

How to Test Bluetooth

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

Contents

1	Background	5
2	Bluetooth SIG Qualification	6
2.1	BQTF (Bluetooth Qualification Test Facility)	6
2.2	Bluetooth Qualification Process	6
2.3	Testing Profiles with PTS.....	8
2.3.1	PTS Radio Module	8
2.3.2	Profile Tuning Suite	8
2.4	Testing Controller with equipments	8
2.4.1	Validated Test Systems.....	8
2.4.2	Other Test Equipments	9
2.4.3	Configuration on DUT.....	9
2.5	Test Specification/ Test Suite.....	9
2.5.1	RF.....	9
2.5.2	(BLE) RF-PHY.....	10
2.5.3	(BLE) LL.....	10
2.5.4	BB	10
2.5.5	LMP	11
2.5.6	HCI.....	11
2.5.7	Profiles/Protocol.....	11
3	BR/EDR Signaling Test	12
3.1	Method #1: Config with QRCT	12
3.1.1	Configuration on DUT.....	12
3.1.1.1	Android + Bluedroid: WCN36x0 or QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X.....	12
3.1.1.2	MDM + Linux + Bluez: QCA9377/QCA61x4	12
3.1.1.3	WOS	12
3.1.2	Configuration on PC	13
3.1.2.1	WOS	15
3.1.3	Configuration on Equipment	15
3.2	Method #2: Config with btconfig or hcitool	15
3.2.1.1	Android+Bluedroid: WCN36x0.....	15

3.2.1.2	Android+Bluedroid:	
QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X	16
3.2.1.3	MDM + Linux + Bluez	16
4	BR/EDR Non-Signaling Test	17
4.1	Method #1: Config with QRCT	17
4.1.1	Configuration on DUT	17
4.1.1.1	Android + Bluedroid: WCN36x0 or QCA61x4/ WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X	17
4.1.1.2	MDM + Linux + Bluez: QCA9377/QCA61x4	17
4.1.1.3	WOS	17
4.1.2	Configuration on PC	18
4.1.2.1	WOS	19
4.1.3	Configuration on Equipment	20
4.2	Method #2: Config with btconfig or hcitool	20
4.2.1.1	Android + Bluedroid: WCN36x0	20
4.2.1.2	Android + Bluedroid: QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X	20
4.2.1.3	MDM + Linux + Bluez: QCA9377/QCA61x4	20
4.2.1.4	HCI Commands for BR/EDR non-signaling tests	20
5	BLE Non-Signaling Test	24
5.1	Method #1: Config with QRCT	24
5.1.1	Configuration on DUT	24
5.1.1.1	Android+Bluedroid: WCN36x0 or QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X	24
5.1.1.2	MDM + Linux + Bluez: QCA9377/QCA61x4	24
5.1.1.3	WOS	24
5.1.2	Configuration on PC	25
5.1.2.1	WOS	25
5.1.3	Configuration on Equipment	26
5.2	Method #2: Config with btconfig or hcitool	26
5.2.1	Configuration on DUT	26
5.2.1.1	Android + Bluedroid: WCN36x0	26
5.2.1.2	Android + Bluedroid: QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X	26

5.2.1.3	MDM + Linux + Bluez: QCA9377/QCA61x4	26
5.2.1.4	HCI Commands for BLE non-signaling tests	26
5.2.2	Configuration on PC/Tester	28
5.3	Method #3: Config with wdsdaemon	28
5.3.1	Configuration on DUT.....	28
5.3.2	Configuration on PC/Tester	29
6	QHS Non-Signaling Test.....	30
6.1	Method #1: Config with QRCT	30
6.1.1	Configuration on DUT.....	30
6.1.1.1	Android+Bluedroid: WCN6850/WCN67X0/WCN785X.....	30
6.1.2	Configuration on PC	30
6.1.3	Configuration on Equipment	31
6.2	Method #2: Config with btconfig or hcitool	31
6.3	Method #3: Config with wdsdaemon	31
7	FCC Certification.....	32
8	ETSI Certification	33
9	CE Certification.....	34
10	CTIA Certification.....	35
11	Google CTS	36
12	Google VTS	37
13	Bluetooth Power Test.....	38
14	Bluetooth IoT/IoP Test	39
15	Reference	40

1 Background

When trying to pass PHY/LL/RF BQB tests of BR/EDR/BLE or measure the hardware performance during FTM tests, you might need this KBA.

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

2 Bluetooth SIG Qualification

2.1 BQTF (Bluetooth Qualification Test Facility)

Generally speaking, for most ODM/OEM, when they have a new Bluetooth end product to launch, the first thing is to find a lab, which will work as consultant to help you to pass all related cases.

There're many options. You can find a detail list at SIG website:

<https://www.bluetooth.com/develop-with-bluetooth/qualification-listing/qualification-test-facilities/>

Please just choose one of them to buy their services, which will save you much effort. Before making decision, please make sure they have experience with Qualcomm Bluetooth solutions. Otherwise, you might suffer a lot when co-work with them.

A Bluetooth Qualification Test Facility (BQTF) is recognized by the Bluetooth SIG as competent to perform "Category A at or below the HCI layer" qualification test cases identified in the Test Case Reference List (TCRL). For example, RF, RF-PHY, BR/EDR protocols, LE protocols, ...

Actually, if you're professional enough, and you only need to run cases of profile layer, you can just submit your test reports as evidence directly at SIG website and buy the DID by yourselves.

If your company is one of BRTF (Bluetooth Recognized Test Facility), you can run all cases by yourselves, even there're "Category A at or below the HCI layer" qualification test cases.

2.2 Bluetooth Qualification Process

The Qualification is needed if it's a new design. If two products share the same design, then you only need to qualify one of them and list other products together.

Generally speaking, you will have your own design (schematic, layout) based on Qualcomm Reference Design. This means you might have your own changes:

- Adding new functionality
- Adding or removing a protocol, profile, or service
- Changing the specification version
- ICS change shall be treated like a new design

Please refer to PRD (Program Reference Document

https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=40972) to understand more about the changes.

Then, you need to follow the **Qualification Process with Required Testing**.

- Login <https://launchstudio.bluetooth.com/>
- Create a project
 - Fill QDID of Qualcomm's qualified listings. You can open CASE to ask Qualcomm supporters based on your build ID or about.html.
 - Select TCRL version
 - Select Product Type. Generally, it's End Product for most ODM/OEM.
 - Controller Core Configuration: BR/EDR/LE Combined Controller
 - Host Core Configuration: BR/LE Host
- Layer Selection
 - List project features
- ICS (Implementation Conformance Statements) Selection
 - Select supported capabilities for each layer
 - Generally, you can just use the ICS of Qualcomm's design
- Generate Test Plan
- Export *.pts file that will be used by PTS
- Perform Qualification Testing
- Upload Test Documentation
 - Upload the completed Test Plan, together with required test evidence.
 - Please refer to Section 3.1 of the Qualification PRD for more information about the test evidence that is required and must be uploaded for each test case category set forth in the Test Plan.
- List Products
 - Design Details
 - List all Products that use this Design (or combination of Designs) and that are distributed under a name that identifies your company as the source of the Product.
- Pay Declaration Fee
 - Select or purchase a Declaration ID
- Declare & Submit

2.3 Testing Profiles with PTS

Steps:

- Create Project
- Load *.pts
- Run cases

Please refer to KBA “How to pass BQB tests with PTS” for more details.

2.3.1 PTS Radio Module

It's used on PC to run PTS cases with DUT. Even you might buy services from BQTF, it's strongly recommended to buy such modules in your own company.

You can run cases by yourself to understand the issue better. And, when you discuss with Qualcomm supporters, it will be easier for your to understand the issue, instead of just copying messages or logs between Qualcomm and BQTF.

You can buy it at the SIG website:

<https://store.bluetooth.com/>

2.3.2 Profile Tuning Suite

For Software related test cases, especially Profiles, you need to install PTS software first (together with PTS Radio Module).

You can download at the SIG website:

<https://www.bluetooth.com/develop-with-bluetooth/qualification-listing/qualification-test-tools/profile-tuning-suite/>

When some cases failed, before opening CASE to get support from Qualcomm, please make sure you have read “BQB test Issues” in KBA “Necessary Logs for Bluetooth Issues” and collected all required logs.

2.4 Testing Controller with equipments

As pointed in the previous Chapter, for “Category A at or below the HCI layer” qualification test cases you need to ask BQTF for help (if you're not BRTF).

2.4.1 Validated Test Systems

If you're NOT BQTF or BRTF, you don't need to but these equipment at all. But it will be helpful if you can understand how to use these systems.

- TÜV Rheinland Group: CTE (RF, RF-PHY)
- **7Layers GmbH: InterLab Bluetooth RF Test Solution (RF, RF-PHY)**
- Allion Labs, Inc. (RF, RF-PHY)
- DEKRA Testing and Certification, S.A.U (BB, RF, RF-PHY)
- CTC Advanced GmbH (RF, RF-PHY)
- China Telecommunication Technology Labs (RF, RF-PHY)
- **Ellisys: Ellisys Bluetooth Qualifier (BB, LMP, LL, HCI)**
- Hyper Taiwan Technology Inc. (RF, RF-PHY)
- Teledyne LeCroy (LL, HCI)

You can find more details at the SIG website:

<https://www.bluetooth.com/develop-with-bluetooth/qualification-listing/qualification-test-tools/validated-recognized-test-equipment/>

2.4.2 Other Test Equipments

You can also verify Bluetooth in your own lab with equipment below:

- Rohde&Schwarz CMW270/CMW280/CMW500 (RF)
- Anritsu MT8852B (RF)
- Angilent N4010A
- Litepoint IQ
- ...

2.4.3 Configuration on DUT

Please refer to chapters below to configure DUT and PC (BR/EDR signaling test and LE non-signaling test).

2.5 Test Specification/ Test Suite

You can download TCRL and Test Specification at SIG website:

<https://www.bluetooth.com/specifications/qualification-test-requirements/>

TCRL: Test Case Reference List

TCMT: Test Case Mapping Table

ICS: Implementation Conformance Statement

IXIT: Implementation Extra Information for Test

2.5.1 RF

Transmitter

Output Power	RF/TRM/CA/BV-01-C
Power Density	RF/TRM/CA/BV-02-C
Power Control	RF/TRM/CA/BV-03-C
TX Output Spectrum-Frequency Range	RF/TRM/CA/BV-04-C
TX Output Spectrum-20 dB Bandwidth	RF/TRM/CA/BV-05-C
TX Output Spectrum-Adjacent Channel Power	RF/TRM/CA/BV-06-C
Modulation Characteristics	RF/TRM/CA/BV-07-C
Initial Carrier Frequency Tolerance	RF/TRM/CA/BV-08-C
Carrier Frequency Drift	RF/TRM/CA/BV-09-C
EDR Relative Transmit Power	RF/TRM/CA/BV-10-C
EDR Carrier Frequency Stability and Modulation Accuracy	RF/TRM/CA/BV-11-C
EDR Differential Phase Encoding Enhanced	RF/TRM/CA/BV-12-C
EDR In-Band Spurious Emission	RF/TRM/CA/BV-13-C
Enhanced Power Control	RF/TRM/CA/BV-14-C
EDR Guard Time	RF/TRM/CA/BV-15-C
EDR Synchronization Sequence and Trailer	RF/TRM/CA/BV-16-C

Receiver

Sensitivity – Single slot packets	RF/RCV/CA/BV-01-C
Sensitivity – Multi slot packets	RF/RCV/CA/BV-02-C
C/I Performance	RF/RCV/CA/BV-03-C
Blocking Performance	RF/RCV/CA/BV-04-C
Intermodulation Performance	RF/RCV/CA/BV-05-C
Maximum Input Level	RF/RCV/CA/BV-06-C
EDR Sensitivity	RF/RCV/CA/BV-07-C
BER Floor Sensitivity	RF/RCV/CA/BV-08-C
EDR C/I Performance	RF/RCV/CA/BV-09-C
EDR Maximum Input Level	RF/RCV/CA/BV-10-C

2.5.2 (BLE) RF-PHY

TBD.

2.5.3 (BLE) LL

TBD.

2.5.4 BB

TBD.

2.5.5 LMP

TBD.

2.5.6 HCI

TBD.

2.5.7 Profiles/Protocol

A2DP	
HFP	
HSP	
AVRCP	
GATT	
AVDTP	
AVCTP	
RFCOMM	
L2CAP	
GAP	
SMP	
SDP	
HID	
HOGP	
ScPP	
IOTP	
DID	
PAN	
OPP	
PBAP	
MAP	
SAP	
SPP	

3 BR/EDR Signaling Test

3.1 Method #1: Config with QRCT

3.1.1 Configuration on DUT

3.1.1.1 Android + Bluebird: WCN36x0 or

QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X

- Turn off Bluetooth in UI: Setting → Bluetooth.
- Run ftmdaemon in ADB root shell:

```
(CMD.exe shell) # adb root
(CMD.exe shell) # adb shell
(adb root shell) # ftmdaemon -n
```

If you wish to unplug USB cable during tests, please start ftmdaemon via the command below instead:

```
(adb root shell) # cd /data
(adb root shell) # nohup ftmdaemon -ndd &
```

3.1.1.2 MDM + Linux + Bluez: QCA9377/QCA61x4

First, please create a file named /etc/bluetooth/firmware.conf and add following lines.

```
IBS=0
DEEP_SLEEP=0
```

Then, please run the commands below:

```
(BT if OFF at first.)
(CMD.exe shell) # adb root
(CMD.exe shell) # adb shell
(adb root shell) # echo 1 > /sys/class/rfkill/rfkill0/state
(adb root shell) # /usr/bin/hciattach /dev/ttyHS0 qca 3000000 -t120 flow -f0
(adb root shell) # /usr/bin/ftmdaemon -n
```

3.1.1.3 WOS

Necessary Register to configure:

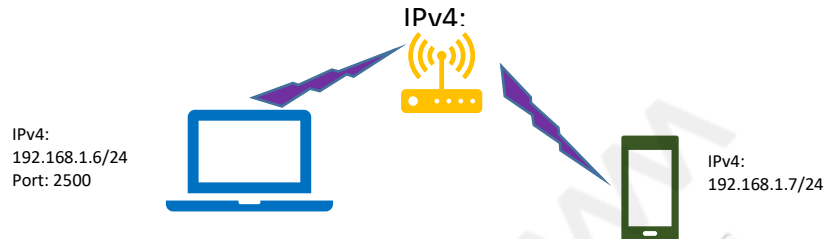
- HKEY_LOCAL_MACHINE\SYSTEM\
 - CurrentControlSet\
 - Services\QCDIAGROUTER\
 - Parameters\
 - ServerName
 - TCPPort
 - Services\QcBluetooth\
 - Parameters\
 - DynamicSwitch

Please set DynamicSwitch to 1, ServerName = "192.168.1.6" (PC IP address) and TCPPort = "2500"

, then reboot.

- Run CMD.exe as Administrator.
- Execute "QcBtftmddiagapp.wp.exe".

Please make sure the PC (which runs QXDM) and DUT (WOS platform) run in the same LAN:



3.1.2 Configuration on PC

- Launch QRCT, and config COM port
- Select menu "FTM Command" → "BT" → "HCI Commands", then QRCT will start a new dialog "BT HCI Commands"

Quadrant
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@quectel.com

QCA Radio Control Toolkit

File View FTM Command Tool Window Custom APIs Help

COM Port ▼ HW Ver QMSL Library Mode ▼ Target ▼ Diag Type ▼

BT HCI Commands

☒ Disable Legacy Support

HCI DUT Mode

HCI DUT Mode

HCI Write Scan Enable

No Scan Enabled Select

Enable

BD Address

Set BD Address Input 6 hex bytes

Get BD Address

HCI Reset

HCI Reset

HCI Set Event Filter

Auto Accept Off Select

Set Filter

HCI User Cmd

Input is in hex format and separated by commas

Send Command

Download NVM and Patch

NVM Location Download NVM

☐ Enable Bluetooth Logs

Get Logs Clear

Logs	Size	HCI Event Payload

- Click "Disable Legacy Support"
- Click "HCI Reset"
- Click "HCI DUT Mode"

Instead of clicking this button, you can also send "HCI User Cmd" to let chip enter DUT mode:

```
0x01, 0x1A, 0x0C, 0x01, 0x03 // HCI Write Scan Enable, Inquiry & Page Scan Enable
```

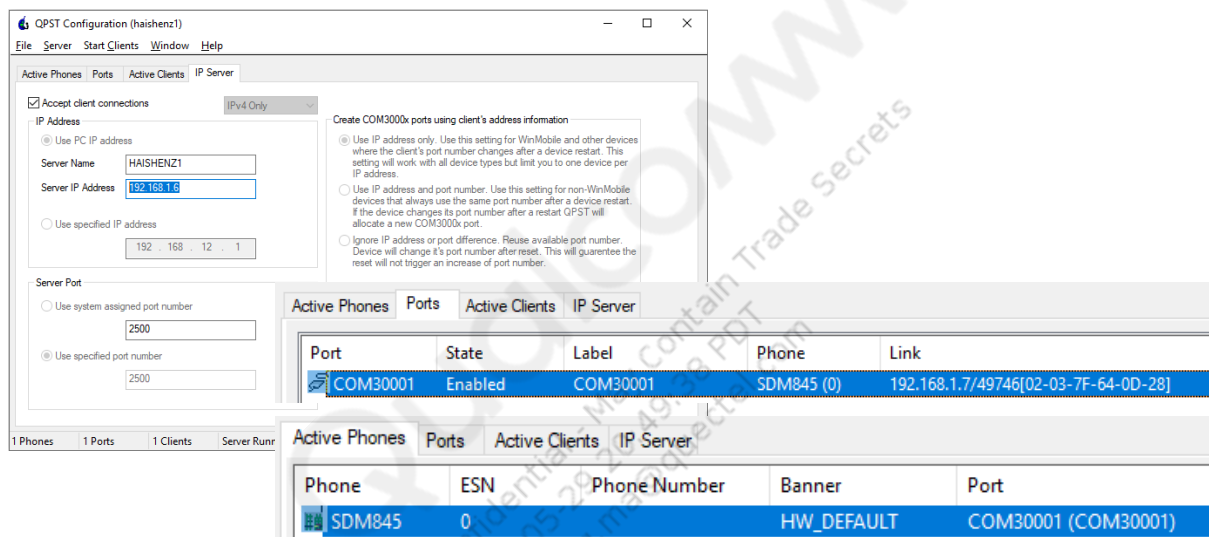
```

0x01, 0x05, 0x0C, 0x03, 0x02, 0x00, 0x02 // HCI Set Event Filter, Auto Accept
Connection
0x01, 0x03, 0x18, 0x00 // HCI Enable Device Under Test Mode
0x01, 0x20, 0x0C, 0x00
0x01, 0x22, 0x0C, 0x00

```

3.1.2.1 WOS

For WOS platform,
We need to configure QPST to simulate a COM port.



Then, you can see a simulated COM port “COM30001”, which can be used for QRCT.

3.1.3 Configuration on Equipment

If you don’t know what to do with the test equipment (for example N4010A/MT8852B), please open a Hardware case to check with Bluetooth Hardware team.

3.2 Method #2: Config with btconfig or hcitool

3.2.1.1 Android+Bluedroid: WCN36x0

Please run “btconfig /dev/smd3 rawcmd <ogf> <ocf> <parameters>” respectively.
You need to specify “/dev/smd3” in the commands.

For example,

```

(CMD.exe shell) # adb shell
(adb root shell) # btconfig /dev/smd3 rawcmd 0x03, 0x0003
(adb root shell) # btconfig /dev/smd3 rawcmd 0x06, 0x03
(adb root shell) # btconfig /dev/smd3 rawcmd 0x03, 0x05, 0x02, 0x00, 0x02

```

```
(adb root shell) # btconfig /dev/smd3 rawcmd 0x03, 0x1A, 0x03
(adb root shell) # btconfig /dev/smd3 rawcmd 0x03, 0x20, 0x00
(adb root shell) # btconfig /dev/smd3 rawcmd 0x03, 0x22, 0x00
```

3.2.1.2 Android+Bluedroid: QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X

Please run wdsdaemon first, then run “btconfig rawcmd <ogf> <ocf> <parameters>” directly. You can find wdsdaemon and btconfig in your Android codes.

First, please start wdsdaemon in one CMD.exe window.

```
(CMD.exe shell) # root
(CMD.exe shell) # adb remount
(CMD.exe shell) # adb shell
(adb root shell) # wdsdaemon -su
```

If you wish to unplug USB cable during tests, please start wdsdaemon via the command below instead:

```
(adb root shell) # cd /data
(adb root shell) # nohup wdsdaemon -su &
```

Then, please start another CMD.exe window to send HCI commands via btconfig. BTW, you don’t need to specify the port name in the commands.

```
(CMD.exe shell) # adb shell
(adb root shell) # btconfig rawcmd 0x03, 0x0003
(adb root shell) # btconfig rawcmd 0x06, 0x03
(adb root shell) # btconfig rawcmd 0x03, 0x05, 0x02, 0x00, 0x02
(adb root shell) # btconfig rawcmd 0x03, 0x1A, 0x03
(adb root shell) # btconfig rawcmd 0x03, 0x20, 0x00
(adb root shell) # btconfig rawcmd 0x03, 0x22, 0x00
```

3.2.1.3 MDM + Linux + Bluez

Please run “hcidtool cmd <ogf> <ocf> <parameters>” respectively. You can find hcidtool in the Bluez stack folder.

For example,

```
(CMD.exe shell) # adb shell
(adb root shell) # hcidtool cmd 0x06 0x0003
(adb root shell) # hcidtool cmd 0x03 0x0005 0x02 0x00 0x02
(adb root shell) # hcidtool cmd 0x03 0x001A 0x03
(adb root shell) # hcidtool cmd 0x03 0x0020 0x00
(adb root shell) # hcidtool cmd 0x03 0x0022 0x00
```


4 BR/EDR Non-Signaling Test

4.1 Method #1: Config with QRCT

4.1.1 Configuration on DUT

4.1.1.1 *Android + Bluebird: WCN36x0 or QCA61x4/ WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X*

- Turn off bluetooth in UI: Setting → Bluetooth.
- Run ftmdaemon in ADB root shell:

```
(CMD.exe shell) # adb root
(CMD.exe shell) # adb shell
(adb root shell) # ftmdaemon -n
```

If you wish to unplug USB cable during tests, please start ftmdaemon via the command below instead:

```
(adb root shell) # cd /data
(adb root shell) # nohup ftmdaemon -ndd &
```

4.1.1.2 *MDM + Linux + Bluez: QCA9377/QCA61x4*

First, please create a file named /etc/bluetooth/firmware.conf and add following lines.

```
IBS=0
DEEP_SLEEP=0
```

Then, please run the commands below:

```
(BT if OFF at first.)
(CMD.exe shell) # adb root
(CMD.exe shell) # adb shell
(adb root shell) # echo 1 > /sys/class/rfkill/rfkill0/state
(adb root shell) # /usr/bin/hciattach /dev/ttyHS0 qca 3000000 -t120 flow -f0
(adb root shell) # /usr/bin/ftmdaemon -n
```

4.1.1.3 *WOS*

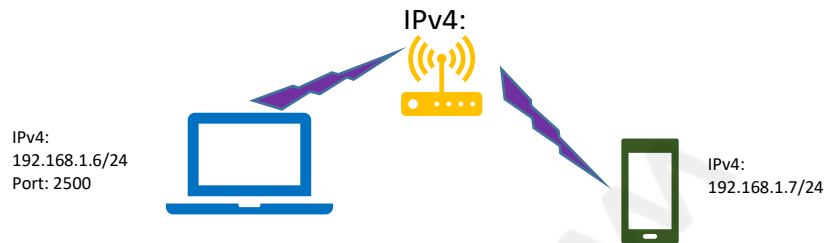
Necessary Register to configure:

- HKEY_LOCAL_MACHINE\SYSTEM\
 - CurrentControlSet\
 - Services\QCDIAGROUTER\
 - Parameters\
 - ServerName
 - TCPPort
 - Services\QcBluetooth\
 - Parameters\
 - DynamicSwitch

Please set DynamicSwitch to 1, ServerName = "192.168.1.6" (PC IP address) and TCPPort = "2500", then reboot.

- Run CMD.exe as Administrator.
- Execute "QcBtftmdiagapp.wp.exe".

Please make sure the PC (which runs QXDM) and DUT (WOS platform) run in the same LAN:



4.1.2 Configuration on PC

- Launch QRCT, and config COM port
- Select menu "FTM Command" → "BT" → "EPTM", then QRCT will start a new dialog "Bluetooth EPTM"

Quadrant
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@quectel.com

QCA Radio Control Toolkit

File View FTM Command Tool Window Custom APIs Help

COM Port HW Ver QMSL Library Mode Target Diag Type

BluetoothEPTM

Settings

Select Bluetooth Chip QCA6174/WCN2243/36xx ☒ Disable Legacy Support

Test BD Address (Hex: MSB->LSB)
bd 35 9c bd 35 9c
6 Space separated values

BT_DH5 Packet Type 339 Payload Length < 339
0 LT Address PSEUDO_RANDOM Transmit Pattern
9 Power ☐ Whitening Enabled
1 Receiver HighGain Enable

Hopping Mode (Channels in Decimal)
☐ Enable Hopping
39 39 39 39 39
5 Space separated values

Rx Burst Config (Rx Packet Number)
0 RX Packet Number 0 = Continuous Rx Burst Config

Tx Only Burst Rx Only Burst Rx Test Stats Stop

Tx Continuous
7 Power
39 Continuous Transmit Channel
Carrier Test Type
1 Pattern Length (1-32)
0 Pattern (Hex UINT32)
Tx Continuous Stop

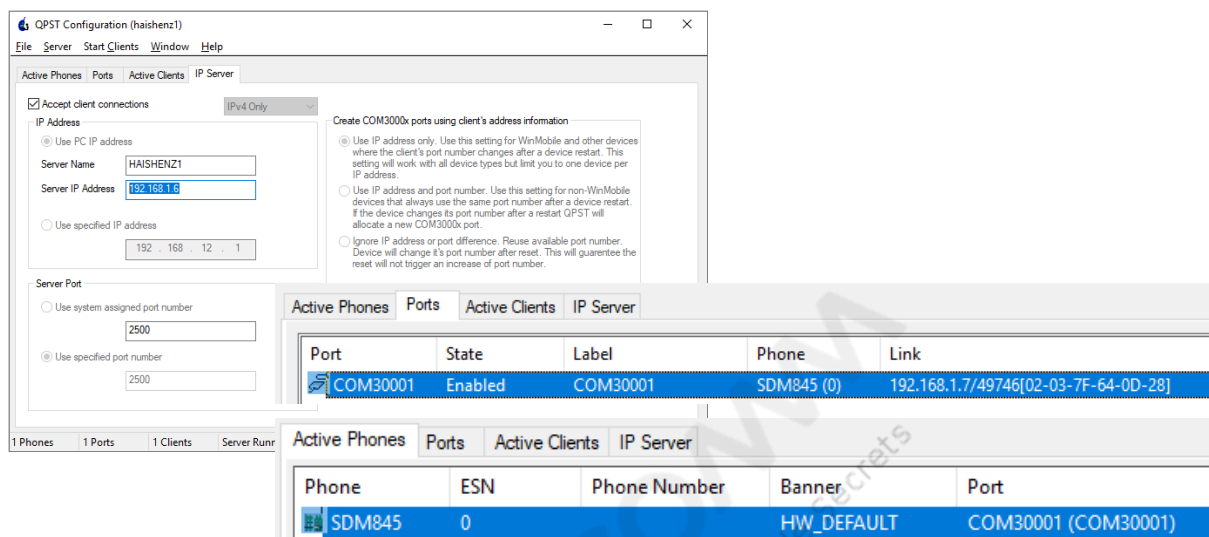
BT Receiver Test Statistics
Total Packets Received: 0
Total Packet Access Errors: 0
Total HEC Errors: 0
Total CRC Errors: 0
Total Packet Bit Errors: 0
Total First Half Bit Errors: 0
Total Last Half Bit Errors: 0
Avg. RSSI Reading: 0
Bit Error Rate (BER%):

- Click "Disable Legacy Support"
- Select Bluetooth Chip as "QCA6174/WCN2243/36xx"
- Click buttons "Tx Only Burst", "Rx Only Burst", "Tx Continuous", ... to do tests
- Click "Stop" to end the test

4.1.2.1 WOS

For WOS platform,

We need to configure QPST to simulate a COM port.



Then, you can see a simulated COM port "COM30001", which can be used for QRCT.

4.1.3 Configuration on Equipment

If you don't know what to do with the test equipment (for example N4010A/MT8852B), please open a Hardware case to check with Bluetooth Hardware team.

4.2 Method #2: Config with btconfig or hcitool

4.2.1.1 Android + Bluebird: WCN36x0

Please run "btconfig /dev/smd3 rawcmd <ogf> <ocf> <parameters>" respectively.

You can refer to previous chapter about how to use btconfig.

You can refer to later chapter about the details of HCI commands for BR/EDR non-signaling tests.

4.2.1.2 Android + Bluebird: QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X

Please run wdsdaemon first, then run "btconfig rawcmd <ogf> <ocf> <parameters>" directly.

You can refer to the chapter about signaling tests above for detail steps.

4.2.1.3 MDM + Linux + Bluez: QCA9377/QCA61x4

Please run "hcitool /dev/smd3 cmd <ogf> <ocf> <parameters>" respectively.

4.2.1.4 HCI Commands for BR/EDR non-signaling tests

4.2.1.4.1 Tx Continuous

For example,

For WCN36x0 related projects, you can run the command below:

```
(adb root shell) # btconfig /dev/smd3 rawcmd 0x3F 0x0004 0x05 0x00 0x09 0x05 0x20 0x00 0x00 0x00 0x00
```

Description:

> 0x3F (OGF)
> 0x004 (OCF)
> 0x05 = Sub command (continuous transmit)
> 0x00 = TX channel, range: 0x00–4F, you can use 0x00/0x27/0x4E match with CH0/39/78
> 0x09 = Transmit power control level (0–9), default 0x09
> 0x04 = Transmit type:
04 1-Carrier
05 GFSK
05 1-PRBS9 GFSK
06 1-PRBS15 GFSK
07 1-Pattern GFSK
08 2-PRBS9 PI/4 DQPSK
09 2-PRBS15 PI/4 DQPSK
0A 2-Pattern PI/4 DQPSK
0B 3-PRBS9 8DPSK
0C 3-PRBS15 8DPSK
0D 3-Pattern 8DPSK
> 0x20 Pattern length, range: 0x01–20, you can use 0x01-20, default 0x20
> 0x00 0x00 0x00 0x00 pattern, default value.

4.2.1.4.2 Tx Burst

For example,

For WCN36x0 related projects, you can run the command below:

```
(adb root shell) #btconfig /dev/smd3 rawcmd 0x3F 0x0004 0x04 0x27 0x27 0x27  
0x27 0x27 0x04 0x04 0x00 0x09 0x01 0x9C 0x35 0xBD 0x9C 0x35 0xBD 0x00 0x1B  
0x00 0x00
```

Description:

> 0x3F (OGF)
> 0x004 (OCF)
> 0x04 = subcommand (Transmit only)
> 0x27 0x27 0x27 0x27 0x27 = hop channels
> 0x04 = Pseudorandom (data pattern) (Please refer to the table below for more details.)
> 0x04 = DH1 packet type (Please refer to the table below for more details.)
> 0x00 = Whitening Off (01 = Whitening on)
> 09 = Power control level (0-9)
> 0x01 = High Rx gain (0x00 = Low Rx gain)

>0x9C 0x35 0xBD 0x9C 0x35 0xBD = BD_ADDR, arbitrary
 >0x00 = Hopping array (0x01 = 79-Channel hopping)
 >0x1B 0x00 = >0x1B 0x00 = Length of test data (Please refer to the table below for more details.)
 >0x00 = LT_ADDR

Table about “data pattern”:

Value	0x00	0x01	0x02	0x03	0x04
Pattern	All '0' bits	All '1' bits	10101010	00001111	Pseudorandom

Table about “packet type”:

Value	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Packet	NULL	POLL	FHS	DM1	DH1	HV1	HV2	HV3
Value	0x08	0x09	0x0A	0x0B	0x0E	0x0F	0x17	0x1C
Packet	DV	AUX1	DM3	DH3	DM5	DH5	EV3	EV4
Value	0x1D	0x24	0x28	0x2A	0x2B	0x2E	0x2F	0x36
Packet	EV5	2-DH1	3-DH1	2-DH3	3-DH3	2-DH5	3-DH5	2-EV3
Value	0x37	0x3C	0x3D					
Packet	3-EV3	2-EV5	3-EV5					

Table about “length of test data”:

Value	0x11 0x00	0x1B 0x00	0x0A 0x00	0x14 0x00	0x1E 0x00	0x13 0x00
Packet	DM1	DH1	HV1	HV2	HV3	DV
Value	0x1D 0x00	0x79 0x00	0xB7 0x00	0xE0 0x00	0x53 0x01	0x1E 0x00
Packet	AUX1	DM3	DH3	DM5	DH5	EV3
Value	0x78 0x00	0xB4 0x00	0x36 0x00	0x53 0x00	0x6F 0x01	0x28 0x02
Packet	EV4	EV5	2-DH1	3-DH1	2-DH3	3-DH3
Value	0xA7 0x02	0xFD 0x03	0x3C 0x00	0x5A 0x00	0x68 0x01	0x1C 0x02
Packet	2-DH5	3-DH5	2-EV3	3-EV3	2-EV5	3-EV5

4.2.1.4.3 Rx Burst

For example,

For WCN36x0 related projects, you can run the command below:

```
(adb root shell) # btconfig /dev/smd3 rawcmd 0x3F 0x0004 0x06 0x27 0x27 0x27
0x27 0x27 0x04 0x04 0x00 0x09 0x01 0x9C 0x35 0xBD 0x9C 0x35 0xBD 0x00 0x1B
0x00 0x00
```

Description:

>0x3F (OGF)
 >0x004 (OCF)
 >0x06 = subcommand (receive only)
 >0x27 0x27 0x27 0x27 0x27 = hop channels
 >0x04 = Pseudorandom (data pattern) (Please refer to the table below for more details.)

>0x04 = DH1 packet type (Please refer to the table below for more details.)
 >0x00 = LT_ADDR

Table about “data pattern”:

Value	0x00	0x01	0x02	0x03	0x04
Pattern	All '0' bits	All '1' bits	10101010	00001111	Pseudorandom

Table about “packet type”:

Value	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Packet	NULL	POLL	FHS	DM1	DH1	HV1	HV2	HV3
Value	0x08	0x09	0x0A	0x0B	0x0E	0x0F	0x17	0x1C
Packet	DV	AUX1	DM3	DH3	DM5	DH5	EV3	EV4
Value	0x1D	0x24	0x28	0x2A	0x2B	0x2E	0x2F	0x36
Packet	EV5	2-DH1	3-DH1	2-DH3	3-DH3	2-DH5	3-DH5	2-EV3
Value	0x37	0x3C	0x3D					
Packet	3-EV3	2-EV5	3-EV5					

Qualcomm
 Confidential - May Contain Trade Secrets
 2022-05-29 20:49:38 PDT
 qingzong.ma@quectel.com

5 BLE Non-Signaling Test

5.1 Method #1: Config with QRCT

5.1.1 Configuration on DUT

5.1.1.1 *Android+Bluedroid: WCN36x0 or*

QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X

- Turn off bluetooth in UI: Setting → Bluetooth.
- Run ftmdaemon in ADB root shell:

```
(CMD.exe shell) # adb root
(CMD.exe shell) # adb shell
(adb root shell) # ftmdaemon -n
```

If you wish to unplug USB cable during tests, please start ftmdaemon via the command below instead:

```
(adb root shell) # cd /data
(adb root shell) # nohup ftmdaemon -ndd &
```

5.1.1.2 *MDM + Linux + Bluez: QCA9377/QCA61x4*

First, please create a file named /etc/bluetooth/firmware.conf and add following lines.

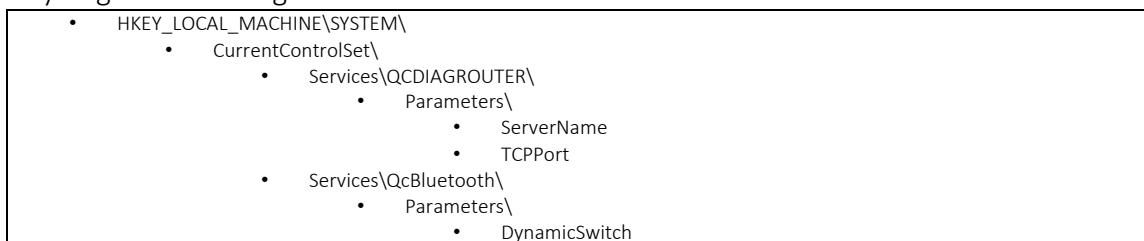
```
IBS=0
DEEP_SLEEP=0
```

Then, please run the commands below:

```
(BT if OFF at first.)
(CMD.exe shell) # adb root
(CMD.exe shell) # adb shell
(adb root shell) # echo 1 > /sys/class/rfkill/rfkill0/state
(adb root shell) # /usr/bin/hciattach /dev/ttyHS0 qca 3000000 -t120 flow -f0
(adb root shell) # /usr/bin/ftmdaemon -n
```

5.1.1.3 *WOS*

Necessary Register to configure:

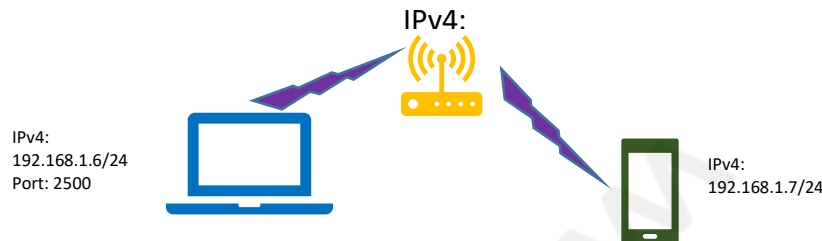


Please set DynamicSwitch to 1, ServerName = "192.168.1.6" (PC IP address) and TCPPort = "2500"

, then reboot.

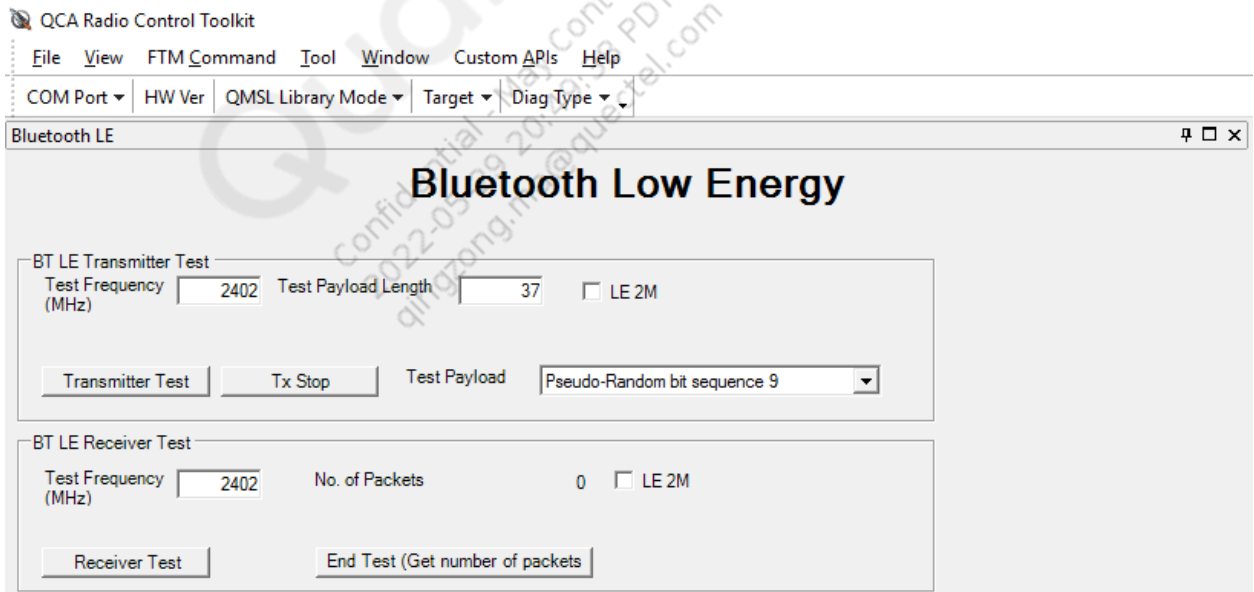
- Run CMD.exe as Administrator.
- Execute “QcBtftmdiagapp.wp.exe”.

Please make sure the PC (which runs QXDM) and DUT (WOS platform) run in the same LAN:



5.1.2 Configuration on PC

- Launch QRCT, and config COM port
- Select menu "FTM Command" → "BT" → "LE", then QRCT will start a new dialog "Bluetooth LE"

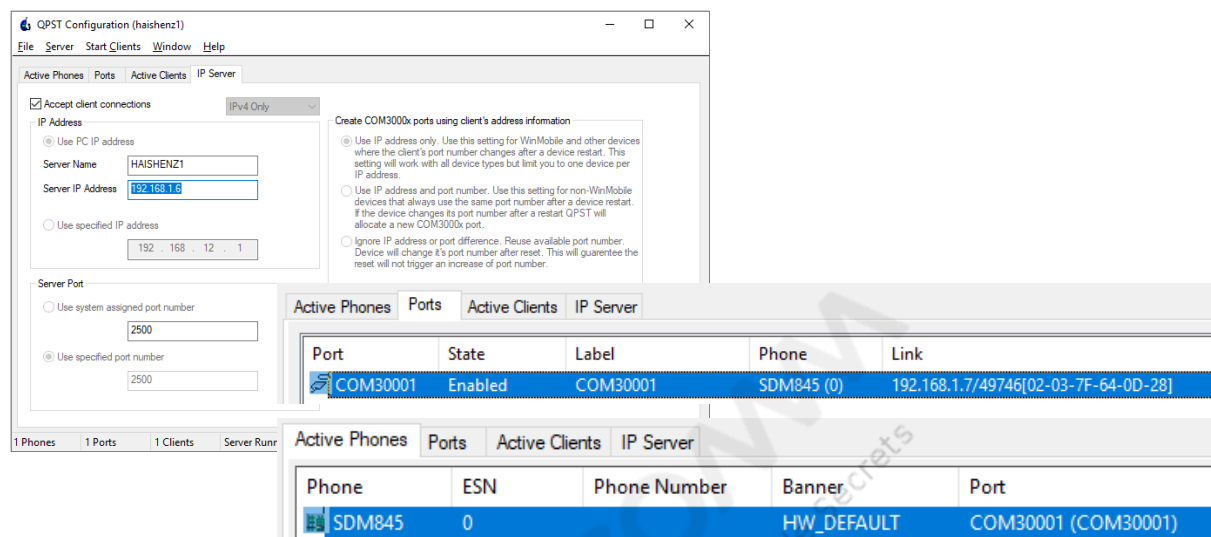


- Select Bluetooth Chip as "QCA6174/WCN2243/36xx"
- If you are testing BLE PHY 2Mbps (Bluetooth 5.0 feature), please make sure to check “LE 2M”.
- Click buttons "Transmitter Test", "Receiver Test", ... to do tests
- Click “Tx Stop” “End Test” to end the test

5.1.2.1 WOS

For WOS platform,

We need to configure QPST to simulate a COM port.



Then, you can see a simulated COM port “COM30001”, which can be used for QRCT.

5.1.3 Configuration on Equipment

If you don't know what to do with the test equipment (for example N4010A/MT8852B), please open a Hardware case to check with Bluetooth Hardware team.

5.2 Method #2: Config with btconfig or hcitool

5.2.1 Configuration on DUT

5.2.1.1 Android + Blueroid: WCN36x0

Please run “btconfig /dev/smd3 rawcmd <ogf> <ocf> <parameters>” respectively.

You can refer to previous chapter about how to use btconfig.

You can refer to later chapter about the details of HCI commands for BR/EDR non-signaling tests.

5.2.1.2 Android + Blueroid: QCA61x4/WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X

Please run wdsdaemon first, then run “btconfig rawcmd <ogf> <ocf> <parameters>” directly.

You can refer to the chapter about BR/EDR signaling test above to get more details.

5.2.1.3 MDM + Linux + Bluez: QCA9377/QCA61x4

Please run “hcitool cmd <ogf> <ocf> <parameters>” respectively.

5.2.1.4 HCI Commands for BLE non-signaling tests

For the HCI details, please refer to Core Specification (Chapter 7.8 LE CONTROLLER COMMANDS).

5.2.1.4.1 Tx (PHY=1Mbps): ogf=0x08, ocf=0x1E

For example,

For WCN36x0 related projects, you can run the command below:

```
(adb root shell) # btconfig /dev/smd3 rawcmd 0x08 0x001E 0x00 0x25 0x00
```

Description:

```
>0x00 = channel, range is 0x00–0x27. You can use 0x00/0x13/0x26 match with CH0/19/38.  
>0x25 = data length range is 0x00-0x25  
>0x00 = packet, range is 0x00-0x07, refer below define for GFSK  
    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'
```

5.2.1.4.2 Tx (PHY=2Mbps, Coded): ogf=0x08, ocf=0x34

For example,

For WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X related projects,

You can run the command below to test LE2M:

```
(adb root shell) # btconfig rawcmd 0x08 0x0034 0x00 0x25 0x00 0x02
```

You can run the command below to test Coded PHY,

5.2.1.4.3 Rx (PHY=1Mbps): ogf=0x08, ocf=0x1D

For example,

For WCN36x0 related projects, you can run the command below:

```
(adb root shell) # btconfig /dev/smd3 rawcmd 0x08 0x001D 0x00
```

Description:

```
>0x00 = channel, range is 0x00–0x27. Frequency range: 2402 MHz–2480 MHz
```

5.2.1.4.4 Rx (PHY=2Mbps, Coded): ogf=0x08, ocf=0x33

For example,

For WCN39x0/QCA639x/WCN6850/WCN67X0/WCN785X related projects,

You can run the command below to test LE2M:

```
(adb root shell) # btconfig rawcmd 0x08 0x0033 0x00 0x02 0x00
```

5.2.1.4.5 Test End: ogf=0x08, ocf=0x1F

For example,

For WCN36x0 related projects, you can run the command below:

```
(adb root shell) # btconfig /dev/smd3 rawcmd 0x08 0x001F
```

5.2.2 Configuration on PC/Tester

Please install USB drivers to enable ADB shell.

If you're using Interlab Test Solution(https://www.7layers.com/test-products/bluetooth_rf_test_set_ups/bluetooth-rf-tester), it shall support to send commands via ADB shell.

For example,

Le Transmitter Test	ADB	▼	HCI	QUERY	▼	btconfig /dev/smd3 rawcmd 0x08 0x001E 0x00 0x25 0x00
Le Receiver Test	ADB	▼	HCI	QUERY	▼	btconfig dev/smd3 rawcmd 0x08 0x001D 0x00
Le Test End	ADB	▼	HCI	QUERY	▼	btconfig /dev/smd3 rawcmd 0x08 0x001F
Le Reset	ADB	▼	HCI	QUERY	▼	btconfig /dev/smd3 rawcmd 0x03 0x0003

If it failed unexpectedly, please test with ADB shell in CMD.exe directly. If all the commands and events are OK, then you have to double confirm with your test agent to configure the Interlab equipments/software properly.

5.3 Method #3: Config with wdsdaemon

You might need this method in the BQB test lab.

Wdadaemon works like a “router” or “switch”, which will read all commands from PC via USB NMEA port and forward all of them to Bluetooth controller, and will read all events from Bluetooth controller and forward all of them to PC via the same USB NMEA port.

PC needs to read and write USB NMEA COM port. If there's no NMEA port at all, then you need to open a case to BSP/USB team to help to enable it on your DUT.

DUT needs to read and write USB NMEA tty device, which is defined in BT_HS_NMEA_DEVICE.

5.3.1 Configuration on DUT

Generally speaking, you can use the command below to start wdsdaemon directly.

```
(Turn off Bluetooth from UI)  
(adb root shell) # wdsdaemon
```

About the definition of BT_HS_NMEA_DEVICE, by default, it's "/dev/ttyGS0". You can getprop sys.usb.config to make sure if ttyGS0 or at_usb1 should be used.

- From Android O/P release, usb driver use f_cdev.c for usb serial function (**sys.usb.config** property will have **serial_cdev** string, like setprop sys.usb.config diag,serial_cdev,rmnet_gsi,adb). In such builds, the device name for the serial port in android side would be at_usb*, there could be **at_usb0/ at_usb1/ at_usb2/...**, AT/MODEM/NMEA services will map to it.
- On old android release (for example M/N release), serial_tty will be used (**sys.usb.config** property will have **serial_tty** string, like diag,serial_tty,serial_smd,adb), thus **tttGS0** can be used.

If you have trouble to understand which USB port shall be used for wdsdaemon, you can turn to BSP/USB team for help.

If your DUT use a different USB NMEA port from the default definition of BT_HS_NMEA_DEVICE, for example, "/dev/at_usb1", please use the command below instead to start the wdsdaemon:

(Turn off Bluetooth from UI)
(adb root shell) # wdsdaemon -i /dev/at_usb1

This change was introduced in CR# 2331351. Please make sure your build has integrated this change to support the "-i" parameters.

5.3.2 Configuration on PC/Tester

Install QPST and DUT driver on PC/tester to simulate COM port.

Tester console is actually a Windows machine and can be operated like our PC.

Configure HCI command output COM port to NMEA port.

Flow control is none and baud rate is 115200.

Please note RF test program can only support COM1 - COM8. If NMEA port number is out of that range, you need to manually adjust the number. The step to do that is:

- Launch device manager
- Right click on NMEA port and select Properties
- Select Port Settings
- Click on Advanced
- Change COM port number

After that, you can send HCI commands (You can refer to these commands in the previous chapter or Bluetooth Core Specification) to the NMEA port directly. And, you can also receive HCI events from the NMEA port.

6 QHS Non-Signaling Test

6.1 Method #1: Config with QRCT

6.1.1 Configuration on DUT

6.1.1.1 *Android+Bluedroid: WCN6850/WCN67X0/WCN785X*

- Turn off bluetooth in UI: Setting → Bluetooth.
- Run ftmdaemon in ADB root shell:

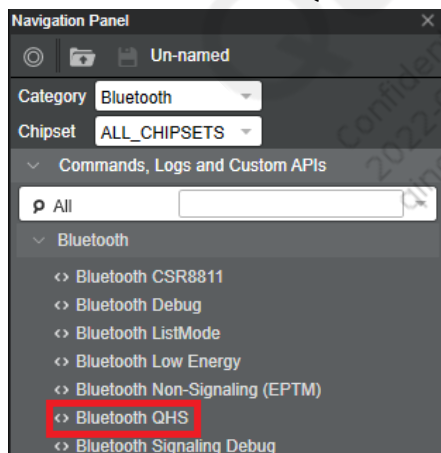
```
(CMD.exe shell) # adb root  
(CMD.exe shell) # adb shell  
(adb root shell) # ftmdaemon -n
```

If you wish to unplug USB cable during tests, please start ftmdaemon via the command below instead:

```
(adb root shell) # cd /data  
(adb root shell) # nohup ftmdaemon -n &
```

6.1.2 Configuration on PC

- Launch QRCT, and config COM port
- Select menu "Bluetooth QHS" in the Navigation Panel.



- Then, you can see a new window for QHS related tests.

Commands

Bluetooth QHS

QHS (QUALCOMM HIGH SPEED)

Select Test Mode: Tx

Frequency (MHz): 2402

Payload Length: 35

Payload Pattern: Pseudo-Random bit sequence 9

PHY_RATE: QHS2P_6Mbps

Power: 9

Max Packets: 0

Start

Stop

End Test (Get Number Of Packets)

No. Of Packets —

- Check “Tx”/ “Rx” or Transmitter or Receiver tests.
- Click buttons "Start", ... to do tests.
- Click “Stop” or “End Test” to end the test.

6.1.3 Configuration on Equipment

If you don't know what to do with the test equipment (for example N4010A/MT8852B), please open a Hardware case to check with Bluetooth Hardware team.

6.2 Method #2: Config with btconfig or hcitool

TBD

6.3 Method #3: Config with wdsdaemon

TBD

7 FCC Certification

TBD

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

8 ETSI Certification

TBD

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qtecel.com

9 CE Certification

TBD

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

10 CTIA Certification

TBD

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

11 Goolge CTS

TBD

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

12 Google VTS

TBD

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

13 Bluetooth Power Test

TBD.

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

14 Bluetooth IoT/IoP Test

TBD

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com

15 Reference

80-VJ734-1

80-WL400-14

80-VE132-7

80-WL024-12

80-y0476-4

Qualcomm
Confidential - May Contain Trade Secrets
2022-05-29 20:49:38 PDT
qingzong.ma@qcom.com