CAN-Gateway Configurator V6

for CAN@net NT and CANbridge NT

USER MANUAL

4.02.0332.20001 1.6 en-US ENGLISH

Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Networks of any inaccuracies or omissions found in this document. HMS Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Networks and is subject to change without notice. HMS Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the product. The effects caused by any direct or indirect use of such aspects of the product are undefined and may include e.g. compatibility issues and stability issues.

Table of Contents

Page

This page intentionally left blank

1 User Guide

Please read the manual carefully. Make sure you fully understand the manual before using the product.

1.1 Related Documents

Document	Author
User Manual *CAN@net NT 100/200/420*	HMS
User Manual *CANbridge NT 200/420*	HMS
Software Design Guide *CAN@net NT 100/200/420 Generic Protocol for Gateway Mode*	HMS
User Manual *CAN@net NT/CANbridge NT Lua ADK*	HMS
User Manual *CAN@net NT C-API*	HMS

1.2 Document History

Version	Date	Description
1.0	April 2018	First release
1.1	June 2018	Minor corrections in chapter Action Rules
1.2	January 2019	New CAN Gateway Configurator version, corrections and additional information in chapters MQTT and J1939 mapping
1.3	March 2019	Layout changes
1.4	March 2020	Added service pack 2 functions
1.5	December 2020	Adjusted links
1.6	June 2021	Added UDP, mapping table changes (Tx Msg Format)

1.3 Trademark Information

Ixxat® is a registered trademark of HMS Industrial Networks. All other trademarks mentioned in this document are the property of their respective holders.

1.4 Conventions

Instructions and results are structured as follows:

► instruction 1

► instruction 2

→ result 1

→ result 2

Lists are structured as follows:

- item 1
- item 2

Bold typeface indicates interactive parts such as connectors and switches on the hardware, or menus and buttons in a graphical user interface.

```
This font is used to indicate program code and other
kinds of data input/output such as configuration scripts.
```

This is a cross-reference within this document: *Conventions, p. 6*

This is an external link (URL): www.hms-networks.com

Safety advice is structured as follows:

! Cause of the hazard!

Consequences of not taking remediate action.

How to avoid the hazard.

Safety signs and signalwords are used dependent on the level of the hazard.

This is additional information which may facilitate installation and/or operation.

! This instruction must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.

⚠ **Caution**

This instruction must be followed to avoid a risk of personal injury.

⚠ **WARNING**

This instruction must be followed to avoid a risk of death or serious injury.

2 Product Description

To use all features the latest version of the CAN-Gateway Configurator as well as the latest firmware of the CAN NT device must be installed. For information about firmware versions below V6 contact Ixxat support.

With the CAN-Gateway Configurator the following products can be configured:

- CANbridge NT 200
- CANbridge NT 420
- CAN@net NT 100
- CAN@net NT 200
- CAN@net NT 420

Depending on the device in use different operating modes and configuration options are possible. The different features are described in detail in the chapters of the respective feature in *Configuring the Device, p. 16*.

For hardware information and how to connect the device observe the user manuals of the respective devices.

2.1 Operational Modes CANbridge NT

2.1.1 Repeater/Star Coupler

The CANbridge NT 200 can be configured as Repeater and the CANbridge NT 420 as Star Coupler. In the Repeater/Star Coupler mode all messages are transmitted unchanged to the other ports in Classic CAN mode. Filters, CAN-ID modifications and CAN-FD mode are not possible.

The following settings and features are possible:

- Expert mode
- Communication Error Severity
- Action Rules

2.1.2 Bridge

The Bridge mode allows free configuration of the transmission of CAN messages. With the CANbridge NT 420 NT bridging between Classic CAN and CAN FD is possible.

The following settings and features are possible:

- Use of Lua ADK
- Expert mode
- Communication Error Severity
- Action Rules
- Mapping table
- J1939 Mapping
- CAN FD/CAN Demultiplexing (only CANbridge NT 420)
- CAN/CAN FD Multiplexing (only CANbridge NT 420)
- Cyclic transmission
- CAN tunnel to transmit messages between two Classic CAN networks via a CAN FD network (only CANbridge NT 420)

2.2 Operational Modes CAN@net NT

2.2.1 ASCII Gateway Mode

In the Gateway mode, the CAN@net NT is hooked to the local intranet or internet (firewall needed). This allows any TCP/UDP host within the reach of this intranet or internet to connect to the CAN@net NT and gain control of the CAN system. The Ethernet host can exchange commands and CAN messages using the ASCII protocol. The server relays the commands and messages to the CAN bus and vice versa.

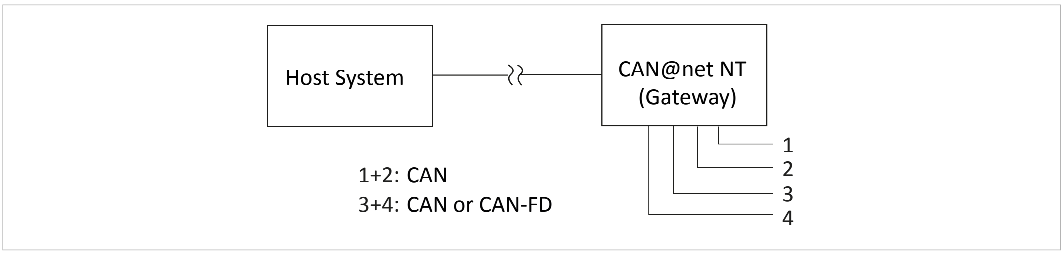


Fig. 1 Gateway mode

For information about the communication in Gateway mode and commands that are used to exchange CAN messages see Software Design Guide *CAN@net NT 100/200/420 Generic Protocol for Gateway Mode* on www.ixxat.com/support-bridges-gateways.

2.2.2 Local CAN Bridge Mode

A single device can be used as Local CAN Bridge, which allows to map individual messages from and to each CAN port of the device. NT 420 devices additionally are capable of CAN FD.

The following settings and features are possible:

- Use of Lua ADK
- Syslog
- MQTT
- Remote access
- Expert mode
- Communication Error Severity
- Action Rules
- Mapping table
- J1939 Mapping
- CAN FD/CAN Demultiplexing (only CAN@net NT 420)
- CAN/CAN FD Multiplexing (only CAN@net NT 420)
- Cyclic transmission

2.2.3 CAN-Ethernet-CAN Bridge Mode

! Exclusively one master device is allowed in the Bridge mode.

The CAN-Ethernet-CAN Bridge mode allows to connect CAN systems over an Ethernet TCP/IP network, for example the local intranet or the internet (firewall needed). Minimum two devices are required for a CAN-Ethernet-CAN Bridge. One has to be configured as master and one as slave. With the NT 100 and NT 200 two devices can be combined to a CAN-Ethernet-CAN bridge. With the NT 420 up to four devices can be combined. The CAN@net NT 420 additionally is capable of CAN FD.

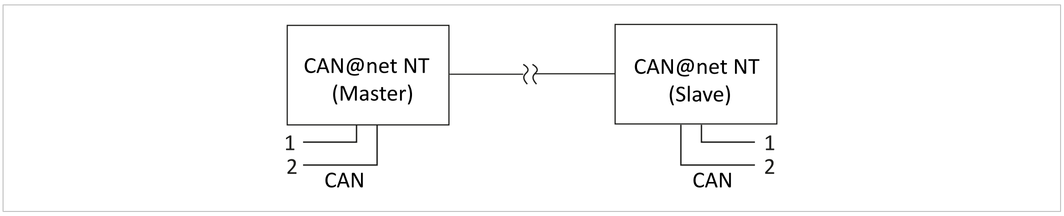


Fig. 2 CAN-Ethernet-CAN Bridge with 2 devices (NT 200)

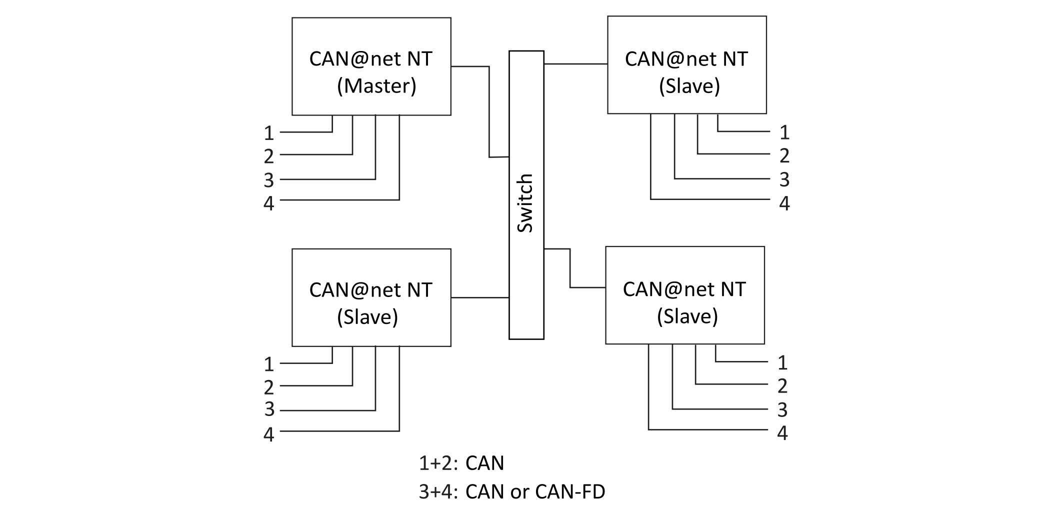


Fig. 3 CAN-Ethernet-CAN Bridge with 4 devices (NT 420)

The following settings and features are possible:

- Use of Lua ADK
- Syslog
- MQTT
- Remote access
- Expert mode
- Communication Error Severity
- Action Rules
- Mapping table
- J1939 Mapping

In the CAN-Ethernet-CAN Bridge mode each device can be configured differently. But to build a Bridge configuration all devices must be configured in one configuration file. The configuration has to be set completely for all devices (Master, Slave 1 to 3) and then the complete configuration has to be downloaded to each device. In the Local CAN Bridge mode only one device is connected and has to be configured.

! Configuration fails, if the individual devices of a CAN-Ethernet-CAN Bridge are configured from different configuration files! Observe that for the configuration of a CAN-Ethernet-CAN Bridge each device must be configured with the same configuration file. If the configuration is changed, the new configuration file has to be downloaded again to all devices.

2.2.4 VCI Interface Mode

> ! The VCI interface mode is only possible via Ethernet.

With the VCI driver the CAN@net NT can be used as a PC interface with Windows. All VCI-based Ixxat tools as well as customer-specific applications based on the VCI driver can be used. The VCI driver offers the possibility to communicate with up to 128 CAN@net NT devices via LAN or internet. The CAN@net NT 420 additionally is capable of CAN FD.

For information about the communication in the VCI mode and commands that are used to exchange CAN messages see Software Design Guides in the VCI download package (available on www.ixxat.com/driver-windows).

2.2.5 ECI Interface Mode

> ! The ECI interface mode is only possible via Ethernet.

With the ECI driver the CAN@net NT can be used as a PC interface with Linux. All ECI-based Ixxat tools as well as customer-specific applications based on the ECI driver can be used. The ECI driver offers the possibility to communicate with up to 32 CAN@net NT devices via LAN or internet. The CAN@net NT 420 additionally is capable of CAN FD.

For information about the communication in the ECI mode and commands that are used to exchange CAN messages see Software Design Guides in the ECI download package (available on the product support pages on www.ixxat.com/driver-linux).

2.3 Add-Ons for Customer Specific Expansions

2.3.1 Lua ADK

With the Lua Application Development Kit customer specific Lua scripts can be executed on the CAN NT device in Bridge operational modes. By using the Lua ADK for handling and processing of communication data the functionality of the standard application can be expanded.

For more information about the Lua ADK see User Manual *CAN@net NT/CANbridge NT Lua ADK* on the product support pages on www.ixxat.com/support-bridges-gateways.

2.3.2 C-API ixcan

The CAN API for C uses the ASCII protocol interface to access the CAN@net NT. The C-API ixcan converts the API calls into corresponding ASCII commands according to the ASCII Gateway Mode of the CAN@net NT. With the application that uses the C-API ixcan the CAN@net NT can be accessed exclusively or in shared access with a Bridge configuration.

For more information about the C-API ixcan see User Manual *CAN@net NT C-API ixcan* on www.ixxat.com/support-bridges-gateways.

3 Installation

3.1 Installing the Software

To create a configuration for the CAN NT device, the CAN-Gateway Configurator running on a Windows system and the Ixxat VCI driver are needed.

The VCI driver is constantly improved and expanded! Check if a newer version is available within the product support pages on www.ixxat.com/support-bridges-gateways.

The CAN-Gateway Configurator and the device firmware are constantly improved and expanded! Check if newer versions are available within the product support pages on www.ixxat.com/support-bridges-gateways.

► Install the VCI driver on a Windows computer (see Installation Guide *VCI Driver*).

► Download the *CAN-Gateway Configurator CANbridge NT & CAN@net NT 100/200/420* package from www.ixxat.com/support-bridges-gateways.

► Start the **Ixxat CanGWconfig Setup**.

→ Installation wizard starts automatically.

► Follow the instructions in installation program.

→ By default the CAN-Gateway Configurator is stored in *C:\Program Files\HMS\Ixxat CAN-Gateway Configurator V6*.

→ The examples for (ASCII, LUA, C-API ixcan, and configuration) are stored in *C:\Users \Public\Documents\HMS\Ixxat CAN-Gateway Configurator\Examples*.

► Check the firmware version in *C:\Users\Public\Documents\HMS\Ixxat CAN-Gateway Configurator\Examples\firmware* and check if a newer firmware version is available on www.ixxat.com/support-bridges-gateways

► If newer firmware version is available, update the firmware (see *Updating the Device Firmware, p. 13*).

► In Windows Start menu open folder **Ixxat CANGWconfig** and start **CAN-Gateway Configurator V6**.

3.2 Checking and Updating the Device Firmware

To use all features the latest firmware versions of the CAN-Gateway Configurator and of the CAN@net NT/CANbridge NT device must be installed.

3.2.1 Checking the Device Firmware

► Make sure, that the latest VCI driver is installed.

► Make sure, that the device is correctly connected to the host computer and to power supply (see User Manual of the respective device for more information).

