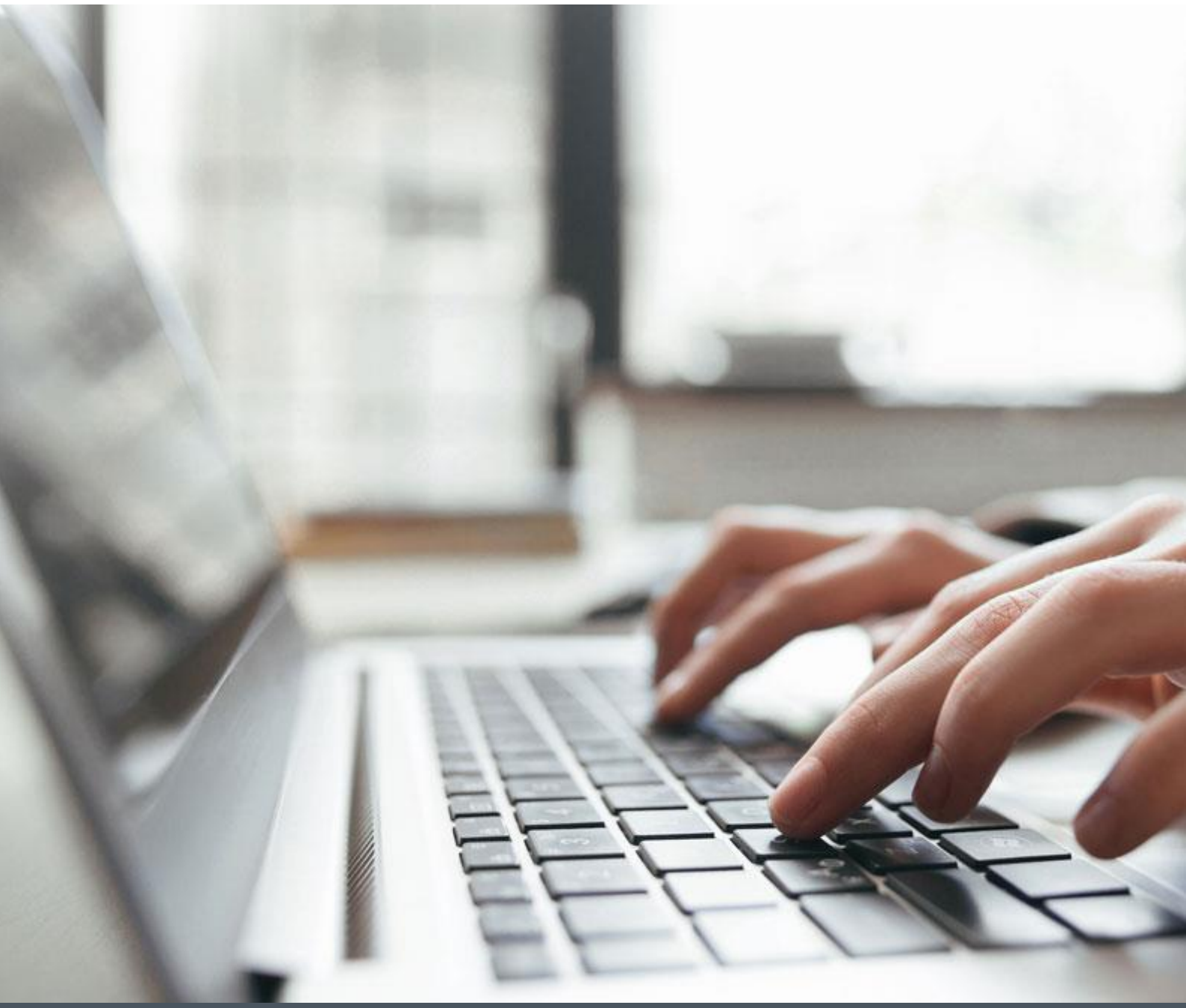


Application Note

Starting up a Nanotec Controller/Drive with TwinCAT3 via EtherCAT

Version 1.0

www.nanotec.com

Contents

1 Prerequisites

The slave drive (Nanotec Controller) must be configured beforehand. Proper operation of the motor and slave drive must be ensured before the example can be used. Make sure that the controller/drive operation is not hindered, e.g. by a stand-alone program running on the slave.

2 Hardware

N5-2-1, Firmware Version FIR-v1650-B527540

3 Software

TwinCAT System Manager v3.1 (Build 4202)

4 Preparing the ESI-File

Before starting to work in TwinCAT, you need to include the ESI-file of the drive.
Therefor copy the ESI-file into the correct TwinCAT directory.
By default, for TwinCAT 3 this should be the following path:
C:\TwinCAT\3.1\Config\Io\EtherCAT
You must decide whether you want to include the driver as a box or an axis.
To do so, open the ESI-File using a text-editor:

```
<?xml version="1.0" encoding="UTF-8"?><EtherCATInfo xmlns:xsi="http://www.w3
        <Vendor>
                <Id>#x0000026C</Id>
                <Name>Nanotec Electronic GmbH &amp;amp; Co. KG</Name>
                <ImageData16x14>424D6E6000000000000076000000280000001000000
        </Vendor>
        <Descriptions>
                <Groups>
                        <Group SortOrder="520">
                                <Type>Drive</Type>
                                <Name LcId="1033">Drives</Name>
                                <Name LcId="1031">Antriebe</Name>
                                <ImageData16x14>424D6E600000000000000760000002
                        </Group>
                </Groups>
                <Devices>
                        <Device Physics="YY">
                                <Type ProductCode="#x00000009" RevisionNo="#
                                <Name LcId="1033">N5 EtherCAT Drive (CoE)</N
                                <Name LcId="1031">N5 EtherCAT Drive (CoE)</N
                                <Info>
                                        <StateMachine>
                                                <Timeout>
                                                        <PreopTimeout>3000</
                                                        <SafeopOpTimeout>100
                                                        <BackToInitTimeout>5
                                                        <BackToSafeopTimeout
                                                </Timeout>
                                        </StateMachine>
                                        <Mailbox>
                                                <Timeout>
                                                        <RequestTimeout>100<
                                                        <ResponseTimeout>500
                                                </Timeout>
                                        </Mailbox>
                                </Info>
                                <GroupType>Drive</GroupType>
                                <Profile>
                                        <ProfileNo>402</ProfileNo>
                                        <AddInfo>x</AddInfo>
```

Here you can find a line called **<AddInfo>x</AddInfo>**. To use the drive as an axis, set the value to “0”, to use it as a box set it to “2”. Save and Close the ESI-file.

