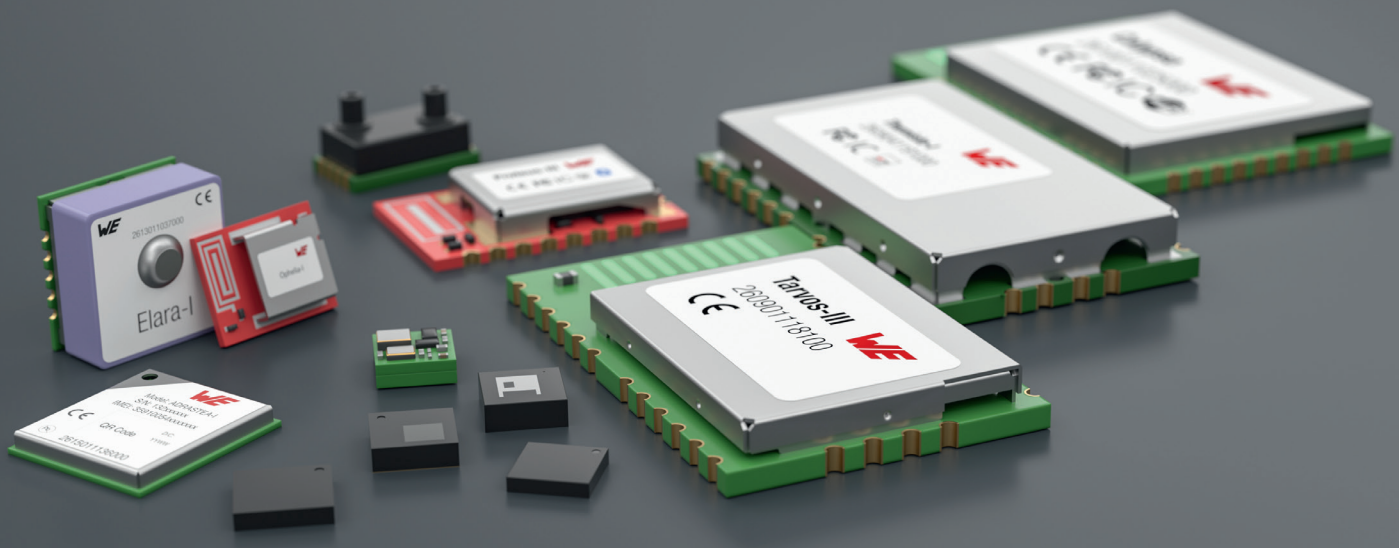


ANR014

Connect a smart phone to a Proteus Bluetooth LE module

Version 1.5

September 9, 2024

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

Revision history

Manual version	Notes	Date
1.0	• Initial version	July 2019
1.1	• Added description for Proteus-III • Updated address of Division Wireless Connectivity & Sensors location	January 2020
1.2	• Added example for connection setup using the Proteus Connect App • Added information on Proteus-III-SPI and the mini EV-Board	March 2021
1.3	• Updated `Important notes`, meta data and document style	July 2023
1.4	• Updated the description using WE UART Terminal [1] PC tool • Updated screen shots of Proteus Connect app • Updated links to source code of mobile apps	October 2023
1.5	• Corrected figure 2 • Proteus Connect app has new name "WE Bluetooth LE Terminal" app	September 2024

Abbreviations

Abbreviation	Name	Description
BTMAC		Bluetooth® conform MAC address of the module used on the RF-interface.
CS	Checksum	Byte wise XOR combination of the preceding fields.
Central		Bluetooth® LE device role that scans for advertising packets & initiates connections, e.g. smart phone.
DTM	Direct test mode	Mode to test Bluetooth® specific RF settings.
GAP	Generic Access Profile	The GAP provides a basic level of functionality that all Bluetooth® devices must implement.
I/O	Input/output	Pinout description.
LPM	Low power mode	Mode for efficient power consumption.
MAC		MAC address of the module.
MTU	Maximum transmission unit	Maximum packet size of the Bluetooth® connection.
Payload		The intended message in a frame / package.
Peripheral		Bluetooth® Low Energy device role that provides services & advertises, e.g. sensor or our Proteus module.
RF	Radio frequency	Describes wireless transmission.
RSSI	Receive Signal Strength Indicator	The RSSI indicates the strength of the RF signal. Its value is always printed in two's complement notation.
SoC		System on Chip.
Soft device		Operating system used by the nRF52 chip.
SPI	Serial Peripheral Interface	Allows the serial communication with the module.
UART	Universal Asynchronous Receiver Transmitter	Allows the serial communication with the module.
[HEX] 0xhh	Hexadecimal	All numbers beginning with 0x are hexadecimal numbers. All other numbers are decimal, unless stated otherwise.

Contents

1 Introduction

The Proteus series is a radio module series that is based on Nordic Semiconductors SoC which presents various Bluetooth® LE and low power features.
By default in the so called command mode, a radio module of the Proteus series can be controlled and configured by the host using predefined commands sent via the UART interface.
This application note describes how to setup a connection between a Bluetooth® LE enabled smart device, e.g. smart phone or tablet, to a Proteus module and how to interchange data in **command mode**.
These steps are described with help of the nRF Connect App [2, 3] which is an open source App providing standard Bluetooth® LE functions for iOS as well as for Android devices. The same is repeated using the WE Bluetooth LE Terminal App [4, 5], which is provided by Würth Elektronik eiSos.

There is a second operation mode, that offers a transparent UART interface to transmit data without any overhead on the UART. For more information concerning this mode, please refer to the application node ANR004_Proteus_Peripheral_Only_Mode [6].

2 Prerequisites

To follow the description in this application note, the following prerequisites may be helpful:

- A Bluetooth® LE enabled smart phone including a suitable App, for example
  - the **WE Bluetooth LE Terminal** App for Android [4, 7] or iOS [5, 7]
  - the Nordic Semiconductor **nRF Connect** App for Android [2] or iOS [3]
- A Proteus EV-Board in factory state, for example
  - the Proteus-I EV-Board with jumpers set as specified in figure 1. Other jumpers not set.
  - the Proteus-II EV-Board with jumpers set as specified in figure 1. Other jumpers not set.
  - the Proteus-III EV-Board with jumpers set as specified in figure 2. Other jumpers not set.
  - the Proteus-III-SPI mini EV-Board with jumpers set as specified in figure 3. Other jumpers not set.

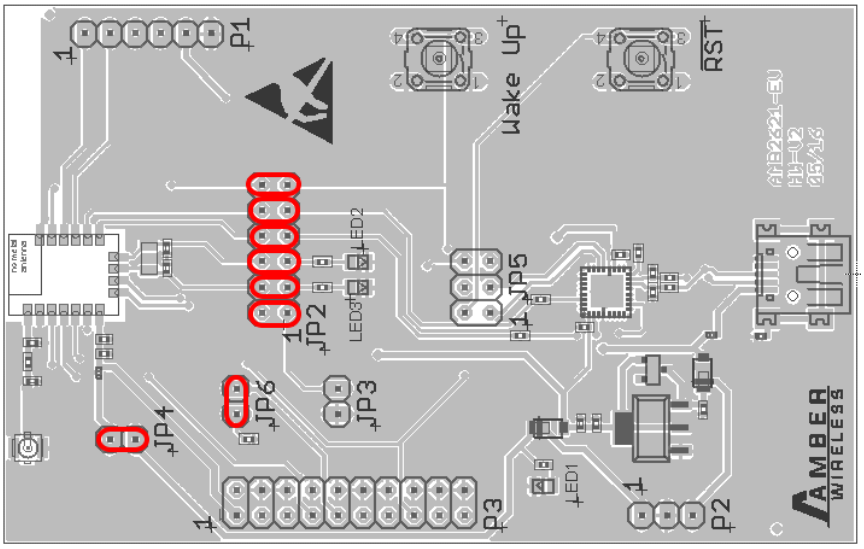


Figure 1: Default jumper placement of the Proteus-I and Proteus-II EV-Board. Red means "jumper must be set".

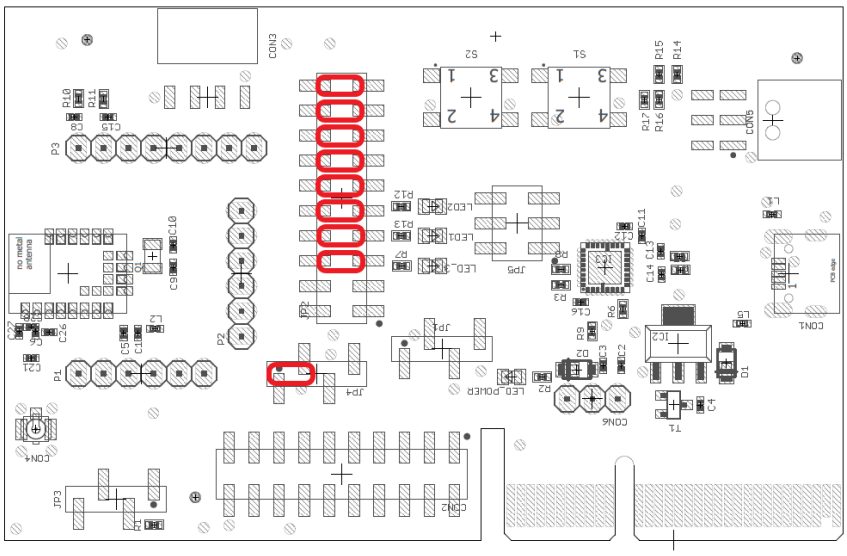


Figure 2: Default jumper placement of the Proteus-III EV-Board. Red means "jumper must be set".

