Application Note

AN2042

D-Series

Getting started with EtherCAT®

V1.00

Please check www.dimetix.com
for the latest version

Abstract

This Application Note describes a simple example of use to get started with the EtherCAT® interface of the Dimetix D-Series laser distance sensors.

Dimetix AG • Degersheimerstr. 14 • 9100 Herisau • Switzerland • Phone +41 71 353 00 00 • Fax +41 71 353 00 01 • info@dimetix.com • www.dimetix.com

Table of content

1 Document scope

This document covers an Application Note written for the Dimetix D-Series Laser Distance Sensors with EtherCAT®[1] interface. The following topics are discussed:

- Safety instructions
- Application Note descriptions

2 Safety instructions

This Application Note is written for qualified system integrators to help doing an application specific sensor configuration.

Before using the D-Series sensor also the safety related information in the D-Series Technical Reference Manuals must be consider.

Looking into the laser beam may be hazardous to the eyes.

- Do not look into the laser beam. Make sure the laser is aimed above or below eye level. (particularly with fixed installations, in machines, etc.).

Take precaution against electrostatic discharge (ESD) when the D-Series laser distance sensors exchangeable cover is open.

- Generally the sensor with removed exchangeable cover is a sensitive device and can be damaged by electrostatic discharge.
- Only handle the device properly grounded and with care.
- No warranty will be granted on improper handling and / or ESD caused problems.

1 EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

3 Introduction

3.1 Overview

This document describes a simple example of use to get started with the EtherCAT® interface of the Dimetix D-Series laser distance sensors. All information and instructions necessary to understand this example of use and to run it on a Beckhoff EtherCAT® master PLC are included. The used example project for the Beckhoff PLC can be downloaded from the Dimetix website (www.dimetix.com).

The following functions are covered by this example:

- Process input / output data
  - Measurement Control – Start / Stop continuous distance measurement
  - Distance Integer / Distance Float – Distance data of the laser sensor
  - Distance Unit – Selected distance unit number for distance data
- Acyclic read / write services
  - Serial Number – Read serial number of laser sensor
  - Distance Unit – Read / Write distance unit number for distance data

Additionally the following protocol specific features are covered too:

- EtherCAT® address
- Explizit device ID, Hot connect group

For detail information about the laser sensor or the Industrial Ethernet (PROFINET®, EtherCAT® or EtherNet/IP™) interface, please see the corresponding Technical Reference Manual on the Dimetix website (www.dimetix.com). Please note, the used designations in this document refer to the previously mentioned Technical Reference Manuals.

For questions, comments or technical support concerning this document please contact us (service@dimetix.com). Please note, we are able to support you regarding our laser distance sensor but we only have limited support possibilities regarding the EtherCAT® networks as well as for the used PLC.

3.2 Prerequisites – Hardware & Software

The following hardware and software are used to create this example:

- PLC hardware: Beckhoff TwinCAT 3.1 (4022) PLC
- PLC software: Beckhoff TwinCAT 3 (V3.1.4022) – Ensure the TwinCAT software is installed and running correctly.
- Sensor hardware: Dimetix laser distance sensor with correct assembled EtherCAT® interface (for details about assembling the exchangeable cover with EtherCAT®, see the Technical Reference Manual of the Industrial Ethernet on the Dimetix website).
- Sensor software: Only the ESI file of the Dimetix sensor. No additional sensor software.

Remark: The TwinCAT V3.1.4022 is almost the latest version, but there are no notable differences between this and newest version.

4 Description file (ESI)

First of all the ESI file of the laser distance sensor must be added manually to the installation directory. The directory can be found on the path "C:\TwinCAT\3.1\Config\Io\EtherCAT". After copying the ESI file to this directory the TwinCAT device descriptions must be updated, TwinCAT –› EtherCAT® Devices –› Reload Device Descriptions. For details see figure 1.

The latest ESI file for the Dimetix laser distance sensor with EtherCAT® can be downloaded from www.dimetix.com/IndustrialEthernet.

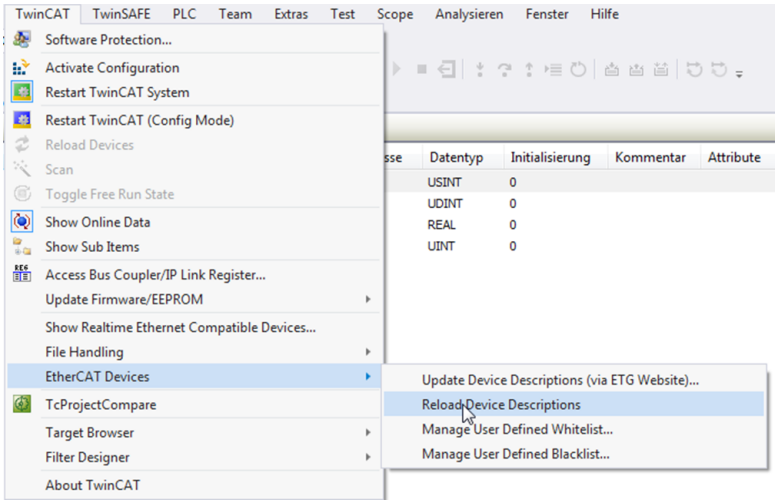

Figure 1: TwinCAT – Reload device descriptions after adding ESI file manually. TwinCAT –› EtherCAT® Devices –› Reload Device Descriptions.

5 Item configuration

In general, EtherCAT® devices can be added manually or by scan to the network

5.1 Manual

A new EtherCAT® slave can be manually added by using the context menu of the master device. Then in the window Insert EtherCAT® device the corresponding device can be selected. See figure 2 and 3 for details about the adding procedure.

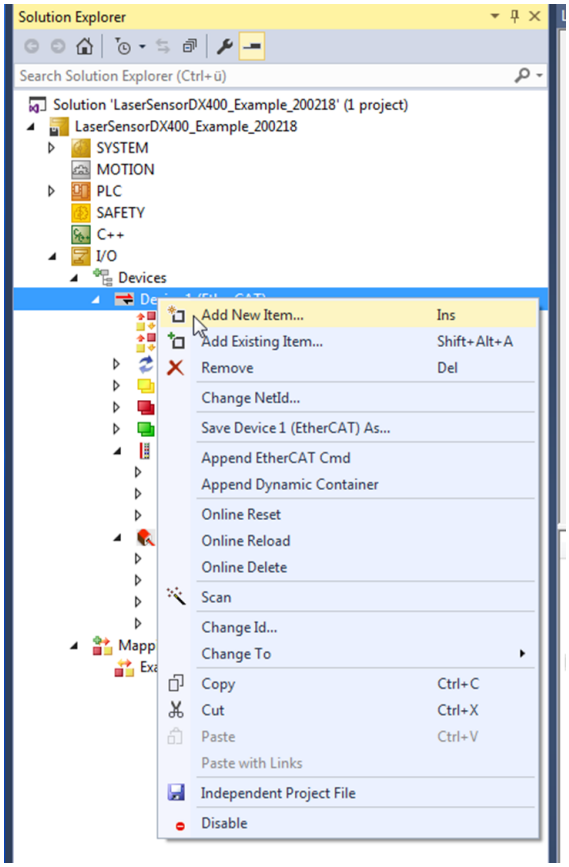

Figure 2: TwinCAT – Adding a new EtherCAT® slave. Context menu (of the master) –› Add New Item... to add a new slave device.

