

IQ3/XNC Modbus Master

Installation, configuration and error handling

Version 2.8
15. mar. 2011

Electrocom
Rødeledsvej 95
DK-5700 Svendborg
Denmark
Tel : +45 8880 7580
www.electrocom.dk



1. Installation

The Modbus Master XNC driver consists of two files:

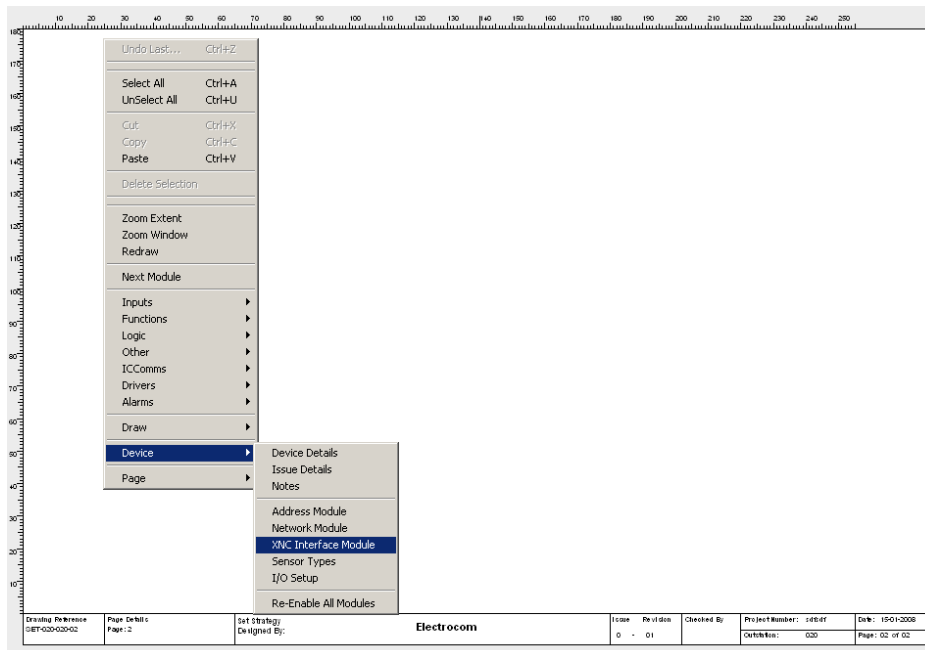
- .xnc file – Application file
- .xnp file – Configuration file

Quick list for the installation:

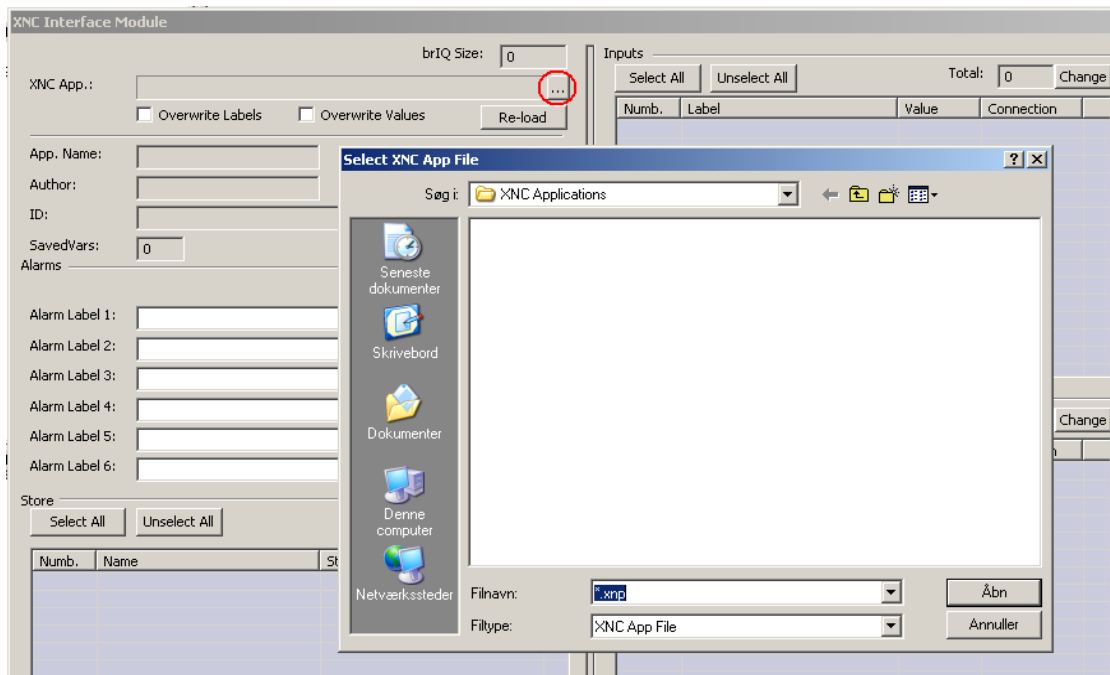
1. Download the strategy with the .xnp file
2. Download the .xnc file

