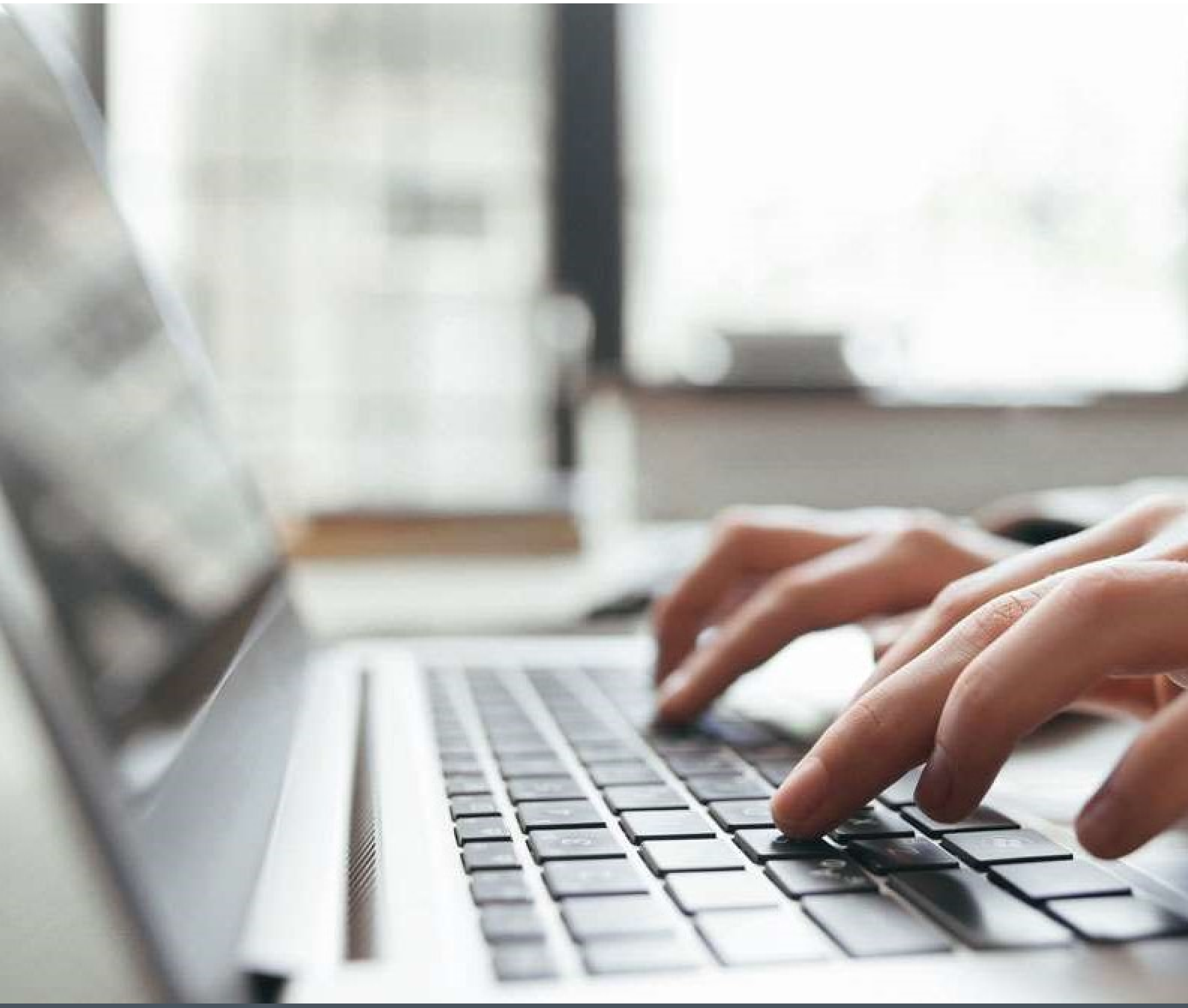


Application Note

Starting up a Nanotec Controller/Drive with OMRON NX1P2

Version 1.1.0

www.nanotec.com

Contents

1 **Prerequisites** 1

2 **Hardware** 1

3 **Software** 1

4 **Creating a new project and adding the drive** 1

5 **Example file for Velocity Mode with SDO commands** 6

5.1 Description of the example project 6

5.2 How to start the motor 7

6 **Example for positioning with Motion Control function blocks** 10

6.1 Setting up the axis 10

6.2 Programming an oscillating movement with MC Function blocks 13

7 **Liability** 16

8 **Copyrights and contact** 16

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.
To ensure correct operation with the OMRON PLC make sure that the interpolation time period in our controller is set to 2ms. This is needed because the minimum cycle time for the OMRON PLC is 2ms whereas our default value is 1ms.
0x60C2:01 = 2

2 Hardware

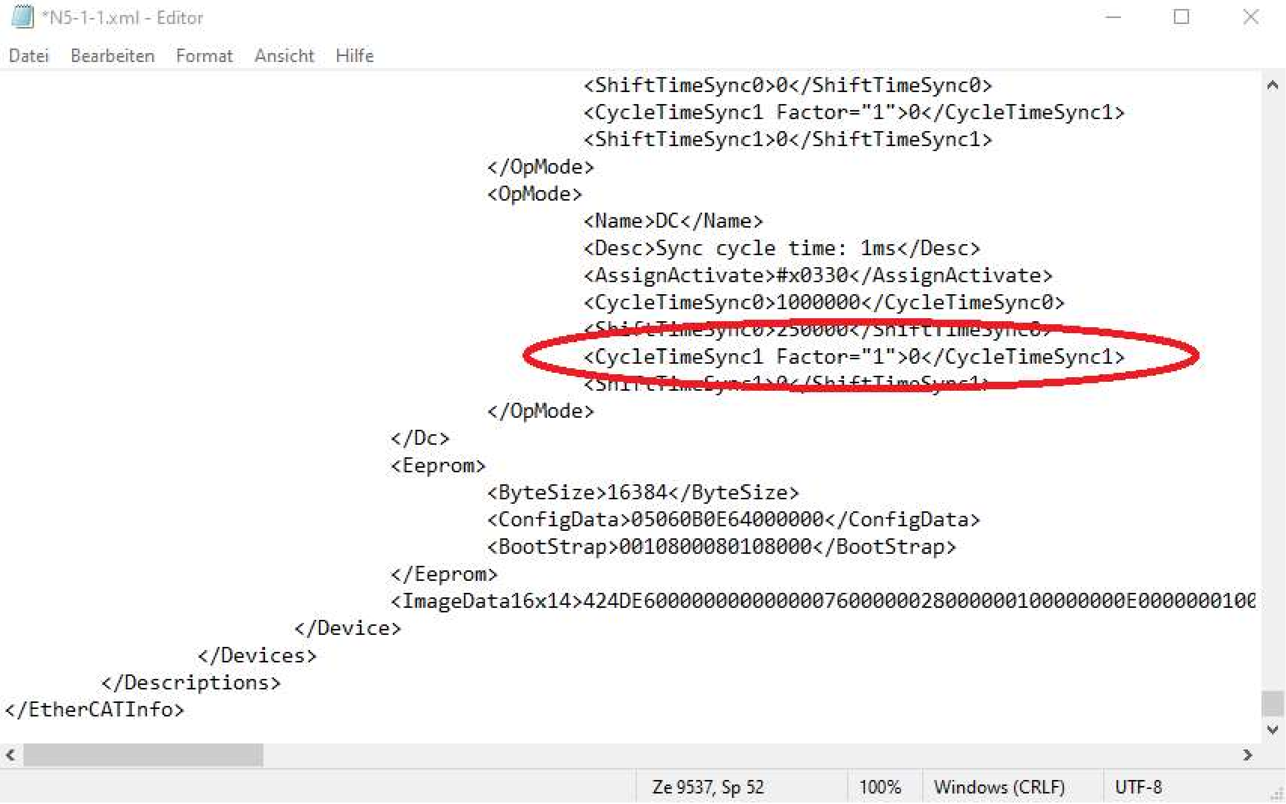
OMRON NX1P2-1040DT, Version 1.40
N5-1-1, Firmware version FIR-v1650-B527540

3 Software

Sysmac Studio, Version 1.30.1.0

4 Creating a new project and adding the drive

1 Prepare the ESI file, so that Distributed Clock Mode can be enabled later. Open the ESI file with an editor, scroll to the very bottom, and adjust the setting for DC. Remove the "-" sign for **CycleTimeSync1 Factor**. It should look like this:

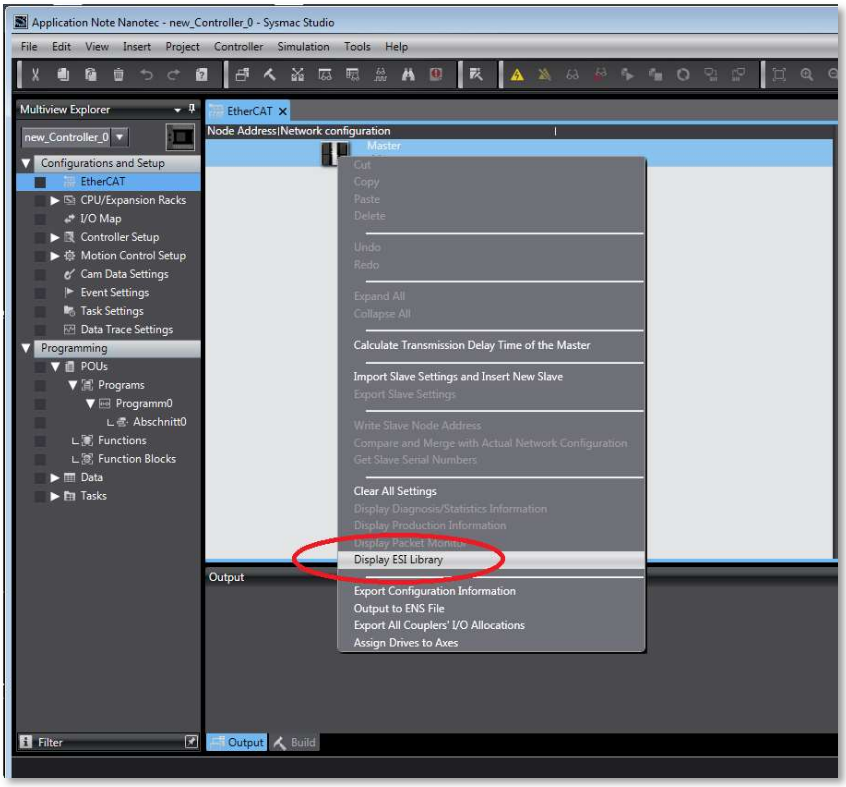


```
                                        <ShiftTimeSync0>0</ShiftTimeSync0>
                                        <CycleTimeSync1 Factor="1">0</CycleTimeSync1>
                                        <ShiftTimeSync1>0</ShiftTimeSync1>
                                </OpMode>
                                <OpMode>
                                        <Name>DC</Name>
                                        <Desc>Sync cycle time: 1ms</Desc>
                                        <AssignActivate>#x0330</AssignActivate>
                                        <CycleTimeSync0>1000000</CycleTimeSync0>
                                        <ShiftTimeSync0>250000</ShiftTimeSync0>
                                        <CycleTimeSync1 Factor="1">0</CycleTimeSync1>
                                        <ShiftTimeSync1>0</ShiftTimeSync1>
                                </OpMode>
                        </Dc>
                        <Eeprom>
                                <ByteSize>16384</ByteSize>
                                <ConfigData>05060B0E64000000</ConfigData>
                                <BootStrap>0010800080108000</BootStrap>
                        </Eeprom>
                        <ImageData16x14>424DE60000000000000076000000280000001000000000E000000010
                </Device>
        </Devices>
    </Descriptions>
</EtherCATInfo>
```

