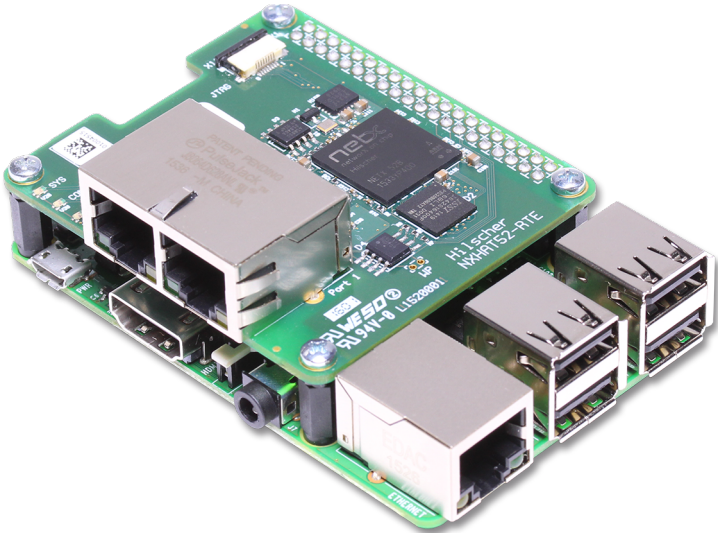
User manual

netHAT

NXHAT52-RTE



Hilscher Gesellschaft für Systemautomation mbH
www.hilscher.com

DOC160610UM01EN | Revision 1 | English | 2016-11 | Released | Public

Table of contents

1 Introduction

1.1 About netHAT

The netHAT module developed by Hilscher allows you real-time communication for the Raspberry Pi® and can thus be operated as a device in all current Real-Time Ethernet networks, e.g. PROFINET, EtherNet/IP or EtherCAT.

The protocol stacks are preconfigured with 32 byte input and 32 byte output data. For communication between Pi and the PLC, Hilscher provides everything you need:

- documentation,
- device description files,
- driver,
- firmware and
- application example.

The NXHAT52-RTE is designed as a HAT (**H**ardware **a**ttached on **t**op), so: Simply plug on the module and communicate!

1.2 Technical introduction

The netHAT contains the Hilscher netX 52 multiprotocol controller. The netX controller includes an ARM 966 CPU for protocol execution and a Hilscher designed controller subsystem for multiprotocol Hardware access, in combination just load the firmware to change from one protocol to the other.

The netX is communicating over a 25 MHz SPI signal located at the HAT socket with the Raspberry PI. The software package for the Raspberry PI includes the Hilscher **cifX Device Driver** for Linux, for communication with the netHAT and configuration files for PROFINET IO Device, EtherCAT Slave or EtherNet/IP Adapter. A simple demo application for easy I/O data exchange is included in the scope of delivery.

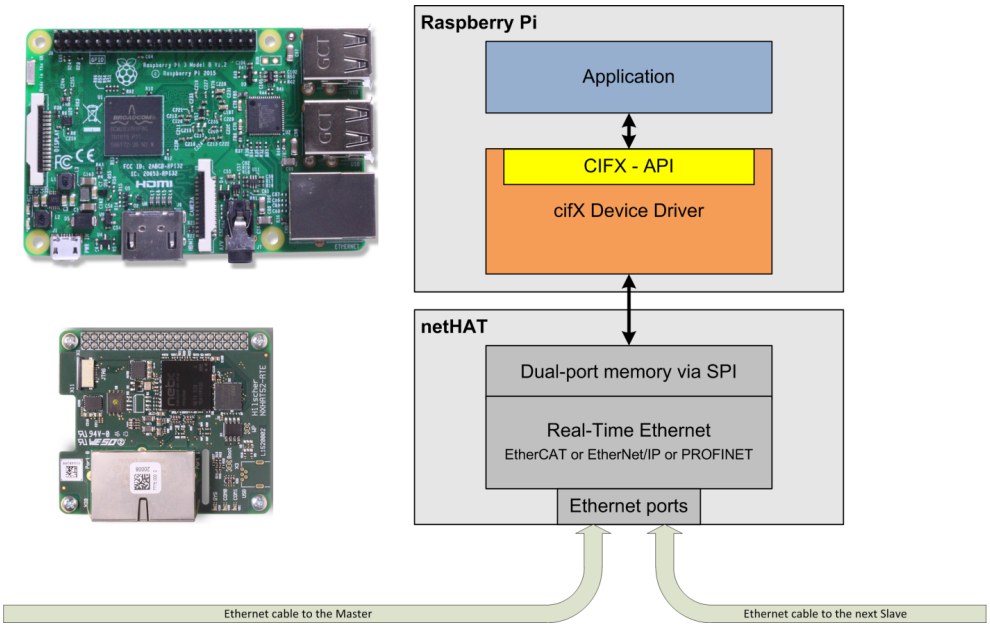


Figure 1: Raspberry Pi and netHAT

1.3 Legal notes

This hardware and software is intellectual property of the Hilscher Gesellschaft für Systemautomation mbH. It can be used for internal development, modification and debugging purpose.

Hilscher hereby grants you a non-sublicensable, non-exclusive, irrevocable license to exercise the hard- and software for non-commercial purposes only. You may not exercise any of the rights granted to you in any manner that is primarily intended for commercial advantage.

The use in productive facilities is strongly prohibited! In no event will Hilscher be liable to you on any legal theory for any special, incidental, consequential, punitive or exemplary damages arising out of the use in productive facilities.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software. Any liability is excluded.

2 netHAT module

2.1 Dimensions

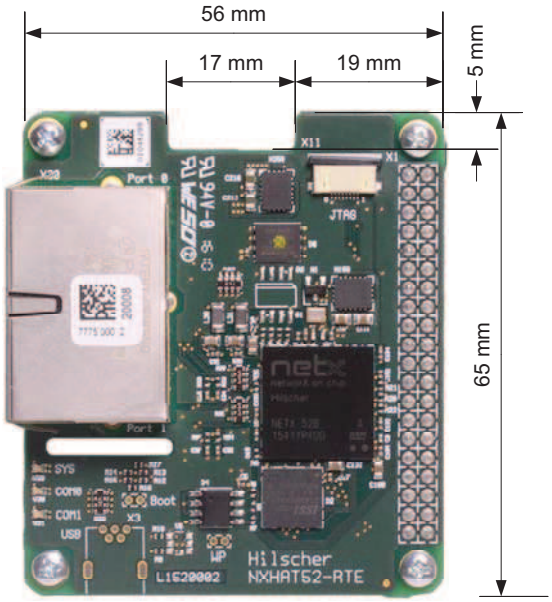
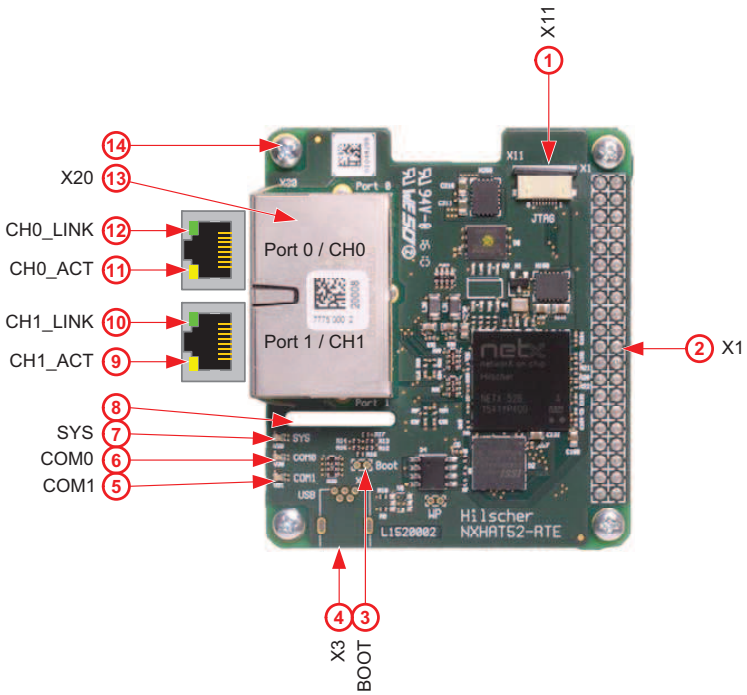


Figure 2: NXHAT 52-RTE dimensions

2.2 Interfaces and device drawing



(1)	JTAG interface (X11) For production purposes at Hilscher only.
(2)	Host interface connector (X1): 40-pin connector
(3)	Boot
(4)	Reserved for future USB interface (X3)
(5)	Communication status LED (COM1)
(6)	Communication status LED (COM0)
(7)	System status LED (SYS)
(8)	Slot to connect a cable to the Raspberry Pi.
(9)	Activity LED port 1 / channel 1
(10)	Link LED port 1 / channel 1
(11)	Activity LED port 0 / channel 0
(12)	Link LED port 0 / channel 0
(13)	2-port Ethernet (2xRJ45) For EtherCAT Slave: Port 0 = in Port 1 = out
(14)	Hole for screw mounting (4 mounting holes in total)

Table 1: netHAT interfaces and positions

2.3 Power supply

The Raspberry Pi supplys power to the netHAT via the host interface connector (X1). Therefore, you have to use a power supply that provides enough power to the Raspberry Pi and all connected equipment like e.g. keyboard. The netHAT module requires 300 mA@5 V.

2.4 Host interface (X1)

Pin	Signal	Description
1	n.c.	-
2	5V	Power supply
3	n.c.	-
4	5V	Power supply
5	n.c.	-
6	GND	Ground
7	n.c.	-
8	n.c.	-
9	GND	Ground
10	n.c.	-
11	n.c.	-
12	n.c.	-
13	n.c.	-
14	GND	Ground
15	n.c.	-
16	n.c.	-
17	n.c.	-
18	SPM_DIRQ#	Serial dual-port memory: Data IRQ
19	SPM_MOSI	Serial dual-port memory: SPI Master out slave in
20	GND	Ground
21	SPM_MISO	Serial dual-port memory: SPI master in slave out
22	SPM_SIRQ#	Serial dual-port memory: Synchron IRQ
23	SPM_CLK	Serial dual-port memory: SPI clock
24	SPM_CS#	Serial dual-port memory: SPI chip select
25	GND	Ground
26	n.c.	-
27	ID_SD	
28	ID_SC	
29	n.c.	-
30	GND	Ground
31	n.c.	-
32	n.c.	-
33	n.c.	-
34	GND	Ground
35	n.c.	-
36	n.c.	-
37	n.c.	-
38	n.c.	-
39	GND	Ground
40	n.c.	-

Table 2: Pin assignment of host interface (X1)

2.5 LED

The netHAT module has 3 dual-color LEDs: One system status LED (SYS) and two bus-specific LEDs (COM0 and COM1).

System status LED (SYS)

The system status LED **SYS** can assume the states described below.

LED	Color	State	Meaning
SYS	**Duo LED yellow/green**		
	(green)	On	Operating System running
	(green / yellow)	Blinking	Second stage bootloader is waiting for firmware.
	(yellow)	On	Bootloader netX (= romloader) is waiting for second stage bootloader.
	(off)	Off	Power supply for the device is missing or hardware defect.

Table 3: System status LED states

LED state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Blinking	The indicator turns on and off cyclically.

Table 4: System LED state definitions

Bus-specific LEDs (COM0 and COM1)

- EtherCAT Slave: see section *LEDs EtherCAT Slave* [▷ page 20]
- EtherNet/IP Adapter: see section *LEDs Ethernet/IP Adapter* [▷ page 26]
- PROFINET IO Device: see section *LEDs PROFINET IO Device* [▷ page 30]

2.6 Technical data

NXHAT42-RE	Parameter	Value
Product	Part number	7775.000
	Brief description	netHAT - netX Raspberry Pi® Module
	Compatible with	Raspberry Pi®3, Pi®2, Zero®
	Module standard	HAT standard for Raspberry Pi®
Communication controller	Type	netX 52
Integrated Memory	RAM	8 MB SDRAM
	Flash	4 MB Quad SPI Flash
Power supply	Supply voltage	5 V DC
	Current	300 mA
	Connector	Via host interface connector
Host interface	Type	Serial dual-port memory interface via SPI
	Baud rate	Max. 50 MHz
	Connector	40-pin
Communication interface	Communication standard	Ethernet
	Interface type	100BASE-TX
	Connector	2 x RJ45
LED indicators	LEDs	SYS System status
		COM0, COM1 Communication status
Permitted ambient conditions	Operating temperature range	0 ... +50°C
Device	Dimensions (L x W x H)	65 mm x 57 mm x 20 mm
	Mounting	Mounted on Raspberry Pi

Table 5: Technical data NXHAT52-RE

3 How to use the netHAT

3.1 Overview

The netHAT ZIP file contains the driver package, several firmware packages, EDS files, documentations, and the application example. You can download the ZIP file from www.nethat.net. The name of the ZIP file is `netHAT_DVD_2016-08-1_V1_0_0_0.zip`.

Folder	Contents
Documentation	Documentations in PDF format
Driver	Driver package
EDS	Device description files for the configuration of the master
Example	Application example including the I/O data exchange
Firmware	Firmware packages

Table 6: netHAT ZIP file overview

The following figure shows an overview about the packages to be installed and executable to be used for testing. The following sections describe the installation and test.

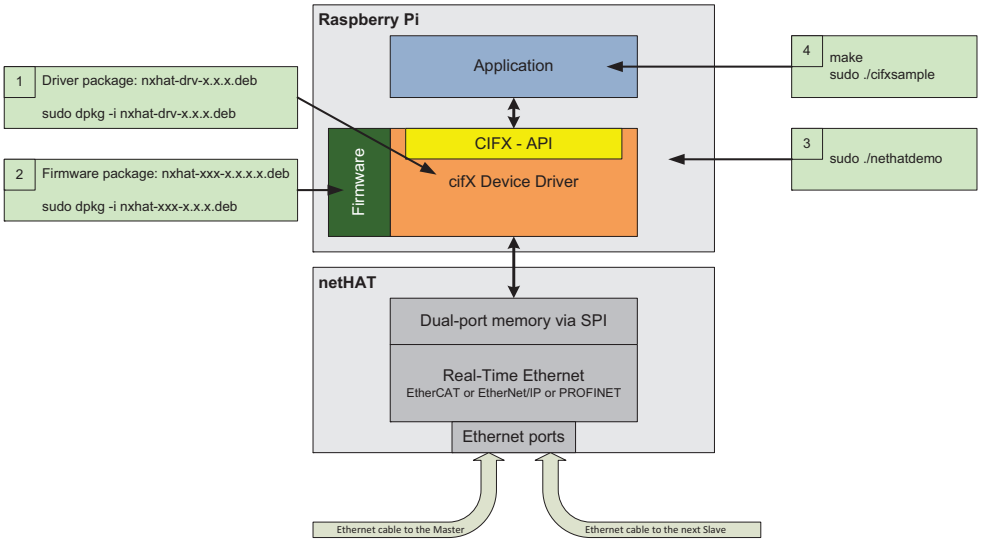


Figure 3: netHAT overview

3.2 Information and updates

The following table lists web addresses where you can get information on and updates for the Raspberry Pi and the netHAT module.

Web address	This site offers you
https://www.raspberrypi.org/	Information, blog, downloads, community, help, forums, and education
http://www.nethat.net/	Firmware, driver, application example, documentations, device description files, updates, forum, and FAQs

Table 7: Information and updates

3.3 Installing the netHAT Raspberry Pi package

Prerequisites

- Raspberry Pi 2 or 3
- Power supply for Raspberry Pi AND any connected equipment
- Raspbian installed on Raspberry Pi
- netHAT NXHAT52-RTE mounted on Raspberry Pi
- Monitor, keyboard, mouse and USB stick (Option 1) or
  Ethernet connection from a PC to Raspberry Pi (Option 2)
- To access your Raspberry Pi, you need to know the user name and password
- To use an Ethernet connection, you need to know the IP address of your Raspberry Pi

Installing packages

Installation of the **driver package** and one **firmware package**.

1. Get the ZIP file:
   - ➢ Download the netHAT ZIP file containing the packages from www.nethat.net and extract the ZIP file.
2. Transfer the packages to the Rasberry Pi:
   - ➢ If you use a Raspberry Pi with monitor, keyboard and mouse, copy the packages on a USB stick, connect the USB stick to the Raspberry Pi, mount the USB stick and copy the packages into a folder (e.g. firmware) of your home directory.
     or
     if you use an Ethernet connection from a PC to your Raspberry Pi, transfer the files from your PC with an SFTP program (e.g. SCP or WinSCP) into a folder (e.g. firmware) of the home directory.
3. Install the **driver package**:
   - ➢ Change into the folder containing the driver package.
   - ➢ Always install the driver `nxhat-drv-x.x.x.deb` (where x.x.x is the version):

```
sudo dpkg -i nxhat-drv-x.x.x.deb
```

4. Install the **firmware package**:
   - ➢ Change into the folder containing the firmware packages.
   - ➢ Install one firmware, e.g. `nxhat-pns-x.x.x.x.deb` (where x.x.x.x is the version):

```
sudo dpkg -i nxhat-pns-x.x.x.x.deb
```

Firmware packages

The following table lists the available firmware packages.

Package	Real-Time Ethernet firmware in the package
nxhat-ecs-x.x.x.x.deb	EtherCAT Slave
nxhat-eis-x.x.x.x.deb	EtherNet/IP Adapter
nxhat-pns-x.x.x.x.deb	PROFINET IO Device

Table 8: Firmware packages

Testing access to the netHAT module

After you have installed the **driver package** and one **firmware package**, you can test if the Raspberry Pi has access to the netHAT module.

➢ Open the demo folder:

```
cd /opt/cifx/demo/
```

➢ Start the demo netHAT demo application:

```
sudo ./nethatdemo
```

This starts the netHAT demo application. The purpose of this application is to check whether the Raspberry Pi can access the netHAT module. This demo application displays the information read from the netHAT module including the name and version of the loaded firmware.

If the Raspberry Pi has access to the netHAT module and the I/O communication is established the demo application displays “`Start read/write IO-Data!`“ at the bottom of the output. To abort the netHAT demo application, hit any key.

If the Raspberry Pi has access to the netHAT module but the I/O communication is not established, the demo application displays “`error 0x800C0021`” at the bottom of the output. Error 0x800C0021 indicates that no cyclic I/O communication is established. To remove this error check that

- the Ethernet cable is connected to the netHAT module
- the master device is configured properly to communicate with the netHAT module
- the right firmware is loaded

3.4 Using the example application

The netHAT ZIP file contains an application example. This application example reads 32 bytes that the netHAT module has received from the PLC and displays this data. This application example writes 32 bytes (incremented once a second) to the netHAT module that sends this data to the PLC.

Prerequisite

The example application is provided as in C source code. A C compiler or a tool chain is required on the Raspberry Pi with that you can create the executable application.

Installing and running the application example

- In your home directory, create a folder, name it netHAT, e.g., and change into this folder.
- Copy all files of the example application into this folder.
- Compile the example application. Enter

```
make
```

- Start the example application

```
sudo ./cifxsample
```

The example application displays the driver version, the device number (`7775000`), the loaded firmware and version and other information read from the netHAT module. Then the example application continuously displays approx. every second the first 32 bytes of the input and output data. To stop the example application, press Ctrl+C.

```
---------- Display Driver Version ----------
Driver Version: LinuxCIFXDrv V1.1.0, based on cifX Toolkit 1.2.0.0
State = 0x00000000
------------------------------------------------------
---------- Communication Channel demo ----------
Communication Channel Info:
Device Number : 7775000
Serial Number : 20025
Firmware : PROFINET IO Device
FW Version : 4.2.7 build 0
FW Date : 06/13/2016
Mailbox Size : 1596
IO Read Data:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
IO Write Data:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
IO Read Data:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
IO Write Data:
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
```

As soon as the PLC sends data to the netHAT module, the data is displayed below `IO Read Data` so you can check it.

3.5 Changing protocol firmware and configuration

Changing the protocol firmware and the configuration is very easy: Install the firmware package containing the protocol you want to use (see list in table *Firmware packages* [page 12]). Install the package using (replace x.x.x.x with the version of your package):

```
sudo dpkg -i nxhat-eis-x.x.x.x.deb
```

Each firmware package replaces the other firmware package.

Start the application program, e.g. the demo application. This causes the driver to detect the firmware change and to download the firmware into the NXHAT52-RTE automatically. The download usually takes 10-20 s.

For advanced users

The Linux driver compares the firmware of the NXHAT52-RE with that of the Raspberry PI in the folder `/opt/cifX/deviceconfig/FW/channel0`. In case of a difference, the firmware in the folder on the Raspberry PI will automatically be downloaded into the NXHAT52-RE.

Additionally, the folder `/opt/cifX/deviceconfig/FW/channel0` contains the default configuration that is installed with the firmware package. A working communication is only guaranteed with the default configuration. Changing configuration files requires technical knowledge.

You must not change any file in this folder. Only change a file if you know what you do.

4 EtherCAT

4.1 Technical data

Feature	Value
Communication	EtherCAT Slave
I/O data	32 byte input 32 bytes output
Functions	Cyclic process data
XML file for the configuration tool of the EtherCAT Master	Hilscher NXHAT52-RTE ECS V4.X.X.xml

Table 9: EtherNet/IP Adapter - Technical data

4.2 Configuring the EtherCAT Master

This description contains hints for using Beckhoff's TwinCAT V2 or V3 software on a PC.

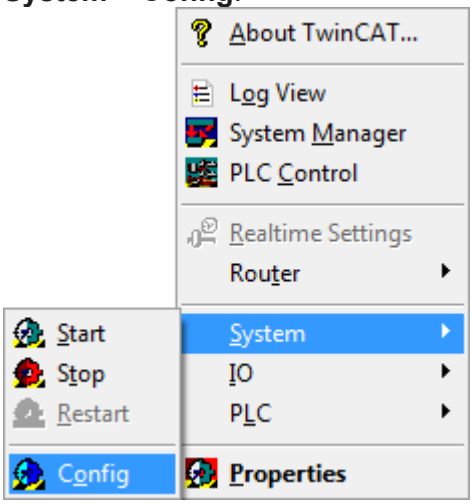
Prerequisits

1. TwinCAT is installed on your PC
2. Raspberry Pi and the netHAT module are prepared for EtherCAT
3. You already have connected an Ethernet cable from the EtherCAT Master to the **IN port** of the netHAT module.

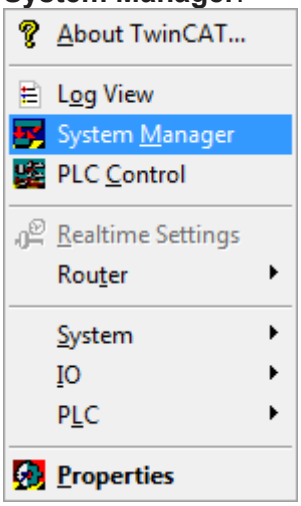
TwinCAT in Config mode

Set TwinCAT into the **Config Mode**.

➢ Right-click the TwinCAT icon in the notification area and click **System** > **Config**.



➢ Right-click the TwinCAT icon in the notification area and click **System Manager**.



⇨ The **TwinCAT System Manager** window opens.

⇨ The status bar indicates Config Mode.

Adding netHAT to TwinCAT

➢ In TwinCAT, select menu **Actions** > **Import XML Description**.

↬ A file selection dialog opens.

➢ Select the file `Hilscher NXHAT52-RTE ECS V4.X.X.xml`.

➢ Click **Open**.

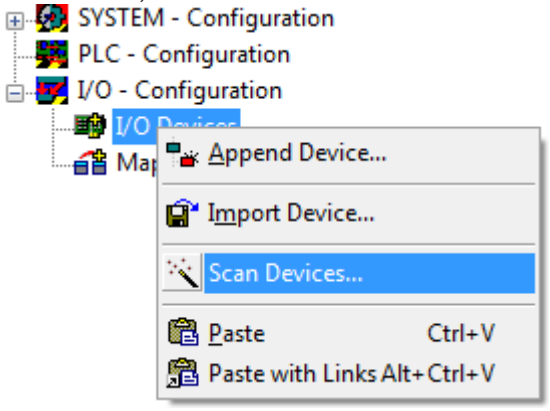
⇨ The netHAT device is added to the device list of TwinCAT.

Configuring the EtherCAT Master

1. Create a project:
   ➢ In TwinCAT, create a new project: Select menu **File** > **New**.
2. Verify and set an Ethernet adapter for the EtherCAT network.
   ➢ In the **TwinCAT System Manager** window, select menu **Options** > **Show Real time Ethernet Compatible Devices**.
   ➢ Verify that an Ethernet adapter appears below **Installed and ready to use devices**. If this is the case, you can close this window and continue with step 3.
   ➢ If no Ethernet Adapter appears below **Installed and ready to use devices**, which is for example the case if TwinCAT is installed the first time on your PC, then you have to install an Ethernet Adapter first.
   ➢ In this case, select a device from **Compatible devices**, e.g. **Local Area Connection**.
   ➢ Click **Install**.
   ➢ Then **Installed and ready to use devices** has an Ethernet adapter e.g. **Local Area Connection**.
   ➢ Close this window.
   ↬ TwinCAT now is prepared.

3. Adding devices to the EtherCAT network

➢ In the **TwinCAT System Manager** window navigation area, right-click **I/O Devices**, then click **Scan Devices**.



↳ The **TwinCAT System Manager** displays **HINT: Not all types of devices can be found automatically**.

➢ Click **Ok**.

↳ The **I/O devices found** dialog box opens.

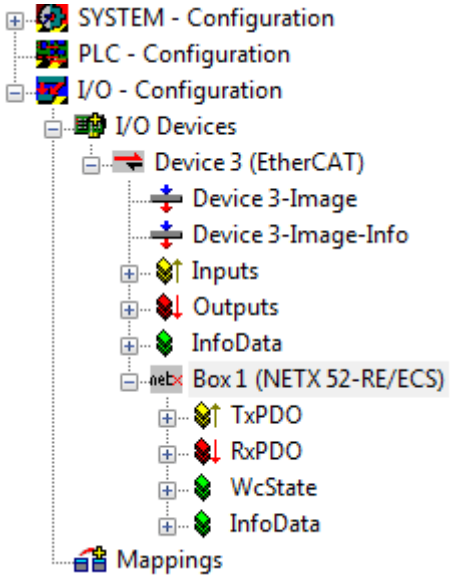
➢ Select the Ethernet connection and click **Ok**.

↳ The **TwinCAT System Manager** displays the dialog box **Scan for boxes**.

➢ Click **Yes**.

↳ TwinCAT scans for boxes.

↳ If TwinCAT finds the netHAT module on EtherCAT network, TwinCAT adds the netHAT module as box.



↳ TwinCAT asks **Activate Free Run**?

➢ Click **Yes**.

⇨ The configuration is prepared.

Testing I/O data

Start example application:

➢ On the Raspberry Pi, start the example application.

```
sudo ./cifxsample
```

Set TwinCAT to Free Run mode:

➢ If TwinCAT is in Config Mode, set TwinCAT to Free Run. Keyboard shortcut Ctrl+F5 toggles between Config Mode and Free Run mode.

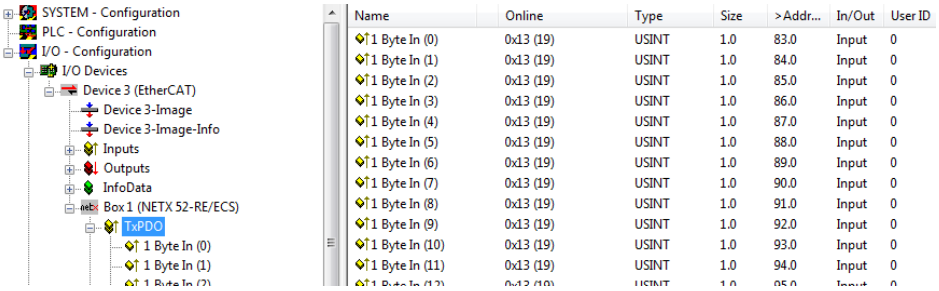
Sending data from the Raspberry Pi to the Master:

The example application increments automatically the 32 bytes send to the master approx. each second. The example application displays below “IO Write Data” the current data written to the netHAT module:

```
IO Write Data:
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
```

➢ In TwinCAT, click TxPDO below the box of the netHAT module.

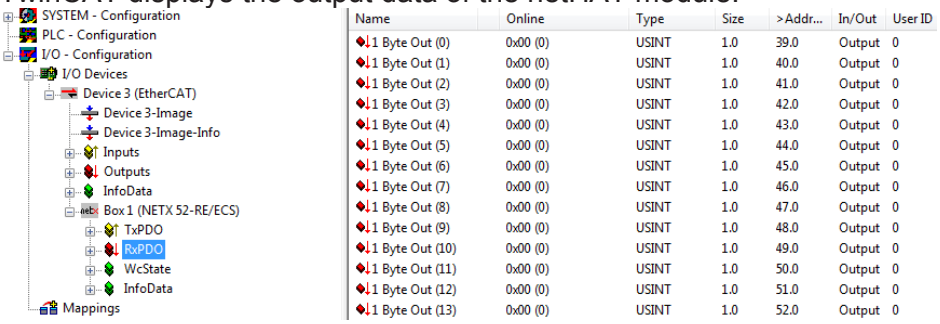
➢ Data received from the netHAT module is displayed in the column Online.



Data from the master to the Raspberry Pi:

➢ In TwinCAT, click RxPDO below the box of the netHAT module.

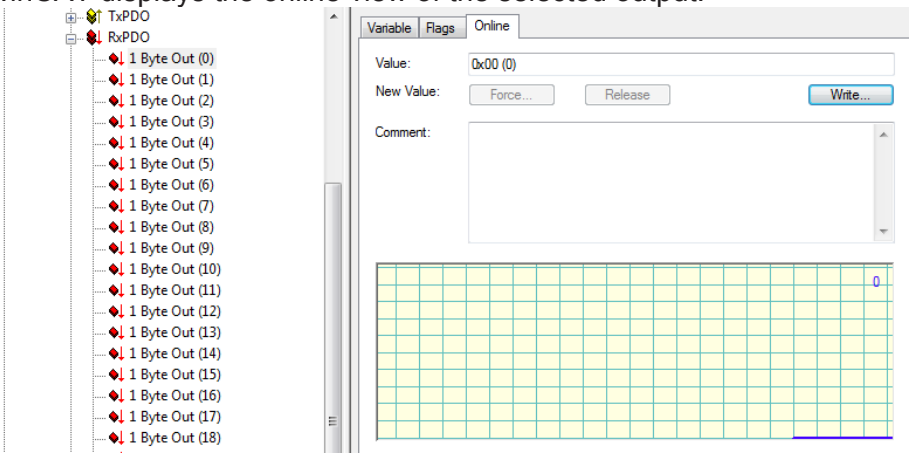
➢ TwinCAT displays the output data of the netHAT module.



➢ Double click an output byte, e.g. **1 Byte out (0)**.

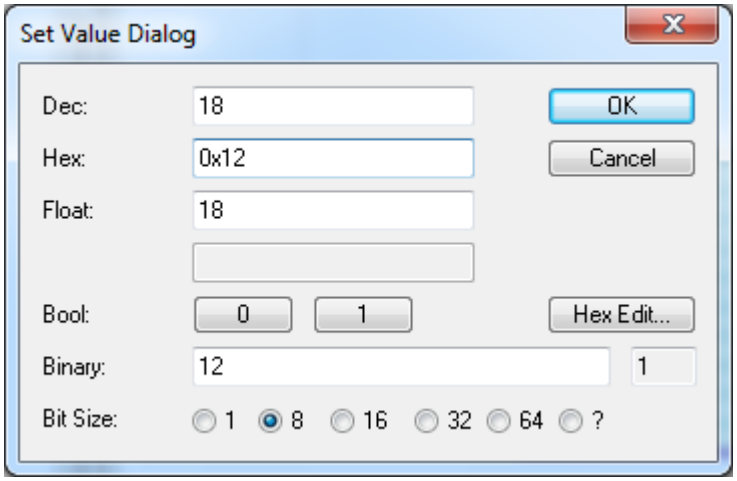
➢ Click on **Online** tab.

➢ TwinCAT displays the online view of the selected output.



➢ Click **Write**.

➢ Enter a value.



➢ Click **Ok**.

➢ The example application displays below “IO Read Data” the current data received from the netHAT module.

IO Read Data:

```
12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

4.3 LEDs EtherCAT Slave

LED	Color	State	Meaning
RUN Position in the device drawing: (6)	**Duo-LED red/green**		
	(off)	Off	**INIT:** The device is in INIT state.
	(green)	Blinking (2.5 Hz)	**PRE-OPERATIONAL:** The device is in PRE-OPERATIONAL state.
	(green)	Single flash	**SAFE-OPERATIONAL:** The device is in SAFE-OPERATIONAL state.
	(green)	On	**OPERATIONAL:** The device is in the OPERATIONAL state.
ERR Position in the device drawing: (5)	**Duo-LED red/green**		
	(off)	Off	**No error:** The EtherCAT communication of the device is in working condition.
	(red)	Blinking (2.5 Hz)	**Invalid configuration:** General Configuration Error Possible reason: State change commanded by master is impossible due to register or object settings.
	(red)	Single flash	**Local error:** Slave device application has changed the EtherCAT state autonomously. Possible reason 1: A host watchdog timeout has occurred. Possible reason 2: Synchronization Error, device enters Safe-Operational automatically.
	(red)	Double flash	**Application watchdog timeout:** An application watchdog timeout has occurred. Possible reason: Sync Manager Watchdog timeout.
L/A IN, **L/A OUT** Ch0 (12), Ch1 (10)	**LED green**		
	(green)	On	**Link:** The device is linked to the Ethernet, but does not send/ receive Ethernet frames.
	(green)	Flickering (load dependant)	**Activity:** The device is linked to the Ethernet and sends/receives Ethernet frames.
	(off)	Off	The device has no link to the Ethernet.
Ch0, Ch1	**LED yellow**		
	(off)	Off	This LED is not used.

Table 10: LED states for the EtherCAT Slave protocol

LED state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Blinking (2.5 Hz)	The indicator turns on and off with a frequency of 2.5 Hz: “on” for 200 ms, followed by “off” for 200 ms.
Single flash	The indicator shows one short flash (200 ms) followed by a long “off“ phase (1,000 ms).
Double flash	The indicator shows a sequence of two short flashes (each 200 ms), separated by a short off phase (200 ms). The sequence is finished by a long off phase (1,000 ms).
Flickering (load dependent)	The indicator turns on and off with a frequency of approximately 10 Hz to indicate high Ethernet activity: on for approximately 50 ms, followed by off for 50 ms. The indicator turns on and off in irregular intervals to indicate low Ethernet activity.

Table 11: LED state definitions for the EtherCAT Slave protocol

For positions in the device drawing, see section *Interfaces and device drawing* [page 6].

5 EtherNet/IP

5.1 Technical data

Feature	Value
Communication	EtherNet/IP Adapter
I/O data	32 byte input 32 bytes output
Functions	Cyclic process data
EDS file for the configuration tool of the EtherNet/IP Scanner	HILSCHER NXHAT52-RTE EIS V1.1.EDS

Table 12: EtherNet/IP Adapter - Technical data

5.2 IP address for the netHAT module

The netHAT module requires a DHCP server to obtain an IP address. In a DHCP server, usually an IP address is assigned to a MAC address. The MAC address of the netHAT module starts with 00:02:A2. In the DHCP server, prepare the IP address for the netHAT module. After a reset of the netHAT module, the module requests an IP address from a DHCP server all the time.

5.3 Configuring the EtherNet/IP Scanner

This description contains hints for using a Rockwell PLC configured with the Studio 5000 software.

Prerequisits

1. A DHCP server is available providing an IP address to the netHAT module.
2. Raspberry Pi and the netHAT module are prepared for EtherNet/IP.
3. You already have connected an Ethernet cable from the Scanner to the netHAT module.

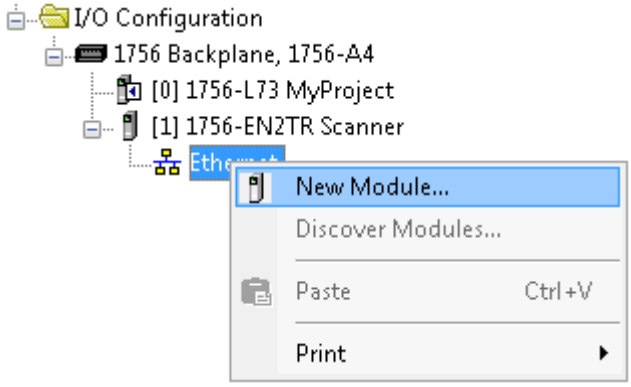
Register EDS file

To register the netHAT module in the Studio 5000 software, use the menu **Tools** > **EDS Hardware Installation Tool**. When ask for the EDS file, select `HILSCHER NXHAT52-RTE EIS V1.1.EDS`.

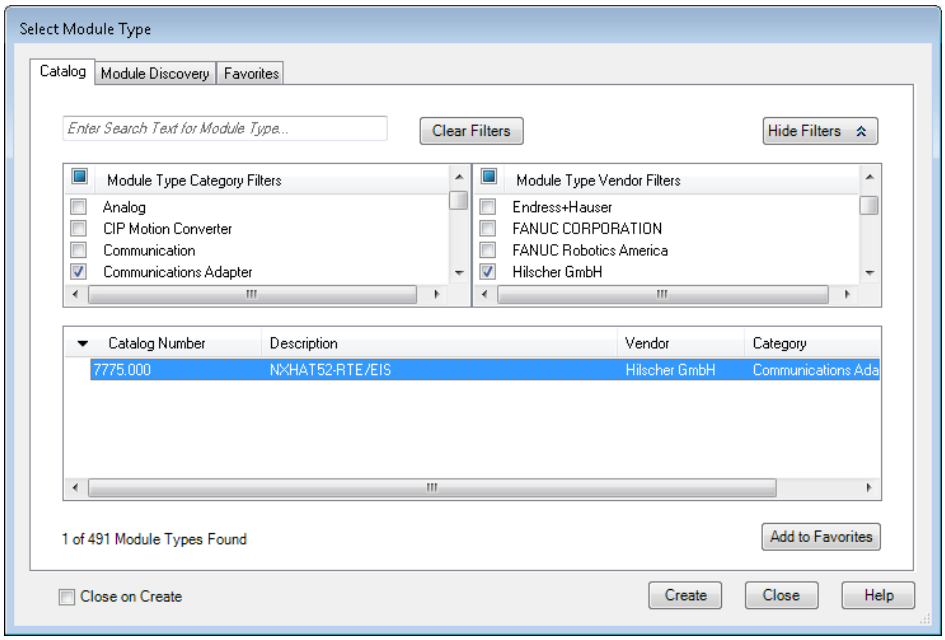
Configuring the EtherNet/IP Scanner

- In Studio 5000, create a new project and select the backplane and controller.
- Select **New module** from the context menu on the backplane.
- Select your EtherNet/IP Scanner module.
- Enter a name for the module, e.g. **Scanner**.
- Set the **Ethernet address**, e.g. IP address 192.168.1.10.
- Close the dialog.

➢ Select t **New module** from the context menu on the **Ethernet** network below the **Scanner**.



➢ Select **Communication Adapter** and **Hilscher GmbH**, then select **NXHAT52-RTE/EIS**.



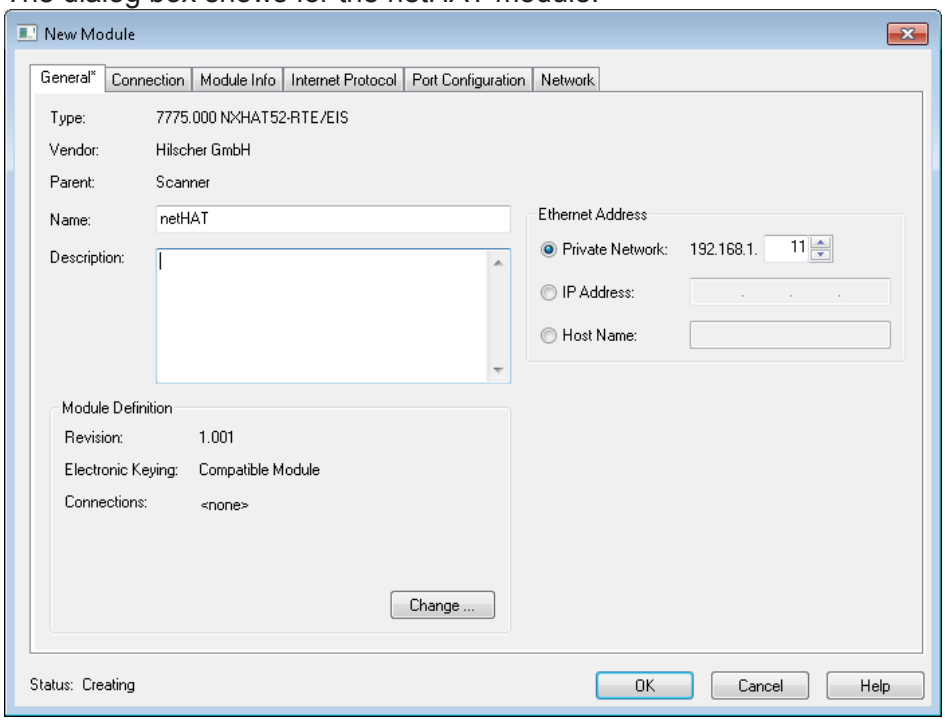
➢ Click **Create**.

➢ The **New module** window opens.

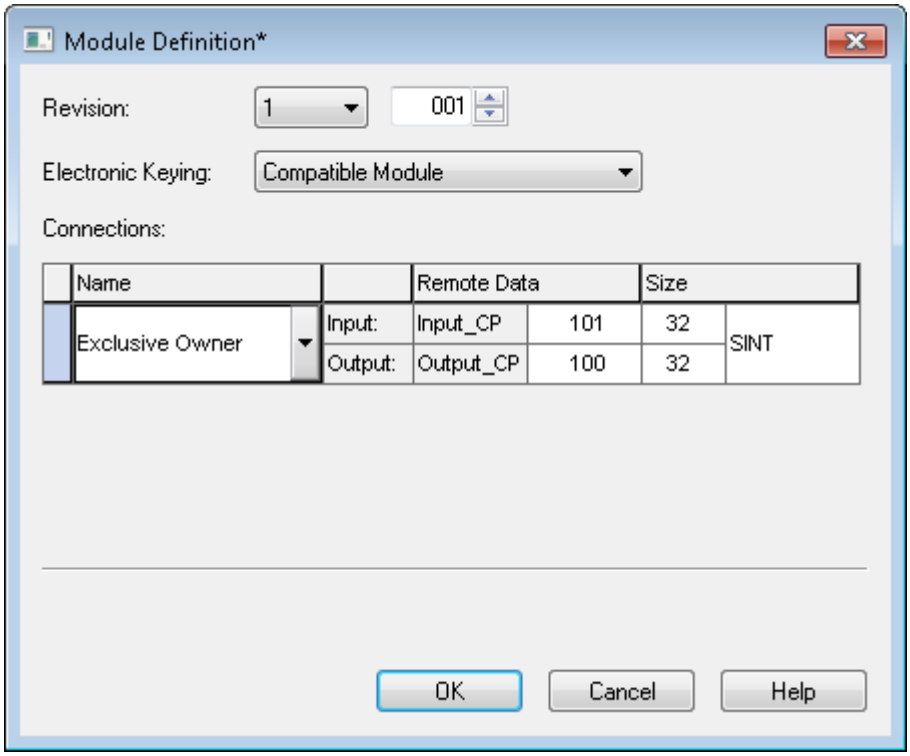
➢ Enter a name, e.g. **netHAT**.

➢ Enter an IP address, e.g. 192.168.1.11.

The dialog box shows for the netHAT module:

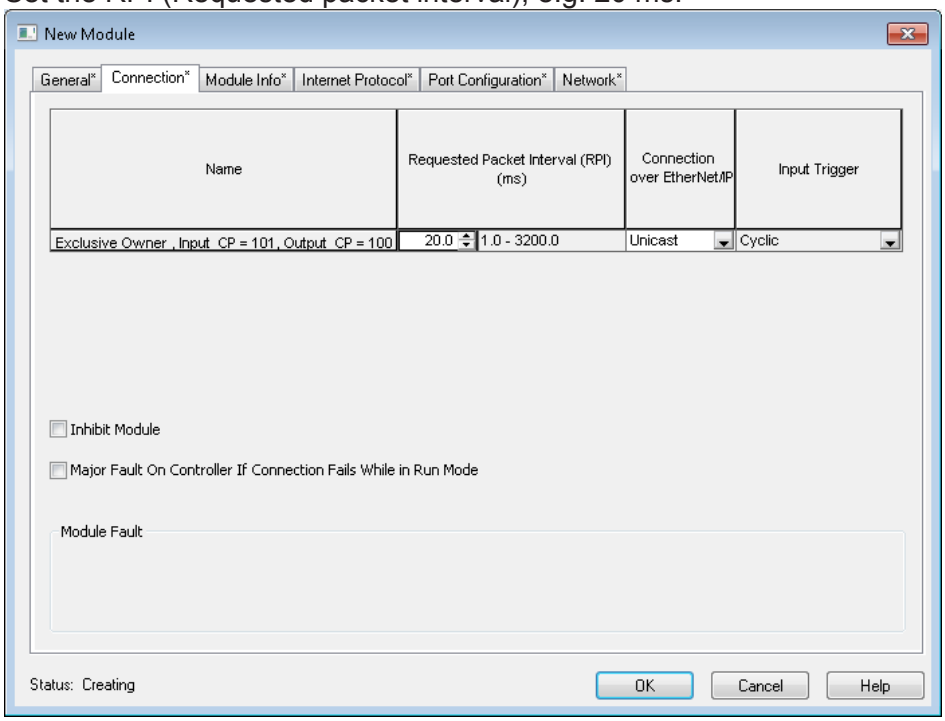


- Use **Change** in the module configuration of Studio 5000.
- Select **Exclusive Owner**.



- Select **Connection** tab.

➢ Set the RPI (Requested packet interval), e.g. 20 ms.



➢ Click **Ok**.

↬ The configuration is prepared.

➢ Go online and download the configuration into the PLC.

➢ Save your project.

Testing I/O data

On the Raspberry Pi, start the example application.

```
sudo ./cifxsample
```

Sending data from the Raspberry Pi to the Scanner:

The example application increments automatically the 32 bytes send to the PLC approx. each second. The example application displays below “IO Write Data” the current data written to the netHAT module:

```
IO Write Data:
09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
```

In Studio 5000, go online with the controller module. Open the **Controller tags**. First, open the input structure (**netHAT:I**) and then open the substructure of the input data (**netHAT:I:Data**). The column **Value** displays the current value of the input data.

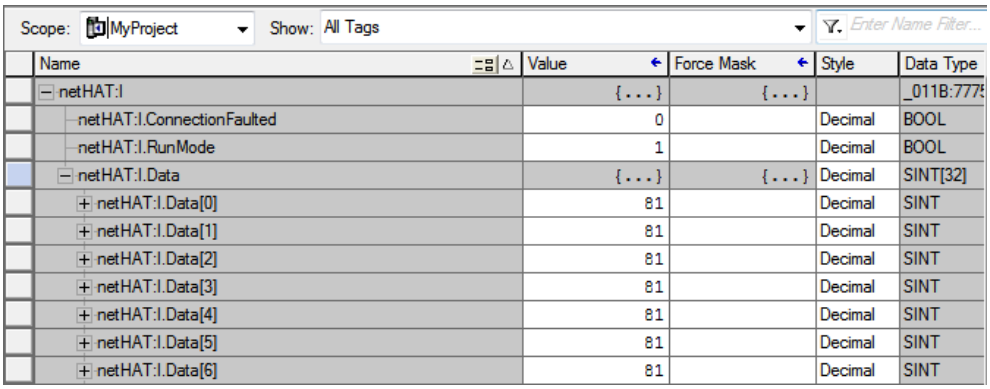


Figure 4: Controller tags input

Data from the PLC to the Raspberry Pi:

In Studio 5000, open the **Controller tags**. First, open the output structure (**netHAT:O**) and then open the substructure output data (**netHAT:O:Data**).

Forcing values requires that the PLC is in Run mode and forcing is enabled (menu **Logic** > **I/O Forcing** > **Enable All I/O Forces**).

In the columns **Value**, enter at **netHAT:O:Data[0]** the value to be send e.g. 123 and press the enter key. **netHAT:O:Data[0]** is the first of the 32 byte output data.

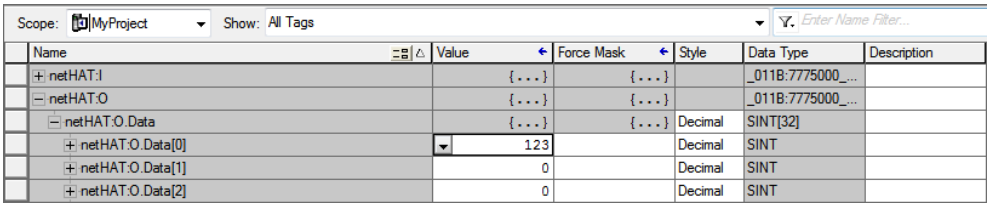


Figure 5: Controller tags output

The example application displays below “IO Read Data” the current data received from the netHAT module:

```
IO Read Data:
7B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

5.4 LEDs Ethernet/IP Adapter

LED	Color	State	Meaning
MS (Module status) Position in the device drawing: (6)	**Duo LED red/green**		
	(green)	On	**Device operational:** The device is operating correctly.
	(green)	Flashing (1 Hz)	**Standby:** The device has not been configured.
	(red/green)	Flashing (1 Hz)	**Self-test:** The device is performing its power up testing.
	(red)	Flashing (1 Hz)	**Minor fault:** The device has detected a recoverable minor fault. E. g. an incorrect or inconsistent configuration can be considered as a minor fault.
	(red)	On	**Major fault:** The device has detected a non-recoverable major fault.
	(off)	Off	**No power:** The power supply to the device is missing.
NS (Network status) Position in the device drawing: (5)	**Duo LED red/green**		
	(green)	On	**Connected:** The device has at least one established connection (even to the Message Router).
	(green)	Flashing (1 Hz)	**No connections:** The device has no established connections, but has obtained an IP address.
	(red/green)	Flashing (1 Hz)	**Self-test:** The device is performing its power up testing.
	(red)	Flashing (1 Hz)	**Connection timeout:** One or more of the connections in which this device is the target have timed out. This status will be finished only if all timed out connections are reestablished or if the device is reset.
	(red)	On	**Duplicate IP:** The device has detected that its IP address is already in use.
	(off)	(Off)	**Not powered, no IP address:** The device does not have an IP address (or is powered off).
LINK CH0 (12), Ch1 (10)	**LED green**		
	(green)	On	The device is linked to the Ethernet.
	(off)	Off	The device has no link to the Ethernet.
ACT Ch0 (11), Ch1 (9)	**LED yellow**		
	(yellow)	Flickering (load dependent)	The device sends/receives Ethernet frames.
	(off)	Off	The device does not send/receive Ethernet frames.

Table 13: LED states for the EtherNet/IP Adapter protocol

LED state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Flashing (1 Hz)	The indicator turns on and off with a frequency of 1 Hz: “on” for 500 ms, followed by “off” for 500 ms.
Flickering (load dependant)	The indicator turns on and off with a frequency of approximately 10 Hz to indicate high Ethernet activity: on for approximately 50 ms, followed by off for 50 ms. The indicator turns on and off in irregular intervals to indicate low Ethernet activity

Table 14: LED state definitions for the EtherNet/IP Adapter protocol

For positions in the device drawing, see section *Interfaces and device drawing* [page 6].

6 PROFINET

6.1 Technical data

Feature	Value
Communication	PROFINET IO Device
I/O data	32 byte input 32 bytes output
Functions	Cyclic process data DCP
GSDML file for configuration tool of the PROFINET Controller	GSDML-V2.32-HILSCHER-NXHAT52-RTE-20160617.xml

Table 15: PROFINET Device - Technical data

6.2 Configuring the IO Controller

This description contains hints for using a Siemens S7 PLC configured with the TIA Portal software.

Prerequisits

1. Raspberry Pi and the netHAT module are prepared for PROFINET.
2. You already have connected an Ethernet cable from the IO Controller to the netHAT module.

Add the netHAT module to the hardware catalog

First, you have to add the netHAT module to the hardware catalog of the TIA Portal. In the TIA Portal, use **Options** > **Manage general station description files (GSD)**. To add the netHAT module to the catalog, select the file `GSDML-V2.32-HILSCHER-NXHAT52-RTE-20160617.xml`.

The netHAT module appears in the catalog below **Other field devices**.

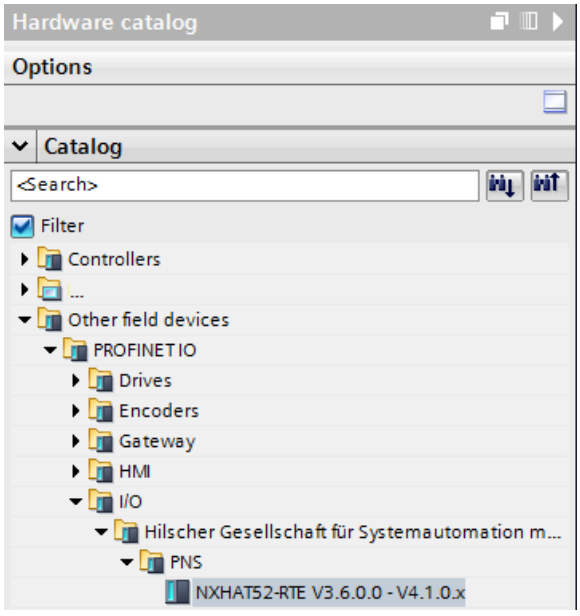


Figure 6: Hardware catalog

Set the name of station

In PROFINET, each device is identified by the Name of Station. The Name of Station has to be set to the device before the IO Controller exchanges I/O data with the device. This function is integrated in the TIA Portal. Alternatively, you can use the **Ethernet Device Configuration** software from Hilscher.

In the TIA Portal, use **Online & Diagnostics** and click **Accessible devices**. Select the **PG/PC interface**. Click **Start search**. Accessible nodes are listed.

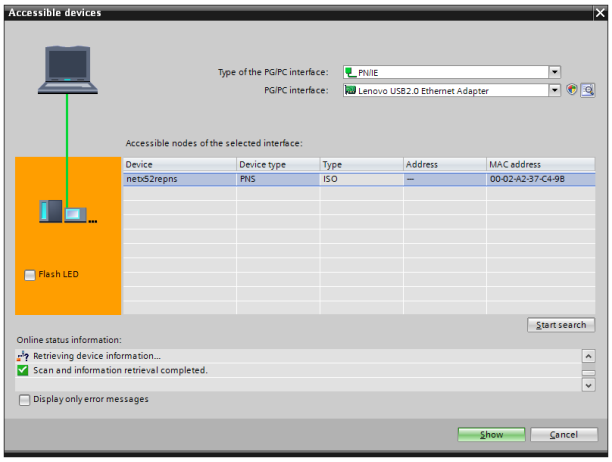


Figure 7: Set name of station (1)

Select the device (single click) and click **Show**. The tab **Devices** in the **Project tree** lists the netHAT module and shows **Online & diagnostics** below the device.

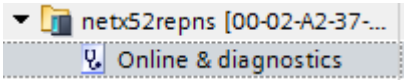


Figure 8: Set name of station (2)

Double-click **Online & diagnostics**. Select **Functions** > **Assign name**.

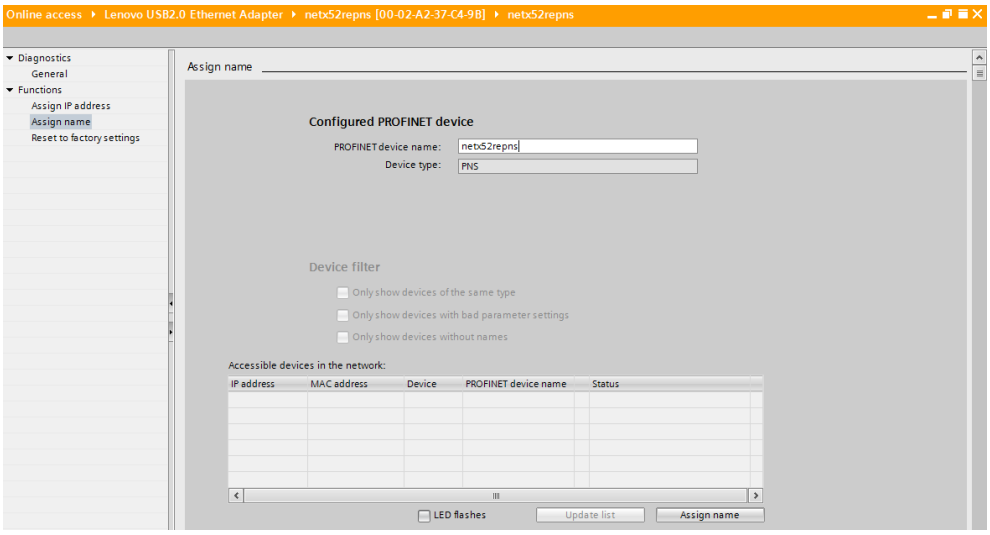


Figure 9: Set name of station (3)

Enter in the field **PROFINET device name** the name you want to assign to the device. Click **Assign name**. Result: The name of station is assigned to the netHAT module.

Configuring the Controller

In the TIA Portal, add the netHAT Module to the network. The netHAT module has 32 byte input and 32 bytes output. In the device configuration, the modules of the netHAT are displayed. The **32 Bytes Input_1** module is located at input address 256...287. The **32 Bytes Output_1** module is located at output address 256...287. The input and output addresses are assigned automatically by the TIA Portal software and can be changed if required. The input and output addresses are relevant for the PLC program to process the input and output data.

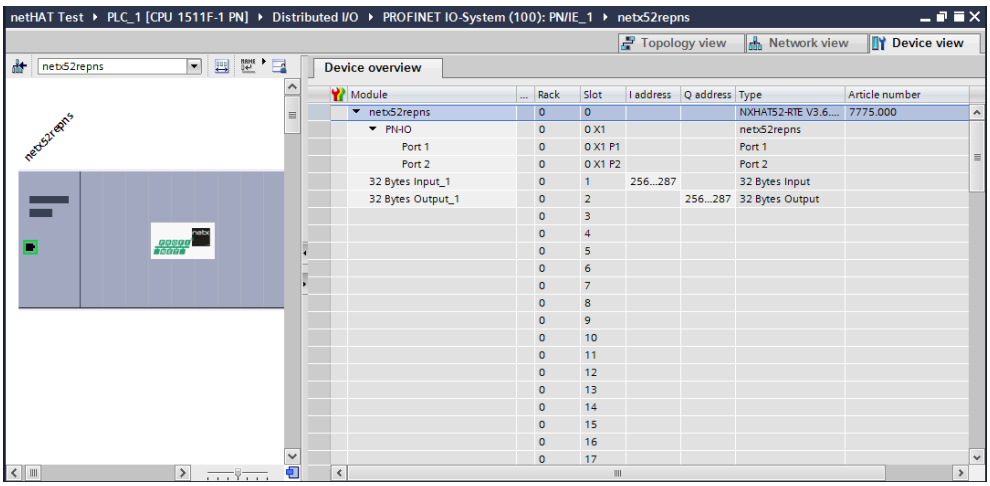


Figure 10: Device configuration

Testing I/O data

On the Raspberry Pi, start the example application.

```
sudo ./cifxsample
```

Sending data from the Raspberry Pi to the IO Controller:

The example application increments automatically the 32 bytes send to the IO Controller approx. each second. The example application displays below “IO Write Data” the current data written to the netHAT module:

```
IO Write Data:
09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09
```

In the TIA Portal, open a watch table. Add the input byte address you want to watch e.g. %IB256 which is the first of the 32 byte input data received from the netHAT module.

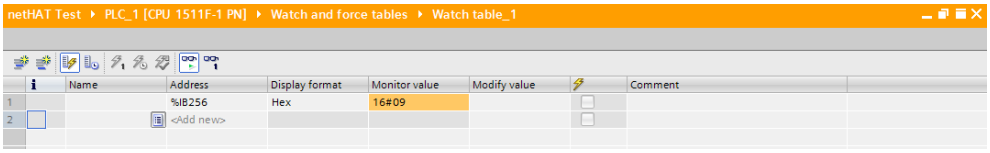


Figure 11: Watch table

Data from the PLC to the Raspberry Pi:

In the TIA Portal, open a force table. Add the output data address you want to force e.g. %QB256 which is the first byte of the 32 byte output data send to the netHAT module. Enter the value to be forced e.g. FF.

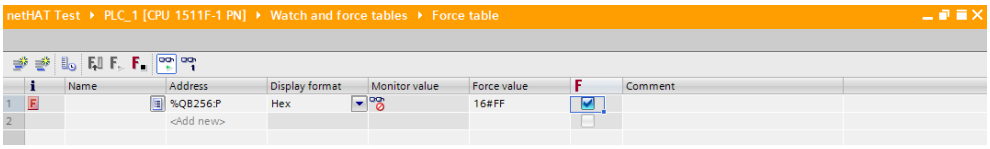


Figure 12: Force table

The example application displays below “IO Read Data” the current data received from the netHAT module:

```
IO Read Data:
FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

6.3 LEDs PROFINET IO Device

LED	Color	State	Meaning
SF (System Failure) Position in the device drawing: (6)	**Duo LED red/green**		
	(off)	(Off)	No error
	(red)	Flashing (1 Hz, 3 s)	DCP signal service is initiated via the bus.
	(red)	On	Watchdog timeout; channel, generic or extended diagnosis present; system error
BF (Bus Failure) Position in the device drawing: (5)	**Duo LED red/green**		
	(off)	Off	No error
	(red)	Flashing (2 Hz)	No data exchange
	(red)	On	No configuration; or low speed physical link; or no physical link
LINK Ch0 (12), Ch1 (10)	**LED green**		
	(green)	On	The device is linked to the Ethernet.
	(off)	Off	The device has no link to the Ethernet.
RX/TX Ch0 (11), Ch1 (9)	**LED yellow**		
	(yellow)	Flickering (load dependent)	The device sends/receives Ethernet frames.
	(off)	Off	The device does not send/receive Ethernet frames.

Table 16: LED states for the PROFINET IO-Device protocol

LED state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Flashing (1 Hz, 3 s)	The indicator turns on and off for 3 seconds with a frequency of 1 Hz: “on” for 500 ms, followed by “off” for 500 ms.
Flashing (2 Hz)	The indicator turns on and off with a frequency of 2 Hz: “on” for 250 ms, followed by “off” for 250 ms.
Flickering (load dependent)	The indicator turns on and off with a frequency of approximately 10 Hz to indicate high Ethernet activity: "on" for approximately 50 ms, followed by "off" for 50 ms. The indicator turns on and off in irregular intervals to indicate low Ethernet activity.

Table 17: LED state definitions for the PROFINET IO-Device protocol

For positions in the device drawing, see section *Interfaces and device drawing* [▸ page 6].

7 Glossary

DCP	Discovery and basic configuration protocol: protocol for identifying and configuring devices which is defined within the PROFINET specification
IP	Internet Protocol: Belongs to the TCP/IP family of protocols and is defined in RFC791. It is based on layer 3 of the ISO/OSI 7 layer model of networking and is a connectionless protocol, i.e. you do not need to open a connection to a computer before sending an IP data packet to it. Therefore IP is not able to guarantee that the IP data packets really arrive at the recipient. On IP level neither the correctness of data nor the consistence and completeness are checked. IP defines special addressing mechanisms, see IP Address.
IP address	Identifies a device or a computer within an IP-based network and is defined in the Internet Protocol Version 4 (IPv4) as a 32-bit number. For ease of notation the address is usually divided into four 8-bit numbers represented in decimal notation and separated by points: a.b.c.d. Each letter stands for an integer value between 0 and 255, e.g. 192.168.30.16. However, not all combinations are allowed, some are reserved for special purposes. The IP address 0.0.0.0 is defined as invalid.
netX	networX on chip, Hilscher network communication controller. High integrated network controller with optimized system architecture for communication and maximum data transfer.
SYNC	Sychronisation Cycle of the Master

List of figures

List of tables

Contacts

HEADQUARTERS

Germany

Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
D-65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support

Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

SUBSIDIARIES

China

Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support

Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France

Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support

Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India

Hilscher India Pvt. Ltd.
Pune
Phone: +91 8888 750 777
E-Mail: info@hilscher.in

Italy

Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support

Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan

Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support

Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea

Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

Switzerland

Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support

Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA

Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support

Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com