UM2058
User manual

BlueNRG GUI SW package

Introduction

This document describes the BlueNRG GUI SW package (STSW-BNRGUI) which provides 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 with and to evaluate the capabilities of both BlueNRG and BlueNRG-MS Bluetooth low energy (BLE) network processors which are low power Bluetooth® smart IC, compliant, respectively, with the Bluetooth® 4.0 and 4.1 specifications, and supporting both master and slave roles. It also supports the BlueNRG-1 Bluetooth low energy system-on-chip which is a low power Bluetooth® smart IC compliant with the Bluetooth® 4.1 specifications, and supporting both master and slave roles.

The BlueNRG GUI allows to send standard and vendor-specific HCI commands to the device controller and to receive events 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 using the standalone script launcher utility which simply works through a PC DOS window, outside the BlueNRG GUI context. A set of scripts examples is provided within the SW package.

Contents

List of tables

List of figures

1 Getting started

This section describes all the system requirements for running the BlueNRG GUI or script launcher utility, as well as the related 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 XP SP3
  - Windows Vista
  - Windows 7
- At least 128 Mb of RAM
- 2 USB ports
- 40 Mb of hard disk space available
- Adobe Acrobat Reader 6.0 or later.

1.2 BlueNRG GUI SW package setup

- Extract the content of the BlueNRG_GUI_-x.x.x. Setup.zip file into a temporary directory.
- 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 using the following folders structure:

- **Application**:
  - It contains BlueNRG_GUI.exe and BlueNRG_Script_Launcher.exe PC applications. It also contains some reference scripts examples.
- **Docs**:
  - It contains the BlueNRG GUI SW package release notes, ACI and events html documentation, scripts examples html documentation.
- **Firmware**:
  - It contains the DTM and USB to Serial binary files to be used for the BlueNRG-1 development kit.
  - It contains the BlueNRG_VCOM_x.x.hex prebuilt binary file to be used for the STM32L1 microcontroller on BlueNRG, BlueNRG-MS kits motherboards. Is also contains the BlueNRG, BlueNRG-MS FW stack versions.
- **PCDriver**:
  - It contains the PC DFU and USB Virtual COM drivers

Notes:

(1) The provided BlueNRG_VCOM_x.x and DTM binary files can be rebuilt using the Virtual_COM_Port and DTM projects available, respectively, on STSW-BLUENRG-DK SW package, and on STSW-BLUENRG1-DK SW package (*Table 8: "References information"*). Each project provides a complete set of source and header files.

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, BlueNRG-MS network processors and BlueNRG-1 system-on-chip..

This utility can send standard and vendor-specific HCI commands to the controller and receive events from it. It lets the user configure each field of the HCI command packets to be sent and analyzes all received packets. In this way BlueNRG, BlueNRG-MS and BlueNRG-1 devices can be easily managed at low level.

2.1 Requirements

In order to use the BlueNRG GUI, make sure you have correctly set up your hardware and software (BlueNRG GUI installed).

2.1.1 BlueNRG, BlueNRG-MS network coprocessors

The STM32L1 in the STEVAL-IDB002V1, STEVAL-IDB005V1, 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 latest BlueNRG_VCOM_x_x.hex available within the BlueNRG GUI SW package on Firmware/STM32L1_prebuilt_images folder.

In order to download this binary image into the internal Flash of the STM32L1, the microcontroller must be put into a special DFU (device firmware upgrade) mode. To enter DFU mode:

1. BlueNRG, BlueNRG-MS development platforms (order codes: STEVAL-IDB002V1, STEVAL-IDB005V1, STEVAL-IDB005V1D)
   - Power up the board
   - Press and hold Push button
   - Reset the board using RESET button (keep Push Button pressed while resetting). The orange LED DL2 starts to blink.
   - Release the Push Button
   - Use BlueNRG GUI to Flash the device with the BlueNRG_VCOM_x_x.hex binary file (Tools -> Flash motherboard FW).

2. BlueNRG, BlueNRG-MS USB dongles (order codes: STEVAL-IDB003V1, STEVAL-IDB006V1)

- Press and hold the SW1 button
- Plug the USB dongle on a PC USB port. The orange LED D3 starts to blink.
- Use BlueNRG GUI to Flash the device with BlueNRG_VCOM_x_x.hex binary file (Tools -> Flash Motherboard FW).

2.1.2 BlueNRG-1 network coprocessor

The STEVAL-IDB007V, BlueNRG-1 kit has been preprogrammed with a demo application (BLE Sensor Demo). Hence, new firmware must be loaded into the BlueNRG-1 in order to configure it as network coprocessor and use the BlueNRG GUI.

The firmware images that must be programmed in order to configure the STEVAL-IDB007V1, BlueNRG-1 device as a network coprocessor are available within the BlueNRG GUI SW package on Firmware/BlueNRG-1/DTM folder.

Two network coprocessor configurations are available:

1. UART (DTM binary file: DTM_UART_16MHz.hex)
2. SPI (DTM binary file: DTM_SPI.hex)

In order to download the selected binary image into the internal Flash of the BlueNRG-1 device, follow these steps:

- BlueNRG-1 development platform (STEVAL-IDB007V1)
  - Connect the BlueNRG-1 board to a PC USB port
  - Open the BlueNRG Flasher tool available on STSW-BLUENRG1-DK SW package (*Table 8: "References information"*)
  - Select the COM port associated to the BlueNRG-1 platform to be configured as a network coprocessor
  - Select the specific DTM binary file through the Select file button
  - Select Mass erase option
  - Press Flash button for programming the device

Notes:

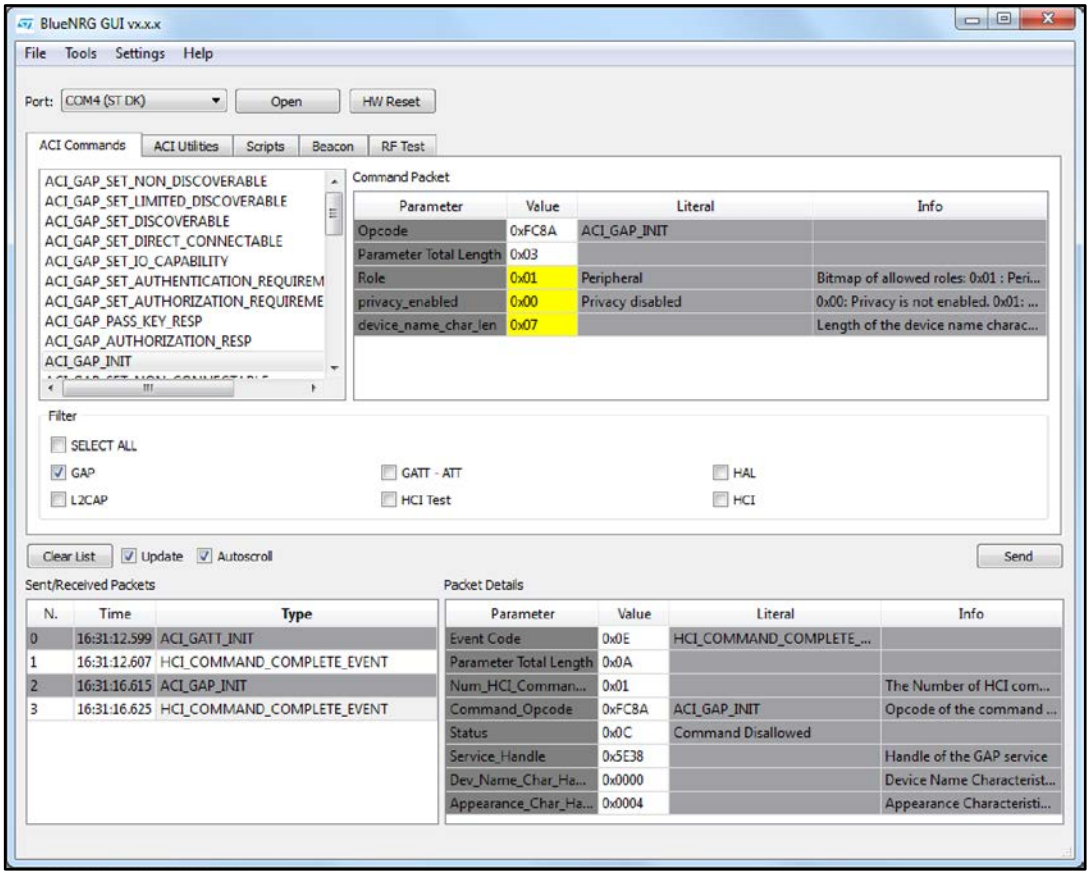
- The BlueNRG GUI, Tools, BlueNRG Updater … utility is intended only for updating a BlueNRG-1 device already configured as a network coprocessor, since it assumes that a specific updater/bootloader code is already loaded on the BlueNRG-1 device.
- DTM_UART_16MHz.hex and DTM_SPI.hex binary image files are built for also including the updater code which allows to update the specific DTM FW version trough the BlueNRG GUI, Tools, BlueNRG Updater … utility. DTM binary images without updater code are also provided within the SW package: DTM_UART_16MHz_NOUPDATER.bin, DTM_UART_32MHz_NOUPDATER.bin for UART configuration and DTM_SPI_NOUPDATER.bin for SPI configuration.
- Once the STEVAL-IDB007V1, BlueNRG-1 platform has been configured as network coprocessor (using DTM_UART_16MHz.hex or DTM_SPI.hex binary files), new DTM binary files versions can be updated directly using the BlueNRG GUI, Tools, BlueNRG Updater … utility as follow:
  - Connect the BlueNRG-1 board to a PC USB port
  - Open the BlueNRG GUI tool
  - Open the COM port associated to the BlueNRG-1 board
  - Select BlueNRG, Tools, BlueNRG Updater …
  - Select the DTM binary images without updater (DTM_UART_16MHz_NOUPDATER.bin, DTM_UART_32MHz_NOUPDATER.bin or DTM_SPI_NOUPDATER.bin)
  - Select Update for programming the new DTM binary image
- No other BlueNRG-1 application binary image can be programmed on a BlueNR-1 device using this BlueNRG GUI, Tools, Updater utility.
- In order to use the STEVAL-IDB007V1 for the BlueNRG-1 SPI network coprocessor configuration, the following hardware changes must be done on such platform: make a short at positions R59, R60, R61, R62 on STEVAL-IDB007V1 platform version 3.

2.2 The BlueNRG graphical user interface

This section describes the main functions of BlueNRG GUI application. You can run this utility by clicking on the BlueNRG GUI icon on the Desktop or under: Start → STMicroelectronics →BlueNRG GUI X.X.X → BlueNRG GUI.

2.2.1 GUI main window

Figure 1: BlueNRG GUI main window



The BlueNRG GUI main window is characterized by different zones. Some of these zones can be resized.

Port and interface selection

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

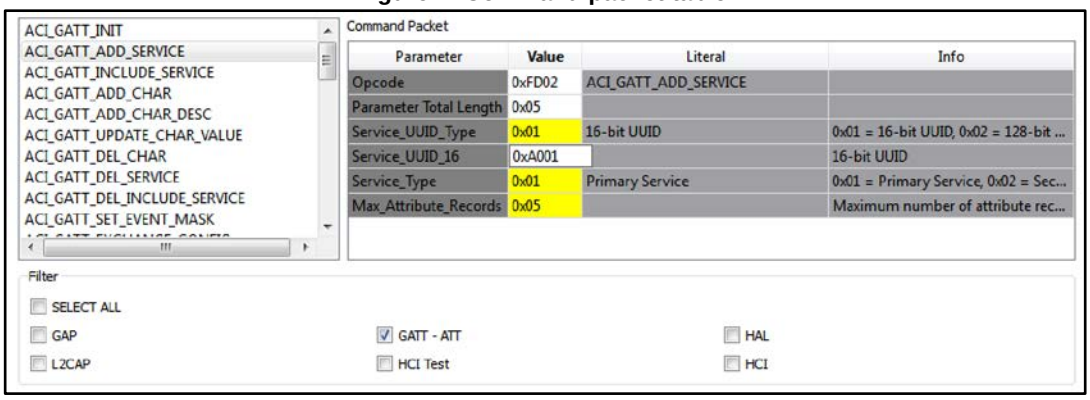
When a COM port is opened the following information are displayed:

- BlueNRG, BlueNRG-MS, BlueNRG-1 HW version
- BlueNRG, BlueNRG-MS FW stack version, BlueNRG-1 FW stack and DTM FW versions
- Motherboard firmware (BlueNRG_VCOM_x_x or USB to Serial) version

HCI commands

The HCI commands tab contains a list of all the available HCI commands. Commands can be filtered by checking/unchecking boxes under the filter section. After clicking on one of the commands, all the packet fields are displayed on the command packet table in the upper-right section of the tab (see *Figure 2: "Command packet table"*).

Figure 2: Command packet table



The command packet table contains four columns:

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

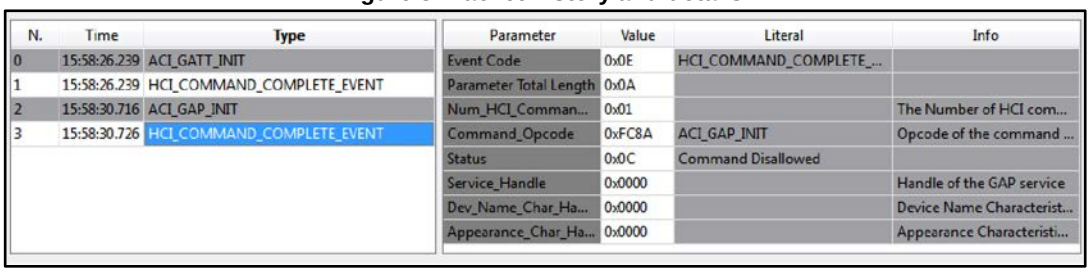
Only the yellow cells of this table can be modified by the user. The Parameter Total Length is fixed or automatically calculated after modifying cell content.

After the fields have been modified (if required) the command can be sent using the Send button.

HCI packet history and details

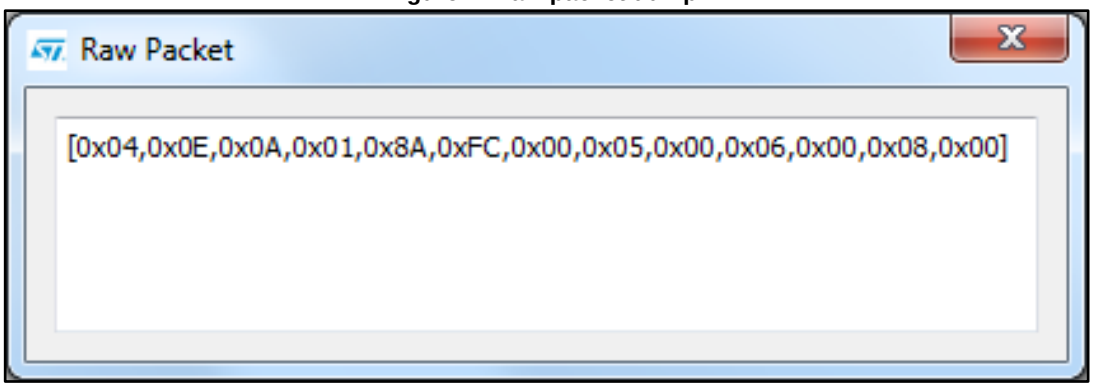
At the bottom of the main window, two tables show packets sent to and received from the BLE controller, as well as other events. The table on the left (sent/received packets) holds a history of all packets (see *Figure 3: "Packet history and details "*). The table on the right (packet details) shows all the details of the selected packet as is done in the command packet table (*Figure 3: "Packet history and details "*).

Figure 3: Packet history and details



Double-clicking on a row of the sent/received packets 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 BLE 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:

1. Save History: it saves the current list of sent commands and received events on a CSV file. It is also possible to save these information on a simple text file.
2. Load History: it loads a list of sent commands and received events, previously stored on a CSV file.
3. Save as Python Script: it allows to store the current list of sent commands and received events as a script file (python format). This script file can be used on GUI Script window, after proper customization (by adding specific code for handling events, parameters), in order to address a user application scenario (refer to *Section 2.2.4: "GUI scripts window"*).

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.

BlueNRG updater

This tool can be used to update the firmware inside the BlueNRG, BlueNRG-MS by using its internal bootloader. BlueNRG_VCOM_x_x.hex firmware must be present on the BlueNRG or BlueNRG-MS motherboard STM32L1 microcontroller and COM port must be open. In order to use this tool these steps must be performed:

1. Go to Tools -> BlueNGR updater
2. Select the correct stack firmware (.img) for BlueNRG, BlueNRG-MS 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, provided the device has been already configured as a network coprocessor with a DTM FW image containing the updater/bootloader (refer to *Section 2.1.2: "BlueNRG-1 network coprocessor"*).

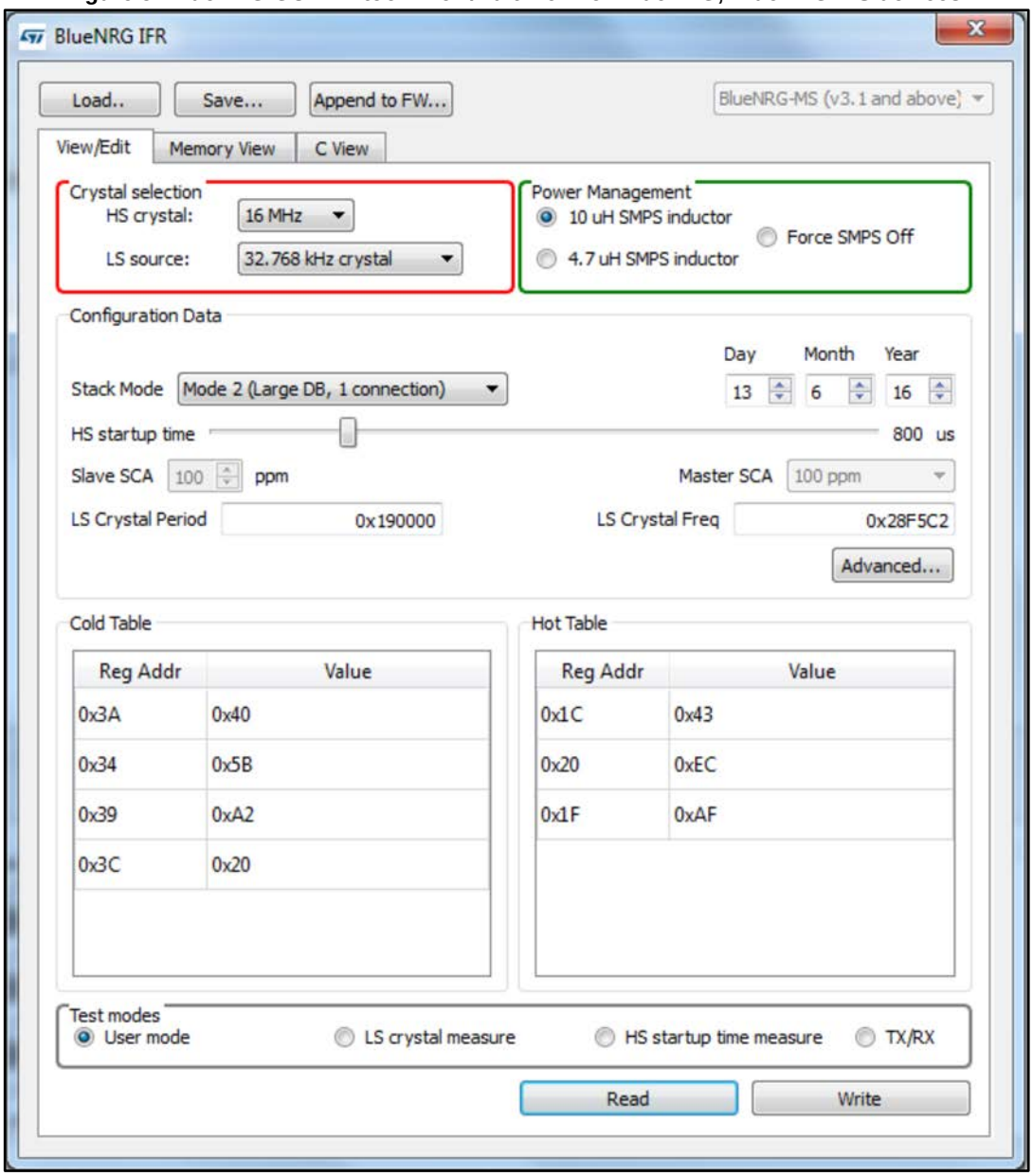
BlueNRG IFR

To preserve the BlueNRG or BlueNRG-MS's flexibility, its firmware uses a table of configurable parameters. This table resides in a sector of the Flash called Information Register (IFR). The BlueNRG GUI IFR tool can read and modify this portion of BlueNRG, BlueNRG-MS Flash. This tool is available in BlueNRG GUI, Tools, BlueNRG IFR... item.

The BlueNRG GUI IFR utility is a tool that allows the customer to define the IFR data in a controlled way. Using this utility is 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 tool: View/Edit view for BlueNRG, BlueNRG-MS devices



In the View/Edit view, the following operations are available:

- Select the high speed (HS) crystal (16 or 32 MHz) and the low speed oscillator source (32 kHz or the internal ring oscillator)
- Set the 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: only on BlueNRG-MS FW stack version > 7.1a, slave/master, simultaneous advertising and scanning, up to 4 connections, small GATT database (RAM2 on during sleep)
- Change HS startup time parameter. This parameter control 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 startup 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 BlueNRG, BlueNRG-MS devices
- Write IFR configuration to BlueNRG, BlueNRG-MS devices

The following general utilities are also available:

- Load button: allows to load a configuration file.
- Save button: allows to save the current parameters into a configuration file.
- Append to FW...: generate a FW stack image with the selected IFR configuration.

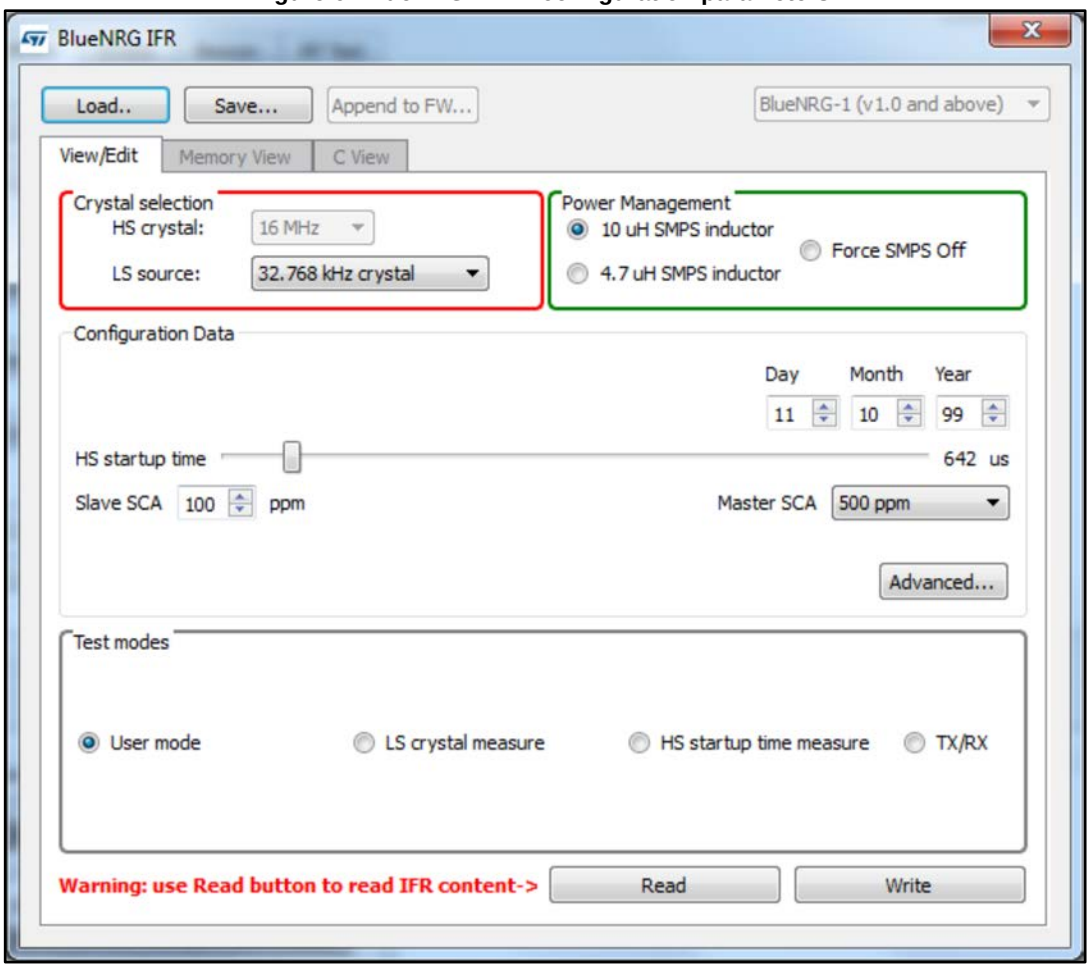
BlueNRG-1 hardware configurations parameters

To preserve BlueNRG-1's flexibility, its firmware uses a table of hardware configurable parameters. The BlueNRG GUI IFR tool can read and modify these hardware configuration parameters. This tool is available in BlueNRG GUI, Tools, BlueNRG IFR... item.

The utility provides the following windows:

- View/Edit view: displays the device hardware configuration parameters regions with related fields and description. The user can modify some of these fields according to his needs.
- Memory and C view s are not applicable on BlueNRG-1 context.

Figure 6: BlueNRG-1 HW configuration parameters



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

- Select the low speed oscillator source (32 kHz or the internal ring oscillator)
- Set the Power Management options (SMPS inductor value or SMPS off configuration)
- Set some test modes for specific tests
- Read hardware configuration parameters content from BlueNRG-1
- Write hardware configuration parameters configuration to BlueNRG-1.

BlueNRG-1 DTM UART HS crystal (16 or 32MHz) is selected at compile time and it cannot be changed through BlueNRG GUI IFR.

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 allows to Flash firmware to the STM32L1 microcontroller on the BlueNRG, BlueNRG-MS motherboards without a JTAG/SWD programmer. This utility uses a bootloader (DFU) that has been programmed 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[a].

On BlueNRG-1 kit development platform, this utility allows to upgrade the USB-to-Serial firmware when needed. User must follow these steps:

- Activate the DFU application manually, acting as follows:
  a. Press and hold the RESET button.
  b. Plug the USB cable to the board.
  c. Release the RESET button.
  d. The red led DL2 blinks to confirm the DFU application is running.
  e. On BlueNRG GUI PC application, select Tools, Flash Motherboard FW...
  f. Press the Apply button of the window.
  g. Select the firmware USB_to_SERIAL.hex available on Firmware/BlueNRG-1 folder and press Open button.
  h. Wait for the end of the upgrade operation.

OTA bootloader

OTA bootloader is a tool that allows to Flash new firmware to the STM32L1 of a remote BlueNRG, BlueNRG-MS motherboard via Bluetooth low energy technology. It also allows to Flash new firmware to a remote BlueNRG-1 device via Bluetooth low energy technology. Refer to the dedicated application notes available on related web pages for detailed 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 and DTM FW version + mode (SPI or UART) on BlueNRG-1 platform, and hardware and firmware version from the BlueNRG, BlueNRG-MS and BlueNRG-1 devices.

Settings

This tool allows to configure the firmware stack version to be used from the GUI (when no device is actually connected to a PC USB port). Further, it allows to configure the GUI serial baud rate (valid only for communication over serial UART and not through USB Virtual COM).

In order to use this function.

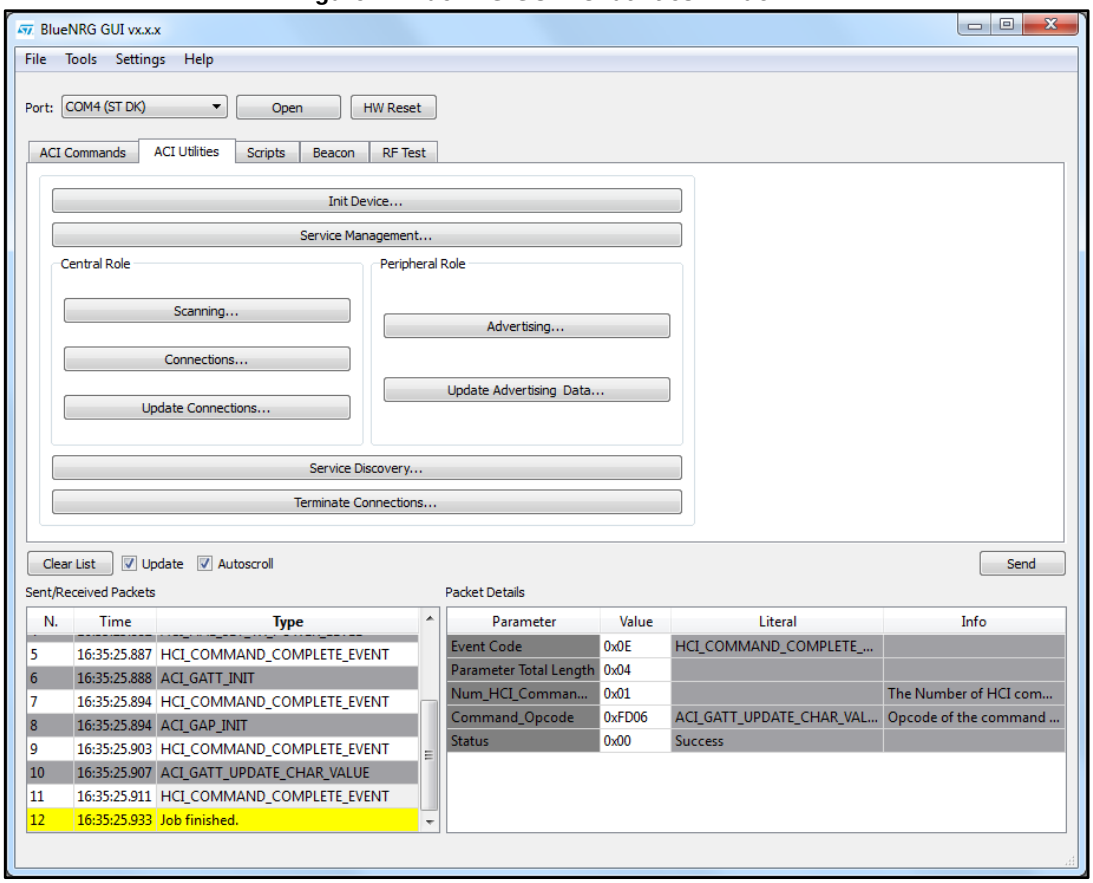
1. Go to Settings --> select FW 7.2 or 7.1 for BlueNRG-MS device, FW 6.4 for BlueNRG device and FW 1.0 for BlueNRG-1 device
2. Go to Settings --> select Set Baud Rate… and choose the value.

2.2.3 GUI ACI utilities window

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

[a] 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())

Figure 7: BlueNRG GUI ACI utilities window



Central and Peripheral roles are supported with the BLE operations described in *Table 1: "GUI ACI utilities window: available general operations"*, *Table 2: "GUI ACI utilities window: available central operations"* and *Table 3: "GUI ACI utilities window: available peripheral operations"*.

Table 1: GUI ACI utilities window: available general operations

Operation	Associated actions	Notes
Init Device…	Allows to initialize a device by selecting: - Role - Stack Mode (1,2,3,4) - Address type (Public, Random) and value - Tx power level - Power mode - Device Name	

Operation	Associated actions	Notes
Service Management…	Allows to add 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 to add a characteristic and related descriptors 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 showed on a specific view.
Service Discovery ..	Allows to discover 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's possible to enable the Notification/Indication.
Terminate Connection...	Allows to terminate the available connections	

Table 2: GUI ACI utilities window: available central operations

Operation	Associated actions	Notes
Scanning	Allows to put 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	

Operation	Associated actions	Notes
Connections	Allows to connect to a peer device by: - Searching for devices in Advertising - Select the device to which to connect - Select 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 to update the connection parameters of available connections by: - Selecting the specific connection to be updated - Set the new connection parameters - Connection interval (min and max) - Latency - Supervision timeout - Connection event length (min and max)	

