OPERATION MANUAL

2511
EtherCAT Integration into TwinCAT

© 2024 burster
praezisionsmesstechnik gmbh & co kg
All rights reserved

Valid from: **01.04.2024**
Applies to: **2511-VXXX1**

Manufacturer:
burster
präzisionsmesstechnik gmbh & co kg
Talstr. 1 - 5
76593 Gernsbach
Germany

P.O. Box 1432
76587 Gernsbach
Germany

Tel.: +49-7224-645-0
Fax.: +49-7224-645-88
Email: info@burster.com
www.burster.com

4539-BA2511ETHERCATEN-5799-041528

Table of Contents

Introduction **3**

1. Creating new project 4
2. Installation of ESI description files 6
3. Scan EtherCAT devices 6
4. Create a sample program 9
5. Further Examples 17

5.1 Read and Write of ‘real’ data types 17

5.2 Read and Write of ‘string’ data types 21

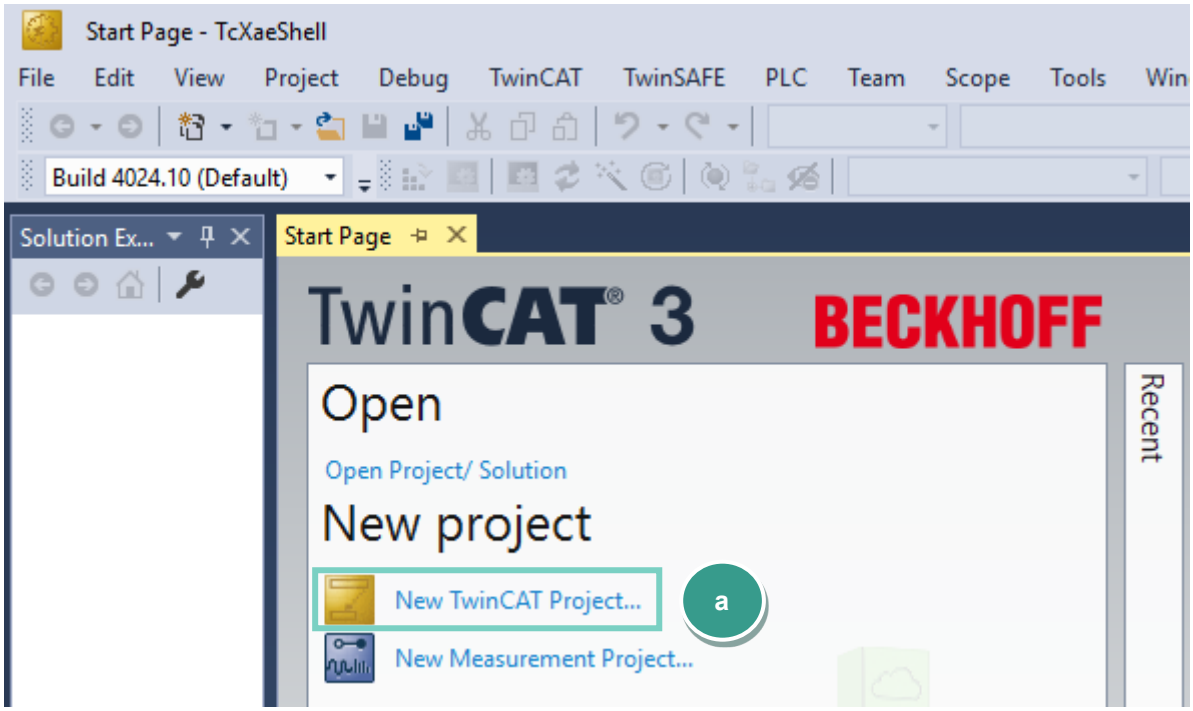
Introduction

This quick start guide describes an approach how you can configure the 2511 via Beckhoff TwinCAT using a Beckhoff PCI-Ethernet Card. Please note that the samples here cannot be directly used in your production line because they have beed extremely simplified to reach a better understanding. Therefore, you may have to complete them by checking of status, error, length values etc.

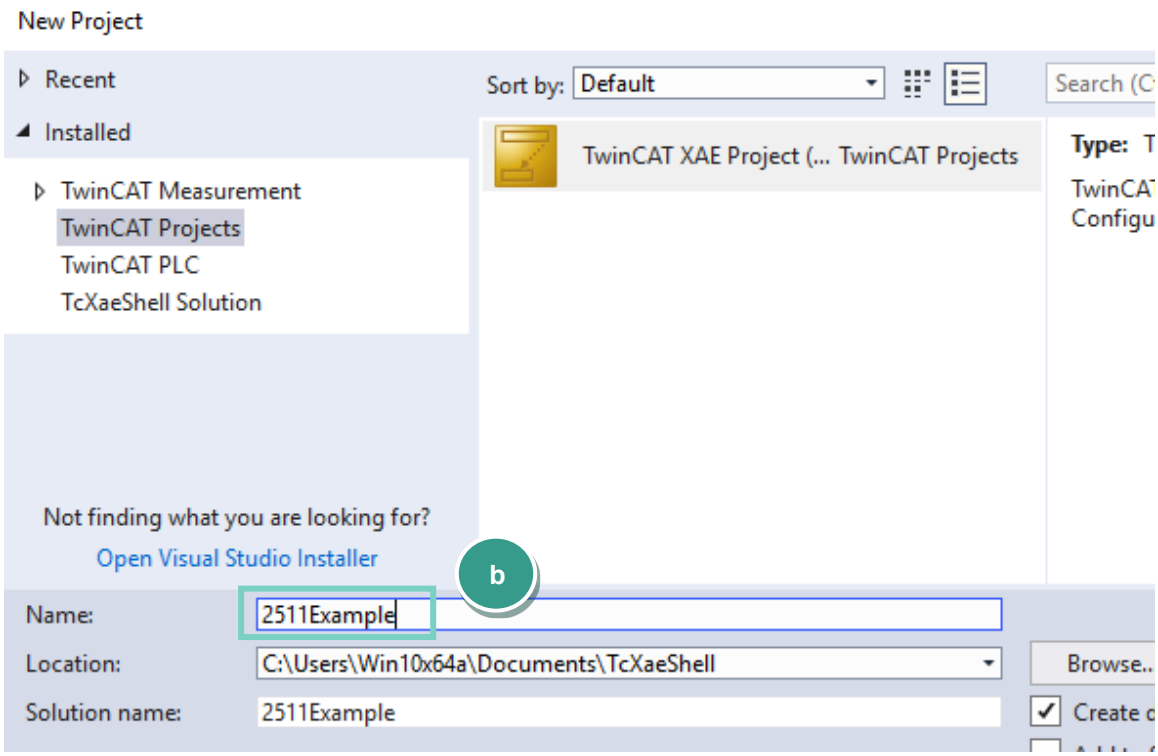
Please also note that you will have to use the 2511 manual to get futher information about input and output parameters (cyclic as well acyclic data transfer)

1. Creating new project

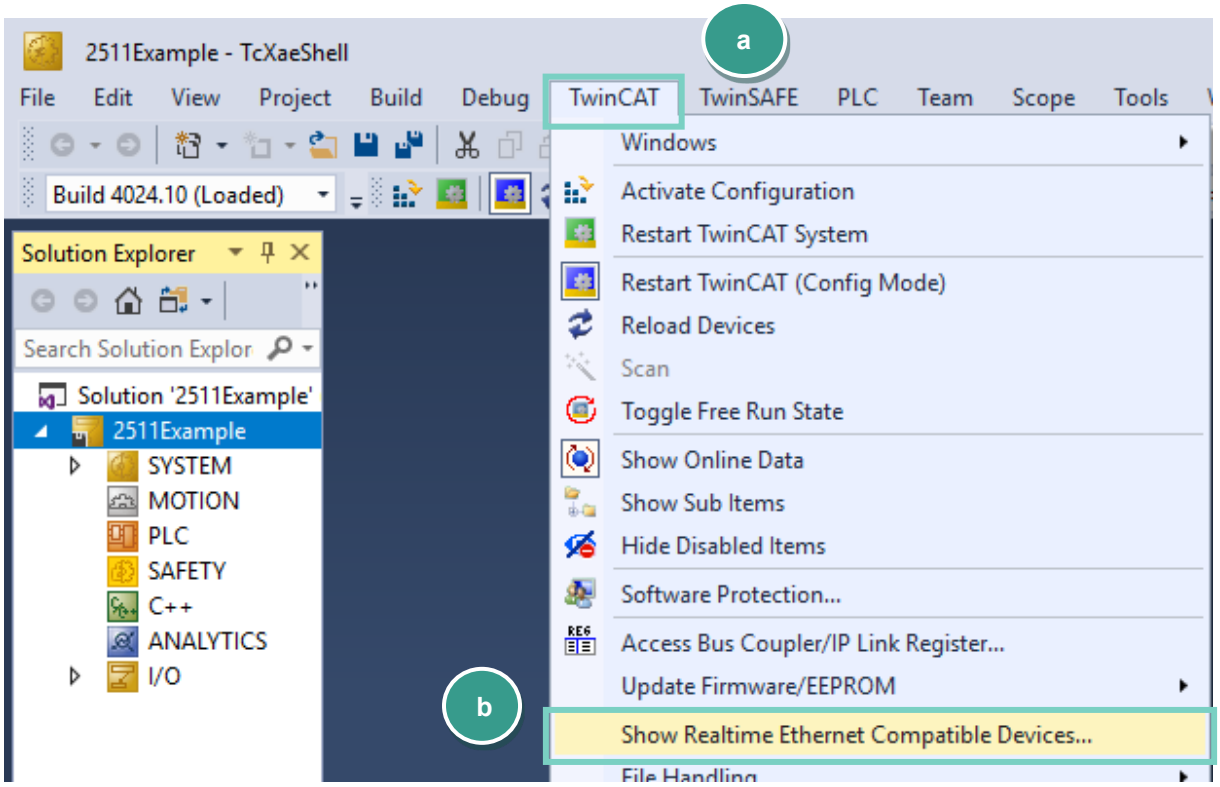
➢ Start the **TwinCAT XAE Shell** and click on ***New TwinCAT Project*** (or via ***File*** → ***New Project***) (a)