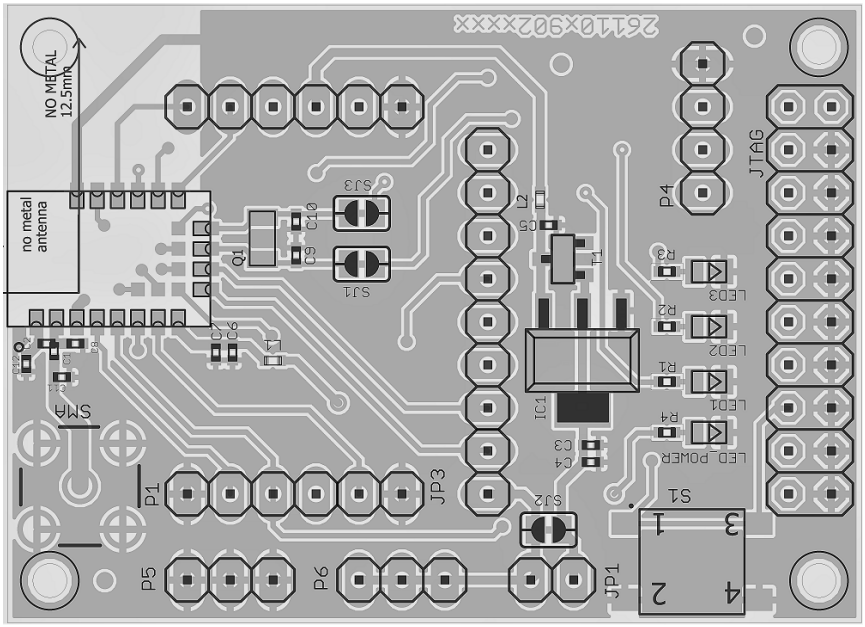


Figure 3: Default jumper placement of the Proteus-III-SPI mini EV-Board.

The complete description of Proteus modules can be found in the respective radio module manual and application notes. This may be helpful to understand the background of the following quick start:

- Proteus-I
  - Proteus-I user manual [8]
  - Proteus-I advanced user guide ANR002_Proteus-I_Advanced_Developer_Guide [9]
- Proteus-II

  - Proteus-II user manual [10]
  - Proteus-II advanced user guide ANR005_Proteus-II_Advanced_Developer_Guide [11]
- Proteus-III
  - Proteus-III user manual [12]
  - Proteus-III advanced user guide ANR009_Proteus-III_Advanced_Developer_Guide [13]
- Proteus-III-SPI
  - Proteus-III-SPI user manual [14]
  - Proteus-III advanced user guide ANR009_Proteus-III_Advanced_Developer_Guide [13]

3 Basics

The setup of a Bluetooth® LE connection to a Proteus radio module contains several steps:

1. Physical connection establishment
   First of all, a physical connection has to be established. Therefore, a central device (usually smart phone) has to connect to the Proteus module which runs as peripheral.

2. Optional: Pairing process
   Second, the pairing process is run that consists of the authentication and exchange of encryption information. The central device must request at least the same security level to access the characteristics of the peripheral (Proteus module).
   - In factory state, the Proteus module has no security enabled and this step can be neglected.
   - Security can be enabled by modifying the user setting `RF_SecFlags`.

If the security level of the central device is lower than the security mode of the peripheral, the central cannot access the peripheral's characteristics. In this case, the central sends the notification enable message, which is ignored by the peripheral. Thus, the central signalizes an open connection, although it does not have access to the peripheral and thus data cannot be transmitted! In some cases, the peripheral may also disconnect to avoid to be blocked by attackers.

3. Optional: Exchange of the maximum transmission unit (MTU)
   Next, the maximum transmission unit can be increased to allow the transmission of larger data packets. The Proteus module allows an MTU of up to 247 bytes, which results in a payload of up to 243 bytes. This step is optional. Not selecting a higher MTU will use the Bluetooth® 4.0 default MTU which results in 19 bytes payload for the user but will be compatible to pre Bluetooth® 4.2 devices.

4. Discover the characteristics of the Proteus module SPP-like profile
   Afterwards, the characteristics offered by the Proteus module have to be discovered by the central. This is needed to share the information how data can be transmitted.

5. Notification enable
   To finalize the connection setup, the notification enabled message has to be sent. With this feature, the peripheral device lets the central know, when there is new data, which is important for bidirectional data transmission. After this step, the channel is open and data transmission can start.

For the description, we assume that a smart phone is the initiator of the connection. Thus, it acts as central and the Proteus module acts as peripheral in figure 4.

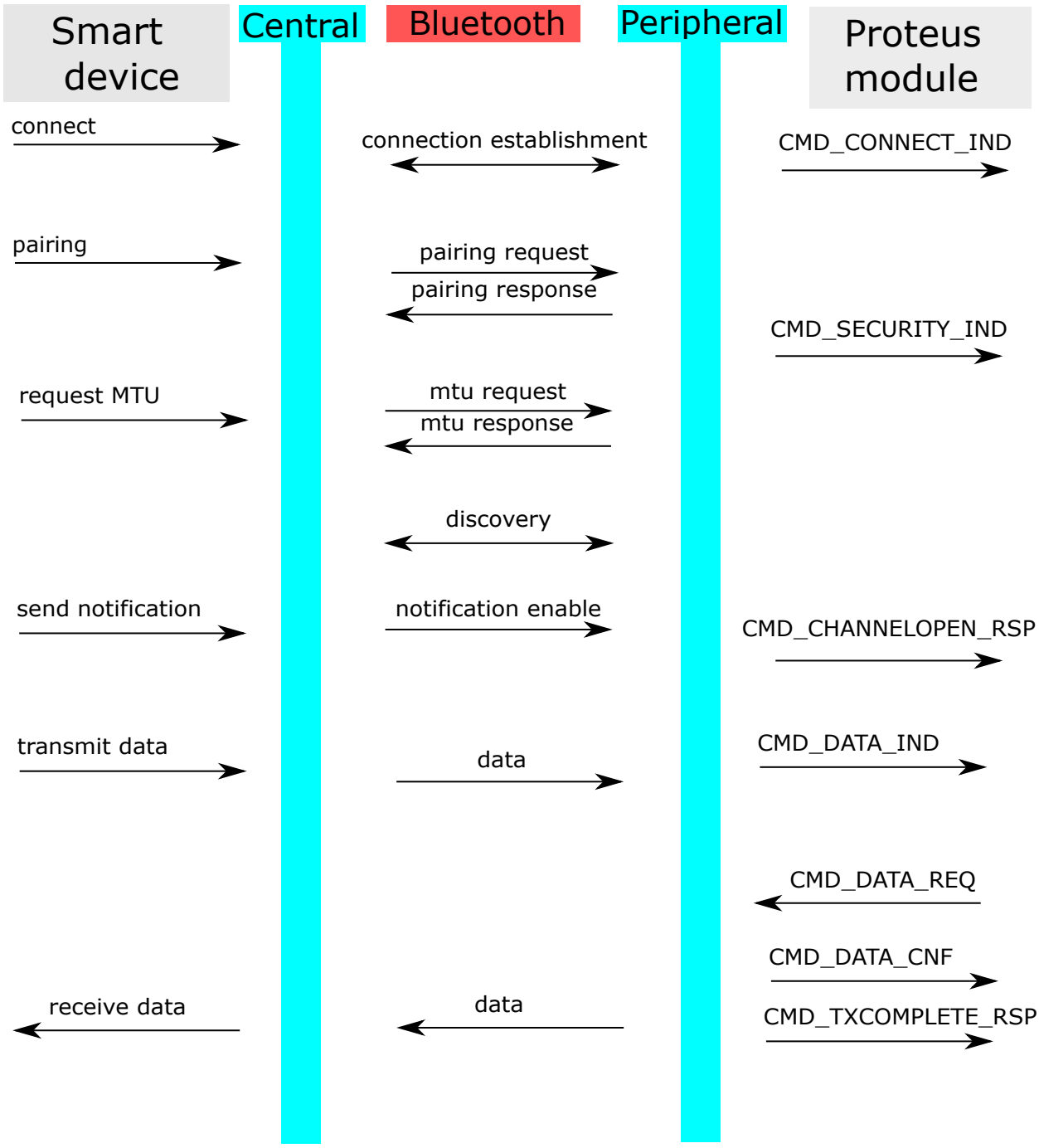


Figure 4: Steps for the connection setup

4 Quick start

The following description demonstrates how to setup a connection with a smart phone to a Proteus radio module. The smart phone acts as central device.
In the next chapter the **WE Bluetooth LE Terminal** App is used. Then the same is done using the Nordic Semiconductor **nRF Connect** App.

4.1 WE Bluetooth LE Terminal App

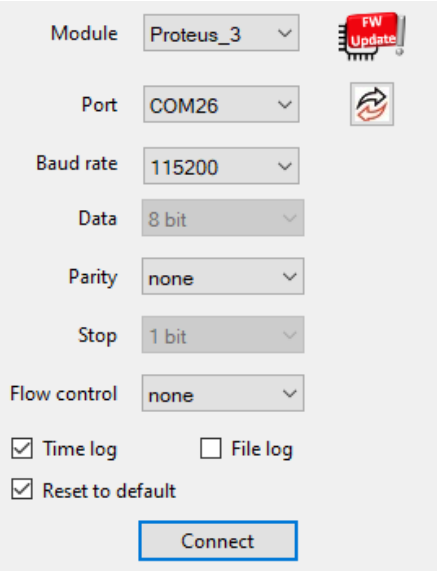
This chapter describes how to setup a connection to the Proteus module in command mode, when a smart phone and the WE Bluetooth LE Terminal App are used.

The WE Bluetooth LE Terminal App for iOS and Android is provided by Würth Elektronik eiSos as executable [4, 5] as well as source code [7].

Please perform the following steps:

1. Connect the Proteus EV-Board to a host.
   In this application note, we assume that a Windows PC and the PC tool WE UART Terminal [1] is used. For Proteus-I, -II and -III EV-Board this can be simply achieved by using a simple USB cable to connect it to a PC.

2. Start the WE UART Terminal, select the right module type and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing the "Connect" button.

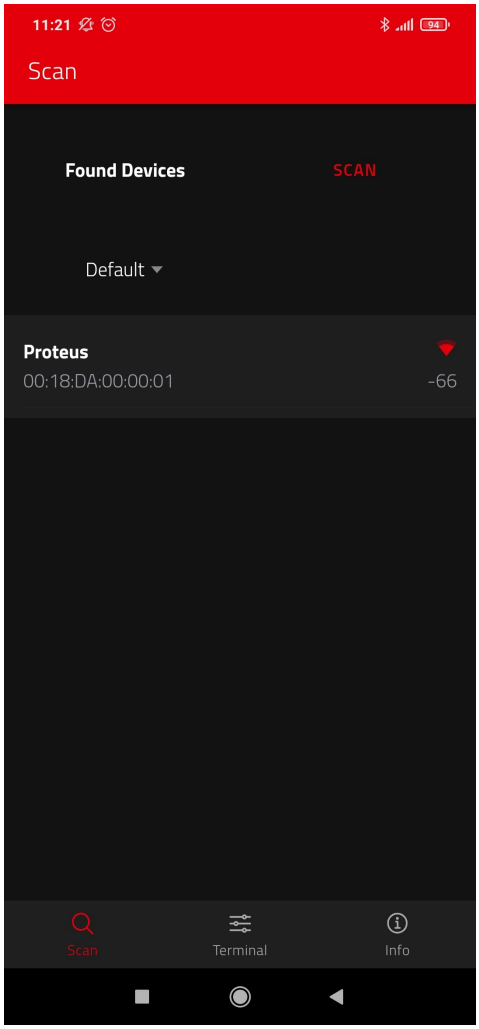


3. Press the reset button on the Proteus EV-Board. The Proteus module outputs a `CMD_GETSTATE_CNF` message to indicate that it is ready for operation.

```
                                    [10:22:08.296]
                                    CMD_GETSTATE_CNF:
                                    02 41 0200 0101 41
```

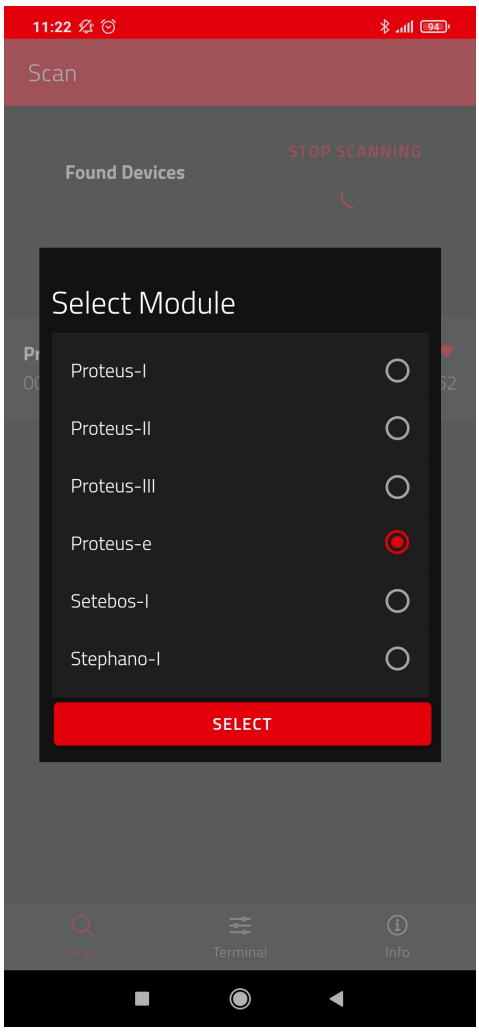
4. By default, the module is advertising. Thus, one LED of the Proteus EV-Board is blinking. Start your smart phone, enable the Bluetooth® LE and location feature and open the **WE Bluetooth LE Terminal** App.

5. Press "Scan" to find the module on the radio.



6. When the module appears in the scan list, select it.

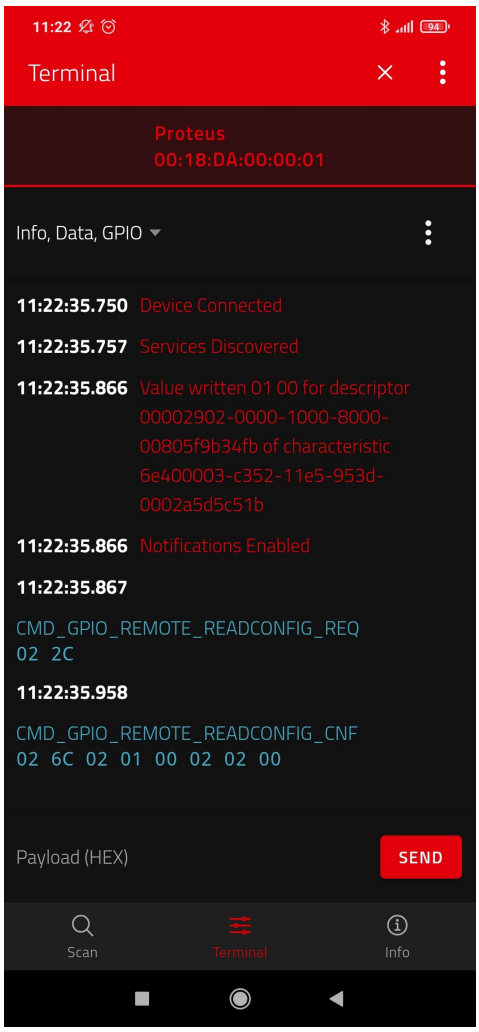
7. A pop-up will come up, where you need to select the current module type.



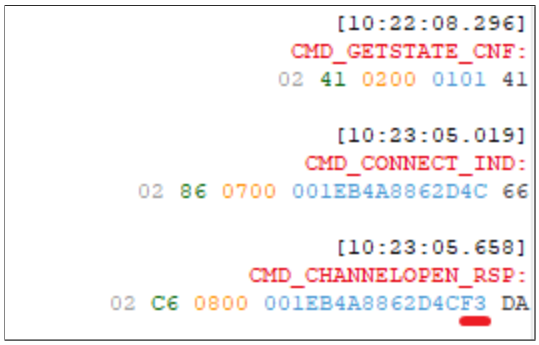
8. As soon as the module has been chosen, the connection setup starts. When the module has received the connection request, it's *LED_1* (*LED_3* on the Proteus-EV) will constantly light up.

9. Optional pairing: In case a security mode has been configured before, the smart phone requests the user for pairing actions. In case of the static passkey authentication, the Proteus requests to enter the static passkey. The default passkey is "123123". The Bluetooth® coupling requirement pop-up is shown on your smart phone.
If the bonding feature is enabled in the authentication settings and the bonding information already exists, a re-entering of the passkey is not required when reconnecting.

10. You are authenticated and the *LED_2* (*LED_2* on the Proteus-EV) is turned on. Now data can be transmitted in both directions.

