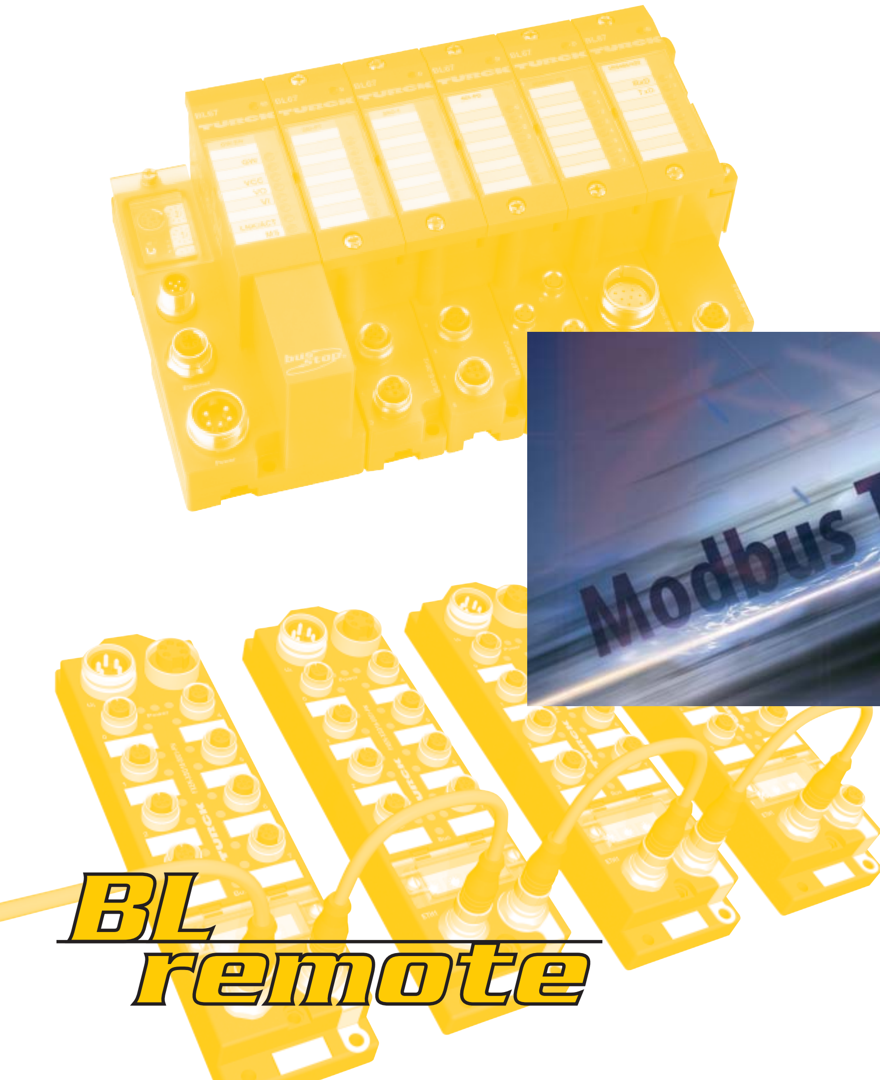
TURCK

Industrial
Automation

DEVICENET™-MASTER USER MANUAL



BL remote

Edition 02/2009

Industrial Automation

Warning!
Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighboring units that are live.
- Follow the engineering instructions of the device concerned.
- Only suitably qualified personnel in accordance with EN 50 110-1/-2 (VDE 0 105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalization. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference do not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60 364-4-41 (VDE 0 100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.
- Emergency stop devices complying with IEC/EN 60 204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency-stop devices must not cause restart.
- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.
- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).
- The electrical installation must be carried out in accordance with the relevant regulations (e. g. with regard to cable cross sections, fuses, PE).
- All work relating to transport, installation, commissioning and maintenance must only be carried out by qualified personnel. (IEC 60 364 and HD 384 and national work safety regulations).
- All shrouds and doors must be kept closed during operation.

Table of Contents

About this Manual

1 The DeviceNet™ master-function

2 DeviceNet™ master with EtherNet/IP

About this Manual

Documentation concept

This manual contains information about the DeviceNet™ master function integrated in TURCK I/O-products.

A DeviceNet™ connector on the device can be used to access a DeviceNet™ subnet with up to 63 manufacturer independent DeviceNet™ nodes (detailed information see chapter 2).

The following chapters contain a short DeviceNet™ master function description and information about the handling of the DeviceNet™ master function for different primary fieldbus systems (EtherNet/IP and Modbus TCP).

All information (hardware, firmware, etc.) about the respective devices and their connection to the primary fieldbus can be found in different device-manuals:

EtherNet/IP

- FXEN - user manual for EtherNet/IP (TURCK-Documentation-No.: English D301155)
- BL67 - user manual for EtherNet/IP (TURCK-Documentation-No.: English D300888)

Modbus TCP

- BL67 user manual for Modbus TCP (TURCK-Documentation-No.: English D300815)

Industrial Automation

Description of symbols used

Warning

This sign can be found next to all notes that indicate a source of hazards. This can refer to danger to personnel or damage to the system (hardware and software) and to the facility.

This sign means for the operator: work with extreme caution.

Attention

This sign can be found next to all notes that indicate a potential hazard.

This can refer to possible danger to personnel and damages to the system (hardware and software) and to the facility.

Note

This sign can be found next to all general notes that supply important information about one or more operating steps. These specific notes are intended to make operation easier and avoid unnecessary work due to incorrect operation.

General information

Attention
Please read this section carefully. Safety aspects cannot be left to chance when dealing with electrical equipment.

This manual contains all necessary information about the prescibed use of the TURCK products with DeviceNet™ master function.
It has been specially conceived for personnel with the necessary qualifications.

Prescribed use

Warning
The devices described in this manual must be used only in applications prescribed in this manual or in the respective technical descriptions, and only with certified components and devices from third party manufacturers.

Appropriate transport, storage, deployment and mounting as well as careful operating and thorough maintenance guarantee the trouble-free and safe operation of these devices.

Notes concerning planning /installation of this product

Warning
All respective safety measures and accident protection guidelines must be considered carefully and without exception.

List of revisions

In comparison to the previous manual edition, the following changes/ revisions have been made:

Tabelle 1: List of revisions

Chapter	Subject	new	changed
all	Additions for the new Modbus TCP-funciton		X
4	DeviceNet™ master with Modbus TCP	X	
5	Application Example: BL67 with Modbus Server Tester	X	

Note
The publication of this manual renders all previous editions invalid.

1 The DeviceNet™ master-function

General information

The TURCK products with DeviceNet™ master function provide a full DeviceNet™ master in addition to their standard primary fieldbus (e. g. EtherNet/IP, Modbus TCP).

The TURCK modules work as gateways between their primary fieldbus and DeviceNet™. Via the DeviceNet™ master, a complete DeviceNet™ network with up to 63 standard DeviceNet™ nodes can thus be connected to the primary fieldbus.
The handling of the DeviceNet™ process data is done in the PLC connected to EtherNet/IP or Modbus TCP for example.

The DeviceNet™ master function is currently implemented in the following TURCK products:

EtherNet/IP

- FXEN (compact I/Os)
  FXEN with EtherNet/IP: FXEN-IM16-0001-IP-DN, page 2-2
- BL67 (modular I/Os)
  BL67 with EtherNet/IP: BL67-GW-EN-IP-DN, page 2-3

Modbus TCP

- BL67 (modular I/Os)
  BL67 with Modbus TCP: BL67-GW-EN-DN, page 4-2

Power supply

General

The DeviceNet™ master on the modules provides CAN_H, CAN_L, Shield, and Ground for the DeviceNet™ communication.

The 24 V DC DeviceNet™ supply voltage (V+/ V-) for the master and the connected DeviceNet™ nodes has to be provided from an external power supply unit.

FXEN

The FXEN station itself is also supplied through the DeviceNet™ cable via the external power supply. No additional power feed at the FXEN module is necessary.

BL67

In BL67, only the DeviceNet™ master is fed through the DeviceNet™ supply voltage (V+/ V-). The modular BL67 station (gateway plus connected I/O modules) has to be supplied additonally through the 7/8’’ power connector at the gateway.

DeviceNet™ subnet

The DeviceNet™ connector on the different devices is used to access a DeviceNet™ subnet with up to 63 vendor independent DeviceNet™ nodes.

Note
The DeviceNet™ master function of the device is only activated if a subnet is connected to the DeviceNet™ connector.

DeviceNet™ connector

Depending on the module type used, the connection of the DeviceNet™ master to the subnet is realized as follows:

FXEN

The FXEN module provides two 7/8“ DeviceNet™ connectors “DN“, one male and one female which are assigned as follows:

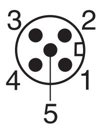
Figure 1: 7/8“-connectors and pin assignment

Pin-No.	
1	Shield
2	V+
3	V-
4	CAN_H
5	CAN_L

BL67

The BL67 gateway provides one male M12 × 1 DeviceNet™ connector “DN“ which is assigned as follows.:

Figure 2: M12-male connector and pin assignment



Pin-No.		
1	Shield	
2	not connected	
3	V-	Ground
4	CAN_H	
5	CAN_L	

Note
Please use an external bus tee or Y-junction for the bus transmission.

Configuration of the DeviceNet™ subnet

DeviceNet™-configuration with SET-button (hardware)

Upon pressing the SET-button at the device, the DeviceNet™ master scans all nodes attached to the DeviceNet™ subnet. The network is then automatically mapped.

All nodes connected to the subnet are read-in with their MAC-ID (DeviceNet™ address), their Vendor ID, their product type and their product code and are stored in the scan list of the DeviceNet™ master.

The I/O data of the connected DeviceNet™ nodes is automatically mapped to the ”Process image” (word alignment) of the master based on the size of the Consumed and Produced Data of each of the nodes.

The CCV is read from the Identity object and stored as the expected configuration consistency value.

BL67 - special scanning behavior

Once the scan of the subnet is complete, the DeviceNet™ master in the BL67-gateway scans the locally connected I/Os mounted right to the gateway and adds their I/O data to the I/O mapping.

Note
Please note, that after every address or baud rate-change a power reset has to be executed.

Attention
In case of a node or network fault, the “Set” button **must not** be pressed. It may remap the I/O data.

DeviceNet™ configuration via IO-ASSISTANT 3 (FDT/DTM)

The configuration of the network can also be done using the software IO-ASSISTANT 3.

The IO-ASSISTANT 3 is a Network Configuration tool on FDT/DTM basis. It helps to create networks offline and online. Configuring a network offline, the network can be checked for validity (i.e., adequate power for the network). Configuring a network online, the network will assume that the master has valid addresses (IP, MAC-ID) and that the connecting devices have unique node addresses. Baud rates for the connecting devices can be set automatically with the „autobaud function“ or set to the desired rate.

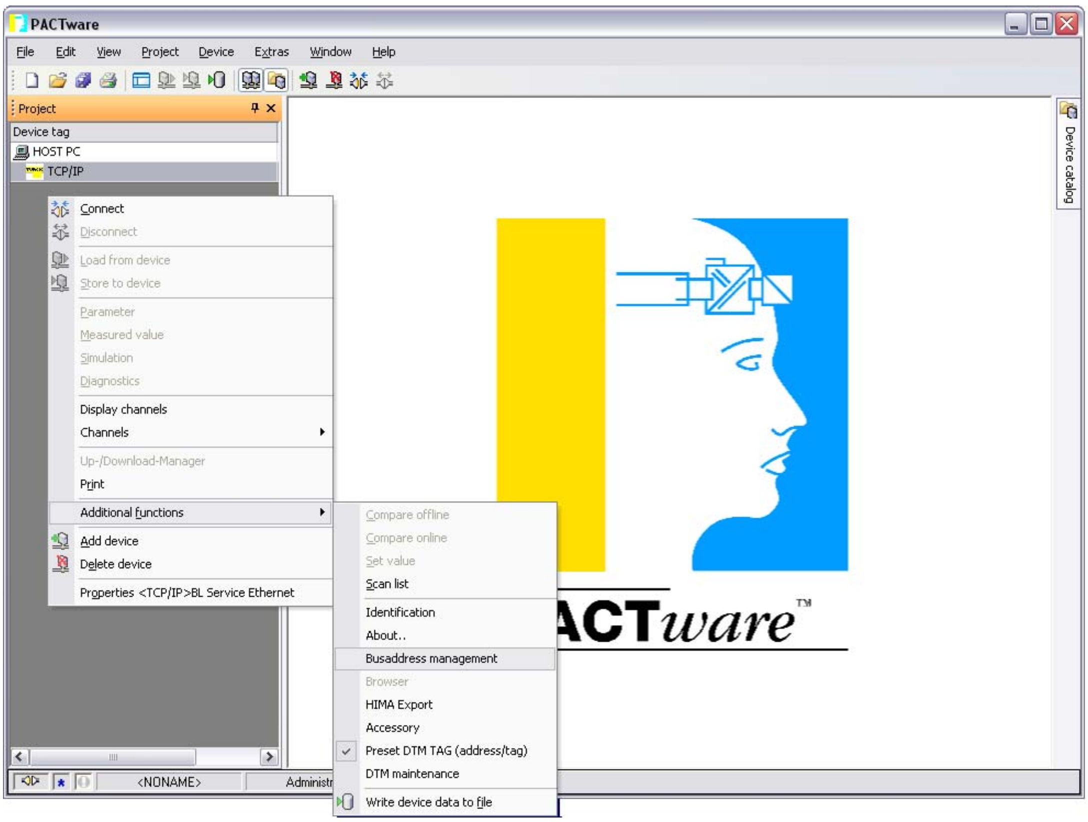
The following example shows the scanning process via IO-ASSISTANT 3 using the FXEN module for EtherNet/IP with DeviceNet™ master (FXEN-IM16-0001-IP-DN). This network is being configured online.

To configure the DeviceNet™ network via the IO-ASSISTANT 3, please carry out the followong steps.

1 Create a new project in PACT*ware*™. If PACT*ware*™ is started for the first time, the software will start with a new project. If not, open create a new project by using „File →New...“.

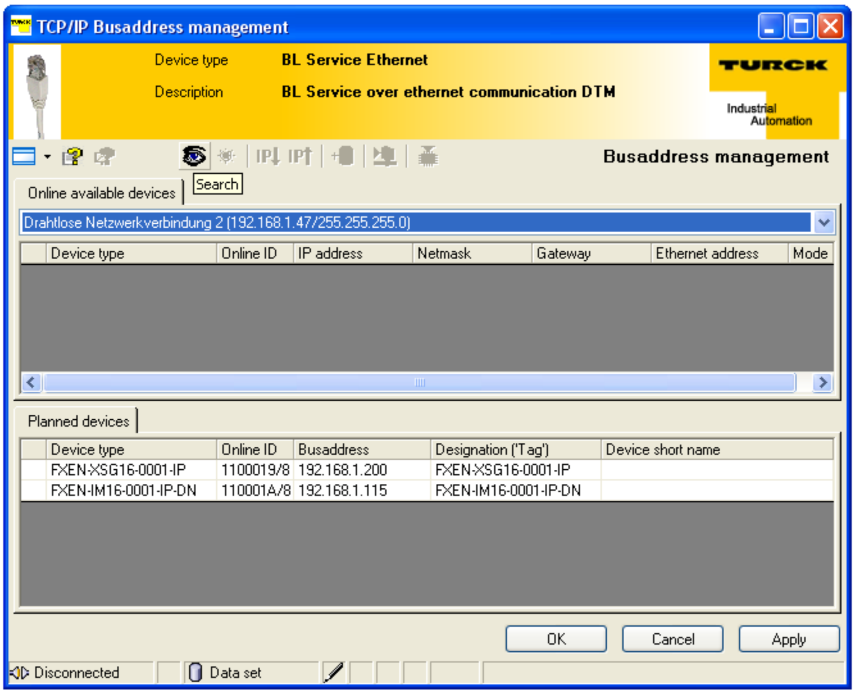
2 Add a TCP/IP-communication DTM “BL Service Ethernet“ to the project and start the Busaddress Management.
To do this, execute a rightclick on the TCP/IP-entry and select „Additional functions → Busaddress management“.

Figure 3: Opening the Busaddress Management



3 Select your network interface and start the network scan using the „search“-button.

Figure 4: Scan function



4 The software scans the Ethernet-Network and all devices which can be found are listed up with their IP-address, their Ethernet-MAC-ID and, if the devices are known to the software, with their device type.

5 Select the module with the DeviceNet™ master, in this example the module FXEN-IM16-0001-IP-DN, and add it to the project using the button „Add Device/DTM to the project“ (see figure below).

6 The software identifies the connected FXEN as a module with DeviceNet™ master and therefore tries to add the connected DeviceNet™ nodes to the station structure.

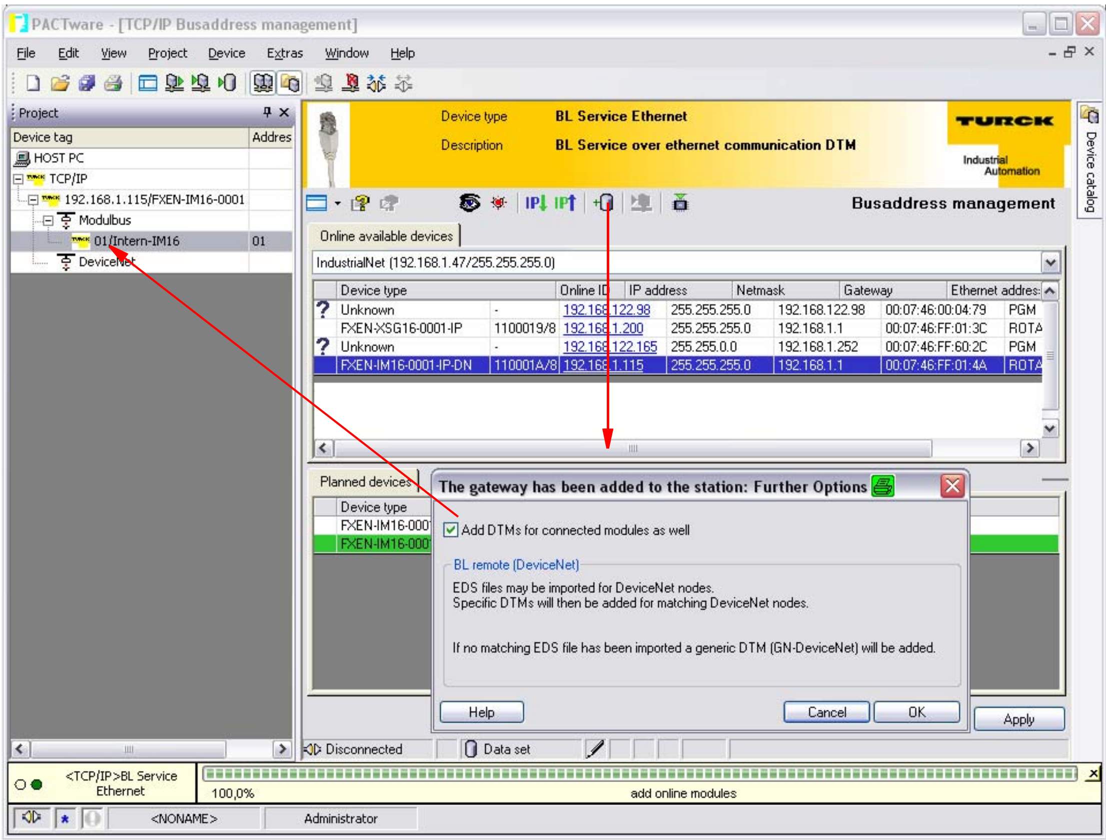
7 In the following dialog box, check “Add DTMs for connected modules” and press ok and the FXEN-module is added to the project structure.

8 If the DeviceNet™ master has already created a scanlist of it’s DeviceNet™ network, all DeviceNet™ nodes stored in this scanlist will also be added to the project.
But:
In this example, this is currently not possible as the DeviceNet™ scanlist in the master has not been created, yet (creating a scan list: see following steps or page 1-5).

Note

For a modular station like BL67, the software would automatically add the whole BL67 station (gateway **and** directly connected BL67 IO-modules) to the project structure.

Figure 5: Adding the device to the project

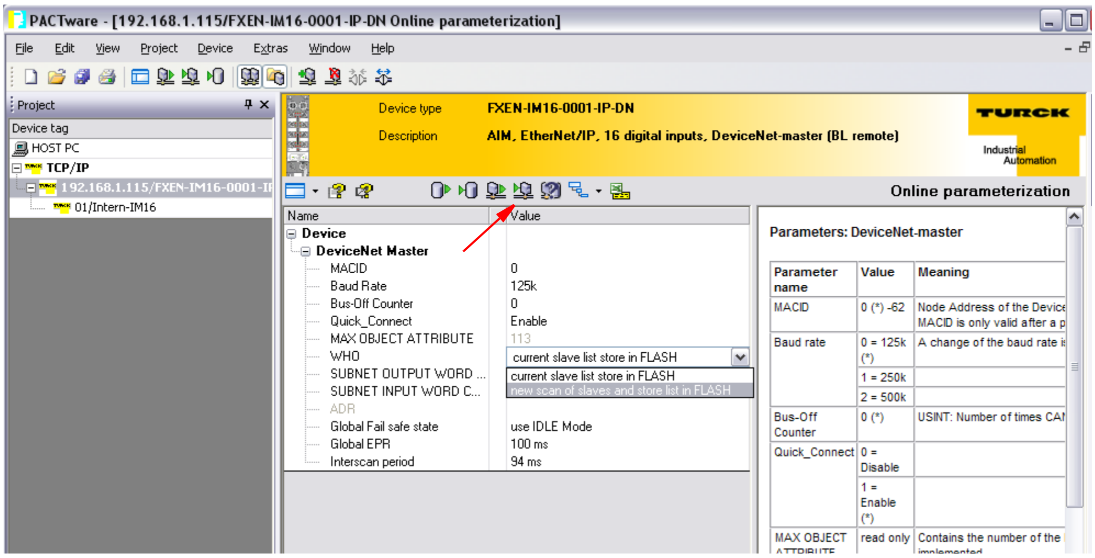


9 Go online with the module: right click on the module → “connect“

10 Open the module’s “Online parameters“: right click on the module → Parameters → Online parameters

11 Set the parameter “Who“ to “New scan of slaves and store list in FLASH“ and send the parameter changes to the device using the button “transmit data to the device“.

Figure 6: Scanning the DeviceNet™



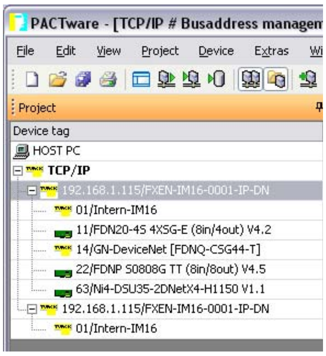
12 The DeviceNet™ master starts the scanning process.

13 A completed and error free scan is indicated by all NET-LEDs at the DeviceNet™ nodes and the DN-LED at the FXEN is constant green. This can take up to 2 minutes.
→ The master stores the scan list.

14 Now, delete the FXEN-module from the project structure and start again with scanning the Ethernet using the „Busaddress Management“ (follow steps 3 to 5).

15 With the scan list stored to the master, the IO-ASSISTANT 3 can readout the list and add the DeviceNet™ nodes to the master.

Figure 7: DeviceNet™ master with connected nodes



The following DeviceNet™ nodes are connected to the FXEN-module:

Tabelle 2: Nodes nodes at connected to the master

Node-Address	Node
11	FDN20-4S4XSG-E DeviceNet™ module with 4 digital input channels and 4 digital channels configurable as in- or output.
14	FDNQ-CSG44-T DeviceNet™ module with 4 digital input channels and 4 digital output channels → The module is shown as Generic (GN)-DeviceNet™ node because it’s EDS-file has not been registered in the I/O-ASSISTANT, yet.
22	FDNP-S0808G-TT DeviceNet™ module with 8 digital input and 8 digital output channels
63	Ni4-DSU35-2DNetX4-H1150 Inductive dual sensor with DeviceNet™ interface.

Address setting for the DeviceNet™ subnet

The MAC-ID (DeviceNet™ address) of all nodes within the DeviceNet™ subnet connected to the DeviceNet™ master of the device are read-in automatically when the subnet is scanned (see also section “Configuration of the DeviceNet™ subnet“). Keep in mind, all nodes must have unique addresses.

MAC-ID of the DeviceNet™ master

- possible range: 0 to 63
- default MAC-ID (DeviceNet™ address) = 0

→ Setting the MAC-ID for the DeviceNet™ subnet can be done

- via IO-ASSISTANT 3 (FDT/DTM),
- via the EtherNet/IP Class DeviceNet™ master class (VSC122, 7Ah)
- via Modbus TCP commands (section “DeviceNet™ master with Modbus TCP“)

Baud rate setting

- Possible baud rate:
  125 Kbit/s, 250 Kbit/s and 500 Kbit/s
- Default setting = 125 Kbit/s

→ Setting the bit rate for the DeviceNet™ subnetcan be done

- via IO-ASSISTANT 3,
- via the EtherNet/IP Class DeviceNet™ master class (VSC122, 7Ah)
- via Modbus TCP commands (section “DeviceNet™ master with Modbus TCP“).

Reset the device to factory settings

If necessary, the complete device including the DeviceNet™ master can be set to its default settings by turning the three rotary switches to position 900 “F_Reset“.

Attention
This reset has effect an **all** parameter settings stored in the device’s FLASH.

FXEN

After a power reset at the device, all parameters in the module’s flash are overwritten.

Attention
Please wait for at least **2 minutes** before setting a new address.

BL67

After a power reset at the device, all parameters in the module’s flash are overwritten. The writing process is indicated by the orange GW LED. The completion of the operation is indicated by the GW LED staying solid green for minimum 20 seconds.

After the reset is done, set the rotary switches back to their original position and execute a power reset again.

Status indicators/ LED behavior

Tabelle 3: LED

LED	Status	Meaning	Remedy
DN	Green	Device is online and communicating. Master is communicating with all registered slaves.	
	Green, flashing	– Auto-discovery in progress.	-
	Red	– Duplicate MAC-ID on the master	– Check the correct assignment of MAC-IDs (DeviceNet™ address). One node has the same MAC-ID as the master.
	Red, flashing	– Empty scanlist (no DeviceNet™ node stored into scan list) – Size errors	– Check the size of the I/O data sent via EtherNet/IP (max. 500 bytes for DeviceNet™ subnet + local I/O)
	Red/green flashing	– DeviceNet™ node missing	– Check the communication to the DeviceNet™ nodes. – Check the Global EPR (Global Expected Packet Rate) and set it to a larger value using the IO-ASSISTANT 3, VSC122 (0×7A), instance 1, Attr. 70h for EtherNet/IP or register 0×308E for Modbus TCP.
MS	OFF	– No supply voltage	–
	Green	– Display of logic connection to Master (1. Ethernet/IP Connection)	–
	Green, flashing	– Ready for operation	–
	Red	– Error	–
	Red, flashing	– DHCP/BOOTP Searching settings	–

Getting started for the DeviceNet™ master

BL67 - start-up behavior

This section assumes that the device is in "Out-of the box state" with all parameters set to default values. It provides a step by step guide on starting up the network.

1. Make sure that the switch position is not 900 (F_Reset).
2. Power-up the device.
3. **DN** LED has to be **OFF**.
4. Setup the MAC-ID (DeviceNet™ address) of the master
   - via IO-ASSISTANT 3 (FDT/DTM)
   - via VSC122 (0×7A), instance 1, Attr. 01h in EtherNet/IP, page 2-11
   - via Modbus TCP register 0x308A.
5. Wait for approx. 1 minute.
6. Make sure that the **GW** LED is **green** for 20 seconds.
7. Power-down the device
8. Power-down all the slaves
9. Turn on the slaves
10. Turn on the device
11. Press and hold the SET-button until the **DN** LED starts **blinking**.

    Depending on the number of slaves in the DeviceNet™ subnet, the time for scanning the subnet will differ. In case of an error free scanning, the **DN** LED will first start **blinking green**, then it will **blink red** and afterwards it will become solid **green**.

    In case of errors during scanning the subnet:

- If the **DN** LED is **red**, duplicate DeviceNet™ MAC-IDs or a Bus-off condition occurred.

  Remedy:
  If a bus-off condition occurred, check the DeviceNet™ subnet for correctly installed bus terminations, check the DeviceNet™ nodes for correct bit rate settings (or check if all nodes are set to autobaud). Power cycle the device and restart this procedure.
- If the **DN** LED is **flashing green/red**, it is possible that the selected Global EPR (Global Expected Packet Rate) is too small. Chose a larger EPR by using either
  - the IO-ASSISTANT 3 (FDT/DTM)
  - VSC122 (0×7A), instance 1, Attr. 70h in EtherNet/IP, page 2-11
  - or Modbus TCP register 0x308E

Note

In case of any of these problems, power cycle the device, modify the master settings and restart this procedure at point 4.

FXEN - start-up behavior

This section assumes that the device is in "Out-of the box state" with all parameters set to default values. It provides a step by step guide on starting up the network.

1. Make sure that the switch position is not 900 (F_Reset).
2. Power-up the device.
3. **DN** LED has to be **OFF**.
4. Setup the MAC-ID (DeviceNet™ address) of the master
   - via IO-ASSISTANT 3 (FDT/DTM)
   - via VSC122 (0×7A), instance 1, Attr. 01h in EtherNet/IP, page 2-11
5. Wait for approx. 1 minute.
6. Power-down the device
7. Power-down all the slaves
8. Turn on the slaves
9. Turn on the device
10. Press and hold the SET-button until the **DN** LED starts **blinking**.

    Depending on the number of slaves in the DeviceNet™ subnet, the time for scanning the subnet will differ. In case of an error free scanning, the **DN** LED will first start **blinking green**, then it will **blink red** and afterwards it will become solid **green**.

    In case of errors during scanning the subnet:

- If the **DN** LED is **red**, duplicate DeviceNet™ MAC-IDs or a Bus-off condition occurred.

  Remedy:
  If a bus-off condition occurred, check the DeviceNet™ subnet for correctly installed bus terminations, check the DeviceNet™ nodes for correct bit rate settings (or check if all nodes are set to autobaud). Power cycle the device and restart this procedure.
- In case of alternate **gree/red** flashing **DN** LED. the selected Global EPR (Global Expected Packet Rate) may be too small. Chose a larger EPR by using either
  - IO-ASSISTANT 3 (FDT/DTM)
  - VSC122 (0×7A), instance 1, Attr. 70h in EtherNet/IP

Note
In case of any of these problems, power cycle the device, modify the master settings and restart this procedure at point 4.

Useful hints:

- Set all nodes in the DeviceNet™ subnet to UCMM, to shorten the time for scanning the DeviceNet™.
- Set all nodes in the DeviceNet™ subnet to autobaud to simplify possible changing of the baud rate.
- Check the DeviceNet™ master alarms to see the current state of the operations (no alarms, Errors, Bus-off, duplicate MAC-ID (DeviceNet™ address) ….).
  - For EtherNet/IP, these alarms can be found in the Vendor Specific Classe VSC122, see section “DeviceNet™ slave class (VSC 123, 7Bh)“, page 2-14.
  - For Modbus TCP these alarms can be found in register 0x3088.

The DeviceNet™ master in the IO-ASSISTANT 3 (FDT/DTM)

The I/O-ASSISTANT 3 is the TURCK project planning software on FDT/DTM basis for configuration, parameterization, set-up support, diagnostics, documentation etc..

The software provides 2 possibilities for accessing the nodes of the DeviceNet™ network.

1 Parameterization via DTMs after installation of EDS-files for the connected nodes

Figure 8: EDS-import

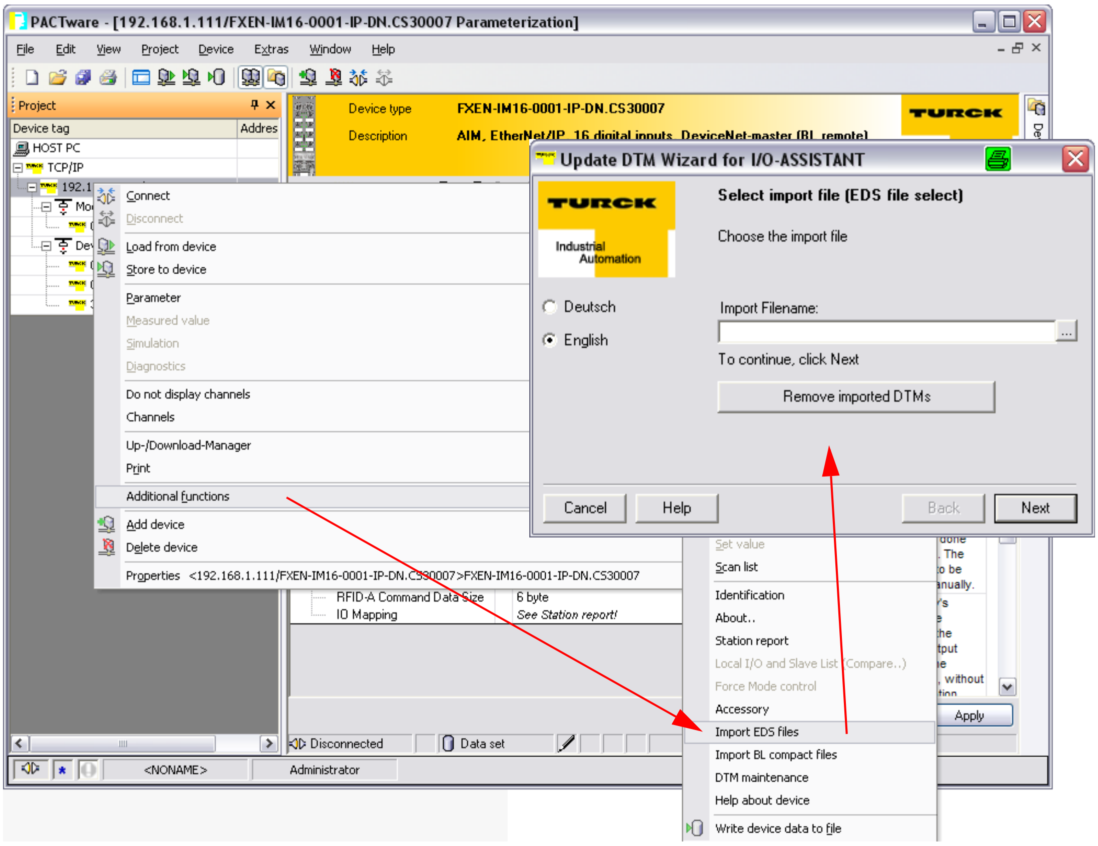
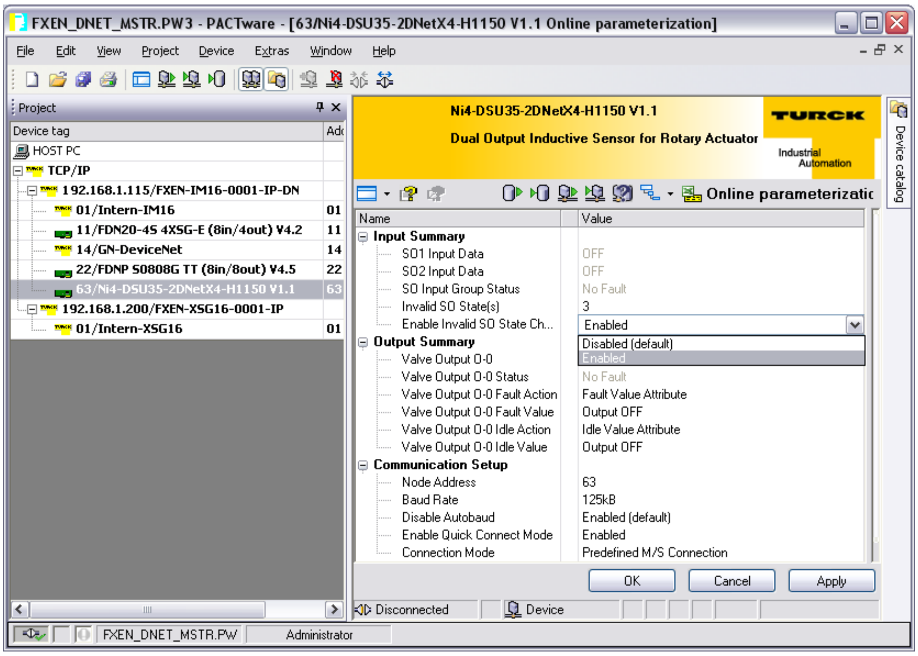
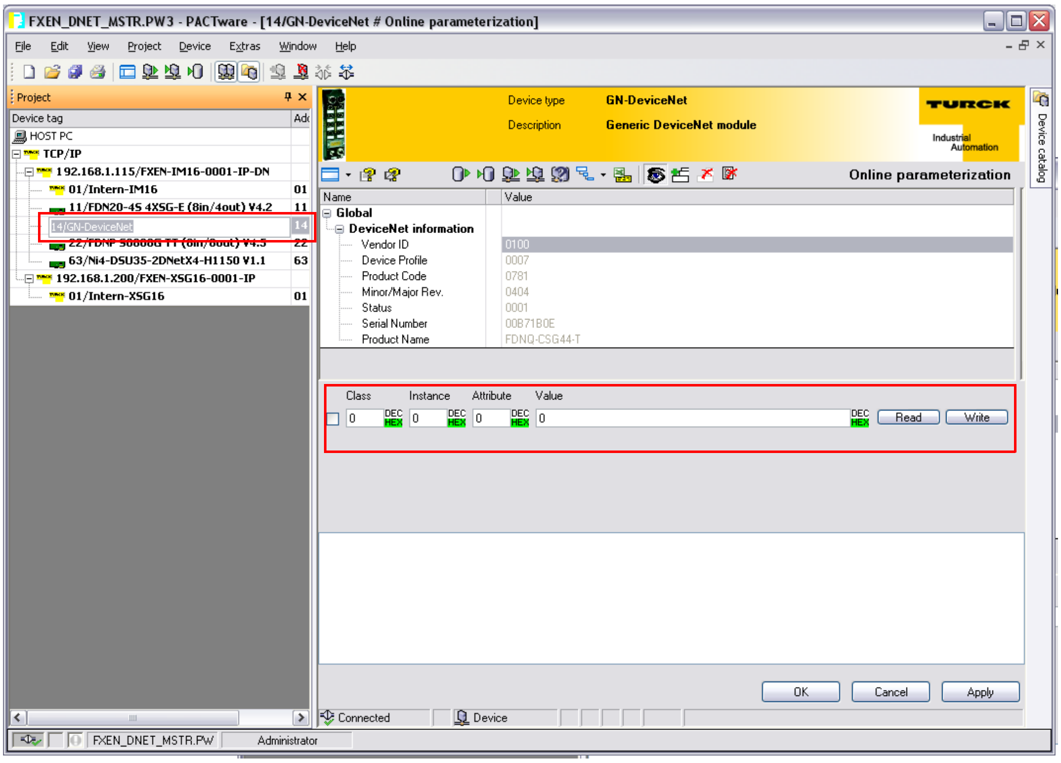


Figure 9: online parameterization via DTM after EDS-import



2 Class Instance Editor for Generic (unknown) DeviceNet™ nodes

Figure 10: Class Instance Editor for the DeviceNet™ master



Note

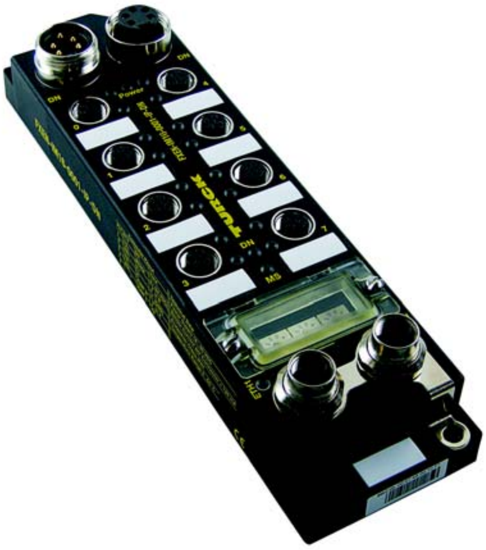
For detailed information about the I/O-ASSISTANT and it's functions, please see the Online Help of the software.
The I/O-ASSISTANT 3 with all TURCK-DTMs as well as the FDT/DTM frame application PACT*ware*™ can be downloaded free of charge from the TURCK homepage www.turck.com.

2 DeviceNet™ master with EtherNet/IP

Modules for EtherNet/IP

FXEN-IM16-0001-IP-DN

Figure 11: FXEN-IM16-0001-IP-DN



The FXEN I/O module with DeviceNet™ master function is part of the FXEN product line. It allows direct connection of 16 inputs to EtherNet/IP and the connection of a complete DeviceNet™ network to the DeviceNet™ master. The process data of the DeviceNet™ subnet are thus handled via EtherNet/IP (see also chapter 1).

The connection to EtherNet/IP is realized via two 4-pole, D-coded M12 x 1-round connectors. The integrated switch allows the creation of a line topology with the I/O modules.

BL67-GW-EN-IP-DN

Figure 12: BL67-GW-EN-IP-DN

The BL67 gateway for EtherNet/IP with DeviceNet™ master function is part of the modular I/O system BL67.

The gateway is used to connect a modular BL67-station with different I/O and technology modules to EtherNet/IP. Additionally, this gateway offers the possibility to connect a complete DeviceNet™ subnet to it and to handle the process data of the DeviceNet™ subnet via EtherNet/IP (see also chapter 1).

The connection to EtherNet/IP is realized via one 4-pole, D-coded M12 x 1-round connector.

Note

The BL67 gateway contains no integrated Ethernet switch!

Process image

The nodes of the DeviceNet™ subnet are mapped into the input image of the EtherNet/IP device.

In the process image, their in- and output data follow the in- and output data of the local I/Os directly placed on the device (e. g. FXEN) or connected to the device (e. g. BL67).

The mapping of the I/O data of the DeviceNet™ subnet is structured according to the nodes' DeviceNet™ MAC-IDs (see chapter 3, "Application example: FXEN at Allen Bradley PLC").

For the explanation of the I/O data mapping the following **example subnet structure** is assumed.

Table 4: Example subnet at DeviceNet™ master

DeviceNet™ MAC-ID	**ModuleD**	**DeviceNet I/O data**	
		Input	**Output**
EtherNet/IP module with DeviceNet™ master		1 word status **A**	1 word control **A**
– local I/O-channels		n byte	n byte
7	DeviceNet™ **node A**	2 byte	2 byte
9	DeviceNet™ **node B**	3 byte	2 byte
25	DeviceNet™ **node C**	1 byte	3 byte
62	DeviceNet™ **node D**	-	4 byte

A *Status and Control byte are mapped into process data (mapping can be deactivated, see VSCs in chapter 3)*

Example: input image

	Byte y	**Byte x**
Word 0	Status word of EtherNet/IP module with DeviceNet™ master	
Word 1	m byte input data of local I/O channels	
...		
Word n		
Word n +1	1 byte input data of subnet node Module **A**, MAC-ID 7	
Word n + 2	1 byte input data of subnet node Module **B**, MAC-ID 9	
Word n + 3	-	Last byte of input data of subnet node Module **B**, MAC-ID 9,
Word n + 4	-	Input data of subnet node Module **C**, MAC-ID 25

Example: output image

	Byte y	Byte x
Word 0	Control word of EtherNet/IP module with DeviceNet™ master	
Word 1	m byte output data of local I/O channels	
...		
Word n		
Word n + 1	2 byte of output data of subnet node Module **A**, MAC-ID 7	
Word n + 2	2 byte of output data of subnet node Module **B**, MAC-ID 9	
Word n + 3	2 byte of output data of subnet node Module **C**, MAC-ID 25	
Word n + 4	-	Last byte of output data of subnet node Module **C**, MAC-ID 25
Word n + 5	4 byte output data of subnet node Module **D**, MAC-ID 62	
Word n + 6		

Status/control words of the DeviceNet™ subnet via I/O data

The DeviceNet™ master provides an additional status information (9 words) and control bytes (1 word) for the DeviceNet™ subnet.

These bytes can be mapped into the process data [enable mapping: VSC122 (7Ah), instance 1, attr. 75h.

Status words

If the 9 status words are mapped into the process data, they are mapped in front of the input data of the DeviceNet™ subnet-nodes, which means they directly follow the input data of the local I/Os at the EtherNet/IP-device (in the example on page 2-4 they would be mapped following byte n).

This status information is structured as follows:

- 1 word for the DeviceNet™ communication (word no. 0)
- 4 words for the “scanlist“ information (word no. 1-4)
- 4 words for “errored nodes“ information (word no. 5-8)

Table 5: Word 0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DNM status	slave missing	empty scan list	CAN error	Dup MacID	subnet input	subnet output	comm. error
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
reserved							

Table 6: Bit meanings

Bit	Meaning
DNM Status	0 = IDLE 1 = RUN
Slave missing	0 = subnet ok 1 = one or more slaves missing
Empty scan list	0 = scanlist ok 1 = The scanlist of the master is empty. No slave has been found during the scan process
CAN error	0 = no error 1 = CAN error (communication problem with CAN controller)
DupMacID	0 = ok 1 = Master DupMacID fault →dupplicate MAC-IDs found in the DeviceNet™ subnode
subnet input	0 = ok 1 = the size of the input data of the subnet is too large (max. number of bytes 500 byte)
subnet output	0 = ok 1 = the size of the output data of the subnet is too large (max. number of bytes 500 byte)
comm. error	0 = no error 1 = communication error or bus off.

The following table represents the scan list of the master:

Each node which has been scanned as being a part of the subnet is indicated by one bit (the order is done by MAC-ID):

0 = no node with this MAC-ID found

1 = node with the MAC-ID found and stored in the master’s scan list

Table 7:								
Word 1	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1	Node 0
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 15	Node 14	Node 13	Node 12	Node 11	Node 10	Node 9	Node 8
...	...							
Word 4	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 55	Node 54	Node 53	Node 52	Node 51	Node 50	Node 49	Node 48
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 63	Node 62	Node 61	Node 60	Node 59	Node 58	Node 57	Node 56

The following bits describe each node status. They show a list of nodes, to which the DeviceNet™ master could not build up a communication:

0 = node present

1 = node not present

Table 8:								
Word 5	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1	Node 0
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 15	Node 14	Node 13	Node 12	Node 11	Node 10	Node 9	Node 8
...	...							
Word 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 55	Node 54	Node 53	Node 52	Node 51	Node 50	Node 49	Node 48
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 63	Node 62	Node 61	Node 60	Node 59	Node 58	Node 57	Node 56

Control word

The control word is used to set the DeviceNet™ master into RUN or IDLE mode.

If the control word is mapped into the process data, it is mapped at the beginning of the output data of the DeviceNet™ subnet-nodes, which means it directly follows the output data of the local I/Os at the EtherNet/IP-device (in the example on page 2-5 they would be mapped following byte n).

This control word is structured as follows:

Table 9: Word 1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved							1=RUN 0=IDLE
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
reserved							

Assembly Instances

TURCK devices with integrated DeviceNet™ master contain two possibilities of process data mapping:

1. Assembly instances **101** and **102**:
   Fixed assembly instances.
   The size of each of those assembly instances is **256** bytes.
2. Assembly instances **103** and **104**:
   Variable assembly sizes. The assembly size is calculated to support the stations I/O-configuration, enabled diagnostics, DeviceNet™ subnet.
   - output assembly instance: **104**
   - input assembly instance. **103**

   The size of each assembly instance can be retrieved through the assembly object and can vary between **2** and **496** bytes.

VSCs for the DeviceNet™ master-function

The VSCs describing the standard EtherNet/IP communication of the devices can be found in the manuals for the Ethernet/IP devices:

- FXEN, document-no.: D301155
- BL67, document-no.: D300888

The manuals can be downloaded from www.turck.com.

DeviceNet™ master class (VSC122, 7Ah)

This class contains parameters and data for the DeviceNet™ master.

Class Instances of the VSCs

Note

Class Instance attributes are the same for each Vendor Specific Class.

Class-specific Object Instances and the corresponding attributes are explained below for the different VSC.

The general VSC - Class Instance attributes are defined as follows:

Table 10: Class instance

Attr. No.	Attribute Name	Get/ Set	Type	Description
100 (0×64)	CLASS REVISION	G	UINT	States the revision number of the class: Maj. Rel. *1000 + Min. Rel.
101 (0×65)	MAX INSTANCE	G	USINT	Contains the number of the highest instance of an object created on this level in the class hierarchy.
102 (0×66)	# OF INSTANCES	G	USINT	Contains the number of Object Instances created in this class.
103 (0×67)	MAX CLASS ATTRIBUTE	G	USINT	Contains the number of the last Class Attribute to be implemented.

Object Instance

Table 11: Object instance

Attr. No. dec. (hex.)	Attribute name	Get/ Set	Type	Description
1	MACID	G/ S	USINT	Node Address, Range 0-63; A change of the master's MACID is only valid after a power-reset 0 = default
2	Baud Rate	G/ S	USINT	0 = 125 kBit/s (default) 1 = 250 kBit/s 2 = 500 kBit/s A change of the baud rate is only valid after a power-reset.
4	Bus-Off Counter	G/ S	USINT	Number of times CAN went the bus-off state
10 (0Ah)	Quick_Connect	G/ S	BOOL	0 = Disable 1 = Enable (Default) ← different to ODVA
100 (64h)	Max object attribute	G	USINT	Contains the number of the last object attribute to be implemented.

Table 11: Object instance

Attr. No. dec. (hex.)	**Attribute name**	**G**et/ **S**et	**Type**	**Description**
101 (65h)	WHO	S	BOOL	0 = current slave list store in FLASH 1 = new scan of slaves and store list in FLASH
102 (66h)	SUBNET OUTPUT WORD COUNT	G	UINT	States the number of output words (consumed words) of the entire subnet
103 (67h)	SUBNET INPUT WORD COUNT	G	UINT	States the number of input words (produced words) of the entire subnet
104 (68h)	OUTPUT	G/ S	ARRAY OF WORD	Contains the output data (consumed data) of the module.
105 (69h)	INPUT	G	ARRAY OF WORD	Contains the input data (produced data) of the module.
106 (6Ah)	Status Array Register of DN Mstr	G	USINT	Bit 0 - Bus Off Bit 1 - Output size too big Bit 2 - Input size too big Bit 3 - Duplicate MAC-ID of the master Bit 4 - DeviceNet™ errors detected Bit 5 - Empty Scanlist Bit 6 - One or more slaves missing Bit 7 – DNM status (RUN=1/IDLE=0)
107 (6Bh)	Present Node	G	ARRAY OF BYTE	Each bit describes one node (8 Byte length)
108 (6Ch)	Node Error	G	ARRAY OF BYTE	Each bit describes one node (8 Byte length)
109 (6Dh)	Slave List	G	ARRAY OF BYTE	Each bit describes one node that should be connected (8 Byte length)
110 (6Eh)	reserved	-	-	-
111 (70h)	Global Fail safe state	G/ S	BYTE	State is activated by loss or termination of Ethernet connection 0 = use IDLE Mode (default) 1 = freeze (all DN outputs are frozen) 2 = force (see VSC 123, attr 113 and 114 for details)
112 (70h)	Global EPR	G/ S	ARRAY OF BYTE	Default = 100 ms Specifies the EPR in ms EPR = 100, setting for 10 slaves and 256 byte process data In case of more than 10 slaves or more than 256 byte process data needed, please increase the Global EPR value.

Table 11: Object instance

Attr. No. dec. (hex.)	Attribute name	Get/ Set	Type	Description
113 (71h)	Interscan period	G	ARRAY OF BYTE	This is a status information which allows the user to know how much margin is left when defining the Global EPR. In order to guarantee error-free communication, please observe that a rest-margin of at least 5 ms should be calculated.
114 -116 (72 h to 74h)	reserved			
117 (75h)	Extended scanner control/ diagnostics	G/ S	WORD	By enabling this parameter – the I/O Data of the DeviceNet-Subnet™ are mapped into the input data of the DeviceNet™ subnet 0 = mapping disabled 1 = mapping enabled

DeviceNet™ slave class (VSC 123, 7Bh)

This class contains parameters and data of each DeviceNet™ slave.

Class Instance (Instance 0)

Note

Please refer to paragraph “Class Instances of the VSCs“, page 2-11, for the description of the class instance for the VSC.

Object Instance 1 to 64 (≅ DeviceNet™ nodes with MAC-ID 0 to 63)

Object Instance **0×01** refers to DeviceNet™ node with MAC-ID **0** (usually this is the master)

Object Instance **0×02** refers to DeviceNet™ node with MAC-ID **1**

...

