The BlueNRG GUI SW package

Introduction

This document describes the BlueNRG GUI SW package (STSW-BNRGUI) with the BlueNRG graphical user interface (GUI) and the standalone script launcher utility.

The BlueNRG GUI is a PC application that can be used to interact and evaluate the capabilities of the BlueNRG-MS, BlueNRG-2N Bluetooth low energy network processors: low power Bluetooth® Low Energy ICs, compliant with the Bluetooth® specifications, with support both for master and slave roles. It also supports the BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS Bluetooth Low Energy systems-on-chip: low power Bluetooth® Low Energy ICs compliant with the Bluetooth® specifications, both in master and slave roles.

The BlueNRG GUI allows standard and vendor-specific HCI commands to be sent to the device controller and events to be received from it. It also provides a script engine, which can be used to load and run user scripts based on Bluetooth Low Energy stack APIs and related stack events. These scripts can also be run via standalone script launcher utility through a PC DOS window, outside the BlueNRG GUI context. A set of sample scripts is provided in the SW package.

1 Getting started

This section describes all system requirements to run the BlueNRG GUI or script launcher utility, as well as the relative SW package installation procedure.

1.1 System requirements

The BlueNRG GUI PC application has the following minimum requirements:

- PC with Intel® or AMD® processor running one of the following Microsoft® operating systems:
  - Windows 10
- At least 2 GB of RAM
- USB ports
- Adobe Acrobat Reader 6.0 or later(1)

1. *Recommended display scale and settings are up to 125 %*

1.2 BlueNRG GUI SW package setup

1. Extract the content of the en.STSW-BNRGUI.zip file into a temporary directory.
2. Extract and launch the BlueNRG_GUI_-x.x.x.Setup.exe file and follow the on-screen instructions.

1.3 BlueNRG GUI SW package structure

The BlueNRG GUI SW package files are organized into the following folders:

- **Application**: contains BlueNRG_GUI.exe and BlueNRG_Script_Launcher.exe PC applications and some sample scripts.
- **Docs**: contain the BlueNRG GUI SW package release notes and html documentation covering ACI, events and sample scripts.
- **Firmware**:
  - BlueNRG-LP, BlueNRG-LPS DTM and USB-CMSISDAP files to be used for the BlueNRG-LP, BlueNRG-LPS development kits
  - BlueNRG-1,2 DTM and USB to serial binary files to be used for the BlueNRG-1, BlueNRG-2 and development kits
  - BlueNRG_VCOM_x.x.hex prebuilt binary file to be used for the STM32L1 microcontroller on BlueNRG-MS kit motherboards
  - BlueNRG-MS FW stack versions
- **PCDriver**: contains the PC DFU and USB Virtual COM (Windows 7) drivers.

Note: The provided BlueNRG-LP, BlueNRG-LPS DTM binary files can be rebuilt using DTM project available in the STSW-BNRGLP-DK SW package (Table 1).

Note: The provided BlueNRG_VCOM_x.x and BlueNRG-1,2 DTM binary files can be rebuilt using the Virtual_COM_Port and DTM projects, respectively, available in the STSW-BLUENRG-DK and STSW-BLUENRG1-DK SW packages (Table 1). Each project provides a complete set of source and header files.

Note: The BlueNRG-2N network coprocessor device is preprogrammed at manufacturing time with the proper authenticated DTM binary image. The latest authenticated binary image version is available on the BlueNRG-2N web site.

2 GUI software description

The BlueNRG GUI included in the software package is a graphical user interface that can be used to interact and evaluate the capabilities of the BlueNRG-MS and BlueNRG-2N network processors, and the BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS systems-on-chip.

This utility can send standard and vendor-specific HCI commands to the controller and receive events from it. It allows the user to configure each field of the HCI command packets to be sent and analyzes all received packets to allow easy low level management of the BlueNRG-MS, BlueNRG-2N, BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS devices.

2.1 Requirements

To use the BlueNRG GUI, ensure your hardware and software are set up (BlueNRG GUI installed) correctly.

Note: When connecting a device platform, before opening the related COM port, please make sure that the expected GUI serial baud rate is set as follows:

- *BlueNRG-LP, BlueNRG-LPS: 921600 (default configuration)*
- *All other devices: 115200*

Refer to Section 2.2.2 Tools for information about how to set the GUI serial baud rate.

2.1.1 BlueNRG-MS network coprocessors

The STM32L1 in the STEVAL-IDB005V1 and STEVAL-IDB005V1D kits has been preprogrammed with a demo application (BlueNRG sensor demo). Hence, new firmware must be loaded into the STM32L1 microcontroller in order to use the BlueNRG GUI. The firmware image that must be programmed is the latest BlueNRG_VCOM_x_x.hex available in the BlueNRG GUI SW package in Firmware/STM32L1_prebuilt_images folder.

To download this binary image into the internal Flash of the STM32L1, the microcontroller must be set in the special DFU (device firmware upgrade) mode.

To enter DFU mode:

- BlueNRG-MS development platforms (order codes: STEVAL-IDB005V1, STEVAL-IDB005V1D):
  1. Power up the board
  2. Press and hold push button
  3. Reset the board using the RESET button (keep the push button pressed while resetting). The orange LED DL2 starts to blink
  4. Release the push button
  5. Use the BlueNRG GUI to flash the device with the BlueNRG_VCOM_x_x.hex binary file (Tools → Flash motherboard FW)
- BlueNRG-MS USB dongle (order codes: STEVAL-IDB006V1):
  1. press and hold the SW1 button
  2. plug the USB dongle on a PC USB port. The orange LED D3 starts to blink
  3. use BlueNRG GUI to flash the device with BlueNRG_VCOM_x_x.hex binary file (Tools → Flash Motherboard FW)

2.1.2 BlueNRG-1 and BlueNRG-2 network coprocessors

The STEVAL-IDB007Vx (x=1,2) (BlueNRG-1) and STEVAL-IDB008Vx (x=1,2) /STEVAL-IDB009Vx (x=1) (BlueNRG-2) kits are preprogrammed with a demo application (Bluetooth LE sensor demo), so new firmware must be loaded into the BlueNRG-1 and BlueNRG-2 devices to configure them as network coprocessors and use the BlueNRG GUI.

The firmware images, that must be programmed in order to configure the devices on the STEVAL-IDB007Vx (x=1,2) (BlueNRG-1) and STEVAL-IDB008Vx (x=1,2) /STEVAL-IDB009Vx (x=1) (BlueNRG-2) boards as network coprocessors, are available in the BlueNRG GUI SW package in the *Firmware/BlueNRG1* and *Firmware/BlueNRG-2* folders, respectively.

Two network coprocessor configurations are available:

1. UART (DTM binary file: DTM_UART.hex)
2. SPI (DTM binary file: DTM_SPI.hex)

To download the selected binary image into the STEVAL-IDB007Vx (x = 1,2) (BlueNRG-1) or STEVAL-IDB008Vx (x=1,2) /STEVAL-IDB009Vx (x=1) (BlueNRG-2) internal Flash:

1. Connect the device platform to a PC USB port
2. On PC computer window, use the the "drag and drop" mass storage upgrade capability for copying the selected binary image into the associated IDB007/8/9VX platform

Another way to download the selected binary image is the following:

1. Connect the device platform to a PC USB port
2. Open the RF-Flasher utility available on STSW-BNRGFLASHER SW package (Table 1)
3. Select the COM port associated with the device platform to be configured as a network coprocessor
4. Select the specific DTM binary file through the select file button
5. Select the mass erase option
6. Press the Flash button to program the device

Notes:

1. The BlueNRG GUI --> Tools --> Stack updater… utility is intended only to update a BlueNRG-1, BlueNRG-2 or a BlueNRG-2N device already configured as a network coprocessor as it assumes that specific updater/ bootloader code is already loaded on the device.
2. The DTM_UART.hex and DTM_SPI.hex binary image files include the updater code, which allows the update of the specific DTM FW version through the BlueNRG GUI --> Tools --> Stack updater… utility. DTM binary images without updater code are also provided in the SW package: DTM_UART_NOUPDATER.bin for UART configuration and DTM_SPI_NOUPDATER.bin for SPI configuration.
3. Once the STEVAL-IDB007Vx (x = 1,2) (BlueNRG-1) or STEVAL-IDB008Vx (x = 1,2) /STEVAL-IDB009Vx (x=1) (BlueNRG-2) platform has been configured as a network coprocessor, new DTM binary file versions can be updated directly using the BlueNRG GUI, Tools, Stack Updater … utility thus:
   a. Connect the device platform (STEVAL board) to a PC USB port
   b. Open the BlueNRG GUI tool
   c. Open COM port associated with the device platform
   d. Select BlueNRG, Tools, BlueNRG updater …
   e. Select the DTM binary images without updater available in Firmware/BlueNRG1 or Firmware/ BlueNRG-2 folders (DTM_UART_NOUPDATER.bin or DTM_SPI_NOUPDATER.bin)
   f. Select update to program the new DTM binary image
4. No other BlueNRG-1 or BlueNRG-2 application binary image can be programmed on a corresponding device using the BlueNRG GUI, Tools, Stack Updater utility.
5. In order to use the STEVAL-IDB007V1 for the BlueNRG-1 SPI network coprocessor configuration, please make sure the following hardware changes have been made on such platform: make a short at positions R59, R60, R61, R62 on STEVAL-IDB007V1 platform.
6. Document content is also valid for the BlueNRG-1, STEVAL-IDB007V1M evaluation platform based on the SPBTLE-1S module with 32 MHz HS crystal and for the BlueNRG-2, STEVAL-IDB008V1M kit (evaluation platform based on the BlueNRG-M2SA module with 32 MHz HS crystal). The associated DTM_UART.hex and DTM_SPI.hex binary images are provided within the folder Firmware/STEVAL-IDB007V1M.
7. The BlueNRG-2N network coprocessor is alredy loaded with a dedicated DTM binary image running only on this specific device. The latest available BlueNRG-2N binary image version is available on the BlueNRG-2N web site with associated documentation and updater instructions.

2.1.3 The BlueNRG-LP, BlueNRG-LPS network coprocessors

A specific firmware must be loaded into the BlueNRG-LP, BlueNRG-LPS device to configure the network coprocessors and use the BlueNRG GUI.

The firmware image, which must be programmed in order to configure the BlueNRG-LP device on the STEVAL-IDB011V1/V2 boards as network coprocessor, are available in the BlueNRG GUI SW package in the *Firmware/ STEVAL-IDB011V1* folder. The firmware image, which must be programmed in order to configure the BlueNRG-LPS device on the STEVAL-IDB012V1 boards as network coprocessor, are available in the BlueNRG GUI SW package in the Firmware/STEVAL-IDB012V1 folder.

The following network coprocessor configurations are available:

1. UART (DTM binary file: DTM_UART_WITH_UPDATER.hex)
2. SPI (DTM binary file: DTM_SPI_WITH_UPDATER,hex)

To download the selected binary image into the STEVAL-IDB01xV1 internal Flash:

1. Connect the device platform to a PC USB port
2. On PC computer window, use the "drag and drop" mass storage upgrade capability for copying the selected binary image into the associated device platform

Once the STEVAL-IDB011V1 platform has been configured as a network coprocessor, new DTM binary file version can be updated directly using the BlueNRG GUI, tools, BlueNRG updater … utility:

1. Connect the device platform (STEVAL board) to a PC USB port
2. Open the BlueNRG GUI tool
3. Open COM port associated with the device platform
4. Select BlueNRG -> Tools -> BlueNRG Updater …
5. Select the DTM binary image (*.bin) to be used for updater available in Firmware/STEVAL-IDB011V1 folder (DTM_UART_FOR_UPDATER.bin or DTM_SPI_FOR_UPDATER.bin)
6. Select "update" to program the new DTM binary image

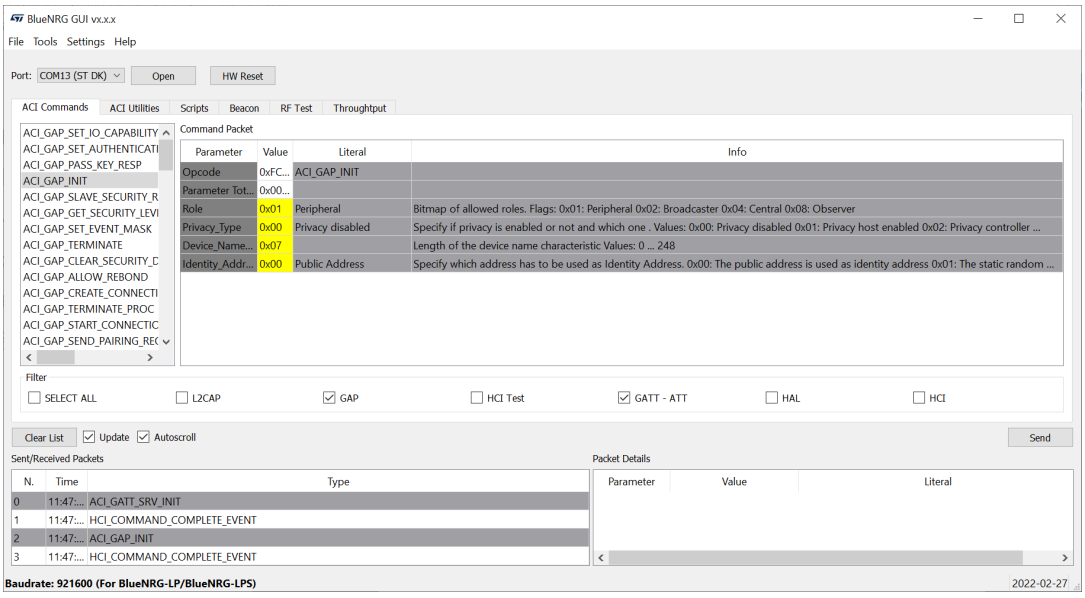
Note: No other application binary image can be programmed on a corresponding device using the BlueNRG Updater utility.