► Make sure that the latest CAN-Gateway Configurator is installed (check within product support pages on www.ixxat.com/support-bridges-gateways).

► Start the Ixxat CAN-Gateway Configurator.

► Open menu **Scan** and select **All Ixxat devices**.

→ Connected devices and firmware version of the devices are shown.

3.2.2 Updating the Device Firmware

! Whether updating is permitted via Ethernet (CAN@net NT) or a password is needed, is defined in the security settings (see *Security Settings, p. 55*). The default password is IXXAT.

The firmware is constantly improved and expanded! Check if a newer firmware version is available within the product support pages on www.ixxat.com/support-bridges-gateways.

With the CAN-Gateway Configurator devices with firmware version 5 and with version 6 can be configured.

If the current firmware of the device in use is V4 or older:

- See update package on www.ixxat.com/support-bridges-gateways for information about updating to V6 or contact Ixxat support.

If the current firmware of the device in use is V5 or V6:

- Check if newer firmware is available on www.ixxat.com/support-bridges-gateways.
- Download and unzip the update package.
- Make sure, that the device is connected to power supply.
- Connect the device to be updated to the computer.
- Make sure that the latest CAN-Gateway Configurator is installed (check within product support pages on www.ixxat.com/support-bridges-gateways).
- Start the CAN-Gateway Configurator.
- In drop down list **Select device type** select the device in use.
- In drop down list **Select device version** select the current firmware version of the device **V5** or **V6**.

! The device is only found if the selected firmware version matches the firmware version of the connected device.

- Scan for devices with button **Scan** ● and select the device in use in the combo box **Target Device**.
- Click button **Connect** .
- Open menu **Target** and select **Read configuration from target**.
- Save the configuration on the computer.
- Open menu **Target** and select **Update Firmware**.
- Select the update file.
  → Firmware of the connected device is updated.
- In the status window check if the update is completed successfully.
- If the device was updated from V5 to V6 , select **V6** in drop-down list **Select device version**.
- If using a V5 configuration, open menu **File** and select **Convert V5 to V6** to convert the configuration to the latest version.
- Write the saved configuration to the device.

HMS recommends to verify configurations that are converted from V5 to V6, to make sure that all settings are working correctly.

4 Connecting the Device in Use

> **!** Connection disturbance possible if extension cable or longer cable is used!
>
> HMS recommends connecting the interface directly with the included cable or via an active USB hub to the computer according to the USB specification.

> **!** The different CAN@net NT types 100, 200 and 420 can not be combined. For CAN-Ethernet-CAN Bridges use either NT 100 devices, NT 200 devices or NT 420 devices.

> *To use all features the latest firmware versions of the CAN-Gateway Configurator and the CAN NT device must be installed.*

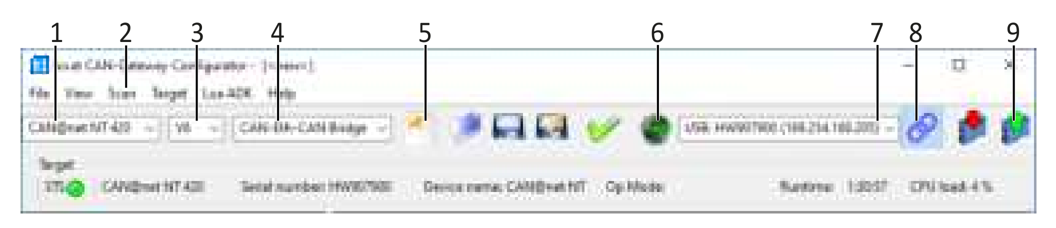


Fig. 4 CAN-Gateway Configurator

- ► Make sure, that the latest driver and the latest CAN-Gateway Configurator is installed (see *Installing the Software, p. 12*).
- ► Make sure, that the device is correctly connected to the host computer and to power supply (see User Manual of the respective device for more information).

> *The CANbridge NT has to be connected via Mini USB cable. The CAN@net NT can be connected via Mini USB cable, Ethernet or a router. HMS recommends to connect each device via Mini USB cable for the first configuration of the device.*

- ► Make sure, that the latest firmware is on the device (see *Checking and Updating the Device Firmware, p. 12*).
- ► To start the Ixxat CAN-Gateway Configurator, in Windows Start menu open folder **Ixxat CANGWconfig** and select **CAN-Gateway Configurator V6**.
- ► To identify the connected devices and the firmware version, open menu **Scan** (**2**) and select **All Ixxat devices**.
  - → Connected devices and firmware version of the devices are shown.
  - → CAN@net NT devices that are connected via a router are not found. IP address and device firmware must be known.
  - → CAN@net NT devices with unknown or invalid IP address are not found, see User Manual *CAN@net NT 100/200/420*, *Scan for Devices with Unknown IP Address* for more information.
- ► Select the type of device in use in the drop-down list **Select device type** (**1**).
- ► Select the firmware version of the device in the drop-down list **Select device version** (**3**).
- ► Select the desired operational mode for the device in use in the drop-down list **Select operational mode** (**4**) (for more information see *Product Description, p. 7*).

- In combo box **Target Device** (**7**) select the device in use.

  or

  If a CAN@net NT is connected via a router, enter the IP address in combo box **Target Device** (**7**).

- Click button **Connect** (**8**) to connect the selected device.

If using the CAN@net NT:

  - For ASCII Gateway, VCI Interface, and ECI Interface mode make sure, that the IP address is in the range of the network in which the device integrated.
  - For CAN-Ethernet-CAN bridge make sure, that the IP addresses of all devices of the bridge are in the same IP range.
  - For more information see User Manual *CAN@net NT 100/200/420*, *Changing IP Address and Device Name*.

- To create a new project file, click button **New** (**5**).

  or

  To change the current configuration, click button **Read from** (**9**) and save the configuration.

- Configure the device in the selected mode (see *Configuring the Device, p. 16*).

5 Configuring the Device

*In the configuration tree open **Info** to add information about the configuration in fields **Author, Configuration Name** and **Additional Info**.*

It is possible to create and save a configuration without a connected device. Saved configurations can be downloaded to connected CAN NT devices with Windows and Linux by using the Command Line Tool (see Downloading the Configuration with Linux, p. 18).

5.1 Basic Configuration Steps

5.1.1 CANbridge NT

- Make sure, that the device is connected and that the desired operational mode is selected (see *Connecting the Device in Use, p. 14*).
- In the configuration tree select **General** and define the general settings (see *General Settings, p. 19*).
- Configure the baud rate settings for all ports in use (see *CAN Ports, p. 22*).
- Configure the mapping table (see *Mapping Table, p. 40*).

! Only messages that are entered in the mapping table are forwarded. By default, no filter is set and all messages are rejected.

- Configure further settings if desired (see respective chapter Action Rules, J1939 Mapping, Cyclic Transmission etc.).
- After the configuration click button **Write to** to write the configuration to the device.
- Click button **Save** or **Save as** to save the configuration.

5.1.2 CAN@net NT Interface Modes (ASCII, VCI, ECI)

! The VCI interface mode and ECI interface mode can only be operated via Ethernet. Configuration is possible via USB.

- Make sure, that the device is connected and that the desired operational mode is selected (see *Connecting the Device in Use, p. 14*).
- In the configuration tree select **Interface**.
- If checkbox **Only for specified device** is enabled, enter the serial number of the device to which the configuration can be written.

If ASCII Gateway Mode is selected:

- Configure the protocol line ending.
- Define the transport protocol (default TCP).
- If UDP is selected, make sure that UDP is also used on the client side.
- Define the IP port.
- If checkbox **Expert Mode** is enabled, select the desired settings (see *Expert Mode, p. 20*).

- After the configuration click button **Write to** to write the configuration to the device.
- Click button **Save** or **Save as** to save the configuration.
- To exchange messages in the Gateway mode, use ASCII commands (for more information see Software Design Guide *CAN@net NT 100/200/420 Generic Protocol for Gateway Mode*).
- In the VCI interface mode configure the Device Server (for more information see Installation Guide VCI Driver).
- For more information about the ECI interface mode, see ECI html help available in ECI download package on product support pages onwww.ixxat.com/support-bridges-gateways.

5.1.3 CAN@net NT Bridge Mode (Local CAN, CAN-Eth-CAN)

In the CAN-Ethernet-CAN Bridge mode each device can be configured differently. But to build a Bridge configuration all devices must be configured in one configuration file. The configuration has to be set completely for all devices (Master, Slave 1 to 3) and then the complete configuration has to be downloaded to each device. In the Local CAN Bridge mode only one device is connected and has to be configured.

! Exclusively one master device is allowed in the Bridge mode.

- Make sure, that the Master device is connected and that the desired operational mode is selected (see *Connecting the Device in Use, p. 14*).
- Configure the following for the Master and for each Slave in use:
  - In the configuration tree select **General** and enter the IP address of the device for CAN-Ethernet-CAN bridges.
  - Define the general settings (see *General Settings, p. 19*).
  - In the configuration tree select **CAN Ports** and configure the baud rate settings for all ports in use (see *CAN Ports, p. 22*).
  - Configure further settings if desired (see respective chapter MQTT, Syslog, Action Rules, etc.).
- Configure the mapping table (see *Mapping Table, p. 40*).

! Only messages that are entered in the mapping table are forwarded. By default, no filter is set and all messages are rejected.

- After the configuration click button **Write to** to write the configuration to the device.
- Click button **Save** or **Save as** to save the configuration.
- For the CAN-Ethernet-CAN Bridge connect the devices one after another and download the configuration to each device.
- Observe that for the configuration of a CAN-Ethernet-CAN Bridge each device must be configured with the same configuration file. If the configuration is changed, the new configuration file has to be downloaded again to all devices.

5.1.4 Downloading the Configuration with Linux

The basic configurations, like the selection of the operating mode, can only be created with the CAN-Gateway Configurator with Windows. A configuration can be created and saved without a connected device and can then be downloaded to connected CAN NT devices with Linux by using the Command Line Tool that is included in the scope of delivery.

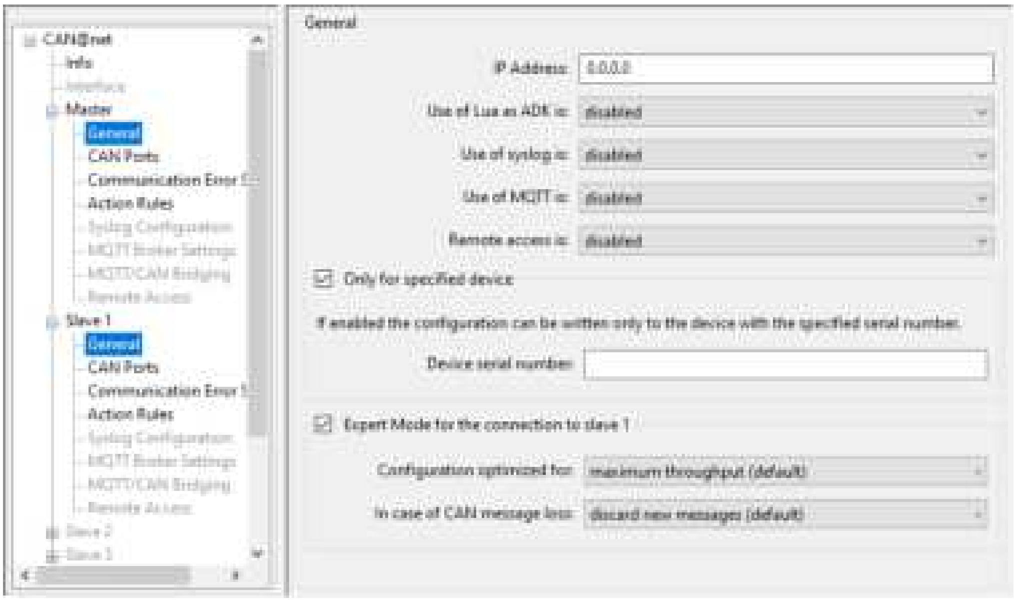
- To be able to read and write configurations on CAN NT devices, copy the included file *60-bgi.rules* to the folder `/etc/udev/rules.d/` (root access required).
- To activate the new rules, execute the following command:

```
udevadm control - -reload-rules
```

- To download a saved configuration file to CAN NT devices, start the command line tool *cangwfile* without parameters.

  → Output shows the syntax, examples and all possible commands.
- Write the configuration to the target device (see *Command Line Program, p. 52* for more information about the Command Line Tool).

5.2 General Settings



General

IP Address: 0.0.0.0

Use of Lua as ADK is: disabled

Use of syslog is: disabled

Use of MQTT is: disabled

Remote access is: disabled

☑ Only for specified device

If enabled the configuration can be written only to the device with the specified serial number.

Device serial number:

☑ Expert Mode for the connection to slave 1

Configuration optimized for: maximum throughput (default)

In case of CAN message loss: discard new messages (default)

Fig. 5 General settings

In the configuration tree in **General** the following settings can be enabled depending on the device in use and the selected operational mode.

Setting	Device	Operational Mode
Lua ADK	CAN@net NT 100/200/420	Local CAN Bridge, CAN-Eth-CAN Bridge
	CANbridge NT 200/420	Bridge
Syslog	CAN@net NT 100/200/420	Local CAN Bridge, CAN-Eth-CAN Bridge
MQTT	CAN@net NT 100/200/420	Local CAN Bridge, CAN-Eth-CAN Bridge
Remote access	CAN@net NT 100/200/420	Local CAN Bridge, CAN-Eth-CAN Bridge
Expert Mode	CAN@net NT 100/200/420	Local CAN Bridge, CAN-Eth-CAN Bridge, ASCII Gateway
	CANbridge NT 200/420	Repeater/Star Coupler, Bridge
CAN tunnel	CANbridge NT 420	Bridge

In the CAN-Ethernet-CAN Bridge mode the settings can be enabled for each connected Master and Slave individually.

5.2.1 Lua ADK

The Ixxat Lua ADK is a firmware extension that is layered over the standard firmware and based on the standard Lua 5.3.5 distribution. By using the Lua ADK for handling and processing of communication data the functionality of the standard application can be expanded. Lua is a powerful, lightweight scripting language for use within the application.

