

OPERATION MANUAL

9311 EtherCAT Integration into TwinCAT

© 2024

burster
präzisionsmesstechnik gmbh & co kg
All rights reserved

Manufacturer:

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

P.O. Box 1432
76587 Gernsbach
Germany

Valid from: **01.03.2024**
Applies to: **DIGIFORCE® 9311-V0x01, V2x01**

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

4517-BAETHERCATEN-5799-031527

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	14
5.1 Read and Write of 'real' data types	14
5.2 Write and Read of 'string' data types	21
5.3 Read of X-Coordinates	23

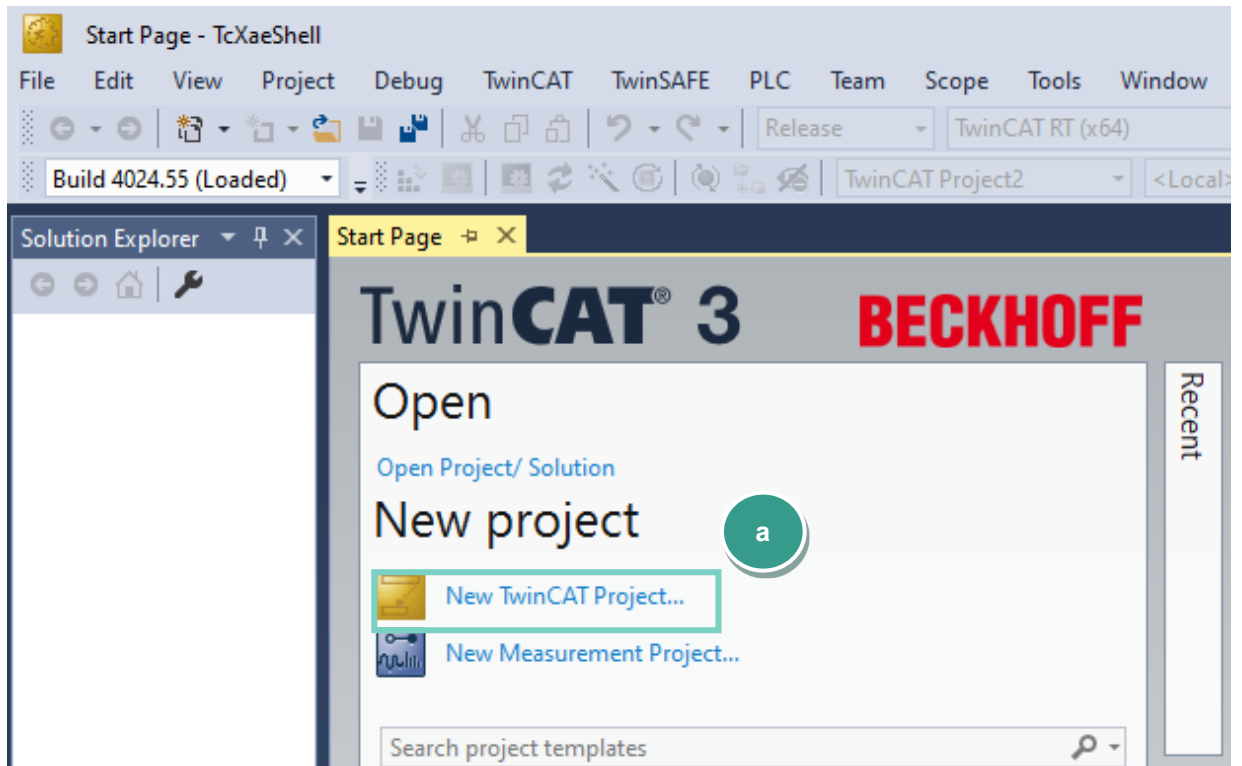
Introduction

This quick start guide describes an approach how you can configure the 9311 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 been extremely simplified to reach a better understanding. Therefore, you may have to complete them by checking of status, error, length values etc.

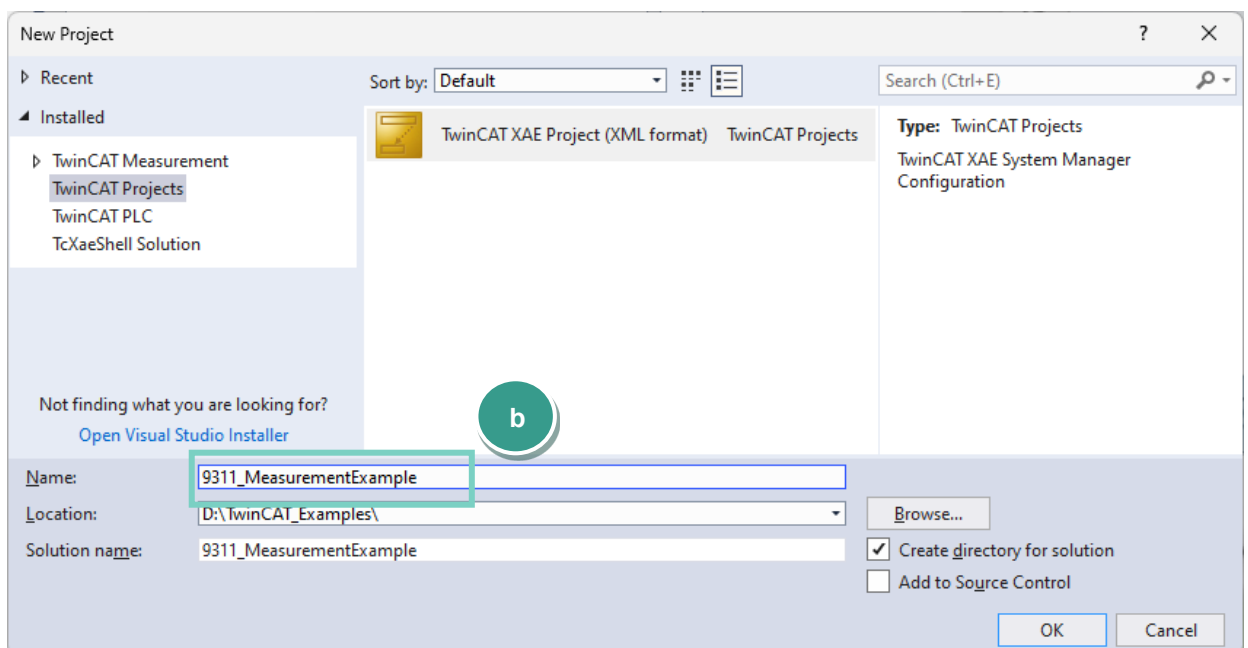
Please also note that you will have to use the OPERATION MANUAL DIGIFORCE® Model 9311 as well as 9311 EtherCAT Manual to get further information about input and output parameters (PDO as well as SDO data transfer)

1. Creating new project

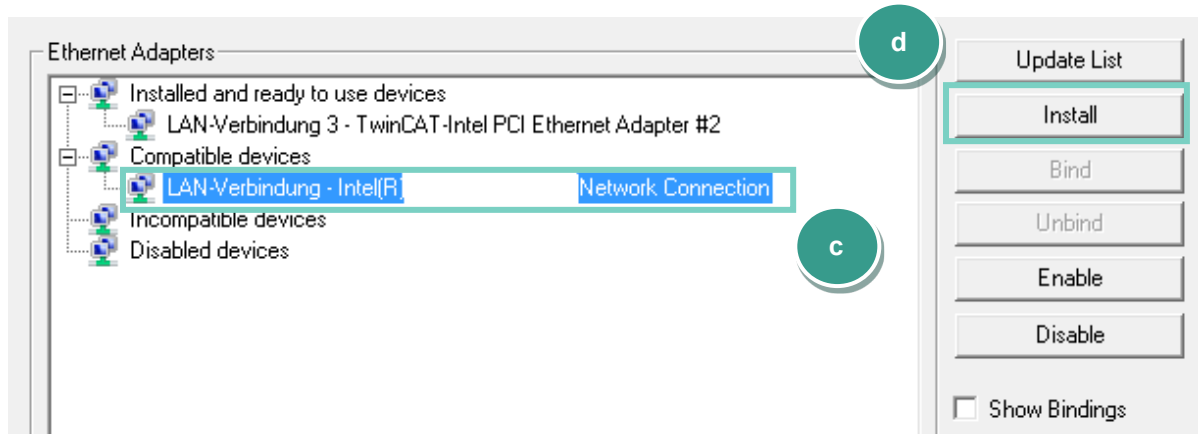
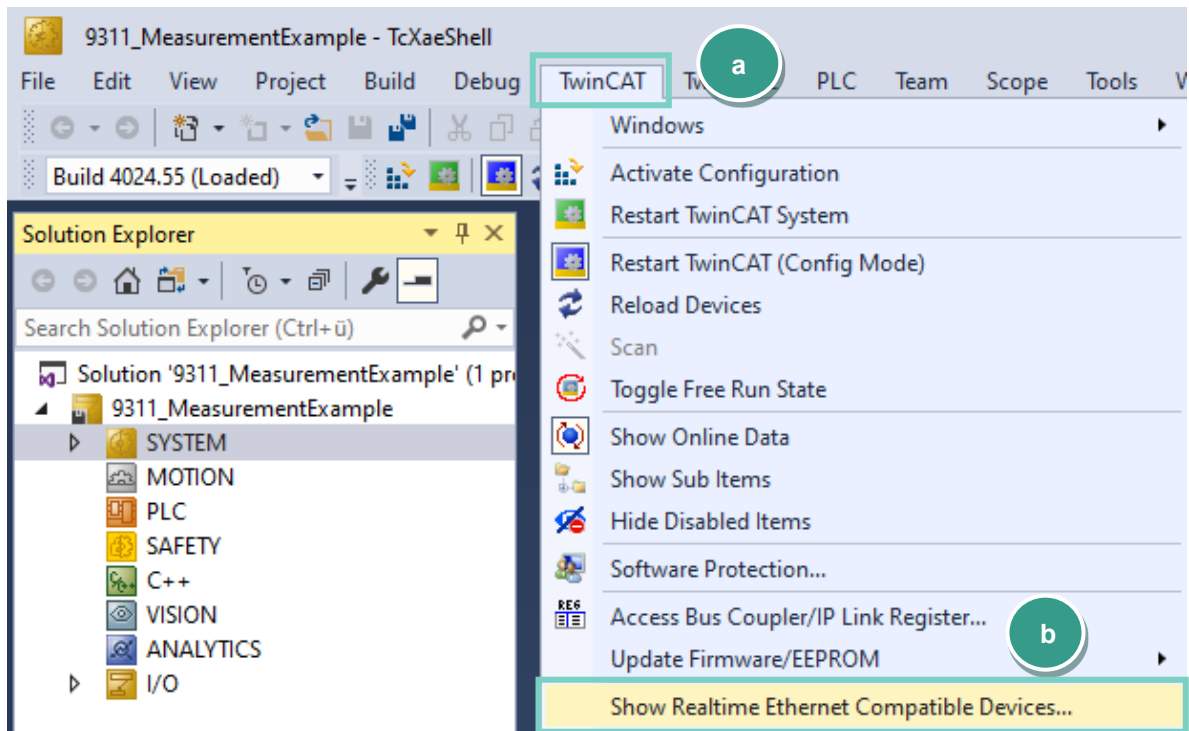
- Start the **TwinCAT XAE Shell** and click on **New TwinCAT Project** (a) (or via **File** → **New Project**)



