# AIROC™ Manufacturing Bluetooth® Test Tool user guide

## About this document

### Scope and purpose

This is the user guide for the AIROC™ Manufacturing Bluetooth® Test Tool (WMBT), which is used to test and verify the RF performance of the Infineon Bluetooth® BR/EDR/LE AIROC™ devices.

### Intended audience

This document is intended for anyone to test and verify the RF performance of the Infineon Bluetooth® AIROC™ devices.

### References


[1] Bluetooth® Core Specification, version 5.2 (see **Bluetooth® Core Specification 5.2**)

[2] AIROC™ HCI UART Control Protocol (002-16618)

[3] BQB RF test setup (002-15369)

# Table of contents

**About this document** ..... **1**
**Table of contents** ..... **2**
**1 Introduction** ..... **3**
**2 Setup** ..... **4**
2.1 Device configuration ..... 4
2.2 Environment variables ..... 4
2.2.1 MBT_BAUD_RATE ..... 4
2.3 TRANSPORT_MODE ..... 4
**3 Reset test** ..... **5**
3.1 Reset ..... 5
3.2 reset_highspeed ..... 5
**4 LE receiver test** ..... **6**
**5 LE transmitter test** ..... **7**
**6 LE test end** ..... **8**
**7 Continuous transmit test** ..... **9**
**8 Continuous receive test** ..... **11**
**9 Radio TX test** ..... **12**
**10 Radio RX test** ..... **14**
**11 BQB RF test** ..... **16**
**12 Read BD_ADDR command** ..... **17**
**13 Factory commit BD_ADDR command** ..... **18**
**Revision history** ..... **19**

# 1 Introduction

The AIROC™ Manufacturing Bluetooth® Test Tool (WMBT) is used to test and verify the RF performance of the Infineon Bluetooth® BR/EDR/LE devices. For LE tests, standard procedures from the Bluetooth® Core Specification are utilized. For BR/EDR tests, a set of vendor-specific commands are introduced and described in this document. Each test sends a Host Controller Interface (HCI) or AIROC™ HCI command to the device and then waits for an HCI command complete event from the device.

# 2 Setup

## 2.1 Device configuration

The Infineon Bluetooth® device to be tested must expose an HCI UART; 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.

The device should be preprogrammed with an application image and should be reset after it has been connected to the PC and the COM port drivers are loaded. The WMBT tool is part of the AIROC™ Bluetooth® SDK and can be found under the *wiced_btsdk* folder in the ModusToolbox™ workspace after an application project has been created and downloaded to the device using ModusToolbox™. For example, on Windows, the default location is:

*C:\Users\<username>\mtw\mtb_shared\wiced_btsdk\tools\btsdk-utils\<branch>\wmbt*

See the device-specific kit guide or quick start guide for any DIP switch or jumper settings to configure the device to expose the HCI UART interface.

## 2.2 Environment variables

### 2.2.1 MBT_BAUD_RATE

Infineon Bluetooth® devices support adjustable baud rates up to 4 Mbps via the `wiced_transport_init()` API included with the AIROC™ Bluetooth® SDK. If this API is not used in an application to re-configure the baud rate, the default rate of 115.2 kbps will be used by the device. The `MBT_BAUD_RATE` environment variable must be set to match what the device is using before running WMBT.

As an example, on Windows, use the following command to configure `MBT_BAUD_RATE` as 3 Mbps:

```
<wmbt-path>\bin> set MBT_BAUD_RATE=3000000
```

## 2.3 TRANSPORT_MODE

The Bluetooth® Core Specification [1] defines the HCI, which provides a standardized communication protocol between the Bluetooth® host stack and Bluetooth® controller. Infineon Bluetooth® devices provide a high level of integration, for example, Bluetooth® controller and embedded Bluetooth® host stack in a single chip, to simplify Bluetooth® product development for customers so that they are not required to be familiar with all HCI commands and events.

Typically, when the embedded stack is used in the Infineon device, and it interfaces with an onboard MCU, the MCU software would likely need to send and receive commands and events to the Infineon device. For such a solution, AIROC™ HCI is defined and provided as an example; see AIROC™ HCI UART Control Protocol [2].

The WMBT provides support for both HCI and AIROC™ HCI via the `TRANSPORT_MODE` environment variable. If AIROC™ HCI is required, your application must implement handlers for `HCI_CONTROL_TEST_COMMAND_ENCAPSULATED_HCI_COMMAND`; see hci_control_test.c included with the watch sample application. HCI should be sufficient for most cases because the devices support this by default. The `TRANSPORT_MODE` environment variable must be set to the required mode before running the WMBT.

As an example, on Windows, use the following command to configure `TRANSPORT_MODE` for HCI:

```
<wmbt-path>\bin>set TRANSPORT_MODE=0
```

# 3 Reset test

This test verifies that the device is correctly configured and connected to the PC. If your application re-configures the baud rate, use the `reset_highspeed` command.

## 3.1 Reset

**Description:**

Sends an HCI Reset command at 115.2 kbps to the device and processes the HCI Command Complete event (See Reference **[1]** [Vol 2, Part E], Section 7.3.2 for details).

**Usage:**

```
wmbt reset COMx
```

**Example:**

```
<wmbt-path>\bin> wmbt reset COM23
Opened COM23 at speed: 115200
Sending HCI Command:
0000 < 01 03 0C 00 >
Received HCI Event:
0000 < 04 0E 04 01 03 0C 00 >
Success
Close Serial Bus
```

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

## 3.2 reset_highspeed

**Description:**

Sends an HCI Reset command at the configured `MBT_BAUD_RATE` to the device and processes the HCI Command Complete event (See Reference **[1]** [Vol 2, Part E], Section 7.3.2 for details).

**Usage:**

```
wmbt reset_highspeed COM23
```

**Example:**

```
<wmbt-path>\bin> wmbt reset_highspeed COM23
Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 03 0C 00 >
Received HCI Event:
0000 < 04 0E 04 01 03 0C 00 >
Success
Close Serial Bus
```

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. Use external test equipment to generate the reference packets.

The frequency on which the device listens for the packets is passed as a parameter. Bluetooth® LE devices use 40 channels, each of which is 2-MHz wide. (See Reference **[1]** [Vol 2, Part E], Section 7.8.28 for details).

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

The following equation can be used to map the channel number to the actual center frequency:

$$Frequency = ( 2 \times Channel ) + 2402MHz$$

**Usage:**

```
wmbt le_receiver_test COMx <rx_frequency>
```

Where:

- rx_frequency = (2402 - 2480) receive frequency, in MHz

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

```
<wmbt-path>\bin> wmbt le_receiver_test COM23 2406
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 1D 20 01 02 >
Received HCI Event:
0000 < 04 0E 04 01 1D 20 00 >
Success
Close Serial Bus
```

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

Use `wmbt le_test_end COMx` 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` *to put the device in an idle state before running this test.*

# 5 LE transmitter test

The LE transmitter test configures the Infineon Bluetooth® device to send test packets at a fixed interval. Use external test equipment to receive and analyze the reference packets.

The frequency at which the device transmits the packets is passed as a parameter. Bluetooth® LE devices use 40 channels, each of which is 2-MHz wide. (See Reference **[1]** [Vol 2, Part E], Section 7.8.28 for details).

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

**Usage:**

```
wmbt le_transmitter_test COMx <tx_frequency> <data_length> <data_pattern>
```

Where:

- rx_frequency = (2402 - 2480) receive frequency, in 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 following example starts the test and instructs the device to transmit packets on channel 2 (2406 MHz), with a 10-byte payload of all ones (1s):

```
<wmbt-path>\bin> wmbt le_transmitter_test COM23 2406 10 4
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 1E 20 03 02 0A 04 >
Received HCI Event:
0000 < 04 0E 04 01 1E 20 00 >
Success
Close Serial Bus
```

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

Use `wmbt le_test_end COMx` to complete the test.

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

# 6 LE test end

This command stops the LE transmitter or LE receiver test that is in progress.

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

**Usage:**

```
wmbt le_test_end COMx
```

The following example stops the active test:

```
<wmbt-path>\bin> wmbt le_test_end COM23
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 1F 20 00 >
Received HCI Event:
0000 < 04 0E 06 01 1F 20 00 00 00 >

Success num_packets_received = 0

Close Serial Bus
```

# 7 Continuous transmit test

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

This test configures the Infineon Bluetooth® device 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:**

```
wmbt tx_frequency_arm COMx <carrier on/off> <tx_frequency> <mode>
<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 0s
  - 4: All 1s
  - 5: Incrementing symbols
- tx_modulation_type: selects 1 Mbps, 2 Mbps, 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 following example turns the carrier ON and instructs the device to transmit an unmodulated pattern at 2402 MHz at 3 dBm.

```
<wmbt-path>\bin> wmbt tx_frequency_arm COM23 1 2402 1 2 3
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 14 FC 07 00 00 01 02 08 03 00 >
Received HCI Event:
0000 < 04 0E 04 01 14 FC 00 >
Success
Close Serial Bus
```

### Continuous transmit test

To stop the test, send the command to the same COM port with the carrier ON/OFF parameter set to zero (0).

```
<wmbt-path>\bin> wmbt tx_frequency_arm COM23 0 2402 1 2 3
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 14 FC 07 01 02 00 00 00 00 00 >
Received HCI Event:
0000 < 04 0E 04 01 14 FC 00 >
Success
Close Serial Bus
```

# 8 Continuous receive test

This test configures the Infineon Bluetooth® device to turn ON the receiver in a non-hopping continuous mode. The frequency to be used by the device is passed as a parameter.

**Usage:**

```
wmbt receive_only COMx <rx_frequency>
```

Where:

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

The following example instructs the Infineon Bluetooth® device to tune the receiver frequency to 2046 MHz.

```
<wmbt-path>\bin> wmbt receive_only COM23 2406
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 2B FC 01 04 >
Received HCI Event:
0000 < 04 0E 04 01 2B FC 00 >
Success
Close Serial Bus
```

# 9 Radio TX test

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

**Usage:**

```
wmbt radio_tx_test COMx <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 the 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 following example instructs the Infineon Bluetooth® device to transmit 0xAA pattern at 2402 MHz frequency using an ACL connection with Basic Rate DM1 packets at –3 dBm.

```
<wmbt-path>\bin> wmbt radio_tx_test COM23 112233445566 2402 2 1 3 17 -3
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)
Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 51 FC 10 66 55 44 33 22 11 01 00 03 01 03 11 >
0010 < 00 08 FD 00 >
```

### Radio TX test

```
Received HCI Event:
0000 < 04 0E 04 01 51 FC 00 >
Success
Close Serial Bus
```

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 Infineon Bluetooth® device to set the radio to camp on a specified frequency. While the test is running, the Bluetooth® device periodically sends reports about received packets.

**Usage:**

```
wmbt radio_rx_test COMx <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.

The Infineon Bluetooth® device generates a statistics report of the RX test every 1 second when testing is performed.

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

```
<wmbt-path>\bin> wmbt radio_rx_test COM23 112233445566 2402 2 1 3 17
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)
Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 52 FC 0E 66 55 44 33 22 11 E8 03 00 03 01 03 >
0010 < 11 00 >
```

### Radio RX test

```
Received HCI Event:
0000 < 04 0E 04 01 52 FC 00 >
Success
Radio RX Test is running. Press the Enter key to stop the test.
```

The WMBT reports connectionless Rx Test statistics every second.

The following example shows the Rx Test statistics report:

```
Statistics Report received:
  [Rx Test statistics]
    Sync_Timeout_Count:     0x0
    HEC_Error_Count:        0x0
    Total_Received_Packets: 0x31f
    Good_Packets:           0x31f
    CRC_Error_Packets:      0x0
    Total_Received_Bits:    0x1a878
    Good_Bits:              0x1a878
    Error_Bits:             0x0
```

Press **Enter** to stop the test.

# 11 BQB RF test

This test issues the commands necessary to configure the Infineon Bluetooth® device into a test mode for BQB RF testing using a Bluetooth® tester; see BQB RF test setup [3].

**Usage:**

```
wmbt enable_bqb_test_mode COMx
```

Before executing this command to configure the device for test mode, you must ensure that your application does not have any timers running or any over-the-air Bluetooth® activity enabled. For example, if advertisements are enabled or a periodic application timer is enabled, it may be possible to interfere with the BQB test results.

**Example:**

```
<wmbt-path>\bin> wmbt enable_bqb_test_mode COM23
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 05 0C 03 02 00 02 >
Received HCI Event:
0000 < 04 0E 04 01 05 0C 00 >
Success
Close Serial Bus
Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 1A 0C 01 03 >
Received HCI Event:
0000 < 04 0E 04 01 1A 0C 00 >
Success
Close Serial Bus
Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 03 18 00 >
Received HCI Event:
0000 < 04 0E 04 01 03 18 00 >
Success
Close Serial Bus
```

# 12 Read BD_ADDR command

This command reads the `BD_ADDR` that is currently programmed in the DUT.

**Usage:**

```
wmbt read_bd_addr COMx
```

**Example:**

```
<wmbt-path>\bin> wmbt read_bd_addr COM23

MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 09 10 00 >
Received HCI Event:
0000 < 04 0E 0A 01 09 10 00 66 55 44 33 22 11 >

Success BD_ADDR = 112233445566

Close Serial Bus
```

# 13 Factory commit BD_ADDR command

This command writes the `BD_ADDR` to the Static Section (SS) area of the flash.

To use this command, the `BD_ADDR` must initially be set to all FFs.

To set the initial `BD_ADDR` to all FFs, build and download the example application into the device with ModusToolbox™ command line make, including the `BT_DEVICE_ADDRESS` directive in your make command.

For example:

```
make BT_DEVICE_ADDRESS=FFFFFFFFFFFF program
```

**Usage:**

```
wmbt factory_commit_bd_addr COMx <bd_addr>
```

**Example:**

```
<wmbt-path>\bin> wmbt factory_commit_bd_addr COM23 112233445566
MBT_BAUD_RATE:  3000000
TRANSPORT_MODE: 0 (HCI)

Opened COM23 at speed: 3000000
Sending HCI Command:
0000 < 01 10 FC 07 66 55 44 33 22 11 00 >
Received HCI Event:
0000 < 04 0E 04 01 10 FC 00 >
Success
Close Serial Bus
```

## Revision history

| Document version | Date of release | Description of changes |
|---|---|---|
| ** | 2016-02-19 | Initial version |
| *A | 2016-09-27 | Updated in Cypress template |
| *B | 2017-07-27 | Updated logo and copyright |
| *C | 2017-08-23 | Updated template |
| *D | 2018-08-30 | Updated to reflect ModusToolbox™<br>Updated Sales and Copyright |
| *E | 2019-02-20 | Updated Factory Commit BD_ADDR Command<br>Updated template |
| *F | 2019-10-15 | Updated for ModusToolbox™ 2.0 paths |
| *G | 2021-03-09 | Updated for ModusToolbox™ 2.2 + BTSDK 3.0 paths |
| *H | 2021-10-28 | Updated terminology per Bluetooth® SIG and marketing |

**Trademarks**
All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2021-10-28**
**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2021 Infineon Technologies AG. All Rights Reserved.**

**Do you have a question about this document?**

**Go to www.cypress.com/support**

**Document reference**
**002-14799 Rev. *H**

**IMPORTANT NOTICE**
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

**WARNINGS**
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.