

Configuration methods for BQB RF-PHY testing and SRRC fixed frequency testing for TI Bluetooth products



Albin Zhang

Wireless Connectivity Solution

ABSTRACT

This document was translated from a simplified Chinese source. (ZHCADW2A)

The purpose of this article is to guide customers on how to prepare the tested device and how to configure it through the tool for Bluetooth RF-PHY certification and State Radio Regulatory Commission (SRRC) certification testing.

This article uses TI's latest-generation CC2340R5 and SDK7.40 version as examples, but the information is equally applicable to other Bluetooth products and SDK versions. The methods discussed apply to all TI Bluetooth-enabled products, including CC264x, CC265x, CC135x, CC267x and CC23xx families of products.

The article does not cover the code compilation and download processes in detail, as they are all available at TI's [SimpleLink Academy](#).

Table of Contents

1 Prepare for the test	2
1.1 Download the necessary kits and tools.....	2
1.2 Test file.....	2
1.3 Test tools.....	2
1.4 Hardware environment.....	3
2 Bluetooth RF-PHY test mode	5
2.1 Test specification command.....	5
2.2 Test with Btool.....	5
2.3 Loopback signaling test of Bluetooth tester.....	8
3 SRRC fixed frequency test	10
3.1 New SRRC rules and applicable terms.....	10
3.2 Test with Btool.....	11
3.3 Implement the test with SmartRF Studio.....	12
4 Summary	13
5 Reference documentation	13

List of Figures

Figure 1-1. UART configuration for LP_EM_CC2340R5.....	2
Figure 1-2. Interface for default UART.....	3
Figure 2-1. SIG test command.....	5
Figure 2-2. Non-signaling connection diagram.....	6
Figure 2-3. Signaling connection diagram.....	9

1 Prepare for the test

The test code used for Bluetooth SIG RF-PHY testing and SRRC fixed frequency testing is the routine - host_test_app. A host computer tool sends 2-wire HCI commands to the chip to configure special modes.

1.1 Download the necessary kits and tools

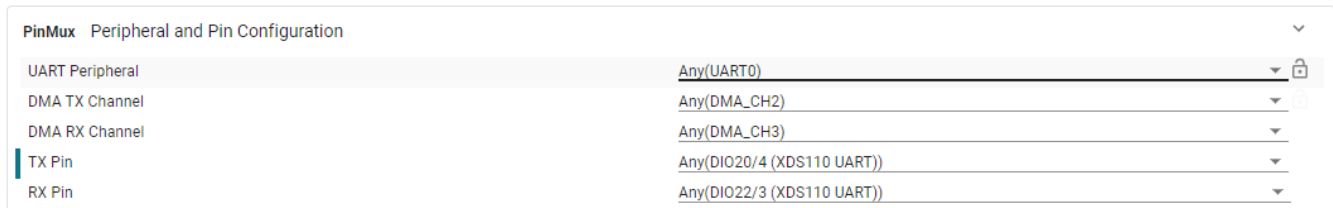
1. The software project, binary files, and the Btool utility for sending HCI commands are all available upon installation of the SDK (<https://www.ti.com/tool/SIMPLELINK-LOWPOWER-SDK>). Download them from the relevant device product catalog.
2. If IDE software CCS is not installed (<https://www.ti.com.cn/tool/cn/CCSTUDIO>), the binaries can be downloaded directly with Uniflash (<https://www.ti.com.cn/tool/cn/UNIFLASH>).

1.2 Test file

TI provides a pre-compiled host_test_app.hex file tailored to the interfaces and RF configurations of the corresponding product's development board. If you install SDK in the default directory, it is usually stored in the following directory:

C:\ti\simplelink_lowpower_f3_sdk_7_40_00_64\examples\rtos\LP_EM_CC2340R5\ble5stack\hexfiles

TI LP-EM-CC2340R5 development board is configured as follows, that is, DIO20 and DIO22 are used as UART interface.



Wireless MCU IO block

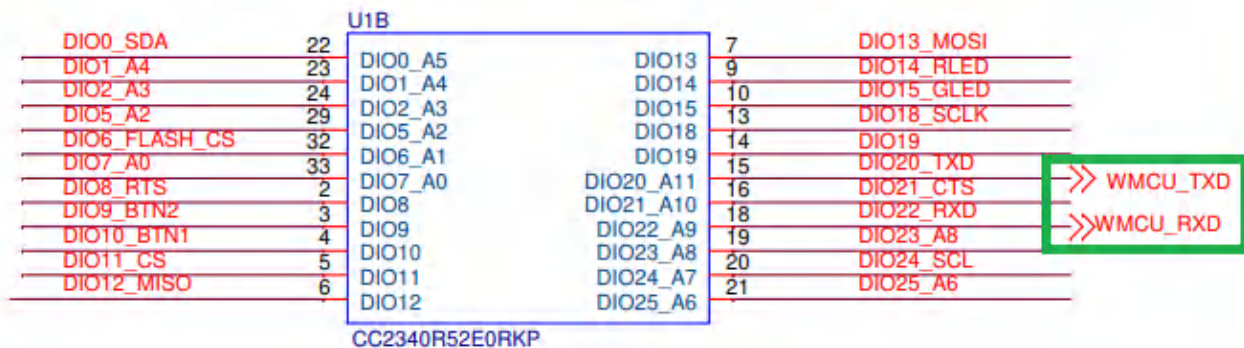


Figure 1-1. UART configuration for LP_EM_CC2340R5

If the user's hardware design changes the mapping for UART, the pre-compiled HEX files are not available and the project files need to be recompiled based on the custom design. The routine is described as follows:

C:\ti\simplelink_lowpower_f3_sdk_7_20_00_29\examples\rtos\LP_EM_CC2340R5\ble5stack\host_test

1.3 Test tools

The tool recommended by TI is Btool, which is available in the installation directory of the respective SDK.

C:\ti\simplelink_lowpower_f3_sdk_7_40_00_64\tools\ble5stack\btool

If users develop their own tools or use other UART serial tools, they may copy the dumped strings captured from Btool log to send the command. The following example demonstrates sending a SIG-defined transmission test packet.

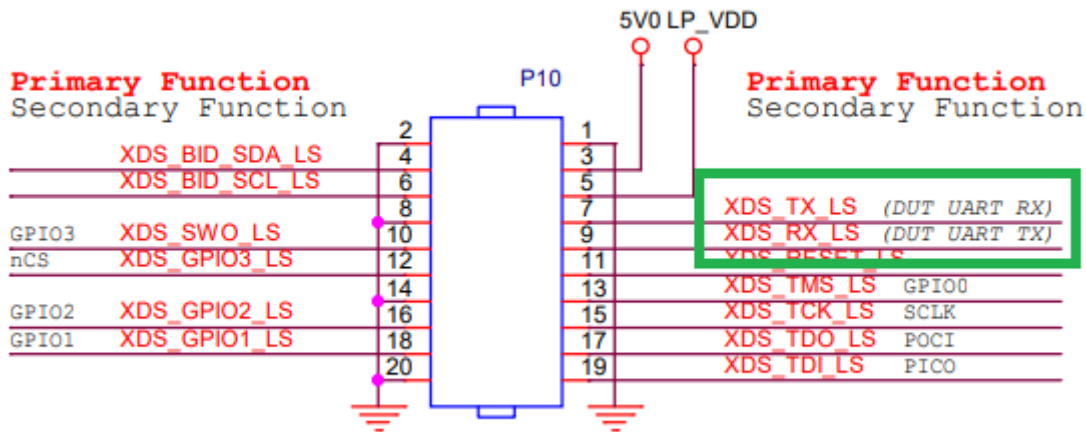
```

-----
[6] : <Tx> - 11:52:38.383
-Type : 0x01 (Command)
-OpCode : 0x201E (HCI_LE_TransmitterTest)
-Data Length : 0x03 (3) byte(s)
TX Channel : 0x00 (0)
TestData Length: 0x25 (37)
TestData : 02
Dump(Tx):
0000:01 1E 20 03 00 25 02 .. ..%.
-----
  
```

1.4 Hardware environment

Unlike SmartRF studio, which uses JTAG, the tests discussed herein are UART-based HCI command formats. Therefore, customers must reserve test points for UART interface when designing the circuit boards.

In addition to JTAG, TI's latest emulator, [LP-XDS110ET](#), comes with a UART-USB adapter. This emulator can be used as a UART-USB adapter for connection to a host computer or Bluetooth tester. This emulator also includes a level-shifting circuit. If the interface level of the device under test differs from the default, you can select the appropriate setting depending on the jumper caps.



20-pin LP-EM Debug Connector

Together with a LP-EM LaunchPad (sold separately) it allows:

- Debugging and programming
- Communicating to the host via UART
- Powering the LP-EM board

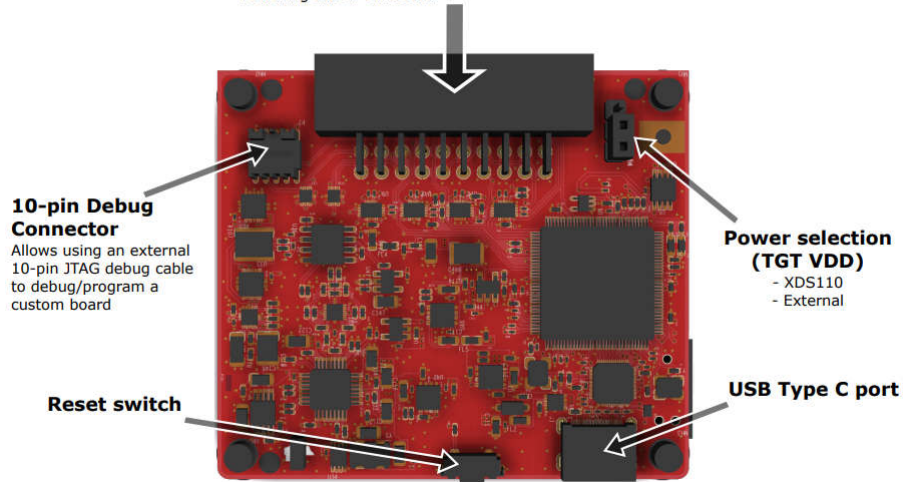


Figure 1-2. Interface for default UART

2 Bluetooth RF-PHY test mode

A product that claims to support Bluetooth at the time of its launch needs to pass Bluetooth certification testing. RF performance is related to the product hardware, so RF-PHY needs to be fully tested in the test lab.

2.1 Test specification command

The core specifications and test specifications for Bluetooth are available for download on SIG website.

[Specifications | Bluetooth® Technology Website](#)

BT5.4 is used in the following example for illustration. The following figure shows commands and return events for HCI/2-wire UART for RF testing in DTM mode. Specific parameter definitions can be found in the specifications. It is not described in detail in this article.

RF Test command / event	HCI command / event	2-wire UART command / event
LE_Transmitter_Test command	HCI_LE_Transmitter_Test command	LE_Transmitter_Test command
LE_Receiver_Test command	HCI_LE_Receiver_Test command	LE_Receiver_Test command
LE_Test_End command	HCI_LE_Test_End command	LE_Test_End command
LE_Status event	HCI_Command_Complete event	LE_Test_Status event
LE_Packet_Report event	HCI_Command_Complete event	LE_Packet_Report event

Figure 2-1. SIG test command

2.2 Test with Btool

This method can be used for instruments that do not support signaling mode and for laboratory analysis issues.

The connection diagram is shown below:

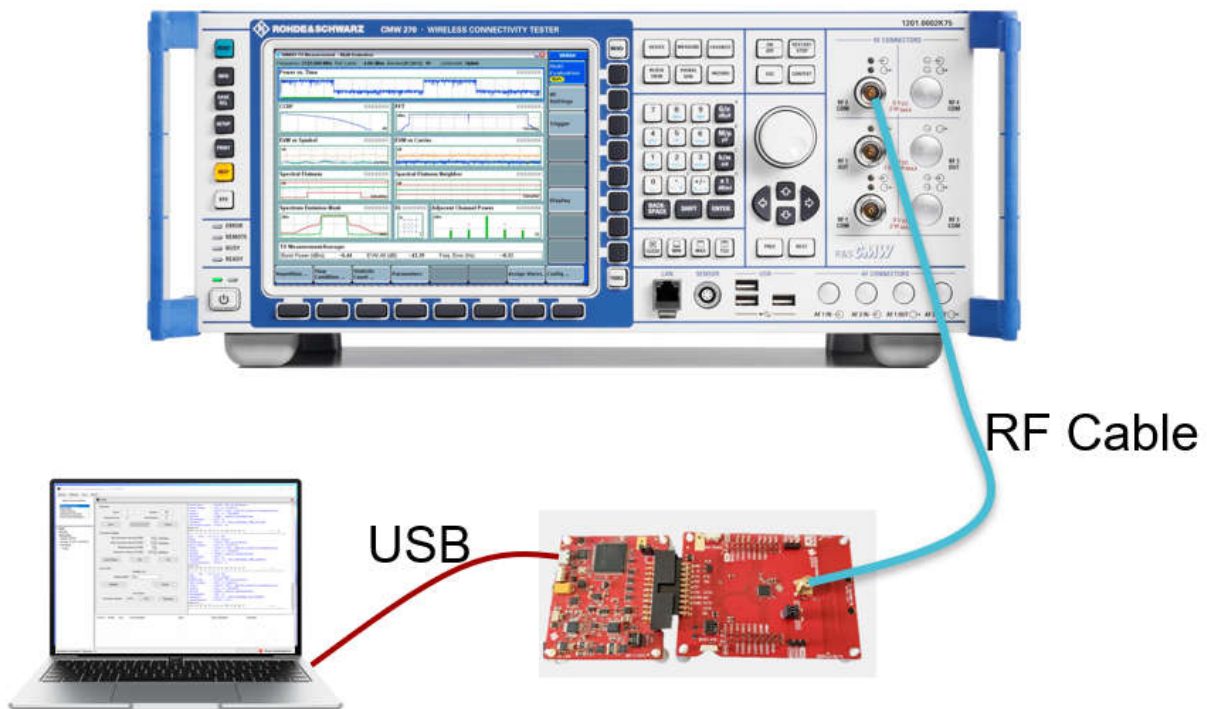
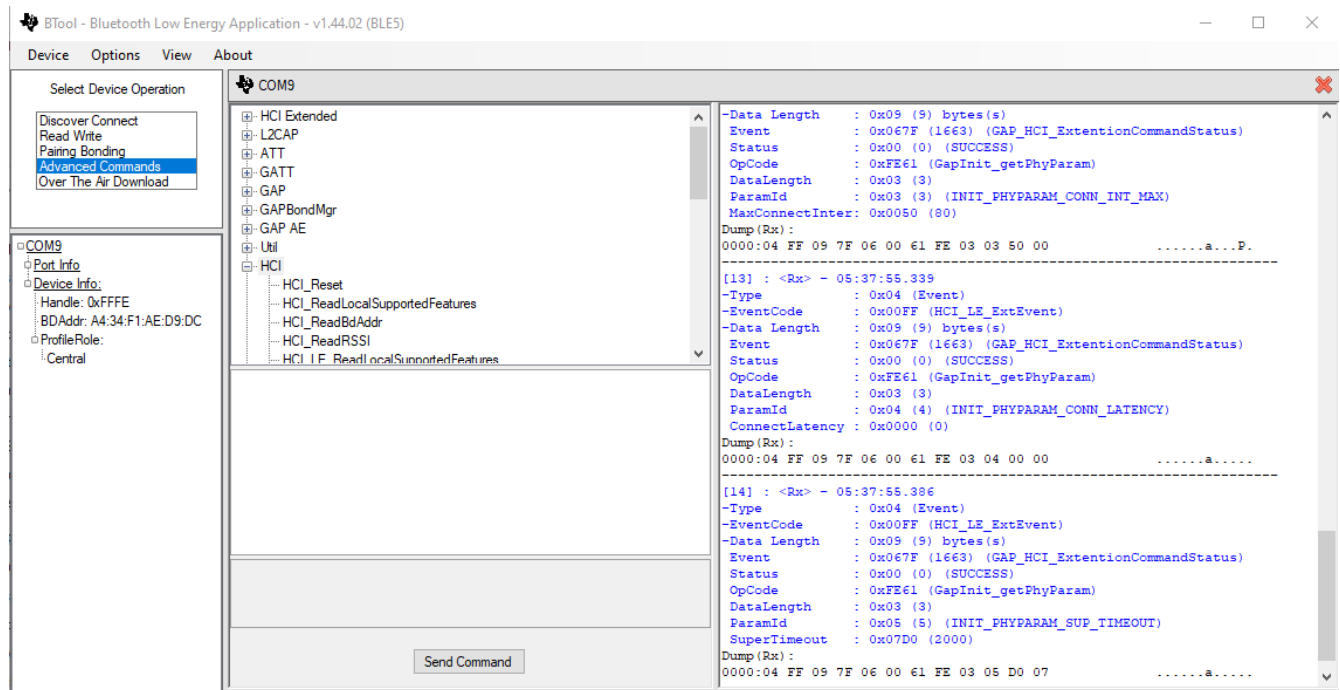


Figure 2-2. Non-signaling connection diagram

Testing in DTM mode uses the HCI command set defined by the Bluetooth specifications, as shown in the figure below:



2.2.1 Transmit HCI_LE_Transmitter_Test

Following BT5, SIG released additional physical layers, including 1M, 2M, and Coded PHY. opCode, matching with it, is 0x2034 command shown in the following figure. Use [HCI_LE_EnhancedTransmitter_Test](#) command.

The specific parameters must be aligned with the test item requirements on the instrument side. For details, refer to [RF-PHY Test Suite](#).

The screenshot shows the BTool interface with the following details:

- Device:** COM9
- Select Device Operation:** Advanced Commands
- Device Info:**
 - Handle: 0xFFFF
 - BDAddr: A4:34:F1:AE:D9:DC
 - ProfileRole: Central
- Misc:**
 - opCode: 0x2034
 - txChannel: 0
 - testDataLength: 37
 - dtmPktPattern: Dtm_PRBS9
 - txPhy: Phy_1_Mbps
- txPhy:** TX PHY Type (1 Byte) - Indicates the TX PHY type to use.
- Send Command:** Button
- Command Log:**
 - [38] : <Tx> - 03:25:59.593
 - Type: 0x01 (Command)
 - OpCode: 0x201F (HCI_LE_TestEnd)
 - Data Length: 0x00 (0) byte(s)
 - Dump (Tx): 0000:01 1F 20 00
 - [39] : <Rx> - 03:25:59.609
 - Type: 0x04 (Event)
 - EventCode: 0x000E (HCI_CommandCompleteEvent)
 - Data Length: 0x06 (6) bytes(s)
 - Packets: 0x01 (1)
 - OpCode: 0x201F (HCI_LE_TestEnd)
 - Status: 0x00 (0) (SUCCESS)
 - NumOfPackets: 0x0000 (0)
 - Dump (Rx): 0000:04 0E 06 01 1F 20 00 00 00
 - [40] : <Tx> - 03:26:14.056
 - Type: 0x01 (Command)
 - OpCode: 0x2034 (HCI_LE_EnhancedTransmitterTest)
 - Data Length: 0x04 (4) byte(s)
 - TxChannel: 0x00 (0)
 - TestData: 0x25 (37)
 - DtmPktPattern: 0x00 (0) (Dtm_PRBS9)
 - TxPhyType: 0x01 (1) (Phy 1 Mbps)
 - Dump (Tx): 0000:01 34 20 04 00 25 00 01

2.2.2 Receive HCI_LE_Receiver_Test

The received command is shown in the figure below and opCode is 0x2033. It is typically used when manually testing sensitivity.

The screenshot shows the BLE5 application window with the 'Advanced Commands' menu selected. The command list on the left includes 'HCI_LE_EnhancedReceiverTest'. The main window displays the command details and the log output. The log shows the following sequence of events:

```

[46] : <Tx> - 03:44:09.599
-Type : 0x01 (Command)
-OpCode : 0x201F (HCI_LE_TestEnd)
-Data Length : 0x00 (0) byte(s)
Dump (Tx):
0000:01 1F 20 00
-----
[47] : <Rx> - 03:44:09.615
-Type : 0x04 (Event)
-EventCode : 0x000E (HCI_CommandCompleteEvent)
-Data Length : 0x06 (6) bytes(s)
Packets : 0x01 (1)
OpCode : 0x201F (HCI_LE_TestEnd)
Status : 0x00 (0) (SUCCESS)
NumOfPackets : 0x0000 (0)
Dump (Rx):
0000:04 0E 06 01 1F 20 00 00 00
-----
[48] : <Tx> - 03:44:11.420
-Type : 0x01 (Command)
-OpCode : 0x2033 (HCI_LE_EnhancedReceiverTest)
-Data Length : 0x03 (3) byte(s)
RxChannel : 0x00 (0)
Phy : 0x01 (1) (Phy 1 Mbps)
DtmModeIndex : 0x00 (0) (Standard Modulation Index)
Dump (Tx):
0000:01 33 20 03 00 01 00
  
```

2.2.3 Test End LE_Test_End command

Before switching commands to transmit or receive parameters, you must first execute the Test End instruction in which opCode is 0x201F. When receiving a test, the return value of the instruction also includes the number of packets received - LE_Packet_Report. The return value result is used in receiving the PER's calculation.

The screenshot shows the BLE5 application window with the 'Advanced Commands' menu selected. The command list on the left includes 'HCI_LE_TestEnd'. The main window displays the command details and the log output. The log shows the following sequence of events:

```

[14] : <Rx> - 05:49:48.486
-Type : 0x04 (Event)
-EventCode : 0x00FF (HCI_LE_ExtEvent)
-Data Length : 0x09 (9) bytes(s)
Event : 0x067F (1663) (GAP_HCI_ExtensionCommandStatus)
Status : 0x00 (0) (SUCCESS)
OpCode : 0xFFE1 (GapInit_getPhyParam)
DataLength : 0x03 (3)
ParamId : 0x05 (5) (INIT_PHYPARAM_SUP_TIMEOUT)
SuperTimeout : 0x07D0 (2000)
Dump (Rx):
0000:04 FF 09 7F 06 00 61 FE 03 04 00 00
-----
[15] : <Tx> - 05:49:55.369
-Type : 0x01 (Command)
-OpCode : 0x201F (HCI_LE_TestEnd)
-Data Length : 0x00 (0) byte(s)
Dump (Tx):
0000:01 1F 20 00
-----
[16] : <Rx> - 05:49:55.388
-Type : 0x04 (Event)
-EventCode : 0x000E (HCI_CommandCompleteEvent)
-Data Length : 0x06 (6) bytes(s)
Packets : 0x01 (1)
OpCode : 0x201F (HCI_LE_TestEnd)
Status : 0x0C (12) (CMD_DISALLOWED)
NumOfPackets : 0x0000 (0)
Dump (Rx):
0000:04 0E 06 01 1F 20 0C 00 00
  
```

2.3 Loopback signaling test of Bluetooth tester

Loopback signaling test is also the most common test method for BQB and OTA antenna testing. All test instructions are the same as those described in the previous sections, except that HCI commands are sent by

the instrument. Please ensure that the COM port configuration on the tester matches that of the device under test.

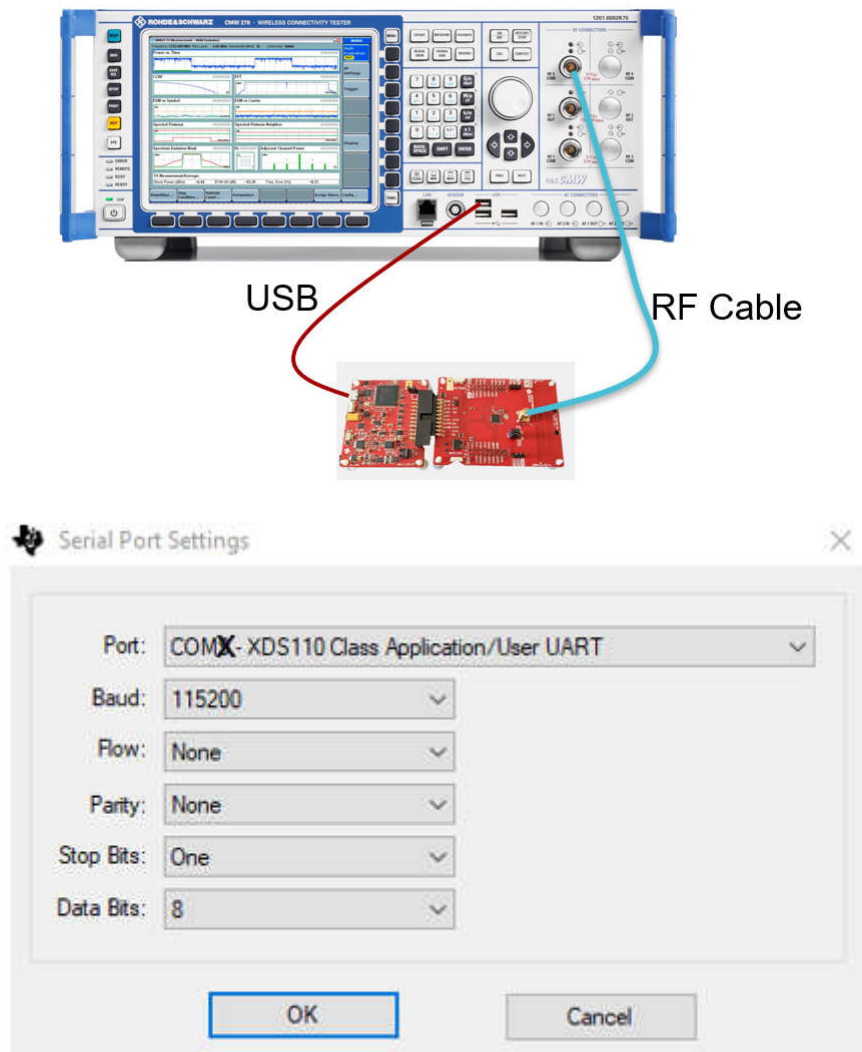


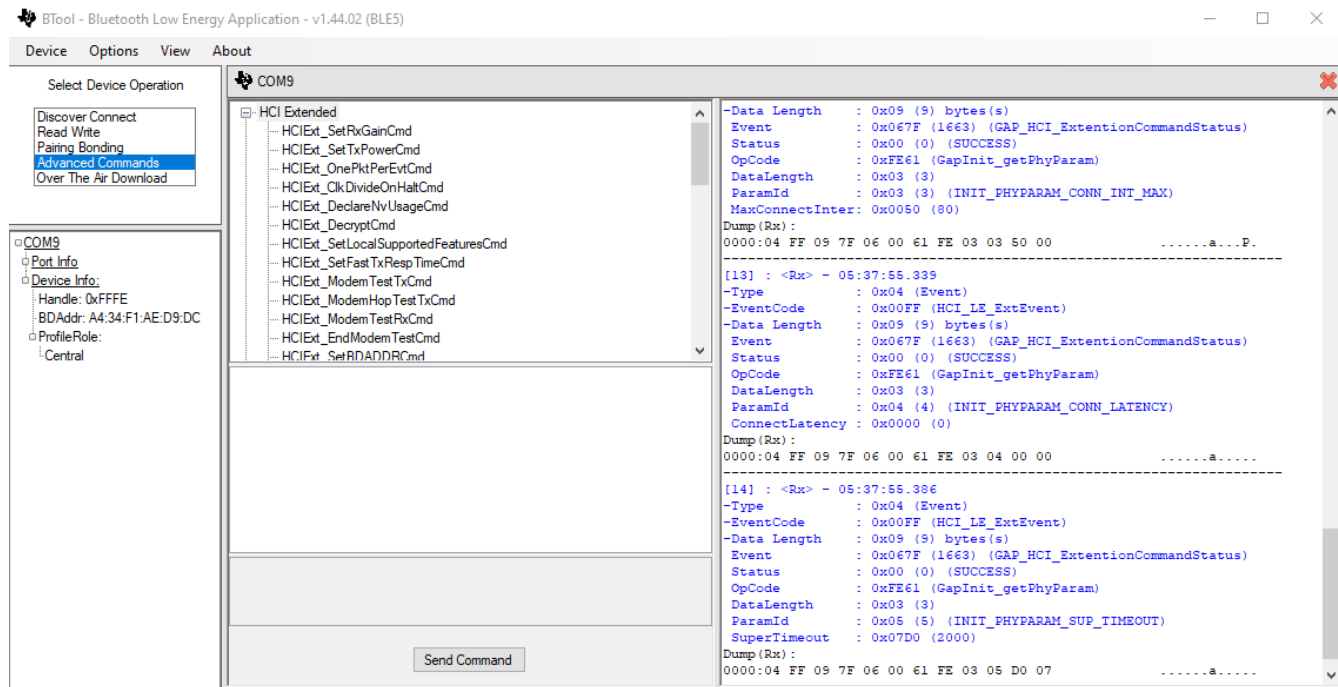
Figure 2-3. Signaling connection diagram

Note: Figure 2-3 is only an example. In practice, CMW requires XDS110 driver to be installed, which is generally difficult to implement. Customers need to use a USB-UART adapter supported by the verification instrument side.

3 SRRC fixed frequency test

This article focuses on the configuration of test items for State Radio Regulatory Commission, and it is also applicable to other certifications such as CE and FCC. Readers may configure it using the methods described below based on the specific testing requirements of the test lab. Typical demands include transmitting and receiving fixed-frequency modulated signals and non-modulated signals in the high, medium, and low frequency bands.

For this test, the command set under extended HCI command is used. These are manufacturer-defined commands when OGF is 0x3F.



3.1 New SRRC rules and applicable terms

A requirement for a yielding mechanism for 2.4GHz Bluetooth device is added to the latest specification [issued by State Radio Regulatory Commission \(SRRC\), the Ministry of Industry and Information Technology, titled Technical Requirements for Interference Avoidance in Radio Transmitter Equipment in 2400MHz, 5100MHz and 5800MHz Bands](#). This requires Bluetooth devices to support Adaptive Frequency Hopping (AFH). AFH is not yet supported in versions prior to SDK8.30. In that case, users may choose equipment that complies with the requirements for Class 4: interference mitigation technologies based on other mechanisms. This item is required to meet an “equivalent utilization” of no greater than 10%.

$$EU = \left(\frac{P_{e.i.r.p}(mW)}{P_{limit}(mW)} \right) \times DC$$

In the formula, EU is the equivalent utilization and DC is the device duty cycle, where P_{limit} is the equivalent isotropic radiated power limit at 2400MHz or 5800MHz in the corresponding Annex 1.

P_{limit} is 100mW for antenna direction gain less than 10dBi.

For devices with an e.i.r.p. of 10dBm (10mW) or less (8dBm for CC2340 series and 5dBm for CC2642 series), assuming that a maximum antenna direction gain is 0dBi, EU will be less than 10% even if the duty cycle is 100%.

In the latest SDK versions (such as SDK9.x and later), devices that claim to be based on the “detection and avoidance” mechanism no longer need to consider the duty cycle.

3.2 Test with Btool

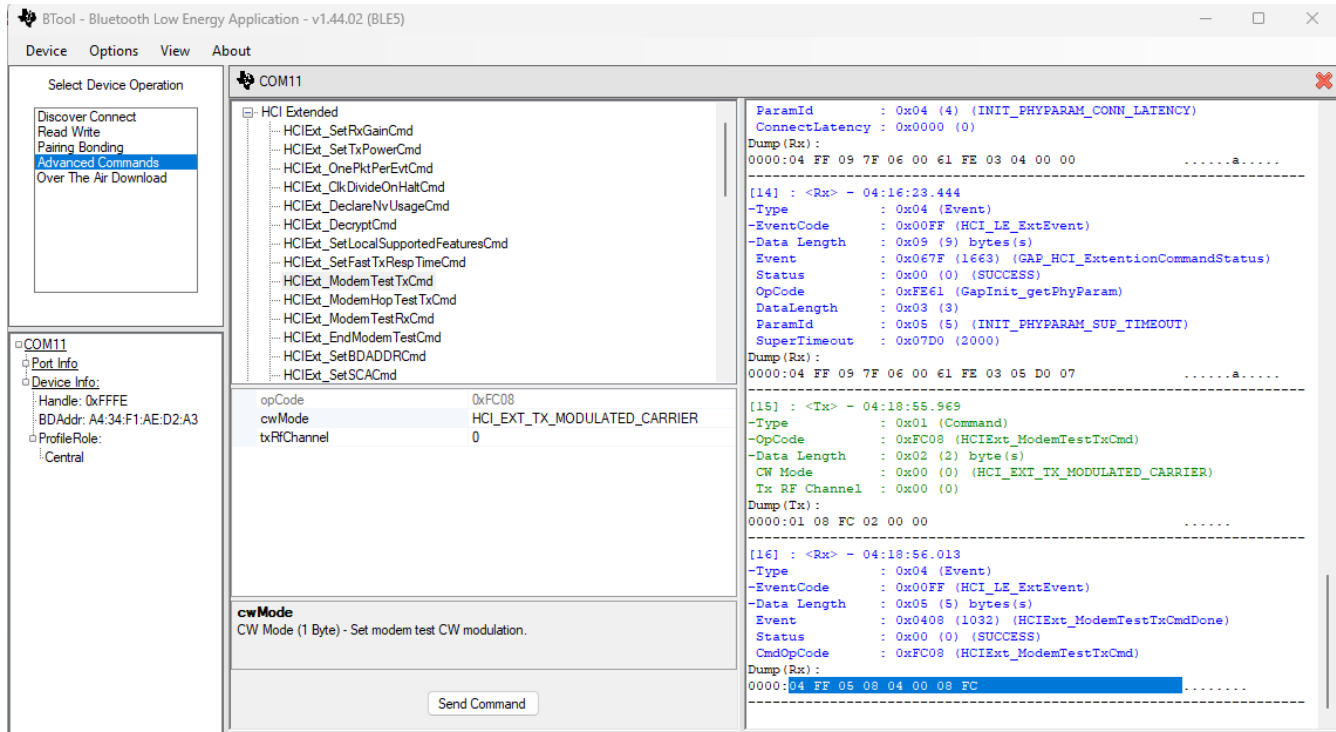
3.2.1 Transmit HCIExt_ModemTestTxCmd

HCIExt_ModemTestTxCmd is used in transmission testing where

cwMode: HCI_EXT_TX_MODULATED_CARRIER is the modulated fixed-frequency signal.

HCI_EXT_TX_UNMODULATED_CARRIER is the non-modulated fixed frequency signal, which can also be used to test the frequency deviation of high-frequency crystal oscillators.

txRfChannel: this parameter configures the channel of the test signal.



The screenshot shows the BTool interface with the following details:

- Device:** COM11
- Select Device Operation:** Advanced Commands
- Device Info:**
 - Handle: 0xFFFFE
 - BDAddr: A4:34:F1:AE:D2:A3
 - ProfileRole: Central
- HCI Extended Command List:**
 - opCode: 0xFC08
 - cwMode: HCI_EXT_TX_MODULATED_CARRIER
 - txRfChannel: 0
- Command Description:**

cwMode
CW Mode (1 Byte) - Set modem test CW modulation.
- Send Command** button is visible.
- Log Output:**

