Introduction

STM32CubeMonitor-RF (STM32CubeMonRF) is a software tool, which helps 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 the packet error rate (PER) on Bluetooth® Low Energy and 802.15.4 technologies.
- It sends commands to the Bluetooth® Low Energy controller for standardized tests.
- It sends and receives Bluetooth® Low Energy commands for fast application prototyping.
- It configures a variety of beacons via Bluetooth® Low Energy commands.
- It transfers data over-the-air (OTA) from one device to another, to configure or program a wireless remote device.
- It sends commands to an OpenThread device for application prototyping.
- It explores a Thread® network and displays it with all the relevant information.
- It provides a sniffer tool to analyze 802.15.4 frames with Wireshark™

This user manual applies to STM32CubeMonitor-RF version 2.13.0 and later.
# Contents

1 Getting started ................................................. 9  
   1.1 Download and setup ......................................... 9  
   1.2 Welcome screen ........................................... 10  
   1.3 Main screen ................................................ 11  
      1.3.1 Menu bar ............................................. 11  
      1.3.2 Connection bar ....................................... 12  
      1.3.3 Panels ............................................... 12  
      1.3.4 Log area ............................................. 14  

2 Connection to target ........................................ 15  
   2.1 Use case description and definitions .................... 15  
   2.2 VCOM/UART connection .................................... 16  
      2.2.1 VCOM connection .................................... 16  
      2.2.2 UART connection .................................... 18  
      2.2.3 VCP device .......................................... 19  
   2.3 Opening COM ............................................. 20  

3 Bluetooth® Low Energy mode ................................. 22  
   3.1 Presentation ............................................... 22  
      3.1.1 Panels ............................................... 22  
   3.2 Bluetooth® Low Energy stack ............................ 23  
   3.3 ACI Commands panel .................................... 23  
      3.3.1 How to send an ACI command ......................... 24  
      3.3.2 Search function .................................... 24  
      3.3.3 Filter usage ........................................ 25  
      3.3.4 How to fill parameters. Fixed field / editable field ........................................ 25  
      3.3.5 Log functionalities ................................ 26  
   3.4 RF test panel ............................................. 29  
      3.4.1 Transmitter test mode (TX) .......................... 30  
      3.4.2 Receiver test mode (RX) ............................ 33  
      3.4.3 PER ................................................ 36  
   3.5 Scripts .................................................. 41  
      3.5.1 Launching scripts .................................. 42
3.5.2 Script recording ................................................. 44
3.5.3 Scripts modification ........................................... 44
3.5.4 Script report .................................................. 45
3.5.5 List of script commands ....................................... 46
3.6 OTA transfer ...................................................... 49
  3.6.1 OTA presentation ............................................ 49
  3.6.2 OTA procedure .............................................. 50
  3.6.3 Use the tool to perform an OTA update ....................... 51
3.7 Beacon ............................................................ 55
  3.7.1 Beacon presentation ......................................... 55
  3.7.2 Beacon configuration methods: ................................ 56
  3.7.3 Configuration of the beacon with STM32CubeMonitor-RF ...... 58
3.8 ACI Utilities ...................................................... 62
  3.8.1 Remote services discovering ................................. 62
  3.8.2 Advertising ................................................ 69
4 OpenThread mode .................................................. 71
  4.1 Presentation .................................................. 71
    4.1.1 Panel .................................................... 71
  4.2 Commands tab ................................................ 73
  4.3 OpenThread scripts tab ........................................ 75
    4.3.1 OpenThread script example ............................... 76
    4.3.2 List of script commands .................................. 76
  4.4 Network Explorer tab ......................................... 77
    4.4.1 Controls ................................................ 77
    4.4.2 Display area ............................................. 79
    4.4.3 Infobox ................................................ 81
    4.4.4 Log area ................................................. 81
5 802.15.4 RF test mode .......................................... 82
  5.1 Presentation ................................................ 82
  5.2 Transmitter test mode (TX) .................................. 83
    5.2.1 Frame mode ............................................ 83
    5.2.2 Continuous modulated mode .............................. 84
    5.2.3 Continuous wave mode .................................. 85
  5.3 Receiver test (RX) mode ..................................... 85
5.3.1 Packet error rate (PER) test ........................................ 86
5.3.2 Link quality assessment (LQI) test ............................... 87
5.3.3 Energy detection (ED) test ........................................ 87
5.3.4 Channel clear assessment (CCA) test ............................ 88

5.4 Packet error rate (PER) mode ....................................... 89
5.4.1 Connecting the additional device to play the role of a packet generator (tester). .................................................. 90
5.4.2 Configure the parameters of the tester ............................ 91
5.4.3 Configure the parameters of the device under test (DUT). ........ 92
5.4.4 Configure the measurement ....................................... 93

6 802.15.4 sniffer .......................................................... 96
6.1 Presentation .............................................................. 96
6.2 Prerequisite .............................................................. 96
6.2.1 Sniffer device ......................................................... 96
6.2.2 Wireshark ............................................................ 96
6.2.3 Python™ .............................................................. 96
6.3 Setup verification .......................................................... 97
6.3.1 Sniffer launch ........................................................ 98
6.3.2 Select interface ....................................................... 98
6.3.3 Channel configuration ............................................... 99
6.3.4 Sniffing start ........................................................ 99

Appendix A Beacon configuration format ............................... 100

Revision history .......................................................... 101
List of tables

Table 1. OTA loader address table ................................................................. 50
Table 2. Search filtering ............................................................................. 52
Table 3. Measurement setting .................................................................... 94
Table 4. Beacon configuration format .......................................................... 100
Table 5. Document revision history .............................................................. 101
# List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Welcome screen</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Main screen</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Menu bar</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Connection bar</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>ACI Commands panel</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Log area</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Typical connection with a Nucleo board</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Connection with a remote device</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>Connection with a second device</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>VCOM connection Bluetooth® Low Energy</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>VCOM connection Thread®</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>UART connection</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>VCP connection</td>
<td>19</td>
</tr>
<tr>
<td>14</td>
<td>Opening COM</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>Successful COM</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>ACI Commands panel</td>
<td>22</td>
</tr>
<tr>
<td>17</td>
<td>How to send an ACI command</td>
<td>24</td>
</tr>
<tr>
<td>18</td>
<td>Search button</td>
<td>25</td>
</tr>
<tr>
<td>19</td>
<td>Fixed parameter</td>
<td>25</td>
</tr>
<tr>
<td>20</td>
<td>Editable parameter</td>
<td>25</td>
</tr>
<tr>
<td>21</td>
<td>Predefined values</td>
<td>25</td>
</tr>
<tr>
<td>22</td>
<td>Help details</td>
<td>26</td>
</tr>
<tr>
<td>23</td>
<td>Log functionalities</td>
<td>26</td>
</tr>
<tr>
<td>24</td>
<td>More button</td>
<td>27</td>
</tr>
<tr>
<td>25</td>
<td>Message details</td>
<td>27</td>
</tr>
<tr>
<td>26</td>
<td>Purple error messages</td>
<td>28</td>
</tr>
<tr>
<td>27</td>
<td>Gray second board messages</td>
<td>28</td>
</tr>
<tr>
<td>28</td>
<td>Save ACI log</td>
<td>29</td>
</tr>
<tr>
<td>29</td>
<td>LOG_ACI_CubeMonitor-RF.csv</td>
<td>29</td>
</tr>
<tr>
<td>30</td>
<td>Test mode selection</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>Change the test mode</td>
<td>30</td>
</tr>
<tr>
<td>32</td>
<td>Select the test mode</td>
<td>30</td>
</tr>
<tr>
<td>33</td>
<td>Transmitter test mode</td>
<td>31</td>
</tr>
<tr>
<td>34</td>
<td>Transmitting message</td>
<td>32</td>
</tr>
<tr>
<td>35</td>
<td>Transmitted packets count</td>
<td>32</td>
</tr>
<tr>
<td>36</td>
<td>Receiver test mode</td>
<td>33</td>
</tr>
<tr>
<td>37</td>
<td>RSSI measurement</td>
<td>34</td>
</tr>
<tr>
<td>38</td>
<td>RSSI measurement graph</td>
<td>35</td>
</tr>
<tr>
<td>39</td>
<td>RF RSSI measurement large display</td>
<td>36</td>
</tr>
<tr>
<td>40</td>
<td>PER definition</td>
<td>36</td>
</tr>
<tr>
<td>41</td>
<td>PER tester connection</td>
<td>37</td>
</tr>
<tr>
<td>42</td>
<td>PER tester connected</td>
<td>38</td>
</tr>
<tr>
<td>43</td>
<td>PER tester configuration</td>
<td>39</td>
</tr>
<tr>
<td>44</td>
<td>DUT configuration</td>
<td>39</td>
</tr>
<tr>
<td>45</td>
<td>PER test parameters</td>
<td>40</td>
</tr>
<tr>
<td>46</td>
<td>PER and RSSI measurement graph</td>
<td>41</td>
</tr>
<tr>
<td>47</td>
<td>Launching scripts</td>
<td>42</td>
</tr>
<tr>
<td>48</td>
<td>Script execution</td>
<td>43</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>49</td>
<td>Script recording buttons</td>
<td>44</td>
</tr>
<tr>
<td>50</td>
<td>Sample script</td>
<td>44</td>
</tr>
<tr>
<td>51</td>
<td>Script report</td>
<td>45</td>
</tr>
<tr>
<td>52</td>
<td>Script verdict</td>
<td>46</td>
</tr>
<tr>
<td>53</td>
<td>Script pause</td>
<td>47</td>
</tr>
<tr>
<td>54</td>
<td>Example</td>
<td>47</td>
</tr>
<tr>
<td>55</td>
<td>Loop simple example</td>
<td>47</td>
</tr>
<tr>
<td>56</td>
<td>Loop second simple example</td>
<td>48</td>
</tr>
<tr>
<td>57</td>
<td>Loop decrement</td>
<td>48</td>
</tr>
<tr>
<td>58</td>
<td>Loop specific increment</td>
<td>48</td>
</tr>
<tr>
<td>59</td>
<td>Nested loop</td>
<td>48</td>
</tr>
<tr>
<td>60</td>
<td>Example of loop script verdict</td>
<td>49</td>
</tr>
<tr>
<td>61</td>
<td>Display of loop script verdict</td>
<td>49</td>
</tr>
<tr>
<td>62</td>
<td>Script loop error</td>
<td>49</td>
</tr>
<tr>
<td>63</td>
<td>Search procedure</td>
<td>52</td>
</tr>
<tr>
<td>64</td>
<td>Scanning</td>
<td>53</td>
</tr>
<tr>
<td>65</td>
<td>No device found</td>
<td>53</td>
</tr>
<tr>
<td>66</td>
<td>Device found</td>
<td>53</td>
</tr>
<tr>
<td>67</td>
<td>Select the device and parameters</td>
<td>54</td>
</tr>
<tr>
<td>68</td>
<td>Configuring in OTA</td>
<td>55</td>
</tr>
<tr>
<td>69</td>
<td>Progress bar</td>
<td>55</td>
</tr>
<tr>
<td>70</td>
<td>Beacon presentation</td>
<td>56</td>
</tr>
<tr>
<td>71</td>
<td>Beacon usage</td>
<td>56</td>
</tr>
<tr>
<td>72</td>
<td>Online beacon</td>
<td>57</td>
</tr>
<tr>
<td>73</td>
<td>Offline beacon</td>
<td>57</td>
</tr>
<tr>
<td>74</td>
<td>Selecting the Beacon mode</td>
<td>58</td>
</tr>
<tr>
<td>75</td>
<td>Common parameters</td>
<td>58</td>
</tr>
<tr>
<td>76</td>
<td>iBeacon parameters</td>
<td>59</td>
</tr>
<tr>
<td>77</td>
<td>Eddystone UID</td>
<td>60</td>
</tr>
<tr>
<td>78</td>
<td>Eddystone URL</td>
<td>60</td>
</tr>
<tr>
<td>79</td>
<td>Online mode transfer configuration</td>
<td>61</td>
</tr>
<tr>
<td>80</td>
<td>Offline mode transfer configuration</td>
<td>61</td>
</tr>
<tr>
<td>81</td>
<td>ACI Utilities panel</td>
<td>62</td>
</tr>
<tr>
<td>82</td>
<td>Select checkbox</td>
<td>62</td>
</tr>
<tr>
<td>83</td>
<td>Scan parameters</td>
<td>63</td>
</tr>
<tr>
<td>84</td>
<td>Scanning</td>
<td>64</td>
</tr>
<tr>
<td>85</td>
<td>Select device</td>
<td>64</td>
</tr>
<tr>
<td>86</td>
<td>Back</td>
<td>64</td>
</tr>
<tr>
<td>87</td>
<td>Init</td>
<td>64</td>
</tr>
<tr>
<td>88</td>
<td>Connecting</td>
<td>65</td>
</tr>
<tr>
<td>89</td>
<td>Connection error</td>
<td>65</td>
</tr>
<tr>
<td>90</td>
<td>Connected icon</td>
<td>65</td>
</tr>
<tr>
<td>91</td>
<td>Services list</td>
<td>65</td>
</tr>
<tr>
<td>92</td>
<td>Characteristics list</td>
<td>66</td>
</tr>
<tr>
<td>93</td>
<td>Read value</td>
<td>66</td>
</tr>
<tr>
<td>94</td>
<td>Write value</td>
<td>67</td>
</tr>
<tr>
<td>95</td>
<td>Indicate value changed</td>
<td>67</td>
</tr>
<tr>
<td>96</td>
<td>Notify value changed</td>
<td>68</td>
</tr>
<tr>
<td>97</td>
<td>Notifying</td>
<td>68</td>
</tr>
<tr>
<td>98</td>
<td>Advertising parameters</td>
<td>69</td>
</tr>
<tr>
<td>99</td>
<td>Advertising</td>
<td>70</td>
</tr>
<tr>
<td>100</td>
<td>Connected</td>
<td>70</td>
</tr>
</tbody>
</table>
### List of figures

<table>
<thead>
<tr>
<th>Figure number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 101</td>
<td>OpenThread - Command tab</td>
<td>71</td>
</tr>
<tr>
<td>Figure 102</td>
<td>OpenThread - Script tab</td>
<td>72</td>
</tr>
<tr>
<td>Figure 103</td>
<td>OpenThread - Network Explorer tab</td>
<td>72</td>
</tr>
<tr>
<td>Figure 104</td>
<td>OpenThread common bottom area</td>
<td>73</td>
</tr>
<tr>
<td>Figure 105</td>
<td>OpenThread command-tab - top area</td>
<td>73</td>
</tr>
<tr>
<td>Figure 106</td>
<td>Script buttons</td>
<td>74</td>
</tr>
<tr>
<td>Figure 107</td>
<td>Command list</td>
<td>74</td>
</tr>
<tr>
<td>Figure 108</td>
<td>Command details</td>
<td>74</td>
</tr>
<tr>
<td>Figure 109</td>
<td>Scripts tab</td>
<td>75</td>
</tr>
<tr>
<td>Figure 110</td>
<td>Sample script</td>
<td>76</td>
</tr>
<tr>
<td>Figure 111</td>
<td>OpenThread network explorer tab</td>
<td>77</td>
</tr>
<tr>
<td>Figure 112</td>
<td>Auto connection functions</td>
<td>77</td>
</tr>
<tr>
<td>Figure 113</td>
<td>Project and background management</td>
<td>78</td>
</tr>
<tr>
<td>Figure 114</td>
<td>Explore and size choice controls</td>
<td>78</td>
</tr>
<tr>
<td>Figure 115</td>
<td>Display area</td>
<td>79</td>
</tr>
<tr>
<td>Figure 116</td>
<td>Zoom and motion controls</td>
<td>80</td>
</tr>
<tr>
<td>Figure 117</td>
<td>Infobox</td>
<td>81</td>
</tr>
<tr>
<td>Figure 118</td>
<td>Log area</td>
<td>81</td>
</tr>
<tr>
<td>Figure 119</td>
<td>Test mode selection</td>
<td>82</td>
</tr>
<tr>
<td>Figure 120</td>
<td>Transmitter test mode</td>
<td>83</td>
</tr>
<tr>
<td>Figure 121</td>
<td>Field and picklist defining the frame</td>
<td>83</td>
</tr>
<tr>
<td>Figure 122</td>
<td>Help frame information</td>
<td>84</td>
</tr>
<tr>
<td>Figure 123</td>
<td>Continuous modulated test mode</td>
<td>84</td>
</tr>
<tr>
<td>Figure 124</td>
<td>Receiver test mode</td>
<td>85</td>
</tr>
<tr>
<td>Figure 125</td>
<td>PER frames received</td>
<td>86</td>
</tr>
<tr>
<td>Figure 126</td>
<td>PER frame reception completed</td>
<td>86</td>
</tr>
<tr>
<td>Figure 127</td>
<td>LQI measurement</td>
<td>87</td>
</tr>
<tr>
<td>Figure 128</td>
<td>ED measurement</td>
<td>88</td>
</tr>
<tr>
<td>Figure 129</td>
<td>CCA measurement</td>
<td>89</td>
</tr>
<tr>
<td>Figure 130</td>
<td>Packet tester connection</td>
<td>90</td>
</tr>
<tr>
<td>Figure 131</td>
<td>PER tester configuration</td>
<td>91</td>
</tr>
<tr>
<td>Figure 132</td>
<td>DUT configuration</td>
<td>92</td>
</tr>
<tr>
<td>Figure 133</td>
<td>PER test parameters</td>
<td>93</td>
</tr>
<tr>
<td>Figure 134</td>
<td>Standard display</td>
<td>94</td>
</tr>
<tr>
<td>Figure 135</td>
<td>Chart display</td>
<td>94</td>
</tr>
<tr>
<td>Figure 136</td>
<td>Large PER display</td>
<td>95</td>
</tr>
<tr>
<td>Figure 137</td>
<td>Large RSSI display</td>
<td>95</td>
</tr>
<tr>
<td>Figure 138</td>
<td>Large LQI display</td>
<td>95</td>
</tr>
<tr>
<td>Figure 139</td>
<td>Prerequisite check</td>
<td>97</td>
</tr>
<tr>
<td>Figure 140</td>
<td>Wireshark interfaces</td>
<td>98</td>
</tr>
<tr>
<td>Figure 141</td>
<td>Wheel</td>
<td>98</td>
</tr>
<tr>
<td>Figure 142</td>
<td>Channel choice</td>
<td>99</td>
</tr>
<tr>
<td>Figure 143</td>
<td>Sniffing</td>
<td>99</td>
</tr>
</tbody>
</table>
1 Getting started

STM32CubeMonitor-RF supports STM32WBxx and STM32WBArxx microcontrollers based on the Arm® Cortex®-M processor.

STM32WBxx refers to STM32WB5x, STM32WB3x, and STM32WB1x while STM32WBArxx refers to STM32WBAr5x.

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.

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

Launching the application opens the welcome screen, where the users select the mode that they want to use: Bluetooth® Low Energy, OPENTHREAD, 802.15.4 RF, or 802.15.4 sniffer. The checkbox *Remember my choice* memorizes the selection so that the next application launch directly opens it, without the welcome screen, except for the 802.15.4 sniffer.
1.3 Main screen

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

![Main screen](image)

1.3.1 Menu bar

The application header provides a menu to use specific tools and displays helpful 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 the connected board.

The `Help` menu provides information about the version of the 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 website
- The STMicroelectronics icon leads to the STMicroelectronics website.

1.3.2 Connection bar

![Connection bar](image)

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 Bluetooth® Low Energy wireless stack. When many tests are performed, the button must be used to reset the stack at the start of each test.

CM0 and CM4 information only applies to STM32WBxx. For STM32WBAxx, hardware and firmware versions are displayed.

1.3.3 Panels

The panels are used to perform specific operations. Each panel regroups different functions. The `ACI Commands` panel example is illustrated in Figure 5.
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 3.3: ACI Commands panel on page 23**
- **Section 3.5: Scripts on page 41**
- **Section 3.7: Beacon on page 55**
- **Section 3.4: RF test panel on page 29**
- **Section 3.8: ACI Utilities on page 62**
1.3.4 Log area

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 3.3.5 on page 26.
2 Connection to target

2.1 Use case description and definitions

STM32CubeMonitor-RF usually connects to one STM32WB[A]xx 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 the main device. This is the board that the user wants to exercise with the tool. It is also named the device under test (DUT).

![Figure 7. Typical connection with a Nucleo board](image)

Some operations, like downloading over-the-air, involve communication with another device. This other device is referred to as the remote device in this document.

![Figure 8. Connection with a remote device](image)
One RF test uses two boards to measure the error rate of a packet transfer. For such a test, a second device is connected; it is named the 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 the host controller interface (HCI) or command-line interface (CLI) commands between STM32CubeMonitor-RF and the wireless stack held in the STM32WB[A]xx part. HCI commands are used for Bluetooth® Low Energy applications. CLI commands 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 Bluetooth® Low Energy**

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 STM32WB[A]xx microcontroller.

For Bluetooth® Low Energy, a special firmware in STM32WB[A]xx called Transparent mode copies the data received on the Rx pin to the Bluetooth® Low Energy stack. Data sent back by the Bluetooth® Low Energy stack follows the reverse path.

The Transparent mode firmware is available in the STM32CubeWB firmware package (refer to folder \Projects\xxx\Applications\BLE\BLE_TransparentMode.).

For the STM32WB[B]xx devices, the wireless stack firmware stm32wb5x_BLE_Stack_full_extended_fw.bin is available in \Projects\STM32WB_Copro_Wireless_Binaries.

For more information, refer to Section 3.2: Bluetooth® Low Energy stack.

For Thread®, the Thread_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.

For STM32WB[B]xx devices, the CLI firmware source code is available in the STM32CubeWB firmware package (refer to the \Projects\xxx\Applications\Thread\Thread_Cmd.cmd folder). The wireless stack firmware stm32wb[B]xx_Thread_FTD_fw.bin is available in \Projects\STM32WB_Copro_Wireless_Binaries.

For STM32WB[B]xx devices, the source is available in the STM32CubeWBA firmware package (refer to \Projects\NUCLEO-WBAxxx\Applications\Thread\Thread_Cmd.cmd folder).

**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.

For STM32WB[B]xx devices, the Phy_802_15_4_Cmd source is available in the STM32CubeWB firmware package (refer to the \Projects\P-NUCLEO-WB55.Nucleo\Applications\Phy_802_15_4\Phy_802_15_4_Cmd folder). The wireless stack firmware stm32wb[B]x_Phy_802_15_4_cmd.bin is available in the \Projects\STM32WB_Copro_Wireless_Binaries folder.
For STM32WBxx devices, the source is available in the STM32CubeWBA firmware package (`\Projects\NUCLEO-WBxxxx\Applications\Phy_802_15_4\ Phy_Cli` folder).

When a USB-to-serial converter replaces the ST-LINK part, the VCOM driver might be installed automatically on the computer. For the converter without an 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, the 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 the 802.15.4 test:

1. Build the Nucleo firmware `Phy_802_15_4_Cli` and flash it. ([STM32Cube_FW_WB_Vx.x.x\Projects\P-NUCLEO-WB55.Nucleo\Applications\ Phy_802_15_4\ Phy_Cli](STM32Cube_FW_WB_Vx.x.x\Projects\P-NUCLEO-WB55.Nucleo\Applications\ Phy_802_15_4\ Phy_Cli))

2. Flash with DFU the wireless stack `stm32wb5x_Phy_802_15_4_fw.bin` in the dongle (binary in `Projects\STM32WB_Copro_Wireless_Binaries`).

3. Move solder bridge SB2 to SB6 (connection of PB7 to CN2.7).

4. Connect the serial cable to PB7 (PC Tx) and PB6 (PC Rx), PB7 is on 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 might be installed automatically on the computer or the user needs to install it manually. This configuration is used with the STM32WB55 USB dongle reference board and the Nucleo sniffer configuration.

1. For Bluetooth® Low Energy:
   The firmware is in \Projects\NUCLEO WB55.USBDongle\Applications\BLE\BLE_TransparentModeVCP.
   The wireless stack is in: \Projects\STM32WB_Copro_Wireless_Binaries\STM32WB5x\stm32wb5x_BLE_Stack_full_extended_fw.bin
   For more information, refer to Section 3.2: Bluetooth® Low Energy stack.

2. For Thread®:
   The firmware source code is in \Projects\NUCLEO WB55.USBDongle\Applications\Thread\Thread_Cli_Cmd.
   The wireless stack is in the
   \Projects\STM32WB_Copro_Wireless_Binaries\STM32WB5x\stm32wb5x_Thread_FT_D_fw.bin folder.

3. For the 802.15.4 sniffer:
   The wireless stack is in the
   \Projects\STM32WB_Copro_Wireless_Binaries\STM32WB5x\stm32wb5x_Mac_802_15_4_fw.bin folder. The firmware binaries are Mac_802_15_4_Sniffer_Dongle.bin and Mac_802_15_4_Sniffer_Nucleo.bin. The location changes with the operating system.
   a) For Windows, the firmware is in the
      <Public documents>\STMicroelectronics\STM32CubeMonitor-RF\sniffer folder,
which means C:\Users\Public\Documents\STMicroelectronics\STM32CubeMonitor-RF\firmwares.

b) For Linux, the firmware is in the <userhome>/STMicroelectronics/STM32CubeMonitor-RF/sniffer folder.

c) For macOS®, the firmware is inside the document folder provided in the setup package: /Users/Public/Documents/STMicroelectronics/STM32CubeMonitor-RF/firmwares/Mac_802_15_4_Sniffer.bin

2.3 Opening COM

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

![Figure 14. Opening COM](image)

The procedure is:
- Connect the board to the computer. If VCOM or VCP is used, a driver needs to be installed. It might take a few seconds at the first connection. For some devices, drivers need to be installed manually.
- Select the serial port to use in the picklist (Comx on Windows and ttyACMx on Linux and Mac). On the Mac, the cu.Bluetooth-Incoming-Port port is the Mac Bluetooth adapter. This port is not connected to the STM32 device and must not be used).
- Click CONNECT

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

![Figure 15. Successful COM](image)

When the CONNECT button is pressed, the software attempts to communicate with the device to read the firmware and hardware versions. If the connection is not working, the tool displays an error and disconnects the COM port.
Caution: In case of a 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 the driver might not install automatically. If the tool is not showing the COM port in the list, check that the drivers are properly installed.
- When a user tries to open a device with the wrong mode, for example, open a Bluetooth® Low Energy device in Thread mode, the device cannot decode the command sent and might freeze or crash the target software. In this case, it is necessary to unplug/replug the board to reset it.
- Delay on Ubuntu®(a):
  - On Ubuntu, the modemmanager process checks the COM port when the board is plugged in. Due to this activity, the COM port is busy for a few seconds, and STM32CubeMonitor-RF is unable to connect.
  - The user must wait for the end of the modemmanager activity before opening the COM port.
  - If the user does not require the modemmanager, it is possible to uninstall it with the sudo apt-get purge modemmanager command.
- Port not visible on Linux:
  - The user might not have the proper access rights for ttyACM. Ubuntu requires adding the user to the dial-out group with the sudo adduser <username> dialout command (replace username with user name).
- If another application opens the port, the tool is unable to connect.
- When a USB device is removed, the Virtual COM port is not closed automatically, and the software might not be informed of the disconnection. If a USB device is inserted when the Virtual COM port is already opened, the board is not mounted in the system. To solve this, close the COM port on STM32CubeMonitor-RF, disconnect the USB cable, and reinsert it. 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 Bluetooth® Low Energy mode

#### 3.1 Presentation

#### 3.1.1 Panels

The panels are used to perform a specific operation. Each panel regroups different functions, as Figure 16 shows when the **ACI Commands** panel is selected.

**Figure 16. ACI Commands panel**

The main panels are **ACI Commands**, **Scripts**, **Beacon**, **RF Tests**, and **ACI Utilities**. They are detailed in the next sections.
3.2 Bluetooth® Low Energy stack

From STM32Cube_FW_WB_V1.14.0, there are some STM32WB_Copro_Wireless_Binaries variants. STM32CubeMonitor-RF needs to use stm32wb5x_BLE_Stack_full_extended_fw.bin.

For more information, refer to https://github.com/STMicroelectronics/STM32CubeWB/tree/master/Projects/STM32WB_Copro_Wireless_Binaries.

For the STM32WBxx, there exist four different variants for the Bluetooth® Low Energy stack, Full, Basic, Link Layer Only, and Link Layer Only Basic. Note that the default BLE_TransparentMode.bin firmware is based on the Basic feature variant and does not support all commands. A new build is required to support extra commands such as extended advertising. Refer to \Middlewares\ST\STM32_WPAN\ble\stack\doc\STM32WBA_BLE_Stack_User_Manual.html for more details.

3.3 ACI Commands panel

The application command interface (ACI) panel is used to send commands to the main device Bluetooth low Bluetooth® Low Energy stack. Categories group commands. These commands allow the user to configure the Bluetooth® Low Energy stack and activate communication with remote devices.
### 3.3.1 How to send an ACI command

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 in the parameters of the command. Default values are used otherwise.
3. Click on **SEND COMMAND**. The command is sent to the main device.

#### 3.3.2 Search function

The search icon is used quickly to select a command in the list:
- Click on the magnifier icon. A text box is created
- Type the name to search. As soon as a character is entered, matching commands are filtered in the list. The match might be any part of the command name. It is not necessary to start from the beginning.
- Click once on the command to select it (do not use double-click).
3.3.3 Filter usage
Features group and name the commands. Groups are:
- HCI
- HCI test
- HAL
- GATT
- GAP
- L2CAP
The picklist at the top of the area allows seeing only some groups to find the commands more easily. Click on Select all to see all the commands in the list.

3.3.4 How to fill parameters. Fixed field / editable field
Some parameters have fixed values and are not editable, while others are free or take only some values. The tool guides the user to fill in the parameters:
- Fixed parameter: this parameter is not editable. Specification or logic defines the value. This applies to the **length** value automatically computed by the tool.

**Figure 19. Fixed parameter**

| Parameter_Total_Len... | 0x03 |

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

**Figure 20. Editable parameter**

| Connection_Handle | 0x002A |

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

**Figure 21. Predefined values**

| Type | 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**

<table>
<thead>
<tr>
<th>Parameter_Total_Len...</th>
<th>0x03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection_Handle</td>
<td>0x002A</td>
</tr>
<tr>
<td>Type</td>
<td>0x01</td>
</tr>
</tbody>
</table>

### 3.3.5 Log functionalities

The log area is on the right part of the screen. 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 a *Command Complete*.

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

Some asynchronous events might come from the device and be 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 the complete text. It is possible to change the log area width to display longer texts.

Details

**Figure 24. More button**

Sometimes, all the information in a message does not fit in the area used for the log. The button in 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 a predefined text for the parameter values (opcode and others). The *Info* column provides some description of the 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 to paste it into other windows.

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

**Color code**

The logs use a color code to identify the device used and highlight errors.
A line with purple text shows that the status in the message is different from zero, which indicates an error.

**Figure 26. Purple error messages**

<table>
<thead>
<tr>
<th>No</th>
<th>Time</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>09:25:32.205</td>
<td>HCI_LE_CREATE_CONNECTION</td>
</tr>
<tr>
<td>9</td>
<td>09:25:32.212</td>
<td>Command Status</td>
</tr>
</tbody>
</table>

Log on a dark gray background comes from a second board. When the 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 the two devices.

**Figure 27. Gray second board messages**

<table>
<thead>
<tr>
<th>No</th>
<th>Time</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>09:27:34.029</td>
<td>HCI_READ_LOCAL_VERSION_INFORMATION</td>
</tr>
<tr>
<td>15</td>
<td>09:27:34.035</td>
<td>Command Complete</td>
</tr>
<tr>
<td>16</td>
<td>09:27:37.634</td>
<td>HCI_READ_LOCAL_VERSION_INFORMATION</td>
</tr>
<tr>
<td>17</td>
<td>09:27:37.672</td>
<td>Command Complete</td>
</tr>
</tbody>
</table>

**Update button**

When the *Update* tick box is not selected, the messages are not added to the log area. The line number continues to increase 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 the log history, untick the box that disables the 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.

The *RESET LOG* button resets the ACI log buffer used for log saving executed by the SAVE LOG action described just after

**Save log**

The *SAVE LOG* button allows saving the ACI log text file in csv format.
3.4 RF test panel

The RF test panel is used to perform the radio frequency tests on the main device. The RF tests are grouped into 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.
The first action after connecting a device is to select the mode to test and then click on the `SELECT TEST MODE` button.

When the user selects 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 30. Test mode selection**

```
<table>
<thead>
<tr>
<th>ACI Commands</th>
<th>Scripts</th>
<th>Beacon</th>
<th>RF Tests</th>
<th>ACI Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
Select test mode

- Transmitter test (TX)
- Packet error test (PER)
- Receiver test (RX)
```

**Figure 31. Change the test mode**

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

**Figure 32. Select the test mode**

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

### 3.4.1 Transmitter test mode (TX)

The TX mode is used to set the Bluetooth® Low Energy transmitter in emission. Two transmission modes are defined: *transmission of data, or emission of tone.*
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 parentheses. The advertising channel index does not follow the frequency order. Channels 37, 38, and 39 are the advertising channels. Refer to the BLUETOOTH SPECIFICATION version 4.2 [Vol 6 part B] chapter 1.4.1 for details.
4. Select the PHY modulation to use (the unsupported modulations by the device are not listed).
5. Click on the START TONE button.

The emission starts, the START TONE button is changed to STOP TONE, and Transmitting information is displayed.
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 payloads are available:

- A pseudo-random bit sequence 9 (PRBS9)
- A pattern of alternating bits 0b11110000
- A pattern of alternating bits 0b10101010
- A pseudo-random bit sequence 15 (PRBS15)
- A pattern of fixed bits 0b11111111
- A pattern of fixed bits 0b00000000
- A pattern of alternating bits 0b00001111
- A pattern of alternating bits 0b01010101

The *Length of data* picklist defines the sequence length. This is the length of the 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, the start button is changed to **STOP TX**, and Transmitting is displayed. The sequence is repeated until the 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 35. 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 (refer to Section 3.4.3: PER for details).
3.4.2 Receiver test mode (RX)

The Receiver mode is used to put the main device in Reception mode and count packets received.

**Figure 36. Receiver test mode**

**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 animation and the button changes to STOP RX.

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

If the number of transmitted packets is known, it might be entered manually in the Transmitted packet number. The Packet Error Rate (PER) is automatically computed (refer to Section 3.4.3: PER for details).

If the Get RSSI checkbox is selected, the tool performs an RSSI measurement.

**RSSI measurement:**

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

When the 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 display, with the blue button on the right (bar chart, arrows, or blue lines).
Figure 37. RSSI measurement

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 38. RSSI measurement graph

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

PER definition

The packet error rate (PER) is an indicator of the quality of transmission between the two devices. The measurement proposed in the tool covers the whole transmission chain from the transmitter to the 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. A high PER means that the transmission is not good.

\[
\text{PER} = 100 \times \frac{\text{Ntx} - \text{Nrx}}{\text{Ntx}} \%
\]

**Figure 40. PER definition**

Ntx: Number of packets sent. Nrx: Number of packets received. PER: Result in percent.
A bad PER might be an issue from the transmitter or the receiver and depends on parameters like the 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 the PER test. In this mode, two devices need to be connected to the computer:
- The first device under test (DUT)
- The second device acts as a packet generator (tester)

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

The configuration of the 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:

**Figure 41. PER tester connection**

- Plug the device into the computer (same requirements as the first device, refer to Chapter 2.2).
- Select the serial port to use in the picklist.
- Click on the CONNECT button.
The board information is displayed on the right.

