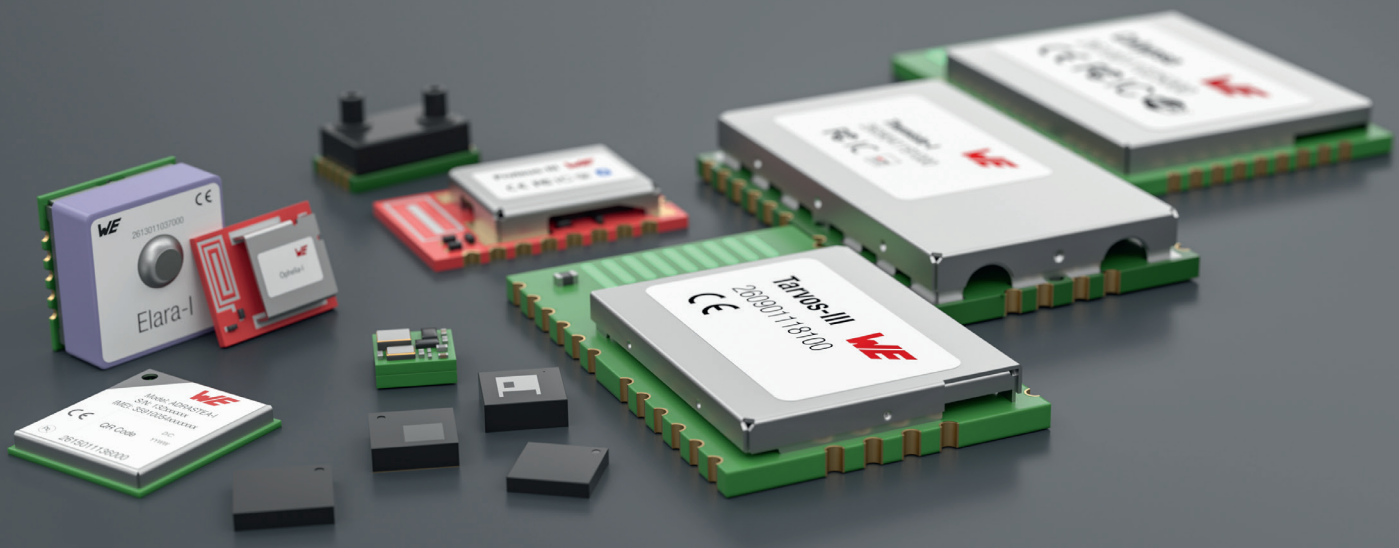


ANR020

Proteus-III/Proteus-e Remote GPIO Control Feature

Version 1.3

September 9, 2024

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

Revision history

Manual version	Notes	Date
1.0	• Initial version	July 2020
1.1	• Updated `Important notes`, meta data and document style • Added info on remote GPIO function of Proteus-e • Added chapter `References`	July 2023
1.2	• Updated images of most recent Proteus Connect app.	October 2023
1.3	• Updated name of Smart Commander PC tool and Proteus Connect app	September 2024

Abbreviations

Abbreviation	Name	Description
CS	Checksum	Byte wise XOR combination of the preceding fields.
I/O	Input/output	Pinout description.
Payload		The intended message in a frame / package.
RF	Radio frequency	Describes wireless transmission.
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-III/Proteus-e is a Bluetooth® LE module based on the nRF52 Nordic Semiconductors SoC which presents various Bluetooth® LE and low power features. It provides a command based UART interface that allows the configuration and control of the Proteus-III/Proteus-e by simple commands. Besides the commands needed for configuration and radio data transmission, various commands are provided to use the so called remote GPIO feature.
This feature allows to write and read up to 6 GPIOs of the Proteus-III/Proteus-e via Bluetooth® LE connection. With help of this, simple applications, like switches or digital level detectors, can be realized with the Proteus-III/Proteus-e without the need of connecting a host controller to it.

This application note describes which steps have to be run to switch a GPIO and how to read the pin level of a GPIO via remote connection. First or all a general description follows, then in the `Examples` section it is demonstrated how the GPIOs can be controlled by a second Proteus-III/Proteus-e radio module or by smart phone.

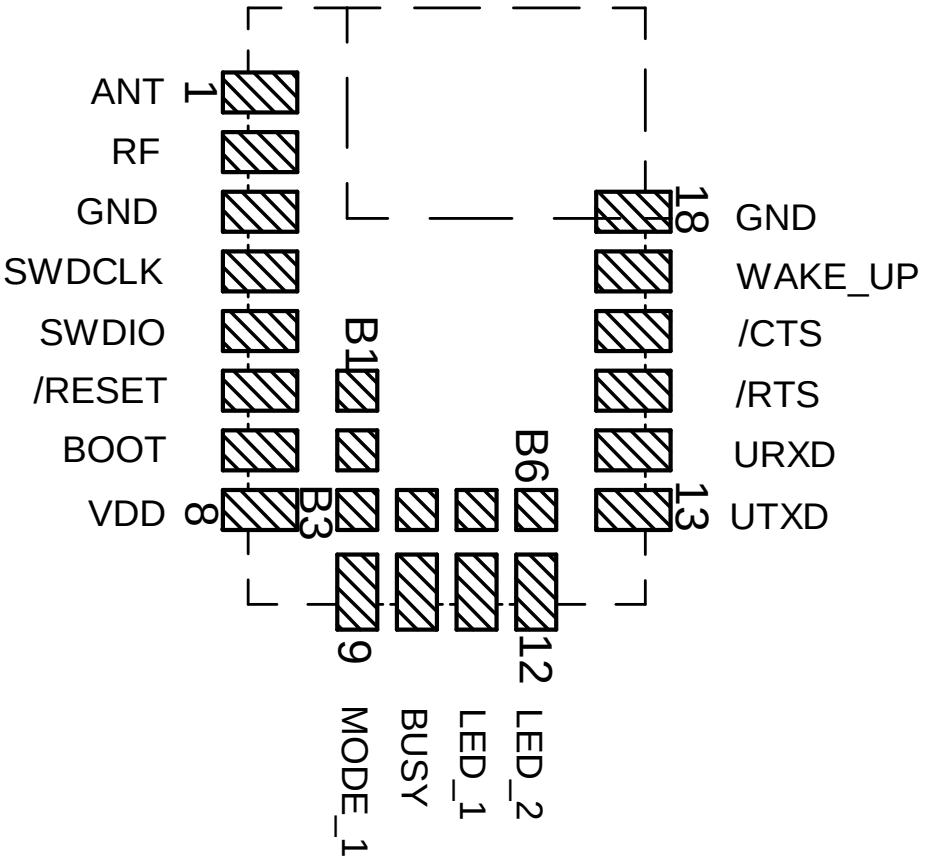
1.1 Supported GPIOs

The supported GPIOs are identified by so called GPIO_IDs, that are used in the commands for GPIO control. The following GPIOs of the Proteus-III/Proteus-e are supported for remote and local access.