Object Instance **0×40** refers to DeviceNet™ node with MAC-ID **63**

Table 12: Object instance 1 to 63

Attr. No. dec. (hex.)	**Attribute name**	**G**et/ **S**et	**Type**	**Description**
4	Get Revision Major Revision Minor Revision		STRUCT OF USINT USINT	Revision Number of the connected slave
100 (64h)	Max. object attribute	G	USINT	Contains the number of the last object attribute to be implemented.
101 (65h)	Attribute list	G	ARRAY OF USINT	List of all attributes that are supported by this instance
102 (66h)	Node Address	G	USINT	Node address of the connected slave.
103 (67h)	Vendor	G	UINT	Vendor ID
104 (68h)	Product code	G/ S	UDITN	e.g. device-ID
105 (69h)	Product type	G	UINT	Product Type see also DeviceNet™Class 1, Instance 1, Attribute 2 for details
106 (6Ah)	Product name Length Name	G	STRUCT OF USINT STRING	
107 (6Bh)	Node Mode	G	BOOL	0 = not present 1 = is present and running

Table 12: Object instance 1 to 63

Attr. No. dec. (hex.)	**Attribute name**	**G**et/ **S**et	**Type**	**Description**
108 (6Ch)	Node State	G	ENUM USINT	0 = Slave not in use 1 - 4 = Slave tries to allocate 5 = Slave sets explicit message EPR 6 = Slave activates timer 10 = Slave sets quick-connect 20 = Get VendorID 21 = Get product type 22 = Get product code 23 = Get revision # 24 = Get serial # 25 = Get product name 26 = Get CCV 30 = Get product size 31 = Get cons size 40 = Allocate Poll message 41 = Set Poll EPR 42 = I/O Mode 50 = Re-connect error wait 60 = Close poll message 61 = Close explicit message 80 = Master 90 = Network Found 91 = Network Not Found
109 (6Dh)	Slave output word count	G	UINT	States the number of output words (consumed bytes) of slave
110 (6Eh)	Slave input word count	G	UINT	States the number of input words (produced bytes) of slave
111 (70h)	Output	G/ S	ARRAY OF WORD	Contains the output data (consumed data) of the slave.
112 (70h)	Input	G	ARRAY OF WORD	Contains the input data (produced data) of the slave.
113 (71h)	Fail safe set	G/ S	ARRAY OF WORD	Contains the output data fail safe values for the slave If there was no set default -> 0.
114 (72h)	Slave Fail safe state	G/ S	BYTE	0 = all outputs set to 0 (default) 1 = all outputs set to 1 2 = use attr 113
115 (73h)	Quick_Connect	G	BOOL	0 = Disabled 1 = Enabled
116 (74h)	CCV	G	UINT	Contains the Configuration Consistency Value

3 Application example: FXEN at Allen Bradley PLC

Hard- and software

Hardware

For this application example, the following hardware is used:

- Allen Bradley PLC, ControlLogix5555™, 1756-PA72/B, firmware version V 16.21.12 with:
  - EtherNet/IP Bridge, 1756-ENBT, firmware version V 4.7.19
- FXEN-IM16-0001-IP-DN, 16 digital input channels and DeviceNet™ master with:
  - Addr. 11, FDN20-4S4XSG-E: DeviceNet™ module with 4 digital input channels and 4 digital channels configurable as in- or output
  - Addr. 14, FDNQ-CSG44-T: DeviceNet™ module with 4 digital input and 4 digital output channels
  - Addr. 22, FDNP-S0808G-TT: DeviceNet™ module with 8 digital input and 8 digital output channels
  - Addr. 63, Ni4-DSU35-2DNetX4-H1150 Inductive dual sensor with DeviceNet™ interface

Software

For this application example, the following software is used:

- BootP/DHCP-Server from Rockwell Automation, version 2.3.2.0
- I/O-ASSISTANT 3 from TURCK (PACTware™ and TURCK-DTMs), version 3.5
- RSLogix™5000 from Rockwell Automation, standard edition, version 16.00.
- RSLinx from Rockwell Automation, version 2.43.01

Network configuration and IP-address-setting

Settings of the network interface card

The TURCK modules for EtherNet/IP are delivered with the default IP address 192.168.1.254.

Note

In order to build up the communication between the TURCK product and a PLC/ PC or a network interface card, both devices have to be hosts in the same network.

To achieve this, you have whether:

- to adjust the gateway's IP address via BootP, DHCP etc. for integrating it into your own network
  or
- to change the IP address of the used PC or network interface card

In this example, the network setting of the network card are the following:

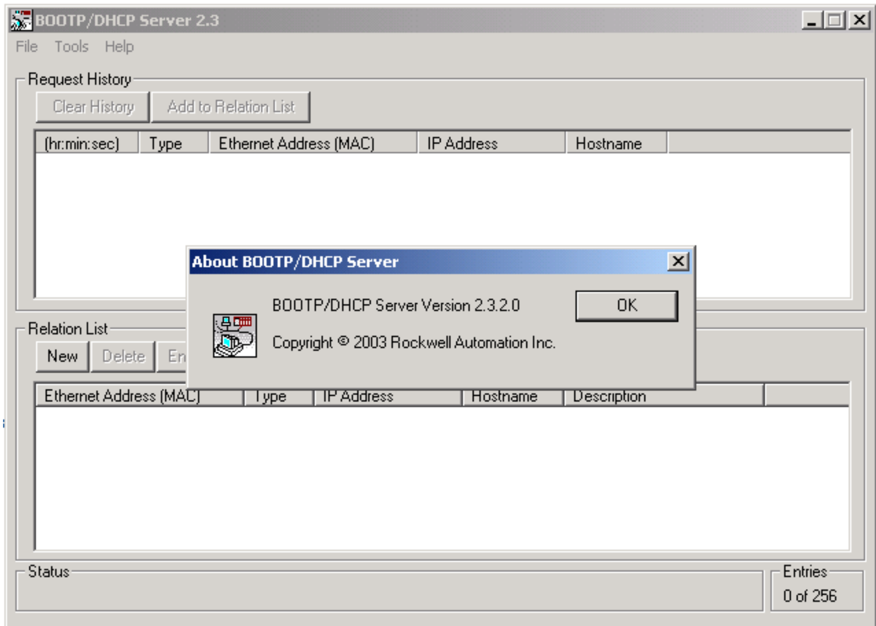
IP address: 192.168.1.47

Subnet mask: 255.255.255.0

Address setting at the FXEN-module via DHCP-mode

In this application example, the IP address is set via DHCP using the software tool "BootP/ DHCP-Server" version 2.3.2.0 from Rockwell Automation.

Figure 13: BootP-Server from Rockwell Automation



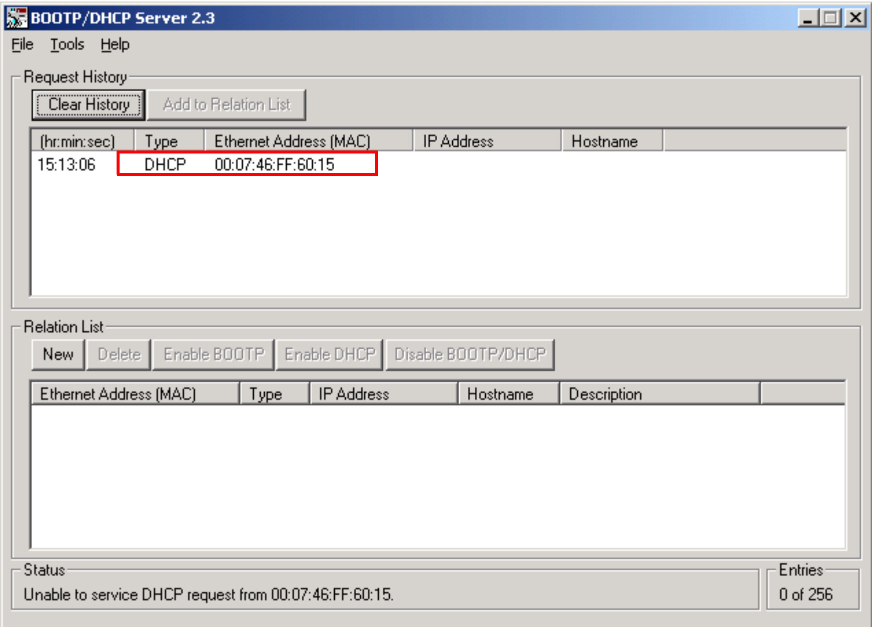
Addresses in the range from 1 to 254 can be allocated in the default subnet 192.168.1. The addresses 0 and 255 are reserved for broadcast messages in the subnet.

Note

The rotary coding switches on the module must be set to "400" or "600" in order to enable the DHCP-Mode or respectively the PGM-DHCP-mode.

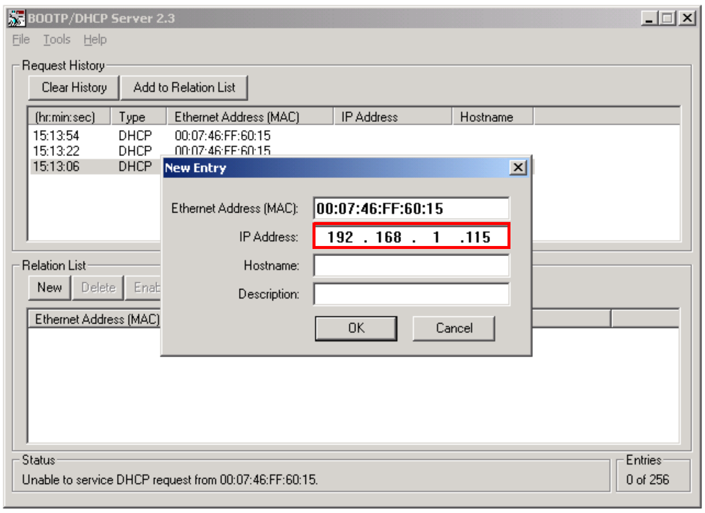
After having been connected to the network, the FXEN-module sends DHCP requests to the server using its MAC-ID.

Figure 14: DHCP-request of FXEN gateway



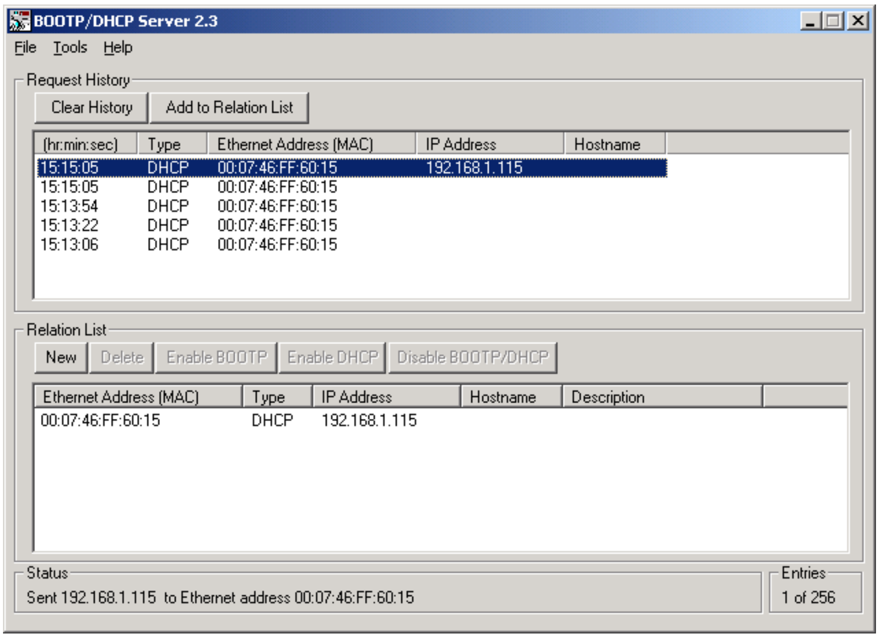
A double click on the request-entry opens the "New Entry" dialog box in which an IP address can be assigned to the module’s MAC-ID.

Figure 15: Setting the IP address via DHCP



The BootP/DHCP-Server sends the IP Address via BootP/DHCP to the FXEN-module and, after a few seconds, the module answers with its new IP address when having stored it.

Figure 16: Setting the IP address via DHCP



The "Relation list" can be stored for further applications. It can serve for permanent assignment of defined IP addresses to MAC-IDs/ modules.

Attention

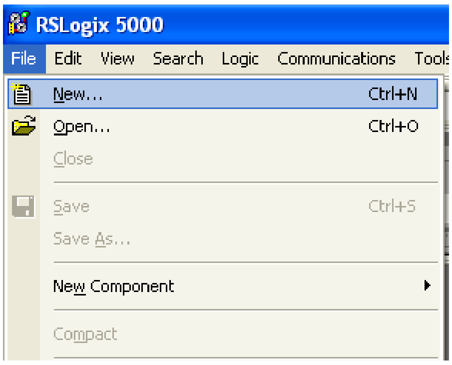
If the BootP/DHCP-server is shut down, the FXEN-module loses the IP address after a power reset!

Configuration of the network in "RSLogix 5000"

The EtherNet/IP hosts (PLC, EtherNet/IP interface, I/O modules) have to be configured using the software "RSLogix 5000" (in this example version 15) from Rockwell Automation.

Start RSLogix and open a new project using the "File" menu.

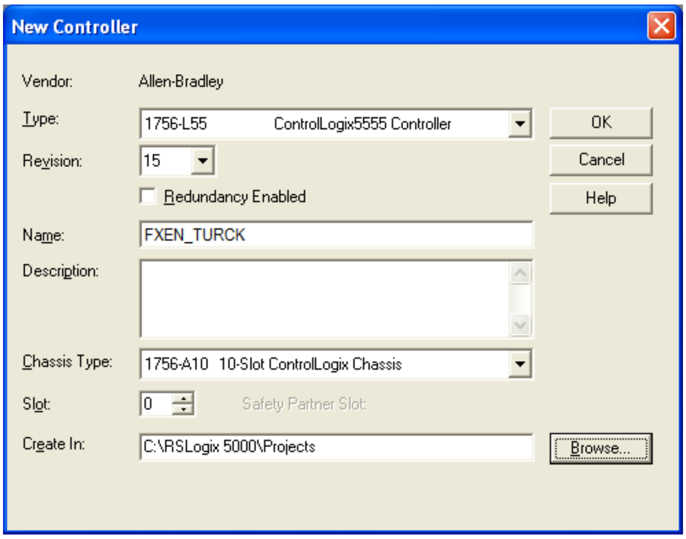
Figure 17: Creating a new project in RSLogix



Configuration of the controller

Enter the information related to the controller depending on your configuration, as well as a name for the controller.

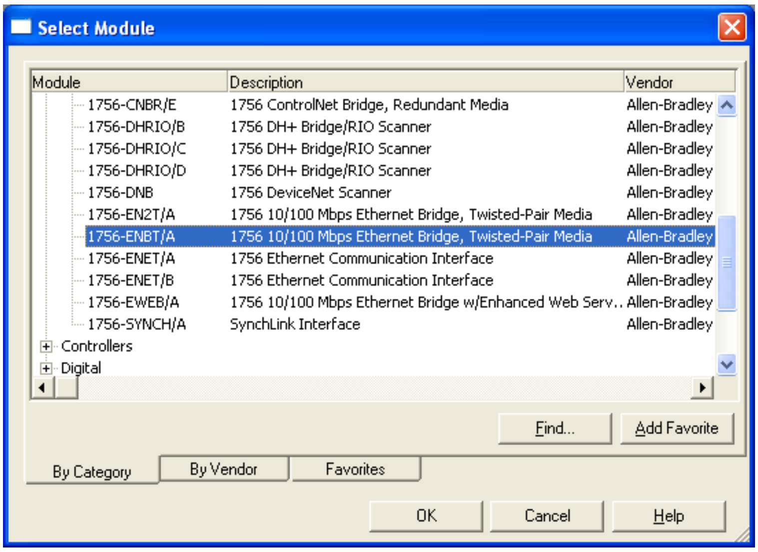
Figure 18: Configuration of the controller



Your project will be opened offline. In order to configure the network, please right-click "I/O Configuration" and select "new Module" to add the first host, the EtherNet/IP bridge, to the network.

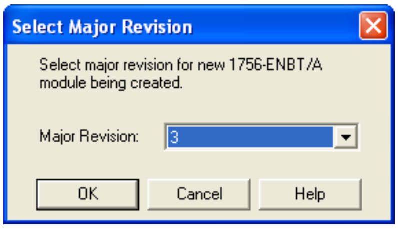
Open "Communications" and select the bridge. In this example this would be 1756-ENBT/A.

Figure 19: Selection of the EtherNet/IP bridge



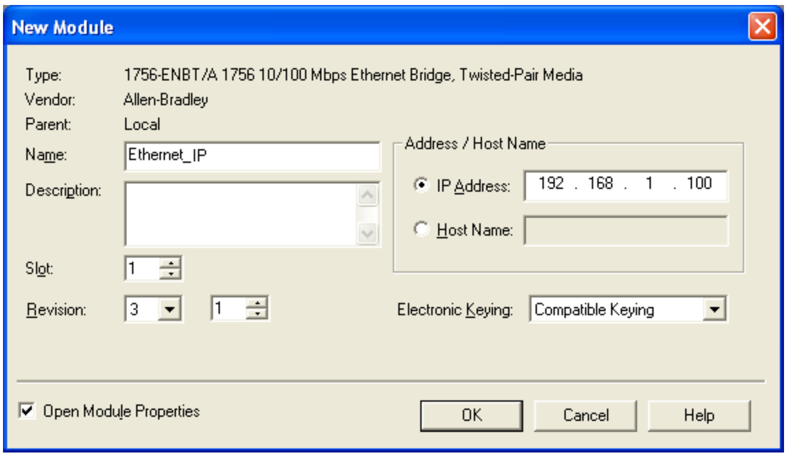
Enter the "Major Revision" of your EtherNet/IP bridge and click "OK".

Figure 20: Major Revision of the EtherNet/IP Bridge



In the following dialog box "New Module" enter the a name for the bridge and define its IP Address (in this example 192.168.1.100).

Figure 21: Configuring the EtherNet/IP Bridge



In the following dialog box "Module Properties: Local..." press "OK". You may also browse offline through the module properties when you click "Next". At this point there is no need for further entry action.

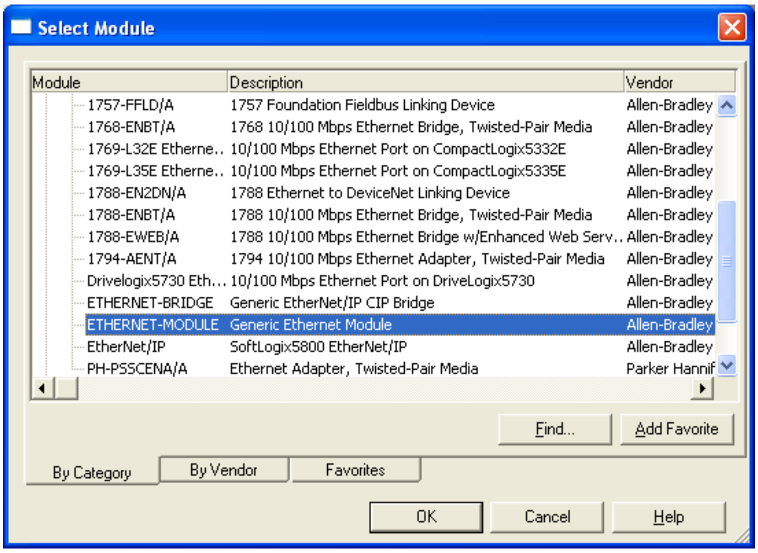
If "Next" is selected, the "Module Properties" window displays information that will be available when the module is online. The configuration of the interface is completed. Press "Finish" to close the dialog box.

Configuration of the FXEN

Add the FXEN to the I/O configuration by using a right-click on the EtherNet/IP bridge module 1756-ENBT/A and select "New Module".

Open "Communications" and select the entry "Generic Ethernet Module" to configure a BL67 gateway.

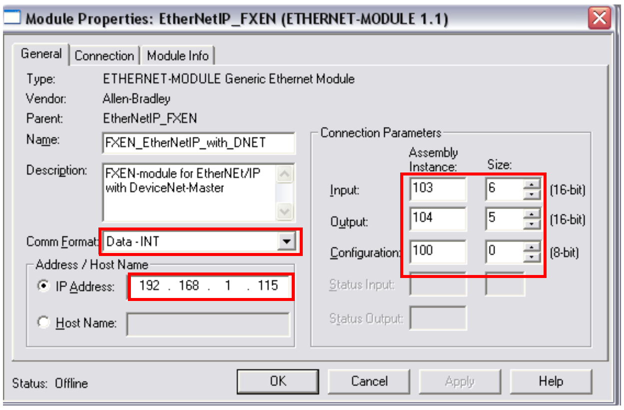
Figure 22: Add generic Ethernet module



Please enter the Connection Parameters for the FXEN as follows.

Assembly Instances 103 and 104 have to contain the exact in- and output size of the FXEN with DeviceNet™ network. In this example this would be:

Figure 23: Configuration of FXEN with DeviceNet™ master and the example network



The exact number of in- and output bytes of an EtherNet/IP-station with DeviceNet™ master and of the connected DeviceNet™ nodes can easily be determined by using the I/O-ASSISTANT. Please read the following section "I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM)", page 3-10.

Note

If the Assembly Instances 101 and 102, the Connection Parameters have to be set to a static in- and output size of 128 words each.

In the "Connection" tab set the "Requested Packet Interval" (RPI) to 10 ms, which normally should be the default setting. For FXEN, the successfully tested RPI range is 5 and higher.

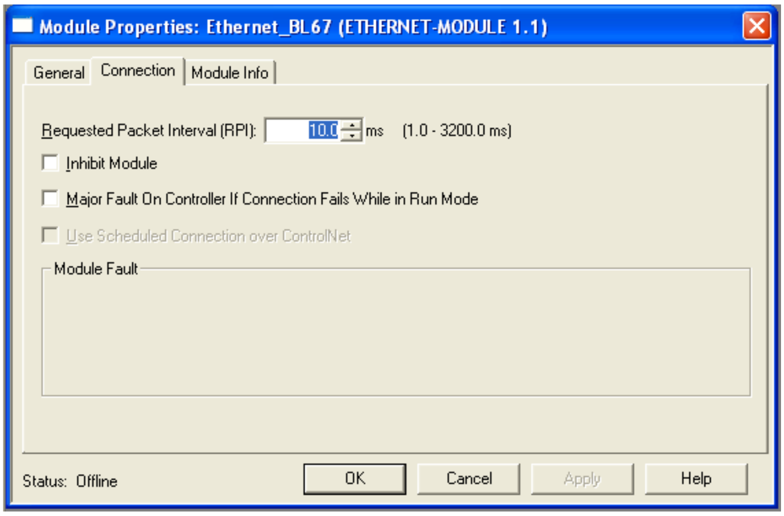


Figure 24: Set connection options for FXEN

I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM)

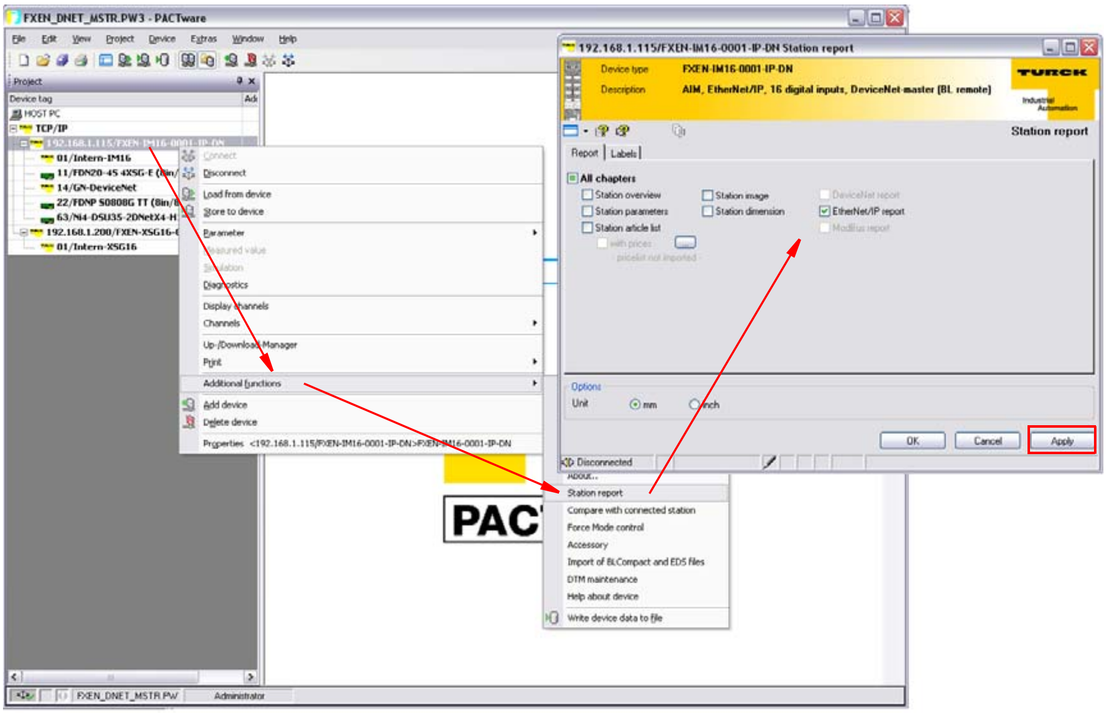
With IO-ASSISTANT 3 (FDT/DTM), an EtherNet/IP-report for each connected EtherNet/IP station can be created.

This EtherNet/IP report is part of the station report for an EtherNet/IP node and contains mapping tables for the complete I/O data (EtherNet/IP station + DeviceNet™ nodes).

1 Created a station report using a right-click on the respective station → Additional functions → station report. The station has to be connected!

2 Activate the EtherNet/IP report check box and create the station report by pressing the "Apply" button.

Figure 25: Creating an EtherNet/IP report



3 The EtherNet/IP report contains a station description as well as separate mapping tables for the in-and output data of the EtherNet/IP-node (status and control word + input data) as well as of the DeviceNet™ nodes connected to the master. For this example configuration it would be as follows:

Figure 26: The EtherNet/IP report for the Example configuration

1.1. Station description

Station address: 192.168.1.115

Adr./Slot	Name	TAG	Descr.	Data Size In	Data Size Out
Slot 0*	FXEN-IM16-0001-IP-DN	192.168.1.115/FXEN -IM16-0001-IP-DN	Term0A	16 bit	16 bit
Slot 1	Intern-IM16	01/Intern-IM16	Term0B	16 bit	
	DeviceNet process data			8 Byte	8 Byte
Total size for in/out data in bytes (rounded on full words)				**12**	**10**

*For detailed information about status/control word see online help

1.2. I/O map for input data

	Byte n+1								Byte n							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Word0*	0A.15	0A.14	0A.13	0A.12	0A.11	0A.10	0A.9	0A.8	0A.7	0A.6	0A.5	0A.4	0A.3	0A.2	0A.1	0A.0
Word1	0B.15	0B.14	0B.13	0B.12	0B.11	0B.10	0B.9	0B.8	0B.7	0B.6	0B.5	0B.4	0B.3	0B.2	0B.1	0B.0
Word2**				DeviceNet Slave Adr. 11 (Byte 1)								DeviceNet Slave Adr. 11 (Byte 0)				
Word3**				DeviceNet Slave Adr. 14 (Byte 1)								DeviceNet Slave Adr. 14 (Byte 0)				
Word4**				DeviceNet Slave Adr. 22 (Byte 1)								DeviceNet Slave Adr. 22 (Byte 0)				
Word5**				DeviceNet Slave Adr. 63 (Byte 1)								DeviceNet Slave Adr. 63 (Byte 0)				

*For detailed information about status/control word see online help
**DeviceNet online process data

Process input data: 12 Byte

1.3. I/O map for output data

	Byte n+1								Byte n							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Word0*	0A.15	0A.14	0A.13	0A.12	0A.11	0A.10	0A.9	0A.8	0A.7	0A.6	0A.5	0A.4	0A.3	0A.2	0A.1	0A.0
Word1				DeviceNet Slave Adr. 11 (Byte 1)								DeviceNet Slave Adr. 11 (Byte 0)				
Word2**				DeviceNet Slave Adr. 14 (Byte 1)								DeviceNet Slave Adr. 14 (Byte 0)				
Word3**				DeviceNet Slave Adr. 22 (Byte 1)								DeviceNet Slave Adr. 22 (Byte 0)				
Word4**				DeviceNet Slave Adr. 63 (Byte 1)								DeviceNet Slave Adr. 63 (Byte 0)				

*For detailed information about status/control word see online help
**DeviceNet online process data

Process output data: 10 Byte

4 The EtherNet/IP report for the station FXEN-IM16-0001-IP-DN with the IP address 192.168.1.115 in this example defines thus an input data size of 12 byte (6 words) and an output data size of 10 byte (5 word).
These sizes have to be entered for the Assembly Instances in RSLogix (see "Configuration of the FXEN", page 3-8).
Those bytes are composed as follows:

Figure 27: In -and output data mapping of the station

Station address: 192.168.1.115

Adr./Slot	Name	TAG	Descr.	Data Size In	Data Size Out	
Slot 0*	FXEN-IM16-0001-IP-DN	192.168.1.115/FXEN -IM16-0001-IP-DN	Term0A	16 bit	16 bit	A
Slot 1	Intern-IM16	01/Intern-IM16	Term0B	16 bit		B
	DeviceNet process data			8 Byte	8 Byte	C
Total size for in/out data in bytes (rounded on full words)				**12**	**10**	D

*For detailed information about status/control word see online help

A *2 byte of input data for the station's Status-Word and 2 byte of output data for the station's Control-Word*
B *2 byte of input data for the 16 internal inputs*
C *8 byte of DeviceNet™ data for the subnet nodes*
D *Total sum of in - and output bytes for the FXEN incl. DeviceNet™ -subnet to be entered in the* ***PLC***

Figure 28: Input data mapping of the station

1.2. I/O map for input data

	Byte n+1								Byte n								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Word0*	0A.15	0A.14	0A.13	0A.12	0A.11	0A.10	0A.9	0A.8	0A.7	0A.6	0A.5	0A.4	0A.3	0A.2	0A.1	0A.0	E
Word1	0B.15	0B.14	0B.13	0B.12	0B.11	0B.10	0B.9	0B.8	0B.7	0B.6	0B.5	0B.4	0B.3	0B.2	0B.1	0B.0	
Word2**	DeviceNet Slave Adr. 11 (Byte 1)								DeviceNet Slave Adr. 11 (Byte 0)								
Word3**	DeviceNet Slave Adr. 14 (Byte 1)								DeviceNet Slave Adr. 14 (Byte 0)								F
Word4**	DeviceNet Slave Adr. 22 (Byte 1)								DeviceNet Slave Adr. 22 (Byte 0)								
Word5**	DeviceNet Slave Adr. 63 (Byte 1)								DeviceNet Slave Adr. 63 (Byte 0)								

*For detailed information about status/control word see online help
**DeviceNet online process data

E *Status-Word of the station*

F *Input data of the modules in the DeviceNet™ subnet (DeviceNet™: addr. 11 to addr. 63), see also* “I/O data mapping for the example station“

Figure 29: Ouput data mapping of the station

1.3. I/O map for output data

	Byte n+1								Byte n								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Word0*	0A.15	0A.14	0A.13	0A.12	0A.11	0A.10	0A.9	0A.8	0A.7	0A.6	0A.5	0A.4	0A.3	0A.2	0A.1	0A.0	G
Word1	DeviceNet Slave Adr. 11 (Byte 1)								DeviceNet Slave Adr. 11 (Byte 0)								
Word2**	DeviceNet Slave Adr. 14 (Byte 1)								DeviceNet Slave Adr. 14 (Byte 0)								H
Word3**	DeviceNet Slave Adr. 22 (Byte 1)								DeviceNet Slave Adr. 22 (Byte 0)								
Word4**	DeviceNet Slave Adr. 63 (Byte 1)								DeviceNet Slave Adr. 63 (Byte 0)								

*For detailed information about status/control word see online help
**DeviceNet online process data

Process output data: 10 Byte

G *Control-Word of the station*

H *Output data of the modules in the DeviceNet™ subnet (DeviceNet™: addr. 11 to addr. 63), see also* “I/O data mapping for the example station“

Note

The I/O-ASSISTANT mapping is depicted in byte format.

