ANR025 PROTEUS-E

QUICK START

VERSION 1.0

MARCH 9, 2022

Revision history

Manual version	Notes	Date
1.0	• Initial version	February 2022

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.
GND	Ground	
Bluetooth LE	Bluetooth Low Energy	
LED	Light Emitting Diode	
LSB	Least Significant bit	
MAC		MAC address of the module.
MSB	Most Significant Bit	
MPS	Maximum Payload Size	The maximum size of the payload, that can be transmitted/received using one Bluetooth® LE transaction.
MTU	Maximum Transmission Unit	Maximum packet size of the Bluetooth® connection.
Payload		The intended message in a frame / package.
PC	Personal Computer	
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	Sistem-on-Chip	
UART	Universal Asynchronous Receiver Transmitter	Allows the serial communication with the module.
USB	Universal Serial Bus	
VDD	Voltage Drain Drain	Supply voltage

Contents

1 Introduction

The Proteus-e is a Bluetooth® module based on the nRF52 Nordic Semiconductors SoC, which provides various Bluetooth® LE and low power features.
In addition to the standard command mode, that uses predefined commands to run and configure the radio module, Würth Elektronik eiSos launches the "transparent mode" on the Proteus-e to use the module as Bluetooth® LE bridge in a simple way. In this mode, a transparent UART interface is provided such that no configuration of the module is required to equip a custom application with it.

The following chapters describe how to establish a connection to the radio module in transparent (see chapter 4) and command mode (see chapter 5).

2 Prerequisites

- A Proteus-e evaluation board in factory state.
- A central device that initiates the connection setup. For example
  - a smart phone with Bluetooth® LE function and the Proteus Connect App or Nordic Semiconductor nRF Connect App
  - a Proteus-I,-II,-III evaluation board, mini evaluation board or Proteus-I,-II USB radio stick.

To be sure that all Proteus devices are in factory state, please run a factory reset before doing any other action.

Please check whether the most recent firmware is installed on any Proteus device used.

3 General information

For a better understanding of the content of this chapter, basic knowledge of the Bluetooth® standard as well as that of the SPP-like profile is of advantage. Please find more details on that in the respective advanced developer guide:

- ANR024 Proteus-e advanced developer guide [3]

3.1 How to choose the operation mode?

The operation mode of the Proteus-e can be selected using different voltage levels of the *MODE_1* pin during module start-up.
The module starts in transparent mode, when a HIGH level is applied at the *MODE_1* pin and a reset is done via the */RESET* pin. If the *MODE_1* pin is LOW during the reset, the module starts in normal operation mode with command interface.

A pull-down is applied to the *MODE_1* pin during start-up. Thus increased currents can occur for a period $\leq$ 1 ms. After the start-up procedure has been finished, the *MODE_1* pin and thus the applied signal level has no function.

In case of the evaluation board for Proteus-e, the *MODE_1* pin is on pin 4 of the P1 pin header. Connect this pin to GND (P4) or leave it open and press the reset button to restart the Proteus-e in command mode. Connect this pin to VDD (P3) and press the reset button to restart the module in transparent mode.

3.2 General connection setup information

Figure 1 shows the steps that have to be performed successively during connection setup:

1. Physical connection establishment
   A physical connection has to be established first. Therefore, a central device (i.e. smart phone) has to connect to the Proteus-e which runs as peripheral.

2. Pairing process (optional, in case the user setting `RF_SecFlags` has been set)
   The authentication and exchange of encryption information is part of the pairing process. The central device must request at least the same security level to access the characteristics of the Proteus-e.

In case the peripheral device has enabled a security mode, but the central device goes on with the next steps without placing the pairing request, the peripheral device disconnects immediately as the required security level is not achieved. The same holds, if the central device places a bonding request with lower security level than required by the peripheral device.

3. Exchange of the maximum transmission unit (MTU) (optional)
   The maximum transmission unit can be increased to allow the transmission of larger data packets. The Proteus-e allows an MTU of up to 247 bytes, which results in a maximum payload size (MPS) of 243 bytes. Not selecting a higher MTU will use the Bluetooth® LE 4.0 default MTU which results in a MPS of 19 bytes, but will be compatible to pre Bluetooth® LE 4.2 devices.

4. Discover the characteristics of the Proteus-e SPP-like profile
   The characteristics offered by the Proteus-e have to be discovered by the central.

5. Notification enable
   To transmit data from the peripheral to the central, the central must enable the notifications on the peripheral's characteristics. After this step, the channel is open and data transmission can start. In case of transparent mode, the UART is enabled at this time.

For the description, we assume that a smart phone is the initiator of the connection. Thus, it acts as central and the Proteus-e acts as peripheral in figure 1.

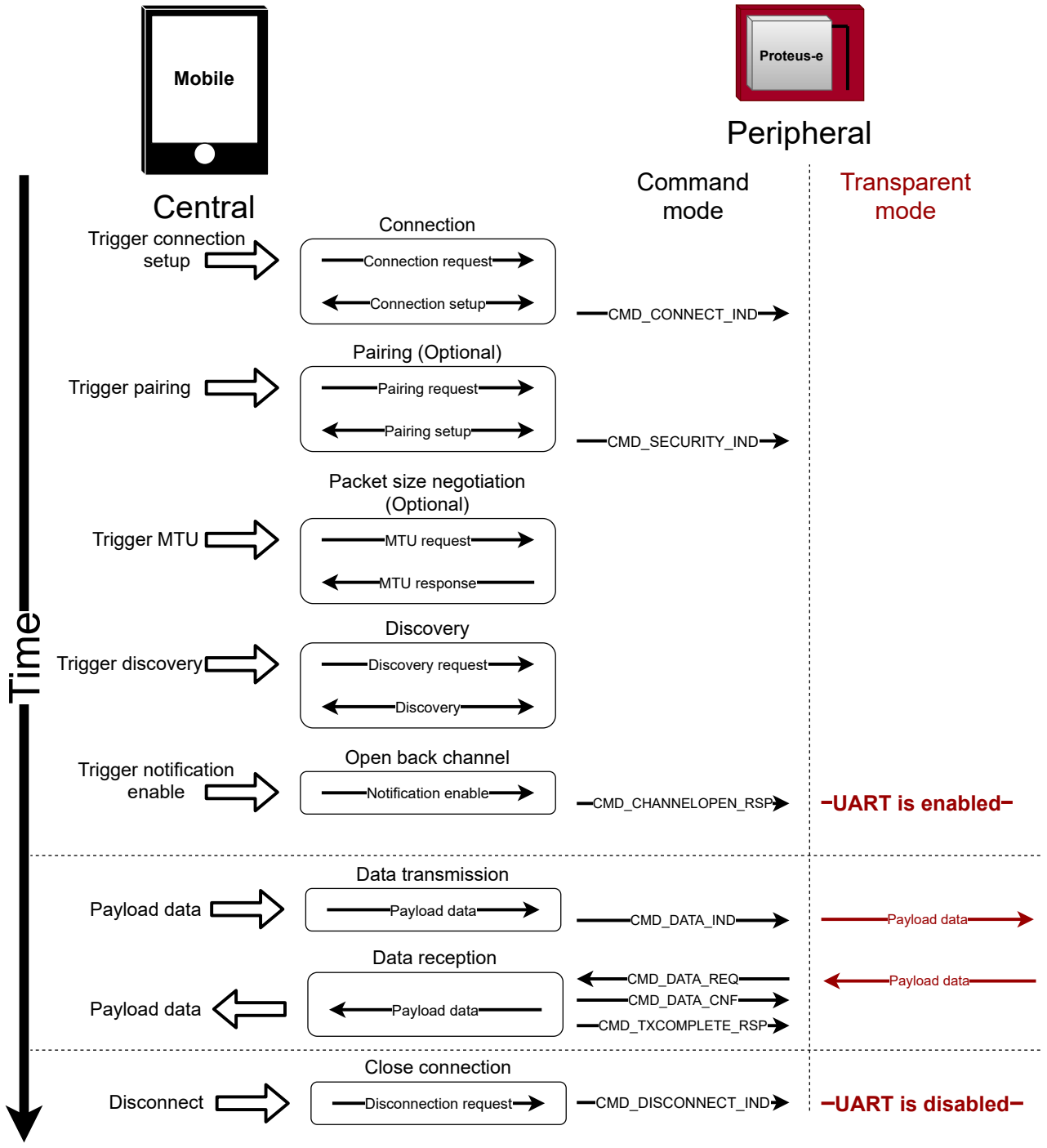


Figure 1: Steps for the connection setup

3.3 Transparent mode: Preconfiguring of the module

Only in case in transparent mode the user settings (such as UART baud rate, security mode or the static passkey value) have to be modified, please start the module in command mode. Then use the commands like `CMD_SET_REQ` to update these user settings and switch back to transparent mode.

For security reasons it is strongly recommended to change the default `RF_StaticPasskey` to a customer specific passkey in case static passkey pairing method is used.

Custom product: Upon request, Würth Elektronik eiSos can apply customer specific configuration(s) during the production process.

4 Transparent mode: Quickstart

In chapter 3.2 it has been described which steps have to be performed by the central device to setup a connection to a Proteus-e radio module running in **transparent mode**. What this means in practice will be shown in this chapter.

4.1 Smart phone using nRFConnect app as central device

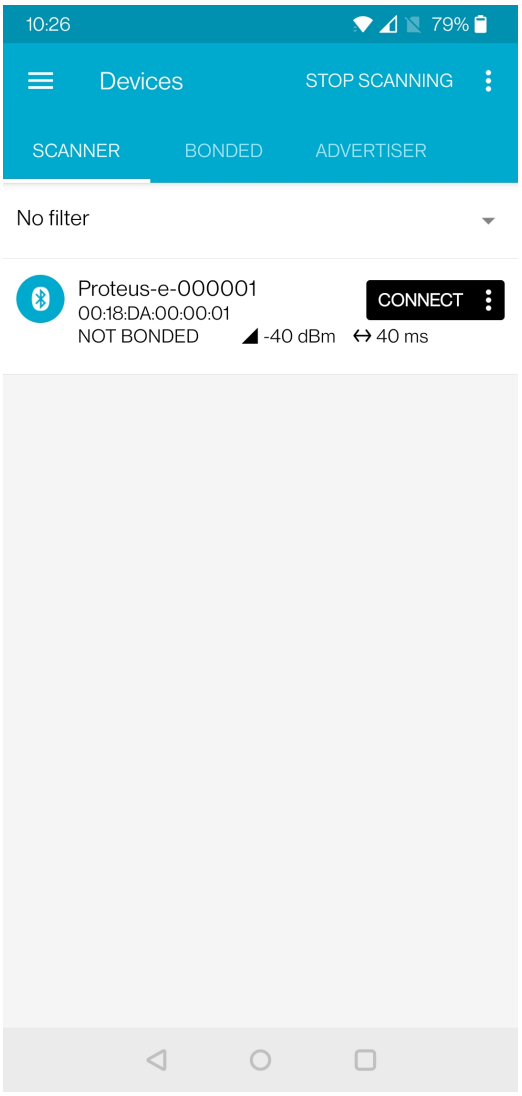
This chapter describes how to setup a connection to the Proteus-e radio module in transparent mode, when a smart phone and the nRF Connect App are used.

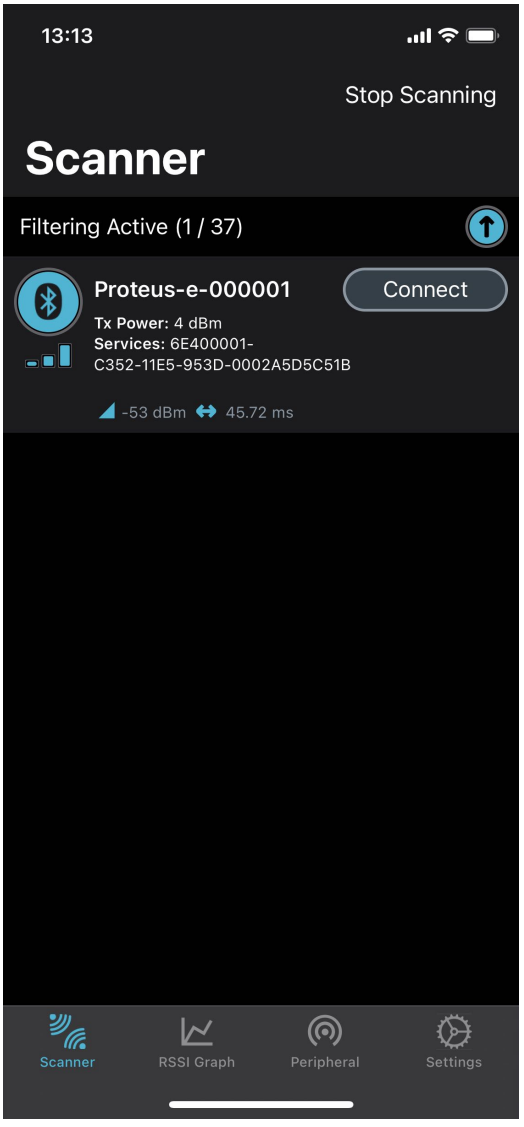
The nRF Connect App is an open source App providing standard Bluetooth® LE functions for iOS as well as for Android devices.

Please perform the following steps:

Android	iOS

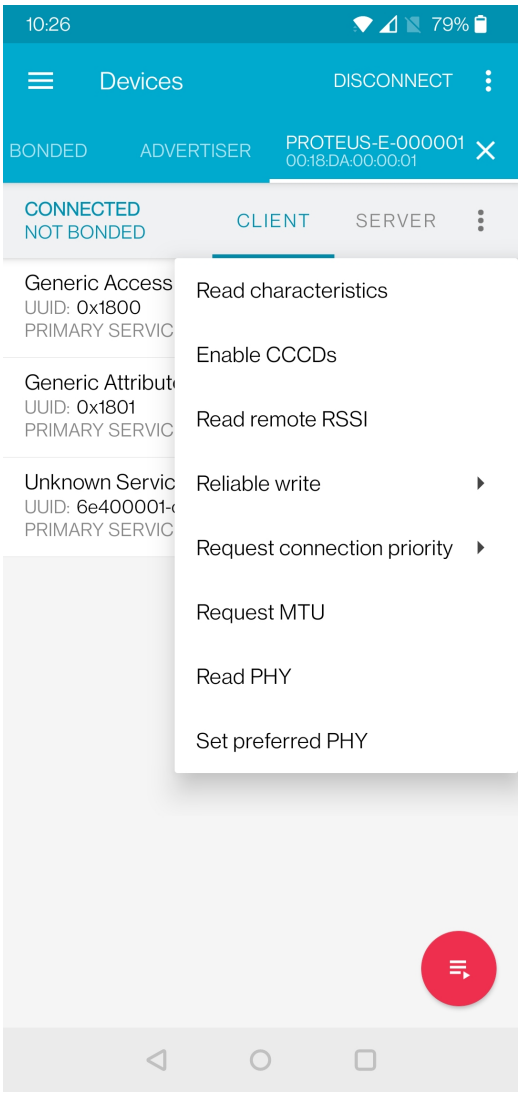
- Connect the module to a PC and open a terminal program using the Proteus-e default UART settings (115200 Baud, 8n1).
- Set the module into transparent mode as described in chapter 3.1. Initially, the module is advertising. Thus, the Proteus-e *LED_1* is blinking slowly.
- Start your smart phone, enable the Bluetooth® LE feature and start the nRF Connect App.
- Press "SCAN" to find the module on the radio.
- When the module appears, press connect.

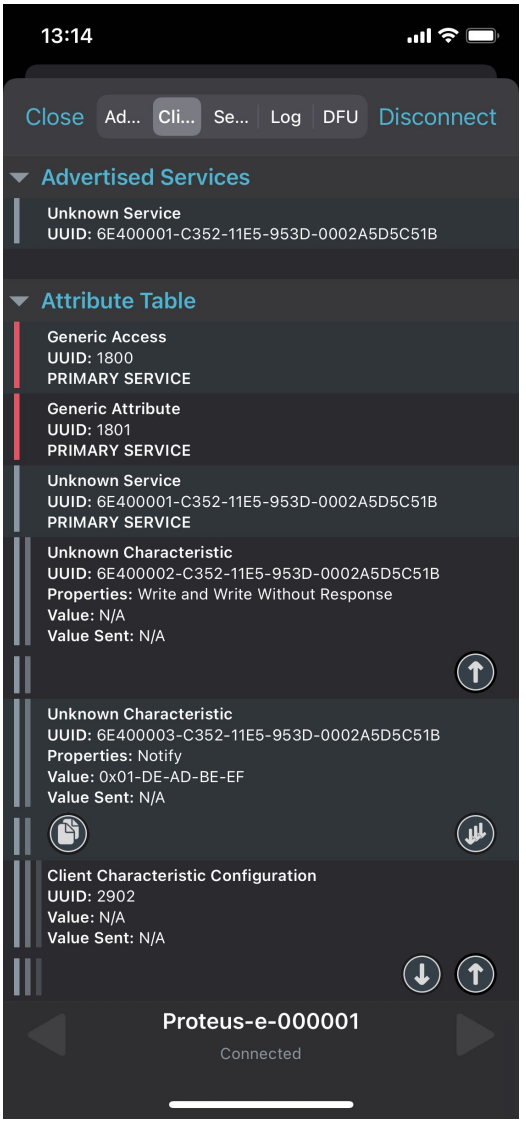


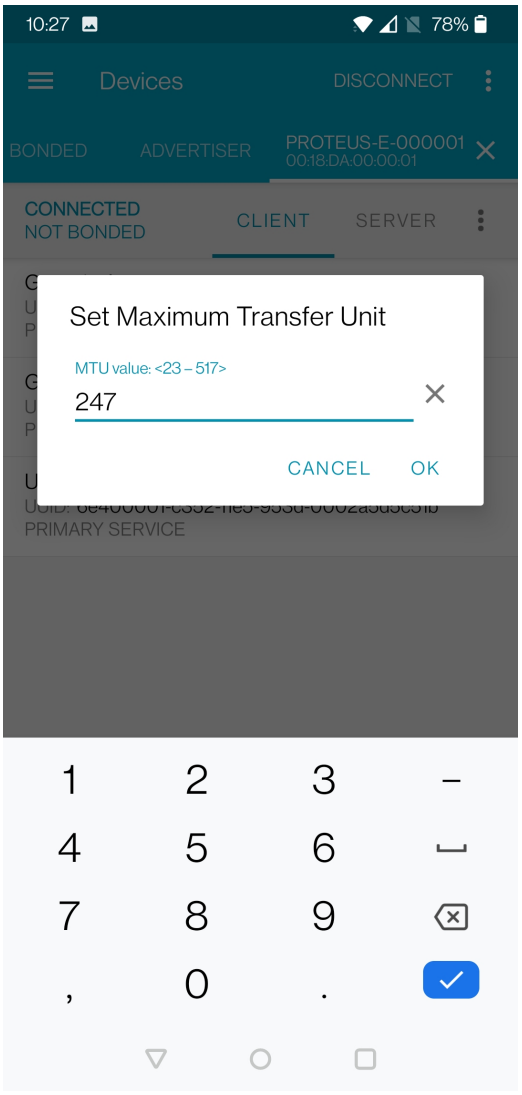


- As soon as the module has received the connection request, the module *LED_1* will blink faster.

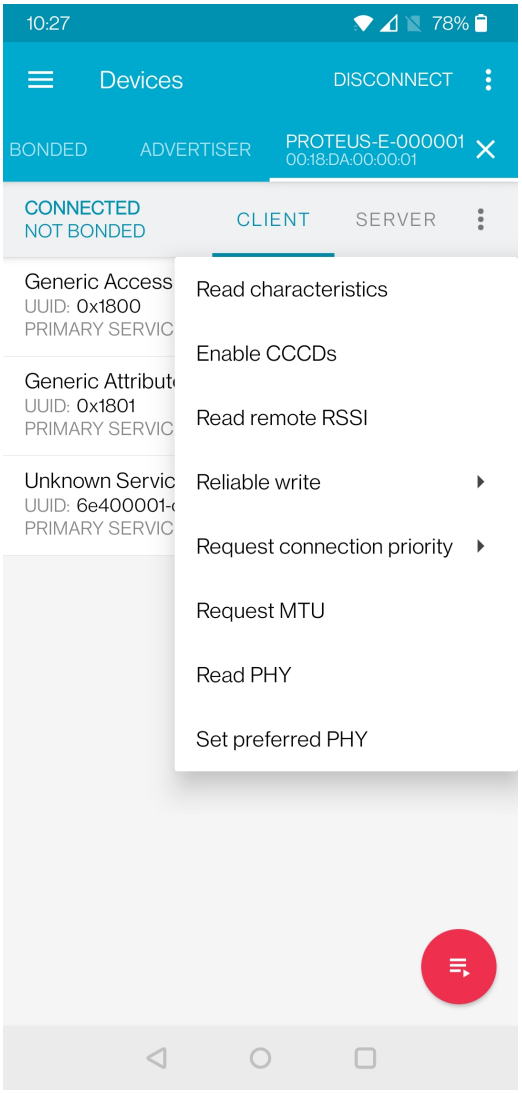
Android	iOS
• Please click on the menu bullets on the right and press "Request MTU" to request for a larger MTU.	• Please click on the "Unknown Service" to start the service discovery and the MTU request.

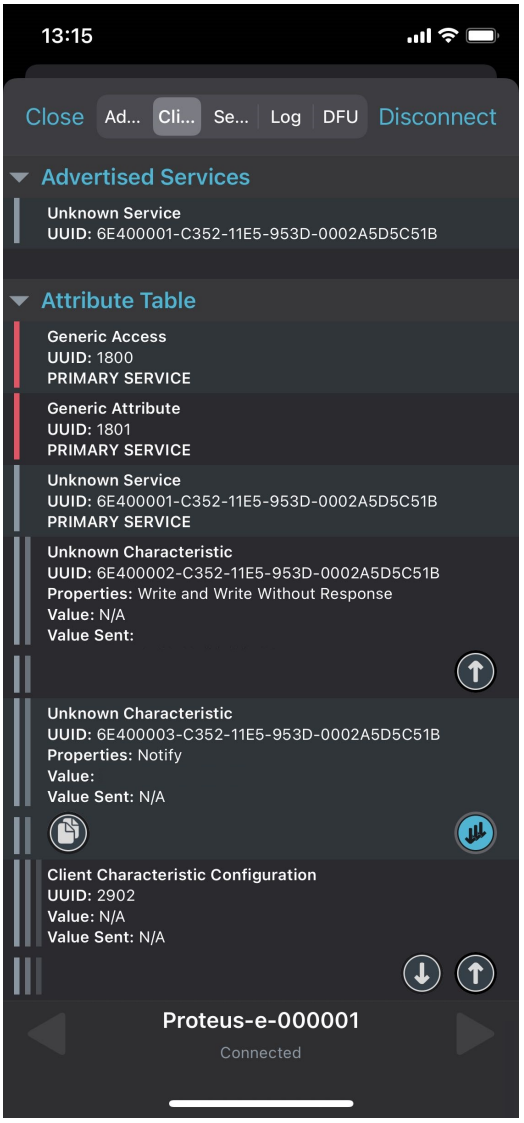




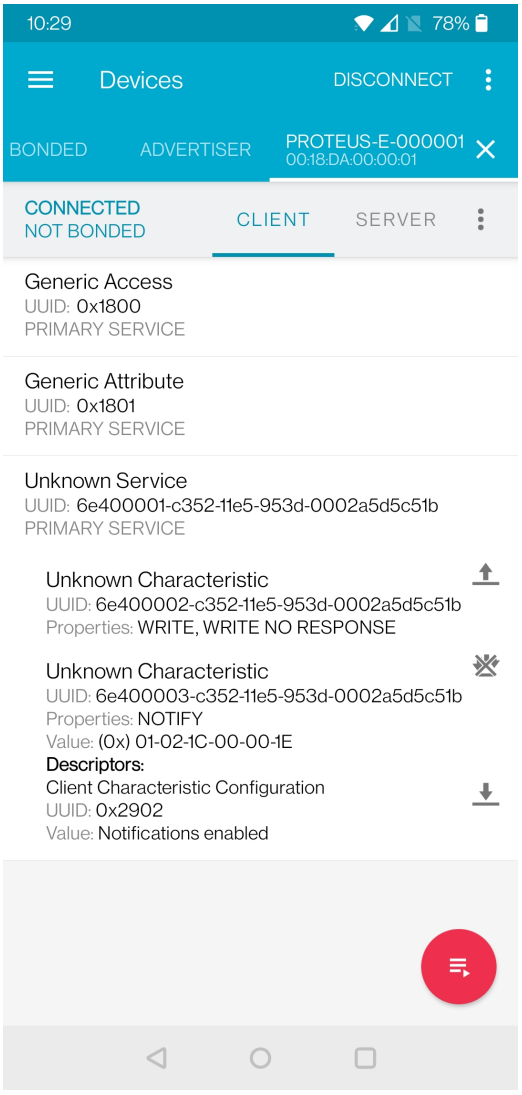
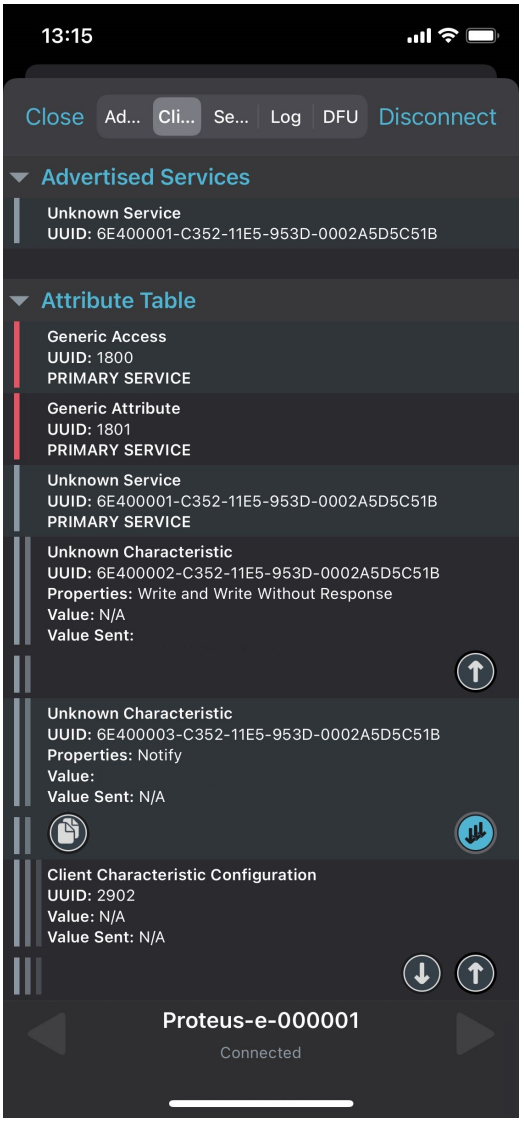
Android	iOS
• The Proteus-e allows an MTU of up to 247 bytes, which results in a maximum payload size (MPS) of 243 bytes. 	• The iOS App runs this step simultaneously in the background, a user-defined MTU is not possible.

Android	iOS
• Again click on the menu bullets on the right and press "Enable services"/"Enable CCCDs" to enable the notifications.	• Press the arrows on the RX-characteristic `6E400003- C352-11E5- 953D -0002A5D5C51B` to enable the notifications. Press it until the symbol turns blue (see below, it has to be pressed at least once). If it is already blue, press it twice such that it is deselected and selected again.

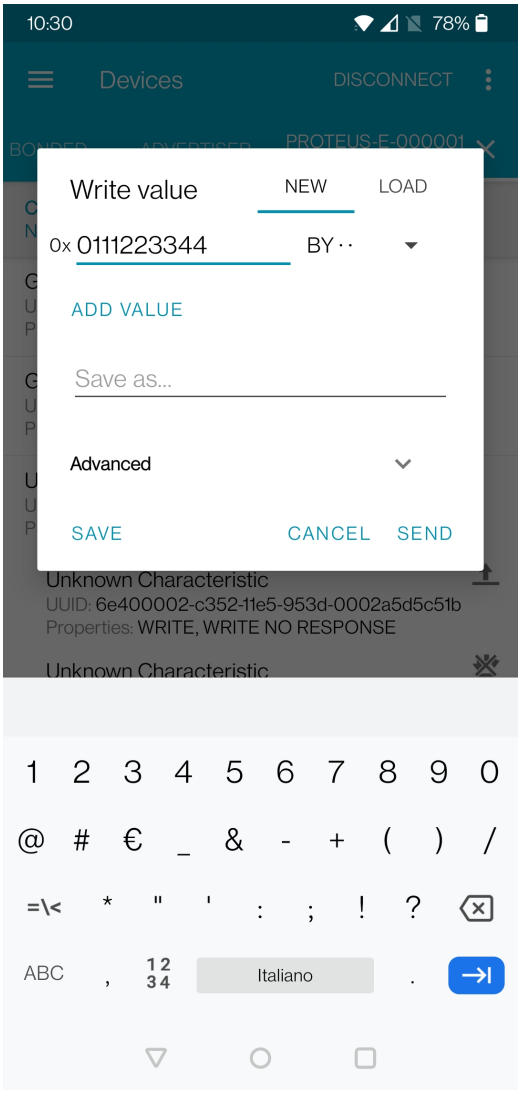
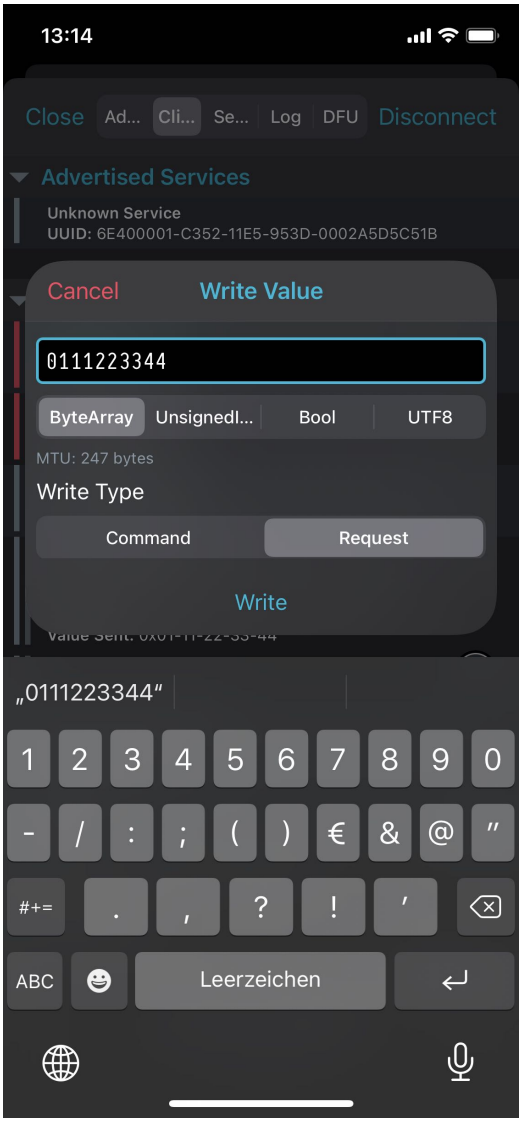




- As soon as the module has received the notification enable request the Proteus-e *LED_1* is static on.

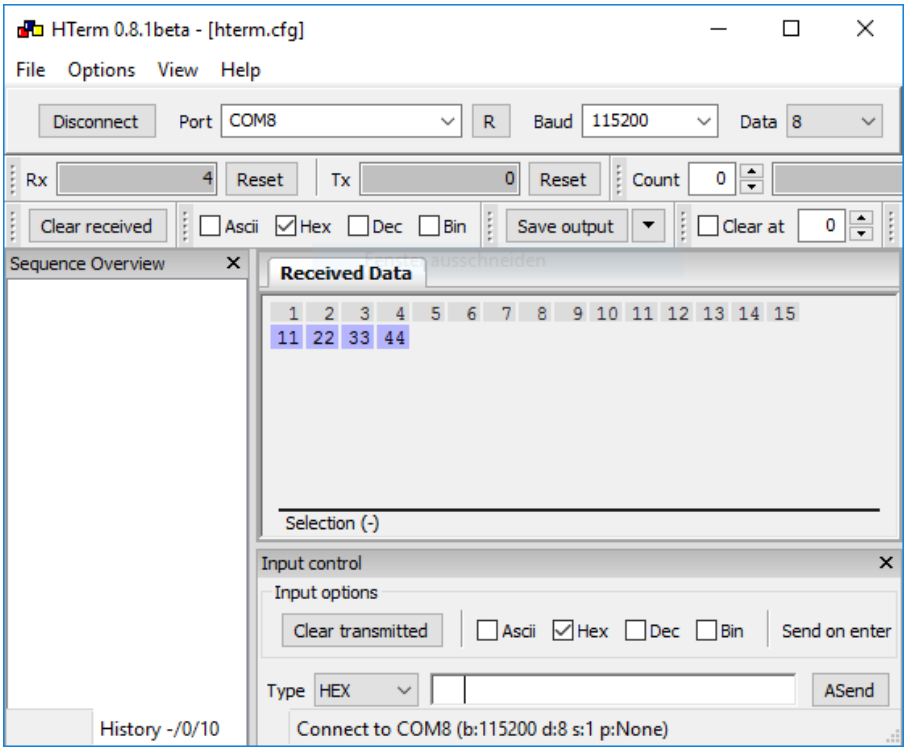
Android	iOS
	

- Now you are fully connected and you can access the characteristics. The maximum size of payload depends on the chosen MTU size. Here we chose 247 bytes, which allows us to send 243 bytes of payload (MPS) via the channel.
- To send data to the Proteus-e, press the arrow next to the TX-characteristic `6E400002-C352-11E5-953D-0002A5D5C51B`.
- Then enter 0x01 as header byte followed by your payload (for example 0x11 0x22 0x33 0x44) and press "SEND". The payload size is dependent on the MPS that was negotiated in the connection process. The smallest supported MTU for all Bluetooth® 4.0 (or newer) devices results in a max payload size (MPS) of 19 bytes.

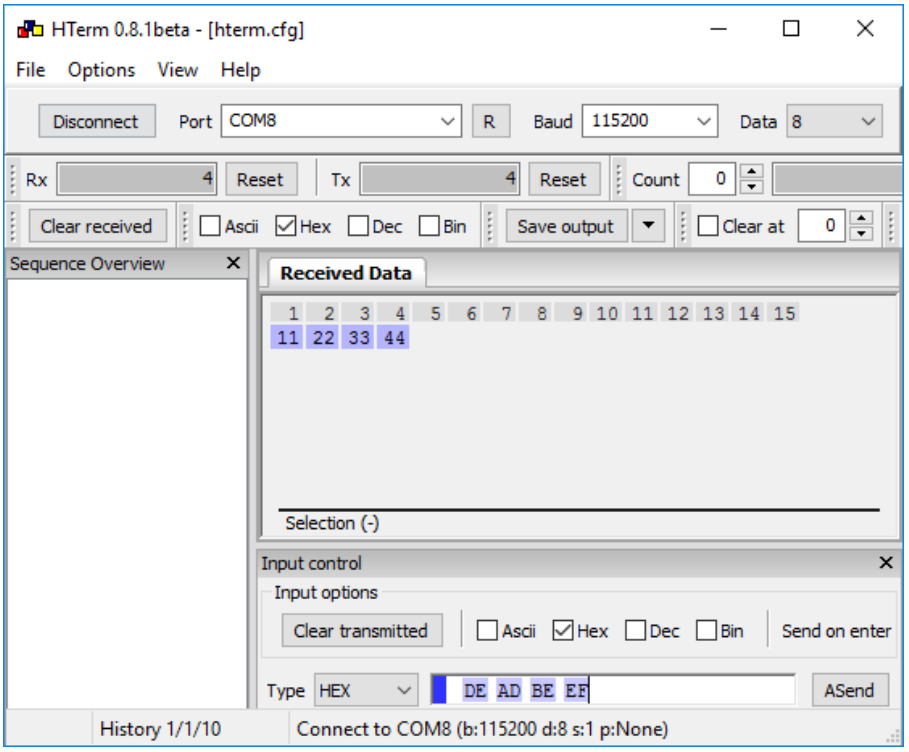
Android	iOS
	

- The payload that has been sent via radio is output by the Proteus-e via the transparent UART interface. This means, that only payload data is transmitted, without any packet header or footer. Thus, the transmitted bytes 0x11 0x22 0x33 0x44 are displayed on the connected terminal program.

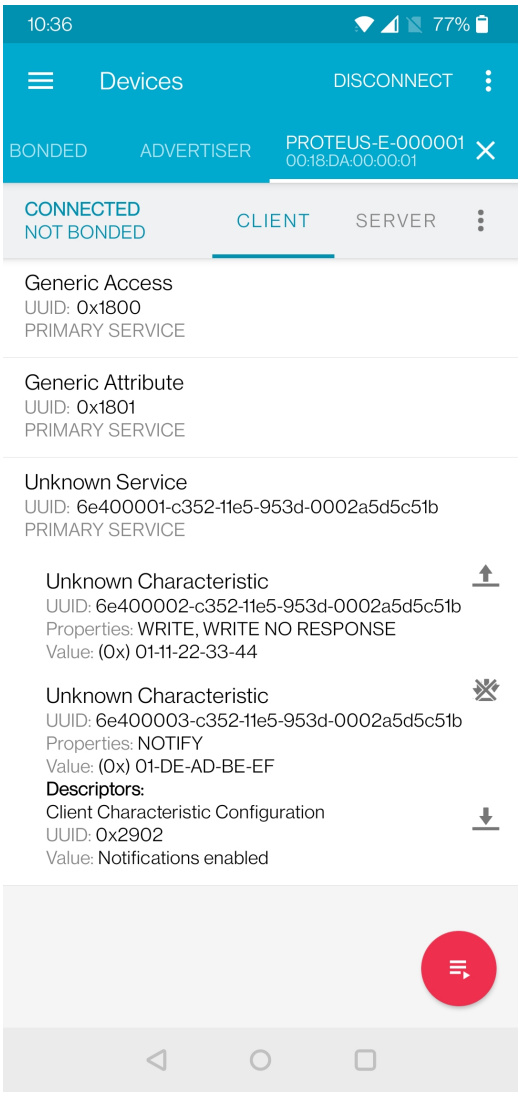
Android	iOS

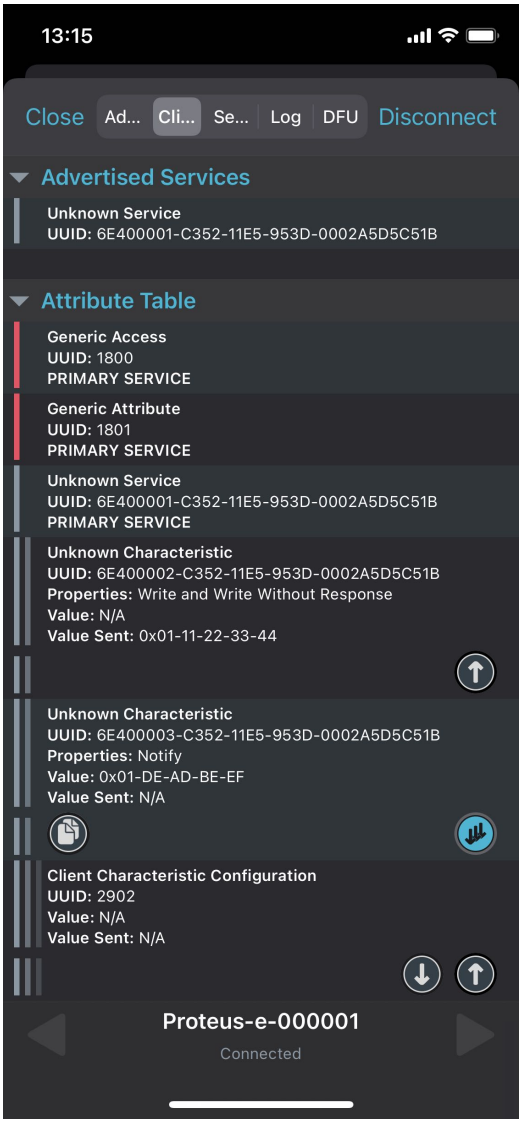


- To send back data, simply enter your payload in the respective terminal program field and press enter. In this example we choose 0xDE 0xAD 0xBE 0xEF. The header 0x01 will be automatically applied by the module and is not to be transmitted by the host.
- Here again the maximum payload size (MPS) must be respected.



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.	





4.2 Smart phone using Proteus Connect app as central device

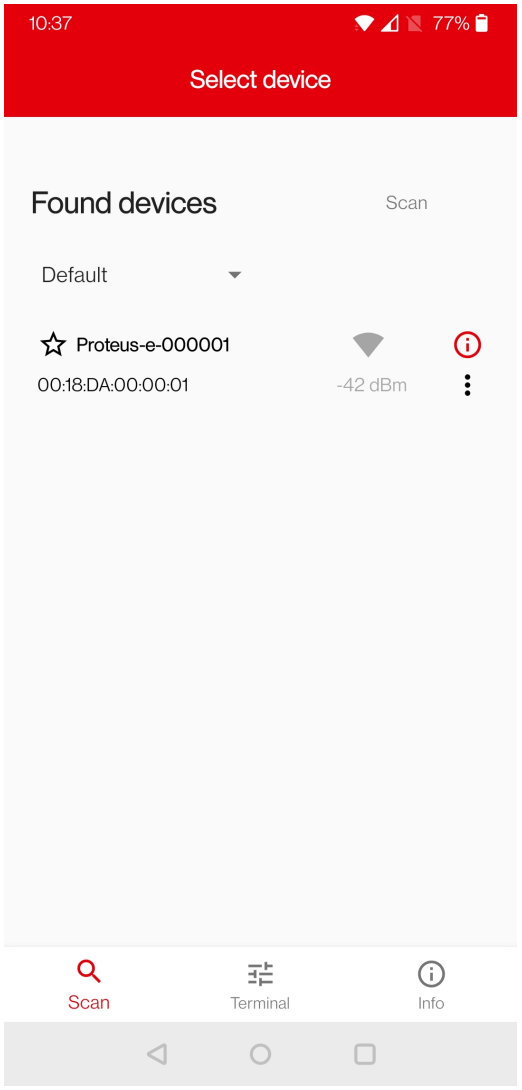
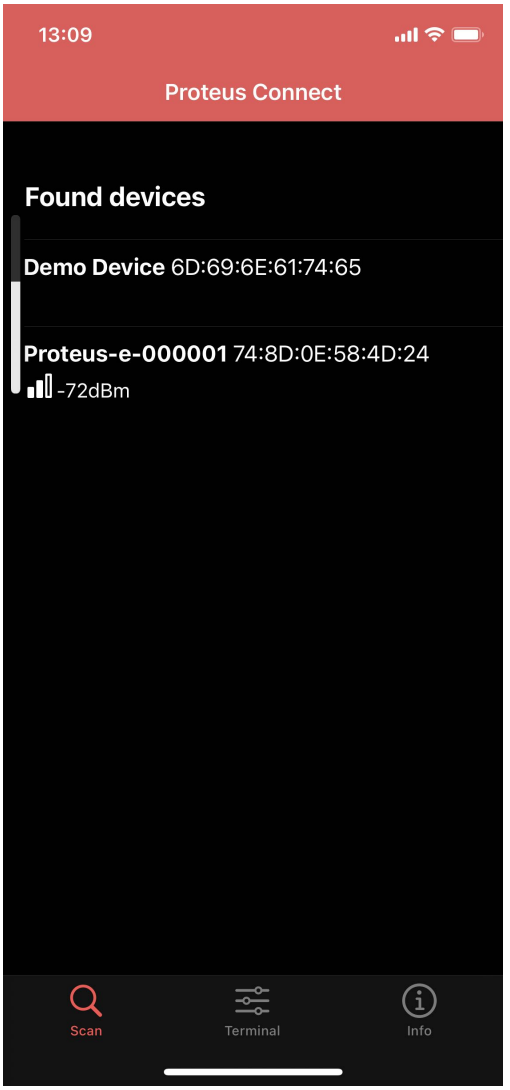
This chapter describes how to setup a connection to the Proteus-e radio module in transparent mode, when a smart phone and the Proteus Connect App are used.

