

GR551x DTM Test Application Note

Version: 2.0

Release Date: 2022-02-20

Copyright © 2022 Shenzhen Goodix Technology Co., Ltd. All rights reserved.

Any excerption, backup, modification, translation, transmission or commercial use of this document or any portion of this document, in any form or by any means, without the prior written consent of Shenzhen Goodix Technology Co., Ltd. is prohibited.

Trademarks and Permissions

GODIX and other Goodix trademarks are trademarks of Shenzhen Goodix Technology Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Disclaimer

Information contained in this document is intended for your convenience only and is subject to change without prior notice. It is your responsibility to ensure its application complies with technical specifications.

Shenzhen Goodix Technology Co., Ltd. (hereafter referred to as "Goodix") makes no representation or guarantee for this information, express or implied, oral or written, statutory or otherwise, including but not limited to representation or guarantee for its application, quality, performance, merchantability or fitness for a particular purpose. Goodix shall assume no responsibility for this information and relevant consequences arising out of the use of such information.

Without written consent of Goodix, it is prohibited to use Goodix products as critical components in any life support system. Under the protection of Goodix intellectual property rights, no license may be transferred implicitly or by any other means.

Shenzhen Goodix Technology Co., Ltd.

Headquarters: 2F. & 13F., Tower B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

TEL: +86-755-33338828 FAX: +86-755-33338099

Website: www.goodix.com



Preface

Purpose

This document introduces the test framework, test methods, and HCI control frame format of Direct Test Mode (DTM) in Bluetooth Low Energy (Bluetooth LE) applications. It describes how to use a Bluetooth tester to test the radio frequency (RF) performances of GR551x System-on-Chips (SoC)s and demonstrates a GR551x DTM example, helping users quickly get started with secondary development.

Audience

This document is intended for:

- GR551x user
- GR551x developer
- GR551x tester
- Hobbyist developer
- Technical writer

Release Notes

This document is the seventh release of GR551x DTM Test Application Note, corresponding to GR551x SoC series.

Revision History

Version	Date	Description
1.0	2019-12-08	Initial release
1.3	2020-03-16	Updated the release time in the footers.
1.5	2020-05-30	Modified the "Deliver HCI Commands" section.
1.6	2020-06-30	Updated the document version based on Software Development Kit (SDK) changes.
1.7	2020-09-20	 Added two private commands: the LE Generate Signal Carrier Wave Command (to generate single carrier waves), and the LE Modify TX Power Command (to set the TX power for DTM TX tests), and updated relevant content. Modified the section "Deliver HCI Commands" to "DTM Tests (Manual Operation)". Added the "Test Considerations" section.
1.8	2021-06-26	 Updated GR551x to GR55xx. Optimized descriptions in the "Initial Operation" section.
1.9	2021-08-09	Changed the section "Supported Development Platform" into "Preparation".
2.0	2022-02-20	Modified the file name of the example firmware based on SDK changes.



Contents

Preface	
1 Introduction	1
2 DTM Test Framework and Procedures	2
2.1 DTM Test Framework	2
2.2 DTM Test Procedures	2
3 Initial Operation	4
3.1 Preparation	4
3.2 Firmware Programming	
3.3 Test and Verification	4
3.3.1 DTM Testing with TLF3000	5
3.3.2 DTM Testing with R&S [®] CMW500	8
4 Application Details	12
4.1 Project Directory	12
4.2 Configure UART Ports	12
4.3 Transplant DTM Function	13
5 DTM Tests (Manual Operation)	14
5.1 Introduction to HCI	14
5.1.1 HCI Command Format	14
5.1.2 HCI Event Format	14
5.2 Deliver HCI Commands	15
5.3 DTM TX Test (Manual Operation)	15
5.3.1 Initiate a DTM TX Test	15
5.3.2 End a DTM TX Test	16
5.4 DTM RX Test (Manual Operation)	17
5.4.1 Initiate a DTM RX Test	
5.4.2 End a DTM RX Test	17
5.5 Private Commands	
5.5.1 LE Generate Signal Carrier Wave Command	
5.5.2 LE Modify TX Power Command	
5.6 Test Considerations	20
6 FAQ	21
6.1 Why Do I Fail to Deliver Commands and Encounter Invalid Commands?	21
6.2 Why Do I Fail to Execute Commands That Have Been Delivered?	21
6.3 Why Do I Fail to Generate Single Carrier Waves?	21
6.4 Why Do I Fail to Modify Transmission Power?	22
7 Appendix: Common HCI Commands in RF Performance Test	2 3



7.1 LE Reset Command	23
7.2 LE Transmitter Test Command	23
7.3 LE Enhanced Transmitter Test Command	24
7.4 LE Receiver Test Command	25
7.5 LE Enhanced Receiver Test Command	26
7.6 LE Test End Command	27
7.7 LE Generate Signal Carrier Wave Command	27
7.8 LE Modify TX Power Command	28



1 Introduction

The Direct Test Mode (DTM) is a standard mechanism defined by Bluetooth Special Interest Group (Bluetooth SIG) to test the radio frequency (RF) performance of Bluetooth devices for all Bluetooth channels. In general, DTM is used in application validation and product line tests, enabling RF engineers to effortlessly verify and optimize Bluetooth RF performances. Tested RF factors include:

- Transmission power and receiver sensitivity
- Frequency offset and drift
- Modulation characteristics
- Packet error rate

The DTM test on GR551x System-on-Chips (SoC)s is based on UART physical interfaces and a Bluetooth tester. The Bluetooth tester controls GR551x RF physical layer (PHY) by using Host Controller Interface commands (HCI CMD) to implement test procedures.

