WÜRTH ELEKTRONIK

ANR014 Proteus Quickstart

Connect a smart phone to a Proteus Bluetooth LE radio module

Version 1.2

March 11, 2021

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 evaluation board	March 2021

Abbreviations and abstract

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, a radio module of the Proteus series can be controlled and configured by the host using predefined commands, in the so called command mode.
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 which is an open source App providing standard Bluetooth® LE functions for iOS as well as for Android devices.

> ! 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 **AN-R004_Proteus_Peripheral_Only_Mode**.

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 **Proteus Connect** App for Android [3][5] or iOS [4][6]
  - the Nordic Semiconductor **nRF Connect** App for Android [1] or iOS [2]
- A Proteus evaluation board in factory state, for example
  - the Proteus-I evaluation board with jumpers set as specified in figure 1. Other jumpers not set.
  - the Proteus-II evaluation board with jumpers set as specified in figure 1. Other jumpers not set.
  - the Proteus-III evaluation board with jumpers set as specified in figure 2. Other jumpers not set.
  - the Proteus-III-SPI mini evaluation board with jumpers set as specified in figure 3. Other jumpers not set.

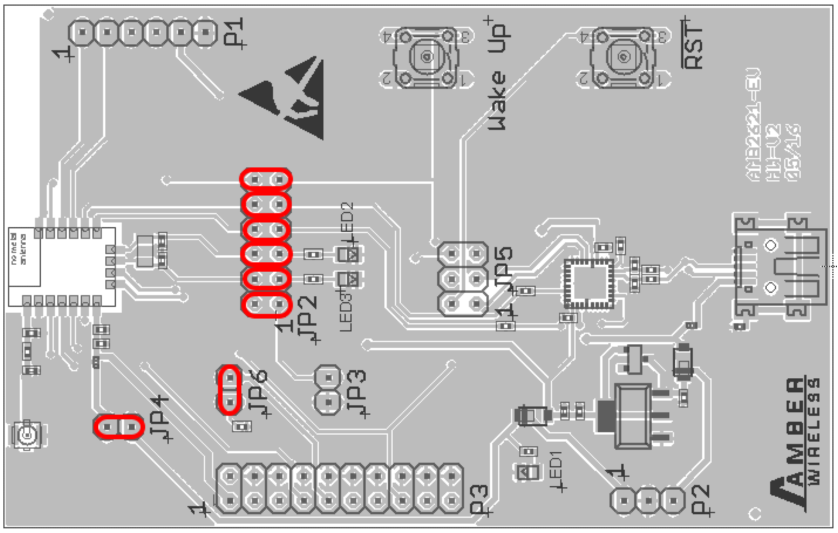


Figure 1: Default jumper placement of the Proteus-I and Proteus-II evaluation board. Red means "jumper must be set".

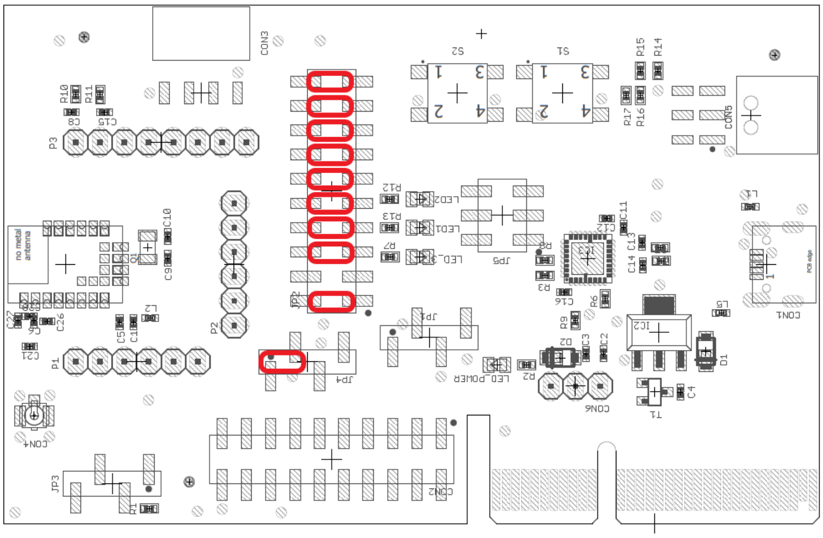


Figure 2: Default jumper placement of the Proteus-III evaluation board. Red means "jumper must be set".

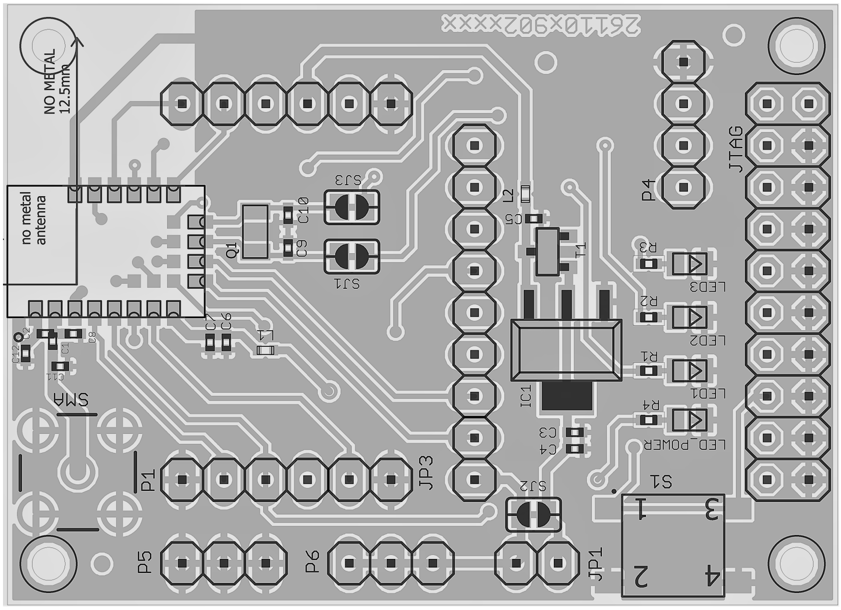


Figure 3: Default jumper placement of the Proteus-III-SPI mini evaluation board.

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 reference manual
  - Proteus-I advanced user guide **ANR002_Proteus-I_Advanced_Developer_Guide**
- Proteus-II
  - Proteus-II reference manual
  - Proteus-II advanced user guide **ANR005_Proteus-II_Advanced_Developer_Guide**
- Proteus-III

  - Proteus-III reference manual
  - Proteus-III advanced user guide **ANR009_Proteus-III_Advanced_Developer_Guide**

- Proteus-III-SPI
  - Proteus-III-SPI reference manual
  - Proteus-III advanced user guide **ANR009_Proteus-III_Advanced_Developer_Guide**

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 send. 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.

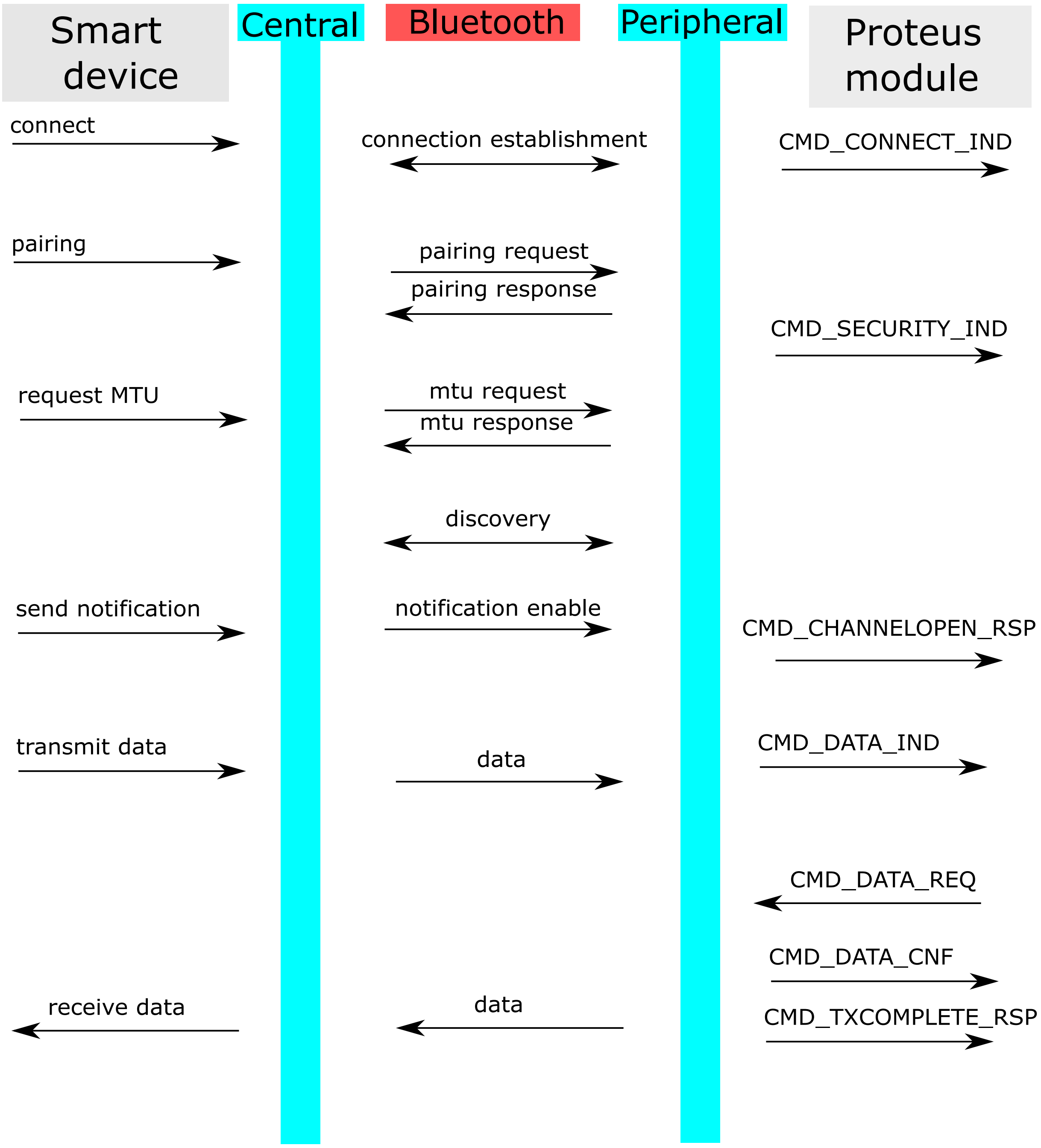


Figure 4: Steps for the connection setup

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.
First of all the **Proteus Connect** App is used. Then the same is done using the Nordic Semiconductor **nRF Connect** App.

4.1 Proteus Connect App

This chapter describes how to setup a connection to the Proteus module 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 [3][4] as well as source code [5][6].

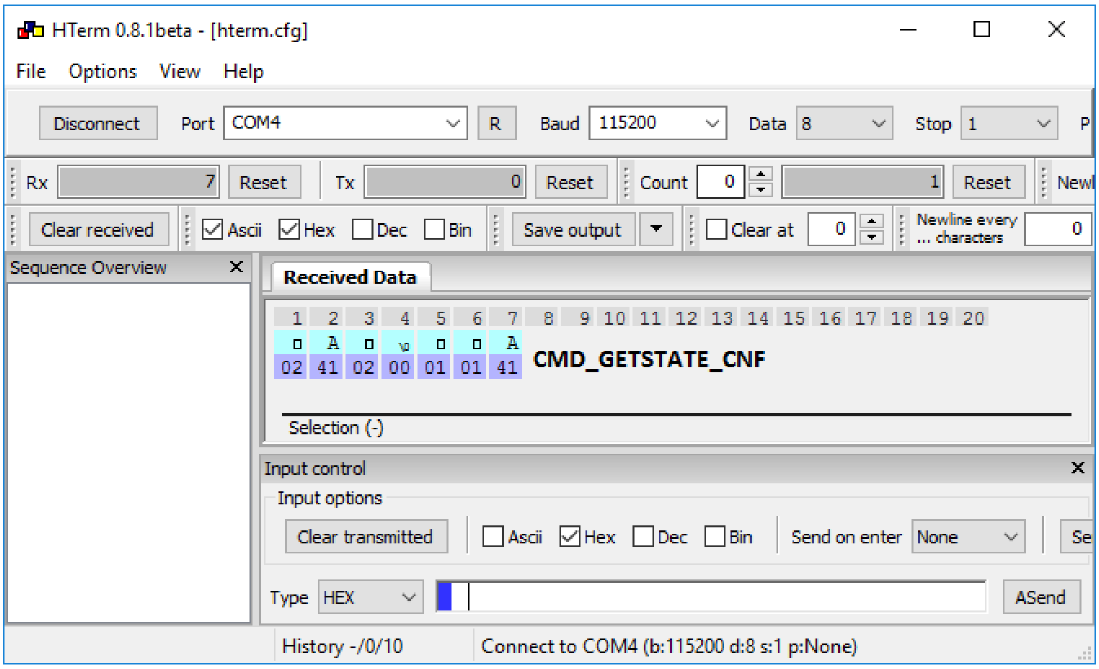
Please perform the following steps:

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. For Proteus-I, -II and -III evaluation board this can be simply achieved by using a simple USB cable to connect it to a PC.
  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 radio module.

