FC2x&FC900E Third-Party Linux Wi-Fi&Bluetooth User Guide

Wi-Fi&Bluetooth Module Series

Version: 1.0.0

Date: 2022-07-04

Status: Preliminary

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local offices. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm.
Or email us at: support@quectel.com.

About the Document

Revision History

Version	Date	Author	Description
-	2021-01-28	Qingzong MA	Creation of the document
1.0.0	2022-07-04	Qingzong MA/ Lucas HUANG/ Lidong ZHOU/ Hardy ZHANG	Preliminary

Contents

Figure Index

1 Introduction

Quectel FC20, FC21 and FC900E modules are low-power and cost-effective Wi-Fi/Bluetooth modules, with Wi-Fi and Bluetooth features supported on third-party Linux system. This document introduces how to enable the two features, including verification method and the description of factory test mode.

2 Enable Wi-Fi

This chapter describes how to enable the Wi-Fi function of FC20, FC21 and FC900E modules, as well as the function verification and the corresponding factory test methods on the example of IMX8QXP platform on third-party Linux system.

2.1. Build Environment

2.1.1. Hardware Environment

1. Prepare the following hardware:
   - Quectel FC20, FC21 or FC900E module
   - IMX8QXP demo board
2. Install the module and demo board as shown below (NXP board MCIMX8QXP-CPU is used in the example):

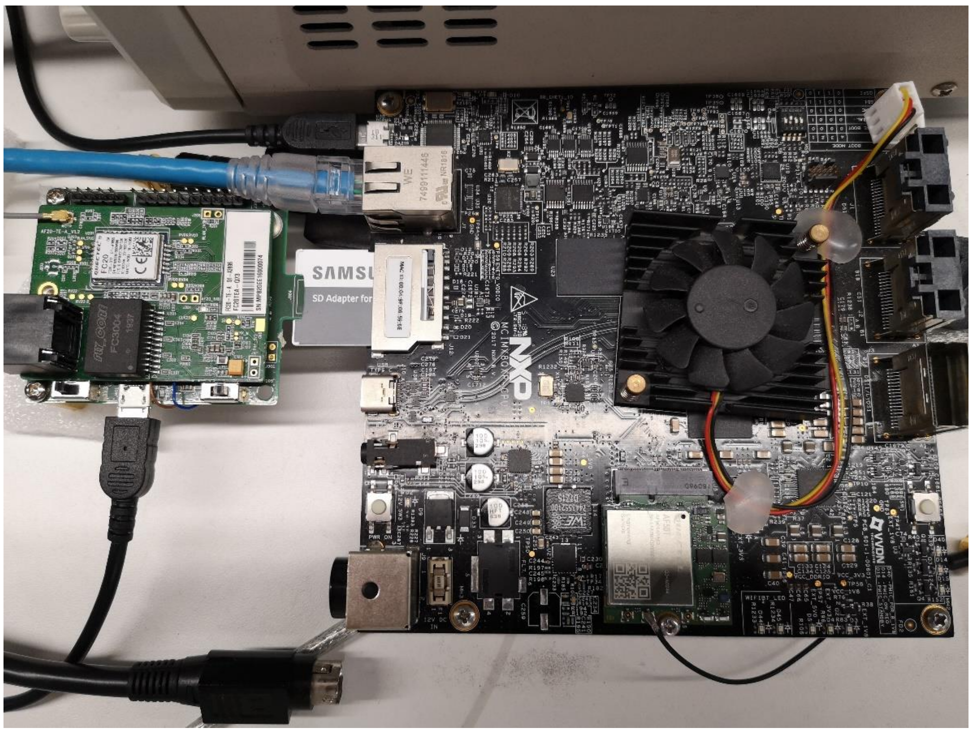


Figure 1: Hardware Environment for Wi-Fi Enablement

2.1.2. Software Environment

Recommended kernel version: 4.14.78
IMX Yocto environment downloading address (Git address):

```
git clone https://source.codeaurora.org/external/imx/linux-imx.git --branch imx_4.14.78_1.0.0_ga
```

2.1.2.1. Install IMX8QXP Tool Chain

Execute the following command to install the IMX8QXP tool chain on host.

```
$ ./fsl-imx-xwayland-glibc-x86_64-meta-toolchain-aarch64-toolchain-4.14-sumo.sh // Install toolchain
    NXP i.MX Release Distro SDK installer version 4.14-sumo
    =======================================================
    Enter target directory for SDK (default: /opt/fsl-imx-xwayland/4.14-sumo):
/home/qingzong/toolchain/
    You are about to install the SDK to "/home/qingzong/toolchain". Proceed[Y/n]? Y
    Extracting SDK........................done
    Setting it up...done
    SDK has been successfully set up and is ready to be used.
    Each time you wish to use the SDK in a new shell session, you need to source the environment
setup script e.g.
 $ . /home/qingzong/toolchain/environment-setup-aarch64-poky-linux
```

2.1.2.2. Configuring IMX8QXP Tool Chain Environment

After the tool chain is installed successfully, execute **source** to configure the running environment. And the command example is shown below:

```
$ source /home/qingzong/toolchain/environment-setup-aarch64-poky-linux
```

NOTE

The command above should be adjusted to your actual directory used.

2.1.2.3. Compiling IMX8QXP Images

Execute the following command to compile IMX8QXP images:

```
$ cd ${kernel-source}        // Please change to your directory
$ make defconfig
$ make menuconfig
$   LDFLAGS="" CC="$CC" make -j24
```

NOTE

Manually open the following configuration items in *menuconfig* file:

```
CONFIG_CFG80211=y
CONFIG_HOSTAP=y
CONFIG_WIRELESS_EXT=y
CONFIG_WEXT_PRIV=y
CONFIG_NL80211_TESTMODE=y
CONFIG_CFG80211_WEXT=y
CONFIG_CFG80211_INTERNAL_REGDB=y
CONFIG_CFG80211_REG_DEBUG=y
CONFIG_BCMDHD=n
CONFIG_RFKILL=y
CONFIG_RFKILL_INPUT=y
CONFIG_RFKILL_REGULATOR=y
CONFIG_RFKILL_GPIO=y
```

2.2. Compile WLAN Driver

2.2.1. Obtain Module SDK

Please contact Quectel Technical Supports for SDK of the relevant module.

2.2.2. Configure IMX8QXP Tool Chain Environment

Configure IMX8QXP tool chain environment, and the command example is shown below:

```
$ source /home/qingzong/toolchain/environment-setup-aarch64-poky-linux
```

NOTE

The command above should be adjusted to your actual directory used.

2.2.3. Configure Compilation Environment

Configure Wi-Fi compilation environment, and the command example is shown below:

```
$ cd <FC2x_target_root>/cnss_host_LEA/chss_proc/host/AIO/build
$ vim scripts/te-f30/config.te-f30            // Modify KERNELPATH, KERNELARCH and TOOLPREFIX
            export KERNELPATH=/home/qingzong/code/kernel-source
            export KERNELARCH=arm64
            export TOOLPREFIX=${CROSS_COMPILE}
$ cd <FC2x_target_root>/cnss_host_LEA/chss_proc/host/AIO/drivers/qcacld-new
$ vim Makefile                                // Modify KERNEL_SRC
      KERNEL_SRC ?= /home/qingzong/code/kernel-source
```

NOTE

<FC2x_target_root> is the root directory that stores the decompressed SDK.

2.2.4. Compile WLAN Driver

Compile WLAN driver, and the command example is shown below:

```
$ cd <FC2x_target_root>/cnss_host_LEA/chss_proc/host/AIO/build
$ make drivers
```

NOTE

<FC2x_target_root> is the root directory that stores the decompressed SDK.

2.3. Install WLAN Firmware and Driver

Install the WLAN firmware and driver to IMX8QXP demo board according to the following steps.

1. The WLAN firmware is in *<FC2x_target_root>/meta_build/load_meta/wlan_firmware/sdio*. Copy the binary files (*bdwlan30.bin*, *otp30.bin*, *qwlan30.bin* and *utf30.bin*) to */lib/firmware/* of IMX8QXP demo

board with a USB flash disk, and the command example is shown below:

```
$ ls <FC2x_target_root>/meta_build/load_meta/wlan_firmware/sdio/
```

2. WLAN driver configuration file is in *<FC2x_target_root>/meta_build/load_meta/host/wlan_host/sdio*. Copy the configuration file *qcom_cfg.ini* to */lib/firmware/wlan/* of IMX8QXP demo board (please create the directory if the target directory */lib/firmware/wlan/* does not exist) with a USB flash disk, and the command example is shown below:

```
$ ls <FC2x_target_root>/meta_build/load_meta/host/wlan_host/sdio/qcom_cfg.ini
```

3. The .ko file of the WLAN driver is in *<FC2x_target_root>/cnss_host_LEA/chss_proc/host/AIO/drivers/qcacld-new/*. Copy file *wlan.ko* to */lib/modules/$(echo $(uname –r))/extra* of the IMX8QXP demo board with a USB flash disk, and the command example is shown below:

```
$ ls <FC2x_target_root>/cnss_host_LEA/chss_proc/host/AIO/drivers/qcacld-new/
```

NOTE

<FC2x_target_root> is the root directory that stores the decompressed SDK.

2.4. Load WLAN Driver

2.4.1. Check SDIO Enumeration

Check if SDIO can be enumerated successfully, and the command example is shown below:

```
root@imx8qxpmek:~# dmesg | grep mmc1
[      1.947683] mmc1: CQHCI version 5.10
[      2.007212] mmc1: SDHCI controller on 5b020000.usdhc [5b020000.usdhc] using ADMA
[      2.067920] mmc1: queuing unknown CIS tuple 0x01 (3 bytes)
[      2.075785] mmc1: queuing unknown CIS tuple 0x1a (5 bytes)
[      2.079252] mmc1: queuing unknown CIS tuple 0x1b (8 bytes)
[      2.079946] mmc1: queuing unknown CIS tuple 0x14 (0 bytes)
[      2.136241] mmc1: queuing unknown CIS tuple 0x80 (1 bytes)
[      3.484201] mmc1: queuing unknown CIS tuple 0x81 (1 bytes)
[      3.489830] mmc1: queuing unknown CIS tuple 0x82 (1 bytes)
[      3.495424] mmc1: new ultra high speed SDR104 SDIO card at address 0001
```

2.4.2. Load WLAN Driver

Execute **insmod** to load WLAN driver, and the command example is shown below:

```
# insmod wlan.ko
```

NOTE

insmod ./wlan.ko country_code=US is used to configure the country code.

2.4.3. Enable Wi-Fi

Enable Wi-Fi, and the command example is shown below:

```
$ ifconfig wlan0 up
$ ifconfig wlan0
wlan0       Link encap:Ethernet   HWaddr 00:03:7f:10:72:12
            UP BROADCAST MULTICAST   MTU:1500   Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:3000
            RX bytes:0 (0.0 B)   TX bytes:0 (0.0 B)
```

2.5. Function Verification

The Wi-Fi working modes of Quectel FC20, FC21 and FC900E modules are STA mode and AP mode. This chapter introduces the function verification methods for both the two modes.

2.5.1. STA Mode

STA (Station), namely the station, is generally the client in the wireless local area network. Please follow the steps below to verify the Wi-Fi function of the module when the Wi-Fi working mode is STA mode.

2.5.1.1. Connect to AP in Open Mode

1. Set an AP (e.g., encryption mode: WPA-PSK; SSID: SoftAP; IP address: 192.168.1.1) and enable the DHCP server of the AP.

2. Modify the configuration file *wpa-sta.conf*, and the command example is shown below:

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
network={
ssid="SoftAP"
key_mgmt=NONE
}
```

3. Connect the module to AP, and the command example is shown below:

```
$ wpa_supplicant -D nl80211 -i wlan0 -c wpa-sta.conf &
```

4. Set IP address of wlan0 after connecting the module to AP successfully, and the command example is shown below:

```
$ ifconfig wlan0 192.168.1.2
```

5. Check the reachability of the IP address with **ping**, and the command example is shown below:

```
$ ping 192.168.1.1
```

6. If the IP address can be pinged successfully, it indicates the module works normally in STA mode.

2.5.1.2. Connect AP in Encryption Mode

1. Set an AP (e.g., encryption mode: WPA-PSK; SSID: SoftAP; IP address: 192.168.1.1), and enable the DHCP server of the AP.

2. Modify the configuration file *wpa-sta.conf*, and the command example is shown below:

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
network={
ssid="SoftAP"
key_mgmt=WPA-PSK
auth_alg=OPEN
pairwise=CCMP
group=CCMP
psk="1234567890"
}
```

3. Connect the module to AP, and the command example is shown below:

```
$ wpa_supplicant -Dnl80211 -iwlan0 -c wpa-sta.conf &
```

4. Set IP address of wlan0 after connecting the module to AP successfully, and the command example is shown below:

```
$ ifconfig wlan0 192.168.1.2
```

5. Check the reachability of the IP address with **ping**, and the command example is shown below:

```
$ ping 192.168.1.1
```

6. If the IP address can be pinged successfully, it indicates the module works normally in STA mode.

2.5.2. AP Mode

AP (Access Point) refers to the wireless access point, which is the creator of a wireless network and the central node of the network. The host has to install the programs hostapd and DHCP to configure wireless access point. Hostapd is a server end program running in user space and provides hotspot access and authentication. DHCP assigns IP address for the devices accessed through the wireless access point.

The configuration files *hostapd-2G.conf* and *hostapd-5G.conf* are located at *<FC2x_target_root>/meta_build/load_meta/hostapd/*.

The following steps show how to test the Wi-Fi feature of FC20, FC21 and FC900E modules when the Wi-Fi mode is AP mode with different wireless LAN standards (11bgn and 11ac).

2.5.2.1. Open Mode (11bgn)

1. Modify the default configuration file *hostapd.conf*, and the command example is shown below:

```
interface=wlan0
ssid=SoftAP
hw_mode=g
channel=1
auth_algs=1
ieee80211n=1
```

2. Run hostapd, and the command example is shown below:

```
# hostapd -dd hostapd.conf &
```

3. Set IP address of the SoftAP, and the command example is shown below:

```
#ifconfig wlan0 192.168.11.1
```

4. Connect to the unencrypted AP with other Wi-Fi device that is in STA mode, and execute **ping** to check the reachability of the IP address.

2.5.2.2. Encryption Mode (11bgn)

1. Modify the configuration file *hostapd.conf*, and the command example is shown below:

```
interface=wlan0
ssid=SoftAP
hw_mode=g
channel=1
auth_algs=3
ieee80211n=1
wpa=3
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
rsn_pairwise=CCMP
```

2. Run hostapd, and the command example is shown below:

```
#hostapd -dd hostapd.conf &
```

3. Set IP address of the SoftAP, and the command example is shown below:

```
#ifconfig wlan0 192.168.11.1
```

4. Connect other Wi-Fi devices that are in STA mode to the AP encrypted by WPA1/2 PSK, and execute **ping** to check the reachability of the IP address.

2.5.2.3. Open Mode (11ac)

1. Modify the default configuration file *hostapd.conf* and store it to */home/root/sbin* with hostapd program, and the command example is shown below:

```
interface=wlan0
ssid=SoftAP
hw_mode=a
```

```
channel=149
auth_algs=1
ieee80211n=1
ieee80211ac=1
```

2. Run hostapd, and the command example is shown below:

```
# hostapd -dd hostapd.conf &
```

3. Set IP address of the SoftAP, and the command example is shown below:

```
#ifconfig wlan0 192.168.11.1
```

4. Connect other Wi-Fi devices that are in STA mode to the unencrypted AP, and execute **ping** to check the reachability of the IP address.

2.5.2.4. Encryption Mode (11ac)

1. Modify the configuration file *hostapd.conf*, and the command example is shown below:

```
interface=wlan0
ssid=SoftAP
hw_mode=a
channel=149
ieee80211n=1
ieee80211ac=1
wpa=3
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
rsn_pairwise=CCMP
```

2. Run hostapd, and the command example is shown below:

```
#hostapd -dd hostapd.conf &
```

3. Set IP address of the SoftAP, and the command example is shown below:

```
#ifconfig wlan0 192.168.11.1
```

4. Connect to the AP encrypted by WPA1/2 PSK using other Wi-Fi devices that are in STA mode, and execute **ping** to check the reachability of the IP address.

2.6. Factory Test Mode

2.6.1. Configure Factory Test Mode with QRCT

This chapter describes steps to perform factory test of Wi-Fi feature and the use of QRCT.

The Qcmbr (QDART Connectivity for non-MSMTM Based Resources) application provides the communication path between QDART tool and on-chip firmware. It connects QDART tool via a TCP/IP socket, thus the target board should be in the same LAN with host PC.

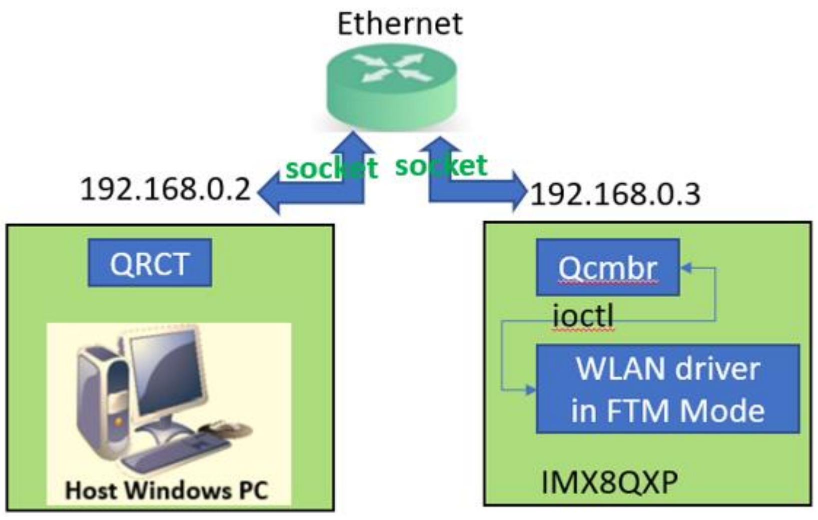


Figure 2: Connection between QDART and Qcmbr

1. Connect the IMX8QXP to PC via an Ethernet cable.
2. Set IP address for IMX8QXP and PC, and make sure PC can ping IMX8QXP successfully.
3. Set the factory test mode and start Wi-Fi, and save the IP address of eth0 properly for subsequent use. The command example is shown below:

```
# insmod /lib/modules/wlan.ko con_mode=5
# ifconfig wlan0 up
# Qcmbr -v
```

2.6.1.1. Run and Configure Tool

1. Run QRCT.
2. Click "**Tool**" → "**User Defined Transport**" → "**OK**" and confirm the prompt message.

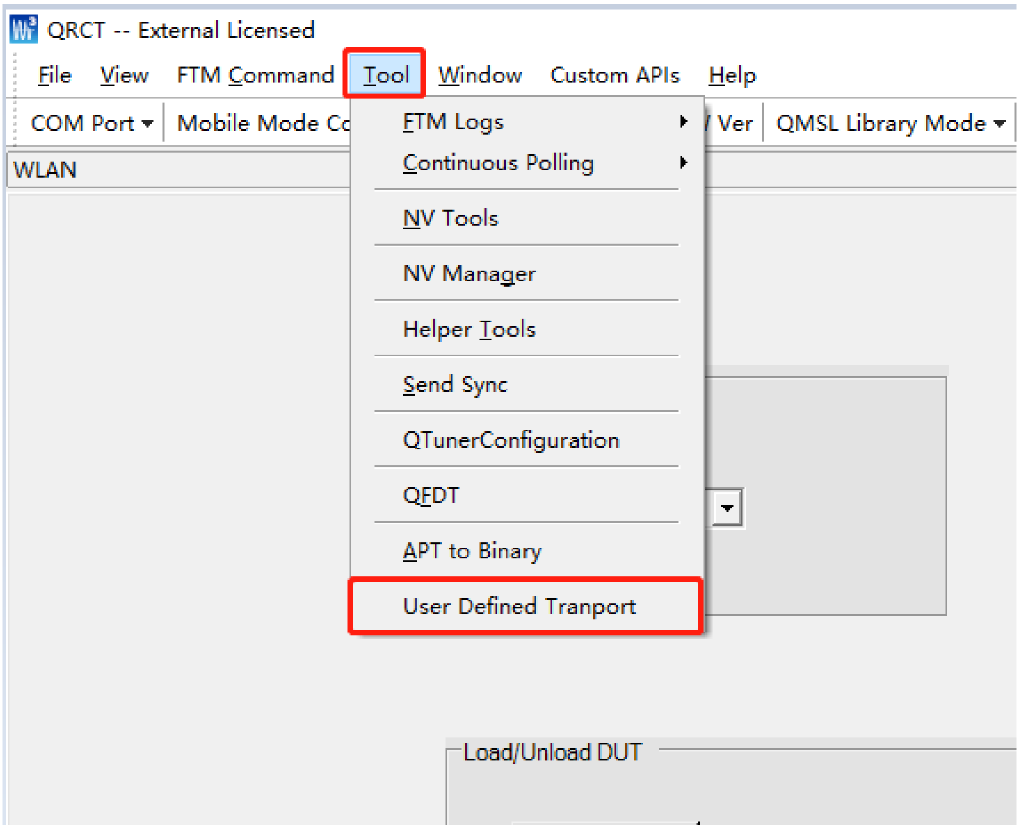
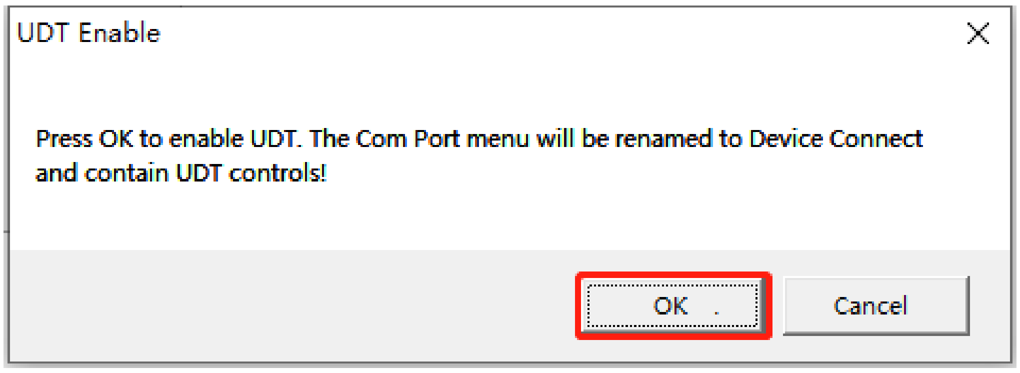


Figure 3: Click "User Defined Transport"


UDT Enable

Press OK to enable UDT. The Com Port menu will be renamed to Device Connect and contain UDT controls!

Figure 4: Confirm Enabling UDT

3. Continue to click "**Device Connect**" → "*** Add New UDT**".

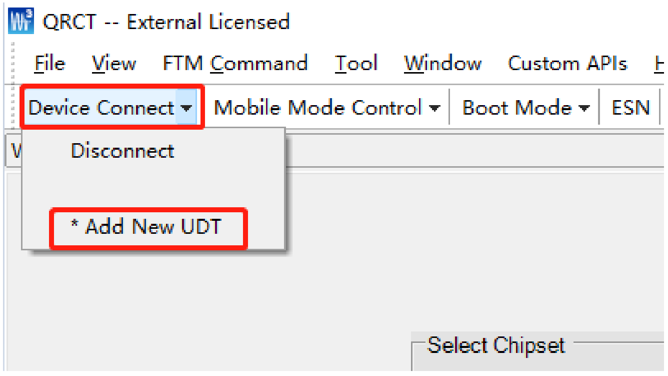
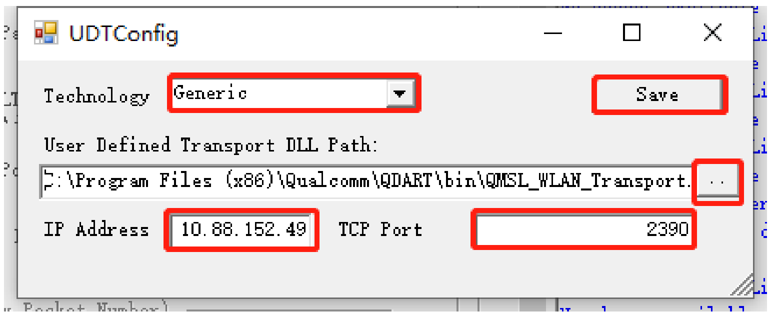


Figure 5: Add New UDT (WLAN)

In the popup box, select "Generic" from "Technology", select *C:\Program Files (x86)\Qualcomm\QDART\bin\QMSL_WLAN_Transport.dll* (the directory is created on the host automatically after installing QRCT. Please contact Quectel Technical Support if the directory does not exist) to "User Defined Transport DLL Path", and input the IP address of eth0 previously saved to "IP Address" and "TCP Port". Then click "**Save**", as shown in the following figure:



Technology: Generic
User Defined Transport DLL Path: C:\Program Files (x86)\Qualcomm\QDART\bin\QMSL_WLAN_Transport.
IP Address: 10.88.152.49
TCP Port: 2390

Figure 6: Configure UDT (WLAN)

4. After the configurations are saved, click "**Device Connect**" → "**10.88.152.49:2390(UDT Generic)**" to disconnect the connection.

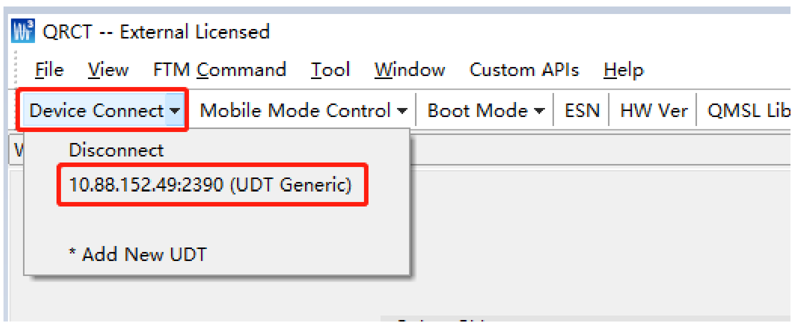

QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect
Mobile Mode Control
Boot Mode
ESN
HW Ver
QMSL Lib
Disconnect
10.88.152.49:2390 (UDT Generic)
* Add New UDT

Figure 7: Disconnect the Connection (WLAN)

NOTE

The steps described in this chapter must be performed, otherwise it may cause the DUT to fail to load. These steps are also required when reopening the tool.

2.6.1.2. Test WLAN

1. Click "**FTM Command**" → "**WLAN**".

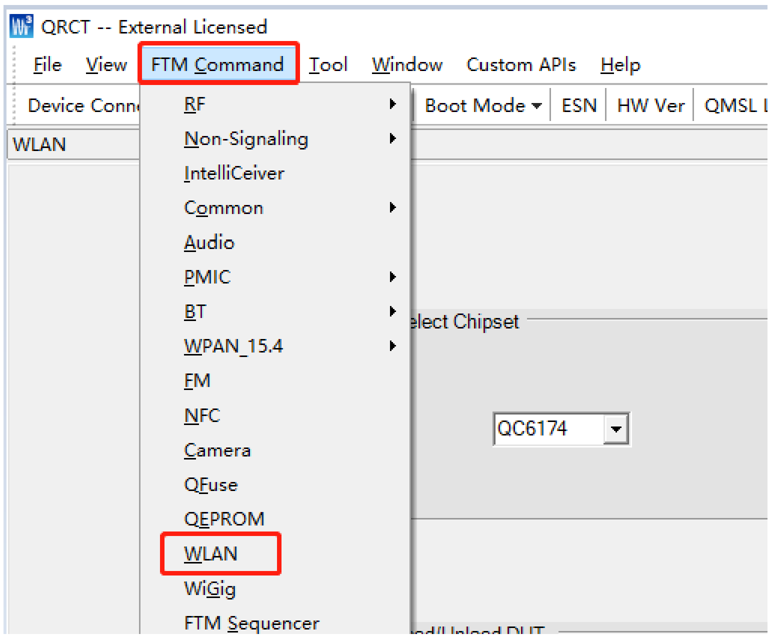
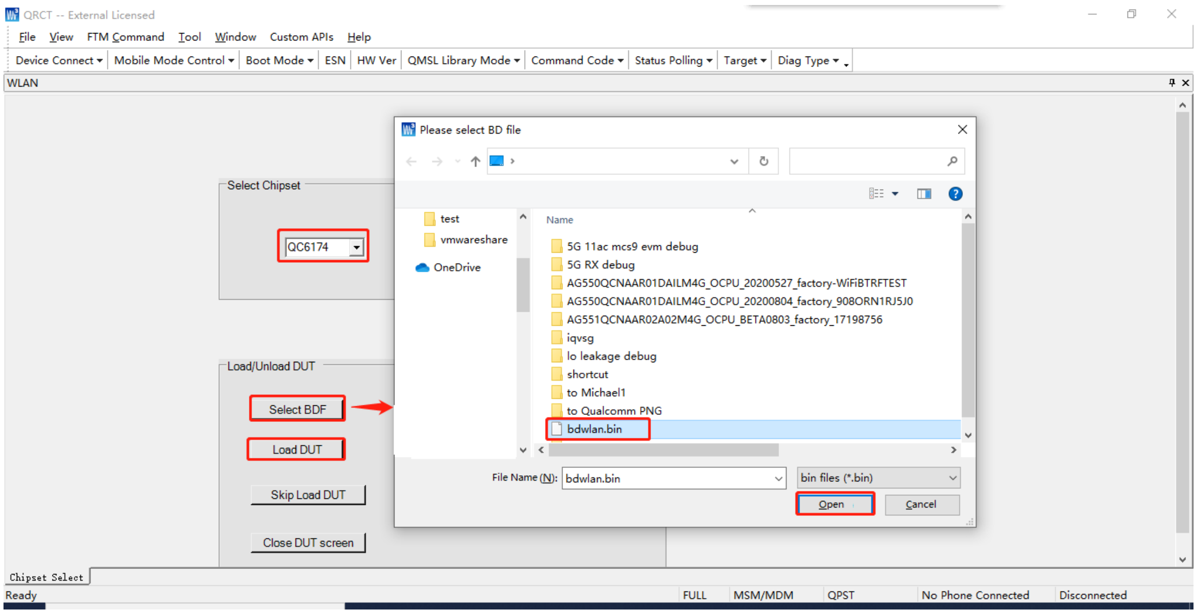


Figure 8: Select WLAN

Then select chipset QC6174, select BDF (*bdwlan.bin*), and click "**Load DUT**" to load DUT, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
WLAN
Load/Unload DUT
Select BDF
Load DUT
Skip Load DUT
Close DUT screen
Open
Cancel
Chipset Select
Ready FULL MSM/MDM QPST No Phone Connected Disconnected

Select Chipset: QC6174

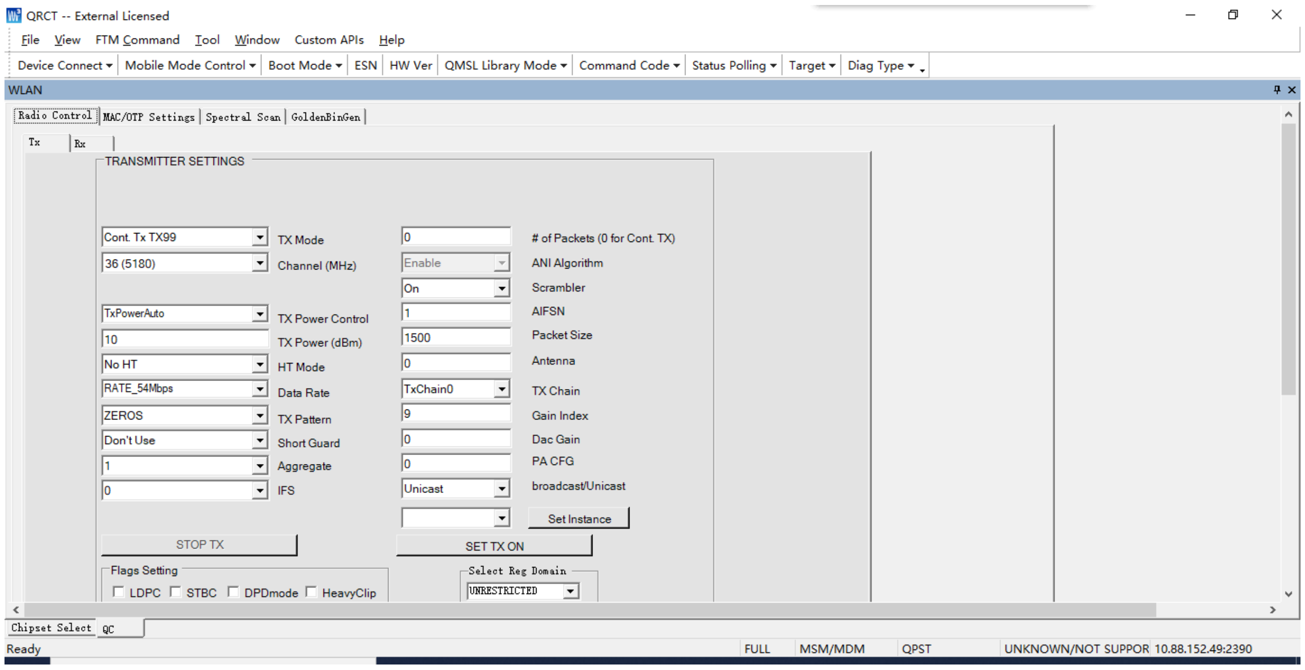
Please select BD file

test
vmwareshare
OneDrive

Name
5G 11ac mcs9 evm debug
5G RX debug
AG550QCNAAR01DAILM4G_OCPU_20200527_factory-WiFiBTRFTEST
AG550QCNAAR01DAILM4G_OCPU_20200804_factory_908ORN1RJ5J0
AG551QCNAAR02A02M4G_OCPU_BETA0803_factory_17198756
iqvsg
lo leakage debug
shortcut
to Michael1
to Qualcomm PNG
bdwlan.bin

File Name (N): bdwlan.bin
bin files (*.bin)

Figure 9: Load DUT


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
WLAN
Radio Control MAC/OTP Settings Spectral Scan GoldenBinGen
Tx Rx
STOP TX
SET TX ON
Set Instance
Chipset Select QC
Ready FULL MSM/MDM QPST UNKNOWN/NOT SUPPOR 10.88.152.49:2390

TRANSMITTER SETTINGS

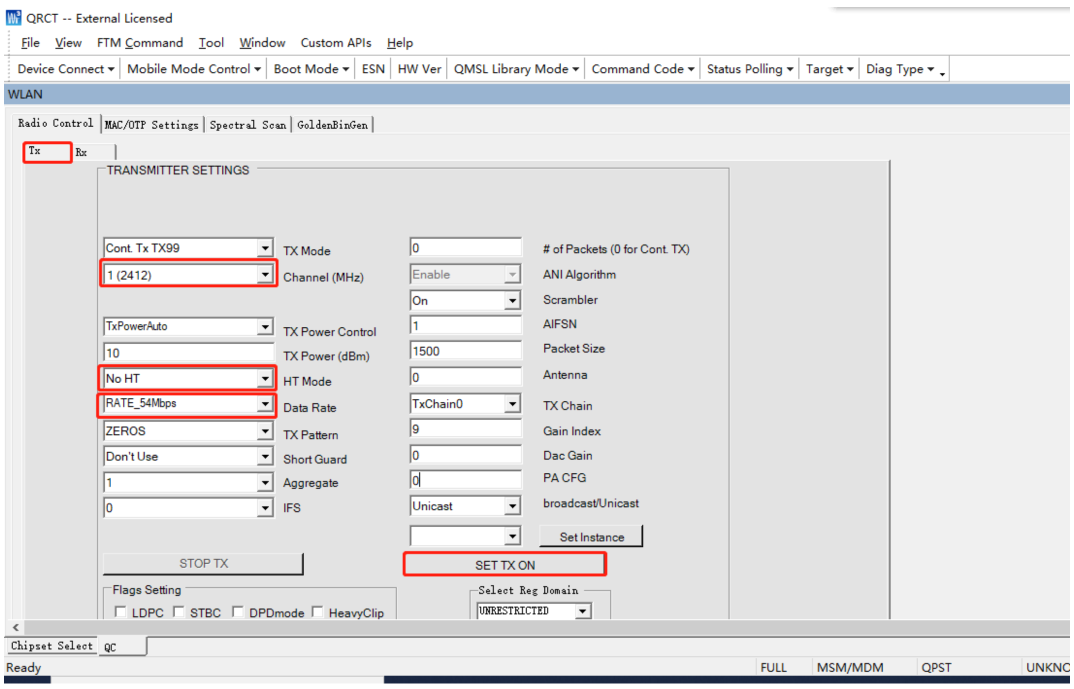
Cont. Tx TX99	TX Mode	0	# of Packets (0 for Cont. TX)
36 (5180)	Channel (MHz)	Enable	ANI Algorithm
		On	Scrambler
TxPowerAuto	TX Power Control	1	AIFSN
10	TX Power (dBm)	1500	Packet Size
No HT	HT Mode	0	Antenna
RATE_54Mbps	Data Rate	TxChain0	TX Chain
ZEROS	TX Pattern	9	Gain Index
Don't Use	Short Guard	0	Dac Gain
1	Aggregate	0	PA CFG
0	IFS	Unicast	broadcast/Unicast

Flags Setting: ☐ LDPC ☐ STBC ☐ DPDmode ☐ HeavyClip

Select Reg Domain: UNRESTRICTED

Figure 10: DUT Loaded Successfully

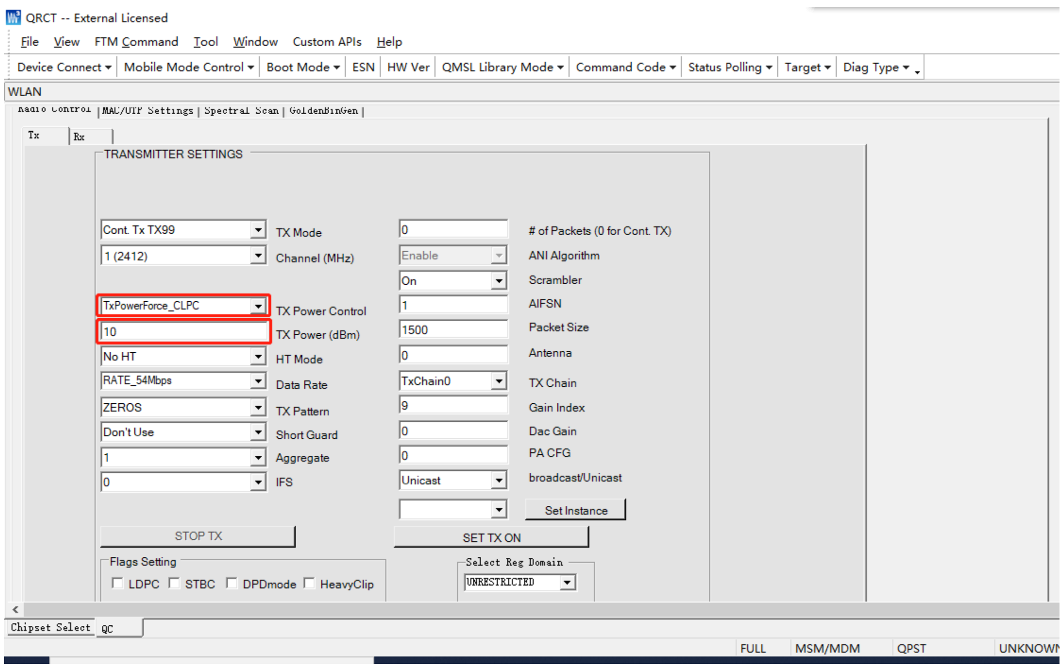
2. In “Tx” column, select “Channel”, “HT Mode” and “Data Rate” (the data rate should be the test data) according to your actual case. Then click “**SET TX ON**”, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
WLAN
Radio Control MAC/OTP Settings Spectral Scan GoldenBinGen
Tx Rx
TRANSMITTER SETTINGS
STOP TX
SET TX ON
Set Instance
Flags Setting
LDPC STBC DPDmode HeavyClip
Chipset Select QC
Ready
FULL MSM/MDM QPST UNKNO

Cont. Tx TX99 TX Mode
1 (2412) Channel (MHz)
TxPowerAuto TX Power Control
10 TX Power (dBm)
No HT HT Mode
RATE_54Mbps Data Rate
ZEROS TX Pattern
Don't Use Short Guard
1 Aggregate
0 IFS
0 # of Packets (0 for Cont. TX)
Enable ANI Algorithm
On Scrambler
1 AIFSN
1500 Packet Size
0 Antenna
TxChain0 TX Chain
9 Gain Index
0 Dac Gain
0 PA CFG
Unicast broadcast/Unicast
Select Reg Domain UNRESTRICTED

Figure 11: Set TX Frequency (2.4 GHz)

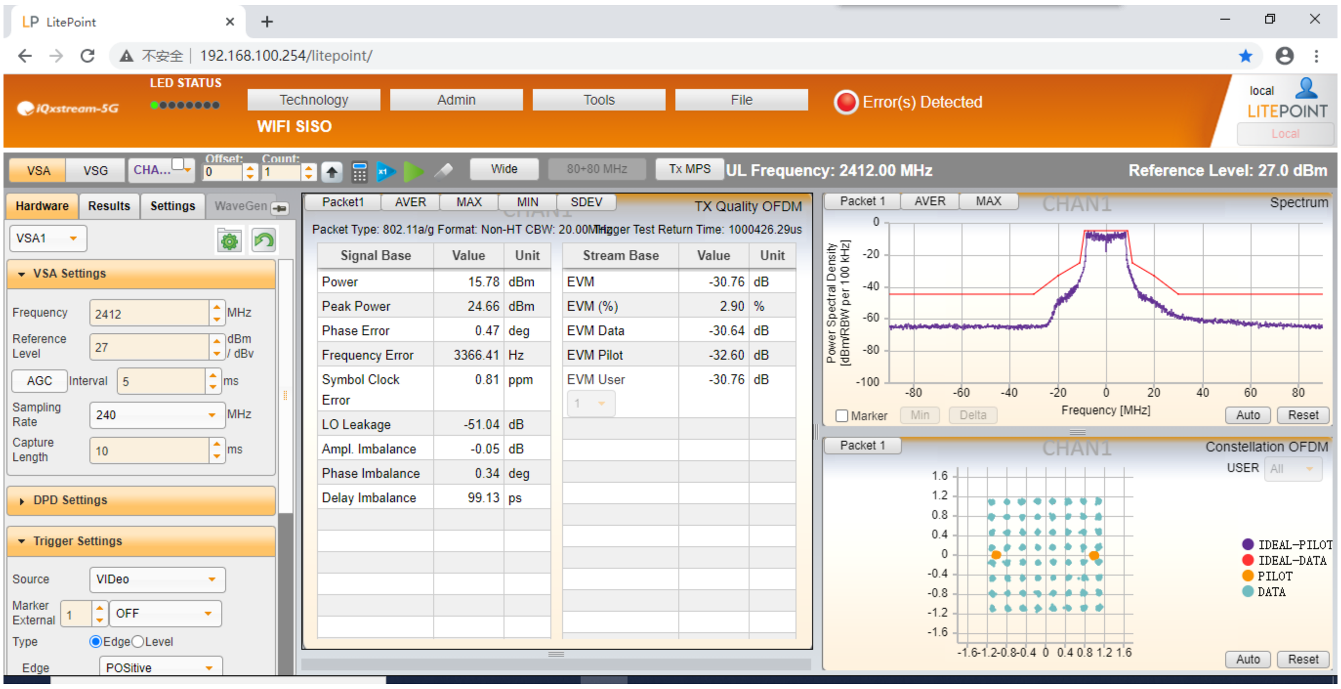
If TX power needs to be changed, please select “TxPowerForce_CLPC”, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
WLAN
Radio Control MAC/OTP Settings Spectral Scan GoldenBinGen
Tx Rx
TRANSMITTER SETTINGS
STOP TX
SET TX ON
Set Instance
Flags Setting
LDPC STBC DPDmode HeavyClip
Chipset Select QC
FULL MSM/MDM QPST UNKNOWN

Cont. Tx TX99 TX Mode
1 (2412) Channel (MHz)
TxPowerForce_CLPC TX Power Control
10 TX Power (dBm)
No HT HT Mode
RATE_54Mbps Data Rate
ZEROS TX Pattern
Don't Use Short Guard
1 Aggregate
0 IFS
0 # of Packets (0 for Cont. TX)
Enable ANI Algorithm
On Scrambler
1 AIFSN
1500 Packet Size
0 Antenna
TxChain0 TX Chain
9 Gain Index
0 Dac Gain
0 PA CFG
Unicast broadcast/Unicast
Select Reg Domain UNRESTRICTED

Figure 12: Change TX Power (2.4 GHz)

After clicking "**SET TX ON**", the RF test device receives and displays the WLAN signal strength, as shown in the following figure:



UL Frequency: 2412.00 MHz Reference Level: 27.0 dBm

VSA Settings: Frequency 2412 MHz; Reference Level 27 dBm / dBv; AGC Interval 5 ms; Sampling Rate 240 MHz; Capture Length 10 ms

Trigger Settings: Source VIDeo; Marker External 1 OFF; Type Edge; Edge POSitive

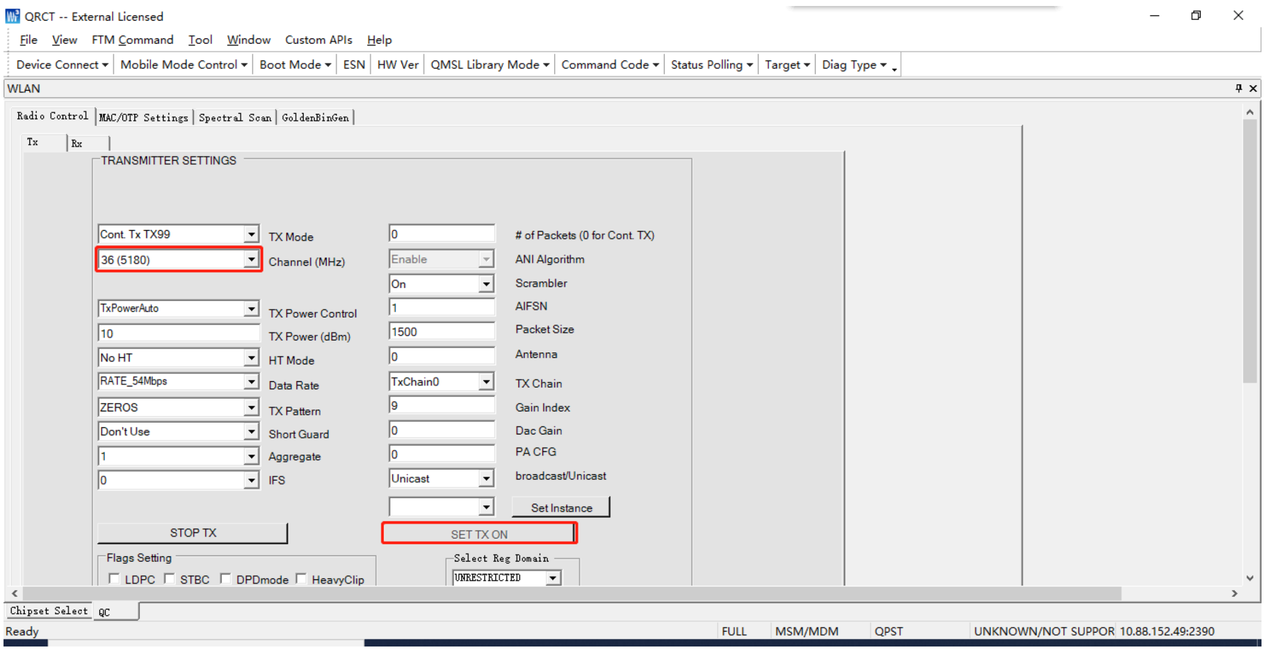
TX Quality OFDM

Packet Type: 802.11a/g Format: Non-HT CBW: 20.00MHz Trigger Test Return Time: 1000426.29us

Signal Base	Value	Unit	Stream Base	Value	Unit
Power	15.78	dBm	EVM	-30.76	dB
Peak Power	24.66	dBm	EVM (%)	2.90	%
Phase Error	0.47	deg	EVM Data	-30.64	dB
Frequency Error	3366.41	Hz	EVM Pilot	-32.60	dB
Symbol Clock Error	0.81	ppm	EVM User	-30.76	dB
LO Leakage	-51.04	dB			
Ampl. Imbalance	-0.05	dB			
Phase Imbalance	0.34	deg			
Delay Imbalance	99.13	ps			

Figure 13: WLAN Signal Strength (2.4 GHz)

3. The 5 GHz TX setting is the same as that of the above 2.4 GHz TX test setting, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
WLAN
Radio Control
MAC/OTP Settings
Spectral Scan
GoldenBinGen
Tx
Rx
STOP TX
SET TX ON
Set Instance
Chipset Select
Ready

TRANSMITTER SETTINGS

Cont. Tx TX99	TX Mode	0	# of Packets (0 for Cont. TX)
36 (5180)	Channel (MHz)	Enable	ANI Algorithm
		On	Scrambler
TxPowerAuto	TX Power Control	1	AIFSN
10	TX Power (dBm)	1500	Packet Size
No HT	HT Mode	0	Antenna
RATE_54Mbps	Data Rate	TxChain0	TX Chain
ZEROS	TX Pattern	9	Gain Index
Don't Use	Short Guard	0	Dac Gain
1	Aggregate	0	PA CFG
0	IFS	Unicast	broadcast/Unicast

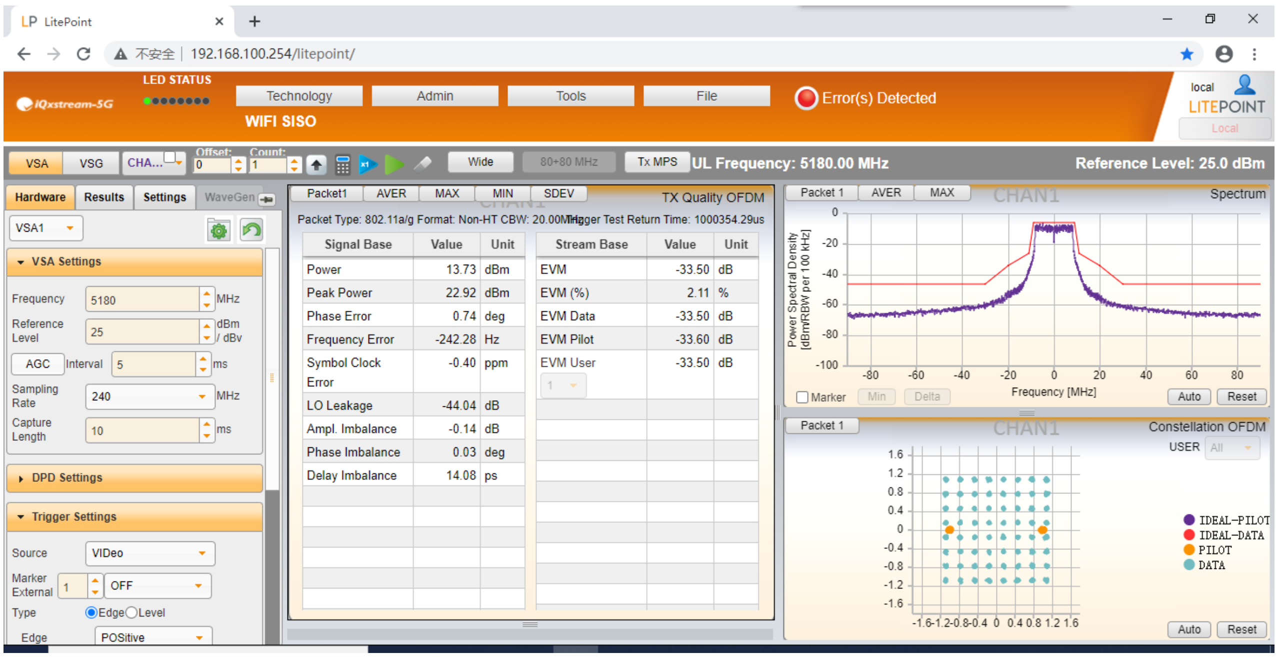
Flags Setting: ☐ LDPC ☐ STBC ☐ DPDmode ☐ HeavyClip

Select Reg Domain: UNRESTRICTED

Chipset Select: QC

FULL MSM/MDM QPST UNKNOWN/NOT SUPPOR 10.88.152.49:2390

Figure 14: Set TX Frequency (5 GHz)



VSA1

VSA Settings

Frequency	5180 MHz
Reference Level	25 dBm / dBv
AGC Interval	5 ms
Sampling Rate	240 MHz
Capture Length	10 ms

DPD Settings

Trigger Settings

Source	VIDeo
Marker External	1 OFF
Type	Edge / Level
Edge	POSitive

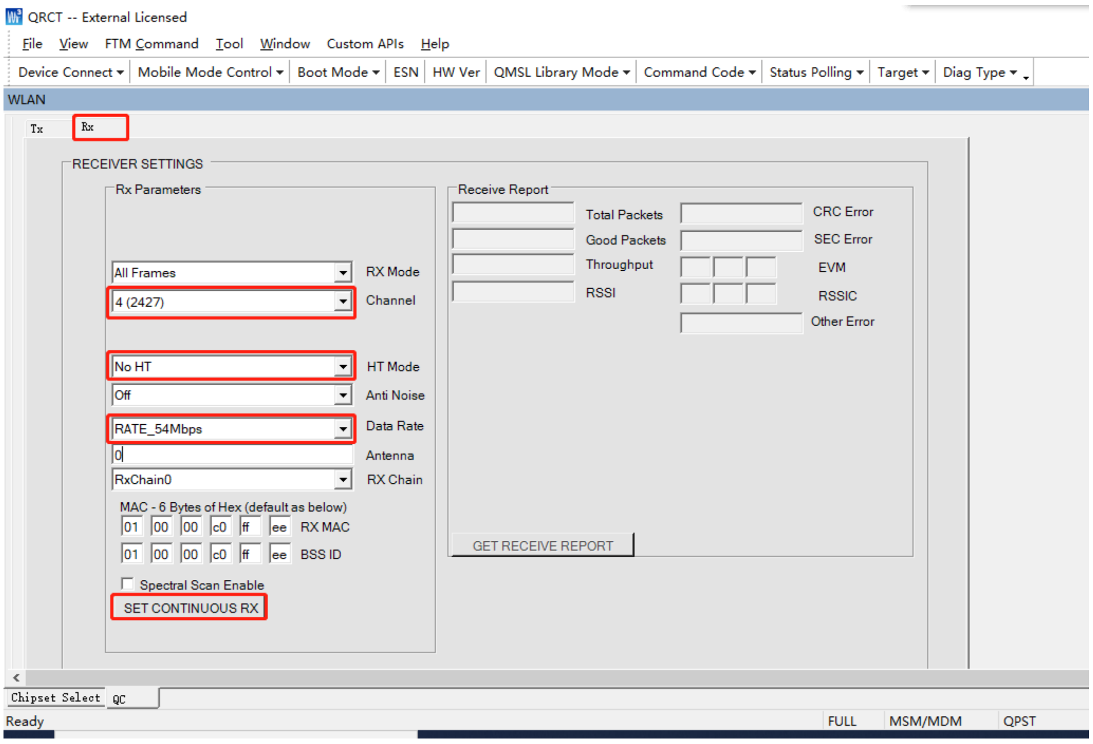
TX Quality OFDM

Packet Type: 802.11a/g Format: Non-HT CBW: 20.00MHz Trigger Test Return Time: 1000354.29us

Signal Base	Value	Unit	Stream Base	Value	Unit
Power	13.73	dBm	EVM	-33.50	dB
Peak Power	22.92	dBm	EVM (%)	2.11	%
Phase Error	0.74	deg	EVM Data	-33.50	dB
Frequency Error	-242.28	Hz	EVM Pilot	-33.60	dB
Symbol Clock Error	-0.40	ppm	EVM User	-33.50	dB
LO Leakage	-44.04	dB			
Ampl. Imbalance	-0.14	dB			
Phase Imbalance	0.03	deg			
Delay Imbalance	14.08	ps			

Figure 15: WLAN Signal Strength (5 GHz)

4. In “Rx” column, select “Channel”, “HT Mode” and “Data Rate” (the data rate should be the test data) according to the actual practice, check “SET CONTINUOUS RX”, and then the RF test device starts sending signal automatically, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
WLAN
Tx
Rx
RECEIVER SETTINGS
Rx Parameters
Receive Report
Total Packets
Good Packets
Throughput
RSSI
CRC Error
SEC Error
EVM
RSSIC
Other Error
GET RECEIVE REPORT
Chipset Select
QC
Ready
FULL
MSM/MDM
QPST

All Frames	RX Mode
4 (2427)	Channel
No HT	HT Mode
Off	Anti Noise
RATE_54Mbps	Data Rate
0	Antenna
RxChain0	RX Chain

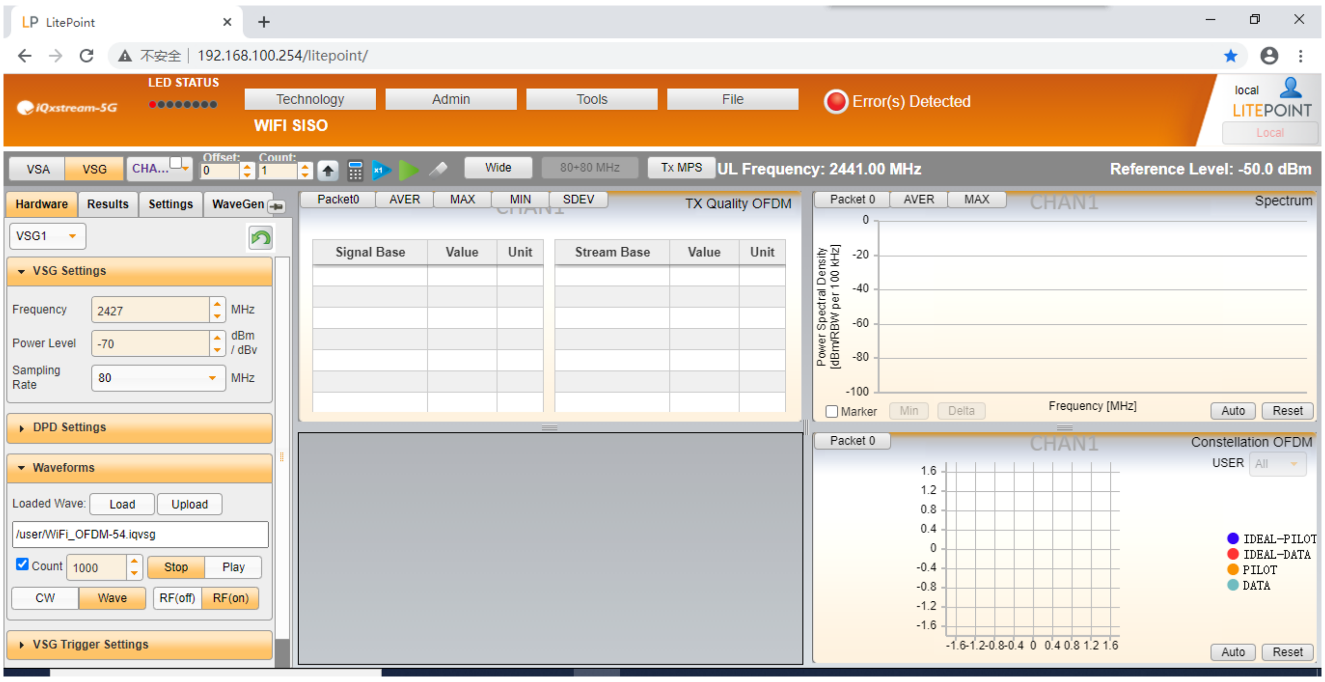
MAC - 6 Bytes of Hex (default as below)

01	00	00	c0	ff	ee	RX MAC
01	00	00	c0	ff	ee	BSS ID

Spectral Scan Enable

SET CONTINUOUS RX

Figure 16: Set RX Frequency (2.4 GHz)



UL Frequency: 2441.00 MHz

Reference Level: -50.0 dBm

VSG1

VSG Settings

Frequency	2427	MHz
Power Level	-70	dBm / dBv
Sampling Rate	80	MHz

DPD Settings

Waveforms

Loaded Wave: Load Upload

/user/WiFi_OFDM-54.iqvsg

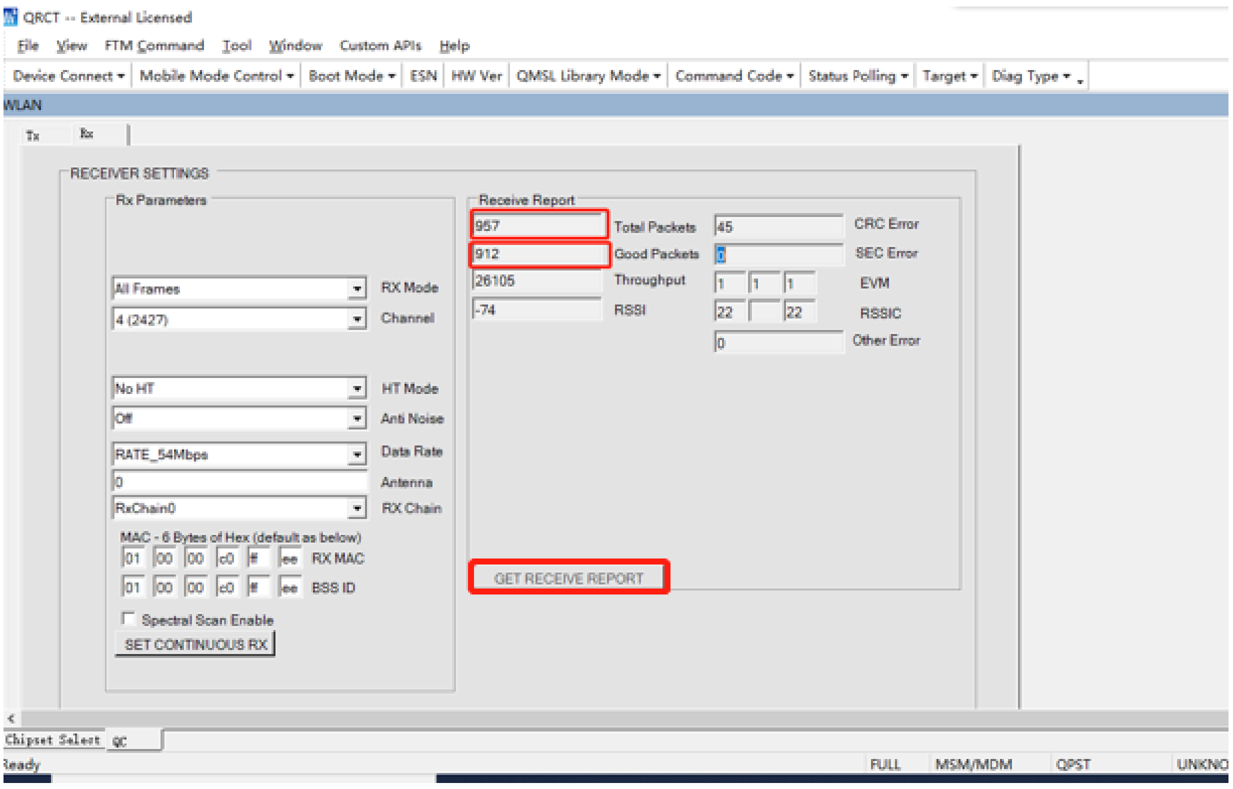
Count 1000 Stop Play

CW Wave RF(off) RF(on)

VSG Trigger Settings

Figure 17: Device Sends Signal (2.4 GHz)

Click “**GET RECEIVE REPORT**” on the QRCT tool to check the received data, as shown in the following figure:



RECEIVER SETTINGS

Rx Parameters

All Frames	RX Mode
4 (2427)	Channel
No HT	HT Mode
Off	Anti Noise
RATE_54Mbps	Data Rate
0	Antenna
RxChain0	RX Chain

MAC - 6 Bytes of Hex (default as below)

01	00	00	c0	ff	ee	RX MAC
01	00	00	c0	ff	ee	BSS ID

Spectral Scan Enable

SET CONTINUOUS RX

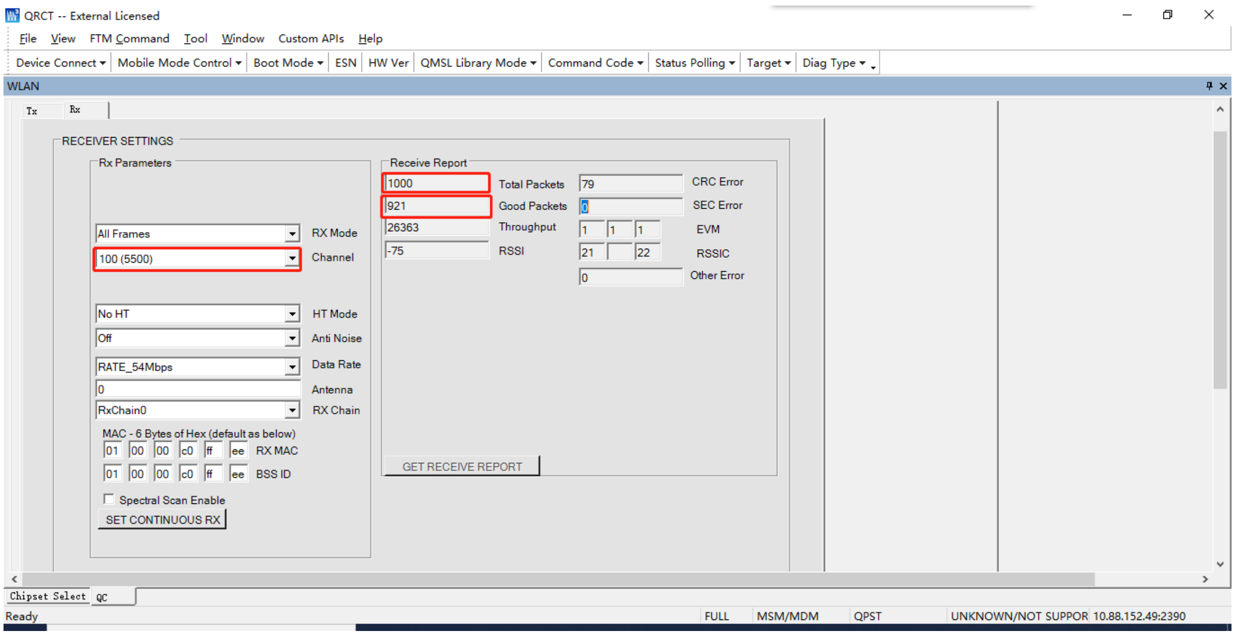
Receive Report

957	Total Packets	45			CRC Error
912	Good Packets	0			SEC Error
26105	Throughput	1	1	1	EVM
-74	RSSI	22		22	RSSIC
		0			Other Error

GET RECEIVE REPORT

Figure 18: Check Received Data (2.4 GHz)

5. The 5 GHz RX setting is the same as that of the above 2.4 GHz RX test setting, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
WLAN
Tx Rx
RECEIVER SETTINGS
Rx Parameters
Receive Report
Spectral Scan Enable
SET CONTINUOUS RX
GET RECEIVE REPORT
Chipset Select QC

RX Mode: All Frames
Channel: 100 (5500)
HT Mode: No HT
Anti Noise: Off
Data Rate: RATE_54Mbps
Antenna: 0
RX Chain: RxChain0
MAC - 6 Bytes of Hex (default as below)
RX MAC: 01 00 00 c0 ff ee
BSS ID: 01 00 00 c0 ff ee

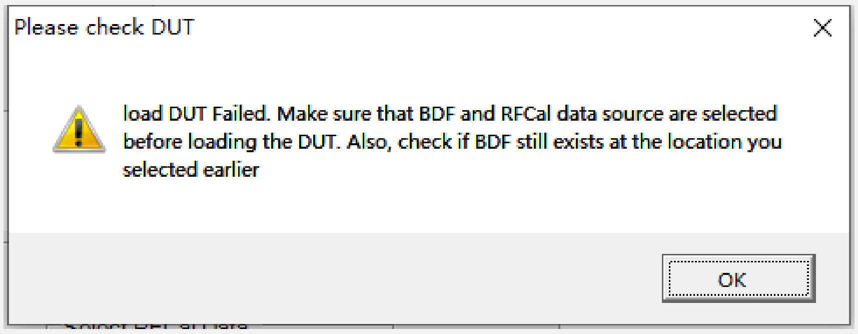
Total Packets: 1000
Good Packets: 921
Throughput: 26363
RSSI: -75
CRC Error: 79
SEC Error: 0
EVM: 1 1 1
RSSIC: 21 22
Other Error: 0

Ready FULL MSM/MDM QPST UNKNOWN/NOT SUPPOR 10.88.152.49:2390

Figure 19: Set RX Frequency (5 GHz)

NOTE

1. If loading DUT fails, input **Qcmbr –v** again after “Remote connection closed” is displayed, and then reopen the QRCT.


Please check DUT

load DUT Failed. Make sure that BDF and RFCal data source are selected before loading the DUT. Also, check if BDF still exists at the location you selected earlier

2. "HT Mode" options:
   No HT: 11g mode;
   CCK: 11b mode;
   HT20: 11n 20M mode;
   HT40: 11n 40M mode;
   VHT20: 11ac 20M mode;
   VHT40: 11ac 40M mode;
   VHT80: 11ac 80M mode.

2.7. Logs Capture

2.7.1. Get WLAN Host and Firmware Version

1. Execute the following command to get WLAN host and firmware version:

```
# iwpriv wlan0 version
```

2. Execute the following command to get the configurations of WLAN driver:

```
# iwpriv wlan0 getConfig
```

2.7.2. Enable WLAN Host Logs

1. Add the following to *qcom_cfg.ini*, reload WLAN module and check the kernel logs.

```
vosTraceEnableTL=255
vosTraceEnableWDI=255
vosTraceEnableHDD=255
vosTraceEnableSME=255
vosTraceEnablePE=255
vosTraceEnablePMC=255
vosTraceEnableWDA=255
vosTraceEnableSYS=255
vosTraceEnableVOSS=255
vosTraceEnableSAP=255
vosTraceEnableHDDSAP=255
vosTraceEnableCFG=255
vosTraceEnableADF=255
vosTraceEnableTXRX=255
vosTraceEnableHTC=255
vosTraceEnableHIF=255
vosTraceEnableHDDSAPDATA=255
vosTraceEnableHDDDATA=255
vosTraceEnableHIF=255
vosTraceEnableHTC=255
vosTraceEnableTXRX=255
vosTraceEnableADF=255
vosTraceEnableCFG=255
```

2. The WLAN logging tool cnss_diag_lite routes WLAN driver verbose messages and firmware debug information to console or saves the logs in a file.

Some of the options of the logging tool options:

-f	saves log to file by text format (stored in folder *wlan_logs* in the same path of the tool)
-c	enables host log (no parameter)
-x	enables firmware diagnose message (0 indicates disablement, 1–31 indicate enablement)
-e	enables firmware diagnose event/log (0 indicates disablement, 1 indicates enablement)
-l	saves log to file in QXDM format (stored in folder *wlan_logs* in the same path of the tool)

NOTE

1. Currently only host logs and firmware diagnose logs could be shown in console window.
2. Only firmware diagnose logs can be saved into file and parsed by the QXDM tool.
3. The above commands can be sorted arbitrarily when they are executed.

Execute the flowing command to enable the host logs and firmware diagnose logs, and save the logs in console files in text format.

```
# cnss_diag_lite -c > host_wlan_log &
```

2.7.3. Enable Module Firmware Debug Logs

1. You can enable WLAN firmware debug logs by modifying *qcom_cfg.ini*:

```
# Enable firmware uart print
gEnablefwprint=0

# Enable firmware log
gEnablefwlog=1
gMulticastHostFwMsgs=1
gFwDebugLogType=255
# Additional firmware log levels
gFwDebugLogLevel=0
gFwDebugModuleLoglevel=1,0,2,0,4,0,5,0,6,0,7,4,8,0,9,0
```

Or executing the following command:

```
#First enable firmware log
gEnablefwlog=1
```

2. Executing the following command to configure the log level:

```
Syntax: iwpriv wlan0 dl_loglevel <0-5>
     Supported log levels
     DBGLOG_VERBOSE = 0,
     DBGLOG_INFO,
     DBGLOG_INFO_LVL_1,
     DBGLOG_INFO_LVL_2,
     DBGLOG_WARN,
     DBGLOG_ERR,
     DBGLOG_LVL_MAX
Syntax: iwpriv wlan0 dl_type <0-3>
     DBGLOG_PROCESS_DEFAULT = 0
     DBGLOG_PROCESS_PRINT_RAW = 1
     DBGLOG_PROCESS_POOL_RAW =2
     DBGLOG_PROCESS_NET_RAW = 3
Syntax: iwpriv wlan0 dl_report <0-1>
     0: report stops
     1: report starts
```

2.7.4. Obtain WLAN Target Firmware Statistic Separately

1. Get the target physical device statistics:

```
# iwpriv wlan0 txrx_fw_stats 1
```

2. Get the receiving reord statistics:

```
# iwpriv wlan0 txrx_fw_stats 2
```

3. Get the receiving rate control statistics:

```
# iwpriv wlan0 txrx_fw_stats 3
```

4. Get the transmitting rate control statistics:

```
# iwpriv wlan0 txrx_fw_stats 6
```

5. Clear firmware statistics corresponding to bit set by <mask>. If <mask>=0x1f, all statistics would be cleared.

```
iwpriv wlan0 txrx_fw_st_rst <mask>
```

2.8. Commands to Fix Rate and Bandwidth

2.8.1. Configure NSS and MCS

1. Configure NSS:

```
iwpriv < interface> nss <nss>
```

<nss>	Number of spatial streams.	
	1	1x1 MCS
	2	2x2 MCS
	3	3x3 MCS

2. Configure MCS:

```
iwpriv < interface> set11ACRates <MCS>
```

<MCS> Modulation and coding scheme. Range: 0–9.

3. For high-throughput rates, control NSS and MCS simultaneously:

```
iwpriv < interface> set11NRates 0x80..0x97 (for HT MCS0..23)
```

2.8.2. Configure CCK Rate

1. Configure rate to 1 Mbps:

```
iwpriv <interface> set11NRates 0x1b1b1b1b
```

2. Configure rate to 2 Mbps:

```
iwpriv <interface> set11NRates 0x1a1a1a1a
```

3. Configure rate to 5.5 Mbps:

```
iwpriv <interface> set11NRates 0x19191919
```

4. Configure rate to 11 Mbps:

```
iwpriv <interface> set11NRates 0x18181818
```

2.8.3. Configure OFDM Rate

1. Configure rate to 6 Mbps:

```
iwpriv <interface> set11NRates 0x0b0b0b0b
```

2. Configure rate to 9 Mbps:

```
iwpriv <interface> set11NRates 0x0f0f0f0f
```

3. Configure rate to 12 Mbps:

```
iwpriv <interface> set11NRates 0x0a0a0a0a
```

4. Configure rate to 18 Mbps:

```
iwpriv <interface> set11NRates 0x0e0e0e0e
```

5. Configure rate to 24 Mbps:

```
iwpriv <interface> set11NRates 0x09090909
```

6. Configure rate to 36 Mbps:

```
iwpriv <interface> set11NRates 0x0d0d0d0d
```

7. Configure rate to 48 Mbps:

```
iwpriv <interface> set11NRates 0x08080808
```

8. Configure rate to 54 Mbps:

```
iwpriv <interface> set11NRates 0x0c0c0c0c
```

2.8.4. Configure Auto Rate Mode

Switch rate mode to auto rate mode:

```
iwpriv <interface> set11NRates 0
```

2.8.5. Control Channel Bandwidth

Restrict the transmission bandwidth for both fixed and auto rates:

```
iwpriv <interface> chwidth <bw_idx>
```

<bw_idx>	Bandwidth.	
	0	20 MHz
	1	40 MHz
	2	80 MHz

3 Enable Bluetooth

3.1. Build Environment

3.1.1. Hardware Environment

1. Prepare the following hardware:

   - Quectel FC20, FC21 or FC900E module (FC21 module is used in the examples below)
   - USB cable
   - Quectel EVB

2. Install the module and EVB as shown below:

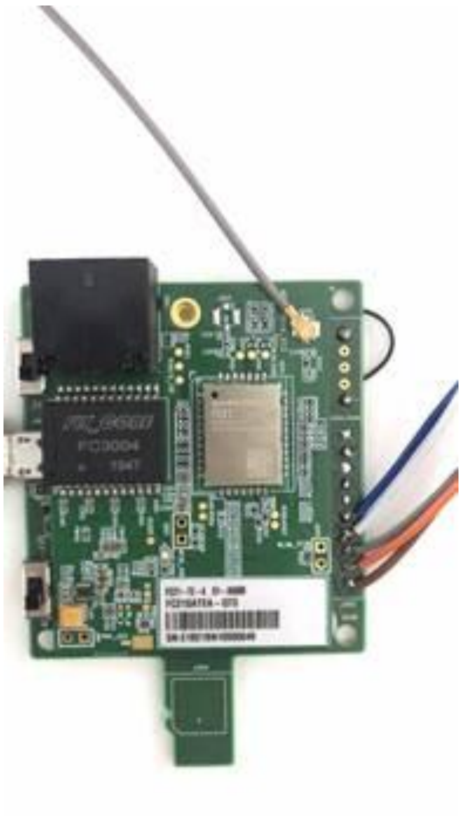


Figure 20: Hardware Environment for Bluetooth

3.1.2. Software Environment

1. Download BlueZ source code.

```
$ git clone git://git.kernel.org/pub/scm/bluetooth/bluez.git
$ git clone git://git.kernel.org/pub/scm/libs/ell/ell.git
```

2. Prepare host environment.

```
$ uname -a
Linux lidong-pc 5.8.0-63-generic #71~20.04.1-Ubuntu SMP Thu Jul 15 17:46:08 UTC 2021 x86_64
x86_64 x86_64 GNU/Linux
```

3. Compile BlueZ source code.

   Apply the patch:

```
$ cd bluez
$ git am 0001-bluetooth-Add-bluetooth-support-for-QCA6174-chip.patch
$ git am 0002-hciattach-set-flag-to-enable-HCI-reset-on-init.patch
$ git am 0003-hciattach-instead-of-strlcpy-with-strncpy-to-avoid-r.patch
$ git am 0004-Add-support-for-Tufello-1.1-SOC.patch
$ git am 0005-bluetooth-Add-support-for-multi-baud-rate.patch
```

   Compile BlueZ:

```
$ ./bootstrap-configure --disable-android --disable-midi
$ make
```

3.2. Function Verification

3.2.1. Download Bluetooth Firmware

Download the Bluetooth firmware to the module, and the command example is shown below:

```
$ sudo ./tools/hciattach /dev/ttyUSB0 qca -t120 3000000 flow
```

3.2.2. Run GATT Server

Run a GATT server of BlueZ stack, and the command example is shown below:

```
$ sudo ./tools/btmgmt -i hci0 name "Quec_FC21"
$ sudo ./tools/btmgmt -i hci0 connectable on
$ sudo ./tools/btmgmt -i hci0 advertising on
$ sudo ./tools/btmgmt -i hci0 le on
$ sudo ./tools/btmgmt -i hci0 power on
$ ./tools/sdptool add --channel=1 GATT SP A2SNK
$ ./tools/gatt-service
```

3.2.3. Scan the Module

Open the nRF Connect App on your phone (central device) to scan the module (peripheral device). And an example is shown below:

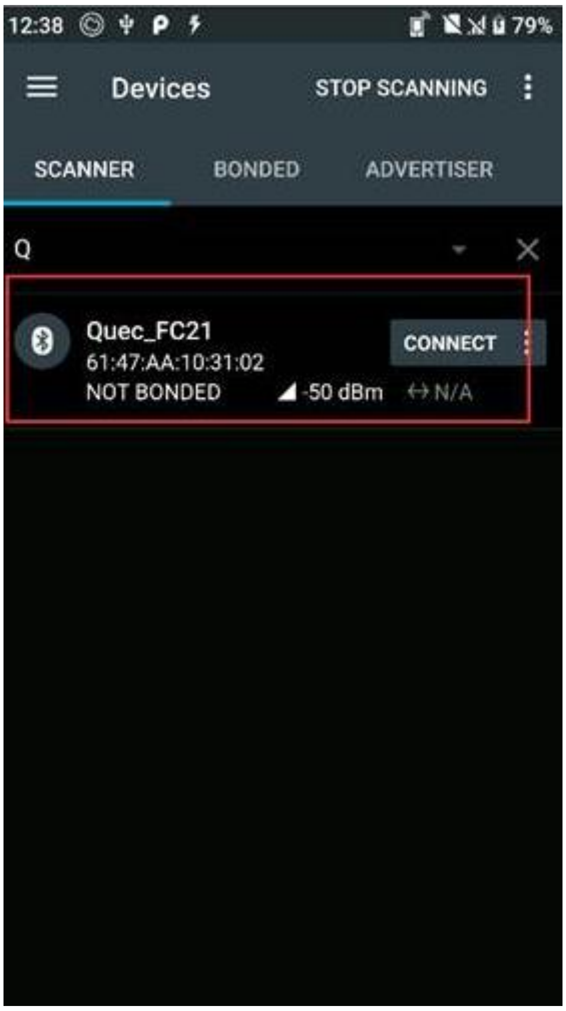


Figure 21: Scan Bluetooth Device with nRF Connect Application

3.3. Factory Test Mode

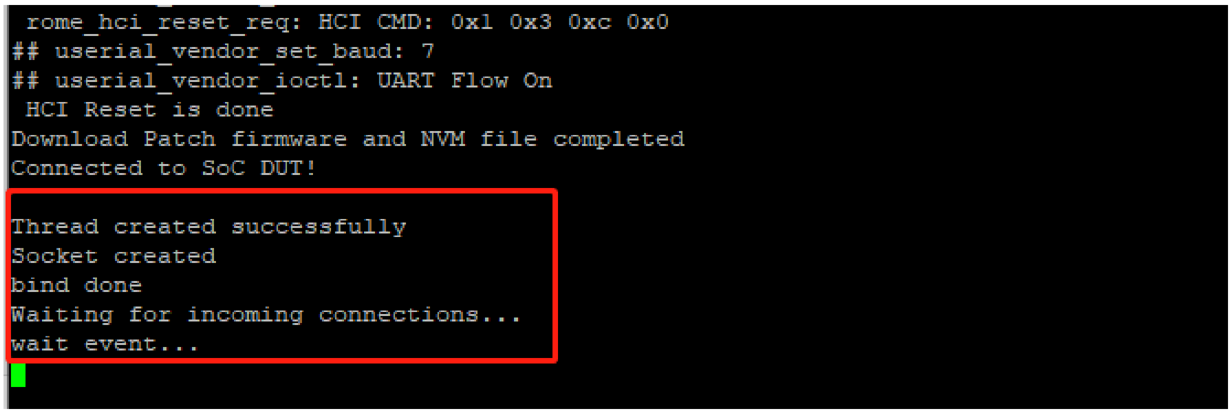
3.3.1. Configure Factory Test Mode with QRCT

This chapter describes steps to perform the factory test of Bluetooth feature and the use of QRCT.

Set the module to factory test mode, and the command example is shown below:

```
#root
# Btdiag UDT=yes PORT=2390 IOType=SERIAL QDARTIOType=ethernet BT-DEVICE=/dev/ttymxc1
BT-BAUDRATE=115200 IPADDRESS=10.88.152.49
```

If the following logs are printed, it indicates that the configuration is completed:



```
 rome_hci_reset_req: HCI CMD: 0x1 0x3 0xc 0x0
## userial_vendor_set_baud: 7
## userial_vendor_ioctl: UART Flow On
 HCI Reset is done
Download Patch firmware and NVM file completed
Connected to SoC DUT!

Thread created successfully
Socket created
bind done
Waiting for incoming connections...
wait event...
```

3.3.2. Run and Configure Tool

1. Run QRCT.

   Click “**Tool**” → “**User Defined Transport**” → “**Yes**” and confirm the prompt message. This step can be skipped if it is performed when testing WLAN

2. Continue to click "**Device Connect**" → "*** Add New UDT**".

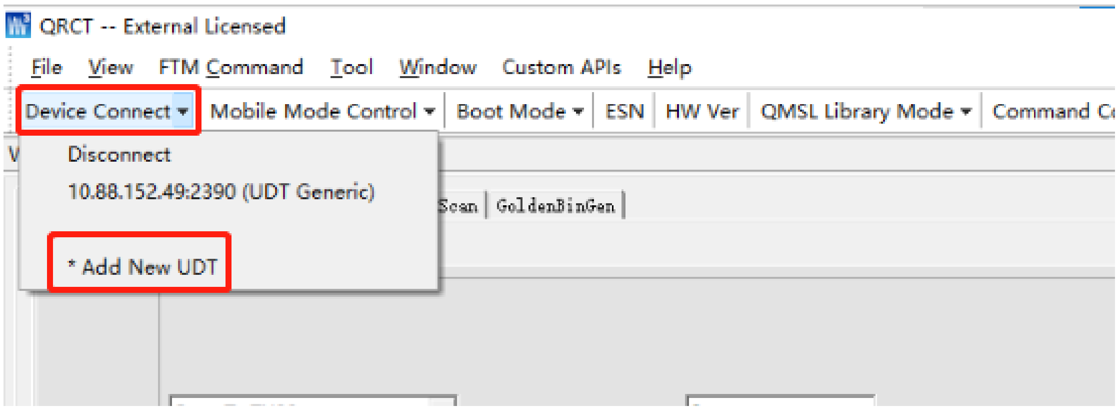
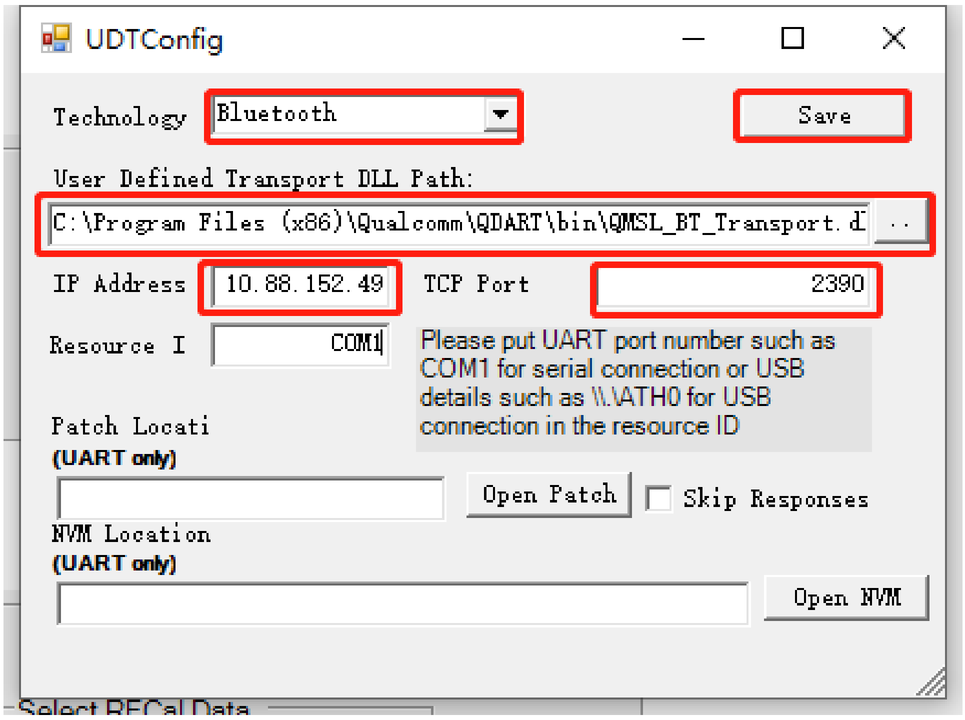


Figure 22: Add New UDT (Bluetooth)

In the popup box, select "Bluetooth" from "Technology", input *C:\Program Files (x86)\Qualcomm\QDART\bin\QMSL_BT_Transport.dll* (the directory is created on the host automatically after installing QRCT. Please contact Quectel Technical Support if the directory does not exist) to "User Defined Transport DLL Path", input the IP address of eth0 to "IP Address", and input the port number to "TCP Port". "ResourceID" is defaulted to COM1, and there is no need to change it. Then click "**Save**", as shown in the following figure:


UDTConfig

Technology: Bluetooth

User Defined Transport DLL Path: C:\Program Files (x86)\Qualcomm\QDART\bin\QMSL_BT_Transport.d

IP Address: 10.88.152.49

TCP Port: 2390

Resource I: COM1

Please put UART port number such as COM1 for serial connection or USB details such as \\.\ATH0 for USB connection in the resource ID

Patch Locati (UART only)

NVM Location (UART only)

Figure 23: Configure UDT (Bluetooth)

3. After saving the configurations, click "**Device Connect**" → "**connectiondetails=COM1…….**".

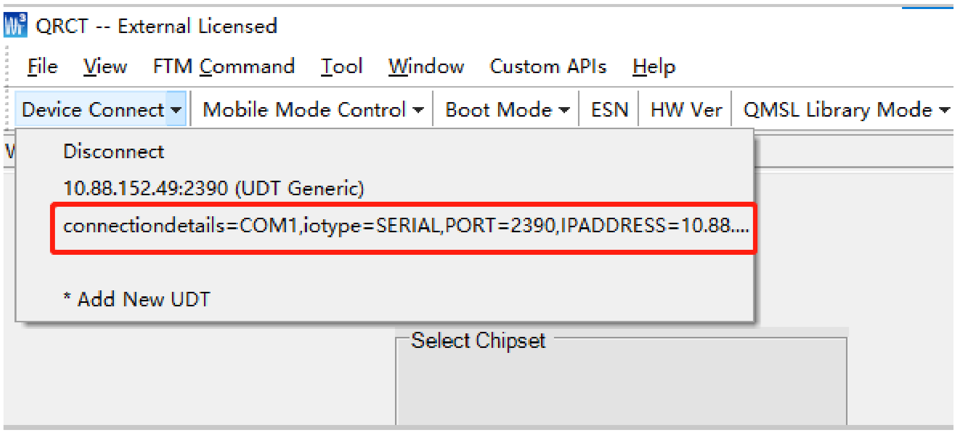

QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode
Disconnect
10.88.152.49:2390 (UDT Generic)
connectiondetails=COM1,iotype=SERIAL,PORT=2390,IPADDRESS=10.88....
* Add New UDT
Select Chipset

Figure 24: Disconnect (Bluetooth)

NOTE

The steps described in this chapter must be performed, otherwise it may cause the DUT to fail to load. These steps are also required when reopening the tool.

3.3.3. Test Bluetooth (BLE)

1. Click "**FTM Command**" → "**BT**" → "**HCI Commands**".

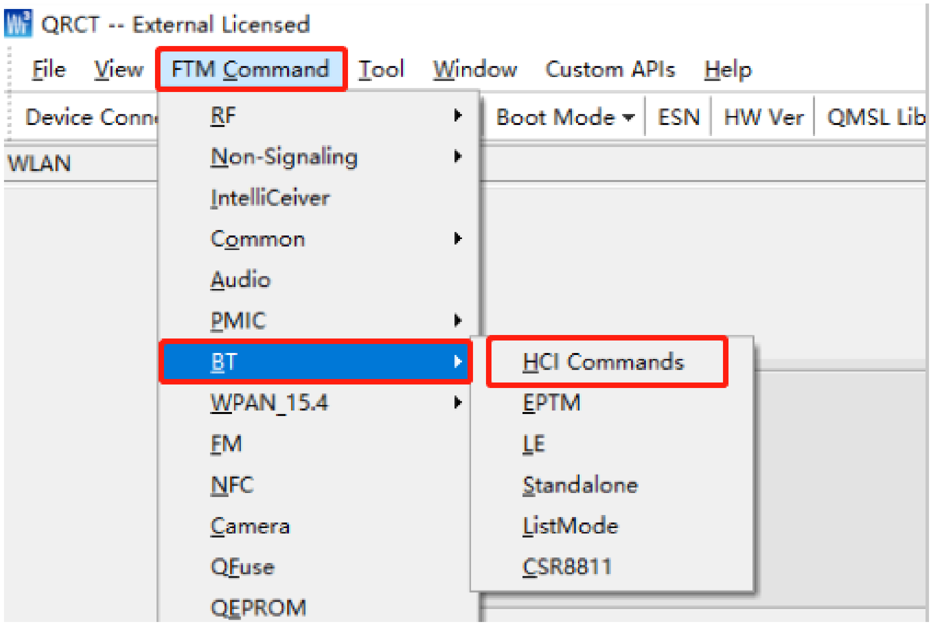


Figure 25: Select HCI Commands

2. Click “**View**” → “**QRCT Debug Message**”.

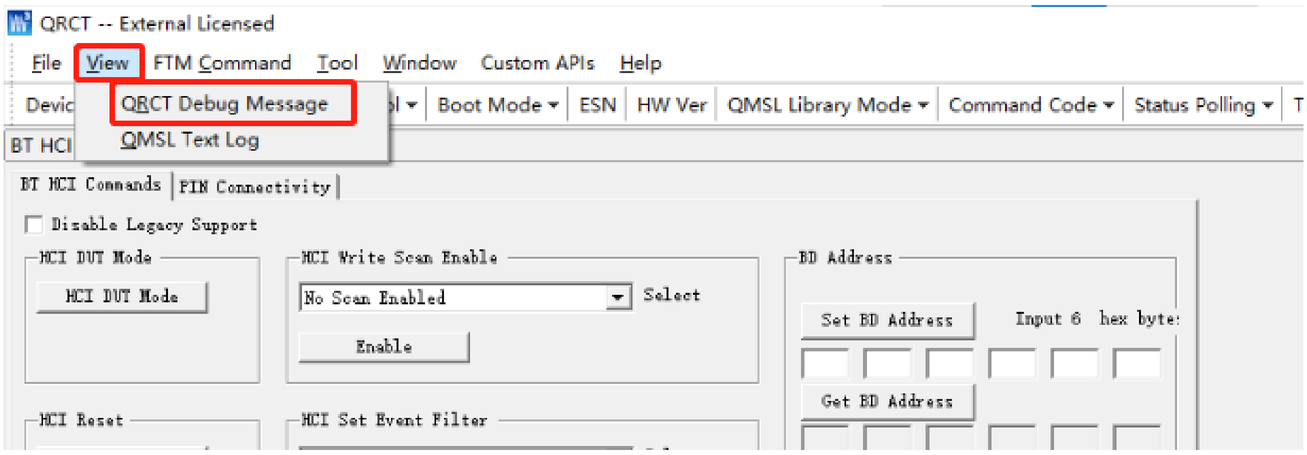
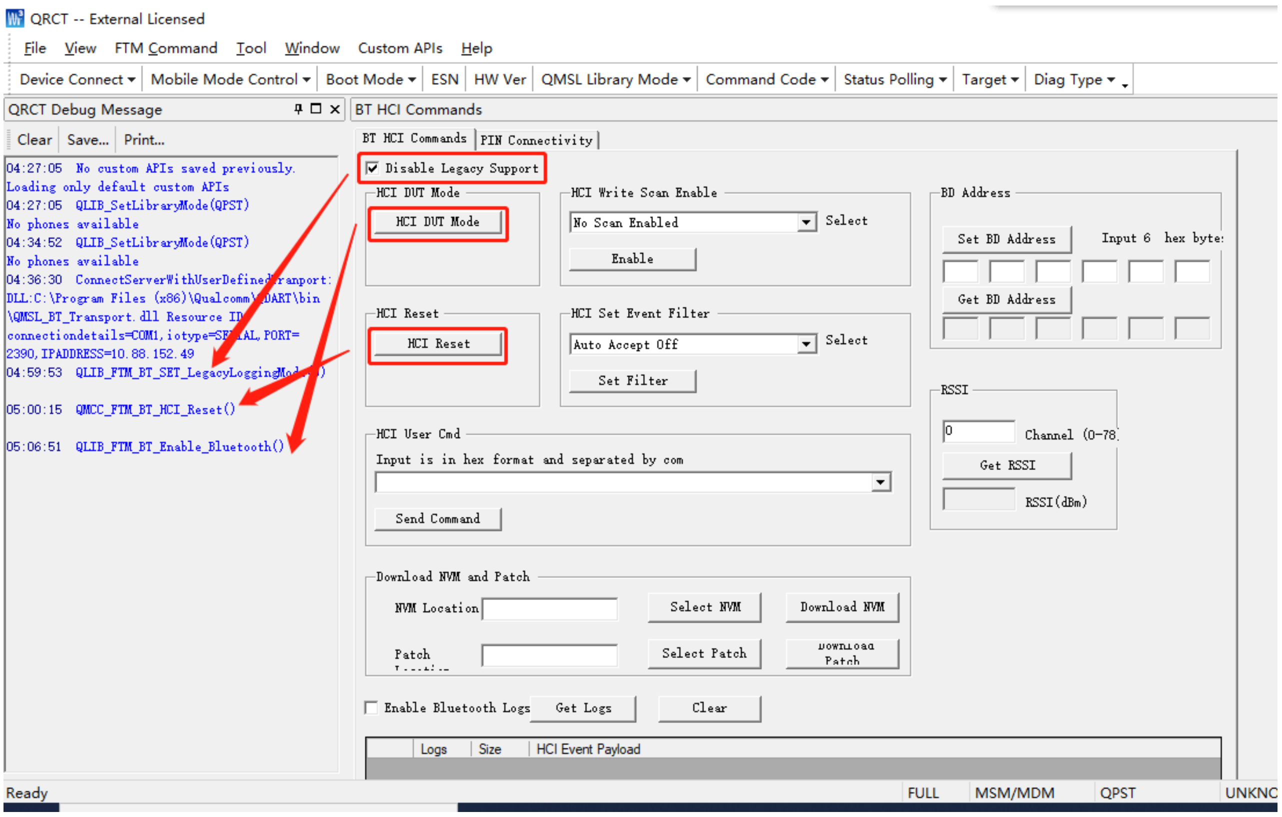


Figure 26: View Debug Message

Check “Disable Legacy Support”, click “**HCI Reset**” to view if there is a failure message in the debug message window, then click “**HCI DUT Mode**”.

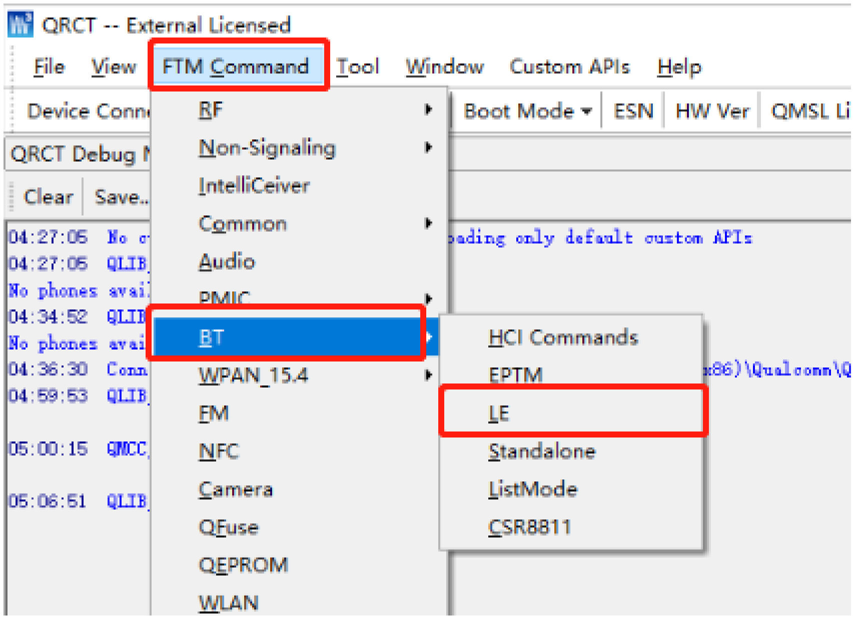

```
04:27:05  No custom APIs saved previously.
Loading only default custom APIs
04:27:05  QLIB_SetLibraryMode(QPST)
No phones available
04:34:52  QLIB_SetLibraryMode(QPST)
No phones available
04:36:30  ConnectServerWithUserDefinedTranport:
DLL:C:\Program Files (x86)\Qualcomm\QDART\bin
\QMSL_BT_Transport.dll Resource ID
connectiondetails=COM1,iotype=SERIAL,PORT=
2390,IPADDRESS=10.88.152.49
04:59:53  QLIB_FTM_BT_SET_LegacyLoggingMode()

05:00:15  QMCC_FTM_BT_HCI_Reset()

05:06:51  QLIB_FTM_BT_Enable_Bluetooth()
```

Figure 27: View Failure Message

3. Click “**FTM Command**” → “**BT**” → “**LE**”.



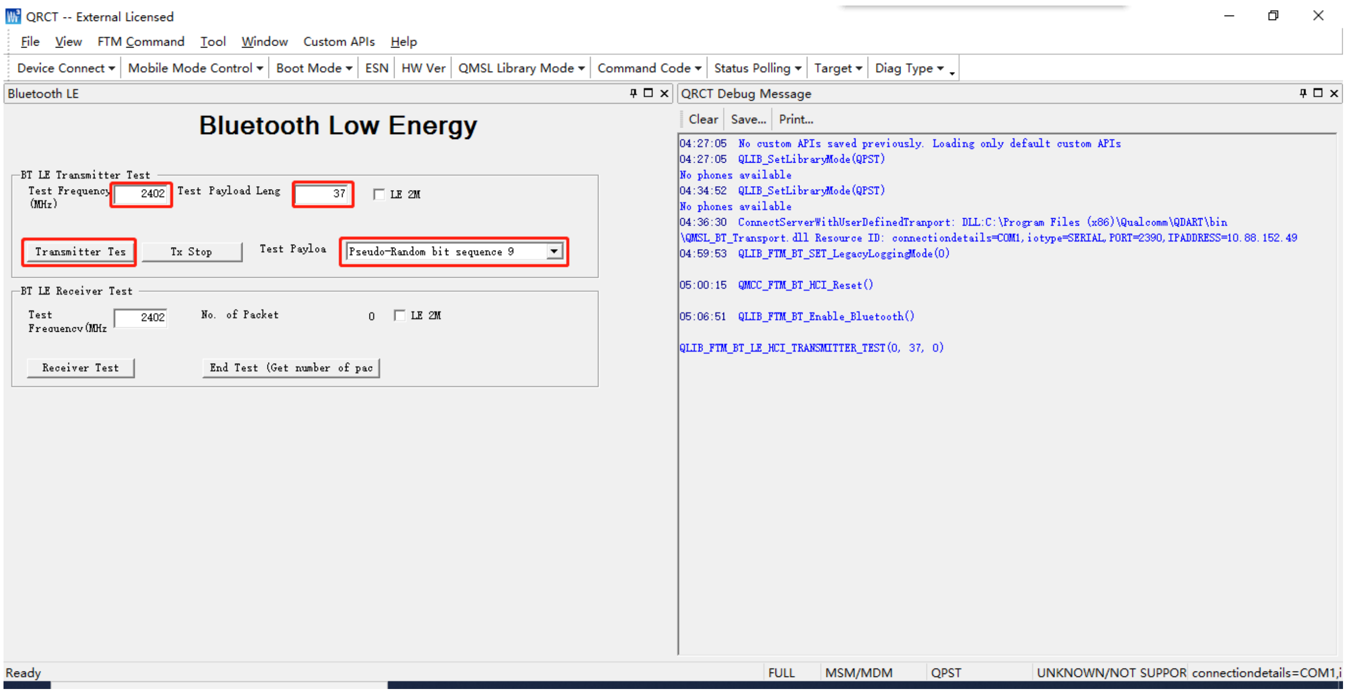
```
04:27:05  No c                           oading only default custom APIs
04:27:05  QLIB
No phones avai
04:34:52  QLIB
No phones avai
04:36:30  Conn                                                            x86)\Qualcomm\Q
04:59:53  QLIB

05:00:15  QMCC

05:06:51  QLIB
```

Figure 28: Select LE Mode

4. In “Transmitter Test” column, set “Test Frequency”, “Test Payload Length” and “Payload Type” according to the actual practice. Then click “**Transmitter Test**”. The test device receives and displays Bluetooth signal strength, as shown in the following figure:



BT LE Transmitter Test

Test Frequency (MHz)	2402	Test Payload Leng	37
☐ LE 2M		Test Payloa	Pseudo-Random bit sequence 9

BT LE Receiver Test

Test Frequency (MHz	2402	No. of Packet	0
☐ LE 2M			

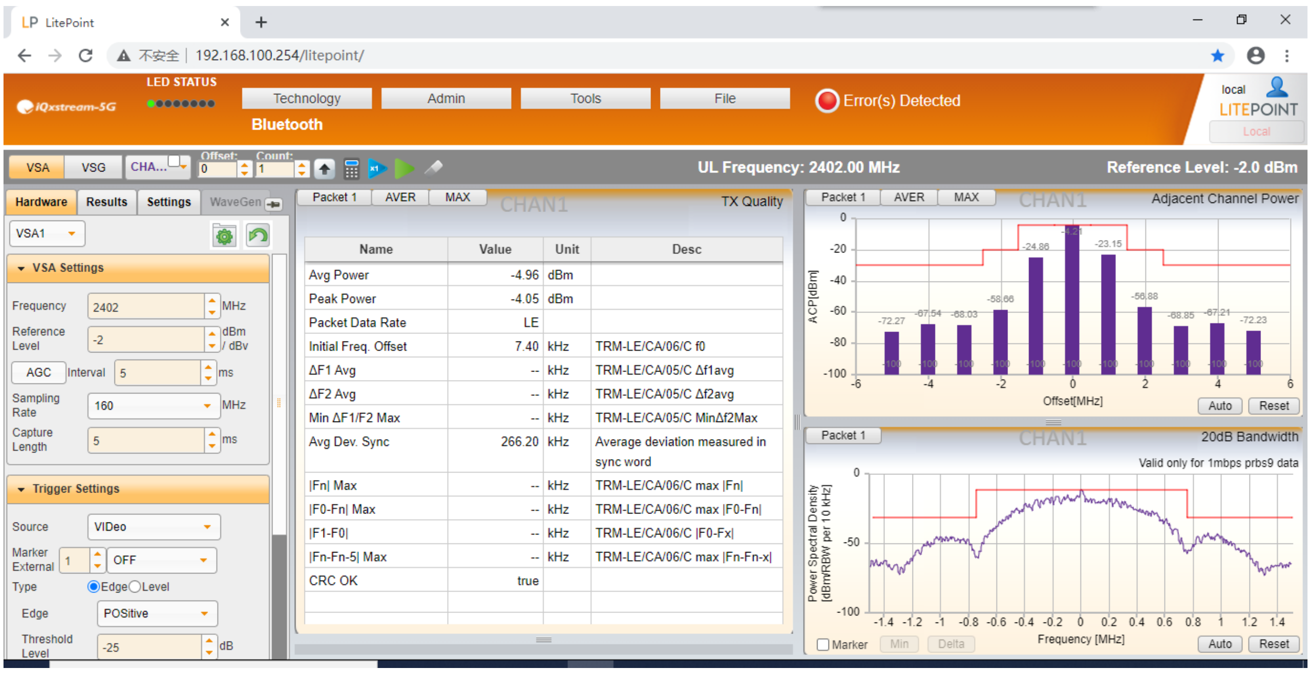
```
04:27:05  No custom APIs saved previously. Loading only default custom APIs
04:27:05  QLIB_SetLibraryMode(QPST)
No phones available
04:34:52  QLIB_SetLibraryMode(QPST)
No phones available
04:36:30  ConnectServerWithUserDefinedTranport: DLL:C:\Program Files (x86)\Qualcomm\QDART\bin
\QMSL_BT_Transport.dll Resource ID: connectiondetails=COM1,iotype=SERIAL,PORT=2390,IPADDRESS=10.88.152.49
04:59:53  QLIB_FTM_BT_SET_LegacyLoggingMode(0)

05:00:15  QMCC_FTM_BT_HCI_Reset()

05:06:51  QLIB_FTM_BT_Enable_Bluetooth()

QLIB_FTM_BT_LE_HCI_TRANSMITTER_TEST(0, 37, 0)
```

Figure 29: Configure BLE



VSA Settings: Frequency 2402 MHz; Reference Level -2 dBm / dBv; AGC Interval 5 ms; Sampling Rate 160 MHz; Capture Length 5 ms

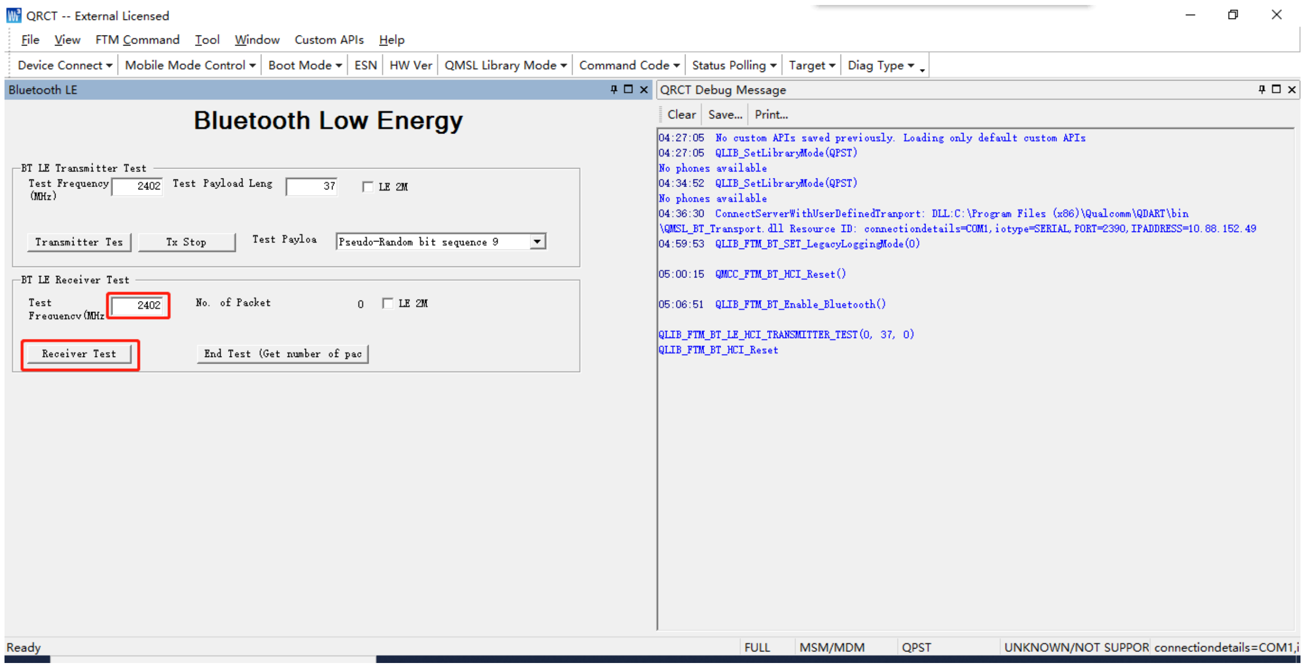
Trigger Settings: Source VIDeo; Marker External 1 OFF; Type [✓ Edge] Level; Edge POSitive; Threshold Level -25 dB

TX Quality

Name	Value	Unit	Desc
Avg Power	-4.96	dBm	
Peak Power	-4.05	dBm	
Packet Data Rate	LE		
Initial Freq. Offset	7.40	kHz	TRM-LE/CA/06/C f0
ΔF1 Avg	--	kHz	TRM-LE/CA/05/C Δf1avg
ΔF2 Avg	--	kHz	TRM-LE/CA/05/C Δf2avg
Min ΔF1/F2 Max	--	kHz	TRM-LE/CA/05/C MinΔf2Max
Avg Dev. Sync	266.20	kHz	Average deviation measured in sync word
|Fn| Max	--	kHz	TRM-LE/CA/06/C max |Fn|
|F0-Fn| Max	--	kHz	TRM-LE/CA/06/C max |F0-Fn|
|F1-F0|	--	kHz	TRM-LE/CA/06/C |F0-Fx|
|Fn-Fn-5| Max	--	kHz	TRM-LE/CA/06/C max |Fn-Fn-x|
CRC OK	true		

Figure 30: Bluetooth Signal Strength

5. Select “Test Frequency” under “Receiver Test” column in the QRCT tool, then click “**Receiver Test**”. The test device will start sending the signal, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling Target Diag Type
Bluetooth LE
Bluetooth Low Energy
BT LE Transmitter Test
Transmitter Tes
Tx Stop
BT LE Receiver Test
Receiver Test
End Test (Get number of pac
QRCT Debug Message
Clear
Save...
Print...
Ready
FULL
MSM/MDM
QPST
UNKNOWN/NOT SUPPOR
connectiondetails=COM1,i

BT LE Transmitter Test: Test Frequency (MHz) 2402; Test Payload Leng 37; LE 2M; Test Payloa Pseudo-Random bit sequence 9

BT LE Receiver Test: Test Frequency (MHz) 2402; No. of Packet 0; LE 2M

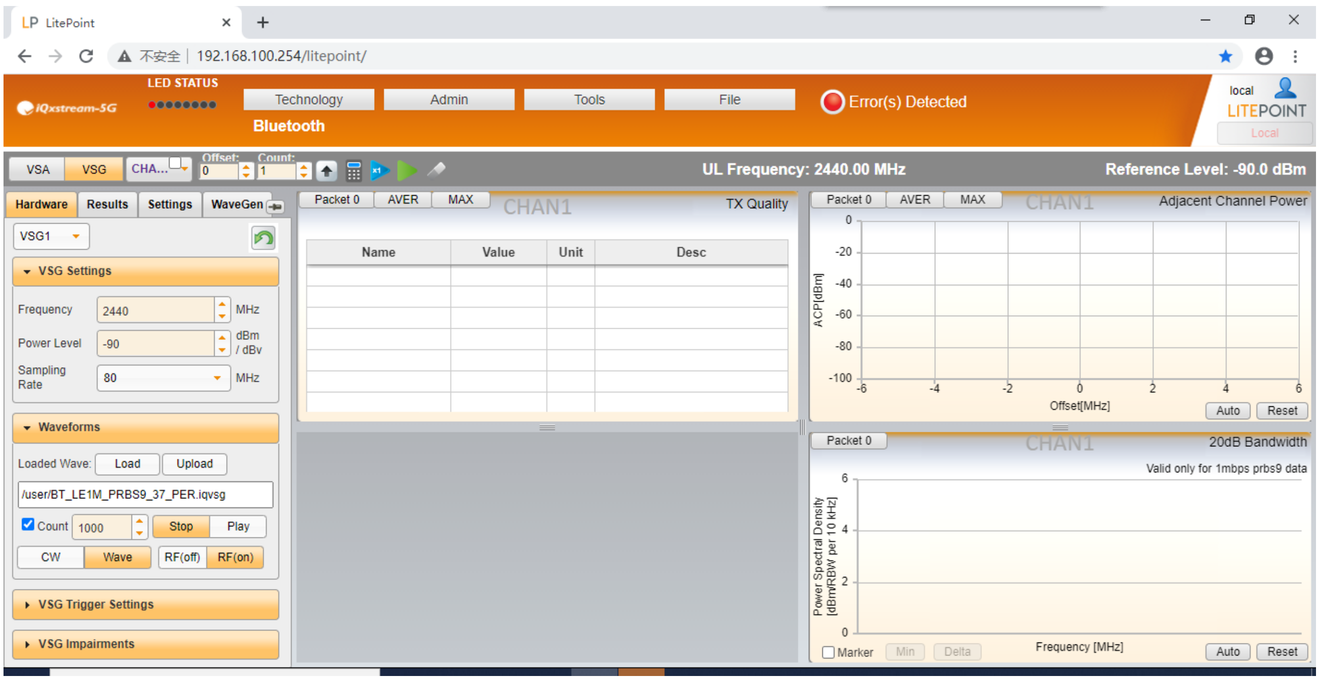
```
04:27:05  No custom APIs saved previously. Loading only default custom APIs
04:27:05  QLIB_SetLibraryMode(QPST)
No phones available
04:34:52  QLIB_SetLibraryMode(QPST)
No phones available
04:36:30  ConnectServerWithUserDefinedTranport: DLL:C:\Program Files (x86)\Qualcomm\QDART\bin
\QMSL_BT_Transport.dll Resource ID: connectiondetails=COM1,iotype=SERIAL,PORT=2390,IPADDRESS=10.88.152.49
04:59:53  QLIB_FTM_BT_SET_LegacyLoggingMode(0)

05:00:15  QMCC_FTM_BT_HCI_Reset()

05:06:51  QLIB_FTM_BT_Enable_Bluetooth()

QLIB_FTM_BT_LE_HCI_TRANSMITTER_TEST(0, 37, 0)
QLIB_FTM_BT_HCI_Reset
```

Figure 31: Configure Receiving Test



UL Frequency: 2440.00 MHz

Reference Level: -90.0 dBm

VSG Settings

Frequency: 2440 MHz

Power Level: -90 dBm / dBv

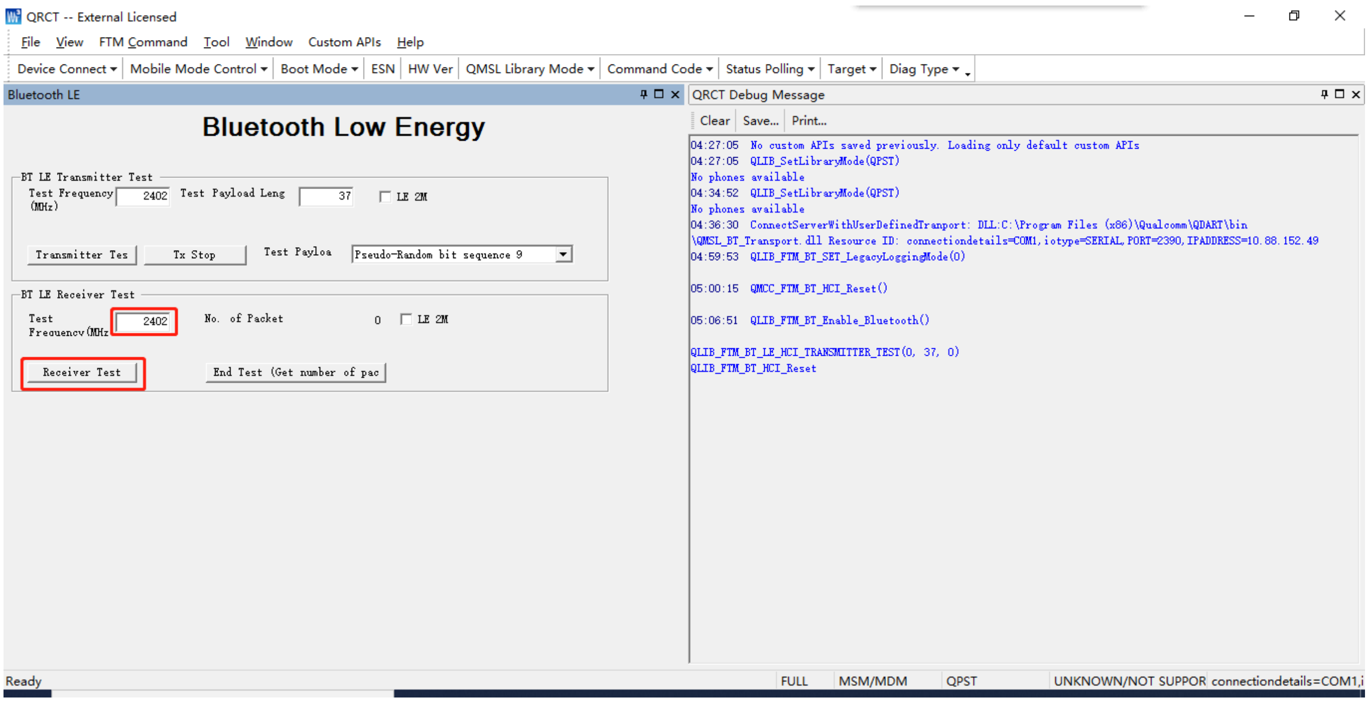
Sampling Rate: 80 MHz

Loaded Wave: /user/BT_LE1M_PRBS9_37_PER.iqvsg

Count: 1000

Figure 32: Send Signal from Test Device

Click “**End Test**” in the QRCT and view the received data.



Bluetooth Low Energy

BT LE Transmitter Test

Test Frequency (MHz): 2402

Test Payload Leng: 37

Test Payloa: Pseudo-Random bit sequence 9

BT LE Receiver Test

Test Frequency (MHz): 2402

No. of Packet: 0

QRCT Debug Message

```
04:27:05  No custom APIs saved previously. Loading only default custom APIs
04:27:05  QLIB_SetLibraryMode(QPST)
No phones available
04:34:52  QLIB_SetLibraryMode(QPST)
No phones available
04:36:30  ConnectServerWithUserDefinedTranport: DLL:C:\Program Files (x86)\Qualcomm\QDART\bin
\QMSL_BT_Transport.dll Resource ID: connectiondetails=COM1,iotype=SERIAL,PORT=2390,IPADDRESS=10.88.152.49
04:59:53  QLIB_FTM_BT_SET_LegacyLoggingMode(0)

05:00:15  QMCC_FTM_BT_HCI_Reset()

05:06:51  QLIB_FTM_BT_Enable_Bluetooth()

QLIB_FTM_BT_LE_HCI_TRANSMITTER_TEST(0, 37, 0)
QLIB_FTM_BT_HCI_Reset
```

Ready | FULL | MSM/MDM | QPST | UNKNOWN/NOT SUPPOR connectiondetails=COM1,i

Figure 33: View Received Data

3.3.4. Test Bluetooth (BR + EDR)

1. Click “**FTM Command**“ → “**BT**“ → “**EPTM**“.

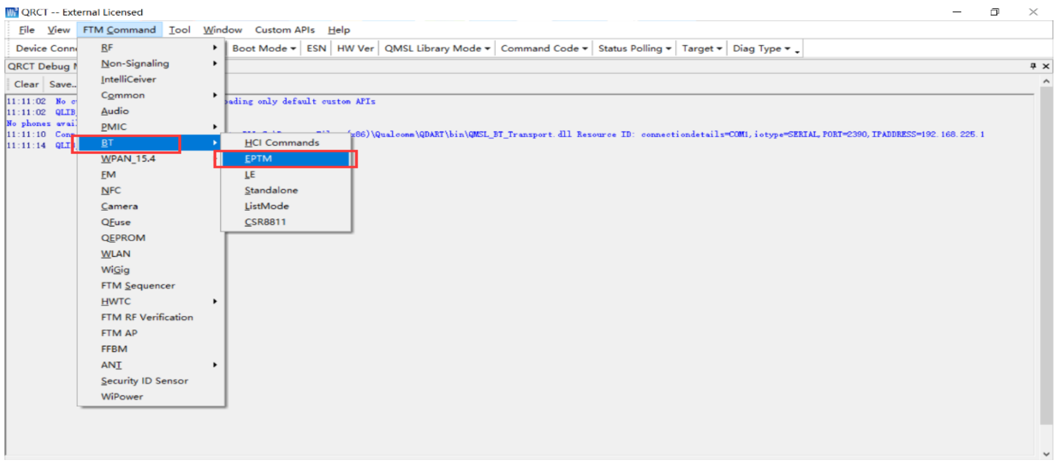
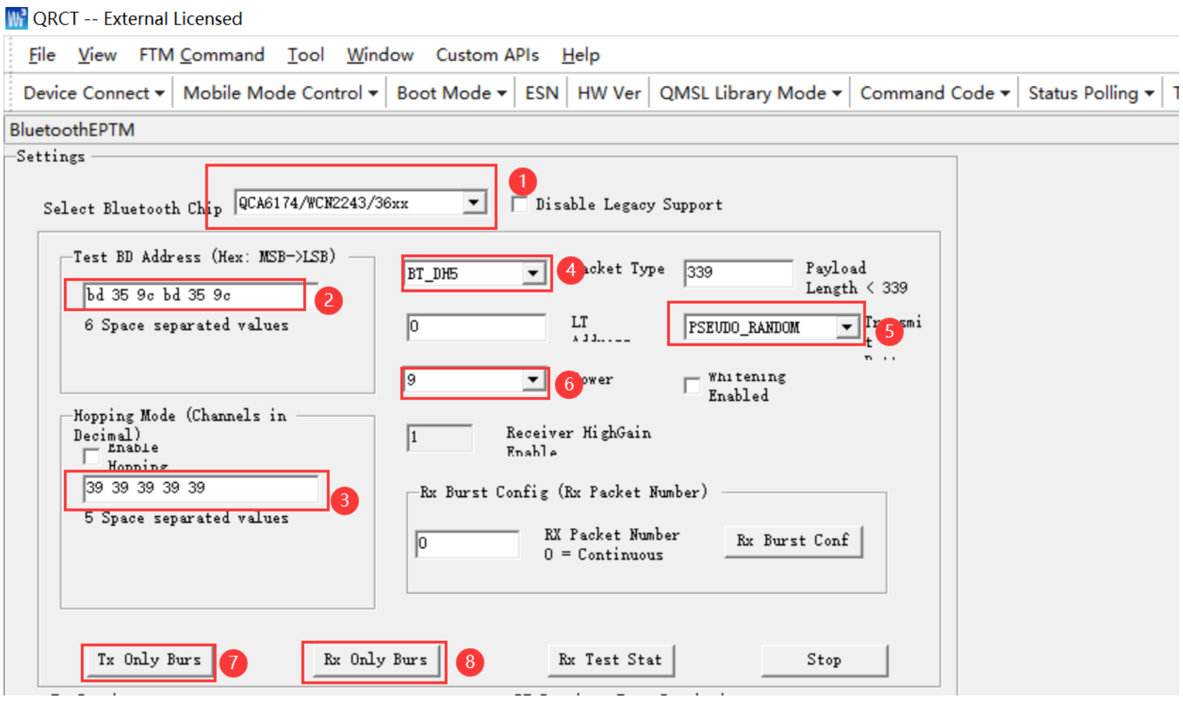


Figure 34: Select EPTM

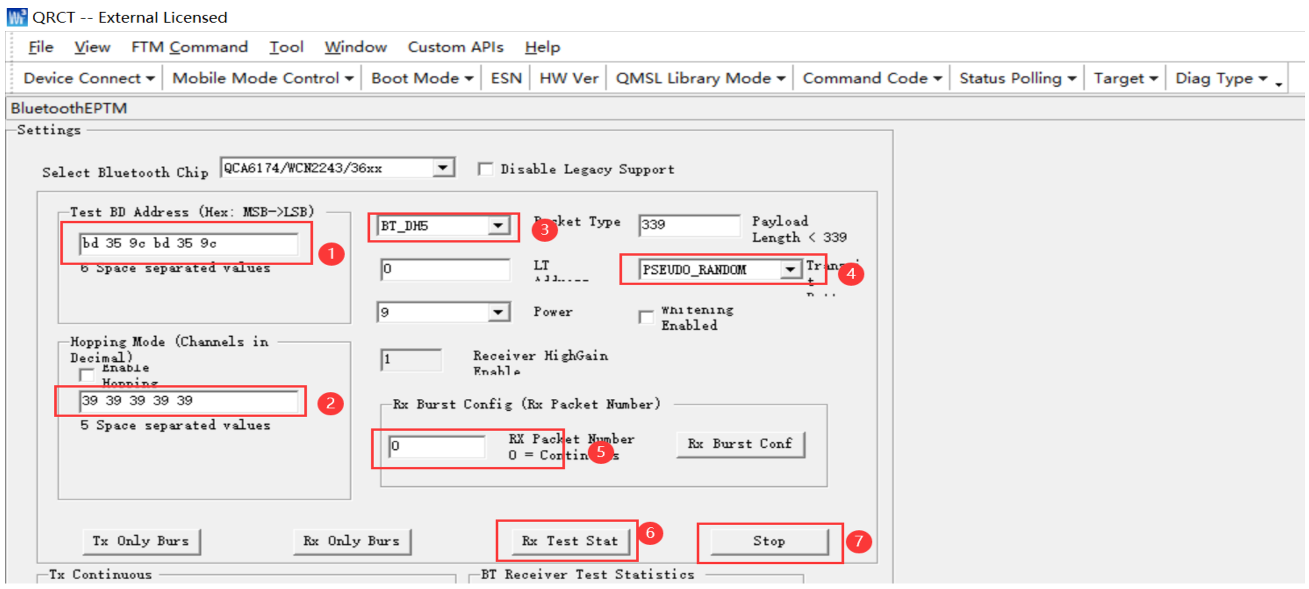
2. In “Settings” of “BluetoothEPTM” window, keep the default values of the Bluetooth chip (①) and BD address (②). Then configure the corresponding channel (③), packet type (④), transmit pattern (⑤) and transmit power (⑥) as needed, as shown in the following figure:


QRCT -- External Licensed
File View FTM Command Tool Window Custom APIs Help
Device Connect Mobile Mode Control Boot Mode ESN HW Ver QMSL Library Mode Command Code Status Polling
BluetoothEPTM
1
2
3
4
5
6
7
8
Rx Burst Conf
Tx Only Burs
Rx Only Burs
Rx Test Stat
Stop

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
Hopping Mode (Channels in Decimal)
Enable Hopping
39 39 39 39 39
5 Space separated values
Packet Type: BT_DH5
Payload Length < 339: 339
LT Address: 0
Transmit Pattern: PSEUDO_RANDOM
Power: 9
Whitening Enabled
Receiver HighGain Enable: 1
Rx Burst Config (Rx Packet Number)
RX Packet Number 0 = Continuous: 0

Figure 35: Bluetooth EPTM RX Test Setting

3. Select “Test Frequency” under “Receiver Test” column, then click “**Rx Test Stat**”. After that, the test device begins to send signal. The “RX Packet Number” (⑤) in the figure below indicates the number of received packets, as shown in the following figure:



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

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

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

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

Figure 36: Bluetooth EPTM RX Test Start

NOTE

If the QRCT is closed, the console prints “Client disconnected”. If it is necessary to test Bluetooth or Wi-Fi again, reboot DUT and then execute **Btdiag UDT=yes PORT=2390 IOType=SERIAL QDARTIOType=ethernet BT-DEVICE=/dev/ttymxc1 BT-BAUDRATE=115200 IPADDRESS=10.88.152.49**. If the DUT is not rebooted, it returns an error after executing the above commands.

4 Appendix Reference

Table 1: Terms and Abbreviations

Abbreviation	Description
AP	Access Point
BLE	Bluetooth Low Energy
BSS	Base Station System
BT	Bluetooth
CCK	Complementary Code Keying
CPU	Central Processing Unit
DAC	Digital to Analog Convertor
DCM	Dual Carrier Modulation
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic Link Library
EDR	Enhanced Data Rate
EPTM	Embedded Product Test Mode
EVB	Evaluation Board
DUT	Device Under Test
FTM	Factory Test Mode
GATT	Generic Attribute Profile
GI	Guard Interval
GPIO	General-Purpose Input/Output
HCI	Host Controller Interface

IP	Internet Protocol
IoT	Internet of Things
LAN	Local Area Network
LDPC	Low-density Parity-check
LE	Low Energy
LTF	Long Training Field
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
NSS	Number of Spatial Streams
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PA	Power Amplifier
PHY	Physical
PPDU	Presentation Protocol Data Unit
PSK	Pre-Shared Key
RF	Radio Frequency
RX	Receive
SDIO	Secure Digital Input/Output
SDK	Software Development Kit
SSID	Service Set Identifier
STA	Station
STBC	Space Time Block Code
TCP	Transmission Control Protocol
TPC	Transmit Power Control
TX	Transmit

UART	Universal Asynchronous Receiver/Transmitter
UDT	UDP-based Data Transfer Protocol
USB	Universal Serial Bus
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access