The Lua ADK supports two operational modes:

- running the Lua script on the target device in autonomous mode (**Use of Lua ADK** set to **enabled in target mode**)
- running the Lua script on the host PC for debugging purposes, communicating with the target device via USB (**Use of Lua ADK** set to **enabled in remote mode**)

For information about the Ixxat Lua ADK see User Manual *CAN@net NT/CANbridge NT Lua ADK* on the product support pages on www.ixxat.com/support-bridges-gateways.

5.2.2 Syslog

If the use of syslog is set to **enabled**, the Syslog configuration is activated. For information how to configure Syslog see *Syslog Configuration, p. 28*.

5.2.3 MQTT

If the use of MQTT is set to **enabled**, the configuration for MQTT Broker settings and MQTT/CAN Bridging is activated. For information how to configure MQTT see *MQTT Configuration, p. 30*.

5.2.4 Remote Access

If Remote access is **enabled**, a device that is used in Bridge mode can be accessed in ASCII Gateway mode simultaneously. The CAN controller must be configured and started by the Bridge mode configuration in the CAN-Gateway Configurator.

The CAN controller is controlled via the Bridge and all ASCII commands related to the control are blocked, this means the CAN controller cannot be stopped or modified via ASCII commands. Cyclic messages cannot be defined via ASCII commands in remote access. CAN messages can be sent and received via the ASCII protocol. To receive CAN messages on the host side via ASCII commands, the messages must be added in the Mapping table of the Bridge configuration. The ASCII device commands can also be used in Remote access.

For more information about the ASCII Interface mode and the commands see *CAN@net NT 100/200/420 Generic Protocol for Gateway Mode*.

- ► If Remote access is enabled, open **Remote Access** in the configuration tree and configure the Protocol line ending, the IP port, and the transport protocol if needed.

5.2.5 Expert Mode

If the checkbox **Expert Mode** is activated, the configuration of the master TCP connection can be optimized for different use cases and the behavior in case of CAN message loss can be configured.

Possible configuration optimizations for TCP/IP with CAN@net NT:

- for maximum throughput (default)
- for minimized latency
- for internet connections
- for slow internet connections

Possible behavior in case of CAN message loss:

- discard new messages (default)
- discard old messages

In the CAN-Ethernet-CAN Bridge mode the Expert mode of the Master can be configured individually for the connection to each connected Slave.

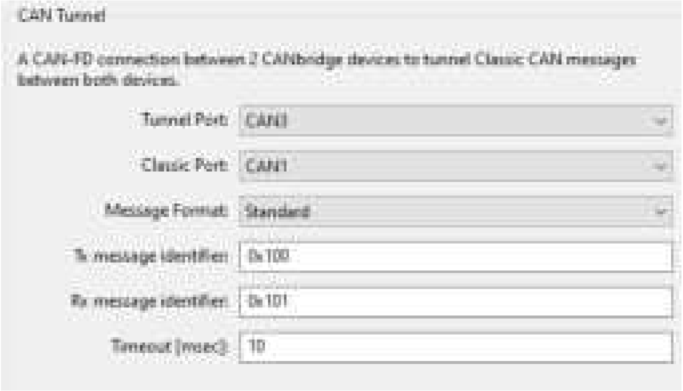
5.2.6 CAN Tunnel

With two CANbridge NT 420 it is possible to transmit messages between two Classic CAN networks via a CAN FD network (CAN tunnel). Only two identifiers are necessary for the CAN FD network. Via these two CAN FD messages all Classic CAN messages are transferred. The busload on the tunnel can be reduced due to the usage of the maximum length of 64 bytes. The **Tx message identifier** of the CAN FD tunnel port of the first device must be configured to the **Rx message identifier** of the CAN FD tunnel port of the second device and vice versa.

> **!** If hexadecimal values are used, they must begin with 0x.
>
> Example: 0x55

- ► To activate a CAN tunnel via CAN FD between two devices, in the configuration tree select **General** and select **enabled** in the field **Use of CAN tunnel**.

  → **CAN Tunnel** is enabled in the configuration tree.
- ► In the configuration tree select **CAN Tunnel**.



CAN Tunnel

A CAN-FD connection between 2 CANbridge devices to tunnel Classic CAN messages between both devices.

Tunnel Port: CAN3

Classic Port: CAN1

Message Format: Standard

Tx message identifier: 0x100

Rx message identifier: 0x101

Timeout [msec]: 10

Fig. 6 CAN tunnel settings

- ► In drop-down list **Tunnel Port** select the transmitting port for the CAN FD messages.
- ► In drop-down list **Classic Port** select the transmitting port for the Classic CAN messages.

  → Classic CAN messages from the **Classic Port** are collected and transmitted as CAN FD messages from the **Tunnel Port** to the second device.
- ► In field **Tx message identifier** enter the identifier of the CAN FD message to be transmitted in decimal or hexadecimal values.
- ► In field **Rx message identifier** enter the identifier of the received CAN FD message in decimal or hexadecimal values.
- ► In field **timeout** specify the maximum time until the CAN FD message is transmitted (even if the 64 bytes are not filled yet).
- ► In **Mapping Table** configure the Classic CAN messages to be transmitted from the defined Classic Port to the defined Tunnel port.
- ► Select tunnel in column **Tx Channel** and define the Classic CAN messages.
- ► Configure the CAN tunnel of the second device.
- ► In drop-down list **Tunnel Port** select the receiving port for CAN FD messages.
- ► In drop-down list **Classic Port** select the receiving port for the Classic CAN messages.

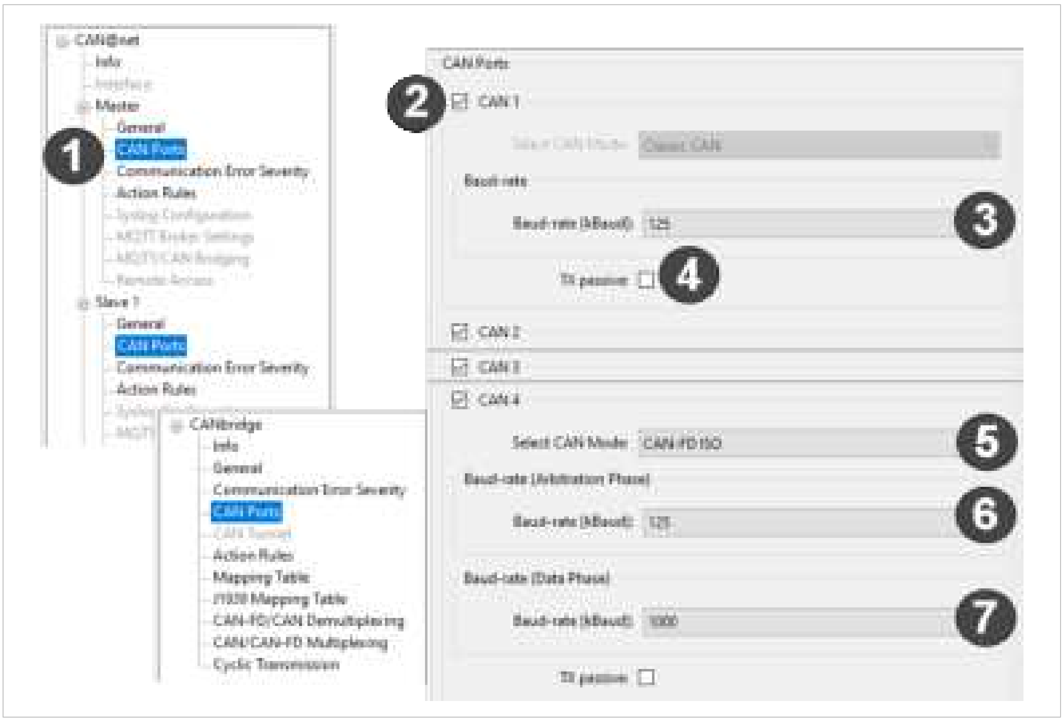
  → CAN FD messages that are received on the tunnel port are divided and transmitted as Classic CAN messages to the configured Classic port.
- ► Make sure, that **Tx message identifier** of the first device matches **Rx message identifier** of the second device.
- ► Make sure, that **Rx message identifier** of the first device matches **Tx message identifier** of the second device.

5.3 CAN Ports

With the CAN@net NT in the CAN-Ethernet-CAN Bridge mode each device can be configured differently. But to build a Bridge configuration all devices must be configured in one configuration file. The configuration has to be set completely for all devices (Master, Slave 1 to 3) and then the complete configuration has to be downloaded to each device.

5.3.1 Baud Rate Settings

With the CANbridge NT in Repeater/Star Coupler mode observe the bus load when setting the baud rates. If the bus load is high on a port with high baud rate but the other port has a low baud rate, a bus overload can occur.



CAN Ports

☑ CAN 1

Select CAN Mode: Classic CAN

Baud-rate

Baud-rate (kBaud): 125

TX passive: □

☑ CAN 2

☑ CAN 3

☑ CAN 4

Select CAN Mode: CAN-FD ISO

Baud-rate (Arbitration Phase)

Baud-rate (kBaud): 125

Baud-rate (Data Phase)

Baud-rate (kBaud): 1000

TX passive: □

Fig. 7 CAN settings NT 420

- ► With CAN@net NT configure the Master and each Slave that is active (black) in the configuration tree.
- ► Select **CAN Ports** in the configuration tree (**1**).
  - → Form to set baud rate of each port of the selected device appears on the right side.
- ► If only certain CAN ports of a device are used, deactivate the check boxes of the CAN ports (**2**) not to be used.
  - → Setting possibilities of deactivated CAN port are disabled.

Baud Rate (3)

- ► Configure the baud rate for each active CAN port in drop-down list (**3**).
- ► With product variants 420 observe the different CAN-FD settings for CAN ports 3 and 4 (see *CAN Mode (5), p. 23*).

Setting the baud rate is possible in different ways:

- predefined CiA baud rate (listed in drop-down list)

- setting with bit timing register (see *User Defined Baud Rates, p. 24*)
- automatic baud rate detection (see *Automatic Baud Rate Detection, p. 26*)

TX Passive Mode (4)

If a CAN port is in TX passive mode, it acts exclusively as listener. It receives messages, but does not transmit messages, nor affect the communication (neither acknowledge bit nor error frames are generated).

► To set a port to TX passive mode, activate the check box **TX passive mode** (**3**).

CAN Mode (5)

The NT 420 supports CAN FD. CAN 1 and CAN 2 are Classic CAN channels.

For CAN 3 and CAN 4 the following CAN modes can be selected:

- Classic CAN
- ISO CAN FD
- Non-ISO CAN FD

CAN FD does not support automatic baud rate detection.

► Select the CAN mode in drop-down list **Select CAN Mode** (**5**).

► In CAN FD mode configure the baud rate for **Arbitration Phase** (**6**) and the baud rate for **Data Phase** (**7**).

Arbitration Phase and Data Phase

CAN FD uses two baud rates: one for the arbitration phase, which is limited to the maximum of Classic CAN (1000 kBit/s) and one for the data phase (up to 10 MBit/s).

5.3.2 User Defined Baud Rates

HMS Industrial Networks recommends using the predefined standard baud rates. If user defined baud rates are used make sure, that the entered values are valid.

If the baud rate is set with the bit timing register of the controller, baud rates that are not defined by CiA can be used.

The clock frequency of the CAN module applied for the calculation of the baud rate is 36 MHz resp. 80 MHz.

Formula for the calculation of the baud rate:

CAN 1 and CAN 2 (Classic CAN)

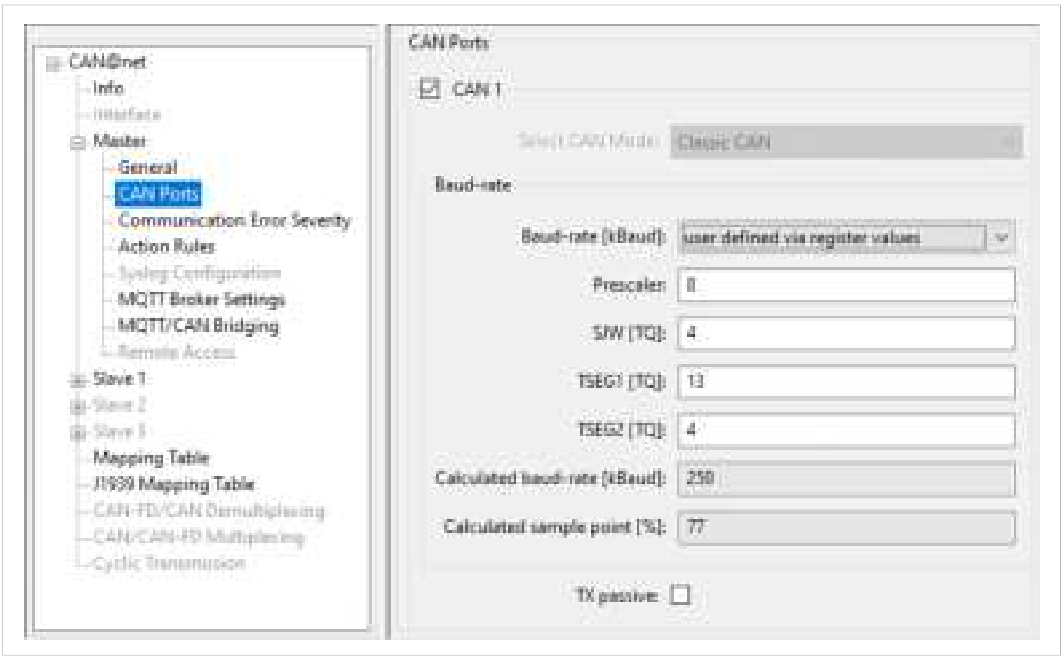
- baud rate [kBaud] = 36 000 / ((TSEG1 + TSEG2 +1) * Prescaler)

CAN 3 and CAN 4 (Classic CAN/CAN FD)

- baud rate [kBaud] = 80 000 / ((TSEG1 + TSEG2 +1) * Prescaler)

► For user defined baud rates select **user defined via register values**.

► Set the values for **Prescaler**, **SJW**, **TSEG1** and **TSEG2**.



