AN14114

RF Test Mode on Linux OS

Rev. 2 — 15 April 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 Linux OS.

1 Introduction

This document provides an overview of how to enable and use the RF test mode on a Linux 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, Bluetooth, and IEEE 802.15.4 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
- IW611
- IW612
- AW611
- IW416

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

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

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 Linux. The commands apply to any Linux-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 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

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.2 Set TX/RX antenna configuration

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

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

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

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

Where:

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
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.3 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, IW612, and AW611.*

Table 1 lists the radio mode indexes for 88W9098.

Table 1. 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)
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:

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

Table 2 lists the radio mode indexes for IW612.

Table 2. Radio mode index values for IW612

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
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.4 Set the operating RF band

Command to set the RF band:

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

Where:

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
```

2.5 Set the channel bandwidth

Command to set the channel bandwidth:

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

Where:

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
```

2.6 Set the RF channel

Command to set the RF channel:

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

Where:

Parameter	Definition
ch	RF channel of operation ***Note:*** *For more information on the supported RF channels, refer to Section 2.13.*

Example of command to set the RF channel to 6:

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


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.7 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.

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.8 Set TX power

Command to set the TX power:

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

Where:

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.*

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.9 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
```

Where:

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 2.12).

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:

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


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.10 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< <NumPkt< <MaxPktExt< <BeamChange< <DCM<
 <Doppler< <MidamblePeriod< <QNum< <BSSID<" << /proc/mwlan/adapter0/config
```

Where:

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 2.12).
`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 0 (default) to 255
`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
`adv coding`	Advanced coding 0 = disable (default) 1 = enable
`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
`NumPkt`	Number of packets Set to default value -1.

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

Parameter	Definition
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

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 >> /proc/mwlan/adapter0/config
```

Example of command to stop any ongoing TX frame:

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


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.11 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 2.11.3.

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

2.11.1 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.

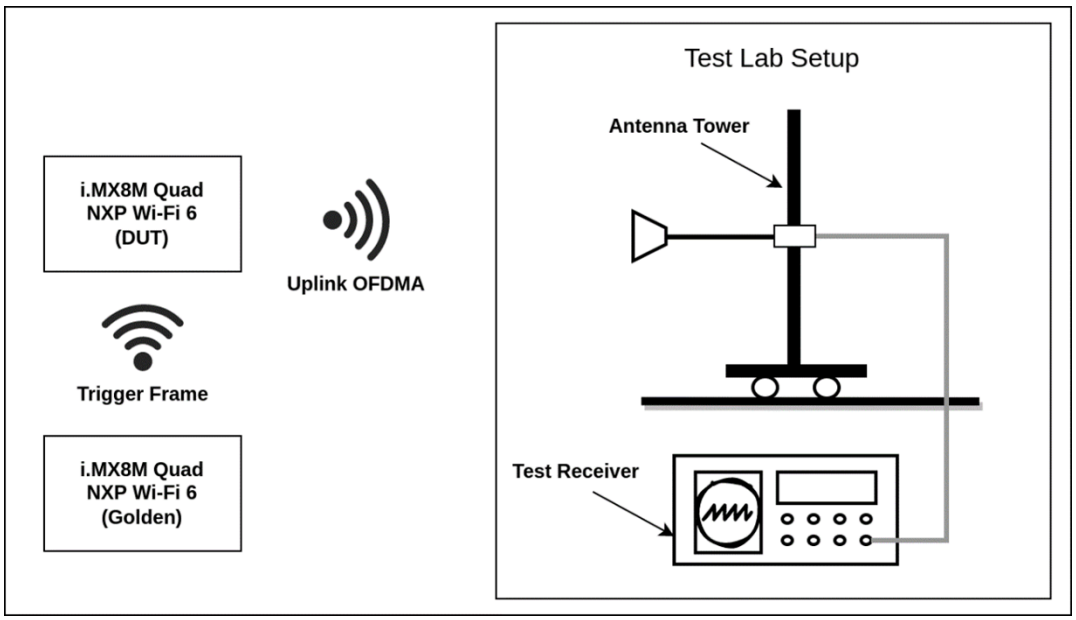

Figure 2. Test setup

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.11.2 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
```

Where:

Parameter	Definition
`enable_TX`	Enable transmit 0 = disable 1 = enable
`standalone_hetb`	Enable standalone UL-OFDMA (Section 2.11.3) 0 = disable 1 = enable
`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

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

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 2.12.
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 3 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 3. 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 band to 5 GHz	`# echo "band=1" >> /proc/mwlan/adapter0/config`
4	Set bandwidth to 20 MHz	`# echo "bw=0" >> /proc/mwlan/adapter0/config`
5	Set 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 0 1 2 5484 0 1000 0 0 0 1 0 0 0 1 0` `1 0 65535 0 511 5 0 0 1 2 0 0 90 0 0 0 0" >> /proc/mwlan/adapter0/config`
7	Start TX_frame Refer to the TX_frame command descriptions in Section 2.10.	`# echo "tx_frame=1 0x2100 0xabababab 200" >> /proc/mwlan/adapter0/config`

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

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

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

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
```

Where:

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

Table 4. 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 band to 5 GHz	`# echo "band=1" >> /proc/mwlan/adapter0/config`
4	Set bandwidth to 20 MHz	`# echo "bw=0" >> /proc/mwlan/adapter0/config`
5	Set 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	

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.11.3 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 `1` 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 5 lists the steps and Wi-Fi RF test mode commands for standalone UL-OFDMA on the DUT.

Table 5. Steps for standalone UL-OFDMA test

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 0" >> /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 the channel to 36	`# echo "channel=36" >> /proc/mwlan/adapter0/config`
6	Enable standalone trigger frame	`# echo "trigger_frame=1 1 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`
7	Start `TX_frame` Refer to Section 2.10.	`# echo "tx_frame=1 0x2100 0xabababab 200" >> /proc/mwlan/adapter0/config`

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.12 Data rates

Table 6 shows 802.11n/a/g/b data rate IDs and data rates.

Table 6. 802.11n/a/g/b data rate ID

Data rate ID	Data rate	Data rate ID	Data rate	Data rate ID	Data rate
1	1 Mbps	6	6 Mbps	15	HT_MCS 0
2	2 Mbps	7	9 Mbps	16	HT_MCS 1
3	5.5 Mbps	8	12 Mbps	17	HT_MCS 2
4	11 Mbps	9	18 Mbps	18	HT_MCS 3
5	Reserved	10	24 Mbps	19	HT_MCS 4
—	—	11	36 Mbps	20	HT_MCS 5
—	—	12	48 Mbps	21	HT_MCS 6
—	—	13	54 Mbps	22	HT_MCS 7
—	—	14	Reserved	—	—
23	HT_MCS8	31	HT_MCS16	39	HT_MCS24
24	HT_MCS9	32	HT_MCS17	40	HT_MCS25
25	HT_MCS10	33	HT_MCS18	41	HT_MCS26
26	HT_MCS11	34	HT_MCS19	42	HT_MCS27
27	HT_MCS12	35	HT_MCS20	43	HT_MCS28
28	HT_MCS13	36	HT_MCS21	44	HT_MCS29
29	HT_MCS14	37	HT_MCS22	45	HT_MCS30
30	HT_MCS15	38	HT_MCS23	46	HT_MCS31

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

Table 7 shows 802.11ac data rate IDs and data rates.

Table 7. 802.11ac data rate ID

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

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

Table 8 shows 802.11ac/802.11ax data rate IDs and data rates.

Table 8. 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 RR : MCS rate number	
Data rate ID **XYRR**	**Data rate**
802.11ac VHT MCS rates[1]	
1100	VHT_SS1_MCS0
1101	VHT_SS1_MCS1
1102	VHT_SS1_MCS2
1103	VHT_SS1_MCS3
1104	VHT_SS1_MCS4
1105	VHT_SS1_MCS5
1106	VHT_SS1_MCS6
1107	VHT_SS1_MCS7
1108	VHT_SS1_MCS8
1109	VHT_SS1_MCS9
802.11ax HE MCS rates	
2100	HE_SS1_MCS0
2101	HE_SS1_MCS1
2102	HE_SS1_MCS2
2103	HE_SS1_MCS3
2104	HE_SS1_MCS4
2105	HE_SS1_MCS5
2106	HE_SS1_MCS6
2107	HE_SS1_MCS7
2108	HE_SS1_MCS8
2109	HE_SS1_MCS9

[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.

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.13 Wi-Fi channels

Table 9. 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	—	—	—	—

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.14 RF test mode command sequence examples

2.14.1 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 2x2 mode (path A).

Table 10. 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 2x2 mode (Path A+B):	`echo "radio_mode=0 9" >> /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	`echo "tx_continuous=0" >> /proc/mwlan/adapter1/ config`

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

2.14.2 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 11. 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** …

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3 Bluetooth RF test mode

This section describes the commands to use RF test mode for Bluetooth Classic and Bluetooth Low Energy on the i.MX 8M Quad EVK with Linux. The commands apply to any Linux-based host.

Bluetooth RF Test Mode parameters use HCI commands following Bluetooth Core Specification v5.3 ([1]) and vendor-specific commands. For more information on vendor-specific commands, refer to the Bluetooth software user manual of your device.

The command parameters in this section are entered in hexadecimal little endian format.

In the command examples, `hci0` refers to the Bluetooth device.

Command to check the name of your device:

```
hciconfig -a
```

Note: *In this section, the bolded hex value 00 in the example command outputs means that the command was set successfully.*

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.1 Bluetooth Classic

This section describes how to use RF test mode commands for Bluetooth Classic.

3.1.1 Enable test mode for qualification

Sequence of commands to enable test mode for qualification.

- Perform an HCI reset.

```
hcitool -i hci0 cmd 0x03 0x0003
```

Command output example:

```
< HCI Command: ogf 0x03, ocf 0x0003, plen 0
> HCI Event: 0x0e plen 4
01 03 0C 00
```

- Enable BR/EDR scan.

```
hcitool -i hci0 cmd 0x03 0x001a 0x3
```

Command output example:

```
< HCI Command: ogf 0x03, ocf 0x001a, plen 1
03
> HCI Event: 0x0e plen 4
01 1A 0C 00
```

- Set event filter.

```
hcitool -i hci0 cmd 0x03 0x0005 0x02 0x00 0x02
```

Command output example:

```
< HCI Command: ogf 0x03, ocf 0x0005, plen 3
02 00 02
> HCI Event: 0x0e plen 4
01 05 0C 00
```

- Enable test mode for qualification

```
hcitool -i hci0 cmd 0x06 0x0003
```

Command output example:

```
< HCI Command: ogf 0x06, ocf 0x0003, plen 0
> HCI Event: 0x0e plen 4
01 03 18 00
```

Note: *The HCI reset command is used to resume normal Bluetooth operation after enabling test mode.*

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.1.2 Set the receive test parameters

Command to set the receive test parameters:

```
hcitool -i hci0 cmd 0x3F 0x0018 <TestScenario> <TXFrequency> <RXFrequency>
 <TestPacketType> <Expected Number of Packets> <Length of Test Data> <TX AM Address> <BD
 Address> <Report error packets>
```

where:

Parameter	Length (bytes)	Definition
OGF	1	0x3F
OCF	2	0x0018
TestScenario	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
TXFrequency	1	Transmit Frequency = (2402+k) MHz, where k is the value of TXChannel k range: 0x00 to 0x4F
RXFrequency	1	Receive Frequency = (2402+k) MHz, where k is the value of RXChannel k range: 0x00 to 0x4F
TestPacketType	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	The expected number of RX packets
Length of Test Data	2	The length of the test data ***Note:*** *This value should not be bigger than the maximum size of the specified test packet type*
TX AM Address	1	Default = 0x01
Transmitter BD Address	6	Transmitter's BD address
Report error packets	1	Report Error Packets • 0x00 = none (default) • 0x01 to 0xFE = number of packets to report

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

Example of command to set the receive test with the following parameters:

- RX Frequency: 2402 MHz
- Test Scenario: 0-pattern
- Test Packet Type: DH5
- Expected Number of Packets: 1500
- Transmitter BD Address: 00:26:1F:EC:F6:4E

```
hcitool -i hci0 cmd 0x3F 0x0018 0x01 0x00 0x00 0x0F 0xDC 0x05 0x00 0x00 0x25 0x00 0x01
 0x4E 0xF6 0xEC 0x1F 0x26 0x00
```

Command output example:

```
< HCI Command: ogf 0x3f, ocf 0x0018, plen 18
  01 00 00 0F DC 05 00 00 25 00 01 4E F6 EC 1F 26 00 00
> HCI Event: 0xfe plen 4
  01 18 FC 00
```

3.1.3 End the receive test

Command to end the receive test:

```
hcitool -i hci0 cmd 0x3F 0x0018 0xFF
```

Command output example:

```
< 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 bolded hex value (0x0190) indicates 400 packets were received by the DUT.*

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.1.4 Set the transmit test parameters

Command to set the transmit test parameters.

```
hcitool -i hci0 cmd 0x3F 0x0019 0x80 0x80 0x80 0x80 <TestScenario> <HoppingMode>
 <TXChannel> <RXChannel> <TXTestInternval> <PacketType> <Length> <Whitening> <Number of
 Test Packets> <TX Power>
```

where:

Parameter	Length (bytes)	Definition
OGF	1	0x3F
OCF	2	0x0019
RXOnStart	1	Set to 0x80.
SyntOnStart 1	1	
TXOnStart	1	
PhdOffStart	1	
TestScenario	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
HoppingMode	1	• 0x00 = fix frequency • 0x01 = hopping set
TXChannel	1	Transmit Frequency = (2402+k) MHz, where k is the value of TXChannel k range: 0x00 to 0x4F
RXChannel	1	Receive Frequency = (2402+k) MHz, where k is the value of RXChannel k range: 0x00 to 0x4F
TXTestInterval	1	Poll interval in frames for the link (In units per 1.25 ms)
PacketType	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	• 0x00 = disabled • 0x01 = enabled

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

Parameter	Length (bytes)	Definition
Number of Test Packets	4	Number of packets to transmit • 0 = infinite
TX Power	1	Signed value of TX power (dBm) • For IW612: range = -20 dBm to 19 dBm[1] • For all the other modules: range = -20 dBm to 12 dBm

[1] Depends on Bluetooth class

Example of command to the transmit test with the following parameters:

- TX Frequency: 2403 MHz
- Test Scenario: 0-pattern
- Hopping Mode: Fixed Frequency
- Packet Type: DM1
- TX power: 4 dBm

```
hcitool -i hci0 cmd 0x3F 0x0019 0x80 0x80 0x80 0x80 0x01 0x00 0x01 0x01 0x0D 0x03 0x0F
 0x00 0x00 0x00 0x00 0x00 0x00 0x04
```

Command output example:

```
< HCI Command: ogf 0x3f, ocf 0x0019, plen 18
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
```

3.1.5 End the transmit command

Issue the command to end the transmit test:

```
hcitool -i hci0 cmd 0x3F 0x0019 0x80 0x80 0x80 0x80 0xFF
```

Command output example:

```
< HCI Command: ogf 0x3f, ocf 0x0019, plen 18
80 80 80 80 FF 00 01 01 0D 03 0F 00 00 00 00 00 00 04
> HCI Event: 0xff plen 6
19 01 2F 0F 00 00
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.2 Bluetooth Low Energy (LE)

This section describes how to use RF test mode commands for Bluetooth LE.

3.2.1 Test Bluetooth LE receiver

To start a test where the DUT receives test reference packets at a fixed interval, use the `LE Receiver Test[V2]` command. For more details on the command, refer to *section 7.8.28* of [1].

```
hcitool -i hci0 cmd 0x08 0x0033 <RX Channel> <PHY> <Modulation Index>
```

Where:

Command parameter	Length (bytes)	Description
OGF	1	OGF for the 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
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 transmitter will have a standard modulation index 0x01 = Assume transmitter will have a stable modulation index

Command example to the receive test with the following parameters:

- RX channel: 2404 MHz
- PHY: LE 2M
- Standard modulation index for the transmitter

```
hcitool -i hci0 cmd 0x08 0x0033 0x01 0x02 0x00
```

Command output example:

```
< HCI Command: ogf 0x08, ocf 0x0033, plen 3
01 02 00
> HCI Event: 0x0e plen 4
01 33 20 00
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.2.2 Set Bluetooth LE TX power

Command to set the Bluetooth LE transmit power level:

```
hcitool -i hci0 cmd 0x3F 0x87 <TX_power>
```

Where:

Command parameter	Length (bytes)	Description
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 0x04 = 4 dBm Minimum value = 0xE2 (-30 dBm) Maximum value = 0x14 (20 dBm)

```
hcitool -i hci0 cmd 0x3F 0x87 0x04
```

Command output example:

```
< HCI Command: ogf 0x3f, ocf 0x0087, plen 1
04
> HCI Event: 0x0e plen 4
01 87 FC 00
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.2.3 Test Bluetooth LE transmitter

To start a test where the DUT generates test reference packets at a fixed interval, use the `LE Transmitter Test[V2]` command. For more details on the command, refer to *section 7.8.29* of [1].

```
hcitool -i hci0 cmd 0x08 0x0034 <TX Channel> <Test Data Length> <Packet Payload>
```

Where:

Command parameter	Length (bytes)	Description
`OGF`	1	OGF for the LE Controller commands set to 0x08
`OCF`	1	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	0x00 to 0xFF Length in bytes of payload data in each packet
`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 the LE 1M PHY 0x02 - Transmitter set to use the LE 2M PHY 0x03 - Transmitter set to use the LE Coded PHY with S=8 data coding 0x04 - Transmitter set to use the LE Coded PHY with S=2 data coding

Example of command to set the transmit test with the following parameters:

- TX Channel: 2404 MHz
- Test Data Length: 1 Byte
- Packet Payload: PRBS9 sequence '11111111100000111101…'
- PHY: LE 2M

```
hcitool -i hci0 cmd 0x08 0x0034 0x01 0x01 0x00 0x02
```

Command output example:

```
< HCI Command: ogf 0x08, ocf 0x0034, plen 4
01 01 00 02
> HCI Event: 0x0e plen 4
01 34 20 00
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.2.4 End Bluetooth LE test

To stop any test which is in progress, use the `LE Test End` command. For more details on the command, refer to *section 7.8.30* of [1].

```
hcitool -i hci0 cmd 0x08 0x001F
```

Command output example:

```
< HCI Command: ogf 0x08, ocf 0x001f, plen 0
> HCI Event: 0x0e plen 6
01 1F 20 00 00 00
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.3 RF test mode command sequence examples

3.3.1 Bluetooth Classic TX test

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

Table 12. Bluetooth Classic TX command sequence

Step	Operation	Command
1	Perform an HCI reset	`hcitool -i hci0 cmd 0x03 0x0003`
2	Enable BR/EDR scan	`hcitool -i hci0 cmd 0x03 0x001a 0x3`
3	Set event filter	`hcitool -i hci0 cmd 0x03 0x0005 0x02 0x00 0x02`
4	Enable direct test mode	`hcitool -i hci0 cmd 0x06 0x0003`
5	Transmit 4 dBm DM1 packets at 2402 MHz with data pattern 0x00	`hcitool -i hci0 cmd 0x3F 0x0019 0x80 0x80 0x80 0x80 0x01 0x00 0x00 0x00 0x0D 0x03 0x0F 0x00 0x00 0x00 0x00 0x00 0x00 0x04`
6	End transmit	`hcitool -i hci0 cmd 0x3F 0x0019 0x80 0x80 0x80 0x80 0xFF`

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.3.2 Bluetooth Classic RX Test

RX test with DM5 packets on RF channel 0

Table 13. Bluetooth Classic RX command sequence

Step	Operation	Command
1	Perform an HCI reset	`hcitool -i hci0 cmd 0x03 0x0003`
2	Enable BR/EDR scan	`hcitool -i hci0 cmd 0x03 0x001a 0x3`
3	Set event filter	`hcitool -i hci0 cmd 0x03 0x0005 0x02 0x00 0x02`
4	Enable direct test mode	`hcitool -i hci0 cmd 0x06 0x0003`
5	Enable receive at 2402 MHz for 3-DH5 packets from TX device 00:26:1F:EC:F6:4E	`hcitool -i hci0 cmd 0x3F 0x0018 0x01 0x00 0x00 0x0F 0xDC 0x05 0x00 0x00 0x25 0x00 0x01 0x4E 0xF6 0xEC 0x1F 0x26 0x00 0x00`
6	Transmit packets to the DUT with an RF tester	
7	End receive test and get packet count	`hcitool -i hci0 cmd 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 00 00 90 01 00 00 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 400 packets were received by the DUT.*

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

3.3.3 Bluetooth LE TX test

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

Table 14. Bluetooth LE TX command sequence

Step	Operation	Command
1	Perform an HCI reset	`hcitool -i hci0 cmd 0x03 0x0003`
2	Set TX power to 4 dBm	`hcitool -i hci0 cmd 0x3F 0x87 0x04`
3	Transmit 4 dBm DM1 packets at 2402 MHz	`hcitool -i hci0 cmd 0x08 0x0034 0x00 0x01 0x00 0x02`
4	End transmit	`hcitool -i hci0 cmd 0x08 0x001F`

3.3.4 Bluetooth LE RX test

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

Table 15. Bluetooth LE RX command sequence

Step	Operation	Command
1	Perform an HCI reset	`hcitool -i hci0 cmd 0x03 0x0003`
2	Enable RX at 2402MHz LE 1M	`hcitool -i hci0 cmd 0x08 0x0033 0x00 0x01 0x00`
3	Transmit packets to the DUT with an RF tester	
4	End receive test and get packet count	`hcitool -i hci0 cmd 0x08 0x001F` Output: `< HCI Command: ogf 0x08, ocf 0x001f, plen 0` `> HCI Event: 0x0e plen 6` `01 1F 20 00 **20 03**` ***Note:*** *The bolded hex value (0x0320) indicates 800 packets were received by the DUT.*

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

4 RF test mode for 802.15.4

This section describes the commands to use RF Test Mode for IEEE 802.15.4 on the i.MX 8M Quad EVK with Linux. The commands apply to any Linux-based host.

Note: *The commands use the ot-ctl utility, which needs to be cross-compiled with the ot-daemon tool. For more information on building ot-ctl utility, refer to [2].*

4.1 Enable or disable 802.15.4 RF test mode

Command to enable 802.15.4 RF test mode:

```
ot-ctl mfgcmd 1
```

Return data: Done

Command to disable 802.15.4 RF test mode:

```
ot-ctl mfgcmd 0
```

Return data: Done

4.2 Reset 802.15.4 radio

Command to reset 802.15.4 RF block:

```
ot-ctl mfgcmd 80 0
```

Return data: Done

4.3 Get the RF channel

Command to get the RF channel.

```
ot-ctl mfgcmd 11
```

Return data: Currently set channel

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

4.4 Set the RF channel

Command to set the RF channel.

```
ot-ctl mfgcmd 12 <Channel Number>
```

Where:

Command parameter	Description
`Channel Number`	RF Channel Input integer range of 11 (default) to 26

Return data: Done

Example of command to set the 802.15.4 channel to 20

```
ot-ctl mfgcmd 12 20
```

4.5 Get TX power level value

Command to get TX power level value:

```
ot-ctl mfgcmd 15
```

Return data: Target power level value in dBm

4.6 Set TX power level value

Command to set TX power level value:

```
ot-ctl mfgcmd 16 <Pwr>
```

Where:

Command parameter	Description
`Pwr`	TX power level in dBm Input integer range of -20 to 15

Example of command to set TX power level to 11 dBm:

```
ot-ctl mfgcmd 16 11
```

Return data: Done

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

4.7 Continuous TX

Command to transmit a continuous modulated signal:

```
ot-ctl mfgcmd 17 <Enable>
```

Where:

Command parameter	Description
`Enable`	Enable the continuous transmission 0 = off 1 = on

Return data: Done

Example of command to set continuous TX mode:

```
ot-ctl mfgcmd 17 1
```

4.8 Get TX payload size

Command to get the TX packet payload size (17-116 bytes).

```
ot-ctl mfgcmd 20
```

Return data: 802.15.4 TX packet payload size in bytes

4.9 Set TX packet payload size

Command to set the TX packet payload size in bytes (17-116 bytes):

```
ot-ctl mfgcmd 21 <Size>
```

Where:

Command parameter	Description
`Size`	Packet payload size in bytes Input integer range of 17 - 116

Return data: Done

Example of command to set the TX payload size to 100 bytes:

```
ot-ctl mfgcmd 21 100
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

4.10 802.15.4 TX burst packet

Command to transmit a specific number of packets with a specific packet gap:

```
ot-ctl mfgcmd 33 <count option> <packet gap>
```

Where:

Command parameter	Description
`count option`	The number of packet(s) 0 = 1 (default) 1 = 25 2 = 100 3 = 500 4 = 1000 5 = 2000 6 = 5000 7 = 10000
`Packet gap`	Packet gap in milliseconds Input integer range of 6 – 20 Minimum packet gap for a given payload size 6: For payload size 17 - 44 bytes 7: For payload size 45 - 76 bytes 8: For payload size 77 - 107 bytes 9: For payload size 108 - 116 bytes The maximum gap is 20 ms

Return data: Done

Example of command to transmit 25 packets at 10 ms packet gap:

```
ot-ctl mfgcmd 33 1 10
```


All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

4.11 Set 802.15.4 TX duty cycle

Command to set 802.15.4 transmitter in duty cycle TX mode:

```
ot-ctl mfgcmd 35 <Enable>
```

Where:

Command parameter	Description
`Enable`	Enable TX with duty cycle 0 = disable (default) 1 = enables

Return data: Done

Example of command to enable transmit with duty cycle:

```
ot-ctl mfgcmd 35 1
```

4.12 Start 802.15.4 RX test

Command to start 802.15.4 RX test:

```
ot-ctl mfgcmd 32
```

Return data: Done

4.13 Get 802.15.4 RX test result

Command the extended RX test result report for 802.15.4 radio.

```
ot-ctl mfgcmd 31
```

Return data: Extended RX test result report for 802.15.4.

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

4.14 Command sequence examples to test 802.15.4 radio

This section provides example of commands for 802.15.4 transmit and receive RF performance tests.

4.14.1 Command sequence to test 802.15.4 radio transmit duty cycle mode

Table 16. 802.15.4 radio transmit duty cycle mode command sequence

Step	Operation	Command
1	Enable the RF test mode	`ot-ctl mfgcmd 1`
2	Set the RF channel to 20 (2450 MHz)	`ot-ctl mfgcmd 12 20`
3	Set the power to 4 dBm	`ot-ctl mfgcmd 16 4`
4	Enable TX in duty cycle mode	`ot-ctl mfgcmd 35 1`
5	Stop TX	`ot-ctl mfgcmd 35 0`

4.14.2 Command sequence to test 802.15.4 radio in receive mode

Table 17. 802.15.4 radio receive mode command sequence

Step	Operation	Command
1	Enable the RF test mode	`ot-ctl mfgcmd 1`
2	Set the RF channel to 26 (2480 MHz)	`ot-ctl mfgcmd 12 26`
3	Start RX	`ot-ctl mfgcmd 32`
4	Transmit packets to the DUT with an RF tester	
5	Return the extended RX result report	`ot-ctl mfgcmd 31` Output: `status: 0` `rx_pkt_count : 500` `total_pkt_count : 500` `rssi : -53` `lqi : 118`

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

5 Acronyms and abbreviations

Table 18. Acronyms and abbreviations

Acronyms	Definition
BT	Bluetooth
CS	Carrier suppression
DUT	Device under test
FW	Firmware
RF	Radio frequency
RX	Receive
TX	Transmit

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

6 References

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

[2] UM11483 - Getting Started with NXP-based Wireless Modules on i.MX 8M Quad EVK Running Linux OS (link)

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

7 Note about the source code in the document

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

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

8 Revision history

Table 19. Revision history

Document ID	Release date	Description
AN14114 v.2	15 April 2024	• Section 2.12 "Data rates": added the table with 802.11ac data rate ID.
AN14114 v.1	7 February 2024	• Initial version

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

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.

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.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

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.

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.

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

Tables

Figures

All information provided in this document is subject to legal disclaimers.

© 2024 NXP B.V. All rights reserved.

Contents

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

© 2024 NXP B.V. **All rights reserved.**

For more information, please visit: https://www.nxp.com

Date of release: 15 April 2024
Document identifier: AN14114