In RSLogix, the in - and output size entries at the Assembly instances are normally depicted in words (DATA -INT) or even in double-words (DATA - DINT).

The I/O-ASSISTANT mapping results have thus to be converted into the respective data format.

Downloading the I/O configuration

If the configuration of the network is completed, it can be downloaded to the controller by using for example the "Communication → Download" command.

In the "Download" dialog box, start the download by pressing the "Download" button.

If an error message is generated, warning, that the communication path can not be found, please open the "Path" menu (see Figure 31: "Communication Path"), select your controller and press "Set Project Path" (see Figure 32: "Set Project Path").

Figure 30: Error message

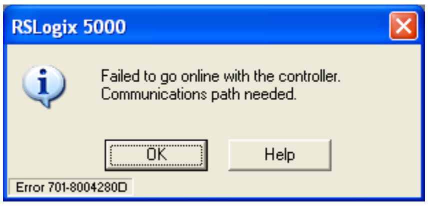


Figure 31: Communication Path

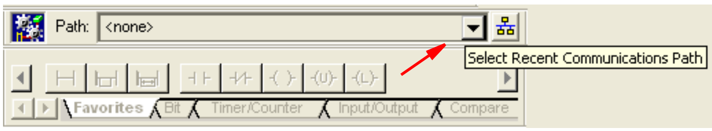
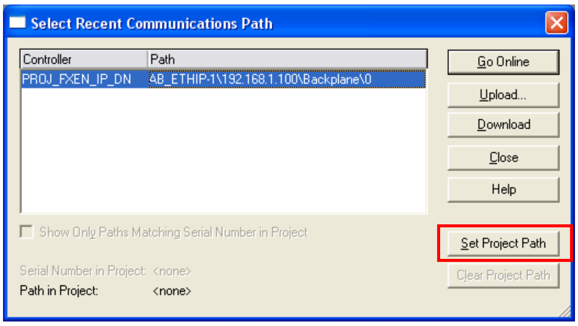


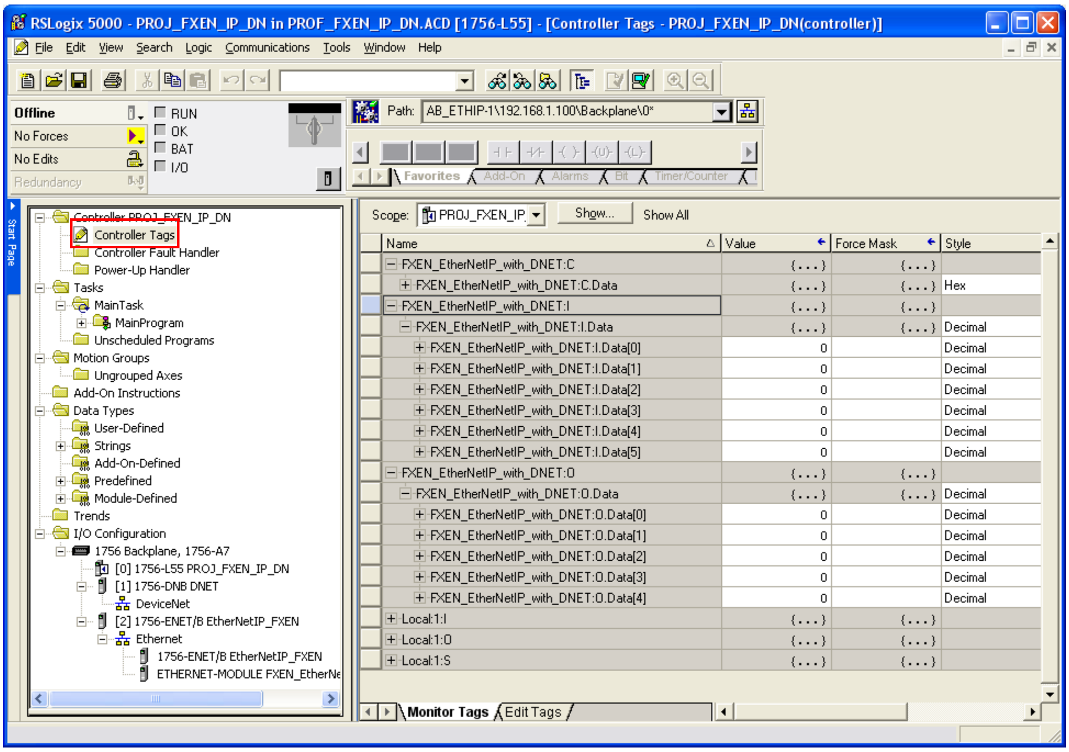
Figure 32: Set Project Path



If the correct communication path is set, it is possible to download the configuration.

Once the I/O configuration is downloaded and the controller is in "Run" or "Remote Run" mode, the I/O-data mapping of the FXEN station is shown in the "Controller Tags":

Figure 33: Controller Tags



The Controller Tags are divided into:

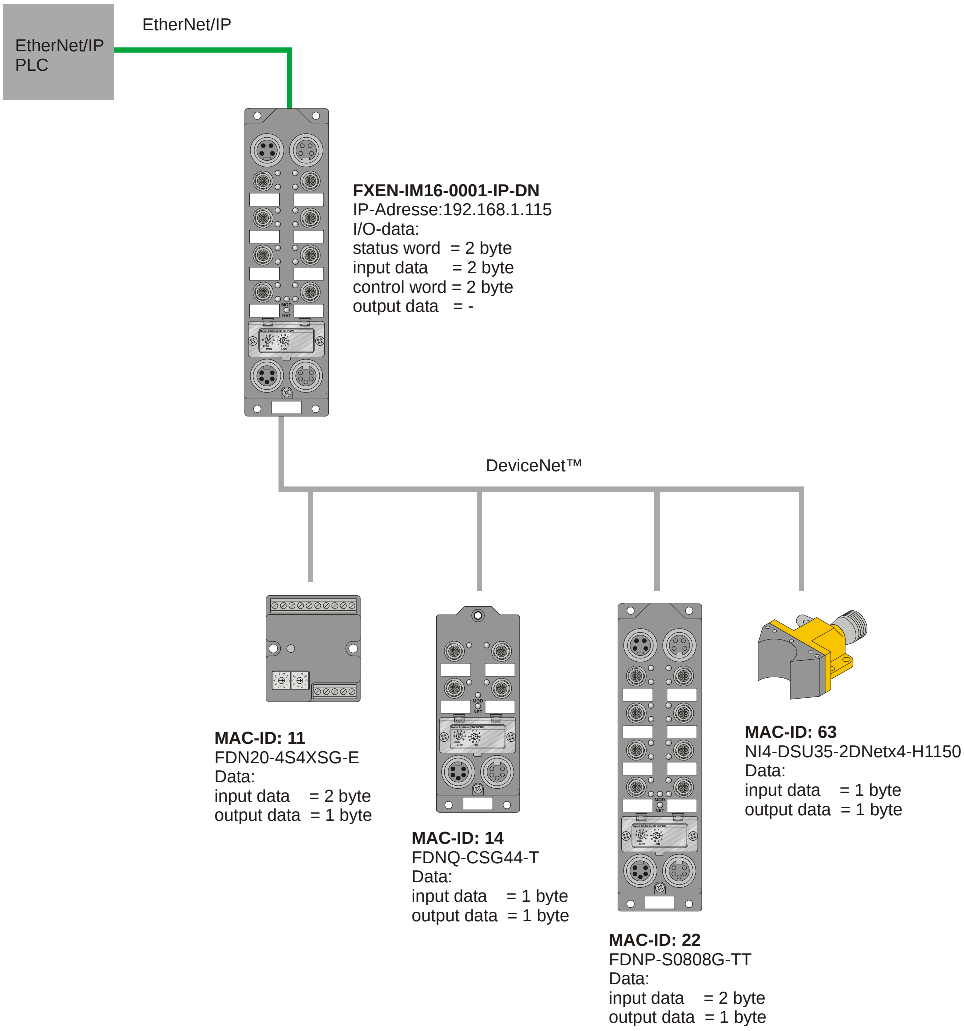
- FXEN_EtherNetIP...: C - mapped configuration data
- FXEN_EtherNetIP...: I - mapped input data
- FXEN_EtherNetIP...: O - mapped output data

3

I/O data mapping for the example station

In order to be able to calculate the I/O-data for the DeviceNet™ nodes, their special I/O data assignments have to be considered.

Figure 34: Example station with DeviceNet™ - subnet



For the I/O data mapping of the DeviceNet™ subnet, please read "I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM)", page 3-10).

Note

The in- and output sizes of the respective DeviceNet™ nodes can be found in the documentation (data sheet, manual etc.) for these products.

- **I/O mapping for a FXEN-IM16-0001-IP-DN**

Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Status Word							
Byte 1								
Byte 2	In7	In6	In5	In4	In3	In2	In1	In0
Byte 3	In15	In14	In13	In12	In11	In10	In 9	In 8
Output	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	Control Word							
Byte 1								

Inx = input x (0 = off, 1 = on)

- **I/O mapping for a FDN20-4S4XSG-E**; MAC-ID: 11

Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	In7	In6	In5	In4	In3	In2	In1	In0
Byte 1	IGS	OGS	reserved					
Output	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	-	-	-	-	Out3	Out2	Out1	Out0

Inx = input x (0 = off, 1 = on)
Outx = output x (0 = off, 1 = on)
IGS = Input Group Status (0 = working, 1 = fault)
OGS = Output Group Status (0 = working, 1 = fault)

- **I/O mapping for a FDNQ-CSG44-T**; MAC-ID: 14

Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	IGS	OGS	-	-	In3	In2	In1	In0
Output	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	-	-	-	-	Out3	Out2	Out1	Out0

Inx = input x (0 = off, 1 = on)
Outx = output x (0 = off, 1 = on)
IGS = Input Group Status (0 = working, 1 = fault)
OGS = Output Group Status (0 = working, 1 = fault)

- **I/O mapping for a FDNP-S0808G-TT**; MAC-ID: 22

Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	In7	In6	In5	In4	In3	In2	In1	In0
Byte 1	IGS	OGS	-	-	-	-	-	-
Output	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	Out7	Out6	Out5	Out4	Out3	Out2	Out1	Out0

Inx = input x (0 = off, 1 = on)

Outx = output x (0 = off, 1 = on)

IGS = Input Group Status (0 = working, 1 = fault)

OGS = Output Group Status (0 = working, 1 = fault)

- **I/O mapping for a Ni4-DSU35-2DNetX4-H1150**; MAC-ID: 63

Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	InErr	-	-	-	-	OutErr	S02_In	S01_In
Output	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	-	-	-	-	-	-	-	ValveOut

S0x_In = Input signal x (0 = off, 1 = on)

OutErr = Output error (0 = no error, 1 = error at output)

InErr = Inout error (0 = no error, 1 = error at input)

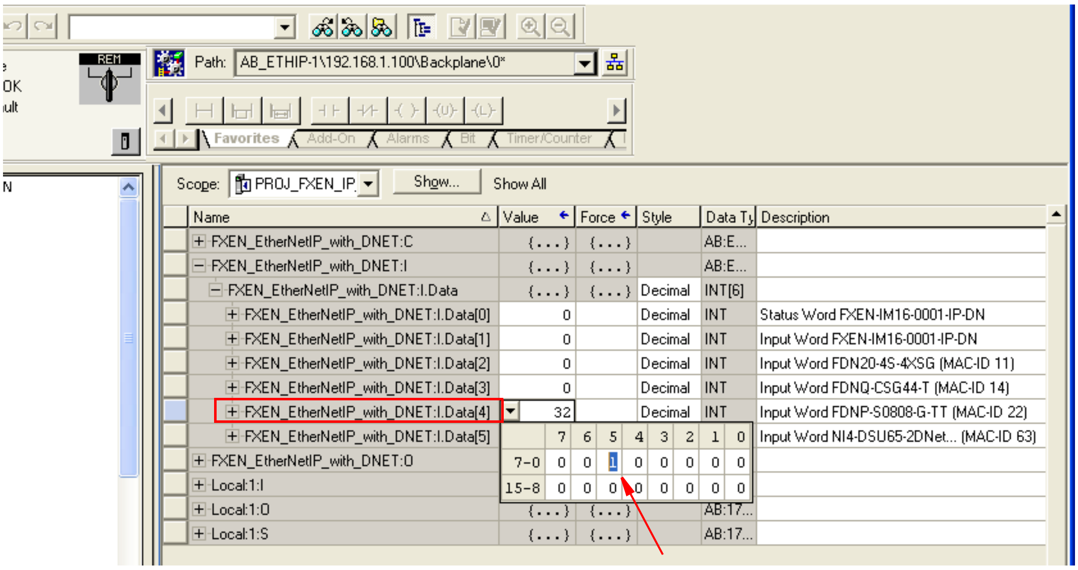
ValveOut = status of valve output (0 = off, 1 = on)

Examples for process data acces

Input data evalutaion

- Input word **I.Data[4]**, Bit 5 is set
  - **I.Data[4]** → input word of DeviceNet™ node with MAC-ID 22 (compare ”I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM)”, page 3-10)
  - Bit 5 of **I.Data[4]** is set → input 5 is active (compare ”I/O data mapping for the example station”, page 3-15)

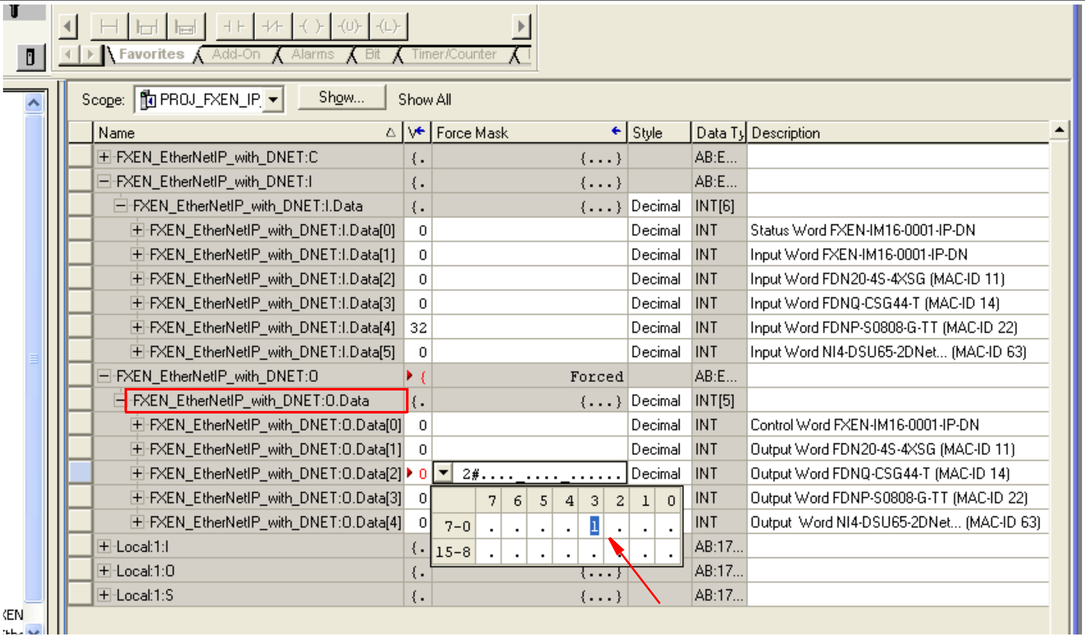
Figure 35: Input active at channel 5, DNet-node with MAC-ID 22



Output data (forcing an output)

- Output channel no. 3 of **a FDNQ-CSG44-T** with MAC-ID 14 has to be set
  - **O.Data[2]** → output word of DeviceNet™ node with MAC-ID 14 (compare ”I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM)”, page 3-10)
  - Bit 3 of **O.Data[2]** is forced → output no. 3 is set (compare ”I/O data mapping for the example station”, page 3-15).

Figure 36: Forcing output channel 3, DNet-node with MAC-ID 14



Explicit Messaging within the PLC program

The access to the EtherNet/IP-classes (Mandatory and Vendor Specific) of an EtherNet/IP-node through Explicit Messages within the PLC program is done via a MSG (Message) instruction.

Before inserting the MSG instruction in your PLC program, please set up the necessary variables (Controller Tags).

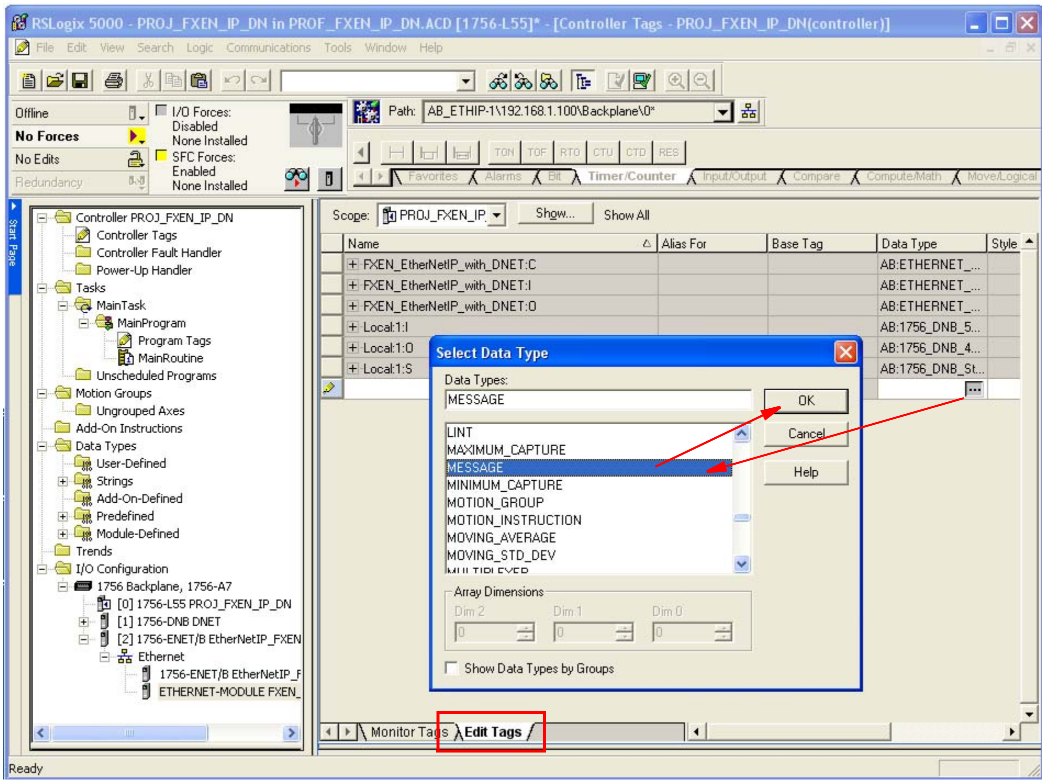
Controller Tag definition

MESSAGE Controller Tag

The MSG instruction in the PLC program requests a Controller Tag (variable structure) of type MESSAGE.

1 To define this Controller Tag open the "Edit Tags" tab and go to the last line of tags. Open the dialog "Select Data Type" in the column "Data Type". Chose the MESSAGE type and confirm your selection with "OK".

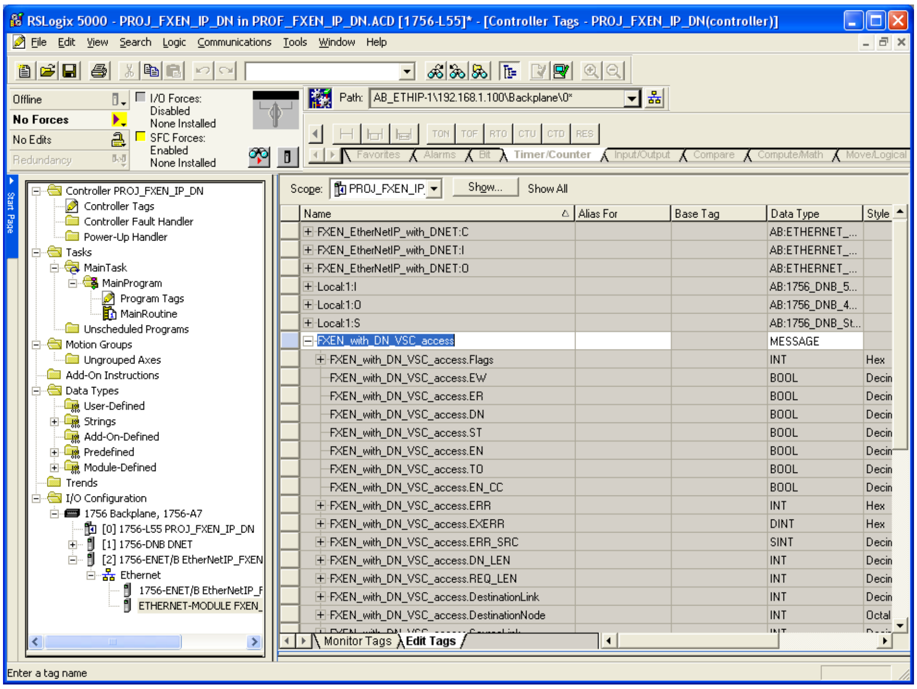
Figure 37: Define a Controller Tag of type MESSAGE



2 Enter a name for the MESSAGE tag structure - in this example "FXEN_VSC_access".

3 The complete MESSAGE structure is created automatically. It contains for example tags for the Class Instance Attribute access.

Figure 38: Structure of the MESSAGE Control Tag

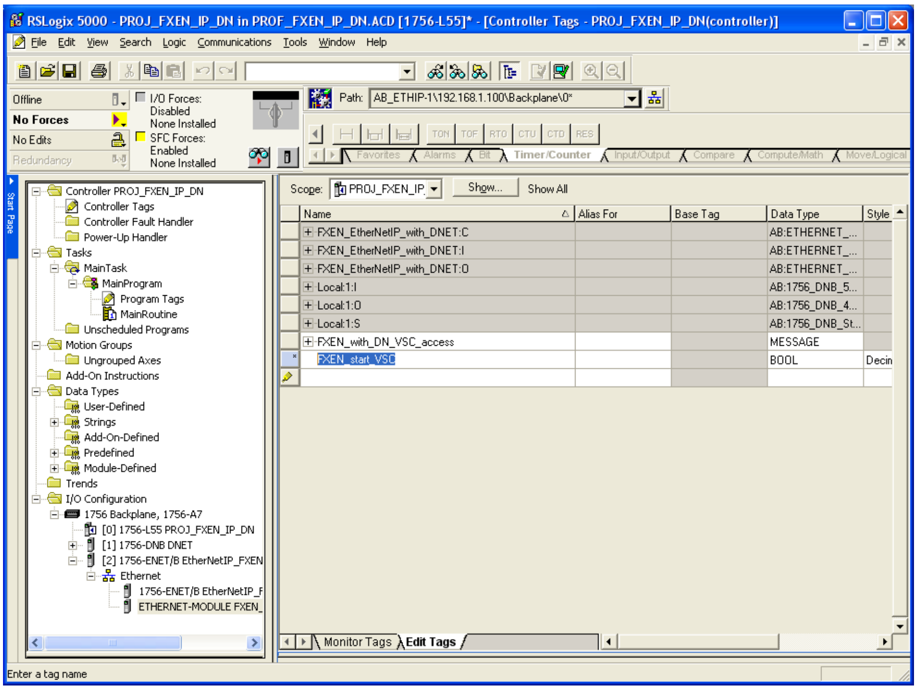


Start bit for MSG instruction

4 The MSG instruction has to be triggered by a start bit which also has to be defined in the Controller Tags.

5 Define a Controller Tag of type BOOL and enter a name - in this example this would be "FXEN_start_VSC".

Figure 39: Start bit for the MSG instruction

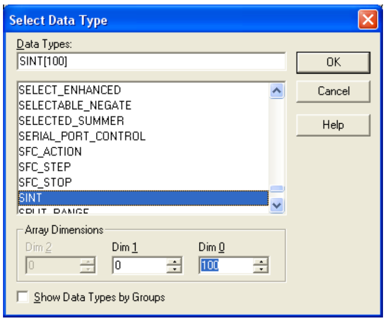


Data array for the MSG instruction

6 The data to be read from the VSCs or to be written to them via the MSG instruction are sent to a data array which has to be defined in the Controller Tags.