☑ CAN 1

Select CAN Mode: Classic CAN

Baud-rate [kBaud]: user defined via register values

Prescaler: 8

SJW [TQ]: 4

TSEG1 [TQ]: 13

TSEG2 [TQ]: 4

Calculated baud-rate [kBaud]: 250

Calculated sample point [%]: 77

TX passive: □

Fig. 8 **Bit timing register**

Calculator for Baud Rate Register Values

With the integrated calculator all necessary register values for a desired baud rate can be calculated.

> Observe that the CAN ports have different controllers and therefore different register values. Make sure, the correct CAN port is selected in the calculator.

- ► To open the calculator click button **Calculator** in the toolbar.
  - → Calculator is opened.
- ► In the drop-down list **CAN port** select the CAN port the user defined register values are used for.
- ► Enter the desired baud-rate in field **Baud-rate**.
- ► Enter the desired sample point ratio in percent in field **Sample-point ratio**.
- ► Click button **Calc**.
  - → Possibilities of values to achieve the desired baud rate and sample point are listed.

Setting Recommendations for CAN FD

> *HMS Industrial Networks recommends using the same bit timing settings in all connected nodes.*

Observe the following recommendations:

- Set arbitration and data phase prescaler as low as possible.
- Configure the same arbitration sample point for all CAN nodes.
- Configure the same data phase primary sample point for all CAN nodes.
- Set SJW for arbitration phase as large as possible.
- Set SJW for data phase as large as required by used oscillator (clock source).

5.3.3 Automatic Baud Rate Detection

! Automatic baud rate detection is exclusively possible if at least two nodes per bus are active.

! Automatic baud rate detection is exclusively possible with Classic CAN.

The ports with activated automatic baud rate detection remain in automatic baud rate detection until the baud rate is detected on each port. Other ports with set or already detected baud rate work regardless of the ports that remain in automatic baud rate detection.

During the automatic baud rate detection CAN 1 LED, CAN 2/3/4 LED (depending on where automatic baud rate detection is enabled) and Status LED indicate the status.

Status	CAN 1 LED	CAN 2 LED	Status LED
Automatic baud rate detection active on both channels	Orange flashing	Orange flashing	Green and orange flashing
Baud rate detected on CAN 1, baud rate detection on CAN 2 active	Off	Orange flashing	Green and orange flashing
Baud rate on CAN 2 detected or adopted from CAN 1, communication present	Green flashing	Green flashing	Green flashing

Adopting a Detected Baud Rate to Further CAN Ports

With Actions Rules it is possible to configure to adopt a detected baud rate of a CAN port to a second CAN port. The baud rate is only adopted, if no baud rate is set or detected on the second port.

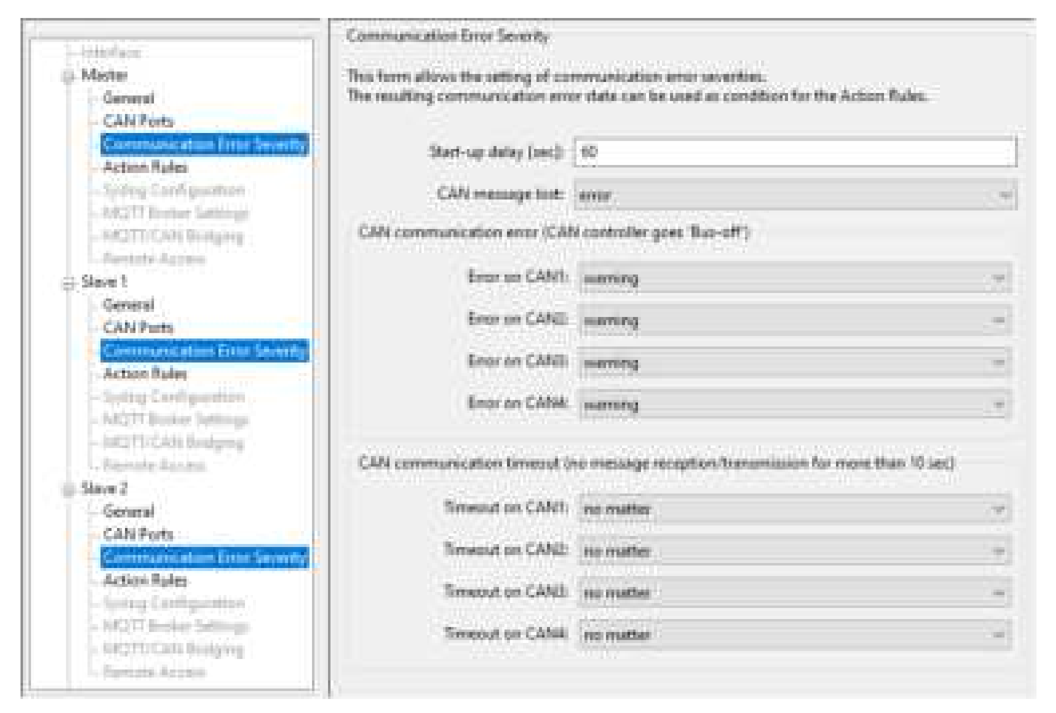
- ► In drop-down list **Baud-rate** (**3**) select **automatic baud-rate detection** for the individual ports.
- ► In the configuration tree select **Action Rules**.
- ► Click in column **IF Event** and select event type **CAN baud-rate detected**.
- ► Select the port and click button **OK**.

  → IF event is entered in the table.
- ► Click in column **THEN action** and select action type **take over CAN baud-rate**.
- ► Select the port on which the detected baud rate is adopted and click button **OK**.

  → THEN-action is entered in the table.
- ► For information about further configuration possibilities see *Action Rules, p. 34*.

5.4 Communication Error Severity

With the CAN@net NT in the CAN-Ethernet-CAN Bridge mode each device can be configured differently. But to build a Bridge configuration all devices must be configured in one configuration file. The configuration has to be set completely for all devices (Master, Slave 1 to 3) and then the complete configuration has to be downloaded to each device.



Communication Error Severity

This form allows the setting of communication error severities.
The resulting communication error state can be used as condition for the Action Rules.

Start-up delay [sec]: 60

CAN message lost: error

CAN communication error (CAN controller goes 'Bus-off')

Error on CAN1: warning

Error on CAN2: warning

Error on CAN3: warning

Error on CAN4: warning

CAN communication timeout (no message reception/transmission for more than 10 sec)

Timeout on CAN1: no matter

Timeout on CAN2: no matter

Timeout on CAN3: no matter

Timeout on CAN4: no matter

Fig. 9 Communication Error Severity

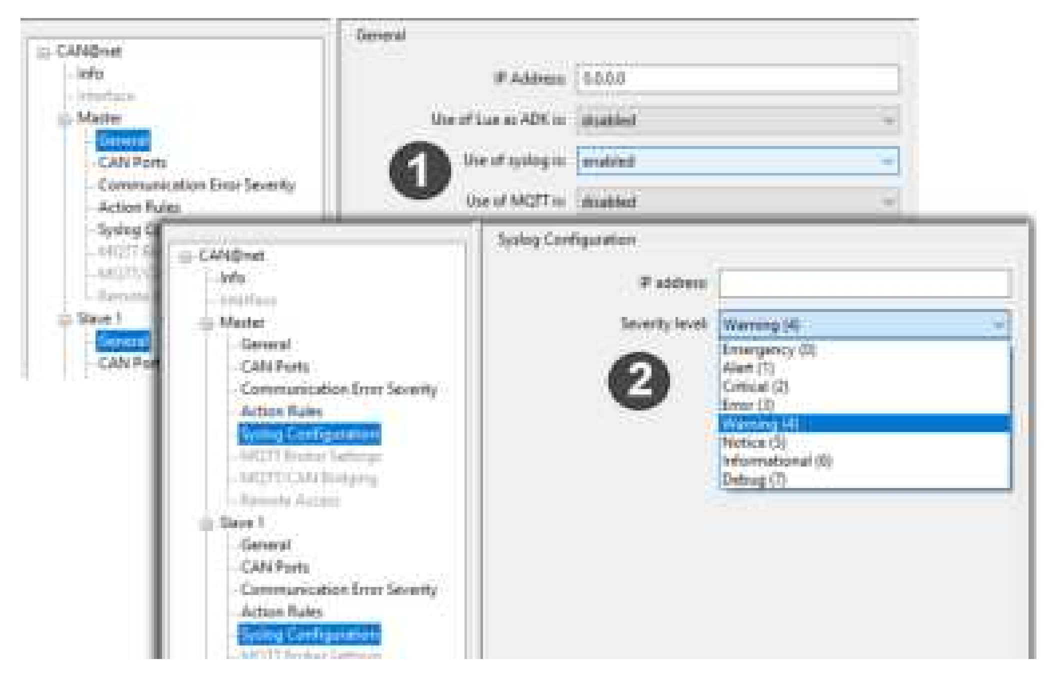
The communication error state can be used as a condition for Action Rules in all autonomous setups. With Actions Rules it is possible to define an event to take place if the device changes in state *warning* or *error*. See *Action Rules, p. 34* for more information.

- **Start-up delay** defines the delay until the monitoring is activated after the power on of the device. (Exception: a bus off is directly handled.)
- **CAN message lost** defines which error state is set after an overload situation inside the device, e.g. at a buffer overflow.
- **CAN communication error** defines which error state is set if a CAN controller goes into bus off state for each CAN port separately.
- **CAN communication timeout** defines which error state is set if no message is received or transmitted for over 10 seconds for each CAN port separately.
- Possible settings:
  - no matter: no reaction
  - warning: a *Communication changed to warning* event is generated
  - error: a *Communication changed to error* event is generated

Observe for the configuration, that the device must be stopped and started to leave the states warning and error.

5.5 Syslog Configuration

With the CAN@net NT in the CAN-Ethernet-CAN Bridge mode each device can be configured differently. But to build a Bridge configuration all devices must be configured in one configuration file. The configuration has to be set completely for all devices (Master, Slave 1 to 3) and then the complete configuration has to be downloaded to each device.



General

IP Address: 0.0.0.0

Use of Lua as ADK is: disabled

Use of syslog is: enabled

Use of MQTT is: disabled

Syslog Configuration

IP address:

Severity level: Warning (4)

Emergency (0)
Alert (1)
Critical (2)
Error (3)
Warning (4)
Notice (5)
Informational (6)
Debug (7)

Fig. 10 Syslog configuration

Syslog messages are only possible with the CAN@net NT via Ethernet.

If Syslog is enabled, standardized log messages can be transmitted to a receiver with a specified IP address. In **Action Rules** it has to be defined which syslog messages are transmitted from the CAN@net NT.

5.5.1 Severity Level

The severity levels are defined by the syslog standard. The severity level of the syslog messages is set in action rules for each message individually. The severity level setting in the menu **syslog configuration** works as a filter. All messages with the selected severity level and lower are forwarded to the syslog server.

Example:

If severity level Error (3) is set, the messages with the following severity levels are forwarded to the syslog server:

- Error (3)
- Critical (2)
- Alert (1)
- Emergency (0)

5.5.2 Enabling the Syslog Configuration

- ► To enable Syslog, in the configuration tree select **General** and select **enabled** in the field **Use of Syslog**.

  → **Syslog Configuration** is enabled in the configuration tree.

- ► In the configuration tree select **Syslog Configuration**.
- ► Define the IP address of the syslog server.

DNS entries are possible with the latest CAN-Gateway Configurator version.

- ► Select the severity level filter in drop-down list **Severity level**.

5.5.3 Defining Syslog Messages

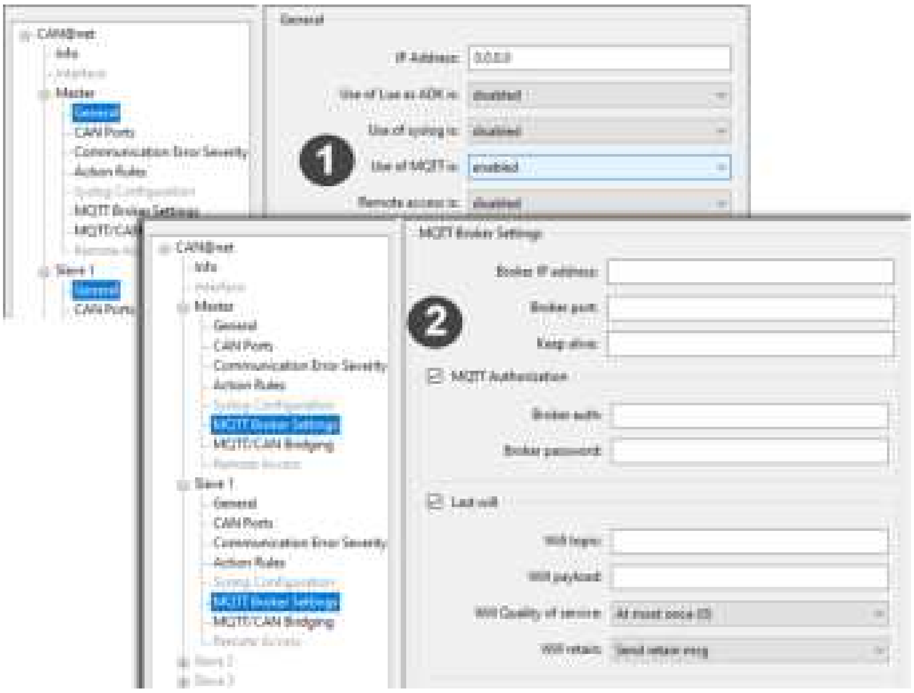
For each syslog message the following has to be defined via action rules:

- event (trigger) to transmit a syslog message
- severity level
- payload (ASCII string)

- ► Enable the syslog configuration (see *Enabling the Syslog Configuration, p. 29*).
- ► Define syslog messages via **Action Rules**:
  - ► Configure an **IF event** to set the trigger for the transmission of a syslog message.
  - ► As **THEN action** select the action type **Send SYSLOG message** and define the message.
- ► For more information see *Action Rules, p. 34*.

5.6 MQTT Configuration

With the CAN@net NT in the CAN-Ethernet-CAN Bridge mode each device can be configured differently. But to build a Bridge configuration all devices must be configured in one configuration file. The configuration has to be set completely for all devices (Master, Slave 1 to 3) and then the complete configuration has to be downloaded to each device.



