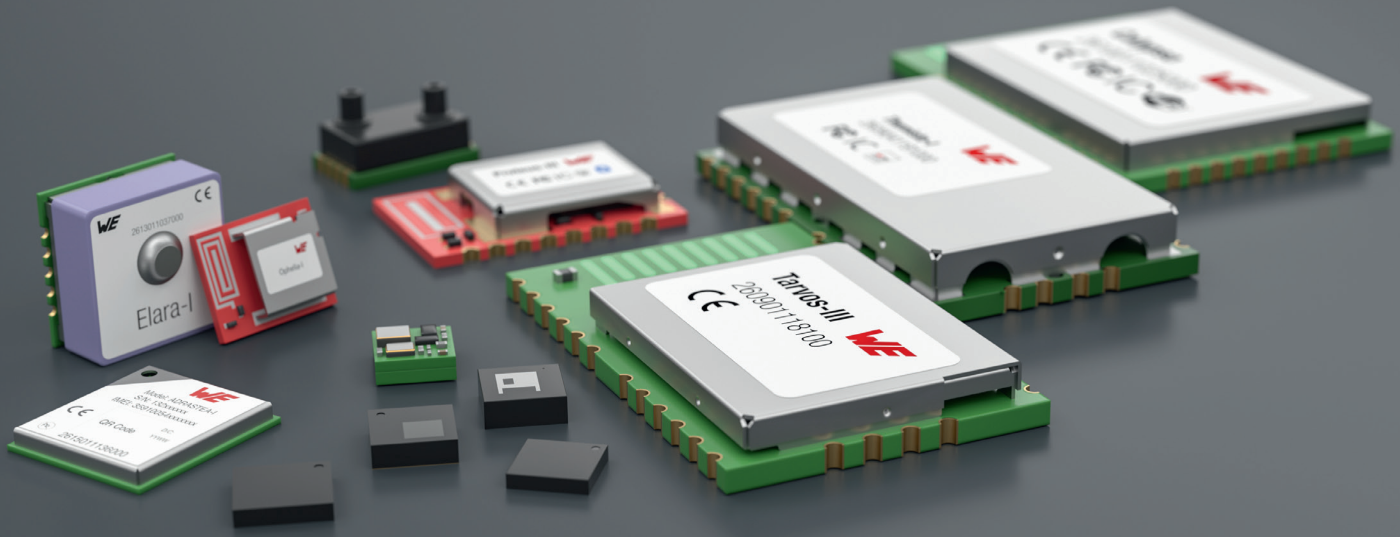


User manual

Stephano-I

2617011025000

Version 1.0

October 4, 2023

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

****

MUST READ

Check for firmware updates

Before using the product make sure you use the most recent firmware version, data sheet and user manual. This is especially important for Wireless Connectivity products that were not purchased directly from Würth Elektronik eiSos. A firmware update on these respective products may be required.

We strongly recommend to include in the customer system design, the possibility for a firmware update of the product.

Revision history

Manual version	FW version	HW version	Notes	Date
1.0	3.2.0.0	4.0	• Initial version	August 2023

Abbreviations

Abbreviation	Name	Description
ACK	Acknowledgement	Acknowledgement pattern confirming the reception of the transmitted data packet.
DC	Duty cycle	Transmission time in relation of one hour. 1% means, channel is occupied for 36 seconds per hour.
FOTA	Firmware over the air	Update mechanism that is run via radio.
FSE	Field Sales Engineer	Support and sales contact person responsible for limited sales area
0xhh [HEX]	Hexadecimal	All numbers beginning with 0x are stated as hexadecimal numbers. All other numbers are decimal.
HIGH	High signal level	Signal level of the VCC
LOW	Low signal level	Signal level of thr ground
LPM	Low power mode	Operation mode for reduced power consumption.
LSB	Least significant bit	
MCU	Micro controller unit	
MSB	Most significant bit	
PL	Payload	The real, non-redundant information in a frame/packet.
RF	Radio frequency	Describes everything relating to the wireless transmission.
UART		Universal Asynchronous Receiver Transmitter allows communicating with the module of a specific interface.
VCC	Supply voltage	

Contents

Overview of helpful application notes

Application note ANR008 - Wireless Connectivity Software Development Kit
http://www.we-online.com/ANR008
To ease the integration of the Würth Elektronik eiSos radio modules into an application, Würth Elektronik eiSos offers the corresponding Software Development Kit (SDK) for most commonly used host processors. This SDK contains drivers and examples in C-code to communicate with the corresponding radio module. This application note shows which SDKs are available and describes how to download and use them.

Application note ANR010 - Range estimation
http://www.we-online.com/ANR010
This application note presents the two most used mathematical range estimation models, Friis and two ray ground reflection, and its implementation in the range estimation tool of the RED-EXPERT.

Application note ANR027 - Bluetooth listing guide
http://www.we-online.com/ANR027
Every product containing Bluetooth® technology needs to be listed at the Bluetooth® SIG (special interest group). This application note explains the steps to be done to gain a Bluetooth® listing for the end product using a Würth Elektronik eiSos Bluetooth® LE radio module.

1 Introduction

The Stephano-I is a compact radio module that comes with a complete WiFi stack based on IEEE 802.11 b/g/n (2.4 GHz). Besides of that, the Stephano-I further brings Bluetooth® LE 5 functions in peripheral and central role enabling low power connectivity to smart devices.
This combination of widely used radio standards and an easy-to-use AT-style command interface makes the Stephano-I an ideal product for any embedded wireless IoT application.

1.1 Operational description

The Stephano-I radio module is intended to be used as a radio sub-system in order to provide WLAN and Bluetooth® LE communication capabilities to the system.
The UART acts as the primary interface between the module and a host micro-controller. The module can be fully configured and operated using a set of AT-commands over UART. Once configured, the module independently manages radio connectivity allowing the host controller to utilize its resources elsewhere.
Therefore a host host micro-controller is required in the end product to control and access the radio module.

1.2 Block diagram

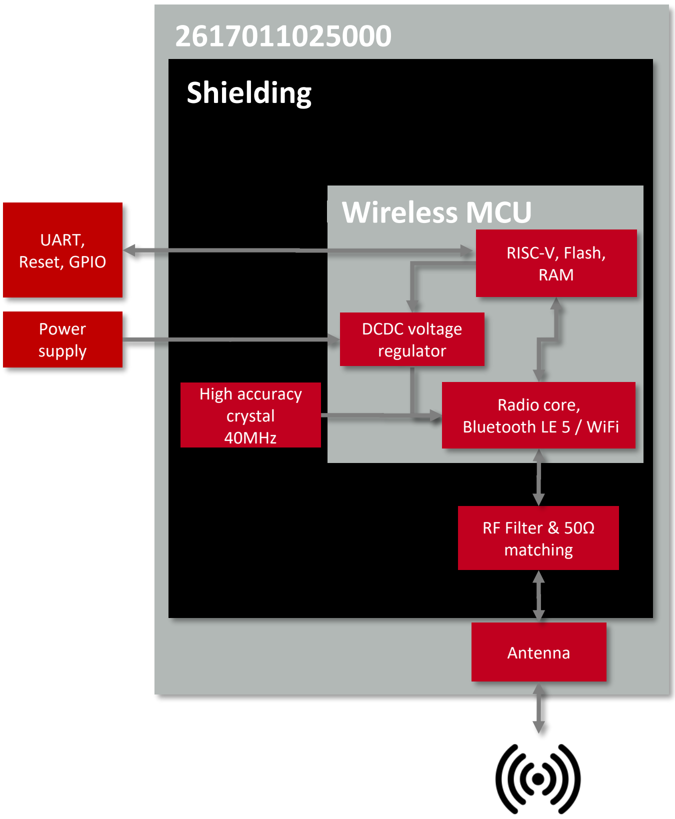


Figure 1: Block diagram

1.3 Ordering information

WE order code	Description
2617011025000	Stephano-I Wifi & Bluetooth® LE module with integrated PCB antenna, Tape & Reel
2617029025000	Evaluation board with Stephano-I module mounted

Table 1: Ordering information

2 Electrical specifications

As not otherwise stated measured on the evaluation board Stephano-I-EV with T = 25 °C, VCC = 3.3 V and 50 Ω conducted.

2.1 Recommended operating conditions

Description	Min.	Typ.	Max.	Unit
VCC	3.0	3.3	3.6	V
Temperature range	-40		85	°C

Table 2: Recommended operating conditions

2.2 Absolute maximum ratings

Description	Min.	Typ.	Max.	Unit
VCC	-0.3		3.6	V

Table 3: Absolute maximum ratings

2.3 Power consumption

2.3.1 Static

Description	Min	Typ.	Max	Unit
WiFi TX current consumption at max output power		167		mA
WiFi RX current consumption		82		mA
Bluetooth® LE TX current consumption at max output power		161		mA
Bluetooth® LE RX current consumption		81		mA
Deep-sleep mode		5		µA
System-off mode (*/RESET* set to *GND*)		1		µA

Table 4: Power consumption

2.4 Radio characteristics

Description	Min	Typ.	Max	Unit
Max output power (power index 80, 20 dBm)		13		dBm
Input sensitivity		-87		dBm
Frequencies	2412		2484	MHz

Table 5: WiFi radio characteristics (radiated)

Description	Min	Typ.	Max	Unit
Max output power[1] (power index 12, 9 dBm)		3.3		dBm
Input sensitivity		-89		dBm
Frequencies	2402		2480	MHz

Table 6: Bluetooth® LE radio characteristics (radiated)

2.5 Pin characteristics

Property	Min	Typ.	Max	Unit
Pin input low voltage	-0.3		0.25×VCC	V
Pin input high voltage	0.75×VCC		VCC+0.3	V
Pin output low voltage			0.1×VCC	V
Pin output high voltage	0.8×VCC			V
Pin output current sunk by any I/O and control pin		40		mA
Pin output current sourced by any I/O and control pin		28		mA
Internal pull-up/pull-down resistor		45		kΩ

Table 7: Pin characteristics

[1] Please refer the Table 14 for the certified power settings. For the certification of the end device, the power index can be adjusted to achieve the maximum certifiable output power.

3 Pinout

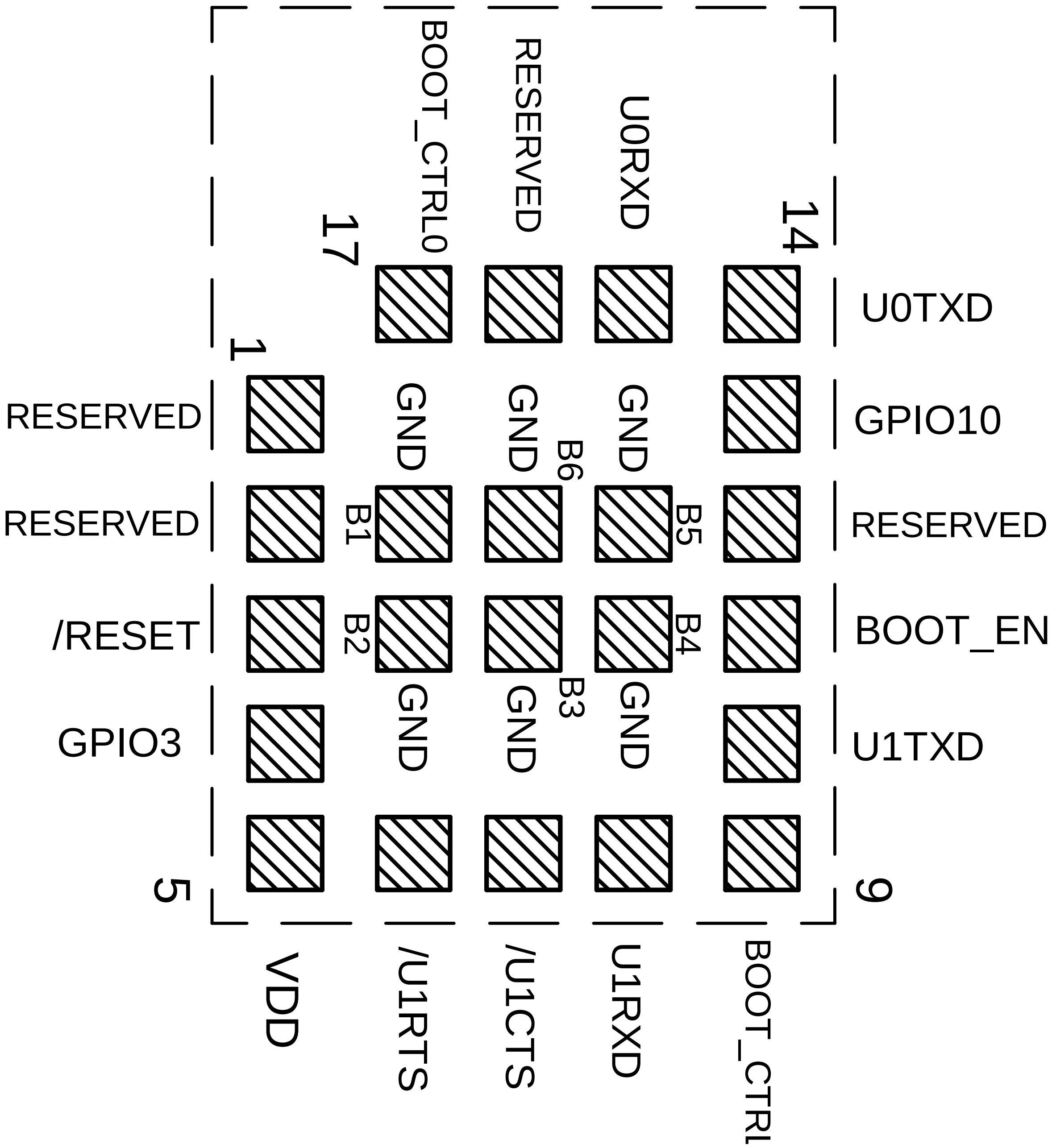


Figure 2: Pinout (top view)

No	µC Pin	Designation	I/O	Description
1	*GPIO0 / XTAL_P* [2]	*RESERVED*	I/O	Reserved pin. Do not connect.
2	*GPIO1 / XTAL_N* [2]	*RESERVED*	I/O	Reserved pin. Do not connect.
3	*CHIP_EN*	*/RESET*	Input	Reset pin. A low signal resets the module.
4	*GPIO3*	*GPIO3*	Input	Reserved pin. Can be used as wake-up source or ADC channel. Do not connect if not needed.
5		*VDD*	Supply	Supply voltage
6	*GPIO4*	*/U1RTS*	Output	/RTS signal of the application UART, if flow control is enabled. Do not connect if not needed.
7	*GPIO5*	*/U1CTS*	Input	/CTS signal of the application UART, if flow control is enabled. Uses internal pull-up [1]. Do not connect if not needed.
8	*GPIO6*	*U1RXD*	Input	Application UART (Reception). Uses internal pull-up resistor[1].
9	*GPIO8*	*BOOT_CTRL1*	Input	Boot control pin. External 1.5 kΩ pull-up resistor needed.
10	*GPIO7*	*U1TXD*	Output	Application UART (Transmission)
11	*GPIO9*	*BOOT_EN*	Input	Boot control pin. Low level: Boot Mode. High level: Application mode. Uses internal pull-up [1]
12	*GPIO18 / USB D-* [3]	*RESERVED*	I/O	Reserved pin. Do not connect.
13	*GPIO10*	*GPIO10*	Input	Reserved pin. Can be used as wake-up source. Do not connect if not needed.
14	*GPIO21*	*U0TXD*	Output	Debug UART (Transmission). Do not connect if not needed.
15	*GPIO20*	*U0RXD*	Input	Debug UART (Reception). Uses internal pull-up resistor[1]. Do not connect if not needed.
16	*GPIO19 / USB D+* [3]	*RESERVED*	I/O	Reserved pin. Do not connect.
17	*GPIO2*	*BOOT_CTRL0*	Input	Boot control pin. External 1.5 kΩ pull-up resistor needed.
B1 -B6	*GND*	*GND*	Supply	Ground

Table 8: Pinout

[1] Internal pull-ups or pull-downs (45 kΩ) are configured at start-up by the firmware installed in the SoC.
[2] Pins available to connect an external crystal in custom firmware only.
[3] Pins available to connect an USB in custom firmware only.

4 Quickstart

4.1 Minimal pin configuration

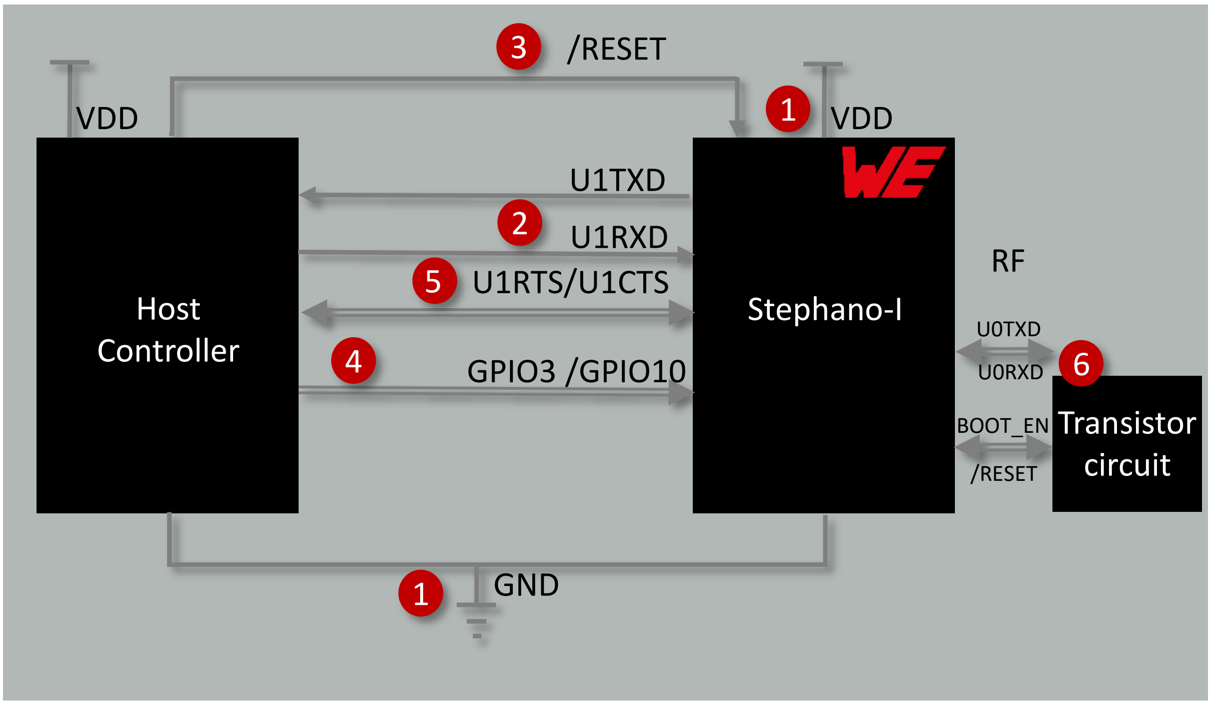


Figure 3: Minimal pin connections

The above image shows the steps to be performed to integrate the Stephano-I into a custom end device.

1. Supply voltage and ground
   First of all, connect the *VDD* and *GND* pins to supply the radio module with power.

2. UART serial interface to the host
   Connect the application UART pins *U1TXD* and *U1RXD* to the host to control the module via host.

3. Reset
   Connect the */RESET* pin to the host to allow a hard reset of the module.

4. (Optional) Wakeup from sleep
   Connect the *GPIO3* or *GPIO10* pin to use it as trigger to wake-up from sleep mode.

5. (Optional) UART flow control
   Connect the */U1RTS* and */U1CTS* pins to the host controller in case the UART flow control shall be used.

6. (Optional) Debug UART and flash control for end device certification
   To run the certification process of the end device, that integrates the Stephano-I radio module, it may be necessary that the pins *U0RXD* and *U0TXD* are accessible. Furthermore a special transistor circuit is needed to control the */RESET* and *BOOT_EN* pins to flash special test firmware. For more details of the hardware layout see chapter `14.3`.

4.2 Power up

After a stable power supply has been applied to the module, the */RESET* pin can be released to high state to boot-up the Stephano-I. As soon as the radio module has booted, a start-up message "ready" is sent via UART to the connected host. From this point on, the Stephano-I is ready to be controlled via AT-commands by the host micro controller.

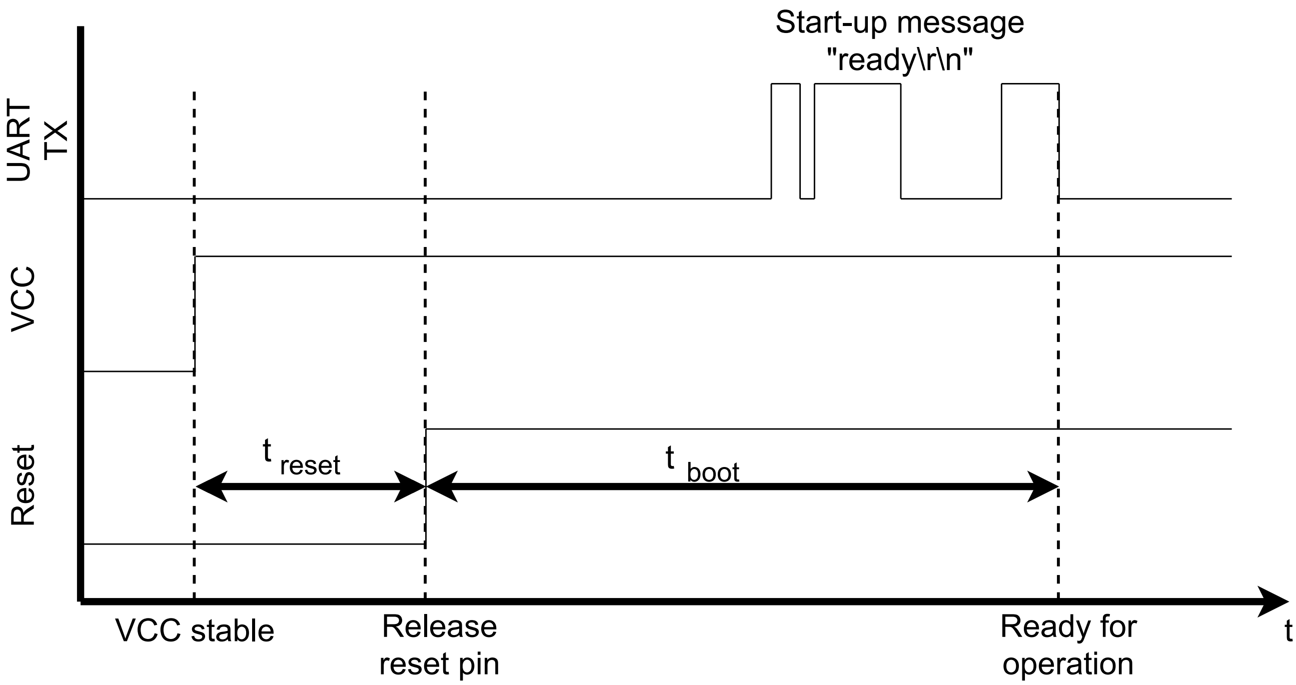


Figure 4: Power up

4.3 Quickstart example

This section describes how to quick start the Stephano-I. In this example the goal is to connect to a router as WiFi station, set up a TCP connection and transmit data.

The below commands are in ASCII notation. Due to simplicity the \r\n that is attached to each command will be neglected in the below example.
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 response or event message is always sent from module to host (⇐).

1. First of all reset the device by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info	Message
⇐ Module is ready for operation	ready

2. Then set to module to "WiFi station" mode and connect to your local WiFi:

Info	Message
⇒ Configure the Stephano-I as WiFi station	AT+CWMODE=1
⇐ Configuration done successfully	OK
⇒ Request a connection setup to an access point with SSID "mySSID" and password "myPWD"	AT+CWJAP="mySSID","myPWD"
⇐ WiFi has been connected and IP has been received	WIFI CONNECTED WIFI GOT IP OK
⇒ Request the assigned IP	AT+CIPSTA?
⇐ The router has assigned "191.178.168.228"	+CIPSTA:ip:"191.178.168.228" +CIPSTA:gateway:"191.178.168.1" +CIPSTA:netmask:"255.255.255.0" OK

3. Now a TCP server is needed that is also connected to the same network. We assume that the server's IP address is 191.178.168.229, and the port is 8080.

4. Then setup a TCP connection from the Stephano-I to that server:

Info	Message
⇒ Request a connection to that server	AT+CIPSTART="TCP","191.178.168.229",8080
⇐ Connection set up successfully	CONNECT OK

5. Now transmit "hello server" (12 bytes):

Info	Message
⇒ Request data transmission of 12 bytes	AT+CIPSEND=12
⇐ Stephano-I is ready for data reception	OK >
⇒ Now send the data	hello server
⇐ Data transmitted successfully to the connected TCP server	Recv 12 bytes SEND OK

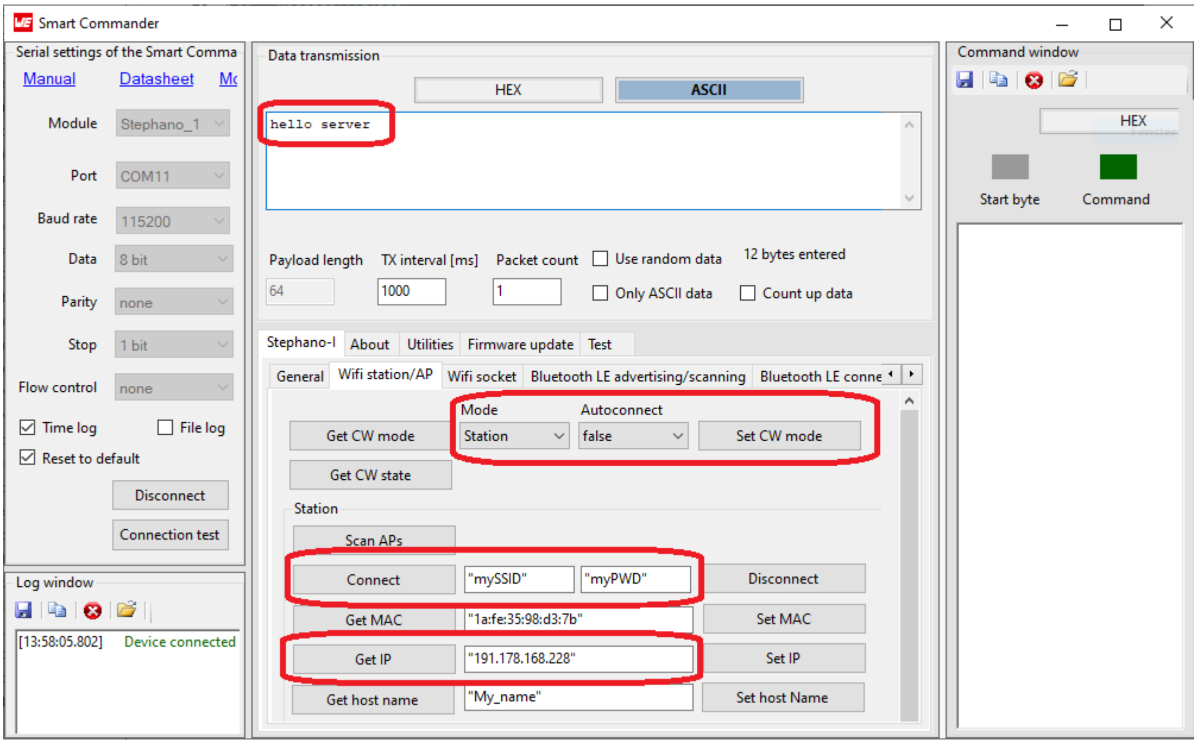
6. Now assume that the server replies with "hello client":

Info	Message
⇐ Server send data to the client	+IPD,12:hello client

The Espressif web page [3] contains numerous examples for Wifi and Bluetooth® LE connectivity as well as for low power operation. Please find also the chapter `Use cases and examples` for more examples.

4.3.1 Smart commander

The Smart commander [12] is a PC tool for Windows. It implements the UART command interface of the Stephano-I. Since all device functions are abstracted to buttons and drop down menus, the Smart commander significantly simplifies prototyping and product evaluation.



Module: Stephano_1
Port: COM11
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

☑ Time log □ File log
☑ Reset to default

[13:58:05.802] Device connected

```
hello server
```

Payload length: 64
TX interval [ms]: 1000
Packet count: 1
□ Use random data
□ Only ASCII data
12 bytes entered
□ Count up data

Mode: Station
Autoconnect: false

Connect "mySSID" "myPWD"
Get MAC "1a:fe:35:98:d3:7b"
Get IP "191.178.168.228"
Get host name "My_name"

Figure 5: Smart Commander PC tool for prototyping

5 Functional description

The Stephano-I radio module is intended to be used as a radio sub-system in order to provide WLAN and Bluetooth® LE communication capabilities to the system.

5.1 Key features

The Stephano-I offers the following key features:

- General features:

  Fast serial interface: The Stephano-I offers a UART-interface to communicate with a host using a simple AT-command interface. A baud rate of up to 5 MBaud, the flow control, the parity as well as the number of start and stop bits can be configured to match the applications needs.

  Latest microprocessor generation provided by Espressif: The heart of the Stephano-I is the ESP32-C3 radio chip offering high performance values combined with low power consumption. It is a 32-bit RISC-V single-core processor with up to 160 MHz.

  Low power capabilities: The Stephano-I provides several sleep modes that allow to shut down different parts of the radio module. This enables the option to reduce the power consumption depending in the application's state.

- Bluetooth® LE features:

  Bluetooth® LE 5 stack: The Bluetooth® LE 5 stack enables fast and energy efficient data transmission using state-of-the-art radio technology.

  All Bluetooth® LE roles supported: The integrated Bluetooth® LE stack supports all Bluetooth® LE roles.

  Peripheral: Provides a simple custom profile "WE SPP-like" for transmission of generic data (see chapter 5.2).

  Central: By use of generic GATT functions allows to communicate to any other Bluetooth® LE device, independent of the provided services.

  Broadcaster: Allows to transmit beacons.

  Observer: Allows to receive beacons.

  Additional Bluetooth® LE 5 radio modes: Besides the legacy Bluetooth® LE data rate of 1 MBit, the Stephano-I provides the advanced data rate of 2 MBit mode for faster data transmission and LE-Coded mode for higher range.

  Pre-configured Bluetooth® LE profiles: Besides the profile for generic data transmission "WE SPP-like", the Stephano-I provides the HID (human interface device) profile.

- WiFi features:

  Radio Standards: The Stephano-I is equipped with IEEE 802.11b/g/n protocol that can act as station as well as soft access point.

  Security: Provides WPAPSK, WPA2PSK and WPA/WPA2PSK encryption method when configured as soft access point.

Full TCP/IP stack on board: A complete TCP/IP stack is included.

Advanced protocols: The Stephano-I can act as MQTT as well as HTTP client for easy connectivity and data transmission.

Provisioning: The credentials of the local WiFi network can be entered via web page or UART command.

FOTA: The Stephano-I provides the opportunity to update the device firmware via WiFi interface. Furthermore, certificates and keys for SSL communication can be updated via this interface.

5.2 Bluetooth LE: WE SPP-like profile

The "WE SPP-like" profile is a custom Bluetooth® profile used in the Würth Elektronik eiSos Bluetooth® LE module series Proteus as well as in Stephano-I when acting as peripheral (server).
The profile consists of a primary service containing two characteristics:

Service	Full UUID	Short explanation
Primary service	6E400001-C352-11E5-953D-0002A5D5C51B	
`RX_CHARACTERISTIC`	6E400002-C352-11E5-953D-0002A5D5C51B	Used to send data from the central to the peripheral
`TX_CHARACTERISTIC`	6E400003-C352-11E5-953D-0002A5D5C51B	Used to send data from the peripheral to the central

Table 13: WE SPP-like UUIDs

- The first characteristic of the Stephano-I primary service is `RX_CHARACTERISTIC`:
  - The data is sent from central/client to peripheral/server using a write command.
  - Server:
    - Has to allow a write command as well as a write without response command.
  - Client:
    - Use write command to send data to the server.
- The second characteristic of the Stephano-I primary service is `TX_CHARACTERISTIC`:
  - The data is sent from peripheral/server to central/client using a notification.
  - Server:
    - Has to allow/enable notifications. Notify client/central when sending data.
  - Client:
    - Has to enable notifications. To do that the central must write the notification enable bit in the `CCCD` (Client Characteristic Configuration Descriptor) of this characteristic.

STOP

Please note, that when communicating with a radio module from Würth Elektronik eiSos Proteus series (Proteus-I,-II,-III,-III-SPI,-e as well as Setebos-I) via Bluetooth® LE, a header byte 0x01 must be prepended to the payload. Otherwise the Proteus module will reject the data message.

6 Host connection

6.1 Serial interface: UART

The configuration in factory state of the UART is 115200 baud with data format of 8 data bits, no parity, 1 stop bit ("8n1") and flow control disabled. The configuration such as baud rate of the UART can be updated by means of the command `AT+UART_DEF` [1].

7 The command interface

The Stephano-I provides a simple AT-command interface for configuration and control of the WiFi and Bluetooth® LE functions. For command interface documentation, please refer to the Espressif command documentation [1].

Please note the Windows PC tool Smart commander (see chapter `Smart commander`) that implements all commands for rapid prototyping.

7.1 Wireless connectivity SDK

Besides of the Smart commander PC tool, there is the so called "Wireless connectivity SDK" [14] which is a implementation of all AT commands into C code. This implementation can be used to quickly integrate the UART connectivity to the connected host micro controller:

```
...
/**
* @brief Bluetooth LE init type
*/
typedef enum ATBluetoothLE_InitType_t
{
    ATBluetoothLE_InitType_None = 0,
    ATBluetoothLE_InitType_Client = 1,
    ATBluetoothLE_InitType_Server = 2,
} ATBluetoothLE_InitType_t;

extern bool ATBluetoothLE_Init(ATBluetoothLE_InitType_t type);
extern bool ATBluetoothLE_SetDeviceID(ATBluetoothLE_Address_t type,
    ATBluetoothLE_DeviceID_t ID);
extern bool ATBluetoothLE_GetDeviceID(ATBluetoothLE_DeviceID_t *ID);
extern bool ATBluetoothLE_SetDeviceName(ATBluetoothLE_DeviceName_t dev_name);
extern bool ATBluetoothLE_GetDeviceName(ATBluetoothLE_DeviceName_t *dev_name);
...
```

Code 1: StephanoI/ATCommands/ATBluetoothLE.h

8 Radio power settings

The radio output power of the Stephano-I can be adjusted by the `AT+RFPOWER` command. Depending on the country of operation, the power index can be modified accordingly.

Country	Maximum Bluetooth® LE power index	Maximum WiFi power index
Factory default	12 (9 dBm)	80 (20 dBm)
TELEC/Japan	12 (9 dBm)	74 (18.5 dBm)[1]
RED/EU	13 (12 dBm)	80 (20 dBm)
FCC/US	13 (12 dBm)	80 (20 dBm)
IC/Canada	13 (12 dBm)	80 (20 dBm)

Table 14: Maximum allowed power setting

!

Please note that the actual radiated output power is lower than the chipset's power setting.

8.1 Supported power indices

Power index	Conducted power value [dBm]
0	-27
1	-24
2	-21
3	-18
4	-15
5	-12
6	-9
7	-6
8	-3
9	0
10	3
11	6
12	9
13	12

Table 15: Bluetooth® LE power table

[1] For country code "Japan" the power index is internally limited to 74 (18.5 dBm).

Power index	Conducted power value [dBm]
40-80	Index × 0.25
40	10
⋮	⋮
80	20

Table 16: WiFi power table

9 Timing parameters

9.1 Throughput - Wi-Fi

9.1.1 Test procedure

The test procedure is as follows:

1. Set the Wi-Fi mode to station using the command `AT+CWMODE`.
2. Connect to the router using the command `AT+CWJAP`.
3. Use a network tool a the PC to create a TCP server.
4. Connect the module to the TCP server as a client over TCP using the command `AT+CIPSTART`.
5. Iterate over:
   a) Send data from the module to the PC by making use of the command `AT+CIPSEND`. If the data length exceeds 8192 bytes, use the `AT+CIPSENDL` command instead.
   b) Wait for the transmission success message (`OK`) that is returned on the UART as result of the previous transmission request.
6. Follow the same procedure for UDP throughput test.

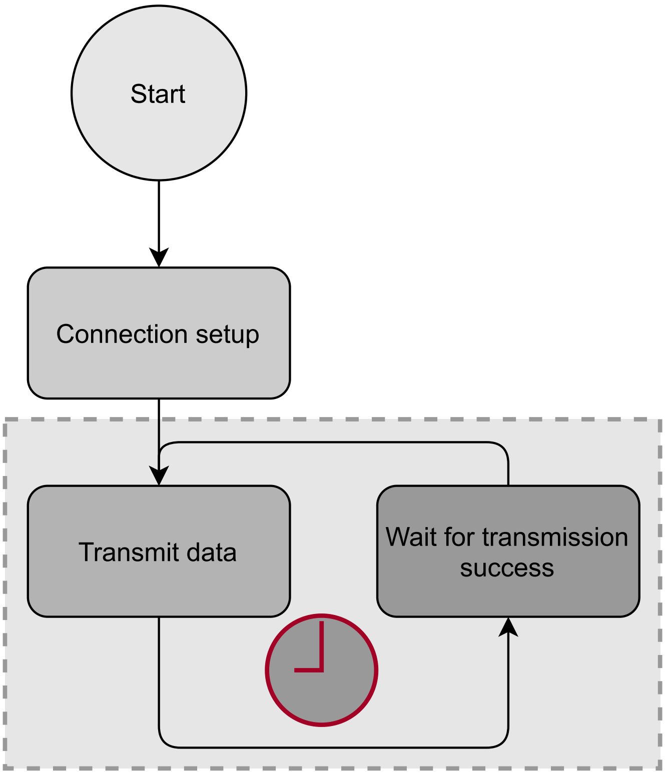


Figure 6: Flow diagram throughput measurement

9.1.2 Test setup and test results

The Stephano-I module is connected via UART to a STM32F4 micro controller, that triggers the repeated data transmission.

Device	Baud rate [Baud]	Packet size [Bytes]	Throughput [kBytes/s]
Stephano-I TCP Client	3000000	8192	168.213
Stephano-I TCP Client	3000000	16384	226.611
Stephano-I UDP Client	3000000	8192	168.559
Stephano-I UDP Client	3000000	16384	227.239

Table 17: Test results via STM32

9.2 Throughput - Bluetooth LE

9.2.1 Test procedure

The test procedure is as follows:

1. Turn WiFi off using the command `AT+CWMODE`.
2. Setup a Bluetooth connection between the two Bluetooth® LE enabled devices. We call the transmitting device "Device A" and the receiving device "Device B".
3. Set the MTU Φ_{ST} of the connection to 512 Bytes using the command `AT+BLECFGMTU`.
4. Switch the PHY to 2 Mbit using the `AT+BLESETPHY` command.
5. Iterate over:
   a) Transmit a data packet of maximum size Φ_{ST} Bytes using a `BLEGATTWRT` (central transmission) or `BLEGATTSNTFY` (peripheral transmission) command.
   b) Wait for the transmission success message (`OK`) that is returned on the UART as result of the previous transmission request.

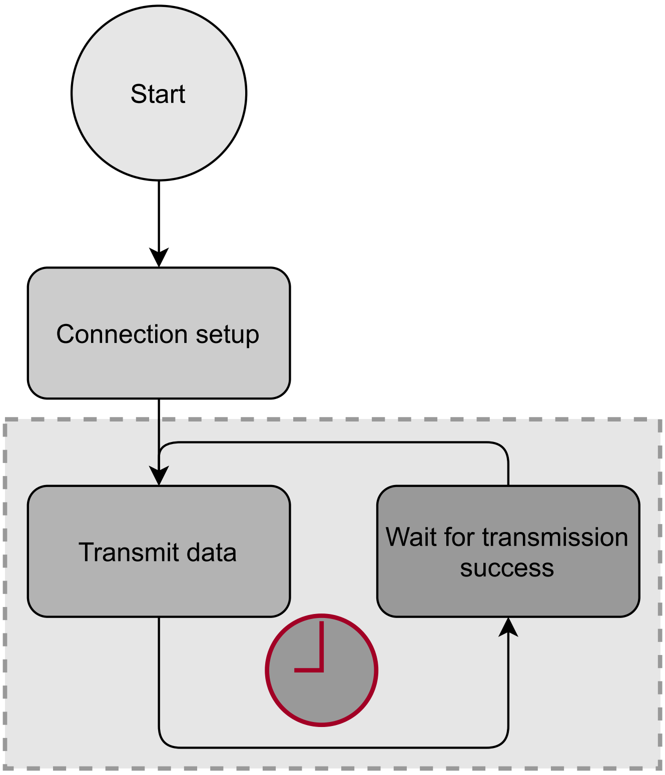


Figure 7: Flow diagram throughput measurement

9.2.2 Test setup and test results

The Stephano-I module is connected via UART to a STM32F4 micro controller, that triggers the repeated data transmission.

Device A	Device B	Baud rate [Baud]	Connection interval [ms]	Throughput [kBytes/s]
Stephano-I Central	Stephano-I Peripheral	3000000	7.5	40.2

Table 18: Test results via STM32

10 Use cases and examples

The Espressif documentation [3] contains numerous examples for Wifi and Bluetooth® LE connectivity as well as for low power operation. Most of these examples are valid for the Stephano-I, too. Exceptions are the examples using Bluetooth® LE profiles, as the Stephano-I implements special profiles invented by Würth Elektronik eiSos.
Please find examples for special applications, like the Würth Elektronik eiSos Bluetooth® LE profile "WE SPP-like", below.

❗ Please note that when using Bluetooth® LE and WiFi interface congruently, the radio performance of each radio standard will be slowed down due to the shared transmission medium.

10.1 Bluetooth LE examples

10.1.1 Central: Connect to Proteus radio module using WE SPP-like profile

In this chapter the goal is to configure the Stephano-I as Bluetooth® LE central. Then connect to a Proteus radio module acting as peripheral for bidirectional data transmission. Proteus-e [7], Proteus-I [8], Proteus-II [9], Proteus-III [11] and Proteus-III-SPI [10] support this function. To do that, the services, characteristics and the corresponding UUIDs of the "WE SPPlike" profile must be known. Please check out chapter 5.2 for that.
In this example, the peripheral is the Proteus device, and the central is the Stephano-I.

❗ Please note that payload data that is written to the `TX_CHARACTERISTIC` or `RX_CHARACTERISTIC` must be prefixed by the WE SPP-like header 0x01. Otherwise, the payload data will be rejected by the Proteus.
In the following example only the header byte is written in hex notation. All other data is ascii notation.

1. First of all reset the device by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info	Message
⇐ Module is ready for operation	ready

2. Then enable the Bluetooth® LE central function of the Stephano-I:

Info	Message
⇒ Enable the central function	AT+BLEINIT=1
⇐ Module is ready for operation	OK

3. The next step is to scan the radio to find available Proteus radio modules:

Info	Message
⇒ Scan for 2 s to find available devices	AT+BLESCAN=1,2
⇐ Found Bluetooth® LE device with MAC "00:18:da:00:00:05" and UUID 1bc5d5a502003d95e51152c30100406e and device name "Prot3"	OK +BLESCAN:"00:18:da:00:00:05", -50, 020106 020a08 11071bc5d5a502003d95e51152c30100406e 060950726f7433,,0

4. Now setup the physical connection:

Info	Message
⇒ Connect to BLE device with MAC 00:18:da:00:00:05 on connection index 0	AT+BLECONN=0,"00:18:da:00:00:05"
⇐ Connection has been setup with MTU of 247 bytes	+BLECFGMTU:0,247 +BLECONN:0,"00:18:da:00:00:05" OK

5. The next step is to discover the services of the Proteus device to see what services it provides:

Info	Message
⇒ Discover services	AT+BLEGATTCPRIMSRV=0
⇐ Found service "Generic Access" and "Generic Attribute"	+BLEGATTCPRIMSRV:0,1,0x1800,1 +BLEGATTCPRIMSRV:0,2,0x1801,1
⇐ Found WE SPP-like primaary service with service index 3	+BLEGATTCPRIMSRV:0,3, 0x6E400001C35211E5953D0002A5D5C51B,1 OK

6. Now discover the characteristic of the WE SPP-like primary services to see what characteristics it provides:

Info	Message
⇒ Discover characteristics of service index 3	AT+BLEGATTCCHAR=0,3
⇐ Found WE SPP-like `RX_CHARACTERISTIC` characteristic with characteristics index 1	+BLEGATTCCHAR:"char",0,3,1, 0x6E400002C35211E5953D0002 A5D5C51B,0x0c
⇐ Found WE SPP-like `TX_CHARACTERISTIC` characteristic with characteristics index 2	+BLEGATTCCHAR:"char",0,3,2, 0x6E400003C35211E5953D0002 A5D5C51B,0x10

⇐ Found descriptor with descriptor index 1 for characteristic index 2	+BLEGATTCCHAR:"desc",0,3,2,1,0x2902 OK

7. To be able to receive data from the Proteus device, the notification of the `TX_CHARACTERISTIC` must be enabled:

Info	Message
⇒ Enable notification by writing 0x0001 (2 bytes) to the descriptor of `TX_CHARACTERISTIC` (char. index 2)	AT+BLEGATTCWR=0,3,2,1,2
⇐ Stephano-I is ready to receive data via UART	>
⇒Now send the data	0x01 0x00 (hex, no ascii)
⇐	OK

8. On the Proteus device, send "Hello Stephano-I" to receive it on the Stephano-I side. See `CMD_DATA_REQ` commands in Proteus user manual [7, 8, 9, 11, 10]:

Info	Message
⇐ Received a notification on the `TX_CHARACTERISTIC` with payload "0x01 Hello Stephano-I"	+NOTIFY:0,3,2,17,0x01 Hello Stephano-I

9. Then transmit "Hello Proteus" to the Proteus device by writing the data to the `RX_CHARACTERISTIC` (char. index 1). Here the header byte 0x01 must be prefixed:

Info	Message
⇒ Write to the `RX_CHARACTERISTIC`	AT+BLEGATTCWR=0,3,1„14
⇐ Stephano-I is ready to receive data via UART	>
⇒ Transmit "Hello Proteus" with a prefixed 0x01 (WE SPP Header)	0x01 Hello Proteus
⇐	OK

10. (Optional) Use the passthrough mode. The passthrough mode allows to bind certain characteristics to the UART interface. Thus, when the passthrough mode has been enabled, data sent via UART to the Stephano-I is forwarded to the bound outgoing characteristic. Furthermore, data sent via Bluetooth® LE to the bound incoming characteristics is forwarded to the UART. Thus the passthrough mode acts like a transparent bridge between UART and Bluetooth® LE interface.

Data sent in passthrough mode does not need to be attached by a "\r\n".

a) Configure and start the passthrough mode. Configuration is done by binding the `TX_CHARACTERISTIC` (srv. index 3, char. index 1) and `RX_CHARACTERISTIC` (srv. index 3, char. index 2) with the UART interface:

Info	Message
⇒ Configure the passthrough mode	AT+BLESPPCFG=1,3,1,3,2,0
⇐	OK
⇒ Start the passthrough mode	AT+BLESPP
⇐	OK >

b) Now transparent data can be send to the Stephano-I for Bluetooth® LE transmission. Here we choose "Hey Proteus":

Info	Message
⇒ Transmit "Hey Proteus" with a prefixed 0x01 (WE SPP Header)	0x01 Hey Proteus

c) Data is received on the smart phone. Smart phone replies with "Hey Stephano-I":

Info	Message
⇐ "Hey Stephano-I" with a prefixed 0x01 (WE SPP Header) is received by the Stephano-I	0x01 Hey Stephano-I

d) Stop passthrough mode:

Info	Message
⇒ Stop the passthrough mode by sending "+++"	+++

The "+++" must be send without "\r\n". The "+++" is not confirmed by an "OK".

11. The data has been received on the Proteus device. Now the connection can be closed if no further data is supposed to be transmitted:

Info	Message
⇒ Disconnect on connection index 0	AT+BLEDISCONN=0
⇐	OK

10.1.2 Peripheral: Receive connection from Proteus radio module using WE SPP-like profile

In this chapter the goal is to configure the Stephano-I as Bluetooth® LE peripheral and use the integrated WE SPP-like Bluetooth® LE profile that is also part of the Proteus Bluetooth® LE module series. Then connect from a Proteus radio module acting as central to the Stephano-I for bidirectional data transmission. Proteus-I [8], Proteus-II [9], Proteus-III [11] and Proteus-III-SPI [10] support this function.
To do that, the services, characteristics and the corresponding UUIDs of the "WE SPPlike" profile must be known. Please check out chapter 5.2 for that.
In this example, the central is the Proteus device, and the peripheral is the Stephano-I.

> Please note that payload data that is written to the `TX_CHARACTERISTIC` or `RX_CHARACTERISTIC` must be prefixed by the WE SPP-like header 0x01. Otherwise, the payload data will be rejected by the Proteus.
> In the following example only the header byte is written in hex notation. All other data is ascii notation.

1. First of all reset the device by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info	Message
⇐ Module is ready for operation	ready

2. Then enable the Bluetooth® LE peripheral function of the Stephano-I:

Info	Message
⇒ Enable the peripheral function	AT+BLEINIT=2
⇐	OK

3. Start service and check what is available on the Stephano-I:

Info	Message
⇒ Create the local services	AT+BLEGATTSSRVCRE
⇐	OK
⇒ Start the local services	AT+BLEGATTSSRVSTART
⇐	OK
⇒ Request which services are available	AT+BLEGATTSSRV?
⇐ WE SPP-like primary service is available on service index 1	+BLEGATTSSRV:1,1, 0x6E400001C35211E5953D0002A5D5C51B,1 OK

⇒ Request which characteristic are available	AT+BLEGATTSCHAR?
⇐ WE SPP-like `RX_CHARACTERISTIC` is available on characteristic index 1	+BLEGATTSCHAR:"char",1,1, 0x6E400002C35211E5953D0002A5D5C51B, 0x0c
⇐ WE SPP-like `TX_CHARACTERISTIC` is available on characteristic index 2	+BLEGATTSCHAR:"char",1,2, 0x6E400003C35211E5953D0002A5D5C51B, 0x10
⇐ Descriptor with index 1 is available for characteristic index 2	+BLEGATTSCHAR:"desc",1,2,1,0x2902 OK

4. The next step is to set the right advertising data such that the Proteus device is able to detect the Stephano-I. Therefore at least the advertising flags and the UUID of the WE SPP-like primary service 1BC5D5A502003D95E51152C30100406E must be placed into the advertising packet:

Info	Message
⇒ Set the advertising data	AT+BLEADVDATA="020106 11071BC5D5A502003D95E51152C30100406E"
⇐	OK
⇒ Start advertising	AT+BLEADVSTART
⇐	OK

5. On the Proteus side, scan for the advertising Stephano-I and connect to that. See `CMD_SCANSTART_REQ` and `CMD_CONNECT_REQ` commands in Proteus user manual [8, 9, 11, 10]:

Info	Message
⇐ Connection has been established on connection index 0 with MTU 247	+BLECONN:0,"00:18:da:00:00:05" +BLECFGMTU:0,247
⇐ 0x0100 has been written to the descriptor of the `TX_CHARACTERISTIC` to enable notifications	+WRITE:0,1,2,1,2, 0x0100

6. On the Proteus device, send "Hello Stephano-I" to receive it on the Stephano-I side. See `CMD_DATA_REQ` command in Proteus user manual [8, 9, 11, 10]:

Info	Message
⇐ Received write on the `RX_CHARACTERISTIC` with payload 0x01 Hello Stephano-I	+WRITE:0,1,1„17,0x01 Hello Stephano-I

7. Then transmit "Hello Proteus" to the Proteus device by writing the data to the

RX_CHARACTERISTIC (char. index 2). Here the header byte 0x01 must be prefixed:

Info	Message
⇒ Notify to the RX_CHARACTERISTIC	AT+BLEGATTSNTFY=0,1,2,14
⇐ Stephano-I is ready to receive data via UART	>
⇒ Transmit "Hello Proteus" with a prefixed 0x01 (WE SPP Header)	0x01 Hello Proteus
⇐	OK

8. (Optional) Use the passthrough mode. The passthrough mode allows to bind certain characteristics to the UART interface. Thus, when the passthrough mode has been enabled, data sent via UART to the Stephano-I is forwarded to the bound outgoing characteristic. Furthermore, data sent via Bluetooth® LE to the bound incoming characteristics is forwarded to the UART. Thus the passthrough mode acts like a transparent bridge between UART and Bluetooth® LE interface.

! Data sent in passthrough mode does not need to be attached by a "\r\n".

a) Configure and start the passthrough mode. Configuration is done by binding the TX_CHARACTERISTIC (srv. index 1, char. index 1) and RX_CHARACTERISTIC (srv. index 1, char. index 2) with the UART interface:

Info	Message
⇒ Configure the passthrough mode	AT+BLESPPCFG=1,1,2,1,1,0
⇐	OK
⇒ Start the passthrough mode	AT+BLESPP
⇐	OK >

b) Now transparent data can be send to the Stephano-I for Bluetooth® LE transmission. Here we choose "Hey Proteus":

Info	Message
⇒ Transmit "Hey Proteus" with a prefixed 0x01 (WE SPP Header)	0x01 Hey Proteus

c) Data is received on the smart phone. Smart phone replies with "Hey Stephano-I":

Info	Message

⇐ "Hey Stephano-I" with a prefixed 0x01 (WE SPP Header) is received by the Stephano-I	0x01 Hey Stephano-I

d) Stop passthrough mode:

Info	Message
⇒ Stop the passthrough mode by sending "+++"	+++

! The "+++" must be send without "\r\n". The "+++" is not confirmed by an "OK".

9. The data has been received on the Proteus device. Now the connection can be closed if no further data is supposed to be transmitted:

Info	Message
⇒ Disconnect on connection index 0	AT+BLEDISCONN=0
⇐	OK

10.1.3 Peripheral: Receive connection from smart phone using WE SPP-like profile

In this chapter the goal is to configure the Stephano-I as Bluetooth® LE peripheral and use the integrated WE SPP-like Bluetooth® LE profile. Then connect from a smart phone acting as central to the Stephano-I for bidirectional data transmission. On the smart phone the **nRF Connect** App [4, 5] is used, that provides all Bluetooth® LE related functions.
To do that, the services, characteristics and the corresponding UUIDs of the "WE SPPlike" profile must be known. Please check out chapter 5.2 for that.
In this example, the central is the smart phone, and the peripheral is the Stephano-I.

In the following example only the header byte is written in hex notation. All other data is ascii notation.

1. First of all reset the device by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info	Message
⇐ Module is ready for operation	ready

2. Then enable the Bluetooth® LE peripheral function of the Stephano-I:

Info	Message
⇒ Enable the peripheral function	AT+BLEINIT=2
⇐	OK

3. Start service and check what is available on the Stephano-I:

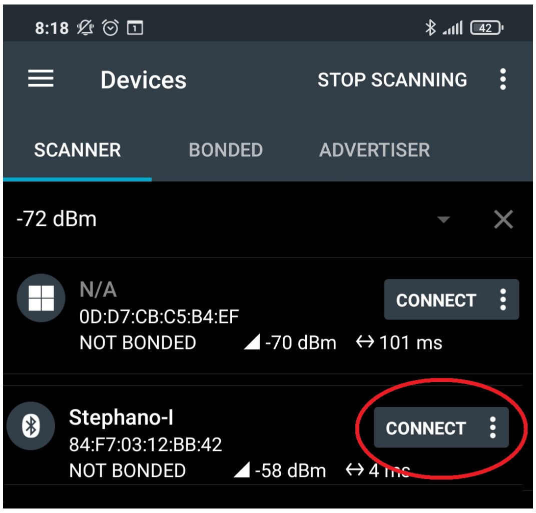
Info	Message
⇒ Create the local services	AT+BLEGATTSSRVCRE
⇐	OK
⇒ Start the local services	AT+BLEGATTSSRVSTART
⇐	OK
⇒ Request which services are available	AT+BLEGATTSSRV?
⇐ WE SPP-like primary service is available on service index 1	+BLEGATTSSRV:1,1, 0x6E400001C35211E5953D0002A5D5C51B,1 OK
⇒ Request which characteristic are available	AT+BLEGATTSCHAR?

⇐ WE SPP-like `RX_CHARACTERISTIC` is available on characteristic index 1	+BLEGATTSCHAR:"char",1,1, 0x6E400002C35211E5953D0002A5D5C51B, 0x0c
⇐ WE SPP-like `TX_CHARACTERISTIC` is available on characteristic index 2	+BLEGATTSCHAR:"char",1,2, 0x6E400003C35211E5953D0002A5D5C51B, 0x10
⇐ Descriptor with index 1 is available for characteristic index 2	+BLEGATTSCHAR:"desc",1,2,1,0x2902 OK

4. The next step is to set the advertising data such that the Stephano-I can be detected by the smart phone. Therefore at least the advertising flags and the name (here we choose "Stephano-I") should be placed into the advertising packet:

Info	Message
⇒ Set the advertising data	AT+BLEADVDATA="020106 0B095374657068616E6F2D49"
⇐	OK
⇒ Start advertising	AT+BLEADVSTART
⇐	OK

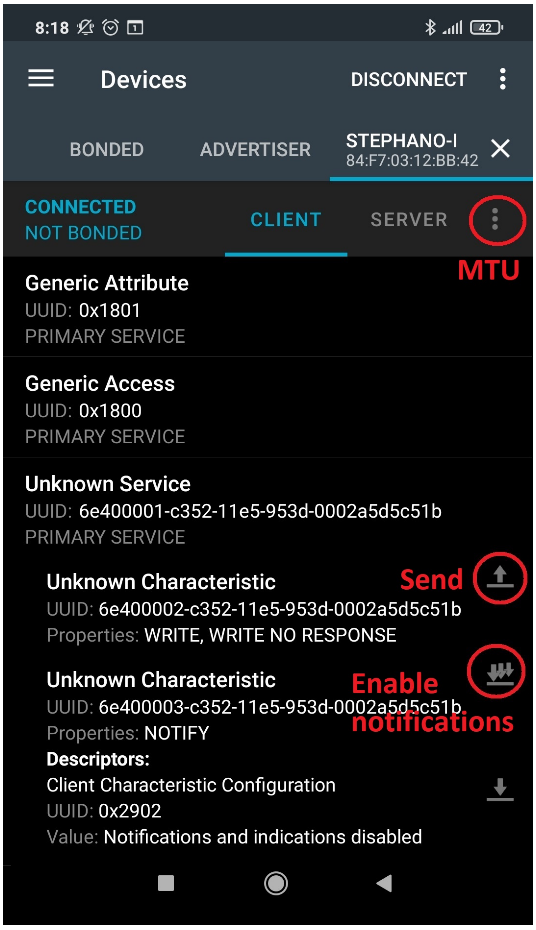
5. On the smart phone side, scan for the advertising Stephano-I and connect to that.



Due to that the Stephano-I outputs the following:

Info	Message
⇐ Connection has been established on connection index 0 with connection interval of 45 ms	+BLECONN:0,"00:18:da:00:00:05" +BLECONNPARAM:0,0,0,36,0,500

6. On the smart phone side, enable the notifications of the `TX_CHARACTERISTIC`.



Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

Generic Access
UUID: 0x1800
PRIMARY SERVICE

Unknown Service
UUID: 6e400001-c352-11e5-953d-0002a5d5c51b
PRIMARY SERVICE

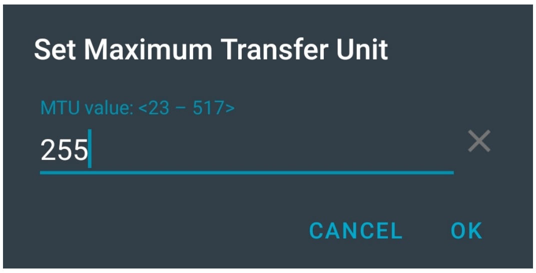
Unknown Characteristic
UUID: 6e400002-c352-11e5-953d-0002a5d5c51b
Properties: WRITE, WRITE NO RESPONSE

Unknown Characteristic
UUID: 6e400003-c352-11e5-953d-0002a5d5c51b
Properties: NOTIFY
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Value: Notifications and indications disabled

Due to that the Stephano-I outputs the following:

Info	Message
⇐ 0x0100 has been written to the descriptor of the `TX_CHARACTERISTIC` to enable notifications	+WRITE:0,1,2,1,2, 0x0100

7. On the smart phone side, set the MTU to 255.



Set Maximum Transfer Unit

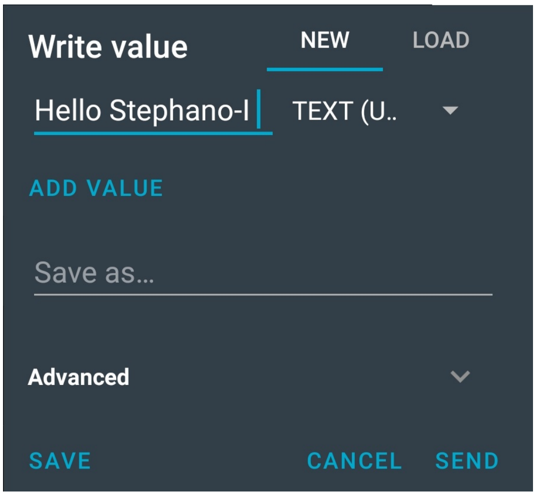
MTU value: <23 – 517>

255

Due to that the Stephano-I outputs the following:

Info	Message
⇐ MTU has been set to 255	+BLECFGMTU:0,255

8. On the smart phone side, send "Hello Stephano-I" to receive it on the Stephano-I side.



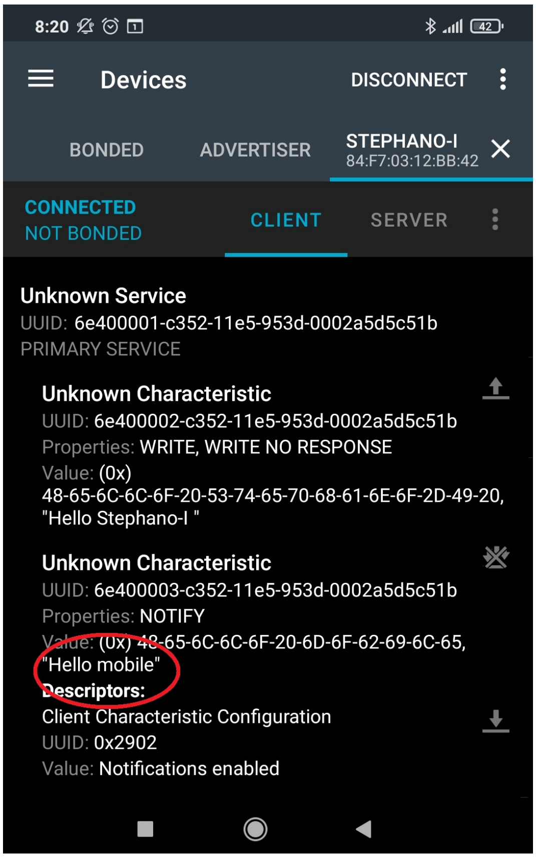
Due to that the Stephano-I outputs the following:

Info	Message
⇐ Received write on the `RX_CHARACTERISTIC` with payload "Hello Stephano-I"	+WRITE:0,1,1,,16,Hello Stephano-I

9. Then transmit "Hello mobile" to the smart phone by writing the data to the `RX_CHARACTERISTIC` (char. index 2):

Info	Message
⇒ Notify to the `RX_CHARACTERISTIC`	AT+BLEGATTSNTFY=0,1,2,12
⇐ Stephano-I is ready to receive data via UART	>
⇒ Transmit "Hello mobile"	Hello mobile
⇐	OK

10. The data has been received on the smart phone device:



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)
48-65-6C-6C-6F-20-53-74-65-70-68-61-6E-6F-2D-49-20,
"Hello Stephano-I "

Unknown Characteristic
UUID: 6e400003-c352-11e5-953d-0002a5d5c51b
Properties: NOTIFY
Value: (0x) 48-65-6C-6C-6F-20-6D-6F-62-69-6C-65,
"Hello mobile"
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Value: Notifications enabled

11. (Optional) Use the passthrough mode. The passthrough mode allows to bind certain characteristics to the UART interface. Thus, when the passthrough mode has been enabled, data sent via UART to the Stephano-I is forwarded to the bound outgoing characteristic. Furthermore, data sent via Bluetooth® LE to the bound incoming characteristics is forwarded to the UART. Thus the passthrough mode acts like a transparent bridge between UART and Bluetooth® LE interface.

> Data sent in passthrough mode does not need to be attached by a "\r\n".

a) Configure and start the passthrough mode. Configuration is done by binding the `TX_CHARACTERISTIC` (srv. index 1, char. index 1) and `RX_CHARACTERISTIC` (srv. index 1, char. index 2) with the UART interface:

Info	Message
⇒ Configure the passthrough mode	AT+BLESPPCFG=1,1,2,1,1,0
⇐	OK
⇒ Start the passthrough mode	AT+BLESPP
⇐	OK >

b) Now transparent data can be send to the Stephano-I for Bluetooth® LE transmission. Here we choose "Hey mobile":

Info	Message
⇒ Transmit "Hey mobile"	Hey mobile

c) Data is received on the smart phone. Smart phone replies with "Hey Stephano-I":

Info	Message
⇐ "Hey Stephano-I" is received by the Stephano-I	Hey Stephano-I

d) Stop passthrough mode:

Info	Message
⇒ Stop the passthrough mode by sending "+++"	+++

! The "+++" must be send without "\r\n". The "+++" is not confirmed by an "OK".

12. Now the connection can be closed if no further data is supposed to be transmitted:

Info	Message
⇒ Disconnect on connection index 0	AT+BLEDISCONN=0
⇐	OK

10.1.4 Central and peripheral: Establish connection between two Stephano-I using WE SPP-like profile

In this chapter the goal is to configure one Stephano-I as Bluetooth® LE peripheral and a second Stephano-I as central device. Then setup a connection between both for bidirectional data transmission.
To do that, the services, characteristics and the corresponding UUIDs of the "WE SPPlike" profile must be known. Please check out chapter 5.2 for that.

> For simplicity reasons, in this chapter the Stephano-I acting a central is called CENTRAL and the Stephano-I acting as peripheral is called PERIPHERAL.

Setup the PERIPHERAL

1. First of all reset the PERIPHERAL by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info - PERIPHERAL	Message
⇐ Module is ready for operation	ready

2. Then enable the Bluetooth® LE peripheral function of the PERIPHERAL:

Info - PERIPHERAL	Message
⇒ Enable the peripheral function	AT+BLEINIT=2
⇐	OK

3. Start service and check what is available on the PERIPHERAL:

Info - PERIPHERAL	Message
⇒ Create the local services	AT+BLEGATTSSRVCRE
⇐	OK
⇒ Start the local services	AT+BLEGATTSSRVSTART
⇐	OK
⇒ Request which services are available	AT+BLEGATTSSRV?
⇐ WE SPP-like primary service is available on service index 1	+BLEGATTSSRV:1,1, 0x6E400001C35211E5953D0002A5D5C51B,1 OK
⇒ Request which characteristic are available	AT+BLEGATTSCHAR?

⇐ WE SPP-like `RX_CHARACTERISTIC` is available on characteristic index 1	+BLEGATTSCHAR:"char",1,1, 0x6E400002C35211E5953D0002A5D5C51B, 0x0c
⇐ WE SPP-like `TX_CHARACTERISTIC` is available on characteristic index 2	+BLEGATTSCHAR:"char",1,2, 0x6E400003C35211E5953D0002A5D5C51B, 0x10
⇐ Descriptor with index 1 is available for characteristic index 2	+BLEGATTSCHAR:"desc",1,2,1,0x2902 OK

4. The next step is to set the advertising data such that the Stephano-I can be detected on the radio by the CENTRAL. Therefore at least the advertising flags and the name (here we choose "Stephano-I") should be placed into the advertising packet:

Info - PERIPHERAL	Message
⇒ Set the advertising data	AT+BLEADVDATA="020106 0B095374657068616E6F2D49"
⇐	OK
⇒ Start advertising	AT+BLEADVSTART
⇐	OK

5. (Optional) Configure the input output capabilities to enable radio security (pairing/bonding):

Info - PERIPHERAL	Message
⇒ Set security level to SC_MITM_BOND, the IO capabilities to "display only" and keylength of 16 bytes	AT+BLESECPARAM=13,0,16,3,3,0
⇐	OK

Setup the CENTRAL

1. Then reset the CENTRAL by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info - CENTRAL	Message
⇐ Module is ready for operation	ready

2. (Optional) Configure the input output capabilities to enable radio security (pairing/bonding):

Info - CENTRAL	Message

⇒ Set security level to SC_MITM_BOND, the IO capabilities to "keyboard only" and keylength of 16 bytes	AT+BLESECPARAM=13,2,16,3,3,0
⇐	OK

3. Then enable the Bluetooth® LE central function of the CENTRAL:

Info - CENTRAL	Message
⇒ Enable the central function	AT+BLEINIT=1
⇐ Module is ready for operation	OK

4. The next step is to scan the radio to find available radio modules:

Info - CENTRAL	Message
⇒ Scan for 2 s to find available devices	AT+BLESCAN=1,2
⇐ Found Bluetooth® LE device with MAC "00:18:da:00:00:05" and device name "Stephano-I"	OK +BLESCAN:"00:18:da:00:00:05", -50, 020106 0B095374657068616E6F2D49,,0

Connect the CENTRAL to the PERIPHERAL

1. Now setup the physical connection:

Info - CENTRAL	Message
⇒ Connect to BLE device with MAC 00:18:da:00:00:05 on connection index 0	AT+BLECONN=0,"00:18:da:00:00:05"
⇐ Connection has been setup	+BLECONN:0,"00:18:da:00:00:05" OK

Info - PERIPHERAL	Message
⇐ Connection has been established on connection index 0	+BLECONN:0,"00:18:da:00:00:06"

2. (Only if security has been enabled before) Now start the pairing and bonding:

Info - CENTRAL	Message
⇒ Start the pairing/bonding	AT+BLEENC=0,1
⇐	OK

Info - PERIPHERAL	Message
⇐ Pairing has been requested	+BLESECREQ:0"
⇒ Accept request	AT+BLEENCRSP=0,1
⇐ Shows pairing key	OK +BLEENCNTFYKEY=0,172512

Info - CENTRAL	Message
⇐ Request to enter the key	+BLESECKEYREQ:0
⇒ Enter the key	AT+BLEKEYREPLY=0,172512
⇐Pairing has been completed	OK +BLESECKEYTYPE:0,32 +BLEAUTHCMPL:0,0

Info - PERIPHERAL	Message
⇐Pairing has been completed	+BLESECKEYTYPE:0,32 +BLEAUTHCMPL:0,0

3. Now the CENTRAL must discover the services of the PERIPHERAL:

Info - CENTRAL	Message
⇒ Discover services	AT+BLEGATTCPRIMSRV=0
⇐ Found service "Generic Access" and "Generic Attribute"	+BLEGATTCPRIMSRV:0,1,0x1800,1 +BLEGATTCPRIMSRV:0,2,0x1801,1
⇐ Found WE SPP-like primaary service with service index 3	+BLEGATTCPRIMSRV:0,3, 0x6E400001C35211E5953D0002A5D5C51B,1 OK

4. Now discover the characteristic of the WE SPP-like primary services to see what characteristics it provides:

Info - CENTRAL	Message
⇒ Discover characteristics of service index 3	AT+BLEGATTCCHAR=0,3
⇐ Found WE SPP-like RX_CHARACTERISTIC characteristic with characteristics index 1	+BLEGATTCCHAR:"char",0,3,1, 0x6E400002C35211E5953D0002 A5D5C51B,0x0c
⇐ Found WE SPP-like TX_CHARACTERISTIC characteristic with characteristics index 2	+BLEGATTCCHAR:"char",0,3,2, 0x6E400003C35211E5953D0002 A5D5C51B,0x10

⇐ Found descriptor with descriptor index 1 for characteristic index 2	+BLEGATTCCHAR:"desc",0,3,2,1,0x2902 OK

5. To be able to receive data from the PERIPHERAL, the notification of the `TX_CHARACTERISTIC` must be enabled:

Info - CENTRAL	Message
⇒ Enable notification by writing 0x0001 (2 bytes) to the descriptor of `TX_CHARACTERISTIC` (char. index 2)	AT+BLEGATTCWR=0,3,2,1,2
⇐ Stephano-I is ready to receive data via UART	>
⇒Now send the data	0x01 0x00 (hex, no ascii)
⇐	OK

Info - PERIPHERAL	Message
⇐ 0x0100 has been written to the descriptor of the `TX_CHARACTERISTIC` to enable notifications	+WRITE:0,1,2,1,2, 0x0100

Transmit data

1. On the CENTRAL side, send "Hello" by writing the data to the `RX_CHARACTERISTIC` (char. index 1):

Info - CENTRAL	Message
⇒ Write to the `RX_CHARACTERISTIC`	AT+BLEGATTCWR=0,3,1„5
⇐ Stephano-I is ready to receive data via UART	>
⇒ Transmit "Hello"	Hello
⇐	OK

Due to that the PERIPHERAL outputs the following:

Info - PERIPHERAL	Message
⇐ Received write on the `RX_CHARACTERISTIC` with payload "Hello"	+WRITE:0,1,1„5,Hello

2. Then transmit "Hello too" from the PERIPHERAL to the CENTRAL by writing the data to the `RX_CHARACTERISTIC` (char. index 2):

Info - PERIPHERAL	Message
⇒ Notify to the `RX_CHARACTERISTIC`	AT+BLEGATTSNTFY=0,1,2,9
⇐ Stephano-I is ready to receive data via UART	>
⇒ Transmit "Hello too"	Hello too
⇐	OK

Info - CENTRAL	Message
⇐ Received a notification on the `TX_CHARACTERISTIC` with payload "Hello too"	+NOTIFY:0,3,2,9,Hello too

(Optional) Use the passthrough mode

The passthrough mode allows to bind certain characteristics to the UART interface. Thus, when the passthrough mode has been enabled, data sent via UART to the Stephano-I is forwarded to the bound outgoing characteristic. Furthermore, data sent via Bluetooth® LE to the bound incoming characteristics is forwarded to the UART. Thus the passthrough mode acts like a transparent bridge between UART and Bluetooth® LE interface.

Data sent in passthrough mode does not need to be attached by a "\r\n".

1. Configure and start the passthrough mode for the CENTRAL. Configuration is done by binding the `TX_CHARACTERISTIC` (srv. index 3, char. index 1) and `RX_CHARACTERISTIC` (srv. index 3, char. index 2) with the UART interface:

Info - CENTRAL	Message
⇒ Configure the passthrough mode	AT+BLESPPCFG=1,3,1,3,2,0
⇐	OK
⇒ Start the passthrough mode	AT+BLESPP
⇐	OK >

2. Configure and start the passthrough mode for the CENTRAL. Configuration is done by binding the `TX_CHARACTERISTIC` (srv. index 1, char. index 1) and `RX_CHARACTERISTIC` (srv. index 1, char. index 2) with the UART interface:

Info - PERIPHERAL	Message
⇒ Configure the passthrough mode	AT+BLESPPCFG=1,1,2,1,1,0

⇐	OK
⇒ Start the passthrough mode	AT+BLESPP
⇐	OK >

3. Now transparent data can be send to the CENTRAL for Bluetooth® LE transmission. Here we choose "Hey there":

Info - CENTRAL	Message
⇒ Transmit "Hey there"	Hey there

4. Data is received on the PERIPHERAL, and it replies with "Hey!":

Info - PERIPHERAL	Message
⇐ "Hey there" is received by the PERIPHERAL	Hey there
⇒ "Hey!" is transmitted to the CENTRAL	Hey!

Info - CENTRAL	Message
⇐ "Hey!" is received	Hey!

5. Stop passthrough mode on the CENTRAL and PERIPHERAL:

Info - CENTRAL	Message
⇒ Stop the passthrough mode by sending "+++"	+++

Info - PERIPHERAL	Message
⇒ Stop the passthrough mode by sending "+++"	+++

The "+++" must be send without "\r\n". The "+++" is not confirmed by an "OK".

Close the connection

1. Now the connection can be closed if no further data is supposed to be transmitted:

Info - CENTRAL	Message
⇒ Disconnect on connection index 0	AT+BLEDISCONN=0
⇐	OK

Info - PERIPHERAL	Message
⇐ Connected has been closed	+BLEDISCONN:0,"00:18:da:00:00:06"

10.2 Wifi examples

10.2.1 Create a TCP socket: Establish connection between two Stephano-I as TCP Server and Client

In this chapter the goal is to configure TCP server in one Stephano-I and create a connection to the server with another Stephano-I as a client.

> For simplicity reasons, in this chapter the Stephano-I acting a server is called SERVER and the Stephano-I acting as client is called CLIENT.

Setup the TCP Server

1. First of all reset the SERVER by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info - SERVER	Message
⇐ Module is ready for operation	ready

2. Configure the SERVER as softAP

Info - SERVER	Message
⇒ Set the Wi-Fi mode to softAP	AT+CWMODE=2
⇐	OK
⇒ Enable multiple connections	AT+CIPMUX=1
⇐	OK
⇒ Set the configuration of the SoftAP	AT+CWSAP="Stephano1_softAP","1234567890",5,3
⇐	OK

3. Setup the TCP Server in the SERVER Module:

Info - SERVER	Message
⇒ Create the TCP Server on port 8089	AT+CIPSERVER=1,8089
⇐	OK

Setup the CLIENT

1. Reset the CLIENT by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. CLIENT will reply

with a start-up message "ready":

Info - CLIENT	Message
⇐ Module is ready for operation	ready

2. Configure the CLIENT as station and connect it to the softAP of the SERVER module:

Info - CLIENT	Message
⇒ Set the Wi-Fi mode to station	AT+CWMODE=1
⇐	OK
⇒ Connect to the SoftAP	AT+CWJAP="Stephano1_softAP","1234567890"
⇐	WIFI CONNECTED WIFI GOT IP OK

3. The next step is to connect the CLIENT Module to SERVER Module as a client over TCP:

Info - CLIENT	Message
⇒ Establish the TCP Socket by connecting to the SERVER	AT+CIPSTART="TCP","192.168.3.102",8089
⇐	CONNECT OK

Transmit data

1. Send data from the SERVER to the CLIENT:

Info - SERVER	Message
⇒ Send 4 bytes of data	AT+CIPSEND=0,4
⇐ SERVER is ready to send data	>
⇒ Transmit "Hello"	Hello
⇐	Recv 4 bytes SEND OK

Info - CLIENT	Message
⇐	+IPD,4:Hello

2. Send data from the CLIENT to the SERVER:

Info - CLIENT	Message
⇒ Send 4 bytes of data	AT+CIPSEND=4

⇐ CLIENT is ready to send data	>
⇒ Transmit "Hello"	Hello
⇐	Recv 4 bytes SEND OK

Info - SERVER	Message
⇐	+IPD,4:Hello

Close the connection

1. Now the socket can be closed if no further data is supposed to be transmitted:

Info - CLIENT	Message
⇒ Disconnect from the server	AT+CIPCLOSE
⇐	CLOSED OK

2. TCP Server can be deleted on the SERVER side:

Info - SERVER	Message
⇒ Delete the server	AT+CIPSERVER=0,1
⇐	OK

10.2.2 MQTT Example: Establish connection from Stephano-I to AWS IoT with MQTT AT commands

In this chapter the goal is to connect the Stephano-I to AWS IoT.

Gaining and uploading the certificates

You need to obtain the device.pem.crt (device certificate), private.pem.key (private key) and Amazon-root-CA-1.pem (root CA certificate) from AWS as a prerequisite as the first step. They must be saved in the module using AT commands:

- Amazon-root-CA-1.pem as mqtt_ca.cert
- device.pem.crt as mqtt_client.crt
- private.pem.key as mqtt_client.key

Please refer to chapter 10.3.1 to do that.

! Please note that each line of the certificate must end with a "\r\n".

MQTT Configuration

1. Use MQTT AT commands to connect to AWS IoT:

Info	Message
⇒ Set the Wi-Fi mode to station	AT+CWMODE=1
⇐	OK
⇒ Connect to the access point	AT+CWJAP=<ssid>,<password>
⇐	OK
⇒ Set the SNTP server	AT+CIPSNTPCFG=1,2,"0.de.pool.ntp.org"
⇐	OK
⇒ Set the MQTT user configuration	AT+MQTTUSERCFG=0,5,"stephano1", "stephano1_aws","1234567890",0,0,""
⇐	OK
⇒ Setup connection to AWS IoT	AT+MQTTCONN=0,"<endpoint>",8883,1
⇐	+MQTTCONNECTED:0,5,<endpoint>,"8883","",1 OK
⇒ Subscribe to MQTT topic	AT+MQTTSUB=0,"topic/stephano1",1
⇐	OK
⇒ Publish message to an MQTT topic	AT+MQTTPUB=0,"topic/stephano1","hello aws!",1,0

⇐	+MQTTSUBRECV:0,"topic/stephano1",10,hello aws! OK

Close the MQTT connection

1. Now the MQTT connection can be closed if no further data is supposed to be transmitted:

Info	Message
⇒ Unsubscribe from the defined topic	AT+MQTTUNSUB=0,"topic/stephano1"
⇐	OK
⇒ Disconnect on connection index 0	AT+MQTTCLEAN=0
⇐	OK

10.2.3 SSL Example: Establish connection from a test PC as client to Stephano-I as server using mutual authentication

In this chapter the goal is to configure a SSL server in one Stephano-I and create a connection from a PC as a client. The certificates used in this example are from Espressif. You can flash your own server certificates as explained in the chapter `10.3.1`.

> For simplicity reasons, in this chapter the Stephano-I acting a server is called SERVER and the test computer is mentioned as PC.

Setup the SSL Server

1. First of all reset the SERVER by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info - SERVER	Message
⇐ Module is ready for operation	ready

2. Configure the SERVER as a station:

Info - SERVER	Message
⇒ Set the Wi-Fi mode to station	AT+CWMODE=1
⇐	OK
⇒ Enable multiple connections	AT+CIPMUX=1
⇐	OK
⇒ Connect to the router	AT+CWJAP="<ssid>","<pwd>"
⇐	"WIFI CONNECTED WIFI GOT IP OK"

3. Setup the SSL Server in the SERVER Module:

Info - SERVER	Message
⇒ Create the SSL Server on port 8070	AT+CIPSERVER=1,8070,"SSL",1
⇐	OK

Configure the PC to start a client request

1. Tools like OpenSSL can be used on PC to create an SSL client request to the server that SERVER has created. The ca, cert and key files used here are obtained from *Espressive's site* on Github. You can use your own client certificates files if you have already flashed the server certificates using SYSMFG command. Add the IP address of the SERVER as assigned by the router in the below command to make an SSL connection request to the SERVER.

```
openssl s_client -CAfile client_ca_00.crt -cert client_cert_00.crt -key
    client_key_00.key -host <ip address> -port 8070
```

Info - SERVER	Message
⇐ SERVER response when a connection is established	0,CONNECT

Transmit data

1. Send data from the SERVER to the CLIENT:

Info - SERVER	Message
⇒ Send 4 bytes of data	AT+CIPSEND=0,4
⇐ SERVER is ready to send data	>
⇒ Transmit "Hello"	Hello
⇐	Recv 4 bytes SEND OK

2. Send 4 bytes of data "Hello" from the PC to the SERVER:

Info - SERVER	Message
⇐	+IPD,4:Hello

Close the connection and delete the server

1. Now the socket can be closed from PC if no further data is supposed to be transmitted:

Info - SERVER	Message
⇐	0,CLOSED OK
⇒ Delete the server	AT+CIPSERVER=0,1
⇐	OK

Eventhough we made only one SSL connection in this example, multiple connections are possible as we have enabled multiple connections in the SERVER.

10.2.4 Wi-Fi provisioning: Configure the Wi-Fi network by remote access

When running the Stephano-I in a new environment for the first time, it must know the credentials like SSID and password of the local Wi-Fi network, first.
To do this via remote access, the so called provisioning feature is used. In this case the Stephano-I is configured as access point with web server, first. Then another device, like a smart phone, connects to it and opens the provided web page in its browser. Then the user can enter the credentials of the local Wi-Fi network. In case the Stephano-I can connect to the local Wi-Fi using the entered data, the provisioning is finished.
In this chapter we will describe that feature.

Setup the access point and web server

1. First of all reset the Stephano-I by applying a high-low-high sequence to the */RESET* pin. This can be done on the Stephano-I evaluation board by pressing the reset button. The module will reply with a start-up message "ready":

Info	Message
⇐ Module is ready for operation	ready

2. Configure the Stephano-I as softAP + station:

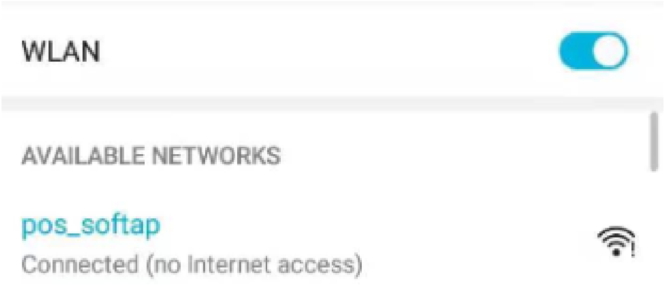
Info	Message
⇒ Set the Wi-Fi mode to soft AP and station	AT+CWMODE=3
⇐	OK
⇒ Enable multiple connections	AT+CIPMUX=1
⇐	OK
⇒ Set the configuration of the SoftAP	AT+CWSAP="pos_softap","",11,0,3
⇐	OK

3. Now start the included web server:

Info	Message
⇒ Start the web server on port 80	AT+WEBSERVER=1,80,25
⇐	OK

Use a browser to enter the credentials of the local Wi-Fi network

1. The Stephano-I acts as an access point, now. Thus a smart phone or PC can use the configured credentials (SSID = "pos_softap", empty pass word) to connect to it.



WLAN

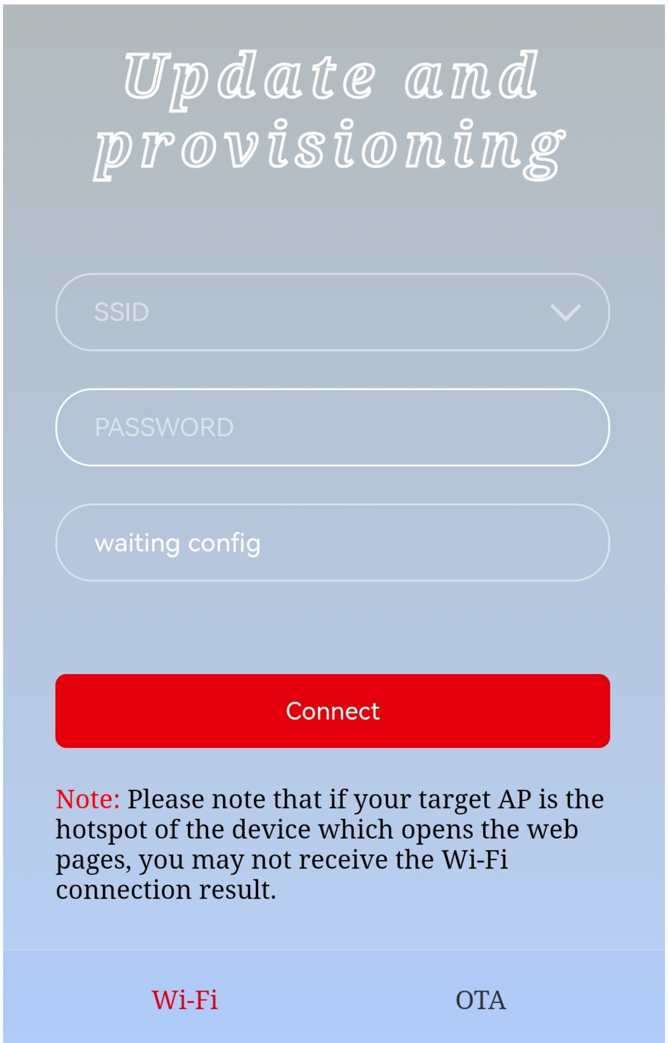
AVAILABLE NETWORKS

pos_softap
Connected (no Internet access)

Figure 8: Connect to access point

2. Now open a browser on the remote device and enter the IP of the Stephano-I. The default IP is 192.168.4.1, or you can query the current SoftAP IP address by command `AT+CIPAP?`.

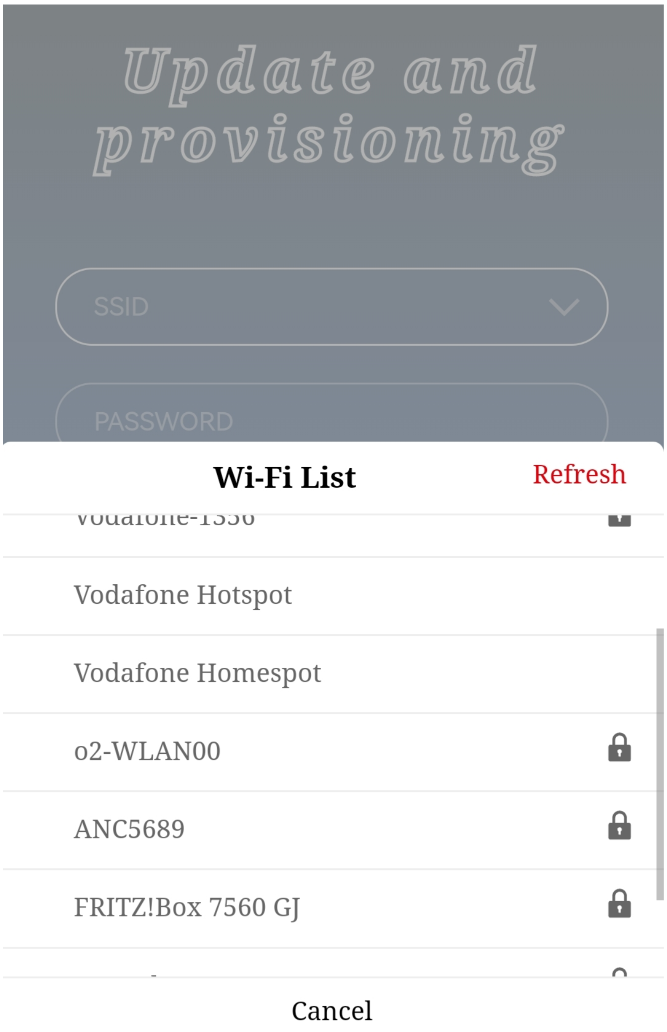
The provisioning has been tested with Chrome browser. In case of failure, set the developer settings to "http only".


Update and provisioning

Note: Please note that if your target AP is the hotspot of the device which opens the web pages, you may not receive the Wi-Fi connection result.

Figure 9: Open web page for provisioning

A web page opens with WIFI tab selected as default. where the credentials of the local Wi-Fi network can be entered. For simplicity reason, the Stephano-I can scan for all available Wi-Fi networks, by pressing the arrow in the drop-down menu.



Update and provisioning

SSID

PASSWORD

Wi-Fi List

Refresh

Vodafone-1356

Vodafone Hotspot

Vodafone Homespot

o2-WLAN00

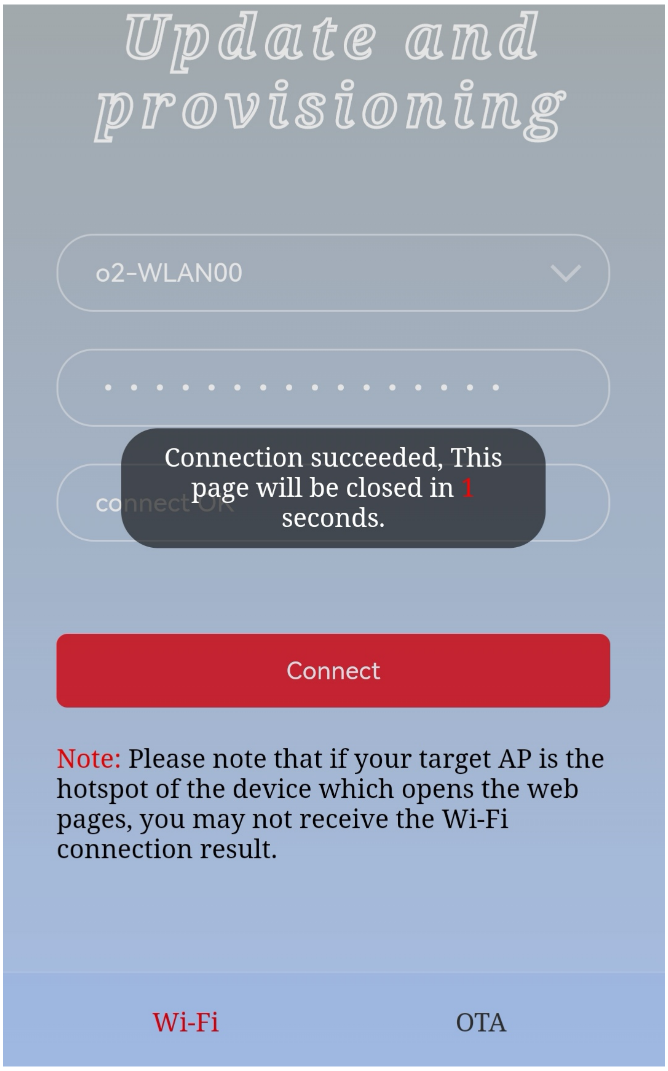
ANC5689

FRITZ!Box 7560 GJ

Cancel

Figure 10: Scan for local networks

3. After entering the credentials(SSID and password) of the local network, the Stephano-I tries to connect to that. In case of success, the web page closes.



Update and provisioning

o2-WLAN00

• • • • • • • • • • • • • • • •

connect OK

Connection succeeded, This page will be closed in 1 seconds.

Note: Please note that if your target AP is the hotspot of the device which opens the web pages, you may not receive the Wi-Fi connection result.

Figure 11: Successfully connected to the local Wi-Fi

At the same time the Stephano-I sends the following messages to the connected host via UART:

Info	Message
⇐ The module has received Wi-Fi connection information	+WEBSERVERRSP:1
⇐ The module has connected to the new network	WIFI CONNECTED
⇐ The module has received an IP	WIFI GOT IP
⇐ The provisioning has finished with success	+WEBSERVERRSP:2

! In case of wrong credentials the `+WEBSERVERRSP:2` message is not printed.

10.2.5 OTA via Wi-Fi: Update of the module firmware or manufacturing user partition over the air

The Stephano-I allows a firmware over the air (FOTA) update. To do this a remote device like a smart phone has to connect to the Stephano-I and open a local web page in its browser. With help of that the new firmware image can be transmitted and applied to the radio module.
This process is described in this chapter.

> ! The steps to upload the manufacturing user partition via web server are the same. To update single certificates or keys, please refer to chapter `10.3.1`.

1. First of all, ensure that the Stephano-I and the device that runs the update, i.e. smart phone or PC, are in the same network:
   a) In case the Stephano-I acts as access point, connect to that.
   b) Or in case the Stephano-I is in station mode, connect to the same network, i.e. access point.

> ! We recommend to use the method, where the Stephano-I does not act as access point to reduce the work load on the Stephano-I and increase update stability.

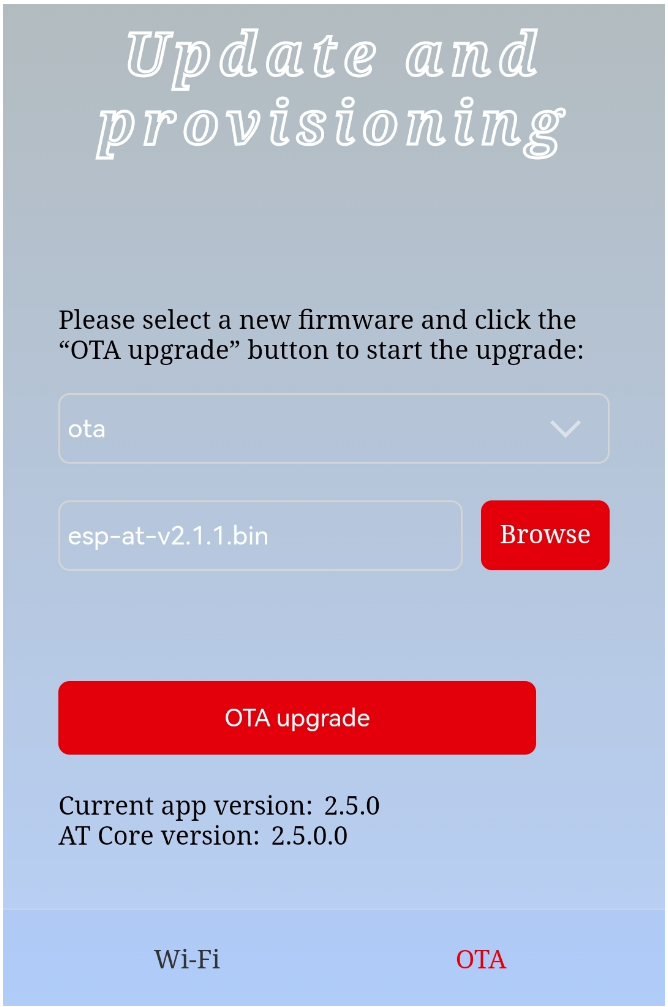
2. Then set the Stephano-I to update mode by starting the included web server:

Info	Message
⇒ Start the web server on port 80	AT+WEBSERVER=1,80,25
⇐	OK

3. Now open a browser on the smart phone or PC and enter the IP of the Stephano-I. The default IP is 192.168.4.1, or you can query the current SoftAP IP address by command `AT+CIPAP?`.

> ! The FOTA has been tested with Chrome browser. In case of failure, set the developer settings to "http only".

4. A web page opens. Click on the OTA tab and select the option "ota" from the "Partition" list. For uploading certificate files, the user can select the corresponding certificate name from the list. In this example we are trying to update the module firmware hence ota is selected.



Update and provisioning

Please select a new firmware and click the "OTA upgrade" button to start the upgrade:

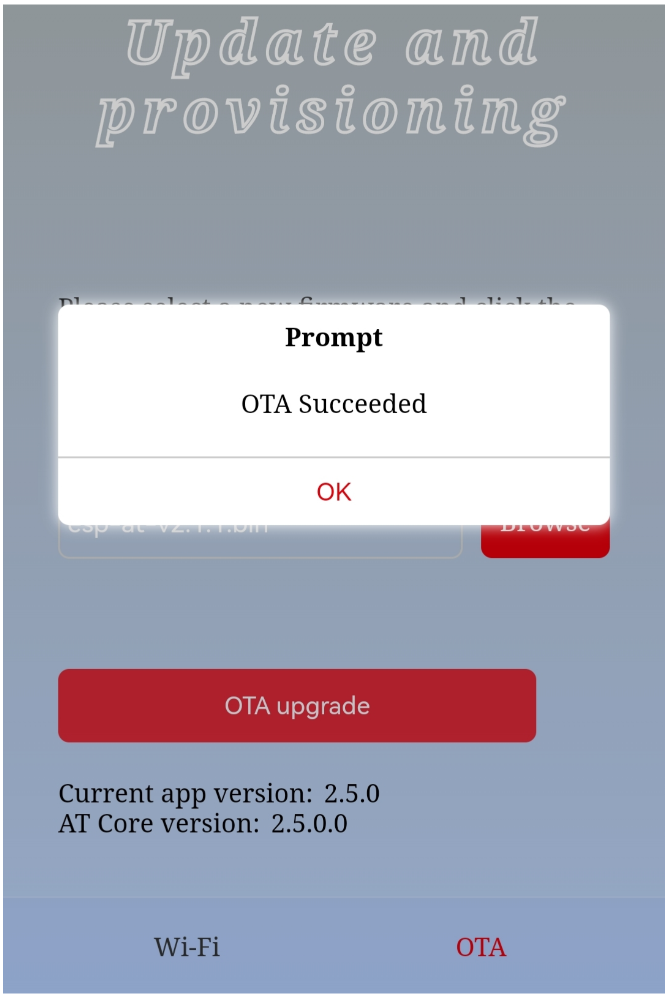
ota

esp-at-v2.1.1.bin

Current app version: 2.5.0
AT Core version: 2.5.0.0

Figure 12: Choose the firmware image

5. Then press the "OTA upgrade" button and wait for the success message.



Update and provisioning

Prompt

OTA Succeeded

Current app version: 2.5.0
AT Core version: 2.5.0.0

Figure 13: Successfully updated the radio module

At the same time the Stephano-I sends the following messages to the connected host via UART:

Info	Message
⇐ The transmission of the new firmware image has started	+WEBSERVERRSP:3
⇐ The new firmware image has been received successfully	+WEBSERVERRSP:4

After a restart of the module, the new firmware image is applied. Thus reset the Stephano-I.

In case of error, a `+WEBSERVERRSP:5` message is printed.

10.3 General examples

10.3.1 Update the certificates and keys for secure communication

To setup a secure communication, the certificates and keys must be updated periodically. The command `AT+SYSMFG` allows to read and write the files in the manufacturing user partition (MFG), which also contains the certificates. This example shows how to update the "mqtt_cert".

1. First of all, we read which files are available in the MFG:

Info - SERVER	Message
⇒ Read the available files	AT+SYSMFG?
⇐	+SYSMFG:"server_key" +SYSMFG:"server_ca" +SYSMFG:"client_cert" +SYSMFG:"client_key" +SYSMFG:"client_ca" +SYSMFG:"mqtt_cert" +SYSMFG:"mqtt_key" +SYSMFG:"mqtt_ca" +SYSMFG:"factory_param" +SYSMFG:"ble_data" OK

2. Now we are interested in updating the "mqtt_cert". Thus we read its key first:

Info - SERVER	Message
⇒ Read the key of the"mqtt_cert"	AT+SYSMFG=1,"mqtt_cert"
⇐ The key of the "mqtt_cert" is "mqtt_cert" and the format is binary (8)	+SYSMFG:"mqtt_cert","mqtt_cert",8 OK

3. Now the update can be run. The new certificate is a .crt-file in ascii format:

Info - SERVER	Message
⇒ Delete the certificate on the module first	AT+SYSMFG=0,"mqtt_cert","mqtt_cert""
⇐	OK
⇒ Write the new file (data length is 1174, format is binary (8))	AT+SYSMFG=2,"mqtt_cert","mqtt_cert",8,1174
⇐	OK >

⇒ Now send the content (1174 bytes)	- - - - -BEGIN CERTIFICATE- - - - -\r\n MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4 ikPmljZbyjANBgkqhkiG9w0BAQsF\r\n . . . - - - - -END CERTIFICATE- - - - -\r\n
⇐ Received the new file	OK

Please note that each line of the certificate must end with a "\r\n".

11 Firmware update

The Stephano-I provides firmware over the air (FOTA) update. To run an update of a single module, the Stephano-I must be set to update mode. In this case a web page that is hosted on the module can be accessed by any web browser. Using that, a new firmware image is loaded on the radio module.
Please find an example demonstrating this process in chapter `10.2.5`.

New firmware images, if existent, are available on the Stephano-I web page [13].

12 Firmware history

Version 3.2.0.0 "Release"

- Initial version using Espressif AT command firmware version 3.2.0.0 including Würth Elektronik eiSos bugfixes and performance improvements
- Known issues:
  - Command `AT+SYSTEMP?` does not work properly
  - Non-connectable advertising (non-connectable beacons) not supported

13 Design in guide

13.1 Advice for schematic and layout

For users with less RF experience it is advisable to closely copy the relating evaluation board with respect to schematic and layout, as it is a proven design. The layout should be conducted with particular care, because even small deficiencies could affect the radio performance and its range or even the conformity.
The following general advice should be taken into consideration:

- A clean, stable power supply is strongly recommended. Interference, especially oscillation can severely restrain range and conformity.
- Variations in voltage level should be avoided.
- LDOs, properly designed in, usually deliver a proper regulated voltage.
- Blocking capacitors and a ferrite bead in the power supply line can be included to filter and smoothen the supply voltage when necessary.

No fixed values can be recommended, as these depend on the circumstances of the application (main power source, interferences etc.).

STOP The use of an external reset IC should be considered if one of the following points is relevant:

- The slew rate of the power supply exceeds the electrical specifications.
- The effect of different current consumptions on the voltage level of batteries or voltage regulators should be considered. The module draws higher currents in certain scenarios like start-up or radio transmit which may lead to a voltage drop on the supply. A restart under such circumstances should be prevented by ensuring that the supply voltage does not drop below the minimum specifications.
- Voltage levels below the minimum recommended voltage level may lead to malfunction. The /Reset pin of the module shall be held on LOW logic level whenever the VCC is not stable or below the minimum operating Voltage.
- Special care must be taken in case of battery powered systems.

- Elements for ESD protection should be placed on all pins that are accessible from the outside and should be placed close to the accessible area. For example, the RF-pin is accessible when using an external antenna and should be protected.
- ESD protection for the antenna connection must be chosen such as to have a minimum effect on the RF signal. For example, a protection diode with low capacitance such as the 8231606A or a 68 nH air-core coil connecting the RF-line to ground give good results.
- Placeholders for optional antenna matching or additional filtering are recommended.
- The antenna path should be kept as short as possible.

Again, no fixed values can be recommended, as they depend on the influencing circumstances of the application (antenna, interferences etc.).

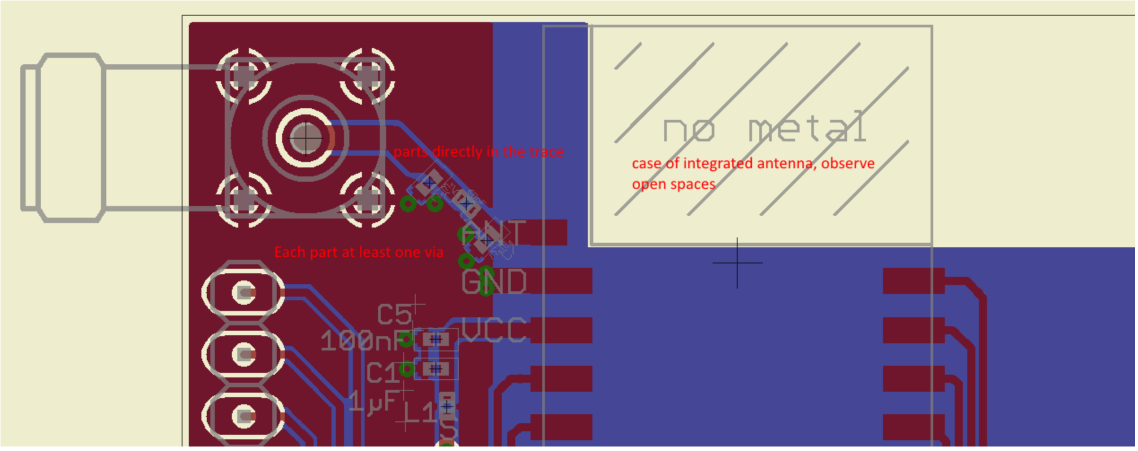

no metal
parts directly in the trace
case of integrated antenna, observe open spaces
ANT
Each part at least one via
GND
C5
100nF
VCC
C1
1µF
L1

Figure 14: Layout

- To avoid the risk of short circuits and interference there should be no routing underneath the module on the top layer of the baseboard.
- On the second layer, a ground plane is recommended, to provide good grounding and shielding to any following layers and application environment.
- In case of integrated antennas it is required to have areas free from ground. This area should be copied from the evaluation board.
- The area with the integrated antenna must overlap with the carrier board and should not protrude, as it is matched to sitting directly on top of a PCB.
- Modules with integrated antennas should be placed with the antenna at the edge of the main board. It should not be placed in the middle of the main board or far away from the edge. This is to avoid tracks beside the antenna.

- Filter and blocking capacitors should be placed directly in the tracks without stubs, to achieve the best effect.

- Antenna matching elements should be placed close to the antenna / connector, blocking capacitors close to the module.

- Ground connections for the module and the capacitors should be kept as short as possible and with at least one separate through hole connection to the ground layer.

- ESD protection elements should be placed as close as possible to the exposed areas.

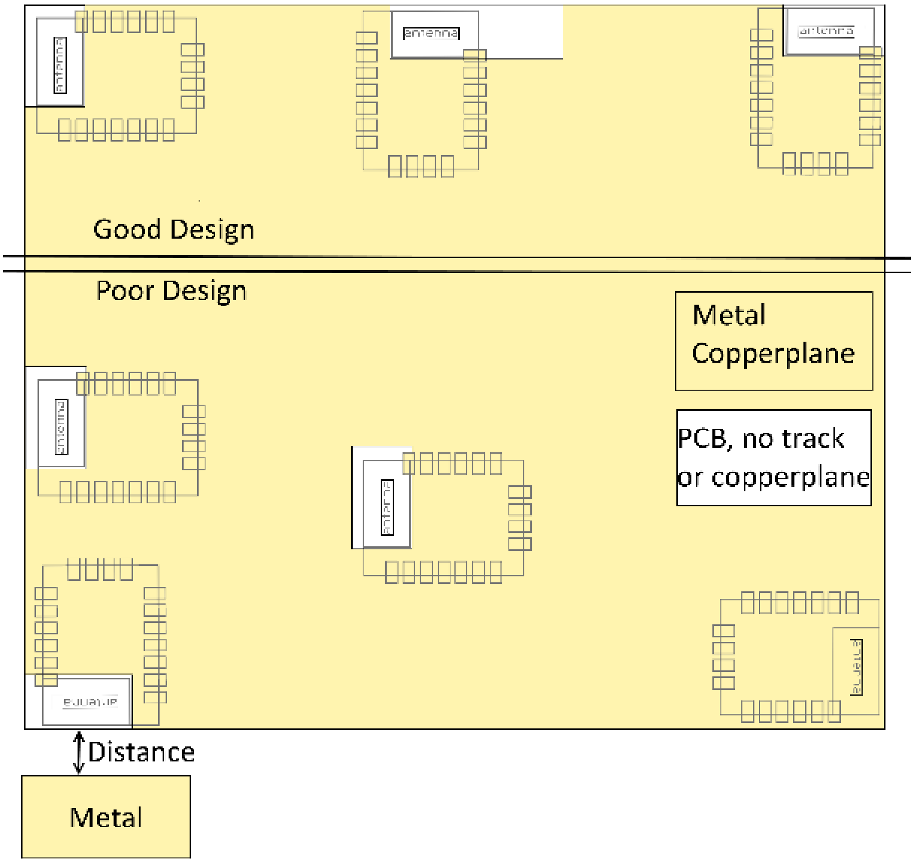


Figure 15: Placement of the module with integrated antenna

13.2 Dimensioning of the micro strip antenna line

The antenna track has to be designed as a 50 Ω feed line. The width W for a micro strip can be calculated using the following equation:

$$W = 1.25 \times \left(\frac{5.98 \times H}{e^{\frac{50 \times \sqrt{\epsilon_r + 1.41}}{87}}} - T_{met} \right) \quad (1)$$

Example:
A FR4 material with ε_r = 4.3, a height H = 1000 µm and a copper thickness of T_{met} = 18 µm will

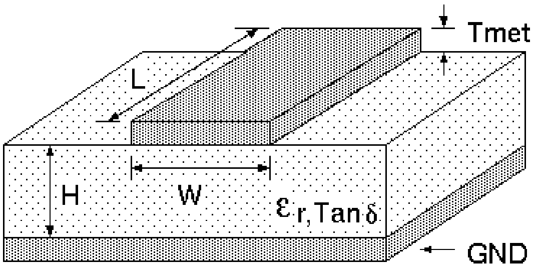


Figure 16: Dimensioning the antenna feed line as micro strip

lead to a trace width of W ~ 1.9 mm. To ease the calculation of the micro strip line (or e.g. a coplanar) many calculators can be found in the internet.

- As rule of thumb a distance of about 3×W should be observed between the micro strip and other traces / ground.
- The micro strip refers to ground, therefore there has to be the ground plane underneath the trace.
- Keep the feeding line as short as possible.

13.3 Antenna solutions

There exist several kinds of antennas, which are optimized for different needs. Chip antennas are optimized for minimal size requirements but at the expense of range, PCB antennas are optimized for minimal costs, and are generally a compromise between size and range. Both usually fit inside a housing.
Range optimization in general is at the expense of space. Antennas that are bigger in size, so that they would probably not fit in a small housing, are usually equipped with a RF connector. A benefit of this connector may be to use it to lead the RF signal through a metal plate (e.g. metal housing, cabinet).
As a rule of thumb a minimum distance of λ /10 (which is 3.5 cm @ 868 MHz and 1.2 cm @ 2.44 GHz) from the antenna to any other metal should be kept. Metal placed further away will not directly influence the behavior of the antenna, but will anyway produce shadowing.

! Keep the antenna as far as possible from large metal objects to avoid electromagnetic field blocking.

! The choice of antenna might have influence on the safety requirements.

In the following chapters, some special types of antenna are described.

13.3.1 Wire antenna

An effective antenna is a λ /4 radiator with a suiting ground plane. The simplest realization is a piece of wire. It's length is depending on the used radio frequency, so for example 8.6 cm 868.0 MHz and 3.1 cm for 2.440 GHz as frequency. This radiator needs a ground plane at its feeding point. Ideally, it is placed vertically in the middle of the ground plane. As this is often not possible because of space requirements, a suitable compromise is to bend the wire away from the PCB respective to the ground plane. The λ/4 radiator has approximately 40 Ω input impedance. Therefore, matching is not required.

13.3.2 Chip antenna

There are many chip antennas from various manufacturers. The benefit of a chip antenna is obviously the minimal space required and reasonable costs. However, this is often at the expense of range. For the chip antennas, reference designs should be followed as closely as possible, because only in this constellation can the stated performance be achieved.

13.3.3 PCB antenna

PCB antenna designs can be very different. The special attention can be on the miniaturization or on the performance. The benefits of the PCB antenna are their small / not existing (if PCB space is available) costs, however the evaluation of a PCB antenna holds more risk of failure than the use of a finished antenna. Most PCB antenna designs are a compromise of range and space between chip antennas and connector antennas.

14 Reference design

Stephano-I was tested and certified on the corresponding Stephano-I evaluation board. For the compliance with the EU directive 2014/53/EU Annex I, the evaluation board serves as reference design.
This is no discrepancy due to the fact that the evaluation board itself does not fall within the scope of the EU directive 2014/53/EU Annex I as the module is tested on the evaluation board, which is also the recommended use.

Further information concerning the use of the evaluation board can be found in the manual of the Stephano-I evaluation board.

14.1 EV-Board

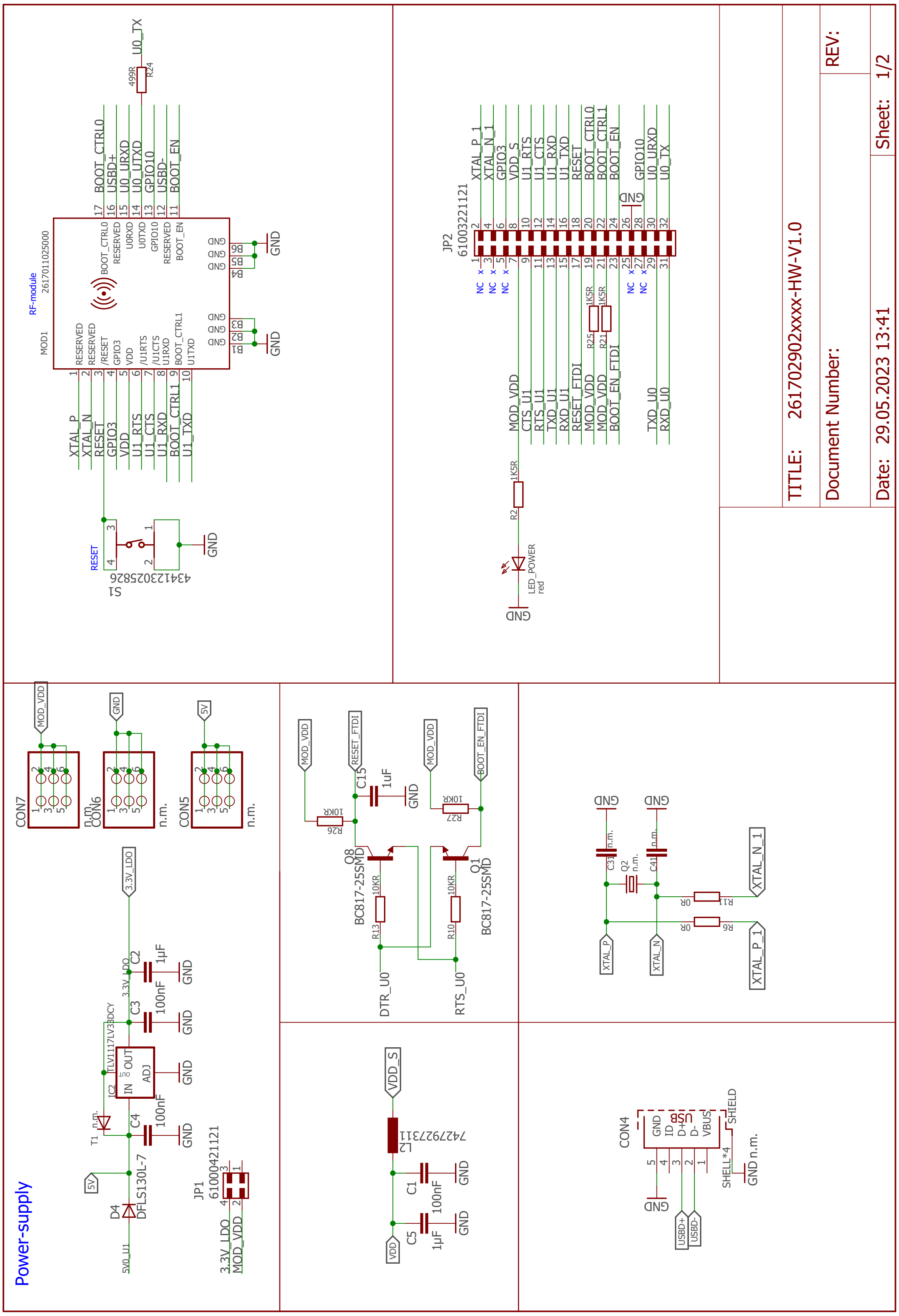


Figure 17: Reference design: Schematic page 1

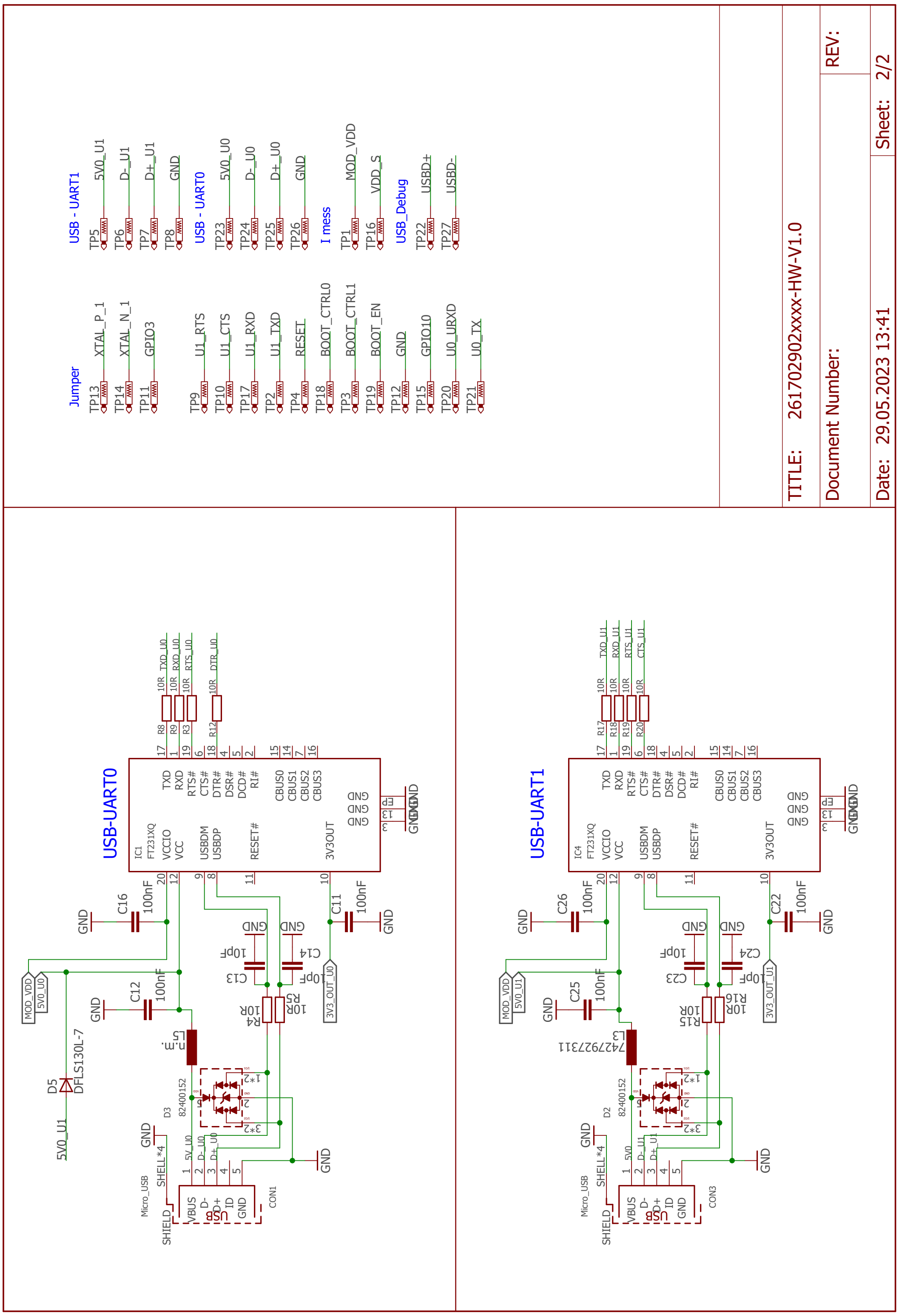


Figure 18: Reference design: Schematic page 2

14.2 Layout

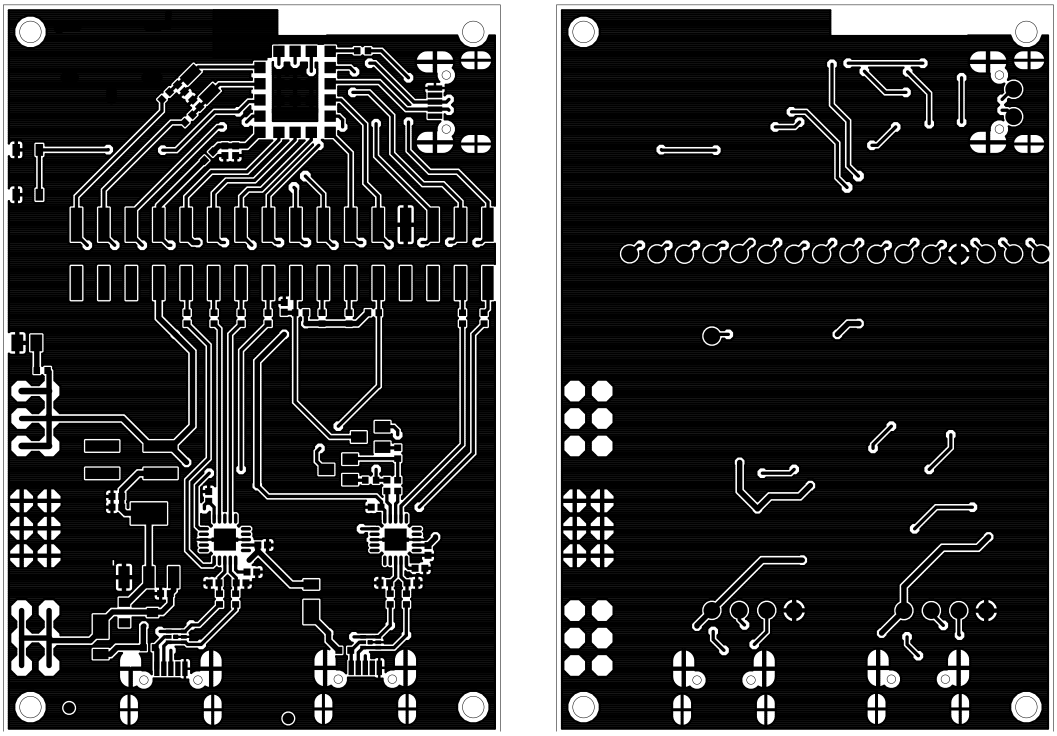


Figure 19: Top layer (top), bottom layer (bottom)

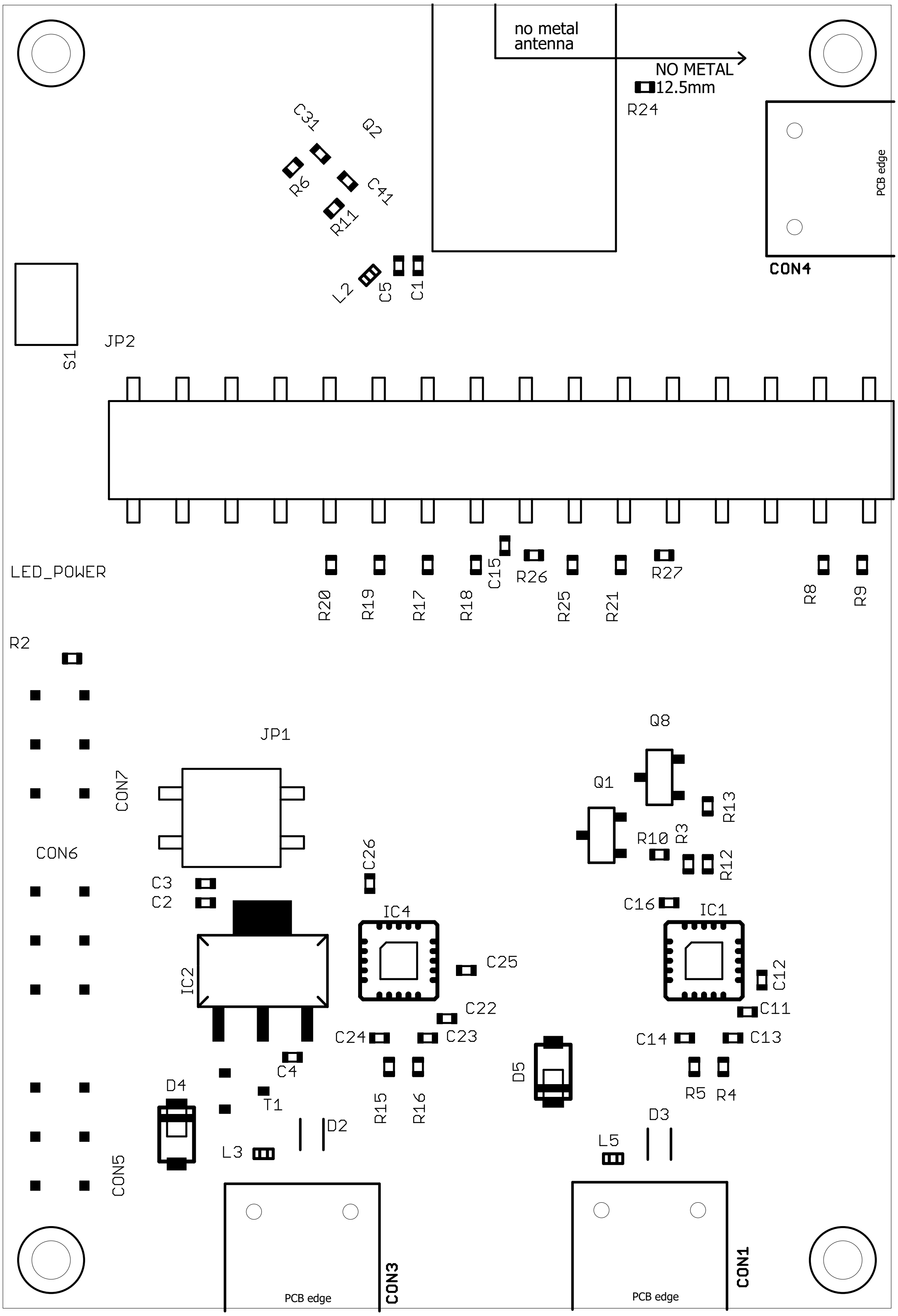


Figure 20: Reference design assembly plan

14.3 Circuit and components for certification tests of the end device

For the certification of the end device, that integrates the Stephano-I, it is necessary to set the Stephano-I to transmission mode and check the radio emissions. To do that special test firmware (ESP RF Test Tool and Test Guide [2]) for the Stephano-I may be needed, that is provided by Espressif.
To control and bring the test firmware on the radio module, the debug UART (UART0) must be accessible and a circuit must be added, that controls the */RESET* and *BOOT_EN* pin of the Stephano-I (see figure 21).

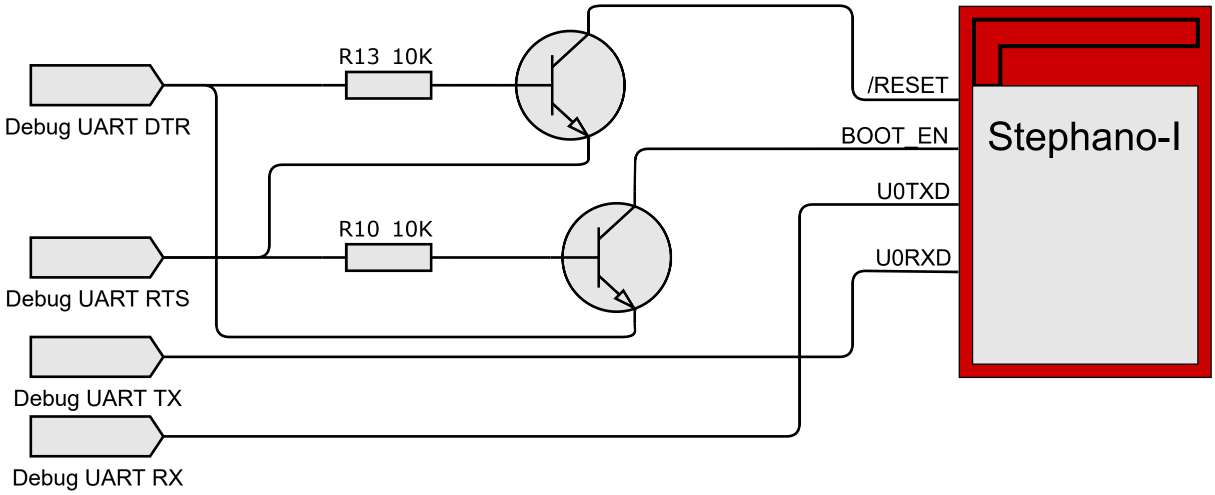


Figure 21: Flash circuit

This adaption is only needed to flash and control test firmware, needed for certification. In the final product the debug UART as well as the flash circuit are without use.

15 Manufacturing information

15.1 Moisture sensitivity level

This wireless connectivity product is categorized as JEDEC Moisture Sensitivity Level 3 (MSL3), which requires special handling.

More information regarding the MSL requirements can be found in the IPC/JEDEC J-STD-020 standard on *www.jedec.org*.

More information about the handling, picking, shipping and the usage of moisture/reflow and/or process sensitive products can be found in the IPC/JEDEC J-STD-033 standard on *www.jedec.org*.

15.2 Soldering

15.2.1 Reflow soldering

Attention must be paid on the thickness of the solder resist between the host PCB top side and the modules bottom side. Only lead-free assembly is recommended according to JEDEC J-STD020.

Profile feature		Value
Preheat temperature Min	$T_{S\ Min}$	150 °C
Preheat temperature Max	$T_{S\ Max}$	200 °C
Preheat time from $T_{S\ Min}$ to $T_{S\ Max}$	t_S	60 - 120 seconds
Ramp-up rate (T_L to T_P)		3 °C / second max.
Liquidous temperature	T_L	217 °C
Time t_L maintained above T_L	t_L	60 - 150 seconds
Peak package body temperature	T_P	see table below
Time within 5 °C of actual peak temperature	t_P	20 - 30 seconds
Ramp-down Rate (T_P to T_L)		6 °C / second max.
Time 20 °C to T_P		8 minutes max.

Table 121: Classification reflow soldering profile, Note: refer to IPC/JEDEC J-STD-020E

Package thickness	Volume mm^3 <350	Volume mm^3 350-2000	Volume mm^3 >2000
< 1.6 mm	260 ℃	260 ℃	260 ℃
1.6 mm - 2.5 mm	260 ℃	250 ℃	245 ℃
> 2.5 mm	250 ℃	245 ℃	245 ℃

Table 122: Package classification reflow temperature, PB-free assembly, Note: refer to IPC/-JEDEC J-STD-020E

It is recommended to solder this module on the last reflow cycle of the PCB. For solder paste use a LFM-48W or Indium based SAC 305 alloy (Sn 96.5 / Ag 3.0 / Cu 0.5 / Indium 8.9HF / Type 3 / 89%) type 3 or higher.

The reflow profile must be adjusted based on the thermal mass of the entire populated PCB, heat transfer efficiency of the reflow oven and the specific type of solder paste used. Based on the specific process and PCB layout the optimal soldering profile must be adjusted and verified. Other soldering methods (e.g. vapor phase) have not been verified and have to be validated by the customer at their own risk. Rework is not recommended.

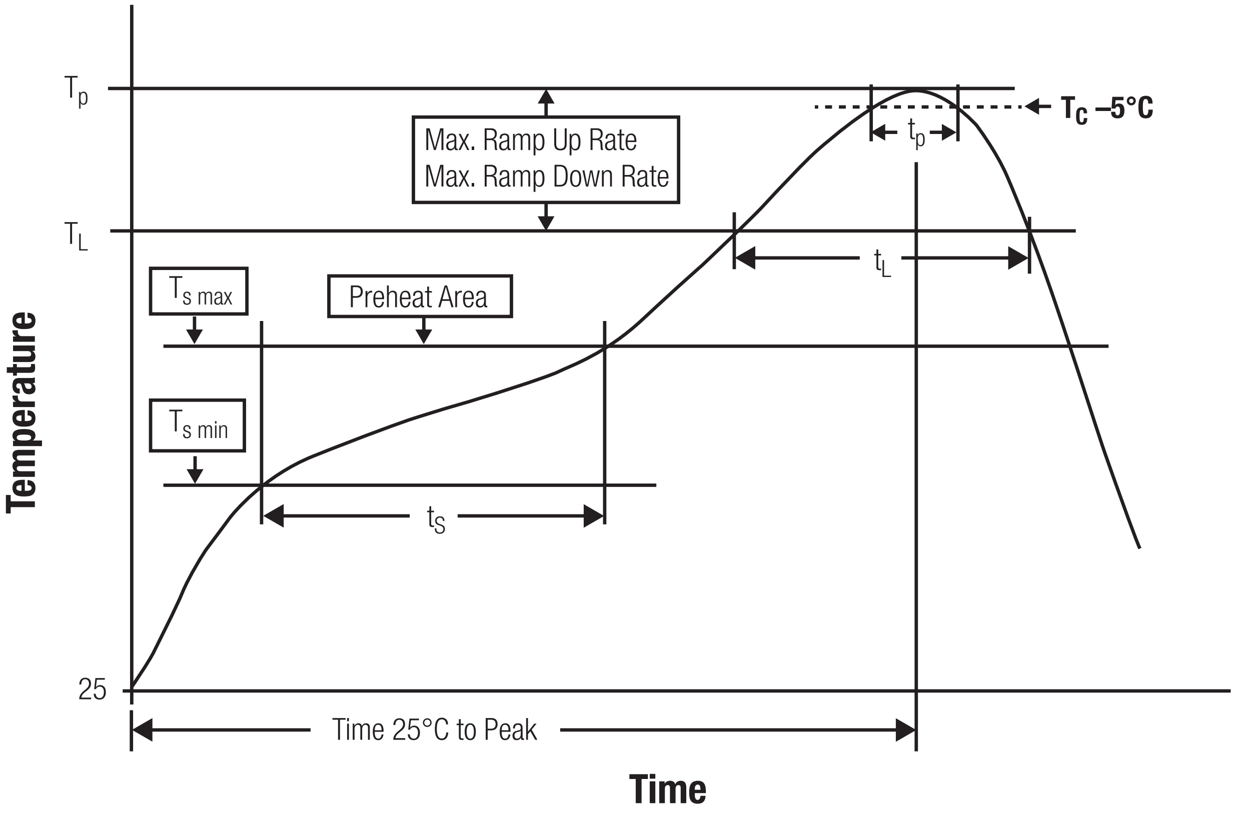


Figure 22: Reflow soldering profile

After reflow soldering, visually inspect the board to confirm proper alignment

15.2.2 Cleaning

Do not clean the product. Any residue cannot be easily removed by washing. Use a "no clean" soldering paste and do not clean the board after soldering.

- Do not clean the product with water. Capillary effects can draw water into the gap between the host PCB and the module, absorbing water underneath it. If water is trapped inside, it may short-circuit adjoining pads. The water may also destroy the label and ink-jet printed text on it.
- Cleaning processes using alcohol or other organic solvents may draw solder flux residues into the housing, which won't be detected in a post-wash inspection. The solvent may also destroy the label and ink-jet printed text on it.
- Do not use ultrasonic cleaning as it will permanently damage the part, particularly the crystal oscillators.

15.2.3 Potting and coating

- If the product is potted in the customer application, the potting material might shrink or expand during and after hardening. Shrinking could lead to an incomplete seal, allowing contaminants into the component. Expansion could damage components. We recommend a manual inspection after potting to avoid these effects.
- Conformal coating or potting results in loss of warranty.
- The RF shield will not protect the part from low-viscosity coatings and potting. An undefined amount of coating and potting will enter inside the shielding.
- Conformal coating and potting will influence the parts of the radio front end and consequently influence the radio performance.
- Potting will influence the temperature behaviour of the device. This might be critical for components with high power.

15.2.4 Other notations

- Do not attempt to improve the grounding by forming metal strips directly to the EMI covers or soldering on ground cables, as it may damage the part and will void the warranty.
- Always solder every pad to the host PCB even if some are unused, to improve the mechanical strength of the module.
- The part is sensitive to ultrasonic waves, as such do not use ultrasonic cleaning, welding or other processing. Any ultrasonic processing will void the warranty.

15.3 ESD handling

This product is highly sensitive to electrostatic discharge (ESD). As such, always use proper ESD precautions when handling. Make sure to handle the part properly throughout all stages of production, including on the host PCB where the module is installed. For ESD ratings, refer to the module series' maximum ESD section. For more information, refer to the relevant chapter 2. Failing to follow the aforementioned recommendations can result in severe damage to the part.

- the first contact point when handling the PCB is always between the local GND and the host PCB GND, unless there is a galvanic coupling between the local GND (for example work table) and the host PCB GND.
- Before assembling an antenna patch, connect the grounds.
- While handling the RF pin, avoid contact with any charged capacitors and be careful when contacting any materials that can develop charges (for example coaxial cable with around 50-80 pF/m, patch antenna with around 10 pF, soldering iron etc.)
- Do not touch any exposed area of the antenna to avoid electrostatic discharge. Do not let the antenna area be touched in a non ESD-safe manner.
- When soldering, use an ESD-safe soldering iron.

15.4 Safety recommendations

It is your duty to ensure that the product is allowed to be used in the destination country and within the required environment. Usage of the product can be dangerous and must be tested and verified by the end user. Be especially careful of:

- Use in areas with risk of explosion (for example oil refineries, gas stations).
- Use in areas such as airports, aircraft, hospitals, etc., where the product may interfere with other electronic components.

It is the customer's responsibility to ensure compliance with all applicable legal, regulatory and safety-related requirements as well as applicable environmental regulations. Disassembling the product is not allowed. Evidence of tampering will void the warranty.

- Compliance with the instructions in the product manual is recommended for correct product set-up.
- The product must be provided with a consolidated voltage source. The wiring must meet all applicable fire and security prevention standards.
- Handle with care. Avoid touching the pins as there could be ESD damage.

Be careful when working with any external components. When in doubt consult the technical documentation and relevant standards. Always use an antenna with the proper characteristics.

❗ Würth Elektronik eiSos radio modules with high output power of up to 500 mW, as for example the radio module Thebe-II, generate a high amount of warmth while transmitting. The manufacturer of the end device must take care of potentially necessary actions for his application.

16 Physical specifications

16.1 Dimensions

Dimensions
9.5 x 13 x 2 mm

Table 123: Dimensions

16.2 Weight

Weight
<1g

Table 124: Weight

16.3 Module drawing

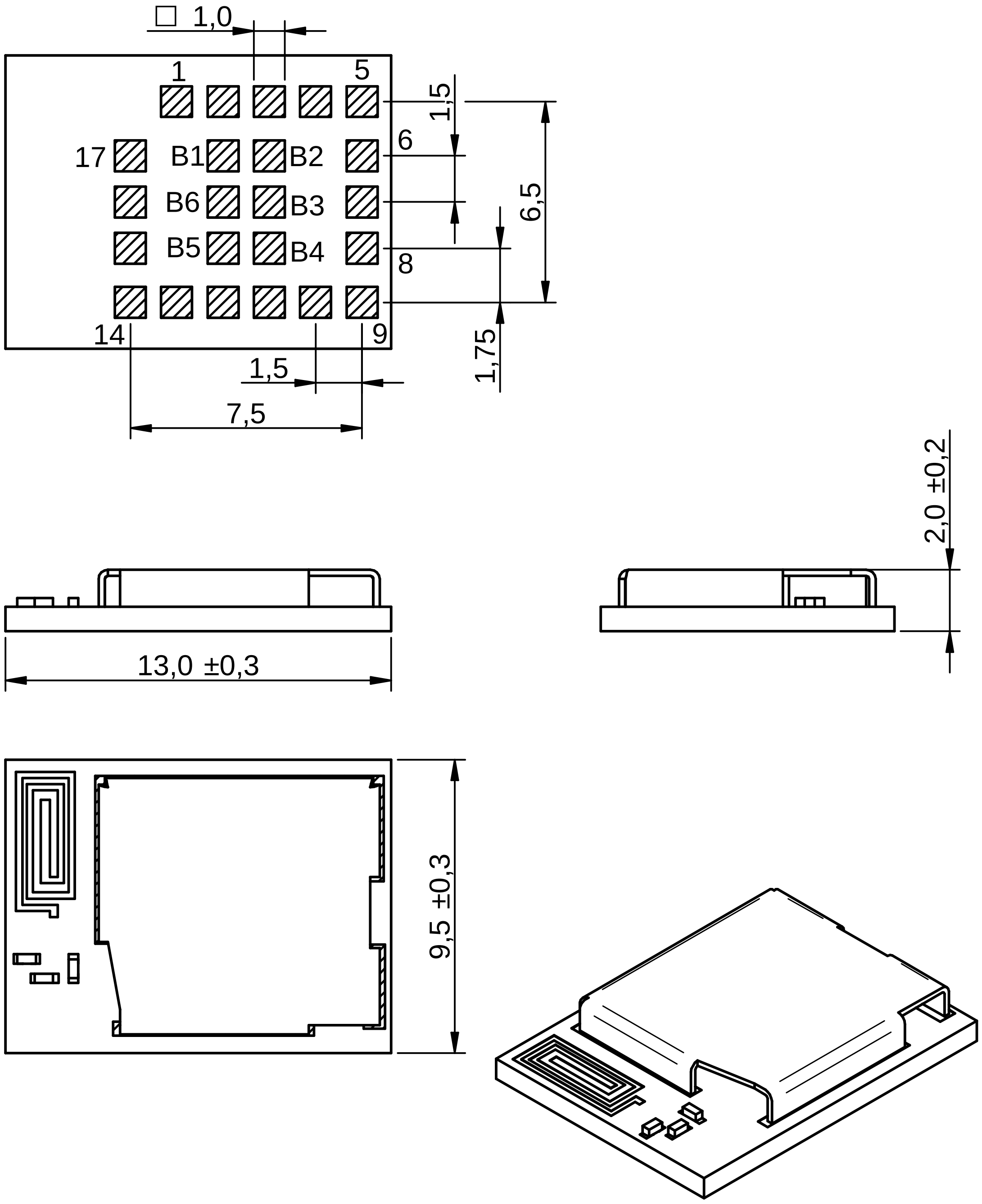


Figure 23: Module dimensions [mm]

16.4 Footprint

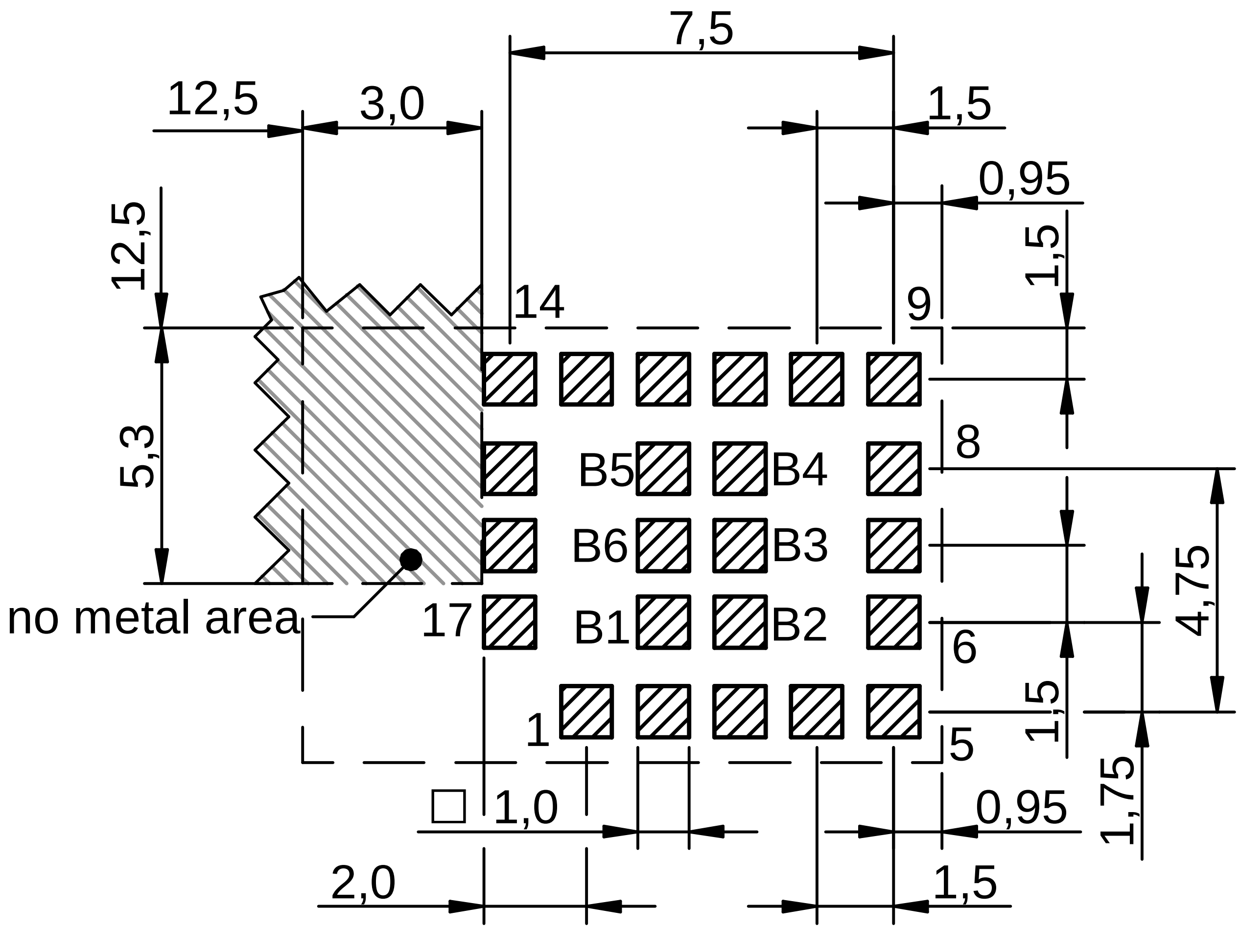


Figure 24: Footprint [mm]

16.5 Antenna free area

To avoid influence and mismatching of the antenna the recommended free area around the antenna should be maintained. As rule of thumb a minimum distance of metal parts to the antenna of λ/10 should be kept (see figure 24). Even though metal parts would influence the characteristic of the antenna, but the direct influence and matching keep an acceptable level.

17 Marking

17.1 Lot number

The 15 digit lot number is printed in numerical digits as well as in form of a machine readable bar code. It is divided into 5 blocks as shown in the following picture and can be translated according to the following table.

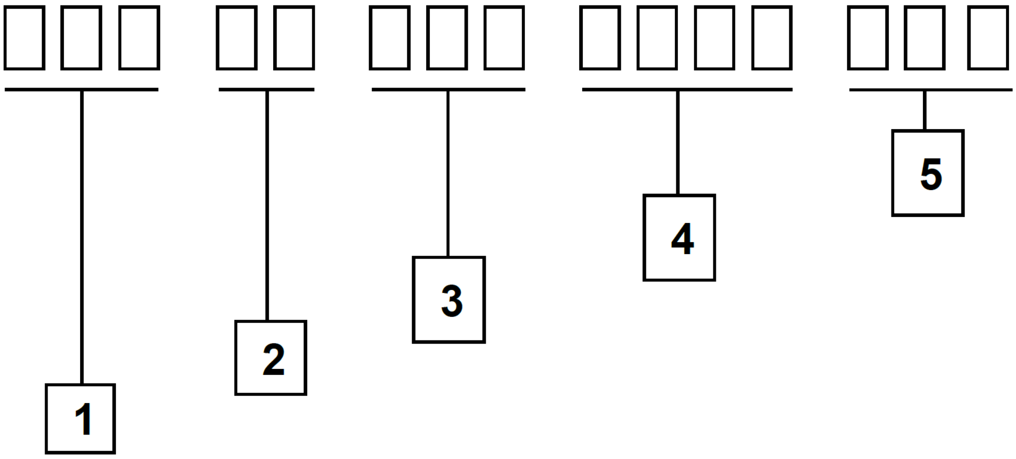


Figure 25: Lot number structure

Block	Information	Example(s)
1	eiSos internal, 3 digits	439
2	eiSos internal, 2 digits	01
3	Hardware version, 3 digits	V2.4 = 024, V12.2 = 122
4	Date code, 4 digits	1703 = week 03 in year 2017, 1816 = week 16 in year 2018
5	Firmware version, 3 digits	V3.2 = 302, V5.13 = 513

Table 125: Lot number details

As the user can perform a firmware update the printed lot number only shows the factory delivery state. The currently installed firmware can be requested from the module using the corresponding product specific command. The firmware version as well as the hardware version are restricted to show only major and minor version not the patch identifier.

17.2 General labeling information

The module labels may include the following fields:

- Manufacturer identification WE, Würth Elektronik or Würth Elektronik eiSos
- WE Order Code and/or article alias
- Serial number or MAC address
- Certification identifiers (CE, FCC ID, IC, TELEC,...)
- Bar code or 2D code containing the serial number or MAC address

If the module is using a serial Number, this serial number includes the product ID (PID) and an 6 digit number. The 6 rightmost digits represent the 6 digit number, followed by the product ID (2 or 3 digits). Some labels indicate the product ID with a "." as marker in-between the 2 fields. The PID and the 6 digit number form together a unique serial number for any wireless connectivity product.
In case of small labels, the 3 byte manufacturer identifier (0x0018DA) of the MAC address is not printed on the labels. The 3 byte counter printed on the label can be used with this 0018DA to produce the full MAC address by appending the counter after the manufacturer identifier.

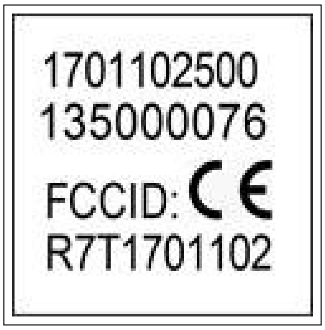


Figure 26: Label of the Stephano-I

18 Information for explosion protection

In case the end product should be used in explosion protection areas the following information can be used:

- The module itself is unfused.
- The maximum power of the module is 21 dBm for external antenna and 15 dBm for internal antenna.
- The total amount of capacitance of all capacitors is 3.61 µF.
- The total amount of inductance of all inductors is 27.9 nH.

19 References

[1] Espressif. AT command documenation for Stephano-I, version 3.2.0.0. https://docs.espressif.com/projects/esp-at/en/release-v3.2.0.0/esp32c3/index.html.

[2] Espressif. Espressif tools download page. https://www.espressif.com/en/support/download/other-tools.

[3] Espressif. Examples for Stephano-I, version 3.2.0.0. https://docs.espressif.com/projects/esp-at/en/release-v3.2.0.0/esp32c3/AT_Command_Examples/index.html.

[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 27 - Bluetooth listing guide. http://www.we-online.com/ANR027.

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

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

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

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

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

[12] Würth Elektronik. Smart Commander PC tool. https://www.we-online.com/SmartCommander.

[13] Würth Elektronik. Web page: Stephano-I radio module. https://www.we-online.de/katalog/de/article/2617011025000.

[14] Würth Elektronik. Wireless Connectivity SDK for STM32 - Radio module drivers in C-code. https://github.com/WurthElektronik/WirelessConnectivity-SDK_STM32.

20 Bluetooth SIG listing/qualification

Type	Data
Design name	Stephano-I
Declaration ID	
QDID	
Specification name	Bluetooth 5.3
Project type	End product

Each product containing intellectual property of the Bluetooth® Special Interest Group (SIG) must be qualified by the SIG to obtain the corresponding Declaration ID.
Due to the qualification of the Stephano-I as end product no further Bluetooth® tests are required. The only arising expenses are those for purchasing a Bluetooth® Declaration ID.
To obtain the Bluetooth® listing of the end device, please refer to the application note ANR027 [6].

21 Regulatory compliance information

21.1 Important notice EU

The use of RF frequencies is limited by national regulations. The Stephano-I has been designed to comply with the RED directive 2014/53/EU of the European Union (EU).
The Stephano-I can be operated without notification and free of charge in the area of the European Union. However, according to the RED directive, restrictions (e.g. in terms of duty cycle or maximum allowed RF power) may apply.

> ❗ Since the module itself is not fused the voltage supply shall be fed from a power source which is class PS2 according to EN 62368-1.

21.2 Important notice FCC

The use of RF frequencies is limited by national regulations. The Stephano-I has been designed to comply with the FCC Part 15.
The Stephano-I can be operated without notification and free of charge in the area of the United States of America. However, according to the FCC Part 15, restrictions (e.g. in terms of maximum allowed RF power and antenna) may apply.

21.3 Conformity assessment of the final product

The Stephano-I is a subassembly. It is designed to be embedded into other products (products incorporating the Stephano-I are henceforward referred to as "final products").
It is the responsibility of the manufacturer of the final product to ensure that the final product is in compliance with the essential requirements of the underlying national radio regulations.
The conformity assessment of the subassembly Stephano-I carried out by Würth Elektronik eiSos does not replace the required conformity assessment of the final product.

21.4 Exemption clause

Relevant regulation requirements are subject to change. Würth Elektronik eiSos does not guarantee the accuracy of the before mentioned information. Directives, technical standards, procedural descriptions and the like may be interpreted differently by the national authorities. Equally, the national laws and restrictions may vary with the country. In case of doubt or uncertainty, we recommend that you consult with the authorities or official certification organizations of the relevant countries. Würth Elektronik eiSos is exempt from any responsibilities or liabilities related to regulatory compliance.

Notwithstanding the above, Würth Elektronik eiSos makes no representations and warranties of any kind related to their accuracy, correctness, completeness and/or usability for customer applications. No responsibility is assumed for inaccuracies or incompleteness.

21.5 FCC Compliance Statement (US)

FCC ID: R7T1701102

This device complies with Part 15 of the FCC Rules.
Operation is subject to the following two conditions:
(1) this device may not cause harmful interference, and
(2) this device must accept any interference received, including interference that may cause undesired operation.
(FCC 15.19)

Modifications (FCC 15.21)
Caution: Changes or modifications for this equipment not expressly approved by Würth Elektronik eiSos may void the FCC authorization to operate this equipment.

21.6 IC Compliance Statement (Canada)

Certification Number: 5136A-1701102
HVIN: 1701102
This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.
Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

21.7 FCC and IC requirements to OEM integrators

This module has been granted modular approval. OEM integrators for host products may use the module in their final products without additional FCC/IC (Industry Canada) certification if they meet the following conditions. Otherwise, additional FCC/IC approvals must be obtained.
The host product with the module installed must be evaluated for simultaneous transmission requirements.

- The users manual for the host product must clearly indicate the operating requirements and conditions that must be observed to ensure compliance with current FCC/IC RF exposure guidelines.
- A label must be affixed to the outside of the host product with the following statements:
  This device contains FCCID: R7T1701102
  This equipment contains equipment certified under ICID: 5136A-1701102
- The final host / module combination may also need to be evaluated against the FCC Part 15B criteria for unintentional radiators in order to be properly authorized for operation as a Part 15 digital device.

- If the final host / module combination is intended for use as a portable device (see classifications below) the host manufacturer is responsible for separate approvals for the SAR requirements from FCC Part 2.1093 and RSS-102. Using the maximum output power, the mounted Stephano-I must respect a distance of 200 mm in WiFi mode and 25 mm in Bluetooth® mode to a human body. If a reduced output power is configured, the safety distance can be recalculated corresponding to the KDB447498 and RSS-102 Issue 5.

OEM requirements:
The OEM must ensure that the following conditions are met.

- End users of products, which contain the module, must not have the ability to alter the firmware that governs the operation of the module. The agency grant is valid only when the module is incorporated into a final product by OEM integrators.
- The end-user must not be provided with instructions to remove, adjust or install the module.
- The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product. Attaching a label to a removable portion of the final product, such as a battery cover, is not permitted.
- The label must include the following text:
  Contains FCC ID: R7T1701102
  The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:
  (i.) this device may not cause harmful interference and
  (ii.) this device must accept any interference received, including interference that may cause undesired operation.

  When the device is so small or for such use that it is not practicable to place the statement above on it, the information required by this paragraph shall be placed in a prominent location in the instruction manual or pamphlet supplied to the user or, alternatively, shall be placed on the container in which the device is marketed. However, the FCC identifier or the unique identifier, as appropriate, must be displayed on the device.
- The user manual for the end product must also contain the text given above.
- Changes or modifications not expressly approved could void the user's authority to operate the equipment.
- The OEM must ensure that timing requirements according to 47 CFR 15.231(a-c) are met.
- The OEM must sign the OEM Modular Approval Agreement.
- The OEM shall perform testing in accordance to 996369 D04 Module Integration Guide V01.

21.8 TELEC radio law approval

Japanese Radio Law Compliance

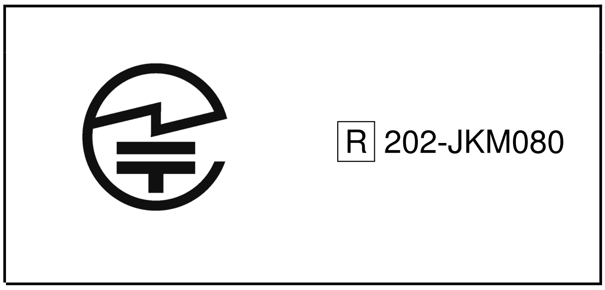
This device has passed the Radio Law approval for Japan through the registered certification body TELEC. The corresponding ARIB (Association of Radio Industries and Businesses) standard has been applied. Accordingly, the market approval is given by the MIC (Ministry of Internal Affairs and Communications).

This device should not be modified (otherwise the granted designation number will become invalid)

21.8.1 Label

Due to the small size of the Stephano-I label, the certification label of the Stephano-I is not placed onto the module label.

2617011025000:



After integration of the Stephano-I in the end device, the corresponding certification label must be recognized from the outside. Otherwise this information must be referenced on the housing as well as in the user manual. E labeling is allowed.

21.9 Certified antennas

The Stephano-I is pre-certified with the following antennas.

Product	Certified antenna
Stephano-I (2617011025000)	PCB antenna included in the Stephano-I

22 Important notes

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

22.1 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.

22.2 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.

22.3 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.

22.4 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.

22.5 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 22.1 and 22.2 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.

22.6 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.

22.7 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.

22.8 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*.

23 Legal notice

23.1 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.

23.2 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. KG and its representatives against any damages arising out of the use of any Würth Elektronik eiSos GmbH & Co. KG components in safety-critical applications.

23.3 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.

23.4 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.

24 License terms

These 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 these license terms are 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 these license terms. You agree to comply with all provisions under these license terms.

24.1 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 these 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 these 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 these 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.

24.2 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.

24.3 Ownership

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

24.4 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.

24.5 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.

24.6 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.

24.7 Applicable law and jurisdiction

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

24.8 Severability clause

If a provision of these 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.

24.9 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 product, you accept these license terms in all terms.

List of Figures

List of Tables

WE

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