



Title: Interfacing Gocator over CANopen using Hilscher Gateway

Table of Contents

1 Overview2

2 Software and Hardware Requirements2

3 Configuring the Gocator2

 3.1 Gocator Modbus Registers3

4 Configuring Hilscher netTap NT-50-CO-EN Gateway4

 4.1 Gateway Hardware Interface4

 4.2 SYCON.net Software Installation4

 4.3 Configuration of Gateway IP Address5

 4.4 Configuration of Gateway in SYCON.net6

 4.4.1 Configure Open Modbus TCP9

 4.4.2 Configure CANopen13

5 Configuring IFM CR0403 BasicController16

 5.1 Setup the programming system via templates16

 5.2 Configuring CANopen Master Parameters17

 5.3 Add Hilscher NT-50 Gateway EDS18

 5.4 Configuring Hilscher NT-50 Gateway Parameters18



1 Overview

This application note demonstrates how to interface to a Gocator via CANopen protocol, using Hilscher Gateway to convert from CANopen to Modbus protocol, which is supported by the Gocator.

2 Software and Hardware Requirements

Requirements	Details
Gocator Firmware	Gocator 3.x or 4.x
Gocator Series	All
PLC	IFM CR0403 BasicController
PLC Software	CoDeSys V2.3
Gateway	Hilscher netTAP (NT-50-CO-EN)
Gateway Software	SYCON.net

3 Configuring the Gocator

The Gocator is to be configured to use the Modbus protocol in Output configuration.

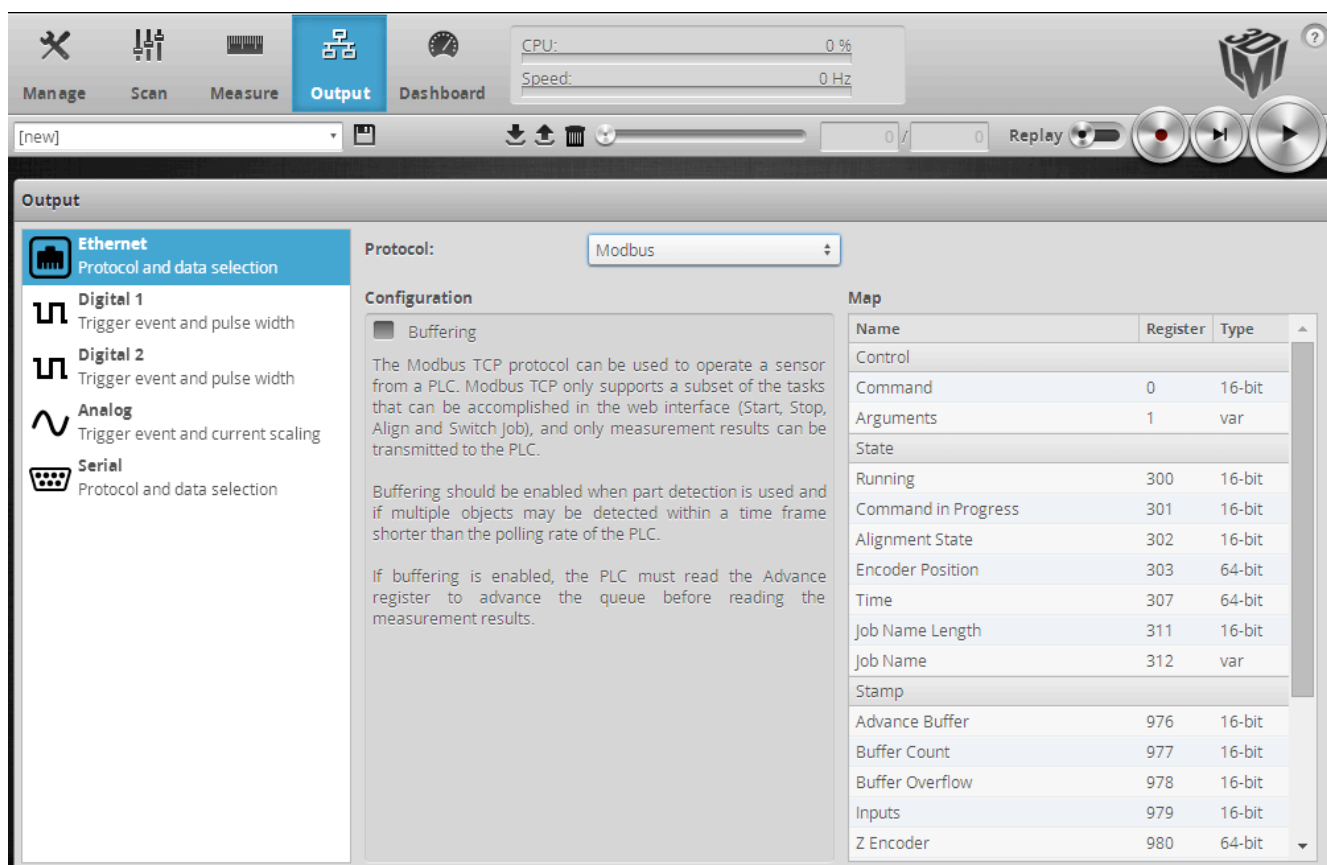


Figure 1 - Configuring the Modbus protocol in the Output panel



3.1 Gocator Modbus Registers

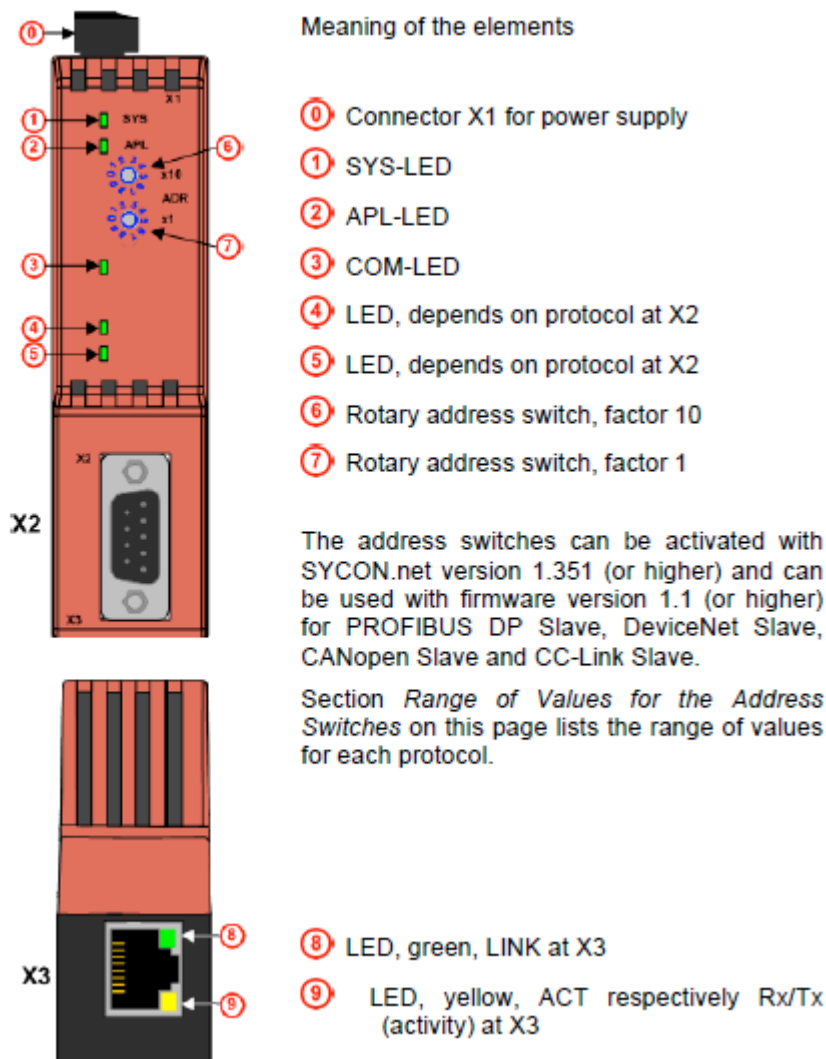
Below is the table with the Gocator modbus registers which need to be mapped by the gateway to the Profinet IO registers. Note the CANopen protocol supports only 32 bytes of data for send Process Data Objects (PDOs) and 32 bytes for receive PDOs.

Modbus Write Registers		
Register Address	Name	DataSize (bit)
0	Command	16
1	Filename char_0	16
-----	-----	-----
31	Filename char_31	16
Modbus Read Registers		
Sensor State Registers		
300	Stopped / Running	16
301	Busy	16
302	Calibration State	16
303 – 306	Encoder Value	64
307 – 310	Time	64
311	Configuration File Length	16
312	Live Configuration Name char_0	16
-----	-----	-----
331	Live Configuration Name char_19	16
Sensor Stamp Registers		
976	Buffer Advance	16
977	Buffer Counter	16
978	Buffer Overflow	16
979	Inputs	16
980 – 983	Encoder Index	64
984 – 985	Exposure	32
986 – 987	Temperature	32
988 – 991	Encoder Value	64
992 – 995	TimeStamp	64
996 – 999	Frame Counter	64
Sensor Measurement Registers		
1000 – 1001	Measurement Value ID 0	32
1002	Measurement Decision ID 0	16
-----	-----	-----
1057 - 1058	Measurement Value ID 19	32
1059	Measurement Decision ID 19	16



4 Configuring Hilscher netTap NT-50-CO-EN Gateway

4.1 Gateway Hardware Interface



Port X2 – D-Sub (DB9) connector for CANopen connection

Port X3 – Ethernet port for Modbus TCP connection

4.2 SYCON.net Software Installation

The gateway is delivered with a “Gateway Solutions” CD, containing configuration tools, drivers, manuals, etc. Run the “Gateway_Solutions.exe” executable from the CDE, screen as shown in **Error! Reference source not found.** will appear. Click on “Install Configuration and Diagnostic Software” to install the necessary software for the gateway.



