



Jerry Kuo, Nathan Block and Clement Chaduc

ABSTRACT

This application report provides the methods about doing RF radio testing on end-equipment devices that use TI SimpleLink™ CC26xx/CC13xx wireless MCU's. Bluetooth® RF radio testing can be done with two types of PC tools: SmartRF™ Studio and BTool. These tools provide similar functions. But they need different methods. Customers can refer to this document and plan for their suitable testing processes.

Table of Contents

1 Introduction	2
1.1 Acronyms	2
2 Verify RF Performance With SmartRF Studio	2
3 Verify RF Performance With BTool	3
3.1 BTool Basic Operation	4
3.2 Perform Tx Test	5
3.3 Perform Rx Test	8
3.4 Perform Reading RSSI	8
3.5 Perform PER Test	9
3.6 Notes for Different PHY Rate	12
3.7 The Difference Between HCI_EXT_ModemTestTxCmd and HCI_LE_Transmitter_Test	18
4 Summary	18
5 References	19
6 Revision History	19

List of Figures

Figure 2-1. Use SmartRF Studio to Verify RF Performance on Tx and Rx Role	3
Figure 3-1. Set HCI_readBdAddr	4
Figure 3-2. Set HCIExt_SetTxPowerCmd	5
Figure 3-3. Set HCIExt_ModemTestTxCmd	6
Figure 3-4. Set HCIExt_EnhancedModemTestTxCmd	7
Figure 3-5. Set HCIExt_ModemTestRxCmd	8
Figure 3-6. Set HCI_ReadRSSI	8
Figure 3-7. Set HCI_LE_ReceiverTest	9
Figure 3-8. Set MISC_GenericCommand	10
Figure 3-9. Set HCI_LE_Transmitter_Test	11
Figure 3-10. Set HCI_LE_TestEnd	12
Figure 3-11. Set HCI_LE_EnhancedReceiverTest	12
Figure 3-12. Set HCI_LE_EnhancedTransmitterTest	13
Figure 3-13. Response After Setting HCI_LE_EnhancedTransmitterTest	13
Figure 4-1. Test Plan Flow Chart	19

List of Tables

Table 1-1. Comparison Between SmartRF Studio and BTool	2
Table 1-2. Acronyms Used in This Document	2
Table 3-1. List of Channel Index	6

Trademarks

SimpleLink™ and SmartRF™ are trademarks of Texas Instruments.
Bluetooth® is a registered trademark of Bluetooth SIG.

All trademarks are the property of their respective owners.

1 Introduction

TI SimpleLink MCU provides two PC tools to test RF performance of Bluetooth devices. These are SmartRF Studio [1] and BTool [2]. Each tool has different requirements and benefits laid out in [Table 1-1](#).

Table 1-1. Comparison Between SmartRF Studio and BTool

SmartRF Studio	BTool
Benefit- Can be used without flashing the board with firmware	Drawback - Requires Production Test Mode (PTM) firmware to be flashed onto the board
Drawback - User needs to connect cJTAG lines to the board to control the device	Benefit - User can control the device by sending BTool set commands over UART

This document mainly focuses on RF radio test. This document introduces the detailed processes for each tool to do RF performance test.

1.1 Acronyms

Table 1-2. Acronyms Used in This Document

Acronym	Description
BLE	Bluetooth Low Energy
cJTAG	Compact JTAG
DUT	Device under Test
FW	Firmware
MCU	Micro Controller Unit
PC	Personal Computer
PCB	Printed Circuit Board
PER	Packet Error Rate
RF	Radio Frequency
RX	Receive
TI LPRF	Texas Instruments Low Power RF
TX	Transmit

2 Verify RF Performance With SmartRF Studio

SmartRF Studio 7 (SmartRF) is a PC software tool for evaluating TI SimpleLink MCU's. It can generate device register values, test RF performance and tune customer specific hardware solutions. SmartRF does not require any code to be written to the SimpleLink Device before operation, making it an ideal tool for the early stages of product development. Users simply connect cJTAG debug lines to their custom PCB and SmartRF Studio can immediately start verifying RF performance.

First, connect the cJTAG lines to the DUT and launch SmartRF Studio. Then, connect the pre-certified device and launch another SmartRF Studio window. Once the two boards are controlled by the two SmartRF Studio application, you can choose DUT to act as Rx or Tx role. For more information, see [\[5\]](#) and [\[6\]](#).

Figure 2-1 shows two SmartRF Studio opened for Continuous Tx and Continuous Rx testing.

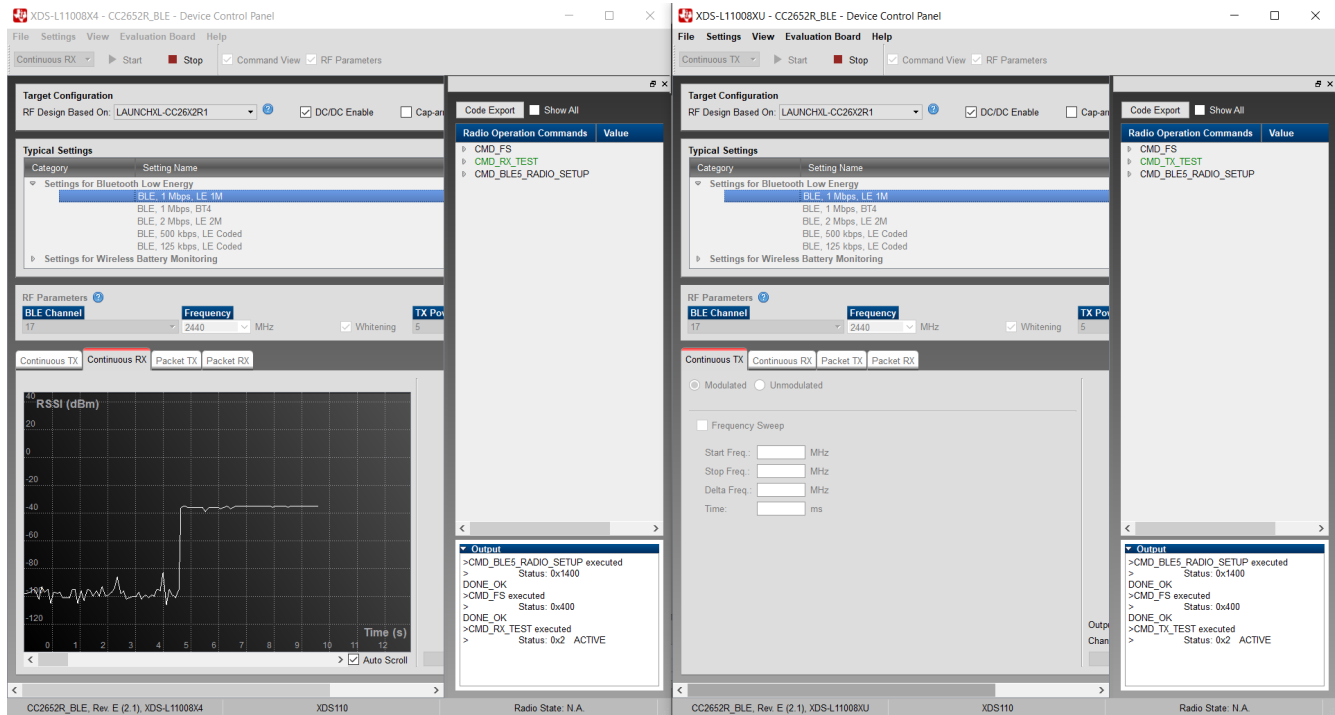


Figure 2-1. Use SmartRF Studio to Verify RF Performance on Tx and Rx Role

There are two types of tests: Continuous Tx/Rx and Packet Tx/Rx. Continuous Tx testing can let Tx side transmit RF signal with configured frequency and transmission power. Continuous Rx testing let Rx side measure the signal strength (RSSI) under configured frequency and transmission power. Packet Tx/Rx can be used to check the link quality between two devices. The user can select preferred settings given by SmartRF Studio 7 or use customized register values to do the packet error rate (PER) test.

3 Verify RF Performance With BTool

BTool is a PC Application that allows you to form a connection between two Bluetooth LE devices. BTool acts as a network processor to let you send HCI vendor specific commands to a CC26xx/CC13xx LaunchPad. Using BTool requires the CC26xx/CC13xx device to be flashed with either the host_test sample application or firmware enabling PTM function [3].

Using BTool, you can perform the same RF tests as SmartRF Studio by sending HCI commands to the CC26xx/CC13xx device. This section discusses how to operate BTool, how to perform Tx and Rx testing, and how to run RSSI and PER tests.

3.1 BTool Basic Operation

[2] introduces the details about BTool. [4] introduce how to use BTool for Bluetooth LE basic operation, which includes where to find the host_test example and how to flash it.

After CC26xx/CC13xx device is flashed with host_test FW, it can receive HCI commands through the UART interface. To verify if the environment is ready, send a basic command, like HCI_readBdAddr. If everything is fine, CC26xx/CC13xx replies its BD Address. Since the BD Address is unique, this command is also useful when you want to record the production history for each device on manufacturing line.

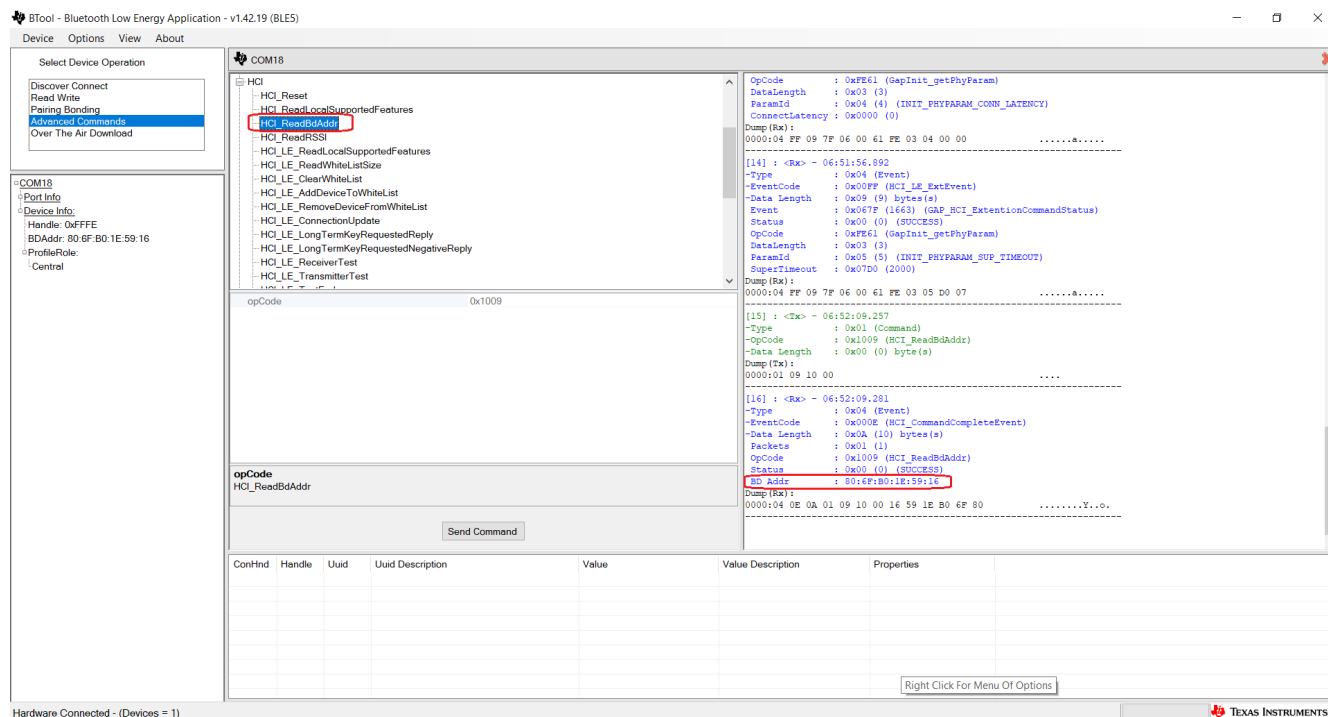


Figure 3-1. Set HCI_readBdAddr

3.2 Perform Tx Test

When the BTool and CC26xx/CC13xx devices are ready for test, start the RF Transmission Power test. [7] introduces this basic RF test. Equipment is needed to measure the power transmitted by the custom PCB, so you can use a Spectrum Analyzer or others dedicated for Bluetooth test, like Litepoint IQxel [8], Anitsu MT8852B [9] or R&S CMW270 [10].

The steps required for the Tx Test are the following:

1. Set Tx Power by setting HCIExt_SetTxPowerCmd

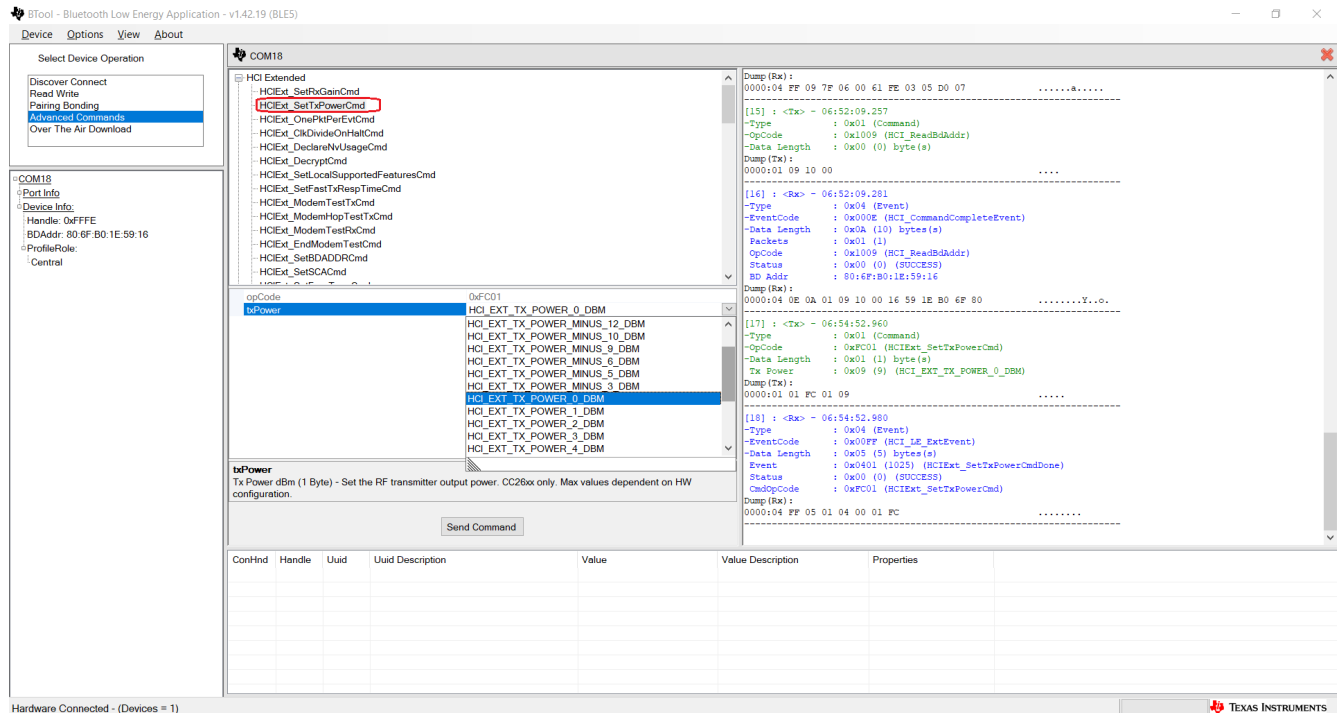


Figure 3-2. Set HCIExt_SetTxPowerCmd

2. Let the CC26xx/CC13xx device enter Tx mode by setting HCIExt_ModemTestTxCmd.

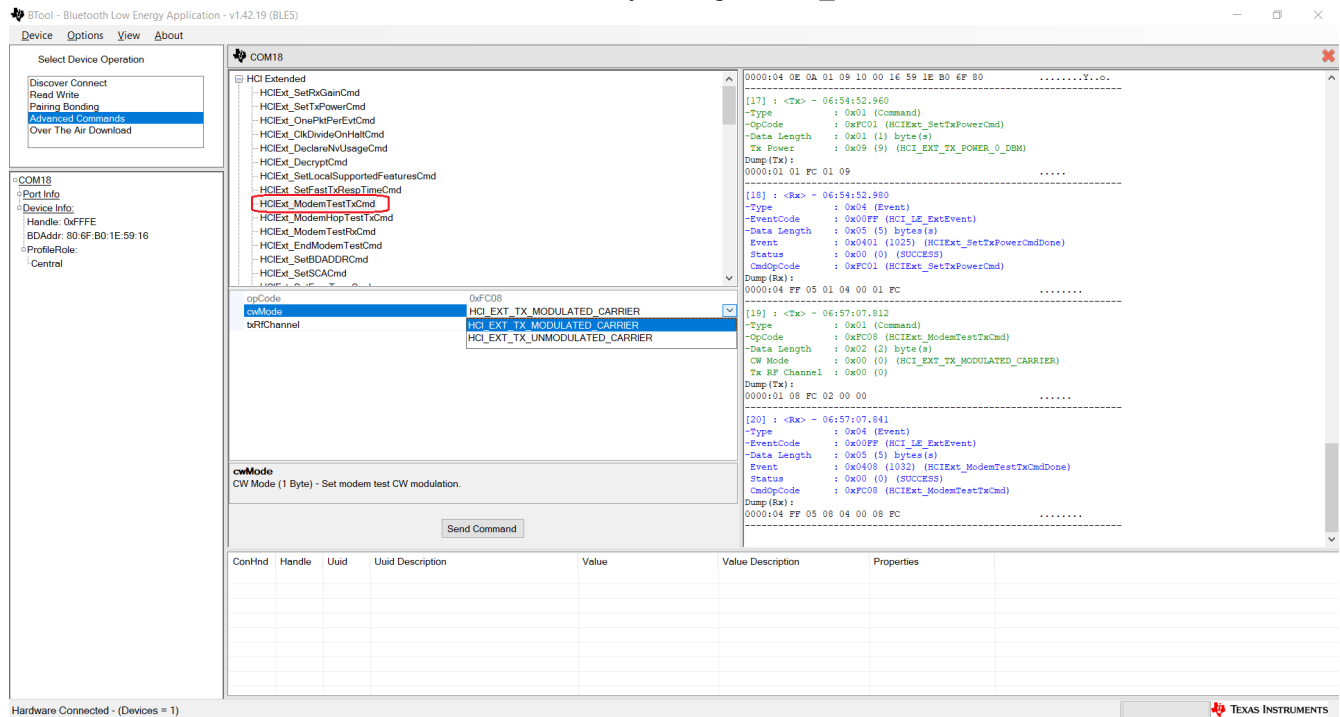


Figure 3-3. Set HCIExt_ModemTestTxCmd

Note the channels passed as a parameter to the HCIExt_ModemTestTxCmd do not follow the Bluetooth Low Energy allocation. Instead, in sequence from 0 (2402 MHz) to 39 (2480 MHz), the Bluetooth Low Energy allocation starts at 37 (2402 MHz), then 0 (2404 MHz) until 39 (2480 MHz).

Table 3-1 lists the channel index between both.

Table 3-1. List of Channel Index

Frequency (MHz)	Value of the Parameter txRfChannel Passed to HCIExt_ModemTestTxCmd	Corresponding Bluetooth Low Energy Channel
2402	0	37
2404	1	0
2406	2	1
...	...	...
2424	11	10
2426	12	38
2428	13	11
...	...	...
2476	37	35
2478	38	36
2480	39	39

3. To test on a different PHY, call a HCIExt_EnhancedModemTestTxCmd.

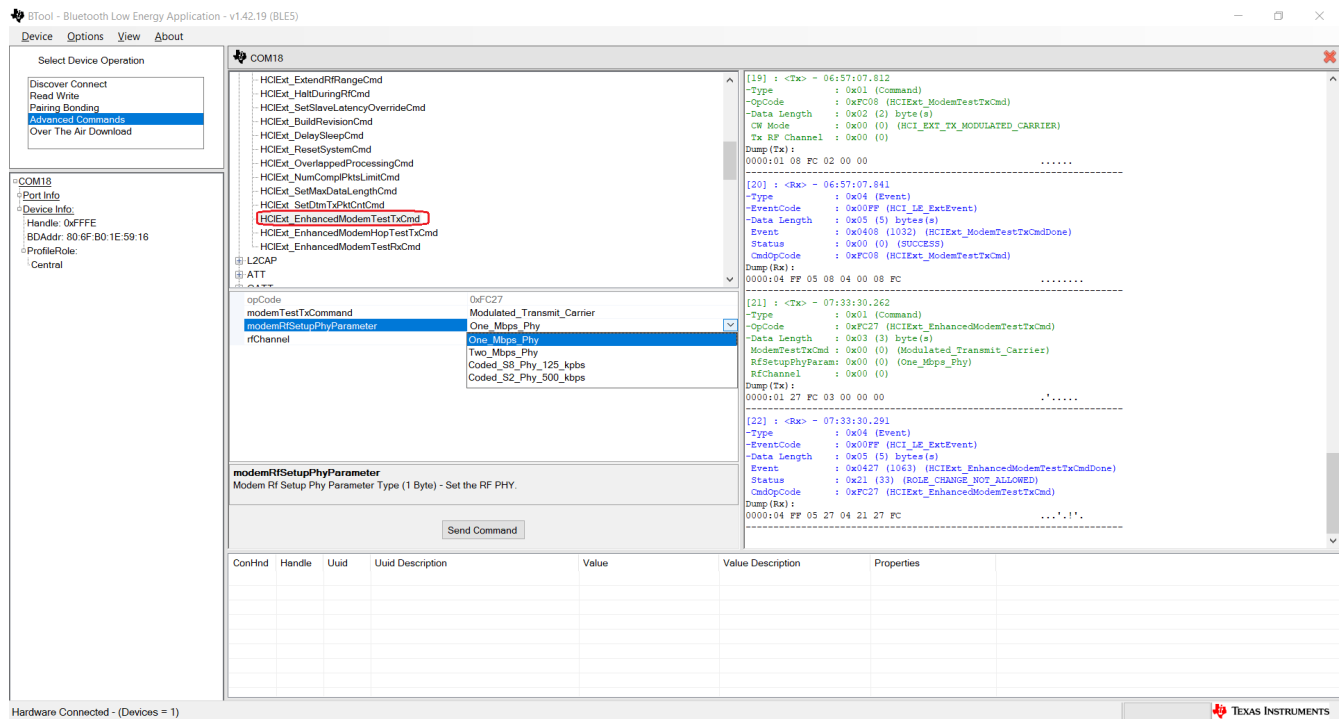


Figure 3-4. Set HCIExt_EnhancedModemTestTxCmd

By following the above steps, you can successfully run a Transmission Power test.

WARNING

Test parameters cannot be changed while a test is running. To perform such changes, you need to stop the current test by issuing the command HCIExt_EndModemTestCmd. When stopping a test via the HCIExt_EndModemTestCmd, it is necessary to re-set the parameters one by one. A common parameter used for compliance testing is the transmit power, which needs to be set using the command HCIExt_SetTxPowerCmd.

3.3 Perform Rx Test

The steps required for the Rx Test are the following:

1. Put the CC26xx/CC13xx device in Rx mode by setting HCIExt_ModemTestRxCmd.

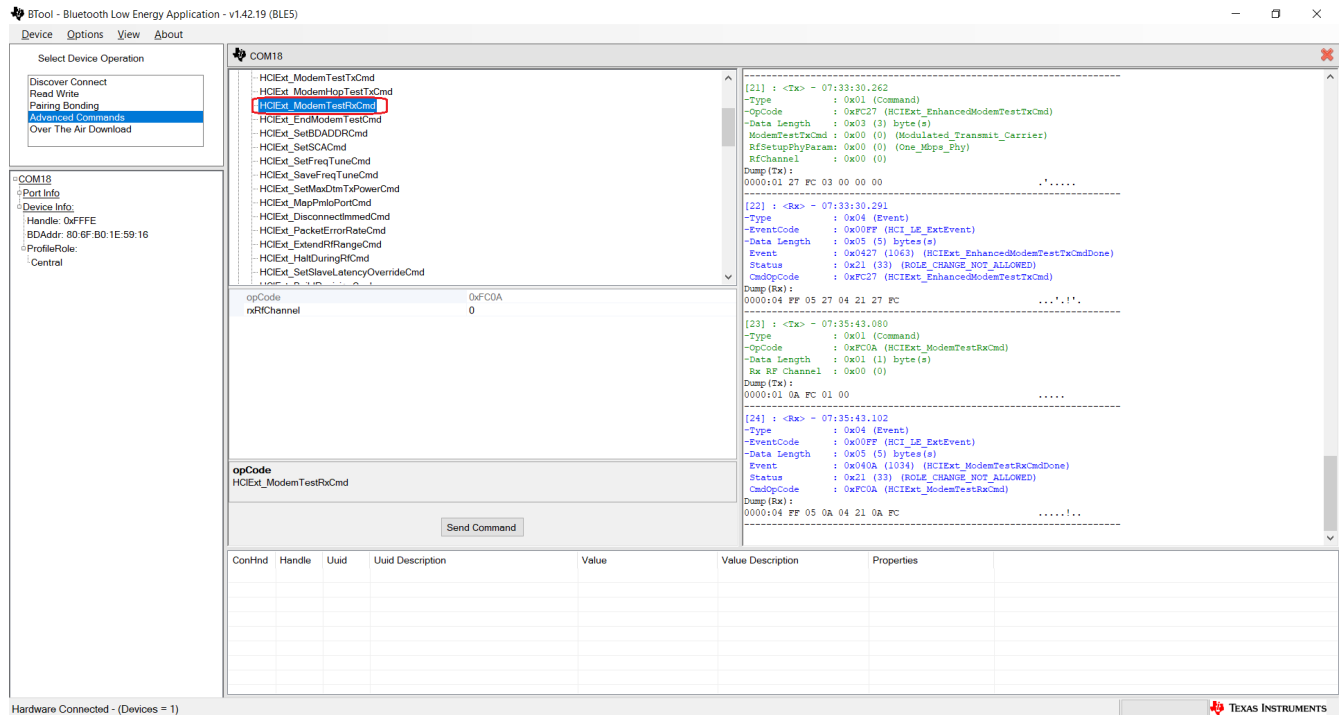


Figure 3-5. Set HCIExt_ModemTestRxCmd

2. After finishing test, send the HCI_LE_TestEnd command to get the test summary.

3.4 Perform Reading RSSI

Send the device the HCI_ReadRSSI command to get the current RSSI reading.

