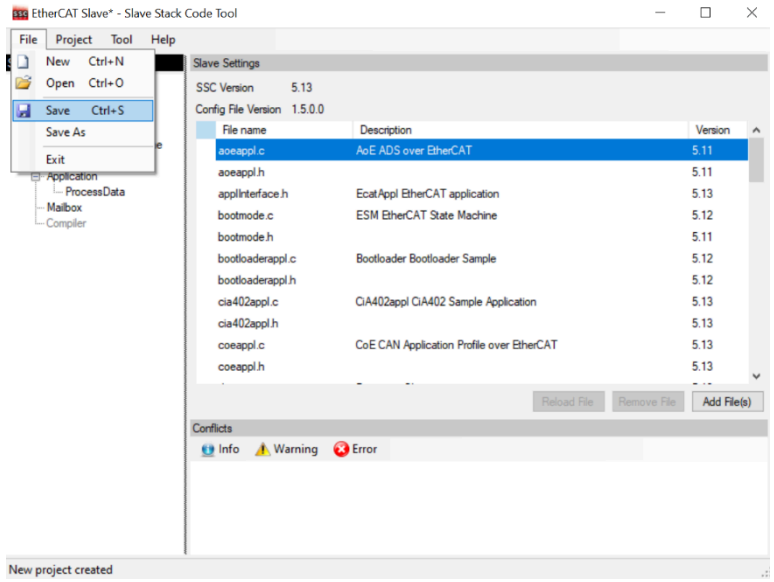


Figure 18. Select File > Save

b. Save the file with an appropriate name. For example, *ECAT-SERVO-MOTOR.esp*.

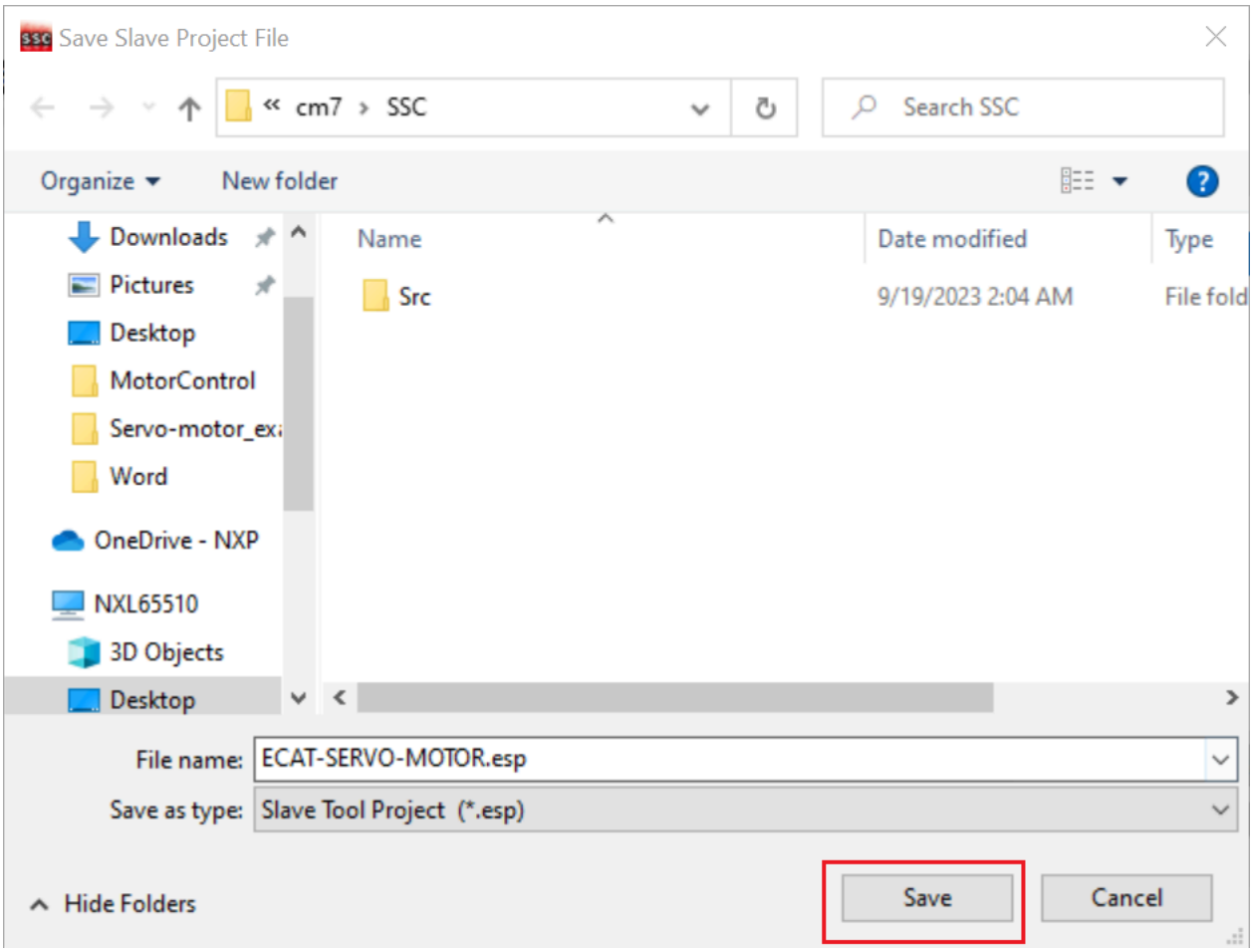


Figure 19. Save the file with the name ECAT-IO.esp

5. Create the slave stack code for the servo motor example.
   a. Click **Project > Create new Slave Files**.

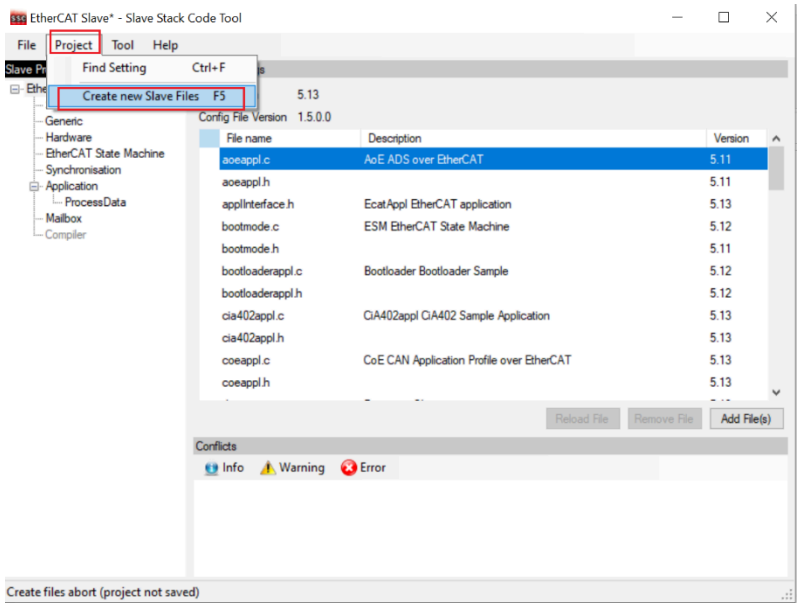


Figure 20. Create new files

The **Create new Slave Files** dialog box appears.

b. To create the code, select **Start**.
By default, the code is located in the *Src* subfolder under the folder where the project is saved.

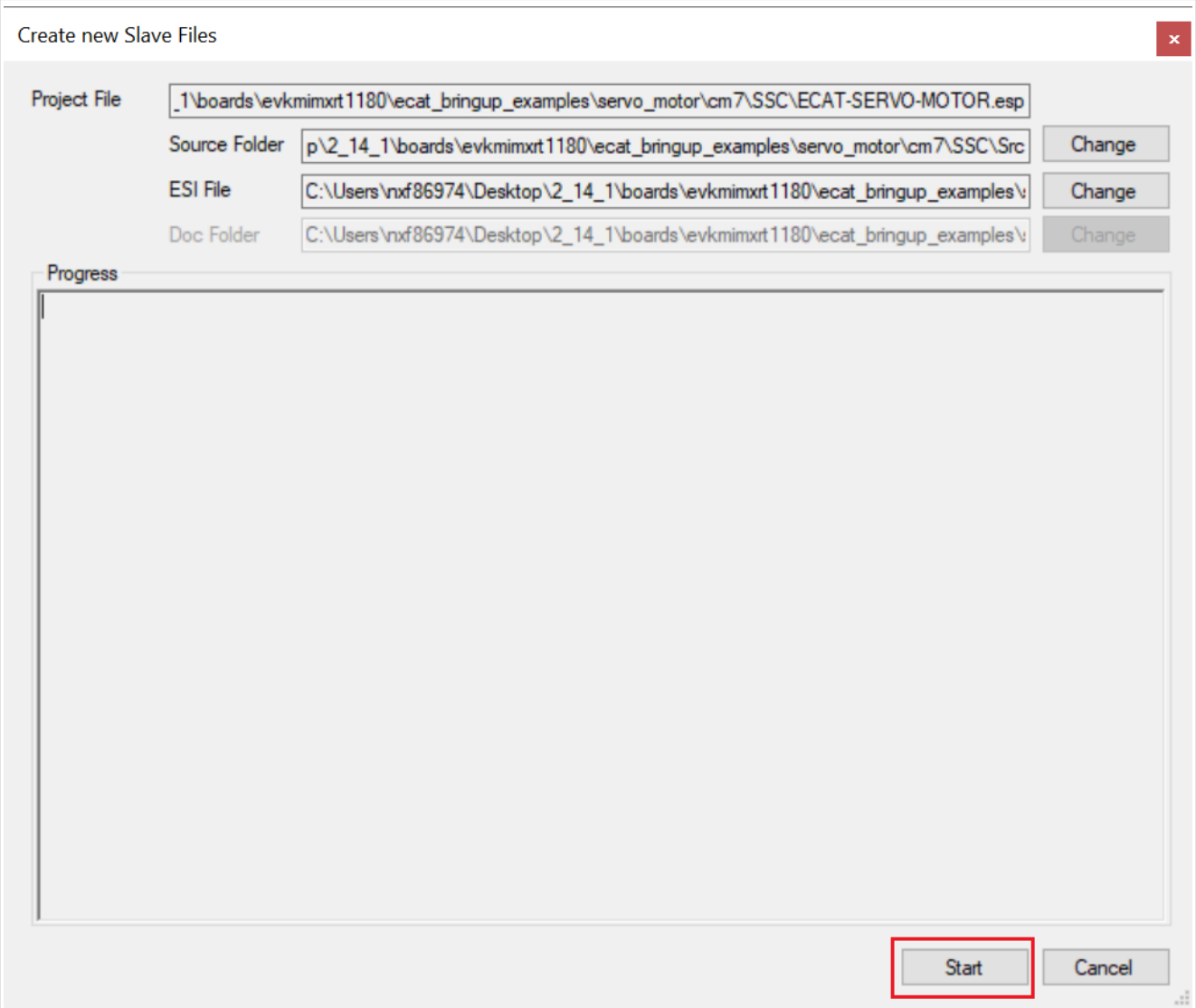
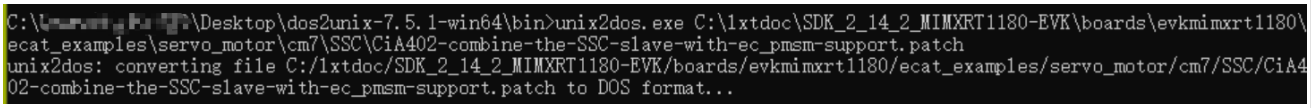


Figure 21. Start to create the code

6. Copy the *Src* subfolder to *boards\evkmimxrt1180\ecat_examples\ servo_motor\cm7\SSC\Src* under the SDK source code.
7. **Apply patch**.
   Change to *.\boards\evkmimxrt1180\ecat_examples\servo_motor\cm7\SSC* path. Install patch tool.
   - To apply the patch on **Linux**:
     a. Download dos2unix command `apt-get install dos2unix`.
     b. Transfer SSC source code format `dos2unix Src/*`.
     c. Apply patch by using the command `patch -d Src < CiA402-combine-the-SSC-slave-with-ec_pmsm-support.patch`.
   - To apply the patch on **Windows**:
     a. Download the `patch.exe` and `Unix2Dos.exe` tools.
        – Download Windows patch utility from http://gnuwin32.sourceforge.net/downlinks/patch-bin-zip.php.
        – Download `Dos2Unix/Unix2Dos-Text` file format converters from https://sourceforge.net/projects/dos2unix/.

b. Transfer the patch format. `$(Dos2Unix/Unix2Dos-DIR)/bin/unix2dos.exe …\boards\evkmimxrt1180\ecat_examples\servo_motor\cm7\SSC\CiA402-combine-the-SSC-slave-with-ec_pmsm-support.patch`.

Figure 22. Transfer patch



c. Apply patch by using the command `./patch.exe -i …\boards\evkmimxrt1180\ecat_examples\servo_motor\cm7\SSC\CiA402-combine-the-SSC-slave-with-ec_pmsm-support.patch -d …\boards\evkmimxrt1180\ecat_examples\servo_motor\cm7\SSC\Src\.`

Figure 23. Apply patch

Note: *Patch and Unix2Dos are command line tools. To check their functions through command like use `patch --help/unix2cos -help`.*

5 Compile and download the image to RT1180

- Toolchain supported
  - GCC Arm Embedded 10.3.1

Note: *For detailed information on the image download and program, see section 4 of the Getting Started with MCUXpresso SDK for MIMXRT1180-EVK document in the SDK package.*

6 TwinCAT project setup

This section lists the steps to set up the TwinCAT project.

6.1 Create a new project

1. Import the ESI file of the servo motor example before TwinCAT starts.
   Copy the ESI file '`ECAT-SERVO-MOTOR.xml`' generated by SSC tool under the SSC project saving directory to `<TwinCAT_installation_folder>/<Version>/Config/io/EtherCAT/`.
2. Select **File > New > Project**.

