STM32CubeWB Nucleo demonstration firmware

Introduction

STM32Cube is an STMicroelectronics original initiative to significantly improve designer's productivity by reducing development effort, time, and cost. STM32Cube covers the whole STM32 portfolio.

STM32Cube includes:

- A set of user-friendly software development tools to cover project development from conception to realization, among which are:
  - STM32CubeMX, a graphical software configuration tool that allows the automatic generation of C initialization code using graphical wizards
  - STM32CubeIDE, an all-in-one development tool with peripheral configuration, code generation, code compilation, and debug features
  - STM32CubeProgrammer (STM32CubeProg), a programming tool available in graphical and command-line versions
  - STM32CubeMonitor-Power (STM32CubeMonPwr), a monitoring tool to measure and help in the optimization of the power consumption of the MCU
- STM32Cube MCU and MPU Packages, comprehensive embedded-software platforms specific to each microcontroller and microprocessor series (such as STM32WB for the STM32WB Series), which include:
  - STM32Cube hardware abstraction layer (HAL), ensuring maximized portability across the STM32 portfolio
  - STM32Cube low-layer APIs, ensuring the best performance and footprints with a high degree of user control over the hardware
  - A consistent set of middleware components such as USB Device, STMTouch (STM32 touch sensing library), STM32_WPAN (Bluetooth® Low Energy 5.0, OpenThread, 802-15-4 MAC), FatFS and FreeRTOS™
  - All embedded software utilities with full sets of peripheral and applicative examples
- STM32Cube Expansion Packages, which contain embedded software components that complement the functionalities of the STM32Cube MCU and MPU Packages with:
  - Middleware extensions and applicative layers
  - Examples running on some specific STMicroelectronics development boards

The STM32CubeWB Nucleo demonstration firmware is built around the STM32Cube hardware abstraction layer (HAL), low-layer (LL) APIs, and board support package (BSP) components, and uses almost the whole STM32 capability to demonstrate Bluetooth® Low Energy peer-to-peer connection between the P-NUCLEO-WB55 USB dongle and Nucleo board.

1 STM32CubeWB main features

STM32CubeWB gathers, in a single package, all the generic embedded software components, required to develop an application on STM32WB microcontrollers. In line with the STM32Cube initiative, this set of components is highly portable, not only to the STM32WB Series but also to other STM32 series.

STM32CubeWB is fully compatible with the STM32CubeMX code generator, which produces initialization code.

The package includes a driver layer (HAL) proposing a set of abstraction services and a low-level hardware layer (LL) proposing a set of register-level functions, together with an extensive set of examples running on STMicroelectronics boards. HAL is available in an open-source BSD license for user convenience.

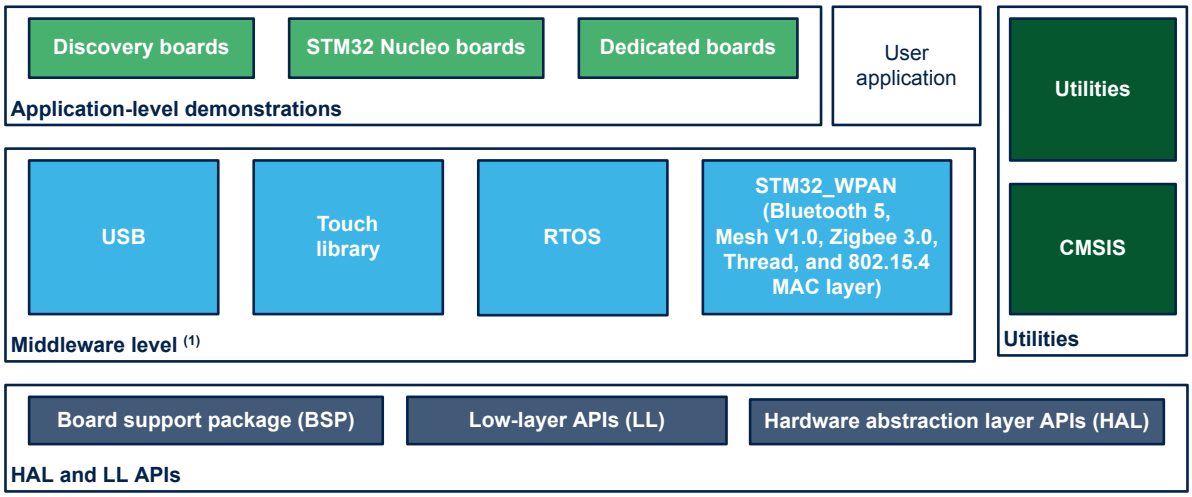
The STM32CubeWB MCU Package also contains a set of middleware components with the corresponding examples. They come in free user-friendly license terms:

- CMSIS-RTOS implementation with FreeRTOS™ open source solution
- Full USB Device stack supporting many classes: Audio, HID, MSC, CDC, and DFU
- STMTouch, touch sensing library solution
- STM32_WPAN, wireless personal area network middleware developed within the STM32WB framework to support Bluetooth® Low Energy (BLE) 5.2, 802.15.4 OpenThread certified stacks and 802-15-4 MAC layer
- FAT file system based on open source FatFS solution

Several applications and demonstrations implementing all these middleware components are also provided in the STM32CubeWB MCU Package.

The block diagram of STM32CubeWB is shown in Figure 1.

Figure 1. STM32CubeWB firmware components



(1) The set of middleware components depends on the product Series.

The STM32WB microcontrollers are based on the Arm® 32-bit Cortex®-M processor.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Getting started with the demonstration

2.1 Hardware requirements

The hardware requirements to start the demonstration application are the following:

- P-NUCLEO-WB55 pack (QF68 Nucleo board and QF48 USB dongle board), using STM32WB Series 32-bit microcontrollers

For more information, refer to the user manual *Bluetooth® Low Energy and 802.15.4 Nucleo pack based on STM32WB Series microcontrollers* (UM2435).

2.2 Hardware configuration

To start using the P-NUCLEO-WB55 pack, follow the recommendations in Section 2.2.1 and Section 2.2.2 in addition to gathering the hardware.

2.2.1 Nucleo board

Check the position of the jumpers on the STM32 Nucleo board as follows:

- JP1 - USB STL
- JP2 ON
- JP3 ON

2.2.2 USB dongle board

Check the position of the jumper on the STM32 USB dongle board, as follows:

- SW2 set to 0

3 Demonstration firmware package

3.1 Demonstration repository

The demonstration enabling the Bluetooth® Low Energy peer-to-peer connection is composed of two projects:

- *P2P_Server_ota* for the Nucleo board (Support of the over-the-air firmware update)
- *P2P_Client* for the USB dongle board

The demonstration sources are located in the project folders of the STM32Cube package for the Nucleo and USB dongle boards.

