

ANR025

PROTEUS-E QUICK START

VERSION 1.3

SEPTEMBER 9, 2024

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

Revision history

Manual version	Notes	Date
1.0	<ul style="list-style-type: none">• Initial version	February 2022
1.1	<ul style="list-style-type: none">• Updated Important notes, meta data and document style	July 2023
1.2	<ul style="list-style-type: none">• Replaced terminal program hterm by Smart Commander [1]• Updated images of Proteus Connect App to most recent version [2, 3]• Updated the links to the mobile apps	October 2023
1.3	<ul style="list-style-type: none">• Updated name of Smart Commander PC tool and Proteus Connect app	September 2024

Abbreviations

Abbreviation	Name	Description
BTMAC		Bluetooth® conform MAC address of the module used on the RF-interface.
CS	Checksum	Byte wise XOR combination of the preceding fields.
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	System-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	4
2	Prerequisites	4
3	General information	5
3.1	How to choose the operation mode?	5
3.2	General connection setup information	5
3.3	Transparent mode: Preconfiguring of the module	6
4	Transparent mode: Quickstart	8
4.1	Smart phone using nRFConnect app as central device	8
4.2	Smart phone using WE Bluetooth LE Terminal app as central device	17
4.3	Proteus module or USB radio stick as central device	22
4.3.1	Using WE UART Terminal to run the workflow	24
5	Command mode: Quickstart	26
5.1	Smart phone using nRFConnect app as central device	26
5.2	Smart phone using WE Bluetooth LE Terminal app as central device	37
5.3	Proteus module or USB radio stick as central device	45
5.3.1	Using WE UART Terminal to run the workflow	46
6	References	49
7	Important notes	50

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

To run the following example you need to have the subsequent requisites:

- A Proteus-e EV-Board in factory state.
- A central device that initiates the connection setup. For example
 - a smart phone with Bluetooth® LE function and the WE Bluetooth LE Terminal App [2, 3] or Nordic Semiconductor nRF Connect App [4, 5]
 - a Proteus-I,-II,-III EV-Board, mini EV-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.

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 [6]

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 ≤ 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 EV-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.

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.

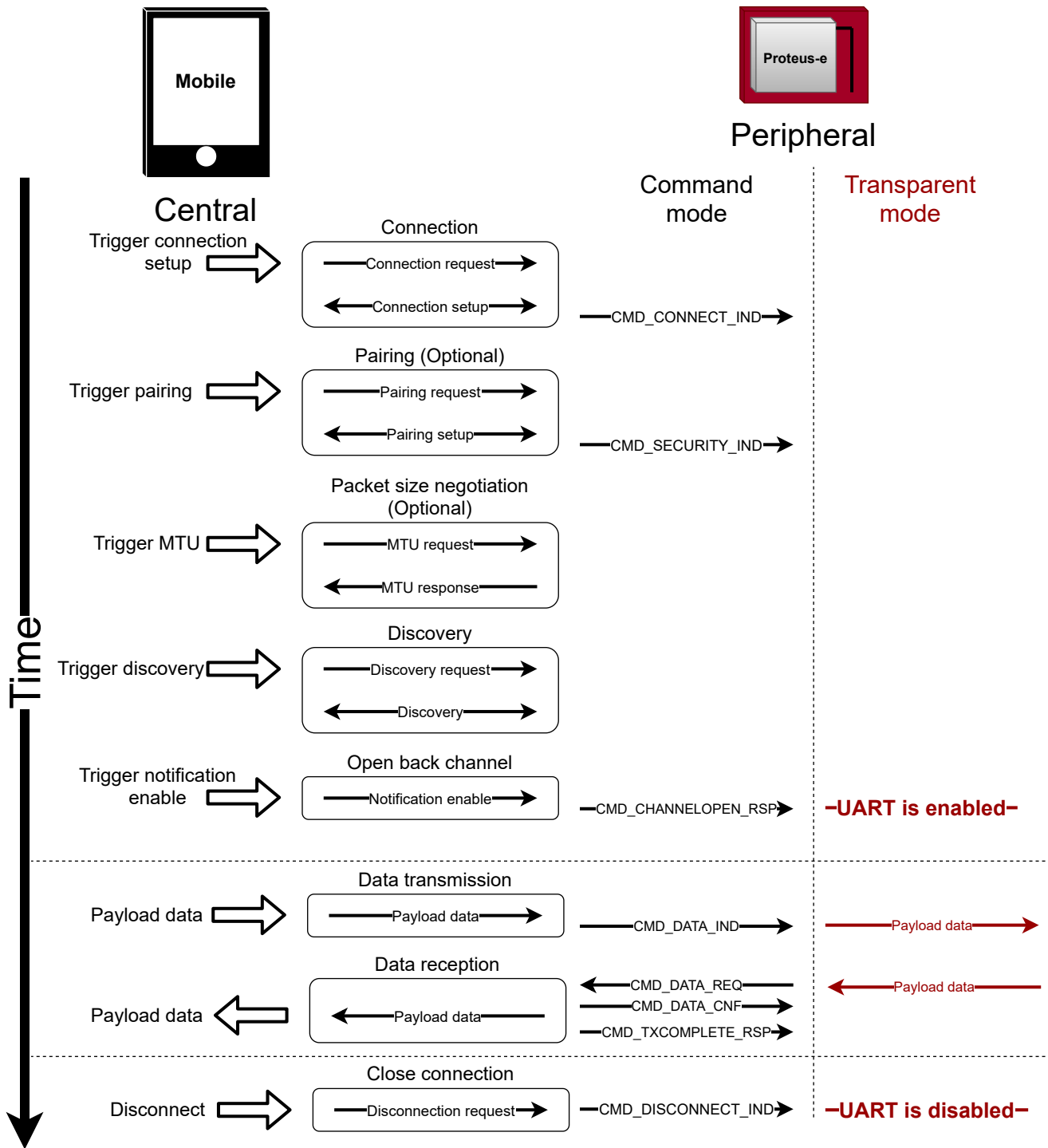


Figure 1: Steps for the connection setup

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.

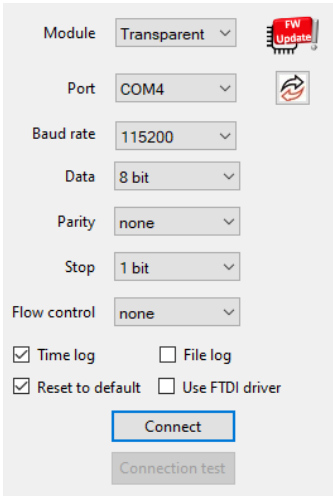
4.1 Smart phone using nRFConnect app as central device

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.



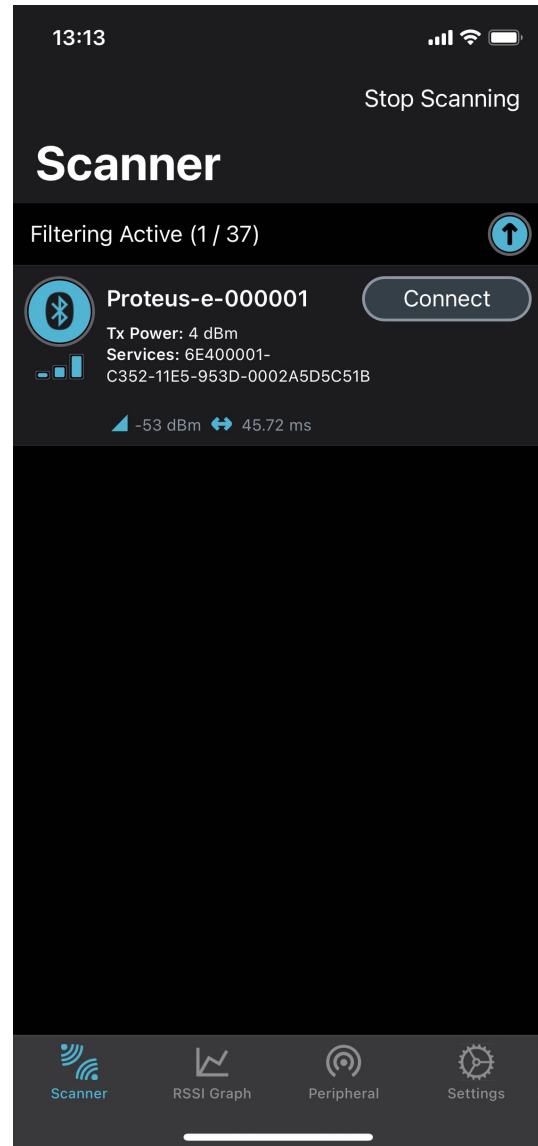
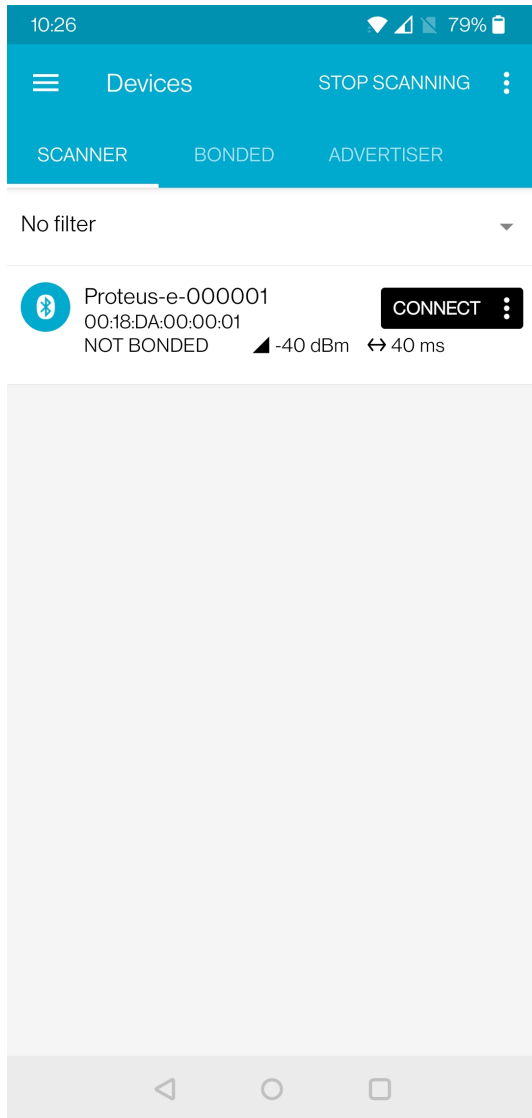
The nRF Connect App is an open source App providing standard Bluetooth® LE functions for iOS as well as for Android devices.

Please perform the following steps:

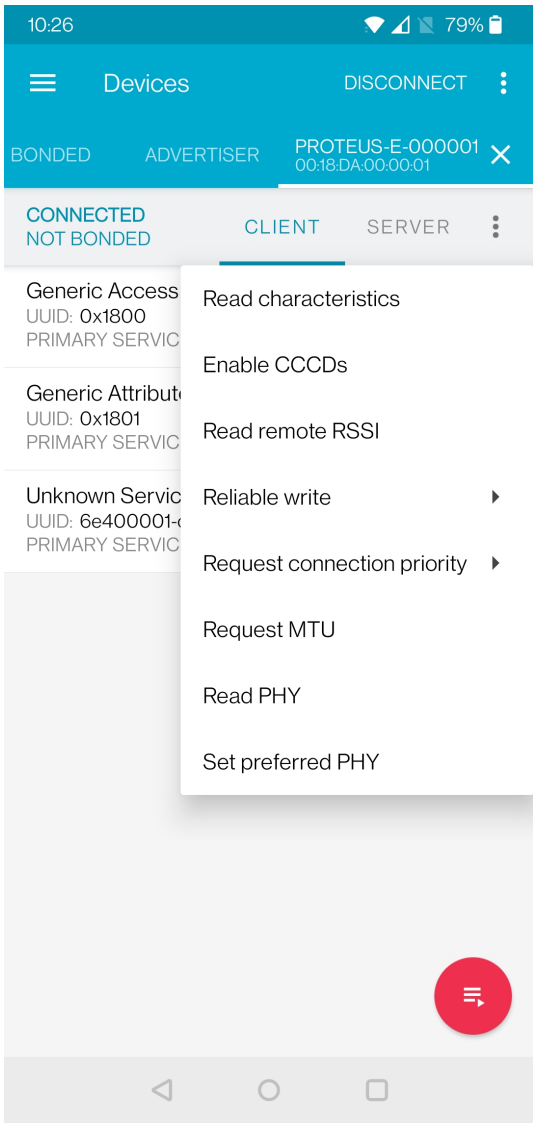
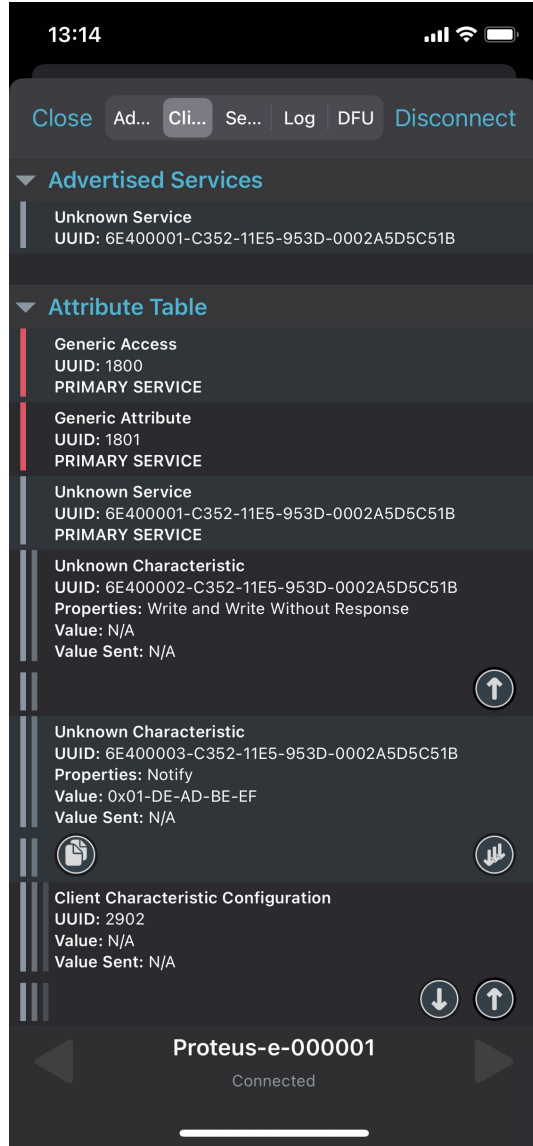
Android	iOS
<ul style="list-style-type: none"> • Connect the Proteus EV-Board to a host. In this application note, we assume that a Windows PC and the PC tool WE UART Terminal [1] is used. For Proteus-e EV-Board this can be simply achieved by using a simple USB cable to connect it to a PC. Set the module into transparent mode as described in chapter 3.1. Initially, the module is advertising. Thus, the Proteus-e <i>LED_1</i> is blinking slowly. • Start the PC tool, select the right module ("Transparent") and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing the "Connect" button. 	
	

