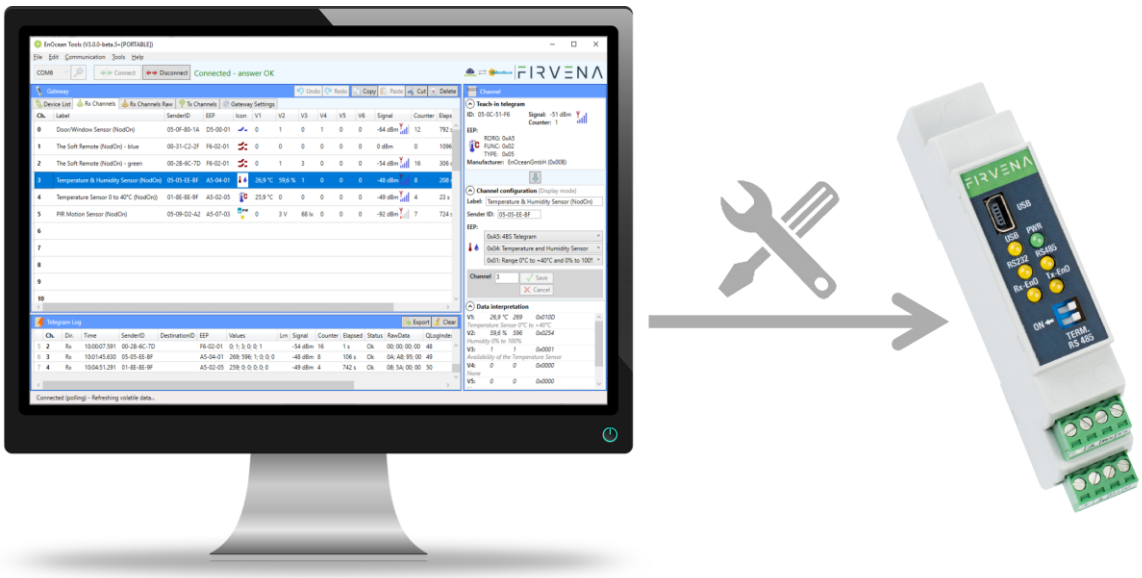
EnOcean Tools User Manual

Introduction

The application EnOcean Tools is a utility intended to be used for configuring the gateway EnOcean / Modbus RTU. It helps you assign EnOcean elements to the gateway and map them to the Modbus holding registers. The application is also a useful verification tool whereby you can evaluate whether your system works well. EnOcean Tools reads and displays the data stored in the Modbus registers and allows to see the states, measured quantities, communication intervals or signal strength of the connected EnOcean elements. There is also a log of incoming and outgoing telegrams in the application.
This manual describes features of the EnOcean Tools application. For a detailed description of the gateway and its Modbus interface, refer the gateway manual.



FIRVENA

Contents

Figures

Tables

1 Communication with Gateway

To communicate with the gateway, the application uses the COM port and Modbus RTU protocol. The application always behaves like a MASTER (sends queries) and the gateway like a SLAVE (responds to queries from the application).
The gateway has three communication ports: USB, RS485 and RS232 port. Although the USB port is designated for configuring the gateway, the RS485 and RS232 port can also be used thanks to a single communication interface.

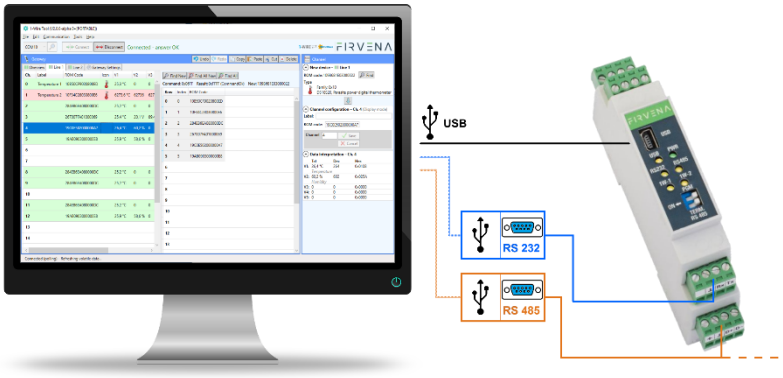


Fig. 1.1 Common connections

1.1 Driver

The USB interface is implemented using an integrated circuit from FTDI. Communication via the USB port needs a driver to be installed on your computer that causes the gateway appears as an additional COM port. The operating system usually performs the driver installation automatically when a gateway is first connected. Alternatively, the driver can be downloaded from the manufacturer's website and installed manually (*http://www.ftdichip.com/Drivers/VCP.htm*).
The operating system assigns a unique COM port during the installation. If more than one gateway is used on the computer, each gateway has different port assigned because the port is assigned to a particular FTDI chip. The COM port can be changed in the Device Manager.
For the RS485 and RS232 port, a USB-to-serial converter is usually needed. If this is the case, the driver and virtual COM port is related to the converter and does not depend on the connected gateway.

1.2 Communication Settings

Complete communication settings are available through the *Communication/Communication Settings* menu item (F11).

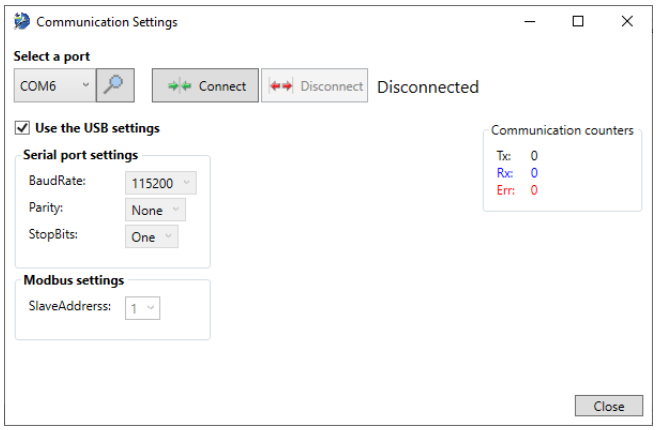


Fig. 1.2 Communication settings overview

1.2.1 USB

Only the COM port must be selected, the remaining settings are fixed for the USB port.
To start communicating with a gateway:

1. Connect the gateway through USB port to your computer
2. Select the COM port
3. Tick the checkbox "*Use the USB settings*"
4. Press "*Connect*" (F3)

If everything is all right, the message *"Connected – answer OK"* is shown.

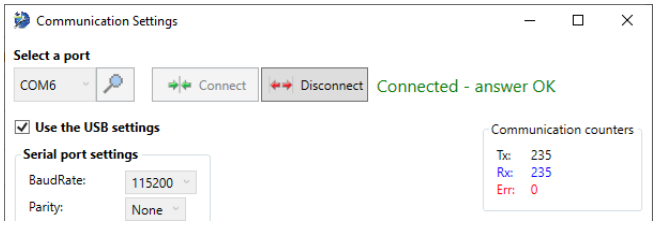


Fig. 1.3 Communication works properly

Otherwise, if communication does not work properly, a red-highlighted error message is shown. Check the communication settings and physical connection and try again.

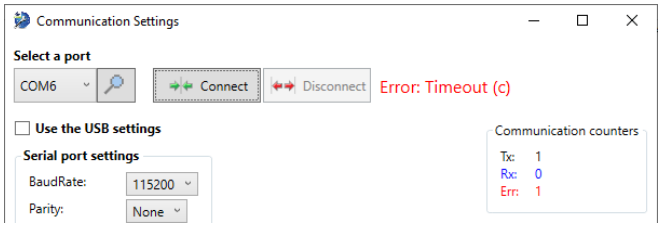


Fig. 1.4 Communication error

***Note:** If there are many COM ports available on the computer, the following procedure can be used to determine the right one:*

1. *Connect the gateway to the computer (or disconnect if it has been already connected)*
2. *Using the F5 key update the list of available ports*
3. *Check out what COM port appears in the list (or disappears)*

1.2.2 RS485 and RS232

The COM port is not associated with the gateway, rather it is associated with a device that is used to access a serial network that is the gateway connected to (e.g. a USB-to-serial converter). Compared to the USB connection, it is also necessary to set parameters of the serial network (*Serial port settings*) and the address of the gateway within the network (*Modbus settings – SlaveAddress*).
To start communicating with a gateway:

1. Connect the gateway through RS485 port or RS232 to your computer
2. Select the COM port
3. Uncheck the checkbox "*Use the USB settings*" to enable further settings
4. Perform the enabled communication settings
5. Press "*Connect*" (F3)

1.3 Getting the Data

The application gets the data automatically when communication is running. First, the configuration registers are read then the application switches to the state where the volatile registers are being looped. The division into the configuration and volatile registers is discussed in the following table. The range corresponds to addresses that are actually read by the application.

Tab. 1 Configuration versus volatile registers

Configuration data – it is supposed only the MASTER can change this data Loaded after the *"Connect"* command executed, then if necessary.		
Area Name	**Range**	**Description**
RxID	400…799	Configuration of Rx channels
TxID	5000…6199	Configuration of Tx channels
RxLabels	10000…13999	Descriptions of Rx channels
TxLabels	14000…19999	Descriptions of Tx channels
Volatile data – this data may change without the participation of MASTER Being updated periodically after configuration reading done.		
RxTelegramQueue	900…939	FIFO memory of recently received Rx telegrams
TxTelegramQueue	960…983	FIFO memory of recently sent Tx telegrams
ServiceRegisters	1000…1069	Data and settings concerning the gateway itself
RxValues	0…399	Last received data
RxRawData	2000…2399	Last received data in raw format

2 Description of the User Interface

In one point of view, the application can be thought as an editor of gateway configuration. This configuration can be stored either in the memory of a gateway or in a file. This chapter introduces the main parts of the user interface.

2.1 Main Window Overview

1. Communication control – basic control of communication with the gateway.
2. Application status – shows the state that the application is currently in.
3. Gateway – an editor of gateway configuration (see chapter 2.4).
4. Teach-in telegram – the last teach-in telegram received by the connected gateway (see chapter 2.5.1).
5. Channel configuration – an editor of the selected Rx or Tx channel (see chapter 2.5.2).
6. Data interpretation – displays the value registers of the selected Rx or Tx channel (see chapter 2.5.3).
7. Telegram log – records incoming and outgoing telegrams (see chapter 2.6).

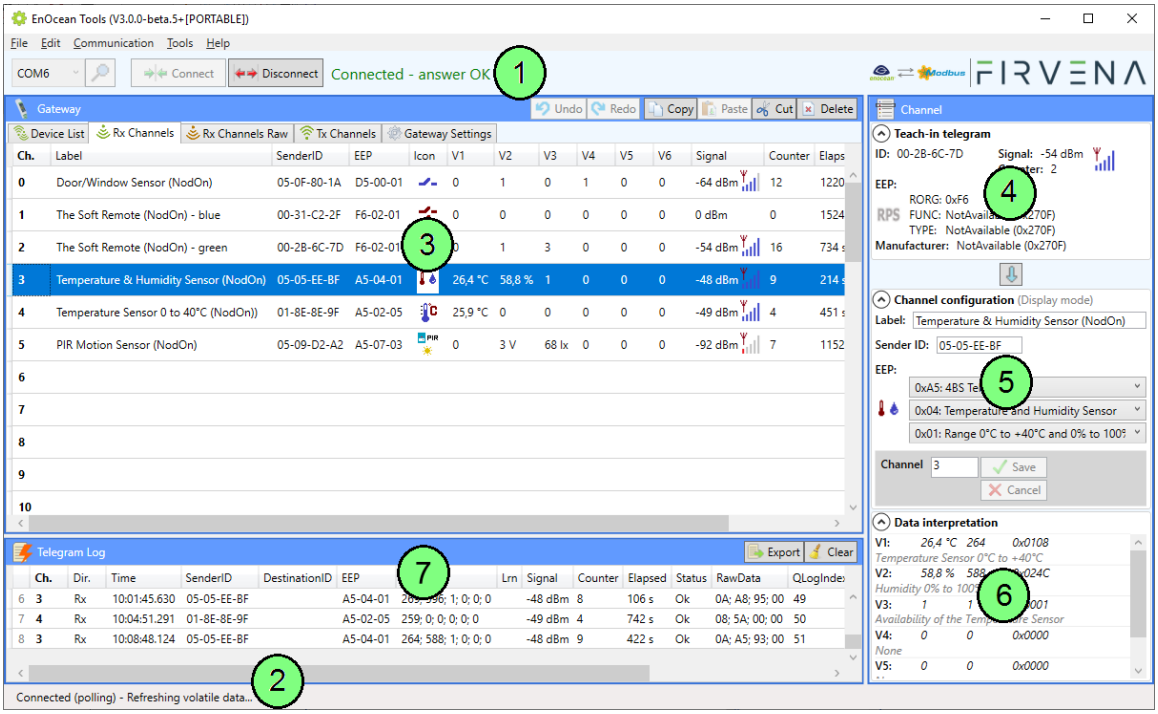


Fig. 2.1 Main window overview

***Note**: The main window can operate under two modes – online and offline. The online mode occurs when there is a gateway connected. In the online mode, overall configuration changes made by the editor are written directly to the gateway, the volatile data (e.g. V1…V6, Teach-in telegram, Telegram Log, etc.) is available and new elements can be assigned. Transmitting channels can also be controlled (Tx telegrams can be sent). In the offline mode, it works only as a configuration file editor and the main window works the same way as the editor window (see chapter 2.2).*

2.2 Editor Window Overview

1. Editor status – shows the state that the editor is currently in.
2. Gateway – an editor of gateway configuration.
3. Teach-in telegram – the last teach-in telegram received by the connected gateway.
4. Channel configuration – an editor of the selected Rx or Tx channel.

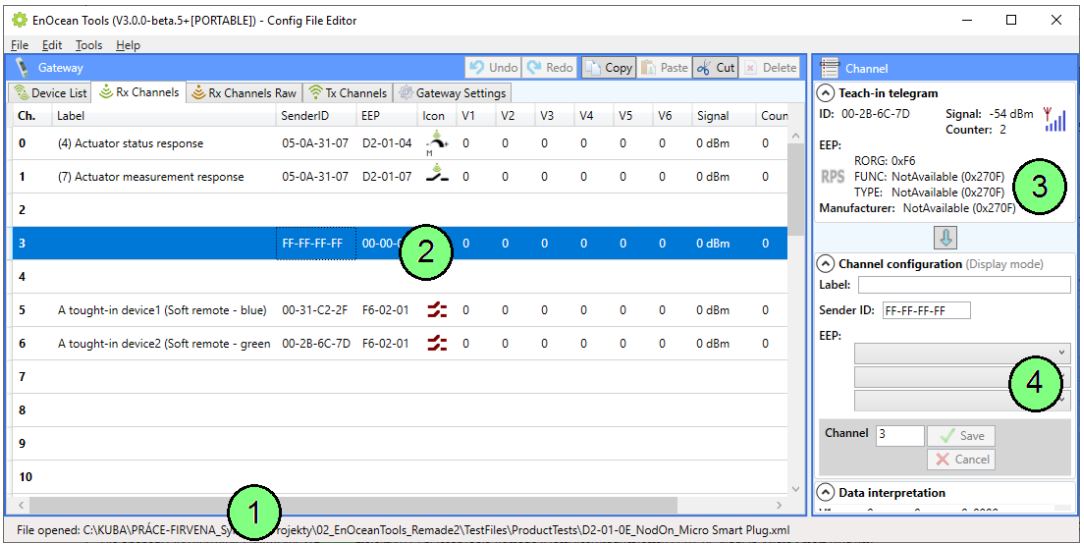


Fig. 2.2 Editor window overview

***Note**: The volatile data displayed (e.g. V1…V6, Signal, etc.) are meaningless in the editor window except the Teach-in telegram where the data are being taken from the main window. This behavior enables the teach-in data received by the connected gateway to be used within the editor window. Other data representing volatile registers are set to their default values (zeros).*

One of the possible uses of the editor window is opening or creating a template configuration file. Such a file can include example configurations or reusable pieces of configuration. Items (channels) can be copied between the windows, as shown in Fig. 2.3.

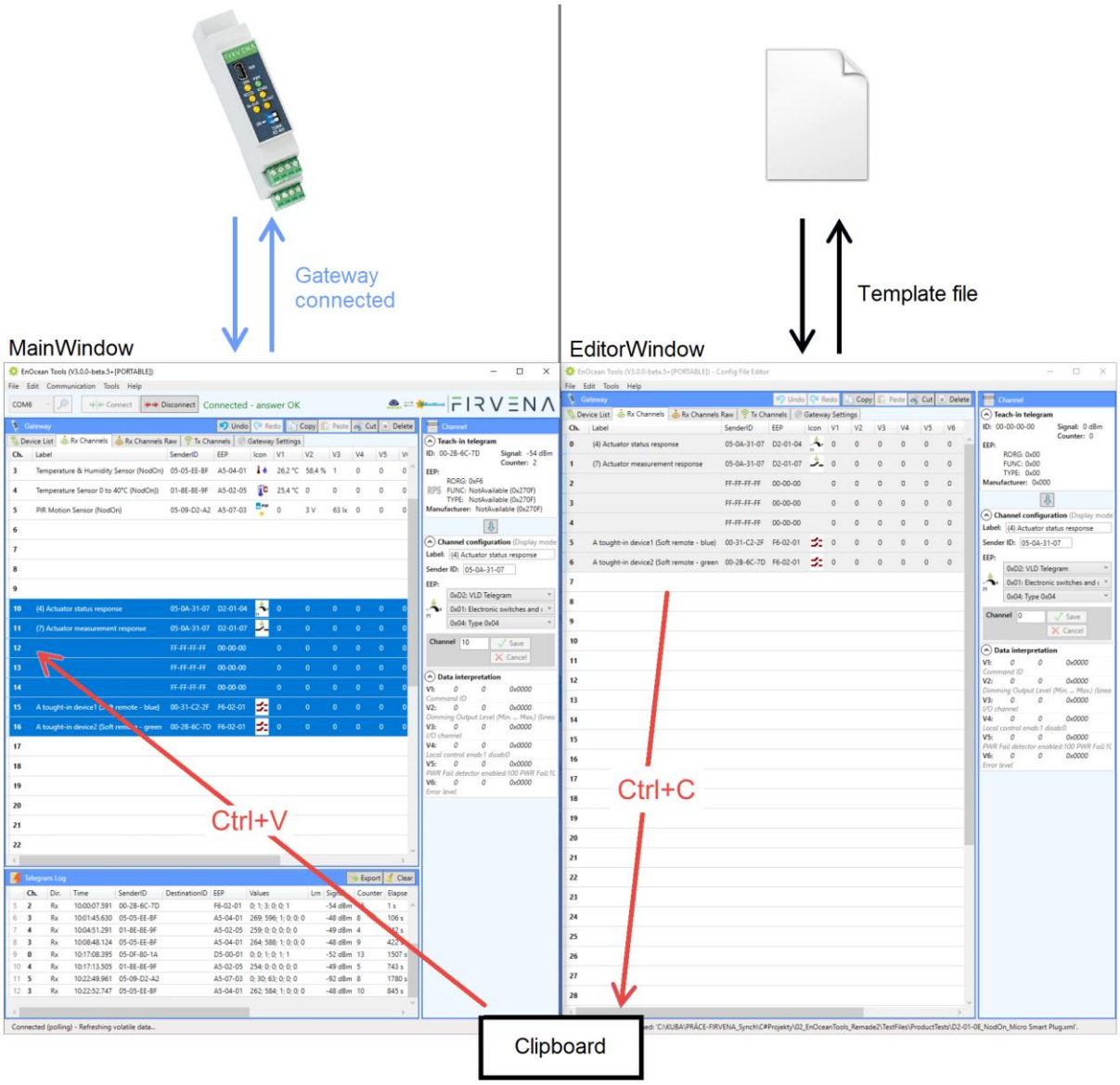


Fig. 2.3 Copying channels between windows.