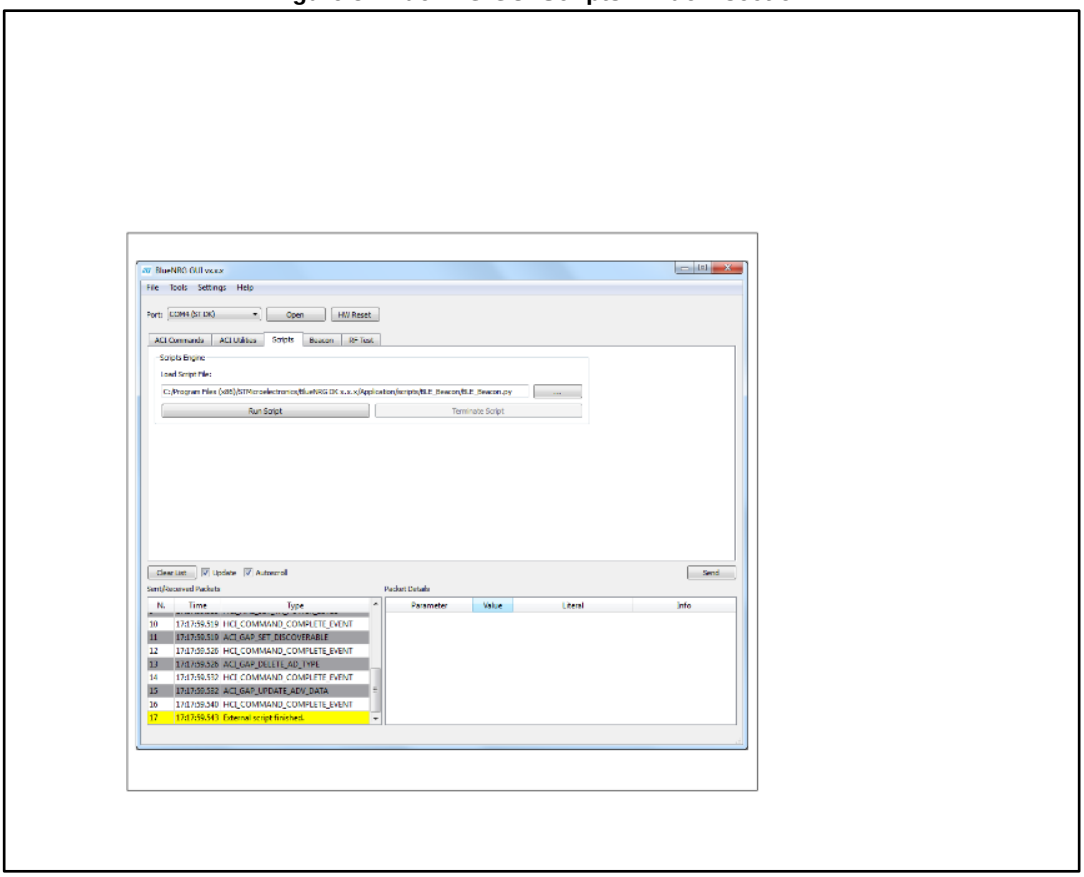
Table 3: GUI ACI utilities window: available peripheral operations

Operation	Associated actions	Notes
Advertising	Allows to put 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: - Allow scan request from any, allow connect request from any - Allow scan request from white list only, allow connect request from any - Allow scan request from any, allow connect request from white list only	
Update Advertising Data	It allows to update the advertising data; It allows to set the scan response data; It allows to update the location UUID, major and minor number defined on the Beacon window	

2.2.4 GUI scripts window

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

Figure 8: BlueNRG GUI Scripts window section



Moreover, the script engine supports other utility commands:

Table 4: GUI Scripts window: utility commands

Command name	Parameters	Description
ERROR	User message	Raises an exception with a user-defined debug message
CLEAR_QUEUE	None	It removes all the pending events collected within the internal event queue.
GET_CHAR	None	It allows to enter a specific char as input (as the C get_char() API)
GET_ALL_COM_PORT	None	It returns the list of all COM ports (ST DK kits and not)
GET_ALL_ST_DK_COM_PORT	None	It returns the list of all ST DK Kits COM ports sorted by COM port number
GET_FILE	None	It allows to select a specific file as input
GET_NAME	None	Returns the device name within an advertising packet
GET_VALUE	Array of bytes	Convert the array of bytes to an integer value. Example: X= [0x33,0x22] GET_VALUE(X) = 0x2233
GET_LIST	Integer, Number of bytes	Convert the integer value to an array of Number of bytes. Example: X= 0x2233 GET_LIST(X, 2)= [0x33,0x22]
GET_STACK_VERSION	None	Return the device information (HW version and FW version) as (hw, fw)

Command name	Parameters	Description
GET_RAND_KEY	None	Returns a random number between 0 and 999999
HW_BOOTLOADER	None	Hardware bootloader activation
HW_RESET	None	HW reset
INFO	String to be displayed	Open a message window and show the input parameter. Script is blocked until user presses OK button
INSERT_PASS_KEY	None	Allows to enter a pass key value used for the security pass key method
IS_BlueNRG	None	Return TRUE if the device is a BlueNRG, FALSE otherwise
IS_BlueNRG_MS	None	Return TRUE if the device is a BlueNRG-MS, FALSE otherwise
IS_BlueNRG_1	None	Return TRUE if the device is a BlueNRG-1, FALSE otherwise
PRINT	String	Print utility: it displays information on GUI Sent/Received Packets
RESET	None	SW reset
SLEEP	time	It sleeps for “time” seconds
SET_MODE	Mode	Set stack mode (1,2,3,4). Mode 4 is supported only on BlueNRG-MS from FW stack version 7.1b. Only for BlueNRG, BlueNRG-MS devices.
SET_PUBLIC_ADDRESS	Public address	Set public address (optional)
SENSORDEMO_GET_TEMPERATURE	None	It allows to get the temperature value from the ACI_ATT_READ_RESP_EVENT (only for the SensorDemo_Central script)
SENSORDEMO_GET_ACCELERATION	None	It allows to get the acceleration values (x,y,z) from the ACI_GATT_NOTIFICATION_EVENT (only for the SensorDemo_Central script)
TIME	None	It 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, BlueNRG-MS, BlueNRG-1 device as a peripheral using a simple python script:

```
#Reset device

HW_RESET()

#Init GATT

ACI_GATT_INIT()
#Init GAP as central device

ACI_GAP_INIT(Role=CENTRAL)
```

When a script is calling 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 an event with 'Event Code' parameter equal to event_code. If no event_code is indicated, the macro-command waits any event. Optional filtering parameters allow to define additional filters 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 to set the event timeout. If no timeout is set, the macro-command waits 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 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 to clear the internal queue 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 related event parameter types

event_code	event par. type	event parameter type value
HCI_LE_META_EVENT	Subevent_ Code	HCI_LE_CONNECTION_COMPLETE_EVENT
		HCI_LE_ADVERTISING_REPORT_EVENT
		HCI_LE_CONNECTION_UPDATE_COMPLETE_EVENT
		HCI_LE_READ_REMOTE_USED_FEATURES_COMPLETE_ EVENT
		HCI_LE_LONG_TERM_KEY_REQUEST_EVENT
HCI_VENDOR_EVENT	Ecode	ACI_BLUE_INITIALIZED_EVENT
		ACI_BLUE_EVENTS_LOST_EVENT

event_code	event par. type	event parameter type value
		ACI_BLUE_CRASH_INFO_EVENT
		ACI_GAP_LIMITED_DISCOVERABLE_EVENT
		ACI_GAP_PAIRING_COMPLETE_EVENT
		ACI_GAP_PASS_KEY_REQ_EVENT
		ACI_GAP_AUTHORIZATION_REQ_EVENT
		ACI_GAP_SLAVE_SECURITY_INITIATED_EVENT
		ACI_GAP_BOND_LOST_EVENT
		ACI_GAP_DEVICE_FOUND_EVENT
		ACI_GAP_PROC_COMPLETE_EVENT
		ACI_GAP_RECONNECTION_ADDRESS_EVENT
		ACI_GAP_ADDR_NOT_RESOLVED_EVENT
		ACI_L2CAP_CONNECTION_UPDATE_RESP_EVENT
		ACI_L2CAP_PROC_TIMEOUT_EVENT
		ACI_L2CAP_CONNECTION_UPDATE_REQ_EVENT
		ACI_GATT_ATTRIBUTE_MODIFIED_EVENT
		ACI_GATT_PROC_TIMEOUT_EVENT
		ACI_ATT_EXCHANGE_MTU_RESP_EVENT
		ACI_ATT_FIND_INFO_RESP_EVENT
		ACI_ATT_FIND_BY_TYPE_VALUE_RESP_EVENT
		ACI_ATT_READ_BY_TYPE_RESP_EVENT
		ACI_ATT_READ_RESP_EVENT
		ACI_ATT_READ_BLOB_RESP_EVENT
		ACI_ATT_READ_MULTIPLE_RESP_EVENT
		ACI_ATT_READ_BY_GROUP_TYPE_RESP_EVENT
		ACI_ATT_WRITE_RESP_EVENT
		ACI_ATT_PREPARE_WRITE_RESP_EVENT
		ACI_ATT_EXEC_WRITE_RESP_EVENT
		ACI_GATT_INDICATION_EVENT
		ACI_GATT_NOTIFICATION_EVENT

event_code	event par. type	event parameter type value
		ACI_GATT_PROC_COMPLETE_EVENT
		ACI_GATT_ERROR_RESP_EVENT
		ACI_GATT_DISC_READ_CHAR_BY_UUID _RESP_EVENT
		ACI_GATT_WRITE_PERMIT_REQ_EVENT
		ACI_GATT_READ_PERMIT_REQ_EVENT
		ACI_GATT_READ_MULTI_PERMIT_REQ_ EVENT
		ACI_GATT_TX_POOL_AVAILABLE_EVENT
		ACI_GATT_SERVER_CONFIRMATION_EV ENT
HCI_DISCONNECTION_COMPLETE_EVENT		
HCI_ENCRYPTION_CHANGE_ EVENT		
HCI_READ_REMOTE_VERSION_INFORMATION_ COMPLETE_EVENT		
HCI_COMMAND_COMPLETE_ EVENT		
HCI_COMMAND_STATUS_EVENT		
HCI_HARDWARE_ERROR_EVENT		
HCI_NUMBER_OF_COMPLET ED_PACKETS_EVENT		
HCI_DATA_BUFFER_OVERFLOW_EVENT		
HCI_ENCRYPTION_KEY_REF RESH_COMPLETE_EVENT		

