
STM32CubeMonitor-RF software tool for wireless performance measurements

Introduction

STM32CubeMonitor-RF (STM32CubeMonRF) is a software tool, which helps the designers to test their products based on STMicroelectronics STM32 wireless microcontrollers.

The tool performs the following operations:

- It sends and receives test packets to check the efficiency of radio frequency boards and compute packet error rate (PER) on BLE and 802.15.4 technologies.
- It sends commands to a Bluetooth® low energy (BLE) controller for standardized tests.
- It sends and receives BLE commands for fast application prototyping.
- It configures a variety of beacons via BLE commands.
- It transfers data over-the-air (OTA) from one device to another, in order to configure or program a remote device without wired connection.
- It sends commands to an OpenThread device for application prototyping.



Contents

1	Getting started	8
1.1	Download and setup	8
1.2	Welcome screen	9
1.3	Main screen	10
1.3.1	Menu bar	10
1.3.2	Connection bar	11
1.3.3	Panels	11
1.3.4	Log area	13
2	Connection to target	14
2.1	Use case description and definitions	14
2.2	VCOM / UART connection	15
2.2.1	VCOM connection	15
2.2.2	UART connection	17
2.2.3	VCP device	17
2.3	Opening COM	18
3	BLE mode	20
3.1	Presentation	20
3.1.1	Panels	20
3.2	ACI Commands panel	21
3.2.1	How to send an ACI command	21
3.2.2	Search function	22
3.2.3	Filter usage	22
3.2.4	How to fill parameters. Fixed field / editable field	22
3.2.5	Log functionalities	23
3.3	RF tests panel	26
3.3.1	Test mode transmitter (TX)	27
3.3.2	Test mode receiver (RX)	30
3.3.3	PER	33
3.4	Scripts	38
3.4.1	Launching scripts	39
3.4.2	Script recording	40

3.4.3	Scripts modification	41
3.4.4	Script report	41
3.4.5	List of scripts commands	43
3.5	OTA transfer	46
3.5.1	OTA presentation	46
3.5.2	OTA procedure	47
3.5.3	Use the tool to perform OTA update	48
3.6	Beacon	51
3.6.1	Beacon presentation	51
3.6.2	Beacon configuration methods:	52
3.6.3	Configuration of beacon with STM32CubeMonitor-RF	54
3.7	ACI utilities	60
3.7.1	Remote services discovering	60
3.7.2	Advertising	67
4	OpenThread mode	69
4.1	Presentation	69
4.1.1	Panel	69
4.2	Commands tab	70
4.3	OpenThread scripts tab	72
4.3.1	OpenThread script example	74
4.3.2	List of script commands	74
5	802.15.4 RF test mode	76
5.1	Presentation	76
5.2	Transmitter (TX) test mode	77
5.3	Receiver test (RX) mode	77
5.4	Packet error rate (PER) mode	78
5.4.1	Connecting the additional device to play the role of packet generator (tester).	79
5.4.2	Configure the parameters of tester.	80
5.4.3	Configure the parameters of device under test (DUT).	81
5.4.4	Configure the measurement.	82
	Appendix A Beacon configuration format	85
	Revision history	86

List of tables

Table 1. OTA loader address table..... 47

Table 2. Search filtering 49

Table 3. Measurement setting 83

Table 4. Beacon configuration format..... 85

Table 5. Document revision history 86



List of figures

Figure 1.	Welcome screen	9
Figure 2.	Main screen	10
Figure 3.	Menu bar	10
Figure 4.	Connection bar	11
Figure 5.	ACI Commands panel	12
Figure 6.	Log area	13
Figure 7.	Typical connection with a Nucleo board	14
Figure 8.	Connection with a remote device	14
Figure 9.	Connection with a second device	15
Figure 10.	VCOM connection BLE	15
Figure 11.	VCOM connection Thread	16
Figure 12.	UART connection	17
Figure 13.	VCP connection	18
Figure 14.	Opening COM	18
Figure 15.	Successful COM	19
Figure 16.	ACI Commands panel	20
Figure 17.	How to send an ACI command	21
Figure 18.	Search button	22
Figure 19.	Fixed parameter	22
Figure 20.	Editable parameter	23
Figure 21.	Predefined values	23
Figure 22.	Help details	23
Figure 23.	Log functionalities	24
Figure 24.	More button	24
Figure 25.	Message details	25
Figure 26.	Purple error messages	25
Figure 27.	Gray second board messages	26
Figure 28.	Test mode selection	26
Figure 29.	Change test mode	27
Figure 30.	Select test mode	27
Figure 31.	Test mode transmitter	28
Figure 32.	Transmitting message	29
Figure 33.	Transmitted packets count	29
Figure 34.	Test mode receiver	30
Figure 35.	RSSI measurement	31
Figure 36.	RSSI measurement graph	32
Figure 37.	RF RSSI measurement big display	33
Figure 38.	PER definition	33
Figure 39.	PER tester connection	34
Figure 40.	PER tester connected	35
Figure 41.	PER tester configuration	35
Figure 42.	DUT configuration	36
Figure 43.	PER test parameters	37
Figure 44.	PER and RSSI measurement graph	38
Figure 45.	Launching scripts	39
Figure 46.	Script execution	40
Figure 47.	Script recording buttons	40
Figure 48.	Sample script	41

Figure 49.	Script report	42
Figure 50.	Script verdict	42
Figure 51.	Script pause	43
Figure 52.	Example	44
Figure 53.	Loop simple example	44
Figure 54.	Loop second simple example	44
Figure 55.	Loop decrement	45
Figure 56.	Loop specific increment	45
Figure 57.	Nested loop	45
Figure 58.	Loop script verdict example	46
Figure 59.	Loop script verdict display	46
Figure 60.	Script loop error	46
Figure 61.	Search procedure	48
Figure 62.	Scanning	49
Figure 63.	No device found	49
Figure 64.	Device found	50
Figure 65.	Select the device and parameters	50
Figure 66.	Configuring in OTA	51
Figure 67.	Progress bar	51
Figure 68.	Beacon presentation	52
Figure 69.	Beacon usage	52
Figure 70.	Online beacon	53
Figure 71.	Offline beacon	53
Figure 72.	Beacon over-the-air configuration	54
Figure 73.	Selecting the beacon mode	54
Figure 74.	Common parameters	55
Figure 75.	iBeacon parameters	55
Figure 76.	Eddystone UID	56
Figure 77.	Eddystone URL	56
Figure 78.	Online mode transfer configuration	57
Figure 79.	Offline mode transfer configuration	57
Figure 80.	OTA mode transfer configuration scan	58
Figure 81.	OTA mode transfer configuration select device	58
Figure 82.	OTA mode transfer configuration transmission	59
Figure 83.	ACI Utilities panel	60
Figure 84.	Select checkbox	60
Figure 85.	Scan parameters	61
Figure 86.	Scanning	62
Figure 87.	Select device	62
Figure 88.	Back	62
Figure 89.	Init	62
Figure 90.	Connecting	63
Figure 91.	Connection error	63
Figure 92.	Connected icon	63
Figure 93.	Services list	63
Figure 94.	Characteristics list	64
Figure 95.	Read value	64
Figure 96.	Write value	65
Figure 97.	Indicate value changed	65
Figure 98.	Notify value changed	66
Figure 99.	Notifying	66
Figure 100.	Advertising parameters	67

Figure 101. Advertising	68
Figure 102. Connected	68
Figure 103. OpenThread - Command tab	69
Figure 104. OpenThread - Script tab	70
Figure 105. OpenThread common bottom area	70
Figure 106. OpenThread command tab - top area	71
Figure 107. Script buttons	71
Figure 108. Command list	72
Figure 109. Command details	72
Figure 110. Scripts tab	73
Figure 111. Sample script	74
Figure 112. Test mode selection	76
Figure 113. Transmitter test mode	77
Figure 114. Receiver test mode	78
Figure 115. Packet tester connection	79
Figure 116. PER tester configuration	80
Figure 117. DUT configuration	81
Figure 118. PER test parameters	82
Figure 119. Standard display	83
Figure 120. Chart display	83
Figure 121. Large PER display	84
Figure 122. Large RSSI display	84
Figure 123. Large LQI display	84

1 Getting started

STM32CubeMonitor-RF supports STM32WBxx microcontrollers based on the Arm^{®(a)} Cortex[®]-M processor.

The logo for Arm, consisting of the word "arm" in a bold, lowercase, sans-serif font.

1.1 Download and setup

STM32CubeMonitor-RF is used with Windows[®], Linux[®] and Mac[®] computers.

The information to install the application is described in the release note, which describes the compatibilities and new features available in the tool.

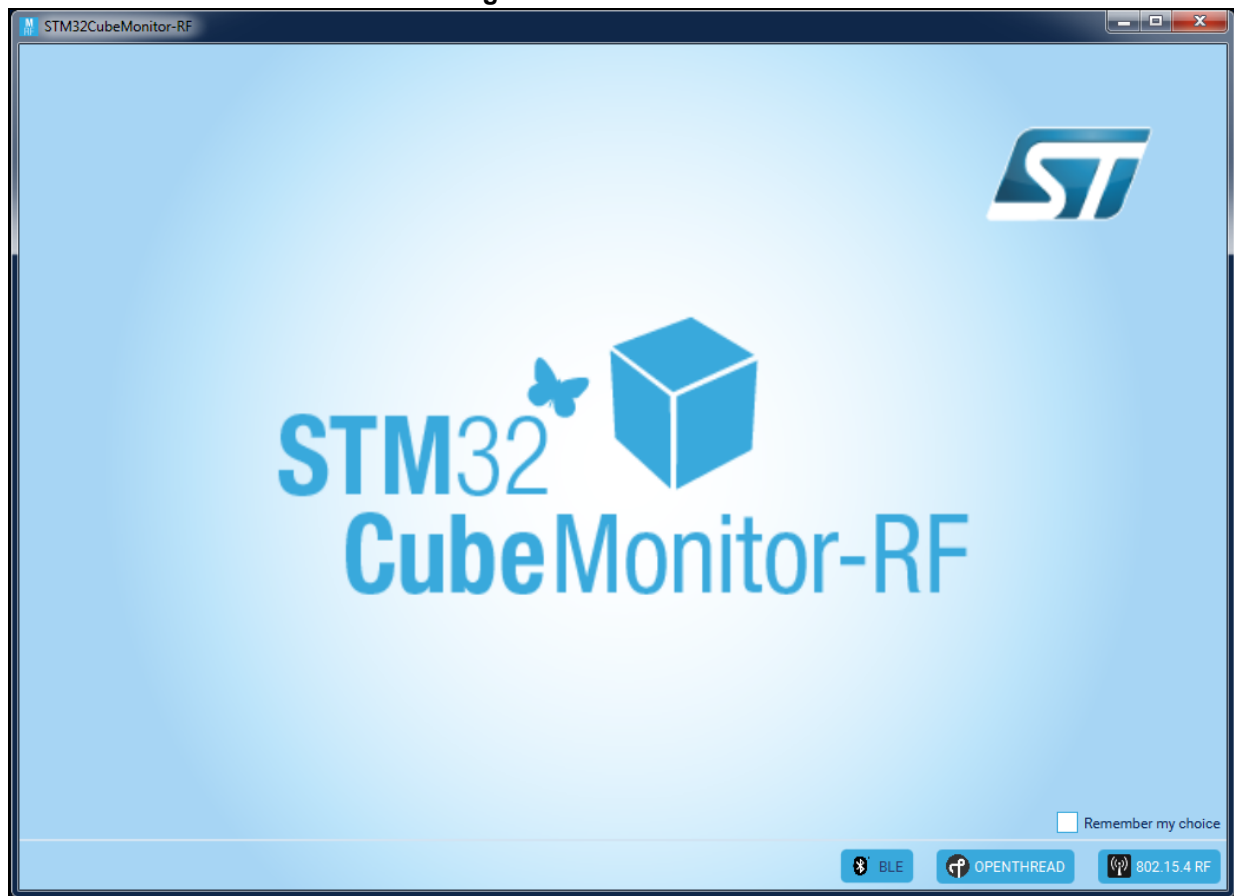
This user manual is applicable to STM32CubeMonitor-RF version 2.4.0 and later.

Refer to the STM32CubeMonRF release note (RN0104) to install and configure the application.

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

1.2 Welcome screen

Figure 1. Welcome screen



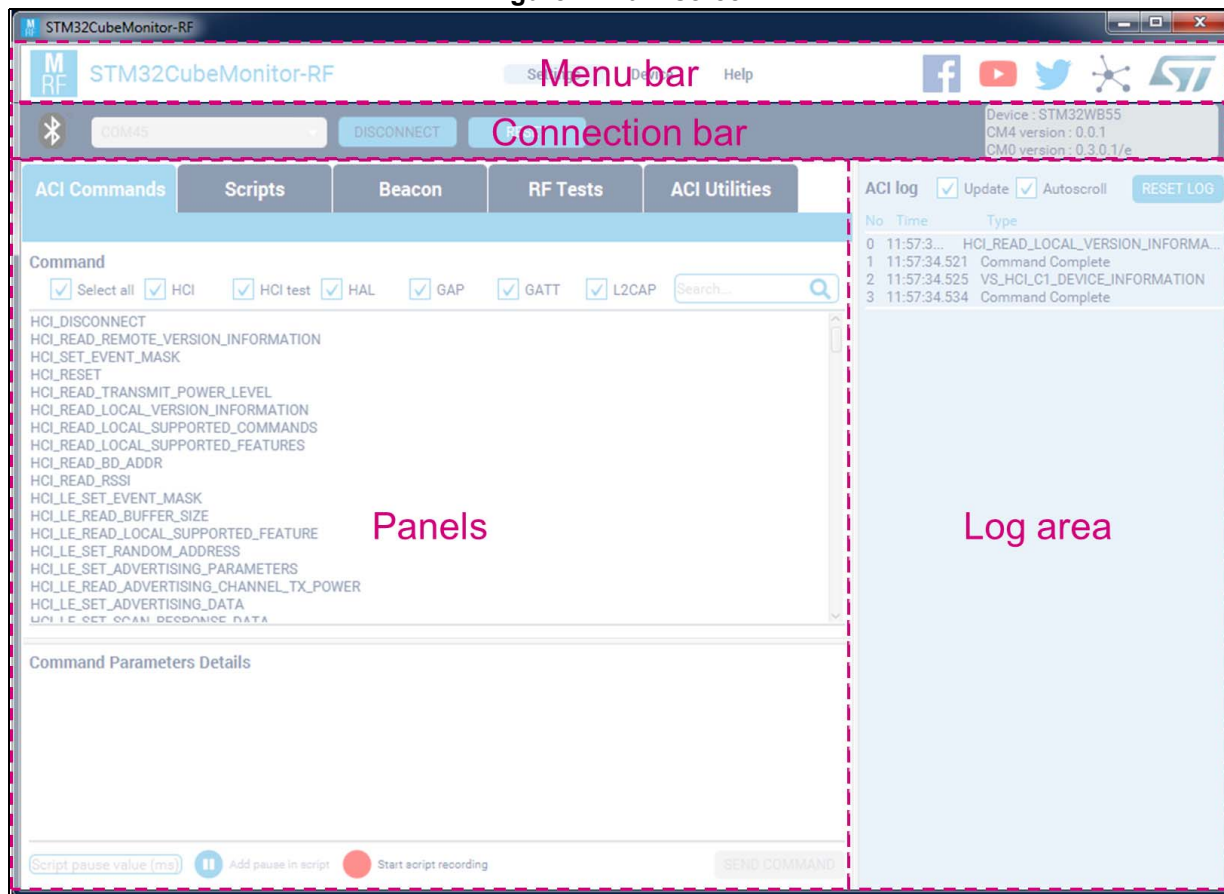
Launching the application opens the welcome screen, where the user selects the mode that he wants to use: BLE, OPENTHREAD or 802.15.4 RF.

The check box 'Remember my choice' memorizes the selection, so that the next application launch directly opens it, without welcome screen.

1.3 Main screen

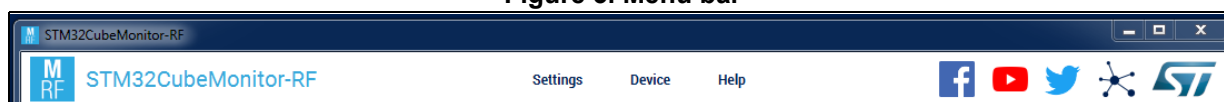
The main screen of the tools is subdivided in 4 parts: menu bar, connection bar, panels and log area.

Figure 2. Main screen



1.3.1 Menu bar

Figure 3. Menu bar



The application header provides a menu to use specific tools and display help information.

The 'Settings' menu allows the mode change as well as the reset of the default mode choice. The reset of the choice makes the welcome screen appear again.

The 'Device' menu provides information and actions related to connected board.

The 'Help' menu provides information about the version of tool used.

The social network links are available in the right corner. This area contains five shortcuts to access social networks:

- The Facebook™ icon leads to the official STMicroelectronics Facebook page
- The YouTube™ icon leads to the official STMicroelectronics YouTube page
- The Tweeter™ icon leads to the official STMicroelectronics Tweeter page.
- The Share icon leads to the ST Community web site
- The STMicroelectronics icon leads to the STMicroelectronics web site.

1.3.2 Connection bar

Figure 4. Connection bar



This part displays information related to the device connected to the application.

The icon on the left side reminds the mode selected.

The picklist helps select the COM port to use, buttons allow connecting to, disconnecting from, or resetting the target.

Information about the part connected is displayed on the right.

The RESET button is used to reinitialize the BLE wireless stack. When many tests are performed, the button should be used to reset the stack at start of each test.

1.3.3 Panels

The panels are used to perform specific operations. Each panel groups different functions. The 'ACI Commands' panel example is illustrated in [Figure 5](#).

Figure 5. ACI Commands panel

Command

☒ Select all
 ☒ HCI
 ☒ HCI test
 ☒ HAL
 ☒ GAP
 ☒ GATT
 ☒ L2CAP

HCI_DISCONNECT
 HCI_READ_REMOTE_VERSION_INFORMATION
 HCI_SET_EVENT_MASK
HCI_RESET
 HCI_READ_TRANSMIT_POWER_LEVEL
 HCI_READ_LOCAL_VERSION_INFORMATION
 HCI_READ_LOCAL_SUPPORTED_COMMANDS
 HCI_READ_LOCAL_SUPPORTED_FEATURES
 HCI_READ_BD_ADDR
 HCI_READ_RSSI
 HCI_LE_SET_EVENT_MASK
 HCI_LE_READ_BUFFER_SIZE
 HCI_LE_READ_LOCAL_SUPPORTED_FEATURE
 HCI_LE_SET_RANDOM_ADDRESS
 HCI_LE_SET_ADVERTISING_PARAMETERS
 HCI_LE_READ_ADVERTISING_CHANNEL_TX_POWER
 HCI_LE_SET_ADVERTISING_DATA

Command Parameters Details

Parameter	Value	Literal	Info
HCI packet indicator	0x01	HCI Command Packet	
Op_Code	0x0C03	HCI_RESET	
Parameter_Total_Length	0x00		

Script pause value (ms)

The main panels are: 'ACI Commands', 'Scripts', 'Beacon', 'RF Tests', and 'ACI utilities'. Each panel is detailed in a specific section of the document:

- [Section 3.2: ACI Commands panel on page 21](#)
- [Section 3.4: Scripts on page 38](#)
- [Section 3.6: Beacon on page 51](#)
- [Section 3.3: RF tests panel on page 26](#)
- [Section 3.7: ACI utilities on page 60](#)

1.3.4 Log area

Figure 6. Log area

The screenshot displays the 'Log area' interface. At the top, there are checkboxes for 'Update' and 'Autoscroll', both of which are checked, and a 'RESET LOG' button. Below this is a table with columns 'No', 'Time', and 'Type'. The table lists various messages, with the 17th message, 'ACI_GATT_INIT', highlighted in yellow. To the right of the table, an arrow labeled 'Message' points to the highlighted row. Below the highlighted row, a detailed view of the message is shown, with an arrow labeled 'Message detail' pointing to it. This detail view includes a table with columns 'Parameter', 'Value', and 'Literal'. The parameters listed are 'HCI packet indicator' (0x01), 'Op_Code' (0xFD01), and 'Parameter_Total_Len...' (0x00). The 'Literal' column provides descriptions for these values: 'HCI Command Packet', 'ACI_GATT_INIT', and an empty space. A '+ More' button is located at the bottom right of the detail view. The log continues with messages 18 through 31, including 'Command Complete', 'ACI_GAP_INIT', 'ACI_GATT_UPDATE_CHAR_VALUE', 'ACI_GAP_START_GENERAL_DISCOVERY_PROC', 'Command Status', and several 'LE Meta Event' entries.

No	Time	Type
5	18:17:48.921	HCI_RESET
6	18:17:48.926	Command Complete
7	18:17:49.466	ACI_HAL_SET_TX_POWER_LEVEL
8	18:17:49.469	Command Complete
9	18:17:50.947	ACI_HAL_TONE_START
10	18:17:50.951	Command Complete
11	18:17:53.961	ACI_HAL_TONE_STOP
12	18:17:53.964	Command Complete
13	18:17:59.764	HCI_RESET
14	18:17:59.769	Command Complete
15	18:17:59.777	ACI_HAL_SET_TX_POWER_LEVEL
16	18:17:59.781	Command Complete
17	18:17:59.789	ACI_GATT_INIT
18	18:17:59.792	Command Complete
19	18:17:59.797	ACI_GAP_INIT
20	18:17:59.801	Command Complete
21	18:17:59.806	ACI_GATT_UPDATE_CHAR_VALUE
22	18:17:59.811	Command Complete
23	18:17:59.815	ACI_GAP_START_GENERAL_DISCOVERY_PROC
24	18:17:59.818	Command Status
25	18:17:59.873	LE Meta Event
26	18:18:00.026	LE Meta Event
27	18:18:00.059	LE Meta Event
28	18:18:00.148	LE Meta Event
29	18:18:00.343	LE Meta Event
30	18:18:00.566	LE Meta Event
31	18:18:01.149	LE Meta Event

Parameter	Value	Literal
HCI packet indicator	0x01	HCI Command Packet
Op_Code	0xFD01	ACI_GATT_INIT
Parameter_Total_Len...	0x00	

MSv48412V2

The log area shows the messages exchanged between the application and the connected devices. The list shows all message names and details. The log area is described in [Section 3.2.5 on page 23](#).