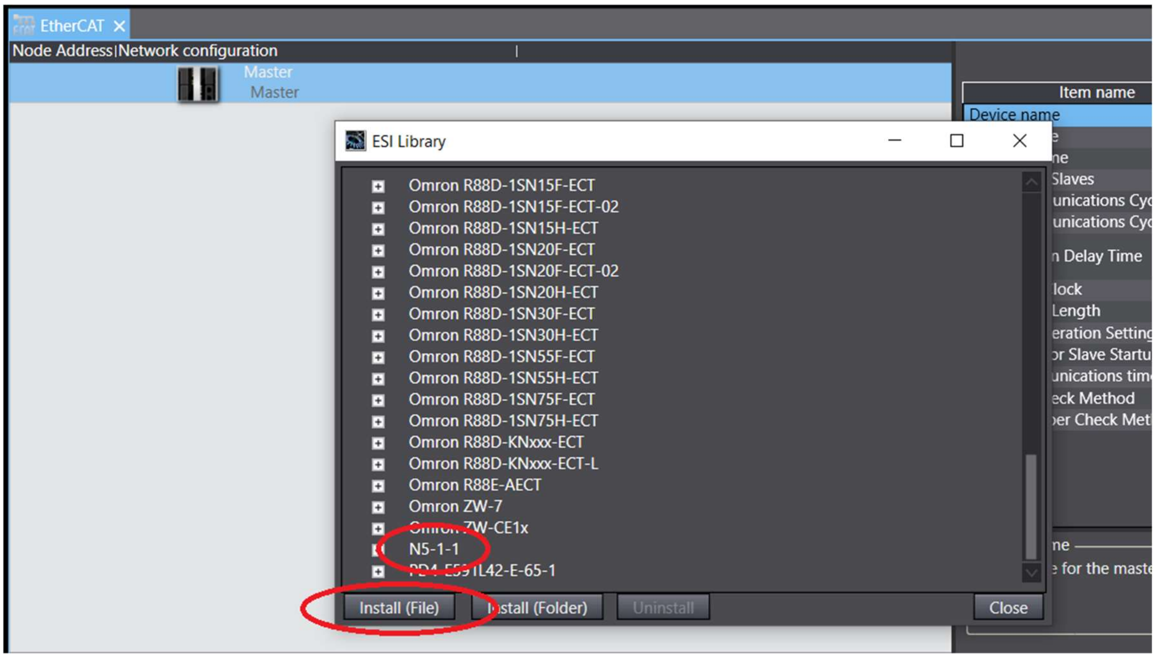
2 Create a new project in Sysmac Studio.

3 Go to **Configurations and Setup -> EtherCAT** in the **Multiview Explorer**.

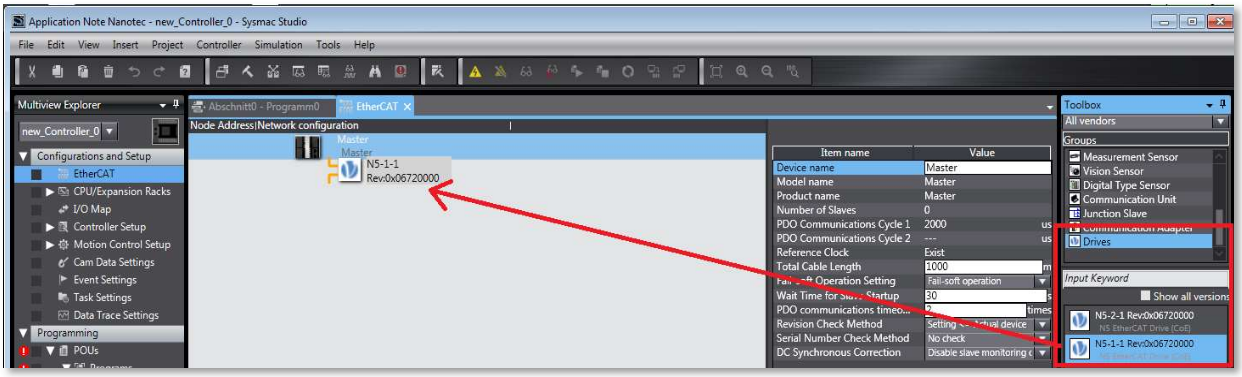
4 Right-click on the **Master** and go to **Display ESI Library** to include the ESI file in your software.



5 To add ESI files, click on **Install (File)**, then choose the ESI file you want to install. After successful installation the drive should be added.

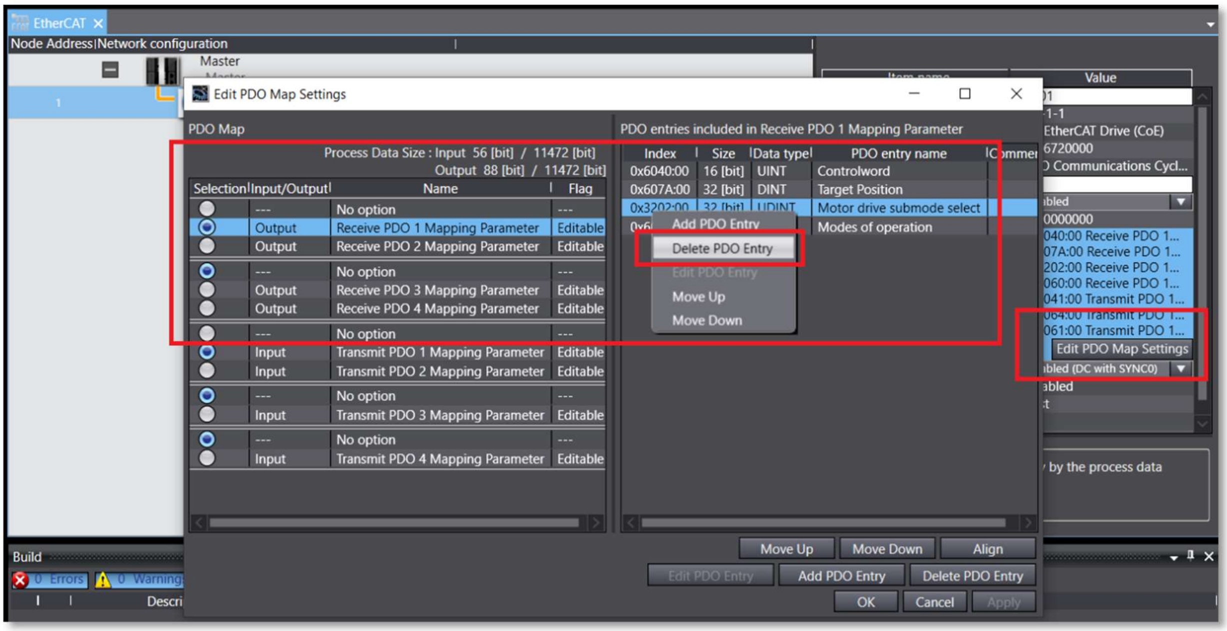


6 Now you should see the added drives in the **Toolbox**. From there you can drag & drop the drive to the **Master**.



Item name	Value
Device name	Master
Model name	Master
Product name	Master
Number of Slaves	0
PDO Communications Cycle 1	2000 us
PDO Communications Cycle 2	--- us
Reference Clock	Exist
Total Cable Length	1000 m
Fail-soft Operation Setting	Fail-soft operation
Wait Time for Slave Startup	30 s
PDO communications timeo...	2 times
Revision Check Method	Setting <= Actual device
Serial Number Check Method	No check
DC Synchronous Correction	Disable slave monitoring c

7 Click on **Edit PDO Map Settings** and select the **Receive PDO 1 Mapping Parameter**. Right-click on **0x3202:00** and **Delete PDO Entry**. This prevents unwanted deactivation of Closed Loop.



PDO Map

Process Data Size : Input 56 [bit] / 11472 [bit]
Output 88 [bit] / 11472 [bit]

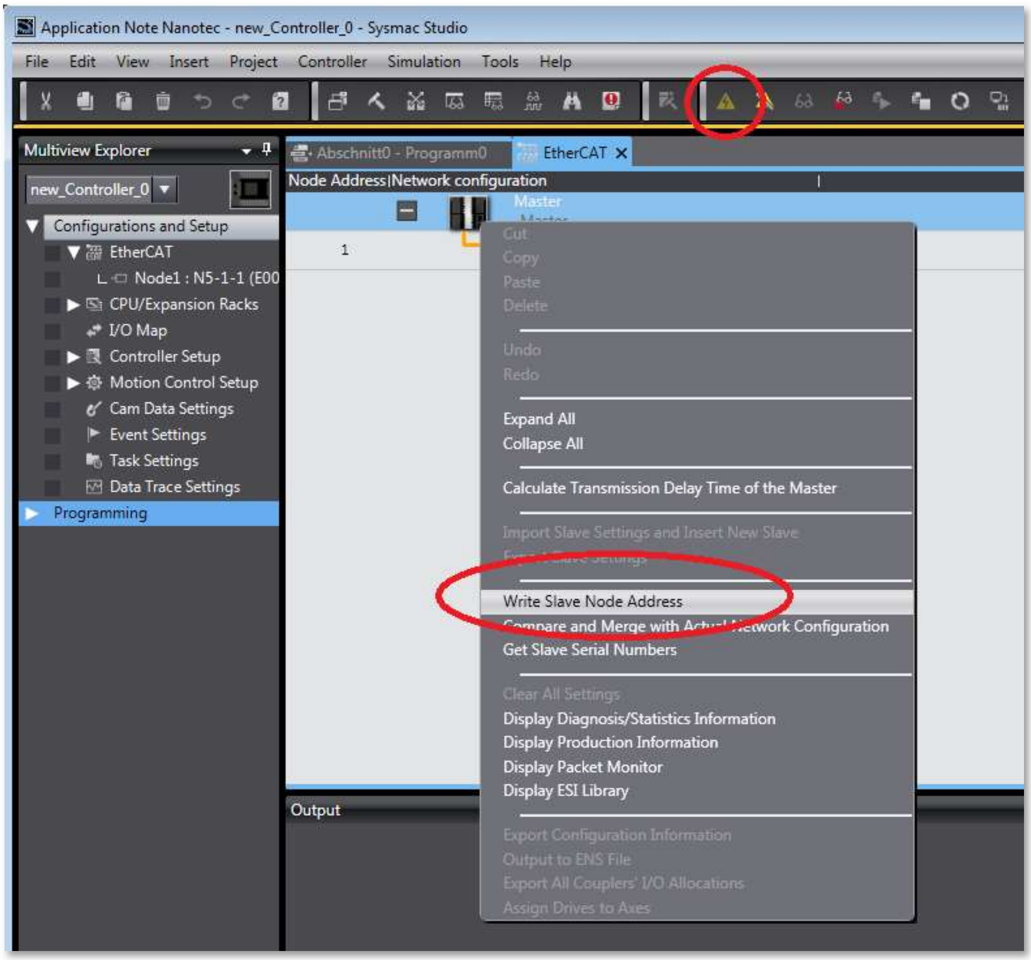
Selection	Input/Output	Name	Flag
○	---	No option	---
●	Output	Receive PDO 1 Mapping Parameter	Editable
○	Output	Receive PDO 2 Mapping Parameter	Editable
●	---	No option	---
○	Output	Receive PDO 3 Mapping Parameter	Editable
○	Output	Receive PDO 4 Mapping Parameter	Editable
○	---	No option	---
●	Input	Transmit PDO 1 Mapping Parameter	Editable
○	Input	Transmit PDO 2 Mapping Parameter	Editable
●	---	No option	---
○	Input	Transmit PDO 3 Mapping Parameter	Editable
●	---	No option	---
○	Input	Transmit PDO 4 Mapping Parameter	Editable

PDO entries included in Receive PDO 1 Mapping Parameter

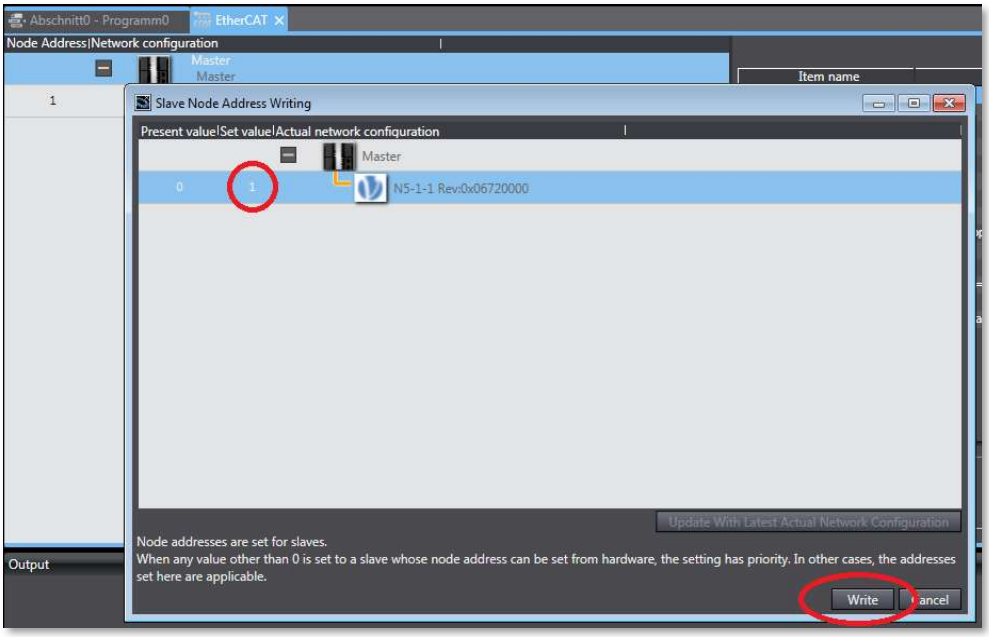
Index	Size	Data type	PDO entry name	Comment
0x6040:00	16 [bit]	UINT	Controlword	
0x607A:00	32 [bit]	DINT	Target Position	
0x3202:00	32 [bit]	UDINT	Motor drive submode select	
0x6[illegible]			Modes of operation	

8 Go Online with the **Master** to set the Node Address for the drives you need.

9 When online, right-click on the **Master** and go to **Write Slave Node Address.**

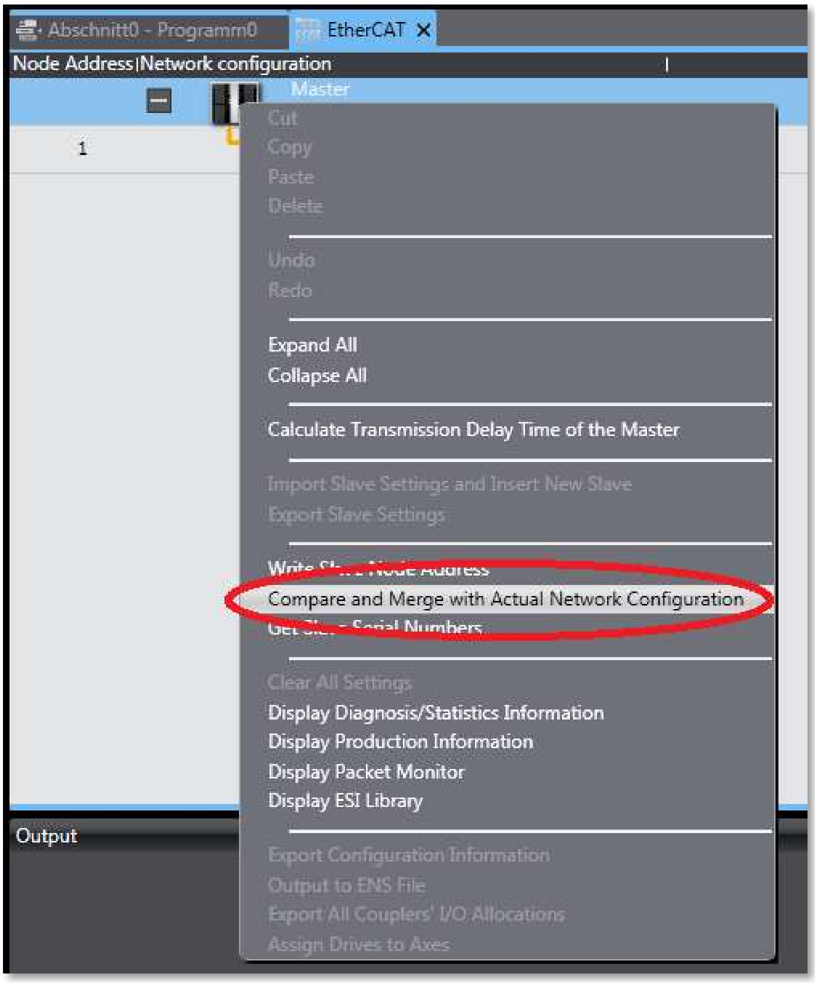


10 Set the desired value and click **Write**. Follow the instructions on the screen and restart the drive.

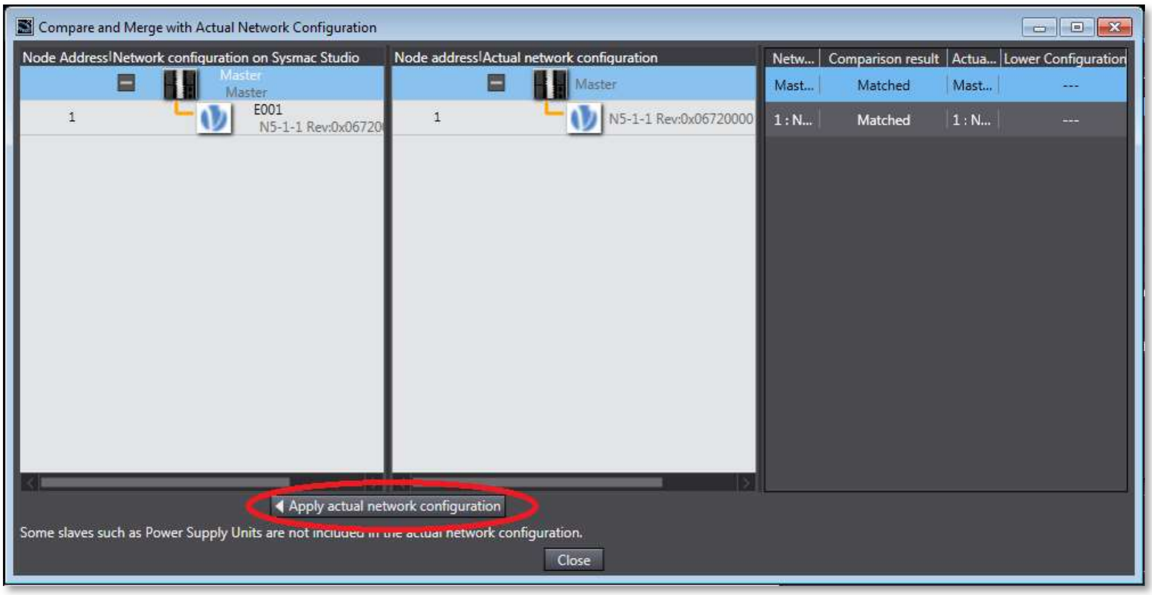

Abschnitt0 - Programm0
EtherCAT
Node Address|Network configuration
Master
Master
Item name
1
Slave Node Address Writing
Present value|Set value|Actual network configuration
Master
0
1
N5-1-1 Rev:0x06720000
Update With Latest Actual Network Configuration
Output
Write
Cancel

Node addresses are set for slaves.
When any value other than 0 is set to a slave whose node address can be set from hardware, the setting has priority. In other cases, the addresses set here are applicable.

11 Verify the correct setup by right-clicking on **Compare and Merge with Actual Network Configuration.**


Abschnitt0 - Programm0
EtherCAT
Node Address|Network configuration
Master
1
Output
Cut
Copy
Paste
Delete
Undo
Redo
Expand All
Collapse All
Calculate Transmission Delay Time of the Master
Import Slave Settings and Insert New Slave
Export Slave Settings
Write Slave Node Address
Compare and Merge with Actual Network Configuration
Get Slave Serial Numbers
Clear All Settings
Display Diagnosis/Statistics Information
Display Production Information
Display Packet Monitor
Display ESI Library
Export Configuration Information
Output to ENS File
Export All Couplers' I/O Allocations
Assign Drives to Axes

12 If they do not match, click on **Apply actual network configuration.**


Compare and Merge with Actual Network Configuration
Node Address|Network configuration on Sysmac Studio
Master
Master
1
E001
N5-1-1 Rev:0x06720
Node address|Actual network configuration
Master
1
N5-1-1 Rev:0x06720000
Apply actual network configuration
Close

Netw...	Comparison result	Actua...	Lower Configuration
Mast...	Matched	Mast...	---
1 : N...	Matched	1 : N...	---

Some slaves such as Power Supply Units are not included in the actual network configuration.

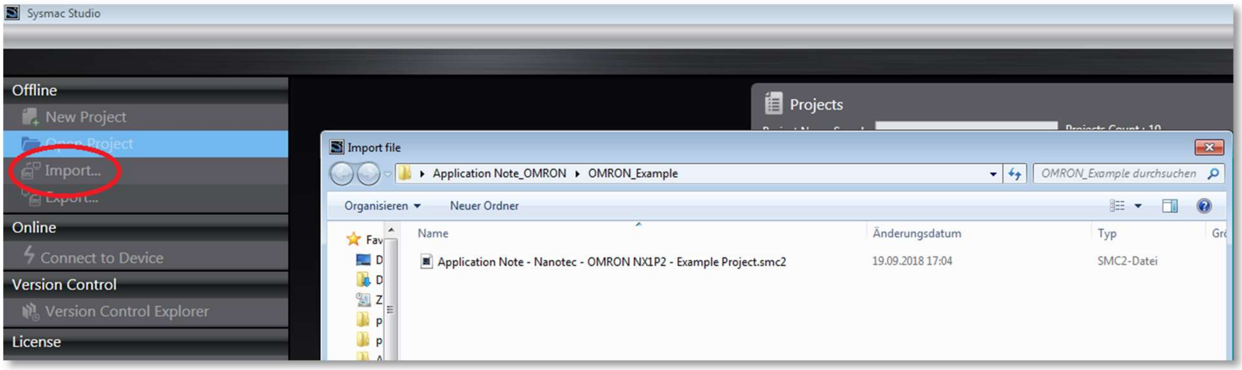
13 If everything was set up correctly, you should now be able to communicate with the drive.

5 Example file for Velocity Mode with SDO commands

The example “Application Note - Nanotec - OMRON NX1P2 - Example Project_V1.1.0” shows you how to drive the motor in the Operation Mode Velocity.

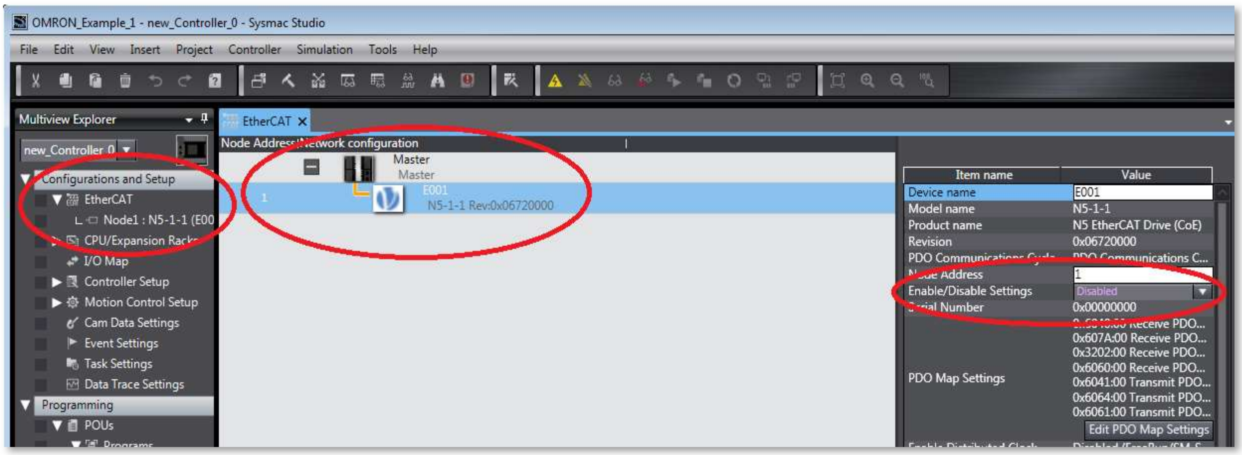
5.1 Description of the example project

Import the example file in the Sysmac Studio.



Name	Änderungsdatum	Typ
Application Note - Nanotec - OMRON NX1P2 - Example Project.smc2	19.09.2018 17:04	SMC2-Datei

The project contains a setup of a master and slave ready for communication via SDO.
Please note that PDO communication is disabled here.



Item name	Value
Device name	E001
Model name	N5-1-1
Product name	N5 EtherCAT Drive (CoE)
Revision	0x06720000
PDO Communications Cycle	PDO Communications C...
Node Address	1
Enable/Disable Settings	Disabled
Serial Number	0x00000000
PDO Map Settings	0x6040:00 Receive PDO... 0x607A:00 Receive PDO... 0x3202:00 Receive PDO... 0x6060:00 Receive PDO... 0x6041:00 Transmit PDO... 0x6064:00 Transmit PDO... 0x6061:00 Transmit PDO... Edit PDO Map Settings
Enable Distributed Clock	Disabled (FreeRun/SM-S...

The program consists of three function blocks that can be executed manually by activating the respective input.

FB_ModeOfOperation: This function block sets the Mode of Operation for a specific slave. The Mode of Operation is set to the value “2” which is Velocity Mode.
To write the Mode of Operation to the drive, set the input to “True”.

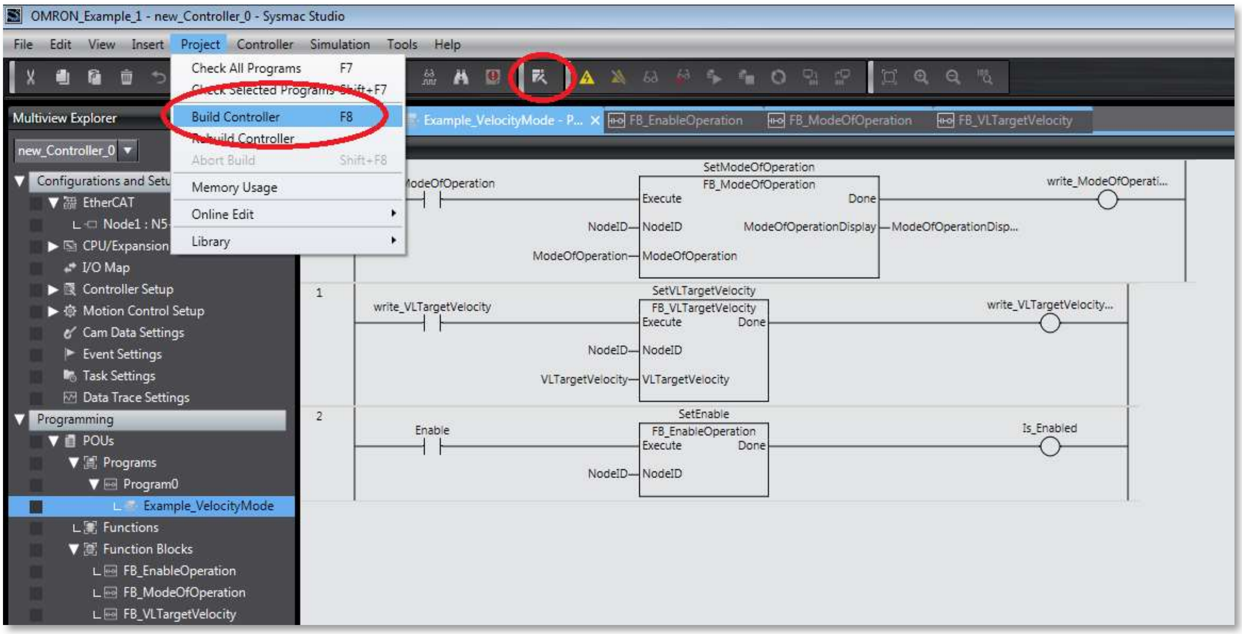
FB_VLTargetVelocity: This function block sets the target velocity for Velocity Mode. You can adjust the value of the variable *VLTargetVelocity*. To write the target velocity to the drive, set the input to “True”.

FB_EnableOperation: This function block puts the drive into the Operation Enabled state. As soon as this state is reached, the motor will be powered and started.
To start the motor, set the input to “True”.
To stop the motor, set the input to “False”.

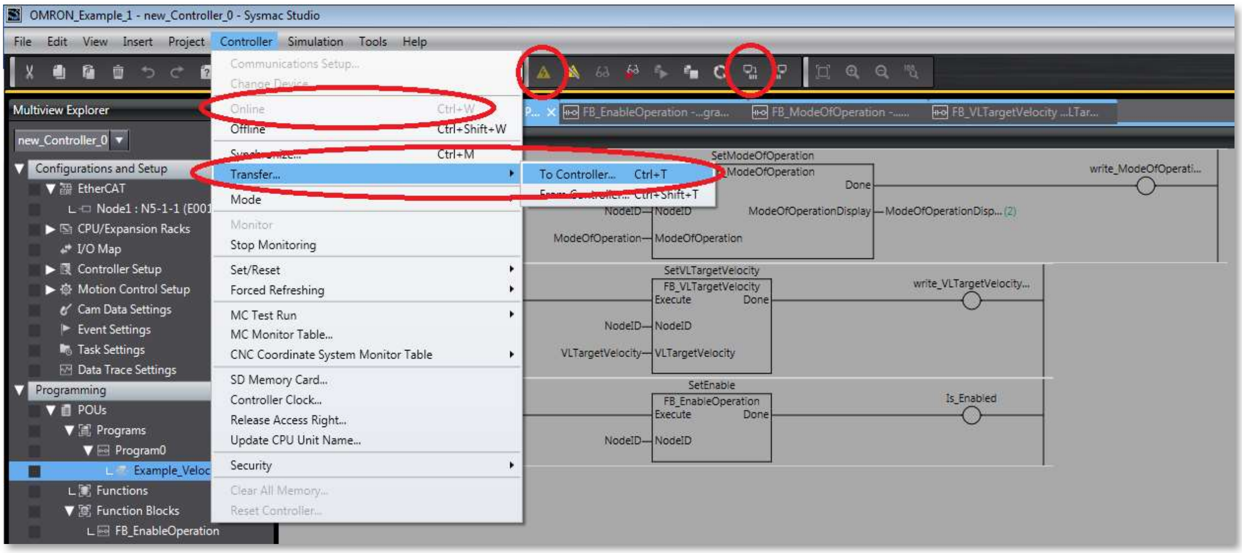
5.2 How to start the motor

Using the example project

1 Build the controller and check for error messages.



2 Go online and transfer the project to the controller.

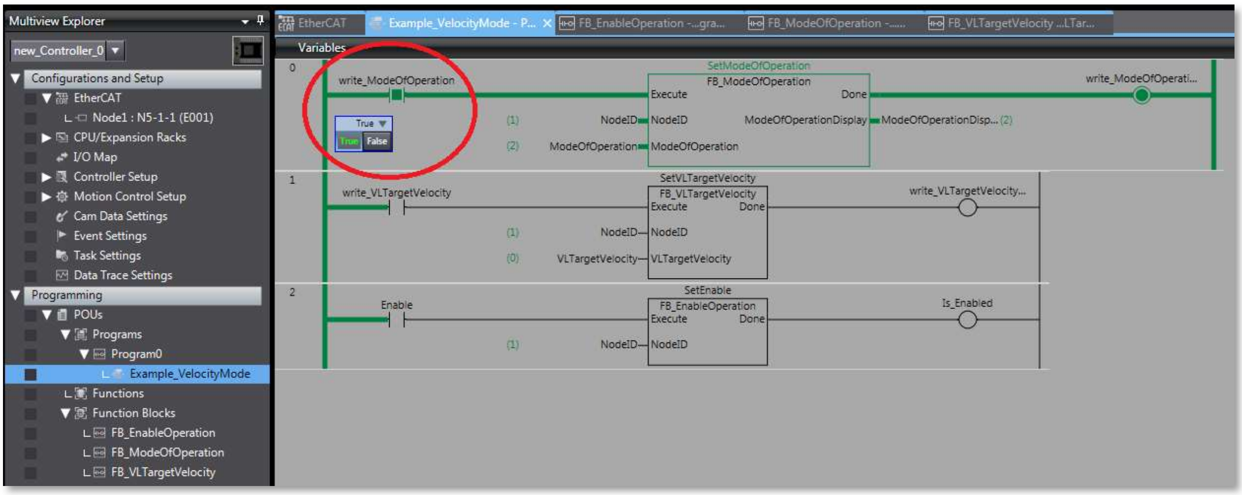


3 Once you enter RUN mode you should see that the program is active.

4 You can manipulate the inputs by double-clicking on the respective input.

Using Function Block FB_ModeOfOperation

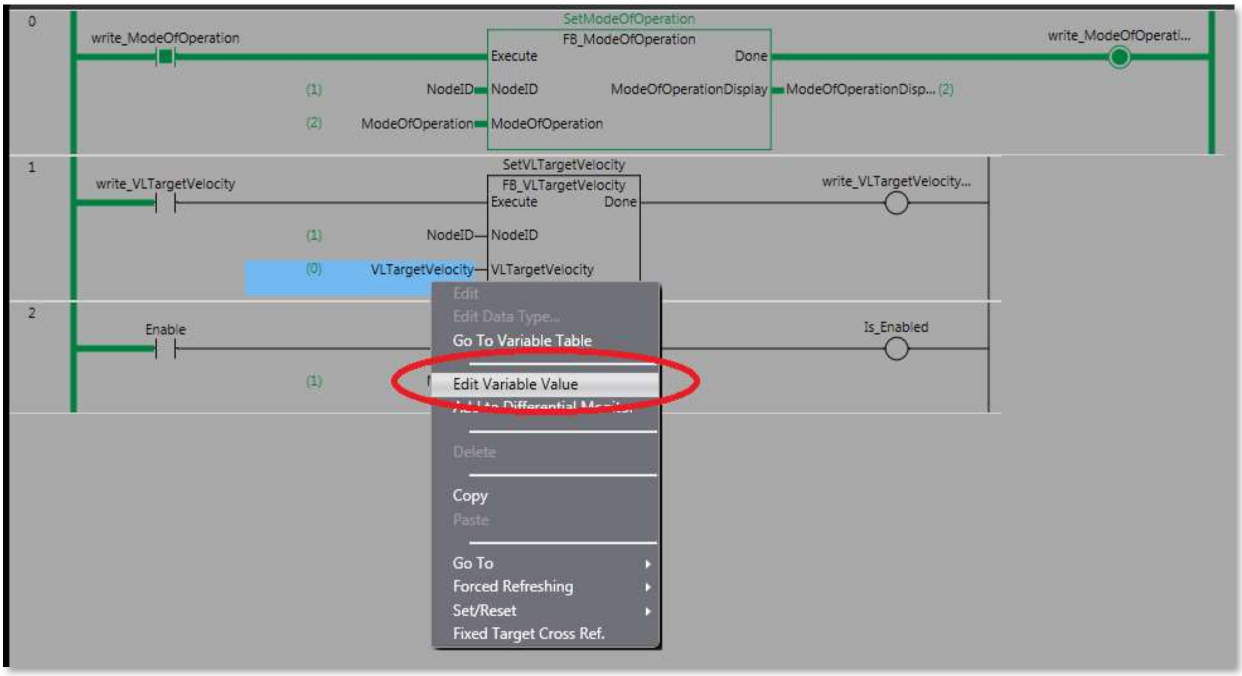
1 Set the input *write_ModeOfOperation* to “True” to set the Mode of Operation to Velocity Mode in the drive. The Node ID is set to “1” by default.



2 You can see which Operation Mode is set by looking at the variable *ModeOfOperationDisplay*. It should be set to the same value as the variable *ModeOfOperation*. Velocity Mode is active if it shows the value “2”.

Using Function Block FB_VLTargetVelocity

1 To set the desired target velocity right-click on the variable and then click **Edit Variable Value**. By default, our controllers are set to rpm. For example, enter the value “60” to set the target velocity to 60rpm.



2 Set the input *write_VLTargetVelocity* to “True” to write the target velocity to the drive.

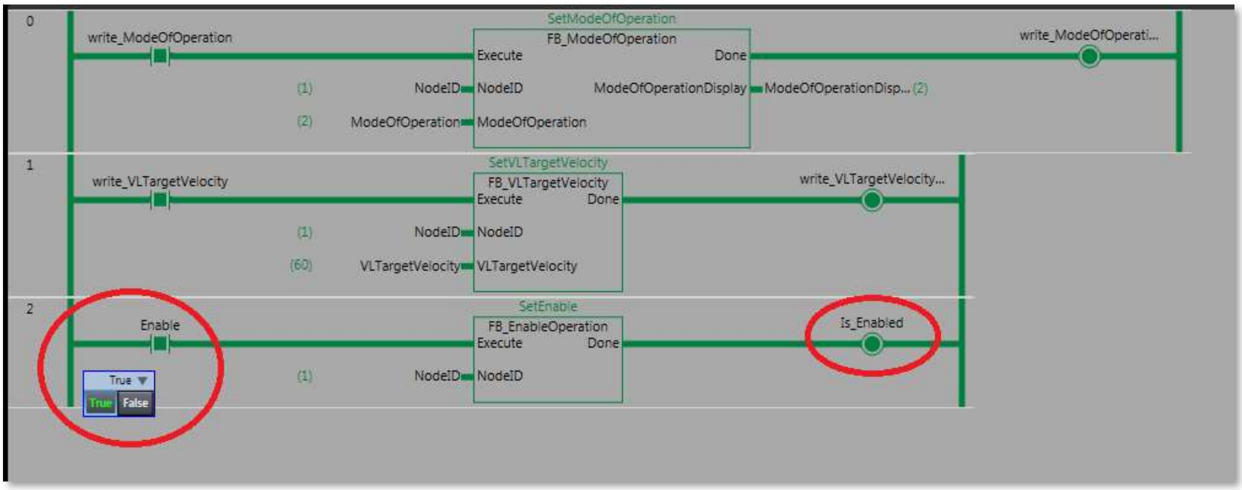
3 To adjust the velocity, simply enter a new value for *VLTargetVelocity* and toggle the input *write_VLTargetVelocity* to “False” and back to “True”. This can also be done while the motor is running.

Using function Block FB_EnableOperation

1 The Mode of Operation and the Target Velocity need to be set before starting the motor.

2 You can start the motor by setting the input *Enable* to “True”. The State Machine should be switched up to the state Operation Enabled. When this is done, the output *Is_Enabled* will be set. This will start the motor movement.

3 To stop the motor, set the input *Enable* to “False”. This will switch the State Machine down, thus stopping the motor movement.

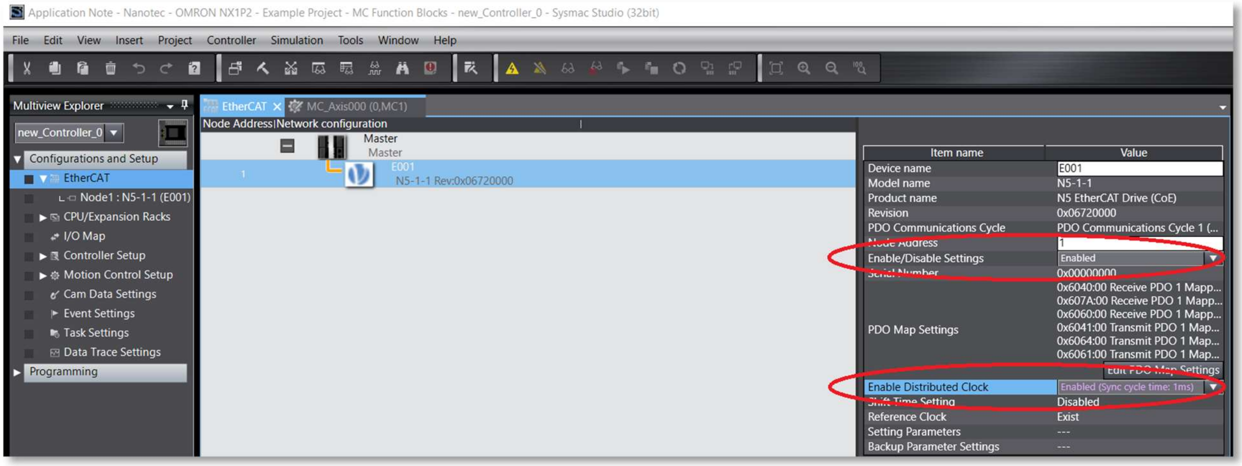


6 Example for positioning with Motion Control function blocks

With this example you will be able to use the Motion Control function blocks to start positioning. The motor will oscillate between two positions.
The project with the complete setup and program is also included as example file “Application Note - Nanotec - OMRON NX1P2 - Example Project - MC Function Blocks_V1.1.0”.

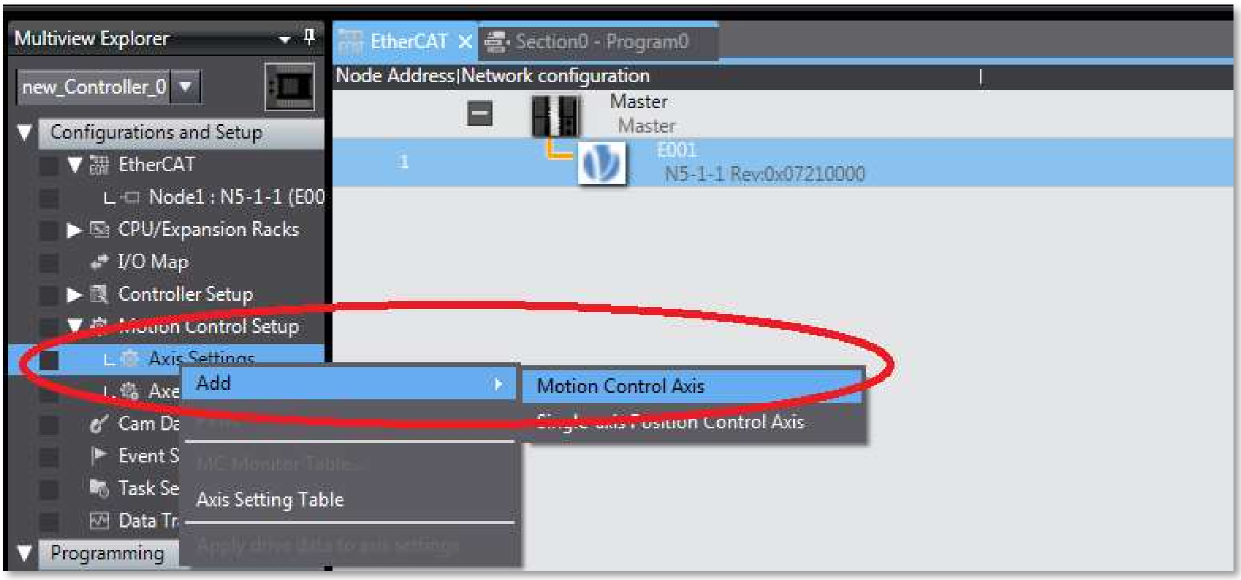
6.1 Setting up the axis

1 Set up communication with the drive as described in chapter **4 Creating a new project and adding the drive**. Please also make sure the prerequisites are met, see chapter **1 Prerequisites**.
Make sure that the drive PDO communication is enabled.
Additionally, make sure that Distributed Clock is enabled.

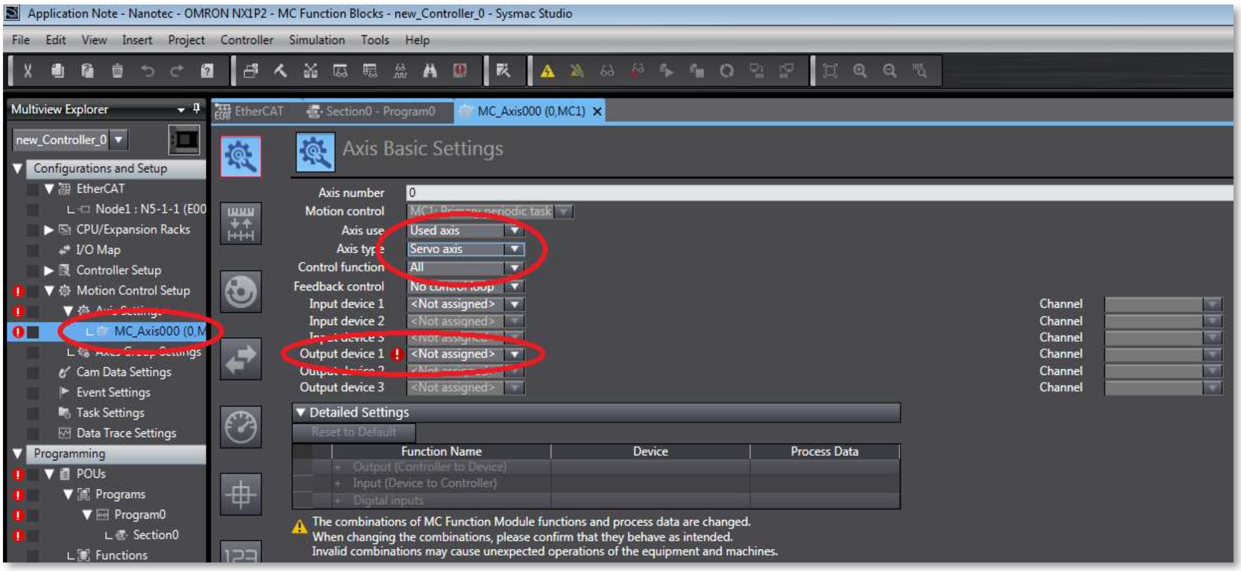


Item name	Value
Device name	E001
Model name	N5-1-1
Product name	N5 EtherCAT Drive (CoE)
Revision	0x06720000
PDO Communications Cycle	PDO Communications Cycle 1 (...
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x6040:00 Receive PDO 1 Mapp... 0x607A:00 Receive PDO 1 Mapp... 0x6060:00 Receive PDO 1 Mapp... 0x6041:00 Transmit PDO 1 Map... 0x6064:00 Transmit PDO 1 Map... 0x6061:00 Transmit PDO 1 Map...
Enable Distributed Clock	Enabled (Sync cycle time: 1ms)
Shift Time Setting	Disabled
Reference Clock	Exist
Setting Parameters	---
Backup Parameter Settings	---

2 Create a new Motion Control axis by right-clicking on **Motion Control Setup ➔ Axis Setting** and go to **Add ➔ Motion Control Axis**.



3 Go to **Axis Basic Settings** and set the axis type to **Servo axis**. Assign the connected drive as **Output device 1**.



Axis Basic Settings

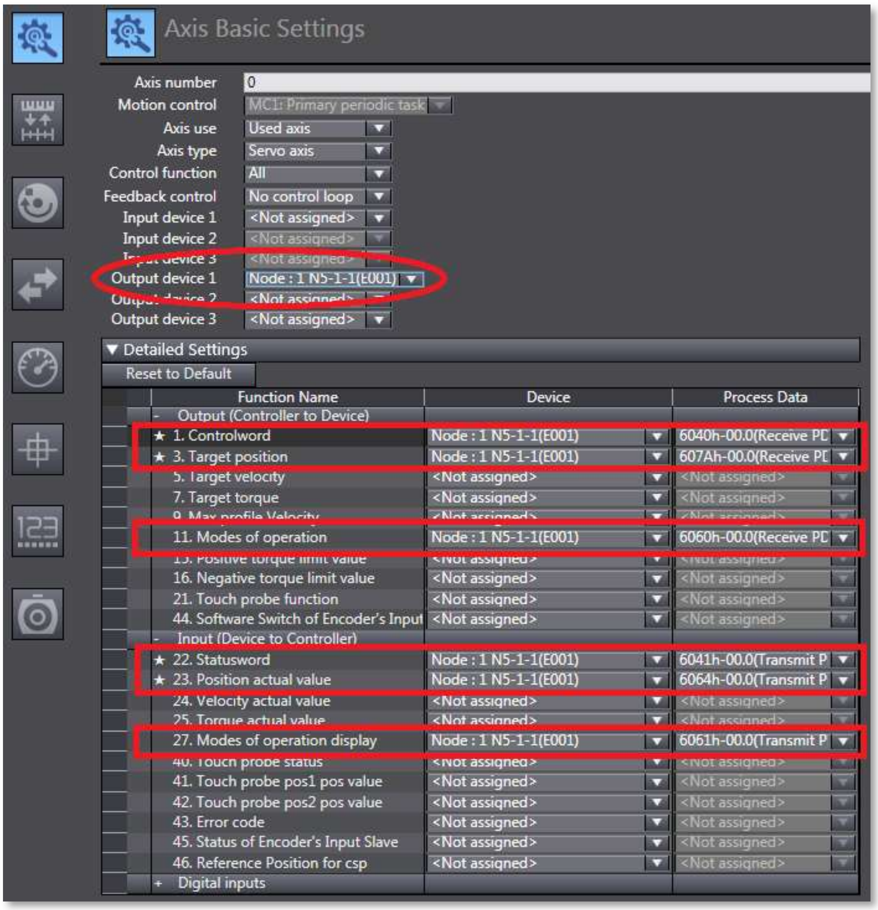
Axis number: 0
Motion control: MC1: Primary periodic task
Axis use: Used axis
Axis type: Servo axis
Control function: All
Feedback control: No control loop
Input device 1: <Not assigned>
Input device 2: <Not assigned>
Input device 3: <Not assigned>
Output device 1: <Not assigned>
Output device 2: <Not assigned>
Output device 3: <Not assigned>

Detailed Settings

Function Name	Device	Process Data
+ Output (Controller to Device)		
+ Input (Device to Controller)		
+ Digital inputs		

The combinations of MC Function Module functions and process data are changed.
When changing the combinations, please confirm that they behave as intended.
Invalid combinations may cause unexpected operations of the equipment and machines.

4 Open **Detailed Settings** and configure the Output-PDOs for **Controlword**, **Target position** and **Modes of operation** as well as the Input-PDOs for **Statusword**, **Position actual value** and **Modes of operation display**.
When done, it should look as shown in the picture below.



Axis Basic Settings

Axis number: 0
Motion control: MC1: Primary periodic task
Axis use: Used axis
Axis type: Servo axis
Control function: All
Feedback control: No control loop
Input device 1: <Not assigned>
Input device 2: <Not assigned>
Input device 3: <Not assigned>
Output device 1: Node : 1 N5-1-1(E001)
Output device 2: <Not assigned>
Output device 3: <Not assigned>

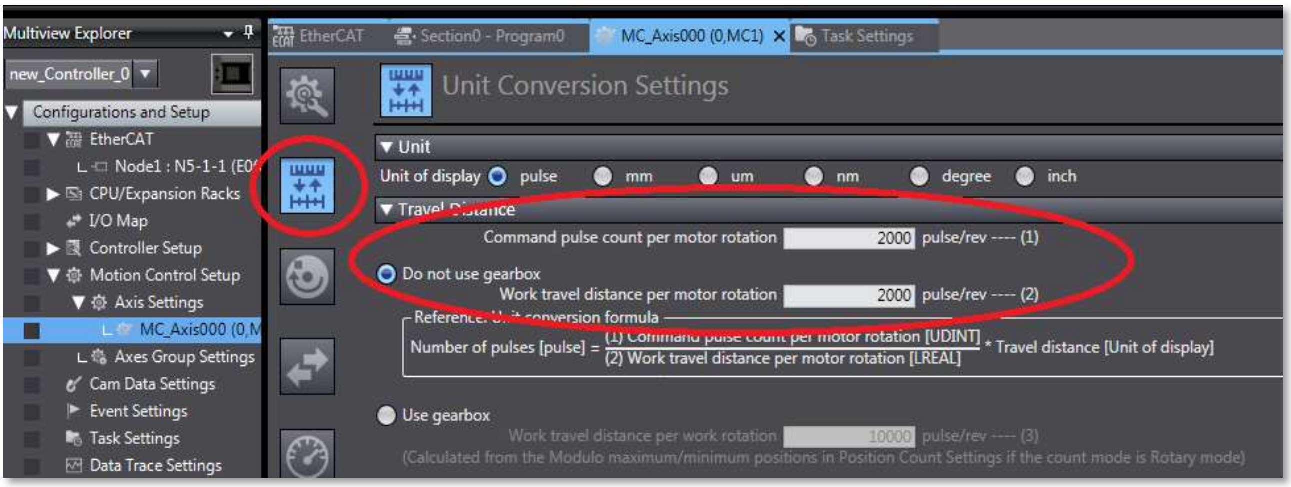
Detailed Settings

Function Name	Device	Process Data
- Output (Controller to Device)		
★ 1. Controlword	Node : 1 N5-1-1(E001)	6040h-00.0(Receive PD
★ 3. Target position	Node : 1 N5-1-1(E001)	607Ah-00.0(Receive PD
5. Target velocity	<Not assigned>	<Not assigned>
7. Target torque	<Not assigned>	<Not assigned>
9. Max profile Velocity	<Not assigned>	<Not assigned>
11. Modes of operation	Node : 1 N5-1-1(E001)	6060h-00.0(Receive PD
15. Positive torque limit value	<Not assigned>	<Not assigned>
16. Negative torque limit value	<Not assigned>	<Not assigned>
21. Touch probe function	<Not assigned>	<Not assigned>
44. Software Switch of Encoder's Input	<Not assigned>	<Not assigned>
- Input (Device to Controller)		
★ 22. Statusword	Node : 1 N5-1-1(E001)	6041h-00.0(Transmit P
★ 23. Position actual value	Node : 1 N5-1-1(E001)	6064h-00.0(Transmit P
24. Velocity actual value	<Not assigned>	<Not assigned>
25. Torque actual value	<Not assigned>	<Not assigned>
27. Modes of operation display	Node : 1 N5-1-1(E001)	6061h-00.0(Transmit P
40. Touch probe status	<Not assigned>	<Not assigned>
41. Touch probe pos1 pos value	<Not assigned>	<Not assigned>
42. Touch probe pos2 pos value	<Not assigned>	<Not assigned>
43. Error code	<Not assigned>	<Not assigned>
45. Status of Encoder's Input Slave	<Not assigned>	<Not assigned>
46. Reference Position for csp	<Not assigned>	<Not assigned>
+ Digital inputs		

5 Go to **Unit Conversion Settings** and set the values according to the drive settings. For the N5 drive the value is set to “2000” (2000 increments per revolution) by default.

Please note:

The value depends on your drive setup. It may be necessary to set a different value than the suggested “2000”. Please set the values accordingly.



Unit Conversion Settings

Unit

Unit of display ● pulse ○ mm ○ um ○ nm ○ degree ○ inch

Travel Distance

Command pulse count per motor rotation 2000 pulse/rev ---- (1)

● Do not use gearbox

Work travel distance per motor rotation 2000 pulse/rev ---- (2)

Reference: Unit conversion formula

Number of pulses [pulse] = (1) Command pulse count per motor rotation [UDINT] / (2) Work travel distance per motor rotation [LREAL] * Travel distance [Unit of display]

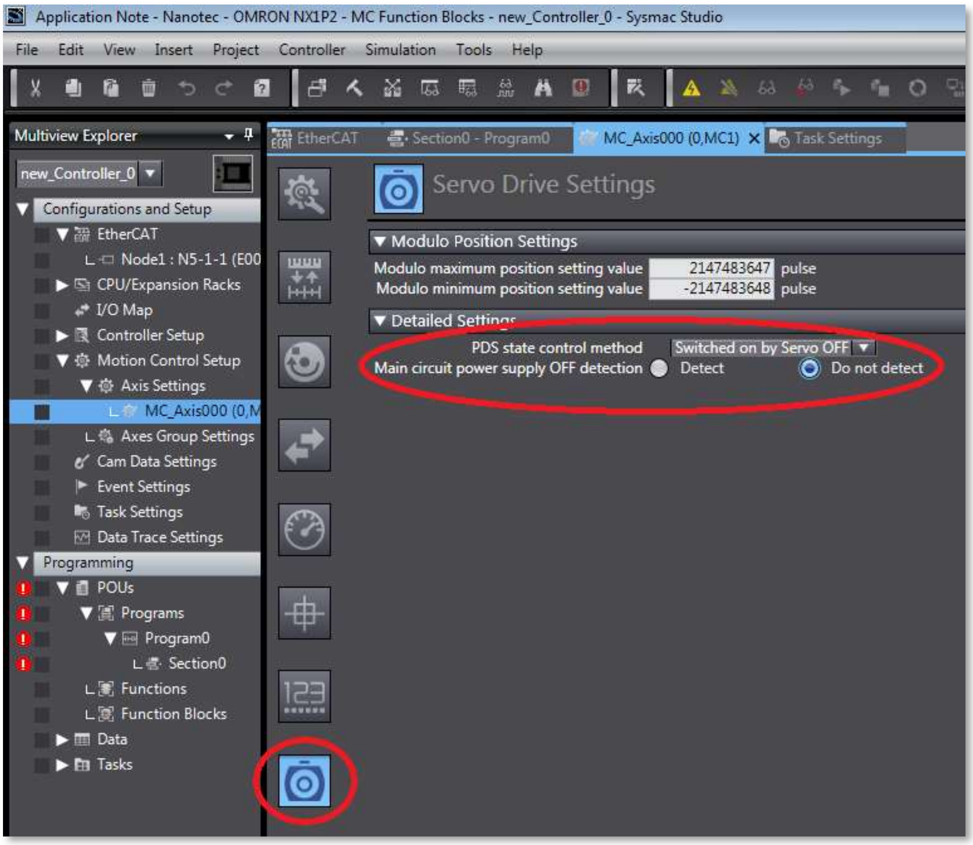
○ Use gearbox

Work travel distance per work rotation 10000 pulse/rev ---- (3)

(Calculated from the Modulo maximum/minimum positions in Position Count Settings if the count mode is Rotary mode)

6 **Important note:**

Go to **Servo Drive Settings** and change **Main circuit power supply OFF detection** to **Do not detect.** If you do not adjust this setting, you may get an error when you try to enable the drive.



Servo Drive Settings

Modulo Position Settings

Modulo maximum position setting value 2147483647 pulse

Modulo minimum position setting value -2147483648 pulse

Detailed Settings

PDS state control method Switched on by Servo OFF

Main circuit power supply OFF detection ○ Detect ● Do not detect

Now the axis should be set up correctly.

6.2 Programming an oscillating movement with MC Function blocks

You can now proceed with programming a short example for a positioning movement.

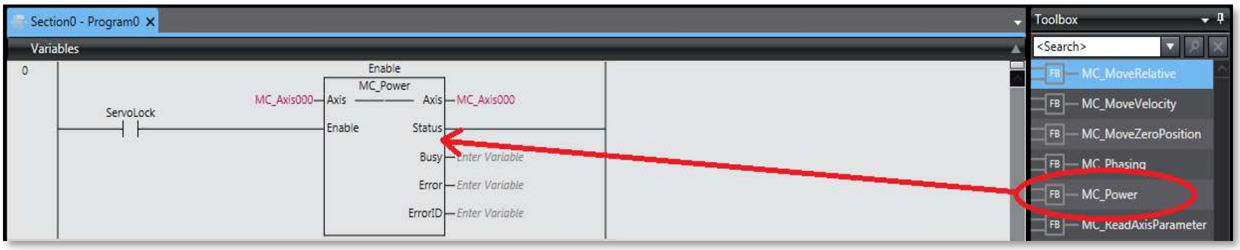
Rung 0 will be used to enable the motor.

1 In Rung 0 you will need one input and the **Motion Control** function block **MC_Power**. You can simply drag & drop the function block from the toolbox to the rung.

2 In the example the input variable is called *ServoLock* and the function block is called **Enable.**

3 Use the axis you set up before. In the example this is “MC_Axis000”.

4 The complete Rung 0 should look as shown in the picture below.



Rung 1 will be used for a repeating series of two relative movements.

1 In Rung 1 you need to include two inputs: One called *StartMovement* and an inverted one called *Complete*.

2 Additionally, you will need function block **MC_MoveRelative** twice. They will simply be called **Move1** and **Move2**.

3 Use the desired axis, in the example it is “MC_Axis000”.

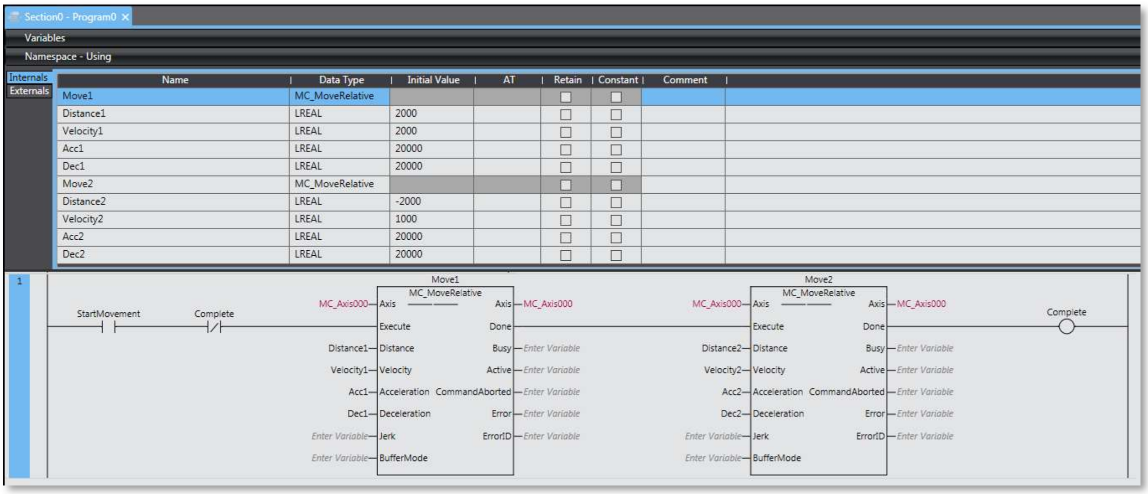
4 Now create variables for the two movement function blocks. When you have created the variables, assign initial values to them.

In the example the distances are set to “2000” and “-2000”. With the axis setup shown in chapter **6.1 Setting up the axis,** this will result in one revolution back and forth. The velocity is set to “2000” respectively “1000” pulses/second, Acceleration and Deceleration are set to “20000” pulses/second².

5 To complete Rung 1, add one Output called *Complete*.

With the Input and Output called *Complete* you will get a repeating functionality. This way the motor will constantly move back and forth until it is stopped.

6 The complete Rung 1 and variables should look like this.



Section0 - Program0

Variables

Namespace - Using

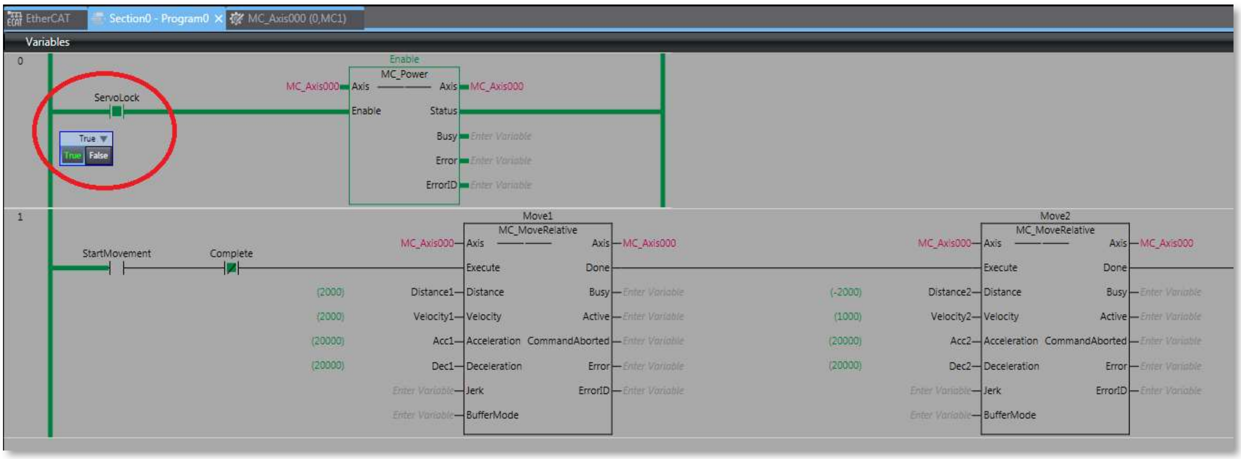
Internals | Externals

Name	Data Type	Initial Value	AT	Retain	Constant	Comment
Move1	MC_MoveRelative			□	□	
Distance1	LREAL	2000		□	□	
Velocity1	LREAL	2000		□	□	
Acc1	LREAL	20000		□	□	
Dec1	LREAL	20000		□	□	
Move2	MC_MoveRelative			□	□	
Distance2	LREAL	-2000		□	□	
Velocity2	LREAL	1000		□	□	
Acc2	LREAL	20000		□	□	
Dec2	LREAL	20000		□	□	

1
StartMovement
Complete
Move1
MC_MoveRelative
MC_Axis000 — Axis — Axis — MC_Axis000
Execute — Done
Distance1 — Distance — Busy — Enter Variable
Velocity1 — Velocity — Active — Enter Variable
Acc1 — Acceleration — CommandAborted — Enter Variable
Dec1 — Deceleration — Error — Enter Variable
Enter Variable — Jerk — ErrorID — Enter Variable
Enter Variable — BufferMode
Move2
MC_MoveRelative
MC_Axis000 — Axis — Axis — MC_Axis000
Execute — Done
Distance2 — Distance — Busy — Enter Variable
Velocity2 — Velocity — Active — Enter Variable
Acc2 — Acceleration — CommandAborted — Enter Variable
Dec2 — Deceleration — Error — Enter Variable
Enter Variable — Jerk — ErrorID — Enter Variable
Enter Variable — BufferMode
Complete

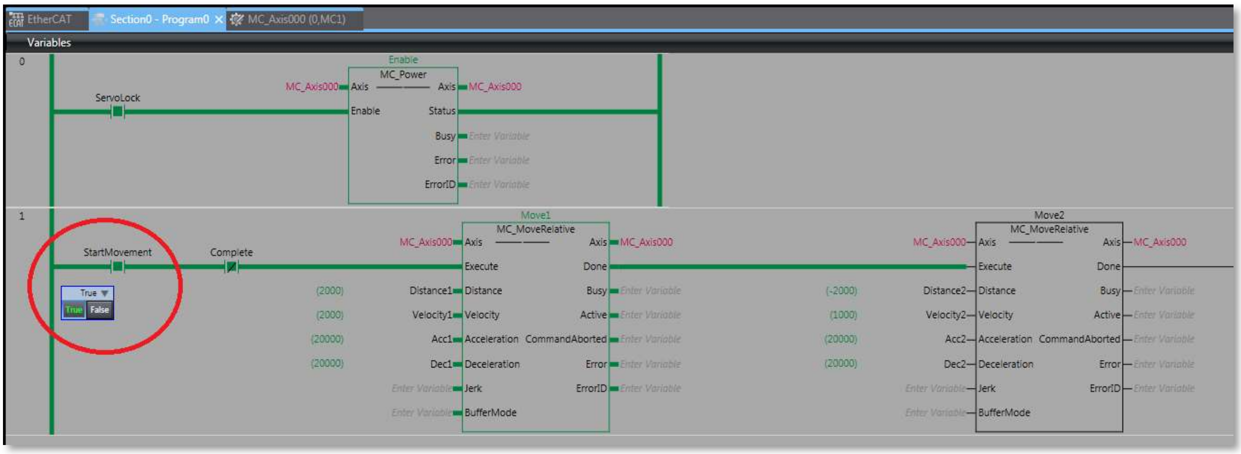
7 To transfer the program to the PLC and operate the program, please follow the instructions given in chapter **5.2 How to start the motor.** After successfully transferring and starting the program, you are now able to operate the drive and motor.

8 To power up the motor go to Rung 0 and set the input *ServoLock* to "True".



EtherCAT
Section0 - Program0
MC_Axis000 (0,MC1)
Variables
0
ServoLock
True
True False
Enable
MC_Power
MC_Axis000 — Axis — Axis — MC_Axis000
Enable — Status
Busy — Enter Variable
Error — Enter Variable
ErrorID — Enter Variable
1
StartMovement
Complete
Move1
MC_MoveRelative
MC_Axis000 — Axis — Axis — MC_Axis000
Execute — Done
(2000) Distance1 — Distance — Busy — Enter Variable
(2000) Velocity1 — Velocity — Active — Enter Variable
(20000) Acc1 — Acceleration — CommandAborted — Enter Variable
(20000) Dec1 — Deceleration — Error — Enter Variable
Enter Variable — Jerk — ErrorID — Enter Variable
Enter Variable — BufferMode
Move2
MC_MoveRelative
MC_Axis000 — Axis — Axis — MC_Axis000
Execute — Done
(-2000) Distance2 — Distance — Busy — Enter Variable
(1000) Velocity2 — Velocity — Active — Enter Variable
(20000) Acc2 — Acceleration — CommandAborted — Enter Variable
(20000) Dec2 — Deceleration — Error — Enter Variable
Enter Variable — Jerk — ErrorID — Enter Variable
Enter Variable — BufferMode

9 To start a movement, set the input *StartMovement* to "True". The drive will then do two movements repeatedly.



EtherCAT
Section0 - Program0
MC_Axis000 (0,MC1)
Variables
0
ServoLock
Enable
MC_Power
MC_Axis000 — Axis — Axis — MC_Axis000
Enable — Status
Busy — Enter Variable
Error — Enter Variable
ErrorID — Enter Variable
1
StartMovement
True
True False
Complete
Move1
MC_MoveRelative
MC_Axis000 — Axis — Axis — MC_Axis000
Execute — Done
(2000) Distance1 — Distance — Busy — Enter Variable
(2000) Velocity1 — Velocity — Active — Enter Variable
(20000) Acc1 — Acceleration — CommandAborted — Enter Variable
(20000) Dec1 — Deceleration — Error — Enter Variable
Enter Variable — Jerk — ErrorID — Enter Variable
Enter Variable — BufferMode
Move2
MC_MoveRelative
MC_Axis000 — Axis — Axis — MC_Axis000
Execute — Done
(-2000) Distance2 — Distance — Busy — Enter Variable
(1000) Velocity2 — Velocity — Active — Enter Variable
(20000) Acc2 — Acceleration — CommandAborted — Enter Variable
(20000) Dec2 — Deceleration — Error — Enter Variable
Enter Variable — Jerk — ErrorID — Enter Variable
Enter Variable — BufferMode

10 To stop the movement after the last two movement commands are executed, set the input *StartMovement* to "False".

11 To stop the movement instantly, set the input *ServoLock* to “False”.

These examples will help you to start up a Nanotec controller/drive with an OMRON PLC.

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