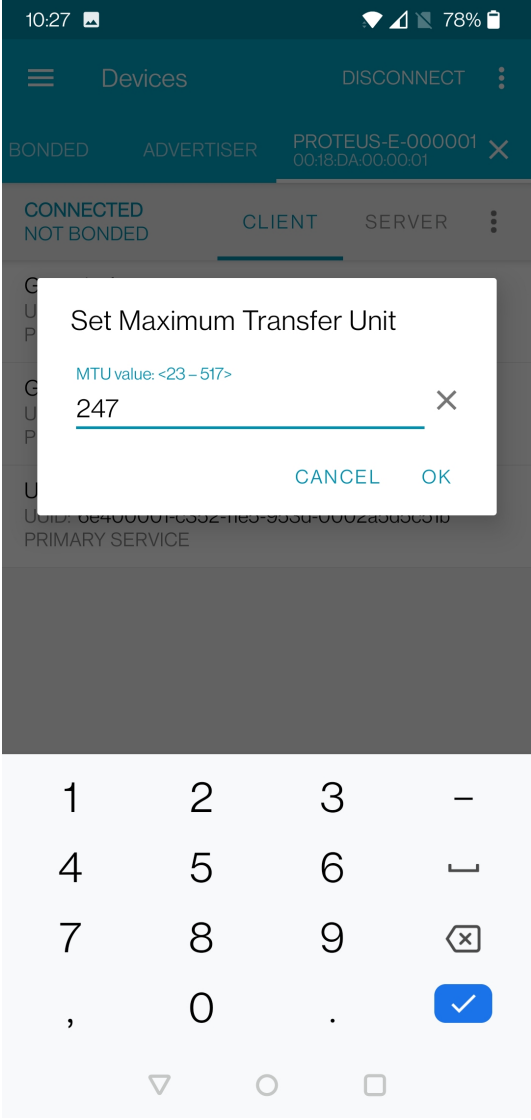
Android	iOS
---------	-----

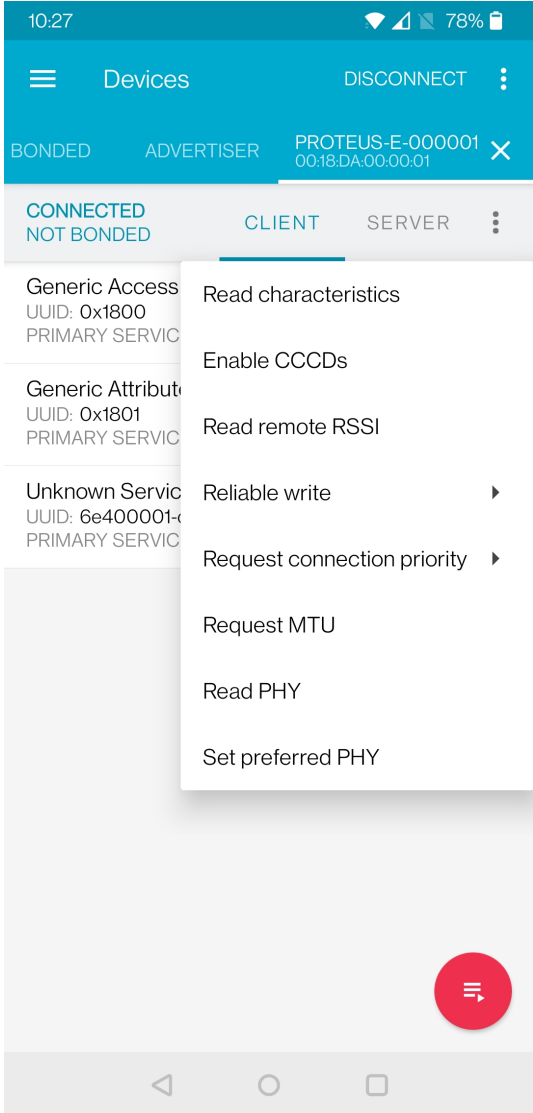
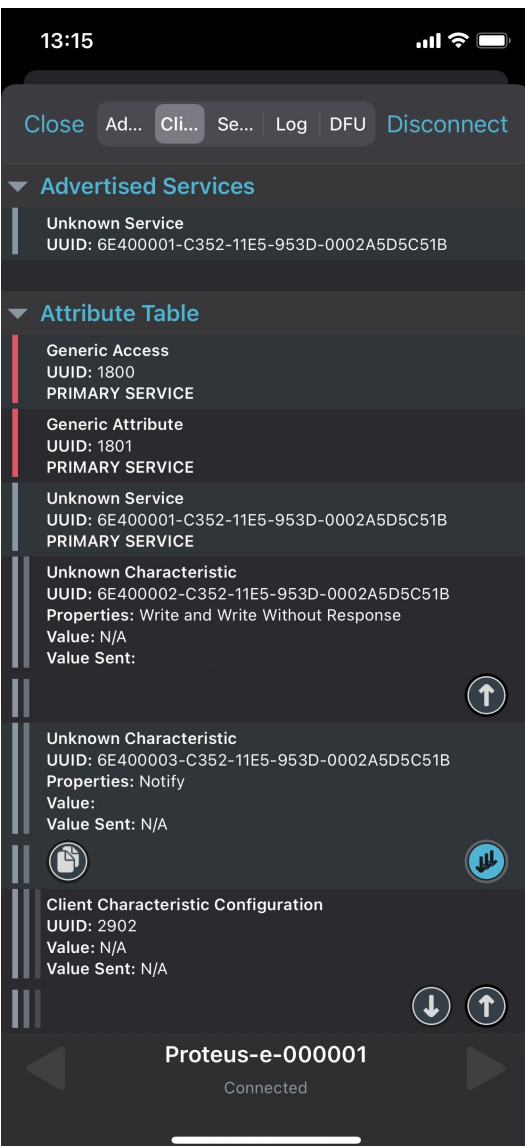
- Start your smart phone, enable the Bluetooth® LE feature and start the nRF Connect App.
- Press "SCAN" to find the module on the radio.
- When the module appears, press the "Connect" button.

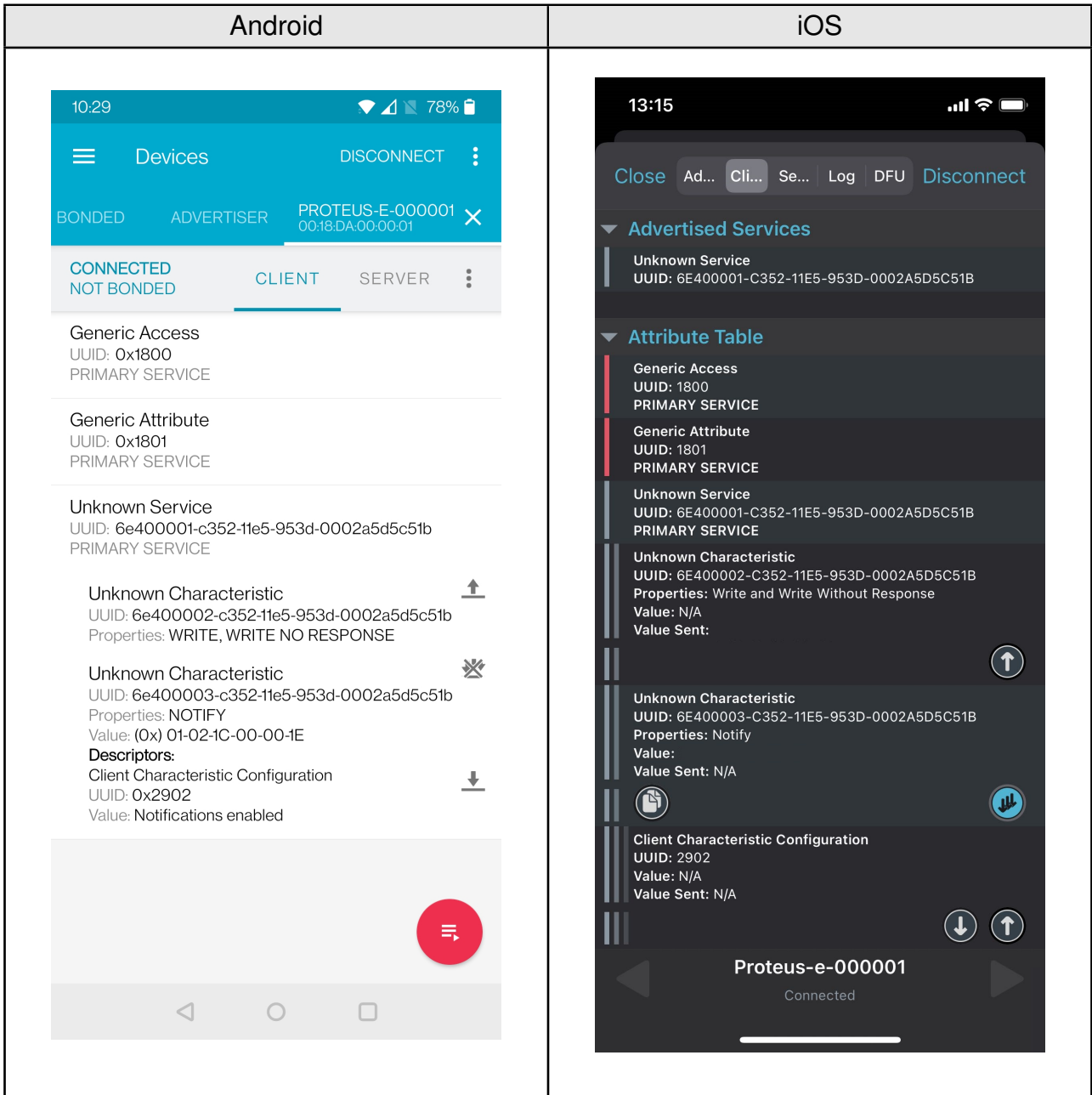


- As soon as the module has received the connection request, the module *LED_1* will blink faster.

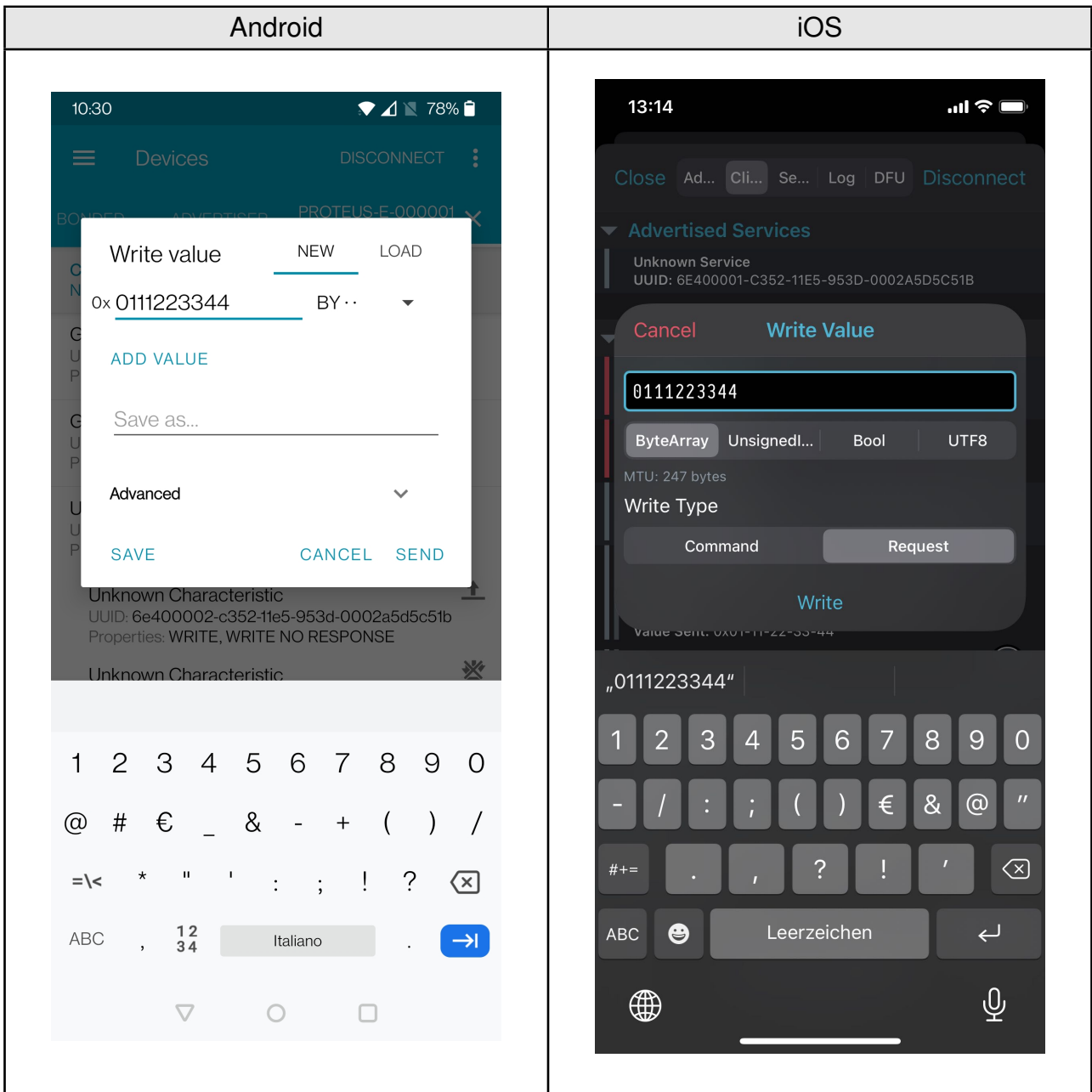
Android	iOS
<ul style="list-style-type: none"> Please click on the menu bullets on the right and press "Request MTU" to request for a larger MTU. 	<ul style="list-style-type: none"> Please click on the "Unknown Service" to start the service discovery and the MTU request. 

