UM3045

User manual

# Getting started with the STSW-PROTEUS software package for the STEVAL-PROTEUS1 industrial sensor node kit

## Introduction

The STSW-PROTEUS is an STM32Cube-based software package for vibration and temperature condition equipment monitoring over Bluetooth® Low Energy and ZigBee connectivity.

The application captures vibration and temperature data from MEMS sensors. It uses them as inputs to perform complex algorithms such as frequency and time domain vibration analysis. Then, it transfers the ready-to-use results into a wireless personal area network (WPAN). This feature makes the IoT node suitable for condition-based maintenance (CBM) and predictive maintenance (PdM) to reduce productivity losses due to an unplanned machine downtime.

The STSW-PROTEUS software package includes two different projects to address both Bluetooth® Low Energy and Zigbee applications.

The Bluetooth® Low Energy application allows directly connecting a smartphone through a dedicated mobile app (STBLESensClassic) to facilitate nodes configuration, local monitoring equipment status, fast firmware upgrade over the air (FUOTA), and data monitoring to the dedicated Azure IoT PnP central cloud dashboard.

The ZigBee application firmware provides an example to make different nodes, such as end devices and routers, communicate within the same ecosystem, thanks to the strength of the mesh network. A coordinator runs on the NUCLEO-WB55RG and collects the network data, displaying them on a PC serial terminal.

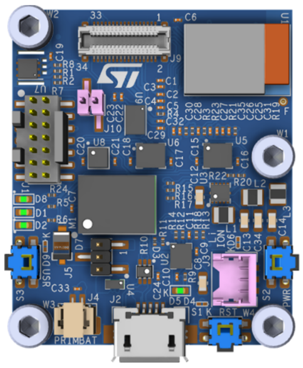
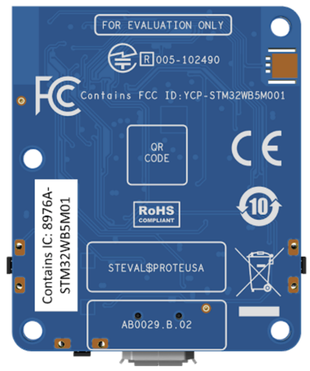
Each project is released as a source code for the STEVAL-PROTEUS1 kit. The projects are developed for three IDEs for the STM32 MCU: IAR Embedded Workbench for Arm, Keil® microcontroller development kit for Arm, and ST integrated development environment for STM32.

# 1 Hardware overview

## 1.1 Compatible hardware platform

The STSW-PROTEUS software package is tailored for the STEVAL-PROTEUS evaluation board, which is the main board of the STEVAL-PROTEUS1 kit.

Table 1. **Compatible hardware platform overview**

| Features | STEVAL-PROTEUS platform | |
|---|---|---|
| Board picture (top and bottom views) |  |  |
| Size (mm) | 34.97 x 29.25 x 7 | |
| Vibration analysis | Up to 6 kHz | |
| Connectivity | Bluetooth® 5 | |
| | zigbee 3.0 | |
| Datalog and GUI | Bluetooth® Low Energy | STBLESensClassic app for Android an iOS |
| | | Azure IoT Central cloud platform |
| | Zigbee | Terminal emulator via UART |
| Development tools | STLINK-V3MINI, STLINK-V3MINIE | |

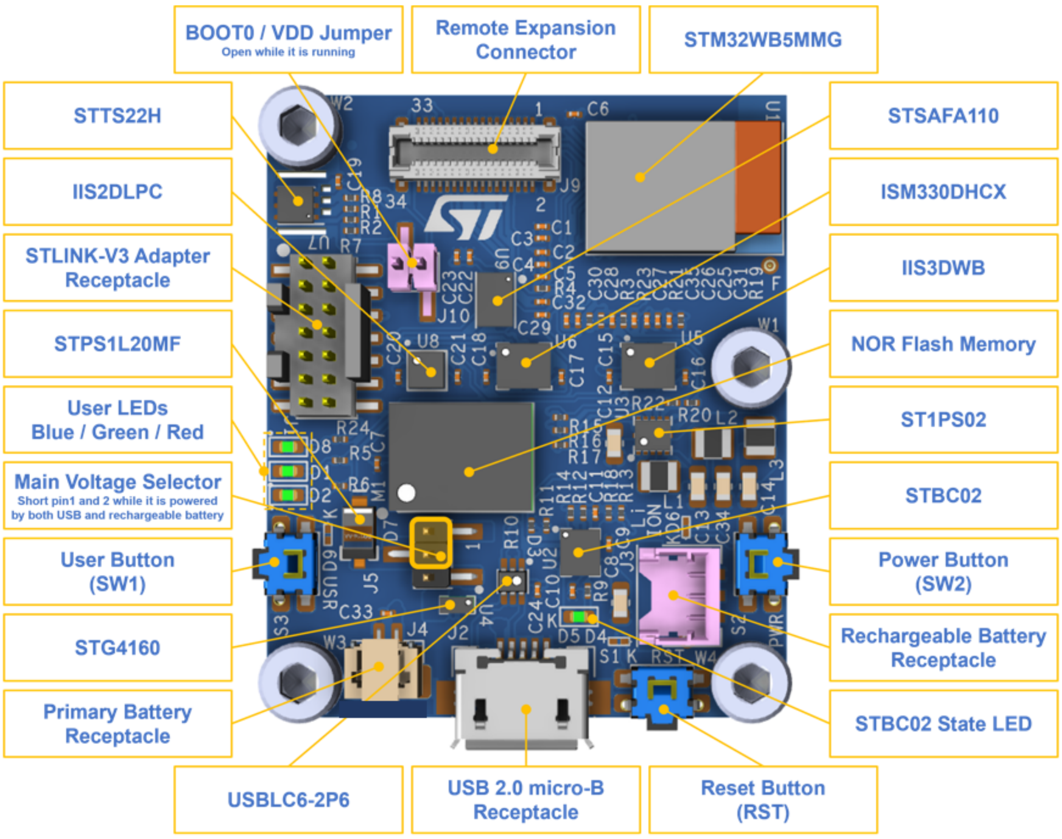
### 1.1.1 STEVAL-PROTEUS

The STEVAL-PROTEUS is the main board of the STEVAL-PROTEUS1 kit. It embeds an ULP and small form factor certified 2.4 GHz wireless module that supports Bluetooth® Low Energy 5.0, Zigbee® 3.0, OpenThread, dynamic and static concurrent modes, and 802.15.4 proprietary protocols.

The STM32WB5MMG provides the best-in-class RF performance thanks to its good receiver sensitivity and a high output power signal. The module is a complete royalty-free protocol stack. It features a dual Arm® core, a dedicated Arm® Cortex®-M0+ for radio and security tasks, and a dedicated Arm Cortex®-M4 CPU with FPU and adaptive real-time accelerator (ART) with a speed up to 64 MHz. It also comes with a 1-Mbyte flash memory and 256-Kbyte SRAM. These features allow running complex algorithms directly through the board, for ready-to-use results.

The module includes ST MEMS sensors for battery-operated applications, targeting IoT and Industry 4.0.

Figure 1. **STEVAL-PROTEUS hardware architecture (top view)**



The board embeds:

- inertial modules:
  - IIS3DWB ultra-wide bandwidth, low-noise, three-axis digital vibration sensor
  - IIS2DLPC high-performance, ULP, 3-axis accelerometer
  - ISM330DHCX 3-axis accelerometer and 3-axis gyroscope with machine learning core (MLC)
- STTS22H low-voltage, ULP, high accuracy temperature sensor. It is placed far from the heat noise sources (the power management and the microcontroller) to provide a more precise temperature measurement. Its exposed pad and the PCB accurate design allow the temperature sensor to be in contact with the surface of the target equipment
- STSAFE-A110 secure element that provides authentication and secure data management services to a local or remote host
- A NOR flash memory that is connected via QSPI to the STM32WB5MMG module for data buffering and event storage
- A remote expansion connector, which exposes the STMOD+ features and more
- For power management:
  - ST1PS02 400 mA step down converter for low-power applications
  - STBC02 Li-Ion linear battery charger

The STEVAL-PROTEUS can be powered through a LiPo rechargeable battery. It can also be powered via USB (5 V at 500 mA) or via a primary battery (which is not included in the kit).

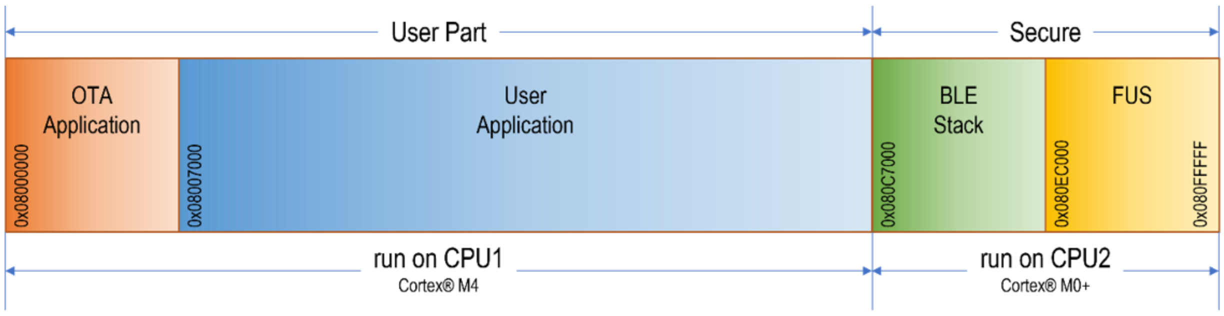
All components are mounted exclusively on the top side of the PCB to ensure easy mounting on other equipment, even when using the provided plastic case that can enclose the board and the provided LiPo battery.

#### 1.1.1.1 *Flash memory management*

The STEVAL-PROTEUS comes with a 1 Mbyte flash memory. It is shared between the two Arm® cores as the user part to be run into the CPU1 (Cortex® M4) and the secure part to be run into the CPU2 (Cortex® M0+). The user part starts from the beginning of the flash memory, whereas the secure part is placed at the end. The latter size depends on the wireless stack to be used.

The Bluetooth® Low Energy application allows running the over-the-air (OTA) firmware update. For this update, download the OTA application firmware from the first main memory address. Then, download the demo application a few pages later.

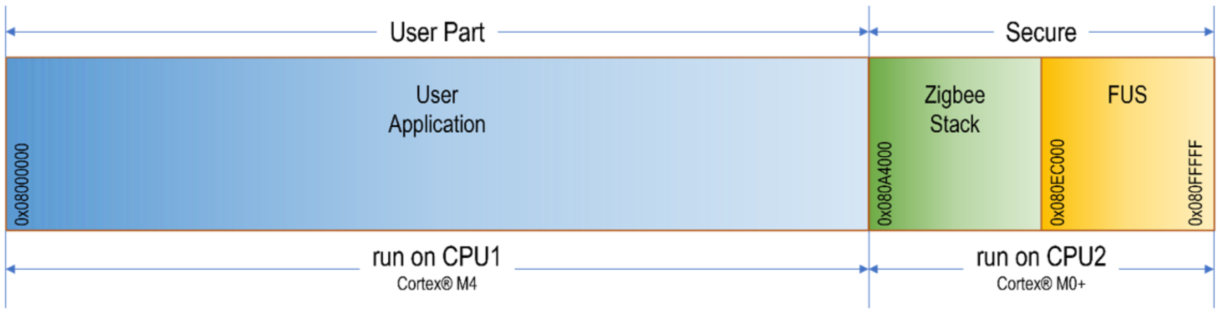
Figure 2. **STEVAL-PROTEUS flash memory map for Bluetooth® Low Energy applications**



AN5247 describes the procedure for OTA firmware update on ST32WB devices with Bluetooth® Low Energy connection.

The Zigbee application does not use the OTA firmware update.

Figure 3. **STEVAL-PROTEUS flash memory map for Zigbee applications**



# 2 Firmware overview

The key features of the STSW-PROTEUS software package are:

- Firmware package to demonstrate an industrial sensor node for condition-based monitoring (CbM) and predictive maintenance (PdM) applications in a WPAN based on Bluetooth® Low Energy or Zigbee connectivity
- Temperature and motion sensors real-time monitoring of meaningful key parameters and equipment status
- STM32 wireless personal area network middleware developed within the STM32WB framework used to support Bluetooth® Low Energy 5 or Zigbee 3.0 applications
- Motion signal processing middleware for vibration analysis in time domain (speed RMS and acceleration peak) and frequency domain (FFT with programmable size, averaging, overlapping, and windowing)
- Configurable alarm and warning thresholds based
- On-board battery status monitor
- Bluetooth® Low Energy application compatible with the STBLESensClassic mobile app for Android and iOS, to display data on the app and bridging data to the Azure IoT central PnP cloud dashboard
- Zigbee mesh network example firmware with end-device and router connected to a coordinator displaying collected data on a PC serial terminal
- Sample implementations available for the STEVAL-PROTEUS1 kit
- Based on STM32Cube software development environment for STM32 microcontrollers
- Free, user-friendly license terms

The software gathers:

- the temperature from the STTS22H
- the 3-axis acceleration from the IIS3DWB and the ISM330DHCX
- the 3-axis gyroscope from the ISM330DHCX
- the battery status from the STBC02

The software manages the STSAFE-A110 for authentication and secure data management services.

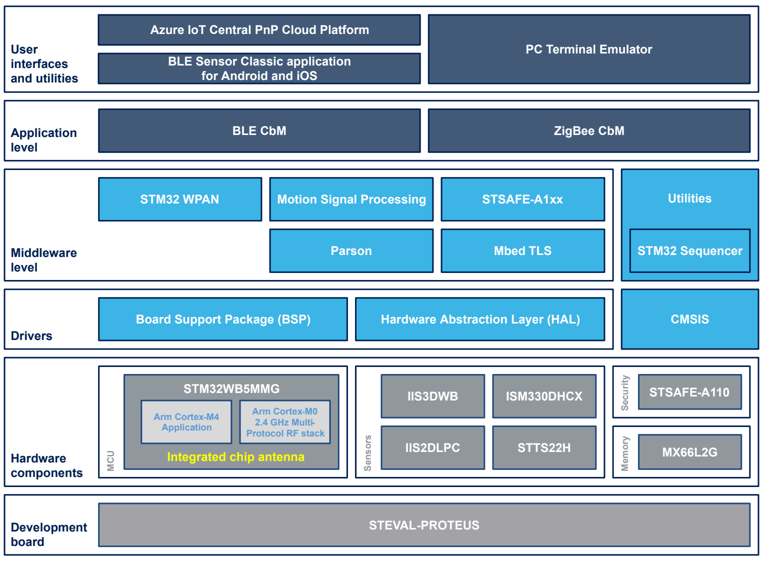
It is also possible to change the system voltage provided by the ST1PS02 via software.

## 2.1 Architecture

The firmware is based on the STM32Cube architecture.

STM32Cube reduces the development effort, time, and cost across the entire STM32 portfolio.

Figure 4. STSW-PROTEUS architecture



STM32Cube includes:

- STM32CubeMX graphical software configuration tool that allows the generation of C initialization code using graphical wizards. For further details, refer to UM1718.
- A comprehensive embedded software platform specific to each series (such as the STM32Cube for the STM32 series), which includes:
  - The hardware abstraction layer (HAL) that enables the portability between different STM32 devices via standardized API calls.
  - A collection of middleware components.
  - RF stacks such as Bluetooth® Low Energy 5.2, Zigbee 3.0.
  - The board support package (BSP) layer, based on the HAL drivers, which provides an API set to the evaluation board and third-party components.

## 2.2 Folder structure

The software package includes the following folders:

- **Doc**: contains a compiled HTML file generated from the source code, which details the software components and APIs.
- **Drivers**: contains the HAL drivers and the board-specific drivers for each supported board or hardware platform, including the on-board components and the CMSIS vendor-independent hardware abstraction layer for Arm Cortex-M processor series.
- **Middlewares**: contains libraries and protocols for WPAN, MotionSP, STSAFE, mbedTLS and Parson.
- **Projects**: contains sample applications in different folders divided per board and connectivity.
  - Each project comes with **Binary folder** that contains the ready-to-use firmware. All projects are available for:
    - IAR Embedded Workbench for ARM (IAR-EWARM).
    - Keil Microcontroller Development Kit for ARM (MDK-ARM-STM32).
    - ST Integrated Development Environment for STM32 (STM32CubeIDE).
  - In addition, each project comes with a dedicated **.ioc file** and the **.extSettings file** for the C project generation, to be used when the project is loaded into STM32CubeMX again. Furthermore, the **extSettings folder** contains the files for each supported IDE.
  - The **STM32WB_Copro_Wireless_Binaries folder** contains the coprocessor wireless firmware binaries to be used.
- **Utilities**: contains the code that is used by the WPAN middleware as the STM32 Sequencer.

The following code block shows the STSW-PROTEUS detailed folder structure.

```
\---STSW-PROTEUS
    |   readme.txt
    |   Release_Notes.html
    |
    +---Doc
    |       STSW-PROTEUS (Bluetooth).chm
    |       STSW-PROTEUS (Zigbee).chm
    |
    +---Drivers
    |   +---BSP
    |   |   +---Components
    |   |   |   +---Common
    |   |   |   +---iis2dlpc
    |   |   |   +---iis3dwb_custom
    |   |   |   +---ism330dhcx_custom
    |   |   |   +---mx6612g
    |   |   |   \---stts22h
    |   |   +---P-NUCLEO-WB55.Nucleo
    |   |   \---STEVAL-PROTEUS
    |   +---CMSIS
    |   |   +---Core
    |   |   +---Device
    |   |   |   \---ST
    |   |   |       \---STM32WBxx
    |   |   +---DSP
    |   |   |   +---Include
    |   |   |   +---Lib
    |   |   |   |   +---ARM
    |   |   |   |   +---GCC
    |   |   |   |   \---IAR
    |   |   |   \---Source
    |   |   \---Include
    |   \---STM32WBxx_HAL_Driver
    +---Middlewares
    |   +---ST
    |   |   +---STM32_MotionSP_Library
    |   |   +---STM32_WPAN
    |   |   \---STSAFE_Axx0
    |   \---Third_Party
    |       +---MbedTLS
    |       \---parson
    +---Projects
    |   +---P-NUCLEO-WB55.Nucleo
    |   |   \---Applications
    |   |       \---Zigbee
    |   |           \---ZB_CRD
    |   |               \---Binary
    |   |                       NucleoWB55_ZB_CRD_CBM_reference.bin
    |   |
    |   +---STEVAL-PROTEUS
    |   |   \---Applications
    |   |       +---BLE
    |   |       |   +---BLE_CbM
    |   |       |   |   |   .extSettings
    |   |       |   |   |   Proteus_BLE_CbM.ioc
    |   |       |   |   |
    |   |       |   |   +---Binary
    |   |       |   |   |       Proteus_BLE_CbM_FUOTA_reference.bin
    |   |       |   |   |
    |   |       |   |   +---Core
    |   |       |   |   +---EWARM
    |   |       |   |   +---extSettings
    |   |       |   |   |   +---IAR
    |   |       |   |   |   +---KEIL
    |   |       |   |   |   \---STM32CubeIDE
    |   |       |   |   +---MDK-ARM
    |   |       |   |   +---STM32CubeIDE
    |   |       |   |   \---STM32_WPAN
    |   |       |   |       +---App
    |   |       |   |       \---Target
    |   |       |   \---BLE_Ota
    |   |       |       \---Binary
    |   |       |               Proteus_BLE_FUOTA_reference.bin
    |   |       |
    |   |       \---Zigbee
    |   |           +---Zigbee_RFD
    |   |           |   |   .extSettings
    |   |           |   |   Proteus_Zigbee_RFD.ioc
    |   |           |   |
    |   |           |   +---Binary
    |   |           |   |       Proteus_Zigbee_RFD_reference.bin
    |   |           |   |
    |   |           |   +---Core
    |   |           |   +---EWARM
    |   |           |   +---extSettings
    |   |           |   |   +---IAR
    |   |           |   |   +---KEIL
    |   |           |   |   \---STM32CubeIDE
    |   |           |   +---MDK-ARM
    |   |           |   +---STM32CubeIDE
    |   |           |   \---STM32_WPAN
    |   |           |       +---App
    |   |           |       +---Target
    |   |           |       \---zcl
    |   |           \---Zigbee_RTR
    |   |               |   .extSettings
```

```
    |   |                |   Proteus_Zigbee_RTR_reference.ioc
    |   |                |
    |   |                +---Binary
    |   |                |       Proteus_Zigbee_RTR_reference.bin
    |   |                |
    |   |                +---Core
    |   |                +---EWARM
    |   |                +---extSettings
    |   |                |   +---IAR
    |   |                |   +---KEIL
    |   |                |   \---STM32CubeIDE
    |   |                +---MDK-ARM
    |   |                +---STM32CubeIDE
    |   |                \---STM32_WPAN
    |   |                    +---App
    |   |                    +---Target
    |   |                    \---zcl
    |   \---STM32WB_Copro_Wireless_Binaries
    |       \---STM32WB5x
    +---Utilities
    |   +---lpm
    |   \---sequencer
    \---_htmresc
```

## 2.3 Bluetooth$^{®}$ Low Energy project details for OTA usage

As explained in Section 1.1.1.1 Flash memory management, to use the OTA feature for the STM32WB, you have to download the related application firmware from the first main memory address. Then, download the demo application a few pages later, from the 0x08007000 address.

Since the project generated using STM32CubeMX considers the firmware starting address as the starting address of the main memory, it is necessary to make some changes to the generated project.

- Ensure that the OTA application firmware has already been downloaded into the STEVAL-PROTEUS from the first main memory address.
- Open *"$PROJ_DIR$\..\STM32_WPAN\App\ble_conf.h"* and ensure that *"BLE_CFG_OTA_REBOOT_CHAR"* is defined as "1".
- Open *"$PROJ_DIR$\..\Core\Src\system_stm32wbxx.c"* and:
  - uncomment the define *"USER_VECT_TAB_ADDRESS"*
  - ensure that *"VECT_TAB_OFFSET"* is defined as *"0x00007000U"*
- Apply the following changes:
  - IAR EWARM
    - Open *“Project->Options->Linker”* and use *"$PROJ_DIR$\stm32wb5mxx_flash_cm4_ota.icf"* as the linker configuration file.
  - Keil MDK
    - Open *“Project->Options->Target”* and set IROM1 as 0x8007000 for the start and 0x79000 for the size.
    - Open *“Project->Options->Linker”* and use *".\stm32wb5mxx_flash_cm4_ota.sct"* as the scatter file.
    - Open *“Project->Options->Linker”* and add two lines in misc controls:
      - `--keep *.o(TAG_OTA_START)`
      - `--keep *.o(TAG_OTA_END)`
  - STM32CubeIDE
    - Open *“Properties->C/C++ Build->Settings->Tool Settings->MCU GCC Linker->General”* and use "${workspace_loc:/${ProjName}/STM32WB5MMGHX_FLASH_ota.ld}" as the linker script.
- Download the generated firmware into the STEVAL-PROTEUS you are using from the 0x08007000 address.

# 3 Application scenario

The STEVAL-PROTEUS is a condition monitoring evaluation sensor node that detects developing faults within machinery, enabling you to implement a predictive maintenance program, reducing productivity loss due to an unplanned machine downtime.

The STEVAL-PROTEUS incorporates sensors to acquire temperature and vibration data to be processed, toggles a switch, recharges, and checks the battery status.

Its behavior inside the WPAN is related to the wireless protocol to be used.

## 3.1 Bluetooth® Low Energy-based application

The Bluetooth® Low Energy application involves two parts: a Bluetooth® Low Energy master and a Bluetooth® Low Energy slave.

The STEVAL-PROTEUS covers the slave (or peripheral) role. It also acts as a Bluetooth® Low Energy GATT server.

The STEVAL-PROTEUS advertises and waits for connection.

A mobile device running the STBLESensClassic app is the master (or central). It also acts as a Bluetooth® Low Energy GATT client.

The mobile device scans for devices and establishes the connection.

The STEVAL-PROTEUS sends data to the Bluetooth® Low Energy GATT client via Bluetooth® Low Energy and to a laptop via UART.

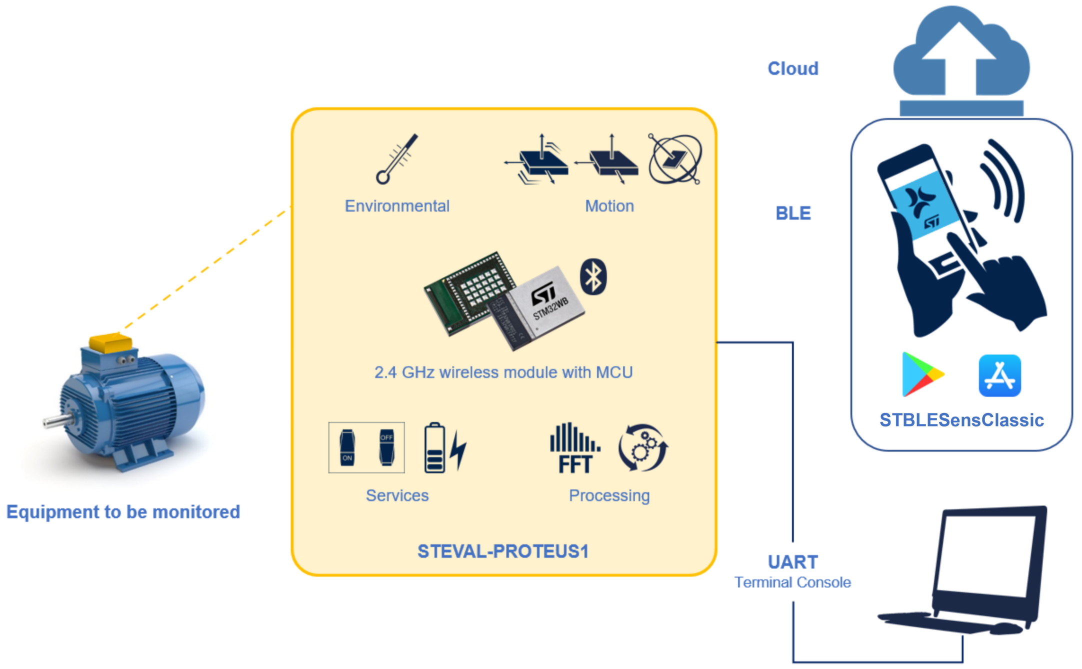
The mobile device that runs the STBLESensClassic app can forward data to the cloud.

### 3.1.1 STEVAL-PROTEUS Bluetooth® Low Energy workflow

The STEVAL-PROTEUS workflow can be summarized as follows:

1. The STEVAL-PROTEUS sensors capture the temperature and vibration data from the machinery (the asset). It executes the frequency and time domain vibration analysis.
2. The STEVAL-PROTEUS sends information via UART to the laptop connected via STLINK-V3MINI.
3. The STEVAL-PROTEUS sends the asset status inside the advertising message.
4. The STEVAL-PROTEUS accepts the Bluetooth® Low Energy connection request by a mobile device that runs the STBLESensClassic app.
5. The STEVAL-PROTEUS sends the requested information, such as environmental or motion data to be plotted as well as frequency and time domain analysis data, via Bluetooth® Low Energy to the mobile device that runs the STBLESensClassic app.
6. The STEVAL-PROTEUS can also send information about the battery status and toggle a switch as requested by the mobile device that runs the STBLESensClassic app.
7. The mobile device that runs the STBLESensClassic app can forward data to cloud.

Figure 5. **STEVAL-PROTEUS basic workflow**



## 3.1.2 STBLESensClassic app for Android and iOS

ST BLE Sensor Classic application is available for Android and iOS and shows the data exported by a Bluetooth® Low Energy device using the BlueST protocol.

All the data received by the app can be logged in CVS files and exported by e-mail.

If the firmware supports the functionality, the application can also show a serial console to exchange string messages with the board. This functionality is also used to upgrade the board firmware.

Figure 6. **QR codes for STBLESensClassic**

Android

iOS

## 3.2 Zigbee-based application

The Zigbee application involves three kinds of devices:

- The coordinator that starts a new Zigbee network.
- The router that can join on a Zigbee network and allow other devices (routers or end-devices) to join the same network.
- The end-device that can join a Zigbee network.

This demo comes with two ready-to-use projects for the STEVAL-PROTEUS acting as a router or end-device. Furthermore, with both roles, the STEVAL-PROTEUS can send data and network information over the Zigbee network as well as information via UART to the PC to which it is connected through an STLINK-V3MINI.

A binary firmware is also provided. It can be downloaded into a NUCLEO-WB55RG to be used as a coordinator. Moreover, it can receive data and network messages from other devices and send them via UART to the PC to which it is connected via the integrated ST-LINK/V2-1.

### 3.2.1 STEVAL-PROTEUS Zigbee workflow

The Zigbee workflow can be summarized as follows:

1. The coordinator is powered-on to start a Zigbee network
2. The STEVAL-PROTEUS end-devices and/or routers are powered-on to join the Zigbee network.
   a. Routers connect to the Zigbee network if they find a coordinator or another router.
   b. End-devices connect to the Zigbee network if they find a coordinator or a router.

   There is a maximum time of 3 minutes to create the network. To use the routers feature fully, connect the coordinator, all the routers, and finally all the end-devices.
   Once a STEVAL-PROTEUS has been successfully connected to the Zigbee network, a blue LED lights up.
3. During the life of a Zigbee network, the coordinator is identified by the 0x0000 address. Any other device is identified by a randomly generated address.
4. Once the network is built and all devices have successfully joined, the device neighbor list is sent to the coordinator. So, it displays on the UART, for any device: the device address, information about the joined devices displaying some network information, and the role of the device (child, parent, sibling).
5. Pressing the USR button on the STEVAL-PROTEUS (end nodes and routers) also enable the sending information about the sensors, such as environmental, battery status, frequency, and time domain status. Pressing the USR button again disables sending this information.
6. The coordinator sends the information received by any STEVAL-PROTEUS board, router, or end-device, labeled with the address of the device that sent it, to a terminal emulator.

See Section 5 Using the Zigbee application for the STEVAL-PROTEUS for further details.

# 4 Using the STEVAL-PROTEUS with the ST BLE Sensor Classic application

## 4.1 Prerequisites

The STEVAL-PROTEUS acts as a Bluetooth® Low Energy slave to be connected to a Bluetooth® Low Energy master, which is a mobile device that runs the STBLESensClassic app.

Download the STBLESensClassic app from Google Play for Android and from App Store for iOS.

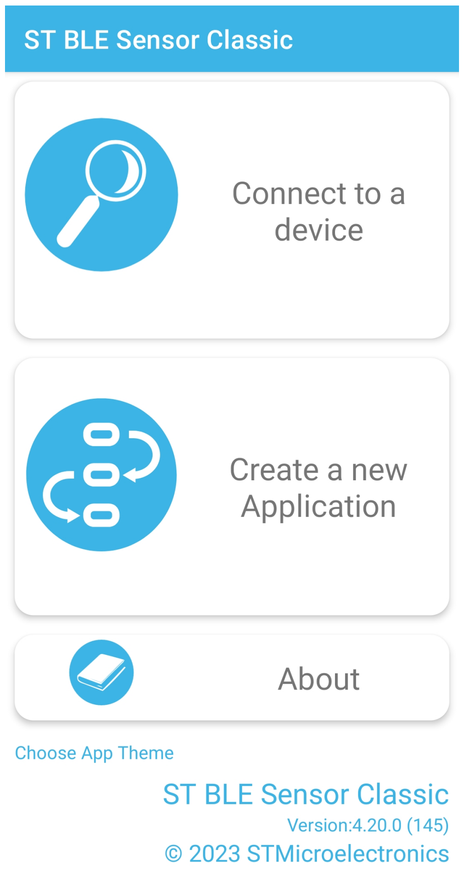
Install the STBLESensClassic on your smartphone before using oit with the STEVAL-PROTEUS.

## 4.2 Bluetooth® Low Energy advertising

When the STEVAL-PROTEUS has been successfully programmed (see 6.2) and has been switched on, it acts as a Bluetooth® Low Energy peripheral device and sends advertising packets to establish a connection with a Bluetooth® Low Energy central device.

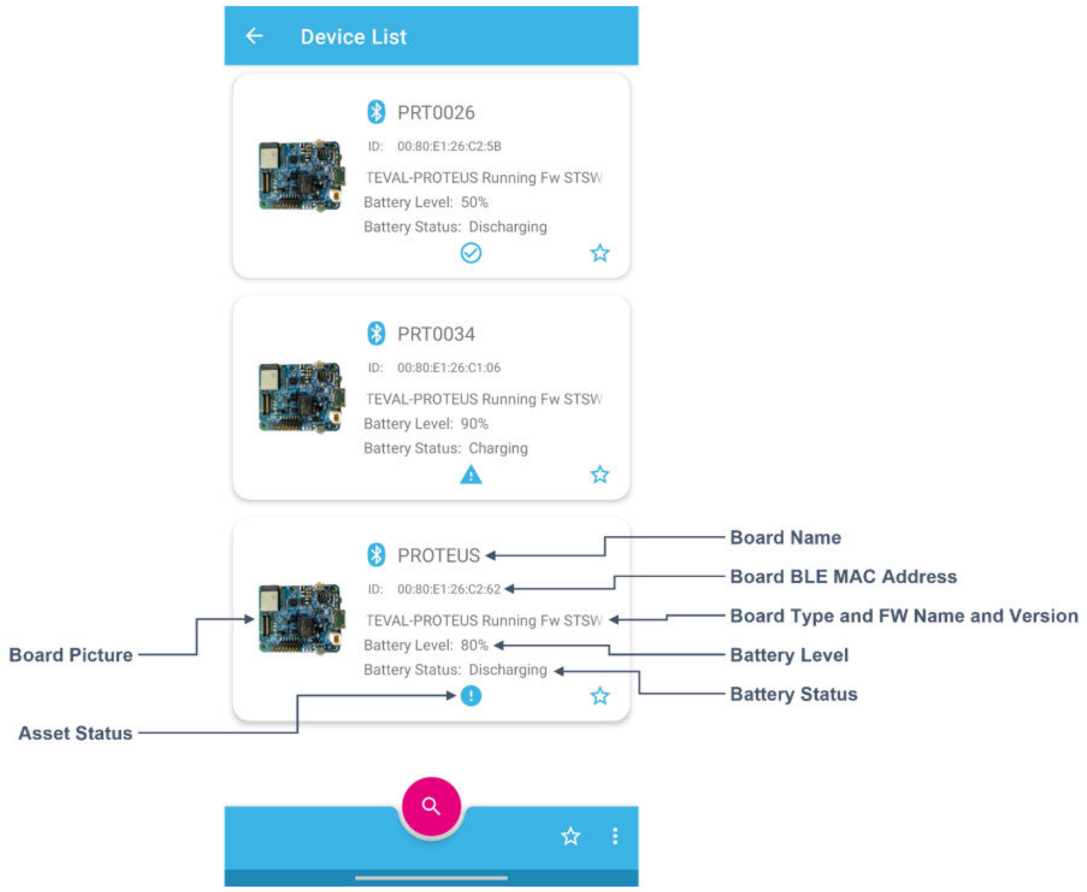
Launching the STBLESensClassic app, you can try to connect a device.

Figure 7. Main page of the STBLESensClassic app for Android



Choosing “Connect one Device”, a list of the connectable Bluetooth® Low Energy peripheral devices appears. Details are provided with each listed board.

Figure 8. Device list page of the STBLESensClassic app for Android



PRT0026
ID: 00:80:E1:26:C2:5B
TEVAL-PROTEUS Running Fw STSW
Battery Level: 50%
Battery Status: Discharging

PRT0034
ID: 00:80:E1:26:C1:06
TEVAL-PROTEUS Running Fw STSW
Battery Level: 90%
Battery Status: Charging

PROTEUS
ID: 00:80:E1:26:C2:62
TEVAL-PROTEUS Running Fw STSW
Battery Level: 80%
Battery Status: Discharging

Periodically, the STEVAL-PROTEUS executes frequency and time domain vibration analysis.

For the implemented thresholds, the results are summarized in three events (*"No alarm"*, *"Warning"* and *"Alarm"*) that are sent into the Bluetooth® Low Energy advertising packet as *"Asset status"*.

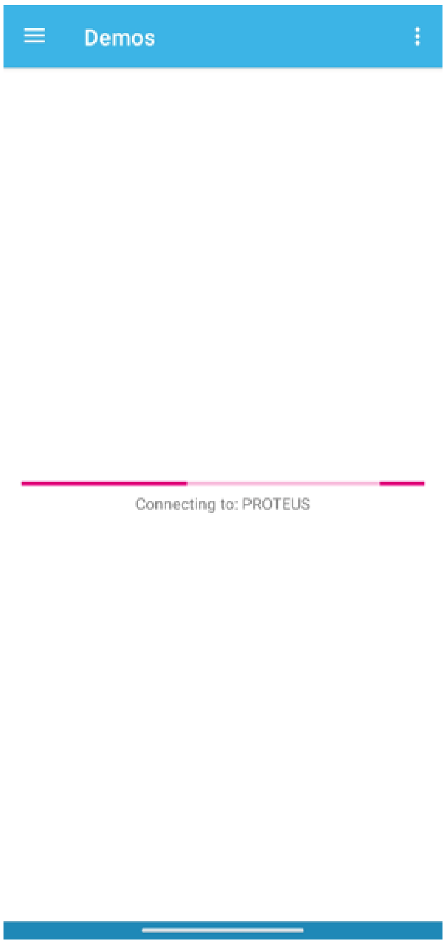
Table 2. Asset status icon

| No alarm | Warning | Alarm |
|---|---|---|
| ✓ | ⚠ | ❗ |

## 4.3 How to connect a Bluetooth® Low Energy device

You can connect a device by simply touching it. After that, you can retrieve all the information that the STBLESensClassic app can manage for the selected board.

Figure 9. Connecting a device



## 4.4 How to choose the demo application

Touching the three lines at the top left side, you can choose the demo to use.

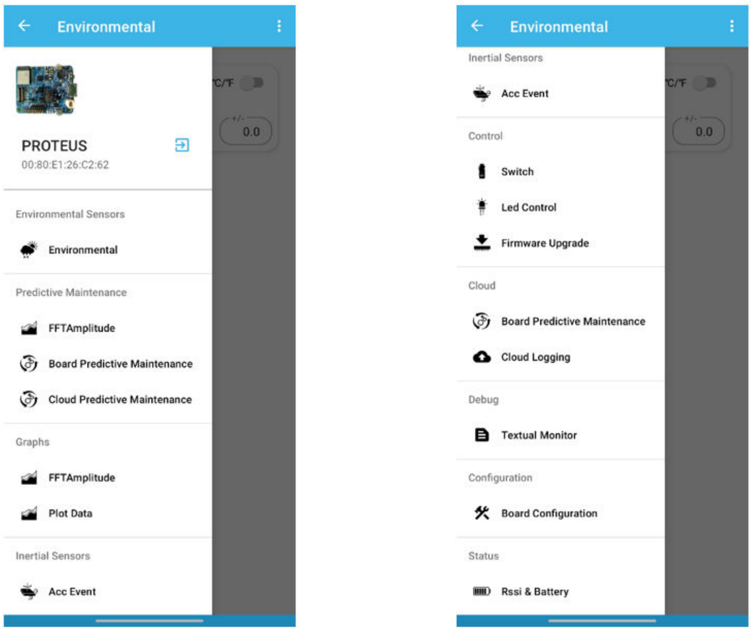


Figure 10. Choosing the demo

You can also choose the demo by tapping *“back”* on any page.

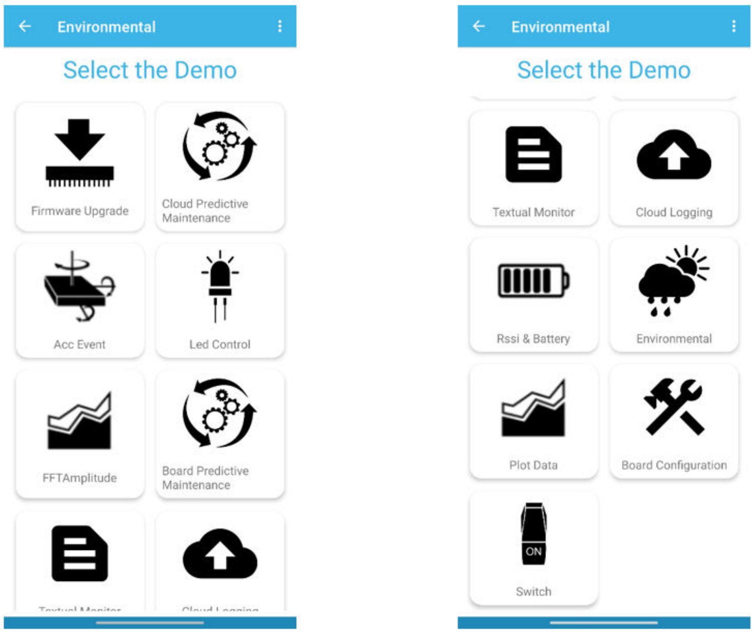


Figure 11. Choosing the demo by tapping "back"

### 4.4.1 Environmental demo

As soon as the STEVAL-PROTEUS is connected, the *"Environmental"* page is shown. You can also choose this demo by touching *"Environmental"* in the STBLESensClassic app menu.

Figure 12. Environmental demo

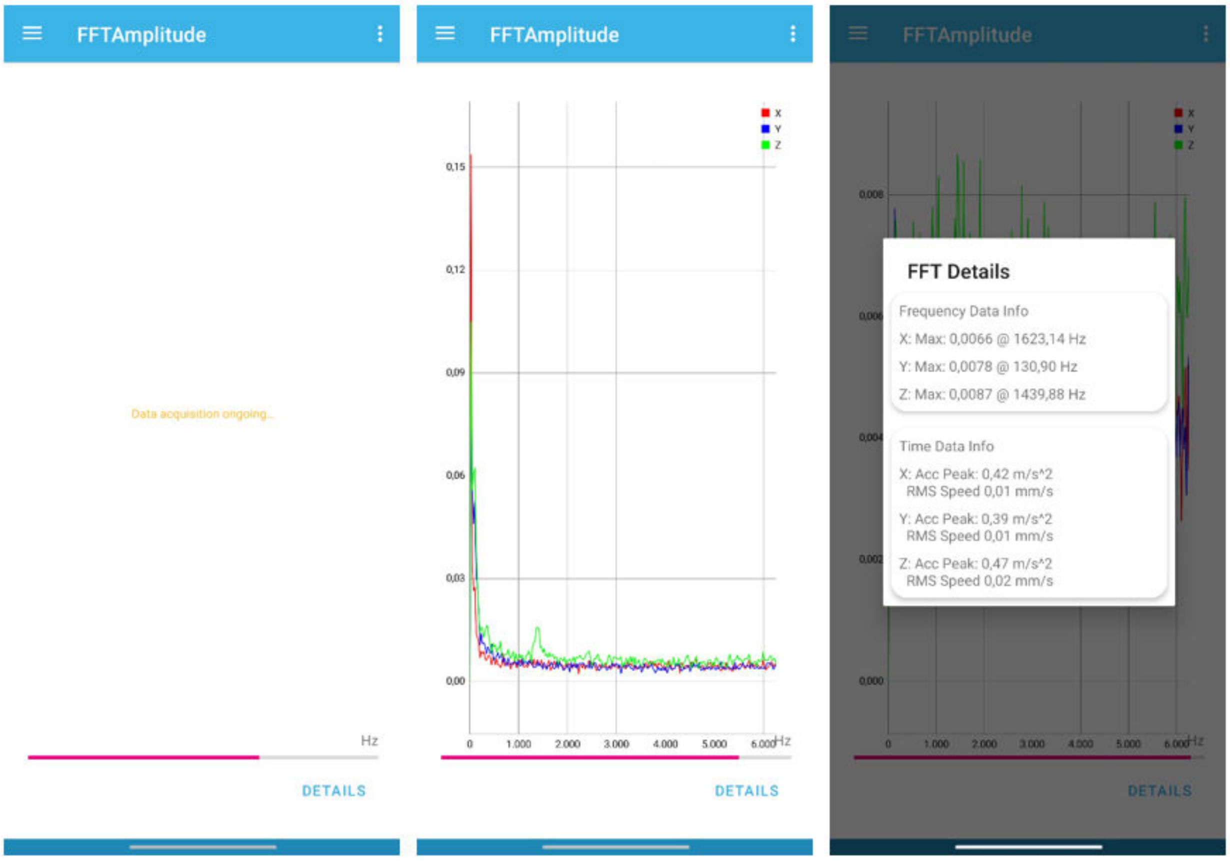
Environmental
21,8 [℃]
℃/℉
+/-
0.0

The STEVAL-PROTEUS periodically gets the temperature value from the STTS22H and sends it to the mobile device that shows the numeric value.

## 4.4.2 FFT amplitude demo

Touching *"FFTAmplitude"* in the STBLESensClassic app menu, the STEVAL-PROTEUS retrieves the acceleration data from IIS3DWB (by default) or from IISM330DHCX (selectable via firmware). Then, it performs the vibration analysis in the frequency and time domains. Finally, it sends the results to the mobile device to show them on the screen.

Figure 13. **FFT amplitude demo**



FFT Details

Frequency Data Info

X: Max: 0,0066 @ 1623,14 Hz

Y: Max: 0,0078 @ 130,90 Hz

Z: Max: 0,0087 @ 1439,88 Hz

Time Data Info

X: Acc Peak: 0,42 m/s^2
RMS Speed 0,01 mm/s

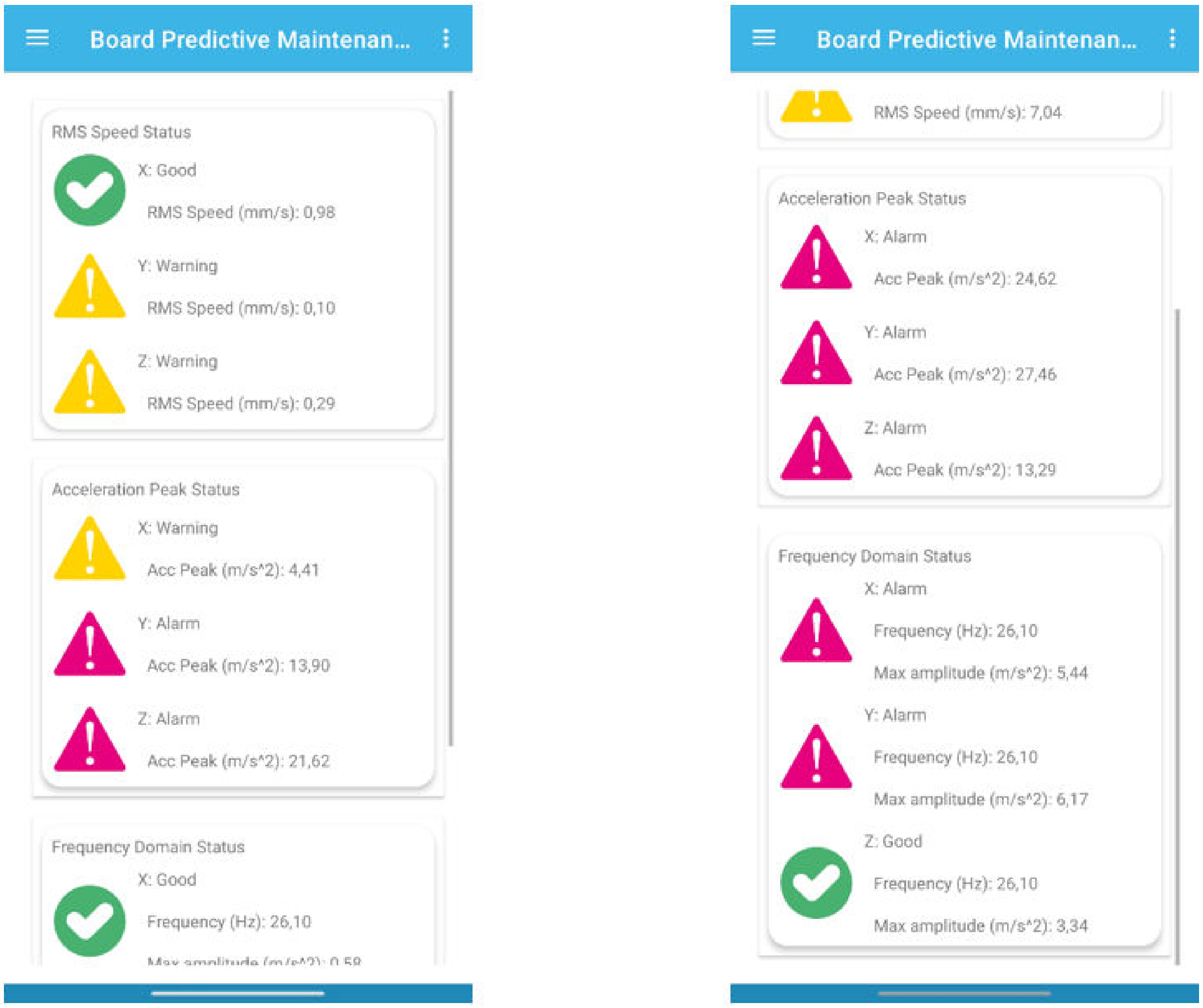
Y: Acc Peak: 0,39 m/s^2
RMS Speed 0,01 mm/s

Z: Acc Peak: 0,47 m/s^2
RMS Speed 0,02 mm/s

### 4.4.3 Board predictive maintenance demo

Touching *"Board Predictive Maintenance"* in the STBLESensClassic app menu, the STEVAL-PROTEUS retrieves the acceleration data from IIS3DWB (by default) or from IISM330DHCX (selectable via firmware): Then, it performs the vibration analysis in the frequency and time domains and compare the results with the thresholds already defined in the firmware. The asset status is sent to the mobile device to show it on the screen.

Figure 14. **Board predictive maintenance demo**



Board Predictive Maintenan...

RMS Speed Status
X: Good
RMS Speed (mm/s): 0,98
Y: Warning
RMS Speed (mm/s): 0,10
Z: Warning
RMS Speed (mm/s): 0,29

Acceleration Peak Status
X: Warning
Acc Peak (m/s^2): 4,41
Y: Alarm
Acc Peak (m/s^2): 13,90
Z: Alarm
Acc Peak (m/s^2): 21,62

Frequency Domain Status
X: Good
Frequency (Hz): 26,10
Max amplitude (m/s^2): 0,58

Board Predictive Maintenan...

RMS Speed (mm/s): 7,04

Acceleration Peak Status
X: Alarm
Acc Peak (m/s^2): 24,62
Y: Alarm
Acc Peak (m/s^2): 27,46
Z: Alarm
Acc Peak (m/s^2): 13,29

Frequency Domain Status
X: Alarm
Frequency (Hz): 26,10
Max amplitude (m/s^2): 5,44
Y: Alarm
Frequency (Hz): 26,10
Max amplitude (m/s^2): 6,17
Z: Good
Frequency (Hz): 26,10
Max amplitude (m/s^2): 3,34

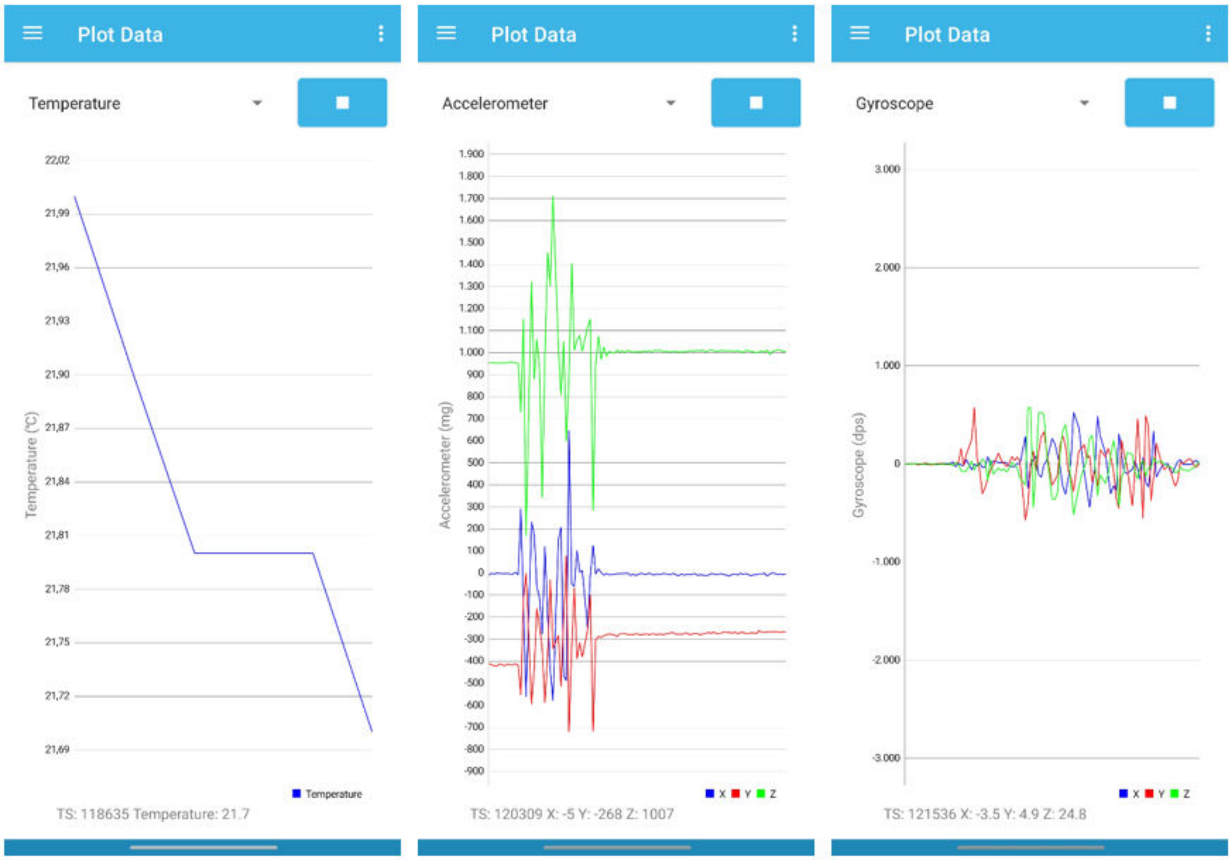
### 4.4.4 Plot data demo

Touching *"Plot Data"* in the STBLESensClassic app menu, the STEVAL-PROTEUS can periodically obtain environmental data from the STTS22H and vibration data from IISM330DHCX (by default for the accelerometer and gyroscope) or from IIS3DWB (selectable via firmware for the accelerometer only).

You can choose the type of data from the drop-down menu. You can also start or stop the acquisition using the related button.

The data are sent to the mobile device that plots them in the related page.

Figure 15. **Plotting data**



TS: 118635 Temperature: 21.7

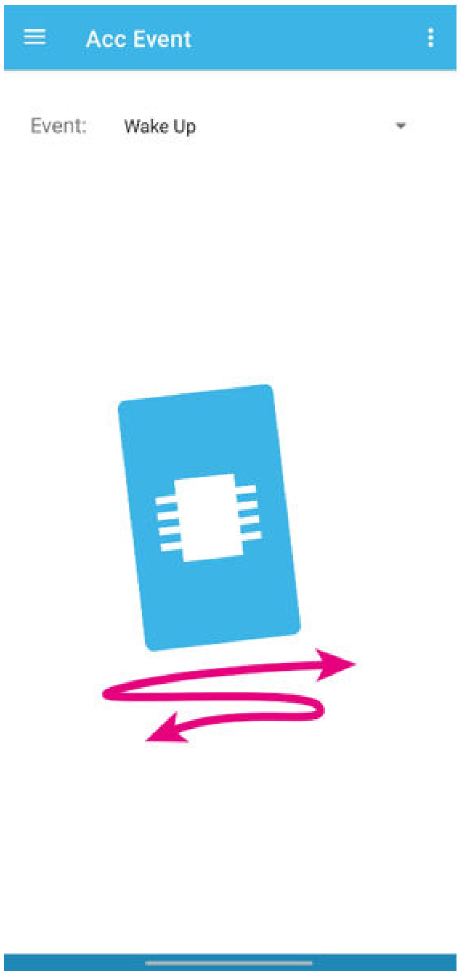
TS: 120309 X: -5 Y: -268 Z: 1007

TS: 121536 X: -3.5 Y: 4.9 Z: 24.8

### 4.4.5 Acceleration event demo

Touching *"Acc Event"* in the STBLESensClassic app menu, if the IIS2DLPC detects a movement, the STEVAL-PROTEUS sends it to the mobile device, which shows an animation of the accelerometer pictogram.

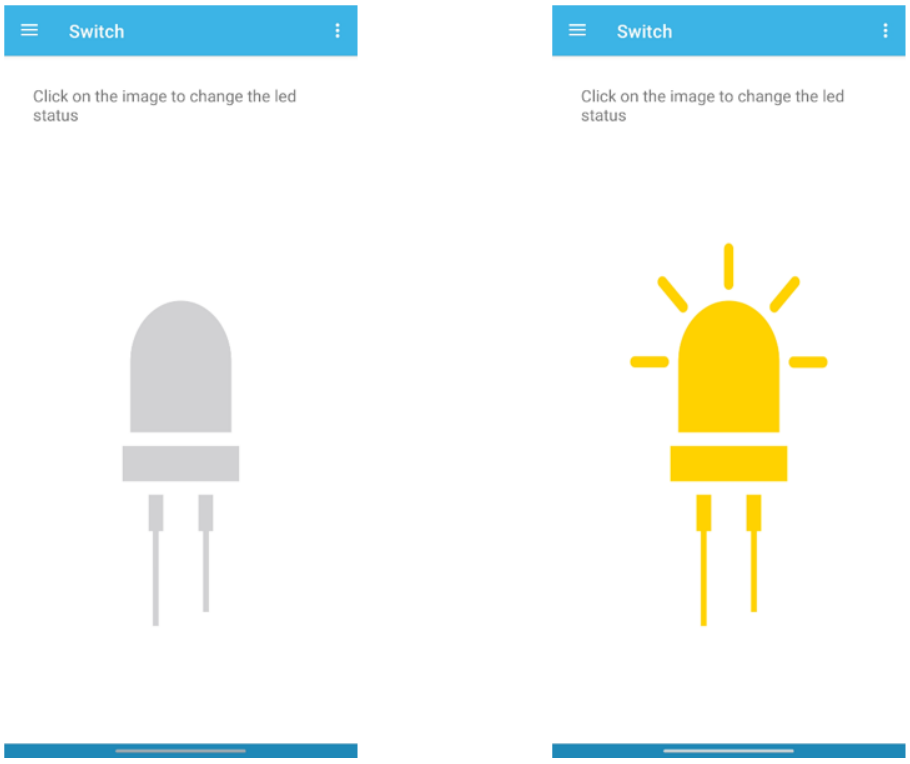
Figure 16. Acceleration event demo



### 4.4.6 Switch demo

Touching *"Switch"* in the STBLESensClassic app menu, you can act on the status of STEVAL-PROTEUS blue LED. Touching the LED pictogram, the mobile device sends a command to the STEVAL-PROTEUS, which switches the state of the dedicated LED.

Figure 17. Switch demo


Switch
Click on the image to change the led status
Switch
Click on the image to change the led status

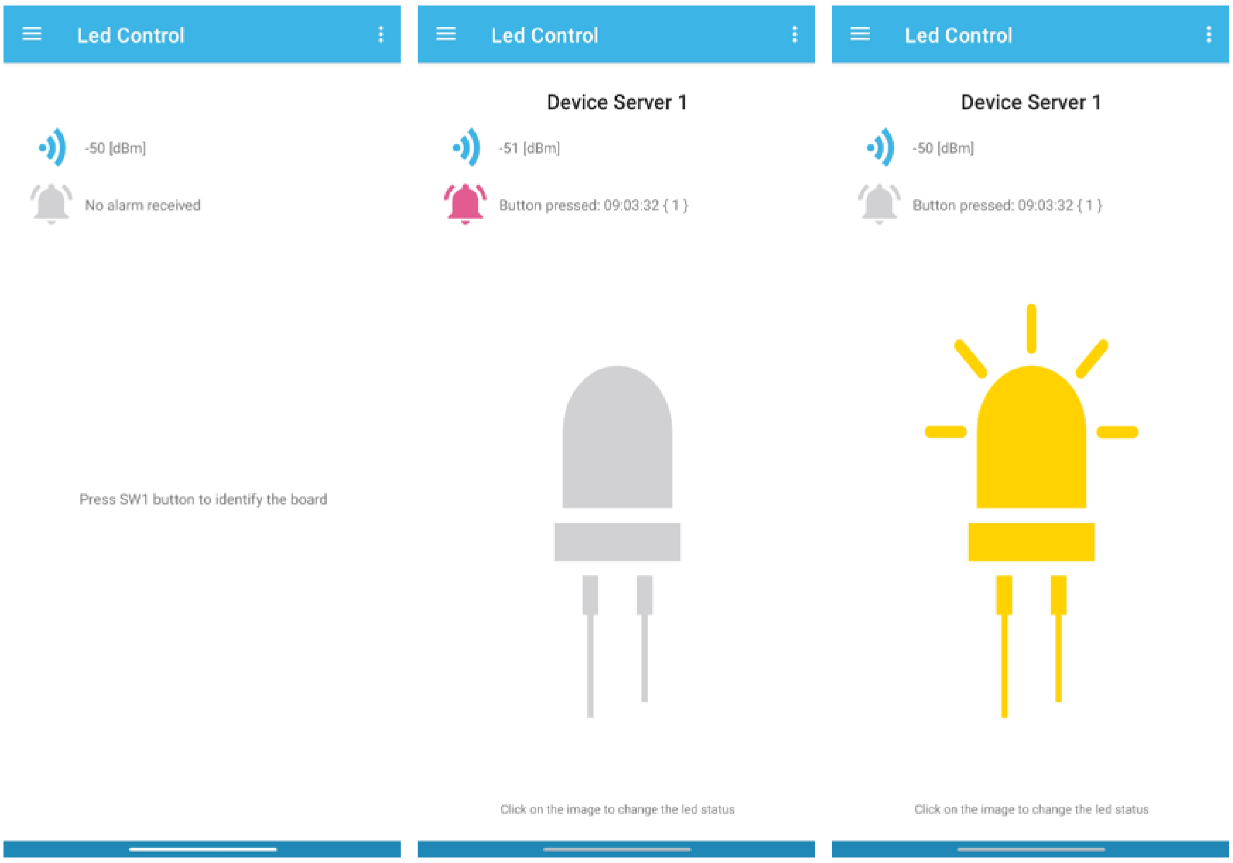
### 4.4.7 LED control demo

Touching *"Led Control"* in the STBLESensClassic app menu, the STEVAL-PROTEUS can send a notification to or receive a command from the mobile device. It also shows the RSSI.

By pressing the SW1 button, the STEVAL-PROTEUS sends a notification to the mobile device. The bell pictogram turns red for a while. The message time is displayed with the received data in curly brackets.

Like the *"Switch demo"*, by touching the LED pictogram, the mobile device sends a command to the STEVAL-PROTEUS, which switches the state of the dedicated LED.

**Figure 18.** **LED control demo**


Led Control
-50 [dBm]
No alarm received
Press SW1 button to identify the board
Led Control
Device Server 1
-51 [dBm]
Button pressed: 09:03:32 { 1 }
Click on the image to change the led status
Led Control
Device Server 1
-50 [dBm]
Button pressed: 09:03:32 { 1 }
Click on the image to change the led status

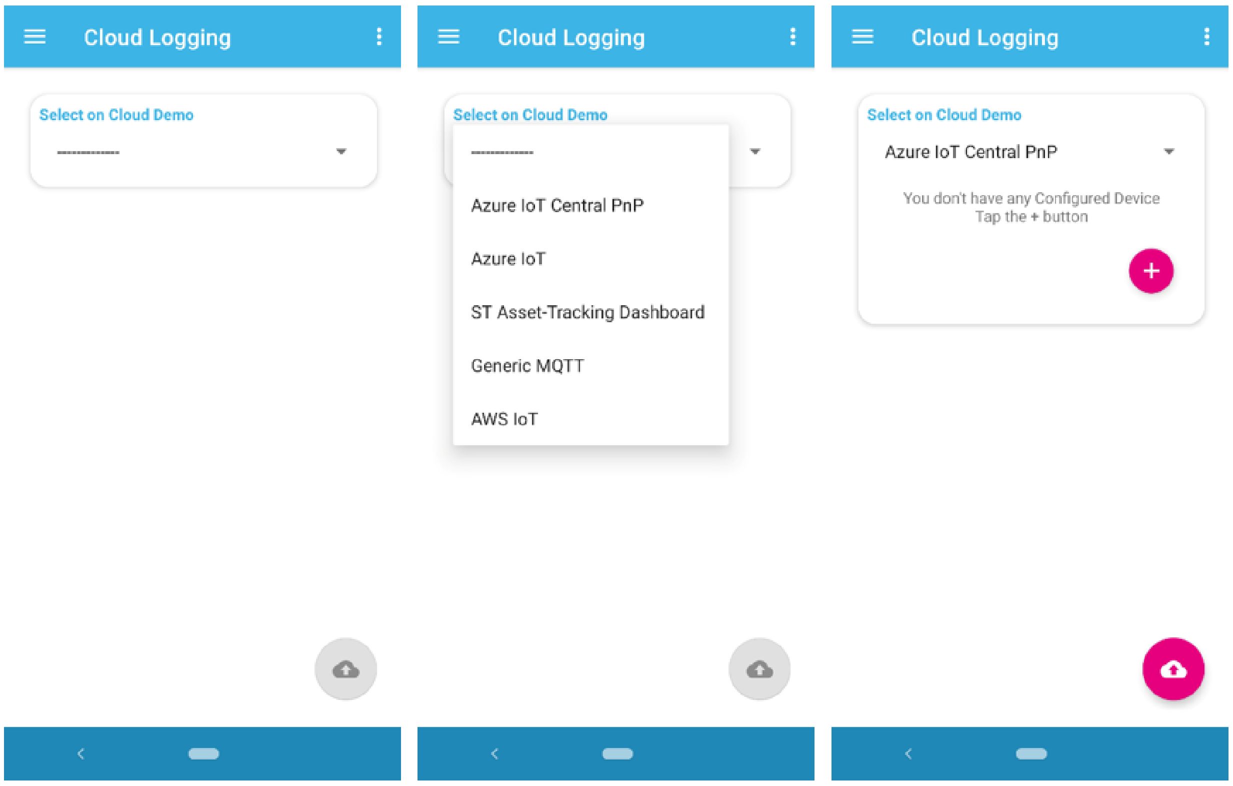
### 4.4.8 Cloud logging demo

Touching *"Cloud Logging"* in the STBLESensClassic app menu, the mobile device can forward the data received by the connected STEVAL-PROTEUS to the Azure IoT Central cloud platform.

You must have a Microsoft account. It can also be created during the first access to the Azure IoT Central cloud platform.

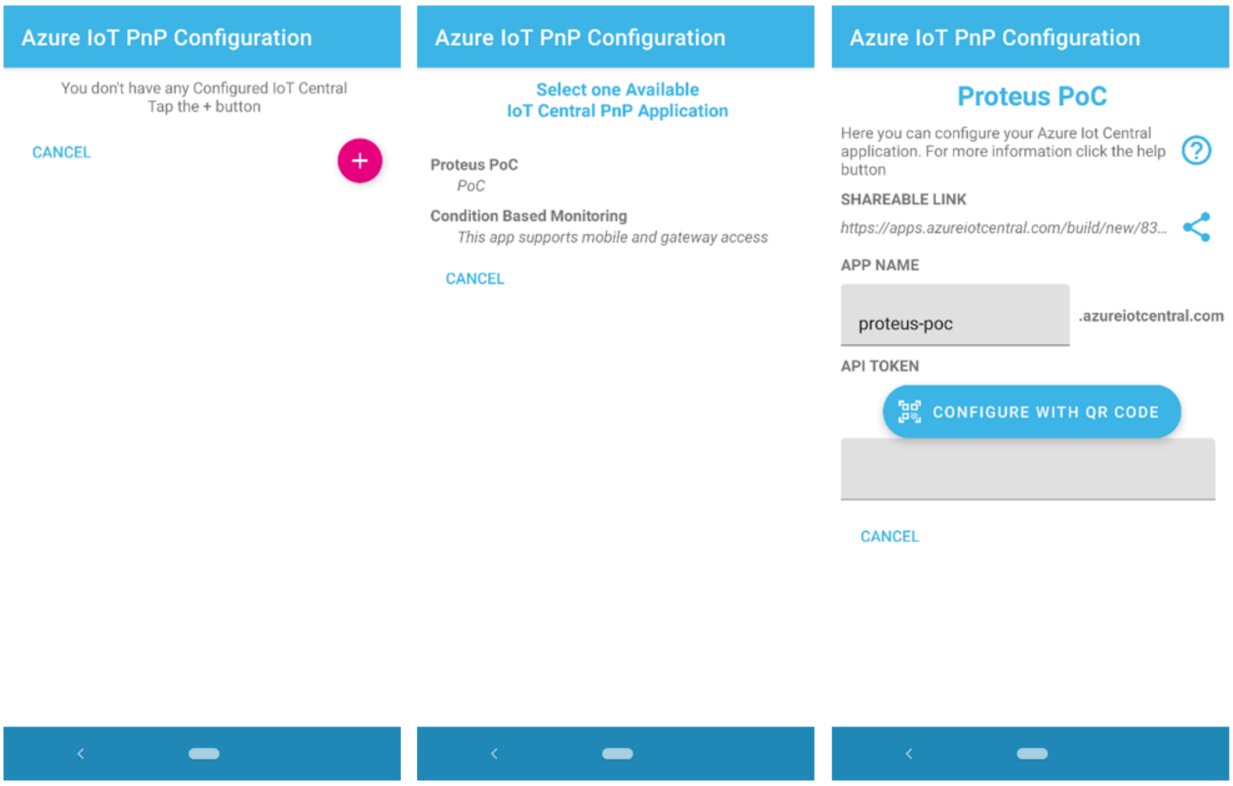
Select the cloud platform you intend to use. It must be *"Azure IoT Central PnP"*.

Figure 19. **Cloud logging**



When using this demo for the first time, there is no configured device, so you have to add a new one.

You must choose the *"Proteus PoC"* application while configuring the Azure IoT PnP.



Azure IoT PnP Configuration

You don't have any Configured IoT Central
Tap the + button

CANCEL

Azure IoT PnP Configuration

Select one Available
IoT Central PnP Application

Proteus PoC
*PoC*

Condition Based Monitoring
*This app supports mobile and gateway access*

CANCEL

Azure IoT PnP Configuration

Proteus PoC

Here you can configure your Azure Iot Central application. For more information click the help button

SHAREABLE LINK
*https://apps.azureiotcentral.com/build/new/83...*

APP NAME
proteus-poc .azureiotcentral.com

API TOKEN

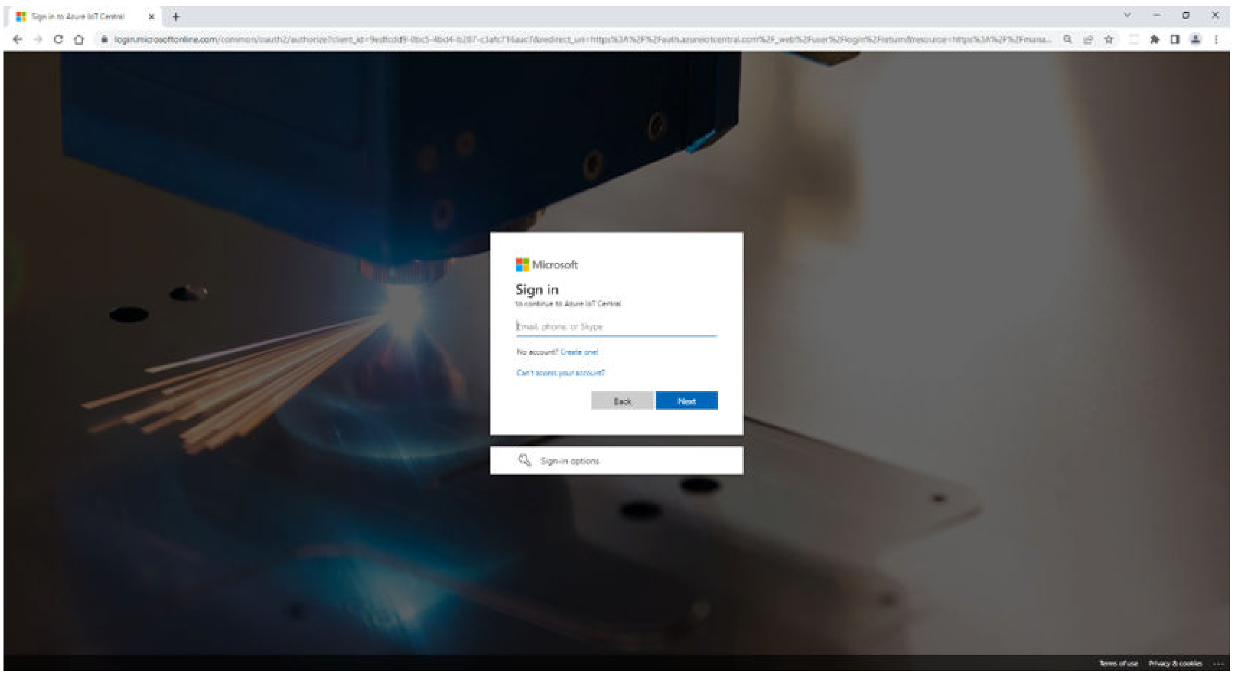
CONFIGURE WITH QR CODE

CANCEL

Figure 20. **Choosing the Proteus PoC application**

After that, you must use the proposed shareable link. So, you need a PC, which can be connected to the Internet, to continue the procedure.

As soon as the web page is opened, your credentials are requested.



Microsoft

Sign in
to continue to Azure IoT Central

Email, phone, or Skype

No account? Create one!

Can't access your account?

Back Next

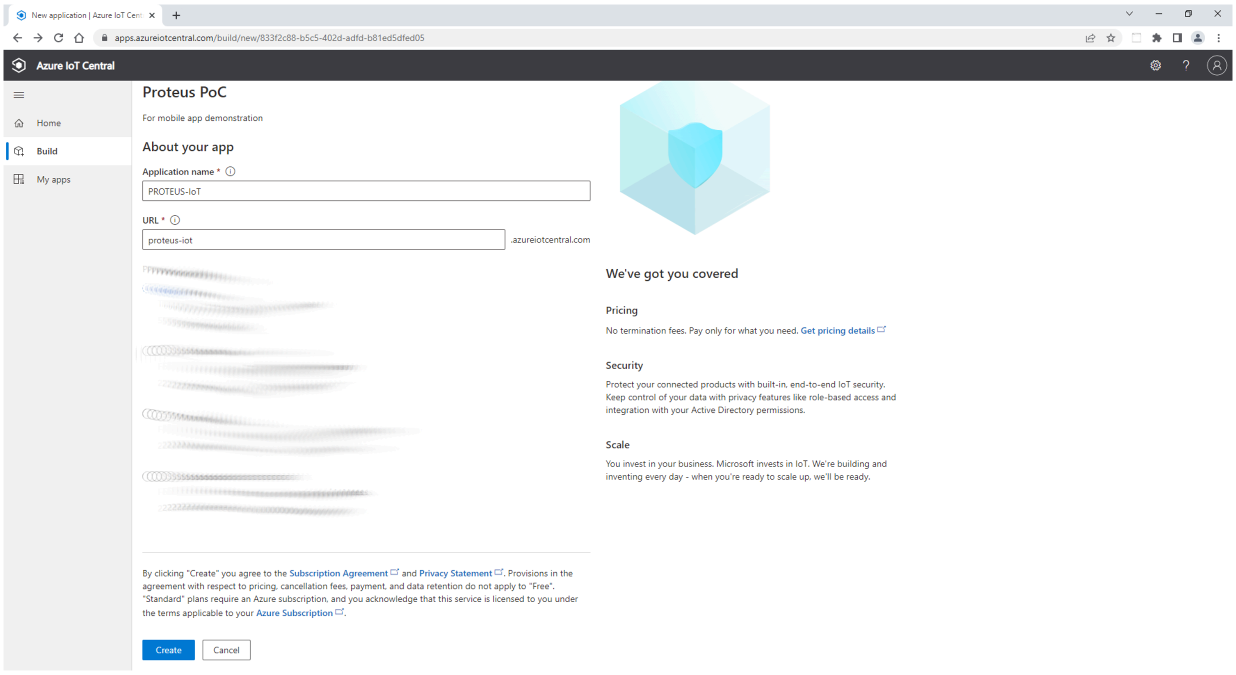
Sign-in options

Terms of use Privacy & cookies ...

Figure 21. **Azure platform - sign in**

It is mandatory to have an Azure IoT Central account.

As soon as the credentials have been accepted, you can build your application to be used with your mobile phone.


New application | Azure IoT Central
apps.azureiotcentral.com/build/new/833f2c88-b5c5-402d-adfd-b81ed5dfed05
Azure IoT Central
Home
Build
My apps
Create
Cancel

Proteus PoC

For mobile app demonstration

About your app

Application name *: PROTEUS-IoT

URL *: proteus-iot .azureiotcentral.com

By clicking "Create" you agree to the Subscription Agreement and Privacy Statement. Provisions in the agreement with respect to pricing, cancellation fees, payment, and data retention do not apply to "Free". "Standard" plans require an Azure subscription, and you acknowledge that this service is licensed to you under the terms applicable to your Azure Subscription.

We've got you covered

Pricing

No termination fees. Pay only for what you need. Get pricing details

Security

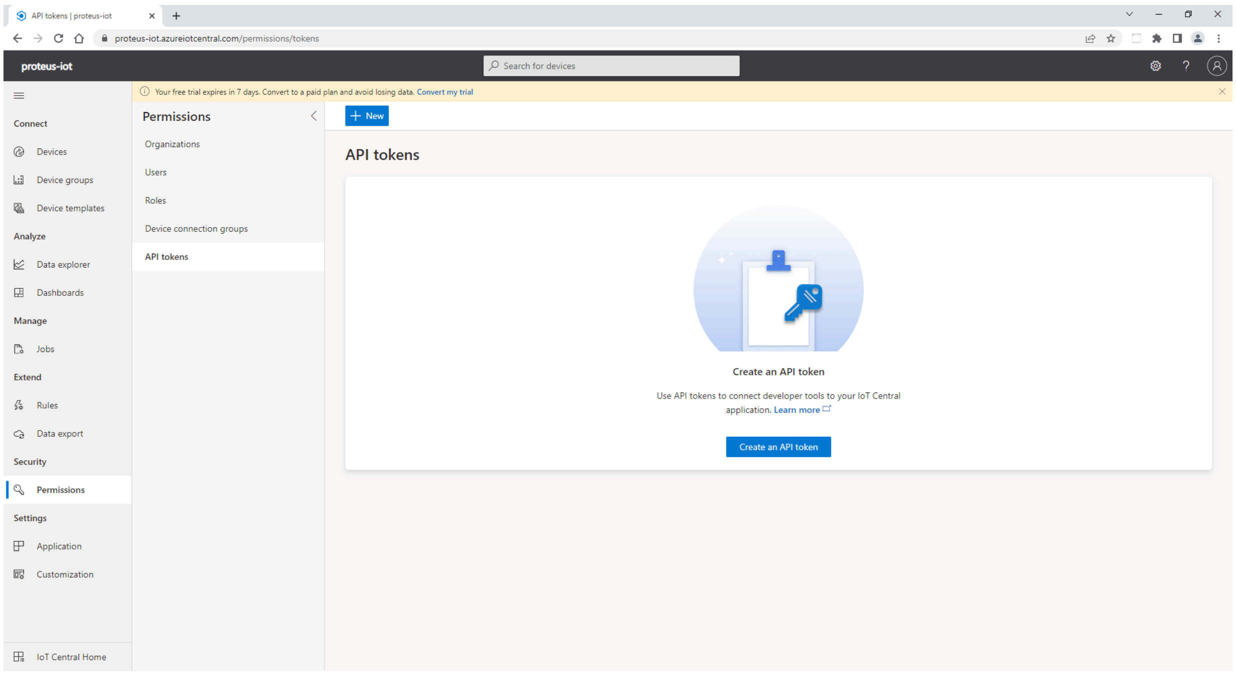
Protect your connected products with built-in, end-to-end IoT security. Keep control of your data with privacy features like role-based access and integration with your Active Directory permissions.

Scale

You invest in your business. Microsoft invests in IoT. We're building and inventing every day - when you're ready to scale up, we'll be ready.

Figure 22. **Building your application**

You must create an API token from [**Security**]>[**Permissions**]>[**API tokens**].



Your free trial expires in 7 days. Convert to a paid plan and avoid losing data. Convert my trial

API tokens

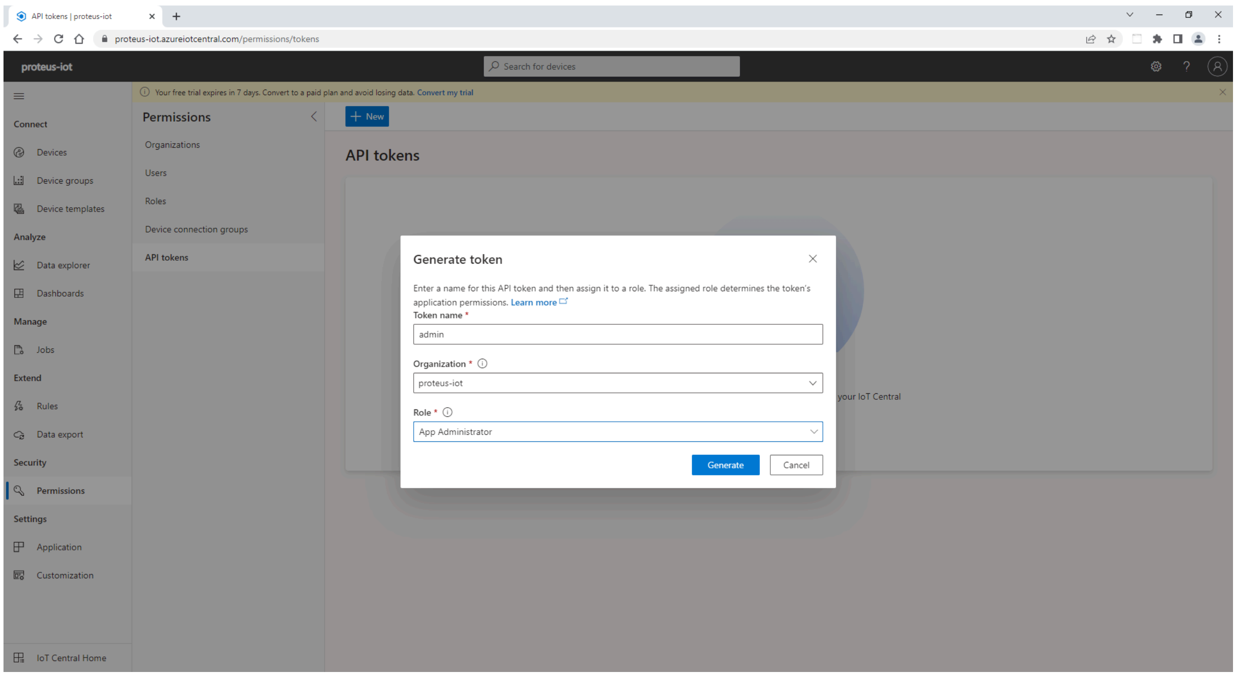
Create an API token

Use API tokens to connect developer tools to your IoT Central application. Learn more

Figure 23. **Creating an API token (1 of 2)**

Just click on *"Create an API token"* to generate a token.

Figure 24. **Creating an API token (2 of 2)**



Generate token

Enter a name for this API token and then assign it to a role. The assigned role determines the token's application permissions. Learn more

Token name *: admin

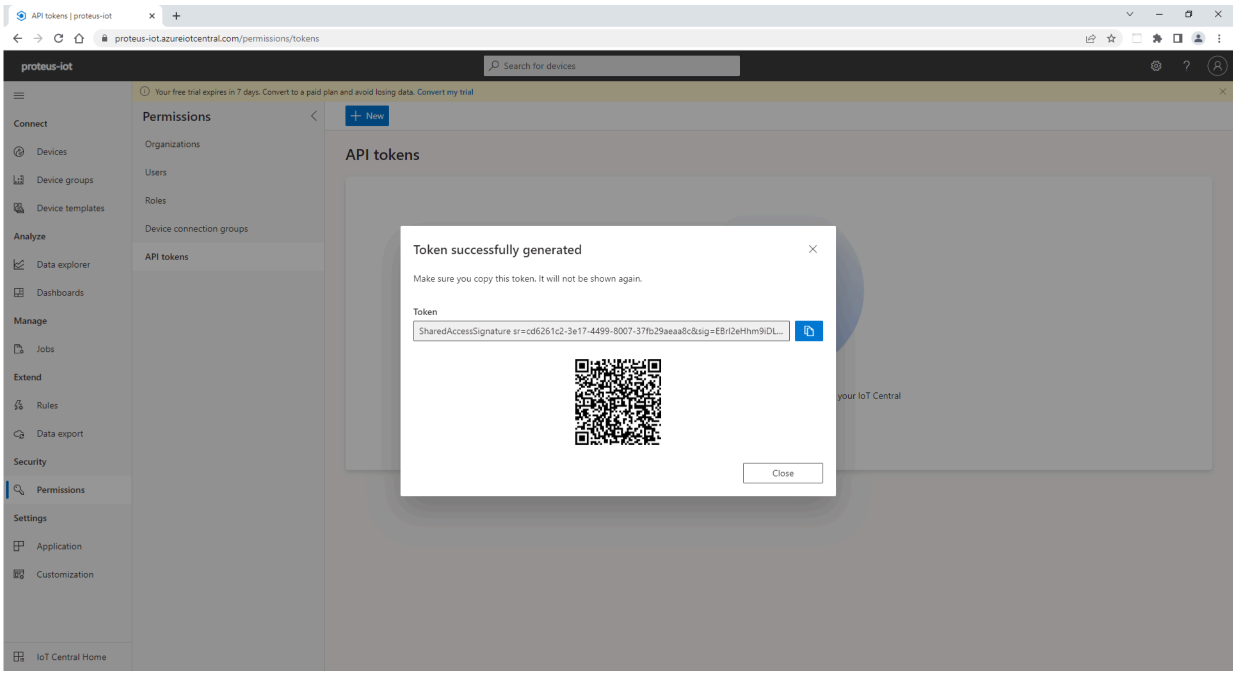
Organization *: proteus-iot

Role *: App Administrator

You must fill the requested field and click on *"Generate"*.

The new token is shown both as text and as QR code.

Figure 25. **Token creation**



Token successfully generated

Make sure you copy this token. It will not be shown again.

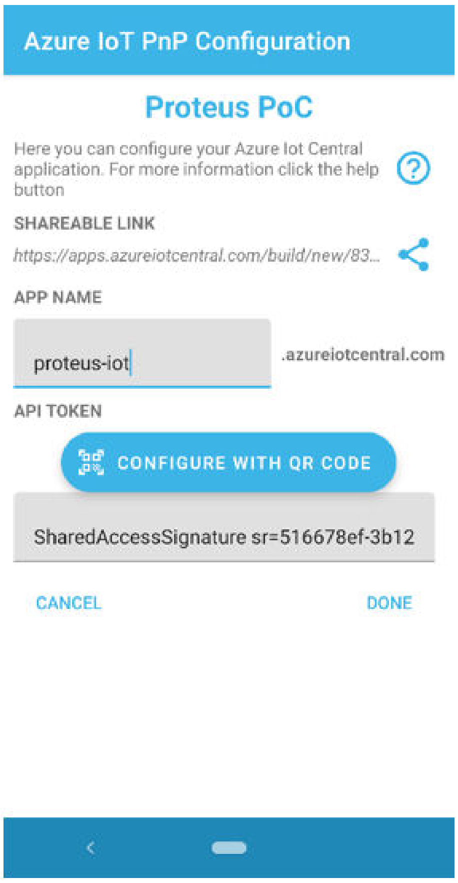
Token: SharedAccessSignature sr=cd6261c2-3e17-4499-8007-37fb29aeaa8c&sig=EBrl2eHhm9iDL...

In your mobile phone, adjust the *"APP NAME"* with the URL related the application already created (see Figure 22. Building your application).

Then, tap on *"CONFIGURE WITH QR CODE"* and scan the QR code related to the token already created (see Figure 25. Token creation).

Finally, tap on *"DONE"* to complete the configuration procedure for your Azure IoT Central application.

Figure 26. Configuration with QR code



Azure IoT PnP Configuration

Proteus PoC

Here you can configure your Azure Iot Central application. For more information click the help button

SHAREABLE LINK

https://apps.azureiotcentral.com/build/new/83...

APP NAME

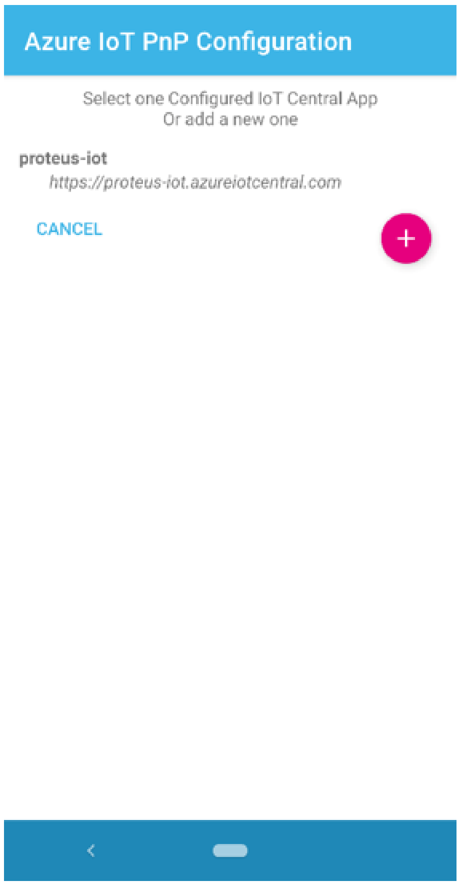
proteus-iot .azureiotcentral.com

API TOKEN

SharedAccessSignature sr=516678ef-3b12

You must select the configured IoT Central application to add a new device.

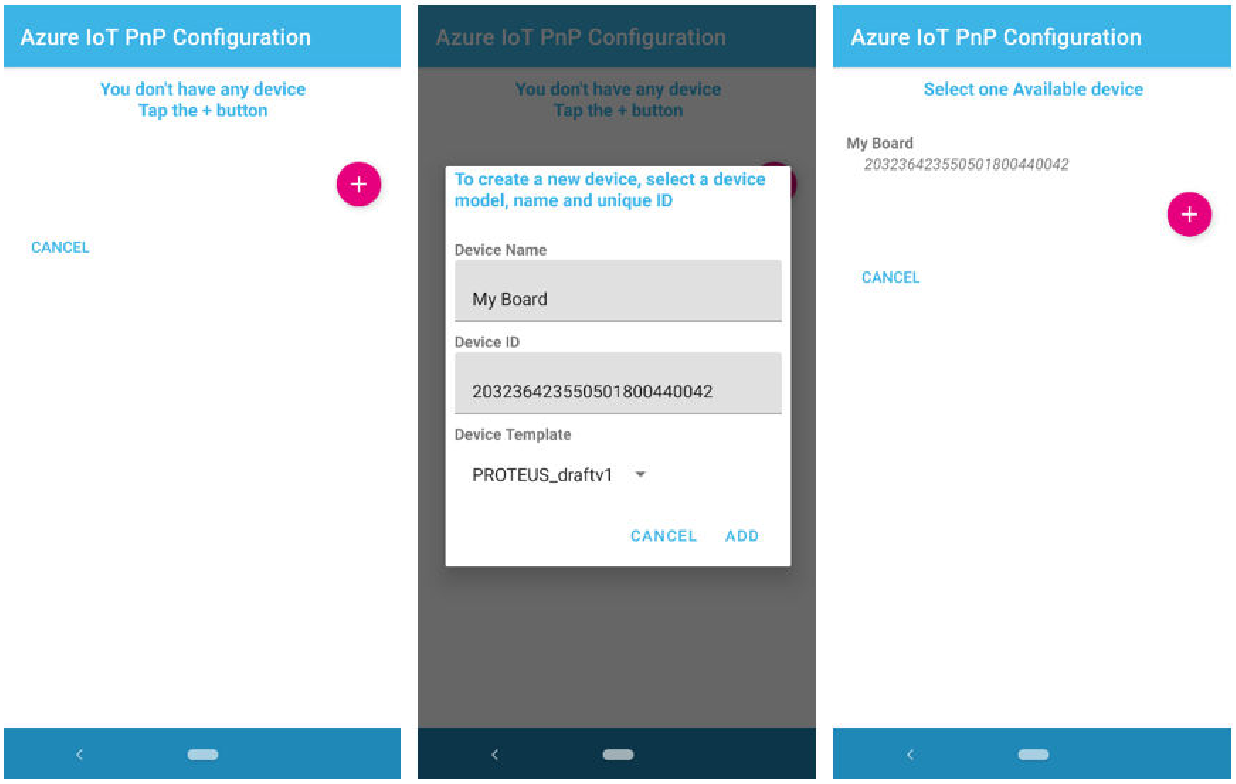
Figure 27. Adding a new device (1 of 2)


Azure IoT PnP Configuration
Select one Configured IoT Central App
Or add a new one
proteus-iot
https://proteus-iot.azureiotcentral.com
CANCEL
+

Fill the form to create the new device and add it.

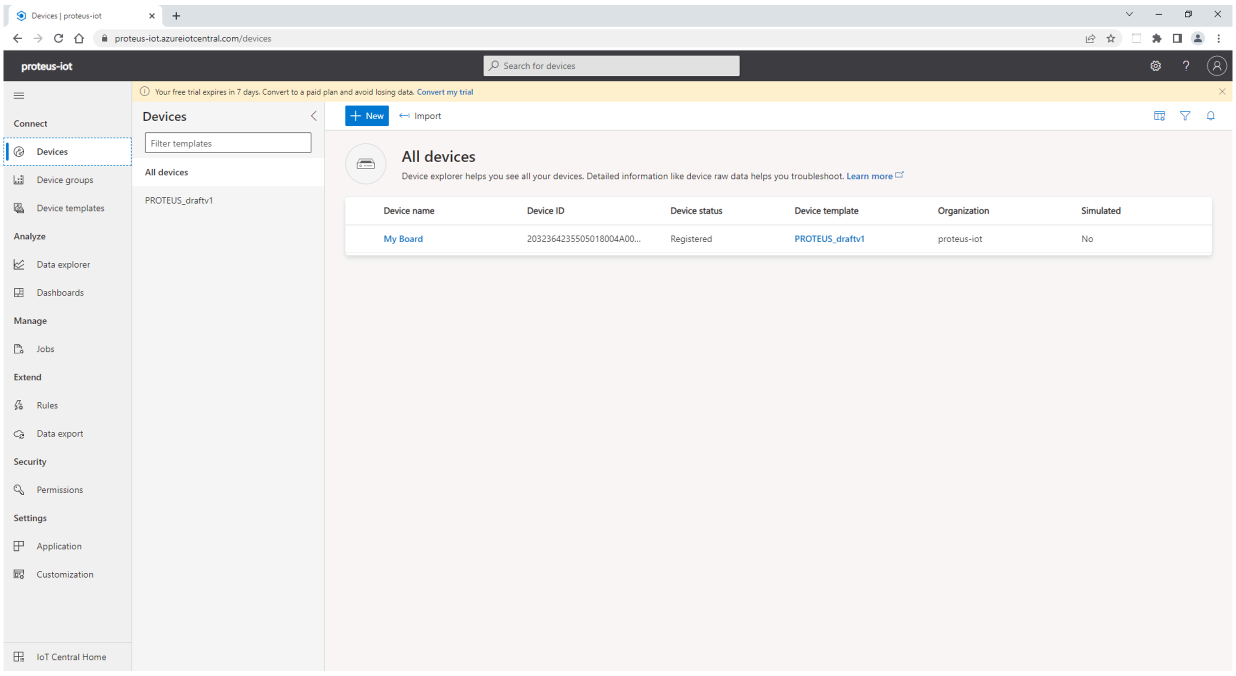
The new device is shown among the available ones.

Figure 28. Adding a new device (2 of 2)


Azure IoT PnP Configuration
You don't have any device
Tap the + button
CANCEL
Azure IoT PnP Configuration
You don't have any device
Tap the + button
To create a new device, select a device model, name and unique ID
Device Name
My Board
Device ID
203236423550501800440042
Device Template
PROTEUS_draftv1
CANCEL
ADD
Azure IoT PnP Configuration
Select one Available device
My Board
203236423550501800440042
CANCEL

In the web page you can see the new device.

Figure 29. New device



Your free trial expires in 7 days. Convert to a paid plan and avoid losing data. Convert my trial

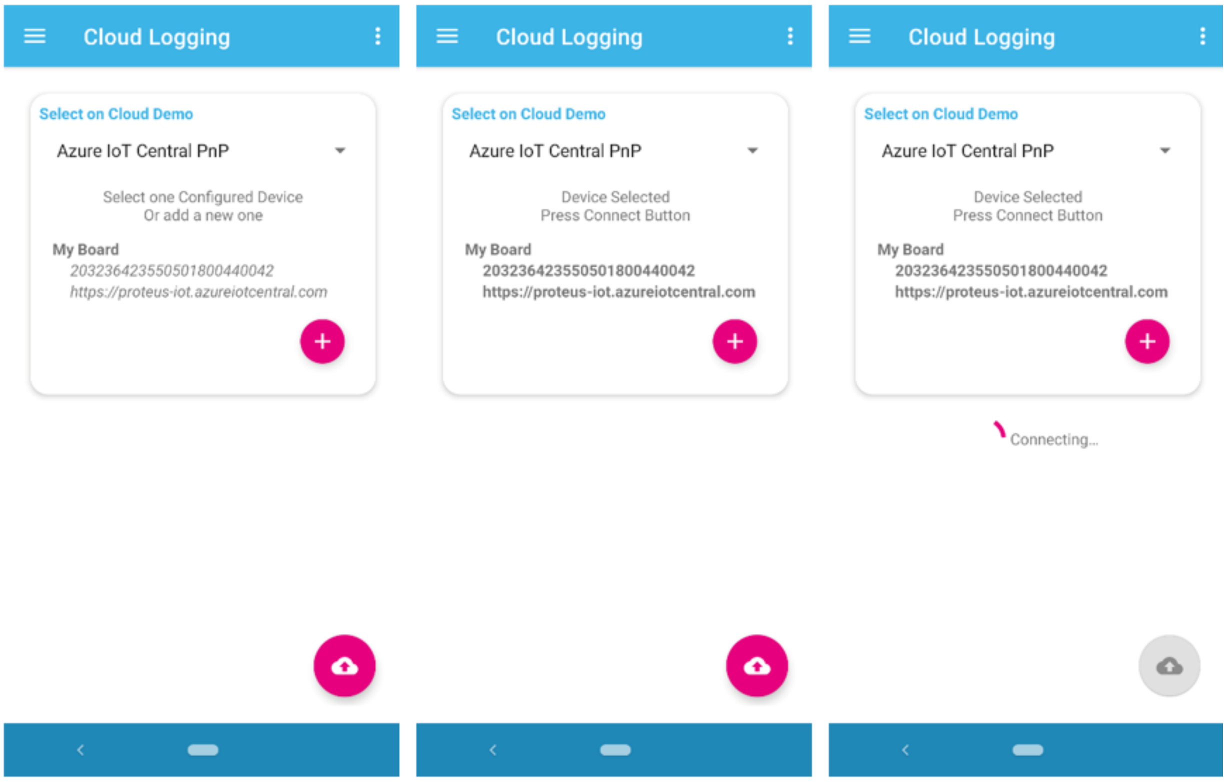
All devices

Device explorer helps you see all your devices. Detailed information like device raw data helps you troubleshoot. Learn more

| Device name | Device ID | Device status | Device template | Organization | Simulated |
|---|---|---|---|---|---|
| My Board | 2032364235505018004A00... | Registered | PROTEUS_draftv1 | proteus-iot | No |

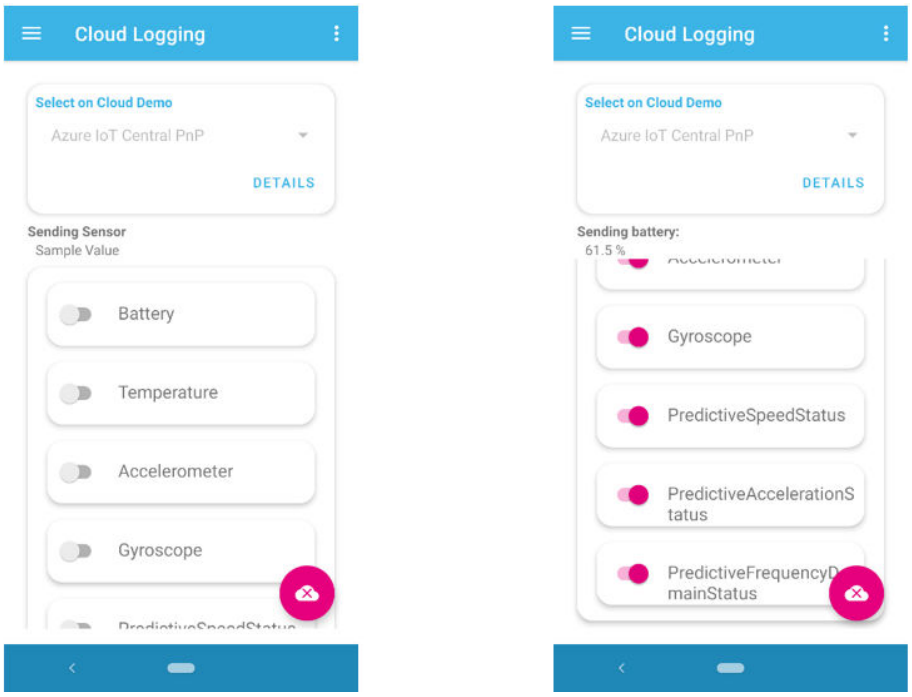
After you have selected the device on your mobile phone, tap on the chosen one and tap on the cloud upload button to connect it to the cloud.

Figure 30. **Connecting the chosen device to the cloud**



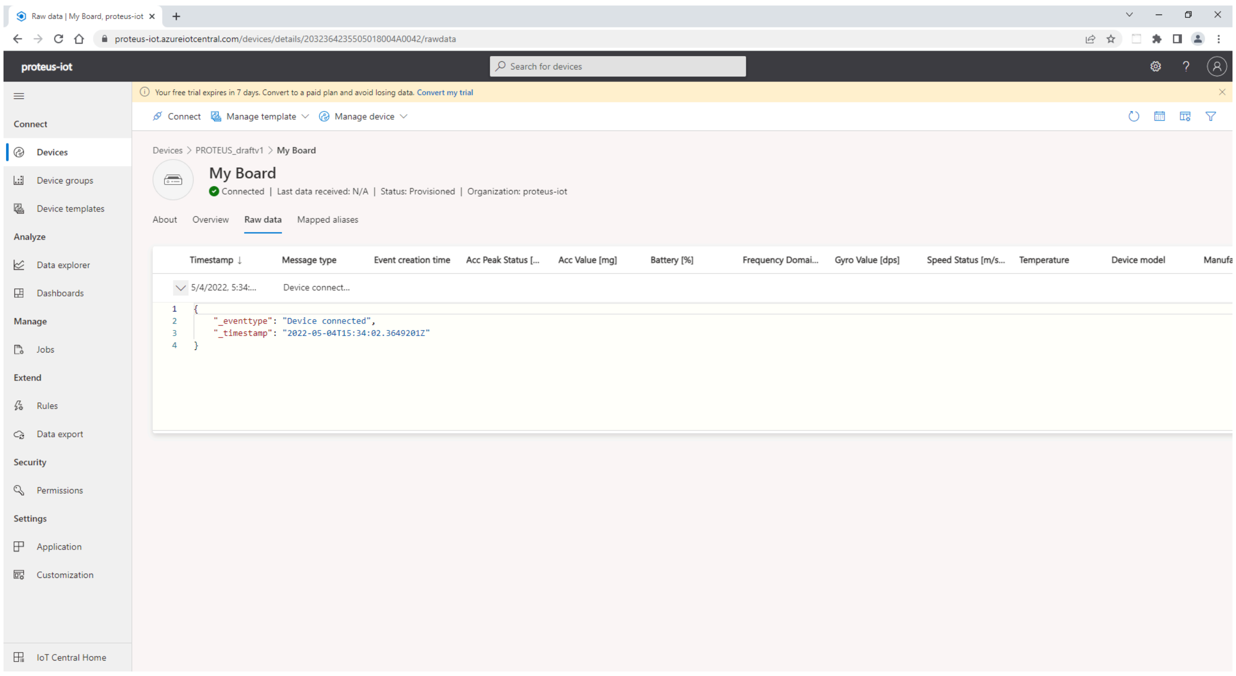
As soon as the connection is on, the data you can upload to the cloud is listed and you can select the ones you prefer.

Figure 31. **Data to upload to the cloud**



In the web page, you can view the data received on the device you are using.

Figure 32. Azure IoT Central PnP cloud dashboard - device connected



Your free trial expires in 7 days. Convert to a paid plan and avoid losing data. Convert my trial

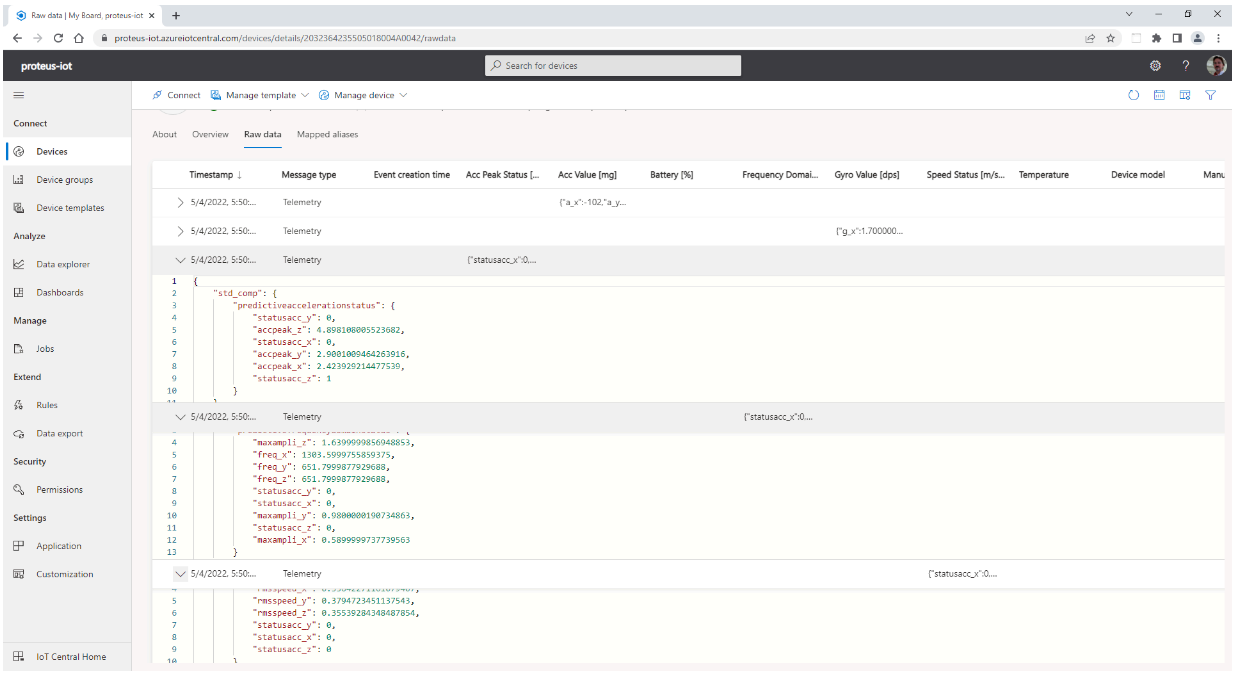
My Board

Connected | Last data received: N/A | Status: Provisioned | Organization: proteus-iot

| Timestamp ↓ | Message type | Event creation time | Acc Peak Status [... | Acc Value [mg] | Battery [%] | Frequency Domai... | Gyro Value [dps] | Speed Status [m/s... | Temperature | Device model | Manufa |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5/4/2022, 5:34:... | Device connect... | | | | | | | | | | |

```
{
    "_eventtype": "Device connected",
    "_timestamp": "2022-05-04T15:34:02.3649201Z"
}
```

Figure 33. Azure IoT Central PnP cloud dashboard - telemetry



| Timestamp ↓ | Message type | Event creation time | Acc Peak Status [... | Acc Value [mg] | Battery [%] | Frequency Domai... | Gyro Value [dps] | Speed Status [m/s... | Temperature | Device model | Manu |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5/4/2022, 5:50:... | Telemetry | | | {"a_x":-102,"a_y... | | | | | | | |
| 5/4/2022, 5:50:... | Telemetry | | | | | | {"g_x":1.700000... | | | | |
| 5/4/2022, 5:50:... | Telemetry | | {"statusacc_x":0,... | | | | | | | | |

```
{
    "std_comp": {
        "predictiveaccelerationstatus": {
            "statusacc_y": 0,
            "accpeak_z": 4.898108005523682,
            "statusacc_x": 0,
            "accpeak_y": 2.9001009464263916,
            "accpeak_x": 2.423929214477539,
            "statusacc_z": 1
        }
```

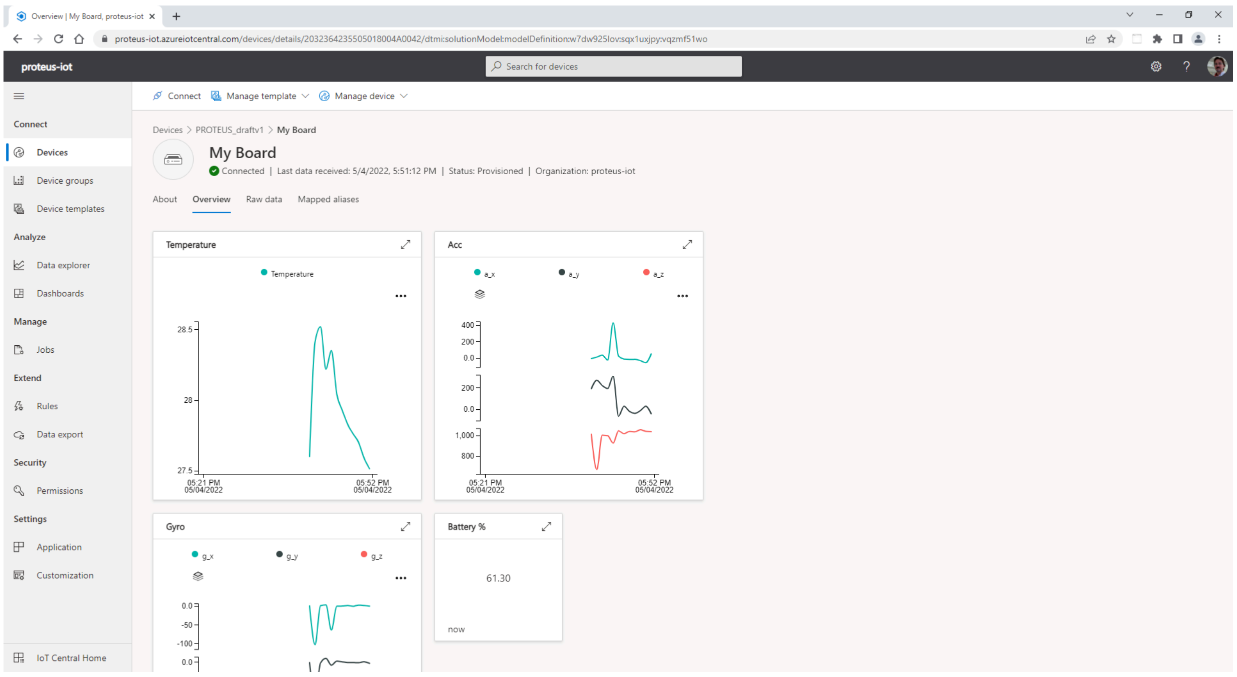
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5/4/2022, 5:50:... | Telemetry | | | | | {"statusacc_x":0,... | | | | | |

```
            "maxampli_z": 1.6399999856948853,
            "freq_x": 1303.5999755859375,
            "freq_y": 651.7999877929688,
            "freq_z": 651.7999877929688,
            "statusacc_y": 0,
            "statusacc_x": 0,
            "maxampli_y": 0.9800000190734863,
            "statusacc_z": 0,
            "maxampli_x": 0.5899999737739563
        }
```

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5/4/2022, 5:50:... | Telemetry | | | | | | | {"statusacc_x":0,... | | | |

```
            "rmsspeed_y": 0.3794723451137543,
            "rmsspeed_z": 0.35539284348487854,
            "statusacc_y": 0,
            "statusacc_x": 0,
            "statusacc_z": 0
```

**Figure 34.** Azure IoT Central PnP cloud dashboard - overview (1 of 3)



My Board

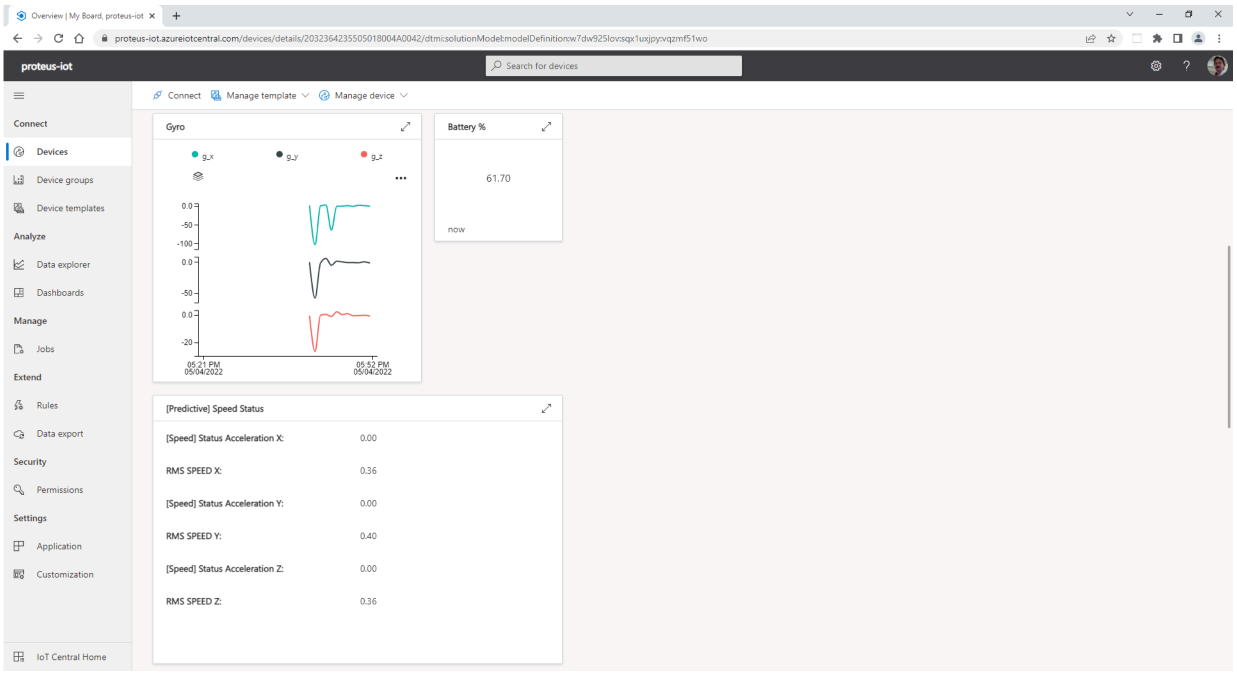
Connected | Last data received: 5/4/2022, 5:51:12 PM | Status: Provisioned | Organization: proteus-iot

Battery %

61.30

now

**Figure 35.** Azure IoT Central PnP cloud dashboard - overview (2 of 3)



Battery %

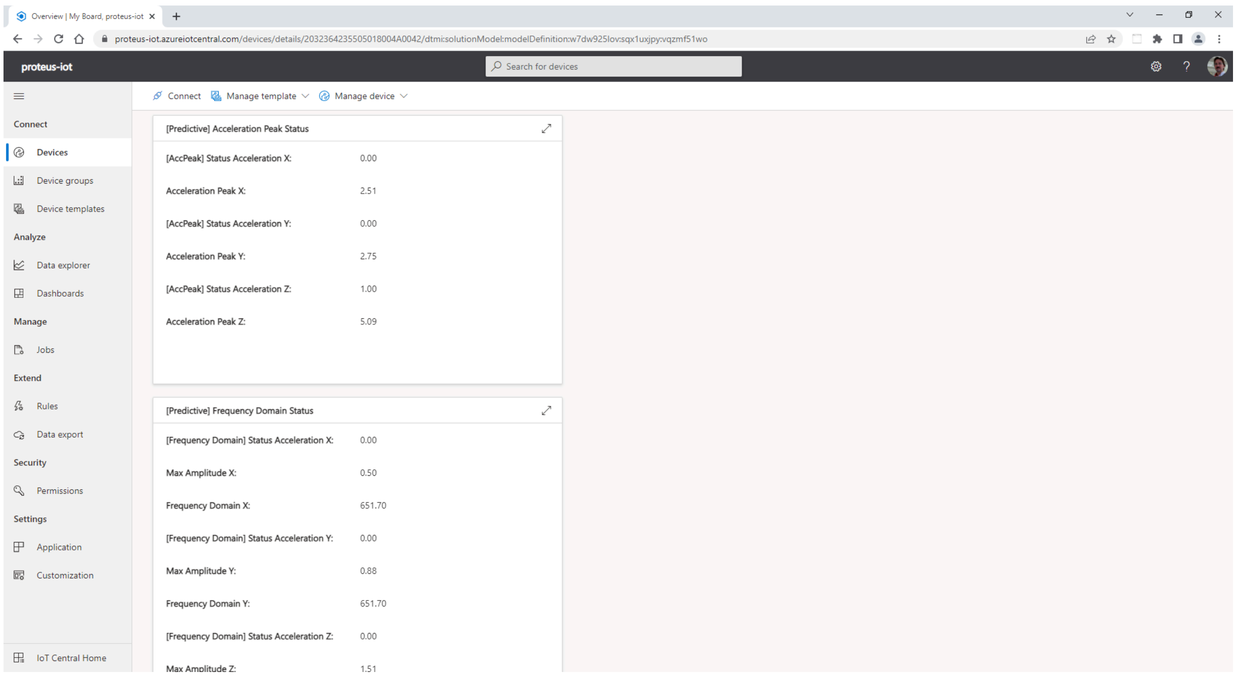
61.70

now

[Predictive] Speed Status

| | |
|---|---|
| [Speed] Status Acceleration X: | 0.00 |
| RMS SPEED X: | 0.36 |
| [Speed] Status Acceleration Y: | 0.00 |
| RMS SPEED Y: | 0.40 |
| [Speed] Status Acceleration Z: | 0.00 |
| RMS SPEED Z: | 0.36 |

Figure 36. Azure IoT Central PnP cloud dashboard - overview (3 of 3)


Overview | My Board, proteus-iot
proteus-iot.azureiotcentral.com/devices/details/2032364235505018004A0042/dtmi:solutionModel:modelDefinition:w7dw925lov:sqx1uxjpy:vqzmf51wo

[Predictive] Acceleration Peak Status

| | |
|---|---|
| [AccPeak] Status Acceleration X: | 0.00 |
| Acceleration Peak X: | 2.51 |
| [AccPeak] Status Acceleration Y: | 0.00 |
| Acceleration Peak Y: | 2.75 |
| [AccPeak] Status Acceleration Z: | 1.00 |
| Acceleration Peak Z: | 5.09 |

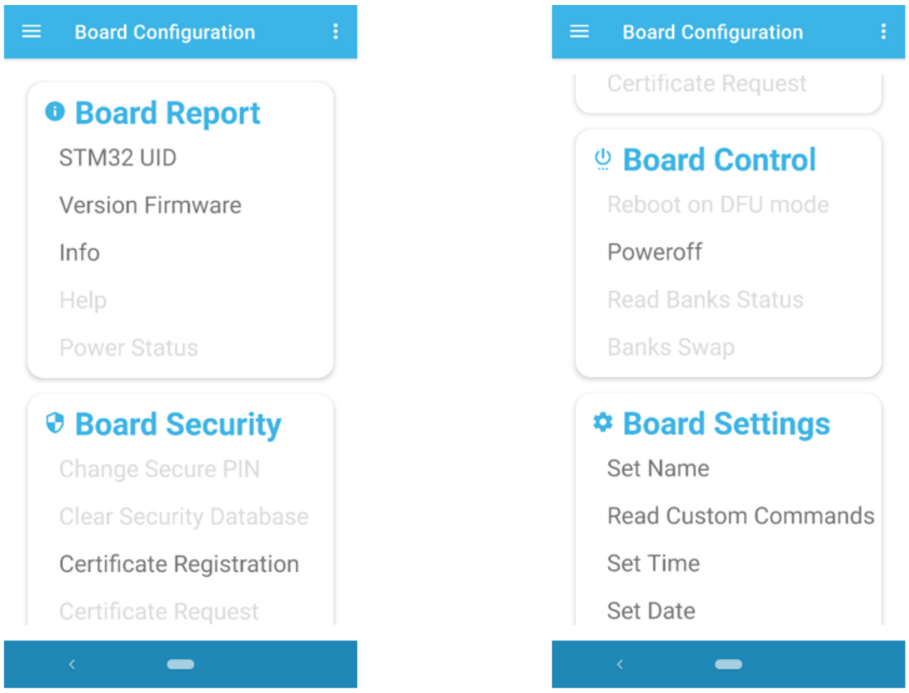
[Predictive] Frequency Domain Status

| | |
|---|---|
| [Frequency Domain] Status Acceleration X: | 0.00 |
| Max Amplitude X: | 0.50 |
| Frequency Domain X: | 651.70 |
| [Frequency Domain] Status Acceleration Y: | 0.00 |
| Max Amplitude Y: | 0.88 |
| Frequency Domain Y: | 651.70 |
| [Frequency Domain] Status Acceleration Z: | 0.00 |
| Max Amplitude Z: | 1.51 |

### 4.4.9 Board configuration

Touching *"Board Configuration"* in the STBLESensClassic app menu, the mobile device can show some information about the board.

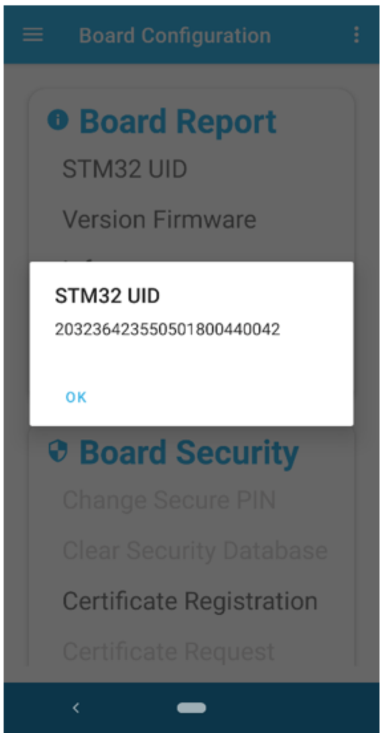
Figure 37. Board configuration



You can:

- Retrieve the STM32 UID (unique 96-bit ID) of the STM32WB mounted in the STEVAL-PROTEUS.

Figure 38. **Retrieving the STM32 UID**

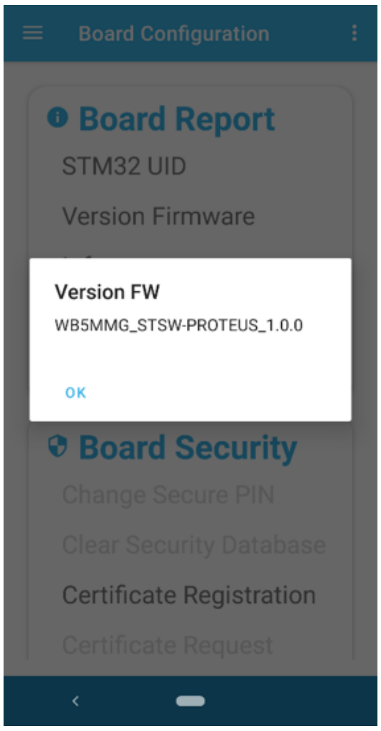


STM32 UID

2032364235505018004400442

- Read the firmware version loaded on the STEVAL-PROTEUS.

Figure 39. **Reading the firmware version**

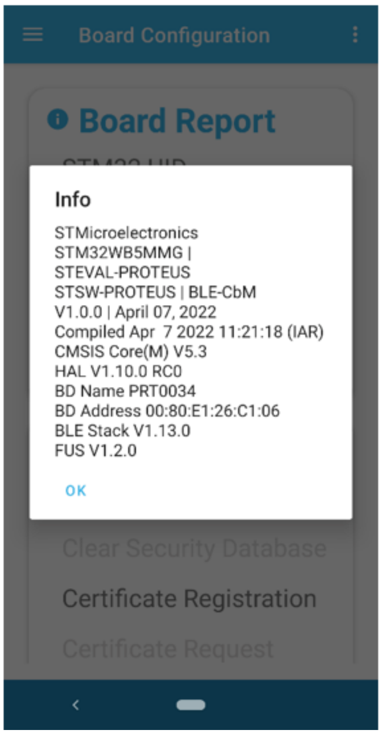


Version FW

WB5MMG_STSW-PROTEUS_1.0.0

- Read the detailed information about the firmware loaded in the STEVAL-PROTEUS.

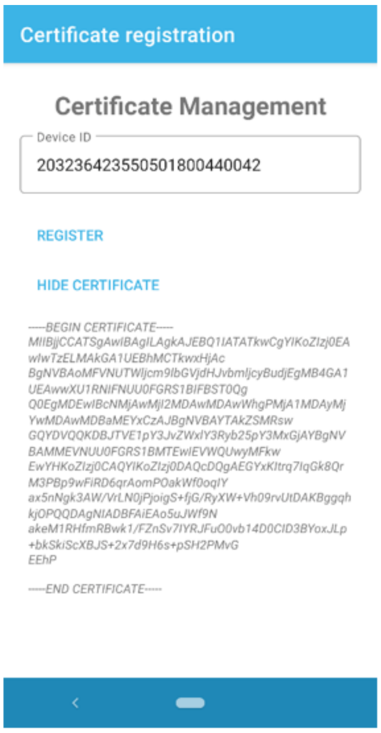
**Figure 40. Reading detailed information on the firmware version**



```
STMicroelectronics
STM32WB5MMG |
STEVAL-PROTEUS
STSW-PROTEUS | BLE-CbM
V1.0.0 | April 07, 2022
Compiled Apr  7 2022 11:21:18 (IAR)
CMSIS Core(M) V5.3
HAL V1.10.0 RC0
BD Name PRT0034
BD Address 00:80:E1:26:C1:06
BLE Stack V1.13.0
FUS V1.2.0
```

- Retrieve the privacy enhanced mail (PEM) certificate provided by the mounted STSAFE-A110 on the STEVAL-PROTEUS.

**Figure 41. Retrieving the PEM**



Device ID: 2032364235505018004400042

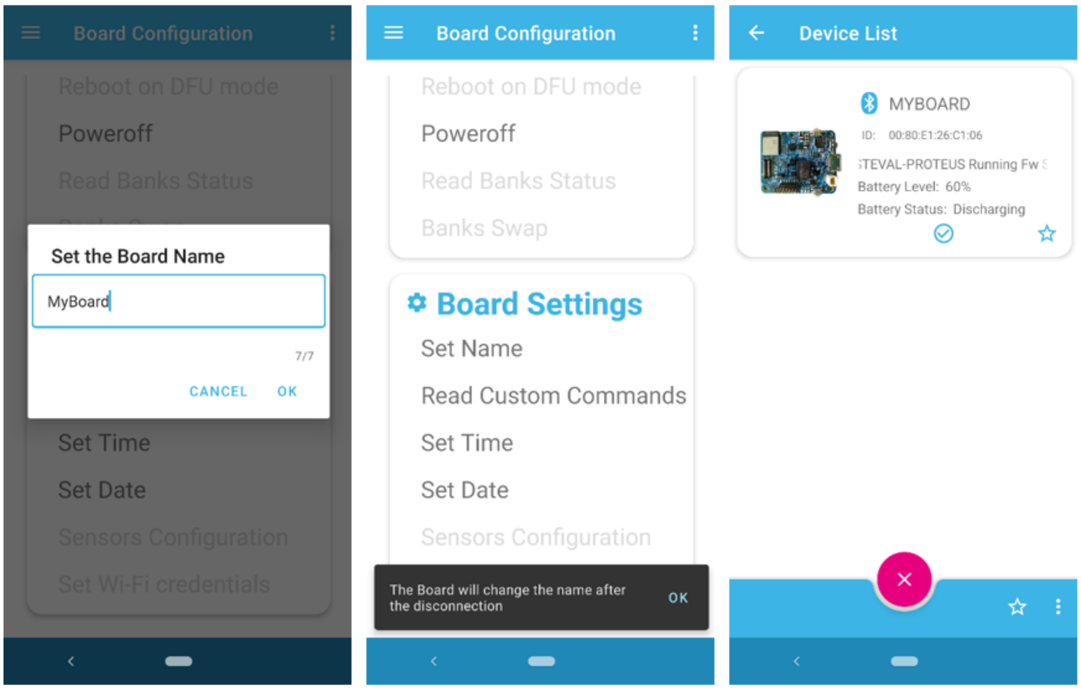
```
-----BEGIN CERTIFICATE-----
MIIBjjCCATSgAwIBAgILAgkAJEBQ1IATATkwCgYIKoZIzj0EA
wIwTzELMAkGA1UEBhMCTkwxHjAc
BgNVBAoMFVNUTWljcm9lbGVjdHJvbmljcyBudjEgMB4GA1
UEAwwXU1RNIFNUU0FGRS1BIFBST0Qg
Q0EgMDEwIBcNMjAwMjI2MDAwMDAwWhgPMjA1MDAyMj
YwMDAwMDBaMEYxCzAJBgNVBAYTAkZSMRsw
GQYDVQQKDBJTVE1pY3JvZWxlY3Ryb25pY3MxGjAYBgNV
BAMMEVNUU0FGRS1BMTEwIEVWQUwyMFkw
EwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEGYxKItrq7IqGk8Qr
M3PBp9wFiRD6qrAomPOakWf0oqIY
ax5nNgk3AW/VrLN0jPjoigS+fjG/RyXW+Vh09rvUtDAKBggqh
kjOPQQDAgNIADBFAiEAo5uJWf9N
akeM1RHfmRBwk1/FZnSv7IYRJFuO0vb14D0CID3BYoxJLp
+bkSkiScXBJS+2x7d9H6s+pSH2PMvG
EEhP

-----END CERTIFICATE-----
```

- Power off the board.

- Change the name of the STEVAL-PROTEUS that you are using so that the device is seen with a new name during the Bluetooth® Low Energy advertising.

Figure 42. **Changing the device name**

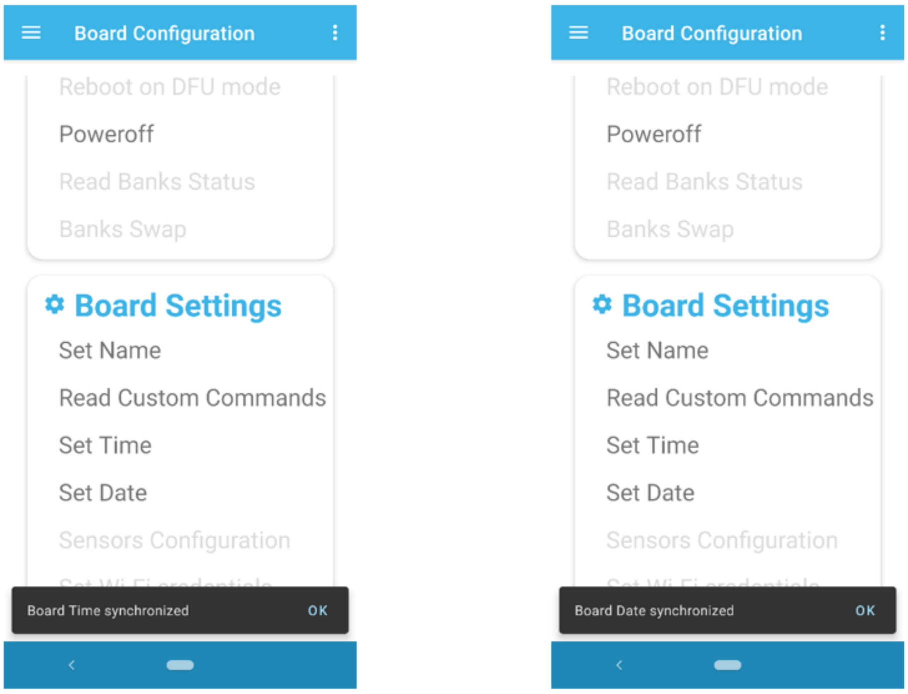


The Board will change the name after the disconnection

ID: 00:80:E1:26:C1:06
STEVAL-PROTEUS Running Fw
Battery Level: 60%
Battery Status: Discharging

- Synchronize the time and date of the STEVAL-PROTEUS and the smartphone.

Figure 43. **Time and date synchronization**



Board Time synchronized

Board Date synchronized

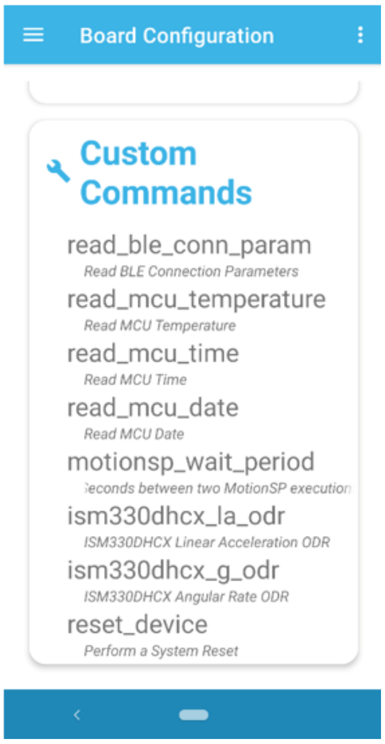
- Read custom commands.

#### 4.4.9.1 *Custom commands*

The STBLESensClassic app can read commands built via firmware. You can see and apply them directly through the same app.

Tap on *"Read Custom Commands"* to add the ones currently available into a dedicated tab.

**Figure 44.** Custom commands

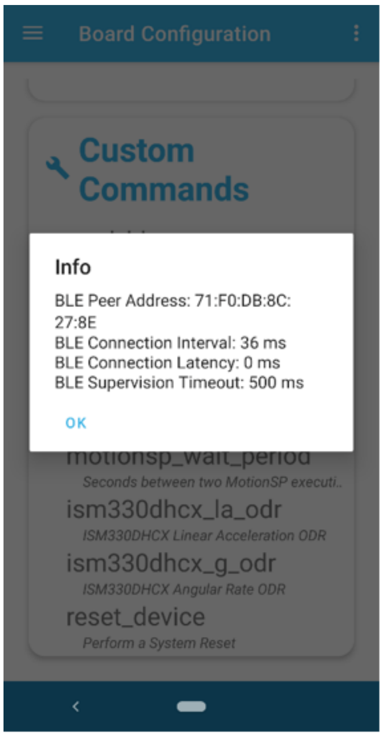


read_ble_conn_param
*Read BLE Connection Parameters*
read_mcu_temperature
*Read MCU Temperature*
read_mcu_time
*Read MCU Time*
read_mcu_date
*Read MCU Date*
motionsp_wait_period
*Seconds between two MotionSP execution*
ism330dhcx_la_odr
*ISM330DHCX Linear Acceleration ODR*
ism330dhcx_g_odr
*ISM330DHCX Angular Rate ODR*
reset_device
*Perform a System Reset*

The implemented commands allow:

- Reading the Bluetooth® Low Energy connection parameters.

Figure 45. Reading the Bluetooth® Low Energy connection parameters

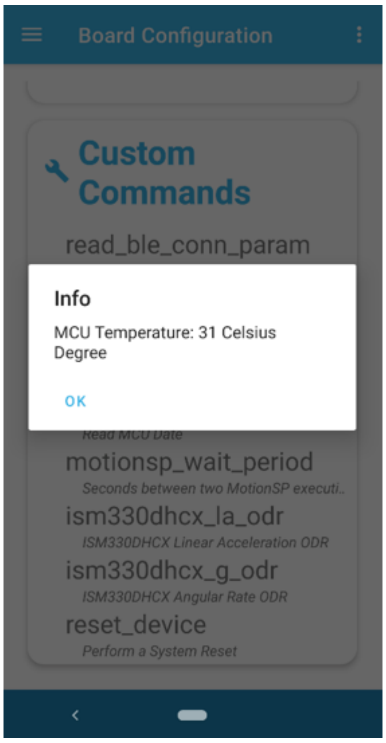


BLE Peer Address: 71:F0:DB:8C:
27:8E
BLE Connection Interval: 36 ms
BLE Connection Latency: 0 ms
BLE Supervision Timeout: 500 ms

OK
motionsp_wait_period
Seconds between two MotionSP executi..
ism330dhcx_la_odr
ISM330DHCX Linear Acceleration ODR
ism330dhcx_g_odr
ISM330DHCX Angular Rate ODR
reset_device
Perform a System Reset

- Reading the MCU temperature.

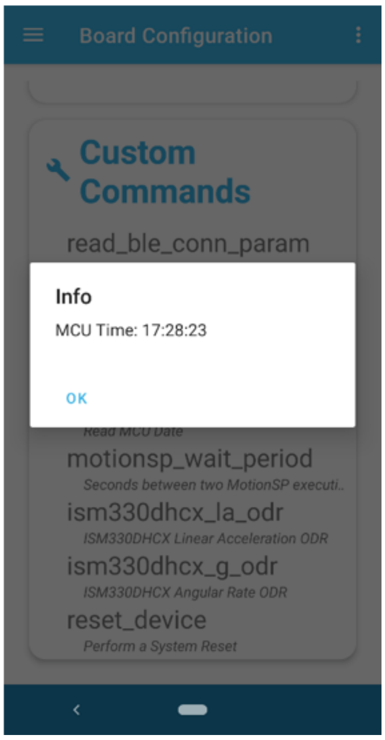
Figure 46. Reading the MCU temperature



MCU Temperature: 31 Celsius
Degree

- Reading the MCU time.

Figure 47. Reading the MCU time



Board Configuration

Custom Commands

read_ble_conn_param

Info

MCU Time: 17:28:23

OK

Read MCU Date

motionsp_wait_period

Seconds between two MotionSP executi..

ism330dhcx_la_odr

ISM330DHCX Linear Acceleration ODR

ism330dhcx_g_odr

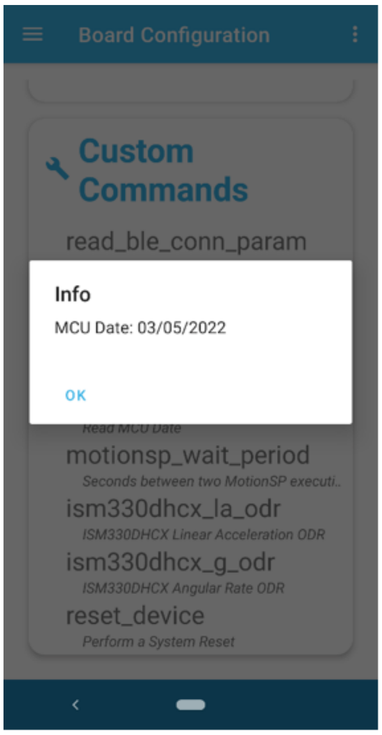
ISM330DHCX Angular Rate ODR

reset_device

Perform a System Reset

- Reading the MCU date.

Figure 48. Reading the MCU date



Board Configuration

Custom Commands

read_ble_conn_param

Info

MCU Date: 03/05/2022

OK

Read MCU Date

motionsp_wait_period

Seconds between two MotionSP executi..

ism330dhcx_la_odr

ISM330DHCX Linear Acceleration ODR

ism330dhcx_g_odr

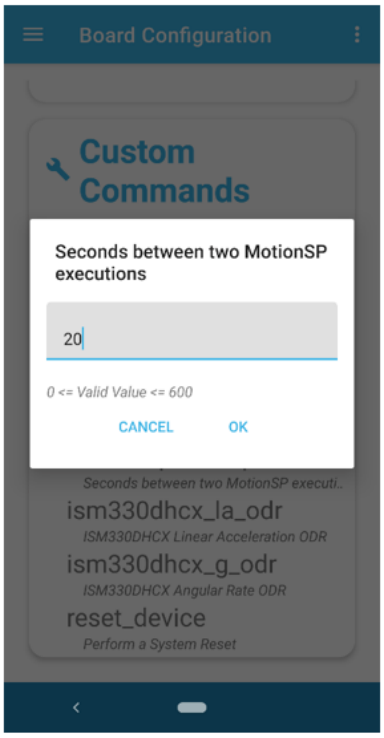
ISM330DHCX Angular Rate ODR

reset_device

Perform a System Reset

- Setting the delay, in seconds, between two consecutive MotionSP executions.

Figure 49. **Setting the delay for MotionSP executions**



Seconds between two MotionSP executions

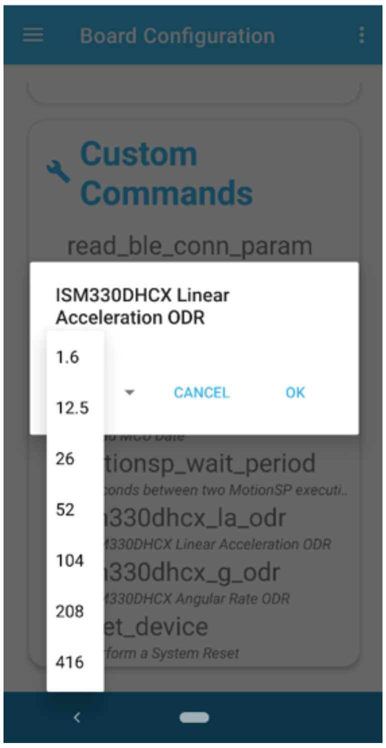
20

0 <= Valid Value <= 600

Seconds between two MotionSP executi..
ism330dhcx_la_odr
ISM330DHCX Linear Acceleration ODR
ism330dhcx_g_odr
ISM330DHCX Angular Rate ODR
reset_device
Perform a System Reset

- Setting the ODR for the ISM330DHCX linear acceleration.

Figure 50. **Setting the ODR for the linear acceleration**



read_ble_conn_param

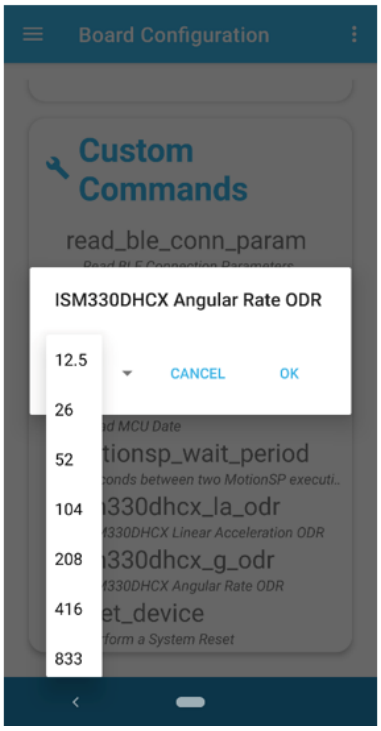
ISM330DHCX Linear Acceleration ODR

1.6
12.5
26
52
104
208
416

tionsp_wait_period
onds between two MotionSP executi..
330dhcx_la_odr
330DHCX Linear Acceleration ODR
330dhcx_g_odr
330DHCX Angular Rate ODR
et_device
form a System Reset

- Setting the ODR for the ISM330DHCX angular rate.

**Figure 51. Setting the ODR for the angular rate**



- Performing a system reset.

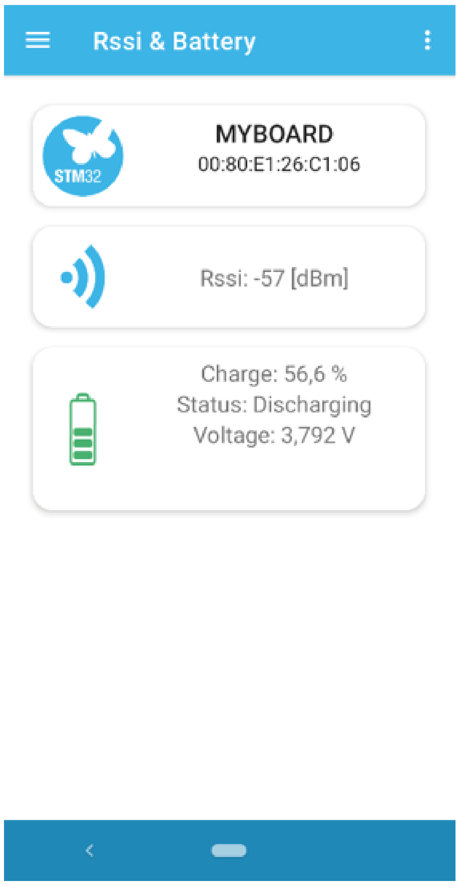
*Note: Other commands can be implemented if needed.*

### 4.4.10 RSSI and battery

Touching *"Rssi & Battery"* in the STBLESensClassic app menu, the following information is shown:

- Board name and MAC address
- RSSI
- Battery information, such as the level and status of the charge and voltage.

Figure 52. **RSSI and battery**



MYBOARD
00:80:E1:26:C1:06

Rssi: -57 [dBm]

Charge: 56,6 %
Status: Discharging
Voltage: 3,792 V

### 4.4.11 Firmware update (fast FUOTA)

Touching *"Firmware Update"* in the STBLESensClassic app menu, you can use the “Fast Firmware Update Over-the-Air” feature to reprogram the firmware to be used by the STEVAL-PROTEUS, both as application, for Cortex® M4, and as wireless stack, for Cortex® M0+.

The firmware used must be developed for the STEVAL-PROTEUS evaluation board.

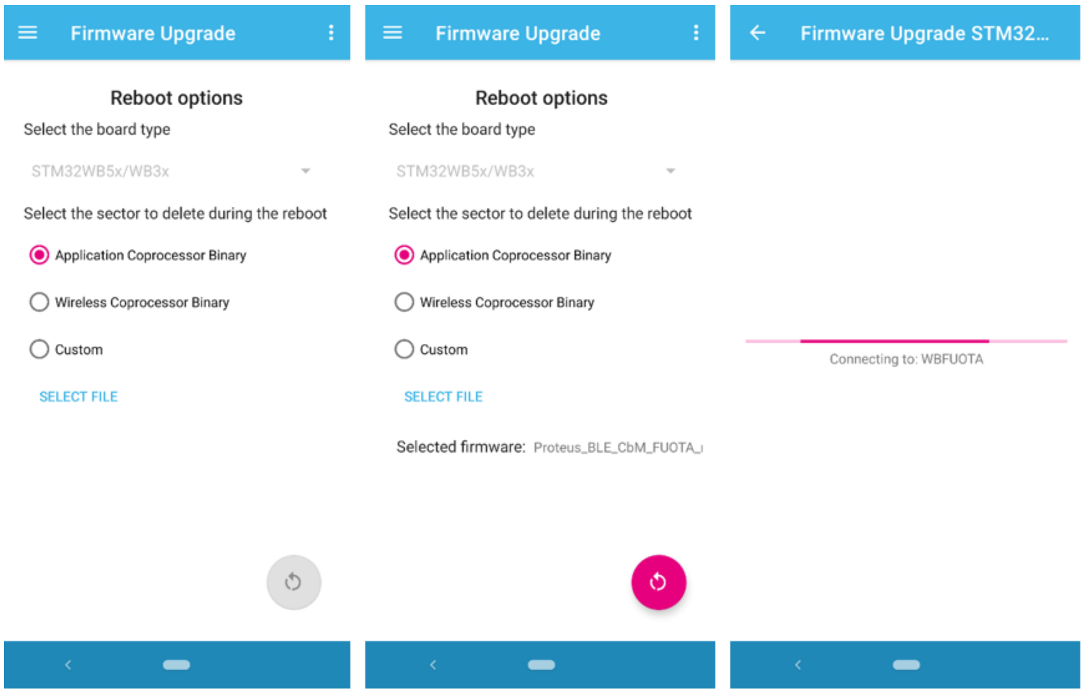
In any case, you can reprogram your STEVAL-PROTEUS as described in Section 6 Download the firmware into the STEVAL-PROTEUS.

### 4.4.11.1 *Application firmware update*

To update the application firmware, select *"Application Coprocessor Binary"* and the appropriate binary file by searching inside the folders of the mobile device.

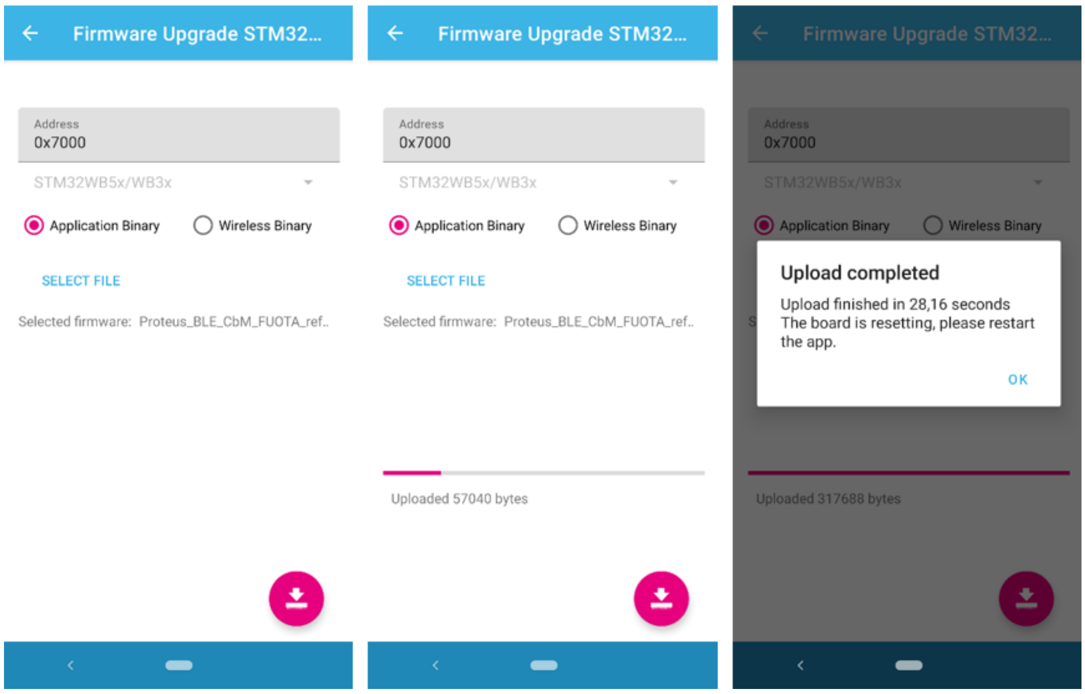
Then tap on the *"Reboot"* button. The STEVAL-PROTEUS reboots as the new Bluetooth® Low Energy slave (WBFUOTA), so that the Bluetooth® Low Energy master (mobile device) connects to it.

Figure 53. **STEVAL-PROTEUS FUOTA for the application firmware**


Firmware Upgrade
Reboot options
Select the board type
STM32WB5x/WB3x
Select the sector to delete during the reboot
Application Coprocessor Binary
Wireless Coprocessor Binary
Custom
SELECT FILE
Firmware Upgrade
Reboot options
Select the board type
STM32WB5x/WB3x
Select the sector to delete during the reboot
Application Coprocessor Binary
Wireless Coprocessor Binary
Custom
SELECT FILE
Selected firmware: Proteus_BLE_CbM_FUOTA_
Firmware Upgrade STM32...
Connecting to: WBFUOTA

After the STEVAL-PROTEUS has been rebooted as WBFUOTA, the STBLESensClassic app shows a new page. The page displays the start address for the firmware that you are going to download as well as the related file. Just tap on the "Download" button to start the update.

Figure 54. STM32WB FUOTA for the application firmware



Selected firmware: Proteus_BLE_CbM_FUOTA_ref..

Uploaded 57040 bytes

Upload completed
Upload finished in 28,16 seconds
The board is resetting, please restart the app.

Uploaded 317688 bytes

After the upload has been completed, the board reboots as STEVAL-PROTEUS, according to the uploaded firmware.

You can find it in the list of the connectable devices.

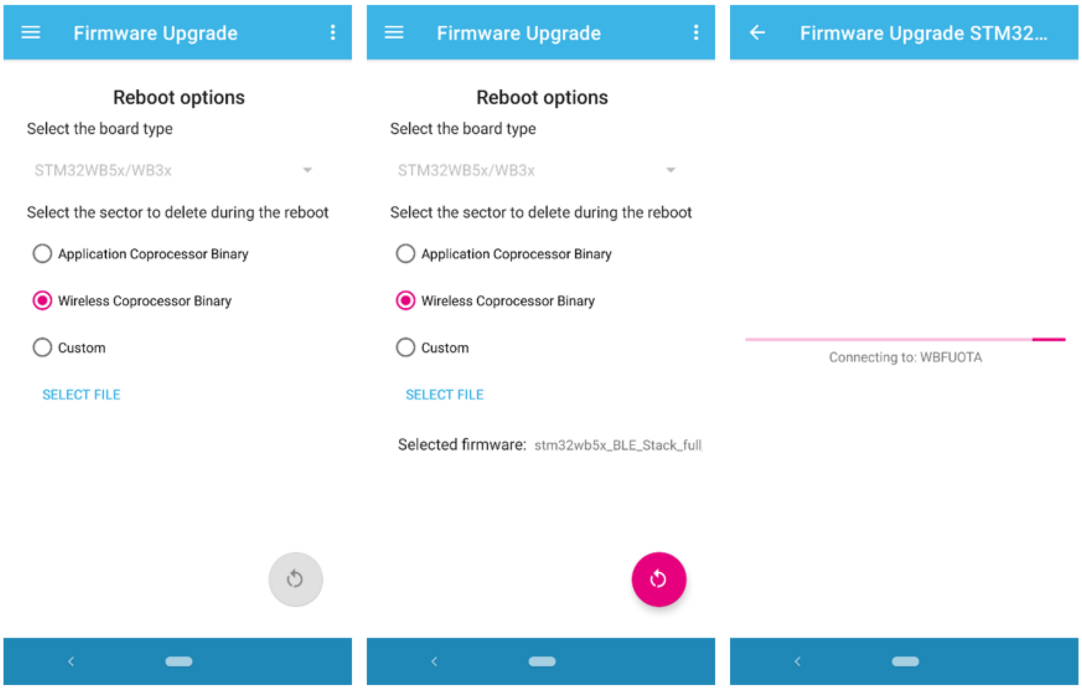
### 4.4.11.2 *Wireless stack firmware update*

*Note: The wireless stack update procedure erases the application firmware. So, you also have to update the latter again.*

To update the wireless stack firmware, select *"Wireless Coprocessor Binary"* and the appropriate binary file by searching inside the folders of the mobile device.

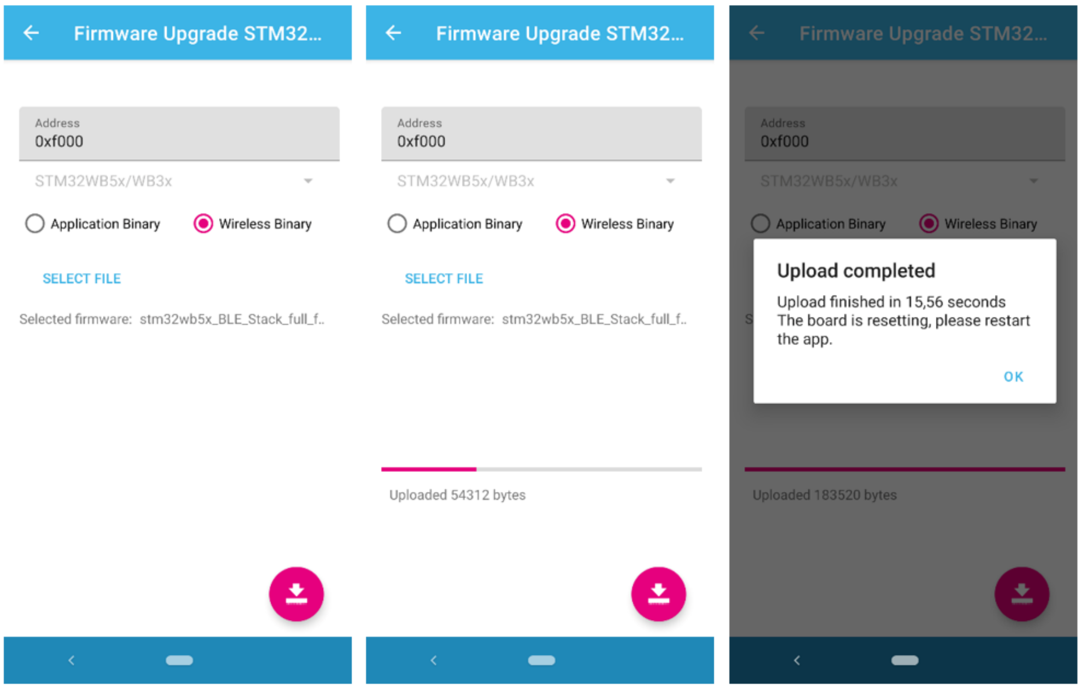
Then tap on the *"Reboot"* button. The STEVAL-PROTEUS reboots as the new Bluetooth® Low Energy slave (WBFUOTA), and the Bluetooth® Low Energy master (mobile device) connects to it.

Figure 55. **STEVAL-PROTEUS FUOTA for wireless stack firmware**


Firmware Upgrade
Reboot options
Select the board type
STM32WB5x/WB3x
Select the sector to delete during the reboot
Application Coprocessor Binary
Wireless Coprocessor Binary
Custom
SELECT FILE
Firmware Upgrade
Reboot options
Select the board type
STM32WB5x/WB3x
Select the sector to delete during the reboot
Application Coprocessor Binary
Wireless Coprocessor Binary
Custom
SELECT FILE
Selected firmware: stm32wb5x_BLE_Stack_full
Firmware Upgrade STM32...
Connecting to: WBFUOTA

After the STEVAL-PROTEUS has been rebooted as WBFUOTA, the STBLESensClassic app shows a new page. This page displays the start address for the firmware that you are going to download as well as the related file. Just tap on the "Download" button to start the update.

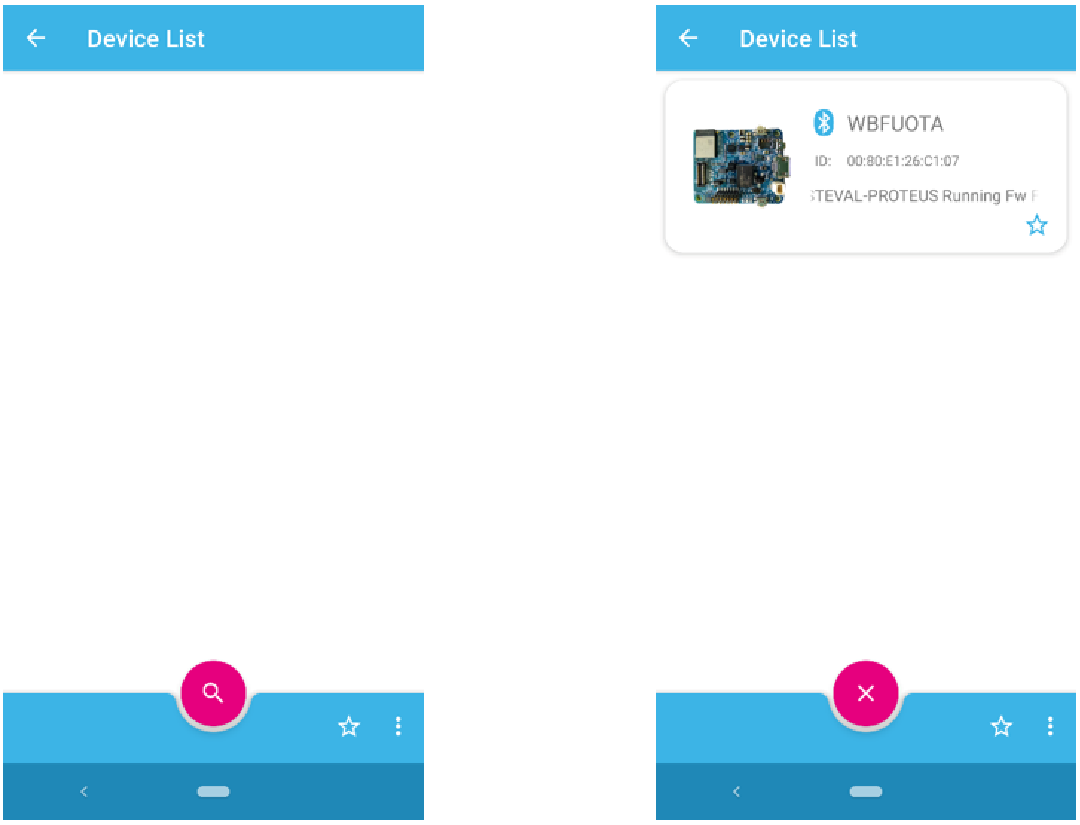
Figure 56. **STM32WB FUOTA for wireless stack firmware**



Upload finished in 15,56 seconds
The board is resetting, please restart the app.

Note: *The process is not complete yet. The firmware has been stored in an area of the flash memory buffer. The FUS copies the binary to the right flash memory area. Do not disconnect the power from the board and do not press any buttons on it. This process can take up to 25 seconds. During this period, the board is no longer visible in the app device list. Just tap on the "Search" button until you see the WBFUOTA board in the list of the Bluetooth® Low Energy connectable devices.*

Figure 57. **Searching the WBFUOTA in the device list**



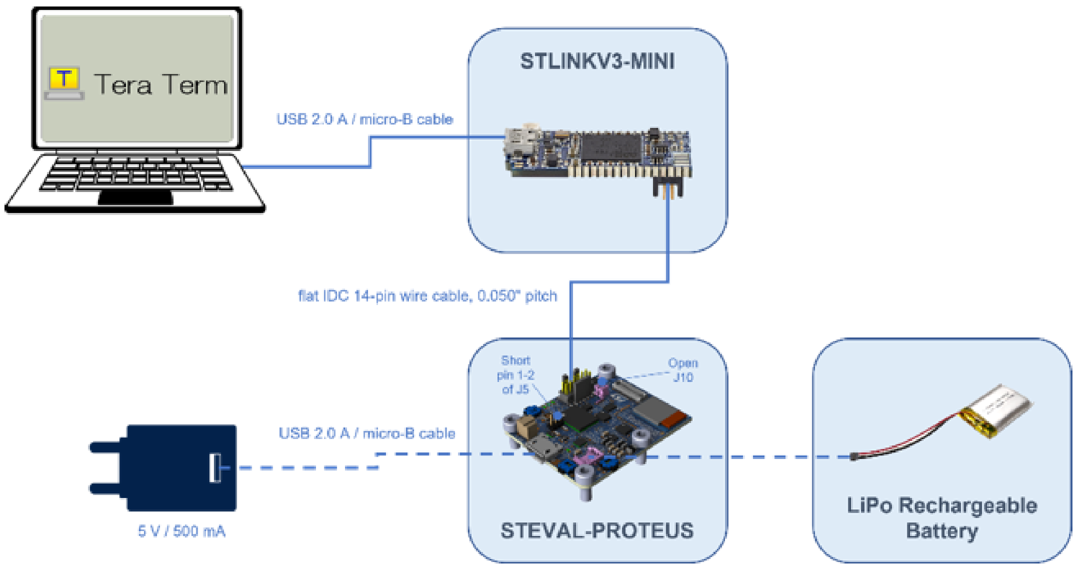
Then, you can proceed with the update of the application firmware.

## 4.5 Terminal emulator

When connected to a PC via the STLINK-V3MINI, which also provides a virtual COM port interface and allows the host PC to communicate with the target microcontroller through a UART, the STEVAL-PROTEUS that runs the STSW-PROTEUS can send information to a terminal console.

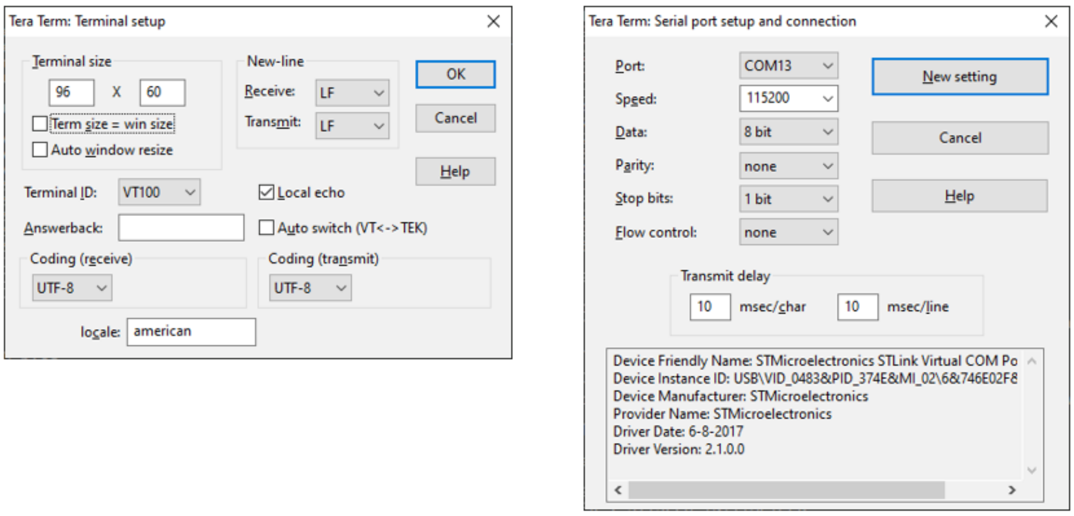
The figure below shows how to connect the STEVAL-PROTEUS to a PC able to run a terminal emulator such as Tera Term.

**Figure 58.** **STEVAL-PROTEUS connection for the terminal emulator**


Tera Term
USB 2.0 A / micro-B cable
STLINKV3-MINI
flat IDC 14-pin wire cable, 0.050" pitch
Short pin 1-2 of J5
Open J10
USB 2.0 A / micro-B cable
5 V / 500 mA
STEVAL-PROTEUS
LiPo Rechargeable Battery

Take care to configure the terminal emulator as shown below.

**Figure 59.** **STEVAL-PROTEUS connection for the terminal emulator**



Tera Term: Terminal setup

Terminal size: 96 X 60
☐ Term size = win size
☐ Auto window resize
New-line
Receive: LF
Transmit: LF
OK
Cancel
Help
Terminal ID: VT100
☑ Local echo
Answerback:
☐ Auto switch (VT<->TEK)
Coding (receive): UTF-8
Coding (transmit): UTF-8
locale: american

Tera Term: Serial port setup and connection

Port: COM13
Speed: 115200
Data: 8 bit
Parity: none
Stop bits: 1 bit
Flow control: none
New setting
Cancel
Help
Transmit delay
10 msec/char
10 msec/line

```
Device Friendly Name: STMicroelectronics STLink Virtual COM Po
Device Instance ID: USB\VID_0483&PID_374E&MI_02\6&746E02F8
Device Manufacturer: STMicroelectronics
Provider Name: STMicroelectronics
Driver Date: 6-8-2017
Driver Version: 2.1.0.0
```

The following code block shows an example of output on the terminal emulator program that runs on a PC.

```
Wireless Firmware version 1.13.0
Wireless Firmware build 5
FUS version 1.2.0

SHCI_SUB_EVT_CODE_READY - WIRELESS_FW_RUNNING
DBGMCU_GetRevisionID= 2001


***************************************************************************
**  STMicroelectronics
**  STM32WB5MMG - Bluetooth Low Energy 5.0 and 802.15.4 module
**  STSW-PROTEUS | BLE-CbM V1.0.0 - April 07, 2022
**  CMSIS Core(M) V5.3
**  HAL V1.10.0 RC0
**  Compiled Apr  7 2022 11:21:15 (IAR)
***************************************************************************
**  MCU Unique device ID is 0x2032364235505018004A0042
**  MCU Flash Size is 1024 KB
***************************************************************************

    BD Name PRT0042
    BD Address 00:80:E1:26:C2:62
    BLE Stack V1.13.0
    BLE Stack Branch 0 Type 5
    FUS V1.2.0



STSAFE-A1xx initialized successfully


STSAFE-A1xx echoed successfully


Pairing:
 1. Check local envelope key presence through STSAFE-A1x0
        => StSafeA_LocalEnvelopeKeySlotQuery
 2. Check host keys presence through STSAFE-A1x0
        => StSafeA_HostKeySlotQuery
 3. Read host keys through NOR flash memory
    Host MAC key:       00112233445566778899AABBCCDDEEFF
    Host cipher key:    11112222333344445555666677778888

STSAFE-A1xx paired successfully


Authentication:
 1. Get size of certificate stored through STSAFE-A's zone 0
    1.1 Read 4 bytes of certificate through STSAFE-A's zone 0
        => Use StSafeA_Read API
    1.2 Size of certificate stored through STSAFE-A's zone 0 is 403 bytes
 2. Extract, parse and verify certificate stored through STSAFE-A's zone 0
    2.1 Read 403 bytes through STSAFE-A's zone 0 corresponding to IoT certificate
        => Use StSafeA_Read API
    2.2 Parse certificate extracted from STSAFE-A's zone 0
 3. Show certificate coded as DER (Distinguished Encoding Rules)
-----BEGIN DER CERTIFICATE-----
3082018F30820134A003020102020B02
0990EB4150D480130139300A06082A86
48CE3D040302304F310B300906035504
0613024E4C311E301C060355040A0C15
53544D6963726F656C656374726F6E69
6373206E763120301E06035504030C17
53544D205354534146452D412050524F
44204341203031302017 0D3230303232
363030303030305A180F323035303032
3236303030303030305A3046310B300906
0355040613024652311B301906035504
0A0C1253544D6963726F656C65637472
6F6E696373311A301806035504030C11
```

```
53545341464 52D41313130204556414C
323059301306072A8648CE3D02010608
2A8648CE3D03010703420004D86B98F5
FBD45DEF18A716482E0A62795F3228AA
1DF73EB918A54685CA2C1270006CBDD6
5AC20123448856964282F1353F1030DE
C12EE4592D3E6274BFF4A8F5300A0608
2A8648CE3D04030203490030460221 00
A43B52C163044248D4B25F585C827546
684D95B772AD3DC1C830D01F706915D6
022100C8BD3DD21105A86558C165F2BE
05CCA15557581C396DC2229CA1B18330
18BFBF
-----END DER CERTIFICATE-----

 4. Parse CA Self Signed certificate
 5. Check STSAFE-A's certificate was signed by CA using cryptographic library
 6. Generate a 32 bytes random number
    => Use StSafeA_GenerateRandom API
 7. Generate signature using STSAFE-A's private key stored into slot 0
    => Use StSafeA_GenerateSignature API
 8. Verify the generated signature's validity using cryptographic library with public key of
STSAFE-A's slot 0 key pair which was extracted from STSAFE-A's certificate
 9. Authentication result (0 means success): 0

STSAFE-A1xx authenticated successfully

-- BAT APPLICATION SERVER : BATTERY CHARGER INITIALIZED
-- ENV APPLICATION SERVER : STTS22H INITIALIZED
-- ENV APPLICATION SERVER : PROTEUS_ENV_INSTANCES_NBR = 1
-- MOTION APPLICATION SERVER : IIS3DWB INITIALIZED
-- MOTION APPLICATION SERVER : ISM330DHCX INITIALIZED
-- MOTION APPLICATION SERVER : IIS2DLPC INITIALIZED
-- MOTION APPLICATION SERVER : PROTEUS_MOTION_INSTANCES_NBR = 3
-- MOTION APPLICATION SERVER : MOTION_SP PARAMETER SET TO DEFAULT VALUES
-- MOTION EXT APPLICATION SERVER : WAKE UP DISABLED
-- MOTION EXT APPLICATION SERVER : DEFAULT ODR SET
-- MOTION EXT APPLICATION SERVER : MOTION EXT CONTEXT INITIALIZED



** Update custom option byte

-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION DISABLED



** Advertising data has been updated

First index in 0 state
Successfully Start Fast Advertising


** Update custom option byte

-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION ENABLED
-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION DISABLED


** Advertising data has been updated


** Update custom option byte

-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION DISABLED


** Advertising data has been updated


** STOP ADVERTISING **
First index in 0 state
Successfully Start Low Power Advertising


** Update custom option byte

-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SPEED NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM ACC NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT ALARM SUBRANGE NOTIFICATION DISABLED


** Advertising data has been updated

HCI_LE_CONNECTION_COMPLETE_SUBEVT_CODE for connection handle 0x801

Connection Parameters ...
Peer Address: 4D:66:CB:AA:12:D1
Connection Interval: 36 ms
Connection Latency: 0
Supervision Timeout: 500 ms
Master Clock Accuracy: 5 %

Read_PHY success
PHY Param  TX= 1, RX= 1

** CONNECTION UPDATE EVENT WITH CLIENT

** CONNECTION UPDATE EVENT WITH CLIENT
```

```
-- PROTEUS APPLICATION SERVER : CONFIG NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : CONSOLE TERM NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : CONSOLE STDERR NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : ENV NOTIFICATION ENABLED

-- CONSOLE APPLICATION SERVER : NOTIFY CLIENT WITH NEW TERM PARAMETER VALUE

-- CONSOLE APPLICATION SERVER : NOTIFY CLIENT WITH NEW TERM PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- PROTEUS APPLICATION SERVER : CONSOLE TERM NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : CONSOLE STDERR NOTIFICATION DISABLED

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

EVT_UPDATE_PHY_COMPLETE
EVT_UPDATE_PHY_COMPLETE, status ok
Read_PHY success
PHY Param  TX= 2, RX= 2
-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- ENV APPLICATION SERVER : NOTIFY CLIENT WITH NEW ENV PARAMETER VALUE

-- PROTEUS APPLICATION SERVER : ENV NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : FFT AMPLITUDE NOTIFICATION ENABLED

-- PROTEUS APPLICATION SERVER : TIME DOMAIN NOTIFICATION ENABLED

-- FFT APLITUDE APPLICATION SERVER : START TO NOTIFY CLIENT WITH NEW FFT AMPLITUDE PARAMETER
VALUE

-- FFT APLITUDE APPLICATION SERVER : CLIENT HAS BEEN NOTIFIED WITH NEW FFT AMPLITUDE PARAMETE
R VALUE

-- TIME DOMAIN APPLICATION SERVER : NOTIFY CLIENT WITH NEW TIME DOMAIN PARAMETER VALUE

-- FFT APLITUDE APPLICATION SERVER : START TO NOTIFY CLIENT WITH NEW FFT AMPLITUDE PARAMETER
VALUE

-- FFT APLITUDE APPLICATION SERVER : CLIENT HAS BEEN NOTIFIED WITH NEW FFT AMPLITUDE PARAMETE
R VALUE

-- TIME DOMAIN APPLICATION SERVER : NOTIFY CLIENT WITH NEW TIME DOMAIN PARAMETER VALUE

-- FFT APLITUDE APPLICATION SERVER : START TO NOTIFY CLIENT WITH NEW FFT AMPLITUDE PARAMETER
VALUE
```

```
-- FFT APLITUDE APPLICATION SERVER : CLIENT HAS BEEN NOTIFIED WITH NEW FFT AMPLITUDE PARAMETE
R VALUE

-- TIME DOMAIN APPLICATION SERVER : NOTIFY CLIENT WITH NEW TIME DOMAIN PARAMETER VALUE

-- FFT APLITUDE APPLICATION SERVER : START TO NOTIFY CLIENT WITH NEW FFT AMPLITUDE PARAMETER
VALUE

-- FFT APLITUDE APPLICATION SERVER : CLIENT HAS BEEN NOTIFIED WITH NEW FFT AMPLITUDE PARAMETE
R VALUE

-- TIME DOMAIN APPLICATION SERVER : NOTIFY CLIENT WITH NEW TIME DOMAIN PARAMETER VALUE

-- BAT APPLICATION SERVER : BATTERY VOLTAGE 3765 mA
-- PROTEUS APPLICATION SERVER : FFT AMPLITUDE NOTIFICATION DISABLED

-- PROTEUS APPLICATION SERVER : TIME DOMAIN NOTIFICATION DISABLED


** DISCONNECTION EVENT WITH CLIENT
First index in 0 state
Successfully Start Fast Advertising
-- MOTION EXT APPLICATION SERVER : WAKE UP DISABLED
-- MOTION EXT APPLICATION SERVER : DEFAULT ODR SET
```

# 5 Using the Zigbee application for the STEVAL-PROTEUS

## 5.1 Prerequisites

The Zigbee application can use the STEVAL-PROTEUS board as an end-device or as a router.

To allow the Zigbee application to work, the minimum requirements are:

- one coordinator (NUCLEO-WB55RG)
- one end-device or router (STEVAL-PROTEUS)

To understand better the capabilities of a Zigbee network, you should use the following topology:

- one coordinator (NUCLEO-WB55RG)
- one router (STEVAL-PROTEUS)
- two end-devices (STEVAL-PROTEUS)

The NUCLEO-WB55RG (Zigbee coordinator) must be connected to a PC with a USB 2.0 A/micro-B cable plugged into its USB receptacle, labeled “ST-LINK”. This allows supplying the board and sending data to the terminal emulator running on the PC.

The STEVAL-PROTEUS (router and as end-device) must be powered in any allowed way.

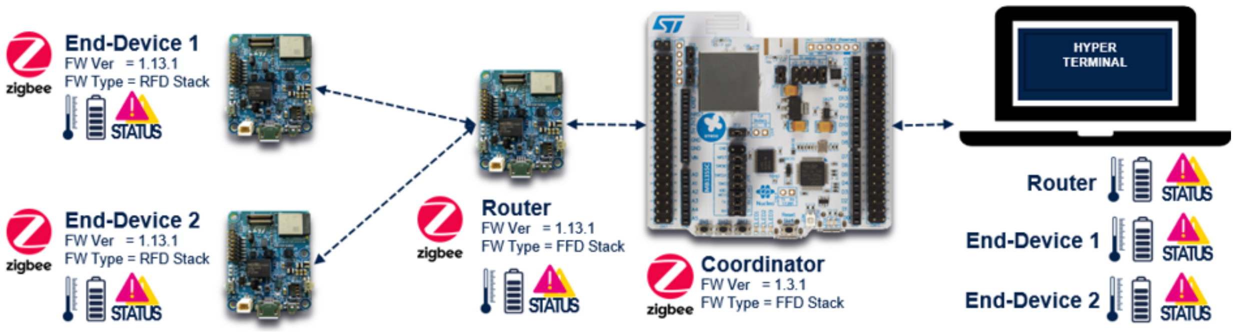
The router and the end-device data are sent to the coordinator via Zigbee. Then, the coordinator displays the data to the PC terminal emulator.

Optionally, for debugging, you can view their output on a terminal emulator running on a PC by using an STLINK-V3MINI.

## 5.2 Zigbee network setup

The figure below shows a Zigbee network example with two end-devices, a router, and a coordinator that send the output to a PC.

Figure 60. **Zigbee network example**



## 5.3 Starting a Zigbee network

The coordinator starts a Zigbee network. If the coordinator has been correctly programmed and powered on, as described in Section 6.2 STEVAL-PROTEUS flash memory programming, after few seconds its blue LED lights up.

The following code block shows the welcome message and Zigbee network start for the Zigbee coordinator, on a terminal emulator.

```
***************************************************************************
 **  Powered by STM32WB55
 **     Multiprotocol wireless 32-bit MCU Arm‡«based Cortex‡«M4
 **     with FPU, Bluetooth ‡³ and 802.15.4 radio solution
 **  MCU Unique device ID is 0x20393143554D5007003E002A
 **  MCU Flash Size is 1024 KB
 **  WIRELESS COPROCESSOR FW VERSION ID = 1.13.1
 **  WIRELESS COPROCESSOR FW Type : FFD Zigbee stack
 ***************************************************************************
```

```
Network config : APP_STARTUP_CENTRALIZED_COORD
ZbStartup Callback (status = 0x00)

INFO: COORDINATOR Long Addr=0x0080e12500082e15, Short Addr=0x0000
```

This output shows a header, which contains some pieces of information, such as the device ID, the firmware version, and the stack type. The coordinator short address is usually 0x0000 in the Zigbee network. After this message, the coordinator starts the network and waits for three minutes for another device that wants to join the network.

To join a STEVAL-PROTEUS as an end-device or as a router, power on the board.

After some seconds, once the connection has been successfully established, a blue LED lights up on the STEVAL-PROTEUS.

The terminal emulator running on the PC connected to the STEVAL-PROTEUS shows this information once the Zigbee network is started.

The following code block shows the output and network join message for the Zigbee router.

```
[M4 APPLICATION] APP_ZIGBEE_Init
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] WIRELESS COPROCESSOR FW:
[M4 APPLICATION] VERSION ID = 1.13.1
[M4 APPLICATION] FW Type : FFD Zigbee stack
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] STMicroelectronics
[M4 APPLICATION] STM32WB5MMG - Bluetooth Low Energy 5.0 and 802.15.4 module
[M4 APPLICATION] STSW-PROTEUS | ZigBee-EndNode-CbM V1.0.0 - April 07, 2022
[M4 APPLICATION] CMSIS Core(M) V5.3
[M4 APPLICATION] HAL V1.10.0 RC0
[M4 APPLICATION] Compiled Apr 15 2022 08:27:33 (IAR)
[M4 APPLICATION] **********************************************************
[M4 APPLICATION] MCU Unique device ID is 0x2032364235505018003500 3E
[M4 APPLICATION] MCU Flash Size is 1024 KB
[M4 APPLICATION] **********************************************************
 [M0] [00000000.000][API]   Init_ZigbeeStack_Infrastructure
[M4 APPLICATION] APP_ZIGBEE_StackLayersInit
[M4 APPLICATION] Network config : APP_STARTUP_CENTRALIZED_ROUTER
 [M0] [00000000.015][PLATFORM]                ZbNlmeResetReq : NLME-RESET.request (warmStar
t = 0)
 [M0] [00000000.011][PLATFORM]        zb_startup_join_nwk_disc : Attempting network discovery
. Scans = 3, Duration = 4
 [M0] [00000000.012][PLATFORM]                 nwk_scan_req : MLME-SCAN.request (wpan0): t
ype=1, page=0, mask=0x00040000, dur=4
 [M0] [00000000.280][API]  Scan Done  - unscan channels 0x0
 [M0] [00000000.281][PLATFORM]                 nwk_scan_req : MLME-SCAN.request (wpan0): t
ype=1, page=0, mask=0x00040000, dur=4
 [M0] [00000000.549][API]  Scan Done  - unscan channels 0x0
 [M0] [00000000.551][PLATFORM]                 nwk_scan_req : MLME-SCAN.request (wpan0): t
ype=1, page=0, mask=0x00040000, dur=4
 [M0] [00000000.818][API]  Scan Done  - unscan channels 0x0
 [M0] [00000000.507][API]   Poll Request - g_MAC_SUCCESS_c State !
 [M0] [00000000.614][PLATFORM]      ZbMonitorHandleTransKeyInd : Adding network key, sequence
 number = 0
 [M0] [00000000.616][PLATFORM]              ZbZdoDeviceAnnce : Sending Device_Annce for 0xa
b55
[M4 APPLICATION] ZbStartup Callback (status = 0x00)
[M4 APPLICATION] INFO:

[M4 APPLICATION] INFO: Router joined to a Zigbee Network.

[M4 APPLICATION] INFO: Router Short Addr=0xab55
```

In the previous output, the coordinator started a Zigbee network and a router connected to it. New nodes (routers and/or end-devices), after power-on, join the network through the closest “parent” device able to manage the join request (only other routers and the coordinator can have this role).

## 5.4 Network information messages

After the Zigbee network is started, any time a node joins it, a message containing the node neighbor list and the related network information is sent from the node to the coordinator. The received information is visible on the terminal emulator. After the first message, the node periodically sends the neighbor list information to the coordinator.

The following code block is an example of the output after a router has been connected to the coordinator.

```
INFO: ROUTER [0xaa16], ID:0x20323642355050180001C0047 Neighbor:[0x0000],COORD  ,PARENT
 , LQI:132,
RSSI:-59
```

This line shows a comma separated list of information about the node neighbors. In this case, the node is a router. The following data are shown: the Zigbee short address, the device ID, the neighbor address, type, relationship, the connection LQI, and RSSI.

The node sends a line for any neighbor. In this case, the node is a router and its only neighbor is the coordinator.

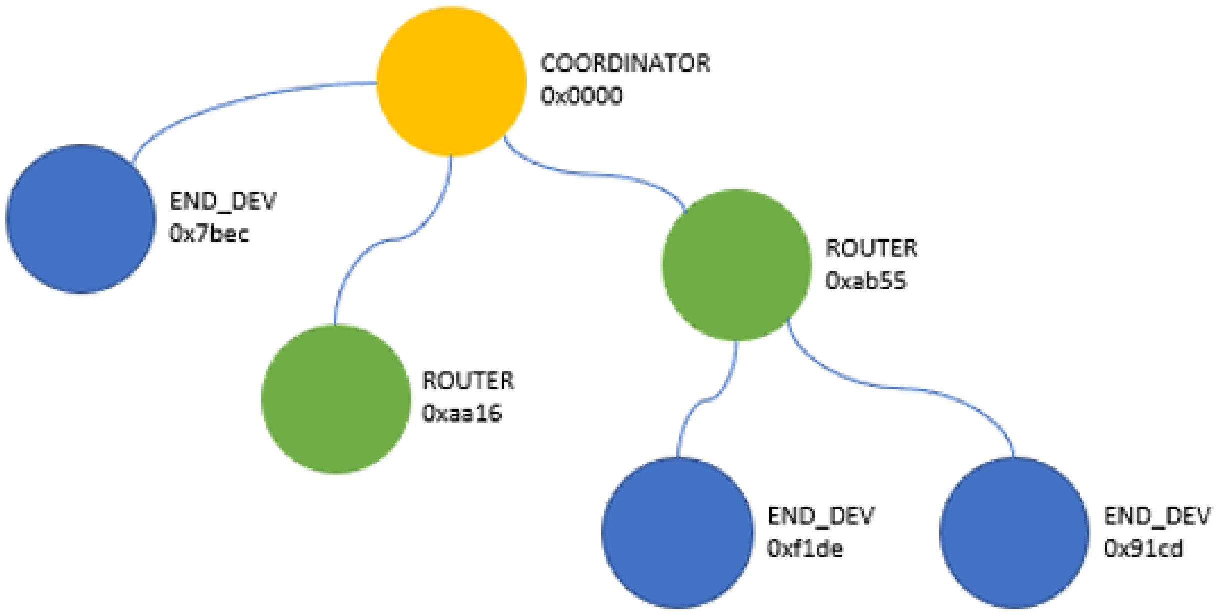
Periodically, any end-device and router send its neighbor list to the coordinator. Once all devices join the Zigbee network, it is possible to collect the messages to create a network map.

The code block below shows the progressive connection of nodes. For example, in the second row, the 0xab55 router has only one neighbor, the coordinator. In the ending four rows, the same router has more neighbors as new nodes join the Zigbee network.

```
INFO: ROUTER [0xaa16], ID:0x20323642355050180001C0047 Neighbor:[0x0000],COORD  ,PARENT , LQI:1
32, RSSI:-59
INFO: ROUTER [0xab55], ID:0x2032364235505018003500 3E Neighbor:[0x0000],COORD  ,PARENT , LQI:1
29, RSSI:-60
INFO: END_DEV[0xf1de], ID:0x20323642355050180002B0042 Neighbor:[0xab55],ROUTER ,PARENT , LQI:1
36, RSSI:-57
INFO: END_DEV[0x7bec], ID:0x2032364235505018003300 3F Neighbor:[0x0000],COORD  ,PARENT , LQI:1
21, RSSI:-62
INFO: END_DEV[0x91cd], ID:0x203236423550501800390043 Neighbor:[0xab55],ROUTER ,PARENT , LQI:1
36, RSSI:-58
INFO: ROUTER [0xaa16], ID:0x20323642355050180001C0047 Neighbor:[0x0000],COORD  ,PARENT , LQI:1
70, RSSI:-59
INFO: ROUTER [0xaa16], ID:0x20323642355050180001C0047 Neighbor:[0xab55],ROUTER ,SIBLING, LQI:1
08, RSSI:0
INFO: ROUTER [0xab55], ID:0x2032364235505018003500 3E Neighbor:[0x0000],COORD  ,PARENT , LQI:1
33, RSSI:-59
INFO: ROUTER [0xab55], ID:0x2032364235505018003500 3E Neighbor:[0x91cd],END_DEV,CHILD  , LQI:1
40, RSSI:-57
INFO: ROUTER [0xab55], ID:0x2032364235505018003500 3E Neighbor:[0xf1de],END_DEV,CHILD  , LQI:1
36, RSSI:-57
INFO: ROUTER [0xab55], ID:0x2032364235505018003500 3E Neighbor:[0xaa16],ROUTER ,SIBLING, LQI:1
23, RSSI:0
```

The below figure shows the network map related to the above neighbor table.

Figure 61. **Zigbee network example**



## 5.5 Environmental and battery status

Any STEVAL-PROTEUS board (router/end-device) can send the temperature and battery-related data collected by its sensor to the Zigbee network.

The coordinator receives the data and displayed them on the serial terminal console.

Data are also visible if the STEVAL-PROTEUS is directly connected to a serial terminal console.

The code block below shows the output of the router, which has the 0xab55 address, about the battery and environmental data information.

These data are printed and sent to the Zigbee network every 20 seconds.

```
[2022-04-15 08:58:38.668] [M4 APPLICATION] INFO: ============================================
===========
[2022-04-15 08:58:38.677] [M4 APPLICATION] INFO: Battery Data Info

[2022-04-15 08:58:38.685] [M4 APPLICATION] INFO: --------------------------------------------
-----------
[2022-04-15 08:58:38.703] [M4 APPLICATION] INFO: Battery Level   =  85.3 %
[2022-04-15 08:58:38.718] [M4 APPLICATION] INFO: Battery Voltage =  4061 mV
[2022-04-15 08:58:38.718] [M4 APPLICATION] INFO: Battery Status  = DISCHARGING
[2022-04-15 08:58:38.733] [M4 APPLICATION] INFO:
[2022-04-15 08:58:38.733] [M4 APPLICATION] INFO: ============================================
===========
[2022-04-15 08:58:38.733] [M4 APPLICATION] INFO: Environmental Data Info

[2022-04-15 08:58:38.750] [M4 APPLICATION] INFO: --------------------------------------------
-----------
[2022-04-15 08:58:38.750] [M4 APPLICATION] INFO: Temperature =  28.90C
```

The code block below shows the output of the coordinator related to two messages received from the device, which has the 0xab55 address, about the battery and temperature.

```
[2022-04-15 08:58:38.692] INFO: ==============================================================
========
[2022-04-15 08:58:38.707] INFO: Battery Data and Status
[2022-04-15 08:58:38.707] INFO: --------------------------------------------------------------
--------
[2022-04-15 08:58:38.721] INFO: DEV=0xab55, Battery Level   =  85.3 %
[2022-04-15 08:58:38.737] INFO: DEV=0xab55, Battery Voltage =  4061 mV
[2022-04-15 08:58:38.737] INFO: DEV=0xab55, Battery Status  =  DISCHARGING
[2022-04-15 08:58:38.754] INFO:
[2022-04-15 08:58:38.754] INFO: ==============================================================
========
[2022-04-15 08:58:38.762] INFO: Environmental Data Info
[2022-04-15 08:58:38.762] INFO: -------------------------------------------------------------
--------
[2022-04-15 08:58:38.762] INFO: DEV=0xab55, Temperature =  28.90C
```

## 5.6 RMS speed, acceleration peak, frequency status, and data messages

RMS speed, acceleration peak, frequency status, and data messages are collected every 20 seconds. You can see them through the PC serial terminal console connected to the STEVAL-PROTEUS or to the coordinator.

These data are sent through separate messages to the coordinator. Once received, they are visible through the PC serial terminal console.

The following code block shows the time domain and frequency status data for the Zigbee router with 0xab55 address.

```
[2022-04-15 08:58:41.290] [M4 APPLICATION] INFO: =========================================
[2022-04-15 08:58:41.300] [M4 APPLICATION] INFO: TD RMS Speed Status and Data
[2022-04-15 08:58:41.309] [M4 APPLICATION] INFO: -----------------------------------------
[2022-04-15 08:58:41.314] [M4 APPLICATION] INFO: X = GOOD     | RMS Speed = 0.04 mm/s
[2022-04-15 08:58:41.318] [M4 APPLICATION] INFO: Y = GOOD     | RMS Speed = 0.02 mm/s
[2022-04-15 08:58:41.334] [M4 APPLICATION] INFO: Z = GOOD     | RMS Speed = 0.16 mm/s
[2022-04-15 08:58:41.334] [M4 APPLICATION] INFO:
[2022-04-15 08:58:41.334] [M4 APPLICATION] INFO: =========================================
[2022-04-15 08:58:41.350] [M4 APPLICATION] INFO: TD Acc Peak Status and Data
[2022-04-15 08:58:41.354] [M4 APPLICATION] INFO: -----------------------------------------
[2022-04-15 08:58:41.356] [M4 APPLICATION] INFO: X = WARNING  | Acc Peak = 0.47 m/s^2
[2022-04-15 08:58:41.374] [M4 APPLICATION] INFO: Y = GOOD     | Acc Peak = 0.38 m/s^2
[2022-04-15 08:58:41.374] [M4 APPLICATION] INFO: Z = GOOD     | Acc Peak = 0.52 m/s^2
[2022-04-15 08:58:41.374] [M4 APPLICATION] INFO:
[2022-04-15 08:58:41.376] [M4 APPLICATION] INFO: =========================================
[2022-04-15 08:58:41.394] [M4 APPLICATION] INFO: Freq Status and Data
[2022-04-15 08:58:41.394] [M4 APPLICATION] INFO: -----------------------------------------
[2022-04-15 08:58:41.406] [M4 APPLICATION] INFO: X = WARNING  | X Max = 0.01 @ 104.43 Hz
[2022-04-15 08:58:41.406] [M4 APPLICATION] INFO: Y = GOOD     | Y Max = 0.01 @ 574.34 Hz
[2022-04-15 08:58:41.409] [M4 APPLICATION] INFO: Z = GOOD     | Z Max = 0.03 @ 104.43 Hz
[2022-04-15 08:58:41.423] [M4 APPLICATION] INFO:
```

The following code block shows the time domain and frequency status data received by the coordinator from the Zigbee router with 0xab55 address.

```
[2022-04-15 08:58:41.334] INFO: =============================================================
========
[2022-04-15 08:58:41.335] INFO: RMS Speed Status and Data
[2022-04-15 08:58:41.341] INFO: -------------------------------------------------------------
--------
[2022-04-15 08:58:41.354] INFO: DEV=0xab55, X = GOOD     | RMS Speed = 0.04 mm/s
[2022-04-15 08:58:41.354] INFO: DEV=0xab55, Y = GOOD     | RMS Speed = 0.02 mm/s
[2022-04-15 08:58:41.357] INFO: DEV=0xab55, Z = GOOD     | RMS Speed = 0.16 mm/s
[2022-04-15 08:58:41.374] INFO:
[2022-04-15 08:58:41.374] INFO: =============================================================
========
[2022-04-15 08:58:41.391] INFO: Acc Peak Status and Data
[2022-04-15 08:58:41.394] INFO: -------------------------------------------------------------
--------
[2022-04-15 08:58:41.394] INFO: DEV=0xab55, X = WARNING  | X Acc Peak = 0.47 m/s^2
[2022-04-15 08:58:41.408] INFO: DEV=0xab55, Y = GOOD     | Y Acc Peak = 0.38 m/s^2
[2022-04-15 08:58:41.408] INFO: DEV=0xab55, Z = GOOD     | Z Acc Peak = 0.52 m/s^2
[2022-04-15 08:58:41.420] INFO:
[2022-04-15 08:58:41.423] INFO: =============================================================
========
[2022-04-15 08:58:41.425] INFO: Freq Status and Data
[2022-04-15 08:58:41.446] INFO: -------------------------------------------------------------
--------
[2022-04-15 08:58:41.446] INFO: DEV=0xab55, X = WARNING  | X Max = 0.01 @ 104.43 Hz
[2022-04-15 08:58:41.451] INFO: DEV=0xab55, Y = GOOD     | Y Max = 0.01 @ 574.34 Hz
[2022-04-15 08:58:41.451] INFO: DEV=0xab55, Z = GOOD     | Z Max = 0.03 @ 104.43 Hz
[2022-04-15 08:58:41.465] INFO:
```

# 6 Download the firmware into the STEVAL-PROTEUS

## 6.1 Prerequisites

Before programming the STEVAL-PROTEUS with the firmware to evaluate its features, you need:

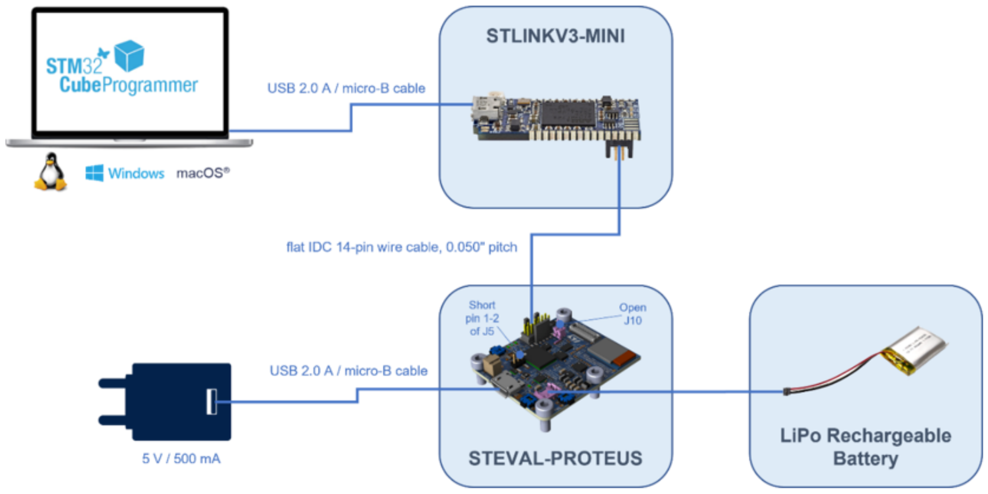
1. a PC/laptop (Windows, MacOS, or Linux)
2. STM32CubeProgrammer software installed on your PC

## 6.2 STEVAL-PROTEUS flash memory programming

The STEVAL-PROTEUS flash memory programming methodology is stack-independent. There are different binaries files for the wireless stack and application.

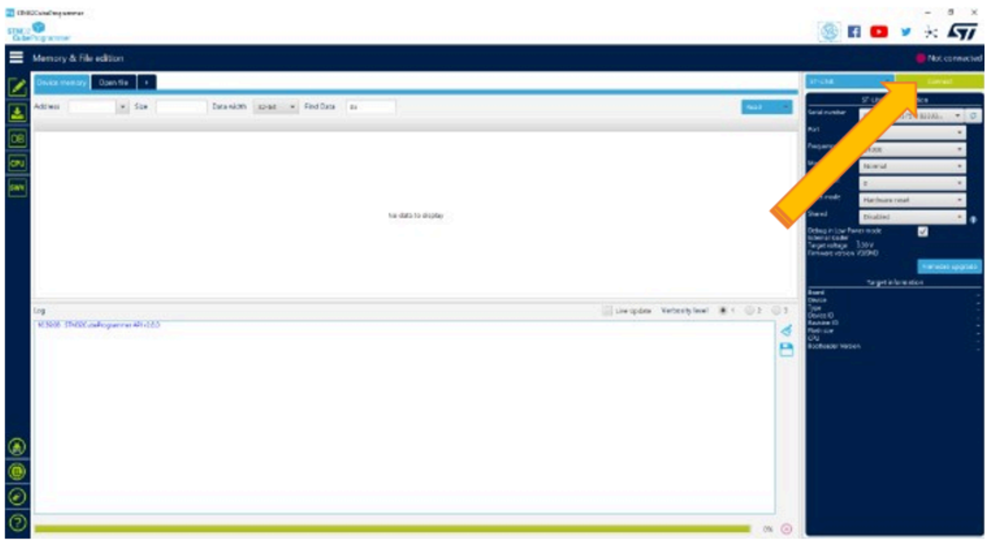
Use the STM32CubeProgrammer software, already installed on your PC, and arrange the connection as shown below.

Figure 62. STEVAL-PROTEUS flash programming scenario


STM32 CubeProgrammer
Windows
macOS®
USB 2.0 A / micro-B cable
STLINKV3-MINI
flat IDC 14-pin wire cable, 0.050" pitch
Short pin 1-2 of J5
Open J10
USB 2.0 A / micro-B cable
5 V / 500 mA
STEVAL-PROTEUS
LiPo Rechargeable Battery

Launch the STM32CubeProgrammer software and connect the attached board, as highlighted below.

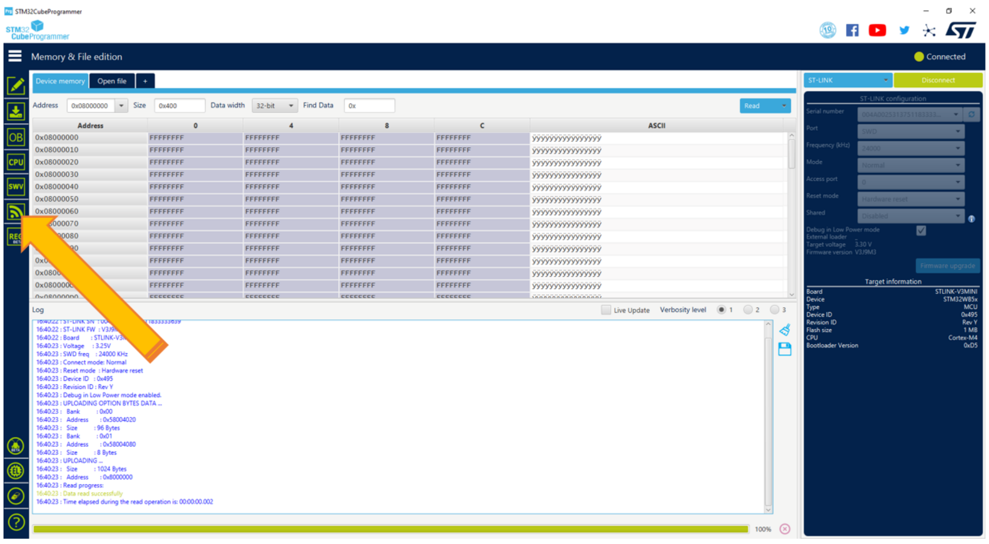
Figure 63. Connecting the board



### 6.2.1 Install the wireless coprocessor binary

**Step 1.** Select “Firmware Upgrade Services”.

Figure 64. **Firmware upgrade services**



```
Address     0x08000000   Size  0x400   Data width  32-bit   Find Data  0x

Address      0         4         8         C         ASCII
0x08000000   FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000010   FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000020   FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000030   FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000040   FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000050   FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000060   FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

```
Log
16:40:22 : ST-LINK FW  : V3J9
16:40:22 : Board       : STLINK-V3
16:40:23 : Voltage     : 3.25V
16:40:23 : SWD freq    : 24000 KHz
16:40:23 : Connect mode: Normal
16:40:23 : Reset mode  : Hardware reset
16:40:23 : Device ID   : 0x495
16:40:23 : Revision ID : Rev Y
16:40:23 : Debug in Low Power mode enabled.
16:40:23 : UPLOADING OPTION BYTES DATA ...
16:40:23 :   Bank          : 0x00
16:40:23 :   Address       : 0x58004020
16:40:23 :   Size          : 96 Bytes
16:40:23 :   Bank          : 0x01
16:40:23 :   Address       : 0x58004080
16:40:23 :   Size          : 8 Bytes
16:40:23 : UPLOADING ...
16:40:23 :   Size          : 1024 Bytes
16:40:23 :   Address       : 0x8000000
16:40:23 : Read progress:
16:40:23 : Data read successfully
16:40:23 : Time elapsed during the read operation is: 00:00:00.002
```

```
ST-LINK configuration
Serial number      004A002531375118333...
Port               SWD
Frequency (kHz)    24000
Mode               Normal
Access port        0
Reset mode         Hardware reset
Shared             Disabled
Debug in Low Power mode   ✓
External loader    _
Target voltage     3.30 V
Firmware version   V3J9M3

Target information
Board              STLINK-V3MINI
Device             STM32WB5x
Type               MCU
Device ID          0x495
Revision ID        Rev Y
Flash size         1 MB
CPU                Cortex-M4
Bootloader Version 0xD5
```

**Step 2.** Select “Start FUS” (see Figure 65, 1).

**Step 3.** Browse to select the stack binary file you are going to use (see Figure 65, 2).

**Step 4.** Set the right start address for downloading the binary (see Figure 65, 3).

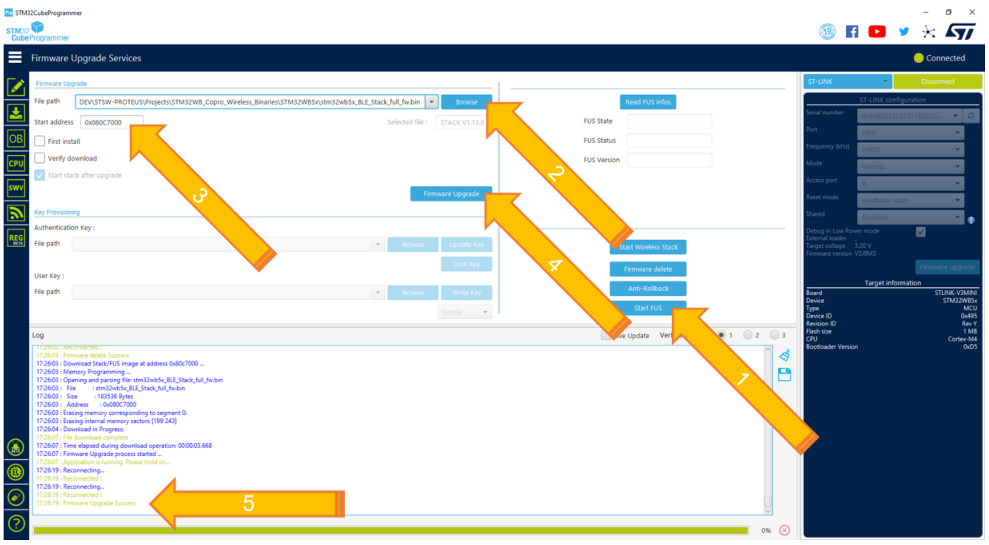
Table 3. **STM32WB coprocessor wireless binaries to be used**

| Platform | Connectivity | Role | Wireless coprocessor binary | Install address |
|---|---|---|---|---|
| STEVAL-PROTEUS | Bluetooth® Low Energy | Slave | stm32wb5x_BLE_Stack_full_fw.bin (V1.13.0) | 0x080C7000 |
| | Zigbee | Router | stm32wb5x_Zigbee_FFD_fw.bin (V1.13.1) | 0x080A4000 |
| | | End-device | stm32wb5x_Zigbee_RFD_fw.bin (V1.13.1) | 0x080B2000 |
| NUCLEO-WB55RG | | Coordinator | stm32wb5x_Zigbee_FFD_fw.bin (V1.13.1) | 0x080A4000 |

**Step 5.** Select “Firmware Upgrade” (see Figure 65, 4).

**Step 6.** Wait for “Firmware Upgrade Success” (see Figure 65, 5).

**Figure 65. Firmware upgrade overview**



```
17:26:02 : Reconnected !
17:26:03 : Firmware delete Success
17:26:03 : Download Stack/FUS image at address 0x80c7000 ...
17:26:03 : Memory Programming ...
17:26:03 : Opening and parsing file: stm32wb5x_BLE_Stack_full_fw.bin
17:26:03 :   File        : stm32wb5x_BLE_Stack_full_fw.bin
17:26:03 :   Size        : 183536 Bytes
17:26:03 :   Address     : 0x080C7000
17:26:03 : Erasing memory corresponding to segment 0:
17:26:03 : Erasing internal memory sectors [199 243]
17:26:04 : Download in Progress:
17:26:07 : File download complete
17:26:07 : Time elapsed during download operation: 00:00:03.668
17:26:07 : Firmware Upgrade process started ...
17:26:07 : Application is running, Please Hold on...
17:26:19 : Reconnecting...
17:26:19 : Reconnected !
17:26:19 : Reconnecting...
17:26:19 : Reconnected !
17:26:19 : Firmware Upgrade Success
```

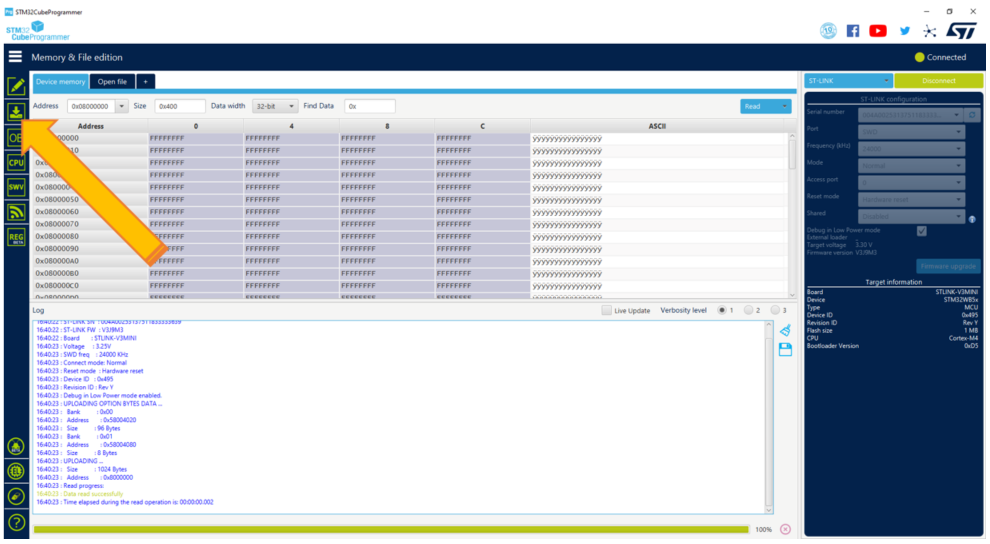
If no error occurs, the wireless coprocessor stack firmware has been successfully downloaded into the STEVAL-PROTEUS flash memory.

## 6.2.2 Install the application firmware

**Step 1.** Launch the STM32CubeProgrammer software and connect the attached board, if not already done.

**Step 2.** Select *“Erasing & Programming”*.

**Figure 66. Erasing & Programming selection**



```
16:40:22 : ST-LINK SN  : 004A00253137511833333639
16:40:22 : ST-LINK FW  : V3J9M3
16:40:22 : Board       : STLINK-V3MINI
16:40:23 : Voltage     : 3.25V
16:40:23 : SWD freq    : 24000 KHz
16:40:23 : Connect mode: Normal
16:40:23 : Reset mode  : Hardware reset
16:40:23 : Device ID   : 0x495
16:40:23 : Revision ID : Rev Y
16:40:23 : Debug in Low Power mode enabled.
16:40:23 : UPLOADING OPTION BYTES DATA ...
16:40:23 :   Bank         : 0x00
16:40:23 :   Address      : 0x58004020
16:40:23 :   Size         : 96 Bytes
16:40:23 :   Bank         : 0x01
16:40:23 :   Address      : 0x58004080
16:40:23 :   Size         : 8 Bytes
16:40:23 : UPLOADING ...
16:40:23 :   Size         : 1024 Bytes
16:40:23 :   Address      : 0x8000000
16:40:23 : Read progress:
16:40:23 : Data read successfully
16:40:23 : Time elapsed during the read operation is: 00:00:00.002
```

**Step 3.** Browse to select the application binary file you are going to use (see, Figure 67, 1).

**Step 4.** Set the right start address to download the binary (seeFigure 67, 2).

**Table 4. STM32WB application binaries to be used**

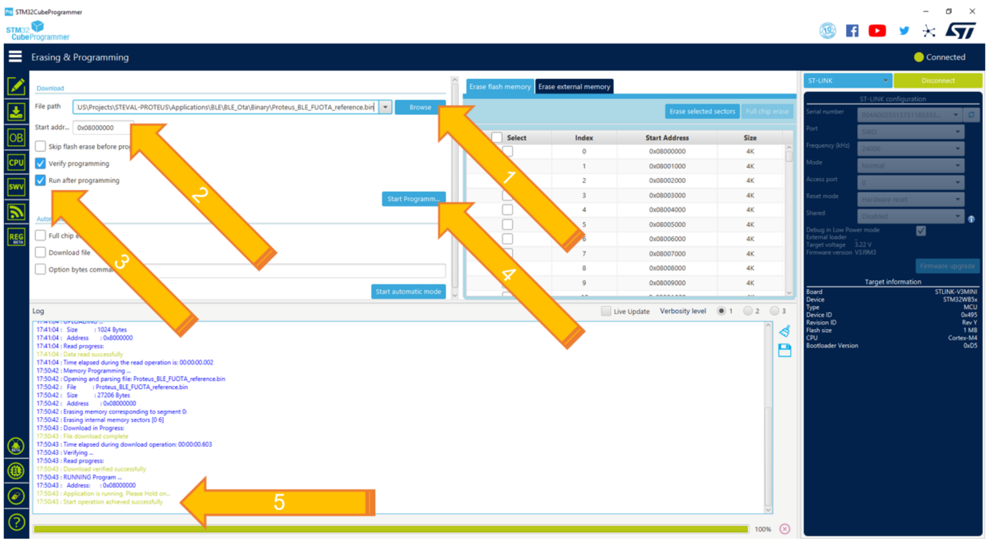
| Platform | Connectivity | Application | Application binary | Install address |
|---|---|---|---|---|
| STEVAL-PROTEUS | BLE | OTA | Proteus_BLE_FUOTA_reference.bin | 0x08000000 |
| | | CbM with OTA | Proteus_BLE_CbM_FUOTA_reference.bin | 0x08007000 |
| | Zigbee | Router CbM | Proteus_Zigbee_RTR_reference.bin | 0x08000000 |
| | | End-device CbM | Proteus_Zigbee_RFD_reference.bin | 0x08000000 |
| NUCLEO-WB55RG | | Coordinator CbM | NucleoWB55_ZB_CRD_CBM_reference.bin | 0x08000000 |

**Step 5.** Check both “Verify programming” and “Run after programming” (see Figure 67, 3).

**Step 6.** Select “Start Programming” (see Figure 67, 4).

**Step 7.** Await the successful completion of all operations (see Figure 67, 5).

**Figure 67. Application firmware installation overview**



File path: US\Projects\STEVAL-PROTEUS\Applications\BLE\BLE_Ota\Binary\Proteus_BLE_FUOTA_reference.bin
Start addr...: 0x08000000
Skip flash erase before programming [ ]
Verify programming [✓]
Run after programming [✓]
Full chip erase [ ]
Download file [ ]
Option bytes commands [ ]

| Select | Index | Start Address | Size |
|---|---|---|---|
| [ ] | 0 | 0x08000000 | 4K |
| [ ] | 1 | 0x08001000 | 4K |
| [ ] | 2 | 0x08002000 | 4K |
| [ ] | 3 | 0x08003000 | 4K |
| [ ] | 4 | 0x08004000 | 4K |
| [ ] | 5 | 0x08005000 | 4K |
| [ ] | 6 | 0x08006000 | 4K |
| [ ] | 7 | 0x08007000 | 4K |
| [ ] | 8 | 0x08008000 | 4K |
| [ ] | 9 | 0x08009000 | 4K |

```
17:41:04 : UPLOADING ...
17:41:04 :   Size          : 1024 Bytes
17:41:04 :   Address       : 0x8000000
17:41:04 : Read progress:
17:41:04 : Data read successfully
17:41:04 : Time elapsed during the read operation is: 00:00:00.002
17:50:42 : Memory Programming ...
17:50:42 : Opening and parsing file: Proteus_BLE_FUOTA_reference.bin
17:50:42 :   File          : Proteus_BLE_FUOTA_reference.bin
17:50:42 :   Size          : 27206 Bytes
17:50:42 :   Address       : 0x08000000
17:50:42 : Erasing memory corresponding to segment 0:
17:50:42 : Erasing internal memory sectors [0 6]
17:50:43 : Download in Progress:
17:50:43 : File download complete
17:50:43 : Time elapsed during download operation: 00:00:00.603
17:50:43 : Verifying ...
17:50:43 : Read progress:
17:50:43 : Download verified successfully
17:50:43 : RUNNING Program ...
17:50:43 :   Address:      : 0x08000000
17:50:43 : Application is running, Please Hold on...
17:50:43 : Start operation achieved successfully
```

| | |
|---|---|
| Serial number | 004A002531375118333... |
| Port | SWD |
| Frequency (kHz) | 24000 |
| Mode | Normal |
| Access port | 0 |
| Reset mode | Hardware reset |
| Shared | Disabled |
| Debug in Low Power mode | ✓ |
| External loader | - |
| Target voltage | 3.22 V |
| Firmware version | V3J9M3 |

Target information

| | |
|---|---|
| Board | STLINK-V3MINI |
| Device | STM32WB5x |
| Type | MCU |
| Device ID | 0x495 |
| Revision ID | Rev Y |
| Flash size | 1 MB |
| CPU | Cortex-M4 |
| Bootloader Version | 0xD5 |

If no error occurs, the application firmware has been successfully downloaded into the STEVAL-PROTEUS flash memory.

## Revision history

Table 5. Document revision history

| Date | Revision | Changes |
|---|---|---|
| 01-Jul-2022 | 1 | Initial release. |
| 21-Jun-2023 | 2 | Updated Figure 5. STEVAL-PROTEUS basic workflow, Figure 6. QR codes for STBLESensClassic, Figure 7. Main page of the STBLESensClassic app for Android, Section 4.4.8 Cloud logging demo.<br>Updated STBLESensor with STBLESensClassic in all the document. |

## Contents

1 **Hardware overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.1 Compatible hardware platform. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.1.1 STEVAL-PROTEUS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2 **Firmware overview** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.1 Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.2 Folder structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
2.3 Bluetooth® Low Energy project details for OTA usage . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
3 **Application scenario** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
3.1 Bluetooth® Low Energy-based application. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
3.1.1 STEVAL-PROTEUS Bluetooth® Low Energy workflow . . . . . . . . . . . . . . . . . . . . . . . . . 9
3.1.2 STBLESensClassic app for Android and iOS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
3.2 Zigbee-based application . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
3.2.1 STEVAL-PROTEUS Zigbee workflow . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
4 **Using the STEVAL-PROTEUS with the ST BLE Sensor Classic application** . . . . . . . . . 12
4.1 Prerequisites . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
4.2 Bluetooth® Low Energy advertising . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
4.3 How to connect a Bluetooth® Low Energy device . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
4.4 How to choose the demo application . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
4.4.1 Environmental demo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
4.4.2 FFT amplitude demo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
4.4.3 Board predictive maintenance demo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
4.4.4 Plot data demo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
4.4.5 Acceleration event demo. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
4.4.6 Switch demo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
4.4.7 LED control demo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
4.4.8 Cloud logging demo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
4.4.9 Board configuration. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
4.4.10 RSSI and battery. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
4.4.11 Firmware update (fast FUOTA) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
4.5 Terminal emulator . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
5 **Using the Zigbee application for the STEVAL-PROTEUS** . . . . . . . . . . . . . . . . . . . . . . . . . 54
5.1 Prerequisites . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
5.2 Zigbee network setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
5.3 Starting a Zigbee network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

5.4 Network information messages . . . . . 56
5.5 Environmental and battery status . . . . . 57
5.6 RMS speed, acceleration peak, frequency status, and data messages . . . . . 58
**6 Download the firmware into the STEVAL-PROTEUS . . . . . 59**
6.1 Prerequisites . . . . . 59
6.2 STEVAL-PROTEUS flash memory programming . . . . . 59
6.2.1 Install the wireless coprocessor binary . . . . . 60
6.2.2 Install the application firmware . . . . . 61
**Revision history . . . . . 63**
**List of tables . . . . . 66**
**List of figures . . . . . 67**

# List of tables

Table 1. Compatible hardware platform overview . . . 2
Table 2. Asset status icon . . . 13
Table 3. STM32WB coprocessor wireless binaries to be used . . . 60
Table 4. STM32WB application binaries to be used . . . 62
Table 5. Document revision history . . . 63

# List of figures

Figure 1. STEVAL-PROTEUS hardware architecture (top view) . . . 3
Figure 2. STEVAL-PROTEUS flash memory map for Bluetooth® Low Energy applications . . . 4
Figure 3. STEVAL-PROTEUS flash memory map for Zigbee applications . . . 4
Figure 4. STSW-PROTEUS architecture . . . 5
Figure 5. STEVAL-PROTEUS basic workflow . . . 10
Figure 6. QR codes for STBLESensClassic . . . 10
Figure 7. Main page of the STBLESensClassic app for Android . . . 12
Figure 8. Device list page of the STBLESensClassic app for Android . . . 13
Figure 9. Connecting a device . . . 14
Figure 10. Choosing the demo . . . 15
Figure 11. Choosing the demo by tapping "back" . . . 15
Figure 12. Environmental demo . . . 16
Figure 13. FFT amplitude demo . . . 17
Figure 14. Board predictive maintenance demo . . . 18
Figure 15. Plotting data . . . 19
Figure 16. Acceleration event demo . . . 20
Figure 17. Switch demo . . . 21
Figure 18. LED control demo . . . 22
Figure 19. Cloud logging . . . 23
Figure 20. Choosing the Proteus PoC application . . . 24
Figure 21. Azure platform - sign in . . . 24
Figure 22. Building your application . . . 25
Figure 23. Creating an API token (1 of 2) . . . 25
Figure 24. Creating an API token (2 of 2) . . . 26
Figure 25. Token creation . . . 26
Figure 26. Configuration with QR code . . . 27
Figure 27. Adding a new device (1 of 2) . . . 28
Figure 28. Adding a new device (2 of 2) . . . 29
Figure 29. New device . . . 29
Figure 30. Connecting the chosen device to the cloud . . . 30
Figure 31. Data to upload to the cloud . . . 30
Figure 32. Azure IoT Central PnP cloud dashboard - device connected . . . 31
Figure 33. Azure IoT Central PnP cloud dashboard - telemetry . . . 31
Figure 34. Azure IoT Central PnP cloud dashboard - overview (1 of 3) . . . 32
Figure 35. Azure IoT Central PnP cloud dashboard - overview (2 of 3) . . . 32
Figure 36. Azure IoT Central PnP cloud dashboard - overview (3 of 3) . . . 33
Figure 37. Board configuration . . . 33
Figure 38. Retrieving the STM32 UID . . . 34
Figure 39. Reading the firmware version . . . 34
Figure 40. Reading detailed information on the firmware version . . . 35
Figure 41. Retrieving the PEM . . . 35
Figure 42. Changing the device name . . . 36
Figure 43. Time and date synchronization . . . 36
Figure 44. Custom commands . . . 37
Figure 45. Reading the Bluetooth® Low Energy connection parameters . . . 38
Figure 46. Reading the MCU temperature . . . 38
Figure 47. Reading the MCU time . . . 39
Figure 48. Reading the MCU date . . . 39
Figure 49. Setting the delay for MotionSP executions . . . 40
Figure 50. Setting the ODR for the linear acceleration . . . 40
Figure 51. Setting the ODR for the angular rate . . . 41
Figure 52. RSSI and battery . . . 42
Figure 53. STEVAL-PROTEUS FUOTA for the application firmware . . . 43

Figure 54. STM32WB FUOTA for the application firmware . . . 44
Figure 55. STEVAL-PROTEUS FUOTA for wireless stack firmware . . . 45
Figure 56. STM32WB FUOTA for wireless stack firmware . . . 46
Figure 57. Searching the WBFUOTA in the device list . . . 47
Figure 58. STEVAL-PROTEUS connection for the terminal emulator . . . 48
Figure 59. STEVAL-PROTEUS connection for the terminal emulator . . . 48
Figure 60. Zigbee network example . . . 54
Figure 61. Zigbee network example . . . 57
Figure 62. STEVAL-PROTEUS flash programming scenario . . . 59
Figure 63. Connecting the board . . . 59
Figure 64. Firmware upgrade services . . . 60
Figure 65. Firmware upgrade overview . . . 61
Figure 66. Erasing & Programming selection . . . 61
Figure 67. Application firmware installation overview . . . 62