2.3 Menu Options

2.3.1 File

Tab. 2 Menu – File options

Item	Meaning
New	Creates an empty, unsaved configuration file and opens it in another window.
Open	Opens an existing configuration file. Then the file can be written to the connected gateway or shown in another window.
Open in Editor	Opens an existing configuration file directly in another window.
Save	Saves changes to the open configuration file.
Save As	Saves the open configuration file as another file.
Close	Forces the application to close.

2.3.2 Edit

Tab. 3 Menu – Edit options

Item	Meaning
Undo	Undoes the last change made to the gateway configuration.
Redo	Undoes the last undo action.
Copy	Copies the selected Rx or Tx channels to the clipboard.
Paste	Writes the copied Rx or Tx channels, if there are any. The writing starts at the first selected channel of the current selection.
Cut	Copies and deletes the selected Rx or Tx channels.
Delete	Deletes the selected Rx or Tx channels.

2.3.3 Communication

Tab. 4 Menu – Communication options

Item	Meaning
Connect	Starts communicating with the gateway. (Enters the online editor mode)
Disconnect	Stops communicating with the gateway. (Leaves the online editor mode)
Communication Settings	Settings of communication between the application and gateway.

2.3.4 Tools

Tab. 5 Menu – Tools options

Item	Meaning
Firmware Loader	Opens a firmware update utility. (see chapter 4)
Clear Gateway	Deletes all Rx and Tx channels.

2.3.5 Help

Tab. 6 Menu – Help options

Item	Meaning
View Help	The manual.
About	Information about the application.
View Changelog	Brief information about differences between the current and recent versions.

2.4 Gateway Window

This window, together with the *Channel* window, can be considered an editor of gateway configuration. It is used to display and edit the configuration stored either in the connected gateway or in the open file. This chapter explains the meaning of individual parts.

2.4.1 Device List Tab

This view is of particular use to manage those EnOcean elements that are able of bidirectional communication or require multiple channels to be used. It allows to operate both on receiving and transmitting channels.
There is a tree list in the tab where the channels are sorted into groups called *"Device"*. The channels having the same EnOcean ID stored are assumed to belong to the same device (the ID corresponds to the Sender ID for Rx channels and Destination ID for Tx channels).

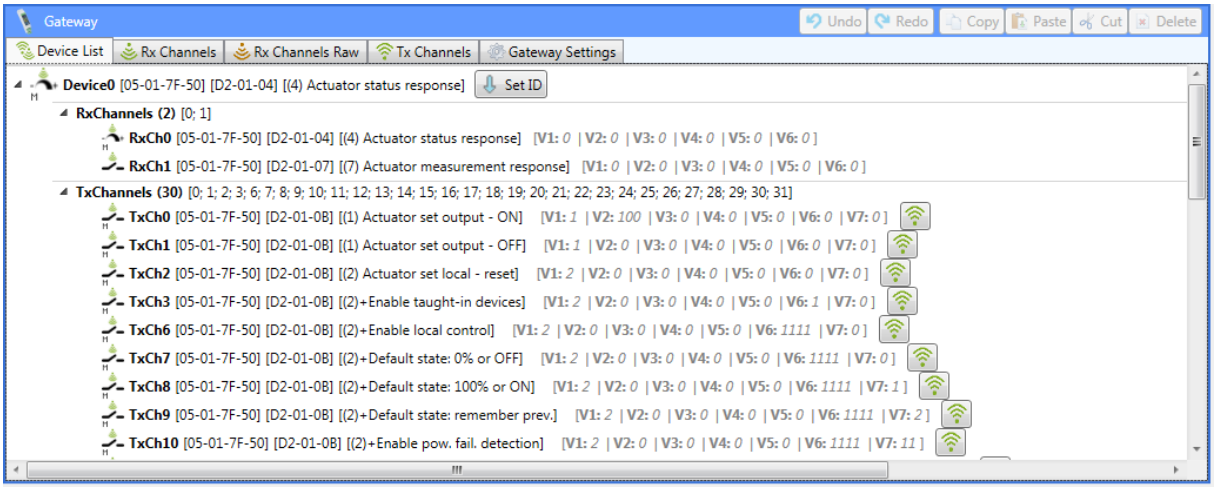


Fig. 2.4 Device List Tab

2.4.2 Rx Channels Tab

This view is to manage the receiving channels. The configuration and incoming data are merged in a datagrid. The individual columns are listed and further described in the table below. The configuration registers can be changed by editing the corresponding field in the datagrid, as seen in the *"Editable"* column. The table also shows which registers each column is related to.

Tab. 7 Rx Channels – the meaning of columns (n is a channel number from 0 to 39)

Header	Description	Editable	Holding Registers
Ch.	Channel number	✗	n
Label	A description up to 40 characters, without diacritics	✓	10000…10099 + 100n
SenderID	Sender ID (the device that sent the telegram) *(Format: "ID3-ID2-ID1-ID0", hexadecimal)*	✓	400…403 + 10n
EEP	EnOcean Equipment Profile, this setting allows the gateway to extract the values from raw data *(Format: "RORG-FUNC-TYPE", hexadecimal)*	✓	404…406 + 10n
Icon	A pictogram representing the EEP	✗	
V1…V6	The last received data *(Format: decimal number or interpretation string)*	✗	0…5 + 10n
Signal	Signal strength in dBm	✗	6 + 10n
Counter	Counter of received telegrams	✗	7 + 10n
Elapsed	Time elapsed since the last telegram was received	✗	8 + 10n
Status	Channel status *(Format: decimal (string))*	✗	9 + 10n

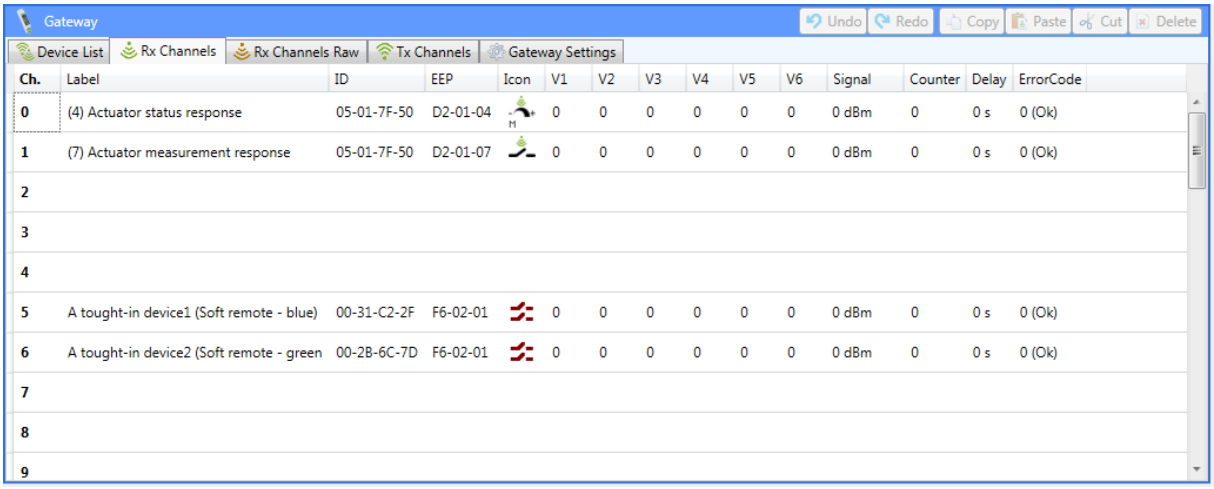


Fig. 2.5 Rx Channels Tab

2.4.3 Rx Channels Raw Tab

In this tab, the raw data of Rx channels can be watched (registers 2000 to 2399). The individual columns are listed and further described in the table below. There are not any editable columns. As far as the mapping of registers concerned, note that the mapping varies across different RORG settings, this is also covered in the table.

Tab. 8 Rx Channels Raw – the meaning of columns (n is a channel number from 0 to 39)

Header	Description	Editable	Holding Registers
Ch.	Channel number	✗	n
Label	A description up to 40 characters, without diacritics	✗	10000…10099 + 100n
SenderID	Sender ID (the device that sent the telegram) *(Format: "ID3-ID2-ID1-ID0", hexadecimal)*	✗	2000…2003 + 10n[1)] 2000…2001 + 10n[2)]
RORG	Radio Organization (radio telegram type), received value *(Format: "RORG", hexadecimal)*	✗	2004 + 10n[1)] 2002 + 10n[2)]
Data	Raw data received *(Format for 1BS, RPS: "DB0 00 00 00", for 4BS: "DB0 DB1 DB2 DB3"; hexadecimal)*	✗	2005…2008 + 10n[1)]
	for VLD telegram e.g. of 5 data bytes: "DB0 DB1 DB2 DB3 DB4 00 00 00 00 00 00 00 00 00"		2003…2009 + 10n[2)]
Status	Status byte received in the telegram *(Format: "Status", hexadecimal)*	✗	2009 + 10n[1)]
	Not available for VLD and MSC		[2)]

1) *For RORG = 0xA5, 0xD5, 0xF6 (4BS, 1BS, RPS telegrams)*
2) *For RORG = 0xD2, 0xD1 (VLD, MSC telegrams)*

***Note**: The mapping depends on the RORG as configured in the register 404 + 10n, not on the RORG received together with a data telegram (and displayed in the RORG column).*

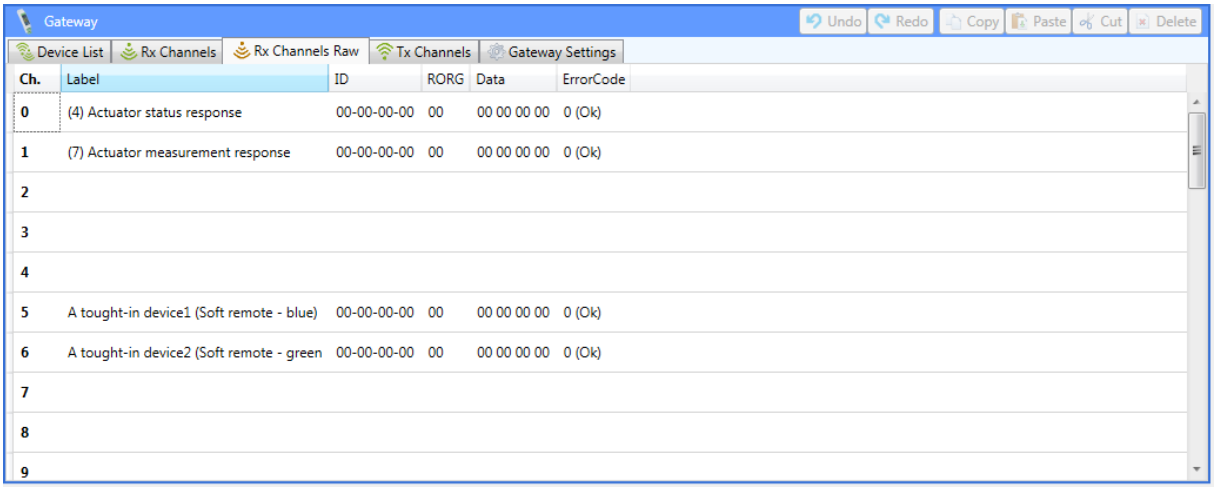


Fig. 2.6 Rx Channels Raw Tab

2.4.4 Tx Channels Tab

This view is to manage the transmitting channels (registers 5000 to 6199). The datagrid involves both configuration and outgoing data. The individual columns are listed and further described in the table below. The configuration registers can be changed by editing the corresponding field in the datagrid, as seen in the *"Editable"* column. The table also shows which registers each column is related to.

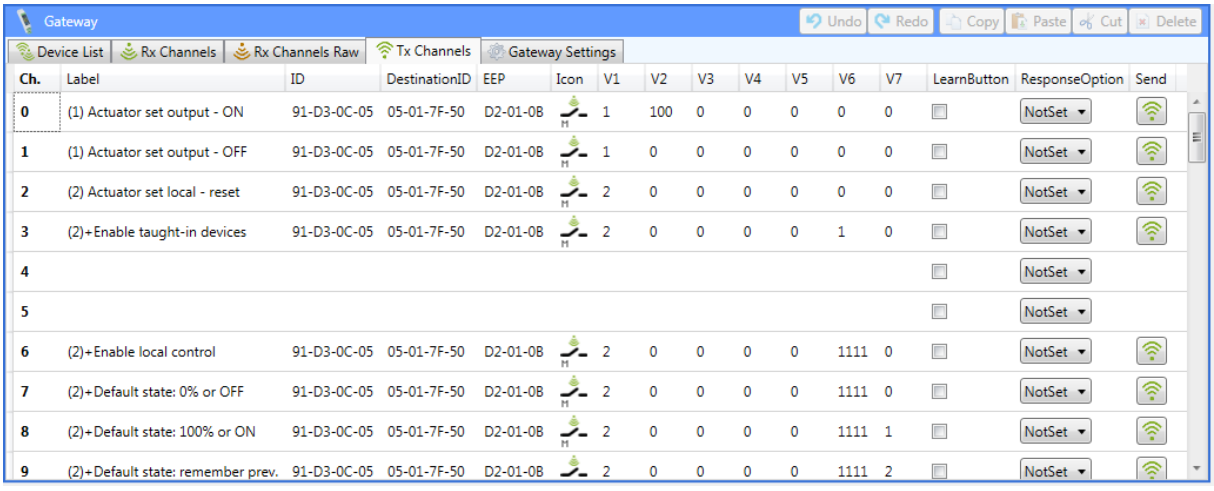
Tab. 9 Tx Channels tab – the meaning of columns (n is a channel number from 0 to 59)

Header	Description	Editable	Holding Registers
Ch.	Channel number	✗	n
Label	A description up to 40 characters, without diacritics	✓	14000…14099 + 100n
SenderID	Sender ID (the device that sends the telegram = the gateway), both Chip ID or Base ID can be used – see chapter 6.1 for explanation of this setting *(Format: "ID3-ID2-ID1-ID0", hexadecimal)*	✓	5000…5003 + 20n
DestinationID	Destination ID (the device that is supposed to receive the telegram) *(Format: "ID3-ID2-ID1-ID0", hexadecimal)*	✓	5004…5007 + 20n
EEP	EnOcean Equipment Profile, this setting allows the gateway to compile the values to raw data *(Format: "RORG-FUNC-TYPE", hexadecimal)*	✓	5008…5010 + 20n
Icon	A pictogram representing the EEP	✗	
V1…V7	The data to be sent *(Format: decimal number or interpretation string)*	✓	5011…5017 + 20n

LearnButton	Switches between the teach-in and data telegram (works only for certain EEPs, see the gateway manual)	✓	5018 + 20n
ResponseOption	Specifies when the telegram to be sent (see Tab. 10)	✓	5019 + 20n
Send	Button for immediate transmit, it is the same as setting the response option to *"SendNow"*	✗	5019 + 20n

Tab. 10 Explanation of the ResponseOption enumeration

Identifier = corresponding value	Command description	Purpose
NotSet = 0	Do nothing.	
SendNow = 1	Send this telegram immediately and set ResponseOption = NotSet.	Testing
ResponseOnReceivedOnce = 2	Send this telegram when a telegram from the target device received (target given by the DestinationID) and set ResponseOption = NotSet.	Bidirectional communication
ResponseOnReceivedAlways = 3	Send this telegram on every telegram from the target device received (target given by the DestinationID).	Bidirectional communication
UteResponseOnce = 4	Send UTE response on the next UTE query, fill in the DestinationID and EEP registers and set ResponseOption = NotSet.	Adding of a device that implements the Universal Bidirectional Teach-in procedure (UTE).
SendThisNowAndNextAfter{x}ms = m (x = 100 + 50(m – 101), m is from 101 to 111)	Send this telegram, then the next after the specified interval and set ResponseOption = NotSet. (e.g. send channel 1 -> wait 250ms -> send channel 2).	Simulating the RPS communication.



Ch.	Label	ID	DestinationID	EEP	Icon	V1	V2	V3	V4	V5	V6	V7	LearnButton	ResponseOption	Send
0	(1) Actuator set output - ON	91-D3-0C-05	05-01-7F-50	D2-01-0B		1	100	0	0	0	0	0		NotSet	
1	(1) Actuator set output - OFF	91-D3-0C-05	05-01-7F-50	D2-01-0B		1	0	0	0	0	0	0		NotSet	
2	(2) Actuator set local - reset	91-D3-0C-05	05-01-7F-50	D2-01-0B		2	0	0	0	0	0	0		NotSet	
3	(2)+Enable taught-in devices	91-D3-0C-05	05-01-7F-50	D2-01-0B		2	0	0	0	0	1	0		NotSet	
4														NotSet	
5														NotSet	
6	(2)+Enable local control	91-D3-0C-05	05-01-7F-50	D2-01-0B		2	0	0	0	0	1111	0		NotSet	
7	(2)+Default state: 0% or OFF	91-D3-0C-05	05-01-7F-50	D2-01-0B		2	0	0	0	0	1111	0		NotSet	
8	(2)+Default state: 100% or ON	91-D3-0C-05	05-01-7F-50	D2-01-0B		2	0	0	0	0	1111	1		NotSet	
9	(2)+Default state: remember prev.	91-D3-0C-05	05-01-7F-50	D2-01-0B		2	0	0	0	0	1111	2		NotSet	

Fig. 2.7 Tx Channels Tab

2.4.5 Gateway Settings Tab

This view visualizes the so-called service registers (registers 1000 to 1099), these are the registers concerning the settings and state information of the gateway itself.
Overview (Fig. 2.8):

1. An editor of the configurable service registers
2. A listing of all service registers

The changes made in the editor must be confirmed with the *"Save"* button to store the new values. To restore the default values, use *"Restore Defaults"* –> *"Save"*.

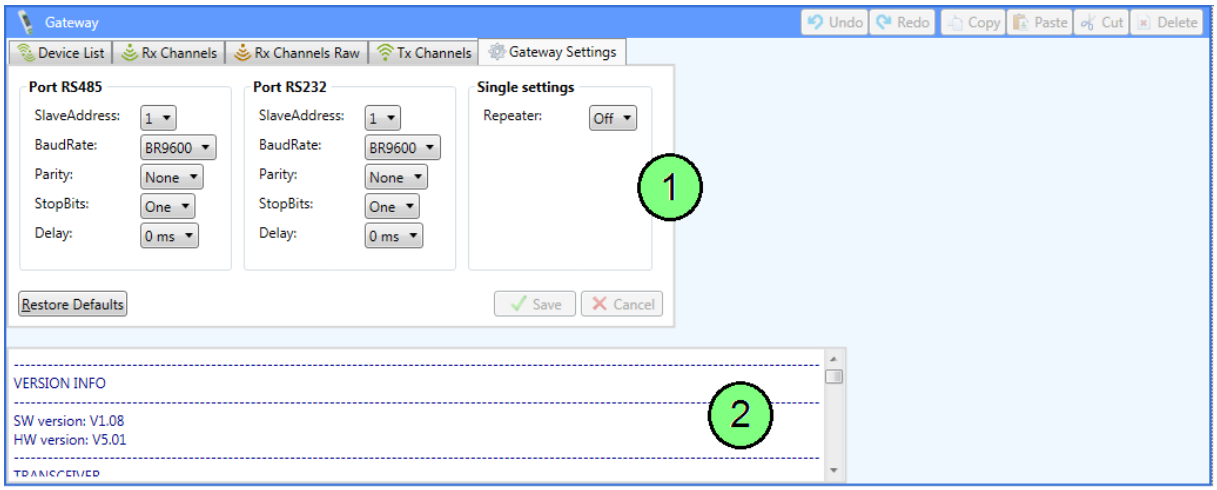


Fig. 2.8 Gateway Settings Tab

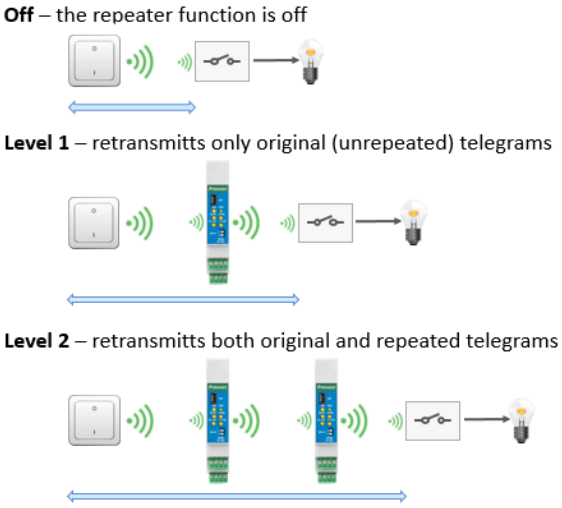


Fig. 2.9 Explanation of the Repeater enumeration

2.5 Channel Window

2.5.1 Teach-in Telegram

Teach-in data is displayed in this box that was received in the last teach-in telegram. The registers from 1012 to 1021 are utilized to get the data. The teach-in data can be used in the *"Channel configuration*" box to configure the selected Rx or Tx channel.

Tab. 11 Teach-in telegram – the meaning of fields

Header	Description	Holding Registers
ID	Sender ID (the device that sent the telegram = a device to be tought-in) *(Format: "ID3-ID2-ID1-ID0", hexadecimal)*	1012…1015
EEP	EnOcean Equipment Profile *(Format: "RORG-FUNC-TYPE", hexadecimal)*	1016…1018
Manufacturer	Manufacturer ID	1019
Signal	Signal strength in dBm	1020
Counter	Number of telegrams received from the same device	1021

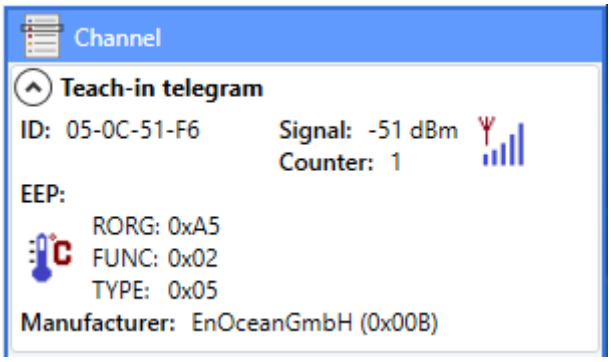


Fig. 2.10 Teach-in telegram

2.5.2 Channel Configuration

This box serves as a channel editor and is different for Rx and Tx channels. There are two states:

1. **Display mode**: Configuration of the currently selected channel is shown.
2. **Editor mode**: Appears when there is an unsaved change made to the content of the box. It happens either by editing the content or by receiving a new teach-in telegram. The display is locked, and channel selection only affects the channel number where the content is to be saved.

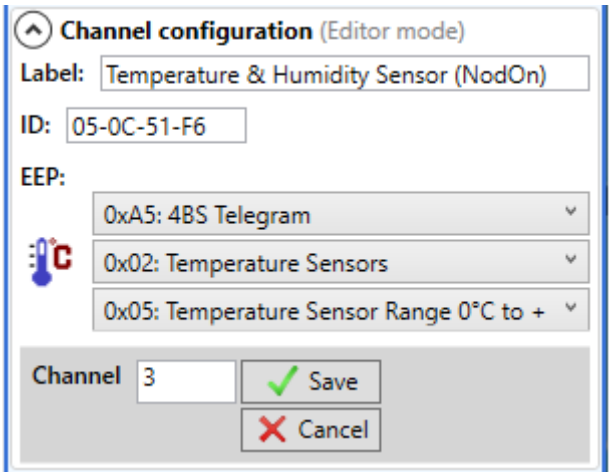


Fig. 2.11 Channel configuration – having unsaved changes

2.5.3 Data Interpretation

Displays the value registers of the selected Rx or Tx channel in three formats: interpretation string, decimal number and hexadecimal number. A description explaining each value is also shown. *"None"* means that the register is not used.

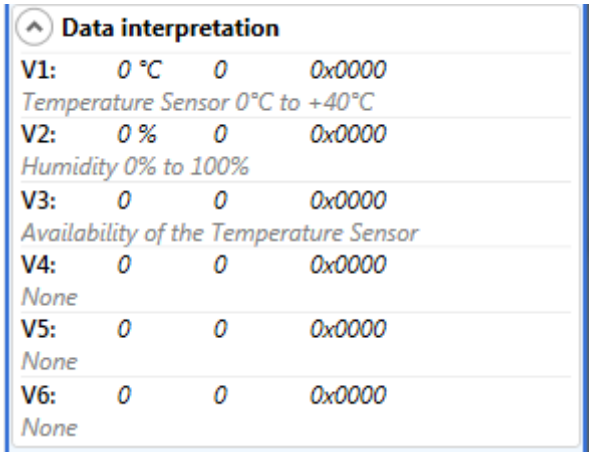


Fig. 2.12 Data interpretation

The value descriptions are also available through tooltips in *"Device List"*, *"Rx Channels"* and *"Tx Channels"* tab.

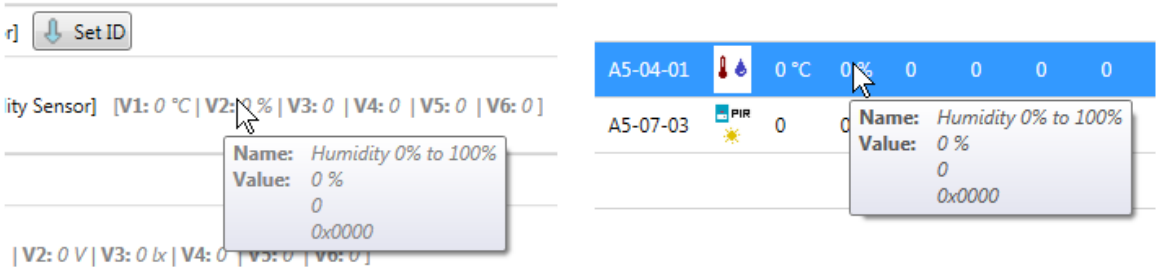


Fig. 2.13 The tooltips explaining a value

2.6 Telegram Log Window

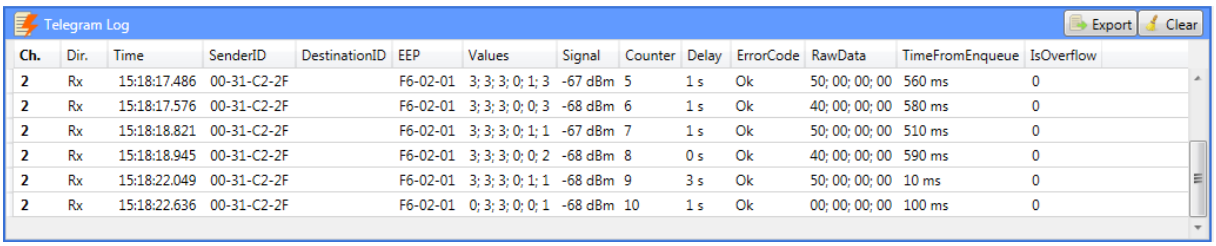
The telegram log records incoming and outgoing radio telegrams. This is achieved by utilizing the FIFO memory of received and sent telegrams that is located within the address range 900…939 and 960…983, respectively. Individual columns of the datagrid are listed and further described in the table below. The table also shows which registers each column is related to.

Tab. 12 Telegram Log – the meaning of columns

Header	Description	Holding Registers	
		Rx	Tx
	Row index		
Ch.	Channel number	900	960
Dir.	Direction		
Time	Estimated event time[3)]		
SenderID	Sender ID (the device that sent the telegram) *(Format: "ID3-ID2-ID1-ID0", hexadecimal)*	911...914	961…964
DestinationID	Destination ID (the device that is the telegram for), **only Tx** *(Format: "ID3-ID2-ID1-ID0", hexadecimal)*		965…968
EEP	EnOcean Equipment Profile *(Format: "RORG-FUNC-TYPE", hexadecimal)*	915…917	969…971
Values	The data received / sent (**Rx** / **Tx**) *(Format: decimal numbers)*	901…906	972…978
Lrn	LearnButton, **only Tx**		979
Signal	Signal strength in dBm, **only Rx**	907	
Counter	Counter of telegrams, **only Rx**	908	
Elapsed	Time elapsed since the last telegram, **only Rx**	909	
Status	Channel status, **only Rx** *(Format: string)*	910	
RawData	Raw data received, **only Rx** *(Format for 1BS, RPS: "DB0 00 00 00", for 4BS: "DB0 DB1 DB2 DB3", for VLD telegram of 5 data bytes: "DB0 DB1 DB2 DB3 DB4 00 00 00 00 00 00 00"; hexadecimal)*	935…938[1)] 933…938[2)]	
QLogIndex	The order of the record	921	982
QIsFull	The value of 1 indicates that the queue is out of memory, thereby some records have been lost.	920	982
QStopwatch	A time in milliseconds measured from the telegram received / sent.	919	981
QCount	Number of remaining records.	918	980
Date	Event date		

1) *RORG = 0xA5, 0xD5, 0xF6 (4BS, 1BS, RPS telegrams)*
2) *RORG = 0xD2 (VLD telegram)*