General

IP Address: 0.0.0.0

Use of Lua as ADK is: disabled

Use of syslog is: disabled

Use of MQTT is: enabled

Remote access is: disabled

MQTT Broker Settings

Broker IP address:

Broker port:

Keep alive:

☑ MQTT Authorization

Broker auth:

Broker password:

☑ Last will

Will topic:

Will payload:

Will Quality of service: At most once (0)

Will retain: Send retain msg

Fig. 11 MQTT configuration

MQTT messages are only possible with the CAN@net NT via Ethernet.

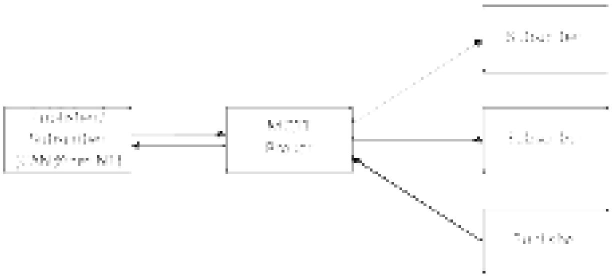


Fig. 12 MQTT Publisher and Subscriber

The CAN@net NT supports MQTT v3.1.1 and can act as publisher and as subscriber. With the MQTT/CAN Bridging module CAN messages of a defined format can be published and received via MQTT. Additionally as publisher the CAN@net NT can publish messages via MQTT that can be individually defined in **Action Rules**. The MQTT broker has to be configured in **MQTT Broker Settings**.

HMS recommends to use a MQTT broker within the local firewall. Observe that MQTT is a open and unprotected protocol and that third parties can read the transmitted messages if a public broker is used.

5.6.1 Enabling the MQTT Configuration

Enabling MQTT

► To enable MQTT, in the configuration tree select **MQTT** and select **enabled** in the field **Use of MQTT**.

→ **MQTT Broker Settings** is enabled in the configuration tree.

► In the configuration tree select **MQTT Broker Settings**.

► Enter the IP address of the desired broker (within the local firewall) in **Broker IP address**.

DNS entries are possible with the latest CAN-Gateway Configurator version.

► Enter the broker port in **Broker Port**.

► Define the keep alive time in field **Keep alive**.

The keep alive functionality assures that the connection is open and both broker and client are connected to one another. When no messages are transmitted and the keep alive time is exceeded the subscribing client has to transmit a ping request to ensure that the connection is still open.

MQTT Authorization

► If the broker in use demands an authorization, activate the checkbox **MQTT Authorization** and enter the authorization and the password of the broker.

Last Will

The last will functionality is used to inform subscribing clients if a publishing client is disconnected. The broker stores the last will message of the publishing client. If this client is disconnected abruptly the broker transmits the message to all subscribing clients.

According to the MQTT specification the last will message is transmitted in the following cases:

- Server detected an I/O error or network failure.
- Client fails to communicate within keep alive time.
- Client closes the network connection without sending a DISCONNECT package.
- Server closes the network connection because of a protocol error.

► To define the last will, activate the checkbox **Last will**.

► Define the will topic and the will payload.

For topic and payload it is possible to use system variables that are replaced with actual values when a message is sent (see System Variables for Topic and Payload, p. 33).

► Select the Quality of service for the last will message in drop-down list **Will Quality of service**.

► Define if a retain message is transmitted in the drop-down list **Will retain**.

5.6.2 Configuring MQTT/CAN Bridging

With the MQTT/CAN Bridging module CAN messages in JSON format can be published and received via MQTT.

- Enable the MQTT configuration (see *Enabling the MQTT Configuration, p. 31*).
- In the configuration tree select **MQTT/CAN Bridging**.
- To subscribe to a CAN message, enter the MQTT topic of the message in the table **MQTT Subscribe**.
  - → If an MQTT message of the defined format is published, it is received by the CAN@net NT.
- To publish a received CAN message via MQTT, enter the message and the MQTT topic in the table **MQTT Publish**.

For the topic it is possible to use system variables that are replaced with actual values when a message is sent (see System Variables for Topic and Payload, p. 33).

Example: JSON Format of a CAN Message

```
{“port”:1, “format”:”csd”,”ident”:256,”data”:[17,34,51,68]}
```

port	CAN port number (NT 100: 1, NT 200: 1...2, NT 420: 1...4)
format	Message format according to CFT: • C – Controller type (C – CAN, F — CAN FD) • F – Frame Format (S – Standard, E – Extended) • T – Frame Type (D – Data, R – RTR) Remote frames (RTR) are only supported by Classic CAN.
ident	Message identifier (decimal)
data	List of data bytes (0..64 values)

5.6.3 Defining MQTT Messages

As publisher the CAN@net NT can publish messages via MQTT that can be individually defined in **Action Rules**.

For each MQTT message the following has to be defined via **Action Rules**:

- event (trigger) to transmit an MQTT message
- message topic (string to filter and route the messages to the subscribers)
- message payload (ASCII string)
- Quality of Service (QoS)

If the Master and the Slave device use the same broker, the topics of the messages of Master and Slave must be different (add e.g. the serial number).

- Enable the MQTT configuration (see *Enabling the MQTT Configuration, p. 31*.
- Define MQTT messages via **Action Rules**:
  - Configure an **IF event** to set the trigger for the transmission of an MQTT message.
  - As **THEN action** select the action type **Send MQTT message** and define the message.
- For information about the configuration possibilities see *Action Rules, p. 34*.

System Variables for Topic and Payload

For topic and payload it is possible to use system variables that are replaced with actual values when a message is sent.

For example, the following keywords are possible:

- device type: $dev_type
- serial number: $ser_num
- firmware version: $fw_ver
- FPGA version: $fpga_ver
- hardware version: $hw_ver
- device name: $dev_name
- configuration type: $conf_type
- IP address: $ip_addr

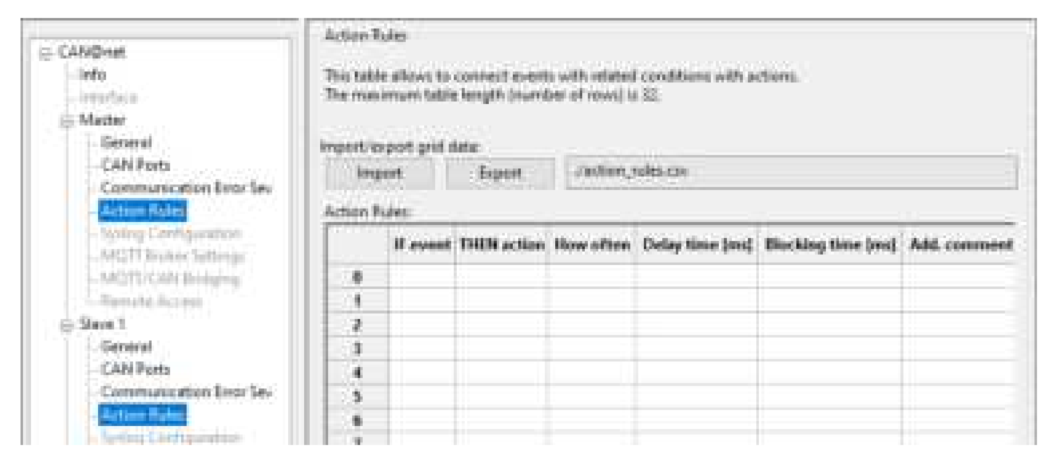
A list of more example variables is integrated in the THEN action **Send MQTT message** in the module Action Rules.

► Use space characters before and after the keyword.

► If the keyword is not separated by space characters, add curly brackets, for example ${ser_num}.

5.7 Action Rules

With the CAN@net NT in the CAN-Ethernet-CAN Bridge mode each device can be configured differently. But to build a Bridge configuration all devices must be configured in one configuration file. The configuration has to be set completely for all devices (Master, Slave 1 to 3) and then the complete configuration has to be downloaded to each device.



Action Rules

This table allows to connect events with related conditions with actions.
The maximum table length (number of rows) is 32.

Import/export grid data: ./action_rules.csv

Action Rules:

	If event	THEN action	How often	Delay time [ms]	Blocking time [ms]	Add. comment
0						
1						
2						
3						
4						
5						
6						

Fig. 13 Action rules

It is possible to set If-Then-Rules for the configuration. Various IF events and THEN actions can be selected and combined in **Action Rules**.

5.7.1 Importing and Exporting Configurations

Configurations can be saved and edited in csv format.

- To load an existing csv file with actions rules, click button **Import** in **Action Rules**.
  - → Window **Select a File** is opened.

If a csv file is imported, already entered action rules are deleted.

- To save configured Action Rules as template, click button **Export**.

5.7.2 Defining a Rule

To define a new action rule configure the columns of the table.

If hexadecimal values are used, they must begin with 0x.

Example: 0x55

- Define an IF event:
  - Click in column **IF Event**.
    - → Window to define an event is opened.
  - In drop-down list **Select Event Type** select the desired event (see *Possible Events, p. 36*).
    - → Depending on the selected type, further configuration options are shown.
  - Set all necessary configurations and click button **OK**.
    - → IF event is entered in the table.

- Define a THEN action equally to the IF event (for possible action types see *Possible Actions, p. 37*).

  → THEN action is entered in the table.

- Define how often the rule is executed in the column **How often**.

  → When the defined event occurs, the action is executed and the counter is decreased by one.

- For endless repetition enter value 0 in column **How often**.
- If column **How often** is 1, enter 0 in column **Blocking time**.
- In column **Blocking time** define the minimum time between two executed actions in milliseconds.

  or

  In column **Delay time** define the delay time between the reception of an event and the execution of the action in milliseconds (for more information see *Delay Time/Blocking Time, p. 35*).

- To add or delete a row, right-click on the left number column and select **Insert new cells** or **Delete cells**.

Delay Time/Blocking Time

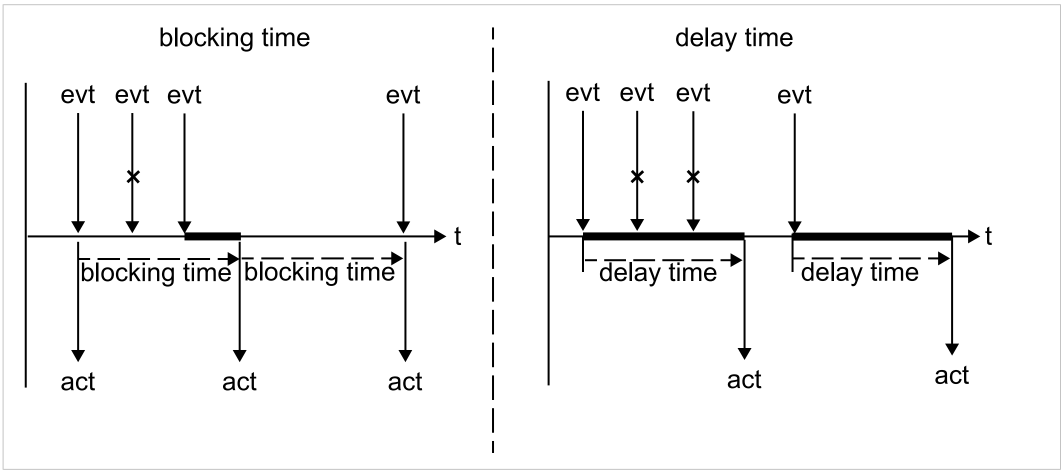


Fig. 14 Blocking time and delay time

It is possible to configure either a delay time or a blocking time.

The blocking time defines the minimum time between two executed actions. The blocking time starts after the execution of an action. During the defined time span no action is executed. When the blocking time is expired, the data of the last received event during the blocking time is used for the action. Other events that occur during the blocking time are discarded.

The delay time defines the time span between the reception of an event and the execution of the action. When the delay time is expired the action is executed. All subsequently events that occur during the delay time are discarded.

Possible Events

Event name	Description	Further settings
CAN message received	If this event is set, the defined action is triggered if the specified CAN message is received on the specified CAN port. With the **and condition** conditions for a specified CAN data byte can be set.	**CAN Port**: port to receive the message **CAN message format**: standard or extended **CAN identifier**: identifier of the message to be received **AND <condition>:** **Select condition**: Data Byte **CAN Data Byte**: data0 to data7 **Arithmetic operator**: equal (==), unequal (!=), higher (>), lower (<), bitwise and (&) **Compare value**: value to compare the specified data byte with (hexadecimal or decimal)
CAN message timeout	With **CAN message timeout** it is possible to monitor, if a port transmits a certain message within a defined time. If **Timeout after** is set to **start delay expired**, the delay time from the power on until the start of the timeout timer has to be defined.	**CAN Port**: port to transmit the message **CAN message format**: standard or extended **CAN identifier**: identifier of the message to be transmitted **Timeout after**: *first message reception* starts the timer after the first reception of the message. *start delay expired* starts the timer after Power on when the configured delay time is expired. **Timeout start delay time**: after power on of the device in milliseconds, only if **Timeout after** is set to **start delay expired** **Timeout value**: in milliseconds
Cyclic timer expired	With *Cyclic Timer expired* it is possible to set a cycle time, so that a defined action is executed when the set cycle time is expired.	**Cycle time**: in milliseconds (>= 10 ms) **Start delay time**: after power on of the device
CAN busoff detected	–	CAN Port
CAN error status passive	Error of CAN controller, controller is in error passive state (controller cannot send error frames)	CAN Port
CAN error status active	No error, controller is in error active state (controller can send error frames if needed), event is only reported if CAN controller was in CAN Error Status Passive	CAN Port
CAN data overrun detected	–	CAN Port
CAN no communication	The event *CAN no communication* is triggered, if there is no CAN communication for 10 seconds on the defined port (observe Start-up delay time configured in *Communication Error Severity, p. 27*).	CAN Port
Power on	Power on of the device	–
Soft reset detected	A soft reset is done when a new configuration is loaded via the CAN-Gateway Configurator	–
Comm. error state changed to warning	In **Communication Error Severity** it is possible to configure that the device changes to status warning in defined events (see *Communication Error Severity, p. 27*).	–
Comm. error state changed to error	In **Communication Error Severity** it is possible to configure that the device changes to status warning in defined events (see *Communication Error Severity, p. 27*).	–
CAN status byte changed	It is possible to monitor, if the error state of the CAN ports changes. A CAN status byte can be of the states Error Active 0x00, Error Passive 0x01, Bus Off 0x02 and Not Available 0xFF.	CAN ports 1 to 4 can be selected via checkboxes.
CAN baud-rate detected	Only possible with Classic CAN, if automatic baud rate is activated for the port.	CAN ports 1 to 4 can be selected via checkboxes.

Possible Actions

Action name	Description	Further settings
Start CAN	–	CAN Port
Stop CAN	–	CAN Port
Send CAN status message	In the CAN status message the error state of CAN1 is in byte 1, CAN2 in byte 2, CAN3 in byte 3 and CAN4 in byte 4 (see*CAN Error State in Bytes 1–4 of the CAN Status Message, p. 38*). Byte 1 is the first data byte after the identifier. The remaining data bytes 5–8 are used as firmware state bytes (see *Data Bytes of CAN Status Message, p. 38*).	**CAN Port**: port to transmit the message to **CAN message format**: standard or extended **CAN identifier**: identifier of the message
Send CAN message	–	**CAN Port**: port to transmit the message to **CAN message format**: standard or extended **CAN identifier**: identifier of the message to be transmitted **CAN data**: decimal and hexadecimal values possible
Send MQTT message	If the subscribing clients are only interested in certain messages, the broker can filter the messages that are sent to the clients via the message topics. Each topic consists of one or more topic levels. Each topic level is separated by a forward slash. A predefined message payload can be selected from the drop-down list.	**Message topic**: UTF-8 string, levels separated by forward slash. If the Master and the Slave device use the same broker, the topics of the messages of Master and Slave must be different (add e.g. the serial number). **Message payload**: if raw message is selected, enter the message in field below. Via the button **Variables** various system variables can be selected and added to the payload. Variables can also be added to the topic. **QoS**: At most once (0), At least one (1), Exactly once (2)
Recover CAN	Recovery from CAN Bus off. Can be used with the IF Action **CAN BusOff detected**.	CAN port
Send SYSLOG message	The message is only forwarded, if the severity level of the message passes the severity level filter set in **syslog configuration** (see *Syslog Configuration, p. 28*).	**Severity level**: INFO, WARN, ERR, CRIT **Message payload**: enter message in field. Via the button **Variables** various system variables can be selected and added to the message.
Write error log	Writes an error log in the device log file, that can be read and saved via the menu **Target — Read and erase LOG file**.	**Prefix**: depending on the selected level a single character is placed in the beginning of the logging entry **Message payload**: enter message in field. Via the button **Variables** various system variables can be selected and added to the message.
Set user LED	With the flash pattern the LED lights up for the specified time. With the blink pattern, the LED is blinking until the device is turned off.	With LED pattern blink: **Cycle time** defines the time for a complete cycle of the LEDs (from on to off). With LED pattern flash: **Duration** defines the time the LED is on.
Write a terminal message	With this action a message can be written to the terminal window.	**Payload**: enter raw message. Via the button **Variables** various system variables can be selected and added to the message.
Set device state	Observe that the device must be stopped and started to leave the states *warning* and *error.*	Severity level: WARN, ERR
take over CAN baud-rate	If automatic baud rate detection is activated on the selected port, but no baud rate is detected, the baud rate that is detected on the port that is defined in the IF action **CAN baud-rate detected** is adopted (see *Automatic Baud Rate Detection, p. 26*).	CAN ports 1 to 4 can be selected via checkboxes.

Data Bytes of CAN Status Message

Byte number	Description	Possible states
1	Error State of CAN 1	See table *CAN Error State in Bytes 1–4 of the CAN Status Message*
2	Error State of CAN 2	
3	Error State of CAN 3	
4	Error State of CAN 4	
5	Global device state	See table *Possible States in Bytse 5–8 of the CAN Status Message*
6	Configuration state	
7	Application state	
8	Action Rules state	

CAN Error State in Bytes 1–4 of the CAN Status Message

Error state	Description
0	Error active
1	Error passive
2	Bus off
3	Stopped
255	CAN not available

Possible States in Bytse 5–8 of the CAN Status Message

State	Description
1	Error occurred
2	Warning occurred
3	Module in state *preoperational*
4	Module in state *configuring*
5	Module in state *operational*
6	Module is not existent

5.7.3 Testing IF Events

To test if an IF event occurs, the terminal window can be used.

► Define an action rule (see *Defining a Rule, p. 34*).

► In column **THEN action** select action type **Write a terminal message**.

► Define the payload of the terminal message in field **Payload** and click button **OK**.

► Click button **Terminal** to open the terminal window.

→ If the defined **IF event** occurs, the defined message is written to the terminal window.

5.7.4 Verify Configured Action Rules

► Open menu **View** and select **Show status window**.

→ Status window is displayed in the CAN-Gateway Configurator.

► Click button **Verify**.

→ Status, errors and warnings are showed in the status window.

5.8 Mapping Table

The CAN-Gateway Configurator allows free routing configurations. Individual messages or message groups can be mapped from and to each CAN port. The route through the device always starts at the receiving CAN controller (message source) and ends at the transmitting CAN controller (message destination).

> **!** Only messages that are entered in the mapping table are forwarded. By default, no filter is set and all messages are rejected.

> **!** To be able to transmit CAN FD messages, CAN port 3 and CAN port 4 must be configured as CAN FD port (see *CAN Ports, p. 22*). If CAN FD messages are forwarded, the receiving CAN controller must be capable of CAN FD to be able to receive these messages.

> **!** Risk of multiply transmitted messages.
>
> Each message that is entered in the mapping table is transmitted. If a message is entered more than once (e.g. overlapping value range or multiply entries of the same message identifier), the message is transmitted for each entry in the list, i.e. the message is transmitted more than once.

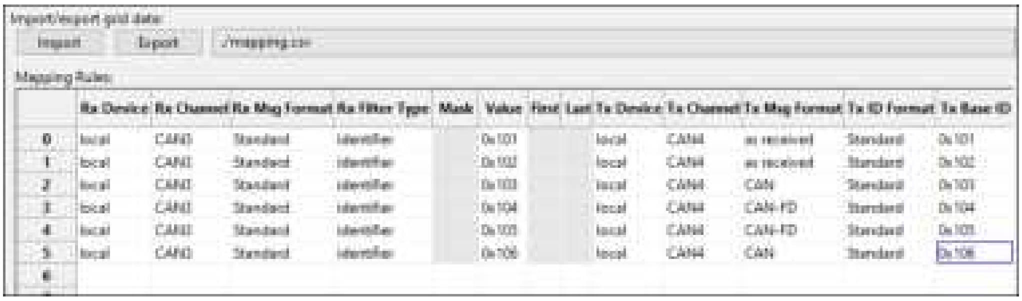
> *It is possible to configure the mapping table offline.*

Entry limitations

The size of the mapping table is limited.

Maximal possible entries:

- maximal 512 rows in total
- limitations of Extended format:
  - maximal 256 identifier entries
  - maximal 8 mask/value entries per CAN



	Rx Device	Rx Channel	Rx Msg Format	Rx Filter Type	Mask	Value	First	Last	Tx Device	Tx Channel	Tx Msg Format	Tx ID Format	Tx Base ID
0	local	CAN3	Standard	identifier		0x101			local	CAN4	as received	Standard	0x101
1	local	CAN3	Standard	identifier		0x102			local	CAN4	as received	Standard	0x102
2	local	CAN3	Standard	identifier		0x103			local	CAN4	CAN	Standard	0x103
3	local	CAN3	Standard	identifier		0x104			local	CAN4	CAN-FD	Standard	0x104
4	local	CAN3	Standard	identifier		0x105			local	CAN4	CAN-FD	Standard	0x105
5	local	CAN3	Standard	identifier		0x106			local	CAN4	CAN	Standard	0x106
6													

Fig. 15 Mapping table

5.8.1 Configuration

► Select **Mapping Table** in the configuration tree.

→ Form to specify the mapping table appears on the right side.

If new mapping tables are loaded, already available mapping table entries are deleted.

► To load an existing mapping table, click button **Import**.

→ Window **Select a File** is opened.

or

► Configure the routing for each group of CAN messages.

! In the extended format a range filter is not possible.

! If hexadecimal values are used, they must begin with 0x.

Example: 0x55

► To select a row left-click on the left number column.

► To edit the cell content, click on the cell.

→ Drop-down list is opened.

► To add a row, right-click on the left number column and select **Insert new cells**.

To verify a configured row:

► Open menu **View** and select **Show status window**.

→ Status window is displayed in the CAN-Gateway Configurator.

► Click button **Verify**.

→ Status, errors and warnings are showed in the status window.

► To save the configured mapping table as template, click button **Export**.

Possible Entries

Column	Possible entries
Rx Device	CANbridge NT: select **local**. CAN@net NT: Master, Slave 1, additionally with NT 420: Slave 2, Slave 3
Rx Channel	CAN1, CAN2, additionally with NT 420: CAN3, CAN4 Defines the receive channel.
Rx Msg Format	Standard, Extended Defines the format in which messages are received, standard (11 bit identifiers) or extended (29 bit identifier).
Rx Filter Type	Identifier, Mask/Value, Range; Range filter only possible in standard format
Mask	Used with Mask/Value filter, defines which bits of an identifier are relevant for the filter and which are not relevant, see *Mask/Value Filter, p. 42* (decimal and hexadecimal values possible).
Value/Identifier	With Mask/Value filter: defines the values for the filter relevant bits (as defined in **Mask**), see *Mask/Value Filter, p. 42* With Identifier filter: defines the identifier Decimal and hexadecimal values possible.
First	First value of range (decimal and hexadecimal values possible)
Last	Last value of range (decimal and hexadecimal values possible)

Column	Possible entries
Tx Device	CANbridge NT: select **local**. CAN@net NT: Master, Slave 1, additionally with NT 420: Slave 2, Slave 3
Tx Channel	CAN1, CAN2, additionally with NT 420: CAN3, CAN4 Defines the transmit channel. If remote access is enabled, it is possible to select **Remote** to transmit the defined messages via ASCII to the host.
Tx Msg Format	CAN (for Classic CAN messages), additionally with NT 420: CAN FD (for CAN FD messages), as received (messages are transmitted in the same format as received) CAN FD messages can only be transmitted on CAN 3 and CAN 4, and if the ports CAN 3 and CAN 4 are configured as CAN FD ports. If field *Tx Msg Format* is left empty, and CAN port 3 or CAN port 4 are configured as CAN FD port, all messages that are transmitted to these ports are converted into CAN FD message format.
Tx ID Format	Standard, Extended Defines the format in which messages are transmitted, standard (11 bit identifiers) or extended (29 bit identifier).
Tx Base ID	With Range and Mask/Value filter: specifies the transmit identifiers to which the received identifiers that passed the filter are mapped. With Identifier filter: specifies the transmit identifier

Examples Tx Base ID

The defined Rx identifiers pass the filter. These valid messages are then mapped to the transmit messages starting at the message identifier set in TX Base ID.

Filter type	Rx	Tx Base ID	Transmitted identifier
Range	First: 0x100 Last: 0x200	0x300	0x300–0x400
Mask/Value	Mask: 0x700 Value: 0x100	0x200	0x200–0x2FF
Identifier	0x123	0x456	0x456

5.8.2 Mask/Value Filter

With the Mask/Value filter (available for either 11 bit or 29 bit identifiers) possible valid identifiers based on bit masks can be defined.

Binary representation of mask:

- binary positions with value 1 are relevant for the filter
- binary positions with value 0 are not relevant for the filter

Binary representation of value:

- Defines the values for the positions that are marked as relevant (1) in mask.
- Values in positions that are marked as not relevant (0) in mask are ignored.

The following formula expresses the condition under which an identifier passes the filter:

- if (value & mask) == (identifier & mask) then identifier is valid

Examples

11 Bit Identifier		
	hex	**bin**
Value	0x700	`0111:0000:0000`
Mask	0x700	`0111:0000:0000`
Result	0x700	`0001:XXXX:XXXX`
	Any identifier between 0x700 and 0x7FF passes the filter, as only the first 3 bits of the mask are marked as relevant.	

29 Bit Identifier		
	hex	**bin**
Value	0x10003344	`0001:0000:0000:0000:0011:0011:0100:0100`
Mask	0x1F00FFFF	`0001:1111:0000:0000:1111:1111:1111:1111`
Result	0x10003344	`0001:0000:XXXX:XXXX:0011:0011:0100:0100`
	All identifiers with 0x10xx3344 (positions xx can be any number) pass the filter.	