Before getting started, you can refer to the following documents.

Table 1-1 Reference documents

Name	Description
GR551x Developer Guide	Introduces GR551x Software Development Kit (SDK) and how to develop and debug applications based on the SDK.
Keil User Guide	Offers detailed Keil operational instructions. Available at www.keil.com/support/man/docs/uv4/ .
RF PHY Bluetooth Test Suite TS	Introduces RF PHY test specifications. For Core-Test Requirements, see <u>www.bluetooth.com</u> .



2 DTM Test Framework and Procedures

This chapter introduces the test framework and fundamental test procedures of DTM.

2.1 DTM Test Framework

A DTM test involves three types of test devices:

- Device Under Test (DUT)
- Upper Tester (UT)
- Lower Tester (LT)

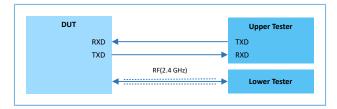


Figure 2-1 DTM test framework

DUT can work as a Transmitter Device (TXD) that transmits (TX) Bluetooth LE packets or a Receiver Device (RXD) that receives (RX) Bluetooth LE packets.

TX test

As a TXD, the DUT transmits test group sequences at a specified frequency to the LT that analyzes the transmission power, frequency offset, and modulation characteristics based on the received signals.

RX test

As an RXD, the DUT receives test group sequences transmitted by the LT at a specified frequency that is the same with that in LT transmission. After receiving, Bit Error Rate (BER) and other indicators received by the DUT are used to measure the RXD performance.

UT communicates with DUT through UART, and LT communicates with DUT through Bluetooth LE links. The UART parameters of DUT are as follows:

Table 2-1 UART	parameter	setting in	וטט-וט	communications
----------------	-----------	------------	--------	----------------

Baud Rate (Recommended)	Bit Size	Checksum Bit	Stop Bit	Flow Control
1200, 2400, 9600, 14400, 19200, 38400, 57600, 115200, 230400, 460800, and 921600	8 bits	N/A	1 bit	Not supported

2.2 DTM Test Procedures

During DTM tests, the UT transmits an HCI test command through a UART interface to the DUT that returns the test Status Event or Packet Report Event to the UT. Before receiving a reply from the DUT, the UT does not deliver another test command. If the UT does not receive any reply from the DUT within a regulated time period, the UT delivers a reset command to the DUT.



The mapping between RF test commands/events and HCl commands/events is shown in Table 2-2.

Table 2-2 Mapping between RF test commands/events and HCI commands/events

RF Test Command/Event	HCI Command/Event
LE_TRANSMITTER_TEST	LE Transmitter Test Command or LE Enhanced Transmitter Test Command
LE_RECEIVER_TEST	LE Receiver Test Command or LE Enhanced Receiver Test Command
LE_TEST_END	LE Test End Command
LE_STATUS	Command Complete Event
LE_PACKET_REPORT	Command Complete Event

Figure 2-2 illustrates the TX, RX, and End procedures of a DTM test using the LE Transmitter Test Command, LE Receiver Test Command, and LE Test End Command as examples.

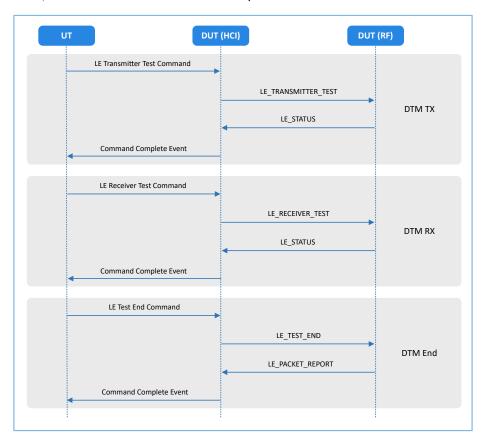


Figure 2-2 DTM test procedure example



3 Initial Operation

This chapter introduces how to quickly verify a DTM example in GR551x SDK.



SDK_Folder is the root directory of GR551x SDK.

3.1 Preparation

You need the following hardware and software to set up a DTM test environment.

Hardware preparation

Table 3-1 Hardware preparation

Name	Description
Development board	GR5515 Starter Kit Board (SK Board)
Connection cable	Micro USB 2.0 serial cable
Bluetooth tester	R&S CBT, MT8852B, CMW500, TLF3000, or others.

Software preparation

Table 3-2 Software preparation

Name	Description
Windows	Windows 7/Windows 10
J-Link driver	A J-Link driver. Available at www.segger.com/downloads/jlink/.
GRUart (Windows)	A serial port debugging tool. Available in SDK_Folder\tools\GRUart.
GProgrammer (Windows)	A programming tool. Available in SDK_Folder\tools\GProgrammer.
TLF3000 RF-PHY Tester and driver	Downloaded at: http://www.fte.com/products/tlf3000.aspx.
R&S®CMW500 tester and driver	Downloaded at: https://www.rohde-schwarz.com.

3.2 Firmware Programming

The source code of the DTM example is in SDK_Folder\projects\ble\dtm\direct_test_mode.

You can programme *direct_test_mode.bin* to the SK Board through GProgrammer. For details, see *GProgrammer User Manual*.

Note:

• The direct_test_mode.bin is in SDK_Folder\projects\ble\dtm\direct_test_mode\build\.