7 Define this array by adding a SINT of a special data width - for example 100 byte- to the Controller Tags and enter a name for it - in this example "FXEN_RW_VSC".

Figure 40: Data array for the MSG instruction

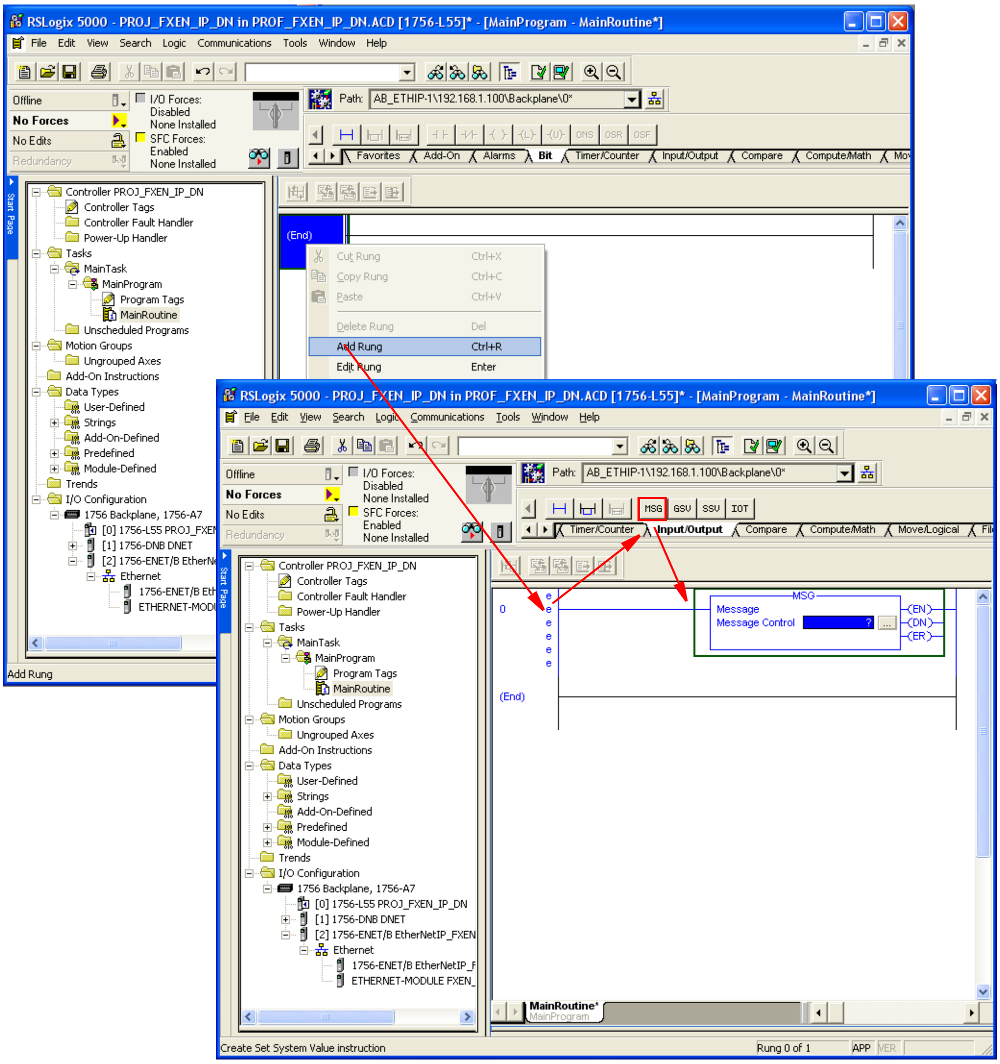


The PLC program

Entering the MSG instruction

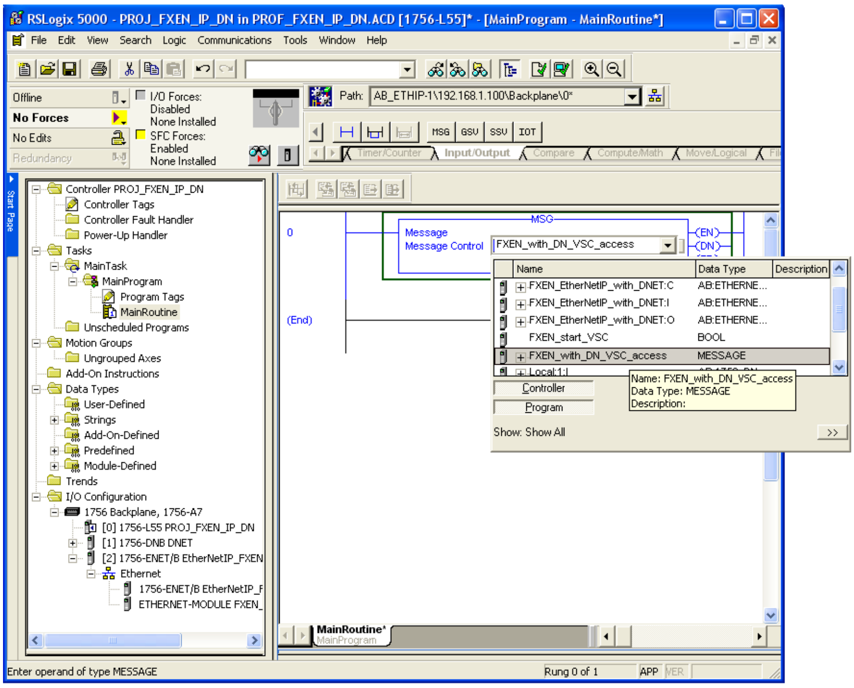
1 Go to the PLC program (Main Routine), add a new network (rung) to the program and insert a MSG instruction.

Figure 41: Add rung and MSG instruction



2 Click on the "?" in the MSG instruction and assign the MESSAGE Controller Tag to be used by the instruction.

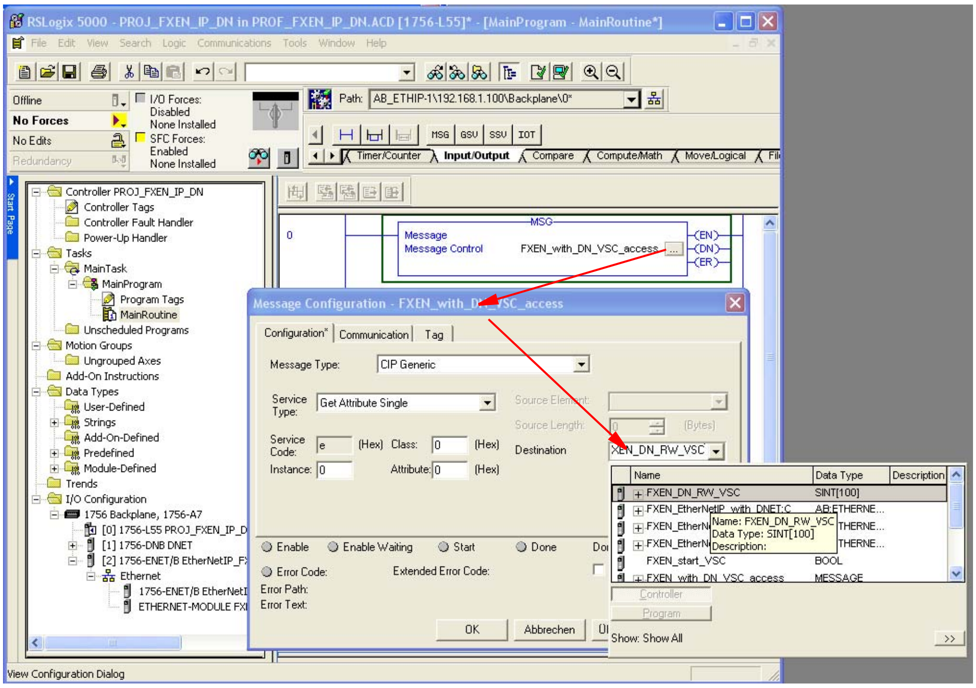
Figure 42: Controller Tag definition for the MSG instruction



3 Open the "Message Configuration" dialog box by pressing the "View Configuration Dialog"-button in the MSG instruction.

4 In this dialog box, chose the "Configuration" tab and define "Message Type", "Service Type" and, for the data read out from the VSCs via MSG instruction, the "Destination" or, for the data to be written to the VSCs via MSG instruction, the "Source Element". In this case this would be the data array defined above "FXEN_RW_acces".

Figure 43: Configuration of the MSG instruction



5 Additionally, please define the Class, Instance and Attribute you want to read out or to write to or set it = 0. Confirm your entries by pressing the "Apply" button and change to the "Communication" tab.

6 Define the Message Path by browsing the I/O Configuration and selecting the EtherNet/IP node with which the PLC has to communicate - in this example this is the FXEN-IM16-0001-IP-DN with the IP address 192.16.1.115.

7 Confirm your settings.

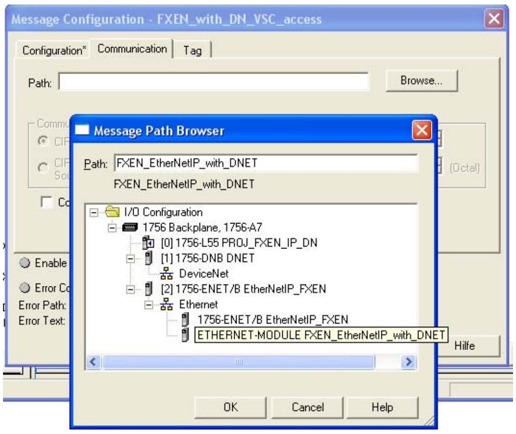


Figure 44: Configuration of the path for the MSG instruction

Triggering the MSG instruction

The MSG instruction has to be triggered by setting a start bit.

8 Add a normally open contact to your rung and assign the "FXEN_start_VSC" bit.

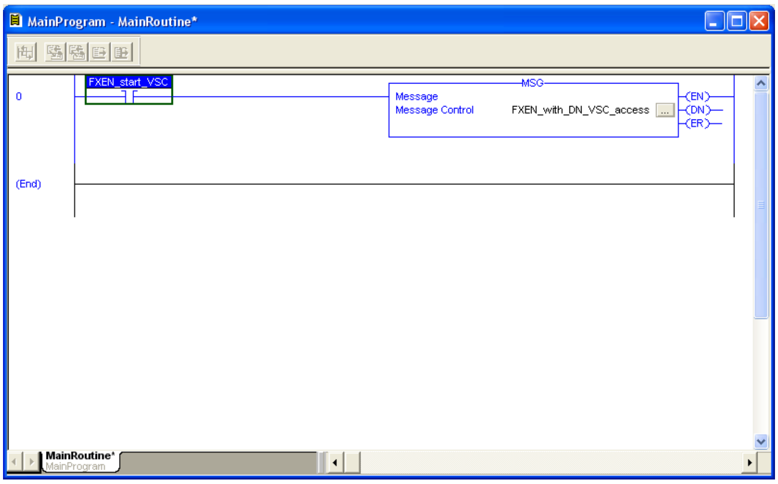


Figure 45: Normally open contact for triggering the MSG instruction

9 Go online with the PLC and download your program.

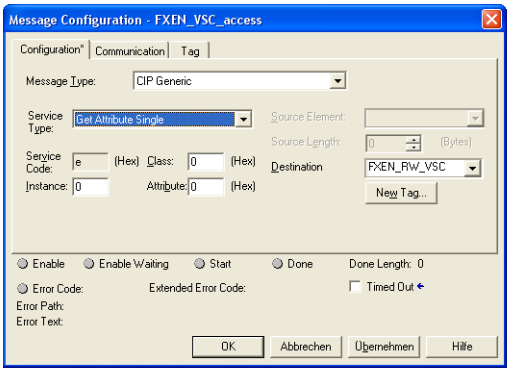
Example for VSC access

Read out product name of FXEN-IM16-0001-IP-DN

The product name of every EtherNet/IP-devices can be found in its Identity Object (Class 01h, Instance 01h, Attribute 07h) defined by the ODVA.

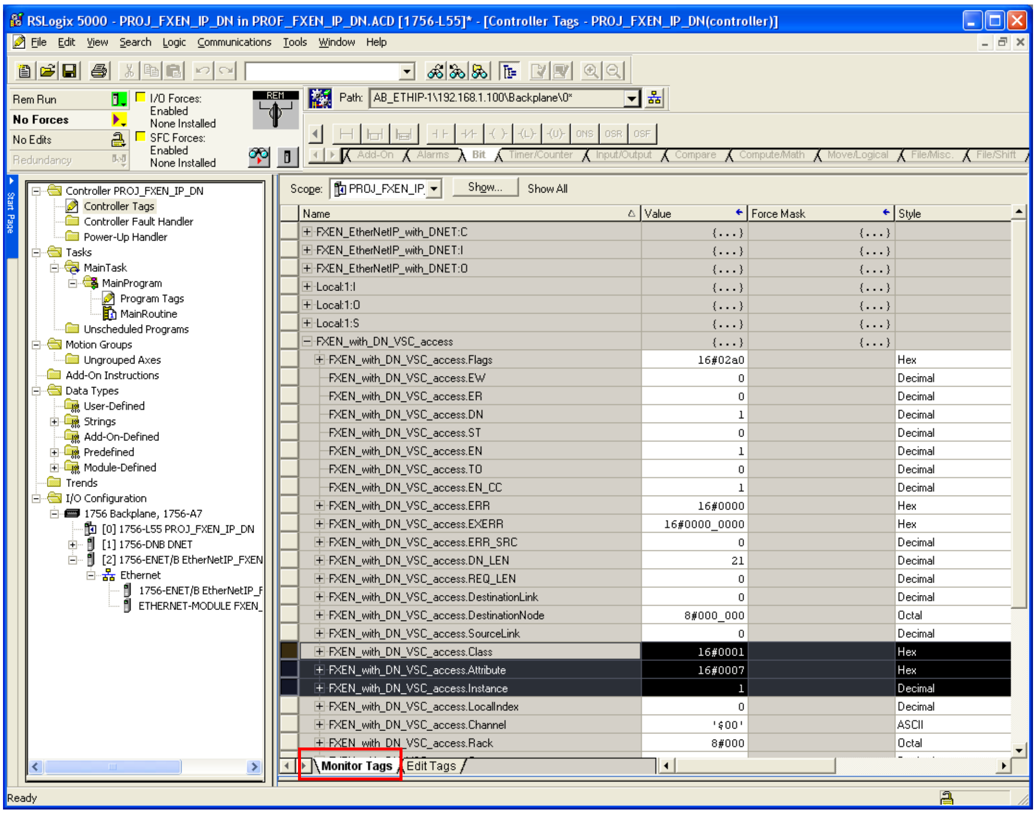
1 In order to read out data from the Classes of an EtherNet/IP-node, the Service Type of the MSG instruction has to be defined as "Get Single Attribute" service.

Figure 46: Get Single Attribute



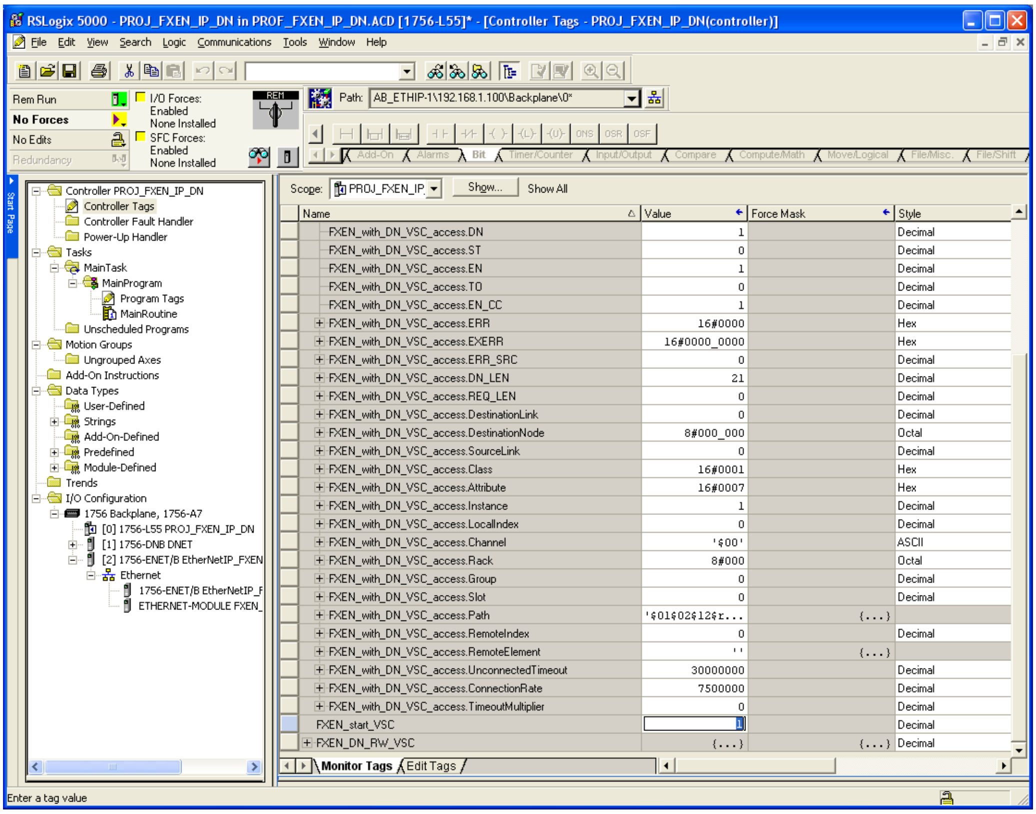
2 Go to the "Monitor Tags" tab in the Controller Tags and enter the Class 01h, Instance 01h, Attribute 07h information into the respective Controller Tag lines (see below).

Figure 47: Monitor Tags



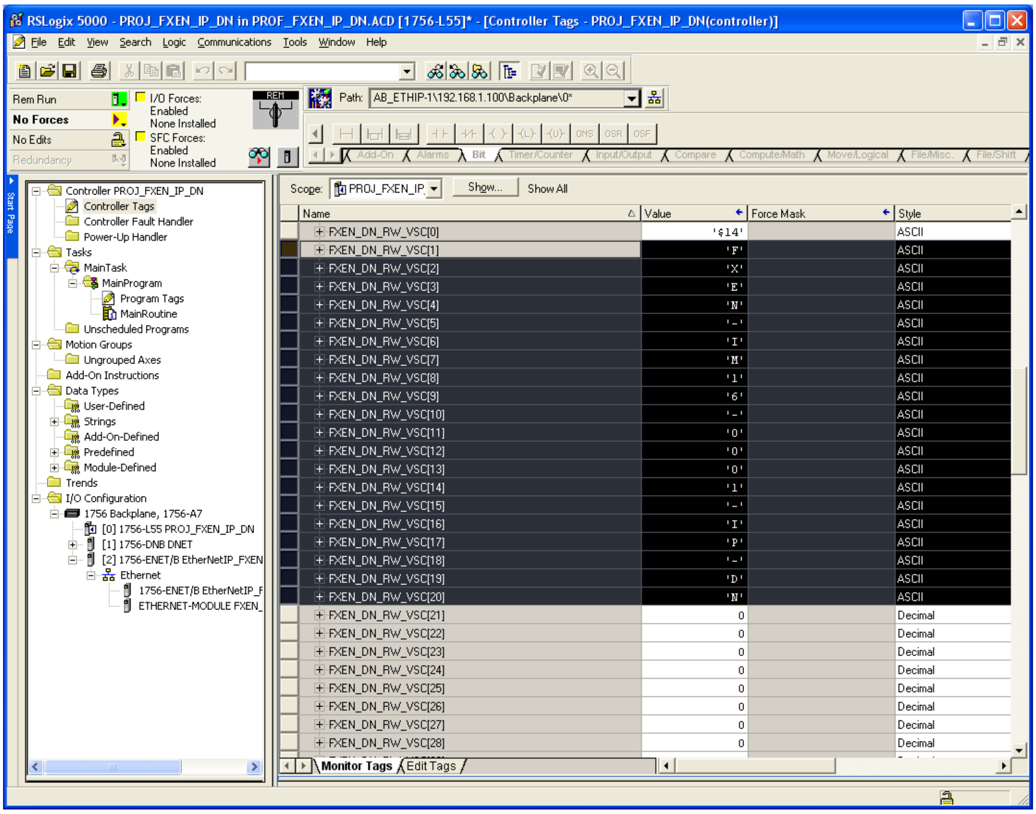
3 Set the start bit "FXEN_start_VSC" to trigger the MSG instruction. Be sure that the I/O Forcing in the PLC in enabled.

Figure 48: Forcing the start bit



4 The result of the Get Single Attribute operation can be found in the Controller Tags "FXEN_RW_VSC".

Figure 49: Forcing the start bit



VSC access to DeviceNet™ master or DeviceNet™ nodes

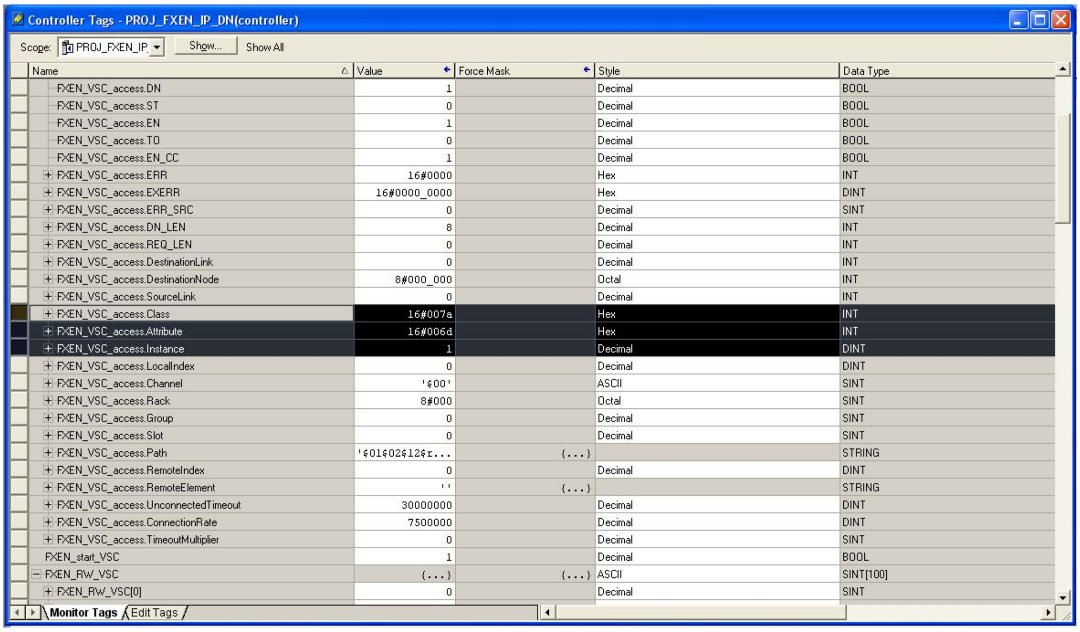
To access the DeviceNet™ master use Vendor Specific Class VSC 122 (7Ah).

To access the connected DeviceNet™ nodes use Vendor Specific Class VSC 123 (7Bh).

Example: Read out the slave list from the master

The Slave list of the master can be read out from VSC122 (7Ah), Instance 1, attribute 109 (6Dh).

Figure 50: Read out the slave list



Controller Tags - PROJ_FXEN_IP_DN(controller)

Scope: PROJ_FXEN_IP Show... Show All

Name	Value	Force Mask	Style	Data Type
FXEN_VSC_access.DN	1		Decimal	BOOL
FXEN_VSC_access.ST	0		Decimal	BOOL
FXEN_VSC_access.EN	1		Decimal	BOOL
FXEN_VSC_access.TO	0		Decimal	BOOL
FXEN_VSC_access.EN_CC	1		Decimal	BOOL
+ FXEN_VSC_access.ERR	16#0000		Hex	INT
+ FXEN_VSC_access.EXERR	16#0000_0000		Hex	DINT
+ FXEN_VSC_access.ERR_SRC	0		Decimal	SINT
+ FXEN_VSC_access.DN_LEN	8		Decimal	INT
+ FXEN_VSC_access.REQ_LEN	0		Decimal	INT
+ FXEN_VSC_access.DestinationLink	0		Decimal	INT
+ FXEN_VSC_access.DestinationNode	8#000_000		Octal	INT
+ FXEN_VSC_access.SourceLink	0		Decimal	INT
+ FXEN_VSC_access.Class	16#007a		Hex	INT
+ FXEN_VSC_access.Attribute	16#006d		Hex	INT
+ FXEN_VSC_access.Instance	1		Decimal	DINT
+ FXEN_VSC_access.LocalIndex	0		Decimal	DINT
+ FXEN_VSC_access.Channel	'$00'		ASCII	SINT
+ FXEN_VSC_access.Rack	8#000		Octal	SINT
+ FXEN_VSC_access.Group	0		Decimal	SINT
+ FXEN_VSC_access.Slot	0		Decimal	SINT
+ FXEN_VSC_access.Path	'$01$02$12$r...	{...}		STRING
+ FXEN_VSC_access.RemoteIndex	0		Decimal	DINT
+ FXEN_VSC_access.RemoteElement	''	{...}		STRING
+ FXEN_VSC_access.UnconnectedTimeout	30000000		Decimal	DINT
+ FXEN_VSC_access.ConnectionRate	7500000		Decimal	DINT
+ FXEN_VSC_access.TimeoutMultiplier	0		Decimal	SINT
FXEN_start_VSC	1		Decimal	BOOL
− FXEN_RW_VSC	{...}	{...}	ASCII	SINT[100]
+ FXEN_RW_VSC[0]	0		Decimal	SINT

Monitor Tags / Edit Tags

Note

Do not forget to set the start bit in "FXEN_start_VSC".

The result can again be found in the Controller Tags "FXEN_RW_VSC" and has to be read a follows:

Each bit of this ARRAY of BYTE (8 byte length) represents one DeviceNet™ MAC-ID:

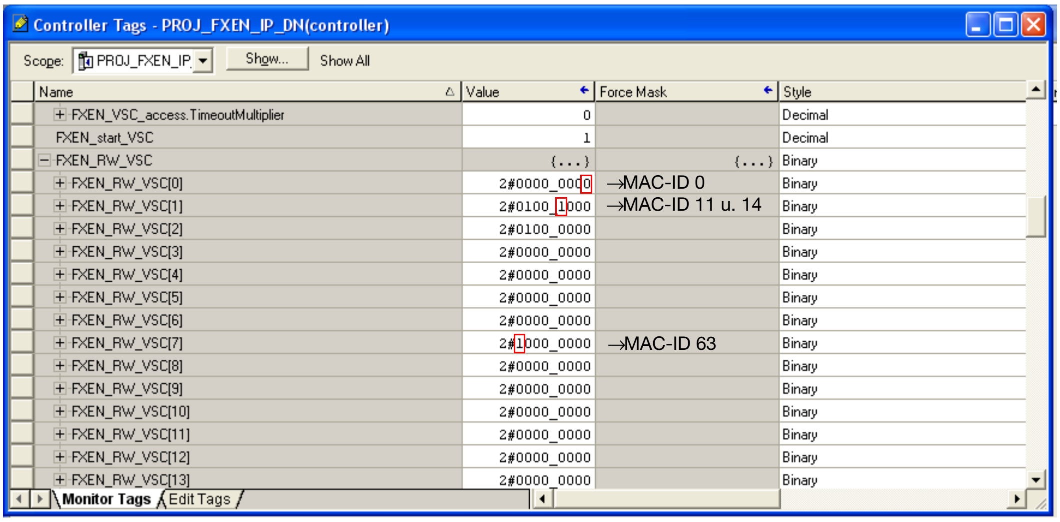
For Example:

- Bit 0 of byte 0 represents MAC-ID 0
- Bit 1 of byte 0 represents MAC-ID 1
  ...
- Bit 8 in byte 8 MAC-ID 63.