1.1. Download the strategy with the .xnp file

Start up SET (System Engineering Tool) and open your strategy. Right-click on the strategy with the mouse and you will get a menu popping up. In the menu choose “Device”->“XNC Interface Module”.



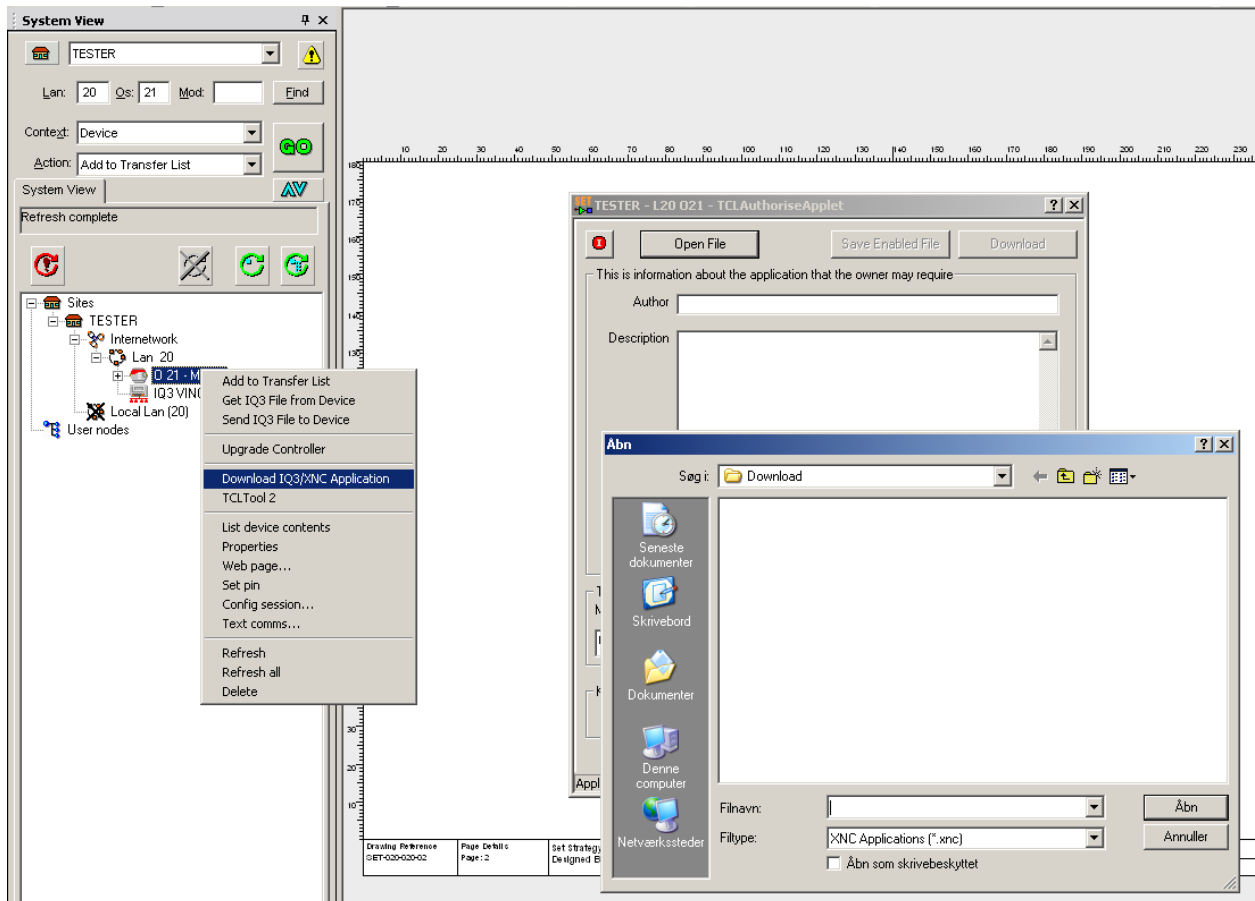
XNC Interface module is opened then press the “...” button which can be seen in the next screenshot marked by a red ring. There will be an open dialog and you can choose the .xnp file.