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
```

Example 3:

```
#Wait HCI_VENDOR_EVENT event_code evt = WAIT_EVENT(HCI_VENDOR_EVENT)
#Usingevt.get_param('Ecode').val it's possible to identify the specific
HCI_VENDOR_EVENT parameter type value
evtCode= evt.get_param('Ecode').val
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 no event occurs within the selected timeout, an exception is raised
WAIT_EVENT(HCI_VENDOR_EVENT, timeout=30,
Ecode=ACI_GATT_PROC_COMPLETE_EVENT)
```

If no timeout parameter is specified, it waits 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)
```

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 …
```

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.

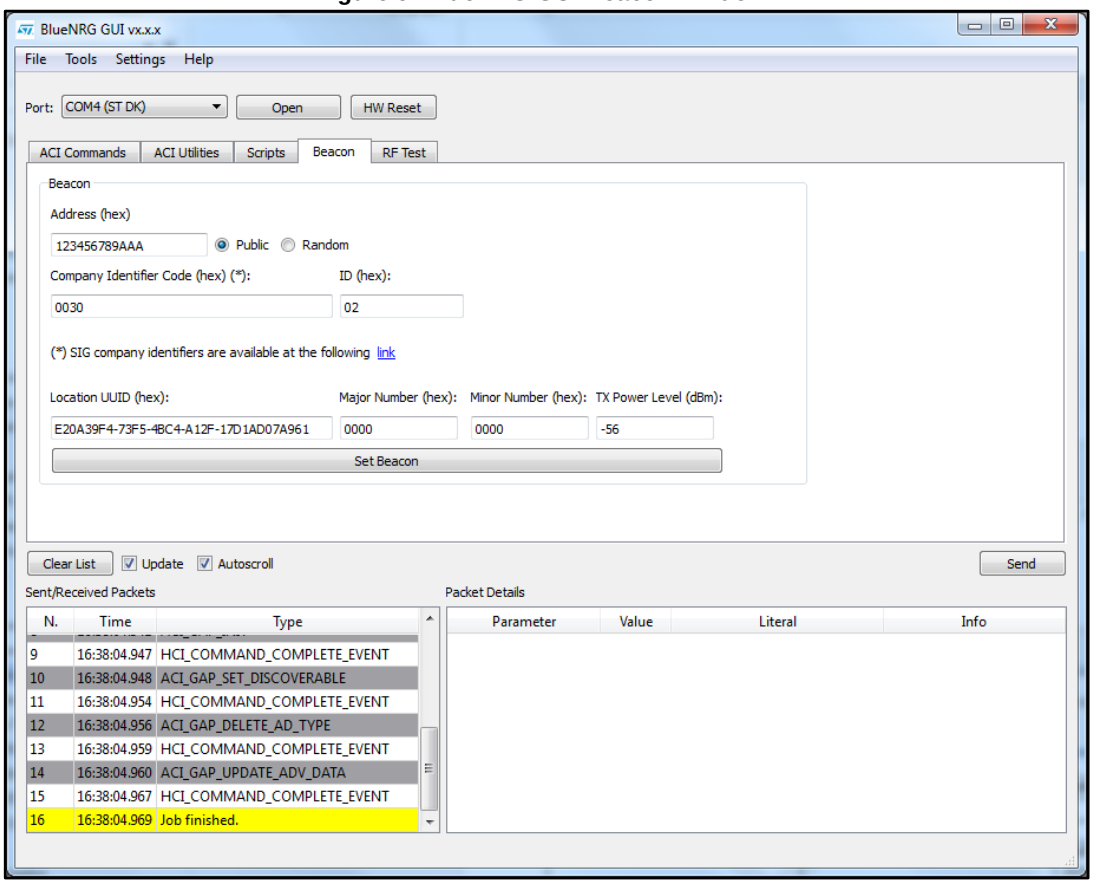
In the BlueNRG GUI SW package and future versions, some reference scripts are available in the GUI/scripts folder.

It is worthy of note that in order to write and use the scripts, the user is required to have some knowledge of the Python language (Python 2.7.6), and a good understanding of the BLE stack ACI commands and related events.

2.2.5 GUI Beacon window

The BlueNRG GUI Beacon window provides some tabs allowing configuration of a BlueNRG, BlueNRG-MS or BlueNRG-1 device as a BLE Beacon device which transmits advertising packets with specific manufacturer data.

Figure 9: BlueNRG GUI Beacon window



The user can configure the following advertising data fields for the BLE 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

Data field	Description	Notes
Major number	Identifier fora group of beacons	Used to group a related set of beacons
Minor number	Identifier fora single beacon	Used to identify a single beacon
TxPower Level	2's complement ofthe Tx power	Used to establish how far you are from device

To configure the selected platform as a BLE beacon device, click on “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 BLE RF channel
2. Perform a BLE Packer Error Rate (PER) tests using BLE Direct Test Mode (DTM) commands

Start/Stop a Tone

For starting a tone on a specific RF BLE Channel performs these steps:

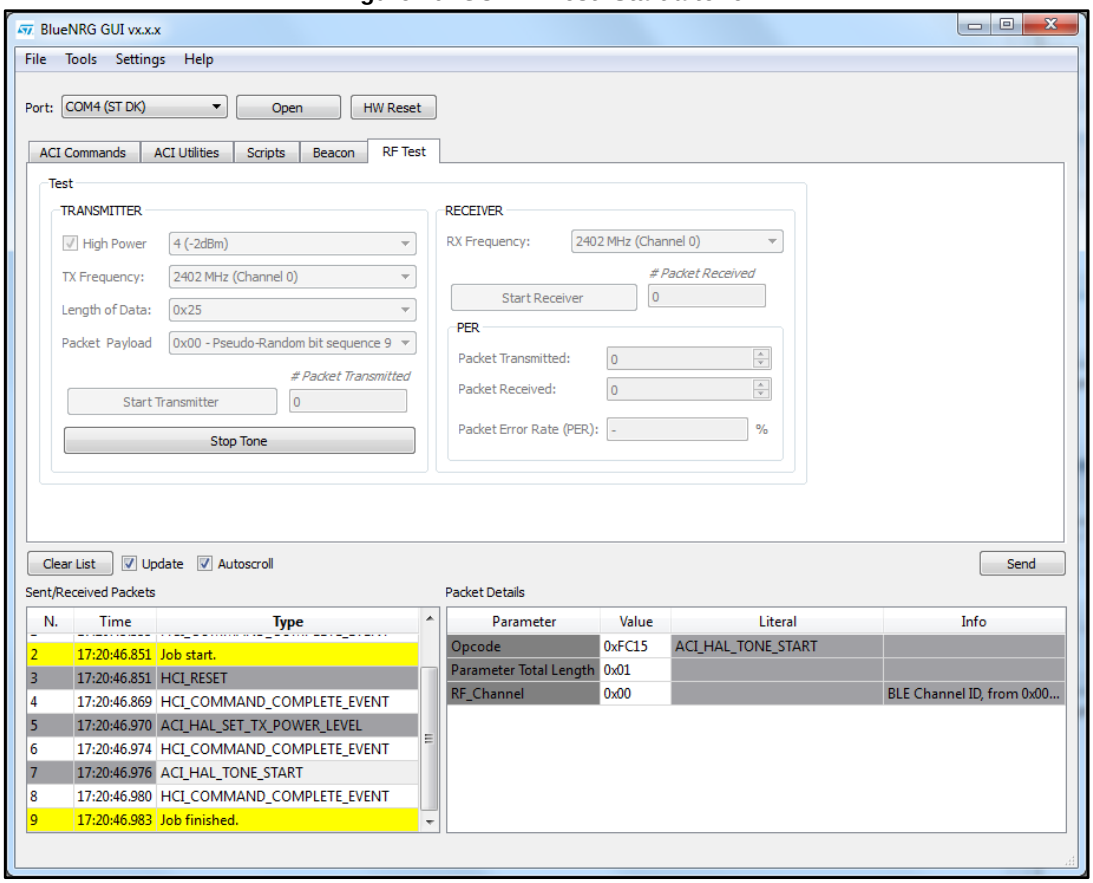
1. Connect a BlueNRG or BlueNRG-MS or BlueNRG-1 platform to PC USB port
2. Launch an instance of BlueNRG GUI
3. Open related COM port
4. Go to RF Test window and in the TRANSMITTER section:
   - Set BLE channel by TX Frequency combo box
   - Set TX power in the related combo box