Figure 2 - Installing Sycon.net Configuration Tool

4.3 Configuration of Gateway IP Address

The device is configured via the Ethernet port. Therefore it is necessary that the device gets an IP address assigned before.

Therefore do the followings steps:

1. Establish an Ethernet connection between the Ethernet network port of your PC and the Ethernet port of the netTAP NT 50 device
2. Start the "Ethernet-Device Setup" software. Therefore select **Start > Programs > SYCON.net System Configurator > Ethernet Device Setup**.
3. Search for connected devices. Therefore click on **Search Devices**.
Devices are searched in the local network using broadcast telegrams.
4. Assign an IP address to the NT 50 device, which should be used for the device configuration.
This address can be stored in a non-volatile memory of the device.

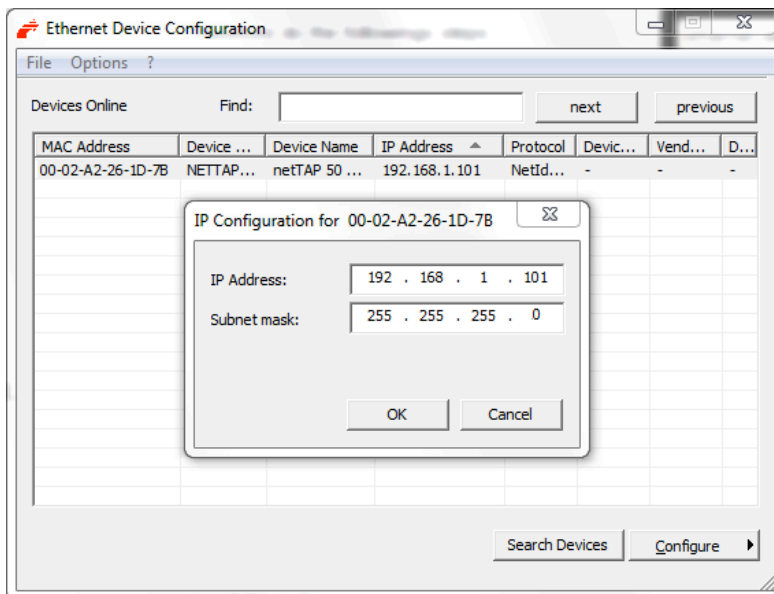


Figure 3 - Ethernet Device Configuration for Gateway

4.4 Configuration of Gateway in SYCON.net

To configure the gateway, open the SYCON tool which will automatically start a new project.

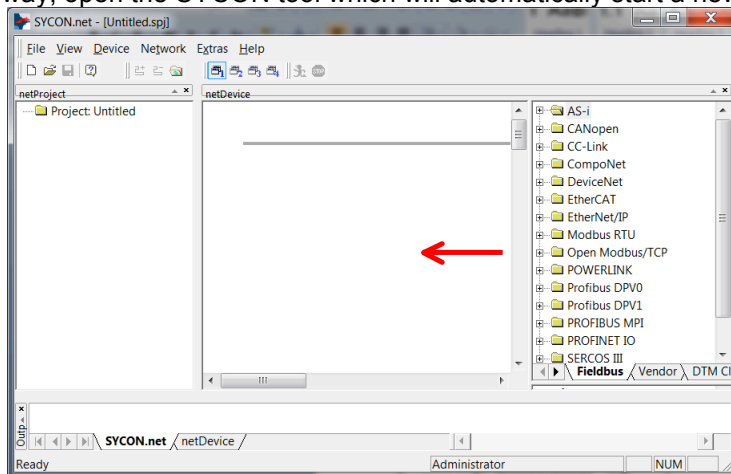


Figure 4 - SYCON, empty project



From the right panel, select the “NT 50-XX-XX” under “Open ModbusTCP / Gateway_Stand-Alone_Slave” and drag this to the grey line in the middle panel.

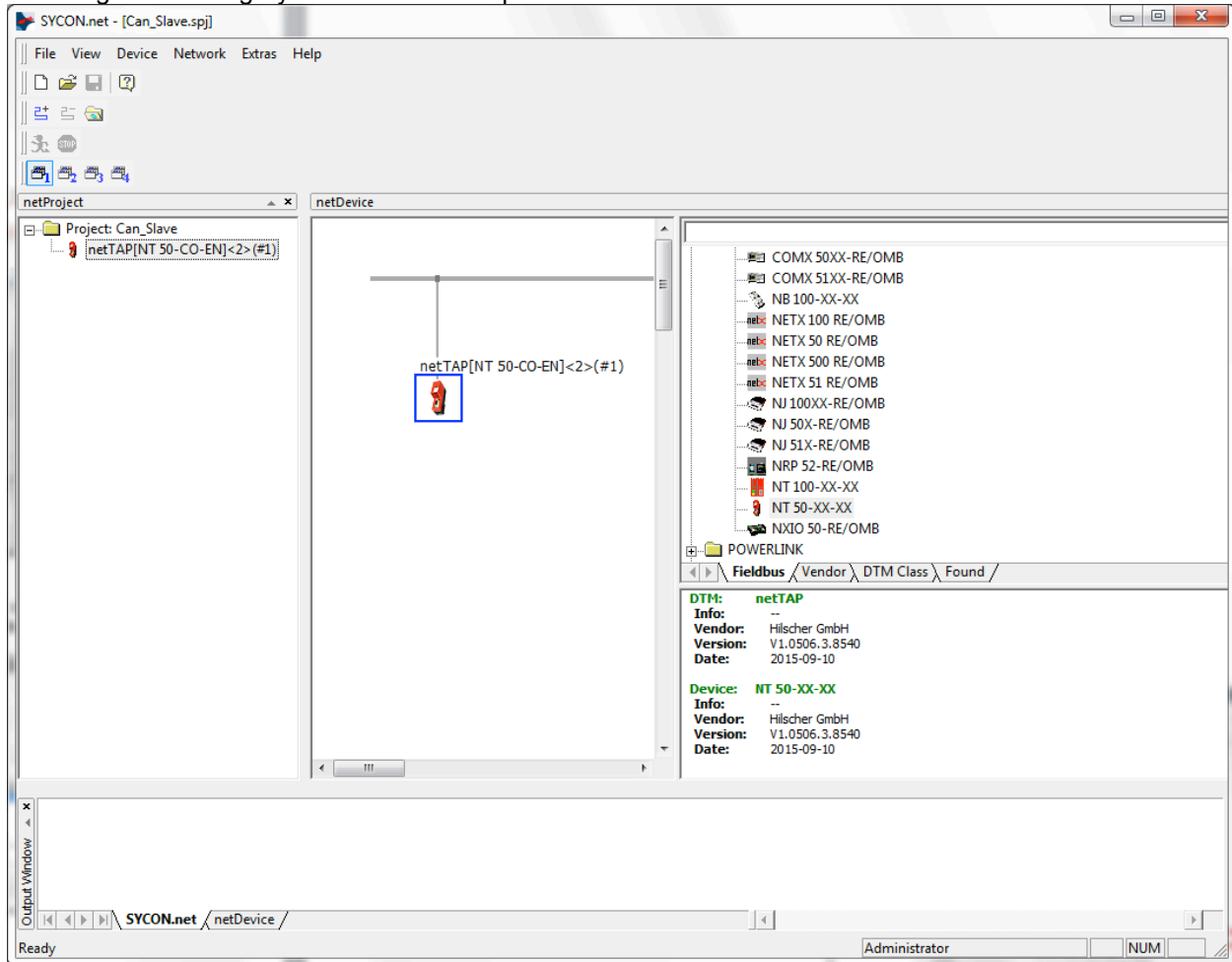


Figure 5 - Selecting the netTap

Right click on the netTap and click “Configuration→Gateway”. Click the “Scan” button and the NT 50-CO-EN device will appear. Select this NT 50-CO-EN and click “Apply” to assign the device.