5 Drive as Box

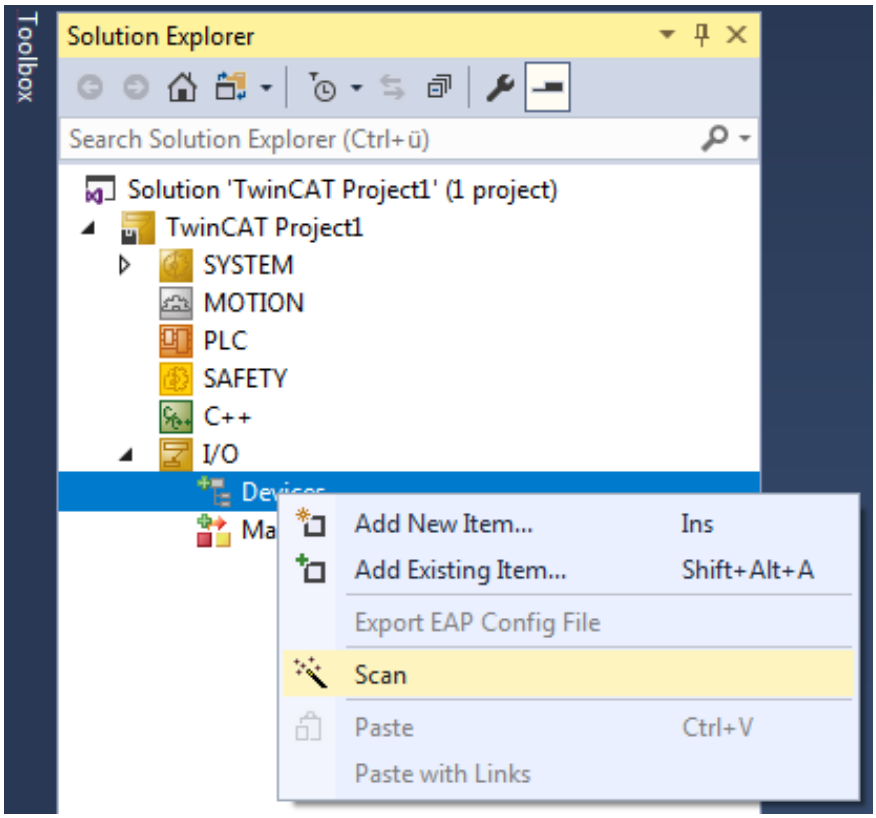
Please don´t forget to set the **AddInfo** to “2”, otherwise the steps in this chapter won´t work as shown. If you want to use the drive as axis, go directly to **6.Drive as Axis**.

5.1 Creating a new project and adding the drive

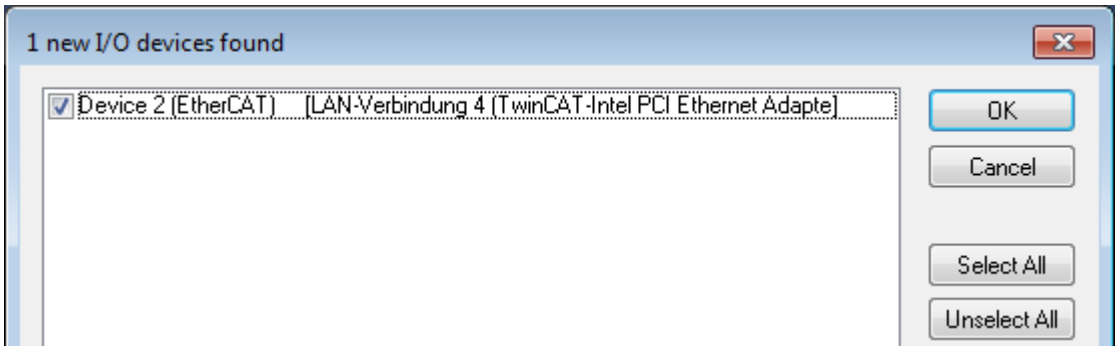
1 Connect the drive via EtherCAT to a network-card supporting EtherCAT-communication or to your PLC and to a power-supply.

2 Create a new **TwinCAT XAE Projekt** in TwinCAT 3.

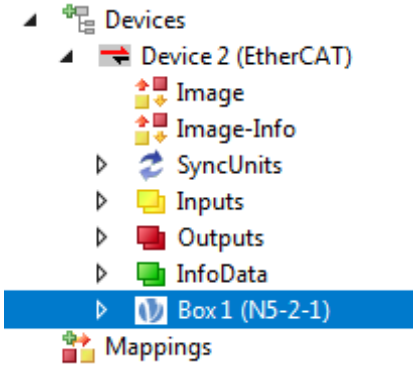
3 Right-click on **Devices** in your **project-tree** and select **Scan**.



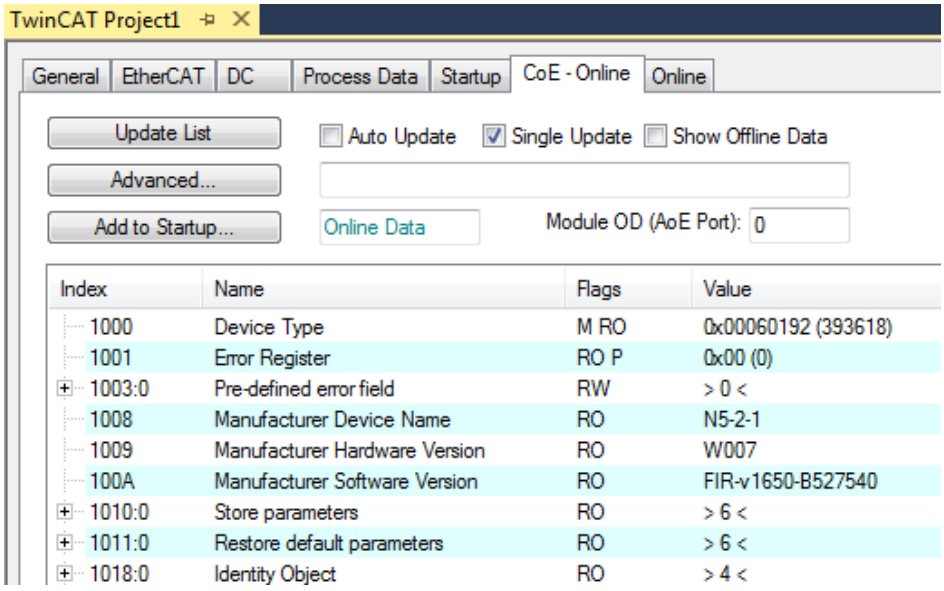
4 Select your network-card or PLC here and click **OK**.



5 After searching for boxes, the drive should show up as **Box 1**.

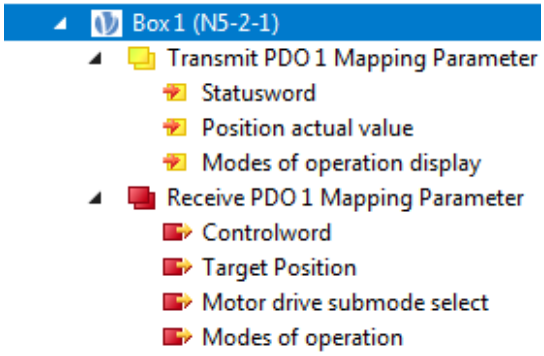


6 By clicking on **Box 1** and selecting the Tab **CoE-Online** you can see the object-dictionary of the drive and could add startup-values if needed.



7 The whole functionality of the controller would be available by changing the **values** in that tab.

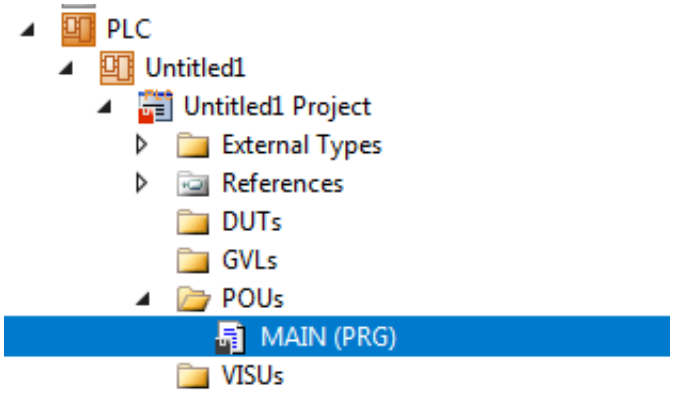
You can also see the default PDO-Mappings in the **project-tree**, which we are going to connect to a program in the next step.



5.2 Creating a Main-PRG and connecting the variables

1 To create a new **PRG** right-click **on PLC -> Add new element** and select a **Standard PCL Project**.

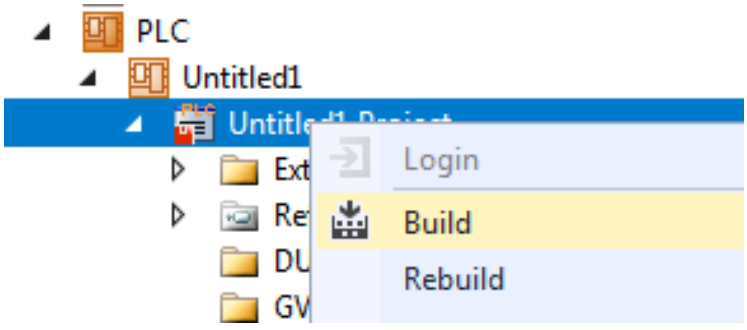
2 Open the **Main (PRG)** in the folder **POUs**.



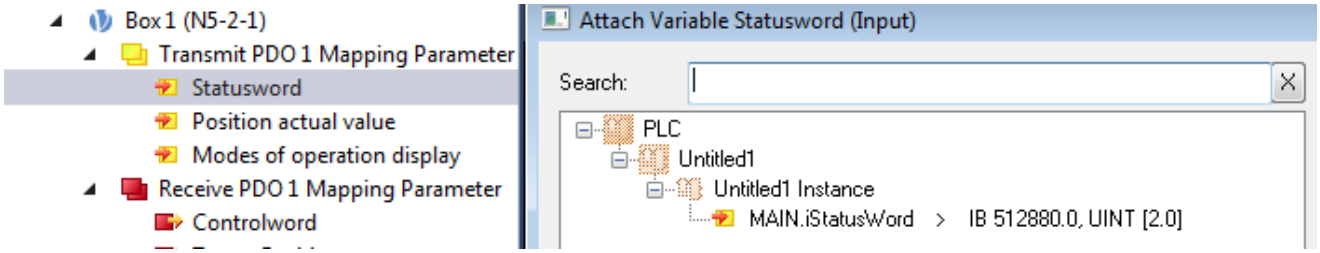
3 Here you can define some variables for the program and link them to the PDO-In- and Outputs of the Box.

```
PROGRAM MAIN
VAR
    //Outputs to the Box
        iControlWord AT%Q*:UINT;
        iModesOfOperation AT%Q*:SINT;
        iTargetPosition AT%Q*:DINT;
        iMotorDriveSubmodeSelect AT%Q*:UDINT;
    //Inputs from the Box
        iStatusWord AT%I*:UINT;
        iModesOfOperationDisplay AT%I*:SINT;
        iActualPosition AT%I*:DINT;
    //Program-variables
        bStart : BOOL;      //Starts to move the motor between the positions 0 and 10000
        iStep : USINT :=0 ; //"State-Machine" for the program

END_VAR
```

4 Right-click on your **project** and select **Build**.



5 To link our variables to the box, simply double-click on the **PDO-In/Outputs** (for example the **Statusword**) of your box and select the corresponding variable:



6 Do this for all **PDOs** you want to use.

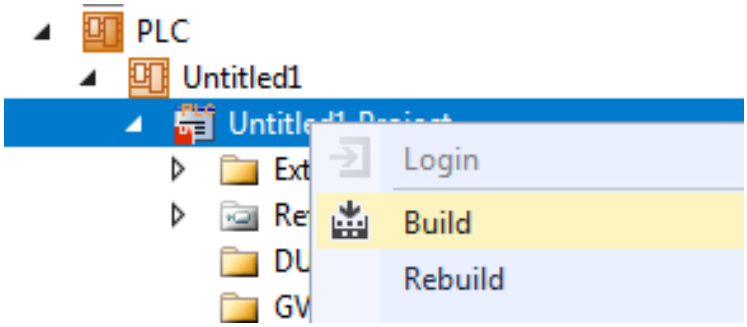
5.3 Simple Positioning-Example

Now you are ready to use the variables in a program.

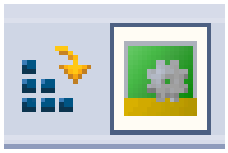
In the following example the motor will permanently move between two positions (in this case the positions are “0” and “10000”).

```
IF(bStart AND (iStep = 0)) THEN
iModesOfOperation := 1;
iMotorDriveSubmodeSelect :=1;
iControlWord :=6;
iStep := 1;
END_IF

IF(NOT bStart AND (iStep > 0)) THEN
iControlWord :=0;
iStep := 0;
END_IF

IF((iStep = 1)AND((iStatusWord AND 16#EF)=16#21)) THEN
iControlWord :=7;
iStep := 2;
END_IF

IF((iStep = 2)AND((iStatusWord AND 16#EF)=16#23)) THEN
iControlWord :=15;
iStep := 3;
END_IF

IF((iStep = 3)AND((iStatusWord AND 16#EF)=16#27)) THEN
iTargetPosition := 0;
iControlWord :=31;
iStep := 4;
END_IF

IF((iStep = 4)AND((iStatusWord AND 16#1000)=16#1000)) THEN
iControlWord :=15;
iStep := 5;
END_IF

IF((iStep = 5)AND((iStatusWord AND 16#400)=16#400)) THEN
iTargetPosition := 10000;
iControlWord :=31;
iStep := 6;
END_IF

IF((iStep = 6)AND((iStatusWord AND 16#1000)=16#1000)) THEN
iControlWord :=15;
iStep := 7;
END_IF

IF((iStep = 7)AND((iStatusWord AND 16#400)=16#400)) THEN
iTargetPosition := 0;
iControlWord :=31;
iStep := 4;
END_IF
```

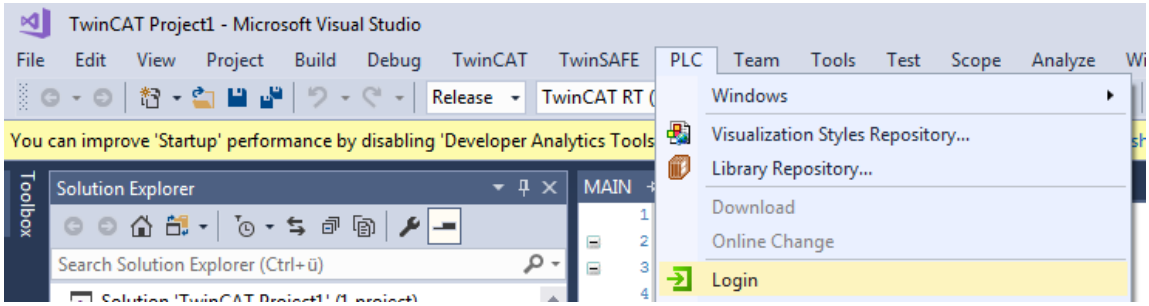
1 **Build** the program.