Now download the strategy to the Trend Controller.

1.2. Download the .xnc file

Switch to “System View” and find the Trend Controller. Right-click on it and choose “Download IQ3/XNC Application”. This will open a new dialog where you press the “Open File” button and an open dialog will appear which is used to find the .xnc file.



Now press the download button and the Modbus Master XNC driver will be installed on the Trend Controller but still needs configuration.

2. Configuration

The Modbus master can be configured according to the Modbus standard. The available communication protocols are TCP/IP, RS232 and RS485. Both ASCII and RTU package formats are supported. Function 1 – 6, 15 and 16 are supported at the moment. The module continuously synchronises registers in Modbus slaves attached to the Modbus master with inputs or outputs in the IQ strategy.

To configure the module use the following steps:

1. Configuration of communication
2. Configuration of slave register to IQ strategy mappings

The Modbus master is configured in the **XNC Interface Module**. String values in the **Store** area defines the configuration. The screen shot below is an example of a configuration of the Modbus master. Predefined configurations (XNP-files) can be selected by clicking “...” next to “XNC App.”.

XNC Interface Module

brIQ Size: 14605

XNC App.: C:\Set Projects\ModbusMaster\XNC Applicat...\ModbusMaster.xnp ...

☐ Overwrite Labels ☐ Overwrite Values

App. Name: Modbus Master - 5 Slaves Version: 2.4

Author: ElectroCom ApS

ID: 0E6B963E-E04F-42AC-8698-B10B-95A3-2832

SavedVars: 0

Alarms

Alarm Label 1:

Alarm Label 2:

Alarm Label 3:

Alarm Label 4:

Alarm Label 5:

Alarm Label 6:

Store

Select All Unselect All Total: 95 Change

Numb.	Name	String Value
1	communication	RS232;9600;8;1;0;ASCII;1;
2	debug_level	0
3	debug_IP	
4	Mapping	1;6;0;1;1;
5	Mapping	
6	Mapping	
7	Mapping	
8	Mapping	
9	Mapping	
10	Mapping	
11	Mapping	
12	Mapping	
13	Mapping	

Inputs

Select All Unselect All Total: 200 Change

Numb.	Label	Value	Connection
1	Temp	0	K1D
2		0	K2D
3		0	K3D
4		0	K4D
5		0	K5D
6		0	K6D
7		0	K7D
8		0	K8D
9		0	K9D
10		0	K10D
11		0	
12		0	
13		0	
14		0	
15		0	

Outputs

Select All Unselect All Total: 200 Change

Numb.	Label	Value	Connection
1		0	S1S
2		0	S2S
3		0	S3S
4		0	S4S
5		0	S5S
6		0	S6S
7		0	S7S
8		0	S8S
9		0	S9S
10		0	S10S
11		0	
12		0	
13		0	
14		0	
15		0	

Info

Drag and Drop items between the 'Connective' and 'Connection' List items to set the Input and Output connections.
Double click on green columns to edit individual labels or values.

OK Cancel

The **String Value** for **communication** specifies how to communicate with the slaves. The **String Values** for **Mapping** specifies how to map the slaves onto the IQ strategy.

The Modbus master can be used in conjunction with the ElectroCom XNC Tool. Configurations for the Modbus master can be made within and debug information from the Modbus master can be viewed from the tool.

In order to use the ElectroCom XNC Tool the Modbus master must know the IP address of the PC that is running the tool. The connection will be made to port 3532. It is recommended that the debug IP address is deleted when no debug is needed since the Trend IQ3/XNC Modbus master otherwise continuously would try to make a connection to that IP address.

- Use the **String Value** for **debug_IP** to specify the IP Address.

The amount of debug information posted to the ElectroCom XNC Tool (debug level) can be selected. Posts with a level lower or equal to the debug level are send.

- Use the **String Value** for **debug_level** to specify the debug level
 - 0 – No debug
 - 1 – Alarms
 - 2 – Warnings
 - 3 – Debug

Input and output labels can be a given arbitrary name in order to make the IQ strategy more readable.

2.1. Configuration of communication

Use the string value of **communication** to configure the protocol and package format. The string contains a number of tokens separated by a ";". The tokens are interpreted according to the desired protocol.