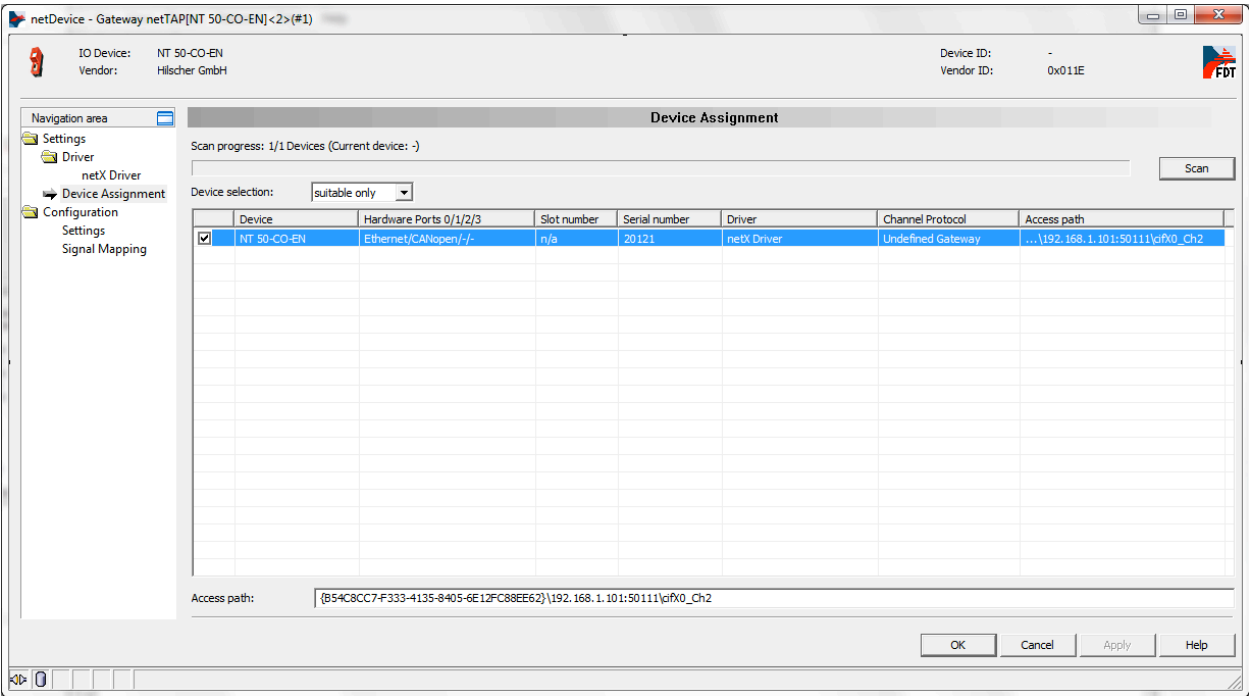


Figure 6 - Device assignment



Configure the Primary network as CANopen Slave and the Secondary network as Open Modbus/TCP. Only the suitable “N5COSOMB.NXF” firmware will appear on the settings panel. Select and download this firmware to the netTap by clicking the “Download” button. After that, click “OK”.

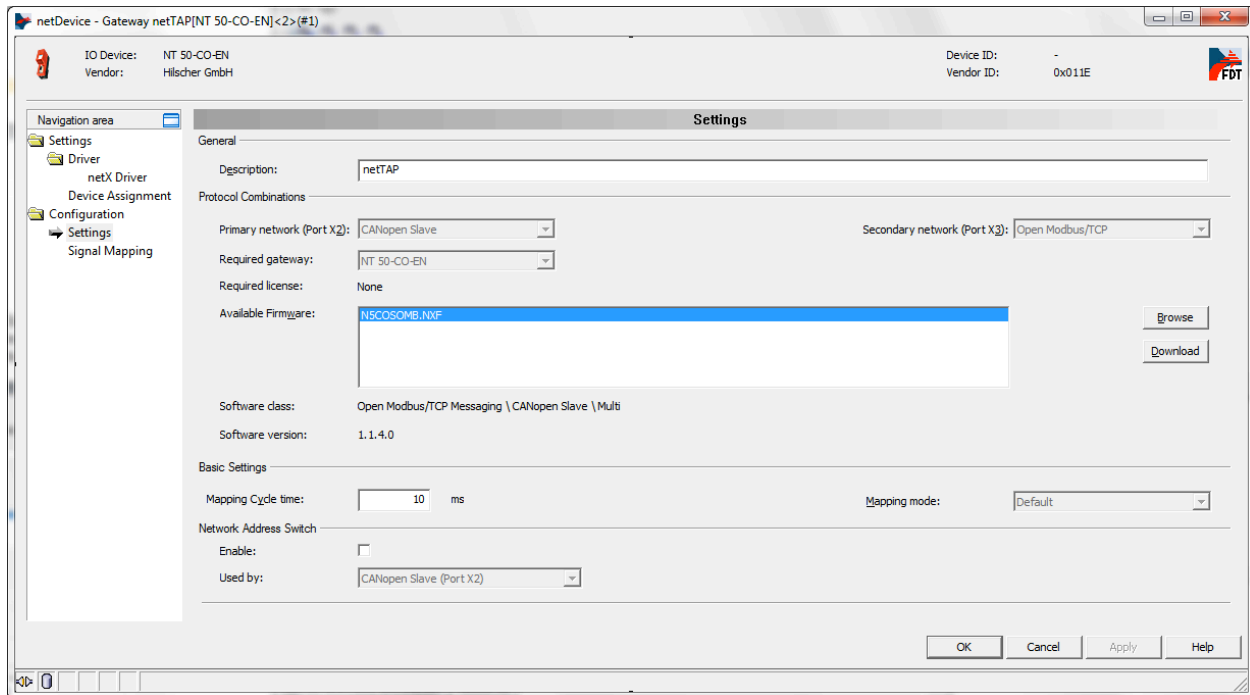
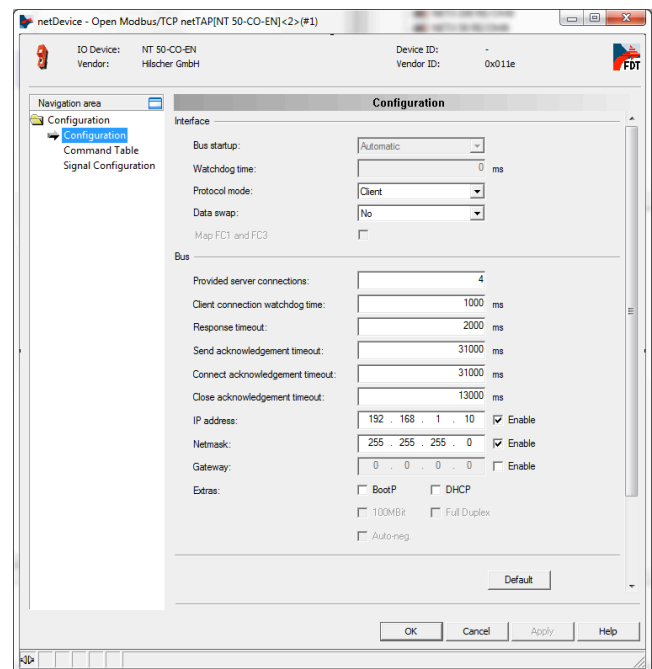


Figure 7 - Downloading firmware to netTap

4.4.1 Configure Open Modbus TCP

Configure the “Open Modbus TCP network” by right clicking on the netTap and then “Configuration→Open Modbus TCP”.

- Change the protocol mode to “Client”
- Set the Data swap to “No”
- Disable DHCP
- Set the Modbus network IP address to a free address with the first 3 digits similar as the address of the sensor.
- Set the Netmask to 255.255.255.0





Add commands to the command table (see Figure 9):

Note the CANopen protocol supports only 32 bytes of data for send Process Data Objects (PDOs) and 32 bytes for receive PDOs. Command Table entries with number of registers larger than 32 bytes will be truncated to 32 bytes.

- Function "Read Input Registers", starting on register 1000 with a length of 32 registers, trigger mode "Cyclically", cycle time 1ms. These are for reading the measurement registers.

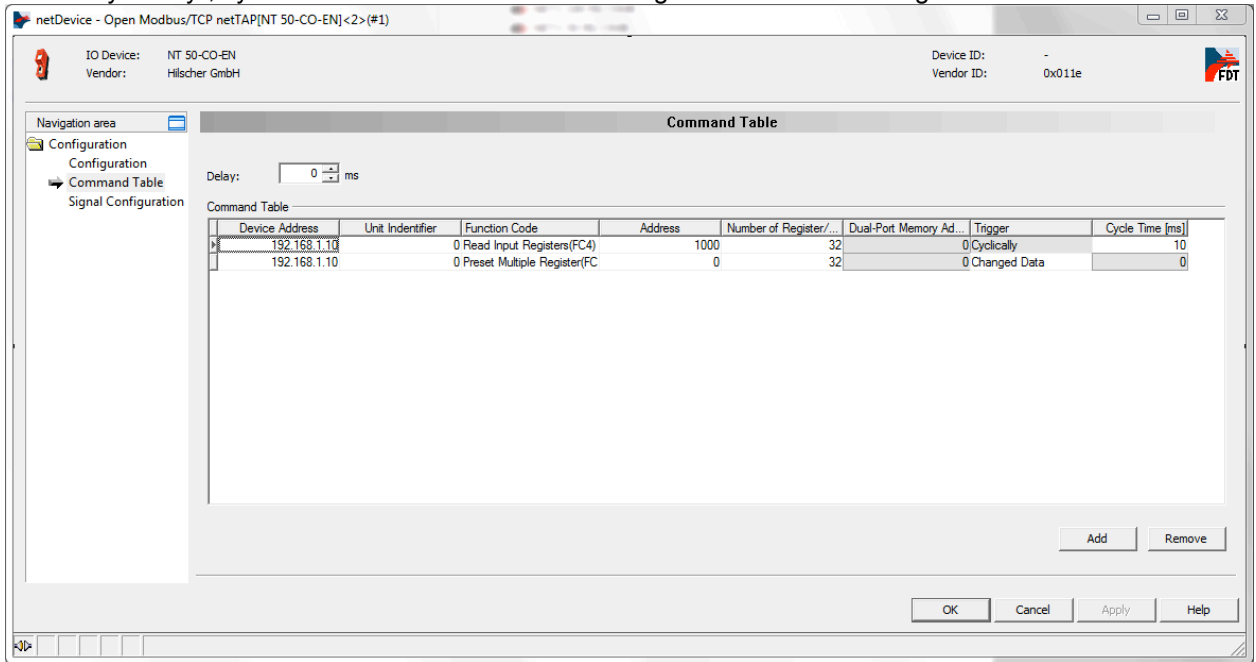


Figure 9 - Command Table

- The Address can be configured to start at different register address to read different sensor registers. For example, setting the Address to start at 300 will allow us to read sensor state registers. Combination of multiple command tables is also possible. In the example command table shown below, the first 6 bytes of registers starting from address 1000 is read, followed by 26



bytes of registers from address 300.