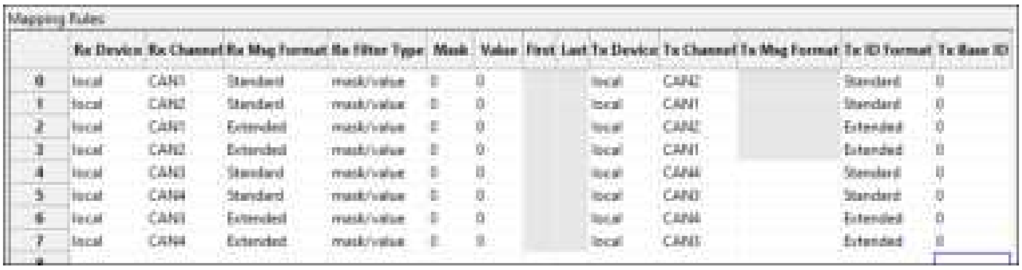
Mask/Value Filter		
Value	**Mask**	**Valid message identifiers which pass the filter**
0x100	0x7FF	0x100
0x100	0x700	0x100–0x1FF
0x000	0x000	0x000–0x7FF

5.8.3 Examples

Mapping Table Example 1

The following mapping table is an example of a CANbridge NT Bridge, that allows all messages (standard and extended) to pass:

- from CAN 1 to CAN 2 and vice versa
- from CAN 3 to CAN 4 and vice versa



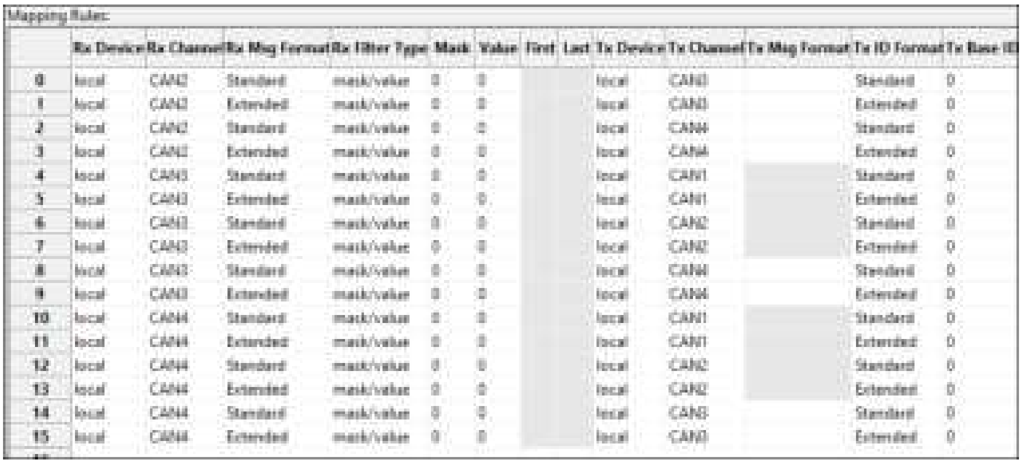
Mapping Rules:

	Rx Device	Rx Channel	Rx Msg Format	Rx Filter Type	Mask	Value	First	Last	Tx Device	Tx Channel	Tx Msg Format	Tx ID Format	Tx Base ID
0	local	CAN1	Standard	mask/value	0	0			local	CAN2		Standard	0
1	local	CAN2	Standard	mask/value	0	0			local	CAN1		Standard	0
2	local	CAN1	Extended	mask/value	0	0			local	CAN2		Extended	0
3	local	CAN2	Extended	mask/value	0	0			local	CAN1		Extended	0
4	local	CAN3	Standard	mask/value	0	0			local	CAN4		Standard	0
5	local	CAN4	Standard	mask/value	0	0			local	CAN3		Standard	0
6	local	CAN3	Extended	mask/value	0	0			local	CAN4		Extended	0
7	local	CAN4	Extended	mask/value	0	0			local	CAN3		Extended	0

Fig. 16 Example 1 mapping table

Mapping Table Example 2

The following mapping table is an example of a CANbridge NT Star Coupler, that allows all messages (standard and extended) to pass from every CAN port to every CAN port.



Mapping Rules:

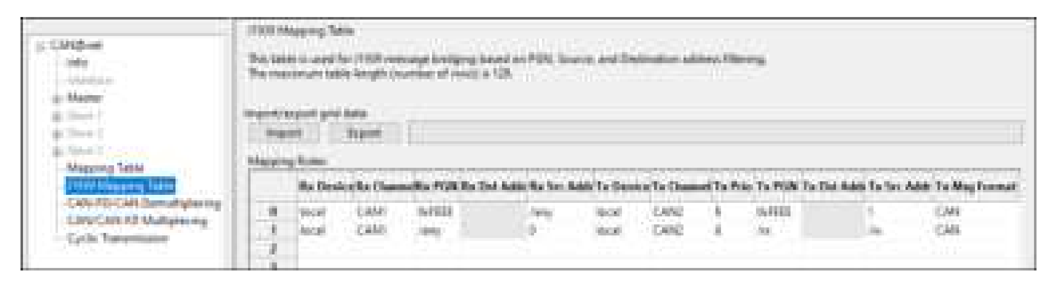
	Rx Device	Rx Channel	Rx Msg Format	Rx Filter Type	Mask	Value	First	Last	Tx Device	Tx Channel	Tx Msg Format	Tx ID Format	Tx Base ID
0	local	CAN2	Standard	mask/value	0	0			local	CAN3		Standard	0
1	local	CAN2	Extended	mask/value	0	0			local	CAN3		Extended	0
2	local	CAN2	Standard	mask/value	0	0			local	CAN4		Standard	0
3	local	CAN2	Extended	mask/value	0	0			local	CAN4		Extended	0
4	local	CAN3	Standard	mask/value	0	0			local	CAN1		Standard	0
5	local	CAN3	Extended	mask/value	0	0			local	CAN1		Extended	0
6	local	CAN3	Standard	mask/value	0	0			local	CAN2		Standard	0
7	local	CAN3	Extended	mask/value	0	0			local	CAN2		Extended	0
8	local	CAN3	Standard	mask/value	0	0			local	CAN4		Standard	0
9	local	CAN3	Extended	mask/value	0	0			local	CAN4		Extended	0
10	local	CAN4	Standard	mask/value	0	0			local	CAN1		Standard	0
11	local	CAN4	Extended	mask/value	0	0			local	CAN1		Extended	0
12	local	CAN4	Standard	mask/value	0	0			local	CAN2		Standard	0
13	local	CAN4	Extended	mask/value	0	0			local	CAN2		Extended	0
14	local	CAN4	Standard	mask/value	0	0			local	CAN3		Standard	0
15	local	CAN4	Extended	mask/value	0	0			local	CAN3		Extended	0

Fig. 17 Example 2 mapping table

5.9 J1939 Mapping Table

> To be able to transmit CAN FD messages, CAN port 3 and CAN port 4 must be configured as CAN FD port (see *CAN Ports, p. 22*) . If CAN FD messages are forwarded, the receiving CAN controller must be capable of CAN FD to be able to receive these messages.

> Risk of multiply transmitted messages.
>
> Each message that is entered in the mapping table is transmitted. If a message is entered more than once (e.g. overlapping value range or multiply entries of the same message identifier), the message is transmitted for each entry in the list, i.e. the message is transmitted more than once.



J1939 Mapping Table

This table is used for J1939 message bridging based on PGN, Source, and Destination address filtering. The maximum table length (number of rows) is 128.

Import/export grid data

Mapping Rules

	Rx Device	Rx Channel	Rx PGN	Rx Dst Addr	Rx Src Addr	Tx Device	Tx Channel	Tx Prio	Tx PGN	Tx Dst Addr	Tx Src Addr	Tx Msg Format
0	local	CAN1	[illegible]		*any	local	CAN2	6	[illegible]		1	CAN
1	local	CAN1	*any		0	local	CAN2	6	*rx		*rx	CAN
2												
3												

Fig. 18 J1939 Mapping example

Line 0 is an example for a PDU2 format message (no specific Rx destination address), received from any source address. Line 1 is an example to get all messages from Rx source address 0.

For information about SAE J1939 and the structure of the parameter group see www.ixxat.com/technologies/all4can/sae-j1939-technology.

For handling of the table see configuration in *Mapping Table, p. 40*.

Entry limitations

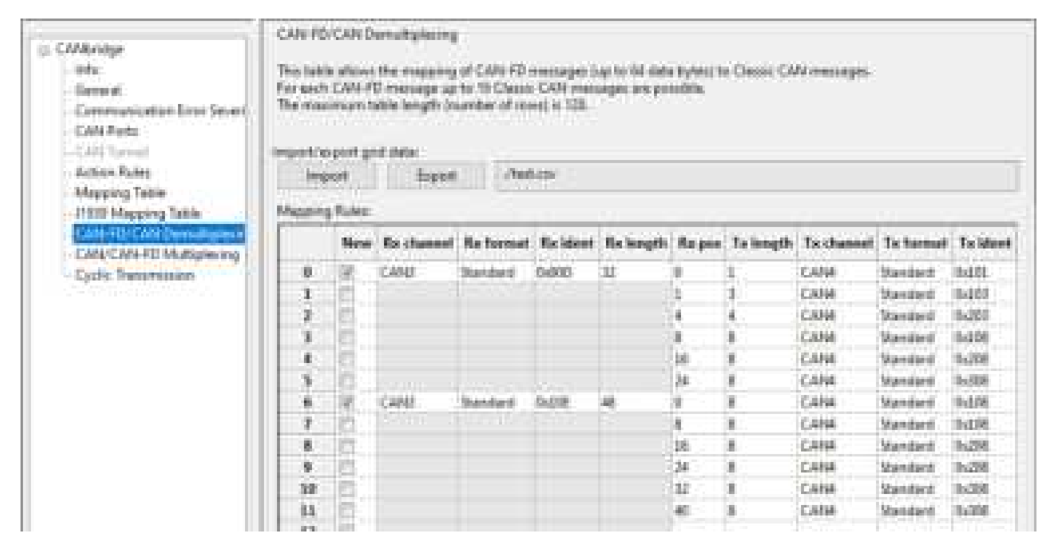
The size of the mapping table is limited. Maximal 128 rows are possible.

Possible entries

Column	Possible entries
Rx Device	CANbridge NT: select **local**. CAN@net NT: Master, Slave 1, additionally with NT 420: Slave 2, Slave 3
Rx Channel	CAN1, CAN2, additionally with NT 420: CAN3, CAN4 Defines the receive channel.
Rx PGN	Receive PGN or */any* Defines the receive PGN (18 bit) including Extended Data Page, Data Page and the PDU specific field for PDU2 format messages. Observe that for PDU1 the last two numbers must be 0.
Rx Dst Addr	*0–255* or */any* Destination address for the receive PGN (PDU1), deactivated if PDU2 format is defined in **RxPGN**
Rx Src Addr	*0–255* or */any* Source address of receive PGN
Tx Device	CANbridge NT: select **local**. CAN@net NT: Master, Slave 1, additionally with NT 420: Slave 2, Slave 3
Tx Channel	CAN1, CAN2, additionally with NT 420: CAN3, CAN4 Defines the transmit channel.
Tx Prio	Priority of transmitted message
Tx PGN	Transmit PGN or */rx* Defines the transmit PGN (18 bit) including Extended Data Page, Data Page and PDU specific field for PDU2 format messages. Observe that for PDU1 the last two numbers must be 0. If */rx* is entered, the value in **RxPGN** is used.
Tx Dst Addr	0–255 or */rx* Destination address for the transmit PGN (PDU1), deactivated if PDU 2 format is defined in **RxPGN**. If */rx* is entered, the value in **RxDst Addr** is used.
Tx Src Addr	0–255 or */rx* Source address of transmit PGN. If */rx* is entered, the value in **RxSrc Addr** is used.
Tx Msg Format	CAN (for Classic CAN messages), additionally with NT 420: CAN FD (for CAN FD messages), as received (messages are transmitted in the same format as received) CAN FD messages can only be transmitted on CAN3 and CAN4, and if the ports CAN3 and CAN4 are configured as CAN FD ports. If field *Tx Msg Format* is left empty, and CAN port 3 or CAN port 4 are configured as CAN FD port, all messages that are transmitted to these ports are converted into CAN FD message format.

5.10 CAN FD/CAN Demultiplexing

Demultiplexing is possible with the CANbridge NT 420 in Bridge mode and with the CAN@net NT 420 in Local CAN Bridge mode. The demultiplexing table allows to divide CAN FD messages of up to 64 data bytes in Classic CAN messages with up to 8 data bytes.



CAN-FD/CAN Demultiplexing

This table allows the mapping of CAN-FD messages (up to 64 data bytes) to Classic CAN messages.
For each CAN-FD message up to 19 Classic CAN messages are possible.
The maximum table length (number of rows) is 128.

Import/export grid data: ./test.csv

Mapping Rules:

	New	Rx channel	Rx format	Rx ident	Rx length	Rx pos	Tx length	Tx channel	Tx format	Tx ident
0	☑	CAN3	Standard	0x800	32	0	1	CAN4	Standard	0x101
1	☐					1	3	CAN4	Standard	0x103
2	☐					4	4	CAN4	Standard	0x203
3	☐					8	8	CAN4	Standard	0x108
4	☐					16	8	CAN4	Standard	0x208
5	☐					24	8	CAN4	Standard	0x308
6	☑	CAN3	Standard	0x208	48	0	8	CAN4	Standard	0x108
7	☐					8	8	CAN4	Standard	0x108
8	☐					16	8	CAN4	Standard	0x208
9	☐					24	8	CAN4	Standard	0x208
10	☐					32	8	CAN4	Standard	0x208
11	☐					40	8	CAN4	Standard	0x308

Fig. 19 Example demultiplexing table

For handling of the table see configuration in *Mapping Table, p. 40*.

Entry limitations

The size of the mapping table is limited. Maximal 32 CAN FD messages are possible.

Possible Entries

Column	Possible entries
New	Activated, Deactivated An activated checkbox defines the start of a new CAN FD message, that is to be divided.
Rx channel	CAN3, CAN4 Defines the receive channel of the CAN FD message.
Rx format	Standard, Extended Defines the format in which CAN FD messages are received, standard (11 bit identifiers) or extended (29 bit identifier).
Rx ident	Identifier of the CAN FD message to be divided
Rx length	Number of data bytes of the CAN FD message to be divided
Rx pos	Starting position in bytes (0..63) within source CAN FD message
Tx length	Number of data bytes of the Classic CAN message to be transmitted (up to 8 data bytes)
Tx channel	CAN1, CAN2, CAN3 (Classic CAN) and CAN4 (Classic CAN) Defines the transmit channel of the Classic CAN messages
Tx format	Standard, Extended Defines the format in which the Classic CAN messages are transmitted, standard (11 bit identifiers) or extended (29 bit identifier).
Tx ident	Identifier of the received Classic CAN message

5.11 CAN/CAN FD Multiplexing

Multiplexing is possible with the CANbridge NT 420 in Bridge mode and with the CAN@net NT 420 in Local CAN Bridge mode. The multiplexing table allows to map up to 8 Classic CAN messages into one CAN FD message.

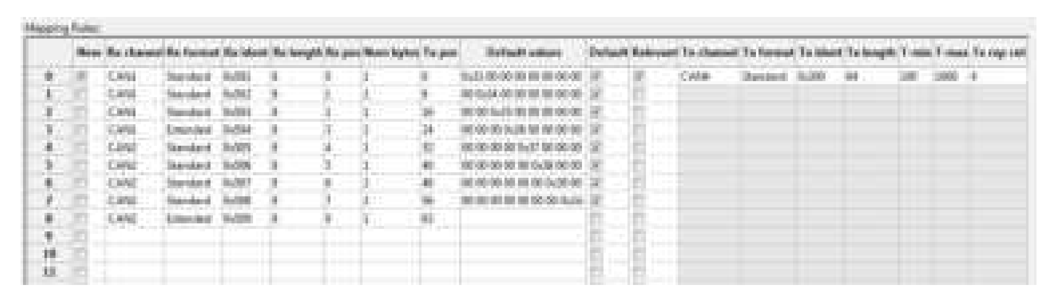


Fig. 20 Example multiplexing table

For handling of the table see configuration in *Mapping Table, p. 40*.

