Building a Fault-tolerant EtherCAT Network

Moxa Technical Support Team

support@moxa.com

Contents

 Released on Aug 26, 2025

About Moxa

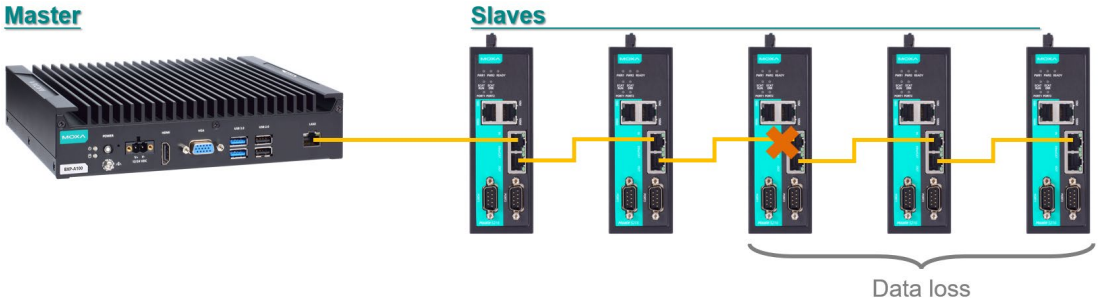
Moxa is a leading provider of edge connectivity, industrial computing, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things. With over 35 years of industry experience, Moxa has connected more than 111 million devices worldwide and has a distribution and service network that reaches customers in more than 91 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for industrial communications infrastructures. Information about Moxa's solutions is available at www.moxa.com.

How to Contact Moxa

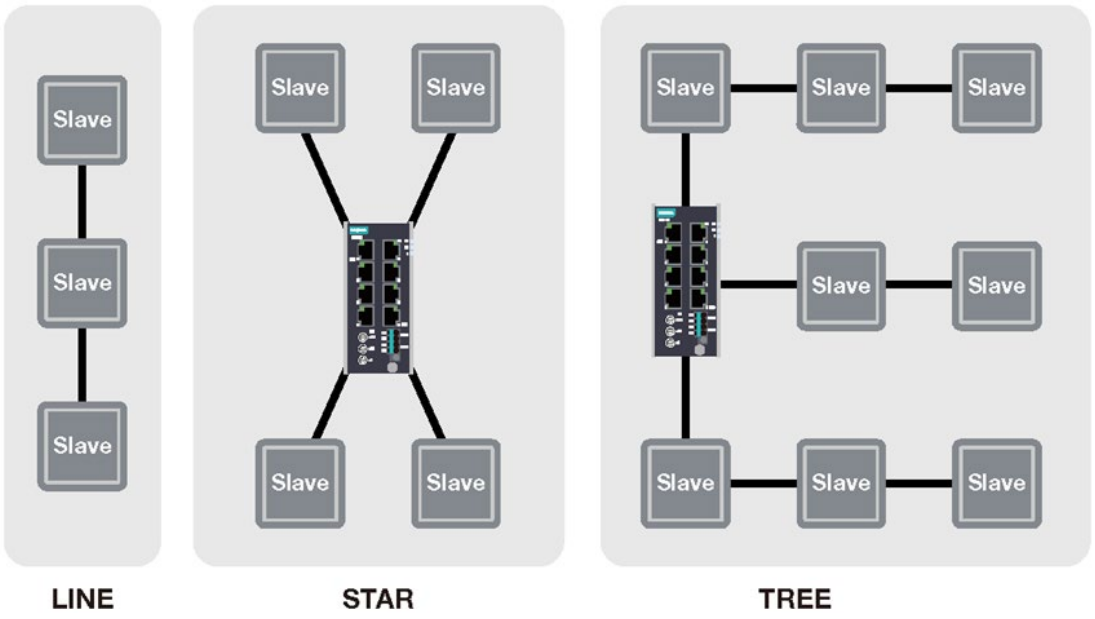
Tel: 1-714-528-6777
Fax: 1-714-528-6778

1 Introduction

Most EtherCAT slave devices come with only two ports, making the daisy-chain topology the standard connection method. However, this setup has a critical weakness: if any single device fails or disconnects, all downstream devices lose communication with the master.



Since failure points are unpredictable, the impact can vary widely depending on where the issue occurs. To solve this, **the Moxa EJS Series EtherCAT junction allows for topology conversion, breaking the single chain into multiple independent branches, like tree, and star topologies.**



This approach isolates faults to a single path, ensuring that other devices remain unaffected—greatly improving network reliability.

This document outlines how to build a robust EtherCAT topology with the Moxa EJS Series in a TwinCAT 3 environment, where the EJS isolates faults to a single path so if one device fails, communication on other branches continues unaffected.

1.1 Target Audience

This application guide is for control systems engineers, automation specialists, and machine builders who are tasked with improving the reliability of their EtherCAT systems. It is ideal for professionals seeking to move beyond basic daisy-chain configurations to implement more robust, fault-tolerant network architectures.

2 System Topology

This document outlines the recommended EtherCAT connection settings to **avoid single-point failures** when using the **Moxa EJS Series**, in conjunction with TwinCAT 3.

Specifically, it demonstrates two key aspects:

1. **EtherCAT Master-Slave Communication:**

   Show how the EtherCAT Master communicates with the connected slave devices, performing read and write operations on I/O data.

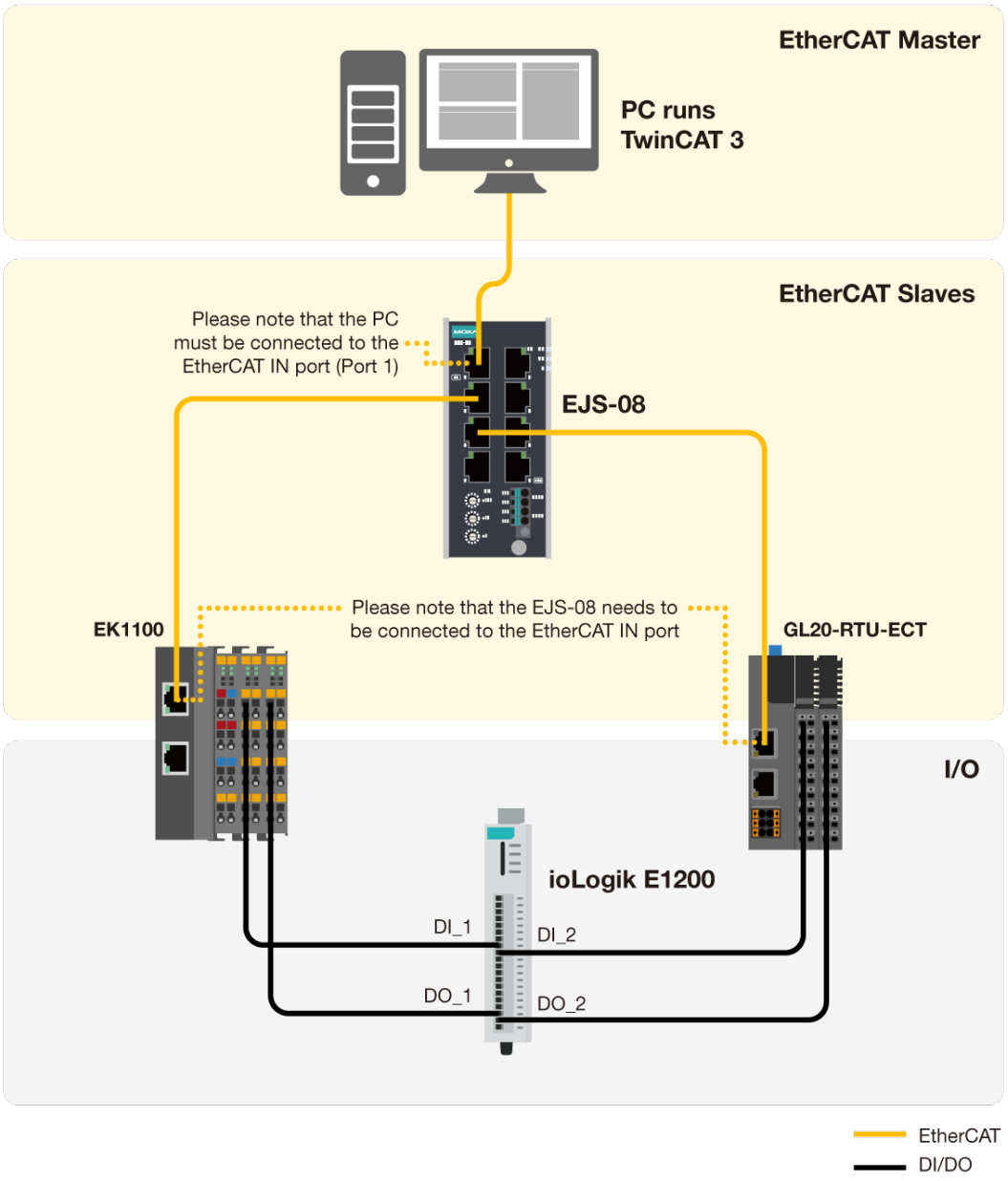
2. **Avoiding Single-Point Failures by Using an EJS:**

   If an EtherCAT slave device on one branch fails—due to disconnection or malfunction—devices on other branches remain unaffected. This independent branch operation helps prevent single-point failures across the system.

The EtherCAT Master is simulated using **TwinCAT 3**, and the EtherCAT slave devices include:

- BECKHOFF EK1100 with EL2008 and EL1008 modules
- INOVANCE GL20-RTU-ECT with 0008ETN and 0800END modules

To facilitate I/O status observation, an ioLogik E1200 is connected to both BECKHOFF and INOVANCE setups, providing an easy and visual way to monitor data changes during the demonstration.



3 Required Equipment and Components

1. **PC Requirements**
   a. **OS:** Win 10 / Win 11
   b. **Ethernet Port:** RJ45 connector
2. **EJS Series**

   Industrial EtherCAT Junction

   Rev: v1.0 or higher
3. **BECKHOFF EK1100**

   EtherCAT coupler, with EL2008 (DO terminal) and EL1008 (DI terminal) modules.
4. **INOVANCE GL20-RTU-ECT**

   EtherCAT coupler, with GL20-0008ETN (DO terminal) and GL20-0800END (DI terminal) modules.
5. **TwinCAT 3**

   As a registered Beckhoff account, you can download the TwinCAT 3 EtherCAT master simulator.

   Version: V3.1

   Download website:

   https://www.beckhoff.com/en-en/support/download-finder/search-result/?download_group=97028248&download_item=650023470

   Installation guide videos (Reference):

   [Chinese] https://www.youtube.com/watch?v=1jNBuxxvJfI

   [English] https://www.youtube.com/watch?v=WkNOm-hMH3k

Note If the PC doesn't have Visual Studio when installing TwinCAT 3, the TwinCAT XAE Shell will be installed automatically.

4 After Installing TwinCAT 3

After installing TwinCAT 3, these steps must be performed.

4.1 Run the win8settick.bat file on each computer

When using a PC/NB as the simulator of the Beckhoff controller, **before** creating the project, you need to run the win8settick.bat file on each computer. This only needs to be done once for each computer.

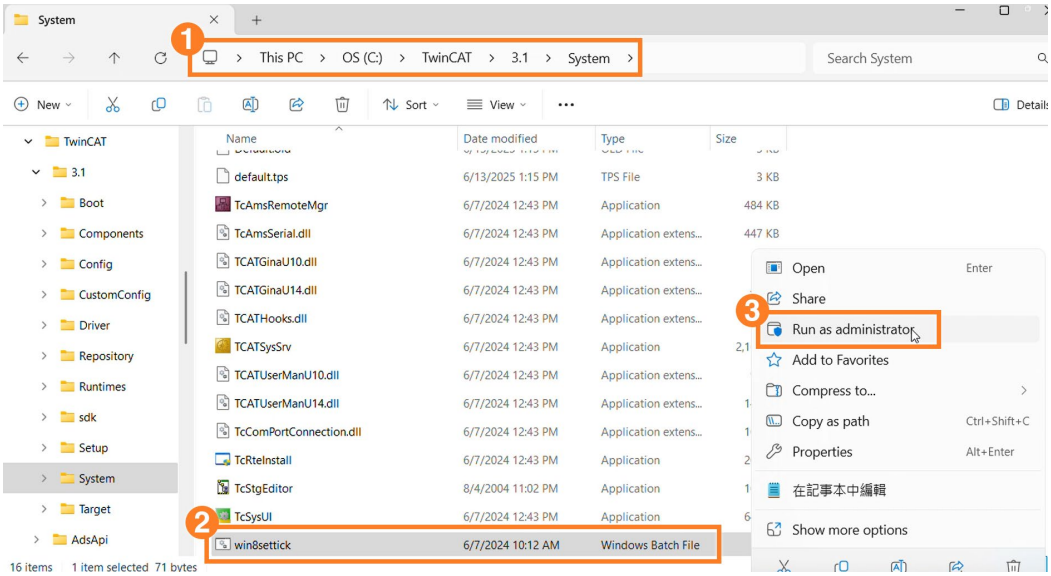
1. Go to the TwinCAT installation folder.

   By default (for TwinCAT 3.1), this will be in `C:\TwinCAT\3.1\System`

2. Right-click the **win8settick.bat** file.
3. Select **Run as administrator**.

   The process will take a few seconds, and the pop-up window will close automatically.

4. After the process is complete, restart the computer.



4.2 Install the TwinCAT Realtime Ethernet driver

You need to install the TwinCAT real-time Ethernet driver for the network card that will be used as the EtherCAT Master.

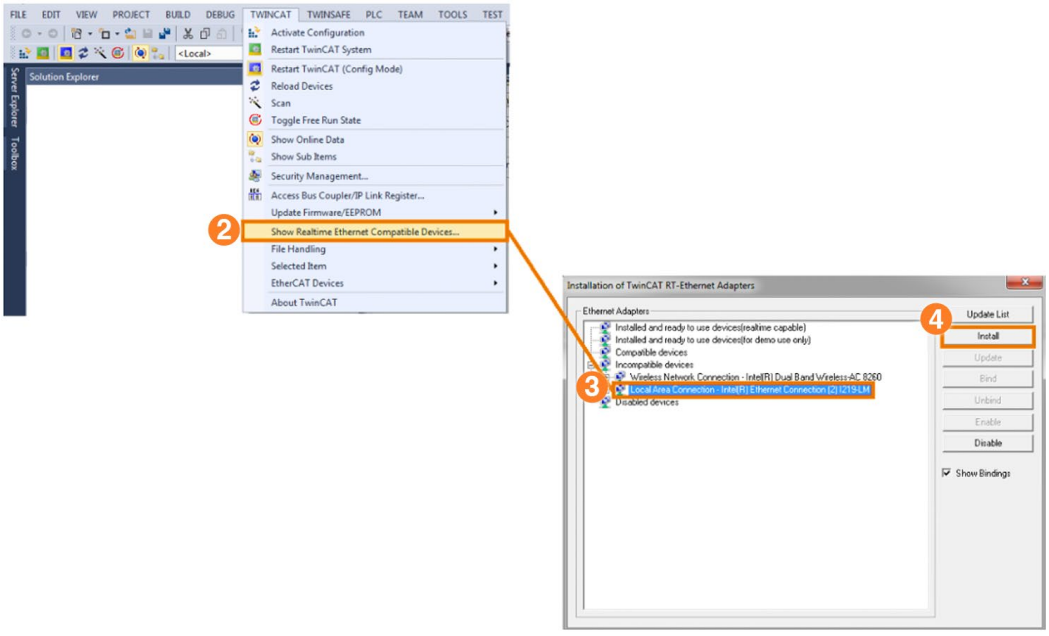
1. Open TwinCAT 3 and create a project.
2. Go to **TWINCAT** > **Show Realtime Ethernet Compatible Devices…**
3. Select the network card that will be used as the EtherCAT Master.

Note If a network card is listed under **Incompatible devices**, this does not mean that it cannot be used to test the EtherCAT communication; this only means that this card will provide weak real-time capabilities. For most testing purposes, this will be sufficient.

4. Select **Install** to install the driver.

   After the driver is installed, the network card will be moved to one of the **Installed and ready-to-use devices** list:

   - **Installed and ready to use devices(realtime capable)**: Shows compatible devices.
   - **Installed and ready to use devices(for demo use only)**: Shows incompatible devices.



4.3 Import ESI files into TwinCAT

To get the most out of your EtherCAT system, importing EtherCAT SubDevice Information (ESI) files is essential. These XML-based files act as detailed guides for each EtherCAT SubDevice, allowing tools like TwinCAT 3 to automatically configure the network and generate an accurate EtherCAT Network Information (ENI) file.