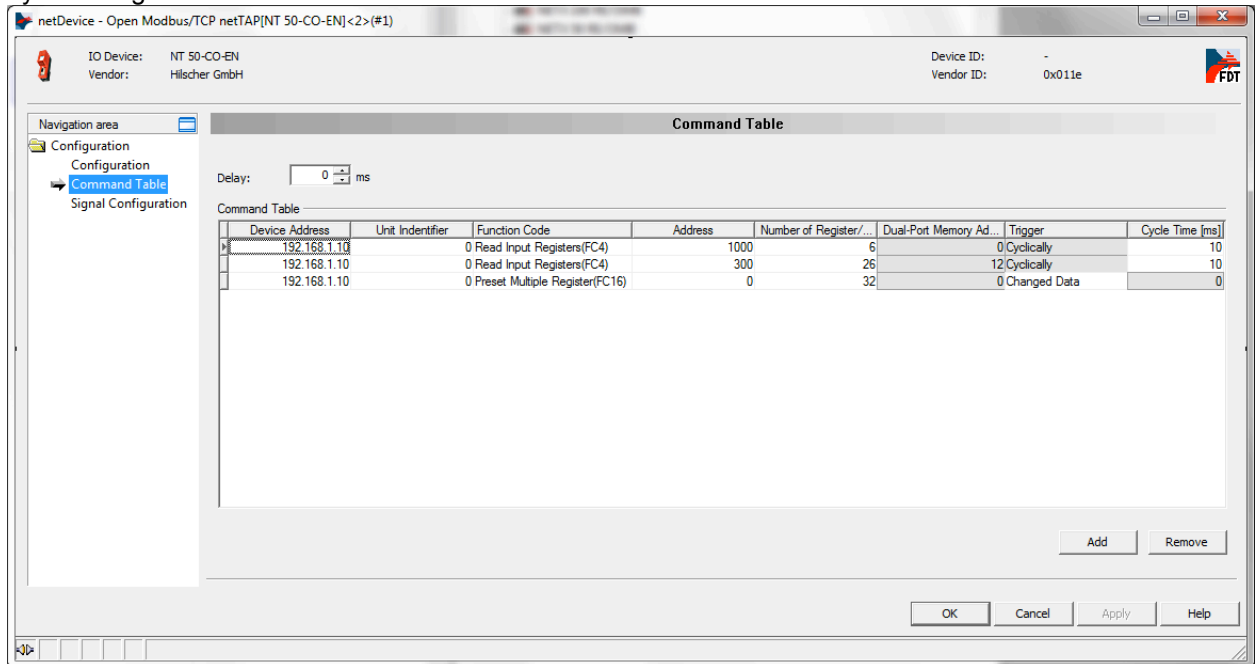


Figure 10 - Command Table

- Function "Preset multiple Registers", starting on register 0 with a length of 32 registers, trigger mode "Changed Data". These are for sending commands to the sensor control registers.

Click "Apply" and the signal configuration will be as shown in Figure 11

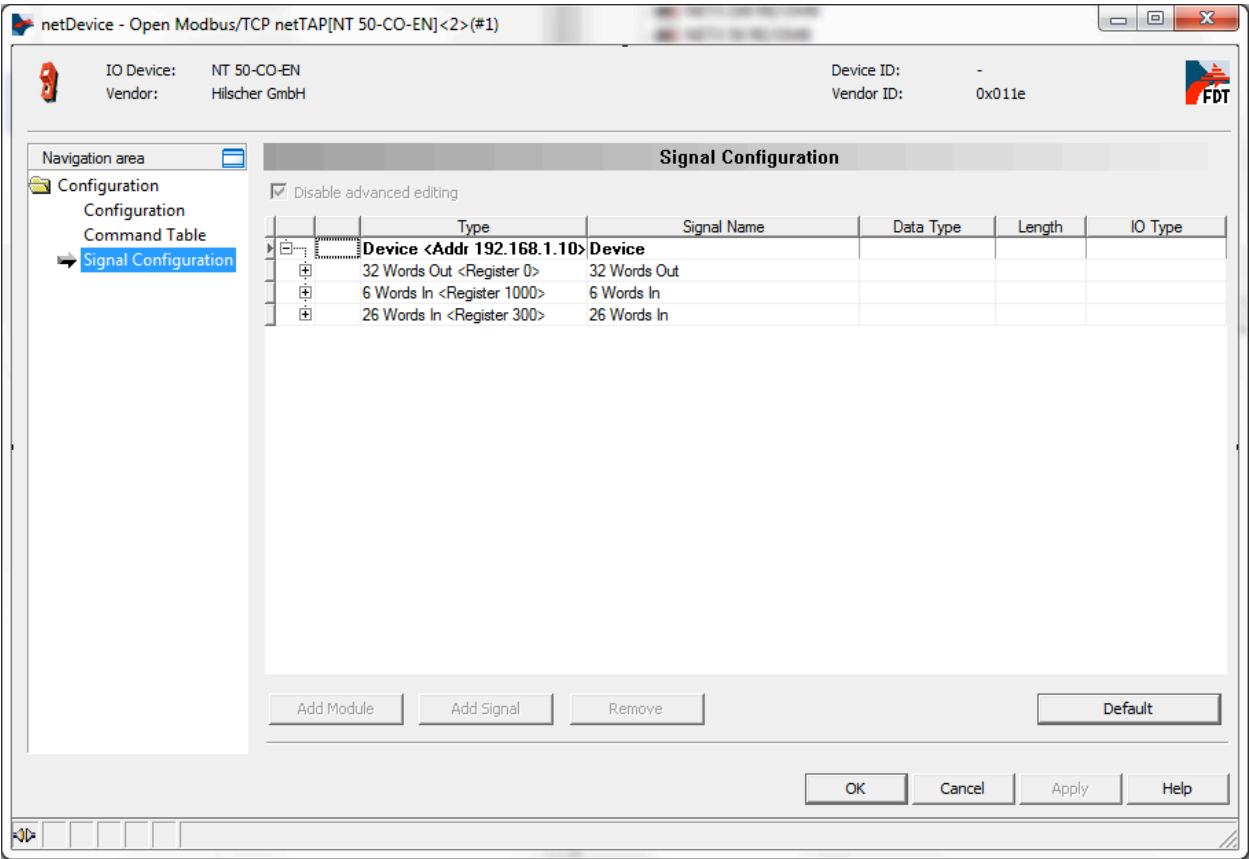


Figure 11: Signal configuration

Click “OK” to close the configuration of Open Modbus/TCP.



Configure the “Signal Mapping” by rightclicking on the netTap and then “Configuration→Gateway”. Enable “Auto Mapping” by setting this to “From Port3 to Port2” and click “Apply”. All the configured modbus registers will be mapped to the Profinet IO Device registers. Now click “OK”.