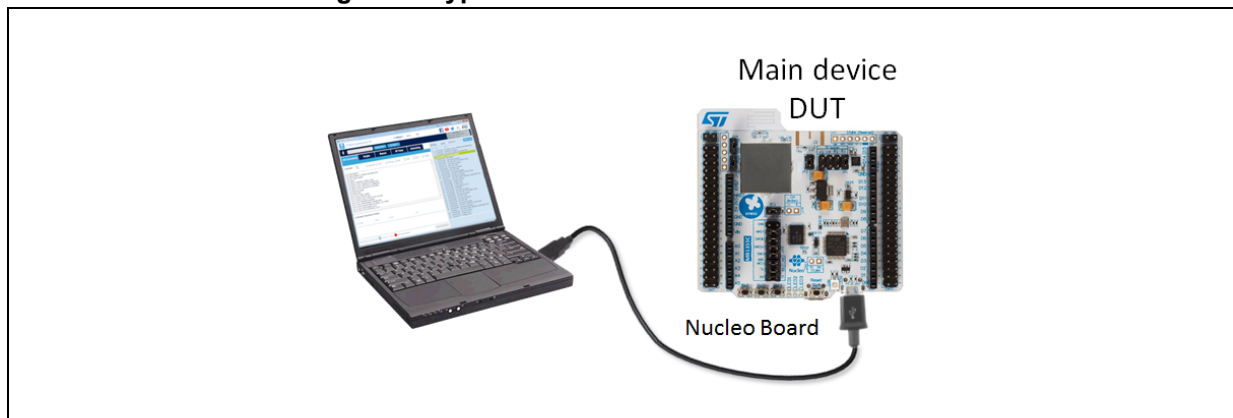
2 Connection to target

2.1 Use case description and definitions

STM32CubeMonitor-RF is usually connected to one STM32WBxx device. The connection is performed through a UART, either by a physical port or a Virtual COM port (VCP).

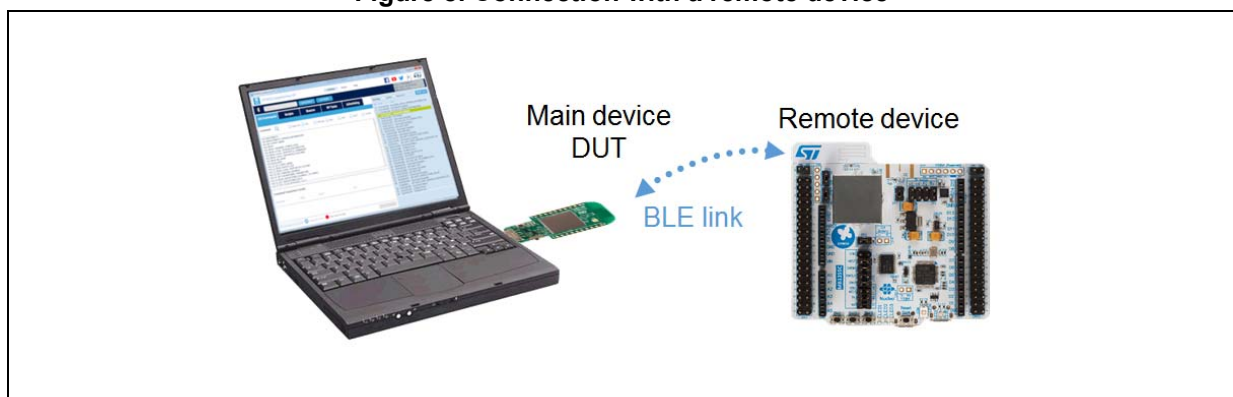
The device connected is usually named 'main device'. This is the board that the user wants to exercise with the tool. This board is also named device under test (DUT).

Figure 7. Typical connection with a Nucleo board



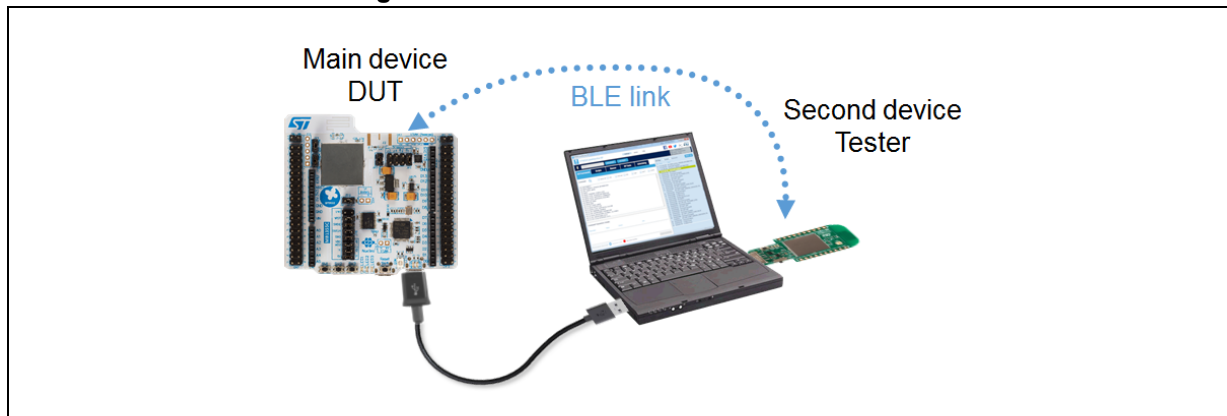
Some operation, like download over-the-air, involves communication with another device. This other device is referred to as 'remote device' in this document.

Figure 8. Connection with a remote device



One RF test makes use of two boards to perform packet transfer error rate measurement. For such a test, a second device is connected; it is named 'second device'. This latter device is the tester, the main device being the device to evaluate (DUT).

Figure 9. Connection with a second device



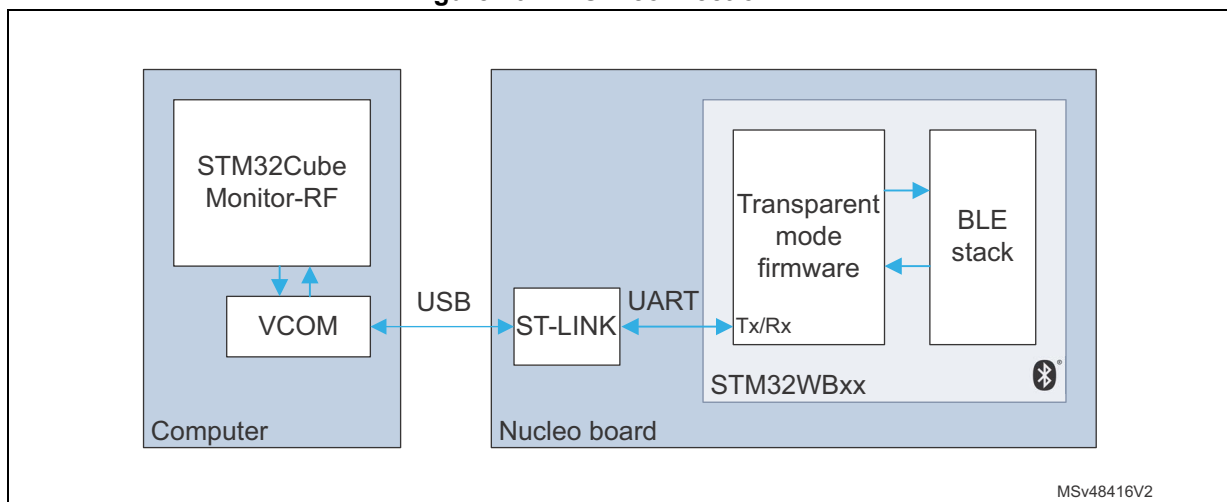
2.2 VCOM / UART connection

The connection must transfer host controller interface (HCI) or command line interface (CLI) commands between STM32CubeMonitor-RF and the wireless stack held in the STM32WBxx part. HCI commands are used for BLE applications, CLI are used for Thread and 802.15.4 RF tests. The application opens a serial port (virtual or physical) and communicates with the target through this link. Many configurations are possible. The most common ones are described in this section.

2.2.1 VCOM connection

The connection with a Nucleo board uses a Virtual COM port and goes through ST-LINK.

Figure 10. VCOM connection BLE



MSv48416V2

The application opens the Virtual COM port and sends the data to the VCOM driver.

When a byte is sent, the VCOM transfers the data over USB to the ST-LINK embedded in the Nucleo board. The ST-LINK transfers the data on UART lines to the STM32WBxx controller.

For BLE, a special firmware in STM32WBxx called 'transparent mode' copies the data received on the Rx pin to the BLE stack. Data sent back by the BLE stack follows the reverse path.

The transparent mode firmwares are available in the STM32CubeWB Firmware Package (see folder \Projects\NUCLEO-WB55.Nucleo\Applications\BLE\ble_transparent_mode).

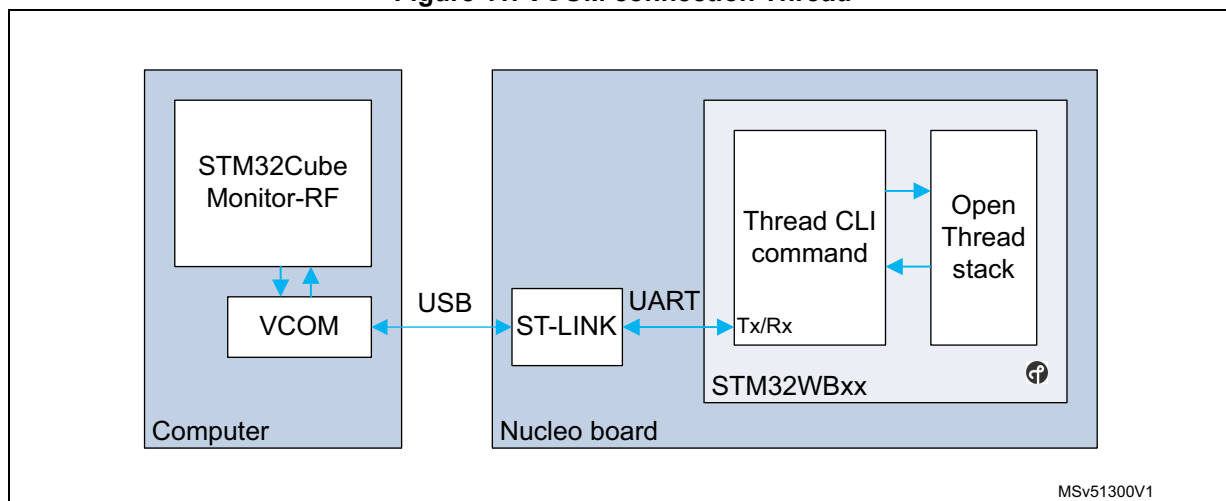
The wireless stack firmware stm32wb5x_BLE_Stack_fw.bin is available in \Projects\STM32WB_Copro_Wireless_Binaries.

For Thread, the 'Thread_Cli_cmd' firmware copies the data from the UART to the OpenThread command line interpreter. Data sent back by the interpreter are forwarded to the UART.

The CLI firmwares are available in the STM32CubeWB Firmware Package (see folder \Projects\NUCLEO-WB55.Nucleo\Applications\Thread\Thread_Cli_Cmd).

The wireless stack firmware stm32wb5x_Thread_FTD_fw.bin is available in \Projects\STM32WB_Copro_Wireless_Binaries.

Figure 11. VCOM connection Thread



For 802.15.4 RF tests, the Cli_Phy_802_15_4 firmware transfers the data from the UART to the 802.15.4 wireless stack. Data sent back by the stack follows the reverse path.

The Cli_Phy_802_15_4_M4.bin firmware binary is provided in the tool folder:

- for Windows®, <Documents>\STMicroelectronics\STM32CubeMonitor-RF\firmwares
- for Linux®, <userhome>/STMicroelectronics/STM32CubeMonitor-RF/firmwares
- for macOS®, it is inside the document folder provided in the setup package

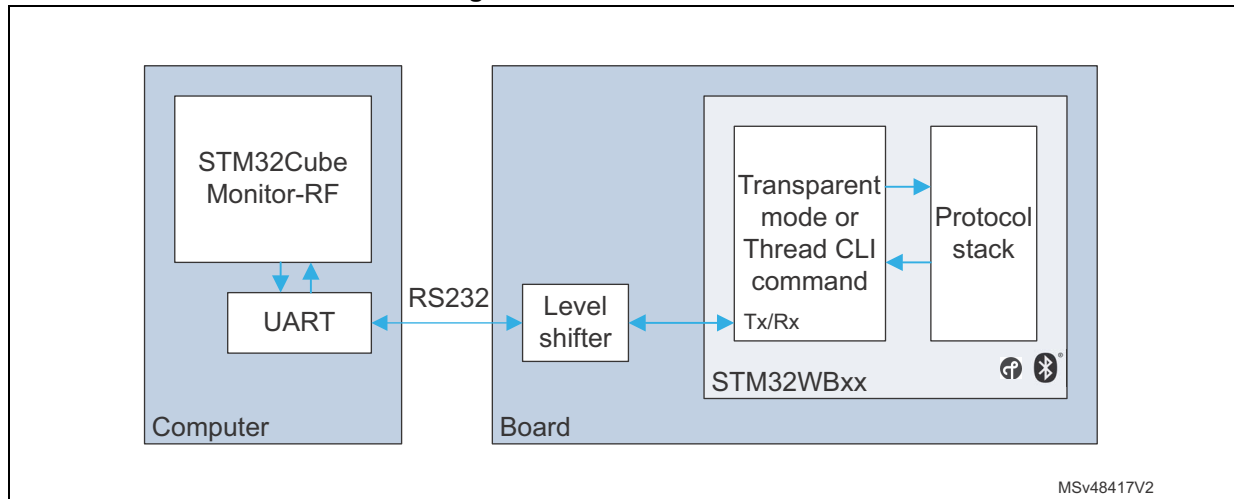
The wireless stack firmware stm32wb5x_rfmonitor_phy802_15_4_fw.bin is available in \Projects\STM32WB_Copro_Wireless_Binaries

When the ST-LINK part is replaced by a USB to serial converter, the VCOM driver may be installed automatically on the computer. For the converter without automatic driver setup, the user must install the VCOM driver manually.

2.2.2 UART connection

It is possible to use a physical UART link to connect directly to any board.

Figure 12. UART connection



In this case, data are sent directly in serial mode through the level shifter. Refer to the transparent mode or CLI command release note for UART configuration.

The UART connection can be used to connect an STM32WB55 USB dongle for 802.15.4 RF tests.

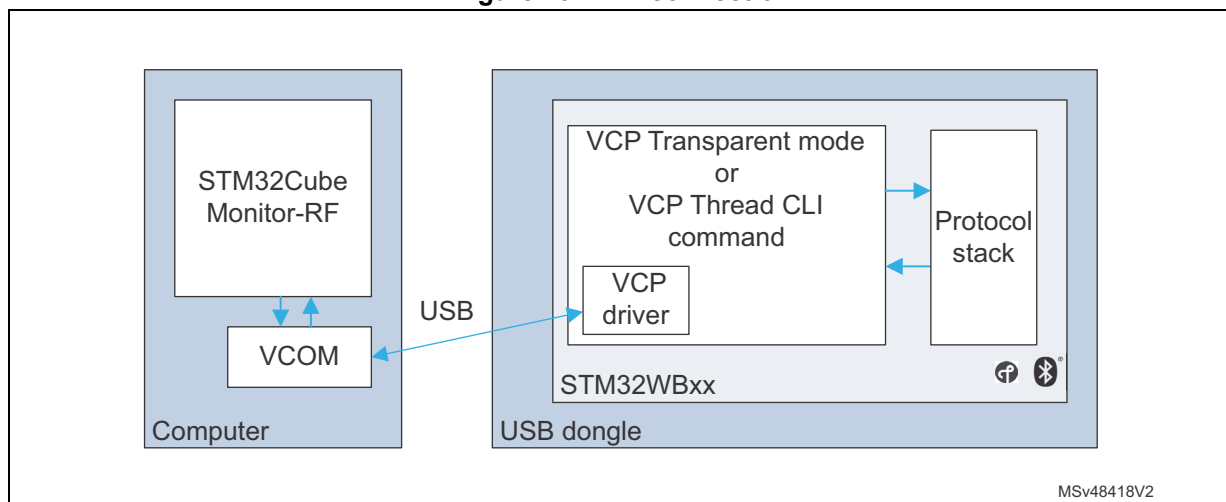
To configure the USB dongle for 802.15.4 test:

1. Flash the Nucleo binary `Cli_Phy_802_15_4_M4.bin` and the wireless stack `stm32wb5x_rfmonitor_phy802_15_4_fw.bin` in the dongle (with DFU).
2. Move solder bridge SB2 to SB6 (connection of PB7 to CN2.7).
3. Connect serial cable to PB7 (PC Tx) and PB6 (PC Rx) (PB7 is CN2.7 and PB6 on CN2.6).

2.2.3 VCP device

In this case, no UART is involved. The data goes directly from the computer to the microcontroller through the USB.

Figure 13. VCP connection



A special VCP firmware is used. It implements a VCP driver to copy the data from the USB port to the protocol stack. The VCOM driver may be installed automatically on the computer or needs to be installed manually by the user. This configuration is used for the STM32WB55 USB dongle reference board.

1. For BLE:
the firmware is in
`\Projects\NUCLEO WB55.USB Dongle\Applications\BLE\ble_transparent_mode_vcp.`
 The wireless stack is in
`\Projects\STM32WB_Copro_Wireless_Binaries\stm32wb5x_BLE_Stack_fw.bin`
2. For Thread:
the firmware is in
`\Projects\NUCLEO-WB55.USB Dongle\Applications\Thread\Thread_Cli_Cmd.`
 The wireless stack is in
`\Projects\STM32WB_Copro_Wireless_Binaries\stm32wb5x_Thread_FTD_fw.bin`

2.3 Opening COM

The first step to use the application is to connect to the device under test in the connection bar:

Figure 14. Opening COM



The procedure is:

- Connect the board to the computer. If VCOM or VCP is used, a driver needs to be installed; it may take a few seconds at the first connection. For some device, drivers need to be installed manually.
- Select the serial port to use in the picklist (Comx on Windows, ttyACMx on Linux and Mac).
- Click CONNECT

The board is connected, and the version is displayed in the right side of the bar.

Figure 15. Successful COM



When the CONNECT button is pressed, the software attempts to communicate with the device to read the firmware and HW versions. If the connection is not working, the tool displays an error and disconnects the COM port.

Caution: In case of connection error, the user must check these points:

- When a board is connected for the first time, it takes some time to load the drivers, or driver may not install automatically. If the tool is not showing the COM port in the list, check that drivers are properly installed.
- Delay on Ubuntu^{®(a)}:
 - On Ubuntu, the modemmanager process is checking the COM port when the board is plugged. Due to this activity, the COM port is busy during a few seconds, and STM32CubeMonitor-RF is unable to connect.
 - The user must wait the end of modemmanager activity before opening the COM port.
 - If modemmanager is not required by the user, it is possible to uninstall it with the command 'sudo apt-get purge modemmanager'.
- Port not visible on Linux:
 - The user may not have the proper access rights for ttyACM. In Ubuntu, it is required to add the user to the dialout group with the command 'sudo adduser <username> dialout' (replace username with user name).
- If the port is opened by another application, the tool is unable to connect.
- When a USB device is removed, the Virtual COM port is not closed automatically, and software may not be informed of the disconnection. If a USB device is inserted when the virtual port is already opened, the board is not mounted in the system. To solve this, close the COM port on STM32CubeMonitor-RF, disconnect and re-insert the USB cable. In some rare cases, it is mandatory to enable or disable the COM port in the OS device manager.

a. Ubuntu is a registered trademark of Canonical Ltd.

3 BLE mode

3.1 Presentation

3.1.1 Panels

The panels are used to perform specific operation. Each panel regroups different functions, as [Figure 16](#) shows it when “ACI Commands” is selected.

Figure 16. ACI Commands panel

ACI Commands | Scripts | Beacon | RF Tests | ACI Utilities

Command

☒ Select all ☒ HCI ☒ HCI test ☒ HAL ☒ GAP ☒ GATT ☒ L2CAP

HCI_DISCONNECT
HCI_READ_REMOTE_VERSION_INFORMATION
HCI_SET_EVENT_MASK
HCI_RESET
HCI_READ_TRANSMIT_POWER_LEVEL
HCI_READ_LOCAL_VERSION_INFORMATION
HCI_READ_LOCAL_SUPPORTED_COMMANDS
HCI_READ_LOCAL_SUPPORTED_FEATURES
HCI_READ_BD_ADDR
HCI_READ_RSSI
HCI_LE_SET_EVENT_MASK
HCI_LE_READ_BUFFER_SIZE
HCI_LE_READ_LOCAL_SUPPORTED_FEATURE
HCI_LE_SET_RANDOM_ADDRESS
HCI_LE_SET_ADVERTISING_PARAMETERS
HCI_LE_READ_ADVERTISING_CHANNEL_TX_POWER
HCI_LE_SET_ADVERTISING_DATA

Command Parameters Details

Parameter	Value	Literal	Info
HCI packet indicator	0x01	HCI Command Packet	
Op_Code	0x0C2D	HCI_READ_TRANSMIT_POWER_L...	
Parameter_Total_Len...	0x03		
Connection_Handle	0x002A		Specifies which Connection_Handle's Tr...
Type	0x01		0x00: Read Current Transmit Power Lev

Script pause value (ms)

The main panels are: ACI Commands, Scripts, Beacon, RF Tests, ACI Utilities. They are detailed in the next sections.

3.2 ACI Commands panel

The application command interface (ACI) panel is used to send commands to the main device BLE stack. Commands are grouped by categories. These commands allow the user to configure the BLE stack and activate the communication with remote devices.

3.2.1 How to send an ACI command

Figure 17. How to send an ACI command

The screenshot shows the ACI Commands panel. At the top, there is a 'Command' section with checkboxes for 'Select all', 'HCI', 'HCI test', 'HAL', 'GAP', 'GATT', and 'L2CAP'. A search bar is also present. Below this is a list of commands, with 'HCI_READ_TRANSMIT_POWER_LEVEL' highlighted in yellow. The bottom section is titled 'Command Parameters Details' and contains a table with four columns: Parameter, Value, Literal, and Info.

Parameter	Value	Literal	Info
HCI packet indicator	0x01	HCI Command Packet	
Op_Code	0x0C2D	HCI_READ_TRANSMIT_POWER_L...	
Parameter_Total_Len...	0x03		
Connection_Handle	0x002A		Specifies which Connection_Handle's Tr...

Before sending any command to the main device, the device must be connected.

To send an ACI command:

1. Select a command name in the command list (for example HCI_READ_TRANSMIT_POWER_LEVEL).
The command parameters are displayed in the Command Parameters Details area.
2. Fill the parameters of the command. Default values are used otherwise.
3. Click on 'SEND COMMAND' The command is sent to main device

3.2.2 Search function

The search icon is used to quickly select a command in the list:

- Click on the magnifier icon. A text box is created
- Type the name to search. As soon as character is entered, matching commands are filtered in the list. The match may be any part of the command name, it is no necessary to start from the beginning.
- Click once on the command to select it (do not use double click).

Figure 18. Search button

3.2.3 Filter usage

The commands are grouped and named by features. Groups are:

- HCI
- HCI test
- HAL
- GATT
- GAP
- L2CAP

The picklist at the top of the area allows to see only some groups to find more easily the commands. Click on 'Select all' to see all commands in the list.

3.2.4 How to fill parameters. Fixed field / editable field

Some parameters have fixed value and are not editable, while others are totally free or take only some values. The tool guides the user to fill the parameters:

- Fixed parameter: this parameter is not editable. The value is defined by specification, or by logic. This applies to 'length' value which is computed by the tool automatically.

Figure 19. Fixed parameter

- Editable parameter: editable parameter is surrounded by a blue rounded box. Value is editable inside the field. Edit is blocked if value is too long for the field.

Figure 20. Editable parameter

Connection_Handle	<input type="text" value="0x002A"/>
-------------------	-------------------------------------

- Predefined values: when choice is limited, a picklist is displayed to help the user to select the values.

Figure 21. Predefined values

Type	<input type="text" value="0x01"/>
------	-----------------------------------

For some parameters, some help is available in the column 'Info'. To see the help details, put the pointer on the wanted parameter info, and a bubble displays the details.

Figure 22. Help details

Parameter_Total_Len...	<input type="text" value="0x03"/>	
Connection_Handle	<input type="text" value="0x002A"/>	
Type	<input type="text" value="0x01"/>	Specifies which Connection_Handle's Transmit Power Level setting to read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

3.2.5 Log functionalities

The log area is located in the right part of the tool. It displays the messages exchanged with the boards.

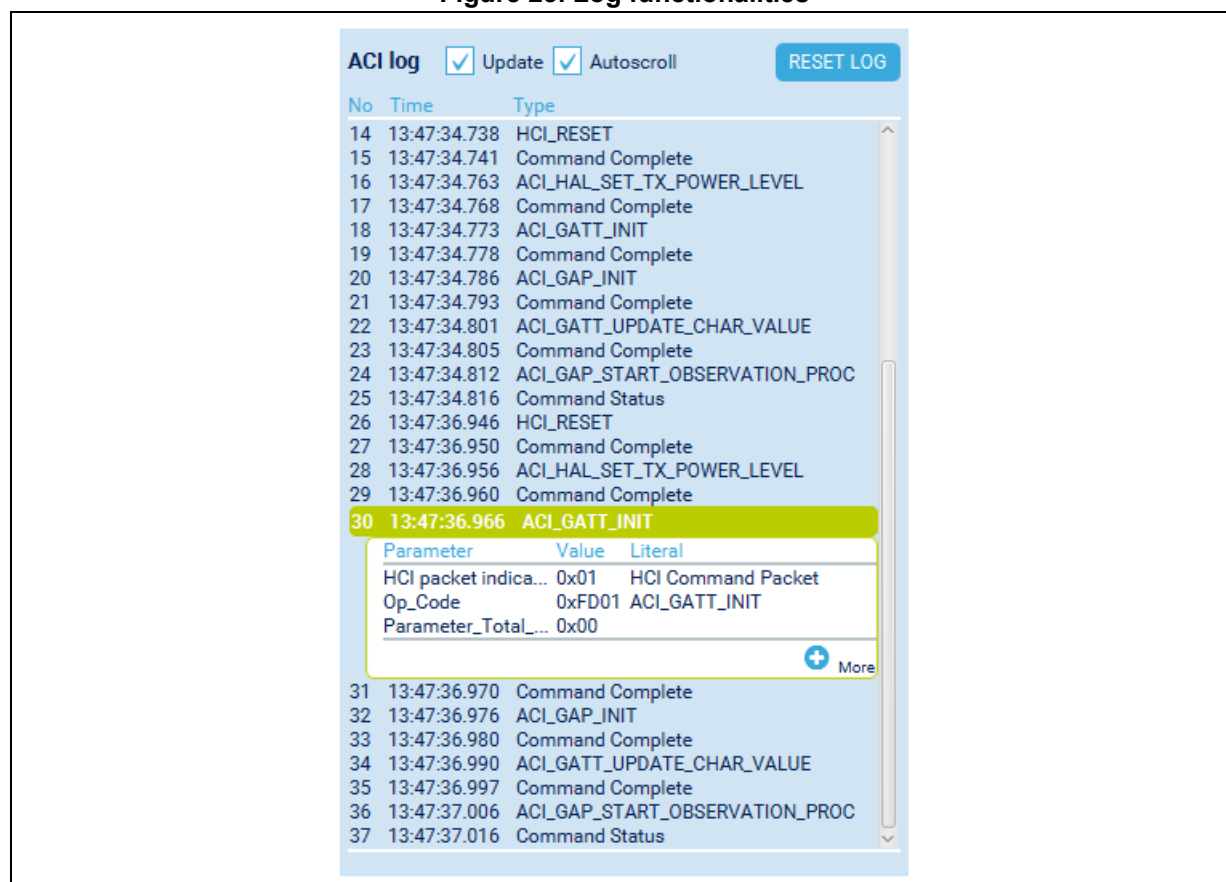
When a command is sent, most of the time an immediate answer comes from the board. It is a command status or command complete.

The commands with 'Command Status' usually have other events coming later. This events are also displayed in the log area.

Some asynchronous event may come from the device, and is also displayed in this area.

The tool keeps the last 1000 lines. When the limit is reached, the oldest lines are automatically discarded.

Figure 23. Log functionalities



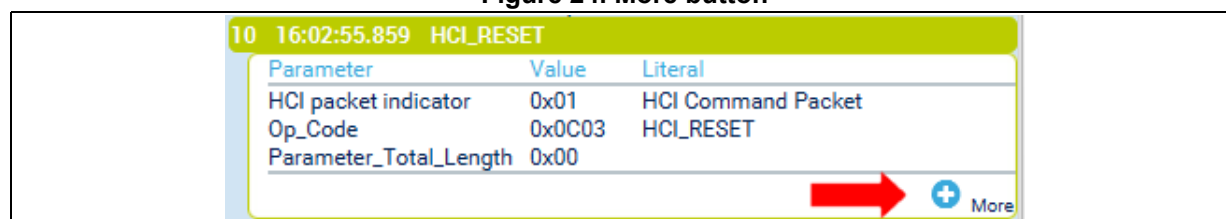
It is possible to scroll in the list with the scroll bar on the right side.

When a line is selected, the content of the selected message is displayed in the green area, with one line for each parameter.

The text ends with ..., when it is not possible to display complete text. It is possible to change the log area width to display longer texts.

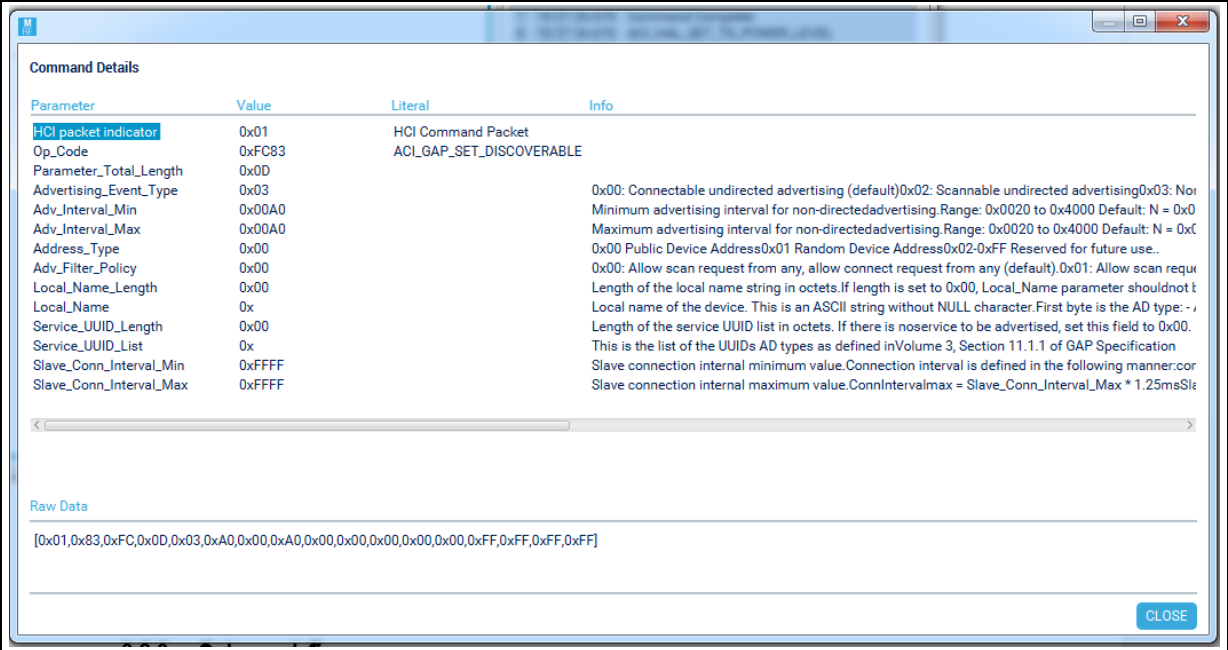
Details

Figure 24. More button



Sometimes, all the information of a message does not fit in the area used for log. The button [Figure 24](#) opens a new window showing the message details:

Figure 25. Message details



The details show all decoded message parameters. The 'Literal' column shows predefined text for the parameter values (Op-Code and others). The 'Info' column provides some description of parameter content.

The raw data in the bottom part is the data sent/received over UART, without decoding.

In this window, it is possible to copy information for pasting it in other windows.

An efficient solution to compare two messages is to open multiple details windows at the same time.

Color code

The logs use color code to identify the device used and highlight errors.

A line with a purple text shows that the status in the message is different from zero, which indicates an error.

Figure 26. Purple error messages

No	Time	Type
8	09:25:32.205	HCI_LE_CREATE_CONNECTION
9	09:25:32.212	Command Status

Log on a dark gray background are coming from a second board. When two boards are connected, the main device (DUT) has a normal color log while the second device tester has a darker background. This is helpful to understand the sequences involving two devices.

Figure 27. Gray second board messages

No	Time	Type
14	09:27:34.029	HCI_READ_LOCAL_VERSION_INFORMATION
15	09:27:34.036	Command Complete
16	09:27:37.634	HCI_READ_LOCAL_VERSION_INFORMATION
17	09:27:37.672	Command Complete

Update button

When the 'Update' tick box is not selected, the messages are not added in the log area. The line number continues to be increased anyway, but is not displayed until the 'Update' tick box is enabled.

Auto scroll

When the 'Auto scroll' box is ticked, the log area always displays the last log received. To check log history, untick the box to disable auto scroll.

Reset Log

The 'RESET LOG' button allows wiping the log displayed in the log area. The line number is not affected, but the memory used by older logs is made free.

3.3 RF tests panel

The RF tests panel is used to perform the radio-frequency tests on the main device. The RF tests are grouped in three test modes: Transmitter (TX), Receiver (RX) and Packet error rate (PER):

- The TX test is dedicated to radio-frequency emission, for tones and packets.
- The RX test is for packet reception.
- The PER test is a quality-transmission test between two devices.

Figure 28. Test mode selection

The screenshot shows a software interface with a top navigation bar containing five tabs: 'ACI Commands', 'Scripts', 'Beacon', 'RF Tests' (which is highlighted), and 'ACI Utilities'. Below the tabs is a section titled 'Test mode' with a subtitle 'Select test mode'. There are three radio button options: 'Transmitter test (TX)' (which is selected), 'Packet error test (PER)', and 'Receiver test (RX)'. At the bottom right of the panel is a blue button labeled 'SELECT TEST MODE'.

The first action after connecting a device is to select the mode to test, and then to click on the SELECT TEST MODE button.

When the user has selected a test mode, it is mandatory to go back to the selection page to change the test mode:

- Click on the 'Change test mode'

Figure 29. Change test mode



- Click on 'test mode' in the top bar.

Figure 30. Select test mode



Note: To avoid incorrect configuration of the device, the test mode is unchangeable, when transmission or reception are ongoing. The user must first stop the transmission and then change test mode.

3.3.1 Test mode transmitter (TX)

The TX mode is used to set the BLE transmitter in emission. Two transmission modes are defined: transmission of data, or emission of tone.

Figure 31. Test mode transmitter

Test mode > Transmitter (TX)

Transmitter

PA Level

31 (+6dBm)

TX Frequency

2402 MHz (Channel 37)

Length of Data

0x25

Packet Payload

0x00 - Pseudo-Random bit sequence 9


PHY

0x01 - Transmitter set to use the LE 1M PHY

Back

START TONE

START TX

Test measurement 

Transmitted packets count

Received packets count

Packet Error Rate (PER):

RSSI

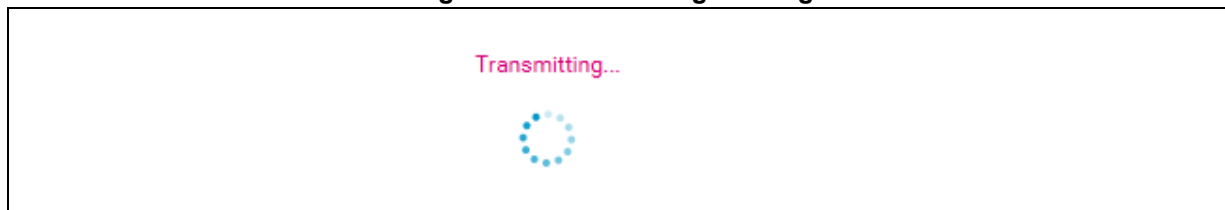
Tone generation

The tone generation performs the emission of a continuous sinus wave on the RF. The parameters for the tone are tone power level and tone frequency. The power level is the power at the chip output.

To start tone generation:

1. Enter the Transmitter panel test Mode
2. Select the power level with the picklist.
3. Select the frequency with the TX Frequency picklist. The list is sorted by frequency; the data/advertising channel index is indicated in parenthesis. Advertising channel index do not follow the frequency order. Channels 37, 38 and 39 are the advertising channels. See BLUETOOTH SPECIFICATION Version 4.2 [Vol 6, Part B] ch1.4.1 for details.
4. Select the PHY modulation to use (The modulation not supported by the device are not listed).
5. Click on 'START TONE' button.

The emission starts, 'START TONE' button is changed to 'STOP TONE', and Transmitting information is displayed:

Figure 32. Transmitting message

6. To stop the tone generation, click on 'STOP TONE' and the emission stops.
It is mandatory to stop transmission to change to another test mode.

Packet transmission

It is possible to send some data packets in test mode. The parameters are power level, transmission frequency, length and content of the data to send.

Power and level parameters are the same as tone parameters.

The packet data is selected in the Packet payload picklist. Eight types of payload are available:

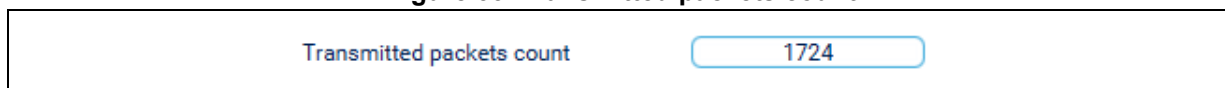
- Pseudo-Random bit sequence 9 (PRBS9)
- Pattern of alternating bits '11110000'
- Pattern of alternating bits '10101010'
- Pseudo-Random bit sequence 15 (PRBS15)
- Pattern of All '1' bits
- Pattern of All '0' bits
- Pattern of alternating bits '00001111'
- Pattern of alternating bits '01010101'

The sequence length is defined by the 'Length of data' picklist. This is the length of data payload in bytes. The PHY box is used to select the modulation.

To start packet emission:

- Select the power level with the picklist.
 - Select the frequency with the TX Frequency picklist.
 - Select the length of the packet to send
 - Select the content of the packet payload
 - Click on 'START TX'
- The emission starts, start button is changed to 'STOP TX', and Transmitting is displayed. The sequence is repeated until test is stopped

To stop the transmission, click on 'STOP TX'. The number of packets transmitted during the test is displayed in the test measurement area.

Figure 33. Transmitted packets count

If the number of packets received by the reception device is known, it is manually entered in the 'Received Packet Number' box, and the Packet Error Rate is automatically computed (see [Section 3.3.3: PER](#) for details).

3.3.2 Test mode receiver (RX)

The receiver mode is used to put the main device in reception mode and count packets received.

Figure 34. Test mode receiver

Packets reception:

- Select the frequency to use.
- Select the PHY and the modulation index to be used.
- Click on 'START RX'. The reception starts, 'Receiving' is displayed with an animation and button change to 'STOP RX'.

To stop reception, click on 'STOP RX'. The count of packet received is retrieved from the main device, and displayed in the 'Received packet number'.

If the number of transmitted packets is known, it may be entered manually in the 'Transmitted packet number'. The 'Packet error rate (PER)' is automatically computed (see [Section 3.3.3: PER](#) for details).

If the 'Get RSSI' check box is selected, the tool performs RSSI measurement.

RSSI measurement:

The RSSI indicates the level of signal received by the RF. The value reported by the RF is not an absolute value because the reception level is dependent of the board layout and antenna design.

When RSSI option is selected, the user must define the measurement interval. The default value is 3 seconds. The RSSI value is displayed at the end of each measurement period.

It is possible to switch between detailed value, plot view and big display, with the blue button on the right (bar chart, arrows or blue lines).

Figure 35. RSSI measurement

Test mode > Receiver (RX)

Receiver

RX Frequency
2402 MHz (Channel 37)

PHY
0x01 - Receiver set to use the LE 1M PHY

Index modulation
0x00 - Assume transmitter will have a standard modulation index

☒ Get RSSI
Measurement period (sec): 3

STOP RX

Test measurement at 2402 MHz (Channel 37)

Transmitted packets count

Received packets count

Receiving...

Packet Error Rate (PER):

RSSI
-107.00 dBm

Note: When the RSSI measurement is performed, the number of received packets is not available in the tool. When the measurement is stopped, the 'Received packet number' field is cleared, and an information message is displayed.

Figure 36. RSSI measurement graph



Note: The graph length is limited to 250 points. When the limit is reached, the oldest points are discarded.

Figure 37. RF RSSI measurement big display

Test mode > Receiver (RX)

Receiver

RX Frequency: 2402 MHz (Channel 37)

PHY: 0x01 - Receiver set to use the LE 1M PHY

Index modulation: 0x00 - Assume transmitter will have a standard modulation index

☒ Get RSSI Measurement period (sec): 3

← Back START RX

Test measurement at 2402 MHz (Channel 37)

RSSI

-62.00 dBm

3.3.3 PER

PER definition

The packet error rate (PER) is an indicator of the quality of transmission between two devices. The measurement proposed in the tool covers the whole transmission chain from transmitter to receiver.

The packet error rate is computed with the number of packets sent and the number of packets received. A good transmission gives a low PER. High PER means that transmission is not good.

Figure 38. PER definition

$$\text{PER} = 100 \times \frac{N_{\text{tx}} - N_{\text{rx}}}{N_{\text{tx}}} \%$$

Ntx: number of packets sent, Nrx number of packets received, PER result in percent.

A bad PER may be an issue from the transmitter or receiver, and depends on parameters like distance between devices, antennas, PCB design and interferences. To limit the parameters influencing the measurements, it is advised to use one reference board with well known performances in the setup.

PER test mode

The tool provides a special test mode dedicated to PER test. In this mode, two devices need to be connected to the computer:

- the first device under test (DUT)
- the second device, used to test the DUT (tester)

After connection of the DUT (main device, connected in the application top bar), the PER test mode is selectable in the RF test page.

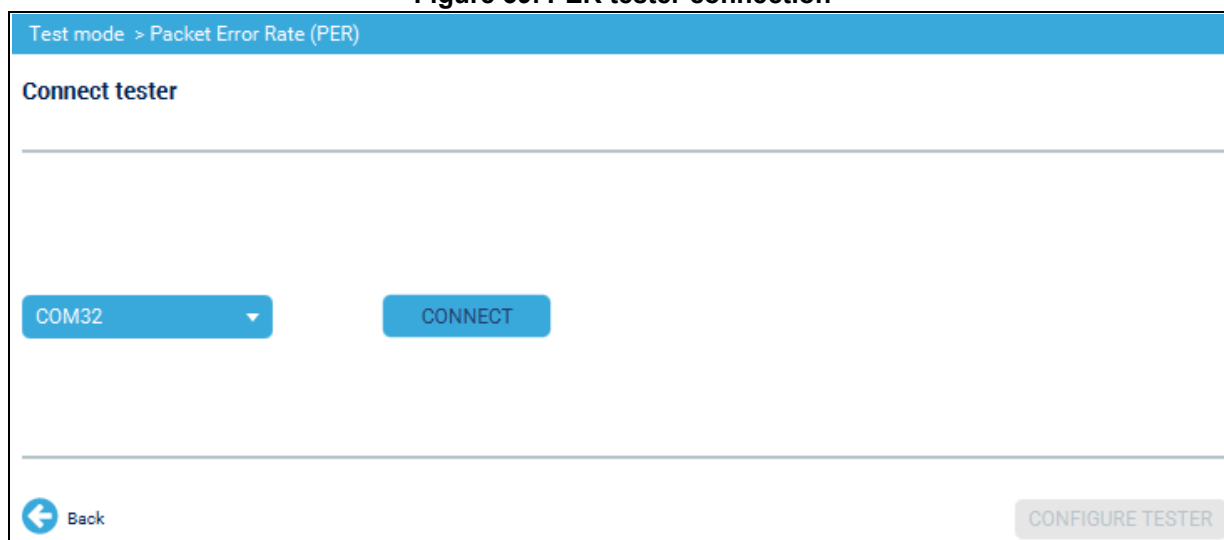
The configuration of PER test is done with a sequence of panels:

- tester connection
- tester configuration
- DUT configuration
- test parameters

The first step is to connect the tester:

PER tester connection

Figure 39. PER tester connection



- Plug the device in the computer (same requirements as first device, see [Chapter 2.2](#)).
- Select the serial port to use in the picklist.
- Click on the 'CONNECT' button.

