

Modbus

Data Format Converter



Copyright

Copyright © NeuroCheck GmbH
All rights reserved.
Version 6.2.6
Neckarstraße 76-1, 71686 Remseck, Germany

Phone: +49 (0) 7146 - 89 56-0
Fax: +49 (0) 7146 - 89 56-29
E-Mail: info@neurocheck.com
Web: www.neurocheck.com

Table of Contents

NeuroCheck Modbus Data Format Converter	3
General Information	3
Introduction	3
Installation	4
Master or Slave	5
Data Model	6
Supported Function Codes	8
Message Description	9
Converter Properties	10
Device	10
Protocol	11
Input Pins	12
Output Pins	13
Input Pin Reset Mode	14
Format Options	15
Pin Configuration Dialog	16
Support Contact	18
Info Dialog	18
Support Services	19

Introduction

Communication with the NeuroCheck Modbus Data Format Converter

A so called Data Format Converter (abbreviated Converter or DFC) is a communication driver that converts data that is generated in NeuroCheck in a format that is suitable to a certain kind of hardware and vice versa. The NeuroCheck Modbus Data Format Converter converts data that is exchanged between NeuroCheck and a device driver of the category **Serial Interface and Ethernet**.

Please note that it is necessary to integrate an appropriate **Ethernet** Device Driver to NeuroCheck in order to communicate with a remote device. See section "[Master or Slave](#)" for further information about the integration of a Modbus communication in NeuroCheck.

The interface to NeuroCheck are so called pins. A pin is a data value that is offered by the converter. A pin is either for input data, that is data that is sent to NeuroCheck, or for output data, that is data that is sent by NeuroCheck. In case of this converter the value of a pin is sent and received via the serial communication protocol Modbus.

Modbus protocol

This DFC provides the **Modbus TCP** protocol. Other Modbus protocol modes (ASCII and RTU) are NOT supported. A part of the Modbus protocol is the Unit ID which is unnecessary in the Modbus TCP mode. The Modbus DFC as Master uses 0 as Unit ID to send requests. The NeuroCheck Modbus DFC as slave will always answer with the Unit ID received from the request. For the various reading and writing operations the Modbus protocol specifies several function codes. See chapter [Supported Function Codes](#) for more details.

Network settings

The Modbus protocol enables communication to devices connected to the same network. Each device intended to communicate using Modbus is given a unique address in the network. Make sure the NeuroCheck device connected to the DFC is set to the right address. Typically the TCP-Port 502 is reserved for Modbus communication. The DFC itself has no properties for the network configuration.

Installation

1. Make sure that you have installed the NeuroCheck Modbus Data Format Converter. The following files must be present in the NeuroCheck installation folder:

- NcFmtCnv.NeuroCheck.Modbus.NET.dll
- NcFmtCnv.NeuroCheck.Modbus.UI.NET.dll

If not, download and install the latest Data Format Converter update from <http://www.neurocheck.com>.

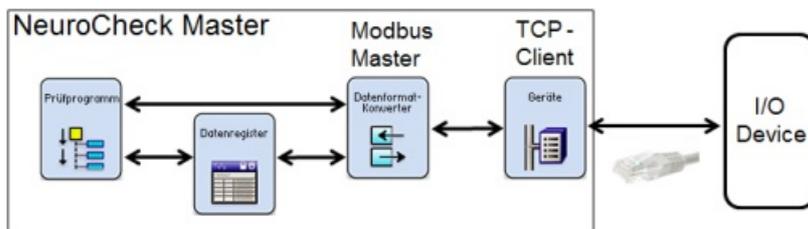
2. Configure the data format converter as new converter in NeuroCheck:
 - a. In NeuroCheck, open the Data Format Converter Manager (Menu **System | Data Format Converter Manager**)
 - b. Click **New ...** to start the NeuroCheck **Data Format Converter Wizard**.
 - c. Select option **Serial interface and Ethernet** on the first page and continue.
 - d. On the second page, select **Modbus** and continue.
 - e. On the last page, select the device that should be used to send and receive data and click **Finish** to close the NeuroCheck **Data Format Converter Wizard**.
 - f. Select the new converter node.
 - g. Choose **Properties ...** to open the dialog for the converter settings and configure the settings for your application (see Converter Properties).
 - h. Close the properties dialog with **OK**.
3. Close the NeuroCheck **Data Format Converter Manager** with **OK**.

Master or Slave

Due to the fact that adding a Ethernet device into the NeuroCheck device manager has to be done before adding a Data Format Converter, you have to be sure about the configuration you need. If you have an I/O device which shall be controlled from NeuroCheck, the device is the Modbus slave and NeuroCheck is the master. In this case add at first a Ethernet TCP-Client device to the NeuroCheck device manger. Then add a new Modbus DFC to the DFC manager. The Modbus DFC reads the information from the device and decides automatically whether the Modbus is a master configuration or a slave configuration. The following images show the two use cases for a master or slave configuration.

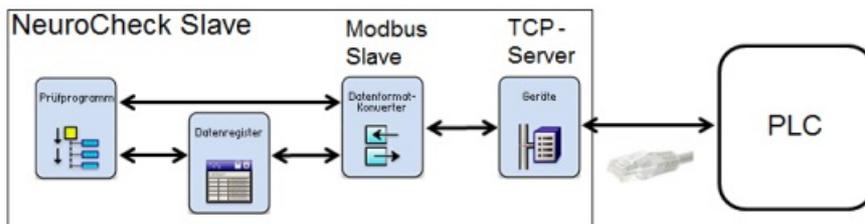
Modbus Master

If you need a Modbus master configuration to control an I/O device, add a TCP-Client to the device manager and then a Modbus DFC. Used for bus coupler and I/O module.



Modbus Slave

If you need a Modbus slave configuration get controlled by a PLC, add a TCP-Server to the device manager and then a Modbus DFC.



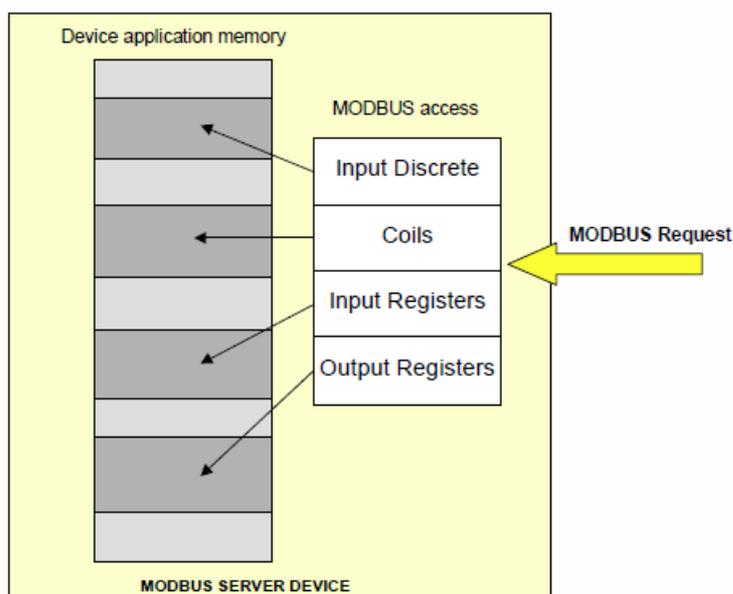
Data Model

Modbus bases its data model on a series of tables that have distinguishing characteristics.

The four primary tables are:

Primary table	Object type	Type of	Available address	Description
Discrete Input	Single bit	Read-Only	0-65535	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	0-65535	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	0-65535	This type of data can be provided by an I/O system.
Holding Registers	16-bit word	Read-Write	0-65535	This type of data can be alterable by an application program.

The implemented Modbus data model in the NeuroCheck Modbus DFC slave configuration is block separated.



4-Byte Integer

To send and receive larger values than 16-bit, the DFC allows to use two Modbus registers for 4 byte integer values. See chapter [Pin Configuration Dialog](#) for further details. In case of 4 byte integer values the Modbus DFC uses **Big-Endian word order**.