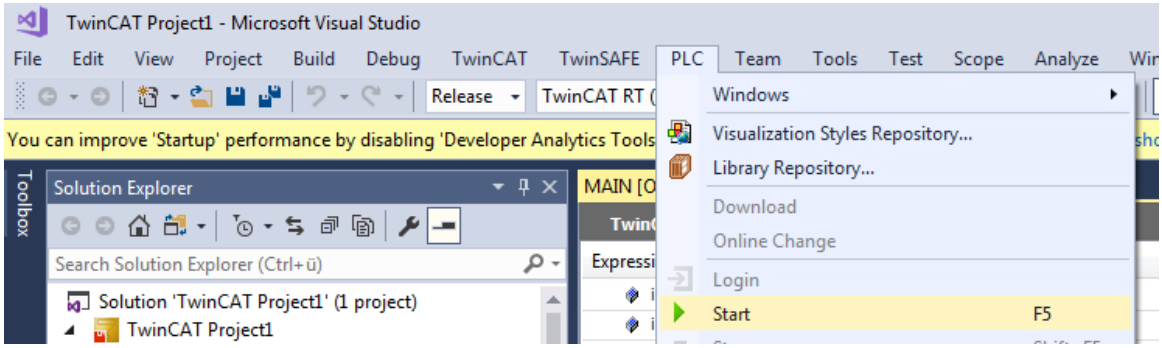
2 To start the program, **activate the configuration** and restart in **Run-Mode.**



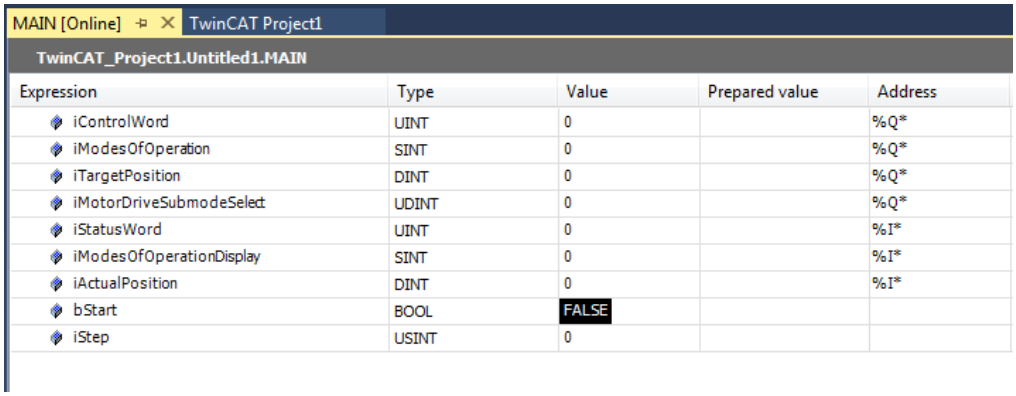
3 **Login**.



4 **Start.**

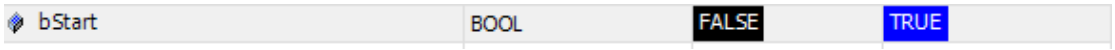


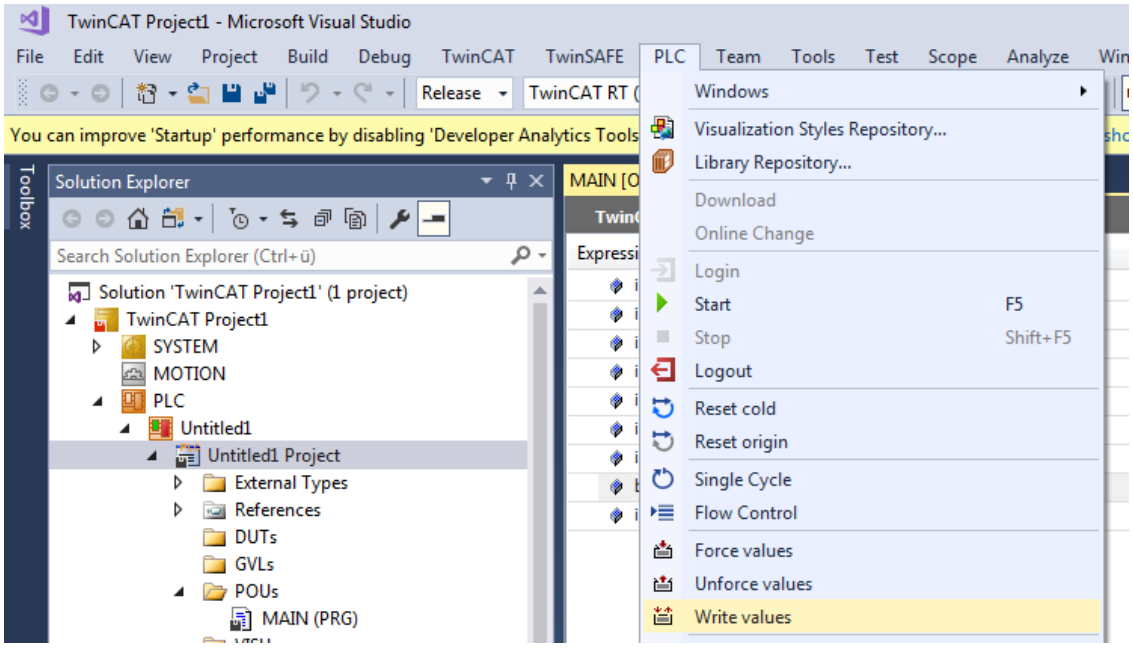
5 You can see the actual **value** of the variables.



Expression	Type	Value	Prepared value	Address
iControlWord	UINT	0		%Q*
iModesOfOperation	SINT	0		%Q*
iTargetPosition	DINT	0		%Q*
iMotorDriveSubmodeSelect	UDINT	0		%Q*
iStatusWord	UINT	0		%I*
iModesOfOperationDisplay	SINT	0		%I*
iActualPosition	DINT	0		%I*
bStart	BOOL	FALSE		
iStep	USINT	0		

6 To start the motor, you need to set the variable *bStart* to “true”. Therefor click on **prepared value** behind the actual value of the variable and click on **Write Values** afterwards:





7 While *bStart* is set to “true”, the motor will move between the two positions. To stop it you need to set *bStart* to “false” again.

