STSW-MKBOX-BLEDK software package for BlueNRG-LP

Introduction

The STEVAL-MKBOXPRO (SensorTile.box PRO) is the new ready-to-use programmable wireless box kit for developing any IoT application based on remote data gathering and evaluation. Exploit the full kit potential by leveraging both motion and environmental data sensing. Enhance the connectivity and smartness of whatever environment that you find yourself in.

The STSW-MKBOX-BLEDK is a software package for the BlueNRG-LP Bluetooth® Low Energy system-on-chip supporting Bluetooth 5.2 specifications. Key highlights of this package are as following:

- Smart sensor node firmware with BlueNRG-LP as controller
- Supports both motion and environmental data sensing
- Over the air (OTA) Firmware upgrade
- Communication with ST BLE Sensor Classic app
- Bluetooth® Low Energy examples like beacon, throughput, and serial Port

Figure 1. STEVAL-MKBOXPRO (SensorTile.box PRO) multisensor and wireless connectivity development kit

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
BLE	Bluetooth Low Energy
MHz	Mega Hertz
MCU	Microcontroller unit
I2C	Inter Integrated Circuit

2 Getting started

2.1 Overview

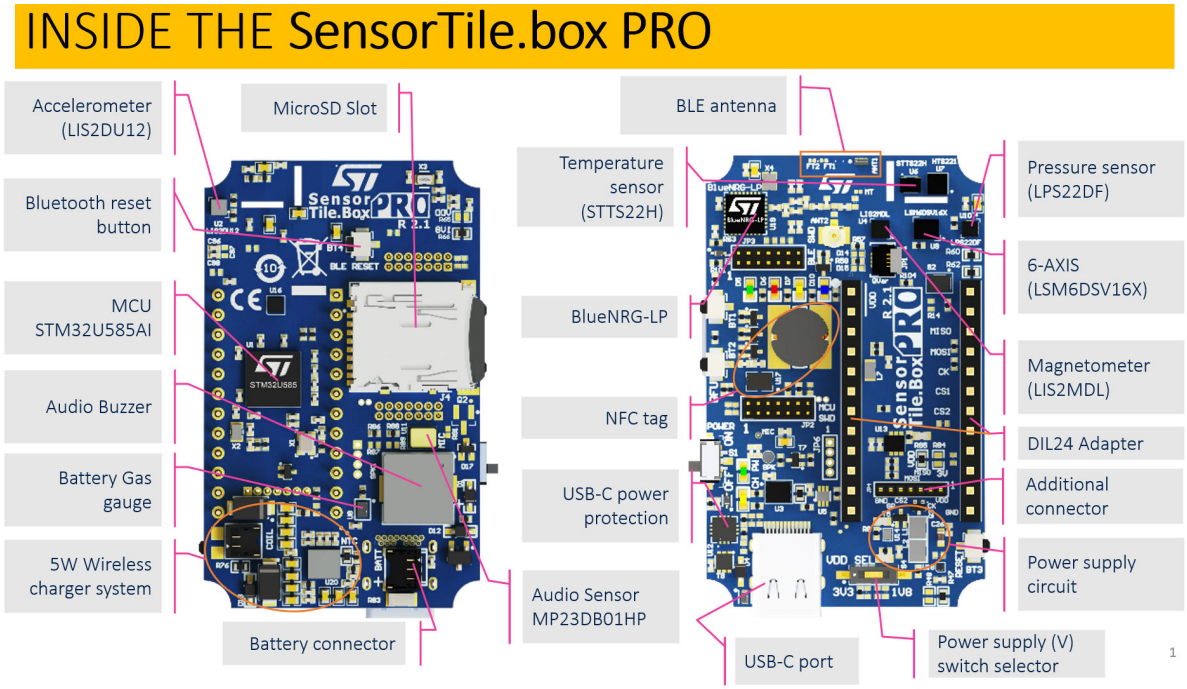
Users can program the BlueNRG-LP to run following BLE Application:

- BLE Sensor Demo
- BLE OTA Reset Manager
- BLE OTA Firmware Manager
- BLE Beacon
- BLE Throughput
- BLE Serial Port Master Slave

There are several ST products used including BLE controllers and sensors on SensorTile.box PRO board:

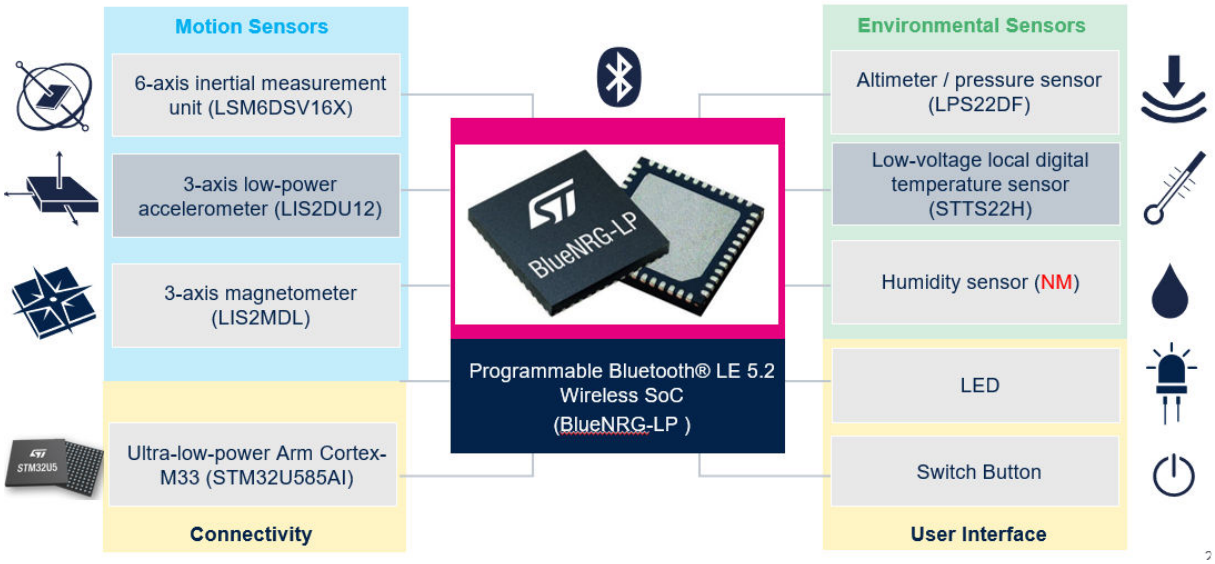
- BlueNRG-LP (Host BLE Microcontroller)
- LSM6DSV16X (iNEMO 3D accelerometer & 3D gyroscope)
- LPS22DF (Pressure Sensor)
- STTS22H (Temperature Sensor)
- LIS2MDL (Magnetic sensor)
- LIS2DU12 (Ultra-low-power 3-axis linear accelerometer)

Figure 2. SensorTile.box PRO PCB Components description



Below is a high-level Block diagram of the board with respect to BlueNRG-LP:

Figure 3. **Example block diagram**



2.2 Folder Structure

The STSW-MKBOX-BLEDK software package files are organized in the following main directories:

- **Firmware**: contains prebuilt binary Bluetooth® Low Energy sample applications
- **Drivers**:
  - **BSP**: SDK drivers providing an API interface to the BlueNRG-LP/BlueNRG-LPS platform hardware resources (LEDs, buttons, sensors, I/O channel)
  - **CMSIS**: BlueNRG-LP/BlueNRG-LPS CMSIS files
  - **Components**: Sensors driver source files
  - **Peripherals_Drivers**: BlueNRG-LP/BlueNRG-LPS drivers for device peripherals (ADC, AES, CRC, DMA, clock, GPIO, I²C, IWDG, LPUART, PWR, RCC, RNG, RTC, SPI, SysTick, TIM, and USART)
- **Middleware**:
  - **Bluetooth® Low Energy**: Bluetooth® Low Energy stack binary library and all the definitions of stack APIs, stack, and events callbacks. Bluetooth® Low Energy stack v3.x configuration header and source files
  - **BLE_Application**: Bluetooth® Low Energy application framework files (Bluetooth® Low Energy stack layers define values, OTA FW upgrade, Bluetooth® Low Energy utilities, master library, GATT, GAP standard profiles, ATT Prepare Write Queue framework)
  - **cryptolib**: AES crypto library
  - **HAL**: hardware abstraction level APIs to abstract certain BlueNRG-LP/BlueNRG-LPS HW/SW features (Crash handler, memory utilities, FIFO management, compiler macros, over-the-air utilities for 2.4 GHz radio proprietary solution, general utilities).
  - **NVMDB**: nonvolatile memory drivers
  - **AESMGR**: AES manager
  - **BLECNTR**: Bluetooth® Low Energy controller manager
  - **PKAMGR**: PKA manager
  - **RNGMGR**: RNG manager
- **Projects**
  - **BLE_Examples:** Bluetooth® Low Energy demonstration application including headers, source files and EWARM and Keil® project files

2.3 Hardware requirements

To use the STSW-MKBOX-BLEDK package you need following:

- STEVAL-MKBOXPRO (SensorTile.box PRO Hardware) – (2 Qty required for BLE_Throughput and BLE Serial Port Application)
- STLINK-V3 Programmer and Cable – (2 Qty required for BLE Throughput and BLE Serial Port Application)
- STEVAL-IDB011V1 (BlueNRG-LP EVK) – (Required for BLE OTA Reset Manager Application)
- PC Windows 10 or higher
- Mobile (Android/IOS)

2.4 Software requirements

To use the STSW-MKBOX-BLEDK package you need following:

- STSW-MKBOX-BLEDK software package.
- IAR ARM v9.20.1/MDK -ARM v5.37.0
- RF-Flasher Utility GUI
- STM32CubePROGRAMMER GUI
- ST BLE Sensor Classic Mobile App for Android/IOS
- BlueNRG-GUI – (Required for BLE OTA Reset Manager Application)
- Tera Term

2.5 Board setup

In the SensorTile.box PRO there are components like sensors, LED, switches connected both BlueNRG-LP and STM32U5 controllers. More than one controller that simultaneously controls these components could lead to unwanted circumstances. To avoid such a case, we need to first mass erase the STM32U5 following below steps:

Step 1. To mass erase the STM32U5 start the DFU Mode:

- Connect the board with the USB cable
- With the board off, press button BT2, then turn on the board and release button BT2

Step 2. Then use the STM32 Cube Programmer: make sure you have the latest STM32CubePROGRAMMER version (old versions might not get recognize the U5 in DFU Mode)

- Select USB, Select the port USB1, click on Connect (see Figure 4)
- Go on the page in Orange Arrow (see Figure 5)
- Click on Full Chip Erase marked by Red circle (see Figure 6)

Figure 4. STM32U5 mass erase steps (step 1)

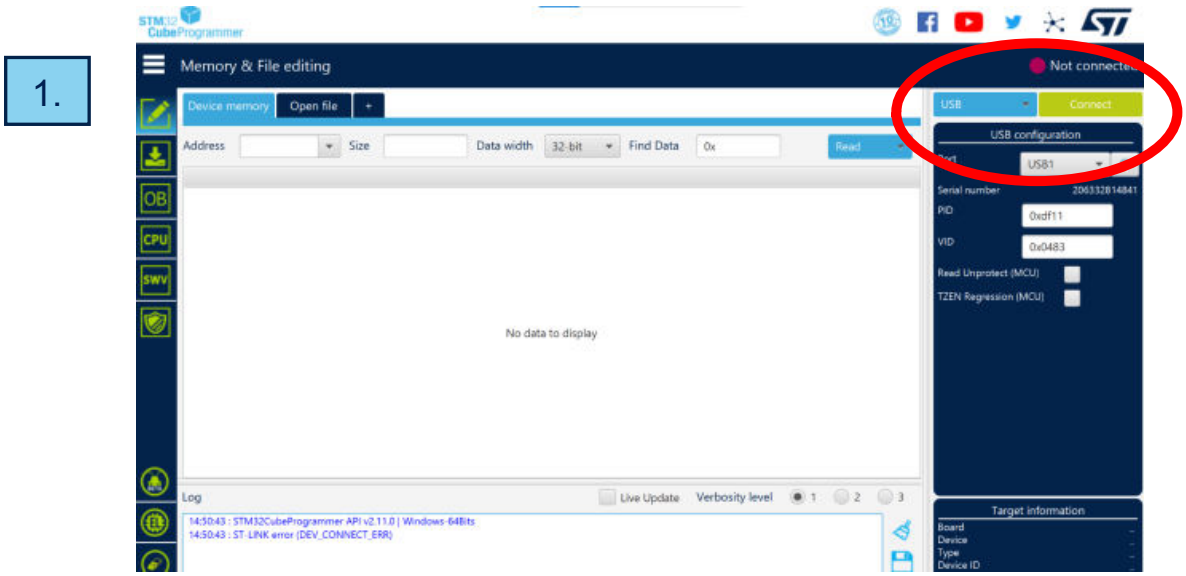


Figure 5. STM32U5 mass erase steps (step 2)