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

Click on CONFIGURE TESTER to set the tester parameters:
• Select the TX power level with the picklist.
• Select the transmission frequency with the TX Frequency picklist.
• Select the length of the packet to send (same as the TX test).
• Select the content of the packet payload. For the PER test, it is recommended to use the Pseudo-Random bit sequence 9 reference pattern. Patterns containing only 0 or 1 must not be used for PER. Other patterns can be used.
• Select the PHY to use.

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

---

**Figure 43. PER tester configuration**

**Figure 44. DUT configuration**
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 might modify it.

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

**Figure 45. PER test parameters**

- **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 values separated by a 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 adds some RSSI measurement between each PER measurement. When activated, the tool performs a PER test for the measurement period, computes PER, and then makes an 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. The report is saved at the end of the tests.

When the option has been configured, click on the **START TEST** button:

- The DUT is set in Reception mode,

- Then the tester starts.
The button is changed to STOP and Testing… is displayed.

The test continues until the user presses the STOP button, or after all channels are 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 bar icon.

**Figure 46. PER and RSSI measurement graph**

---

### 3.5 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.
3.5.1 Launching scripts

Figure 47. Launching scripts

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 directly enter the file name.
- Click on the 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.
- The script is manually stopped with the Stop script button.
Script examples are provided with the tool, such as sample script, loop, and beacon creation.

For Windows, scripts are in the `C:\Users\Public\Documents\STMicroelectronics\STM32CubeMonitor-RF\scripts` (public documents) folder.

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

For macOS, it is inside the document folder provided in the setup package.
3.5.2 Script recording

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

Figure 49. Script recording buttons

Use the red button to start recording. Pause is inserted with the Add pause in script button. At the end of the recording, click the Stop button. The tool asks for the script name before saving.

3.5.3 Scripts modification

The 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 50. Sample script

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

# Wait a 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 ignored by the tool. Empty lines are skipped.
Other lines are commands. The line starts with the command name, followed by parameters in brackets separated by a semicolon.

### 3.5.4 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 51. Script report**

<table>
<thead>
<tr>
<th>Command</th>
<th>Sent</th>
<th>ACI status</th>
<th>ACI raw result</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCI_RESET</td>
<td>OK</td>
<td>0x00</td>
<td>[0x04,0x0E,0x04,0x01,0x03,0x0C,0x00]</td>
</tr>
<tr>
<td>ACI_HAL_SET_TX_POWER_LEVEL</td>
<td>OK</td>
<td>0x00</td>
<td>[0x04,0x0E,0x04,0x01,0x0F,0xFC,0x00]</td>
</tr>
<tr>
<td>ACI_HAL_TONE_START</td>
<td>OK</td>
<td>0x00</td>
<td>[0x04,0x0E,0x04,0x01,0x15,0xFC,0x00]</td>
</tr>
<tr>
<td>ACI_HAL_TONE_STOP</td>
<td>OK</td>
<td>0x00</td>
<td>[0x04,0x0E,0x04,0x01,0x16,0xFC,0x00]</td>
</tr>
</tbody>
</table>

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

- `Verdict_`
- Script name
- Current date
- Current time
- `.txt` extension

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

The *ACI status* column has the status of the ACI response. `0x00` is a success status while other values are errors.

At the end of script execution, a popup with the verdict (error found or finish successfully) is displayed and asks if the report must be saved:
If the user presses yes, the report is generated in the folder of the current script. If the user presses no, the report is not saved.

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

### 3.5.5 List of script 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 with `0x`. The optional parameters can be left empty. The length is dependent on the size parameter in the ACI command.

*Note*: The Command Packet Type, Opcode, and Parameter Total Length are filled in by the application. They must not be added to 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’s comment.
The OK button allows continuing the script.

Command: Pause *(user comment)*

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

### Figure 54. 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**

A loop can be used in the script to repeat some actions automatically.

**Loop usage**

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

- *Loop (count, 0, 5)*: This instruction indicates the beginning of the repeated section. The `count` variable is the name given to the counter. The first value is the start value and the second one is the end value. In this example, the counter `count` is increased from 0 to 5. There are six iterations.

- *EndLoop* indicates the end of the loop. If the counter reaches the end value, execution continues to the next line. If the counter does not reach the end value, it is updated, and execution goes back to the *Loop* instruction.

### Figure 55. Loop simple example

```
Loop (count; 1; 3)
Pause ("test the loop")
EndLoop
```
This script, given as an example in Figure 55, displays three times: test the loop.

**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 56. Loop second simple example**

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

The script in Figure 56 displays The loop counter is 1, then The loop counter is 2, and finally, The loop counter is 3.

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

If count = 10, 0xA replaces &[count].

**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 57. Loop decrement**

```
Loop {mycount; 3; 1}
```

In the countdown example in Figure 57, 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 58:

**Figure 58. 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 a different counter name.

**Figure 59. Nested loop**

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

The script provided as a nested loop example in Figure 59 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 60 generates the verdict shown in Figure 61:

**Figure 60. Example of loop script verdict**

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

**Figure 61. Display of loop script verdict**

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

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

In case of an 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 62. Script loop error**

3.6 OTA transfer

3.6.1 OTA presentation

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

In this section, the computer or device sending the data is named the *Source device*. 
The source device transfers the data and the OTA loader to the address requested by the user.

The implementation example does not include security in the transfer process. The users are expected to change their loader or application to perform the security verification based on their requirements.

The OTA process is described in the application note *Over-the-air application and wireless firmware update for STM32WB series microcontrollers* (AN5247), available on www.st.com. Read these 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 Bluetooth low Bluetooth® Low Energy services used.

### OTA loader

The OTA loader is the first application that starts at boot or reboot. OTA loader checks the boot conditions, and if the flash memory is empty.

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

The loader fits in the first six sectors of the flash memory. So, the block at the address 0x6000 is free and used to upload user data.

<table>
<thead>
<tr>
<th>Flash memory address:</th>
<th>Flash memory content:</th>
</tr>
</thead>
<tbody>
<tr>
<td>+0x0000</td>
<td>OTA bootloader</td>
</tr>
<tr>
<td>+0x1000</td>
<td>OTA bootloader</td>
</tr>
<tr>
<td>+0x…</td>
<td>OTA bootloader…</td>
</tr>
<tr>
<td>+0x6000</td>
<td>Free for user data</td>
</tr>
<tr>
<td>+0x7000</td>
<td>User application</td>
</tr>
<tr>
<td>+0x8000</td>
<td>User application</td>
</tr>
<tr>
<td>+0x…</td>
<td>User application…</td>
</tr>
</tbody>
</table>

- In the STM32WBxx sample code, the binary is stored at the 0x7000 address. In the STM32WBAxx, the binary is stored at the 0x7C00 address. The bootloader starts at this address after upload.

### 3.6.2 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 the OTA mode

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

Connection in OTA mode and data transfer

The source device first connects to the OTA loader and discovers the details of the service and characteristics to be able to transfer the data. Then the sequence is:
1. Configure the target device to send an indication to the source device.
2. Write in the target device the command to initiate the procedure, with the exact storage address.
3. Write each block of data. Depending on the optimized or not MTU size, the blocks are 20- or 248-byte long, and the binary must be transferred in many blocks.
4. At the end of the last block, write the confirmation that all blocks have been sent.
5. The source device waits for the reboot confirmation from the target.

3.6.3 Use the tool to perform an OTA update

The OTA function is available in the device menu in the menu bar. Click on the 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 Bluetooth® Low Energy devices and lists all the devices with OTA capabilities.
The tool provides an advertising filter to refine the search procedure with an advertising message.

**Table 2. Search filtering**

<table>
<thead>
<tr>
<th>Filter</th>
<th>Search method</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>No filter</td>
<td>Scans all Bluetooth® Low Energy devices and provides the list.</td>
<td>Some devices listed are not compatible with OTA.</td>
</tr>
<tr>
<td>Advertising filter</td>
<td>Scans all Bluetooth® Low Energy devices and provides a list of devices with ST OTA information.</td>
<td>Gives only the list of compatible devices.</td>
</tr>
</tbody>
</table>

To start the search, click on the **SEARCH FOR DEVICES** button.
The search procedure starts.

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

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

**Device and parameters selection**

After the search procedure, if one or more devices are found, the user selects the device with the picklist *Select device*. 
The picklist displays the list of boards found:

- For a device with Bluetooth® Low Energy characteristics:
  - Bluetooth® Low Energy address - Device name - OTA enabled
- For a device already in OTA mode:
  - Bluetooth® Low Energy address - Device name - OTA loader

Select the firmware target:

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

Select the device type:

- STM32WB5x/WB3x product lines
- STM32WB1x product line. Note that for this device type, only Target CPU1: M4 can be updated.
- STM32WBAxx product line. Note that for this device type, only Target CPU1 can be updated

The Optimize MTU size option allows the user to increase the ATT maximum transmission unit (MTU) from 20 to 248 bytes.

The image base address is the place where the binary file must be stored on the target device. It is a hexadecimal value and must be a multiple of 0x800 for STM32WB5x/WB3x, 0x1000 for STM32WB1x, or 0x2000 for STM32WBAxx to match with the flash memory.
sector. For the wireless stack, the address is the temporary location in the CPU1 user part area.

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

The configuration is finished and the 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 68. Configuring in OTA](image)

2. Second step: the transfer process to the OTA bootloader is performed. The data are transferred in blocks of 20 bytes. To avoid the overload of log windows, the log information related to block transfer is not displayed. Only the flow control event and errors are displayed.

3. A progress bar monitors the memory load.

![Figure 69. Progress bar](image)

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.

### 3.7 Beacon

#### 3.7.1 Beacon presentation

A beacon is an active device discoverable by other devices.

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

The data shared by the beacon are very small. A connected device receives them and the application on the device is notified of a beacon presence. The application uses the cloud to get more information and act accordingly.
When an application is informed of beacon proximity, it uses the beacon identification to request the web server more information about the beacon. The application gets information related to the geographical position of the beacon or the action to perform, like displaying commercial ads or starting an interactive application.

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

- iBeacon: This is Apple format. The beacon broadcasts fixed content, to identify easily the beacon.
- Eddystone UID: Google defines it. The beacon transmits fixed content (UID), the box that is a unique ID, referenced in the Google database to interact with applications.
- Eddystone URL: Another Google format. It provides a short URL, to use for the Physical web.
- Eddystone TLM: An additional beacon advertising information providing beacon information (battery status, temperature).
- Eddystone EID: Like UID, but broadcasts encrypted data to provide better security.

3.7.2 Beacon configuration methods:

STM32CubeMonitor-RF is used to generate and configure beacons. 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 the Advertising mode and configure the content of the advertising packet. The main device acts as a beacon until turned off.
The main advantage of this method is to configure quickly a beacon with a board in Transparent mode. The drawback is that the configuration is lost when the board is reset or powered off.

**Figure 72. Online beacon**

![Online beacon](image)

**Offline beacon**

The Offline beacon mode is used to prepare the configuration of a board not directly connected to the 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 the target flash memory at the 0x6000 address. The beacon firmware reads the data and configures the advertising block accordingly. Details of the configuration file are described in *Table 4: Beacon configuration format*.

The interest of the method is to have an 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 the configuration file must be transferred manually to the target device so it is less flexible than the Online mode.

**Figure 73. Offline beacon**

![Offline beacon](image)

**Selecting the Beacon mode**

The selection of the configuration mode is the first action to prepare the beacon. The user must choose the mode when selecting the beacon tab.
Select one of the two bullets and click on **SELECT CONFIGURATION MODE**.

### 3.7.3 Configuration of the beacon with STM32CubeMonitor-RF

To configure the beacon,

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

#### Common parameters

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

![Figure 75. Common parameters](image)

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

To save batteries, the power level of the beacon might be lowered, reducing consumption and visibility. Using high power extends the range of visibility but drains more power. The user needs to define the power level based on the 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 enough to estimate the real distance:

- Some beacons might transmit with high power, while others are using low power.
- The design of the beacon antenna might be 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 measure the received level at one meter. Then the beacon is reconfigured with the value measured at one meter in the Reference TX power level field.

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

- Set the address in the box and tick the public address to use the entered address.
- Tick the random address checkbox. A random address is used.
- If nothing is selected, the default public address of the board is used.

### iBeacon parameters

First, select the type: iBeacon (default choice)

The users must check the Apple website for information about the iBeacon structure and the conditions to use iBeacon for their project: [https://developer.apple.com/ibeacon/](https://developer.apple.com/ibeacon/)


The company code is a value based on the Bluetooth® Low Energy SIG group-assigned values. For iBeacon, the Apple value is used: 0x004C. The assigned values are available on the SIG website: [https://www.bluetooth.com/specifications/assigned-numbers/company-identifiers](https://www.bluetooth.com/specifications/assigned-numbers/company-identifiers).


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

When all parameters are updated, click on **CONFIGURE**. The data are ready for transfer (refer to *Configuration transfer*).
Eddystone UID parameters

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

- **NameSpace**, 10 bytes. Used to group some beacons in a logical pool. Google describes the way to generate the value. Refer to [https://github.com/google/eddystone/tree/master/eddystone-uid](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 the **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 the beacon is on, and the number of frames transmitted. The TLM information is not known by the tool. So, the firmware must manage directly by itself. Consequently:

- The TLM option is not used for the Online configuration mode.
- For Offline and OTA modes, a bit is set in the configuration file (refer to Appendix A).

Eddystone URL parameters

The Eddystone URL format is just sending a URL in the advertising message. To optimize space, the start and end of the URL might 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 a prefix. The URL is parsed, and if the end of the URL is encodable, the tool encodes it. A long URL does not work, it is advised to use the URL short service to get a short URL.

The TLM option is the same as the UID beacon.

Configuration transfer

The transfer depends on the selected configuration mode.
1. Online mode transfer configuration

**Figure 79. Online mode transfer configuration**

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

2. Offline mode transfer configuration

**Figure 80. Offline mode transfer configuration**

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 memory programmer, or any other tool.
3.8 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 their services and characteristics.

Figure 81. ACI Utilities panel

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

Figure 82. Select checkbox

3.8.1 Remote services discovering

The remote services discovery performs a scan of the remote devices in the area.
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 the SCAN button to start the discovery.

The search procedure starts, and it is possible to stop it using the STOP button.
If no remote device is found, the tool indicates No device found. Otherwise, the user chooses one of the devices found in the Select Device box.

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

- Click on the back button

Or connect to the selected remote device, by clicking on the CONNECT button.
If the connection fails, an error is displayed.

**Figure 89. Connection error**

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

**Figure 90. Connected icon**

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

**Figure 91. Services list**
The user can select a parameter and, depending on it, read, or write a value and be notified of the value change. Note that read and write long characteristics are not supported, or authenticated signed write.

To read a value, the user clicks on the READ button.

To write a value, the user enters the new value and clicks on the WRITE button.
There are two ways to be informed of a value change: either via the indicated method or a notification, depending on the method property supported by the remote device.

To receive an indication upon value change, the user can click on the *INDICATE* button.

To receive a notification upon value change, the user can click on the *NOTIFY* button.
Upon each change, a notification (resp. indication) is received and the new value is displayed. The user can be informed of multiple characteristic value changes at the same time. To stop the notification (resp. indication), the user can click on the _UN-NOTIFY_ button (resp. _UN-INDICATE_).

On disconnection, all registered notifications are removed.
3.8.2 Advertising

To activate the 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 the advertising interval.
7. Enter an optional target connection interval.
8. Click on the START ADVERTISING button to start the procedure.

The search procedure starts, and the advertising icon appears in blue. It is possible to stop it using the STOP ADVERTISING button.
The connect icon might appear in blue if a remote device connects. In that case, advertising stops.
4  OpenThread mode

4.1  Presentation

4.1.1  Panel

The OpenThread main panel is organized with three tabs, **Commands**, **Scripts**, and **Network Explorer**.

**Figure 101. OpenThread - Command tab**
The first two tabs have one common bottom area, the terminal area.

**Figure 103. OpenThread - Network Explorer tab**

- **Project**
- **Background**
- **Plaid (0x334)**
- **Channel 12**
- **Medium**
- **EXPLORER**
- **LOGGER**
The terminal area is used to show the messages exchanged between the application and the target. The commands sent to the target can be seen and the responses received from the target. Those messages can be cleared with the rubber icon.

The bottom line with a $ character is a command line. The user types the command with the parameters and presses <Enter> to send the command. The commands sent with this line are recorded in the 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 parameter area to fill the line, then the user can modify the line and send a command with the entering key. The commands list and parameters area are described in the 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.

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 and SEND COMMAND. The READ button sends the command without parameters to read the value. The SEND COMMAND button sends the command with parameters to the target.
The Start script recording and Add pause in script buttons allow saving a script. This part is described in Section 4.3: OpenThread scripts tab.

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

The command name and the definition are in the upper part of the command details area. Below is 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 to launch the script stored in a text file.

Figure 109. Scripts tab

The scripts use the same syntax as the Bluetooth® Low Energy scripts. The OpenThread specificities are described in this chapter. Consult the Bluetooth® Low Energy script description in Section 3.5: Scripts on page 41 for general information.
4.3.1 OpenThread script example

```
#STM32CubeMonitor-RF sample script
# OpenThread ping node script

#Pause command
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 Bluetooth® Low Energy, but the Send command is modified to send Thread® commands.
The OpenThread commands are sent with the *Send* instruction:

Send (OPENTHREAD_CMD_NAME Parameter1Value Parameter2Value).

The part inside the brackets is the command line to send.

### 4.4 Network Explorer tab

This feature can only be used if the DUT has the `Thread_Cli_cmd` firmware to be able to copy data from the UART to the OpenThread command-line interpreter. Refer to Section 2.2.1 for further information about firmware.

This tab is dedicated to the exploration and display of the network to which the DUT device is attached. The representation of the network is displayed in the central area. There are some basic control functions in the up-left corner of the pane. Just below, there is information on the selected node in an infobox plus logs of the exploration.

#### 4.4.1 Controls

![Figure 112. Auto connection functions](image)
The network explorer tab easily configures the panId and channel of the DUT device. The panId is entered in hexadecimal format, with no need to specify 0x. The value must be contained between 0x0000 and 0xffff. The value 0xffff means a nonconfigured panId. The channel is defined in decimal format and must be contained in [11;26]. The panId must be configured before configuring the channel. As from the STM32WB V1.14 version, the network key is initialized to a random value, so this parameter is set to 00112233445566778899aabbccddeeff. It is the default controller key used by the previous firmware.

For both parameters, if the filled value is in the wrong format, nothing is changed, and the actual value of the device remains displayed. Moreover, if a network exploration is ongoing neither parameter can be changed.

At the first connection of the DUT or when switching to the network explorer tab, the tool checks the current values of both parameters and displays them in the fields as information.

![Figure 113. Project and background management](image)

The two menu buttons on the top of the control area give control to the project itself and the background image.

The Project menu proposes three choices:
1. New choice cleans the current session by resetting the display area and stopping the ongoing exploration if there is one.
2. Open choice opens a file explorer to choose a backup of a project to use in the session. When a project is loaded from the Open choice box, there is a two-step process:
   - The saved image is first restored as the background of the right area.
   - Then when a scan is started, if a saved device in the project is detected, it is instantaneously displayed in its last place, with its former nickname. This association is based on the unique MAC addresses of the devices.
3. Save choices saves the current project. In this backup, there is the background image, the location of the icons on that image, and the nicknames of the devices.

The Background menu allows either to:
- Remove the background image.
- Open a file explorer to put an image in the display area as a background.

![Figure 114. Explore and size choice controls](image)

Once the DUT device is connected to a Thread® network, the EXPLORE button starts the network exploration sequences. It turns to STOP when the exploration is ongoing.

The Choice box at the left of the EXPLORE button allows choosing the size of the icon between three standard sizes: Small, Medium, and Large. It can be changed at any time. The size of the icons is adapted according to the dimensions of the background image.
4.4.2 Display area

Figure 115. Display area

The result of the network exploration is displayed with icons representing the devices of the network and their links. Each icon gives three types of information:

1. The color of the borders and the logo (pink for a Leader, cyan for a Router, and green for a Child) gives the role.
2. The number on the right side of the logo gives the ID.
3. The eventual nickname is written above the logo.
In the display area, it is possible to make several kinds of movement:

- Holding the left click of the mouse can move an icon everywhere inside the right area. If dragged on another icon, it turns gray and is automatically replaced if dropped on another icon to avoid overlays.
- Zoom in or out is done with the mouse wheel. The motion is centered on the mouse pointer.
- The whole content of the right area can be moved by holding the right click of the mouse. There are constraints to this movement though because what defines the background of the area (imported image or default blank background) cannot go completely off the area.
- A double-click (left) anywhere on the area centers the background and restores the zoom x1.
4.4.3 Infobox

Figure 117. Infobox

An infobox can be instantiated just below the control area by clicking on the concerned icon. It allows modification of the node nickname and indicates its role, ID, and MAC address.

4.4.4 Log area

Figure 118. Log area

The log area is in the bottom-left part of the tab. It prints the last two exploration results of the network in written form. This area is updated after each new exploration.
5 802.15.4 RF test mode

5.1 Presentation

The RF test 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 sets the device in reception.
- The PER test sets the device in reception and one additional device is used as a packet generator.

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

To change the mode, it is necessary to come back to this panel. There is a Back key and a breadcrumb link in each test panel to come back to this Test mode selection panel.
5.2 Transmitter test mode (TX)

This test mode configures the 802.15.4 device in emission. Two TX modes are available, Frame and Continuous modulated modes.

The user must:
- Select the power level (+6 dBm to -21 dBm).
- Select the TX frequency (Channel 11 - 2405 MHz to Channel 26 - 2480 MHz).
- Select the TX mode, Frame, Continuous modulated, or Continuous wave.

5.2.1 Frame mode

This mode allows the user to send a MAC frame. Either the user selects one frame available in the picklist or it fills the field.
Note: In the picklist, there is the frame required for the certification test of TX to RX turnaround time.

The help information is visible with a mouse over the question mark, on the right side of the field.

**Figure 122. Help frame information**

The START TX button is enabled when the frame in the field is valid. Press the start button to launch the transmission, the button is disabled until the frame is transmitted.

### 5.2.2 Continuous modulated mode

This mode transmits a continuous signal where pseudo-random binary sequence (PRBS) data is sent over PHY PDU.

Press the START TX button to launch the transmission. The label of the button is switched to STOP TX and allows the user to stop the transmission.

**Figure 123. Continuous modulated test mode**
5.2.3 Continuous wave mode

This mode transmits a continuous signal with no modulation. Press the START TX button to launch the transmission. The label of the button is switched to STOP TX and allows the user to stop the transmission.

5.3 Receiver test (RX) mode

This test mode configures the device in reception and requires an external generator.

Four tests are available:
1. PER (packet error rate): Requires an external frame generator.
2. LQI (link quality indicator): Requires an external frame generator.
3. ED (energy detection): Requires an external continuous wave generator.
4. CCA (channel clear assessment): Requires an external frame generator.

Note: LQI, ED, and CCA tests are available with PHY valid CLI version v1.8.1 and upper.

Figure 124. Receiver test mode
5.3.1 Packet error rate (PER) test

This test requests to use an external frame generator and follow the procedure below:

- Select the channel to be tested.
- Press the START RX button; the device enters Receiver mode and the button switches to STOP RX.
- With one external generator, send the frames to test in the frequency selected above.
- On the application side, the frames received appear in a gray part. This part is available from PHY valid CLI version v1.8.0 and upper.

   Figure 125. PER frames received

   • Once the frames are completely sent, press the STOP RX button. The three fields NB frames, RSSI, and LQI are filled. The button switches to START RX.

   Figure 126. PER frame reception completed

   • According to the number of frames sent, the PER can be calculated with the value in the NB frames received field.
5.3.2 Link quality assessment (LQI) test

This test requests to use an external frame generator and follow the procedure below:

- Select the channel to be tested.
- Either the measurement is done in continuous (default mode) or step-by-step checking the Single measurement item.
- With one external generator, send the RF signal to test in the frequency selected above.
- Press the START RX button to launch the LQI measurement.
- The instantaneous measurement appears on the right side and is also reported in the chart.

Figure 127. LQI measurement

5.3.3 Energy detection (ED) test

This test requests to use an external frame generator and follow the procedure below:

- Select the channel to be tested.
- Either the measurement is done in continuous (default mode) or step-by-step checking the Single measurement item.
- With one external generator, send the RF CW signal in the frequency selected above.
- Press the START RX button to launch the ED measurement.
The instantaneous measurement appears on the right side and is also reported in the chart.

**Figure 128. ED measurement**

### 5.3.4 Channel clear assessment (CCA) test

This test requests to use an external frame generator and follow the procedure below:

- Select the channel to be tested.
- Either the measurement is done continuously (default mode) or step by step when checking the *Single measurement* item.
- With one external generator, send the RF signal to test in the frequency selected above.
- Press the *START RX* button to launch the CCA measurement.
The instantaneous measurement appears on the right side and is also reported in the chart.

**5.4 Packet error rate (PER) mode**

This mode configures the device in reception and one other device to play the role of the 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 the number of frames sent
  \[ \text{PER} = \frac{100 \times (\text{Number of frames sent} - \text{number of frames received})}{\text{Number of frames sent}} \]

Four steps are necessary:
- Connect the additional device to play the role of a packet generator (tester).
- Configure the parameters of the tester.
- Configure the parameters of the device under test (DUT).
- Configure the measurement.
5.4.1 Connecting the additional device to play the role of a packet generator (tester).

Figure 130. Packet tester connection

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

When the second device is connected, it is not possible to change the mode. First, the user needs to disconnect the device and then press the back button.

Click on CONFIGURE TESTER to set the tester parameters.
5.4.2 Configure the parameters of the tester.

Figure 131. PER tester configuration

The user must:

- Select the power level in the *Power Level* picklist.
- Select the frequency in the *TX frequency* picklist. This parameter is used only for the Single measurement mode. It is not used for Continuous or Multiple-channel modes. It is applied to the tester device.

Click on *CONFIGURE DUT* to set the Device Under Test configuration.
5.4.3  Configure the parameters of the device under test (DUT).

Figure 132. DUT configuration

The user must:

- Select the frequency in the **RX frequency** picklist. It is the frequency of the DUT.

Click on **CONFIGURE PARAM** to set the test configuration:
5.4.4 Configure the measurement.

![Figure 133. PER test parameters](image)

Three measurement modes are available:

- **Single measurement** measures once the frame number is defined. The frequency of the tester is the one defined in the PER tester configuration panel (TX frequency). The frequencies of DUT are as defined in the DUT configuration panel.

- **Continuous measurement** repeats the measurement on frame number until the user presses the Stop Test key. The DUT and tester frequencies are identical. They are defined once in the DUT configuration panel.

- **Multiple channels** measure the frequency defined in the Fill channel list field. The default values are 11-26, which means that all channels are in the range of 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 the Stop Test key.

The result of continuous and multiple-channel measurements can be saved in a csv file. The user must check the Save test verdict in file checkbox and define the file name by the SELECT FILE key before starting the test.
Three display modes are available:

1. **Standard display**

   There are the PER and RSSI values and LQI for one channel.

   ![Figure 134. Standard display](image)

2. **Chart display**

   In the same chart, there are the PER and RSSI values and LQI for channels that the user defines.

   ![Figure 135. Chart display](image)

---

**Table 3. Measurement setting**

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Single</th>
<th>Continuous</th>
<th>Multiple channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous measurement checkbox</td>
<td>Unchecked</td>
<td>Checked</td>
<td>Checked</td>
</tr>
<tr>
<td>Multiple-channel checkbox</td>
<td>Unchecked</td>
<td>Unchecked</td>
<td>Checked</td>
</tr>
<tr>
<td>Save test verdict in file checkbox</td>
<td>Not available</td>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>

---
3. Large display

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

**Figure 136. Large PER display**

![Large PER display](image)

**Figure 137. Large RSSI display**

![Large RSSI display](image)

**Figure 138. Large LQI display**

![Large LQI display](image)
6 802.15.4 sniffer

6.1 Presentation

The 802.15.4 sniffer allows the user to detect and log 802.15.4 packets between the devices communicating in the neighborhood of the sniffer device. Packets captured by the device are logged and formatted in a readable format, thanks to Wireshark, an external free software tool.

6.2 Prerequisite

6.2.1 Sniffer device

To configure the device as a sniffer, refer to Section 2.2.3: VCP device. Once done, connect the STM32WBx5 Nucleo board to the host computer using the USB_USER connector. Make sure that the 5 V sources jumper connector is plugged into the USB MCU.

6.2.2 Wireshark

Install Wireshark v2.4.6 or later, available from http://www.wireshark.org, and add the installation path to the path environment variable if it is not already done. Once done, the user must copy the Python™ sniffer script stm32cubeMonRf_sniffer.py and the associated stm32cubeMonRf_sniffer.bat file in the Wireshark extcap directory. Files are available in the sniffer directory where the tool is installed, by default for Windows: \Program Files (x86)\STMicroelectronics\STM32CubeMonitor-RF\sniffer. The Wireshark extcap path is available in the Help/About Wireshark menu under the Folders tab. Under macOS and Linux, note that the stm32cubeMonRf_sniffer.py file must have execute permission.

6.2.3 Python™

Install Python™ v2.7.x or later available from https://www.python.org/downloads. Then add the installation path to the path environment variable if it is not already done. The user also needs to install the Python™ serial port extension, pyserial, available from https://pypi.org/project/pyserial.
6.3 Setup verification

The sniffer can be invoked using the 802.15.4 SNIFFER button available on the welcome screen or through the Settings/mode menu available in the menu bar. In both cases, the tool checks that the prerequisites are fulfilled.

If this is not the case, the user is asked to correct it. Otherwise, the following pop-up window is displayed. To launch the sniffer, click on the LAUNCH button.

Figure 139. Prerequisite check
6.3.1 Sniffer launch

Once Wireshark is launched, the user is proposed to choose the interface to sniff.

Figure 140. Wireshark interfaces

6.3.2 Select interface

Choose the interface corresponding to the device configured for sniffing by clicking on the wheel.

Figure 141. Wheel
6.3.3 Channel configuration

The user is asked to choose the channel to be sniffed.

Figure 142. Channel choice

6.3.4 Sniffing start

Once the channel is selected, click on start. The sniffed packet list appears at the top of the tool, the details of the selected packet in the middle, and the packet byte in the bottom.

Figure 143. Sniffing
Appendix A  Beacon configuration format

The beacon configuration file is binary. Its content is explained in Table 4.

Table 4. Beacon configuration format

<table>
<thead>
<tr>
<th>Byte #</th>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 the address type is 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>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>27-Nov-2017</td>
<td>1</td>
<td>Initial version</td>
</tr>
</tbody>
</table>
| 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 Bluetooth® Low Energy 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 the 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 are reported in the application note. |
| 30-Mar-2020 | 6        | Added:  
  - STM32WB35 support with updated paths  
  - Section 4.4 on Thread® network exploration feature |
| 12-Nov-2020 | 7        | Updated:  
  - Tool version 2.6.0  
  - Six new sections:  
    - Section 5.2.1: Frame mode  
    - Section 5.2.2: Continuous modulated mode  
    - Section 5.3.1: Packet error rate (PER) test  
    - Section 5.3.2: Link quality assessment (LQI) test  
    - Section 5.3.3: Energy detection (ED) test  
    - Section 5.3.4: Channel clear assessment (CCA) test  
  - dealing with all the applicable tests in the Transmitter test mode (TX) and Receiver test mode (RX) |
<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>08-Feb-2021</td>
<td>8</td>
<td>All modifications linked to the new feature 802.15.4 sniffer&lt;br&gt;Updated:&lt;br&gt;– Introduction&lt;br&gt;– Section 1.2: Welcome screen with Figure 1&lt;br&gt;– Section 2.2.3: VCP device&lt;br&gt;– Figure 61, Figure 62, Figure 65, Figure 105, and Figure 114&lt;br&gt;Added:&lt;br&gt;– Section 6: 802.15.4 sniffer with Figure 141 to Figure 145</td>
</tr>
<tr>
<td>22-Jul-2021</td>
<td>9</td>
<td>All modifications linked to the new tool version 2.8.0&lt;br&gt;Updated:&lt;br&gt;– Section 2.2: VCOM / UART connection&lt;br&gt;– Section 3.3.3: PER&lt;br&gt;– Section 3.4.3: Scripts modification&lt;br&gt;– Section 3.5.3: Use the tool to perform an OTA update&lt;br&gt;– Section 3.6.3: Configuration of the beacon with STM32CubeMonitor-RF&lt;br&gt;– Figure 61, Figure 62, Figure 65, and Figure 80 to Figure 82</td>
</tr>
<tr>
<td>07-Jul-2022</td>
<td>10</td>
<td>Updated:&lt;br&gt;– Section 4.4: Network Explorer tab with the default master key of the former firmware&lt;br&gt;– Section 2.2.1: VCOM connection to Section 2.2.3: VCP device linked to the added section below&lt;br&gt;Added:&lt;br&gt;– Section 3.2: Bluetooth® Low Energy stack</td>
</tr>
<tr>
<td>29-Nov-2022</td>
<td>11</td>
<td>All modifications linked to the new tool version 2.10.0&lt;br&gt;Added:&lt;br&gt;– Section 2.3: Opening COM regarding connection error in the wrong mode&lt;br&gt;Updated:&lt;br&gt;– Section 3.2: Bluetooth® Low Energy stack with Table 1&lt;br&gt;– Section 3.5.1: Launching scripts path&lt;br&gt;– Figure 72: Selecting the Beacon mode&lt;br&gt;Removed:&lt;br&gt;– Beacon OTA configuration including former Figure 72&lt;br&gt;– Configuration transfer OTA mode scan transfer including former Figure 80 to Figure 82</td>
</tr>
</tbody>
</table>
Table 5. Document revision history (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>06-Mar-2023</td>
<td>12</td>
<td>All modifications linked to the new tool version 2.11.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Support to STM32XWBxx microcontrollers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Save log in Section 3.3.5: Log functionalities with new Figure 28 and Figure 29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Section 5.2.3: Continuous wave mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Figure 2, Figure 6, Figure 23, Figure 97, and Figure 120</td>
</tr>
<tr>
<td>04-Jul-2023</td>
<td>13</td>
<td>All modifications linked to the new tool version 2.12.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed Table 1 Bluetooth® Low Energy stack and STM32CubeMonitor-RF feature compatibility</td>
</tr>
<tr>
<td>23-Nov-2023</td>
<td>14</td>
<td>All modifications linked to the new tool version 2.13.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Applicable tool version quoted in Introduction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Figure 2, Figure 3, Figure 63, Figure 64, and Figure 67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Source code paths in Section 2.2.1: VCOM connection</td>
</tr>
</tbody>
</table>
IMPORTANT NOTICE – 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 acknowledgment.

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, 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.

© 2023 STMicroelectronics – All rights reserved