Android	iOS
<ul style="list-style-type: none">The Proteus-e allows an MTU of up to 247 bytes, which results in a maximum payload size (MPS) of 243 bytes.  <p>The screenshot shows the Proteus-e Android application interface. At the top, it says 'Devices' and 'DISCONNECT'. Below that, it shows 'BONDED' and 'ADVERTISER' with the name 'PROTEUS-E-000001' and MAC address '00:18:DA:00:00:01'. A modal dialog box titled 'Set Maximum Transfer Unit' is open, showing 'MTU value: <23 - 517>' and a text input field containing '247'. There are 'CANCEL' and 'OK' buttons at the bottom of the dialog. Below the dialog is a numeric keypad with a blue checkmark button.</p>	<ul style="list-style-type: none">The iOS App runs this step simultaneously in the background, a user-defined MTU is not possible.

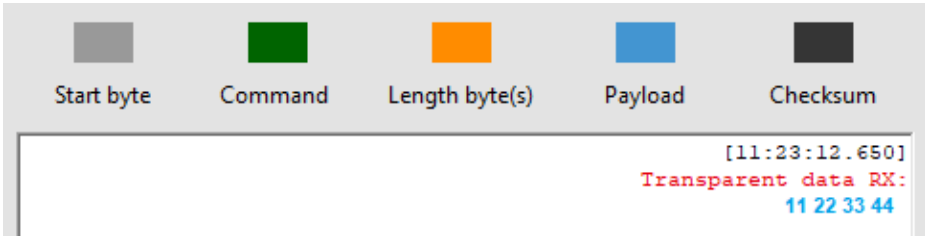
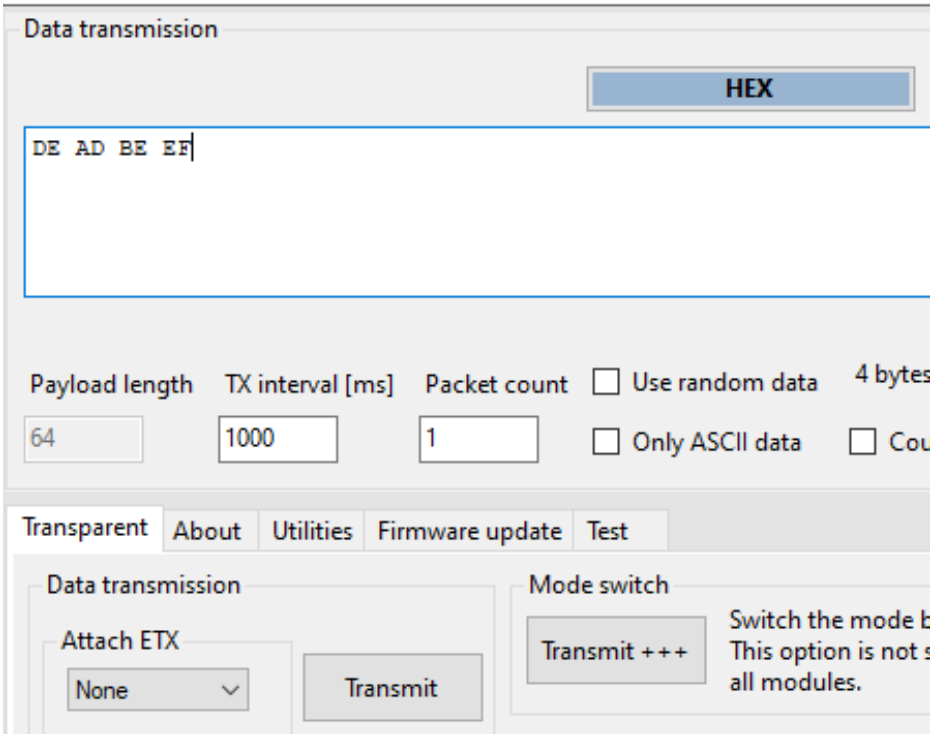
Android	iOS
<ul style="list-style-type: none"> Again click on the menu bullets on the right and press "Enable services"/"Enable CCCDs" to enable the notifications. 	<ul style="list-style-type: none"> Press the arrows on the RX-characteristic 6E400003- C352-11E5- 953D -0002A5D5C51B to enable the notifications. 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. 
<ul style="list-style-type: none"> As soon as the module has received the notification enable request the Proteus-e LED_1 is static on. 	

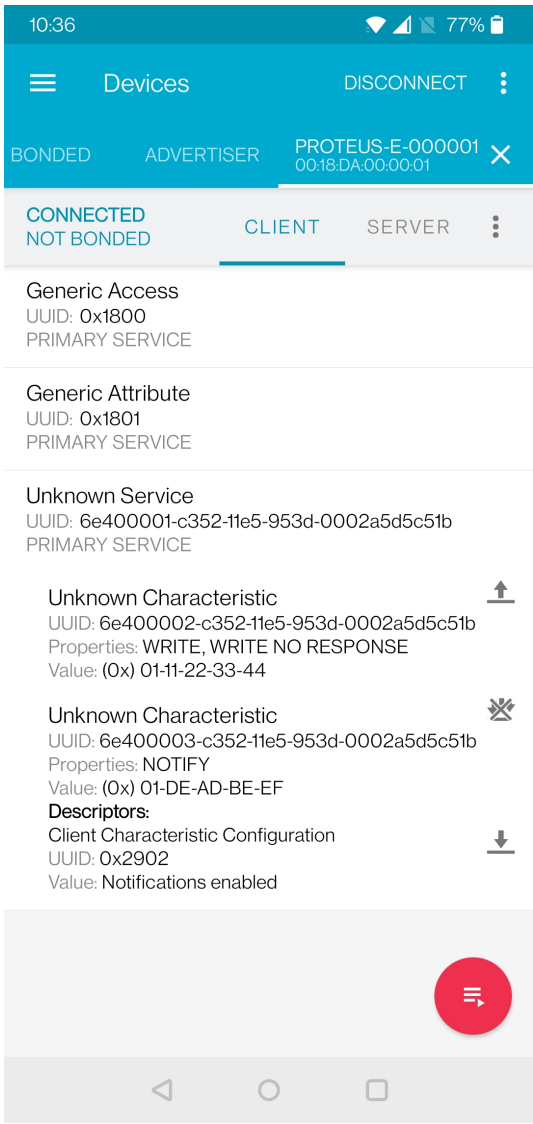
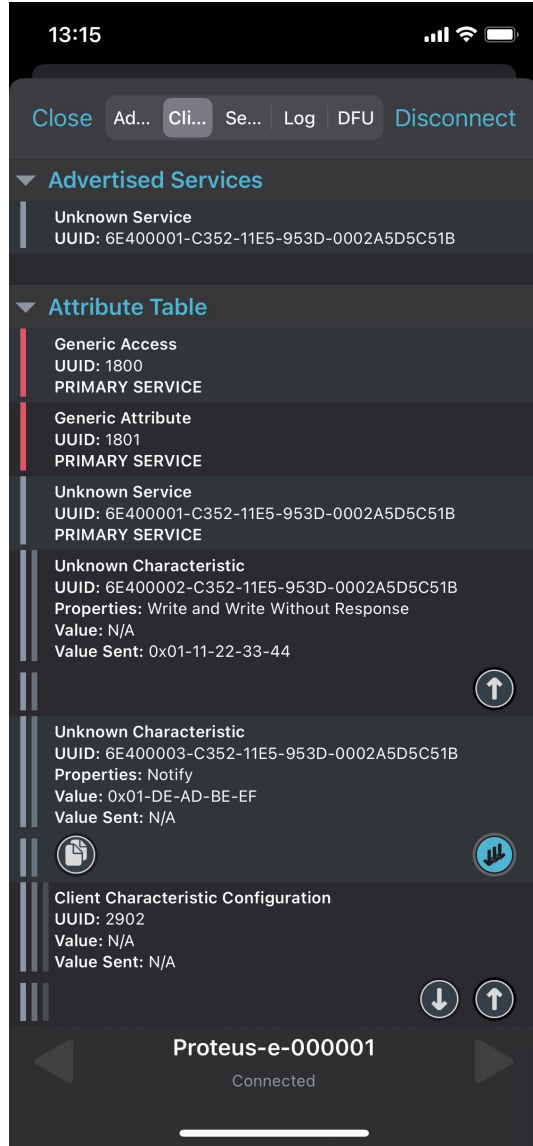


- Now you are fully connected and you can access the characteristics. The maximum size of payload depends on the chosen MTU size. Here we chose 247 bytes, which allows us to send 243 bytes of payload (MPS) via the channel.
- To send data to the Proteus-e, press the arrow next to the TX-characteristic 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.



- 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
	
<ul style="list-style-type: none"> To send back data, simply enter your payload in the respective field and press "Transmit" button. 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
<ul style="list-style-type: none"> 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. 	
 <p>The screenshot shows the Proteus-e Android app interface. At the top, it displays the time 10:36, signal strength, Wi-Fi, and battery at 77%. The main header shows 'Devices' and a 'DISCONNECT' button. Below this, there's a section for 'BONDED' devices, showing 'PROTEUS-E-000001' with advertiser '00:18:DA:00:00:01'. A 'CONNECTED' indicator is visible. The main list shows services: 'Generic Access' (UUID: 0x1800), 'Generic Attribute' (UUID: 0x1801), 'Unknown Service' (UUID: 6e400001-c352-11e5-953d-0002a5d5c51b), and two 'Unknown Characteristic' entries with their respective UUIDs and properties. The bottom navigation bar is visible.</p>	 <p>The screenshot shows the Proteus-e iOS app interface. At the top, it displays the time 13:15, signal strength, Wi-Fi, and battery. The main header shows 'Close', 'Ad...', 'Cli...', 'Se...', 'Log', 'DFU', and 'Disconnect' buttons. Below this, there's a section for 'Advertised Services' showing 'Unknown Service' (UUID: 6E400001-C352-11E5-953D-0002A5D5C51B). The 'Attribute Table' section lists services: 'Generic Access' (UUID: 1800), 'Generic Attribute' (UUID: 1801), 'Unknown Service' (UUID: 6E400001-C352-11E5-953D-0002A5D5C51B), and two 'Unknown Characteristic' entries with their respective UUIDs and properties. The bottom navigation bar is visible.</p>

4.2 Smart phone using WE Bluetooth LE Terminal app as central device

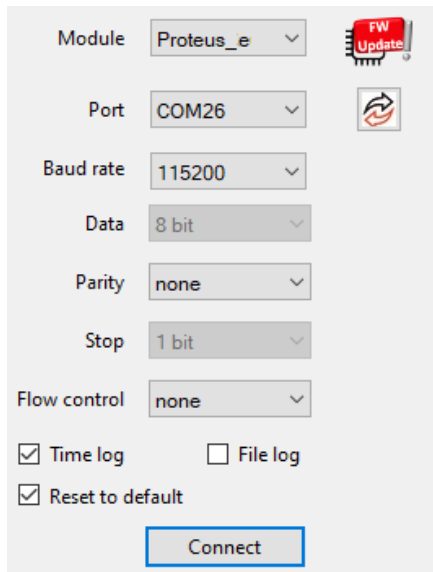
This chapter describes how to setup a connection to the Proteus-e radio module in transparent mode, when a smart phone and the WE Bluetooth LE Terminal App are used.



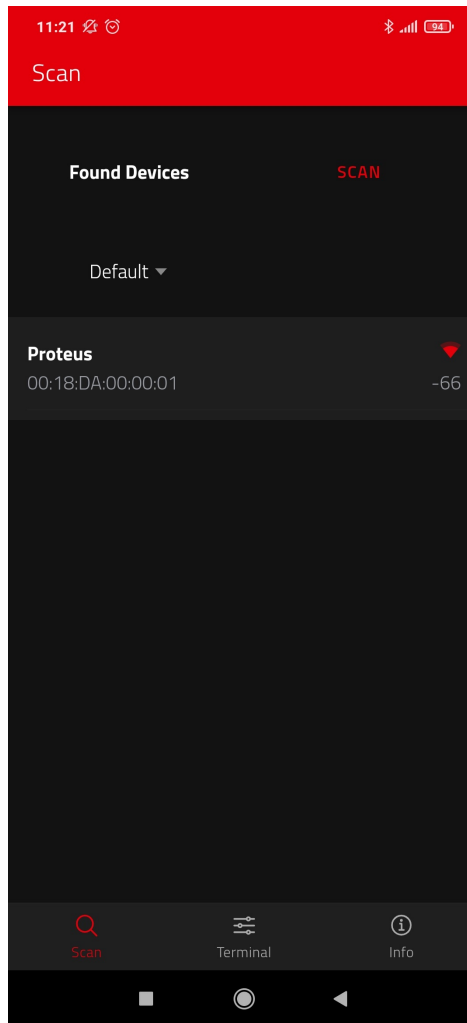
The WE Bluetooth LE Terminal App for iOS and Android is provided by Würth Elektronik eiSos as executable [2, 3] as well as source code [7].

Please perform the following steps:

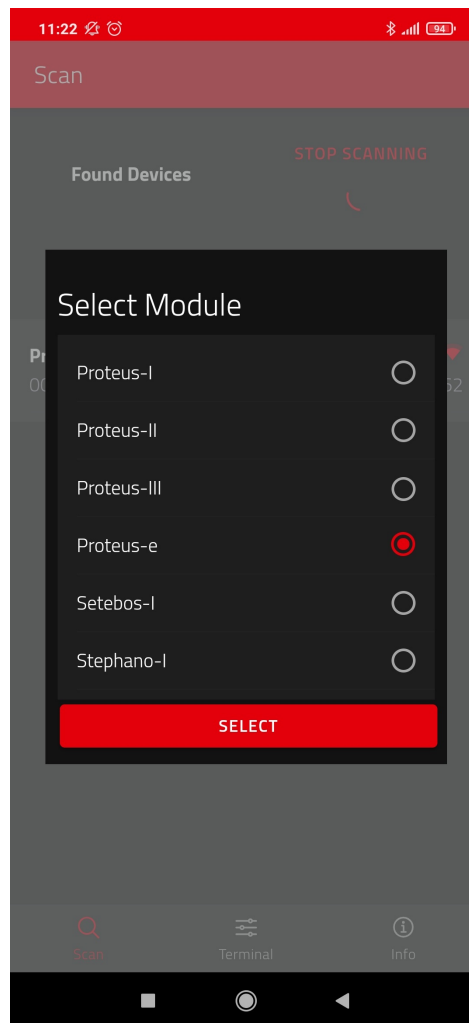
1. Connect the Proteus EV-Board to a host.
In this application note, we assume that a Windows PC and the PC tool WE UART Terminal [1] is used. For Proteus-e EV-Board this can be simply achieved by using a simple USB cable to connect it to a PC. Set the module into transparent mode as described in chapter 3.1. Initially, the module is advertising. Thus, the Proteus-e *LED_1* is blinking slowly.
2. Start the PC tool, select the right module ("Proteus-e") and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing the "Connect" button.



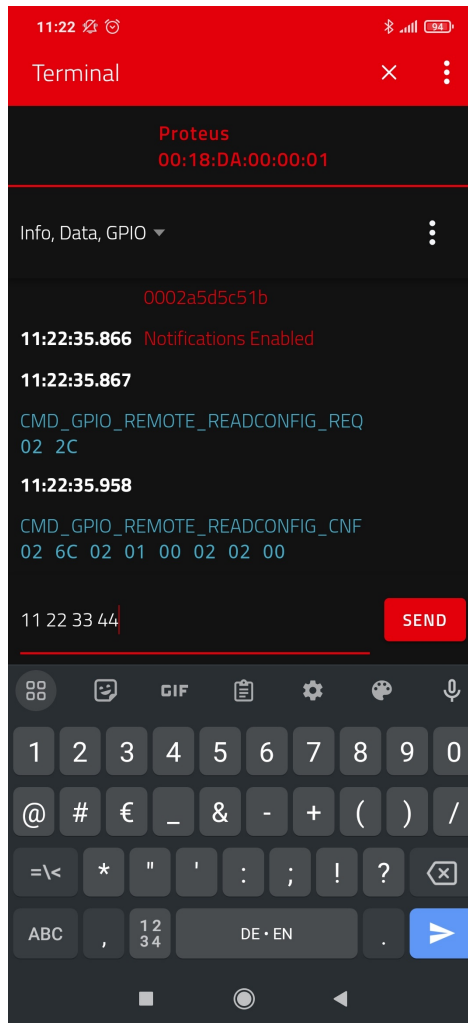
3. Start your smart phone, enable the Bluetooth® LE feature and start the WE Bluetooth LE Terminal App. Please note that Bluetooth® LE function of Android devices is only available if the location services are enabled in addition.
4. Press "Scan" to find the module on the radio.



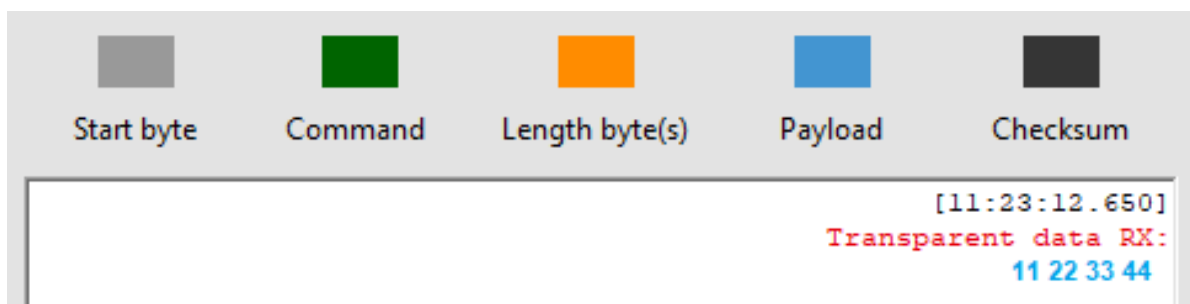
5. When the module appears, click on it. A pop-up will come up, where you need to select the current module type.



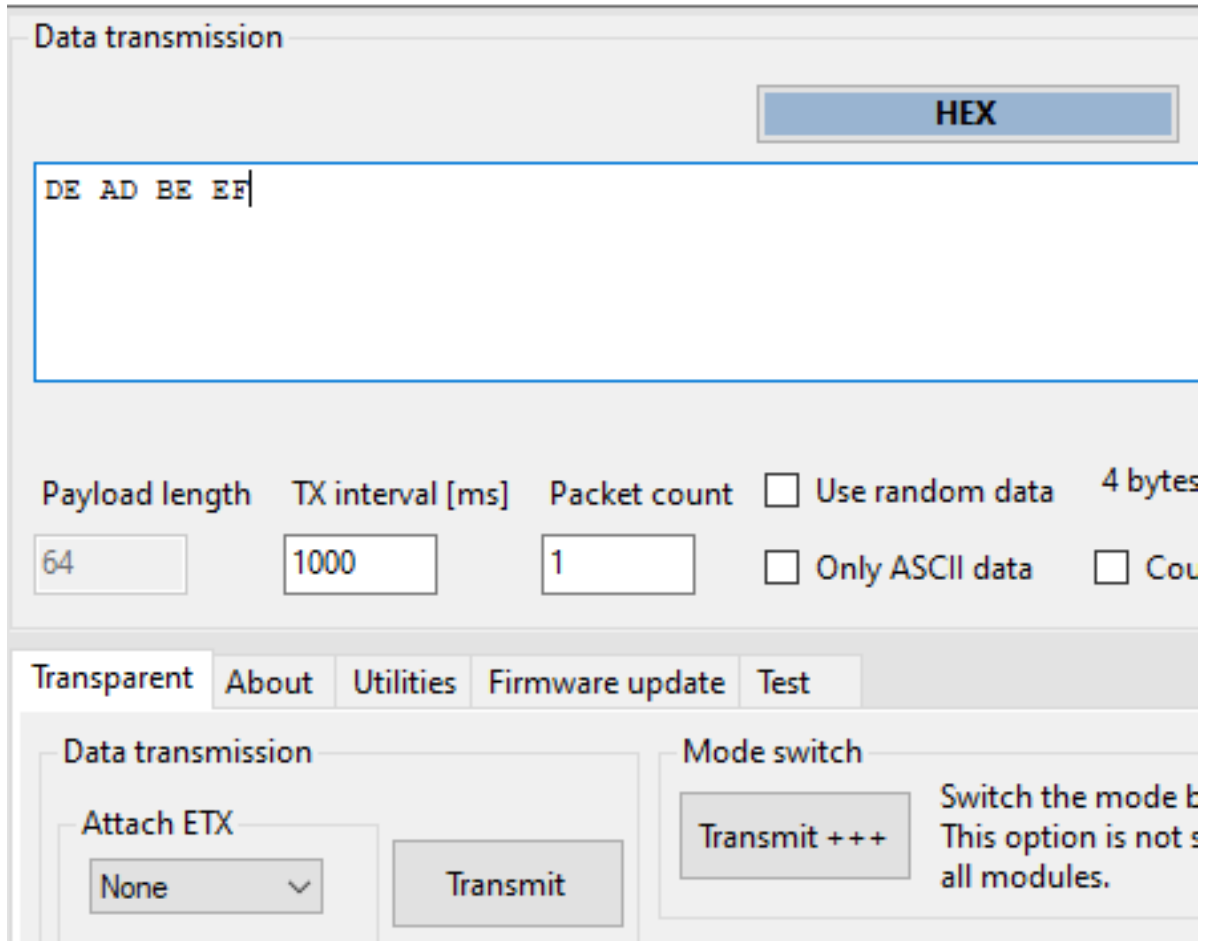
As soon as the connection has been setup successfully *LED_1* is turned static on. Now data can be transmitted in both directions.



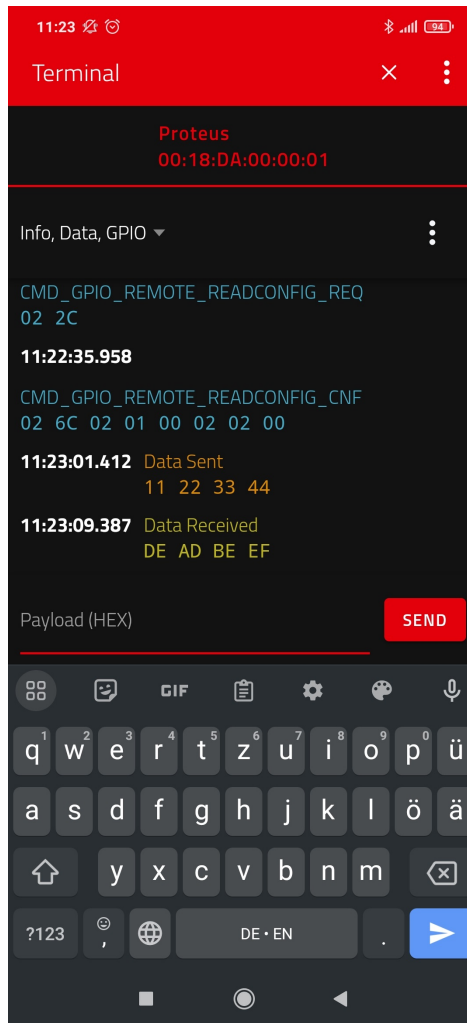
- 6. First of all, we want to send data from the smart phone to the radio module. To do so, enter your payload (for example 0x11 0x22 0x33 0x44) and press "SEND". The maximum payload size (MPS) is dependent on the MTU that was negotiated in the connection process. The smallest supported MTU for all Bluetooth® 4.0 (or newer) devices results in a max payload size (MPS) of 19 bytes. iOS and Android usually allow up to 243 bytes.
- 7. 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.



- To send back data simply enter your payload in the respective field and click on "Transmit". 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 the maximum payload size (MPS) must be respected.



- 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.



4.3 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	

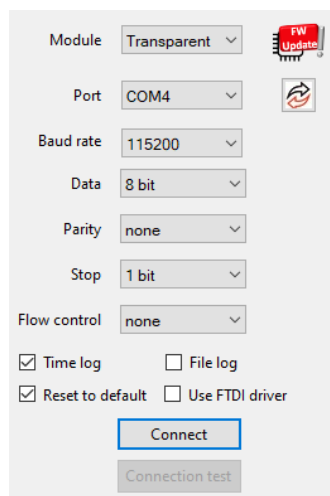
4. Now **Proteus_central** closes the connection.

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	

4.3.1 Using WE UART Terminal to run the workflow

The above work flow can be easily applied using the WE UART Terminal [1] PC tool.

1. First open two instances of the WE UART Terminal.
2. On each instance, select the right module type ("Transparent" on the Proteus-e instance, Proteus-I,-II or -III on the central instance) and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing the "Connect" button.



3. Then run the above workflow by clicking on the respective buttons in WE UART Terminal:

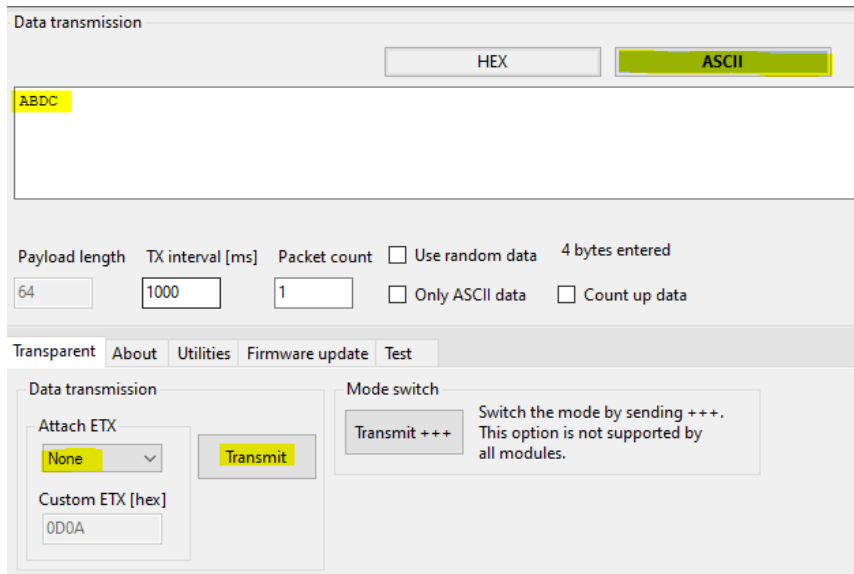


Figure 2: Important fields and buttons of Transparent (Proteus-e) instance of WE UART Terminal