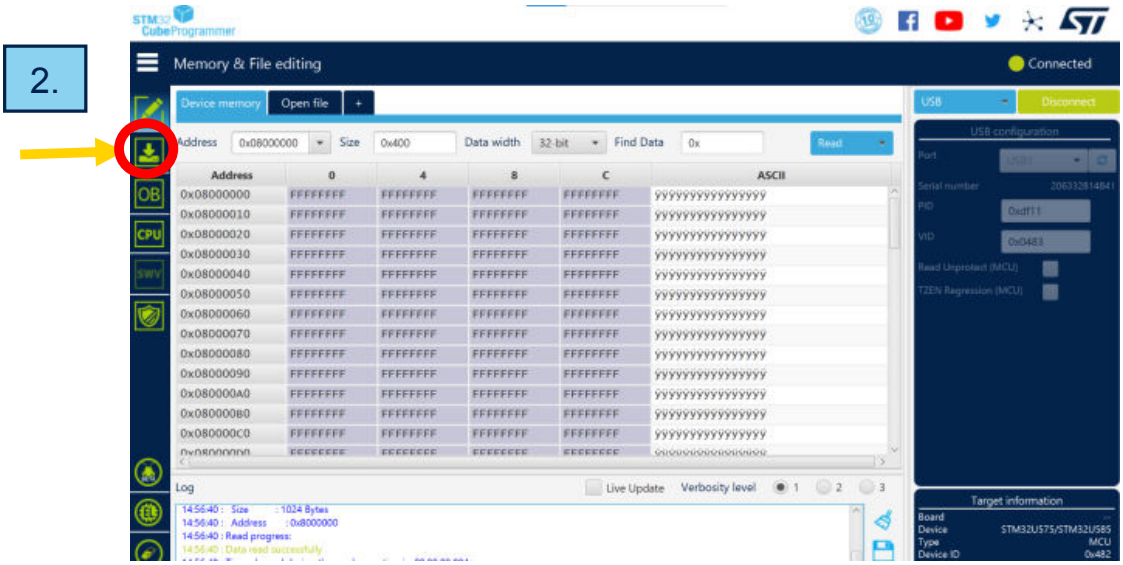
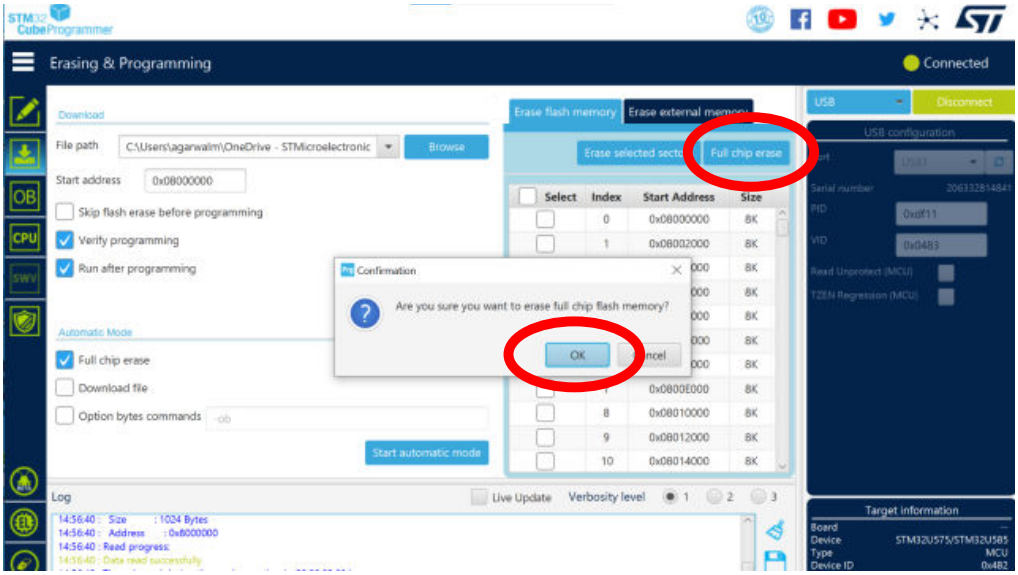


Figure 6. **STM32U5 mass erase steps (step 3)**

3.



3 BLE applications

3.1 BLE Sensor demo

The Bluetooth® Low Energy sensor demo application for ST BLE Sensor Classic app is supported on the SensorTile.box PRO kit. This application shows how to implement a sensor demo custom profile application tailored for interacting with the ST BLE Sensor Classic smartphone app. Once configured and connected, the BlueNRG-LP device sends the data collected from the motion sensor (accelerometer, gyroscope, and magnetometer) and environmental sensor (pressure and temperature) to the ST BLE Sensor Classic smartphone app, which displays this information.

Figure 7. Sensors orientation

Running the application:

Two versions (Android and iOS) of the smartphone ST BLE Sensor Classic app are available for download.

Step 1. Connect the STLINK V3 programmer probe to JP3 header on SensorTile.box PRO board. Connect the programmer to PC with a USB Cable

Step 2. Build the related BLE Sensor Demo application (STSW-MKBOX BLEDK\Projects\BLE_Examples\ BLE_SensorDemo) on the selected platform (pre-built image is available on STSW-MKBOX-BLEDK\Firmware\ BLE_Examples\BLE_SensorDemo\BlueNRG-LP\Release folder and can be loaded through RF Flasher Utility)

Step 3. Load the firmware on the hardware either using IAR/KEIL workspace or flashing Binary File(.hex/.bin) directly using RF Flasher Utility GUI

Note: While flashing BLE_SensorDemo binary(.bin) using RF Flasher Utility enter the image "Flash from Address" as "0x10040000".

Step 4. Remove the STLINK Debugger and Power reset the Board using S1 switch

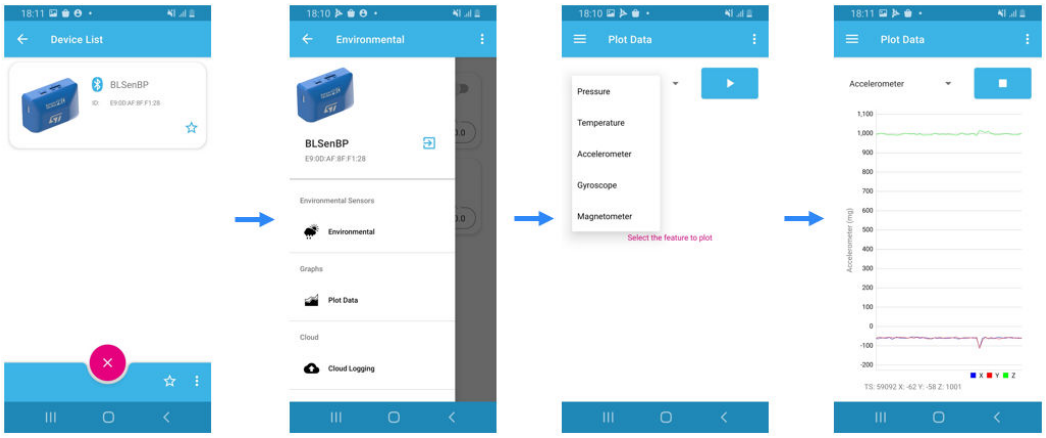
Step 5. Install the app and launch it.

Step 6. The app starts scanning for the BlueNRG-LP Sensor Demo peripheral device. A device called "BLESenBP" appears on the screen

Step 7. Select the "BLESenBP" name and connect to the selected platform. The ST BLE Sensor Classic app enables notifications on the motion characteristic (Accelerometer, Gyroscope, Magnetometer) and on the environment characteristics (pressure and temperature) and displays the received environment characteristic values on the screen

Step 8. Select the ST BLE Sensor Classic Demo, Plot Data window and select sensor options to plot the received values. The sensor values are displayed on a graphical chart.

Figure 8. Expected output on ST BLE Sensor Classic App



3.2 BLE beacon

This is a BLE beacon demo that shows how to configure a BlueNRG-LP device to advertise specific manufacturing data and allow another BLE device to know if it is in the range of the BlueNRG-LP beacon device. It provides a reference example about how to use the BLE over-the-air (OTA) service manager firmware upgrade capability. This example also allows us to use the extended advertising feature to configure beacon also on a secondary advertising channel. Further a specific configuration (PeriodicAdv) allows to configure a beacon with periodic advertising. On BlueNRG-LPS, the AoA_Tag configuration allows to configure a BlueNRG-LPS device as an AoA tag for connectionless scenario.

The beacon demo configures a BlueNRG-LP device in advertising mode (nonconnectable mode) with specific manufacturing data. It transmits advertisement packets at regular intervals which contain the following manufacturing data:

Figure 9. Manufacturing data table

Data field	Description	Notes
Company identifier code	SIG company identifier (1)	Default is 0x0030 (STMicroelectronics)
ID	Beacon ID	Fixed value
Length	Length of the remaining payload	NA
Location UUID	Beacons UUID	It is used to distinguish specific beacons from others
Major number	Identifier for a group of beacons	It is used to group a related set of beacons
Minor number	Identifier for a single beacon	It is used to identify a single beacon
Tx Power	2's complement of the Tx power	It is used to establish how far you are from device

- (1): SIG company identifiers are available on https://www.bluetooth.com/specifications/assigned-numbers/company-identifiers/
- NA: Not Applicable

Note: OTA service manager support requires to build application by enabling CONFIG_OTA_USE_SERVICE_MANAGER (preprocessor, linker) options and through files: OTA_btl.[ch] (refer to Release_with_OTA_ServiceManager IAR workspace).

