GR551x Bluetooth Low Energy Throughput Example Application

Version: 1.8

Release Date: 2020-12-15

Shenzhen Goodix Technology Co., Ltd.

Trademarks and Permissions

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 how to use and verify a Bluetooth Low Energy (Bluetooth LE) throughput example in a GR551x SDK, to help 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 sixth release of *GR551x Bluetooth Low Energy Throughput Example Application*, 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	Updated the logos in the headers.
1.6	2020-06-30	Updated the document version based on SDK changes.
1.7	2020-11-09	Updated Figure 3-3 in “Section 3.4 Test and Verification”.
1.8	2020-12-15	Updated GRToolbox UI figures based on software update.

Contents

1 Introduction

The Bluetooth LE throughput example in this document demonstrates how link parameters influence the data throughput in Bluetooth LE connections. These link parameters include Connection Interval, Maximum Transmission Unit (MTU), Data Length, PHY, and TX Power. This example can also verify Bluetooth LE data throughput of GR551x SoCs in different transfer modes (Notify, Write, and Notify & Write).

This document introduces how to use a Bluetooth LE throughput example in a GR551x SDK to verify Bluetooth LE data throughput of GR551x SoCs.

Before you get started, it is recommended to refer to the following documents.

Table 1-1 Reference documents

Name	Description
GR551x Sample Service Application and Customization	Introduces how to apply and customize Goodix Sample Service in developing Bluetooth LE applications based on GR551x SDK.
GR551x Developer Guide	Introduces the software/hardware and quick start guide of GR551x SoCs.
Bluetooth Core Spec v5.1	Offers official Bluetooth standards and core specification (v5.1) from Bluetooth SIG. Available at https://www.bluetooth.com/specifications/bluetooth-core-specification/.
Bluetooth GATT Spec	Provides details about Bluetooth profiles and services. Available at www.bluetooth.com/specifications/gatt.
J-Link/J-Trace User Guide	Provides J-Link operational instructions. Available at www.segger.com/downloads/jlink/UM08001_JLink.pdf.
Keil User Guide	Offers detailed Keil operational instructions. Available at www.keil.com/support/man/docs/uv4/.

2 Profile Overview

2.1 Device Roles

Goodix Throughput Profile defines two device roles:

- Throughput Server: Initiate advertising, wait for a connection request from Throughput Client, receive data from Throughput Client, and send data to Throughput Client using Notify.
- Throughput Client: Send a connection request, establish connections with Throughput Server, send data to Throughput Server using Write without Response, and receive data from Throughput Server.

2.2 Throughput Service

Throughput Service interactions between Throughput Server and Throughput Client are shown in Figure 2-1.

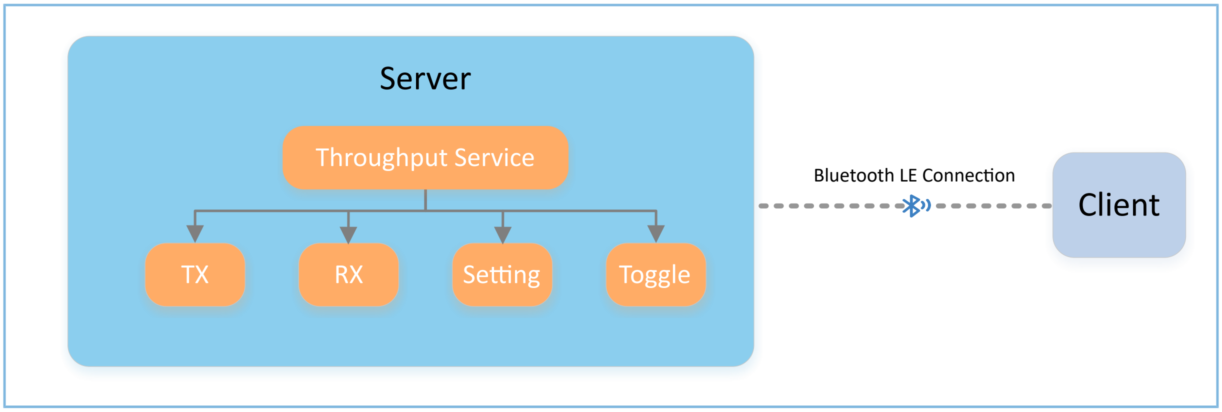


Figure 2-1 Server-client interactions

The Throughput Service (THS) is a type of service customized by Goodix. THS is intended for setting the data transfer rate and parameters in Bluetooth LE THS tests, and the 128-bit vendor-specific UUID of which is A6ED0301-D344-460A-8075-B9E8EC90D71B.

THS characteristics include:

- TX: Send data to THS Client.
- RX: Receive data from THS Client.
- Setting: Receive parameter settings of THS tests, such as MTU, PHY, TX Power, and Connection Interval, and notify setting results to THS Client.
- Toggle: Start/Stop THS tests.

THS characteristics are described in Table 2-1.

Table 2-1 THS characteristics

Characteristic	UUID	Type	Support	Security	Property
TX	A6ED0302-D344-460A-8075-B9E8EC90D71B	128 bits	Mandatory	None	Notify

Characteristic	UUID	Type	Support	Security	Property
RX	A6ED0303-D344-460A-8075-B9E8EC90D71B	128 bits	Mandatory	None	Write without Response
Setting	A6ED0304-D344-460A-8075-B9E8EC90D71B	128 bits	Mandatory	None	Notify, Write without Response
Toggle	A6ED0305-D344-460A-8075-B9E8EC90D71B	128 bits	Mandatory	None	Write without Response

Three transfer modes are available for Bluetooth LE THS tests:

- Unidirectional transfer from THS Server: THS Server sends TX characteristic value to THS Client by means of Notify.
- Unidirectional transfer from THS Client: THS Client sends RX characteristic value to THS Server by means of Write without Response.
- Bidirectional transfer between THS Server and THS Client: Simultaneous data transfers between each other are allowed.

In the three transfer modes mentioned above, THS Server receives Setting characteristic value from THS Client to set parameters for THS tests, and notifies setting results to THS Client.

3 Initial Operation

This chapter introduces how to use a Bluetooth LE throughput example (THS Server and THS Client) in a GR551x SDK.

Note:

SDK_Folder is the root directory of GR551x SDK.

3.1 Preparation

Perform the following tasks before running the Bluetooth LE throughput example.

- **Hardware preparation**

Table 3-1 Hardware preparation

Name	Description
J-Link debug probe	JTAG emulator launched by SEGGER. For more information, visit www.segger.com/products/debug-probes/j-link/.
Development board	GR5515 Starter Kit Board (GR5515 SK Board) (2 boards in total)
Cable	Micro USB 2.0 cable
Android Phone	A mobile phone running on Android 4.4 (KitKat) or later versions

- **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/.
Keil MDK5	An integrated development environment (IDE). Available at www.keil.com/download/product/.
GRToolbox (Android)	A Bluetooth LE debugging tool for GR551x. Available in `SDK_Folder\tools\GRToolbox`.
GProgrammer (Windows)	A GR551x programming tool. Available in `SDK_Folder\tools\GProgrammer` .

3.2 Hardware Connection

Connect a GR5515 Starter Kit Board to a PC with a Micro USB 2.0 cable.

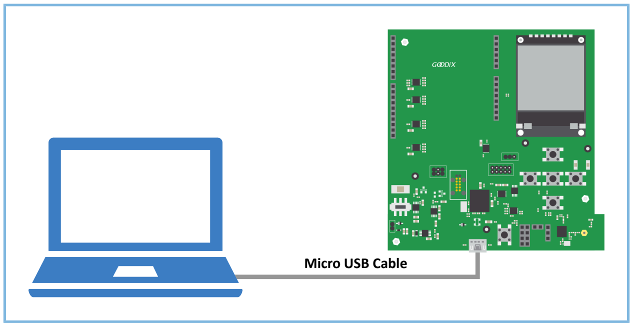


Figure 3-1 Hardware connection

3.3 Firmware Download

Users need to download *ble_app_throughput_fw.bin* firmware to GR5515 SK Board A which serves as THS Server. Download *ble_app_throughput_c_fw.bin* firmware to GR5515 SK Board B which serves as THS Client.

For details on downloading firmware to the GR5515 SK Boards, see *GProgrammer User Manual*.

Note:

- The *ble_app_throughput_fw.bin* is in `SDK_Folder\projects\ble\ble_peripheral\ble_app_throughput\build`.
- The *ble_app_throughput_c_fw.bin* is in `SDK_Folder\projects\ble\ble_central\ble_app_throughput_c\build`.

  SDK_Folder is the root directory of GR551x SDK.

3.4 Test and Verification

In this document, two scenarios are involved in Bluetooth LE THS tests:

- Scenario 1: Perform a THS test between GR5515 SK Board A and a mobile phone.
- Scenario 2: Perform a THS test between GR5515 SK Board A and B.

3.4.1 Test Between GR5515 SK Board A and a Mobile Phone

Perform Bluetooth LE THS test and verification between THS Server (GR5515 SK Board A) and THS Client (an Android phone), as shown in Figure 3-2.

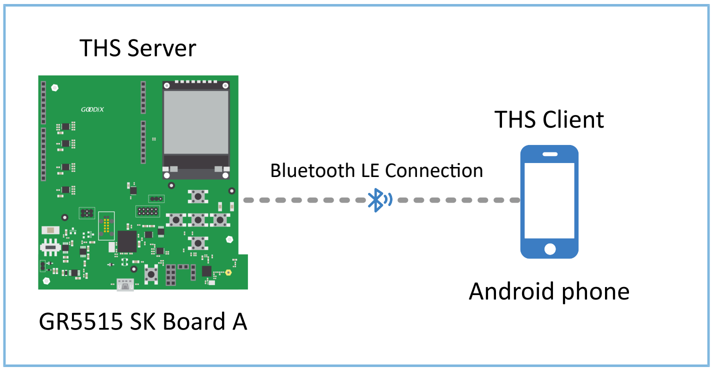


Figure 3-2 Test between GR5515 SK Board A and a mobile phone

Steps for the test between GR5515 SK Board A and the mobile phone are described below:

1. Enable the Bluetooth.

   Enable the Bluetooth of the mobile phone and power on GR5515 SK Board A.

2. Scan Goodix THS devices.

   Run GRToolbox on the mobile phone and tap **Application** > **THS**.

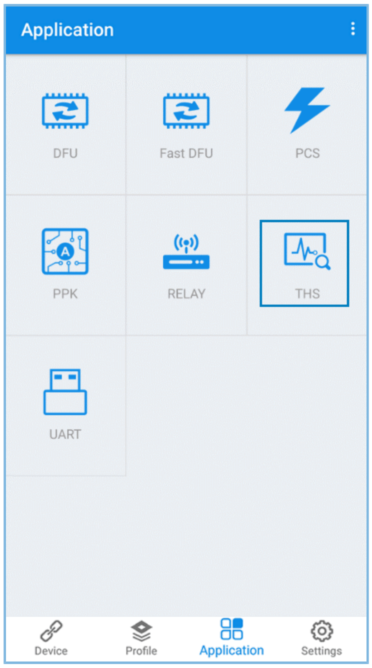
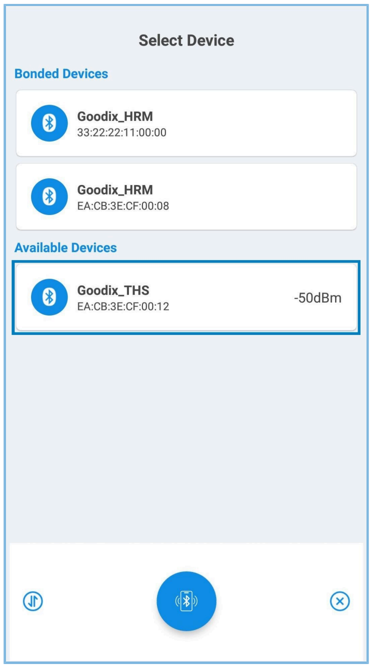


Figure 3-3 Choosing **THS**

   Start scanning. A GR5515 SK Board with the advertising name **Goodix_THS** is discovered, as shown in Figure 3-4.


Select Device

Bonded Devices

Goodix_HRM
33:22:22:11:00:00

Goodix_HRM
EA:CB:3E:CF:00:08

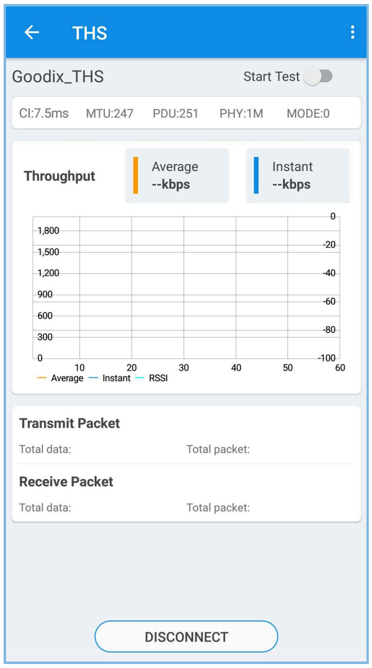
Available Devices

Goodix_THS
EA:CB:3E:CF:00:12
-50dBm

Figure 3-4 Discovering **Goodix_THS**

3. Connect **Goodix_THS**.

   Tap and connect **Goodix_THS** to enter the THS test interface, as shown in Figure 3-5.



THS

Goodix_THS Start Test

CI:7.5ms MTU:247 PDU:251 PHY:1M MODE:0

Throughput

Average
--kbps

Instant
--kbps

Transmit Packet

Total data: Total packet:

Receive Packet

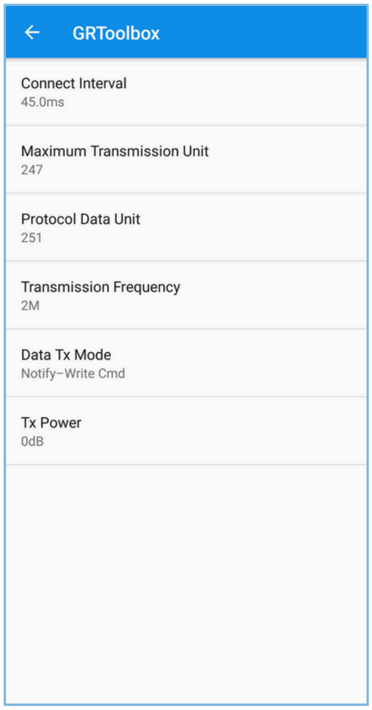
Total data: Total packet:

DISCONNECT

Figure 3-5 THS test interface

4. Set test parameters.

   Tap ⋮ in the upper-right corner of the THS test interface, to enter the test parameter setting interface and configure parameters, as shown in Figure 3-6.



GRToolbox

Connect Interval
45.0ms

Maximum Transmission Unit
247

Protocol Data Unit
251

Transmission Frequency
2M

Data Tx Mode
Notify–Write Cmd

Tx Power
0dB

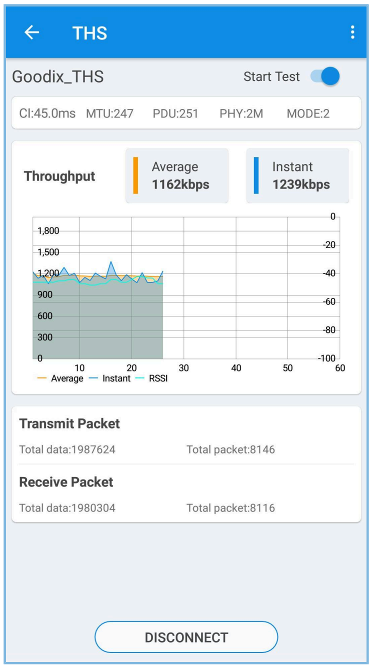
Figure 3-6 Test parameter setting interface

Note:

Users need to set the MTU value on GRToolbox before connecting the mobile phone to the GR551x SK Board. This is because an MTU exchange takes place immediately once the mobile phone is connected to the GR5515 SK Board, and only one MTU exchange takes place during one connection.

5. Start the THS test.

   Tap to start the test. The average value and instant value of Bluetooth LE data throughputs between the mobile phone and the GR5515 SK Board are displayed in a statistical graph, as shown in Figure 3-7.



THS

Goodix_THS Start Test

CI:45.0ms MTU:247 PDU:251 PHY:2M MODE:2

Throughput

Average 1162kbps

Instant 1239kbps

Transmit Packet

Total data:1987624 Total packet:8146

Receive Packet

Total data:1980304 Total packet:8116

DISCONNECT

Figure 3-7 Starting the THS test

3.4.2 Test Between THS Server and THS Client

Perform Bluetooth LE THS test and verification between THS Server (GR5515 SK Board A) and THS Client (GR5515 SK Board B), as shown in Figure 3-8.

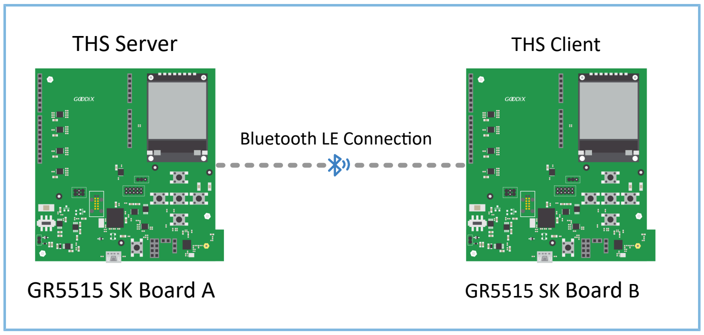


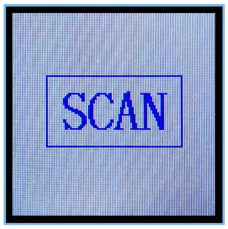
Figure 3-8 Test between THS Server and THS Client

Steps for Bluetooth LE THS test and verification are described below:

1. Power on Board A and Board B.

   Enable advertising after THS Server (Board A) is powered on.

   Wait for scanning THS Server after THS Client (Board B) is powered on. The state is displayed on the screen of Board B, as shown in Figure 3-9.

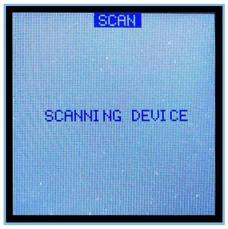


SCAN

Figure 3-9 Board B: waiting for scanning

2. Start scanning and establish connection with Board A.

   Press **OK** on THS Client (Board B). Board B starts scanning (Figure 3-10). After THS Server (Board A) is discovered, Board B sends a connection request to Board A automatically.

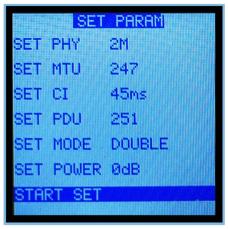


SCAN

SCANNING DEVICE

Figure 3-10 Board B: scanning device

   When THS Client (Board B) is connected to THS Server (Board A) successfully, the parameter setting interface of Board B automatically displays, as shown in Figure 3-11.

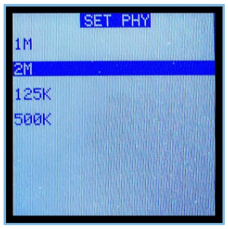


```
   SET PARAM
SET PHY    2M
SET MTU    247
SET CI     45ms
SET PDU    251
SET MODE   DOUBLE
SET POWER 0dB
START SET
```

Figure 3-11 Parameter setting interface on Board B

3. Set THS parameters.

   Press **UP**/**DOWN** on THS Client (Board B) to select the parameters to be set (such as **SET PHY**). Press **OK** to enter the setting interface. The **SET PHY** interface is shown in Figure 3-12.



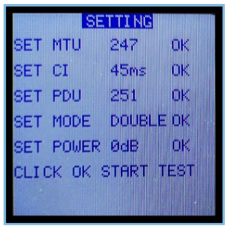
```
SET PHY
1M
2M
125K
500K
```

Figure 3-12 **SET PHY** interface

In the **SET PHY** interface, select a PHY parameter on demand, and press **OK**.

4. Start the THS test.

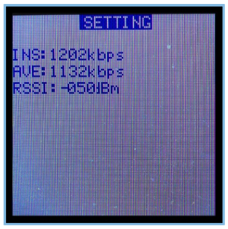
   After all THS parameters are set, select **START TEST** and press **OK**, as shown in Figure 3-13.



```
SETTING
SET MTU   247    OK
SET CI    45ms   OK
SET PDU   251    OK
SET MODE  DOUBLE OK
SET POWER 0dB    OK
CLICK OK START TEST
```

Figure 3-13 **SETTING** interface

The THS test results are displayed on the screen. An example is shown in Figure 3-14.



```
SETTING
INS:1202kbps
AVE:1132kbps
RSSI:-050dBm
```

Figure 3-14 THS test results

Parameter descriptions of the THS test results are shown in Table 3-3:

Table 3-3 Parameter description of THS test results

Parameter	Description
INS	Instant Throughput
AVE	Average Throughput
RSSI	Received Signal Strength Indication

To update test parameters during the test, press **LEFT** to go back to the parameter setting interface, and reset parameters.

📖 **Note**:

THS Client (Board B) supports parameter setting and real-time output of test results on GRUart. Users can also press buttons on Board B to set parameters and view test results on the display, as mentioned in "Section 3.4.2 Test Between THS Server and THS Client". Users can select either of the two approaches. For more information about formats of serial port commands, see "Chapter 5 Set THS Parameters on GRUart".

4 Application Details

This chapter introduces the project directory and implementation procedures of the Bluetooth LE THS example (including THS Server and THS Client).

4.1 Project Directory of Bluetooth LE THS Server

The source code and project file of the Bluetooth LE THS Server example are in `SDK_Folder\projects\ble\ble_peripheral\ble_app_throughput`, and project file is in the Keil_5 folder.

Double-click the project file, *ble_app_throughput.uvprojx*, to view the ble_app_throughput project directory structure of the Bluetooth LE THS Server example in Keil. For related files, see Table 4-1.

Table 4-1 File description of ble_app_throughput

Group	File	Description
gr_profiles	ble_prf_utils.c	This file contains profile-related operational tools.
	ths.c	This file implements Throughput Service.
	ota.s	This file implements OTA Service.
user_callback	user_gap_callback.c	This file implements GAP callback, such as connection, disconnection, and GAP parameter update.
	user_gatt_common_callback.c	This file implements GATT common callback, such as MTU exchange.
	user_sm_callback.c	This file implements SM callback, such as pairing and bonding.
user_platform	user_periph_setup.c	This file configures App logs, device address, and power management mode.
user_app	main.c	This file contains the main() function.
	user_app.c	This file implements profile registration and logical processing for THS Server applications.
	throughput.c	This file handles THS events.

4.2 Project Directory of Bluetooth LE THS Client

The source code and project file of the Bluetooth LE THS Client example are in `SDK_Folder\projects\ble\ble_central\ble_app_throughput_c`, and project file is in the Keil_5 folder.

Double-click the project file, *ble_app_throughput_c.uvprojx*, to view the ble_app_throughput_c project directory structure of the Bluetooth LE THS Client example in Keil. For related files, see Table 4-2.

Table 4-2 File description of ble_app_throughput_c

Group	File	Description
gr_profiles	ble_prf_utils.c	This file contains profile-related operational tools.
	ths_c.c	This file implements Throughput Service Client Profile.
user_callback	user_gap_callback.c	This file implements GAP callback, such as connection, disconnection, and GAP parameter update.

Group	File	Description
	user_gatt_common_callback.c	This file implements GATT common callback, such as MTU exchange.
user_platform	user_periph_setup.c	This file configures device serial port, device address, and device buttons.
	user_interrupt.c	This file contains the serial port interrupt handler.
user_app	main.c	This file contains the main() function.
	user_app.c	This file implements profile registration and logical processing for THS Client applications.
	throughput_c.c	This file processes Throughput Service Client events and provides throughput statistics.
	user_gui.c	This file implements THS Client GUI.

4.3 Implementation Procedures

Interactions between THS Server and THS Client are shown in Figure 4-1:

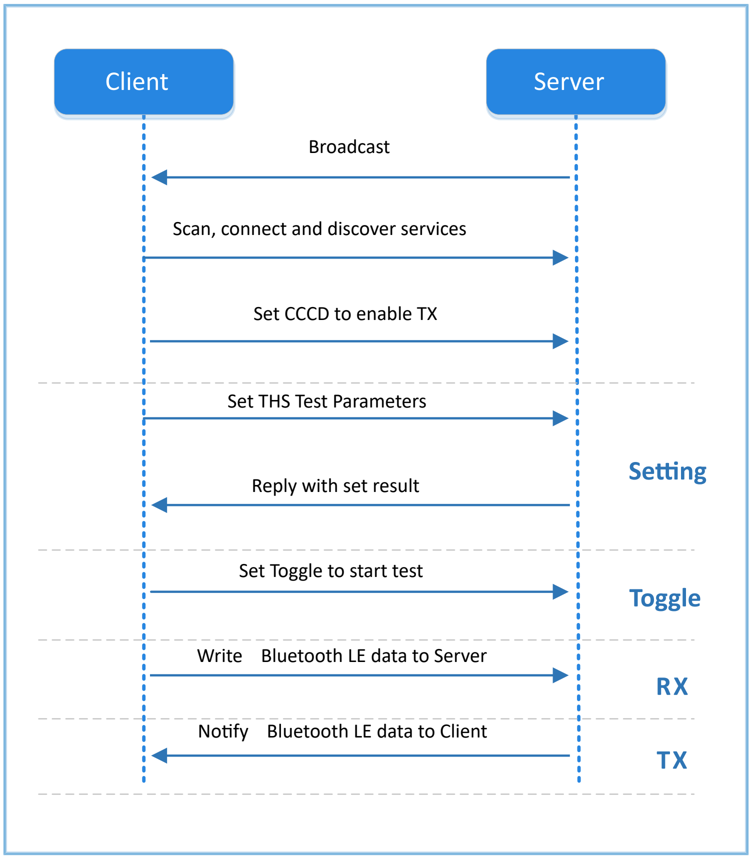


Figure 4-1 Implementation procedures

5 Set THS Parameters on GRUart

During the Bluetooth LE THS test, THS Client receives commands input on GRUart to test related parameters. All input commands should end with a newline ('\r\n'). The specific command formats are defined as follows.

5.1 Scan Device

Table 5-1 Scanning command

Command	SCAN
Description	None
Note	None
Example	SCAN
Response	Results from GRUart: • Start scan device. • Connected. • Throughput Service discovery completely. • Enabled TX Notification. • Enabled SETTING Notification.

5.2 Update Connection Parameters

Table 5-2 Connection parameter update command

Command	CI:<conn_interval_min>:<conn_interval_max>:<latency>:<timeout>
Description	<conn_interval_min>: minimum value of connection interval (unit: 1.25 ms) <conn_interval_max>: maximum value of connection interval (unit: 1.25 ms) <latency>: connection latency <timeout>: connection timeout (unit: 10 ms)
Note	If a fixed connection interval is required, set the maximum and the minimum connection intervals to the same value. For tests on optimal throughput, set the latency value to 0. Timeout > (1 + Latency) x conn_interval x 2
Example	CI:12:12:0:100
Response	Setting results from GRUart

5.3 Set MTU

Table 5-3 MTU setting command

Command	MTU:<mtu_value>

Description	<mtu_value>: MTU; value range: 23 to 247
Note	The MTU value is exchanged for once only during one connection.
Example	MTU: 247
Response	Setting results from GRUart

5.4 Set PDU

Table 5-4 PDU setting command

Command	PDU:<payload_octets>:<time>
Description	<payload_octets>: payload octets < time >: time for TX
Note	None
Example	PDU:251:2120
Response	Setting results from GRUart

5.5 Set PHY

Table 5-5 PHY setting command

Command	PHY:<tx_phy>:<rx_phy>:<phy_opt>
Description	<tx_phy>: Preferred transmit PHYs • 1: 1M PHY • 2: 2M PHY • 4: Coded PHY <rx_phy>: Preferred receive PHYs • 1: 1M PHY • 2: 2M PHY • 4: Coded PHY < phy_opt >: Options for PHY • 0: Host has no preferred coding when transmitting on the LE Coded PHY. • 1: Host prefers that S=2 coding be used when transmitting on the LE Coded PHY. • 2: Host prefers that S=8 coding be used when transmitting on the LE Coded PHY.
Note	None
Example	PHY:1:1:0
Response	Setting results from GRUart

5.6 Test Mode

Table 5-6 Test mode command

Command	TRANS_MODE:<mode>
Description	<mode>: test mode 0: Send data from THS Server only (Notify). 1: Send data from THS Client only (Write). 2: Send data from both THS Server and THS Client.
Note	None
Example	TRANS_MODE: 2
Response	Setting results from GRUart

5.7 Start/Stop Test

Table 5-7 Test start/stop command

Command	TOGGLE_SET:<start_or_stop>
Description	< start_or_stop >: Start or stop the test. 0: Stop the test. 1: Start the test.
Note	None
Example	TOGGLE_SET: 1
Response	Setting results from GRUart