GR5xx Throughput Example Application

Version: 3.1

Release Date: 2023-11-06

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: Floor 12-13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

TEL: +86-755-33338828 Zip Code: 518000

Website: www.goodix.com

Preface

Purpose

This document introduces how to use and verify a throughput example in the Bluetooth Low Energy (Bluetooth LE) GR5xx Software Development Kit (SDK), to help users quickly get started with secondary development.

Audience

This document is intended for:

- Device user
- Developer
- Test engineer
- Hobbyist developer
- Technical writer

Release Notes

This document is the third release of *GR5xx Throughput Example Application*, corresponding to Bluetooth LE GR5xx System-on-Chip (SoC) series.

Revision History

Version	Date	Description
1.0	2023-01-10	Initial release
3.0	2023-03-30	Updated descriptions about GR5xx SoCs.
3.1	2023-11-06	Updated the approaches for obtaining GProgrammer, GRToolbox, and GRUart.

Contents

1 Introduction

The GR5xx throughput example in this document demonstrates how link parameters influence the data throughput in Bluetooth Low Energy (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 GR5xx System-on-Chips (SoCs) in different transfer modes (Notify, Write, and Notify & Write).

This document introduces how to use a throughput example in the GR5xx Software Development Kit (SDK) to verify Bluetooth LE data throughput of GR5xx SoCs.

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

Table 1-1 Reference documents

Name	Description
GR5xx Sample Service Application and Customization	Introduces how to apply and customize Goodix Sample Service in developing Bluetooth LE applications based on GR5xx SDK.
Developer guide of the specific GR5xx SoC	Introduces GR5xx SDK and how to develop and debug applications based on the SDK.
Bluetooth Core Spec	Offers official Bluetooth standards and core specification from Bluetooth SIG.
Bluetooth GATT Spec	Provides details about Bluetooth profiles and services. Available at https://www.bluetooth.com/specifications/gatt/.
J-Link/J-Trace User Guide	Provides J-Link operational instructions. Available at https://www.segger.com/downloads/jlink/UM08001_JLink.pdf.
Keil User Guide	Offers detailed Keil operational instructions. Available at https://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.

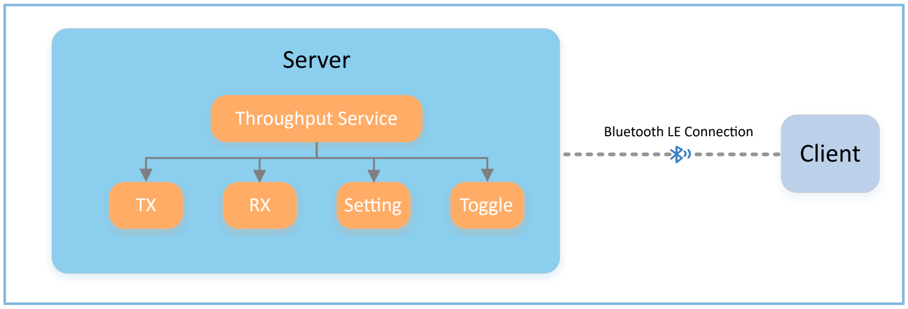


Figure 2-1 Server-client interactions

The Throughput Service (THS) is a type of service customized by Goodix. THS is intended for setting the data transfer rate and parameters in 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, Connection Interval, and Data Length, 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 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 throughput example (THS Server and THS Client) in the GR5xx SDK.

Note:

SDK_Folder is the root directory of the GR5xx SDK in use.

3.1 Preparation

Perform the following tasks before running the GR5xx throughout example.

- **Hardware preparation**

Table 3-1 Hardware preparation

Name	Description
Development board	Two Starter Kit Boards (SK Boards) of the corresponding SoC
Connection cable	USB Type C cable (Micro USB 2.0 cable for GR551x SoCs)
Android phone	A mobile phone running on Android 5.0 (KitKat) and later

- **Software preparation**

Table 3-2 Software preparation

Name	Description
Windows	Windows 7/Windows 10
J-Link driver	A J-Link driver. Available at https://www.segger.com/downloads/jlink/.
Keil MDK5	An integrated development environment (IDE). MDK-ARM Version 5.20 or later is required. Available at https://www.keil.com/download/product/.
GRToolbox (Android)	A Bluetooth LE debugging tool. Available at https://www.goodix.com/en/software_tool/grtoolbox.
GProgrammer (Windows)	A programming tool. Available at https://www.goodix.com/en/software_tool/gprogrammer_ble.
GRUart (Windows)	A serial port debugging tool. Available at https://www.goodix.com/en/download?objectId=43&objectType=software.