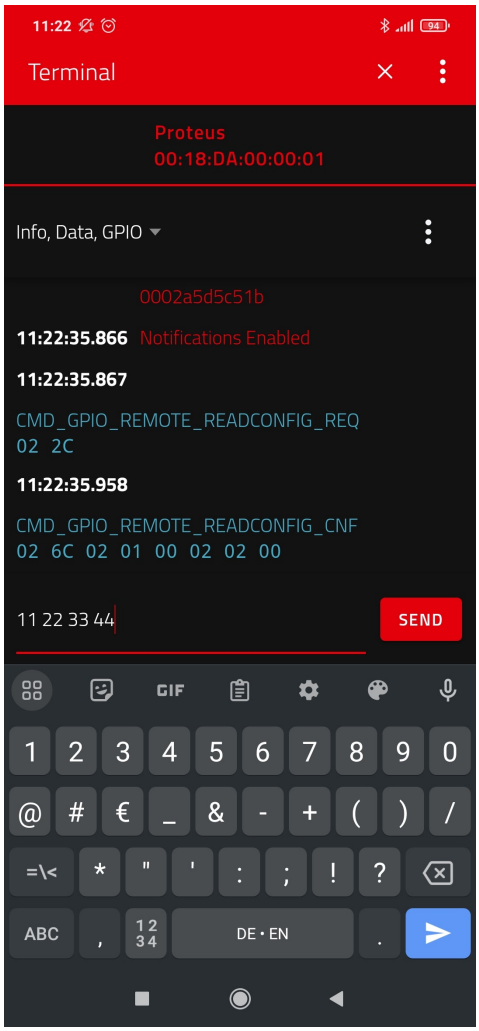


11. On the Proteus side, the radio module has sent the corresponding `CMD_CONNECT_IND` and `CMD_CHANNELOPEN_RSP` in between. These messages indicate that a connection has been setup and a link has been opened. The `CMD_CHANNELOPEN_RSP` message contains the MTU (maximum transmission unit) of the current link, which defines the maximum supported packet payload length. In this example it's 0xF3 (243_{dec}) bytes payload per packet.



12. Now, we want to send data from the smart phone to the radio module. To do so, enter your payload (for example 0x11 0x22 0x33 0x44) in the respective field and press "SEND" (see next image). The allowed payload size is dependent on the MTU that was negotiated

in the connection process. The smallest supported MTU for all Bluetooth® 4.0 (or newer) devices results in a max payload of 19 bytes. Android usually allows up to 243 bytes, iOS up to 181 bytes.



13. The payload that has been sent is output by the Proteus module via UART. In the terminal program a `CMD_DATA_IND` message has been received that contains the `BTMAC` of the sending device and the transmitted payload 0x11 0x22 0x33 0x44. The format of the `CMD_DATA_IND` message is as follows:

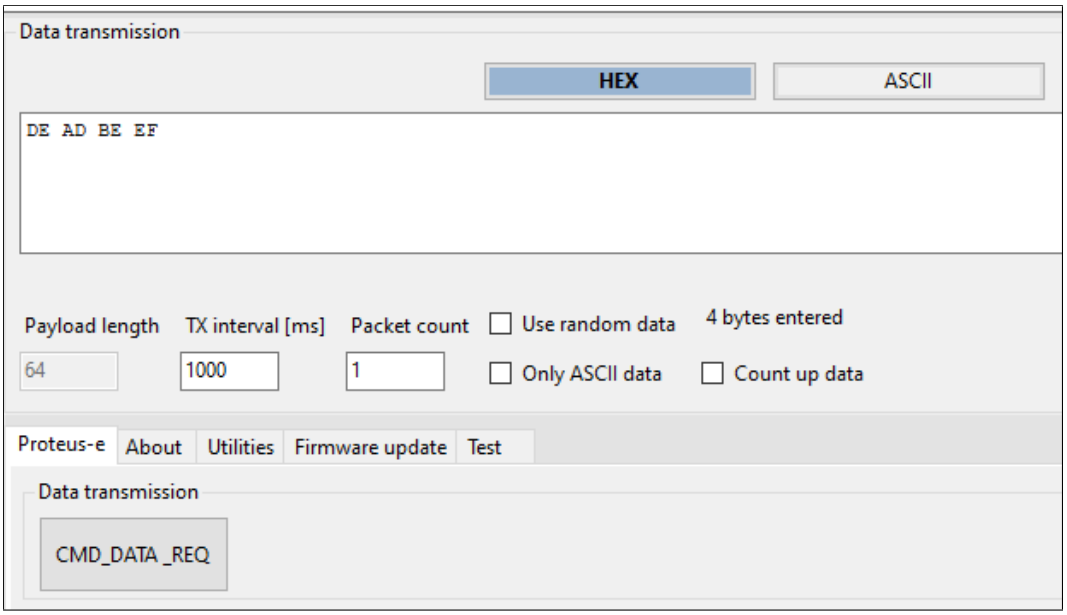
Start signal	Command	Length	BTMAC	RSSI	Payload	CS
0x02	0x84	2 Bytes	6 Bytes	1 Byte	(Length - 7) Bytes	1 Byte
0x02	0x84	0x0B 0x00	0x1E 0xB4 0xA8 0x86 0x2D 0x4C	0XC5	0x11 0x22 0x33 0x44	E9

```
                                    [10:22:08.296]
                                CMD_GETSTATE_CNF:
                              02 41 0200 0101 41

                                    [10:23:05.019]
                                CMD_CONNECT_IND:
                    02 86 0700 001EB4A8862D4C 66

                                    [10:23:05.658]
                              CMD_CHANNELOPEN_RSP:
                  02 C6 0800 001EB4A8862D4CF3 DA

                                    [10:23:20.067]
                                 CMD_DATA_IND:
          02 84 0B00 1EB4A8862D4CC511223344 E9
```

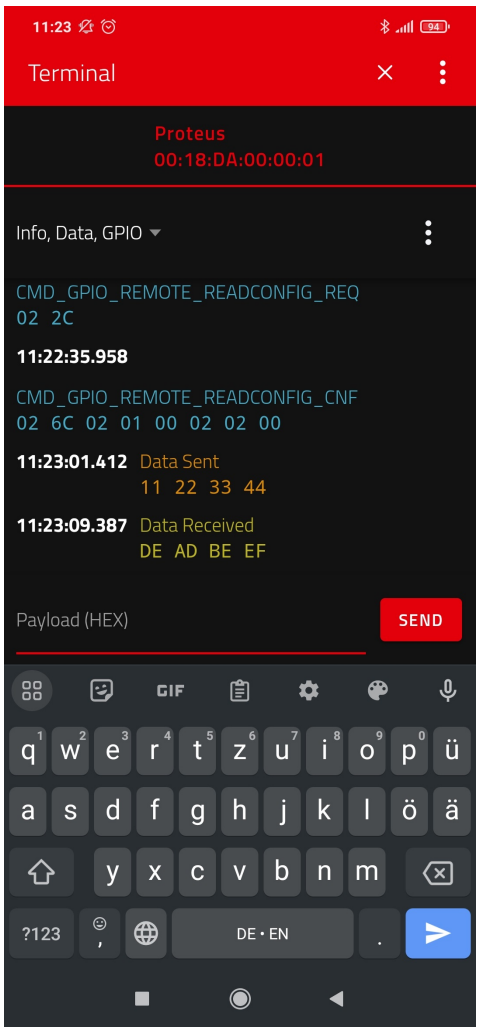
14. To send back data (here we choose 0xDE 0xAD 0xBE 0xEF) to the smart phone a `CMD_DATA_REQ` message must be sent to the module from the host. The format of the `CMD_DATA_REQ` message is as follows, where the check sum (CS) is calculated as XOR of the preceding bytes:

Start signal	Command	Length	Payload	CS
0x02	0x04	2 Bytes	Length Bytes	1 Byte
0x02	0x04	0x04 0x00	0xDE 0xAD 0xBE 0xEF	0x20

To do that in WE UART Terminal, please enter only the payload in the following text field and press the `CMD_DATA_REQ` button. On button press, the remaining command parts are added by the WE UART Terminal.



15. The received data is shown in the status window of the app.



16. When sending the `CMD_DATA_REQ` to the Proteus module, it responds with two different messages. First a `CMD_DATA_CNF` message is returned, as soon as the request was interpreted. Then a `CMD_TXCOMPLETE_RSP` message is returned as soon as the data has been transmitted.

```
[10:24:29.005]
CMD_DATA_REQ:
02 04 0400 DEADBEEF 20

                                                    [10:24:29.018]
                                                    CMD_DATA_CNF:
                                                    02 44 0100 00 47

                                                    [10:24:29.110]
                                                    CMD_TXCOMPLETE_RSP:
                                                    02 C4 0100 00 C7
```

17. To disconnect the smart phone from the Proteus module, press the "X" button in the **WE Bluetooth LE Terminal** App. The Proteus module will output a `CMD_DISCONNECT_IND` message to indicate that the connection has been closed. After disconnecting the Proteus module starts advertising again, such that a new connection can be setup.

```
                                                    [10:24:35.267]
                                                    CMD_DISCONNECT_IND:
                                                    02 87 0100 13 97
```

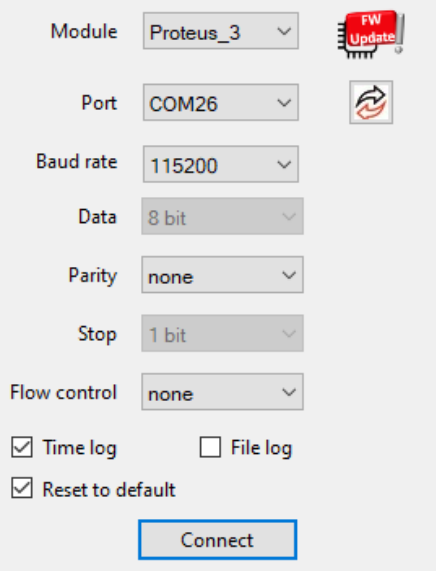
4.2 nRF Connect App

This chapter describes how to setup a connection to the Proteus module in command mode, when a smart phone and the **nRF Connect** App [2, 3] are used. Please perform the following steps:

Android	iOS

- Connect the Proteus EV-Board to a host.
  In this application note, we assume that a Windows PC and the PC tool WE UART Terminal [1] is used. For Proteus-I, -II and -III EV-Board this can be simply achieved by using a simple USB cable to connect it to a PC.

- Start the PC tool, select the right module and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing "Connect".

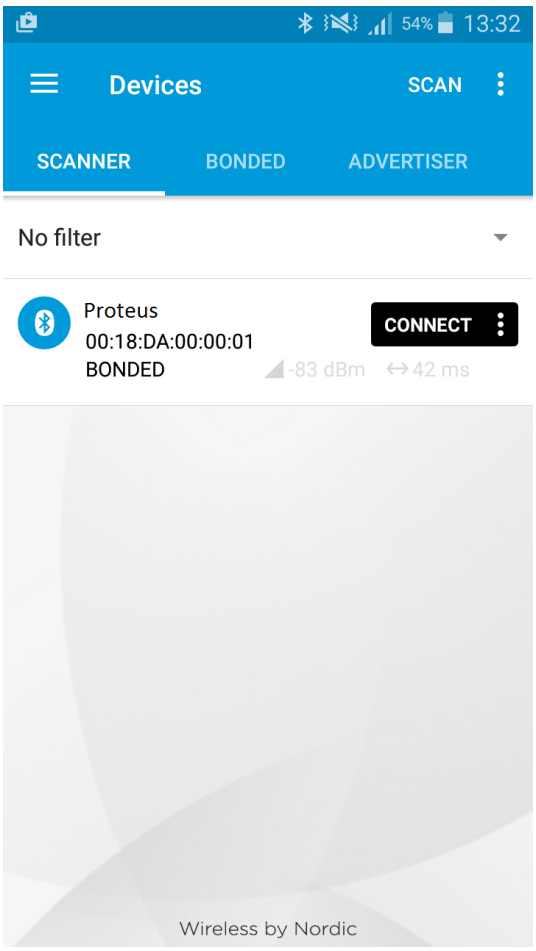


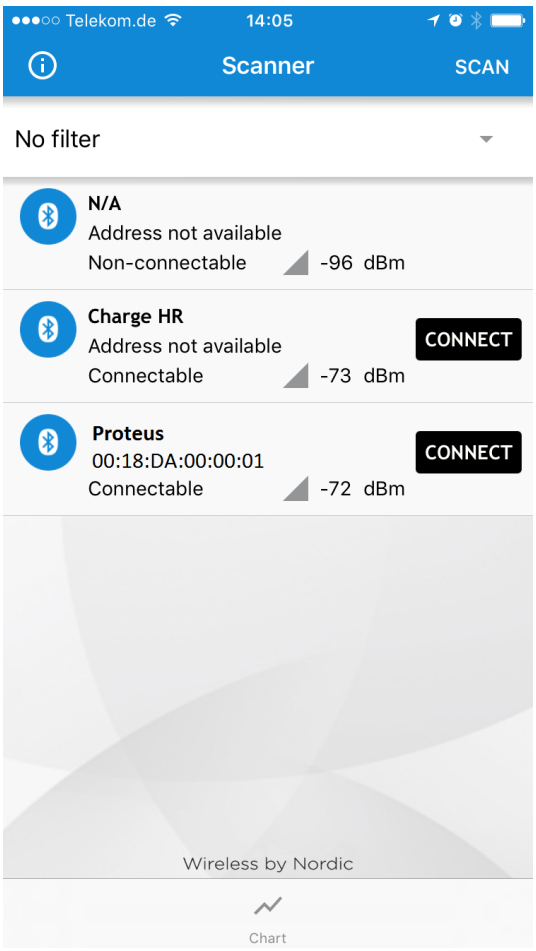
- Press the reset button on the Proteus EV-Board. The Proteus module outputs a `CMD_GETSTATE_CNF` message to indicate that it is ready for operation.

```
[10:22:08.296]
CMD_GETSTATE_CNF:
02 41 0200 0101 41
```

Android	iOS

- Initially, the module is advertising. Thus, one LED of the Proteus EV-Board is blinking.
- Start your smart phone, enable the Bluetooth® LE feature and start the **nRF Connect** App.
- Press "SCAN" to find the module on the radio. In case several Proteus modules are found, the Bluetooth® MAC 0x0018DAxxxxxx can be used to detect the right one. The Bluetooth® MAC consists of the module's serial number, that can be also found on the module label.

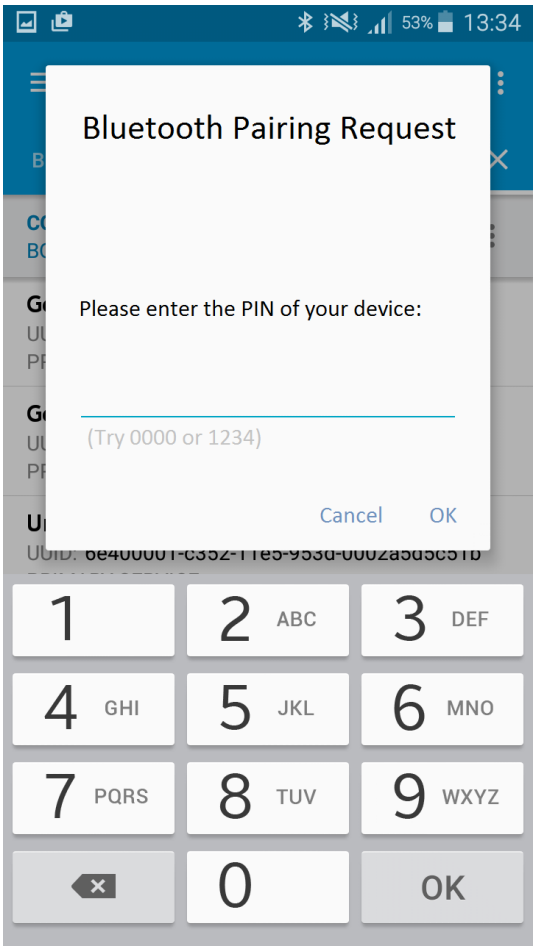


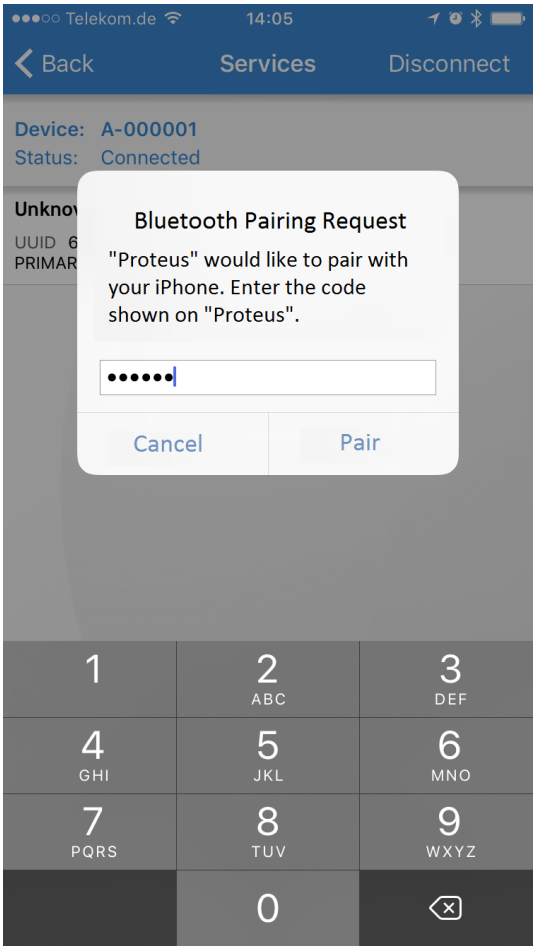


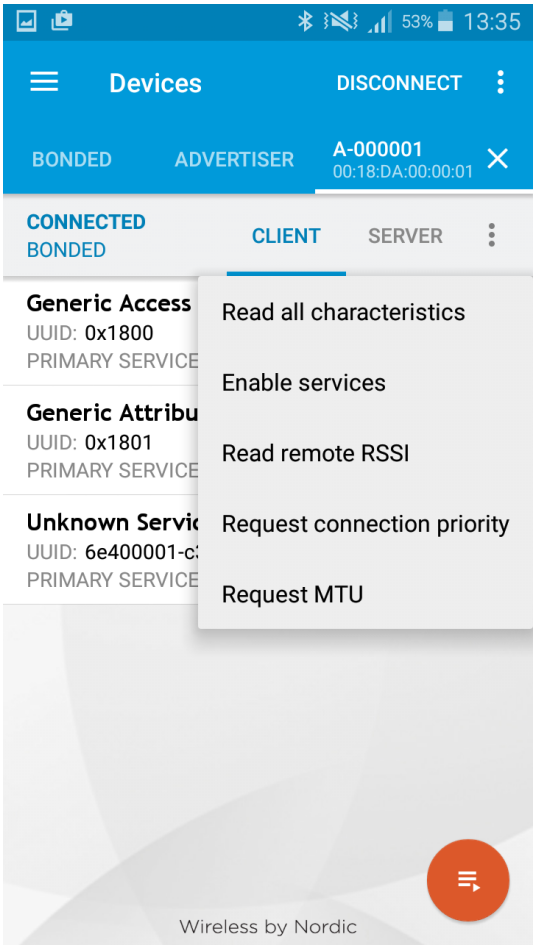
- When the module appears, press the "CONNECT" button.

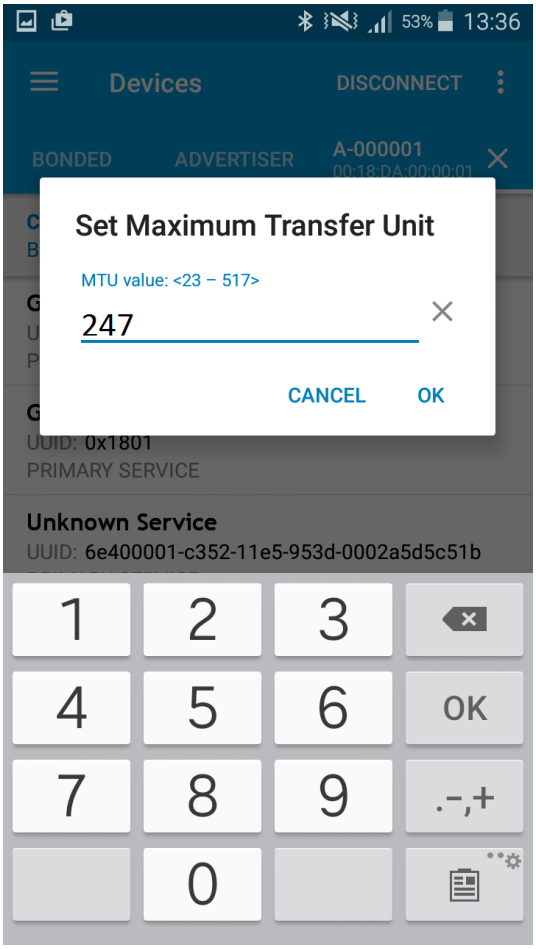
Android	iOS

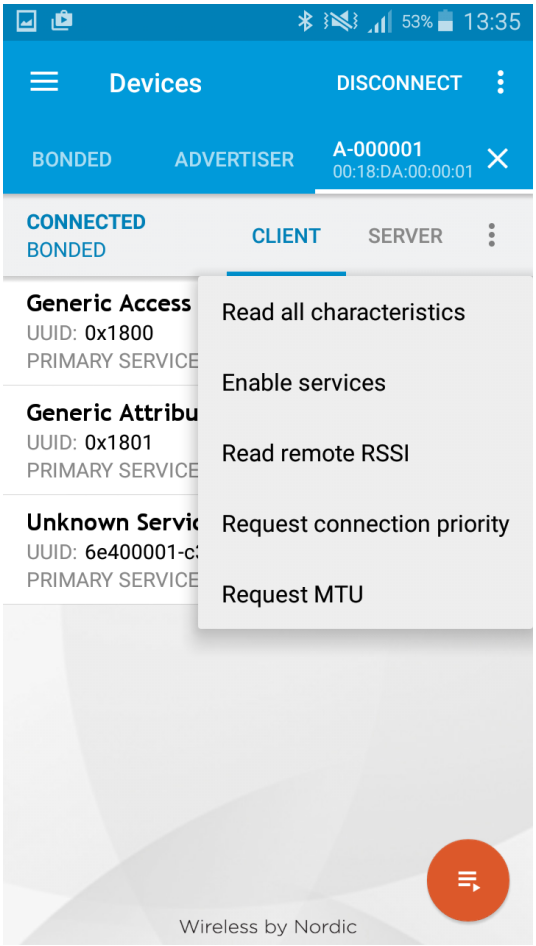
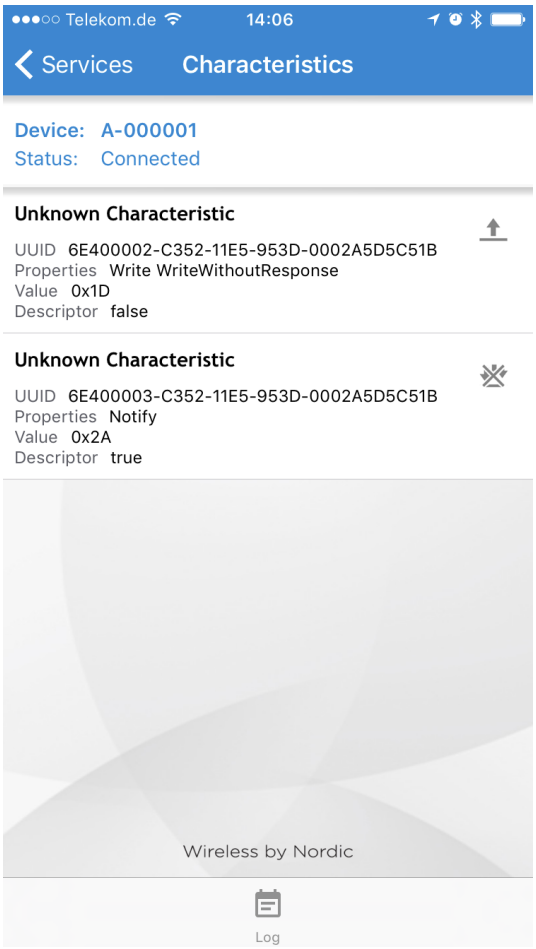
- As soon as the module has received the connection request from the smart phone the blinking LED will switch to constant on.
- Optional pairing: In case a security mode has been configured before, the smart phone requests the user for pairing actions. In case of the static passkey authentication, the Proteus requests to enter the static passkey. The default passkey is "123123". The Bluetooth® coupling requirement pop-up is shown on your smart phone.
  If the bonding feature is enabled in the authentication settings and the bonding information already exists, a re-entering of the passkey is not required when re-connecting.





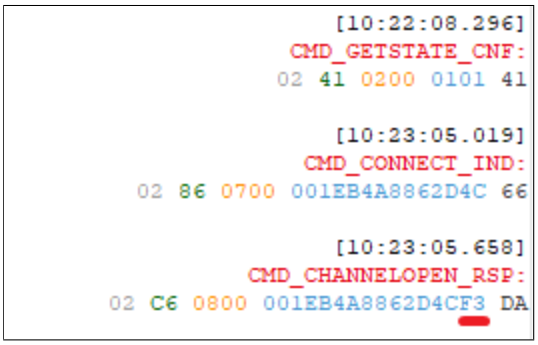
Android	iOS
• Please click on the menu bullets on the right and press "Request MTU" to request for a larger MTU.	• Please click on the "Unknown Service" to start the service discovery and the MTU request.
	

Android	iOS
• The Proteus module allows a MTU of up to 247 bytes, which results in a payload size of 243 bytes. 	• The iOS App runs this step simultaneously in the background, a user-defined MTU is not possible.

Android	iOS
• Again click on the menu bullets on the right and press "Enable services" to enable the notifications.	• Press the arrow on the RX-characteristic 6E400003- C352-11E5- 953D -0002A5D5C51B to enable the notifications. Press it until a cross appears (see below, it has to be pressed at least once). If a cross is already shown press it twice so the cross disappears and then reappears.
	
• As soon as the module has received the notification enable request the second LED on the Proteus EV-Board is turned on. Now you are fully connected and you can access the characteristics to transmit and receive data.	

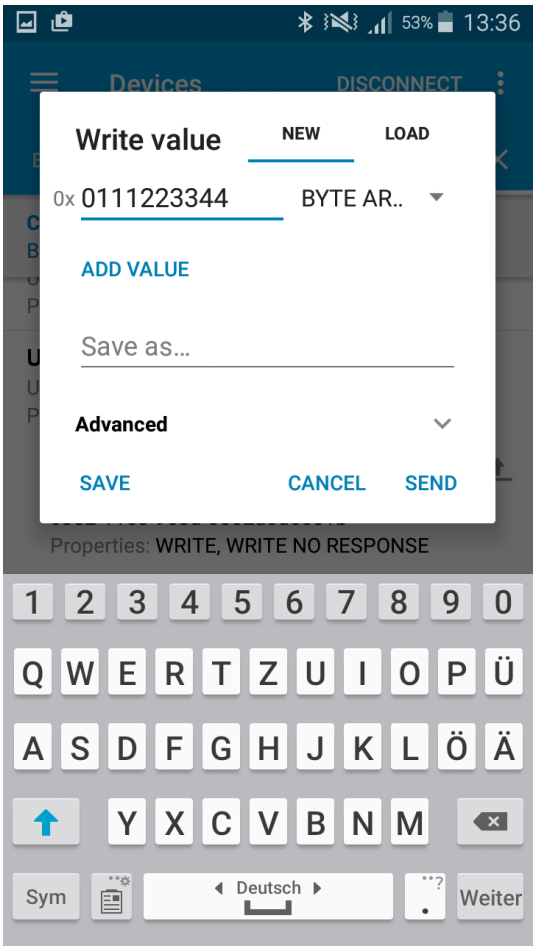
Android	iOS

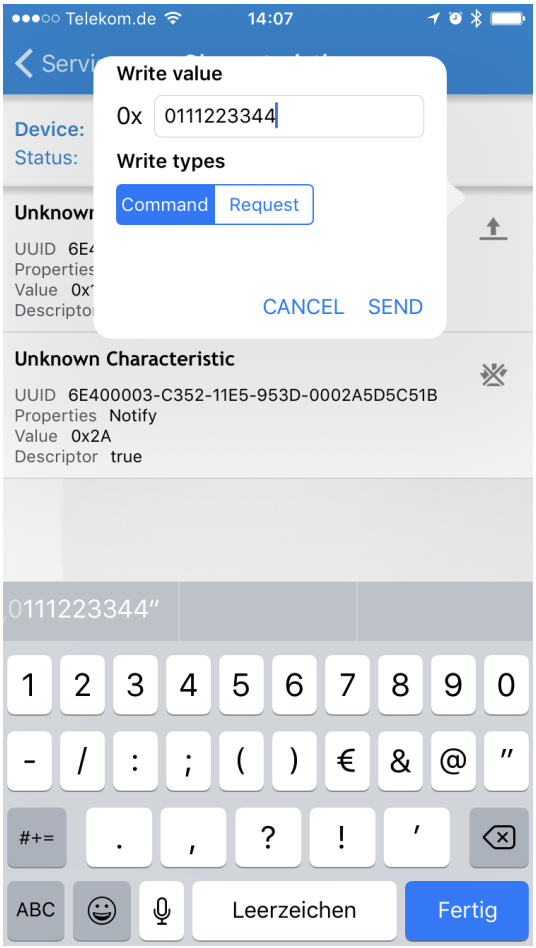
- On the Proteus side, the radio module sent the corresponding `CMD_CONNECT_IND` and `CMD_CHANNELOPEN_RSP` in between. These messages indicate that a connection has been setup and a link has been opened. The `CMD_CHANNELOPEN_RSP` message contains the MTU (maximum transmission unit) of the current link, which defines the maximum supported packet payload length. In this example it's 0xF3 (243_{dec}) bytes payload per packet.



Android	iOS

- To send data to the Proteus module, press the arrow next to the TX-characteristic `6E400002-C352-11E5-953D-0002A5D5C51B` in the **nRF Connect** App.
- First enter 01 right behind the 0x as header byte, followed by your payload (for example 0x11 0x22 0x33 0x44) and press "SEND" to start the transmission. The maximum allowed payload size is dependent on the MTU that was selected in the connection process (see `CMD_CHANNELOPEN_RSP` message on the previous page).

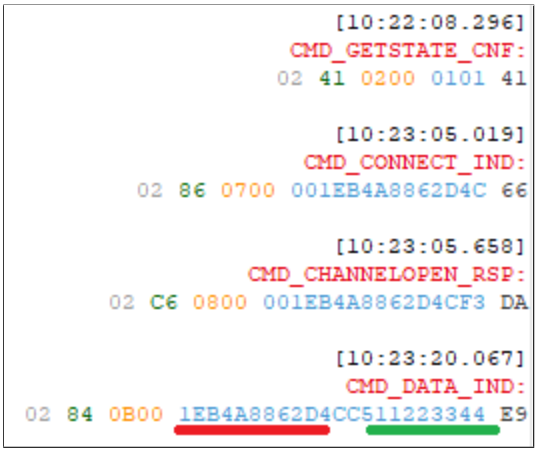




Android	iOS

- The payload that has been sent is output by the Proteus module via UART. In the terminal program a `CMD_DATA_IND` message has been received that contains the `BTMAC` of the sending device and the transmitted payload 0x11 0x22 0x33 0x44. The format of the `CMD_DATA_IND` message is as follows:

Start signal	Command	Length	BTMAC	RSSI	Payload	CS
0x02	0x84	2 Bytes	6 Bytes	1 Byte	(Length - 7) Bytes	1 Byte
0x02	0x84	0x0B 0x00	0x1E 0xB4 0xA8 0x86 0x2D 0x4C	0XC5	0x11 0x22 0x33 0x44	E9

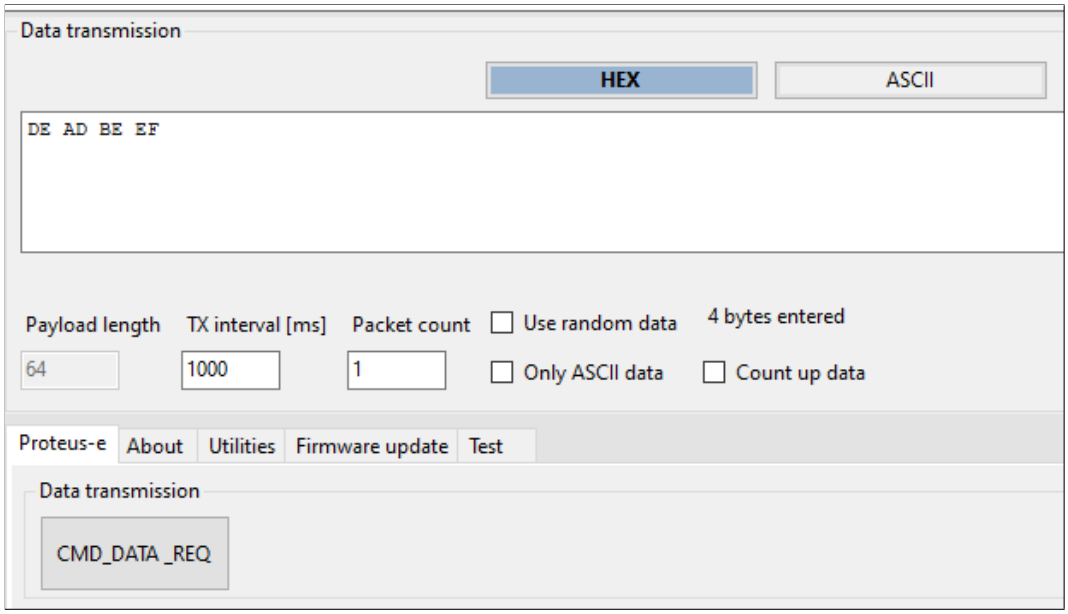


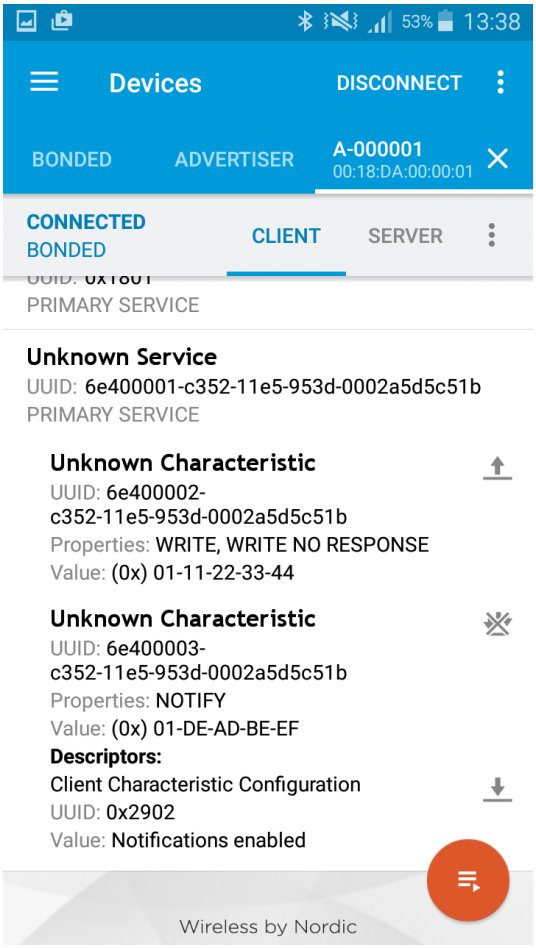
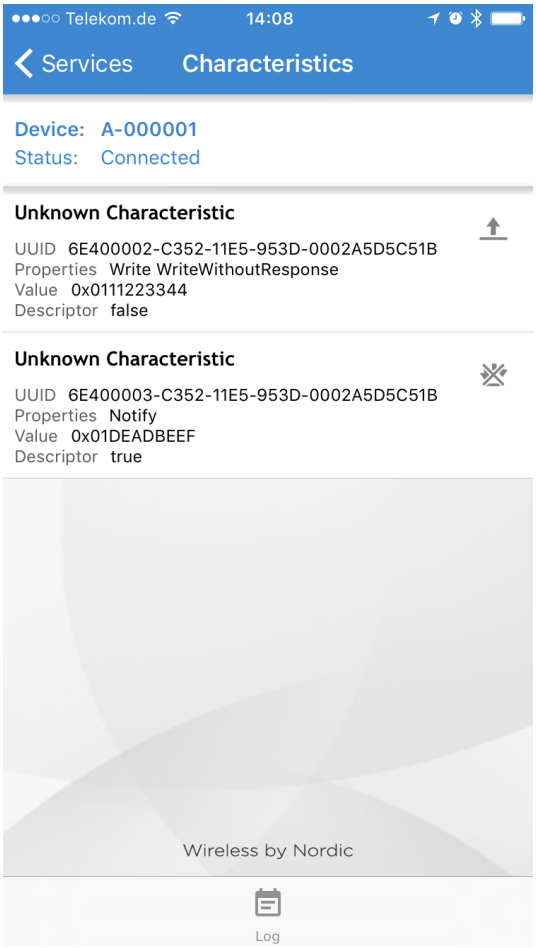
Android	iOS

- To send back data to the smart phone simply insert your payload (here we choose 0xDE 0xAD 0xBE 0xEF) in a `CMD_DATA_REQ` message. The format of the `CMD_DATA_REQ` message is as follows, where the check sum (CS) is calculated as XOR of the preceding bytes:

Start signal	Command	Length	Payload	CS
0x02	0x04	2 Bytes	Length Bytes	1 Byte
0x02	0x04	0x04 0x00	0xDE 0xAD 0xBE 0xEF	0x20

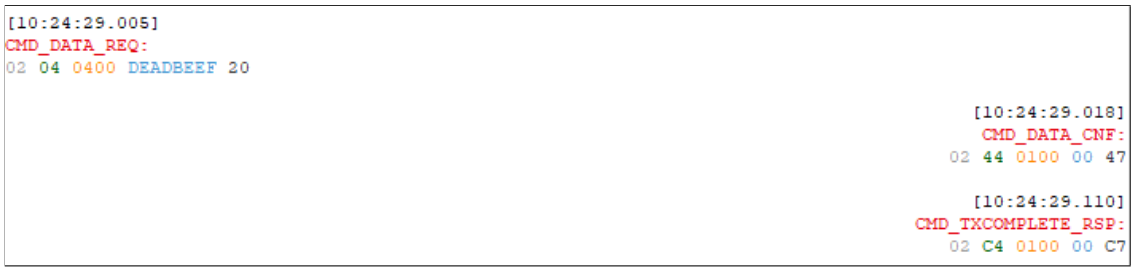
- The header 0x01 of the radio frame header will be automatically applied by the module and is not part of the payload of the `CMD_DATA_REQ` message. To do that in WE UART Terminal, please enter only the payload in the following text field and press the `CMD_DATA_REQ` button. On button press, the remaining command parts are added by the WE UART Terminal.



Android	iOS
• The received data can be found in the RX-characteristic `6E400003-C352-11E5-953D-0002A5D5C51B`. It contains the header byte 0x01 and the payload 0xDE 0xAD 0xBE 0xEF.	
	

Android	iOS

- When sending the CMD_DATA_REQ to the Proteus module, it responds with two different messages. First a CMD_DATA_CNF message is returned, as soon as the request was interpreted. Then a CMD_TXCOMPLETE_RSP message is returned as soon as the data has been transmitted.



Android	iOS

- To disconnect the smart phone from the Proteus module, press the "DISCONNECT" button in the **nRF Connect** App. The Proteus module will output a CMD_DISCONNECT_IND message to indicate that the connection has been closed.

- After disconnecting the Proteus module starts advertising again, such that a new connection can be setup.

5 References

[1] Würth Elektronik. WE UART Terminal PC tool (Smart Commander). https://www.we-online.de/wcs-software.

[2] Nordic Semiconductor. nRF Connect app for Android. https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp.

[3] Nordic Semiconductor. nRF Connect app for iOS. https://apps.apple.com/us/app/nrf-connect-for-mobile/id1054362403.

[4] Würth Elektronik. WE Bluetooth LE Terminal app for Android. https://play.google.com/store/apps/details?id=com.eisos.android.terminal.

[5] Würth Elektronik. WE Bluetooth LE Terminal app for iOS. https://apps.apple.com/de/app/proteus-connect/id1533941485.

[6] Würth Elektronik. Application note 4 - Proteus peripheral only mode. http://www.we-online.com/ANR004.

[7] Würth Elektronik. Source code of WE Bluetooth LE Terminal app (cross platform). https://github.com/WurthElektronik/Proteus-Connect.

[8] Würth Elektronik. Proteus-I user manual. https://www.we-online.de/katalog/de/manual/2608011024000.

[9] Würth Elektronik. Application note 2 - Proteus-I advanced developer guide. http://www.we-online.com/ANR002.

[10] Würth Elektronik. Proteus-II user manual. https://www.we-online.de/katalog/de/manual/2608011024010.

[11] Würth Elektronik. Application note 5 - Proteus-II advanced developer guide. http://www.we-online.com/ANR005.

[12] Würth Elektronik. Proteus-III user manual. https://www.we-online.de/katalog/de/manual/2611011024000.

[13] Würth Elektronik. Application note 9 - Proteus-III(-SPI) advanced developer guide. http://www.we-online.com/ANR009.

[14] Würth Elektronik. Proteus-III-SPI user manual. https://www.we-online.de/katalog/de/manual/2611011024010.

6 Important notes

The Application Note and its containing information ("Information") is based on Würth Elektronik eiSos GmbH & Co. KG and its subsidiaries and affiliates ("WE eiSos") knowledge and experience of typical requirements concerning these areas. It serves as general guidance and shall not be construed as a commitment for the suitability for customer applications by WE eiSos. While WE eiSos has used reasonable efforts to ensure the accuracy of the Information, WE eiSos does not guarantee that the Information is error-free, nor makes any other representation, warranty or guarantee that the Information is completely accurate or up-to-date. The Information is subject to change without notice. To the extent permitted by law, the Information shall not be reproduced or copied without WE eiSos' prior written permission. In any case, the Information, in full or in parts, may not be altered, falsified or distorted nor be used for any unauthorized purpose.

WE eiSos is not liable for application assistance of any kind. Customer may use WE eiSos' assistance and product recommendations for customer's applications and design. No oral or written Information given by WE eiSos or its distributors, agents or employees will operate to create any warranty or guarantee or vary any official documentation of the product e.g. data sheets and user manuals towards customer and customer shall not rely on any provided Information. THE INFORMATION IS PROVIDED "AS IS". CUSTOMER ACKNOWLEDGES THAT WE EISOS MAKES NO REPRESENTATIONS AND WARRANTIES OF ANY KIND RELATED TO, BUT NOT LIMITED TO THE NON-INFRINGEMENT OF THIRD PARTIES' INTELLECTUAL PROPERTY RIGHTS OR THE MERCHANTABILITY OR FITNESS FOR A PURPOSE OR USAGE. WE EISOS DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT RELATING TO ANY COMBINATION, MACHINE, OR PROCESS IN WHICH WE EISOS INFORMATION IS USED. INFORMATION PUBLISHED BY WE EISOS REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE A LICENSE FROM WE eiSos TO USE SUCH PRODUCTS OR SERVICES OR A WARRANTY OR ENDORSEMENT THEREOF.

The responsibility for the applicability and use of WE eiSos' components in a particular customer design is always solely within the authority of the customer. Due to this fact it is up to the customer to evaluate and investigate, where appropriate, and decide whether the device with the specific characteristics described in the specification is valid and suitable for the respective customer application or not. The technical specifications are stated in the current data sheet and user manual of the component. Therefore the customers shall use the data sheets and user manuals and are cautioned to verify that they are current. The data sheets and user manuals can be downloaded at *www.we-online.com*. Customers shall strictly observe any product-specific notes, cautions and warnings. WE eiSos reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time without notice.

WE eiSos will in no case be liable for customer's use, or the results of the use, of the components or any accompanying written materials. IT IS CUSTOMER'S RESPONSIBILITY TO VERIFY THE RESULTS OF THE USE OF THIS INFORMATION IN IT'S OWN PARTICULAR ENGINEERING AND PRODUCT ENVIRONMENT AND CUSTOMER ASSUMES THE ENTIRE RISK OF DOING SO OR FAILING TO DO SO. IN NO CASE WILL WE EISOS BE LIABLE FOR CUSTOMER'S USE, OR THE RESULTS OF IT'S USE OF THE COMPONENTS OR ANY ACCOMPANYING WRITTEN MATERIAL IF CUSTOMER TRANSLATES, ALTERS, ARRANGES, TRANSFORMS, OR OTHERWISE MODIFIES THE INFORMATION IN ANY WAY, SHAPE OR FORM.

If customer determines that the components are valid and suitable for a particular design and wants to order the corresponding components, customer acknowledges to minimize the risk of loss and harm to individuals and bears the risk for failure leading to personal injury or death due to customers usage of the components. The components have been designed and developed for usage in general electronic equipment only. The components are not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where a failure of the components is reasonably expected to cause severe personal injury or death, unless WE eiSos and customer have executed an agreement specifically governing such use. Moreover WE eiSos components are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation, transportation signal, disaster prevention, medical, public information network etc. WE eiSos must be informed about the intent of such usage before the design-in stage. In addition, sufficient reliability evaluation checks for safety must be performed on every component which is used in electrical circuits that require high safety and reliability functions or performance. COSTUMER SHALL INDEMNIFY WE EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF THE COMPONENTS IN SUCH SAFETY-CRITICAL APPLICATIONS.

List of Figures

List of Tables

Contact
Würth Elektronik eiSos GmbH & Co. KG
Division Wireless Connectivity & Sensors

Max-Eyth-Straße 1
74638 Waldenburg
Germany

Tel.: +49 651 99355-0
Fax.: +49 651 99355-69
www.we-online.com/wireless-connectivity