1.1.1 Proteus-III

No	GPIO_ID	Supported functions
B1	1	Input, Output
B2	2	Input, Output
B3	3	Input, Output, PWM
B4	4	Input, Output, PWM
B5	5	Input, Output, PWM
B6	6	Input, Output, PWM

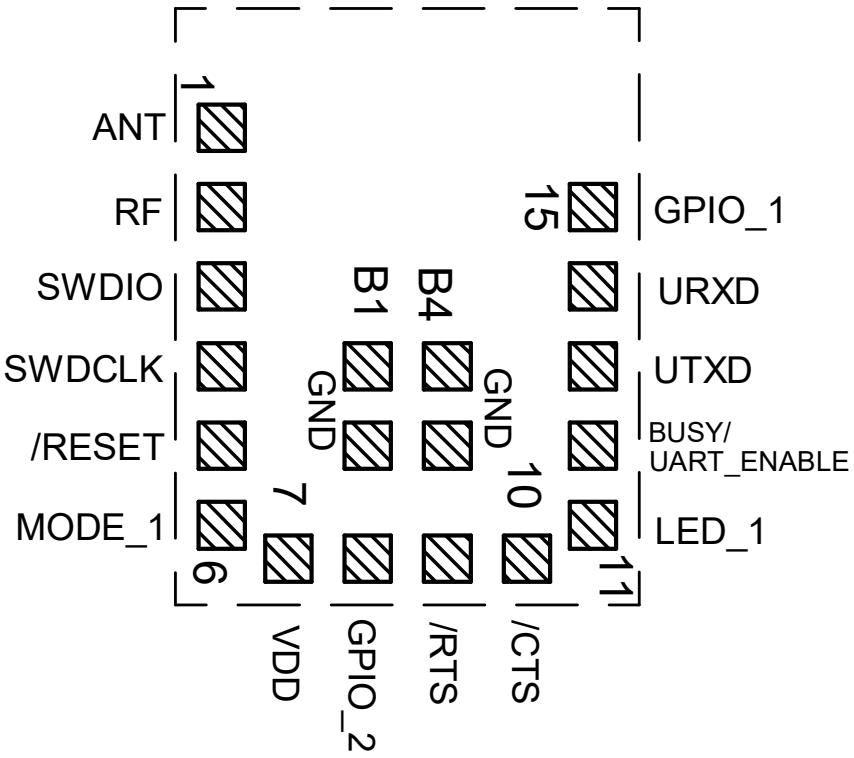
Table 1: Supported GPIO_IDs of Proteus-III



1.1.2 Proteus-e

No	GPIO_ID	Supported functions
GPIO_1	1	Input, Output
GPIO_2	2	Input, Output

Table 2: Supported GPIO_IDs of Proteus-e



2 General description

This chapter is based on a test setup with two radio modules. "Module 1" is the one, whose GPIOs are configured and controlled. It can be run without host controller. "Module 2" is the remote device which sends the commands for GPIO control via radio to "Module 1". It must be controlled via host controller "Host 2" to send the correct commands. Instead of the remote Proteus-III/Proteus-e "Module 2", another remote device such as a smart phone can also be used.

Chapter 3.1 contains an example, where the Proteus-III/Proteus-e "Module 1" is controlled by another Proteus-III[1] "Module 2". Chapter 3.2 shows the example, where the Proteus-III/Proteus-e "Module 1" is controlled by a smart phone.

The full description of the necessary UART commands can be found in the Proteus-III/Proteus-e manual [1, 2], where the radio commands are described in detail in the application note ANR009 [3] for Proteus-III and ANR024 [4] for Proteus-e.

2.1 Remote GPIO configuration

To use the remote GPIO control feature of the Proteus-III/Proteus-e "Module 1", the GPIOs of interest must be configured first. This configuration defines the GPIO function of the pin. Either an output pin, or an input pin with/without pull resistor or a PWM function, in case of Proteus-III.

To do so, the remote device "Module 2" must setup a Bluetooth® LE connection to the radio module "Module 1" and send a CMD_GPIO_REMOTE_WRITECONFIG_REQ command via Bluetooth® LE thereafter (see figure 1).
The configuration is stored in flash memory, such that it is retained even after a device restart. It can be read back by the remote device "Module 2" using the CMD_GPIO_REMOTE_READCONFIG_REQ command (see figure 2).

After the configuration has been done, the configured GPIOs are ready to be controlled.

[1] Proteus-e does not offer the central function. Thus only a Proteus-III can initiate the connection to another Proteus-III/Proteus-e to control it's GPIOs.

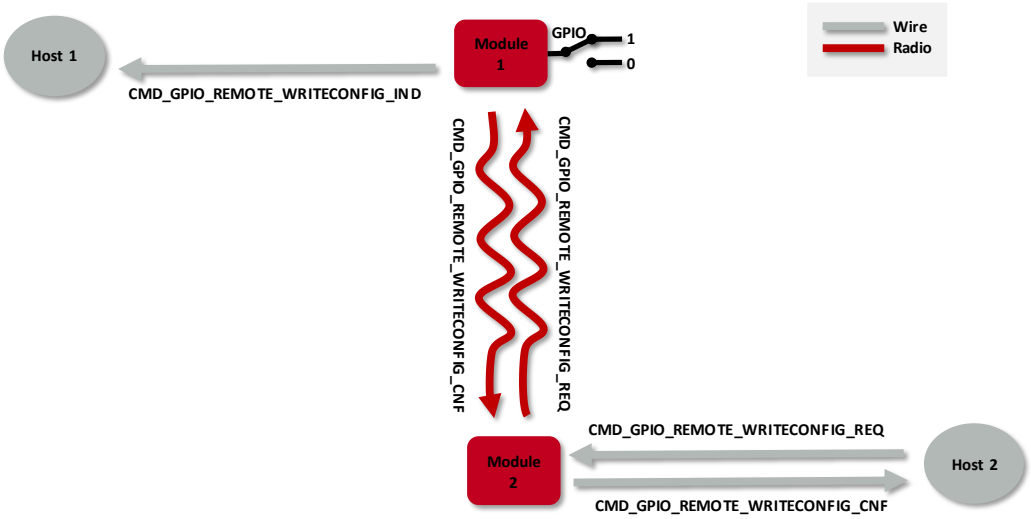


Figure 1: Configure the local GPIOs via remote device host

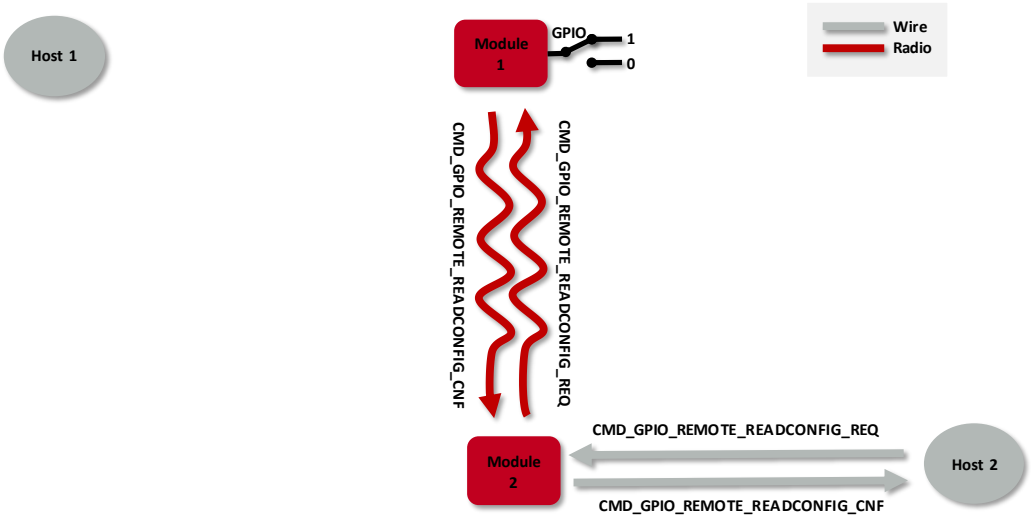


Figure 2: Read the configuration of the local GPIOs via remote device host

2.1.1 Local GPIO configuration

In case no host controller is connected to "Module 1", please go to the next chapter `Remote control`.