Running the Application:

Step 1. Connect the STLINK V3 programmer probe to JP3 header on SensorTile.box PRO board. Connect the programmer to PC with a USB Cable

Step 2. Build the related BLE Beacon application (STSW-MKBOX BLEDK\Projects\BLE_Examples\ BLE_Beacon) on the selected platform (pre-built image is available on STSW-MKBOX-BLEDK\Firmware\ BLE_Examples\BLE_Beacon\Release folder and can be loaded through RF Flasher Utility)

Step 3. Load the firmware on the hardware either using IAR/KEIL workspace or flashing Binary File (.hex/.bin) directly using RF Flasher Utility GUI

Step 4. Remove the Debugger, Power reset the Board using S1 switch. Plug the Debugger again to PC for serial terminal

Step 5. Open Serial terminal like Tera-Term and select the serial port. Set the baud rate as 115200

Step 6. The device will start advertising according to the configuration selected (Release, Extended, Periodic) and the Printf logs will be visible in serial terminal

Step 7. Download and Open the BLE scanner mobile App and search for the Beacon Device

Figure 10. BLE Beacon Tera Term Settings

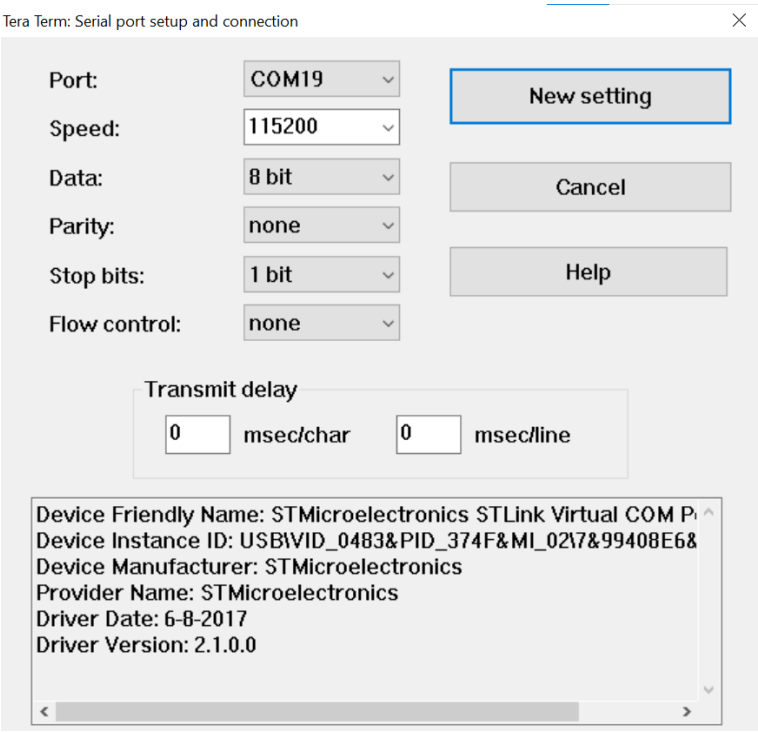


Figure 11. BLE Beacon Tera term output: release configuration

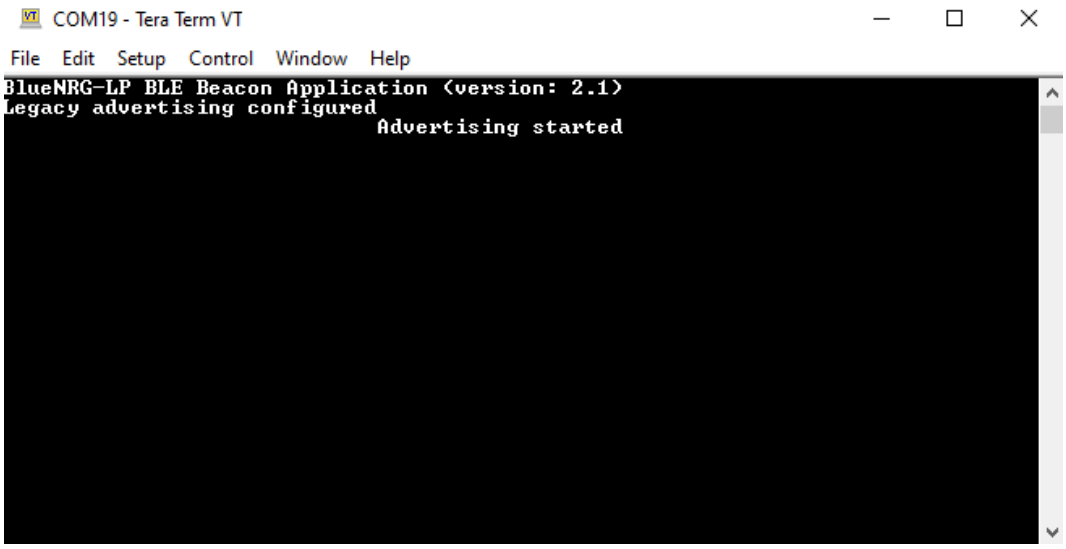


Figure 12. **BLE Beacon Tera term output: extended configuration**

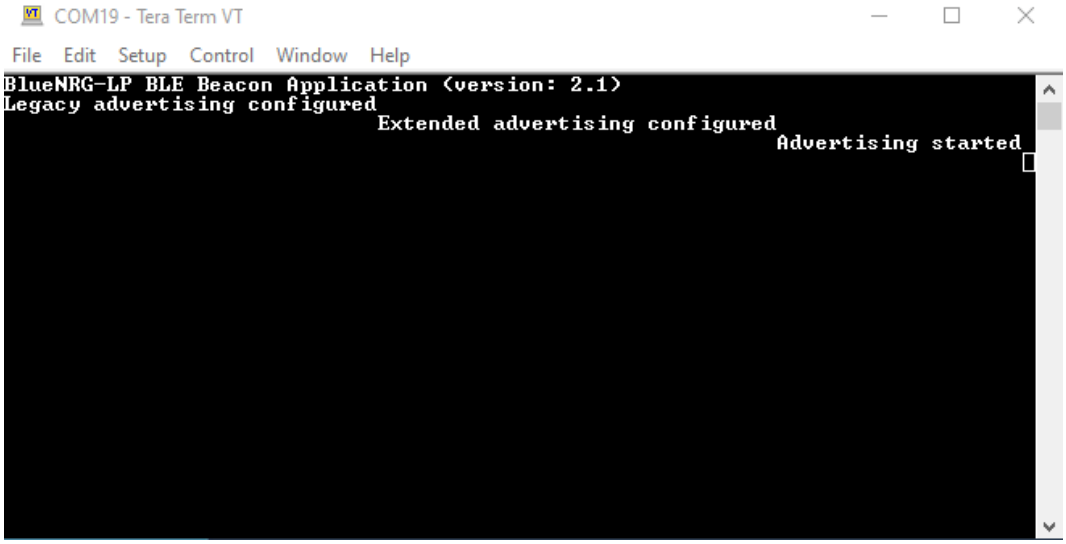
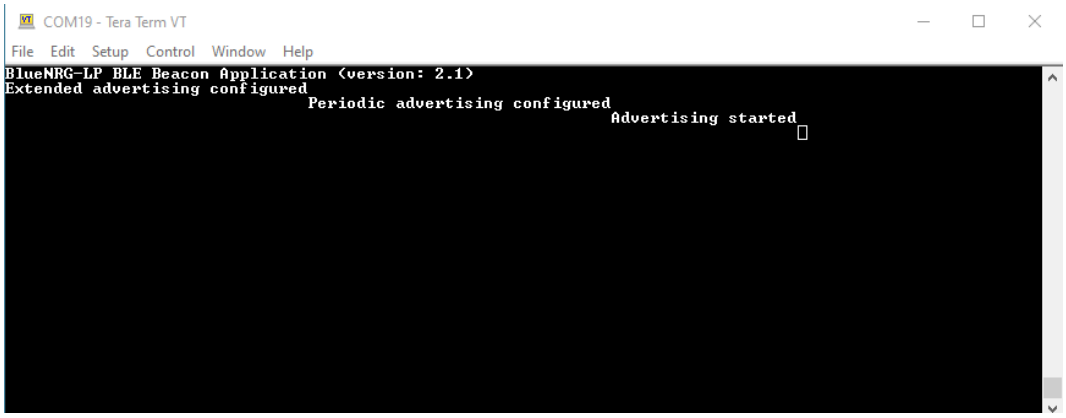