6 Drive as Axis

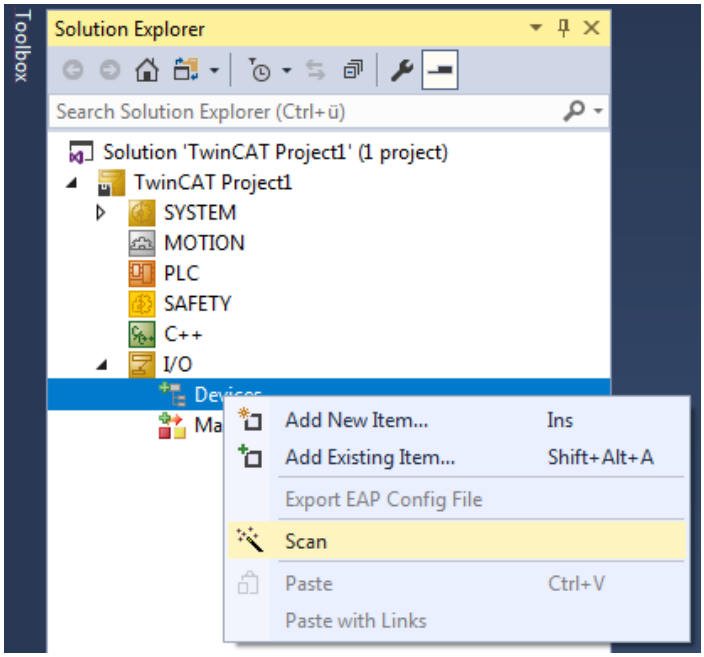
Please don´t forget to set the **AddInfo** to “0”, otherwise the steps in this chapter won´t work as shown.

6.1 Creating a new project and adding the axis

1 Connect the drive via EtherCAT to a network-card supporting EtherCAT-communication or to your PLC and to a power-supply.

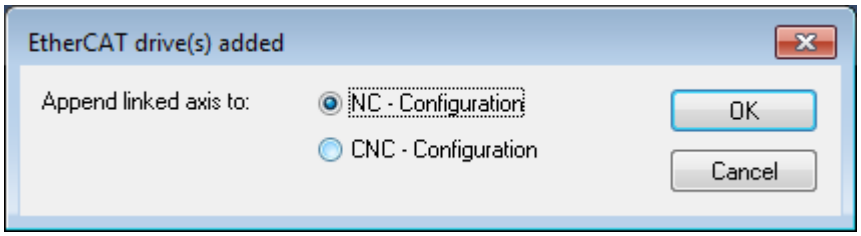
2 Create a new **TwinCAT XAE Projekt** in **TwinCAT 3**.

3 Right-click on **Devices** in your **project-tree** and select **Scan**.

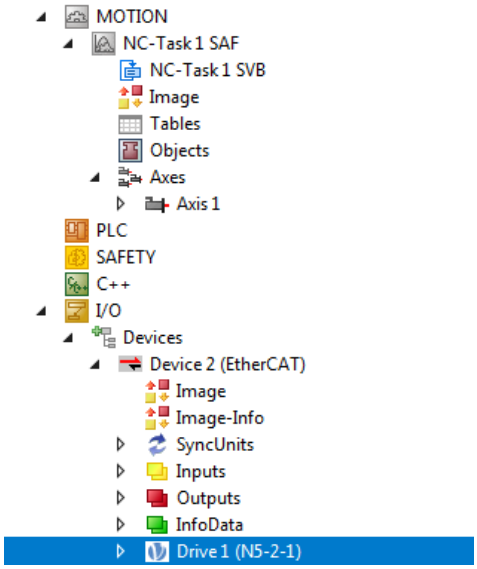


4 Select your network-card or PLC here and click **OK**.

5 **TwinCAT** will ask you whether to add the drive as **NC** or **CNC** axis, the next steps will use the **NC-configuration**:

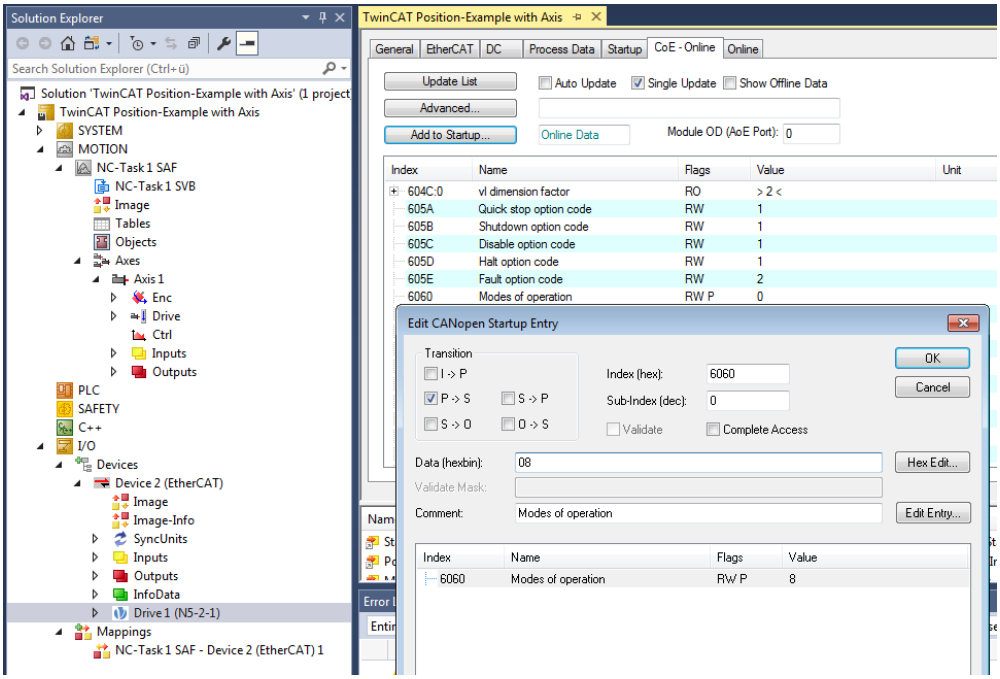


6 Now the drive should be shown as **Drive** at devices and a new **NC-Task** with one axis is generated:

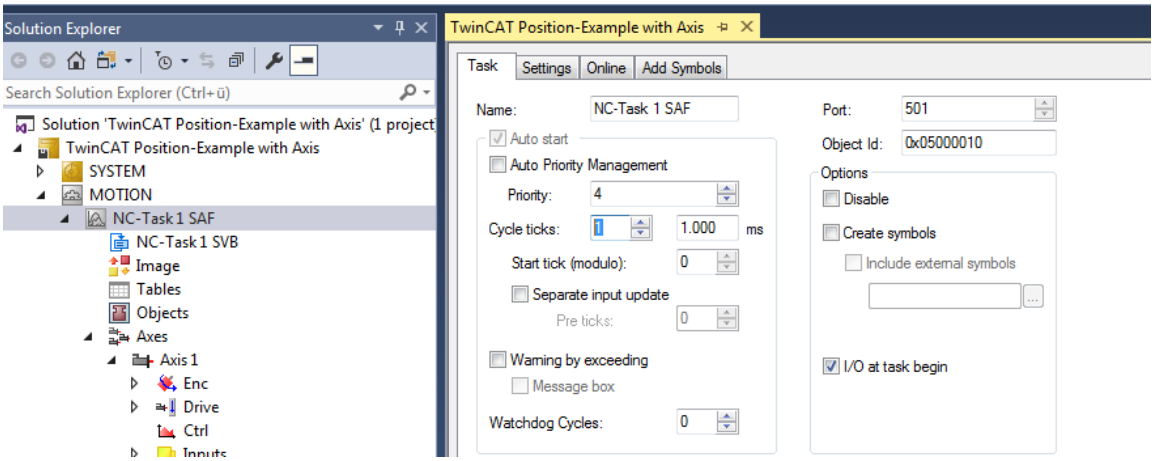


7 In the next step make some general settings:

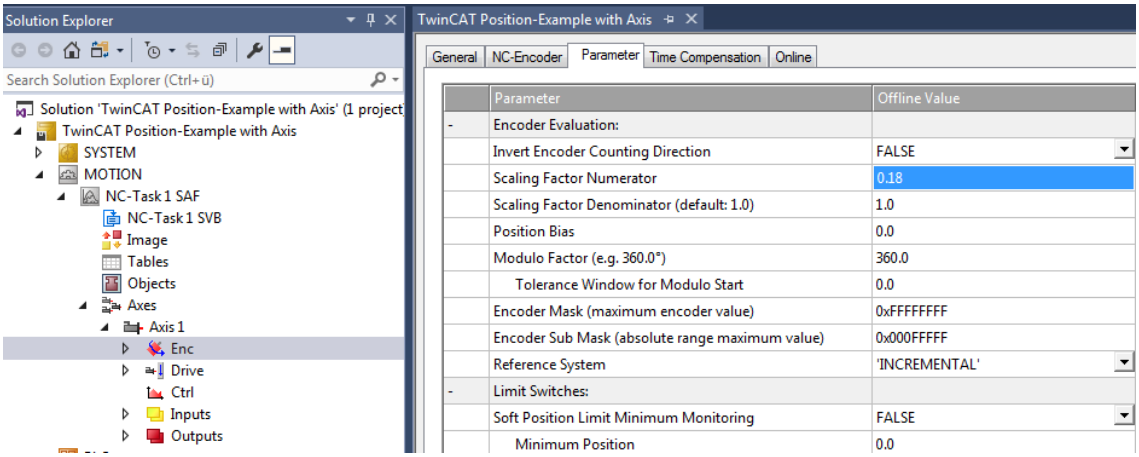
- Set the drive to start up in *operation mode* “8” (Cyclic Synchronous Position Mode). Click on **Drive 1** and select the Tab **CoE-Online**. Scroll down and mark the **Index 6060**, then click on **Add to Startup** and set the value to “8”.



- Set the **NC-Task Cycle ticks** to “1”.

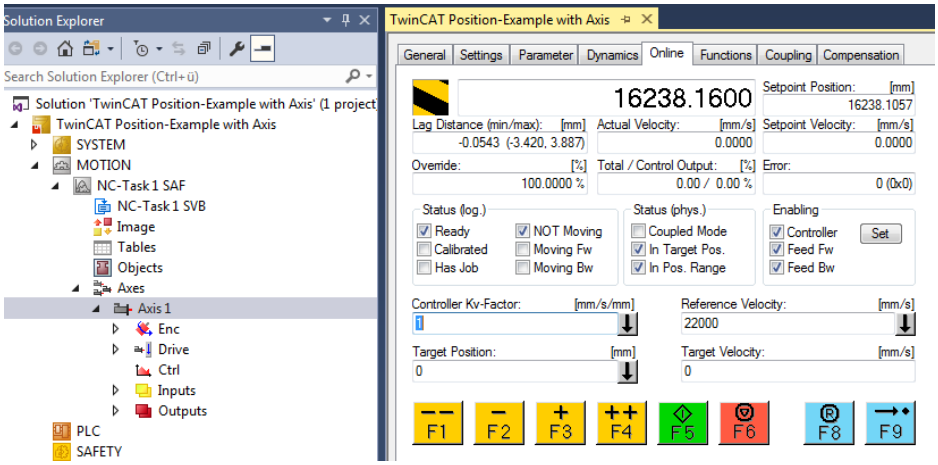


- Change the **Encoder Scaling Factor** to “0.18”.

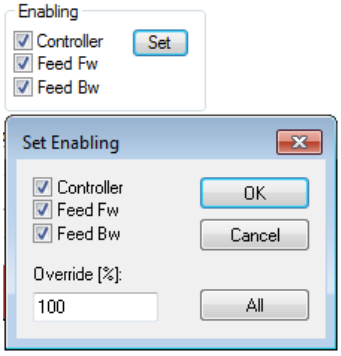


8 **Activate** this configuration.

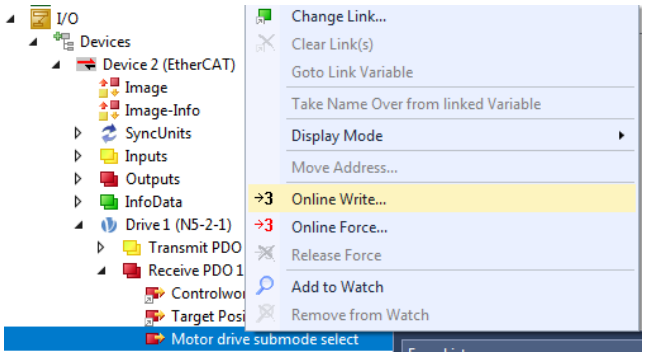
9 We can use the **Online-Control-Terminal** of the axis now.



10 Click on **Set** and activate the 3 **checkboxes** and set the **Override** to “100%” to enable the motor.



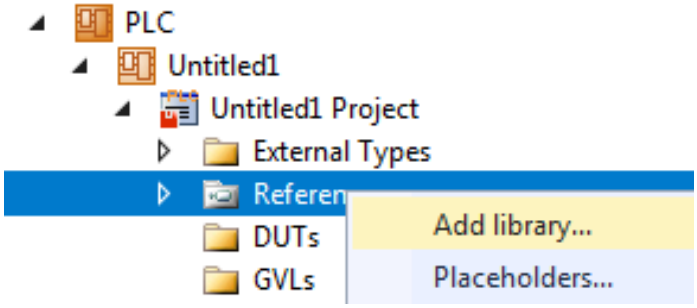
11 If your motor has an encoder, activating Closed Loop is recommended here. This can be done by selecting the **drive -> RPDO1 -> Motor drive submode select**. With a right-click you can select **Online-Write** and set the value to “1” (dec) to activate the Closed Loop for a stepper and “65” (dec) for a BLDC.



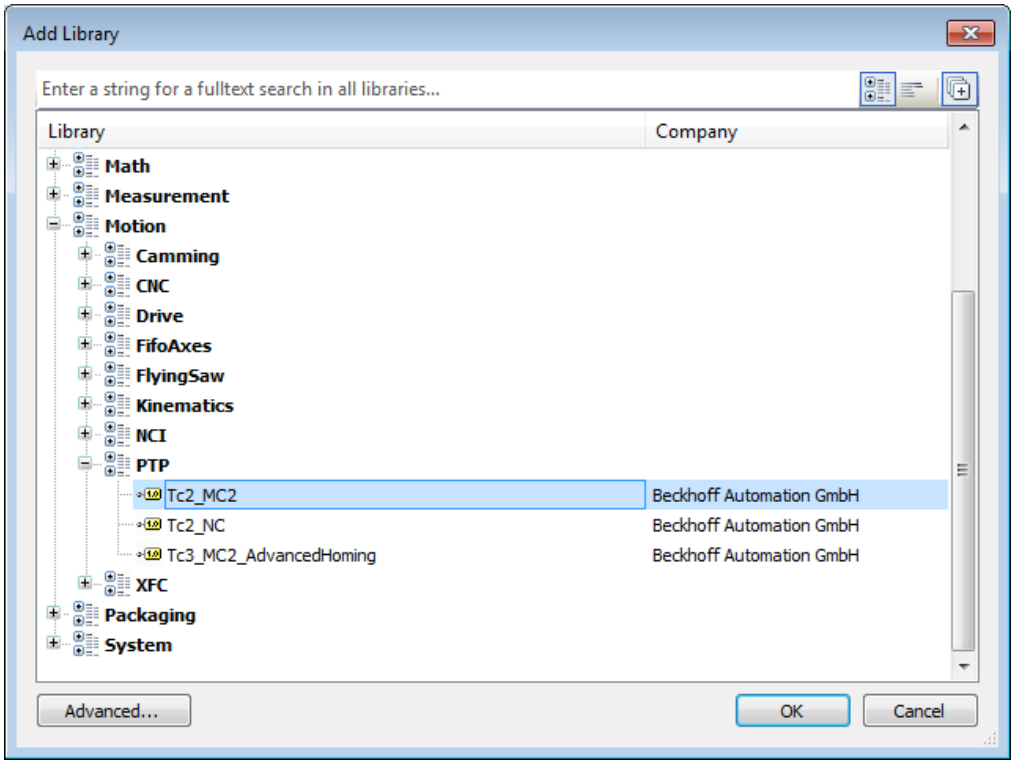
6.2 Creating a Main-PRG and connecting the variables

1 To use the axis in a program, you first need to build a **PLC-project** and add the library **Tc_MC2**. Therefor right-click on **PLC** and **Add new item**. Select a **Standard PLC Project**.

2 In this project, right-click on **References** and **Add library**.



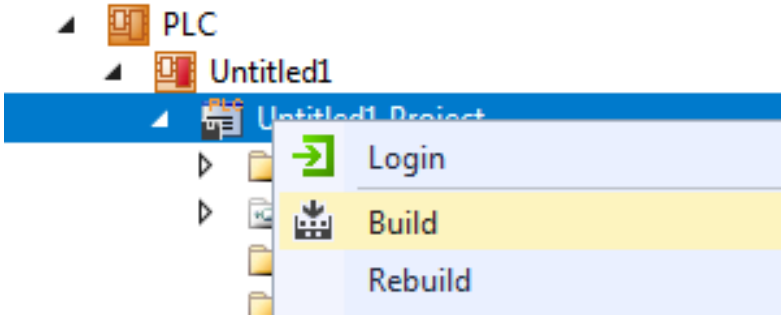
3 Select the library **Tc2_MC**.



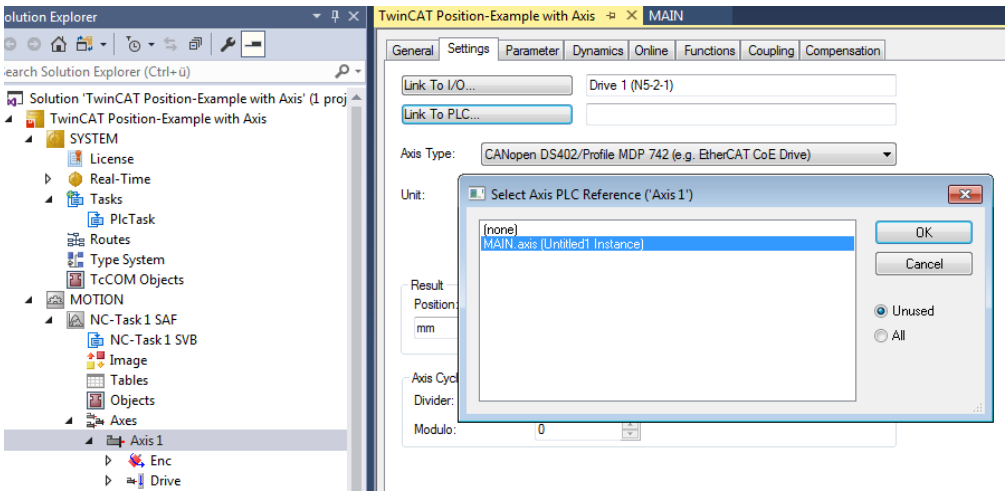
4 Open the **MAIN(PRG)** in the **POUs**-folder and create a variable of the type **AXIS_REF** as well as some other variables you need to control the MC-functions.

```
PROGRAM MAIN
VAR
    axis: AXIS_REF;
    fbAxisPower: MC_Power;
    fbAxisMoveAbsolute: MC_MoveAbsolute;
    bEnable: BOOL := FALSE;
    fOverride: LREAL := 100;
    bMove: BOOL := FALSE;
    fTargetPosition: LREAL := 90;
    fTargetVelocity: LREAL := 5;
    iMotorDriveSubmodeSelect AT%Q*:UDINT;

END_VAR
```

5 Right-click on your **PLC-project** and **build** it:



7 After that, you need to link our axis to the variable with the name *axis*. Therefor open the **settings** of your axis, click on **Link to PLC** and select the variable from your **MAIN**.



8 The variable *iMotorDriveSubmodeSelect* is used to control if the drive is in Open or Closed Loop. You must link this to **RPDO1**. Double-click on the **RPDO1-output** with the name **Motor drive submode select** and select this variable from the **MAIN**.

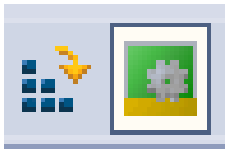
6.3 Simple Positioning-Example

1 The **TwinCat-MC-functions** can be used now for a simple example where the motor should be powered on and perform some simple absolute positioning tasks.

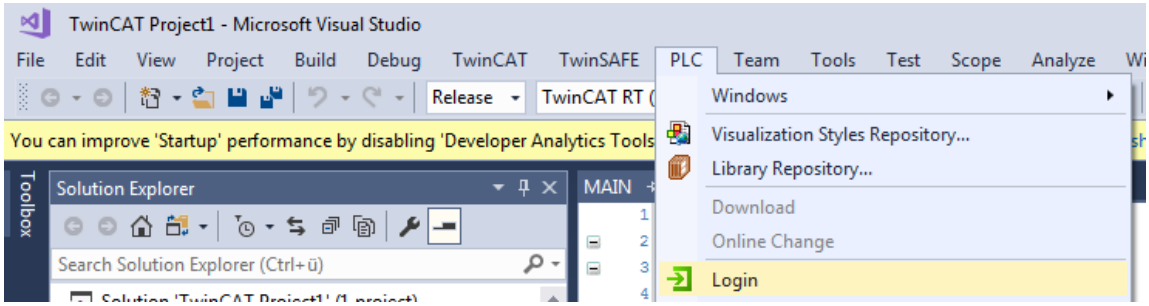
2 Therefor we are using the **MC-functions MC_Power** and **MC_MoveAbsolute**.

```
axis.ReadStatus();

fbAxisPower(
    Axis:= axis,
    Enable:= bEnable,
    Enable_Positive:= bEnable,
    Enable_Negative:= bEnable,
    Override:= fOverride,
    BufferMode:= ,
    Options:= ,
    Status=> ,
    Busy=> ,
    Active=> ,
    Error=> ,
    ErrorID=>
);

fbAxisMoveAbsolute(
    Axis:= axis,
    Execute:= bMove,
    Position:= fTargetPosition,
    Velocity:= fTargetVelocity,
    Acceleration:= ,
    Deceleration:= ,
    Jerk:= ,
    BufferMode:= ,
    Options:= ,
    Done=> ,
    Busy=> ,
    Active=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=>
);
```

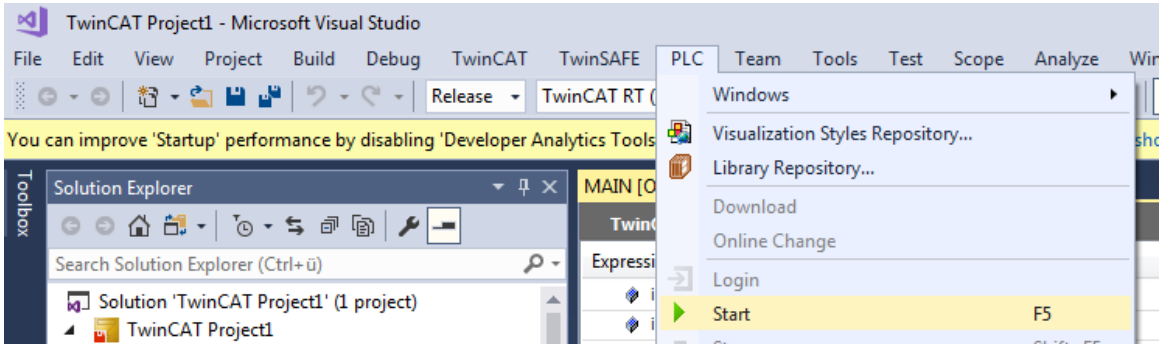
3 Build the **PLC-project** again, **activate** the configuration and **restart in Run-Mode**.



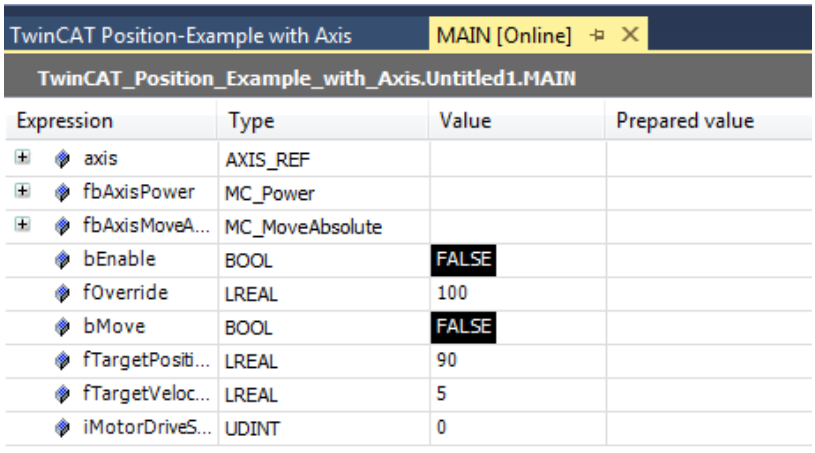
4 **Login**.



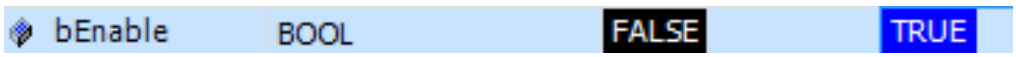
5 **Start**.

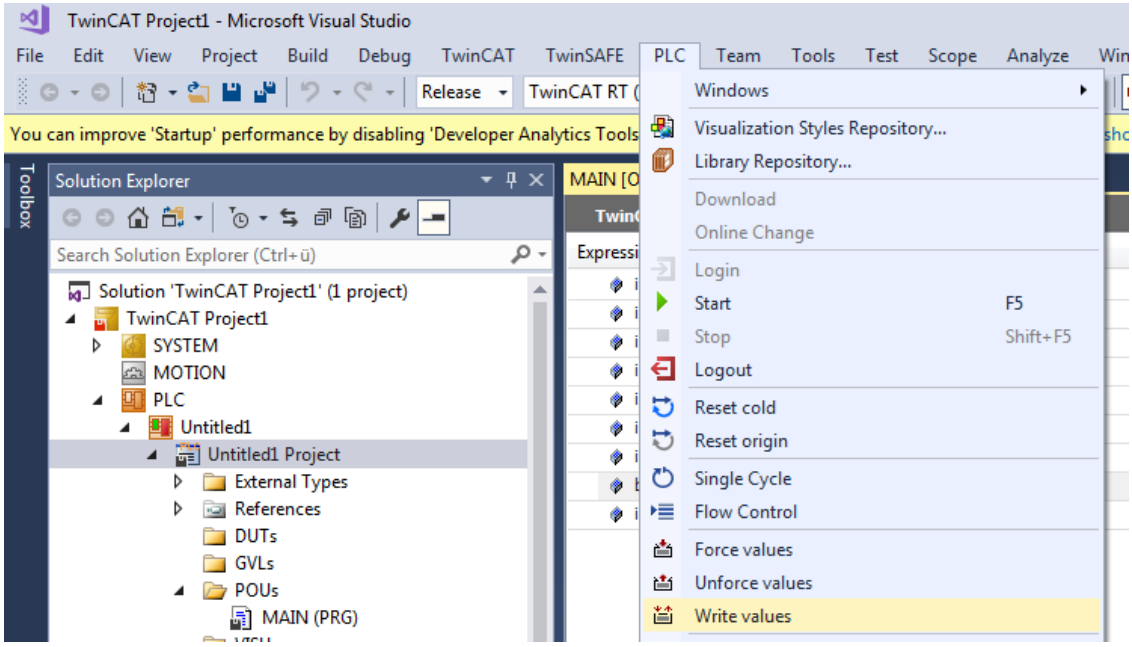


6 You can now see the actual **value** of the variables.



7 To power the motor, you need to set the variable *bEnable* to “true”. Therefor click on **prepared value** behind the actual **value** of the variable and click on **Write Values** afterwards.





8 While *bEnable* is set to “true”, the motor will be powered on.

9 To start an absolute position, set *bMove* to “true”. The motor will move with the speed of *fTargetVelcoity* to the position *fTargetPosition*.

10 By setting *iMotorDriveSubmodeSelect* to “1” (dec) for a stepper or to “65” (dez) for a BLDC, you can activate Closed Loop.

7 Liability

This Application Note is based on our experience with typical user requirements in a wide range of industrial applications. The information in this Application Note is provided without guarantee regarding correctness and completeness and is subject to change by Nanotec without notice.

It serves as general guidance and should not be construed as a commitment of Nanotec to guarantee its applicability to all customer applications without additional tests under the specific conditions and – if and when necessary – modifications by the customer.

The provided information does not replace datasheets and other product documents. For the latest version of our datasheets and documentations please visit our website at www.nanotec.com.

The responsibility for the applicability and use of the Application Note in a particular customer application lies solely within the authority of the customer. It is the customer's responsibility to evaluate, investigate and decide, whether the Application Note is valid and suitable for the respective customer application, or not.

Defects resulting from the improper handling of devices and modules are excluded from the warranty. Under no circumstances will Nanotec be liable for any direct, indirect, incidental or consequential damages arising in connection with the information provided.

In addition, the regulations regarding the liability from our Terms and Conditions of Sale and Delivery shall apply.

8 Copyrights and contact

Nanotec Electronic GmbH & Co. KG
Kapellenstraße 6
85622 Feldkirchen
Germany

Tel. +49 (0)89 900 686-0
Fax +49 (0)89 900 686-50
info@nanotec.de

www.nanotec.com

07.19 Germany