2.2 The BlueNRG graphical user interface

This section describes the main functions of the BlueNRG GUI application. You can run this utility by clicking on the BlueNRG GUI icon on the desktop or under: start menu folder ST BlueNRG GUI x.x.x → BlueNRG GUI.

2.2.1 GUI main window

Figure 1. BlueNRG GUI main window



The BlueNRG GUI main window has different zones, some of which can be resized.

Port and interface selection

The uppermost zone allows the user to open the COM port associated to the BLE controller.

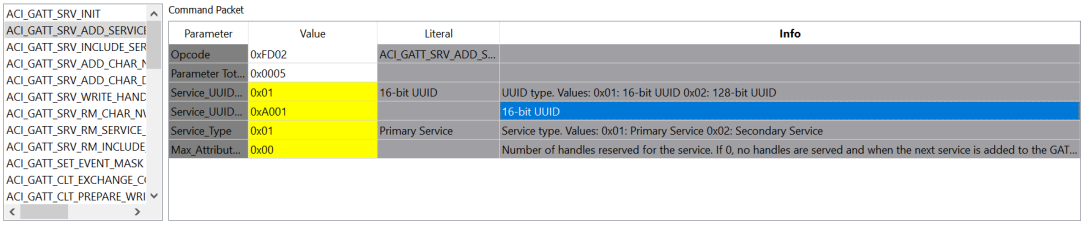
When a COM port is opened the following information is displayed:

- BlueNRG-MS, BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP and BlueNRG-LPS HW version
- BlueNRG-MS FW stack version, or BlueNRG-1 or BlueNRG-2 or BlueNRG-2N, or BlueNRG-LP or BlueNRG-LPS FW stack and DTM FW versions
- Motherboard GUI firmware: BlueNRG_VCOM_x_x for BlueNRG-MS kits, USB to serial for BlueNRG-1, BlueNRG-2 kits and USB CMSISDAP for BlueNRG-LP, BlueNRG-LPS kit version.

ACI commands

The ACI commands tab contains a list of all the available commands. Commands can be filtered by checking/ unchecking boxes under the filter section. After clicking on one of the commands, all packet fields are displayed on the command packet table in the upper-right section of the tab.

Figure 2. Command packet table



The command packet table contains four columns:

- **Parameter**: name of the packet field as per volume 2, part E of Bluetooth specification
- **Value**: field value represented in hexadecimal format (right-click on a cell to change the format)
- **Literal**: meaning of the current field value
- **Info**: description of the corresponding field

Only the yellow cells of this table can be modified by the user. *Parameter Total Length* is fixed or automatically calculated after modifying cell content.

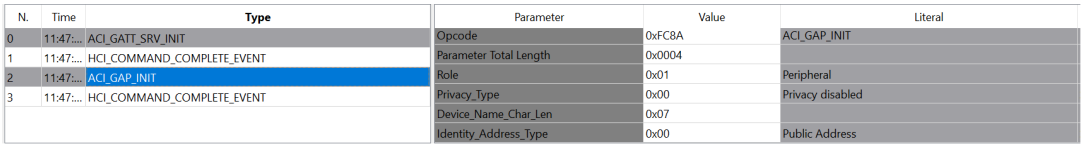
After fields have been modified, the command can be sent using the send button.

Packet history and details

Two frames at the bottom of the main window show packets sent to and received from the Bluetooth LE controller, as well as other events:

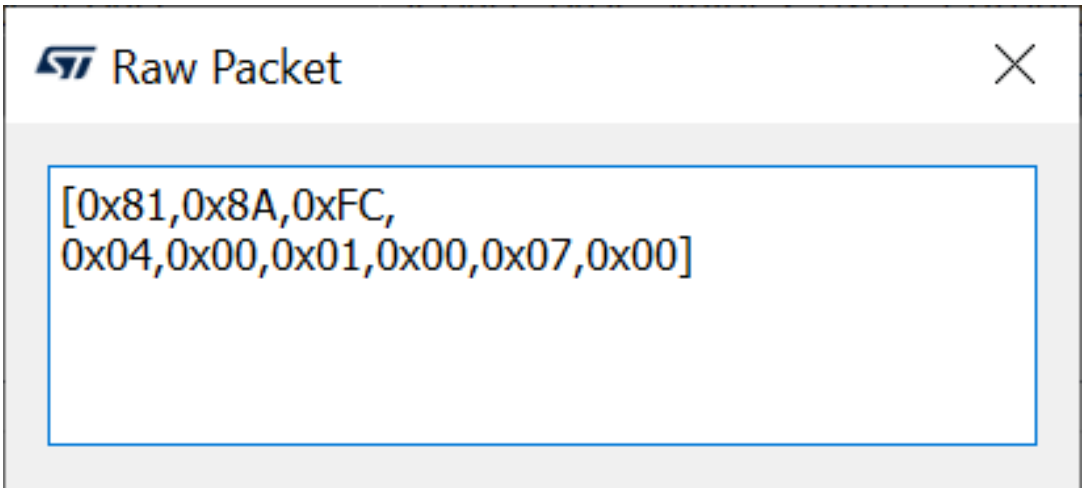
- The frame on the left (sent/received packets) holds a history of all packets
- The frame on the right (packet details) shows all the details of the selected packet as per the command packet table

Figure 3. Packet history and details



Double-clicking on a row of the sent/received packet table shows the raw packet.

Figure 4. Raw packet dump



Some events (displayed in yellow cells) can provide other information. HCI packets sent towards the Bluetooth LE controller are displayed in gray cells while received packets are shown inside white cells.

The sent/received packets table can be cleared by clicking on clear list button. Update and auto-scrolling check boxes enable or disable updating and auto-scrolling of the sent/received packets table while new packets are sent or received (however, information is still printed).

The sent/ received packets can be stored and later reloaded on the GUI, by using the utilities provided on file menu:

- **Save History and Save History as Text**: saves the current list of sent commands and received events to a CSV file or to a simple text file
- **Load History**: loads a list of sent commands and received events from a CSV file
- **Save as Python Script**: stores the current list of sent commands and received events as a script file (python format), which can be (by adding specific code for handling events, parameters) and then used in the GUI script window as part of a user application scenario (refer to Section 2.2.4 GUI script window)

- **Save As C Code...**: stores the current list of APIs and events on sent/received packet table as header and source files (C language), which can be used as starting point for user C code development. The generated header and source files are provided within a complete toolchains framework (IAR, KEIL Atollic and WISE-Studio), which allows user to get a basic reference code to be customized and integrated for addressing the specific user application scenario (STM32L4, BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS network coprocessor framework is also generated). User is requested to:
  1. On the BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS devices, add the configuration header file with proper values required for the correct BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS Bluetooth LE radio initialization, based on specific user application scenario:
     a. For Bluetooth LE stack v2.0, v2.1x, v3.x, this file can be generated using the BlueNRG-X Radio Initialization Parameters Wizard available on the BlueNRG-1_2 SW package (STSW-BLUENRG1-DK) or BlueNRG-LP, BlueNRG-LPS SW package (STSW-BNRGLP-DK).
     b. Save As C Code ... window also allows user to directly launch the BlueNRG-X Radio Initialization Parameters Wizard in order to define the radio initialization parameters and store the associated user configuration file on the generated inc folder.
  2. On the BlueNRG-1, BlueNRG-2 devices, aci_gap_init() API, make sure that the device_name_char_len parameter is configured exactly with the actual device name characteristic length
  3. On the BlueNRG-1, BlueNRG-2 devices, make sure that each aci_gatt_add_service(), Max_Attribute_Records parameter is configured exactly with the required number of attributes for the user application scenario, in order to avoid API failure:
     a. For each defined service, the correct Max_Attribute_Records parameter value is reported on the BlueNRG-X Radio Initialization Parameters Wizard, output window #define MAX_NUMBER_ATTRIBUTE_SERVICEx (x = 1, .. number of added services): just use the associated define numeric value
  4. On the BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS devices, the generated IDE projects are built with the full stack modular configuration option (BLE_STACK_CONFIGURATION=BLE_STACK_FULL_CONFIGURATION on Bluetooth LE stack v2.1x and BLE_STACK_FULL_CONF on Bluetooth LE stack v3.x). User can customize it according to his specific application scenario.
  5. Customize APP_Tick() function with user state machine and associated states
  6. Customize the required event callback functions taking into account application states
  7. Remove possible duplications of event callback functions
  8. Customize APIs execution sequence based on application state machine, event callbacks and state updates (i.e.: start service or characteristic discovery, read, write, notification,.. once the device is connected: connection complete event callback has been called)
  9. Remove repeated calls to the same APIs, if not needed (or review code by adding a simple for/while code iteration)
  10. Customize API input parameters with proper references to application variables
  11. Customize interrupt service routines based on user application needs (refer to generated default user functions section)
  12. Customize, modify generated default user functions
  13. Replace the linker files with the latest available version on associated DK SW packages
  14. Enable debug messages by setting DEBUG 1 on generated source files

After the BLE_User_main.c, user.c, user.h has been generated, look for the comments with prefix `USER ACTION IS NEEDED`. These comments highlight some possible post-built code customization to be performed on user side, in order to address his specific application scenario.

2.2.2 Tools

The BlueNRG GUI has some functions that can be accessed through the tools menu. These tools are described in this section.

Stack updater

This tool can be used to update the firmware in the BlueNRG-MS devices by using their internal bootloader. The BlueNRG_VCOM_x_x.hex firmware must be present on the BlueNRG-MS motherboard STM32L1 microcontroller and the COM port must be open.

To use this tool:

1. Go to tools → Stack updater
2. Select the correct stack firmware for the selected device
3. Press update to start the update procedure

If the procedure completes with no errors, the new firmware has been loaded into the device internal Flash.

This tool can also be used to update the network coprocessor firmware inside the BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP and BlueNRG-LPS devices, provided that it has been already configured as a network coprocessor with a DTM FW image containing the update functionality (refer to Section 2.1.2 BlueNRG-1 and BlueNRG-2 network coprocessors and Section 2.1.3 The BlueNRG-LP, BlueNRG-LPS network coprocessors).

IFR//Device Configuration

To preserve the flexibility of the BlueNRG-MS device, the firmware uses a parameter configuration table in a sector of the Flash called information register (IFR). The BlueNRG GUI IFR/Device Configuration tool can read and modify this portion of the BlueNRG-MS Flash.

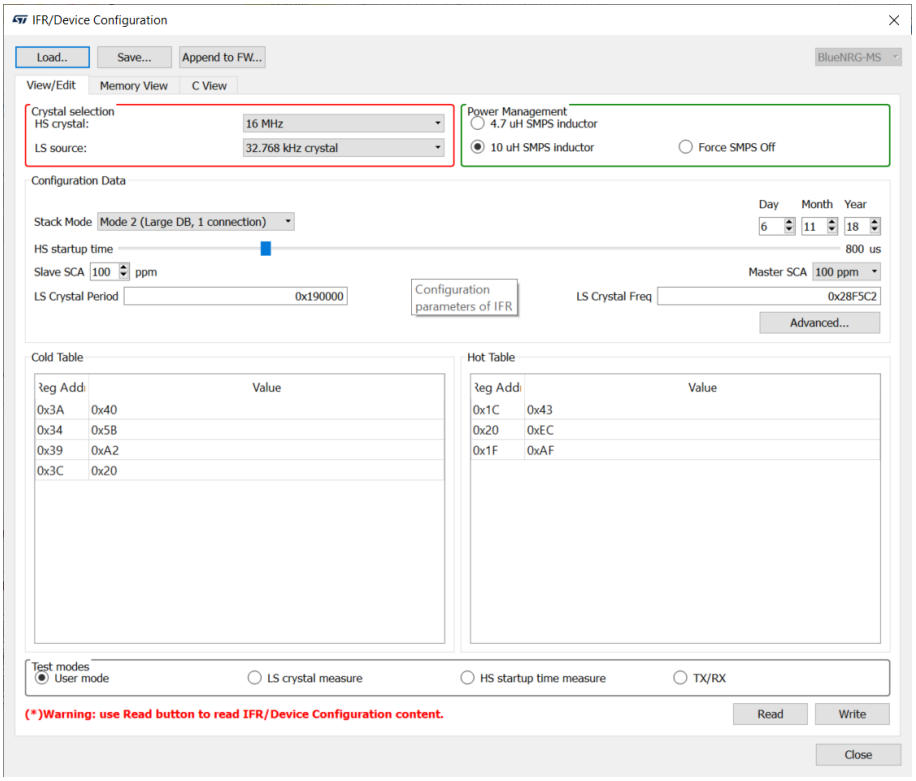
This tool is available in BlueNRG GUI → Tools → IFR/device configuration → item.

The BlueNRG GUI IFR/Device Configuration utility helps you define the IFR data and represents the only supported mode to define IFR data based on customer needs.