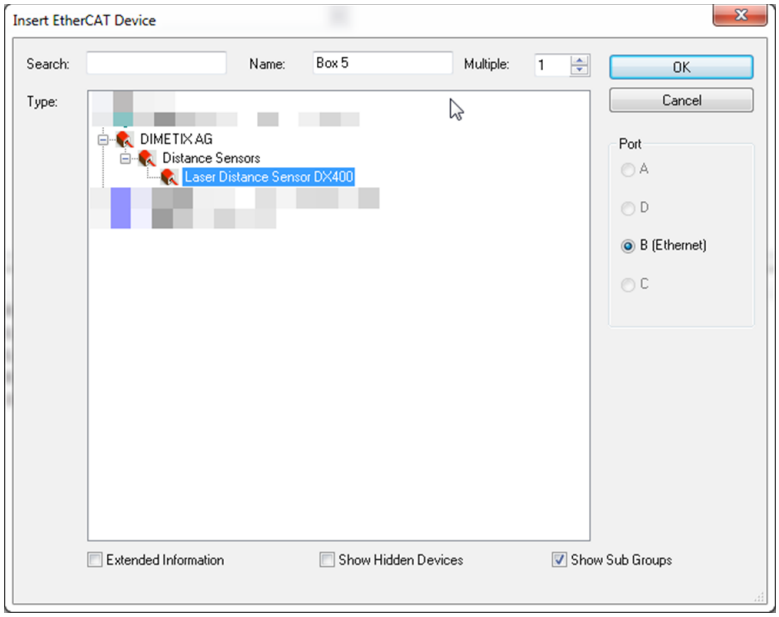

Figure 3: Insert EtherCAT® device – Select the corresponding EtherCAT® device, in this example "Laser Distance Sensor DX400".

5.2 Scan

A new EtherCAT® slave can be automatically added by using the scan functionality of the TwinCAT. Before it is possible to perform a scan, the TwinCAT must be restarted in the "Config Mode". See figure 4 for details about restarting in "Config Mode".

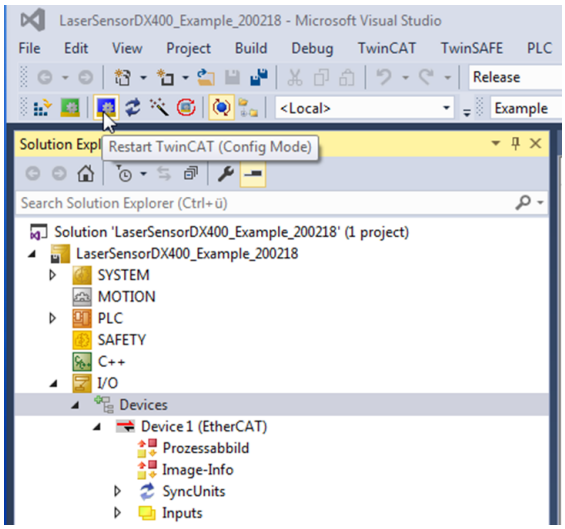

Figure 4: TwinCAT – Activation of the "Config Mode". This mode is a prerequisite for the network scan.

After selecting the master, the network scan can be done by clicking the Scan button. See figure 5 for details how to start the network scan.

Remark: Restarting the TwinCAT in the "Config Mode" is necessary before doing a network scan.

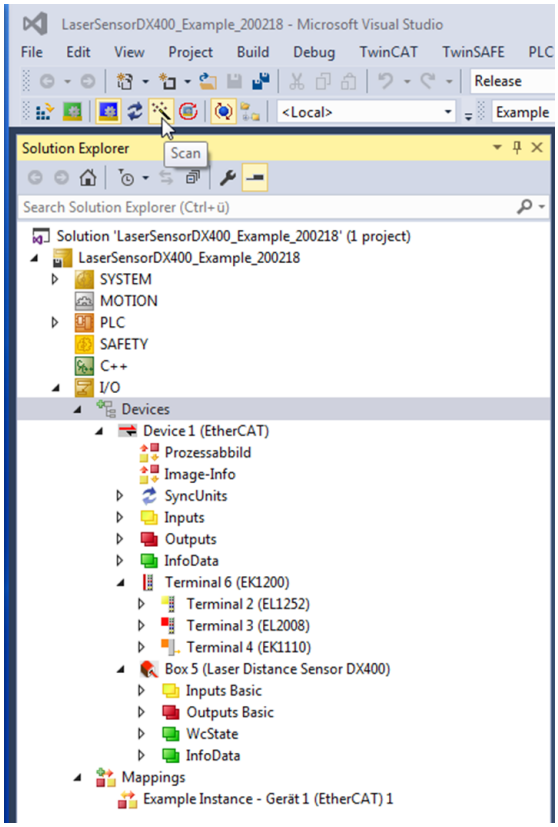

Figure 5: TwinCAT – Start of a network scan with "Scan" button to search for connected EtherCAT® slave devices.

6 Network view

Generally in the network view all added slaves are shown in the IO device group (after a network scan). See figure 6 for the network view.

Remark: The network topology line, ring or tree are possible.

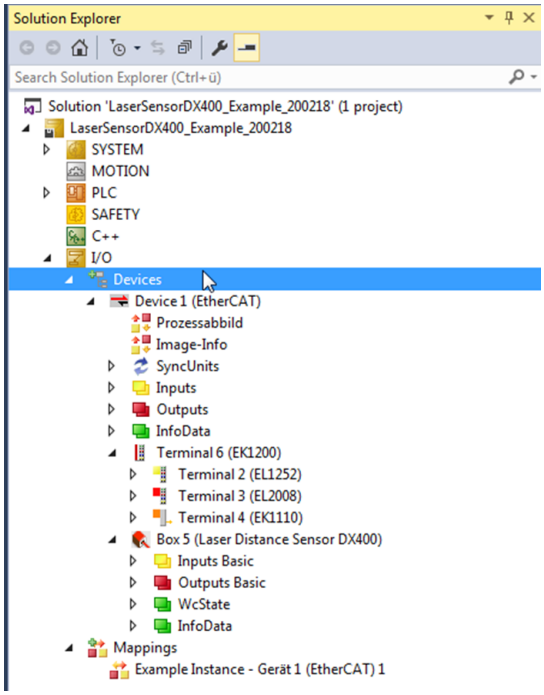

Figure 6: TwinCAT – Network view: All added devices are shown in the group I/O –› Devices.

7 Device view

The device view in figure 7 shows the detail information of a selected device. This can be achieved by double clicking the corresponding device in the network view. In the tab General the name of the box (device) can be modified if desired.

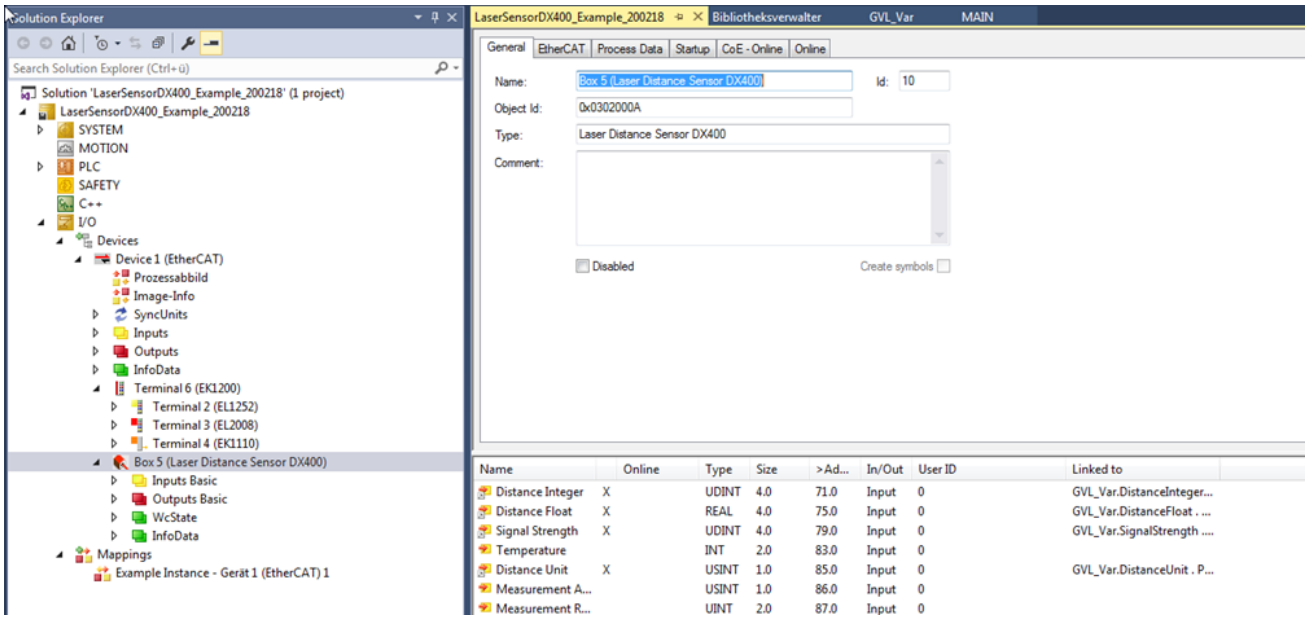

Figure 7: TwinCAT – Device view: Double-click on the desired EtherCAT® device to open the device view.