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



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



*You can find information of supported network controllers on:

https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_overview/9309844363.html&id

2. Installation of ESI description files

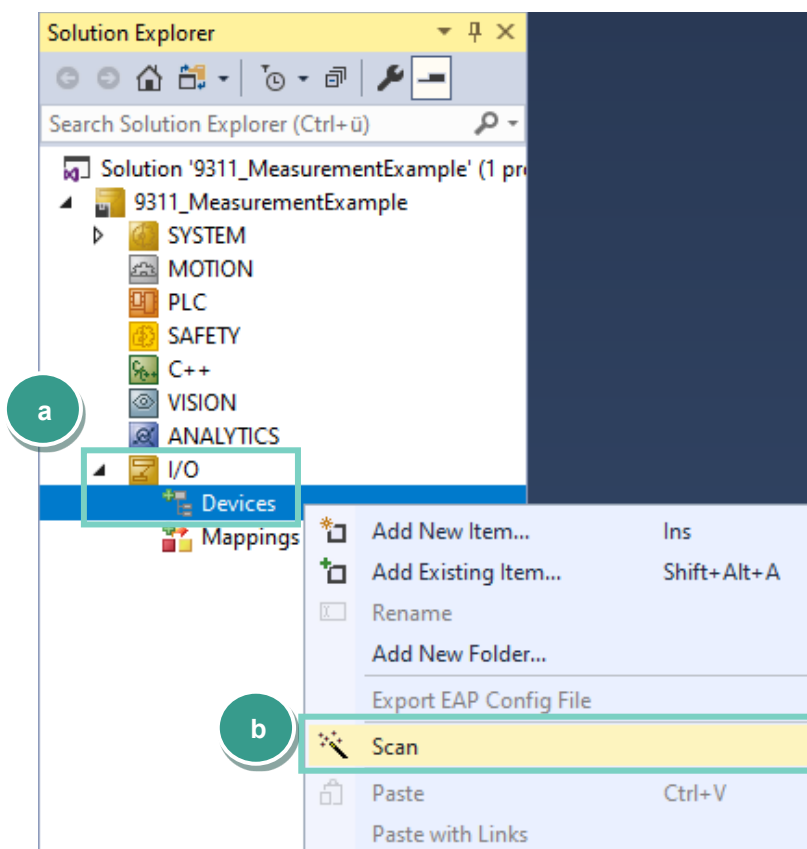
Note: Please make sure that your ESI file is compatible to the field bus firmware in the 9311 when you download an ESI file from our homepage <https://www.burster.com/>

- Copy the ESI file *burster_9311.xml* into directory *C:\TwinCAT\3.1\Config\Io\OnboardIo* and additionally into *C:\TwinCAT\3.1\Config\Io\EtherCAT*

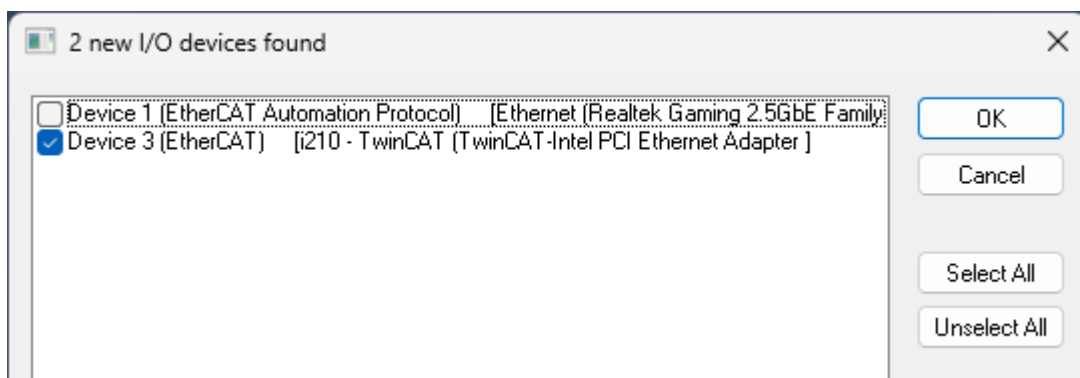
Note: If you use the TwinCAT 2 The ESI directory would be *C:\TwinCAT\Io\EtherCAT*

3. Scan EtherCAT devices

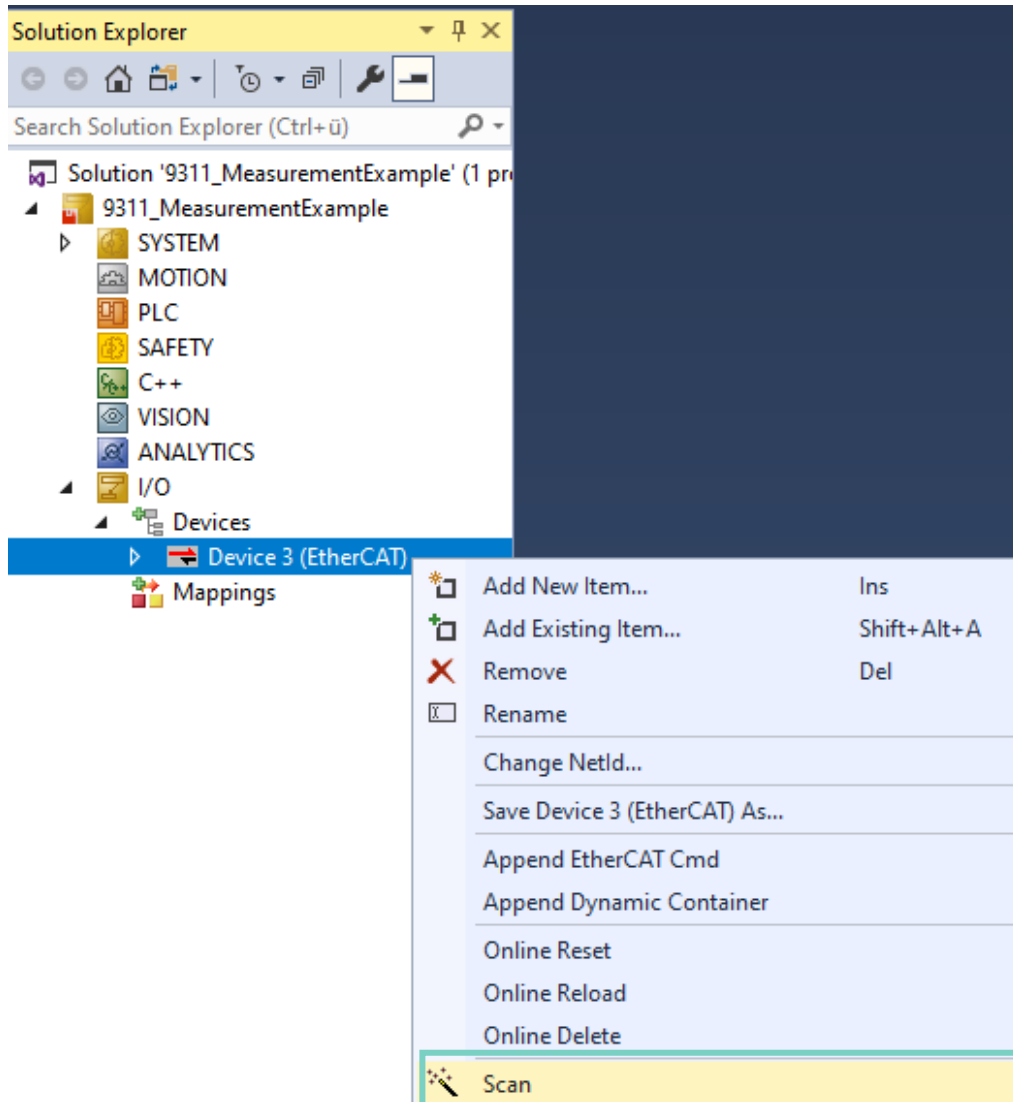
- Connect the 9311 to your EtherCAT master, right click *I/O* → **Devices** (a) in the project tree und select **Scan** (b):



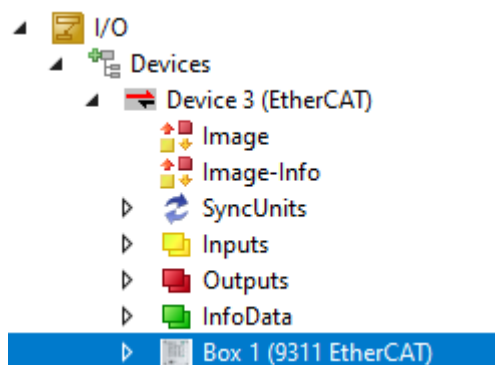
- Now, you can select an EtherCAT compatible device in the new window and click OK:



- At this point you have to 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 the TwinCAT was not able to find the ESI file, confirm the question to use online description and after a while you should be able to see the 9311 device in the project tree:



- Confirm the request to activate **Free Run**

- To see the process data, please click on the 9311 in the project tree (a) and select the **Process Data** tab (b):

The screenshot shows the Siemens TIA Portal interface for configuring a 9311 EtherCAT device. The 'Process Data' tab is active, displaying the following data:

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	3	Outputs	
3	92	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	92.0	Inputs	MF	3	0
0x1600	3.0	Outputs	MF	2	0

PDO Assignment (0x1C12):

0x1600

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default
0x2001:01	0.1	0.0	9311_OUT_READY	BIT	
0x2001:02	0.1	0.1	9311_OUT_OK	BIT	
0x2001:03	0.1	0.2	9311_OUT_NOK	BIT	
0x2001:04	0.1	0.3	9311_OUT_NOK_ONL	BIT	
0x2001:05	0.1	0.4	9311_OUT_S1	BIT	

Device List:

Name	Online	Type	Size	> Addr...	In/Out	User ID	Linked to
9311_OUT_READY	0	BIT	0.1	54.0	Input	0	
9311_OUT_OK	0	BIT	0.1	54.1	Input	0	
9311_OUT_NOK	0	BIT	0.1	54.2	Input	0	
9311_OUT_NOK_...	0	BIT	0.1	54.3	Input	0	
9311_OUT_S1	0	BIT	0.1	54.4	Input	0	
9311_OUT_S2	0	BIT	0.1	54.5	Input	0	
9311_OUT_ADJ1	0	BIT	0.1	54.6	Input	0	
9311_OUT_ADJ6	0	BIT	0.1	54.7	Input	0	
9311_OUT_ADJ2	0	BIT	0.1	55.0	Input	0	
9311_OUT_ADJ3	0	BIT	0.1	55.1	Input	0	
9311_OUT_ADJ4	0	BIT	0.1	55.2	Input	0	
9311_OUT_ADJ5	0	BIT	0.1	55.3	Input	0	
9311_OUT_Glob...	0	BIT	0.1	56.0	Input	0	
9311_OUT_Overl...	0	BIT	0.1	56.1	Input	0	
9311_OUT_Wind...	0	BIT	0.1	56.2	Input	0	
9311_OUT_Wind...	0	BIT	0.1	56.3	Input	0	
9311_OUT_Wind...	0	BIT	0.1	56.4	Input	0	
9311_OUT_Thres...	0	BIT	0.1	56.5	Input	0	
9311_OUT_Thres...	0	BIT	0.1	56.6	Input	0	
9311_OUT_Trape...	0	BIT	0.1	56.7	Input	0	
9311_OUT_Trape...	0	BIT	0.1	57.0	Input	0	

Error List:

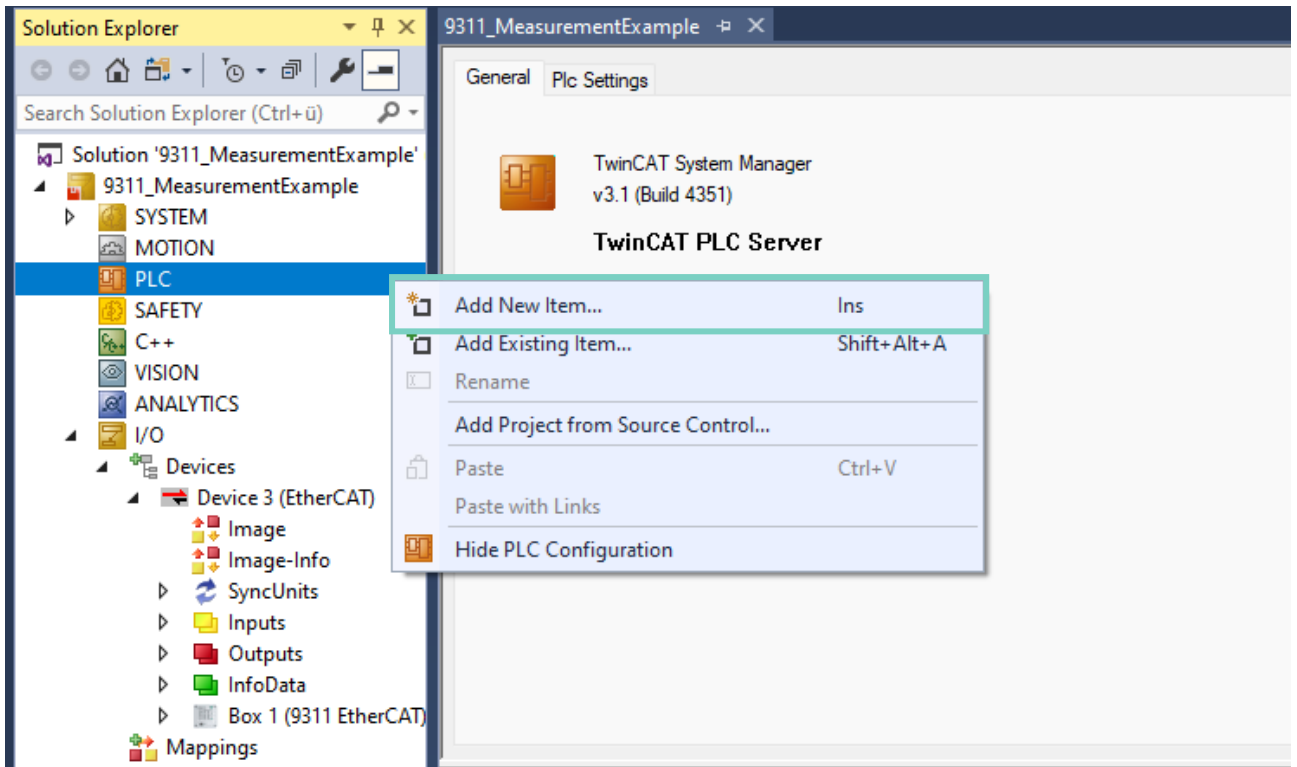
Entire Solution | 0 Errors | 0 Warnings | 0 of 2 Messages | Clear | Build + IntelliSense

Description	Project	File	Line
-------------	---------	------	------

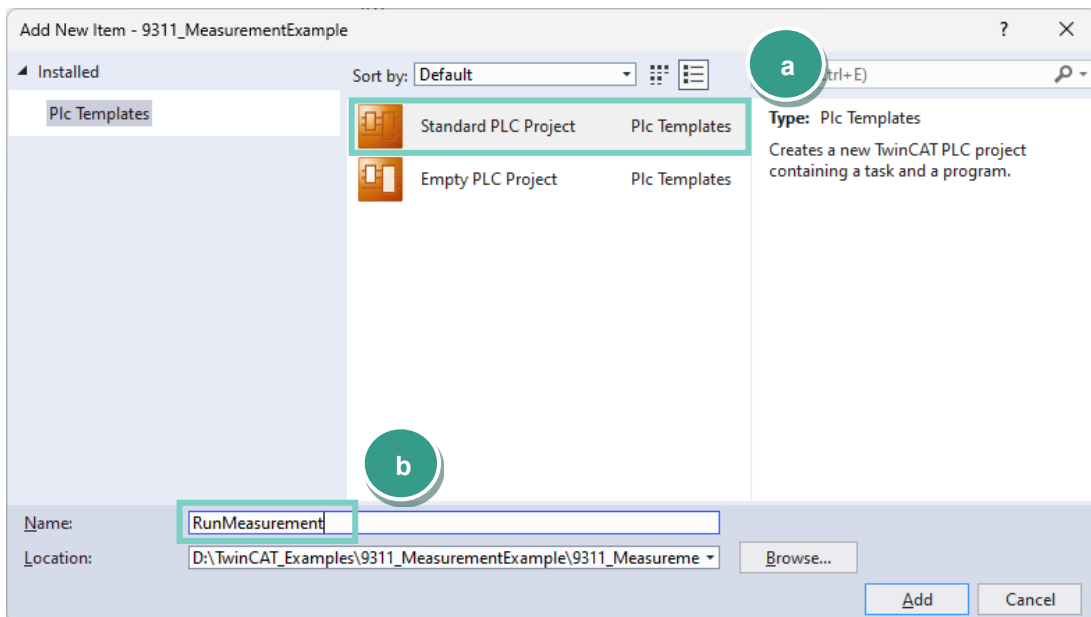
4. Create a sample program

In this section, you will learn how to create a simple PLC program to perform periodical measurements via PDO (Process Data Object). You will need to refer to section *EtherCAT data protocol* in 9311 EtherCAT operation manual to understand the meaning of input bytes.