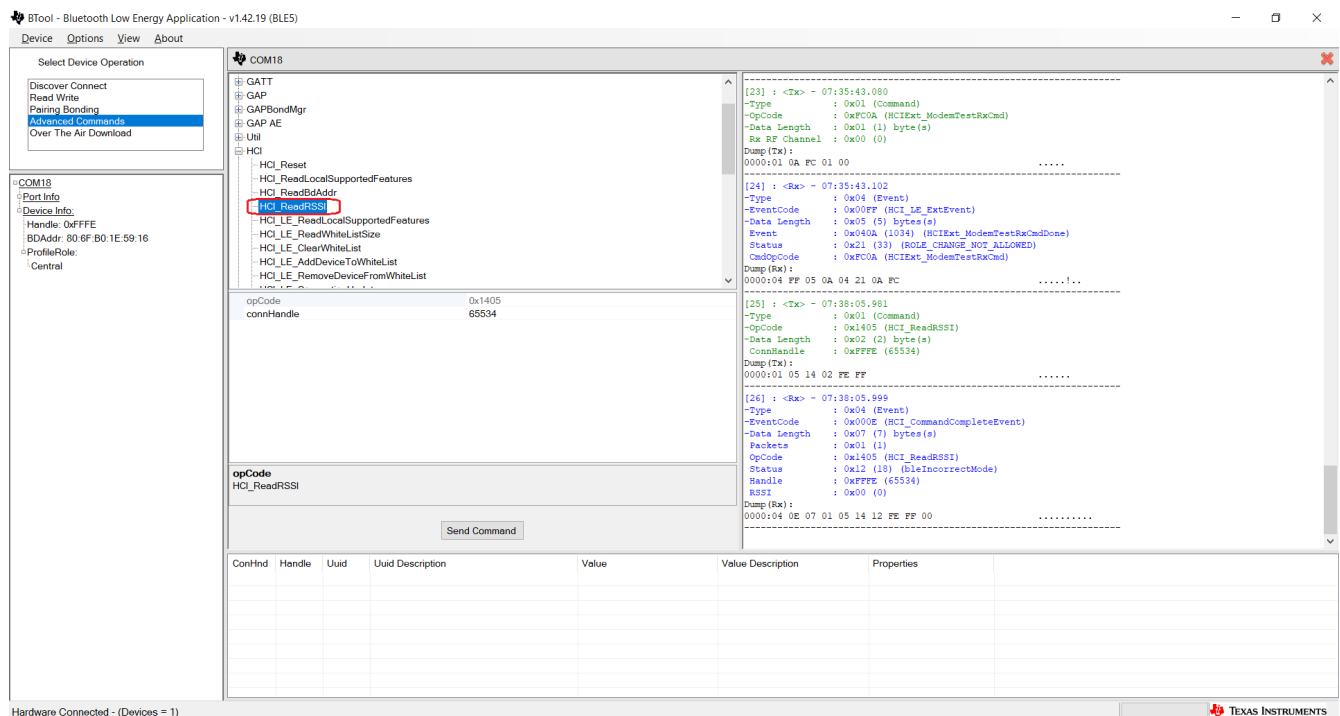


Figure 3-6. Set HCI_ReadRSSI

3.5 Perform PER Test

A quick and simple way to verify a device's RF performance on a manufacturing line is through a Packet Error Rate (PER) Test. The tester can use one pre-certified device to verify the DUT's performance. Two CC26xx/CC13xx devices must be flashed with host_test FW and connected to a PC running the BTool application for each device. One device acts as the Rx side, and the other acts as the Tx side. Below are the steps for the PER test.

1. On Rx side, let CC26xx/CC13xx device enter Rx mode by sending the device a HCI_LE_ReceiverTest command.

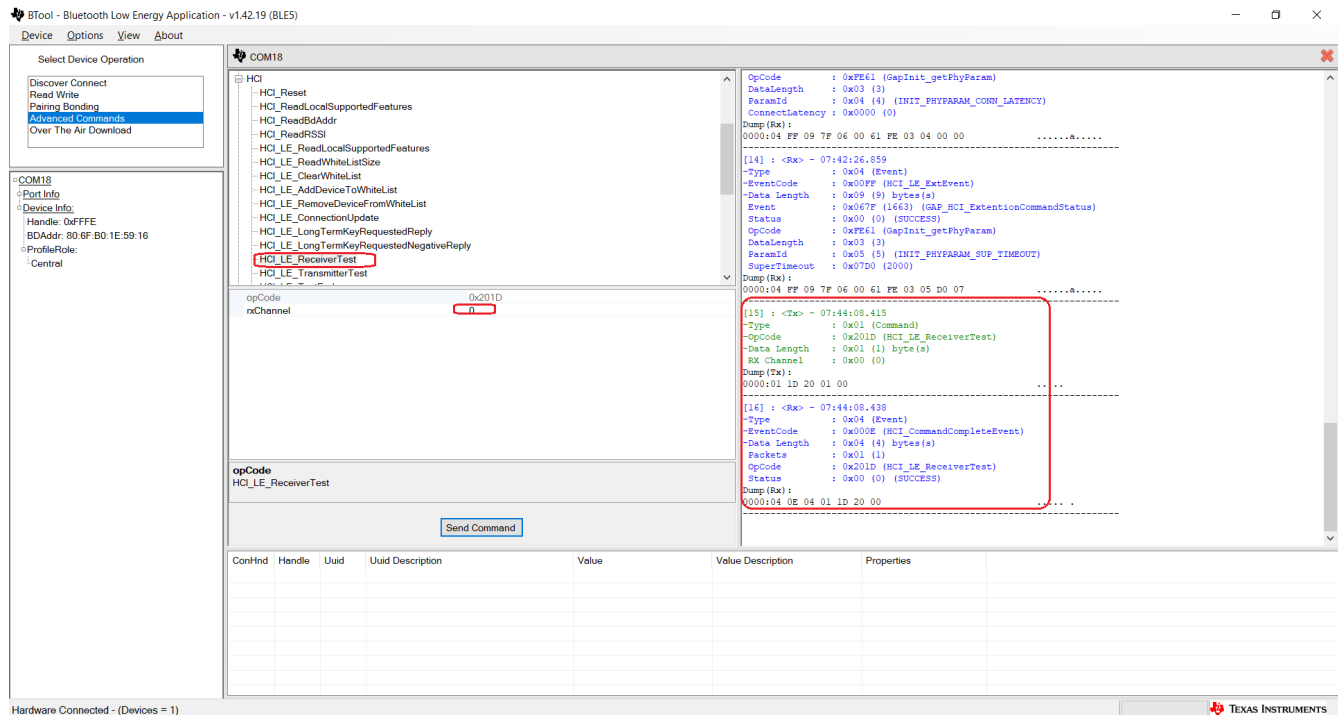


Figure 3-7. Set HCI_LE_ReceiverTest

“rxChannel” is the test channel.

2. On Tx side, configure the expected packets by sending the device a `MISC_GenericCommand`.

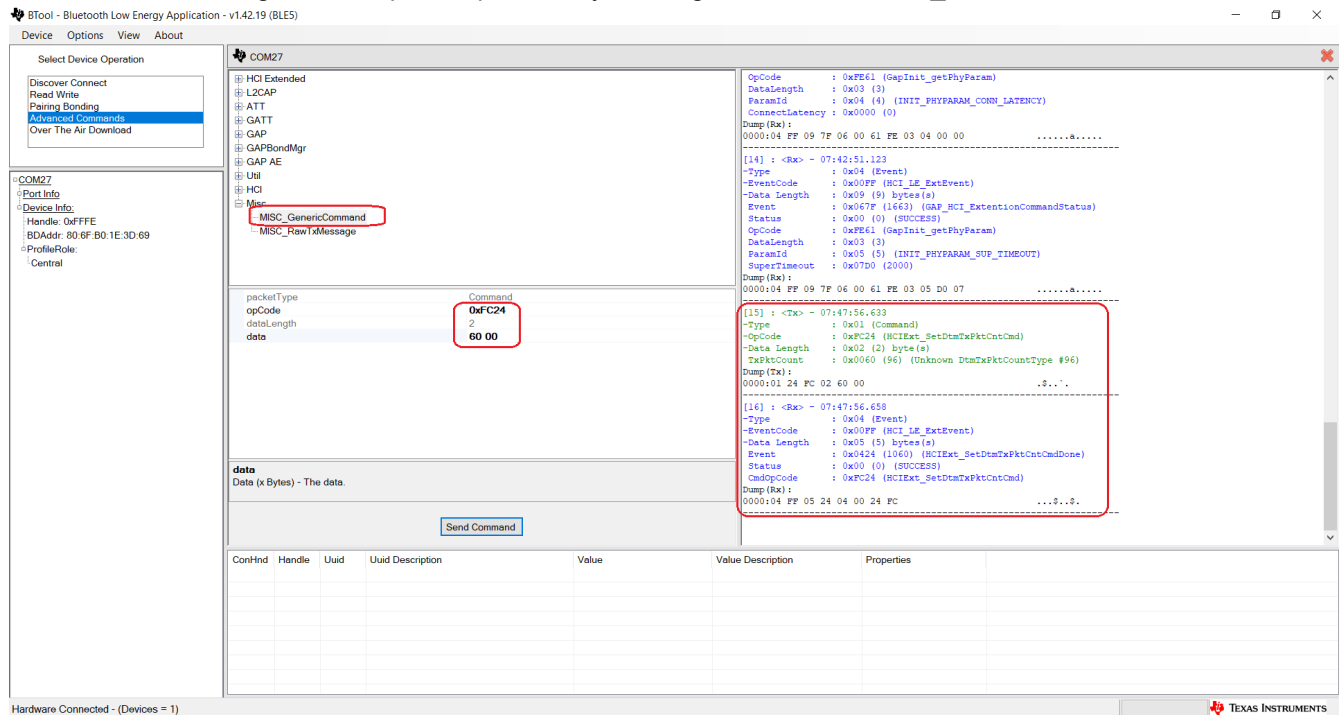


Figure 3-8. Set MISC_GenericCommand

opcode: 0xFC24

data: <the length to be transmitted in hex format>

above example “60 00” means 0x0060

data Length: no need to input

The screenshot displays the nRF Connect software interface for a Bluetooth Low Energy application. The top menu bar includes "Device", "Options", "View", and "About". The left sidebar shows the "Select Device Operation" menu with options like "Discover Connect", "Read Write", "Pairing Bonding", "Advanced Commands", and "Over The Air Download". The "Advanced Commands" menu is expanded, showing a list of commands under the "COM27" section. The "HCI LE_TransmitterTest" command is selected and highlighted with a red box. Below the command list, the "testData" section shows the test data (x Bytes) - The test data. The "Send Command" button is visible. The right pane displays the command details and the resulting data. The command details show the opcode as 0x201E, the tx channel as 37, and the test data as 04. The resulting data shows the command execution details, including the event code, data length, and status. The data is displayed in a hex dump format, with the first line showing the command opcode and the second line showing the test data. The data is also displayed in a table format with columns for "ConfId", "Handle", "Uuid", "Uuid Description", "Value", "Value Description", and "Properties".

txChannel: test channel, need to be the same as Rx side

testDataLength: length of payload bytes in each test packet, which can be any value from 0x00 to 0x25

testData: code for the type of data in the packet payload. The following values can be used

- 0x00 Pseudo-Random bit sequence 9
- 0x01 Pattern of alternating bits '11110000'
- 0x02 Pattern of alternating bits '10101010'
- 0x03 Pseudo-Random bit sequence 15
- 0x04 Pattern of All '1' bits
- 0x05 Pattern of All '0' bits
- 0x06 Pattern of alternating bits '00001111'
- 0x07 Pattern of alternating bits '0101'

4. On Rx side, after finishing test, by sending the “HCI_LE_TestEnd” to get test summary.

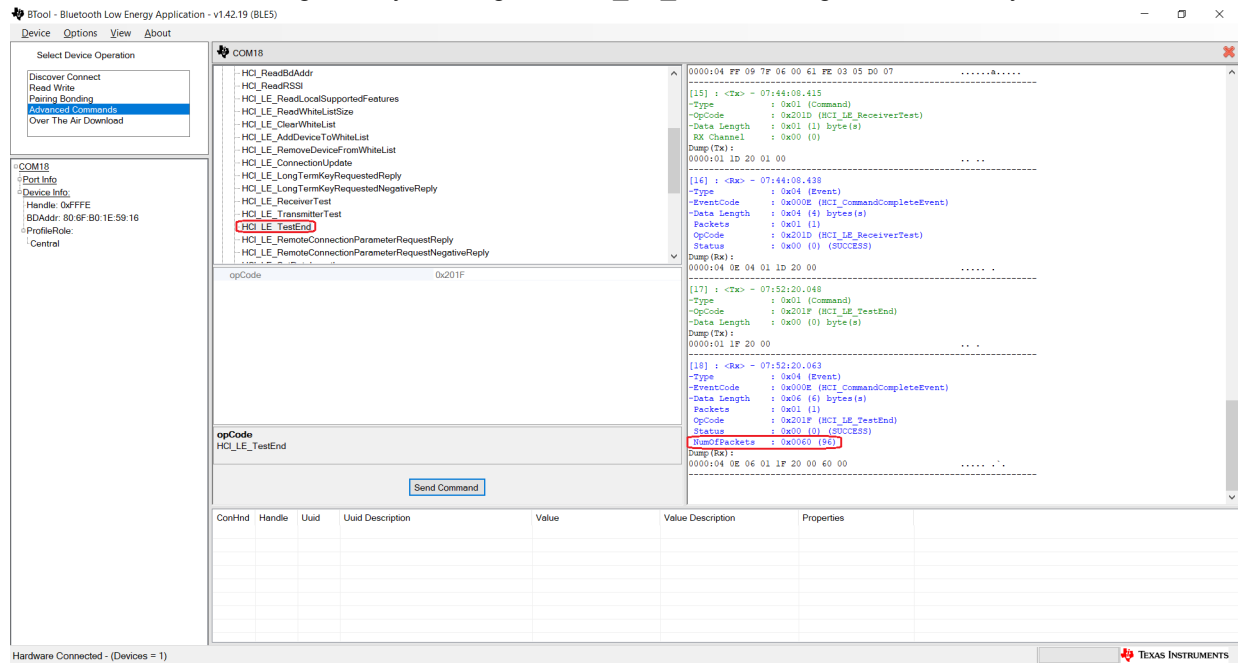


Figure 3-10. Set HCI_LE_TestEnd

NumberOfPackets: Total number of received packets

3.6 Notes for Different PHY Rate

To run the PER test on different PHY's, the sequence of commands are different from the above section in the following ways:

In step 1 of [Section 3.5](#) on Rx side, send the device a HCI_LE_EnhancedReceiverTest command instead of the HCI_LE_ReceiverTest command.

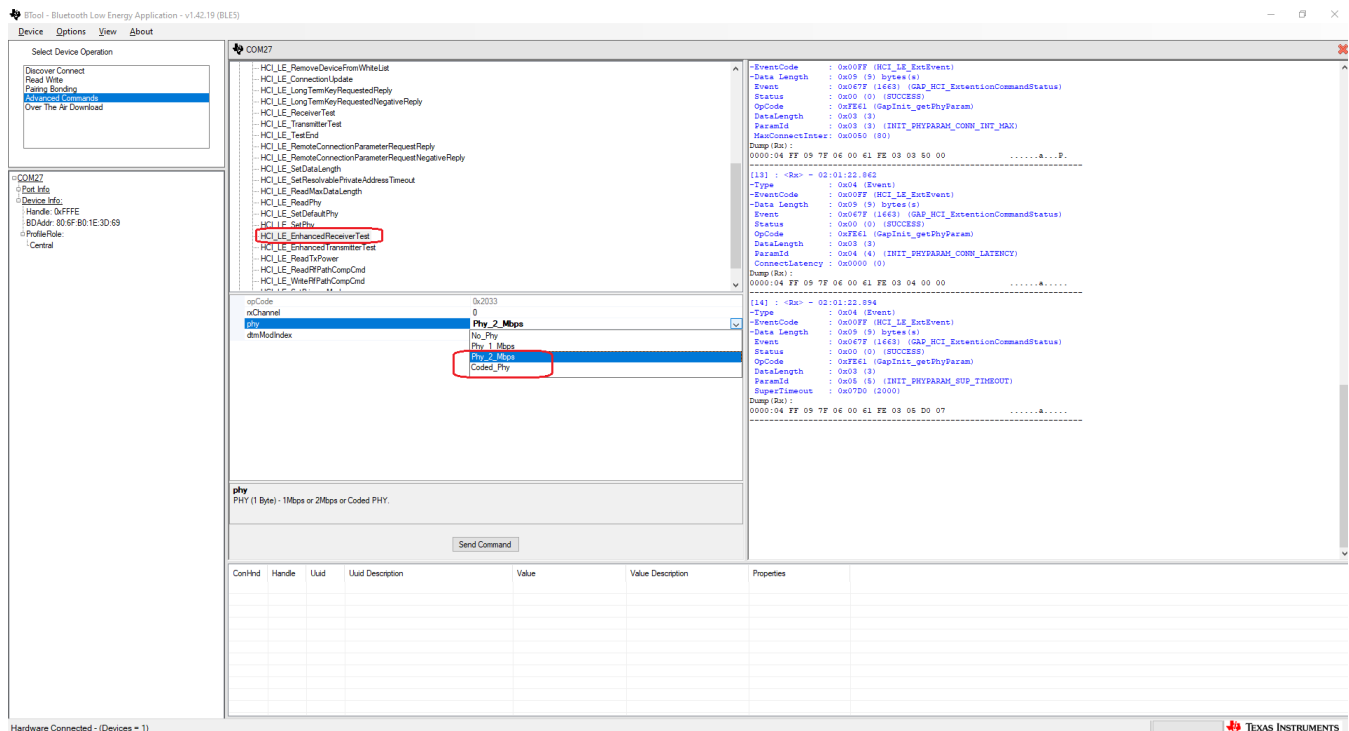


Figure 3-11. Set HCI_LE_EnhancedReceiverTest

In step 3 of [Section 3.5](#) on Rx side, send the device a HCI_LE_EnhancedTransmitterTest command instead of the HCI_LE_TransmitterTest command.

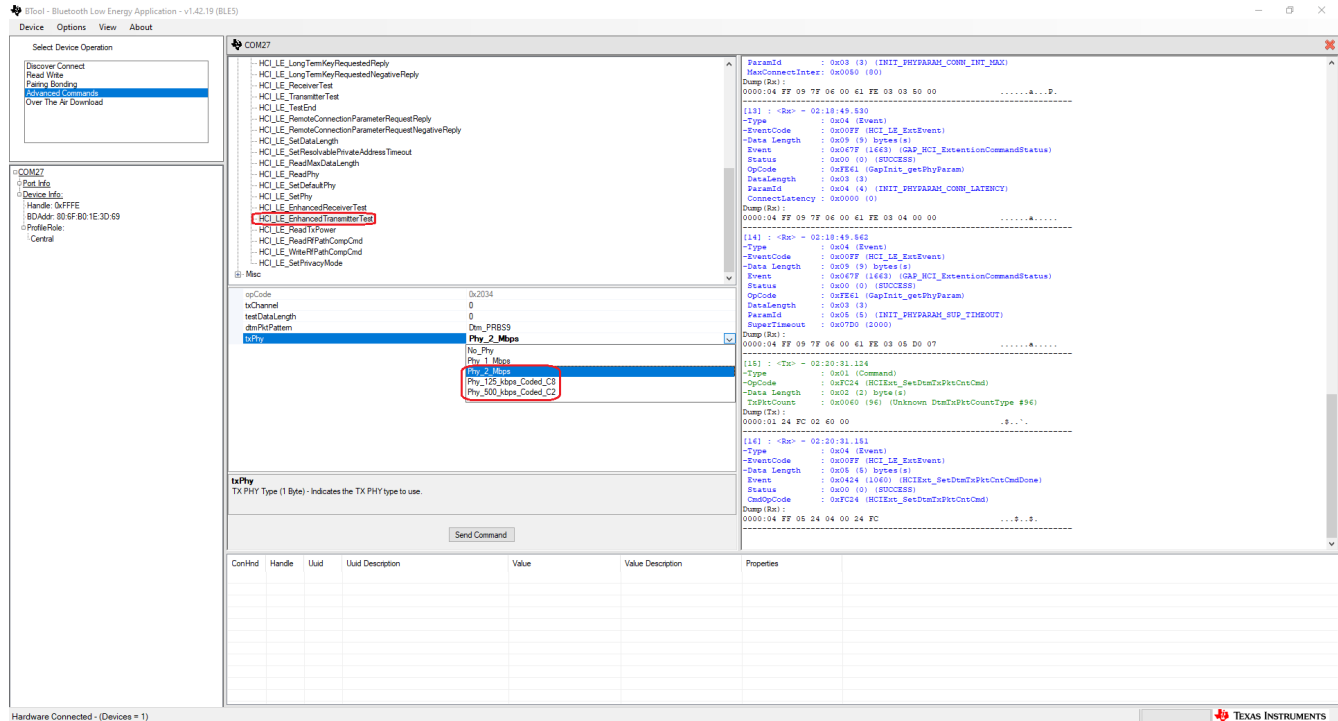


Figure 3-12. Set HCI_LE_EnhancedTransmitterTest

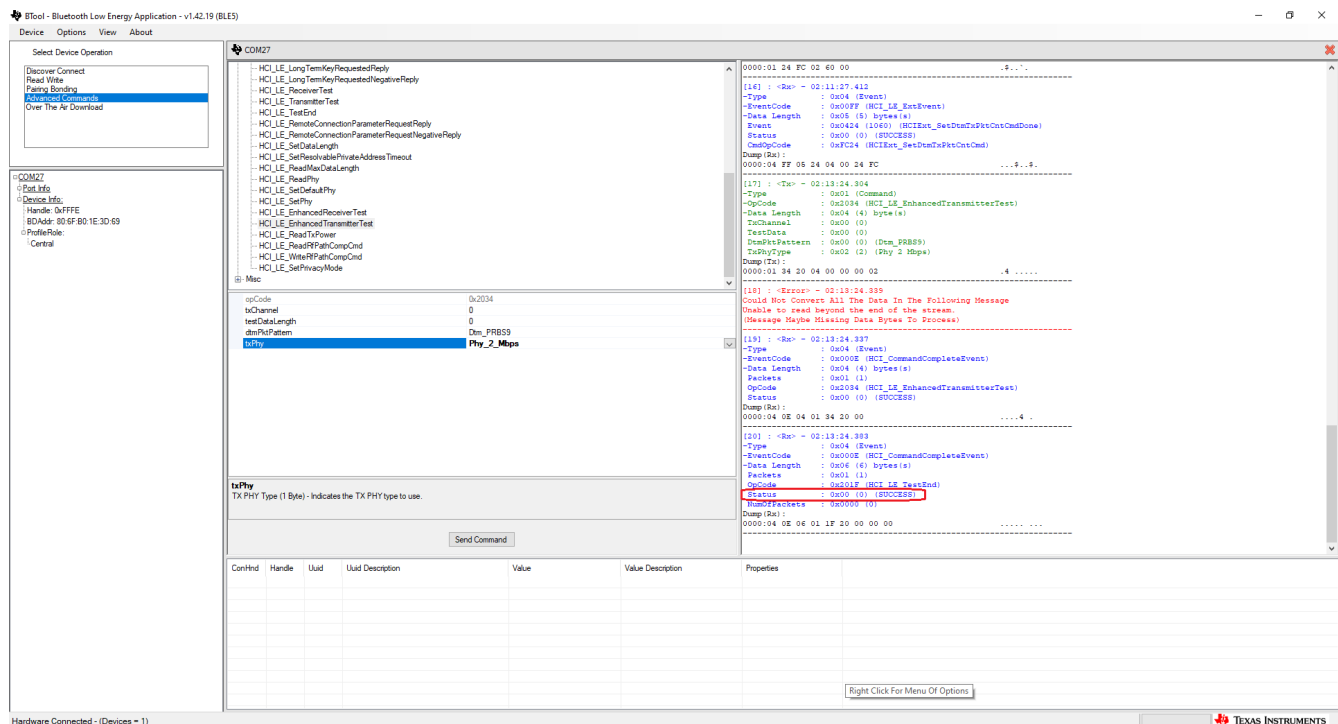


Figure 3-13. Response After Setting HCI_LE_EnhancedTransmitterTest

Below lists the test logs for different PHY rate.

3.6.1 Test logs for 2Mbps

Logs on Rx side:

```
[29] : <Tx> - 10:12:53.928
-Type           : 0x01 (Command)
-OpCode         : 0x2033 (HCI_LE_EnhancedReceiverTest)
-Data Length    : 0x03 (3) byte(s)
-RxChannel      : 0x00 (0)
Phy             : 0x02 (2) (Phy 2 Mbps)
DtmModeIndex    : 0x00 (0) (Standard Modulation Index)
Dump(Tx):
0000:01 33 20 03 00 02 00          .3 ....
-----
[30] : <Error> - 10:12:53.952
Could Not Convert All The Data In The Following Message
Unable to read beyond the end of the stream.
(Message Maybe Missing Data Bytes To Process)
-----
[31] : <Rx> - 10:12:53.952
-Type           : 0x04 (Event)
-EventCode      : 0x000E (HCI_CommandCompleteEvent)
-Data Length    : 0x04 (4) bytes(s)
Packets         : 0x01 (1)
OpCode         : 0x2033 (HCI_LE_EnhancedReceiverTest)
Status         : 0x00 (0) (SUCCESS)
Dump(Rx):
0000:04 0E 04 01 33 20 00          ....3 .
-----
[32] : <Tx> - 10:13:02.889
-Type           : 0x01 (Command)
-OpCode         : 0x201F (HCI_LE_TestEnd)
-Data Length    : 0x00 (0) byte(s)
Dump(Tx):
0000:01 1F 20 00                  .. .
-----
[33] : <Rx> - 10:13:02.912
-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    : 0x0064 (100)
Dump(Rx):
0000:04 0E 06 01 1F 20 00 64 00    .....d.
-----
```

Logs on Tx side:

```
[40] : <Tx> - 10:12:35.100
-Type           : 0x01 (Command)
-OpCode         : 0xFC24 (HCIExt_SetDtmTxPktCntCmd)
-Data Length    : 0x02 (2) byte(s)
TxPktCount      : 0x0064 (100) (Unknown DtmTxPktCountType #100)
Dump(Tx):
0000:01 24 FC 02 64 00             .$..d.
-----
[41] : <Rx> - 10:12:35.122
-Type           : 0x04 (Event)
-EventCode      : 0x00FF (HCI_LE_ExtEvent)
-Data Length    : 0x05 (5) bytes(s)
Event           : 0x0424 (1060) (HCIExt_SetDtmTxPktCntCmdDone)
Status         : 0x00 (0) (SUCCESS)
CmdOpCode       : 0xFC24 (HCIExt_SetDtmTxPktCntCmd)
Dump(Rx):
0000:04 FF 05 24 04 00 24 FC       ...$..$.
-----
[42] : <Tx> - 10:12:56.528
-Type           : 0x01 (Command)
-OpCode         : 0x2034 (HCI_LE_EnhancedTransmitterTest)
-Data Length    : 0x04 (4) byte(s)
TxChannel       : 0x00 (0)
TestData        : 0x28 (40)
DtmPktPattern   : 0x00 (0) (Dtm_PRBS9)
TxPhyType       : 0x02 (2) (Phy 2 Mbps)
Dump(Tx):
```

```

0000:01 34 20 04 00 28 00 02                .4 ..(..
-----
[43] : <Error> - 10:12:56.557
Could Not Convert All The Data In The Following Message
Unable to read beyond the end of the stream.
(Message Maybe Missing Data Bytes To Process)
-----
[44] : <Rx> - 10:12:56.556
-Type           : 0x04 (Event)
-EventCode      : 0x000E (HCI_CommandCompleteEvent)
-Data Length    : 0x04 (4) bytes(s)
  Packets       : 0x01 (1)
  OpCode        : 0x2034 (HCI_LE_EnhancedTransmitterTest)
  Status        : 0x00 (0) (SUCCESS)
Dump (Rx):
0000:04 0E 04 01 34 20 00                ....4 .
-----
[45] : <Rx> - 10:12:56.610
-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          .....
-----

```

3.6.2 Test logs for 500Kbps

Logs on Rx side:

```

[38] : <Tx> - 10:16:37.520
-Type           : 0x01 (Command)
-OpCode         : 0x2033 (HCI_LE_EnhancedReceiverTest)
-Data Length    : 0x03 (3) byte(s)
  RxChannel     : 0x00 (0)
  Phy           : 0x03 (3) (Coded Phy)
  DtmModeIndex  : 0x00 (0) (Standard Modulation Index)
Dump (Tx):
0000:01 33 20 03 00 03 00                .3 ....
-----
[39] : <Error> - 10:16:37.545
Could Not Convert All The Data In The Following Message
Unable to read beyond the end of the stream.
(Message Maybe Missing Data Bytes To Process)
-----
[40] : <Rx> - 10:16:37.541
-Type           : 0x04 (Event)
-EventCode      : 0x000E (HCI_CommandCompleteEvent)
-Data Length    : 0x04 (4) bytes(s)
  Packets       : 0x01 (1)
  OpCode        : 0x2033 (HCI_LE_EnhancedReceiverTest)
  Status        : 0x00 (0) (SUCCESS)
Dump (Rx):
0000:04 0E 04 01 33 20 00                ....3 .
-----
[41] : <Tx> - 10:16:53.433
-Type           : 0x01 (Command)
-OpCode         : 0x201F (HCI_LE_TestEnd)
-Data Length    : 0x00 (0) byte(s)
Dump (Tx):
0000:01 1F 20 00                .. .
-----
[42] : <Rx> - 10:16:53.450
-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  : 0x00C8 (200)
Dump (Rx):
0000:04 0E 06 01 1F 20 00 C8 00          .....
-----

```

Logs on Tx side:

```
[50] : <Tx> - 10:15:55.208
-Type           : 0x01 (Command)
-OpCode         : 0xFC24 (HCIExt_SetDtmTxPktCntCmd)
-Data Length    : 0x02 (2) byte(s)
TxPktCount      : 0x00C8 (200) (Unknown DtmTxPktCountType #200)
Dump(Tx):
0000:01 24 FC 02 C8 00          .$....
-----
[51] : <Rx> - 10:15:55.230
-Type           : 0x04 (Event)
-EventCode      : 0x00FF (HCI_LE_ExtEvent)
-Data Length    : 0x05 (5) bytes(s)
Event           : 0x0424 (1060) (HCIExt_SetDtmTxPktCntCmdDone)
Status          : 0x00 (0) (SUCCESS)
CmdOpCode       : 0xFC24 (HCIExt_SetDtmTxPktCntCmd)
Dump(Rx):
0000:04 FF 05 24 04 00 24 FC    ...$..$.
-----
[52] : <Tx> - 10:16:46.505
-Type           : 0x01 (Command)
-OpCode         : 0x2034 (HCI_LE_EnhancedTransmitterTest)
-Data Length    : 0x04 (4) byte(s)
TxChannel       : 0x00 (0)
TestData        : 0x28 (40)
DtmPktPattern   : 0x00 (0) (Dtm_PRBS9)
TxPhyType       : 0x04 (4) (Phy 500 kbps Coded C2)
Dump(Tx):
0000:01 34 20 04 00 28 00 04    .4 ..(..
-----
[53] : <Error> - 10:16:46.529
Could Not Convert All The Data In The Following Message
Unable to read beyond the end of the stream.
(Message Maybe Missing Data Bytes To Process)
-----
[54] : <Rx> - 10:16:46.527
-Type           : 0x04 (Event)
-EventCode      : 0x000E (HCI_CommandCompleteEvent)
-Data Length    : 0x04 (4) bytes(s)
Packets         : 0x01 (1)
OpCode          : 0x2034 (HCI_LE_EnhancedTransmitterTest)
Status          : 0x00 (0) (SUCCESS)
Dump(Rx):
0000:04 0E 04 01 34 20 00      ....4 .
-----
[55] : <Rx> - 10:16:46.896
-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    .....
-----
```

3.6.3 Test logs for 125Kbps

Logs on Rx side

```
[45] : <Tx> - 10:20:05.328
-Type           : 0x01 (Command)
-OpCode         : 0x2033 (HCI_LE_EnhancedReceiverTest)
-Data Length    : 0x03 (3) byte(s)
RxChannel       : 0x00 (0)
Phy             : 0x03 (3) (Coded Phy)
DtmModeIndex    : 0x01 (1) (Stable Modulation Index)
Dump(Tx):
0000:01 33 20 03 00 03 01      .3 ....
-----
[46] : <Error> - 10:20:05.356
Could Not Convert All The Data In The Following Message
Unable to read beyond the end of the stream.
(Message Maybe Missing Data Bytes To Process)
-----
```

```
[47] : <Rx> - 10:20:05.355
-Type           : 0x04 (Event)
-EventCode      : 0x000E (HCI_CommandCompleteEvent)
-Data Length    : 0x04 (4) bytes(s)
  Packets       : 0x01 (1)
  OpCode        : 0x2033 (HCI_LE_EnhancedReceiverTest)
  Status        : 0x00 (0) (SUCCESS)
Dump (Rx):
0000:04 0E 04 01 33 20 00          ....3 .
-----
[48] : <Tx> - 10:20:31.248
-Type           : 0x01 (Command)
-OpCode         : 0x201F (HCI_LE_TestEnd)
-Data Length    : 0x00 (0) byte(s)
Dump (Tx):
0000:01 1F 20 00          . . .
-----
[49] : <Rx> - 10:20:31.272
-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  : 0x00C8 (200)
Dump (Rx):
0000:04 0E 06 01 1F 20 00 C8 00    .....
-----
```

Logs on Tx side

```
[58] : <Tx> - 10:19:31.728
-Type           : 0x01 (Command)
-OpCode         : 0xFC24 (HCIExt_SetDtmTxPktCntCmd)
-Data Length    : 0x02 (2) byte(s)
  TxPktCount     : 0x00C8 (200) (Unknown DtmTxPktCountType #200)
Dump (Tx):
0000:01 24 FC 02 C8 00          .$....
-----
[59] : <Rx> - 10:19:31.751
-Type           : 0x04 (Event)
-EventCode      : 0x00FF (HCI_LE_ExtEvent)
-Data Length    : 0x05 (5) bytes(s)
  Event          : 0x0424 (1060) (HCIExt_SetDtmTxPktCntCmdDone)
  Status        : 0x00 (0) (SUCCESS)
  CmdOpCode     : 0xFC24 (HCIExt_SetDtmTxPktCntCmd)
Dump (Rx):
0000:04 FF 05 24 04 00 24 FC    ...$.$.
-----
[60] : <Tx> - 10:20:23.408
-Type           : 0x01 (Command)
-OpCode         : 0x2034 (HCI_LE_EnhancedTransmitterTest)
-Data Length    : 0x04 (4) byte(s)
  TxChannel      : 0x00 (0)
  TestData       : 0x28 (40)
  DtmPktPattern  : 0x00 (0) (Dtm_PRBS9)
  TxPhyType      : 0x03 (3) (Phy 125 kbps Coded C8)
Dump (Tx):
0000:01 34 20 04 00 28 00 03    .4 ..(..
-----
[61] : <Error> - 10:20:23.437
Could Not Convert All The Data In The Following Message
Unable to read beyond the end of the stream.
(Message Maybe Missing Data Bytes To Process)
-----
[62] : <Rx> - 10:20:23.430
-Type           : 0x04 (Event)
-EventCode      : 0x000E (HCI_CommandCompleteEvent)
-Data Length    : 0x04 (4) bytes(s)
  Packets       : 0x01 (1)
  OpCode        : 0x2034 (HCI_LE_EnhancedTransmitterTest)
  Status        : 0x00 (0) (SUCCESS)
Dump (Rx):
0000:04 0E 04 01 34 20 00          ....4 .
-----
[63] : <Rx> - 10:20:24.177
-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
-----

```

3.7 The Difference Between HCI_EXT_ModemTestTxCmd and HCI_LE_Transmitter_Test

For FCC regulatory testing, set HCI_EXT_ModemTestTxCmd (or HCIExt_EnhancedModemTestTxCmd for different PHY rate) instead of HCI_LE_Transmitter.

The HCI_EXT_ModemTestTxCmd includes an input for modulated or unmodulated carrier. Before setting HCI_EXT_ModemTestTxCmd, set the HCIExt_SetTxPowerCmd for Tx power.

After it has been sent the HCI_LE_Transmitter_Test command, the CC26xx/CC13xx device sends out LE test packets. This can be used for In-band emission. The HCI_LE_TransmitterTest command configures the Bluetooth LE device to transmit at maximum power. To change the TxPower used, use the command HCI_EXT_SetMaxDtmTxPowerCmd in advance.

Below summarizes the test commands for different tests:

- For FCC testing, set HCIExt_SetTxPowerCmd, then set HCIExt_ModemTestTxCmd (or HCIExt_EnhancedModemTestTxCmd)
- For in-band emission testing, set HCI_EXT_SetMaxDtmTxPowerCmd, then set HCI_LE_Transmitter_Test.

[11] and [12] provide the details for the HCI commands.

4 Summary

At the early stage while developing Bluetooth Low Energy products with CC26xx/CC135x device, you can use SmartRF Studio to do RF performance testing. With the Bluetooth tester [8][9][10], you can flash the CC26xx/CC13xx device with host_test FW or firmware enabling Production Test Mode. Then, the Bluetooth tester and CC26xx/CC13xx will communicate through HCI interface and automatically run the testing.

For the manufacturing line, user can use SmartRF Studio or BTool (HCI commands) to do RF test depending on the availability of JTAG debug lines.

For the products with external host/MCU connecting to CC26xx/CC13xx device, clone the HCI raw commands and let the external host/MCU set HCI commands to CC26xx/CC13xx device. Before the test, the external host/MCU must flash the host_test FW through the bootloader of CC26xx/CC13xx device [13]. After finishing test, the external host/MCU can flash the released/production FW to CC26xx/CC13xx device.

Figure 4-1 summarizes the guide about how to plan for the Bluetooth RF radio test. SimpleLink MCU provides flexible tools for RF performance testing. You can evaluate your own resource and adopt the most suitable test processes.

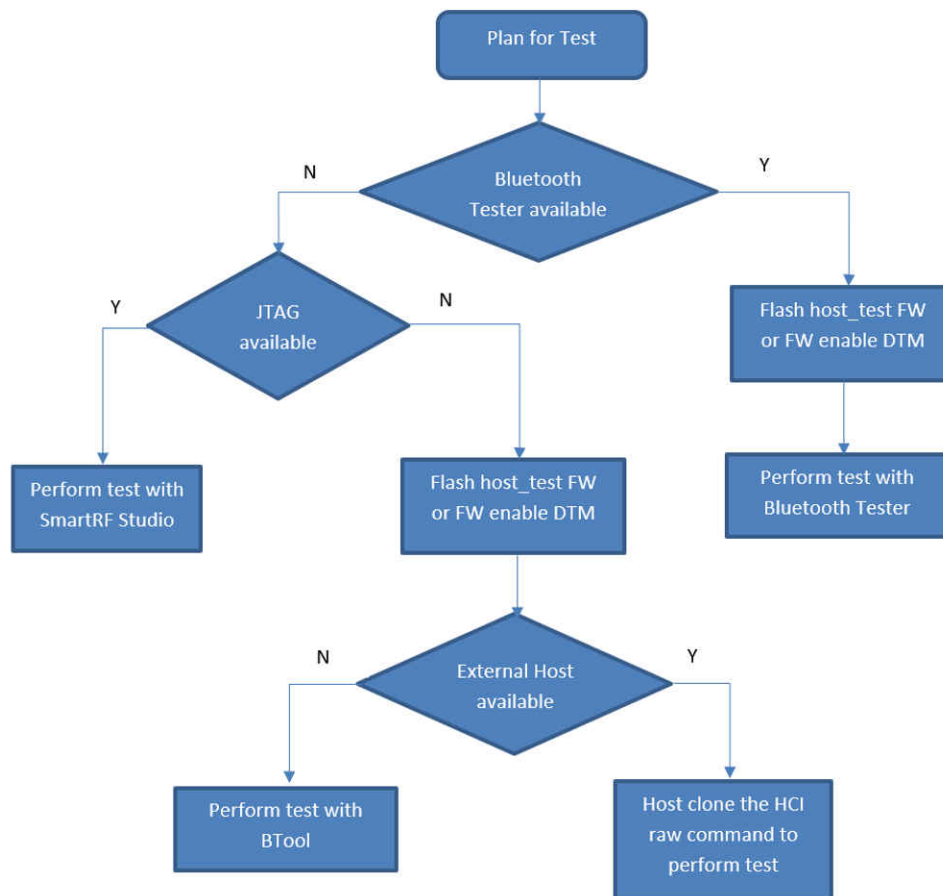


Figure 4-1. Test Plan Flow Chart

5 References

1. [SMARTRF-STUDIO](#)
2. [Using BTool](#)
3. [Defining Application Behavior](#)
4. [Bluetooth Low Energy Fundamentals](#)
5. Texas Instruments: [SmartRF Studio 7 Tutorial](#) (Exercise 1 and 2)
6. [Task 1: SmartRF Studio ↔ SmartRF Studio](#)
7. Texas Instruments: [Basic RF Testing of CCxxxx Devices](#)
8. <https://www.litepoint.com/chipsets/#>
9. [Bluetooth Test Set: MT8852B](#)
10. [R&S@CMW270 wireless connectivity tester](#)
11. [HCI Commands](#)
12. [HCI Vendor Specific Guide](#)
13. Texas Instruments: [CC2538/CC26x0/CC26x2 Serial Bootloader Interface](#)

6 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision * (February 2022) to Revision A (August 2022)	Page
• Updated Section 3.2	5
• Updated Section 3.3	8

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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 will 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](#) 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.

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated