Your Global Automation Partner

TURCK

TN-UHF-...-WIN
UHF Read/Write Head

Instructions for Use

Contents

1 About these Instructions

These operating instructions describe the structure, functions and the use of the product and will help you to operate the product as intended. Read these instructions carefully before using the product. This is to avoid possible damage to persons, property or the device. Retain the instructions for future use during the service life of the product. If the product is passed on, pass on these instructions as well.

1.1 Target groups

These instructions are aimed at qualified personal and must be carefully read by anyone mounting, commissioning, operating, maintaining, dismantling or disposing of the device.

1.2 Explanation of symbols used

The following symbols are used in these instructions:

	**DANGER** DANGER indicates a dangerous situation with high risk of death or severe injury if not avoided.
	**WARNING** WARNING indicates a dangerous situation with medium risk of death or severe injury if not avoided.
	**CAUTION** CAUTION indicates a dangerous situation of medium risk which may result in minor or moderate injury if not avoided.
	**NOTICE** NOTICE indicates a situation which may lead to property damage if not avoided.
	**NOTE** NOTE indicates tips, recommendations and useful information on specific actions and facts. The notes simplify your work and help you to avoid additional work.
▶	**CALL TO ACTION** This symbol denotes actions that the user must carry out.
⇨	**RESULTS OF ACTION** This symbol denotes relevant results of actions.

1.3 Other documents

Besides this document the following material can be found on the Internet at www.turck.com:

- Data sheet
- Approvals
- Configuration manual

1.4 Naming convention

Common synonyms for "data carriers" include "tag", "transponder", and "mobile storage device". Read/write heads are also described as "transceivers" or "readers".

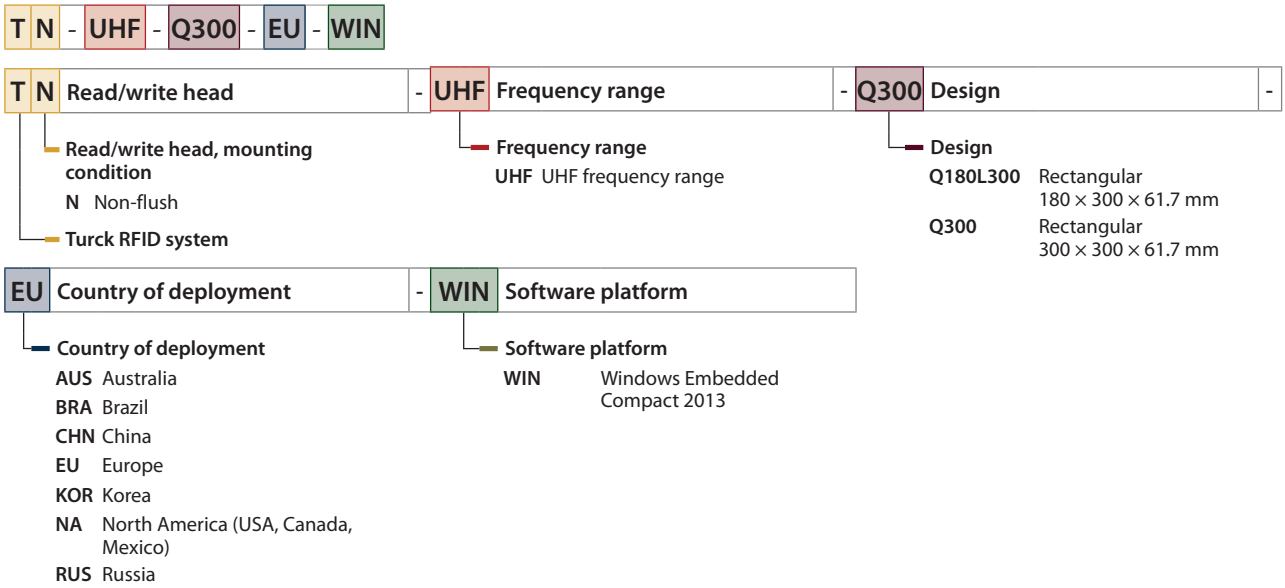
1.5 Feedback about these instructions

We make every effort to ensure that these instructions are as informative and as clear as possible. If you have any suggestions for improving the design or if some information is missing in the document, please send your suggestions to techdoc@turck.com.

2 Notes on the Product

2.1 Product identification

These instructions apply to the following UHF read/write heads:



2.2 Scope of delivery

The scope of delivery includes:

- Read/write head
- Wall bracket (metal rail)
- Quick Start Guide

2.3 Legal requirements

The devices are subject to the following EU directives:

- 2014/30/EU (electromagnetic compatibility)
- 2011/65/EU (RoHS Directive)
- 2014/53/EU (RED Directive)

2.4 Manufacturer and service

Hans Turck GmbH & Co. KG
Witzlebenstraße 7
45472 Mülheim an der Ruhr
Germany

Turck supports you with your projects, from initial analysis to the commissioning of your application. The Turck product database contains software tools for programming, configuration or commissioning, data sheets and CAD files in numerous export formats. You can access the product database at the following address: **www.turck.de/products**

For further inquiries in Germany contact the Sales and Service Team on:

- Sales: +49 208 4952-380
- Technology: +49 208 4952-390

Outside Germany, please contact your local Turck representative.

3 For Your Safety

The product is designed according to state-of-the-art technology. However, residual risks still exist. Observe the following warnings and safety notices to prevent damage to persons and property. Turck accepts no liability for damage caused by failure to observe these warning and safety notices.

3.1 Intended use

These devices are designed solely for use in industrial areas.

The read/write heads with an integrated RFID interface are used for contactless data exchange with the BL ident tags in the Turck UHF RFID system. The following table shows the operating frequency of the devices:

Type code	Operating frequency	Region
TN-UHF-…-EU-WIN	865…868 MHz	Europe, Turkey, India
TN-UHF-…-NA-WIN	902…928 MHz	North America (USA, Canada, Mexico)
TN-UHF-…-CHN-WIN	920.5…924.5 MHz	China
TN-UHF-…-KOR-WIN	917…920.8 MHz	Korea
TN-UHF-…-BRA-WIN	915…928 MHz	Brazil
TN-UHF-…-RUS-WIN	866…868 MHz	Russia
TN-UHF-…-AUS-WIN	920…926 MHz	Australia

These devices may only be started up under the following conditions:

- The particular frequency range is permissible for the use of UHF-RFID.
- The operating frequency range of the devices is compliant with the permissible UHF RFID range of the region.
- A valid certification and/or approval is available for the region of use.

The integrated RFID interface enables the read/write heads to communicate directly via TCP/IP with higher-level systems such as ERP systems. Read data is sent to the higher-level system via the device.

Four configurable digital channels are also provided for connecting digital sensors and actuators.

The devices may only be used as described in these instructions. Any other use is not in accordance with the intended use. Turck accepts no liability for any resulting damage.

3.2 General safety notes

- The device only meets the EMC requirements for industrial areas and is not suitable for use in residential areas.
- The device may only be assembled, installed, operated, parameterized and maintained by professionally-trained personnel.
- The device may only be used in accordance with applicable national and international regulations, standards and laws.
- Any extended stay within the area of radiation of the UHF read/write heads may be harmful to health. Observe minimum distances from the actively radiating surface of the UHF read/write head:

Region	Max. permissible total radiant output power	Safety distance
Europe, Russia, China	2 W ERP (according to ETSI)	> 0.24 m
USA, Canada, Brazil, Korea, Australia, New Zealand	4 W EIRP	> 0.30 m
Singapore	0.5 W ERP	> 0.24 m

- The radiation of the UHF read/write heads may have an adverse effect on the operation of electrically controlled medical equipment. Keep an additional distance from active radiation sources up to the maximum transmission distance.

4 Product Description

The devices are designed with an aluminum housing and degree of protection IP67. The active face is made out of plastic. Devices are available with an integrated antenna (Q300) or for connecting external antennas (Q180). Both device variants are suitable for connecting up to four external passive UHF RFID antennas.

The terminals for the Ethernet and for digital I/Os are M12 sockets. The device has an M12 plug connector for connecting the power supply. Terminals are provided for up to four external antennas.

4.1 Device overview

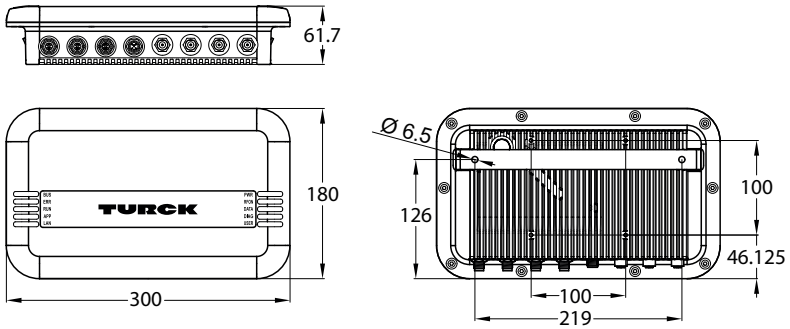


Fig. 1: Dimensions – TN-UHF-Q180L300…

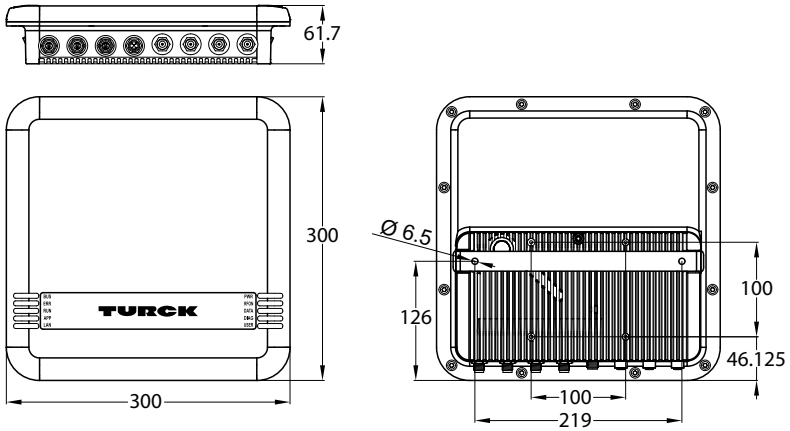


Fig. 2: Dimensions – TN-UHF-Q300…

4.1.1 Indication elements

The device has the following LED indicators:

- Power supply
- Group and bus errors
- Status
- Diagnostics

An audible alarm can also be set using software tools.

4.2 Properties and features

- TCP/IP
- Freely programmable compact module based on Windows Embedded Compact 2013
- Programming languages .Net, C++, C#
- API and SDK available on request
- Implementation of the protocol required
- 4 RP-TNC terminals for passive external UHF RFID antennas
- 4 configurable digital channels as 0.5 A PNP inputs and/or outputs
- 10 Mbps/100 Mbps transfer rate
- Integrated web server
- LED indications and diagnostics

4.3 Operating principle

The read/write heads are used for contactless data exchange with tags. For this the controller sends commands and data via the interface to the read/write head and receives the corresponding response data from the read/write head. The reading of the IDs of all RFID tags in the read area or the writing of an RFID tag with a specific production date are examples of typical commands. To communicate with the tag, the data of the read/write head is coded and transferred via an electromagnetic field, which at the same time supplies the tags with power.

A read/write head contains a transmitter and a receiver, an interface to the interface and a coupling element (coil and dipole antenna) for communicating with the tag. Electromagnetic wave propagation is used for the transmission between read/write head and tag on devices for the UHF range.

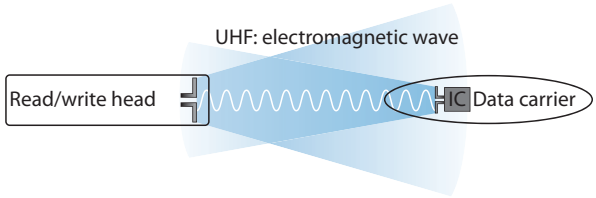


Fig. 3: Operating principle of UHF-RFID

The antenna of the read/write head generates electromagnetic waves. This produces a transmission window as a so-called air interface in which the data exchange with the tag takes place. The size of the transmission window depends on the combination of read/write heads and tags, as well as on the relevant environmental conditions.

Each read/write head can communicate with a number of tags. This requires the read/write head and the tag to operate in the same frequency range. Depending on the power and frequency used, the device ranges vary from a few millimeters up to several meters. The specified maximum read/write distances only represent typical values under laboratory conditions without allowing for the effect of materials. The achievable distances may vary due to component tolerances, the mounting situation in the application, ambient conditions and the effect of materials (particularly metal and liquids).

4.4 Functions and operating modes

The devices operate with an integrated or external antenna (TN-UHF-Q300…) or only with an external antenna (TN-UHF-Q180L300…). The devices enable passive UHF tags to be read or written in single and multitag operation. For this the devices form a transmission zone that varies in size and range according to the tags used and the operating conditions of the application. Refer to the data sheets for the applicable maximum read/write distances. The devices can be fully tested, configured and parameterized from a PC using the specified software tools.

The Windows Embedded Compact 2013 operating system enables the device functions to be programmed with .Net, C++ or C#. It is also possible to integrate middleware functions on the device.

Sensors and actuators can be connected to the configurable digital channels. Up to four 3-wire PNP sensors or two PNP DC actuators with a maximum output current of 2 A per output can be connected. An external power supply is required in order to use the digital channels as outputs.

4.5 Technical accessories

Accessories for mounting, connecting and parameterizing can be found in product database or the Accessories List for TBEN (D301367) under **www.turck.com**. The accessories are not part of the scope of delivery.

5 Installing

The device is provided with a bracket in accordance with VESA 100 × 100 for mounting. The device is provided with four M4 threaded holes spaced 100 mm apart (horizontally and vertically). The maximum length of the screws is 8 mm plus the thickness of the VESA bracket. The devices can be mounted in any position.

- ▶ Fasten the device with four M4 screws to a bracket in accordance with VESA 100 × 100.

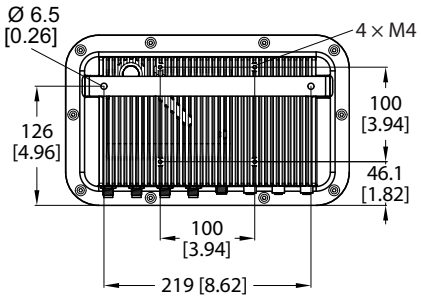


Fig. 4: Rear view – TN-UHF-Q180…

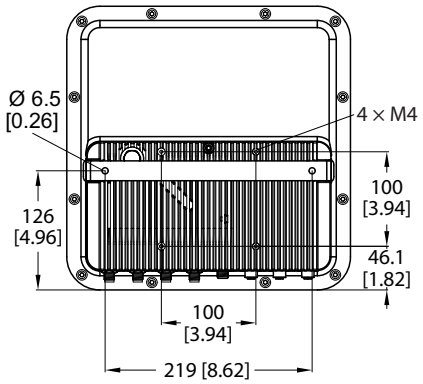


Fig. 5: Rear view – TN-UHF-Q300…

6 Connecting

6.1 Connecting devices to Ethernet

The device is provided with a 4-pin M12 female connector for connecting the device to an Ethernet system.

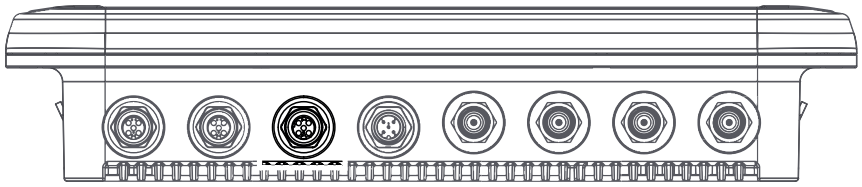


Fig. 6: M12 Ethernet connector

- ▶ Connect the device to Ethernet as per the following pin assignment (max. tightening torque 0.8 Nm).

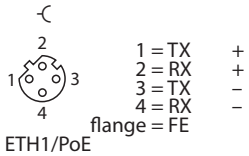


Fig. 7: Pin assignment of the Ethernet connections

NOTE
With PoE transfer the power supply via PoE Mode A with 4-wire cables.

6.2 Connecting the power supply

The device is provided with a 5-pin M12 plug connectors for connecting the power supply.

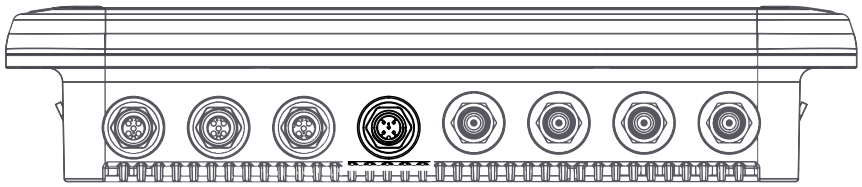


Fig. 8: M12 plug connector for connecting the power supply

- ▶ Connect the device to the power supply as per the following pin assignment (max. tightening torque 0.8 Nm).

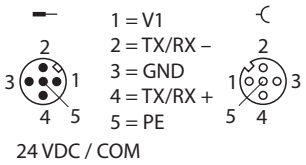


Fig. 9: Pin assignment of the power supply terminals

6.3 Connecting digital sensors and actuators

The device has two 5-pin M12 plug connectors for connecting digital sensors and actuators.

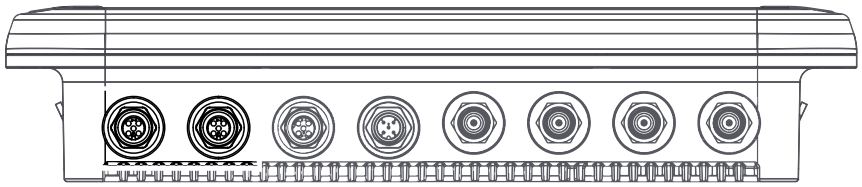


Fig. 10: M12 plug connectors for connecting digital sensors and actuators

NOTE
When operating via PoE (Power over Ethernet) the digital channels cannot be used as outputs.

- Connect sensors and actuators to the device as per the following pin assignment (max. tightening torque 0.8 Nm).

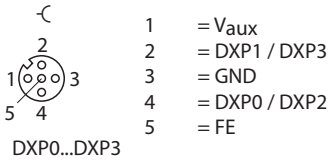


Fig. 11: Connections for digital sensors and actuators – pin assignment

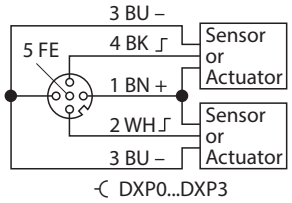


Fig. 12: Connections for digital sensors and actuators – wiring diagram

Hans Turck GmbH & Co. KG | T +49 208 4952-0 | F +49 208 4952-264 | more@turck.com | www.turck.com

6.4 Connecting external antennas

The device is provided with four RP-TNC sockets for connecting up to four external antennas. The input impedance is 50 Ω.

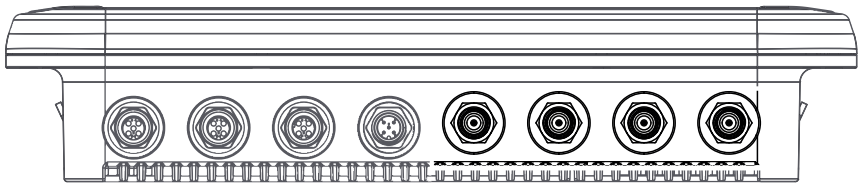


Fig. 13: RP-TNC sockets for connecting external antennas

- ▶ Connect external antennas with an RP-TNC antenna cable to the device (max. tightening torque 0.8 Nm).

7 Commissioning

7.1 Parameterizing read/write heads with the DTM

The UHF settings of the device can be assigned additional parameters via a DTM.

All the required Turck software components can be downloaded via the Turck Software Manager. The Turck Software Manager is available free of charge from www.turck.com.

NOTE
The parameterization function up to firmware version V2.0.39.3937 is only available in English. All parameters are written in the DTM.

The individual read/write heads are available in different variants. When a connection is made to a connected read/write head, the DTM automatically detects the relevant device and deactivates menu items that are not supported. The connection cannot be established if a different variant than set in the project tree is connected.

NOTE
Adjustable parameters are indicated in the DTM with a green arrow. Fixed parameters are indicated by gray arrows.

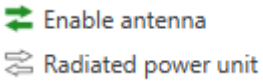


Fig. 14: DTM – example of adjustable and fixed parameters

Requirements for extended parameter setting

- PACTware must be installed.
- The DTM for UHF read/write heads must be installed.

7.1.1 Connecting the device with the PC

- Open PACTware.
- Right-click **Host PC** in the project tree.
- Click **Add device**.
- Add **RS485 RFID**.
- Confirm selection with **OK**.

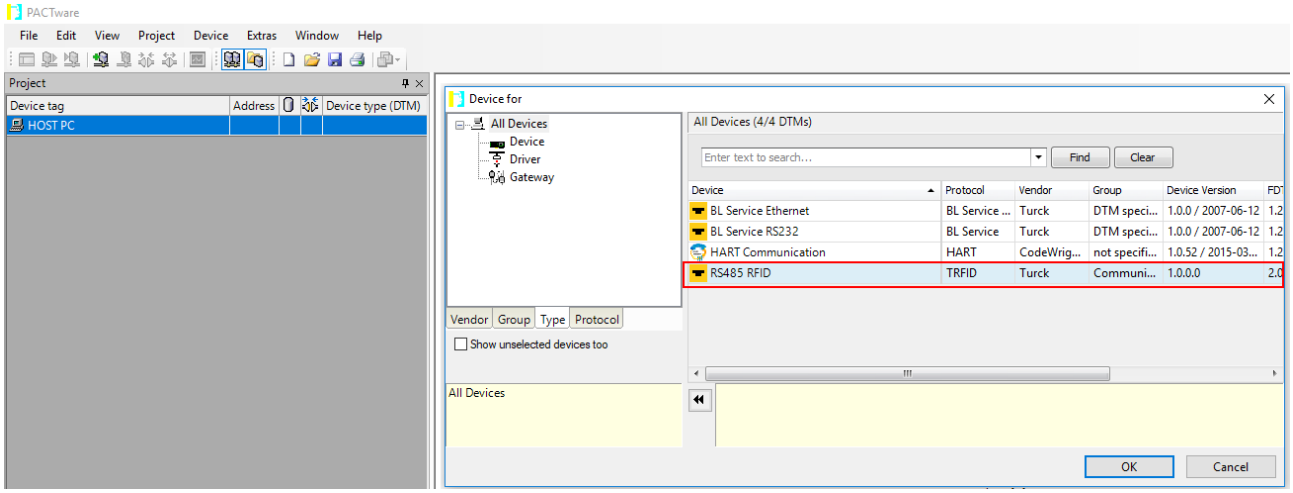


Fig. 15: Selecting RS485 RFID

- Right-click the Ethernet adapter.
- Start the **Topology scan**.