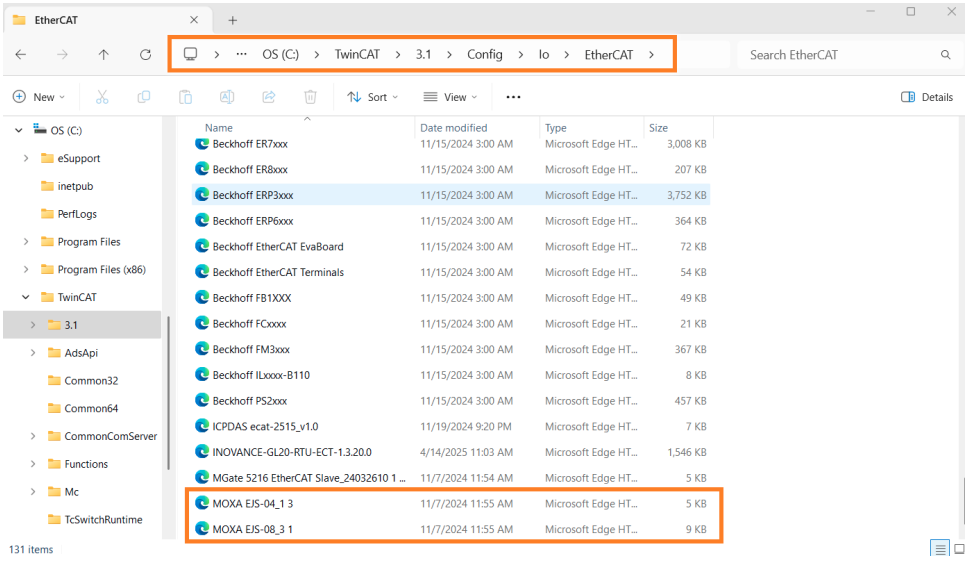
Using ESI files ensures smoother setup, easier integration, and reliable communication—especially in systems with devices from different vendors. For the best performance and compatibility, always include the correct ESI files for every device in your network.

1. Use the links below and download the ESI files for your EJS Series device.
   - EJS-04:

     https://www.moxa.com/en/products/accessories/ethercat-automation-devices/ejs-04-series#resources
   - EJS-08:

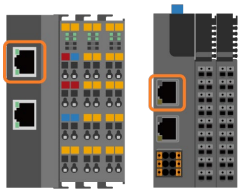
     https://www.moxa.com/en/products/accessories/ethercat-automation-devices/ejs-08-series#resources
2. Copy the ESI file to TwinCAT's \Config\Io\EtherCAT folder.

   By default (for TwinCAT 3.1), this will be in **C:\TwinCAT\3.1\Config\Io\EtherCAT**



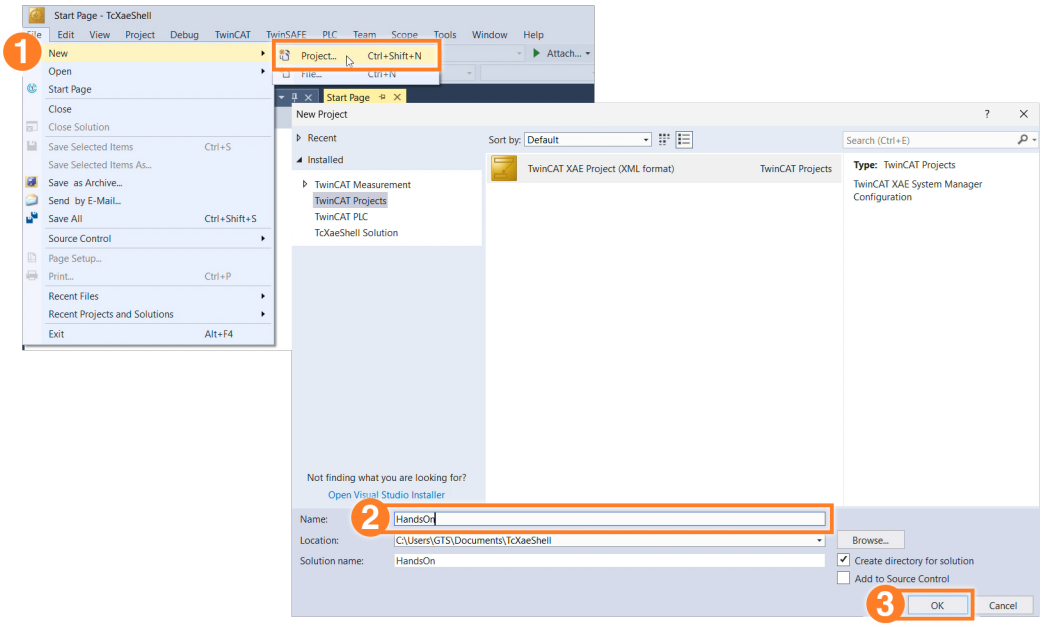
5 TwinCAT 3 Settings

Note In this scenario, please make sure that the cable is connected to the **EtherCAT In** port of the slave devices, or you may encounter errors.



5.1 Create a new TwinCAT project

1. In TwinCAT 3, go to **FILE** > **New** > **Project...**
2. Enter a **Name** for your TwinCAT project.
3. Select **OK**.



5.2 Add EtherCAT slave devices (Method 1)

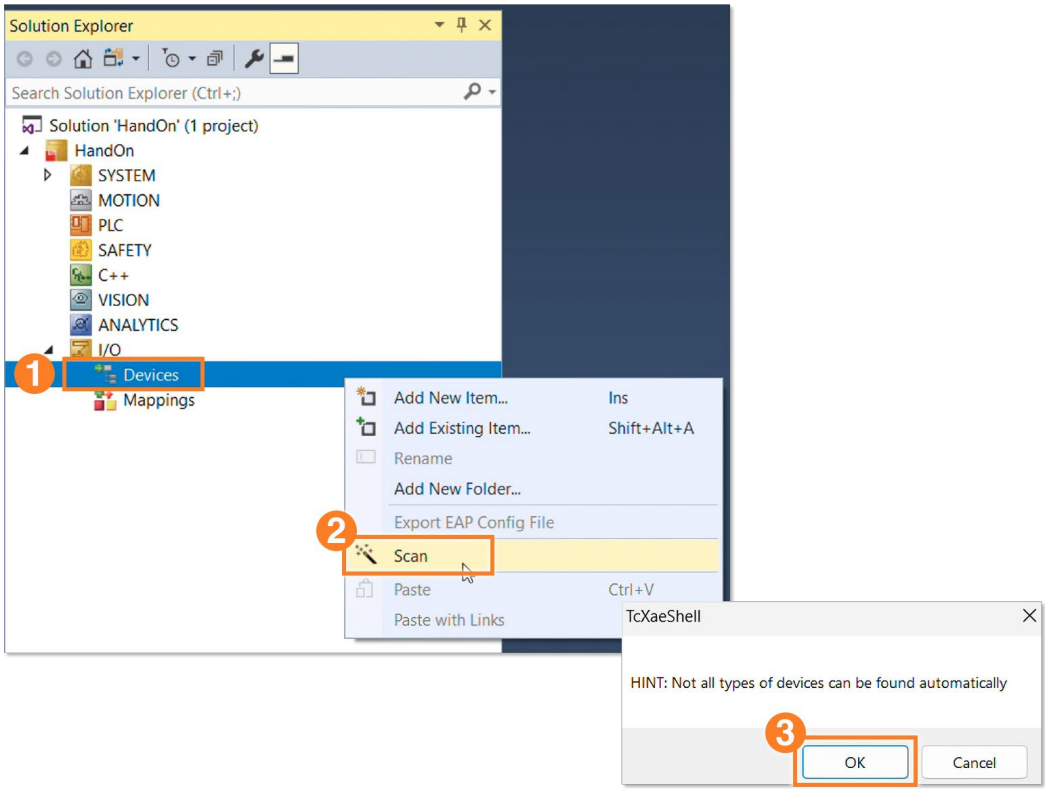
This method will scan for EtherCAT devices currently in the network.

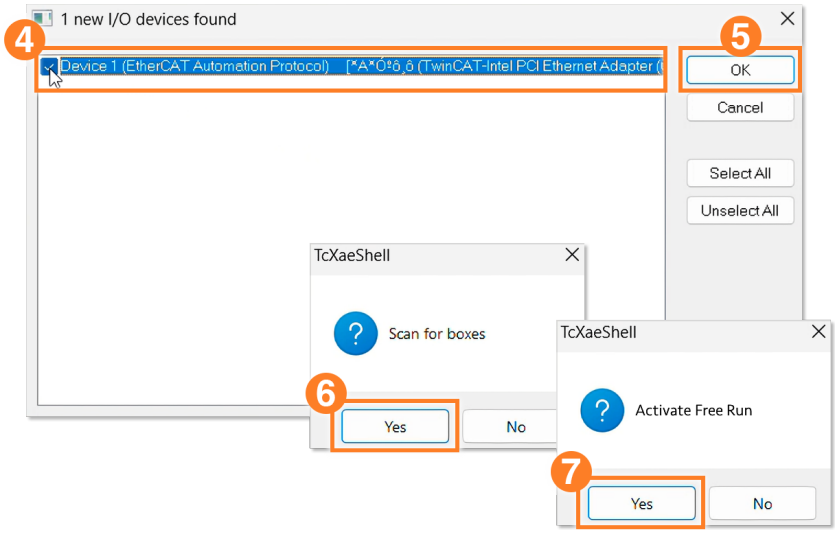
Note Before performing this task, please make sure the EJS is properly connected to the EK1100, GL20-RTU-ECT, and the PC running TwinCAT 3.