RS232

In order to use this protocol the baud rate, the number of data bits, the number of stop bits, the parity, the package format and the delay in seconds between updates must be specified according to the following list.

- Baud rate: 2400, 4800, 9600, 19200, 38400, 57600 ...
- Data bits: 7 or 8
- Stop bits: 1 or 2
- Parity: 0 (NONE), 1 (ODD) or 2 (EVEN)
- Package format: ASCII or RTU
- Delay: Time in seconds

The configuration string must comply with the format below.

RS232	;	Baud rate	;	Data bits	;	Stop bits	;	Parity	;	Package format	;	Delay	;
-------	---	-----------	---	-----------	---	-----------	---	--------	---	----------------	---	-------	---

Example of a configuration for RS232 using 9600 baud, 8 data bits, 1 stop bit, no parity, ASCII package format and 1 second delay between updates:

RS232;9600;8;1;0;ASCII;1;

RS485

In order to use this protocol the type, the baud rate, the number of data bits, the number of stop bits, the parity, the package format and the delay in seconds between updates must be specified according to the following list.

- Type: RS485 2 wire or RS485 4 wire
- Baud rate: 2400, 4800, 9600, 19200, 38400, 57600 ...
- Data bits: 7 or 8
- Stop bits: 1 or 2
- Parity: 0 (NONE), 1 (ODD) or 2 (EVEN)
- Package format: ASCII or RTU
- Delay: Time in seconds

The configuration string must comply with the format below.

Type	;	Baud rate	;	Data bits	;	Stop bits	;	Parity	;	Package format	;	Delay	;
------	---	-----------	---	-----------	---	-----------	---	--------	---	----------------	---	-------	---

Example of a configuration for RS485 2 wire using 9600 baud, 8 data bits, 1 stop bit, no parity, RTU package format and 1 second delay between updates:

RS485 2 wire;9600;8;1;0;RTU;1;

TCP/IP

In order to use this protocol only the delay in seconds between updates must be specified.

The configuration string must comply with the format below.

TCP/IP	:	Delay	:
--------	---	-------	---

Example of a configuration for TCP/IP and a 2 second delay between updates:

TCP/IP;2;

2.2. Configuration of slave register to IQ strategy mappings

A mapping specifies how registers at a Modbus slave are mapped onto the IQ strategy. A mapping is defined by the IP address (only when using TCP/IP), the address of the slave, the function to use, the starting register address in the slave, the starting number of IQ strategy input/output and the number of registers. A mapping configuration string must comply with the format below.

IP address (only TCP/IP)	;	Slave address	;	Function number	;	Slave register address	;	Starting IQ strategy input/output	;	Number of registers	;
--------------------------------	---	------------------	---	--------------------	---	------------------------------	---	---	---	------------------------	---

Example of a mapping (not TCP/IP) from slave with address 1, using function 2, from register starting at 0, to output starting at 1 and a total of 2 registers are mapped:

1;2;0;1;2;

Example of a mapping (TCP/IP) from slave with address 1, using function 2, from register starting at 0, to output starting at 1 and a total of 2 registers are mapped:

192.168.10.98;1;2;0;1;2;

Mapping strings are specified using the string values for **Mapping**.

The maximum number of allowed slaves are determined by the version of the master. Versions for 1, 5 and 32 slaves are available.

Whether or not the registers are mapped to input or output is dependent on the function. Some functions read from the slave and are therefore mapped to output, and some functions write to the slave and are therefore mapped to input. See the table below or other Modbus documentation (www.modbus.org) for description of the functions:

<i>Function number</i>	<i>Function name</i>	<i>Read/Write</i>	<i>Input/Output at slave</i>	<i>1 bit/16bit at slave</i>
1	Read Coil Status	Read	Output	1 bit
2	Read Input Status	Read	Input	1 bit
3	Read Holding Registers	Read	Output	16 bit
4	Read Input Registers	Read	Input	16 bit
5	Force Single Coil	Write	Output	1 bit
6	Preset Single Register	Write	Output	16 bit
15	Force Multiple Coils	Write	Output	1 bit
16	Force Multiple Registers	Write	Output	16 bit

2.3. Configuration of the Slave registers data format

Modbus Slave units can have values in the Modbus registers presented in a lot of different data formats. This situation only occurs when using Modbus functions 3, 4, 6 and 16.

The affected Modbus registers are Holding and Input Registers. They can contain a value from 0 to 65535. When using different data formats you can have negative, bigger and decimal values.