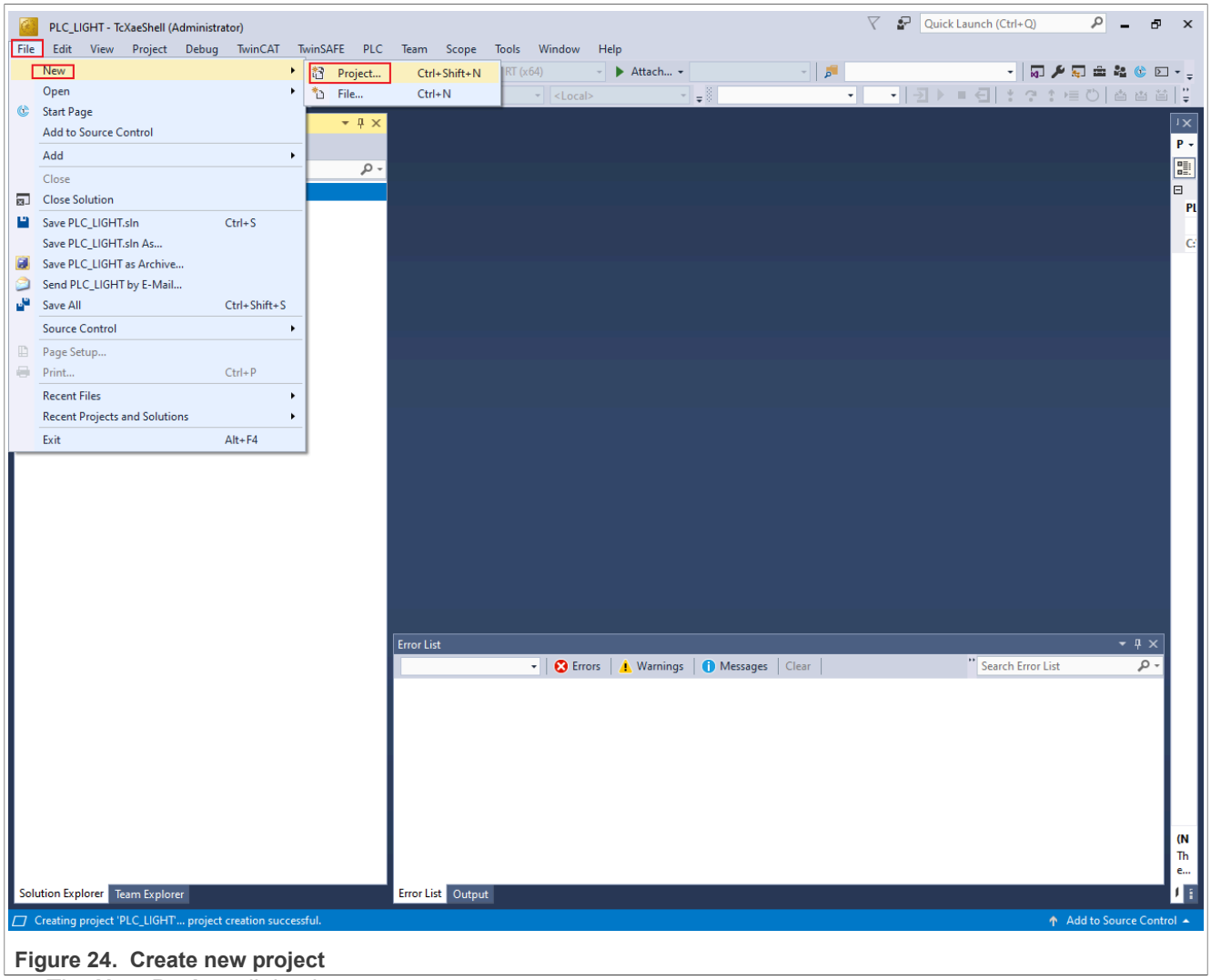


Figure 24. Create new project

The **New Project** dialog box appears.

3. Select **TwinCAT Projects**.
4. Click **OK**.

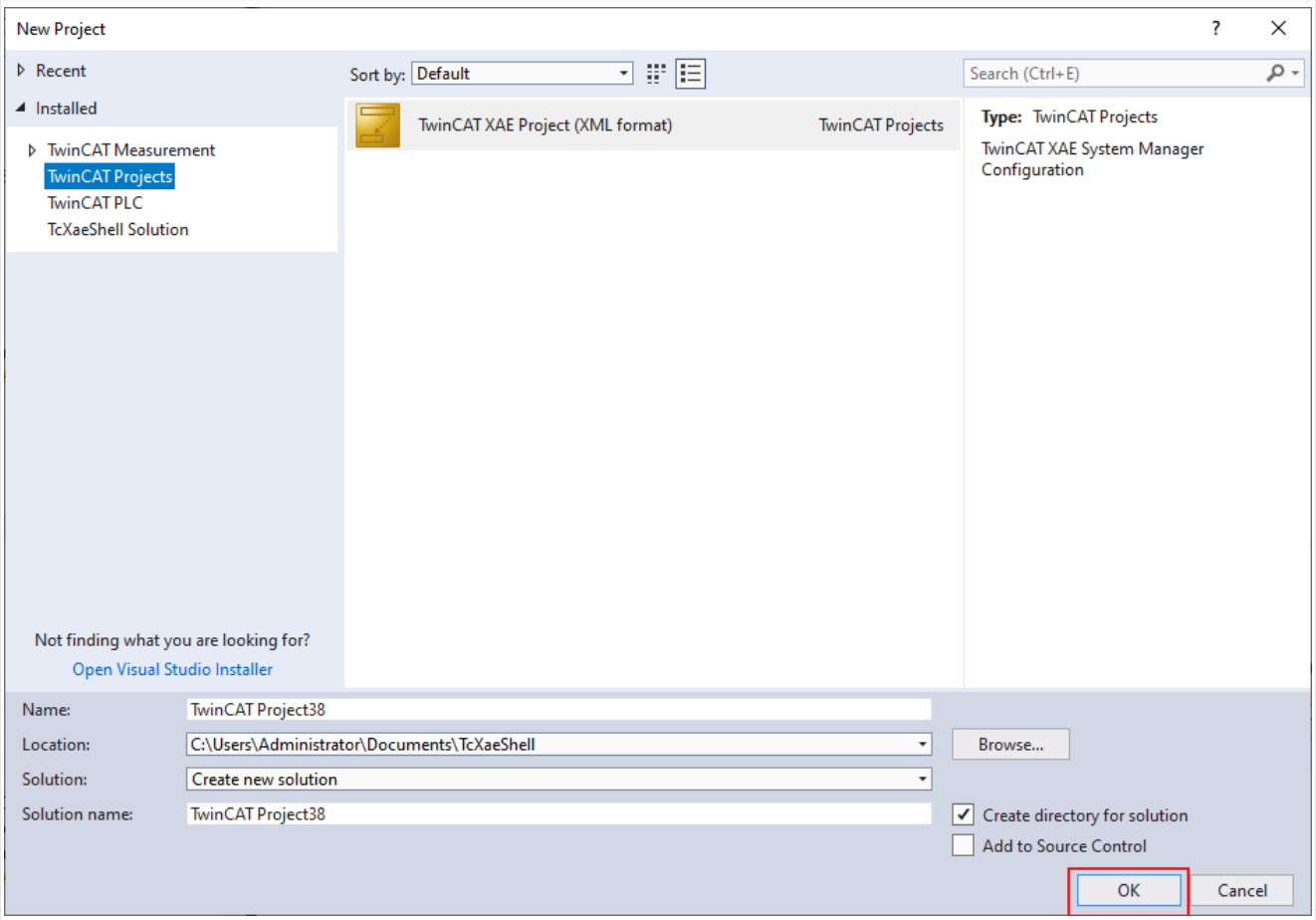


Figure 25. Select project

6.2 Scan the for slave devices

1. In the **Solution Explorer** view, expand **I/O**.
2. Right-click on **Device** and select **Scan**.

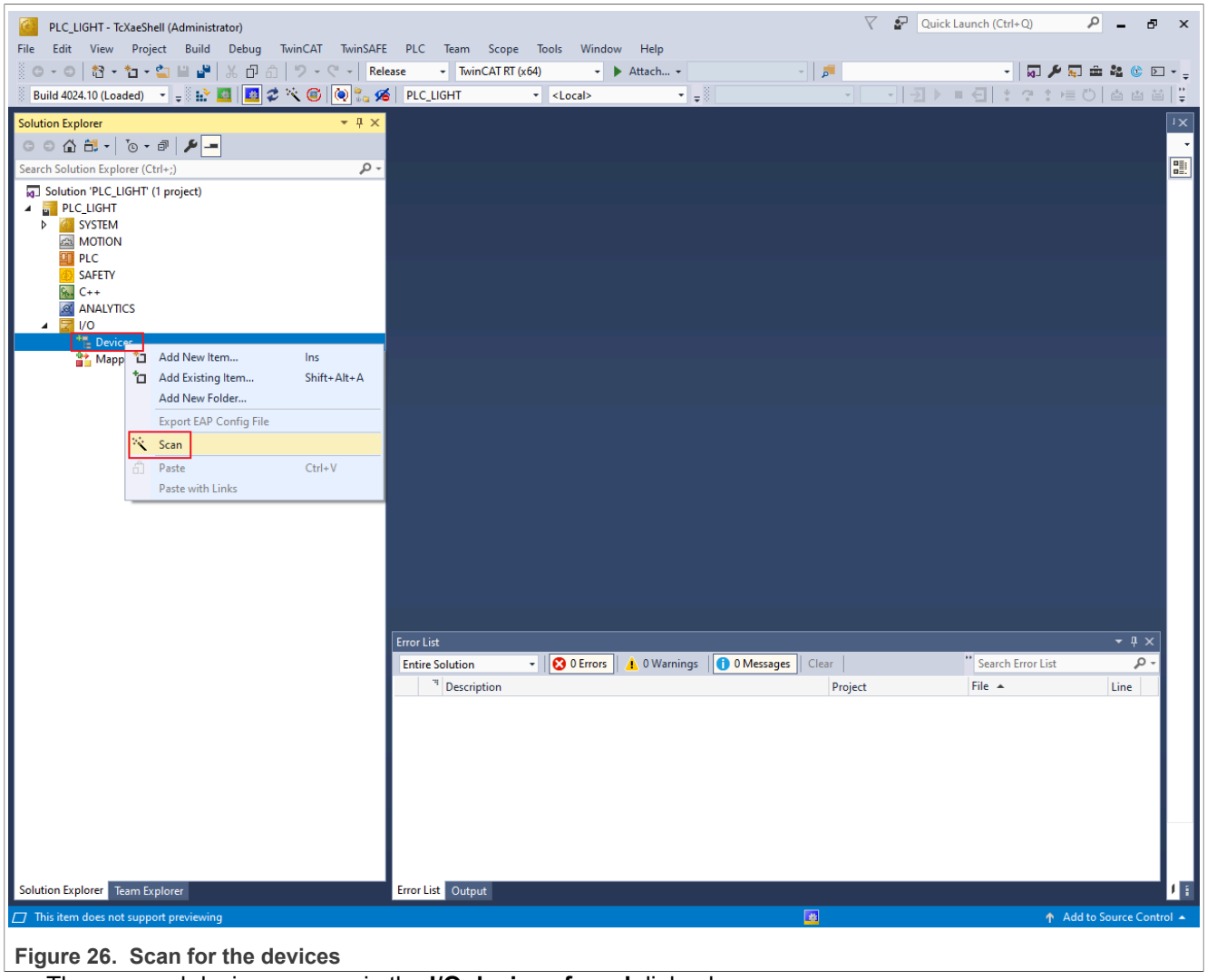


Figure 26. Scan for the devices

The scanned devices appear in the **I/O devices found** dialog box.

3. Select the network interface connected with the MIMXRT1180-EVK board.
4. Click **OK**.

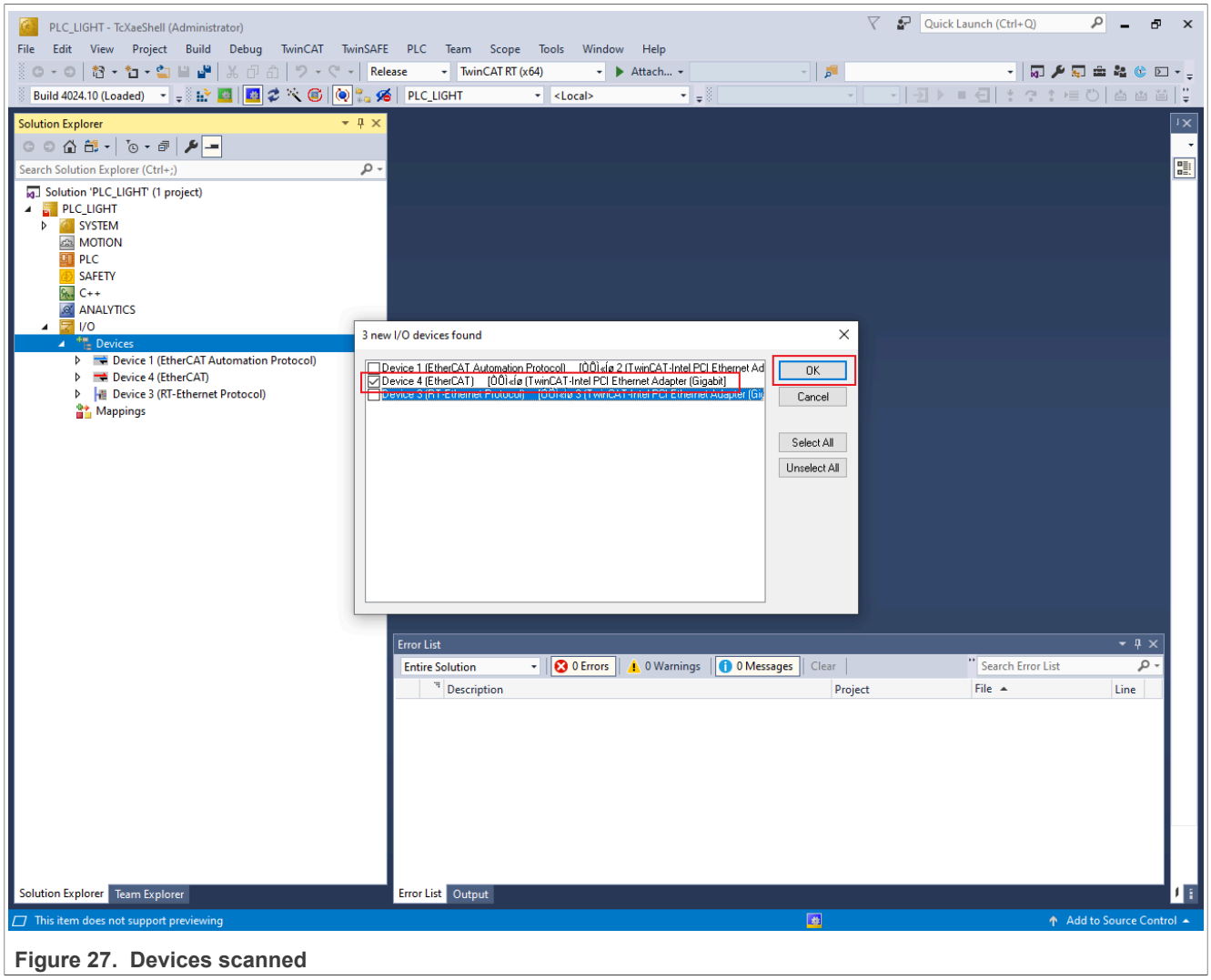


Figure 27. Devices scanned

6.3 Update the ESI file to E2PROM

Note: *The E2PROM must be updated if the servo motor example is set up first time on the MIMXRT1180-EVK.*

1. Under **Device4**, double-click Box 1 (ECAT-SERVO-MOTOR).The **TwinCAT Project** dialog box appears.
2. Click the **EtherCAT** tab.
3. Click the **Advanced Settings** button.
   The **Advanced Settings** dialog box appears.
4. From the left pane of the **Advanced Settings** dialog box, select **ESC Access > Smart View**.
5. Click the **Write E2PROM** button.
6. From the available EEPROM list, **select NXP > ECAT > ECAT-SERVO-MOTOR**.