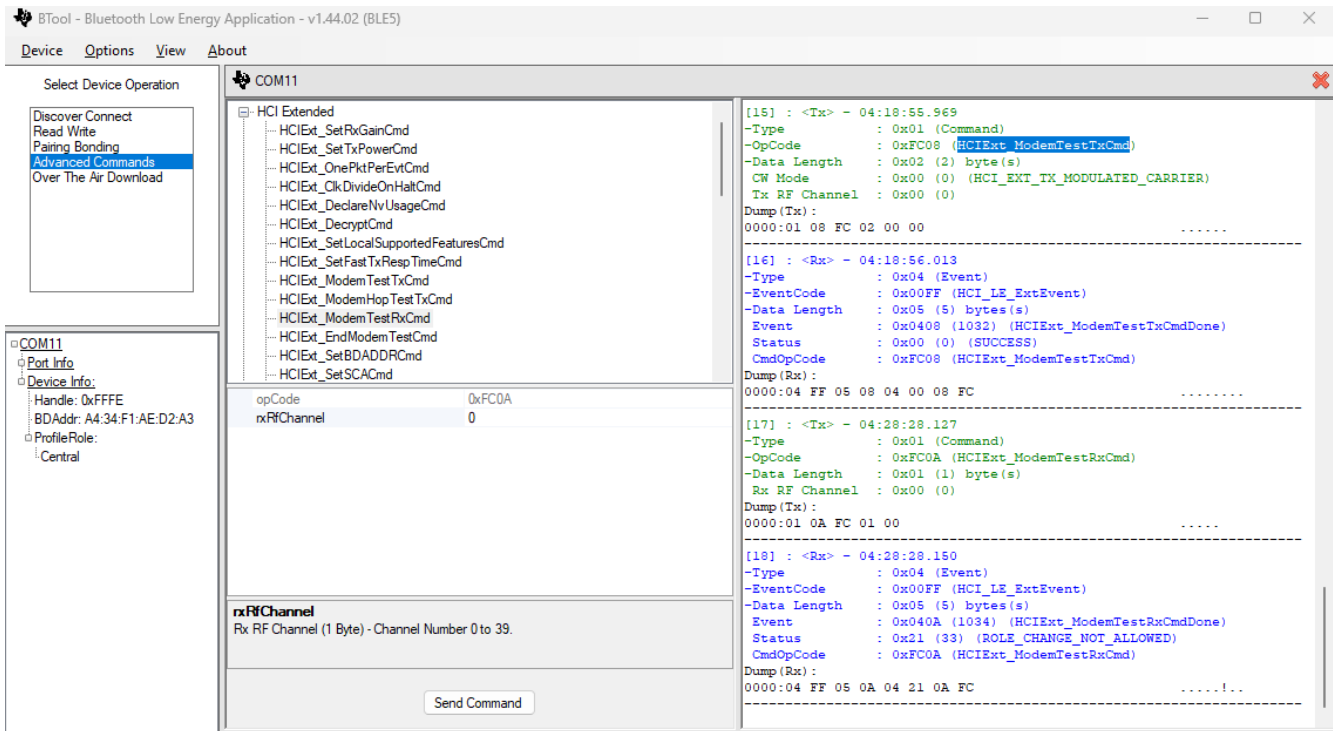
```

ParamId      : 0x04 (4) (INIT_PHYPARAM_CONN_LATENCY)
ConnectLatency : 0x0000 (0)
Dump (Rx):
0000:04 FF 09 7F 06 00 61 FE 03 04 00 00 .....a.....
-----
[14] : <Rx> - 04:16:23.444
-Type       : 0x04 (Event)
-EventCode  : 0x00FF (HCI_LE_ExtEvent)
-Data Length : 0x09 (9) bytes(s)
Event       : 0x067F (1663) (GAP_HCI_ExtentionCommandStatus)
Status      : 0x00 (0) (SUCCESS)
OpCode      : 0xFE61 (GapInit_getPhyParam)
DataLength  : 0x03 (3)
ParamId     : 0x05 (5) (INIT_PHYPARAM_SUP_TIMEOUT)
SuperTimeout : 0x07D0 (2000)
Dump (Rx):
0000:04 FF 09 7F 06 00 61 FE 03 05 D0 07 .....a.....
-----
[15] : <Tx> - 04:18:55.969
-Type       : 0x01 (Command)
-OpCode     : 0xFC08 (HCIExt_ModemTestTxCmd)
-Data Length : 0x02 (2) byte(s)
CW Mode     : 0x00 (0) (HCI_EXT_TX_MODULATED_CARRIER)
Tx RF Channel : 0x00 (0)
Dump (Tx):
0000:01 08 FC 02 00 00 .....
-----
[16] : <Rx> - 04:18:56.013
-Type       : 0x04 (Event)
-EventCode  : 0x00FF (HCI_LE_ExtEvent)
-Data Length : 0x05 (5) bytes(s)
Event       : 0x0408 (1032) (HCIExt_ModemTestTxCmdDone)
Status      : 0x00 (0) (SUCCESS)
CmdOpCode   : 0xFC08 (HCIExt_ModemTestTxCmd)
Dump (Rx):
0000:04 FF 05 08 04 00 08 FC .....

```

3.2.2 Receive HCIExt_ModemTestRxCmd

HCIExt_ModemTestRxCmd can configure the chip to enter reception mode. txRfChannel parameter configures the test signals.

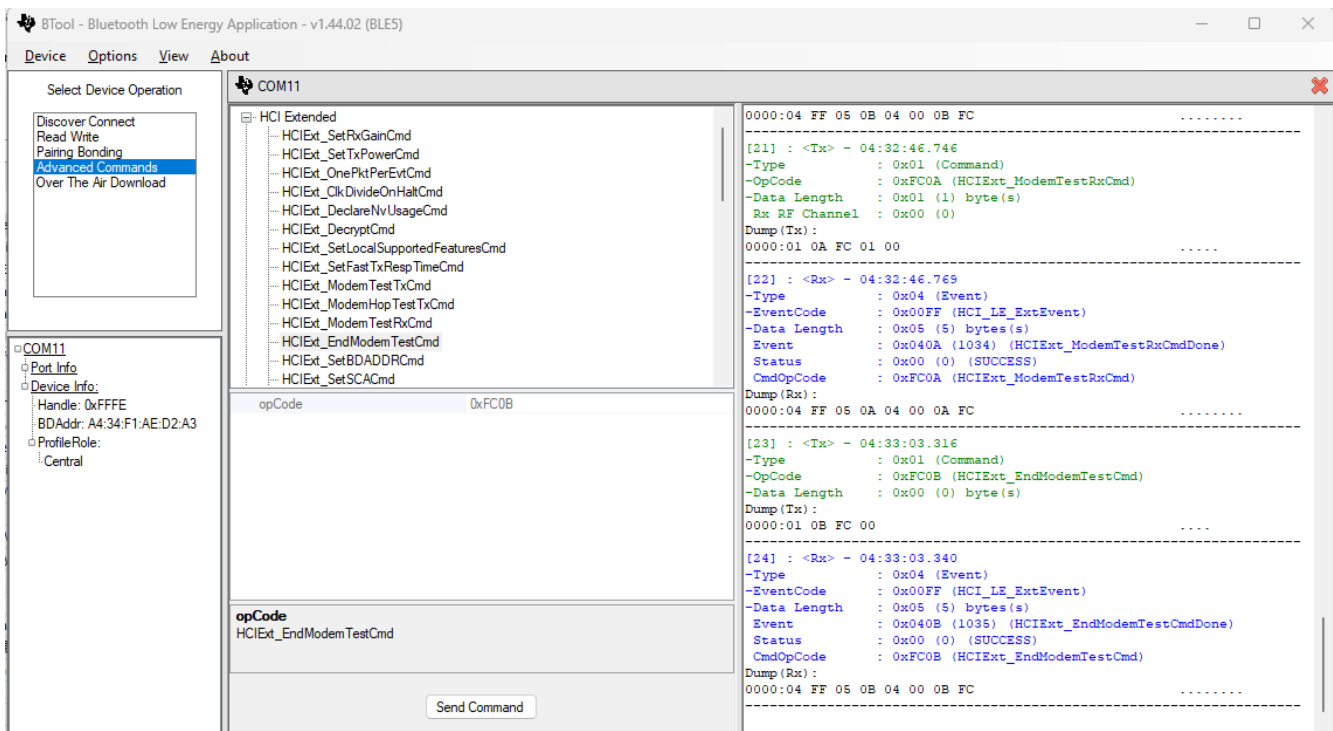


The screenshot shows the BTTool interface with the following details:

- Device:** COM11
- Select Device Operation:** Advanced Commands
- COM11 Device Info:**
 - Handle: 0xFFFF
 - BDAddr: A4:34:F1:AE:D2:A3
 - Profile Role: Central
- HCI Extended Commands:**
 - opCode: 0xFC0A
 - rxRFChannel: 0
- rxRFChannel:** Rx RF Channel (1 Byte) - Channel Number 0 to 39.
- Log:**
 - [15] : <Tx> - 04:18:55.969
 - Type : 0x01 (Command)
 - OpCode : 0xFC08 (HCIExt_ModemTestTxCmd)
 - Data Length : 0x02 (2) byte(s)
 - Cmd OpCode : 0x00 (0) (HCI_EXT_TX_MODULATED_CARRIER)
 - Tx RF Channel : 0x00 (0)
 - Dump (Tx): 0000:01 08 FC 02 00 00
 - [16] : <Rx> - 04:18:56.013
 - Type : 0x04 (Event)
 - EventCode : 0x00FF (HCI_LE_ExtEvent)
 - Data Length : 0x05 (5) bytes(s)
 - Event : 0x0408 (1032) (HCIExt_ModemTestTxCmdDone)
 - Status : 0x00 (0) (SUCCESS)
 - CmdOpCode : 0xFC08 (HCIExt_ModemTestTxCmd)
 - Dump (Rx): 0000:04 FF 05 08 04 00 08 FC
 - [17] : <Tx> - 04:28:28.127
 - Type : 0x01 (Command)
 - OpCode : 0xFC0A (HCIExt_ModemTestRxCmd)
 - Data Length : 0x01 (1) byte(s)
 - Rx RF Channel : 0x00 (0)
 - Dump (Tx): 0000:01 0A FC 01 00
 - [18] : <Rx> - 04:28:28.150
 - Type : 0x04 (Event)
 - EventCode : 0x00FF (HCI_LE_ExtEvent)
 - Data Length : 0x05 (5) bytes(s)
 - Event : 0x040A (1034) (HCIExt_ModemTestRxCmdDone)
 - Status : 0x21 (33) (ROLE_CHANGE_NOT_ALLOWED)
 - CmdOpCode : 0xFC0A (HCIExt_ModemTestRxCmd)
 - Dump (Rx): 0000:04 FF 05 0A 04 21 0A FC

3.2.3 Test End HCIExt_EndModemTestCmd

The test end command is [HCIExt_EndModemTestCmd](#). It is used when switching transmission and reception command parameters.



The screenshot shows the BTTool interface with the following details:

- Device:** COM11
- Select Device Operation:** Advanced Commands
- COM11 Device Info:**
 - Handle: 0xFFFF
 - BDAddr: A4:34:F1:AE:D2:A3
 - Profile Role: Central
- HCI Extended Commands:**
 - opCode: 0xFC0B
 - opCode: HCIExt_EndModemTestCmd
- Log:**
 - [21] : <Tx> - 04:32:46.746
 - Type : 0x01 (Command)
 - OpCode : 0xFC0A (HCIExt_ModemTestRxCmd)
 - Data Length : 0x01 (1) byte(s)
 - Rx RF Channel : 0x00 (0)
 - Dump (Tx): 0000:01 0A FC 01 00
 - [22] : <Rx> - 04:32:46.769
 - Type : 0x04 (Event)
 - EventCode : 0x00FF (HCI_LE_ExtEvent)
 - Data Length : 0x05 (5) bytes(s)
 - Event : 0x040A (1034) (HCIExt_ModemTestRxCmdDone)
 - Status : 0x00 (0) (SUCCESS)
 - CmdOpCode : 0xFC0A (HCIExt_ModemTestRxCmd)
 - Dump (Rx): 0000:04 FF 05 0A 04 00 0A FC
 - [23] : <Tx> - 04:33:03.316
 - Type : 0x01 (Command)
 - OpCode : 0xFC0B (HCIExt_EndModemTestCmd)
 - Data Length : 0x00 (0) byte(s)
 - Dump (Tx): 0000:01 0B FC 00
 - [24] : <Rx> - 04:33:03.340
 - Type : 0x04 (Event)
 - EventCode : 0x00FF (HCI_LE_ExtEvent)
 - Data Length : 0x05 (5) bytes(s)
 - Event : 0x040B (1035) (HCIExt_EndModemTestCmdDone)
 - Status : 0x00 (0) (SUCCESS)
 - CmdOpCode : 0xFC0B (HCIExt_EndModemTestCmd)
 - Dump (Rx): 0000:04 FF 05 0B 04 00 0B FC

3.3 Implement the test with SmartRF Studio

In addition to using DTM/PTM mode of the code, TI provides SmartRF Studio tool for debugging the lab's transmission and reception. This suite of tools can also be used for RF testing in radio management. SmartRF

Studio can control the physical layer of the chip directly without the need to download software. However, this tool needs an emulator, so it must be connected to a computer during testing. Connecting the USB cable may introduce interference from stray signals. TI recommends to use DTM/PTM mode for radio certification.

Refer to <https://www.ti.com/tool/SMARTRFTM-STUDIO> for SmartRF studio and user manual

4 Summary

This article provides a concise description of the test process and methodology, so that the reader can minimize the learning of RF and software to achieve the goal of completing certification testing. For more information, refer to the relevant documentation in *5 Reference Documentation*.

5 Reference documentation

1. <https://www.ti.com/tool/SMARTRFTM-STUDIO>
2. [Configuring Bluetooth LE devices for Direct Test Mode](#)
3. [How to Do RF Radio Test With Your Bluetooth Product \(Rev. A\)](#)
4. [How to Certify Your Bluetooth Product \(Rev. K\)](#)
5. <https://www.ti.com/tool/LP-XDS110ET>
6. [Technical Requirements for Interference Avoidance in Radio Transmitter Equipment in 2400MHz, 5100MHz and 5800MHz Bands](#)

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#), [TI's General Quality Guidelines](#), or other applicable terms available either on [ti.com](#) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025