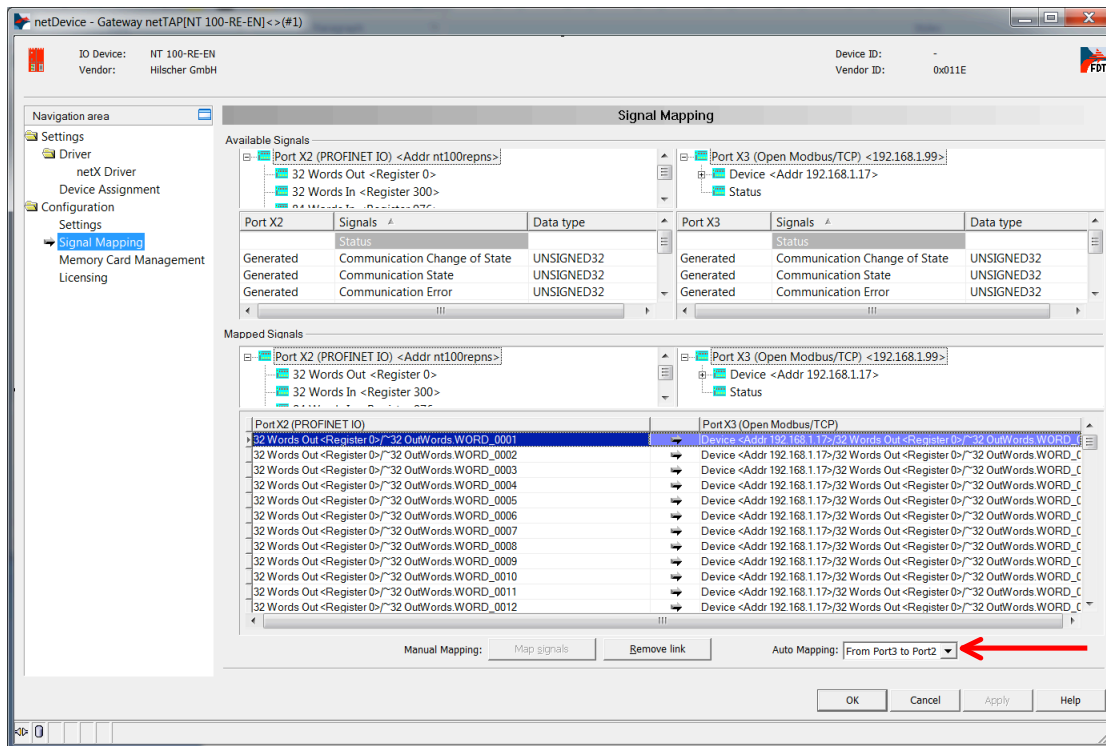


Figure 12: Signal mapping

The configuration of the gateway is now done and could be downloaded to the netTap. To do this, rightclick on the netTap and click “Download”. The netTap is now ready for use.

4.4.2 Configure CANopen

Configure the “CANopen” by right clicking on the netTap and then “CANopen Slave”. We are configuring the netTap to act as a slave device on the CANopen network, and the IFM CR0403 as the CANopen master device.

Ensure Node ID and baud rate are set up to be the same as the configured CANopen slave device in CoDeSys (Refer to Figure 20 - Hilscher NT-50 CoDeSys configuration)

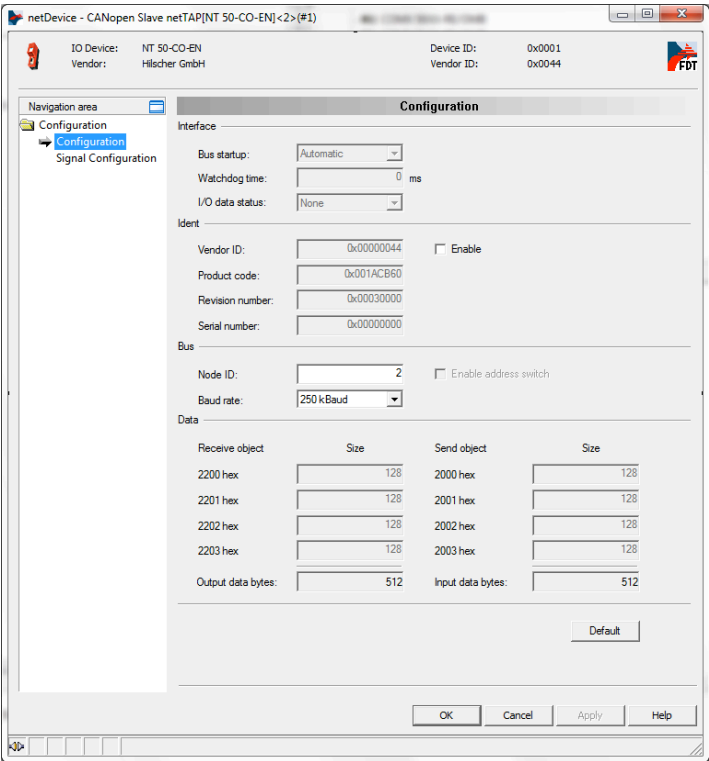


Figure 13 - CANopen slave configuration

Signal Configuration reflects the mapped signals.

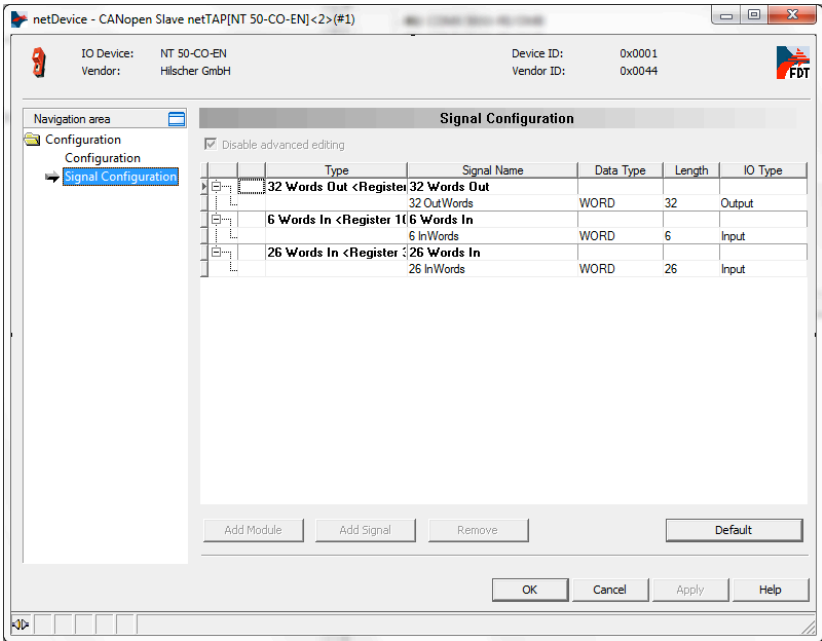


Figure 14 - CANopen slave signal configuration



To download the completed configuration project file to the gateway, right click on the gateway and click Download. After download is complete, the gateway will have to be power cycled. Keep in mind the IP address has to be configured again every time the gateway power cycles (Refer to Figure 3 - Ethernet Device Configuration for Gateway).

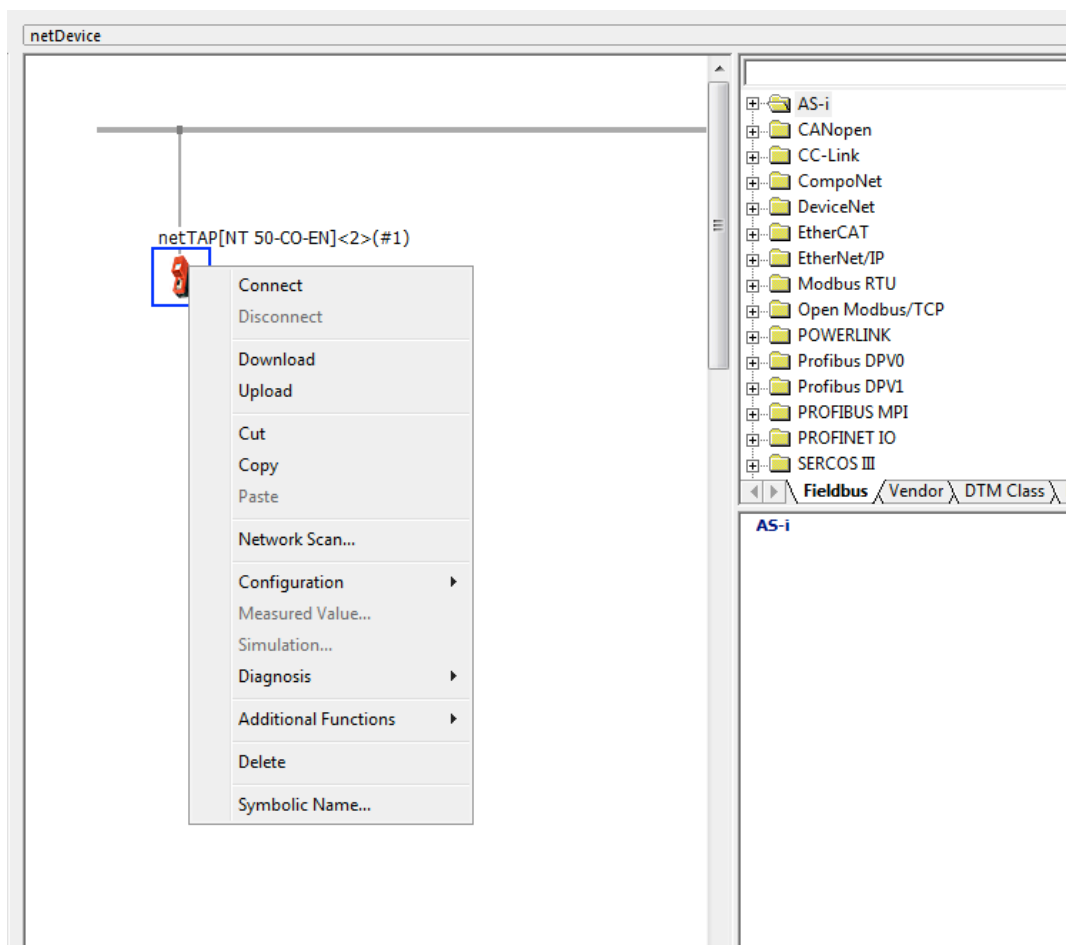


Figure 15 - Download to netTap

After the gateway has been properly configured, gateway can be connected to monitor various diagnostic data.

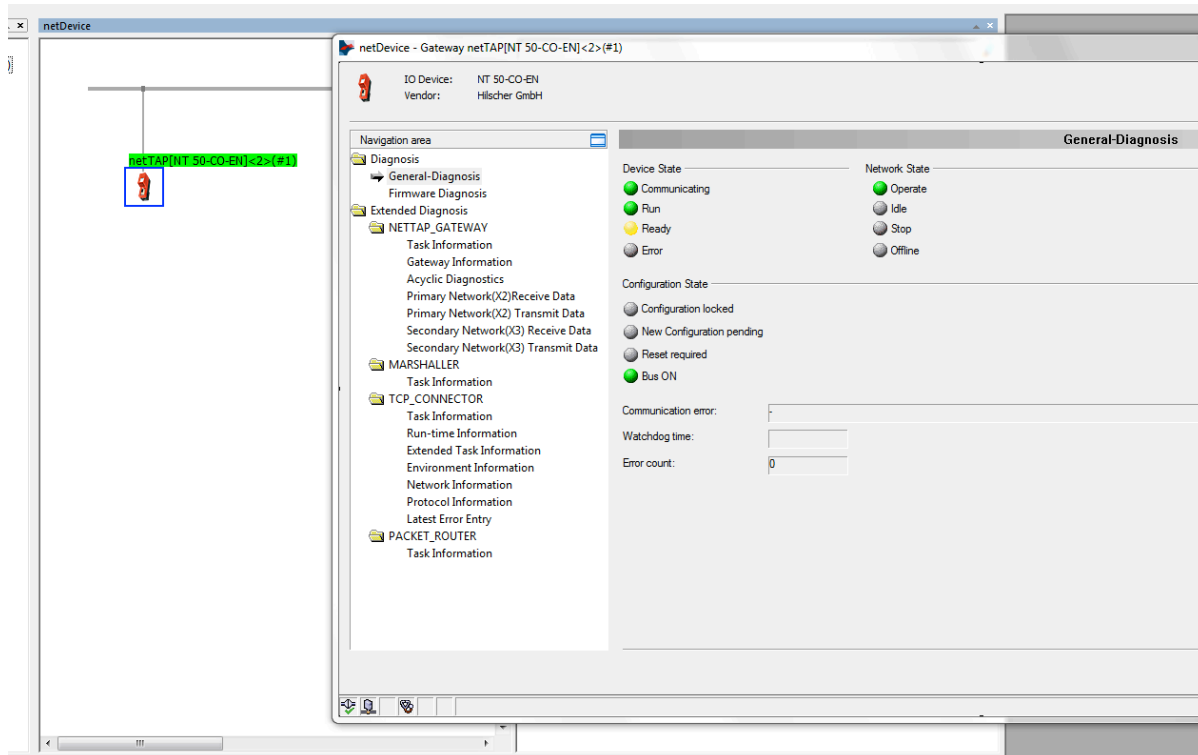


Figure 16 - Connect to netTap

5 Configuring IFM CR0403 BasicController

5.1 Setup the programming system via templates

IFM offers ready-to-use templates, by means of which the programming system can be set up quickly and completely.

Create a new project from existing template in CoDeSys **File -> New from template...**

lfm_Template_CR0403Master_V03xxxx_V02.pro is a good template to configure the CR0403 as a CANopen master.

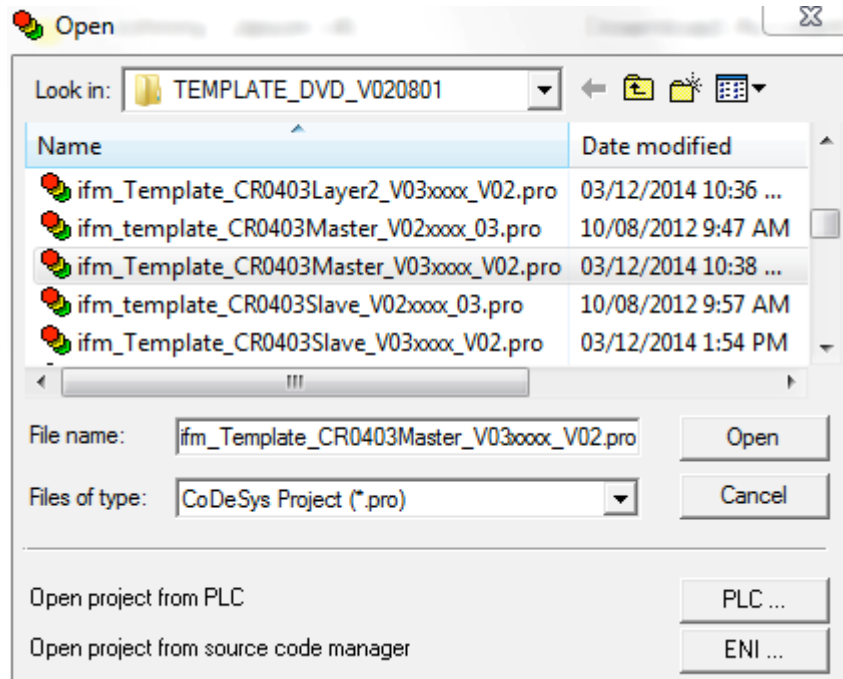


Figure 17 - New CoDeSys project from template

5.2 Configuring CANopen Master Parameters

Ensure the baud rate is configured to be the same as the Hilscher netTap gateway; we have the baud rate set as 250000.

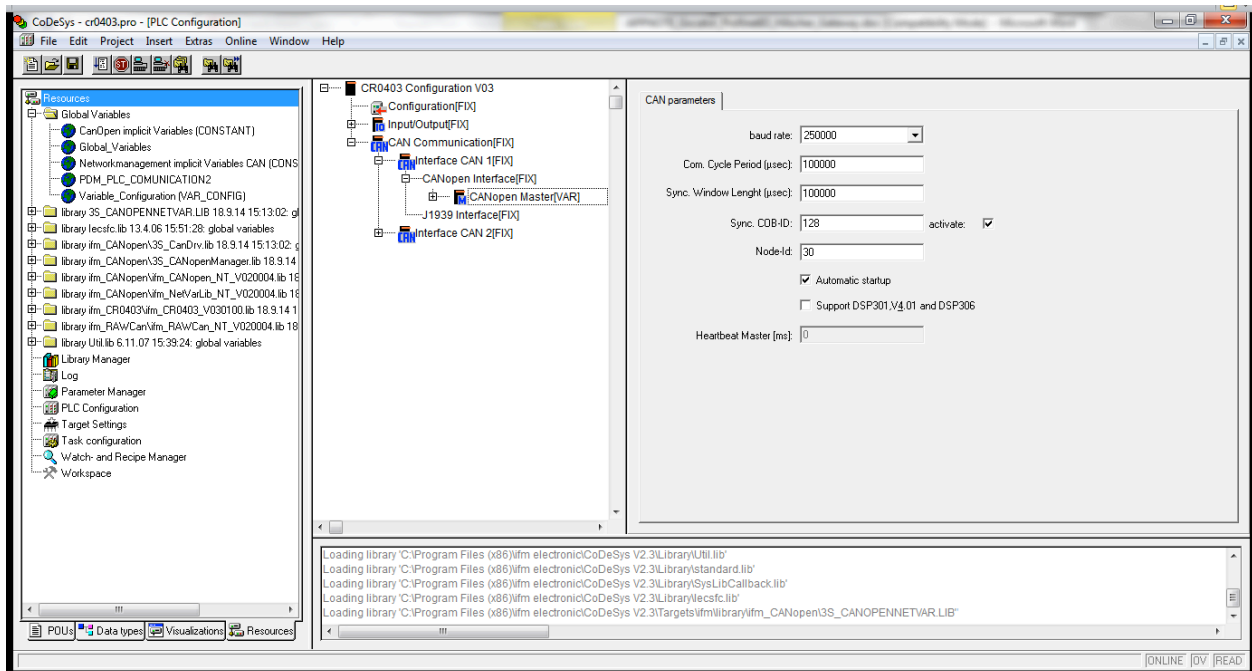


Figure 18 - CANopen Master Parameters



5.3 Add Hilscher NT-50 Gateway EDS

EDS files can be added to CoDeSys through **Extras -> Add configuration file...**

The Gateway Solutions DVD included with the NT-50 gateway contains EDS files. The EDS file to add is “NT50_CO_COS.eds”

After the EDS files have been added, the Hilscher netTap gateway can be added to corresponding CAN interface. In this case, the Hilscher netTap is connected to Interface CAN1