1. In the Solution Explorer, right-click on **I/O** > **Devices**.
2. Select **Scan**.
3. In the **HINT: Not all types of devices can be found automatically** window, select **OK**.
   Your PC will scan for EtherCAT devices on the network.
4. Select the network card to use as the EtherCAT master device.

Note You should select the network card you installed the TwinCAT Realtime Ethernet driver for. The EtherCAT slave devices should be connected to this card.

5. Select **OK**.
6. In the **Scan for boxes** window, select **Yes**.
7. In the **Activate Free Run** window, select **Yes**.





5.3 Add EtherCAT slave devices (Method 2)

If Method 1 does not work, you may need to manually add an EtherCAT master device before scanning for slave devices.

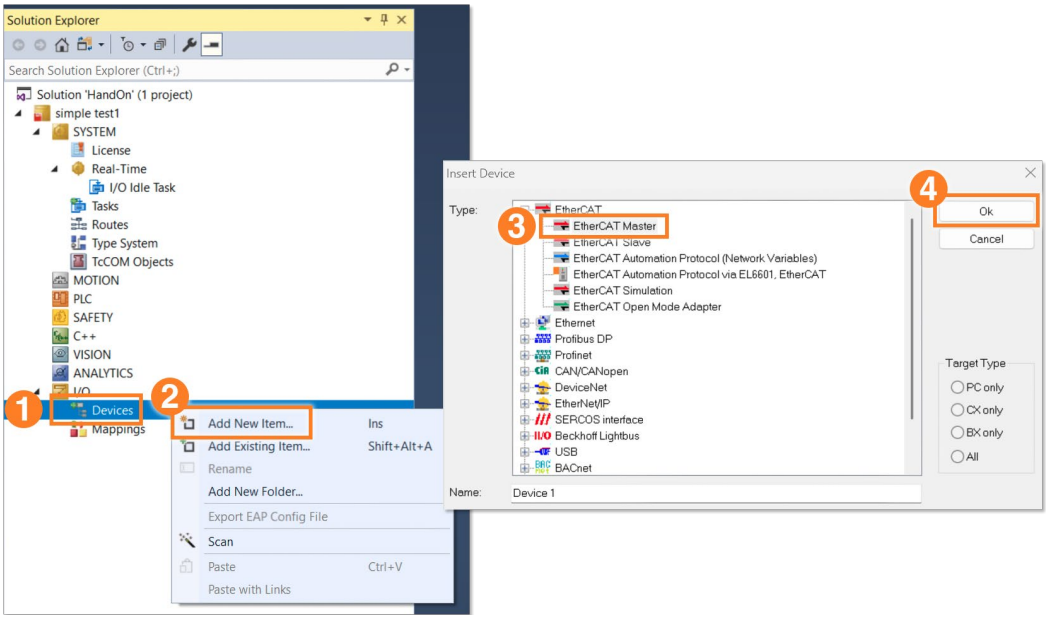
5.3.1 Add EtherCAT master device

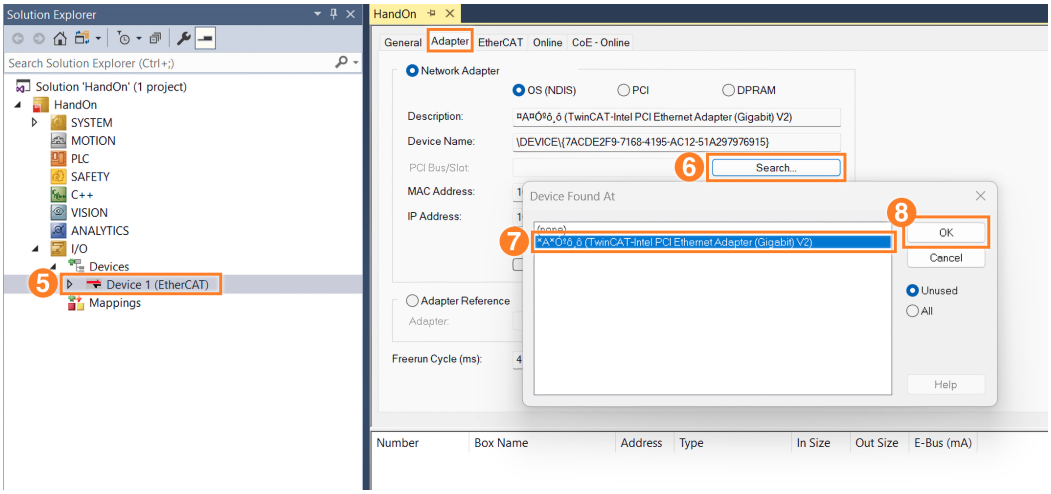
1. In the Solution Explorer, right-click on **I/O** > **Devices**.
2. Select **Add New Item...**
3. Select **EtherCAT** > **EtherCAT Master**.
4. Select **Ok**.

   This will create a new device in **I/O** > **Devices**.
5. Double-click on the device you created in **I/O** > **Devices**.
6. In the **Adapter** tab, select **Search...**
7. Select the network card to use as the EtherCAT master device.

Note You should select the network card you installed the TwinCAT Realtime Ethernet driver for. The EtherCAT slave devices should be connected to this card.

8. Select **OK**.

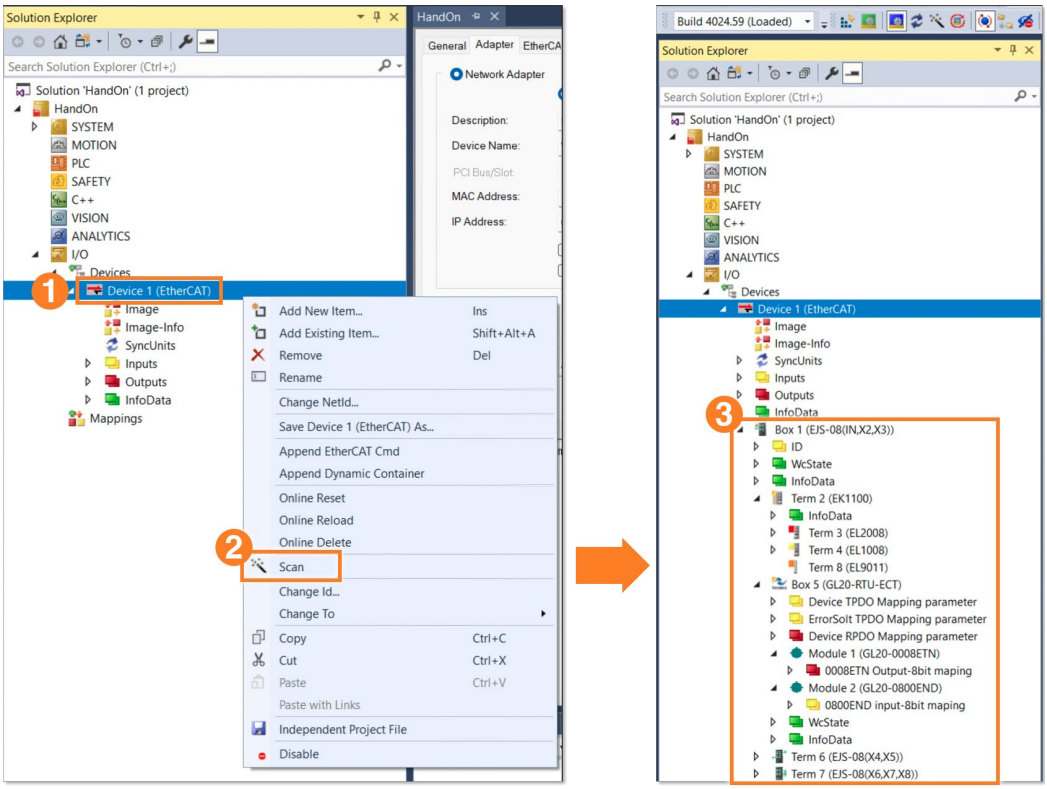

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'HandOn' (1 project)
simple test1
SYSTEM
License
Real-Time
I/O Idle Task
Tasks
Routes
Type System
TcCOM Objects
MOTION
PLC
SAFETY
C++
VISION
ANALYTICS
I/O
Devices
Mappings
1
2
Add New Item... Ins
Add Existing Item... Shift+Alt+A
Rename
Add New Folder...
Export EAP Config File
Scan
Paste Ctrl+V
Paste with Links
Insert Device
Type:
EtherCAT
3
EtherCAT Master
EtherCAT Slave
EtherCAT Automation Protocol (Network Variables)
EtherCAT Automation Protocol via EL6601, EtherCAT
EtherCAT Simulation
EtherCAT Open Mode Adapter
Ethernet
Profibus DP
Profinet
CAN/CANopen
DeviceNet
EtherNet/IP
SERCOS interface
Beckhoff Lightbus
USB
BACnet
4
Ok
Cancel
Target Type
PC only
CX only
BX only
All
Name:
Device 1


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'HandOn' (1 project)
HandOn
SYSTEM
MOTION
PLC
SAFETY
C++
VISION
ANALYTICS
I/O
Devices
5
Device 1 (EtherCAT)
Mappings
HandOn
General
Adapter
EtherCAT
Online
CoE - Online
Network Adapter
OS (NDIS)
PCI
DPRAM
Description:
Device Name:
\DEVICE\{7ACDE2F9-7168-4195-AC12-51A297976915}
PCI Bus/Slot:
6
Search...
MAC Address:
IP Address:
Device Found At
7
8
OK
Cancel
Unused
All
Help
Adapter Reference
Adapter:
Freerun Cycle (ms):
Number
Box Name
Address
Type
In Size
Out Size
E-Bus (mA)

5.3.2 Scan for EtherCAT slave devices

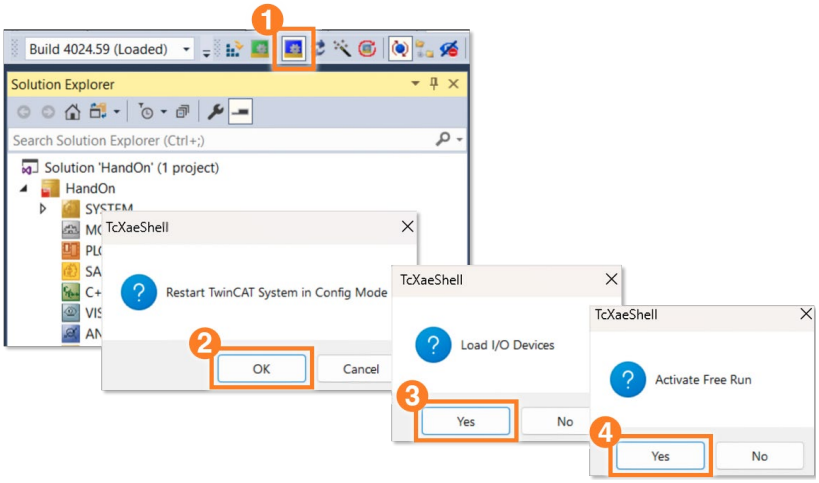
1. In the Solution Explorer, right-click on the EtherCAT master device you created in **I/O** > **Devices**.
2. Select **Scan**.

   This scans for EtherCAT slave devices connected to the device, and adds them under the device in **I/O** > **Devices**.
3. Verify that all EtherCAT slave devices connected to the EtherCAT master have been found and identified.



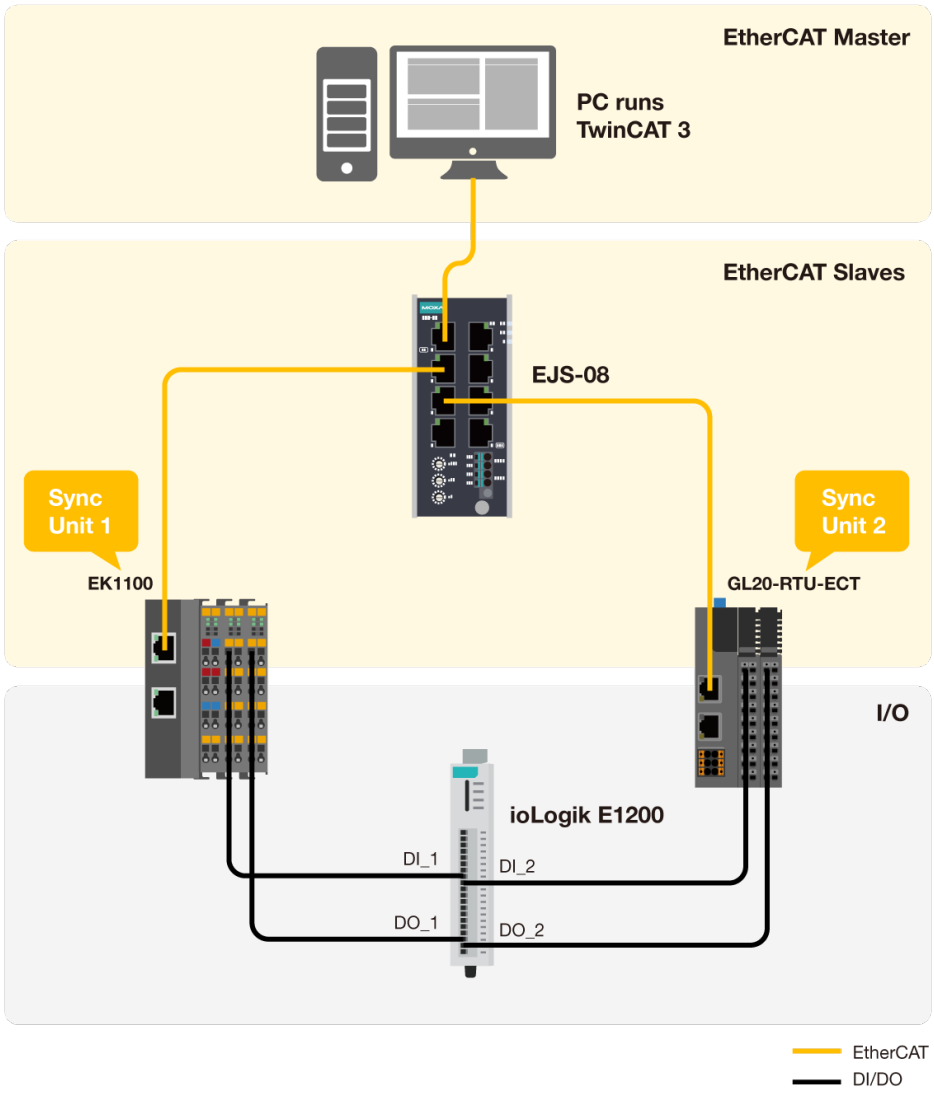
5.3.3 Set TwinCAT to Free Run mode

1. Click on the **Config mode** icon  .
2. In the **Restart TwinCAT System in Config Mode** window, select **Yes**.
3. In the **Load I/O Devices** window, select **Yes**.
4. In the **Activate Free Run** window, select **Yes**.



5.4 Sync Units

Sync Units define independent units of I/O devices. If a fieldbus device fails within a Sync Unit, then all devices of this Sync Unit will be marked as faulty, but **devices in other Sync Units will be unaffected**. This way, if one path fails, device on the other path will remain unaffected. Sync Units should be created if an EtherCAT slave device is expected to fail.



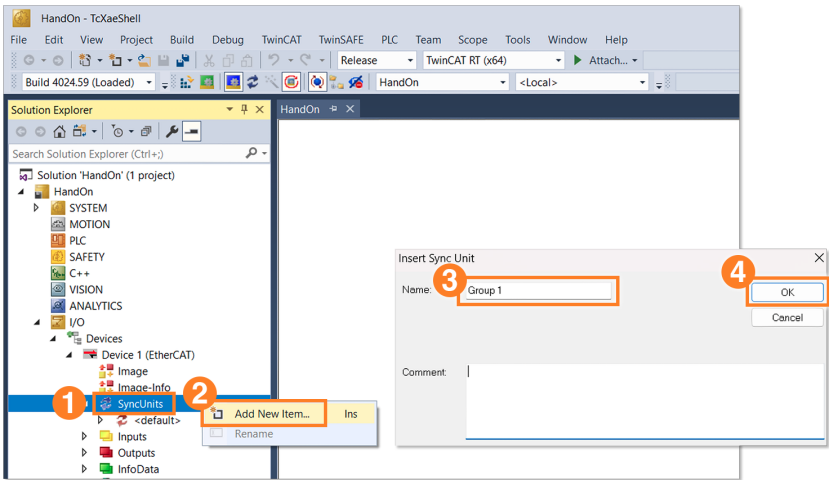
For our topology, we will create two Sync Units:

- **Sync Unit 1**: This will contain the EK1100. Because of its implementation in TwinCAT, we will need to add its EL2008 and EL1008 modules to this Sync Unit.
- **Sync Unit 2**: This will contain the GL20-RTU-ECT.

Note The EK1100 and GL20-RTU-ECT should be put in different Sync Units. Otherwise, the EtherCAT master device will discard all data due to mismatches in the working counter.

5.4.1 Create a Sync Unit

1. In Solution Explorer, right-click **I/O** > **Devices** > *Your EtherCAT master device* > **Sync Units**.
2. Select **Add New Item...**
3. Enter a **Name** for the Sync Unit.
4. Select **OK**.
5. Repeat steps 1 to 3 to create the second Sync Unit.

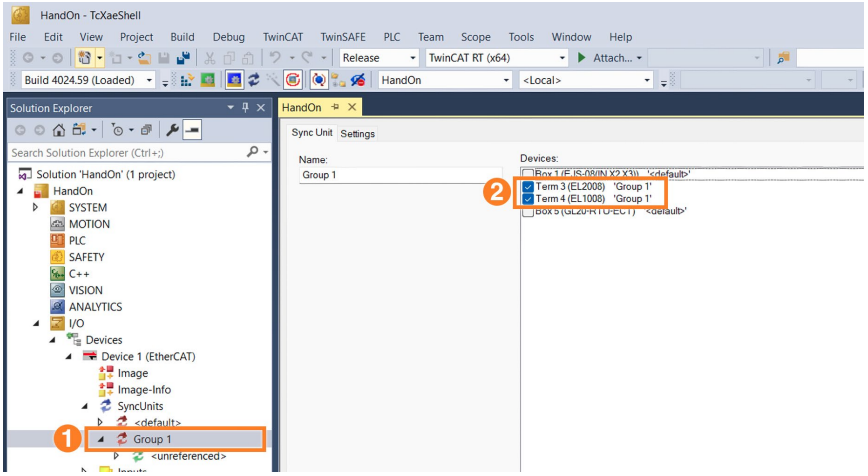


5.4.2 Assign devices to a Sync Unit

1. In Solution Explorer, select **I/O** > **Devices** > *Your EtherCAT master device* > **Sync Units** > *Your Sync Unit*.
2. In the **Sync Unit** tab of the project window, select the devices to add to the Sync Unit.

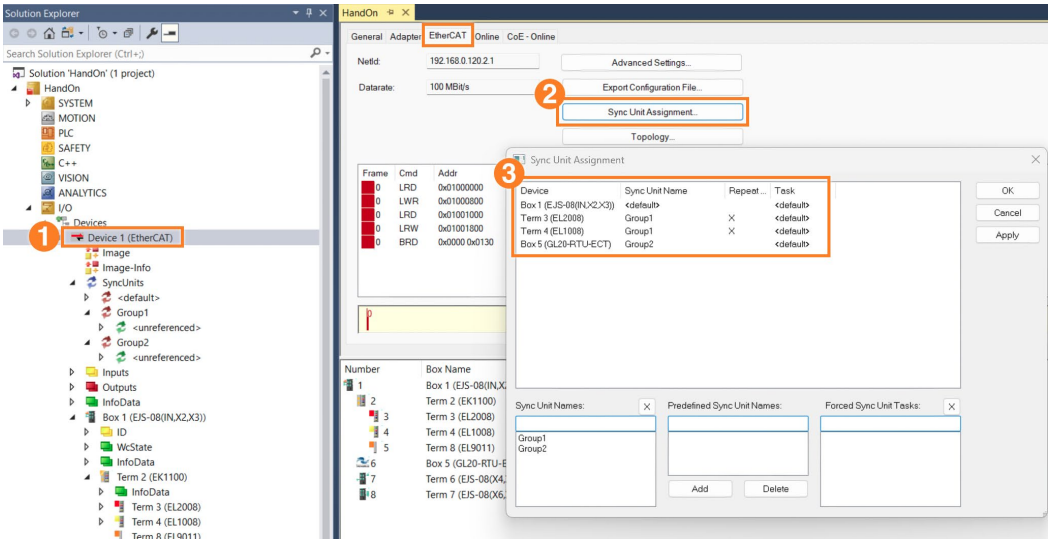
   For **Sync Unit 1**, select the EL2008 and EL1008.

   For **Sync Unit 2**, select the GL20-RTU-ECT.
3. Repeat steps 1 to 2 to assign devices for the second Sync Unit.



5.4.3 Check Sync Unit assignments

1. In Solution Explorer, select **I/O** > **Devices** > *Your EtherCAT master device*.
2. In the **EtherCAT** tab of the project window, select **Sync Unit Assignment...**
3. Verify the devices are assigned to the correct Sync Units.



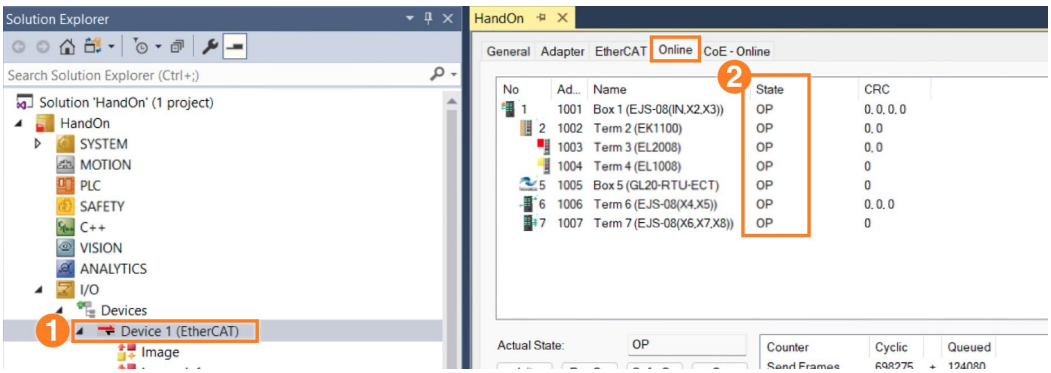
Note After you modify the Sync Unit settings, set TwinCAT to Free Run mode again to make sure the modifications are set successfully. Refer to 5.3.3 for more details.

5.5 Check working status

Check to make sure EtherCAT slave devices work properly after adding them.

1. In **Solution Explorer**, select **I/O** > **Devices** > *Your EtherCAT master device*.
2. In the **Online** tab of your project window, make sure the **State** of all devices is Operational (**OP**).

   If the devices are not in OP state, perform the steps in 5.5.1.

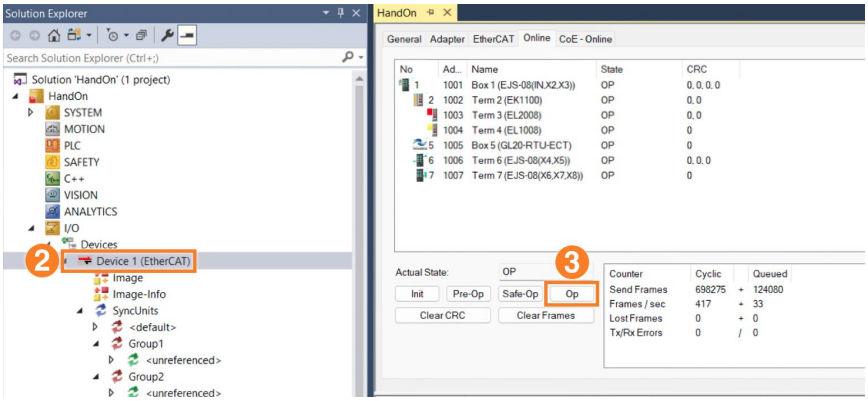


5.5.1 Set devices to OP state

If the devices are not in OP state, perform the following steps:

1. Set TwinCAT to Free Run mode. Refer to 5.3.3 for more details.
2. In **Solution Explorer**, select **I/O** > **Devices** > *Your EtherCAT master device*.
3. In the **Online** tab of the project window, select the **Op** button.

   This will set all devices to OP mode.

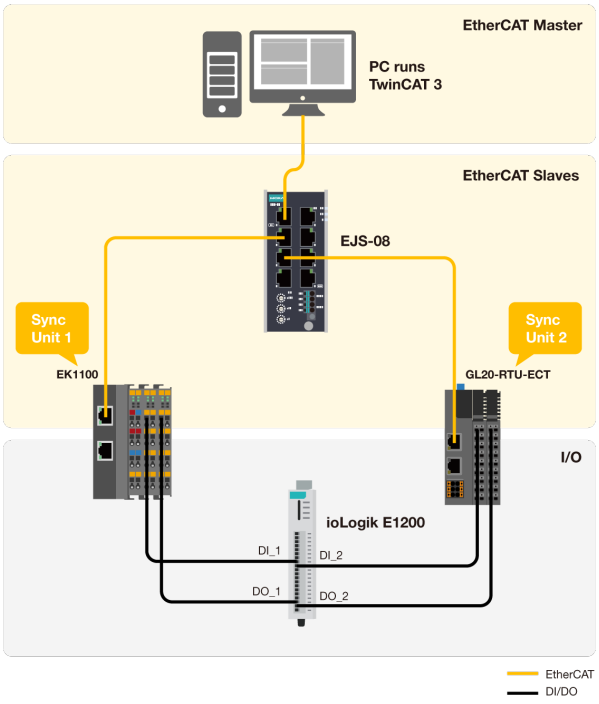


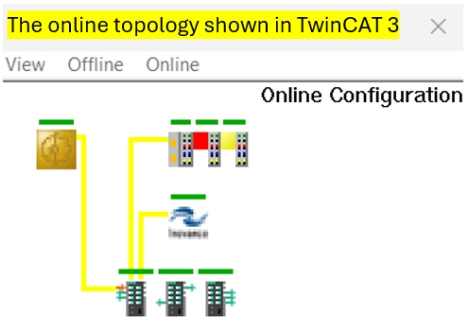
6 Basic Communication Test

In a typical EtherCAT setup, a single disconnection can bring down all downstream devices. However, when using a Moxa EJS, the impact is limited—if one path fails, communication on other paths continues without interruption. This makes it much easier to maintain network stability and isolate faults during testing or real-world failures.

6.1 Verify communication works under normal conditions

To verify communication works, we will test communication from TwinCAT to the target device for both paths, and from the target device back to TwinCAT for both paths.

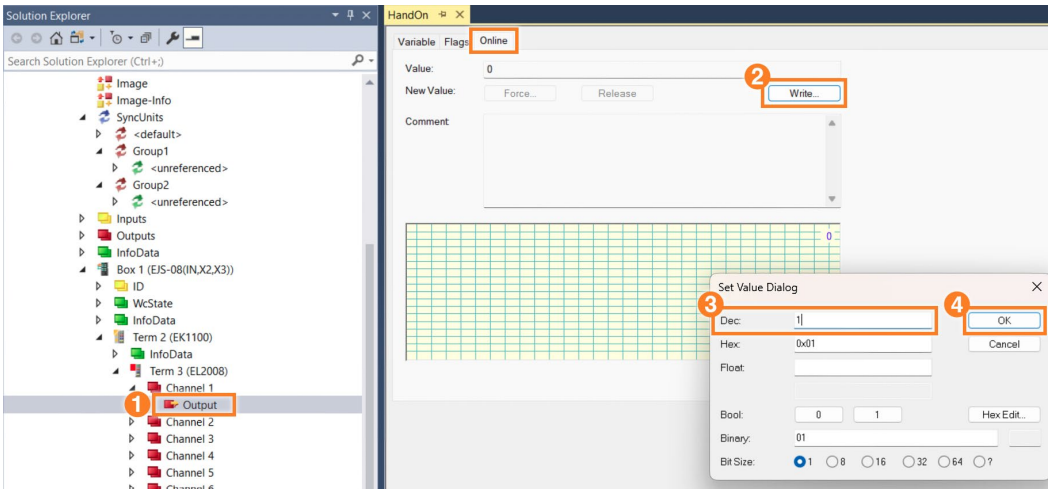


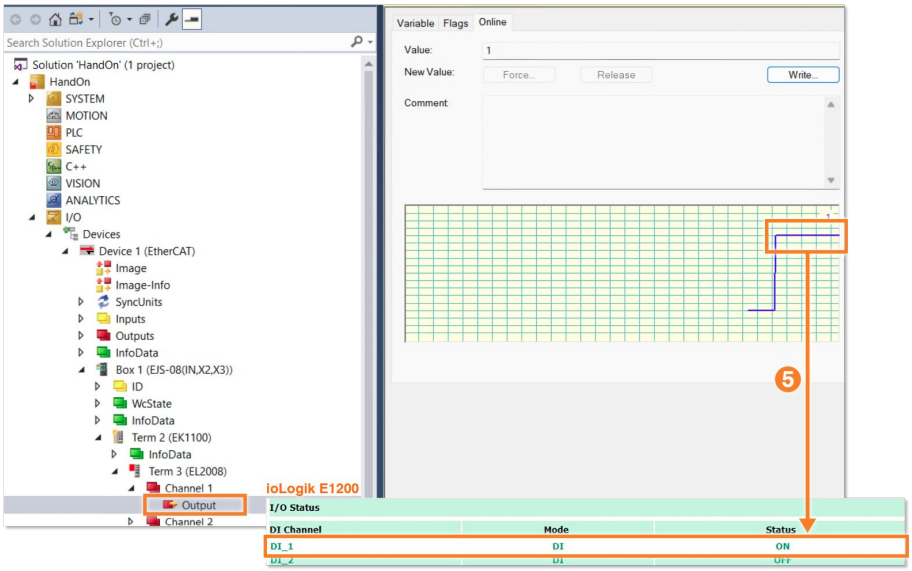


6.1.1 Test communication from TwinCAT to target device

1. In **Solution Explorer**, select **I/O** > **Devices** > *Your EJS device* > *Your EK1100 device* > *Your EL2008 module* > **Channel 1** > **Output**.
2. In your project window, go to the **Online** tab and select **Write...**
3. Enter 1 (ON) or 0 (OFF) to send in **Dec**.
4. Select **OK**.

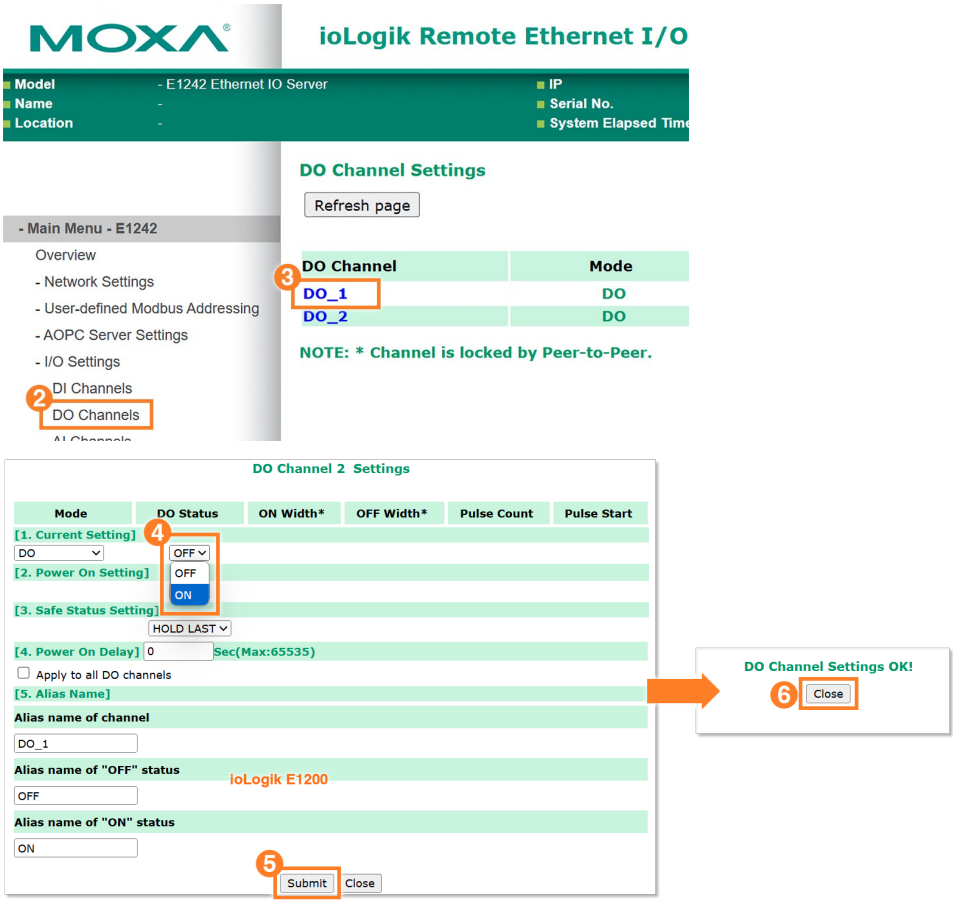
   The graph in your project window's **Online** tab should change accordingly.
5. Log in to the web interface for your ioLogik E1200 and verify that the DI_1 status has changed accordingly.
6. Repeat steps 1 to 5 using **I/O** > **Devices** > *Your GL20-RTU-ECT device* > *Your GL20-0008ETN module* > *Your module's output* for step 1 to test communication through the GL20-RTU-ECT.

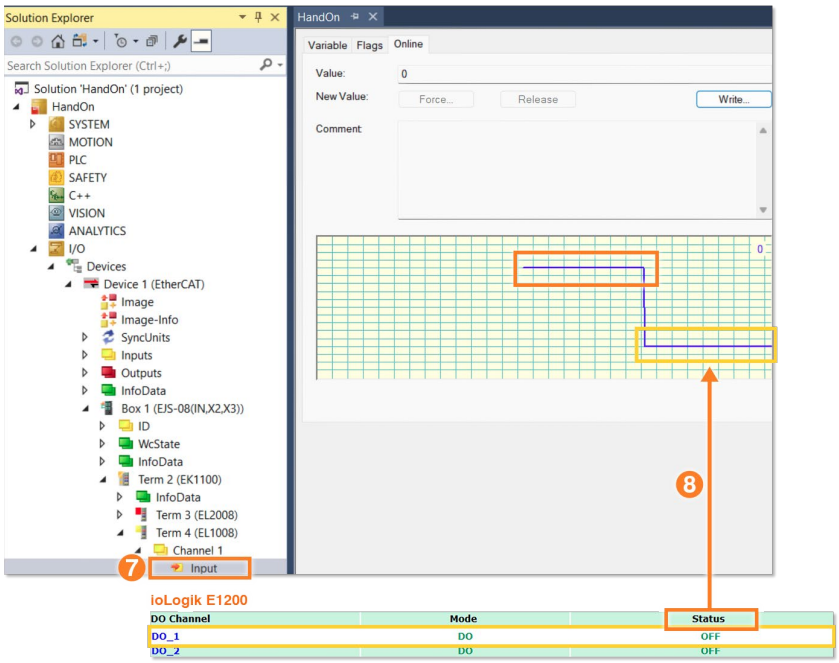




6.1.2 Test communication from the target device to TwinCAT

1. Log in to the web interface for the ioLogik 1200.
2. Go to **Main Menu** > **I/O Settings** > **DO Channels**.
3. Select **DO_1**.
4. For **[1. Current Setting]**, use the drop-down menus to select **OFF** or **ON**.
5. Select **Submit**.
6. Select **Close**.
7. In TwinCAT 3, in the **Solution Explorer**, select **I/O** > **Devices** > *Your EJS device* > *Your EK1100 device* > *Your EL1008 module* > **Channel 1** > **Input**.
8. In the **Online** tab of your project window, verify that the status has changed accordingly.
9. Repeat steps 7 to 8 using **I/O** > **Devices** > *Your GL20-RTU-ECT device* > *Your GL20-0800END module* > *Your module's input* for step 7 to test communication through the GL20-RTU-ECT.





6.2 Verify communication works when there is a single point failure

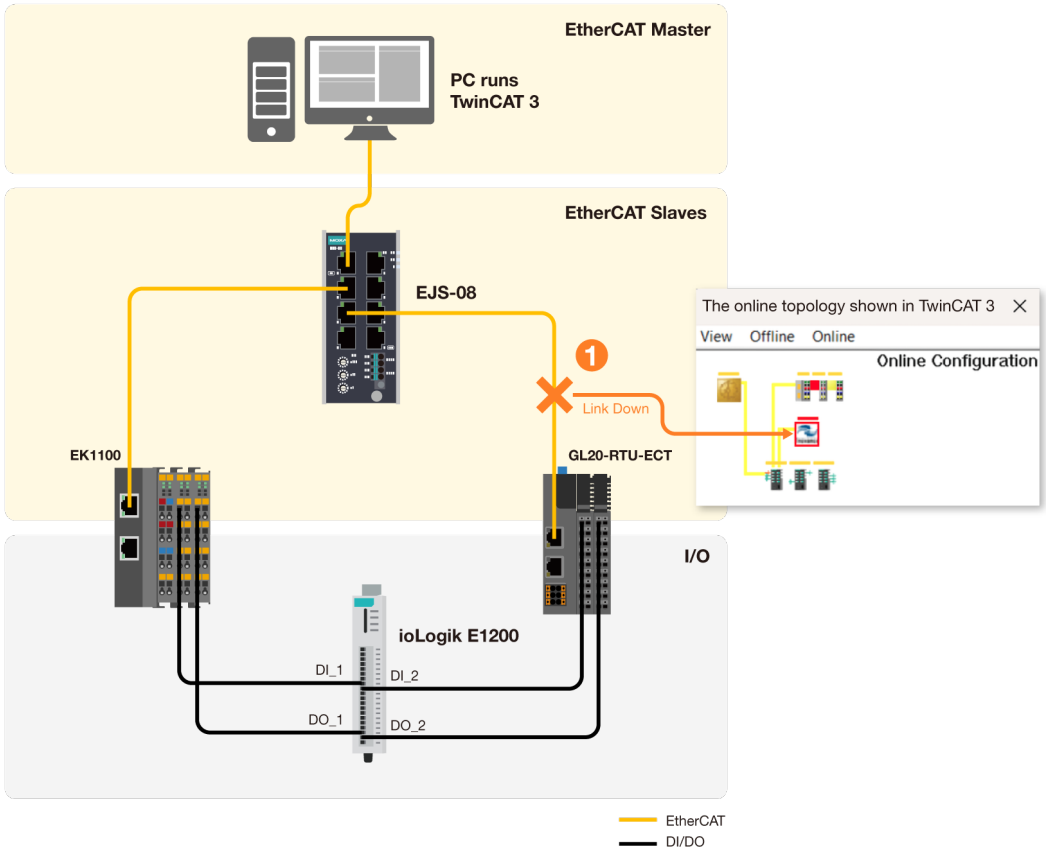
In this part, we will see that the communication between TwinCAT 3 and the EK1100 is not affected even when the other EtherCAT slave device (GL20-RTU-ECT) loses connection.

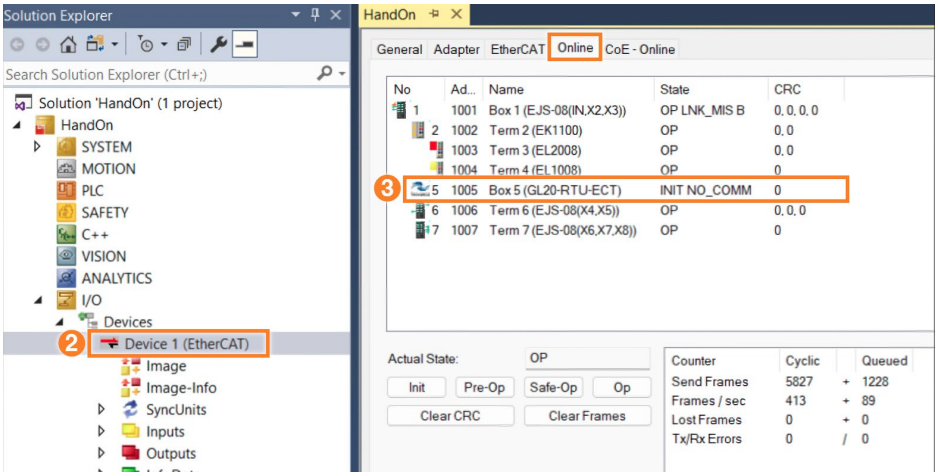
6.2.1 Disconnect the GL20-RTU-ECT

1. Disconnect the cable connecting the EJS-08 to the GL20-RTU-ECT.

   This will act as our single point failure for this test.

2. In **Solution Explorer**, select **I/O** > **Devices** > *Your EtherCAT master device*.
3. In your project window, go to the **Online** tab and verify that the state of the GL20-RTU-ECT device is **INIT NO_COMM**, which indicates it is disconnected from the EtherCAT master device.





6.2.2 Verify communication still works

1. Test communication from TwinCAT to the target device. Refer to 6.1.1 for details.
2. Test communication from the target device to TwinCAT. Refer to 6.1.2 for details.

7 Conclusion

The MOXA EJS Series improves EtherCAT network reliability by separating communication into independent branches. When a device or connection fails, only the affected branch is impacted—other paths of the network continue working normally. This design helps reduce downtime, simplifies troubleshooting, and makes both testing and real-world operation more stable and efficient.