Figure 13. **BLE Beacon Tera term output: periodic configuration**



3.3 BLE Throughput

This Throughput demo has two roles:

- The **server** that exposes the Throughput service. It is the slave
- The **client** that uses the Throughput service. It is the master

The Throughput Service contains two Characteristics:

- The TX Characteristic: the client can enable notifications on this characteristic. When the server has data to be sent, it sends notifications which contains the value of the TX Characteristic
- The RX Characteristic: it is a writable characteristic. When the client has data to be sent to the server, it writes a value into this characteristic

The following interactive options are available on Server side (open HyperTerminal with 115200, 8, None,1, None):

Press u to Send data len update request for 27 bytes

Press m to Send data len update request for 100 bytes

Press U to Send data len update request for 251 bytes

Press z to Enable/disable slow throughput

Press 1 to Change L2C COS MTU value

Press 2 to Change L2C COS MPS value

Press c to Send connection parameter update request

Press f to Enable/disable flushable PDUs

Press e to Toggle notify
Press p to Print APP flags
Press s to Read LE PHY (TX, RX)
Press d to Set LE RX PHY to Coded
Press D to Set LE TX PHY to Coded
Press t to Set LE TX PHY to 1 Mbps
Press r to Set LE RX PHY to 1 Mbps
Press T to Set LE TX PHY to 2 Mbps
Press R to Set LE RX PHY to 2 Mbps
Press x to System reset
Press ? to Print help

The following interactive options are available on Client side (open HyperTerminal with 115200, 8, None,1, None):

Press u to Send data len update request for 27 bytes
Press m to Send data len update request for 100 bytes
Press U to Send data len update request for 251 bytes
Press a to Send ATT_MTU exchange
Press z to Enable/disable slow throughput
Press 1 to Change L2C COS MTU value
Press 2 to Change L2C COS MPS value
Press b to Switch bidirectional test on-off
Press n to Send notifications
Press i to Send indication
Press p to Print APP flags
Press s to Read LE PHY (TX, RX)
Press d to Set LE RX PHY to Coded
Press D to Set LE TX PHY to Coded
Press t to Set LE TX PHY to 1 Mbps
Press r to Set LE RX PHY to 1 Mpbs
Press T to Set LE TX PHY to 2 Mbps
Press R to Set LE RX PHY to 2 Mpbs
Press x to System reset
Press ? to Print help

Note: The Client and Server workspaces (by default) allow to target a unidirectional throughput test: server device sends characteristic notifications (20 bytes) to the client device. The required serial port baud rate is 115200.

Running the Application:

Step 1. Take two pairs each of SensorTile.box PRO and STLINK V3 hardware. Connect the STLINK V3 programmer probe to JP3 header on both SensorTile.box PRO boards. Connect the programmers to PC with USB Cables

Step 2. Program the client side (STSW-MKBOX BLEDK\Projects\BLE_Examples\ BLE_Throughput) on the selected platform (pre-built image is available on STSW-MKBOX- BLEDK\Firmware\ BLE_Examples\BLE_Throughput\Client folder and can be loaded through RF Flasher Utility) and Power reset it

Step 3. The platform is seen on the PC as a virtual COM port. Open the port in a serial terminal tool. The required serial port baud rate is 115200

Step 4. Program the server side (STSW-MKBOX BLEDK\Projects\BLE_Examples\ BLE_Throughput) on the selected platform (pre-built image is available on STSW-MKBOX- BLEDK\Firmware\ BLE_Examples\BLE_Throughput\Server folder and can be loaded through RF Flasher Utility) and Power reset it

While flashing BLE_Throughput binary(.bin) using RF Flasher Utility enter the image “Flash from Address” as “0x10040000”

Step 5. The platform is seen on the PC as a virtual COM port. Open the port in a serial terminal tool. The required serial port baud rate is 115200

Step 6. The two platforms try to establish a connection. As soon as they get connected, the slave continuously sends notification of a characteristic to the client

Step 7. User can play with the provided commands in order to select the following link layer packet length: 27 (default),100 and 251 (maximum allowed) bytes

Step 8. User can also perform an ATT MTU exchange command on server side in order to allow server to increase the ATT_MTU size (247 bytes)

Step 9. He can also enable the bidirectional throughput on client side (client writes on RX characteristic)

Step 10. User can set the PHY to be used for the communication (1Mbps-default, 2Mbps, ...)

Figure 14. BLE Throughput Tera Term output - 1

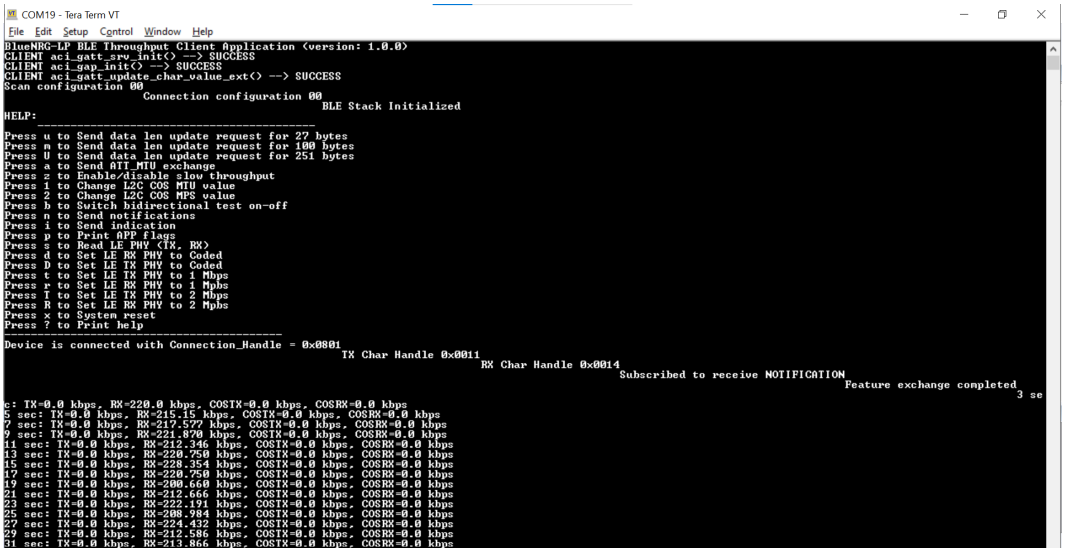
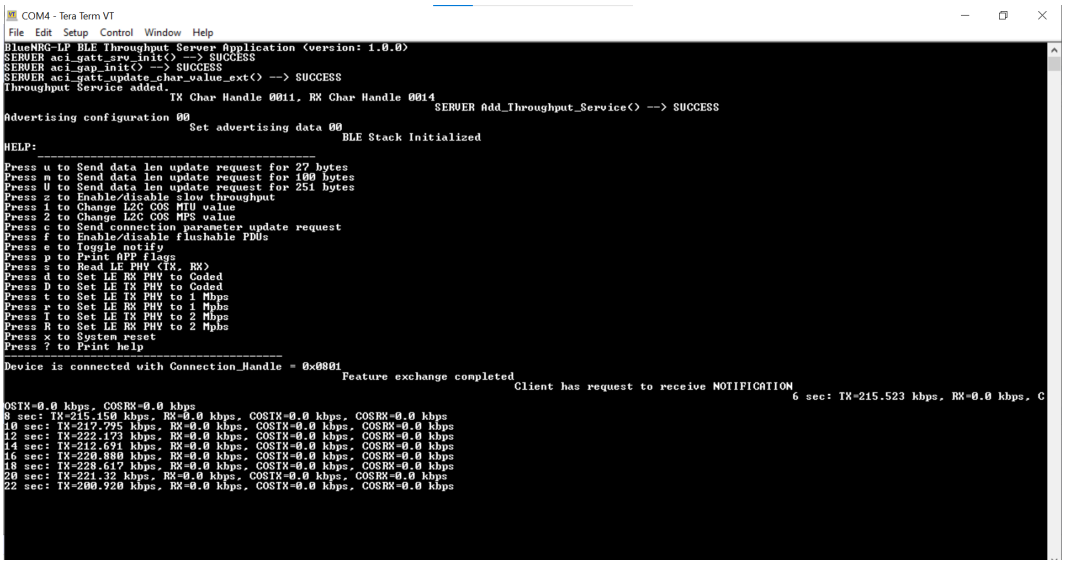


Figure 15. BLE Throughput Tera Term output - 2



3.4 BLE OTA reset manager

- The application implements the OTA Reset Manager which, at reset, passes control to the latest valid Bluetooth LE application updated through the Bluetooth LE Over-The-Air (OTA) Service
- The OTA Reset Manager is a basic application which is stored at BlueNRG-LP FLASH base address (0x10040000) and it allows to transfer of control towards the new upgraded application every time we reset
- The new application must add the OTA service and related characteristics defined on files OTA_btl.c
- At device reset, the reset manager will take care of jumping to the location of the last image that was successfully loaded by the OTA bootloader

Note:

- *Before downloading the OTA Reset Manager performs a device Mass Erase of the selected BlueNRG-LP device (use IAR, Project, Download, Erase Memory). Then, open the IAR project related to a Lower Application with OTA Service and download it on the selected device. At this stage, the BlueNRG-LP device is ready for performing OTA upgrades.*
- *Refer BLE_SensorDemo projects for related OTA update examples (Lower and Higher Applications with OTA service configurations).*

Running the Application:

Step 1. Take One SensorTile.box PRO hardware and one BlueNRG-LP EVK

Step 2. Connect the STLINK V3 programmer probe to JP3 header on SensorTile.box PRO board. Connect the programmer to PC with a USB Cable

Step 3. Open the BLE_OTA_ResetManager.eww (STSW-MKBOX- BLEDK\Projects\BLE_Examples \BLE_OTA_ResetManager) IAR project, and select project, download, erase memory to erase the device Flash

Step 4. Build and download the related OTA reset manager application (STSW-MKBOX-BLEDK\Projects\BLE_Examples \BLE_OTA_ResetManager) on the selected platform (pre-built image is available on STSW-MKBOX-BLEDK\Firmware\BLE_Examples\BLE_OTA_ResetManager folder and can be loaded through RF Flasher utility)

Note: While flashing BLE_OTA_ResetManager binary(.bin) using RF Flasher Utility enter the image "Flash from Address" as "0x10040000"

Step 5. Open the BLE_SensorDemo.eww IAR project, LowerApp_OTA workspace (STSW-MKBOX-BLEDK\Projects \BLE_Examples\BLE_SensorDemo) and build and download the related application image on the selected platform (pre-built image is available on STSW-MKBOX- BLEDK\Firmware \BLE_Examples \BLE_SensorDemo\BlueNRG-LP\LowerApp_OTA folder and can be loaded through RF Flasher Utility)

Step 6. Remove the STLink Debugger from PC, Power reset the Board using S1 swit

Step 7. Program the BlueNRG-LP kit platform with the required DTM application to be used with the BlueNRG GUI (pre-built DTM image DTM_UART_WITH_UPDATER.hex is available on STSW-MKBOX-BLEDK\Firmware\BLE_Examples \DTM_BlueNRG-LP_EVK folder and can be loaded through RF Flasher Utility GUI)

Step 8. Open the BlueNRG GUI on the PC and select the COM port related to the BlueNRG-LP EVK platform (STEVAL-IDB011V1) configured in step 6, through the dropdown "Port" and press 'Open'

Step 9. Select 'Tools' - 'OTA bootloader' to open up the dialog containing OTA FW upgrade actions and press 'Search for devices'

Step 10. After 'Search for devices', the GUI starts the discovery process and returns with some information about the address and application names of the devices running OTA FW upgrade service within the radio range. Once the previous process ends, the device list can be opened through the combo box arrow below the 'Search for devices' button and the user can choose the device they intend to connect for the firmware upgrade process using the "Connect" button (application address and names are displayed). If the user realizes he has connected the wrong device, he can just press the 'Force Disconnection' button and get back to the device selection within the combo box

Step 11. After the device selection, the connection through ‘Connect’ button and reading of the related free memory range, the user is requested to provide the new image file compiled with a base address and size fitting within the expected range on the slave device. For that purpose, the user can open the BLE SensorDemo IAR workspace HigherApp_OTA, and build the related BLE_SensorDemo_HigherApp_OTA.bin image (pre-built image is available on STSW-MKBOX-BLEDK\Firmware\BLE_Examples\BLE_SensorDemo\BlueNRG-LP\ HigherApp_OTA folder) and contain a version of the sensor demo that is compiled with a base address equal to the highest address

Step 12. Once user browses to the selected application image built with the proper base address, they can select the ‘Update’ button to start the OTA firmware upgrade process: a progress bar provides awareness for the time needed until the process ends

Step 13. Click on “Force Disconnection”

Step 14. On process completion the new application is launched

Figure 16. BLE OTA Reset Manager Firmware upgrade using BlueNRG-GUI - 1

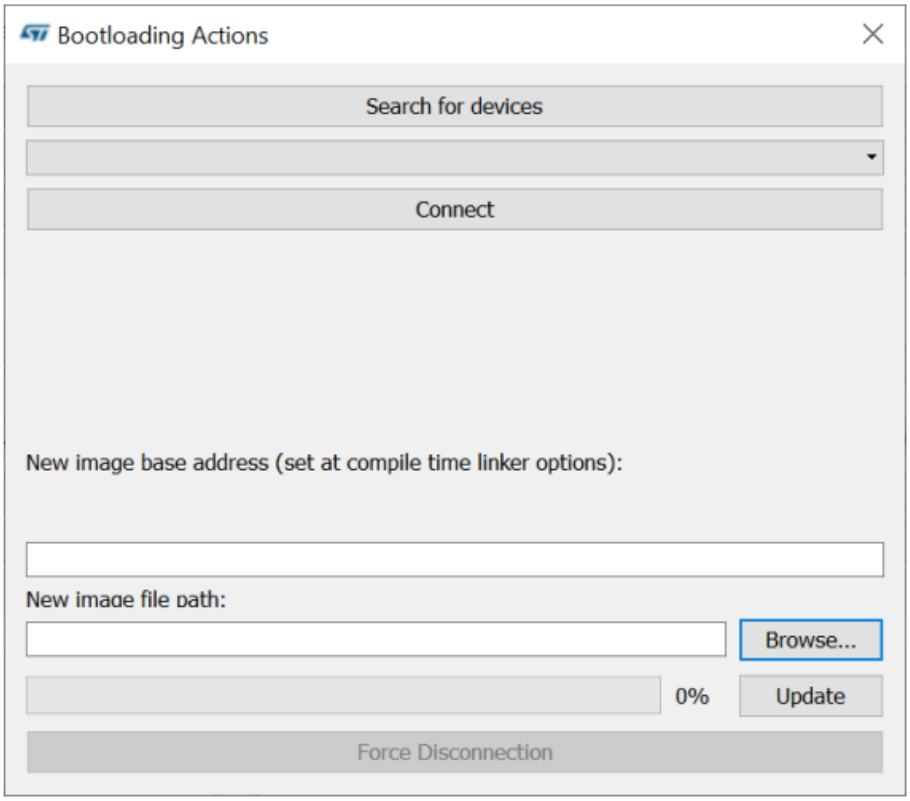


Figure 17. **BLE OTA Reset Manager Firmware upgrade using BlueNRG-GUI - 2**

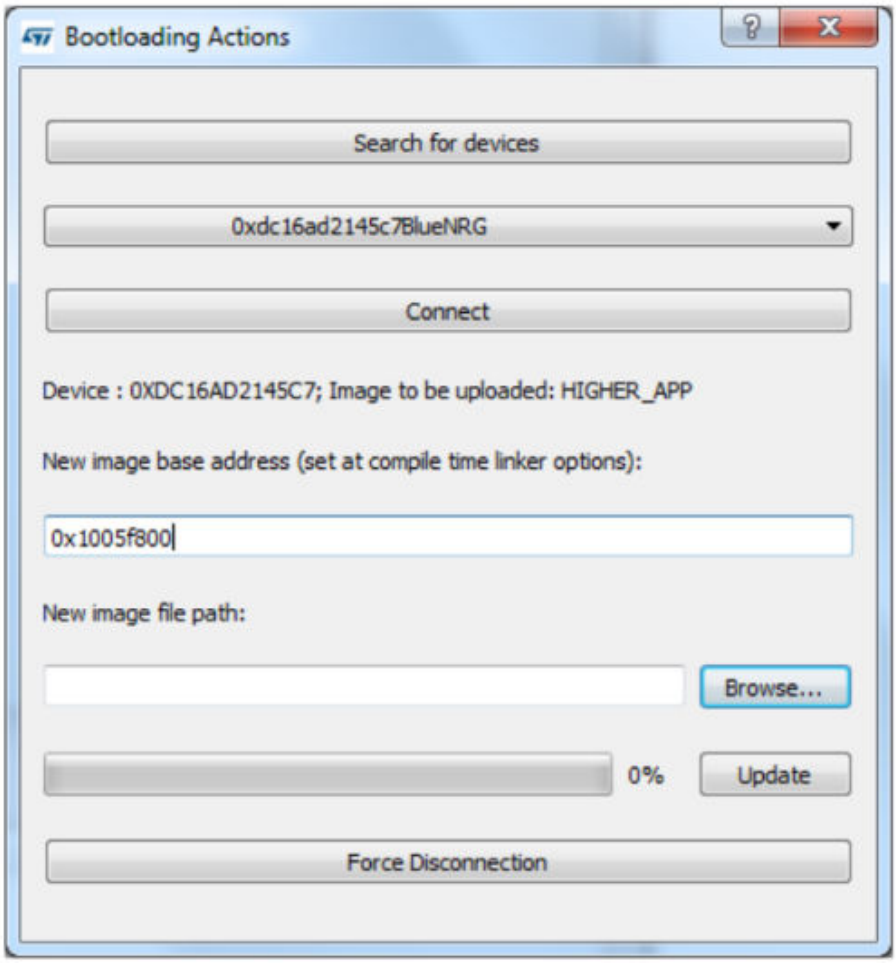
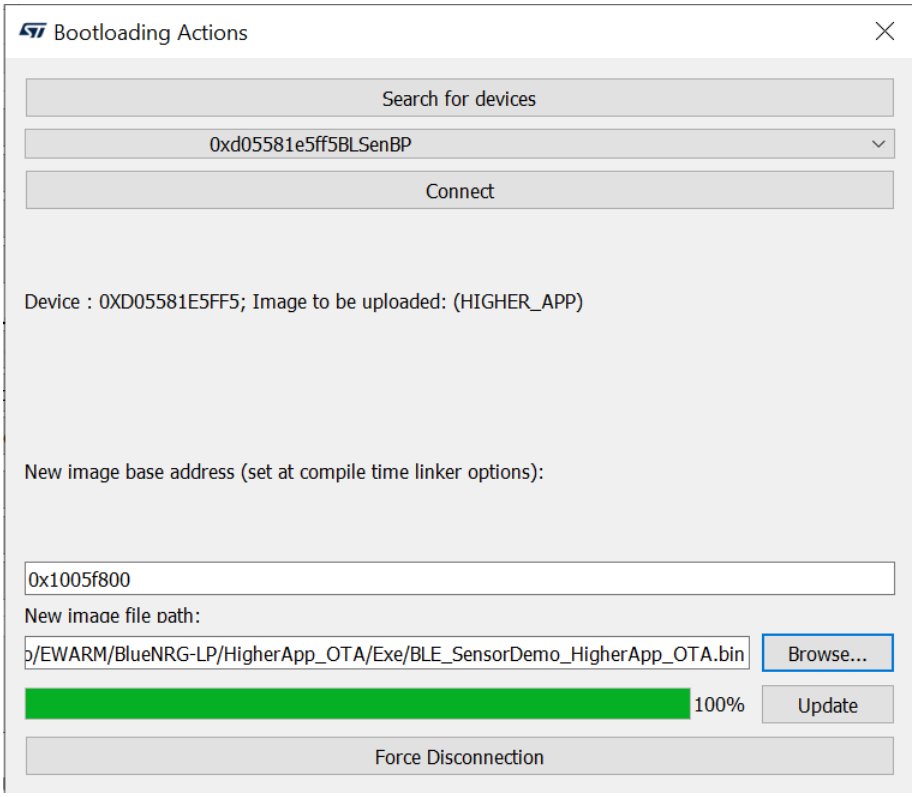