In case a host controller is connected to "Module 1", both, writing and reading the GPIO configuration can be done locally using the commands `CMD_GPIO_LOCAL_WRITECONFIG_REQ` and `CMD_GPIO_LOCAL_READCONFIG_REQ` (see figure 3 and figure 4). In this case, the host controller of "Module 1" must send the respective commands via UART to the "Module 1".

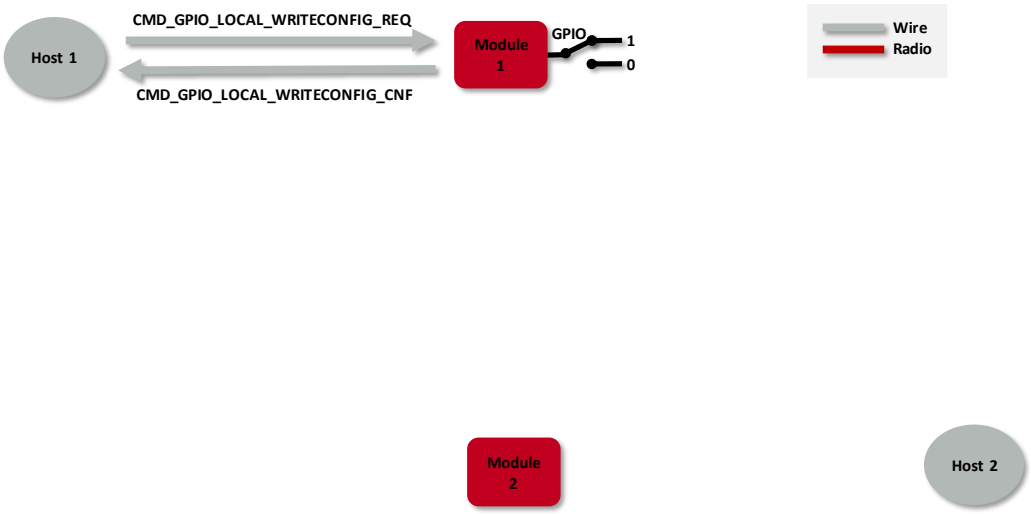


Figure 3: Configure the local GPIOs via local host

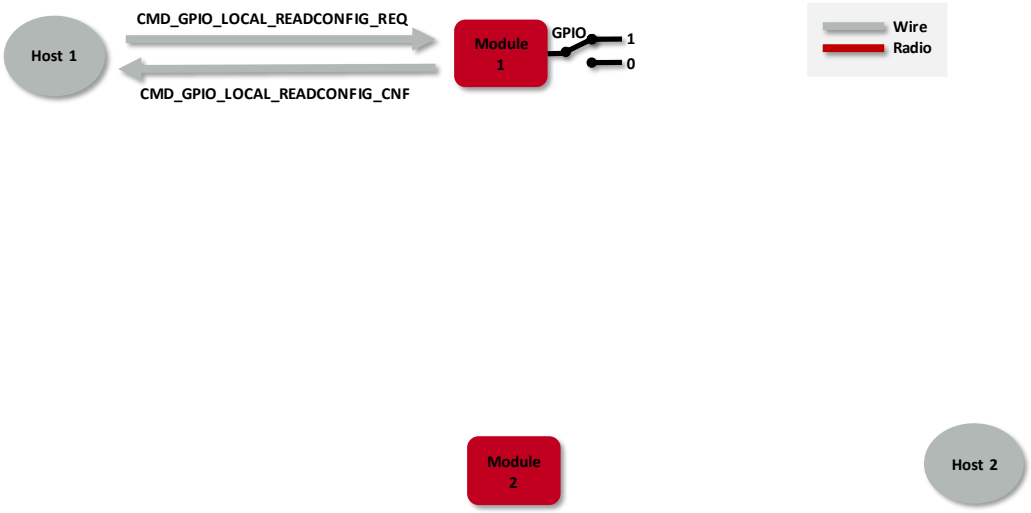


Figure 4: Read the configuration of the local GPIOs via local host

2.2 Remote control

To control a GPIO via remote device, first setup a Bluetooth® LE connection to the radio module "Module 1" and send the respective commands, `CMD_GPIO_REMOTE_WRITE_REQ` for setting GPIO output values (see figure 5), or `CMD_GPIO_REMOTE_READ_REQ` for reading GPIO values (see figure 6).
In case a host controller is connected to "Module 1", each time the GPIOs are written to via remote connection, the local host is informed using a `CMD_GPIO_REMOTE_WRITE_IND` message.

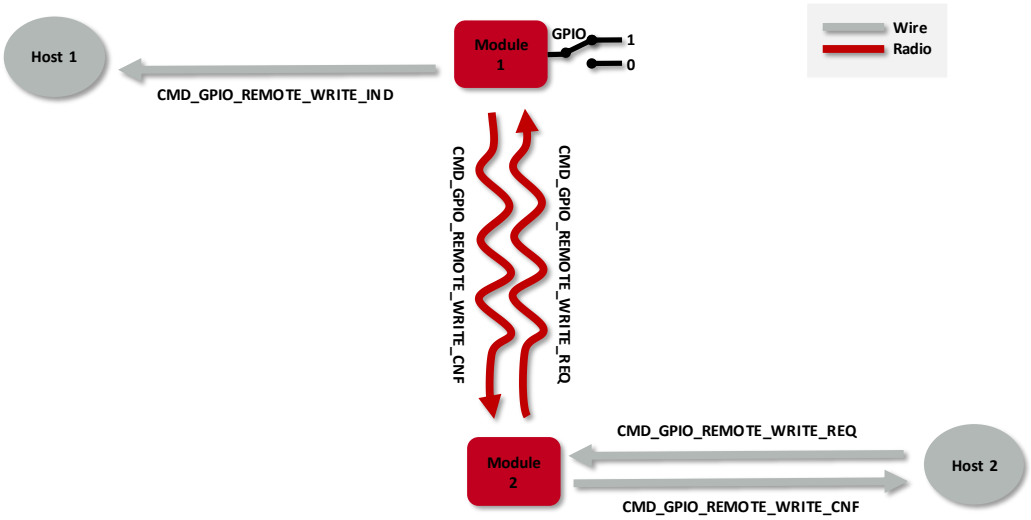


Figure 5: Set the output value of a GPIO via remote device

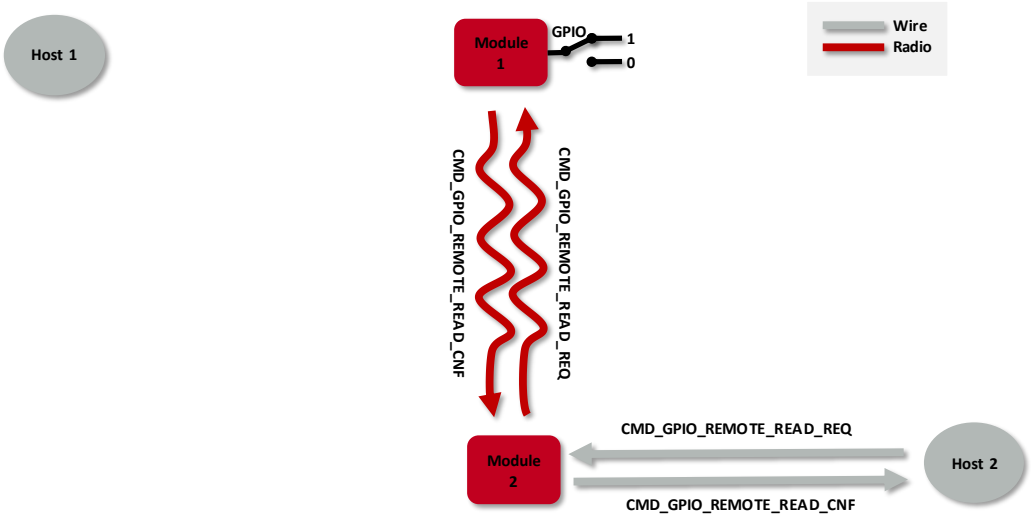


Figure 6: Read the input value of a GPIO via remote device

2.2.1 Local control

In case a host controller is connected to "Module 1", it also can write and read the GPIO status of the radio module "Module 1". To do so, the host controller must send the respective commands, `CMD_GPIO_LOCAL_WRITE_REQ` to set GPIO output values (see figure 7), or `CMD_GPIO_LOCAL_READ_R` to read GPIO values (see figure 8). Each time the GPIOs are written to via local host, the connected remote device is informed using a `CMD_GPIO_LOCAL_WRITE_IND` message.

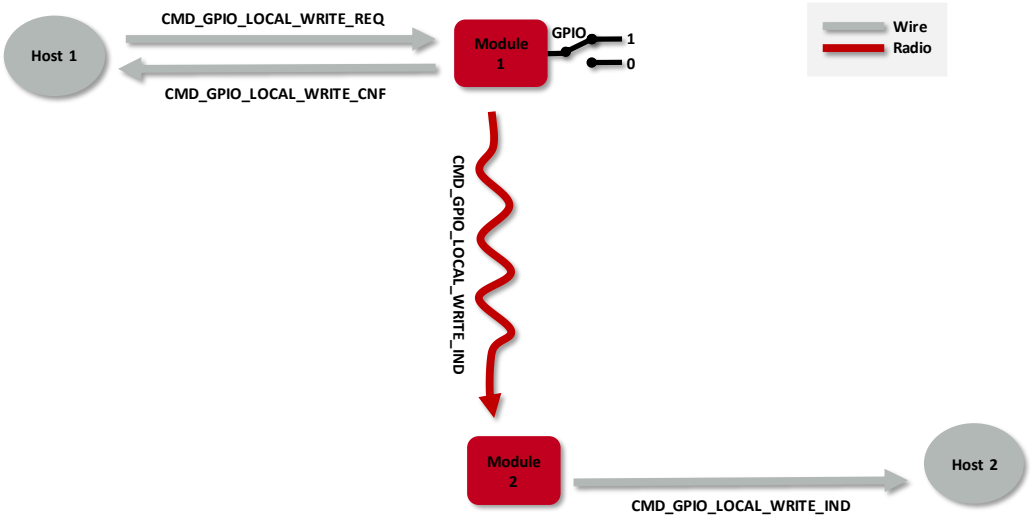


Figure 7: Set the output value of a GPIO via host controller

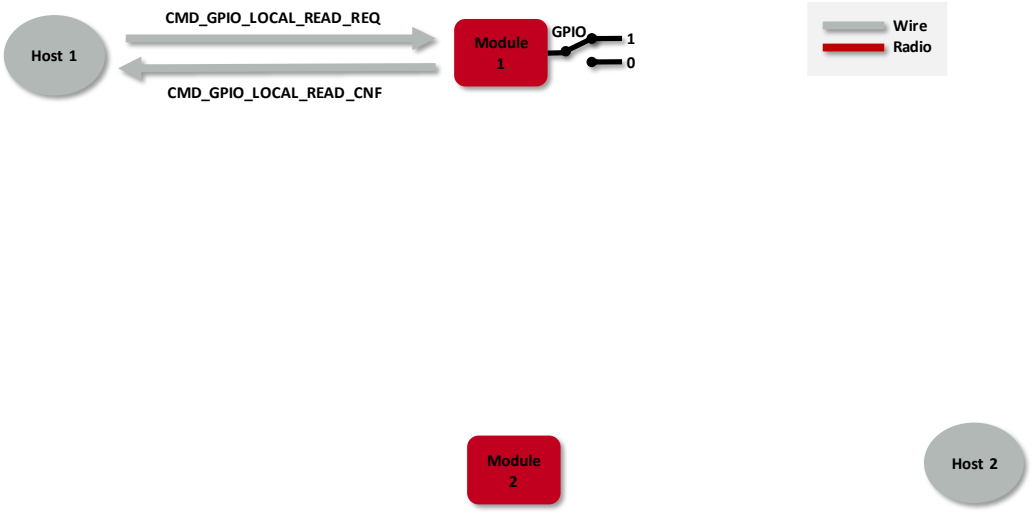


Figure 8: Read the input value of a GPIO via host controller

3 Examples

The below commands are in hexadecimal notation. The arrow in the left column describes, whether it's a message from host to radio module, or vice versa. A request command is always sent from host to module (⇒). An indication, confirmation or response message is always sent from module to host (⇐).

3.1 Example 1: Control "Module 1" by "Module 2"

This chapter describes how to setup a Bluetooth® LE connection between two radio modules. One of them must be a Proteus-III radio module. It is demonstrated how to configure and control the GPIOs of one of both modules via Bluetooth® LE connection.
The setup is as shown above, where "Module 1" is the one, whose GPIOs are switched via radio. It can run without host controller. "Module 2" is the one which must be connected to a host controller "Host 2", to send the corresponding configuration and control commands.

For demonstration purposes, two Proteus EV-Boards are appropriate. As "host" a Windows computer including the WE UART Terminal [5] PC tool can be taken. The USB connector of the EV-Board allows an easy connection to the Windows computer.

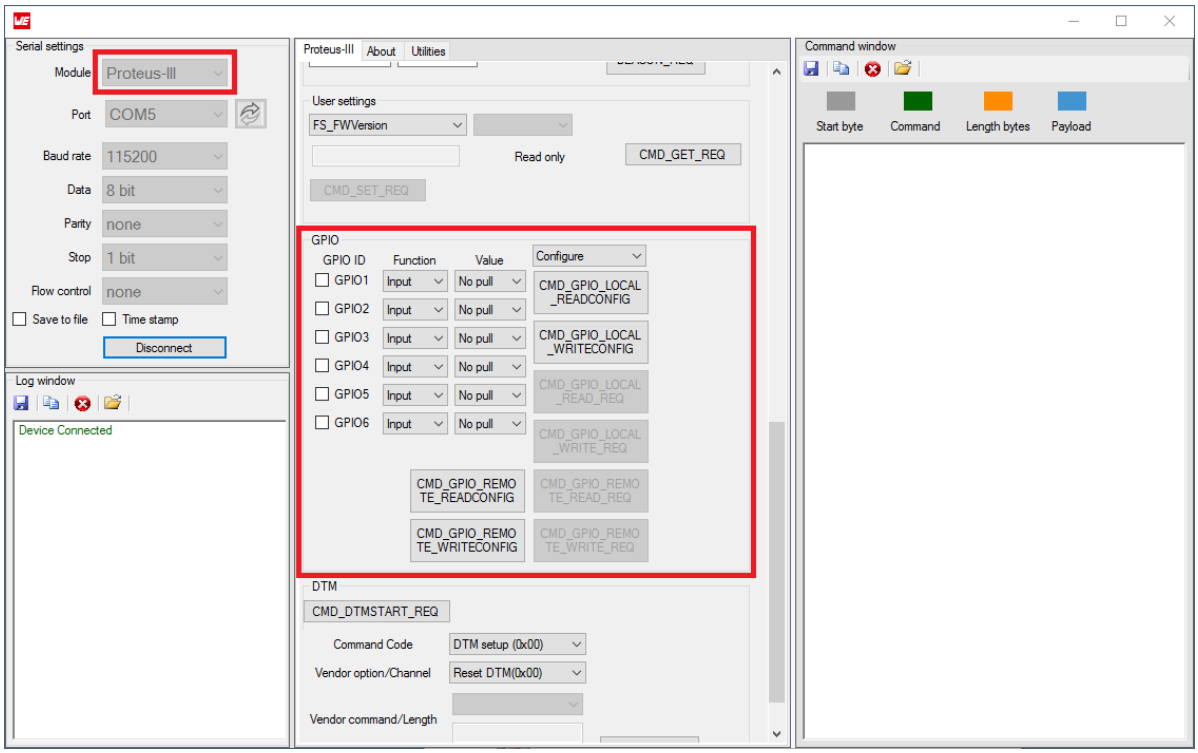


Figure 9: WE UART Terminal PC tool

The Würth Elektronik eiSos's tool WE UART Terminal includes the function to control the GPIOs of the Proteus-III/Proteus-e. This tool allows to generate the commands, that are shown below, by clicking the corresponding buttons in the WE UART Terminal GUI menu.

3.1.1 Boot-up and connection setup

1. Power-up the modules and make their UARTs accessible by the host(s) (115200 Baud, 8n1). After the power-up or after reset the following sequence is sent from the module to the host.

Info	Module 1	Module 2
⇐ Response `CMD_GETSTATE_CNF`: Module 1 started in `ACTION_IDLE` mode.	02 41 02 00 01 01 41	
⇐ Response `CMD_GETSTATE_CNF`: Module 2 started in `ACTION_IDLE` mode.		02 41 02 00 01 01 41

2. Request the MAC address `FS_BTMAC` of both modules.

Info	Module 1	Module 2
⇒ Request `CMD_GET_REQ` with settings index 4		02 10 01 00 04 17
⇐ Response `CMD_GET_CNF`: `FS_BTMAC` of Module 2 is 0x55 0x00 0x00 0xDA 0x18 0x00		02 50 07 00 00 55 00 00 DA 18 00 C2
⇒ Request `CMD_GET_REQ` with settings index 4	02 10 01 00 04 17	
⇐ Response `CMD_GET_CNF`: `FS_BTMAC` of Module 1 is 0x11 0x00 0x00 0xDA 0x18 0x00	02 50 07 00 00 11 00 00 DA 18 00 86	

3. Connect Module 2 to Module 1 via Bluetooth®.

Info	Module 1	Module 2
⇒ Request `CMD_CONNECT_REQ` with `FS_BTMAC` of Module 1		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 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	

⇐ Indication `CMD_CHANNELOPEN_RSP`: Channel opened successfully to 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 E3
⇐ 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	

4. Now the Bluetooth® LE connection is open, and the configuration and control of the GPIOs of "Module 1" can be done.

3.1.2 Configure and control a GPIO as output

If the Bluetooth® LE connection has been setup, as shown in chapter `3.1.1`, the following steps can be run to configure the GPIO *B1* with GPIO_ID 1 as output pin.

1. First of all configure the GPIO *B1* with GPIO_ID 1 as output pin with default level LOW.

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_WRITECONFIG_REQ`: Configure GPIO *B1* (GPIO_ID 1) as output with default level LOW		02 28 04 00 03 01 02 00 2E
⇐ Response `CMD_GPIO_REMOTE_WRITECONFIG_CNF`: GPIO with GPIO_ID 1 has been configured successfully		02 68 04 00 00 02 01 00 6D
⇐ Indication `CMD_GPIO_REMOTE_WRITECONFIG_IND`: GPIO with GPIO_ID 1 has been configured to output LOW by remote device	02 A8 04 00 03 01 02 00 AE	

2. Read the current configuration

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_READCONFIG_REQ`:		02 2C 00 00 2E
⇐ Response `CMD_GPIO_REMOTE_READCONFIG_CNF`: GPIO with GPIO_ID 1 has been configured to output LOW, GPIO_ID 2 - 6 are not configured		02 6C 14 00 00 03 01 02 00 02 02 00 02 03 00 02 04 00 02 05 00 02 06 00 7E

3. Switch the GPIO *B1* with GPIO_ID 1 to HIGH.

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_WRITE_REQ`: Set the GPIO *B1* (GPIO_ID 1) to HIGH		02 29 03 00 02 01 01 2A
⇐ Response `CMD_GPIO_REMOTE_WRITE_CNF`: GPIO with GPIO_ID 1 has been successfully		02 69 04 00 00 02 01 00 6C
⇐ Indication `CMD_GPIO_REMOTE_WRITE_IND`: GPIO with GPIO_ID 1 has been set to HIGH by remote device	02 A9 03 00 02 01 01 AA	

4. Read the current state of the GPIO *B1* with GPIO_ID 1.

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_READ_REQ`: Read the state of GPIO *B1* (GPIO_ID 1)		02 2A 02 00 01 01 2A
⇐ Response `CMD_GPIO_REMOTE_READ_CNF`: GPIO with GPIO_ID 1 is HIGH		02 6A 04 00 00 02 01 01 6E

5. Switch the GPIO *B1* with GPIO_ID 1 again to LOW.

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_WRITE_REQ`: Set the GPIO *B1* (GPIO_ID 1) to LOW		02 29 03 00 02 01 00 2B
⇐ Response `CMD_GPIO_REMOTE_WRITE_CNF`: GPIO with GPIO_ID 1 has been set successfully		02 69 04 00 00 02 01 00 6C
⇐ Indication `CMD_GPIO_REMOTE_WRITE_IND`: GPIO with GPIO_ID 1 has been set to LOW by the remote device	02 A9 03 00 02 01 00 AB	

3.1.3 Configure and control a GPIO as input

If the Bluetooth® LE connection has been setup, as shown in chapter `3.1.1`, the following steps can be run to configure the GPIO *B1* with GPIO_ID 1 as input pin.

1. First of all configure the GPIO *B1* with GPIO_ID 1 as input pin with default level LOW.

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_WRITECONFIG_REQ`: Configure GPIO *B1* (GPIO_ID 1) as input with pull down resistor		02 28 04 00 03 01 01 01 2C

⇐ Response `CMD_GPIO_REMOTE_WRITECONFIG_CNF`: GPIO with GPIO_ID 1 has been configured successfully		02 68 04 00 00 02 01 00 6D
⇐ Indication `CMD_GPIO_REMOTE_WRITECONFIG_IND`: GPIO with GPIO_ID 1 has been configured to input with pull down resistor by the remote device	02 A8 04 00 03 01 01 01 AE	

2. Read the current configuration

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_READCONFIG_REQ`:		02 2C 00 00 2E
⇐ Response `CMD_GPIO_REMOTE_READCONFIG_CNF`: GPIO with GPIO_ID 1 has been configured to input with pulldown, GPIO_ID 2 - 6 are not configured		02 6C 14 00 00 03 01 01 01 02 02 00 02 03 00 02 04 00 02 05 00 02 06 00 7C

3. Leave the GPIO *B1* open and read the current state of the GPIO *B1* with GPIO_ID 1.

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_READ_REQ`: Read the state of GPIO *B1* (GPIO_ID 1)		02 2A 02 00 01 01 2A
⇐ Response `CMD_GPIO_REMOTE_READ_CNF`: GPIO with GPIO_ID 1 is LOW		02 6A 04 00 00 02 01 00 6F

4. Now, apply a HIGH signal to the GPIO *B1* and read the current state of the GPIO *B1* with GPIO_ID 1.

Info	Module 1	Module 2
⇒ Request `CMD_GPIO_REMOTE_READ_REQ`: Read the state of GPIO *B1* (GPIO_ID 1)		02 2A 02 00 01 01 2A
⇐ Response `CMD_GPIO_REMOTE_READ_CNF`: GPIO with GPIO_ID 1 is HIGH		02 6A 04 00 00 02 01 01 6E

3.2 Example 2: Control "Module 1" by smart phone

This chapter describes how to setup a Bluetooth® LE connection between a smart phone and a Proteus-III/Proteus-e radio module, and how to configure and control the GPIOs of the Proteus-III/Proteus-e "Module 1" via the Bluetooth® LE connection.
"Module 1" itself can run without host controller as its GPIOs are controlled via radio. For demonstration purposes, we here again use a Proteus-III/Proteus-e EV-Board, which is connected to a Windows computer including the WE UART Terminal [5] PC tool, which is the host "Host 1".

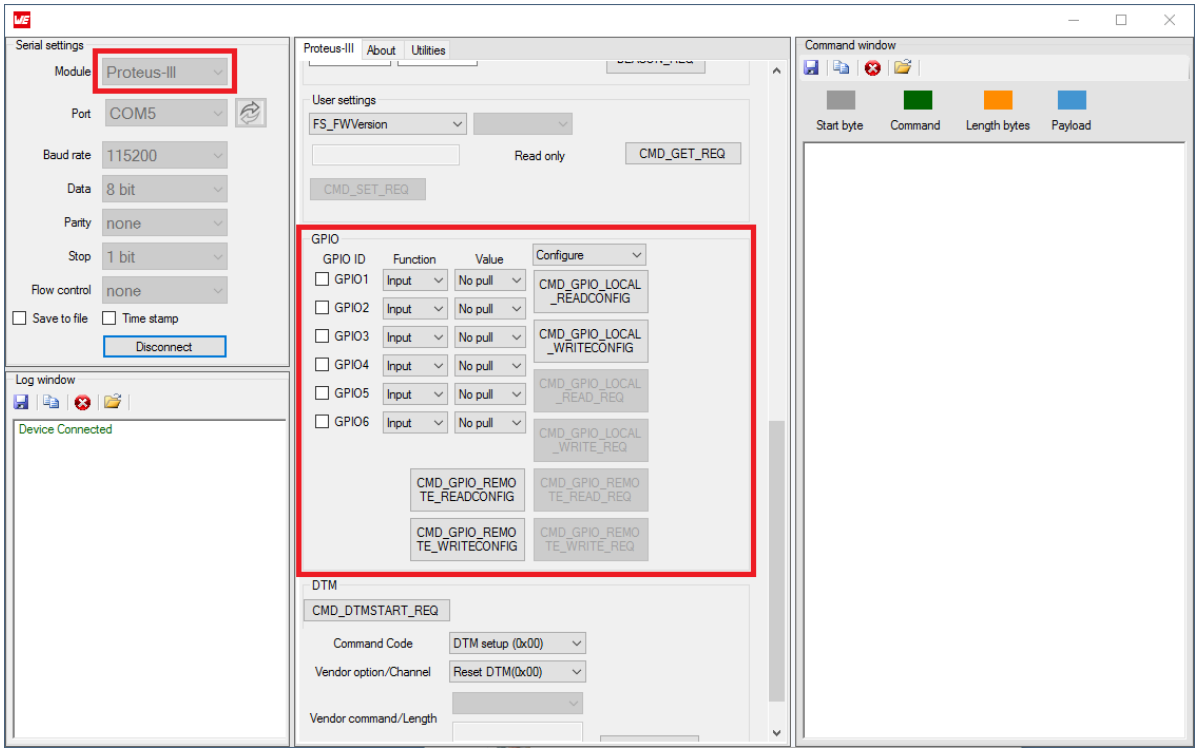


Figure 10: WE UART Terminal PC tool

Instead of "Module 2" we here use an Android smart phone including the app "WE Bluetooth LE Terminal", which allows the operation with all radio modules from the Proteus series. Besides the functions for connection setup and data transmission, this app contains the functions of GPIO configuration and control, which we will focus on in this chapter.

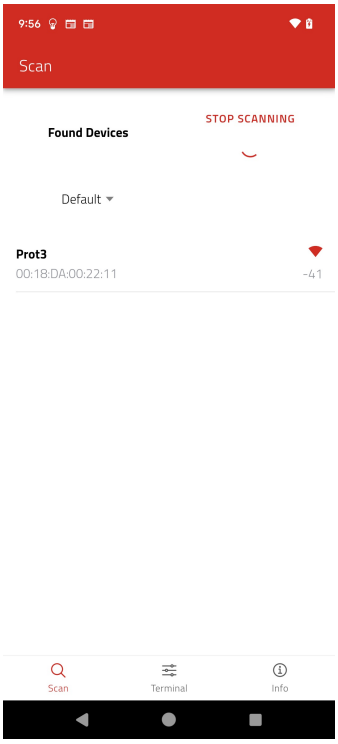
3.2.1 Boot-up and connection setup

- Connect the Proteus EV-Board to a PC using an USB cable.
- Open a terminal program using the Proteus default UART settings (115200 Baud, 8n1).
- Press the reset button on the Proteus EV-Board. The Proteus module outputs a `CMD_GETSTATE_CNF` to indicate that it is ready for operation.

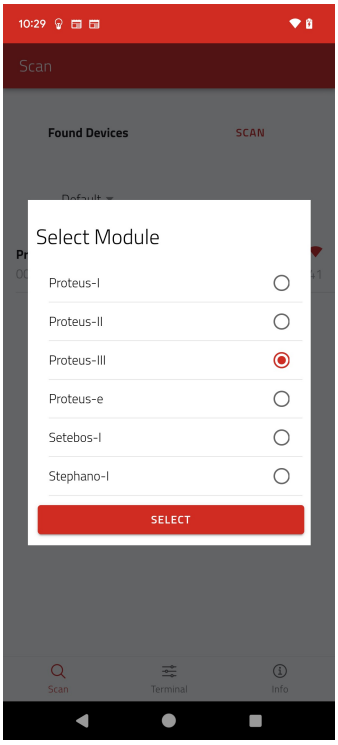
Info	Module 1

⇐ Response `CMD_GETSTATE_CNF`: Module 1 started in `ACTION_IDLE` mode.	02 41 02 00 01 01 41

- Then open the "WE Bluetooth LE Terminal" app and press "Scan". As soon as the Proteus-III/Proteus-e appears in the scan list, click on it to start the connection setup.

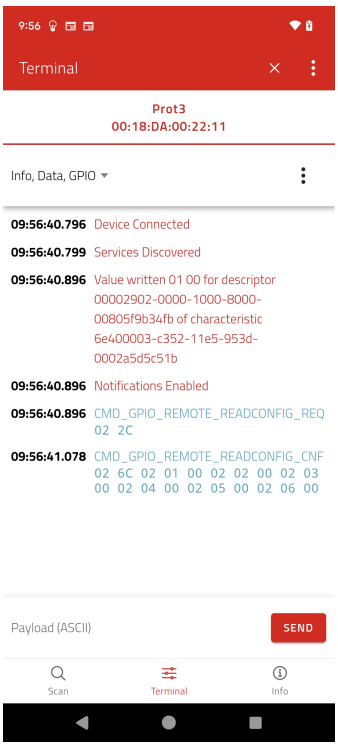


- In the app a new window pops up, which prompts you to select the Bluetooth® LE module you are connecting to.



- After selecting the module, the app switches to the the terminal tab which displays connection setup messages in the log window. The app automatically sends a `CMD_GPIO_REMOTE_READCONFIG_REQ` message (02 2C) to the radio module, which responds with a `CMD_GPIO_REMOTE_READCONFIG_CNF` message (02 6C 02 01 00 02 02 00 02 03 00 02 04 00 02 05 00 02 06 00), which states that all pins with GPIO_ID 1 to 6 are not configured.

Please note that the format of the radio commands differs from the format of the UART commands. The documentation of the radio command format for configuration and control of the GPIOs can be found in application note ANR009 [3] for Proteus-III and ANR024 [4] for Proteus-e.



- On the radio module side, the Proteus-III/Proteus-e outputs its connection setup related messages.

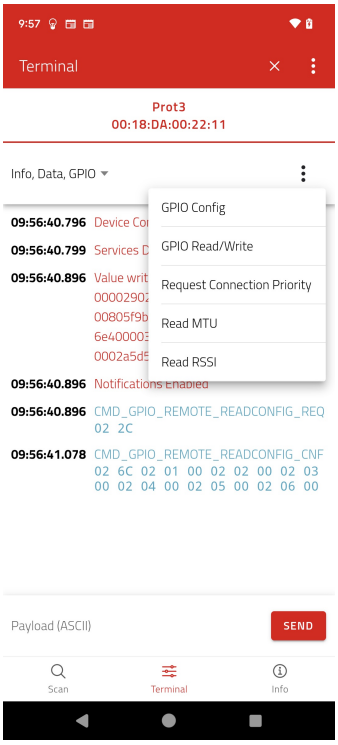
Info	Module 1
⇐ Indication `CMD_CONNECT_IND`: Physical connection established successfully to device with `FS_BTMAC` 0x55 0x00 0x00 0xDA 0x18 0x00	02 86 07 00 00 55 00 00 DA 18 00 14
⇐ Indication `CMD_CHANNELOPEN_RSP`: Channel opened successfully to device with `FS_BTMAC` 0x55 0x00 0x00 0xDA 0x18 0x00 and maximum payload size of 0xF3 (243 Bytes) per packet	02 C6 08 00 00 55 00 00 DA 18 00 F3 A7

- Now the Bluetooth® LE connection is open, and the configuration and control of the GPIOs of "Module 1" can be done.

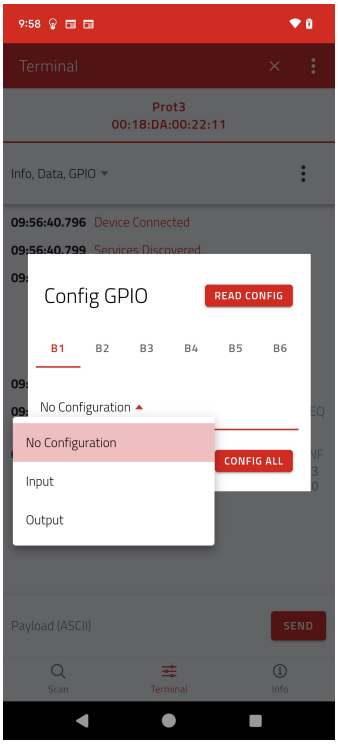
3.2.2 Configure and control a GPIO as output

If the Bluetooth® LE connection has been setup, as shown in chapter `3.2.1`, the following steps can be run to configure the GPIO *B1* with GPIO_ID 1 as output pin.

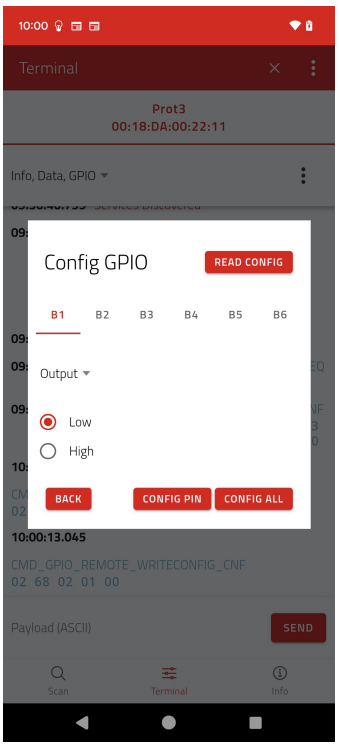
1. First of all configure the GPIO *B1* with GPIO_ID 1 as output pin with default level LOW. To do so, press the "..."-menu button and then "GPIO Config" .



2. A menu opens, which allows the configuration of the GPIOs. First of all, select pin *B1* and then chose "Output" from the pin configuration dropdown.



3. Now select LOW in the app and press "CONFIG PIN".

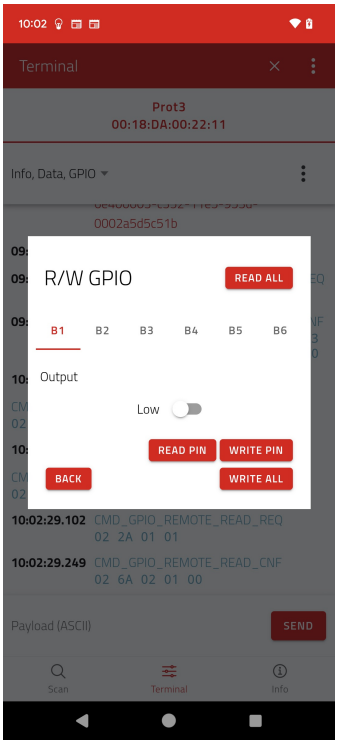


4. Pressing this button sends a `CMD_GPIO_REMOTE_WRITECONFIG_REQ` message (02 28 03 01 02 00 / configure GPIO with GPIO_ID 1 to output LOW) to the radio module, which responds with a `CMD_GPIO_REMOTE_WRITECONFIG_CNF` message (02 68 02 01 00 / configured GPIO with GPIO_ID 1 successfully).

5. On the radio module side, the Proteus-III/Proteus-e outputs the corresponding indication message

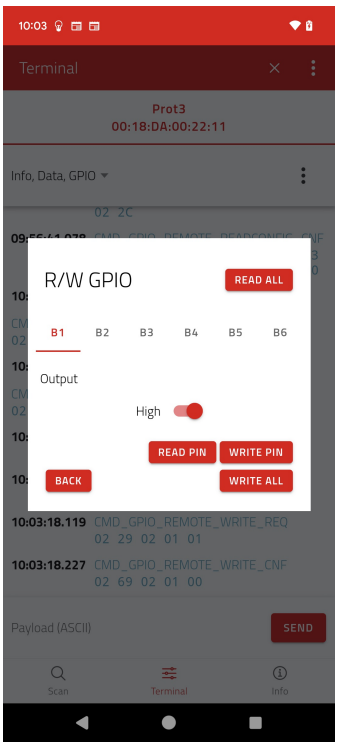
Info	Module 1
⇐ Indication `CMD_GPIO_REMOTE_WRITECONFIG_IND`: The GPIO with GPIO_ID 1 has been configured to output LOW by the remote device	02 A8 04 00 03 01 02 00 AE

6. Now, go to the sub menu "GPIO Read/Write" and press the refresh button to read all GPIO states.



7. Pressing this button sends a `CMD_GPIO_REMOTE_READ_REQ` message (02 2A 06 01 02 03 04 05 06 / request state of GPIO_ID 1 to 6) to the radio module, which responds with a `CMD_GPIO_REMOTE_READ_CNF` message (02 6A 02 01 00 02 02 FF 02 03 FF 02 04 FF 02 05 FF 02 06 FF), which states that the GPIO with GPIO_ID 1 is LOW, but the GPIOs with GPIO_ID 2 to 6 are not configured.

8. Next, press "B1", move the slider to HIGH and press "WRITE PIN" to set the GPIO with GPIO_ID 1 to HIGH.



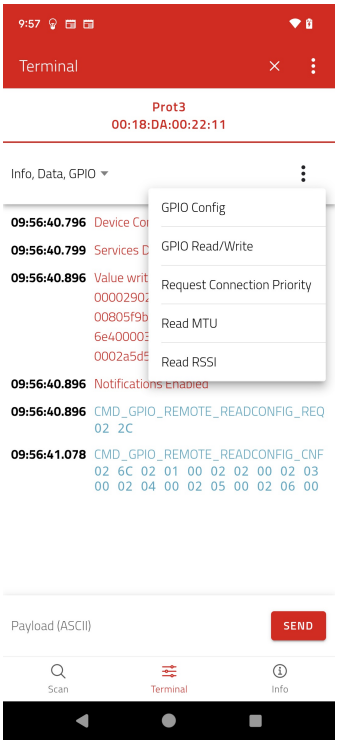
9. Pressing these buttons sends a `CMD_GPIO_REMOTE_WRITE_REQ` message (02 29 02 01 01 / set the GPIO with GPIO_ID 1 to HIGH) to the radio module, which responds with a `CMD_GPIO_REMOTE_WRITE_CNF` message (02 69 02 01 00), which states that the GPIO with GPIO_ID 1 has been set successfully.
10. On the radio module side, the Proteus-III/Proteus-e outputs the corresponding indication message

Info	Module 1
⇐ Indication `CMD_GPIO_REMOTE_WRITE_IND`: The GPIO with GPIO_ID 1 has been set to HIGH by the remote device	02 A9 03 00 02 01 01 AA

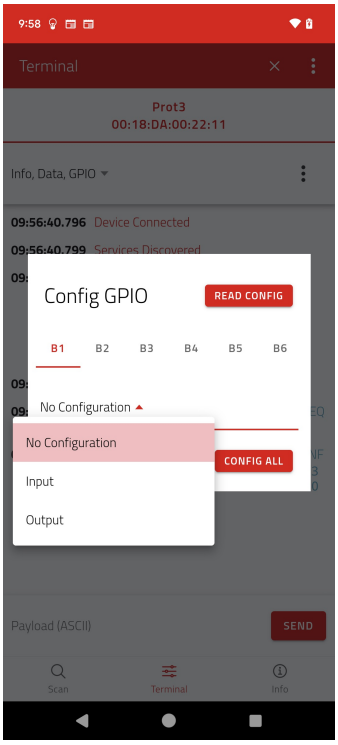
3.2.3 Configure and control a GPIO as input

If the Bluetooth® LE connection has been setup, as shown in chapter `3.2.1`, the following steps can be run to configure the GPIO *B1* with GPIO_ID 1 as input pin.

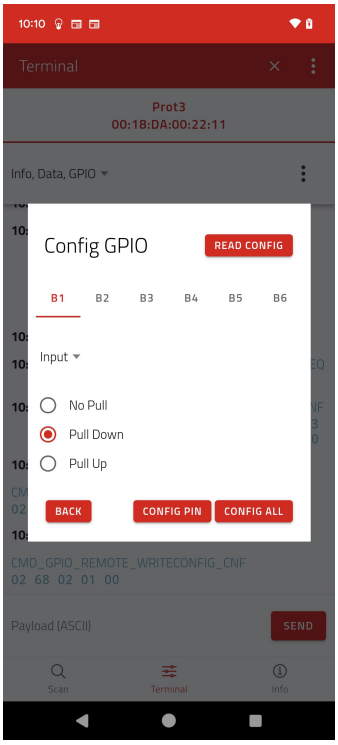
1. First of all configure the GPIO *B1* with GPIO_ID 1 as input pin with pull down resistor. To do so, press the "..."-menu button and then "GPIO Config" .



2. A menu opens, which allows the configuration of the GPIOs. First of all, select pin *B1* and then chose "Input" from the pin configuration dropdown.



3. Now select Pull Down in the app and press "CONFIG PIN".

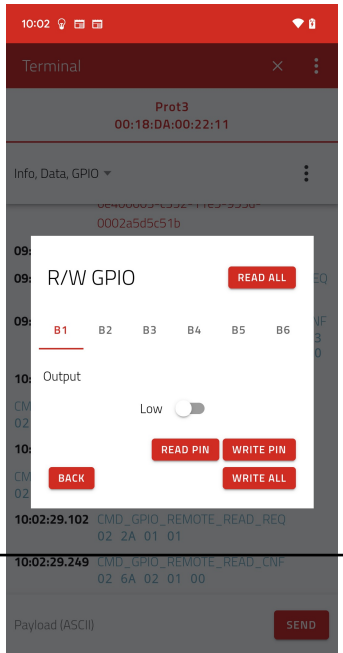


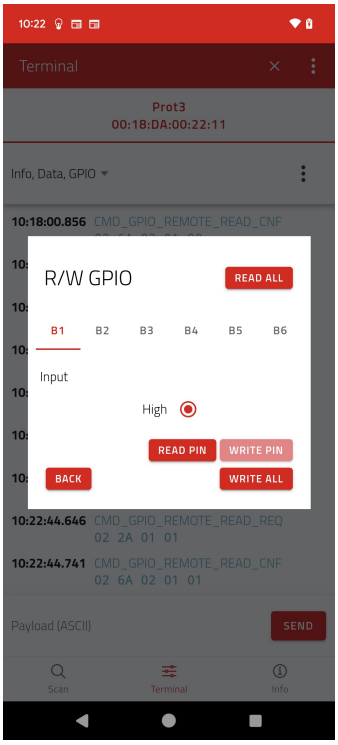
4. Pressing this button sends a `CMD_GPIO_REMOTE_WRITECONFIG_REQ` message (02 28 03 01 01 01 / configured GPIO with GPIO_ID 1 to input pull down) to the radio module, which responds with a `CMD_GPIO_REMOTE_WRITECONFIG_CNF` message (02 68 02 01 00 / configured GPIO with GPIO_ID 1 successfully).

5. On the radio module side, the Proteus-III/Proteus-e outputs the corresponding indication message

Info	Module 1
⇐ Indication `CMD_GPIO_REMOTE_WRITECONFIG_IND`: The GPIO with GPIO_ID 1 has been configured to input pull down by the remote device	02 A8 04 00 03 01 01 01 AC

6. Now, go to the sub menu "GPIO Read/Write" and press the refresh button to read all GPIO states.





9. Pressing these buttons sends a `CMD_GPIO_REMOTE_READ_REQ` message (02 2A 01 01 / read the GPIO with GPIO_ID 1 to the radio module, which responds with a `CMD_GPIO_REMOTE_READ_CNF` message (02 6A 02 01 01), which states that the GPIO with GPIO_ID 1 is HIGH.

4 References

[1] Würth Elektronik. Proteus-e user manual. https://www.we-online.de/katalog/de/manual/2612011024000.

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

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

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

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

5 Important notes

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

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

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

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

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

List of Figures

List of Tables

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

Max-Eyth-Straße 1
74638 Waldenburg
Germany

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