7.1 EtherCAT® address

The EtherCAT® address can be accessed over the tab EtherCAT® in the device view (see figure 8 for details). This address will be automatically assigned to the slave device during the scan procedure.

Remark: The advanced settings allows further EtherCAT® specific settings. Normally this is not needed or this values are determined over the ESI file.

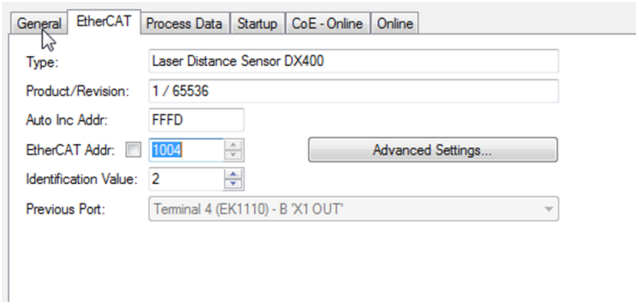

Figure 8: Device view – EtherCAT® tab: EtherCAT® address of the selected device. Address is automatically set by master during scan.

7.2 Process data

The available process data of the device are shown in the Process Data tab in the device view (see figure 9 fro details).

Remark: For the Dimetix Laser Distance Sensor DX400 the process data are not changeable and therefore corresponds always to the Input Basic and Output Basic on the left side in the tree structure.

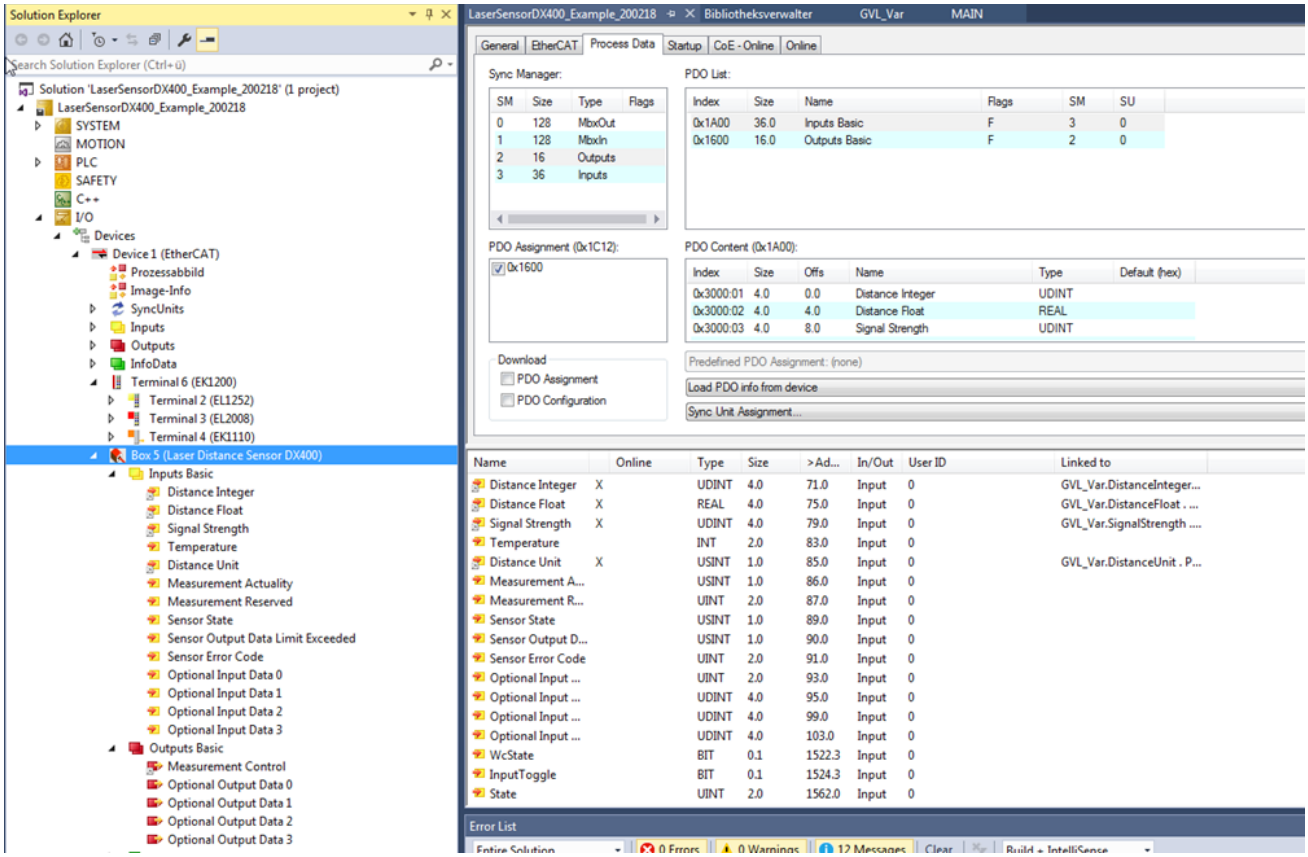

Figure 9: Device view – Process Data tab: Available process data of the device. Not changeable for the DX400 device and therefore corresponds to the Input Basic and Output Basic on the left (in the tree structure).

7.3 CoE object list / dictionary

The CoE object list can also be accessed over the tab CoE – Online in the device view (see figure 10 for details). In the CoE object list either the online data (read from the device) or the offline data (read from the ESI file) are shown.

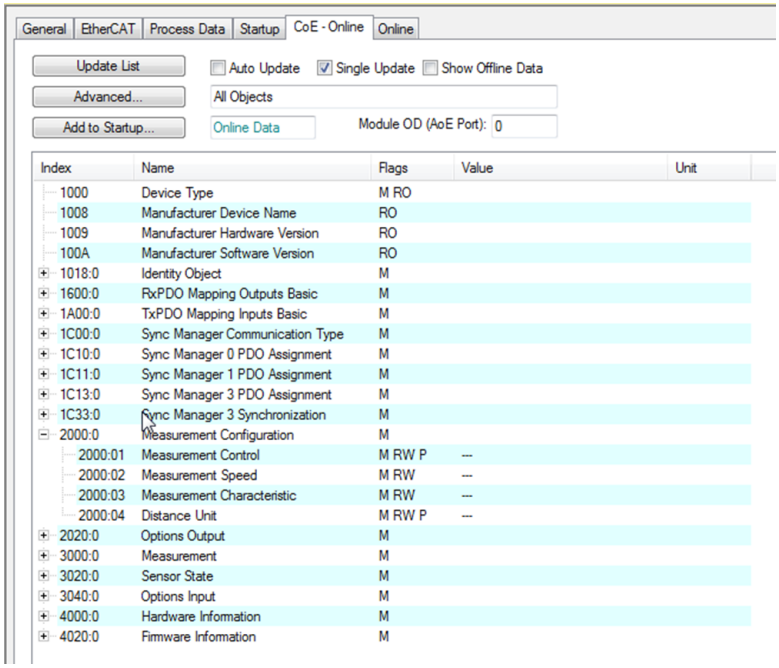

Figure 10: Device view – CoE-Online tab: Overview of the object directory

By clicking the Advanced button in the CoE – Online tab the source / type of the data can be selected between online (from device) or offline (from ESI file). See figure 11 for more information.

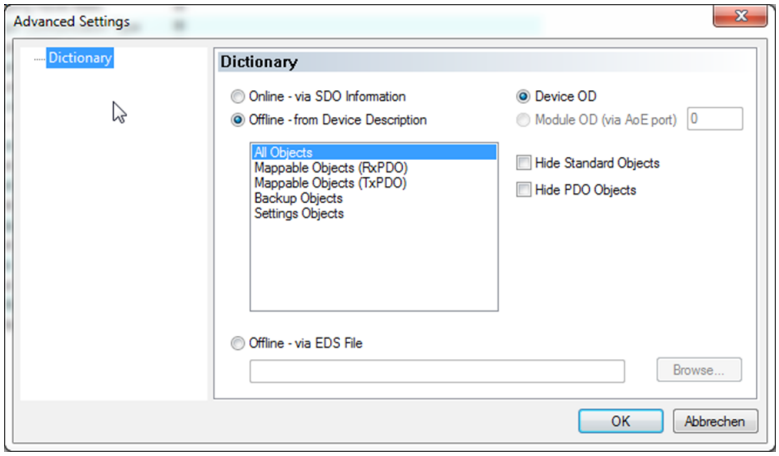

Figure 11: Device view – CoE-Online tab –› Advanced... button: Differentiation between online and offline version.

7.4 EtherCAT® state machine

The EtherCAT® state machine in figure 12 shows the state of the slave device. This data can be accessed over the tab Online in the device view. Generally after booting or activating the configuration, the state machine is switched to Operational (OP) by the master.

Remark: The exchange of the process data (input and output) is only allowed in the state Operational (OP). In other states than Operational no process data exchange is done.

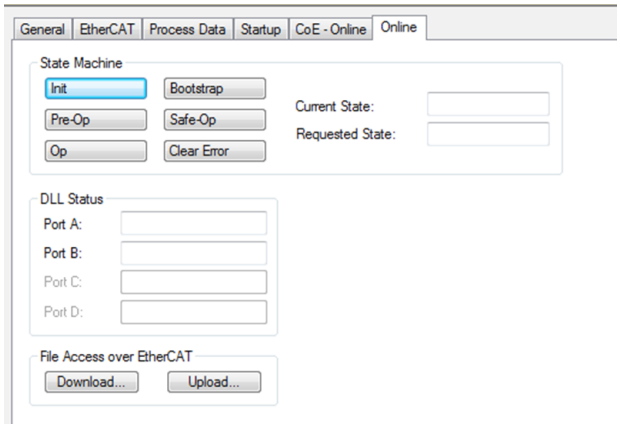

Figure 12: Device view – Online tab: Information about EtherCAT® state machine controlled by master. Normal operation state: Operational (OP) wit exchange of process data.

8 PLC application

8.1 PLC task – Cycle time / ticks

In the Tasks group of the project tree the "PlcTask" task can be found (see figure 13 for details). This task is generated automatically with the project and is used to set the cycle time of the task itself and therefore of the slave device too. In this example with a cycle time of 10 ms.

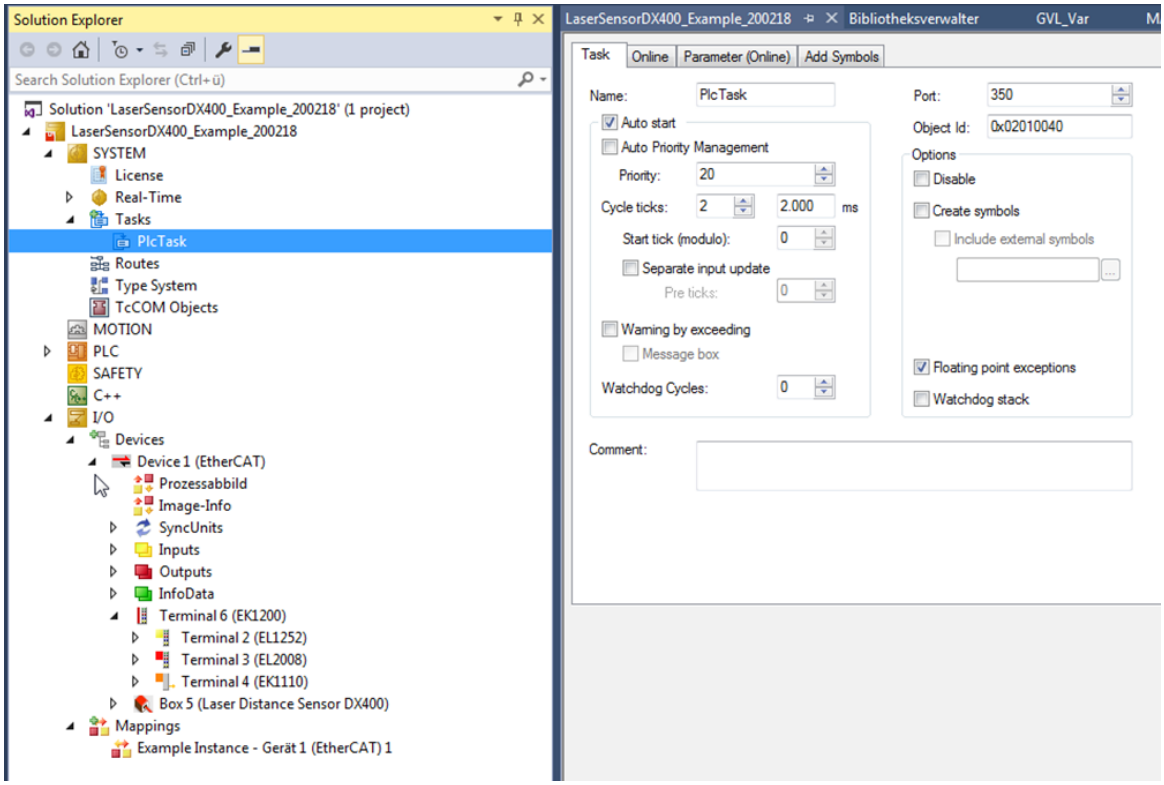

Figure 13: TwinCAT Tasks – PLC task: Configuration of the cycle time / ticks of the PLC task and in this case of the slave device too. An overview of all tasks can be found in the Real-Time group in the project tree, see figure 14.

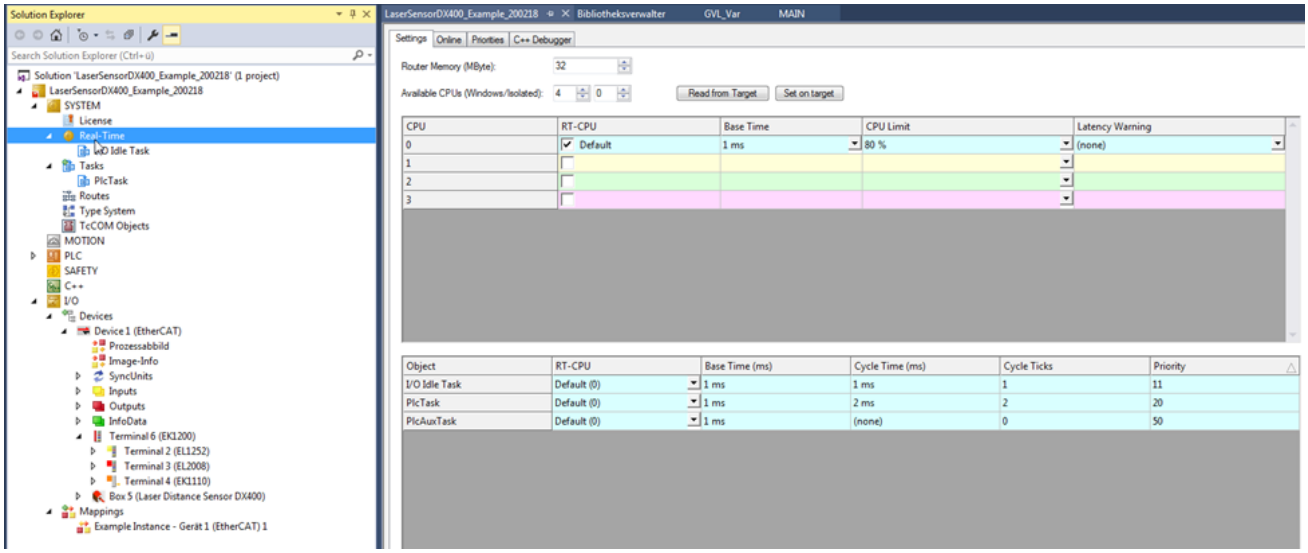

Figure 14: TwinCAT Real-Time – Overview of all existing tasks in the Real-Time view.

8.2 POU & GVL

In the PLC group of the project tree the MAIN (PRG) program can be found in the POUs folder and the GVL_Var in the GVLs folder (see figure 15 for details). This main program is also generated automatically while creating the project.

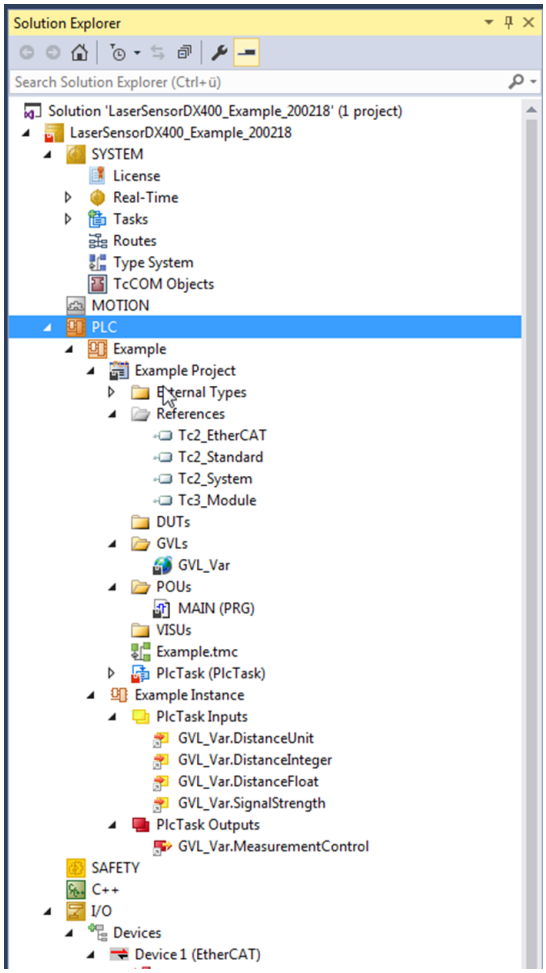

Figure 15: TwinCAT PLC – POUs –› MAIN (PRG) program: Main program of the PLC application (created automatically with new project).

8.2.1 Main program variable

Needed local variables can be declared by adding them to the variable list in the MAIN program. In figure 16 the main program variable (only local) used for this example are listed.

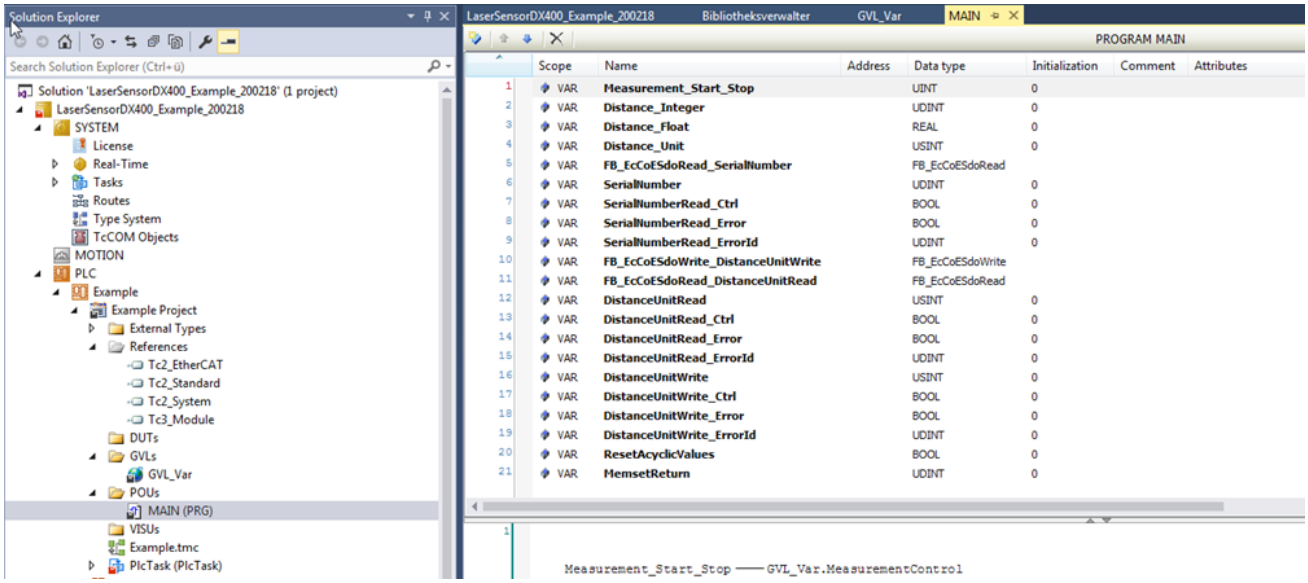

Figure 16: TwinCAT PLC – POUs –› MAIN (PRG): Declaration of program variables. Available program variables are used for this example.

8.2.2 Global variable

The input and output data (process data) of the slave devices must be linked to the global variables. The global variables are available in the GVL_Var file, see figure 17 for details.

Remark: To link the process data to the global variables the address information must be set to "%I" for input or "%Q" for output data. After the project compilation the automatically generated links will be displayed in the PlcTask Inputs and Outputs. More details can also be found on the Beckhoff website.

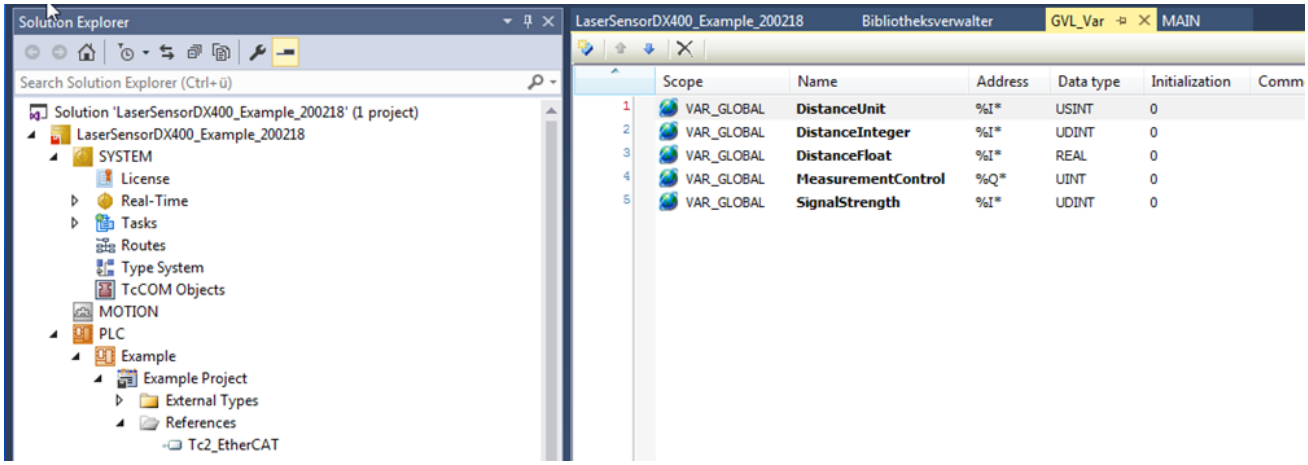

Figure 17: TwinCAT PLC – GVLs –› GVL_Var: Declaration of global variables. Process data (Input & Output) of the slave devices must be linked to global variables. The available variables are used in this example.