3.3 Test and Verification



The DTM test on GR551x SoCs adopts a two-wire UART to transmit HCI frame formats to control the RF PHY. Therefore, any testers supporting HCI format can be used in DTM tests and verification on GR551x SoCs. This section introduces the DTM test and verification procedures by using TLF3000 and R&S[®]CMW500.

3.3.1 DTM Testing with TLF3000

Set up a DTM test environment with TLF3000.

- 1. Install TLF3000 RF-PHY Tester software and drivers on a PC.
- 2. Connect the TLF3000 hardware to the PC and the SK Board to the PC with Micro USB cables respectively. Connect the RF pin on the DUT to the TX/RX port on the TLF3000 hardware through a cable. The details are shown in the figure below.

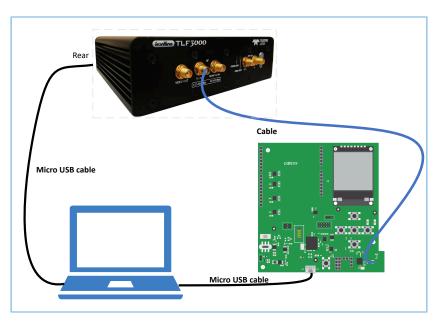


Figure 3-1 Hardware connection in a TLF3000 test environment

DTM test procedures with TLF3000:

1. Power the TLF3000 hardware on, start the TLF3000 software, and select the TLF3000 model used in the DTM test. Click **RUN APP** to enter the TLF3000 software GUI.



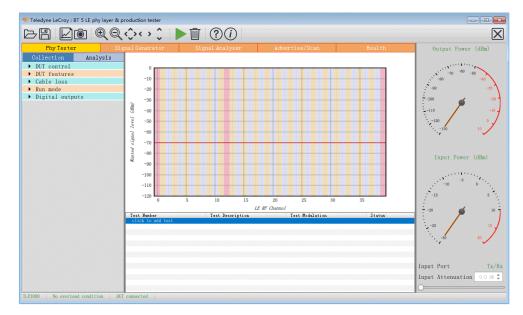


Figure 3-2 Hardware connection in a TLF3000 test environment

 Configure DUT control parameters. Select a serial port from the Comport list. Select H4 from Interface and 115200 from Baud rate. Select None from the HW flow, SW flow, and Parity bits lists. Select One from the Stop bits list.

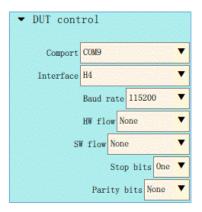


Figure 3-3 DUT control configuration

3. Open the **DUT features** pane, and click **Query DUT** to obtain the DUT-related features.



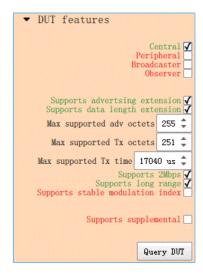


Figure 3-4 Obtaining DUT features

4. Select a running mode in the **Run mode** pane.

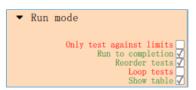


Figure 3-5 Configuring Run mode

Click click to add test, and the Phy Level Test window opens. Select test cases in the TP/TRM-LE/CA/BV-12 tab.
 Configure the Channels, Packet Lengths, and Number of packets for the selected test cases. After configuration, click Apply to complete test adding.

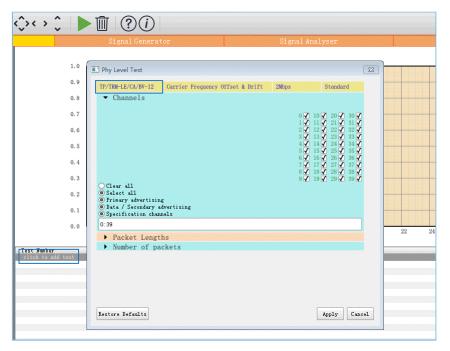


Figure 3-6 Selecting test cases



6. Click ■ to start the DTM test. When the test completes, a window containing logs in HTML format automatically opens. Click ■ to store the logs in a HTML file, or click ■ to save the log screenshot.

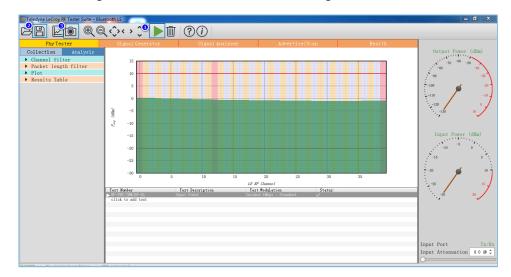


Figure 3-7 Running test cases

3.3.2 DTM Testing with R&S[®]CMW500

Set up a DTM test environment with R&S CMW500.

- 1. Install R&S CMW500 automation software tool (R&S CMWrun Sequencer Software Tool) and drivers on a PC.
- 2. Connect the R&S CMW500 USB port to the UART port on the SK Board with a Micro USB cable. Connect the RF COM port on the R&S CMW500 to the RF port on the SK Board with a cable. The connection diagram is displayed in the the figure below.

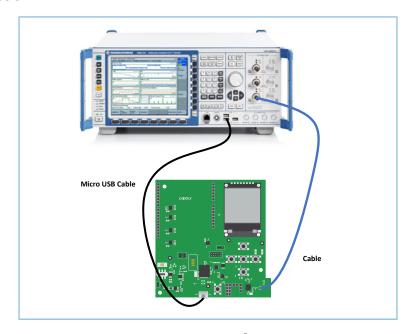


Figure 3-8 Hardware connection in an R&S CMW500 test environment



DTM test procedures with R&S®CMW500:

- 1. Configuring the Bluetooth signaling test mode on the R&S CMW500. In this mode, the R&S CMW500 automatically delivers HCI commands to interact with the SK Board, enabling automated DTM tests.
 - (1). On the R&S CMW500, press **MEASURE** to open the **Measurement Controller** window, and select **Measurements** in **Bluetooth**.

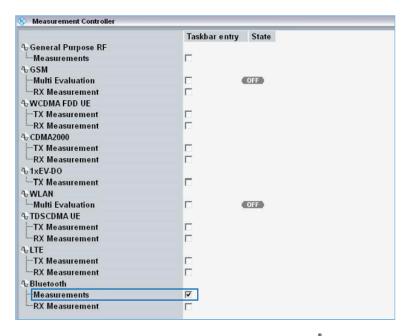


Figure 3-9 Configuring Bluetooth measurements for R&S CMW500

(2). On the R&S CMW500, press **SIGNALGEN** to open the **Generator/Signaling Controller** window, and select **Signaling in Bluetooth**.

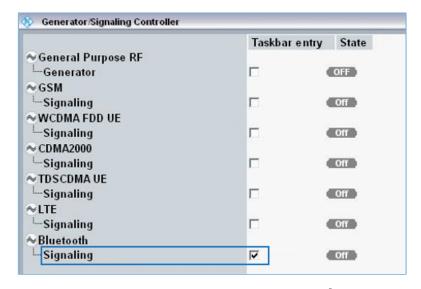


Figure 3-10 Configuring Bluetooth signaling mode for R&S CMW500

2. Connecting to UART



- (1). Confirm that the R&S CMW500 serial drivers have been properly installed on the PC. Make sure that the R&S CMW500 tester has been correctly connected to the UART port of the SK Board.
- (2). The R&S CMW500 tester automatically identifies the COM port used by the UART, and developers only need to configure the baud rate, odd-even parity, and other parameters related to the DTM test firmware. After setting these parameters, click **Connection Check** to connect the R&S CMW500 tester to the UART. You can implement the DTM test after successful connection to UART.

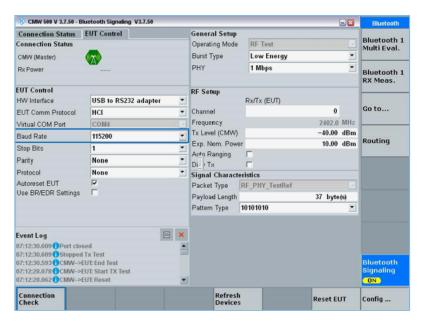


Figure 3-11 Connecting to UART

3. Running DTM automation test

Start R&S CMWrun, and select **CMW RUN**. Load the Bluetooth LE test program, BLE_PHY_5_0_0, and click be to run the DTM test.



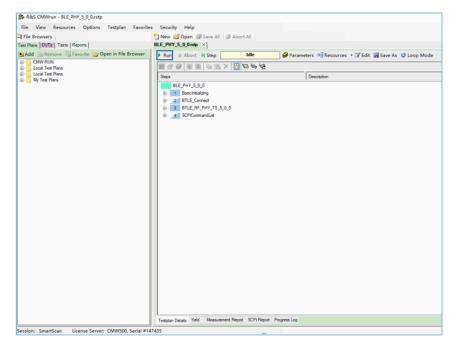


Figure 3-12 R&S®CMW500 DTM test interface

4. Generating a test report.

After the test completes, the R&S[®]CMW500 software tool automatically generates a test report, as shown in Figure 3-13.

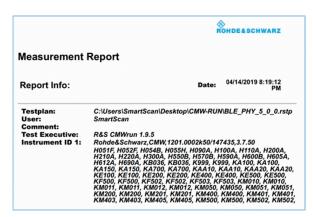


Figure 3-13 CMW500 DTM test report

Note:

The R&S CMW500 automation software tool (R&S CMWrun) should be purchased by users.



4 Application Details

Settings for serial ports vary, depending on the DUT. Users can modify the configurations on the serial ports of the direct_test_mode project to meet the DTM test requirements from the DUT. By default, an SK Board is used as the DUT for direct_test_mode.

4.1 Project Directory

The direct_test_mode project is in SDK_Folder\projects\ble\dtm\direct_test_mode, and the project file is located in the Keil 5 folder.

Double-click the project file, *direct_test_mode.uvprojx*, to view the direct_test_mode project directory structure of the DTM application in Keil. For related files, see Table 4-1.

Table 4-1 Description of direct_test_mode files

Group	File	Description
gr_libraries	hci_uart.c	This file adapts to the UART drivers of Bluetooth LE Stack DTM test.
user_app	main.c	This file contains the main() function, enabling registration of HCI UART interfaces.

4.2 Configure UART Ports

GR551x provides two UART ports: UART0 and UART1. UART0 is used by the SK Board, and GPIO10/GPIO11 are used for TX/RX of UART. GPIO2/GPIO5 are used for signal flow control, with baud rate at 115200. Configurations shall be made based on I/O mapping of the SK Board, which are described below:

Path: user_app\GR5515_SK.h under direct_test_mode project directory

Name: HCI UART IO CONFIG

```
/******HCI UART IO CONFIG****************/
#define HCI_UART_ID
                                       APP UART ID 0
#define HCI UART FLOW ON
                                      0
#define HCI UART BAUDRATE
                                      115200
#define HCI_UART_TRN_PORT
                                     APP IO TYPE NORMAL
#define HCI UART FLOW PORT
                                     APP IO TYPE NORMAL
#define HCI UART TX PIN
                                     APP IO PIN 10
#define HCI UART RX PIN
                                     APP IO PIN 11
#define HCI UART CTS PIN
                                      APP IO PIN 2
#define HCI UART RTS PIN
                                     APP IO PIN 5
#define HCI UART TX PINMUX
                                      APP IO MUX 2
#define HCI UART RX PINMUX
                                      APP IO MUX 2
#define HCI UART CTS PINMUX
                                     APP IO MUX 0
#define HCI UART RTS PINMUX
                                      APP IO MUX 0
#define HCI UART TRIGGER PIN
                                       AON GPIO PIN 1
```



Note:

- Configurations listed above are used when an SK Board serves as a DUT. You need to modify the settings of the serial ports according to the hardware layout of the DUT in use.
- If you want to use UART1 as the HCI communications interface, define the HCI_UART_ID macro as APP_UART_ID_1 in *GR5515_SK.h.* Modify the UART pin macro in *GR5515_SK.h* based on the UART I/O mapping on the DUT. For more information about I/O multiplexing on GR551x SoCs, refer to *GR551x Datasheet*.

4.3 Transplant DTM Function

This section elaborates on how to transplant the DTM function to a Bluetooth LE application project.

To do that, follow the steps below:

- 1. Add the hci_uart.c file to the application project, and refer to hci_uart.c in the main.c file.
- 2. Call the ble_stack_init() function to initialize the Bluetooth Protocol Stack.
- 3. During initialization, call the ble_hci_uart_init() function to initialize the UART port and communications protocol stack required in a DTM test.

Path: user_app\main.c under direct_test_mode.uvprojx

Name: main();

```
#include "hci_uart.h"

static uint8_t s_hci_buffer[256] = {0};

int main (void)
{
    ble_stack_init(&app_ble_callback, &heaps_table);/*< init ble stack*/
    ble_hci_uart_init(s_hci_buffer, sizeof(s_hci_buffer));
    //loop
    while(1)
    {
        }
}</pre>
```



5 DTM Tests (Manual Operation)

In general, DTM tests are performed by dedicated instruments automatically. The DUT communicates with instrument(s) through HCl commands. Alternatively, users can also simulate a DTM test instrument to perform tests by sending HCl commands manually.

This section introduces how to use GRUart, a serial port debugging tool, to deliver HCI commands and to control the DUT to enter the TX or RX mode, as well as some more functions. Operations of UT are customized according to the Core-Test Requirements, and are not described in this document.



DTM tests shall be performed in shielding boxes.

5.1 Introduction to HCI

The UT controls the DUT to perform DTM tests by delivering HCI commands. The HCI commands are used to configure the DUT status or send requests to the DUT to complete related operations. The HCI Event Packets returned to the UT from the DUT are used to respond to relevant information. Each HCI command corresponds to an HCI event.

5.1.1 HCI Command Format

The data structure of an HCI command is shown in Figure 5-1.

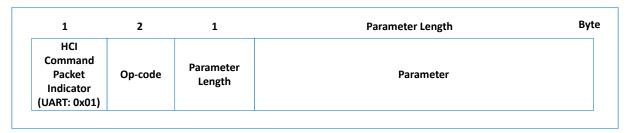


Figure 5-1 Format of an HCI Command Packet

- HCI Command Packet indicator: Indicates the HCI type. The indicator byte for the UART interface is 0x01.
- Op-code: a 2-byte code that identifies the command type.
- Parameter Length: the total length of all parameters. Length: 1 byte.
- Parameter: parameter description. The parameter quantity and length vary depending on the command.

5.1.2 HCI Event Format

The data structure of an HCI event is shown in Figure 5-2.



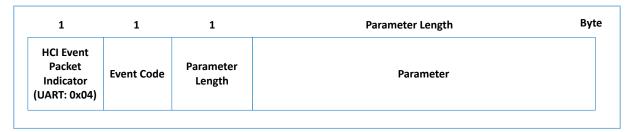


Figure 5-2 Format of an HCI Event Packet

- HCI Event Packet indicator: Indicates the HCI type. The indicator byte for the UART interface is 0x04.
- Event Code: a 1-byte code that identifies the command type.
- Parameter Length: the total length of all parameters. Length: 1 byte.
- Parameter: event parameter. The parameter quantity and length vary depending on the command.

5.2 Deliver HCI Commands

Before delivering an HCI test command, complete the following steps:

- Power the GR551x SoC on (with DTM test firmware installed) to wait for HCl commands.
- Start GRUart, and select baud rate, data bit, stop bit, and parity parameters based on the UART configurations in the DTM test firmware. Enable the UART.

Use GRUart to deliver an HCI command. DTM-test-related HCI commands include:

- LE Reset Command
- LE Transmitter Test Command
- LL Enhanced Transmitter Test Command
- LE Receiver Test Command
- LE Enhanced Receiver Test Command
- LE Test End Command

The sections below introduce three common HCI commands related to DTM tests: LE Transmitter Test Command, LE Receiver Test Command, and LE Test End Command.

5.3 DTM TX Test (Manual Operation)

5.3.1 Initiate a DTM TX Test

The LE Transmitter Test Command is used to enable the DTM TX test. The GR551x SoC generates a specific type of test packets on a specified channel at a fixed interval.



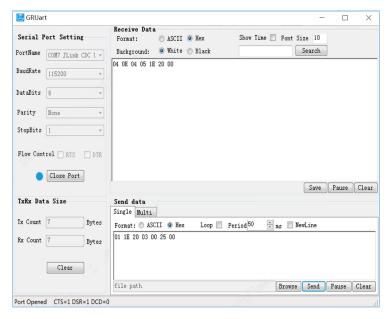


Figure 5-3 Transmitting command and response

As shown in Figure 5-3, the GRUart sends an HCI command in hexadecimal format, 01 1E 20 03 00 25 00, to the tested GR551x SoC. The command indicates that the RF physical layer (PHY) sends a 37-byte pseudorandom binary sequence 9 (PRBS9) test sequence packet on a channel at a frequency of 2402 kHz. After the GR551x SoC executes the command, the GRUart receives an HCI event in hexadecimal format, 04 0E 04 05 1E 20 00.

For command details, see "Section 7.2 LE Transmitter Test Command".

5.3.2 End a DTM TX Test

The LE Test End Command is used to end an ongoing DTM test. To start a new DTM test, the ongoing DTM test shall be terminated.

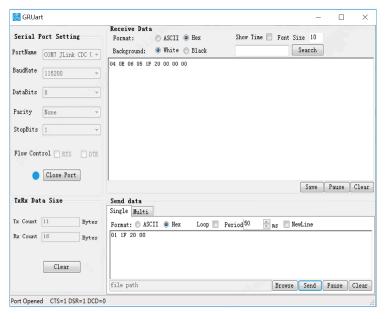


Figure 5-4 To end an ongoing test



As shown in Figure 5-4, GRUart sends the hexadecimal HCl command 01 1F 20 00 to the GR551xx SoC to terminate the ongoing DTM test. When the GR551x SoC starts to act accordingly, GRUart will receive the reply, which is a hexadecimal HCl event 04 0E 06 05 1F 20 00 00 00.

For command details, see "Section 7.6 LE Test End Command".

5.4 DTM RX Test (Manual Operation)

5.4.1 Initiate a DTM RX Test

The LE Receiver Test Command is used to enable the DTM RX test. The GR551x SoC sniffs data packets at a fixed frequency and records the number of received data packets.

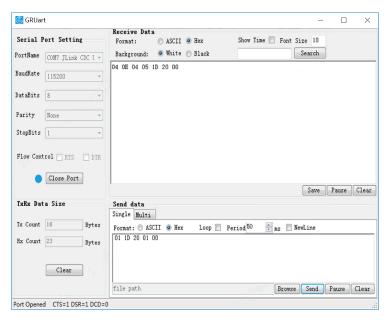


Figure 5-5 Receiving command and response

As shown in Figure 5-5, the GRUart sends an HCI command in hexadecimal format, 01 1D 20 01 00, to the tested GR551x SoC. The command indicates that the RF PHY sniffs data packets on a channel at a frequency of 2402 kHz. After the GR551x SoC executes the command, the GRUart receives an HCI event in hexadecimal format, 04 0E 04 05 1D 20 00.

For command details, see "Section 7.4 LE Receiver Test Command".

5.4.2 End a DTM RX Test

The LE Test End Command stops all ongoing tests. In RX test mode, the number of the received packets is the number of packets that have been actually received.



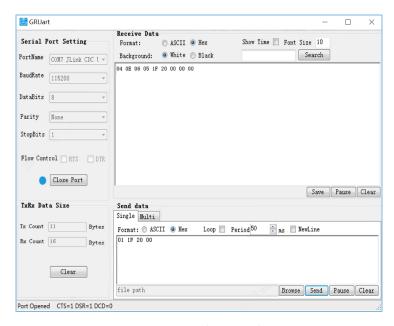


Figure 5-6 End command

As shown in Figure 5-6, the GRUart sends an HCl command in hexadecimal format, 01 1F 20 00, to the tested GR551x SoC. The command indicates that the DTM test ends. After the GR551x SoC executes the command, the GRUart receives an HCl event in hexadecimal format, 04 0E 06 05 1F 20 00 00 00.

For command details, see "Section 7.6 LE Test End Command".

5.5 Private Commands

To simplify DTM test for users, Goodix provides customized private HCl commands in the GR551x DMT project. These customized commands are not standard HCl commands defined in *Bluetooth Core Spec*. This section introduces two private HCl commands: LE Generate Signal Carrier Wave Command and LE Modify TX Power Command.

5.5.1 LE Generate Signal Carrier Wave Command

Use the LE Generate Signal Carrier Wave Command to generate single carrier waves, to start a single-carrier test. The GR551x SoC can generate single-carrier packets on the designated channel.



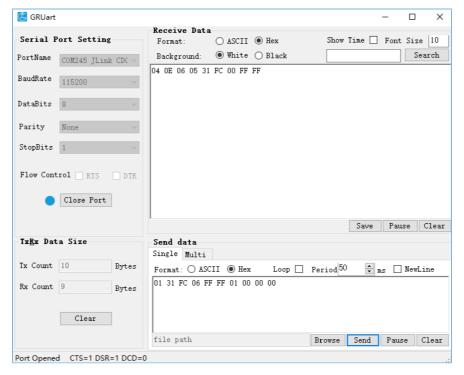


Figure 5-7 Sending command to generate single carrier waves

As shown in Figure 5-7 above, GRUart sends the hexadecimal HCI command 01 31 FC 06 FF FF 01 00 00 00 to GR551x, which means the RF PHY sends a single-carrier test sequence packet to the channel (frequency: 2402 kHz). After the GR551x SoC finishes this task, GRUart will receive the hexadecimal HCI event 04 0E 06 05 31 FC 00 FF FF replied from GR551x.

For command details, see "Section 7.7 LE Generate Signal Carrier Wave Command".

Note:

If there is an ongoing task before you use the LE Generate Signal Carrier Wave Command, end the ongoing task first by using the LE Test End Command.

5.5.2 LE Modify TX Power Command

The LE Modify TX Power Command is used to modify the transmission power for a DTM TX test (initiated by the LE Transmitter Test Command). By default, the transmission power in a DTM TX test is 7 dBm.



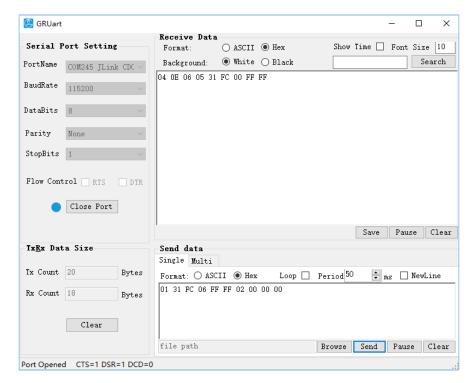


Figure 5-8 Sending the command to modify transmission power

As shown in Figure 5-8 above, GRUart sends the hexadecimal HCl command 01 31 FC 06 FF FF 02 00 00 0 0 to the GR551x SoC to set the transmission power to 0 dBm. After the GR551x SoC completes this task, GRUart will receive the hexadecimal HCl event 04 0E 06 05 31 FC 00 FF FF replied from GR551x.

For command details, see "Section 7.8 LE Modify TX Power Command".

Note:

- The LE Modify TX Power Command can also be used in DTM TX tests initiated by the LE Enhanced Transmitter Test Command.
- Transmission power must be configured prior to a DTM TX test.
- If there is an ongoing task before you use the LE Modify TX Power Command, end the ongoing task first by using the LE Test End Command.

5.6 Test Considerations

- 1. See *RF-PHY.TS.pdf* (available at <u>www.bluetooth.com</u>) for test standards.
- 2. Cable loss shall be taken into account in results of a transmission power test.
- 3. In a DTM TX test, the LE Transmitter Test Command and the LE Enhanced Transmitter Test Command are interchangeable.
- 4. In a DTM RX test, the LE Receiver Test Command and the LE Enhanced Receiver Test Command are interchangeable.



6 FAQ

This chapter describes possible problems, reasons, and solutions during the DTM test.

6.1 Why Do I Fail to Deliver Commands and Encounter Invalid Commands?

Description

Developers fail to deliver commands when using the GRUart serial port debugging tool or testers. Or the delivered commands are invalid.

Analysis

An error occurs in configurations of the GRUart serial port debugging tool, testers, or the serial cable.

Solution

Check whether the UART port on the DUT matches to the UART pin in the DTM firmware. Fore details, see "Section 4.2 Configure UART Ports".

If the UART pin is correctly used, check the configurations (baud rate, stop bit, and data bit) on GRUart and testers. Check the RX and TX pins are correctly connected to each other.

6.2 Why Do I Fail to Execute Commands That Have Been Delivered?

Description

Execution of control commands delivered by serial port tools fails.

Analysis

The failure is caused by incorrectly input commands.

Solution

When using serial port debugging tools to deliver commands, pay attention to the following:

- All commands should be delivered in hexadecimal format.
- The commands shall conform to the HCI-related regulations.
- Do not deliver line breaks using Enter on keyboard when delivering commands.
- Commands of the same type cannot be executed concurrently. End the ongoing command (if any) before running a command.

6.3 Why Do I Fail to Generate Single Carrier Waves?

Description

After GRUart sends the LE Generate Signal Carrier Wave Command, it receives a reply from the GR551x SoC, but the spectrum analyzer has not detected the correct packet.

Analysis

There are ongoing task(s).



- Solution
 - Send the LE Test End Command to end the ongoing task(s).
 - 2. Send the LE Generate Signal Carrier Wave Command to generate single carrier waves.

6.4 Why Do I Fail to Modify Transmission Power?

Description

After GRUart sends the LE Modify TX Power Command, it receives a reply from the GR551x SoC, but according to the spectrum analyzer, the power of the existing waves remains the same.

Analysis

Running LE Modify TX Power Command does not immediately change the power of the current wave.

- Solution
 - 1. Send the LE Test End Command to end the ongoing task(s).
 - 2. Send the LE Modify TX Power Command to modify the transmission power.
 - 3. Send the LE Transmitter Test Command to start a DTM transmission test.



The LE Modify TX Power Command cannot be used in tests with single carrier waves.



7 Appendix: Common HCI Commands in RF Performance Test

7.1 LE Reset Command

Table 7-1 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x03
Command Opcode MSB	0x0c
Parameter Length	0x00

Table 7-2 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E
Parameter Length	0x04
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x03
Command_Opcode_MSB	0x0C
	0x00: Command succeeded.
Status	0x01 to 0xFF: Command failed. See "Volume 2, Part D: Error Codes" in <u>Bluetooth Core Spec</u> for
	a list of error code and descriptions.

7.2 LE Transmitter Test Command

Table 7-3 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x1E
Command Opcode MSB	0x20
Parameter Length	0x03
Frequency	= (F – 2402)/2, where F ranges from 2402 MHz to 2480 MHz. Range: 0x00 to 0x27
Data Length	0x01 to 0xFF: Length in bytes of payload data in each packet.
Package Type	0x00: Pseudorandom binary sequence 9 0x01: Pattern of alternating bits '11110000'



Byte Description	Value
	0x02: Pattern of alternating bits '10101010'
	0x03: Pseudorandom binary 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'

Table 7-4 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E
Parameter Length	0x04
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x1E
Command_Opcode_MSB	0x20
	0x00: Command succeeded.
Status	0x01 to 0xFF: Command failed. See "Volume 2, Part D: Error Codes" in <u>Bluetooth Core Spec</u> for
	a list of error code and descriptions.

7.3 LE Enhanced Transmitter Test Command

Table 7-5 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x34
Command Opcode MSB	0x20
Parameter Length	0x04
Fraguancy	= (F – 2402)/2, where F ranges from 2402 MHz to 2480 MHz.
Frequency	Range: 0x00 to 0x27
Data Length	0x01 to 0xFF: Length in bytes of payload data in each packet.
Package Type	0x00: Pseudorandom binary sequence 9
	0x01: Pattern of alternating bits '11110000'
	0x02: Pattern of alternating bits '10101010'
	0x03: Pseudorandom binary sequence 15
	0x04: Pattern of All '1' bits



Byte Description	Value
	0x05: Pattern of All '0' bits
	0x06: Pattern of alternating bits '00001111'
	0x07: Pattern of alternating bits '0101'
	0x01: Transmitter set to use the LE 1M PHY
РНҮ	0x02: Transmitter set to use the LE 2M PHY
	0x03: Transmitter set to use the LE Coded PHY with S = 8 data coding
	0x04: Transmitter set to use the LE Coded PHY with S = 2 data coding.

Table 7-6 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E
Parameter Length	0x04
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x34
Command_Opcode_MSB	0x20
	0x00: Command succeeded.
Status	0x01 to 0xFF: Command failed. See "Volume 2, Part D: Error Codes" in <u>Bluetooth Core Spec</u> for
	a list of error code and descriptions.

7.4 LE Receiver Test Command

Table 7-7 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x1D
Command Opcode MSB	0x20
Parameter Length	0x01
Frequency	= (F – 2402)/2, where F ranges from 2402 MHz to 2480 MHz. Range: 0x00 to 0x27

Table 7-8 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E



Byte Description	Value
Parameter Length	0x04
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x1D
Command_Opcode_MSB	0x20
	0x00: Command succeeded.
Status	0x01 to 0xFF: Command failed. See "Volume 2, Part D: Error Codes" in <u>Bluetooth Core Spec</u> for
	a list of error code and descriptions.

7.5 LE Enhanced Receiver Test Command

Table 7-9 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x33
Command Opcode MSB	0x20
Parameter Length	0x03
Frequency	= (F – 2402)/2, where F ranges from 2402 MHz to 2480 MHz.
	Range: 0x00 to 0x27
	0x01: Receiver set to use the LE 1M PHY
PHY	0x02: Receiver set to use the LE 2M PHY
	0x03: Receiver set to use the LE Coded PHY
Modulation index	0x00: Assume transmitter will have a standard modulation index
	0x01: Assume transmitter will have a stable modulation index

Table 7-10 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E
Parameter Length	0x04
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x33
Command_Opcode_MSB	0x20
	0x00: Command succeeded.
Status	0x01 to 0xFF: Command failed. See "Volume 2, Part D: Error Codes" in <u>Bluetooth Core Spec</u> for
	a list of error code and descriptions.



7.6 LE Test End Command

Table 7-11 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x1F
Command Opcode MSB	0x20
Parameter Length	0x00

Table 7-12 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E
Parameter Length	0x06
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x1F
Command_Opcode_MSB	0x20
Status	0x00: Command succeeded. 0x01 to 0xFF: Command failed. See "Volume 2, Part D: Error Codes" in <u>Bluetooth Core Spec</u> for a list of error code and descriptions.
Number of Packets Received (LSB)	0xXX
Number of Packets Received (MSB)	0xXX

7.7 LE Generate Signal Carrier Wave Command

Table 7-13 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x31
Command Opcode MSB	0xFC
Parameter Length	0x06
Reserved	0xFFFF
Private Command	0x0001: Generate signal carrier wave.
Frequency	= (F – 2402)/2, where F ranges from 2402 MHz to 2480 MHz.
	Range: 0x0000–0x0027



Table 7-14 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E
Parameter Length	0x06
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x31
Command_Opcode_MSB	0xFC
Status	0x00: Command succeeded. 0x01 to 0xFF: Command failed. See "Volume 2, Part D –Error Codes" in <u>Bluetooth Core Spec</u> for a list of error code and descriptions.
Reserved	0xffff

7.8 LE Modify TX Power Command

Table 7-15 Command format

Byte Description	Value
HCI Command Packet	0x01
Command Opcode LSB	0x31
Command Opcode MSB	0xFC
Parameter Length	0x06
Reserved	0xFFFF
Private Command	0x0002: Modify transmission power.
Transmission Power	Range: 0x0000 - 0x0007 dbm

Table 7-16 Returned events

Byte Description	Value
HCI Event Packet	0x04
Event Code	0x0E
Parameter Length	0x06
Num_HCI_Command_Packets	0x05
Command_Opcode_LSB	0x31
Command_Opcode_MSB	0xFC
	0x00: Command succeeded.
Status	0x01 to 0xFF: command failed. See "Volume 2, Part D –Error Codes" in <u>Bluetooth Core Spec</u> for a list
	of error codes and descriptions.



Byte Description	Value
Reserved	OxFFFF