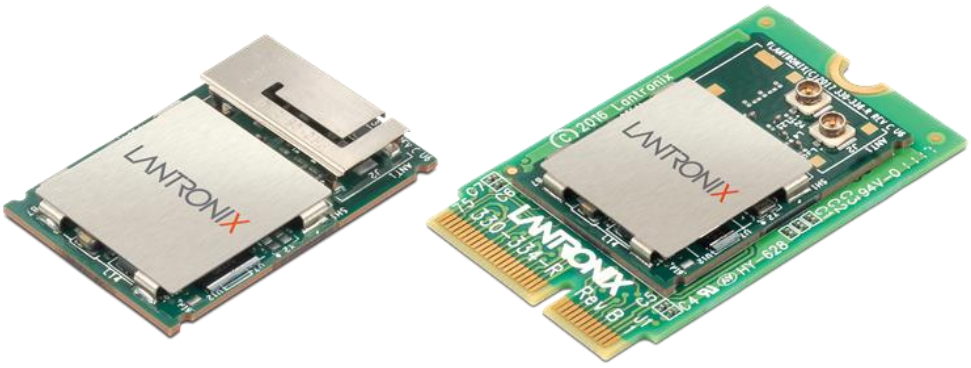
Application Note:

xPico 200® Certification Firmware Instructions



Lantronix, Inc.
7535 Irvine Center Drive
Suite 100
Irvine, CA 92618
Tel: +1 (949) 453-3990

Part Number APP-0001
Revision B August 2018

Intellectual Property

Lantronix and *xPico* are registered trademarks of Lantronix, Inc. in the United States and other countries.

Patented: http://patents.lantronix.com; additional patents pending.

Wi-Fi is a registered trademark of Wi-Fi Alliance Corporation. All other trademarks and trade names are the property of their respective holders.

Contacts

Lantronix, Inc.
7535 Irvine Center Drive, Suite 100
Irvine, CA 92618, USA
Toll Free: 800-526-8766
Phone: 949-453-3990
Fax: 949-453-3995

Technical Support
Online: www.lantronix.com/support

Sales Offices
For a current list of our domestic and international sales offices, go to the Lantronix web site at www.lantronix.com/about/contact

Disclaimer

All information contained herein is provided “AS IS.” Lantronix undertakes no obligation to update the information in this publication. Lantronix does not make, and specifically disclaims, all warranties of any kind (express, implied or otherwise) regarding title, non-infringement, fitness, quality, accuracy, completeness, usefulness, suitability or performance of the information provided herein. Lantronix shall have no liability whatsoever to any user for any damages, losses and causes of action (whether in contract or in tort or otherwise) in connection with the user’s access or usage of any of the information or content contained herein. The information and specifications contained in this document are subject to change without notice.

This equipment has to be tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

1. Reorient or relocate the receiving antenna.
2. Increase the separation between the equipment and receiver.
3. Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
4. Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This device is intended only for OEM Integrators. The OEM integrator should be aware of the following important considerations.

Revision History

Date	Rev.	Comments
April 2018	A	Preliminary Draft.
August 2018	B	Updates for new MFG Loader

For the latest revision of this product document, please check our online documentation at www.lantronix.com/support/documentation.

Table of Contents

Overview

This document provides instructions on leveraging Lantronix® xPico® certificates. In many cases, the Lantronix modular transmitter certification can be leveraged without acquiring full re-certification effort in products that use the Lantronix module.

The xPico 200 series has modular approval for FCC, IC, EU, Japan, China and Australia/New Zealand. All xPico certificates can be found in the xPico 200 module documentation at https://www.lantronix.com/products/xpico-200/#docs-downloads/.

It is recommended that you consult with your certification laboratory to develop your certification plan for your product that includes the xPico 200 series module. The xPico 200 series module certification tests were completed at Bureau Veritas in Hsinchu, Taiwan.

Requirements for Leveraging Lantronix xPico 200 Certificates

The following conditions are required to leverage Lantronix modular transmitter certifications:

- Following the antenna and layout instructions in the *xPico 200 Series Integration Guide*.
- Using antennas of similar type and equal or less gain than the antennas listed in the *xPico 200 Series Integration Guide*.
- Positioning the xPico 200 module at least 20 cm from a human body and the transmitting antennas at least 20 cm from all other transmitters. Lantronix has not completed SAR testing on the xPico 200 module.
- Running EMC tests including the FCC 15-part B and EN 301 489 -1/-17. When leveraging the modular certification, the transmitter and receiver-specific tests normally do not need to be performed.
- Placing certifications for the xPico 200 transmitter IDs for various regions on the end-product label according to conditions listed in the *Compliance* section of the *xPico 200 Series Data Sheet.* See https://www.lantronix.com/products/xpico-200/#docs-downloads.
- Running two certification testing modes. See *Certification Test Modes*.

Disqualifications to Leveraging Lantronix xPico 200 Certificates

The following conditions disqualify the leveraging of Lantronix xPico 200 certifications:

- Using a different type of antenna than that shown in the *xPico 200 Series Integration Guide*.
- Using an antenna with higher gain than the antennas called out in the *xPico 200 Series Integration Guide*.
- Installing the xPico 200 module antenna in a location where it is expected to be less than 20 cm from a human body. Under this scenario, SAR testing would need to be completed.
- Installing the xPico 200 module antenna within 20 cm of another transmitter module.
- Installing the xPico 200 module in a country or region not referenced in the *Compliance* section of the *xPico 200 Series Data Sheet.* See https://www.lantronix.com/products/xpico-200/#docs-downloads.

Certification Test Modes

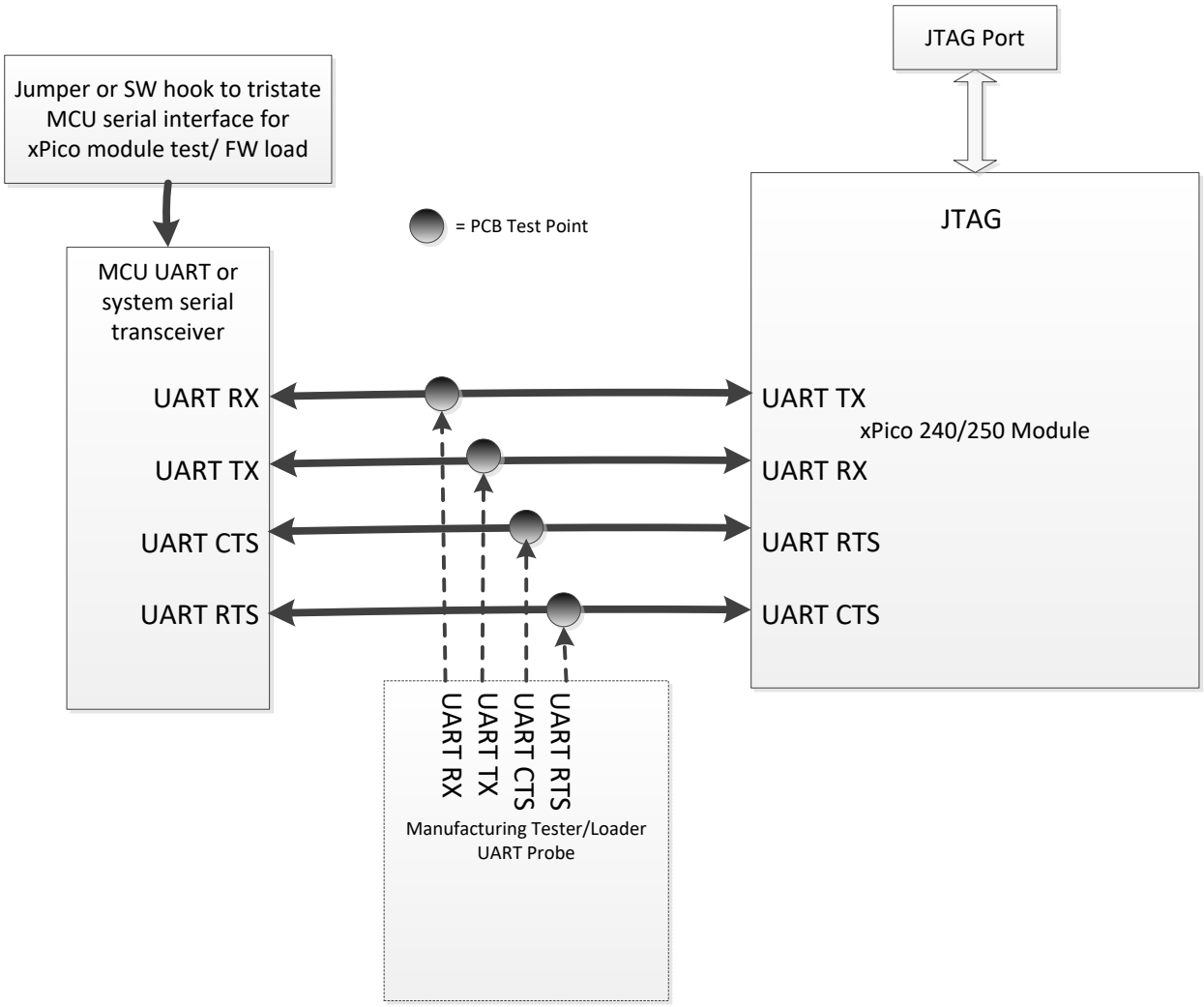
Certification testing requires both normal mode testing and continuous mode testing. The number of units needed for certification testing depends on the regional certifications planned. Consult with your certification lab prior to testing for their recommended quantity of normal mode and continuous mode test units.

- **Normal mode testing** is conducted with the xPico 200 module running the base application software. In this case the unit should be configured to run tests with the module interfaces both idle and fully exercised. For Ethernet and Wi-Fi interfaces, iperf or some other network utilization method can be used. Serial and USB ports should utilize a similar program to exercise the used ports for certification EMC tests.
- **Continuous mode testing** is conducted in cases where full transmitter certification or re-certification is required. The certification lab will require testing on some units that run continuous mode transmitter and receiver tests. To run the continuous transmit and receive mode tests, special firmware needs to be loaded on the xPico 200 module.

Hardware Requirements for Continuous Mode Testing

The following hardware requirements are needed for continuous mode testing. The continuous mode software can be downloaded at https://www.lantronix.com/products/xpico-200/#docs-downloads/.

- To run the xPico 240/250 transmitter and receiver tests for full certification, access to the module serial port is required.
- The serial port connections are required to load a special firmware image that allows continuous transmit, continuous receive, and other tests required for transmitter certification.
- Lantronix recommends adding test points for the serial port to assist with loading the certification test code. The current implementation requires that this data is loaded via the serial interface. It's recommended to include an option to tristate other devices connected to the serial port line while the manufacturing test loading is in progress. An example of the recommended manufacturing test point is shown in the figure below.



See the evaluation board schematic and artwork for recommended transceiver, DB9, and port connections in the *xPico 200 Series Evaluation Kit User Guide*.

Continuous Mode Software Load

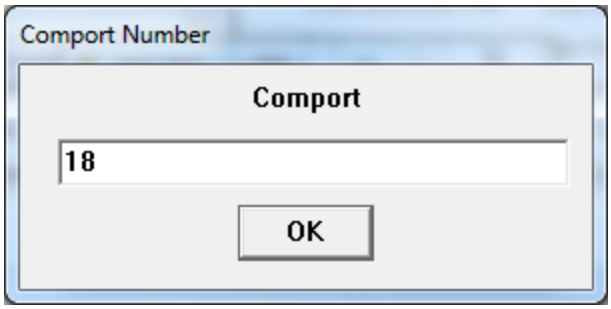
To use the module for continuous mode testing for certification, download and load the Wi-Fi continuous mode software.

1. Download the xPico 200 continuous mode software from https://www.lantronix.com/products/xpico-200/#docs-downloads/.
2. Copy the `xPico200_MFG_Test` directory to the C:\ drive of your PC.

   Note: If you copy it to another directory on your PC, you may need to modify the load scripts to point to a different directory.
3. Install Tera Term 4.73 on your PC.

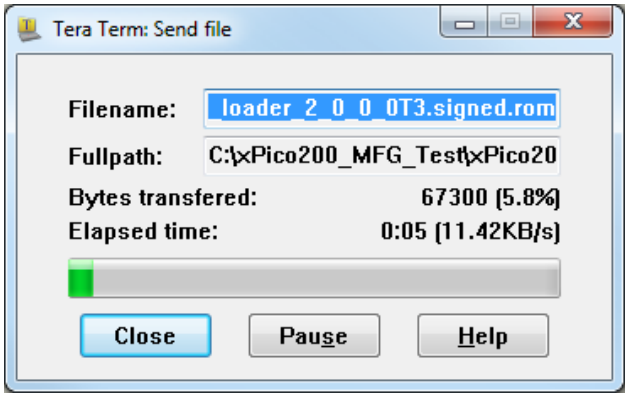
   Note: The ***xP200-cmd_teraterm – wifi.bat*** file contains a link to the ttermpro.exe executable. You may need to modify this link if the executable is in a different directory on your PC.

4. Connect the serial port of the module to your PC.
5. Double click the ***xP200-cmd_teraterm – wifi.bat*** file in the xPico200_MFG_Test directory.
6. In the Comport Number window that opens, enter the COM port that's connected to the xPico 200 module and click **OK**.

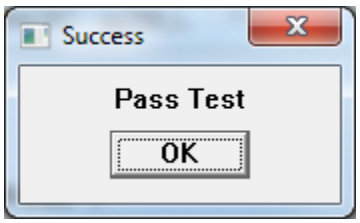


7. Click **OK** to start.

8. **Assert the module default pin and power up the module**. The Tera Term window will display the firmware transfer progress.



9. When the firmware has successfully loaded, a Pass Test window will be displayed.



Note: In some cases, you may need to reset the Tera Term window after loading the firmware.

Running the Wi-Fi Continuous Mode Tests

Wi-Fi continuous mode uses the Cypress WL commands. For more information on WL tool commands please see the Cypress reference document at http://www.cypress.com/file/385966/download.

The Wi-Fi scripts directory includes several scripts that can be pasted into the Tera Term command prompt.

Example scripts are shown below.

802.11g Transmit Script.

```
wl down
wl country ALL
wl band b
wl chanspec -c 1 -b 2 -w 20 -s 0
wl mpc 0
wl ampdu 1
wl bi 65000
wl frameburst 1
wl rateset 54b
wl up
wl txant 0
wl antdiv 0
wl nrate -r 54
wl phy_watchdog 0
wl disassoc
wl phy_forcecal 1
wl phy_activecal
wl txpwr1 -1
wl pkteng_start 00:90:4c:aa:bb:cc tx 20 1024 0
```

In the ***wl chanspec -c 1 -b 2 -w 20 -s 0*** line, ***-c 1*** sets the channel to channel 1. Change the ***1*** parameter to any integer from 1 to 14 to specify the channel.

Use the ***wl nrate -r 54*** line to modify the bit rate. The **-r 54** sets the rate to 54Mbps.

wl nrate -m 0 -s 0 is for MCS0.

wl nrate -m 7 -s 0 is for MCS7.

The ***txant*** parameter sets the antenna to either antenna 0 or antenna 1

802.11a Transmit Script

```
wl down
wl country ALL
wl band a
wl chanspec -c 100 -b 5 -w 20 -s 0
wl mpc 0
wl ampdu 1
wl bi 65000
wl frameburst 1
wl rateset 54b
wl up
wl txant 0
wl antdiv 0
wl nrate -r 54
wl phy_watchdog 0
wl disassoc
wl phy_forcecal 1
wl phy_activecal
wl txpwr1 -1
wl pkteng_start 00:90:4c:aa:bb:cc tx 20 1024 0
```

Similar to the 802.11g script above, change the ***chanspec*** line to set the channel using the parameter after the **-c** parameter and change the ***nrate*** line to set the bit rate.

The ***txant*** parameter sets the antenna to either antenna 0 or antenna 1.

802.11n Transmit Script

```
wl down
wl mpc 0
wl phy_txpwrctrl 1
wl phy_watchdog 0
wl country ALL
wl PM 0
wl band a
wl 5g_rate -v 7 -s 1 -b 40
wl chanspec -c 38 -b 5 -w 40 -s -1
wl up
wl antdiv 0
wl txant 0
wl txpwr1 -1
wl phy_forcecal 1
wl phy_activecal
wl scansuppress 1
wl pkteng_start 00:90:4c:aa:bb:cc tx 100 2048 0
```

Similar to the 802.11g script above change the ***chanspec*** line to set the channel using the parameter after the ***-c*** parameter and change the ***5g_rate*** line to set the bit rate.

The ***-b*** parameter in the ***5g_rate*** line and the ***-w*** parameter in the ***chanspec*** line sets the bandwidth. The xPico 200 supports 20Mhz and 40Mhz bandwidth for 802.11n in the 5Ghz band.

The ***txant*** parameter sets the antenna to either antenna 0 or antenna 1.

Transmit Stop Command

```
wl pkteng_stop tx
```

The ***wl pkteng_stop tx*** command stops the transmit test.

802.11 2Ghz Receive Mode Script

```
wl down
wl mpc0
wl country ALL
wl band b
wl scansuppress 1
wl channel 1
wl bi 65535
wl up
wl out
wl channel
wl rateset 11b
wl up
wl txant 0
wl antdiv 0
wl nrate -r 1
wl counters
wl counters
```

The ***nrate*** and ***channel*** parameters above work like the parameters from the above transmit scripts.

The ***wl counters*** command will output receive mode statistics. The ***rxdfrmocast*** statistic lists the received frames, which is useful when connected to an external signal generator. Run the ***wl counters*** command before and after sending external frames to the unit and calculate the difference between the before and after ***rxdfrmocast*** number to get the number of received good frames.

802.11 5Ghz Receive Mode Script

```
wl down
wl mpc 0
wl country ALL
wl band a
wl scansuppress 1
wl chanspec -c 159 -b 5 -w 40 -s 0
wl bi 65535
wl up
wl out
wl channel
wl up
wl txant 0
wl antdiv 0
wl nrate -m 7 -s 0
wl counters
```

The ***nrate***, ***channel***, and ***chanspec*** parameters above work like the parameters from the above transmit scripts.

The ***wl counters*** command will output receive mode statistics. The ***rxdfrmocast*** statistic lists the received frames, which is useful when connected to an external signal generator. Run the ***wl counters*** command before and after sending external frames to the unit and calculate the difference between the before and after ***rxdfrmocast*** number to get the number of received good frames.

Bluetooth Continuous Mode Testing

The steps in http://www.cypress.com/file/298091/download describe how to run the Lantronix xPico 200 Bluetooth continuous mode scripts. These scripts are based on the Cypress `mybluetool` commands for the Cypress processor inside of the xPico 200 module.

Running the BT Test Modes

To run the BT Test Modes, follow these steps:

1. Follow the instructions for each type of test in the Cypress document at http://www.cypress.com/file/298091/download
2. Load the continuous mode firmware as described in the Continuous Mode Software Load section.
3. Once the firmware is loaded, run the ***bt_hci_reset*** command twice. The second time this command is run, it should respond with SUCCESS.
4. After the reset command run the ***bt_download_firwmare*** command.
5. Table 1 lists the remaining BT commands and syntax. The equivalent commands from the Cypress document are included.

   Note: There are slight differences between the Cypress document command and the Lantronix command.

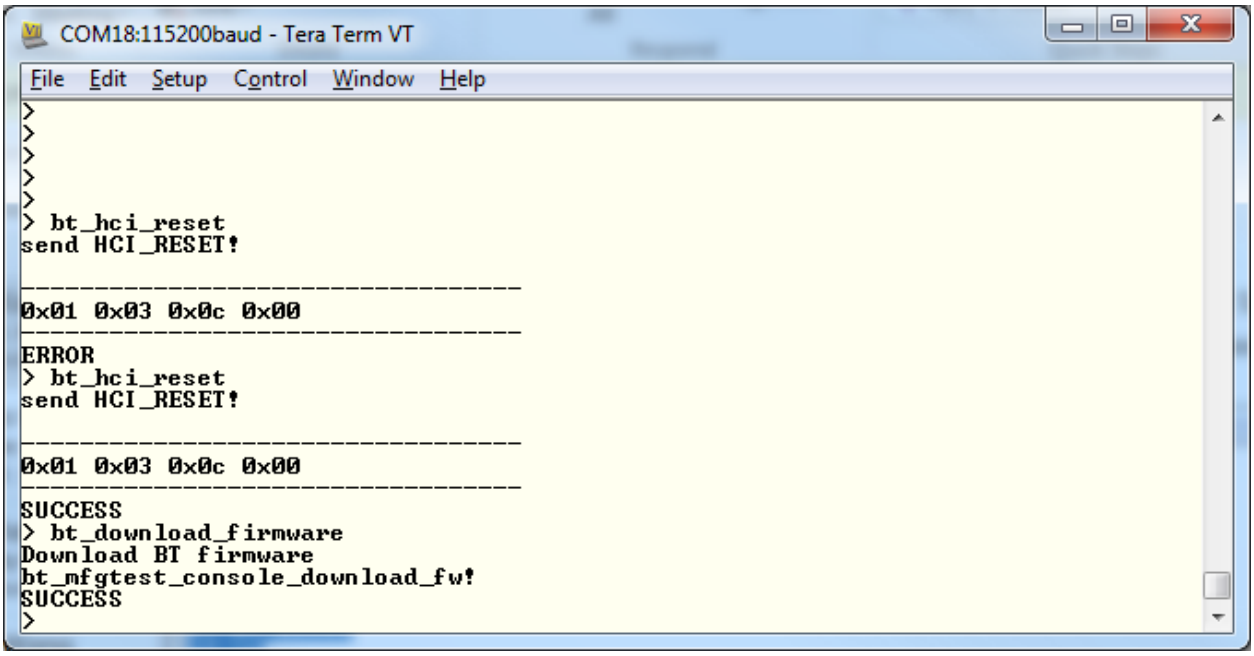


Table 1: BT Test Commands

BT Test Command	Description

`bt_hci_reset`
 	
`Cypress equivalent command`
`mbt reset`
	This command resets the BT device on the module.
	Run this prior to running any other certification test.
	This can also be used to stop any of the running BT tests.

`bt_download_firmware`	Run this command prior to any of the tests below to load the BT test firmware into the radio.

`bt_le_rx_test <rx_channel>`
 	
`Cypress equivalent command`
`mbt le_receiver_test`
	This command runs the LE receiver test.
	rx_channel = receive frequency minus 2402 divided by 2.
	Acceptable channels are 0 to 39 where 0 is 2402Mhz and 39 is 2480 Mhz.

`bt_le_tx_test <tx_channel> <data_length> <data_pattern>`
 	
`Cypress equivalent command`
`mbt le_transmitter_test`
	This command runs the LE transmit test.
	tx_channel = receive frequency minus 2402 divided by 2.
	Acceptable channels are 0 to 39 where 0 is 2402 Mhz and 39 is 2480 Mhz.
	Data_length can be from 0 to 37.
	Data_pattern can be from 0 to 7.
	
	Data Pattern Definitions
	0: Pseudo-random bit sequence 9
	1: Pattern of alternating bits: 11110000
	2: Pattern of alternating bits: 10101010
	3: Pseudo-random bit sequence 15
	4: Pattern of all 1s
	5: Pattern of all 0s
	6: Pattern of alternating bits: 00001111
	7: Pattern of alternating bits: 0101

BT Test Command	Description

`bt_le_test_end`
 	
`Cypress equivalent command`
`mbt le_test_end`
	The LE test end script stops the LE transmitter or LE receiver tests.

`bt_set_tx_frequency_arm <carrier On/Off> <tx_frequency> <tx_mode> <tx_modulation_type> <tx_power>`
 	
`Cypress equivalent command`
`mbt tx_frequency_arm`
	This test turns on or off the transmitter carrier.
	carrier on/off:
	1: carrier on
	0: carrier off
	
	tx_frequency can be from 2402 MHz to 2480 MHz
	tx_mode selects unmodulated or modulated with pattern
	0: Unmodulated
	1: PRBS9
	2: PRBS15
	3: All Zeros
	4: All Ones
	5: Incrementing Symbols
	
	tx_modulation_type selects 1 Mbps, 2 Mbps, or 3 Mbps modulation. This is ignored if mode is unmodulated.
	0: GFSK
	1: QPSK
	2: 8PSK
	3: LE
	
	tx_power can be from –25 dBm to +3 dBm

`bt_receive_only <rx_frequency>`
 	
`Cypress equivalent command`
`mbt receive_only`
	This test instructs the BT radio to receive on a specific frequency.
	Rx_frequency can be from 2402 to 2480.

BT Test Command	Description

`bt_radio_tx_test <bd_addr> <frequency> <modulation_type> <logical_channel> <bb_packet_type> <packet_length> <tx_power>`
 	
`Cypress equivalent command`
`mbt radio_tx_test`
	This command runs the connectionless transmitter test.
	
	bd_addr is the BD_ADDR of Tx device (6 bytes), for example 00112233445566.
	
	Frequency can be set to 0 to use a normal Bluetooth hopping sequence or from 2402 MHz to 2480 MHz to transmit on a specified frequency without hopping.
	
	modulation_type sets the data pattern.
	
	0: 0x00 8-bit Pattern
	
	1: 0xFF 8-bit Pattern
	
	2: 0xAA 8-bit Pattern
	
	3: 0xF0 8-bit Pattern
	
	4: PRBS9 Pattern
	
	logical_channel sets logical channel to Basic Rate (BR) or Enhanced Data Rate (EDR) for ACL packets.
	
	0: EDR
	
	1: BR
	
	bb_packet_type sets the baseband packet type to use.
	
	3: DM1
	
	4: DH1/2-DH1
	
	8: 3-DH1
	
	10: DM3/2-DH3
	
	11: DH3/3-DH3
	
	14: DM5/2-DH5
	
	15: DH5/3-DH5
	
	packet_length can be from 0 to 65535. The device will limit the maximum packet length based on the baseband packet type. For example, if DM1 packets are sent, the maximum packet size is 17 bytes.
	
	tx_power can be from –25 dBm to +3 dBm.

BT Test Command	Description

`bt_radio_rx_test <bd_addr> <frequency> <modulation_type> <logical_channel> <bb_packet_type> < packet_length> <test duration>`
 	
`Cypress equivalent command`
`mbt radio_rx_test`
	This test sets the BT radio to receive on a specific frequency and sends reports about received packets.
	
	bd_addr is the BD_ADDR for the remote Tx device (6 bytes).
	
	Frequency sets the frequency used to listen from 2402 MHz to 2480 MHz.
	
	modulation_type sets the data pattern to compare received data.
	
	0: 0x00 8-bit pattern
	
	1: 0xFF 8-bit pattern
	
	2: 0xAA 8-bit pattern
	
	3: 0xF0 8-bit pattern
	
	4: PRBS9 pattern
	
	logical_channel sets the logical channel to BR or EDR for ACL packets.
	
	0: EDR
	
	1: BR
	
	bb_packet_type sets the packet type of the expected packets.
	
	3: DM1
	
	4: DH1/2-DH1
	
	8: 3-DH1
	
	10: DM3/ 2-DH3
	
	11: DH3/3-DH3
	
	14: DM5/2-DH5
	
	15: DH5/3-DH5
	
	packet_length can be from 0 to 65535. The device compares the length of the received packets with the specified packet_length.
	
	Test duration is the time in seconds from 1 to 200 to run the test.

BT Test Command	Description

`bt_le_enhanced_tx_test <tx_channel> <data_length> <data_pattern> <PHY>`
 	
`Cypress equivalent command`
`mbt le_enhanced_transmitter_test`
	This command runs the enhanced LE transmit test.
	tx_channel = receive frequency minus 2402 divided by 2.
	Acceptable channels are 0 to 39 where 0 is 2402 Mhz and 39 is 2480 Mhz.
	Data_length acceptable values are 0 to 37.
	Data_pattern acceptable values are 0 to 7.
	
	Data Pattern Definitions
	0: Pseudo-random bit sequence 9
	1: Pattern of alternating bits: 11110000
	2: Pattern of alternating bits: 10101010
	3: Pseudo-random bit sequence 15
	4: Pattern of all 1s
	5: Pattern of all 0s
	6: Pattern of alternating bits: 00001111
	7: Pattern of alternating bits: 0101
	
	PHY: (1 - 2)
	1: Transmitter set to transmit data at 1Ms/s.
	2: Transmitter set to transmit data at 2Ms/s.

`bt_le_enhanced_rx_test <rx_channel> <PHY> <modulation_index>`
 	
`Cypress equivalent command`
`mbt le_enhanced_receiver_test`
	This command runs the enhanced LE receiver test.
	rx_channel = receive frequency minus 2402 divided by 2.
	Acceptable channels are 0 to 39 where 0 is 2402Mhz and 39 is 2480 Mhz.
	
	PHY: (1 - 2)
	1: Transmitter set to transmit data at 1Ms/s
	2: Transmitter set to transmit data at 2Ms/s
	
	modulation_index: (1 - 2)
	1: Assume transmitter will have a standard modulation index.
	2: Assume transmitter will have a stable modulation index.

BT Test Command	Description

`bt_connectionless_dut_loopback_mode COMx`
 	
`Cypress equivalent command`
`mbt connectionless_dut_loopback_mode`
	This command sets up a connectionless loopback test with an external tester for analyzing both RX and TX.
	
	Once the connectionless command is entered, the unit will prompt for the interactive arguments below.
	
	Remote_Device_BD_ADDR: This is the BD_ADDR of the remote transmitting device. [Size: 6 bytes]
	
	LT_ADDR: This is the logical transport address of the BT link. [Size: 1 byte] [Range: 0x01 - 0x07]
	
	Number_Of_Tests: This is the number of tests to be executed. [Size: 1 byte] [Range: 0x01 - 0x10]
	
	The following arguments repeat depending on the number of tests:
	
	Retry Offset: When a timeout occurs, subtract the offset to go back to the earlier test.
	
	[Size: 6 bits {31:26} in little endian uint32]
	[Range: 0x01 - 0x3f].
	
	Number_Packets: This is the number of packets to be received for this test.
	
	[Size: 15 bits {25:11} in little endian uint32]
	[Range: 0x01 - 0x7fff].
	
	TxPowerIndex: This is the power table index to use.
	
	[Size: 3 bits {10:8} in little endian uint32]
	[Range: 0x00 - 0x07].
	
	RxChannel: This is the frequency offset in MHz from 2402 MHz.
	
	[Size: 7bits {7:1} in little endian uint32]
	[Range: 0x00 - 0x7f].
	
	Packet Table Type: This defines the type of the packet.
	
	[Size: 1bit {0:0} in little endian uint32].
	
	0x0: Basic Rate Packet Types
	
	0x1: EDR packet types
	
	Retry Time Out: This is the time required to retry.
	
	[Size: 1 byte] [Range: 0x01 - 0xff].
	
	Test Scenarios:
	
	[Size: 1 byte].
	
	0x0: RX-TX Loop Back Mode.
	
	0x1: RX only with BER stats.

Full List of Loader Commands.

In order to see a list of commands, enter “?” at the command prompt to view the command list. For reference, the list of commands is shown below.

?

Console Commands:

?

help [<command> [<example_num>]]

- Prints help message or command example.

loop <times> <semicolon_separated_commands_list>

- Loops the specified commands for specified number of times.

$?

- Prints the return value of the last executed command.

antenna <0|1|3>

- Selects the antenna. 3 = Auto

get_ap_info

- Gets the AP information.

get_access_category_parameters_sta

- Gets access category parameters for STA.

get_associated_sta_list

- Gets the list of associated clients.

get_btc_params [SSID]

- Finds AP.

get_counters [-t seconds][-n][-v][rx][tx][rate]

- Gets counters. Options:

  -t num_secs: duration to collect counters

  -n: normalizes counters to a per second basis

  -v: verbose (only nonzero counters are printed by default)

  Category: rx | tx | rate (default: print all categories)

get_country

- Gets country.

set_country <US/0|KR/4>

- Sets country.

get_rate

- Gets current rate.

get_data_rate

- Gets data rate.

get_mac_addr

- Gets the device MAC address.

get_preferred_association_band

- Gets the preferred radio band for association.

get_pmk <key>

- Gets PMK.

get_random

- Gets a random number.

get_rssi

- Gets the received signal strength of the AP (client mode only).

get_soft_ap_credentials

- Gets SoftAP credentials.

get_tx_power

- Gets the tx power in dBm.

join <ssid> <open|wpa_aes|wpa_tkip|wpa2|wpa2_tkip|wpa2_fbt> [key] [channel] [ip netmask gateway]

When any parameter has spaces, use quotes, e.g. "my ssid," "my wpa2 key."

- Joins an AP. DHCP is assumed if an IP address isn't specified.

leave

- Leaves an AP.

scan [scan_type] [0 = Infra | 1 = Adhoc | 2 = Any] [optional_ssid] [optional_mac] [(optional_channel_list)] [(optional_extended_params)]

scan_type flag: 0 = active, 1 = passive, 2 = pno, 4 = prohibit, 8 = no bssid filter

Example: scan 0 - - - (1,2,3,4) "-" denotes default

- Scans all enabled channels and prints a list of APs found.

set_preferred_association_band <0 = Auto | 1 = 5GHz | 2 = 2.4GHz>

- Sets the preferred radio band for association.

set_tx_power <0-31>

- Sets the tx power in dBm.

status

- Prints the status of the Wi-Fi and network interfaces.

wifi_powersave <mode> [delay]

- Enables/disables Wi-Fi powersave mode. 0 = disable. 1 = PS Poll. 2 = Wait [delay] ms before entering powersave

wifi_resume [<ip> <netmask> <gateway>]

- Resumes networking after deep-sleep.

wlan_chip_log

- Dumps WLAN chip console log.

wlog

- Dumps WLAN chip console log.

wwd_stats [clear]

- Dumps WWD stats. clear=1. This will reset stats after printing.

find_ap [SSID]

- Finds AP.

country [US|KR|JP]

- Gets/sets ccode.

reset_wifi_counters <set> [value]

- Resets persistent Wi-Fi statistics counters.

wlver

- Gets WLAN Firmware Version.

clmver

- Gets WLAN CLM Version.

memuse

- Gets WLAN memory usage.

wlog_stream [start | stop]

- Starts/stops continuous WLAN logging.

dump <ampdu>|<amsdu>|<all>

- Gets dump from the FW.

Ex: dump ampdu, FW binary should have the ampdu dump feature enabled.

ampdu_clear_dump

- Clears ampdu dump.

get_curr_band

- Gets the current radio band.

get_bw

- Gets the bandwidth.

get_channel

- Gets the channel number.

get_pm_mode

- Gets the WLAN PM mode.

wilog

- Prints wlan ioctls, iovars, and events.

disable_11n

- Disables 11n mode operation.

ds1_config <ulp_wait> [type [value1] [value2] [value3]]

- Configures deep sleep (ds1).

Examples: ds1_config <ulp wait: ex. 5000> gtk

ds1_config <ulp wait: ex. 5000> magic keep_alive <period msecs: ex. 20> <packet data: ex. 0x3243567abcdef> pattern <offset in packet: ex. 2> <mask: ex. ff> <pattern: ex. ff> arp_hostip <v4 address: ex. 192.168.1.115>

ds1_config 5000 all

ds1_config 5000 deauth

ds1_enable

- Enables Wi-Fi Deep Sleep; enter sleep when able.

ds1_enter <magic|keep_alive|pattern|arp_hostip> <ulp_wait> [value1] [value2] [value3] [-wowl=1] [-wowl_os=3] [-read_back]

- Enters deep sleep (ds1) with given offload.

Examples: ds1_enter keep_alive <ulp wait: ex. 5500> <period msecs: ex. 20> <packet data: ex. 0x3243567abcdef>

ds1_enter pattern <ulp wait: ex. 8> <offset in packet: ex. 20> <mask: ex. 0xffe008> <pattern: ex. 0x34567890>

ds1_enter arp_hostip <ulp wait: ex. 8> <v4 address: ex. 192.168.1.115>

Debug options:

-read_back: reads back and prints firmware values after set.

-wowl=0x2 or -wowl_os=0x2: force wowl or wowl_os value in firmware to be 2 in process of going to DS.

ds1_wake

- Wakes from DS1 and goes to Powersave mode.

ds1_disable

- Disables DS1.

ds1_status

- Current DS1 status.

enable_11n

- Enables 11n mode operation.

fbtoverds [value]

- Use of FBT (Fast BSS Transition) Over-the-DS (Distribution System) is allowed.

fbt_cap

- Driver four-way handshake and reassoc (WLFBT).

get_noise

- Gets PHY noise after successful TX.

join_adhoc <ssid> <open|wpa_aes|wpa_tkip|wpa2|wpa2_tkip> [key] [ip netmask gateway]

When any parameter has spaces, use quotes, e.g. "my ssid," "my wpa2 key."

- Joins the specified IBSS. No DHCP assumed.

join_specific <ssid> <bssid> <channel> <open|wpa_aes|wpa_tkip|wpa2|wpa2_tkip> [key] [ip netmask gateway]

When any parameter has spaces, use quotes, e.g. "my ssid," "my wpa2 key."

- Joins the specified AP. DHCP is assumed if an IP address isn't specified.

phyrate_dump <bin size>

- Dumps the phyrate log and bin averages to the console.

scan_disable <1 = disable scan|0 = enable scan>

- Disables scanning in FW; aborts any active scan.

roam_delta <2g|5g|all> [value]

- Sets or gets roam delta value. roam_delta 2g 5

roam_trigger <2g|5g|all> [value]

- Sets or gets roam trigger value. roam_trigger all -27

set_data_rate <1|2|5.5|6|9|11|12|18|24|36|48|54>

- Sets the data rate.

set_legacy_rate <1|2|5.5|6|9|11|12|18|24|36|48|54>

- Sets the legacy (CCK/OFDM) rate on PHY.

set_mcs_rate <MCS Index> <Override MCS only>

- Sets the MCS rate on PHY.

set_listen_interval <listen interval> <time unit>

- Sets the listen interval in time unit. 0 = Beacon Intervals. 1 = DTIM Intervals.

start_ap <ssid> <open|wpa2|wpa2_aes|wep|wep_shared> <key> <channel> <wps> [ip] [netmask]

When any parameter has spaces, use quotes.

E.g. start_ap "my ssid" wpa2 "my wpa2 key " 11 no_wps 192.168.2.1 255.255.255.0. Default settings for IP and subnet mask are 192.168.0.1 and 255.255.255.0 or the last IP and subnet specified through this command if applicable.

- Starts AP mode.

stop_ap

- Stops AP mode.

test_ap <ssid> <open|wpa2|wpa2_aes> <key> <channel> <wps> <iterations>

When any parameter has spaces, use quotes.

E.g. test_ap "my ssid" wpa2 "my wpa2 key " 11

- Test AP mode.

test_join <ssid> <open|wep|wpa_aes|wpa_tkip|wpa2|wpa2_tkip> [key] [ip netmask gateway] <iterations>

When any parameter has spaces, use quotes, e.g. "my ssid," "my wpa2 key."

- Tests joining an AP. DHCP is assumed if an IP address isn't specified.

test_join_specific <ssid> <bssid> <channel> <open|wep|wpa_aes|wpa_tkip|wpa2|wpa2_tkip> [key] [ip netmask gateway] <iterations>

When any parameter has spaces, use quotes, e.g. "my ssid," "my wpa2 key."

- Tests joining an AP. DHCP is assumed if an IP address isn't specified.

test_cred <ssid> <bssid> <channel> <open|wep|wpa_aes|wpa_tkip|wpa2|wpa2_tkip> [key]

When any parameter has spaces, use quotes, e.g. "my ssid," "my wpa2 key."

- Tests joining an AP.

peek [address]

- Dumps memory.

poke [address] [value]

- Writes memory.

peek_wifi [address] <# repeat>

- Dumps Wi-Fi memory for one or the given number of contiguous memory locations.

poke_wifi [address] [value]

- Writes Wi-Fi memory.

rrm <set|get> [value]

- Enables or disables an RRM report.

rrm set [value]: [value] is the bit mask to set the corresponding report

rrm get: shows the currently set RRM reports

E.g. rrm set +Channel_load_Measurement

rrm set 0x100

rrm get

rrm_nbr_req [SSID]

- Sends 11k neighbor report measurement request (works only when associated).

rrm_lm_req [da]

- Sends 11k link measurement request.

rrm_bcn_req [bcn mode] [da] [duration] [random int] [channel] [ssid] [repetitions]

- Sends 11k beacon measurement request.

rrm_chload_req [regulatory] [da] [duration] [random int] [channel] [repetitions]

- Sends 11k channel load measurement request.

rrm_noise_req [regulatory] [da] [duration] [random int] [channel] [repetitions]

- Sends 11k noise measurement request.

rrm_frame_req [regulatory] [da] [duration] [random int] [channel] [ta] [repetitions]

- Sends 11k frame measurement request.

rrm_stat_req [da] [random int] [duration] [peer] [group id] [repetitions]

- Sends 11k stat measurement request.

rrm_nbr_list

- Gets 11k neighbor report list.

rrm_nbr_del_nbr [bssid]

- Deletes node from 11k neighbor report list.

rrm_nbr_add_nbr [bssid] [bssid info] [regulatory] [channel] [phytype]

- Adds node to 11k neighbor report list.

rrm_bcn_req_thrtl_win <set|get> [value]

- Sets or gets the beacon throttle window (milliseconds) in which off-channel time is computed.

rrm_bcn_req_max_off_chan_time <set|get> [value]

- Sets or gets the maximum scan time allowed in beacon throttle window (milliseconds).

rrm_bcn_req_traff_meas_per <set|get> [value]

- Sets or gets the milliseconds period to check traffic.

wnm <set|get> [value] Example: [value] can be bitwise OR of below values

WL_WNM_BSSTRANS 0x00000001

WL_WNM_PROXYARP 0x00000002

WL_WNM_MAXIDLE 0x00000004

WL_WNM_TIMBC 0x00000008

WL_WNM_TFS 0x00000010

WL_WNM_SLEEP 0x00000020

WL_WNM_DMS 0x00000040

WL_WNM_FMS 0x00000080

WL_WNM_NOTIF 0x00000100

WL_WNM_WBTEXT 0x00000200

- Sets or gets the WNM capabilities.

wnm_bsstrans_query optional [SSID]

- Sends the BSS transition query with SSID or NULL (works only when associated).

wnm_bsstrans_resp <set|get> [value] Example: [value] can be one of the following

WL_BSSTRANS_POLICY_ROAM_ALWAYS = 0

WL_BSSTRANS_POLICY_ROAM_IF_MODE = 1

WL_BSSTRANS_POLICY_ROAM_IF_PREF = 2

WL_BSSTRANS_POLICY_WAIT = 3

WL_BSSTRANS_POLICY_PRODUCT = 4

- Sends the BSS transition response policy.

mfp_cap <set|get> [value]

- Sets or gets the PMF (Protected Management Frame).

wlf <command> [value]

- Runs a wl command; must specify formatting, e.g. wlf -i <command> <int>

wlf -ui <command> <uint>

wlf -s <command> <string>

Force a set: wlf -set <command>

Specify len of get: wlf -length=200 <command>

nan <command> [value]

- Runs nan command followed by attribute name and followed by set, e.g. nan <nan_cluster_id> <set> if get then nan <nan_cluster_id>

wlver

- Gets the WLAN Firmware Version.

clmver

- Gets the WLAN CLM Version.

mesh_set_channel [channel]

- Sets the channel.

mesh_auto_peer [1=enable, 0=disable]

- Enables or disables the mesh auto peer channel.

mesh_filter [mesh peer mac address]

- Filters the MAC address of mesh peer.

join_mesh [ssid]

- Joins the mesh network by SSID.

mesh_mcast_rebroadcast [1=enable, 0=disable]

- Enables or disables the mesh mcast rebroadcast.

ping <destination> [-i <interval in ms>] [-n <number>] [-l <length>]

- Pings the specified IP or host.

reboot

- Reboots the device.

get_time

- Prints the current time.

sleep

- Sleeps number of milliseconds.

prng_bit_dump <num>

- Number of bytes to print using 1s and 0s.

prng <num>

- Bytes number

prng_set_algorithm <string>

- Sets the name of the algorithm from prng_get_algorithms.

prng_get_algorithms

- Gets the algorithm.

mcu_powersave <0|1>

- Enables or disables the MCU powersave.

wiced_init <0|1>

- Calls wiced_deinit/wiced_init.

loglevel_set [facility] <loglevel>

- Sets a new wiced logging level.

loglevel_get

- Gets the current wiced logging levels for all facilities.

mcu_powersave_mode <0|1>

- <MCU powersave mode: 0 - deep-sleep, 1 - normal sleep>

hibernation <sleep_ms>

- Forces the chip to hibernate for the specified number of milliseconds.

mcu_powersave_clock <0|1> <0|1|2|3|4>

- <Clock request or release: 0 - release, 1 - request> <Which clock: 0 - ALP available, 1 - HT available, 2 - have at least ILP on backplane, 3 - at least ALP, 4 - at least HT>

mcu_powersave_tick <0|1|2>

- <RTOS tick mode: 0 - always tickless, 1 - never tickless, 2 - tickless if MCU power-save enabled>

mcu_powersave_freq <freq_mode>

- <CPU/backplane frequency mode>

mcu_powersave_sleep <0|1> <sleep_ms>

- 0 - RTOS sleeping where the CPU can wake up earlier if requested by other threads though the current one remains in sleep state.
- 1 - forced sleeping where the platform is forced to ignore all interrupts except the timer and sleeps the specified amount of time.

mcu_powersave_info

- Prints the powersave information.

mcu_wlan_powersave_stats

- Prints the WLAN powersave statistics.

mcu_powersave_gpio_wakeup_enable <input_pin_pull_mode> <trigger>

- Enables waking from deep-sleep via GPIO.

mcu_powersave_gpio_wakeup_disable

- Disables waking from deep-sleep via GPIO.

mcu_powersave_gpio_wakeup_ack

- If GPIO generates a wake up event, it remains triggered till acked.

mcu_powersave_gci_gpio_wakeup_enable <pin> <input_pin_pull_mode> <trigger>

- Enables waking up from deep-sleep via GPIO.

mcu_powersave_gci_gpio_wakeup_disable <pin>

- Disables waking up from deep-sleep via GPIO.

mcu_powersave_gci_gpio_wakeup_ack <pin>

- If GPIO generates a wake up event, it remains triggered till acked.

thread_list

- [ThreadX only] Lists the currently running threads.

thread_kill <thread id>

- [ThreadX only] Kills the specified running thread.

thread_spawn [--priority <priority>] [--interval <interval_in_ms>] <cmds>

- [ThreadX only] Spawns a new thread to execute the specified console command. Optional parameter --priority <priority> & --interval <interval_in_ms>

ethernet_up [ip netmask gateway]

- Brings up the Ethernet. DHCP is assumed if an IP address isn't specified.

ethernet_down

- Takes down the Ethernet.

ethernet_ping <destination> [-i <interval in ms>] [-n <number>] [-l <length>]

- Pings the specified IP or host via Ethernet.

network_suspend

- Suspends the network.

network_resume

- Resumes the network.

wl [<command> [option...]]

- wl commands

bt_hci_reset

- Sends the BT HCI reset command.

bt_download_firmware

- Downloads firmware to BT device.

bt_le_tx_test

- BT packet transmit test through the BLE channel.

bt_le_rx_test

- BT packet receive test through the BLE channel.

bt_le_test_end

- Stops BLE transmit/receive test.

bt_radio_tx_test

- BT packet transmit test through the BR/EDR channel.

bt_radio_rx_test

- BT packet receive test through the BR/EDR channel.

bt_read_bd_addr

- Reads the BD Address.

bt_write_bd_addr

- Writes the BD Address.

bt_le_enhanced_tx_test

- Enhanced BT packet transmit test through the BLE channel.

bt_le_enhanced_rx_test

- Enhanced BT packet receive test through the BLE channel.

bt_set_tx_frequency_arm

- BT transmits on frequency.

bt_receive_only

- BT receives on frequency.

>