   - Click on "Start Tone" button

For stopping a tone on a specific RF BLE Channel performs these steps:

1. Go to RF Test window and in the TRANSMITTER section:

- 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 provides RF Test that allows, using the BLE Direct Test Modes commands, to target a packet error rate test scenario.

Two sections are available:

1. TRANSMITTER section for transmitting reference packets at a fixed interval
2. RECEIVER section for receiving reference packets at a fixed interval

TRANSMITTER section

This section permits to 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

Clicking on "Start Transmitter" button, test reference packets will be sent at a fixed interval.

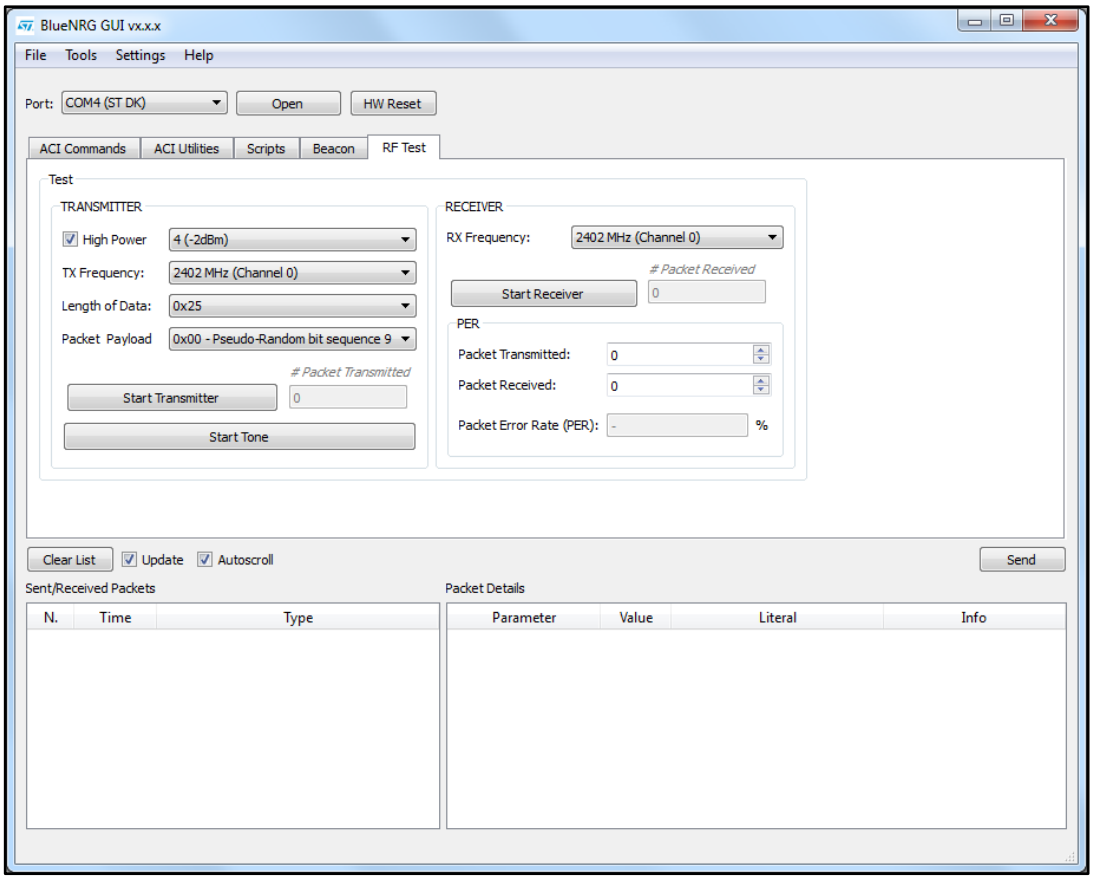
RECEIVER section

This section permits to set the following items:

- The Frequency of Receiver

Clicking on "Receiver Test" button, test reference packets will be received at 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 standard BLE 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 or or BlueNRG-MS or BlueNRG-1) to PC USB ports
2. Open two instances of BlueNRG GUI on both devices (called, respectively, TX and RX devices)
3. In each instance of BlueNRG GUI, Open COM Port related to TX/RX device
4. Ensure that 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
   - 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
- 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

a. 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

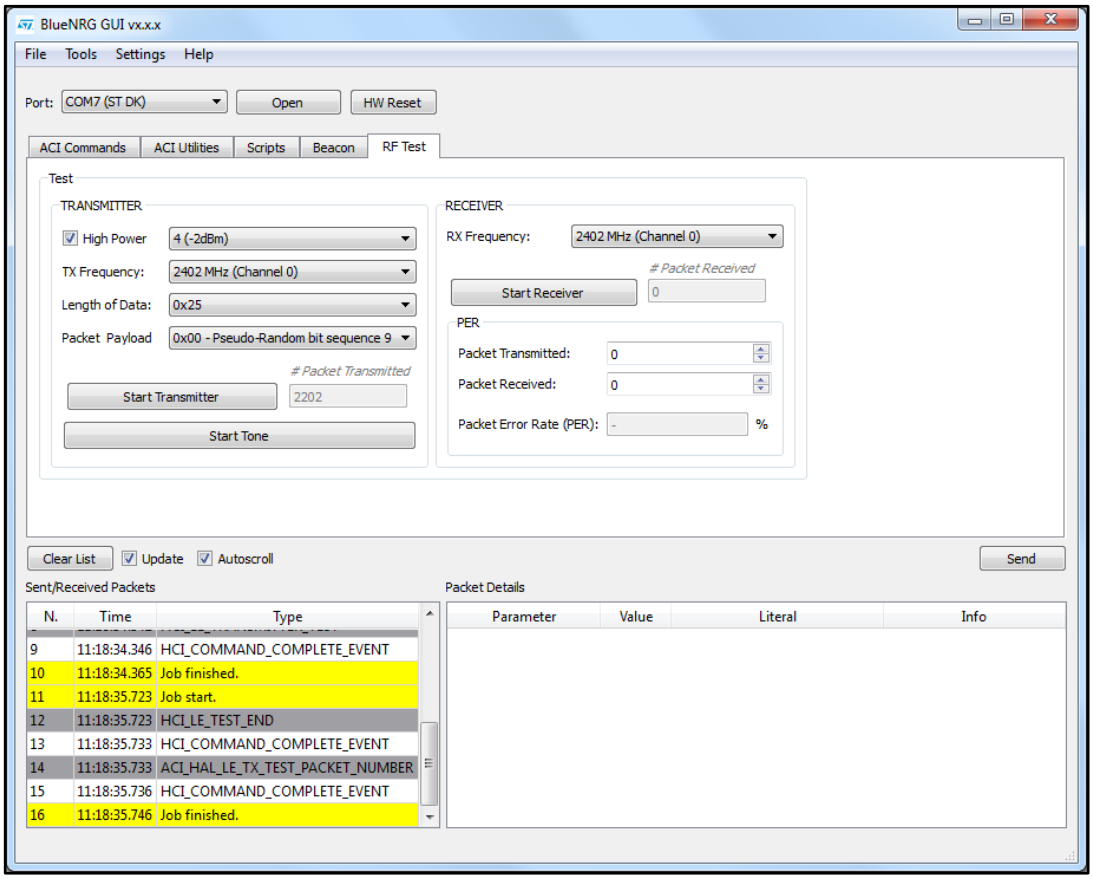
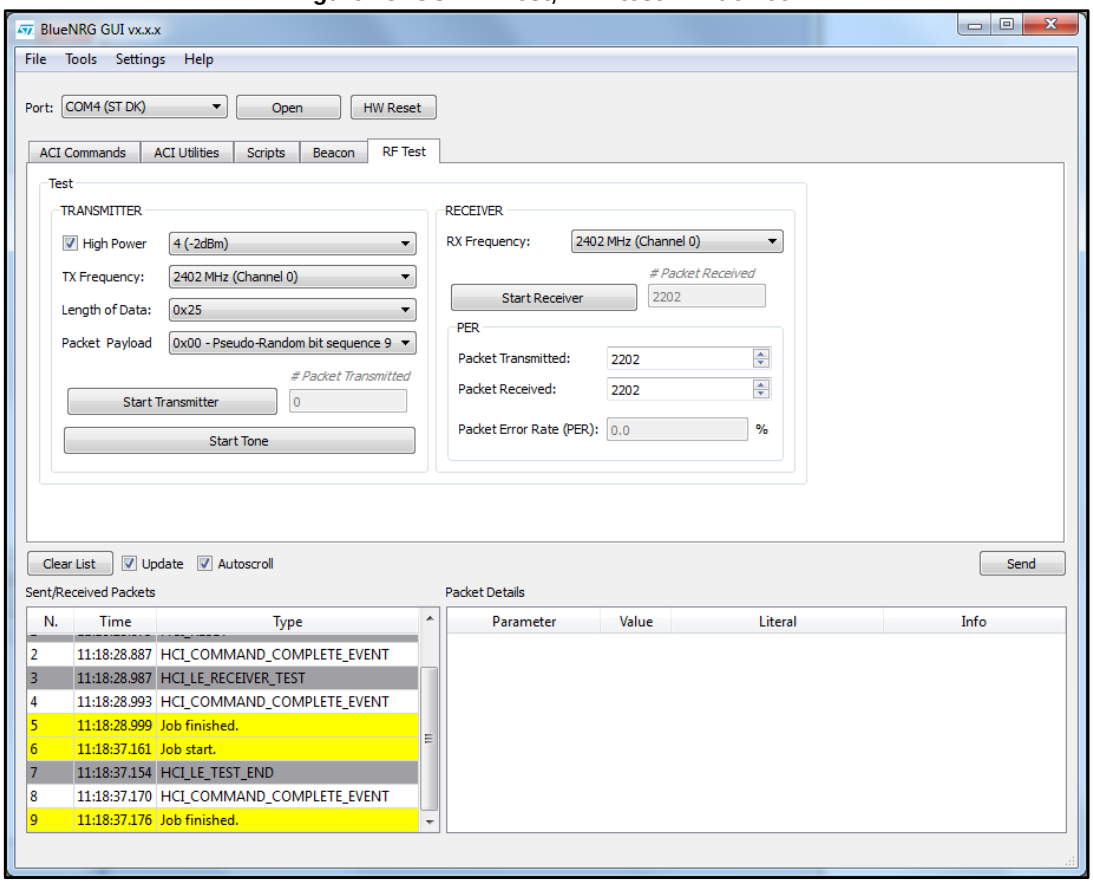


Figure 13: GUI RF Test, PER test: RX device



3 Script launcher

The script launcher is a standalone utility allowing to run a script without the need to use the BlueNRG GUI script engine tool.

The script launcher utility (BlueNRG_Script_Launcher.exe) is included on BlueNRG GUI software package within the folder Application.

3.1 Requirements

In order to use the Script launcher utility on a specific device, the related platform must be connected to a PC USB port and loaded with the same prebuilt binary file used for the BlueNRG GUI, as described on *Section 2.1 Requirements* (BlueNRG_VCOM_x_x.hex for BlueNRG, BlueNRG-MS platforms and specific DTM binary file for the BlueNRG-1 platform).

3.2 Script launcher utility options

In order to use the script launcher utility on a specific device, a Windows DOS shell must be opened and user has to launch the related BlueNRG_Script_Launcher.exe (type –h for getting the list of all supported options):

C:\Program Files (x86)\STMicroelectronics\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:
- -s SCRIPT_FILE, --script SCRIPT_FILE set script file name
- -z, --save save log in a file (*.csv, *.txt)

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

C:\Program Files (x86)\STMicroelectronics\BlueNRG GUI GUI x.x.x\Application BlueNRG_Script_Launcher.exe -g

List of available port COM:

COM1

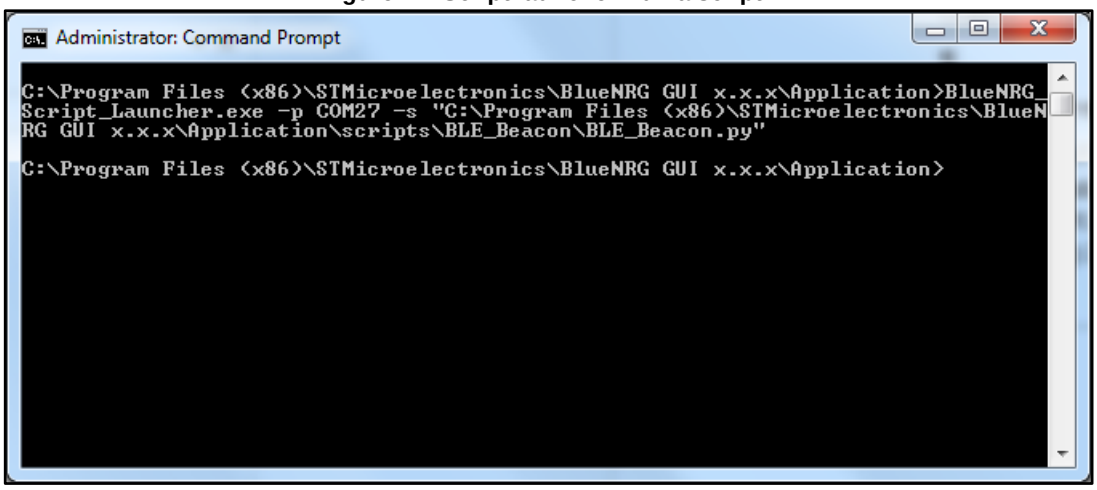
COM27 (ST DK)

Options for running a script on a connected device

C:\Program Files (x86)\STMicroelectronics\BlueNRG GUI x.x.x\Application>BlueNRG_Script_Launcher.exe -p COM27 –s C:\Program Files (x86)\STMicroelectronics\BlueNRG GUI x.x.x\Application\scripts\BLE_Beacon\BLE_Beacon.py

COM27 is the platform Virtual COM port.

Figure 14: Script launcher: run a script

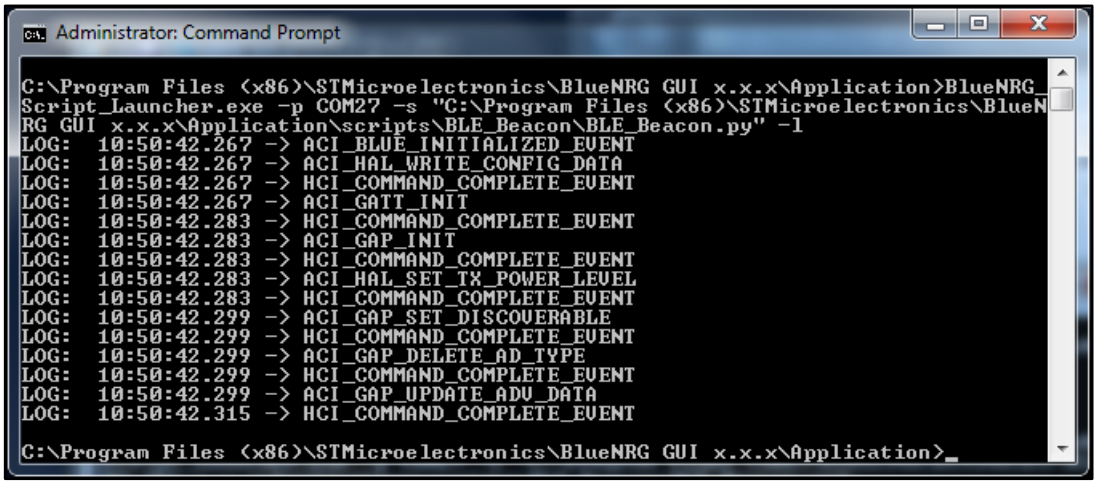


Options for running a script on a connected device with log data

C:\Program Files (x86)\STMicroelectronics\BlueNRG GUI x.x.x\Application>BlueNRG_Script_Launcher.exe -p COM27 –s C:\Program Files (x86)\STMicroelectronics\BlueNRG GUI x.x.x\Application\scripts\BLE_Beacon\BLE_Beacon.py -l

COM27 is the platform Virtual COM port.

Figure 15: Script launcher: run a script with log data



4 References

Table 8: References information

What	Where	Description
STSW-BNRGUI	www.st.com/bluenrg-1 or *www.st.com/bluenrg-ms*, Tools and Software section	BlueNRG GUI SW package
STSW-BLUENRG-DK	*www.st.com/bluenrg-ms*, Evaluation tools section	BlueNRG DK SW package for BlueNRG, BlueNRG-MS kits
STSW-BLUENRG1-DK	*www.st.com/bluenrg-1*, Evaluation tools section	BlueNRG-1 DK SW package for BlueNRG-1 kits

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
SW	Software
USB	Universal serial bus

6 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-July-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"*

IMPORTANT NOTICE – PLEASE READ CAREFULLY

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

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

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

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

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

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