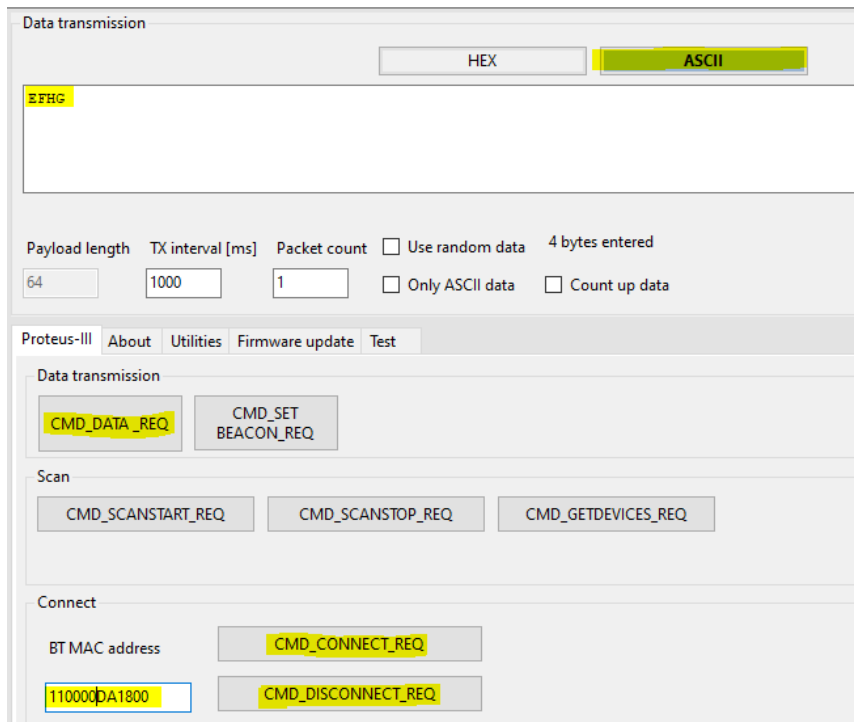


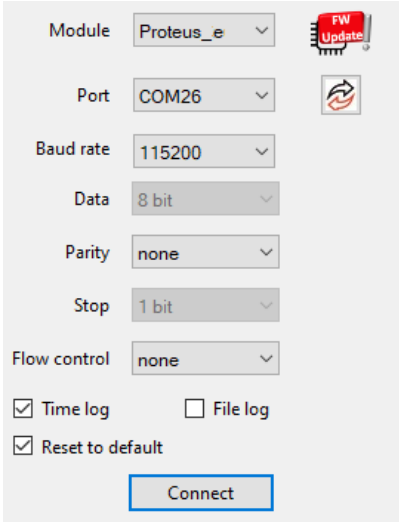
Figure 3: Important fields and buttons of central device instance of WE UART Terminal

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.

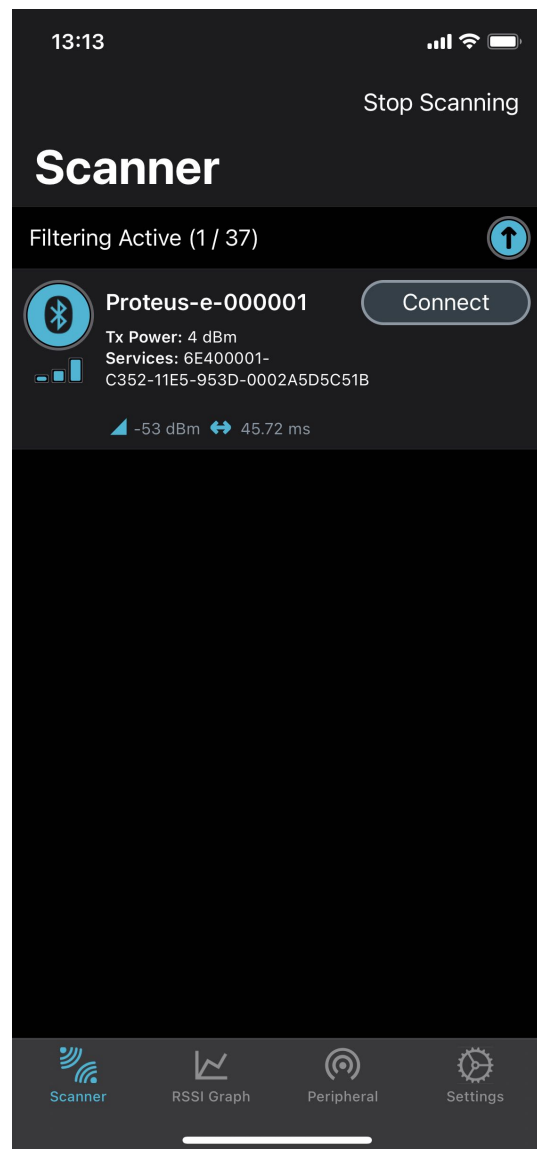
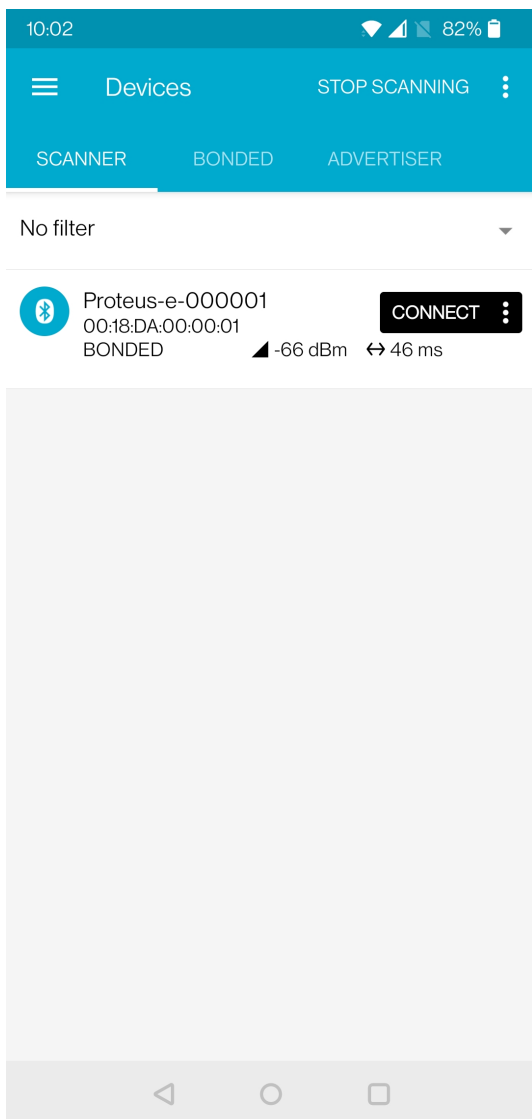
5.1 Smart phone using nRFConnect app as central device

This chapter describes how to setup a connection to the Proteus module in command mode, when a smart phone and the **nRF Connect App** [4, 5] are used. Please perform the following steps:

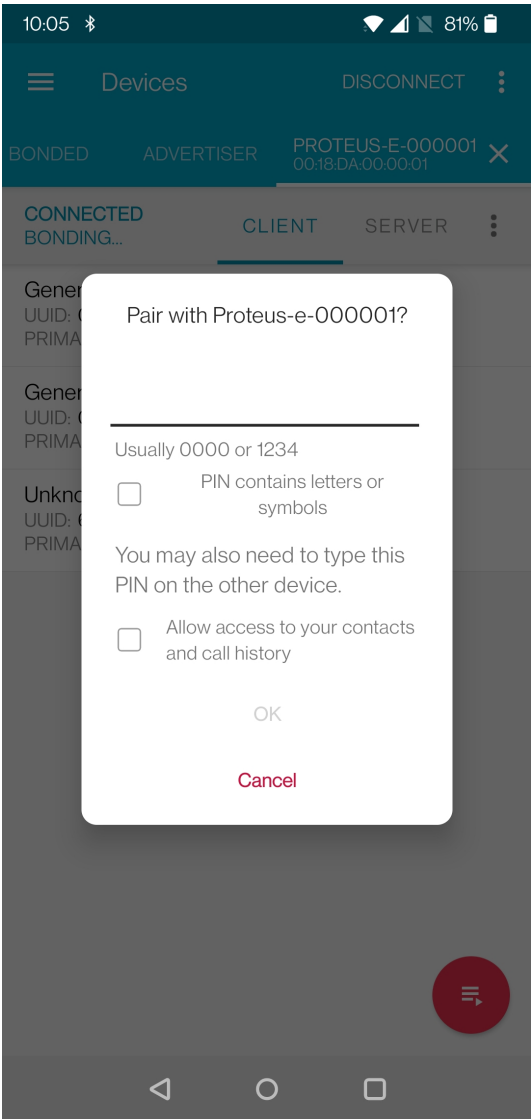
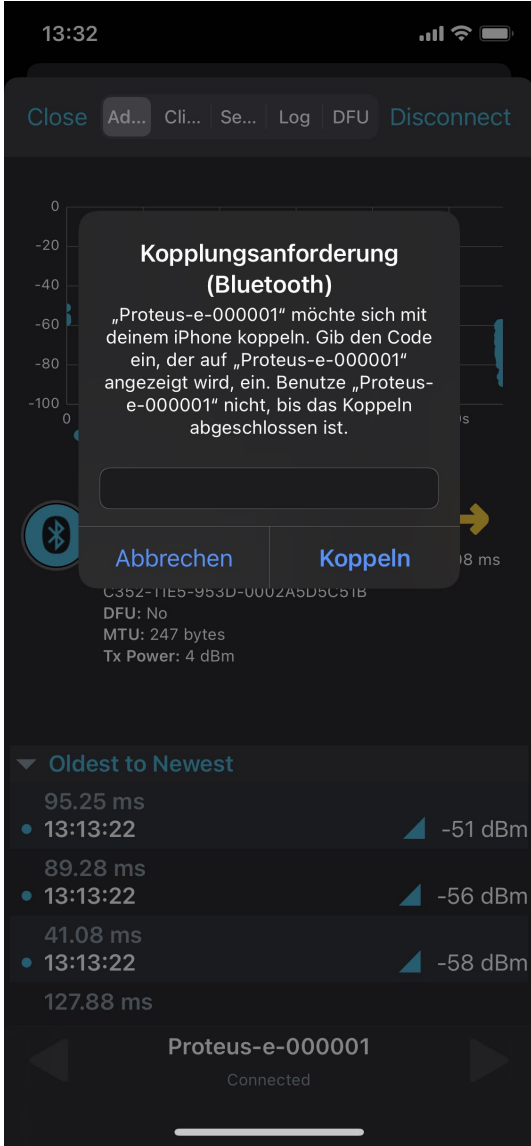
Android	iOS
<ul style="list-style-type: none"> Connect the Proteus EV-Board to a host. In this application note, we assume that a Windows PC and the PC tool WE UART Terminal [1] is used. For the Proteus EV-Boards this can be achieved by using a simple USB cable to connect it to a PC. Start the PC tool, select the right module and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing "Connect". 	
	
<ul style="list-style-type: none"> Press the reset button on the Proteus EV-Board. The Proteus module outputs a CMD_GETSTATE_CNF message to indicate that it is ready for operation. 	
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <pre style="margin: 0;">[10:22:08.296] CMD_GETSTATE_CNF: 02 41 0200 0101 41</pre> </div>	

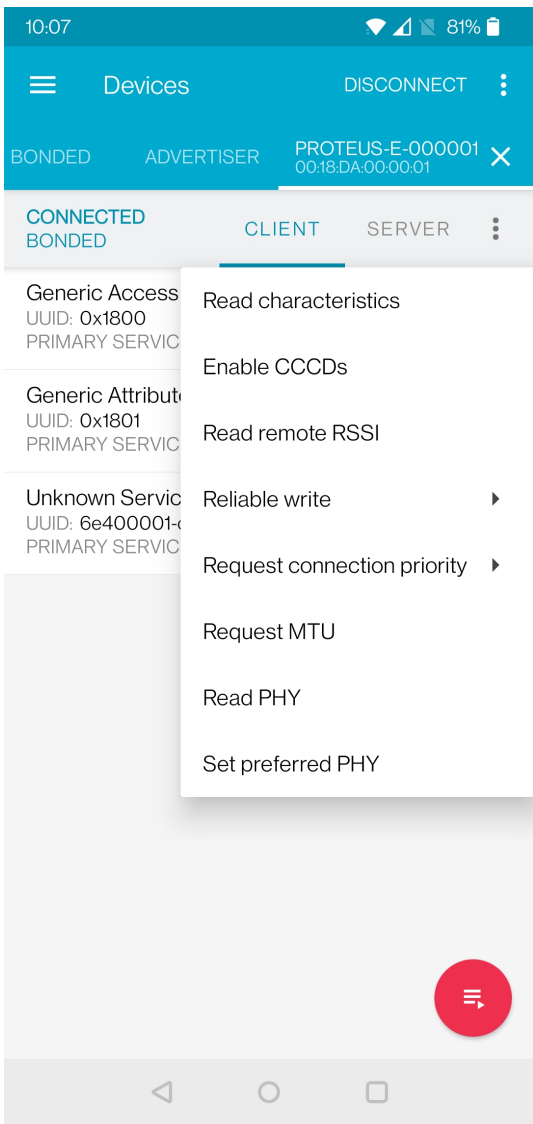
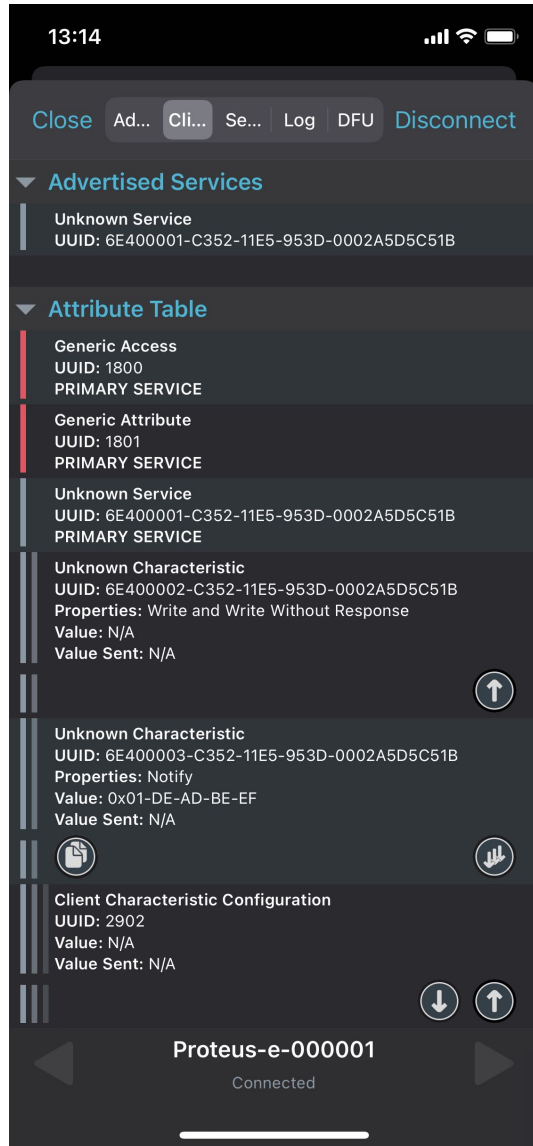
Android	iOS
---------	-----