4-Byte Float

To send and receive floating point values, the DFC allows to use two Modbus registers. See chapter [Pin Configuration Dialog](#) for further details. In case of 4 byte floating point values the Modbus DFC uses **Big-Endian word order**.

Strings

To send and receive data strings, the DFC allows to use up to 123 Modbus registers for a NeuroCheck pin of type string. The string value from the NeuroCheck data register is written into the Modbus registers in the same order. If the string has an odd number of characters the last 8 bit of the last Modbus register will be 0x00.

Single bit

To send and receive boolean signals from a modbus device that does not support Discrete Inputs and Coils you can map a modbus 16-bit register (Input Register, Holding register) to 16 individual boolean data pins. See [Pin Configuration Dialog](#) for details.

Supported Function Codes

NeuroCheck as Modbus Slave

Running NeuroCheck as Modbus slave, the DFC supports the following Modbus Function Codes:

Function Code	Description
01 (0x01)	Read Coils
02 (0x02)	Read Discrete Inputs
03 (0x03)	Read Holding Registers
04 (0x04)	Read Input Registers
05 (0x05)	Write Single Coil
06 (0x06)	Write Single Register
15 (0x0F)	Write Multiple Coils
16 (0x10)	Write Multiple Registers
23 (0x17)	Read/Write Multiple Registers

NeuroCheck as Modbus Master

Running NeuroCheck as Modbus Master, the DFC uses the following function codes for communication. Make sure the slave devices supports these function codes.

Function Code	Usage
02 - Read Discrete Inputs	The function code is used for input pins with data type: (Data) Boolean, (Process) Signal and (Process) Free Signal. The master will send a Modbus request for every pin separately.
04 - Read Input Registers	The function code is used for input pins with data type: (Data) Integer, (Data) Float, (Data) String and (Process) Identifier. The master will send a Modbus request for every pin separately.
05 - Write Single Coil	The function code is used for output pins with data type: (Data) Boolean, (Process) Signal and (Process) Free Signal.
06 - Write Single Register	The function code is used for output pins with data type: (Data) Integer, (Data) Float, (Data) String and (Process) Identifier.
15 - Write Multiple Coils	The function code is used for output pins with data type: (Data) Boolean, (Process) Signal and (Process) Free Signal.
16 - Write Multiple Registers	The function code is used for output pins with data type: (Data) Integer, (Data) Float, (Data) String and (Process) Identifier.

Message Description

This category describes a message created by the DFC, a so called Modbus request. The examples show the request in hexadecimal format. Remember that the Modbus protocol uses the big endian byte order.

Modbus TCP Request - Read

The following example shows the Modbus request for reading one discrete input with the Modbus address 8000 (decimal).

Request	00	00	00	00	00	06	00	02	1F	40	00	01
	Transaction ID		Protocol ID		Message Length		Unit ID	Function Code	Start Address		Data (Quantity)	

Transaction ID: This is a unique ID for each message sent by the master. Typically it will always be zero because the master only sends one request at the same time. The transaction IDs are handled like a list. Every time the response has the same transaction ID as the request it is available again. If a second request has to be sent before the response is received by the master the transaction ID will be incremented.

Protocol ID: These two bytes will always be 0 in the Modbus TCP protocol.

Message length: These two bytes indicates the length of the remaining message, not the complete message length.

Unit ID: This ID is for device identification. The NeuroCheck Modbus Master will always use the ID 0 because the identification in the Modbus TCP protocol is part of the Network Protocol, which can be set by the IP Address in the hardware device.

Function code: This is the specific code for the write or read action for the slave device. In this example the code is 0x02 = 2 (decimal) which means that the master wants to read the status of one or more discrete inputs in the slave device.

Start address: The function code 2 (decimal) needs a starting address which is set to 0x1F40 in this example.

Data (quantity): This is the number of discrete inputs the master wants to read. In this case only one discrete input will be read.

Modbus TCP Request - Write

This example shows the request to set the output pin 9000 (decimal) to true.

Request	00	00	00	00	00	06	00	05	23	28	FF	00
	Transaction ID		Protocol ID		Message Length		Unit ID	Function Code	Address		Data (Value)	

Function code: In this example the code is 0x05 = 5 (decimal) which means that the master wants to write a single coil in the slave device.

