AN214799

Manufacturing Bluetooth Test Tool WICED™ Studio 4

Associated Part Family: CYW2070x, CYW20735, CYW20719

This document provides information and test examples on using the manufacturing Bluetooth test tool to test and verify the RF performance of the CYW207xx family of SoC Bluetooth devices. It is intended for development and test engineers who are working with those devices.

Contents

1 Introduction

The manufacturing Bluetooth test tool (MBT) is used to test and verify the RF performance of the Cypress SoC Bluetooth BR/EDR/LE devices. For LE tests standard procedures from the Bluetooth Core Specification 0 are utilized. For BR/EDR tests a set of vendor specific commands are introduced and described in this document. Each test sends an HCI command to the device and then waits for an HCI Command Complete event from the device.

2 Setup

2.1 Device Configuration

The USB transport is used to test the CYW20704 device which should be connected to any PC USB port. To run over USB the BTWUSB driver package should be installed and the device should be enumerated in the "Bluetooth Devices" section of the Windows Device Manager.

It is assumed that all other Cypress Bluetooth devices expose HCI UART and that this UART can be connected to a COM port or to a Serial to USB device of a PC. The HCI UART supports HCI Commands and Events described in this document.

For the CYW20706, the device should be powered on after it has been connected to the PC.

Check the device specific Quick Start Guide for any DIP switch settings to configure the device in HCI mode.

2.2 Environment Variables

To configure MBT to run over USB set the system environment variable:

MBT_TRANSPORT=BTWUSB-0

If more than one device is plugged in, the consecutive devices can be tested by setting the environment variable to BTWUSB-1 and so on.

To configure MBT to run over the HCI UART set the following system environment variables:

MBT_TRANSPORT=COMxx

DOWNLOAD_BAUDRATE=4000000

APPLICATION_BAUDRATE=3000000

The COMxx corresponds to the COM port assigned to the device's HCI UART. Consult the Quick Start Guide for the chip for the instructions how to find out the COM port.

When a COM port is selected, a higher speed can be used for faster downloading by setting the DOWNLOAD_BAUDRATE environment variable. The download speed is dependent on the platform. E.g. 3000000 or 4000000.

If the DOWNLOAD_BAUDRATE environment variable is not set, the default speed of 115200 bits per second will be used.

The APPLICATION_BAUDRATE environment variable needs to be set if the application transport is configured for a speed different than 115200 bits per second.

3 Reset Test

This test verifies that the device is correctly configured and connected to the PC.

Usage: **`mbt reset`**

The example below sends an HCI Reset command to the device and processes the HCI Command Complete event (Refer to the Reference [1] [Vol 2, Part E], Section 7.3.2 for details).

```
<WICED-Studio>\wiced_tools\mbt\win32>mbt reset
Sending HCI Command:
```

```
0000 < 03 0C 00 >
Received HCI Event:
0000 < 0E 04 01 03 0C 00 >
Success
```

The last byte of the HCI Command Complete event is the operation status, where 0 signifies success.

4 LE Receiver Test

This test configures the chip to receive reference packets at a fixed interval. External test equipment should be used to generate the reference packets.

The channel on which the device listens for the packets is passed as a parameter. BLE devices use 40 channels, each of which is 2 MHz wide. (Refer to Reference [1] [Vol 2, Part E], Section 7.8.28 for details).

- Channel 0 maps to 2402 MHz
- Channel 39 maps to 2480 MHz

Usage: `mbt le_receiver_test <rx_channel>`

Where:

- rx_channel = receive frequency minus 2402 divided by 2. For example, if the desired receive frequency is 2406 MHz then the rx_channel = (2406 – 2402) / 2 = 2.

The channel range is 0–39 (2402–2480 MHz).

The example below starts the LE receiver test on Channel 2 (2406 MHz):

```
<WICED-Studio>\wiced_tools\mbt\win32>mbt le_receiver_test 2
Sending HCI Command:
0000 < 1D 20 01 02 >
Received HCI Event:
0000 < 0E 04 01 1D 20 00 >
Success
LE Receiver Test running, to stop execute mbt le_test_end
```

The last byte of the HCI Command Complete event is the operation status, where 0 signifies success. Use `mbt le_test_end` to complete the test and print the number of received packets.

Note: This test will fail if the device is running another test: use `le_test_end` or `reset` to put the CYW207xx in idle state before running this test.

5 LE Transmitter Test

The LE Transmitter Test configures the CYW207xx to send test packets at a fixed interval. External test equipment may be used to receive and analyze the reference packets.

The channel on which the CYW207xx transmits the packets is passed as a parameter. BLE devices use 40 channels, each of which is 2 MHz wide.

- Channel 0 maps to 2402 MHz
- Channel 39 maps to 2480 MHz.

The other two parameters specify the length of the test data and the data pattern to be used (Refer Reference [1] [Vol 2, Part E], Section 7.8.29 for details).

Usage: `mbt le_transmitter_test <tx_channel> <data_length> <packet_payload>`

Where:

- tx_channel = transmit frequency minus 2402 divided by 2. For example, if the transmit frequency is 2404 MHz then the tx_channel = (2404 – 2402) / 2 =1. The channel range is 0–39 (2402–2480 MHz):
- data_length = 0–37
- data_pattern = 0–7

  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

The example below starts the test and instructs the device to transmit packets on Channel 2 (2406 MHz), with a 10-byte payload of all ones (1s):

```
<WICED-Studio>\wiced_tools\mbt\win32>mbt le_transmitter_test 2 10 4
Sending HCI Command:
0000 < 1E 20 03 02 0A 04 >
Received HCI Event:
0000 < 0E 04 01 1E 20 00 >
Success
LE Transmitter Test running, to stop execute mbt le_test_end
```

The last byte of the HCI Command Complete event is the status of the operation, where 0 signifies success.

Use `mbt le_test_end COMx` to complete the test.

Note: This test will fail if the device is running another test: use `le_test_end` or `reset` to put the CYW207xx in idle state before running this test.

6 LE Test End

This command stops the LE Transmitter or LE Receiver Test that is in progress on the CYW207xx.

The number of packets received during the test is reported by the device and printed out. The value will always be zero (0) if the LE Transmitter Test was active (Refer to the Reference [1] [Vol 2, Part E], Section 7.8.30 for details).

Usage: `mbt le_test_end`

The example below stops the active test:

```
<WICED-Studio>\wiced_tools\mbt\win32>mbt le_test_end
Sending HCI Command:
0000 < 1F 20 00 >
Received HCI Event:
0000 < 0E 06 01 1F 20 00 00 00 >
Success num_packets_received 0
```

7 Continuous Transmit Test

Note: Unlike the LE tests, this test uses 79 frequencies, each 1 MHz wide.

This test configures the CYW207xx to turn the carrier ON or OFF. When the carrier is ON the device transmits according to the specified transmit mode, modulation type, frequency, and power level.

Usage: `mbt set_tx_frequency_arm <carrier on/off> <tx_frequency> <tx_mode> <tx_modulation_type> <tx_power>`

Where:

- carrier on/off:

  1: carrier ON

  0:carrier OFF
- tx_frequency: (2402 – 2480) transmit frequency, in 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, 2Mbps, or 3 Mbps modulation. Ignored if mode is unmodulated.

  0: GFSK

  1: QPSK

  2: 8PSK

  3: LE
- tx_power = (–25 to +3) transmit power, in dBm

The example below turns the carrier ON and instructs the CYW207xx to transmit an unmodulated pattern on 2402 MHz at 3 dBm.

```
<WICED-Studio>\wiced_tools\mbt\win32> mbt set_tx_frequency_arm 1 2402 1 2 3
Sending HCI Command:
0000 < 14 FC 07 00 02 01 02 08 03 00 >
Received HCI Event:
0000 < 0E 04 01 14 FC 00 >
Success
```

To stop the test, send the command a second time to the same COM port with the carrier on/off parameter set to zero (0). No other parameters are used.

```
<WICED-Studio>\wiced_tools\mbt\win32>mbt set_tx_frequency_arm 0
Sending HCI Command:
0000 < 14 FC 07 01 02 00 00 00 00 00 >
Received HCI Event:
0000 < 0E 04 01 14 FC 00 >
Success
```

8 Continuous Receive Test

This test configures CYW207xx to turn on the receiver in a non-hopping continuous mode. The frequency to be used by the CYW207xx is passed as a parameter.

Usage: `mbt receive_only <rx frequency>`

Where:

- rx_frequency = (2402 – 2480) receiver frequency, in MHz

The example below instructs the CYW207xx to set the receiver to frequency of 2046 MHz.

```
<WICED-Studio>\wiced_tools\mbt\win32> mbt receive_only 2406
Sending HCI Command:
0000 < 2B FC 01 06 >
Received HCI Event:
0000 < 0E 04 01 2B FC 00 >
Success
```

9 Radio TX Test

This test is the connectionless transmit test that sends Bluetooth packets. The test configures the CYW2070X to transmit the selected data pattern which is governed by a specified frequency and a specified logical channel at a specified power level.

Usage: `mbt radio_tx_test <bd_addr> <frequency> <modulation_type> <logical_channel> <bb_packet_type> <packet_length> <tx_power>`

Where:

- bd_addr: BD_ADDR of Tx device (6 bytes)
- frequency: Set to 0 to use a normal Bluetooth hopping sequence, or 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: 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: 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: –25 dBm to +3 dBm

The example below instructs the CYW2070X to transmit 0xAA pattern on 2402 MHz frequency using an ACL connection with Basic Rate DM1 packets at –3 dBm.

```
<WICED-Studio>\wiced_tools\mbt\win32> mbt radio_tx_test 2402 2 1 3 17 -3
Sending HCI Command:
0000 < 51 FC 10 45 23 01 3A 70 20 01 00 03 01 03 11 00 08 FD 00 >
Received HCI Event:
0000 < 0E 04 01 51 FC 00 >
Success
```

The last byte of the HCI Command Complete event is the operation status, where 0 signifies that the operation was successful and the test started to run. The test continues to run until the board is reset.

10 Radio RX Test

This test issues a command to the CYW2070X to set the radio to camp on a specified frequency. While the test is running, the CYW2070X periodically sends reports about received packets.

Usage: `mbt radio_rx_test <bd_addr> <frequency> <modulation_type> <logical_channel> <bb_packet_type> < packet_length>`
Where:

- bd_addr: BD_ADDR of Tx device (6 bytes)
- frequency: Frequency to listen to 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: 0 to 65535. The device compares the length of the received packets with the specified packet_length.

CYW2070x generates a statistics report of the RX Test every 1 second when testing is performed.

The example below instructs the CYW2070X to tune the receiver on 2402 MHz frequency. The test verifies that the 0xAA pattern is received using DM1 packet types (Basic Rate).

```
<WICED-Studio>\wiced_tools\mbt\win32>mbt radio_rx_test 20703A012345 2402 1 2 3 17
Sending HCI Command:
0000 < 52 FC 0E 45 23 01 3A 70 20 E8 03 00 03 01 03 11 00
Received HCI Event:
0000 < 0E 04 01 52 FC 00 >
Success
Rx Test is running. Press any key to stop the test.
```

The MBT reports connectionless Rx Test statistics every second.

The example below shows the Rx Test statistics report:

```
<WICED-Studio>\wiced_tools\mbt\win32>
0000 < 07 01 00 00 00 00 00 00 00 1D 03 00 00 1D 03 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 >
Sync_Timeout_Count = 0x1
HEC_Error_Count = 0x0
Total_Received_Packets =
0x31D
Good_Packets = 0x31D
CRC_Error_Packets = 0x0
Total_Received_Bits = 0x0
Good_Bits = 0x0
Error_Bits = 0x0
```

Press any key to stop the test.

11 Connectionless DUT Loopback Mode

The basic concept for this test is derived using the loopback mode. The tester transmits a specific packet to the DUT which is retransmitted. This structure enables the tester to analyze both the TX and RX characteristics.

Usage: `mbt connectionless_dut_loopback_mode [Interactive Input Arguments]`
Where:

- Remote_Device_BD_ADDR: BD_ADDR of the remote transmitting device. [Size: 6 bytes]
- LT_ADDR: The logical transport address of the BT link. [Size: 1 byte] [Range: 0x01 - 0x07]
- Number_Of_Tests: 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: Number of packets to be received for this test.

  [Size: 15 bits {25:11} in little endian uint32] [Range: 0x01 - 0x7fff].
- TxPowerIndex: Power table index to use.

  [Size: 3 bits {10:8} in little endian uint32] [Range: 0x00 - 0x07].
- RxChannel: Frequency offset in MHz from 2402 MHz.

[Size: 7bits {7:1} in little endian uint32] [Range: 0x00 - 0x7f].

- Packet Type: Defines the type of the packet. The following defines the value which can be provided:

  0: BR

  1: EDR

  [Size: 1bit {0:0} in little endian uint32].
- Retry Time Out: The time required to retry.

  [Size: 1 byte] [Range: 0x01 - 0xff].
- Test Scenarios: The following defines the value which can be provided:

  0: Rx-Tx Loopback Mode

  1: Rx-only with BER stats

  [Size: 1 byte].

The example below runs the loopback command for one test:

```
<WICED-Studio>\wiced_tools\mbt\win32>mbt connectionless_dut_loopback_mode
Enter the following parameters in hexadecimal [0xXX]
Enter the BD address [6 bytes of length]:
0x00 0x00 0x00 0xc0 0xff 0xee
Enter the LT address [0x01-0x07]:
0x07
Enter the number of test [0x01-0x10]:
0x01
Enter the retry offset[1]-[0x01-0x3f]:
0x01
Enter the no of packets[1]-[0x01-0x7fff]:
0x64
Enter the TX power[1]-[0x00-0x07]:
0x00
Enter the RX power[1]-[0x00-0x7f]:
0x28
Enter the packet type[1]-[0-Basic][1-Enhanced]:
0x0
Enter the retry timeout[1]-[0x01-0xff]:
0x28
Enter the test_scenarios[1]:
0x0
Sending HCI Command:
0000 < 54 FC 0E EE FF C0 00 00 00 07 01 50 20 03 04 >
0010 < 28 00 >
Received HCI Event:
0000 < 0E 06 01 54 FC 00 00 00 >
Success
```

The last three bytes of the command complete event are the operation status, test state, and Test Rnd, where 0 signifies success.

References

[1] Bluetooth Core Specification, Version 4.2 (see Bluetooth Core Specification 4.2)

[2] WICED Studio Quick Start Guide for BT CYW20706 (001-CYWICED004)

Document History

Document Title: AN214799 - Manufacturing Bluetooth Test Tool

Document Number: 002-14799

Revision	ECN	Submission Date	Description of Change
**	5450962	10/17/2016	Initial revision - preliminary

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Cypress Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

WICED IoT

Uniting CDC and WICED Solutions

Technical Support

cypress.com/support

PSoC is a registered trademark and WICED and PSoC Creator are trademarks of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com