The utility provides the following windows:

- **View/Edit view**: displays the IFR regions with related fields and description. The user can modify some of these fields according to his needs.
- **Memory view**: displays the IFR field memory addresses and related values that are generated by BlueNRG GUI according to the specified values.
- **C view**: displays the C language structure related to the IFR configuration data region matching the View/Edit and Memory view.

Figure 5. BlueNRG GUI IFR/device configuration tool: view/edit view for the BlueNRG-MS device



In the view/edit view, the following operations are available:

- Select high speed (HS) crystal (16 or 32 MHz) and low speed oscillator source (32 kHz or the internal ring oscillator)
- Set power management options (SMPS inductor or SMPS off configuration)
- Change stack mode. Each mode has a different functionality:
  - Mode 1: slave/master, 1 connection only, small GATT database (RAM2 off during sleep)
  - Mode 2: slave/master, 1 connection only, large GATT database (RAM2 on during sleep)
  - Mode 3: slave/master, 8 connections, small GATT database (RAM2 on during sleep)
  - Mode 4: slave/master, simultaneous advertising and scanning, up to 4 connections, small GATT database (RAM2 on during sleep) (only on BlueNRG-MS FW stack versions above 7.1a)
- Change HS start-up time parameter. This parameter controls the time offset between the wakeup of the device and the start of RX/TX phase. It must be big enough to allow the device to be ready to transmit or receive after wakeup from sleep. This time depends on the start-up time of the high speed crystal
- Change sleep clock accuracy. This must reflect the actual clock accuracy, depending on the low speed oscillator or crystal in use
- Set low speed (LS) crystal period and frequency
- View/change date to distinguish between different versions of configurations
- View registers that are written into the radio (hot and cold table)
- Set some test modes for specific tests
- Read IFR content from the BlueNRG-MS device
- Write IFR configuration to the BlueNRG-MS device

The following general utilities are also available:

- Load button: loads a configuration file
- Save button: saves the current parameters into a configuration file

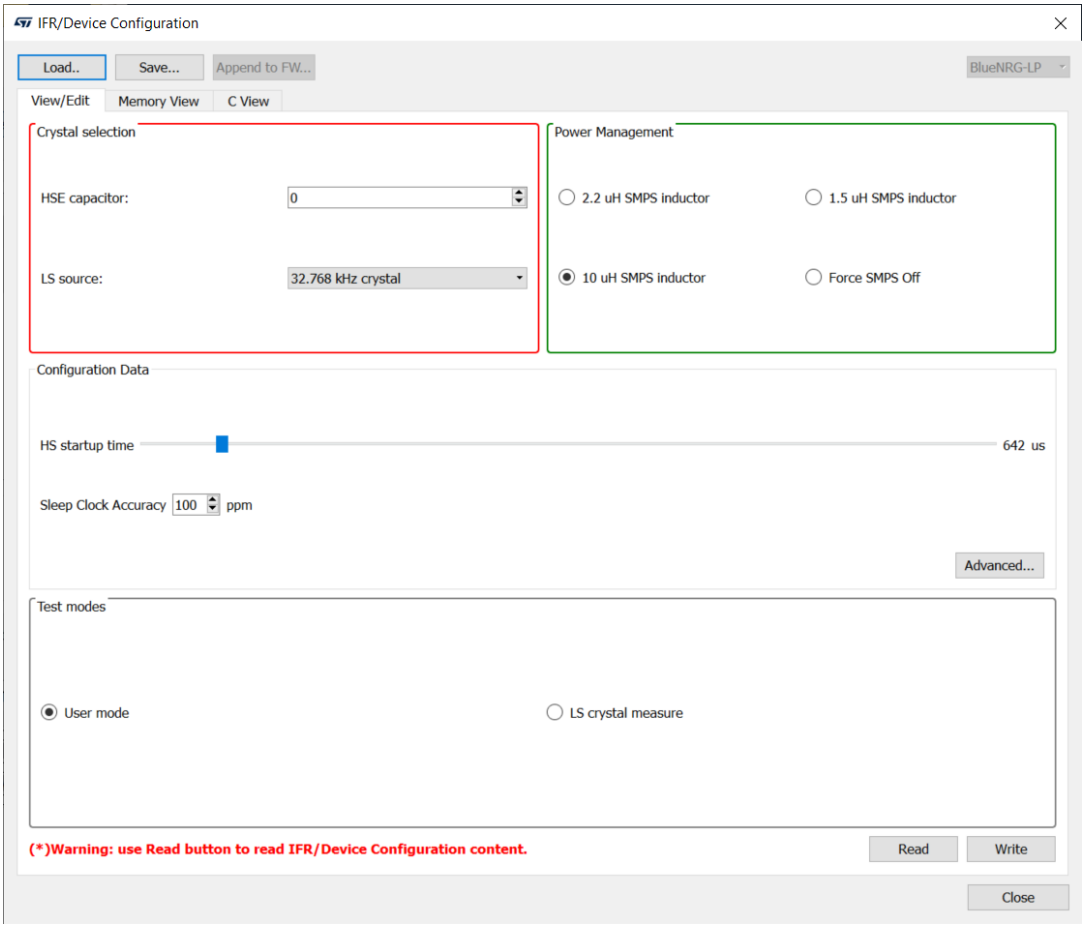
Device configuration for the BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP and BlueNRG-LPS

To preserve the flexibility, the BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP, BlueNRG-LPS device firmware uses a hardware configuration table, which can be read and modified by BlueNRG GUI IFR/device configuration tool; this tool is available in BlueNRG GUI, Tools, IFR/device configuration item.

The utility provides the following windows:

- **View/Edit view**: displays the device configuration parameter regions with related fields and description. The user can modify some of these fields as required
- **Memory view**: displays the device configuration field memory addresses and related values that are generated by BlueNRG GUI according to the specified values
- **C view**: displays the C language structure related to the device configuration data region matching the View/Edit and Memory view

Figure 6. Device configuration



In the View/Edit view, only the following operations are supported:

1. Select low speed oscillator source (32 kHz or the internal ring oscillator)
2. Set power management options (SMPS inductor value or SMPS off configuration)
3. Set some test modes for specific tests
4. Read device configuration parameter content from the BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP and BlueNRG-LPS
5. Write device configuration parameter configuration to the BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP and BlueNRG-LPS

Note: The BlueNRG-1 or BlueNRG-2 DTM UART HS crystal (16 or 32 MHz) is selected at compile time and cannot be changed through BlueNRG GUI IFR/device configuration (same for the BlueNRG-LP and BlueNRG-LPS devices: 32 MHz only).

The following general utilities are also available:

- Load button: allows loading a configuration file
- Save button: allows saving the current parameters into a configuration file

Flash motherboard firmware

The BlueNRG GUI embeds a utility that let you Flash firmware to the STM32L1 microcontroller on the BlueNRG-MS motherboard without a JTAG/SWD programmer. This utility uses a bootloader (DFU) located in the first 12 kB of the STM32L1 Flash.

Any application to be programmed to the STM32L1 by this tool must first consider that the lower area of the Flash is used by the bootloader (two precautions must be taken for any STM32L1 firmware: 1) change memory regions in linker script (vector table and Flash must start at 0x08003000); 2) change the vector table offset (NVIC_SetVectorTable()))

On the BlueNRG-1 and BlueNRG-2 development platform kits, this utility let you upgrade the USB-to-Serial firmware when needed.

To do so, activate the DFU application manually:

1. On the BlueNRG GUI PC application, select Tools → Flash Motherboard FW..
2. Select the BlueNRG-1 as HW type
3. Press the apply button of the window
4. Press and hold the RESET button
5. Plug the USB cable to the board
6. Release the RESET button
7. Red LED DL2 blinks to confirm the DFU application is running
8. On BlueNRG GUI PC application, select Tools → Flash Motherboard FW...
9. Select the HW type of the device to update the flash motherboard firmware
10. Press the apply button of the window
11. Select the firmware USB_to_SERIAL.hex available in the firmware/BlueNRG-1 or firmware/BlueNRG-2 folder and press the open button
12. Wait for the end of the upgrade operation

BlueNRG-LP, BlueNRG-LPS Flash USB-CMSISDAP firmware

The BlueNRG GUI embeds a utility that let user Flash USB-CMISDAP.hex firmware on BlueNRG-LP, BlueNRG-LPS STEVAL kit.

To do so:

1. On the BlueNRG GUI PC application, select Tools → Flash Motherboard FW...
2. Select the BlueNRG-LP as HW type
3. Press the apply button of the window
4. Plug the USB cable to the board
5. Select My Computer on the Windows Dialog
6. Select the ST IDB011V1 ("x") where "x" is the Windows driver location (D, E, F, ...)
7. Select the firmware USB_CMSISDAP_LP.hex available in the Firmware/STEVAL-IDB011V1 folder or the firmware USB_CMSISDAP_LPS.hex available in the Firmware/STEVAL-IDB012V1 folder
8. Select My Computer and choose the device in MAINTENANCE mode
9. The ST IDB011V1 ("x") device USB_CMSISDAP is upgraded and now it is available on My Computer window.

Note: If the device is still in MAINTENANCE mode, please unplug and plug again the USB cable.

OTA bootloader

The OTA bootloader let you Flash new firmware to the STM32L1 on a remote BlueNRG-MS motherboard via Bluetooth low energy technology. It also let you Flash new firmware to a remote BlueNRG-1 or BlueNRG-2 or BlueNRG-LP or BlueNRG-LPS device via Bluetooth Low Energy technology (refer to the dedicated device application notes available on www.st.com for further information).

Get production data

From the tools menu it is possible to retrieve production information stored at platform manufacturing time. This data is stored in the EEPROM available on the platform kits.

Get version

The Get version tool is used to retrieve the version of the BlueNRG GUI firmware (BlueNRG_VCOM_x_x.hex) on the STM32L1 controller, USB to serial FW version on the BlueNRG-1 and BlueNRG-2 platforms, CMSIS-DAP FW version on the BlueNRG-LP, BlueNRG-LPS platforms, and hardware and firmware versions from the BlueNRG-MS, BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS devices.

Settings

This tool let you configure the firmware stack version to be used from the GUI (when no device is actually connected to a PC USB port). It also let you configure the GUI serial baud rate (serial UART communication only).

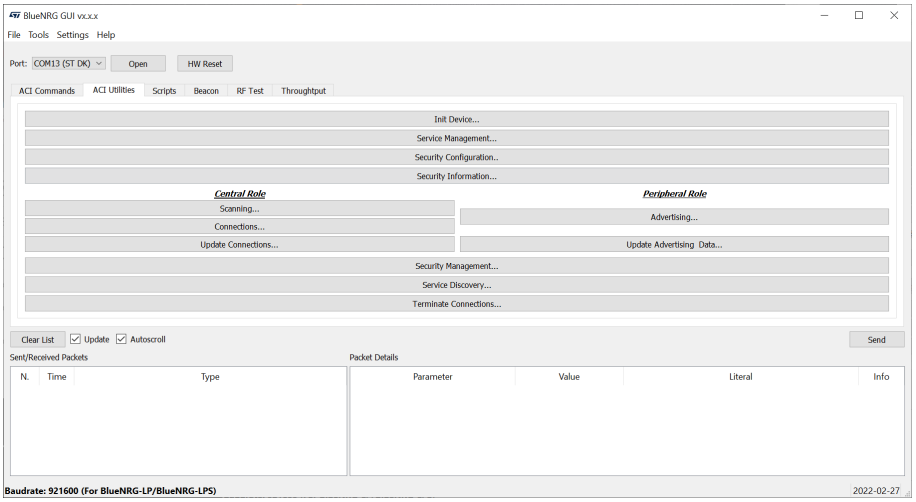
In order to use this function.

1. Go to settings → Fw Stack Version-> select FW 7.x or 7.1 for the BlueNRG-MS device, FW 1.0 or FW 2.x for the BlueNRG-1 device and 2.x for the BlueNRG-2 and FW 3.x for the BlueNRG-LP, BlueNRG-LPS devices.
2. Go to settings → select Set Baud Rate… and choose the value (default value is 921200 for the BlueNRG-LP and BlueNRG-LPS devices and 115200 for all other devices).

2.2.3 GUI ACI utility window

The BlueNRG GUI ACI utility window provides several tabs to allow testing of some application scenarios.

Figure 7. BlueNRG GUI ACI utility window



Central and peripheral roles are supported with the Bluetooth LE operations described in the following tables.

Table 1. GUI ACI utilities window: available general operations

Operation	Associated actions	Notes
Init Device…	Allows a device to be initialized by selecting: - Role - Stack mode (1,2,3,4) - Address type (public, random) and value - Tx power level - Power mode - Device name	
Security configuration	Allows the device security database to be erased Allows security settings to be configured: - IO capabilities - Authentication requirements	
Service management…	Allows adding a service by selecting: - UUID type (16 or 128 bits) - Service type (primary or secondary) - Set max. number of records For each service, it allows a characteristic and related descriptors to be added by selecting: - UUID type (16 or 128 bits) - Properties - Security permissions - Variable length or not - Length - GATT event mask - Encryption key size	After a characteristic is defined, the user can edit its parameters and/or delete it. Once a service and its characteristics, descriptors have been defined, click OK to add them to the GATT database. The defined GATT database is shown on a specific view. The "Save View" button allows the defined services and associated characteristics to be stored in order to be reloaded later with the "Load View" button
Security management ...	Allows security operations (send slave security request, send pairing request, allow rebond, ...) to be performed	
Security information	Allows the list of the bonded devices to be got	
Service discovery ..	Allows discovery of all services and related characteristics of available connections.	Service start handle, end handle and UUID are showed. For each selected service the related characteristics information are showed (attribute handle, property, value handle and UUID). For the available characteristic with notify or indication property, it is possible to enable the notification/indication. It is also possible to store as view the discovered services and associated characteristics for a later usage through the "Load View" button on "Service Management …" window. This allows user to build his own "Services/ Characteristics View" database from discovered peer device services and characteristics, and store them for possible future application scenarios.
Terminate connection...	Allows terminating the available connections	

The "Service Management..." window provides the load and save view buttons. A set of predefined views related to standard services and profiles are available on the BlueNRG GUI SW package, Application/ Services_View folder ("Alert Notification Server, Blood Pressure Sensor, Glucose Sensor, Heart Rate Sensor, Health Thermometer Sensor, Find Me Target, Phone Alert Server, HID, Time, …")

User can load them through the "Load View" button and add to his application database.