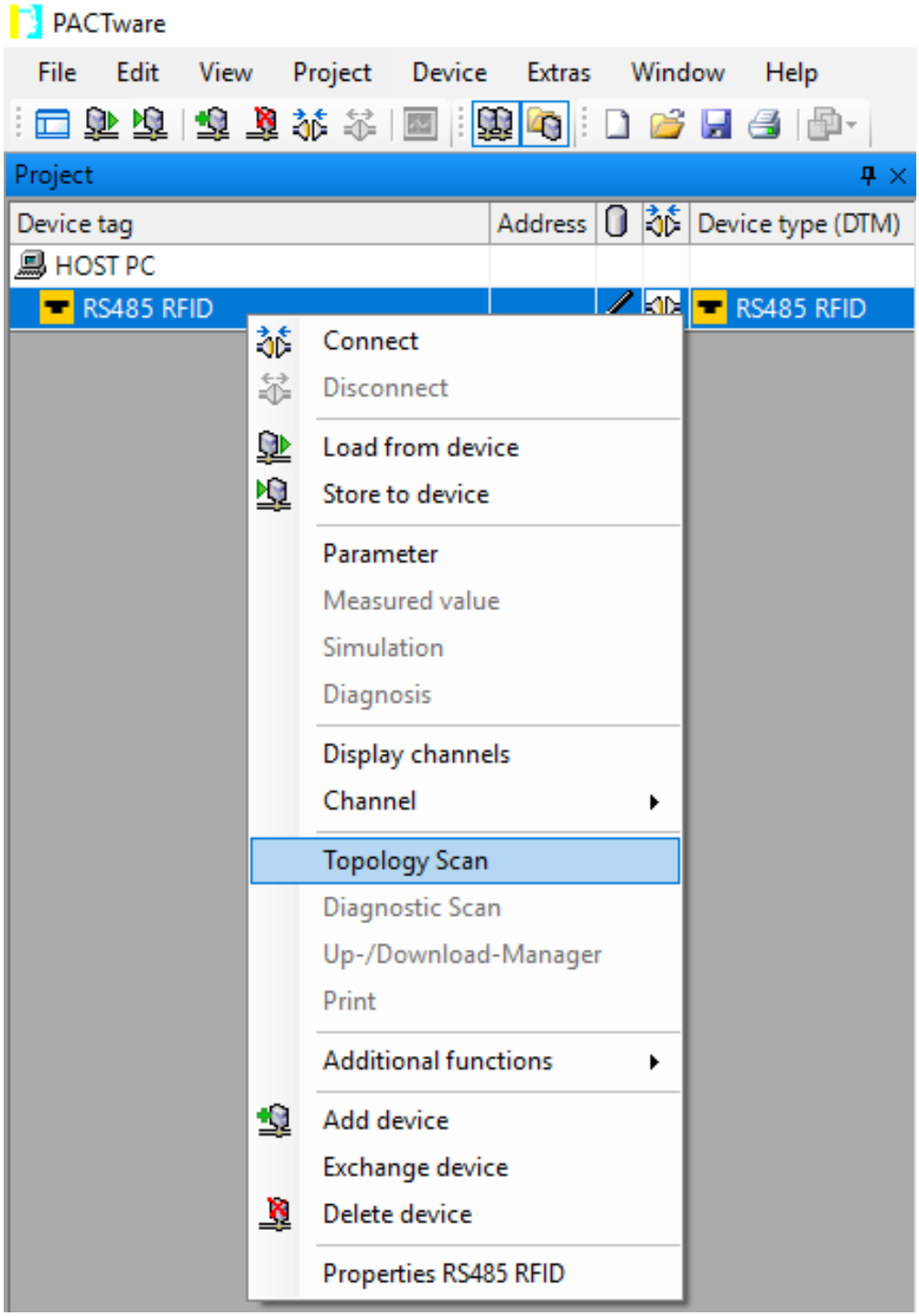


Fig. 16: Starting the Topology scan

The connected devices are automatically detected and added to the project tree.

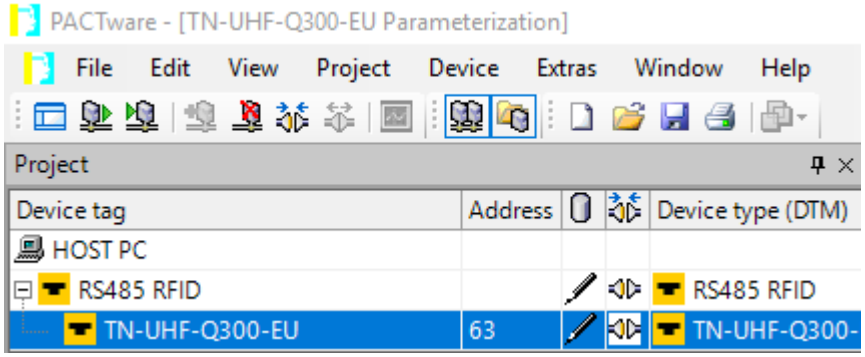


Fig. 17: Project tree

7.1.2 Starting the extended read/write head parameter setting

- Right-click the device.
- Start the parameter setting: Choose **Parameterization** or **Online parameterization**. The device must be connected to the PC for the **Online parameterization**.

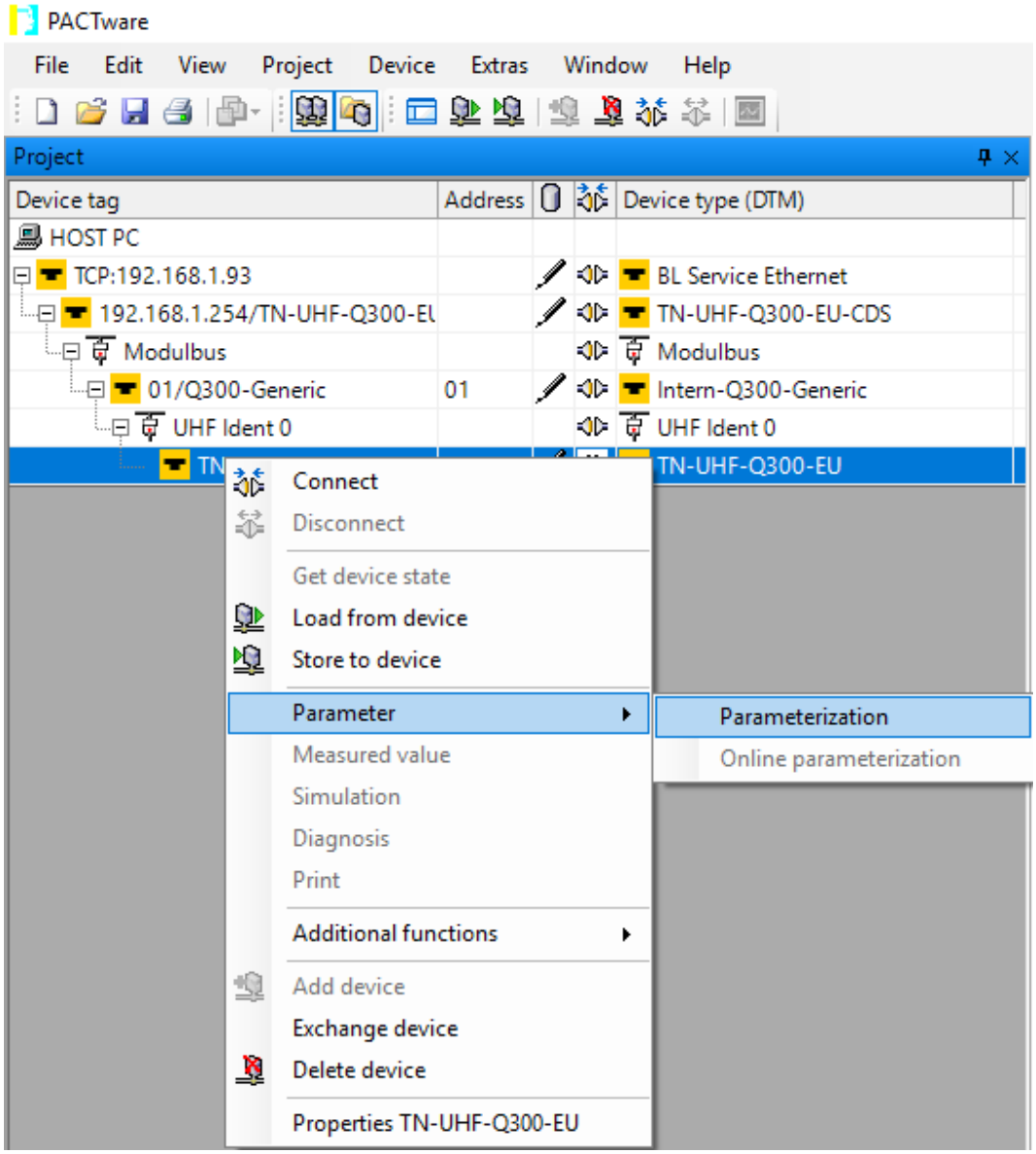


Fig. 18: Starting the parameterization

7.1.3 DTM main menu – overview

Fig. 19: DTM – main menu

The main menu provides the following functions:

Icon	Function	Description
	Show and hide information bar	Shows the information bar for the connected device and DTM version at the top of the screen.
	DTM help	Starts the DTM help.
	Device help	Opens the data sheet of the connected read/write head.
	Expert mode ON/OFF	Opens the drop-down menu to select the access level. The following access levels are available: ■ Basic (default setting) ■ Advanced ■ Administrator (password-protected)
	Display channel wise	Toggles the view between standard display and channel-wise display.
	Load data from database	Loads previously stored parameters from the database (e.g. an existing project).
	Store data in database	Transfers the current read/write head parameters to the database of the current project.
	Read data from device	Reads the set parameters from the device.
	Transmit data to device	Transfers the set parameters to the device.
	Compare displayed values with database	Compares the values displayed in the DTM with the values saved in the database.
	CSV export current values	Exports the current values from the DTM to a CSV file.

The following setup windows can be opened in tabs via the main menu:

- Basic setup
- Antenna
- Antenna configuration
- Communication
- EPC Class1 Gen2
- Post read filter
- Signaling

7.1.4 Choosing the access level

Three access levels are available for setting the device parameters. Different parameters can be set depending on the access level.

NOTE
Modifications made in the **Administrator** access level can result in serious changes to operation. The **Administrator** access level is therefore only available for Turck service technicians. All relevant settings for the successful parameter setting of an application are available in the **Advanced** access level.

Access level	Description	Initial password
Basic	Basic access for configuration and commissioning	Not required
Advanced	Extended access, e.g. for applications	Not required
Administrator	Administrator access for critical security or wireless settings	Required

The current access level is displayed in the top right screen area of the DTM.

Fig. 20: Display of the access level

7.1.5 Setting multiplex operation

In multiplex operation, several antennas can be controlled or switched on in sequence. The example below shows the activation of the antennas in sequence. The multiplex operation can consist of up to 16 sequences and can be used, for example, for gate applications.

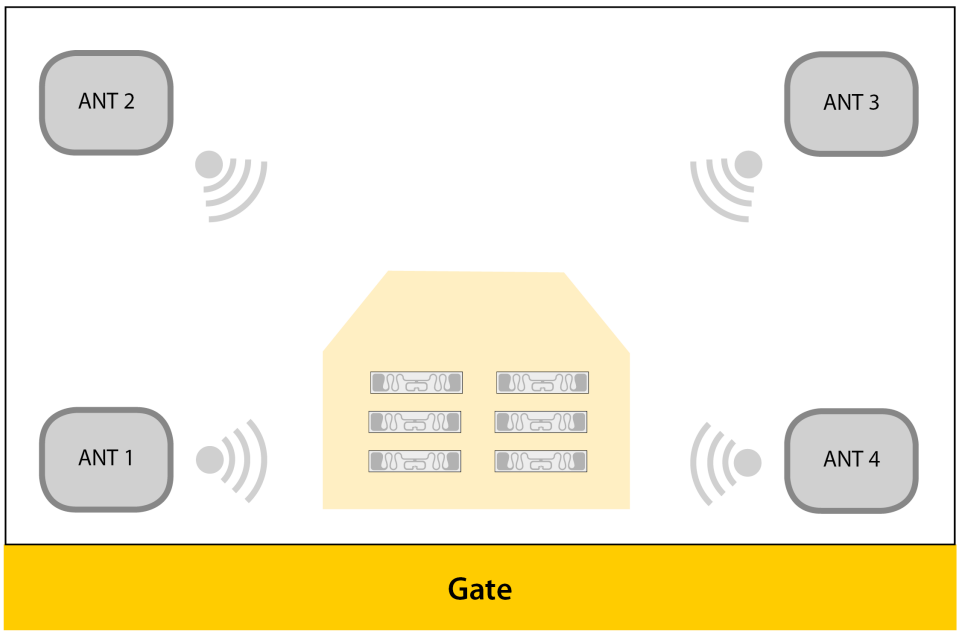


Fig. 21: Gate application – schematic representation

Configuring multiplex operation – example

- Choose the **Antenna** tab in the main menu.
- At **Antenna** → **Antenna multiplexing** → **Number of entries** enter the number of antennas.

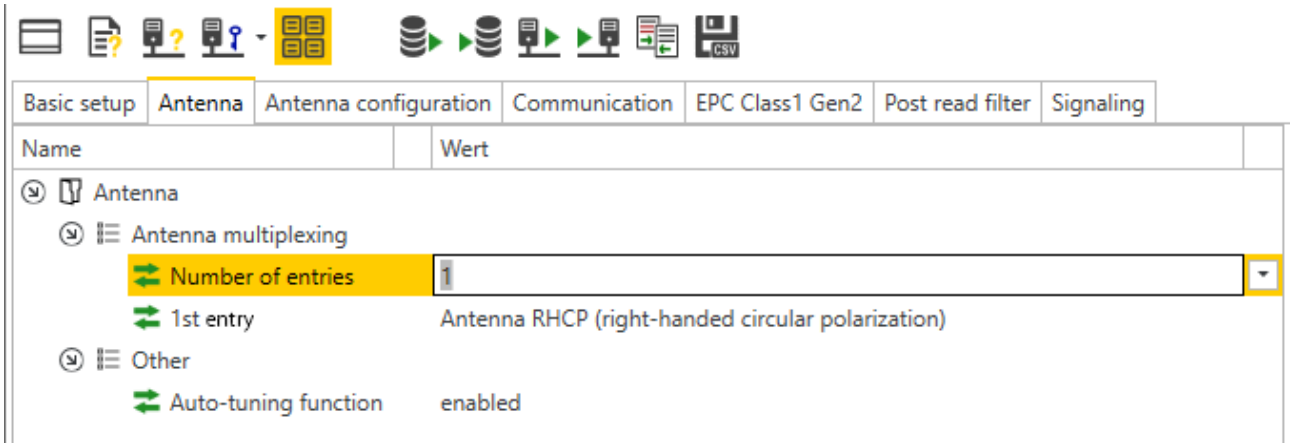


Fig. 22: Entering the number of antennas

- Assign antennas with functions (e.g. internal antenna: RHCP, LHCP, external antenna)

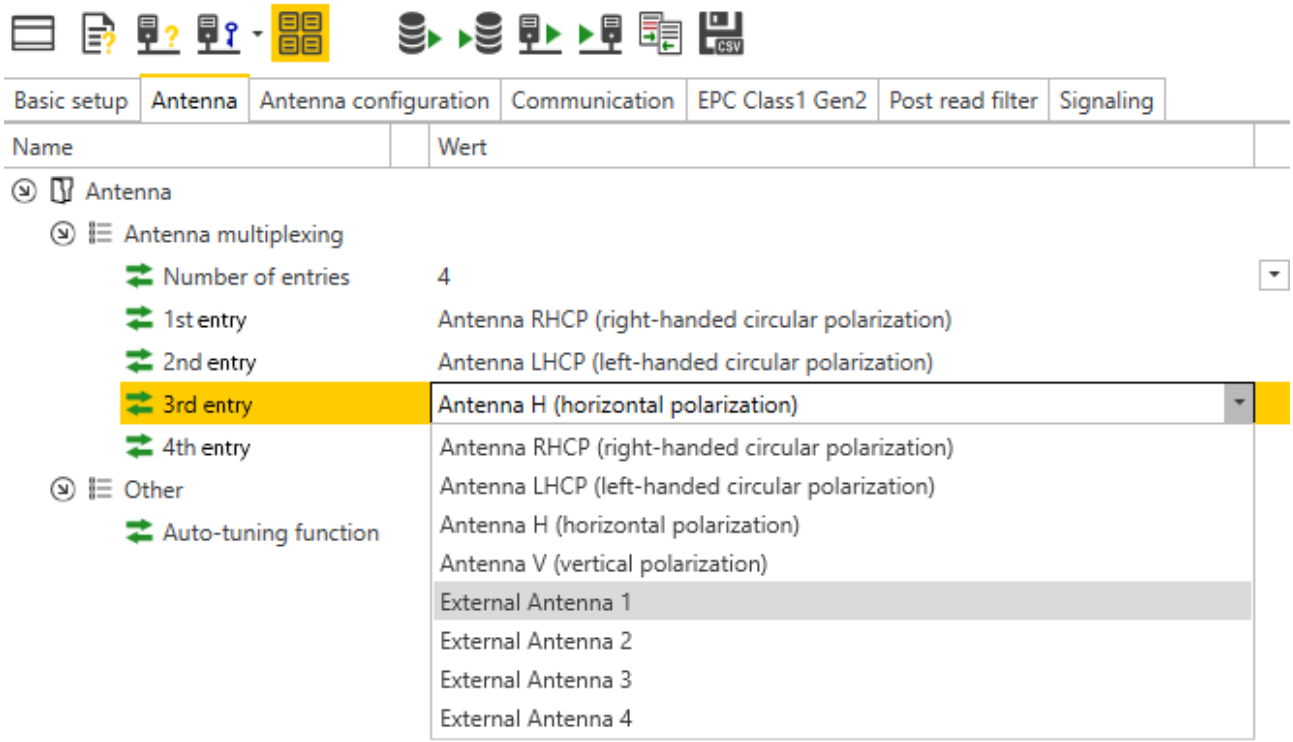


Fig. 23: Example: setting multiplex operation

- Click **Accept** to save the settings.

- For all antennas used set at **Antenna configuration** → **Maximal transmit time** the time in which the particular antenna is to remain active and stay switched on.

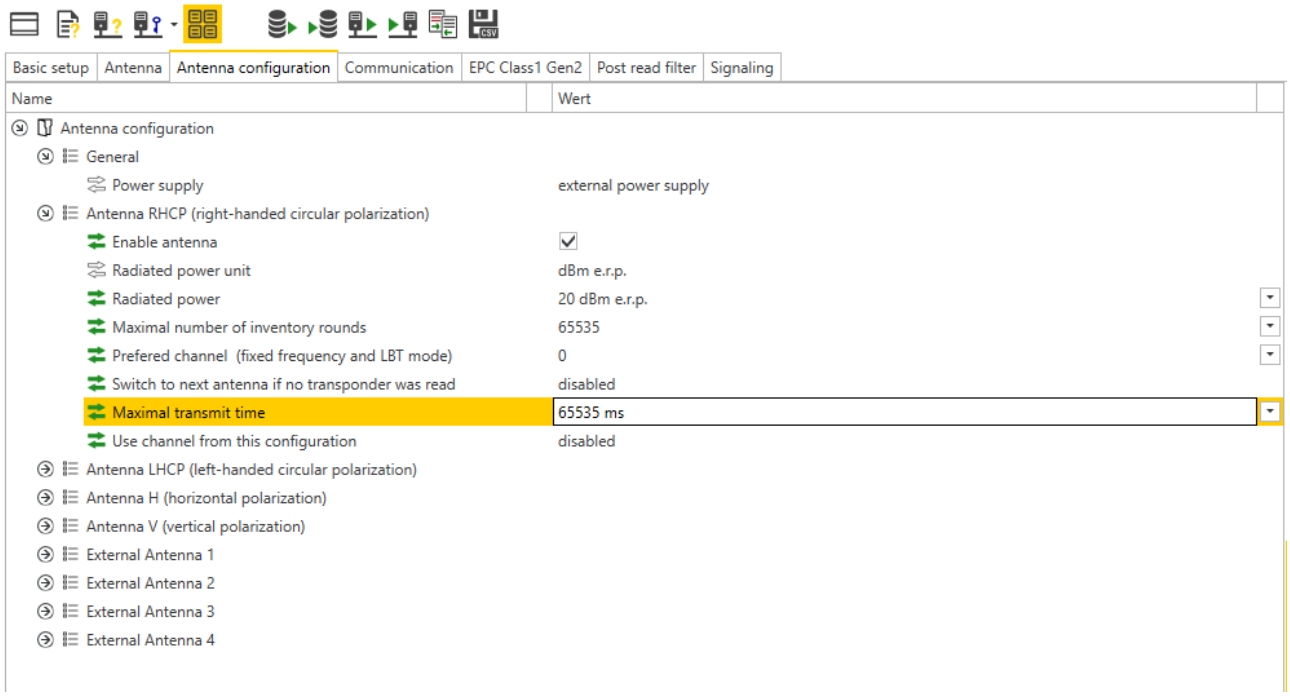


Fig. 24: Setting the maximum transmit time

7.1.6 Setting antenna power

The antenna power of the read/write head can be set for the specific application. The radiated power for the integrated antenna can be entered directly in the DTM. The power must be calculated for external antennas.

The following parameters must be used to calculate the radiated power (P_{ERP}):

P_{cond}	Power to be output at the TNC socket of the read/write head
dB	Cable attenuation
G_{HW}	Antenna gain of the external antenna

NOTE
Refer to the data sheets of the components used for the cable attenuation and antenna gain.

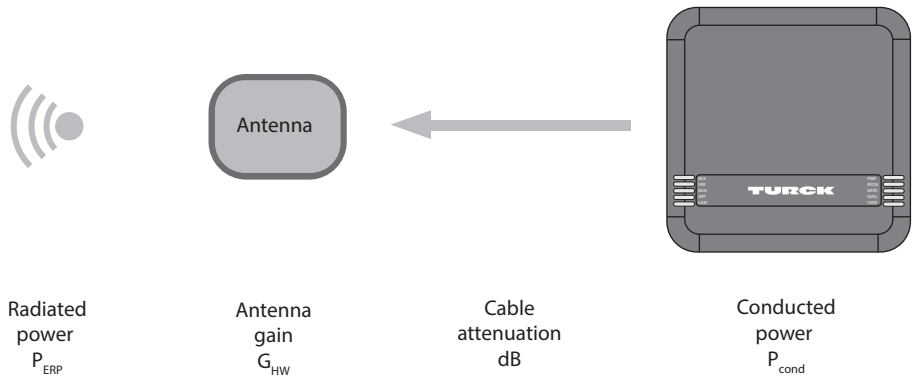


Fig. 25: Power calculation – relevant variables (schematic representation)

The power can be calculated with the following formula:

$P_{ERP} = G_{HW} - dB + P_{cond}$

Setting antenna power – Restrictions of radio regulations

Some national regulations restrict the degree of freedom available for creating an RFID system. You as the operator are responsible for ensuring that regulations are observed.

- ETSI
  - Radiated power P_{ERP}: Max. 33 dBm ERP
- FCC
  - Radiated power P_{ERP}: Max. 36 dBm EIRP
  - P_{cond}: Max. 30 dBm with antenna gain $G_{HW} \leq 6$ dbi

NOTE
The DTM indicates impermissible configurations with an exclamation mark. A transmission to the device is prevented.

Calculating radiated power

The effective radiated power (ERP) is the power that is radiated from an antenna into free space. To make it possible to compare the technical properties of different antenna, the power specifications given are always in relation to a reference antenna.