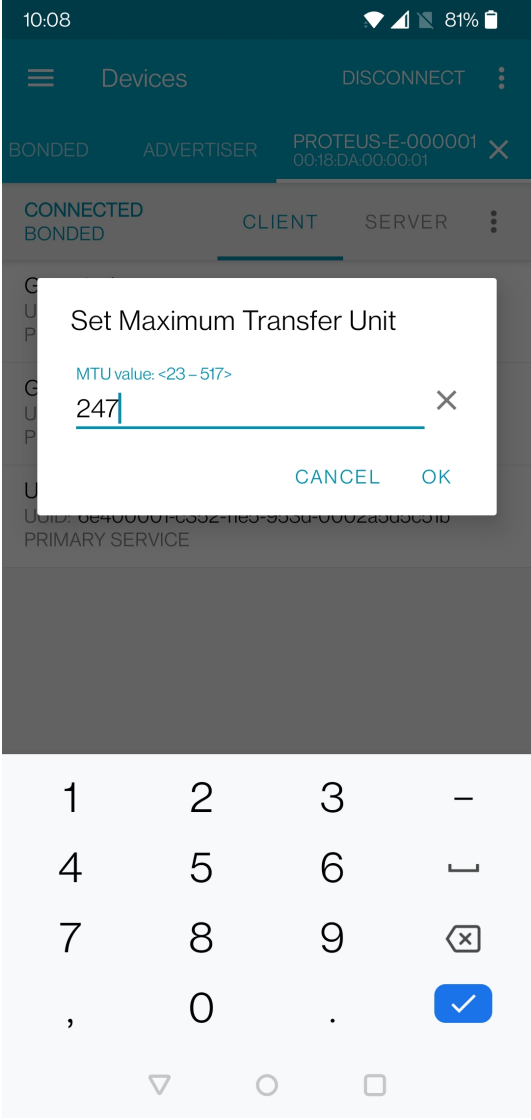
- Initially, the module is advertising. Thus, one LED of the Proteus EV-Board is blinking.
- Start your smart phone, enable the Bluetooth® LE feature and start the **nRF Connect** App.
- Press "SCAN" to find the module on the radio. In case several Proteus modules are found, the Bluetooth® MAC 0x0018DAxxxxxx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

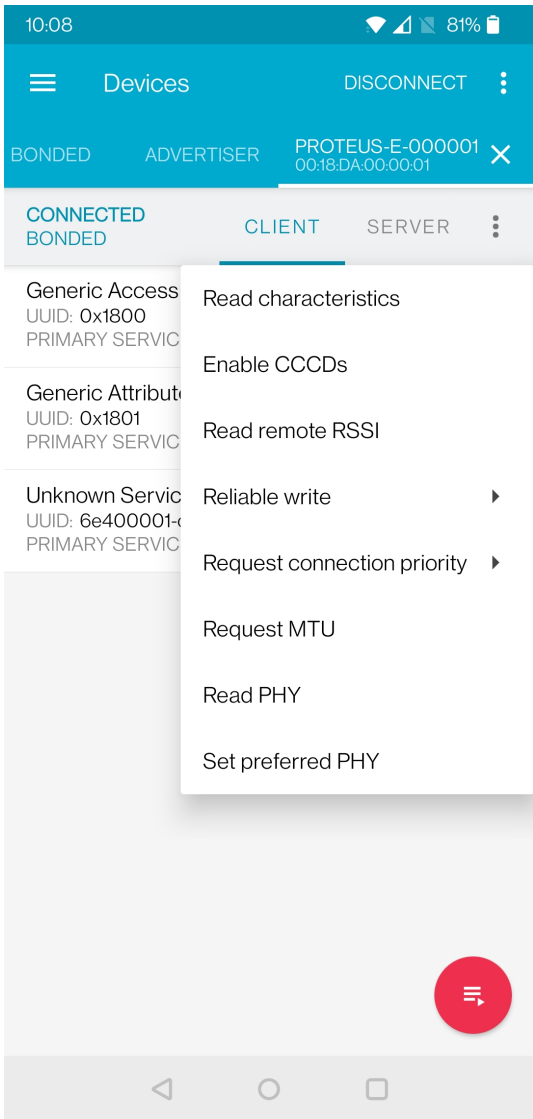
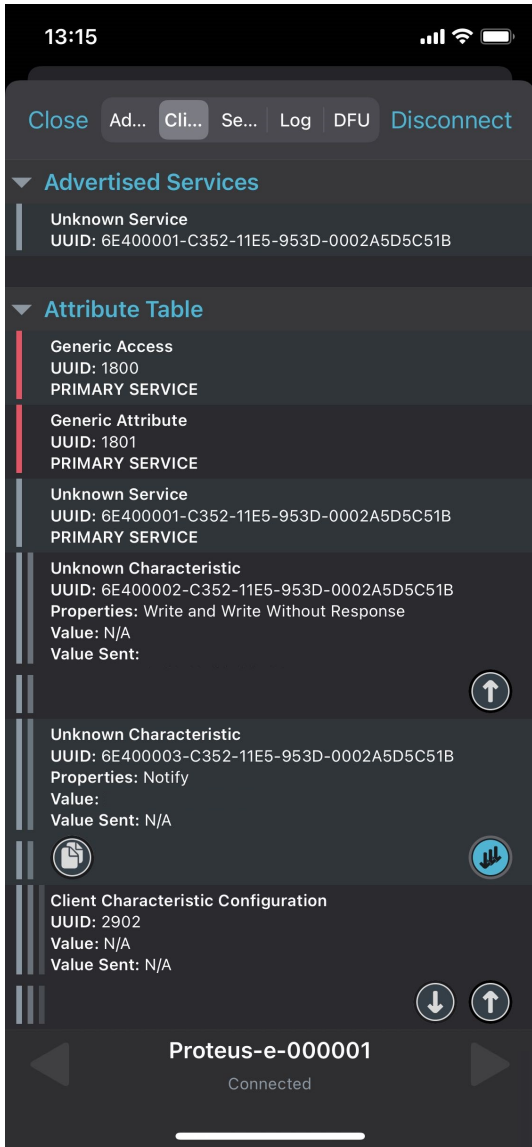


- When the module appears, press the "CONNECT" button.

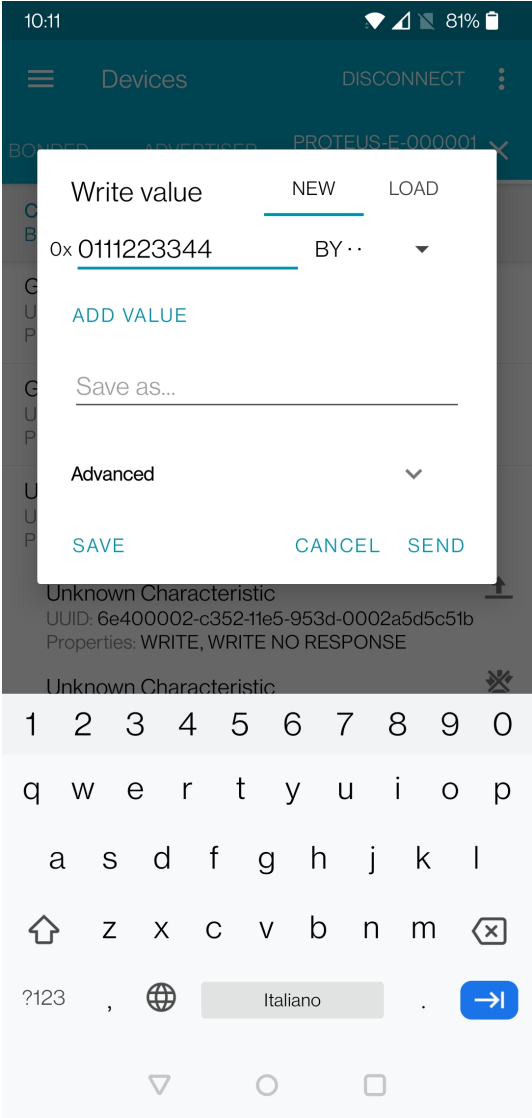
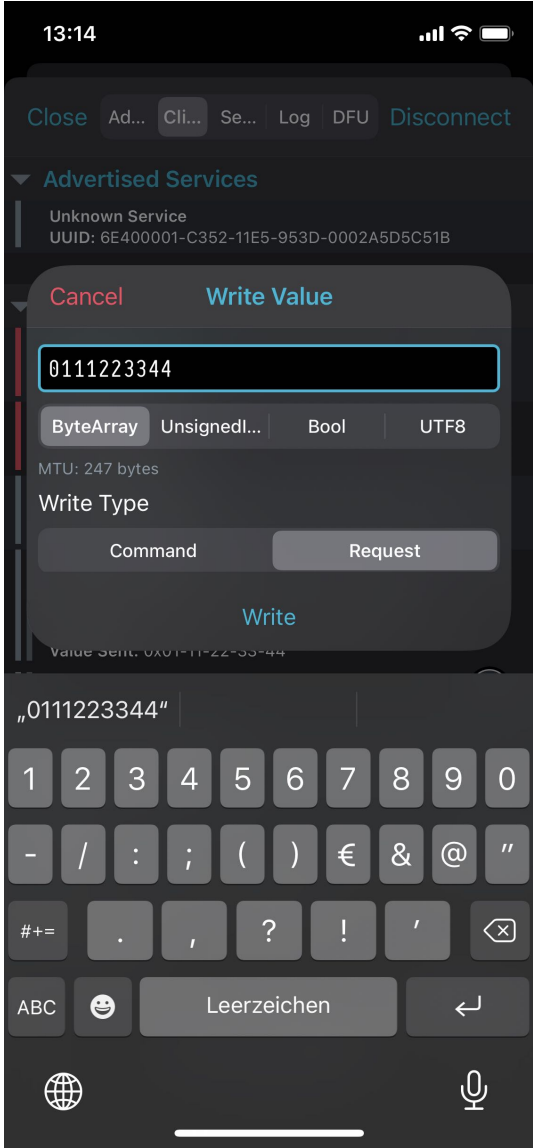
Android	iOS
<ul style="list-style-type: none"> As soon as the module has received the connection request from the smart phone the blinking LED will blink faster. Optional pairing: In case a security mode has been configured before, the smart phone requests the user for pairing actions. In case of the 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. 	
 <p>The screenshot shows an Android phone's Bluetooth settings. A dialog box titled "Pair with Proteus-e-000001?" is displayed. It includes a text input field with the hint "Usually 0000 or 1234", a checkbox for "PIN contains letters or symbols", and a checkbox for "Allow access to your contacts and call history". The dialog has "OK" and "Cancel" buttons.</p>	 <p>The screenshot shows an iPhone's Bluetooth settings. A dialog box titled "Kopplungsanforderung (Bluetooth)" is displayed. The text inside reads: "„Proteus-e-000001“ möchte sich mit deinem iPhone koppeln. Gib den Code ein, der auf „Proteus-e-000001“ angezeigt wird, ein. Benutze „Proteus-e-000001“ nicht, bis das Koppeln abgeschlossen ist." Below the text is a text input field and two buttons: "Abbrechen" and "Koppeln". The background shows the Bluetooth device list with "Proteus-e-000001" listed as "Connected".</p>

Android	iOS
<ul style="list-style-type: none"> Please click on the menu bullets on the right and press "Request MTU" to request for a larger MTU. 	<ul style="list-style-type: none"> Please click on the "Unknown Service" to start the service discovery and the MTU request. 

Android	iOS
<ul style="list-style-type: none">The Proteus module allows a MTU of up to 247 bytes, which results in a maximum payload size (MPS) of 243 bytes.  <p>1 2 3 - 4 5 6 _ 7 8 9 ⊗ , 0 . ✓</p>	<ul style="list-style-type: none">The iOS App runs this step simultaneously in the background, a user-defined MTU is not possible.

Android	iOS
<ul style="list-style-type: none"> Again click on the menu bullets on the right and press "Enable services"/"Enable CCCDs" to enable the notifications. 	<ul style="list-style-type: none"> 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. 
<ul style="list-style-type: none"> As soon as the module has received the notification enable request, the LED on the EV-Board is turned static on. Now you are fully connected and you can access the characteristics to transmit and receive data. 	

Android	iOS
<ul style="list-style-type: none">On the Proteus side, the radio module sent the corresponding CMD_CONNECT_IND and CMD_CHANNELOPEN_RSP in between. These messages indicate that a connection has been setup and a link has been opened. The CMD_CHANNELOPEN_RSP message contains the MPS (maximum payload size) of the current link. In this example it is 0xF3 (243_{dec}) bytes payload per packet. <div data-bbox="512 607 1046 943" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"><pre>[10:22:08.296] CMD_GETSTATE_CNF: 02 41 0200 0101 41 [10:23:05.019] CMD_CONNECT_IND: 02 86 0700 001EB4A8862D4C 66 [10:23:05.658] CMD_CHANNELOPEN_RSP: 02 C6 0800 001EB4A8862D4CF3 DA</pre></div>	

Android	iOS
<ul style="list-style-type: none"> 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					
<ul style="list-style-type: none"> The payload that has been sent is output by the Proteus module via UART. In the terminal program a CMD_DATA_IND message has been received, that contains the BTMAC of the sending device and the transmitted payload 0x11 0x22 0x33 0x44. The format of the CMD_DATA_IND message is as follows: 						
Start signal	Command	Length	BTMAC	RSSI	Payload	CS
0x02	0x84	2 Bytes	6 Bytes	1 Byte	(Length - 7) Bytes	1 Byte
0x02	0x84	0x0B 0x00	0x1E 0xB4 0xA8 0x86 0x2D 0x4C	0XC5	0x11 0x22 0x33 0x44	E9

```

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

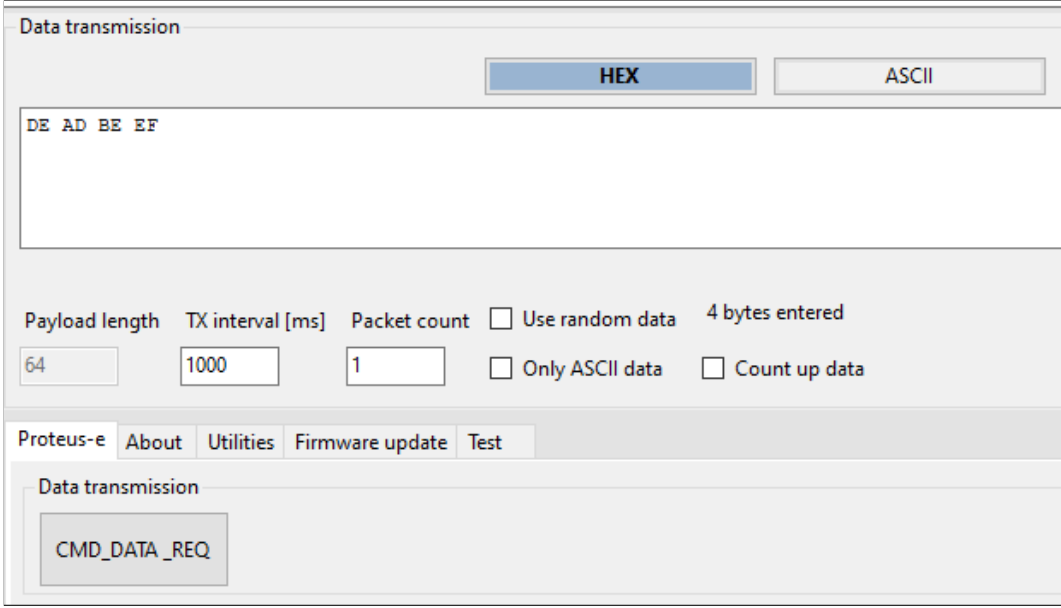
[10:23:05.019]
  CMD_CONNECT_IND:
02 86 0700 001EB4A8862D4C 66

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

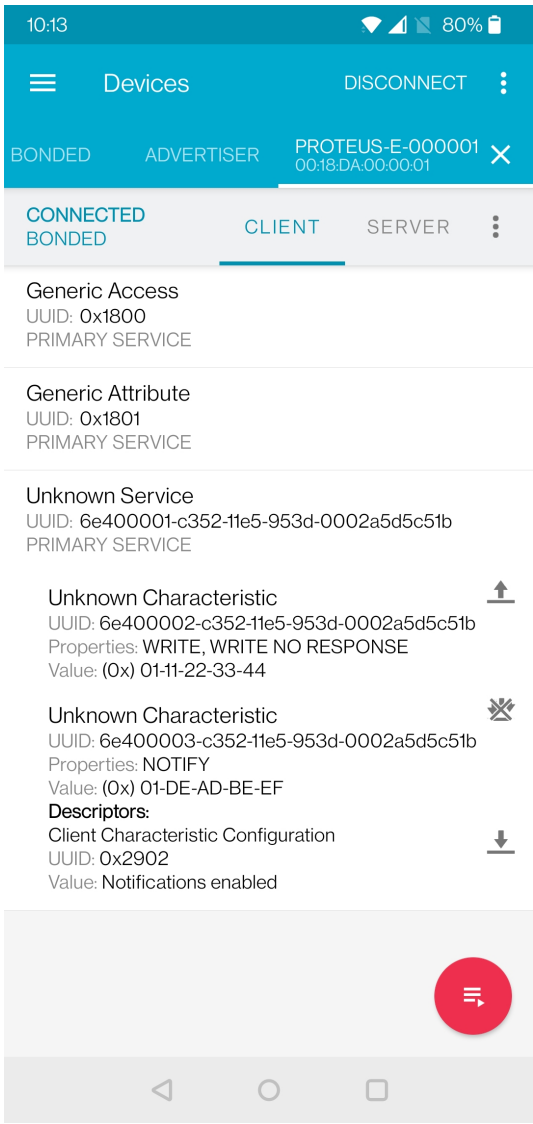
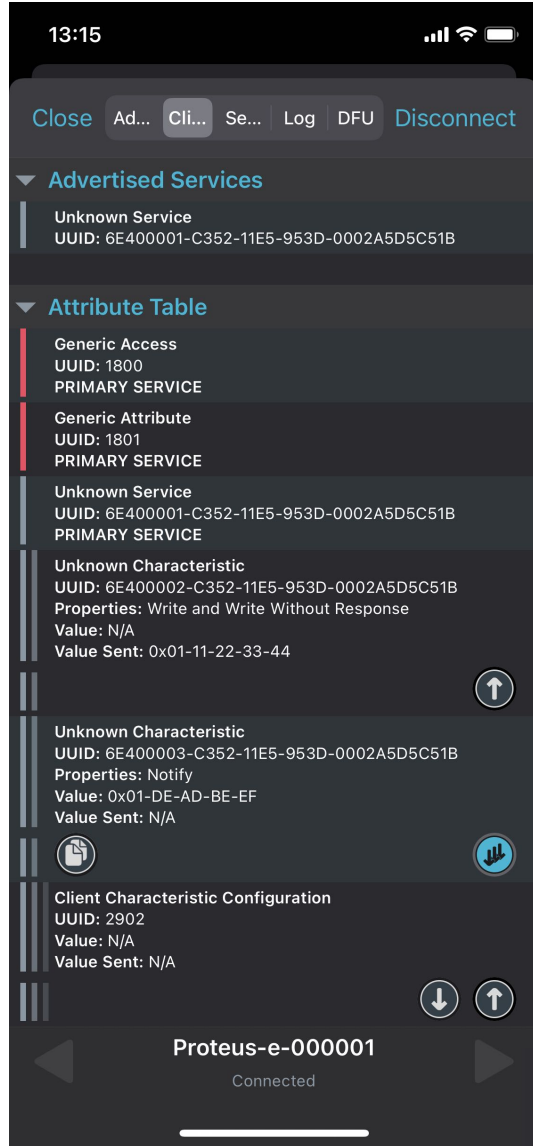
[10:23:20.067]
  CMD_DATA_IND:
02 84 0B00 1EB4A8862D4CCC511223344 E9
            
```

Android		iOS		
<ul style="list-style-type: none">To send back data to the smart phone simply insert your payload (here we choose 0xDE 0xAD 0xBE 0xEF) in a CMD_DATA_REQ message. The format of the CMD_DATA_REQ message is as follows, where the check sum (CS) is calculated as XOR of the preceding bytes:				
Start signal	Command	Length	Payload	CS
0x02	0x04	2 Bytes	Length Bytes	1 Byte
0x02	0x04	0x04 0x00	0xDE 0xAD 0xBE 0xEF	0x20

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



The screenshot shows the WE UART Terminal interface. At the top, there are two tabs: 'HEX' (selected) and 'ASCII'. Below the tabs is a text input field containing the hexadecimal string 'DE AD BE EF'. Underneath the input field are several configuration options: 'Payload length' (set to 64), 'TX interval [ms]' (set to 1000), 'Packet count' (set to 1), and three checkboxes: 'Use random data' (unchecked), 'Only ASCII data' (unchecked), and 'Count up data' (unchecked). At the bottom of the interface, there is a menu bar with 'Proteus-e', 'About', 'Utilities', 'Firmware update', and 'Test'. Below the menu bar is another 'Data transmission' section containing a button labeled 'CMD_DATA_REQ'.

Android	iOS
<ul style="list-style-type: none"> 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
<ul style="list-style-type: none">When sending the <code>CMD_DATA_REQ</code> to the Proteus module, it responds with two different messages. First, a <code>CMD_DATA_CNF</code> message is returned, as soon as the request was interpreted. Then a <code>CMD_TXCOMPLETE_RSP</code> message is returned as soon as the data has been transmitted. <pre data-bbox="213 499 1347 757">[10:24:29.005] CMD_DATA_REQ: 02 04 0400 DEADBEEF 20 [10:24:29.018] CMD_DATA_CNF: 02 44 0100 00 47 [10:24:29.110] CMD_TXCOMPLETE_RSP: 02 C4 0100 00 C7</pre>	

Android	iOS
<ul style="list-style-type: none">To disconnect the smart phone from the Proteus module, press the "DISCONNECT" button in the nRF Connect App. The Proteus module will output a <code>CMD_DISCONNECT_IND</code> message to indicate that the connection has been closed. <pre data-bbox="512 1196 1045 1301">[10:24:35.267] CMD_DISCONNECT_IND: 02 87 0100 13 97</pre> <ul style="list-style-type: none">After disconnection, the Proteus module starts advertising again, such that a reconnection can be performed.	

5.2 Smart phone using WE Bluetooth LE Terminal app as central device

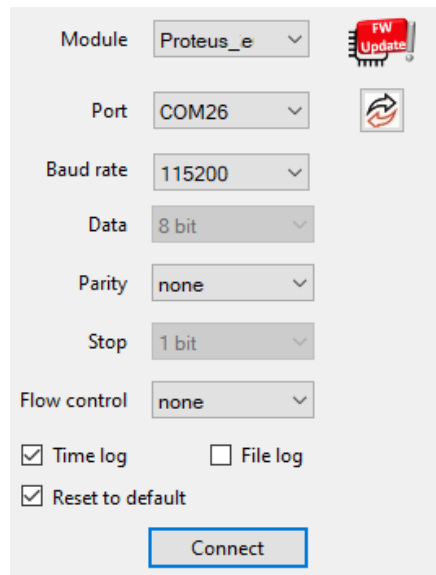
This chapter describes how to setup a connection to the Proteus-e in command mode, when a smart phone and the WE Bluetooth LE Terminal App are used.



The WE Bluetooth LE Terminal App for iOS and Android is provided by Würth Elektronik eiSos as executable [2, 3] as well as source code [7].

Please perform the following steps:

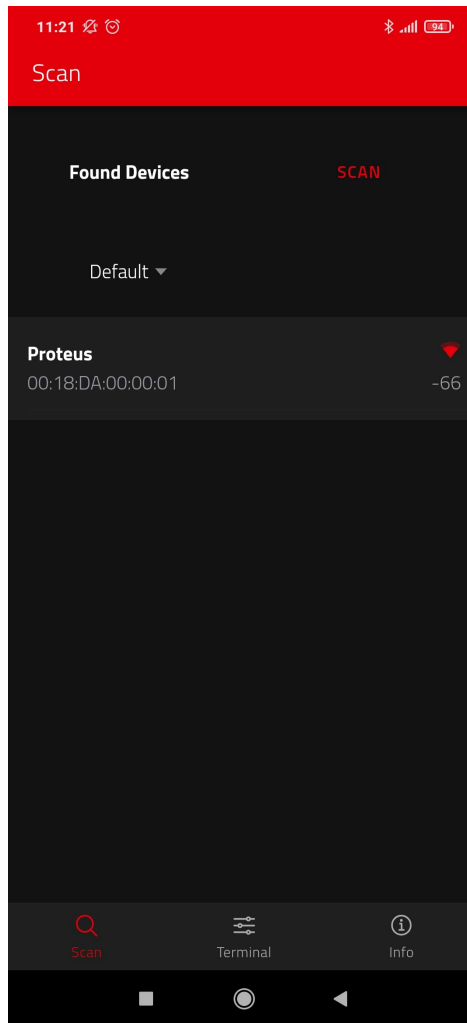
1. Connect the Proteus EV-Board to a host.
In this application note, we assume that a Windows PC and the PC tool WE UART Terminal [1] is used. For Proteus-e EV-Board this can be simply achieved by using a simple USB cable to connect it to a PC.
2. Start the WE UART Terminal, select the right module type and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing the "Connect" button.



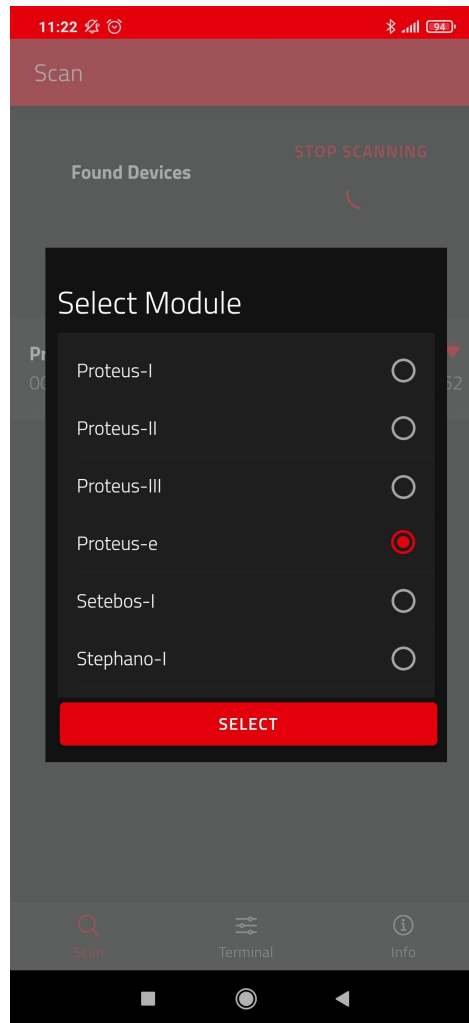
3. Press the reset button on the Proteus EV-Board. The Proteus module outputs a `CMD_GETSTATE_CNF` message to indicate that it is ready for operation.

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

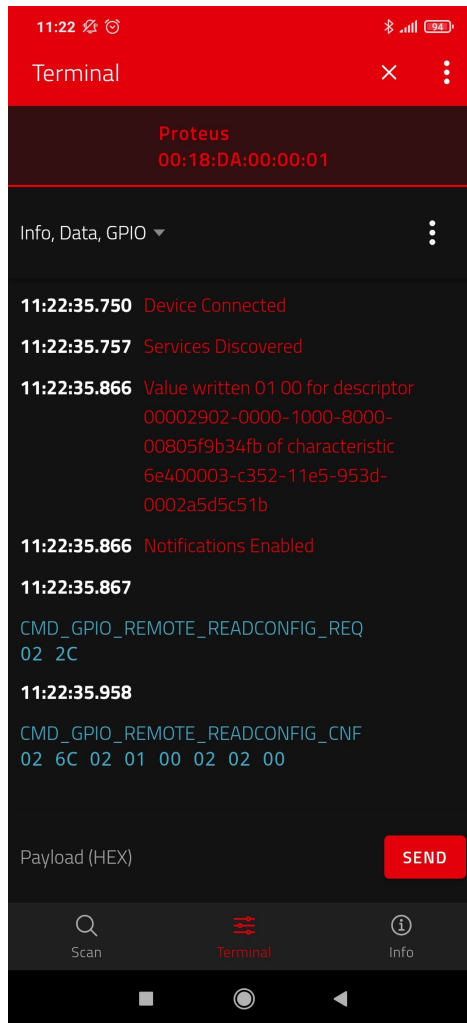
4. By default, the module is advertising. Thus, one LED of the Proteus EV-Board is blinking. Start your smart phone, enable the Bluetooth® LE and location feature and open the **WE Bluetooth LE Terminal App**.
5. Press "Scan" to find the module on the radio.



6. When the module appears in the scan list, select it. A pop-up will come up, where you need to select the current module type.



7. Optional pairing: In case a security mode has been configured before, the smart phone requests the user for pairing actions. In case of the static passkey authentication, the Proteus requests to enter the static passkey. The default passkey is "123123". The Bluetooth® coupling requirement pop-up is shown on your smart phone. If the bonding feature is enabled in the authentication settings and the bonding information already exists, a re-entering of the passkey is not required when reconnecting.
8. Now you are authenticated and the *LED_1* is turned static on. Now data can be transmitted in both directions.



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

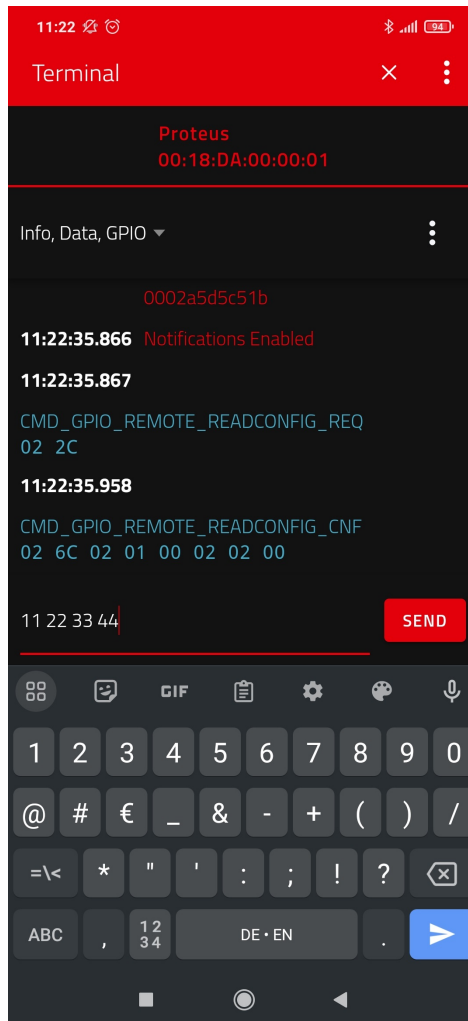
```
[10:22:08.296]
  CMD_GETSTATE_CNF:
02 41 0200 0101 41

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

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

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

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



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

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

```

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

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

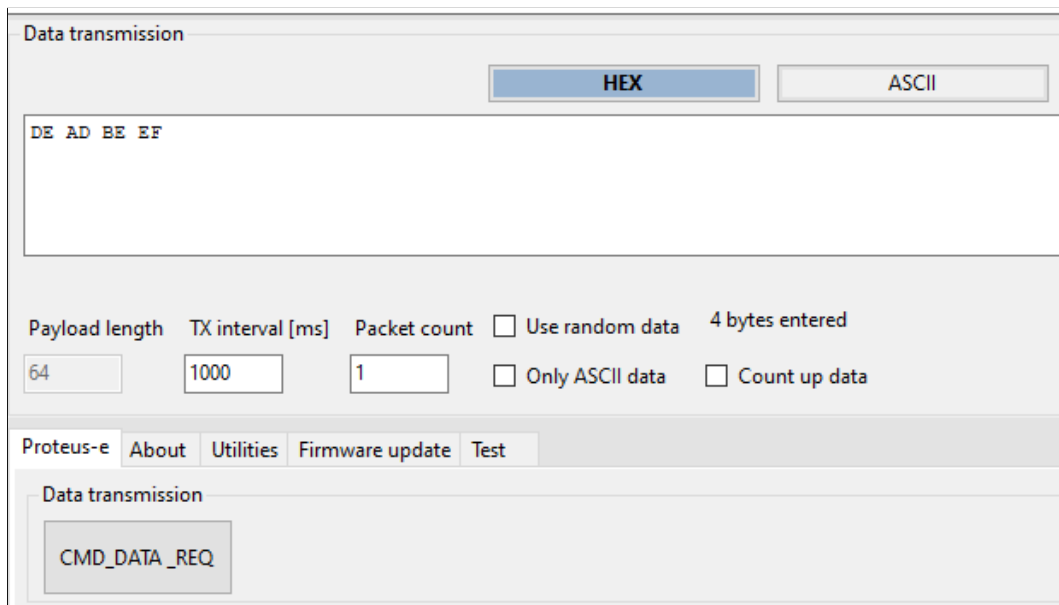
[10:23:05.658]
CMD_CHANNELOPEN_RSP:
02 C6 0800 001EB4A8862D4CF3 DA

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

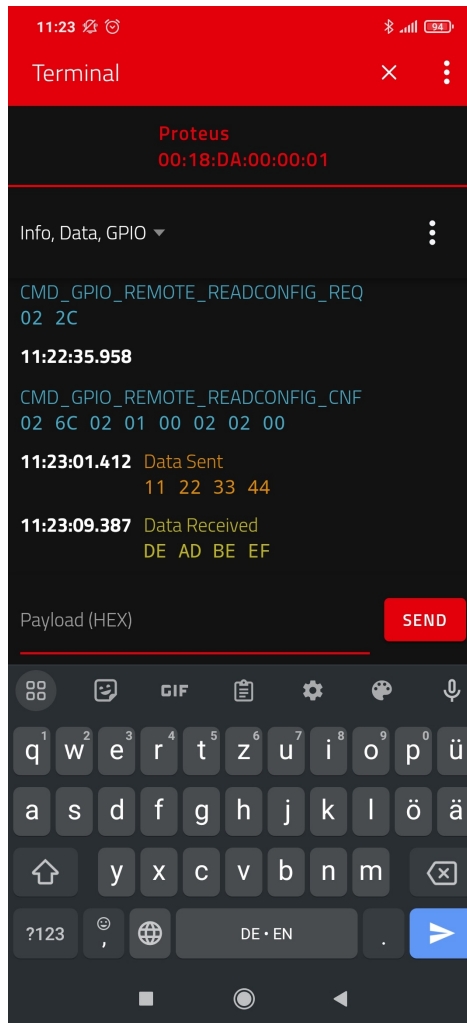
12. To send back data (here we choose 0xDE 0xAD 0xBE 0xEF) to the smart phone a CMD_DATA_REQ message must be sent to the module from the host. The format of the CMD_DATA_REQ message is as follows, where the check sum (CS) is calculated as XOR of the preceding bytes:

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

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



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



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

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

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

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

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

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

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	

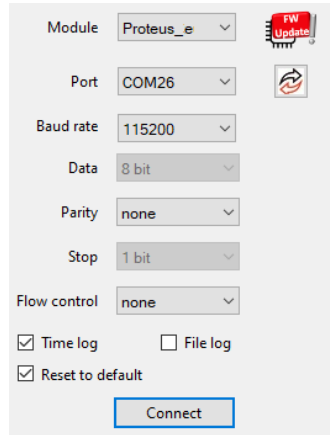
4. Now **Proteus_central** closes the connection.

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

5.3.1 Using WE UART Terminal to run the workflow

The above work flow can be easily applied using the WE UART Terminal [1] PC tool.

1. First open two instances of the WE UART Terminal.
2. On each instance, select the right module type (Proteus-e on one instance, Proteus-I,-II or -III on the central instance) and open a COM port using the Proteus default UART settings (115200 Baud, 8n1) by pressing the "Connect" button.



3. Then run the above workflow by clicking on the respective buttons in WE UART Terminal:

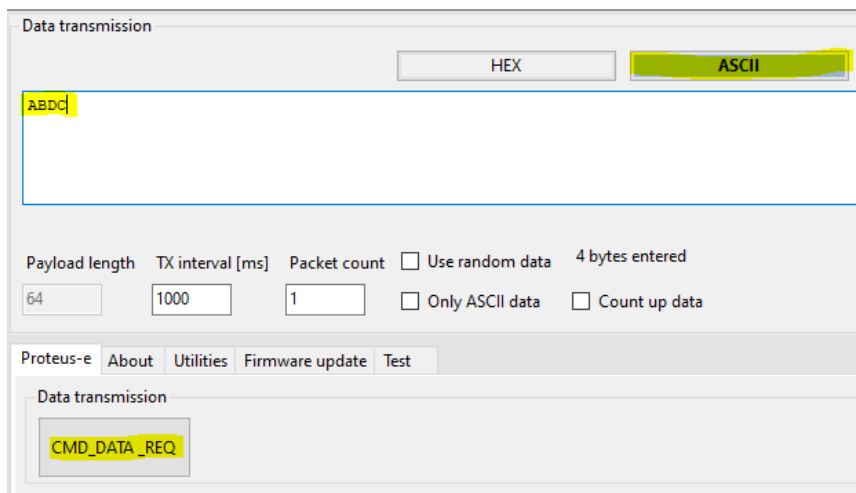


Figure 4: Important fields and buttons of Proteus-e instance of WE UART Terminal

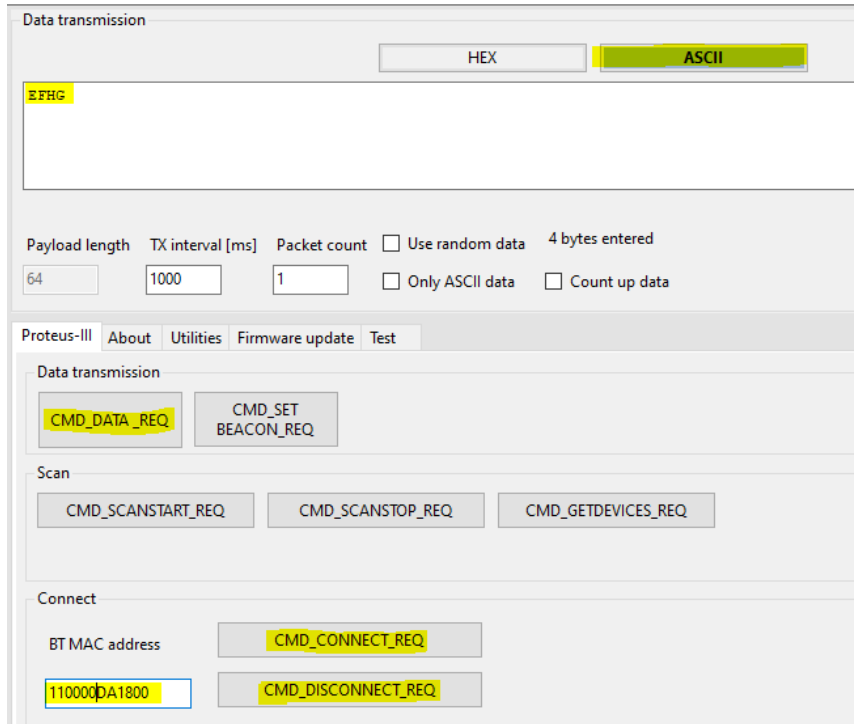


Figure 5: Important fields and buttons of central device instance of WE UART Terminal

6 References

- [1] Würth Elektronik. WE UART Terminal PC tool (Smart Commander). <https://www.we-online.de/wcs-software>.
- [2] Würth Elektronik. WE Bluetooth LE Terminal app for Android. <https://play.google.com/store/apps/details?id=com.eisos.android.terminal>.
- [3] Würth Elektronik. WE Bluetooth LE Terminal app for iOS. <https://apps.apple.com/de/app/proteus-connect/id1533941485>.
- [4] Nordic Semiconductor. nRF Connect app for Android. <https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp>.
- [5] Nordic Semiconductor. nRF Connect app for iOS. <https://apps.apple.com/us/app/nrf-connect-for-mobile/id1054362403>.
- [6] Würth Elektronik. Application note 24 - Proteus-e advanced developer guide. <http://www.we-online.com/ANR024>.
- [7] Würth Elektronik. Source code of WE Bluetooth LE Terminal app (cross platform). <https://github.com/WurthElektronik/Proteus-Connect>.

7 Important notes

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

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

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

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

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

List of Figures

1	Steps for the connection setup	7
2	Important fields and buttons of Transparent (Proteus-e) instance of WE UART Terminal	25
3	Important fields and buttons of central device instance of WE UART Terminal	25
4	Important fields and buttons of Proteus-e instance of WE UART Terminal	47
5	Important fields and buttons of central device instance of WE UART Terminal	48

List of Tables



Contact

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

Max-Eyth-Straße 1
74638 Waldenburg
Germany

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

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT