Driving efficiency and sustainability

EtherCAT® - Master

PLC500, PLC500ED, PLC500MC
PLC410

Application Note

Application Note

PLC410, PLC500, PLC500ED, PLC500MC

Document: 10013132619

Revision: 00

Publication Date: 04/2025

The information below describes the reviews made in this manual.

Version	Revision	Description
1.4.3	R00	First edition.

1 INTRODUCTION

This Application Note is intended to assist in the use of the **EtherCAT®** protocol in WEG PLCs, models PLC410, PLC500, PLC500ED, and PLC500MC. Throughout this document, the PLC500 is used as an example. However, the information presented is applicable to the other PLC models described above. It should be noted that the data provided may change slightly due to the continuous development and updating of products and tools.

The PLC500MC model offers SoftMotion functionalities through the EtherCAT protocol, being specially designed to meet more complex motion control applications. For more information about the PLC500MC, as well as advanced EtherCAT protocol configurations, refer to the Application Note available on the WEG website.

In addition to providing an overview of the EtherCAT protocol, this document presents communication interfaces, installation recommendations, network topologies, and a guide to establishing EtherCAT communication using a PLC500 as the master in **CODESYS®**.

For more information regarding hardware, interfaces, and communication protocols, refer to the User Manual of the respective product, available on the WEG website. For a deeper and more detailed description of EtherCAT network operation, access the online help at CODESYS Online Help.

ATTENTION!

This application note is intended for professionals trained in industrial networks. The installation and configuration of devices must be done according to the manufacturer's manual.

NOTE!

It is recommended to use **CODESYS V3.5 SP19** or higher, as well as the latest version of the configuration libraries for EtherCAT.

1.1 TERMS AND DEFINITIONS USED

CNC: technology that uses computers to control machines and tools, providing greater precision and repeatability in manufacturing (Computer Numerical Control).

CODESYS: PLC500 programming software. Programming platform that allows developing, configuring, and monitoring solutions for industrial automation and system integration.

ESC: chip integrated into slave devices compatible with the EtherCAT protocol, allowing real-time processing of protocol frames (*EtherCAT Slave Controller*).

ETG: international organization that promotes the EtherCAT protocol for industrial automation, supporting the development and interoperability between devices from various manufacturers (*EtherCAT Technology Group*).

Jitter: term used to describe the variation in delay or instability in data transmission time in a network or system.

SDO: acyclic communication method used for direct access to a device's object dictionary on the EtherCAT network.

PDO: cyclic communication method used for real-time data exchange between devices on the EtherCAT network.

MAC: unique identifier assigned to network devices to ensure correct communication between them on a local network (*Media Access Control*).

XML: structured data format that organizes information hierarchically and readably. In the context of EtherCAT, XML files describe the properties and configurations of devices on the network (*Extensible Markup Language*).

1.2 THE ETHERCAT PROTOCOL

EtherCAT (**Ether**net for **C**ontrol **A**utomation **T**echnology) is an industrial communication protocol based on Ethernet. The protocol is optimized to ensure high efficiency in data exchange, allowing it to be processed “on the fly” as it passes through devices. This mechanism significantly reduces delays that can be found in other industrial networks.

With support for jitter below 1 µs, EtherCAT is ideal for applications requiring high precision and synchronization, such as motion control systems, CNC machine automation, robotics, and automated production lines. For more information about the protocol, visit the Ethercat Technology Group (ETG) website.

1.3 REFERENCE DOCUMENTS

It is recommended to consult the EtherCAT network-related documents presented in Table 1.1 on page 1-2.

Table 1.1: *Reference Documents.*

Document	Version	Source
EtherCAT Installation Guideline	1.0.4	ETG
EtherCAT Diagnosis for Users	1.0	ETG

1.4 IMPORTANT NOTICE ABOUT CYBERSECURITY AND COMMUNICATIONS

WEG PLCs, models PLC410, PLC500, PLC500ED, and PLC500MC, have the capability to connect and exchange information through networks and communication protocols. Although they have been designed and tested to ensure proper operation with other automation systems using the protocols mentioned in this manual, it is essential that the customer understands the responsibilities associated with information and cybersecurity when using this equipment.

Therefore, it is the customer’s sole responsibility to adopt defense-in-depth strategies and implement policies and measures to ensure the security of the system as a whole, including communications sent and received by the equipment. These measures include, but are not limited to, the installation of firewalls, antivirus and antimalware programs, data encryption, authentication control, and physical access control of users.

WEG and its affiliates are not responsible for damages or losses resulting from cybersecurity breaches, including, but not limited to, unauthorized access, intrusion, leakage and/or theft of data or information, denial of service, or any other form of security breach. The use of this product in conditions for which it was not specifically designed is not recommended and may cause damage to the product, the network, and the automation system.

In this regard, it is imperative that the customer understands that external interventions through third-party programs, such as sniffers or similar programs, have the potential to cause interruptions or restrictions in the functionality of the equipment.

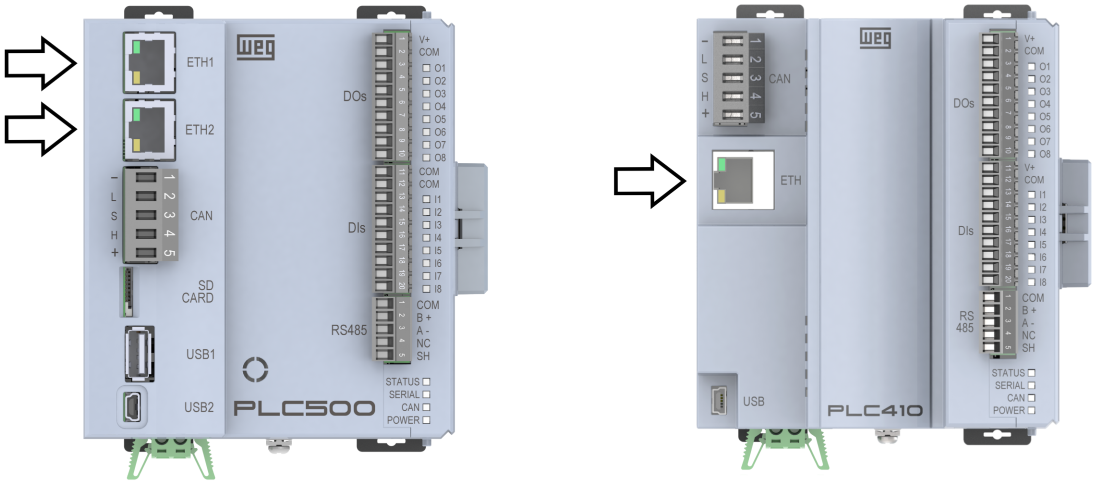
1.5 TRADEMARKS

EtherCAT® is a registered trademark of Beckhoff Automation GmbH.

All other trademarks are the property of their respective owners.

2 ETHERNET INTERFACE

EtherCAT communication is carried out through Ethernet connections, indicated in Figure 2.1 on page 2-1 for PLC500 and PLC410.



(a) (b)

Figure 2.1: Indication of Ethernet connections of PLCs. (a) PLC500 and (b) PLC410.

The pin distribution of the connector follows the Ethernet 1000BASE-TX standard. The Ethernet interface of PLC410 supports speeds up to 100 Mbps, while the Ethernet interfaces of PLC500 reach up to 1000 Mbps.

The interface is compatible with various communication protocols, including EtherCAT. For instructions on how to configure these networks, refer to the product Application Notes available on the WEG website.

2.1 INDICATION LEDS

The Ethernet ports have LEDs for indicating network speed and link/activity, as indicated in Figure 2.2 on page 2-1. These LEDs have the behavior described by Table 2.1 on page 2-1 and Table 2.2 on page 2-1.

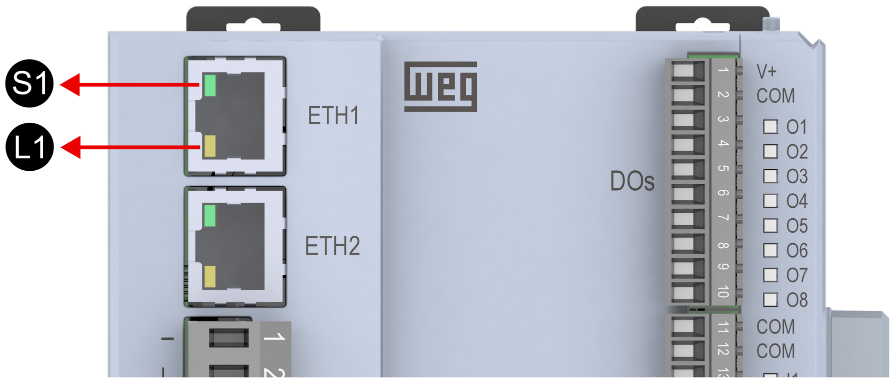


Figure 2.2: Speed and link/activity LEDs of the PLC500 Ethernet interface.

Table 2.1: LED S1 - Speed.

State	Description
Off	Equipment off or 10 Mbps link
Solid green	100 Mbps link

Table 2.2: LED L1 - Link/Activity.

State	Description
Off	Equipment off or no link
Solid amber	With link and no network activity
Blinking amber	With link and network activity

2.2 INSTALLATION OF THE ETHERCAT NETWORK

The EtherCAT network, like many industrial communication networks, is often applied in harsh environments with high exposure to electromagnetic interference, requiring certain precautions to ensure a low communication error rate during its operation.

ATTENTION!
It is recommended to use certified passive components (cables, connectors) for industrial applications.

The recommended characteristics for the cable used in the installation are:

- Standard Ethernet cable, 1000Base-TX, CAT 5e or higher.
- Shielded cable.
- Maximum length of 100 m for connection between devices.

Devices that do not have the ESC integrated can impact network performance, cause frame loss, and even prevent EtherCAT from functioning. Due to the nature of the protocol, non-compatible devices can leave the upstream link active even in the absence of a connection, causing frame loss.

The use of switches in EtherCAT networks is discouraged, as it can negatively impact network performance, including propagation delays, communication jitter, and alteration in the routing order of EtherCAT frames. If it is necessary to use a switch, ensure that there is only one branch of the EtherCAT network, as shown in Figure 2.3 on page 2-2. The use of two branches in a switch that is not specific for managing EtherCAT frames will not work as expected.

ATTENTION!
To ensure reliable network management and advanced diagnostic capabilities, it is recommended to use devices on the network that are compatible with EtherCAT.

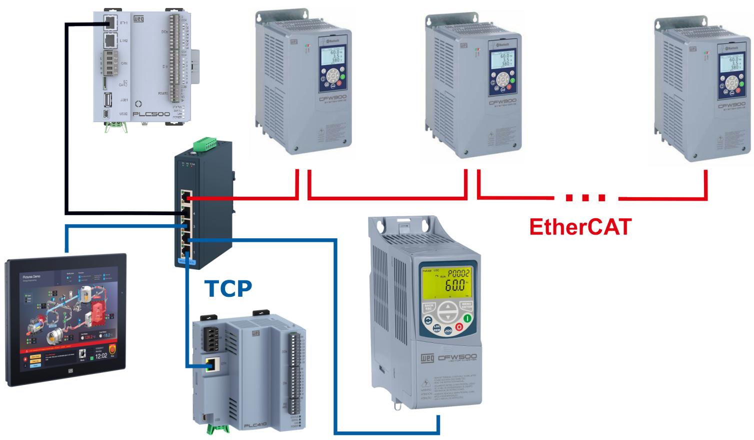


Figure 2.3: *EtherCAT network topology with other TCP protocols on Ethernet port 1 of PLC500 using a switch.*

Proper grounding connection is essential to minimize electromagnetic interference problems in industrial environments. It is important to avoid connecting the cable to multiple grounding points, especially in locations where there are potential differences between the grounding points. Additionally, it is recommended that signal and communication cables be installed in dedicated routes, keeping a distance from power cables.

For detailed information on the correct installation of an EtherCAT network, it is recommended to consult the ETG documents presented in Table 1.1 on page 1-2.

DANGER!

Improper grounding installations can cause EtherCAT network failures and pose a risk of fatal electric shock.

2.3 ETHERCAT NETWORK TOPOLOGIES

Network topologies in an EtherCAT system can vary according to project needs and installation architecture. The protocol has a network scan functionality, being able to automatically detect the installed topology. Notable topologies include the daisy-chain topology, which allows the connection of devices in sequence, as shown in Figure 2.4 on page 2-3; and the ring topology, which allows network communication redundancy in case of a device failure, as shown in Figure 2.5 on page 2-3.

Figure 2.4: *EtherCAT daisy-chain topology.*

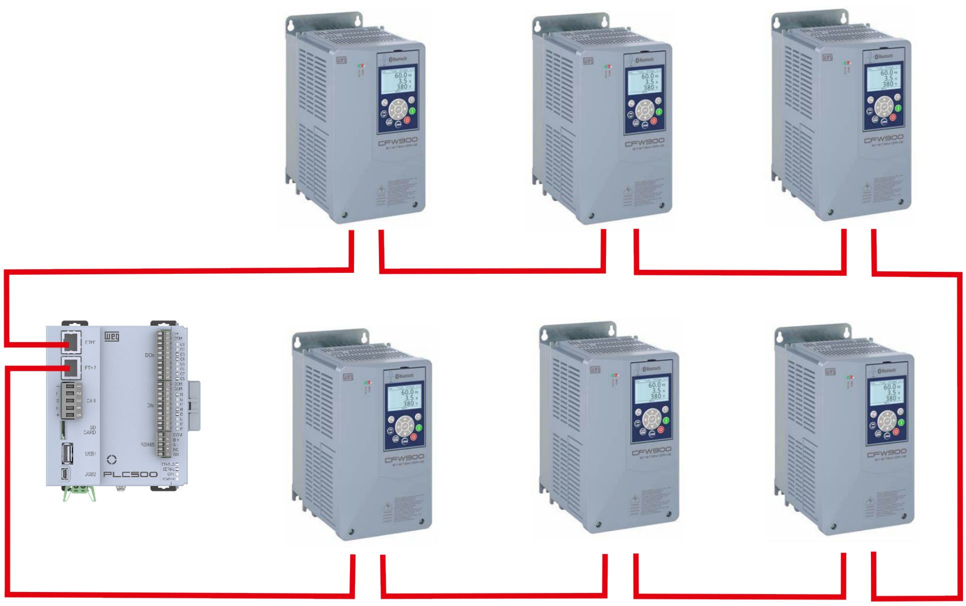


Figure 2.5: *EtherCAT ring topology.*

NOTE!

The PLC410 does not have a second Ethernet interface, so it is not possible to implement the ring topology with this product.

3 PROJECT IN CODESYS

This chapter presents the steps on how to create a project and add the EtherCAT network master device in CODESYS.

3.1 PROJECT CREATION

In the CODESYS software, create a new project using the Standard project template, choose the directory and application name. Then, select the PLC500 device and the desired programming language, as shown in Figure 3.1 on page 3-1.

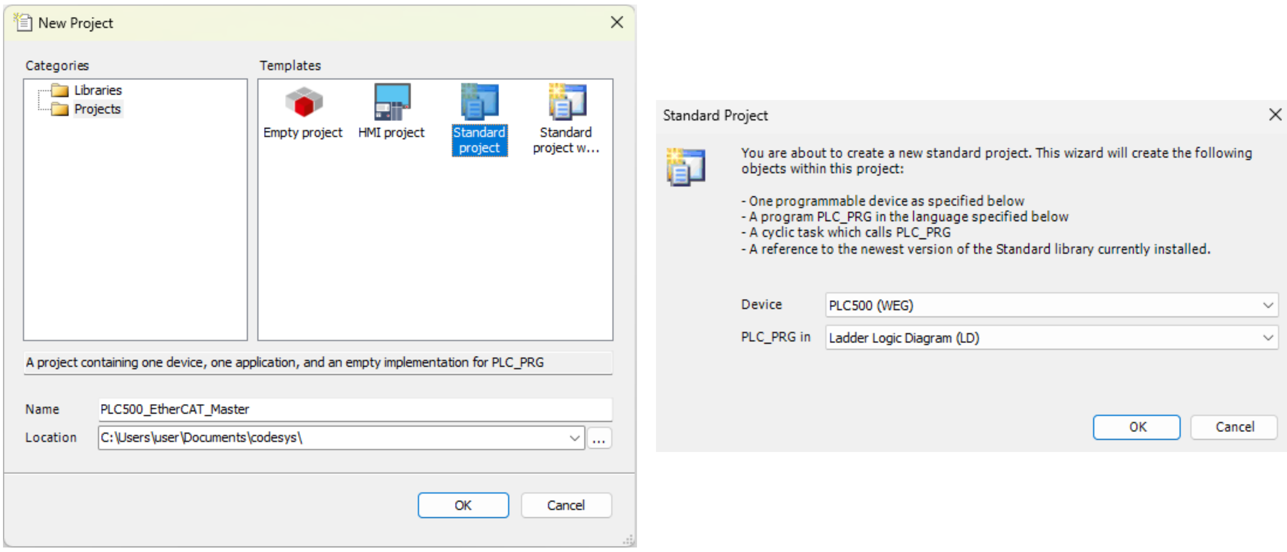


Figure 3.1: *Project configuration in Codesys.*

NOTE!

If the PLC410, PLC500, PLC500ED, or PLC500MC device is not yet available in the options, you should download and install the **WEG package** for CODESYS. Check the **Product Manual** available on the WEG website to find the necessary steps and configurations.

With the PLC500 device selected, the project tree will be displayed, as shown in Figure 3.2 on page 3-1.

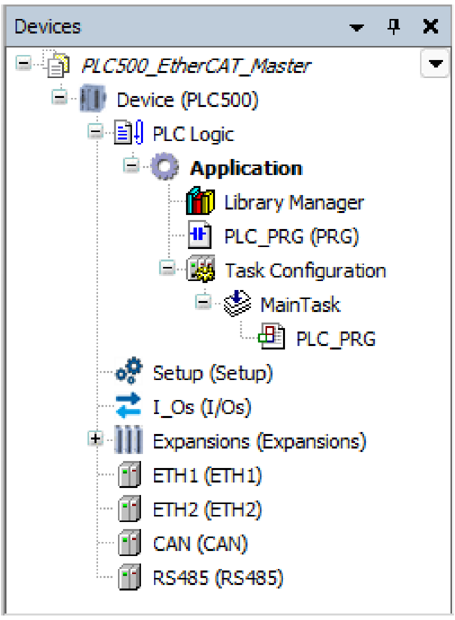


Figure 3.2: *Initial project tree of PLC500.*

3.2 ADDING THE MASTER DEVICE

Right-click on **Device** and select **Add Device**. A new window will open. Expand the **Fieldbuses**, **EtherCAT**, and **Master** icons through the **+** symbol until the **EtherCAT Master** device appears as shown in Figure 3.3 on page 3-2. With the device selected, click on **Add Device**.

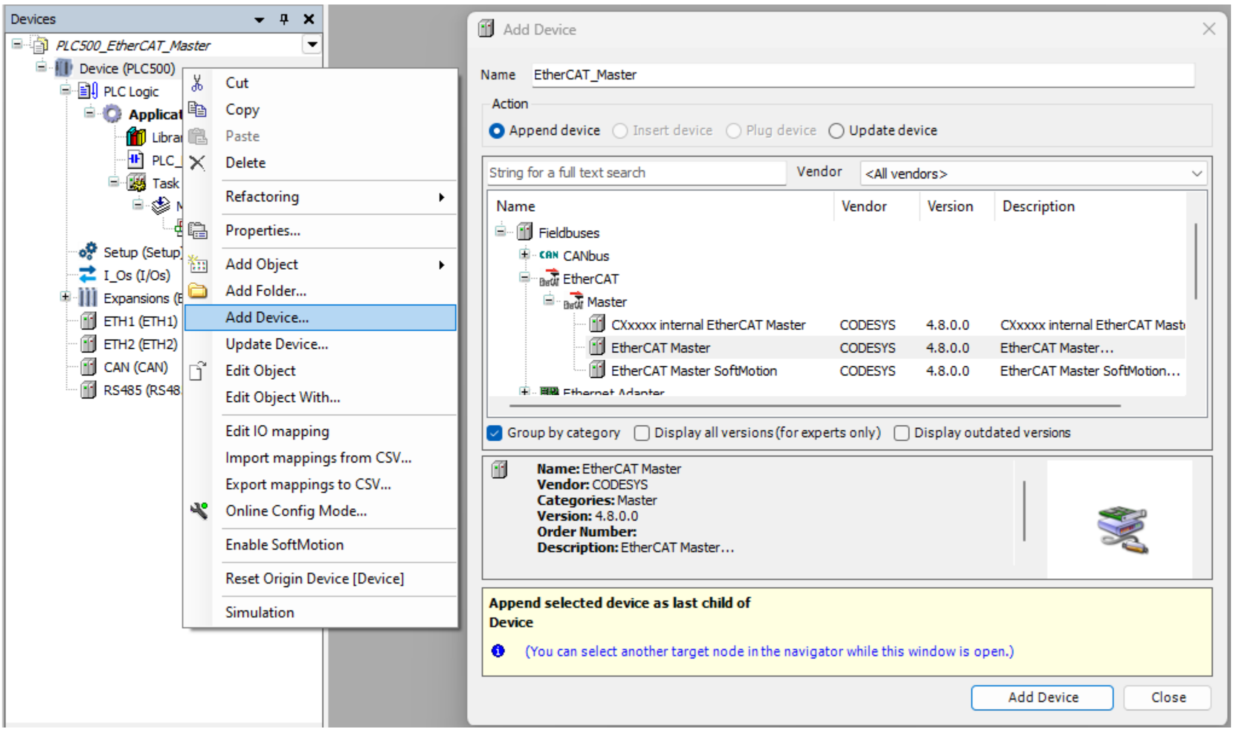


Figure 3.3: *Adding EtherCAT master device.*

The device will be incorporated into the project tree and the **EtherCAT_Task** will be automatically added, as illustrated in Figure 3.4 on page 3-2.

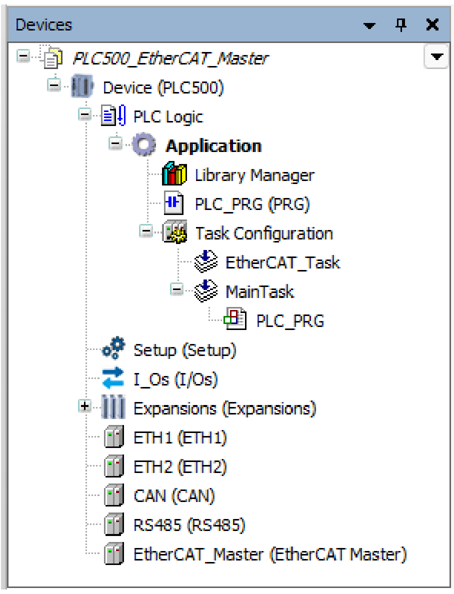


Figure 3.4: *EtherCAT_Master device added to the project tree.*

3.3 ADDING SLAVE DEVICE USING XML FILE

Right-click on the **EtherCAT Master** and select **Add Device**. A new window will open. Devices that already have their XML installed in CODESYS will be displayed. Select the desired device with its corresponding firmware version and click on **Add Device**, as shown in Figure 3.5 on page 3-3.

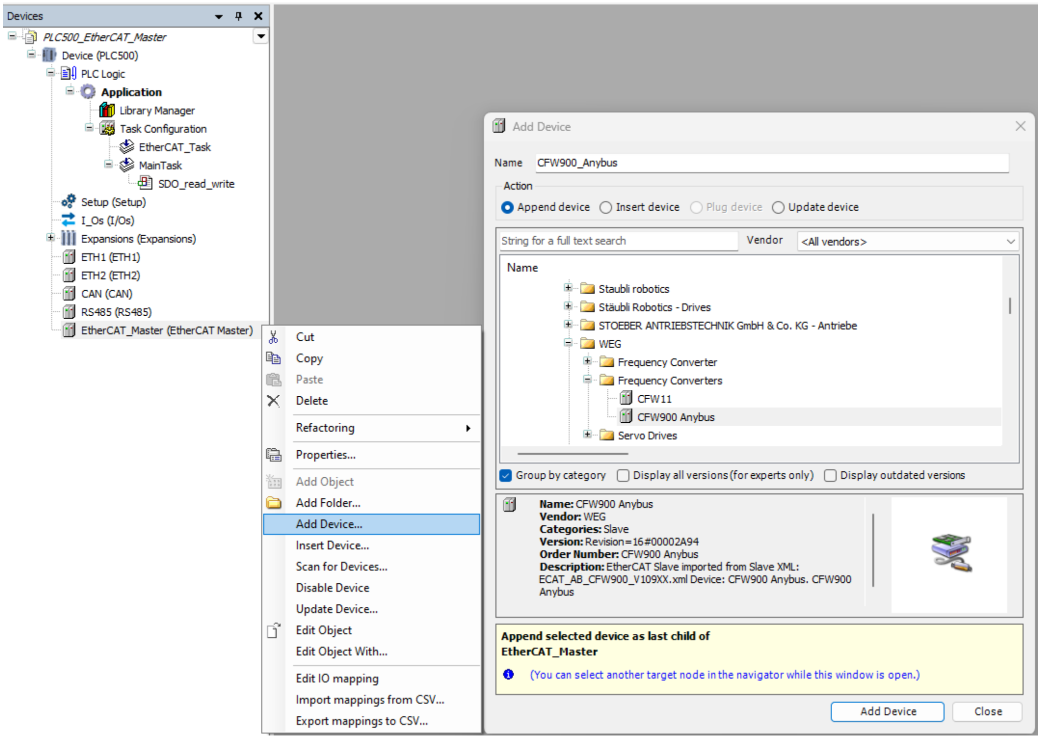


Figure 3.5: Adding EtherCAT slave device using the repository.

If the equipment is not found in the previous window, you must obtain its XML file from the manufacturer and import it into CODESYS. Next, the XML file of the CFW900 will be installed as an example, which is available on the product page on the WEG website. The XML file installation should be done in **Tools** → **Device Repository**, as shown in Figure 3.6 on page 3-3.

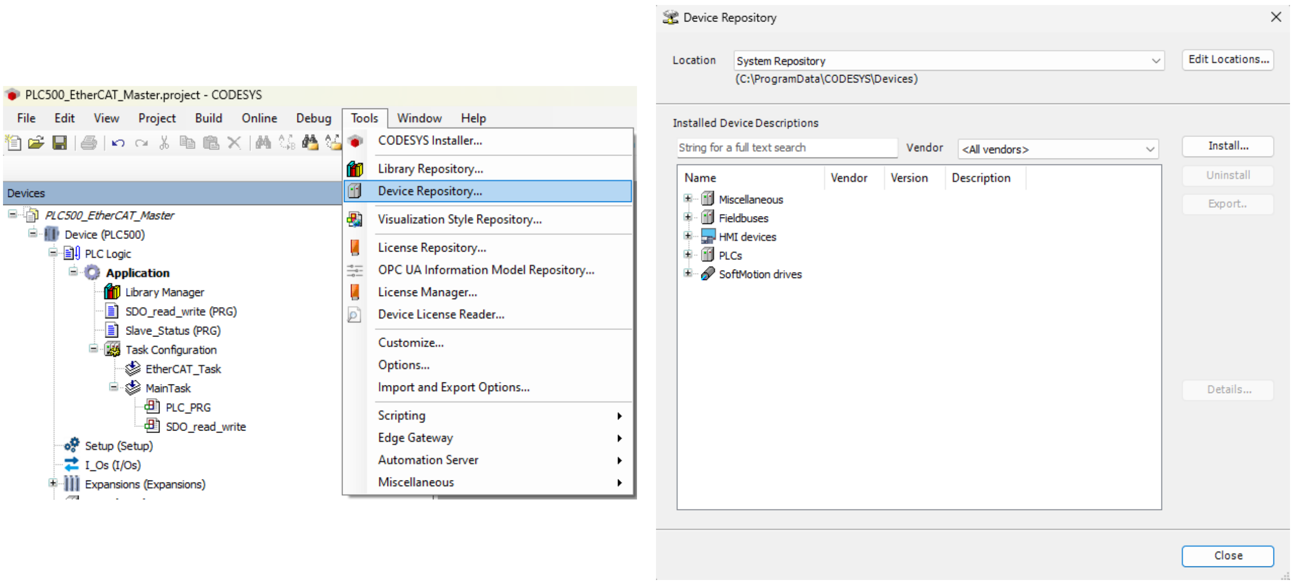


Figure 3.6: Installing new device in Device Repository in CODESYS.

Click on **Install...** and select the XML file, as shown in Figure 3.7 on page 3-4. After selecting the XML, the **Device Repository** window will display a message that the device has been installed and its icon will be shown in the list of devices.

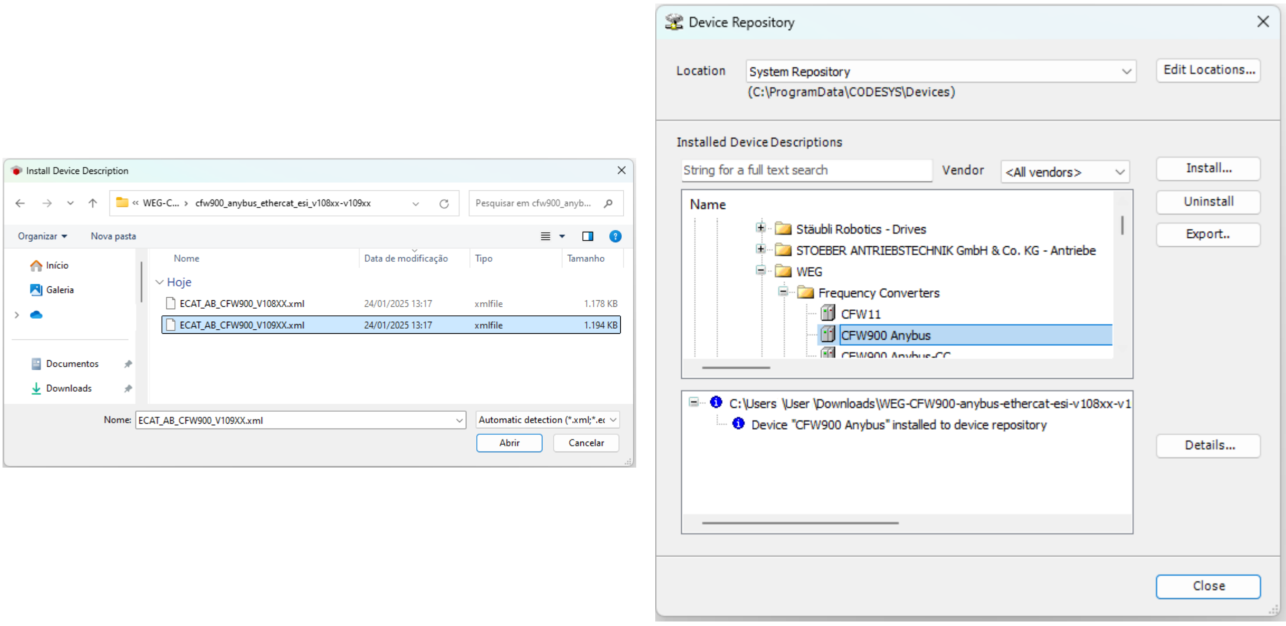


Figure 3.7: Selecting the XML file and successful device installation.

3.4 ADDING SLAVE DEVICE USING NETWORK SCAN

Another way to add an EtherCAT slave device in CODESYS is through network scanning. For this, the device must be configured to operate as an EtherCAT slave and its XML file must be installed in CODESYS.

NOTE!

Depending on the slave device, it may be necessary to perform a pre-configuration for it to operate correctly on the EtherCAT network. For the CFW900, for example, it is necessary to parameterize via HMI to enable communication via Anybus. Refer to the documentation available for your respective device on the manufacturer's website.

Right-click on the **EtherCAT Master** and select **Scan for Devices...**, the devices already configured as EtherCAT slaves and connected via Ethernet will be displayed, as shown in the example demonstrated in Figure 3.8 on page 3-4.

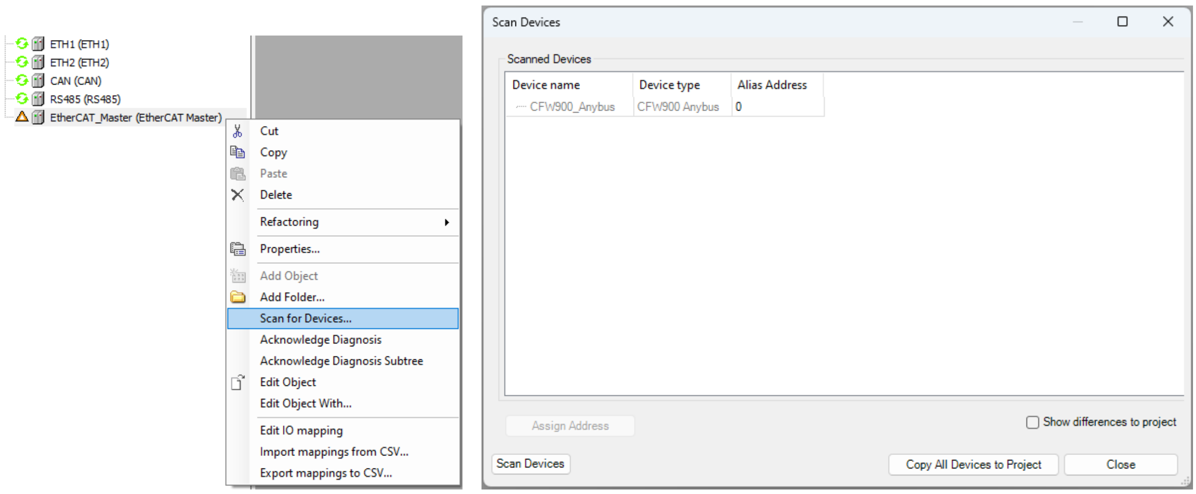


Figure 3.8: EtherCAT slave devices detected through network scan.

> **NOTE!**
>
> The **Scan for Devices** option only works if the PLC500 is already configured as an EtherCAT master. Otherwise, it is necessary to download the program to perform the configuration. Details of the EtherCAT master configuration can be seen in Section 4.1 MASTER DEVICE CONFIGURATION on page 4-1.

To individually add the detected devices, select and click on **Copy to project**, it is also possible to copy all detected devices to the project through **Copy All Devices to Project**.

> **NOTE!**
>
> If the XML file of the detected device is not installed, the message "**Attention! The device was not found in the repository**" will be displayed, and it will not be possible to add it to the project. Obtain the XML file from your manufacturer and follow the installation steps presented earlier.

At this point, the **EtherCAT Master** interface should have the added devices, as shown in the example in Figure 3.9 on page 3-5.

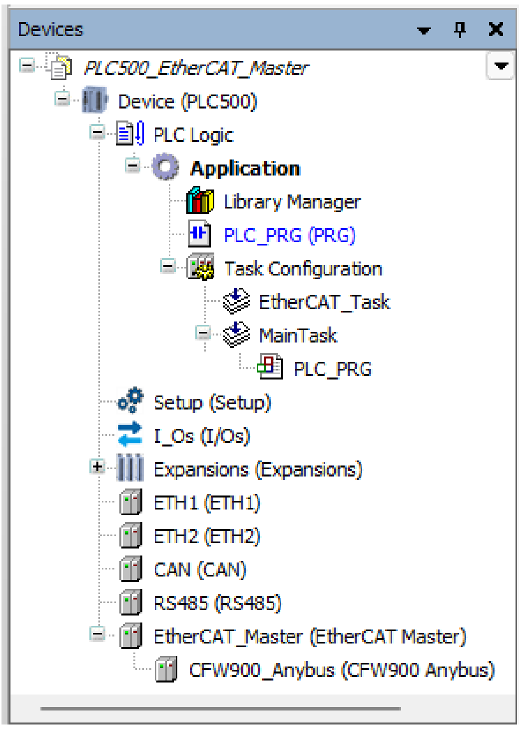


Figure 3.9: *Devices added for EtherCAT communication.*

4 CONFIGURATION OF ETHERCAT NETWORK

This chapter presents the steps on how to configure an EtherCAT network with the PLC500 acting as the master in CODESYS.

4.1 MASTER DEVICE CONFIGURATION

Double-click on the **EtherCAT Master** to open the general configuration window. Select the **Autoconfig Master/Slave** option so that the main master/slave configurations are automatically performed based on the device description file.

Select the **Select network by name** option to configure the Ethernet interface by specifying only the name of the port that will be used for EtherCAT communication. The Ethernet interfaces 1 and 2 of the PLC500 are named **eth1** and **eth2**, respectively. The Ethernet interface of the PLC410 is named **eth1**.

To establish automatic connection with EtherCAT slaves in case of communication loss and restoration, enable the **Automatically restart slaves** option. In Figure 4.1 on page 4-1, the general configurations for the PLC500 as EtherCAT master using the **eth1** port as the main communication port are shown.

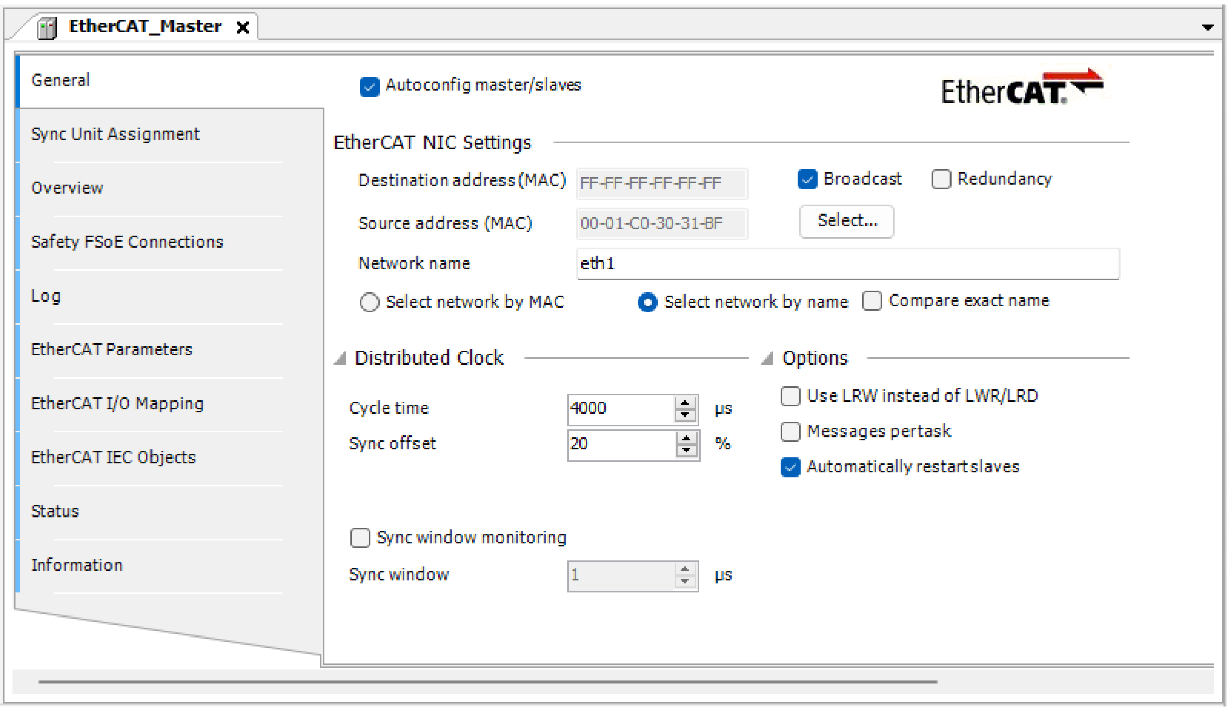


Figure 4.1: *General EtherCAT master configurations.*

NOTE!

For more information on other available configurations for the EtherCAT master, refer to the online help available on the CODESYS Online Help website.

4.2 SLAVE DEVICE CONFIGURATION

Double-click on the slave device added to the project tree to open its general configurations. Depending on the added device, it may be necessary to enable the **Expert settings** option so that the variables in **Process data** can be modified. As an example, the configuration for communication with a CFW900 inverter will be performed.

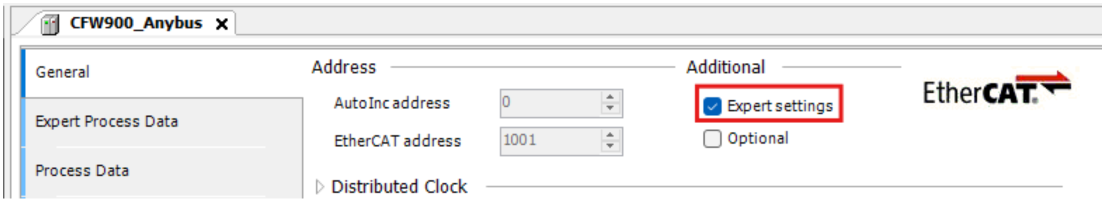


Figure 4.2: *Enabling advanced settings for the EtherCAT slave device.*

The CFW900 comes pre-configured from the factory for reading and writing 2 WORD type data, allowing up to 50 input words and 50 output words for cyclic data. With the **Expert settings** option enabled, click on the **Expert Process Data** window that will appear as an option below **General**.

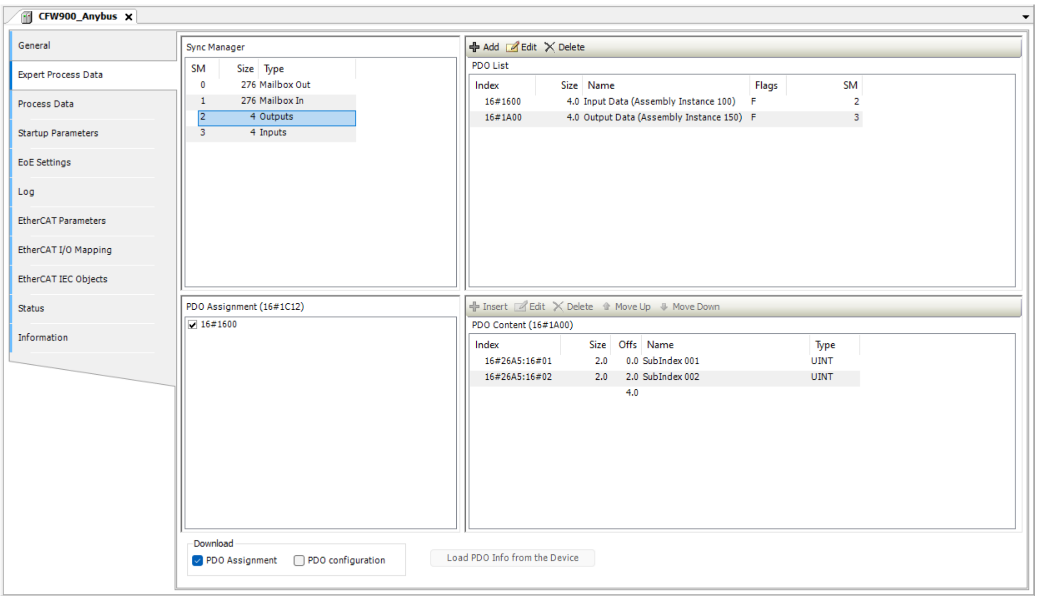


Figure 4.3: *Expert Process Data Window.*

If it is necessary to edit a PDO that has the **F** flag, click on **Edit**, as illustrated in Figure 4.4 on page 4-2. Disable the **Fixed content** option and click **Ok**. Now, it will be possible to edit the number of variables and the names of each PDO.

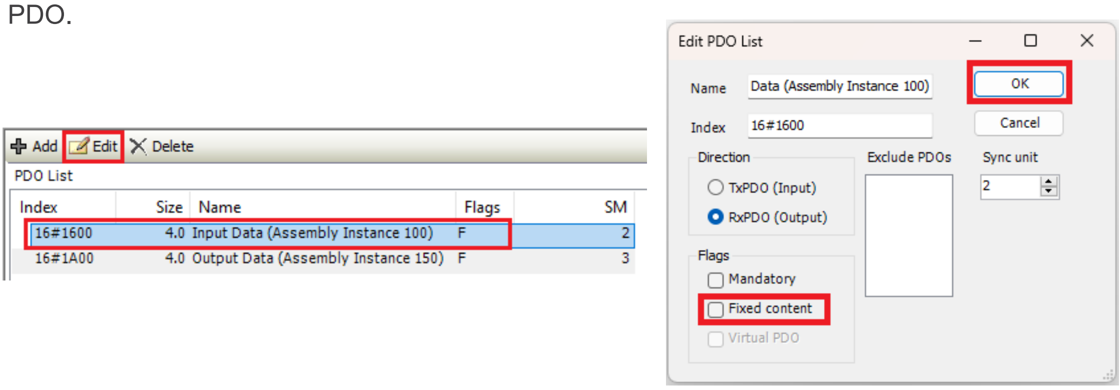


Figure 4.4: *Enabling to edit a fixed PDO.*

To edit the objects within the PLC500 transmission PDO to the CFW900, select **Input Data (Assembly Instance 100)** in the PDO list and click **Edit** in **PDO Content**. In the example, only the names of the objects will be changed for better visualization later, as shown in Figure 4.5 on page 4-3; however, it is possible to add more write words as needed for the application.

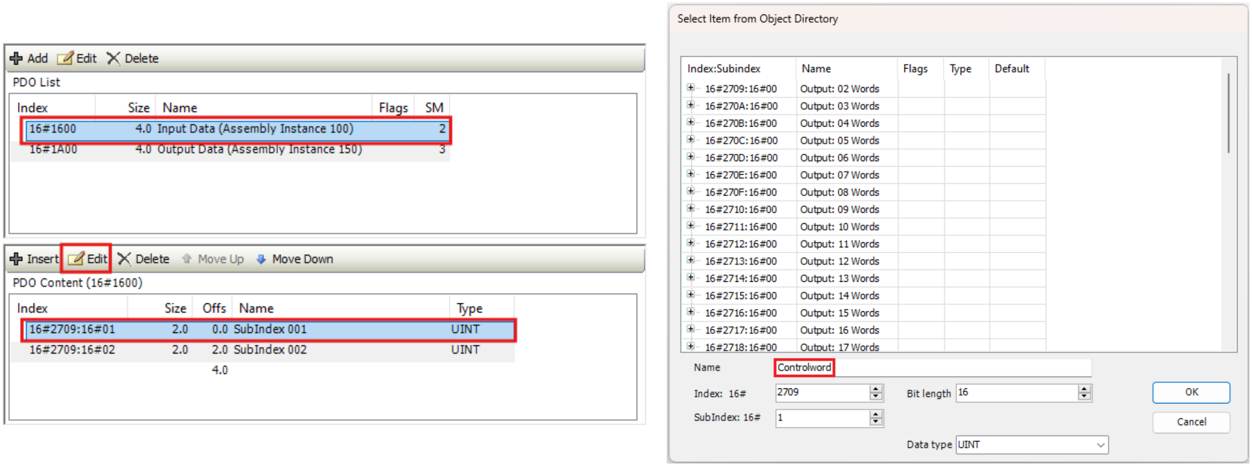


Figure 4.5: *Editing objects within a PDO.*

After editing the **Input Data (Assembly Instance 100)** transmission/write PDOs and **Output Data (Assembly Instance 150)** reception/read PDOs, it is possible to view the mapped objects in the **Process data** tab.

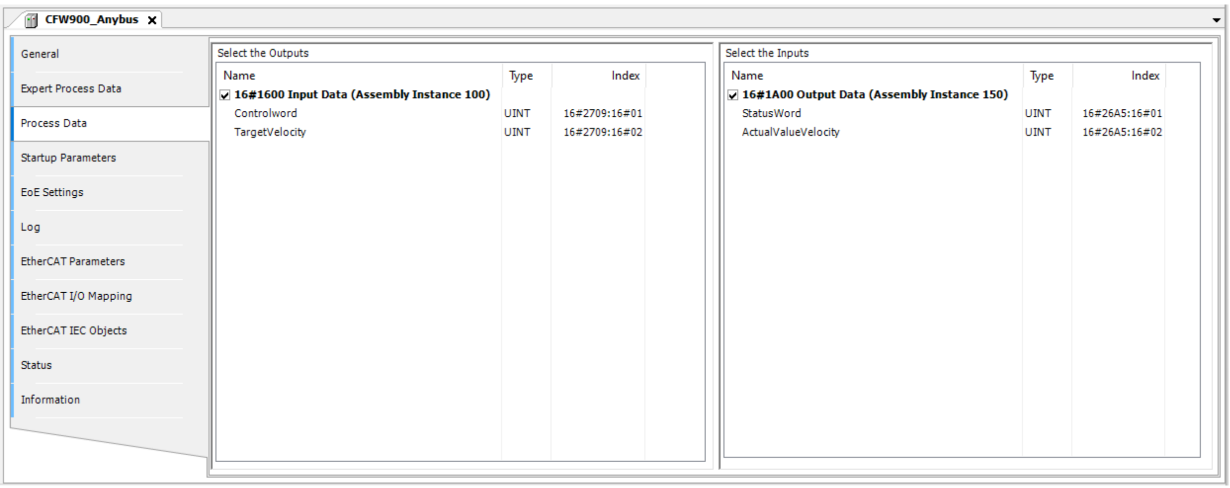


Figure 4.6: *Configured data for transmission and reception PDOs.*

4.3 PARAMETERIZATION THROUGH INITIALIZATION SDOS

Specifically for the CFW900, control and monitoring parameters must be parameterized in the Anybus read and write area. This can be done locally on the inverter's HMI or through SDOs during the initialization of the EtherCAT network.

For parameterization during network initialization, access the **Startup Parameters** tab and click **Add**. In the list, select the object with index 16#2514 and sub-index 16#00 **C9.2.1.1 Reading Data Word #1** and configure the parameter value associated with the variable for reading. In the example in Figure 4.7 on page 4-4, the value **680** of the CFW900 Status Word 1 parameter is configured as the first reading data.

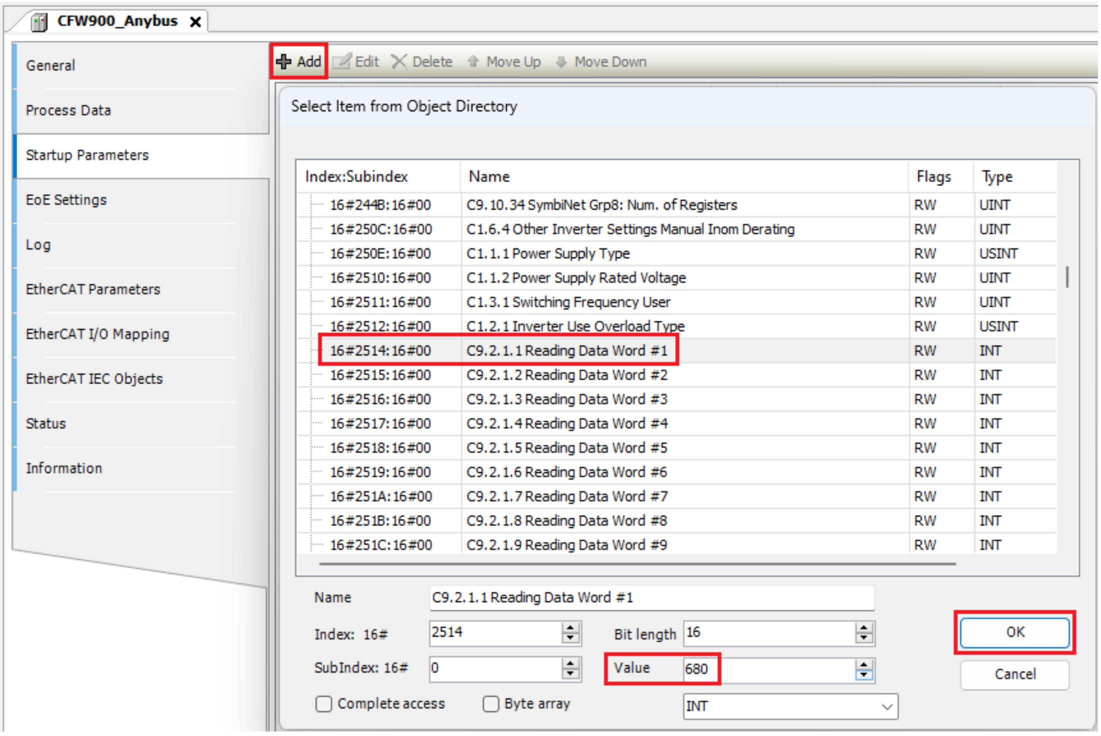


Figure 4.7: *Adding CFW900 startup SDO.*

> **NOTE!**
>
> Parameterization via SDOs through the EtherCAT network may vary depending on the slave device. Refer to the product documentation available on the manufacturer's website.

After configuring the startup SDOs, they will be visible on the page, as illustrated in Figure 4.8 on page 4-4. As an example, the configuration was performed to parameterize, via network, 2 write words (660 → Control Word, 661 → Speed Reference) and 2 read words (680 → Status Word 1, 681 → Speed).

> **NOTE!**
>
> A description of the SDO can be added in the **Comment** field to facilitate identification of the configuration performed.

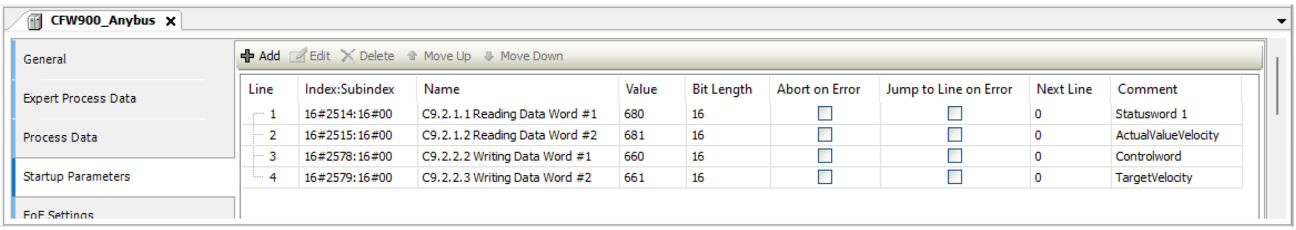


Figure 4.8: *Adding CFW900 startup SDO.*

> **NOTE!**
>
> The CFW900 is pre-configured to read 2 write and read words from the first index. To increase the amount of data exchanged on the EtherCAT network, it is necessary to parameterize the quantity in parameter C9.7.4 for write data and in parameter C9.7.2 for read data. For more information, refer to the Anybus Communication Manual available on the CFW900 page on the WEG website.

4.4 ETHERCAT REDUNDANCY CONFIGURATION

The EtherCAT ring network is a topology that allows communication redundancy between devices, using the two Ethernet interfaces of the PLC500. This configuration ensures that, in case of failure in one of the communication paths, such as a cable or an interface, the other path can take over, maintaining the integrity and continuity of communication.

To configure the redundancy mode, double-click on the **EtherCAT Master** to open the general configuration window. Select the **Redundancy** option; thus, the options for a second configuration interface, **RedundancyEtherCAT NIC Settings**, will be available. The same steps and recommendations used in Section 4.1 MASTER DEVICE CONFIGURATION on page 4-1 for the main Ethernet interface configuration can be followed.

In Figure 4.1 on page 4-1, the general configurations for the PLC500 as EtherCAT master using the **eth1** port as the main communication port and **eth2** for redundancy are shown.

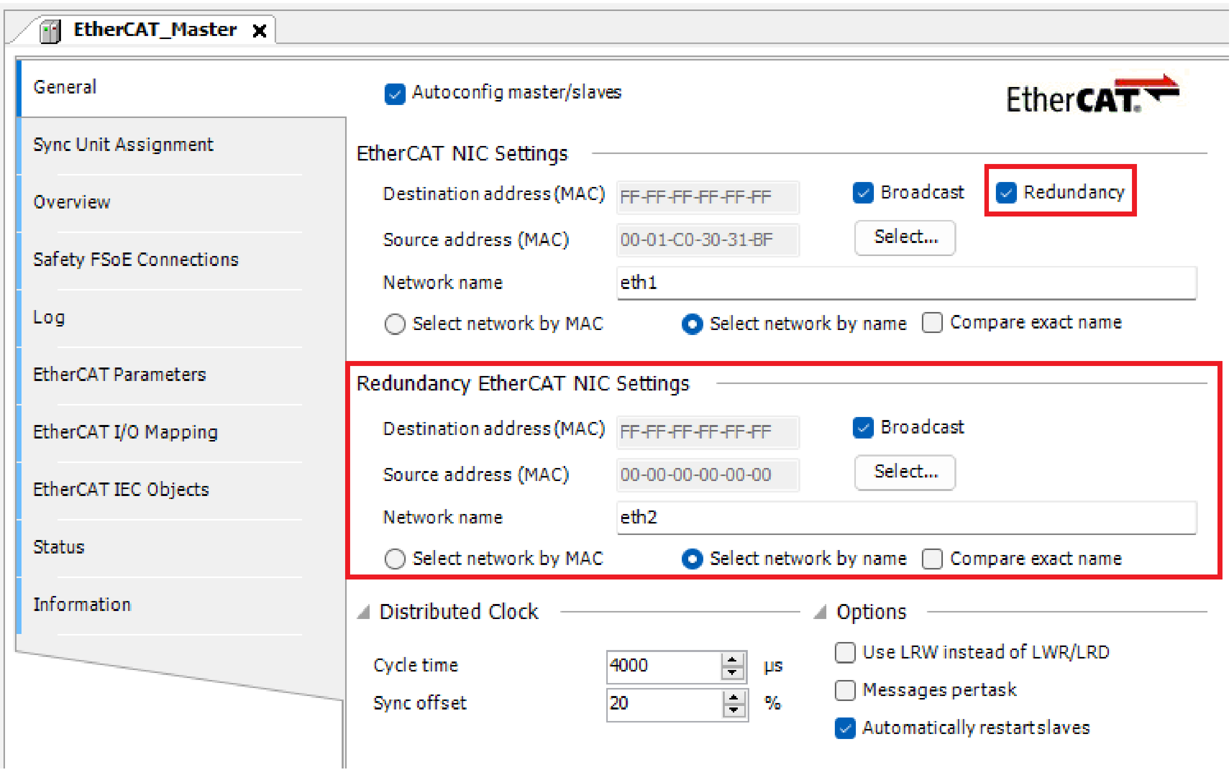


***Figure 4.9:** General EtherCAT master configurations with redundancy.*

> **NOTE!**
>
> The EtherCAT ring network with redundancy has additional fault diagnosis functionalities in CODESYS. Check Section 6.3 ETHERCAT NETWORK REDUNDANCY DIAGNOSTICS on page 6-4 for more information.

5 ACYCLIC ETHERCAT COMMUNICATION

The **EtherCATStack** library in CODESYS offers function blocks to perform acyclic communication using SDOs. These function blocks can be used for configuring slaves during the application runtime. If device configuration is only needed during initialization, use the **Startup Parameters** configuration page, explained in detail in Section 4.3 PARAMETERIZATION THROUGH INITIALIZATION SDOS on page 4-3. Below, structured text code examples using a CFW900 as an EtherCAT slave device are presented.

> **NOTE!**
>
> Another EtherCAT slave device can be used for acyclic communication with SDOs. Refer to the product documentation for information on the object dictionary available for this type of communication.

To read an object from the CFW900, the **ETC_CO_SdoRead** block can be used. The device provides several parameters that can be read via SDO. Figure 5.1 on page 5-1 presents the instance and variable declaration, while Figure 5.2 on page 5-1 shows the structured text code for the SDO_READ_ETHCAT_ST application.

SDO_READ_ETHCAT_ST

```
PROGRAM SDO_READ_ETHCAT_ST
VAR
    EtherCAT_SDO_Read : ETC_CO_SdoRead;
    execute_read : BOOL;
    data_read : WORD;
END_VAR
```

Figure 5.1: *Variable declaration for SDO_READ_ETHCAT_ST program.*

SDO_READ_ETHCAT_ST - Structured text (ST)

```
EtherCAT_SDO_Read (
    xExecute:= execute_read,
    xAbort:=,
    usiCom:=1,
    uiDevice:=CFW900_Anybus.SlaveAddr, // <name of slave device in project tree>.SlaveAddr
    usiChannel:=1,
    wIndex:= 16#257A, // Write Word #3 Anybus I/O
    bySubindex:= 0,
    udiTimeOut:= 500, // Timeout in ms
    pBuffer:= ADR(data_read ), // Variable type must match parameter size
    szSize:= SIZEOF(data_read ),
    xDone=> ,
    xBusy=> ,
    xError=> ,
    eError=> ,
    udiSdoAbort=> ,
    szDataRead=> );
```

Figure 5.2: *SDO_READ_ETHCAT_ST program in structured text.*

To write to an object of the CFW900, the **ETC_CO_SdoWrite** block can be used. Figure 5.3 on page 5-1 presents the instance and variable declaration, while Figure 5.4 on page 5-2 shows the structured text code for the SDO_WRITE_ETHCAT_ST application.

SDO_WRITE_ETHCAT_ST

```
PROGRAM SDO_WRITE_ETHCAT_ST
VAR
    EtherCAT_SDO_Write : ETC_CO_SdoWrite;
    execute_write : BOOL;
    data_write : WORD := 100;
END_VAR
```

Figure 5.3: *Variable declaration for SDO_WRITE_ETHCAT_ST program.*

SDO_WRITE_ETHCAT_ST - Structured text (ST)

```
EtherCAT_SDO_Write (
    xExecute:= execute_write,
    xAbort:=,
    usiCom:=1,
    uiDevice:=CFW900_Anybus.SlaveAddr, // <name of slave device in project tree>.SlaveAddr
    usiChannel:=1,
    wIndex:= 16#257A, // Write Word #3 Anybus I/O
    bySubindex:= 0,
    udiTimeOut:= 500, // Timeout in ms
    pBuffer:= ADR(data_write ), // Variable type must match parameter size
    szSize:= SIZEOF(data_write ),
    eMode:= 0, // 0: ETC_CO_AUTO automatic mode
    xDone=> ,
    xBusy=> ,
    xError=> ,
    eError=> ,
    udiSdoAbort=> ,
    szDataWritten=> );
```

Figure 5.4: *SDO_WRITE_ETHCAT_ST program in structured text.*

NOTE!

For more information about the **EtherCATStack** library, refer to the documentation available in the **Library Manager** in CODESYS or access the online help on the CODESYS Online Help website.

6 MONITORING

This chapter presents some methods for monitoring EtherCAT network variables and diagnosing communication errors in CODESYS.

6.1 VARIABLE MONITORING

Variables declared in the network can be monitored through two methods. The first possibility is to add them to the CODESYS program and monitor the variable values online. The second method requires only enabling **Always Update Variables** at the bottom of the **EtherCAT I/O Mapping** tab, as indicated by the red arrow in Figure 6.1 on page 6-1.

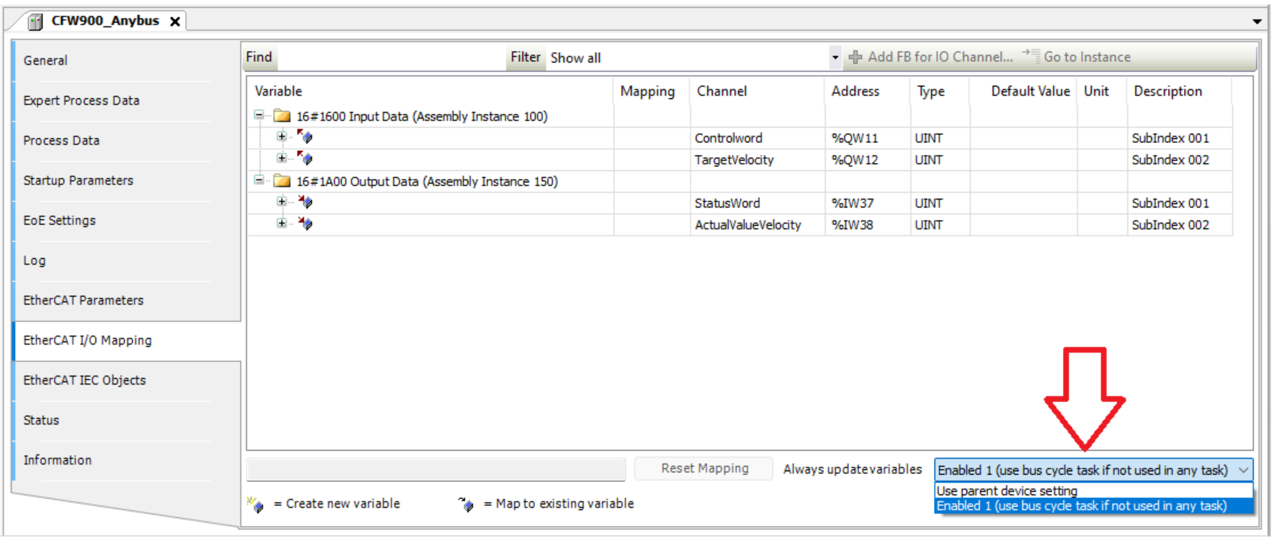


Figure 6.1: *Option to always update EtherCAT variable values with bus cycle task.*

Figure 6.2 on page 6-1 shows the CODESYS variable monitoring screen for the EtherCAT slave through online monitoring.

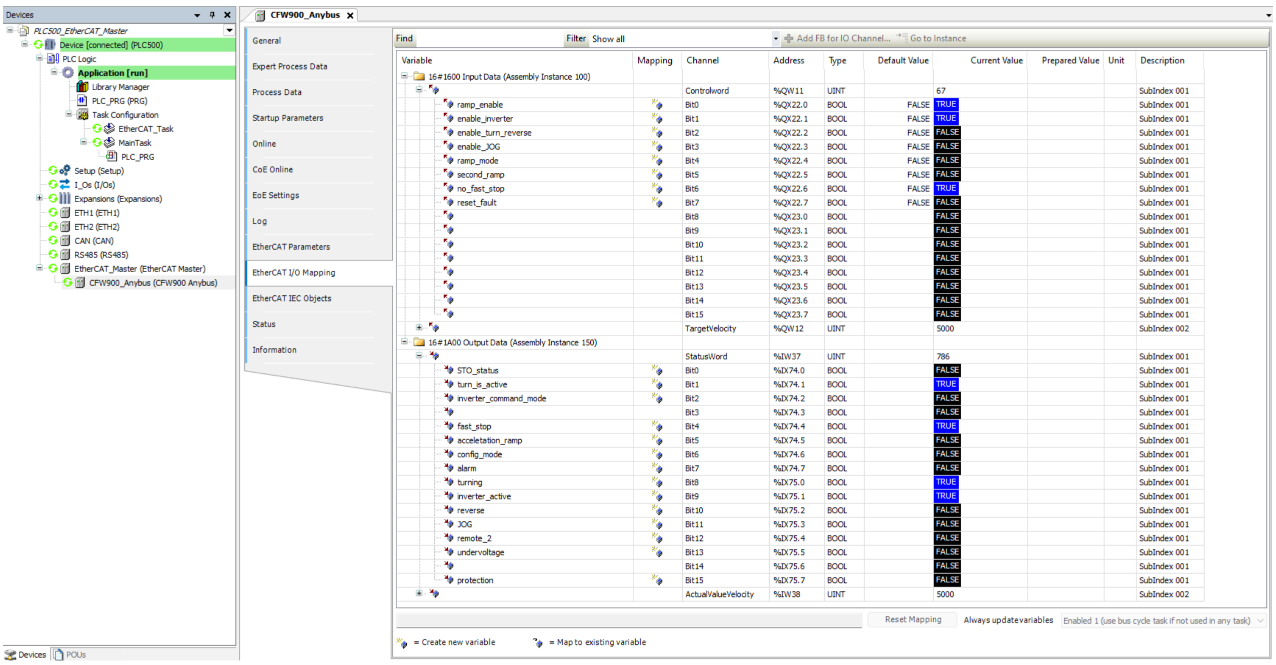


Figure 6.2: *Monitoring EtherCAT variables online in CODESYS.*

6.2 COMMUNICATION ERRORS

Network state monitoring can be done in **Devices**, indicating the state of each communication stage and reporting the status (**Status**). When encountering connection problems, as shown in Figure 6.3 on page 6-2, check if the cables are properly connected, the state of the Ethernet port LED used in the PLC, and then review the configurations performed.

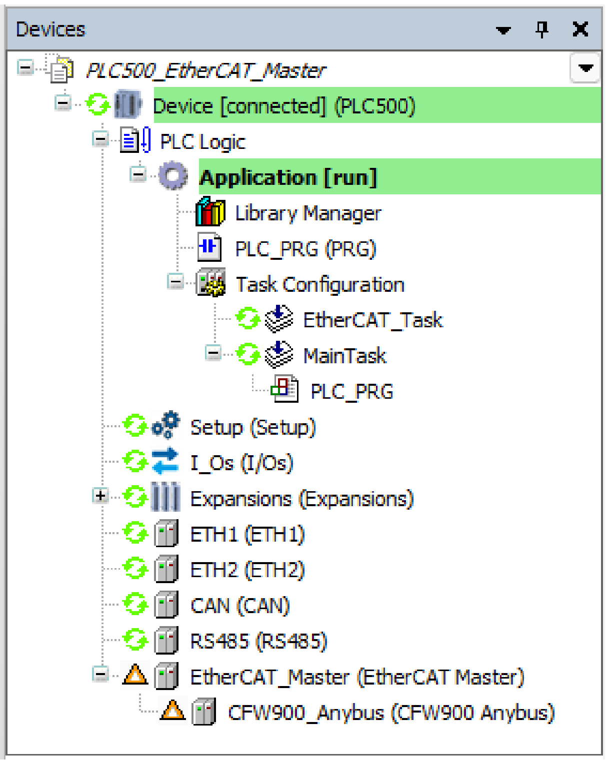


Figure 6.3: *Indication of communication failures.*

In online mode, access the **Status** tab of the **EtherCAT_Master** as shown in Figure 6.4 on page 6-2. In this location, CODESYS will inform what problem it is encountering to perform the communication. In this case, the cable was not well connected to the slave device.

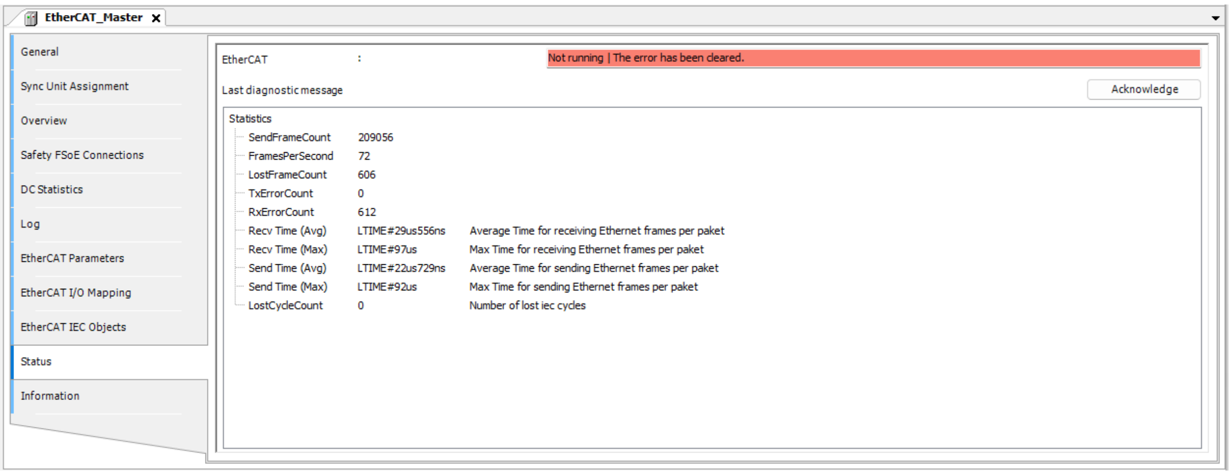


Figure 6.4: *Status tab of EtherCAT master device.*

By reconnecting the cable and waiting a few moments, the network communication is restored and the device icons turn green. An exclamation mark icon is shown on the device indicating that an error has been declared, as shown in Figure 6.5 on page 6-3.

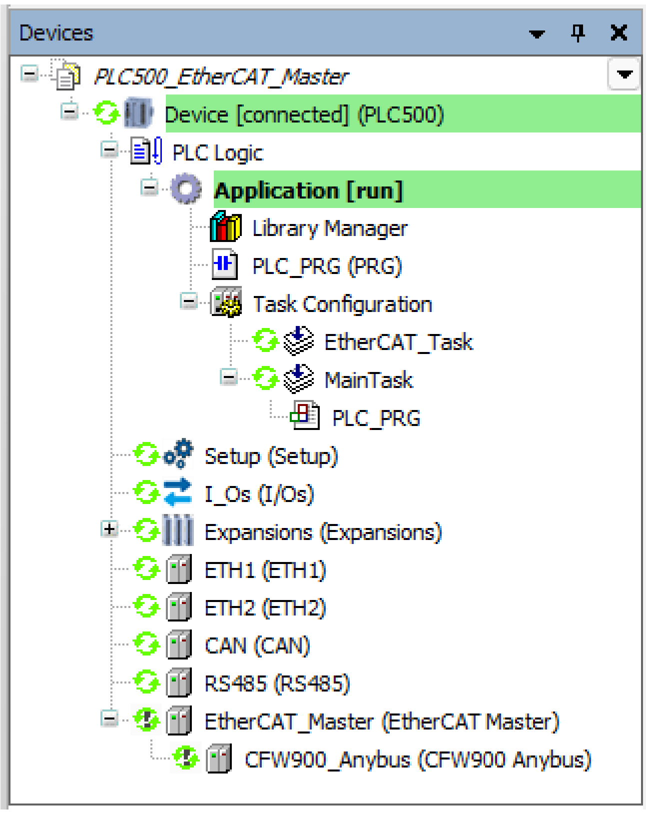


Figure 6.5: *Indication of declared error in the EtherCAT network.*

To remove the warning, simply click **Acknowledge**, as indicated in Figure 6.6 on page 6-3.

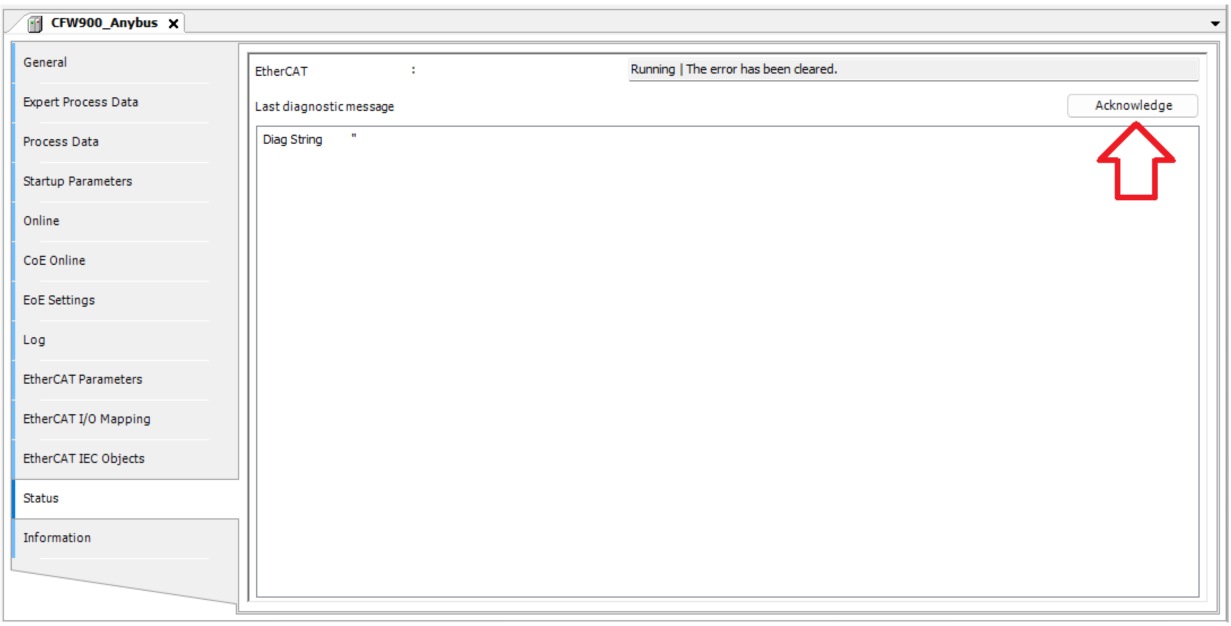


Figure 6.6: *Acknowledging declared error in the EtherCAT network.*

After acknowledging the warning also in the **EtherCAT Master** and verifying the proper functioning of the network, the device icons will be displayed as illustrated in Figure 6.7 on page 6-4.

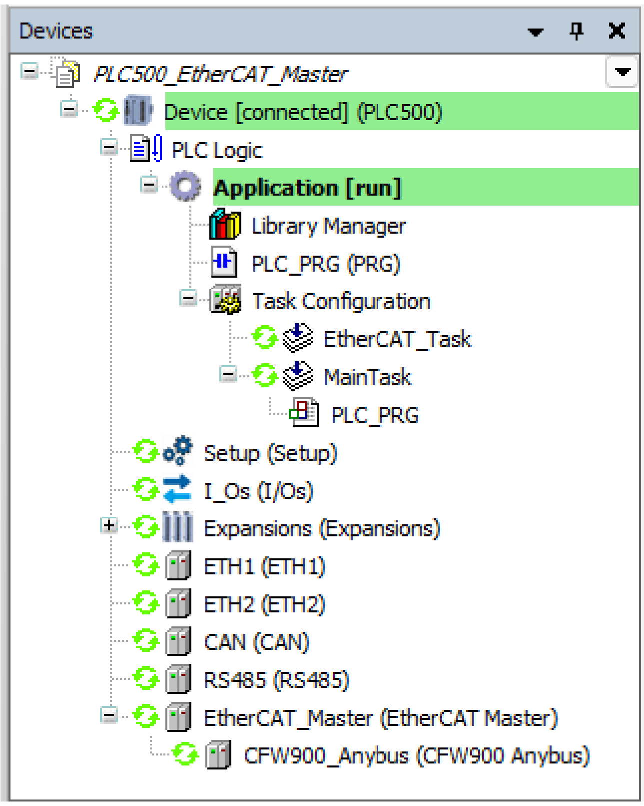


Figure 6.7: *Properly configured communication and devices communicating.*

6.3 ETHERCAT NETWORK REDUNDANCY DIAGNOSTICS

In a ring-configured EtherCAT network with redundancy, communication can continue to function even after the loss of the link between two devices. In Figure 6.8 on page 6-4, an example of a link loss case between two CFW900s is shown.

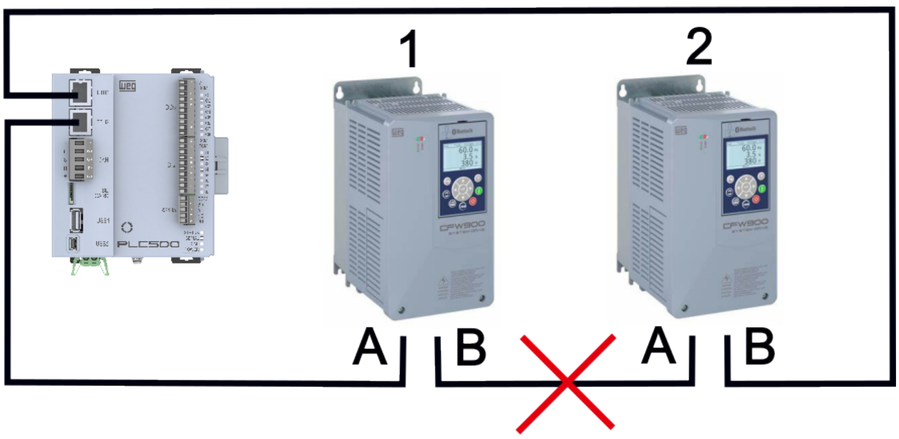


Figure 6.8: *EtherCAT communication with redundancy between PLC500 and two CFW900s.*

In CODESYS, it is possible to specifically visualize which link was lost in the network. Go to **EtherCAT Master** → **Overview**. On this page, all states of the EtherCAT slaves are shown in the **State** column. In the example of the previous figure, even after the loss of link B of inverter 1 (LINK_MIS B) with link A of inverter 2 (LINK_MIS A), EtherCAT communication is not interrupted because the PLC500 automatically redirects communication to the second Ethernet port (LINK_ADD B). This situation can be visualized in CODESYS in Figure 6.9 on page 6-5.

NOTE!

If there is more than one link loss in the EtherCAT ring network, it is not possible to guarantee that the entire network will maintain communication. The location of the fault will be more difficult to identify, as multiple link losses can cause interruptions in different segments of the network.

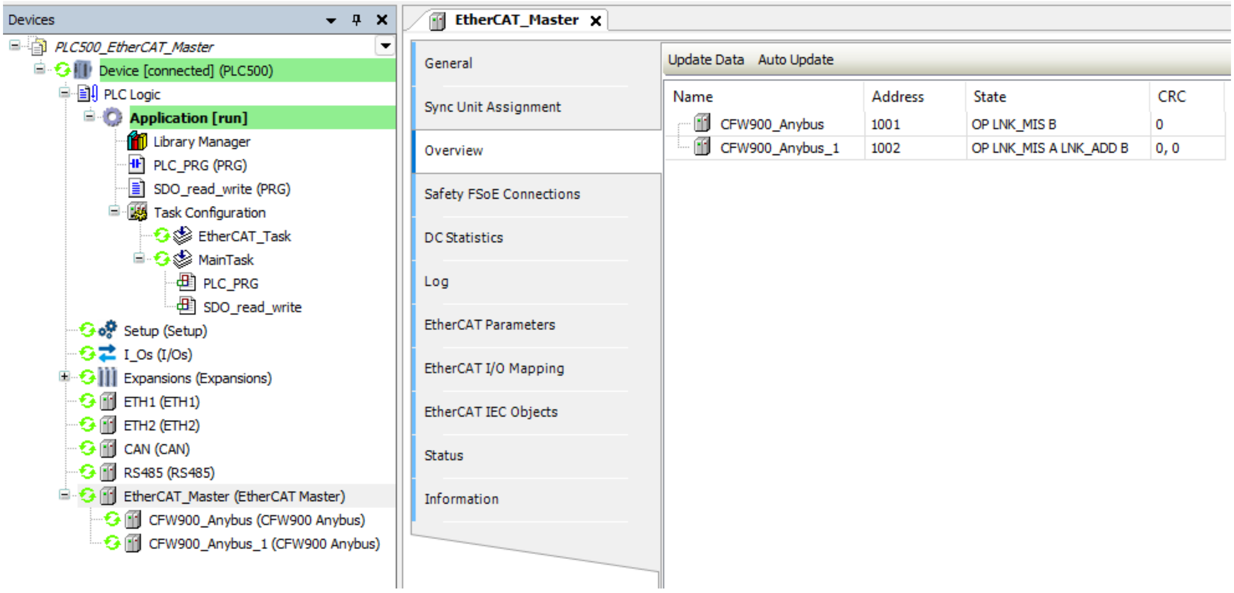


***Figure 6.9:** Identification of link loss between EtherCAT slaves with redundancy in CODESYS.*

The states of the EtherCAT slaves are described in the table below.

Name	Description
INIT	The slave is in the initialization phase.
PREOP	The slave is in the ”Pre-Operational” state.
SAFEOP	The slave is in the ”Safe-Operational” state.
OP	The slave is in the ”Operational” state.
BOOT	The slave is in the ”Bootstrap” state.
ERR	An error status is set on the slave.
NO_COMM	Communication with the slave is not possible.
LINK_MISS	A connection to the intended link is missing (Port = A, B, C, or D).
LINK_ADD	Additional connection on the displayed link.

NOTE!

For more information on configuring the EtherCAT network, as well as the available methods and functions, it is recommended to consult the documentation available in the **Library Manager** or access the online help in CODESYS Online Help.

BRAZIL
WEG DRIVES & CONTROLS - AUTOMAÇÃO LTDA.
Av. Prefeito Waldemar Grubba, 3000
89256-900 - Jaraguá do Sul - SC
Phone: 55 (47) 3276-4000
Fax: 55 (47) 3276-4060
www.weg.net