8.3 Measurement control

The "MeasurementControl", a part of the cyclic process output data, is used to start and stop the distance measurements of the laser distance sensor. In this example the "MeasurementControl" can be set to "1" or "0" with the associated main program variable. See figure 18 for the corresponding main program 1.

Remark: The global variables are direct linked to the process input and output data. For details see chapter 8.2.2.

Figure 18: MAIN (PRG) – Main program 1: Assigning the program variable Measurement Control to the corresponding global variable. The Measurement Control is used to start / stop distance measurements.

8.4 Input values

The input values, a part of the process input data, include e.g. the distance information of the laser distance sensor. In this example the input values are copied from the associated global variable to the corresponding main program variable. See figure 19 for the according main program 2 to 4.

Remark: The global variables are direct linked to the process input and output data. For details see chapter 8.2.2.

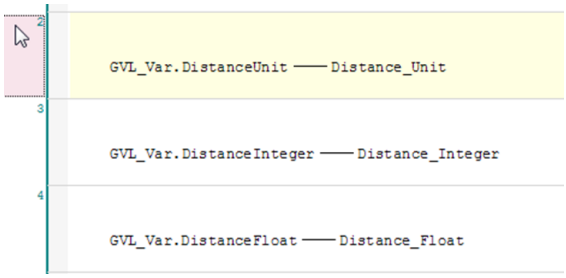

Figure 19: MAIN (PRG) – Main program 2 / 3 / 4: Assigning the global variables Distance Unit, Distance Integer and Distance Float to the corresponding program variables.

8.5 Acyclic read / write record services

The acyclic read and write record services are used to e.g. read device information and to configure the sensor. In this example the serial number and the distance unit are used to demonstrate the basic principle of reading or writing acyclic parameters. See chapter 8.5.3 to 8.5.5 for the corresponding main programs.

8.5.1 Basic information (SDO function blocks)

The mentioned read or write record services can be programmed using the FB_EcCoESdoRead and FB_EcCoESdoWrite function blocks. See figure 20 for an example of the FB_EcCoESdoRead block. Sometimes these blocks must first be made accessible by adding the appropriate EtherCAT® library, see figure 22 and 23 for details.

Some selected information for the usage of these blocks is shown in the table below. Detailed information can be found in the Beckhoff documentation.

Function block configuration	Descriptions
sNetId	NetId of the master device. See figure 21 for details where to find this NetId information.
nSlaveAddr	EtherCAT® slave address. See chapter 7.1 for details where to find this address.
nSubIndex	Subindex of the corresponding parameter in the object directory. See the Technical Reference Manual of the Industrial Ethernet for the corresponding subindex information.
nIndex	Index of the corresponding parameter in the object directory. See the Technical Reference Manual of the Industrial Ethernet for the corresponding index information.
pDstBuf	Pointer (address) of the destination variable. E.g. Read access –› Read value will be written to this destination. Write access –› Write value is read from this destination.
bExecute	Edge-triggered input for execution of the SDO function block.
tTimeout	Timeout value for the execution.
bError, nErrorId	Error flag and error ID in error case.

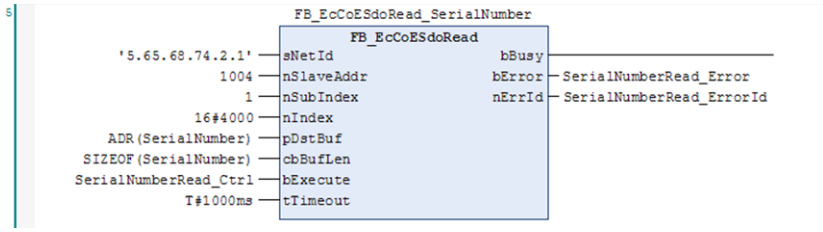

Figure 20: MAIN (PRG) – Main program 5: Example of acyclic service (SDO read function block) to read data of the connected device.

The master NetId can be found in the device view of the master device. For details see figure 21

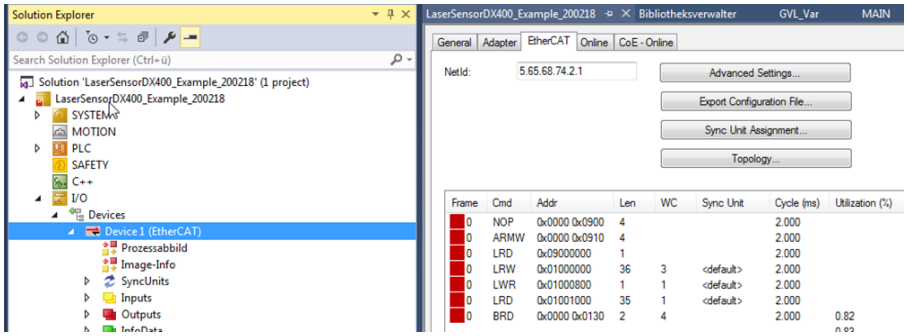

Figure 21: TwinCAT I/O – EtherCAT® tab: NetId of the master device. This value ois used for the acyclic SDO function blocks.
If the function blocks are not yet available, the corresponding EtherCAT® library must be installed. For details how to do the installation, see figure 22 and 23. After this step the SDO function blocks for the acyclic read or write record service should be available.

Remark: This EtherCAT® library should be available in the TwinCAT software without additions.

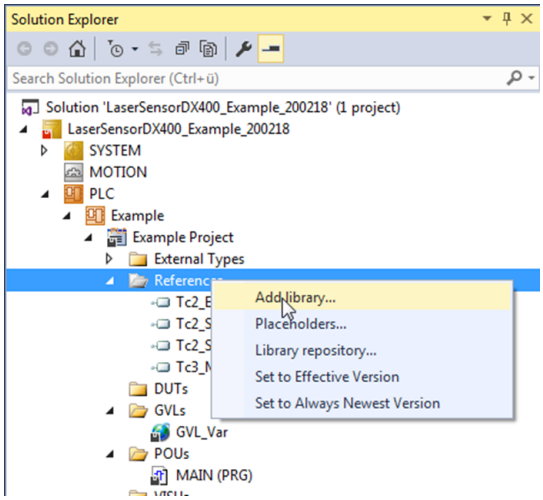

Figure 22: TwinCAT PLC – References context menu: Open Add library... to add new libraries.

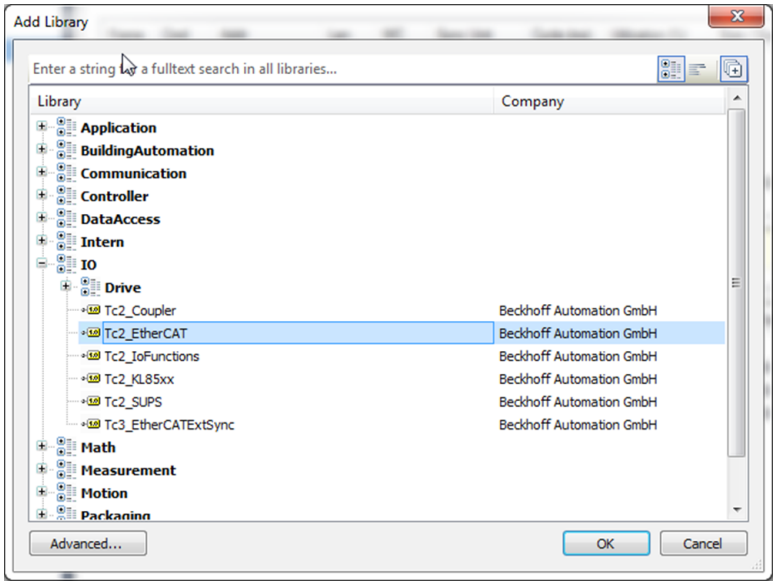

Figure 23: Add library... selection window: Select the Tc2_EtherCAT® library to add as a reference.

