AN14365

RF Test Mode on Android OS

Rev. 1.0 — 16 December 2024

Application note

Document information

Information	Content
Keywords	RF test mode, production firmware, regulatory, compliance
Abstract	Describes how to enable and use RF test mode on Android OS.

1 Introduction

This document provides an overview of how to enable and use the RF test mode on an Android host. RF test mode feature is used to set RF parameters for transmit and receive testing for regulatory compliance. The feature is available for use on the production software.

RF test mode is compatible with Wi-Fi and Bluetooth/Bluetooth LE radios.

Note: *This document assumes that you are familiar with [2] and that you have used the production firmware to bring up the radios on your device.*

1.1 Supported devices

- 88W8987
- 88W8997
- 88W9098
- IW416
- IW611
- IW612

Note: *For more information on the software compatibility, refer to the software release notes of your device.*

2 Wi-Fi RF test mode

This section describes the commands to use RF test mode for Wi-Fi on the i.MX 8M Quad EVK with Android. The commands apply to any Android-based host.

Wi-Fi RF test mode commands can be set by using the echo command to pass parameters to a configuration file located in the */proc/mwlan/adapterX/config* directory.

Note: *In the command examples,* `adapter0` *refers to the Wi-Fi device name. To check your device name, refer to the /proc/mwlan/ directory for* `adapterX` *after loading the drivers and firmware. Adjust this parameter for your device.*

2.1 List of commands for Wi-Fi RF test mode

Table 1 lists the commands available for RF test mode of Wi-Fi.

Table 1. List of commands for Wi-Fi RF test mode

Command
Enable RF test mode and check RF parameters
Set TX/RX antenna configuration
Set the radio mode
Set the operating RF band
Set the channel bandwidth
Set the RF channel
Get and reset the packet error rate
Set TX power
Set TX continuous mode
Set TX frame
Testing 802.11ax Uplink-OFDMA transmit

2.2 Enable RF test mode and check RF parameters

RF test mode must be enabled before setting other RF test mode parameters.

Command to **enable** RF test mode:

```
echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config
```

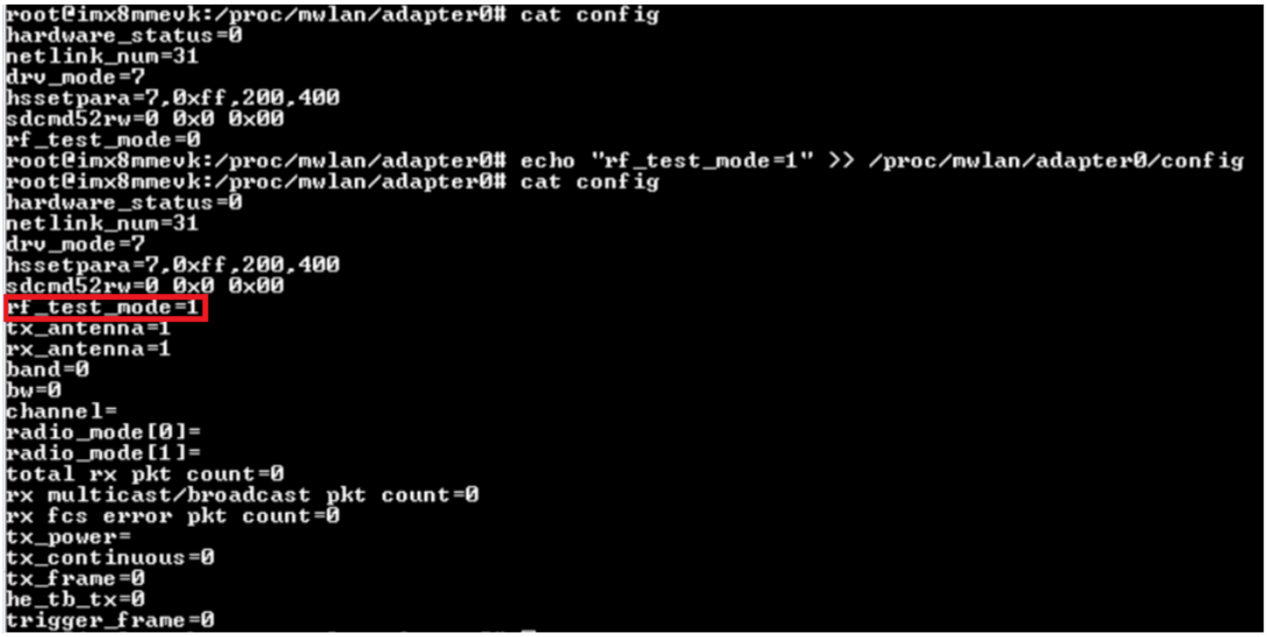
Command to **disable** RF test mode:

```
echo "rf_test_mode=0" >> /proc/mwlan/adapter0/config
```

Command to check your set RF test mode parameters by printing the contents of the configuration file:

```
cat /proc/mwlan/adapter0/config
```

Figure 1 shows the example of RF test mode enabled, with the output value of *rf_test_mode=1* in the configuration file.



```
root@imx8mmevk:/proc/mwlan/adapter0# cat config
hardware_status=0
netlink_num=31
drv_mode=7
hssetpara=7,0xff,200,400
sdcmd52rw=0 0x0 0x00
rf_test_mode=0
root@imx8mmevk:/proc/mwlan/adapter0# echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config
root@imx8mmevk:/proc/mwlan/adapter0# cat config
hardware_status=0
netlink_num=31
drv_mode=7
hssetpara=7,0xff,200,400
sdcmd52rw=0 0x0 0x00
rf_test_mode=1
tx_antenna=1
rx_antenna=1
band=0
bw=0
channel=
radio_mode[0]=
radio_mode[1]=
total rx pkt count=0
rx multicast/broadcast pkt count=0
rx fcs error pkt count=0
tx_power=
tx_continuous=0
tx_frame=0
he_tb_tx=0
trigger_frame=0
```

Figure 1. Example of command output with `rf_test_mode` enabled

2.3 Set TX/RX antenna configuration

Command to set the antenna configuration for **transmission mode**:

```
echo "tx_antenna=<mode>" >> /proc/mwlan/adapter0/config
```

Command to set the antenna configuration for **receive mode**:

```
echo "rx_antenna=<mode>" >> /proc/mwlan/adapter0/config
```

Table 2. Command parameters

Parameter	Description
`TX mode`	TX antenna 1 = Path A 2 = Path B 3 = Paths A and B ***Note:*** *For 1x1 devices, select Path A.*
`RX mode`	RX antenna 1 = Path A 2 = Path B 3 = Paths A and B ***Note:*** *For 1x1 devices, select Path A.*

Example of command to set the antenna configuration for transmission mode (Path A):

```
echo "tx_antenna=1" >> /proc/mwlan/adapter0/config
```

Example of command to set the antenna configuration for receive mode (Path B):

```
echo "rx_antenna=2" >> /proc/mwlan/adapter0/config
```

2.4 Set the radio mode

Command to set the radio mode:

```
echo "radio_mode=<index for radio 0> <index for radio 1>" >> /proc/ mwlan/adapter0/config
```

Note: *This command is only for 88W9098, IW611, and IW612.*

Table 3 lists the radio mode indexes for 88W9098.

Table 3. Radio mode index values for 88W9098

Radio mode index	Band and antenna path configuration
0	Sets the radio in power down mode. ***Note:*** *Radio mode index for devices with dual Wi-Fi radios only.*
1	Sets the radio in 5 GHz band, 2x2 mode (Path A+B) ***Note:*** *Radio mode index for devices with dual Wi-Fi radios only.*
3	Sets the radio in 5 GHz band, 1x1 mode (Path A)
4	Sets the radio in 5 GHz band, 1x1 mode (Path B)
9	Sets the radio in 2.4 GHz band, 2x2 mode (Path A+B) ***Note:*** *Radio mode index for devices with dual Wi-Fi radios only.*
11	Sets the radio in 2.4 GHz band, 1x1 mode (Path A)
14	Sets the Radio in 2.4 GHz band, 1x1 mode (Path B)

Example of command to set the radio 0 in 5 GHz band with 2x2 mode, and radio 1 in power down mode:

```
echo "radio_mode=1 0" >> /proc/mwlan/adapter0/config
```

Example of command to set the radio 0 in power down mode, and radio 1 in 2.4 GHz band with 2x2 mode:

```
echo "radio_mode=0 9" >> /proc/mwlan/adapter1/config
```

Table 4 lists the radio mode indexes for IW612 and IW611.

Table 4. Radio mode index values for IW612 and IW611

Radio mode index	Description
3	Radio in 5 GHz band
11	Radio in 2.4 GHz band

Note: *In the following examples, the “radio index 1" is set to 0 as IW612 device has only one Wi-Fi radio.*

Example of command to set the Wi-Fi radio in 5 GHz band for an IW612:

```
echo "radio_mode=3 0" >> /proc/mwlan/adapter0/config
```

Example of command to set the Wi-Fi radio in 2.4 GHz band for an IW612:

```
echo "radio_mode=11 0" >> /proc/mwlan/adapter0/config
```

2.5 Set the operating RF band

Command to set the RF band:

```
echo "band=<RF band>" >> /proc/mwlan/adapter0/config
```

Table 5. Command parameters

Parameter	Definition
`RF band`	RF band 0 = 2.4 GHz 1 = 5 GHz

Example of command to set the RF band to 5 GHz:

```
echo "band=1" >> /proc/mwlan/adapter0/config
```

2.6 Set the channel bandwidth

Command to set the channel bandwidth:

```
echo "bw=<bandwidth>" >> /proc/mwlan/adapter0/config
```

Table 6. Command parameters

Parameter	Definition
`bandwidth`	Channel bandwidth 0 = 20 MHz 1 = 40 MHz 4 = 80 MHz

Example of command to set the channel bandwidth to 20 MHz:

```
echo "bw=0" >> /proc/mwlan/adapter0/config
```

2.7 Set the RF channel

Command to set the RF channel:

```
echo "channel=<ch>" >> /proc/mwlan/adapter0/config
```

Table 7. Command parameters

Parameter	Definition
`ch`	RF channel of operation ***Note:*** *For more information on the supported RF channels, refer to Section 2.14.*

Example of command to set the RF channel to 6:

```
echo "channel=6" >> /proc/mwlan/adapter0/config
```

2.8 Get and reset the packet error rate

Command to get and reset the packet error rate in the configuration file:

```
echo "get_and_reset_per" >> /proc/mwlan/adapter0/config
```

Command to verify the packet count parameters:

```
cat /proc/mwlan/adapter0/config
```

Command output example:

```
hardware_status=0 netlink_num=31 drv_mode=7 sdcmd52rw=0 0x0 0x00 rf_test_mode=1
 TX_antenna=1 RX_antenna=1
band=1 bw=0 channel=36
radio_mode[0]= radio_mode[1]=
total RX pkt count=500
RX multicast/broadcast pkt count=500
RX fcs error pkt count=0
TX_power=
TX_continuous=0
TX_frame=0
he_tb_TX=0
```

The bold text in the command output example shows the packet count data after 500 packets were received by the DUT. The command output includes the total packet count, the multicast/broadcast packet count, and the frame check sequence (FCS) error packet count.

2.9 Set TX power

Command to set the TX power:

```
echo "tx_power=<power> <modulatio> <path id>" >> /proc/mwlan/adapter0/config
```

Table 8. Command parameters

Parameter	Definition
`power`	Transmit power level in dBm Integer range of -1 to 24 -1 = power level is determined by the firmware and accounts for country code restrictions Else = user input power level
`modulation`	Signal modulation 0 = CCK 1 = OFDM 2 = MCS
`path id`	TX signal path name 0 = path A 1 = path B 2 = path A + path B ***Note:*** *For 1x1 devices, select Path A.*

Example of command to set the TX power to 16 dBm using MCS modulation on path A:

```
echo "tx_power=16 2 0" >> /proc/mwlan/adapter0/config
```

Example of command to determine the power level by firmware with OFDM modulation on path B:

```
echo "tx_power=-1 1 1" >> /proc/mwlan/adapter0/config
```

Note: *If the power level is set to -1 and the current channel is not allowed in the set country code, the command returns an error.*

2.10 Set TX continuous mode

Command to set the TX continuous mode parameters:

```
echo "tx_continuous=<start/stop> <continuous wave mode> <payload pattern> <cs mode>
 <active subchannel> <TX data rate>" >> /proc/mwlan/adapter0/config
```

Table 9. Command parameters

Parameter	Definition
`start/stop`	Start/stop transmit 0 = disable 1 = enable
`transmit mode`	Transmit mode 0 = continuous packet mode 1 = continuous wave mode
`payload pattern`	Payload pattern value in the range of 0 to 0xFFFFFFFF.
`cs mode`	TX Carrier Suppression (CS) enable *Applicable only when continuous packet mode is set 0 = disable 1 = enable
`active subchannel`	Active sub-channel 0 = low 1 = upper 3 = both
`TX data rate`	Transmit data rate index corresponding to legacy/HT/VHT rates (Section 2.13).

Example of command to start TX continuous mode in continuous packet mode with 0xAAA payload pattern, TX carrier suppression disabled, both subchannels active, and a data rate of 12 Mbps:

```
echo "tx_continuous=1 0 0xAAA 0 3 0x7" >> /proc/mwlan/adapter0/config
```

Example of command to stop TX continuous mode in continuous packet mode:

```
echo "tx_continuous=0" >> /proc/mwlan/adapter0/config
```

Example to start continuous wave mode:

```
echo "tx_continuous=1 1 0xAAA 0 3 0x7" >> /proc/mwlan/adapter0/config
```

Example of command to stop continuous wave mode:

```
echo "tx_continuous=0 1 0xAAA 0 3 0x7" >> /proc/mwlan/adapter0/config
```

2.11 Set TX frame

Command to set the TX frame parameters

```
echo "tx_frame=<start/stop> <TX data rate> <payload pattern> <payload length> <adjust
 burst SIFS gap> <adjust SIFS> <short preamble> <active subchannel> <short GI> <adv
 coding> <beamforming> <greenfield mode> <STBC> <Signal Bw> <NumPkt> <MaxPktExt>
 <BeamChange> <DCM> <Doppler> <MidamblePeriod> <QNum> <BSSID>" >> /proc/mwlan/adapter0/
config
```

Table 10. Command parameters

Parameter	Definition
`start/stop`	Start/stop transmit 0 = disable 1 = enable
`TX data rate`	Transmit the data rate index corresponding to the legacy/HT/VHT rates (Section 2.13).
`payload pattern`	Payload pattern value in the range of 0 to 0xFFFFFFFF.
`payload length`	Payload length value in the range of 1 to 0x400.
`adjust burst SIFS gap`	Adjust burst SIFS gap enable 0= disable (default) 1 = enable
`adjust SIFS`	Burst SIFS duration in microseconds range of 10 µs (default) to 255 µs
`short preamble`	Short preamble enable 0= disable (default) 1 = enable
`active subchannel`	Active subchannel selection 0 = lower (default) 1 = upper 3 = both
`short GI`	Short guard interval 0 = disable (default) 1 = enable

Table 10. Command parameters...*continued*

Parameter	Definition
`adv coding`	Advanced coding 0 = disable (default) 1 = enable ***Note:*** *Set 1 (enable) for 802.11ax MCS10/MCS11, 40 MHz/80 MHz configuration test.*
`beamforming`	Beamforming enable 0 = disable (default) 1 = enable
`greenfield mode`	Greenfield mode enable 0 = disable (default) 1 = enable
`STBC`	Space time block coding enable 0 = disable (default) 1 = enable
`Signal Bw`[1]	Signal bandwidth 0 = 20 MHz 1 = 40 MHz 4 = 80 MHz -1 = Set to default
`NumPkt`	Number of packets. Set to default value -1.
`MaxPktExt`	Max packet extension. Set to default value -1.
`BeamChange`	Beam change Set to default value -1.
`DCM`	DCM enable Set to default value -1.
`Doppler`	Doppler enable Set to default value -1.
`MidamblePeriod`	Midamble periodicity Set to default value -1.
`QNum`	Transmit queue number that holds the trigger-based response packets. Set to default value -1.
`BSSID`	Basic service set identifiers Format: xx:xx:xx:xx:xx:xx

[1] The command in Section 2.6 is used to set the channel bandwidth whereas the bandwidth set using tx_frame parameter is for the signal. If the signal bandwidth is set to -1 or greater than the channel bandwidth, the signal bandwidth will be set to be the same as the channel bandwidth.

Example of command to start TX frame with at 12 Mbps, 0xAAA payload pattern, and a packet length of 0x256:

```
echo "tx_frame=1 0x7 0xAAA 0x256 0 20 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1
 05:43:3f:c4:51" >> /proc/mwlan/adapter0/config
```

Example of command to stop any ongoing TX frame:

```
echo "tx_frame=0" >> /proc/mwlan/adapter0/config
```

2.12 Testing 802.11ax Uplink-OFDMA transmit

This section shows how to run Uplink (UL)-OFDMA test by using Wi-Fi RF test mode commands. In the standard test setup, two boards are required to accomplish the test. One board, known as the golden unit, is used to transmit the trigger frame to the DUT. The other board is the DUT, which responds to the trigger frame sent by the golden unit.

A standalone test setup with only one board (DUT) used to test UL-OFDMA transmit may also be used. See Section 2.12.3.

Note: *This section only applies to devices that support 5 GHz Wi-Fi 6.*

2.12.1 Test setup

Figure 2 shows the standard setup for the UL-OFDMA test in the test lab. One i.MX8M Quad (Golden) is used to send a trigger frame and a second i.MX8M Quad (DUT) is used to respond to the trigger frame with a UL-OFDMA signal. A horn antenna receives the UL-OFDMA signal from the DUT and the signal is analyzed with a test receiver.

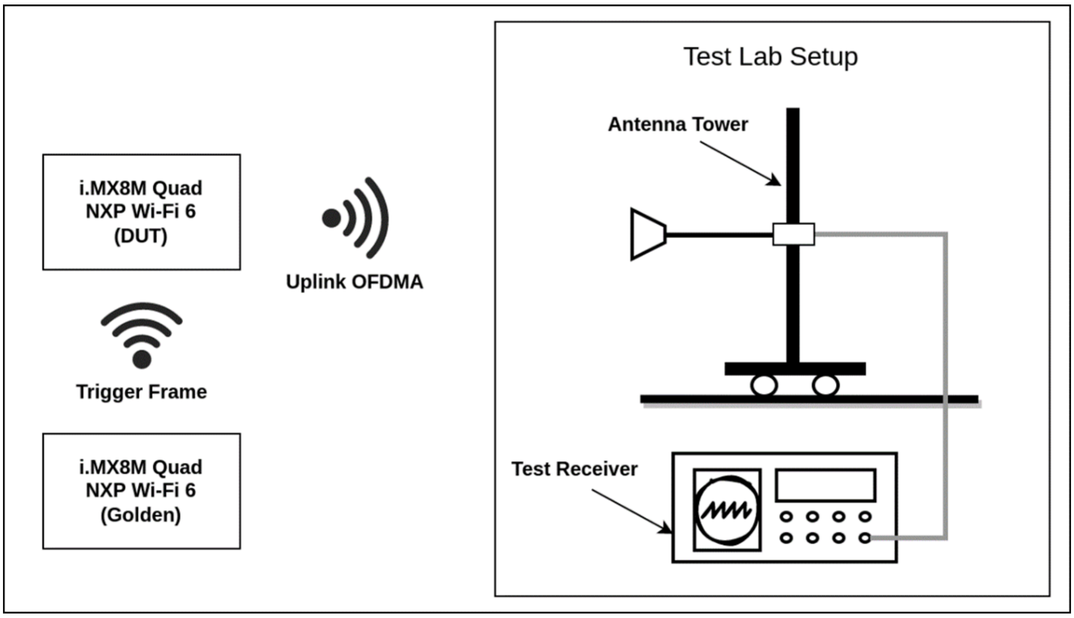


Figure 2. Test setup

2.12.2 Configure the golden unit and DUT for UL-OFDMA transmission

Command to set the trigger frame parameters on the golden unit:

```
echo "trigger_frame=<enable_TX> <standalone_hetb> <frame ctrl type> <frame ctrl subtype>
 <frame duration> <trigger type> <UlLen> <MoreTF> <CSRequired> <UlBw> <LTFType> <LTFMode>
 <LTFSymbol> <UlSTBC> <LdpcESS> <ApTXPwr> <PreFecPadFct> <PeDisambig> <SpatialReuse>
<Doppler> <HeSig2> <AID12> <RUAllocReg> <RUAlloc> <UlCodingType> <UlMCS> <UlDCM><SSAlloc>
 <UlTargetRSSI> <MPDU_MU_SF> <TID_AL> <AC_PL> <Pref_AC>” >> /proc/ mwlan/adapter0/config
```

Table 11. Command parameters

Parameter	Definition
`enable_TX`	Enable transmit 0 = disable 1 = enable
`standalone_hetb`	Enable standalone UL-OFDMA (Section 2.12.3) 0 = disable 1 = Trigger based UL-OFDMA 2 = Standalone based UL-OFDMA 3 = SU-OFDMA
`frame ctrl type`	Set to 1
`frame ctrl subtype`	Set to 2
`frame duration`	Set to 5484
`trigger type`	Set to 0
`UlLen`	Set to 1000
`MoreTF`	Set to 0
`CSRequired`	Set to 0
`UlBw`	Bandwidth 0 = 20 MHz 1 = 40 MHz 2 = 80 MHz
`LTFType`	Set to 1
`LTFMode`	Set to 0
`LTFSymbol`	Select the Long Training Field Symbol (LTFS) 0 = 1xHELTF for 1SS 1 = 2xHELTF for 2SS
`UlSTBC`	Set to 0
`LdpcESS`	Set to 1
`ApTXPwr`	Set to 0
`PreFecPadFct`	Set to 1
`PeDisambig`	Set to 0
`SpatialReuse`	Set to 65535
`Doppler`	Set to 0
`HeSig2`	Set to 511

Table 11. Command parameters*...continued*

Parameter	Definition
`AID12`	Set to 5
`RUAllocReg`	Set to 0
`RUAlloc`	RU index. The RU index value for 20 MHz, 40 MHz, and 80 MHz channel bandwidths are shown in Figure 3, Figure 4, and Figure 5 respectively.
`UlCodingType`	Set to 1
`UlMCS`	MCS data rate. Refer to Section 2.13.
`UlDCM`	Set to 0
`SSAlloc`	Select the spatial stream 0 = 1SS 1 = 2SS
`UlTargetRSSI`	Set to 90
`MPDU_MU_SF`	Set to 0
`TID_AL`	Set to 0
`AC_PL`	Set to 0
`Pref_AC`	Set to 0

Note: *The DUT transmits UL-OFDMA for each trigger frame it receives. Modify the transmit duty cycle by adjusting the TX time gap of the trigger frames on the golden unit. The RU index and MCS data rate of the UL-OFDMA transmission are based on the received trigger frame.*

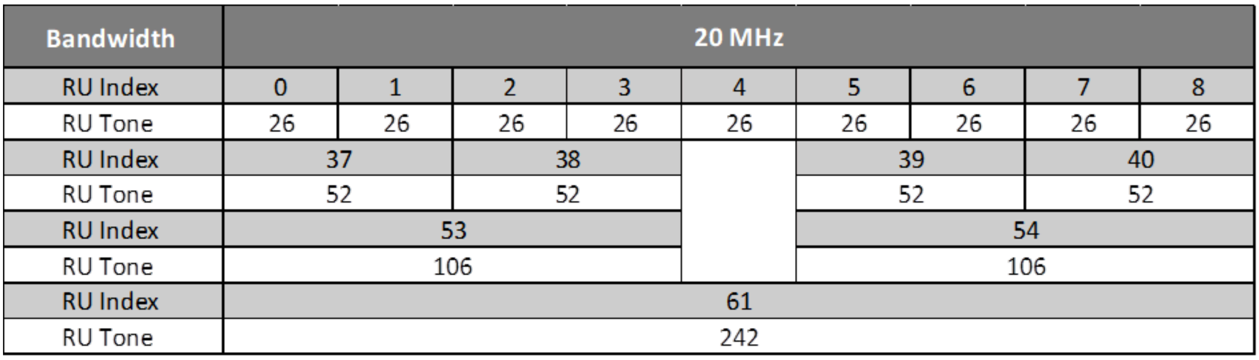
Table 12 lists the steps and commands for HE-trigger frame generation on the golden unit under the following conditions:

- 5 GHz path A+B
- Channel 36 and 20 MHz channel bandwidth

Table 12. Steps for HE-trigger frame generation on the golden unit

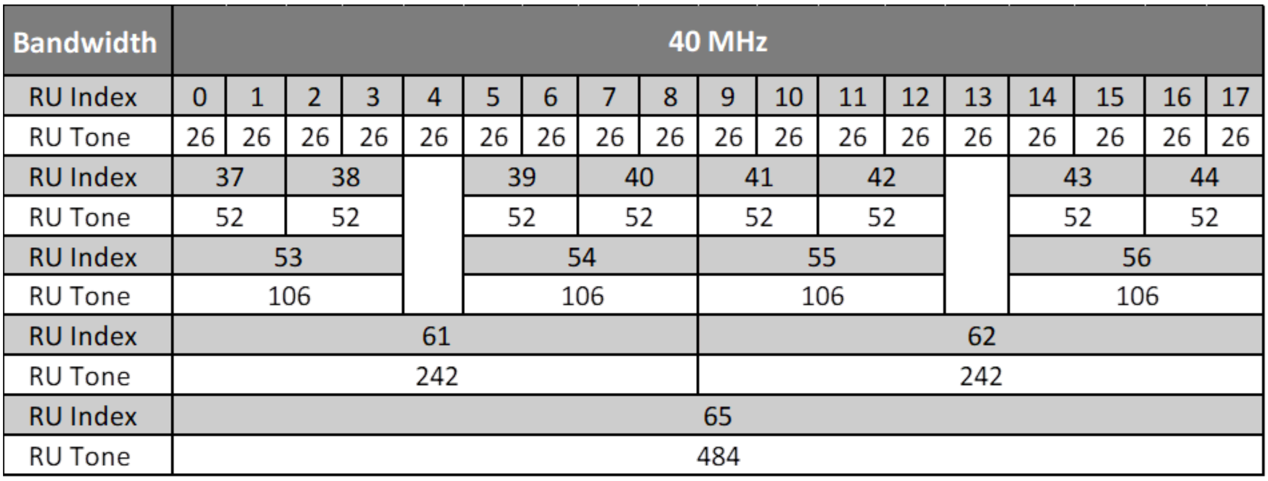
Step	Operation	Command
1	Enable RF test mode	`# echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set radio mode to 5 GHz (2x2 mode)	`# echo "radio_mode=1 0" >> /proc/mwlan/adapter0/config`
3	Set the band to 5 GHz	`# echo "band=1" >> /proc/mwlan/adapter0/config`
4	Set the bandwidth to 20 MHz	`# echo "bw=0" >> /proc/mwlan/adapter0/config`
5	Set the channel to 36	`# echo "channel=36" >> /proc/mwlan/adapter0/config`
6	Enable trigger frame with RU index 0, data rate MCS2	`# echo "trigger_frame=1 1 1 2 5484 0 1000 0 0 0 1 0 0 0 1 0 1 0 65535 0 511 5 0 0 1 0x2102 0 0 90 0 0 0 0" >> /proc/ mwlan/ adapter0/config`
7	Start TX_frame Refer to the TX_frame command descriptions in Section 2.11.	`# echo "tx_frame=1 0x2102 0xabababab 0x256 0 20 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 05:43:3f:c4:51" >> /proc/mwlan/ adapter0/config`

Figure 3, Figure 4, and Figure 5 show the RU index values for the 20 MHz, 40 MHz, and 80 MHz bandwidths respectively.



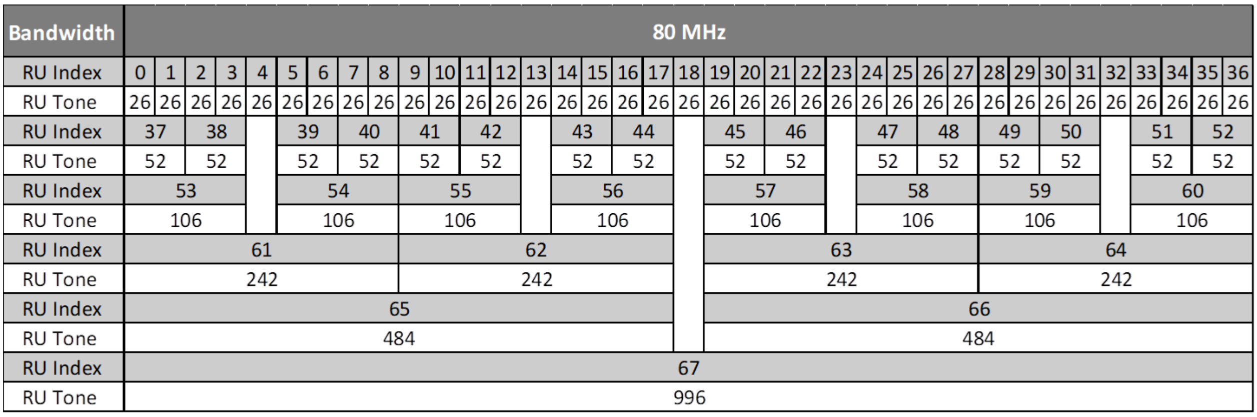
Bandwidth	20 MHz								
RU Index	0	1	2	3	4	5	6	7	8
RU Tone	26	26	26	26	26	26	26	26	26
RU Index	37		38			39		40	
RU Tone	52		52			52		52	
RU Index	53					54			
RU Tone	106					106			
RU Index	61								
RU Tone	242								

Figure 3. RU index values for 20 MHz bandwidth



Bandwidth	40 MHz																	
RU Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
RU Tone	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
RU Index	37		38			39		40		41		42			43		44	
RU Tone	52		52			52		52		52		52			52		52	
RU Index	53					54				55					56			
RU Tone	106					106				106					106			
RU Index	61									62								
RU Tone	242									242								
RU Index	65																	
RU Tone	484																	

Figure 4. RU index values for 40 MHz bandwidth



Bandwidth	80 MHz																																				
RU Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
RU Tone	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
RU Index	37		38			39		40		41		42			43		44			45		46			47		48		49		50			51		52	
RU Tone	52		52			52		52		52		52			52		52			52		52			52		52		52		52			52		52	
RU Index	53					54				55					56					57					58				59					60			
RU Tone	106					106				106					106					106					106				106					106			
RU Index	61									62										63									64								
RU Tone	242									242										242									242								
RU Index	65																			66																	
RU Tone	484																			484																	
RU Index	67																																				
RU Tone	996																																				

Figure 5. RU index values for 80 MHz bandwidth

Command to set the trigger frame response parameters on the DUT:

```
echo “he_tb_tx=<enable/exit> <Qnum> <AID> <AXQ0_MU_Timer> <TXPwr>” >> /proc/mwlan/
adapter0/config
```

Table 13. Command parameters

Parameter	Definition
`enable/exit`	Enter/exit trigger frame response mode 0 = exit trigger frame response mode (default) 1 = enter trigger frame response mode
`Qnum`	Transmit queue number that holds the trigger-based response packets. 1 = trigger-based test (default)
`AID`	Station ID Value set to 5.
`AXQ0_MU_Timer`	Arbitrary timer value to ensure SU packets are not transmitted. Units are in 8 ms. Set the value to be larger than the trigger frame interval. It is suggested to set the value to 400 (400 *8 = 3200 ms)
`TXPwr`	Transmit power in dBm.

Table 14 list the steps and Wi-Fi RF test mode commands for HE-Trigger response frame generation on the DUT.

Table 14. Steps for HE-trigger response frame generation on the DUT

Step	Operation	Command
1	Enable RF test mode	`# echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set radio mode to 5 GHz (2x2 mode)	`# echo "radio_mode=1 0" >> /proc/mwlan/adapter0/config`
3	Set the band to 5 GHz	`# echo "band=1" >> /proc/mwlan/adapter0/config`
4	Set the bandwidth to 20 MHz	`# echo "bw=0" >> /proc/mwlan/adapter0/config`
5	Set the channel to 36	`# echo "channel=36" >> /proc/mwlan/adapter0/config`
6	Start HE TB-TX with TX power set to 9 dBm.	`# echo "he_tb_tx=1 1 5 400 9" >> /proc/mwlan/adapter0/config`
7	Measure the TX-power value and EVM for the HE trigger response frame using an RF tester	

2.12.3 Testing standalone UL-OFDMA

The UL-OFDMA test can also be performed with just the DUT, without the golden unit.

In the standalone OFDMA test:

- The DUT sends UL-OFDMA signals directly without the golden unit.
- The parameter `<standalone_hetb>` is set to `2` for `trigger_frame` command.

Note: *This test setup is not typically used compared to the standard setup at the test lab. Consult with your test lab to determine if this test setup can be used for regulatory compliance testing.*

Table 15 lists the steps and Wi-Fi RF test mode commands for standalone UL-OFDMA on the DUT. The example applies to Wi-Fi 5 GHz, 1x1 path A, 80 MHz fequency band.

Table 15. Steps for standalone UL-OFDMA test example (Wi-Fi 5 GHz, 1x1 path A, 80 MHz fequency band)

Step	Operation	Command
1	Enable RF test mode	`# echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set the radio mode in 5 GHz (1x1 mode)	`# echo "radio_mode=3" >> /proc/mwlan/adapter0/config`
3	Set the band to 5 GHz	`# echo "band=1" >> /proc/mwlan/adapter0/config`
4	Set the bandwidth to 80 MHz	`# echo "bw=4" >> /proc/mwlan/adapter0/config`
5	Set TX power to 10 dBm	`# echo "tx_power=10 2 0" >> /proc/mwlan/adapter0/config`
6	Set the channel to 36	`# echo "channel=36" >> /proc/mwlan/adapter0/config`
7	Enable standalone trigger frame (enable RU index 67 and data rate MCS0)	`# echo "trigger_frame=1 2 1 2 5484 0 1000 0 0 2 1 0 0 0 1 0 1 0 65535 0 511 5 0 67 1 0 0 0 90 0 0 0 0" >> /proc/mwlan/adapter0/config`
8	Start `TX_frame` Refer to Section 2.11.	`# echo "tx_frame=1 0x2100 0xabababab 0x256 0 20 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 05:43:3f:c4:51" >> /proc/mwlan/adapter0/config`

2.13 Data rates

Table 16 shows 802.11n/a/g/b data rate IDs and data rates.

Table 16. 802.11n/a/g/b data rate ID

Data rate ID (in hex)	Data rate	Data rate ID (in hex)	Data rate
0x0	1 Mbps	0xe	HT_MCS 0
0x1	2 Mbps	0xf	HT_MCS 1
0x2	5.5 Mbps	0x10	HT_MCS 2
0x3	11 Mbps	0x11	HT_MCS 3
0x4	Reserved	0x12	HT_MCS 4
0x5	6 Mbps	0x13	HT_MCS 5
0x6	9 Mbps	0x14	HT_MCS 6
0x7	12 Mbps	0x15	HT_MCS 7
0x8	18 Mbps	0x16	HT_MCS 8
0x9	24 Mbps	0x17	HT_MCS 9
0xa	36 Mbps	0x18	HT_MCS 10
0xb	48 Mbps	0x19	HT_MCS 11
0xc	54 Mbps	0x1a	HT_MCS 12
0xd	Reserved	0x1b	HT_MCS 13
		0x1c	HT_MCS 14
		0x1d	HT_MCS 15
		0x2e	HT_MCS 32

Table 17 shows 802.11ac data rate IDs and data rates.

Table 17. 802.11ac data rate ID

Data rate ID (in hex)	Data rate
0x100	VHT_SS1_MCS0
0x101	VHT_SS1_MCS1
0x102	VHT_SS1_MCS2
0x103	VHT_SS1_MCS3
0x104	VHT_SS1_MCS4
0x105	VHT_SS1_MCS5
0x106	VHT_SS1_MCS6
0x107	VHT_SS1_MCS7
0x108	VHT_SS1_MCS8
0x109	VHT_SS1_MCS9
0x110	VHT_SS2_MCS0
0x111	VHT_SS2_MCS1
0x112	VHT_SS2_MCS2
0x113	VHT_SS2_MCS3
0x114	VHT_SS2_MCS4
0x115	VHT_SS2_MCS5
0x116	VHT_SS2_MCS6
0x117	VHT_SS2_MCS7
0x118	VHT_SS2_MCS8
0x119	VHT_SS2_MCS9

Table 18 shows 802.11ac/802.11ax data rate IDs and data rates.

Table 18. 802.11ac/802.11ax data rate ID

Rate number format : (XYRR) X : 1 - 11ac VHT MCS rates, 2 - 11ax HE MCS rates Y: Number of streams. 1 - SS1, 2 - SS2 RR : MCS rate number	
Data rate ID XYRR	**Data rate**
802.11ac VHT MCS rates[1]	
0x1100	VHT_SS1_MCS0
0x1101	VHT_SS1_MCS1
0x1102	VHT_SS1_MCS2
0x1103	VHT_SS1_MCS3
0x1104	VHT_SS1_MCS4
0x1105	VHT_SS1_MCS5
0x1106	VHT_SS1_MCS6
0x1107	VHT_SS1_MCS7
0x1108	VHT_SS1_MCS8
0x1109	VHT_SS1_MCS9
0x1200	VHT_SS2_MCS0
0x1201	VHT_SS2_MCS1
0x1202	VHT_SS2_MCS2
0x1203	VHT_SS2_MCS3
0x1204	VHT_SS2_MCS4
0x1205	VHT_SS2_MCS5
0x1206	VHT_SS2_MCS6
0x1207	VHT_SS2_MCS7
0x1208	VHT_SS2_MCS8
0x1209	VHT_SS2_MCS9
802.11ax HE MCS rates	
0x2100	HE_SS1_MCS0
0x2101	HE_SS1_MCS1
0x2102	HE_SS1_MCS2
0x2103	HE_SS1_MCS3
0x2104	HE_SS1_MCS4
0x2105	HE_SS1_MCS5
0x2106	HE_SS1_MCS6
0x2107	HE_SS1_MCS7
0x2108	HE_SS1_MCS8

Table 18. 802.11ac/802.11ax data rate ID *...continued*

Rate number format : (XYRR) **X : 1 - 11ac VHT MCS rates, 2 - 11ax HE MCS rates** **Y: Number of streams. 1 - SS1, 2 - SS2** **RR : MCS rate number**	
Data rate ID **XYRR**	**Data rate**
0x2109	HE_SS1_MCS9
0x210a	HE_SS1_MCS10
0x210b	HE_SS1_MCS11
0x2200	HE_SS2_MCS0
0x2201	HE_SS2_MCS1
0x2202	HE_SS2_MCS2
0x2203	HE_SS2_MCS3
0x2204	HE_SS2_MCS4
0x2205	HE_SS2_MCS5
0x2206	HE_SS2_MCS6
0x2207	HE_SS2_MCS7
0x2208	HE_SS2_MCS8
0x2209	HE_SS2_MCS10
0x220a	HE_SS2_MCS11

[1] Not all VHT rates are available for all the bandwidths and Tx antenna configurations. The error code `0xE` means that the rate and hardware combination is not supported.

2.14 Wi-Fi channels

Table 19. Wi-Fi channel list

Channel number	Frequency	Channel number	Frequency	Channel number	Frequency
2.4 GHz channel					
1	2412	2	2417	3	2422
4	2427	5	2432	6	2437
7	2442	8	2447	9	2452
10	2457	11	2462	12	2467
13	2472	—	—	—	—
5 GHz channel					
36	5180	38	5190	40	5200
42	5210	44	5220	46	5320
48	5420	52	5260	54	5270
56	5280	58	5290	60	5300
62	5310	64	5320	100	5500
102	5510	104	5520	106	5530
108	5540	110	5550	112	5560
116	5580	118	5590	120	5600
122	5610	124	5620	126	5630
128	5640	132	5660	134	5670
136	5680	138	5690	140	5700
142	5710	144	5720	149	5745
151	5775	153	5765	155	5775
157	5785	159	5795	161	5805
165	5825	167	5835	169	5845
171	5855	173	5865	175	5875
177	5885	—	—	—	—

2.15 RF test mode command sequence examples

2.15.1 2.4 GHz TX command sequence using TX_continuous

TX on radio 1 configured for 2.4 GHz, RF channel 6, 20 MHz bandwidth, 8 dBm target power with MCS modulation, HE SS1 MCS8 rate, and 1x1 mode (path A).

Table 20. 2.4 GHz TX command sequence using TX_continuous

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter1/config`
2	Set radio mode 1 in 2.4 GHz band with 1x1 mode (path A):	`echo "radio_mode=0 11" >> /proc/mwlan/adapter1/config`
3	Set TX antenna mode (path A)	`echo "tx_antenna=1" >> /proc/mwlan/adapter1/config`
4	Set 20 MHz bandwidth	`echo "bw=0" >> /proc/mwlan/adapter1/config`
5	Set RF channel 6	`echo "channel=6" >> /proc/mwlan/adapter1/config`
6	Set TX power to 8 dBm, MCS modulation and TX signal path A	`echo "tx_power=8 2 0" >> /proc/mwlan/adapter1/config`
7	Start TX with continuous wave mode at HE SS1 MCS8 rate with a 0xAAA pattern	`echo "tx_continuous=1 1 0xAAA 0 3 0x2108" >> / proc/mwlan/adapter1/config`
8	Stop TX continuous wave	`echo "tx_continuous=0 1 0xAAA 0 3 0x2108" >> /proc/mwlan/adapter1/config`

2.15.2 5 GHz RX command sequence

RX on radio 0 configured for 5 GHz, RF channel 36, 40 MHz bandwidth, and 2x2 path A+B.

Table 21. 5 GHz RX command sequence

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set radio mode 0 in 5 GHz band with 2x2 mode (Path A+B):	`echo "radio_mode=1 0" >> /proc/mwlan/adapter0/config`
3	Set 5 GHz band	`echo "band=1" >> /proc/mwlan/adapter0/config`
4	Set 40 MHz bandwidth	`echo "bw=1" >> /proc/mwlan/adapter0/config`
5	Set RX antenna mode (both path)	`echo "rx_antenna=3" >> /proc/mwlan/adapter0/config`
6	Set channel 36	`echo "channel=36" >> /proc/mwlan/adapter0/config`
7	Reset the packet error rate	`echo "get_and_reset_per" >> /proc/mwlan/adapter0/config`
8	Send a number of packets to the DUT	
9	Get and reset the packet error rate	`echo "get_and_reset_per" >> /proc/mwlan/adapter0/config`

10	Verify the packet-related parameters	`cat /proc/mwlan/adapter0/config`
		Output:
		`…`
		`total rx pkt count=1500`
		`rx multicast/broadcast pkt count=1500`
		`rx fcs error pkt count=0`
		`…`

2.16 RF test mode command sequence examples for 88W9098

2.16.1 5 GHz TX command sequence using tx_frame

TX on radio 0 configured for 5 GHz RF channel 44, 20 MHz bandwidth, 10 dBm target power with MCS modulation, HE SS1 MCS11 rate, and 1x1 mode (path A).

Table 22. 5 GHz TX command sequence using tx_frame

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set radio mode 0 in 5 GHz band with 1x1 mode (path A)	`echo "radio_mode=3 0" >> /proc/mwlan/adapter0/config`
3	Set TX antenna mode (path A)	`echo "tx_antenna=1" >> /proc/mwlan/adapter0/config`
4	Set 20 MHz bandwidth	`echo "bw=0" >> /proc/mwlan/adapter0/config`
5	Set RF channel 44	`echo "channel=44" >> /proc/mwlan/adapter0/config`
6	Set TX power to 10 dBm, MCS modulation and TX signal path A	`echo "tx_power=10 2 0" >> /proc/mwlan/adapter0/config`
7	Start TX at HE SS1 MCS11 rate	`echo "tx_frame=1 0x210b 0xAAA 0x100 1 20 0 0 0 1 0 0 0 -1 -1 -1 -1 -1 -1 -1 05:43:3f:c4:51" >> /proc/mwlan/adapter0/config`
8	Stop TX	`echo "tx_frame=0" >> /proc/mwlan/adapter0/config`

2.16.2 5 GHz TX command sequence using tx_frame for HE-ER SU

TX on radio 0 configured for 5 GHz RF channel 44, 20 MHz bandwidth, 10 dBm target power with MCS modulation, HE SS1 MCS1 rate, short preamble, and 1x1 mode (path A) for HE-ER SU.

Table 23. 5 GHz TX command sequence using tx_frame for HE-ER SU

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set radio mode 0 in 5 GHz band with 1x1 mode (path A)	`echo "radio_mode=3 0 " >> /proc/mwlan/adapter0/config`
3	Set TX antenna mode (path A)	`echo "tx_antenna=1" >> /proc/mwlan/adapter0/config`
4	Set 20 MHz bandwidth	`echo "bw=0" >> /proc/mwlan/adapter0/config`
5	Set RF channel 44	`echo "channel=44" >> /proc/mwlan/adapter0/config`
6	Set TX power to 10 dBm, MCS modulation and TX signal path A	`echo "tx_power=10 2 0" >> /proc/mwlan/adapter0/config`
7	Start TX at HE SS1 MCS1 rate and short preamble for HE-ER SU[1]	`echo "tx_frame=1 0x2101 0xAAA 0x100 1 20 1 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 05:43:3f:c4:51" >> /proc/ mwlan/adapter0/config`
8	Stop TX	`echo "tx_frame=0" >> /proc/mwlan/adapter0/config`

[1]
- 242 tones RU: MCS supported are MCS0-2. Set signalBw to 0 and MCS to 0, 1 or 2.
- 106 tone RU: MCS supported is MCS0. Set signalBw to 1 and MCS to 0
- Set short preamble to 0x1 in tx_frame command for HE-ER SU in RF_TEST_MODE
- preambleType: Use the default value (-1) unless required.

  For legacy 802.11b: preamble type
  - 0 = long
  - 1 = short

  For legacy 802.11g: not valid

  For 802.11n: GreenField PPDU indicator
  - 0 = HT-mix
  - 1 = HT-GF

  For 802.11ac: not valid

  For 802.11ax: PPDU type
  - 0 = HE-SU
  - 1 = HE-EXT-SU
  - 2 = HE-MU
  - 3 = HE-Trigger-based

2.16.3 2.4 GHz TX command sequence using tx_frame

TX on radio 1 configured for 2.4 GHz RF channel 11, 40 MHz bandwidth, 9 dBm target power with MCS modulation, HE SS1 MCS4 rate, and 1x1 mode (path B).

Table 24. 2.4 GHz TX command sequence using tx_frame

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter1/config`
2	Set radio mode 1 in 2.4 GHz band with 1x1 mode (path B):	`echo "radio_mode=0 14" >> /proc/mwlan/adapter1/config`
3	Set TX antenna mode (path B)	`echo "tx_antenna=2" >> /proc/mwlan/adapter1/config`
4	Set 40 MHz bandwidth	`echo "bw=1" >> /proc/mwlan/adapter1/config`
5	Set RF channel 11	`echo "channel=11" >> /proc/mwlan/adapter1/config`
6	Set TX power to 9 dBm, MCS modulation and TX signal path B	`echo "tx_power=9 2 1" >> /proc/mwlan/adapter1/config`
7	Start TX at HE SS1 MCS4 rate	`echo "tx_frame=1 0x2104 0xAAA 0x100 1 20 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 05:43:3f:c4:51" >> /proc/mwlan/adapter1/config`
8	Stop TX	`echo "tx_frame=0" >> /proc/mwlan/adapter1/config`

2.16.4 5 GHz TX command sequence using tx_continuous

TX on radio 0 configured for 5 GHz RF channel 36, 40 MHz bandwidth, 11 dBm target power with MCS modulation, HE SS1 MCS9 rate, and 2x2 mode (both paths).

Table 25. 5 GHz TX command sequence using tx_continuous

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set radio mode 0 in 5 GHz band with 2x2 mode (Path A+B)	`echo "radio_mode=1 0" >> /proc/mwlan/adapter0/config`
3	Set TX antenna mode (both path)	`echo "tx_antenna=3" >> /proc/mwlan/adapter0/config`
4	Set 40 MHz bandwidth	`echo "bw=1" >> /proc/mwlan/adapter0/config`
5	Set RF channel 36	`echo "channel=36" >> /proc/mwlan/adapter0/config`
6	Set TX power to 11 dBm, MCS modulation and TX signal path A + path B	`echo "tx_power=11 2 2" >> /proc/mwlan/adapter0/config`
7	Start TX at HE SS1 MCS9 rate	`echo "tx_continuous=1 0 0xAAA 0 3 0x2109" >> /proc/mwlan/adapter0/config`
8	Start TX continuous wave mode at HE SS1 MCS9 rate	`echo "tx_continuous=1 1 0xAAA 0 3 0x2109" >> /proc/mwlan/adapter0/config`
9	Stop TX continuous wave mode	`echo "tx_continuous=0 1 0xAAA 0 3 0x2109" >> /proc/mwlan/adapter0/config`

Note: *For TX, refer to either step 7 or step 8 based on whether the continuous wave mode is enabled or disabled.*

2.16.5 2.4 GHz TX command sequence using tx_continuous

TX on radio 1 configured for 2.4 GHz RF channel 6, 20 MHz bandwidth, 8 dBm target power with MCS modulation, HE SS1 MCS8 rate, and 1x1 mode (path A).

Table 26. 2.4 GHz TX command sequence using tx_continuous

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter1/config`
2	Set radio mode 1 in 2.4 GHz band with 1x1 mode (path A):	`echo "radio_mode=0 11" >> /proc/mwlan/adapter1/config`
3	Set TX antenna mode (path A)	`echo "tx_antenna=1" >> /proc/mwlan/adapter1/config`
4	Set 20 MHz bandwidth	`echo "bw=0" >> /proc/mwlan/adapter1/config`
5	Set RF channel 6	`echo "channel=6" >> /proc/mwlan/adapter1/config`
6	Set TX power to 8 dBm, MCS modulation and TX signal path A	`echo "tx_power=8 2 0" >> /proc/mwlan/adapter1/config`
7	Start TX at HE SS1 MCS8 rate	`echo "tx_continuous=1 0 0xAAA 0 3 0x2108" >> /proc/mwlan/adapter1/config`
8	Start TX with continuous wave mode at HE SS1 MCS8 rate	`echo "tx_continuous=1 1 0xAAA 0 3 0x2108" >> /proc/mwlan/adapter1/config`
9	Stop TX continuous wave mode	`echo "tx_continuous=0 1 0xAAA 0 3 0x2108" >> /proc/mwlan/adapter1/config`

Note: *For TX, refer to either step 7 or step 8 based on whether continuous wave mode is enabled or disabled.*

2.16.6 5 GHz RX command sequence

RX on radio 0 configured for 5 GHz RF channel 36, 40 MHz bandwidth, and 2x2 path A+B.

Table 27. 5 GHz RX command sequence

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter0/config`
2	Set radio mode 0 in 5 GHz band with 2x2 mode (Path A+B):	`echo "radio_mode=1 0" >> /proc/mwlan/adapter0/config`
3	Set 5 GHz band	`echo "band=1" >> /proc/mwlan/adapter0/config`
4	Set 40 MHz bandwidth	`echo "bw=1" >> /proc/mwlan/adapter0/config`
5	Set RX antenna mode (both path)	`echo "rx_antenna=3" >> /proc/mwlan/adapter0/config`
6	Set channel 36	`echo "channel=36" >> /proc/mwlan/adapter0/config`
7	Reset the packet error rate	`echo "get_and_reset_per" >> /proc/mwlan/adapter0/config`
8	Inject a definite number of packets at a specific power level	
9	Get and reset the packet error rate	`echo "get_and_reset_per" >> /proc/mwlan/adapter0/config`
10	Verify the packet-related parameters	`cat /proc/mwlan/adapter0/config`

2.16.7 2.4 GHz RX command sequence

RX on radio 1 configured for 2.4 GHz RF channel 6, 20 MHz bandwidth, and 1x1 path A.

Table 28. 2.4 GHz RF command sequence

Step	Operation	Command
1	Enable the RF test mode	`echo "rf_test_mode=1" >> /proc/mwlan/adapter1/config`
2	Set radio mode 1 in 2.4 GHz band with 1x1 mode (Path A):	`echo "radio_mode=0 11" >> /proc/mwlan/adapter1/config`
3	Set 2.4 GHz band	`echo "band=0" >> /proc/mwlan/adapter1/config`
4	Set 20 MHz bandwidth	`echo "bw=0" >> /proc/mwlan/adapter1/config`
5	Set RX antenna mode (path A)	`echo "rx_antenna=1" >> /proc/mwlan/adapter1/config`
6	Set channel 6	`echo "channel=6" >> /proc/mwlan/adapter1/config`
7	Reset the packet error rate	`echo "get_and_reset_per" >> /proc/mwlan/adapter1/config`
8	Inject a definite number of packets at a specific power level	
9	Get and reset the packet error rate	`echo "get_and_reset_per" >> /proc/mwlan/adapter1/config`
10	Verify the packet-related parameters	`cat /proc/mwlan/adapter1/config`

3 Bluetooth RF test mode

This section describes the Bluetooth/Bluetooth LE RF test mode feature execution on the i.MX 8M Quad Android platform using the NXP *btapp utility*. The *btapp utility* is used to run HCI commands on Android OS. Contact your NXP support representative to get the *btapp utility* file.

Note: *Bluetooth cannot be accessed simultaneously from UI and btapp utility. Use btapp utility to reboot the platform after RF operations. [not clear – What is UI (user interface)? How do we use UI?*

3.1 Install the btapp utility

Bluetooth UART port name

This section shows how to set *btapp* utility for the Bluetooth/Bluetooth LE RF test mode validation.

Note: *In this section, the Bluetooth UART port "ttymxc2" is used in the command line inputs. Replace "ttymxc2" with the port name of your device.*

Command to check the UART port name of your device:

```
evk_8mq:/ # ls /dev/ttymxc*
```

Step 1 - Contact your NXP support representative to receive the *btapp* utility file.

Step 2 – Bring up i.MX 8M Quad evaluation board with Android BSP (refer to [2]).

Step 3 – Use `adb` command line tool on a PC to disable `verity` on the i.MX board then reboot the device.

```
nxp@nxp:~$ adb root
nxp@nxp:~$ adb disable-verity
nxp@nxp:~$ adb reboot
```

Step 4 – Use `adb` command line tool on a PC to copy *btapp* executable file to i.MX 8M Quad evaluation board.

```
nxp@nxp:~$ adb root
nxp@nxp:~$ adb remount
nxp@nxp:~$ adb push btapp_imx_android /system/bin/
```

Step 5 – Set the evaluation board into super user mode.

```
evk_8mq:/ # su
```

Step 6 – Issue HCI reset command to check that *btapp* utility is working.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x03 0x03
```

Command output example:

```
ogf:3, ocf:3 opcode:c03, argc = 2
< HCI Command: ogf 0x03, ocf 0x0003, plen 4 01 03 0C 00
> HCI Event: 0x0e plen 4 01 03 0C 00
```

Note: *The "**00**" value bold in the HCI Event return means that the HCI command was successfully executed. The same HIC event return value applies to all the HCI commands in this section.*

3.2 Enable test mode for qualification

This section describes the method to enter test mode for qualification.

The `HCI_Enable_Device_Under_Test_Mode` command allows the local BR/EDR controller to enter test mode via LMP test commands. For more details about the commands, refer to [1].

Step 1 – Issue `HCI Reset` command.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x03 0x03
```

Command output example:

```
ogf:3, ocf:3 opcode:c03, argc = 2
< HCI Command: ogf 0x03, ocf 0x0003, plen 4 01 03 0C 00
> HCI Event: 0x0e plen 4 01 03 0C 00
```

Step 2 – Enable BR/EDR scan.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x03 0x001a 0x03
```

Command output example:

```
ogf:3, ocf:1A opcode:c1a, argc = 3
< HCI Command: ogf 0x03, ocf 0x001a, plen 5
01 1A 0C 01 03 > HCI Event: 0x0e plen 4 01 1A 0C 00
```

Step 3 – Issue event filter command.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x03 0x0005 0x02 0x00 0x02
```

Command output example:

```
ogf:3, ocf:5 opcode:c05, argc = 5
< HCI Command: ogf 0x03, ocf 0x0005, plen 7
01 05 0C 03 02 00 02 > HCI Event: 0x0e plen 4 01 05 0C 00
```

Step 4 – Enable test mode for qualification.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x06 0x0003
```

Command output example:

```
ogf:6, ocf:3 opcode:1803, argc = 2
< HCI Command: ogf 0x06, ocf 0x0003, plen 4
01 03 18 00 > HCI Event: 0x0e plen 4 01 03 18 00
```

3.3 Disable Bluetooth scanning

Before issuing an RF test mode transmit or receive command, issue an HCI reset and disable Bluetooth scanning. Bluetooth LE Scan and BR/EDR scan should not be ON while any RF test is ongoing.

Step 1 – Issue HCI reset command.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x03 0x03
```

Command output example:

```
ogf:3, ocf:3 opcode:c03, argc = 2
< HCI Command: ogf 0x03, ocf 0x0003, plen 4 01 03 0C 00
> HCI Event: 0x0e plen 4 01 03 0C 00
```

Step 2 – Disable BR/EDR scan.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x03 0x001a 0x00
```

Command output example:

```
ogf:3, ocf:1A opcode:c1a, argc = 3
< HCI Command: ogf 0x03, ocf 0x001a, plen 5
01 1A 0C 01 00 > HCI Event: 0x0e plen 4 01 1A 0C 00
```

Step 3 – Disable Bluetooth LE scan.

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x08 0x000C 0x00 0x00
```

Command output example:

```
ogf:8, ocf:C opcode:200c, argc = 4
< HCI Command: ogf 0x08, ocf 0x000C, plen 6
01 0C 20 02 00 00 > HCI Event: 0x0e plen 4 01 0C 20 00
```

3.4 RF test mode for Bluetooth classic

This section describes RF test mode commands, and the usage of Bluetooth BR/EDR RF test mode feature in the Android BSP.

3.4.1 BR/EDR receiver test

Command to perform BR/EDR receiver test (disable Bluetooth scanning before issuing the command).

```
evk_8mq:/ # btapp_imx_android ttymxc2 <ogf> <ocf> <test_scenario> <tx_frequency>
<rx_frequency> <test_packet_type> <expected_number_of_packets>
<length_of_test_data> <tx_am_address> <transmitter_BD_address> <report_error_packets>
```

Where:

Table 29. BR/EDR receiver test command parameters

Parameter	Length (octet)	Definition
`ogf`	1	0x3F
`ocf`	2	0x0018
`test_scenario`	1	Test scenario • 0x01 = receiver test, 0–pattern • 0x02 = receiver test, 1–pattern • 0x03 = receiver test, 1010–pattern • 0x04 = receiver test, PRBS–pattern • 0x09 = receiver test, 1111 0000–pattern • 0xFF = abort test mode
`tx_frequency`	1	Transmit frequency f = (2402+k) MHz
`rx_frequency`	1	Receive frequency f = (2402+k) MHz
`test_packet_type`	1	Test packet type • 0x03 = DM1 • 0x04 = DH1 • 0x0A = DM3 • 0x0B = DH3 • 0x0E = DM5 • 0x0F = DH5 • 0x14 = 2-DH1 • 0x18 = 3-DH1 • 0x1A = 2-DH3 • 0x1B = 3-DH3 • 0x1E = 2-DH5 • 0x1F = 3-DH5
`expected_number_of_packets`	4	Expected number of packets to transmit.

Table 29. BR/EDR receiver test command parameters*...continued*

Parameter	Length (octet)	Definition
`length_of_test_data`	2	Length in bytes of the test packet type. The length must not be longer than the maximum size of the specified `test_packet_type`. • DM1: 17 data bytes maximum • DH1: 27 data bytes maximum • DM3: 121 data bytes maximum • DH3: 183 data bytes maximum • DM5: 224 data bytes maximum • DH5: 339 data bytes maximum • 2-DH1: 54 data bytes maximum • 3-DH1: 83 data bytes maximum • 2-DH3: 367 data bytes maximum • 3-DH3: 552 data bytes maximum • 2-DH5: 679 data bytes maximum • 3-DH5: 1021 data bytes maximum
`tx_am_address`	1	Indicates if the Bluetooth device (BD) address is present: • 0x00 = not present • 0x01 = present (default)
`transmitter_BD_address`	6	BD address of the transmitter. This is used to derive the access code
`report_error_packets`	1	Report error packets • 0x00 = none (default) • 0x01 to 0xFE = number of packets to report

Example of command to enable a receive test with DH5 packets on CH 0:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x3F 0x0018 0x01 0x00 0x00 0x0F 0xDC 0x05 0x00 0x00
 0x25 0x00 0x01 0xEE 0xFF 0xC0 0x88 0x00 0x00 0x00
```

Command output example:

```
ogf:3F, ocf:18 opcode:fc18, argc = 20
< HCI Command: ogf 0x3f, ocf 0x0018, plen 22
01 18 FC 12 01 00 00 0F DC 05 00 00 25 00 01 EE FF C0 88 00 00 00
> HCI Event: 0x0e plen 4 01 18 FC 00
```

Note: *The command sets the receive test parameters. An HCI reset command is required after the test to resume normal Bluetooth operations.*

Command to end the receiver test:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x3F 0x0018 0xFF
```

Command output example:

```
ogf:3F, ocf:18 opcode:fc18, argc = 20
< HCI Command: ogf 0x3f, ocf 0x0018, plen 5
01 18 FC 12 FF
> HCI Event: 0xff plen 70
01 00 DC 05 00 00 4C 04 00 00 90 01 00 00 90 01 00 00 90 01
00 00 00 00 00 00 00 00 00 00 90 01 00 00 00 00 00 00 4C 04
00 00 64 00 00 00 60 C6 06 00 70 DA 05 00 30 00 00 00 D0 39
00 00 90 E2 00 00 D4 FF FF FF
```

Note: *The bolded hex value (0x0190) indicates that the DUT received 400 packets.*

Table 30 describes the content of the command output.

Table 30. Content of the command output for the end receiver test

Name	Length (octets)	Definition
Status	1	0x00 = completed 0x01 = aborted
Total Packets (Expected)	4	(in hexadecimal)
No Rx Count	4	(in hexadecimal)
Successful Correlation Count	4	(in hexadecimal)
HEC Match Count	4	(in hexadecimal)
HEC Match CRC Packets Count	4	(in hexadecimal)
Payload Hdr Error Count	4	(in hexadecimal)
CRC Error Count	4	(in hexadecimal)
Total Packet Received	4	(in hexadecimal)
Packet OK Count	4	(in hexadecimal)
Drop Packet Count	4	(in hexadecimal)
Packet Error Rate (%)	4	(in hexadecimal)
Total Number of Bits (Expected)	4	(in hexadecimal)
Total Number of Bit Errors (Lost + Drop)	4	(in hexadecimal)
Bit Error Rate	4	(in hexadecimal)
Total Number of Bytes (Received)	4	(in hexadecimal)
Total Number of Bit Errors (Received)	4	(in hexadecimal)
Average RSSI	4	(in decimal)

3.4.2 BR/EDR transmitter test

Command to start BR/EDR transmit test (disable Bluetooth scanning before issuing the command):

```
evk_8mq:/ # btapp_imx_android ttymxc2 <ogf> <ocf> <rx_on_start> <sync_on_start>
            <tx_on_start> <phd_on_start> <test_scenario> <hopping_mode> <tx_channel>
                <rx_channel> <tx_test_interval> <packet_type> <length> <whitening>
                    <number_of_test_packets> <tx_power>
```

Table 31. BR/EDR transmit test command parameters

Parameter	Length (octet)	Definition
`ogf`	1	0x3F
`ocf`	2	0x0019
`rx_on_start`	1	Set the four parameters to 0x80.
`sync_on_start`	1	
`tx_on_start`	1	
`phd_on_start`	1	
`test_scenario`	1	Test scenario • 0x01 = PATTERN_00 (data pattern: 0x00) • 0x02 = PATTERN_FF (data pattern: 0xFF) • 0x03 = PATTERN_55 (data pattern: 0x55) • 0x04 = PATTERN_PRBS (data pattern: 0xFE) • 0x09 = PATTERN_0F (data pattern: 0x0F) • 0xFF = exit test
`hopping_mode`	1	• 0x00 = fix frequency • 0x01 = hopping set
`tx_channel`	1	Transmit frequency = (2402+k) MHz, where k is the value of Tx Channel
`rx_channel`	1	Receive frequency = (2402+k) MHz, where k is the value of Rx Channel
`tx_test_interval`	1	User-defined test interval calculated as: *TX interval (in ms) = x * 1.25* where *x* is the interval multiplier based on Bluetooth packet type. Valid multiplier values: • 1 to 4 = values for 1-slot packets • 2 to 5 = values for 3-slot packets • 3 to 6 = values for 5-slot packets The default interval multiplier value is set to 1 for a 1-slot packet.

Table 31. BR/EDR transmit test command parameters*...continued*

Parameter	Length (octet)	Definition
`packet_type`	1	Transmit packet type • 0x03 = DM1 • 0x04 = DH1 • 0x0A = DM3 • 0x0B = DH3 • 0x0E = DM5 • 0x0F = DH5 • 0x14 = 2-DH1 • 0x18 = 3-DH1 • 0x1A = 2-DH3 • 0x1B = 3-DH3 • 0x1E = 2-DH5 • 0x1F = 3-DH5
`length`	2	Length of test data
`whitening`	1	Whitening • 0x00 = disabled • 0x01 = enabled
`number_of_test_packets`	4	Number of test packets • 0 = infinite (default)
`tx_power`	1	Signed value of TX power (dBm) • Range = -20 dBm to 12 dBm (default = 4 dBm)

Example of command to enable a transmit test with DM1 packets on CH 1 at 4 dBm:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x3F 0x0019 0x80 0x80 0x80 0x80 0x01 0x00 0x01 0x01
 0x0D 0x03 0x0F 0x00 0x00 0x00 0x00 0x00 0x00 0x04
```

Command output example:

```
ogf:3F, ocf:19 opcode:fc19, argc = 20
< HCI Command: ogf 0x3f, ocf 0x0019, plen 22
01 19 FC 12 80 80 80 80 01 00 01 01 0D 03 0F 00 00 00 00 00 00 04
> HCI Event: 0x0e plen 4 01 19 FC 00
```

Note: *This command sets the transmit test parameters. Issue HCI reset command after this test to resume normal Bluetooth operations.*

Command to end the transmit test:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x3F 0x0019 0x80 0x80 0x80 0x80 0xFF
```

Command output example:

```
ogf:3F, ocf:19 opcode:fc19, argc = 7
< HCI Command: ogf 0x3f, ocf 0x0019, plen 9
01 19 FC 12 80 80 80 80 FF
> HCI Event: 0x0e plen 4 01 19 FC 00
```

3.5 RF test mode for Bluetooth Low Energy (LE)

This section describes the different RF test mode commands, their usage, and the verification procedures of Bluetooth LE RF test mode feature in the Android BSP.

Note: *For more details about the standard commands mentioned in this section, refer to [1].*

3.5.1 Bluetooth LE receiver test

Command to start a Bluetooth LE receiver test (disable Bluetooth scanning before issuing the command):

```
evk_8mq:/ # btapp_imx_android ttymxc2 <ogf> <ocf> <rx_channel> <phy> <modulation_index>
```

Where:

Table 32. Bluetooth LE receiver test command parameters

Parameter	Length (octet)	Definition
ogf	1	OGF for Bluetooth LE controller commands set to 0x08.
ocf	1	OCF to write the Bluetooth LE receiver test set to 0x0033.
rx_channel	1	RF Channel to be used by the receiver channel number = (F- 2402) / 2 Input range: 0x00 to 0x27 Frequency range: 2402 MHz to 2480 MHz
phy	1	PHY to be used by the receiver: • 0x01 = Receiver set to use the LE 1M PHY • 0x02 = Receiver set to use the LE 2M PHY • 0x03 = Receiver set to use the LE Coded PHY
modulation_index	1	Modulation index: • 0x00 = Assume the transmitter has a standard modulation index • 0x01 = Assume the transmitter has a stable modulation index

Example of command to enable a receive test with LE 2M PHY on CH 1:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x08 0x0033 0x01 0x02 0x00
```

Command output example:

```
ogf:8, ocf:33 opcode:2033, argc = 5
< HCI Command: ogf 0x08, ocf 0x0033, plen 7 01 33 20 03 01 02 00
> HCI Event: 0x0e plen 4 01 33 20 00
```

Command to stop Bluetooth LE receiver test:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x08 0x001F
```

Command output example:

```
ogf:8, ocf:1F opcode:201f, argc = 2
< HCI Command: ogf 0x08, ocf 0x001f, plen 4 01 1F 20 00
> HCI Event: 0x0e plen 6 01 1F 20 00 00 00
```

3.5.2 Bluetooth LE transmitter test

Command to set the Bluetooth LE TX power before enabling the TX test (disable Bluetooth scanning before issuing the command):

```
evk_8mq:/ # btapp_imx_android ttymxc2 <ogf> <ocf> <tx_power>
```

Where:

Table 33. Bluetooth LE TX power command parameters

Parameter	Length (octet)	Definition
ogf	1	OGF for vendor specific command set to 0x3F
ocf	2	OCF to write Bluetooth LE transmit power set to 0x0087
tx_power	1	Transmit power level for Bluetooth LE in dBm. Default value = 0x00 Minimum value = 0xE2 (-30 dBm) Maximum value = 0x14 (20 dBm)

Example of command to set the TX power to 3 dBm:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x3F 0x87 0x03
```

Command output example:

```
ogf:3F, ocf:87 opcode:fc87, argc = 3
< HCI Command: ogf 0x3f, ocf 0x0087, plen 5
01 87 FC 01 03 > HCI Event: 0x0e plen 4 01 87 FC 00
```

Command to start a Bluetooth LE transmitter test:

```
evk_8mq:/ # btapp_imx_android ttymxc2 <ogf> <ocf> <tx_channel> <test_data_length>
<packet_payload> <phy>
```

Where:

Table 34. Bluetooth LE transmit test command parameters

Parameter	Length (octet)	Definition
ogf	1	OGF for Bluetooth LE controller commands set to 0x08.
ocf	2	OCF to write the Bluetooth LE transmit test set to 0x0034.
tx_channel	1	Transmit channel. N = (F-2402) / 2 Range: 0x00 to 0x27 Frequency range: 2402 MHz to 2480 MHz
test_data_length	1	Length in bytes of payload data in each packet. Range: 0x00 to 0xFF
packet_payload	1	Packet payload. Refer Bluetooth Core Spec v5.3 for more details.
phy	1	PHY to be used by the receiver: • 0x01 = Transmitter set to use Bluetooth LE 1M PHY. • 0x02 = Transmitter set to use Bluetooth LE 2M PHY. • 0x03 = Transmitter set to use Bluetooth LE coded PHY with S=8 data coding. • 0x04 = Transmitter set to use Bluetooth LE coded PHY with S=2 data coding.

Example of command to enable a transmit test with LE 2M PHY on CH 1:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x08 0x0034 0x01 0x01 0x00 0x02
```

Command output example:

```
ogf:8, ocf:34 opcode:2034, argc = 6
< HCI Command: ogf 0x08, ocf 0x0034, plen 8 01 34 20 04 01 01 00 02
> HCI Event: 0x0e plen 4 01 34 20 00
```

Command to stop Bluetooth LE transmit test:

```
evk_8mq:/ # btapp_imx_android ttymxc2 0x08 0x001F
```

Command output example:

```
ogf:8, ocf:1F opcode:201f, argc = 2
< HCI Command: ogf 0x08, ocf 0x001f, plen 4 01 1F 20 00
> HCI Event: 0x0e plen 6 01 1F 20 00 00 00
```

3.6 RF test mode command sequence examples

3.6.1 Bluetooth Classic TX test

TX test with DM1 packets on RF channel 0 at 4 dBm.

Table 35. Bluetooth Classic TX test command sequence

Step	Operation	Command
1	Perform an HCI reset	`btapp_imx_android ttymxc2 0x03 0x0003`
2	Disable Bluetooth BR/EDR scan	`btapp_imx_android ttymxc2 0x03 0x001a 0x00`
3	Disable Bluetooth LE scan	`btapp_imx_android ttymxc2 0x08 0x000C 0x00 0x00`
4	Transmit 4 dBm DM1 packets at 2402 MHz with data pattern 0x00	`btapp_imx_android ttymxc2 0x3F 0x0019 0x80 0x80 0x80 0x80 0x01 0x00 0x00 0x00 0x0D 0x03 0x0F 0x00 0x00 0x00 0x00 0x00 0x00 0x04`
5	End transmit	`btapp_imx_android ttymxc2 0x3F 0x0019 0x80 0x80 0x80 0x80 0xFF`

3.6.2 Bluetooth Classic RX test

Bluetooth Classic RX test

RX test with DM5 packets on RF channel 0.

Table 36. Bluetooth Classic RX test command sequence

Step	Operation	Command
1	Perform an HCI reset	`btapp_imx_android ttymxc2 0x03 0x0003`
2	Disable Bluetooth BR/EDR scan	`btapp_imx_android ttymxc2 0x03 0x001a 0x00`
3	Disable Bluetooth LE scan	`btapp_imx_android ttymxc2 0x08 0x000C 0x00 0x00`

4	Enable receive at 2402MHz for 3-DH5 packets from TX device 00:26:1F:EC:F6:4E	`btapp_imx_android ttymxc2 0x3F 0x0018 0x01 0x00 0x00`
		` 0x0F 0xDC 0x05 0x00 0x00 0x25 0x00 0x01 0x4E 0xF6 0xEC`
		` 0x1F 0x26 0x00 0x00`

5	Transmit packets to the DUT with an RF tester	

6	End receive test and get packet count	`btapp_imx_android ttymxc2 0x3F 0x0018 0xFF`
		Output:
		`< HCI Command: ogf 0x3f, ocf 0x0018, plen 1`
		`FF`
		`> HCI Event: 0xff plen 70`
		`01 00 DC 05 00 00 4C 04 00 00 90 01 00 00 90 01 00 00`
		` 90 01 00 00 00 00 00 00 00 00 00 00 ` **`90 01`** ` 00 00 00 00`
		` 00 00 4C 04 00 00 64 00 00 00 60 C6 06 00 70 DA 05 00`
		` 30 00 00 00 D0 39 00 00 90 E2 00 00 D4 FF FF FF`
		Note: *The hex value (0x0190) in bold indicates that the DUT received 400 packets.*

3.6.3 Bluetooth LE TX test

TX test with 2M LE packets on RF channel 0 at 4 dBm.

Table 37. Bluetooth LE TX test command sequence

Step	Operation	Command
1	Perform an HCI reset	`btapp_imx_android ttymxc2 0x03 0x0003`
2	Disable Bluetooth BR/EDR scan	`btapp_imx_android ttymxc2 0x03 0x001a 0x00`
3	Disable Bluetooth LE scan	`btapp_imx_android ttymxc2 0x08 0x000C 0x00 0x00`
4	Set TX power to 4 dBm	`btapp_imx_android ttymxc2 0x3F 0x87 0x04`
5	Transmit 4 dBm DM1 packets at 2402 MHz	`btapp_imx_android ttymxc2 0x08 0x0034 0x00 0x01 0x00 0x02`
6	End transmit	`btapp_imx_android ttymxc2 0x08 0x001F`

3.6.4 Bluetooth LE RX test

RX test with 1M LE packets on RF channel 0.

Table 38. Bluetooth LE RX test command sequence

Step	Operation	Command
1	Perform an HCI reset	`btapp_imx_android ttymxc2 0x03 0x0003`
2	Disable Bluetooth BR/EDR scan	`btapp_imx_android ttymxc2 0x03 0x001a 0x00`
3	Disable Bluetooth LE scan	`btapp_imx_android ttymxc2 0x08 0x000C 0x00 0x00`
4	Enable RX at 2402MHz LE 1M	`btapp_imx_android ttymxc2 0x08 0x0033 0x00 0x01 0x00`
5	Transmit packets to the DUT with an RF tester	

6	End receive test and get packet count	`btapp_imx_android ttymxc2 0x08 0x001F`
		Output:
		`< HCI Command: ogf 0x08, ocf 0x001f, plen 0`
		`> HCI Event: 0x0e plen 6`
		`01 1F 20 00 20 03`
		Note: *The hex value (0x0320) in bold indicates that the DUT received 800 packets.*

4 Abbreviations

Table 39. Abbreviations

Acronyms	Definition
BR	basic rate (Bluetooth terminology)
BT	Bluetooth
CS	Carrier suppression
DH	data high rate (Bluetooth terminology)
DM1	data medium 1 (Bluetooth terminology)
DUT	Device under test
EDR	enhanced data rate (Bluetooth terminology)
FW	Firmware
HCI	Host command interface
LE	low energy (Bluetooth terminology)
RF	Radio frequency
RX	Receive
TX	Transmit

5 References

[1] Bluetooth Core Specification v5.3 (link)

[2] UM11558 – Getting Started with NXP-based Wireless Modules on i.MX 8M Quad EVK Running Android OS (link)

6 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

7 Revision history

Table 40. Revision history

Document ID	Release date	Description
AN14365 v.1.0	16 December 2024	• Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.