<i>Data format</i>	<i>Possible values</i>	<i>Size</i>	<i>Description</i>
Unsigned16	0 to 65535	1	This is a normal Modbus register
Unsigned32	0 to 4294967295	2	Adding 2 registers gives a higher value
Signed16	-32767 to 32767	1	Can contain negative values
Signed32	-2147483647 to 2147483647	2	Adding 2 register gives higher negative values
Float32	Decimal values	2	Float can contain decimal and negative values

It is optional to extend a IQ strategy mapping with data format. Example showing Modbus registers read in data format Float32.

1;3;0;1;2;F32L;

There has been added "F32L" to the IQ strategy mapping. The first character tells if the data format is signed, unsigned or float. In this example it is a float, "F". This table will show all the combinations.

<i>Data format</i>	<i>Format</i>	<i>Size</i>	<i>Order</i>	<i>Mapping</i>
Unsigned16	"U"	"16"	"*" *	"U16"
Unsigned32	"U"	"32"	"L" or "B"	"U32L" or "U32B"
Signed16	"S"	"16"	"*" *	"S16"
Signed32	"S"	"32"	"L" or "B"	"S32L" or "S32B"
Float32	"F"	"32"	"L" or "B"	"F32L" or "F32B"

* For Unsigned16 and Signed16 Order do not have any importance and is ignored.

Order is used for those data formats which are bigger than 1 Modbus Register. The order tells which Modbus Register is the least significant part or most significant part. "L" for little part first (least significant part) and "B" for big part first (most significant part).

3. Error handling

When an error occurs a LED will be flashed at the Trend controller. There are defined three different errors:

- No response from slave – Green LED flashes (number 2)
- Error in response from slave – Red LED flashes (number 3)
- LRC/CRC error in package – Yellow LED flashes (number 4)

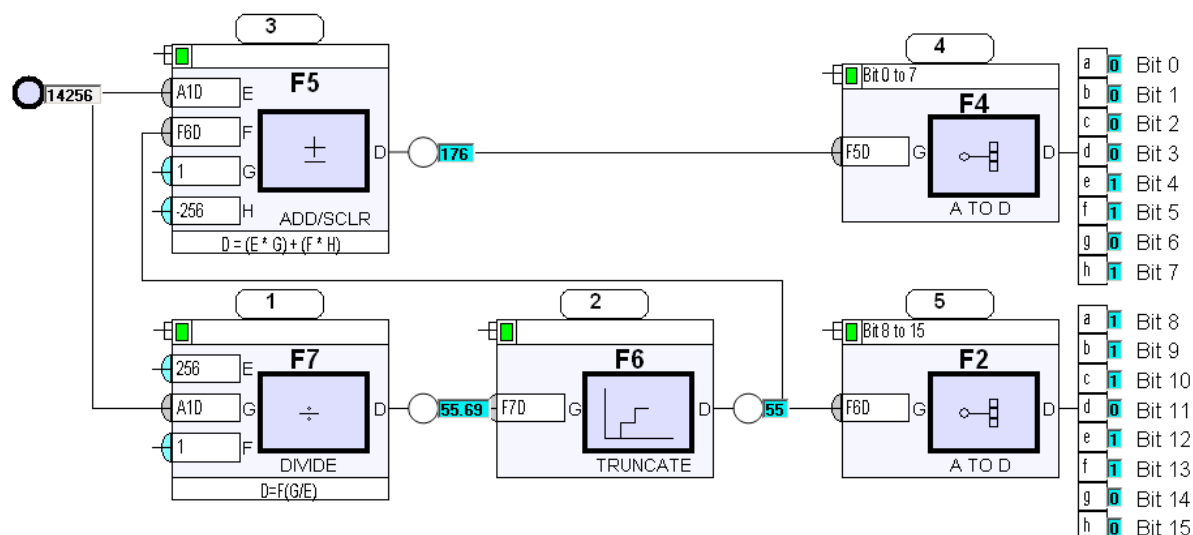
During each pause a yellow LED (number 1) and a green LED (number 2) flashes.

Only on serial versions of the XNC have LEDs.

4. FAQ

FAQ is short for Frequently Asked Questions. This is a compilation of questions and answer which can have the solution for a problem with the XNC Modbus Master.

Q: How to split a value from a Modbus Register to bits in SET strategy?



Q: How to combine several bits to a value for a Modbus Register in SET strategy?