If a bit is set to 1, a node with the corresponding MAC-ID has been found in the DeviceNet™ subnet.

In this example, the nodes with MAC-IDs 11, 14, 22 and 63 could be found.

Figure 51: Slave list from example network



Example: Read out the Maj./ Min. revision of DeviceNet™ subnet node with MAC-ID 11

The Maj./ Min. revision of each DeviceNet™ node can be read out from it's Identity Object.

VSC 123 is used to read out the Identity object of every standard DeviceNet™ node (not only TURCK products) via a TURCK Vendor Specific Class.

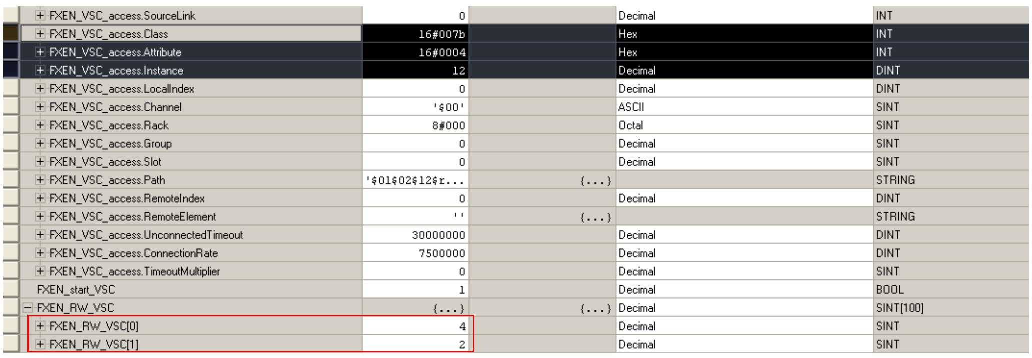
1 To read out the Maj./ Min. revision of the DeviceNet™ node with MAC-ID 11, the followings VSC access has to be done:
VSC 123 (7Bh), Instance 12, Attribute 04h)

Note

Please observe, that the instance **12** is used to read out data from MAC-ID **11**.

The instance-no. corresponds to the node's MAC-ID + 1. Please read also "DeviceNet™ slave class (VSC 123, 7Bh)", page 2-14.

Figure 52: Major/Minor revision of DeviceNet™ *node 11*



2 The result of this VSC access is again sent to "FXEN_RW_VSC".

3 The Maj. revision of the node is 4, the Min. revision 2.

4 DeviceNet™ master with Modbus TCP

Modules for Modbus TCP

BL67-GW-EN-DN

Figure 53: BL67-GW-EN-DN

The BL67 gateway for Modbus TCP with DeviceNet™ master function is part of the modular I/O system BL67.

The gateway is used to connect a modular BL67-station with different I/O and technology modules to Modbus TCP. Additionally, this gateway offers the possibility to connect a complete DeviceNet™ subnet to it and to handle the process data of the DeviceNet™ subnet via Modbus TCP (see also chapter 1).

The connection to Modbus TCP is realized via one 4-pole, D-coded M12 x 1-round connector.

Note
The BL67 gateway contains no integrated Ethernet switch!

Modbus data layout

The DeviceNet™ master's register layout is structured as follows:

Table 13: Modbus Layout

A *If status- and control-word are enabled, otherwise the DN master's process data start at this register address.*

Direction	In place	Register	Process data	Comment
Input	Always	0x0000	Local process data	
	Always (See Note Below)		Status of GW word	
	Always	Module bus Diagnostics		
	Always	1 Bit for each Module		
	Must be enabled	0x0c00 **A**	DN-master status word	0x3089 register enables this information
	Must be enabled		Slave in scanlist 0-15	0x3089 register enables this information
	Must be enabled		Slave in scanlist 16-31	0x3089 register enables this information
	Must be enabled		Slave in scanlist 32-47	0x3089 register enables this information
	Must be enabled		Slave in scanlist 48-63	0x3089 register enables this information
	Must be enabled		Slave error present 0-15	0x3089 register enables this information
	Must be enabled		Slave error present 16-31	0x3089 register enables this information
	Must be enabled		Slave error present 32-47	0x3089 register enables this information
	Must be enabled		Slave error present 48-63	0x3089 register enables this information
	always	0xC00 **A**	DN-master process data	0x3089 register enables this information. Then scanlist defined Slave Process Data. Padded Byte must be added if Slave Byte Data size is odd
	always	0xC00	DN-master process data	In scanlist defined slave process data. Padded byte must be added if slave byte data size is odd

Table 13: Modbus Layout

Direction	In place	Register	Process data	Comment
Output	Always	0x800	Local Process Data	
	Must be enabled	0x3D00 **A**	DN –master control word	0x3089 register enables this information. Then scanlist defined slave output data. Padded byte must be added if slave byte data size is odd.
	Always	0x3D00	DN master process data	In scanlist defined slave output data. Padded byte must be added if slave byte data size is odd.

Table 14: Gateway Status Word

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
V_{out} l	V_{out} h	Lin o		I/O cfg w			I/O Diag
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
I/O err	FM	DN M	DN S	I/O cfg	I/O com	V_{in} l	V_{in} h
...							

Table 15: Meaning of the status bits

Bit	Meaning
I/O err	The communication controller for the I/O system is faulty
FM	The I/O-ASSISTANT Force Mode is active at the gateway.
DN M	DeviceNet Master error: – duplicate MAC-ID has been detected – bus condition is off – wrong data size
DN S	DeviceNet slaves error – not all slaves are in poll state
I/O cfg	The I/O configuration has been changed and is now incompatible
I/O com	No communication on the I/O module bus
Vin l	V_{in} too low
Vin h	V_{in} too high
Vout l	V_{out} too low
Vout h	V_{out} too high
Lin o	line overcurrent

Table 15: Meaning of the status bits

Bit	Meaning
I/O cfg w	Warning, the I/O configuration has been changed
I/O diag	I/O diagnostics active

Process data

The DN master IO data is mapped into registers 0x3C00 - 0x3DFF of the Modbus/TCP register table.

Based on the attached nodes, the scan list is filled in and the I/O data is automatically mapped into the data block (word aligned) based on the consume and produce size.

The consume/produce data of the slave is mapped in order of the actual node addresses on the DeviceNet bus.

Optionally, the control/status information is appended in front of the Process Data - based on the value of the register "0x3089".

For the explanation of the I/O data mapping the following **example subnet structure** is assumed.

Table 16: Example subnet at DeviceNet™ master

DeviceNet™ MAC-ID	**Module**	**DeviceNet I/O data**	
Node address		**Input**	**Output**
2	DeviceNet™ **node A**	2 byte	
7	DeviceNet™ **node B**	2 byte	2 byte
9	DeviceNet™ **node C**	3 byte	7 byte

Note
Again, a padded byte must be added if the slave data size is odd.

Example: input image

	Byte 1	**Byte 0**
Word 0	process data, node address 2	
Word 1	process data, node address 7	
...	process data, node address 9	
Word n-1	padded byte	process data, node address 9
Word n	unused	

Example: output image

	Byte 1	Byte 0
Word 0	process data, node address 7	
Word 1	...	
...		
Word n-2	process data, node address 9	
Word n-1	process data, node address 9	
Word n	padded byte	process data, node address 9

Status/control words of the DeviceNet™ subnet via I/O data

The DeviceNet™ master provides an additional status information (9 words) and control bytes (1 word) for the DeviceNet™ subnet. Status and control information will be mapped in front of Input and Output DeviceNet™ Slave data if the information is enabled. The information is enabled in register 0x3089 (0=Disabled, 1=Enabled (Default)).

Status words

If the 9 status words are enabled and mapped into the process data, they are located in registers **0x3C00** …**0x3Cff**. If they are disabled, then Word 0 would be receiving Input Data from Device Net Slaves.

This status information is structured as follows:

- 1 word for the DeviceNet™ communication (word no. 0)
- 4 words for the "scanlist" information (word no. 1-4)
- 4 words for "error nodes" information (word no. 5-8)

Table 17: Word 0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DNM status	slave missing	empty scan list	CAN error	Dup MacID	subnet input	subnet output	comm. error
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
reserved							

Table 18: Bit meanings

Bit	Meaning
DNM Status	0 = IDLE 1 = RUN
Slave missing	0 = subnet ok 1 = one or more slaves missing
Empty scan list	0 = scanlist ok 1 = The scanlist of the master is empty. No slave has been found during the scan process
CAN error	0 = no error 1 = CAN error (communication problem with CAN controller)
DupMacID	0 = ok 1 = Master DupMacID fault →duplicate MAC-IDs found in the DeviceNet™ subnode
subnet input	0 = ok 1 = the size of the input data of the subnet is too large (max. number of bytes 500 byte)
subnet output	0 = ok 1 = the size of the output data of the subnet is too large (max. number of bytes 500 byte)
comm. error	0 = no error 1 = communication error or bus off.

The following table represents the scan list of the master:

Each node which has been scanned as being a part of the subnet is indicated by one bit (the order is done by Node):

0 = no node is found in GW scan list

1 = node is found and stored in the master's scan list

Table 19: Word 1

Word 1	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1	Node 0
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 15	Node 14	Node 13	Node 12	Node 11	Node 10	Node 9	Node 8
...				...				
Word 4	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 55	Node 54	Node 53	Node 52	Node 51	Node 50	Node 49	Node 48
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 63	Node 62	Node 61	Node 60	Node 59	Node 58	Node 57	Node 56

The following bits describe each node status. They show a list of nodes, to which the DeviceNet™ master could not build up communication:

0 = node present

1 = node not present (node in error)

Table 20: Word 5

Word 5	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1	Node 0
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 15	Node 14	Node 13	Node 12	Node 11	Node 10	Node 9	Node 8
...				...				
Word 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Node 55	Node 54	Node 53	Node 52	Node 51	Node 50	Node 49	Node 48
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Node 63	Node 62	Node 61	Node 60	Node 59	Node 58	Node 57	Node 56

Control word

The control word is used to set the DeviceNet™ master into RUN or IDLE mode.

If the 1 control word is enabled and mapped into the process data, they are located in registers **0x3D00** … **0x3DFF**. If they are disabled, then Word 0 would be receiving Output Data from Device Net Slaves.

This control word is structured as follows:

Table 21: Word 0

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
reserved							1=RUN 0=IDLE
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
reserved							

RUN/IDLE Mode

Default operation: The RUN/IDLE Mode is basically controlled by Modbus/TCP. As soon as the Modbus/TCP connection is established – the subnet goes into the Run Mode. When there is no Modbus/TCP connection present – the gateway is in the IDLE Mode.

Run/IDLE functionality can be overridden by the IO-ASSISTANT Force Mode. In IDLE Mode no output data are sent to the slaves, but input data from the slaves are still read in. Every subnet slave switched into IDLE Mode will set its output to the preconfigured IDLE Status.

In Run Mode output data are sent and input data from the slaves are read in.

Note

The control over Run/IDLE Mode of the DeviceNet™ master is established via the parameter Enable Control/Status Information (0x3089). If the Control/Status information is disabled the default operation will take place. If the Control/Status information is enabled then the Run/IDLE control is part of the I/O Data.

Register layout

This layout contains parameters and data for the DeviceNet™ master.

Note

There are other existing registers associated with a BL67 GW-EN-DN Module. Please refer to the TURCK document no. D300815 to identify the complete register layout for this module.

DeviceNet™ master register layout is as follows:

Table 22: Register layout

Register Range	Access	Name	Type	Description
0x3000 … 0x303F	RO	Input Sizes	Array of Short	This array contains the information of the input sizes of each slave (2 bytes per slave)
0x3040… 0x307F	RO	Output Sizes	Array of Short	This array contains the information of the output sizes of each slave (2 bytes per slave)
0x3080 … 0x3083	RO	Scanlist	Array of bits	Bit position 0…63 define whether the specific slave is present in the scanlist
0x3084 … 0x3087	RO	Error Slaves	Array of bits	Value one of each bit in bit positions 0...63 defines that the specific slave is present in the scanlist but the master couldn’t establish I/O Connection to that slave

Table 22: Register layout

Register Range	Access	Name	Type	Description
0x3088	RO	DeviceNet™ Master Diagnostics	Byte	Bit 0 = Bus off Bit 1 = Output Size is too big Bit 2 = Input Size is too big Bit 3 = Duplicate MacID of the master Bit 4 = CAN Error (possibly due to Bus off) Bit 5 = Empty Scanlist Bit 6 = one or more slaves missing Bit 7 = DNM RUN/IDLE
0x3089	RW	Enable Control, Status information in the DeviceNet process data	Short	If enabled – the first write register is control register, the first 18 read registers are status registers Changes of enable/disable take effect only after restart 0 = disable 1 = enable
0x308A	RW	MacID of the DeviceNet master	Short	Node address, range 0-63, changes of master MacID take effect only after restart 0 = default
0x308B	RW	Baud Rate of the DeviceNet master	Short	0 = 125k (default) 1 = 250k 2 = 500k Changes to baud rate take effect only after restart
0x308C	RW	Quick Connect	Short	0 = disable 1 = enable
0x308D	RW	Subnet Discovery trigger (WHO)	Short	0 = disable 1 = enable (default) – different from ODVA
0x308E	RW	Global EPR	Short	Specifies the EPR in ms, default = 100ms With 100ms the subnet limitations are about: 10 slaves and 256 bytes process data If there are more than 10 slaves or more than 256 bytes of data needed, the user should increase the Global EPR value
0x308F	RO	Interscan period	Short	Interscan period in ms This is status information which allows the user to know how much margin allowed when defining the Global EPR.

Table 22: Register layout

Register Range	Access	Name	Type	Description
0x3100 … 0x34FF	RO	Slave Input Data	Array	Up to 32 bytes input data per slave allocated on the 32 byte boundary. (If the slave data is bigger than 32 bytes – it will be truncated.)
0x3500 … 0x38FF	RW	Slave Output Data	Array	Up to 32 bytes output data per slave allocated on the 32 byte boundary. (If the slave data is bigger than 32 bytes – it will be truncated.)
0x3900 … 0x39FF	RW	Get/Set Object Command Request	Structure	This register set allows to get/set specific system objects which provide a powerful mechanism to configure the DeviceNet™ master parameters, send explicit messages to slaves. The format of this structure is the following: – Byte 0: – Bits 0...3 → req_counter → used for synchronization purposes, – Bits 4...7 = Command 1 = Set 0 = Get – Byte 1,2,3,4: UID – Byte 5,6: Inst – Byte 7,8: Length of the data following byte – Byte 9…: Data (up to 503 bytes of data)

Table 22: Register layout

Register Range	Access	Name	Type	Description

0x3A00 … 0x3AFF	RO	Get/Set Object Command Response	Structure	This register set provides the response to execution of the “object command” described above.
				The format of this structure is the following:
				– Byte 0:
				  – Bits 0 to 3 → req_counter →used for synchronization purposes,
				  – Bits 4...7 = Command
				    1 = Set
				    0 = Get
				– Byte 1, 2, 3, 4: UID
				– Byte 5, 6: Inst
				– Byte 7, 8: Length of the data following byte
				– Byte 9…: Data (up to 503 bytes of data)

0x3B00 … 0x3B3F	RO	Input Offsets	Array of short	Input offsets of each of the slaves on the scanlist
0x3B40… 0x3B7F	RO	Output Offsets	Array of short	Output offsets of each of the slaves on the scanlist
0x3C00 … 0x3CFF	RO	DNM Input	Array of Byte	DeviceNet™ master Input Assembly – this object contains the status/diagnostic data (if enabled) and all slave input data
0x3D00 … 0x3DFF	RO	DNM Output	Array of Byte	DeviceNet™ master Output Assembly – this object is used to control the master via the master control word (if enabled) and to send the subnet data to the DeviceNet™ slaves

5 Application Example: BL67 with Modbus Server Tester

Hard- and software

Hardware

For this application example, the following hardware is used:

- BL67-GW-EN-DN, Modbus TCP Gateway with DeviceNet™ master with:
  - addr. 1, FDNL-CSG88-T: DeviceNet™ module with 8 digital input channels and 8 digital output channels
  - addr. 3, FDNQ-XSG08: DeviceNet™ module with 4 digital input and 4 digital output channels
  - addr. 11, FDNL-L1600-T: DeviceNet™ module with 16 digital input channels
  - addr. 22, FDNL-L0800-T: DeviceNet™ module with 8 discrete input channels
  - also, there are two IO modules connected to the BL67, one BL67-4DI-P and BL67-4DO-0.5A-P

Software

For this application example, the following software is used:

- Modbus Server Tester version 1.5 (Modbus.org)
- IO-ASSISTANT 3 (FDT/DTM) from TURCK (TURCK-DTMs + FDT-Frame Application PACTware™), version 3.5

Network configuration and IP-address-setting

Settings of the network interface card

The TURCK modules for Modbus TCP are delivered with the default IP address 192.168.1.254.

Note
In order to build up the communication between the TURCK product and a PLC/ PC or a network interface card, both devices have to be hosts in the same network.

To achieve this, you have to either:

- to adjust the gateway's IP address via BootP, DHCP etc. for integrating it into your own network
  or
- to change the IP address of the used PC or network interface card

In this example, the network card setting is the following:

IP address: 192.168.1.1

Subnet mask: 255.255.255.0

Address setting at the BL67-module

In this application example, the IP address is set via the rotary switch to 192.168.1.7.

Addresses in the range from 1 to 254 can be allocated in the default subnet 192.168.1. The addresses 0 and 255 are reserved for broadcast messages in the subnet.

Note
The rotary coding switches on the module must be set to "400" or "600" in order to enable the DHCP-Mode or respectively the PGM-DHCP-mode.

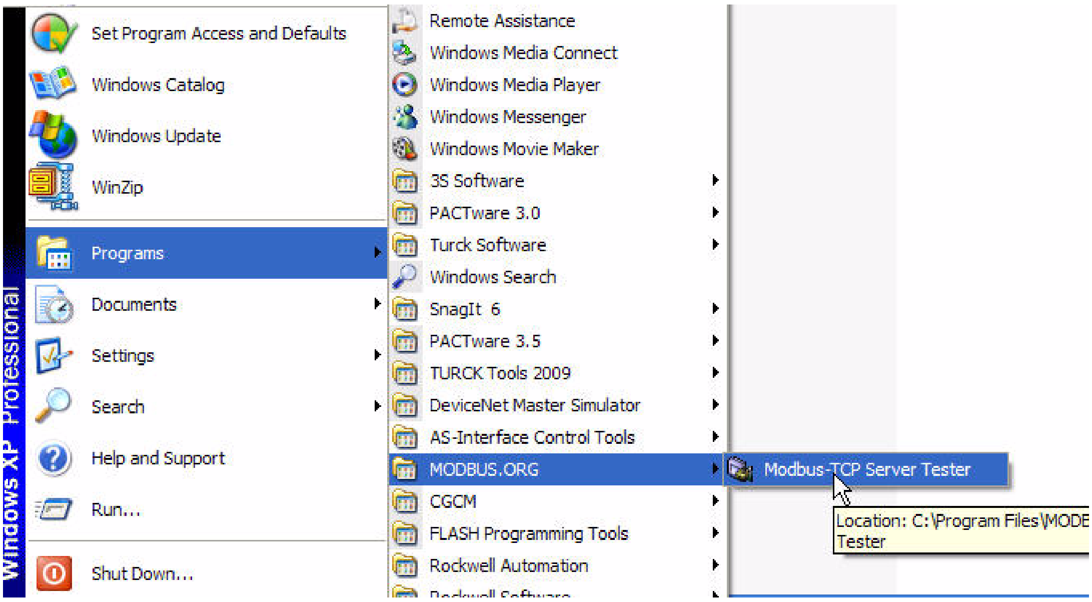
Modbus communications configuration

Modbus Server Tester configuration

The Modbus Server Tester Software is used to test I/O Mapping. The user can confirm bits, bytes, and registers being passed back and forth. The Software can also force outputs and read inputs.

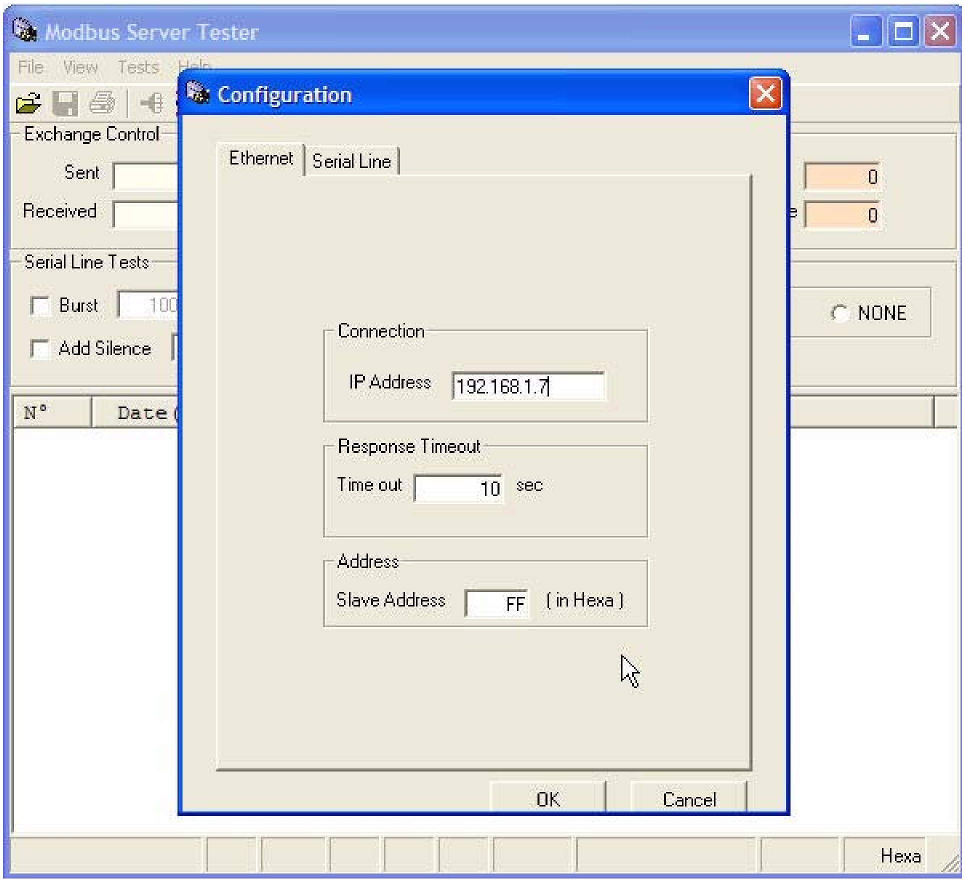
Start the software by going to Program Files → Modbus.org→ Modbus-TCP Server Tester.

Figure 54: Open Modbus Tester



When the Modbus Server Tester is running, the user will be asked to select how they are communicating with the device (Ethernet or Serial). For the BL67-EN-DN, Ethernet is selected. Then, the IP address of the Gateway is entered.

Figure 55: Setting the IP address for Modbus Tester to connect with Gateway



The user will be notified if the connection has taken place or not.

Figure 56: Modbus Tester connection success

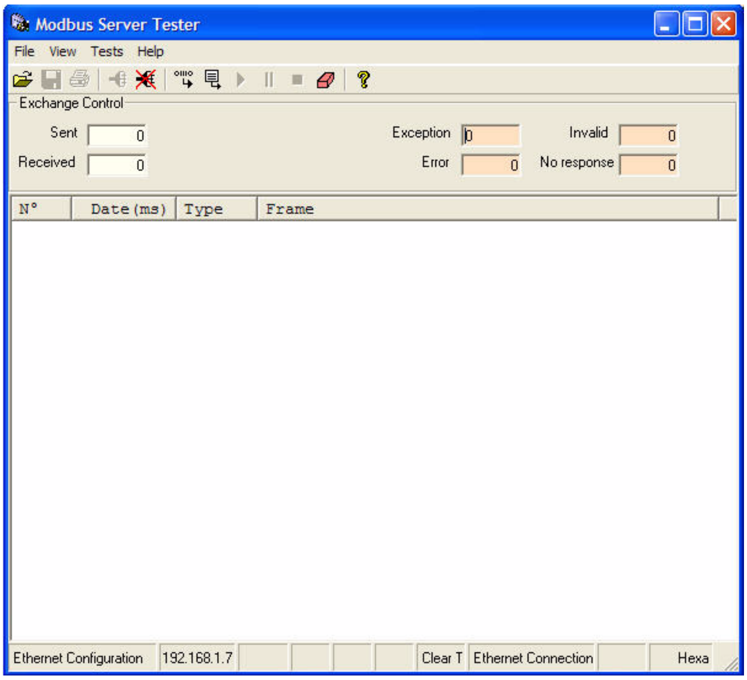


Figure 57: Modbus Tester connection failed

I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM)

With the IO-ASSISTANT 3 (FDT/DTM), an Modbus-report for each connected Modbus station can be created.

This Modbus report is part of the station report for an Modbus node and contains mapping tables for the complete I/O data (Modbus station + DeviceNet™ nodes).

1 Created a station report using a right-click on the respective station → Additional functions → station report. The station has to be connected!