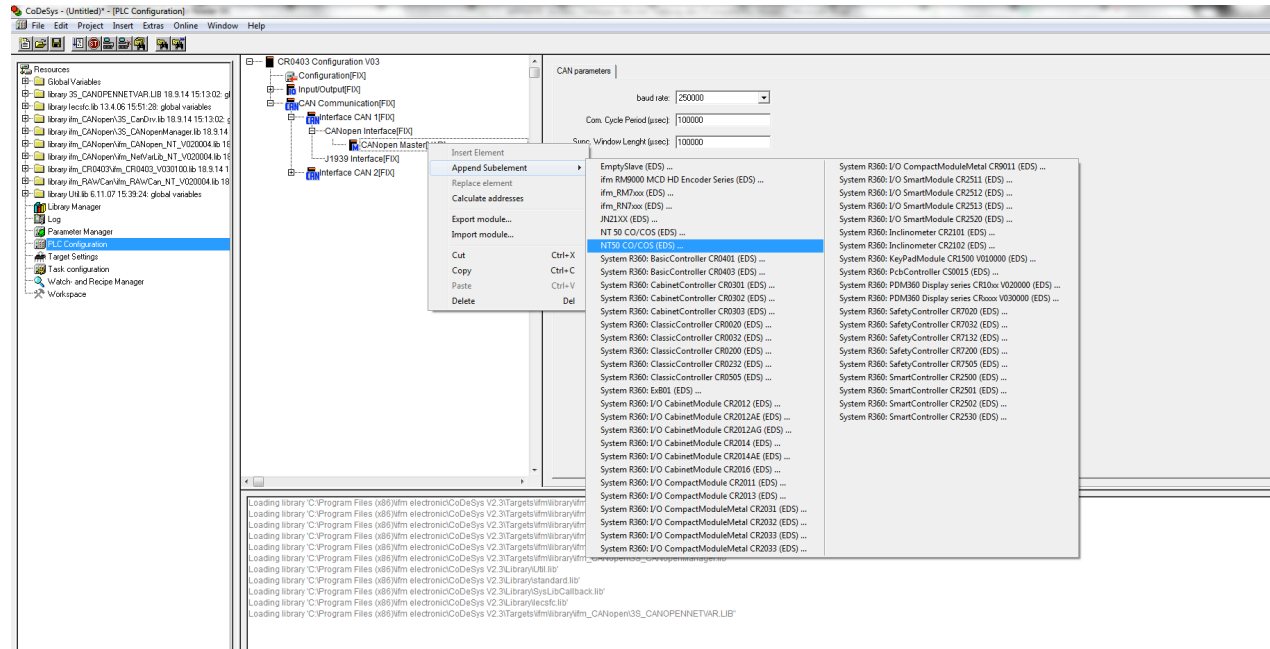


Figure 19 - Adding Hilscher NT-50 EDS

5.4 Configuring Hilscher NT-50 Gateway Parameters

Node ID in the CAN parameters must be configured to be the same as the Node ID configured in the CANopen slave configuration in SYCON.net (Refer to Figure 13 - CANopen slave configuration).

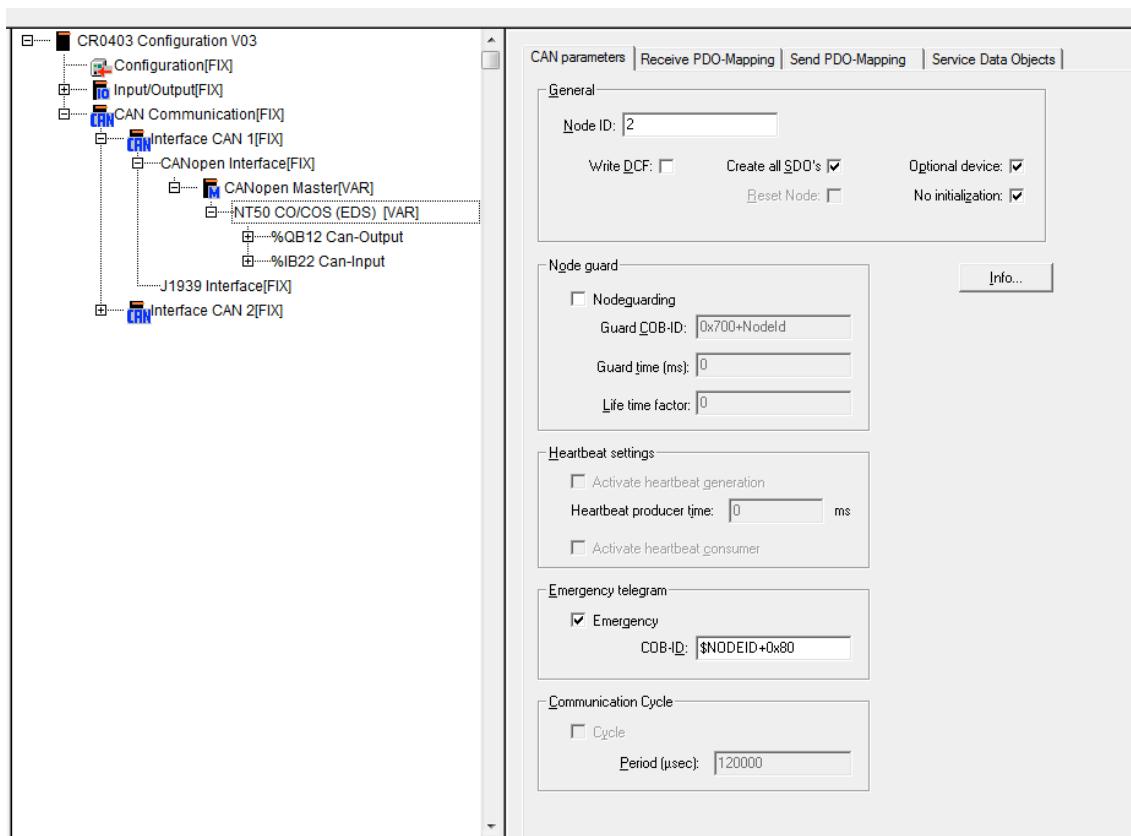


Figure 20 - Hilscher NT-50 CoDeSys configuration

NOTE: The default PDO mapping included in the NT-50 CO/COS EDS is incorrect. Number of PDOs must be limited to 4 PDOs of 8 bytes each for each Receive PDO and Send PDO.