8.5.2 Parameter initialization

For EtherCAT® no specific parameter initialization procedure is used during the establishment of the connection. Basically the parameters of the object directory are handled with the acyclic read or write services (acyclic CoE protocol, using the EtherCAT® mailbox with SDO upload and SDO download).

Remark: The EtherCAT® state machine (see also chapter 7.4) allows different function levels. Therefore e.g. in the state PreOperational without cyclic process data exchange, the acyclic services can already be used for the configuration of the device.

The handling of the device configuration is application specific but in normal case only needed to be done once before starting the measurement. In such a case, it may be useful to use the PreOperational state mentioned above. However, the configuration can of course also be done in the Operational state.

8.5.3 Serial number – Read

The main program 5 in figure 24 shows the acyclic read service FB_EcCoESdoRead to read the serial number of the slave device. This read service can be triggered with the associated main program variable "SerialNumberRead_Ctrl", where this SDO Upload service (via CoE) is only executed once because of the edge-triggered execution input of the function block.

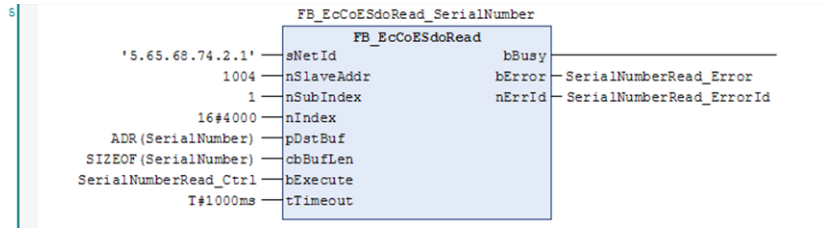

Figure 24: MAIN (PRG) – Main program 5: Acyclic service (SDO read function block) to read "Serial Number" of the device.

8.5.4 Distance unit – Read

The main program 6 in figure 25 shows the acyclic read service FB_EcCoESdoRead to read the distance unit of the slave device. This read service can be triggered with the associated main program variable "DistanceUnitRead_Ctrl", where this SDO Upload service (via CoE) is only executed once because of the edge-triggered execution input of the function block.

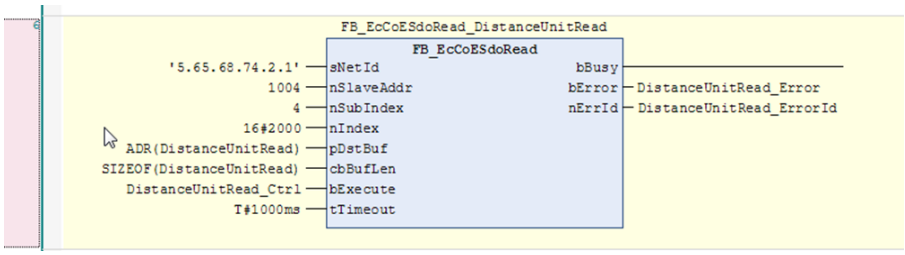

Figure 25: MAIN (PRG) – Main program 6: Acyclic service (SDO read function block) to read "Distance Unit" of the device.

8.5.5 Distance unit – Write

The main program 7 in figure 26 shows the acyclic write service FB_EcCoESdoWrite to write the distance unit of the slave device. This write service can be triggered with the associated main program variable "DistanceUnitWrite_Ctrl", where this SDO Download service (via CoE) is only executed once because of the edge-triggered execution input of the function block.

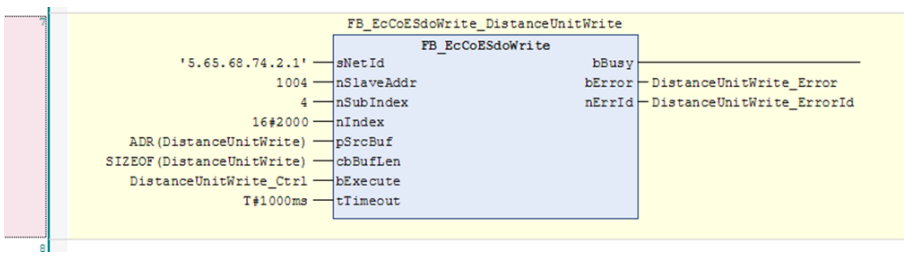

Figure 26: MAIN (PRG) – Main program 7: Acyclic service (SDO write function block) to write "Distance Unit" of the device.

8.6 Reset acyclic values

In this example the main program variables "DistanceUnitRead" and "SerialNumber" (see chapter 8.2.1 for the main program variable list) can be cleared by the associated main program variables "ResetAcyclicValues". See figure 27 for the corresponding main program used to reset the acyclic values.

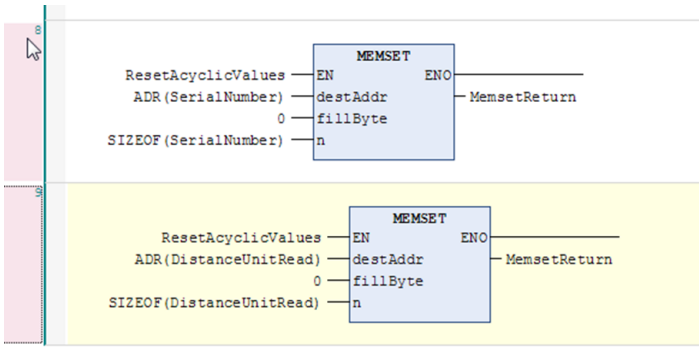

Figure 27: MAIN (PRG) – Main program 8 & 9: Reset some main program variables used for acyclic read services.

9 Explicit device ID – Hot connect

The HotConnect feature allows to distinguish identical devices using the explicit device ID. Otherwise this can be a problem if two identical devices are connected in reverse. Since the EtherCAT® addresses are assigned by the master based on the sequence in the network, it is basically not possible to detect if two identical devices are connected in reverse order. For this reason the explicit device ID is used additionally to the EtherCAT® address. For the Laser Distance Sensor DX400 the Station Alias method can be used for the explicit device ID.

A device can be added to a HotConnect group using the context menu of the desired slave device. See figure 28 for more details.

Remark: Only the Station Alias method can be used for explicit identification. No other method are supported e.g. physical ID switches or AL status register 0x0134 methods.

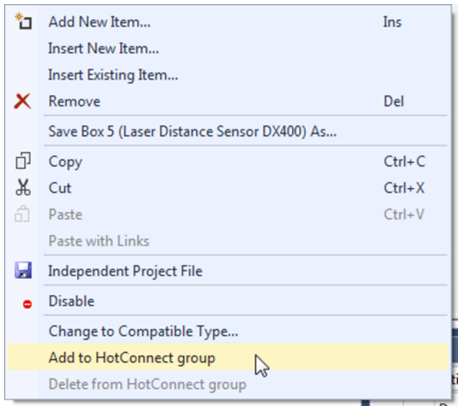

Figure 28: TwinCAT – Context menu of desired salve device –› Select Add to HotConnect group: Enabling the hot connect group feature.

The corresponding configuration of the Station Alias can then be done according figure 29 and 30.

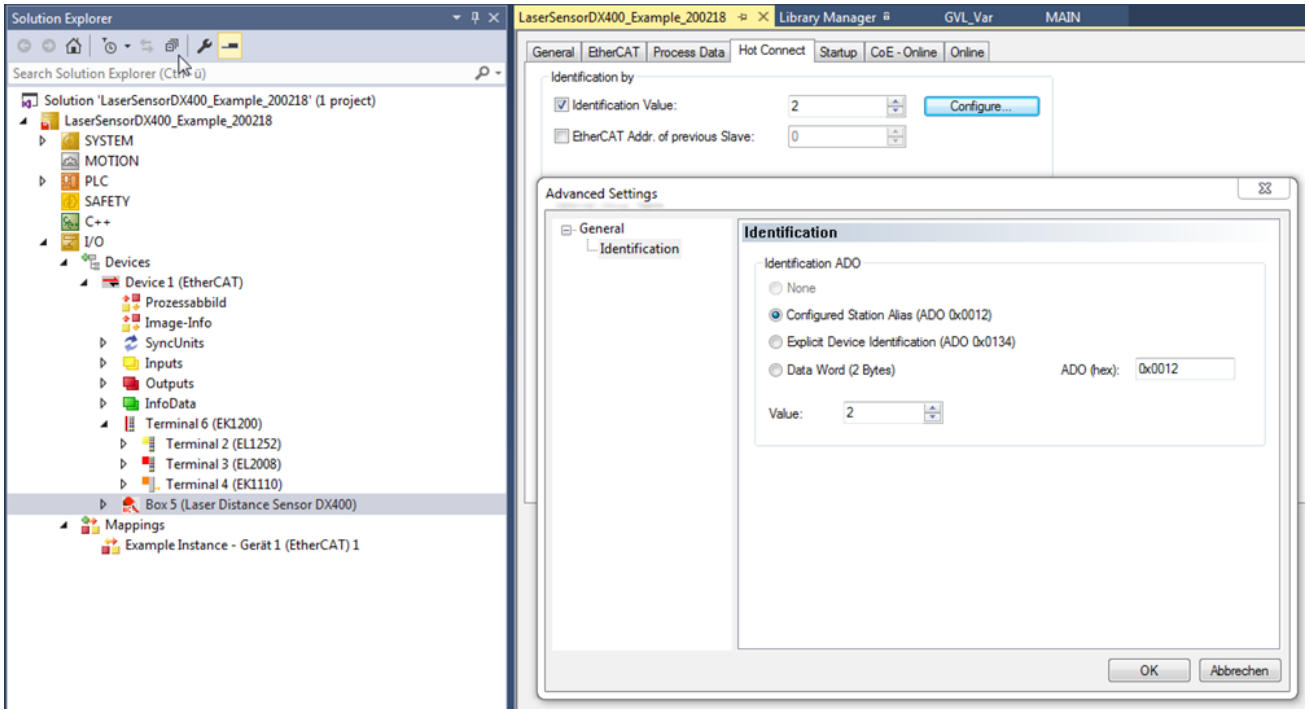

Figure 29: TwinCAT – Device view (Double-click on the desired EtherCAT® device) –› Hot Connect tab –› Configure... button. Configuration of the Station Alias in the value field. In this example: 2.
The configuration of the Station Alias value of the device can be done according figure 30. The Station Alias will then be written to the SII EEPROM.

Remark: This value is only effective after a power cycle of the EtherCAT® slave device.

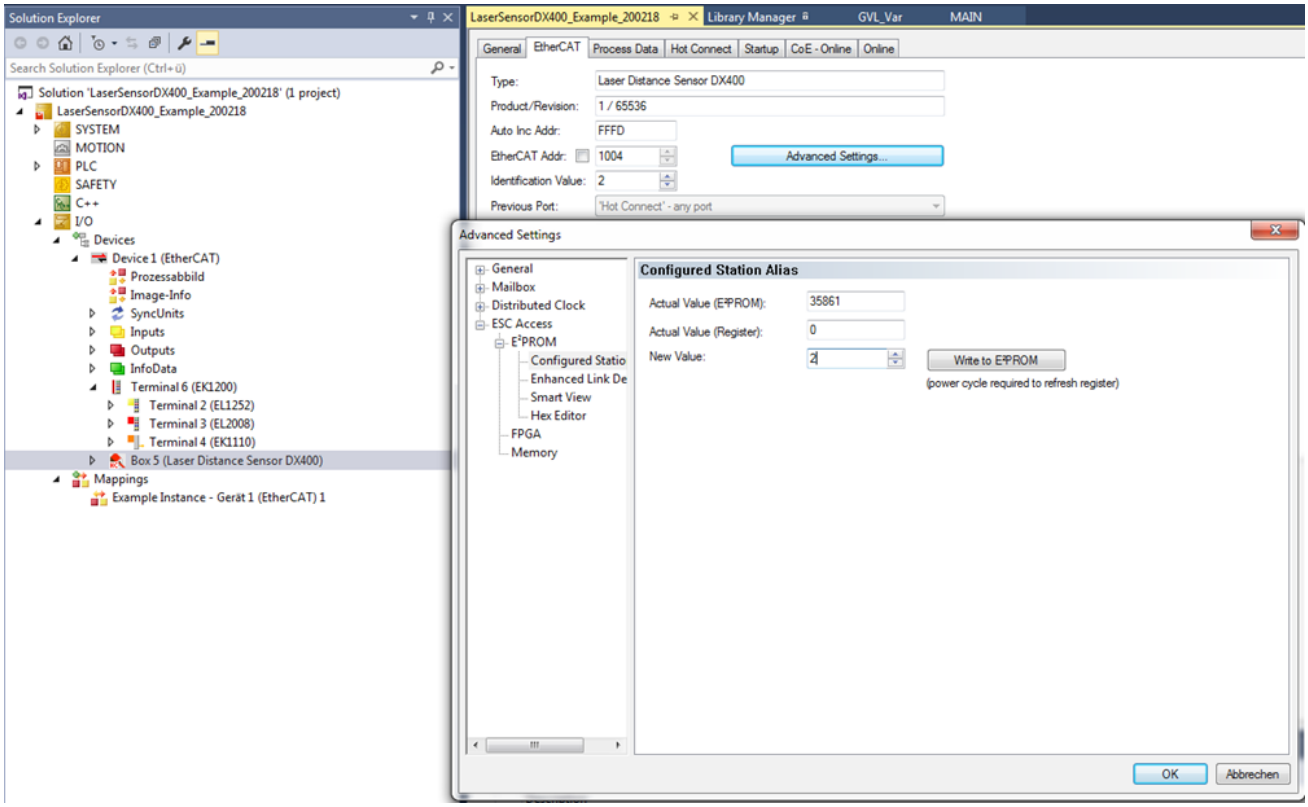

Figure 30: TwinCAT – Device view –› EtherCAT® tab –› Advanced Settings... button. Configuration of the Station Alias in the SII EEPROM of the slave device by Write to EEPROM button. The value must be as before, in this example: 2.

10 Glossary

ESI	EtherCAT® Slave Information. Describes the properties of a slave device in XML format.
EtherCAT®	EtherCAT® is one of the most popular Industrial Ethernet interfaces
GVL	Global Variable List. Global variables or constants used in the program. Detailed information can be found in the Beckhoff documentation.
HotConnect	The HotConnect feature allows to distinguish identical devices using the explicit device ID. See also chapter 9 and the Beckhoff documentation.
PLC	Programmable Logic Controller
POU	Program Organization Unit. A POU is a unit in a program execution model. Detailed information can be found in the Beckhoff documentation.
Process data	Cyclic data communication of the Industrial Ethernet interfaces.
SDO	Service Data Object. Detailed information can be found in the Beckhoff documentation.
Station Alias	Device identification or also known as "second Slave Address" or "Station Alias". See also chapter 9 and the Beckhoff documentation.

11 Revision history

The release versions and the changes of this technical reference manual are listed below.

Date	Revision	Changes
03.03.2020	V0.01	Initial version of the EtherCAT® Application Note.
19.05.2020	V0.02	Internal feedback and revision version.
20.05.2020	V1.00	First release of the EtherCAT® Application Note.

Important Notice

Dimetix reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products, documentation and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is up to date and complete. All products are sold subject to Dimetix terms and conditions of sale supplied at the time of order acknowledgment.

Dimetix AG
Degersheimerstr. 14
CH-9100 Herisau
Switzerland

Tel. +41 71 353 00 00
Fax +41 71 353 00 01
info@dimetix.com
www.dimetix.com