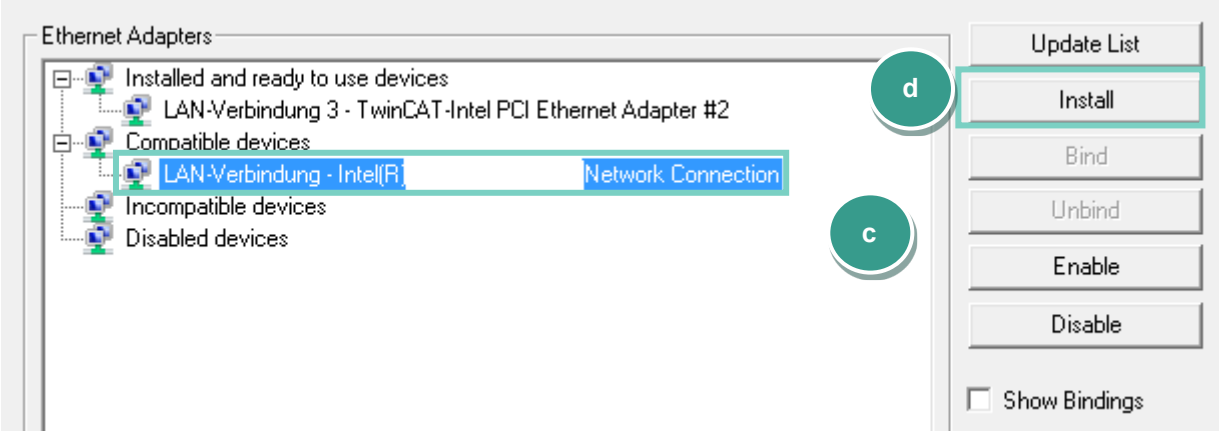


➢ Select ***TwinCAT XAE Project***, assign a project a name (b) and click ***OK***



➢ Go to ***TwinCAT*** (a), select ***Show Real Time Ethernet Compatible Devices…****(b)* and look for you're a EtherCAT Master device under Compatible devices (c). Afterwards click the ***Install*** button (d).


2511Example - TcXaeShell
File
Edit
View
Project
Build
Debug
TwinCAT
TwinSAFE
PLC
Team
Scope
Tools
Build 4024.10 (Loaded)
Solution Explorer
Search Solution Explor
Solution '2511Example'
2511Example
SYSTEM
MOTION
PLC
SAFETY
C++
ANALYTICS
I/O
Windows
Activate Configuration
Restart TwinCAT System
Restart TwinCAT (Config Mode)
Reload Devices
Scan
Toggle Free Run State
Show Online Data
Show Sub Items
Hide Disabled Items
Software Protection...
Access Bus Coupler/IP Link Register...
Update Firmware/EEPROM
Show Realtime Ethernet Compatible Devices...
a
b


Ethernet Adapters
Installed and ready to use devices
LAN-Verbindung 3 - TwinCAT-Intel PCI Ethernet Adapter #2
Compatible devices
LAN-Verbindung - Intel(R)
Network Connection
Incompatible devices
Disabled devices
Update List
Install
Bind
Unbind
Enable
Disable
Show Bindings
c
d

2. Installation of ESI description files

Note: Please make sure that your ESI file is compatible to the field bus firmware in the 2511.

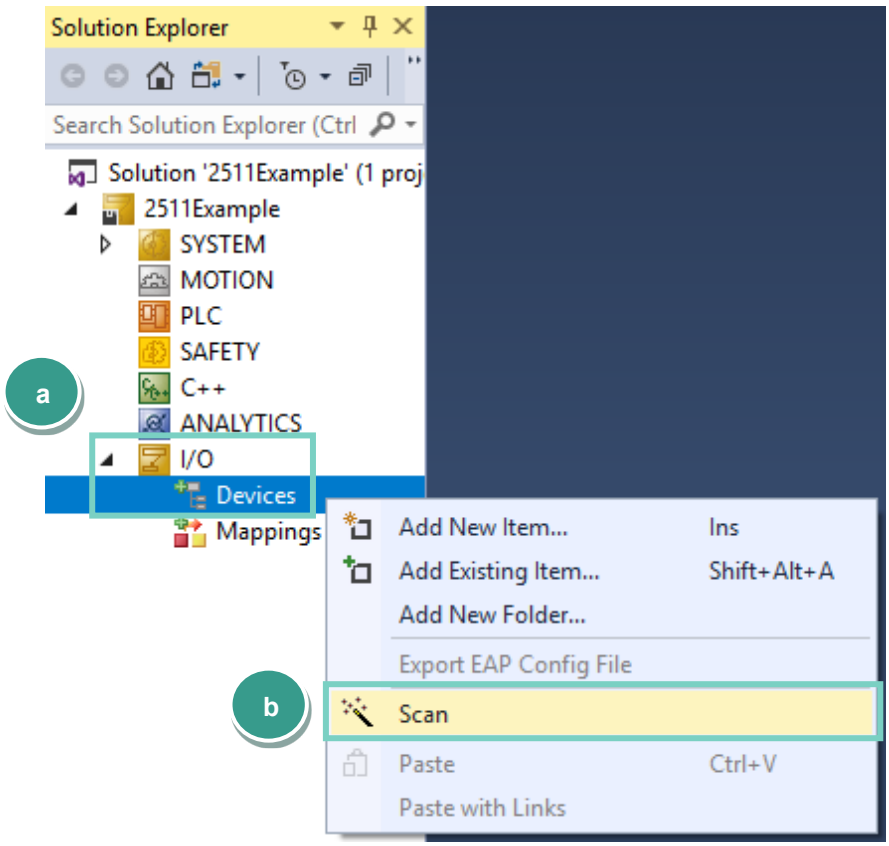
- Copy the ESI file into directory ***C:\TwinCAT\3.1\Config\Io\OnboardIo***
  and additionally into ***C:\TwinCAT\3.1\Config\Io\EtherCAT***

  Note: you will find the corresponding ESI files on burster.com

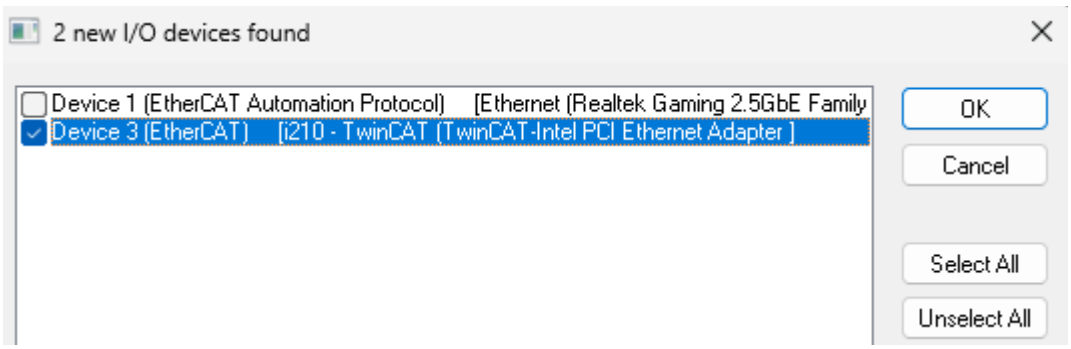
  Note: If you use the TwinCAT 2 The ESI directory would be ***C:\TwinCAT\Io\EtherCAT***