- Open the terminal program using the Proteus 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.



HTerm 0.8.1beta - [hterm.cfg]

Port COM4 · Baud 115200 · Data 8 · Stop 1

Rx 7 · Tx 0 · Count 0 · 1

☑ Ascii ☑ Hex ☐ Dec ☐ Bin · ☐ Clear at 0 · Newline every ... characters 0

Received Data

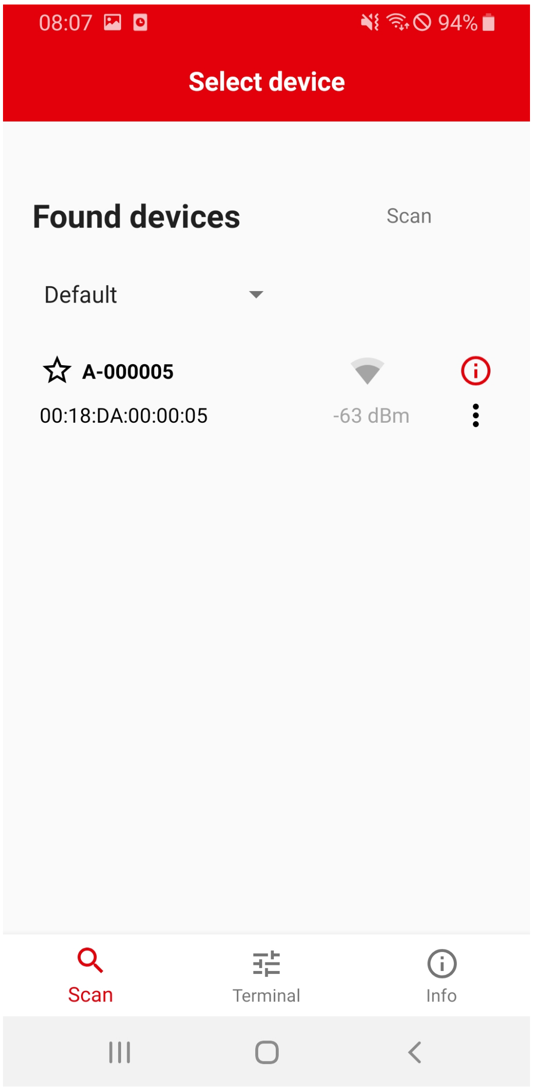
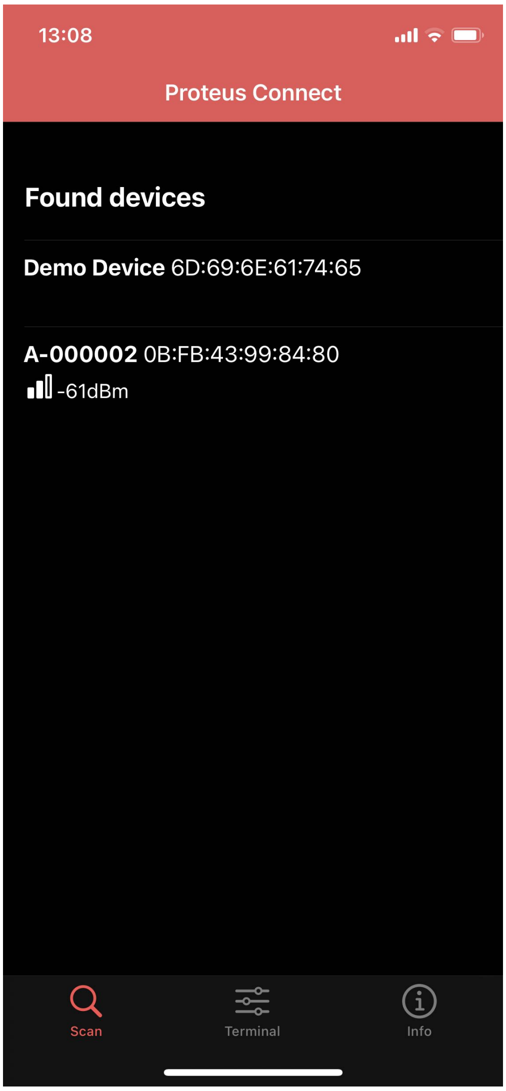
```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 □  A  □  \0 □  □  A
02 41 02 00 01 01 41  CMD_GETSTATE_CNF
```

Input options: ☐ Ascii ☑ Hex ☐ Dec ☐ Bin · Send on enter None · Type HEX

History -/0/10 · Connect to COM4 (b:115200 d:8 s:1 p:None)

Android	iOS

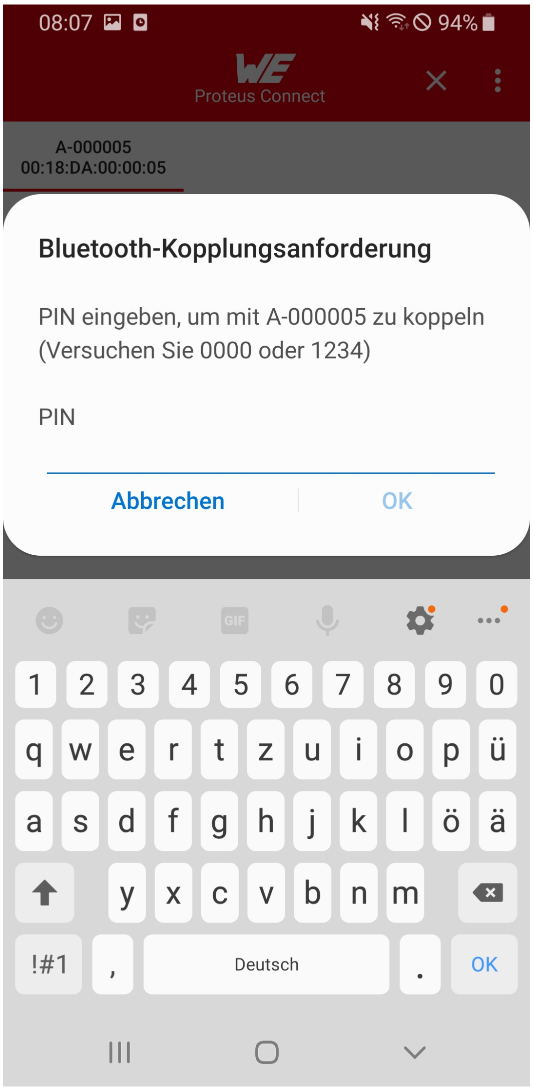
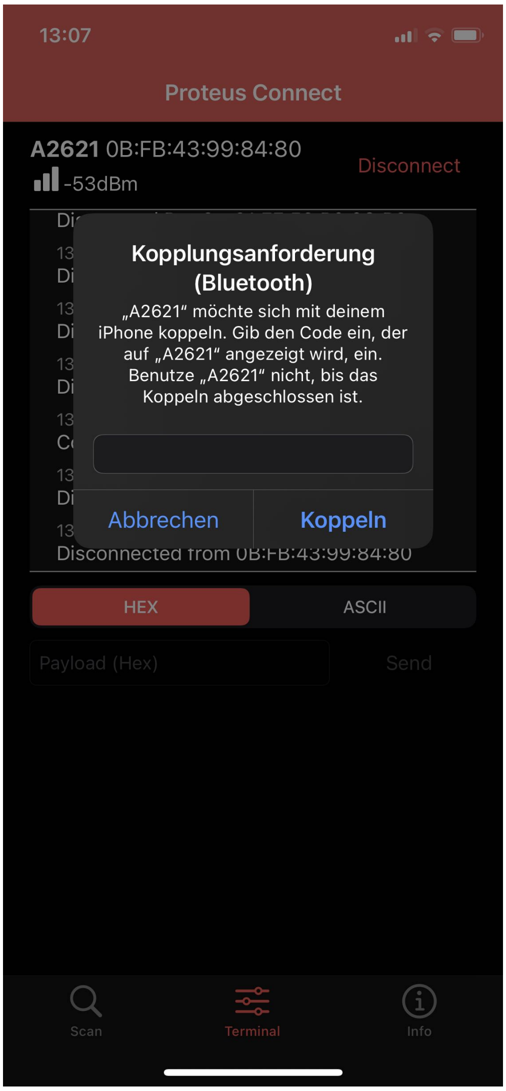
- Initially, the module is advertising. Thus, one LED of the Proteus evaluation board is blinking.
- Start your smart phone, enable the Bluetooth® LE feature and start the **Proteus Connect** App.

Android	iOS
• Press "Scan" to find the module on the radio.  08:07 94% Select device Found devices Scan Default ☆ A-000005 00:18:DA:00:00:05 -63 dBm Scan Terminal Info	 13:08 Proteus Connect Found devices **Demo Device** 6D:69:6E:61:74:65 **A-000002** 0B:FB:43:99:84:80 -61dBm Scan Terminal Info

- 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* (*LED_3* on the Proteus-EV) will constantly light up.

<table>
<tr><th>Android</th><th>iOS</th></tr>
<tr><td colspan="2">

- 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.

</td></tr>
<tr><td>08:07 94%
Proteus Connect
A-000005
00:18:DA:00:00:05
Bluetooth-Kopplungsanforderung
PIN eingeben, um mit A-000005 zu koppeln (Versuchen Sie 0000 oder 1234)
PIN
Abbrechen OK
 </td>
<td>13:07
Proteus Connect
A2621 0B:FB:43:99:84:80 Disconnect
-53dBm
Kopplungsanforderung (Bluetooth)
„A2621“ möchte sich mit deinem iPhone koppeln. Gib den Code ein, der auf „A2621“ angezeigt wird, ein. Benutze „A2621“ nicht, bis das Koppeln abgeschlossen ist.
Abbrechen Koppeln
Disconnected from 0B:FB:43:99:84:80

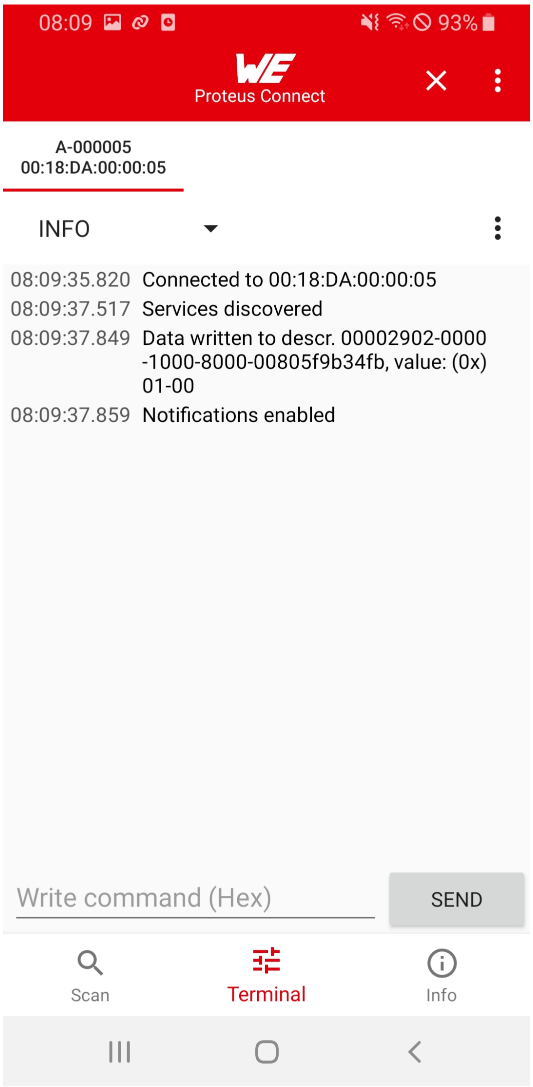
HEX
ASCII
Payload (Hex)
Send
Scan
Terminal
Info

</td></tr>
</table>

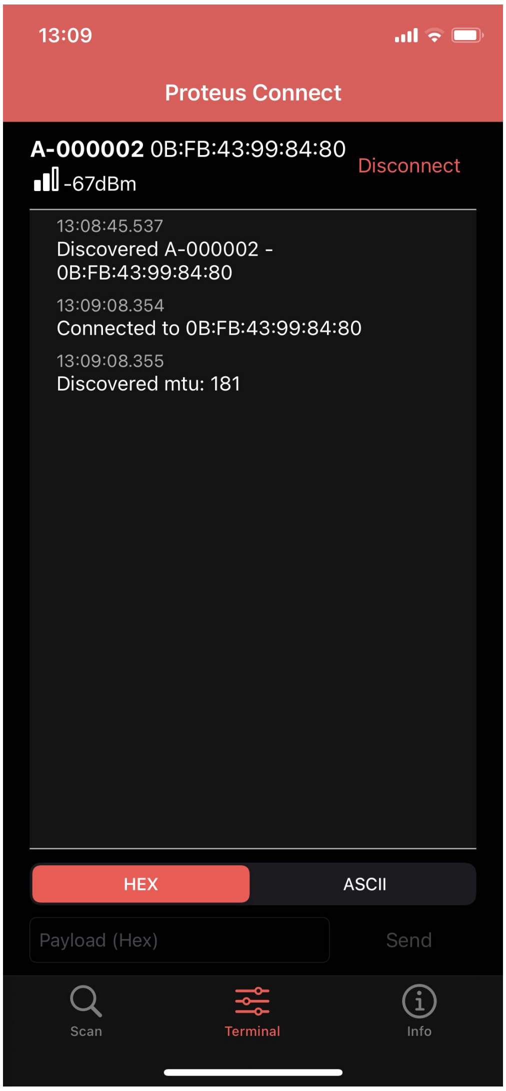
In few cases the Android may show an "authentication timeout" pop-up message, when entering the key. In this case, please proceed entering the key and simply do a reconnect. On this reconnect, the entered key information is reused and the connection is opened.