3) *The event time is obtained using a value read from the register 919 / 981 (QStopwatch). The value is subtracted from the system time recorded at the time close before a data query was sent to read the queue.*



Ch.	Dir.	Time	SenderID	DestinationID	EEP	Values	Signal	Counter	Delay	ErrorCode	RawData	TimeFromEnqueue	IsOverflow
2	Rx	15:18:17.486	00-31-C2-2F		F6-02-01	3; 3; 3; 0; 1; 3	-67 dBm	5	1 s	Ok	50; 00; 00; 00	560 ms	0
2	Rx	15:18:17.576	00-31-C2-2F		F6-02-01	3; 3; 3; 0; 0; 3	-68 dBm	6	1 s	Ok	40; 00; 00; 00	580 ms	0
2	Rx	15:18:18.821	00-31-C2-2F		F6-02-01	3; 3; 3; 0; 1; 1	-67 dBm	7	1 s	Ok	50; 00; 00; 00	510 ms	0
2	Rx	15:18:18.945	00-31-C2-2F		F6-02-01	3; 3; 3; 0; 0; 2	-68 dBm	8	0 s	Ok	40; 00; 00; 00	590 ms	0
2	Rx	15:18:22.049	00-31-C2-2F		F6-02-01	3; 3; 3; 0; 1; 1	-68 dBm	9	3 s	Ok	50; 00; 00; 00	10 ms	0
2	Rx	15:18:22.636	00-31-C2-2F		F6-02-01	0; 3; 3; 0; 0; 1	-68 dBm	10	1 s	Ok	00; 00; 00; 00	100 ms	0

Fig. 2.14 Telegram Log

2.6.1 Context Menu

Tab. 13 Telegram Log – context menu

Item	Meaning
Export Log	Allows saving of the current log to file. *Supported data formats: XML*
Clear Log	Discards the current log.
Autoscroll	If checked, the datagrid is scrolled down automatically whenever a new record is logged in.

3 How to Configure Channels

3.1 Assigning EnOcean elements – unidirectional

This teach-in procedure is only applicable for unidirectional communication when the gateway only receives data.
Let's have a sensor we want to assign to the channel 0. The procedure is as follows (see Fig. 3.1):

1. In the "Rx Channels" tab, select the channel 0
2. Push the pairing button to transmit a teach-in telegram (see note 1).
3. The received teach-in telegram is displayed in the *"Channel"* panel.
4. Click "Save" to confirm changes.
5. Now the sensor is assigned to channel 0 and its data is available through the Modbus interface.

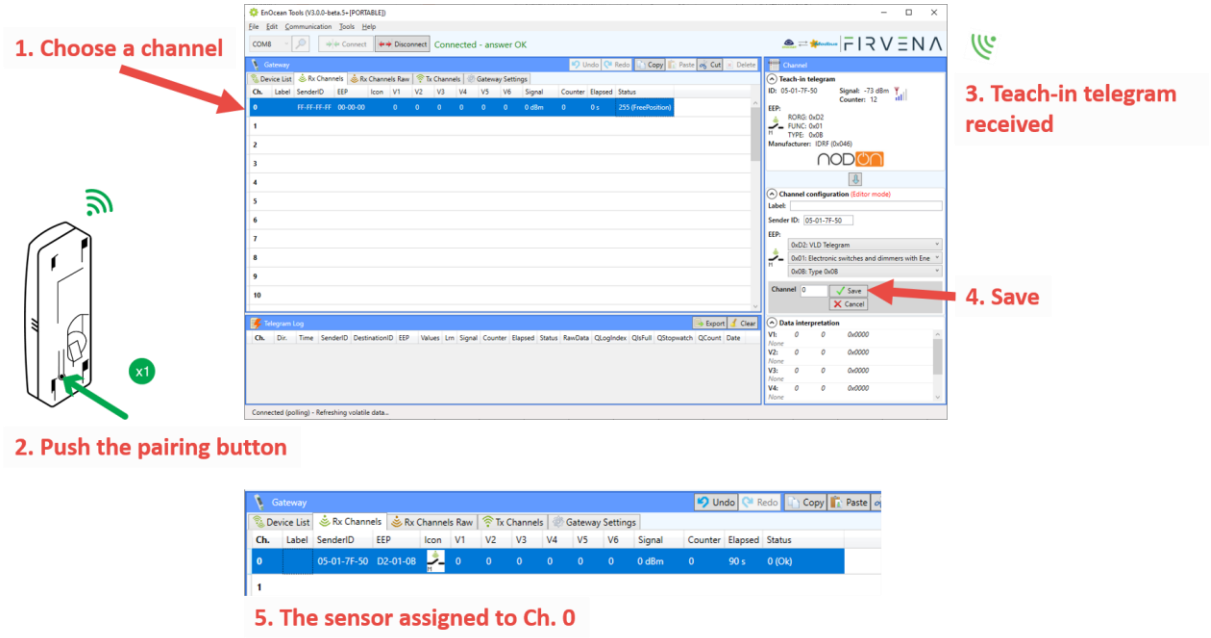


Fig. 3.1 Adding elements – teach-in procedure

Notes:

1) The point 2 may be manufacturer specific, refer to the datasheet or manual of the EnOcean device.

Optionally, the Label can be set for the device. The Label is used in the RxLabels registers – see Tab. 1.
The knowledge of EEP allows the gateway to interpret the received data correctly. Some types of EnOcean devices do not provide the EEP information in their teach-in telegram or even not have a special telegram for teach-in (e.g. buttons and switches). In that case, the EEP must be set manually, it is usually given by a label on the device or a datasheet.
Devices can also by assigned manually by entering the Sender ID and EEP.

3.2 Assigning EnOcean elements – bidirectional – 4BS

The **4BS teach-in Variation 3** is a type of communication that devices of type **EEP A5-xx-xx** use for bidirectional pairing. It is used by valve actuators such as **A5-20-01** and **A5-20-06**. This teach-in procedure is applicable for EnOcean devices with bidirectional communication profiles when the gateway is supposed to receive data from the device and also transmit data to the assigned device.

Let's have a valve actuator (A5-20-06) we want to pair with the gateway. We use Rx channel 0 for receiving data and Tx channel 0 for sending commands.

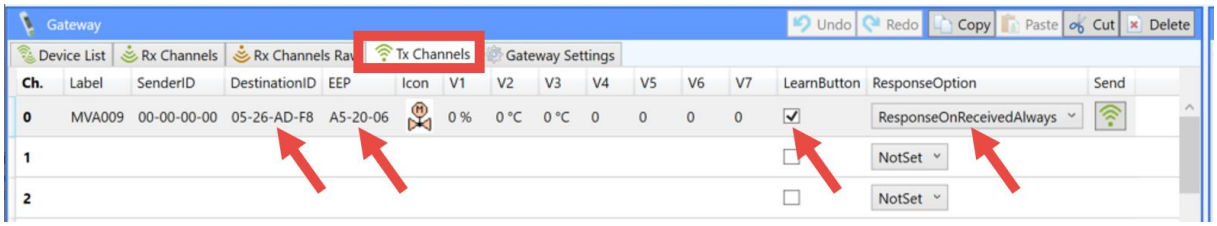
First, we have to define a Tx channel:

1. In the "Tx Channels" tab, select the channel 0
2. Put the actuator to the pairing mode (see note 1)
3. The actuator will report pairing error, but we get its ID and EEP
4. Optionally, set the "Sender ID" and confirm with "Save" (see note 2 and chapter 6.1)

Now, we can proceed with bidirectional pairing:

1. Tick the *LearnButton* and set *ResponseOption=ResponseOnReceivedAlways*

Channel 0 is in the teach-in mode now. It should look like this:

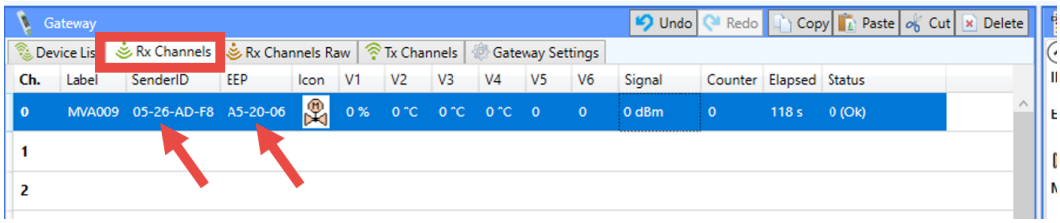


2. Put the actuator to the pairing mode again
3. The actuator should signalize successful teach-in
4. Untick the *LearnButton*
5. Now the actuator can receive commands sent by the gateway.

To receive the actual status, assign the actuator to Rx channel:

1. Go to "Rx Channels" tab, data from the previous teach-in should be automatically filled in the Channel configuration box.
2. Select the channel 0 and "Save"

Rx channel should look like this:



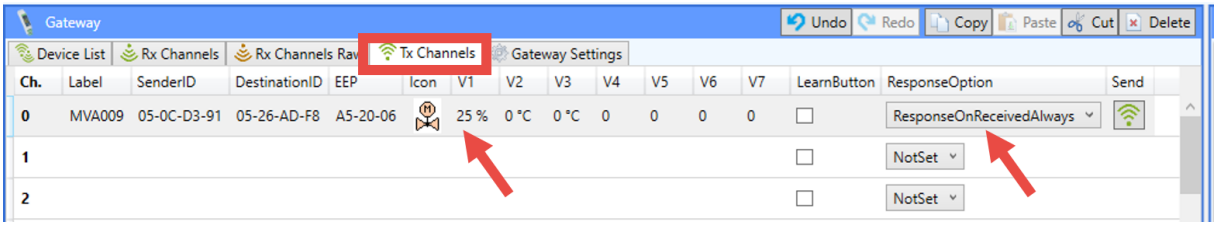
Notes:

1) Follow the procedure given by the manufacturer of the actuator.

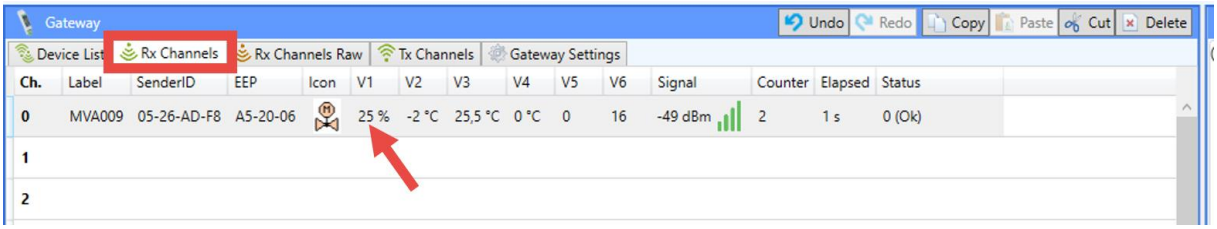
2) This ID will be sent in the teach-in response to the controller. The "Sender ID" uniquely identifies the Tx channel or gateway. Leave it 00-00-00-00 to use unique ID (EURID) of the EnOcean transceiver that is in the gateway. Or select one of the possible Base ID's.

Communication test:

1. In the Tx Channels tab, set V1 (Valve Position), e.g. 25 %, and leave the setting *ResponseOption=ResponseOnReceivedAlways*



2. Invoke communication test on the valve side (e.g. button press combination, manufacturer specific) or wait for the valve to wake up.
3. The valve should send the actual status telegram and update its position to 25 %
4. Check the reported status in the Rx Channels (see note 1)
5. Invoke communication test again
6. Now, the reported V1 (Actual Position) in the Rx Channels tab should also be 25 %



Notes:

1) The value sent by the valve is always delayed by one message because the valve first sends its status and then the gateway responds with the new value that is set in the Tx Channels.

3.3 Assigning EnOcean elements – bidirectional – UTE

The UTE (Universal Teach-in) is a type of communication that EnOcean devices use for pairing, it is usually used by devices with **EEP D2-xx-xx**.
This teach-in procedure is applicable for EnOcean devices with bidirectional communication profiles when the gateway is supposed to receive data from the device and also transmit data to the assigned device.
Let's have a roller shutter (D2-05-00) we want to assign to the channel 0. The procedure is as follows (see Fig. 3.2):

1. In the "Tx Channels" tab, select the channel 0
2. Optionally, set the "Sender ID" and confirm with "Save" (see note 1 and chapter 6.1)
3. Set "ResponseOption > UteResponseOnce"

Channel 0 is in the teach-in mode now.

4. Put the controller to the pairing mode (see note 2)
5. Gateway receives the telegram and sends a teach-in response (see note 3)
6. If teach-in successful, the gateway is now paired with the controller. It means the "Destination ID" is automatically filled by the unique ID of the assigned controller and the "Sender ID" (ID of the Tx channel) is stored in the memory of the controller.
7. Now the controller can receive commands sent by the gateway. To transmit a telegram, enter data to be sent (V1…V7) and use "Send" button.

To receive the actual status, assign the roller shutter to an Rx channel as D2-05-00.

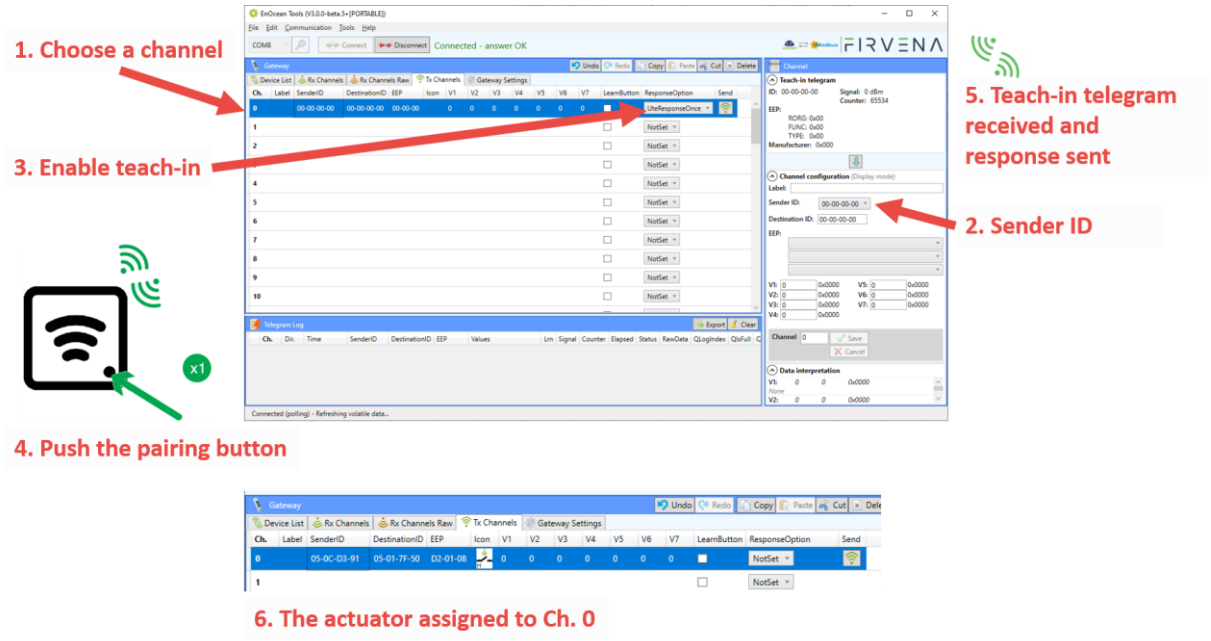


Fig. 3.2 Adding elements – bidirectional teach-in procedure

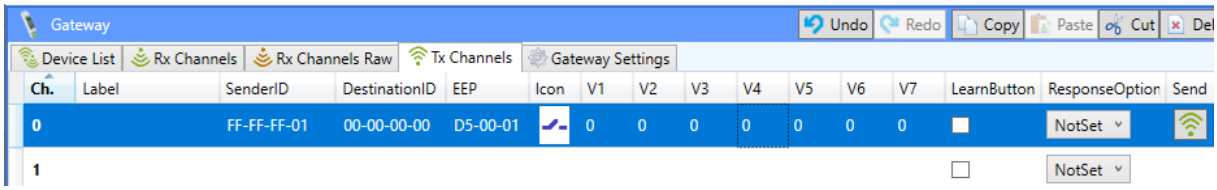
Notes:

1) *This ID will be sent in the teach-in response to the controller. The "Sender ID" uniquely identifies the Tx channel or gateway. Leave it 00-00-00-00 to use unique ID (EURID) of the EnOcean transceiver that is in the gateway. Or select one of the possible Base ID's.*
2) *As well as for the unidirectional teach-in procedure, follow the procedure given by the manufacturer of the controller. For example, the NodOn roller shutter is put to the teach-in mode by 3 consecutive presses of the "PRESS" button.*
3) *The device being assigned usually signals successful teach-in, e.g. by flashing LED. If it signals an error, the teach-in must be repeated.*
4) *It is possible to configure multiple channels for the same controller, which allows to predefine different commands. Once paired, use Copy&Paste to duplicate channels.*

3.4 Simulating an Element

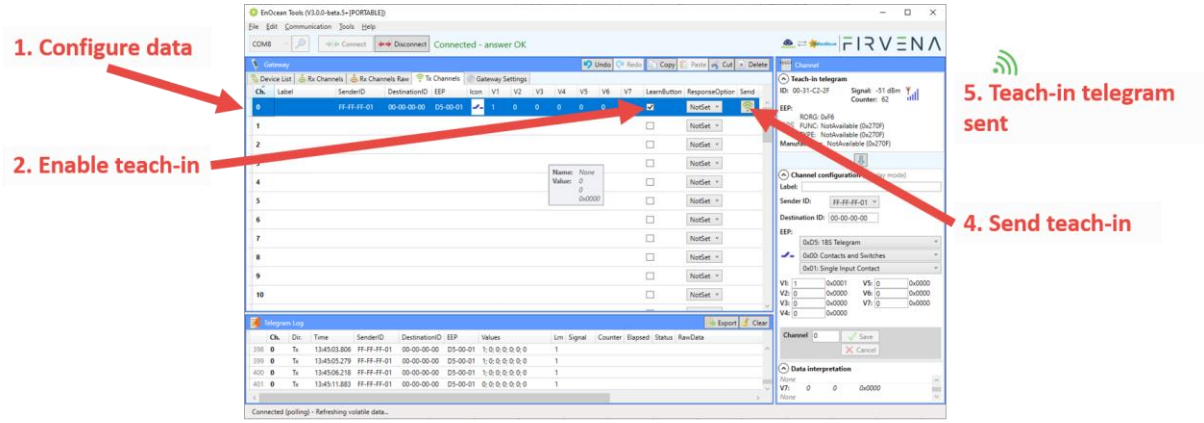
The gateway can be used to simulate a real EnOcean device. This function is necessary when controlling actuators that do not implement any bidirectional profile, typically some types of relay switches.
For example, we want to control a relay switch that supports reception of a Door/Window Contact D5-00-01. The procedure is as follows.
First, define a virtual device of type D5-00-01, for example at channel 0:

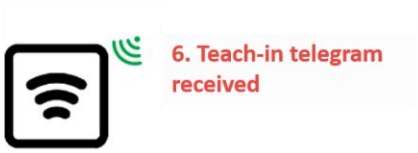
1. In the "Tx Channels" tab, select a channel (e.g. Ch. 0)
2. Set the "Sender ID" (e.g. FF-FF-FF-01, see note 1), leave "Destination ID" 00-00-00-00 for broadcast, 00-00-00-00 is the same as entering FF-FF-FF-FF.
3. Select the type of device (D5-00-01)
4. Click "Save" to confirm changes
5. The device is saved to the selected channel



Second, pair the virtual device with the controller:

1. Configure the data (V1…V7) (see note 2)
2. Tick the “LearnButton” to enable teach-in mode
3. Put the controller to the pairing mode (follow the procedure given by the manufacturer of the switch)
4. Click on the “Send” button to transmit a teach-in telegram
5. The gateway will transmit a teach-in telegram of the virtual device
6. The controller will receive the teach-in telegram and save the virtual device

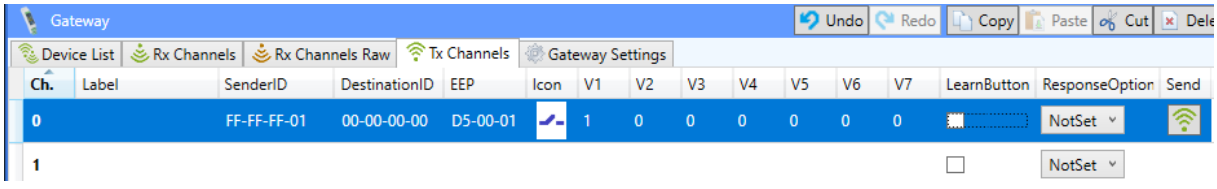




Third, test the connection:

1. Untick the “LearnButton”
2. Set the data to be sent V1…V7 (V1 = 0:open or 1:closed)
3. Use “Send” to transmit a data telegram

To receive the actual state of the relay switch, assign it to an Rx channel.



Notes:

1) *Each virtual device must have a unique ID, this is given by the “Sender ID” setting. Therefore, the “Sender ID” must be a Base ID from the range of FF-FF-FF-01…7F, this enables existence of multiple different virtual devices.*
2) *This point is necessary only for D5-00-01 and F6-xx-xx devices, other devices do not have data in their teach-in telegram and V1…V7 are not used during the teach-in mode. In the case of D5-00-01, the state transmitted in the teach-in telegram may be*

interpreted by actuator (controller) as the turn on state (it can turn on either when contact open or when closed). The F6-xx-xx devices do not have special teach-in telegram at all and only data telegrams are sent, just as D5-00-01, the data sent during pairing mode may be interpreted by actuator, e.g. to match the turn on button. Behavior for paired devices is sometimes configurable using special software provided by the manufacturer of the actuator. Such as software allows to change the settings of actuator via EnOcean Remote Commisioning or NFC.

3.4.1 F6-02-01 short press simulation:

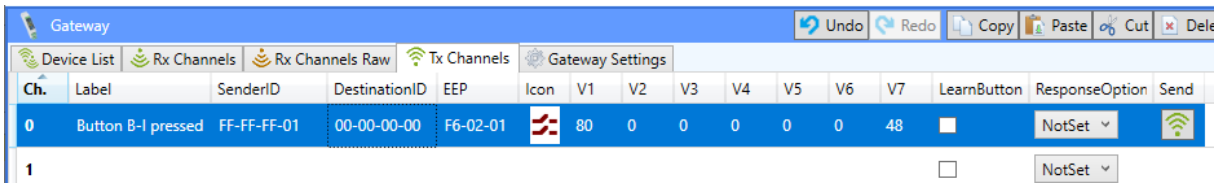
Simulation of RPS communication is a bit more complicated, because sometimes we need to simulate a short button press.
Meaning of Values:

- V1: Data byte 0 (DB0)
- V7: Status byte
- V2…V6: Not used

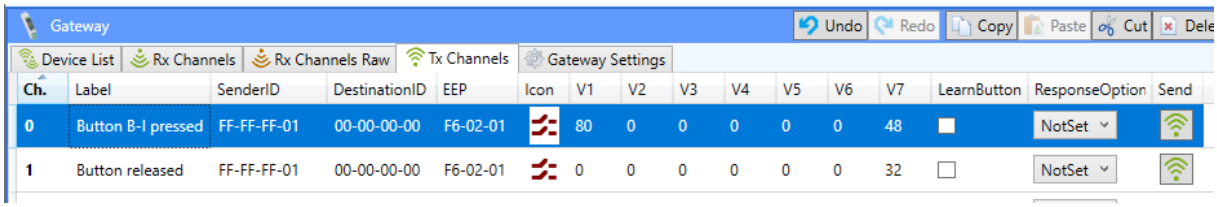
We need two bounded channels Channel 0 – simulates telegram sent on button pressed, Channel 1 – simulates telegram sent on button released.

1. Configure the “Button B-I pressed” telegram on the channel 0:

   V1 = 0x50: Button B-I pressed
   V7(Status) = 0x30



2. Configure the “Button released” telegram on the channel 1:
   V1 = 0x00: No action
   V7(Status) = 0x20



3. Set “ResponseOption > SendThisNowAndNextAfter200ms” to simulate a 200 ms long short press.

3.5 Removing EnOcean elements

EnOcean elements are removed by deleting the associated channels.

1. Select a channel
2. Click the “Delete” button or press the Delete key

To remove all elements, select all channels (Ctrl+A) and use Delete.
If more than one channel belongs to the device being removed, use the “Device List” tab to select channels.

4 Firmware Update

To update the firmware:

1. Connect the gateway through USB, RS485 or RS232 port to the computer; perform communication settings if necessary and from the menu bar select *“Tools/Firmware Loader”*.
2. In the window shown, select the firmware file (Fig. 4.1 – 3).
3. By the *“Program Memory”* button start downloading of the firmware (Fig. 4.1 – 5).
4. After downloading finished, the device is forced to reboot by the application and the communication is lost. Then, within a few seconds, reprogramming should perform. Wait until the communication is renewed and the result code is returned.
5. Check firmware version (Fig. 4.1 – 2).

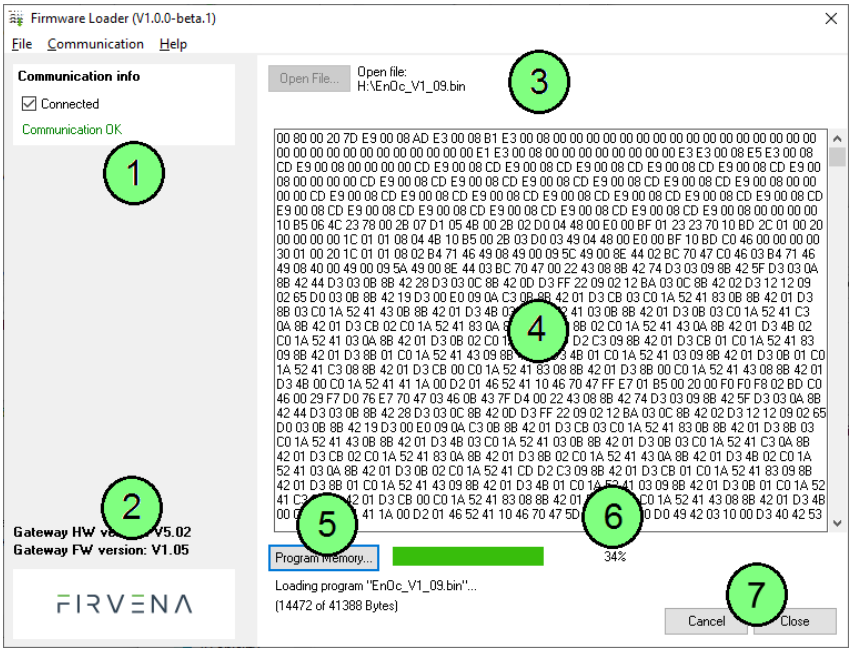


Fig. 4.1 Firmware Loader – main window overview

5 Backup of Gateway Configuration

The configuration of the gateway can be exported to a file for later recovery or other purposes.
To back up the configuration:

1. Connect the gateway and let the configuration to be loaded.
2. Use the Save command (*Ctrl+S* or *File/Save*).
3. Define the file path and name in the dialog shown.
4. Confirm *"Save"*.

To restore the configuration:

1. Connect the gateway and check if the communication works properly.
2. Use the Open command (*Ctrl+O* or *File/Open*).
3. Select a file and confirm *"Open"* in the dialog shown.
4. A prompt dialog is shown, select *"Yes"* to confirm the gateway configuration is to be overwritten.

Note: *Select "No" if you only wish to open the file, it will be shown in a new window.*

6 Appendix

6.1 Chip ID vs. Base ID

There is a setting called Sender ID concerning transmitting channels. The Sender ID identifies the device that transmits a telegram, in the case of Tx channels, the transmitting device is the gateway itself. Only the Chip ID or one of 128 possible Base IDs are allowed as Sender ID. This chapter should clarify what is the difference between the Chip ID and Base ID and which one to use.

The Chip ID is a unique identifier and is different for each gateway. It is an ID of the module implementing the EnOcean communication interface. Based on that fact, the Chip ID is referred to as Gateway ID in the application. The Base ID is an ID from the certain range of IDs allowed to be a Base ID and it is not unique.

The advantage of the Chip ID is the uniqueness, one device cannot be mistaken for another. But this makes the replacement of one gateway for another difficult because the teach-in procedure must be repeated with every device that stores the Sender ID of the current gateway, such devices may be switches, dimmers, valve actuators and so on. It is especially when these devices are hard to reach or there is many of them. Above stated problem can be solved using a Base ID when configuring Tx channels. For the EnOcean / Modbus RTU gateway, it can be an ID from FF-FF-FF-01 to FF-FF-FF-80.

Another application of the Base ID may be when we want the gateway to simulate one or more EnOcean elements (e.g. temperature sensor). For the preceding problem, we would probably use one Base ID common to all channels; however, in this case, the gateway must be able to transmit under different Sender IDs. It follows that a different Base ID must be used for each virtual transmitter.

Revision History

Tab. 14 Document revision history

Date	Version	Modifications made
2023-03-14	2.0	Initial release
2023-08-02	2.1	Added Ch. 3.2 (Valve actuator pairing)
2025-01-08	2.2	Corrected Rx Channels Raw Tab