3. Scan EtherCAT devices

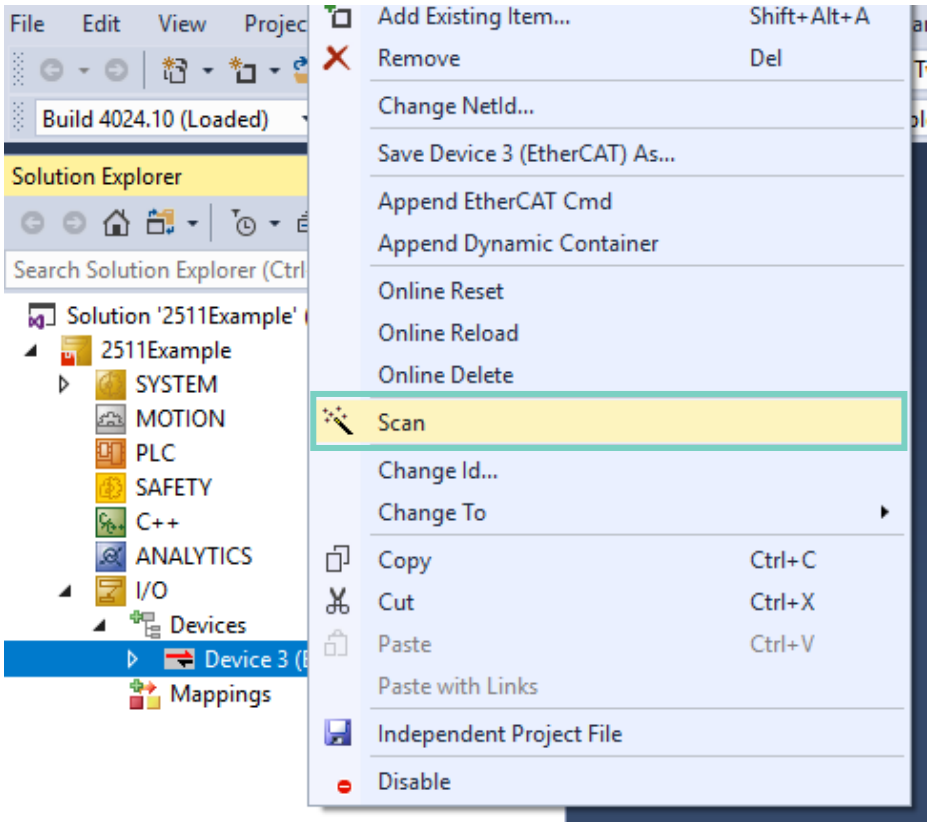
- Right click ***I/O*** → ***Devices*** (a) in the project tree und select ***Scan*** (b):



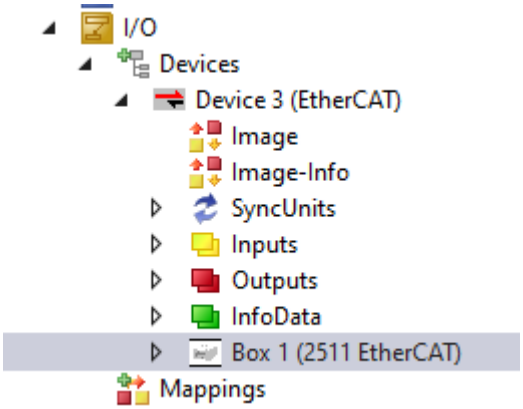
- Now, you can select an EtherCAT compatible device in the new window and click OK:



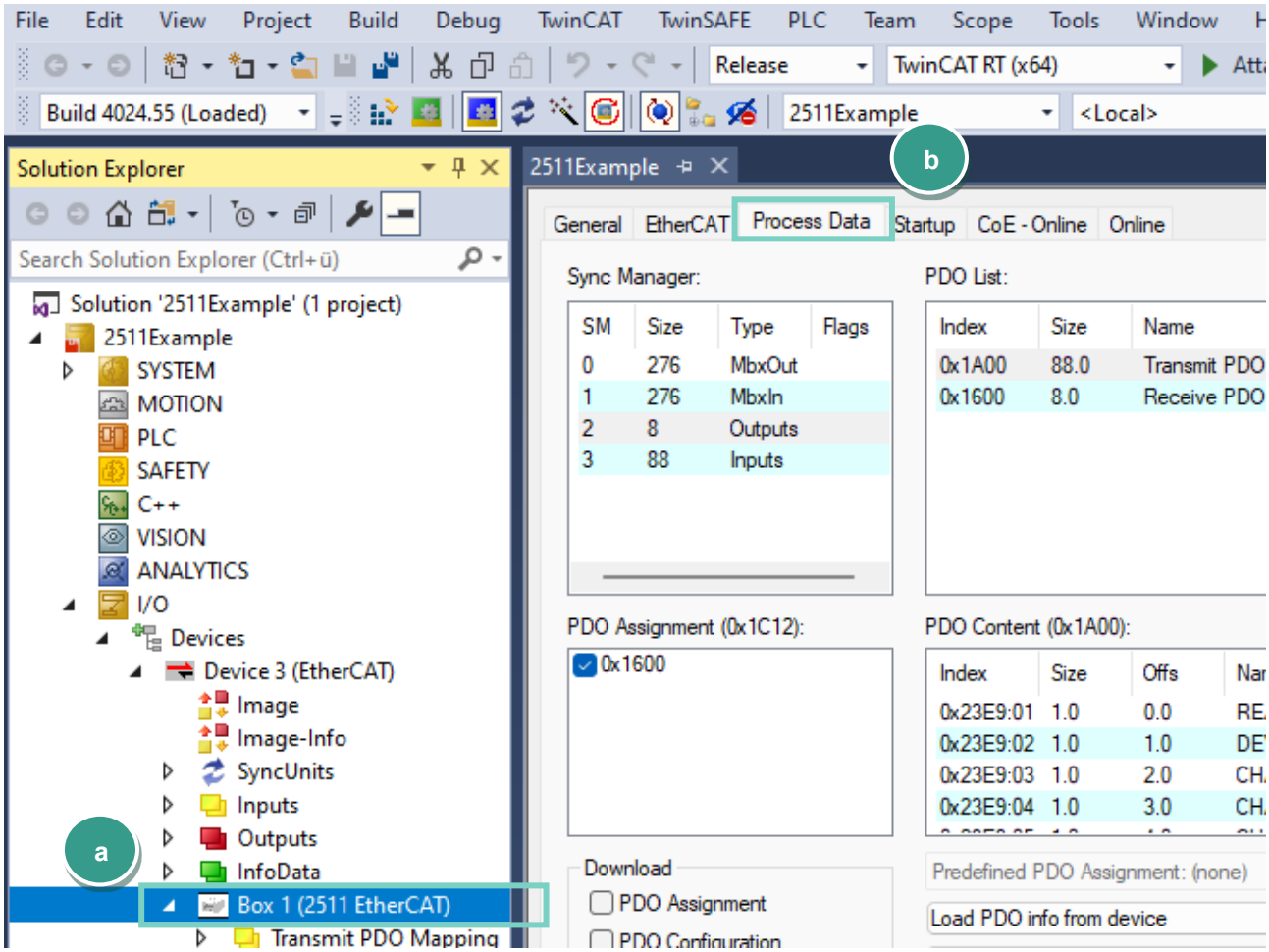
- At this point you are ready to connect the 2511 to your EtherCAT master and perform a device search by confirming the ***Scan for boxes*** request or later by right-clicking on the found EtherCAT device and selecting ***Scan*** in the context menu as shown below:



- If asked, confirm to use online description and after a while you should be able to see the 2511 device in the project tree:



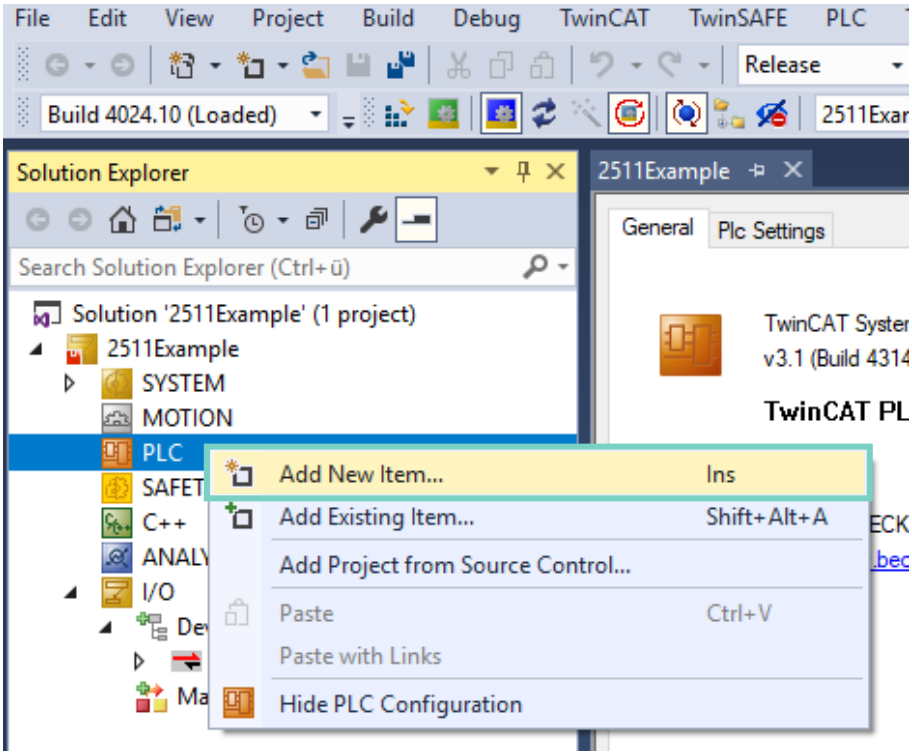
- To see the process data, please click on the **2511 EtherCAT** in the project tree (a) and select the **Process Data** tab (b):


File Edit View Project Build Debug TwinCAT TwinSAFE PLC Team Scope Tools Window
Release
TwinCAT RT (x64)
Build 4024.55 (Loaded)
2511Example
<Local>
Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution '2511Example' (1 project)
2511Example
SYSTEM
MOTION
PLC
SAFETY
C++
VISION
ANALYTICS
I/O
Devices
Device 3 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Outputs
InfoData
Box 1 (2511 EtherCAT)
Transmit PDO Mapping
a
b
2511Example
General
EtherCAT
Process Data
Startup
CoE - Online
Online
Sync Manager:
SM Size Type Flags
0 276 MbxOut
1 276 MbxIn
2 8 Outputs
3 88 Inputs
PDO List:
Index Size Name
0x1A00 88.0 Transmit PDO
0x1600 8.0 Receive PDO
PDO Assignment (0x1C12):
0x1600
PDO Content (0x1A00):
Index Size Offs
0x23E9:01 1.0 0.0
0x23E9:02 1.0 1.0
0x23E9:03 1.0 2.0
0x23E9:04 1.0 3.0
Download
PDO Assignment
PDO Configuration
Predefined PDO Assignment: (none)
Load PDO info from device

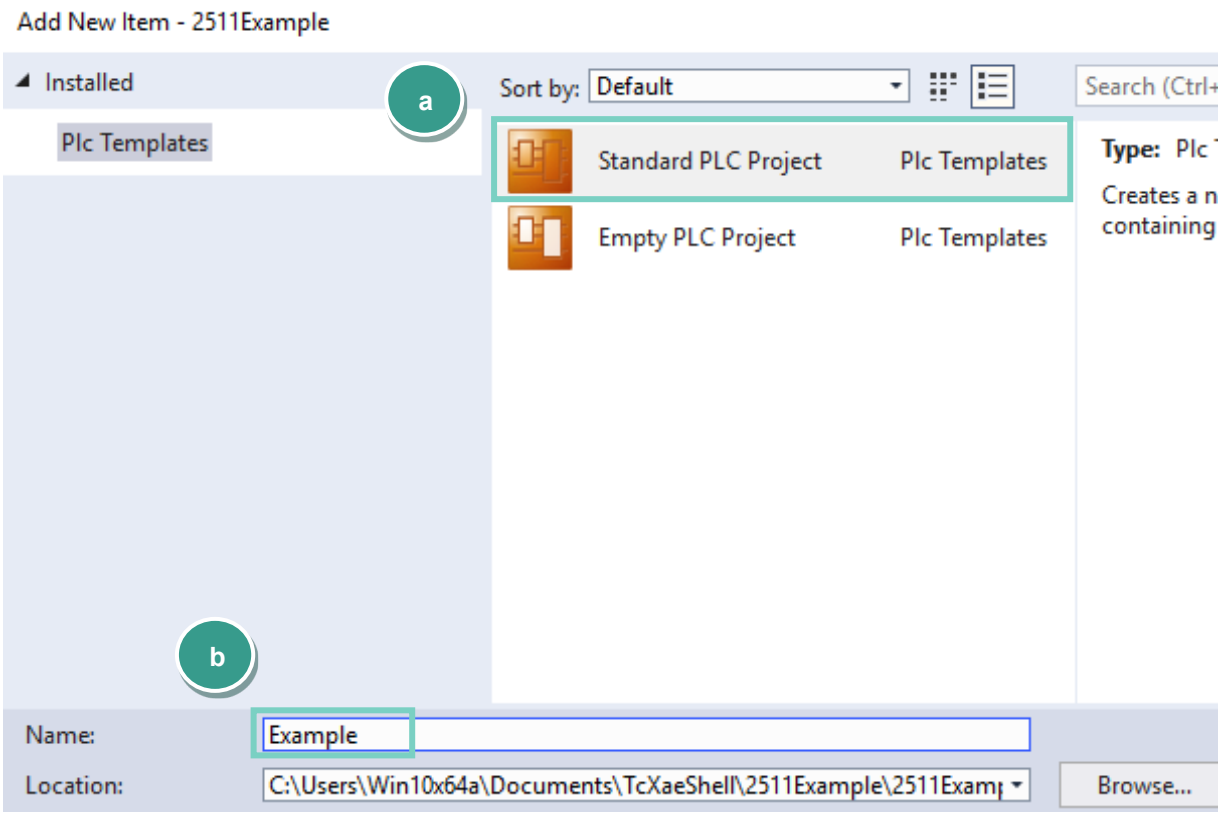
4. Create a sample program

In this section, you will learn how to create a simple PLC program to execute a measurement via PDO (Process Data Object). You will need to refer to ***2511 EtherCAT Manual*** documentation to understand the meaning of input bytes.

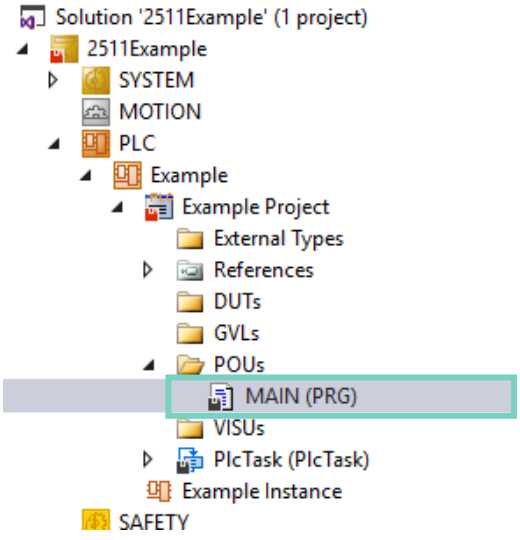
- Right-click ***PLC*** in the project tree and select ***Add New Item***...



- Select ***Standard PLC Project*** (a) in the **Add New Item** dialog, enter ***Example*** as project name (b) and click ***Add***

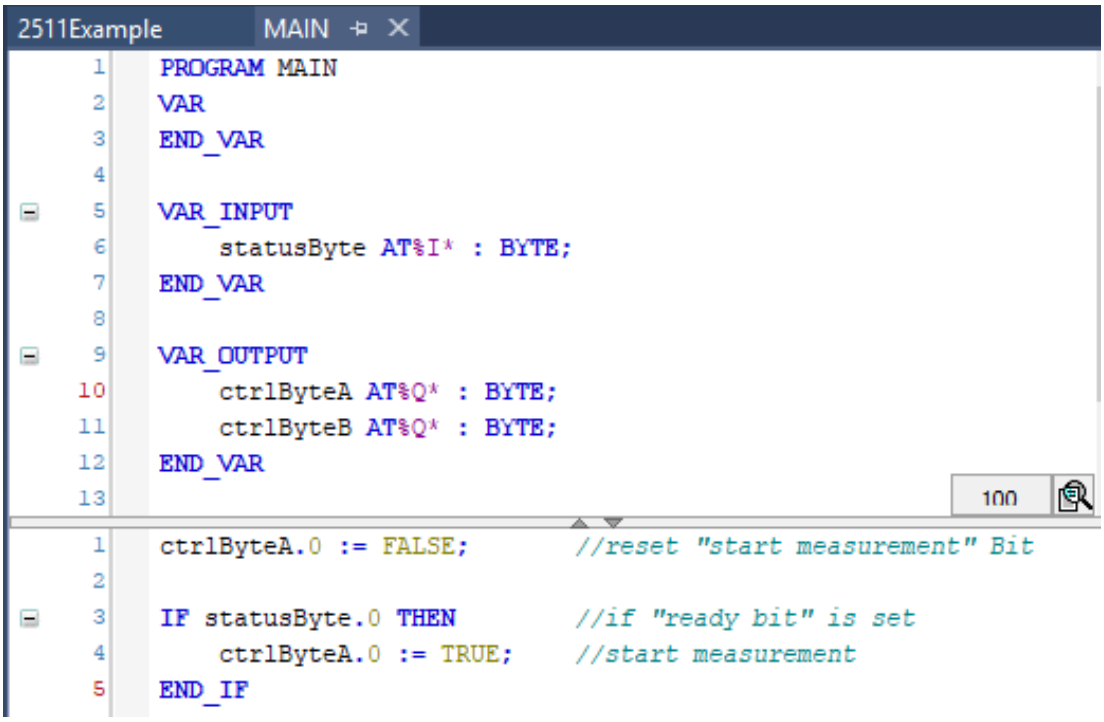


➢ Next, open the **MAIN (PRG)** file from ***PLC*** → ***Example Project*** → ***POUs*** with double click on it:

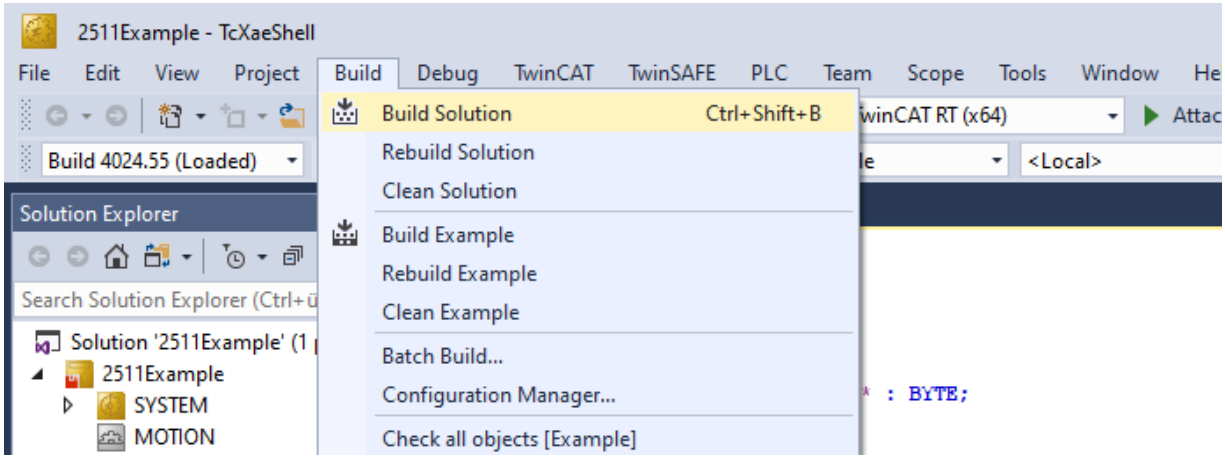


Example 1: Reading and Writing of PDOs

➢ Type in the following text in the ***MAIN*** block



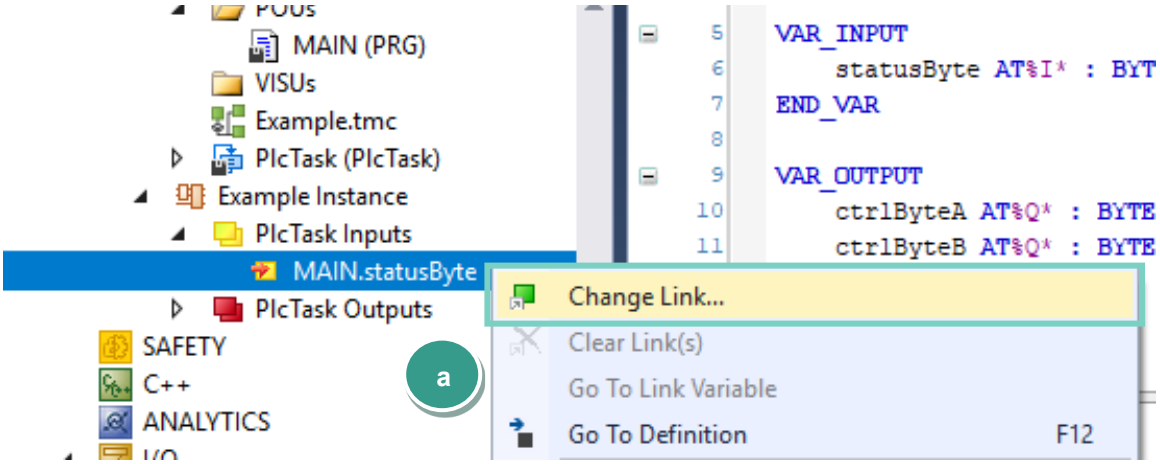
- Goto ***Build*** → ***Build Solution***

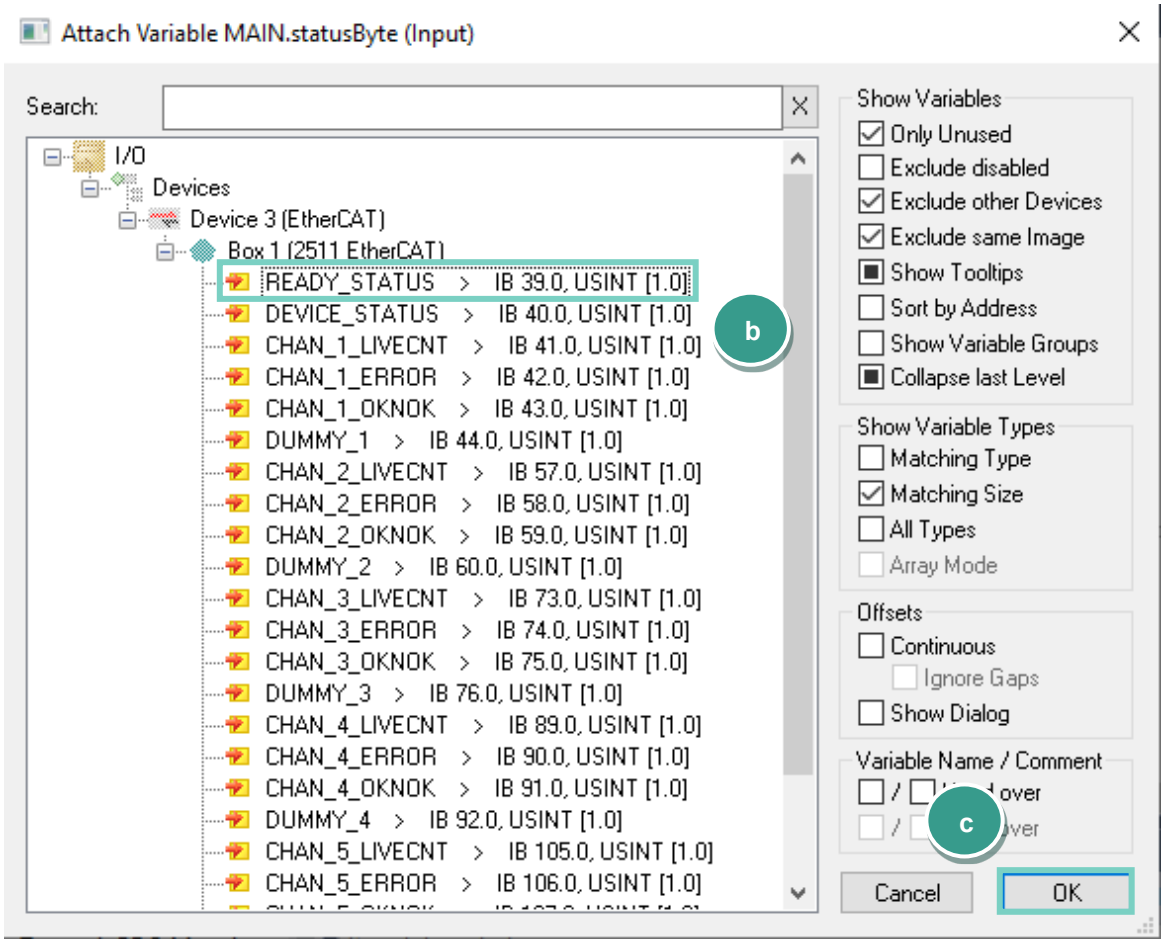


- Assign the **input** and **output** variables to the corresponded PDOs with the right-click on a variable and select ***Change Link***…(a) from the context menu, select a corresponded PDO (b) and click OK (c):

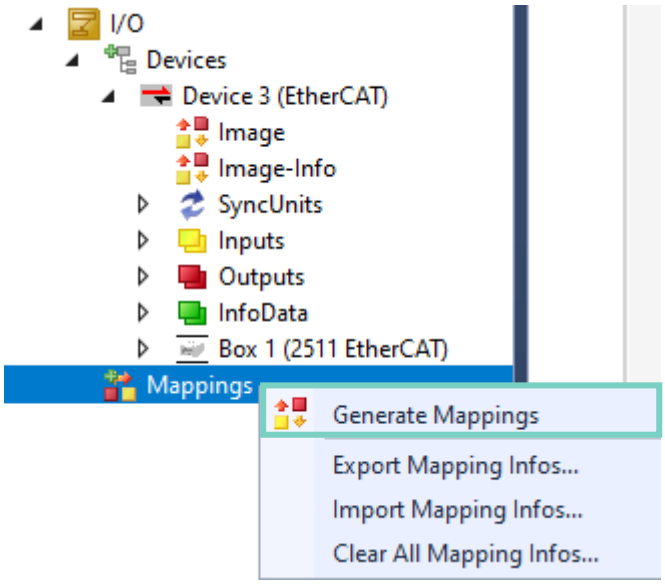
Assignment:

MAIN.statusByte → READY_STATUS
MAIN.ctrlByteA → CONTROL_A
MAIN.ctrlByteB → CONTROL_B

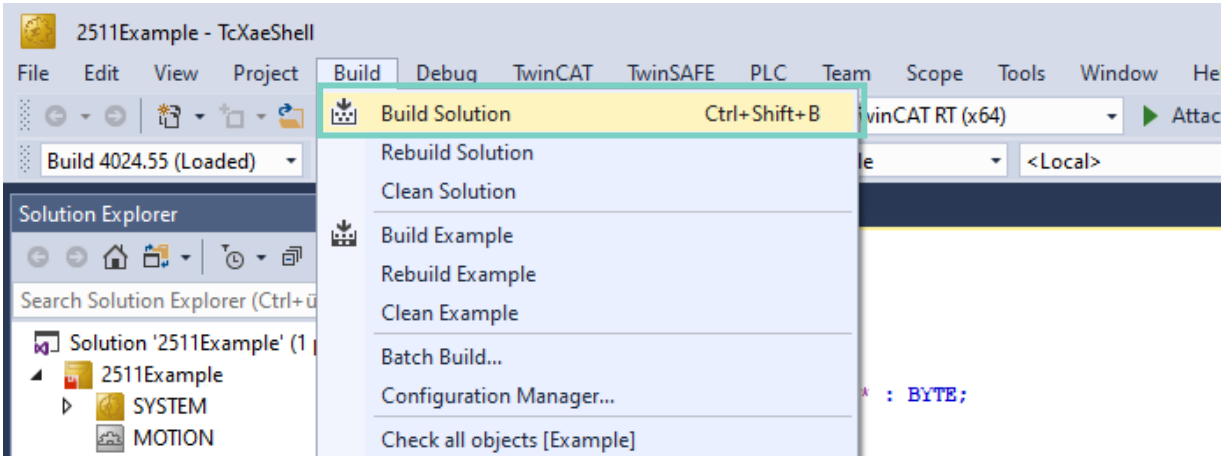




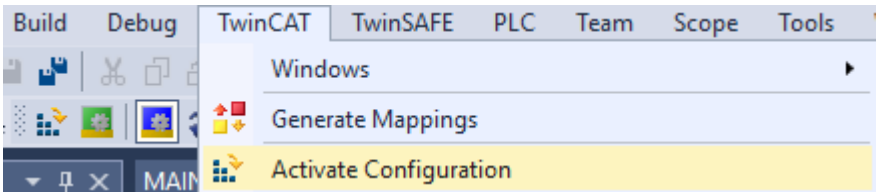
➢ Right-click ***Mappings*** → ***Generate Mapping***:

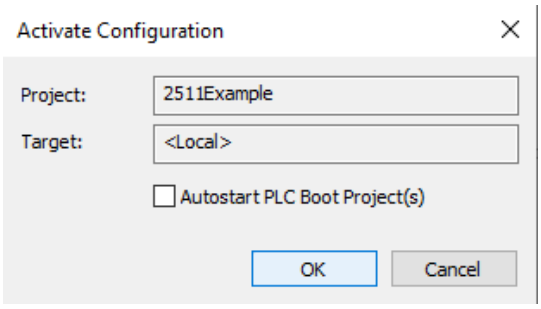


- Goto ***Build*** → ***Build Solution*** to build the project:

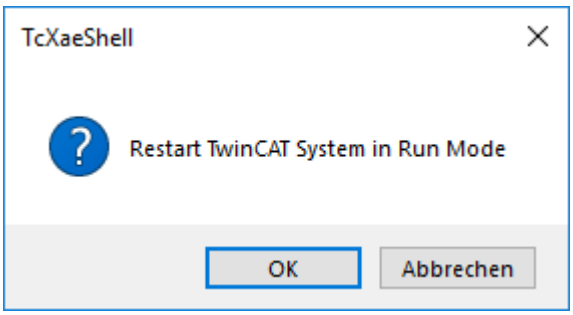


- Activate configuration via ***TwinCAT*** → ***Activate Configuration***

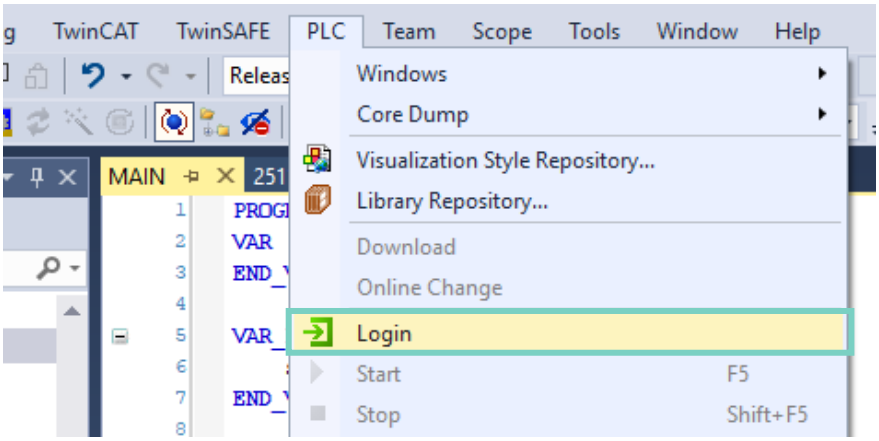




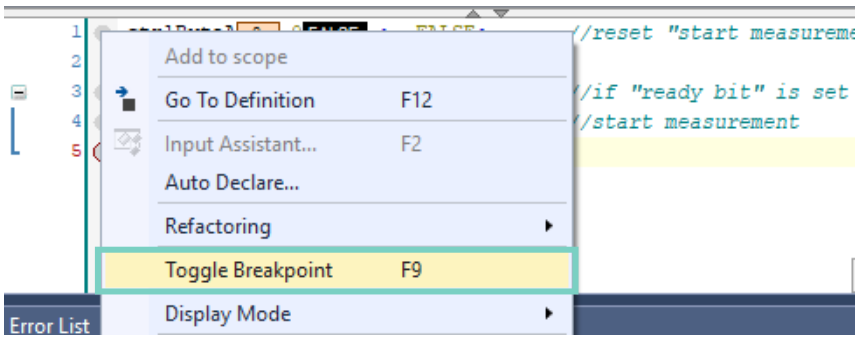
- Confirm starting in **Run Mode**:

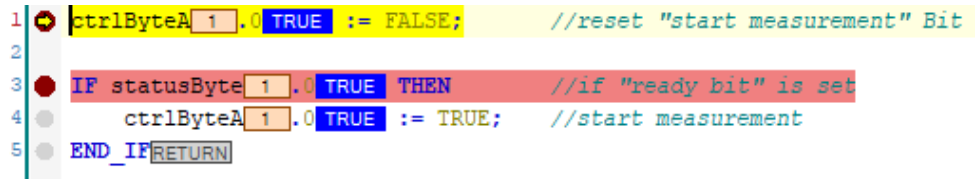


➢ Goto ***PLC*** → ***Login***: and if asked, confirm that program should be downloaded into the controller



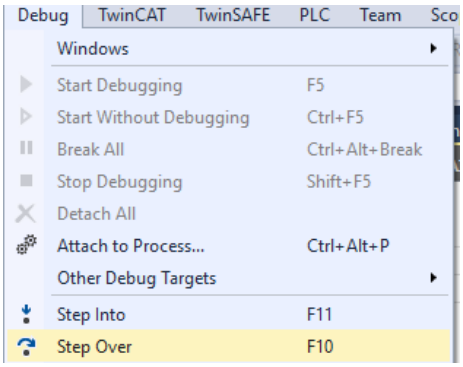
➢ Set a breakpoint (F9 or right-click → Toggle Breakpoint) to the first line to control the program execution step by step:





➢ Press the F5 key or click on the green start symbol to start the program execution:

- Execute the program line by line with the key **F10** oder via ***Debug*** → ***Step over***



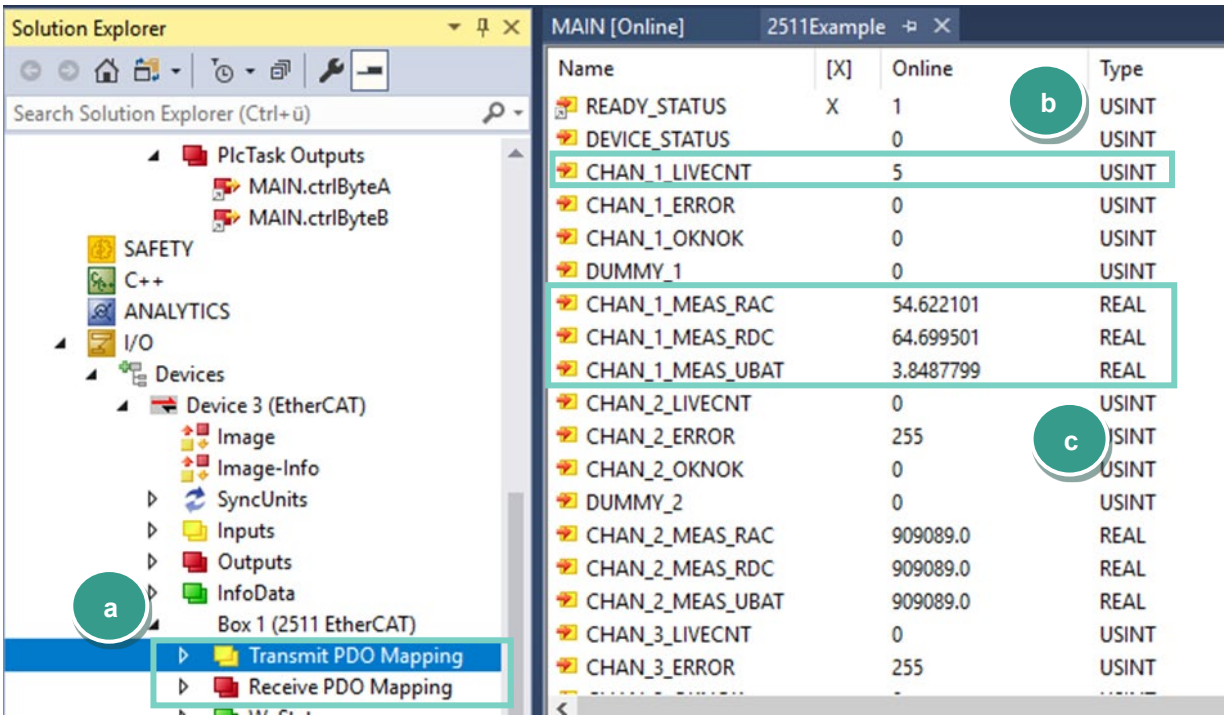
- The first bit of the control byte A will be set and a measurement will be started if the first bit of the status byte (ready bit) is set:

```
1  ctrlByteA[0].0 FALSE := FALSE;        //reset "start measurement" Bit
2
3  IF statusByte[1].0 TRUE THEN          //if "ready bit" is set
4      ctrlByteA[0].0 FALSE := TRUE;     //start measurement
5  END_IF RETURN
```

```
1  ctrlByteA[1].0 TRUE := FALSE;         //reset "start measurement" Bit
2
3  IF statusByte[1].0 TRUE THEN          //if "ready bit" is set
4      ctrlByteA[1].0 TRUE := TRUE;      //start measurement
5  END_IF RETURN
```

- To start another measurement the “start measurement” bit has to be reset (first line). Otherwise the “ready bit” will remain false and no measurement will be executed.

- In the project tree select ***Box_1 (2511 EtherCAT)*** → ***Transmit PDO Mapping*** (a) to control the results. The CHAN_1_LIVECNT increments with each measurement (b) and the current measuremet values (c) are shown for each channel


Solution Explorer
Search Solution Explorer (Ctrl+ü)
PlcTask Outputs
MAIN.ctrlByteA
MAIN.ctrlByteB
SAFETY
C++
ANALYTICS
I/O
Devices
Device 3 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Outputs
InfoData
Box 1 (2511 EtherCAT)
Transmit PDO Mapping
Receive PDO Mapping
a
MAIN [Online]
2511Example
Name
[X]
Online
Type
READY_STATUS X 1 USINT
DEVICE_STATUS 0 USINT
CHAN_1_LIVECNT 5 USINT
CHAN_1_ERROR 0 USINT
CHAN_1_OKNOK 0 USINT
DUMMY_1 0 USINT
CHAN_1_MEAS_RAC 54.622101 REAL
CHAN_1_MEAS_RDC 64.699501 REAL
CHAN_1_MEAS_UBAT 3.8487799 REAL
CHAN_2_LIVECNT 0 USINT
CHAN_2_ERROR 255 USINT
CHAN_2_OKNOK 0 USINT
DUMMY_2 0 USINT
CHAN_2_MEAS_RAC 909089.0 REAL
CHAN_2_MEAS_RDC 909089.0 REAL
CHAN_2_MEAS_UBAT 909089.0 REAL
CHAN_3_LIVECNT 0 USINT
CHAN_3_ERROR 255 USINT
b
c

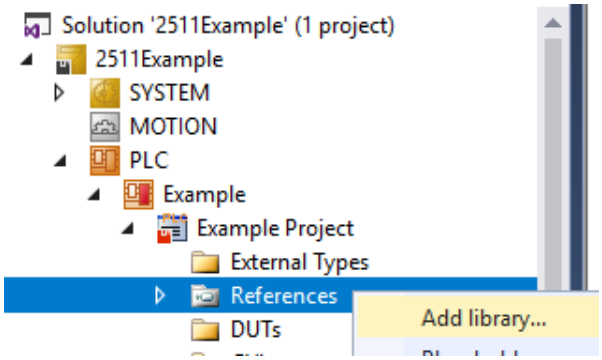
5. Further Examples

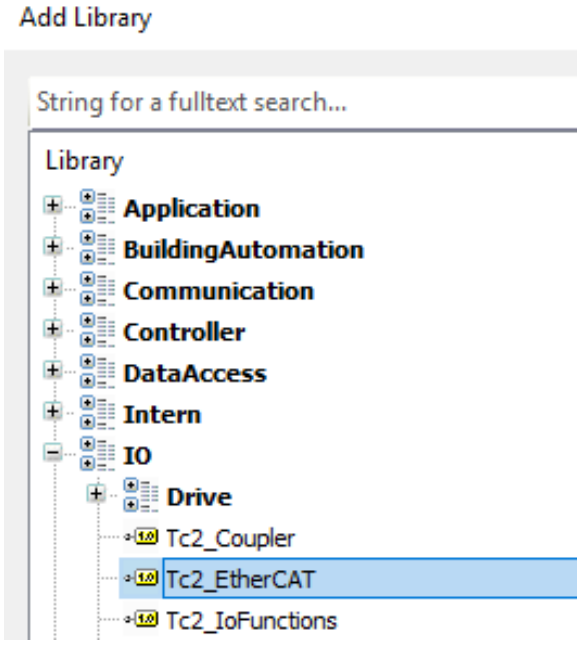
5.1 Read and Write of 'real' data types

Example 2: Set and Get the Lower Limit R_{DC} for Channel 1

This example shows you how to write and read the Lower Limit R_{DC} for Channel 1

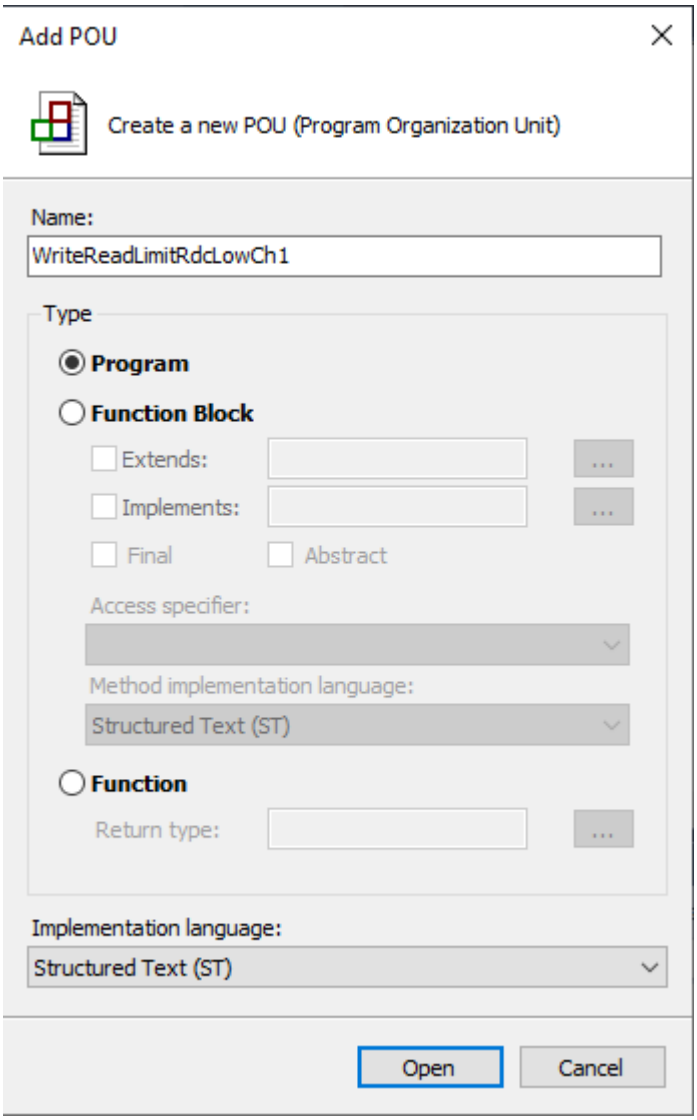
- Add the ***Tc2_EtherCAT*** library to your project to be able to use *FB_EcCoESdoRead* and *FB_EcCoESdoWrite* function blocks via ***References*** → ***Add library***



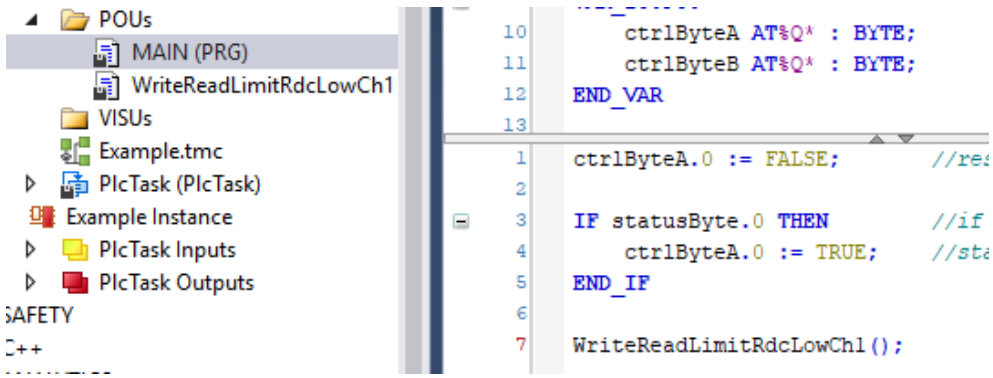


- Add a new **POU** (Program Organization Unit)

➢ Rename it to ***WriteReadLimitRdcLowCh1*** and click ***OK***:



➢ Insert the call of the ***WriteReadLimitRdcLowCh1*** in the ***MAIN*** POU:

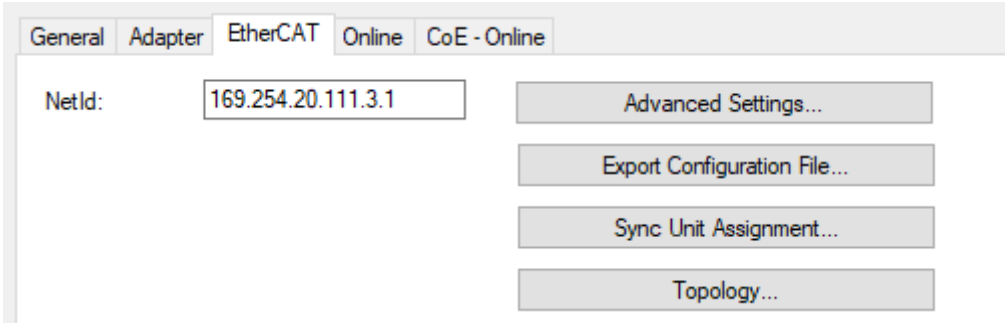


➢ Type in the following code into the created ***WriteReadLimitRdcLowCh1*** POU

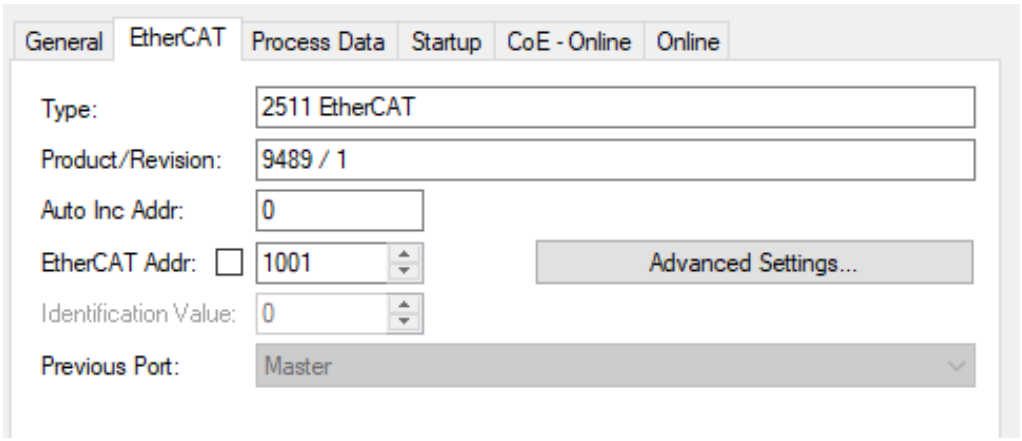
Source code:

```
PROGRAM WriteReadLimitALow
VAR
	fbSdoWrite		: FB_EcCoESdoWrite;
	fbSdoRead		: FB_EcCoESdoRead;
	sNetId			: T_AmsNetId := '169.254.20.111.3.1';  // see note 1 below
	nSlaveAddr		: UINT := 1001;                        // see note 2 below
	nIndex			: WORD := 16#2409;                     // CoE Object - Limit A Lower Value
	nSubIndex		: BYTE := 0;                           // is always 0
	fLimitRdcLowCh1	: REAL := 1.23;                        // data to be written to 2511
	bExecute		: BOOL := TRUE;
	bError			: BOOL;
	nErrId			: UDINT;
END_VAR

fbSdoWrite(
	sNetId		:= sNetId,
	nSlaveAddr	:= nSlaveAddr,
	nIndex		:= nIndex,
	nSubIndex	:= nSubIndex,
	pSrcBuf		:= ADR(fLimitRdcLowCh1),
	cbBufLen	:= SIZEOF(fLimitRdcLowCh1),
	bExecute	:= bExecute
);

IF NOT fbSdoWrite.bBusy THEN
	bExecute := FALSE;
	IF NOT bError THEN
		(* write successful *)
		bError := FALSE;
		nErrId := 0;
	ELSE
		(* write failed *)
		bError := fbSdoWrite.bError;
		nErrId := fbSdoWrite.nErrId;
	END_IF

	fbSdoWrite(bExecute := FALSE);
END_IF

fLimitRdcLowCh1 := 0.0;

fbSdoRead(sNetId:= sNetId,nSlaveAddr :=nSlaveAddr, nIndex:=nIndex, nSubIndex :=nSubIndex,
pDstBuf:= ADR(fLimitRdcLowCh1), cbBufLen:=SIZEOF(fLimitRdcLowCh1), bExecute:=TRUE);
bError:=fbSdoRead.bError;
nErrId:=fbSdoRead.nErrId;
```

Note 1: You will find the ***NetId*** if you click your EtherCAT master device in the project tree and select the tab **EtherCAT:**



Note 2: You will find the EtheCAT slave address if you click the 2511 device in the project tree and select the tab **EtherCAT:**



- Build the project via ***Build*** → ***Build Solution***, click on the **Login** symbol and set a break point (F9) in the first code line:

```
fbSdoWrite(
        sNetId := sNetId,
        nSlaveAddr := nSlaveAddr,
        nIndex := nIndex,
        nSubIndex := nSubIndex,
        pSrcBuf := ADR(fLimitRdcLowCh1),
        cbBufLen := SIZEOF(fLimitRdcLowCh1),
        bExecute := bExecute
);
```

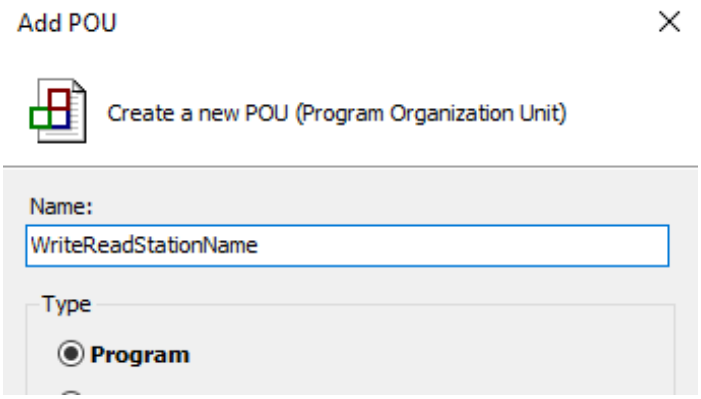
- Start the program execution with the ***F5*** key or via ***PLC*** → ***Start*** and go step for step (***F10***) through the whole program until you reach the last line. Check if the witten und read values are identical:

```
fLimitRdcLowCh1 := 0.0;

fbSdoRead(sNetId := sNetId, nSlaveAddr
bError :=fbSdoRead.bError ;
nErrId :=fbSdoRead.nErrId ;
```

5.2 Read and Write of 'string' data types

Example 3: Write and read the station name of the 2511:

- Create a new POU as described above and name it ***WriteReadStationName***:



- Write or copy the following source code into the new POU:

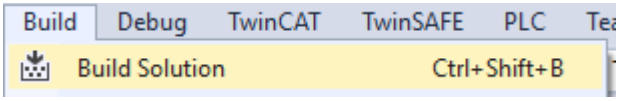
```
PROGRAM ReadSerial
VAR
	fbSdoWrite		: FB_EcCoESdoWrite;
	fbSdoRead		: FB_EcCoESdoRead;
	sNetId			: T_AmsNetId := '169.254.20.111.3.1';  // see note 1 in the previous section
	nSlaveAddr		: UINT := 1001;                        // see note 2 in the previous section
	nIndex			: WORD := 16#246A;                     // CoE Object – Station Name
	nSubIndex		: BYTE := 0;                           // is always 0
	abStationName	: STRING := 'New Name';                // station name to write
	bExecute		: BOOL := TRUE;
	bError			: BOOL;
	nErrId			: UDINT;
END_VAR

fbSdoWrite(                                                //write new station name
	sNetId		:= sNetId,
	nSlaveAddr	:= nSlaveAddr,
	nIndex		:= nIndex,
	nSubIndex	:= nSubIndex,
	pSrcBuf		:= ADR(abStationName),
	cbBufLen	:= 20,
	bExecute	:= bExecute);

IF NOT fbSdoWrite.bBusy THEN
		bExecute := FALSE;
		IF NOT bError THEN
			bError := FALSE;
			nErrId := 0;
		ELSE
			bError := fbSdoWrite.bError;
			nErrId := fbSdoWrite.nErrId;
		END_IF
		fbSdoWrite(bExecute := FALSE);
END_IF

abStationName := 'CleartoReadAgain';                       //clear variable to read again
```

```
fbSdoRead(                                          //read current station name
        sNetId := sNetId,
        nSlaveAddr := nSlaveAddr,
        nIndex := nIndex,
        nSubIndex := nSubindex,
        pDstBuf := ADR(abStationName),
        cbBufLen := 20,
        bExecute := bExecute);

bError := fbSdoRead.bError;
nErrId := fbSdoRead.nErrId;
```

- Insert a call for the POU in the **MAIN** block:

```
END_IF

WriteReadStationName();
```

- Build the Project via ***Build → Build Solution:***



- Log in ***PLC → Login***, set a break point to the first line and click ***PLC → Start (F5)*** to run the program. Press **F10** to execute the program step by step

```
1  fbSdoWrite(                                      //write new station name
2          sNetId  '169.254.96'      := sNetId  '169.254.96',
3          nSlaveAddr  1001     := nSlaveAddr  1001,
4          nIndex  9322         := nIndex  9322,
5          nSubIndex  0      := nSubIndex  0,
6          pSrcBuf  16#FFFFC5864C4A2A08       := ADR(abStationName  'New Name'),
7          cbBufLen  20          := 20,
8          bExecute  TRUE       := bExecute  TRUE);
9
10 IF NOT fbSdoWrite.bBusy  TRUE  THEN
```

- First the new station name “New Name” will be written in the device (line 1 – 8)

```
20 END_IF
21
22 abStationName  'CleartoRea'  := 'CleartoReadAgain';     //clear station
23
24 fbSdoRead(                                    //read current station name
25     sNetId  ''  := sNetId  '169.254.96',
```

- The variable “*abStationName”* will be overwritten in line 22

- Now the station name will be read again. Make sure the station name on line 29 is the same as in line 6 to confirm that the task was succesfull

```
24  fbSdoRead(                                    //read current station name
25      sNetId '169.254.96 ▸ := sNetId '169.254.96 ▸ ,
26      nSlaveAddr 1001 := nSlaveAddr 1001 ,
27      nIndex 9322 := nIndex 9322 ,
28      nSubIndex 0 := nSubindex 0 ,
29      pDstBuf 16#FFFFC5864C4A2A08 := ADR(abStationName 'New Name' ),
30      cbBufLen 20 := 20,
31      bExecute TRUE := bExecute TRUE );
32
33  bError FALSE := fbSdoRead.bError FALSE ;
34  nErrId 0 := fbSdoRead.nErrId 0 ; RETURN
```