Android	iOS

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



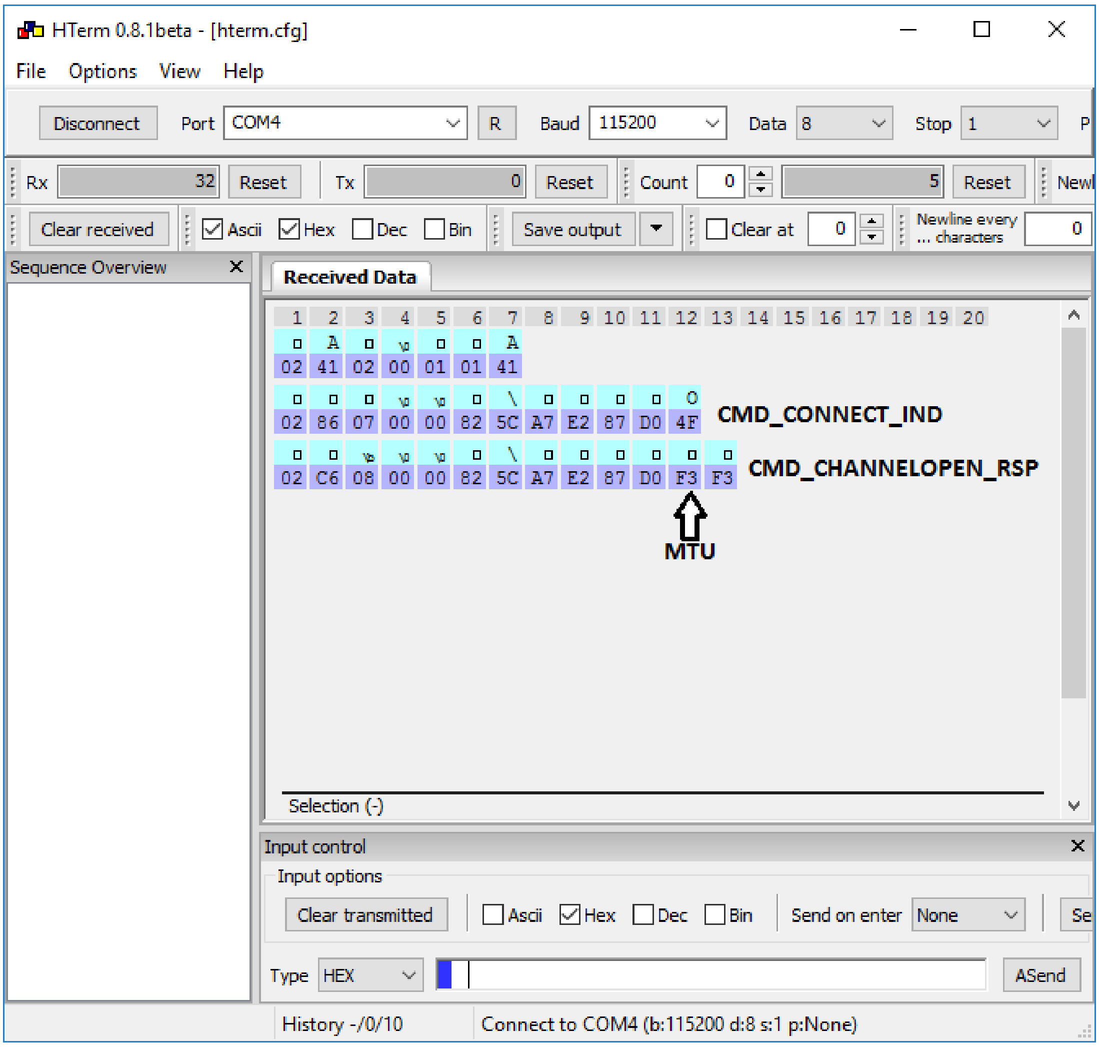
```
08:09:35.820  Connected to 00:18:DA:00:00:05
08:09:37.517  Services discovered
08:09:37.849  Data written to descr. 00002902-0000
              -1000-8000-00805f9b34fb, value: (0x)
              01-00
08:09:37.859  Notifications enabled
```



```
13:08:45.537
Discovered A-000002 -
0B:FB:43:99:84:80
13:09:08.354
Connected to 0B:FB:43:99:84:80
13:09:08.355
Discovered mtu: 181
```

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 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.



Port COM4 | Baud 115200 | Data 8 | Stop 1

Rx 32 | Tx 0 | Count 0 | 5

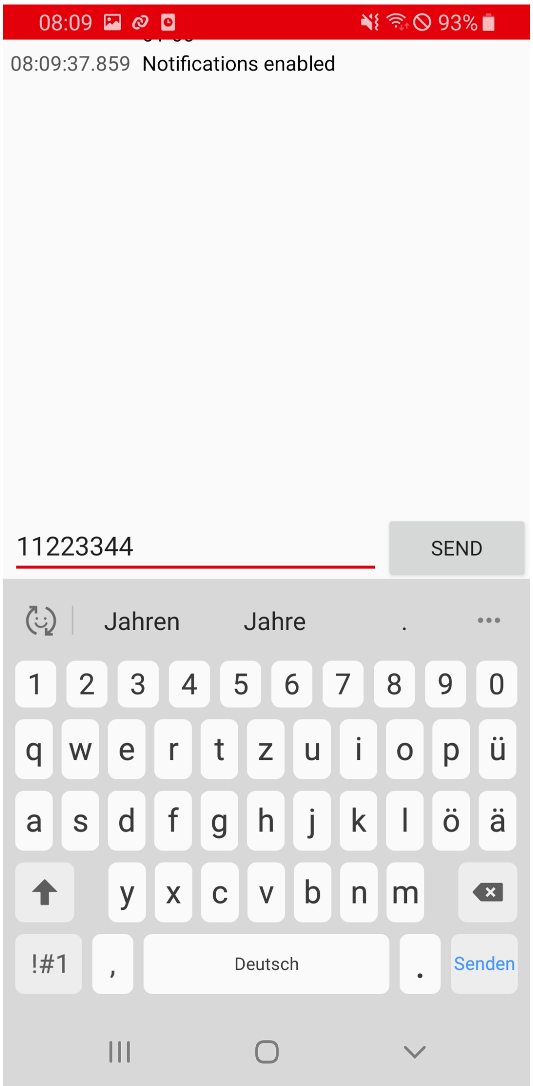
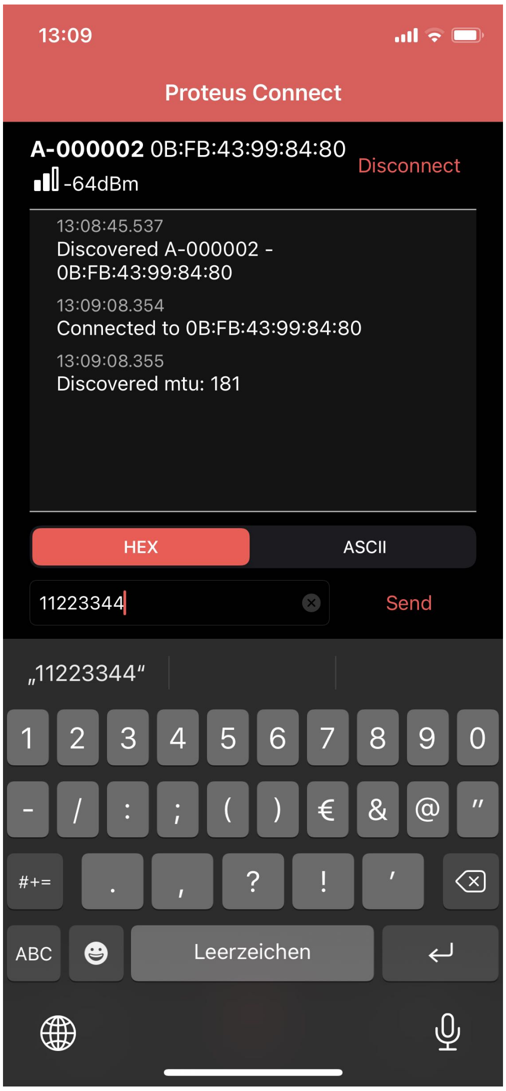
☑ Ascii ☑ Hex □ Dec □ Bin | □ Clear at 0 | Newline every ... characters 0

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 □  A  □  \0 □  □  A
02 41 02 00 01 01 41
 □  □  □  \0 \0 □  \  □  □  □  □  O
02 86 07 00 00 82 5C A7 E2 87 D0 4F
 □  □  \b \0 \0 □  \  □  □  □  □  □  □
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3
```

□ Ascii ☑ Hex □ Dec □ Bin | Send on enter None

Type HEX

History -/0/10 | Connect to COM4 (b:115200 d:8 s:1 p:None)

<table>
<tr><th>Android</th><th>iOS</th></tr>
<tr><td colspan="2">• 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 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.</td></tr>
<tr><td>• Android usually allows up to 243 bytes.</td><td>• iOS usually allows up to 181 bytes.</td></tr>
<tr><td>
08:09:37.859 Notifications enabled
11223344
 </td><td>
13:09
Proteus Connect
A-000002 0B:FB:43:99:84:80
-64dBm
13:08:45.537
Discovered A-000002 - 0B:FB:43:99:84:80
13:09:08.354
Connected to 0B:FB:43:99:84:80
13:09:08.355
Discovered mtu: 181
HEX
11223344

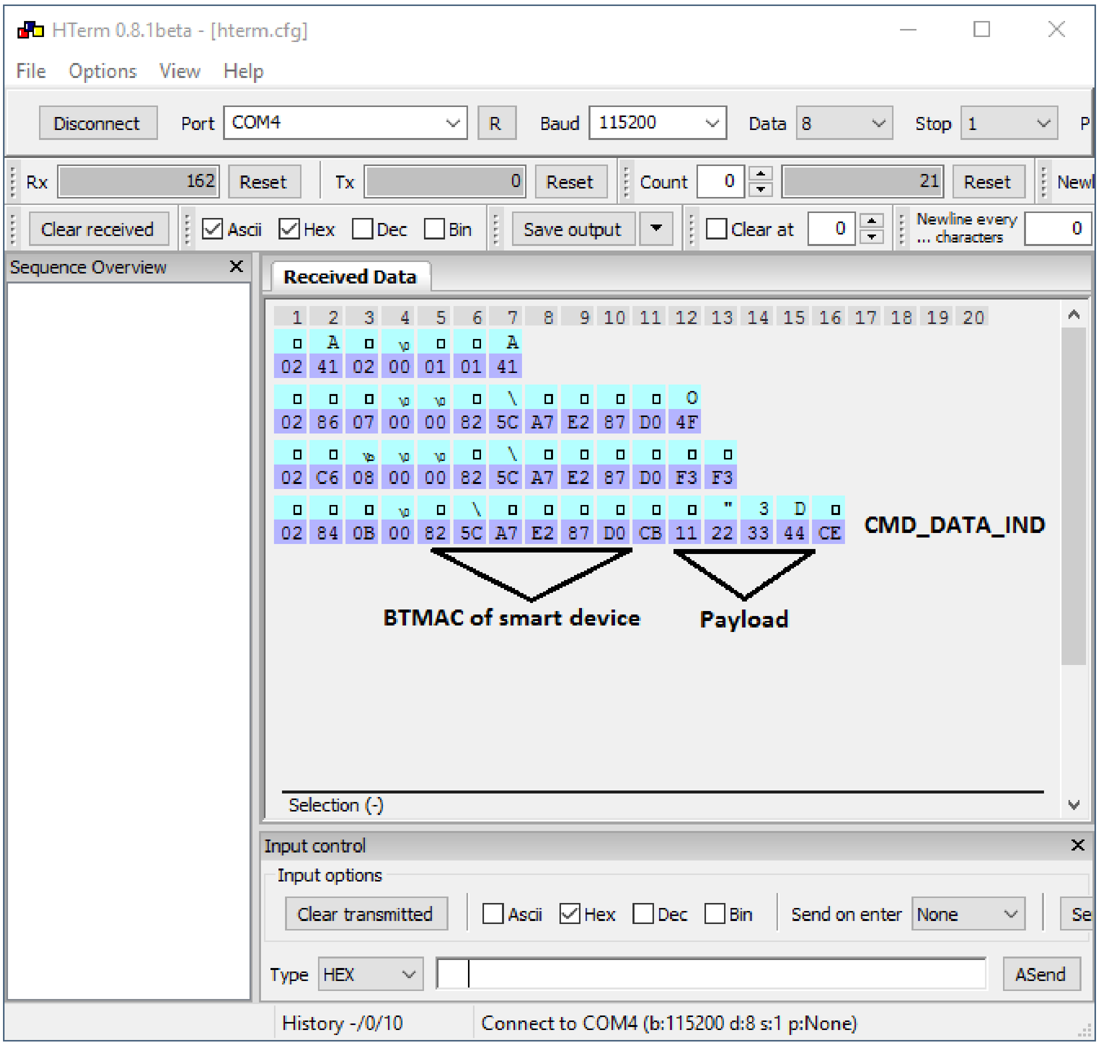
Disconnect
ASCII
Send
„11223344“
Leerzeichen
ABC

</td></tr>
</table>

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



Port COM4 Baud 115200 Data 8 Stop 1

Rx 162 Tx 0 Count 0 21

☑ Ascii ☑ Hex □ Dec □ Bin □ Clear at 0 Newline every ... characters 0

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 □  A  □  \0 □  □  A
02 41 02 00 01 01 41

 □  □  □  \0 \0 □  \  □  □  □  □  O
02 86 07 00 00 82 5C A7 E2 87 D0 4F

 □  □  \b \0 \0 □  \  □  □  □  □  □  □
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3

 □  □  □  \0 □  \  □  □  □  □  □  □  "  3  D  □
02 84 0B 00 82 5C A7 E2 87 D0 CB 11 22 33 44 CE
```

□ Ascii ☑ Hex □ Dec □ Bin Send on enter None

Type HEX

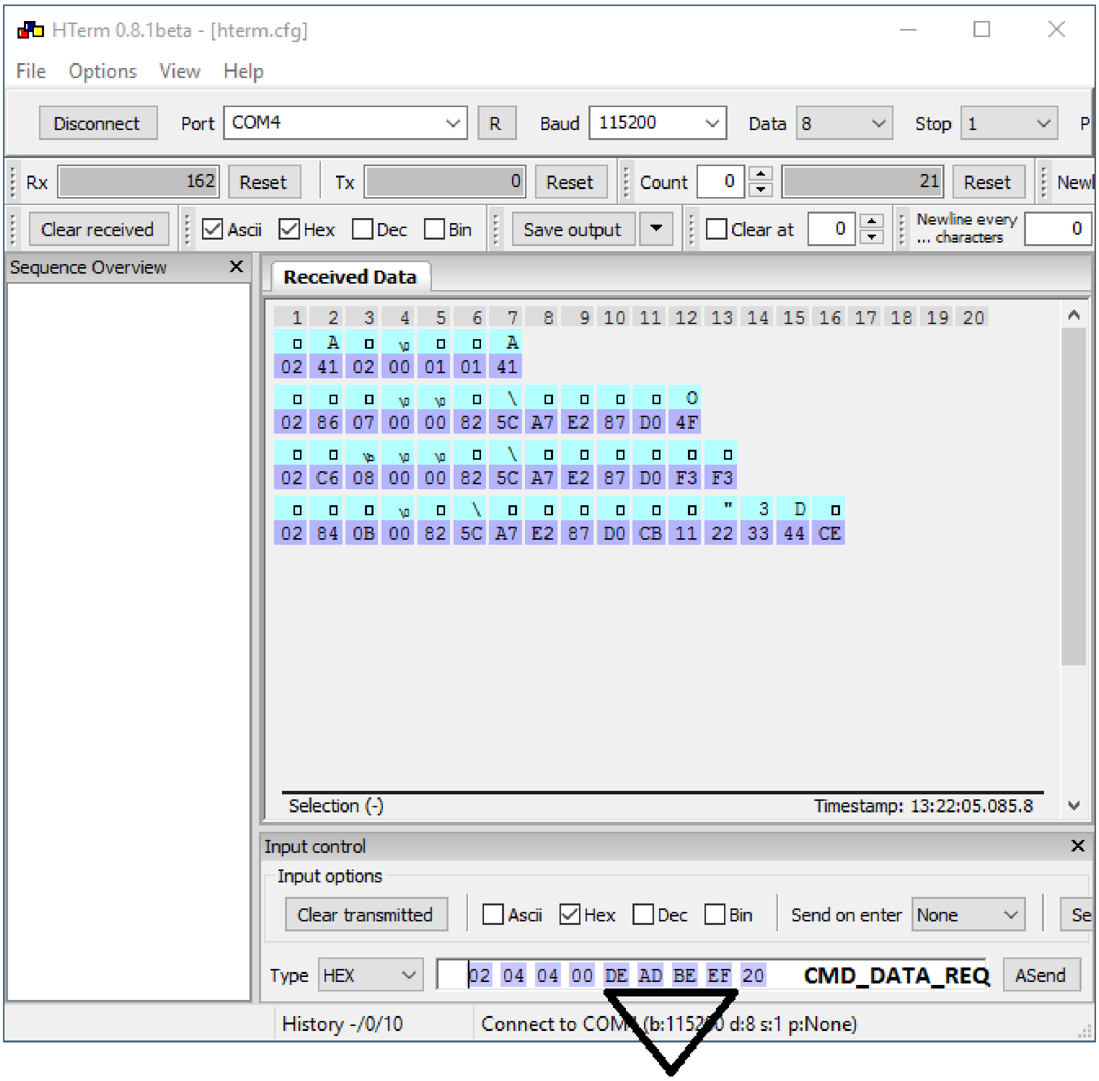
History -/0/10 Connect to COM4 (b:115200 d:8 s:1 p:None)

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.



Port COM4 · Baud 115200 · Data 8 · Stop 1

Rx 162 · Tx 0 · Count 0 · 21 · Newline every ... characters 0 · Clear at 0

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
02 41 02 00 01 01 41
02 86 07 00 00 82 5C A7 E2 87 D0 4F
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3
02 84 0B 00 82 5C A7 E2 87 D0 CB 11 22 33 44 CE
```

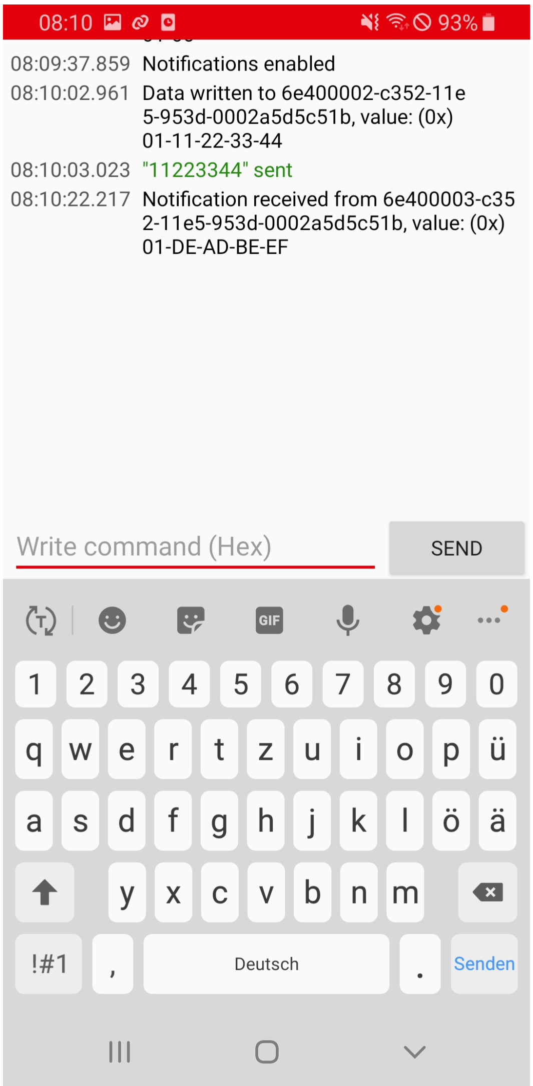
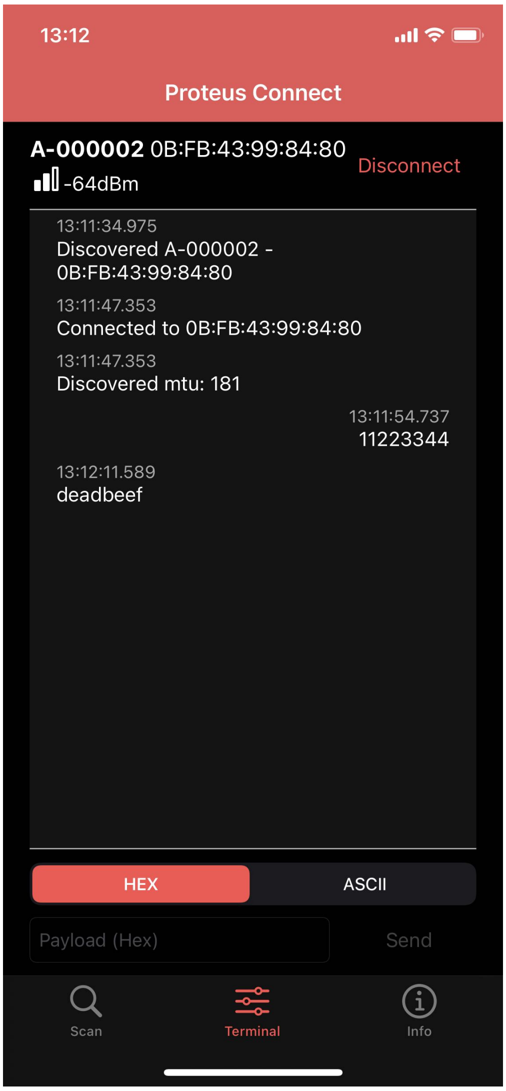
Selection (-) · Timestamp: 13:22:05.085.8

Send on enter None

Type HEX · 02 04 04 00 DE AD BE EF 20 **CMD_DATA_REQ**

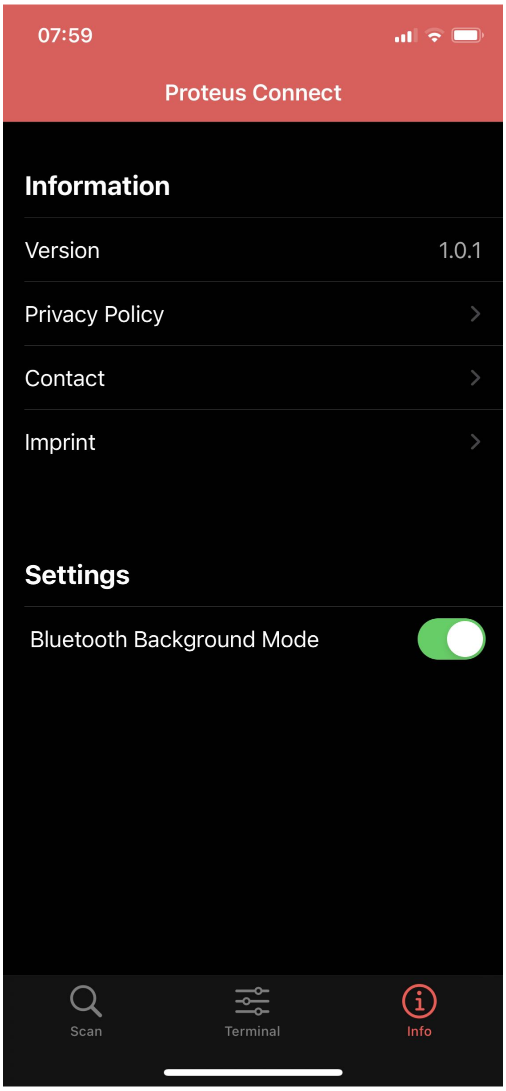
History -/0/10 · Connect to COM4 (b:115200 d:8 s:1 p:None)

Payload, no header 0x01 needed

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.
 08:10 93% 08:09:37.859 Notifications enabled 08:10:02.961 Data written to 6e400002-c352-11e5-953d-0002a5d5c51b, value: (0x) 01-11-22-33-44 08:10:03.023 "11223344" sent 08:10:22.217 Notification received from 6e400003-c352-11e5-953d-0002a5d5c51b, value: (0x) 01-DE-AD-BE-EF Write command (Hex) SEND	 13:12 Proteus Connect A-000002 0B:FB:43:99:84:80 Disconnect -64dBm 13:11:34.975 Discovered A-000002 - 0B:FB:43:99:84:80 13:11:47.353 Connected to 0B:FB:43:99:84:80 13:11:47.353 Discovered mtu: 181 13:11:54.737 11223344 13:12:11.589 deadbeef HEX ASCII Payload (Hex) Send Scan Terminal Info

4.1.1 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 behavior, the background service of the Proteus Connect App must be enabled by going to the info tab and selecting the "Bluetooth Background Mode" slider.



Information

Version 1.0.1

Privacy Policy

Contact

Imprint

Settings

Bluetooth Background Mode ●

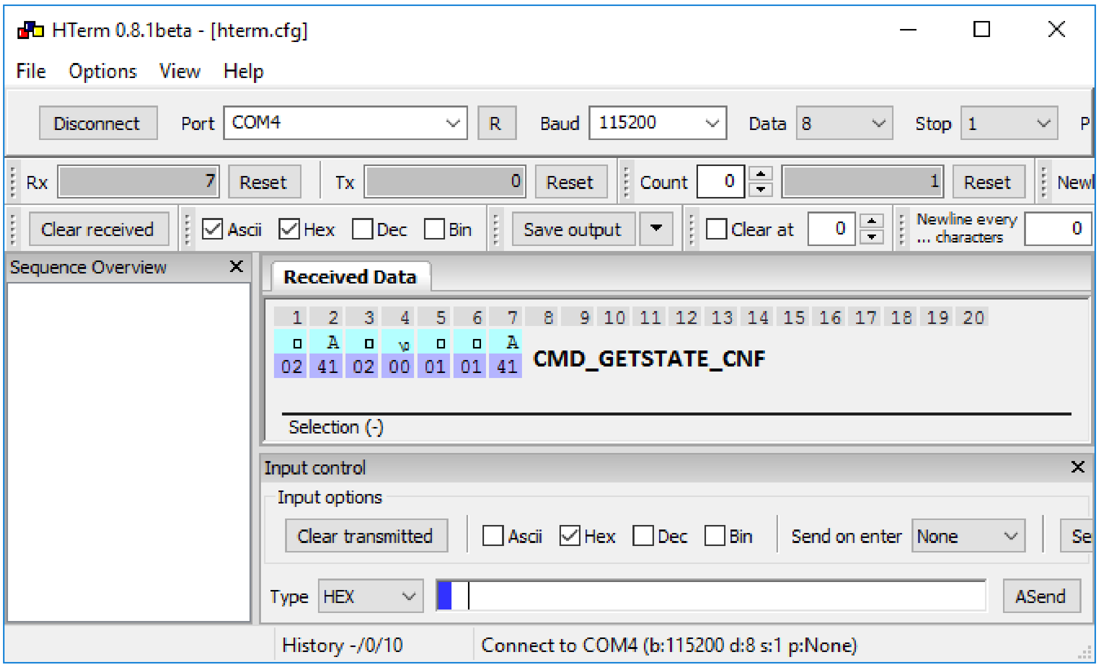
Figure 5: Enable the background service on iOS

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 [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. For Proteus-I, -II and -III evaluation board this can be simply achieved by using a simple USB cable to connect it to a PC.
  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 radio module.
- Open the terminal program using the Proteus 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.



HTerm 0.8.1beta - [hterm.cfg]

Port COM4 · Baud 115200 · Data 8 · Stop 1

Rx 7 · Tx 0 · Count 0 · 1

☑ Ascii ☑ Hex □ Dec □ Bin · □ Clear at 0 · Newline every ... characters 0

Received Data:

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 □  A  □  ␀  □  □  A
02 41 02 00 01 01 41  CMD_GETSTATE_CNF
```

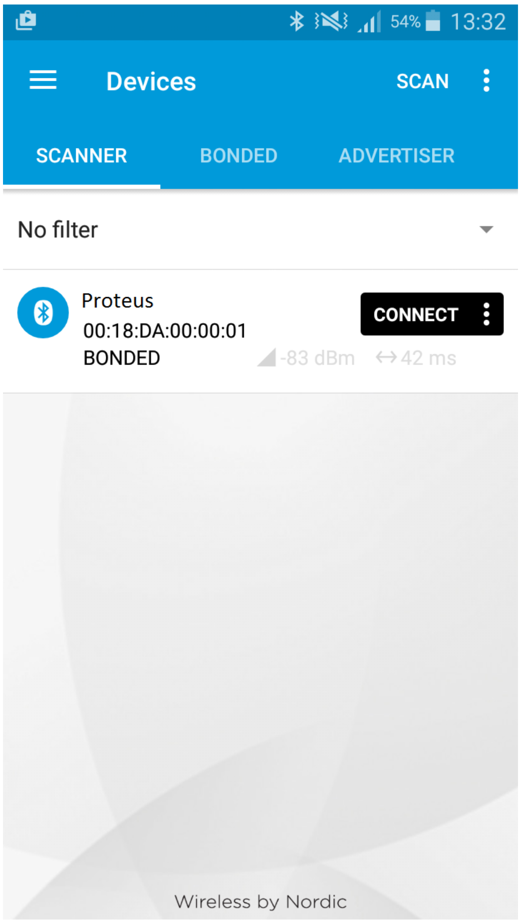
Selection (-)

Input options: □ Ascii ☑ Hex □ Dec □ Bin · Send on enter None · Type HEX

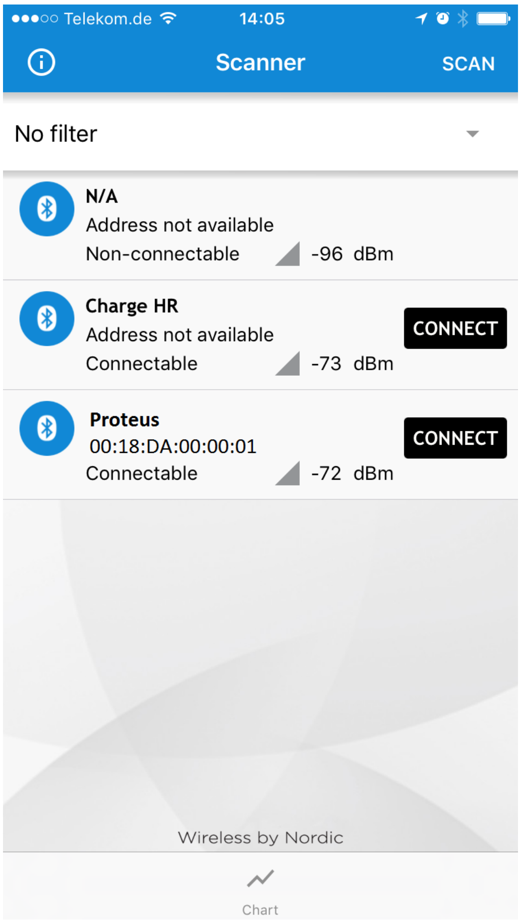
History -/0/10 · Connect to COM4 (b:115200 d:8 s:1 p:None)

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.



Proteus
00:18:DA:00:00:01
BONDED
-83 dBm ↔ 42 ms



N/A
Address not available
Non-connectable -96 dBm

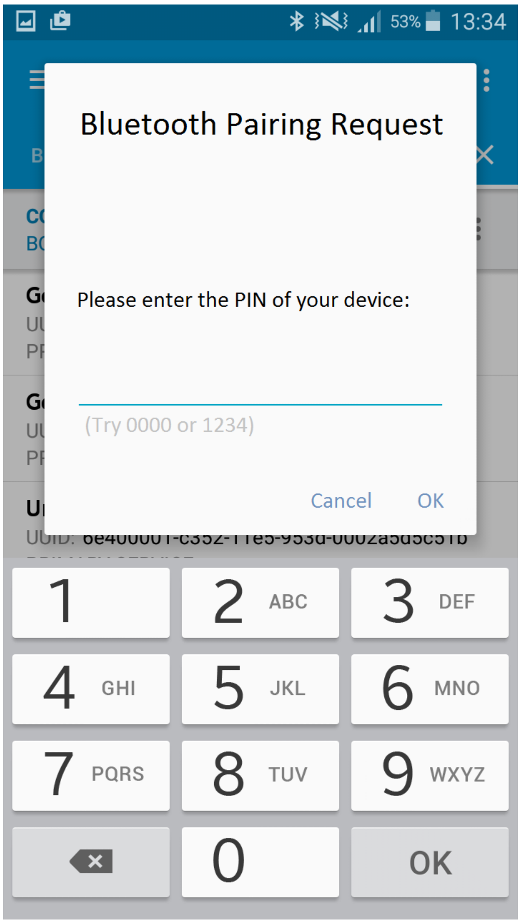
Charge HR
Address not available
Connectable -73 dBm

Proteus
00:18:DA:00:00:01
Connectable -72 dBm

- When the module appears, press "CONNECT".

Android	iOS

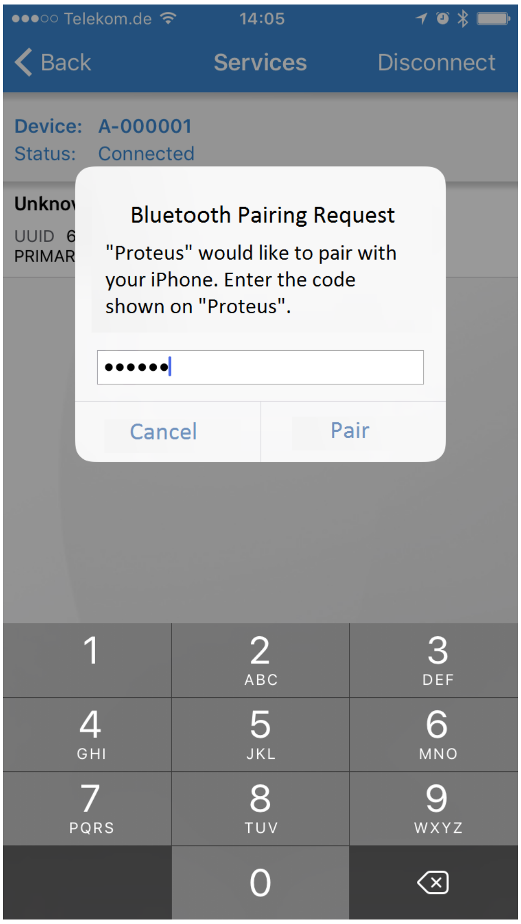
- As soon as the module has received the connection request from the smart phone the blinking LED will switch to constant on.
- Optional pairing: In case a security mode has been configured before, the smart phone requests the user for pairing actions. In case of the static passkey authentication, the Proteus requests to enter the static passkey. The default passkey is "123123". The Bluetooth® coupling requirement pop-up is shown on your smart phone. 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.



Bluetooth Pairing Request

Please enter the PIN of your device:

(Try 0000 or 1234)

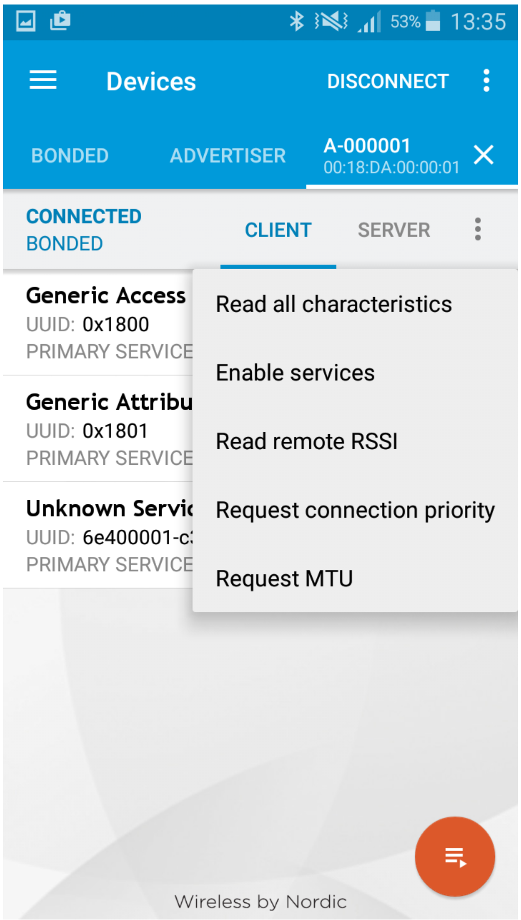
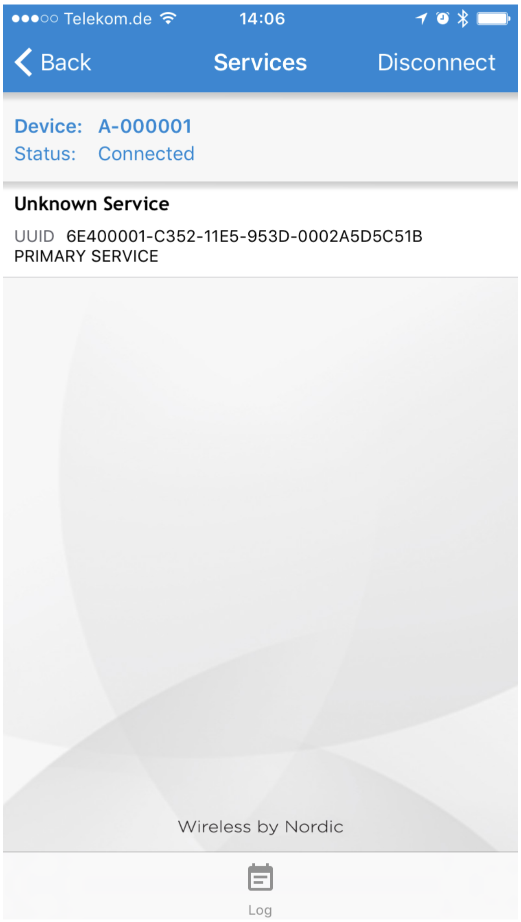


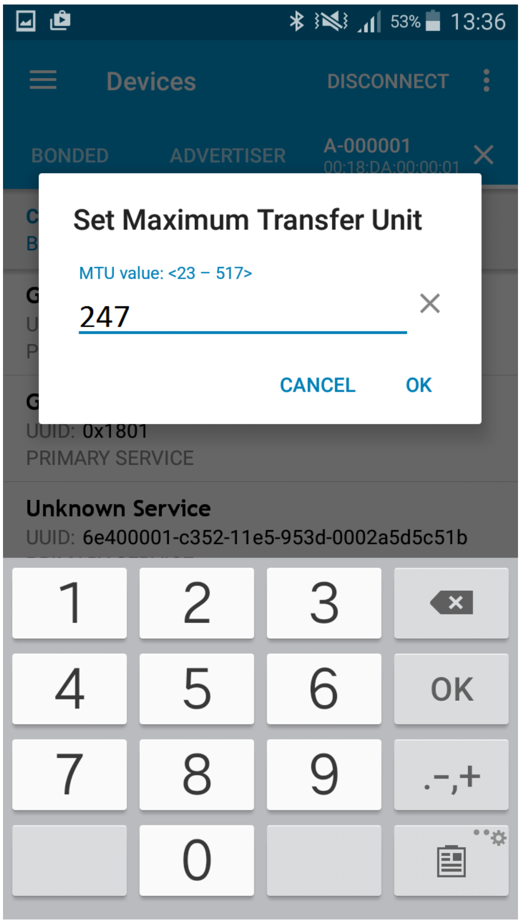
Device: A-000001
Status: Connected

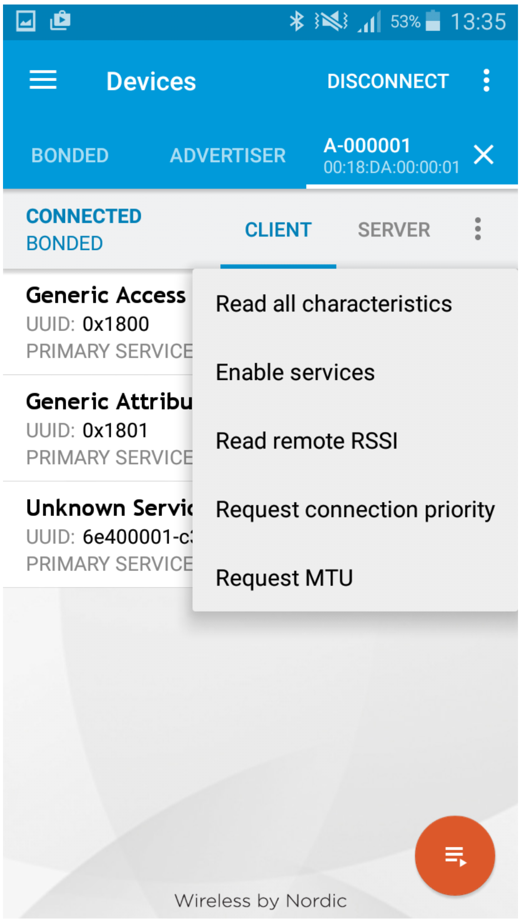
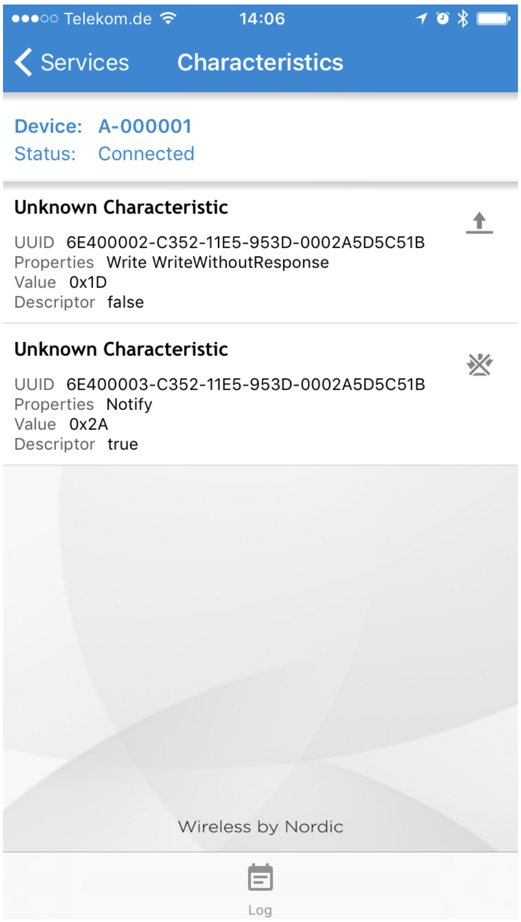
Bluetooth Pairing Request

"Proteus" would like to pair with your iPhone. Enter the code shown on "Proteus".

••••••

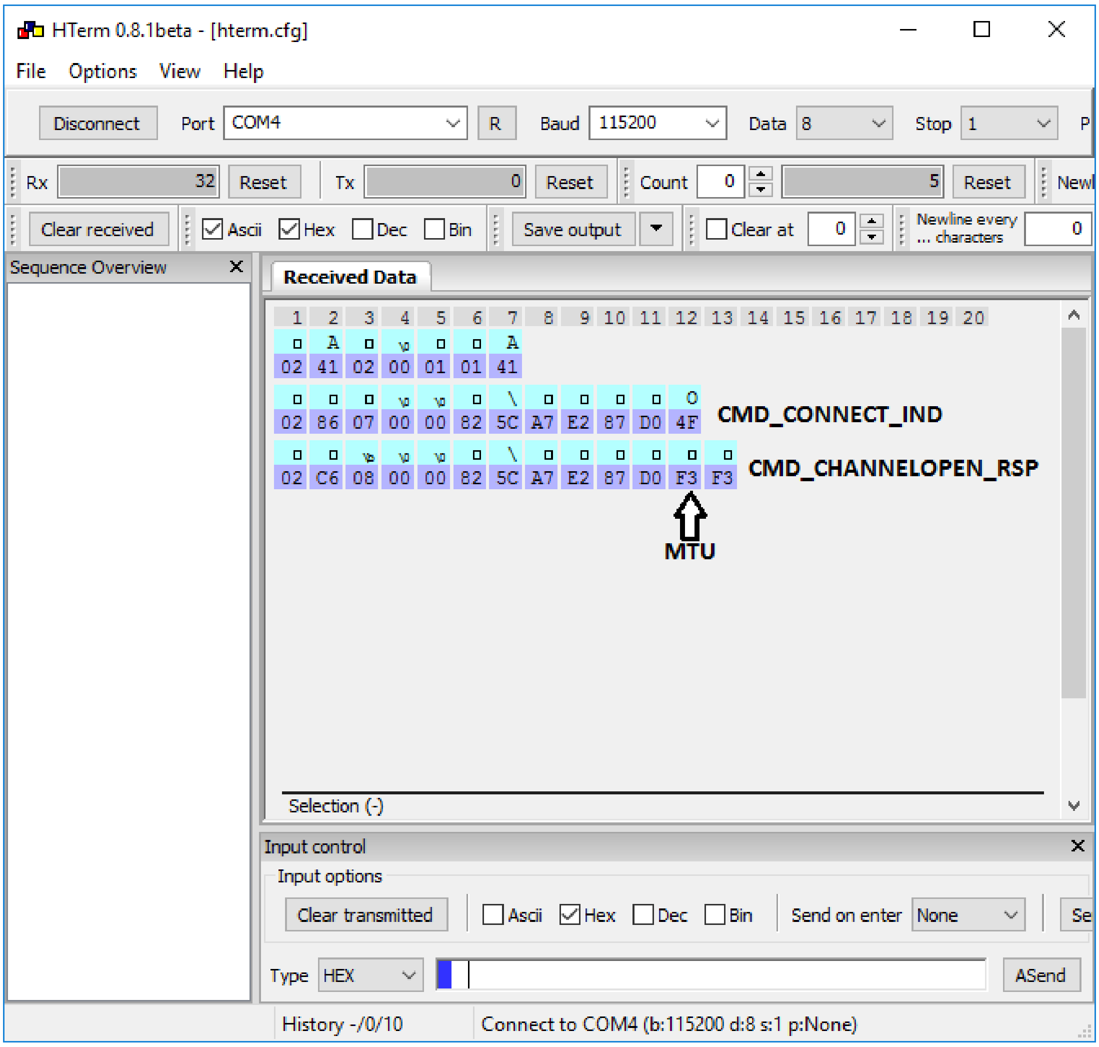
Android	iOS
• Please click on the menu bullets on the right and press "Request MTU" to request for a larger MTU.  53% 13:35 Devices DISCONNECT BONDED ADVERTISER A-000001 00:18:DA:00:00:01 ✕ CONNECTED BONDED CLIENT SERVER Generic Access UUID: 0x1800 PRIMARY SERVICE Generic Attribu UUID: 0x1801 PRIMARY SERVICE Unknown Servic UUID: 6e400001-c PRIMARY SERVICE Read all characteristics Enable services Read remote RSSI Request connection priority Request MTU Wireless by Nordic	• Please click on the "Unknown Service" to start the service discovery and the MTU request.  ●●●○○ Telekom.de 14:06 Back Services Disconnect Device: A-000001 Status: Connected Unknown Service UUID 6E400001-C352-11E5-953D-0002A5D5C51B PRIMARY SERVICE Wireless by Nordic Log

Android	iOS
- The Proteus module allows a MTU of up to 247 bytes, which results in a payload size of 243 bytes.  Set Maximum Transfer Unit MTU value: <23 – 517> 247 CANCEL OK	- The iOS App runs this step simultaneously in the background, a user-defined MTU is not possible.

<table>
<tr><th>Android</th><th>iOS</th></tr>
<tr><td>- Again click on the menu bullets on the right and press "Enable services" to enable the notifications.</td><td>- 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.</td></tr>
<tr><td>
53% 13:35
Devices DISCONNECT
BONDED ADVERTISER A-000001 00:18:DA:00:00:01
CONNECTED BONDED CLIENT SERVER
Generic Access UUID: 0x1800 PRIMARY SERVICE
Generic Attribu UUID: 0x1801 PRIMARY SERVICE
Unknown Servic UUID: 6e400001-c: PRIMARY SERVICE
Read all characteristics
Enable services
Read remote RSSI
Request connection priority
Request MTU
Wireless by Nordic</td><td>
●●●○○ Telekom.de 14:06
Services Characteristics
Device: A-000001
Status: Connected
Unknown Characteristic
UUID 6E400002-C352-11E5-953D-0002A5D5C51B
Properties Write WriteWithoutResponse
Value 0x1D
Descriptor false
Unknown Characteristic
UUID 6E400003-C352-11E5-953D-0002A5D5C51B
Properties Notify
Value 0x2A
Descriptor true
Wireless by Nordic
Log</td></tr>
<tr><td colspan="2">- As soon as the module has received the notification enable request the second LED on the Proteus evaluation board is turned on. Now you are fully connected and you can access the characteristics to transmit and receive data.</td></tr>
</table>

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 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.



HTerm 0.8.1beta - [hterm.cfg]