Start address: The function code 5 (decimal) needs the address of the coil which will be set.

Data (value): The value for the coil. In this case the 'output pin' is set to 'true' (0xFF 0x00).

All Modbus TCP requests have the same length of the message header (identification for the Modbus protocol). After the "Function Code", all Modbus requests differ. For further information about function code specific messages and their response, have a look at the official 'Modbus Application Protocol' documentation on www.modbus.org.

Converter Properties Dialog: Device

This page provides information about the device the converter is connected with.

Element	Description
Name	Indication about the Modbus configuration (Master or Slave) and the name of the connected device.
Index	The index of the device within the list of devices of the category Serial interface and Ethernet.

Converter Properties Dialog: Protocol

This page provides protocol specific properties. The properties are available for Modbus master configurations only.

Element	Description
Timeout	Time in milliseconds to wait for a Modbus answer after the Modbus request is sent.

Converter Properties Dialog: Input pins

You can specify the properties of the input pins.

Element	Description
Input pins	<p>List of all defined input pins. Select an entry for editing. The columns of the table are:</p> <ul style="list-style-type: none"> • Name: The name of the pin for identification in NeuroCheck. • Type: The type of data that can be exchanged via this pin. • Modbus address: The used Modbus address of the pin. Modbus address must be unique in its primary table (s. Data Model). • Modbus data type: The Modbus data type from one of the four primary tables (s. Data Model). This property can not be selected. It is set automatically because it depends on the Modbus configuration (master or slave), communication direction (pin input or output) and pin type.
Add ...	Click here to define a new pin in the Pin Configuration Dialog and append it to the list.
Edit ...	Click here to open the Pin Configuration Dialog to edit the currently selected pin.
Remove	Click here to remove the selected pin from the list.
Duplicate	Click here to duplicate the current selected pin and open its configuration in the Pin Configuration Dialog . The Modbus data address will be incremented automatically.
Import...	Click here to open the dialog with the selection of predefined pin configurations. Importing a pin configuration will replace all configured input and output pins.
Up	Click here to swap the currently selected pin with the pin above.
Down	Click here to swap the currently selected pin with the pin below.



If the Modbus DFC is configured as Modbus Master, a context menu will be available for selected pins. This context menu opens a dialog that shows the Modbus request this pin will create when it is set in NeuroCheck. To send the request for the selected pin, copy the string (displayed in hexadecimal) from the dialog and paste it in the NeuroCheck Serial Test Communication dialog. It is not possible to send Modbus request directly from the data format converter.

Converter Properties Dialog: Output pins

You can specify the properties of the output pins.

Element	Description
Output pins	<p>List of all defined output pins. Select an entry for editing. The columns of the table are:</p> <ul style="list-style-type: none"> • Name: The name of the pin for identification in NeuroCheck. • Type: The type of data that can be exchanged via this pin. • Modbus address: The used Modbus address of the pin. Modbus address must be unique in its primary table (s. Data Model). • Modbus data type: The Modbus data type from one of the four primary tables (s. Data Model). This property can not be selected. It is set automatically because it depends on the Modbus configuration (master or slave), communication direction (pin input or output) and pin type.
Add ...	Click here to define a new pin in the Pin Configuration Dialog and append it to the list.
Edit ...	Click here to open the Pin Configuration Dialog to edit the currently selected pin.
Remove	Click here to remove the selected pin from the list.
Duplicate	Click here to duplicate the current selected pin and open its configuration in the Pin Configuration Dialog . The Modbus data address will be incremented automatically.
Import...	Click here to open the dialog with the selection of predefined pin configurations. Importing a pin configuration will replace all configured input and output pins.
Up	Click here to swap the currently selected pin with the pin above.
Down	Click here to swap the currently selected pin with the pin below.



If the Modbus DFC is configured as Modbus Master, a context menu will be available for selected pins. This context menu opens a dialog that shows the Modbus request this pin will create when it is set in NeuroCheck. To send the request for the selected pin, copy the string (displayed in hexadecimal) from the dialog and paste it in the NeuroCheck Serial Test Communication dialog. It is not possible to send Modbus request directly from the data format converter.