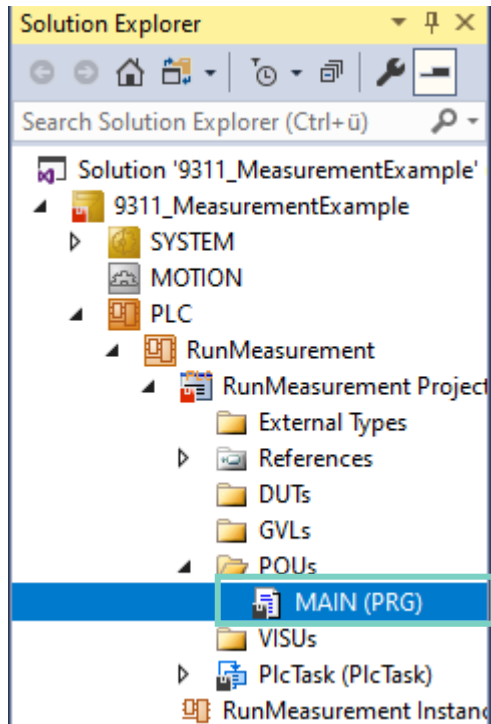
- Right-click **PLC** in the project tree and select **Add New Item...**



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



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



Example 1: Reading and Writing of PDOs

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

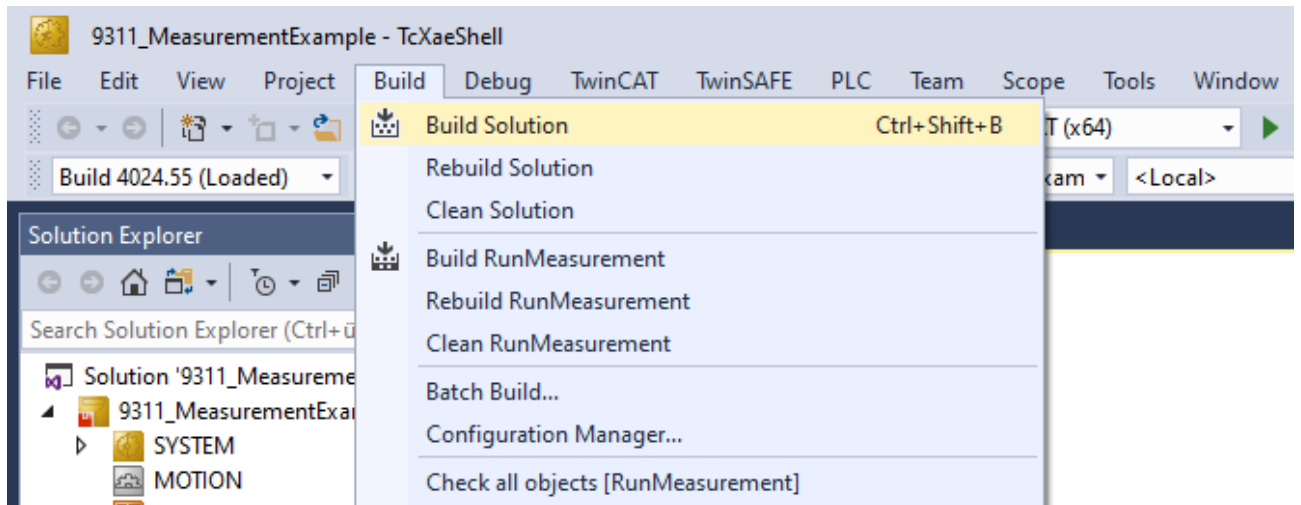
The screenshot shows the IDE with the 'MAIN (PRG)' file selected in the Solution Explorer. The code in the editor is as follows:

```

1 PROGRAM MAIN
2 VAR
3 ton : TON; // Timer on-delay
4 nStep : INT;
5 END_VAR
6
7 VAR_INPUT
8   ready AT%I* : BOOL;
9 END_VAR
10
11 VAR_OUTPUT
12   measStart AT%Q* : BOOL;
13 END_VAR
14
15 CASE nStep OF
16 0:
17   ton(IN := TRUE, PT := T#1S);
18 IF (ton.Q) THEN
19   ton (IN := FALSE);
20   nStep := 1;
21 END_IF
22 1:
23 IF ready = TRUE THEN // Ready bit is set
24   measStart := TRUE; // Start a new measurement
25 ELSE
26   measStart := FALSE; // Stop the measurement
27 END_IF
28 ton (IN := TRUE);
29 nStep := 0;
30 END_CASE

```

- Goto **Build** → **Build Solution**

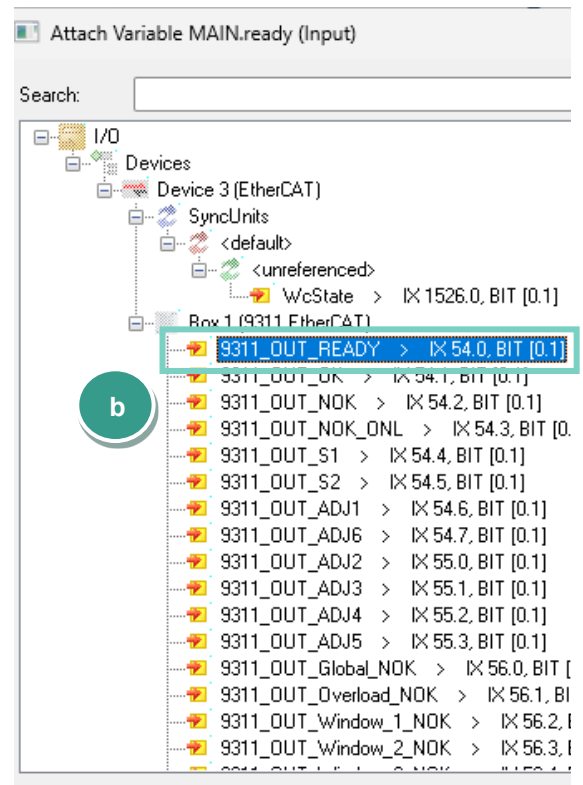
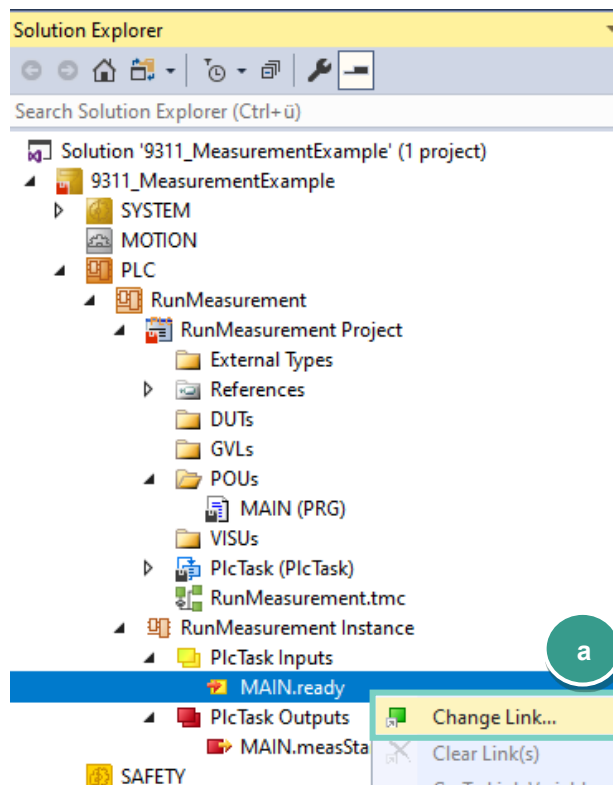


- Assign the **input** and **output** variables to the corresponded PDOs with the right-click on a variable and select **Change Link...**(a) from the context menu, select a corresponded PDO (b)