Port: COM4 · Baud: 115200 · Data: 8 · Stop: 1

Rx: 32 · Tx: 0 · Count: 0 · 5

☑ Ascii ☑ Hex □ Dec □ Bin · □ Clear at 0 · Newline every ... characters: 0

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
02 41 02 00 01 01 41
02 86 07 00 00 82 5C A7 E2 87 D0 4F        CMD_CONNECT_IND
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3     CMD_CHANNELOPEN_RSP
                                 ⇧
                                MTU
```

Selection (-)

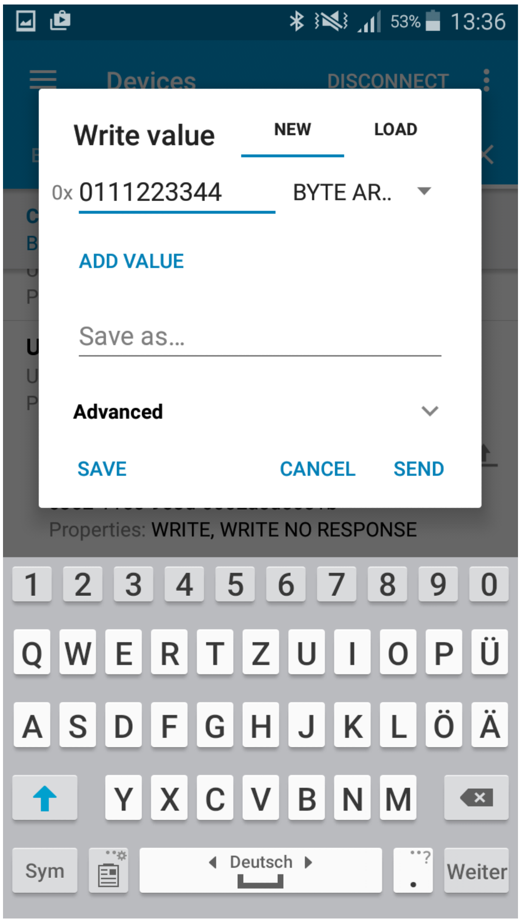
Input options: □ Ascii ☑ Hex □ Dec □ Bin · Send on enter: None · Type: HEX

History -/0/10 · Connect to COM4 (b:115200 d:8 s:1 p:None)

<table>
<tr><th>Android</th><th>iOS</th></tr>
<tr><td colspan="2">

- 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).

</td></tr>
<tr><td>

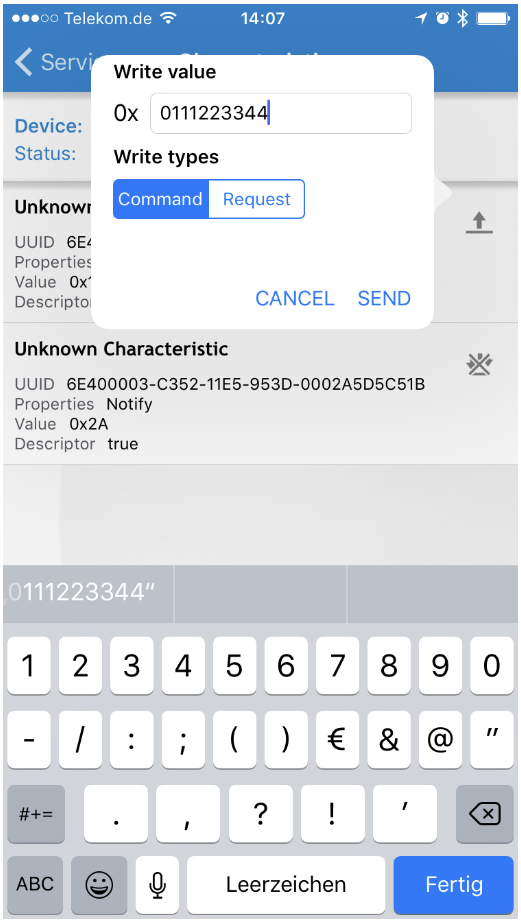


Write value

0x 0111223344 BYTE AR..

Properties: WRITE, WRITE NO RESPONSE

</td><td>



Write value

0x 0111223344

Write types: Command | Request

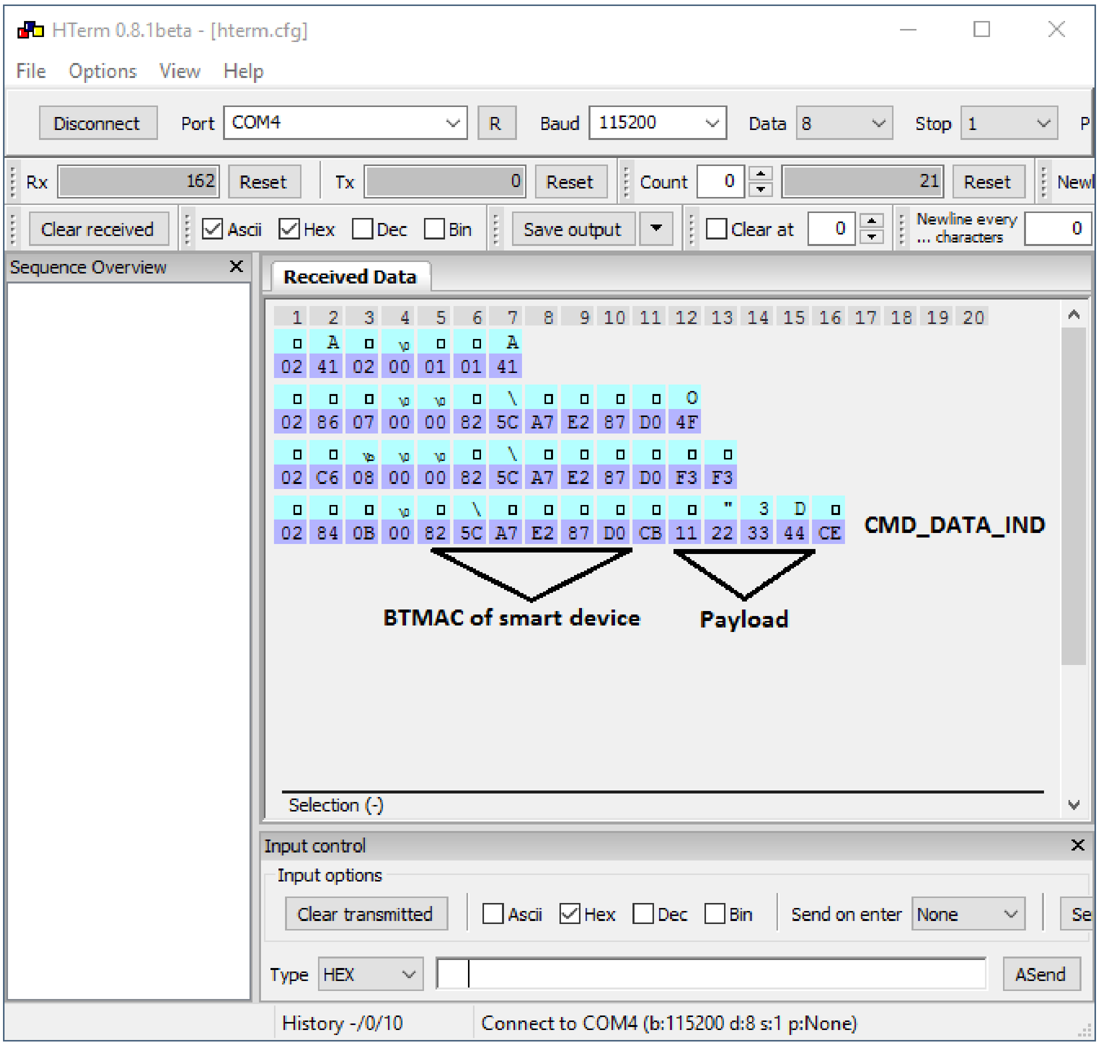
Unknown Characteristic
UUID 6E400003-C352-11E5-953D-0002A5D5C51B
Properties Notify
Value 0x2A
Descriptor true

</td></tr>
</table>

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



HTerm 0.8.1beta - [hterm.cfg]

Port COM4, Baud 115200, Data 8, Stop 1

Rx 162, Tx 0, Count 0, 21

☑ Ascii ☑ Hex ☐ Dec ☐ Bin

Received Data

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 □  A  □ \0  □  □  A
02 41 02 00 01 01 41

 □  □  □ \0 \0  □  \  □  □  □  □  O
02 86 07 00 00 82 5C A7 E2 87 D0 4F

 □  □  □ \0 \0  □  \  □  □  □  □  □  □
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3

 □  □  □ \0  □  \  □  □  □  □  □  □  "  3  D  □
02 84 0B 00 82 5C A7 E2 87 D0 CB 11 22 33 44 CE
```

Input options: ☐ Ascii ☑ Hex ☐ Dec ☐ Bin, Send on enter None, Type HEX

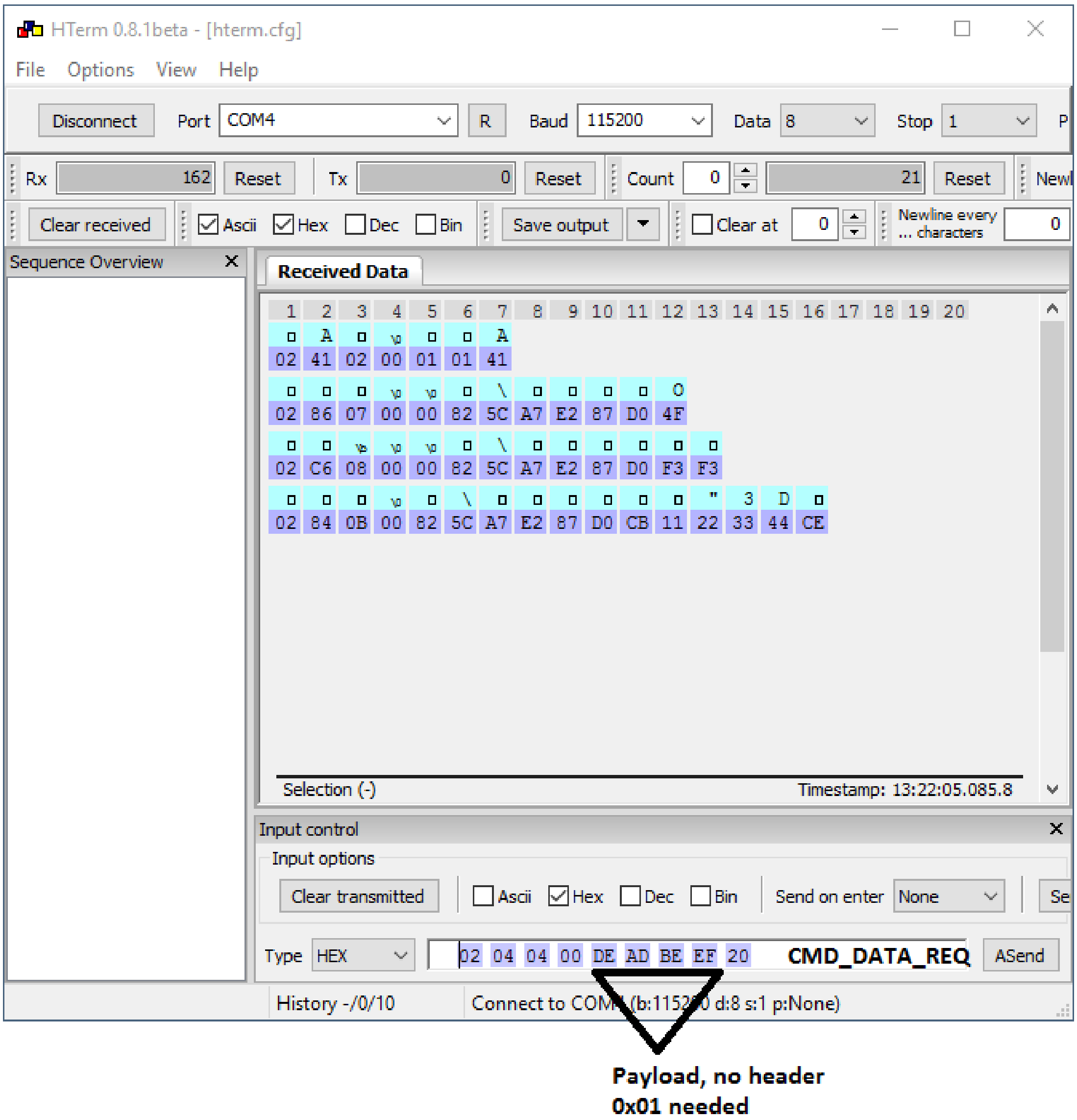
History -/0/10 Connect to COM4 (b:115200 d:8 s:1 p:None)

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.