3.2 Firmware Programming

The source code of the GR5xx throughput example is in:

- `SDK_Folder\projects\ble\ble_peripheral\ble_app_throughput` (for THS Server)
- `SDK_Folder\projects\ble\ble_central\ble_app_throughput_c` (for THS Client)

You can download *ble_app_throughput.bin* and *ble_app_throughput_c.bin* to the SK Board A (serving as THS Server) and SK Board B (serving as THS Client) respectively through GProgrammer.

For details, see *GProgrammer User Manual*.

Note:

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

3.3 Test and Verification

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

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

3.3.1 Test Between SK Board A and a Mobile Phone

This section introduces how to perform GR5xx THS test and verification between THS Server (SK Board A) and THS Client (an Android phone).

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

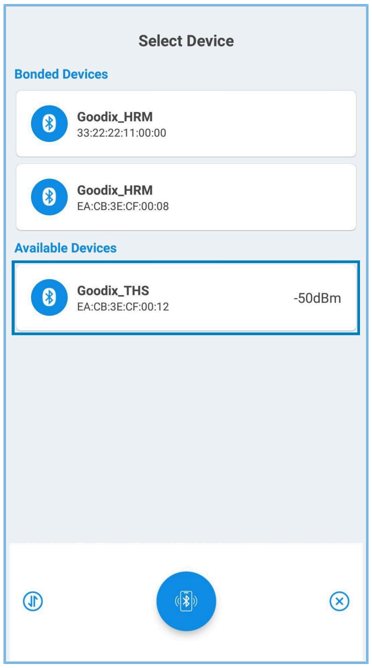
1. Enable the Bluetooth.

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

2. Scan Goodix THS devices.

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

   Start scanning. An SK Board with the advertising name **Goodix_THS** is discovered, as shown in Figure 3-1.



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

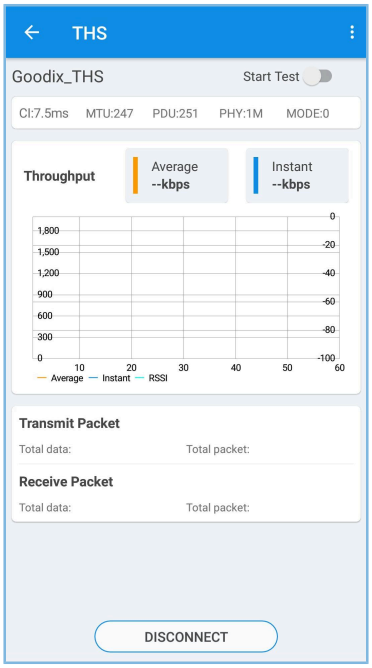
Figure 3-1 Discovering **Goodix_THS**

Note:

Screenshots of GRToolbox in this document are for reference only, to help users better understand the software operation. In the case of interface differences due to version changes, the interface of GRToolbox in practice shall prevail.

3. Connect **Goodix_THS**.

   Tap and connect **Goodix_THS** to enter the THS test interface, as shown in Figure 3-2.



Goodix_THS

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

Average --kbps

Instant --kbps

Transmit Packet

Total data: Total packet:

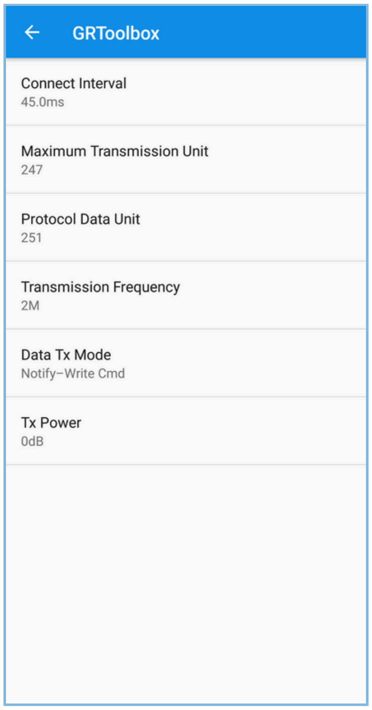
Receive Packet

Total data: Total packet:

Figure 3-2 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 3-3.



Connect Interval
45.0ms

Maximum Transmission Unit
247

Protocol Data Unit
251

Transmission Frequency
2M

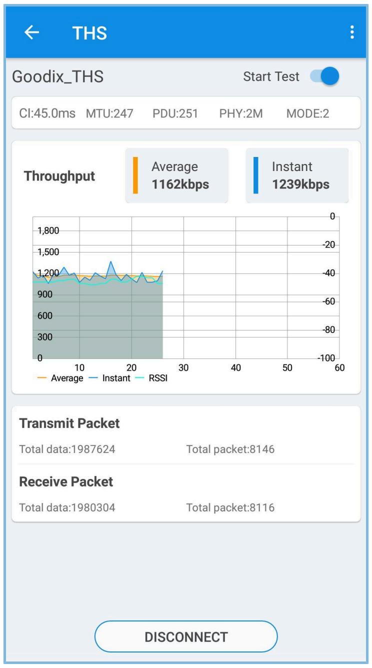
Data Tx Mode
Notify−Write Cmd

Tx Power
0dB

Figure 3-3 Test parameter setting interface

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 SK Board are displayed in a statistical graph, as shown in Figure 3-4.



Goodix_THS

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

Figure 3-4 Starting the THS test

3.3.2 Test Between THS Server and THS Client

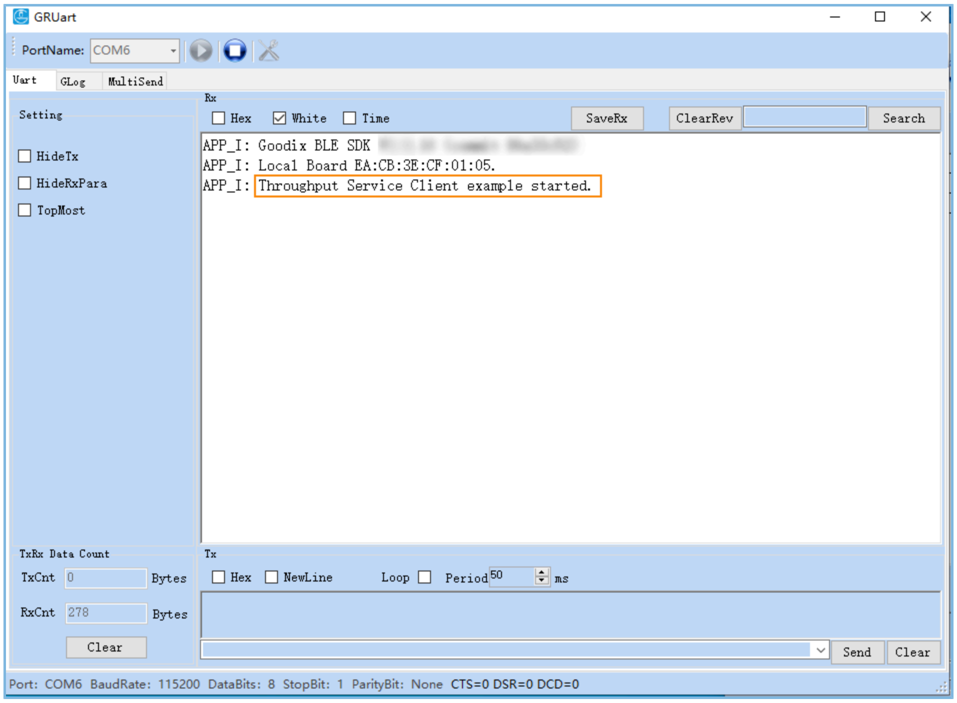
This section introduces GR5xx THS test and verification between THS Server (SK Board A) and THS Client (SK Board B).

Steps for GR5xx THS test and verification are described below:

1. Power on SK Board A and SK Board B.

   Power on THS Server (SK Board A) to start advertising.

   Power on THS Client (SK Board B) to get ready for scanning THS Server (SK Board A). At this time, "Throughput Service Client example started." is printed on GRUart, as shown in Figure 3-5.



PortName: COM6

☐ HideTx
☐ HideRxPara
☐ TopMost

☐ Hex ☑ White ☐ Time

```
APP_I: Goodix BLE SDK [redacted]
APP_I: Local Board EA:CB:3E:CF:01:05.
APP_I: Throughput Service Client example started.
```

TxCnt 0 Bytes

RxCnt 278 Bytes

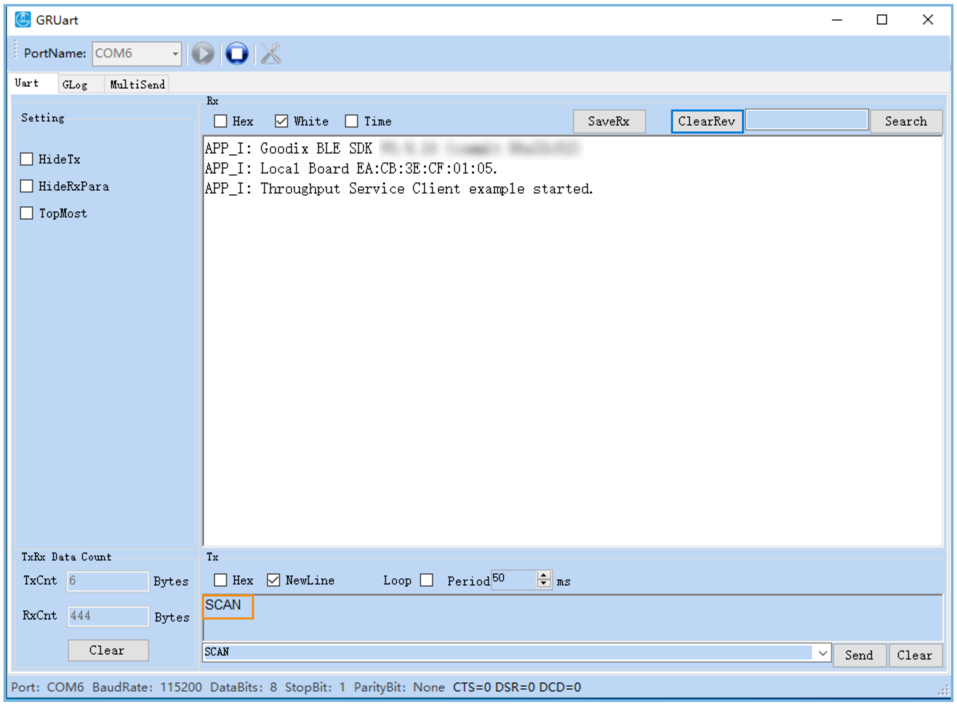
☐ Hex ☐ NewLine Loop ☐ Period 50 ms

Port: COM6 BaudRate: 115200 DataBits: 8 StopBit: 1 ParityBit: None CTS=0 DSR=0 DCD=0

Figure 3-5 GRUart: getting ready for scanning

2. Start scanning and establish connection with SK Board A.

   After the SCAN command is sent via GRUart, SK Board B starts scanning (Figure 3-6). After THS Server (SK Board A) is discovered, SK Board B sends a connection request to SK Board A automatically.



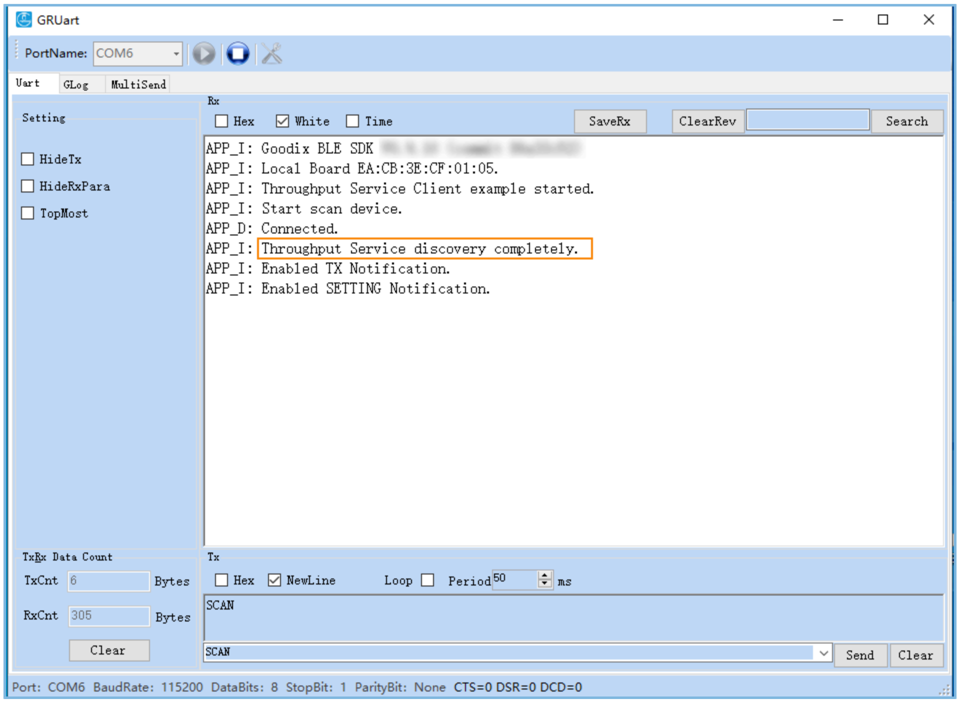
☐ HideTx
☐ HideRxPara
☐ TopMost
☐ Hex ☑ White ☐ Time

```
APP_I: Goodix BLE SDK [redacted]
APP_I: Local Board EA:CB:3E:CF:01:05.
APP_I: Throughput Service Client example started.
```

TxCnt 6 Bytes
RxCnt 444 Bytes
☐ Hex ☑ NewLine Loop ☐ Period 50 ms
SCAN
SCAN
Port: COM6 BaudRate: 115200 DataBits: 8 StopBit: 1 ParityBit: None CTS=0 DSR=0 DCD=0

Figure 3-6 GRUart: scanning device

When THS Client (SK Board B) is connected to THS Server (SK Board A), "Throughput Service discovery completely." is printed on GRUart, and setting of all THS parameter commands starts, as shown in Figure 3-7; for command format, refer to "Chapter 5 Setting THS Parameters on GRUart".



☐ HideTx
☐ HideRxPara
☐ TopMost
☐ Hex ☑ White ☐ Time

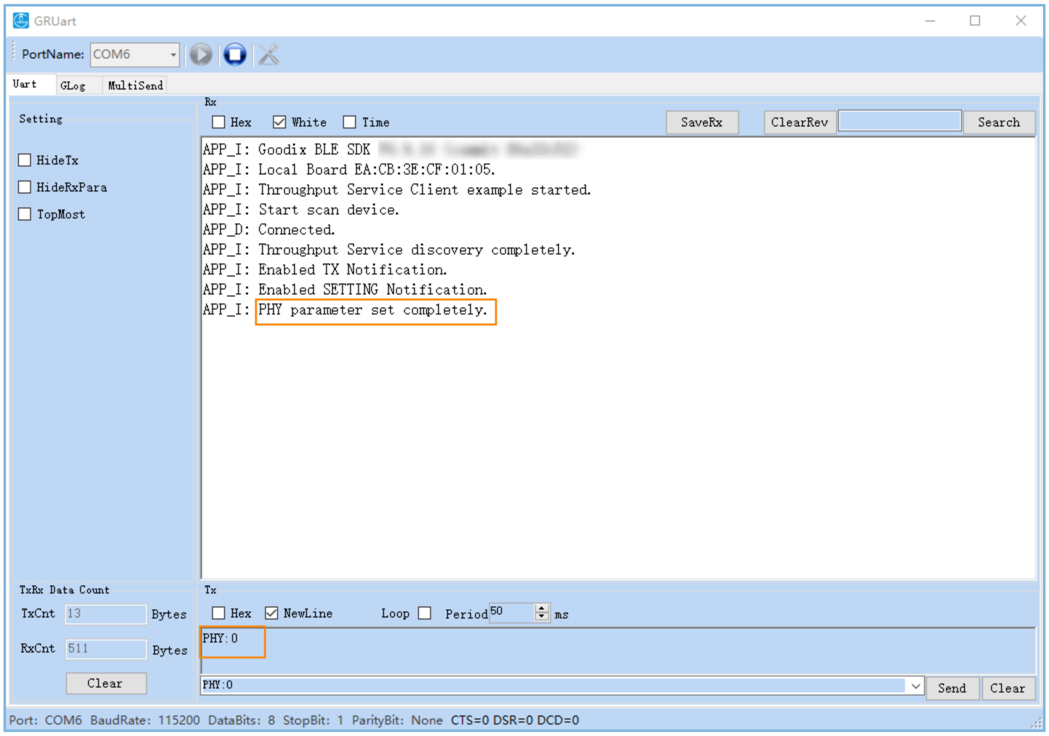
```
APP_I: Goodix BLE SDK [redacted]
APP_I: Local Board EA:CB:3E:CF:01:05.
APP_I: Throughput Service Client example started.
APP_I: Start scan device.
APP_D: Connected.
APP_I: Throughput Service discovery completely.
APP_I: Enabled TX Notification.
APP_I: Enabled SETTING Notification.
```

TxCnt 6 Bytes
RxCnt 305 Bytes
☐ Hex ☑ NewLine Loop ☐ Period 50 ms
SCAN
SCAN
Port: COM6 BaudRate: 115200 DataBits: 8 StopBit: 1 ParityBit: None CTS=0 DSR=0 DCD=0

Figure 3-7 Successful connection interface on GRUart

3. Set THS parameters.

   Input a corresponding THS parameter setting command (for example, PHY：0) into GRUart and send the command. The PHY setting interface is shown in Figure 3-8.



□ HideTx
□ HideRxPara
□ TopMost

□ Hex ☑ White □ Time

```
APP_I: Goodix BLE SDK [redacted]
APP_I: Local Board EA:CB:3E:CF:01:05.
APP_I: Throughput Service Client example started.
APP_I: Start scan device.
APP_D: Connected.
APP_I: Throughput Service discovery completely.
APP_I: Enabled TX Notification.
APP_I: Enabled SETTING Notification.
APP_I: PHY parameter set completely.
```

TxCnt 13 Bytes

RxCnt 511 Bytes

□ Hex ☑ NewLine Loop □ Period 50 ms

PHY：0

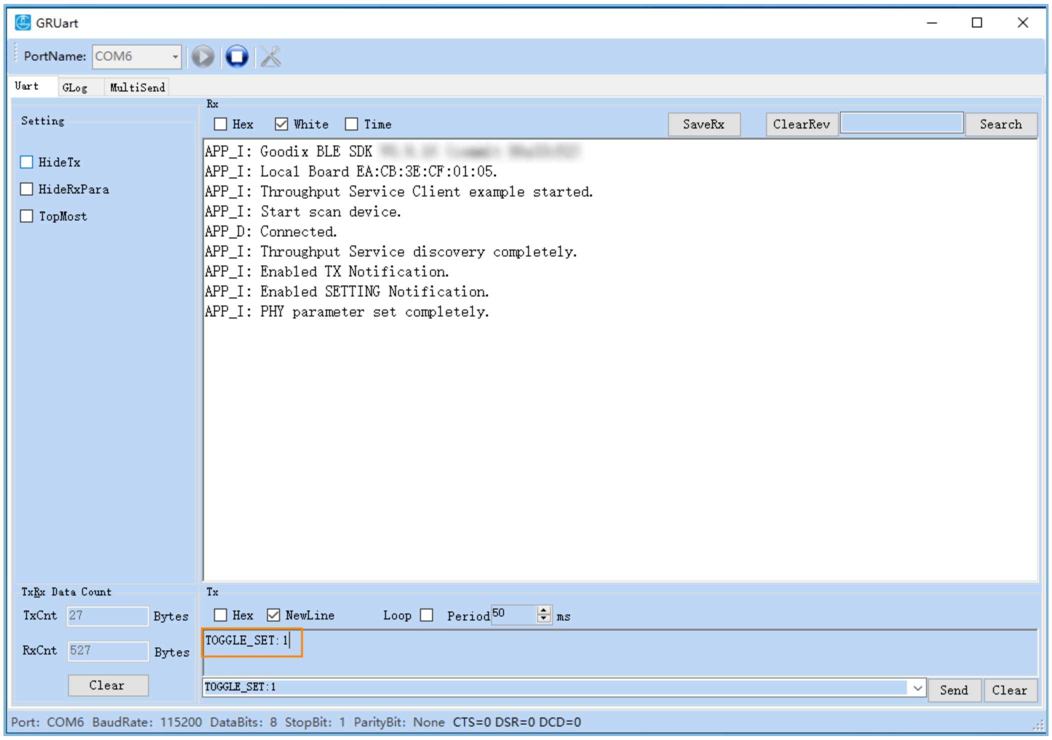
PHY:0

Port: COM6 BaudRate: 115200 DataBits: 8 StopBit: 1 ParityBit: None CTS=0 DSR=0 DCD=0

Figure 3-8 PHY setting interface

4. Start the THS test.

   After all THS parameters are set, send the TOGGLE_SET：1 command via GRUart to start testing, as shown in Figure 3-9.



□ Hex ☑ White □ Time

□ HideTx
□ HideRxPara
□ TopMost

```
APP_I: Goodix BLE SDK [redacted]
APP_I: Local Board EA:CB:3E:CF:01:05.
APP_I: Throughput Service Client example started.
APP_I: Start scan device.
APP_D: Connected.
APP_I: Throughput Service discovery completely.
APP_I: Enabled TX Notification.
APP_I: Enabled SETTING Notification.
APP_I: PHY parameter set completely.
```

TxRx Data Count
TxCnt 27 Bytes
RxCnt 527 Bytes
Clear

□ Hex ☑ NewLine Loop □ Period 50 ms

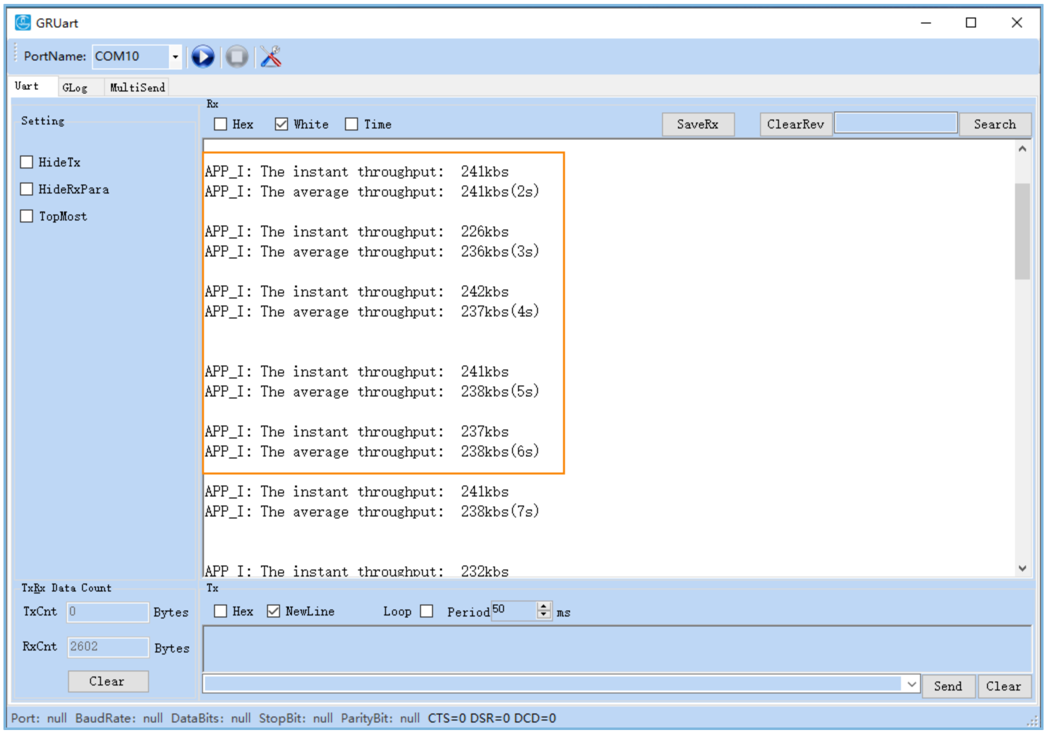
```
TOGGLE_SET: 1
```

TOGGLE_SET:1

Port: COM6 BaudRate: 115200 DataBits: 8 StopBit: 1 ParityBit: None CTS=0 DSR=0 DCD=0

Figure 3-9 Test start interface

The THS test results of SK Board A are displayed on GRUart. An example is shown in Figure 3-10.



□ Hex ☑ White □ Time

□ HideTx
□ HideRxPara
□ TopMost

```
APP_I: The instant throughput:  241kbs
APP_I: The average throughput:  241kbs(2s)

APP_I: The instant throughput:  226kbs
APP_I: The average throughput:  236kbs(3s)

APP_I: The instant throughput:  242kbs
APP_I: The average throughput:  237kbs(4s)


APP_I: The instant throughput:  241kbs
APP_I: The average throughput:  238kbs(5s)

APP_I: The instant throughput:  237kbs
APP_I: The average throughput:  238kbs(6s)

APP_I: The instant throughput:  241kbs
APP_I: The average throughput:  238kbs(7s)

APP_I: The instant throughput:  232kbs
```

TxRx Data Count
TxCnt 0 Bytes
RxCnt 2602 Bytes
Clear

□ Hex ☑ NewLine Loop □ Period 50 ms

Port: null BaudRate: null DataBits: null StopBit: null ParityBit: null CTS=0 DSR=0 DCD=0

Figure 3-10 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
instant throughput	Instant throughput
average throughput	Average throughput

To update test parameters during the test, send the `TOGGLE_SET:0` command via GRUart to pause the test and then re-send a parameter setting command.

4 Application Details

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

4.1 Project Directory of THS Server

The source code and project file of the 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 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	ths.c	Implements Throughput Service.
user_platform	user_periph_setup.c	Configures App logs, device address, and power management mode.
user_app	main.c	Contains the main() function.
	user_app.c	Implements profile registration and logical processing for THS Server applications.
	throughput.c	Handles THS events.

4.2 Project Directory of THS Client

The source code and project file of the 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 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	ths_c.c	Implements Throughput Service Client Profile.
user_platform	user_periph_setup.c	Configures device serial port, device address, and device buttons.
	user_interrupt.c	Contains the serial port interrupt handler.
user_app	main.c	Contains the main() function.
	user_app.c	Implements profile registration and logical processing for THS Client applications.
	throughput_c.c	Processes Throughput Service Client events and provides throughput statistics.

4.3 Running Procedures

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

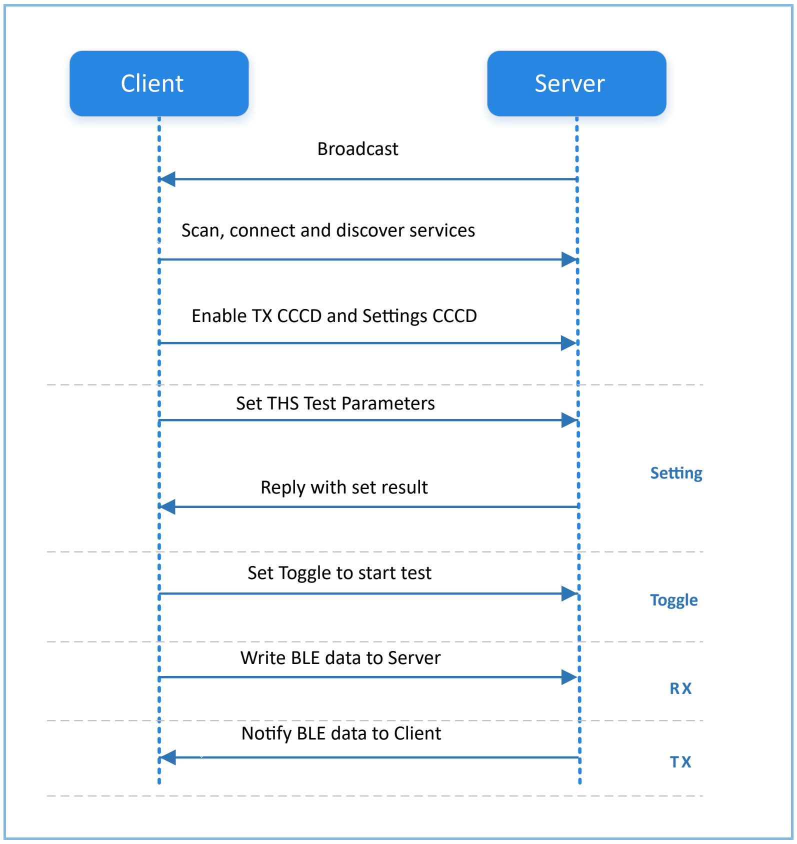


Figure 4-1 Running procedures

5 Setting THS Parameters on GRUart

During the GR5xx 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 Scanning Device

Table 5-1 Scanning command

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

5.2 Updating 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 Setting MTU

Table 5-3 MTU setting command

Command	MTU:<mtu_value>

Description	<mtu_value>: MTU; value range: 23 to 512
Example	MTU: 247
Response	Setting results from GRUart

5.4 Setting 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 Setting 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 Setting Test Mode

Table 5-6 Test mode setting 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 Setting TX Power

Table 5-7 TX Power setting command

Command	TX_PWR:<tx_power_value>
Description	<tx_power_value>: TX power value; options: "-20", "-4", "0", "2", "4", "7"
Note	None
Example	TX_PWR:2
Response	Setting results from GRUart

5.8 Starting/Stopping Test

Table 5-8 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