Table 2. GUI ACI utility window: available central operations

Operation	Associated actions	Notes
Scanning	Allows placing the device in scanning mode by selecting: - GAP procedure (limited, general, general-connection establishment and terminate general-connection establishment procedures) - Enable or disable filters - Set own address type - Set passive or active scan - Set scanning interval and window	
Connections	Allows connecting to a peer device by: - Searching for the devices in advertising - Selecting the device to which to connect - Selecting the connection parameters - Peer address and type - Scan interval and window - Connection interval (min. and max.) - Latency - Supervision timeout - Connection event length (min. and max.)	The addresses of the detected advertising devices are displayed
Update connections	Allows updating the connection parameters of available connections by: - Selecting the specific connection to be updated - Setting the new connection parameters - Connection interval (min. and max.) - Latency - Supervision timeout - Connection event length (min. and max.)	

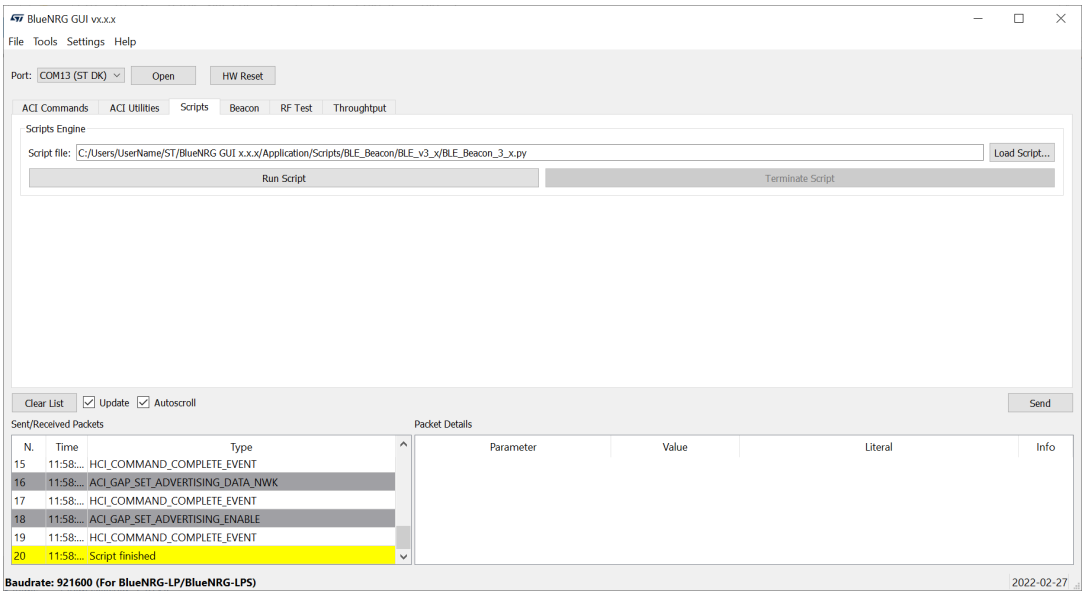
Table 3. GUI ACI utility window: available peripheral operations

Operation	Associated actions	Notes
Advertising	Allows placing a peripheral device in advertising mode by selecting: - Discoverable mode (limited, non-discoverable and general discoverable) - Type (ADV_IND, ADV_SCAN_IND, ADV_NONCONN_IND) - Set local name and type (complete or short) - Advertising intervals (min. and max.) - Policy: - Allows scan request from any, allows connect request from any - Allows scan request from white list only, allow connect request from any - Allows scan request from any, allows connect request from white list only	
Update advertising data	Allows updating the advertising data; Allows setting the scan response data; Allows updating the location UUID, major and minor number defined on the Beacon window	

2.2.4 GUI script window

The GUI script window allows the user to load and run a python script built using the available set of ACI commands and related events. For a list of supported HCI and ACI script commands and parameters, refer to those commands available in the GUI ACI command window.

Figure 8. BlueNRG GUI script window section



The script engine supports other utility commands listed in the following table.

Table 4. GUI script window: utility commands

Command name	Parameters	Description
ERROR	User message	Raises an exception with a user-defined debug message
CLEAR_QUEUE	None	Removes all the pending events collected within the internal event queue
GET_CHAR	None	Allows a specific char to be entered as input (as the C get_char() API)
GET_ALL_COM_PORT	None	Returns the list of all COM ports (ST DK kits and not)
GET_ALL_ST_DK_COM_PORT	None	Returns the list of all ST DK kits COM ports sorted by COM port number
GET_FILE	None	Allows a specific file to be selected as input
GET_NAME	None	Returns the device name within an advertising packet
GET_VALUE	Array of bytes	Converts the array of bytes to an integer value; e.g.,: X= [0x33,0x22] GET_VALUE(X) = 0x2233
GET_LIST	Integer, number of bytes	Converts the integer value to an array of number of bytes; e.g.,: X= 0x2233 GET_LIST(X, 2)= [0x33,0x22]
GET_STACK_VERSION	None	Returns the device information (HW version and FW version) as (hw, fw)
GET_RAND_KEY	None	Returns a random number between 0 and 999999
HW_BOOTLOADER	None	Hardware bootloader activation (it is not supported on the BlueNRG-LP, BlueNRG-LPS devices kit platforms)
HW_RESET	None	HW reset
INFO	String to be displayed	Opens a message window and show the input parameter. Script is blocked until user presses OK button
INSERT_FLOAT_NUMBER	None	Allows a float value
INSERT_INT_NUMBER	None	Allows an integer value to be entered
INSERT_PASS_KEY	None	Allows entering a pass key value for the security pass key method
IS_BlueNRG_MS	None	Returns TRUE if the device is a BlueNRG-MS, FALSE otherwise
IS_BlueNRG_1	None	Returns TRUE if the device is a BlueNRG-1, FALSE otherwise
IS_BlueNRG_2	None	Returns TRUE if the device is a BlueNRG-2, FALSE otherwise
IS_BlueNRG_2N	None	Returns TRUE if the device is a BlueNRG-2N, FALSE otherwise
IS_BlueNRG_LP	None	Returns TRUE if the device is a BlueNRG-LP, FALSE otherwise
IS_BlueNRG_LPS	None	Returns TRUE if the device is a BlueNRG-LPS, FALSE otherwise
IS_BlueNRG_LPX	None	Returns TRUE if the device is a BlueNRG-LP or a BlueNRG-LPS, FALSE otherwise
PRINT	String	Print utility: displays information on GUI sent/received packets (only integer decimal format is supported)
RESET	None	SW reset
SLEEP	time	Sleeps for the period 'time' in seconds
SET_MODE	Mode	Sets stack mode (1,2,3,4). Mode 4 is only supported on BlueNRG-MS from FW stack version 7.1b. Only for the BlueNRG-MS devices.
SET_PUBLIC_ADDRESS	Public address	Set public address (optional)
SENSORDEMO_GET_TEMPERATURE	None	Allows retrieval of the temperature value from the ACI_ATT_READ_RESP_EVENT (only for the SensorDemo_Central script)
SENSORDEMO_GET_ACCELERATION	None	Allows retrieval of the acceleration values (x,y,z) from the ACI_GATT_NOTIFICATION_EVENT

Command name	Parameters	Description
		(only for the SensorDemo_Central script)
TIME	None	Returns the time as a floating point number expressed in seconds since the epoch, in UTC

The following pseudo code describes how to initialize a BlueNRG-MS, BlueNRG-1 or BlueNRG-2 device as a peripheral using a simple python script:

```
#Reset device
HW_RESET()
#Init GATT
if BLE_STACK_V_3_X:
    ACI_GATT_SRV_INIT()
else:
    ACI_GATT_INIT()

#Init GAP as central device
ACI_GAP_INIT(Role=CENTRAL)
```

When a script calls a command which generates specific events, the script can detect them by using the WAIT_EVENT (event_code=None, timeout=None, continueOnEvtMiss=False, **param_checks) command.

Table 5. WAIT_EVENT macro-command

Command name	Description	Parameters	Return
WAIT_EVENT	It waits for an event with 'Event Code' parameter equal to event_code. If no event_code is indicated, the macro-command waits for any event. Optional filtering parameters allow additional filters to be defined on event fields	- event_code = none (default) - timeout = none (default) - continueOnEvtMiss = false (default) - param_checks = optional filtering parameters	- An event with its parameters None, if a timeout occurs and the input parameter "continueOnEvtMiss" is set to true - An event with its parameters - None, if a timeout occurs and the input parameter "continueOnEvtMiss" is set to true

The WAIT_EVENT macro-command waits for an event with 'Event Code' parameter equal to event_code. If no event_code is indicated, the macro-command waits for any event.

The timeout parameter allows the event timeout to be set. If no timeout is set, the macro-command awaits until an event occurs. If a timeout (greater than zero) is set and continueOnEvtMiss is false and no event occurs before the timeout, an HCITimeoutError error happens. Otherwise, if the input parameter continueOnEvtMiss is true and a timeout (greater than zero) is set, the macro-command returns the value none even when no event occurs before the timeout.

If one or more optional filtering parameters are specified, the macro-command performs a check on them and it returns only the first detected event that satisfies these parameters. The events received before the one returned are discarded.

The WAIT_EVENT() command return value can be:

- an event
- none, if a timeout occurs and the input parameter "continueOnEvtMiss" is set to true

An HCI timeout error exception is raised when a timeout occurs. Each coming event is stored in an internal queue. User is requested to call a specific WAIT_EVENT utility command in order to handle each received event and remove it from the internal queue. The CLEAR_QUEUE utility command allows the internal queue to be cleared in order to remove all the collected events not handled through the WAIT_EVENT utility command. The event_code parameter can be one of the following values:

Table 6. Event codes with relative event parameter types

event_code	Event parameter type	Event parameter type value
HCI_LE_META_EVENT	Subevent _Code	Refer to specific device Bluetooth LE Stack ACI APIs and events doxygen documentation available on STSW-BNRGUI SW package, Docs Docs\gui_aci_html: ACI events, HCI LE meta events section
HCI_VENDOR_EVENT	Ecode	Refer to specific device Bluetooth LE Stack ACI APIs and events doxygen documentation available on STSW-BNRGUI SW package, Docs Docs\gui_aci_html: ACI events, ACI GAP events ACI GATT/ATT events ACI L2CAP events ACI HAL event sections
NA	NA	Refer to specific device Bluetooth LE stack ACI APIs and events doxygen documentation available on STSW-BNRGUI SW package, Docs Docs\gui_aci_html: ACI events, HCI event section

Below are some code examples using the WAIT_EVENT() macro-command:

Example 1:

```
#Wait any events
evt = WAIT_EVENT()
if (evt.event_code == HCI_LE_META_EVENT):
     #User specific code ……
elif (evt.event_code==HCI_VENDOR_EVENT):
      #User specific code ……
```

Example 2:

```
# Wait an HCI_LE_META_EVENT
evt = WAIT_EVENT(HCI_LE_META_EVENT)
# Using evt.get_param('Subevent_Code').val it's possible to identify the specific
HCI_LE_META_EVENT
# parameter type value
evtCode = evt.get_param('Subevent_Code').val
# Check if received event is HCI_LE_CONNECTION_COMPLETE_EVENT
if (evtCode == HCI_LE_CONNECTION_COMPLETE_EVENT):
  # If Connection Complete Status is success, get connection handle
  if evt.get_param('Status').val==0x00:
    conn_handle= evt.get_param('Connection_Handle').val
```

Note: HCI_LE_ENHANCED_CONNECTION_COMPLETE_EVENT on the BlueNRG-LP, BlueNRG-LPS Bluetooth LE stack v3.x

Example 3:

```
#Wait HCI_VENDOR_EVENT event_code
evt = WAIT_EVENT(HCI_VENDOR_EVENT)
#Using evt.get_param('Ecode').val it's possible to identify the specific HCI_VENDOR_EVENT
parameter type value
evtCode= evt.get_param('Ecode').val
if BLE_STACK_V_3_X:
   if(evtCode == ACI_GATT_CLT_NOTIFICATION_EVENT):
       conn_handle=evt.get_param('Connection_Handle').val
else:
   if(evtCode == ACI_GATT_NOTIFICATION_EVENT):
       conn_handle=evt.get_param('Connection_Handle').val
```

Example 4:

```
#Wait the Ecode ACI_GATT_PROC_COMPLETE_EVENT (HCI_VENDOR_EVENT #event_code).
if BLE_STACK_V_3_X:
   WAIT_EVENT(HCI_VENDOR_EVENT, timeout=30, Ecode=ACI_GATT_CLT_PROC_COMPLETE_EVENT)
else:
   WAIT_EVENT(HCI_VENDOR_EVENT, timeout=30, Ecode=ACI_GATT_PROC_COMPLETE_EVENT)
```

Note: If no timeout parameter is specified, it awaits until the ACI_GATT_PROC_COMPLETE_EVENT.

Example 5:

```
#Wait an event for 10 seconds with continueOnEvtMiss set to True#
If no event occurs, the script continues (no exception is raised).
WAIT_EVENT(timeout=10, continueOnEvtMiss =True)
```

Note: If continueOnEvtMiss parameter is set to false and if no event within the selected timeout occurs, an exception is raised.

Example 6:

```
#Wait the HCI_DISCONNECTION_COMPLETE_EVENT event_code
WAIT_EVENT(HCI_DISCONNECTION_COMPLETE _EVENT)
```

Example 7:

```
#Create a Connection and wait for the HCI_LE_CONNECTION_COMPLETE_EVENT
ACI_GAP_CREATE_CONNECTION(Peer_Address=[0x12, 0x34, 0x00, 0xE1, 0x80, 0x02])
event= WAIT_EVENT(HCI_LE_META_EVENT,
timeout=30,Subevent_Code=HCI_LE_CONNECTION_COMPLETE_EVENT)
if event.get_param('Status').val==0x00:
   # Store the connection handle
   conn_handle= event.get_param('Connection_Handle').val
   #User defined code…
```

Note: ACI_GAP_CREATE_CONNECTION() has different parameters on the BlueNRG-LP, BlueNRG-LPS Bluetooth LE stack v3.x. Refer to related API documentation.

GUI script engine loading and running steps

To load and run a python script using the BlueNRG GUI script engine, the following steps must be observed:

1. In the BlueNRG GUI, Scripts window, Script Engine section, click on tab "…", browse to the script location and select the script
2. Click on the "Run Script" tab to run the script. The execution flow (commands and events) is displayed in the BlueNRG GUI "Sent/Received Packets" section

In the BlueNRG GUI SW package and future versions, some reference scripts are available in the GUI/scripts folder.

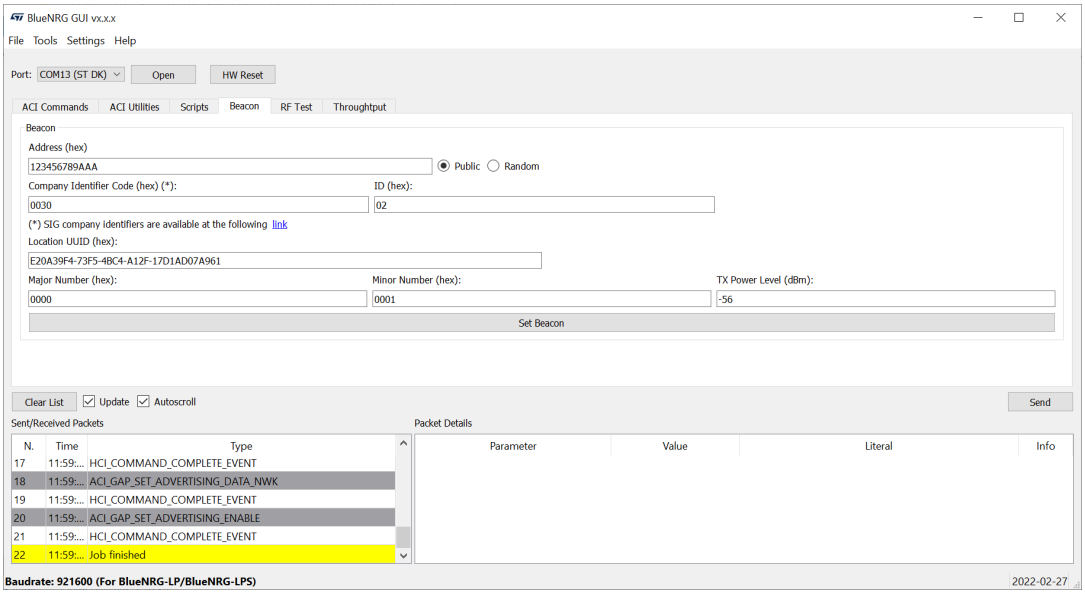
Note: To write and use the scripts, the user should have some knowledge of the Python language (Python 3.7.6), and a good understanding of the Bluetooth LE stack ACI commands and related events.

Note: Python 3.7.6 is not backward compatible with Python 2.7.6. Refer to https://www.python.org/ for more information about Python 2 not anymore supported and migration steps to Python 3.

2.2.5 GUI Beacon window

The BlueNRG GUI Beacon window provides some tabs to configure a BlueNRG-MS, BlueNRG-2N, BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS devices as a Bluetooth LE Beacon device that transmits advertising packets with specific manufacturer data.

Figure 9. **BlueNRG GUI Beacon window**



The user can configure the advertising data fields in the following table for the Bluetooth LE Beacon device through the BlueNRG GUI Beacon window configuration parameters.

Table 7. **BlueNRG GUI Beacon window configuration parameters**

Data field	Description	Notes
Address	Device address	
Public or random	Device address type	
Company identifier code	SIG company identifier	Default is 0x0030 (STMicroelectronics)
ID	Beacon ID	Fixed value
Location UUID	Beacons UUID	Used to distinguish specific Beacons from others
Major number	Identifier for a group of beacons	Used to group a related set of Beacons
Minor number	Identifier for a single beacon	Used to identify a single Beacon
TxPower Level	2's complement of the Tx power	Used to establish how far you are from the device

To configure the selected platform as a Bluetooth LE Beacon device, click on the “Set Beacon” tab.

2.2.6 GUI RF test window

The BlueNRG GUI provides the RF test window that permits to perform the following tests:

1. Start/stop a tone on a specific Bluetooth LE RF channel
2. Perform a Bluetooth LE packer error rate (PER) tests using Bluetooth LE direct test mode (DTM) commands

Start/stop a tone

To start a tone on a specific RF Bluetooth LE channel:

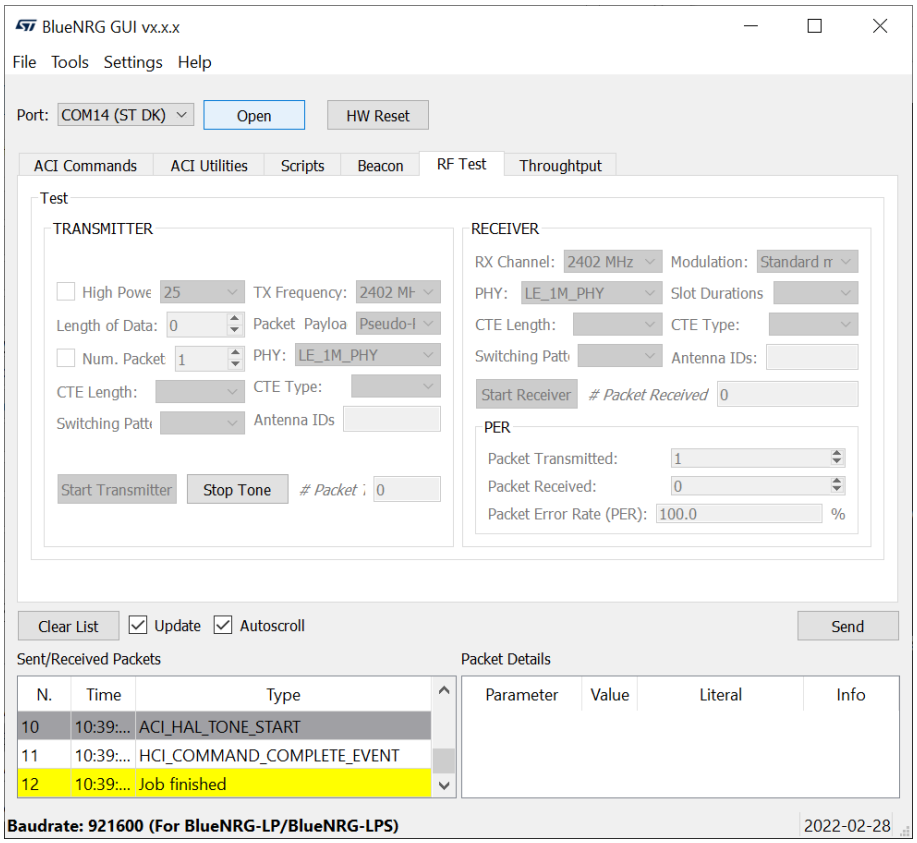
1. Connect a BlueNRG-MS, BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP and BlueNRG-LPS platforms to PC USB port
2. Launch an instance of BlueNRG GUI
3. Open the relative COM port

4. Go to RF test window and in the TRANSMITTER section:
   a. Set Bluetooth LE channel by TX frequency combo box
   b. Set TX power in the related combo box
   c. Click on the "Start Tone" button

To stop a tone on a specific RF Bluetooth LE channel:

1. Go to RF test window and in the TRANSMITTER section:
   a. Click on stop tone button (stop button is available only when a tone is started)

Figure 10. GUI RF test: start a tone



Direct test mode (DTM) tests

The BlueNRG GUI RF test let you target a packet error rate test scenario using the Bluetooth LE direct test mode commands.

Two sections are available:

1. TRANSMITTER section to transmit reference packets to a fixed interval
2. RECEIVER section to receive reference packets to a fixed interval

TRANSMITTER section

This section let you set the following items:

- The power level of transmitter
- The frequency of transmitter
- Length of data to transmit in each packet
- Packet payload format as defined on Bluetooth low energy specification, direct test mode section
- PHY used by transmitter
- Check the capability to set the number of packets to transmit
- PHY used by transmitter

- Modulation

Clicking on "Start Transmitter" button, test reference packets are sent to a fixed interval.

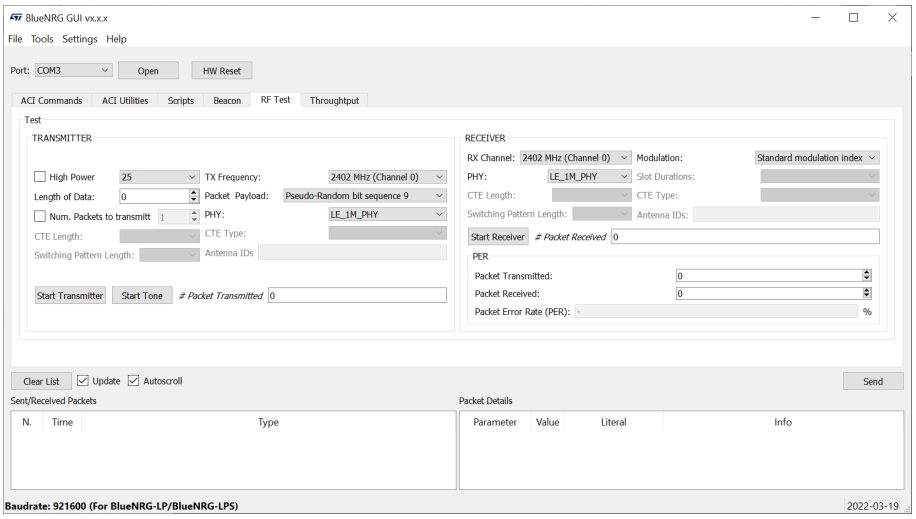
RECEIVER section

This section let you set the following items:

- The frequency or channel of receiver
- PHY used by receiver
- Modulation

Clicking on "Start Receiver" button, test reference packets are received to a fixed interval.

Figure 11. GUI RF test: transmitter and receiver sections



Packet error rate (PER) test procedure

To perform a packet error rate test using the standard Bluetooth LE direct test mode commands (HCI_LE_Transmitter_Test, HCI_LE_Receiver_Test and HCI_LE_Test_End), the following steps are needed:

Start PER test

1. Connect two platforms (BlueNRG-MS, BlueNRG-2N, BlueNRG-1, BlueNRG-2, BlueNRG-LP and BlueNRG-LPS) to PC USB ports
2. Open two instances of BlueNRG GUI on both devices (called TX and RX devices, respectively)
3. In each instance of BlueNRG GUI, open COM port relative to the TX/RX device
4. Ensure the antennas are plugged into the devices where applicable
5. In the GUI related to RX device:
   - Go to RF test window, RECEIVER section
   - Set RX frequency
   - Set PHY
   - Set modulation
   - Click on "Start Receiver" button to start receiver test

6. In the GUI related to TX device:
   - Go to RF test window, TRANSMITTER section
   - Set TX power
   - Set TX frequency
   - Set length of data
   - Set packet payload format
   - Set PHY
   - Set number of packets to transmit (optional)
   - Click on "Start Transmitter" button, to start transmitter test

Stop PER test

1. In the GUI related to TX device:
   - Go to RF test window, TRANSMITTER section
   - Click on "Stop Transmitter" button. The number of transmitted packets are displayed on #Packet Transmitted field.
2. In the GUI related to RX device
   - Go to RF test window, RECEIVER section
   - Click on "Stop Receiver" button. The number of received packets are displayed on #PacketReceived field.

Get PER (packet error rate) value

1. In the GUI related to RX device:

- Go to RF test window, RECEIVER section
- In the PER section, insert the number of transmitted packet from TX device in the packet transmitted field (read this value from TRANSMITTER section in the GUI related to TX device)
- The PER (packet error rate) value is showed in the packet error rate field

Figure 12. **GUI RF test, PER test: TX device**

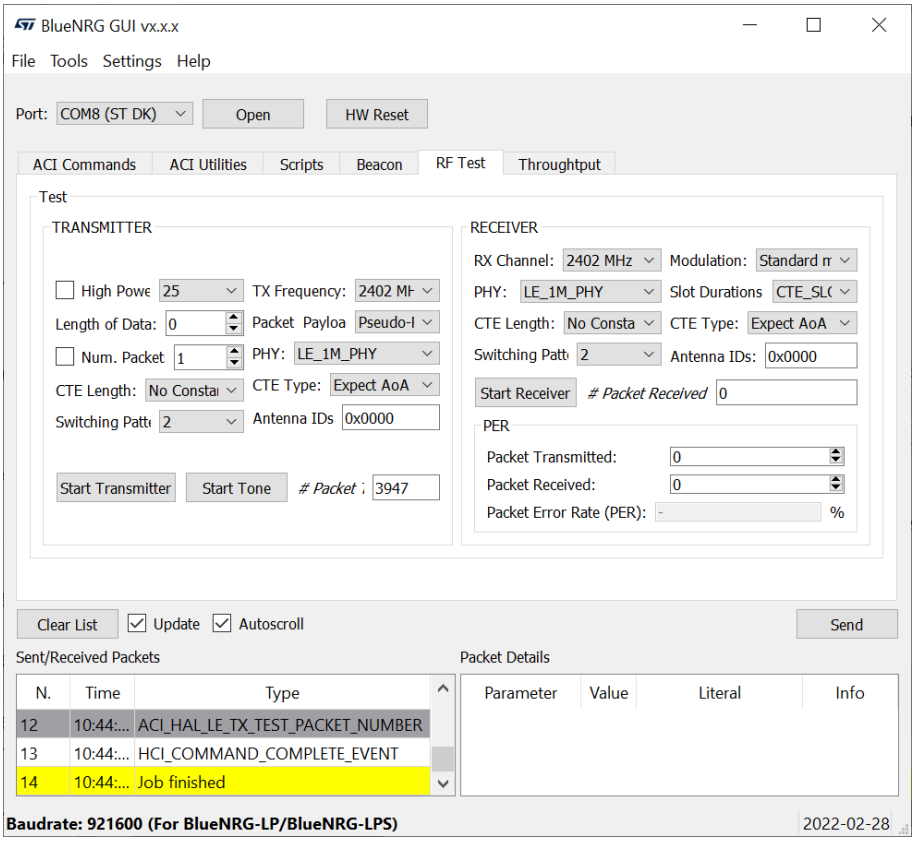
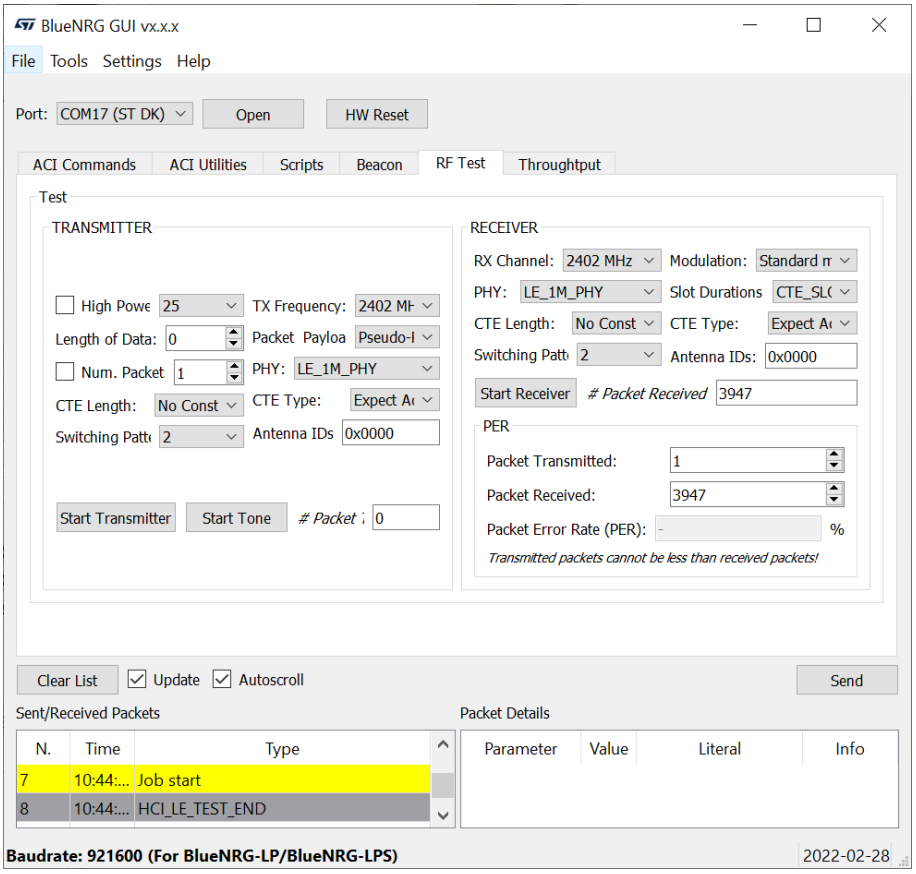


Figure 13. GUI RF Test, PER test: RX device



Note: Starting from the STSW-BNRGUI v3.2.0 or later, BlueNRG-1,2 Bluetooth LE stack v2.x, BlueNRG-LP, BlueNRG-LPS Bluetooth LE stack v3.x, if user specifies the number of packets to be transmitted on a related field (number of packets to transmit), the ACI_HAL_TRANSMITTER_TEST_PACKETS API is used to perform the transmitter test. This API is a customized version of the HCI_LE_TRANSMITTER_TEST API which allows the number of packets to be specified and to be sent. Please notice that user must uncheck the field "Num. Packets to transmit" in order to use the standard HCI_LE_TRANSMITTER_TEST API.

An example follows about a TX/RX context with a TX device sending 1000 packets which are received from the RX device.

Figure 14. **GUI RF Test, PER test: Send 1000 packets**

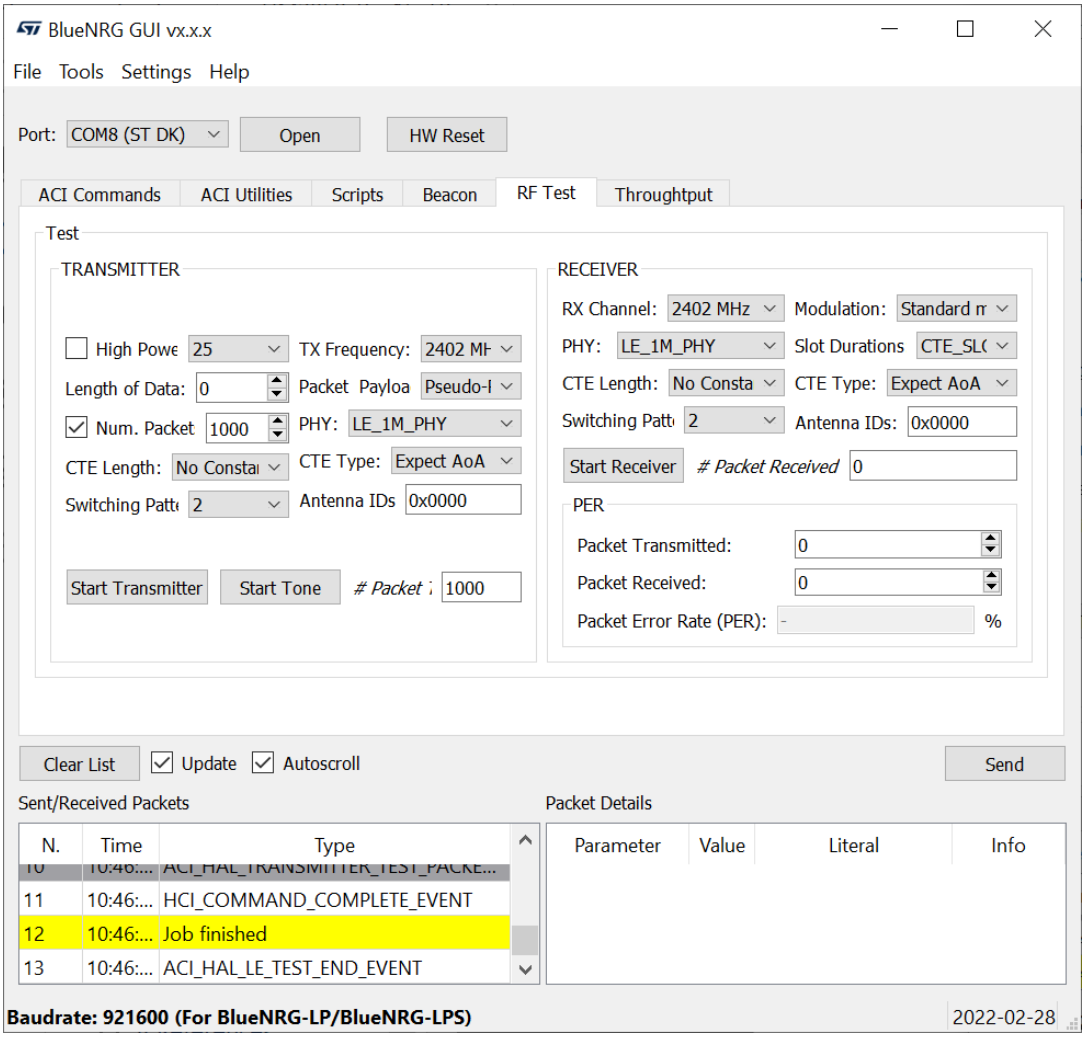
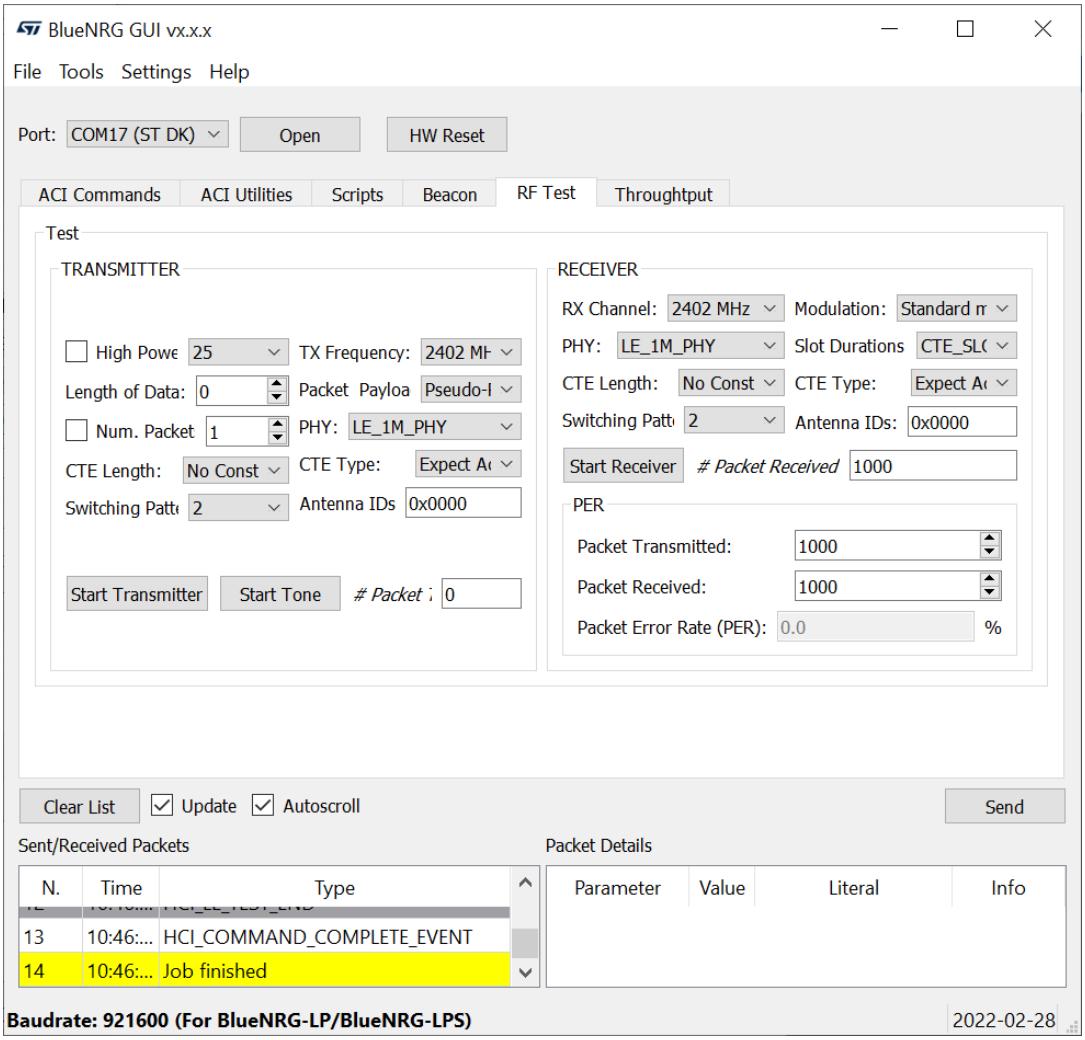