Figure 2. **Folder structure - P2P server - Nucleo board**

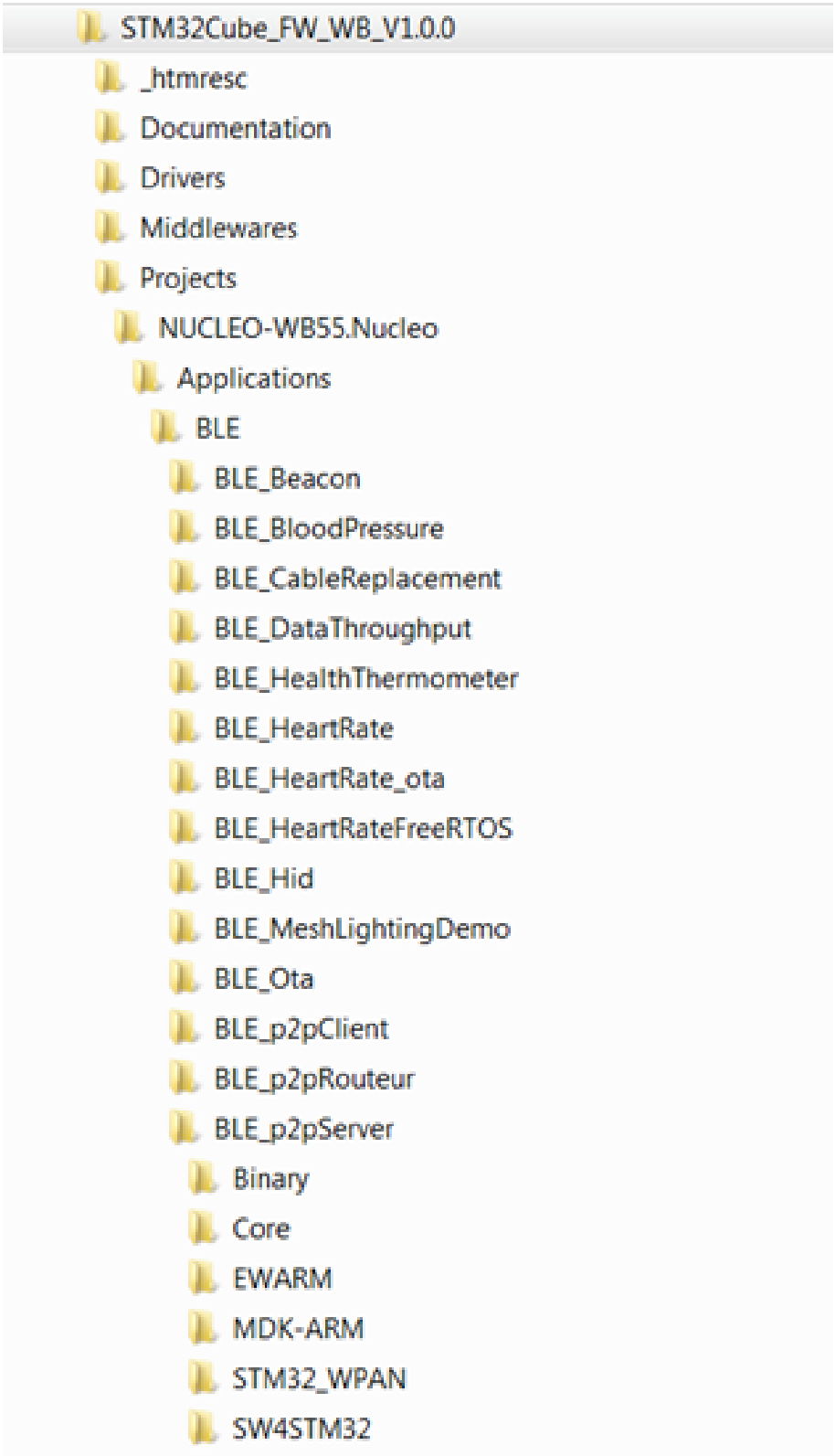
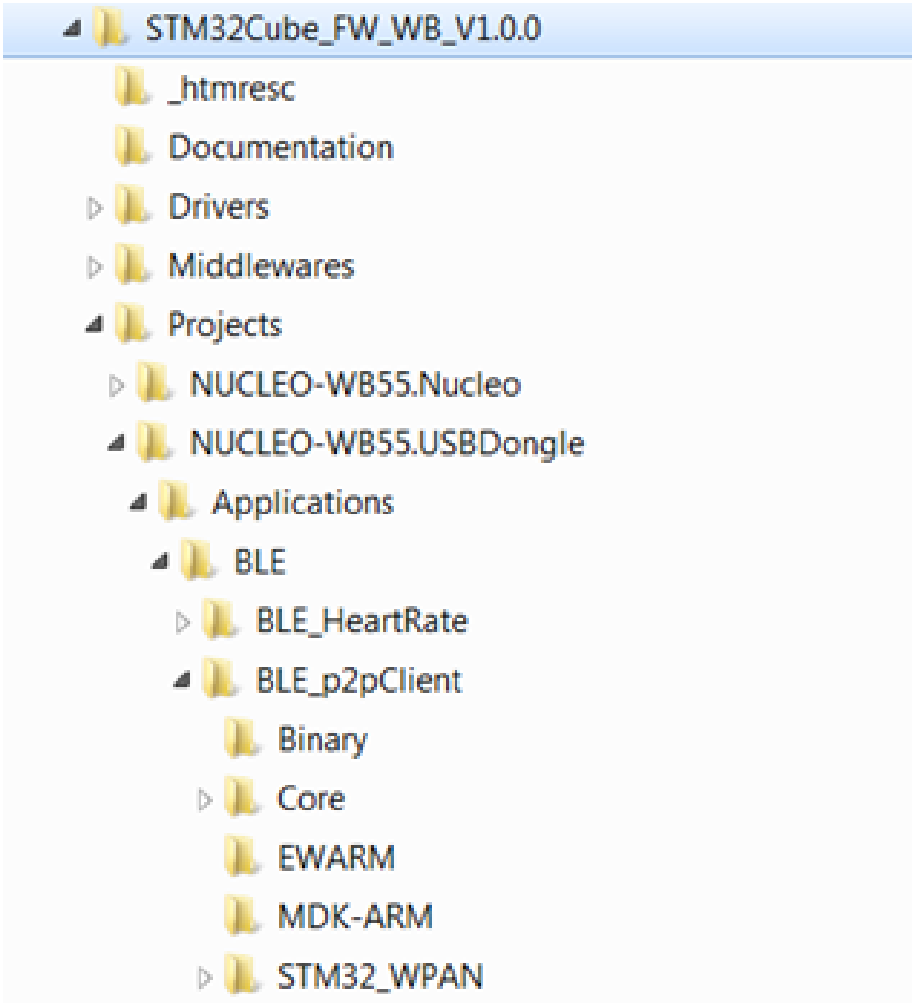


Figure 3. **Folder structure – P2P client - USB dongle board**

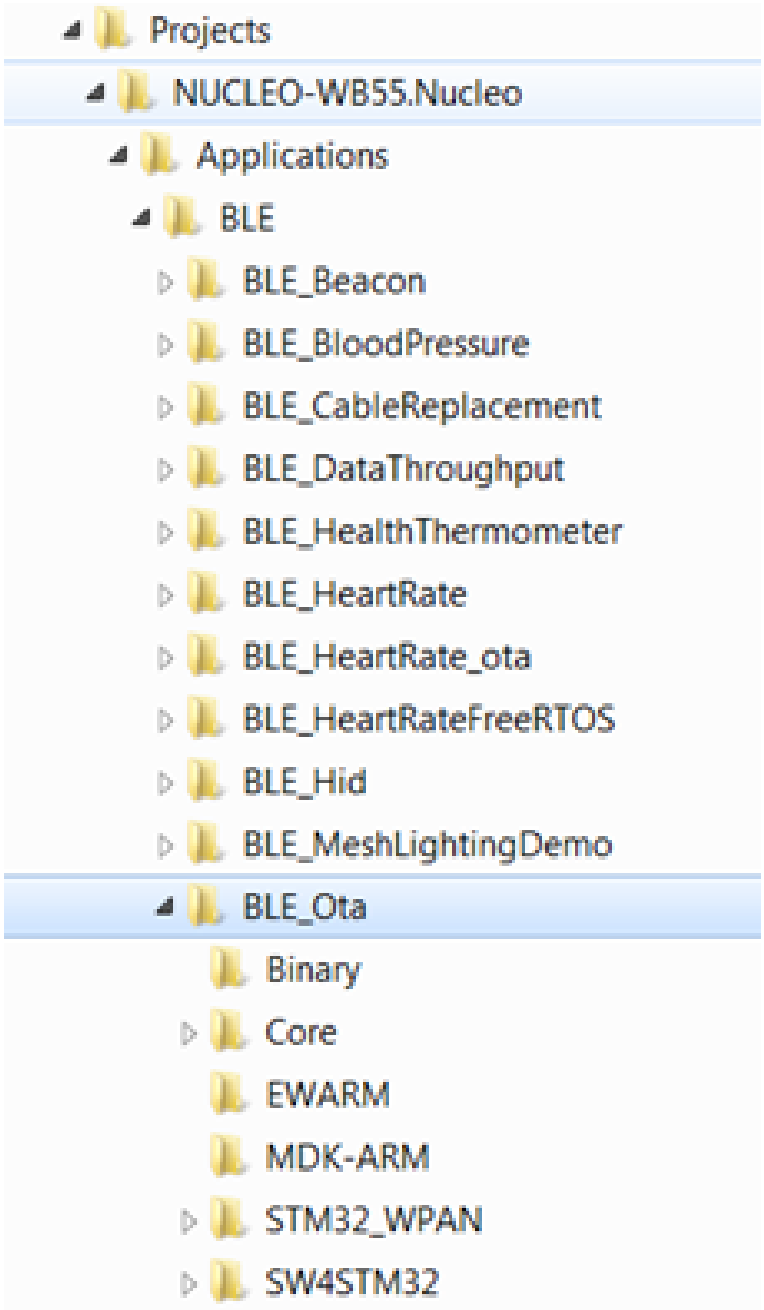


The demonstration available with the Nucleo board enabling the Bluetooth® Low Energy application over-the-air firmware update is composed of three projects:

1. BLE_Ota for the Nucleo board
2. P2P_Server_ota for the Nucleo board (Support of the over-the-air firmware update)
3. BLE_HeartRate_ota for the Nucleo board (Support of the over-the-air firmware update)

The user can retrieve the ready to use demonstration binaries on the resource page of the P-NUCLEO-WB55 at *www.st.com*. (P-NUCLEO-WB55 compiled demo binary). More information is provided in the application note *Building wireless applications with STM32WB Series microcontrollers* (AN5289).

Figure 4. **Folder structure – P2P client - USB dongle board**



4 Functional description of the demonstration

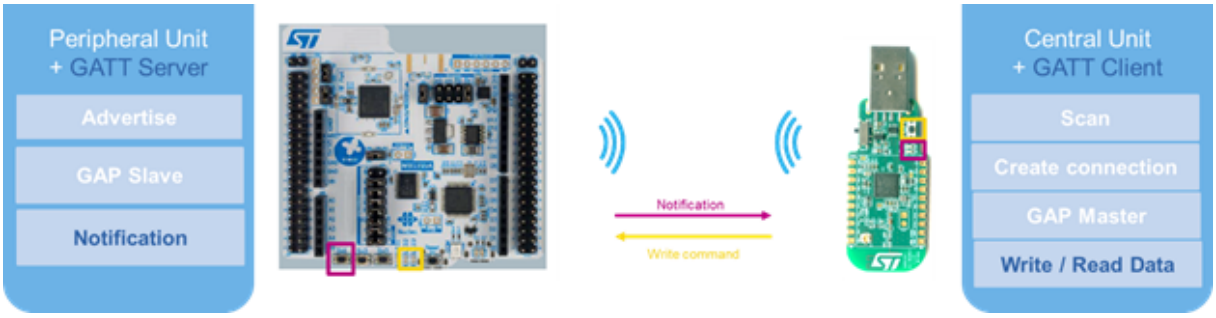
The P-NUCLEO-WB55 pack is provided with the following boards:

- Nucleo board preloaded with Wireless BLE stack and P2P server application with over-the-air firmware update support.
- USB dongle board preloaded with Wireless BLE stack and P2P client application

The combination of the different components results in the following different demonstrations:

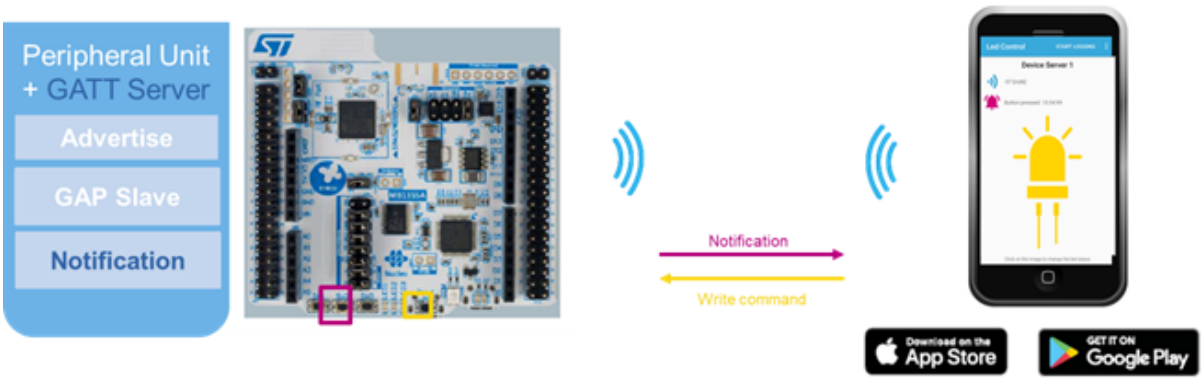
- P2P server and P2P client communication:

Figure 5. P2P server – P2P client demonstration



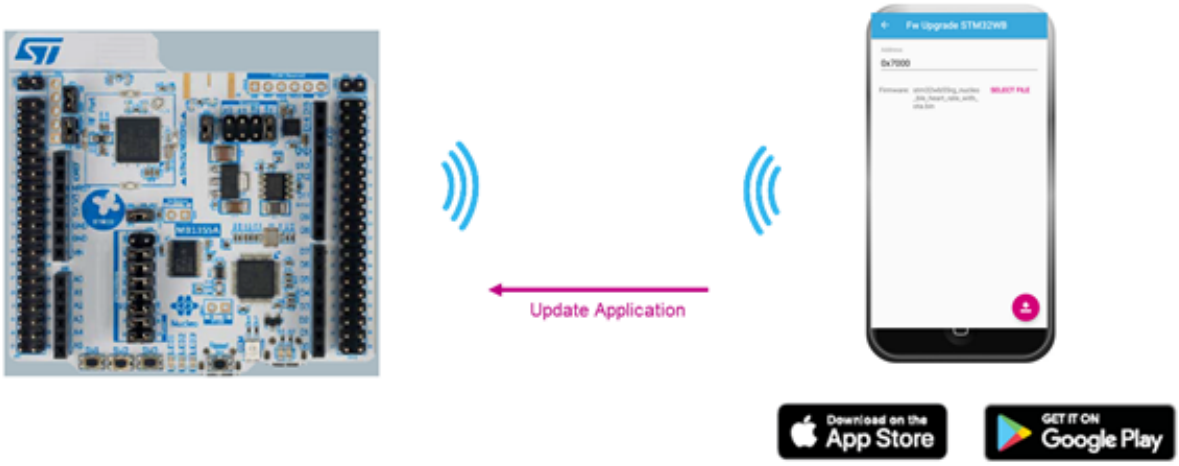
- P2P server and ST BLE sensor smartphone application

Figure 6. P2P server connected to ST BLE sensor smartphone application



- Over-the-air firmware update and ST BLE sensor smartphone application

Figure 7. OTA firmware update with ST BLE sensor smartphone application



4.1 P2P server and P2P client sequence

The Peripheral device (*BLE_p2pServer*) starts advertising (during 1 minute), the green LED blinks for each advertising event.

Make sure *BLE_p2pServer* advertises, if not press the reset button or switch off/on to restart advertising.

The Central device (*BLE_p2pClient*) starts scanning when pressing the User button (SW1) on the USB dongle board.

- *BLE_p2pClient* switches on the blue LED.
- Scan request takes about 5 seconds.

Then, it automatically connects to the *BLE_p2pServer*.

- The blue LED turns off and the green LED starts blinking. BLE Connection is done.

When pressing SW1 on the board, the blue LED toggles on the remote one.

The SW1 button can be pressed independently on the GATT client or the GATT server.

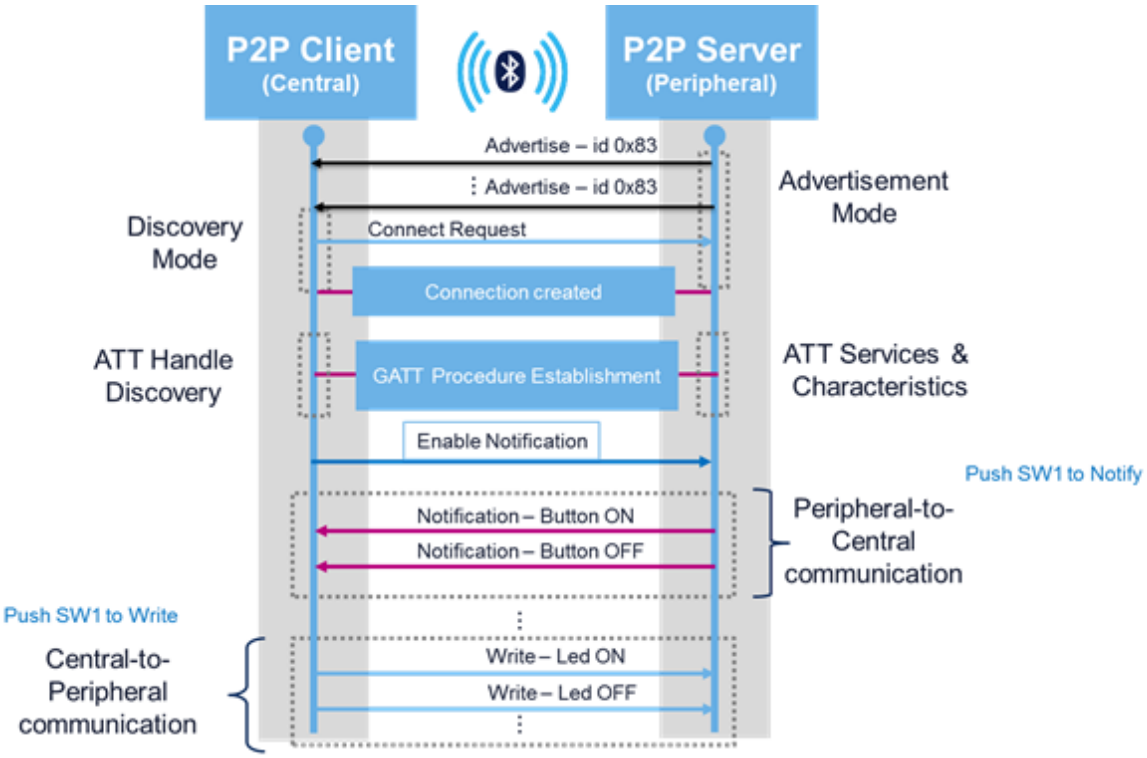
When the P2P server is located on a Nucleo board, the connection interval can be modified from 50 ms to 1 s and vice-versa using SW2.

- The green LED on the two boards blinks for each connection event, which means quickly when 50 ms and slowly when 1 s.
- Passing from 50 ms to 1 s is instantaneous, but from 1 s to 50 ms takes around 10 seconds.

The SW1 event, switching on/off the blue LED, depends on the connection Interval event.

- The delay from SW1 action to blue LED change on the remote side is rather fast.

Figure 8. **P2P server/client communication sequence**



4.2 P2P server and ST BLE sensor smartphone application

The ST BLE sensor mobile application (SMA) supports the detection of the STM32WB P2P server.

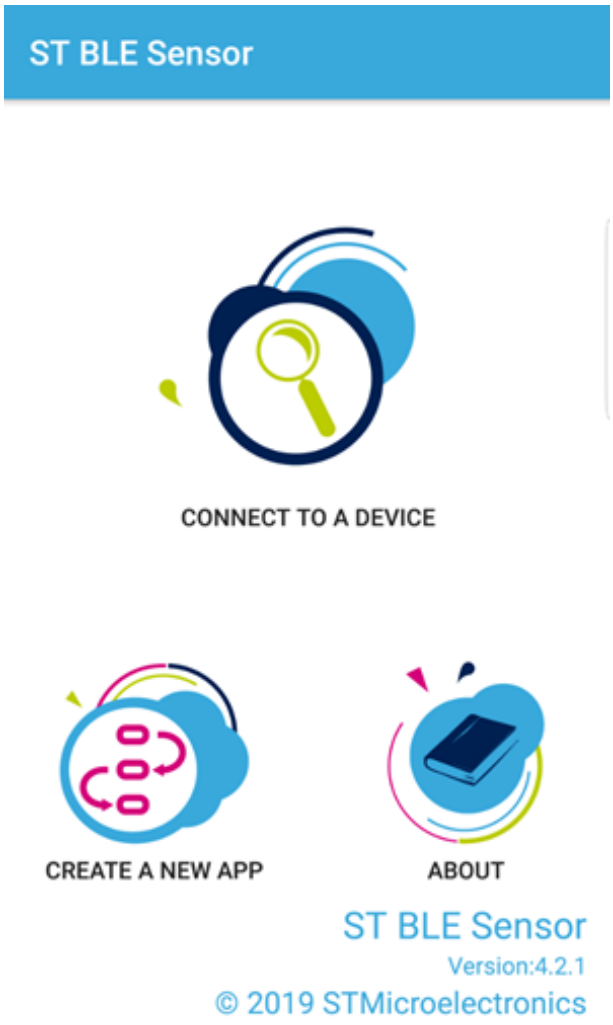
- https://play.google.com/store/apps/details?id=com.st.bluems
- https://itunes.apple.com/us/app/st-bluems/id993670214?mt=8

The peripheral device (BLE_p2pServer) starts advertising (during 1 minute), the green LED blinks for each advertising event.

Make sure BLE_p2pServer advertises, if not press the reset button or switch OFF/ON to restart advertising.

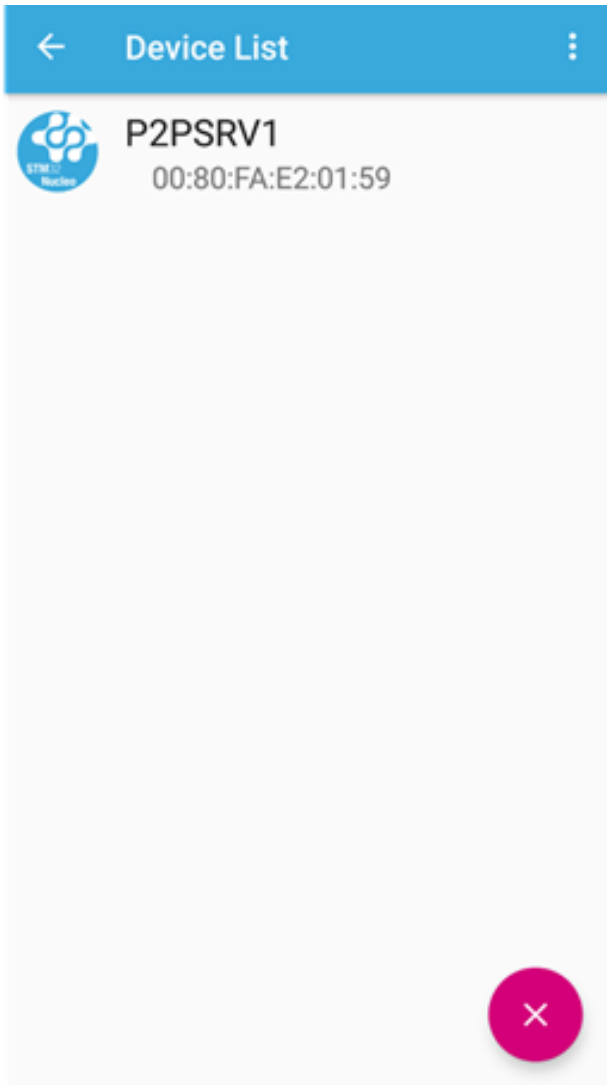
Open the *ST BLE Sensor Application* and push the picture above *CONNECT TO A DEVICE* to scan remote devices.

Figure 9. **ST BLE sensor smartphone application**



All BLE_p2pServer boards in advertising mode are detected and displayed with the local name and their Bluetooth® address.