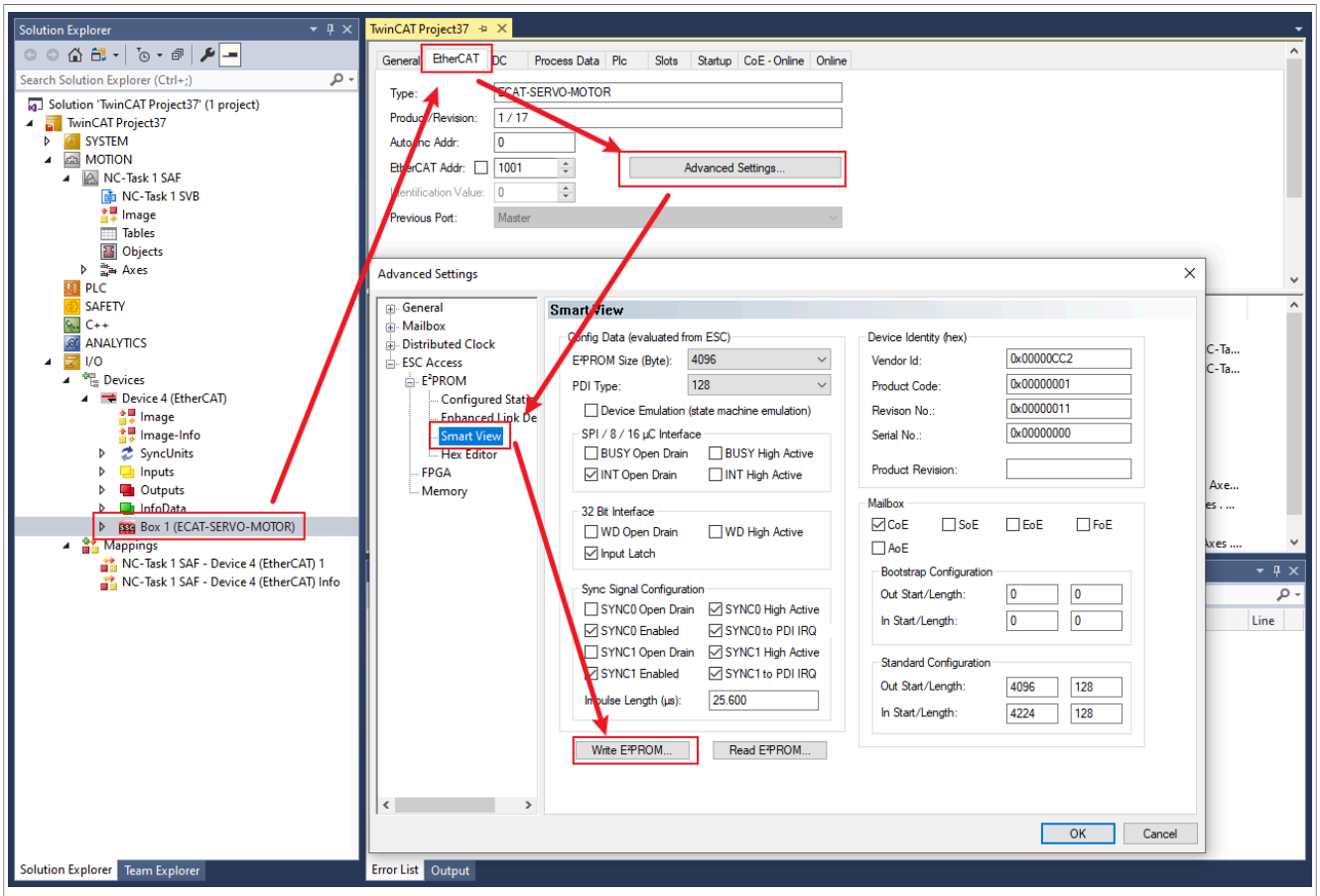


Figure 28. Update the ESI file to E2PROM

7. Click **OK**.

   Note: *Delete Device4, rescan, and add Device4 after Write E2PROM success.*

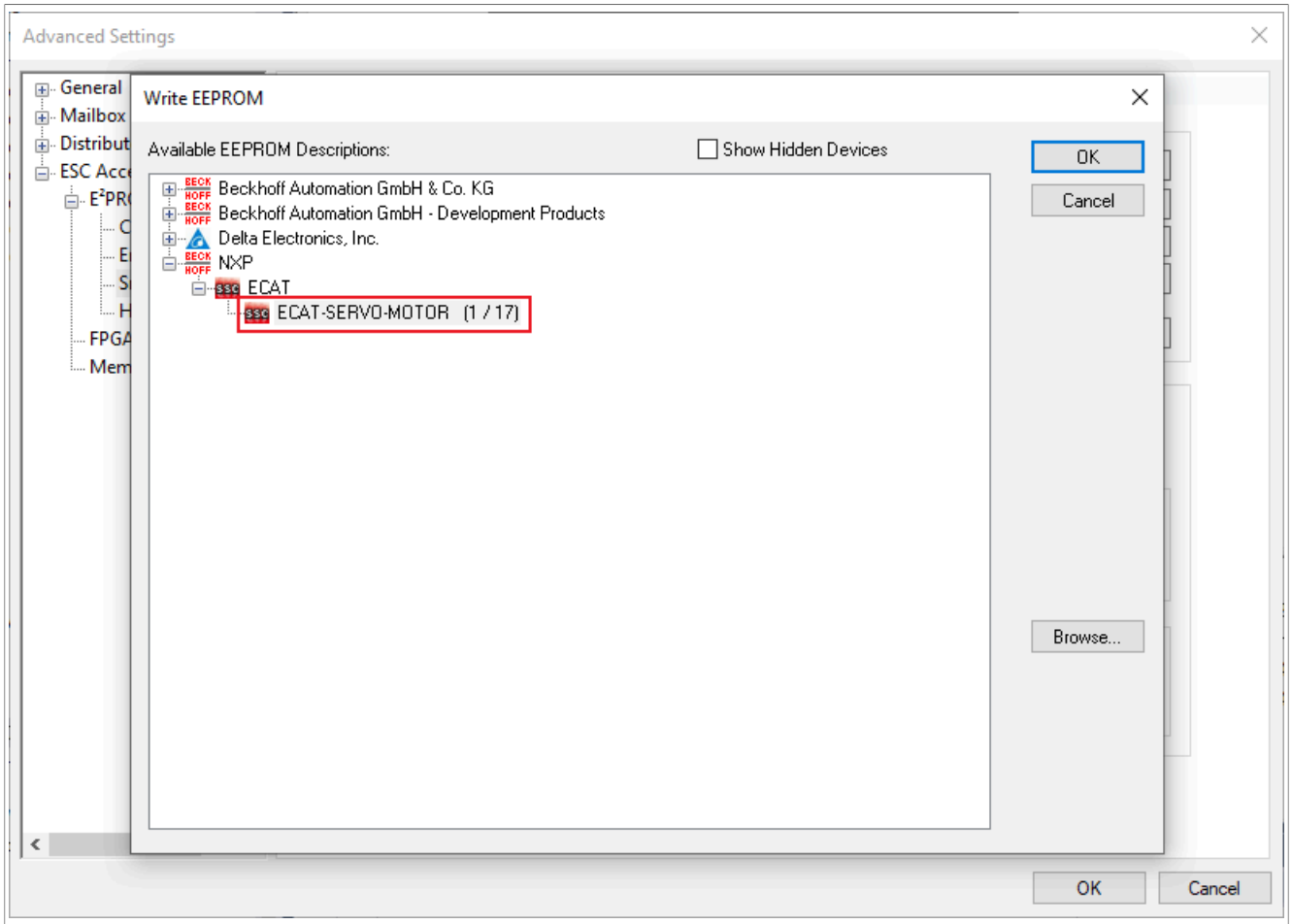


Figure 29. Write EEPROM

6.4 Configure the slave

1. Click **Box 1(ECAT-SERVO-MOTOR)**.
2. Click the **DC** tab.
3. From the **Operation Mode** field, select the **DC-Synchron** option.

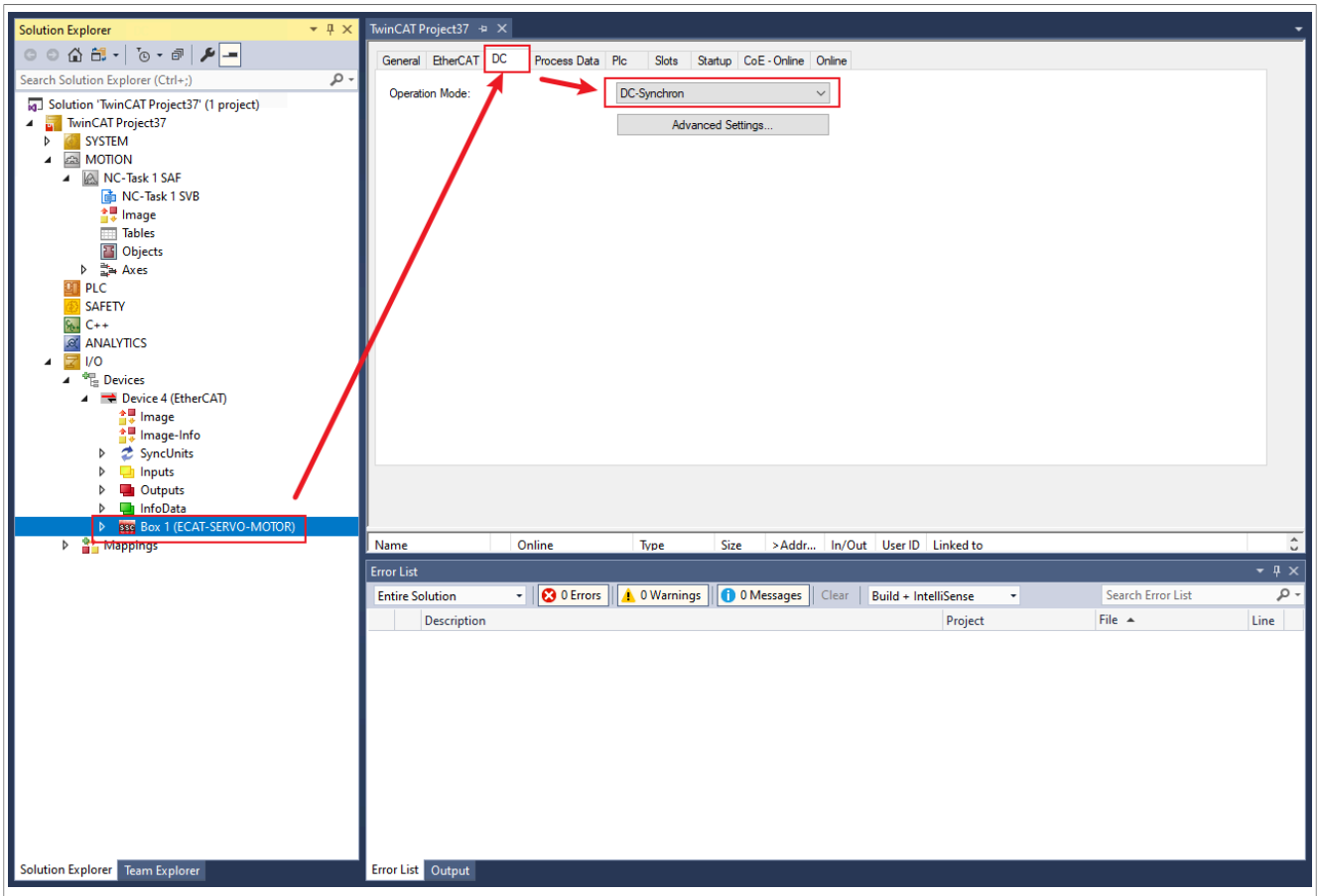


Figure 30. Select the operation mode

6.5 Select the csp mode

1. Click **Box 1(ECAT-SERVO-MOTOR)**.
2. Click the **Slots** tab.
3. Select *Axis 0'* > *'x'*.
4. Click *Axis 0'* > *'csp - axis'* > '<' to select the csp mode.

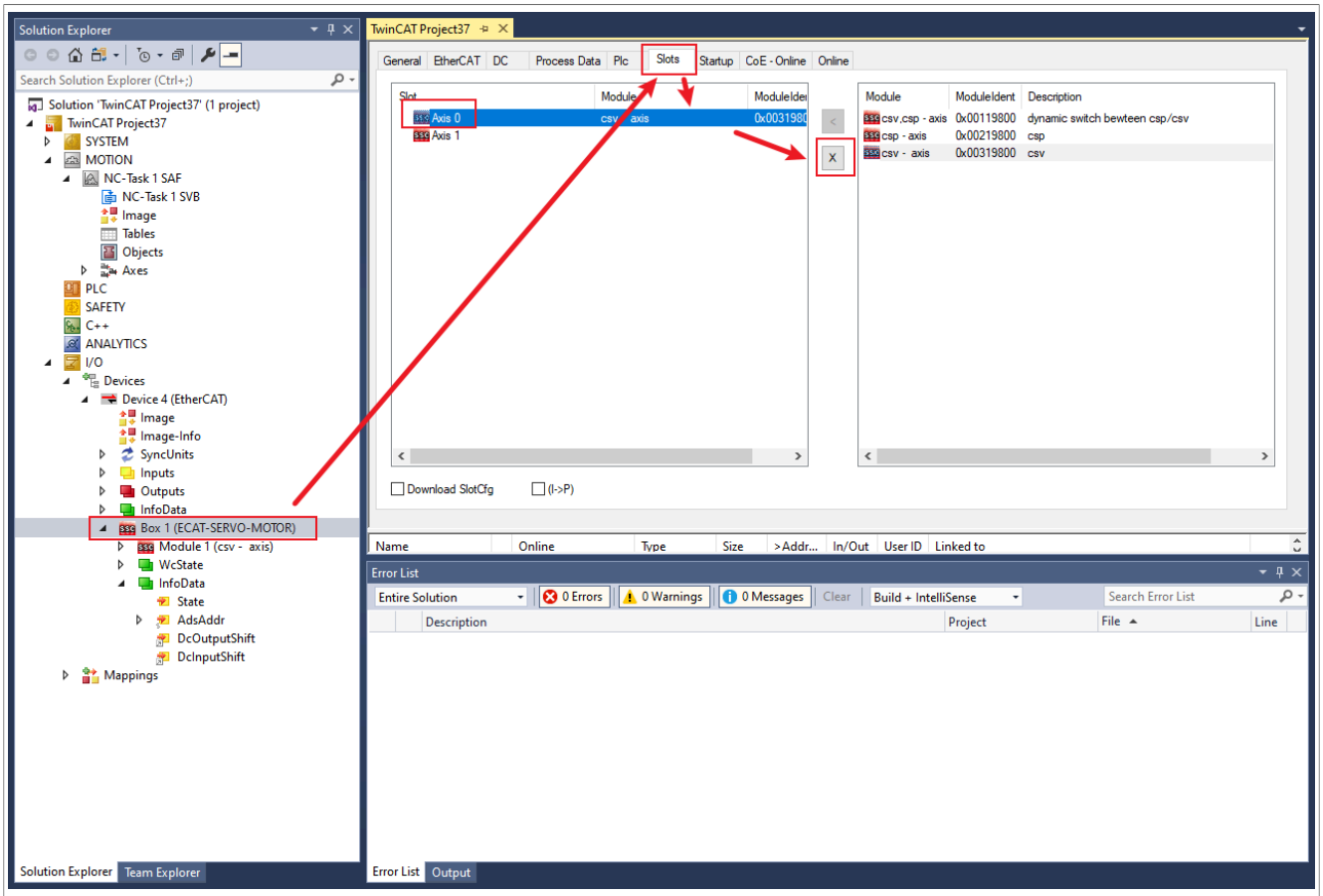


Figure 31. Select slots

The results are shown in Figure 32.

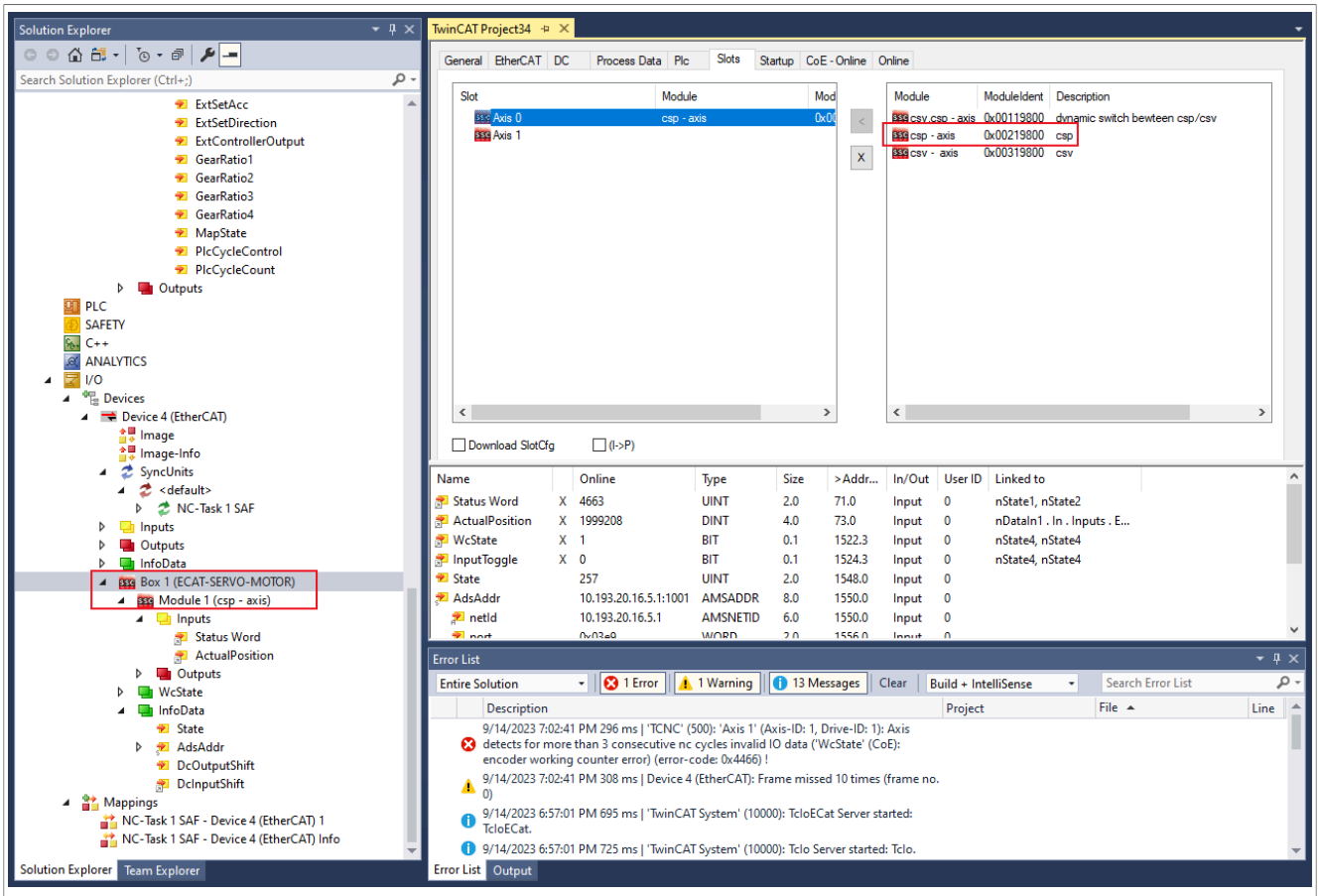


Figure 32. Select the csp mode

6.6 Add new Axes

1. Right-click **MOTION** and click **Add New Item**.

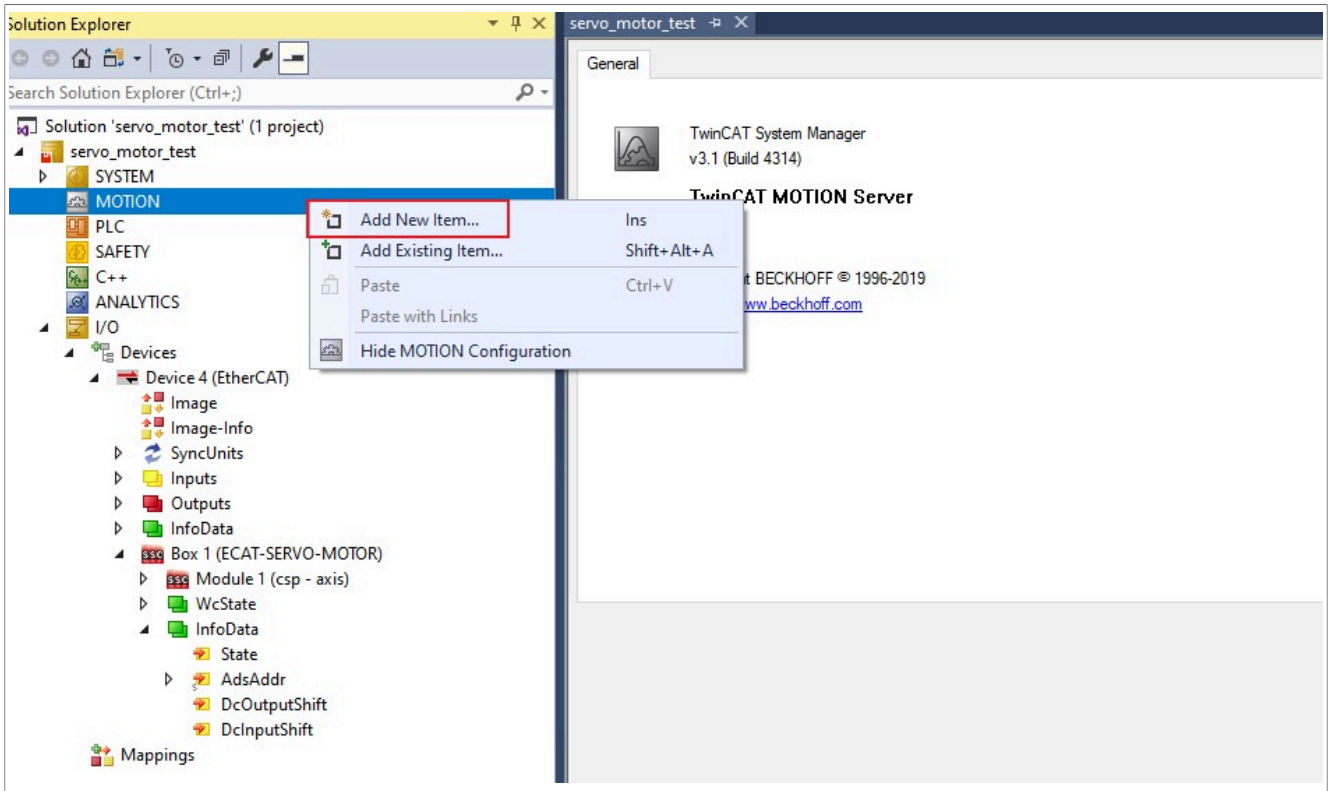


Figure 33. Add new item

The **Insert Motion Configuration** dialog box appears.

2. Select **NC/PTP NCI Configuration**.
3. Click **OK**.

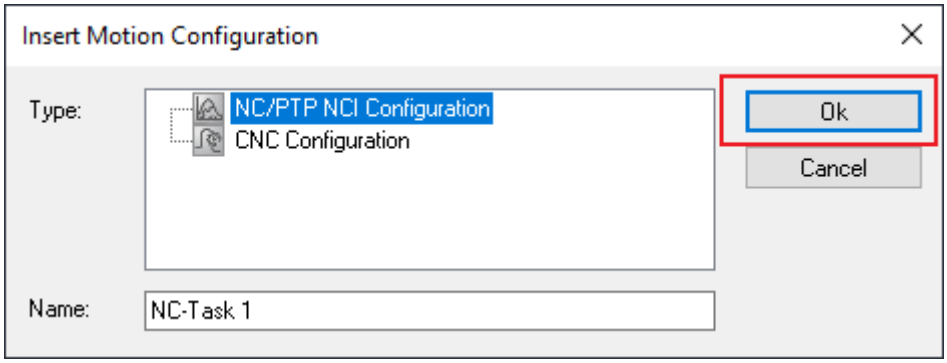


Figure 34. Select NC/PTP NCI configuration

4. Right-click **Axes** and click **Add New Item > OK**.

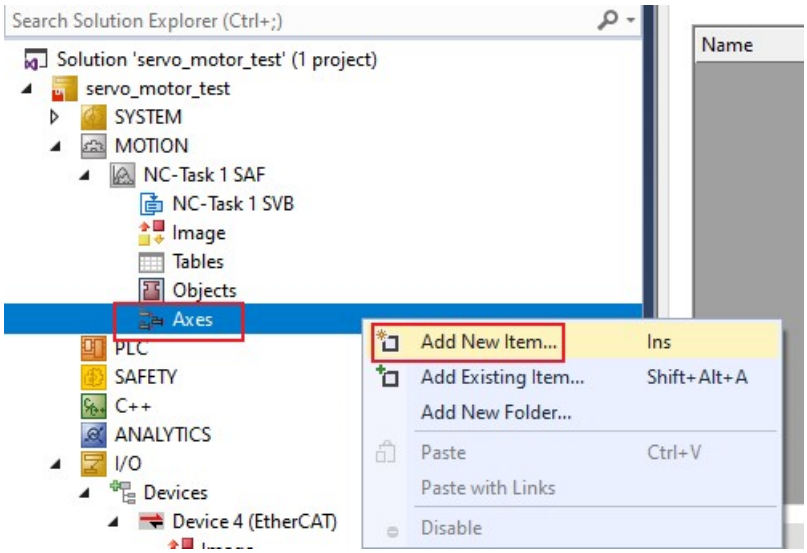


Figure 35. Add axes

The **Insert NC Axis** dialog box appears.

5. Provide the name in the **Insert NC Axis** text box.
6. Click **OK**.

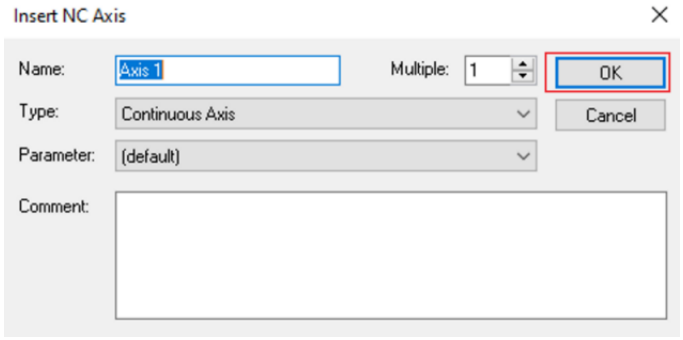


Figure 36. Insert NC Axis

6.7 Link I/O

1. Select **Axes > Axis1** from the **Solution Explorer**,
2. Click the **Settings** tab.
3. Select **Link To I/O > CANopen DS402, EtherCAT CoE**.
4. Click **OK**.

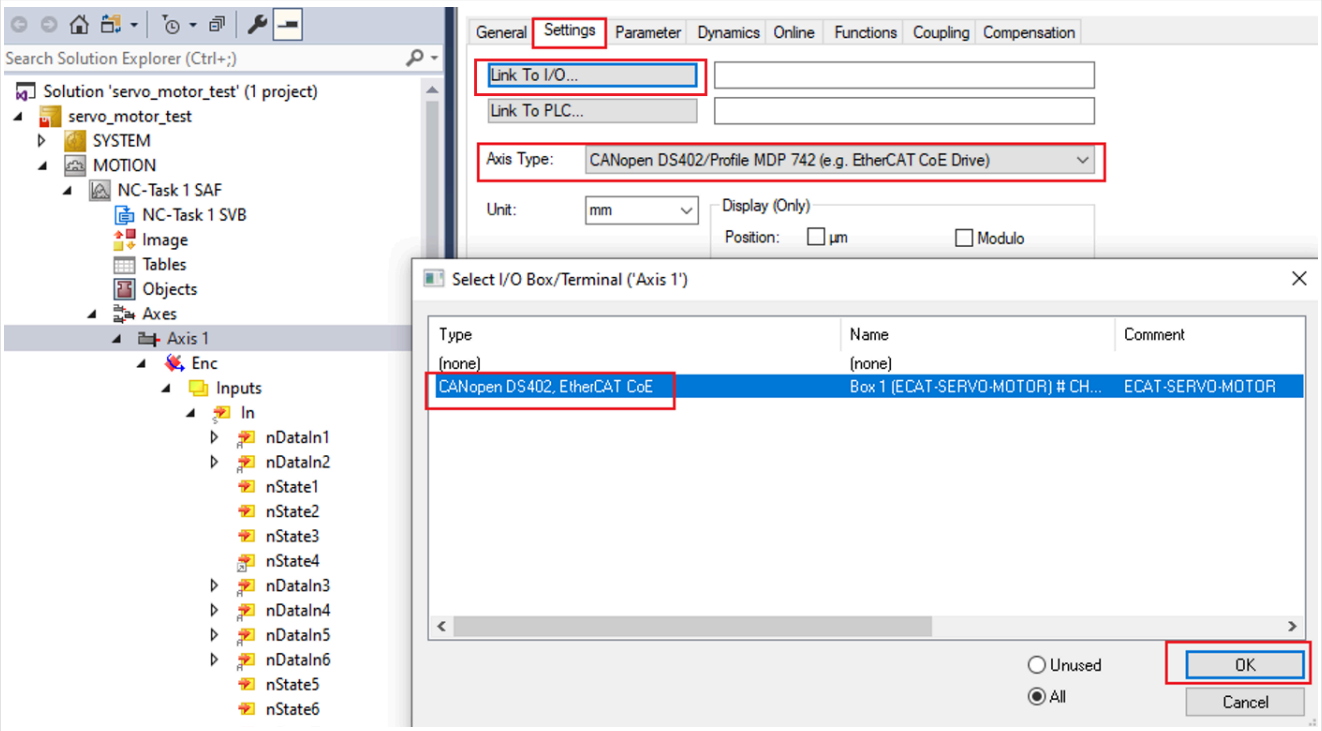


Figure 37. Link I/O

6.8 Configure Enc.

1. Select **Axes > Axis1 > Enc > Inputs > In > nDataIn1** from the **Solution Explorer**.
   The **TwinCAT Project** dialog box appears.
2. Click the **> Linked to** button.
   The **Attached Variable nDataIn1 (Input)** dialog box appears.
3. Select **Box1(ECAT-SERVO-MOTOR) > ActualPosition**.
4. Select **All Types** checkbox on the right to display all variables.
5. Click **OK**.

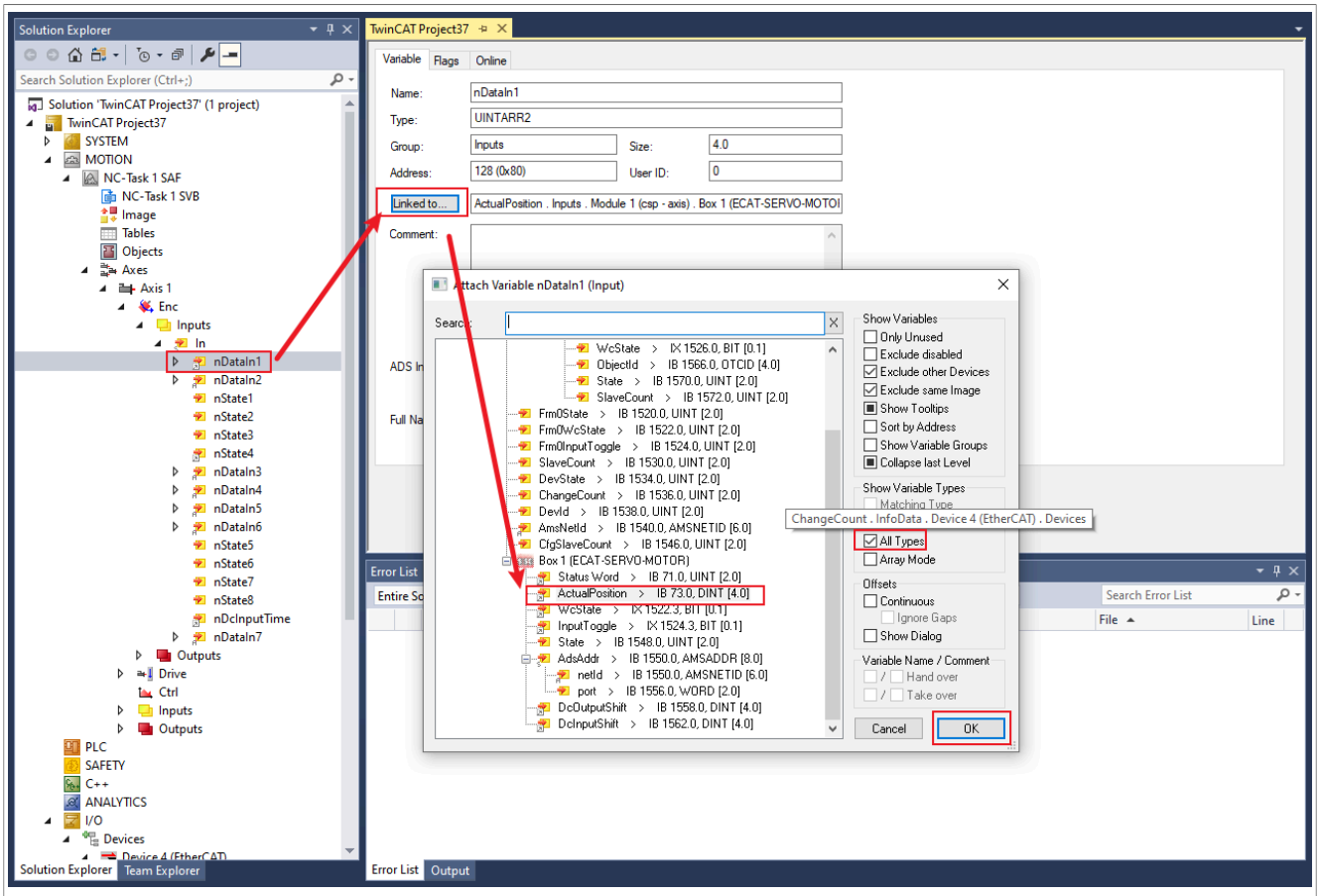


Figure 38. Configure Enc

6. Select **Axis1 > Enc >Inputs > In > nState4 > Linked to** from the **Solution Explorer** to link **InputToggle**.

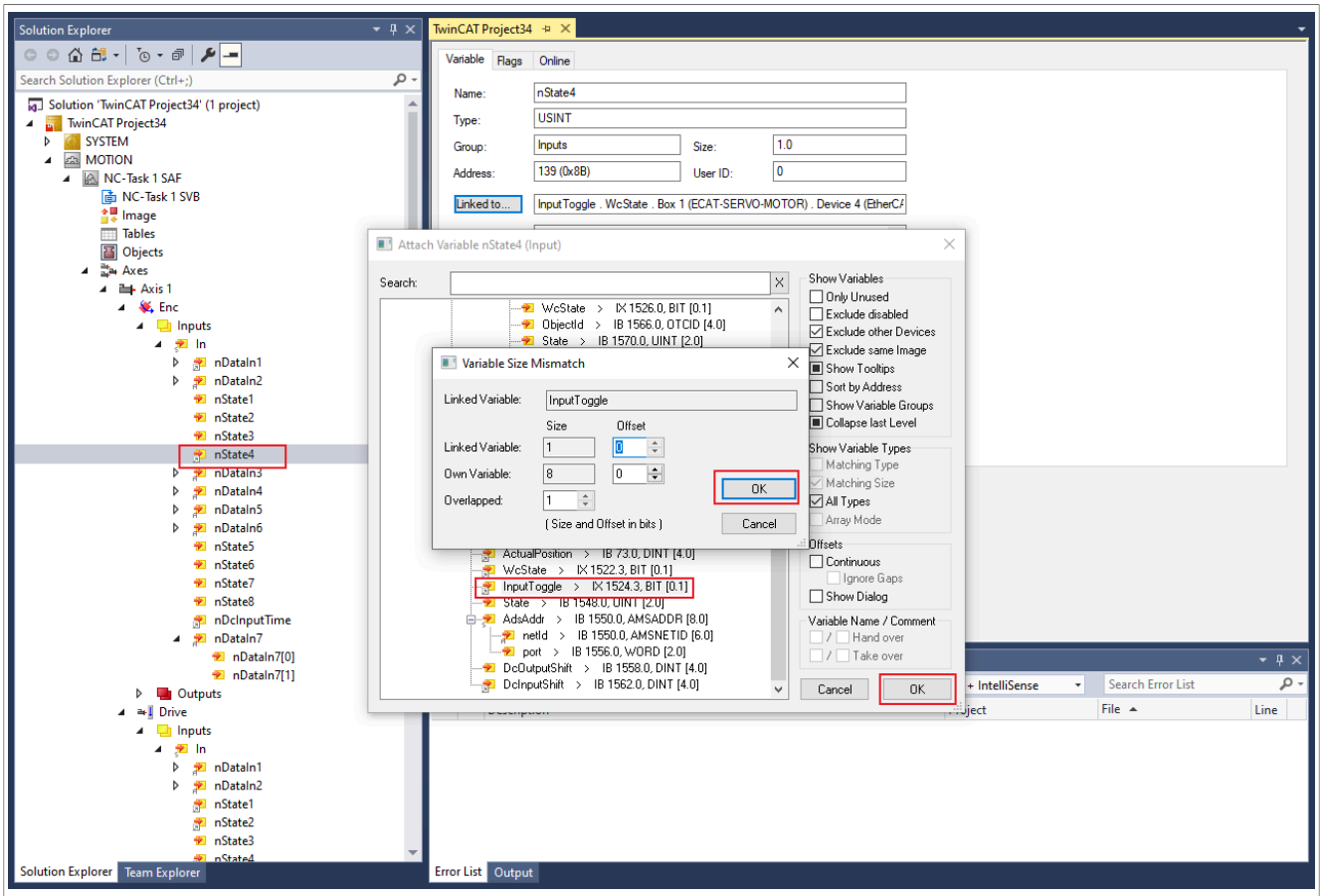


Figure 39. Set InputToggle

7. Select **Axes > Axis1 > Enc > Inputs > In > nDcInputTime > Linked to** to link **DcInputShift**.

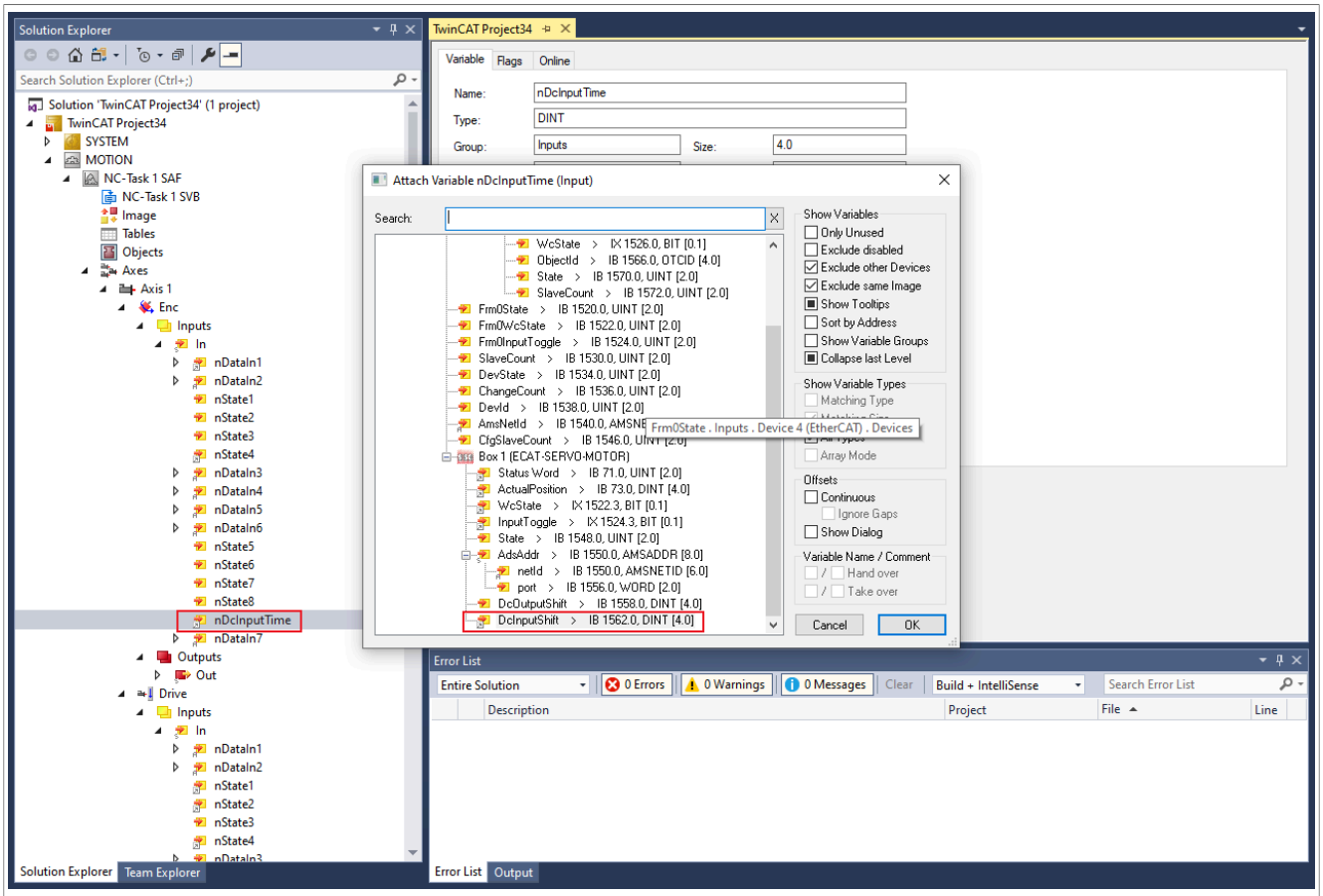


Figure 40. Link DcInputShift

6.9 Configure scaling factor

The encoder density of Tecknic2311P motor is 8,000 counts/rev. Tp configure Scaling Factor 1/8000; convert pulse counters to mm using below formula：

```
Target position(mm) = Motor revolutions(counts) * Scaling Factor
```

1. Select **Axes > Axis1 > Enc** from the **Solution Explorer**.
   The **TwinCAT Project** dialog box appears.
2. Click the **Parameter** tab.
3. Set **Scaling Factor Numerator** to `1`.
4. Set **Scaling Factor Denominator** to `8000`.

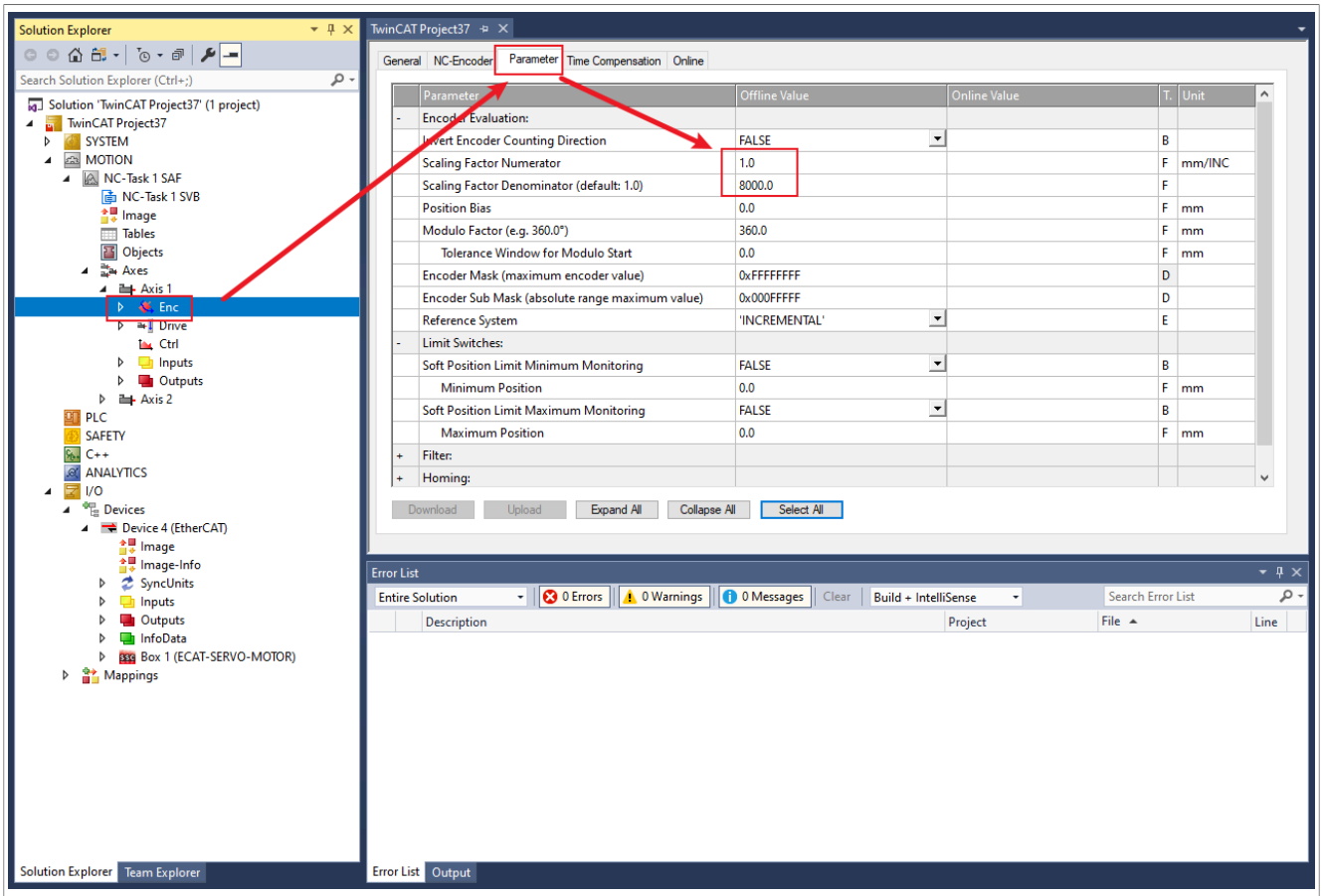


Figure 41. Configure the scaling factor

6.10 Configure drive

1. Select **Axes > Axis1 > Drive > Inputs > In > nState1** from the **Solution Explorer**.
   The **TwinCAT Project** dialog box appears.
2. Click the **Linked to** button.
   The **Attached Variable nState1 (Input)** dialog box appears.
3. Select **Box1(ECAT-SERVO-MOTOR) > Status Word Linked** to link **StatusWord**.
4. Click **OK**.

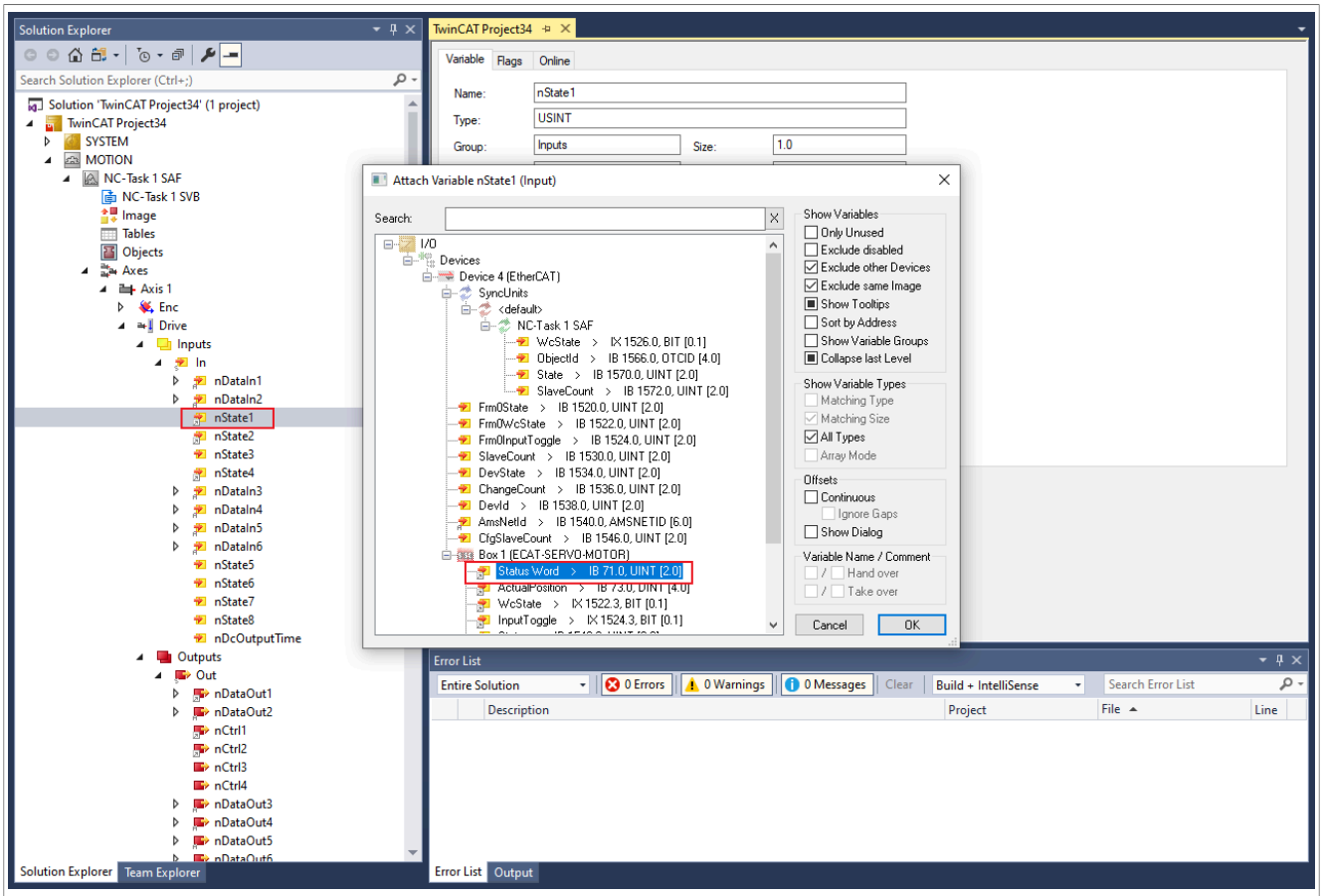


Figure 42. Configure drive

5. Click **OK** to link `StatusWord`.

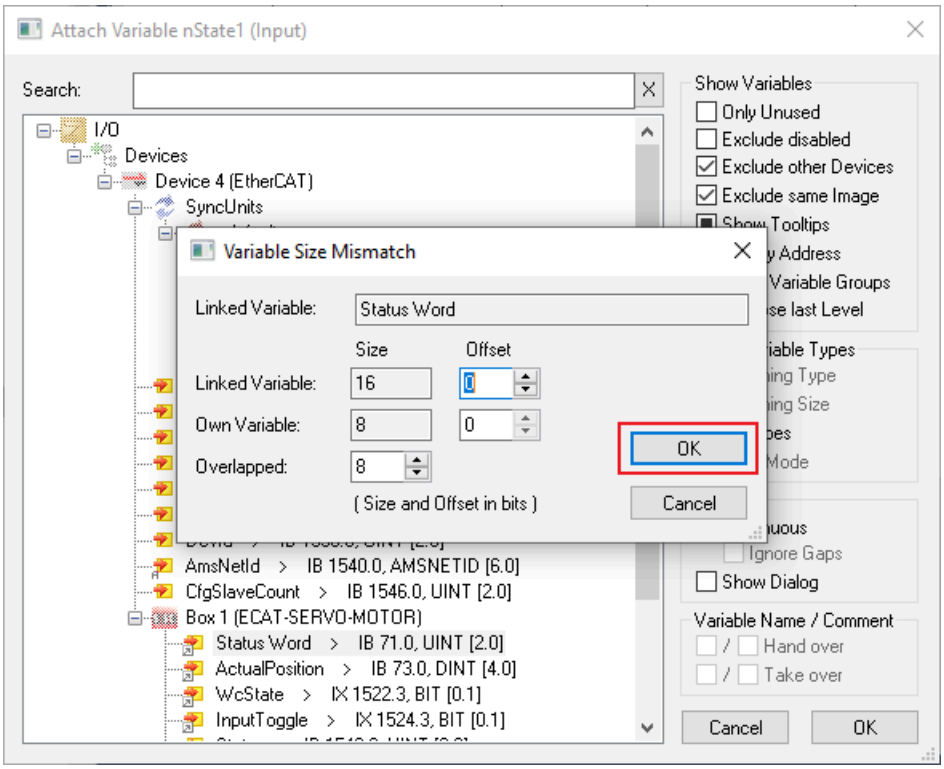


Figure 43. Attach variable nState1

6. Click **Axes > Axis1 > Drive > Inputs > In > nState2 > Linked to** to link `StatusWord`.

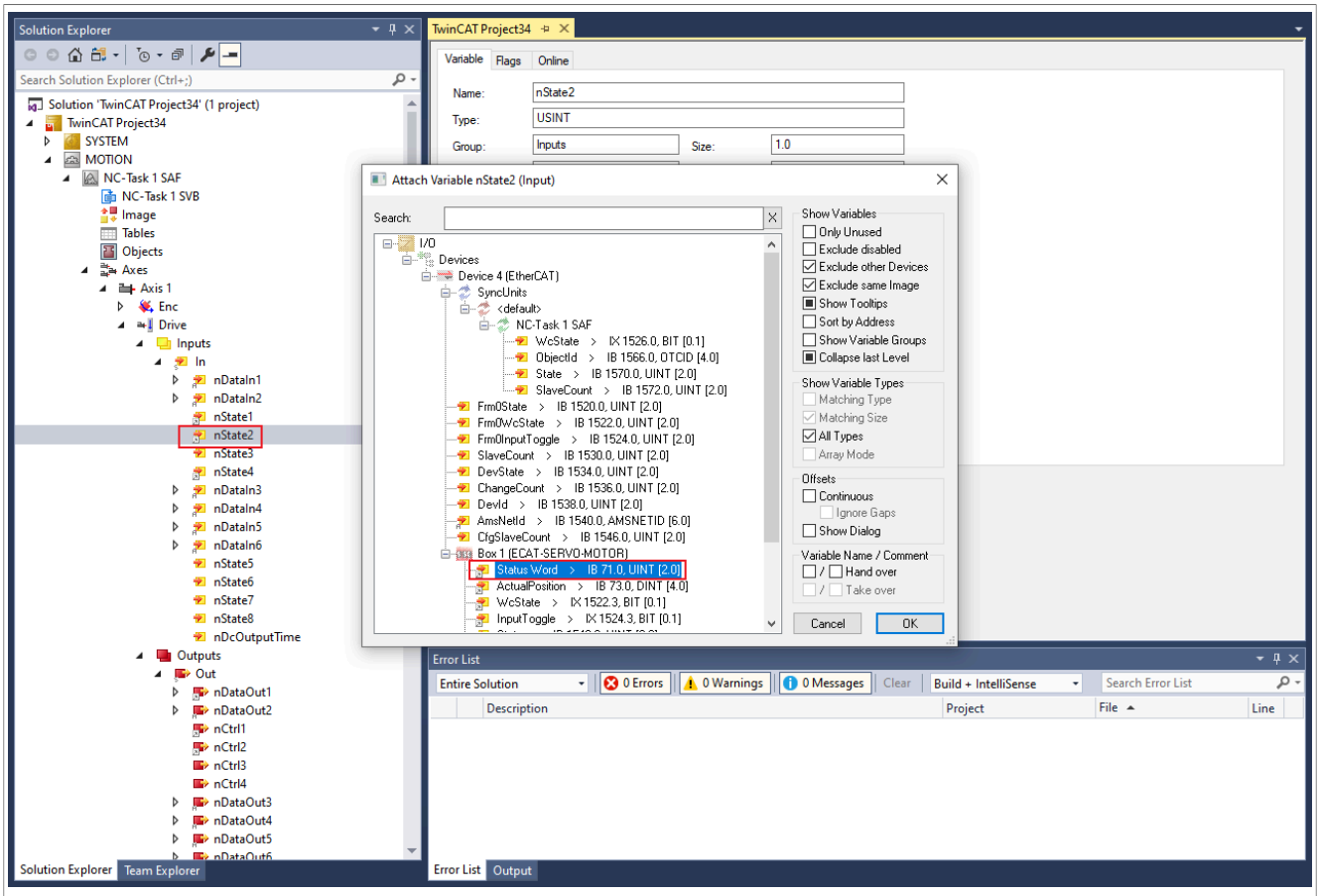


Figure 44. Link StatusWord

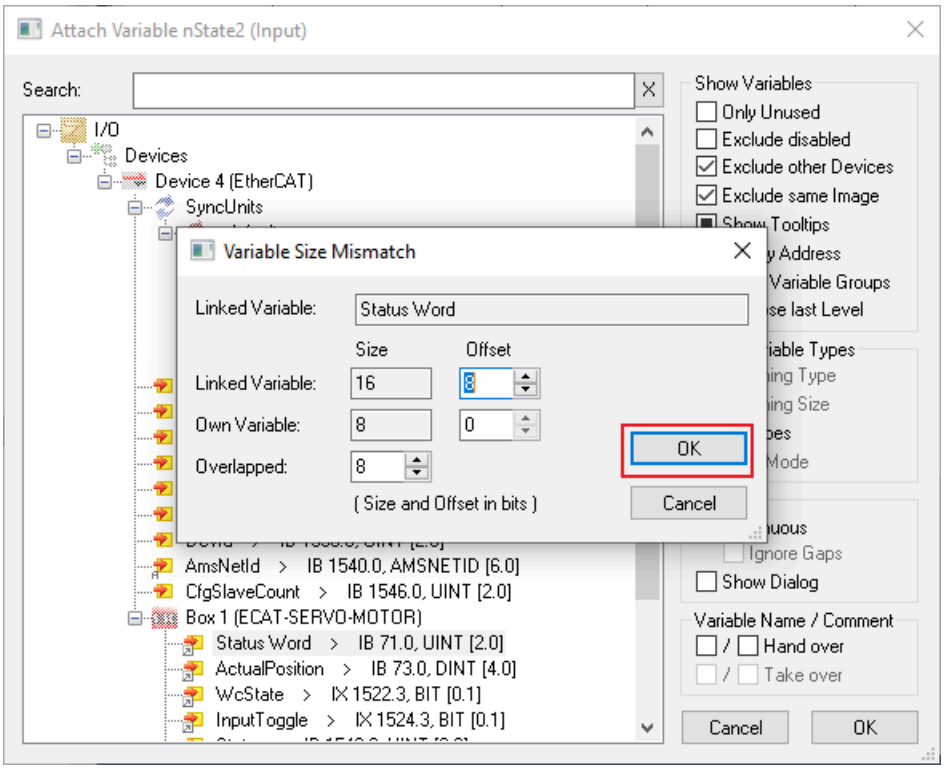


Figure 45. Attach variable nState2

7. Click **Axes > Axis1 > Drive > Inputs > In > nState4 > Linked to** to link **WcState**.

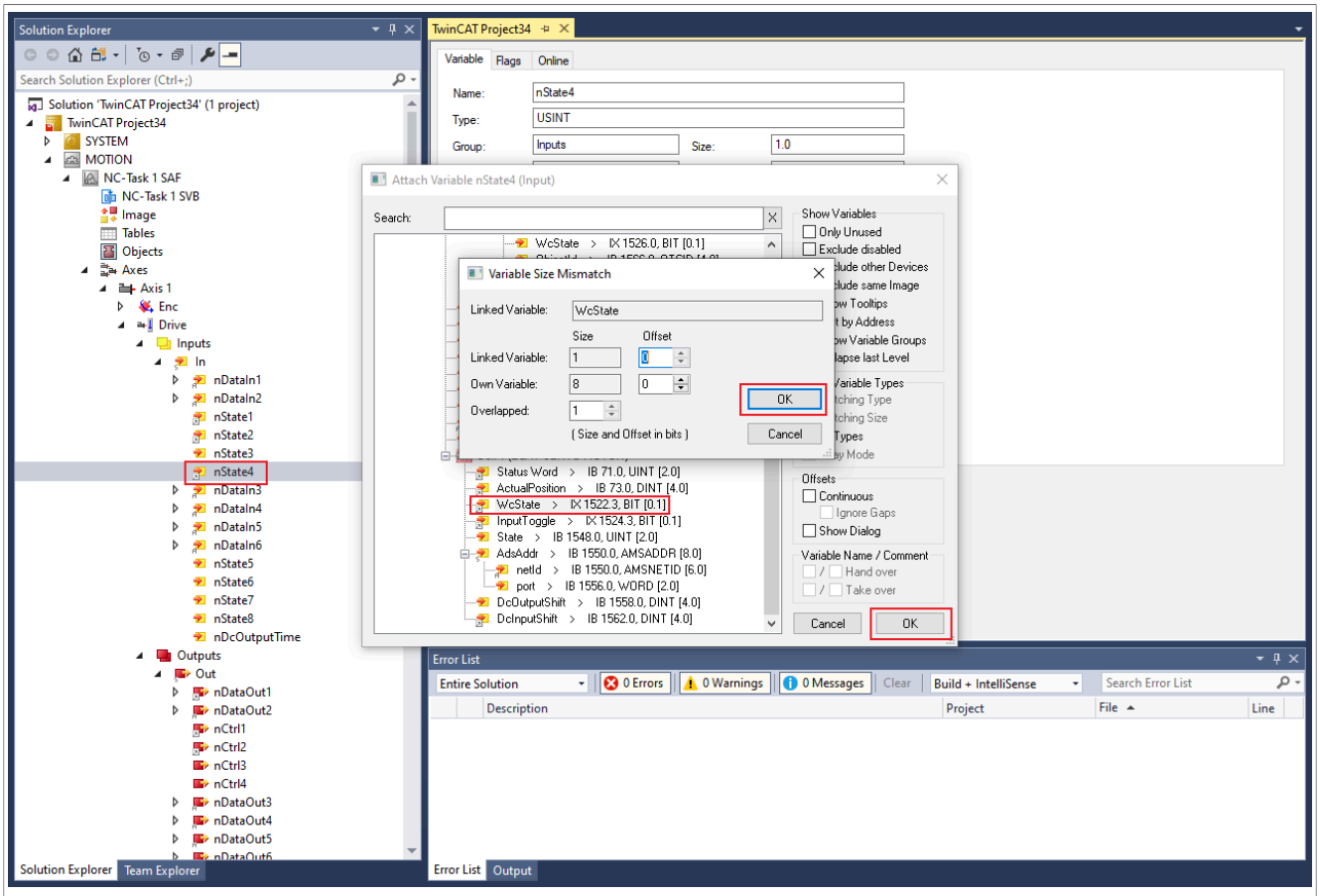


Figure 46. WcState and InputToggle

8. Click **OK**.

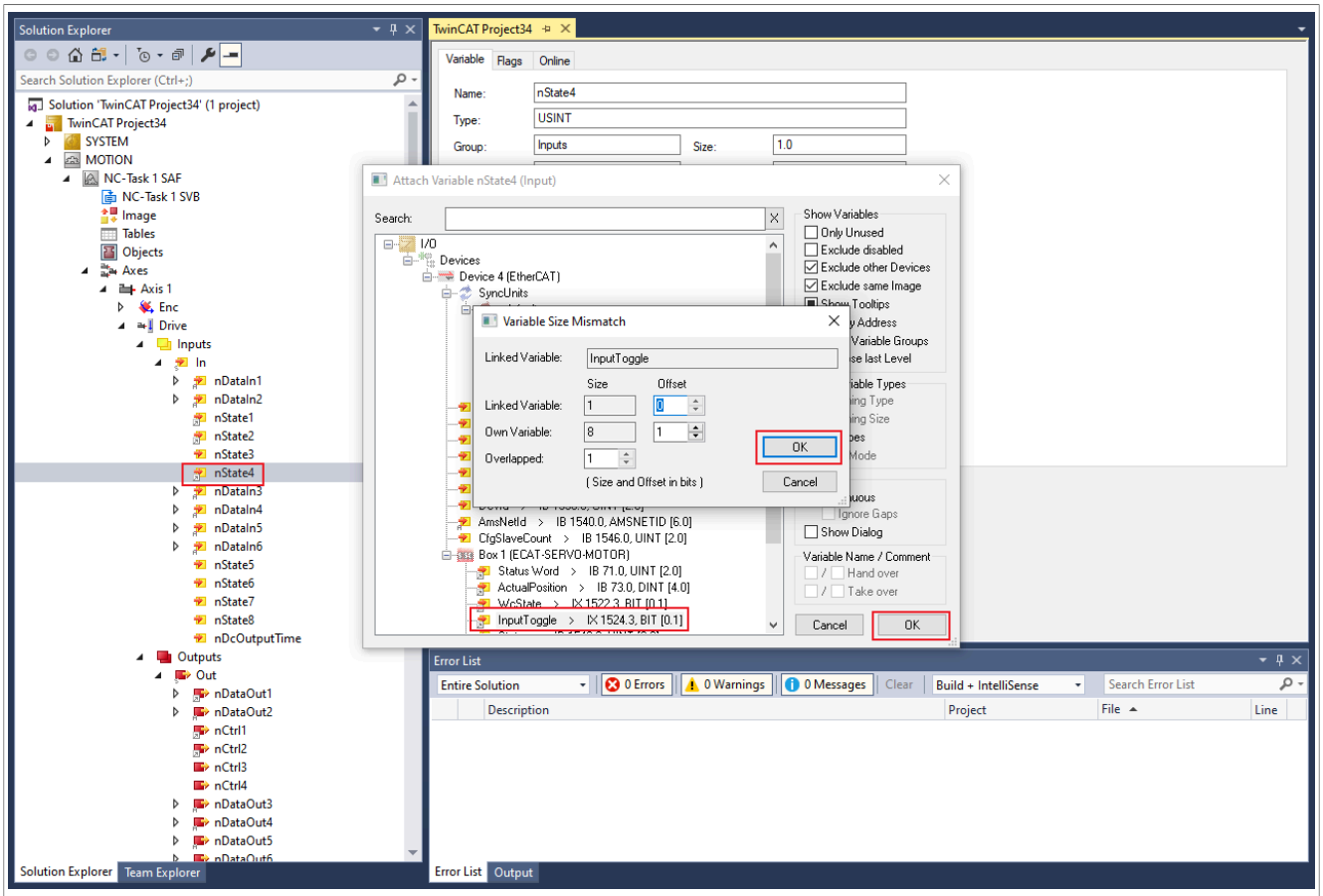


Figure 47. Attach variable nState4

9. Click **Axes > Axis1 > Drive > Outputs > Out > nDataOut1 > Linked to** to link **TargetPosition**.

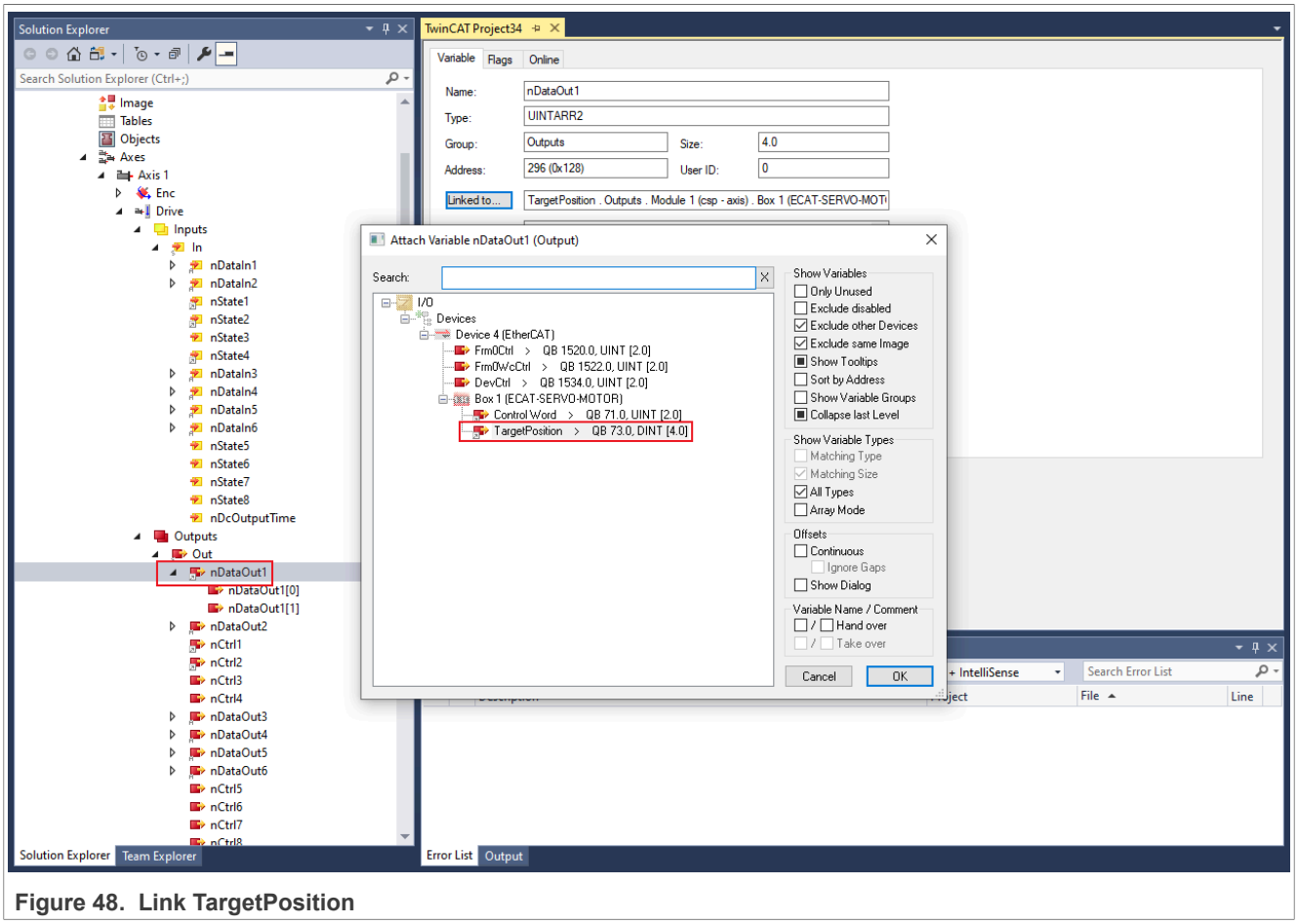


Figure 48. Link TargetPosition

10. Click **Axes > Axis1 > Drive > Outputs > Out > nCtrl1 > Linked to** to link **ControlWord**.

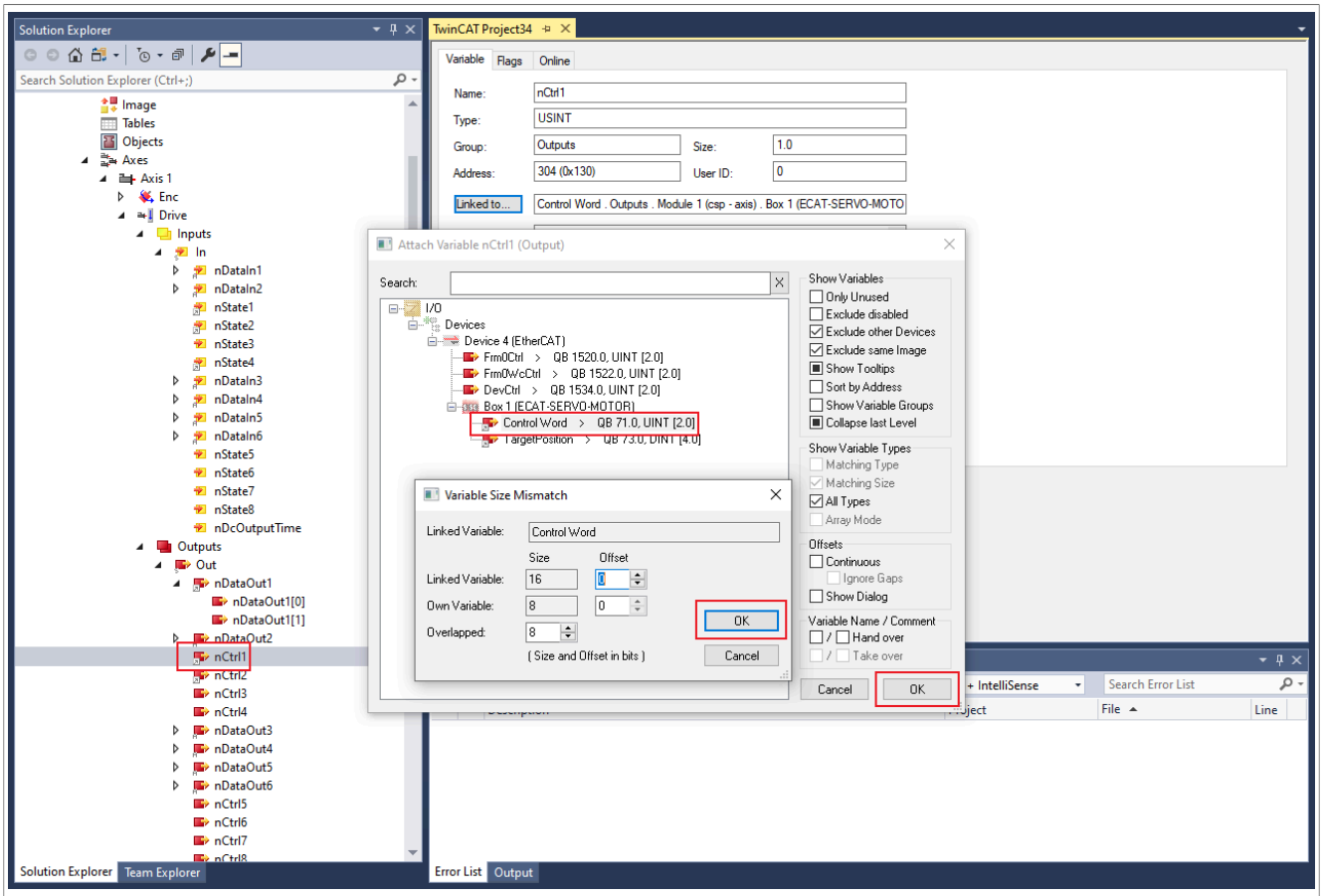


Figure 49. Link ControlWord

11. Click **Axes > Axis1 > Drive > Outputs > Out > nCtrl2 > Linked to** to link **ControlWord**.

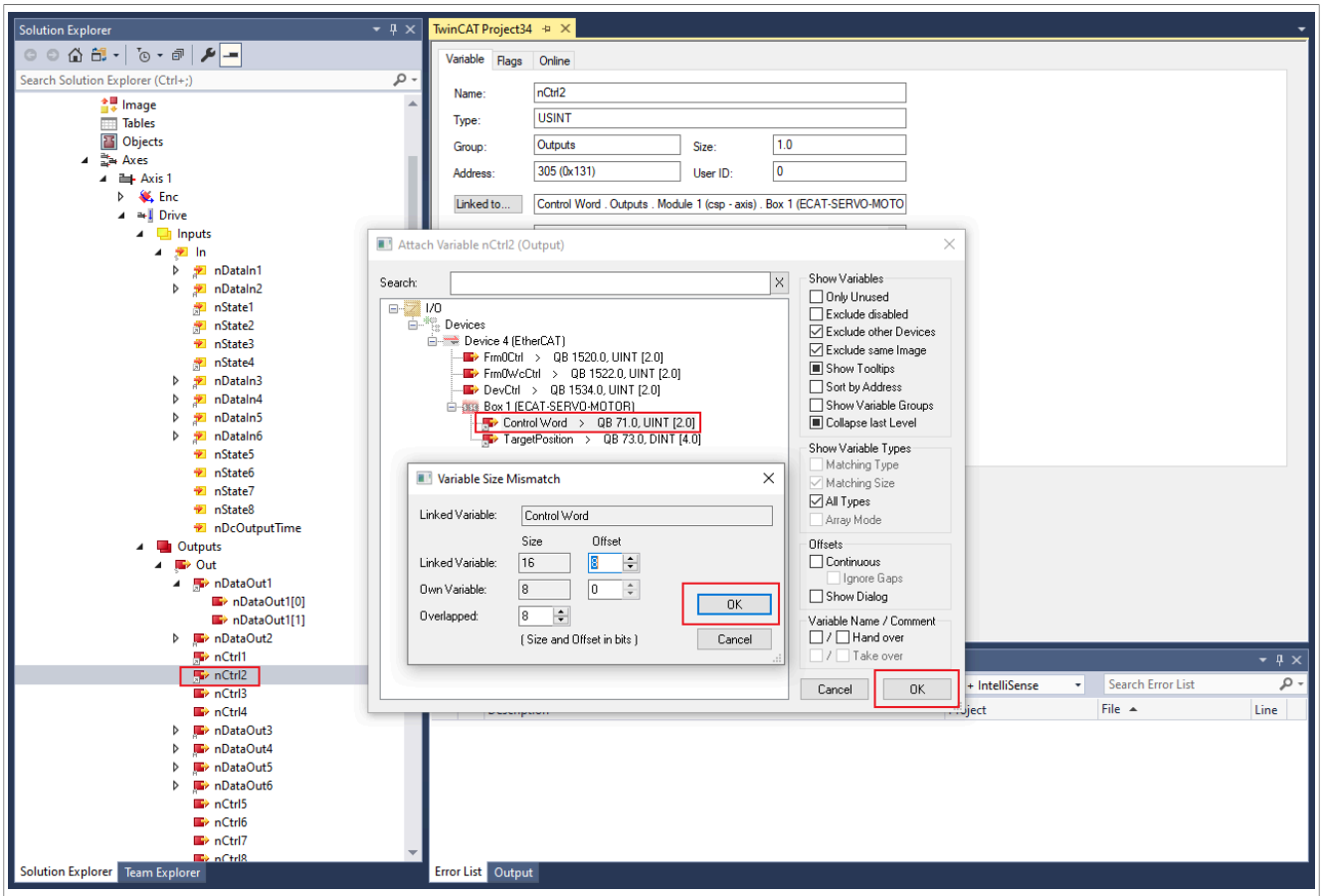


Figure 50. Attach Variable nCTrl2 (Output)

6.11 Stop position lag monitoring

1. Click **Axes > Axis1 > Parameter > Monitoring > Position Lag Monitoring** to choose `FALSE`.

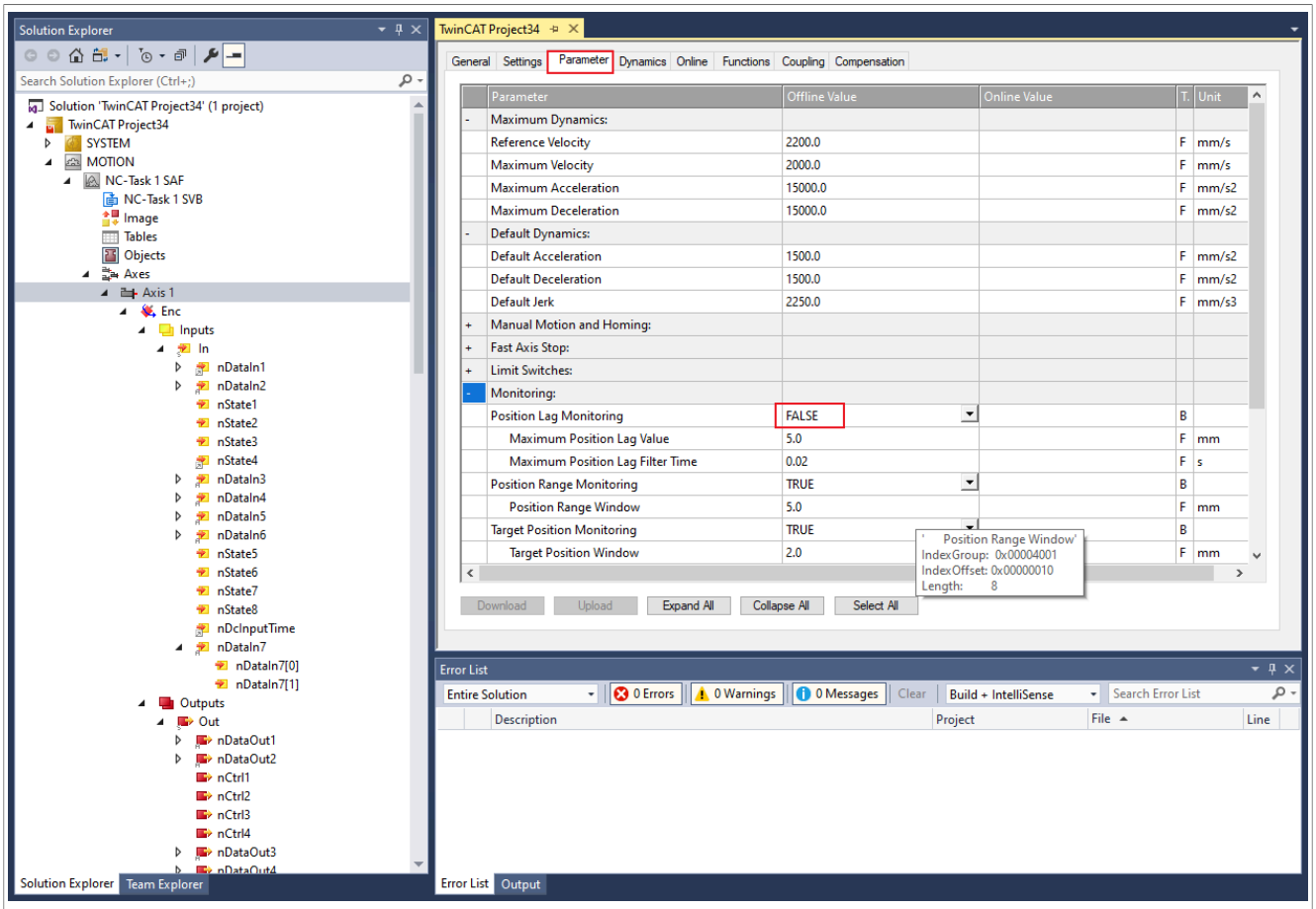


Figure 51. Stop position lag monitoring

6.12 Start to run motor

1. Activate the configuration by clicking **TwinCAT > Activate Configuration**.

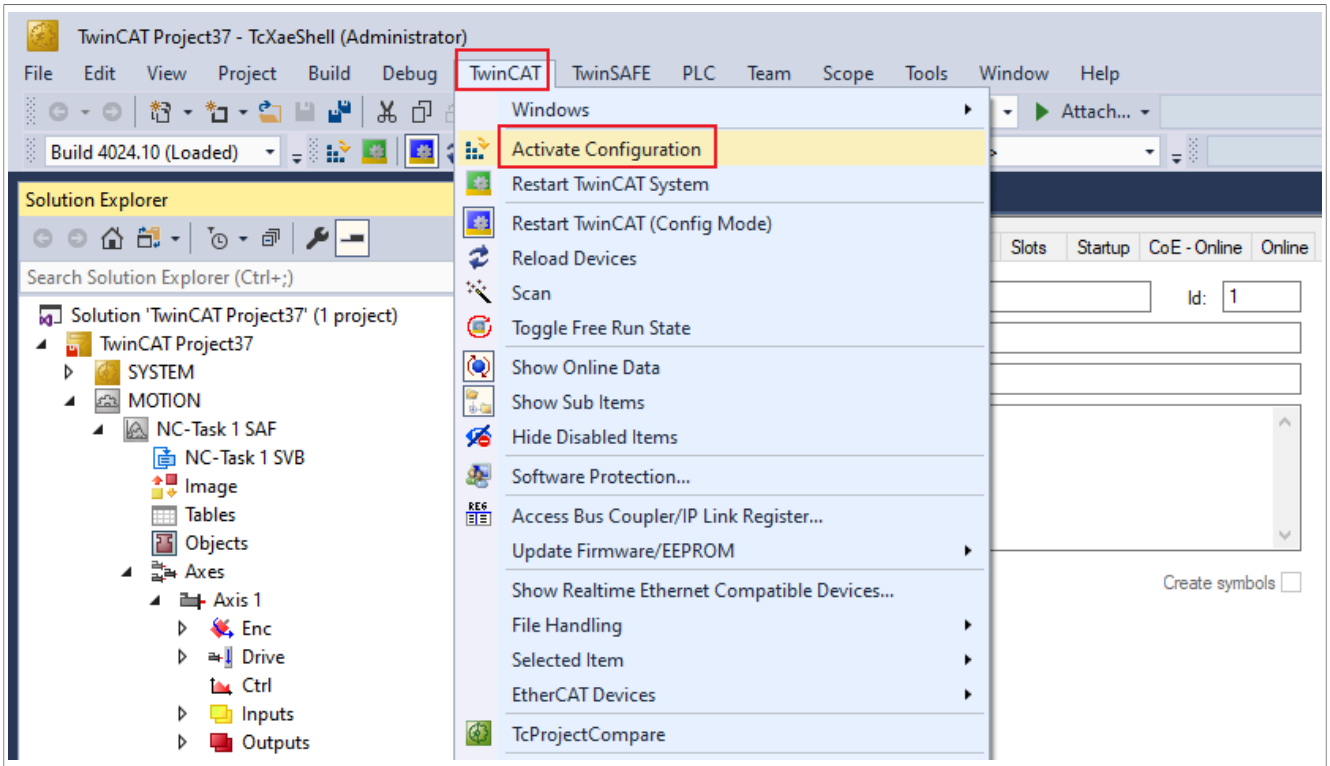


Figure 52. Activate configuration

2. Click **Axes > Axis1.**
3. Click the **Online** tab.
4. Click **Set**.
5. Set enabling and click **All**.

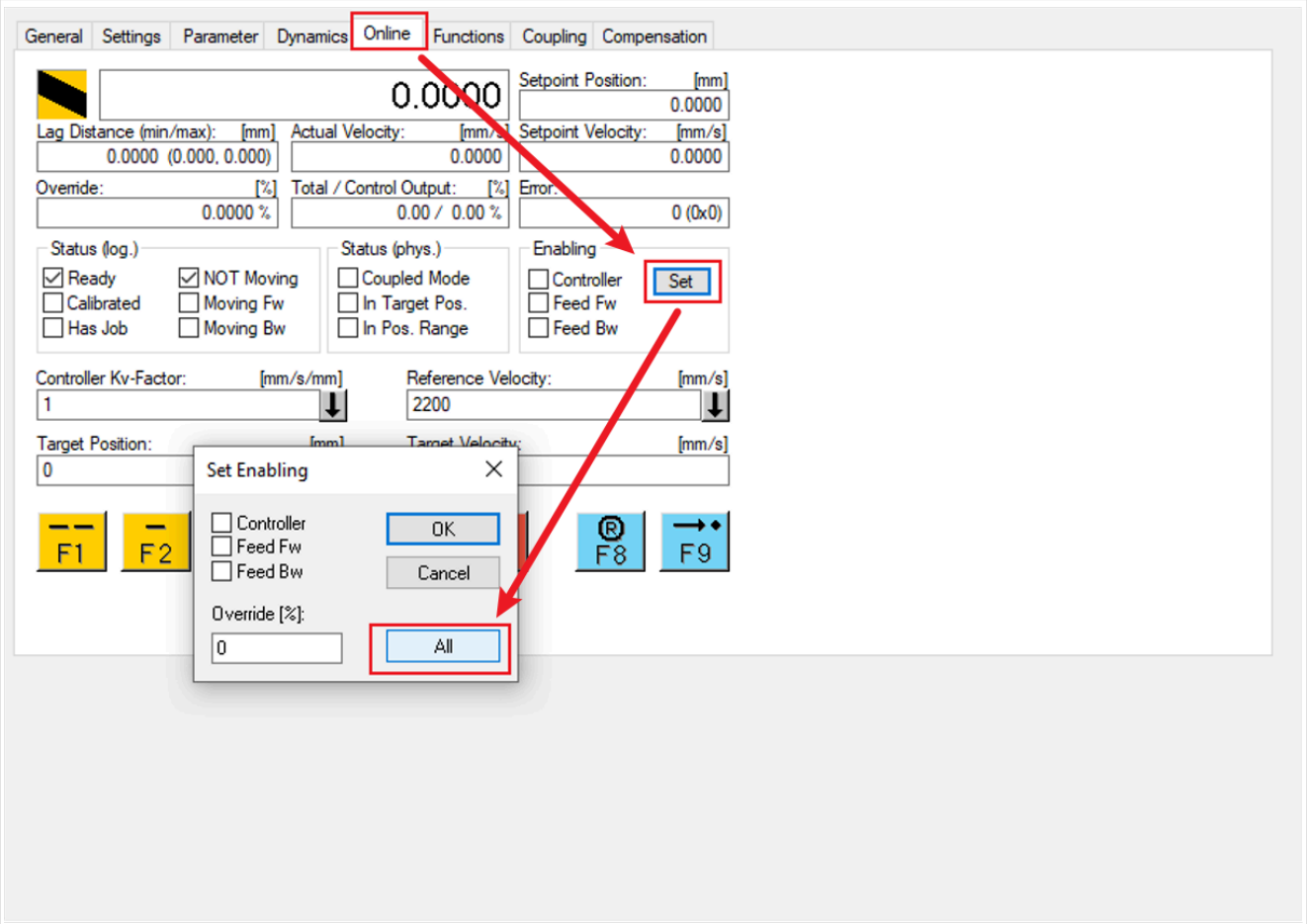


Figure 53. Start to run motor

6. Set **Target Position** and click the **F5** button. The motor will run to the target position.

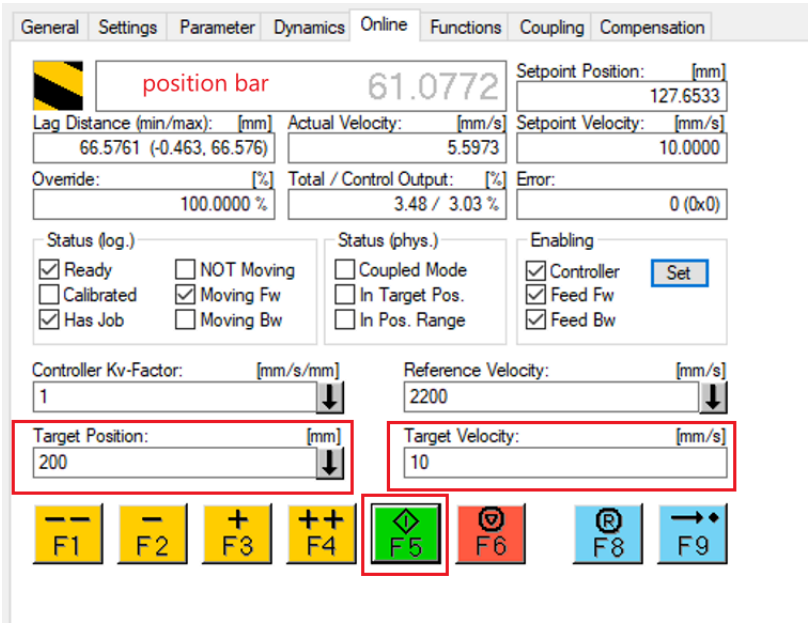


Figure 54. Set target position

Note: *If the motor does not run and the position bar is gray, follow the following steps.*

a. Click **Device 4 > Box 1 > WcState > WcState > Clear Link(s)**.

b. Click **Device 4 > Box 1 > WcState > InputToggle > Clear Link(s)**.

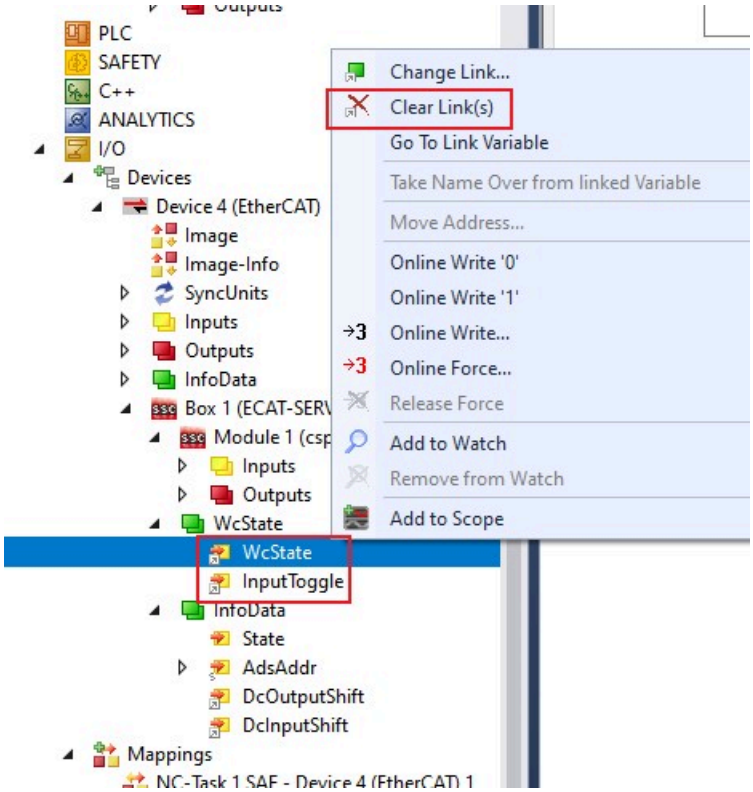


Figure 55. Clear links

7. Repeat step Section 6.12 "Start to run motor"..

7 Tuning motor parameters

Tuning motor parameters, constants and controller parameters can be necessary to achieve the required behavior and responses of a servo control application. There can be another motor type and different conditions than described in this document, so tuning parameters can be important. For the tuning motor parameters, perform the following steps:

1. Download the latest SDK package with the motor control software for permanent magnet synchronous motor (PMSM) from https://mcuxpresso.nxp.com/en/builder?hw=MIMXRT1180-EVK&components=nxp_motor_control.
2. Use the motor control example (*mc_pmsm*) for tuning motor parameters, constants, and so on. For more information about the application tuning, refer to the motor control application tuning (MCAT) training https://www.nxp.com/design/training/motor-control-application-tuning-mcat-tool-pmsm-servo-tuning:TIP-MCAT-PMSM.
3. When the parameters are tuned, use the updated m1_pmsm_appconfig.h header file with tuned parameters in the servo motor example.
   Note: *The application timings (slow state machine, fast state machine) must be the same for the motor control and servo motor examples.*

8 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

9 Revision history

Table 3 summarizes the revisions to this document.

Table 3. Revision history

Revision number	Release date	Description
1	06 November 2023	Initial public release
2	10 January 2023	Updated for MCUXpresso SDK 2.15.000.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

For more information, please visit: https://www.nxp.com

Date of release: 10 January 2024
Document identifier: eCAT Servo Motor Example