Port COM4 Baud 115200 Data 8 Stop 1

Rx 162 Tx 0 Count 0 21

☑ Ascii ☑ Hex □ Dec □ Bin □ Clear at 0 Newline every ... characters 0

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
02 41 02 00 01 01 41
02 86 07 00 00 82 5C A7 E2 87 D0 4F
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3
02 84 0B 00 82 5C A7 E2 87 D0 CB 11 22 33 44 CE
```

Selection (-) Timestamp: 13:22:05.085.8

□ Ascii ☑ Hex □ Dec □ Bin Send on enter None

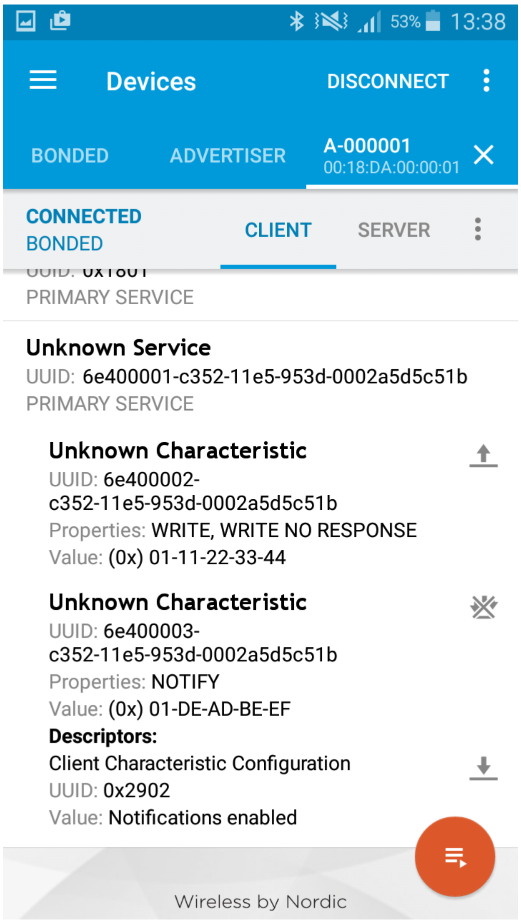
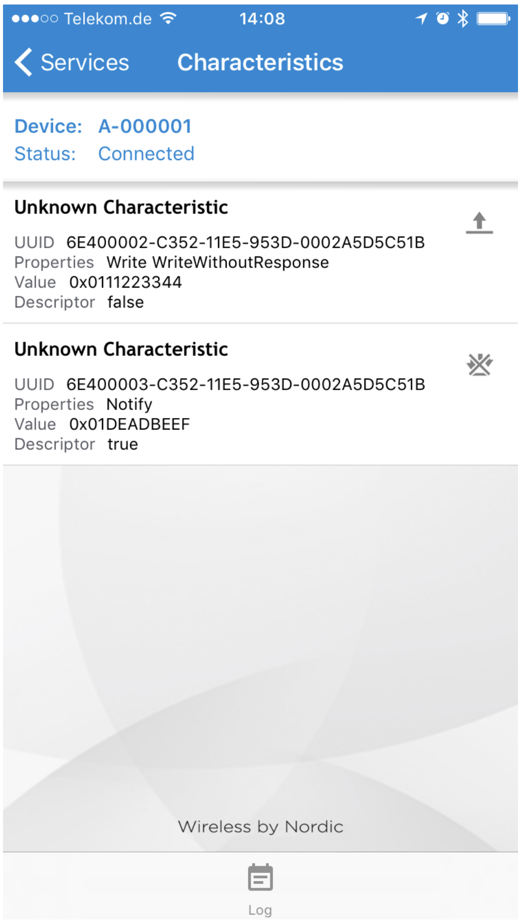
Type HEX `02 04 04 00 DE AD BE EF 20` **CMD_DATA_REQ**

History -/0/10 Connect to COM4 (b:115200 d:8 s:1 p:None)

Payload, no header 0x01 needed

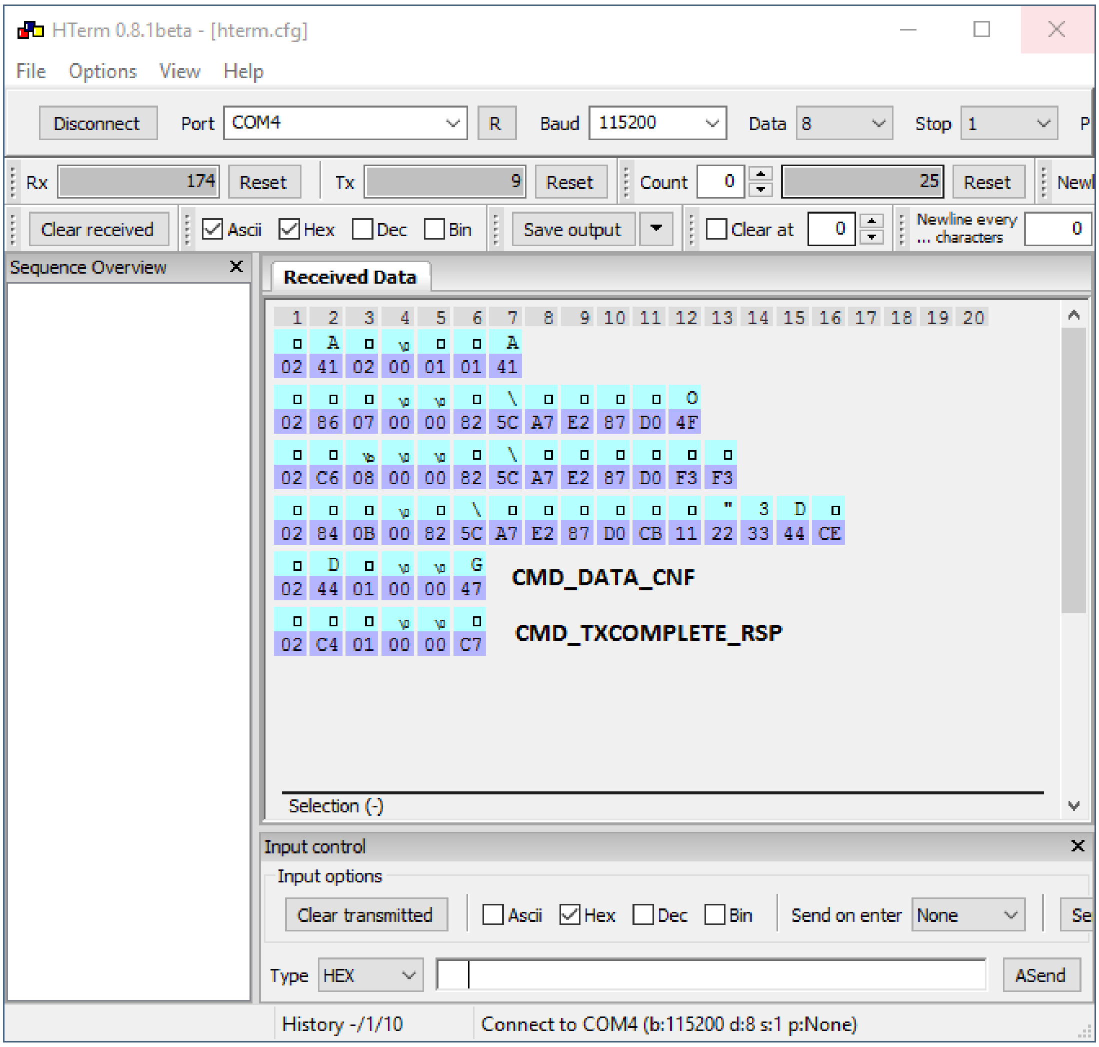
<table>
<tr><th>Android</th><th>iOS</th></tr>
<tr><td colspan="2">

- 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.

</td></tr>
<tr><td>

53% 13:38
Devices DISCONNECT
BONDED ADVERTISER A-000001 00:18:DA:00:00:01
CONNECTED
BONDED
CLIENT SERVER
UUID: 0x1801
PRIMARY SERVICE
Unknown Service
UUID: 6e400001-c352-11e5-953d-0002a5d5c51b
PRIMARY SERVICE
Unknown Characteristic
UUID: 6e400002-c352-11e5-953d-0002a5d5c51b
Properties: WRITE, WRITE NO RESPONSE
Value: (0x) 01-11-22-33-44
Unknown Characteristic
UUID: 6e400003-c352-11e5-953d-0002a5d5c51b
Properties: NOTIFY
Value: (0x) 01-DE-AD-BE-EF
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Value: Notifications enabled
Wireless by Nordic
</td><td>

●●●○○ Telekom.de 14:08
Services Characteristics
Device: A-000001
Status: Connected
Unknown Characteristic
UUID 6E400002-C352-11E5-953D-0002A5D5C51B
Properties Write WriteWithoutResponse
Value 0x0111223344
Descriptor false
Unknown Characteristic
UUID 6E400003-C352-11E5-953D-0002A5D5C51B
Properties Notify
Value 0x01DEADBEEF
Descriptor true
Wireless by Nordic
Log
</td></tr>
</table>

<table>
<tr><th>Android</th><th>iOS</th></tr>
<tr><td colspan="2">

- 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.



HTerm 0.8.1beta - [hterm.cfg]

Port COM4 Baud 115200 Data 8 Stop 1

Rx 174 Tx 9 Count 0 25

☑ Ascii ☑ Hex ☐ Dec ☐ Bin ☐ Clear at 0 Newline every ... characters 0

Received Data

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
02 41 02 00 01 01 41
02 86 07 00 00 82 5C A7 E2 87 D0 4F
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3
02 84 0B 00 82 5C A7 E2 87 D0 CB 11 22 33 44 CE
02 44 01 00 00 47    CMD_DATA_CNF
02 C4 01 00 00 C7    CMD_TXCOMPLETE_RSP
```

Input options: ☐ Ascii ☑ Hex ☐ Dec ☐ Bin Send on enter None

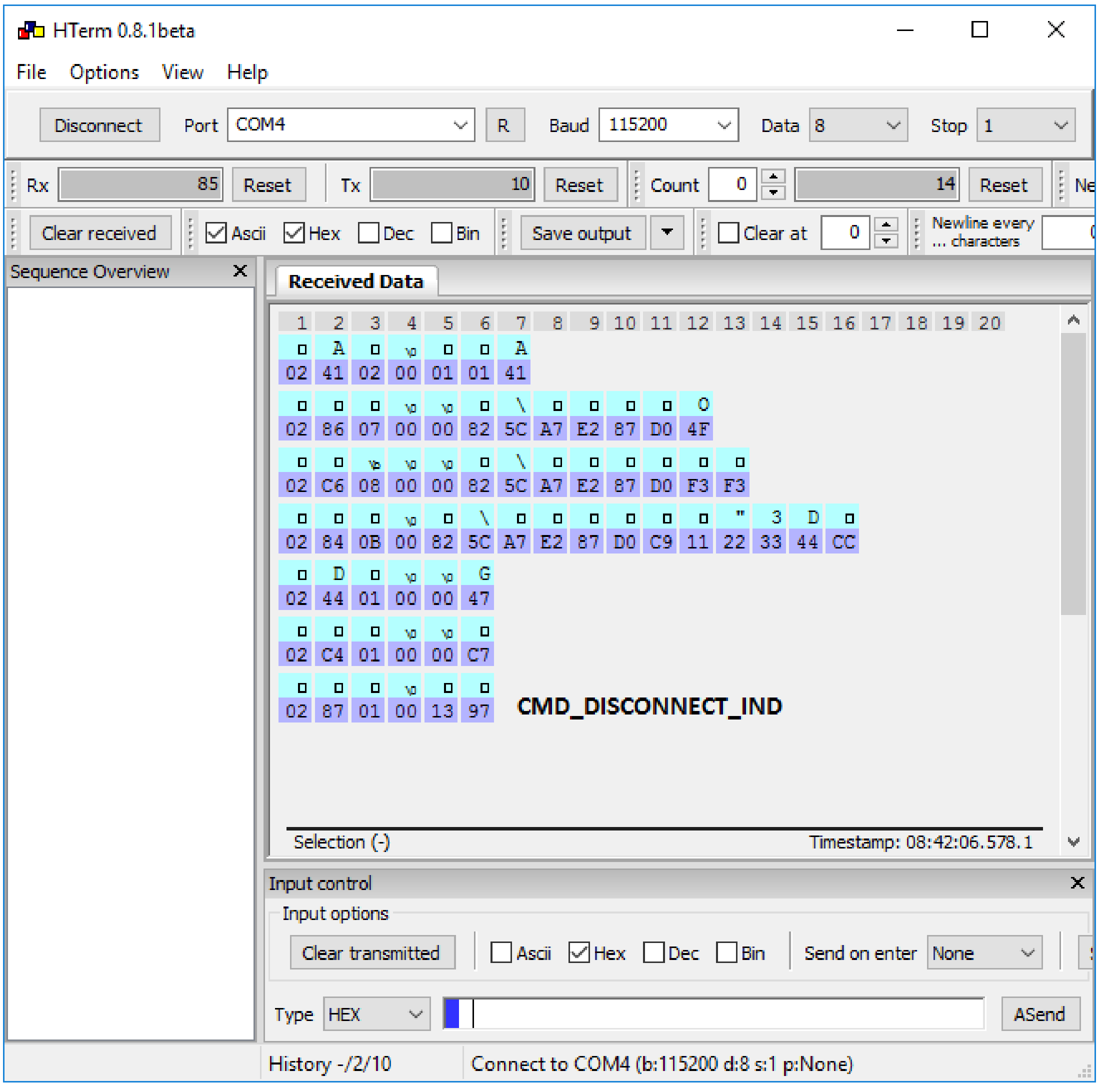
Type HEX

History -/1/10 Connect to COM4 (b:115200 d:8 s:1 p:None)

</td></tr>
</table>

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.



Port COM4 Baud 115200 Data 8 Stop 1

Rx 85 Tx 10 Count 0 14

☑ Ascii ☑ Hex □ Dec □ Bin □ Clear at 0 Newline every ... characters

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 □  A  □     □  □  A
02 41 02 00 01 01 41
 □  □  □        □  \  □  □  □  □  O
02 86 07 00 00 82 5C A7 E2 87 D0 4F
 □  □           □  \  □  □  □  □  □  □
02 C6 08 00 00 82 5C A7 E2 87 D0 F3 F3
 □  □  □     □  \  □  □  □  □  □  □  "  3  D  □
02 84 0B 00 82 5C A7 E2 87 D0 C9 11 22 33 44 CC
 □  D  □        G
02 44 01 00 00 47
 □  □  □        □
02 C4 01 00 00 C7
 □  □  □     □  □
02 87 01 00 13 97   CMD_DISCONNECT_IND
```

Selection (-) Timestamp: 08:42:06.578.1

Input options: □ Ascii ☑ Hex □ Dec □ Bin Send on enter None

Type HEX

History -/2/10 Connect to COM4 (b:115200 d:8 s:1 p:None)

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

5 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. Proteus Connect app for Android. https://play.google.com/store/apps/details?id=com.eisos.android.terminal.

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

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

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

6 Important notes

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

6.1 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.

6.2 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.

6.3 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.

6.4 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.

6.5 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 6.1 and 6.2 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.

6.6 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.

6.7 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.

6.8 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*.

7 Legal notice

7.1 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.

7.2 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.

7.3 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.

7.4 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.

8 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.

8.1 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.

8.2 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.

8.3 Ownership

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

8.4 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.

8.5 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.

8.6 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.

8.7 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.

8.8 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.

8.9 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