Figure 15. **GUI RF Test, PER test: Receive 1000 packets**



2.2.7 GUI RF throughput window

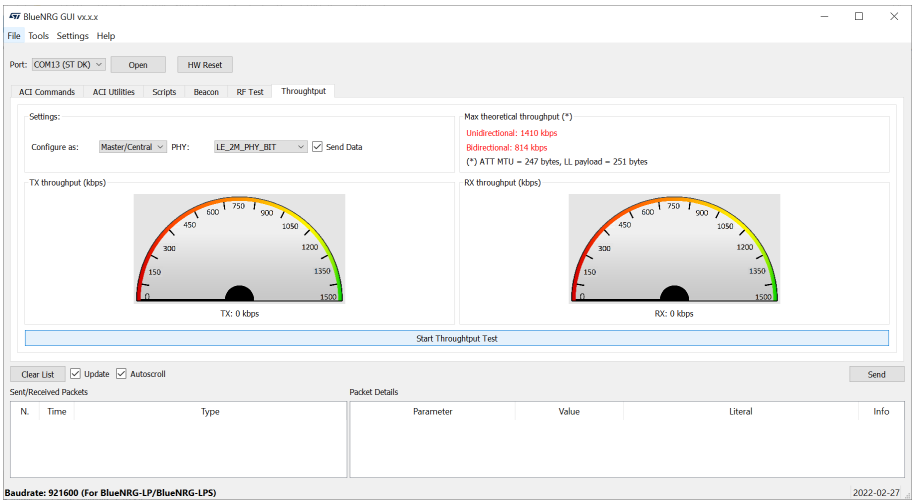
The BlueNRG GUI RF throughput window allows throughput tests to be performed by setting up a Bluetooth LE communication between two devices, master and slaves, and setting up one of the following scenarios:

1. Unidirectional communication (characteristic notification) from slave (GATT server) to master (GATT client) device
2. Unidirectional communication (write without response) from master (GATT client) to slave (GATT server) device
3. Bidirectional communication (characteristic notification) from slave (GATT server) to master (GATT client) device and (write without response) from master (GATT client) to slave (GATT server) device

The theoretical throughput is calculated taking in account the ATT MTU = 247 bytes and LL payload = 251 bytes for the BlueNRG-1, BlueNRG-2, BlueNRG-LP devices (it supports data length extension feature) and ATT MTU = 247 bytes and LL payload = 27 bytes for the BlueNRG-1 device.

Note: *1. This feature requires the BlueNRG-1, 2 devices to be loaded with a dedicated DTM binary image (DTM_UART_Throughput.hex) stored on Firmware/{BlueNRG1|BlueNRG2}/DTM folders (serial baudrate is 115200 for the BlueNRG-1,2 devices). The BlueNRG-LP DTM binary image (DTM_UART_WITH_UPDATER.hex) is available on Firmware/STEVAL-IDB011V1/DTM folder.*

Figure 16. BlueNRG GUI RF throughput window



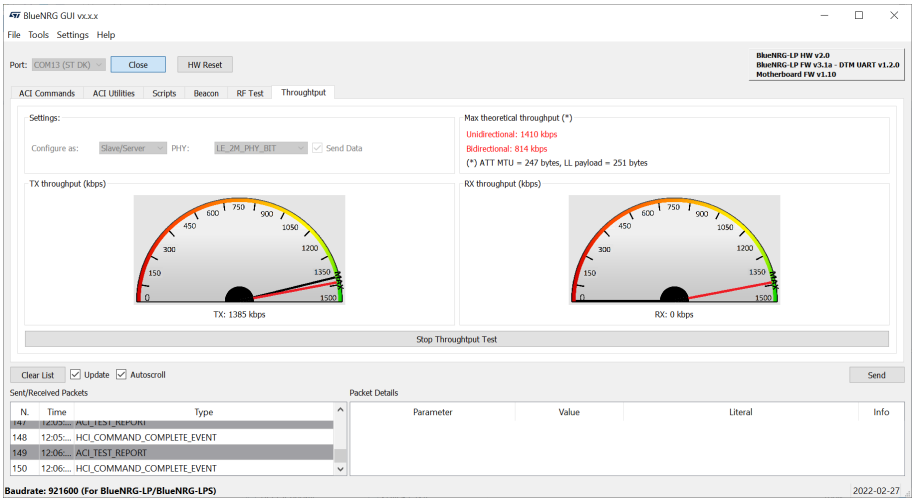
Unidirectional (slave to master) throughput test

1. Connect two platforms (BlueNRG-LP taken as example) to PC USB ports
2. Open two instances of the BlueNRG GUI on both devices (called master and slave devices, respectively)
3. In each instance of the BlueNRG GUI, open COM port relative to the master/slave device
4. Ensure the antennas are plugged into the devices, where applicable

Slave device

1. On Settings, Configure As: select Slave/Server and set Send data checkbox
2. On Settings, select Send Data to enable communication from slave to master device (characteristic notification)
3. Click on "Start Throughput" button to start throughput test:
   a. Slave device enters discoverable mode
   b. Once Master device connects to Slave device, the actual throughput value (numeric value and black line) and the theoretical expected throughput value (numeric value and red line) are displayed on Slave TX window and on Master RX window
4. Click on “Stop throughput” to stop the test

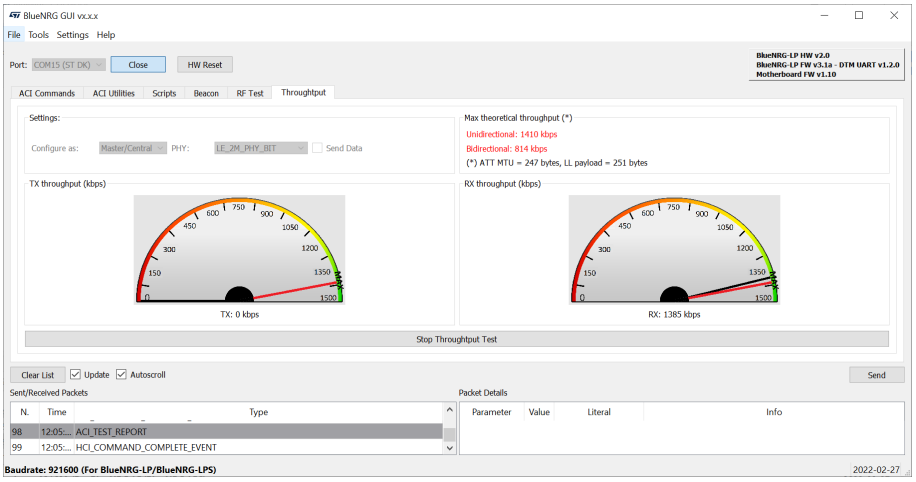
Figure 17. **BlueNRG GUI RF throughput unidirectional, slave sends data**



Master device

1. On settings, configure as: select Master/Central and unset Send Data checkbox
2. Click on "Start Throughput" button to start throughput test:
   a. Master connects to slave device and then it enables characteristic notifications: slave device starts notifications flow
   b. Bluetooth LE communication from slave to master (characteristic notification) is started.The actual throughput value (numeric value and black line) and the theoretical expected throughput value (numeric value and red line) are displayed on Slave TX window and on Master RX window.
3. Click on “Stop throughput” to stop the test

Figure 18. **BlueNRG GUI RF throughput unidirectional, master receives data**



Unidirectional (master-to-slave) throughput test

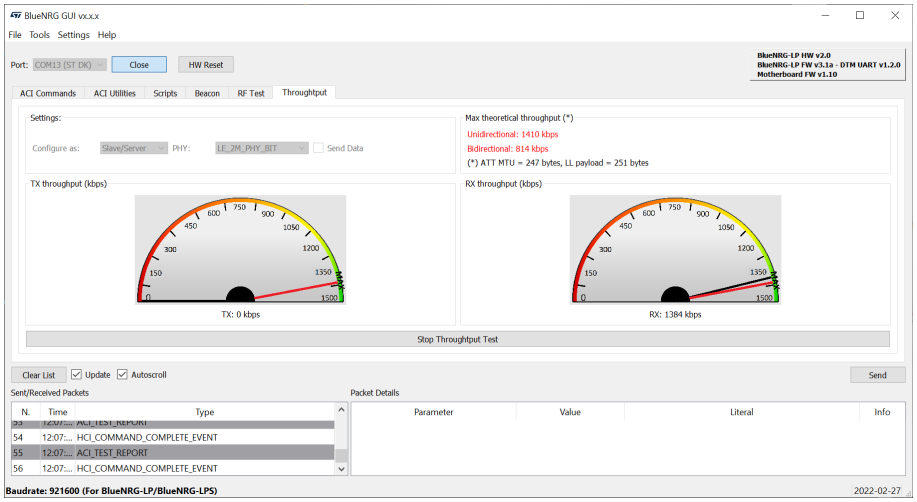
1. Connect two platforms (BlueNRG-LP as example) to PC USB ports
2. Open two instances of the BlueNRG GUI on both devices (called master and slave devices, respectively)
3. In each instance of the BlueNRG GUI, open COM port relative to the master/slave device
4. Ensure the antennas are plugged into the devices where applicable

Slave device

1. On Settings, Configure As: select Slave/Server and unset Send Data checkbox

2. Click on "Start Throughput" button to start throughput test:
   a. Slave device enters discoverable mode
   b. Once master device connects to slave device, the actual throughput value (numeric value and black line) and the theoretical expected throughput value (numeric value and red line) are displayed on slave RX window and on master TX window
3. Click on “Stop throughput” to stop the test

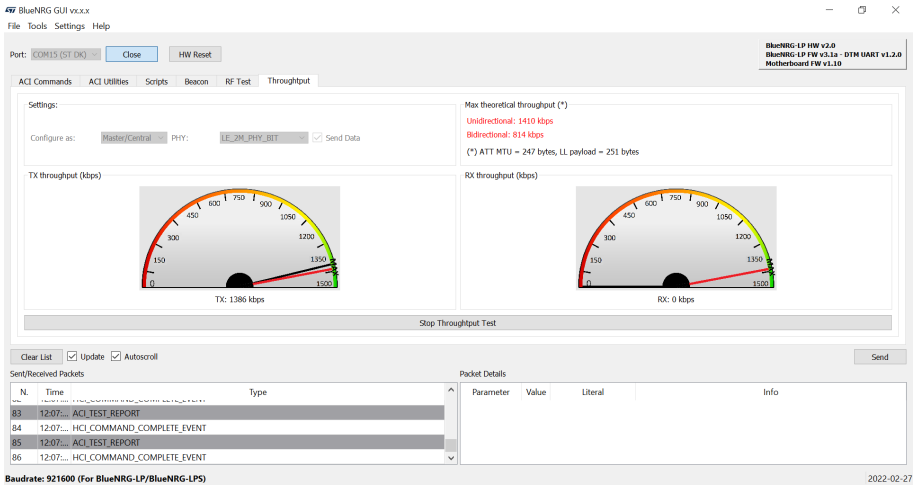
Figure 19. BlueNRG GUI RF throughput unidirectional:slave receives data



Master device

1. On Settings, Configure As: select Master/Central and set Send Data checkbox
2. On Settings, select Send Data to enable communication from master to slave device (characteristic write without response)
3. Click on "Start Throughput" button to start throughput test:
   a. Master connects to slave device
   b. Bluetooth LE communication from master to slave (characteristic write without response) is started. Once master device connects to slave device, the actual throughput value (numeric value and black line) and the theoretical expected throughput value (numeric value and red line) are displayed on slave RX window and on master TX window
4. Click on “Stop throughput” to stop the test.

Figure 20. BlueNRG GUI RF throughput unidirectional:master sends data



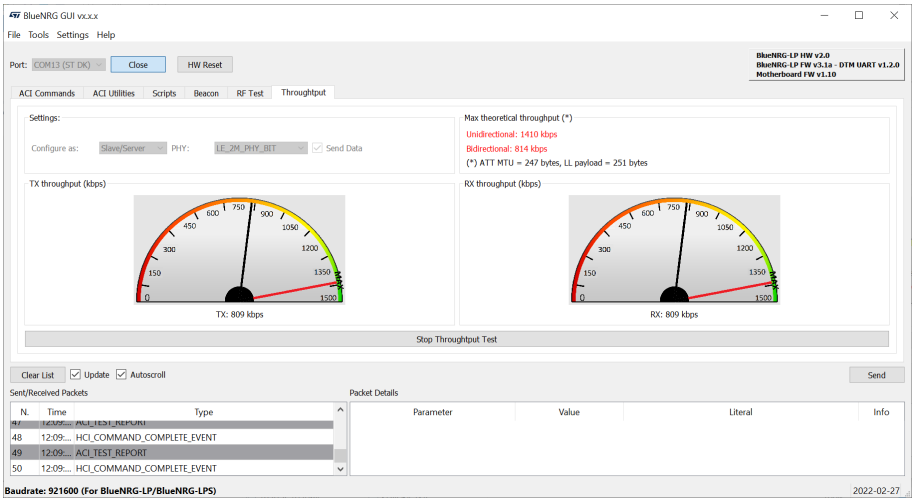
Bidirectional (slave-to-master, master-to-slave) throughput test

1. Connect two platforms (BlueNRG-LP as example) to PC USB ports
2. Open two instances of the BlueNRG GUI on both devices (called master and slave devices, respectively)
3. In each instance of the BlueNRG GUI, open COM port relative to the master/slave device
4. Ensure the antennas are plugged into the devices where applicable

Slave device

1. On Settings, Configure As: select Slave/Server
2. On Settings, select Send Data to enable communication from slave-to-master device (characteristic notification)
3. Click on "Start Throughput" button to start throughput test:
   a. Slave device enters discoverable mode
   b. Once master device connects to slave device, the actual throughput value (numeric value and black line) and the theoretical expected throughput value (numeric value and red line) are displayed on slave master TX, RX windows
4. Click on "Start Throughput" button to start throughput test

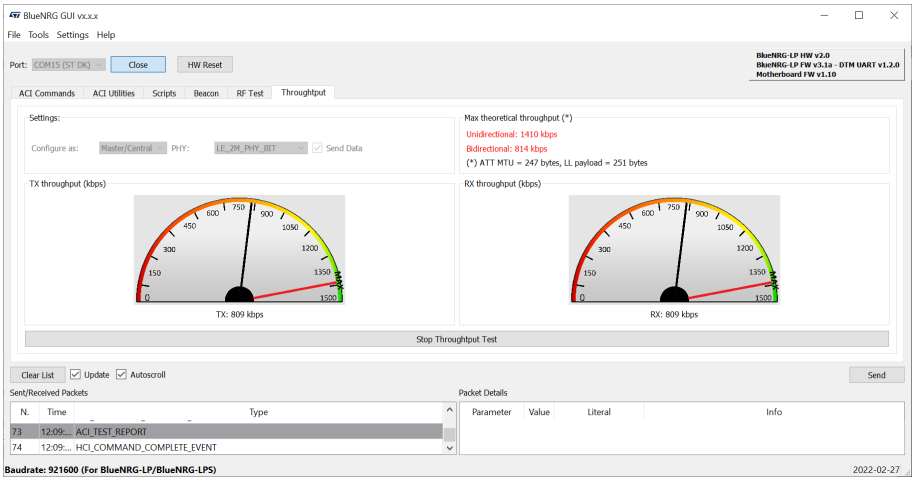
Figure 21. BlueNRG GUI RF throughput bidirectional:slave sends data



Master device

1. On Settings, Configure As: select Master/Central and set Send Data checkbox
2. On Settings, select Send Data to enable communication from master to slave device (characteristic write without response)
3. Click on "Start Throughput" button to start throughput test:
   a. Master connects to slave device and then it enables characteristic notifications: slave device starts notifications flow
   b. Bluetooth LE communication from slave to master (characteristic notification) is started. The actual throughput value is displayed the actual throughput value (numeric value and black line) and the theoretical expected throughput value (numeric value and red line) are displayed on slave, master TX, RX windows
4. Click on “Stop throughput” to stop the test

Figure 22. **BlueNRG GUI RF throughput bidirectional:master sends data**



3 Script launcher utility

The script launcher is a standalone utility, which let you run a script without using the BlueNRG GUI script engine tool.

The script launcher utility (BlueNRG_Script_Launcher.exe) is included in the BlueNRG GUI software package in the application folder.

3.1 Requirements

To use the script launcher utility on a specific device, the corresponding platform must be connected to a PC USB port and loaded with the same pre-built binary file used for the BlueNRG GUI, as described in Section 2.1 Requirements (BlueNRG_VCOM_x_x.hex for the BlueNRG-MS platforms and specific DTM binary files for the BlueNRG-1, BlueNRG-2, BlueNRG-2N, BlueNRG-LP and BlueNRG-LPS platforms).

Note: When using the script launcher utility on a specific device and related COM port, please make sure that the expected serial baud rate is properly set through the option -b:

- *BlueNRG-LP, BlueNRG-LPS serial baud rate: 921600 (default configuration)*
- *All other devices, serial baud rate: 115200*

3.2 Script launcher utility options

To use the script launcher utility on a specific device, open a window DOS shell and launch the relative BlueNRG_Script_Launcher.exe (type –h to list all the supported options):

C:\Users\{UserName}\ST\BlueNRG GUI x.x.x\Application>BlueNRG_Script_Launcher.exe -h:

- -h, --help show this help message and exit
- -v, --version show program's version number and exit
- -g, --get get list of existing COM ports and exit
- -d, --debug debug script file (python pdb)
- -l, --log log data
- -p PORT, --port PORT select COM port
- -b BAUD_RATE, --baud rate BAUD_RATE set Baud Rate: 921600 (default).
  WARNING: 921600 is valid for the BlueNRG-LP, BlueNRG-LPS device ONLY;
  115200 MUST BE SET for all other devices
- -s SCRIPT_FILE, --script SCRIPT_FILE set script file name
- -z, --save save log in a file (*.csv, *.txt)

Option to get the list of all available COMx ports (devices are connected to the PC USB)

C:\Users\UserName\ST\BlueNRG GUI x.x.x\Application> BlueNRG_Script_Launcher.exe -g

List of available COM ports:

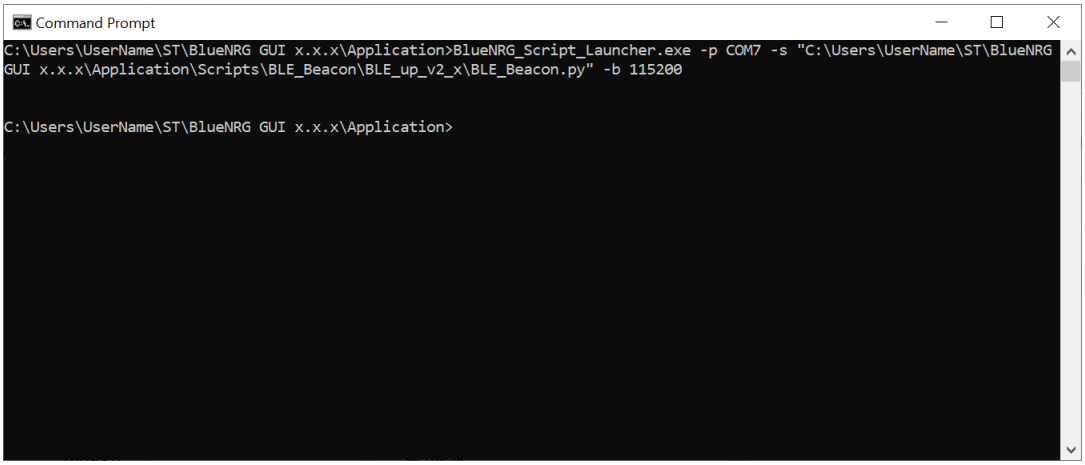
- COM7 (ST DK)

Options to run a script on a connected device

C:\Users\UserName\ST\BlueNRG GUI x.x.x\Application>BlueNRG_Script_Launcher.exe -p COM7 –s C:\Users\UserName\ST\BlueNRG GUI x.x.x\Application\Scripts\BLE_Beacon\BLE_up_v2_x\BLE_Beacon.py -b 115200

COM7 is the platform virtual COM port.

Figure 23. **Script launcher: run a script**

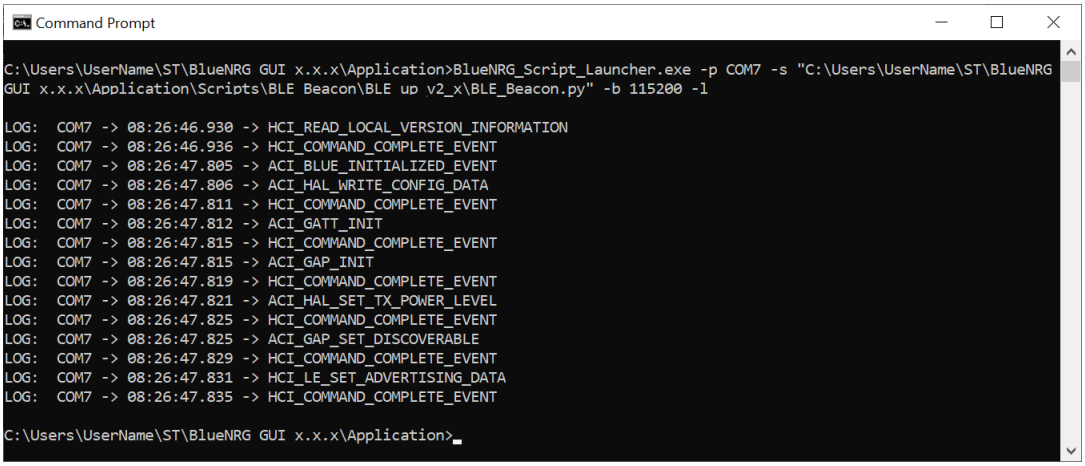


Options to run a script on a connected device with log data

C:\Users\UserName\ST\BlueNRG GUI x.x.x\Application>BlueNRG_Script_Launcher.exe -p COM7 –s C:\Users\UserName\ST\BlueNRG GUI x.x.x\Application\Scripts\BLE_Beacon\BLE_up_v2_x\BLE_Beacon.py -b 115200 -l

COM7 is the platform virtual COM port.

Figure 24. **Script launcher: run a script with log data**



4 References

Table 8. Reference information

What	Where	Description
STSW-BNRGGUI	BlueNRG-1, BlueNRG-2, BlueNRG-MS, BlueNRG-LP, BlueNRG-LPS tools and software section	BlueNRG GUI SW package
STSW-BLUENRG-DK	BlueNRG-MS, evaluation tools section	BlueNRG DK SW package for the BlueNRG-MS kits
STSW-BLUENRG1-DK	BlueNRG-1, BlueNRG-2 evaluation tools section	BlueNRG-1 and BlueNRG-2 DK SW package for the BlueNRG-1 and BlueNRG-2 Bluetooth LE v2.x stack family
STSW-BNRGLP-DK	BlueNRG-LP, BlueNRG-LPS evaluation tools section	BlueNRG-LP, BlueNRG-LPS DK SW package for the BlueNRG-LP, BlueNRG-LPS Bluetooth LE v3.x stack family
DS13280	BlueNRG-2N	The BlueNRG-2N datasheet
STSW-BNRGFLASHER	-STSW-BNRGFLASHER	The RF-Flasher Utility SW package

5 List of acronyms

Table 9. **List of acronyms used in this document**

Term	Meaning
BLE	Bluetooth low energy
DFU	Device firmware upgrade
HW	Hardware
IFR	Information register
SoC	System-on-chip
SW	Software
USB	Universal serial bus

Revision history

Table 10. **Document revision history**

Date	Version	Changes
24-May-2016	1	Initial release.
24-Jun-2016	2	Added reference to BlueNRG-1 device and minor text changes.
26-Jul-2016	3	Minor fixes for BlueNRG-1 support.
07-Oct-2016	4	Updated Section 2.1.1: "BlueNRG, BlueNRG-MS network coprocessors" and Section 2.2.2: "Tools"
20-Jun-2017	5	Minor text and formatting changes Added reference to BlueNRG-2 SoC
23-Oct-2017	6	Added: reference to Save As C Code utility, INSERT_INT_NUMBER and INSERT_FLOAT_NUMBER utility commands
05-Apr-2018	7	Updated Section 2.1.2 BlueNRG-1 and BlueNRG-2 network coprocessors, Section 2.2.1 GUI main window and Section 2.2.4 GUI script window.
07-Dec-2018	8	Updated Section 2.2.1 GUI main window, Section 2.2.3 GUI ACI utility window, Section 2.2.4 GUI script window.
05-Mar-2020	9	Updated figures throughout Section 2.2 The BlueNRG graphical user interface. Removed reference to the STSW-BNRG_V1DK in References and added Section 2.2.7 GUI RF throughput window. Updated Section 2.1.2 BlueNRG-1 and BlueNRG-2 network coprocessors.
14-Jul-2020	10	Added the BlueNRG-LP device reference to the document. Updated from figure 9 to figure 20.
24-Aug-2020	11	Added reference to the BlueNRG-2N network coprocessor device.
14-Jan-2021	12	Updated Section 2.2.7 GUI RF throughput window.
10-May-2021	13	Updated Section 1.1 System requirements and Section 2.2.4 GUI script window.
04-Oct-2021	14	Updated Section 2.2.1 GUI main window and Section 2.2.6 GUI RF test window.
06-Apr-2022	15	Added the reference to the BlueNRG-LPS and removed reference to the BlueNRG device. Updated Figure 1. BlueNRG GUI main window, Figure 2. Command packet table, Figure 3. Packet history and details , Figure 4. Raw packet dump, Figure 7. BlueNRG GUI ACI utility window, Figure 8. BlueNRG GUI script window section, Figure 9. BlueNRG GUI Beacon window, Figure 10. GUI RF test: start a tone, Figure 11. GUI RF test: transmitter and receiver sections, Figure 12. GUI RF test, PER test: TX deviceFigure 13. GUI RF Test, PER test: RX device, Figure 14. GUI RF Test, PER test: Send 1000 packets, Figure 15. GUI RF Test, PER test: Receive 1000 packets, Figure 16. BlueNRG GUI RF throughput window, Figure 17. BlueNRG GUI RF throughput unidirectional, slave sends data, Figure 18. BlueNRG GUI RF throughput unidirectional, master receives data, Figure 19. BlueNRG GUI RF throughput unidirectional:slave receives data, Figure 21. BlueNRG GUI RF throughput bidirectional:slave sends data and Figure 22. BlueNRG GUI RF throughput bidirectional:master sends data.
17-May-2022	16	Updated Table 4. GUI script window: utility commands.

Contents

List of tables

List of figures