Entry limitations

The size of the mapping table is limited. Maximal 32 CAN FD messages are possible.

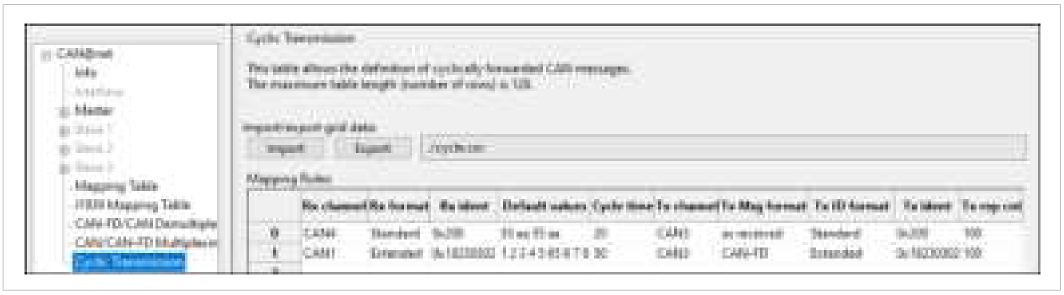
Possible Entries

Column	Possible entries
New	Activated, Deactivated An activated checkbox defines the start of a new CAN FD message, that is to be transmitted.
Rx channel	CAN1, CAN2, CAN3 (Classic CAN) and CAN4 (Classic CAN) Defines the receive channel of the Classic CAN message.
Rx format	Standard, Extended Defines the format in which Classic CAN messages are received, standard (11 bit identifiers) or extended (29 bit identifier).
Rx ident	Identifier of the Classic CAN message to be received
Rx length	Number of data bytes of the Classic CAN message to be received
Rx pos	0–7 Start position to copy from (a part of the Classic CAN message can be selected to be transmitted)
Num bytes	0–8 Number of bytes to be copied
Tx pos	0–63 Position in the CAN FD message to copy the Classic CAN message to
Default values	If the receive message fails, the default values are transmitted instead (if **Default** is activated). Number of data bytes has to match Rx length.
Default	Activated: If the receive message fails, the defined default values are transmitted. Deactivated: If the receive message fails, former values are transmitted.
Relevant	Activated: If the Classic CAN message fails or the cycle time is violated, the transmitting of the CAN FD message is cancelled after the number of transmit repetitions defined in Tx rep cnt is expired.
Tx channel	CAN3, CAN4 Defines the transmit channel of the CAN FD message.
Tx format	Standard, Extended Defines the format in which CAN FD messages are transmitted, standard (11 bit identifiers) or extended (29 bit identifier).
Tx ident	Identifier of the transmitted CAN FD message
Tx length	Number of data bytes of the CAN FD message to be transmitted (up to 8 data bytes)
T-min	0–65000

Column	Possible entries
	Blocking time in milliseconds (minimum cycle time) between two CAN FD messages. Timer starts with reception of the first message after power on. The CAN FD message is not transmitted earlier, independent of the receive frequency of Classic CAN messages.
T-max	1–65000 Maximum cycle time between two CAN FD messages. The CAN FD message is transmitted latest after the defined time independent of whether all Classic CAN messages are received or not.
Tx rep cnt	1–65000 Maximum number of transmit repetitions of the CAN FD message, if Classic CAN messages marked as relevant are received with incomplete data. The transmitting is started again after all relevant messages are received in the time frame T-max * Tx rep cnt. If no message is marked as relevant, the feature is deactivated.

5.12 Cyclic Transmission

> **!** To be able to transmit CAN FD messages, CAN port 3 and CAN port 4 must be configured as CAN FD port (see *CAN Ports, p. 22*) . If CAN FD messages are forwarded, the receiving CAN controller must be capable of CAN FD to be able to receive these messages.



Cyclic Transmission

This table allows the definition of cyclically forwarded CAN messages.
The maximum table length (number of rows) is 128.

Import/export grid data:

./cyclic.csv

Mapping Rules:

	Rx channel	Rx format	Rx ident	Default values	Cycle time	Tx channel	Tx Msg format	Tx FD format	Tx ident	Tx rep cnt
0	CAN4	Standard	0x200	11 aa 11 aa	20	CAN3	as received	Standard	0x200	100
1	CAN1	Extended	0x18230002	1 2 3 4 5 6 7 8	30	CAN3	CAN-FD	Extended	0x18230002	100

Fig. 21 Cyclic transmission

Cyclic transmission is possible with the CANbridge NT in Bridge mode and with the CAN@net NT in Local CAN Bridge mode.

It is possible to send CAN messages cyclically and precisely timed. A configured CAN message is transmitted once in the defined cycle time. Even if more or less of the configured CAN message are received.

By changing the cycle time the following settings are possible:

- To reduce the number of CAN messages that are transmitted to the receiver, the cycle time can be increased.
- To increase the number of CAN messages that are transmitted to the receiver, the cycle time can be reduced.

If no current CAN message is received, it is possible to transmit default messages or earlier transmitted messages to the receiver.

To automatically stop the cyclic transmission a repetition counter can be defined. The repetition counter is decremented after each transmission of a CAN messages. When the counter reaches the value 0 the default values are used (if specified) for one further transmission and then the cyclic transmission is stopped.

The cyclic transmission is started with the reception of the first message.

Entry limitations

The size of the mapping table is limited. Maximal 128 rows are possible.

For handling of the table see configuration in *Mapping Table, p. 40*.

Possible Entries

Column	Possible entries
Rx channel	CAN1, CAN2, additionally with NT 420: CAN3, CAN4 Defines the receive channel.
Rx format	Standard, Extended Defines the format in which messages are received, standard (11 bit identifiers) or extended (29 bit identifier).
Rx ident	Identifier of the message to be received
Default values	If the receive message fails, the default values are transmitted instead.
Cycle time	Message cycle time in units of 0.5 ms, valid values: 1–65535 (= 0.5 ms to 32767.5 ms)
Tx channel	CAN1, CAN2, additionally with NT 420: CAN3, CAN4 Defines the transmit channel.
Tx Msg Format	CAN (for Classic CAN messages), additionally with NT 420: CAN FD (for CAN FD messages), as received (messages are transmitted in the same format as received) CAN FD messages can only be transmitted on CAN3 and CAN4, and if the ports CAN3 and CAN4 are configured as CAN FD ports. If field *Tx Msg Format* is left empty, and CAN port 3 or CAN port 4 are configured as CAN FD port, all messages that are transmitted to these ports are converted into CAN FD message format.
Tx ID format	Standard, Extended Defines the format in which messages are transmitted, standard (11 bit identifiers) or extended (29 bit identifier).
Tx ident	Identifier of the transmitted message
Tx rep cnt	Maximum number of transmit repetitions if the receive message is missing. Valid values: 0-65535. Value 0 sets endless transmission. If the counter expires, the cyclic message is stopped.

6 Dashboard

With the dashboard that is integrated in the CAN-Gateway Configurator it is possible to keep track of transmitted messages in the software, and, for example, to monitor in the receive counters and transmit counters of the various modules if all transmitted messages are received or if messages are lost.

- ► To open the dashboard, click button **Dashboard** in the toolbar.
  - – Rx counter counts all messages that are received by the respective module.
  - – Tx counter counts all messages that are transmitted by the respective module.
  - – Module memory displays the size of each memory pool and the available free entries.
  - – Modules Master 1, Master 2, and Master 3 display the connection from the Master to the respective Slave 1, 2, and 3.
- ► For more information about each module, see the mouseover help text in each module.

7 Command Line Program

The integrated command line program *CanGWfile* works via USB and with the CAN@net NT also via TCP. The command line tool is available for Windows and for Linux (cangwfile).

Output when Started Without Parameters

```
####  Ixxat CAN-Gateway File Utility V6.00  ####

Syntax:
  CanGWfile TCP <IP-address or 'any'> <command>
                         [<file-type> <file-name>] [<options>]

  CanGWfile USB <serial-no or 'any'>  <command>
                         [<file-type> <file-name>] [<options>]

Examples:
  CanGWfile USB HW906505 w CONF conf.txt
  CanGWfile TCP 196.168.178.20 re ERR error.txt
  CanGWfile TCP 196.168.178.20 i
Possible commands:
  w  - write file to target device
  wv - write file with verify
  r  - read file from target device
  re - read and erase file on target device
  i  - read device identification
  s  - scan and output found devices
  h  - output the historical program calls
Possible file types: CONF, ERR, IPC, CXML, HFU, LUA
  In case of 'CXML', the CONF file will be generated and
     additionally written.
Possible options:
  -p<password>  - device password for devices with security level >= 2
  -init         - re-initialize the device to activate the flashed files
  -reset        - reset the device to activate the flashed files
  -terminal     - for Lua ADK outputs to the terminal
  -s<serial-no> - requested serial number for TCP network devices
```

The following commands can be processed:

- reading the device identification (command i)
- reading and deleting the error memory and log files (command re)
- reading and writing a configuration (command r/w)
- writing and verifying a configuration (command wv)
- reading the file that saves all calls including parameters (command h), file is automatically stored on the PC and can also be opened and deleted manually
- scanning for devices (command s)

File types:

- CXML: device configuration created and saved by the CAN-Gateway Configurator (*.cxml)
- CONF: device configuration exported by the CAN-Gateway Configurator as txt file
- ERR: error/log file
- IPC: file for the IP configuration (can be uploaded, changed and downloaded)
- HFU: HMS firmware update file to update the firmware (available on product support pages on www.ixxat.com/support-bridges-gateways)

Descriptions of possible options:

- init: restarting the device and activating the loaded configuration (option for Lua scripts)
- terminal: after writing the configuration the program changes to terminal mode and displays all terminal outputs from the device on the screen (option for Lua scripts). Can be cancelled with any key.
- s<serial-no>: if the IP address of the device is unknown or if several devices are addressed with “any”, it is possible to address exactly one device with the serial number.

8 Reset to Factory Settings

It is possible to reset the configuration of a connected device to factory settings.

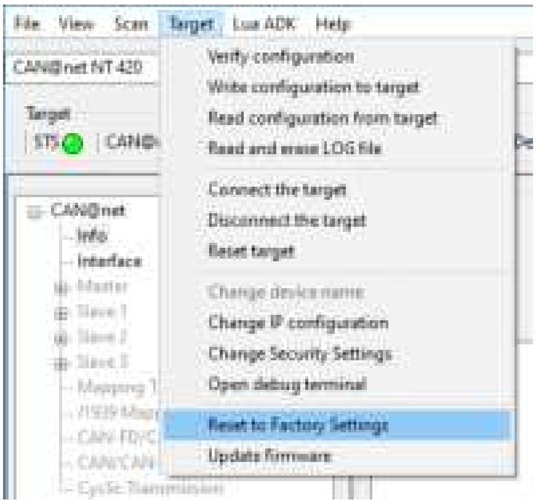
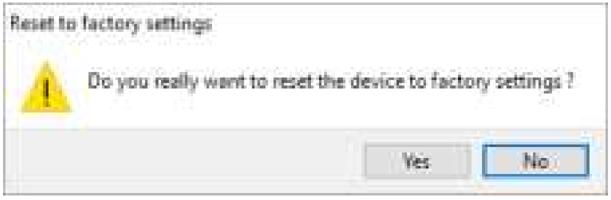


Fig. 22 Menu Reset to factory settings

- ► Make sure that the device is connected via USB.
- ► Open menu **Target** and select **Reset to Factory Settings**.


Reset to factory settings

Do you really want to reset the device to factory settings ?

- ► Click button **Yes** to confirm the reset.

9 Security Settings

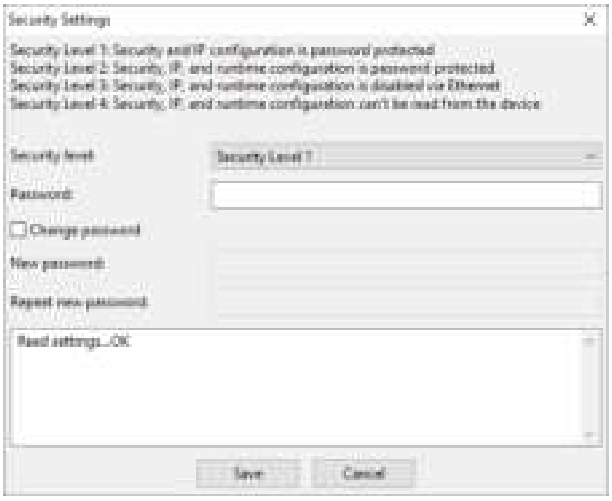
The CAN-Gateway Configurator has three security levels.

The default device password is IXXAT.

HMS Industrial Networks recommends to change the default password.

Action	Security level 1	Security level 2	Security level 3	Security level 4
Changing the password, security level and IP configuration via USB	Only with password	Only with password	Only with password	Only with password
Changing the password, security level and IP configuration via Ethernet	Only with password	Only with password	Locked	Locked
Changing the runtime configuration and updating the firmware (CODE, FPGA) via USB	Permitted	Only with password	Only with password	Locked
Changing the runtime configuration and updating the firmware (CODE, FPGA) via Ethernet	Permitted	Only with password	Locked	Locked
Reading a file (log, config, cxml, lua) via USB	Permitted	Permitted	Permitted	Locked
Reading a file (log, config, cxml, lua) via Ethernet	Permitted	Permitted	Locked	Locked

- To change the security level, open menu **Target** and select **Change Security Settings**.



Security Settings

Security Level 1: Security and IP configuration is password protected
Security Level 2: Security, IP, and runtime configuration is password protected
Security Level 3: Security, IP, and runtime configuration is disabled via Ethernet
Security Level 4: Security, IP, and runtime configuration can't be read from the device

Security level: Security Level 1

Password:

□ Change password

New password:

Repeat new password:

Read settings...OK

Fig. 23 Security Settings

If the device is connected via Ethernet and the security level is set to 3 or 4, the access via Ethernet is locked. For changes the device then has to be connected via USB.

This page intentionally left blank

A Lua License

Copyright © 1994–2019 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

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.

Box 4126
300 04 Halmstad, Sweden

info@hms.se