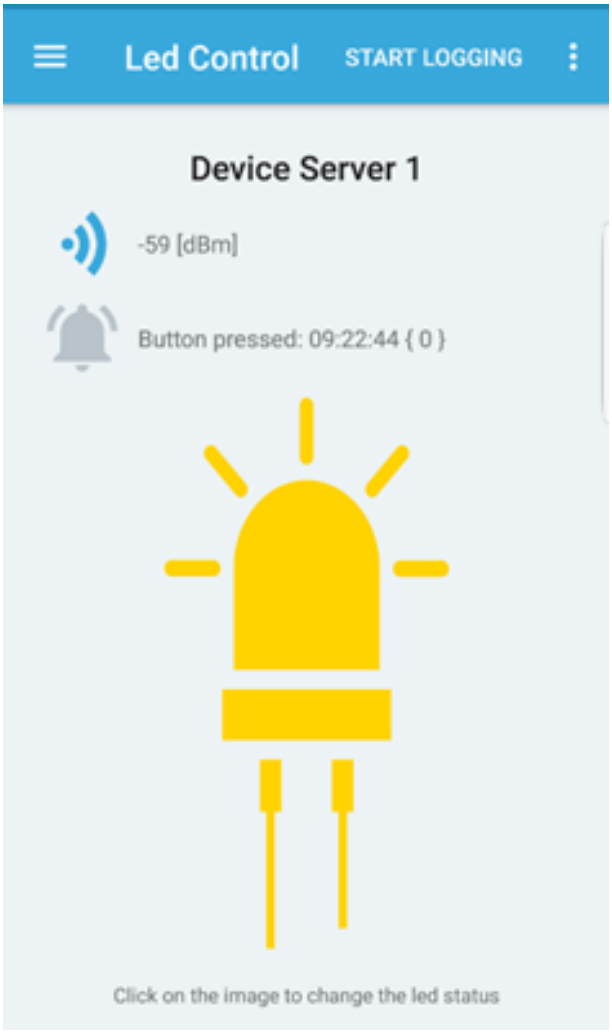
Figure 10. **ST BLE sensor – Scan result**



- Select the device to connect

- Once connection established:
  - On the Nucleo board (Server device), press the SW1 button to toggle the smartphone button status

Figure 11. ST BLE sensor – Server button pressed result



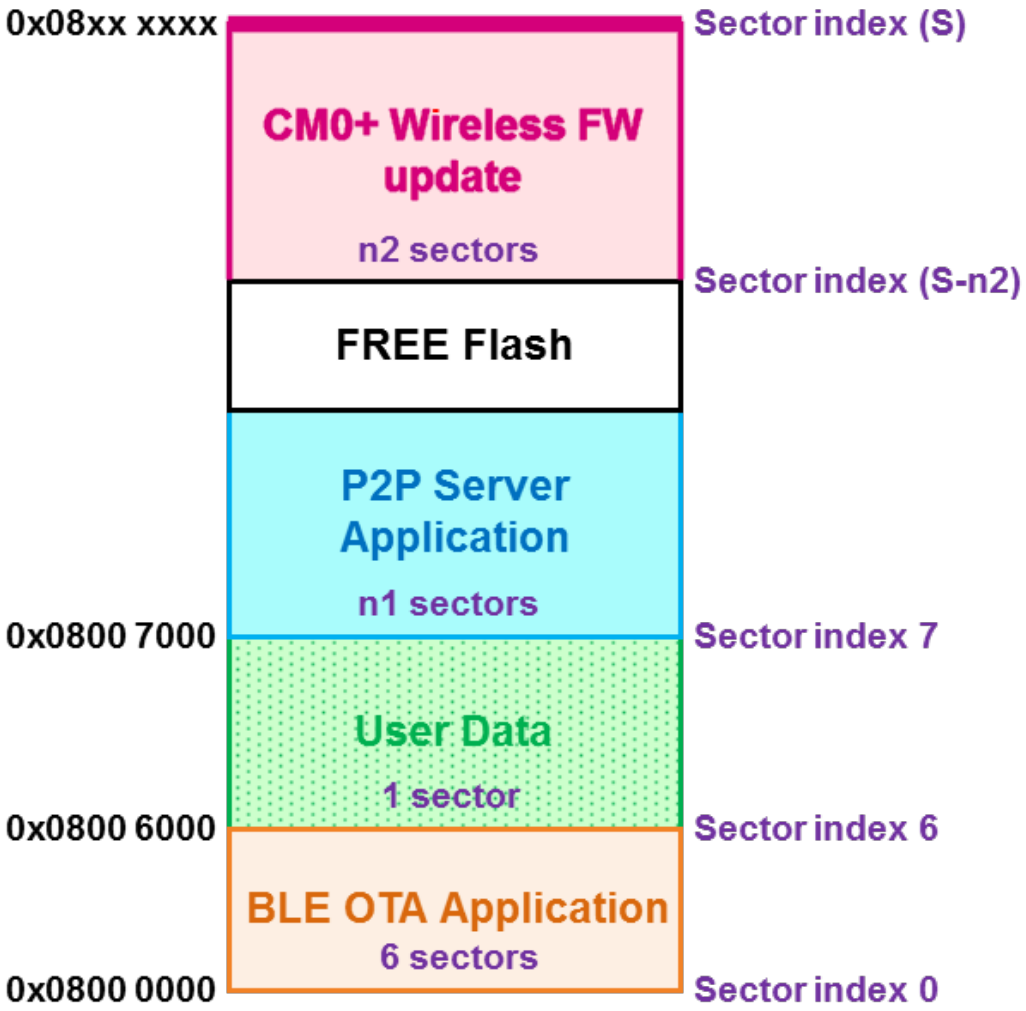
  - On the smartphone, push the lamp to switch ON/OFF the Nucleo board blue LED1.

Figure 12. ST BLE sensor – Smartphone button pressed result

4.3 Over-the-air (OTA) firmware update for the application

By default, the Nucleo board is preloaded with the BLE OTA and P2P server application as described below:

Figure 13. ST BLE sensor – Server button pressed result



The BLE OTA is a standalone binary/application that cannot be updated. It can either:

- Jump to an existing application – sector index 7
- Run and install the ST BLE OTA service to upload any data from a remote in a specified area.

The loaded applications at Sector index 7 must support the *Reboot* characteristics, like in the following examples:

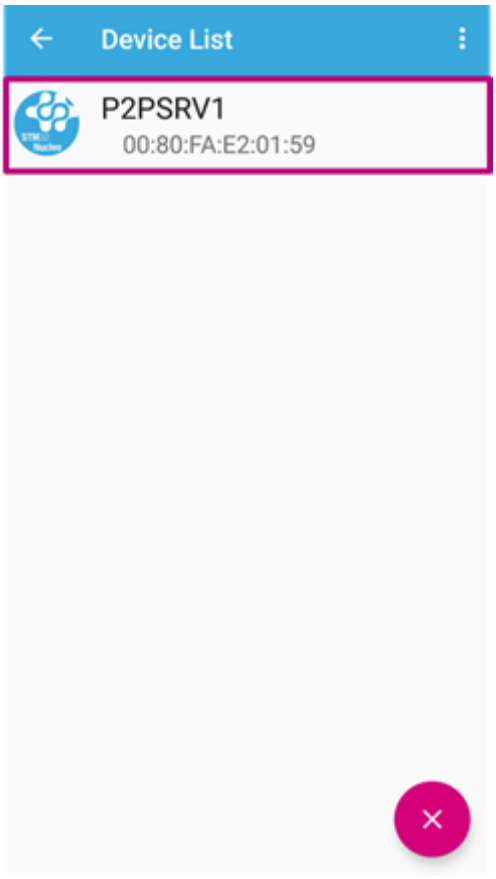
- `Projects\NUCLEO-WB55.Nucleo\Applications\BLE\BLE_HeartRate_ota\Binary\BLE_p2pServer_ota_reference.bin`
- `Projects\ NUCLEO-WB55.Nucleo\Applications\BLE\BLE_p2pServer_ota\Binary\BLE_HeartRate_ota_reference.bin`

The smartphone application *ST BLE Sensor* supports the over-the-air firmware update. The *.bin* of the compiled applications to update must be copied into the smartphone memory.

Here are the steps to update the application:

1. Connect to P2P_Server Application.

Figure 14. ST BLE sensor – Connect to P2PSRV1



2. Move to Reboot Options panel.
3. Select *Application* and click on *Select File* to choose the binary file of the Heart Rate Application to upload on the Nucleo board (`BLE_HeartRate_ota_reference.bin`).

4. Start the reboot of the Nucleo board application by clicking on the Reboot button

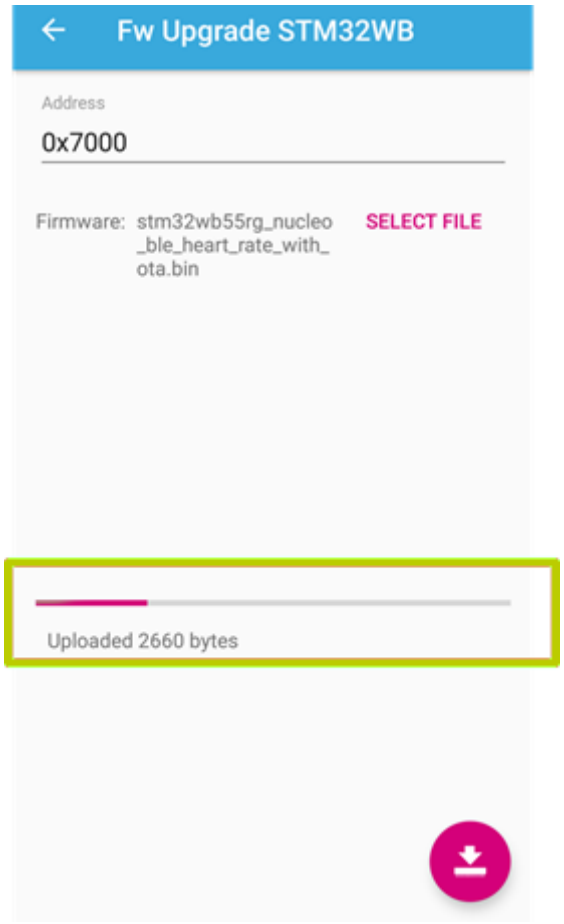
Figure 15. ST BLE sensor – Reboot options panel

5. On the Firmware Update STM32WB panel, choose the address to upload the application binary file (default: 0x7000)

Figure 16. ST BLE sensor – Firmware update panel

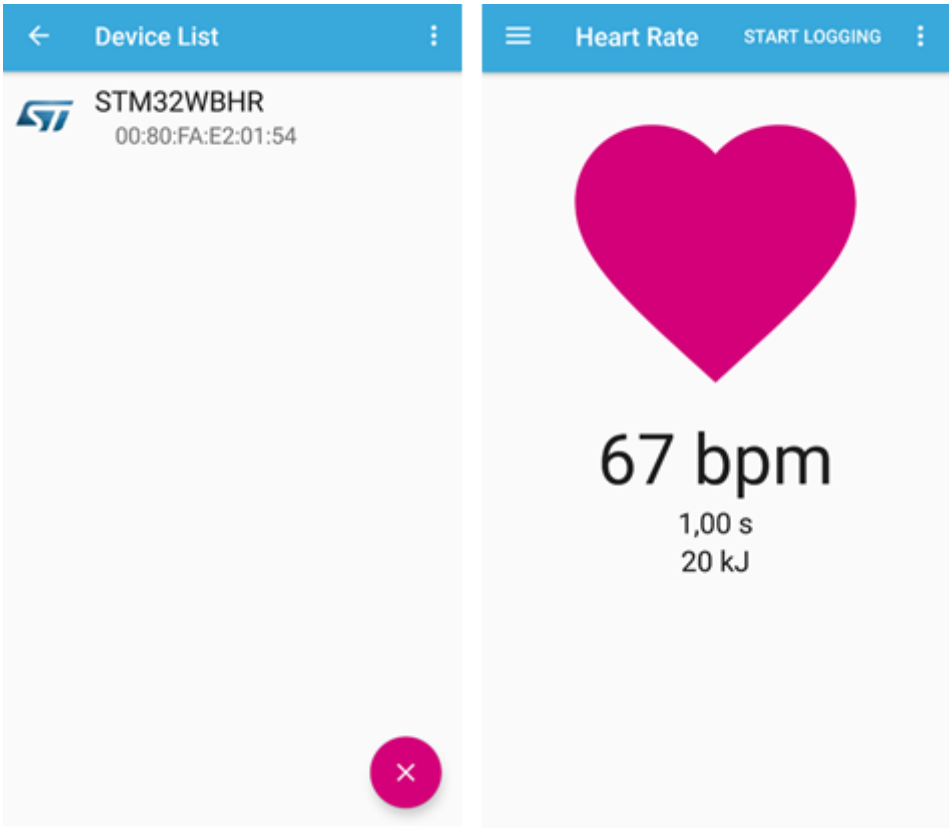
6. At this stage, the new binary file is transferred to the Nucleo board.

Figure 17. ST BLE sensor – Upload ongoing



7. Once the upload is finished, the new application (*BLE Heart Rate*) starts. With the *ST BLE Sensor* application, it is possible to connect to the heart rate sensor and receive the notification.

Figure 18. ST BLE sensor – Heart rate sensor



4.4 HyperTerminal traces

With the Nucleo board applications, the serial COM port of the ST-LINK provides debug traces.

Figure 19. HyperTerminal configuration

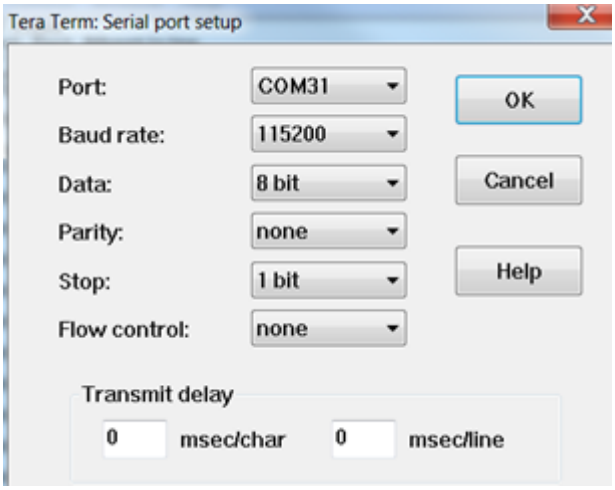
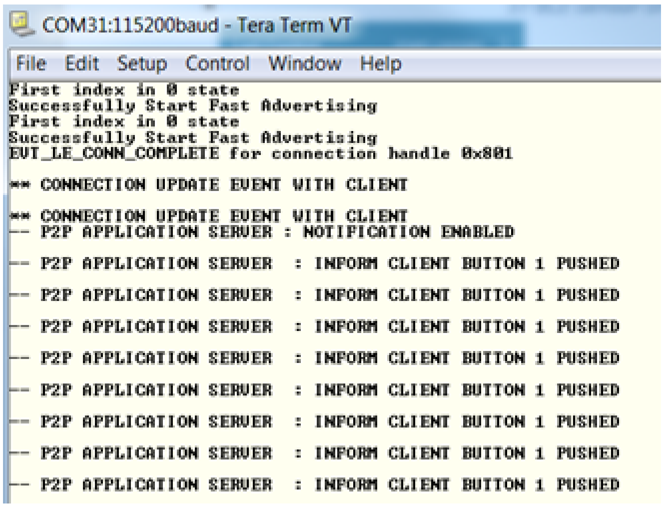


Figure 20. **P2P Server debug traces**



4.5 Programming firmware application

To program the STM32 Nucleo board with the demonstration application, proceed as follows:

1. Install the preferred Integrated Development Environment (IDE)
2. Install the ST-LINK/V2-1 driver available from the STMicroelectronics website

There are two ways of programming the STM32 Nucleo board:

- Method 1:

  Using the preferred in-system programming tool and, depending on the STM32 Nucleo board, upload the `project.hex` or `project.bin` from the firmware package available under `Projects\NUCLEO-WB55.Nucleo\Applications\BLE\”project”\Binary`.
- Method 2:

  Choose one of the supported toolchains, such as IAR Embedded Workbench®, Keil® MDK-ARM, or STM32CubeIDE, and follow the steps below:
  - Open the application folder:
    `Projects\NUCLEO-WB55.Nucleo\Applications\BLE\”project”`
  - Double click on the project file (for example `Project.eww` for EWARM)
  - Rebuild all files: go to [**Project**] and select [**Rebuild all**]
  - Load the project image: go to [**Project**] and select [**Debug**]
  - Run the program: go to [**Debug**] and select [**Go**]

The demonstration software as well as other software examples that allow the user to discover the STM32 microcontroller features are available on the STMicroelectronics website at www.st.com/stm32nucleo.

Revision history

Table 1. **Document revision history**

Date	Revision	Changes
18-Apr-2019	1	Initial release.
27-Apr-2021	2	Updated: • Bluetooth® Low Energy 5.2 supported version • STM32Cube description in Introduction

Contents

1 **STM32CubeWB main features** 2

2 **Getting started with the demonstration** 3

2.1 Hardware requirements 3

2.2 Hardware configuration 3

2.2.1 Nucleo board 3

2.2.2 USB dongle board 3

3 **Demonstration firmware package** 4

3.1 Demonstration repository 4

4 **Functional description of the demonstration** 8

4.1 P2P server and P2P client sequence 9

4.2 P2P server and ST BLE sensor smartphone application 10

4.3 Over-the-air (OTA) firmware update for the application 14

4.4 HyperTerminal traces 19

4.5 Programming firmware application 20

Revision history 21

Contents 22

List of tables 23

List of figures 24

List of tables

Table 1. Document revision history 21

List of figures

Figure 1. STM32CubeWB firmware components . . . 2
Figure 2. Folder structure - P2P server - Nucleo board . . . 5
Figure 3. Folder structure – P2P client - USB dongle board . . . 6
Figure 4. Folder structure – P2P client - USB dongle board . . . 7
Figure 5. P2P server – P2P client demonstration . . . 8
Figure 6. P2P server connected to ST BLE sensor smartphone application . . . 8
Figure 7. OTA firmware update with ST BLE sensor smartphone application . . . 9
Figure 8. P2P server/client communication sequence . . . 10
Figure 9. ST BLE sensor smartphone application . . . 11
Figure 10. ST BLE sensor – Scan result . . . 12
Figure 11. ST BLE sensor – Server button pressed result . . . 13
Figure 12. ST BLE sensor – Smartphone button pressed result . . . 13
Figure 13. ST BLE sensor – Server button pressed result . . . 14
Figure 14. ST BLE sensor – Connect to P2PSRV1 . . . 15
Figure 15. ST BLE sensor – Reboot options panel . . . 16
Figure 16. ST BLE sensor – Firmware update panel . . . 17
Figure 17. ST BLE sensor – Upload ongoing . . . 18
Figure 18. ST BLE sensor – Heart rate sensor . . . 19
Figure 19. HyperTerminal configuration . . . 19
Figure 20. P2P Server debug traces . . . 20

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2021 STMicroelectronics – All rights reserved