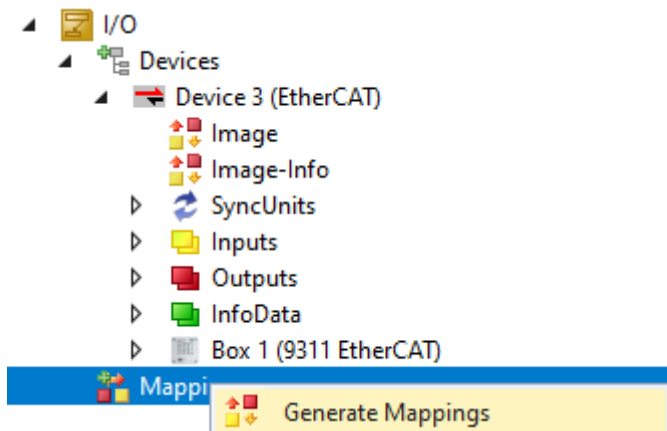
Assignment:

MAIN.ready → *9311_OUT_READY*

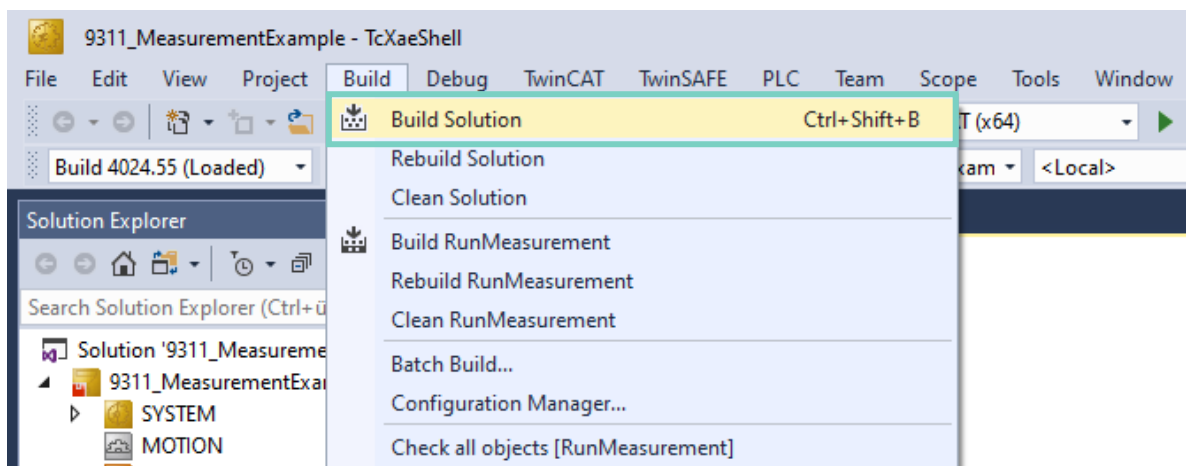
MAIN.measStart → *9311_IN_START*



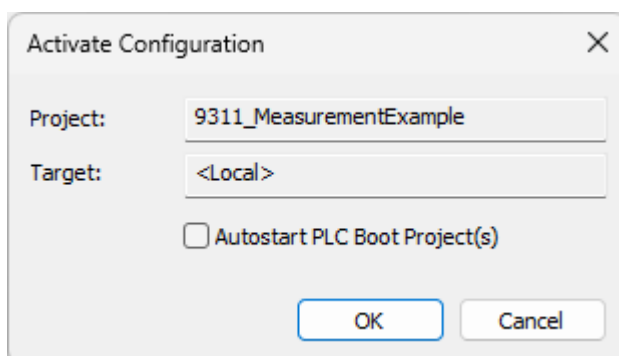
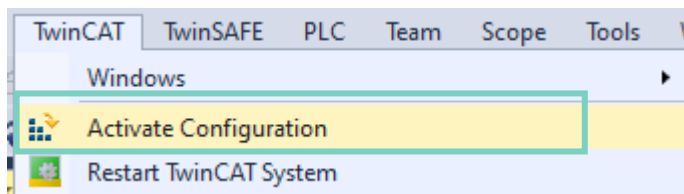
- Right-click **Mappings** → **Generate Mapping**:



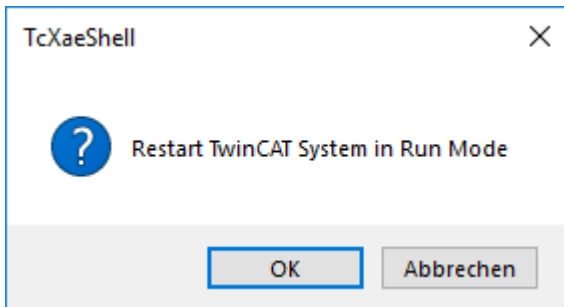
- Goto **Build** → **Build Soution** to build the project:



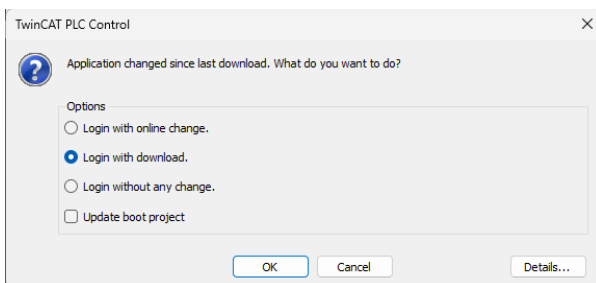
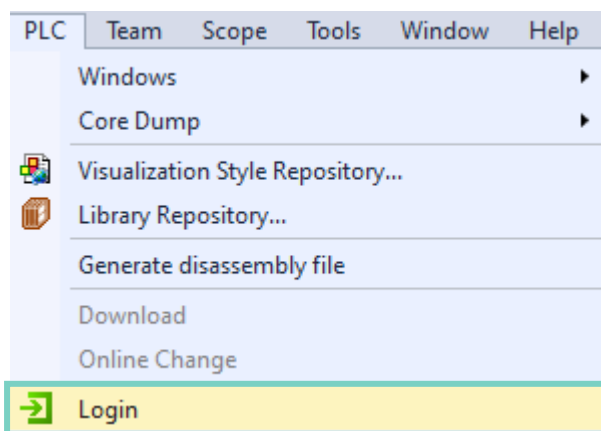
- Activate configuration via **TwinCAT** → **Activate Configuration**



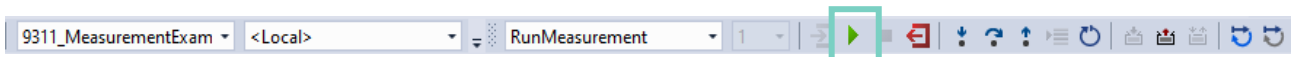
- Confirm starting in **Run Mode**:



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



- Press the **F5** key or click on the green start symbol to start the program execution



To perform a one-second measurement we use a TON (Time on-delay) function block. When one second is over (*ton.Q* is set) the program goes over from case 0 to case 1 (*nStep := 1*). In step 1 we check if the *ready* bit is set and if so, we start a new measurement, otherwise we stop it. After that, we restart the timer, go to step 0 again and so on.

Note: If measurement does not start: please make sure that **Control via EtherCAT** is selected in the deviceEtherCAT menu: **Basic setup** → **EtherCAT**.

5. Further examples

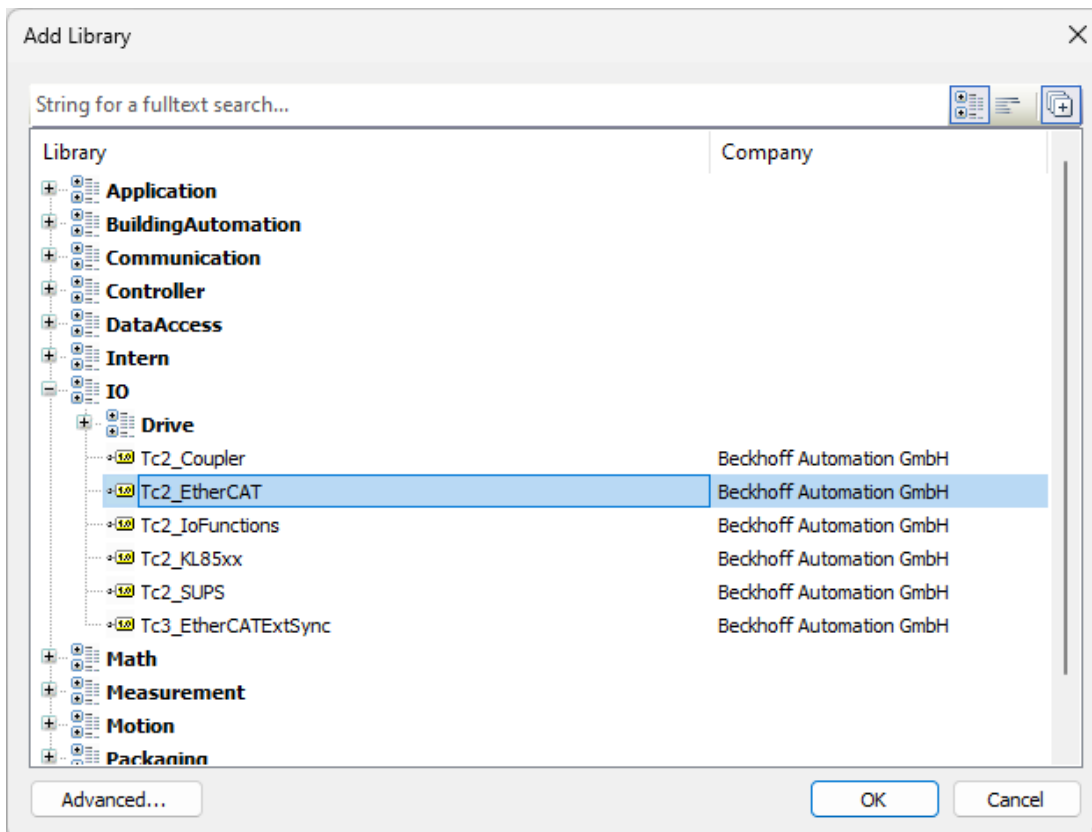
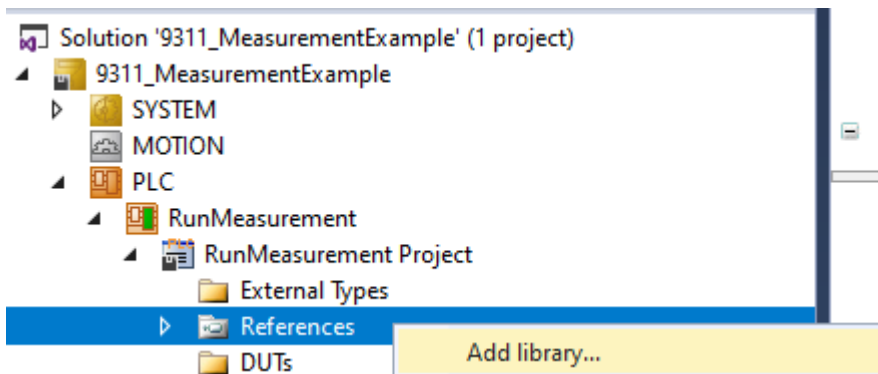
In this chapter, we will perform read & write operations on SDO (Service Data Objects). These are described in section SDO – Service Data Objects of the 9311 EtherCAT operation manual.

5.1 Read and Write of 'real' data types

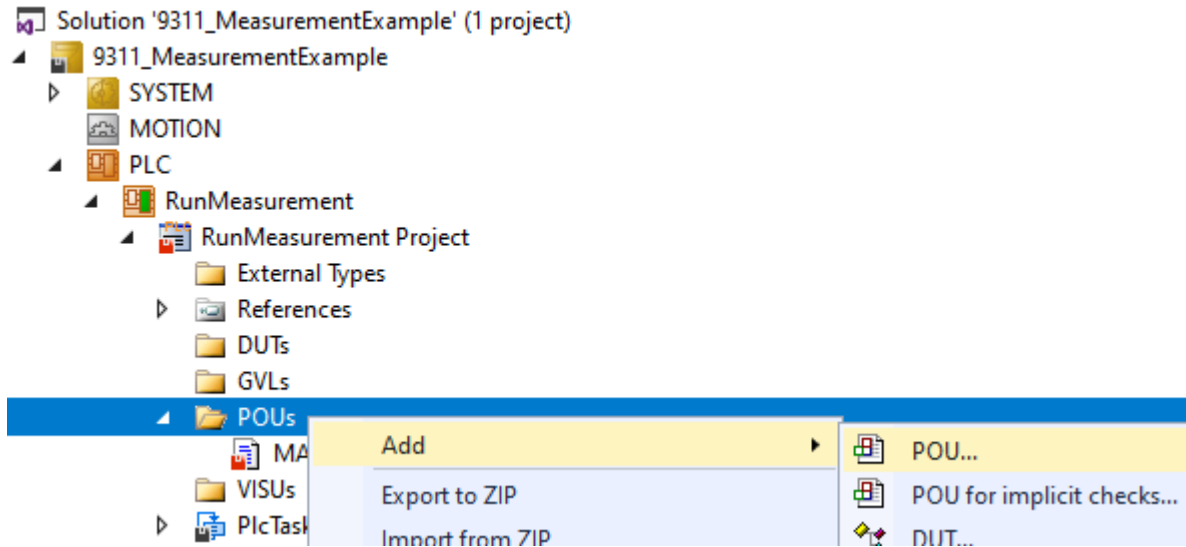
Example 2: Set and get the limits of an evaluation window (Index 0x2039, Subindices 11-15)

This example shows you how to write and read the limits of evaluation window 1

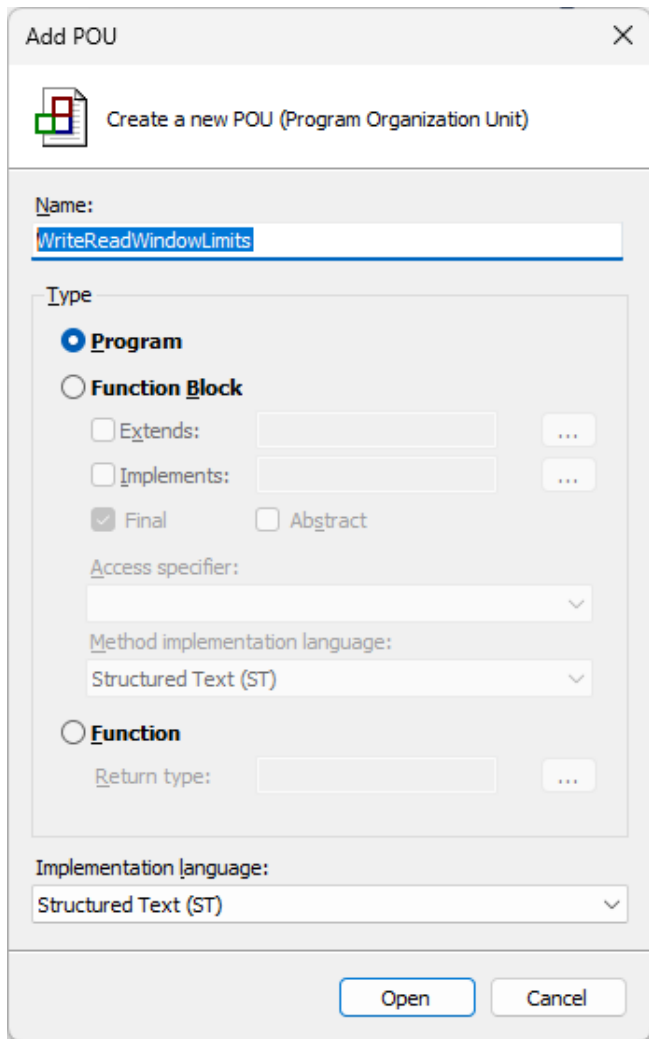
- Add the **Tc2_EtherCAT** library to your project to be able to use *FB_EcCoESdoRead* and *FB_EcCoESdoWrite* function blocks via **References** → **Add library**



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



- Rename it to **WriteReadWindowLimits** and click **Open**:



- Insert the call of the **WriteReadWindowLimits** in the **MAIN** POU:

```

4   ton(IN := TRUE, PT := T#1S);
5   IF (ton.Q) THEN
6       ton (IN := FALSE);
7       nStep := 1;
8   END_IF
9   1:
10  IF ready = TRUE THEN           // Ready bit is set
11      measStart := TRUE;       // Start a new measurement
12  ELSE
13      measStart := FALSE;     // Stop the measurement
14  END_IF
15  ton (IN := TRUE);
16  nStep := 0;
17  END_CASE
18  *)
19  WriteReadWindowLimits();

```

- Type in the following code into the created **WriteReadWindowLimits** POU

Source code:

```

PROGRAM WriteReadWindowLimits
VAR
    fbSdoWrite    : FB_EcCoESdoWrite;
    fbSdoRead     : FB_EcCoESdoRead;
    sNetId        : T_AmsNetId := '192.168.19.1.4.1'; // see note 1 below
    nSlaveAddr    : UINT := 1001; // see note 2 below
    bExecute      : BOOL := TRUE;
    bError        : BOOL;
    nErrId        : UDINT;
    xMin          : REAL; // Xmin coordinate of window 1
    xMax          : REAL; // Xmax coordinate of window 1
    yMin          : REAL; // Ymin coordinate of window 1
    yMax          : REAL; // Ymax coordinate of window 1
    xMinRead      : REAL; // Xmin coordinate has been read
    xMaxRead      : REAL; // Xmax coordinate has been read
    yMinRead      : REAL; // Ymin coordinate has been read
    yMaxRead      : REAL; // Ymax coordinate has been read
    event         : BYTE; // Acknowledgement for indices 11, 12, 13, 14
    nStep         : INT := 0;
END_VAR

xMin := 5.53;
xMax := 6.24;
yMin := 7.35;
yMax := 8.89;

CASE nStep OF
0:
fbSdoWrite(
    sNetId      := sNetId,
    nSlaveAddr := nSlaveAddr, // see note 2 in the previous section
    nIndex     := 16#2039,
    nSubIndex  := 11,

```



```
pSrcBuf      := ADR(xMin),
cbBufLen     := SIZEOF(xMin),
bExecute     := bExecute
);
bError := fbSdoWrite.bError;
nErrId := fbSdoWrite.nErrId;
nStep := 1;
1:
fbSdoWrite(bExecute := FALSE);
IF NOT fbSdoWrite.bBusy THEN
fbSdoWrite(
  sNetId      := sNetId,
  nSlaveAddr  := nSlaveAddr,
  nIndex      := 16#2039,
  nSubIndex   := 12,
  pSrcBuf     := ADR(xMax),
  cbBufLen    := SIZEOF(xMax),
  bExecute    := bExecute
);
bError := fbSdoWrite.bError;
nErrId := fbSdoWrite.nErrId;
nStep := 2;
END_IF;

2:
fbSdoWrite(bExecute := FALSE);
IF NOT fbSdoWrite.bBusy THEN
fbSdoWrite(
  sNetId      := sNetId,
  nSlaveAddr  := nSlaveAddr,
  nIndex      := 16#2039,
  nSubIndex   := 13,
  pSrcBuf     := ADR(yMin),
  cbBufLen    := SIZEOF(yMin),
  bExecute    := bExecute
);
bError := fbSdoWrite.bError;
nErrId := fbSdoWrite.nErrId;
nStep := 3;
END_IF

3:
fbSdoWrite(bExecute := FALSE);
IF NOT fbSdoWrite.bBusy THEN
fbSdoWrite(
  sNetId      := sNetId,
  nSlaveAddr  := nSlaveAddr,
  nIndex      := 16#2039,
  nSubIndex   := 14,
  pSrcBuf     := ADR(yMax),
  cbBufLen    := SIZEOF(yMax),
  bExecute    := bExecute
);
bError := fbSdoWrite.bError;
nErrId := fbSdoWrite.nErrId;
nStep := 4;
END_IF

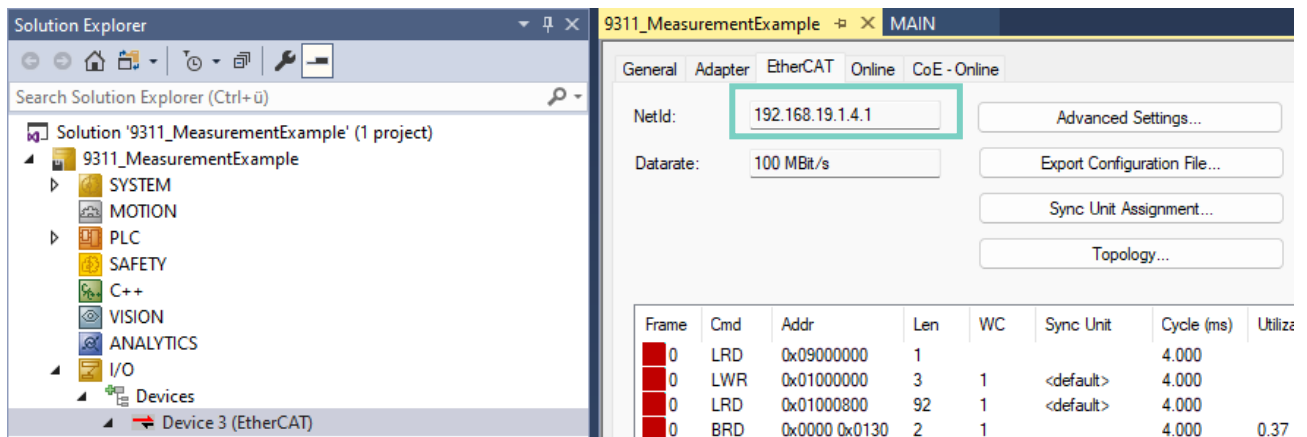
4:
fbSdoWrite(bExecute := FALSE);
IF NOT fbSdoWrite.bBusy THEN
```

```
fbSdoWrite(
  sNetId      := sNetId,
  nSlaveAddr  := nSlaveAddr,
  nIndex      := 16#2039,
  nSubIndex   := 15,
  pSrcBuf     := ADR(event),
  cbBufLen    := SIZEOF(event),
  bExecute    := bExecute
);
nStep := 5;
END_IF
5:
fbSdoRead(bExecute := FALSE);
IF NOT fbSdoRead.bBusy THEN
fbSdoRead(sNetId:= sNetId,
          nSlaveAddr :=nSlaveAddr,
          nIndex:=16#2039,
          nSubIndex := 11,
          pDstBuf:= ADR(xMinRead),
          cbBufLen:=SIZEOF(xMinRead),
          bExecute:=bExecute
);
bError:=fbSdoRead.bError;
nErrId:=fbSdoRead.nErrId;
nStep := 6;
END_IF;
6:
fbSdoRead(bExecute := FALSE);
IF NOT fbSdoRead.bBusy THEN
fbSdoRead(sNetId:= sNetId,
          nSlaveAddr :=nSlaveAddr,
          nIndex:=16#2039,
          nSubIndex := 12,
          pDstBuf:= ADR(xMaxRead),
          cbBufLen:=SIZEOF(xMaxRead),
          bExecute:=bExecute
);
bError:=fbSdoRead.bError;
nErrId:=fbSdoRead.nErrId;
nStep := 7;
END_IF;
7:
fbSdoRead(bExecute := FALSE);
IF NOT fbSdoRead.bBusy THEN
fbSdoRead(sNetId:= sNetId,
          nSlaveAddr :=nSlaveAddr,
          nIndex:=16#2039,
          nSubIndex := 13,
          pDstBuf:= ADR(yMinRead),
          cbBufLen:=SIZEOF(yMinRead),
          bExecute:=bExecute
);
bError:=fbSdoRead.bError;
nErrId:=fbSdoRead.nErrId;
nStep := 8;
END_IF;
8:
fbSdoRead(bExecute := FALSE);
IF NOT fbSdoRead.bBusy THEN
```

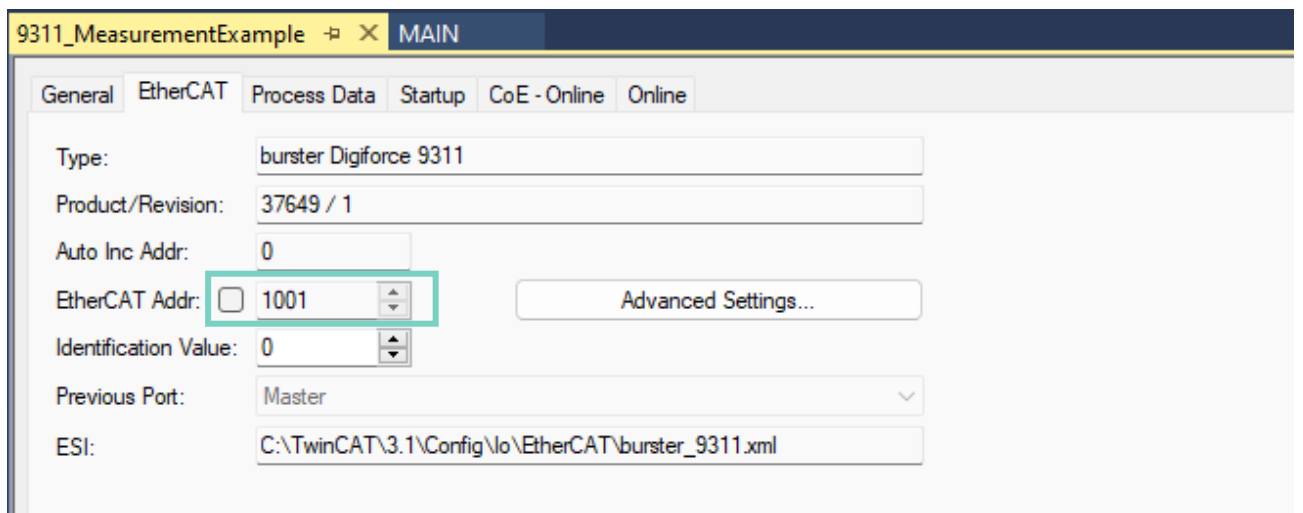
```


fbSdoRead(sNetId:= sNetId,
          nSlaveAddr :=nSlaveAddr,
          nIndex:=16#2039,
          nSubIndex := 14,
          pDstBuf:= ADR(yMaxRead),
          cbBufLen:=SIZEOF(yMaxRead),
          bExecute:=bExecute
);
bError:=fbSdoRead.bError;
nErrId:=fbSdoRead.nErrId;
nStep := 9;
END_IF;
9:
fbSdoRead(bExecute := FALSE);
RETURN;
END_CASE
    
```

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 EtherCAT slave address if you click the 9311 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 last **RETURN** code line:

```

128 8:
129 ● fbSdoRead(bExecute FALSE := FALSE);
130 ● IF NOT fbSdoRead.bBusy TRUE THEN
131 ● fbSdoRead(sNetId '192.168.19 ▶ := sNetId '192.168.19 ▶,
132     nSlaveAddr 1001 := nSlaveAddr 1001,
133     nIndex 8249 := 16#2039,
134     nSubIndex 14 := 14,
135     pDstBuf 18446603358358153180 := ADR(yMaxRead 8.89),
136     cbBufLen 4 := SIZEOF(yMaxRead 8.89),
137     bExecute FALSE := bExecute TRUE
138 );
139 ● bError FALSE := fbSdoRead.bError FALSE;
140 ● nErrId 0 := fbSdoRead.nErrId 0;
141 ● nStep 9 := 9;
142 END_IF;
143 9:
144 ● fbSdoRead(bExecute FALSE := FALSE);
145 ● RETURN;
146 ● END_CASE RETURN
    
```

- Start the program execution with the **F5** key or via **PLC** → **Start** and check if the written und read values are identical:

WriteReadWindowLimits [Online] × MAIN [Online]			
_9311_MeasurementExample.RunMeasurement.WriteReadWindowLimits			
Expression	Type	Value	Comment
xMin	REAL	5.53	Xmin coordinate of window 1
xMax	REAL	6.24	Xmax coordinate of window 1
yMin	REAL	7.35	Ymin coordinate of window 1
yMax	REAL	8.89	Ymax coordinate of window 1
xMinRead	REAL	5.53	Xmin coordinate has been read
xMaxRead	REAL	6.24	Xmax coordinate has been read
yMinRead	REAL	7.35	Ymin coordinate has been read
yMaxRead	REAL	8.89	Ymax coordinate has been read

5.2 Write and Read of 'string' data types

Example 3: Write a serial number into order sheet, read it back and compare (Index 0x2030, Subindex 65):

- Create a new POU as described above and name it **WriteOrderSheetSN1**:

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

```

PROGRAM WriteOrderSheetSN1
VAR
orderSheetSN1  : STRING := 'B1234567890';
fbSdoWrite    : FB_EcCoESdoWrite;
fbSdoRead     : FB_EcCoESdoRead;
sNetId        : T_AmsNetId := '192.168.19.1.4.1';
nSlaveAddr    : UINT := 1001;
bExecute      : BOOL := TRUE;
bError        : BOOL;
nErrId        : UDINT;
END_VAR

fbSdoWrite(
  sNetId      := sNetId,           // see note 1 in the previous sections
  nSlaveAddr  := nSlaveAddr,      // see note 2 in the previous sections
  nIndex      := 16#2030,
  nSubIndex   := 65,
  pSrcBuf     := ADDR(orderSheetSN1),
  cbBufLen    := INT_TO_UDINT(LEN(orderSheetSN1)),
  bExecute    := bExecute
);
IF NOT fbSdoWrite.bBusy THEN
  bExecute := FALSE;
IF NOT fbSdoWrite.bError THEN // write successful
  bError := FALSE;
  nErrId := 0;
ELSE // write failed
  bError := fbSdoWrite.bError;
  nErrId := fbSdoWrite.nErrId;
END_IF
fbSdoWrite(bExecute := FALSE);
END_IF

fbSdoWrite(bExecute := FALSE);
orderSheetSN1 := ";

```

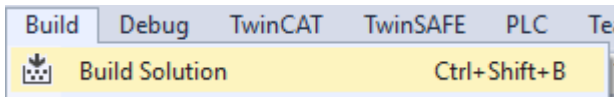
```
fbSdoRead(sNetId:= sNetId,
          nSlaveAddr :=nSlaveAddr,
          nIndex:=16#2030,
          nSubIndex := 65,
          pDstBuf:= ADR(orderSheetSN1),
          cbBufLen:=SIZEOF(orderSheetSN1),
          bExecute:=bExecute
);

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

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

```
1
2
3 WriteOrderSheetSN1 ();
4
-
```

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




- Log in **PLC** → **Login**, set a break point in front of *fbSdoRead* line and click **PLC** → **Start (F5)** to run the program

```
24 orderSheetSN1 := '';
25
26 fbSdoRead(sNetId := sNetId '192.168.19',
27          nSlaveAddr :=nSlaveAddr 1001,
28          nIndex :=16#2030,
29          nSubIndex := 65,
30          pDstBuf := ADR(orderSheetSN1),
31          cbBufLen :=SIZEOF (orderSheetSN1),
32          bExecute FALSE :=bExecute TRUE
33 );
34 bError FALSE :=fbSdoRead.bError FALSE ;
35 nErrId 0 :=fbSdoRead.nErrId 0 ;RETURN
```

- Make sure that the string *orderSheetSN1* is empty and press the key **F10** to execute *fbSdoRead* function:

```
26 fbSdoRead(sNetId '192.168.19' := sNetId '192.168.19',
27          nSlaveAddr 1001 :=nSlaveAddr 1001,
28          nIndex 8240 :=16#2030,
29          nSubIndex 65 := 65,
30          pDstBuf 18446603358358136152 := ADR(orderSheetSN1 'B123456789'),
31          cbBufLen 81 :=SIZEOF (orderSheetSN1 'B123456789'),
32          bExecute TRUE :=bExecute TRUE
33 );
```

- Check if the written serial number has been read correctly (variable *orderSheetSN1*)

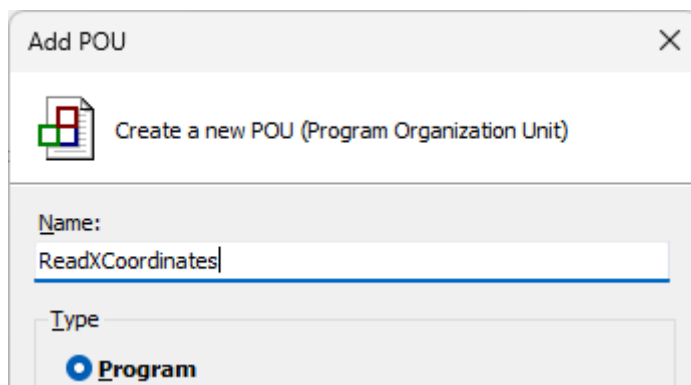
Expression	Type	Value	Comment
 orderSheetSN1	STRING	'B1234567890'	

Note: you can also control the serial number, you wrote into the order sheet in device menu **Basic setup** → **Order Sheet** or via our PC-Software DigiControl

5.3 Read of X-Coordinates

Example 3: Read-out X-coordinates of current measurement curve (Index 0x2083, Subindex 10, 11)

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



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

```
PROGRAM ReadXCoordinates
VAR
    fbSdoRead      : FB_EcCoESdoRead;
    sNetId         : T_AmsNetId := '192.168.19.1.4.1'; // see note 1 in previous sections
    nSlaveAddr     : UINT := 1001; // see note 2 in previous sections
    bExecute       : BOOL := TRUE;
    bError         : BOOL;
    nErrId         : UDINT;
    nStep          : INT := 0;
    indexLastCoord : UDINT := 0; // Index of the last coordinate
    xCoordinates   : ARRAY [0..5000] OF REAL;
END_VAR

CASE nStep OF
0:
    fbSdoRead(sNetId:= sNetId,
              nSlaveAddr :=nSlaveAddr,
              nIndex:=16#2083,
              nSubIndex := 10,
              pDstBuf:= ADR(indexLastCoord),
              cbBufLen:=SIZEOF(indexLastCoord),
```

```

        bExecute:=bExecute
    );
    bError:=fbSdoRead.bError;
    nErrId:=fbSdoRead.nErrId;
    nStep := 1;

    1:
    fbSdoRead(bExecute := FALSE);
    IF NOT fbSdoRead.bBusy AND indexLastCoord > 0 THEN // if 0, there is no curve
    fbSdoRead(sNetId:= sNetId,
        nSlaveAddr :=nSlaveAddr,
        nIndex:=16#2083,
        nSubIndex := 11,
        pDstBuf:= ADR(xCoordinates),
        cbBufLen:=(indexLastCoord + 1) * 4,
        bExecute:=bExecute
    );
    bError:=fbSdoRead.bError;
    nErrId:=fbSdoRead.nErrId;
    nStep := 2;
    END_IF;

    2:
    fbSdoRead(bExecute := FALSE);
    RETURN;
    END_CASE

```

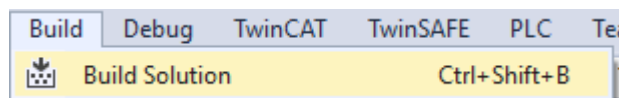
- Insert a call for the POU in the **MAIN** block:

```

1
2
3 ReadXCoordinates();
4
5

```

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



- Set a break point on the **RETURN** line, and start the program execution by pressing **F5**:

```

32 2:
33 fbSdoRead(bExecute FALSE := FALSE);
34 RETURN;
35 END_CASE RETURN

```


- Check the read index of the last curve coordinate and the curve x-values:

_9311_MeasurementExample.RunMeasurement.ReadXCoordinates			
Expression	Type	Value	Comment
xCoordinates	ARRAY [D..5000] O...		
xCoordinates[0]	REAL	0	
xCoordinates[1]	REAL	0.01	
xCoordinates[2]	REAL	0.02	
xCoordinates[3]	REAL	0.03	
xCoordinates[4]	REAL	0.04	
xCoordinates[5]	REAL	0.05	
xCoordinates[6]	REAL	0.06	

```

17 fbSdoRead(bExecute FALSE := FALSE);
18 IF NOT fbSdoRead.bBusy TRUE AND indexLastCoord 220 > 0 THEN // if 0, there is no curve
19 fbSdoRead(sNetId '192.168.19' := sNetId '192.168.19',
20 nSlaveAddr 1001 := nSlaveAddr 1001,
21 nIndex 8323 := 16#2083,
22 nSubIndex 11 := 11,
23 pDstBuf 18446803358767498552 := ADR(xCoordinates),
24 cbBufLen 884 := (indexLastCoord 220 + 1) * 4,
25 bExecute FALSE := bExecute TRUE
26 );
27 bError FALSE := fbSdoRead.bError FALSE;
28 nErrId 0 := fbSdoRead.nErrId 0;
29
30 nStep 2 := 2;
31 END_IF;
32 2;
33 fbSdoRead(bExecute FALSE := FALSE);
34 RETURN;
    
```