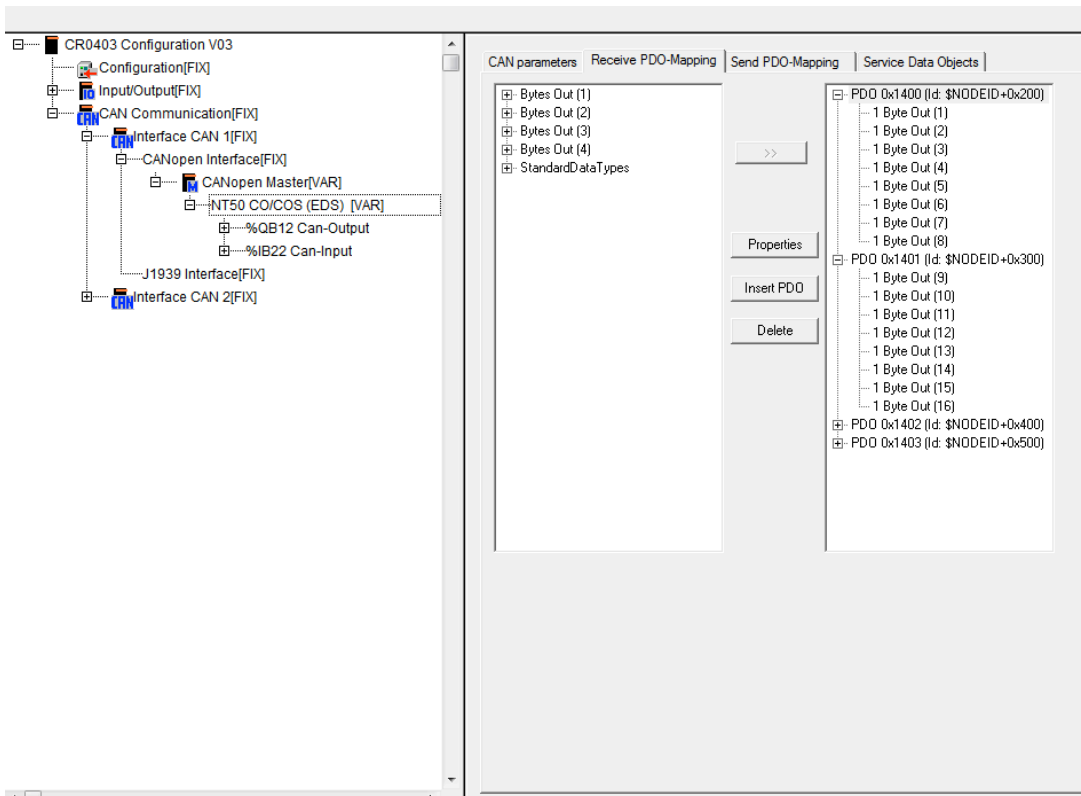


Figure 21 - Receive PDO Mapping

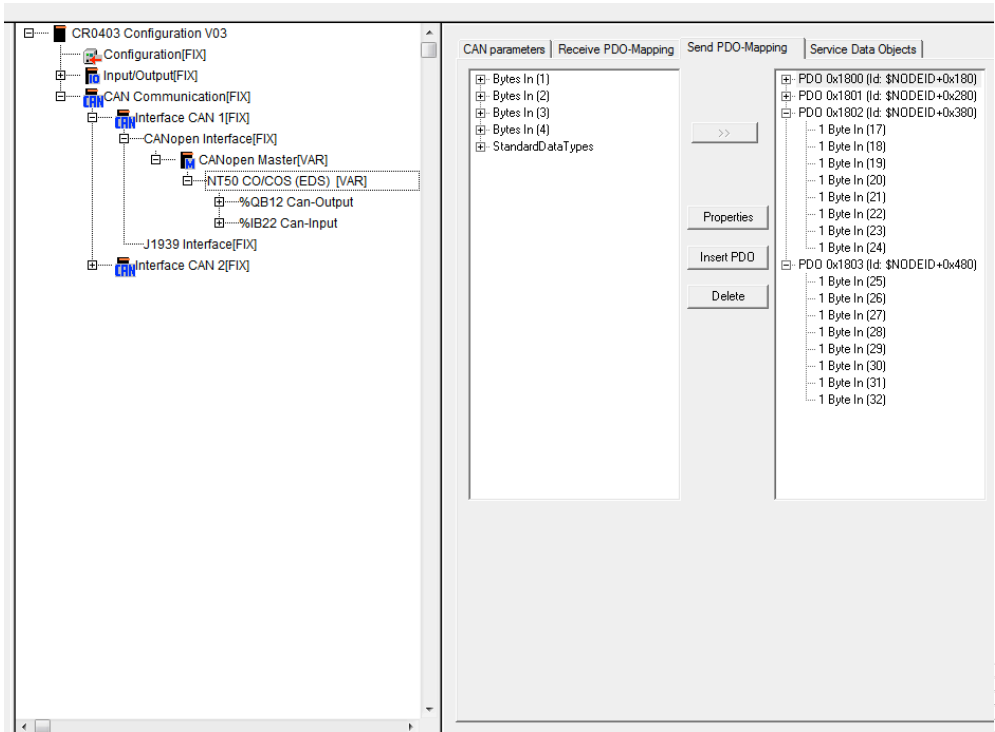


Figure 22 - Send PDO Mapping



The pre-configured Service Data Objects from the EDS corresponds to the settings on the NT-50 gateway and can be left unchanged.

Index	Name	Value	Type	Default
2200sub1	1 Byte Out (1)	0x0	Unsigned8	0x0
2200sub2	1 Byte Out (2)	0x0	Unsigned8	0x0
2200sub3	1 Byte Out (3)	0x0	Unsigned8	0x0
2200sub4	1 Byte Out (4)	0x0	Unsigned8	0x0
2200sub5	1 Byte Out (5)	0x0	Unsigned8	0x0
2200sub6	1 Byte Out (6)	0x0	Unsigned8	0x0
2200sub7	1 Byte Out (7)	0x0	Unsigned8	0x0
2200sub8	1 Byte Out (8)	0x0	Unsigned8	0x0
2200sub9	1 Byte Out (9)	0x0	Unsigned8	0x0
2200suba	1 Byte Out (10)	0x0	Unsigned8	0x0
2200subb	1 Byte Out (11)	0x0	Unsigned8	0x0
2200subc	1 Byte Out (12)	0x0	Unsigned8	0x0
2200subd	1 Byte Out (13)	0x0	Unsigned8	0x0
2200sube	1 Byte Out (14)	0x0	Unsigned8	0x0
2200subf	1 Byte Out (15)	0x0	Unsigned8	0x0
2200sub10	1 Byte Out (16)	0x0	Unsigned8	0x0
2200sub11	1 Byte Out (17)	0x0	Unsigned8	0x0
2200sub12	1 Byte Out (18)	0x0	Unsigned8	0x0
2200sub13	1 Byte Out (19)	0x0	Unsigned8	0x0
2200sub14	1 Byte Out (20)	0x0	Unsigned8	0x0
2200sub15	1 Byte Out (21)	0x0	Unsigned8	0x0
2200sub16	1 Byte Out (22)	0x0	Unsigned8	0x0
2200sub17	1 Byte Out (23)	0x0	Unsigned8	0x0
2200sub18	1 Byte Out (24)	0x0	Unsigned8	0x0
2200sub19	1 Byte Out (25)	0x0	Unsigned8	0x0
2200sub1a	1 Byte Out (26)	0x0	Unsigned8	0x0
2200sub1b	1 Byte Out (27)	0x0	Unsigned8	0x0
2200sub1c	1 Byte Out (28)	0x0	Unsigned8	0x0
2200sub1d	1 Byte Out (29)	0x0	Unsigned8	0x0
2200sub1e	1 Byte Out (30)	0x0	Unsigned8	0x0
2200sub1f	1 Byte Out (31)	0x0	Unsigned8	0x0
2200sub20	1 Byte Out (32)	0x0	Unsigned8	0x0
2200sub21	1 Byte Out (33)	0x0	Unsigned8	0x0
2200sub22	1 Byte Out (34)	0x0	Unsigned8	0x0
2200sub23	1 Byte Out (35)	0x0	Unsigned8	0x0
2200sub24	1 Byte Out (36)	0x0	Unsigned8	0x0
2200sub25	1 Byte Out (37)	0x0	Unsigned8	0x0

Figure 23 - Service Data Objects

After all of the configurations are complete, it can be downloaded to the CR0403 BasicController through **Online -> Login**, CoDeSys will prompt whether new program should be downloaded to the controller if the existing program on the controller is different.

The CR0403 BasicController can be run after logged in **Online -> Run**. PDO data values can be monitored in CANopen Interface from **PLC Resources**.

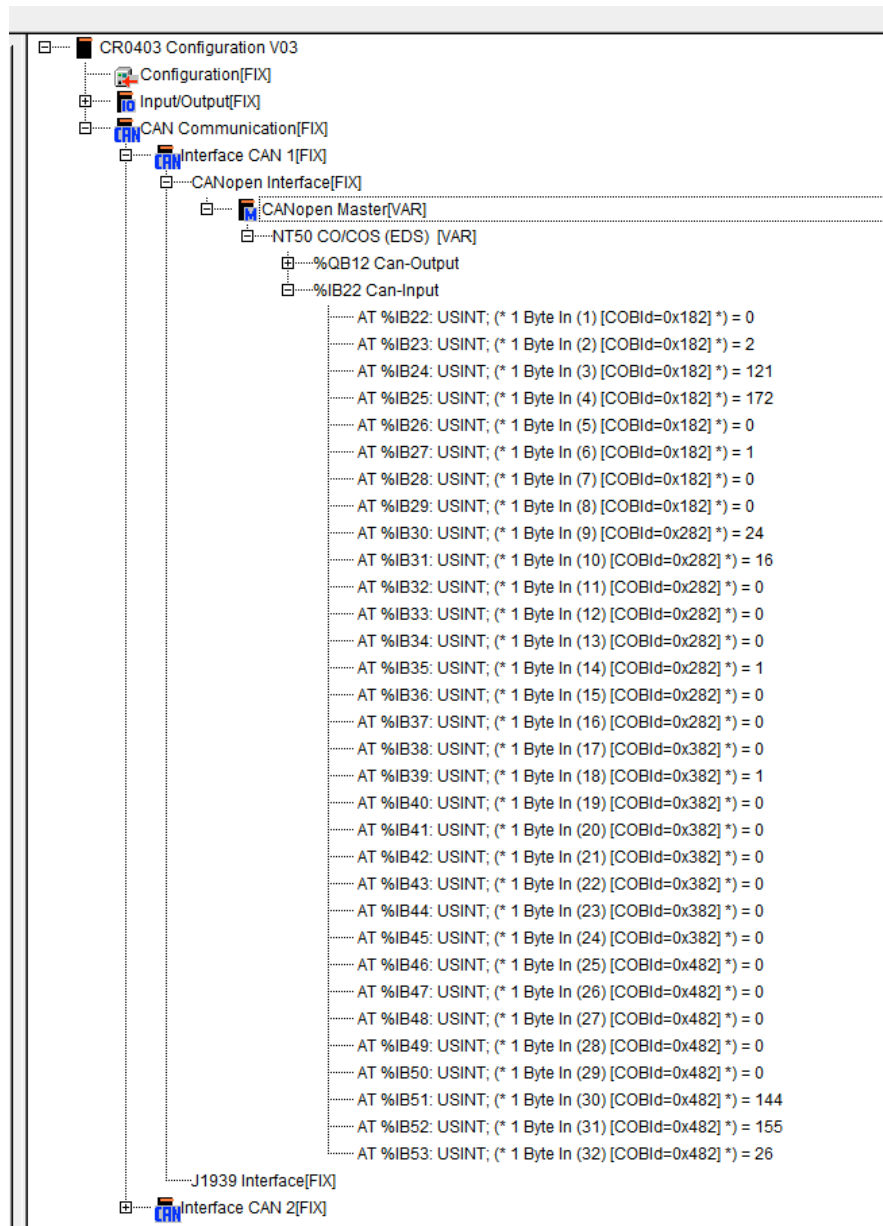


Figure 24 - PDO Data Values

As per the previous Figure 11: Signal configuration, the first 6 words (12 bytes) of the Can-Input correspond to register 1000 to 1005, and following that are 25 words corresponding to Modbus registers 300 to 325.