- EIRP = equivalent isotropic radiated power (reference: isotropic antenna)
- ERP = effective radiated power (reference: with the length of λ/2)

The radiated power can be stated in watts or in dBm. The following table shows approximate values as a guide for converting between dBm and mW:

dBm	mW	dBm	mW	dBm	mW	dBm	mW
1	1.25	9	8	17	50	25	316
2	1.6	10	10	18	63	26	400
3	2	11	13	19	80	27	500
4	2.5	12	16	20	100	28	630
5	3	13	20	21	125	29	800
6	4	14	25	22	160	30	1000
7	5	15	32	23	200	…	…
8	6	16	40	24	250	33	2000

The formula for calculating the exact values is: **dBm = 10 × lg (P/1 mW)**

Converting antenna gain

The antenna gain can be specified in the following units:

dBd	Antenna gain in relation to a dipole
dBi	Antenna gain in relation to an isotropic radiator (linear)
dBic	Antenna gain in relation to an isotropic radiator (circular)

The different units can be converted as follows:

- G_{HW} = dBd
- G_{HW} = dBi - 2.15
- G_{HW} = dBic - 5.15

Setting the power for external antennas via the DTM

When supplied via Power over Ethernet (PoE), the radiated power for the internal antenna is limited to 1 W. With external antennas 1 W of output power is provided at the TNC socket. The power supply type is set automatically via **Antenna configuration** → **Power supply** to the **external power supply** value.

- ▶ Set the radiated power via **Antenna Configuration** → **Radiated power** (here: 33 dBm e.r.p.).

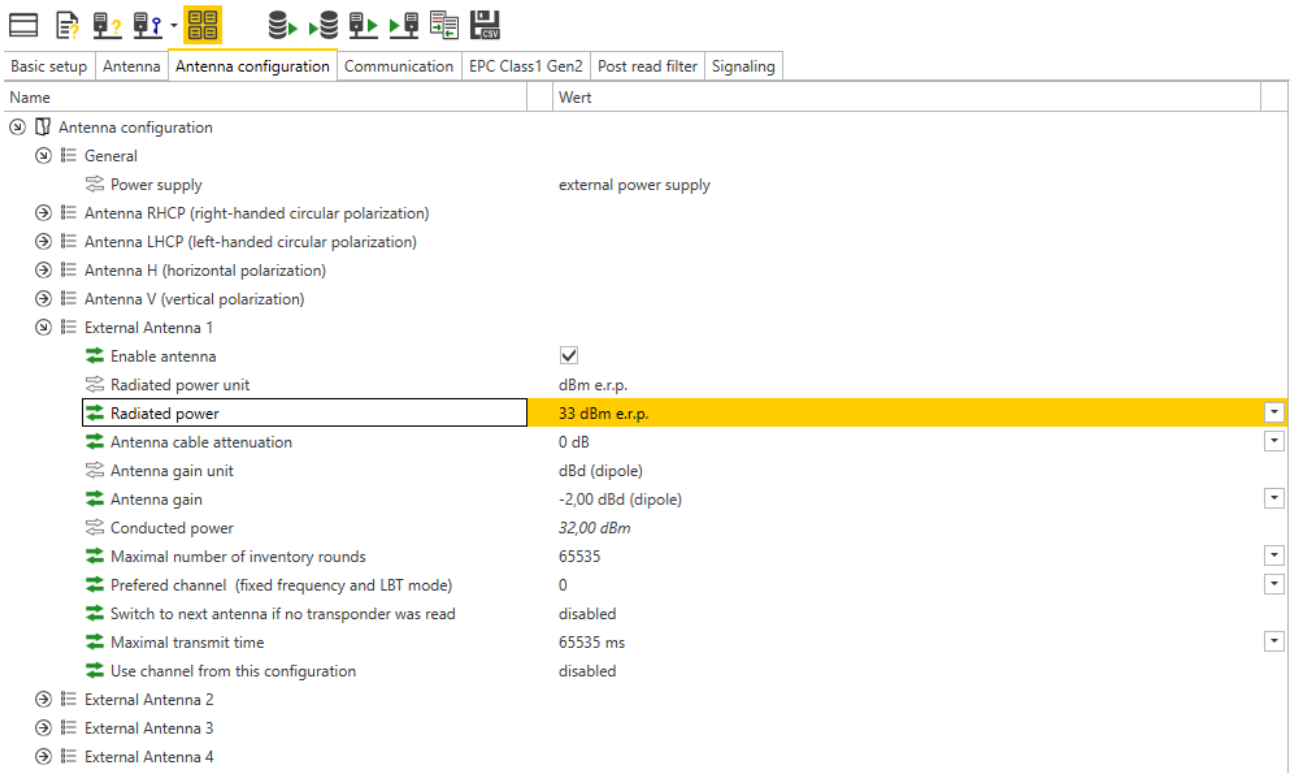


Fig. 26: Setting the radiated power

Hans Turck GmbH & Co. KG | T +49 208 4952-0 | F +49 208 4952-264 | more@turck.com | www.turck.com

- Refer to the data sheet of the cable used for the cable attenuation.
- Enter the cable attenuation at **Antenna cable attenuation**.

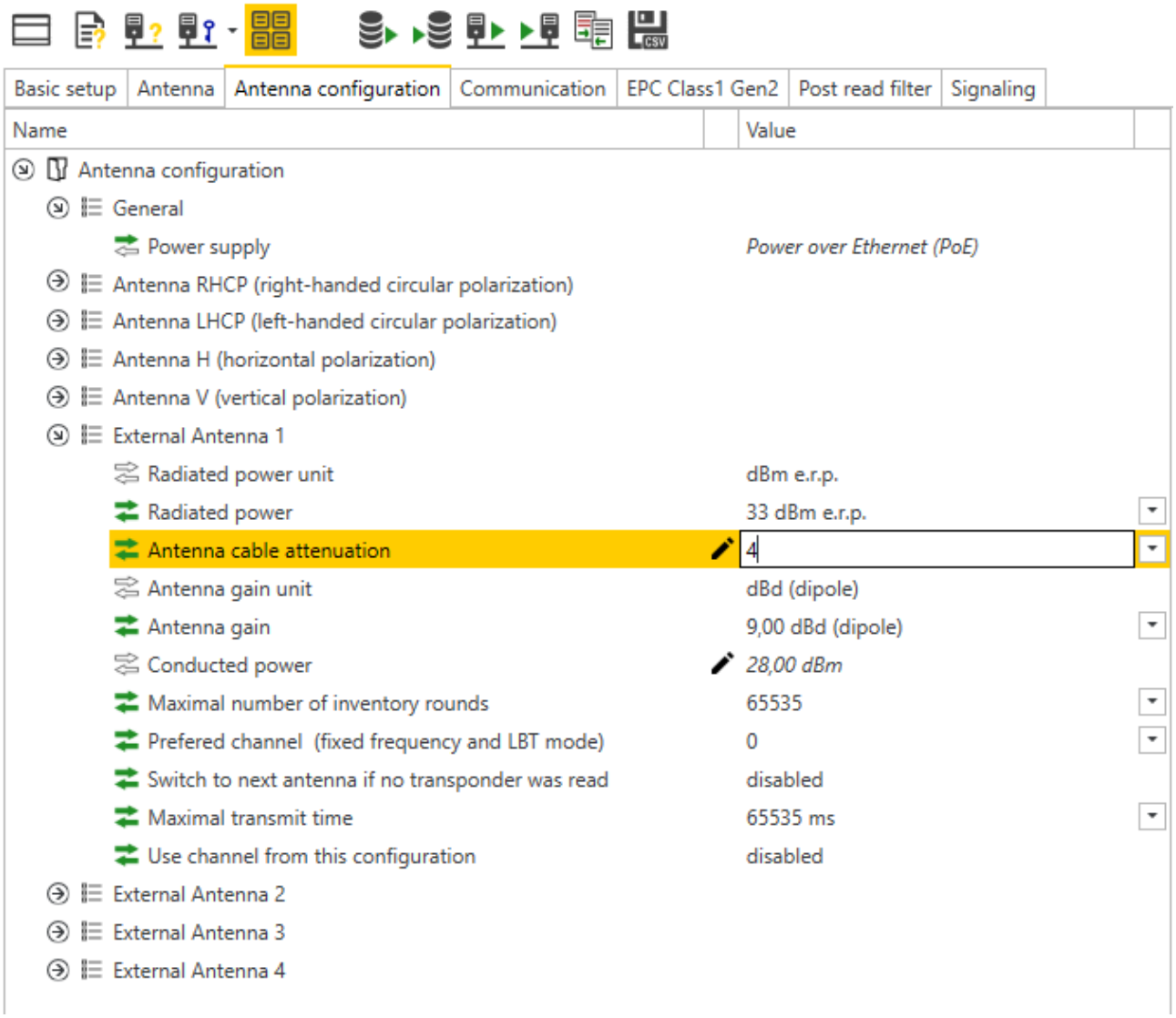


Fig. 27: DTM – entering the cable attenuation

- ▶ Refer to the data sheet of the external antenna for the antenna gain.
- ▶ Set the unit for the antenna gain at **Antenna gain unit** (here: dBd).

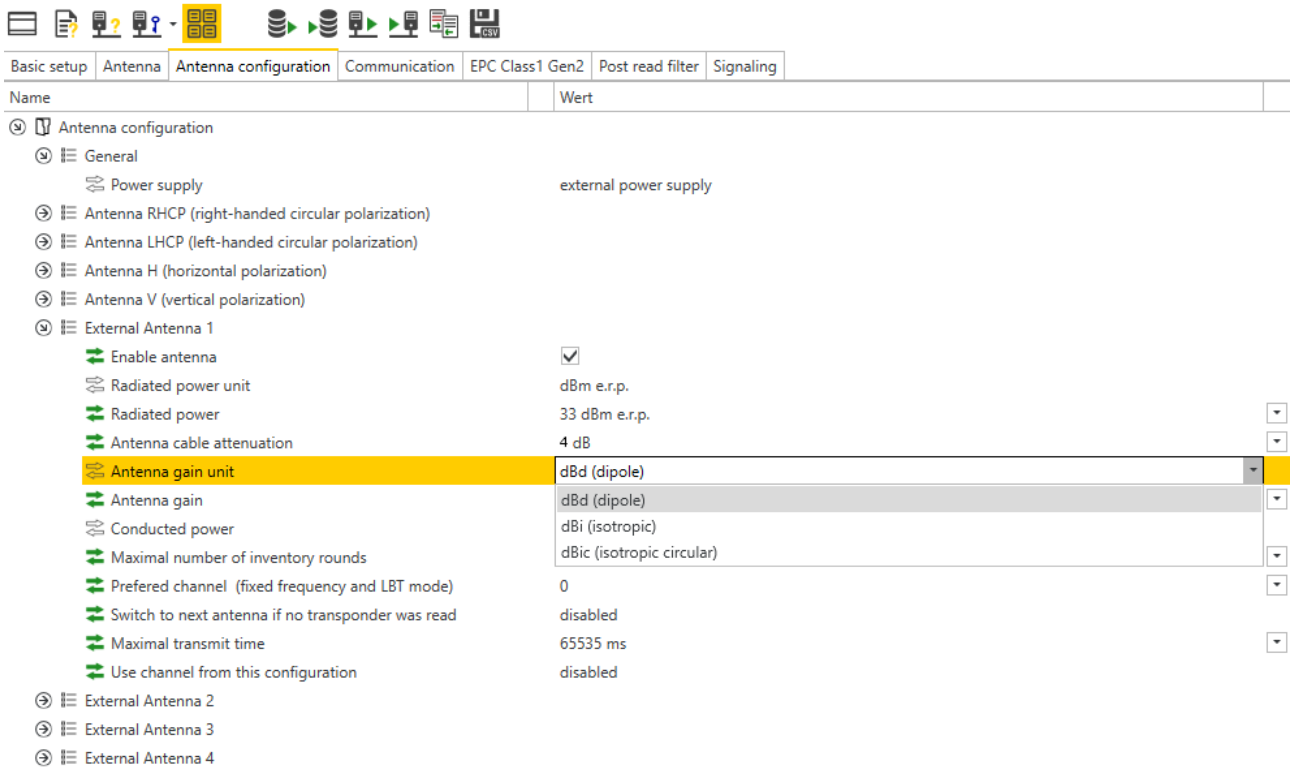


Fig. 28: Setting the unit for the antenna gain

▶ Set antenna gain at **Antenna gain** (here: 9.00).

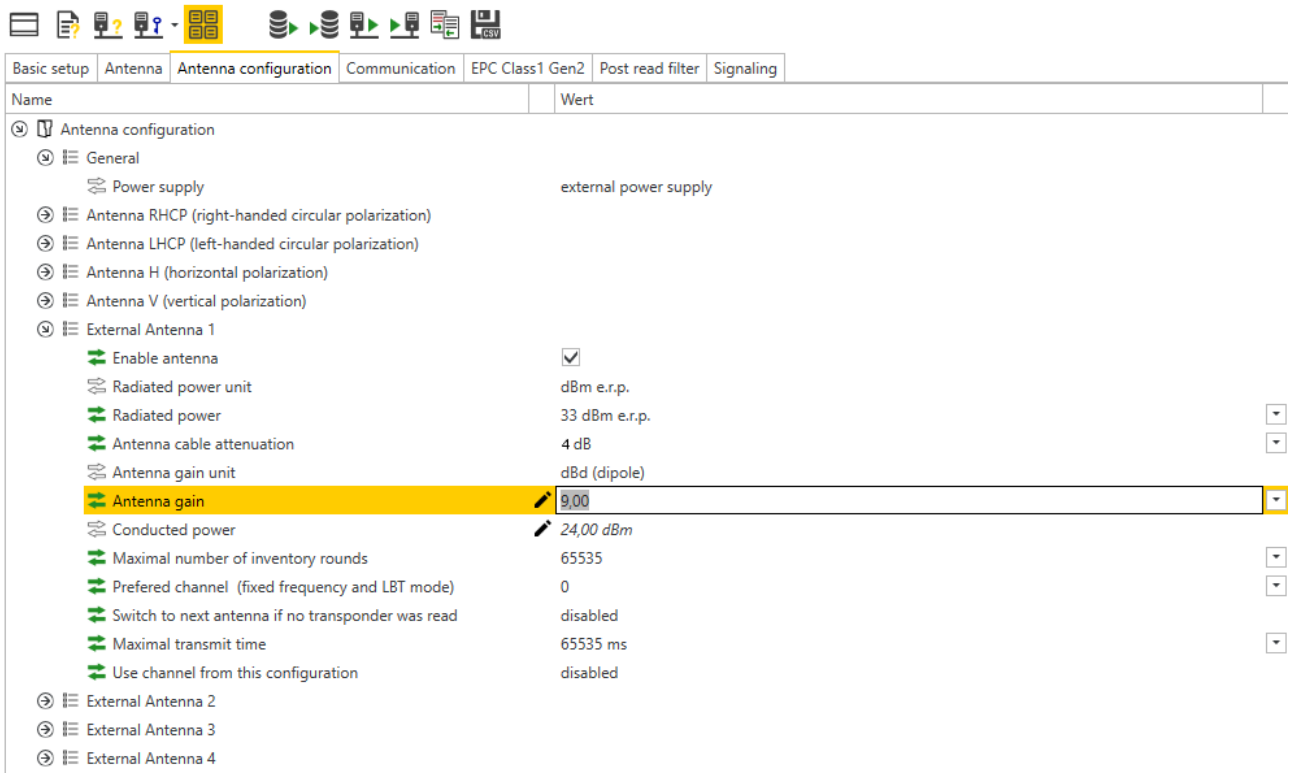


Fig. 29: Setting antenna gain

The power at the TNC socket (P_{cond}) is calculated automatically by the DTM and displayed at **Conducted power**.

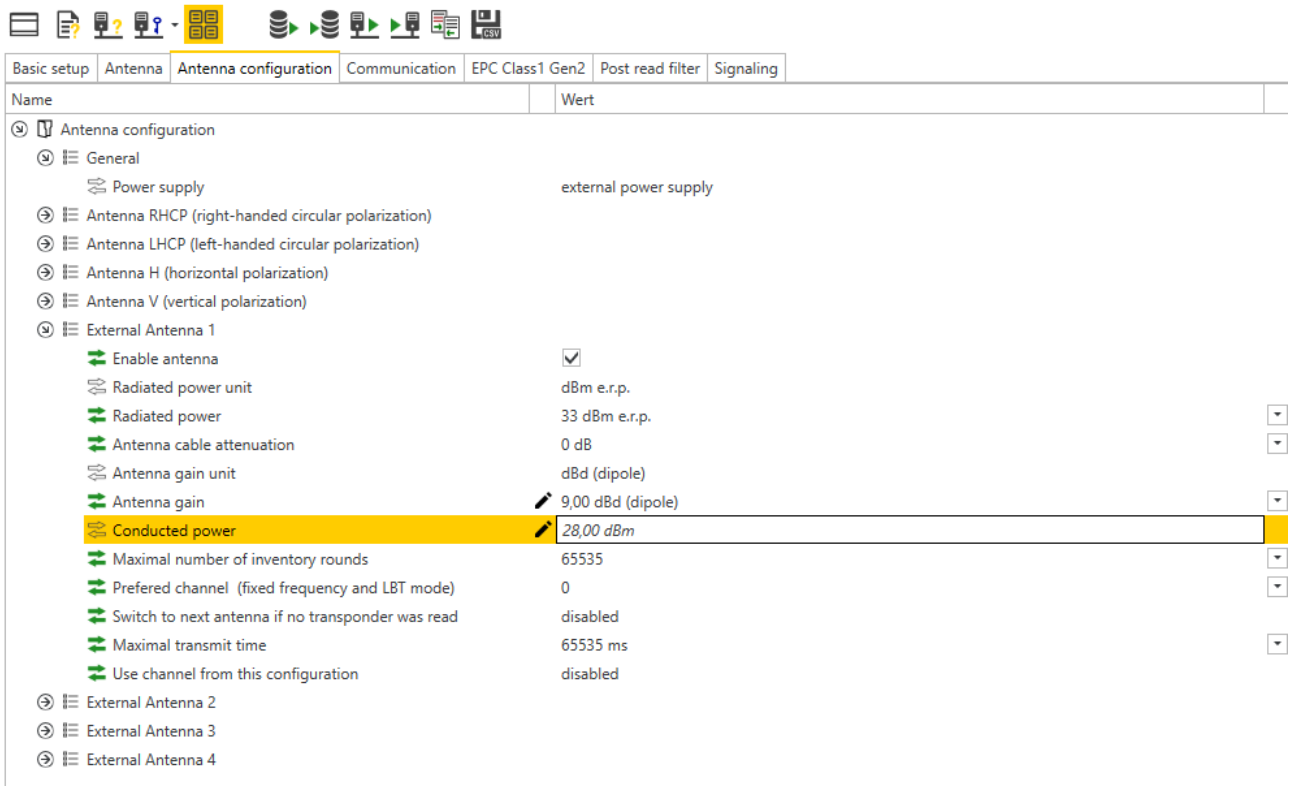


Fig. 30: Display of the power at the TNC socket

- ▶ Click **Accept** to save the settings.
- ▶ Set the power for each additional antenna separately.

7.1.7 Setting antenna polarization

The antenna polarization can be switched via the DTM. Switching the polarization makes it possible to change null spots caused by interference. The detection rate can be increased by switching the polarization. Polarization switching is suitable for example in single tag applications in particularly metallic environments.

The following graphics schematically illustrate the possibilities of antenna polarization.

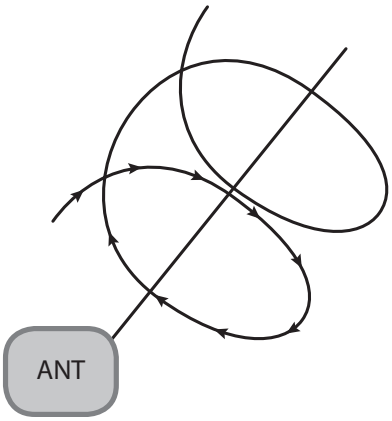


Fig. 31: Antenna polarization circular (RHCP)

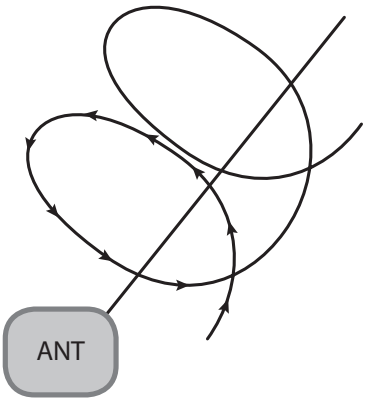


Fig. 32: Antenna polarization circular (LHCP)

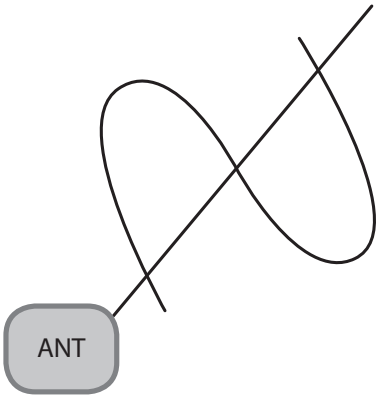


Fig. 33: Antenna polarization linear (vertical)

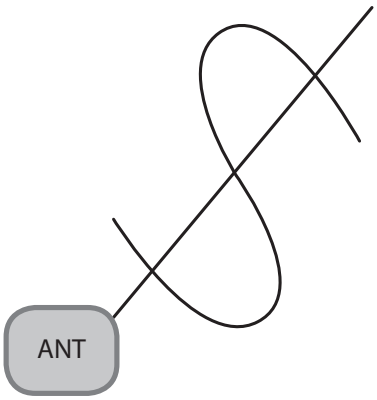


Fig. 34: Antenna polarization linear (horizontal)

Switching antenna polarization

Polarization switching is activated in the DTM via the multiplex settings.

- Set at **Antenna** → **Number of entries** the value **2**.
- Set at **Antenna** → **1st entry** the value **Antenna LHCP**.
- Set at **Antenna** → **2nd entry** the value **Antenna RHCP**.

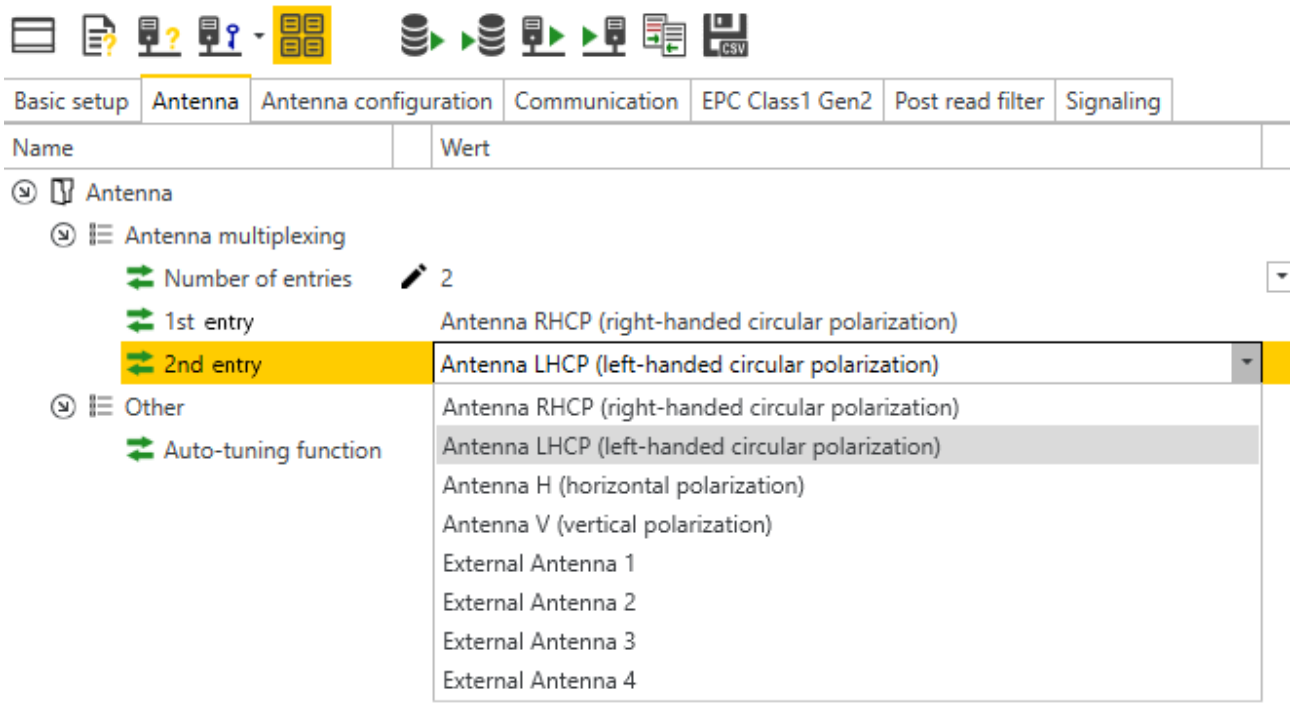


Fig. 35: Switching antenna polarization

- ▶ At **Antenna configuration** → **Maximal transmit time** set the time up to the polarization switch or activate the **Switch to next antenna if no transponder was read** option.
- ⇨ If the **Switch to next antenna if no transponder was read** option is enabled, the read/write head automatically switches after an Inventory operation without reading to the next multiplex sequence (**Entry**).

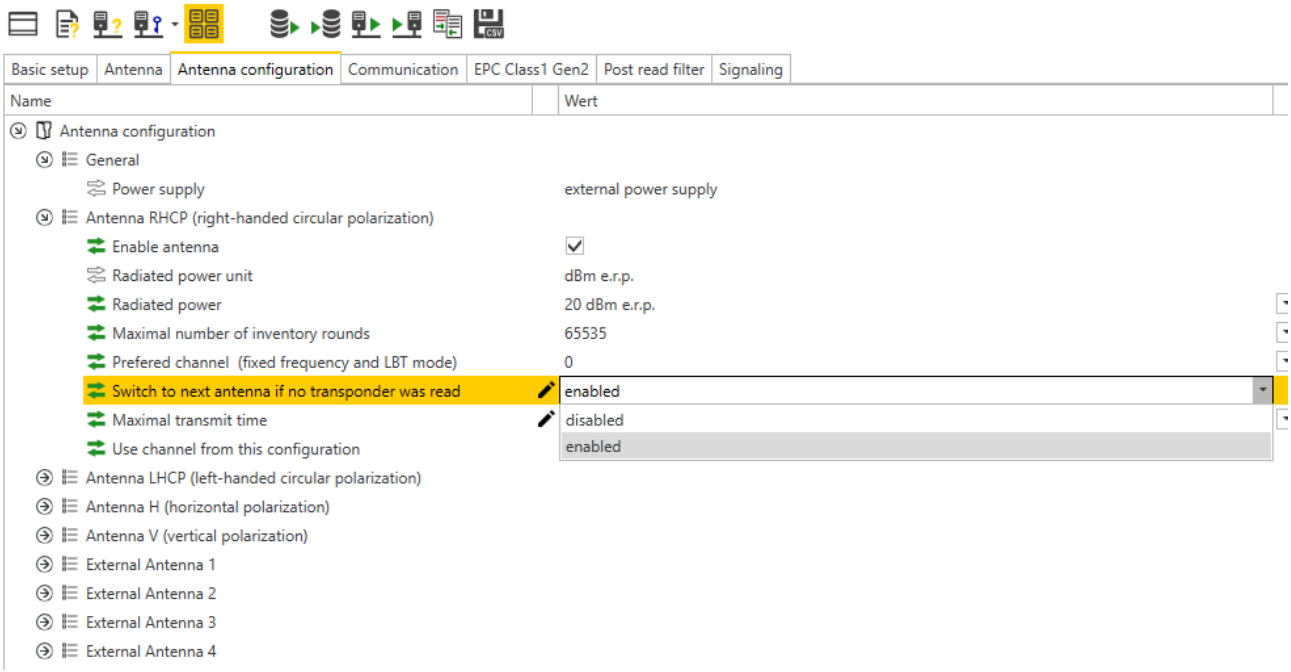


Fig. 36: Switching polarization automatically

7.1.8 Switching on presence sensing mode

In order to use the Continuous presence sensing mode command, the Presence sensing mode must be activated in the read/write head. In Presence sensing mode, the read/write heads are automatically switched on as soon as a tag is located in the detection range.

- ▶ Choose **Basic setup** → **General** → **Device mode** and set the **presence sensing mode** option.

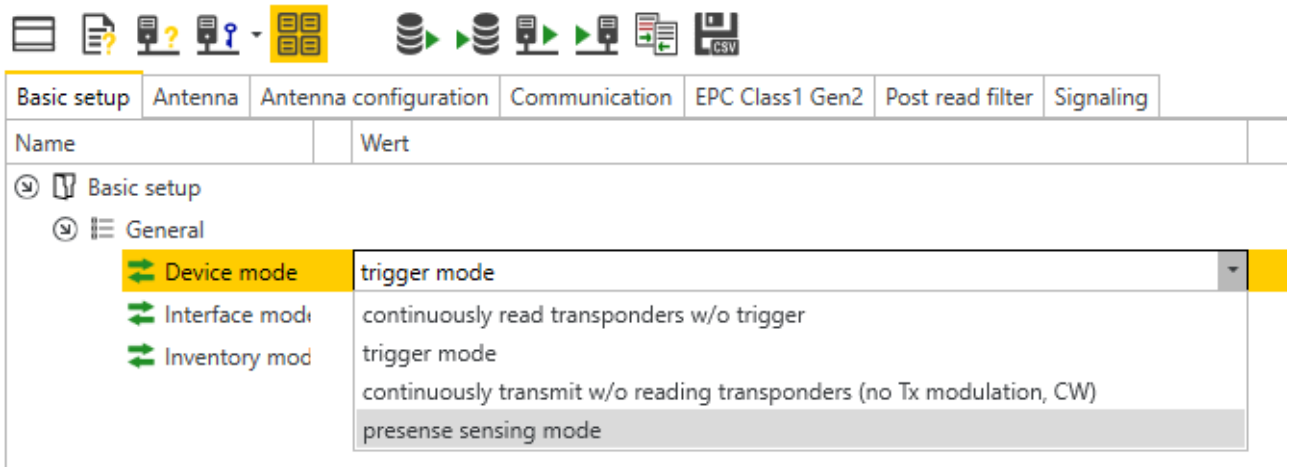


Fig. 37: Switching on presence sensing mode

The Advanced access level allows the **Tag data delay time** and **Carrier delay time** parameters to be set individually.

- **Tag data delay time**: Time in which the read/write head searches for a tag. If a tag is found, the field is switched on. In the Basic access level, the parameter is set by default to 100 ms.
- **Carrier delay time**: Time until the read/write head switches off the field after the last read operation. In the Basic access level, the parameter is set by default to 65535 ms.

NOTE
Report mode is recommended for the RFID test since the read tag information items appear in the RFID test window and do not have to be polled individually.

7.1.9 Transferring the RSSI value – communication

The **Communication** tab is used to set the parameters for the configuration of the deBus messages. All parameters and the settable values are written in the DTM.

Example: switch on RSSI transmission

- ▶ Switch on RSSI transmission: choose **Communication** → **Message data content** → **Transponder RSSI** and select the **enabled** option.
- ⇨ The RSSI value is displayed with the inventory in the read data.

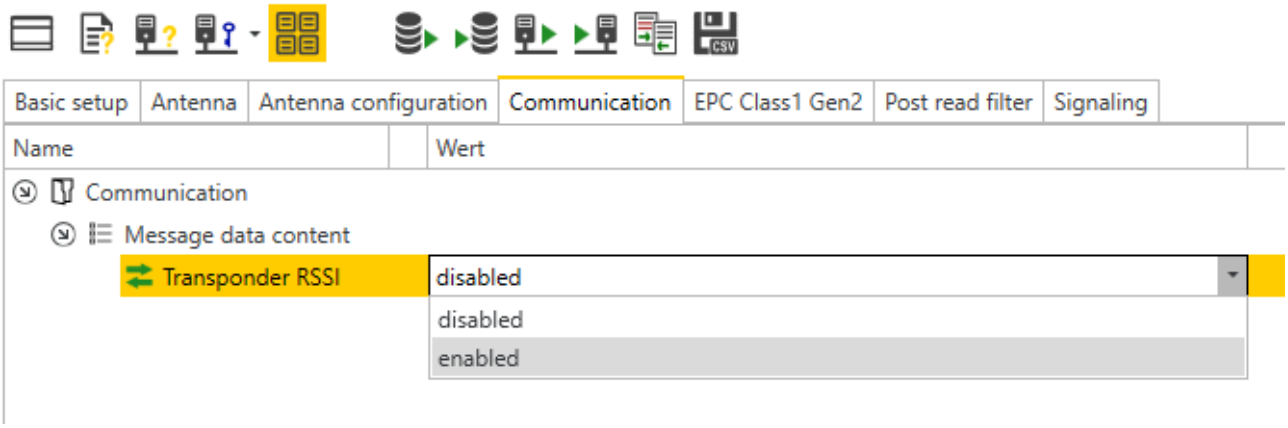


Fig. 38: Switching on RSSI transmission

7.1.10 Setting the air interface parameters – EPC Class 1 Gen 2

The EPC Class1 Gen2 tab is used to set the **EPC Class1 Gen2** parameters for the air interface. The parameters set are used if the read/write head performs an Inventory command. All parameters and the settable values are written in the DTM.

Example: set the tag reset

The tag reset function makes it possible to set how often a tag returns a signal to the read/write head with an Inventory operation. The tag reset function is only useful in single-tag applications.

- **EPC Class 1 Gen 2** → **Inventory** → **Inventory profile 1: Transponder reset** → enter value (here: 150 ms).

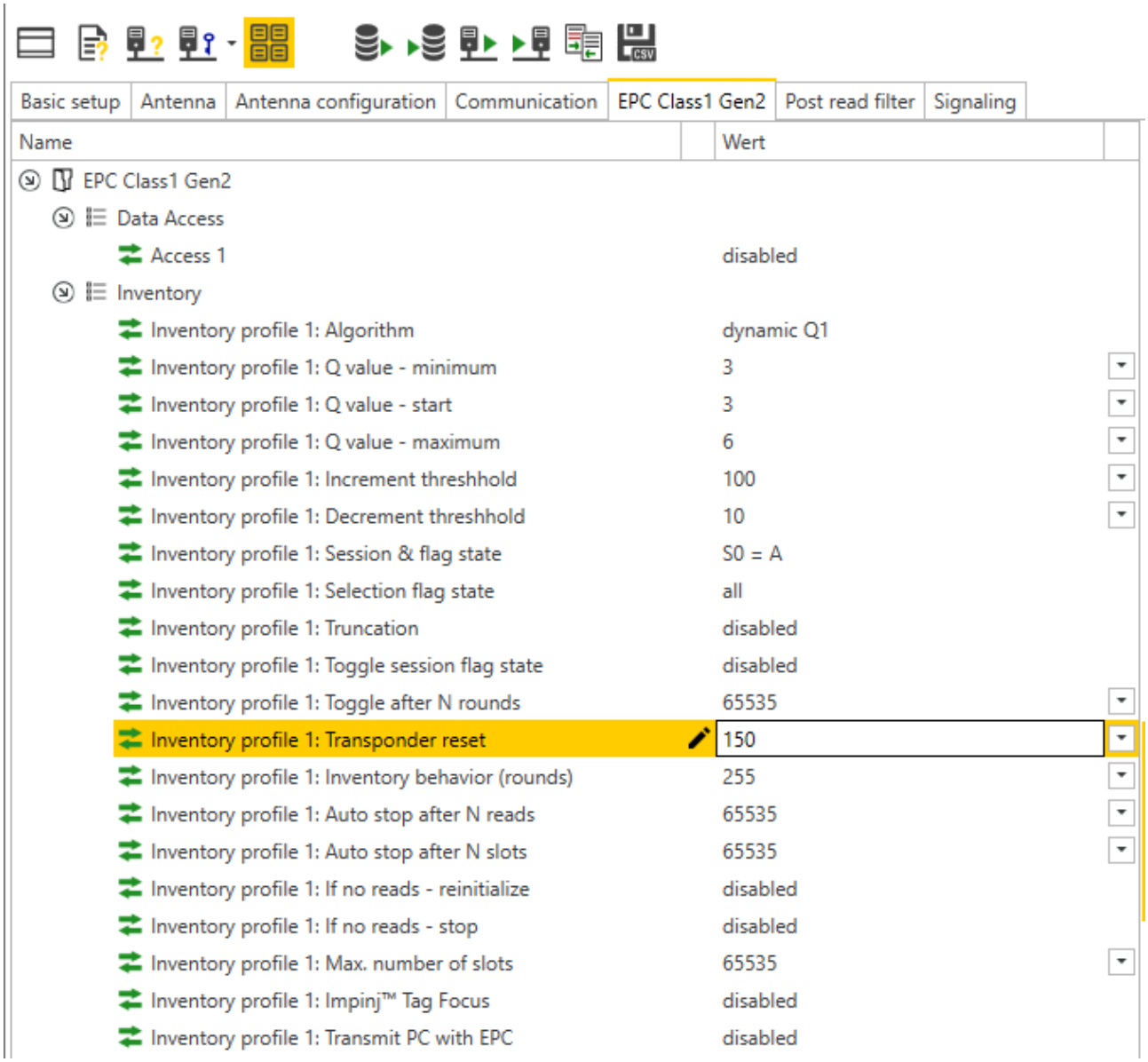


Fig. 39: Setting the tag reset

7.1.11 Setting the RSSI filter – Post read filter

The **Post read filter** tab enables parameters to be set in order to filter event messages.

The set filters do not reduce the data traffic on the air interface and are not suitable for multi-tag applications with many tags or high passing speeds. All parameters and the settable values are written in the DTM.

Example: set the RSSI filter

An RSSI filter makes it possible to prevent unwanted read operations. All read operations with an RSSI outside of the set limit values are filtered out and not displayed.

- At **Post read filter** → **RSSI filter** switch on the RSSI filter.

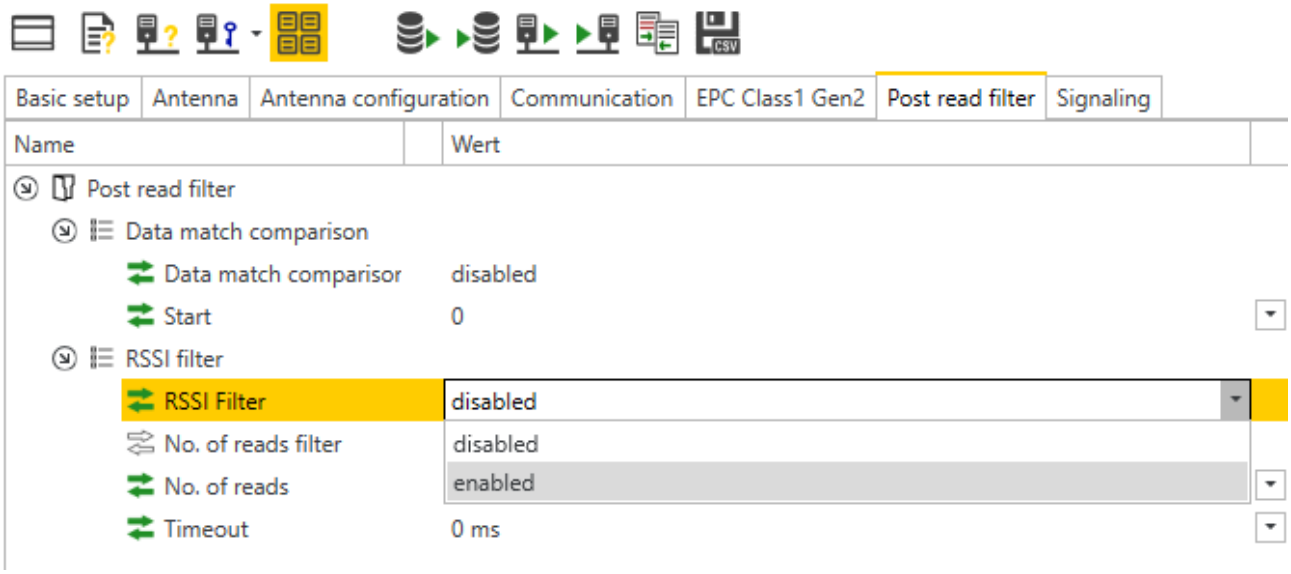


Fig. 40: Switching on the RSSI filter

▶ Set the limit values at **Post read filter** → **RSSI filter** → **Lower threshold**.
⇨ Example: all read operations below an RSSI value of -45 dBm are filtered out.

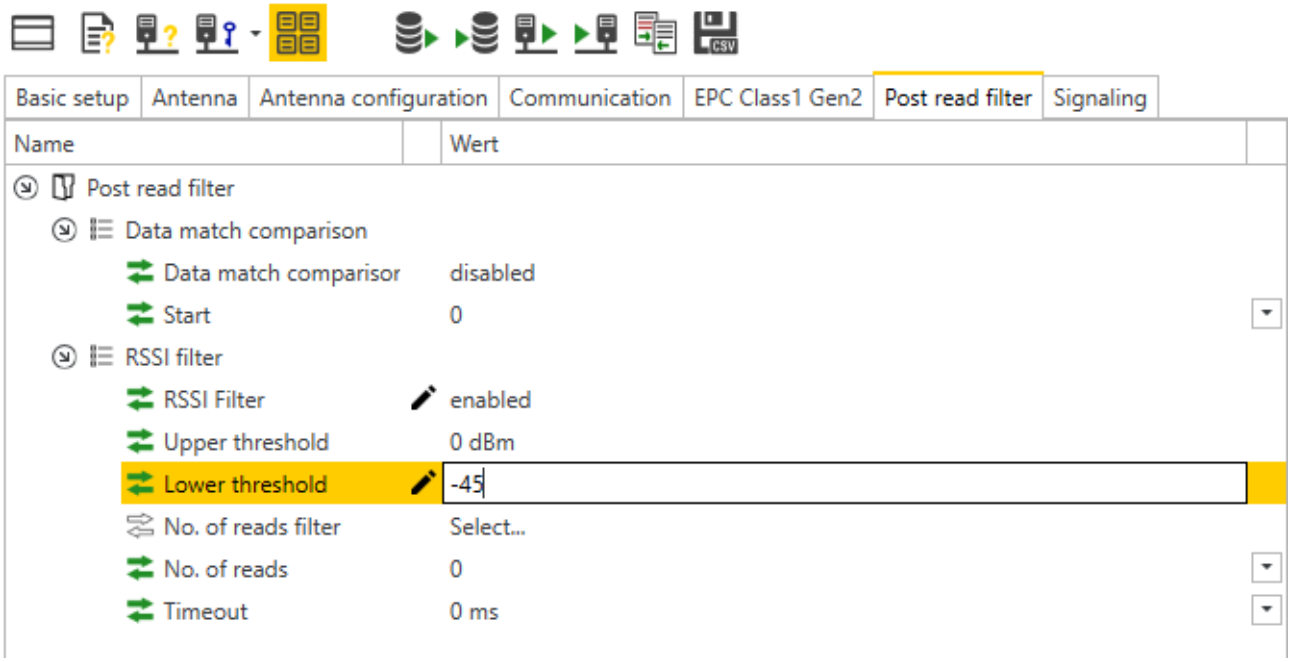


Fig. 41: Example – setting the limit value for RSSI

7.1.12 Setting LED indication – Signaling

The **Signaling** tab enables the default settings for the USER LEDs to be edited. All parameters and the settable values are written in the DTM.

7.2 Testing read/write heads with the DTM

The following functions can be executed via the RFID Test in the DTM:

- Displaying read data
- Displaying the protocol of the communication between host or PC and read/write head
- Logging of the interface communication between host or PC and read/write head
- Sending of user-specific deBus commands
- Writing of tags with a user-defined number
- Sending tag-specific commands

Requirements for the RFID test

- PACTware must be installed.
- The DTM for UHF read/write heads must be installed.
- The connection between the read/write head and the PC must be established.
- A project must have been created in PACTware.

7.2.1 Starting the RFID Test

- Right-click the device in the project tree.
- In the context menu choose **Additional functions** → **RFID Test**.

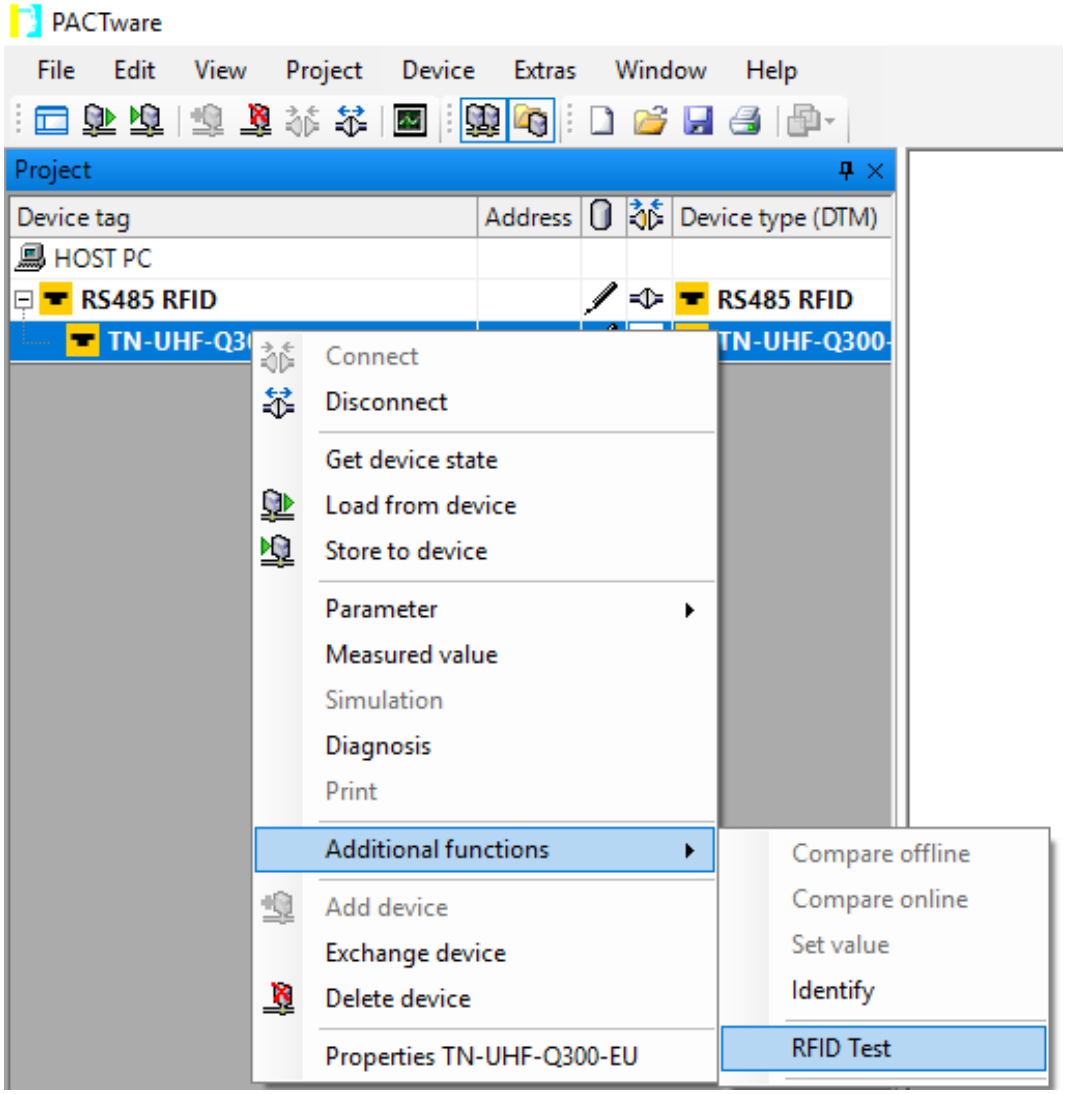


Fig. 42: Starting the RFID Test

7.2.2 Start window – overview

The **RFID Test** window consists of the following elements:

- Main menu
- Basic test
- Tag actions
- Reader Status
- Logger

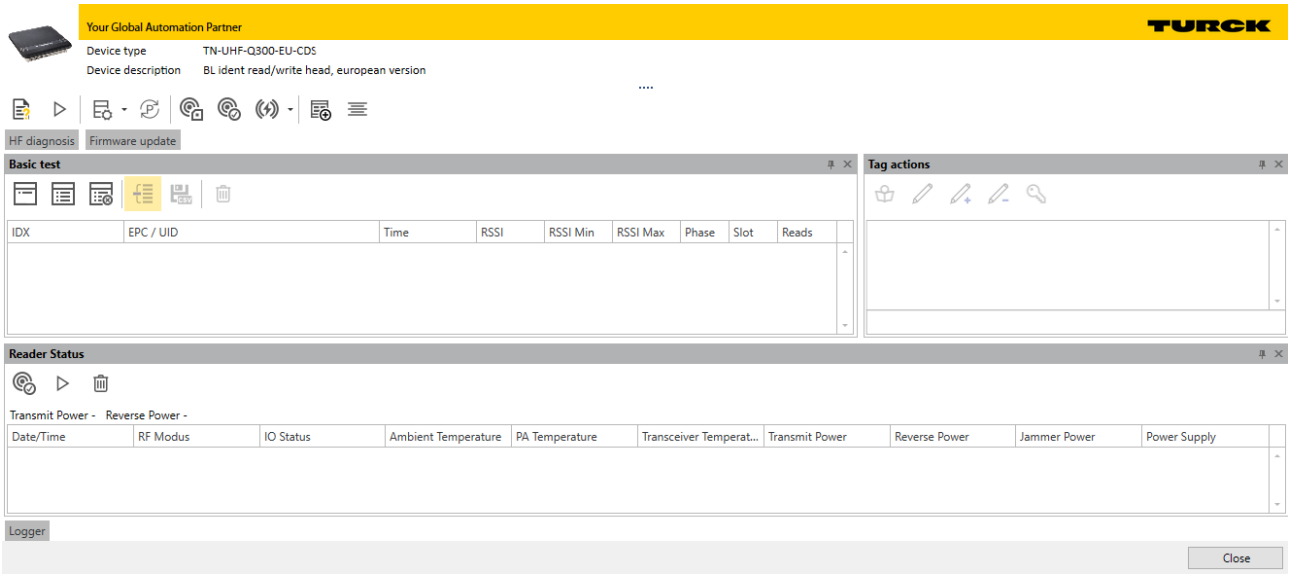


Fig. 43: RFID Test – overview of the start window

7.2.3 RFID Test – main menu

Fig. 44: RFID Test – main menu

The main menu provides the following functions:

Icon	Function	Description
	DTM help	Starts the DTM help.
	Trigger start/ON or Trigger stop/OFF	Starts the trigger for command execution (standard view). Ends the trigger for command execution (displayed after clicking the start button).
	Configure message content	Displays the content to be transferred with a read operation. The following can be selected: ■ Phase ■ RSSI ■ Socket ■ Time
	Switch mode (report/polling)	Switches between Report mode (automatic read/write) and polling mode (read/write started through an explicit polling command).
	Get read/write head status	Calls the status of the read/write head and provides the information in the **Logger** window.
	Get read/write head version	Calls the following information from the read/write head and provides the information in the **Logger** window: ■ Hardware revision ■ Firmware status ■ Serial number
	Reset the read/write head	Offers three ways of resetting the read/write head: ■ Voltage reset ■ Factory reset: Reset to factory settings ■ Reset read/write head status When resetting to factory settings, any modified transfer rate or RS485 address is not changed because the read/write head could not otherwise be addressed any longer.
	Set current window layout as default	Saves the individually set window layout.
	Reset window layout	Resets the window layout.
HF diagnosis	HF diagnosis	Opens the window for HF diagnostics.
Firmware update	Firmware update	Opens the window for the firmware update.

7.2.4 RFID Test – Basic test window

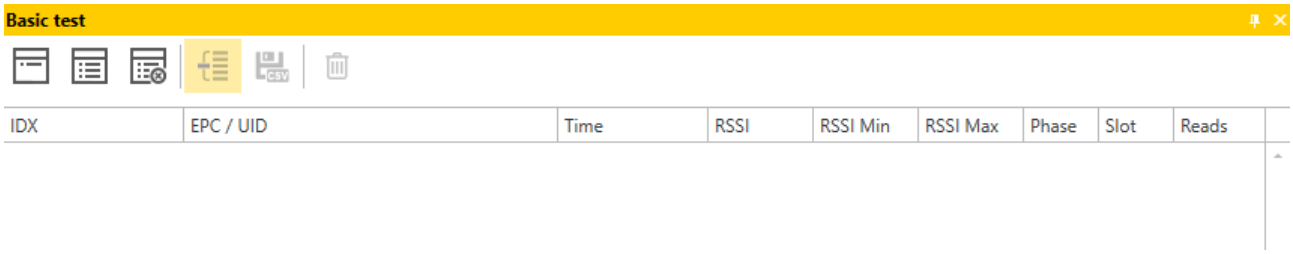


Fig. 45: RFID Test – Basic test window

The following functions are available in the **Basic test** window:

Icon	Function	Description
	Polling	Shows the first tag in the polling memory of the device in the tag list. The function is only available in polling mode.
	Poll all	Shows all tags in the polling memory of the device in the tag list. The function is only available in polling mode.
	Clear polled tags from read/write head	Clears the polling memory of the read/write head.
	Group tag list	Combines readings of tags with the same EPC.
	CSV export current values	Saves the tag list in CSV format.
	Clear tag list	Deletes the list of displayed tags.

The queried data is displayed in the tag list. The content of the message can be set via the **Configure message content** function.

NOTE
If the polling memory of the read/write head is full, the ERR LED is lit red and indicates an internal error.

7.2.5 RFID Test – Tag actions window

The functions in the **Tag actions** window are available if a tag is selected in the tag list of the **Basic test** window.

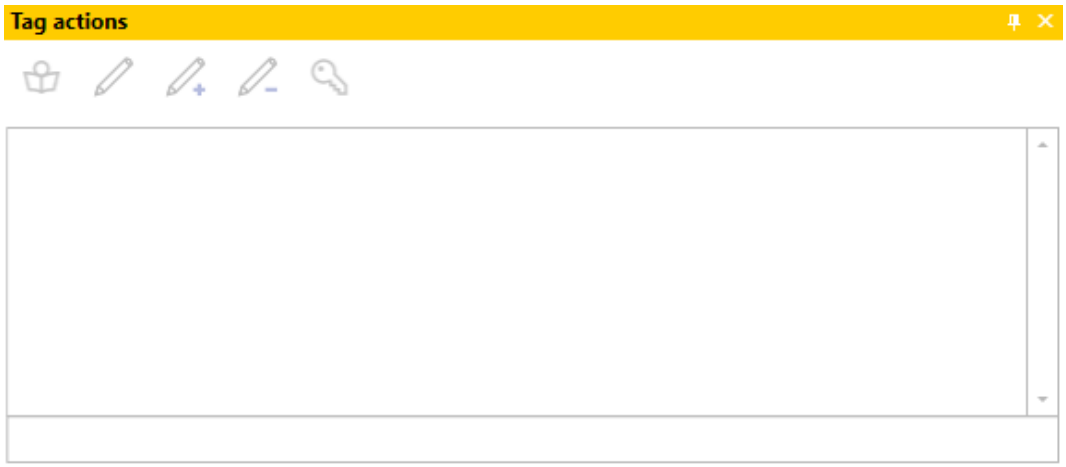


Fig. 46: RFID Test – Tag actions window

The following functions are available in the **Tag actions** window:

Icon	Function	Description
	Read RFID tag memory	Starts the read operation. The chip type is automatically displayed. One word is always read with the first read operation. The following parameters can be set for other read operations: ■ Memory bank (TID, EPC/UID, PC, access password or kill password) ■ Start word ■ Number of words The read data is displayed in the **Data** area.
	Write RFID tag memory	Starts the write operation. The chip type is automatically displayed. The following parameters can be set for the write operations: ■ Memory bank (TID, EPC/UID, PC, access password or kill password) ■ Start word ■ Number of words Data to be written is displayed in the **Data** area.
	Auto-increment	The EPC is incremented automatically by 1.
	Auto-decrement	The EPC is decremented automatically by 1.
	Switch on/off access password	Switches the password for write or read access on or off.

Example: execute tag actions

- Position tag in detection range of the read/write head.
- Activate the trigger for the read/write head in the main menu.

Fig. 47: Main menu – activating trigger

- **Basic test** window: execute polling command in order to display tag in the tag list.
- **Basic test** window: select tag from the tag list.

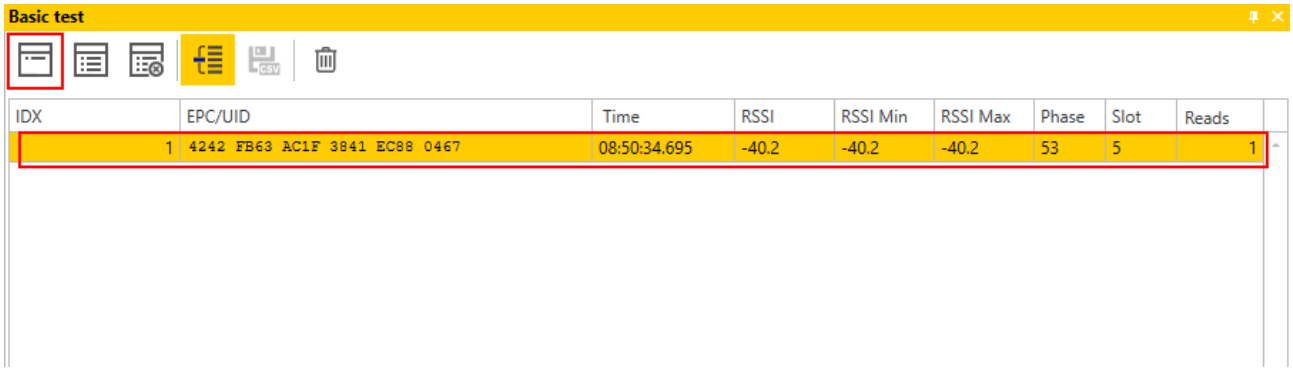


Fig. 48: Basic test – tag selection

- **Tag actions** window: To read, select **Memory location**, **Start word** or **Word length** and click the appropriate icon.
- To write enter values under **Data** and confirm with **OK**

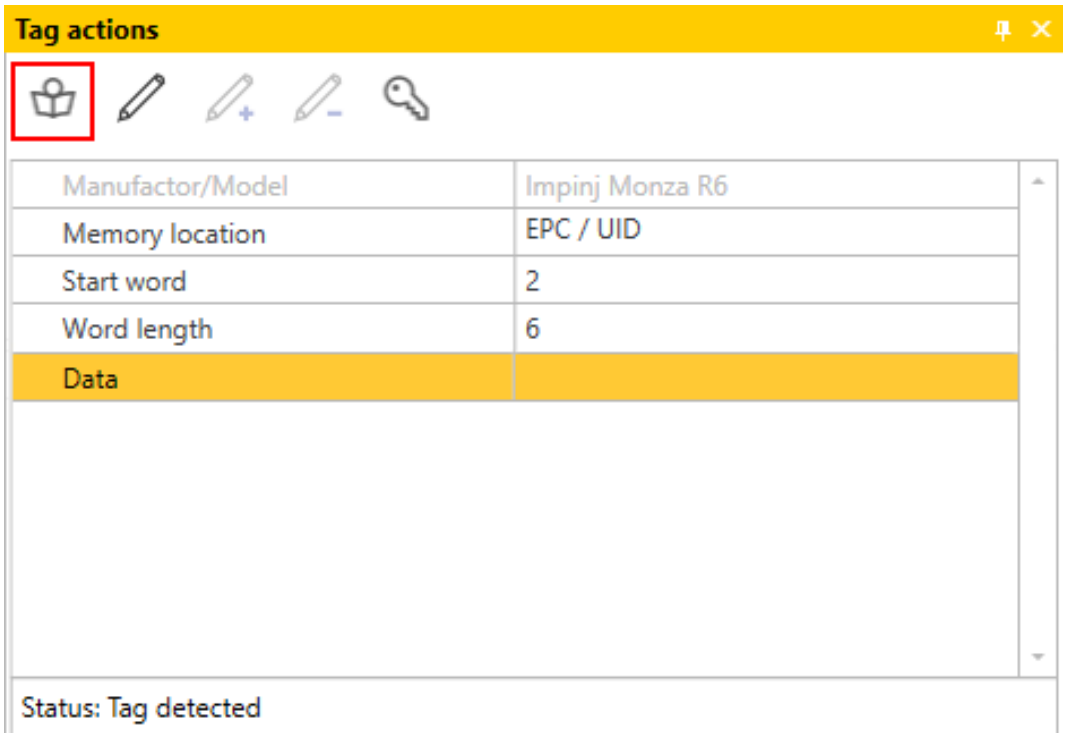


Fig. 49: Execute tag action (example: read)

⇨ Successful access is displayed via the status message at the bottom of the window.

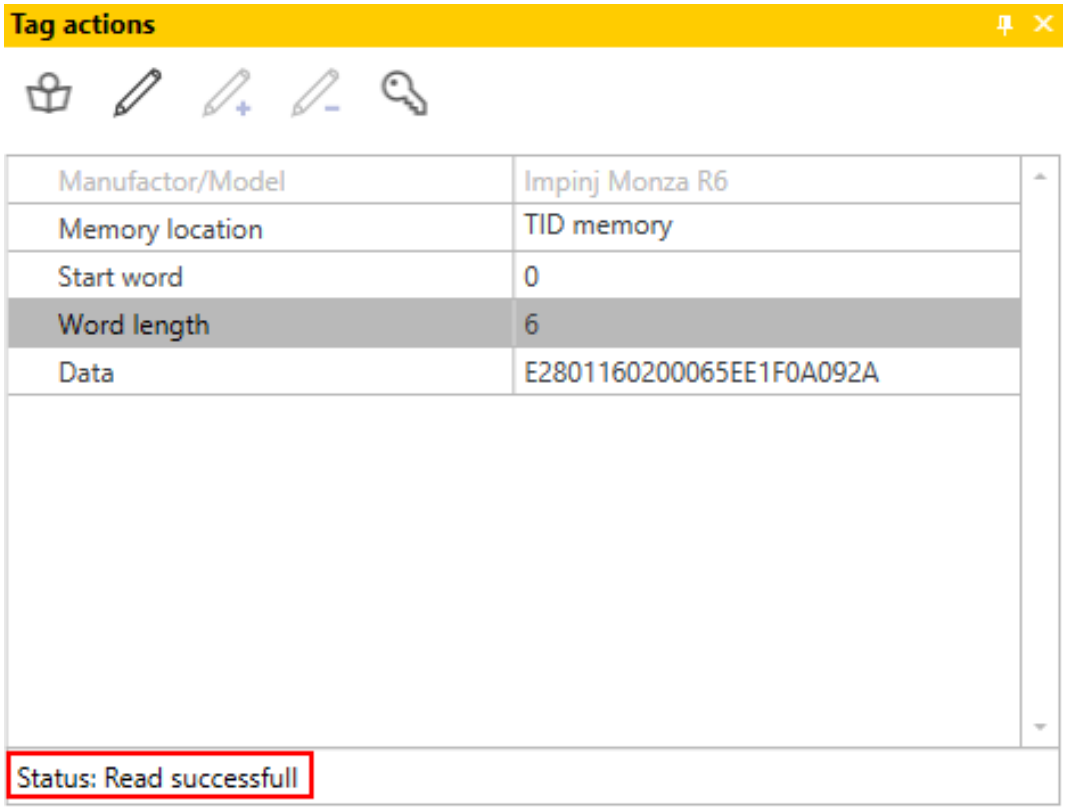


Fig. 50: Example: read successful

7.2.6 RFID Test – Logger window

The **Logger** window displays read/write information and error messages. The list can be cleared via the **Delete** button.

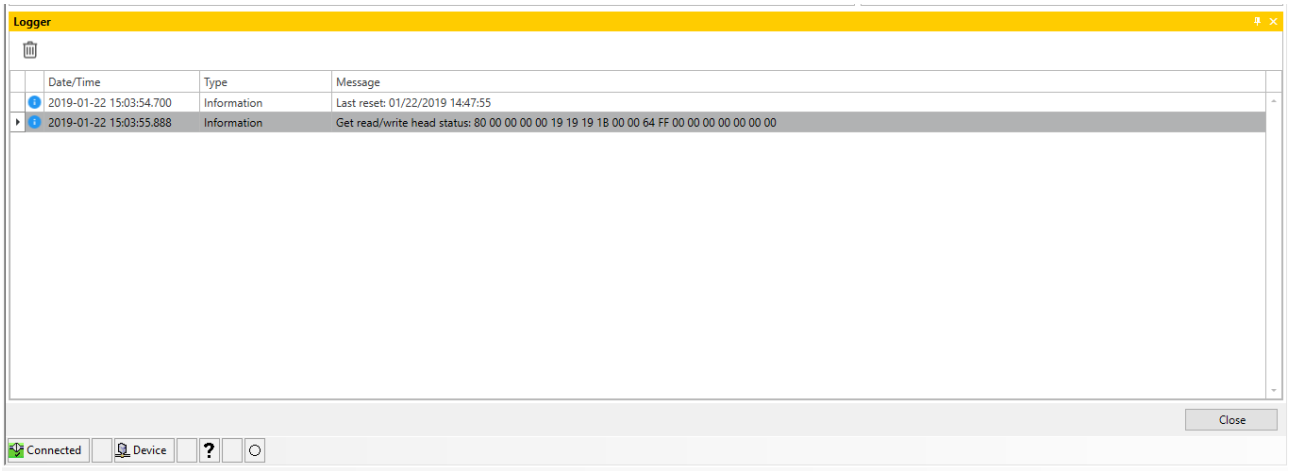


Fig. 51: Messages in the Logger window

7.2.7 HF diagnosis window

Interference frequencies affecting the respective channels are displayed in the **HF diagnosis** window.

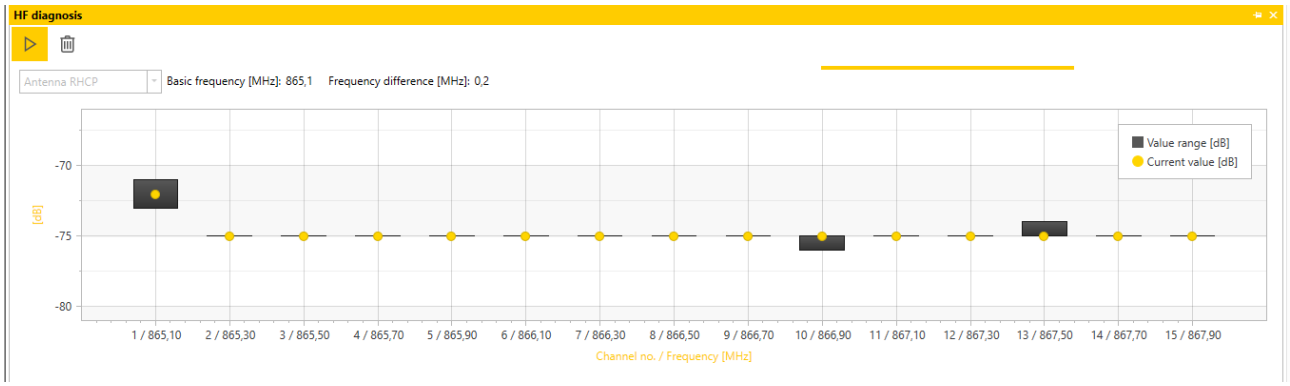


Fig. 52: HF diagnosis window

The following functions can be executed in the **HF diagnosis** window:

Icon	Function	Description
▷	Start/stop HF diagnosis	Starts or closes the HF diagnosis.
🗑	Clear values	Deletes the displayed values.

7.3 System start

7.3.1 Requirements

- Windows operating system (at least Windows 7)
- Visual Studio 2012/2013/2015 (**https://www.visualstudio.com/de-de/products/visual-studio-express-vs.aspx**)
- Application Builder (**https://www.microsoft.com/en-us/download/details.aspx?id=38819**)
- SDK (e.g. Turck_AM335x_RFID_SDKx.msi) is installed.

7.3.2 Creating the application

Creating the application in C/C++

- ▶ Start Visual Studio.
- ▶ Choose **New Project…**.
- ▶ In the left window area choose **Templates → Visual C++ → Windows Embedded Compact → AM335x_Turck_RFID_SDKx**.
- ▶ In the central window area choose **Win32 Console Application**.
- ▶ Enter a name in the lower area.
- ▶ Click **OK**.
- ⇨ In Application Builder View, Visual Studio starts with an automatically generated basic console application.
- ▶ Add `#include "windows.h"`.
- ▶ Add the following lines to the function **wmain**:

```
printf("Hello World\n");
Sleep(5000);
```

- ▶ Compile the application: Press [F7].

Creating the application in C#

- ▶ Start Visual Studio.
- ▶ Choose **New Project…**.
- ▶ In the left window area choose **Templates → Other Languages→ Visual C# → Windows Embedded Compact → AM335x_Turck_RFID_SDKx**.
- ▶ Choose **Console Application** in the central window area.
- ▶ Enter a name in the lower area.
- ▶ Click **OK**.
- ⇨ In Application Builder View, Visual Studio starts with an automatically generated basic console application.
- ▶ At the beginning of the **main cs** file add `using System.Threading;` and `using System.Diagnostics;`.
- ▶ Add the following lines in the Main Method:

```
Debug.WriteLine("Hello World");
Thread.Sleep(5000);
```

- ▶ Compile the application: Press [F7].

7.4 Drivers

7.4.1 Ethernet

The Ethernet driver supports the AM335x CPSW3G Ethernet controller in single MAC mode.

The MAC0 port of the Ethernet controller is available at the X300 port. The Ethernet port is available for the system as Ethernet device CPSW3G1.

The FTP server and the Telnet server are enabled by default (without authentication).

For details on WinSock see
https://msdn.microsoft.com/EN-US/library/ee494651%28v=VS.80,d=hv.2%29.aspx.

7.4.2 NandFlash

The NandFlash driver supports the AM335x GPMC controller and the fitted NAND Flash card.

The NANDFlash driver is loaded automatically at startup und transparently maps the FAT partition to the directory **/Mounted_Volume** by default.

The device registry is stored permanently in the NANDFlash file system by default.

7.4.3 UART

The UART driver supports the UART1 device of the AM335x. There are no flow control signals available. The UART1 is accessed as COM1:device, which uses the Win32 Serial Port API.
For details see
https://msdn.microsoft.com/EN-US/library/ee488234%28v=VS.80,d=hv.2%29.aspx.

It is possible to influence the UART Receive FIFO by adjusting the value of the key **HKEY_LOCAL_MACHINE\Drivers\BuiltIn\UART1\RxFifoTriggerLevel** (possible values UART1-63).

7.4.4 GPIO

The GPIO driver supports the AM335x GPIOs.

The GPIO driver is available as the GIO1:device. The driver is accessible via the driver functions of the stream interface. For details see **https://msdn.microsoft.com/EN-US/library/ee488234%28v=VS.80,d=hv.2%29.aspx**.

The GPIOs have to be identified by their GPIO ID. They are defined in the **gpio_defines.h** file included in the SDK. The BSP supports the following GPIOs:

AM335xGPIO	GPIO-ID	GPIO define
GPIO0_6	6	GPIO_6
GPIO0_8	8	GPIO_8
GPIO0_12	12	GPIO_12
GPIO0_13	13	GPIO_13
GPIO0_19	19	GPIO_19
GPIO1_8	40	GPIO_40
GPIO1_9	41	GPIO_41
GPIO1_15	47	GPIO_47
GPIO1_16	48	GPIO_48
GPIO1_17	49	GPIO_49
GPIO1_18	50	GPIO_50
GPIO1_19	51	GPIO_51
GPIO1_20	52	GPIO_52
GPIO1_21	53	GPIO_53
GPIO1_22	54	GPIO_54
GPIO1_23	55	GPIO_55
GPIO1_24	56	GPIO_56
GPIO2_9	73	GPIO_73
GPIO2_22	86	GPIO_86
GPIO2_23	87	GPIO_87
GPIO2_24	88	GPIO_88
GPIO2_25	89	GPIO_89
GPIO3_9	105	GPIO_105
GPIO3_10	106	GPIO_106
GPIO3_15	111	GPIO_111
GPIO3_16	112	GPIO_1112
GPIO3_17	113	GPIO_113
GPIO3_18	114	GPIO_114
GPIO3_19	115	GPIO_115

The driver supports the following IOControl codes (defined in **gpio_ioctls.h**):

IOCTL_GPIO_SETBIT

Sets the corresponding GPIO to level 1.

Parameter	
`lpInBuffer`	Pointer to DWORD containing the GPIO ID to be set

IOCTL_GPIO_CLRBIT

Sets the corresponding GPIO to level 0.

Parameter	
`lpInBuffer`	Pointer to DWORD containing the GPIO ID to be set

IOCTL_GPIO_GETBIT

Reads from the level of the corresponding GPIO.

Parameter	
`pInBuffer`	Pointer to DWORD containing the GPIO ID to be read
`lpOutBuffer`	Pointer to DWORD receiving the current level

IOCTL_GPIO_SETMODE

Configures the mode of the corresponding GPIO.

Parameter	
`lpInBuffer`	Pointer to an array of two DWORDs containing the GPIO ID (array element 0) and the mode (array element 1) to be set

The following modes are supported (defined in **gpio_defines.h**):

- `GPIO_DIR_OUTPUT`: Configure GPIO as an output
- `GPIO_DIR_INPUT`: Configure GPIO as an input
- `GPIO_INT_LOW_HIGH`: Enable interrupt on rising edge
- `GPIO_INT_HIGH_LOW`: Enable interrupt on falling edge
- `GPIO_INT_LOW`: Enable low level interrupt
- `GPIO_INT_HIGH`: Enable high-level interrupt
- `GPIO_DEBOUNCE_ENABLE`: Enable debounce

IOCTL_GPIO_GETMODE

Returns the current mode of the corresponding GPIO.

Parameter	
`lpInBuffer`	Pointer to a DWORD containing the GPIO ID
`lpOutBuffer`	Pointer to a DWORD receiving the mode

IOCTL_GPIO_CONFIGURE_PERIODIC_UPDATE

Sets the corresponding GPIO to 1Hz or 2Hz in toggle mode.

Parameter	

`lpInBuffer`	Pointer to an `IOCTL_GPIO_PERIODIC_INFO` structure (defined in **gpio_ioctls.h**), containing the GPIO ID and the periodic mode to be set:
	`typedef struct {`
	`UINT uGpioID;`
	`UINT dwPeriMode;`
	`} IOCTL_GPIO_PERIODIC_INFO;`

The following modes are supported (defined in **gpio_defines.h**):

- `GPIO_PERIODIC_DISABLE`: Disables periodic toggling of the GPIO
- `GPIO_PERIODIC_1HZ`: Enables periodic 1Hz toggling of the GPIO
- `GPIO_PERIODIC_2HZ`: Enables periodic 2Hz toggling of the GPIO

IOCTL_GPIO_CONFIGURE_AUTO_RECOVERY

Configures the auto recovery feature of a GPIO.

Parameter	

`lpInBuffer`	Pointer to an `IOCTL_OC_INFO` structure (defined in **gpio_ioctls.h**), containing the GPIO ID to enable auto-recovery, the overcurrent detection of the GPIO and the mode (automatic or manual recovery):
	`typedef struct {`
	`UINT uGpioID;`
	`UINT uOCGpioID;`
	`BYTE bOCLevel;`
	`BYTE bMode;`
	`} IOCTL_GPIO_OC_INFO;`

The following trigger level conditions are supported for the `bOCLevel` parameter (defined in **gpio_defines.h**):

- `GPIO_AUTOREC_OC_LOWLEV`: Overcurrent condition of uGpioID is triggered by a low level interrupt at the uOCGpioID
- `GPIO_AUTOREC_OC_HIGHLEV`: Overcurrent condition of uGpioID is triggered by a high level interrupt at the uOCGpioID
- `GPIO_AUTOREC_OC_SAMELEV`: Overcurrent condition of uGpioID is triggered when uOCGpioID is at the same level
- `GPIO_AUTOREC_OC_DIFFLEV`: Overcurrent condition of uGpioID is triggered when uOCGpioID is at the complementary level

The following modes are supported for the `bMode` parameter (defined in **gpio_defines.h**):

- `GPIO_AUTOREC_OC_DISABLED`: Overcurrent detection is disabled
- `GPIO_AUTOREC_OC_SHUTDOWN`: Overcurrent condition is handled by a shutdown (manual recovery)
- `GPIO_AUTOREC_OC_AUTOREC`: Overcurrent condition is handled with a shutdown and automatic recovery after a safe amount of time

IOCTL_GPIO_GETIRQ

Returns the IRQ of the corresponding GPIO.

Parameter	
`lpInBuffer`	Pointer to a DWORD containing the GPIO ID
`lpOutBuffer`	Pointer to a DWORD receiving the IRQ number

IOCTL_GPIO_SET_DEBOUNCE_TIME

Sets the debounce time of a GPIO (bank).

Parameter	

`lpInBuffer`	Pointer to an `IOCTL_GPIO_SET_DEBOUNCE_TIME_IN` structure (defined in **gpio_ioctls.h**), containing the GPIO ID and the debounce time to be set:
	`typedef struct {`
	`UINT gpioId;`
	`UINT debounceTime;`
	`} IOCTL_GPIO_SET_DEBOUNCE_TIME_IN;`

The debounce time is calculated as follows:

Debounce time = (DEBOUNCETIME + 1) × 31 µs. The debounce time applies globally to all GPIOs of the same bank.

IOCTL_GPIO_GET_DEBOUNCE_TIME

Returns the debounce time for the GPIO (bank).

Parameter	
`lpInBuffer`	Pointer to a DWORD containing the GPIO ID
`lpOutBuffer`	Pointer to a DWORD receiving the debounce time

IOCTL_GPIO_INIT_INTERRUPT

Initializes a GPIO interrupt.

Parameter	

`lpInBuffer`	Pointer to an `IOCTL_GPIO_INIT_INTERRUPT_INFO` structure (defined in **gpio_ioctls.h**):
	`typedef struct {`
	`UINT uGpioID;`
	`DWORD dwSysIntrID;`
	`HANDLE hEvent;`
	`} IOCTL_GPIO_INIT_INTERRUPT_INFO;`

uGpioID must be set to the GPIO ID and hEvent must be set to an Event Handle. The used SysIntr is returned in the dwSysIntrID element.

IOCTL_GPIO_ACK_INTERRUPT

Acknowledges a GPIO interrupt.

Parameter	

`lpInBuffer`	Pointer to an `IOCTL_GPIO_INTERRUPT_INFO` structure (defined in **gpio_ioctls.h**):
	`typedef struct {`
	`UINT uGpioID;`
	`DWORD dwSysIntrID;`
	`} IOCTL_GPIO_INTERRUPT_INFO;`

IOCTL_GPIO_DISABLE_INTERRUPT

Disables a GPIO interrupt.

Parameter	
`lpInBuffer`	Pointer to an `IOCTL_GPIO_INTERRUPT_INFO` structure

7.4.5 SPI

The SPI driver supports the AM335x MCSPI0 interface. The SPI0 interface is available as the SPI1:device.

The driver supports the following functions (defined in **sdk_spi.h**):

HANDLE SPIOpen(LPCTSTR pSpiName)

Opens the driver for subsequent use.

Parameter	
`pSpiName`	String containing the device name (`SPI1:`)

Return value
Handle to the driver

VOID SPIClose(HANDLE hContext)

Closes the driver after use.

Parameter	
`hContext`	Handle returned by `SPIOpen()`

BOOL SPILockController(HANDLE hContext, DWORD dwTimeout)

Locks the access to the driver to the current thread.

Parameter	
`hContext`	Handle returned by `SPIOpen()`
`dwTimeout`	Timeout for enabling the lock

Return value	
`TRUE`	Successful
`FALSE`	not successful

BOOL SPIUnLockController(HANDLE hContext)

Unlocks the access to the driver.

Locks the access to the driver to the current thread.

Parameter	
`hContext`	Handle returned by `SPIOpen()`

Return value	
`TRUE`	Successful
`FALSE`	not successful

BOOL SPIConfigure(HANDLE hContext, DWORD address, DWORD config)

Configures the SPI device for subsequent actions.

Parameter	
`hContext`	Handle returned by `SPIOpen()`
`address`	Chipselect number (only CS0 is supported)
`config`	DWORD containing the desired configuration. The configuration has to be set according to the `MCSPI_CH0CONF` register description in the AM335x Technical Reference Manual.

Return value	
TRUE	Successful
FALSE	not successful

BOOL SPIEnableChannel(HANDLE hContext)

Enables the channel configured by the `SPIConfigure()` address parameter and therefore activates the corresponding chipselect.

Parameter	
`hContext`	Handle returned by `SPIOpen()`

Return value	
`TRUE`	Successful
`FALSE`	not successful

BOOL SPIDisableChannel(HANDLE hContext)

Disables the channel previously enabled by `SPIEnableChannel()`.

Parameter	
`hContext`	Handle returned by `SPIOpen()`

Return value	
`TRUE`	Successful
`FALSE`	not successful

BOOL SPISetSlaveMode(HANDLE hContext)

Configures the SPI controller for slave mode.

Parameter	
`hContext`	Handle returned by `SPIOpen()`

Return value	
`TRUE`	Successful
`FALSE`	Not successful

DWORD SPIRead(HANDLE hContext, DWORD size, VOID *pBuffer)

Reads from the SPI bus.

Parameter	
`hContext`	Handle returned by `SPIOpen()`
`size`	Number of bytes to be read
`pBuffer`	Pointer to the receive buffer

Return value
Number of bytes actually read

DWORD SPIWrite(HANDLE hContext, DWORD size, VOID *pBuffer)

Writes to the SPI bus.

Parameter	
`hContext`	Handle returned by `SPIOpen()`
`size`	Number of bytes to write
`pBuffer`	Pointer to the send buffer

Return value
Number of bytes actually written

DWORD SPIWriteRead(HANDLE hContext, DWORD size, VOID *pOutBuffer, VOID *pInBuffer)

Reads and writes from/to the SPI bus simultaneously.

Parameter	
`hContext`	Handle returned by `SPIOpen()`
`size`	Number of bytes to be read/written
`pOutBuffer`	Pointer to the send buffer
`pInBuffer`	Pointer to the receive buffer

Return value
Number of bytes actually read/written

DWORD SPIAsyncWriteRead(HANDLE hContext, DWORD size, VOID *pOutBuffer, VOID *pInBuffer)

Reads and writes from/to the SPI bus simultaneously via DMA.

Parameter	
`hContext`	Handle returned by `SPIOpen()`
`size`	Number of bytes to be read/written
`pOutBuffer`	Pointer to the send buffer
`pInBuffer`	Unused, set to `NULL`

Return value
Value of the size parameter

DWORD SPIWaitForAsyncWriteReadComplet(HANDLE hContext, DWORD size, VOID *pOutBuffer)

Waits for the DMA transfer to be completed.

Parameter	
`hContext`	Handle returned by `SPIOpen()`
`size`	Number of bytes to be read/written
`pOutBuffer`	Pointer to the receive buffer

Return value
Value of the size parameter

7.4.6 I2C

The I2C proxy driver (accessible in usermode) supports the AM335x I2C0 interface. The I2C0 interface is available as the I2C1:device.

The I2C proxy driver is available using the file API (`CreateFile()`,`ReadFile()`, `WriteFile()`,`SetFilePointer()`).

`SetFilePointer()` is used to select the base subaddress which is accessed in subsequent ReadFile() or WriteFile() calls. The following IOControl codes are available for selecting the I2C device address and baud rate (defined in **i2cproxy.h**):

IOCTL_I2C_SET_SLAVE_ADDRESS

Sets the slave address of the I2C device to be accessed.

Parameter	
`lpInBuffer`	Pointer to a DWORD containing the slave address

IOCTL_I2C_SET_SUBADDRESS_MODE

Sets the subaddress mode.

Parameter	
`lpInBuffer`	Pointer to a DWORD containing the desired subaddress mode. The following modes are available (defined in **sdk_i2c.h**): ■ `I2C_SUBADDRESS_MODE_0`: No device subaddresses ■ `I2C_SUBADDRESS_MODE_8`: 1 byte of subaddresses ■ `I2C_SUBADDRESS_MODE_16`: 2 bytes of subaddresses ■ `I2C_SUBADDRESS_MODE_24`: 3 bytes of subaddresses ■ `I2C_SUBADDRESS_MODE_32`: 4 bytes of subaddresses

IOCTL_I2C_SET_BAUD_INDEX

Sets the I2C speed (baud rate).

Parameter	
`lpInBuffer`	Pointer to a DWORD containing the desired baud rate. The following baud rates are available (defined in **sdk_i2c.h**): ■ `SLOWSPEED_MODE`: 100 kHz ■ `FULLSPEED_MODE`: 400 kHz ■ `HIGHSPEED_MODE_1P16`: 1.6 MHz ■ `HIGHSPEED_MODE_2P4`: 2.4 MHz ■ `HIGHSPEED_MODE_3P2`: 3.2 MHz

7.4.7 TBOX API

The TBOX API is implemented in **TBOX_API_Q300_LIB.lib** and **TBOX_API_Q300_DLL.dll** as required and documented in **TBOX_API_Q300.h**.

7.4.8 Using the application

In order for the device to respond to deploy/debug requests from the Application Builder, device-side components must be manually activated, for example via telnet, after the device is started:

- ▶ Connect to the running device via telnet (default IP address): 192.168.1.100).
- ▶ Execute `start conmanclient3 & cmaccept3` on the device command line.
- ⇨ The Application Builder is ready for use.
- ▶ Execute the application on the device: Press [F5] (or **Debug** → **Start Debugging**).
- ⇨ The Application Builder switches to Debug View and executes the application.
- ⇨ The output window (Debug) displays the string `Hello World`.

7.4.9 Debugging the application

- ▶ Placing a breakpoint at a specific line of source code: Click the vertical gray bar in front of the line.
- ⇨ The breakpoint is indicated with a red dot.

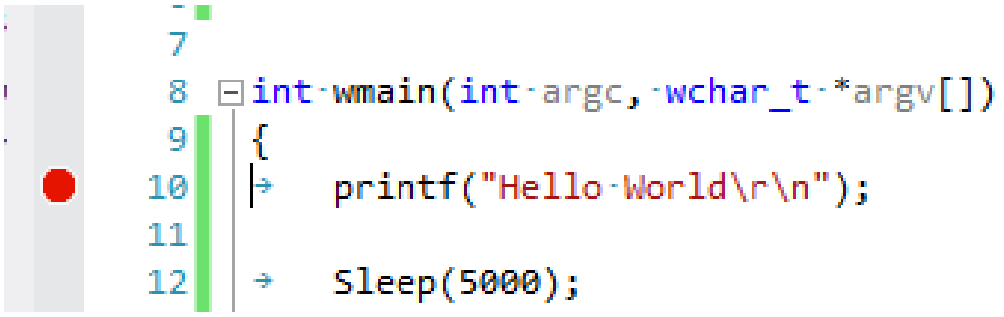


Fig. 53: Debugging the application

The application stops at the selected breakpoint if the application is run the next time on the target device. This is indicated by a yellow arrow on top of the breakpoint.

In general, remote debugging using Visual Studio works much like the local debugging of an application, including single-steps, Callstack display, Memory display and so on.

7.4.10 Using a network socket in C#

Network communication can be implemented in C# by means of the Sockets class. The following example code creates and opens a TCP socket to a server at 192.168.4.80 on port 80.

For detailed information about communication using the socket, refer to **https://msdn.microsoft.com/en-us/library/system.net.sockets.socket_members(v=vs.90)**.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Diagnostics;

using System.Net;
using System.Net.Sockets;

namespace TurckWinSock
{
 class Program
 {
  static void Main(string[] args)
  {
   Debug.WriteLine("Hello World");

   //create an endpoint with the server IP and port 80
   IPAddress ip = new IPAddress(new byte[] {192, 168, 4, 80});
   IPEndPoint IpEnd = new IPEndPoint(ip, 80);

   //create a socket object
   Socket TestSocket = new Socket(IpEnd.AddressFamily, Socket-
Type.Stream, ProtocolType.Tcp);

   try
   {
    //connect the socket with the server
    TestSocket.Connect(IpEnd);

    //check the connection
    if (TestSocket.Connected)
    {
     Debug.WriteLine("socket connected");
    }
    else
    {
     Debug.WriteLine("socket connection failed");
    }

    //do something with the socket
    Thread.Sleep(5000);

    //close the socket
    TestSocket.Shutdown(SocketShutdown.Both);
    TestSocket.Close();
   }
   catch (Exception e)
   {
    Debug.WriteLine("exception while connecting socket");
   }
  }
 }
}
```

7.4.11 Using a TBOX-API library

In order to abstract the functionality of the DXPs and COM ports, an API library (TBOX API) is provided.

An overview and basic description of the functionality provided by the API library can be found in the **TBOX_API.h** header file.

Procedure with a C/C++ application

- In order to use the API library in a C/C++ application: Simply include the **TBOX_API_Q300.h** header file and statically link the **TBOX_API_Q300_LIB.lib** to your application.
- Copy **TBOX_API.h** and **TBOX_API_LIB.lib** to the application directory and add the library as an additional dependency to the linker.
- Select **PROJECT** → **Project Properties** (Alt + F7).
- In the left window area select **Configuration Properties** → **Linker** → **Input**.
- In the right window area, click on the **Additional Dependencies** drop-down list and select **<Edit…>**.

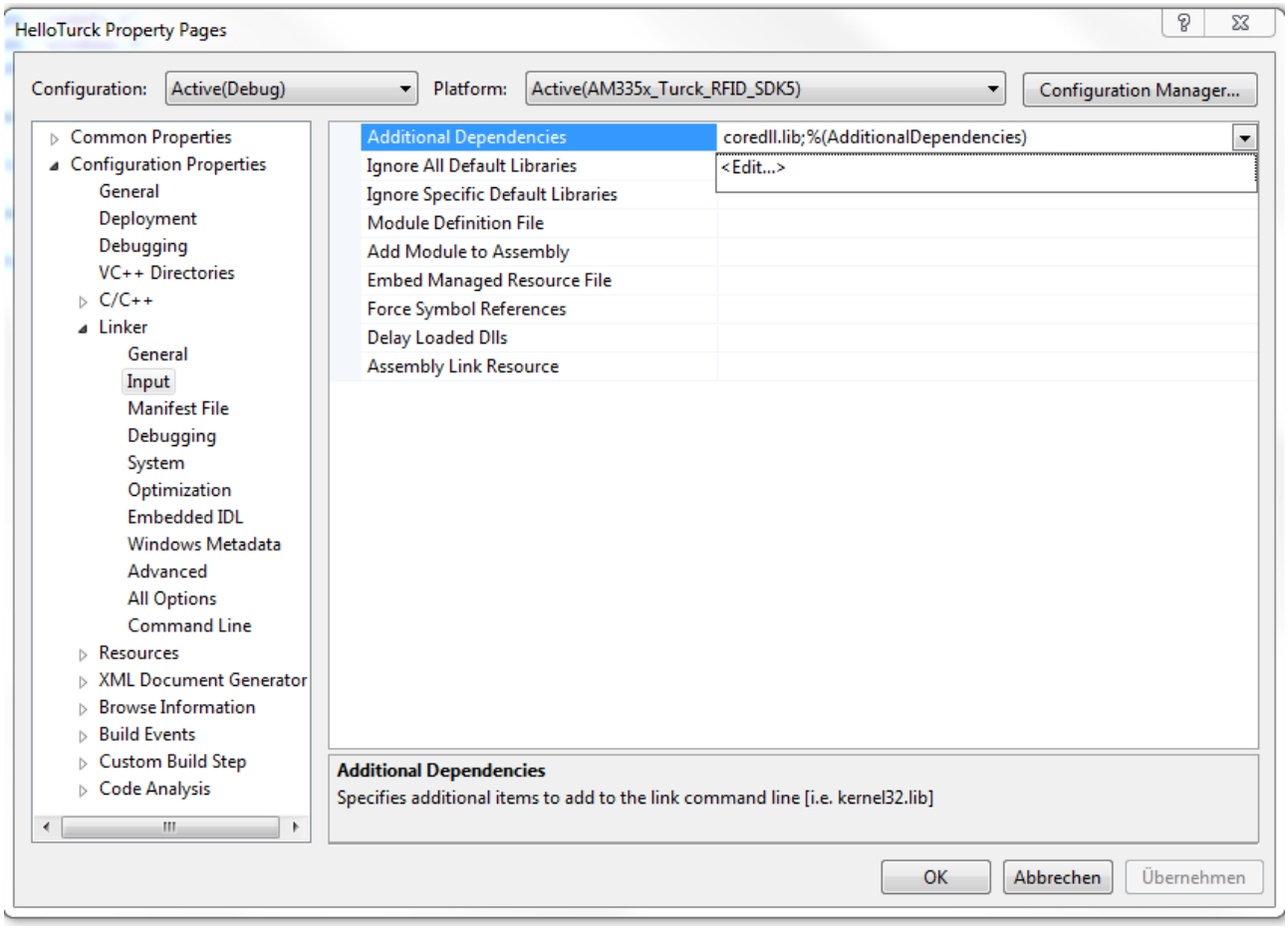


Fig. 54: Property pages

▶ In the editing fieldbus enter **TBOX_API_Q300_LIB.lib** and confirm.

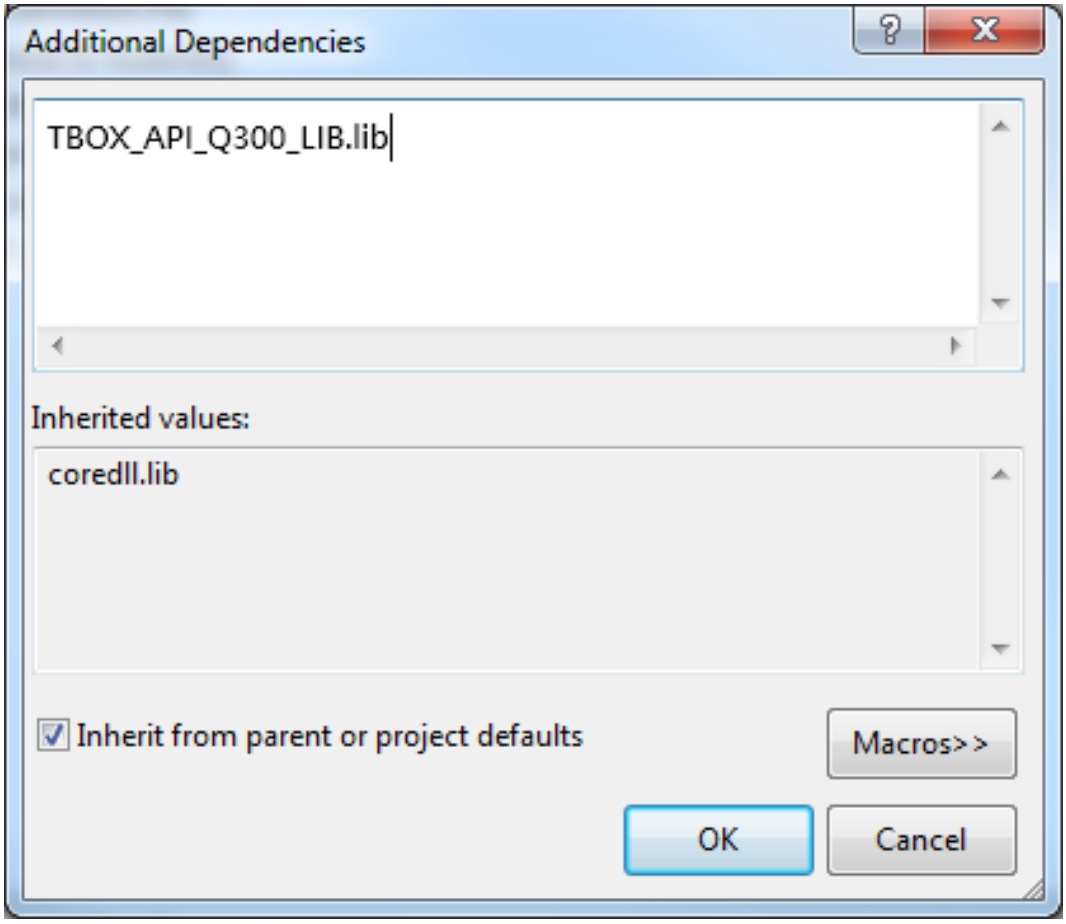


Fig. 55: Additional dependencies

▶ Include the **TBOX_API_Q300.h** header file by adding the line `#include "TBOX_API_Q300.h";` at the top of the main source file.

⇨ It is possible to use the TBOX API via the application. Before using other API functions, `TBOX_Init();` must be called first. This resets all settings to their default values and puts the hardware in a defined state.

Example 1: let the green App LED flash:

```
TBOX_SYS_LED_GreenSet(LED_APP, LED_2HZ);
```

Example 2: Configure a DXP for input and read it's current value:

```
TBOX_DXP_Init(1, FALSE, FALSE);
DWORD dwVal = TBOX_DXP_Get(1);
printf("DXP 18 Level: %d\n", dwVal);
```

Example 3: Read all DXP inputs at once:

```
dwVal = TBOX_DXP_Get(-1);
for(unsigned int i=0; i < 4; i++)
  printf("DXP %d Level: %d\n", i, (dwVal & (1 << i)) >> i);
```

The values of all DXPs are read out by using **-1** as the DxpNo parameter. The return value is a bit field with the DXP levels value encoded in the bit corresponding to it's number (e.g. bit0 for DXP 0, bit1 for DXP1 and so on).

7.4.12 Procedure with a C# application

In order to use the API in C# applications, it is also provided as a DLL (**TBOX_API_DLL.dll**). The DLL is not part of the default OS image, so it must be copied to the device first.

Before using a function from the DLL, it must first be imported in your C# application. Example:

```
[DllImport("TBOX_API_Q300_DLL.dll")]
public extern static int TBOX_Init();

[DllImport("TBOX_API_Q300_DLL.dll")]
public extern static void TBOX_SYS_LED_GreenSet(int LedNo, int
State);
```

After the import, the DLL can be called from within the C# application. Example:

```
TBOX_Init();
TBOX_SYS_LED_GreenSet(4, 0xCCCC);
```

8 Operation

8.1 LEDs

The device has the following LED indicators:

- Power supply
- Group and bus errors
- Status
- Diagnostics

PWR LED	Meaning
Off	No power supply
Green	Power supply error-free
Yellow	Undervoltage within tolerance range
Red	Undervoltage outside of tolerance range

RFON LED	Meaning
Off	Wireless field deactivated
Green	Wireless field activated

DATA LED	Meaning
Off	No tag in the field, no data transfer
Yellow flashing	Tag in the field, data transfer via the air interface

DIAG LED	Meaning
Off	No error
Red	Error

The following multicolor LEDs are freely programmable. The tables below describe the default display functions.

DXP LEDs (digital channels, LEDs DXP0...3)		
LED green	**LED red**	**Meaning**
Off	Off	No I/O signal present
Lit	Off	I/O signal present
Off	Lit	Overload at output
Flashing	Flashing	Overload of the auxiliary voltage

APPL LED	Meaning
Flashing white	Wink command active

9 Troubleshooting

If the device does not work as expected, proceed as follows:

- Exclude environmental disturbances.
- Check the connections of the device for errors.
- Check device for parameterization errors.

If the malfunction persists, the device is faulty. In this case, decommission the device and replace it with a new device of the same type.

9.1 Rectifying errors

Errors are displayed by an ERR LED lit red on the device.

Calling error messages in the DTM and rectifying them

NOTE
Contact Turck if the error persists after the read/write head is reset.

- Right-click the device in the project tree.
- Select **Diagnosis** in the context menu.

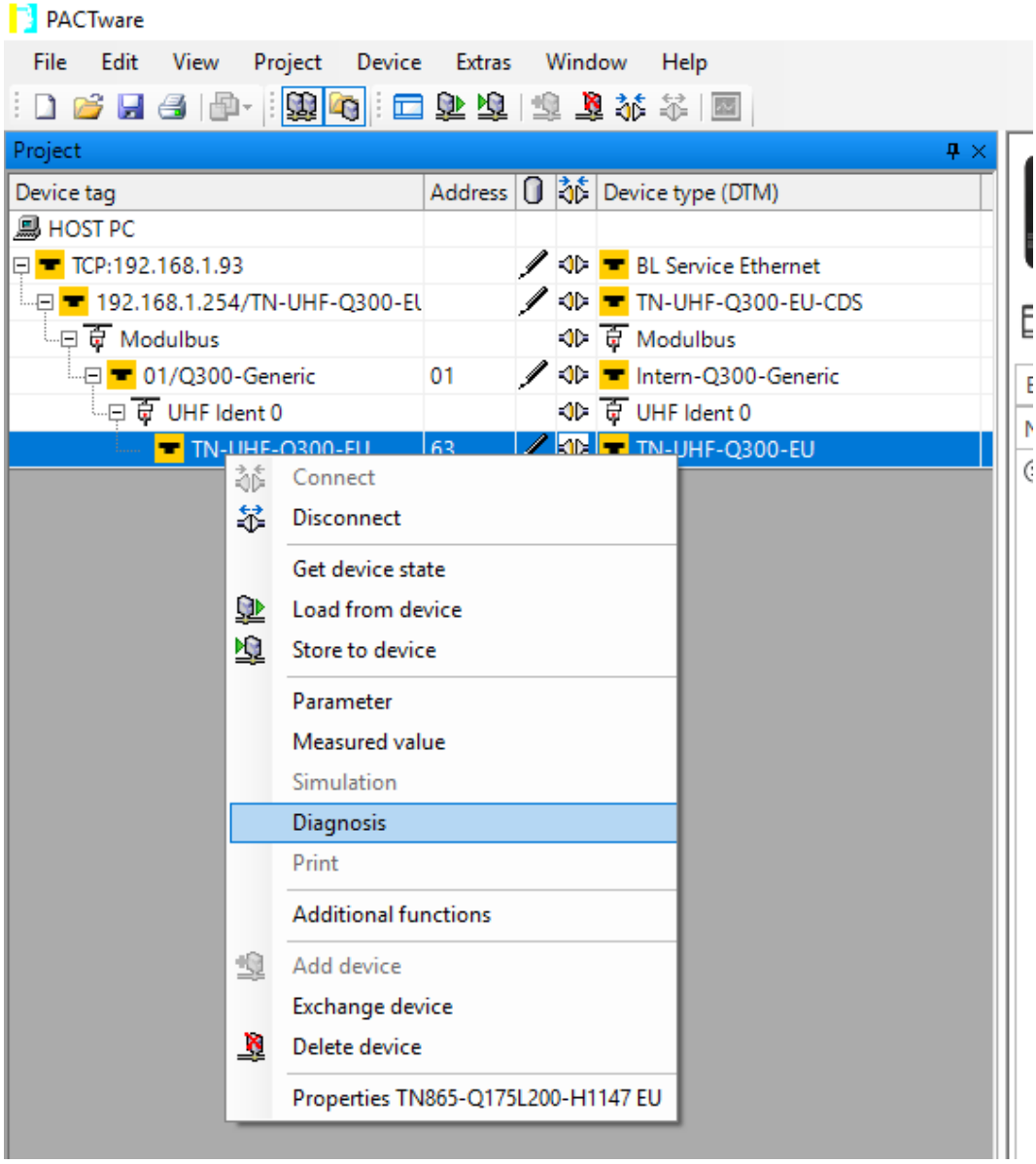


Fig. 56: Project tree – starting diagnosis

Hans Turck GmbH & Co. KG | T +49 208 4952-0 | F +49 208 4952-264 | more@turck.com | www.turck.com

➪ The diagnosis window opens in the DTM.

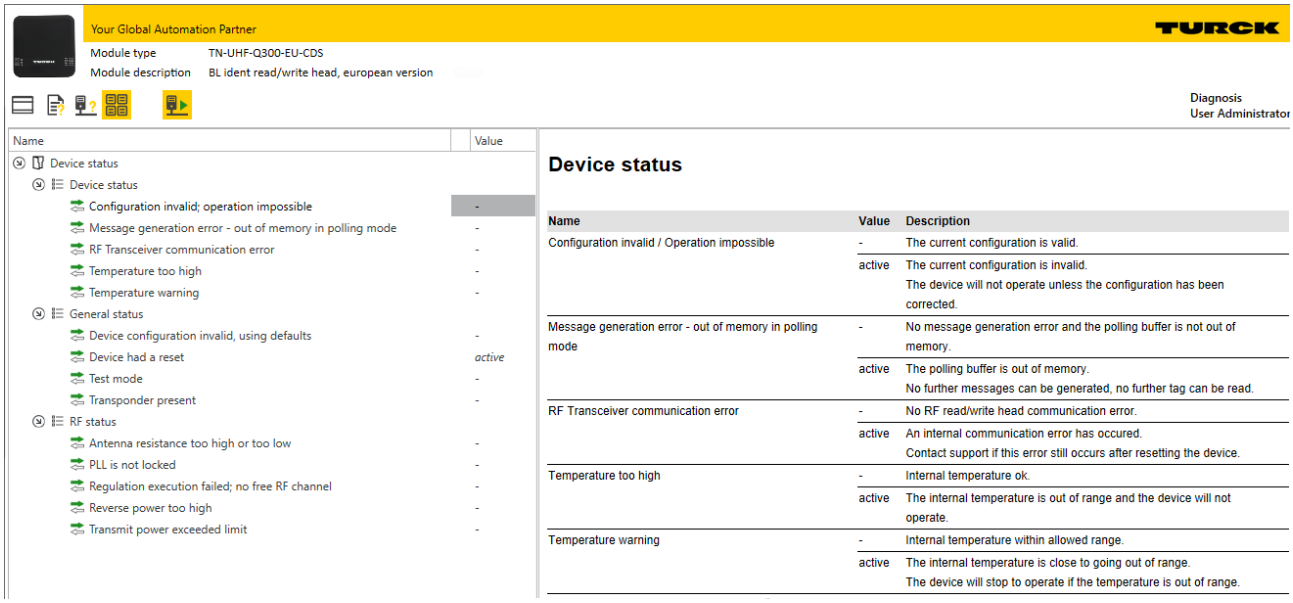


Fig. 57: DTM – diagnosis

Rectifying error messages:

- ▶ Click the **Reset read/write head** button in the RFID Test main menu.
- ▶ Select **Reset the read/write head** in the drop-down menu.

➪ The read/write head is reset.

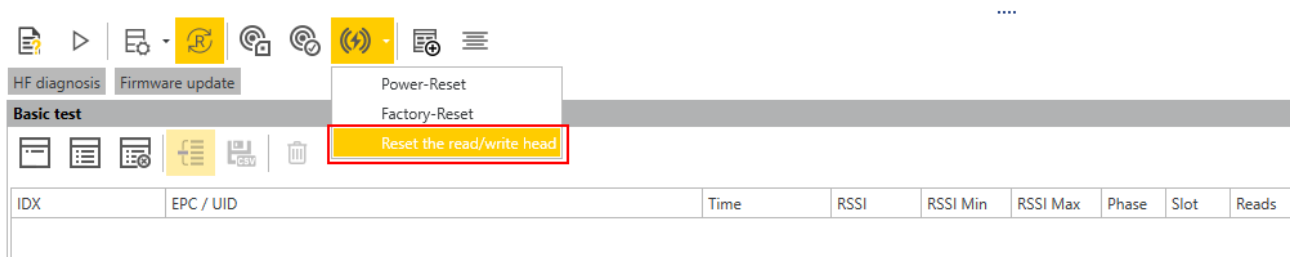


Fig. 58: DTM – Reset the read/write head

10 Maintenance

Ensure that the plug connections and cables are always in good condition.

The devices are maintenance-free, clean dry if required.

10.1 Carrying out a device update

The system image in the default configuration comprises an update application by which XLDR, Eboot, MAC config. and the operating system can be updated.

- XLDR update: Use `update x xldrnand.1block` from the command line of the device.
- Eboot update: Use `update e eboot.nb0` from the command line of the device.
- Updating the system image: Using `update i NK.nb0` from the command line of the device.
- Updating the MAC configuration:
  Use `update c bootmac_WINCE_Update_NAND_AUTOBOOT.nb0` from the command line of the device.

The Windows (PC) application **EbootCFG.exe** is provided for creating the file for the update of the MAC configuration.

Example:

```
EbootCFG.exe 11:22:33:44:55:66
```

The application outputs two files:

- **Bootmac_initial.nb0**: Use only for the initial startup via external flashing

NOTE
Eboot uses Ethernet autostart when writing **Bootmac_initial.nb0**
Operating system does not start automatically from the Flash.

- ▶ Only use the file for the initial startup.

- **Bootmac_WINCE_Update_NAND_AUTOBOOT.nb0**: Provided for use with the update application on the device; also contains the autostart settings for NAND

10.2 Carrying out a UHF update via the DTM

The firmware for the UHF functions of the device can be updated via FDT/DTM. The PACTware FDT frame application, the DTM for the device and the latest firmware can be downloaded free of charge from www.turck.com.

NOTICE
Interruption of the power supply during the firmware update
Risk of device damage due to faulty firmware update
- ▶ Do not interrupt the power supply during the firmware update.
- ▶ During the firmware update do not reset the power supply.

Example: Updating the firmware with the PACTware FDT frame application

- ▶ Launch PACTware.
- ▶ Right-click HOST PC → **Add device**.

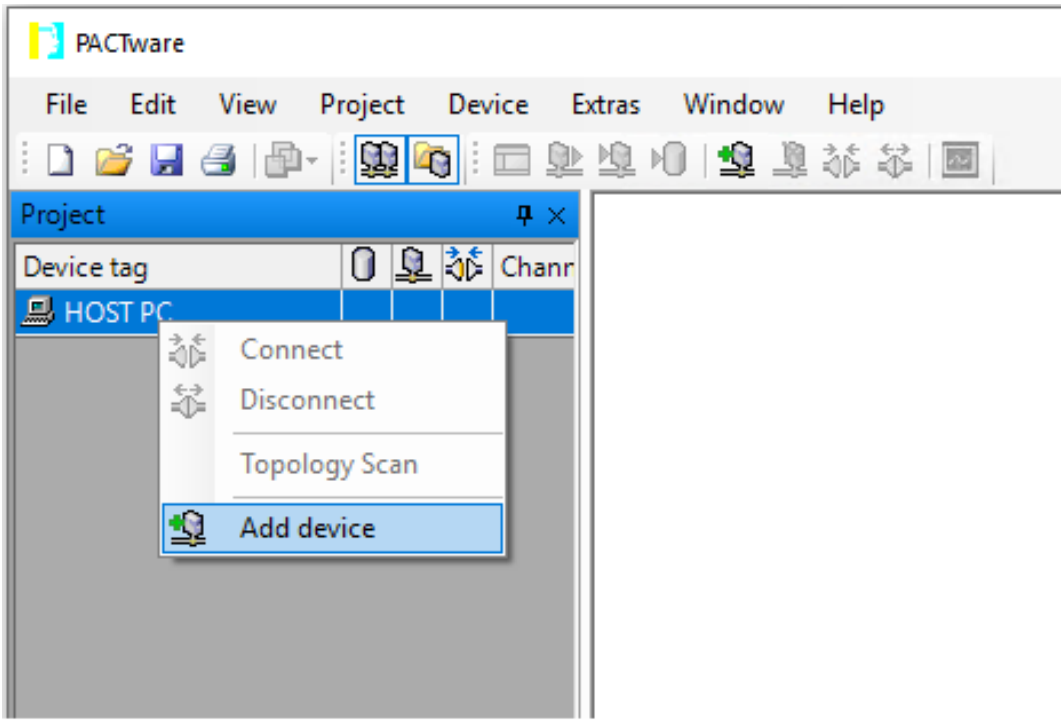


Fig. 59: Adding a device in PACTware

▶ Select **RS485 RFID** and confirm with **OK**.

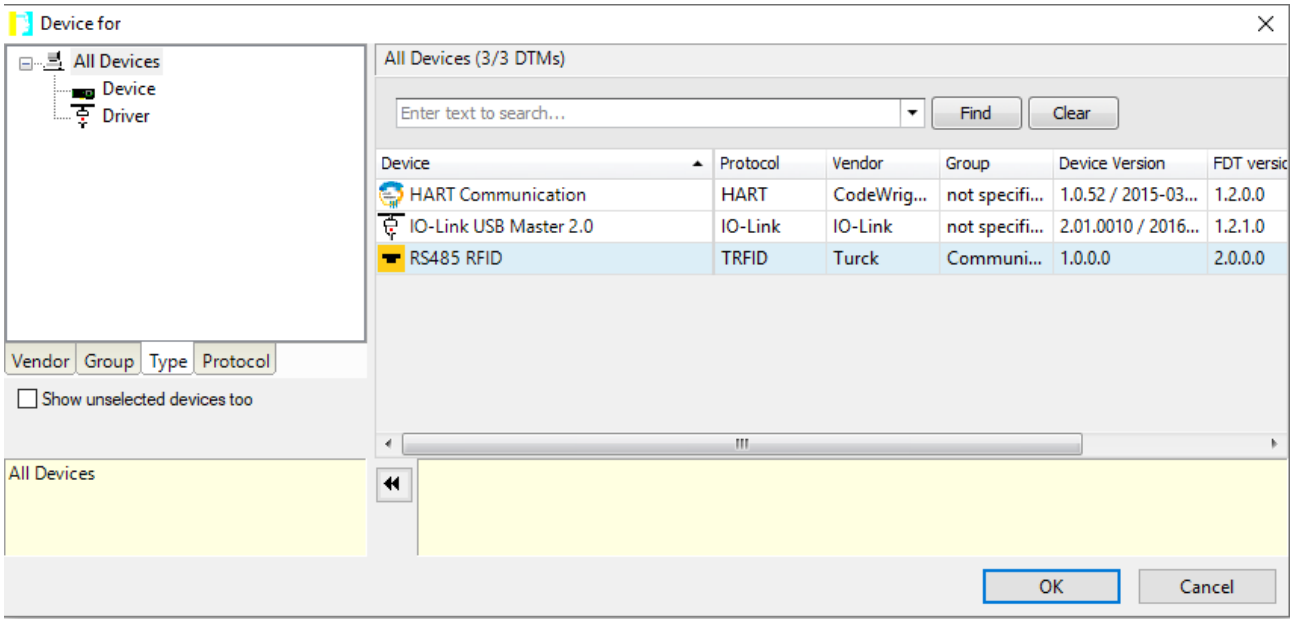


Fig. 60: Selecting the RS485 interface

- Right-click **RS485 RFID**.
- Click **Add device** in the context menu.

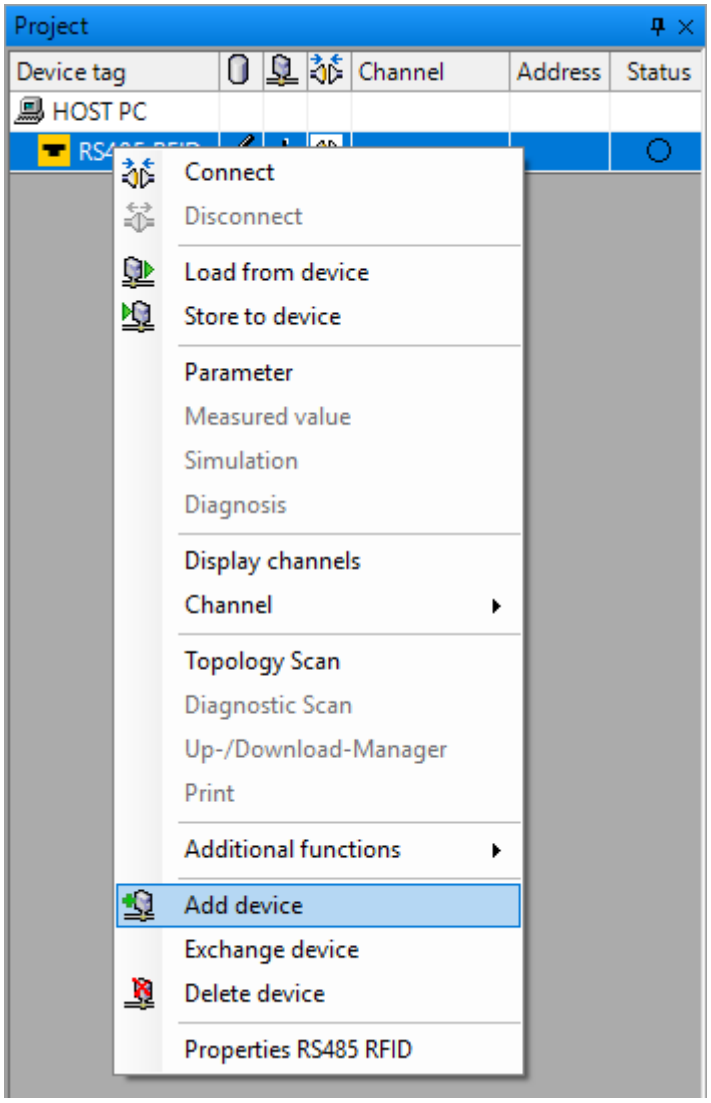


Fig. 61: Adding a read/write head

- Select the required read/write head.

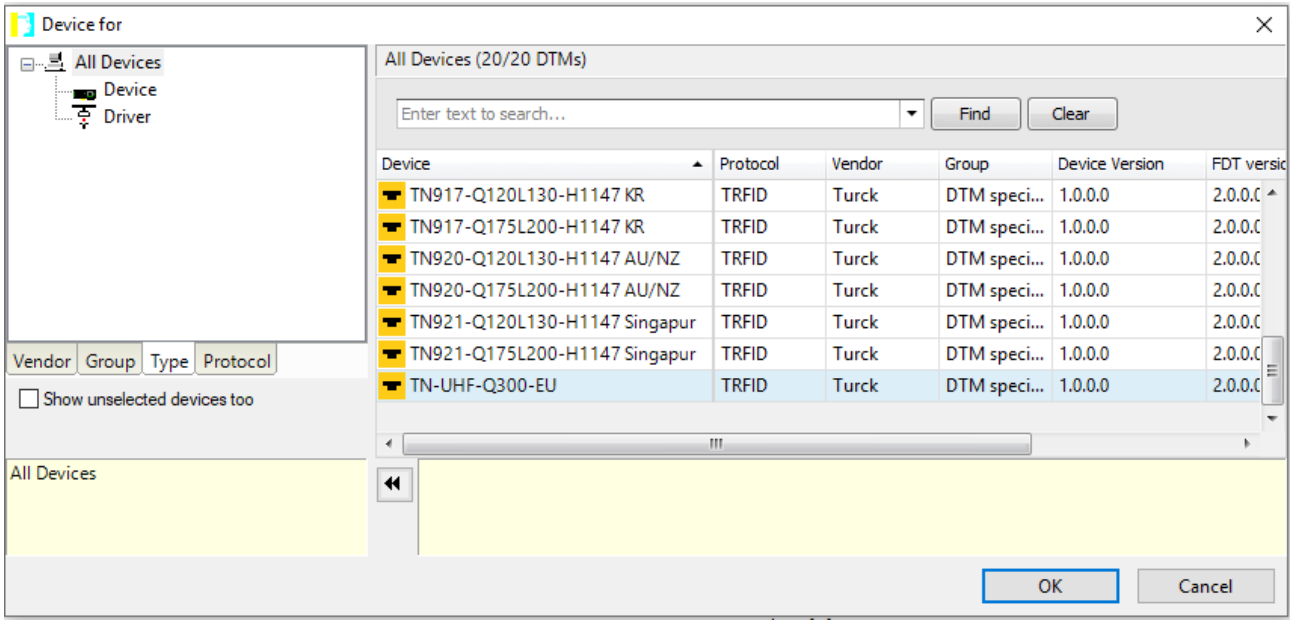


Fig. 62: Example – selecting a read/write head

- Right-click **TN-UHF-Q300-EU**.
- Establish connection to the host PC: Click **Establish connection** in the context menu.

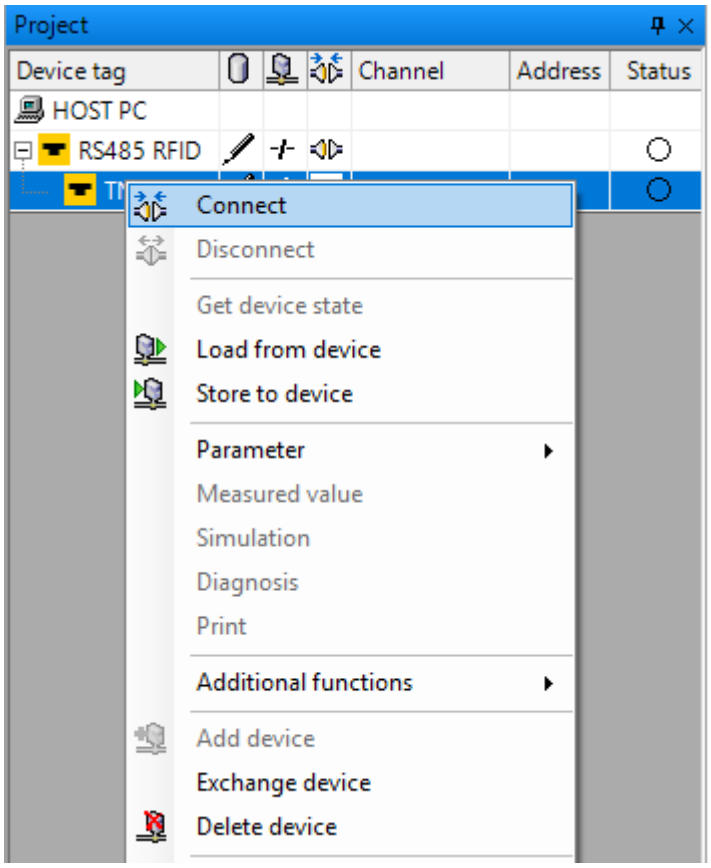


Fig. 63: Establishing a connection

- Right-click **TN-UHF-Q300-EU**.
- In the context menu choose **Other Functions** → **RFID Test**.

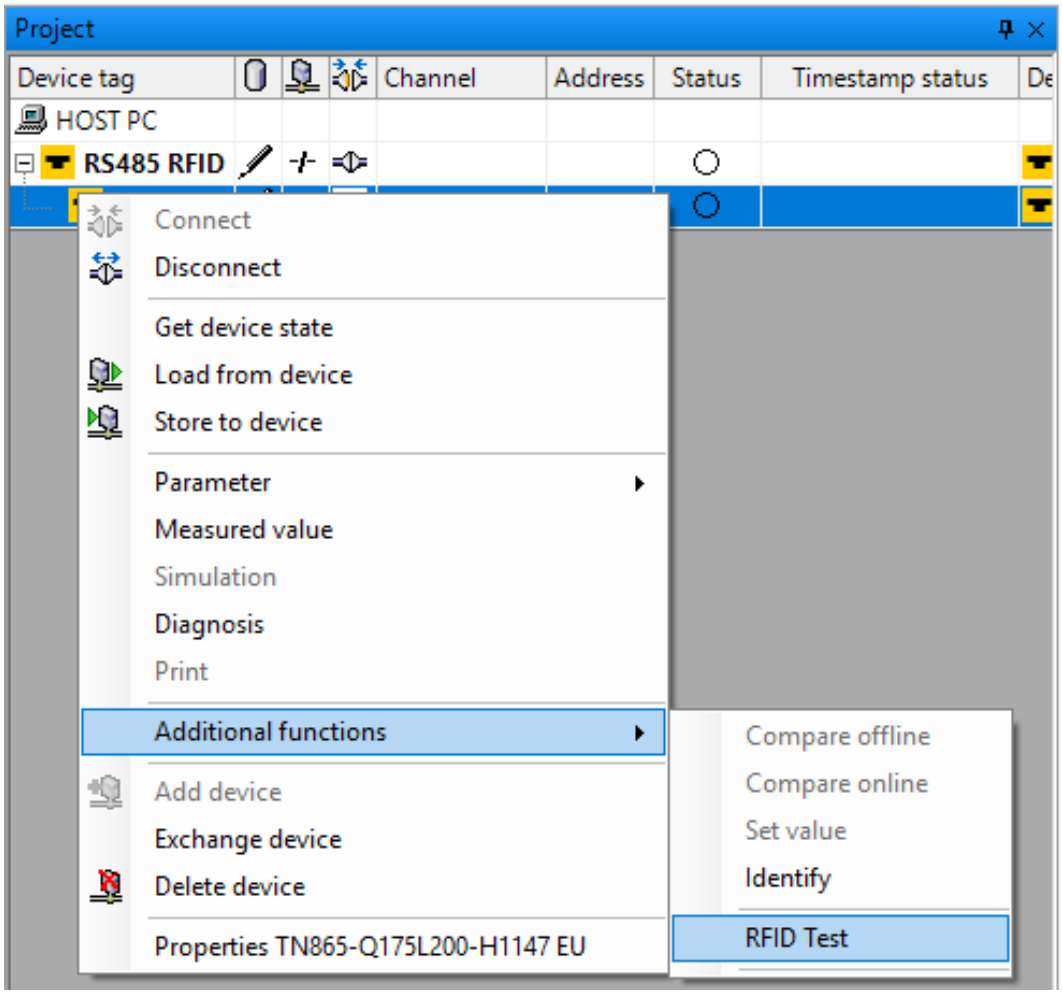


Fig. 64: Starting the RFID test

▶ Click **Firmware Update**.

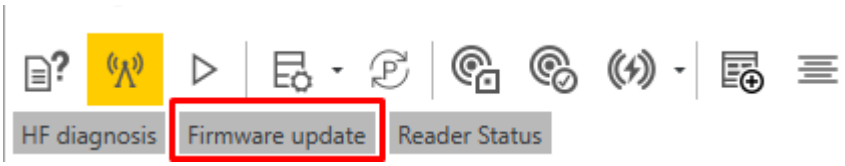


Fig. 65: Firmware Update – Opening the menu

▶ Click the **Select BL File** button.

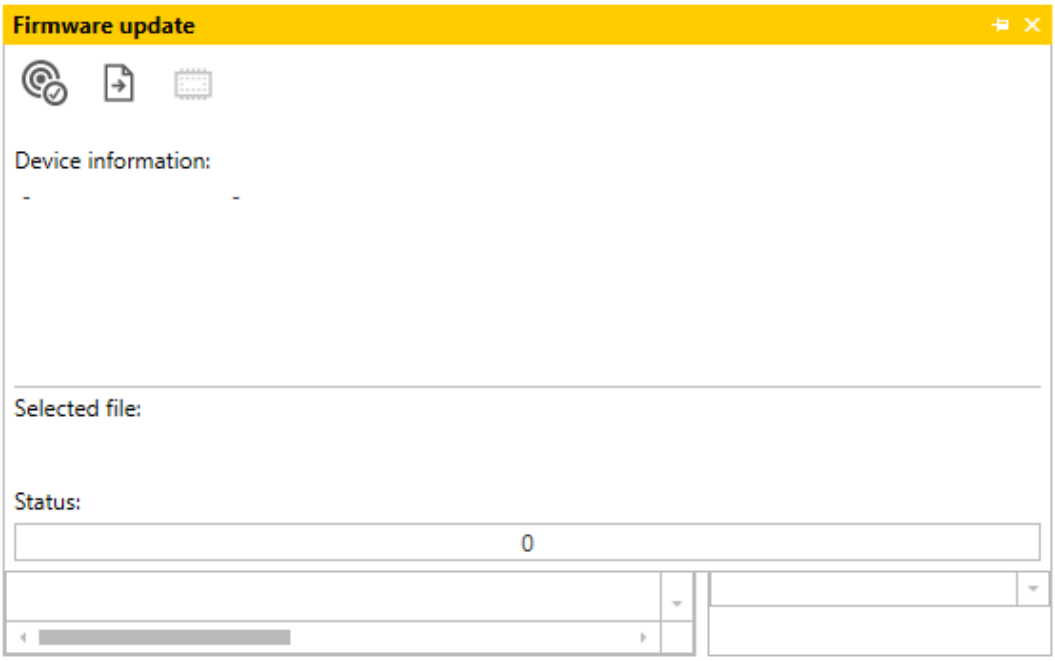


Fig. 66: Button – selecting a BL file

Hans Turck GmbH & Co. KG | T +49 208 4952-0 | F +49 208 4952-264 | more@turck.com | www.turck.com

▶ Select the update file from the local memory location.

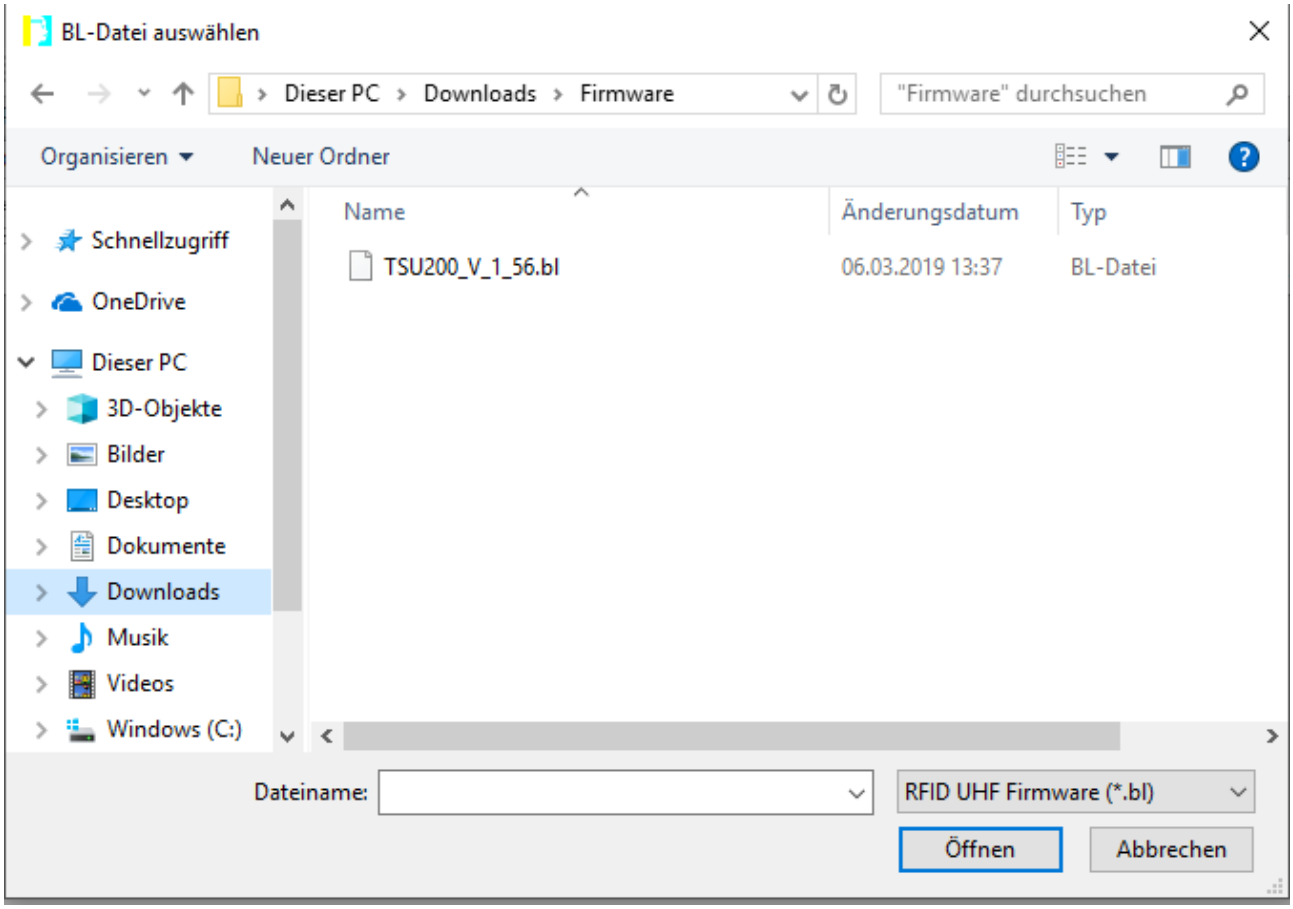


Fig. 67: Local memory location of the update file

▶ Start the update: Click **Firmware update**.

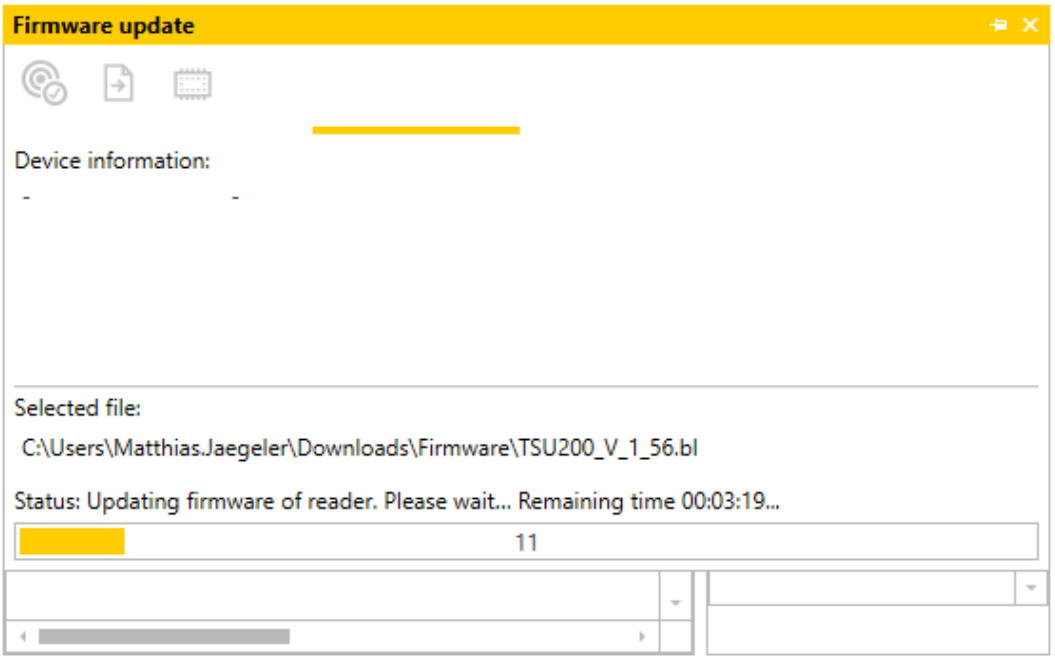


Fig. 68: Starting the firmware update

A completed firmware update is indicated by a status message and a yellow progress bar in the DTM.

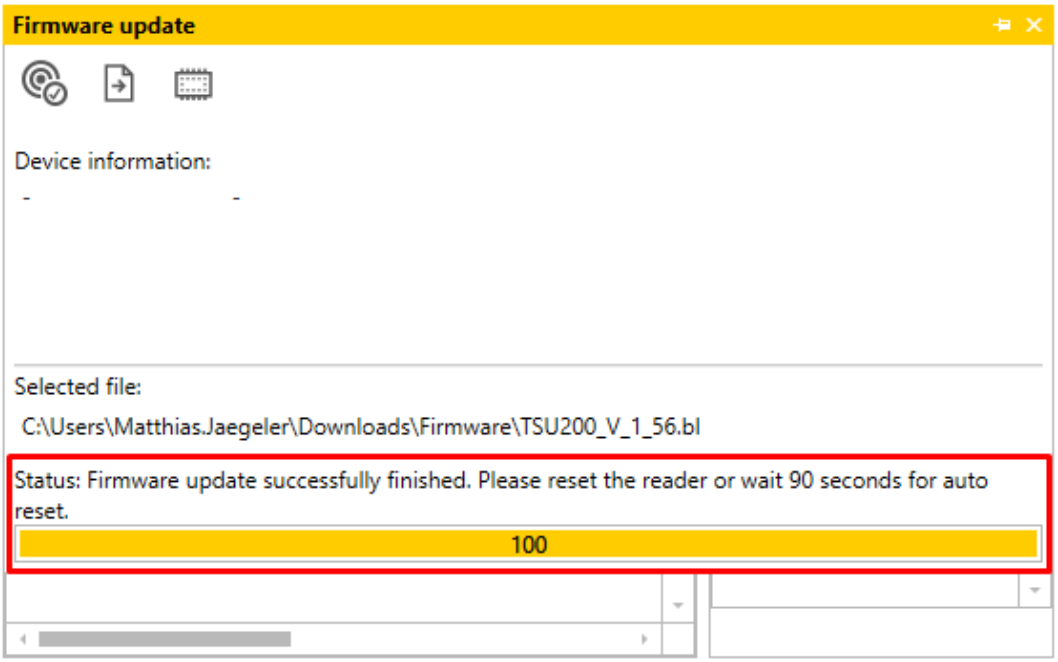


Fig. 69: Successful firmware update

▶ Reset the read/write head to complete the firmware update.

Hans Turck GmbH & Co. KG | T +49 208 4952-0 | F +49 208 4952-264 | more@turck.com | www.turck.com

11 Repair

The device must not be repaired by the user. The device must be decommissioned if it is faulty. Observe our return acceptance conditions when returning the device to Turck.

11.1 Returning devices

Returns to Turck can only be accepted if the device has been equipped with a Decontamination declaration enclosed. The decontamination declaration can be downloaded from
https://www.turck.de/en/retoure-service-6079.php
and must be completely filled in, and affixed securely and weather-proof to the outside of the packaging.

12 Disposal

The devices must be disposed of correctly and must not be included in general household garbage.

13 Technical Data

Technical data	
Electrical data	
Operating voltage	18…30 VDC
DC rated operational current	≤ 1000 mA
Data transmission	Electromagnetic AC field
Wireless communication and protocol standards	ISO 18000-6C EN 302208 EPCglobal Gen 2
Antenna polarization	Circular/linear, adjustable
Antenna half power beam width	65°
Output function	Read/write
Mechanical data	
Mounting condition	Non-flush
Ambient temperature	-20…+50 °C
Dimensions	300 × 300 × 61.7 mm
Housing material	Aluminum, AL, silver
Material of active face	Fiber glass reinforced polyamide, PA6-GF30, black
Vibration resistance	55 Hz (1 mm)
Shock resistance	30 g (11 ms)
Type of protection	IP67
No. of channels	4
Electrical connection	RP-TNC
Input impedance	50 Ω
System description	
Processor	ARM Cortex A8, 32-bit, 800 MHz
ROM memory	512 MB Flash
RAM memory	512 MB DDR3
System data	
Ethernet transfer rate	10 Mbit/s / 100 Mbit/s
Ethernet connection technology	1 × M12, 4-pin, D-coded
Digital inputs	
No. of channels	4
Connection technology of inputs	M12, 5-pin
Input type	PNP
Switch threshold	EN 61131-2 Type 3, PNP
Signal voltage Low signal	< 5 V
Signal voltage High signal	> 11 V
Signal current Low signal	<1.5 mA
Signal current High signal	> 2 mA
Type of input diagnostics	Channel diagnostics

Technical data	
Digital outputs	
No. of channels	4
Connection technology of outputs	M12, 5-pin
Output type	PNP
Type of output diagnostics	Channel diagnostics

TURCK

Over 30 subsidiaries and over
60 representations worldwide!

100003076 | 2020/10

www.turck.com