Figure 40. PER tester connected

Test mode > Packet Error Rate (PER)

Connect tester

COM32

DISCONNECT

Device : STM32WB55
CM4 version : 0.0.1
CM0 version : 0.2.7.1/d

Disconnect 2nd device to change test mode

CONFIGURE TESTER

- The board information is displayed on the right.

When second device is connected, it is not possible to change mode. Disconnect the device first, and then use the 'back' button.

Click on 'CONFIGURE TESTER' to set the tester parameters:

Figure 41. PER tester configuration

Test mode > Packet Error Rate (PER) > COM33

Configure tester (COM33)

PA Level: 31 (+6dBm)

TX Frequency: 2402 MHz (Channel 37)

Length of Data: 0x25

Packet Payload: 0x00 - Pseudo-Random bit sequence 9

PHY: 0x01 - Transmitter set to use the LE 1M PHY

Back

CONFIGURE DUT

- Select the TX power level with the picklist.
- Select the transmission frequency with 'TX Frequency' picklist.
- Select the length of packet to send (same as TX test).
- Select the content of packet payload (same as TX test).
- Select the PHY to use.

Click on 'CONFIGURE DUT' to set the Device Under Test configuration:

Figure 42. DUT configuration

The screenshot displays a web-based configuration interface for a Device Under Test (DUT). At the top, a blue breadcrumb trail reads 'Test mode > Packet Error Rate (PER) > COM33 > COM26'. Below this, the title 'Configure Device under test (DUT) (COM26)' is shown. The interface contains three configuration rows, each with a label on the left and a dropdown menu on the right. The first row is 'RX Frequency' with the value '2402 MHz (Channel 37)'. The second row is 'PHY' with the value '0x01 - Receiver set to use the LE 1M PHY'. The third row is 'Index modulation' with the value '0x00 - Assume transmitter will have a standard modulation index'. At the bottom left, there is a 'Back' button with a left-pointing arrow. At the bottom right, there is a blue button labeled 'CONFIGURE PARAM'.

Select the reception frequency, the PHY and the modulation index for the receiver board. The tool uses by default the same frequency as the tester, but the user may modify it.

Click on 'CONFIGURE PARAM' to set the test configuration:

Figure 43. PER test parameters

Test mode > Packet Error Rate (PER) > COM37 > COM32 > Settings

Configure additional settings

☒ PER tests on multiple channels Fill channel List:

☒ Get RSSI Measurement period (sec):

☐ Save test verdict in file

← Back START TEST

Test measurement

Transmitted packets count

Received packets count

Packet Error Rate (PER):

RSSI

- ‘PER tests on multiple channels’: when this option is selected, the PER test is performed on a list of predefined channels. When the box is ticked, the Channel list is displayed. Value 0-39 indicates all channels between 0 and 39. It is also possible to put value separated by comma: ‘0,1,5’ or to mix: ‘0,1,10-15’.
The measurement period is the time of each PER test to be performed.
- ‘Get RSSI’: this option add some RSSI measurement between each PER measurement. When it is activated, the tool performs a PER test for the Measurement period, compute PER, and then make a RSSI check.
- ‘Save test verdict in file’: this option generates a test report of the measurements. When the option is selected, a ‘SELECT FILE’ button is displayed. The user must select the report file before starting the tests. Report is saved at the end of the tests.

When option have been configured, click on ‘START TEST’ button:

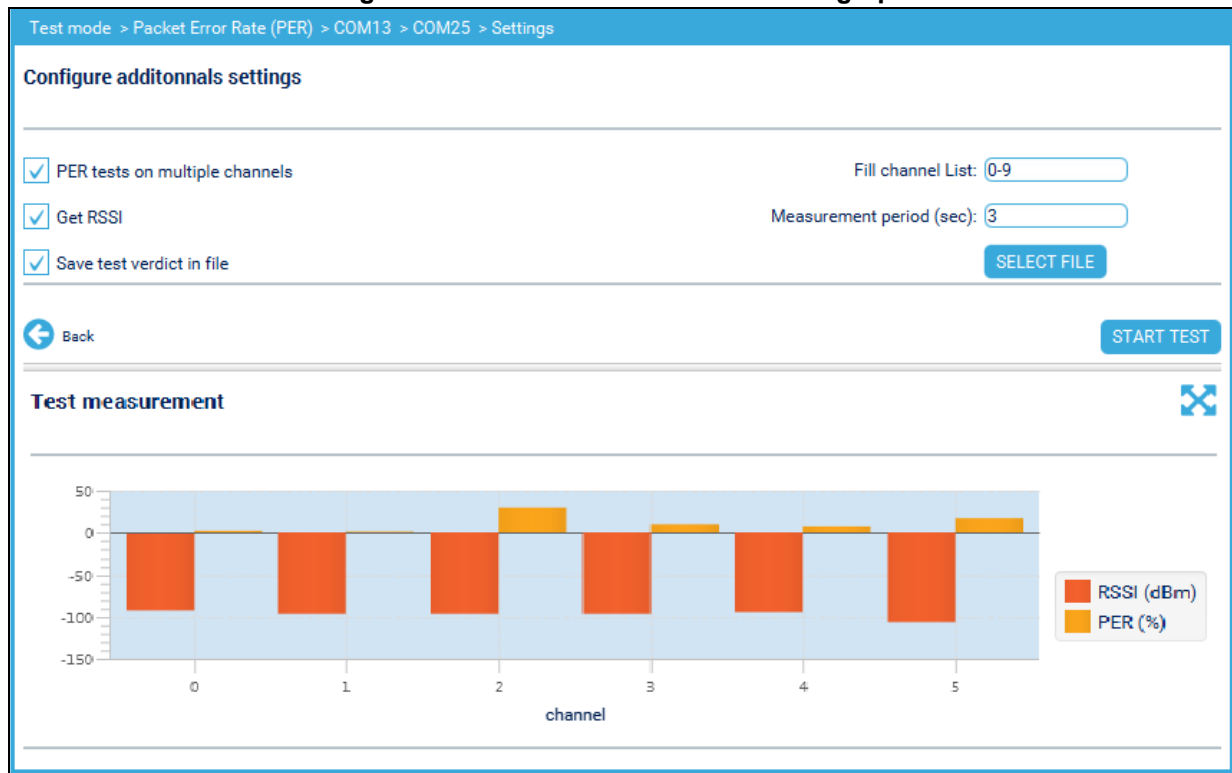
- the DUT is set in reception mode,
- then the tester starts.

Button is changed to ‘STOP’ and ‘Testing...’ is displayed.

Test continues until the user presses the ‘STOP’ button, or when all channels have been measured for multichannel tests.

The result is displayed in the bottom part. It is possible to switch between the numerical values and the chart with the blue bars icon.

Figure 44. PER and RSSI measurement graph



3.4 Scripts

Scripts are used to execute in sequence some commands stored in a text file. Scripts help avoid entering each command manually for repetitive tasks.

3.4.1 Launching scripts

Figure 45. Launching scripts

The screenshot displays the STM32CubeMonitor-RF software interface. At the top, there is a dark blue header bar containing a Bluetooth icon, a dropdown menu showing 'COM45', and two buttons: 'DISCONNECT' and 'RESET'. Below the header is a navigation bar with five tabs: 'ACI Commands', 'Scripts', 'Beacon', 'RF Tests', and 'ACI Utilities'. The 'Scripts' tab is currently selected. The main area of the 'Scripts' tab is titled 'Script' and contains a 'Generate report' checkbox (which is checked), a text input field with the file path 'C:\Users\Public\Documents\STMicroelectronics\STM32CubeMonitor-RF\scripts\BLE_Sample', and a 'BROWSE' button to the right of the input field. At the bottom right of the main area is a 'START SCRIPT' button.

Scripts are stored in text files and are editable with any text editor.

To execute a script:

- Select the script file with the browse button or enter file name directly.
- Click on 'Start script' button.
- The script is displayed and executed. The line in execution is highlighted in green. The ACI results are updated in the log area.
- Script is manually stopped with the 'Stop script' button.

Figure 46. Script execution

Script

☒ Generate report

C:\Program Files\STMicroelectronics\STM32CubeMonitor-RF\scripts\Sample Script.txt

BROWSE

```

# Send reset command :
Send(HCI_RESET)

# Wait few milliseconds
Wait(500)

# Send another command : Set power level
Send(ACI_HAL_SET_TX_POWER_LEVEL;0x01;0x07)

# Pause command
Pause("This is a pause")

# Start Tone
Send(ACI_HAL_TONE_START;0x04;0x00)

# Wait 3 seconds
Wait(3000)

# Send stop tone
Send (ACI_HAL_TONE_STOP)

```

START SCRIPT

Script examples are provided with the tool (Sample script, loop, beacon creation, etc).

For Windows[®] scripts are in folder
<Documents>\STMicroelectronics\STM32CubeMonitor-RF\scripts (public documents).

For Linux[®], they are in <userhome>/STMicroelectronics/STM32CubeMonitor-RF/scripts.

For macOS[®], it is inside the document folder provided in the setup package.

3.4.2 Script recording

The ACI commands used in the ACI panel is directly recorded in a script. Some script buttons are located at the bottom of ACI panel:

Figure 47. Script recording buttons

Script pause value (ms)

Add pause in script

Start script recording

Use the red button to start recording. Pause is inserted with the 'Add pause in script' button.

At the end of recording, click the 'Stop' button. The tool asks the script name before saving.

3.4.3 Scripts modification

Script is created or modified with a text editor. It uses a simple syntax to list the ACI command to send and the action to perform.

Figure 48. Sample script

```
# Send reset command :
Send(HCI_RESET)

# Wait few milliseconds
Wait(500)

# Send another command : Set power level
Send(ACI_HAL_SET_TX_POWER_LEVEL;0x01;0x07)

# Start Tone
Send(ACI_HAL_TONE_START;0x00)

# Wait 3 seconds
Wait(3000)

# Send stop tone
Send (ACI_HAL_TONE_STOP)

# Pause command
Pause("End of script")
```

The lines starting with # are comments and are ignored by the tool. Empty lines are skipped. Other lines are commands. Line starts with the command name, followed by parameters in brackets, separated by semicolon.

3.4.4 Script report

It is possible to have a script report generated at the end of script execution. The script report stores the status of each ACI command executed by the script.

Figure 49. Script report

SCRIPT REPORT			
Script name : SampleScript.txt			
Test date : 18/12/2017 17:27:52			
Verdict : SUCCESS No error detected			
Command	Sent	ACI status	ACI raw result
HCI_RESET	OK	0x00	
[0x04, 0x0E, 0x04, 0x01, 0x03, 0x0C, 0x00]			
ACI_HAL_SET_TX_POWER_LEVEL	OK	0x00	
[0x04, 0x0E, 0x04, 0x01, 0x0F, 0xFC, 0x00]			
ACI_HAL_TONE_START	OK	0x00	
[0x04, 0x0E, 0x04, 0x01, 0x15, 0xFC, 0x00]			
ACI_HAL_TONE_STOP	OK	0x00	
[0x04, 0x0E, 0x04, 0x01, 0x16, 0xFC, 0x00]			
END of report			

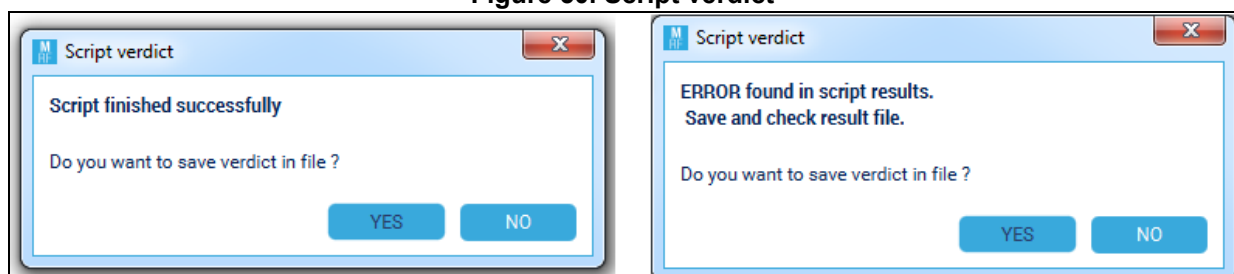
The result is stored in a new file, in same path as script, with a name in the form: 'verdict_SampleScript_18-12-2017_17-27-52'. The name is built with concatenation of:

- 'verdict_'
- the script name
- the current date
- the current time
- .txt extension

In the report, the 'Sent' column holds the status of command transfer to board. If parameters are missing, command is not sent.

The 'ACI status' column, have the status of ACI response. 0x00 is a success status, other values are errors.

At the end of script execution, a popup with verdict (error found or finish successfully) is displayed and asks if report should be saved:

Figure 50. Script verdict

If the user presses 'yes', report is generated in the folder of current script. If the user presses 'no', report is not saved.

If the tick box 'Generate report' is not checked, no report is generated at the end of script. The script successfully means there was no error in the script syntax, and the status of operations was OK (error code = 0). The value measured or the performance are not verified, there is no PASS/FAIL criteria on the results.

3.4.5 List of scripts commands

Send an ACI command:

The ACI commands are sent with the instruction 'Send'

```
'Send(ACI_CMD_NAME;Parameter1Value;Parameter2Value;...)'
```

The elements inside the parenthesis are separated by semicolons.

The first element is the command name. It is the name as it is displayed in the tool.

The next elements are the parameters. The value must be entered in hexadecimal format and start by 0x. The optional parameters can be left empty. The length is dependent of parameter size in the ACI command.

Note: The Command Packet Type, OpCode and Parameter Total Length are filled by the application. They must not be added in the parameters.

Wait for a specific time

It is possible to add a delay with the instruction 'Wait'

```
Wait(3000)
```

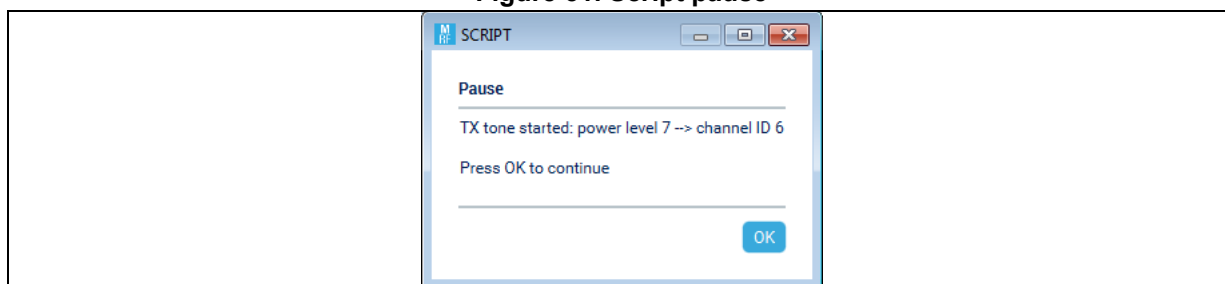
This instruction delays the script execution for 3 seconds. Time is given in milliseconds.

In the ACI panel screen, a pause is inserted in the script with the 'Add pause in script' button.

Pause command in the script

The Pause command adds a pause during the proceeding of the script. This command opens a pop-up window customized with the user comment.

Figure 51. Script pause



The 'OK' button allows to continue the script.

Command: Pause ('User comment')

The user text must be enclosed between quote marks (").

Figure 52. Example

```
# Pause demo script

# Start Tone
Send(ACI_HAL_TONE_START; 0x04)

# Pause command
Pause("TX tone started")

# Send stop tone
Send (ACI_HAL_TONE_STOP)
```

Loop command in the script

Loop can be used in script to repeat some actions automatically.

Loop usage

To repeat a part of a script, the commands must be enclosed between 2 instructions:

- **Loop (count,0,5);** this instruction indicates the beginning of the repeated section. 'count' is the name given to the counter, the first value is the start value, and the second one the end value. In this example, the counter count is be increased from 0 to 5; there are 6 iterations.
- **EndLoop:** indicate the end of the loop. If the counter reaches the end value, execution continues on the next line. If the counter has not reached the end value, the counter is updated, and execution goes back to the 'Loop' instruction.

Figure 53. Loop simple example

```
Loop(count; 1; 3)
Pause("test the loop")
EndLoop
```

This script given as an example in [Figure 53](#) displays 'test the loop' 3 times.

Using the counter value

It is possible to use the counter value in other lines of the script to change the parameter values during script execution. When the counter name is embedded inside square brackets, the tool inserts the counter value.

Figure 54. Loop second simple example

```
Loop(count ; 1 ;3)
Pause("The loop counter is [count]")
EndLoop
```

The script in [Figure 54](#) displays 'The loop counter is 1', then 'The loop counter is 2', and finally, 'The loop counter is 3'.

Some parameters require hexadecimal value. In this case, add an ampersand (&) after the first bracket. The tool replaces the counter name by the hexadecimal value.

If count = 10, [&count] is replaced by 0xA.

Special count option

The counter value can increase or decrease. If the start value is bigger than the end value, the counter is decremented.

Figure 55. Loop decrement

```
Loop(mycount; 3; 1)
```

In the countdown example in [Figure 55](#), mycount takes values 3, 2, and 1.

The counter can have a specific increment value when a third value is added to the loop instruction, as shown in [Figure 56](#):

Figure 56. Loop specific increment

```
Loop(mycount; 1; 6; 2)
```

This example counts with a step of 2. Successive values are 1, 3, and 5. The loop stops at 5 because 7 is higher than 6.

The loop can include another loop. It is mandatory to use different counter name.

Figure 57. Nested loop

```
Loop(row;4;5)
  Loop (column;3;2)
    Pause ("coord: [row] [column]")
  EndLoop
EndLoop
```

The script provided as nested loop example in [Figure 57](#) displays: 'coord: 4 3', 'coord: 4 2', 'coord: 5 3', and 'coord 5 2'.

Loop script verdict

The loop generates some special lines in the verdict file. The added lines help the user to follow the execution.

The script shown in [Figure 58](#) generates the verdict shown in [Figure 59](#):

Figure 58. Loop script verdict example

```
Loop (FREQ, 13, 15)
EndLoop
```

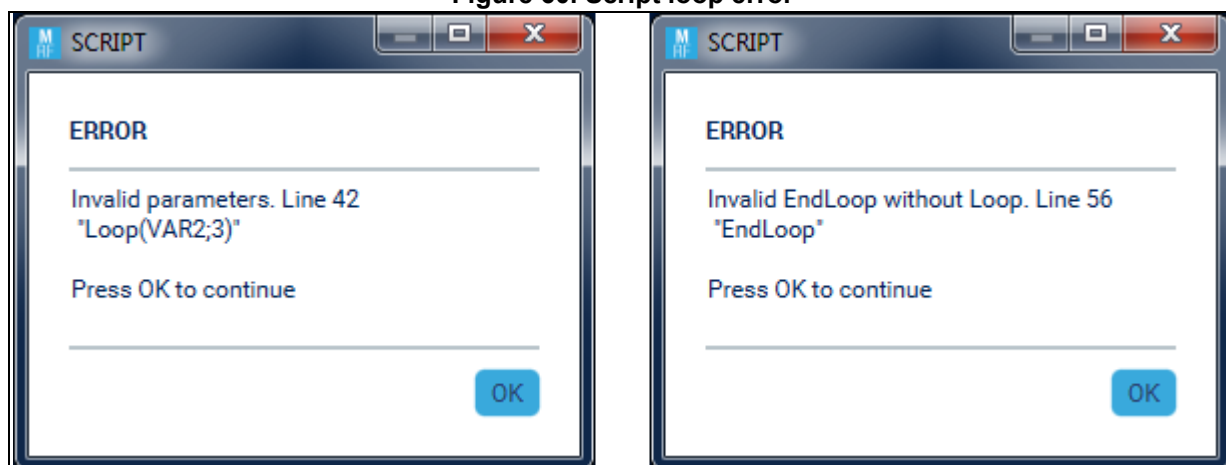
Figure 59. Loop script verdict display

```
-- Loop Start (FREQ=13) --
-- Loop (FREQ=14) --
-- Loop (FREQ=15) --
-- Loop End (FREQ) --
```

The beginning and end of loop are indicated, and the counter value is also inserted in decimal for each iteration.

In case of error in an instruction, a pop up warns the user when the script is executed, and the line is skipped. If a Loop instruction is missing or invalid, the EndLoop generates the 'Invalid EndLoop without Loop' warning message.

Figure 60. Script loop error



3.5 OTA transfer

3.5.1 OTA presentation

Over-the-air (OTA) transfer executes the transfer of data from a device to a remote device without cable. The data are applicative data, like user configuration, picture, music or firmware. STM32CubeMonitor-RF provides a transfer function from the computer to the remote device over BLE.

In this section, the computer or device sending the data is named 'Source device'.

The data are transferred by the source device and the OTA loader to the address requested by the user.

The implementation example does not include security in the transfer process. It is expected that the user changes his loader or application to perform the security verification based on the customer requirements.

The OTA process is described in *Over-the-air application and wireless firmware update for STM32WB Series microcontrollers* application note (AN5247), available on www.st.com. Read this documents for the details of device configuration and OTA procedure. In this user manual, there is only a summary of the procedure, to explain how to use the tool. Read the application note to get detailed information about the target software and the BLE services used.

OTA loader

The OTA loader is the first application started at boot or reboot. OTA loader checks the boot conditions, and if Flash is empty.

When the bootloader start in OTA mode, the loader creates an OTA service and some characteristics required to perform the OTA transfer. This attributes are used to perform the transfer.

The loader is fitting in the first 6 sectors of the Flash memory, so the block at address 0x6000 is free and used to upload the user data.

Table 1. OTA loader address table

Flash address:	Flash content:
+0x0000	OTA bootloader
+0x1000	OTA bootloader
+0x...	OTA bootloader...
+0x6000	Free for user data
+0x7000	User application
+0x8000	User application
+0x....	User application...

- In the STM32WB55 sample code, the binary is stored at address 0x7000, and bootloader starts at this address after upload.

3.5.2 OTA procedure

The OTA procedure occurs between one source device and the target device. The process is based on operations:

1. Activate the OTA mode on the target device
2. Connect in OTA mode and transfer data.

Activation of OTA mode

The computer sends an indication to target device to reboot in OTA mode, with the information about download. The target restarts in OTA mode and erases the flash area required for the transfer.

Connect in OTA mode and transfer data

The source device first connects to the OTA loader and discover the details of service and characteristics to be able to transfer the data. Then the sequence is:

1. Configure the target device to send indication to the source device
2. Write in the target device the command to initiate the procedure, with exact storage address
3. Write each block of data. The blocks of data are 20-byte long, and the binary must be transferred in many blocks.
4. At the end of last block, write the confirmation that all blocks have been sent
5. The source device waits for the reboot confirmation from the target

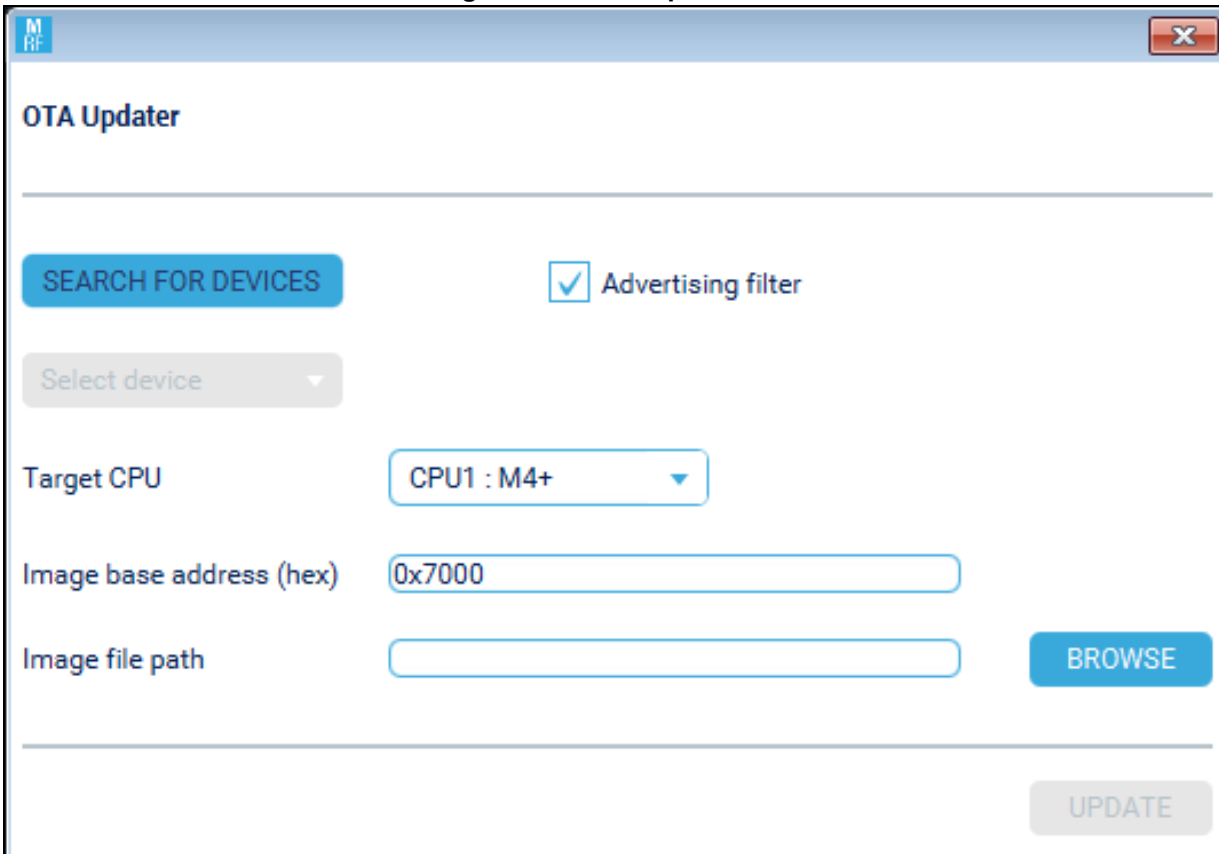
3.5.3 Use the tool to perform OTA update

The OTA function is available in the device menu in the menu bar. Click on device and then click on OTA updater.

Search procedure

The first operation is to find the target device. The tool needs to perform a scan of BLE devices and list all the devices with OTA capabilities.

Figure 61. Search procedure



The screenshot shows a software window titled "OTA Updater" with a standard Windows-style title bar (minimize, maximize, close buttons). The interface is clean and modern, with a light blue header bar. Below the header, there's a horizontal line. The main area contains several interactive elements: a blue button labeled "SEARCH FOR DEVICES", a checkbox labeled "Advertising filter" which is checked, a grey button labeled "Select device" with a dropdown arrow, a "Target CPU" label next to a dropdown menu showing "CPU1 : M4+", an "Image base address (hex)" label next to a text input field containing "0x7000", an "Image file path" label next to an empty text input field, a blue "BROWSE" button, and a grey "UPDATE" button at the bottom right. The window has a thin blue border.

The tool provides an advertising filter to refine the search procedure with advertising message.

Table 2. Search filtering

Filter	Search method	Comment
No filter	Scans all BLE devices and provides the list.	Some devices listed are not compatible for OTA.
Advertising filter	Scans all BLE devices, and provides a list of devices with ST OTA information.	Gives only the list of compatible devices.

To start search, click on the 'SEARCH FOR DEVICES' button.

Figure 62. Scanning

The screenshot shows the 'OTA Updater' application window. At the top left is the 'M RF' logo. The main area contains a 'SEARCH FOR DEVICES' button on the left and a checked 'Advertising filter' checkbox on the right. Below the search button is a 'Select device' dropdown menu. In the center, the text 'Scanning....' is displayed in pink, accompanied by a circular progress indicator on the right. Further down, there are three input fields: 'Target CPU' with a dropdown set to 'CPU1 : M4+', 'Image base address (hex)' with the value '0x7000', and 'Image file path' which is empty. To the right of the 'Image file path' field is a 'BROWSE' button. At the bottom right of the window is a disabled 'UPDATE' button.

The search procedure starts.

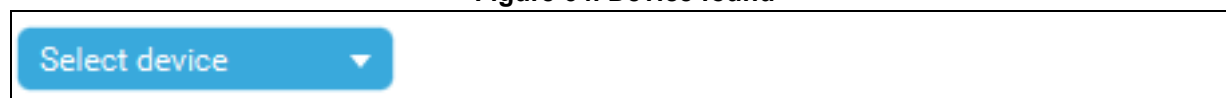
If no target device is found, the tool indicates 'No device found'.

Figure 63. No device found

This screenshot shows the same 'OTA Updater' window after the search has completed without finding any devices. The 'SEARCH FOR DEVICES' button and the checked 'Advertising filter' checkbox remain. The 'Select device' dropdown is still present. The central status text has changed to 'No device found' in pink. The 'Image base address (hex)' field still shows '0x7000', and the 'Image file path' field is still empty with its 'BROWSE' button. The 'UPDATE' button at the bottom right remains disabled.

If a candidate device is found, the select device box changes to blue.

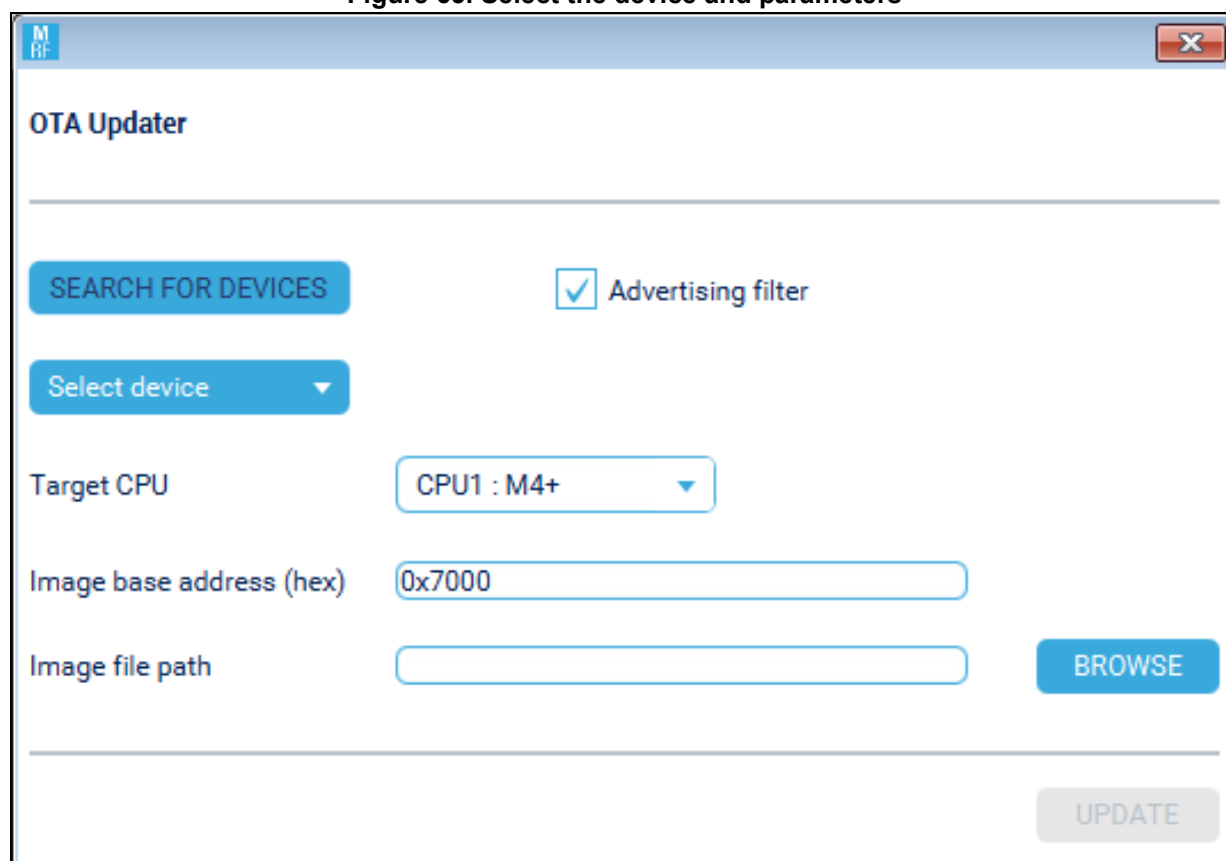
Figure 64. Device found



Select the device and parameters

After the search procedure, if one or more devices are found, the user selects the device with the picklist 'Select device'.

Figure 65. Select the device and parameters



The picklist displays the list of boards found:

- For device with BLE characteristic:
BLE address - Device name - OTA enabled
- For device already in OTA mode:
BLE address - Device name - OTA loader

Select the firmware target:

- For user data or user application firmware, select the "CPU1: M4+"
- For Wireless stack, select the "CPU2 : M0"

The image base address is the place where the binary file must be stored on the target device. It is hexadecimal value, and must be multiple of 0x1000 to match with Flash sector. For wireless stack, the address is the temporary location in CPU1 User part area.

The image file path is the binary file to load. Enter the path in the box, or use BROWSE button to select the file to download.

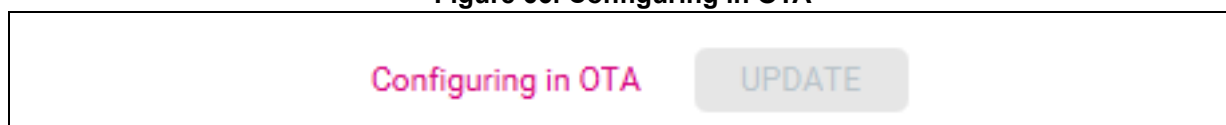
The configuration is finished, software is ready to start the update procedure.

Flashing the remote device

Press the 'UPDATE' button to start flashing the target device.

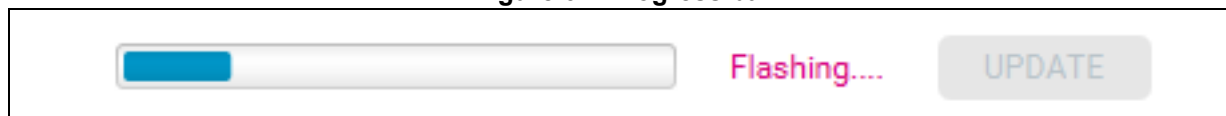
1. First step: if the selected device has an OTA characteristic, the tool first restarts the device in OTA bootloader mode. The indication 'Configuring in OTA' is displayed

Figure 66. Configuring in OTA



2. Second step, the transfer process to the OTA bootloader is performed. The data are transferred by block of 20 bytes. In order to avoid overload of log windows, the log information related to block transfer are not displayed. Only the flow control event and errors are displayed.
3. The progress is indicated by a progress bar.

Figure 67. Progress bar



At the end of the update process:

4. The target device reboots.
5. The user closes the OTA panel, or starts a new search to flash another device.

3.6 Beacon

3.6.1 Beacon presentation

A beacon is an active device discoverable by other devices.

The beacon device only sends information by advertisement and do not receive any data.

The data shared by beacon are very small. This data are received by a connected device and application on the device are notified of beacon presence. The application uses the cloud to get more information and act accordingly.

Figure 68. Beacon presentation



When an application is informed of beacon proximity, it uses the beacon identification to request web server more information about the beacon. Application gets information related to geographical position of beacon, or action to perform, like displaying commercial ads, or start an interactive application.

Figure 69. Beacon usage



Many organizations have created beacons. The specifications from Apple® and Google® are frequently used:

- iBeacon: this is Apple format. The beacon broadcasts a fix content, allowing to identify the beacon.
- Eddystone UID: this is defined by Google. The beacon transmits fixed content (UID), which is a unique ID, referenced in Google database to interact with applications.
- Eddystone URL: another Google format, provide a short URL, to use for 'Physical web'.
- Eddystone TLM: an additional beacon advertising information providing beacon information (battery status, temperature).
- Eddystone EID: similar to UID, but broadcasts encrypted data to provide better security.

3.6.2 Beacon configuration methods:

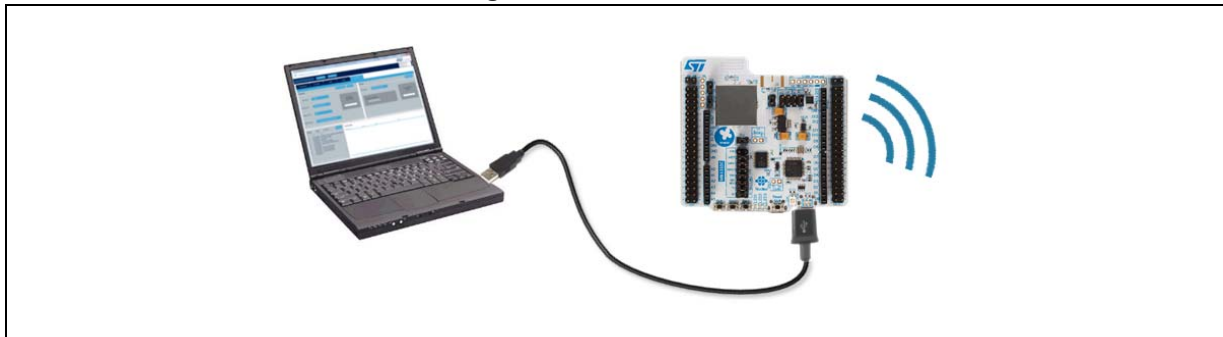
STM32CubeMonitor-RF is used to generate and configure beacon. Different methods have been defined to accommodate the user's needs. This chapter describes the different methods supported.

Online beacon

In online mode, the tool is directly configuring the main device in a beacon. The tool sends ACI commands to configure the boards in advertising mode and configure the content of advertising packet. The main device acts as a beacon as long as it is powered.

The main advantage of this method is to quickly configure a beacon with a board in transparent mode. The drawback is that the configuration is lost when board is reset or powered off.

Figure 70. Online beacon

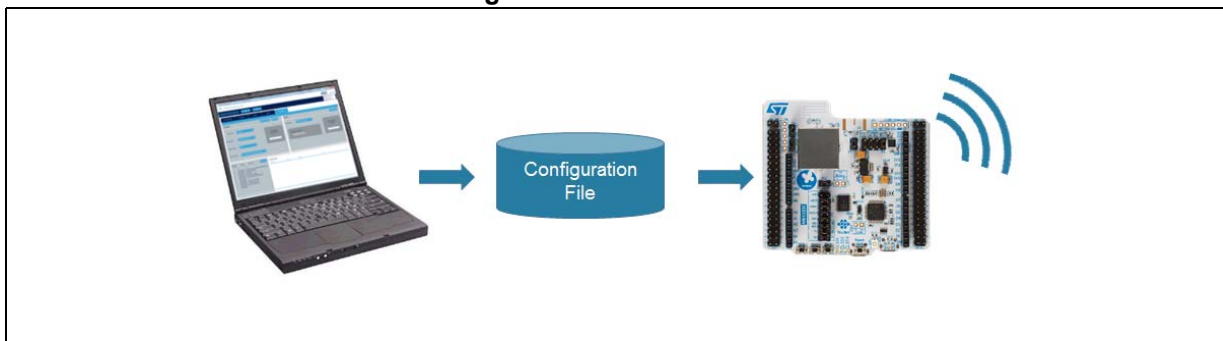


Offline beacon

The offline beacon mode is used to prepare the configuration of a board not directly connected to STM32CubeRF-Monitor. The parameters to configure the beacon are stored in a data file. The file is used to configure a target board running the beacon example firmware. The file must be stored in target Flash at address 0x6000. The beacon firmware reads the data and configures the advertising block accordingly. Details of configuration file is described in [Table 4: Beacon configuration format](#).

The interest of the method is to have independent beacon, which is useful if the user needs many beacon boards at the same time. It is possible to keep many configuration files to change configuration quickly. The drawback is that configuration file must be transferred manually in the target device, so it is less flexible than 'Online' mode.

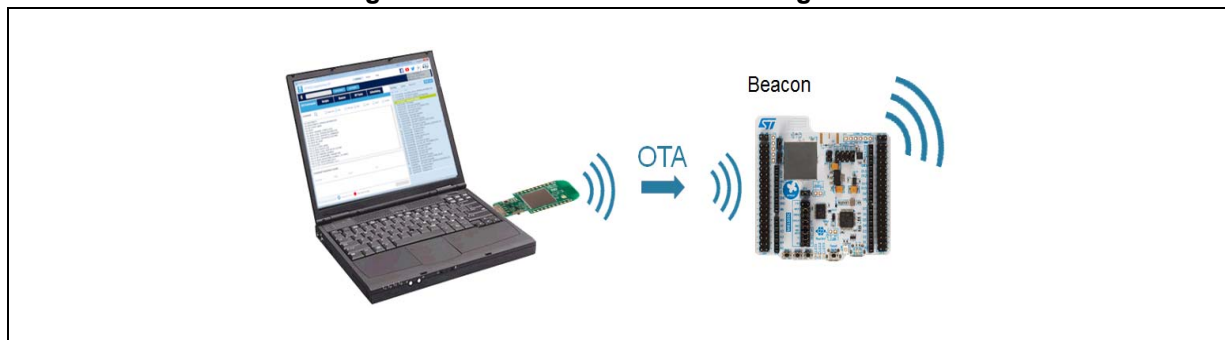
Figure 71. Offline beacon



Beacon over-the-air configuration

This method is similar to Offline mode, but the tool uses the OTA procedure to transfer the configuration file directly into the target Flash. A target board with a beacon demo firmware directly restarts after OTA with the updated parameters. (OTA needs to be enabled). The main device is used to transfer the file to the target device.

Figure 72. Beacon over-the-air configuration



Selecting the beacon mode

The selection of the configuration mode is the first action to prepare the beacon. The user must select the mode when he selects the beacon tab.

Figure 73. Selecting the beacon mode

ACI Commands	Scripts	Beacon	RF Tests	ACI Utilities
Configuration mode				
Select configuration mode				
<input checked="" type="radio"/> Directly configure device through wired connection (On line) <input type="radio"/> Generate configuration and update device through wired connection (Off line) <input type="radio"/> Generate configuration and update peer device over the air (OTA)				
SELECT CONFIGURATION MODE				

Select one of the three bullet and click on 'SELECT CONFIGURATION MODE'.

3.6.3 Configuration of beacon with STM32CubeMonitor-RF

To configure the beacon:

1. Select the configuration method.
2. Fill the beacon parameters, some are common for all beacons, others are specific for the beacon type.
3. Generate/transfer the configuration. Additional information may be required according to configuration method.

Common parameters

Some beacon parameters are common for all kinds of beacon. The common parameters are at the top of the beacon panel:

Figure 74. Common parameters

Parameters	
Reference TX power level (dBm)	<input type="text" value="-56"/>
PA Level	<input type="text" value="31 (+6dBm)"/>
Beacon Address	<input type="text" value="123456789AAA"/>
	<input checked="" type="checkbox"/> Public Address <input type="checkbox"/> Random Address

The first parameter is the 'Reference TX power level', and the second parameter is the real 'TX power level'.

In order to save batteries, the power level of beacon may be lowered, reducing consumption and visibility. Using high power extends the range of visibility, but drains more power. The power level needs to be defined by the user based on power source and beacon purpose.

The device detecting the beacon needs to estimate if the beacon is close or far. Unfortunately, the received power level is not sufficient to estimate the real distance:

- Some beacon may transmit with high power, while other are using low power.
- The design of beacon antenna may be more or less efficient.

The reference power information is added to help determine the distance. This is the power level received at one meter from the beacon. The application uses this value and the received strength to estimate the distance, independently of the real TX power used and the beacon characteristics.

The easiest solution to fill this parameter is to configure a beacon with the required Tx level, and then to measure the received level at 1 meter. Then beacon is reconfigured with the value measured at 1 meter in the 'Reference TX power level' field.

The second set of parameters is the beacon address. There are 3 possibilities:

- Set the address in the box and tick 'public address'. The address entered is used.
- Tick the 'random address' check box. A random address is used.
- If nothing is selected, the board default public address is used.

iBeacon parameters

Figure 75. iBeacon parameters

Select Type	<input type="text" value="iBeacon"/>
Company ID Code (hex)	<input type="text" value="0030"/>
Beacon UUID (hex)	<input type="text" value="D9B9EC1F392543D080A91E39D4CEA95C"/>
Beacon Major code (hex)	<input type="text" value="0010"/>
Beacon Minor code (hex)	<input type="text" value="0003"/>
<input type="button" value="Change mode"/> <input type="button" value="CONFIGURE"/>	

First, select the type: iBeacon (default choice)

The user must check the Apple web site for information about iBeacon structure and the condition to use iBeacon for his project: <https://developer.apple.com/ibeacon/>

More information is also available at <https://en.wikipedia.org/wiki/IBeacon>

The company code is a value based on BLE SIG group assigned values. For iBeacon, the Apple value is used: 0x004C. The assigned values are available on the SIG web site: <https://www.bluetooth.com/specifications/assigned-numbers/company-identifiers>

The beacon UUID is the unique identifier for a group of beacon. Apple explains how to define the identifier in the document 'Getting started with iBeacon' available at <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>.

The major and minor codes are defined by the user to identify logically different beacons sharing the same UUID.

When all parameters are updated, click on 'CONFIGURE'. The data are ready for transfer (see [Transfer the configuration](#)).

Eddystone UID parameters

Figure 76. Eddystone UID

The Eddystone UID parameters are the beacon UID, a 16-byte identifier, formed by:

- NameSpace, 10 bytes. Used to group some beacon in a logical pool. The way to generate the value is described by Google, see <https://github.com/google/eddystone/tree/master/eddystone-uid>
- Beacon instance, 6 bytes. Give a unique id inside the pool.

When a beacon is discovered on a smartphone, the UID value is not directly usable by the phone application. Google offers a cloud service to associate one or more data with a beacon. The smartphone application retrieves this information to perform the required actions.

The last option is Enable TLM tick box. When TLM is used, the beacon interleaves some status information inside the normal beacon advertisement. The TLM frame has information about battery level, temperature, the time beacon is on, and the number of frames transmitted. The TLM information is not known by the tool. So it must be managed directly by the firmware. As a consequence:

- The TLM option is not used for Online configuration mode.
- For Off Line and OTA mode, a bit is set in the configuration file. (see annex A)

Eddystone URL parameters

Figure 77. Eddystone URL

The Eddystone URL format is just sending an URL in the advertising message. In order to optimize space, start and end of URL may be compressed.

1. Select the URL prefix: the prefix is encoded in 1 byte in the advertising.
2. Fill the rest of the URL in the URL box, without prefix. The URL is parsed, and if the end of the URL is encodable, the tool encodes it. Long URL does not work, it is advised to use URL short service to get a short URL.

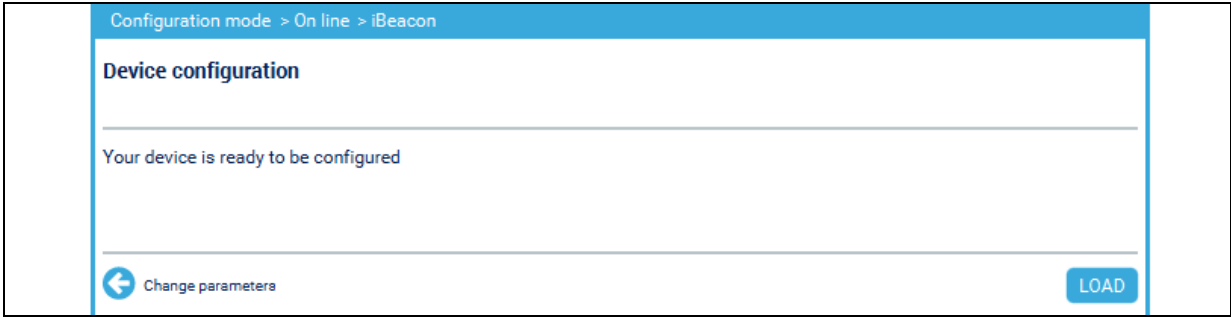
The TLM option is the same as UID beacon.

Transfer the configuration

The transfer depends on the selected configuration mode.

1. Online mode transfer configuration

Figure 78. Online mode transfer configuration

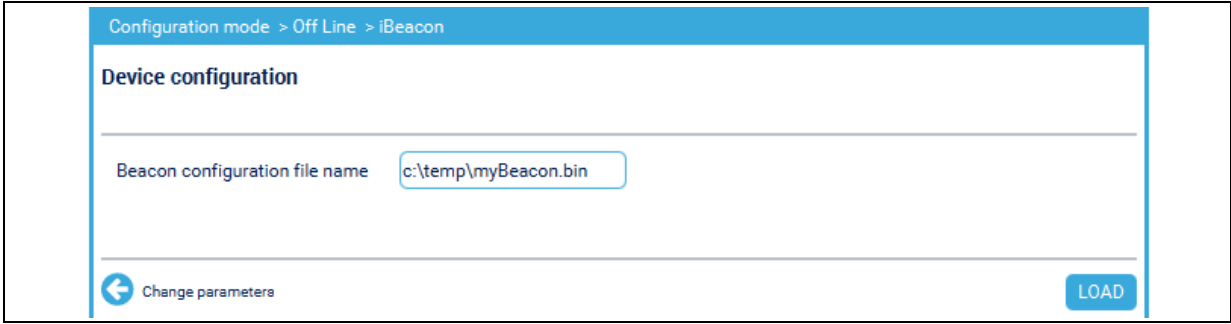


The screenshot shows a web interface for online mode transfer configuration. At the top, a blue header bar contains the text "Configuration mode > On line > iBeacon". Below this, the section is titled "Device configuration". A message states "Your device is ready to be configured". At the bottom left, there is a blue circular button with a left-pointing arrow and the text "Change parameters". At the bottom right, there is a blue rectangular button labeled "LOAD".

No extra parameters are required: just click on 'LOAD' and the main device is initialized and configured in beacon.

2. Offline mode transfer configuration

Figure 79. Offline mode transfer configuration



The screenshot shows a web interface for offline mode transfer configuration. At the top, a blue header bar contains the text "Configuration mode > Off Line > iBeacon". Below this, the section is titled "Device configuration". There is a label "Beacon configuration file name" followed by a text input field containing the value "c:\temp\myBeacon.bin". At the bottom left, there is a blue circular button with a left-pointing arrow and the text "Change parameters". At the bottom right, there is a blue rectangular button labeled "LOAD".

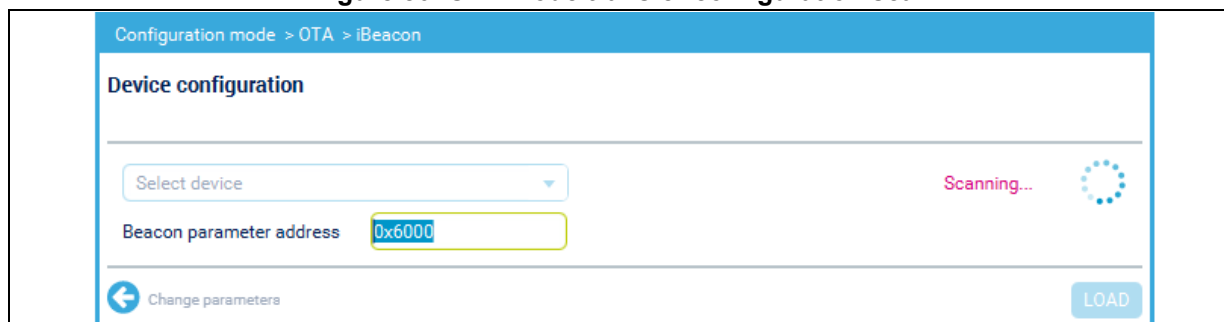
Indicate first the name of the file to create, including the path. If no path is provided, the file is stored in the tool directory. The file is then copied to the target device using a Flash programmer, or any other tool.

3. OTA mode scan transfer configuration:

a) Scan

As soon as the page is displayed, the tool asks the main device to search OTA capable devices in the area. 'Scanning' is displayed in the windows.

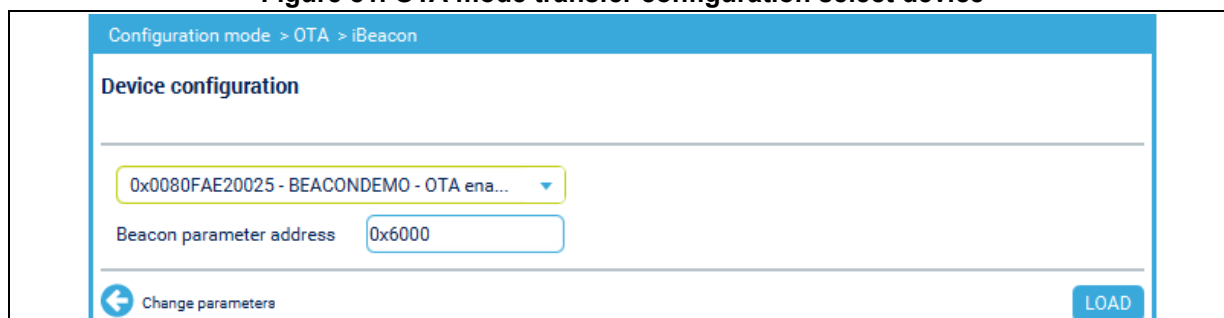
Figure 80. OTA mode transfer configuration scan



b) Select device

When the device is found, the list is updated, and the user selects the device to configure. The address to store the beacon data is 0x6000 in ST example firmware. The value is editable if another address is used.

Figure 81. OTA mode transfer configuration select device

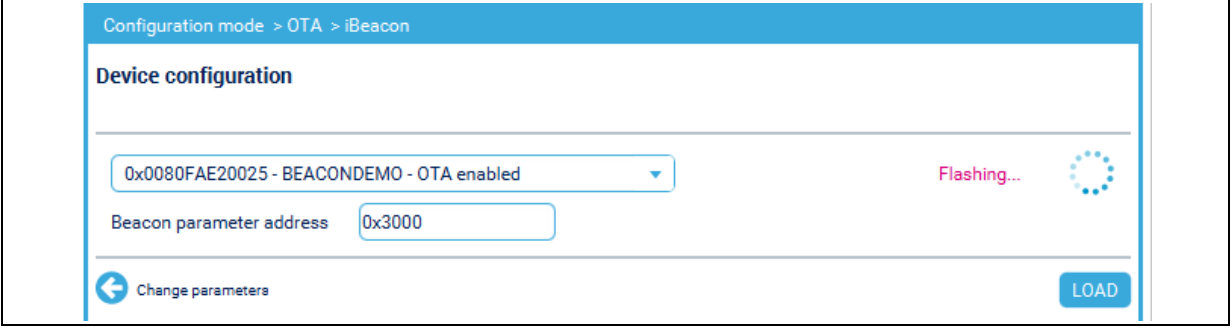


c) Load the file

Click on 'LOAD' and the tool flashes the binary data.

Most of the time, beacons are not connectible, and are moved in OTA loader by the user's action. If beacon is in OTA loader mode, data are transmitted directly.

If the device is OTA enabled, the tool changes it in OTA loader mode first, and then transfers the data.

Figure 82. OTA mode transfer configuration transmission

The screenshot shows a web-based configuration interface for an STM32 device in OTA mode. At the top, a blue header bar contains the text "Configuration mode > OTA > iBeacon". Below this, the section "Device configuration" is displayed. A dropdown menu shows the selected device as "0x0080FAE20025 - BEACONDEMO - OTA enabled". To the right of the dropdown, the text "Flashing..." is shown in pink, accompanied by a circular loading icon. Below the dropdown, there is a label "Beacon parameter address" followed by a text input field containing "0x3000". At the bottom left, there is a blue circular button with a left-pointing arrow and the text "Change parameters". At the bottom right, there is a blue rectangular button labeled "LOAD".

After transfer, application is restarted, and beacon is configured.

3.7 ACI utilities

The ACI Utilities panel is used to configure the device to perform either the advertising signal or to discover remote devices and explore it services and characteristics.

Figure 83. ACI Utilities panel

ACI CommandsScriptsBeaconRF TestsACI Utilities

Init

Initialization parameters

☐ Discover remote services

☐ Advertising

Address

0x112233445566

Power

31 (+6dBm)

Name

STM32WB

Discoverability mode

General discoverable

Adv type

0x00 - ADV_IND (Connectable undirected advertising)

Advertising channel map

☒ CH37☐ CH38☐ CH39

Own address type

0x00 - Public Device Address

Advertising interval (20 to 10240 ms)

1280Min1280Max

Slave connection interval (7.5 to 4000 ms)

MinMaxUse empty value for non specific Min/Max

SCAN

START ADVERTISING

The first action is to select to discover remote services, to manage advertising, or both, by clicking appropriate checkbox.

Figure 84. Select checkbox

☐ Discover remote services

☐ Advertising

3.7.1 Remote services discovering

The remote services discovery performs a scan of the remote devices in the area.



Figure 85. Scan parameters

ACI Commands	Scripts	Beacon	RF Tests	ACI Utilities
Init				
Initialization parameters				
<input checked="" type="checkbox"/> Discover remote services <input type="checkbox"/> Advertising				
Address		0x112233445566		
Power		31 (+6dBm)		
Name		STM32WB		
Discoverability mode		General discoverable		
Adv type		0x00 - ADV_IND (Connectable undirected advertising)		
Advertising channel map		<input checked="" type="checkbox"/> CH37 <input type="checkbox"/> CH38 <input type="checkbox"/> CH39		
Own address type		0x00 - Public Device Address		
Advertising interval (20 to 10240 ms)		<input type="text" value="1280"/> Min <input type="text" value="1280"/> Max		
Slave connection interval (7.5 to 4000 ms)		<input type="text"/> Min <input type="text"/> Max Use empty value for non specific Min/Max		
				<input type="button" value="SCAN"/> <input type="button" value="START ADVERTISING"/>

To perform a scan of the available devices:

1. Enter the device address
2. Select the power level with the picklist
3. Enter the device name
4. Click on "SCAN" button to start the discovery

The search procedure starts, and it is possible to stop it using "STOP" button.

Figure 86. Scanning



If no remote device is found, the tool indicates “No device found”. Otherwise, the user chooses one of the device found in the “Select Device” box.

Figure 87. Select device



At this stage, the user performs another scan procedure upon request.

- Click on the back button

Figure 88. Back



- Click on the “Init” in the top bar

Figure 89. Init



Or connect to the selected remote device, by clicking on “CONNECT” button.

Figure 90. Connecting



If connection fails, an error is displayed.

Figure 91. Connection error

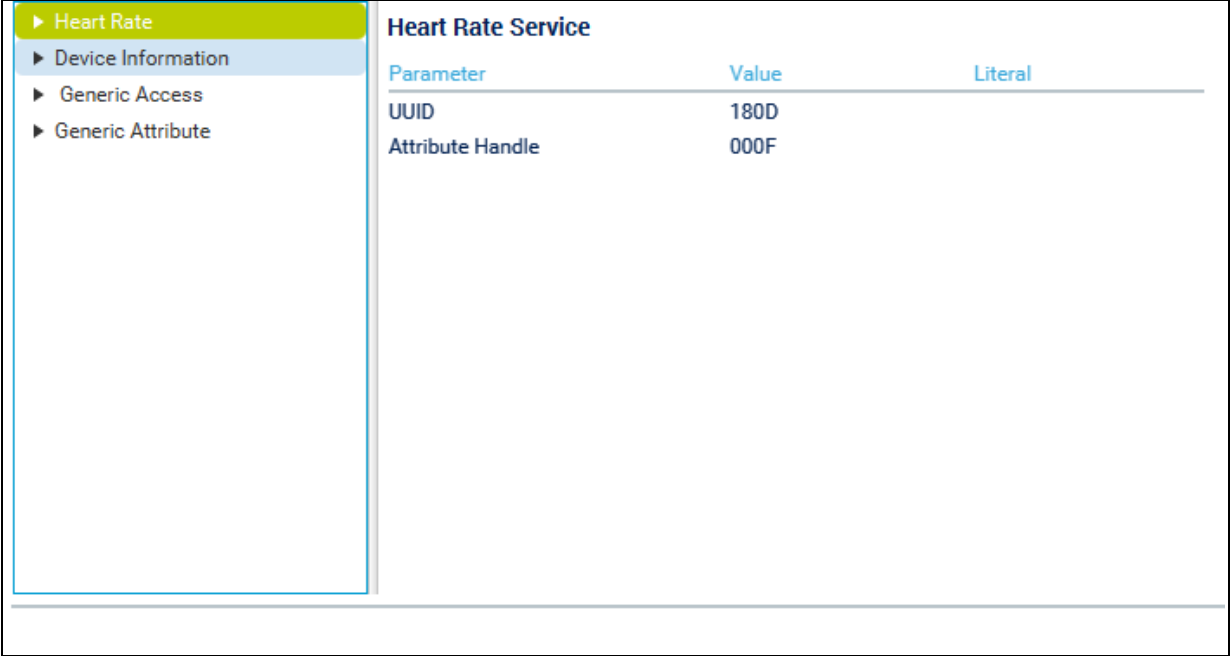


Once connected, the connect icon appear in blue and the list of available services is proposed.

Figure 92. Connected icon



Figure 93. Services list



When the user selects a service, its details are displayed. Clicking on the arrow displays the characteristics linked to the above service.

Figure 94. Characteristics list

<div>▼ Heart Rate</div> <div>Heart Rate Control Point</div> <div>Body Sensor Location</div> <div>Heart Rate Measurement</div> <div>▶ Device Information</div> <div>▶ Generic Access</div> <div>▶ Generic Attribute</div>	Heart Rate Service		
	Parameter	Value	Literal
	UUID	180D	
	Attribute Handle	000F	

The user can select a characteristic and, depending on each of them, can read or write a value and be notified of a value changed. Note that read and write long characteristic are not supported neither authenticated signed write.

To read a value, the user clicks on “Read button”.

Figure 95. Read value

<div>▼ Heart Rate</div> <div>Heart Rate Control Point</div> <div>Body Sensor Location</div> <div>Heart Rate Measurement</div> <div>▶ Device Information</div> <div>▶ Generic Access</div> <div>▶ Generic Attribute</div>	Body Sensor Location Characteristic		
	Parameter	Value	Literal
	UUID	2A38	
	Handle	0013	
	Properties	02	Read
	Value handle	0014	
	Value length	8bit	
	Value	0x04	

READ

To write a value, user enters the new value and click on “WRITE” button.

Figure 96. Write value

Heart Rate Control Point Characteristic	
Parameter	Value
UUID	2A39
Handle	0015
Properties	08
Value handle	0016
Value length	8bit
Value	<input type="text" value="123456"/>

WRITE

There are two ways to be informed on a value change, either via indicate method or via notification depending on the method property supported by the remote device.

To receive indication upon value change, the user can click on “INDICATE” button.

Figure 97. Indicate value changed

Service Changed Characteristic	
Parameter	Value
UUID	2A05
Handle	0002
Properties	20
Value handle	0003
Value length	uint16
Value	
Client Characteristic Configuration UUID	2902
Client Characteristic Configuration handle	0004

INDICATE

To receive notification upon value change, the user can click on “NOTIFY” button.

Figure 98. Notify value changed

Parameter	Value	Literal
UUID	2A37	
Handle	0010	
Properties	10	Notify
Value handle	0011	
Value length	8bit	
Value		
Client Characteristic Configuration UUID	2902	
Client Characteristic Configuration handle	0012	

Upon each change, a notification (resp. indication) is received and the new value is displayed. The user can be informed on multiple characteristics value changes the same time. To stop notification (resp. indication), the user can click on “UN-NOTIFY” button (resp. “UN-INDICATE”).

On disconnection, all registered notifications are removed.

Figure 99. Notifying

No.	Time	Type
673	15:57:07.874	Vendor Specific Event
674	15:57:07.875	Vendor Specific Event
675	15:57:07.875	Vendor Specific Event
...	15:57:08....	ACL_GATT_DISC_ALL_CHAR_OF_S...
677	15:57:08.416	Command Status
678	15:57:08.484	Vendor Specific Event
679	15:57:08.542	Vendor Specific Event
680	15:57:08.602	Vendor Specific Event
681	15:57:08.663	Vendor Specific Event
682	15:57:08.663	Vendor Specific Event
6...	15:57:09.0...	ACL_GATT_DISC_ALL_CHAR_DE...
684	15:57:09.021	Command Status
685	15:57:09.082	Vendor Specific Event
686	15:57:09.143	Vendor Specific Event
687	15:57:09.203	Vendor Specific Event
688	15:57:09.283	Vendor Specific Event
689	15:57:09.342	Vendor Specific Event
690	15:57:09.402	Vendor Specific Event
691	15:57:09.402	Vendor Specific Event
692	16:01:54.986	ACL_GATT_READ_CHAR_VAL
693	16:01:54.989	Command Status
694	16:01:55.060	Vendor Specific Event
695	16:01:55.060	Vendor Specific Event
696	16:02:37.313	ACL_GATT_WRITE_CHAR_VALUE
697	16:02:37.317	Command Status
698	16:02:37.380	Vendor Specific Event
699	16:02:38.221	Vendor Specific Event
700	16:02:39.081	Vendor Specific Event
701	16:02:39.921	Vendor Specific Event
702	16:02:40.761	Vendor Specific Event
703	16:02:41.621	Vendor Specific Event
704	16:02:42.461	Vendor Specific Event
705	16:02:43.321	Vendor Specific Event
706	16:02:44.161	Vendor Specific Event
707	16:02:45.021	Vendor Specific Event
708	16:02:45.861	Vendor Specific Event

3.7.2 Advertising

Figure 100. Advertising parameters



ACI Commands	Scripts	Beacon	RF Tests	ACI Utilities
Init				
Initialization parameters				
<div> <input type="checkbox"/> Discover remote services <input checked="" type="checkbox"/> Advertising </div>				
Address		0x112233445566		
Power		31 (+6dBm)		
Name		STM32WB		
Discoverability mode		General discoverable		
Adv type		0x00 - ADV_IND (Connectable undirected advertising)		
Advertising channel map		<input checked="" type="checkbox"/> CH37 <input type="checkbox"/> CH38 <input type="checkbox"/> CH39		
Own address type		0x00 - Public Device Address		
Advertising interval (20 to 10240 ms)		1280 Min 1280 Max		
Slave connection interval (7.5 to 4000 ms)		Min Max Use empty value for non specific Min/Max		
				<div> SCAN START ADVERTISING </div>

To activate advertising mode:

1. Enter the device address
2. Select the power level with the picklist
3. Enter the device name
4. Select the advertising type with the picklist
5. Select at least one channel from 37, 38 and 39.
6. Enter advertising interval
7. Enter optional slave connection interval
8. Click on "START ADVERTISING" button to start the procedure



The search procedure starts, advertising icon appear in blue, it is possible to stop it using "STOP ADVERTISING" button.

Figure 101. Advertising

ACI Commands	Scripts	Beacon	RF Tests	ACI Utilities
Init				
Initialization parameters  				
<input type="checkbox"/> Discover remote services <input checked="" type="checkbox"/> Advertising				
Address		<input type="text" value="0x112233445566"/>		
Power		<input type="text" value="31 (+6dBm)"/>		
Name		<input type="text" value="STM32WB"/>		
Discoverability mode		<input type="text" value="General discoverable"/>		
Adv type		<input type="text" value="0x00 - ADV_IND (Connectable undirected advertising)"/>		
Advertising channel map		<input checked="" type="checkbox"/> CH37 <input type="checkbox"/> CH38 <input type="checkbox"/> CH39		
Own address type		<input type="text" value="0x00 - Public Device Address"/>		
Advertising interval (20 to 10240 ms)		<input type="text" value="1280"/> Min <input type="text" value="1280"/> Max		
Slave connection interval (7.5 to 4000 ms)		<input type="text"/> Min <input type="text"/> Max Use empty value for non specific Min/Max		
<div>SCAN</div> <div>STOP ADVERTISING</div>				

The connect icon may appear in blue if a remote device connects. In that case, advertising stops.

Figure 102. Connected

Initialization parameters  

4 OpenThread mode

4.1 Presentation

4.1.1 Panel

The OpenThread main panel is organized with two tabs, 'Commands' and 'Scripts'.

Figure 103. OpenThread - Command tab

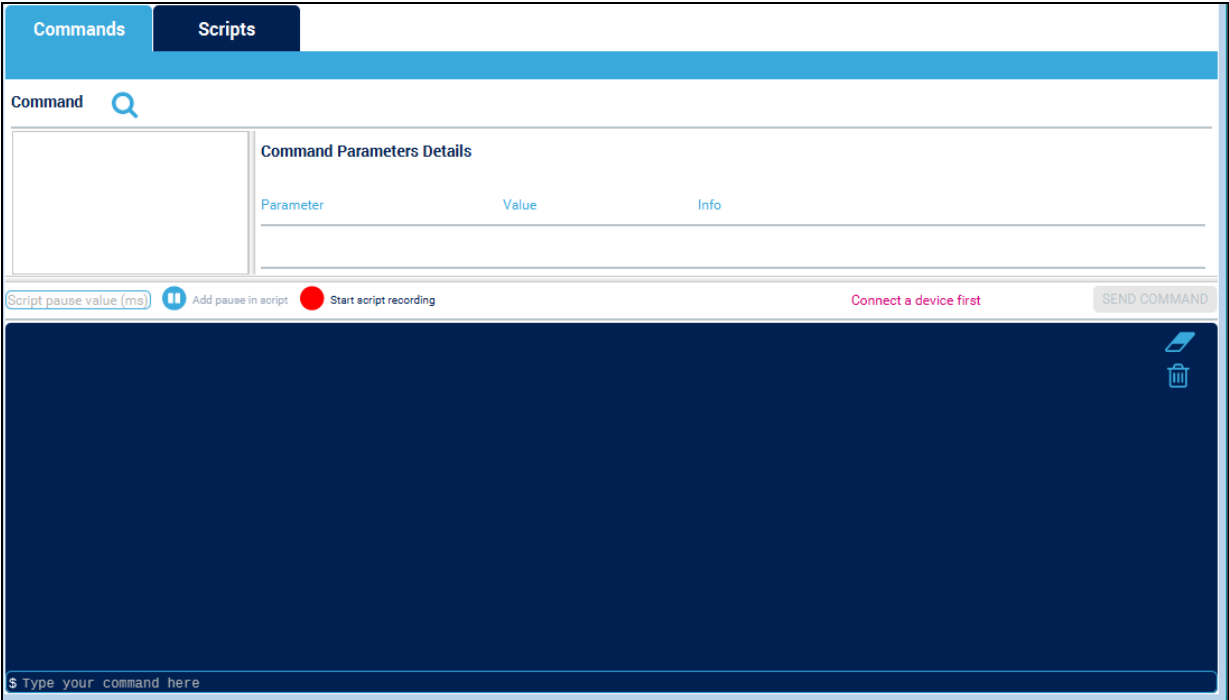
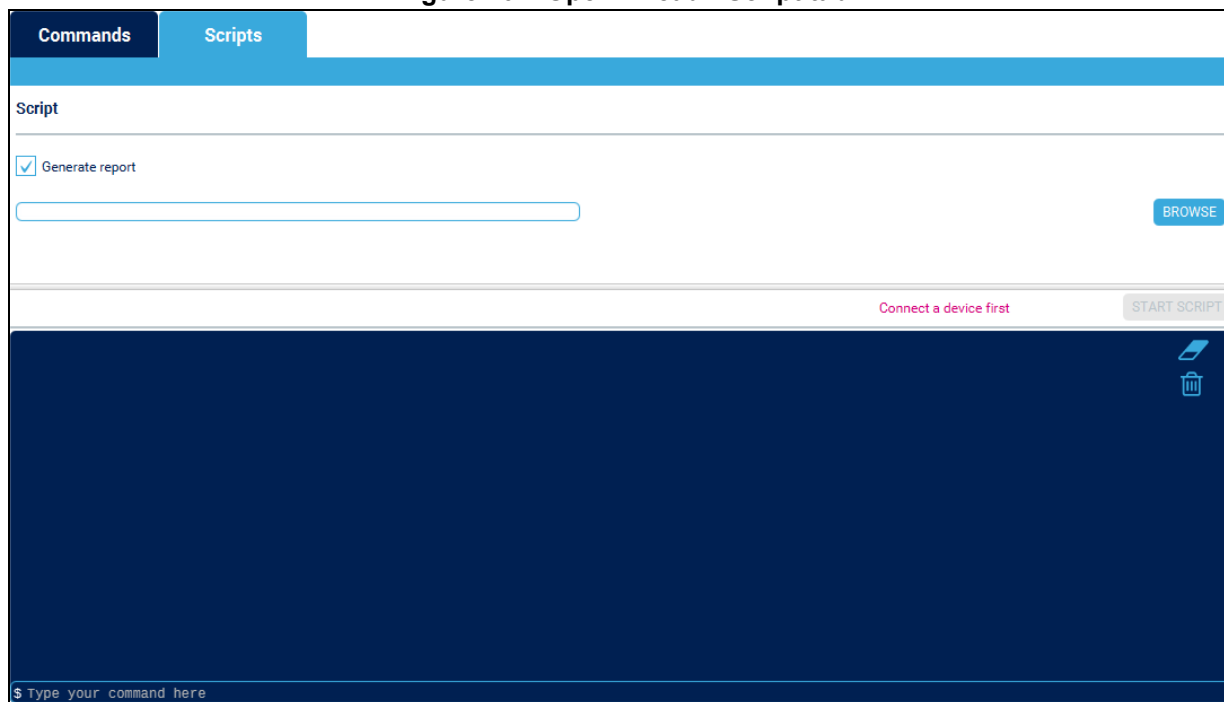
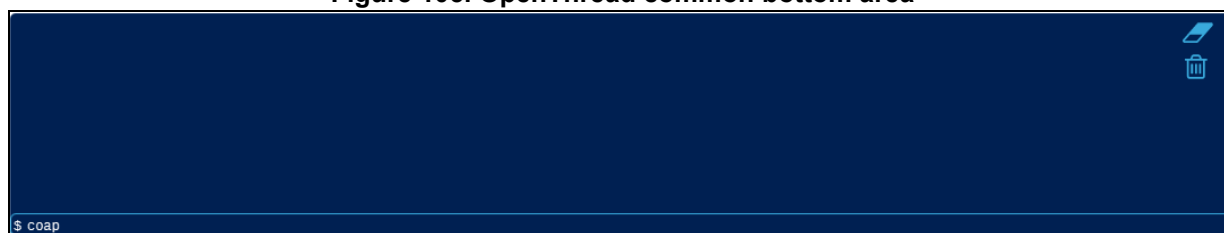


Figure 104. OpenThread - Script tab



Those 2 tabs have one common bottom area, the terminal area.

Figure 105. OpenThread common bottom area



The terminal area is used to show the messages exchanged between application and target. We can see the commands sent to target and the responses received from target. Those messages can be cleared with the rubber icon.

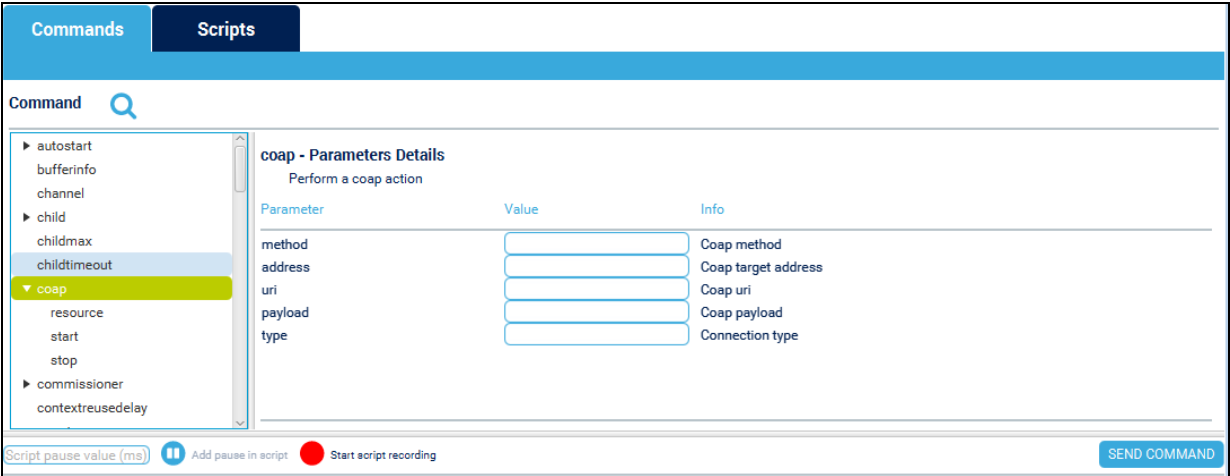
The bottom line with \$ character is a command line, the user types the command with the parameters and press enter to send command. The command sent with this line are recorded in history file and can be recalled with up and down arrows. This history can be deleted with the trash icon.

One other way is using the commands list and parameters area to fill the line, then the user can modify the line and send command with enter key. The commands list and parameters area are described in chapter Commands tab.

4.2 Commands tab

This tab is dedicated to the OT commands and parameters. The top area gives access to the commands list and parameters. Some commands can be used to read and send values, others are only commands sent to the OpenThread stack.

Figure 106. OpenThread command tab - top area



For commands used to send data, the 'SEND COMMAND' button sends the command with parameters to the target.

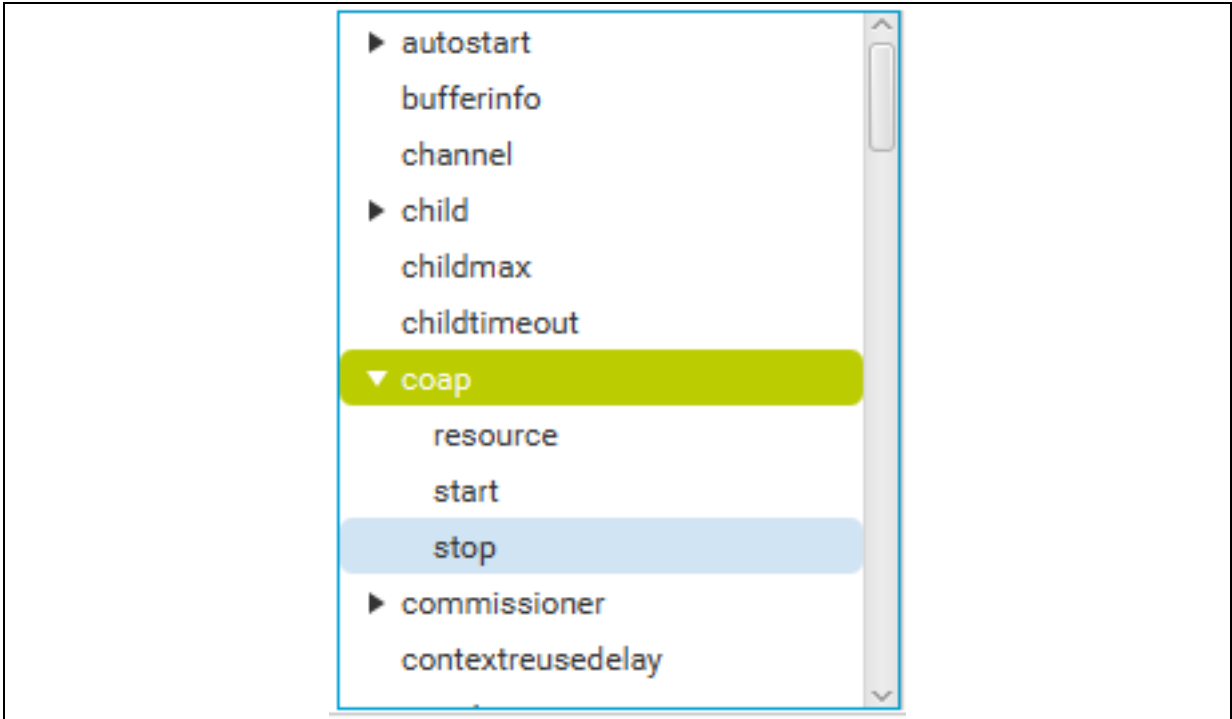
For commands able to read information, two buttons are available: 'READ', 'SEND'. The read button sends the command without parameters to read the value, the 'SEND COMMAND' button sends the command with parameters to the target.

Figure 107. Script buttons



The buttons 'Start script recording' and 'Add pause in script' allow saving a script. This part is described in [Section 4.3: OpenThread scripts tab](#).

Figure 108. Command list



The command list is arranged in alphabetical order, and accessible from tree, for example below coap command we find coap resource, coap start and coap stop commands.

Figure 109. Command details

coap - Parameters Details

Perform a coap action

Parameter	Value	Info
method	<input type="text"/>	Coap method
address	<input type="text"/>	Coap target address
uri	<input type="text"/>	Coap uri
payload	<input type="text"/>	Coap payload
type	<input type="text"/>	Connection type

The command name and the definition are in the upper part of the command details area. Below one table of parameters with the parameter name, there is one writable field to define the value, and information concerning this parameter.

4.3 OpenThread scripts tab

The OpenThread scripts tab is used for launching script stored in text file.

Figure 110. Scripts tab

Commands Scripts

Script

☒ Generate report

V:\MMS_MCD_TOOLS\05_BLE_Radio_Control_Specific\7-Sprint Release\00 -test files\OT_No

BROWSE

START SCRIPT

The scripts use the same syntax as BLE scripts. The OpenThread specificities are described in this chapter. Consult the BLE script description in [Section 3.4: Scripts on page 38](#) for general information.

4.3.1 OpenThread script example

Figure 111. Sample script

```
#STM32CubeMonitor-RF sample script
# OpenThread Ping Node Script

#Pause commande
Pause("Ready to start the test")

#Send reset command:
Send(reset)

#Set channel
Send(channel 11)

#Set the PAN ID:
Send(panid 0x1234)

#Bring up the IPv6 interface:
Send(ifconfig up)

#Start Thread protocol operation:
Send(thread start)

#Wait for a few seconds and verify that the device has become a
Thread Leader:
wait(5000)

#check state
Send(state)

#ipaddr
Send(ipaddr)
```

4.3.2 List of script commands

The OpenThread scripts use the same commands as BLE, but the Send command is modified to send Thread commands.

The OpenThread commands are sent with the instruction 'Send'

```
'Send(OPENTHREAD_CMD_NAME Parameter1Value Parameter2Value) '
```

The part inside brackets is the command line to send.

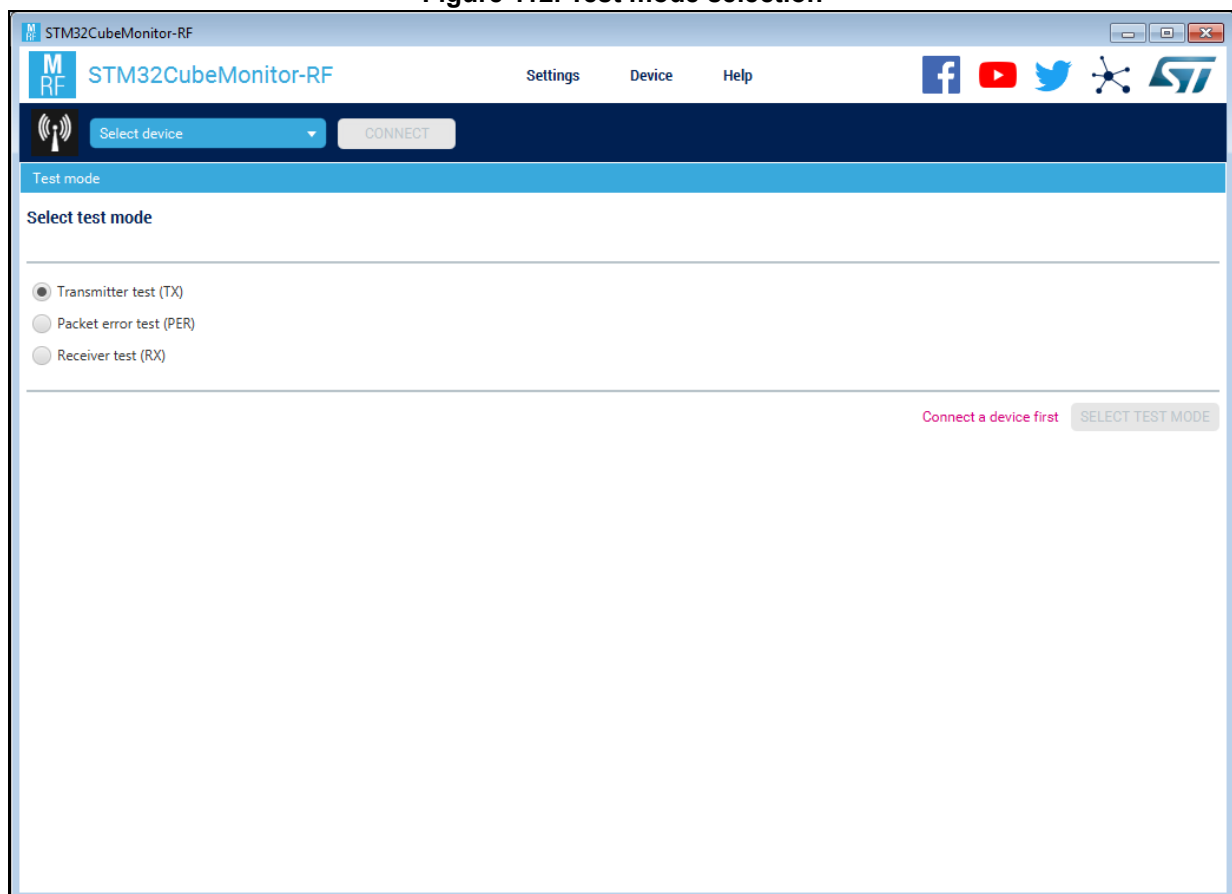
5 802.15.4 RF test mode

5.1 Presentation

The RF tests panel performs the 802.15.4 radio-frequency tests on the main device. Three test modes are available: Transmitter (TX), Receiver (RX) and Packet error rate (PER):

- The TX test sets the device in emission (TX continuous).
- The RX test set the device in reception.
- The PER test set the device in reception and one additional device is used as packet generator.

Figure 112. Test mode selection



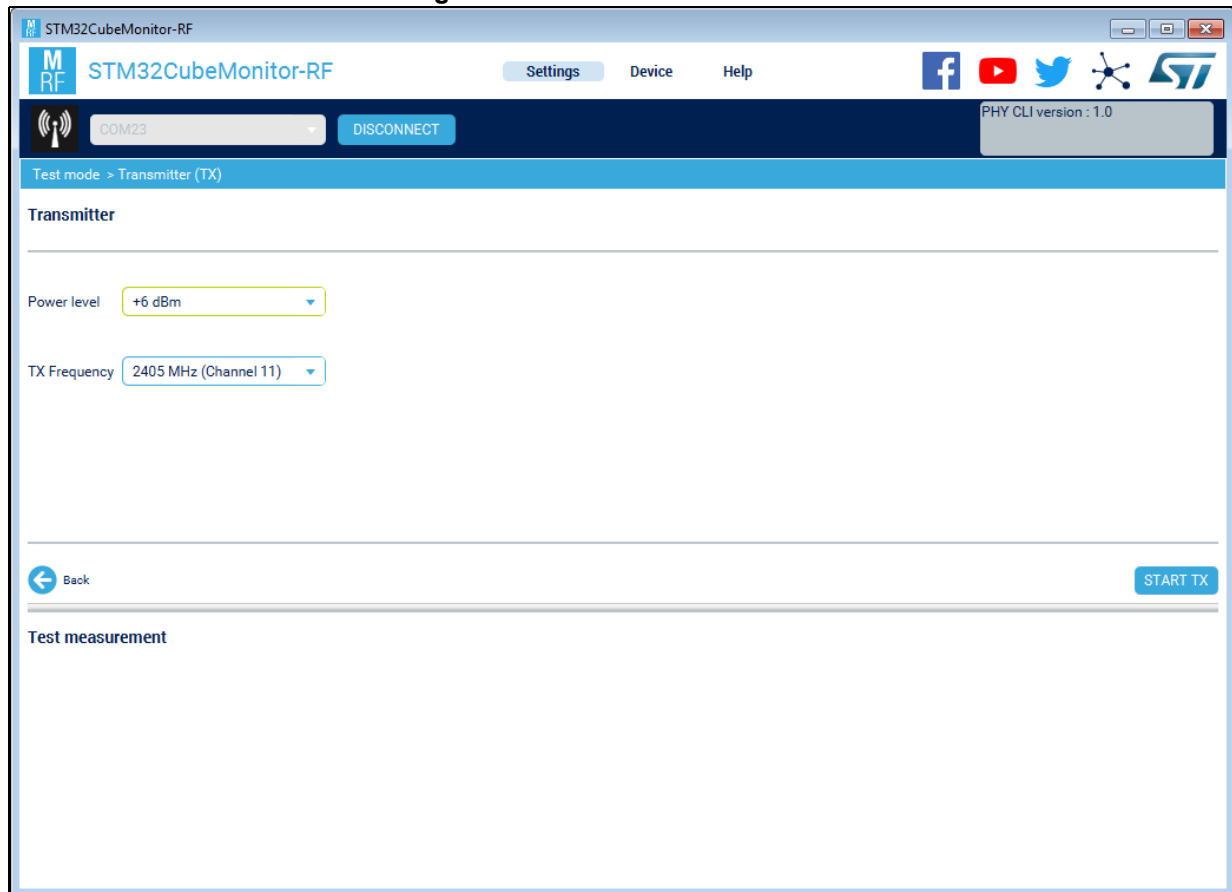
The user selects the mode by checking the radio button and press "SELECT TEST MODE" key to switch on new panel.

To change the mode, it is necessary to come back to this panel. There are one "Back" key and one breadcrumb link in each test panel to come back to this panel "Test mode selection".

5.2 Transmitter (TX) test mode

This test mode configures the 802.15.4 device in emission. This emission is continuous (without packet) at one power level and one frequency.

Figure 113. Transmitter test mode



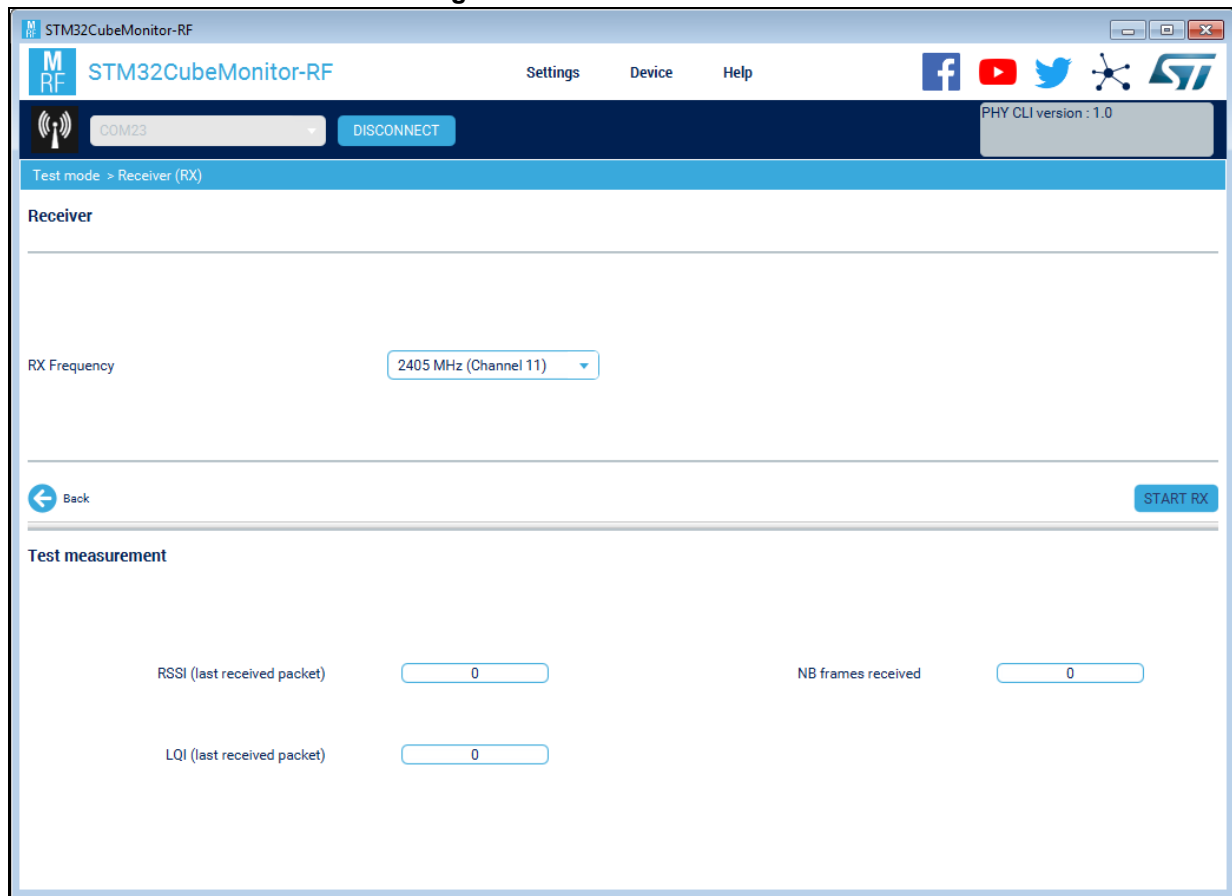
The user must:

- Select the power level (+6 dBm to -21 dBm).
- Select the TX frequency (channel11 - 2405MHz to channel 26 - 2480 MHz).
- Press “Start TX” key, the emission starts, press “Stop TX” key to interrupt the test.

5.3 Receiver test (RX) mode

This test mode configures the device in reception.

Figure 114. Receiver test mode



The user must:

- Select one frequency in the pick list
- Press Start RX key: the device is configured in the receiver mode at the selected frequency.
- Press Stop RX key: the device stops the reception and supply the result, RSSI, LQI and number of frame. Those results appear in the low side.

5.4 Packet error rate (PER) mode

This mode configures the device in reception and one other device to play the role of generator.

The tool makes three measurements:

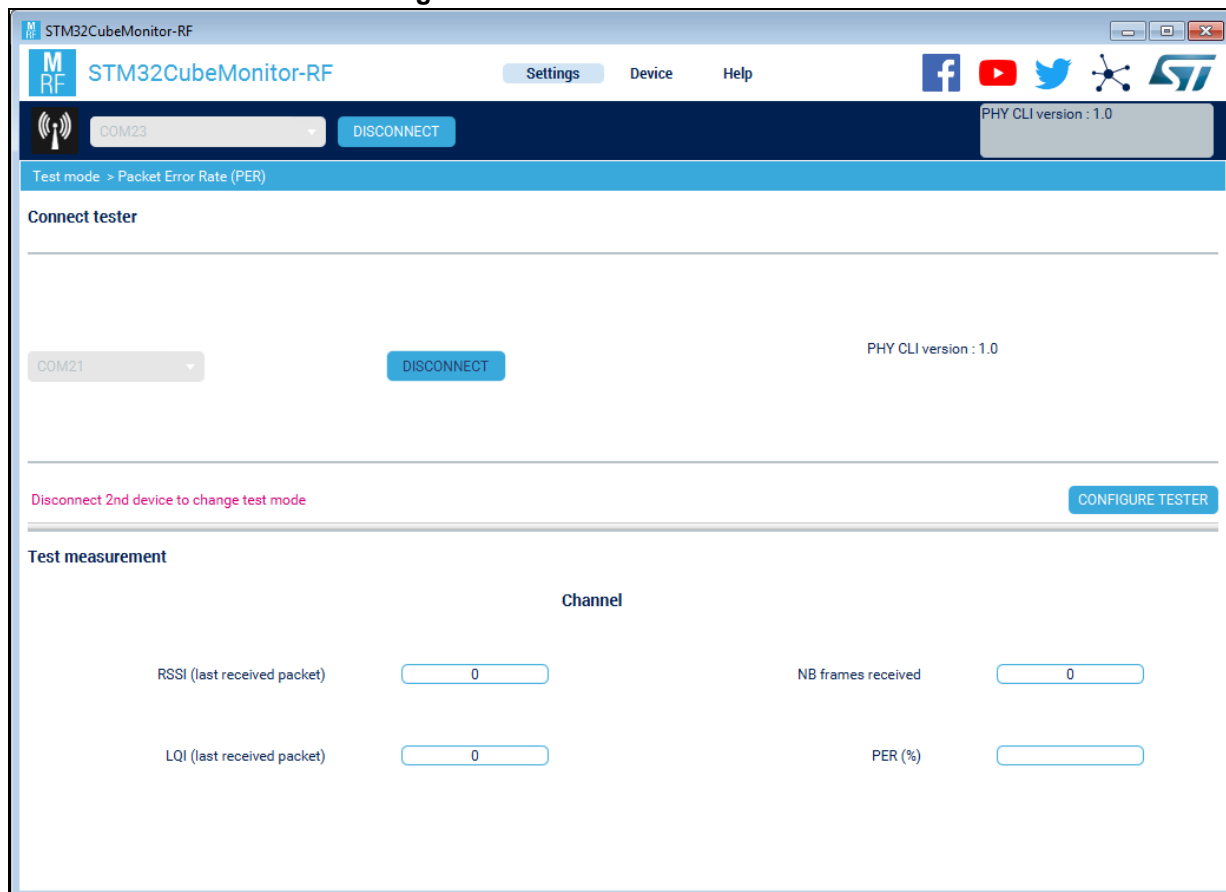
- RSSI: Received Signal Strength Indication
- LQI: Link Quality Indicator
- PER: Packet Error Rate - computed with the number of frames received and number of frames sent $(\text{Number of frames received} / \text{Number of Frames sent}) * 100$

Four steps are necessary:

- Connect the additional device for playing the role of packet generator (tester).
- Configure the parameters of tester.
- Configure the parameters of device under test (DUT).
- Configure the measurement.

5.4.1 Connecting the additional device to play the role of packet generator (tester).

Figure 115. Packet tester connection



- Plug one additional device in the computer (same requirements as first device, see [Section 2.2](#)).
- Select the serial port to use in the picklist.
- Click on the 'CONNECT' key, the device information must appear in the right side of the "connect" key.

When second device is connected, it is not possible to change mode. The user need disconnect the device first, and then use the 'back' button.

Click on 'CONFIGURE TESTER' to set the tester parameters.

5.4.2 Configure the parameters of tester.

Figure 116. PER tester configuration

STM32CubeMonitor-RF

Settings Device Help

COM23 DISCONNECT PHY CLI version : 1.0

Test mode > Packet Error Rate (PER) > Configure tester

Configure tester (COM21)

Power level +6 dBm

TX Frequency 2405 MHz (Channel 11)

Back CONFIGURE DUT

Test measurement

Channel			
RSSI (last received packet)	0	NB frames received	0
LQI (last received packet)	0	PER (%)	

The user must:

- Select the power level in the pick list "Power Level".
- Select the frequency in the pick list "TX frequency". Only the single measurement mode uses this parameter, no continuous or multiple channels modes. It is applied to tester device.

Click on 'CONFIGURE DUT' to set the Device Under Test configuration.

5.4.3 Configure the parameters of device under test (DUT).

Figure 117. DUT configuration

The screenshot shows the STM32CubeMonitor-RF software interface. At the top, there's a header with the STM logo and 'STM32CubeMonitor-RF'. Below this is a navigation bar with 'Settings', 'Device', and 'Help' tabs. A status bar shows 'COM23' and a 'DISCONNECT' button. The main area is titled 'Test mode > Packet Error Rate (PER) > Configure tester > Configure DUT'. The sub-header is 'Configure Device Under Test (DUT) (COM23)'. Under 'RX Frequency', a dropdown menu is set to '2405 MHz (Channel 11)'. At the bottom, there's a 'Test measurement' section with a 'Channel' label. It contains four data fields: 'RSSI (last received packet)' with a value of 0, 'NB frames received' with a value of 0, 'LQI (last received packet)' with a value of 0, and 'PER (%)' with an empty field. A 'Back' button is on the left and a 'CONFIGURE PARAM' button is on the right of the measurement section.

The user must:

- Select the frequency in the pick list "RX frequency". It is the frequency of the DUT.

Click on 'CONFIGURE PARAM' to set the test configuration:

5.4.4 Configure the measurement.

Figure 118. PER test parameters

STM32CubeMonitor-RF

Settings Device Help

COM23 DISCONNECT

PHY CLI version : 1.0

Test mode > Packet Error Rate (PER) > Configure tester > Configure DUT > Configure param

Configure Additional Settings

☒ Continuous measurement NB frames transmitted 100

☒ Multiple channels Fill channel List: 11-26

☒ Save test verdict in file No file chosen SELECT FILE

Back START TEST

Test measurement

Channel			
RSSI (last received packet)	0	NB frames received	0
LQI (last received packet)	0	PER (%)	

Three measurement modes are available:

- **Single measurement** measures once the frames number defined. The frequency of the tester is the one defined in panel "PER tester configuration" (TX frequency). The frequencies of DUT is as defined in panel "DUT configuration".
- **Continuous measurement** repeats the measurement on frames number until the user press "Stop Test" key. The frequency of DUT and Tester is the same, it's the one defined in the panel DUT configuration.
- **Multiple channels** measures the frequency defined in "Fill channel list" field. The default values are "11-26" this means all channels in range 11 to 26. It is possible to use a comma to define channel by channel: "12,15,24" or mix both: "11,14-20,25,26". The user can interrupt the test with "Stop Test" key.

The result of continuous and multiple channels measurements can be saved in a csv file. The user must check the "Save test verdict in file" check box, and must define the name of file by "SELECT FILE" key before starting the test.

Table 3. Measurement setting

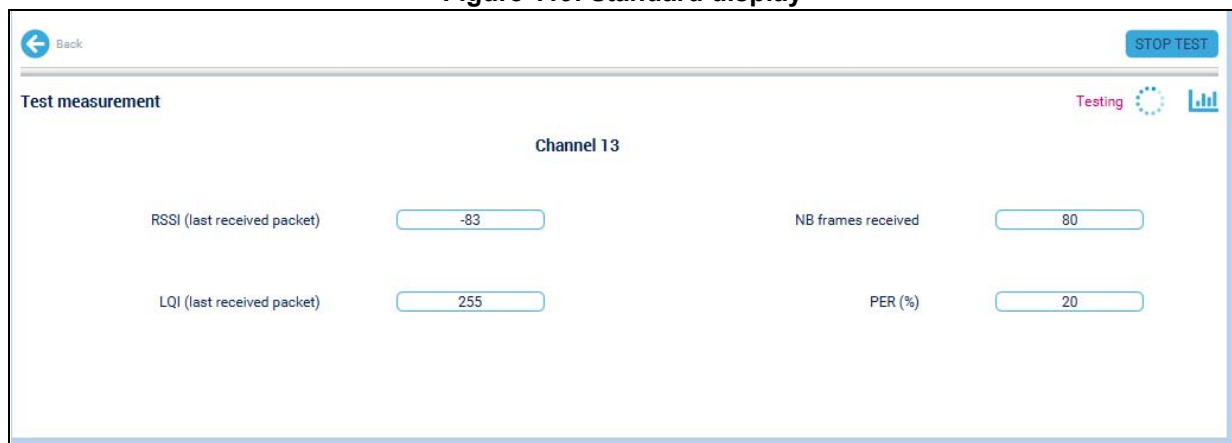
Measurements	Single	Continuous	Multiple channels
Continuous measurement check box	Unchecked	Checked	Checked
Multiple Channels check box	Unchecked	Unchecked	Checked
Save test verdict in file check box	Not available	Available	Available

Three display modes are available:

1. Standard display

There are the PER value, RSSI value and LQI for one channel.

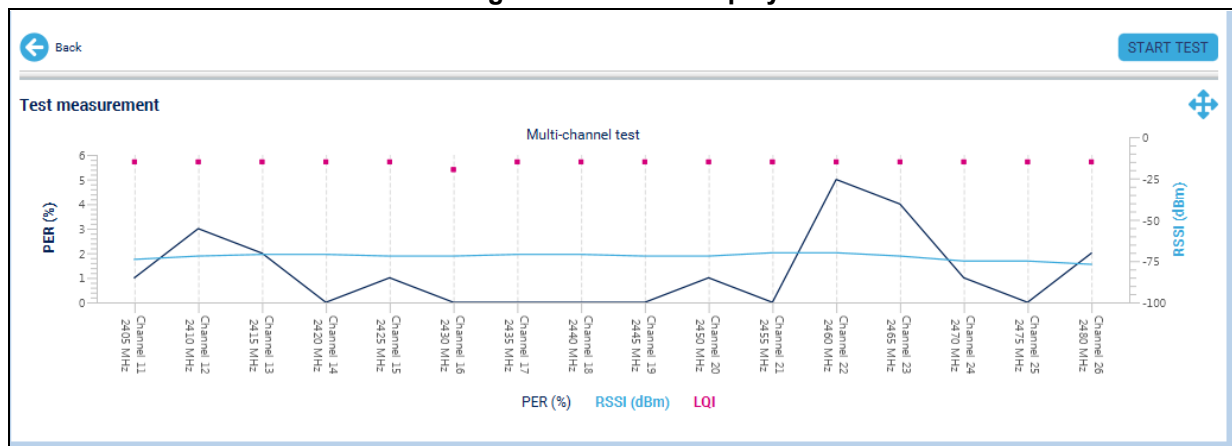
Figure 119. Standard display



2. Chart display

In the same chart there are the PER value, RSSI value and LQI for channels the user defined.

Figure 120. Chart display



3. Large display

It is possible to switch from a PER display to RSSI and LQI with the arrow icons on left or right side.

Figure 121. Large PER display

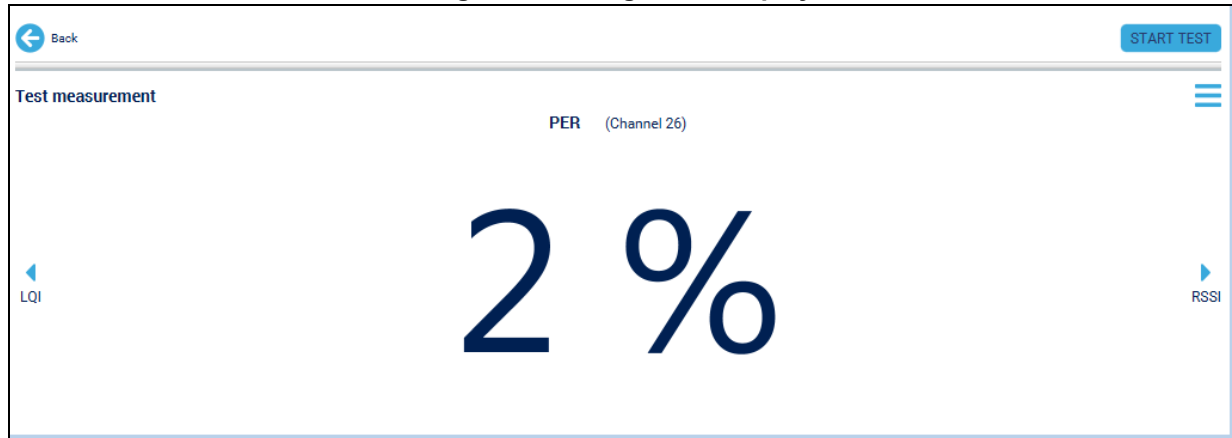
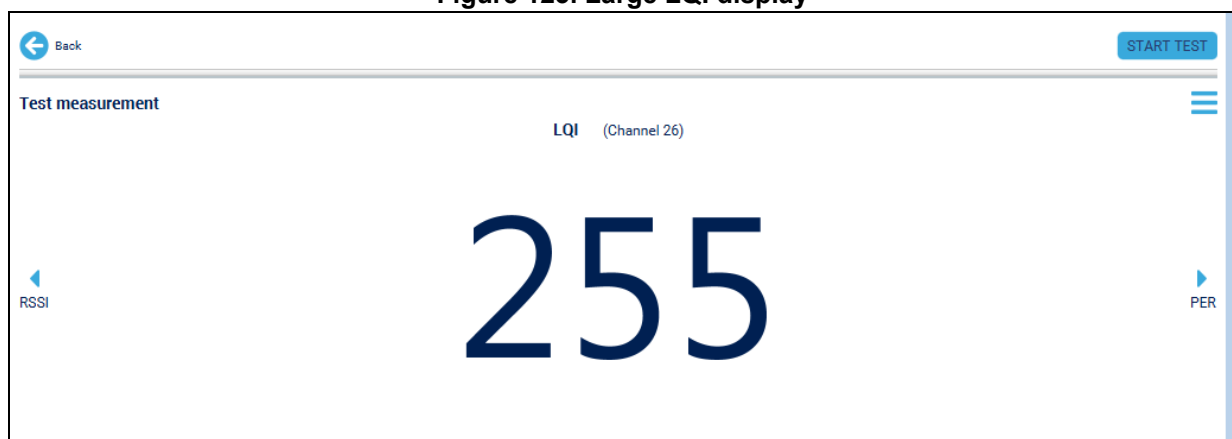


Figure 122. Large RSSI display



Figure 123. Large LQI display



Appendix A Beacon configuration format

The beacon configuration file is a binary file. Its content is explained in [Table 4](#).

Table 4. Beacon configuration format

Byte #	Name	Value	Description
0	Address type	0,1,2	Address type: 0 = board default address 1 = random address 2 = static address provided in the block
1 - 6	Address	address	static address for the beacon. Valid only if address type=2
7	Tx power	0x00-0x1F	Tx power to be used for the beacon. Value 'PA_Level' of command ACI_HAL_SET_TX_POWER_LEVEL: 0 to 31
8	Beacon additional feature	0 or 1	0: No additional feature 1: TLM activated Other values reserved
9	Advertising payload length	13-32	Length of payload data
10 - 41	Advertising payload	-	Beacon advertisement payload, to be inserted in the advertisement

Revision history

Table 5. Document revision history

Date	Revision	Changes
27-Nov-2017	1	Initial version
25-Jan-2018	2	Updated: Introduction Section 3.3.2: Test mode receiver (RX) Added: Two tables: Table 2: Specific AD encoding for code example and Table 3: Search filtering Twelve new figures Section 3.2.1: How to send an ACI command Section 3.2.2: Search function Section 3.4.4: Script report Section : Pause command in the script Section 3.5.3: Advertising change for OTA in ST example
23-Aug-2018	3	Complete content reorganized to explain tool support to the original BLE mode in Section 3 and the new OpenThread mode in Section 4 .
13-Feb-2019	4	Updated: Section 4: OpenThread mode and most of the figures with new version tool Added: Section 5: 802.15.4 RF test mode
12-Jul-2019	5	Updated: Tool version 2.4.0 Section 3.5: OTA transfer simplified. Details reported in application note.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

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

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

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

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

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

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

© 2019 STMicroelectronics – All rights reserved