Figure 18. **BLE OTA Reset Manager Firmware upgrade using BlueNRG-GUI - 3**



3.5 BLE OTA service manager

- The application implements a basic standalone Bluetooth LE Over The Air(OTA) firmware upgrade. It provides the Bluetooth LE Over-The-Air Service management for handling OTA firmware upgrade of a Bluetooth LE application which doesn't have any Bluetooth LE OTA service
- The OTA Service Manager is a basic application which only supports the OTA Bootloader service
- It provides the Bluetooth LE OTA bootloader service to any Bluetooth LE application stored at fixed base address on user Flash which doesn't include any OTA service. It also includes the OTA Reset Manager functionalities in order to transfer the control to the proper valid application, after a Bluetooth LE OTA session
- User is only requested to load the OTA_ServiceManager application and then build any application using it with the CONFIG_OTA_SERVICE_MANAGER as preprocessor option
- Further, for jumping to the OTA Service Manager application, the application can call the OTA_Jump_To_Service_Manager_Application function (i.e., just using a platform button to activate such call)

Note: Refer to Use_OTA_ServiceManager workspaces on BLE_Beacon and BLE_SensorDemo IAR projects for all related examples.

Running the Application:

Step 1. Take One SensorTile.box PRO hardware.

Step 2. Connect the STLINK V3 programmer probe to JP3 header on SensorTile.box PRO board. Connect the programmer to PC with a USB Cable.

Step 3. Open the BLE_OTA_ServiceManager.eww (STSW-MKBOX- BLEDK\Projects\BLE_Examples \BLE_OTA_ServiceManager) IAR project, select project, download, erase memory to erase the device Flash.

Step 4. Build and download the related OTA service manager application (STSW-MKBOX-BLEDK\Projects\BLE_Examples \BLE_OTA_ServiceManager) on the selected platform (pre-built image is available on STSW-MKBOX- BLEDK\Firmware \BLE_Examples\BLE_OTA_ServiceManager folder and can be loaded through RF Flasher utility).

Note: While flashing BLE_OTA_ServiceManager binary(.bin) using RF Flasher Utility enter the image "Flash from Address" as "0x10040000".

Step 5. Remove the Debugger from PC, Power reset the Board using S1 switch.

Step 6. Open the ST BLE Sensor Classic Mobile app and click 'Select one Device'.

Step 7. Save the firmware binary(.bin) to be flashed in Smart Phone at a known location. For example, pre-built image is available on STSW-MKBOX- BLEDK\Firmware\BLE_Examples\BLE_SensorDemo\BlueNRG-LP\Use_OTA_ServiceManager

Step 8. Search for device in the App, the SensorTile.box PRO board will be visible as 'OTAServiceMgr'., from the Options select "Firmware Upgrade".

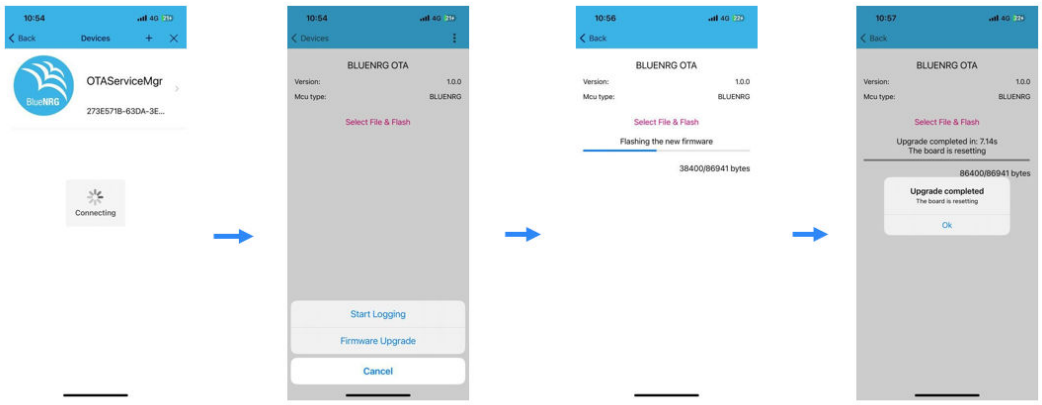
Step 9. Browse the binary file (.bin file saved in **Step 7**) to flash in BlueNRG-LP.

Step 10. Power reset the Board, it will now Run the Updated Firmware.

Step 11. SensorTile.box PRO board will be visible as 'BLSenBP'. Click on it to connect. Sensor's data will be visible on mobile applications.

Step 12. To update the firmware again, button PUSH1 on SensorTile.box PRO board, this will force it to run BLE_OTA_ServiceManager Application.

Figure 19. BLE OTA Service Manager Firmware upgrade using ST BLE Sensor Classic Mobile App



3.6 BLE serial port master slave

- This is a Serial Port demo that shows how to implement a simple 2-way communication between only two SensorTile.box PRO Kits(single application for both devices).
- The application will listen for keys typed in one node and, on return press, it will send them to the remote node. The remote node will listen for RF messages, and it will output them in the serial port. In other words everything typed in one node will be visible to the other node and vice versa.

- The Serial Port Master and Slave application configures a BlueNRG-LP device as Central & Peripheral using the aci_gap_init (GAP_CENTRAL_ROLE|GAP_PERIPHERAL_ROLE, ...). Then it performs the following steps:
  - It starts discovery procedure for a another BlueNRG-LP device configured with the same binary image.
  - If such device is found within a random time, it starts a connection procedure and wait until a connection is established.
  - If discovery procedure time expires without finding another Serial Port Master and Slave device, the device enters in discovery mode, waiting for another Serial Port Master and Slave device to discover and connect to it.
  - If discovery mode time expires without any connection established, the device starts again a discovery procedure for another Serial Port Master and Slave device.
  - When a connection is established, the device role are defined (Client or Server) and the Serial Port communication channel can be used.
- The Serial Port Service contains 2 Characteristics:
  - The TX Characteristic: the Client device can enable notifications on this characteristic. When the Server device has data to be sent, it will send notifications which will contains the value of the TX Characteristic
  - The RX Characteristic: it is a writable characteristic. When the Client device has data to be sent to the Server device, it will write a value into this characteristic.
- The maximum length of the characteristic value is 20 bytes.

Running the Application:

Step 1. Take two pairs each of SensorTile.box PRO and STLINK V3 hardware. Connect the STLINK V3 programmer probe to JP3 header on both SensorTile.box PRO boards. Connect the programmers to PC with USB Cables.

Step 2. Program both SensorTile.box PRO boards (STSW-MKBOX BLEDK\Projects\BLE_Examples\ BLE_SerialPort_Master_Slave) on the selected platform (pre-built image is available on STSW-MKBOX- BLEDK\Firmware\ BLE_Examples\ BLE_SerialPort_Master_Slave folder and can be loaded through RF Flasher Utility) .

Step 3. Remove the Debugger from PC, Power reset the Board using S1 switch.

Step 4. Both platforms are seen on the PC as a virtual COM port. Open the port in a serial terminal tool. The required serial port baud rate is 115200.

Step 5. The application will listen for keys typed in one node and, on return press, it will send them to the remote node. The remote node will listen for RF messages, and it will output them in the serial port. In other words, everything typed in one node will be visible to the other node and vice versa.

Figure 20. BLE SerialPort Master Slave Tera Term Output

Revision history

Table 2. **Document revision history**

Date	Revision	Changes
01-Sep-2023	1	Initial release.

Contents

List of tables

List of figures