2 Activate the Modbus report check box and create the station report by pressing the "Apply" button.

Figure 58: Creating an Modbus report

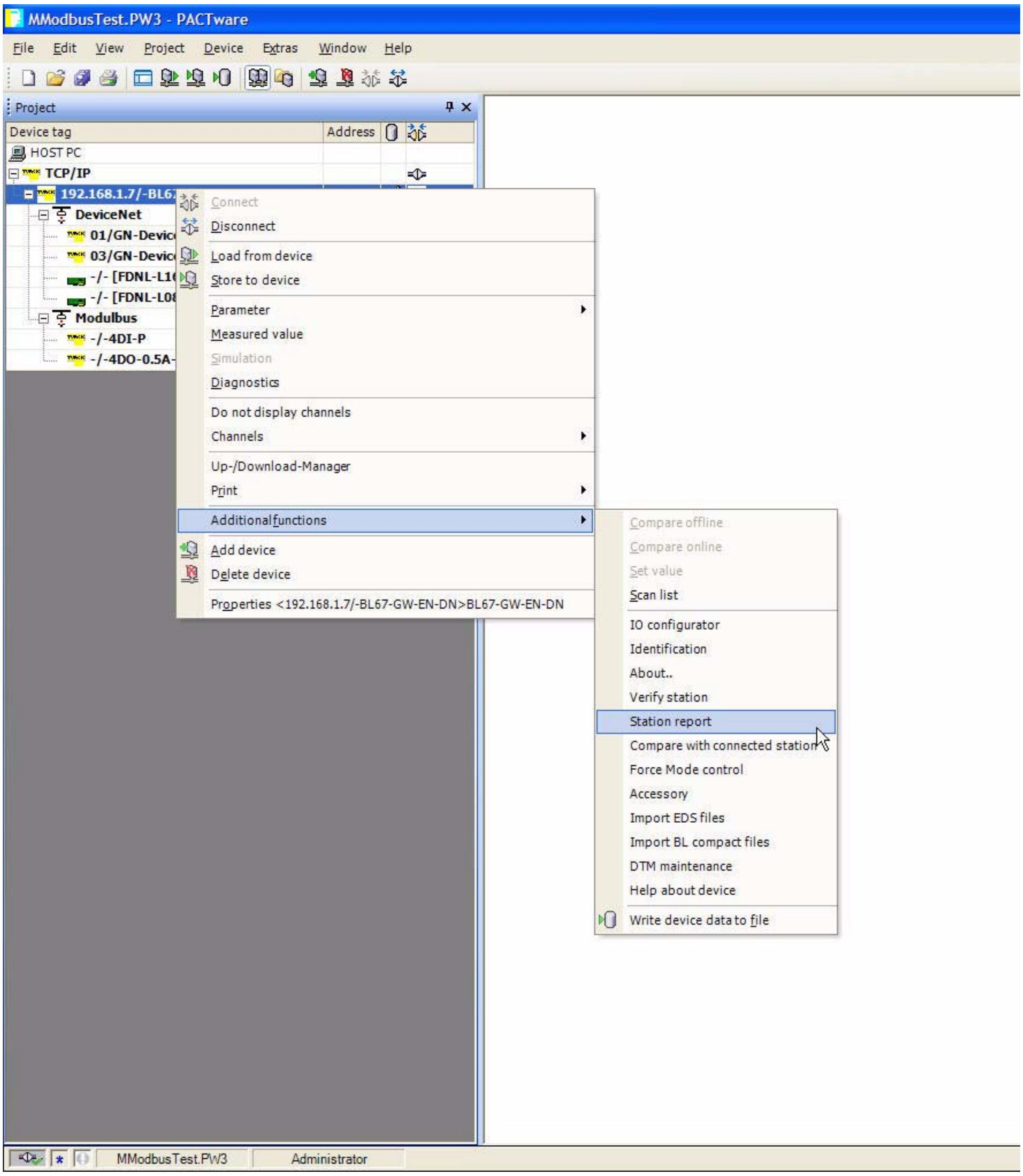
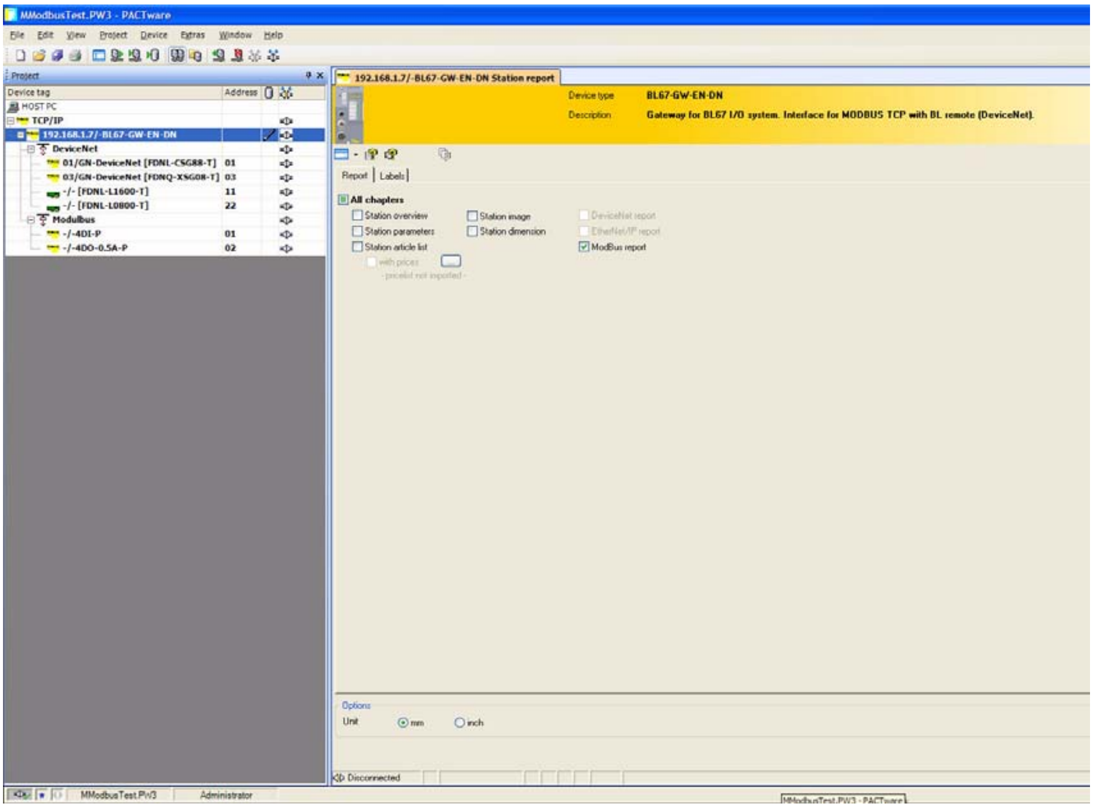
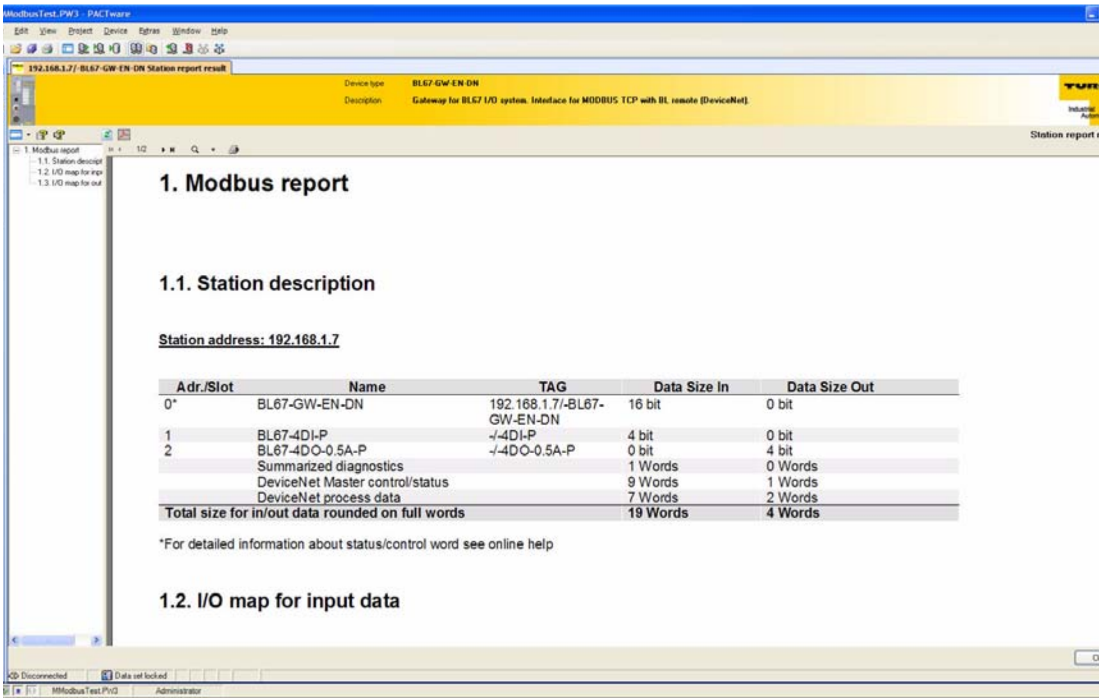


Figure 59: Creating an Modbus report



3 The Modbus report contains a station description as well as separate mapping tables for the in-and output data of the Modbus-node (status and control word + input data) as well as of the DeviceNet™ nodes connected to the master. For this example configuration it would be as follows:



1. Modbus report

1.1. Station description

Station address: 192.168.1.7

Adr./Slot	Name	TAG	Data Size In	Data Size Out
0*	BL67-GW-EN-DN	192.168.1.7/-BL67-GW-EN-DN	16 bit	0 bit
1	BL67-4DI-P	-/-4DI-P	4 bit	0 bit
2	BL67-4DO-0.5A-P	-/-4DO-0.5A-P	0 bit	4 bit
	Summarized diagnostics		1 Words	0 Words
	DeviceNet Master control/status		9 Words	1 Words
	DeviceNet process data		7 Words	2 Words
Total size for in/out data rounded on full words			19 Words	4 Words

*For detailed information about status/control word see online help

1.2. I/O map for input data

Figure 60: The Modbus report for the example configuration

4 The Modbus report for the station BL67-GW-EN-DN with the IP address 192.168.1.7 in this example defines thus an input data size of 38 byte (19 words) and an output data size of 8 byte (4 word).

Those bytes are composed as follows:

- 1 byte (16 bit) of input data for the Station's Status word
- 4 bits of input data for the 4DI-P module
- 4 bits of output data for the 4DO-0.5A-P module
- 1 word of input data for the Gateway Diagnostics
- 9 words of input data for the DN Master Status
- 1 word of output data for the DN Master Control
- 7 words of input data from the Slaves process data
- 2 words of output data from the Slaves control data

Figure 61: Input data mapping of the station

1.2. I/O map for input data

Register		Bit position															
Hex	**Dec**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x0000	0000	-	-	-	-	-	-	-	-	-	-	-	-	01.03	01.02	01.01	01.00
*0x0001	0001	GW.15	GW.14	GW.13	GW.12	GW.11	GW.10	GW.09	GW.08	GW.07	GW.06	GW.05	GW.04	GW.03	GW.02	GW.01	GW.00
**0x0003	0003	-	-	-	-	-	-	-	-	-	-	-	-	-	-	M01	M00
0x3C00	15360	DeviceNet Master status high byte								DeviceNet Master status low byte							
0x3C01	15361	Node in scanlist 8 - 15								Node in scanlist 0 - 7							
0x3C02	15362	Node in scanlist 24 - 31								Node in scanlist 16 - 23							
0x3C03	15363	Node in scanlist 40 - 47								Node in scanlist 32 - 39							
0x3C04	15364	Node in scanlist 56 - 63								Node in scanlist 48 - 55							
0x3C05	15365	Node error 8 - 15								Node error 0 - 7							
0x3C06	15366	Node error 24 - 31								Node error 16 - 23							
0x3C07	15367	Node error 40 - 47								Node error 32 - 39							
0x3C08	15368	Node error 56 - 63								Node error 48 - 55							
***0x3C09	15369	DeviceNet node adr. 1 (Byte 1)								DeviceNet node adr. 1 (Byte 0)							
***0x3C0A	15370	DeviceNet node adr. 3 (Byte 1)								DeviceNet node adr. 3 (Byte 0)							
***0x3C0B	15371	DeviceNet node adr. 11 (Byte 1)								DeviceNet node adr. 11 (Byte 0)							
***0x3C0C	15372	DeviceNet node adr. 11 (Byte 3)								DeviceNet node adr. 11 (Byte 2)							
***0x3C0D	15373	DeviceNet node adr. 11 (Byte 5)								DeviceNet node adr. 11 (Byte 4)							
***0x3C0E	15374	DeviceNet node adr. 22 (Byte 1)								DeviceNet node adr. 22 (Byte 0)							
***0x3C0F	15375	DeviceNet node adr. 22 (Byte 3)								DeviceNet node adr. 22 (Byte 2)							

Description: 1.Column=Register, n. Column=Modul number.register
*) GW: gateway status-/diagnostics bits
**) M: module diagnostics (1 bit for each module)
***) Correct position not available in this firmware version

Process input data: 19 Words

A *Status-word of the station*
B *Input data of the modules in the DeviceNet™ subnet (DeviceNet™: addr. 1 to addr. 22),*

Figure 62: Output data mapping of the station

1.3. I/O map for output data

Register		Bit position															
Hex	**Dec**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x0800	2048	-	-	-	-	-	-	-	-	-	-	-	-	02.03	02.02	02.01	02.00
0x3D00	15616	DeviceNet Master command high byte								DeviceNet Master command low byte							
***0x3D01**	15617	DeviceNet node adr. 1 (Byte 1)								DeviceNet node adr. 1 (Byte 0)							
***0x3D02**	15618	DeviceNet node adr. 3 (Byte 1)								DeviceNet node adr. 3 (Byte 0)							

Description: 1.Column=Register, n. Column=Modul number.register
*) Correct position not available in this firmware version

C *Control-word of the station*
D *Output data of the modules in the DeviceNet™ subnet (DeviceNet™: addr. 1 to addr. 22)*

Note

Please observe:
The IO-ASSISTANT mapping is depicted in byte format.
The mapping results have to be converted into the respective data format.

I/O data mapping for the example station

In order to be able to calculate the I/O-data for the DeviceNet™ nodes, their special I/O data assignments have to be considered.

For the I/O data mapping of the DeviceNet™ subnet, please read I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM), I/O-mapping report via IO-ASSISTANT 3 (FDT/DTM)).

Note

The in- and output sizes of the respective DeviceNet™ nodes can be found in the documentation (data sheet, manual etc.) for these products.

- **I/O mapping for a FDNL-CSG88-T**; MAC-ID: 1

Input	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	In7	In6	In5	In4	In3	In2	In1	In0
Byte 1	IGS	OGS	-	-	-	-	-	-
Output	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	Out7	Out6	Out5	Out4	Out3	Out2	Out1	Out0

Inx = input x (0 = off, 1 = on)

Outx = output x (0 = off, 1 = on)

IGS = Input Group Status (0 = working, 1 = fault)

OGS = Output Group Status (0 = working, 1 = fault)

- **I/O mapping for a FDNQ-XSG08-T**; MAC-ID: 3

Input	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	In7	In6	In5	In4	In3	In2	In1	In0
Byte 1	IGS	OGS	-	-	-	-	-	-
Output	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Byte 0	Out7	Out6	Out5	Out4	Out3	Out2	Out1	Out1

Inx = input x (0 = off, 1 = on)

Outx = output x (0 = off, 1 = on)

IGS = Input Group Status (0 = working, 1 = fault)

OGS = Output Group Status (0 = working, 1 = fault)

- **I/O mapping for a FDNL-L1600-T**; MAC-ID: 11

Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	In7	In6	In5	In4	In3	In2	In1	In0
Byte 1	In15	In14	In13	In12	In11	In10	In9	In8
Byte 2	ISS-7	ISS-6	ISS-5	ISS-4	ISS-3	ISS-2	ISS-1	ISS-0
Byte 3	ISS-15	ISS-14	ISS-13	ISS-12	ISS-11	ISS-10	ISS-9	ISS-8
Byte 4	IOS-7	IOS-6	IOS-5	IOS-4	IOS-3	IOS-2	IOS-1	IOS-0
Byte 5	IOS-15	IOS-14	IOS-13	IOS-12	IOS-11	IOS-10	IOS-9	IOS-8

Inx = input x (0 = off, 1 = on)
ISS = Input Short Status (0 = working, 1 = fault)
IOS = Input Open Status (0 = working, 1 = fault)

- **I/O mapping for a FDNL=L0800-T**; MAC-ID: 22

Input	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	In7	In6	In5	In4	In3	In2	In1	In0
Byte 1	ISS-7	ISS-6	ISS-5	ISS-4	ISS-3	ISS-2	ISS-1	ISS-0
Byte 2	IOS-7	IOS-6	IOS-5	IOS-4	IOS-3	IOS-2	IOS-1	IOS-0

Inx = input x (0 = off, 1 = on)
ISS = Input Short Status (0 = working, 1 = fault)
IOS = Input Open Status (0 = working, 1 = fault)

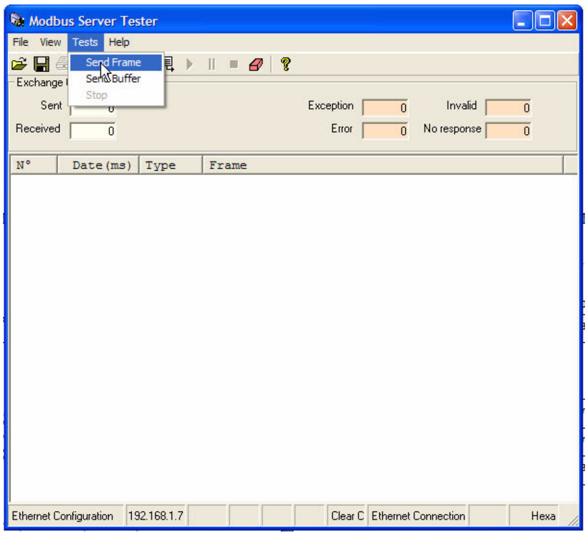
Examples of communication parameter changes and I/O testing

Baud Rate change

The registers for the Baud Rate Setting are Read/Write. When writing a new Baud Rate setting, the gateway will need to have its power cycled so the new setting(s) will be implemented. This example will show you how to set the Baud Rate. The same sequence can be used for all applicable parameter settings.

- the register to change the Baud Rate is 0×308B
  - Under "Tests" select and click on "Send Frame" or click on the "Send Frame" icon on the toolbar

Figure 63: Send Frame



1 In the dialog box, double-click on function code 06, "Write Single Register". A "Request Data" dialog box appears. The function code is set and cannot be changed. The Modbus Server Tester uses the Hex Modbus addressing format. For the Baud Rate, the register address is 0×308B, and for the example, the rate will change to 250 kBit/s. Therefore, the register value would be "1". When complete, click "Finish".

Note

Refer to chapter 4 for the Modbus register layout, if any questions on where parameters are located in respective registers.

Figure 64: Send New Frame

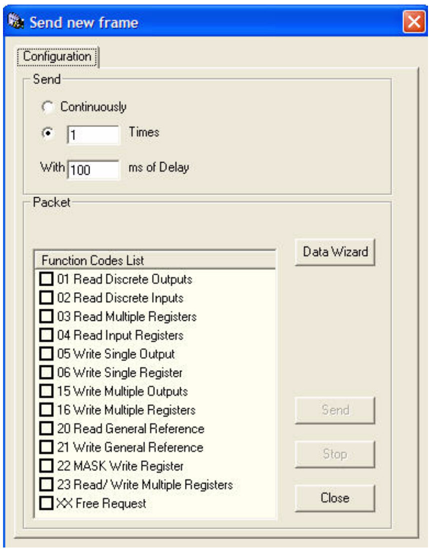
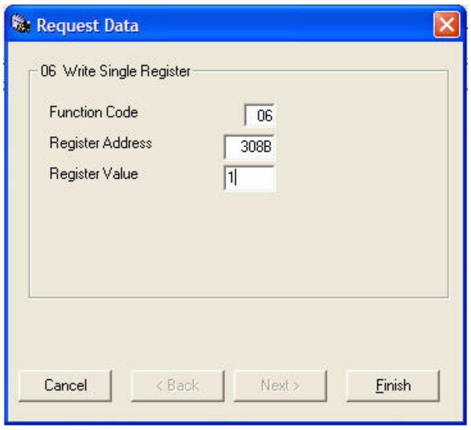


Figure 65: Write to Register 0×308B to set baud rate to 250 kBit/s



2 After clicking "Finish", the Function code 06 "Write Single Register" should be checked. Press "Send", the results will show that a "1" was sent to register 0×308B.

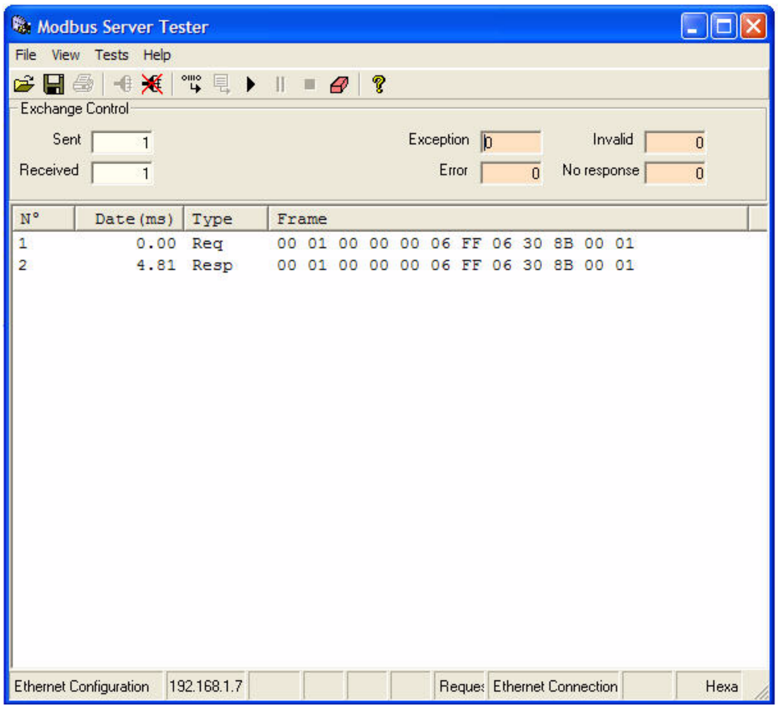
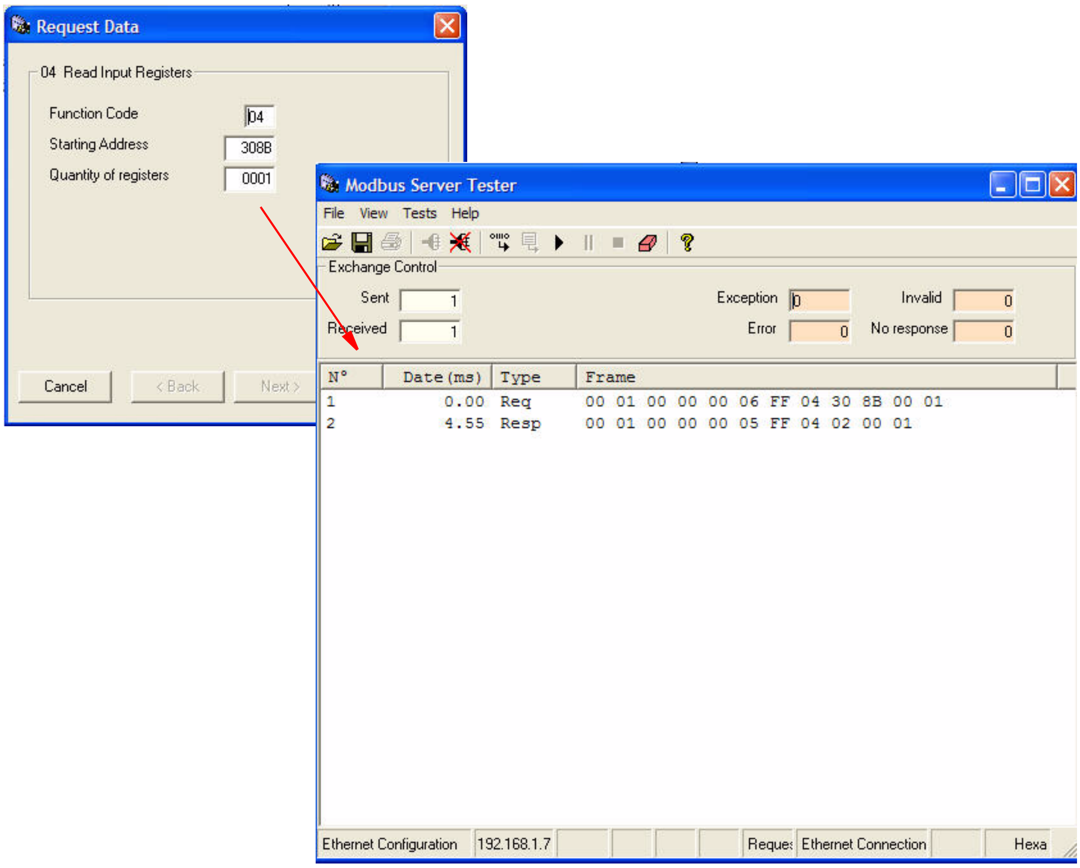


Figure 66: Write results

3 After the results have been confirmed, cycle power on the gateway so the changes will be implemented. When power resumes and the gateway DN has a steady green LED signal. Click on the "Send Frame" icon and double-click on Function Code 04 "Read Input Registers". Enter "Starting address" to be "308B", and the "Quantity of Registers" to be "1". Click "Finish" and then send for the Read Results to confirm that the Baud Rate has been changed to 250 kBit/sec. or 1 in register.

Figure 67: Read results



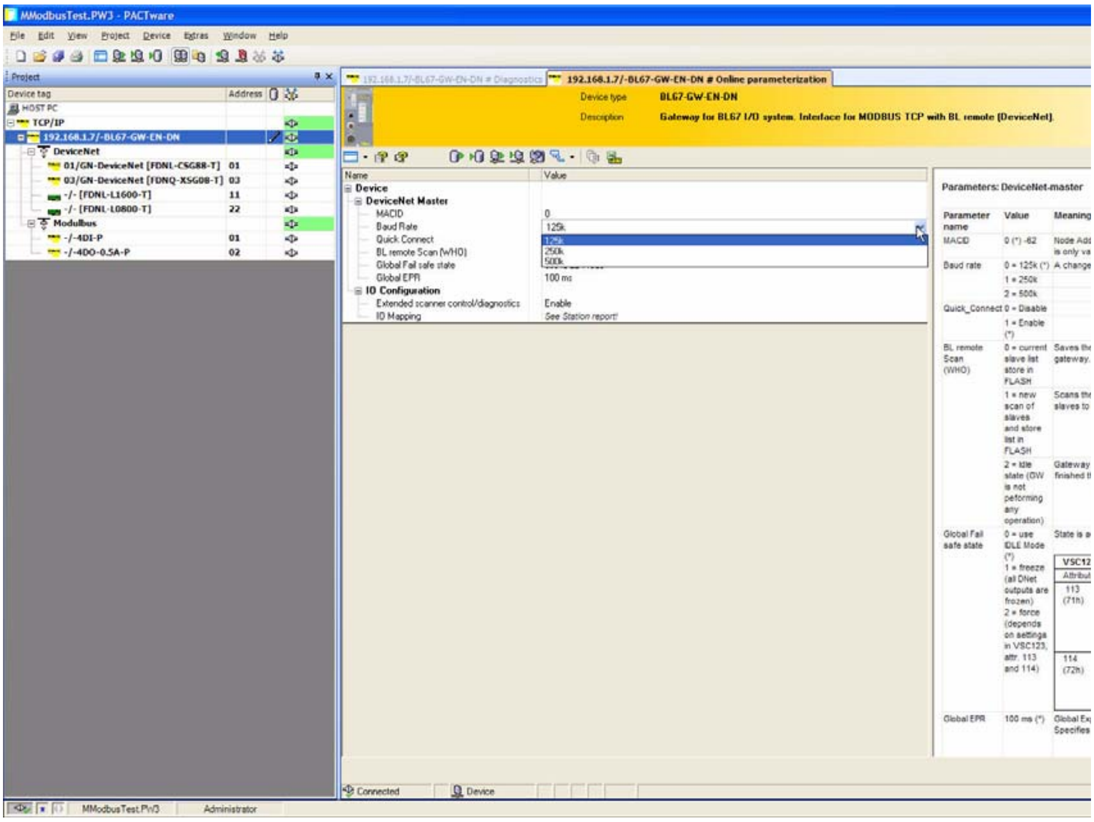
Note

This example would be a good method for changing DN master parameter settings (i.e., MAC-ID). After writing to the register, make sure you cycle power on the gateway and then read the register to ensure that the change has been implemented.

Parameterization with IO-ASSISTANT 3 (FDT/DTM)

The IO-ASSISTANT 3 (FDT/DTM) can make Baud Rate and MAC-ID changes easier. With online parameterization a user can set the baud rate and MAC-ID among other things. Here's an example. Please keep in mind that the gateway will still need to have power cycled for the changes to take effect.

Figure 68: Online parameters

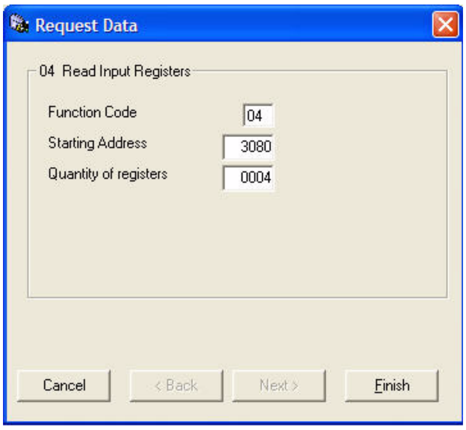


Scanlist and Node Error present

There are two places to see the nodes present in the scanlist. Register 0×3080 to 0×3083 and if the status/control Information is enabled (0×3089), registers 0×3C01 to 0×3C04.
For this portion of the example, the nodes present in the scanlist will be identified in registers 0×3080 to 0×3083. The nodes present in the scanlist and a node error will be created when the status and control Information is enabled.

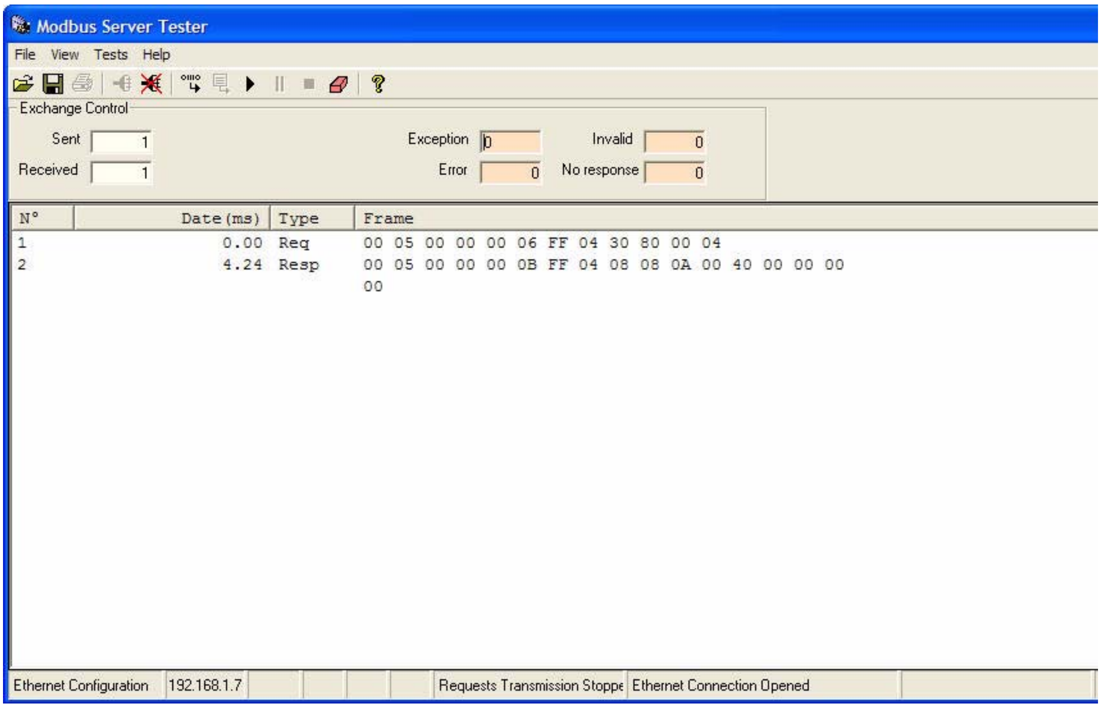
1 Click on the "Send Frame" icon and double click on Function code 04. In the Request Data dialog box for Function Code 4, please enter "3080" in "Starting Address" and "4" in "Quantity of registers. When complete, click "Finish".

Figure 69: Reading the scanlist for nodes present with register 0×3080



2 When the "Send Frame" dialog box appears, make sure Function code 04 checkbox is checked and then press "Send".

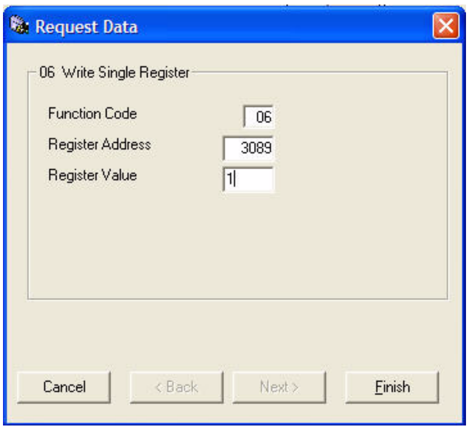
Figure 70: Register 0×3080 Results



3 The results breakdown is as follows: 0A byte number is for Nodes 1 & 3 (2 + 8 = A hex) and 08 byte number is for Node 11. 40 hex byte number is for Node 22.

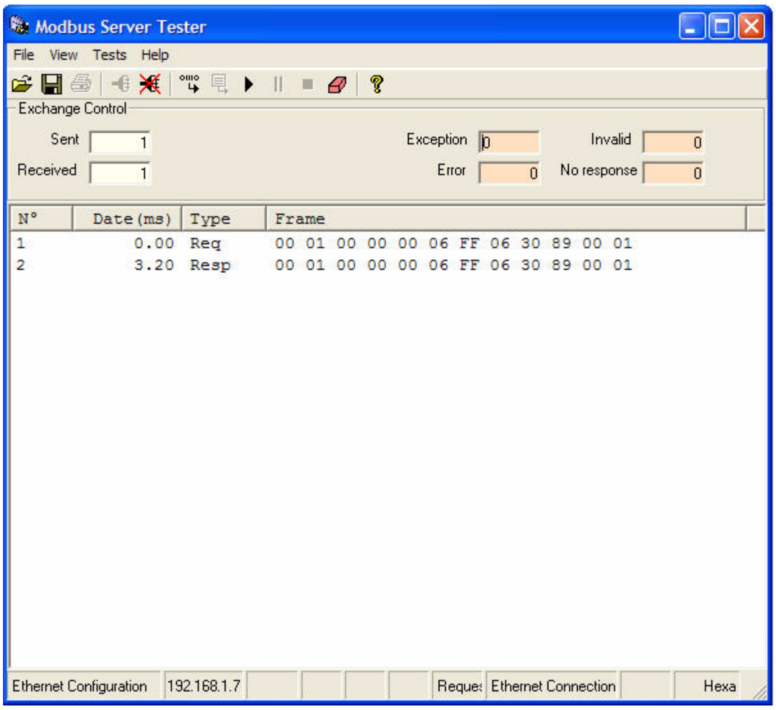
4 To look at the nodes present in scanlist in registers 0×3C01 to ox3C04, the register 0×3089 must have be enabled. To enable the register value must be equal to 1. Click on the "Send Frame" icon then double click on the write single register function code 06. Enter 3089 in the "Register Address" text box and "1" in the "Register value" text box. When complete, click "Finish".

Figure 71: Enable Status/ Control Info



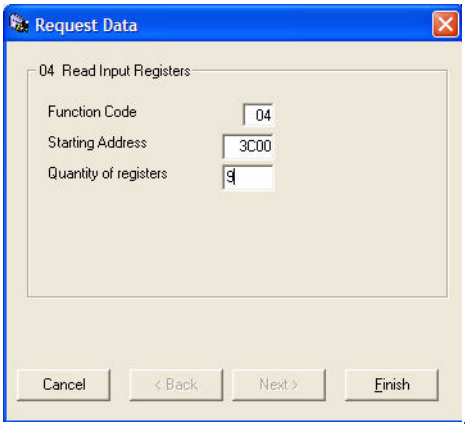
5 When the "Send Frame" dialog box appears, make sure Function code 06 checkbox is checked and then press "Send". The results will then appear.

Figure 72: Results of "Enable Status/ Control Info"



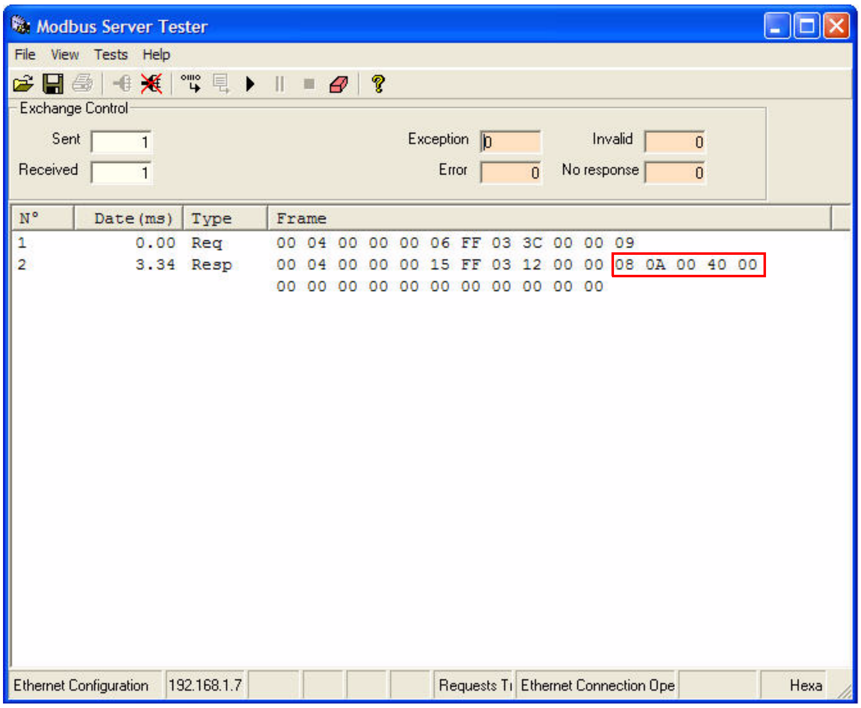
6 To look at the nodes present in scanlist in registers 0×3C01 to 0×3C04, click on the "Send Frame" icon then double click on the read input register function code 04.
Enter 3C00 in the "Starting Address" text box and 9 in the "Quantity of registers" text box (Nine Registers is enough to see the scanlist and potential node errors as well).
When complete, click "Finish".

Figure 73: View registers 3C00 to 3C08



7 When the "Send Frame" dialog box appears, make sure Function code 04 checkbox is checked and then press "Send". The results will then appear.

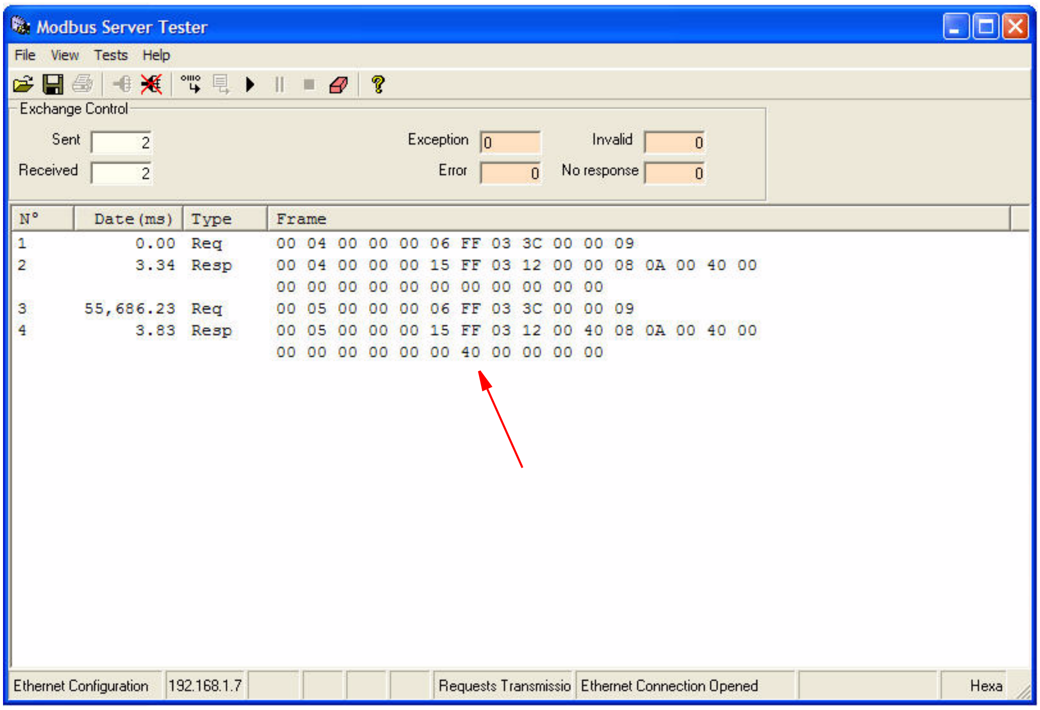
Figure 74: View results of registers 0×3C00 to 0×3C08



8 The results breakdown is as follows: 0A byte number is for nodes 1 & 3 (2 + 8 = A_{hex}) and 08 byte number is for node 11. 40 byte number is for node 22.

9 To produce a node error, node 22 was pulled of the device net network. Then re-send the read input register function code 04.

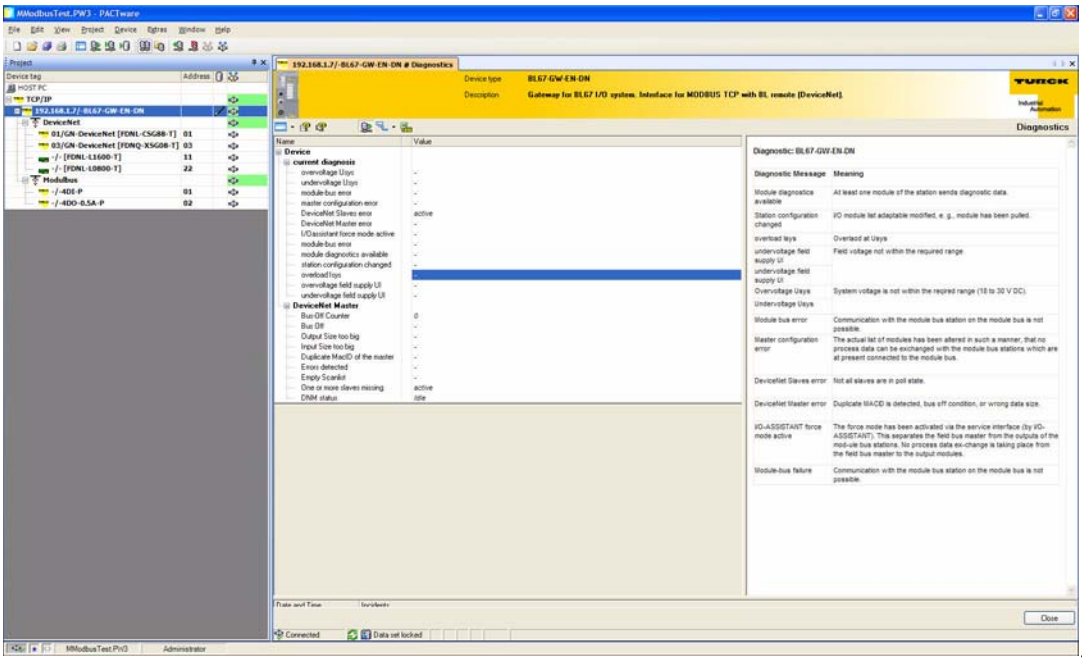
Figure 75: View Results of registers 3C00 to 3C08 with node 22 missing



10 The byte number value 40 shows that node 22 is missing.

Again, IO-ASSISTANT 3 (FDT/DTM) can make this easier. With the Gateway module connected and online, right click on the module and select Diagnostics. The user will be notified that a node is missing.

Figure 76: Diagnostics: Node Missing



Reading slave inputs and forcing slave outputs

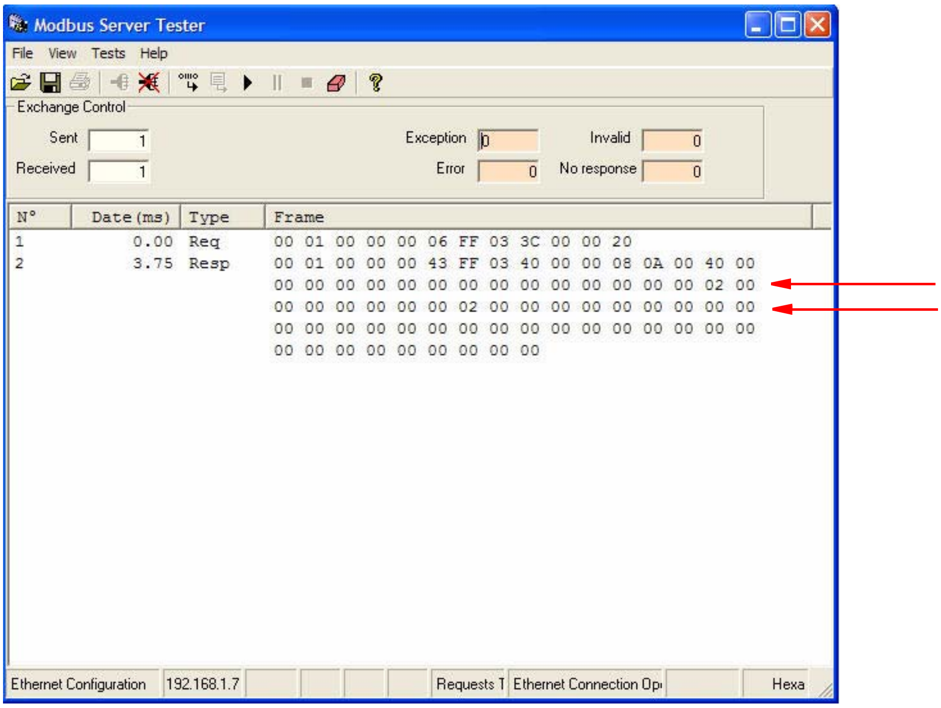
To read inputs and force outputs from slaves, the method is similar to what is done to change parameters. To receive input feedback, read function codes are needed. To force outputs, write function codes are needed.

For this slave input portion of the example, two proximity sensors will be used. The sensors will be connected to node 3 input 1 and node 22 input 1. For the slave output portion, two arbitrary bits will be forced from two different nodes.

1 Click on the "Send Frame" icon and double click on Function code 04. In the "Request Data" dialog box for Function Code 4, please enter "3C00" in the "Starting Address" and "20" in the "Quantity of Registers". When complete, click "Finish".

2 When the "Send Frame" dialog box appears, make sure Function code 04 checkbox is checked and then press "Send". The results will then appear. Status/control Information is enabled still from the previous section.

Figure 77: Input feedback from the 2 proximity sensors. Both sensors are energized



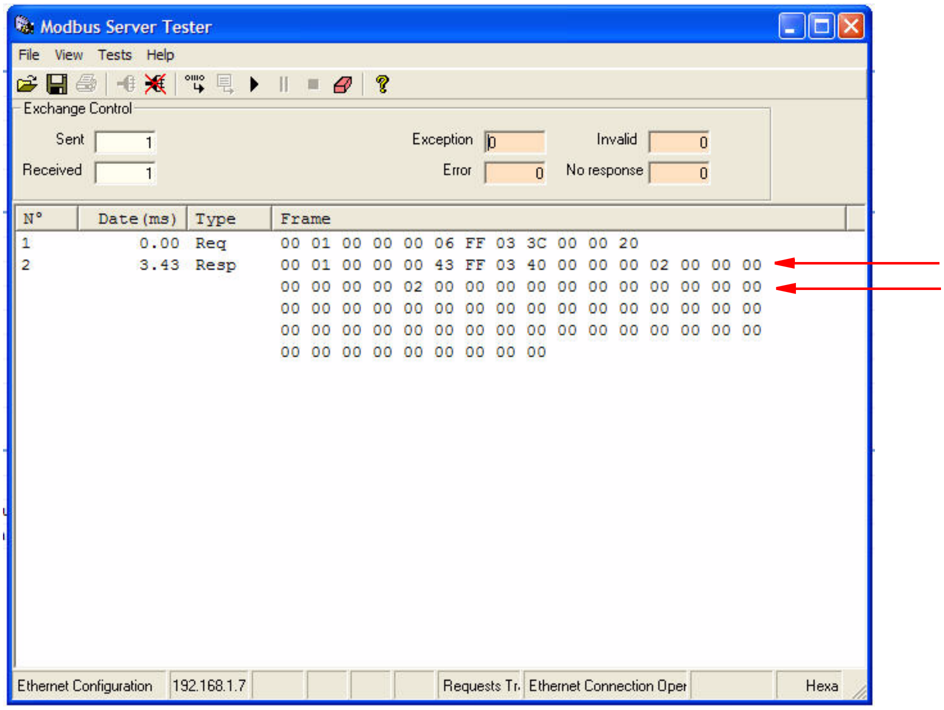


Figure 78: Same results except the status/control information is disabled

Forcing outputs

In this portion of the example, the first output for node 1 will be energized via the Modbus Server Tester software. The status/control information is enabled.

> **Note**
>
> f the status/control information is enabled, address 0×3D00 contains the control information. To force outputs, bit 0 of this register needs to be "1". This indicates that the gateway is in Run Mode.

1 Click on the "Send Frame" icon and double click on Function code 04. In the "Request Data" dialog box for Function Code 4, please enter "3D00" in the "Starting Address" and "1" in the "Quantity of Registers". When complete, click "Finish".

2 When the "Send Frame" dialog box appears, make sure Function code 04 checkbox is checked and then press "Send". The Results will then appear. This verifies that the gateway is in Run Mode.

Figure 79: Verify Run Mode

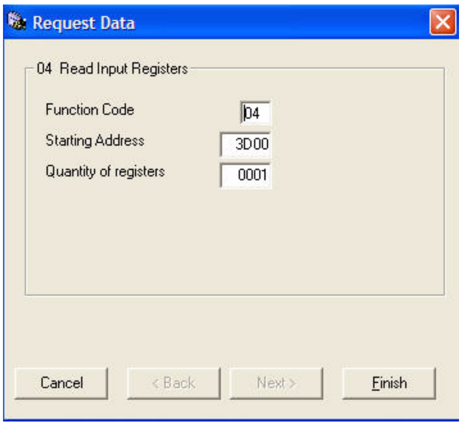
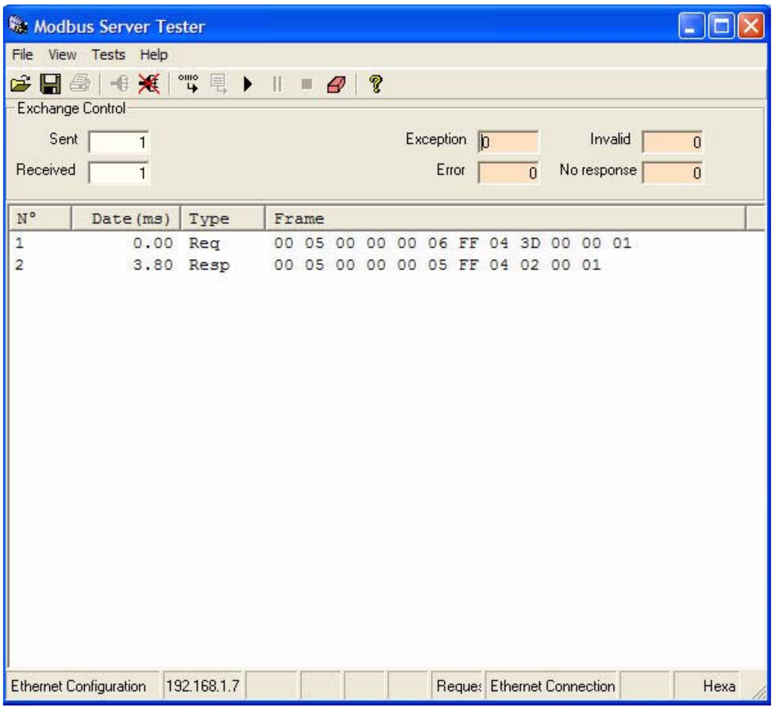


Figure 80: Run Mode is verified



3 Click on the "Send Frame" icon and double click on Function code 06. In the "Request Data" dialog box for Function Code 6, please enter "3D01" in the "Register Address" and "1" in the "Register Value". When complete, click "Finish".

4 When the "Send Frame" dialog box appears, make sure Function code 06 checkbox is checked and then press "Send". The results will then appear. This will force the first output of node 1 on.

Figure 81: Turn node 1, output 1 to ON

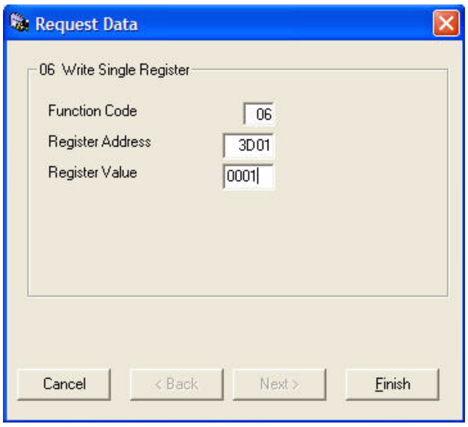
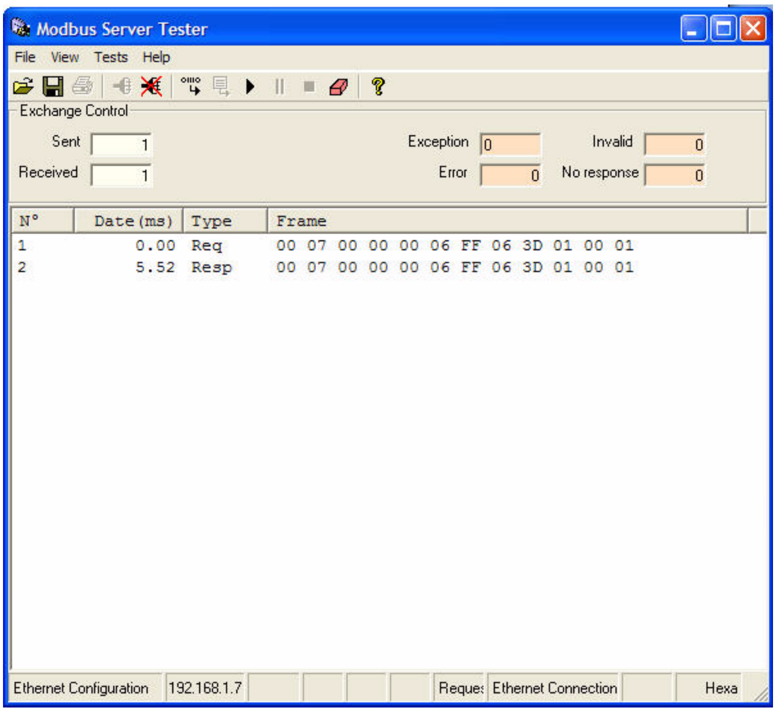


Figure 82: Node 1, output 1 is energized



6 Index

TURCK

Industrial
Automation

D301118 0209

Hans Turck GmbH & Co. KG
45472 Mülheim an der Ruhr
Germany
Witzlebenstraße 7
Tel. +49 (0) 208 4952-0
Fax +49 (0) 208 4952-264
E-Mail more@turck.com
Internet www.turck.com