Converter Properties Dialog: Input pin reset mode

This page specifies the property for the input pin reset mode.

Element	Description
Reset all pins	Resets the input pin after the value is read by NeuroCheck. That means, that an input pin will return its value only when the pin received new data from the connected device. If the pin is read without received data, the pin will return an error.
Reset '(Process) Signals' only	This mode resets only the configured input pins with the data type (Process) Signal after the pin is read by NeuroCheck. The received data in the other pins will not be reset.
Do not reset input pins	In this mode, the configured input pins are not reset after they are read by NeuroCheck. The connected device has to reset the signals and data pin values.

Remarks:

The configured input pins are reset after changing the reset mode and closing the properties dialog with the OK button.

Converter Properties Dialog: Format Options

This page configures the format options.

Element	Description
Replacement	Select or deselect to replace the "\0" character in a input string.
Replace character	Choose the replacing character.

Pin Configuration Dialog

☑ Screenshot of Properties Dialog

This dialog specifies the properties of a pin. It is used for the creation of a new pin as well as for editing an existing pin. The dialog is opened by clicking the buttons Add ..., Edit ... or Duplicate on the pages [Input pins](#) or [Output pins](#).

Element	Description
Name	The name of the pin for identification in NeuroCheck.
Data type	Specify here the type of data that can be exchanged via this pin. Possible values are: <ul style="list-style-type: none"> • (Data) Boolean: The pin is used to exchange a Boolean value as data. • (Data) Integer: The pin is used to exchange an integer value as data. • (Data) Float: The pin is used to exchange a floating point value as data. • (Data) String: The pin is used to exchange a string value as data. • (Process) Signal: The pin is used to exchange a Boolean value for remote control. • (Process) Identifier: The pin is used to exchange an integer value for remote control. • (Process) Free signal: The pin is used to exchange a Boolean value used by I/O check functions.
Modbus data type	The Modbus data type from one of the four primary tables (s. Data Model). Normally this property is read-only and not be selected. It is set automatically because it depends on the Modbus configuration (master or slave), communication direction (pin input or output) and pin type. Only in case of boolean pin data types ((Data) Boolean, (Process) Signal, (Process) Free signal) you can configure a regular Modbus bit (Discrete Input, Coil) or a bit in a 16-bit Modbus register. If you map a 16-bit Modbus register to a boolean pin data types you have to specify the bit index. Of course you can configure 16 boolean pins at a single Modbus address for a 16-bit modbus register.
Bit index	Index of bit in a 16-bit modbus register for boolean pin data types ((Data) Boolean, (Process) Signal, (Process) Free signal). See Modbus data type above for more information.
Start address	0-65535 (must be unique in its Modbus data type).

Element	Description
Data size	<p>The number of needed Modbus addresses depends on the data size of the pin. For the different pin data types following sizes are configurable:</p> <ul style="list-style-type: none">• (Data) Boolean: 1 bit.• (Data) Integer: 2 or 4 byte.• (Data) Float: 4 byte.• (Data) String: 2 to 246 byte with 2 byte step size. Each character has 1 byte.• (Process) Signal: 1 bit.• (Process) Identifier: 2 or 4 byte.• (Process) Free signal: 1 bit.

Info Dialog

This dialog displays information about the NeuroCheck data format converter.

Element	Description
Description	The description of the data format converter
File	The name of the file that provides the data format converter
Version	The version of the data format converter
Copyright	The copyright of the data format converter

Support Services

For technical support, please contact your local NeuroCheck partner or NeuroCheck GmbH:

Phone: +49 (0) 7146 - 89 56-40

E-Mail: support@neurocheck.com

Web: www.neurocheck.com

Before contacting us, please provide some important information about your system:

- **Information about your NeuroCheck installation and your PC setup:**

Use the NeuroCheck Diagnostics tool to check your installation and computer configuration.

The NeuroCheck Diagnostics is installed in the "Tools" folder within your NeuroCheck installation.

- **Log file information:**

Logging for NeuroCheck can be activated in **System > Software Settings > Diagnosis > Logging**.