The Proteus Connect App (for iOS [7] and Android [6]) is provided by Würth Elektronik eiSos as executable as well as source code.

Please perform the following steps:

Android	iOS
• Connect the module to a PC and open a terminal program using the Proteus-e default UART settings (115200 Baud, 8n1). • Set the module into transparent mode as described in chapter 3.1. Initially, the module is advertising. Thus the Proteus-e *LED_1* is blinking slow. • Start your smart phone, enable the Bluetooth® LE feature and start the Proteus Connect App.	

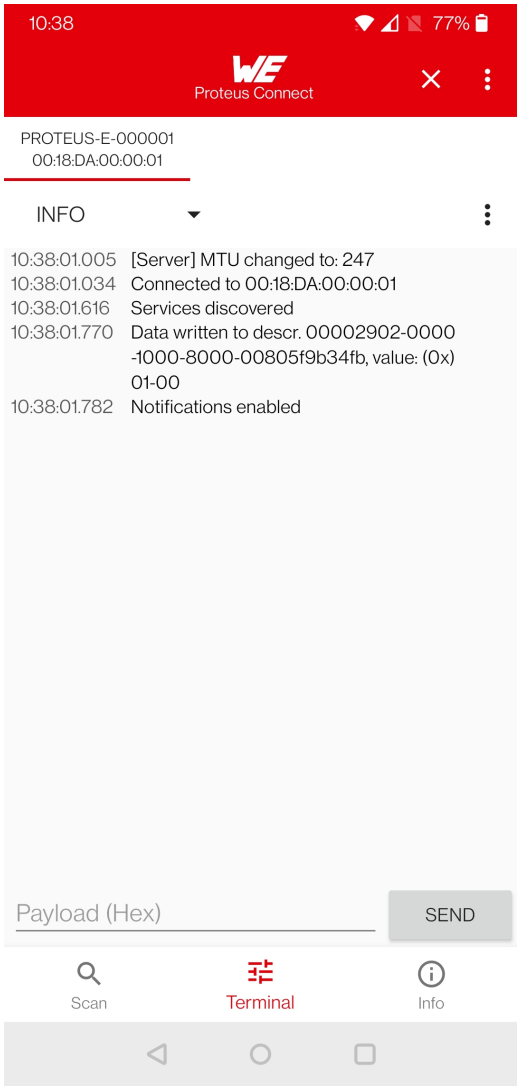
Please note that Bluetooth® LE function of Android devices is only available if the location services are enabled in addition.

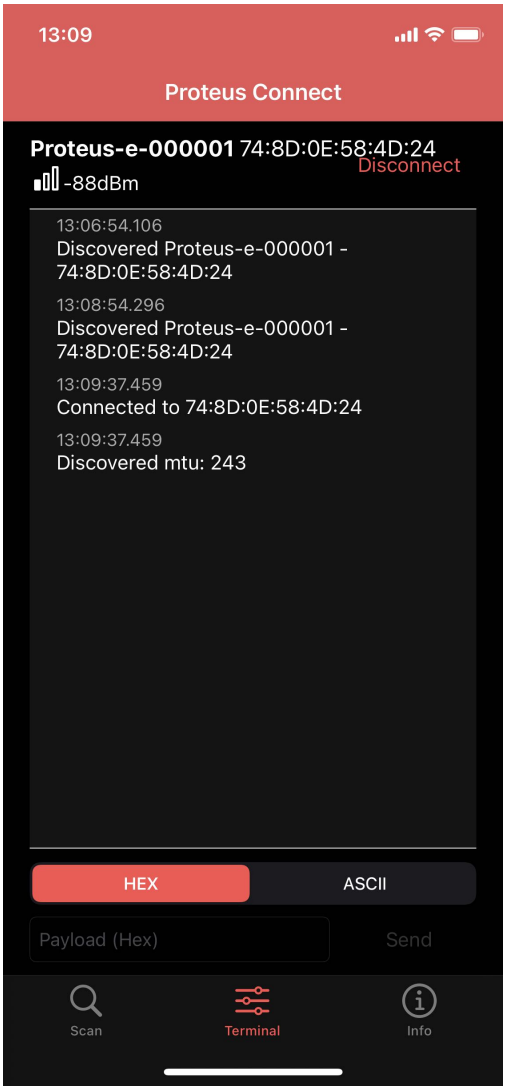
Android	iOS
• Press "Scan" to find the module on the radio.  	

- When the module appears, press connect.
- As soon as the module has received the connection request, the module *LED_1* blinks fast.

Android	iOS

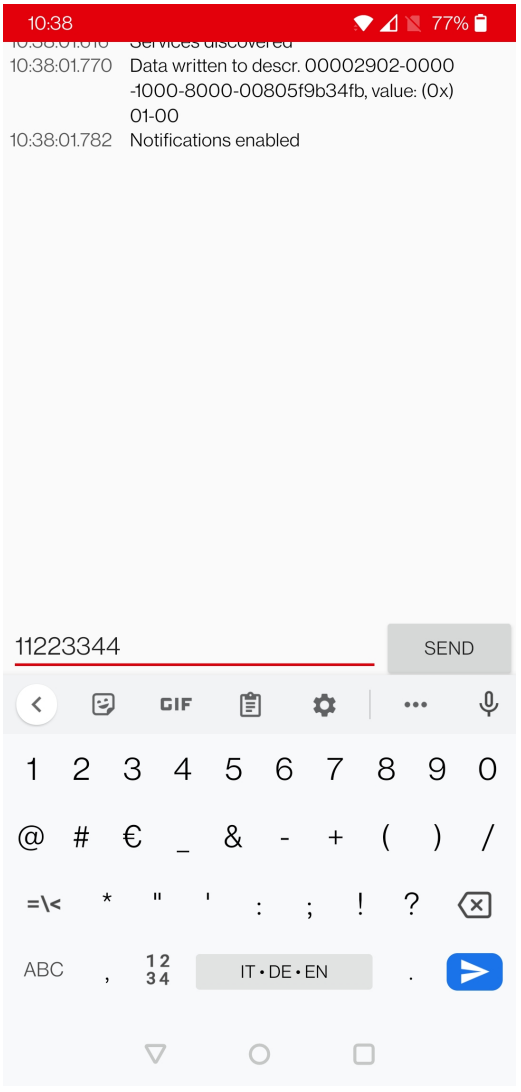
- As soon as the connection has been setup successfully *LED_1* is turned static on. Now data can be transmitted in both directions.

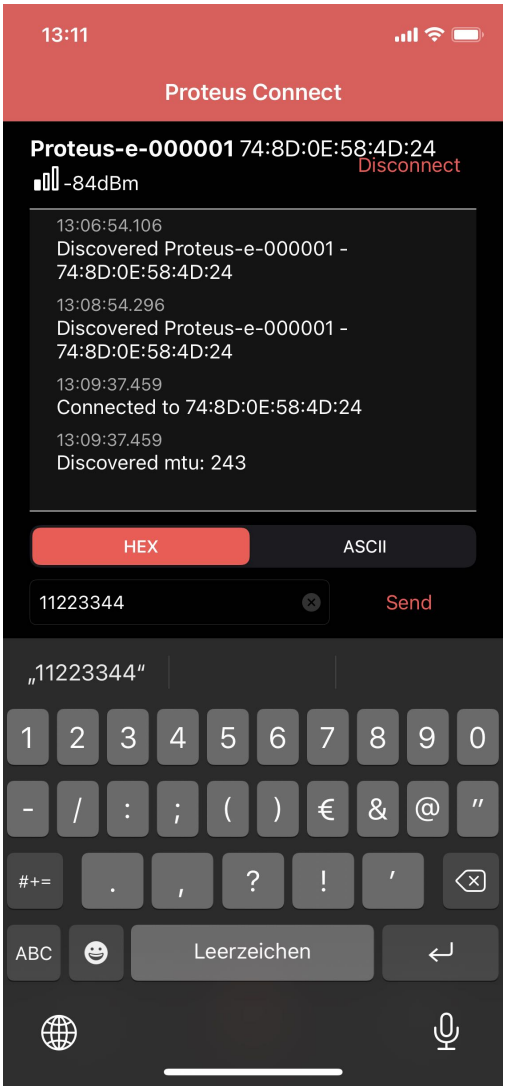




Android	iOS

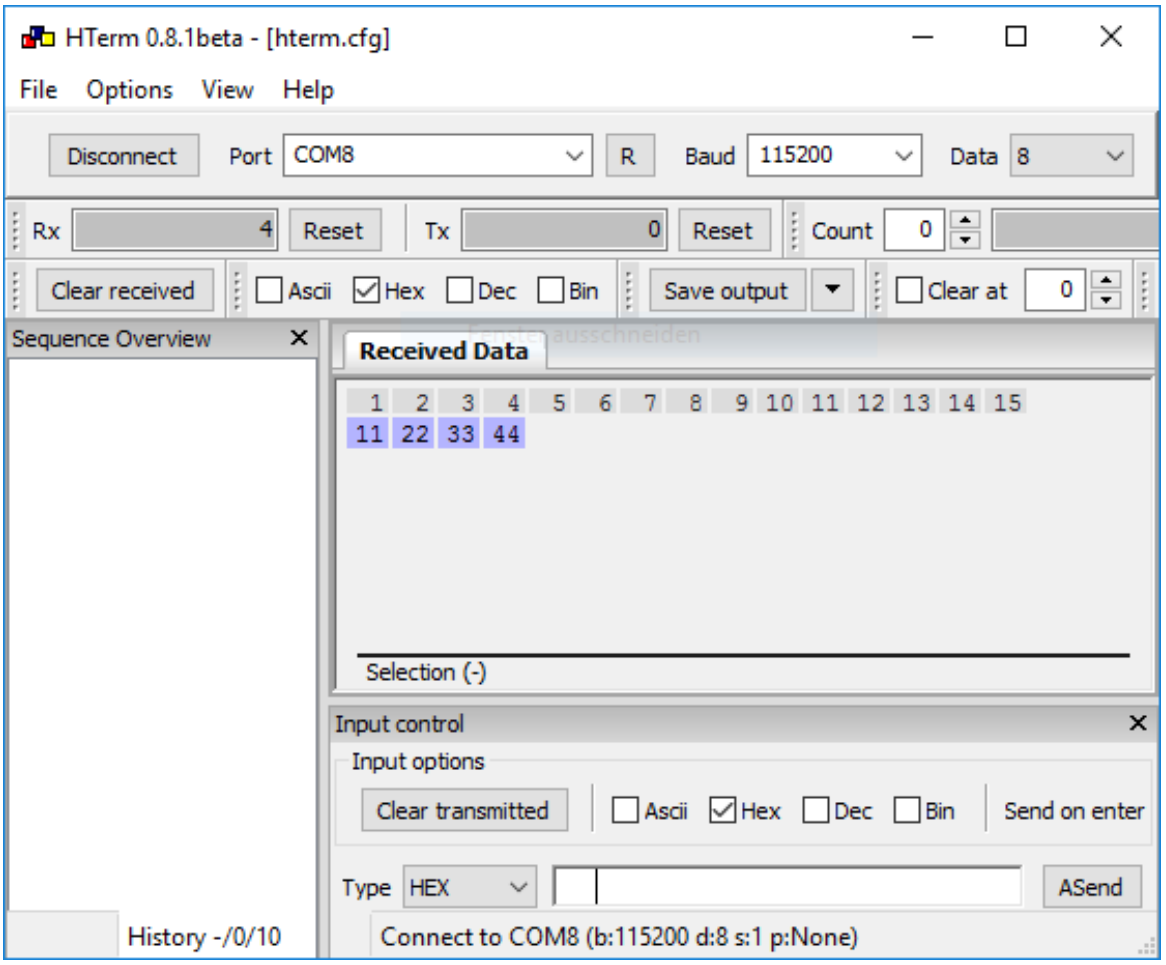
- First of all, we want to send data from the smart phone to the radio module. To do so, enter your payload (for example 0x11 0x22 0x33 0x44) and press "SEND". The maximum payload size (MPS) is dependent on the MTU that was negotiated in the connection process. The smallest supported MTU for all Bluetooth® 4.0 (or newer) devices results in a max payload size (MPS) of 19 bytes. iOS and Android usually allow up to 243 bytes.





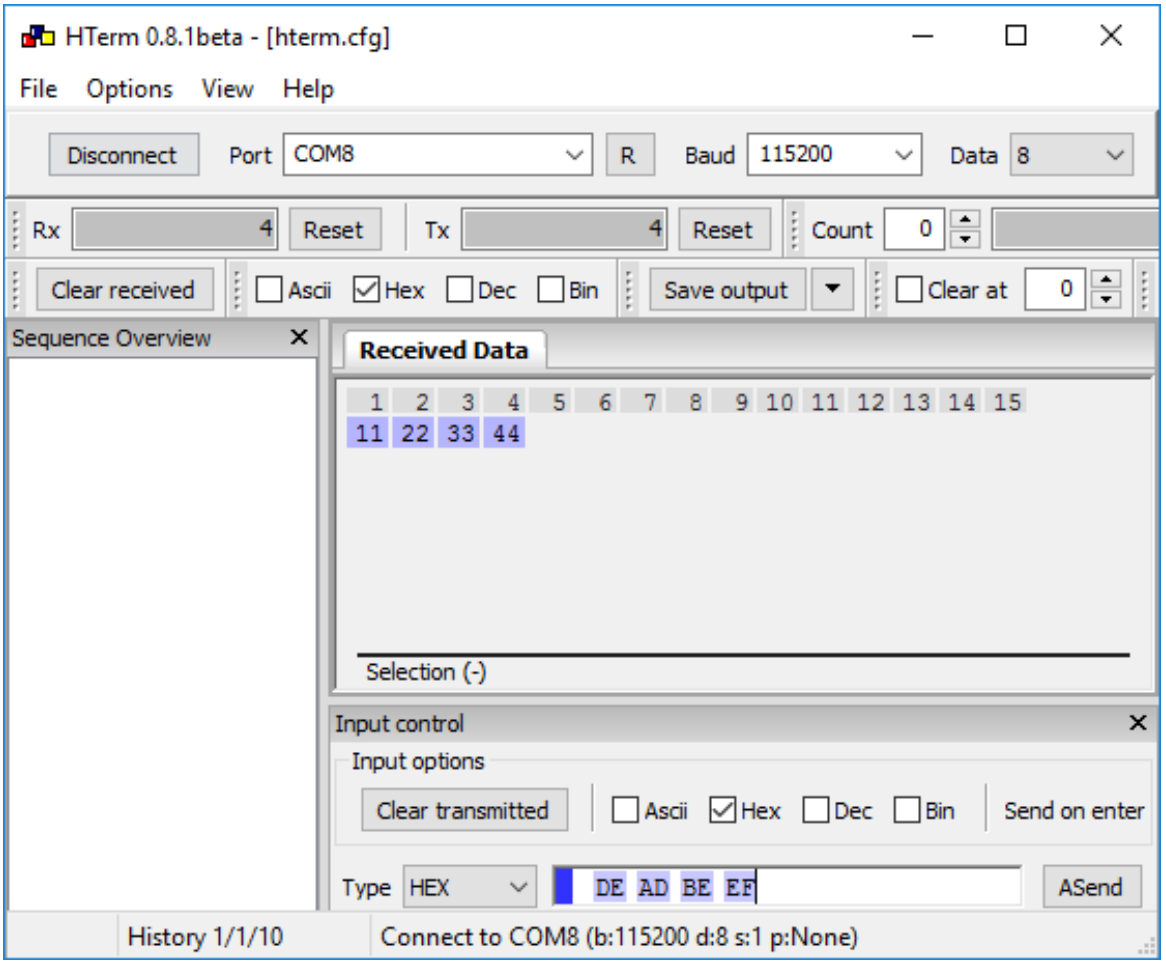
Android	iOS

- The payload that has been sent via radio is output by the Proteus-e via UART. In transparent mode, a transparent UART interface is used. This means, that only payload data is transmitted, without any packet header or footer. Thus, the transmitted bytes 0x11 0x22 0x33 0x44 are displayed on the connected terminal program.

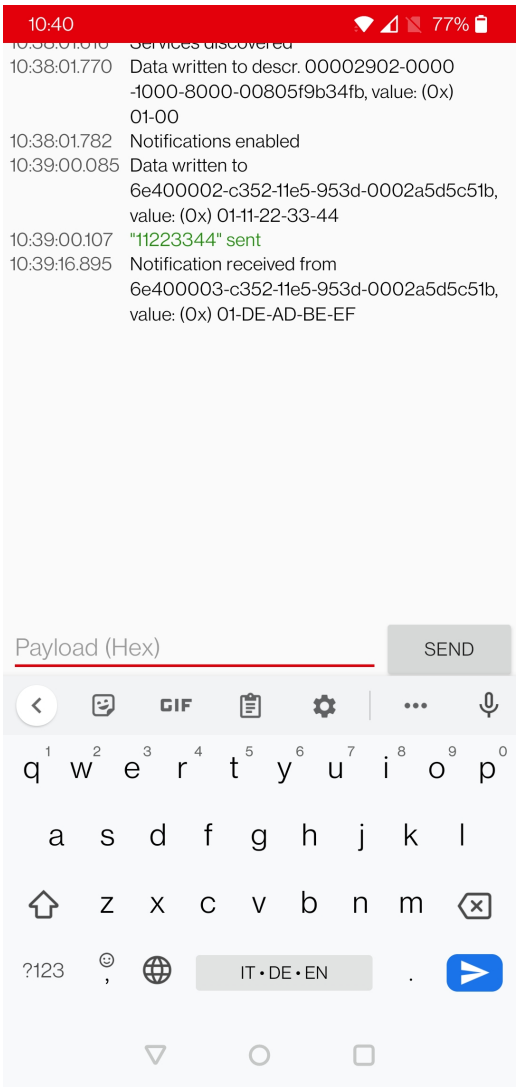


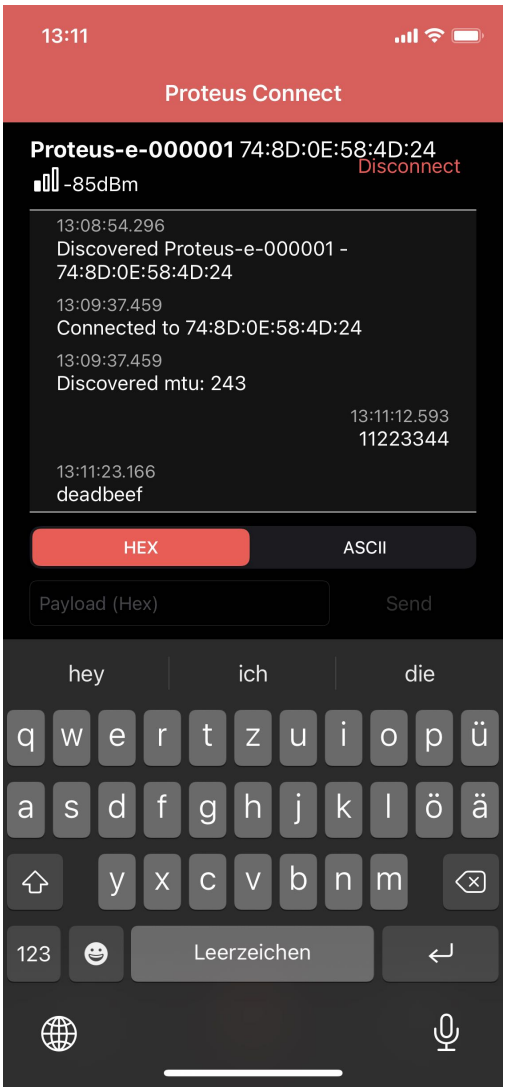
Android	iOS

- To send back data simply enter your payload in the respective terminal program field and press enter. In this example we choose 0xDE 0xAD 0xBE 0xEF. The header 0x01 will be automatically applied by the module and is not to be transmitted by the host.
- Here again the maximum payload size (MPS) must be respected.



Android	iOS
• The received data is shown in the status window. It contains the header byte 0x01 and the payload 0xDE 0xAD 0xBE 0xEF, that has been entered in the terminal program.	• The received data is shown in the status window.





4.2.1 Background service on iOS

By default, iOS disconnects the Bluetooth® LE connection, in case the Proteus Connect App is put to background. To avoid this behaviour, the background service of the Proteus Connect App must be enabled by going to the info tab and selecting the "Bluetooth Background Mode" slider.

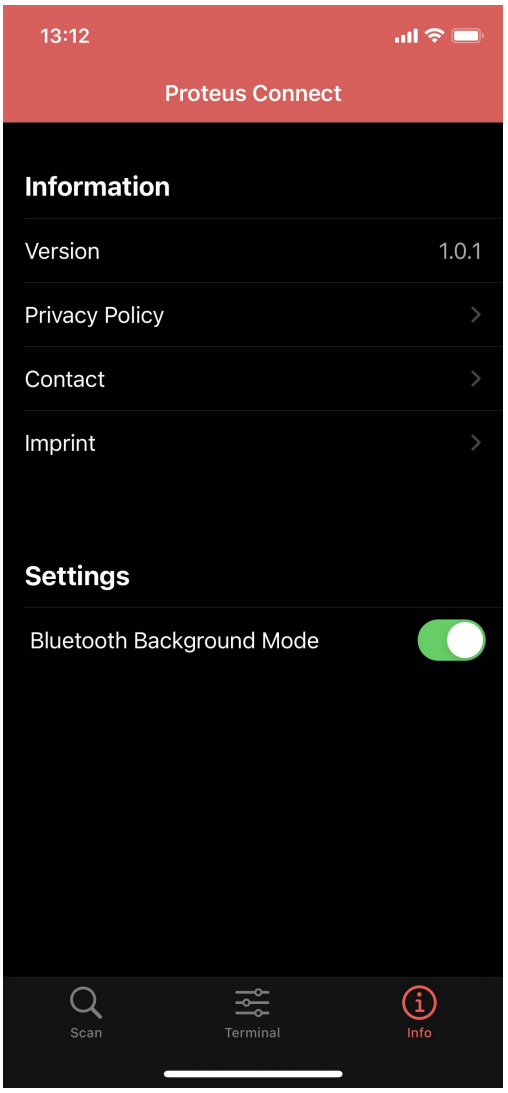


Figure 2: Enable the background service on iOS

4.3 Proteus module or USB radio stick as central device

This chapter describes how to setup a connection to the Proteus-e radio module in transparent mode, when another Proteus radio module or even Proteus USB radio stick is used as central device.

The Proteus-e does not support the role of central device.

For reasons of simplicity, we will call the Proteus radio module or USB radio stick that is intended to setup the connection to the Proteus module running in transparent mode, **Proteus_central**. Furthermore, we will call the Proteus-e module running in transparent mode, **Proteus_peripheral**.

Please note that the **Proteus_central** must run in command mode to initiate the connection setup.

In this example we assume that the MAC of the **Proteus_peripheral** is 0x0018DA000011.

1. Connect **Proteus_central** to the **Proteus_peripheral** via Bluetooth® LE.

Info	**Proteus_central**	**Proteus_peripheral**
⇒ Request `CMD_CONNECT_REQ` with `FS_BTMAC` of **Proteus_peripheral**	02 06 06 00 11 00 00 DA 18 00 D1	
⇐ Response `CMD_CONNECT_CNF`: Request understood, try to connect now	02 46 01 00 00 45	
⇐ Indication `CMD_CONNECT_IND`: Physical connection established successfully to the module with `FS_BTMAC` 0x11 0x00 0x00 0xDA 0x18 0x00	02 86 07 00 00 11 00 00 DA 18 00 50	
⇐ Channel opened successfully to the module with `FS_BTMAC` 0x11 0x00 0x00 0xDA 0x18 0x00 and maximum payload size of 0xF3 (243 Bytes) per packet	02 C6 08 00 00 11 00 00 DA 18 00 F3 EC	

2. Now the connection is active. Thus, data can be sent in each direction. Let us send a string "ABCD" from **Proteus_peripheral** to **Proteus_central**.

The RSSI values will be different in your tests.

Info	**Proteus_central**	**Proteus_peripheral**
⇒ Transparent send "ABCD" to **Proteus_central**		41 42 43 44
⇐ Indication `CMD_DATA_IND`: Received string "ABCD" from `FS_BTMAC` 0x11 0x00 0x00 0xDA 0x18 0x00 with RSSI of 0xCA (-54dBm)	02 84 0B 00 11 00 00 DA 18 00 CA 41 42 43 44 90	

3. Reply with "EFGH" to the **Proteus_peripheral**.

Info	**Proteus_central**	**Proteus_peripheral**
⇒ Request `CMD_DATA_REQ`: Send "EFGH" to **Proteus_peripheral**	02 04 04 00 45 46 47 48 0E	
⇐ Response `CMD_DATA_CNF`: Request received, send data now	02 44 01 00 00 47	
⇐ Transparent received string "EFGH"		45 46 47 48
⇐ Response `CMD_TXCOMPLETE_RSP`: Data transmitted successfully	02 C4 01 00 00 C7	

4. Now **Proteus_central** closes the connection.

Info	**Proteus_central**	**Proteus_peripheral**
⇒ Request `CMD_DISCONNECT_REQ`: Disconnect	02 07 00 00 05	
⇐ Response `CMD_DISCONNECT_CNF`: Request received, disconnect now	02 47 01 00 00 44	
⇐ Indication `CMD_DISCONNECT_IND`: Connection closed	02 87 01 00 16 92	

5 Command mode: Quickstart

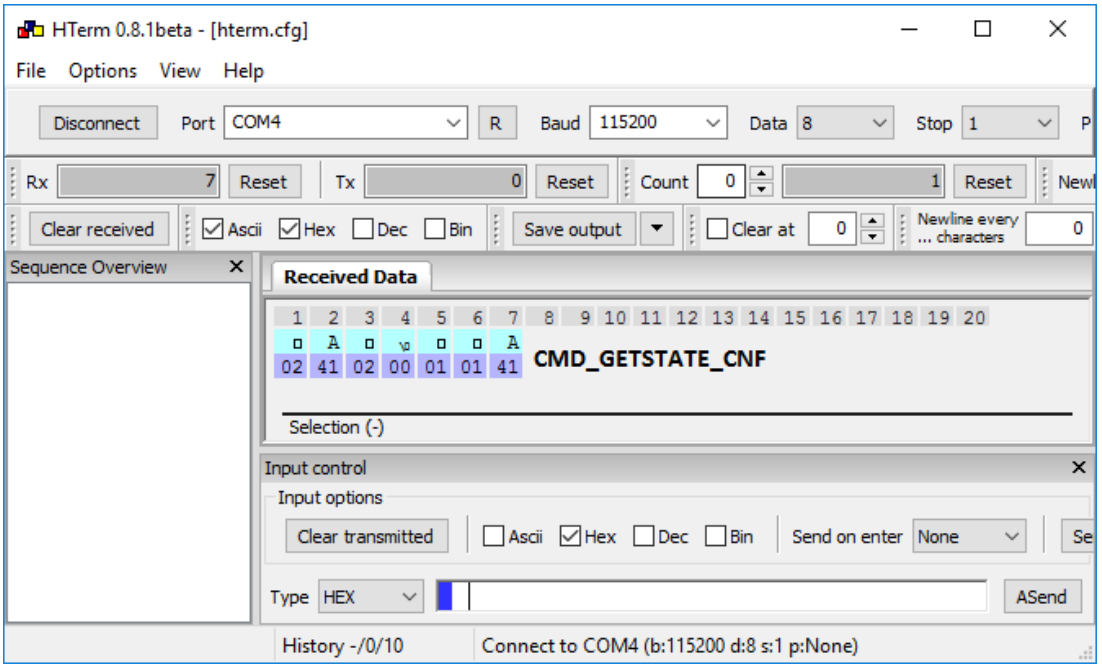
In chapter 3.2 it has been described which steps have to be performed by the central device to setup a connection to a Proteus-e radio module running in **command mode**. What this means in practice will be shown in this chapter.

5.1 Smart phone using nRFConnect app as central device

This chapter describes how to setup a connection to the Proteus module in command mode, when a smart phone and the **nRF Connect** App [1][2] are used. Please perform the following steps:

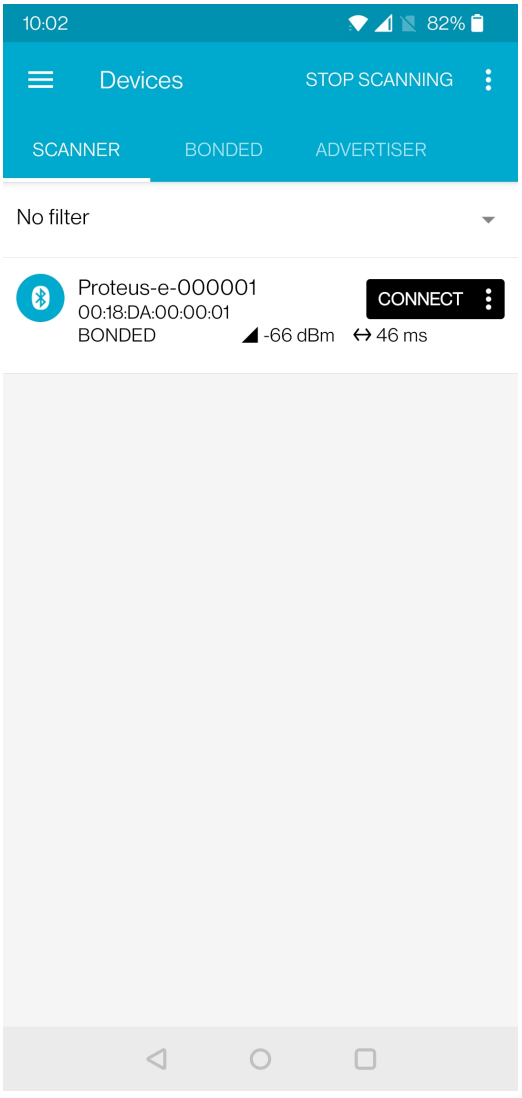
Android	iOS

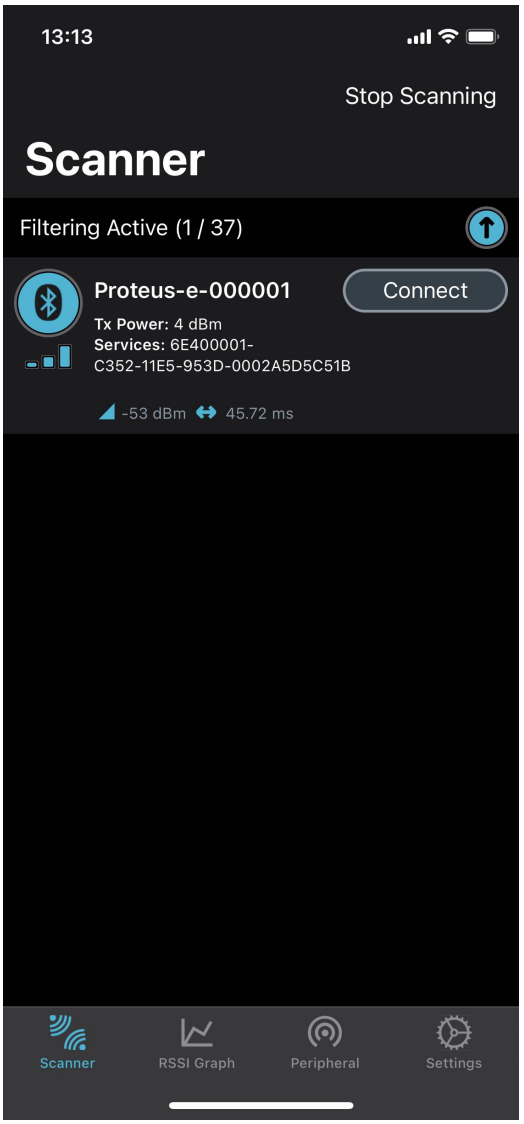
- Connect the Proteus evaluation board to a host.
  In this application note, we assume that a Windows PC and the terminal program *hterm* is used.
  To make life easy, also the SmartCommander PC tool provided by Würth Elektronik eiSos can be used. This tool implements all commands of the radio module.
- Open the terminal program using the Proteus-e default UART settings (115200 Baud, 8n1).
- Press the reset button on the evaluation board. The module outputs a `CMD_GETSTATE_CNF` (0x02410200010141) message to indicate that it is ready for operation.



Android	iOS

- Initially, the module is advertising. Thus, one LED of the Proteus evaluation 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.

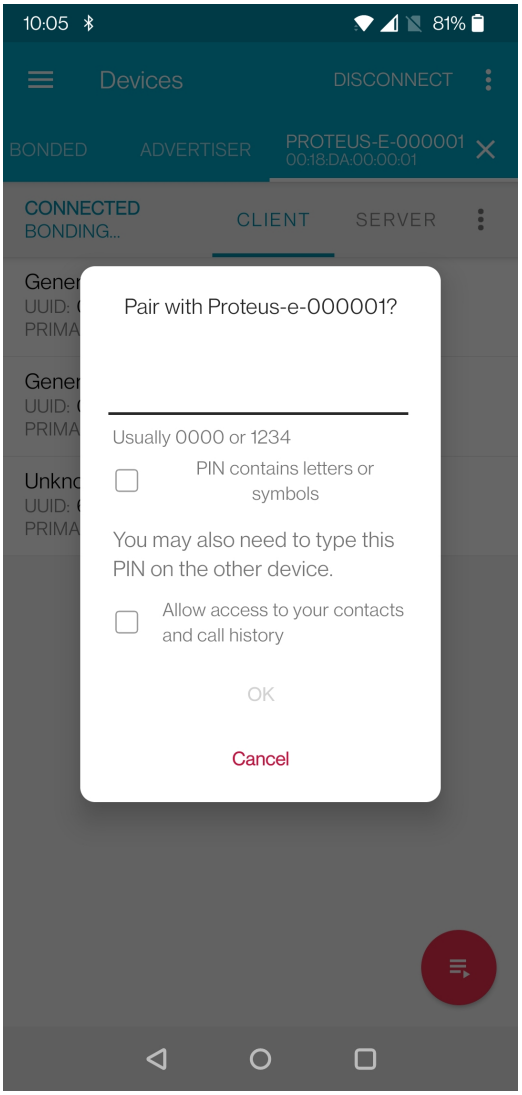


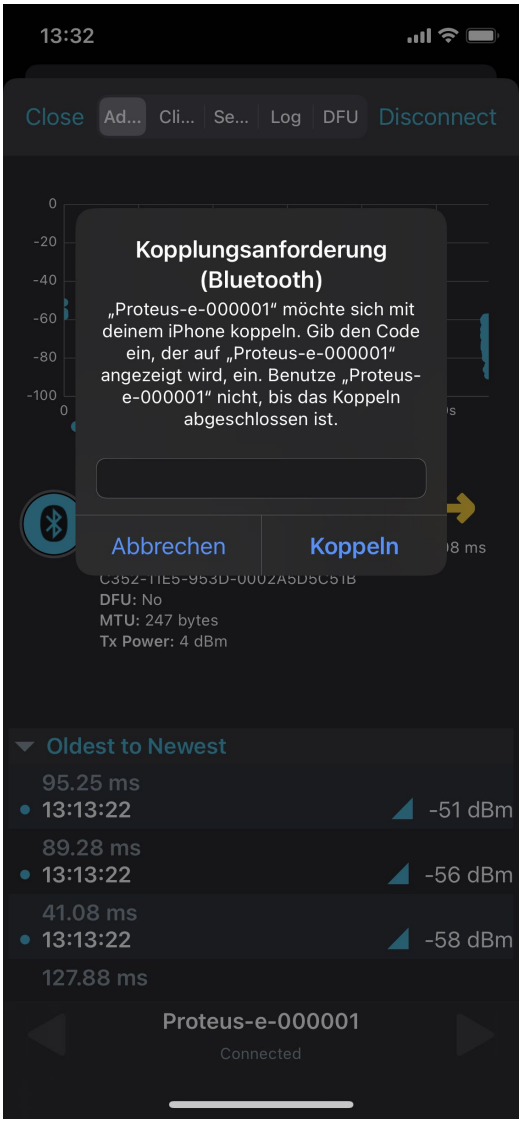


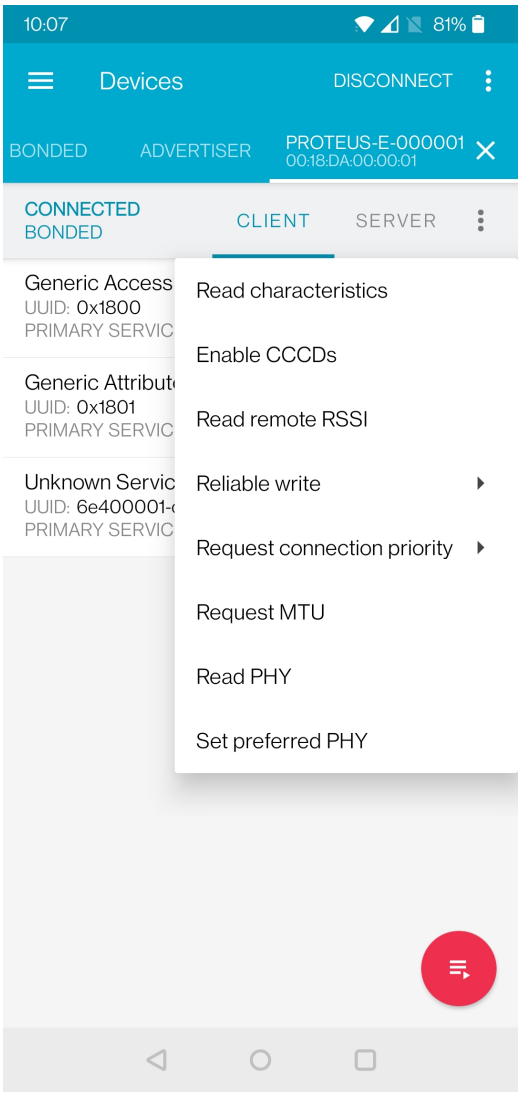
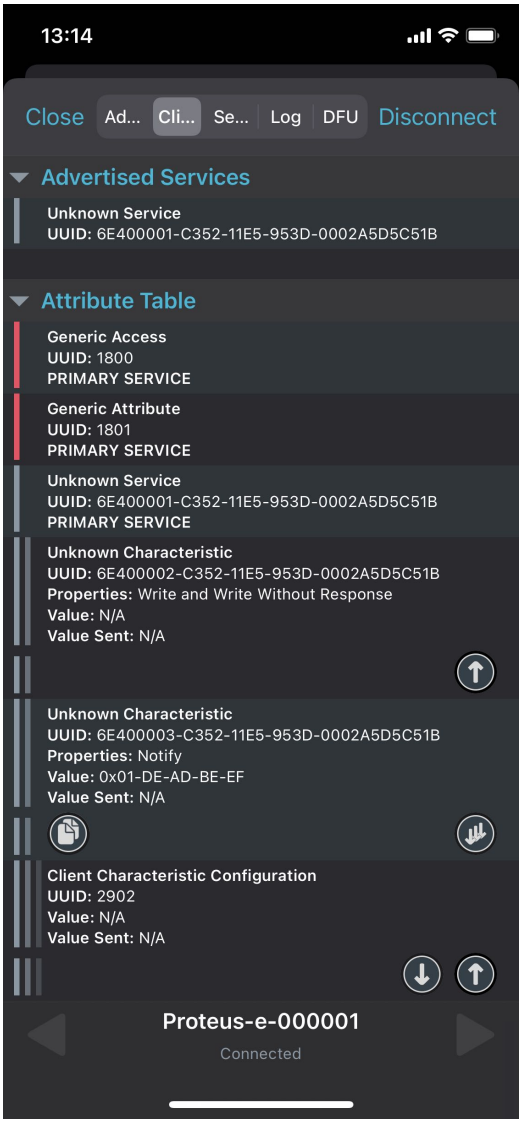
- When the module appears, press "CONNECT".

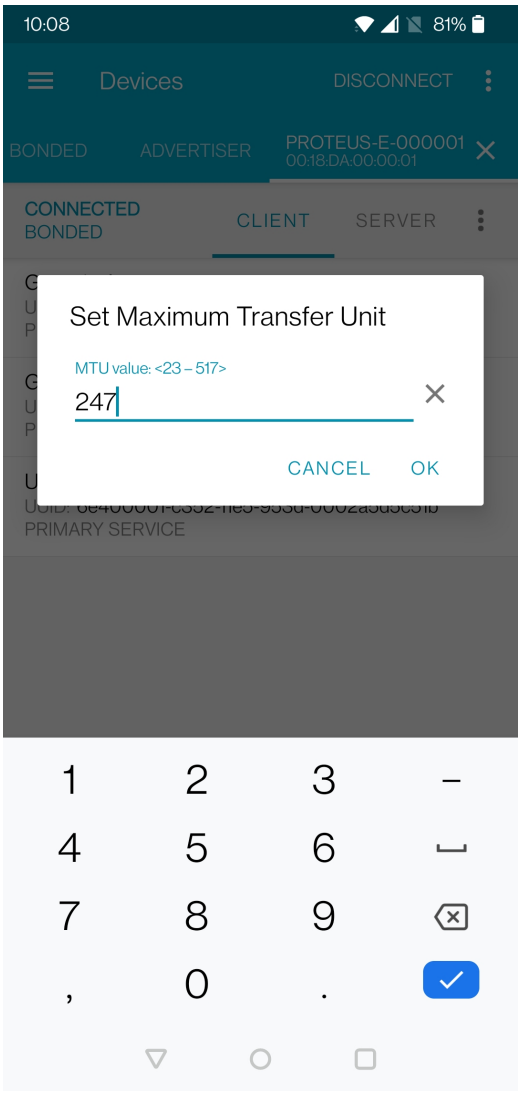
Android	iOS

- As soon as the module has received the connection request from the smart phone the blinking LED will blink faster.
- 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. When 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.

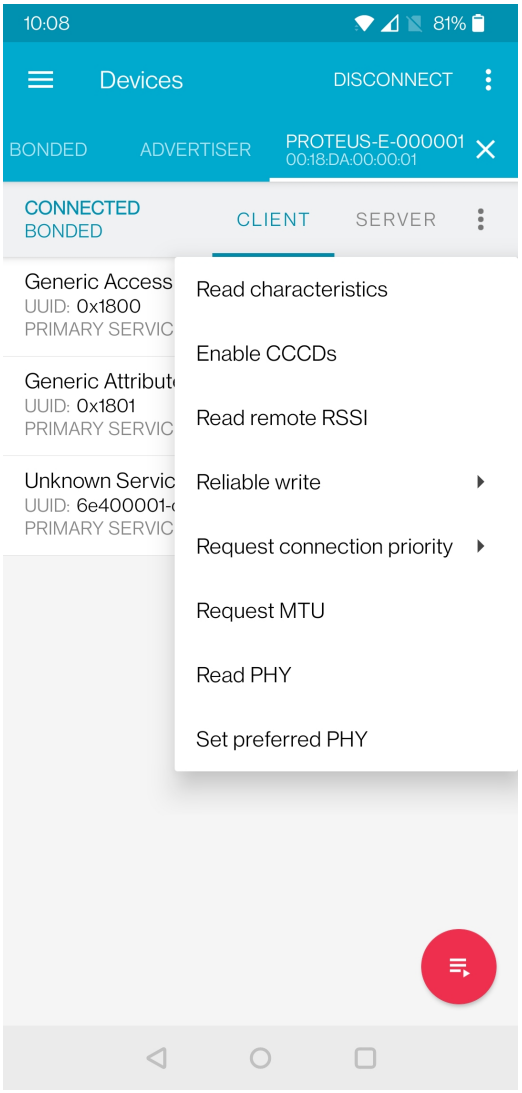


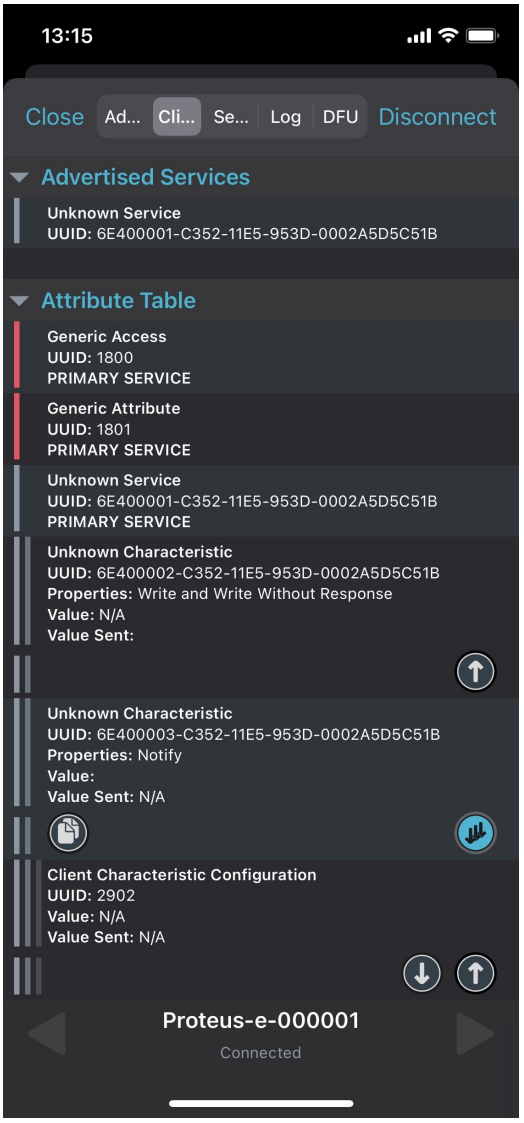


Android	iOS
• Please click on the menu bullets on the right and press "Request MTU" to request for a larger MTU.	• Please click on the "Unknown Service" to start the service discovery and the MTU request.
	

Android	iOS
• The Proteus module allows a MTU of up to 247 bytes, which results in a maximum payload size (MPS) 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"/"Enable CCCDs" to enable the notifications.	• Press the arrows on the RX-characteristic `6E400003- C352-11E5- 953D -0002A5D5C51B` to enable the notifications. Press it until the symbol turns blue (see below, it has to be pressed at least once). If it is already blue press it twice such that it is deselected and selected again.

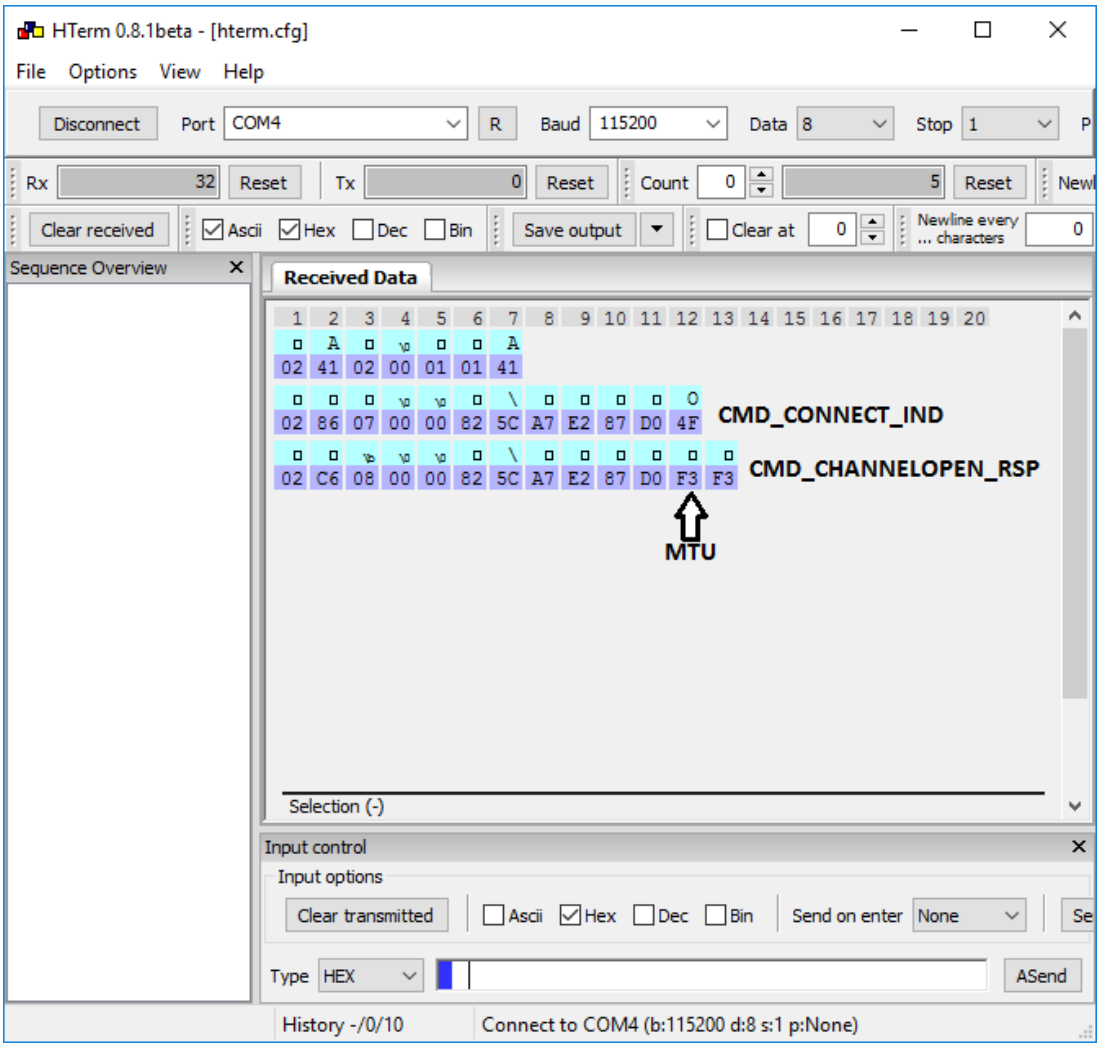




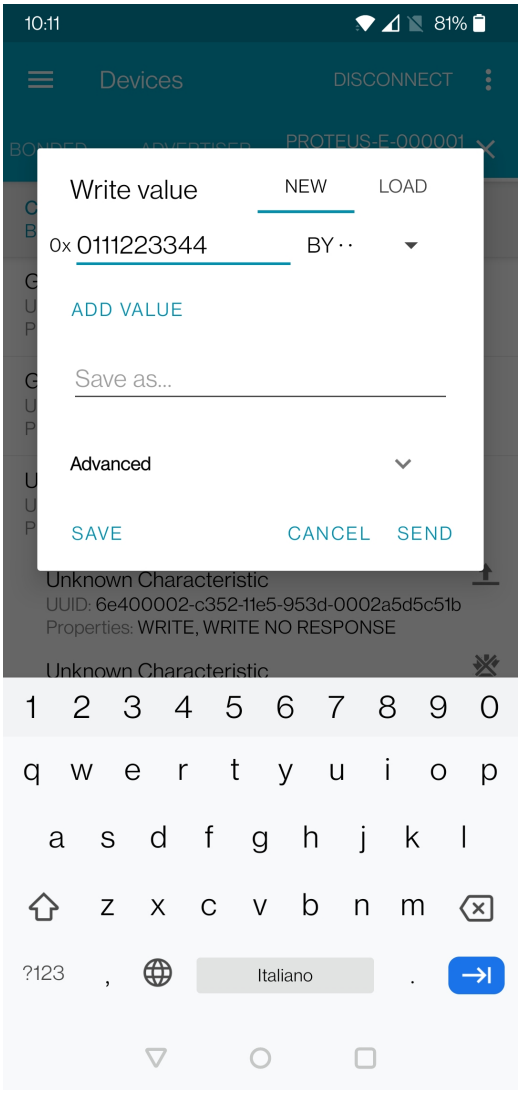
- As soon as the module has received the notification enable request, the LED on the evaluation board is turned static on. Now you are fully connected and you can access the characteristics to transmit and receive data.

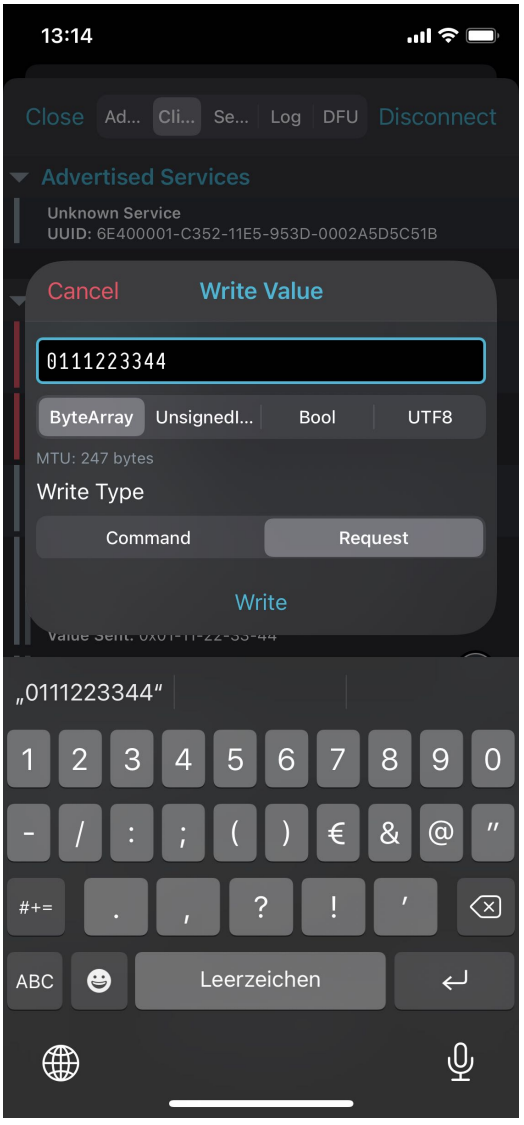
Android	iOS

- On the Proteus side, the radio module sent the corresponding `CMD_CONNECT_IND` (0x02860700...) and `CMD_CHANNELOPEN_RSP` (0x02C60800...) in between. These messages indicate that a connection has been setup and a link has been opened. The `CMD_CHANNELOPEN_RSP` message contains the MPS (maximum payload size) of the current link. In this example it is 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.	

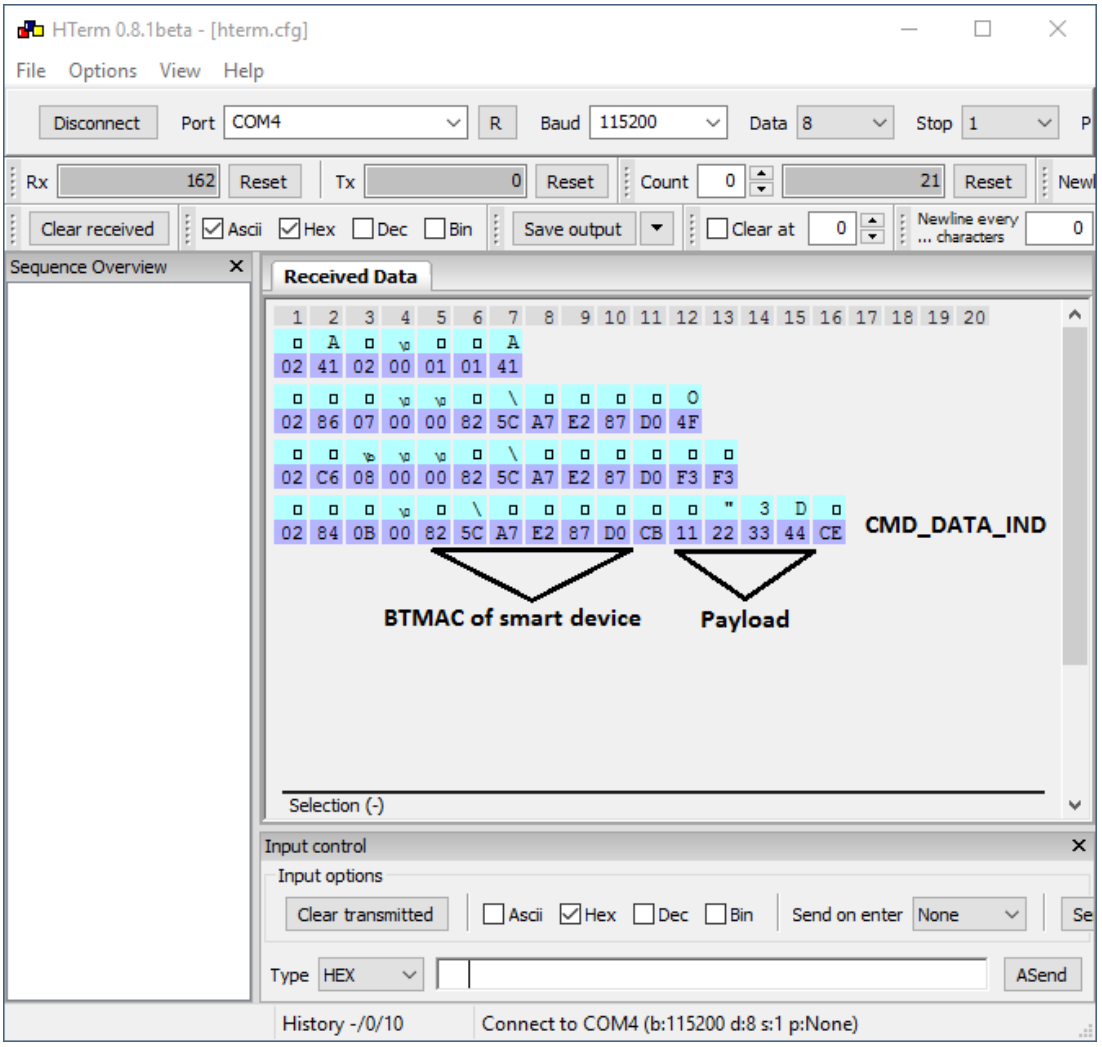




Android	iOS

- 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	0x82 0x5C 0xA7 0xE2 0x87 0xD0	0XCB	0x11 0x22 0x33 0x44	0xCE

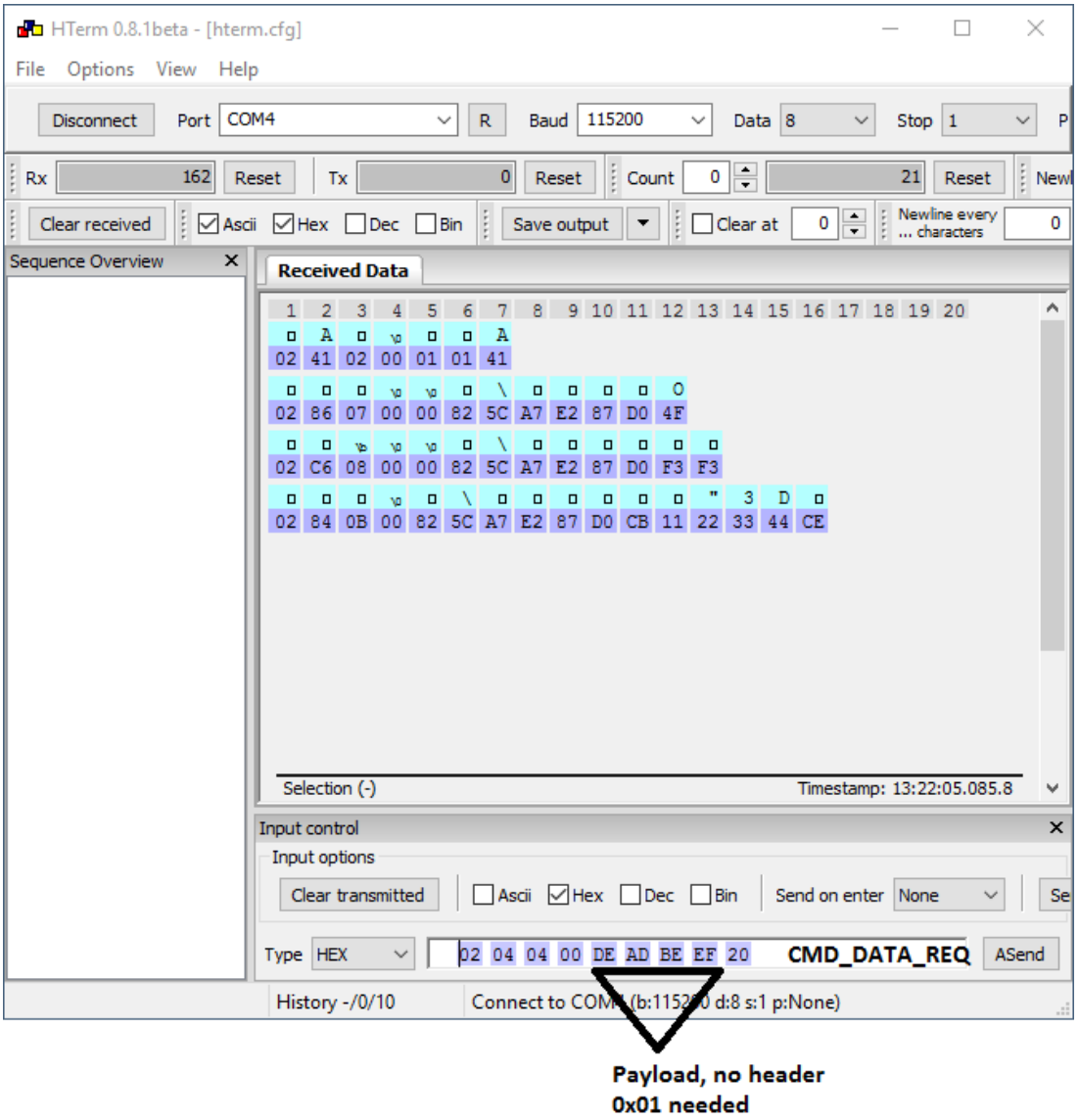


Android	iOS

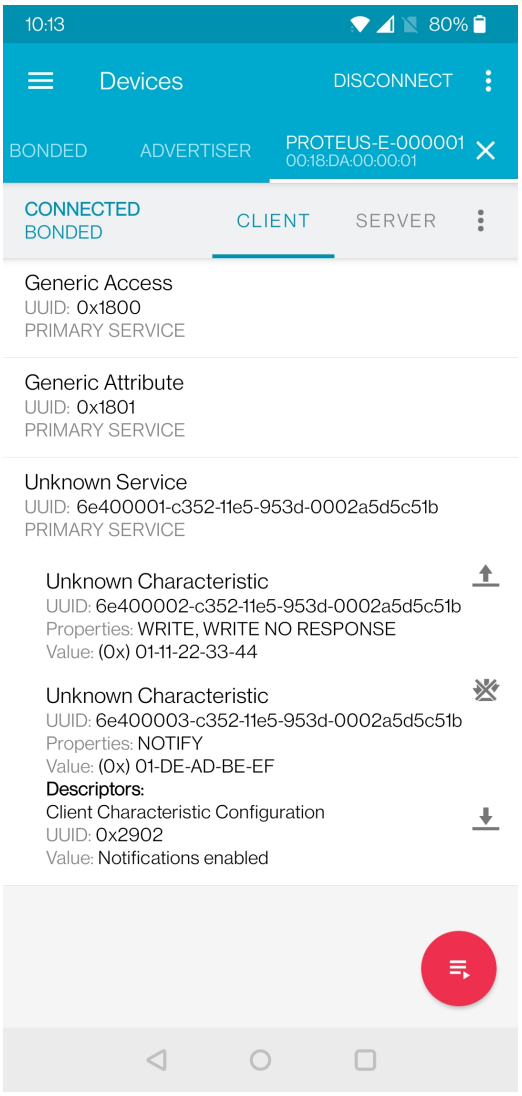
- To send back data to the smart phone simply insert your payload (here we choose 0xDE 0xAD 0xBE 0xEF) in a `CMD_DATA_REQ` message. The format of the `CMD_DATA_REQ` message is as follows, where the check sum (CS) is calculated as XOR of the preceding bytes:

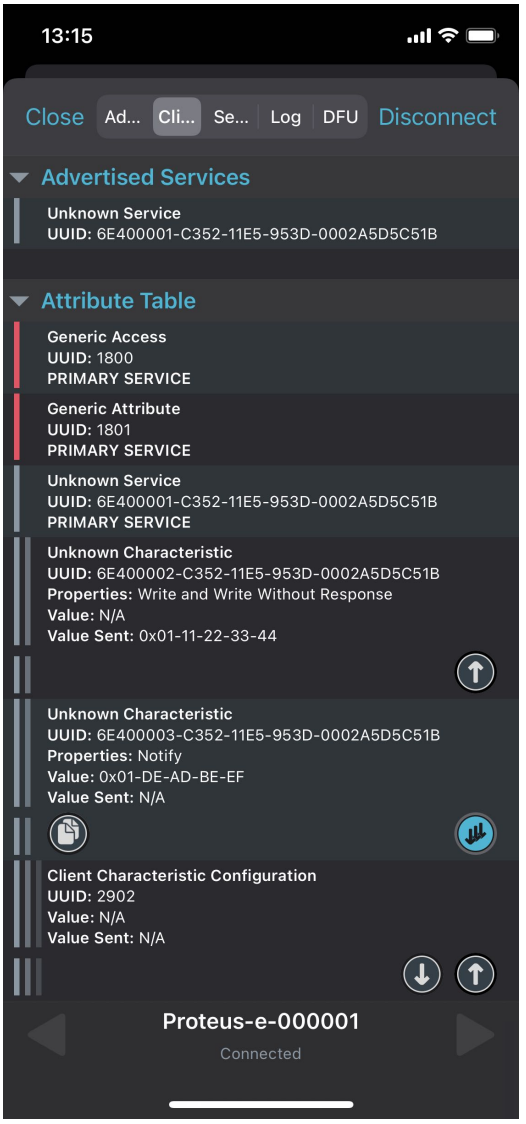
Start signal	Command	Length	Payload	CS
0x02	0x04	2 Bytes	Length Bytes	1 Byte
0x02	0x04	0x04 0x00	0xDE 0xAD 0xBE 0xEF	0x20

- 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.



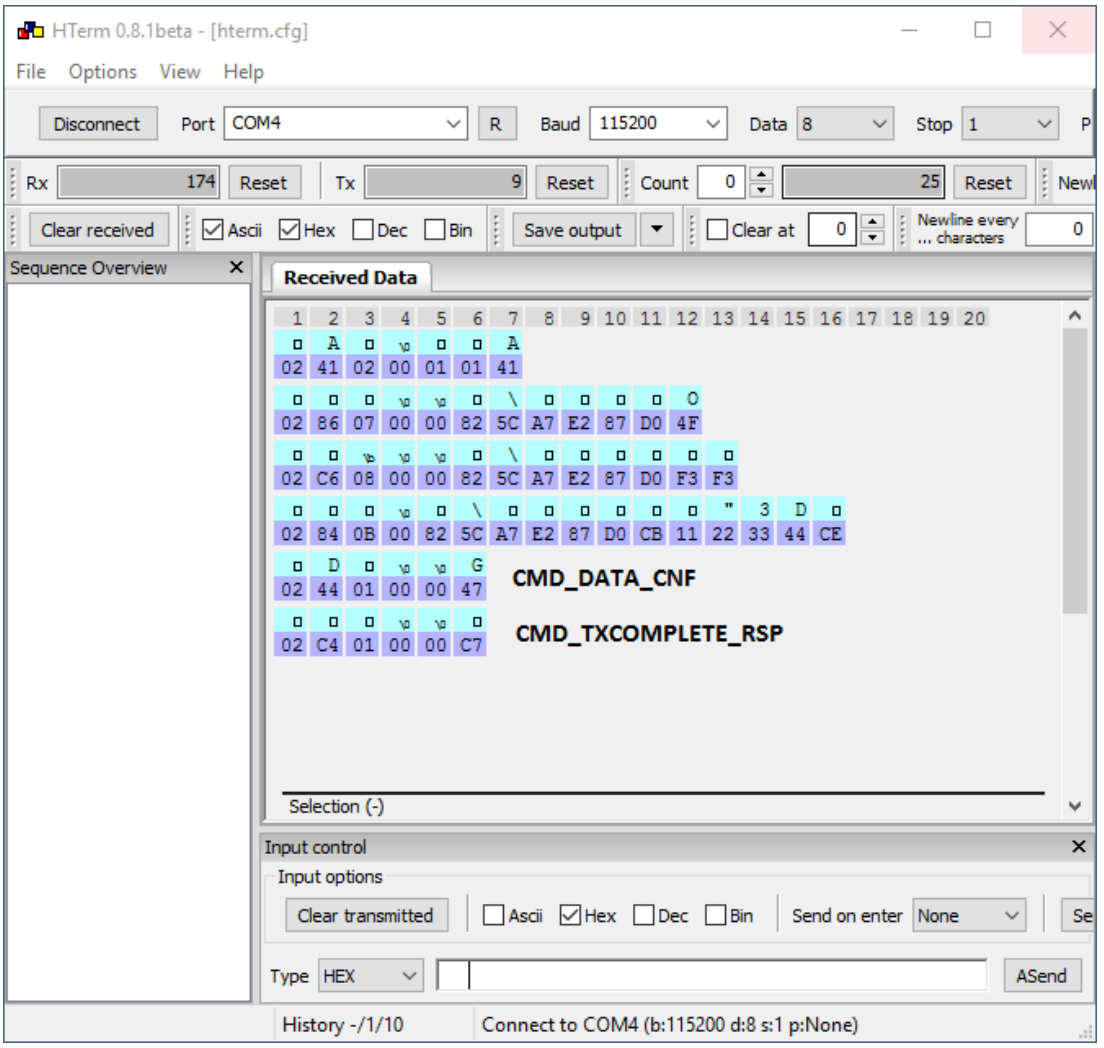
Android	iOS
• The received data can be found in the RX-characteristic `6E400003-C352-11E5-953D-0002A5D5C51B`. It contains the header byte 0x01 and the payload 0xDE 0xAD 0xBE 0xEF.	





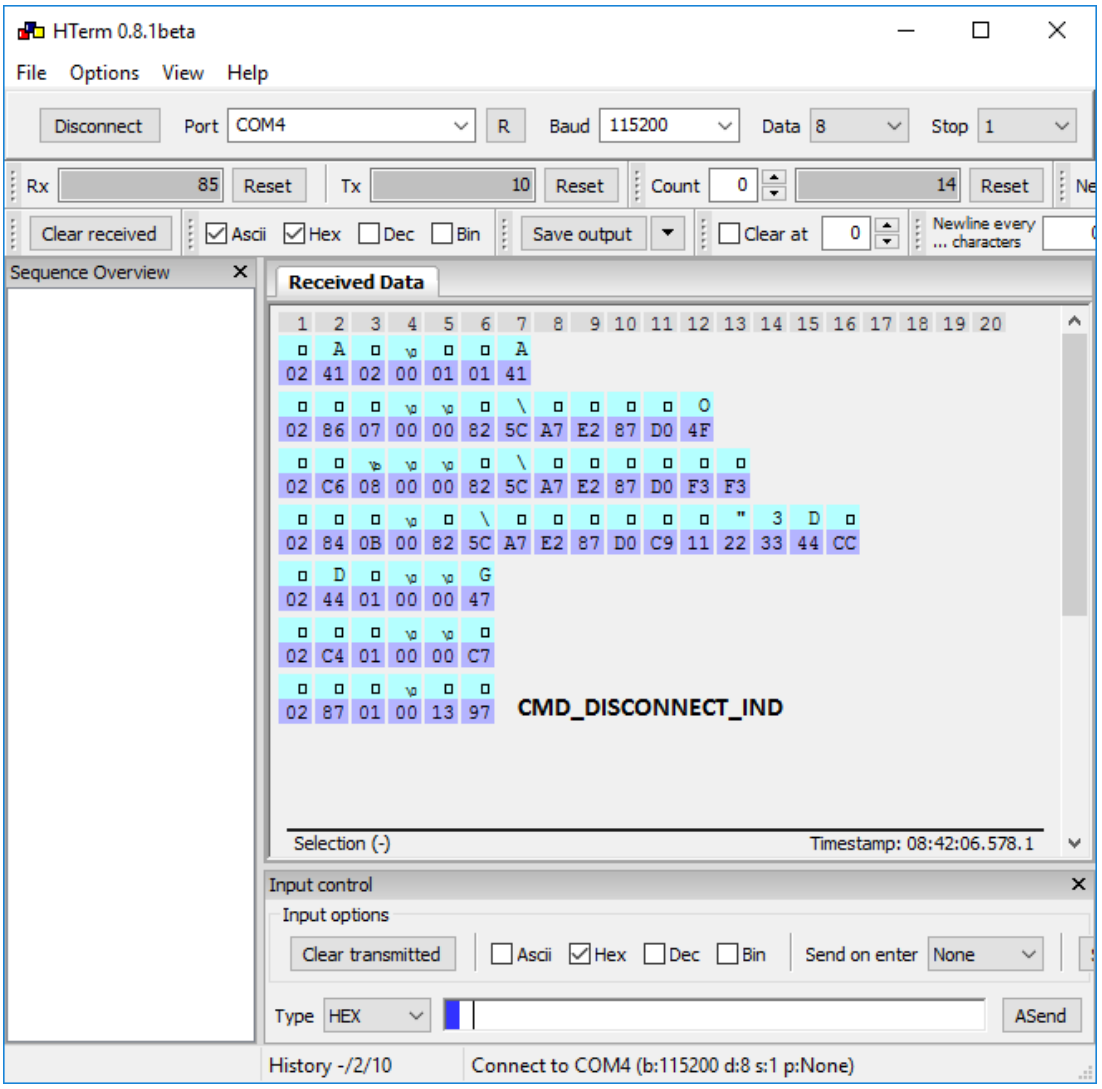
Android	iOS

- When sending the CMD_DATA_REQ to the Proteus module, it responds with two different messages. First, a CMD_DATA_CNF (0x024401000047) message is returned, as soon as the request was interpreted. Then a CMD_TXCOMPLETE_RSP (0x02C4010000C7) 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` (0x028701001397) message to indicate that the connection has been closed.



- After disconnection, the Proteus module starts advertising again, such that a reconnection can be performed.

5.2 Smart phone using Proteus Connect app as central device

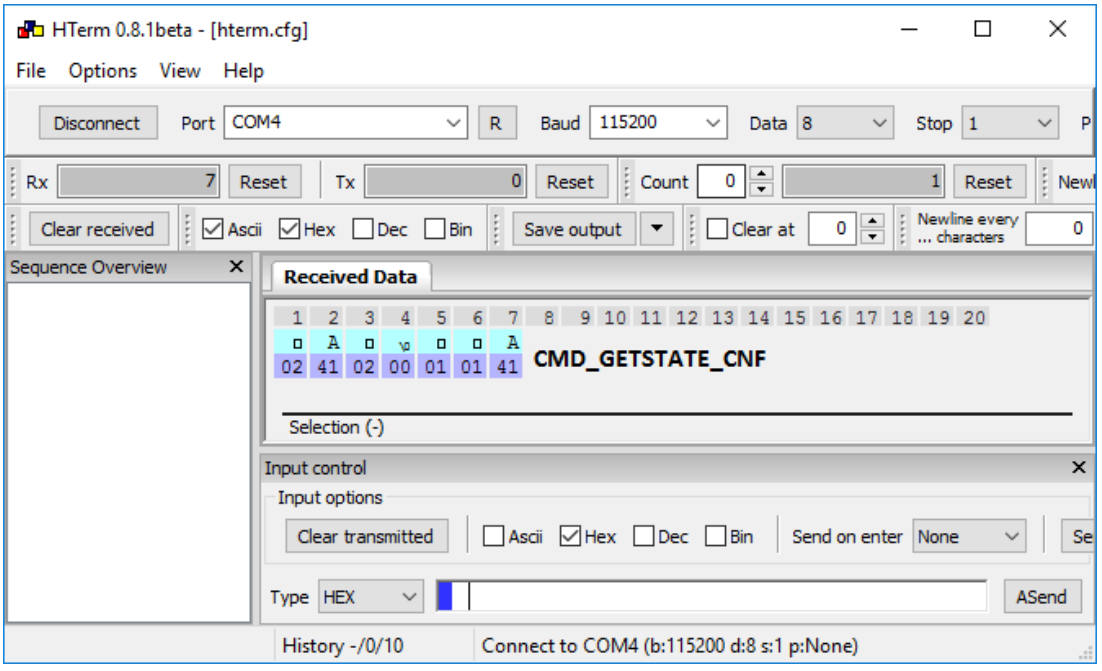
This chapter describes how to setup a connection to the Proteus-e in command mode, when a smart phone and the Proteus Connect App are used.

The Proteus Connect App for iOS and Android is provided by Würth Elektronik eiSos as executable [4][5] as well as source code [6][7].

Please perform the following steps:

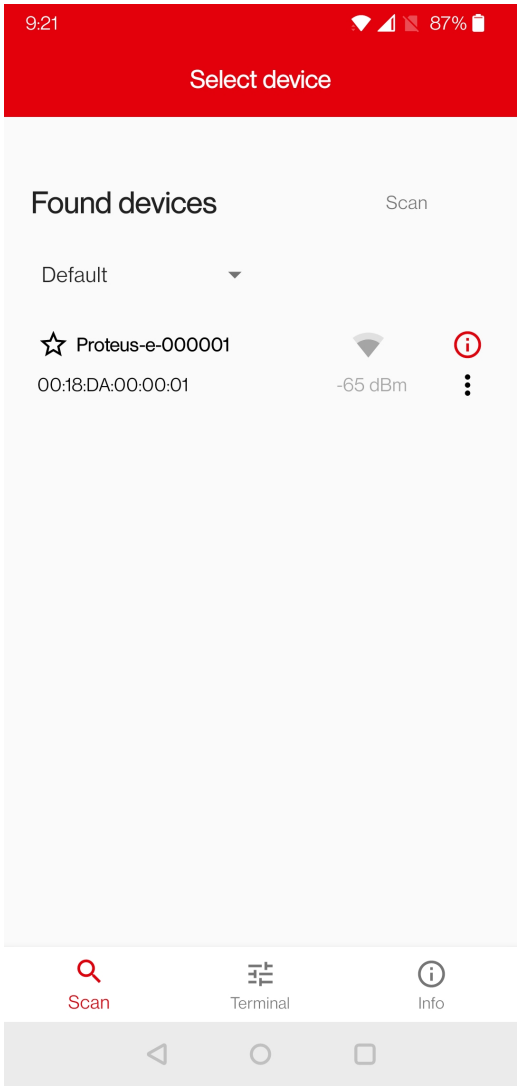
Android	iOS

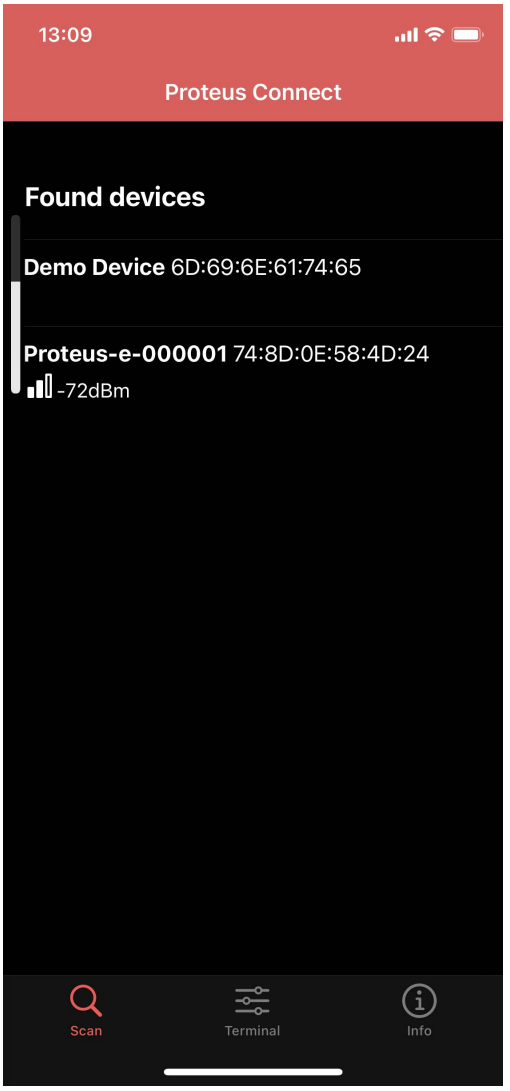
- Connect the Proteus evaluation board to a host.
  In this application note, we assume that a Windows PC and the terminal program *hterm* is used.
  To make life easy, also the SmartCommander PC tool provided by Würth Elektronik eiSos can be used. This tool implements all commands of the Proteus-e.
- Open the terminal program using the default UART settings (115200 Baud, 8n1).
- Press the reset button on the Proteus evaluation board. The Proteus module outputs a `CMD_GETSTATE_CNF` (0x02410200010141) message to indicate that it is ready for operation.



Android	iOS

- Initially, the module is advertising. Thus, one LED of the evaluation board is blinking.
- Start your smart phone, enable the Bluetooth® LE and location feature and start the **Proteus Connect** App.
- Press "Scan" to find the module on the radio.

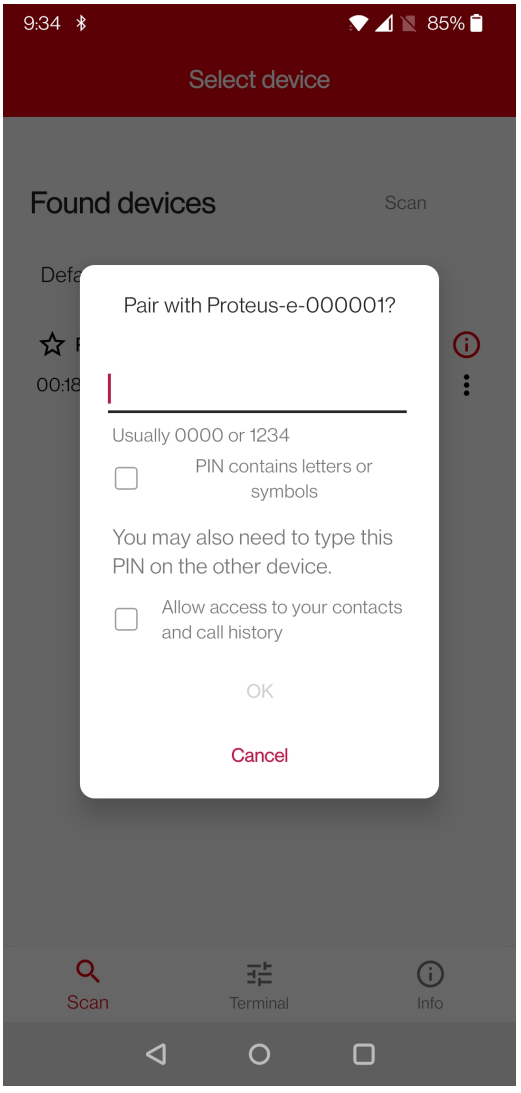




- When the module appears, select it to start the connection process.
- As soon as the module has received the connection request the module *LED_1* blinks faster.

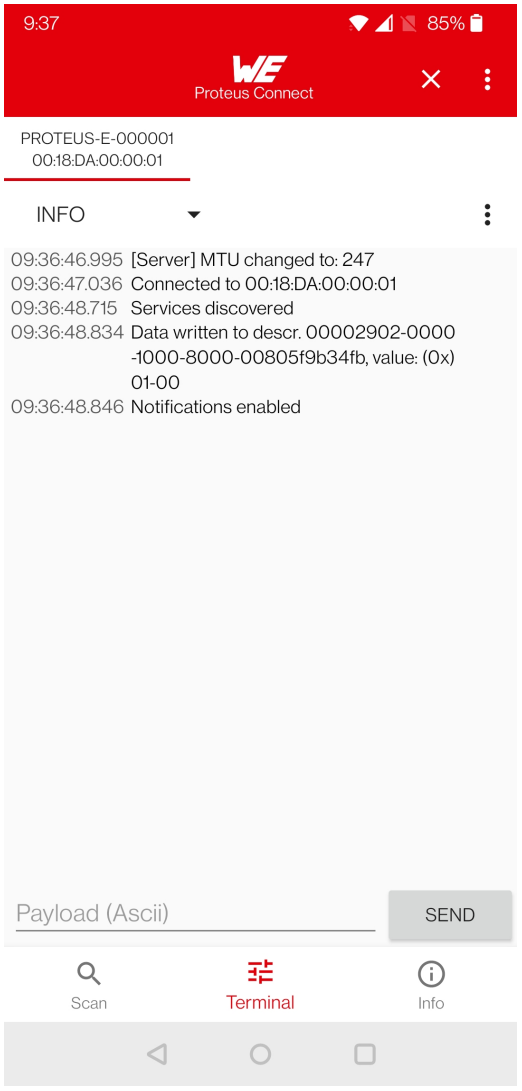
Android	iOS

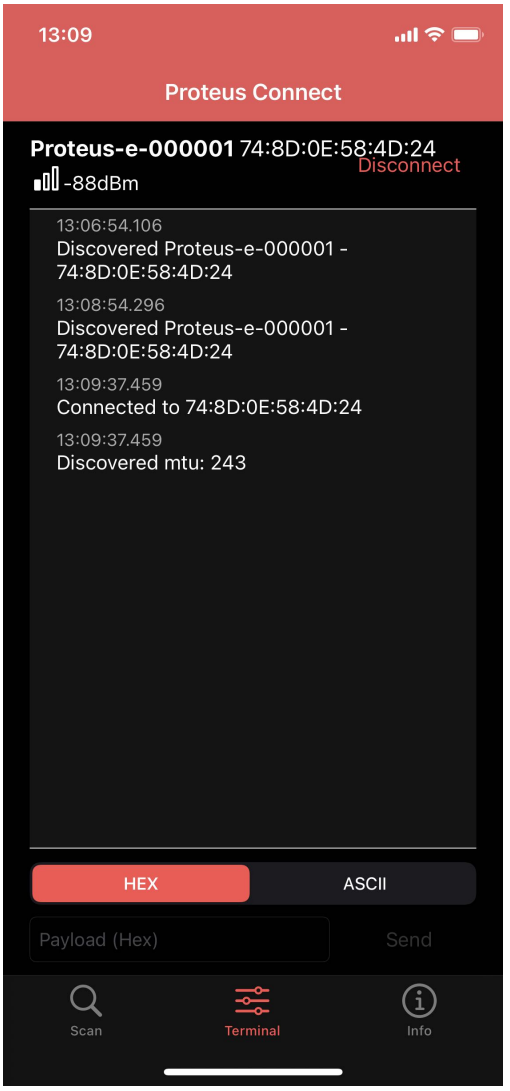
- Optional pairing: In case a security mode has been configured before, the smart phone requests the user for pairing actions. In case of the static passkey authentication, the Proteus requests to enter the static passkey. The default passkey is "123123". The Bluetooth® coupling requirement pop-up is shown on your smart phone. When 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.



In few cases, the Android may show an "authentication timeout" pop-up message, when entering the key. In this case, please proceed entering the key and simply do a reconnect. On this reconnect, the entered key information is reused and the connection is opened.

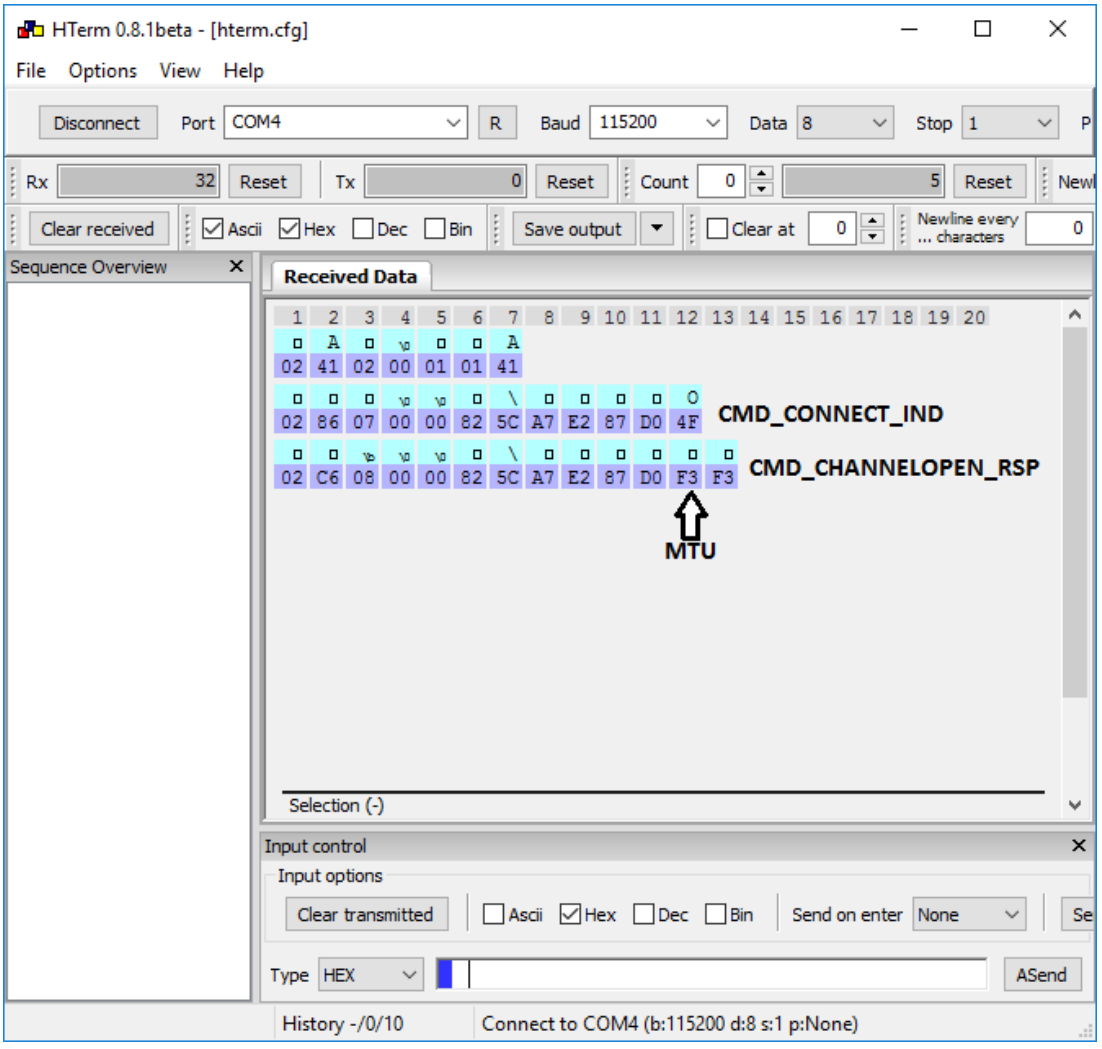
Android	iOS
• Now you are authenticated and the *LED_1* is turned static on. Now data can be transmitted in both directions.	





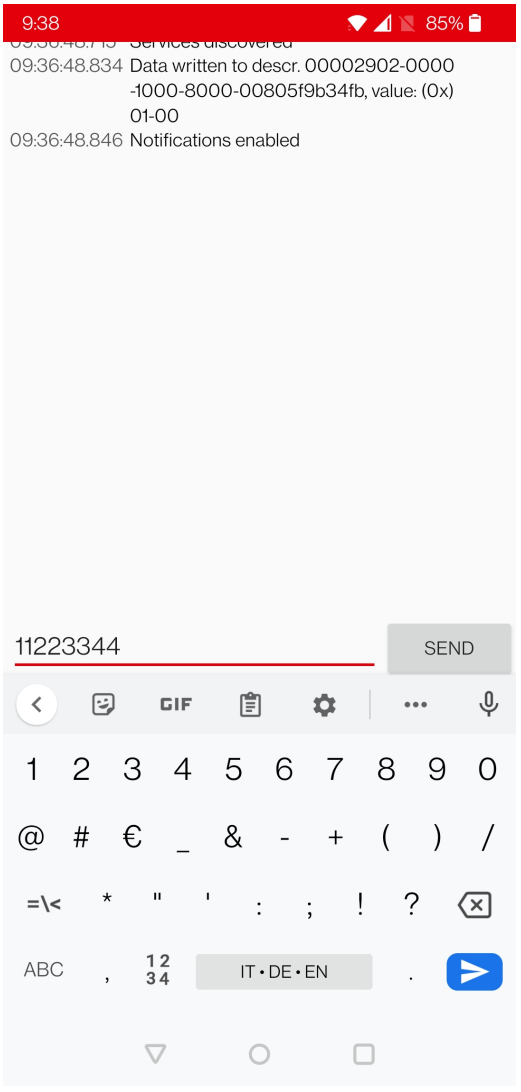
Android	iOS

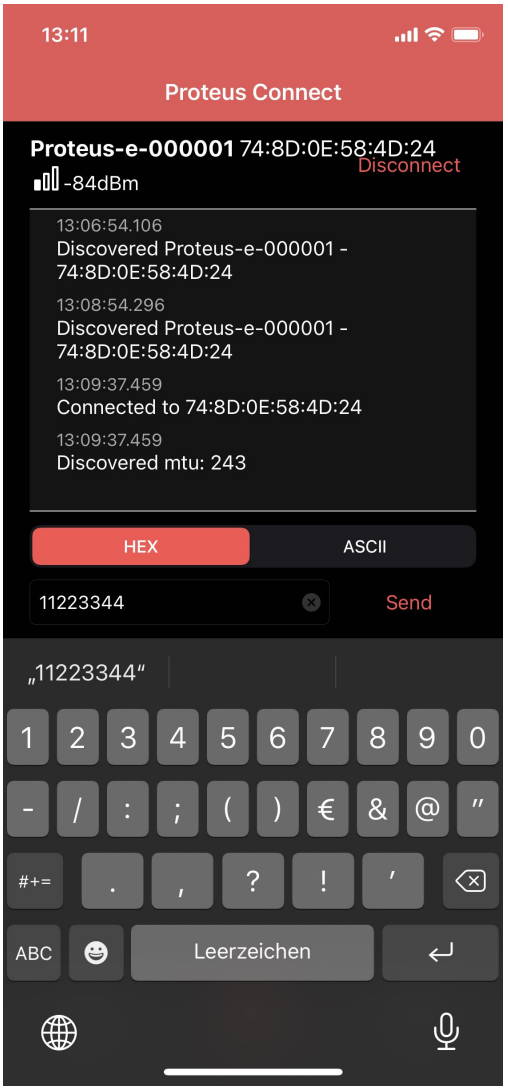
- On the Proteus-e side, the radio module sent the corresponding `CMD_CONNECT_IND` (0x02860700...) and `CMD_CHANNELOPEN_RSP` (0x02C60800...) in between. These messages indicate that a connection has been setup and a link has been opened. The `CMD_CHANNELOPEN_RSP` message contains the MPS (maximum payload size) of the current link. In this example it is 0xF3 (243_{dec}) bytes payload per packet.



Android	iOS

- First of all, 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". The allowed payload size is dependent on the MPS that was negotiated in the connection process. The smallest supported MTU for all Bluetooth® 4.0 (or newer) devices results in a max payload size (MPS) of 19 bytes. iOS and Android usually allow up to 243 bytes.

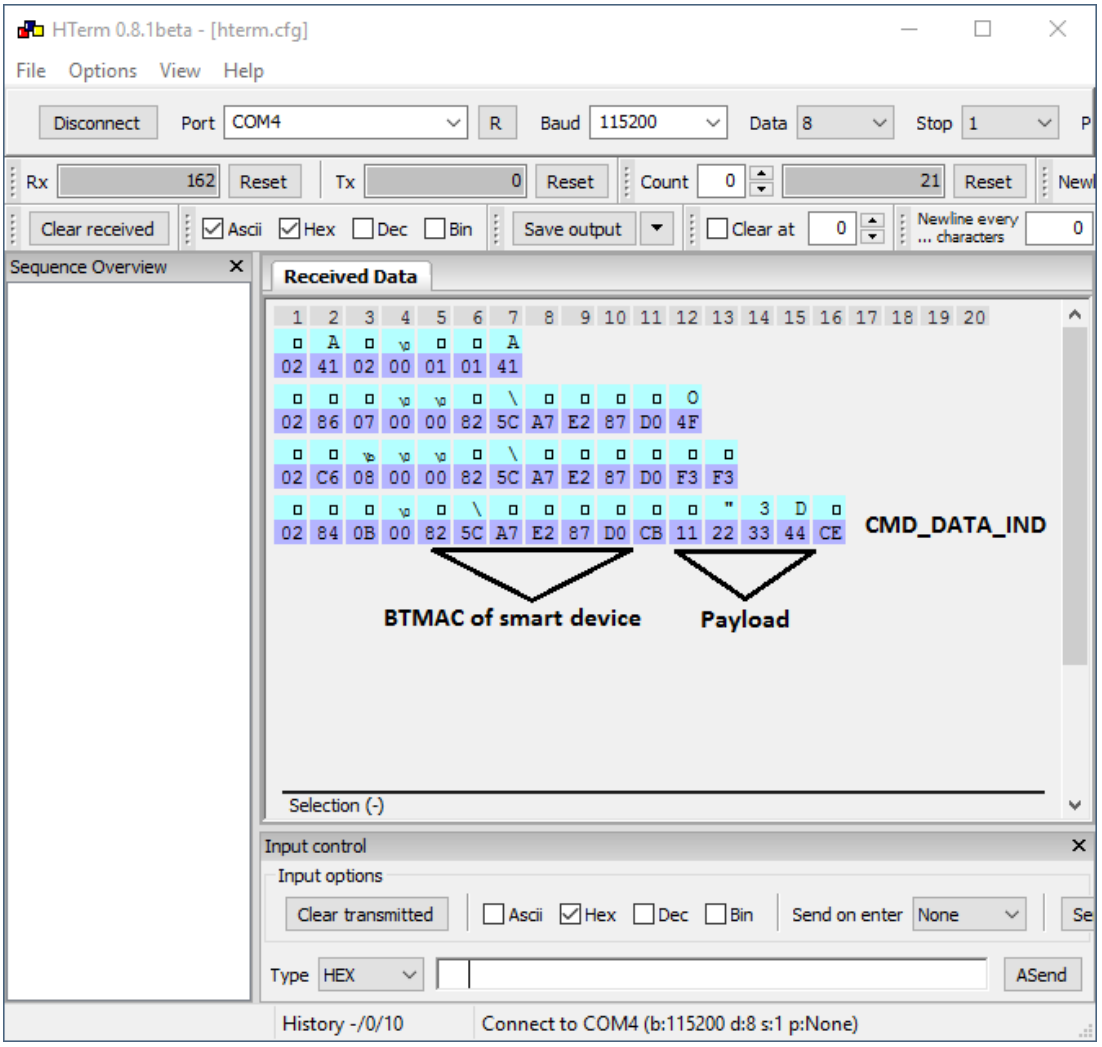




Android	iOS

- 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	0x82 0x5C 0xA7 0xE2 0x87 0xD0	0XCB	0x11 0x22 0x33 0x44	0xCE

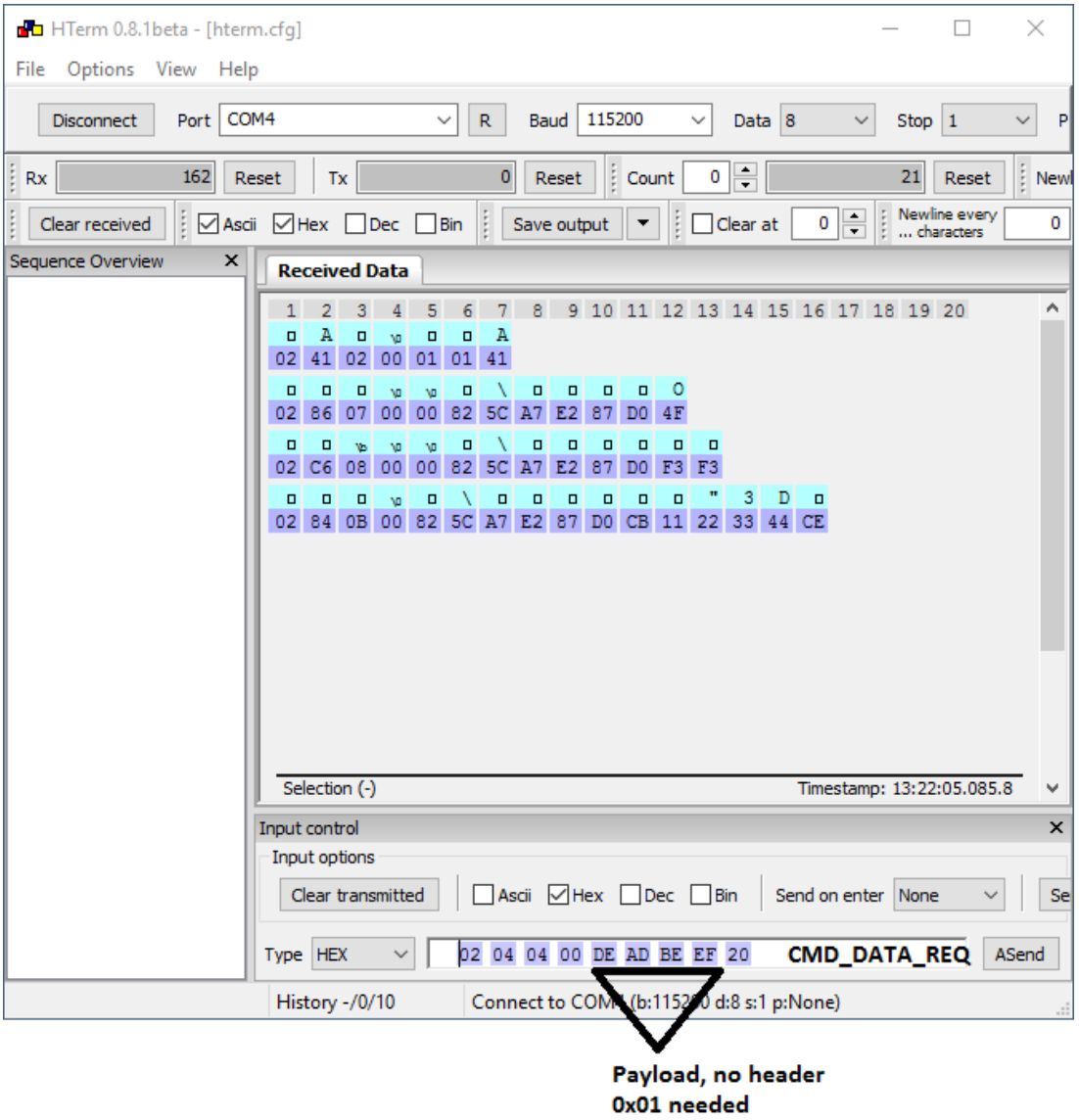


Android	iOS

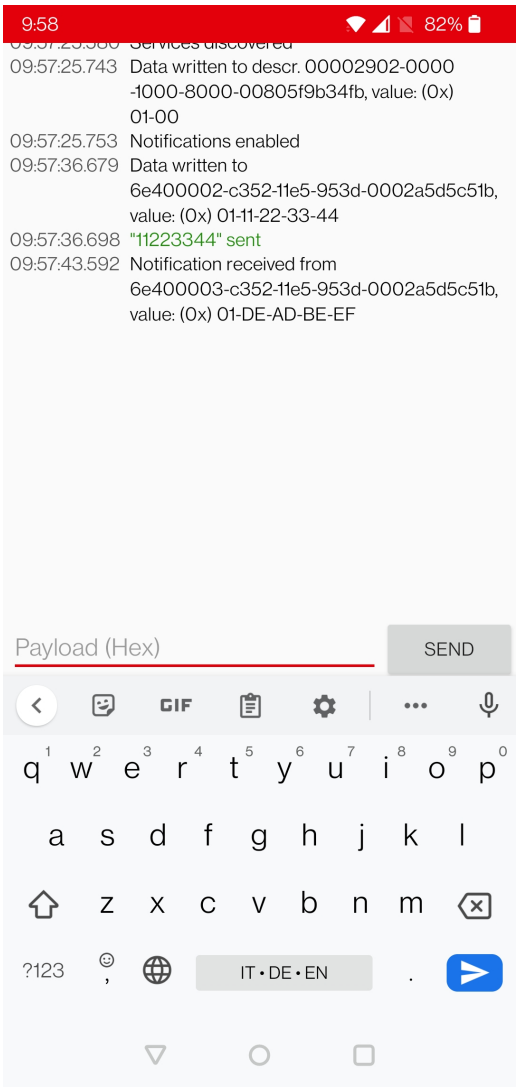
- To send back data to the smart phone simply insert your payload (here we choose 0xDE 0xAD 0xBE 0xEF) in a `CMD_DATA_REQ` message. The format of the `CMD_DATA_REQ` message is as follows, where the check sum (CS) is calculated as XOR of the preceding bytes:

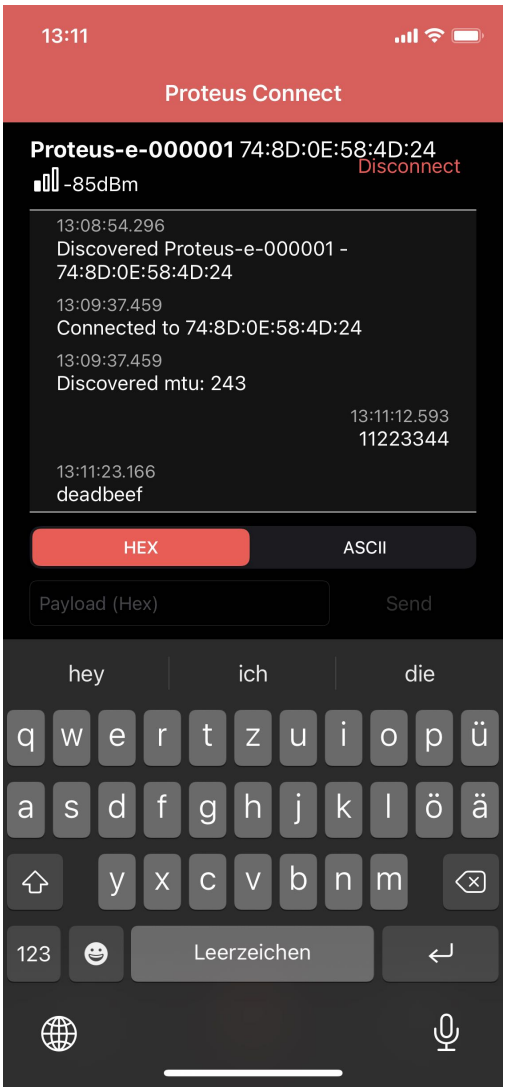
Start signal	Command	Length	Payload	CS
0x02	0x04	2 Bytes	Length Bytes	1 Byte
0x02	0x04	0x04 0x00	0xDE 0xAD 0xBE 0xEF	0x20

- 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.



Android	iOS
• The received data is shown in the status window. It contains the header byte 0x01 and the payload 0xDE 0xAD 0xBE 0xEF, that has been entered in the terminal program.	• The received data is shown in the status window.





5.2.1 Background service on iOS

By default, iOS disconnects the Bluetooth® LE connection, in case the Proteus Connect App is put to background. To avoid this behaviour, the background service of the Proteus Connect App must be enabled by going to the info tab and selecting the "Bluetooth Background Mode" slider.

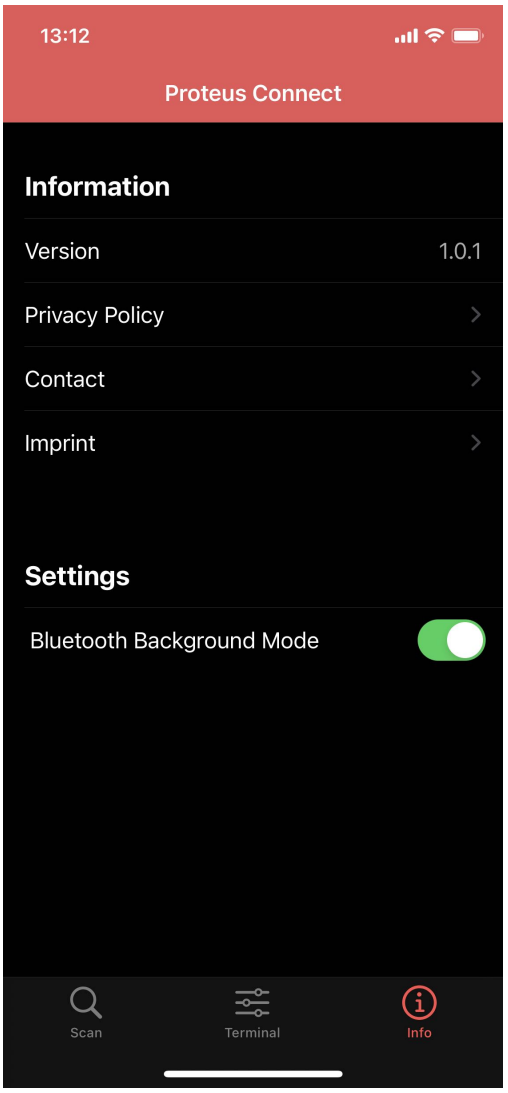


Figure 3: Enable the background service on iOS

5.3 Proteus module or USB radio stick as central device

This chapter describes how to setup a connection to the Proteus-e radio module in command mode, when another Proteus radio module (Proteus-I,-II,-III) or even Proteus USB radio stick is used as central device.

For reasons of simplicity, we will call the Proteus radio module or USB radio stick that is intended to setup the connection to the Proteus-e, **Proteus_central**. Furthermore, we will call the Proteus-e module, **Proteus_peripheral**.

Please note that the **Proteus_central** must run in command mode to initiate the connection setup.

In this example, we assume that the MAC of the **Proteus_peripheral** is 0x0018DA000011, and the MAC of the **Proteus_central** is 0x0018DA000055.

1. Connect **Proteus_central** to the **Proteus_peripheral** via Bluetooth® LE.

Info	**Proteus_central**	**Proteus_peripheral**
⇒ Request `CMD_CONNECT_REQ` with `FS_BTMAC` of **Proteus_peripheral**	02 06 06 00 11 00 00 DA 18 00 D1	
⇐ Response `CMD_CONNECT_CNF`: Request understood, try to connect now	02 46 01 00 00 45	
⇐ Indication `CMD_CONNECT_IND`: Physical connection established successfully to the module with `FS_BTMAC` 0x11 0x00 0x00 0xDA 0x18 0x00	02 86 07 00 00 11 00 00 DA 18 00 50	
⇐ Indication `CMD_CONNECT_IND`: Physical connection established successfully to module with `FS_BTMAC` 0x55 0x00 0x00 0xDA 0x18 0x00		02 86 07 00 00 55 00 00 DA 18 00 14
⇐ Channel opened successfully to the module with `FS_BTMAC` 0x11 0x00 0x00 0xDA 0x18 0x00 and maximum payload size of 0xF3 (243 Bytes) per packet	02 C6 08 00 00 11 00 00 DA 18 00 F3 EC	
⇐ Indication `CMD_CHANNELOPEN_RSP`: Channel opened successfully to module with `FS_BTMAC` 0x55 0x00 0x00 0xDA 0x18 0x00 and maximum payload size of 0xF3 (243 Bytes) per packet		02 C6 08 00 00 55 00 00 DA 18 00 F3 A7

2. Now the connection is active. Thus, data can be sent in each direction. Let us send a string "ABCD" from **Proteus_peripheral** to **Proteus_central**.

Info	Proteus_central	Proteus_peripheral
⇒ Request CMD_DATA_REQ: Send "ABCD" to **Proteus_central**		02 04 04 00 41 42 43 44 06
⇐ Response CMD_DATA_CNF: Request received, send data now		02 44 01 00 00 47
⇐ Indication CMD_DATA_IND: Received string "ABCD" from FS_BTMAC 0x11 0x00 0x00 0xDA 0x18 0x00 with RSSI of 0xCA (-54dBm)	02 84 0B 00 11 00 00 DA 18 00 CA 41 42 43 44 90	
⇐ Response CMD_TXCOMPLETE_RSP: Data transmitted successfully		02 C4 01 00 00 C7

3. Reply with "EFGH" to the **Proteus_peripheral**.

Info	Proteus_central	Proteus_peripheral
⇒ Request CMD_DATA_REQ: Send "EFGH" to **Proteus_peripheral**	02 04 04 00 45 46 47 48 0E	
⇐ Response CMD_DATA_CNF: Request received, send data now	02 44 01 00 00 47	
⇐ Indication CMD_DATA_IND: Received string "EFGH" from FS_BTMAC 0x55 0x00 0x00 0xDA 0x18 0x00 with RSSI of 0xC1 (-63dBm)		02 84 0B 00 55 00 00 DA 18 00 C1 45 46 47 48 D7
⇐ Response CMD_TXCOMPLETE_RSP: Data transmitted successfully	02 C4 01 00 00 C7	

4. Now **Proteus_central** closes the connection.

Info	Proteus_central	Proteus_peripheral
⇒ Request CMD_DISCONNECT_REQ: Disconnect	02 07 00 00 05	
⇐ Response CMD_DISCONNECT_CNF: Request received, disconnect now	02 47 01 00 00 44	
⇐ Indication CMD_DISCONNECT_IND: Connection closed	02 87 01 00 16 92	
⇐ Indication CMD_DISCONNECT_IND: Connection closed		02 87 01 00 13 97

6 References

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

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

[3] Würth Elektronik. Application note 24 - Proteus-e advanced developer guide. http://www.we-online.com/ANR024.

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

[5] Würth Elektronik. Proteus Connect app for iOS. https://apps.apple.com/de/app/proteus-connect/id1533941485.

[6] Würth Elektronik. Source code of Proteus Connect app for Android. https://github.com/WurthElektronik/Proteus-Connect-Android.

[7] Würth Elektronik. Source code of Proteus Connect app for iOS. https://github.com/WurthElektronik/Proteus-Connect-iOS.

7 Important notes

The following conditions apply to all goods within the wireless connectivity product range of Würth Elektronik eiSos GmbH & Co. KG:

7.1 General customer responsibility

Some goods within the product range of Würth Elektronik eiSos GmbH & Co. KG contain statements regarding general suitability for certain application areas. These statements about suitability are based on our knowledge and experience of typical requirements concerning the areas, serve as general guidance and cannot be estimated as binding statements about the suitability for a customer application. The responsibility for the applicability and use 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, where appropriate to investigate and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for the respective customer application or not. Accordingly, the customer is cautioned to verify that the documentation is current before placing orders.

7.2 Customer responsibility related to specific, in particular safety-relevant applications

It has to be clearly pointed out that the possibility of a malfunction of electronic components or failure before the end of the usual lifetime cannot be completely eliminated in the current state of the art, even if the products are operated within the range of the specifications. The same statement is valid for all software sourcecode and firmware parts contained in or used with or for products in the wireless connectivity and sensor product range of Würth Elektronik eiSos GmbH & Co. KG. In certain customer applications requiring a high level of safety and especially in customer applications in which the malfunction or failure of an electronic component could endanger human life or health, it must be ensured by most advanced technological aid of suitable design of the customer application that no injury or damage is caused to third parties in the event of malfunction or failure of an electronic component.

7.3 Best care and attention

Any product-specific data sheets, manuals, application notes, PCN's, warnings and cautions must be strictly observed in the most recent versions and matching to the products firmware revisions. This documents can be downloaded from the product specific sections on the wireless connectivity homepage.

7.4 Customer support for product specifications

Some products within the product range may contain substances, which are subject to restrictions in certain jurisdictions in order to serve specific technical requirements. Necessary information is available on request. In this case, the field sales engineer or the internal sales person in charge should be contacted who will be happy to support in this matter.

7.5 Product improvements

Due to constant product improvement, product specifications may change from time to time. As a standard reporting procedure of the Product Change Notification (PCN) according to the JEDEC-Standard, we inform about major changes. In case of further queries regarding the PCN, the field sales engineer, the internal sales person or the technical support team in charge should be contacted. The basic responsibility of the customer as per section 7.1 and 7.2 remains unaffected. All wireless connectivity module driver software ¨wireless connectivity SDK¨ and it's source codes as well as all PC software tools are not subject to the Product Change Notification information process.

7.6 Product life cycle

Due to technical progress and economical evaluation we also reserve the right to discontinue production and delivery of products. As a standard reporting procedure of the Product Termination Notification (PTN) according to the JEDEC-Standard we will inform at an early stage about inevitable product discontinuance. According to this, we cannot ensure that all products within our product range will always be available. Therefore, it needs to be verified with the field sales engineer or the internal sales person in charge about the current product availability expectancy before or when the product for application design-in disposal is considered. The approach named above does not apply in the case of individual agreements deviating from the foregoing for customer-specific products.

7.7 Property rights

All the rights for contractual products produced by Würth Elektronik eiSos GmbH & Co. KG on the basis of ideas, development contracts as well as models or templates that are subject to copyright, patent or commercial protection supplied to the customer will remain with Würth Elektronik eiSos GmbH & Co. KG. Würth Elektronik eiSos GmbH & Co. KG does not warrant or represent that any license, either expressed or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, application, or process in which Würth Elektronik eiSos GmbH & Co. KG components or services are used.

7.8 General terms and conditions

Unless otherwise agreed in individual contracts, all orders are subject to the current version of the "General Terms and Conditions of Würth Elektronik eiSos Group", last version available at *www.we-online.com*.

8 Legal notice

8.1 Exclusion of liability

Würth Elektronik eiSos GmbH & Co. KG considers the information in this document to be correct at the time of publication. However, Würth Elektronik eiSos GmbH & Co. KG reserves the right to modify the information such as technical specifications or functions of its products or discontinue the production of these products or the support of one of these products without any written announcement or notification to customers. The customer must make sure that the information used corresponds to the latest published information. Würth Elektronik eiSos GmbH & Co. KG does not assume any liability for the use of its products. Würth Elektronik eiSos GmbH & Co. KG does not grant licenses for its patent rights or for any other of its intellectual property rights or third-party rights.

Notwithstanding anything above, Würth Elektronik eiSos GmbH & Co. KG makes no representations and/or warranties of any kind for the provided information related to their accuracy, correctness, completeness, usage of the products and/or usability for customer applications. Information published by Würth Elektronik eiSos GmbH & Co. KG regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof.

8.2 Suitability in customer applications

The customer bears the responsibility for compliance of systems or units, in which Würth Elektronik eiSos GmbH & Co. KG products are integrated, with applicable legal regulations. Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of Würth Elektronik eiSos GmbH & Co. KG components in its applications, notwithstanding any applications-related in-formation or support that may be provided by Würth Elektronik eiSos GmbH & Co. KG. Customer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences lessen the likelihood of failures that might cause harm and take appropriate remedial actions. The customer will fully indemnify Würth Elektronik eiSos GmbH & Co. KGand its representatives against any damages arising out of the use of any Würth Elektronik eiSos GmbH & Co. KG components in safety-critical applications.

8.3 Trademarks

AMBER wireless is a registered trademark of Würth Elektronik eiSos GmbH & Co. KG. All other trademarks, registered trademarks, and product names are the exclusive property of the respective owners.

8.4 Usage restriction

Würth Elektronik eiSos GmbH & Co. KG products have been designed and developed for usage in general electronic equipment only. This product is not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where a failure of the product is reasonably expected to cause severe personal injury or death,

unless the parties have executed an agreement specifically governing such use. Moreover, Würth Elektronik eiSos GmbH & Co. KG products are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train control, ship control), transportation signal, disaster prevention, medical, public information network etc. Würth Elektronik eiSos GmbH & Co. KG 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 electronic component, which is used in electrical circuits that require high safety and reliability function or performance. By using Würth Elektronik eiSos GmbH & Co. KG products, the customer agrees to these terms and conditions.

9 License terms

This License Terms will take effect upon the purchase and usage of the Würth Elektronik eiSos GmbH & Co. KG wireless connectivity products. You hereby agree that this license terms is applicable to the product and the incorporated software, firmware and source codes (collectively, "Software") made available by Würth Elektronik eiSos in any form, including but not limited to binary, executable or source code form.
The software included in any Würth Elektronik eiSos wireless connectivity product is purchased to you on the condition that you accept the terms and conditions of this license terms. You agree to comply with all provisions under this license terms.

9.1 Limited license

Würth Elektronik eiSos hereby grants you a limited, non-exclusive, non-transferable and royalty-free license to use the software and under the conditions that will be set forth in this license terms. You are free to use the provided Software only in connection with one of the products from Würth Elektronik eiSos to the extent described in this license terms. You are entitled to change or alter the source code for the sole purpose of creating an application embedding the Würth Elektronik eiSos wireless connectivity product. The transfer of the source code to third parties is allowed to the sole extent that the source code is used by such third parties in connection with our product or another hardware provided by Würth Elektronik eiSos under strict adherence of this license terms. Würth Elektronik eiSos will not assume any liability for the usage of the incorporated software and the source code. You are not entitled to transfer the source code in any form to third parties without prior written consent of Würth Elektronik eiSos.
You are not allowed to reproduce, translate, reverse engineer, decompile, disassemble or create derivative works of the incorporated Software and the source code in whole or in part. No more extensive rights to use and exploit the products are granted to you.

9.2 Usage and obligations

The responsibility for the applicability and use of the Würth Elektronik eiSos wireless connectivity product with the incorporated Firmware in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to you to evaluate and investigate, where appropriate, and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for your respective application or not.
You are responsible for using the Würth Elektronik eiSos wireless connectivity product with the incorporated Firmware in compliance with all applicable product liability and product safety laws. You acknowledge to minimize the risk of loss and harm to individuals and bear the risk for failure leading to personal injury or death due to your usage of the product.
Würth Elektronik eiSos' products with the incorporated Firmware are not authorized for use in safety-critical applications, or where a failure of the product is reasonably expected to cause severe personal injury or death. Moreover, Würth Elektronik eiSos' products with the incorporated Firmware are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train control, ship control), transportation signal, disaster prevention, medical, public information network etc. You shall inform Würth Elektronik eiSos about the intent of such usage before design-in stage. In certain customer applications requiring a very high level of safety and in which the malfunction or failure of an electronic component could endanger human life or

health, you must ensure to have all necessary expertise in the safety and regulatory ramifications of your applications. You acknowledge and agree that you are solely responsible for all legal, regulatory and safety-related requirements concerning your products and any use of Würth Elektronik eiSos' products with the incorporated Firmware in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Würth Elektronik eiSos. YOU SHALL INDEMNIFY WÜRTH ELEKTRONIK EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF WÜRTH ELEKTRONIK EISOS' PRODUCTS WITH THE INCORPORATED FIRMWARE IN SUCH SAFETY-CRITICAL APPLICATIONS.

9.3 Ownership

The incorporated Firmware created by Würth Elektronik eiSos is and will remain the exclusive property of Würth Elektronik eiSos.

9.4 Firmware update(s)

You have the opportunity to request the current and actual Firmware for a bought wireless connectivity Product within the time of warranty. However, Würth Elektronik eiSos has no obligation to update a modules firmware in their production facilities, but can offer this as a service on request. The upload of firmware updates falls within your responsibility, e.g. via ACC or another software for firmware updates. Firmware updates will not be communicated automatically. It is within your responsibility to check the current version of a firmware in the latest version of the product manual on our website. The revision table in the product manual provides all necessary information about firmware updates. There is no right to be provided with binary files, so called "Firmware images", those could be flashed through JTAG, SWD, Spi-Bi-Wire, SPI or similar interfaces.

9.5 Disclaimer of warranty

THE FIRMWARE IS PROVIDED "AS IS". YOU ACKNOWLEDGE THAT WÜRTH ELEKTRONIK 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 YOUR INTENDED PURPOSE OR USAGE. WÜRTH ELEKTRONIK 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 THE WÜRTH ELEKTRONIK EISOS' PRODUCT WITH THE INCORPORATED FIRMWARE IS USED. INFORMATION PUBLISHED BY WÜRTH ELEKTRONIK EISOS REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE A LICENSE FROM WÜRTH ELEKTRONIK EISOS TO USE SUCH PRODUCTS OR SERVICES OR A WARRANTY OR ENDORSEMENT THEREOF.

9.6 Limitation of liability

Any liability not expressly provided by Würth Elektronik eiSos shall be disclaimed.
You agree to hold us harmless from any third-party claims related to your usage of the Würth Elektronik eiSos' products with the incorporated Firmware, software and source code. Würth

Elektronik eiSos disclaims any liability for any alteration, development created by you or your customers as well as for any combination with other products.

9.7 Applicable law and jurisdiction

Applicable law to this license terms shall be the laws of the Federal Republic of Germany. Any dispute, claim or controversy arising out of or relating to this license terms shall be resolved and finally settled by the court competent for the location of Würth Elektronik eiSos' registered office.

9.8 Severability clause

If a provision of this license terms is or becomes invalid, unenforceable or null and void, this shall not affect the remaining provisions of the terms. The parties shall replace any such provisions with new valid provisions that most closely approximate the purpose of the terms.

9.9 Miscellaneous

Würth Elektronik eiSos reserves the right at any time to change this terms at its own discretion. It is your responsibility to check at Würth Elektronik eiSos homepage for any updates. Your continued usage of the products will be deemed as the acceptance of the change.
We recommend you to be updated about the status of new firmware and software, which is available on our website or in our data sheet and manual, and to implement new software in your device where appropriate.
By ordering a wireless connectivity product, you accept this license terms in all terms.

List of Figures

List of Tables

WÜRTH ELEKTRONIK

more than you expect

Internet of Things

Monitoring & Control

Automated Meter Reading

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