AIROC™ test and debug tool user guide

About this document

Version

1.5.0

Scope and purpose

The AIROC™ Test and Debug Tool is a standalone, cross-platform application designed for testing and debugging AIROC™ devices. The tool communicates with **Bluetooth®** the Host/Controller Interface (HCI) and with **Wi-Fi devices** using WL commands executed on a Raspberry Pi controller through a **UART interface**. It supports both **UART** and **USB** physical interfaces for Bluetooth® testing.

The tool provides the following capabilities:

- Send HCI commands to Bluetooth® controllers
- Receive HCI events from Bluetooth® controllers
- Program Bluetooth® controllers
- Run ACL, SCO, and ISOC throughput tests for Bluetooth® controllers
- Send WL commands to Wi-Fi controllers
- Run non-signaling tests for Wi-Fi controllers

Intended audience

This document helps you understand how to use the AIROC™ test and debug tool.

Document conventions

Convention	Explanation
Bold	Emphasizes hotkeys and the names of menu items, tabs, buttons, labels and windows
Italics	Denotes file names and paths
`Courier New`	Denotes command line inputs
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Acronyms and abbreviations

Term	Explanation
ACL	asynchronous connection-less link
BIG	broadcast isochronous group
BIS	broadcast isochronous stream
BTP	A preset file used to program AIROC™ Bluetooth® controllers.
CIG	connected isochronous group
CIS	connected isochronous stream
GUI	graphical user interface
HCI	Host/Controller interface
ISOC	isochronous channel
LAN	Local Area Network

About this document

Term	Explanation
SCO	synchronous connection-oriented link
UART	universal asynchronous receiver-transmitter
USB	universal serial bus
LAN	Local Area Network
WL	Wireless LAN

Table of contents

1 Introduction 5

1.1 System requirements 5
1.2 Package contents 5
1.3 Supported devices 5

2 Installation procedures 7

2.1 Windows 7
2.2 Ubuntu Linux 7
2.3 macOS 8

3 GUI application 9

3.1 Open GUI application 9
3.2 Command line options 9
3.3 Main window description 10
3.4 Select device 12
3.5 Send HCI commands 15
3.6 Receive HCI events 16
3.7 Reset device 17
3.8 Read BD_ADDR of device 18
3.9 Increase connection accept timeout for device 19
3.10 Set baud rate for a selected UART device 20
3.11 Create ACL connection between selected devices 21
3.12 Create an LE connection between selected devices 25
3.13 ACL throughput tests 29
3.14 Create SCO connection between selected devices 33
3.15 SCO throughput tests 36
3.16 Create CIS between selected devices 41
3.17 Create BIS between selected devices 47
3.18 ISOC throughput tests 65
3.19 Wi-Fi non-signaling tests 71
3.20 IEEE 802.15.4 tests 76
3.21 Program a device 78
3.22 Logging 79

4 Console application 82

4.1 Open the console application 82
4.2 Command line options 82
4.3 Open device 82
4.4 Close a device 82
4.5 List HCI commands 83
4.6 Send HCI commands and receive responses 83
4.7 Program device 84
4.8 Print help 85
4.9 Exit console application 86

5 Troubleshooting 87

Table of contents

1 Introduction

1.1 System requirements

The AIROC™ test and debug tool supports the following operating systems:

- Windows 10 and 11 64-bit
- Ubuntu Linux 18.04 LTS, 20.04 LTS, 22.04 LTS
- macOS Big Sur, Monterey, Ventura

The following system configuration is recommended:

- Processor: 1 GHz or faster
- RAM: 4 GB or more
- Hard disk space: 350 MB
- Display: 1280x1024 or more

1.2 Package contents

The tool package contains:

- *airocdebugtoolgui* executable – The GUI application.
- *airocdebugtoolconsole* executable – The console application.
- Drivers used to communicate with Bluetooth® devices:
  - BTWUSB driver (Windows only)
  - KitProg drivers (Windows only)
  - WICED UART driver (Windows only)
  - BTUSB driver (source code, Linux only)
  - BTCOMTT driver (the driver for kernel-mode throughput testing, Windows only).

1.3 Supported devices

The tool supports the following AIROC™ devices.

Stand-alone Bluetooth® chips:	**Combo chips**
CYW20702/5/89071	CYW4330/43303/40183
CYW20730	CYW43340
CYW20732/6/7	CYW4343W/4343A1/43438
CYW20733	CYW4343A2/43439
CYW20703/4/6/7/34/89072	CYW4339/89335
CYW20735B0	CYW43455/4345C0
CYW20735B1	CYW4350C0/4354
CYW20819/20	CYW4354A2/4356
CYW20719/21/39	CYW4356A3/4354
CYW20829	CYW43569/43570
	CYW4359D0/89459/5459x
	CYW4373

Introduction

Stand-alone Bluetooth® chips:	Combo chips
	CYW4349B1/4359B1/89359
	CYW43012
	CYW5557x/CYW89570
	CYW55500

2 Installation procedures

This section describes how to install the tool on different operating systems.

2.1 Windows

1. Run *airocdebugtool-1.5.0.<build>-win-x64.exe*.
2. Follow the installation wizard instructions.

Note: *The airocdebugtool-1.5.0.<build>-win-x64.exe file may require Administrator privileges to install drivers and Microsoft Visual C++ Redistributable.*

2.2 Ubuntu Linux

1. Open a terminal and navigate to the folder where the *airocdebugtool-1.5.0.<build>-linux-x64.deb* file is located.
2. Install the *airocdebugtool-1.5.0.<build>-linux-x64.deb* package. For example:

   ```
   sudo dpkg -i airocdebugtool-1.5.0.<build>-linux-x64.deb
   ```

   You will see the application located in */opt/Tools/AirocDebugTool/<version>*.
3. Navigate to the application folder and provide permissions for the *ChipLoad* and *cgs* executables. For example:

   ```
   cd /opt/Tools/AirocDebugTool/<version>
   sudo chmod +x ChipLoad
   sudo chmod +x cgs
   ```

2.2.1 Build and install the BTUSB driver

To connect to Bluetooth® devices using the USB transport on Ubuntu Linux, first build and install the BTUSB driver. The source code of the BTUSB driver is located in the *drivers/btusb* subdirectory of the installation directory. The BTUSB driver supports Linux kernel 5.8.

To build and install the BTUSB driver:

1. Disconnect all AIROC™ Bluetooth® devices from your computer.
2. Open a terminal and navigate to the folder where the source code of the BTUSB driver is located. For example:

   ```
   cd /opt/Tools/AirocDebugTool/<version>/drivers/btusb
   ```
3. Build and install the BTUSB driver:

   ```
   sudo make
   sudo chmod +x install.sh
   sudo ./install.sh
   ```

Note: *BTUSB is incompatible with BlueZ. Installing the BTUSB driver using install.sh will uninstall BlueZ if installed.*

2.3 macOS

1. Run the *airocdebugtool-1.5.0.<build>-mac-x64.pkg* file.
2. Follow the installation wizard instructions.

3 GUI application

This section describes how to run and use the GUI application.

3.1 Open GUI application

This section describes how to open the GUI application on different operating systems.

3.1.1 Windows

Navigate to the folder where the tool is installed, and run the *airocdebugtoolgui.exe* executable.

3.1.2 Ubuntu Linux

1. Open a terminal and navigate to the folder where the tool is installed. For example:

   ```
   cd /opt/Tools/AirocDebugTool/<version>
   ```

2. Run the *airocdebugtoolgui* executable. For example:

   ```
   ./airocdebugtoolgui
   ```

3.1.3 macOS

Navigate to the folder where the tool is installed, and run the *airocdebugtoolgui* executable.

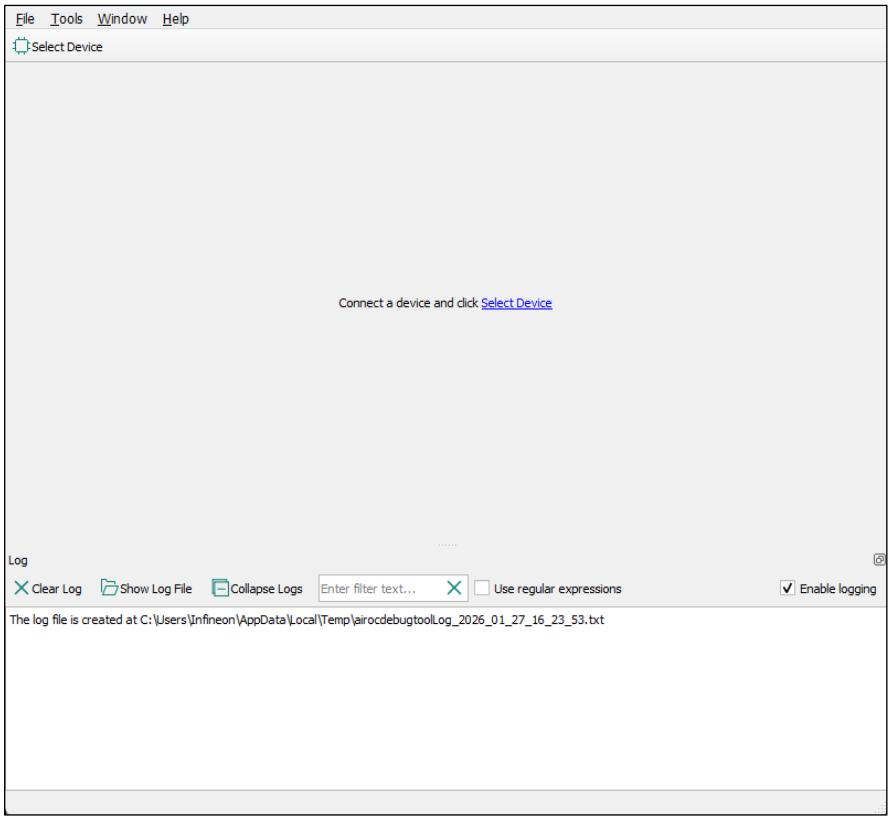
3.2 Command line options

The following table lists command line options which you can use when opening the GUI.

Option	Description
`--help, -h, -?`	Displays help on command line options.
`--help-all`	Displays help on command line options including Qt specific options.
`--override-hcidef-file <filepath>`	Overrides the default HCI definition file.
`--override-lmpdef-file <filepath>`	Overrides the default LPM definition file.
`--override-hciacldef-file <filepath>`	Overrides the default HCI ACL definition file.
`--override-hciscodef-file <filepath>`	Overrides the default HCI SCO definition file.
`--override-hciisocdef-file <filepath>`	Overrides the default HCI ISOC definition file.

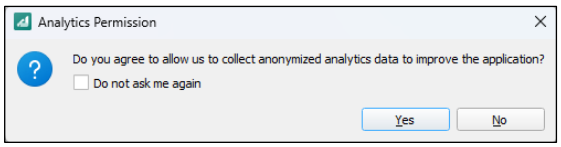
3.3 Main window description

When you open the GUI application, the main window opens.

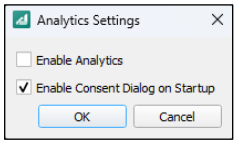


3.3.1 Analytics

When you open the GUI application for the first time, the "Analytics Permission" dialog appears. Analytics collect application usage data to improve most-frequently used features. Click **Yes** to agree, or **No** to decline the collection of non-personal data.

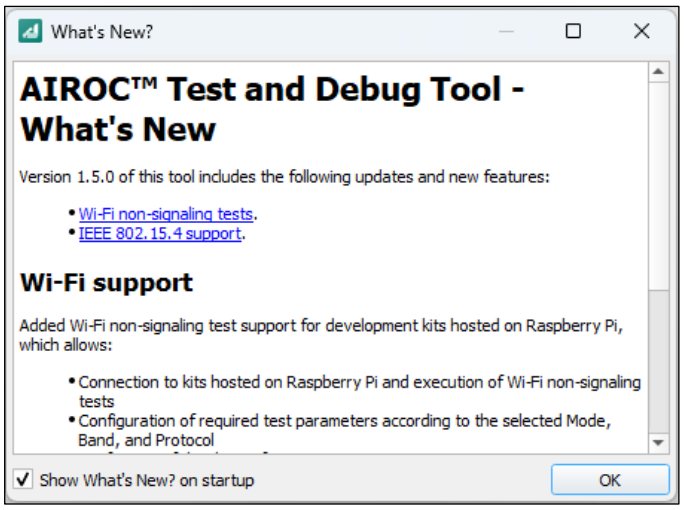


To prevent the dialog from appearing when the application starts, select **Do not ask me again**. You can change these settings at any time using the Analytics settings in the Tools menu.



3.3.2 "What's New?" dialog

When you open the GUI application for the first time, the "What's New?" dialog opens.



To prevent the dialog from opening when the application starts, deselect the **Show What's New? on startup** check box and click **OK**. To open the dialog again, click **Help > What's New?** on the main menu.

The main window contains the following menus:

3.3.3 File menu

- **Select Device (Ctrl+O)** – Open a device.
- **Exit (Alt+F4)** – Exit the application.

3.3.4 Tools menu

- **Log**
  - **Clear Log (Ctrl+D) –** Clears the log window.
  - **Collapse Logs/Expand Logs (Alt+D)** – Collapses or expands the logs displayed in the log window.
  - **Replay Log (Ctrl+R)** – Shows the "Replay Log" dialog allowing you to send HCI commands from a log file.
  - **Show Log File** – Shows the location of the log file.
  - **Options (Ctrl+L)** – Shows the "Log Options" dialog allowing you to set the log options.
- **Rename Devices** – Shows the "Rename Devices" dialog allowing you to change names of the selected devices.
- **Restore Defaults** – Resets all settings to their defaults.
- **Program Device (Ctrl+P)** – Shows the "Program Device" dialog allowing you to program AIROC™ Bluetooth® controllers connected to the computer.

3.3.5 Window menu

- **Reset Window Layout** – Resets the window layout to default.

3.3.6 Help menu

- **View Help (F1)** – Opens this user guide.

GUI application

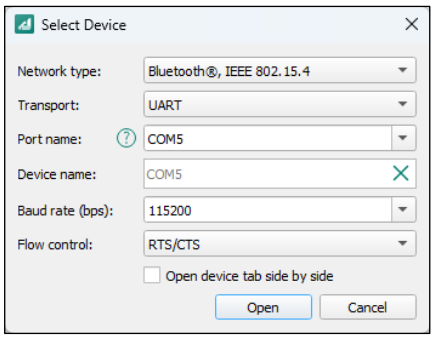
- **What's New**? – Opens the "What's New?" dialog.
- **About AIROC™ test and debug tool** – Shows the **About** dialog displaying information about the tool.

3.4 Select device

Select a device before you can send HCI commands, receive HCI events, and run throughput tests for AIROC™ Bluetooth® devices. Connect the device to your computer and do one of the following:

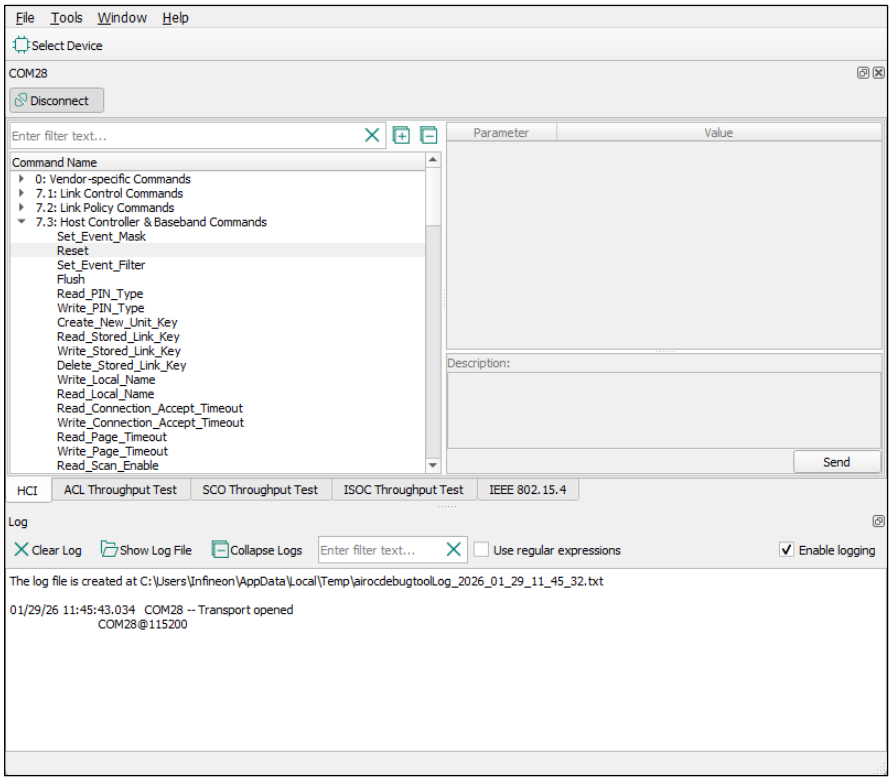
- Select **File > Select Device** on the main menu or press **Ctrl+O**.
- Click **Select Device** on the main toolbar.
- Click the **Select Device** link on the main window.

The "Select Device" dialog opens.



- Select **Network Type** to specify whether the configuration is for **Wi-Fi** or **Bluetooth** testing.
- Select Transport and Port name.
- If you select UART, select **Baud rate** and **Flow control**.
- Optionally, you can set **Device name** and check **Open device tab side by side**.

Click **Open**. When a device is selected, a new device tab or dockable window opens.



GUI application

To check if the device is connected successfully, you can send the **Reset** or **Read_BD_ADDR** command and see if the device responds in the log window.

Note: *For some adapter boards, you might need to install the FTDI virtual COM port (VCP) driver first.*

To select multiple devices, you can use the same GUI instance.

*To select a new device, connect the device to your computer and press **Ctrl+O** or click **Select Device** on the main toolbar. You can also select **File > Select Device** on the main menu.*

3.4.1 Select port name for UART

If you select UART, the **Port name** drop-down list is auto-populated with available port names. Some AIROC™ Bluetooth® devices have two UART ports. Usually, you should use the port with the lower number to select the device.

3.4.2 Select port name for USB

If you select USB, the **Port name** drop-down list is not auto-populated, so enter the port name.

- On Windows, port names have the form **usb<number>**. The port name of the first device that is connected to the computer is usually **usb0**. Port names of other devices are usually subsequent: **usb1**, **usb2**, etc.
- On Linux, port names have the form **btusb<number>**. The port name of the first device that is connected to the computer is usually **btusb0**. Port names of other devices are usually subsequent: **btusb1**, **btusb2**, etc.
- On mac OS, the USB transport is not supported.

Note: *The USB device port name can be changed after programming the device.*

3.4.3 Set device name

Optionally, you can set **Device name** for the selected device. If not set, the device name is the same as the port name.

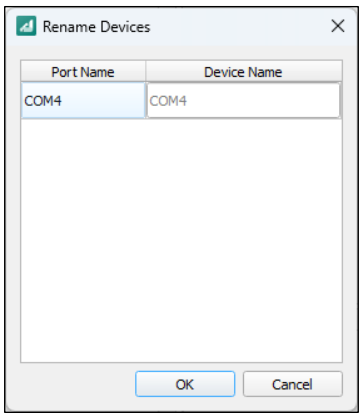
3.4.4 Open device tab side by side

When you select multiple devices using the same GUI instance, you may optionally check **Open device tab side by side** to open a new device tab side by side with a previously opened device tab.

3.4.5 Rename selected device

When a device is selected, to rename the device, select **Tools > Rename Devices** on the main menu.

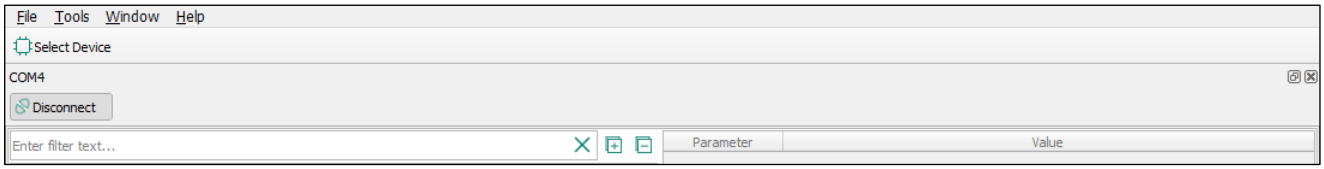
The "Rename Devices" dialog opens.



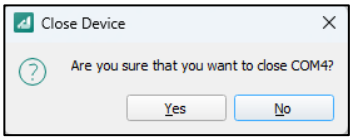
To rename a device, set a new device name on the **Device Name** column and click **OK**.

3.4.6 Disconnect/Connect selected device

When a device is selected, to disconnect/connect the selected device, click **Disconnect/Connect** on the toolbar of the device tab or dockable window. Once disconnected, you cannot send HCI commands to the selected device, receive HCI events from the selected device or run throughput tests for the selected device.



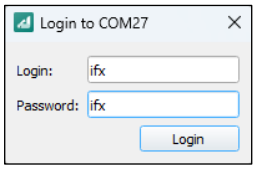
Alternatively, to disconnect the selected device, close the device tab or dockable window. A confirmation dialog opens, click **Yes** to confirm disconnecting the device.



3.4.7 Logging into the Device

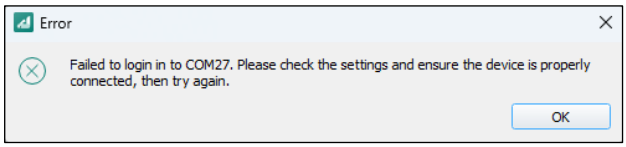
For Wi-Fi devices on Raspberry Pi kits, logging in is required to enable communication with the WL tool before running tests. After selecting the device in **Select Device**, a login dialog will appear. Enter the following default credentials when prompted:

- **Username:** ifx
- **Password:** ifx



If the login fails, the tool will display an error and automatically disconnect the kit.

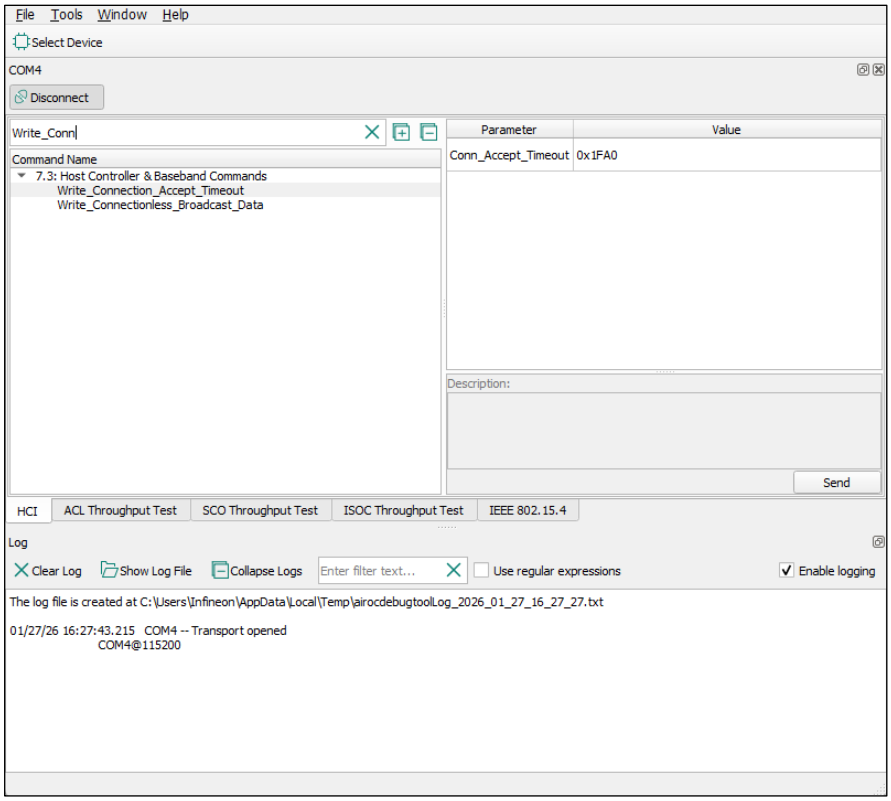
GUI application



3.5 Send HCI commands

To send an HCI command to the selected device, select the **HCI** tab and select **Command Name** on the list of supported commands on the device tab or dockable window. To filter commands by name, use the filter text box (case-insensitive).

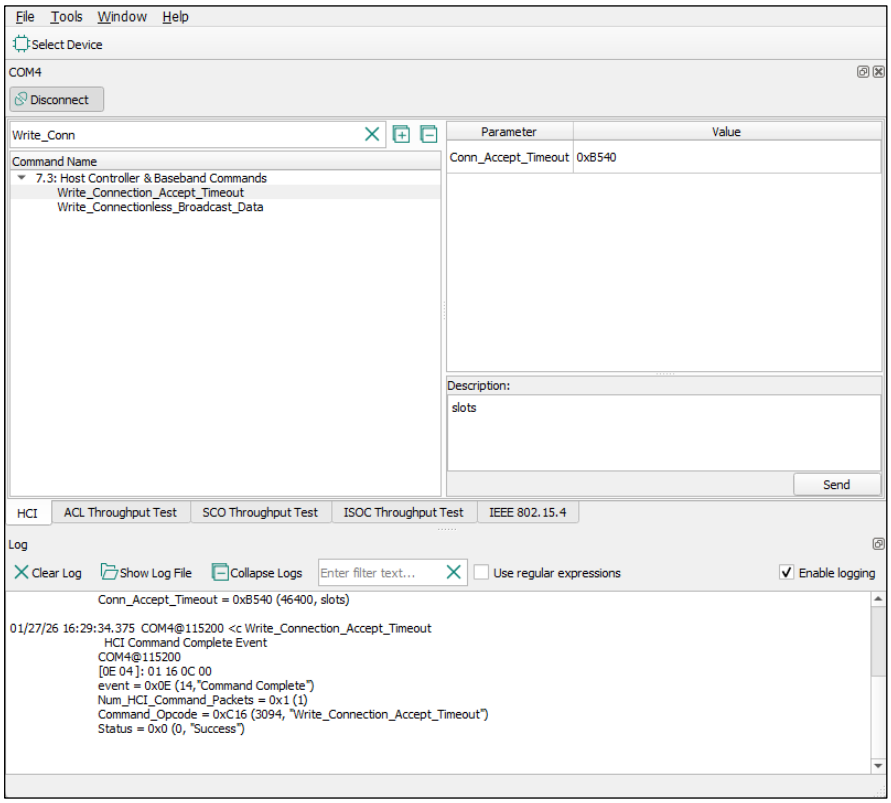
If the selected command has parameters, the list of command parameters displays on the right of the device tab or dockable window.



To send the selected command, set command parameters (if any), and click **Send** or double-click the command name on the list of supported commands.

If the command generates HCI events, event information is printed on the log window.

GUI application



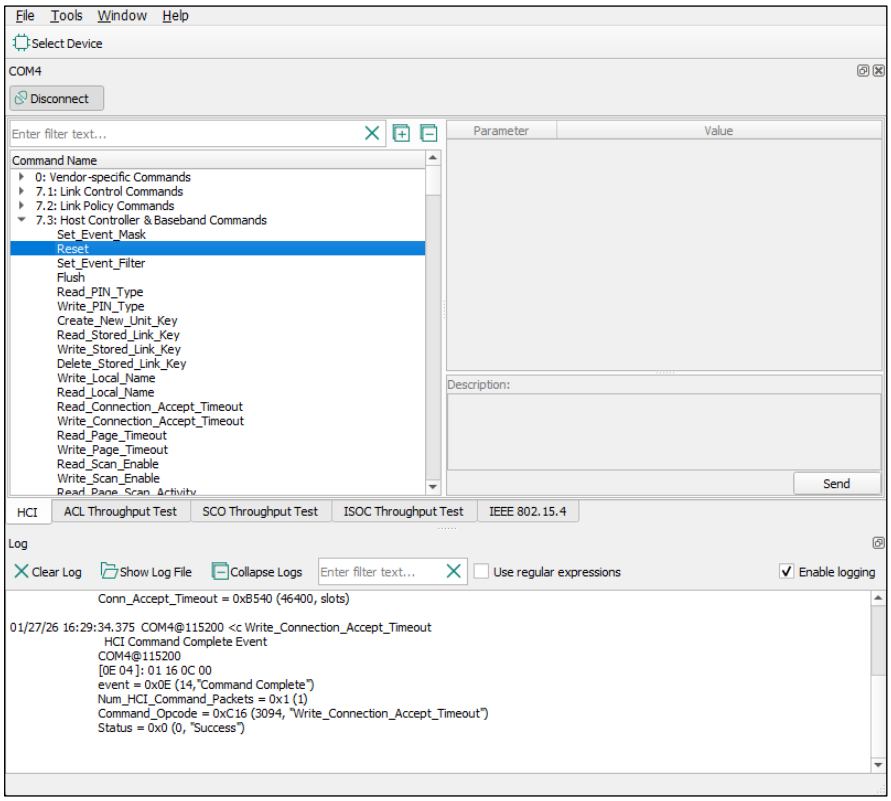
3.6 Receive HCI events

When the selected device generates HCI events, the event information is printed on the log window.

3.7 Reset device

To reset the selected device:

1. Select the **HCI** tab.
2. Select the **Reset** command.
3. Click **Send**.

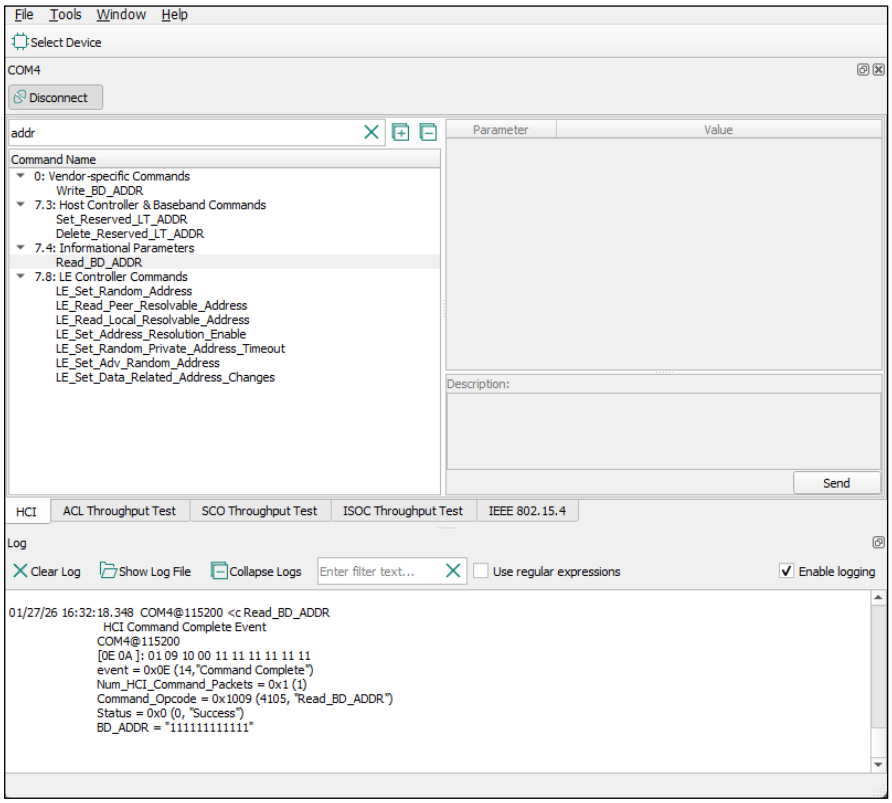


*Note: When you send the **Reset** command, the current operational state of the device is lost and the parameters, for which default values are defined, are reverted to their default values.*

3.8 Read BD_ADDR of device

To read BD_ADDR of the selected device:

1. Select the **HCI** tab.
2. Select the **Read_BD_ADDR** command.
3. Click **Send**.

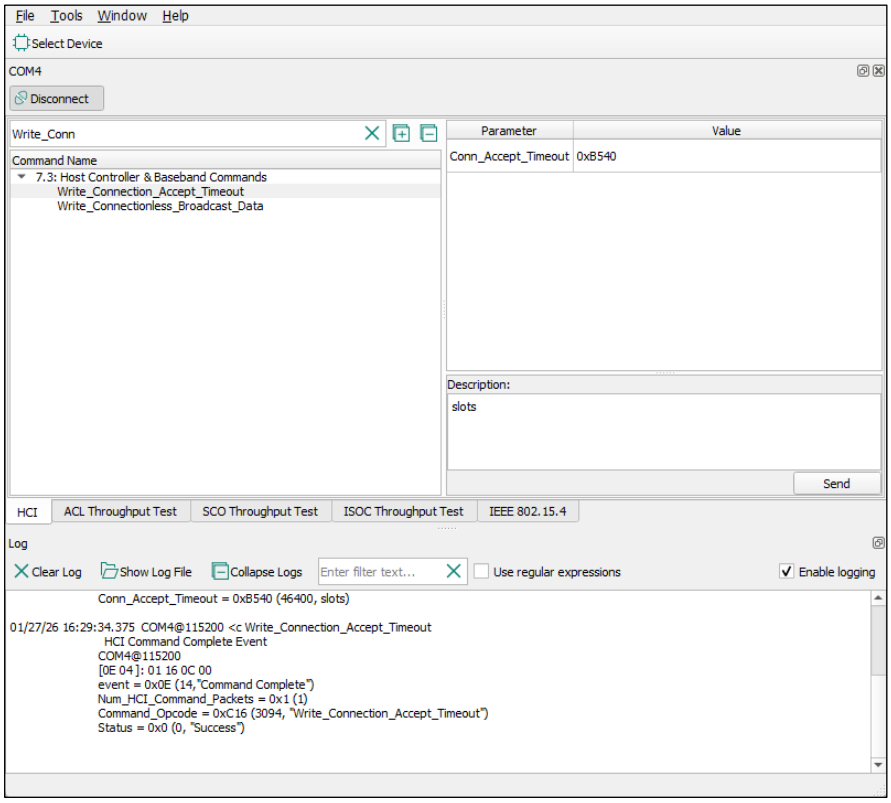


This generates the **Connection Complete** event, and information is printed on the log window. The **BD_ADDR** parameter of the **Connection Complete** event is the device address.

3.9 Increase connection accept timeout for device

To increase the connection, accept the timeout for the selected device:

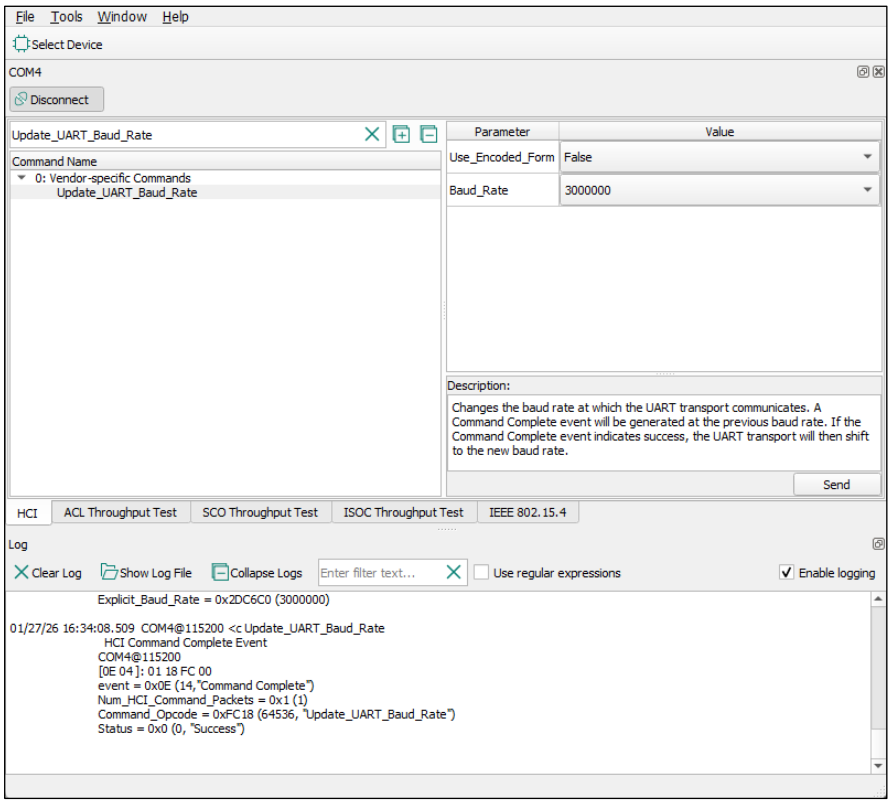
1. Select the **HCI** tab.
2. Select the **Write_Connection_Accept** command.
3. Set the **Conn_Accept_Timeout** command parameter. The maximum value of the **Conn_Accept_Timeout** command parameter is 0xB540, which corresponds to 29 seconds.
4. Click **Send**.



3.10 Set baud rate for a selected UART device

To set the baud rate for a selected UART device:

1. Select the **HCI** tab.
2. Select the **Update_UART_Baud_Rate** command.
3. Set the **Baud_Rate** command parameter.
4. Click **Send**.



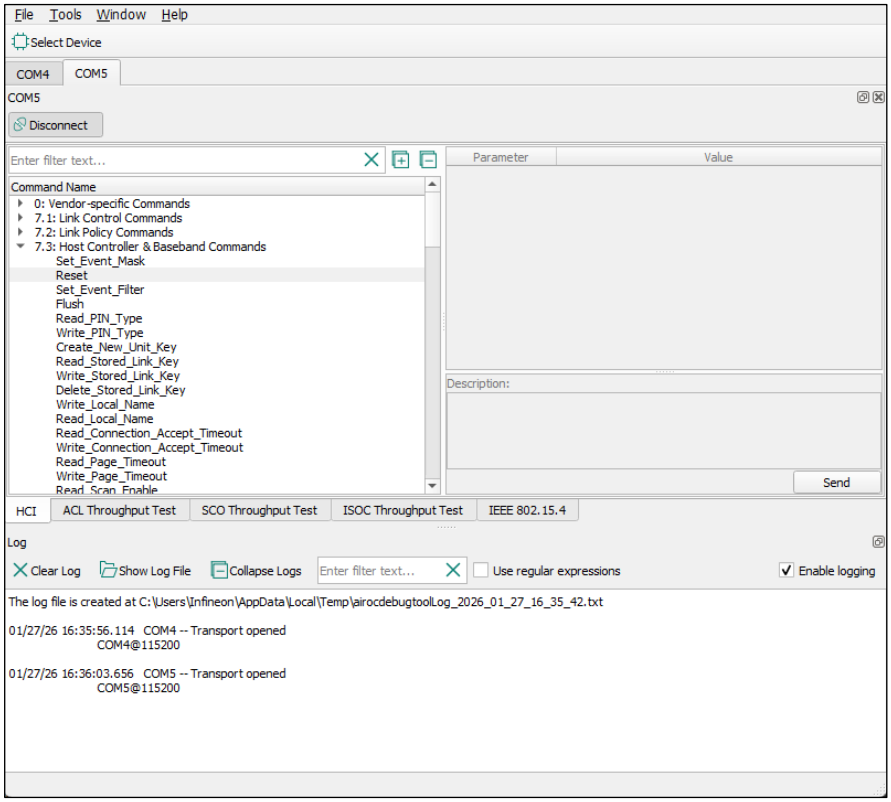
3.11 Create ACL connection between selected devices

To create an ACL connection between two AIROC™ Bluetooth® devices:

1. Select the devices.
2. Increase connection accept timeout for the peripheral device (recommended).
3. Enable scan for the peripheral device.
4. Send **Create_Connection** to the central device.
5. Send **Accept_Connection_Request** to the peripheral device.

3.11.1 Select devices

Select the devices before you create an ACL connection between two devices: connect two devices to your computer and refer to the Select a device section.



When the devices are selected, it is recommended to send the **Read_BD_ADDR** command to both devices. This will auto-populate the device address command parameters on the next steps of the process.

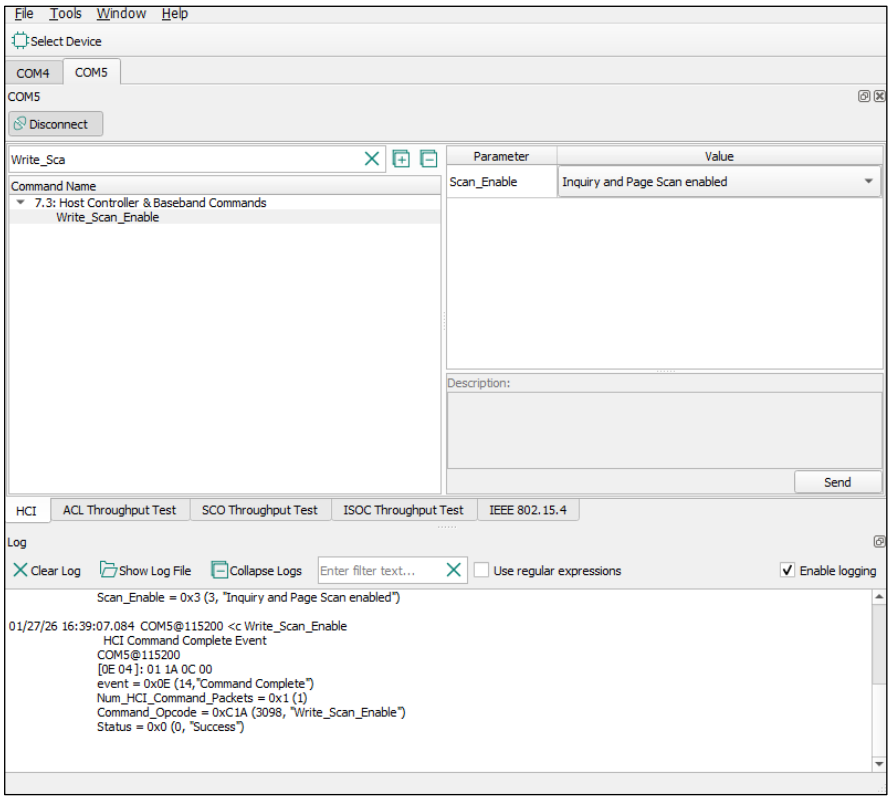
3.11.2 Increase connection accept timeout for the peripheral device

Before you create a connection between the selected devices, we recommend increasing the connection accept timeout for the peripheral device: refer to the Increase connection accept timeout for a device section.

3.11.3 Enable scan for the peripheral device

To create a connection between the selected devices, enable scan for the peripheral device:

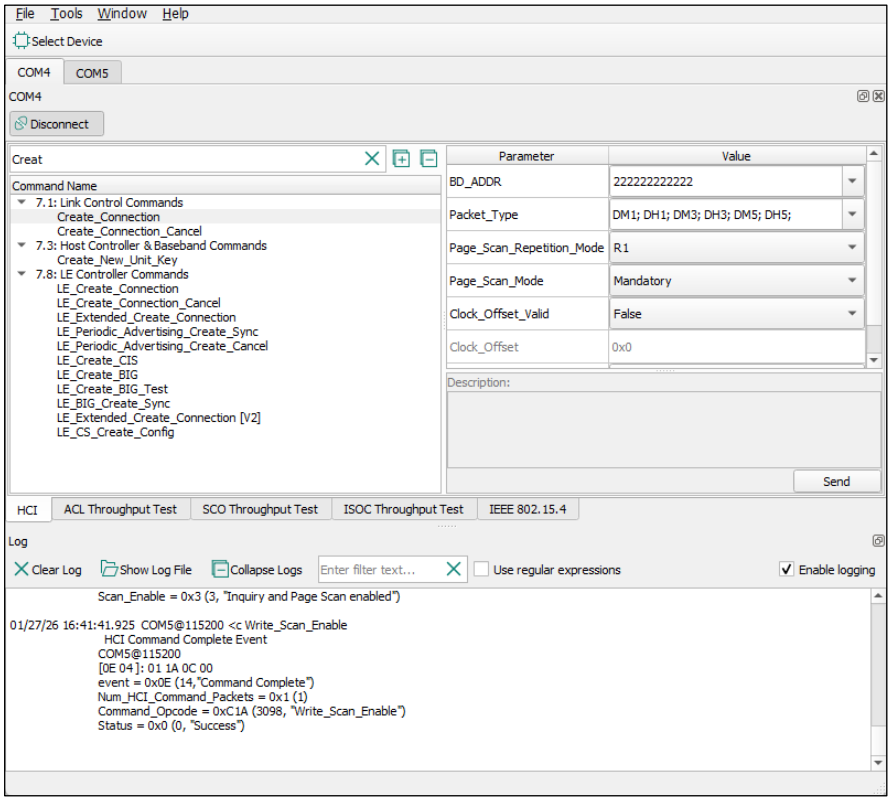
1. Select the **HCI** tab of the peripheral device.
2. Select the **Write_Scan_Enable** command.
3. Set the **Scan_Enable** parameter to **Inquiry and Page Scan enabled**.
4. Click **Send**.



3.11.4 Send Create_Connection to central device

When scan for the peripheral device is enabled, to create a connection between the selected devices, send the **Create_Connection** command to the central device:

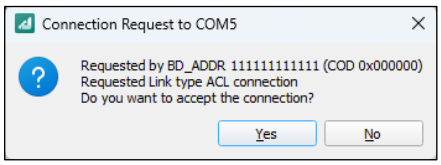
1. Select the **HCI** tab of the central device.
2. Select the **Create_Connection** command.
3. Set the command parameters.
4. Click **Send**.



*Note: Set the **BD_ADDR** parameter of the **Create_Connection** command to the address of the peripheral device. To get the peripheral device address, use the **Read_BD_ADDR** command.*

3.11.5 Send Accept_Connection_Request to peripheral device

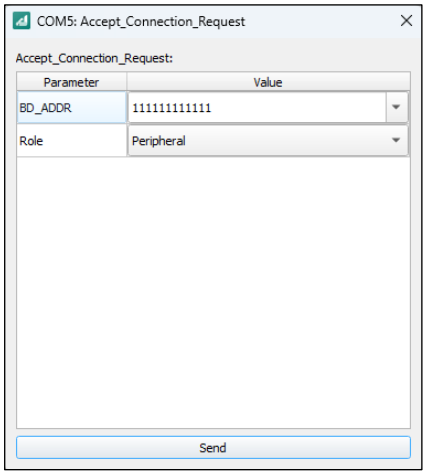
When the peripheral device gets a connection request from the central device, a dialog opens to ask if you want to accept the request.



Note: Some AIROC™ Bluetooth® devices may need antennas. If the connection request dialog does not open, check if the devices need antennas and, if yes, check if the antennas are connected to the devices.

GUI application

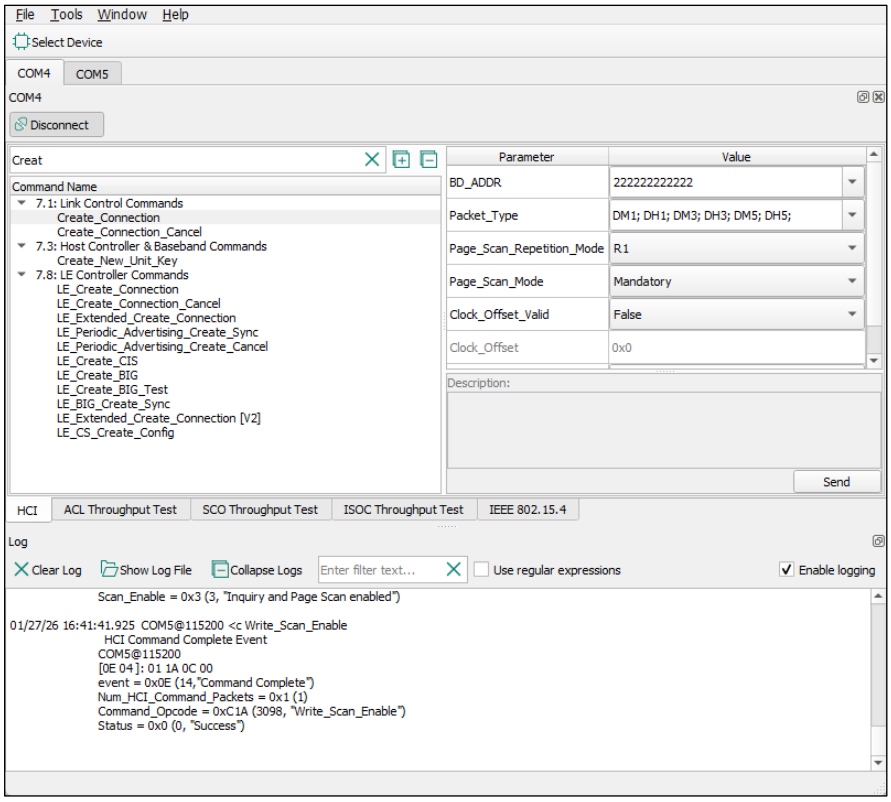
Click **Yes**. The next dialog opens, set the parameters of the **Accept_Connection_Request** command and click **Send**. The **Accept_Connection_Request** command will be sent to the peripheral device within the connection accept timeout.



Note: *Set the **BD_ADDR** parameter of the **Accept_Connection_Request** command to the address of the central device. To get the central device address, use the **Read_BD_ADDR** command.*

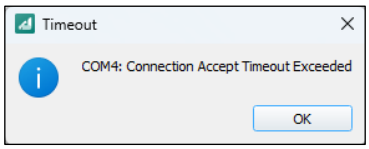
If the connection is successful, the **Connection Complete** event is generated for both devices. The event information is printed on the log window. The **Connection_Handle** parameter of the **Connection Complete** event is the handle of the created connection.

Note: *The connection handles for the central device and the peripheral device may have different values.*



Note: *When a connection request is received, if you click **No** to reject the connection request, the **Reject_Connection_Request** command will be sent to the peripheral device within the connection accept timeout.*

*If you do not send the **Accept_Connection_Request** or **Reject_Connection_Request** command within the connection accept timeout, the **Connection Complete** event with **Connection Timeout Exceeded** status is generated for both devices.*



*Sending the **Accept_Connection_Request** or **Reject_Connection_Request** command after the timeout exceeds will result in the **Invalid HCI Command Parameters** status.*

3.12 Create an LE connection between selected devices

To create an LE connection between two AIROC™ Bluetooth® devices:

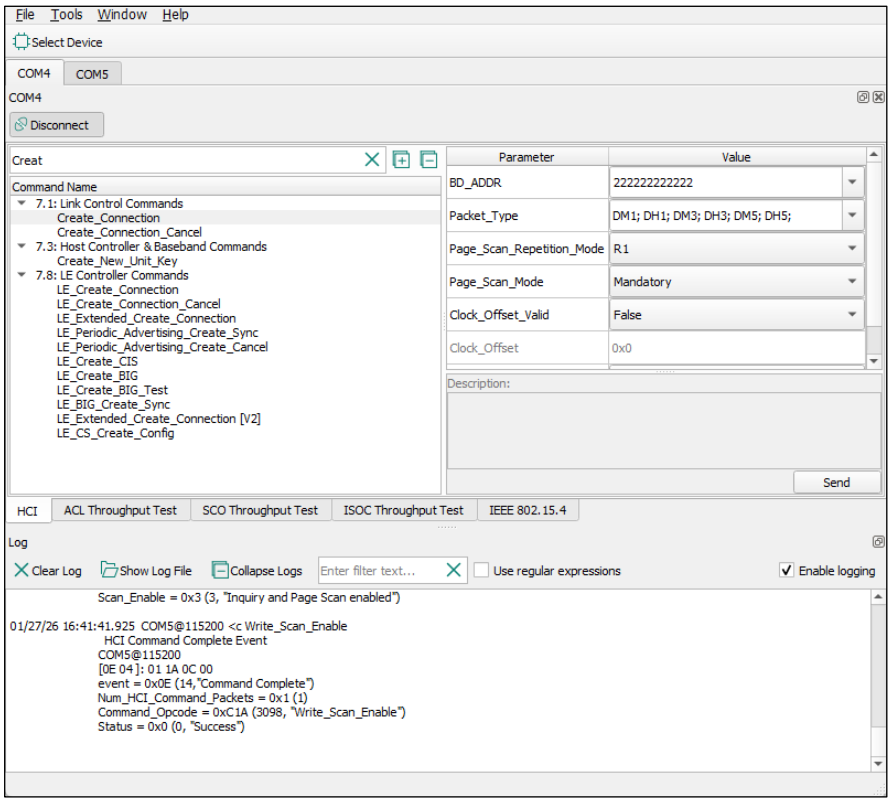
1. Select the devices.
2. Enable advertising on the peripheral device.
3. Send **LE_Create_Connection** to the central device.

3.12.1 Select devices

Select the devices before you create an LE connection between two devices: connect two devices to your computer and refer to the Select a device section.

3.12.2 Enable LE ISO Host Feature on both devices

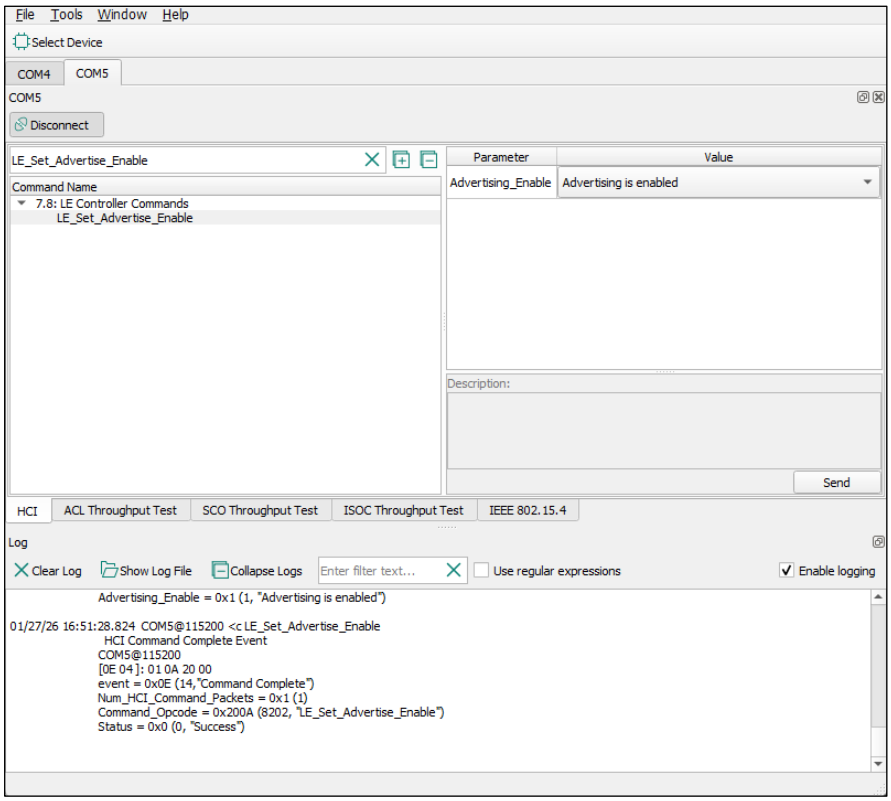
1. Select the **HCI** tab of the both devices.
2. Select the **LE_ISO_Set_Host_Feature** command.
3. Set the **Bit_Number** parameter to **0x20**.
4. Set the **Bit_Value** parameter to **Host feature is enabled**.
5. Click **Send**.



3.12.3 Enable advertising on peripheral device

To create an LE connection between the selected devices, enable advertising for the peripheral device:

1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Set_Advertise_Enable** command.
3. Set the **Advertising_Enable** parameter to **Advertising is enabled**.
4. Click **Send**.

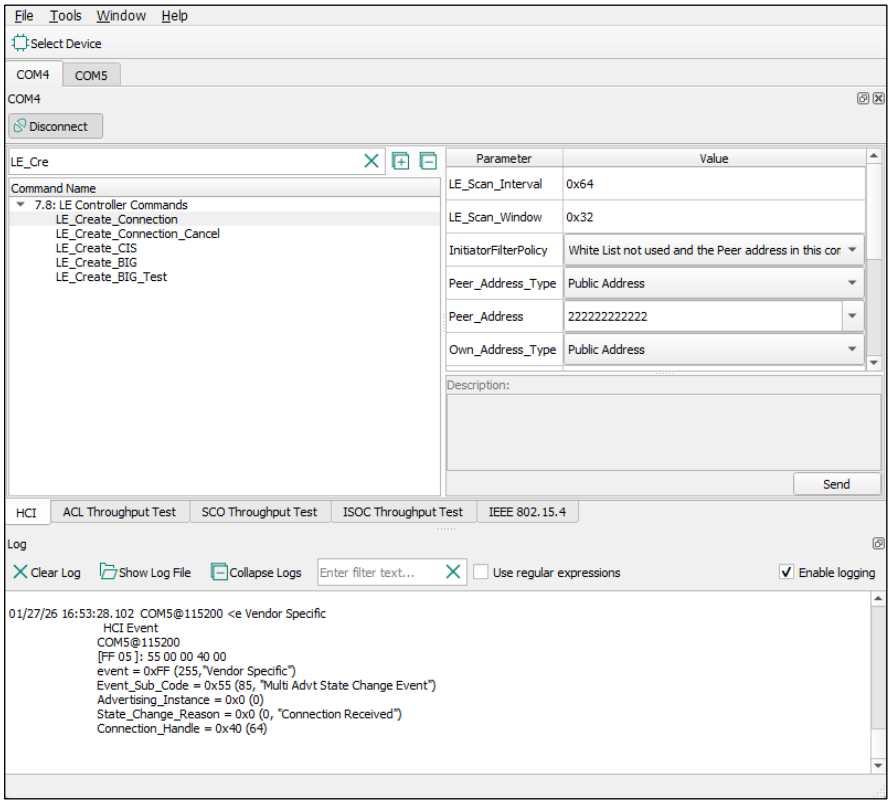


Note: Enable the advertising on the peripheral device every time when you create an LE connection.

3.12.4 Send LE_Create_Connection to central device

The advertising for the peripheral device is enabled – create an LE connection between the selected devices and send the **LE_Create_Connection** command to the central device:

1. Select the **HCI** tab of the central device.
2. Select the **LE_Create_Connection** command.
3. Set the command parameters.
4. Click **Send**.



Note: *Set the **Peer_Address** parameter of the **Create_Connection** command to the address of the peripheral device. To get the address of the peripheral device, use the **Read_BD_ADDR** command.*

*To make the LE connection more stable, you may want to increase the value of the **Supervision_Timeout** parameter. The maximum value of the **Supervision_Timeout** parameter is 0xC80. We recommend the 0x3FE value.*

If the connection is created successfully, the **LE Connection Complete Event** event is generated for both devices. Event information is printed in the log window. The **Connection_Handle** parameter of the **LE Connection Complete Event** is the handle of the created connection.

Note: *The connection handle for the central device and the connection handle for the peripheral device may have different values.*

3.13 ACL throughput tests

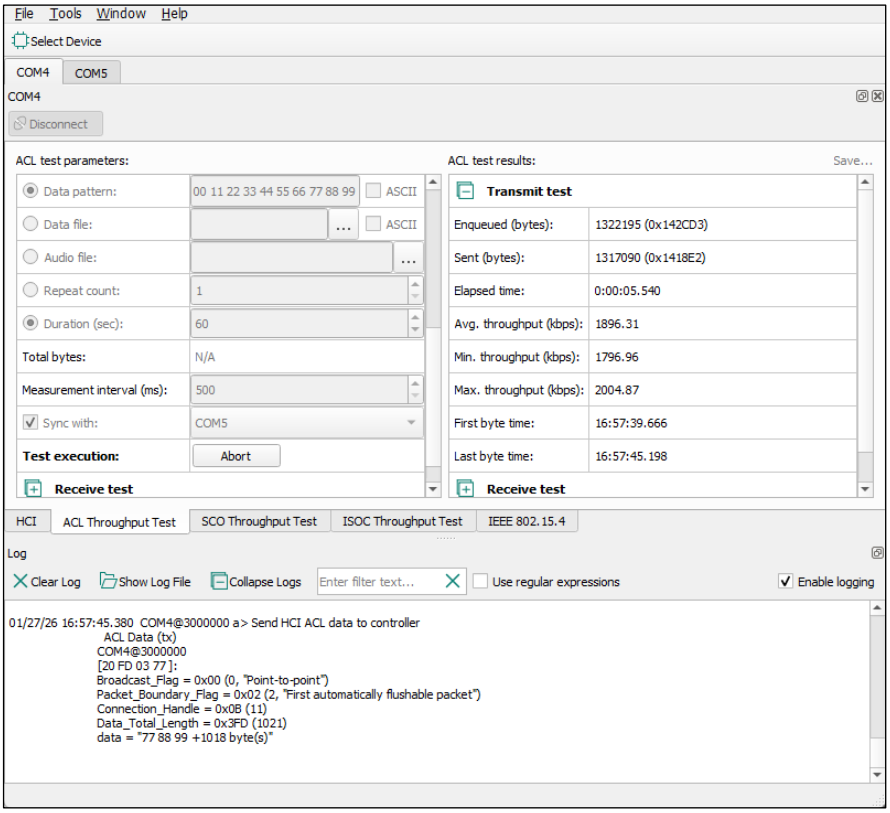
When a connection between the selected devices is created, you can run ACL throughput tests to check how fast ACL data is transferred between the devices.

Note: To run ACL throughput tests for UART devices, we recommend you set the baud rate for both devices to 3000000 bps.

3.13.1 ACL transmit test

To check how fast ACL data is transmitted by a Bluetooth® controller, run the ACL transmit test:

1. Select the **ACL Throughput Test** tab.
2. Expand the **Transmit test** section of the **ACL test parameters** column.
3. Set the test parameters.
4. Click **Start test**.



The following table lists parameters of the ACL transmit test.

Parameter	Description	Values
Connection handle	The connection handle of the connection between the devices.	A valid connection handle.
Connection type	Defines the type of the connection.	Auto-detect Classic LE
Operation type	Defines the operation type.	Run throughput test Send ACL data

GUI application

Parameter	Description	Values
Mode	Defines the data transfer mode.	POINT_TO_POINT ACTIVE_PERIPHERAL_BROADCAST
Send on first receive	If checked, data transmission starts only when the first byte is received from another device. First, start he ACL receive test for the same device.	Checked or unchecked
Packet boundary flag	Defines the packet boundary flag. Enabled if **Operation type** is set to **Send ACL data**.	START_NON_FLUSHABLE_PACKET CONTINUATION_PACKET START_OF_PACKET
Fixed write size (bytes)	Defines the size of data blocks. If not set, the size of data blocks is defined by the size of the ACL buffer.	Integer values between 1 and 65535.
Isochronous interval (ms)	Defines the interval between data blocks (in milliseconds). If not set, the **Number of Completed Packets** event is used to define when a data block can be sent.	Integer values between 1 and 65535.
Data pattern	Sets the data to be transmitted.	If ASCII is not checked, a space-separated sequence of bytes. If ASCII is checked, a valid ASCII string.
Data file	The data file from which the data to transmit will be read.	The absolute path to an existing file.
Audio file	A WAV audio file from which read the audio data to transmit will be read. The tool will encode the audio data using the SBC codec before transmission.	The absolute path to an existing file.
Repeat count	Defines how many times the data pattern, data file or audio file will be transmitted.	Integer values between 1 and 2147483647.
Duration (sec)	The expected minimum duration of the test (in seconds). The actual duration of the test can be longer.	Integer values between 1 and 2147483647.
Total bytes	The total number of bytes to be transmitted (read-only).	Integer values, or N/A if the total number of bytes cannot be defined.
Measurement interval (ms)	Defines the time interval (in milliseconds) used to calculate the throughput. Any instant throughput is calculated as a number of bytes sent during the measurement interval, divided by the time interval. Throughput values can be calculated only when the duration of the test is longer than the measurement interval.	Integer values between 500 and 2147483647.
Sync with	Allows to synchronize the transmit test and the receive test for another device. If checked, the receive test for the other device is started automatically before the transmit test starts. The transmit and receive tests parameters are synchronized.	A valid port name of the selected device.

To check the test results, expand the **Transmit test** section of the **ACL test results** column. The following table lists the test result parameters.

GUI application

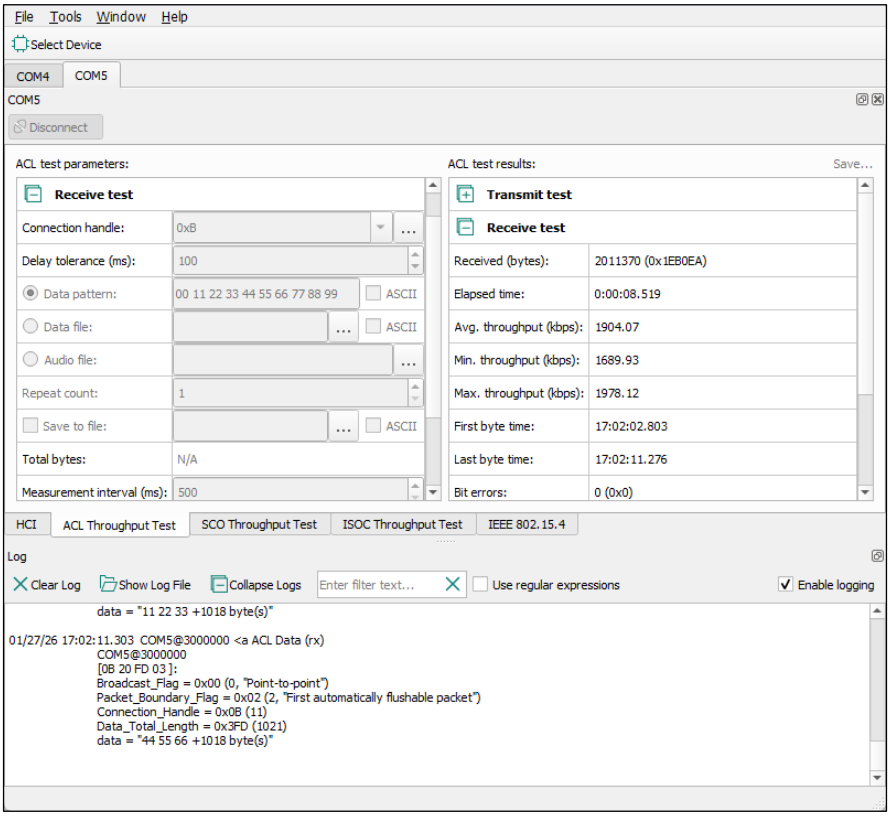
Parameter	Description
Enqueued (bytes)	The number of bytes sent to the Bluetooth® controller.
Sent (bytes)	The number of bytes completed (transmitted or flushed) by the Bluetooth® controller.
Elapsed time	The test duration.
Avg. throughput (kbps)	The average transmit throughput (in kilobytes per second). Transmit throughput is calculated as a number of bytes completed (transmitted or flushed) by the controller within the measurement time interval, divided by the time interval.
Min. throughput (kbps)	The minimum transmit throughput (in kilobytes per second).
Max. throughput (kbps)	The maximum transmit throughput (in kilobytes per second).
First byte time	Indicates when the first byte was sent to the Bluetooth® controller.
Last byte time	Indicates when the last byte was sent to the Bluetooth® controller.

*Note: To save the test results to a text file, click the **Save...** link located above the test results section.*

3.13.2 ACL receive test

The ACL receive test checks how fast ACL data is received by a Bluetooth® controller. To run the ACL receive test:

1. Select the **ACL Throughput Test** tab.
2. Expand the **Receive test** section of the **ACL test parameters** column.
3. Set test parameters.
4. Click **Start test**.



GUI application

The following table lists parameters of the ACL receive test.

Parameter	Description	Values
Connection handle	The connection handle of the connection between the devices.	A valid connection handle.
Delay tolerance (ms)	Acceptable time interval (in milliseconds) between the time when data is sent to the transmitting Bluetooth® controller and the time when the data is received by the receiving Bluetooth® controller. The measured intervals can be larger than real values.	Integer values between 1 and 65535.
Data pattern	The data that is expected to be received.	If ASCII is not checked, a space-separated sequence of bytes. If ASCII is checked, a valid ASCII string.
Data file	A data file with the data that is expected to be received.	The absolute path to an existing file.
Audio file	A WAV audio file with the audio data that is expected to be received	The absolute path to an existing file.
Repeat count	Defines how many times the data pattern, data file or audio file is expected to be received.	Integer values between 1 and 2147483647.
Save to file	Defines if the received data will be saved to a file. If **Audio file** is selected, the tool will decode the received data using the SBC codec.	The absolute path to the file where the received data will be saved.
Total bytes	The total number of bytes that are expected to be received (read-only).	Integer values, or N/A if the total number of bytes cannot be defined.
Measurement interval (ms)	Defines time interval (in milliseconds) which is used to calculate throughput. At any moment of time, the instant throughput is calculated as the number of bytes, that were received during the measurement interval, divided by the time interval. Throughput values can be calculated only when the duration of the test is longer than the measurement interval.	Integer values between 500 and 2147483647
Sync with	Allows to synchronize the receive test and the transmit test for another device. If checked, the transmit test for another device is started automatically after the receive test starts. Parameters of the receive and transmit tests are synchronized.	A valid port name of the selected device.

To check test results, expand the **Receive test** section of the **ACL test results** column. The following table lists test result parameters.

Parameter	Description
Received (bytes)	The number of bytes that were received.
Elapsed time	Duration of the test.
Avg. throughput (kbps)	The average receive throughput (in kilobytes per second). Receive throughput is calculated as the number of bytes, received within the measurement interval, divided by the time interval.
Min. throughput (kbps)	The minimum receive throughput (in kilobytes per second).
Max. throughput (kbps)	The maximum receive throughput (in kilobytes per second).

GUI application

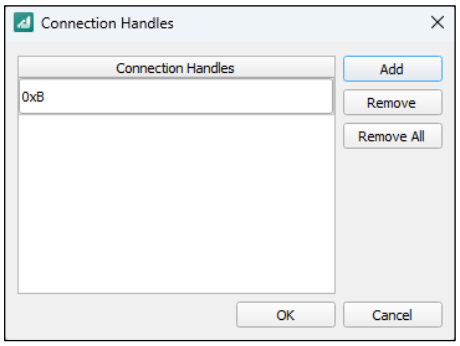
Parameter	Description
First byte time	Indicates when the first byte was received.
Last byte time	Indicates when the last byte was received.
Bit errors	The total number of bits that were received incorrectly.
Delayed packets	The total number of packets that were delayed by more than the value of delay tolerance. The measured intervals can be greater than real values.
Average delay (ms)	The average delay of a packet (in milliseconds).
Maximum delay (ms)	The maximum delay of a packet (in milliseconds).

*Note: To save test results to a test file, click the **Save…** link located above the test results section.*

The receive test stops automatically if the expected number of bytes is known and received. Otherwise, you need to stop the receive test manually by clicking **Abort**.

3.13.3 Simultaneous ACL throughput tests

To run ACL transmit or receive throughput tests for multiple connections, in test settings, click the "**...**" button next to **Connection handle** to open the Connection Handles dialog.



In the Connection Handles dialog, enter the handles of connections for which you want to run the test, and click **OK**. In test settings, use the **Connection handle** drop-down list to switch between the connection handles. When you click **Start test**, the test is started for all connection handles.

Note: When you run a throughput test for multiple connections, you cannot synchronize the receive and transmit tests.

3.14 Create SCO connection between selected devices

To create an SCO connection between two AIROC™ Bluetooth® devices:

1. Select the devices.
2. Create an ACL connection between the selected devices.
3. Increase connection accept timeout for the peripheral device (recommended).
4. Send **Setup_Synchronous_Connection** to the central device.
5. Send **Accept_Synchronous_Connection_Request** to the peripheral device.

3.14.1 Select devices

Select the devices before you create an SCO connection between two devices: connect two devices to your computer and refer to the Select a device section.

3.14.2 Create ACL connection between selected devices

To create an SCO connection between the selected devices, refer to the Create an ACL connection between selected devices section.

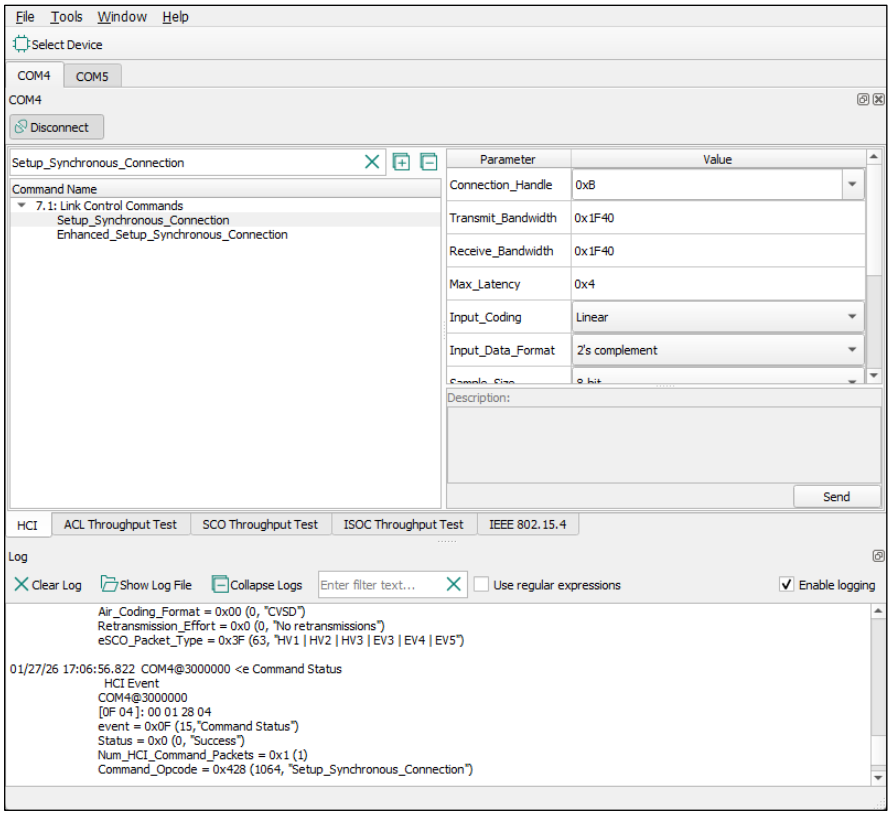
3.14.3 Increase connection accept timeout for peripheral device

Before you send the **Setup_Synchronous_Connection** command to the central device, refer to the Increase connection accept timeout for a device section.

3.14.4 Send Setup_Synchronous_Connection to central device

When the ACL connection is created, to create an SCO connection, send the **Setup_Synchronous_Connection** command to the central device:

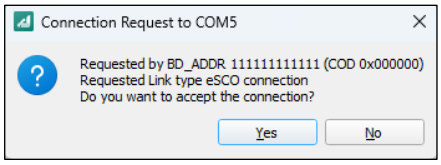
1. Select the **HCI** tab of the central device.
2. Select the **Setup_Synchronous_Connection** command.
3. Set command parameters.
4. Click **Send**.



Note: *Set the **Connection_Handle** parameter of the **Setup_Synchronous_Connection** command to the connection handle of the ACL connection. The other parameter values shall correspond to the audio data to be transmitted.*

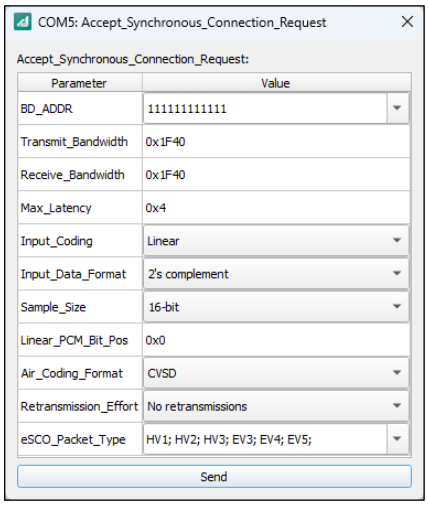
3.14.5 Send Accept_Synchronous_Connection_Request to peripheral device

When the peripheral device gets the synchronous connection request from the central device, a dialog opens to ask if you want to accept the request.



To accept the request, click **Yes**.

On the next dialog, set the **Accept_Synchronous_Connection_Request** command parameters and click **Send**. The **Accept_Synchronous_Connection_Request** command will be sent to the peripheral device within the connection accept timeout.

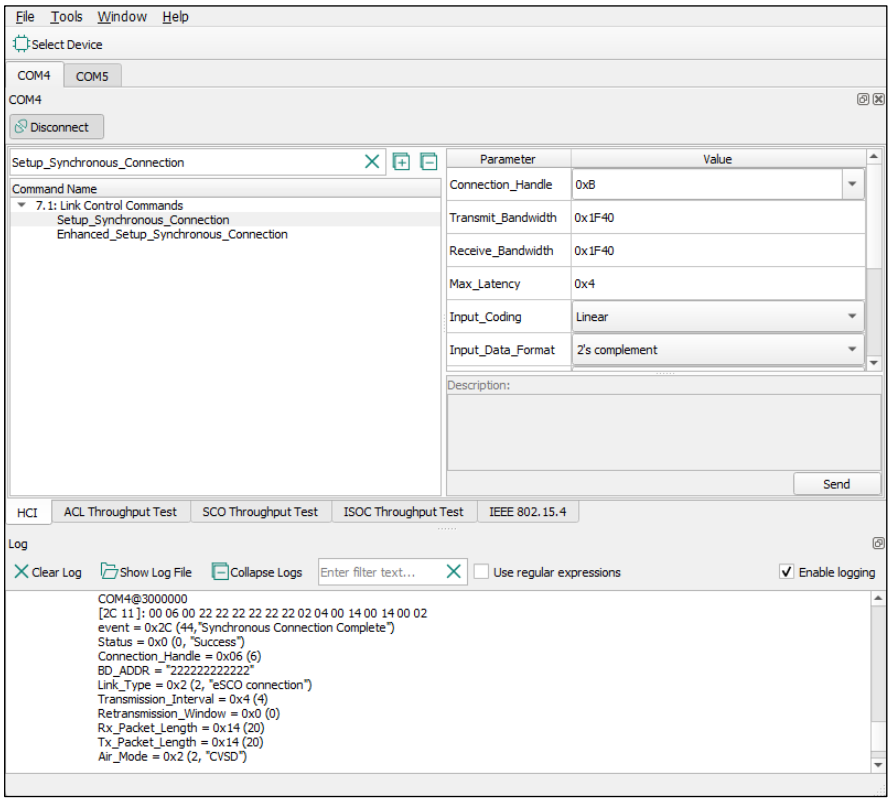


*Note: The parameters of the **Accept_Synchronous_Connection_Request** command shall match the parameters of the synchronous connection between the devices. Set the **BD_ADDR** parameter of the **Accept_Synchronous_Connection_Request** command to the address of the central device. To get the address, use the **Read_BD_ADDR** command.*

If the connection is created successfully, the **Synchronous Connection Complete** event is generated. The event information is printed in the log window. The **Connection_Handle** parameter of the **Synchronous Connection Complete** event is the connection handle of the created connection.

Note: The connection handles for the central device and peripheral device may have different values.

GUI application



Note: *When the synchronous connection accept request is received, if you click **No** to reject the connection request, the **Reject_Synchronous_Connection_Request** command will be sent to the peripheral device within the connection accept timeout.*

3.15 SCO throughput tests

When the SCO connection between the selected devices is created, you can run SCO throughput tests to check how fast SCO data is transferred between the devices.

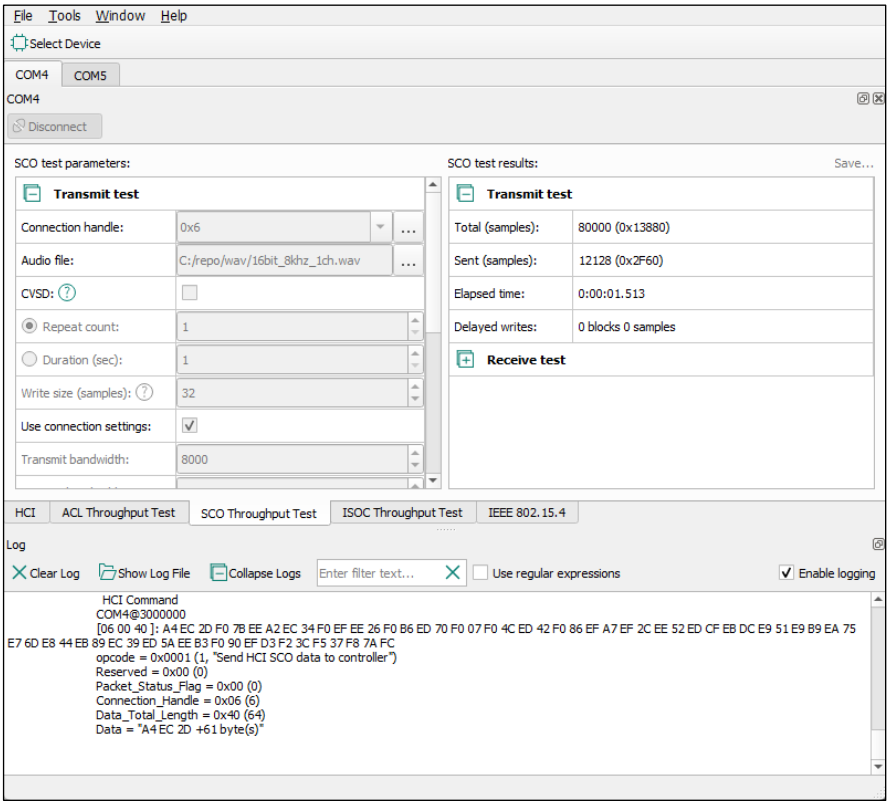
Note: *To run SCO throughput tests for UART devices, we recommend you set the baud rate for both devices to 3000000 bps.*

*For some AIROC™ Bluetooth® devices, before you run SCO throughput tests, you may need to send the **Write_SCO_PCM_Int_Param** command.*

3.15.1 SCO transmit test

The SCO transmit test checks how fast SCO data is transmitted by a Bluetooth® controller. To run the SCO transmit test:

1. Select the **SCO Throughput Test** tab.
2. Expand the **Transmit test** section of the **SCO test parameters** column.
3. Set test parameters.
4. Click **Start test**.



The following table lists the SCO transmit test parameters.

Parameter	Description	Values
Connection handle	Connection handle of the SCO connection.	A valid connection handle.
Audio file	A WAV audio file from which the audio data to transmit will be read.	The absolute path to an existing WAV file.
CVSD	Enables or disables CVSD encoding. If enabled, the tool will encode audio data using the CVSD codec and pass the encoded data to the controller. Please make sure that the audio data is not encoded on the controller to avoid double encoding. Enabled only if **Audio file** is selected.	Checked or unchecked.
Repeat count	Defines how many times the audio file will be sent.	Integer values between 1 and 2147483647.
Duration (sec)	Expected minimum duration of the test (in seconds). The actual duration of the test can be longer.	Integer values between 1 and 2147483647.

GUI application

Parameter	Description	Values
Write size	Defines how many samples in a transaction slot can be filled. In case of USB transport, available only on Linux.	Integer values between 1 and 127.
Use connection settings	Defines whether to use the settings that were set when creating the SCO connection or define the connection settings manually.	Checked or unchecked.
Transmit bandwidth	The value of transmit bandwidth. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Receive bandwidth	The value of receive bandwidth. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Sample size	Defines sample size. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Input coding	Defines input coding. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Input data format	Defines input data format. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Air coding	Defines air coding. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
PCM	The value of PCM. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Sync with	Allows to synchronize the transmit test and the receive test for another device. If checked, the receive test for another device is started automatically before the transmit test starts. Parameters of the receive and transmit tests are synchronized.	A valid port name of the selected device. First, establish a SCO connection between the devices.

To check transmit test results, expand the Transmit test section of the SCO test results column. The following table lists test result parameters.

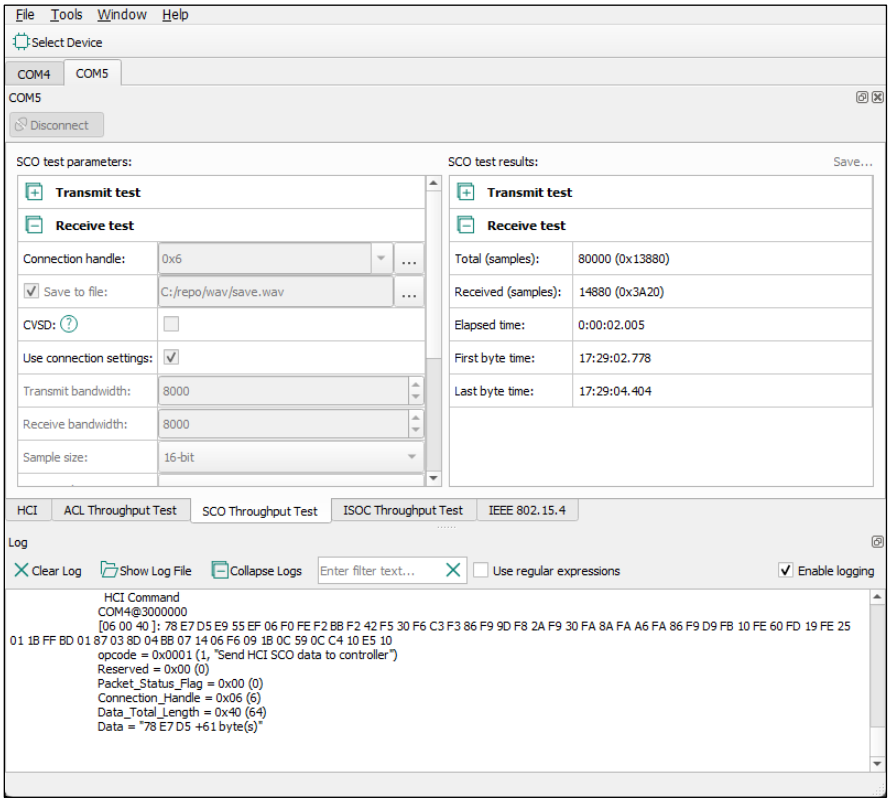
Parameter	Description
Total (samples)	The total number of samples to be transmitted, or N/A if the total number of samples cannot be defined.
Sent (samples)	The number of samples that were transmitted.
Elapsed time	Duration of the test.
Delayed writes	The number of blocks and samples that were delayed.

*Note: To save test results to a text file, click the **Save...** link located above the test results section.*

3.15.2 SCO receive test

The SCO receive test checks how fast SCO data is received by a Bluetooth® controller. To run the SCO receive test:

1. Select the **SCO Throughput Test** tab.
2. Expand the **Receive test** section of the **SCO test parameters** column.
3. Set test parameters.
4. Click **Start test**.



The following table lists parameters of the SCO receive test.

Parameter	Description	Values
Connection handle	Connection handle of the SCO connection.	A valid connection handle.
Save to file	Defines if the received data will be saved to a file.	The absolute path to the WAV file to save the received data to.
CVSD	Enables or disables CVSD decoding. If enabled, the tool will decode the received audio data using the CVSD codec.	Checked or unchecked.
Use connection settings	Defines whether to use the settings that were set when creating the SCO connection or define the connection settings manually.	Checked or unchecked.
Transmit bandwidth	The value of transmit bandwidth. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.

GUI application

Parameter	Description	Values
Receive bandwidth	The value of receive bandwidth. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Sample size	Defines sample size. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Input coding	Defines input coding. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Input data format	Defines input data format. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Air coding	Defines air coding. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
PCM	The value of PCM. Available only if **Use connection settings** is unchecked.	Set to the correct value for the created connection.
Sync with	Allows to synchronize the receive test and the transmit test for another device. If checked, the transmit test for another device is started automatically after the receive test starts. Parameters of the receive and transmit tests are synchronized.	A valid port name of the selected device.

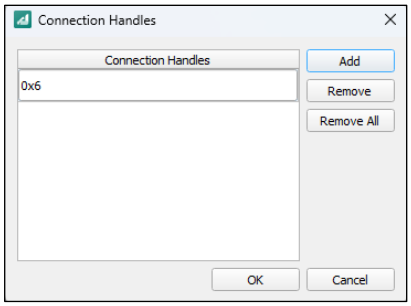
To check test results, expand the **Receive test** section of the **ACL test results** column. The following table lists test result parameters and explains their meanings.

Parameter	Description
Total (samples)	The number of samples that are expected to be received, or N/A if the total number of samples cannot be defined.
Received (samples)	The number of received samples.
Elapsed time	Duration of the test.
First byte time	Indicates when the first byte was received.
Last byte time	Indicates when the last byte was received.

Note: The SCO receive test is always stopped manually.

3.15.3 Simultaneous SCO throughput tests

To run SCO transmit or receive throughput tests for multiple connections, in test settings, click the "**...**" button next to **Connection handle** to open the "Connection Handles" dialog.



Enter the handles of connections for which you want to run the test, click **OK**. In the test settings, use the **Connection handle** drop-down list to switch between the connection handles. When you click **Start test**, the test is started for all connection handles.

Note: *When you run a throughput test for multiple connections, you cannot synchronize the receive and transmit tests.*

For a USB device, you can run an SCO throughput test for at most three connections.

For USB devices, test settings shall be the same for all connections.

3.16 Create CIS between selected devices

Note: *Isochronous channels are supported only by devices that support Bluetooth® 5.2 or later.*

To create a CIS between two devices:

1. Select the devices.
2. Enable isochronous channels for the selected devices.
3. Create an LE connection.
4. Set CIG parameters for the central device.
5. Increase connection accept timeout for the peripheral device (recommended).
6. Send **LE_Create_CIS** to the central device.
7. Send **LE_Accept_CIS_Request** to the peripheral device.

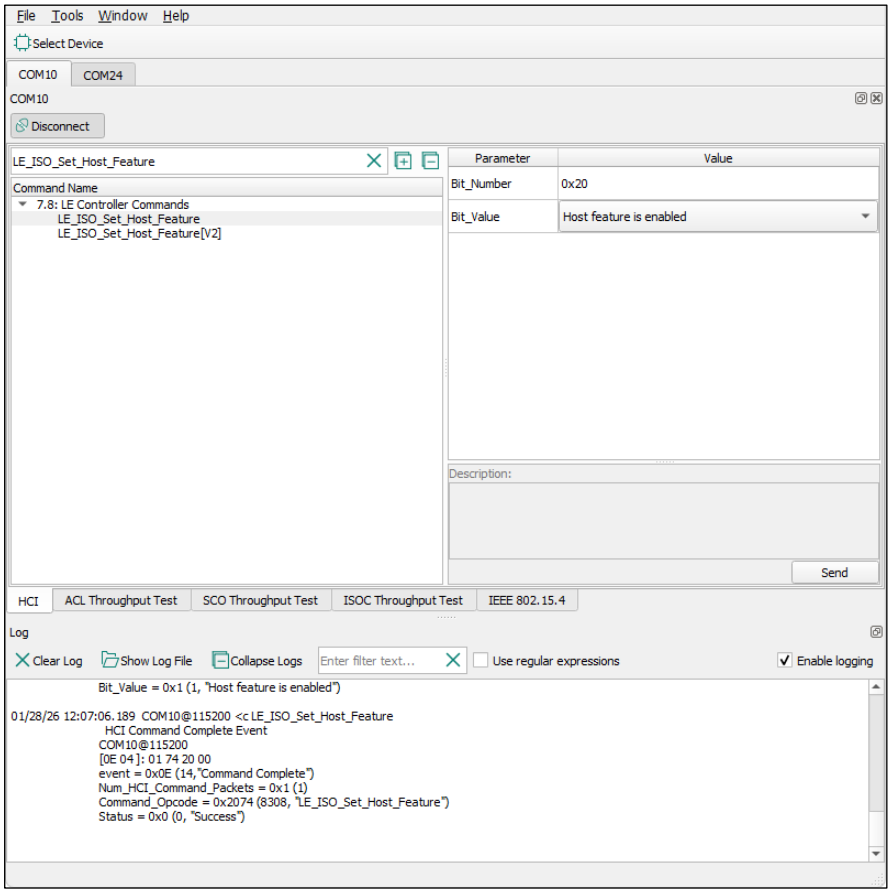
3.16.1 Select the devices

Select the devices before you create a CIS between two devices: connect two AIROC™ Bluetooth® devices to your computer and refer to the Select a device section.

3.16.2 Enable isochronous channels for the selected devices

To create a CIS between the selected devices, enable isochronous channels for the selected devices:

1. Select the **HCI** tab.
2. Select the **LE_ISO_Set_Host_Feature** command.
3. Set the **Bit_Number** command parameter to 0x20 and the **Bit_Value** command parameter to **Host feature is enabled**.
4. Click **Send**.



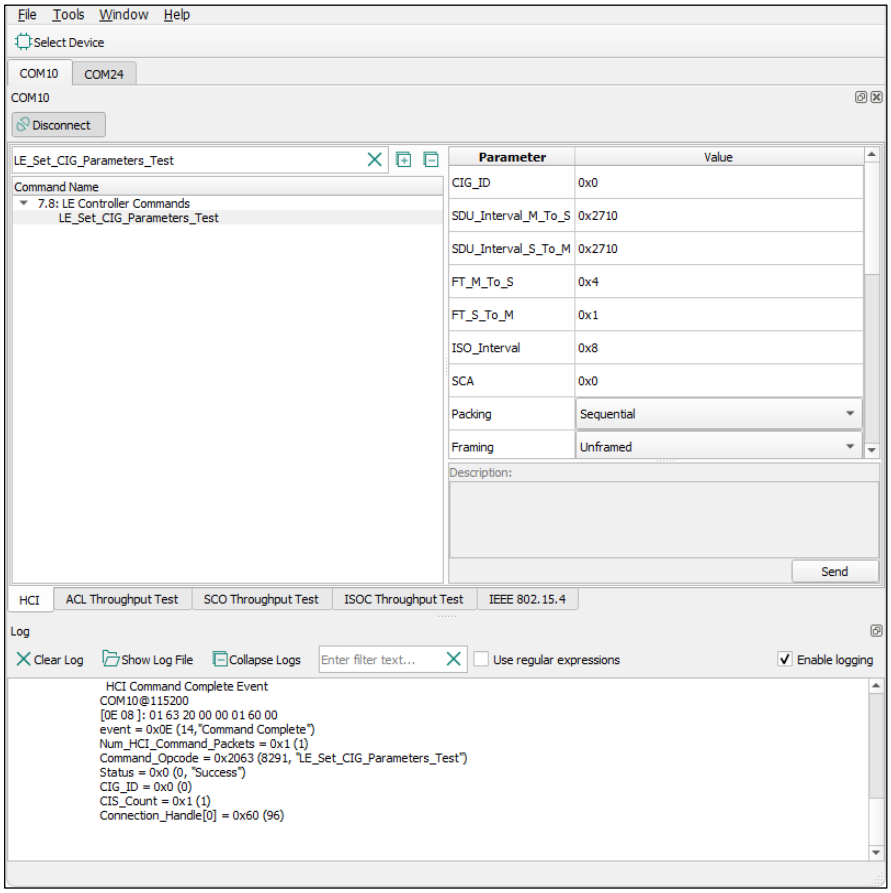
3.16.3 Create LE connection

When isochronous channels are enabled for the selected devices, refer to the Create an LE connection between selected devices section.

3.16.4 Set CIG parameters for central device

When the LE connection between the selected devices is created, to create a CIS between the selected devices, set CIG parameters for the central device:

1. Select the **HCI** tab of the central device.
2. Select the **LE_Set_CIG_Parameters** or **LE_Set_CIG_Parameters_Test** command.
3. Set the command parameters.
4. Click **Send**.



If the CIG parameters for the central device are set successfully, a **Command Complete** event is generated, and CIS connection handles are reported.

Recommended: use the following table lists parameters for the **LE_Set_CIG_Parameters_Test** command.

Parameter	Values
CIG_ID	0x0
SDU_Interval_M_To_S	0x2710
SDU_Interval_S_To_M	0x2710
FT_M_To_S	0x4
FT_S_To_M	0x1
ISO_Interval	0x8
SCA	0x0
Packing	Sequential

GUI application

Parameter	Values
Framing	Unframed
CIS_Count	0x1
CIS_ID[0]	0x0
NSE[0]	0x4
Max_SDU_M_To_S[0]	0xFB
Max_SDU_S_To_M[0]	0x0
Max_PDU_M_To_S[0]	0xFB
Max_PDU_S_To_M[0]	0x0
PHY_M_To_S[0]	the Host prefers to use the 1Ms/s receiver PHY
PHY_S_To_M[0]	the Host prefers to use the 1Ms/s receiver PHY
BN_M_To_S[0]	0x2
BN_S_To_M[0]	0x0

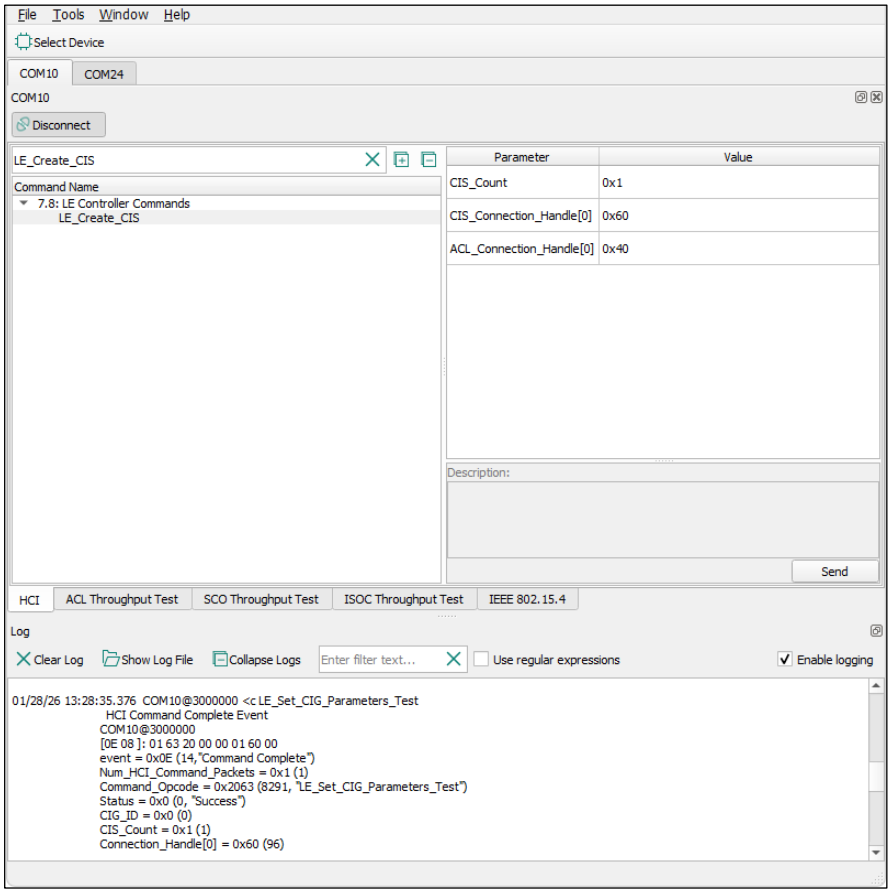
3.16.5 Increase connection accept timeout for peripheral device

Before you send the **LE_Create_CIS** command to the central device, refer to the Increase connection accept timeout for a device section.

3.16.6 Send LE_Create_CIS to central device

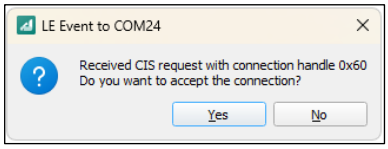
When CIG parameters for the central device are set, to create a CIS between the selected devices, send the **LE_Create_CIS** command to the central device:

1. Select the **HCI** tab of the central device.
2. Select the **LE_Create_CIS** command.
3. Set command parameters.
4. Click **Send**.



3.16.7 Send LE_Accept_CIS_Request to peripheral device

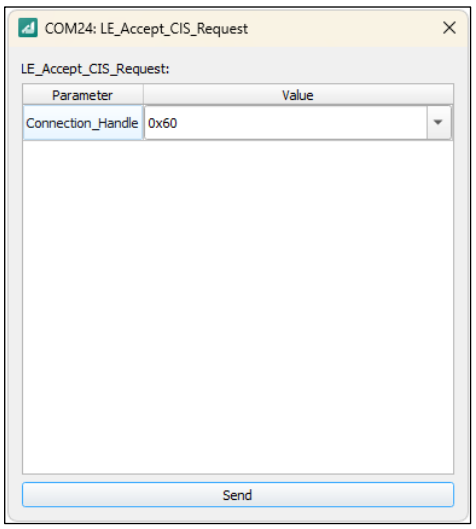
When the peripheral device gets a CIS request from the central device, a dialog opens to ask if you want to accept the request.



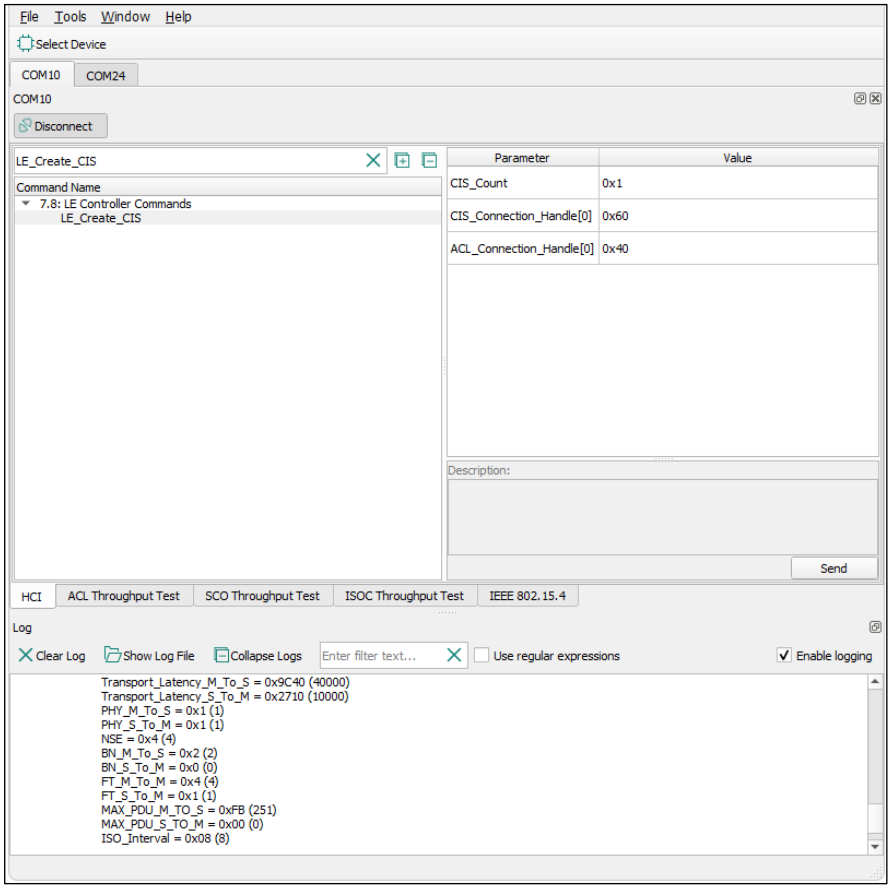
To accept the request, click **Yes**.

GUI application

On the next dialog, set the parameters of the **LE_Accept_CIS_Request** command and click **Send**. The **LE_Accept_CIS_Request** command will be sent to the peripheral device within the connection accept timeout.



If the CIS is created successfully, an **LE CIS Established** event is generated. The event information is printed on the log window.



3.17 Create BIS between selected devices

To create a BIS between two devices:

1. Select the devices.
2. Enable isochronous channels for the selected devices.
3. Configure data path for the peripheral device.
4. Set extended advertising parameters for the central device.
5. Enable extended advertising for the central device.
6. Set periodic advertising parameters for the central device.
7. Enable periodic advertising for the central device.
8. Enable **LE Create BIG Complete** event for the central device.
9. Create a BIG.
10. Disable advertising reports for the peripheral device (recommended).
11. Add the central device to the whitelist of the peripheral device.
12. Enable scanning on the peripheral device.
13. Add the central device to the periodic advertiser list of the peripheral device.
14. Enable **LE Periodic Advertising Sync Established** event for the peripheral device.
15. Synchronize the peripheral device with the periodic advertising from the central device.
16. Disable scanning on the peripheral device.
17. Enable **LE BIG Sync Established** event for the peripheral device.
18. Synchronize the peripheral device to the BIG.
19. Stop receiving the advertising packets.

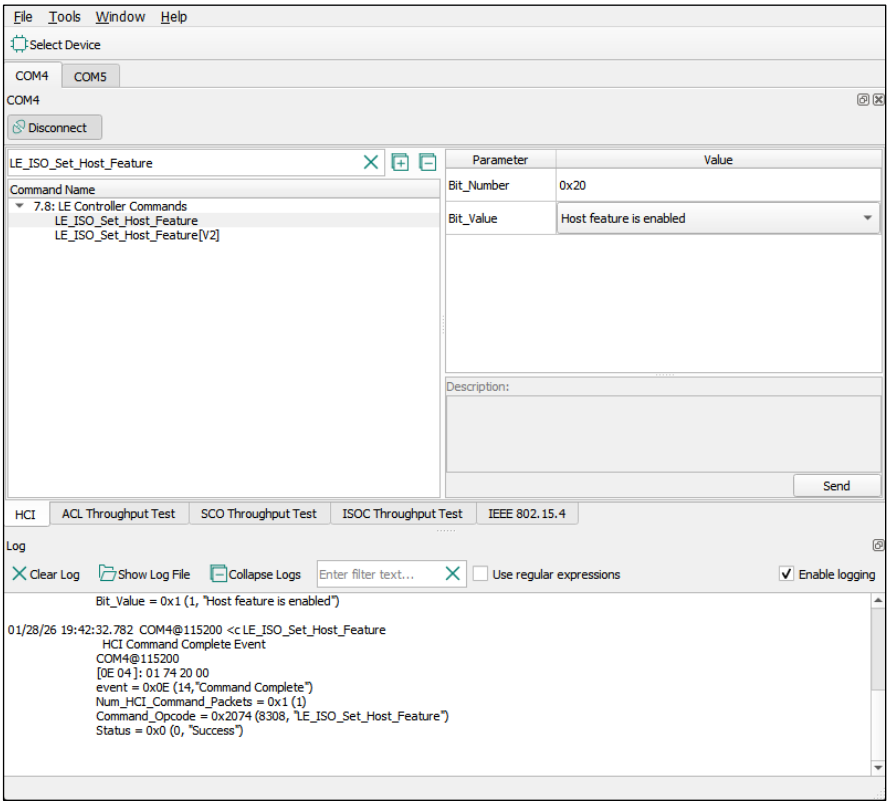
3.17.1 Select the devices

Before you create a BIS between two devices, select the devices. To select the devices, connect two AIROC™ Bluetooth® devices to your computer and refer to the Select a device section.

3.17.2 Enable isochronous channels for selected devices

To create a BIS between the selected devices, enable isochronous channels for the selected devices:

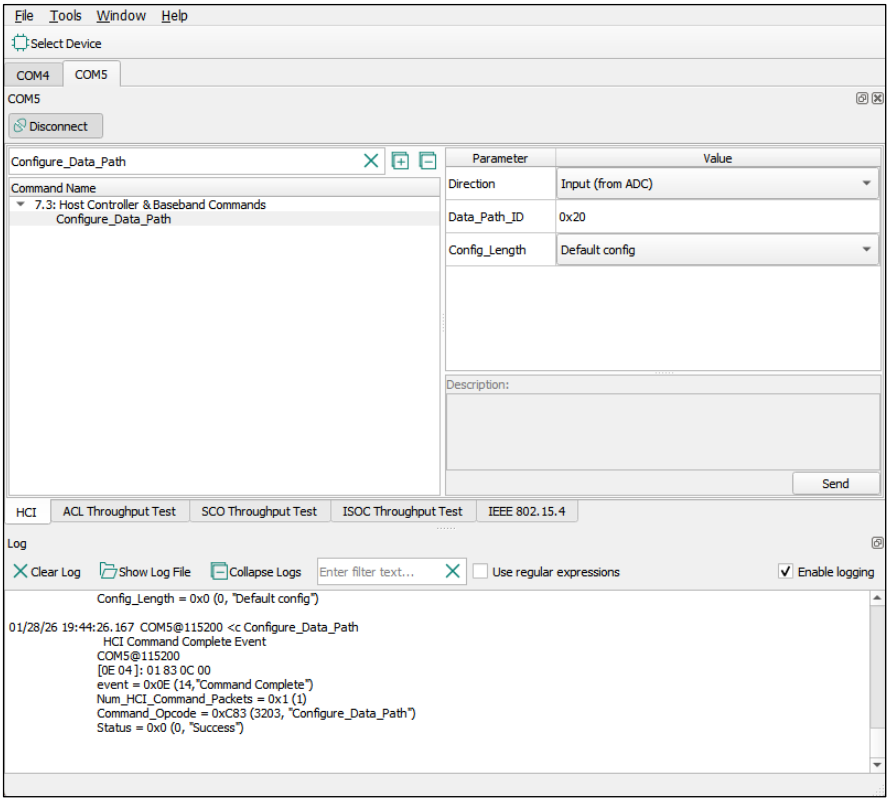
1. Select the **HCI** tab.
2. Select the **LE_ISO_Set_Host_Feature** command.
3. Set the **Bit_Number** command parameter to 0x20 and the **Bit_Value** command parameter to **Host feature is enabled**.
4. Click **Send**.



3.17.3 Configure data path for peripheral device

When isochronous channels are enabled for the selected devices, to create a BIS between the selected devices, configure the data path for the peripheral device:

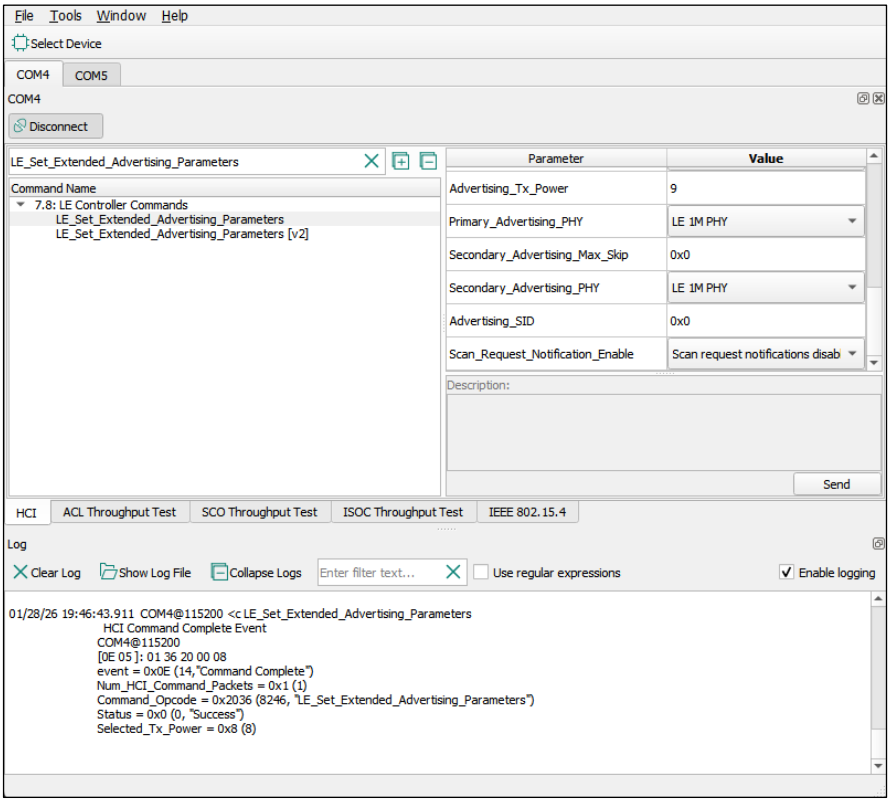
1. Select the **HCI** tab of the peripheral device.
2. Select the **Configure_Data_Path** command.
3. Set the **Data_Path_ID** command parameter to 0x20.
4. Click **Send**.
5. Set the **Data_Path_ID** command parameter to 0x21.
6. Click **Send**.



3.17.4 Set extended advertising parameters for central device

When data path for the peripheral device is configured, to create a BIS between the selected devices, set extended advertising parameters for the central device:

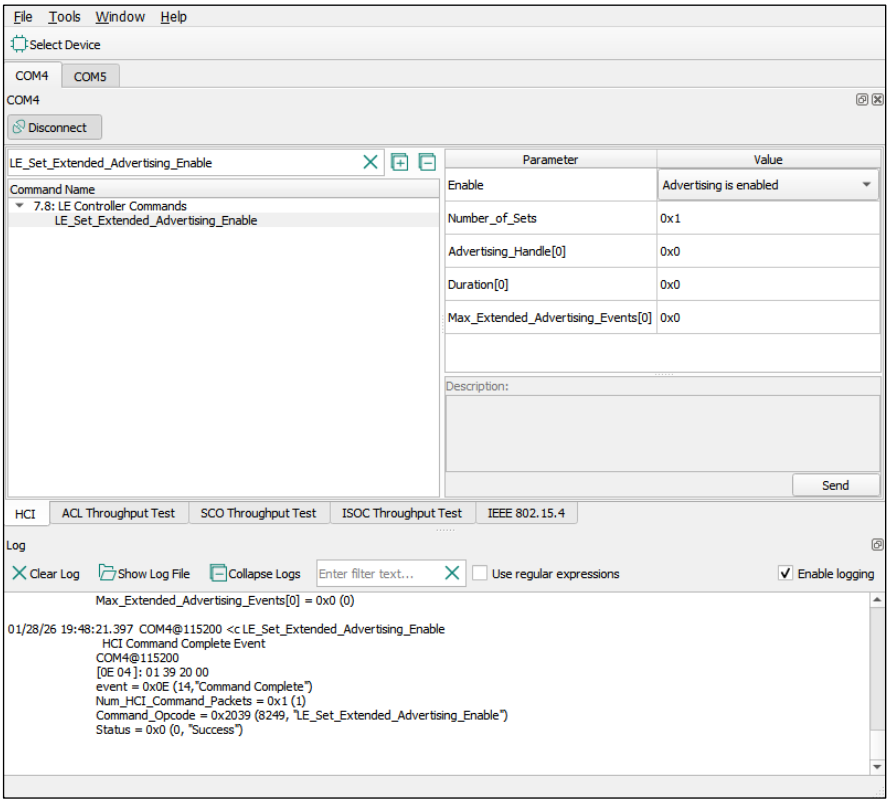
1. Select the **HCI** tab of the central device.
2. Select the **LE_Set_Extended_Advertising_Parameters** command.
3. Set the command parameters.
4. Click **Send**.



3.17.5 Enable extended advertising for central device

When extended advertising parameters for the central device are set, to create a BIS between the selected devices, enable the extended advertising for the central device:

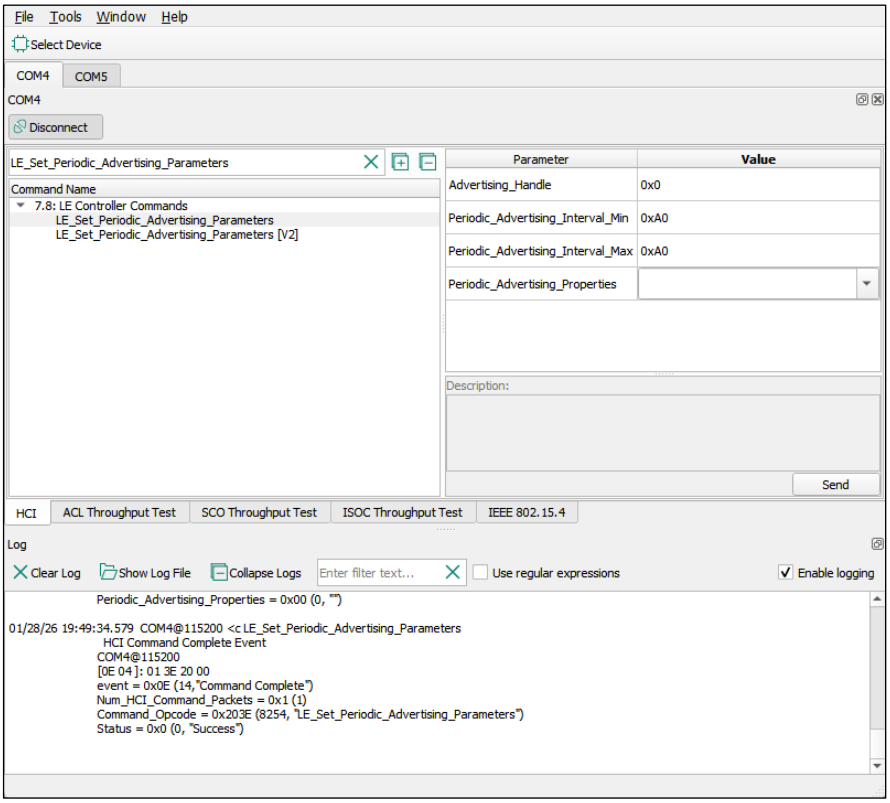
1. Select the **HCI** tab of the central device.
2. Select the **LE_Set_Extended_Advertising_Enable** command.
3. Set the **Enable** command parameter to **Advertising is enabled**.
4. Click **Send**.



3.17.6 Set periodic advertising parameters for central device

When extended advertising for the central device is enabled, to create a BIS between the selected devices, set the periodic advertising parameters for the central device:

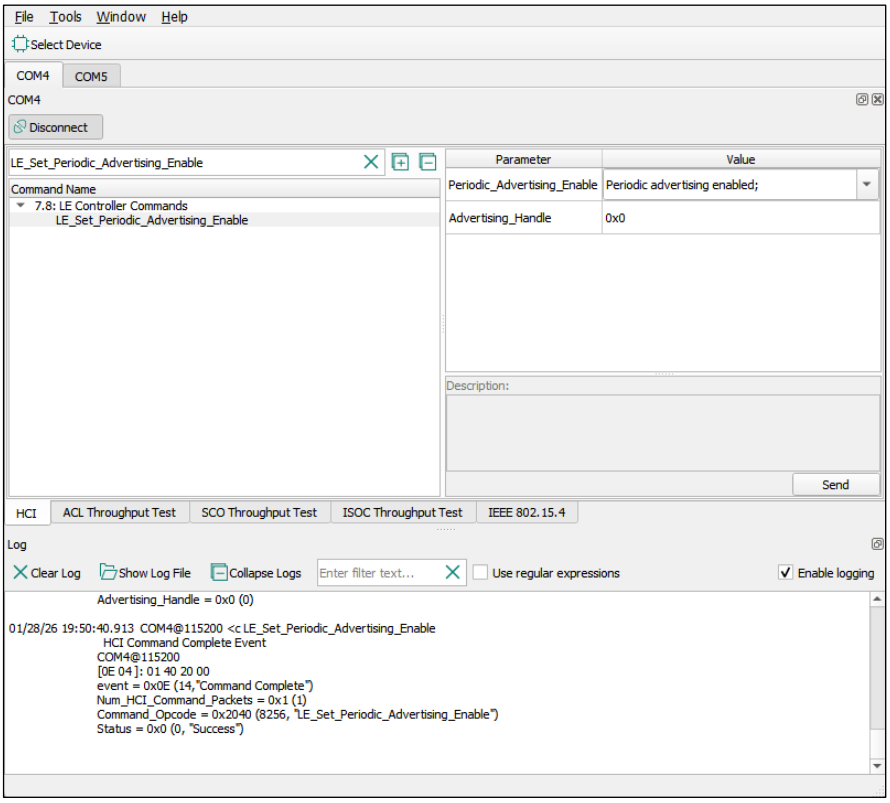
1. Select the **HCI** tab of the central device.
2. Select the **LE_Set_Periodic_Advertising_Parameters** command.
3. Set the command parameters.
4. Click **Send**.



3.17.7 Enable periodic advertising for central device

When periodic advertising parameters for the central device are set, to create a BIS between the selected devices, enable the periodic advertising for the central device:

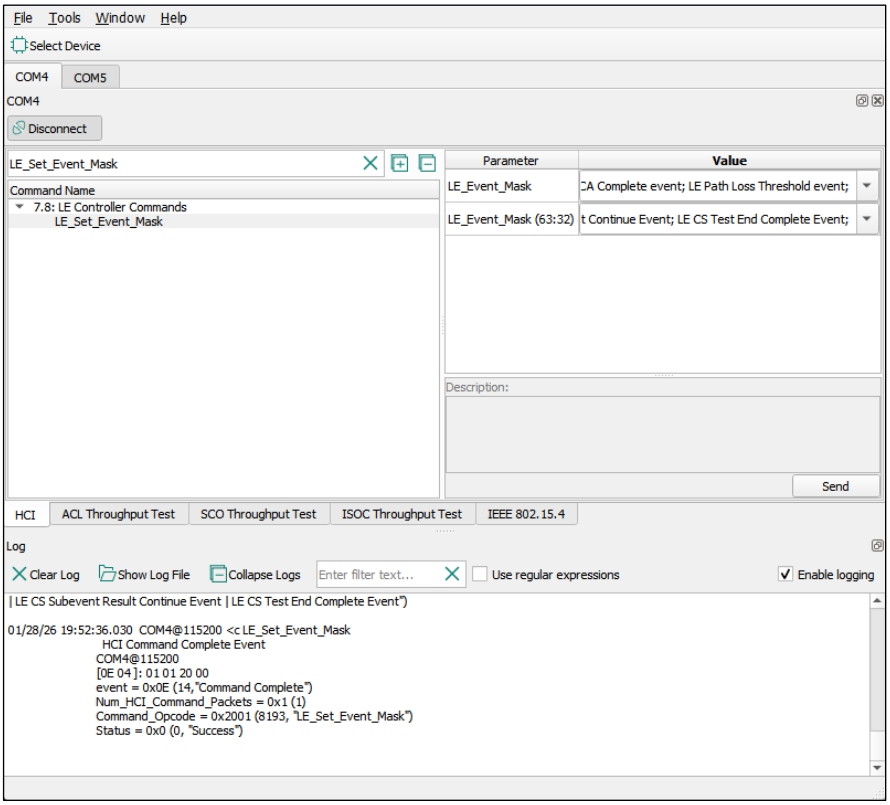
1. Select the **HCI** tab of the central device.
2. Select the **LE_Set_Periodic_Advertising_Enable** command.
3. Set the **Periodic_Advertising_Enable** command parameter to **Periodic advertising is enabled**.
4. Click **Send**.



3.17.8 Enable LE Create BIG Complete event for central device

When periodic advertising for the central device is enabled, to create a BIS between the selected devices, enable **LE Create BIG Complete event** for the central device:

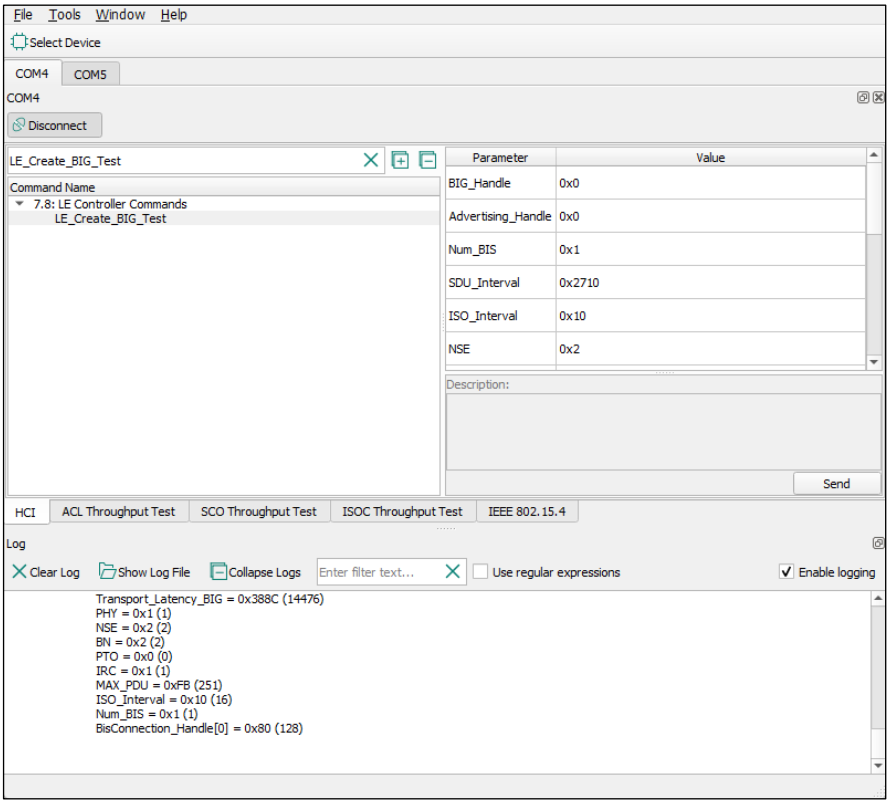
1. Select the **HCI** tab of the central device.
2. Select the **LE_Set_Event_Mask** command.
3. Open the **LE_Event_Mask** parameter and check **LE Create BIG Complete event**.
4. Click **Send**.



3.17.9 Create BIG

When **LE Create BIG Complete event** for the central device is enabled, to create a BIS between the selected devices, create a BIG:

1. Select the **HCI** tab of the central device.
2. Select the **LE_Create_BIG** or **LE_Create_BIG_Test** command.
3. Set the command parameters.
4. Click **Send**.

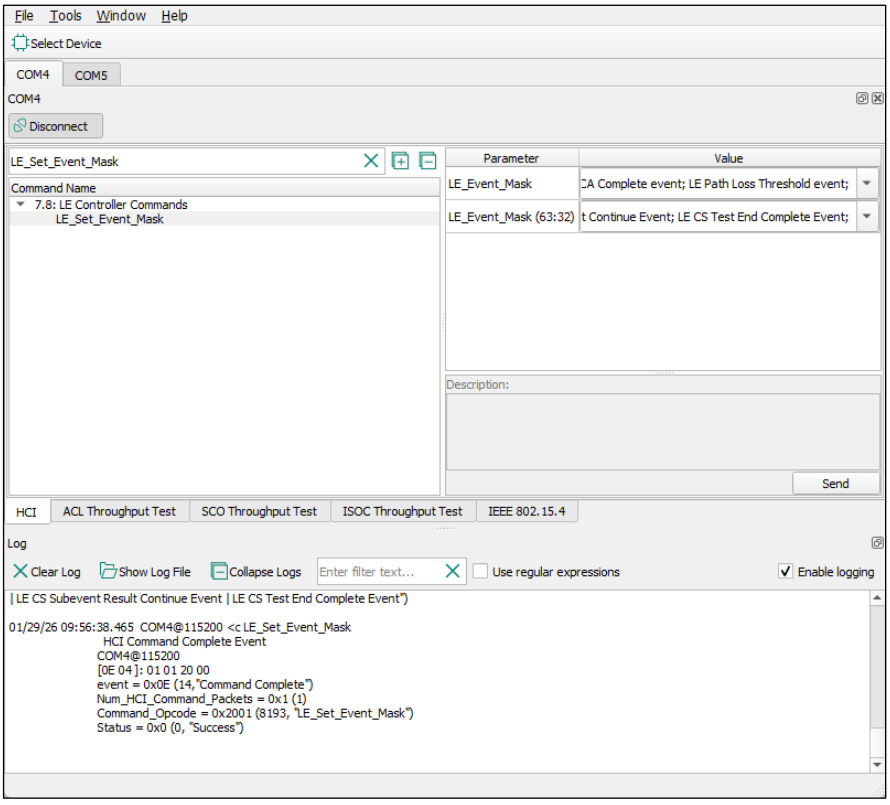


If a BIG is created successfully, a **LE Create BIG Complete event** is generated and BIS connection handles are reported.

3.17.10 Disable advertising reports for peripheral device

When a BIG is created, to create a BIS between the selected devices, we recommend you disable the advertising reports for the peripheral device:

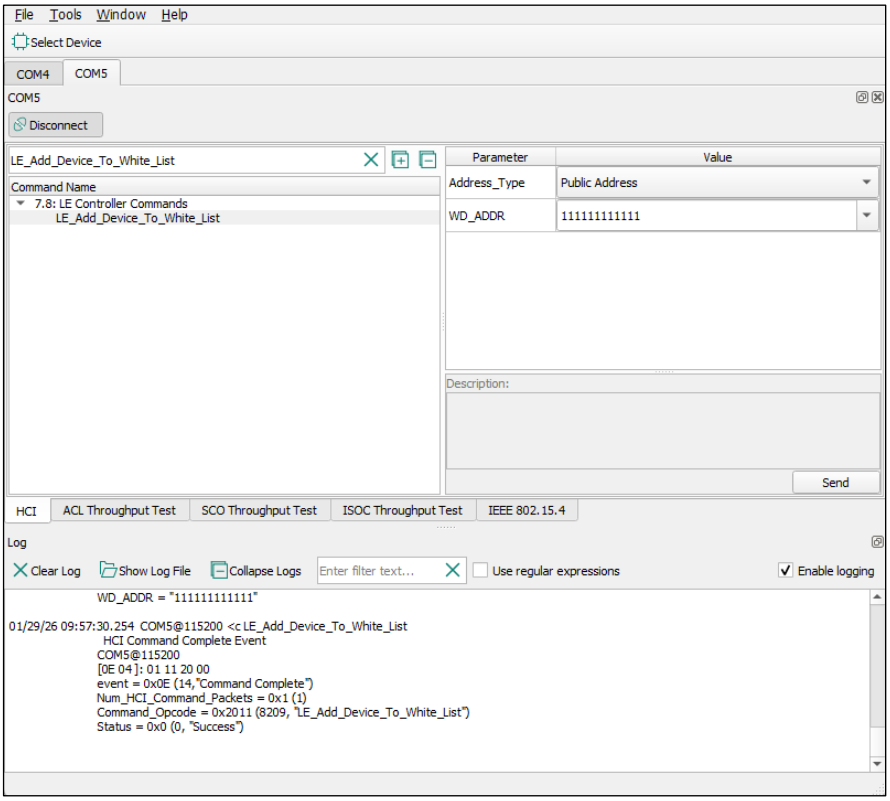
1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Set_Event_Mask** command.
3. Open the **LE_Event_Mask** parameter and uncheck **LE Extended Advertising Report Event** and **LE Periodic Advertising Report Event.**
4. Open the **LE_Event_Mask (63:32)** parameter and uncheck **LE BIGInfo Advertising Report Event**.
5. Click **Send**.



3.17.11 Add the central device to the whitelist of the peripheral device

When a BIG is created, to create a BIS between the selected devices, add the central device to the whitelist of the peripheral device:

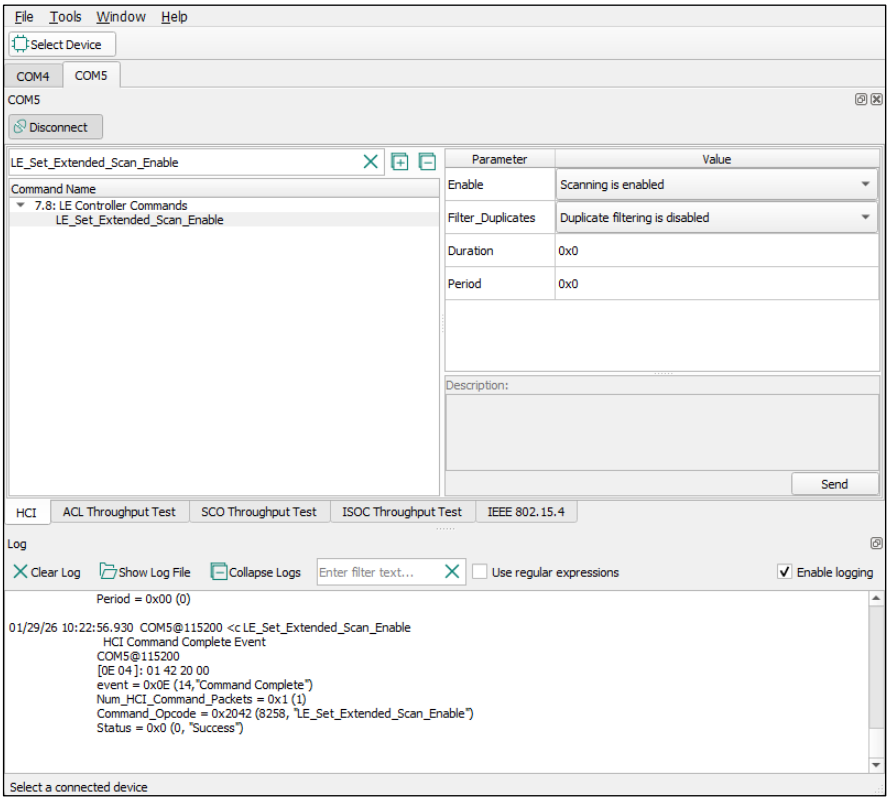
1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Add_Device_To_White_List** command.
3. Set the **WD_ADDR** command parameter to BD_ADDR of the central device.
4. Click **Send**.



3.17.12 Enable scanning on peripheral device

When the central device is added to the whitelist of the peripheral device, to create a BIS between the selected devices, enable scanning on the peripheral device:

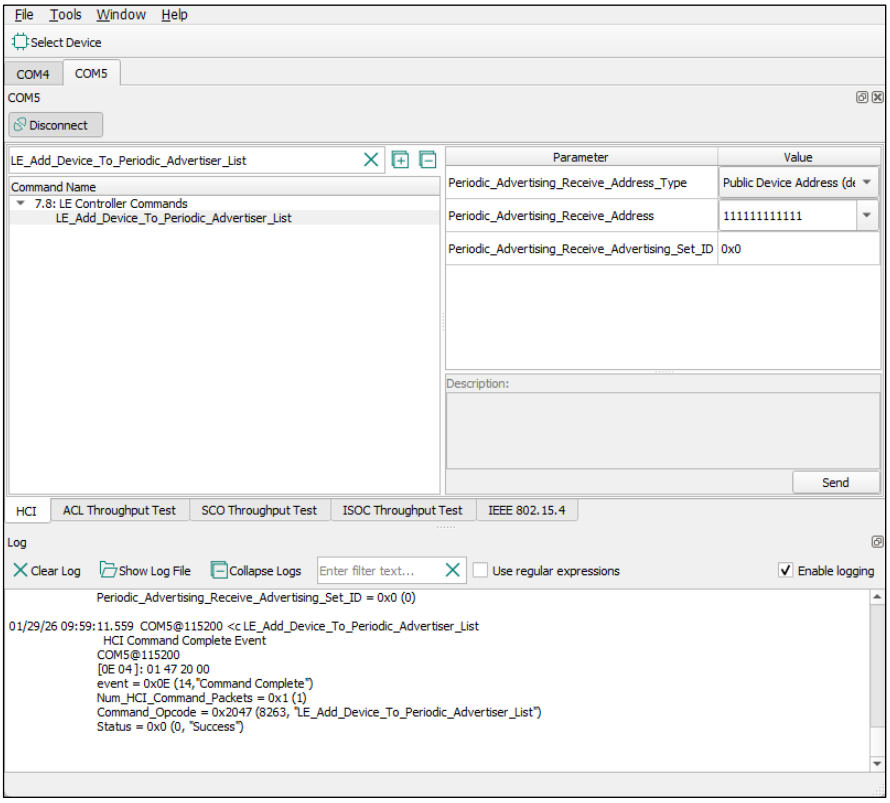
1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Set_Extended_Scan_Enable** command.
3. Set the **Enable** command parameter to **Scanning is enabled**.
4. Click **Send**.



3.17.13 Add central device to periodic advertiser list of peripheral device

When scanning on the peripheral device is enabled, to create a BIS between the selected devices, add the central device to the periodic advertiser list of the peripheral device:

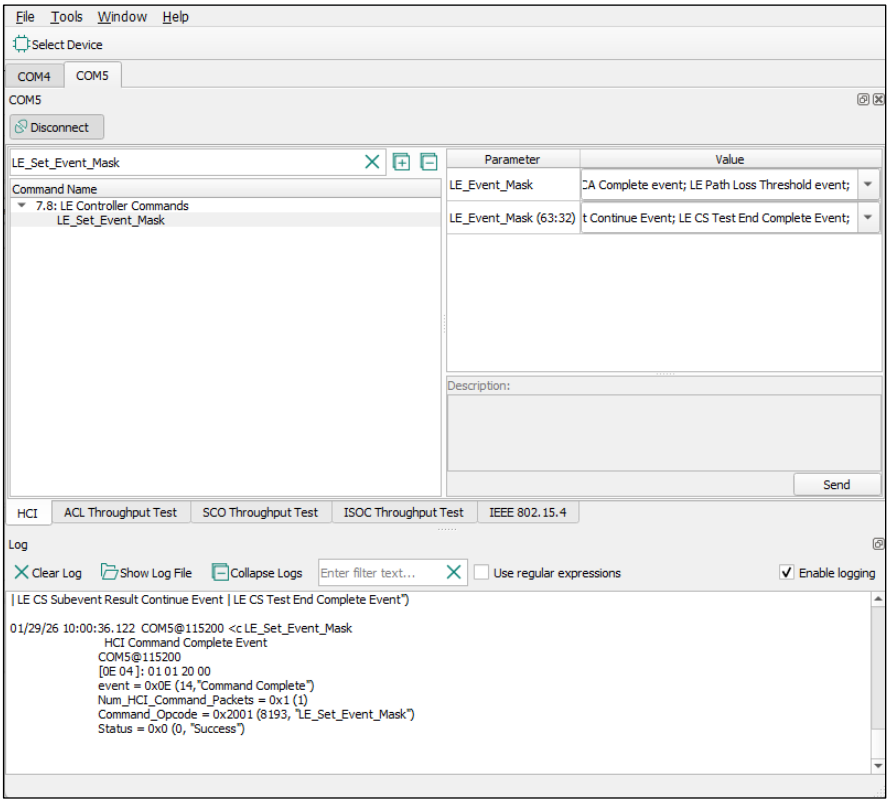
1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Add_Device_To_Periodic_Advertiser_List** command.
3. Set the **Periodic_Advertising_Receive_Address** command parameter to BD_ADDR of the central device.
4. Click **Send**.



3.17.14 Enable LE Periodic Advertising Sync Established event for peripheral device

When the central device is added to the periodic advertiser list of the peripheral device, to create a BIS between the selected devices, enable **LE Periodic Advertising Sync Established event** for the peripheral device:

1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Set_Event_Mask** command.
3. Open the **LE_Event_Mask** parameter and check **LE Periodic Advertising Sync Established event**.
4. Click **Send**.

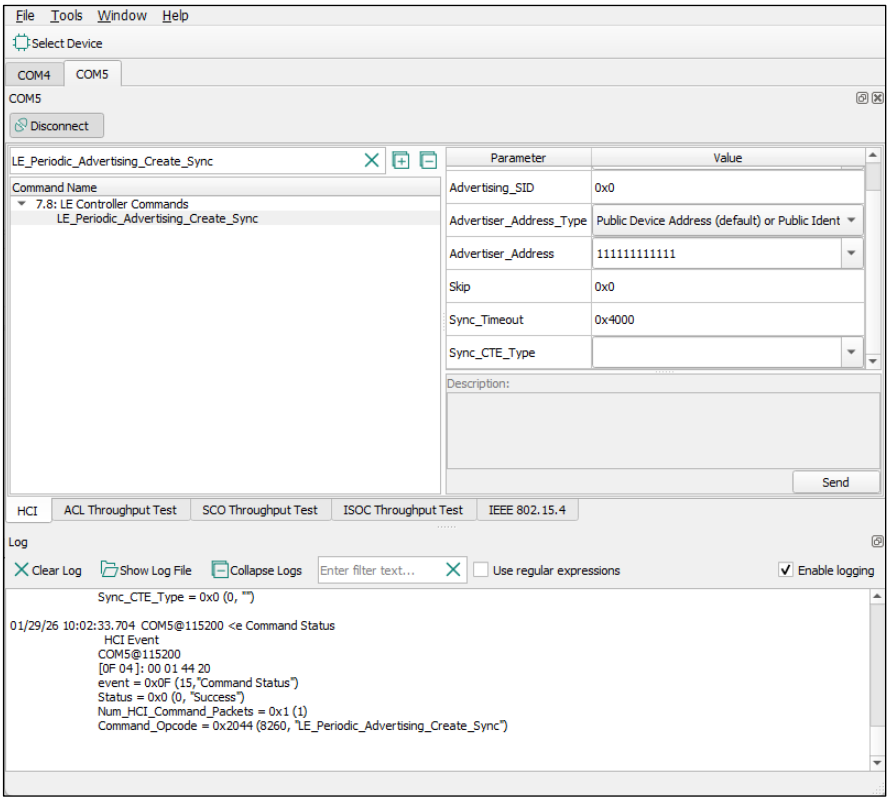


3.17.15 Synchronize peripheral device with periodic advertising from central device

When an **LE Periodic Advertising Sync Established event** for the peripheral device is enabled, to create a BIS between the selected devices, synchronize the peripheral device with the periodic advertising from the central device.

*Note: Synchronization can occur only when scanning is enabled by the **LE_Set_Extended_Scan_Enable** command.*

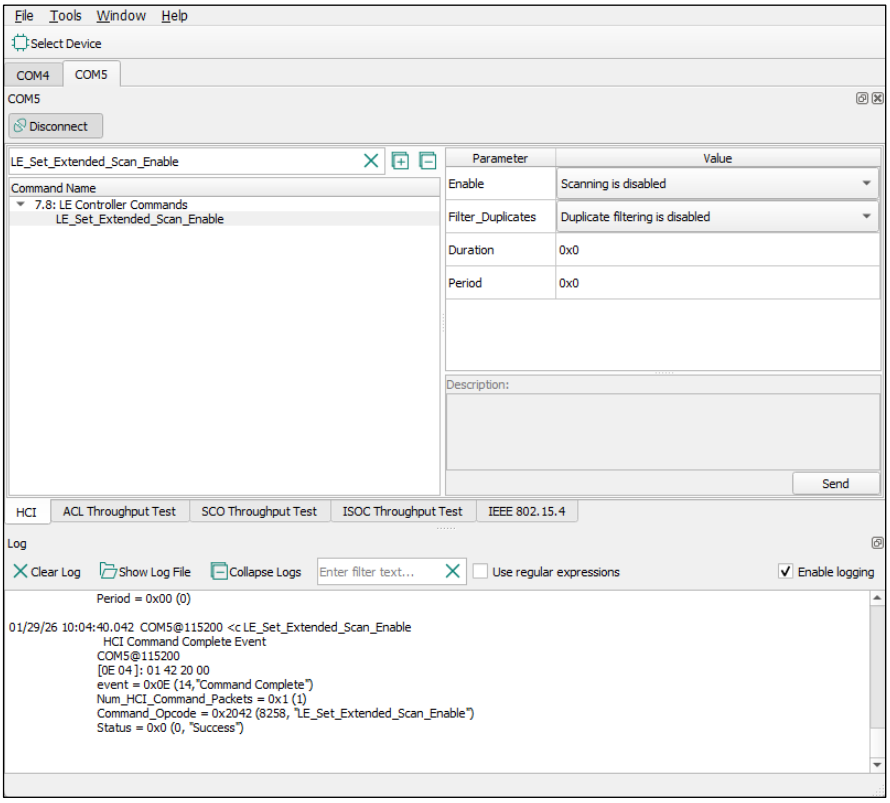
1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Periodic_Advertising_Create_Sync** command.
3. Set the command parameters.
4. Click **Send**.



3.17.16 Disable scanning on the peripheral device

When the peripheral device is synchronized with periodic advertising from the central device, you can disable scanning on the peripheral device:

1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Set_Extended_Scan_Enable** command.
3. Set the **Enable** command parameter to **Scanning is disabled**.
4. Click **Send**.



3.17.17 Enable LE BIG Sync Established event for the peripheral device

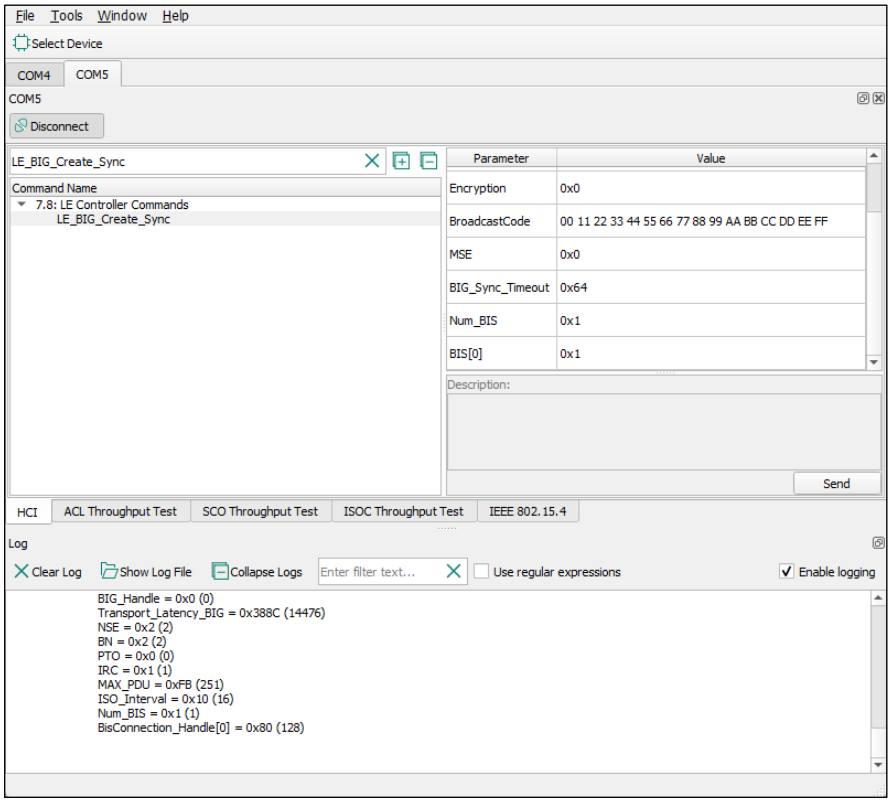
Before you synchronize the peripheral device to the BIG, to create a BIS between the selected devices, enable **LE BIG Sync Established event** for the peripheral device:

1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Set_Event_Mask** command.
3. Open the **LE_Event_Mask** parameter and check **LE BIG Sync Established event**.
4. Click **Send**.

3.17.18 Synchronize peripheral device to BIG

When an **LE BIG Sync Established event** for the peripheral device is enabled, to create a BIS between the selected devices, synchronize the peripheral device to the BIG:

1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_BIG_Create_Sync** command.
3. Set the command parameters.
4. Click **Send**.

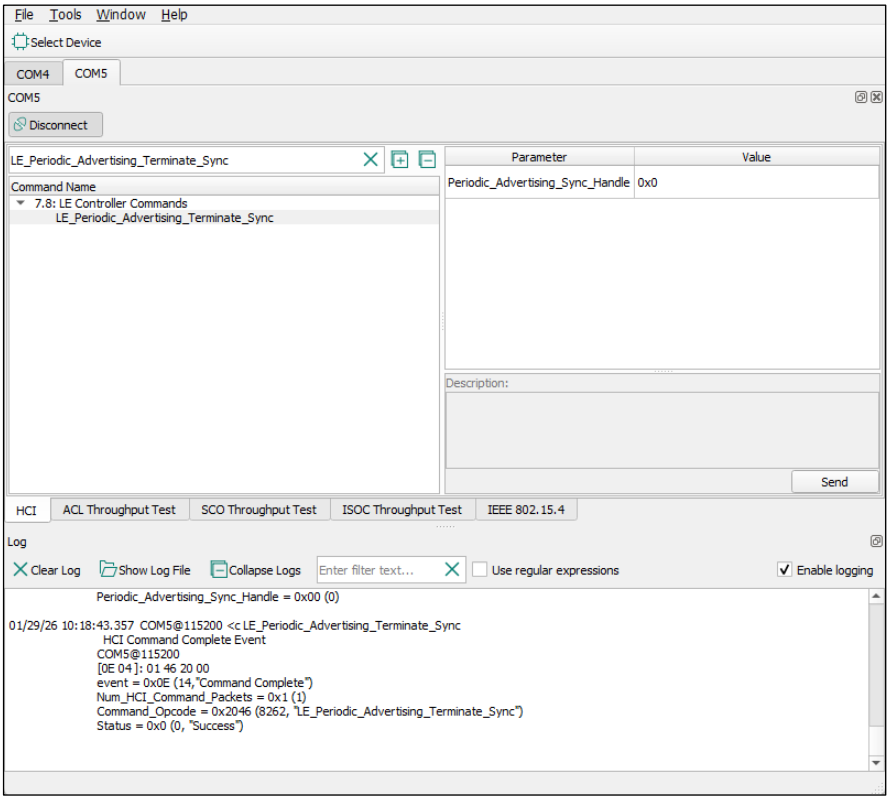


If the peripheral device is synchronized to the BIG successfully, a **LE BIG Sync Established event** is generated and BIS connection handles are reported.

3.17.19 Stop receiving advertising packets

When the peripheral device is synchronized with the BIG, you can stop receiving the advertising packets on the peripheral device:

1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Periodic_Advertising_Terminate_Sync** command.
3. Set the command parameters.
4. Click **Send**.



3.18 ISOC throughput tests

When a CIS or BIS between the selected devices is created, you can run ISOC throughput tests to check how fast ISOC data is transferred between the devices.

To run an ISOC throughput test, select the **ISOC Throughput Test** tab on a device tab or dockable window.

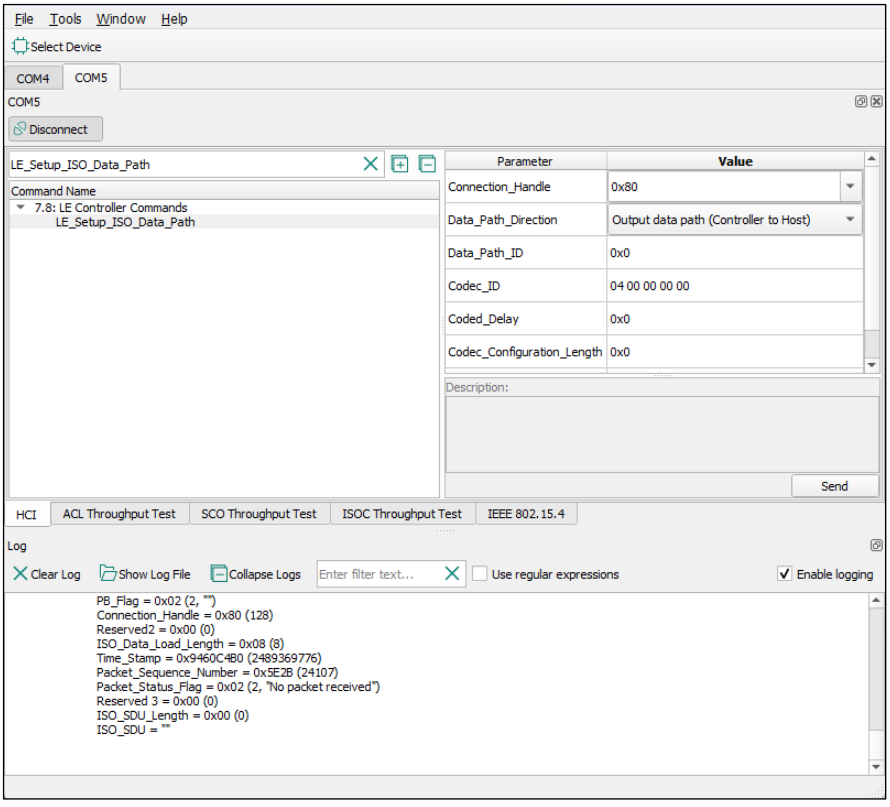
Note: *To run ISOC throughput tests for UART devices, we highly recommend you set the baud rate for both devices to 3000000 bps.*

3.18.1 Setup ISO data path

Before you run ISOC throughput tests, set the ISO data path for the selected devices.

First, set the ISO data path for the peripheral device:

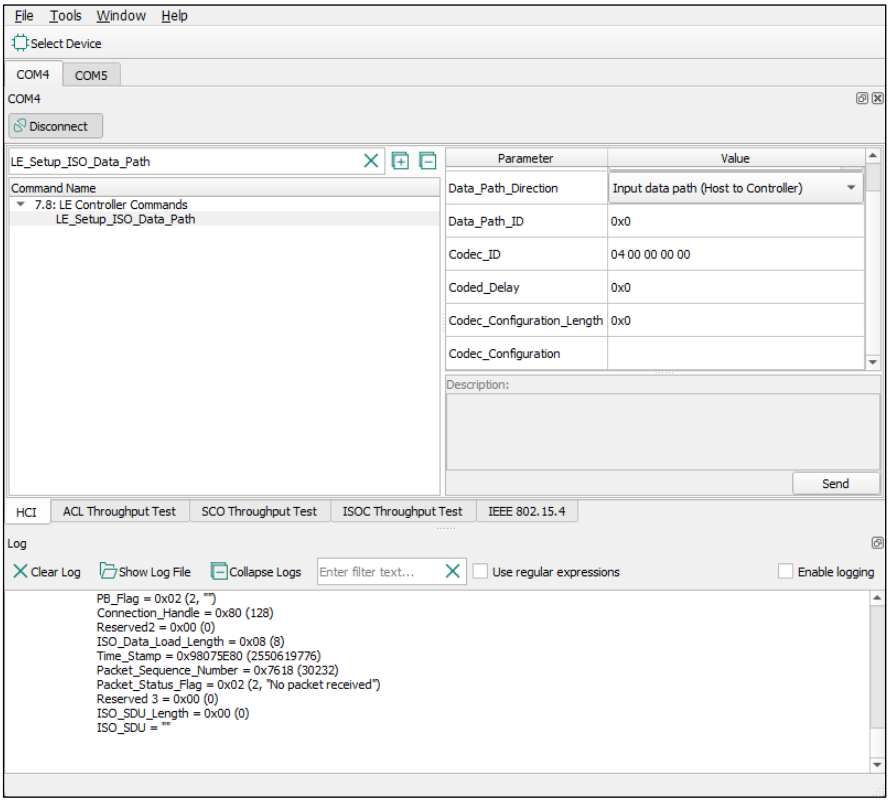
1. Select the **HCI** tab of the peripheral device.
2. Select the **LE_Setup_ISO_Data_Path** command.
3. Set the **Data_Path_Direction** command parameter to **Output data path (Controller to Host)**.
4. Click **Send**.



GUI application

Next, set ISO data path for the central device:

1. Select the **HCI** tab of the central device.
2. Select the **LE_Setup_ISO_Data_Path** command.
3. Set the **Data_Path_Direction** command parameter to **Input data path (Host to Controller)**.
4. Click **Send**.

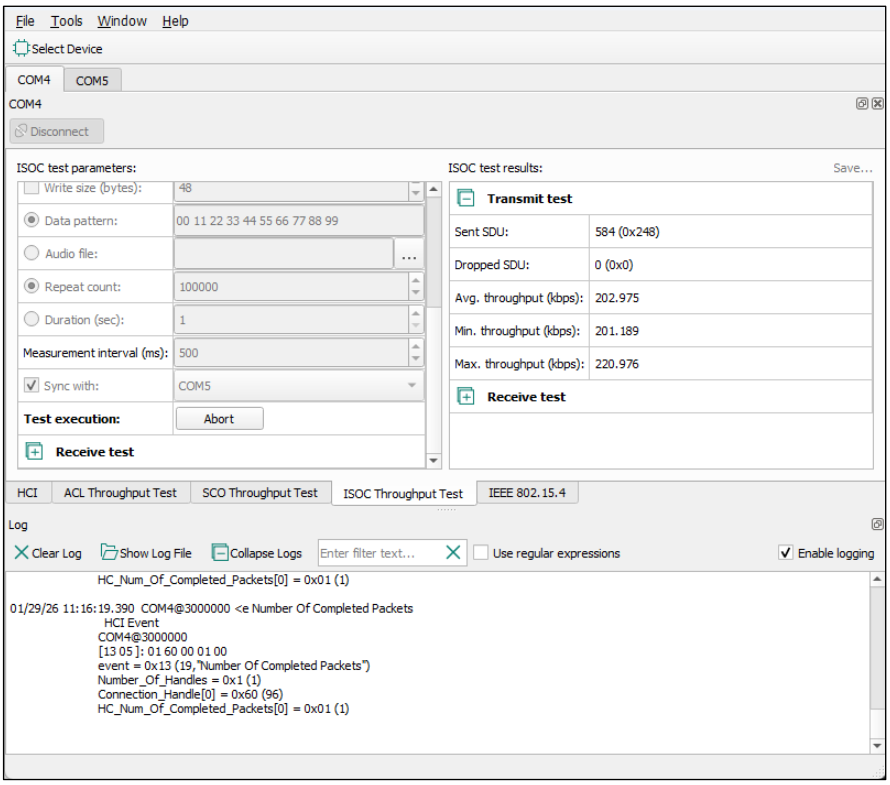


3.18.2 ISOC transmit test

The ISOC transmit test checks how fast ISOC data is transmitted by a Bluetooth® controller.

To run the ISOC transmit test:

1. Expand the **Transmit test** section of the **ISOC test parameters** column.
2. Set the test parameters.
3. Click **Start test**.



The following table lists parameters of the ISOC transmit test.

Parameter	Description	Values
Connection handle	Connection handle of CIS or BIS.	A valid connection handle.
SDU length (bytes)	SDU length in bytes.	Integer values between 1 and 65535. This value shall correspond to CIS or BIS parameters.
SDU interval (us)	SDU interval in microseconds.	Integer values between 255 and 1048575. This value shall correspond to CIS or BIS parameters.
Time stamp	Defines if time stamp will be included in ISOC packets.	Checked or unchecked.
Write size (bytes)	Defines the size of data blocks. If not set, the size of data blocks is defined by the size of the ACL buffer.	Integer values between 1 and 65535.
Data pattern	Sets the data to be transmitted.	A space-separated sequence of bytes in hexadecimal format.
Audio file	A WAV file with the audio data to be transmitted.	The absolute path to an existing WAV file.

GUI application

Parameter	Description	Values
Repeat count	Defines how many times the data pattern or audio file will be transmitted. Enabled only if **Data pattern** is used.	Integer values between 1 and 2147483647
Duration (sec)	Expected minimum duration of the test (in seconds). The actual duration of the test can be longer.	Integer values between 1 and 2147483647.
Measurement interval (ms)	Defines the time interval (in milliseconds) used to calculate throughput. At any moment of time, the instant throughput is calculated as the number of bytes sent during the measurement interval, divided by the time interval. Throughput values can be calculated only when the duration of the test is longer than the measurement interval.	Integer values between 500 and 2147483647.
Sync with	Allows to synchronize the transmit test and the receive test for another device. If set, the receive test for another device is started automatically before the transmit test starts. The transmit and receive tests parameters are synchronized.	A valid port name of the selected device.

To check the transmit test results, expand the **Transmit test** section of the **ISOC test results** column. The following table lists the test result parameters.

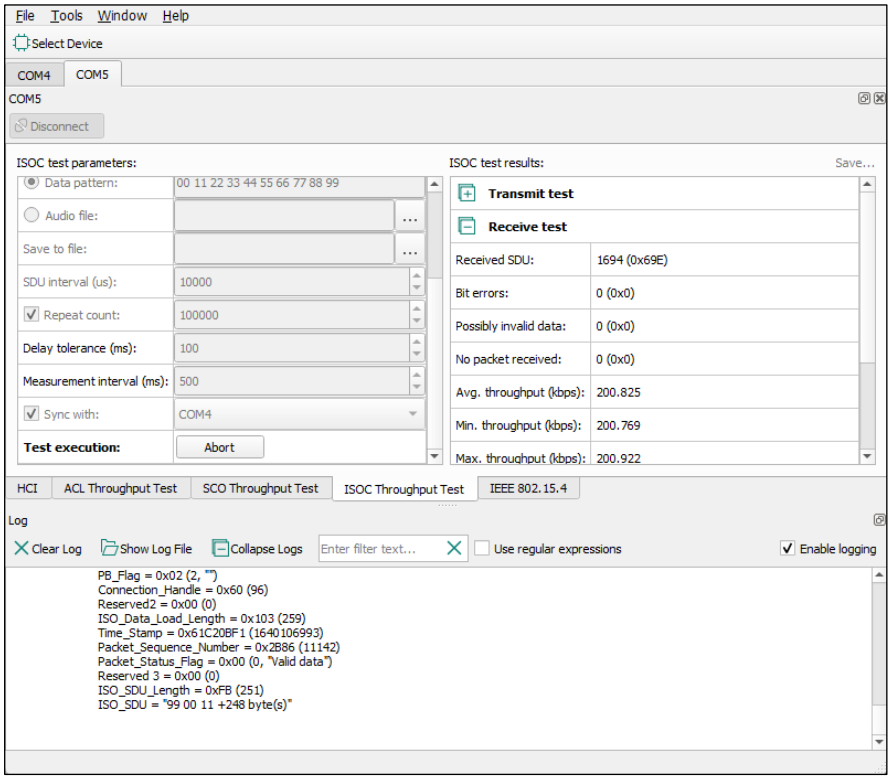
Parameter	Description
Sent SDU	The number of SDU sent to the Bluetooth® controller.
Dropped SDU	The number of SDU dropped by the Bluetooth® controller.
Avg. throughput (kbps)	The average transmit throughput (in kilobytes per second). The transmit throughput is calculated as a number of bytes completed (transmitted or flushed) by the controller within the measurement interval, divided by the time interval.
Min. throughput (kbps)	The minimum transmit throughput (in kilobytes per second).
Max. throughput (kbps)	The maximum transmit throughput (in kilobytes per second).

*Note: To save the results to a text file, click the **Save…** link located above the results section.*

3.18.3 ISOC receive test

The ISOC receive test checks how fast ISOC data is received by a Bluetooth® controller. To run the ISOC receive test:

1. Expand the **Receive test** section of the ISOC **test parameters** column.
2. Set test parameters.
3. Click **Start test**.



The following table lists the ISOC receive test parameters.

Parameter	Description	Values
Connection handle	Connection handle of CIS or BIS.	A valid connection handle.
Data pattern	Sets the data to be transmitted.	A space-separated sequence of bytes.
Audio file	A WAV file with the audio data expected to be received.	The absolute path to an existing WAV file.
Save to file	A WAV file to save the received data to. Available only if the **Audio** file is used.	The absolute path to the WAV file where to save the received audio data.
SDU interval (us)	The SDU interval (in microseconds). Available only if **Audio file** is used.	Integer values between 255 and 1048575. This value shall correspond to the CIS or BIS parameters.
Repeat count	Defines how many times the data pattern or data file will be transmitted. Enabled only if data pattern is used.	Integer values between 1 and 2147483647
Delay tolerance (ms)	Acceptable time interval (in milliseconds) between the time when data is sent to the transmitting Bluetooth® controller and the time when the data is received by the receiving Bluetooth® controller. The measured intervals can be larger than real values.	Integer values between 1 and 65535.

GUI application

Parameter	Description	Values
Measurement interval (ms)	Defines the time interval (in milliseconds) used to calculate throughput. At any time, the instant throughput is calculated as a number of bytes sent during the measurement interval, divided by the measurement interval. Throughput values can be calculated only when the duration of the test is greater than the measurement interval.	Integer values between 500 and 2147483647.
Sync with	Allows to synchronize the receive test and the transmit test for another device. If set, the transmit test for another device is started automatically after the receive test starts. Parameters of the receive and transmit tests are synchronized.	A valid port name of the selected device.

To check test results, expand the **Receive test** section of the **ACL test results** column. The following table lists test result parameters.

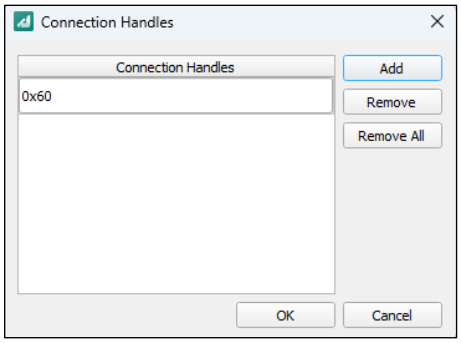
Parameter	Description
Received SDU	The number of SDU received by the device.
Bit errors	The total number of bits that were received incorrectly.
Possibly invalid data	The number of packets marked with the “Possibly invalid data” status flag.
No packet received	The number of packets marked with the “No packet received” status flag.
Avg. throughput (kbps)	The average transmit throughput (in kilobytes per second). Transmit throughput is calculated as the number of bytes, completed (transmitted or flushed) by the controller within the measurement interval, divided by the time interval.
Min. throughput (kbps)	The minimum transmit throughput (in kilobytes per second).
Max. throughput (kbps)	The maximum transmit throughput (in kilobytes per second).
Delayed packets	The total number of packets that were delayed by more than the value of delay tolerance. The measured intervals can be greater than real values.
Maximum delay (ms)	The maximum time interval (in ms) between the timestamp when a packet was sent by the transmitting device and the timestamp when the packet was received by the receiving device.
Average delay (ms)	The average time interval (in ms) between the timestamp when a packet was sent by the transmitting device and the timestamp when the packet was received by the receiving device.

*Note: To save the results to a text file, click the **Save…** link located above the results section.*

The receive test stops automatically if the expected number of bytes is received. Otherwise, you need to stop the receive test manually by clicking **Abort**.

3.18.4 Simultaneous ISOC throughput tests

To run ISOC transmit or receive throughput tests for multiple connections, in test settings, click the "**...**" button next to **Connection handle**. The Connection Handles dialog opens.



Enter the handles of connections for which you want to run the test and click **OK**.

Note: When you run a throughput test for multiple connections, you cannot synchronize the receive and transmit tests.

3.19 Wi-Fi non-signaling tests

This section provides a detailed guide for performing Wi-Fi non-signaling tests. The tests focus on validating the performance of a Wi-Fi chipset in transmit (Tx) and receive (Rx) modes, enabling rigorous validation at the physical layer without requiring network association or authentication.

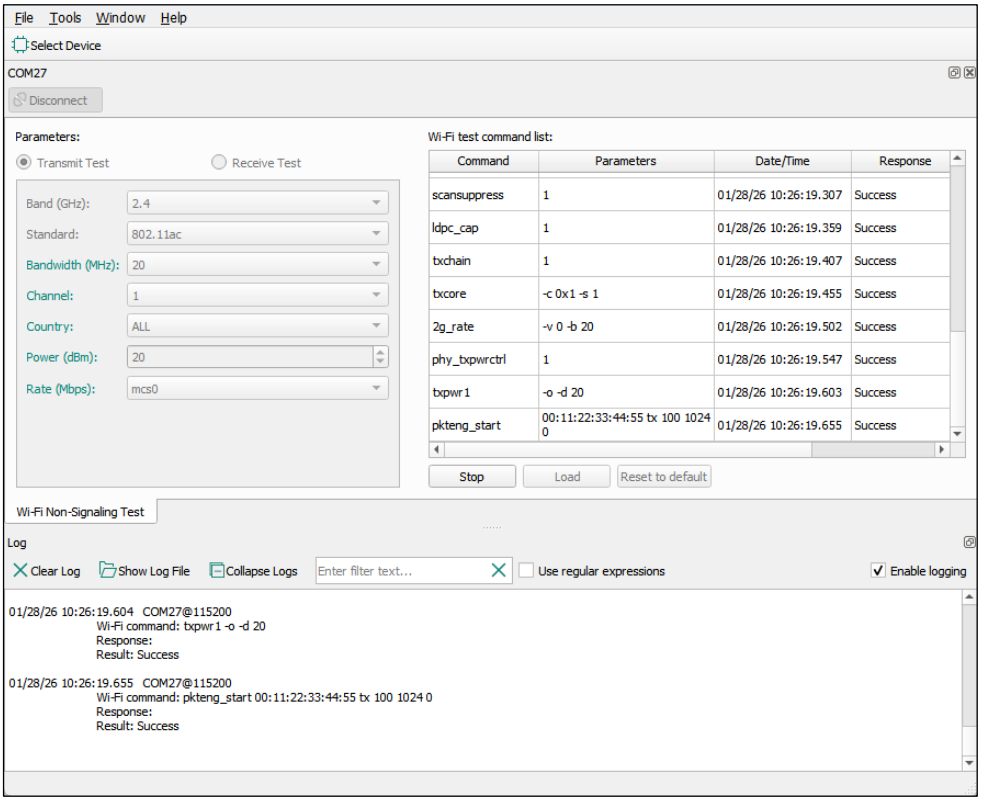
3.19.1 Executing non-signaling Tx test

The non-signaling **Tx test** verifies the transmission capabilities of a Wi-Fi chipset by continuously sending packets based on the configured parameters. The results of the Tx test require external analysis using tools like **LitePoint** or similar RF analyzers.

To start the test, open the **Wi-Fi Test Configuration Form** and configure the following parameters:
Set **Mode** to **Tx mode for transmission**, select the **Band** (e.g., 2.4 GHz, 5 GHz, or 6 GHz), and pick **Standard** (e.g., 802.11b, g, n, ac, ax). These are mandatory inputs required for configuring the test.

If required, additional variables such as **channel**, **bandwidth**, etc., may appear in the form, depending on the configuration file. Adjust these as necessary to align with the test plan or use their default values. The availability and range of such variables are predefined in the configuration file.

Click **Start** to begin the test. The WL tool sends sequential commands (as defined in the configuration) to set up a chipset and start transmitting packets. Logs will be generated in real time, displaying the commands issued, their status (Success/Failure), timestamps, and responses from the chipset.



Click **Stop**. This sends the **pkteng_stop** command to terminate the transmission and reset the chipset to its default state. Since the tool does not calculate results, transmission metrics such as power, signal stability, and modulation performance must be analyzed with **external RF analyzer**.

3.19.2 Executing non-signaling Rx test

The non-signaling **Rx test** evaluates the reception capabilities of a Wi-Fi chipset by monitoring a specific channel and capturing incoming packets. The result of the Rx test, presented as the **Packet Error Rate (PER),** will be displayed in the logs.

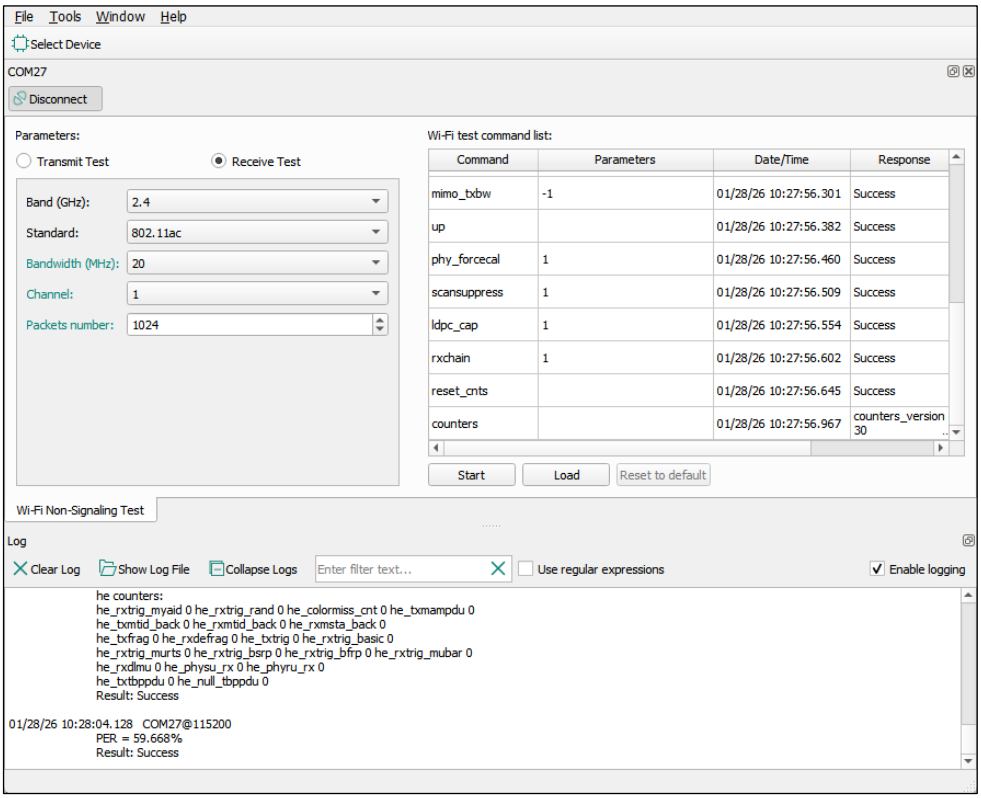
To configure the test, open the **Wi-Fi Test Configuration Form**. Select **Rx mode for reception**, select **Band** (e.g., 2.4 GHz, 5 GHz, 6 GHz), and select **Standard** (e.g., 802.11b, g, n, ac, ax) – the primary parameters for starting the test.

Based on the configuration file, additional receiver-specific variables may become available, such as **channel**, **bandwidth**, etc. Adjust these variables as needed or use their default values if applicable. If external equipment, such as a signal generator, is used to transmit test packets, ensure it is aligned with the channel and other parameters set in this test.

Click **Start** to initiate **Rx** mode. The WL tool applies the configurations and places the chipset into packet-monitoring mode. During the test, real-time packet counters are displayed in the interface, with key metrics like:

- **goodfcs:** Successfully received frames with valid checksums.
- **Dropped Frames:** Packets that failed validation or were lost.

GUI application



When the test is complete, click **Stop**. The tool automatically calculates **Packet Error Rate (PER)** based on the counter's data and displays it.

3.19.3 Custom configuration file

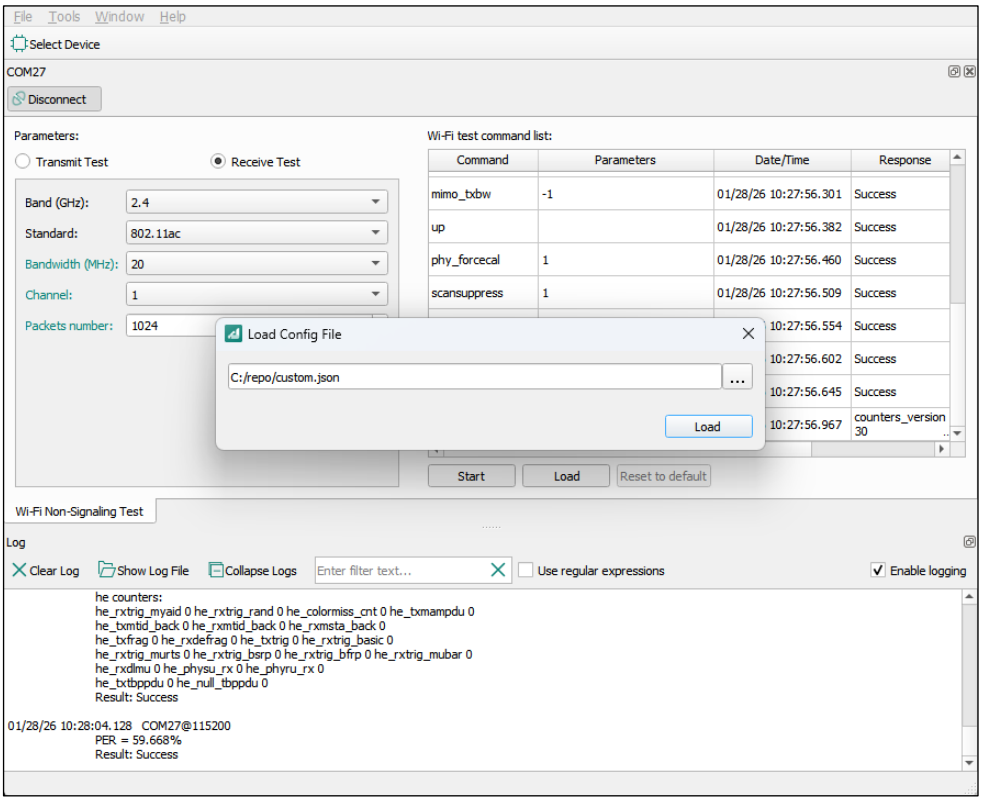
The AIROC™ Wi-Fi Test Tool allows the user to load custom configuration files to modify or define additional test parameters not covered by the default configuration. These files let you tailor the tool's behavior to specific requirements, such as expanded parameter ranges (e.g., **channel**, **bandwidth**, **cower**), new variables, or modified WL command sequences.

3.19.3.1 Load custom configuration file

Once the custom file is prepared, it can be loaded into the Wi-Fi Test Tool. To load custom configuration:

- Open the Wi-Fi form.
- Click **Load** on the Wi-Fi form.
- In the file selection dialog, navigate to your *custom.json* file.
- Click **Load** on the selection dialog.

GUI application



If there is an issue with the file, the tool will display an error and retain the most recently used configurations. If the file is valid, it will overwrite the current configuration of the **Wi-Fi tab**. To return to the default settings, click the **Reset to default** button on the **Wi-Fi tab**.

3.19.3.2 Configuration file syntax rules

The file must follow a standardized structure to ensure the compatibility with the tool. This includes defining both modes, their associated variables, and commands. Below are the syntax rules and key elements to follow.

- **Modes (Tx, Rx):** Define test configurations for transmission and reception.
- **Frequency Bands:** Specify valid bands (e.g., 2.4 GHz, 5 GHz).
- **Protocol Standards:** Include standards like 802.11b, 802.11ac, etc.
- **Variables:** List adjustable parameters (e.g., channel, power, bandwidth):
    1. Use **supported** for valid options (e.g., [1, 6, 11]).
    2. Use **min/max** for numeric ranges.
    3. Define **default** value for list or numeric ranges.
- **Commands:** Include sequences of WL commands with placeholders (e.g., %channel%).

3.19.3.3 Example custom file

Below is an example of a valid custom configuration file:

```
{
  "tx": {
    "2.4 GHz": {
      "802.11b": {
        "variables": {
          "power": { "min": 0, "max": 22, "default": 20 }
        },
        "commands": [
          "up",
          "channel 6",
          "txpwr1 -o -d %power%",
          "pkteng_start 00:11:22:33:44:55 tx 100 1024 0"
        ]
      }
    }
  },
  "rx": {
    "5 GHz": {
      "802.11ac": {
        "variables": {
          "channel": { "supported": [36, 40, 44], "default": 36 }
        },
        "commands": [
          "up",
          "channel %channel%",
          "pkteng_start 00:11:22:33:44:55 rx 100 0",
          "counters"
        ]
      }
    }
  }
}
```

3.20 IEEE 802.15.4 tests

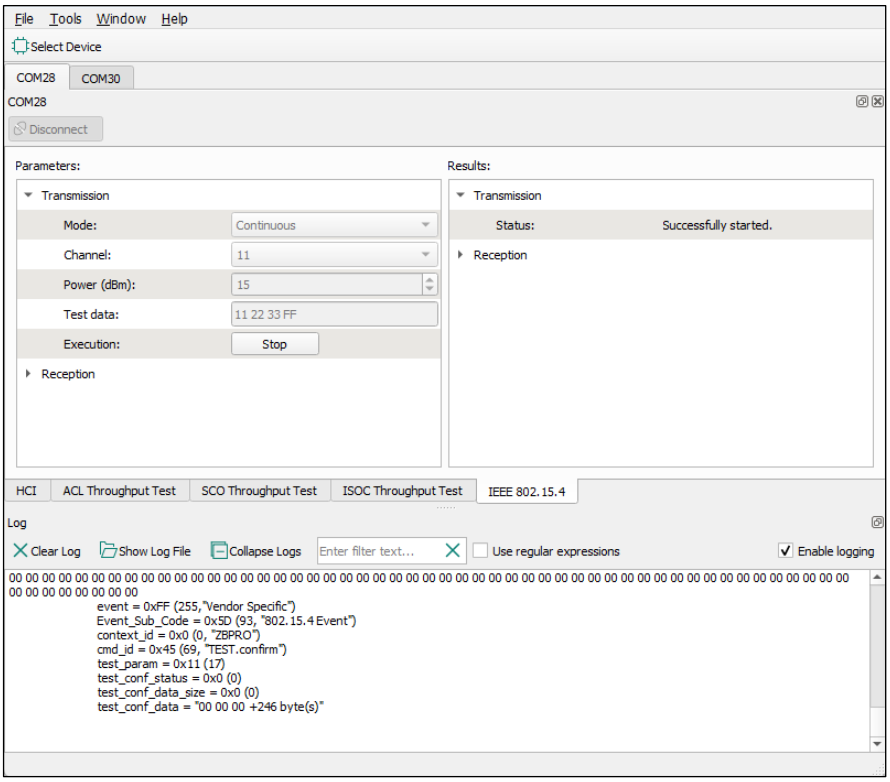
This section provides a detailed guide to performing IEEE 802.15.4 tests.

3.20.1 Run Tx test

There are two options for starting the Tx test: **Continuous** and **Finite**.

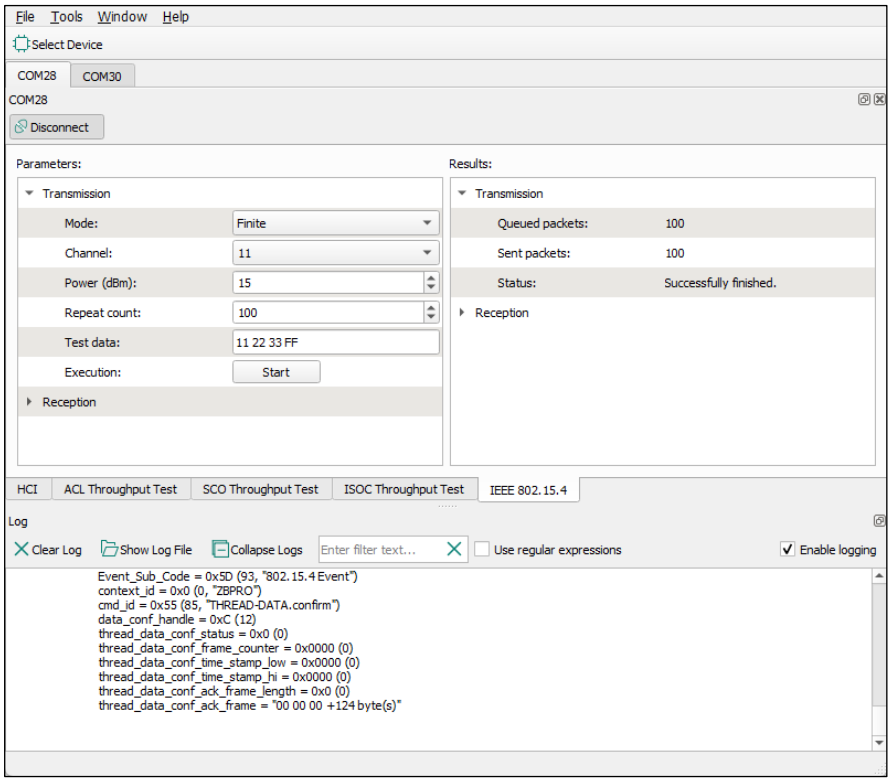
Continuous mode sends packets in an endless loop until stopped by the user. It is recommended to use **Continuous** mode with an RF analyzer only. The **Results** section only displays the start/stop status of the test. To start the **Continuous** test:

- Open the **IEEE 802.15.4 tab**.
- Expand the **Transmission** section on the parameters section.
- Select **Continuous** mode.
- Choose the channel, enter the test data (HEX array), set transmit power.
- Click **Start**.


File Tools Window Help
Select Device
COM28 COM30
COM28
Disconnect
Parameters:
Results:
Transmission
Mode: Continuous
Channel: 11
Power (dBm): 15
Test data: 11 22 33 FF
Execution: Stop
Reception
Transmission
Status: Successfully started.
Reception
HCI ACL Throughput Test SCO Throughput Test ISOC Throughput Test IEEE 802.15.4
Log
Clear Log Show Log File Collapse Logs Enter filter text... Use regular expressions Enable logging
00 00 00

GUI application

2. Select **Finite** mode.
3. Choose the channel, enter the test data (HEX array), set repeat count and transmit power.
4. Click **Start**.



The transmission will automatically stop if an error occurs during the process or after the specified number of packets has been sent. Also, the test can be stopped manually by clicking **Stop**.

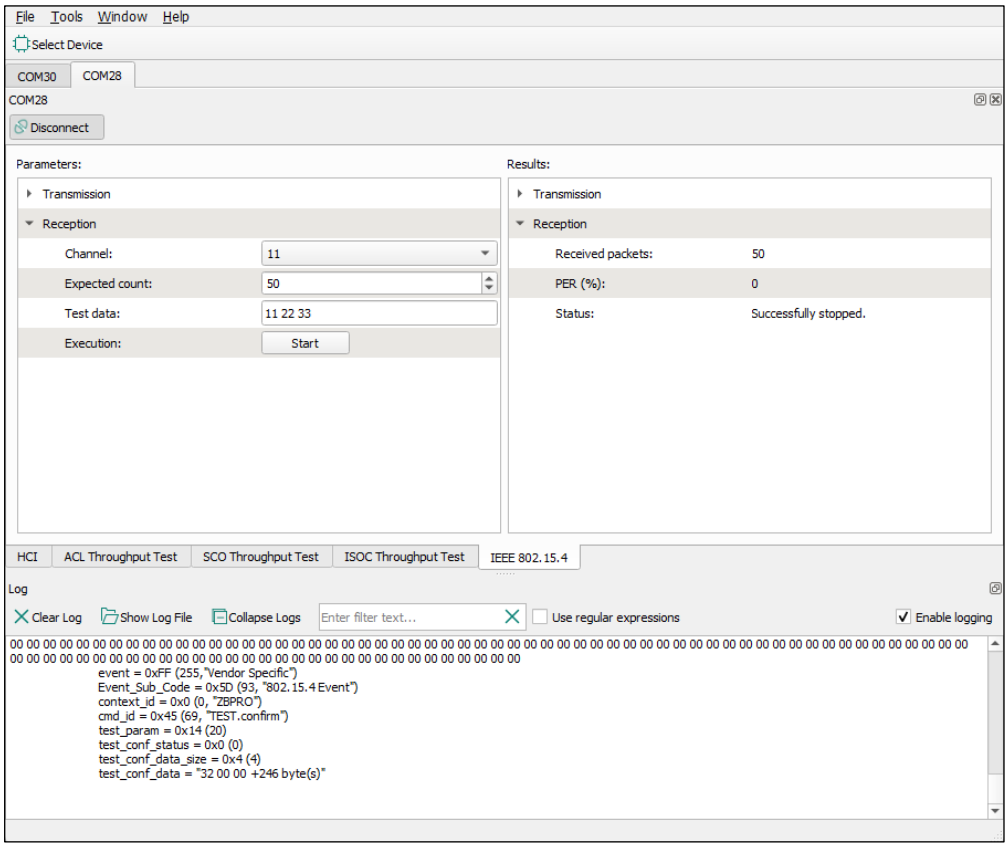
After the test finishes the results will display **Queued packets** (the number of packets sent to the device), **Sent packets** (the actual number of packets sent over the air) and the test execution status.

3.20.2 Run Rx test

The **Rx test** can be used with a signal generator or with another device running the finite **Tx test**. To start the **Rx test:**

1. Open the **IEEE 802.15.4 tab**.
2. Expand the **Reception** field in the parameters section.
3. Select the channel, enter the test data (HEX array), and set the **Expected Count** (the number of packets you expect; this is required to calculate the Packet Error Rate).
4. Click **Start**.

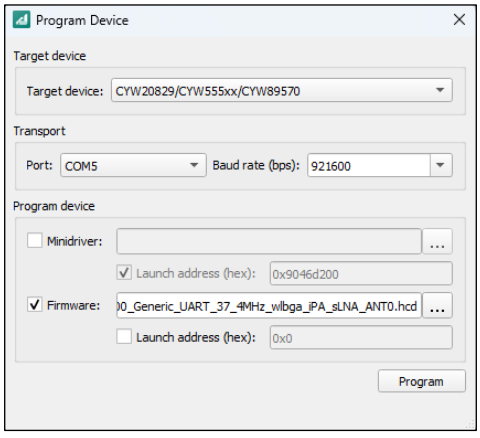
GUI application



To stop the test, click **Stop**. The results in the **Reception** field will provide information about the number of received packets, the **Packet Error Rate** and the test execution status.

3.21 Program a device

To program an AIROC™ Bluetooth® controller, connect the device to your computer and select **Tools > Program Device** from the main menu or press **Ctrl+P**. The "Program Device" dialog opens.

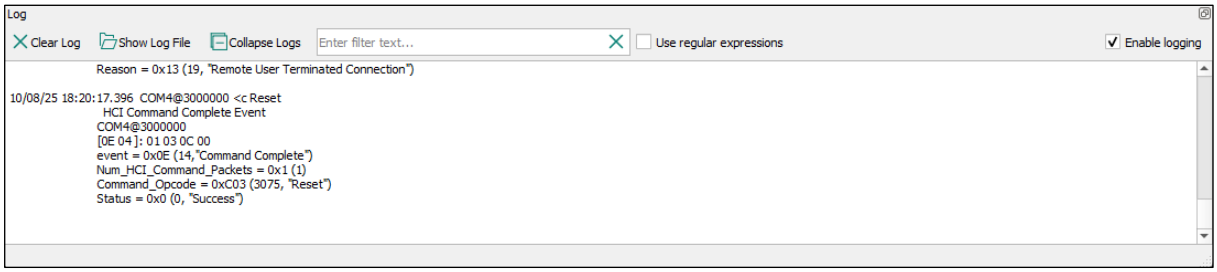


Select **Target device** to match the device to program and set the parameter values. Different target devices may have different parameters. When the parameters are set, click **Program** the device.

Note: *If the device is selected in the GUI, the device tab or dockable window closes. For the USB transport, the port name of the programmed device can change after programming.*

3.22 Logging

The GUI prints information about HCI commands and HCI events in the log window and in the log file. The location of the log file is printed in the log window when the application starts.



3.22.1 Clear log

Click **Clear Log** to erase the contents of the **Log** window. You can also select **Tools > Clear Log** on the main menu.

3.22.2 Show log file

Click **Show Log File** to open the location of the log file on disk. You can also select **Show Log File** on the main menu.

3.22.3 Collapse and expand logs

Click **Collapse Logs** to display the first line of each log message that is printed in the log window. The first line of a log message usually indicates the port name and the name of the HCI command or event. Click **Expand logs** to display complete log messages in the log window.

3.22.4 Filter logs

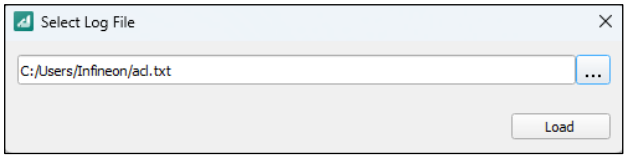
The **Filter** text box will limit messages shown in the Log window to match the filter text. To use regular expressions, check the **Use Regular expression** checkbox next to the filter text box.

3.22.5 Enable/disable logging

Toggle the **Enable logging** checkbox off to disable logging if there is a negative impact on performance. Toggle it back on to re-enable logging.

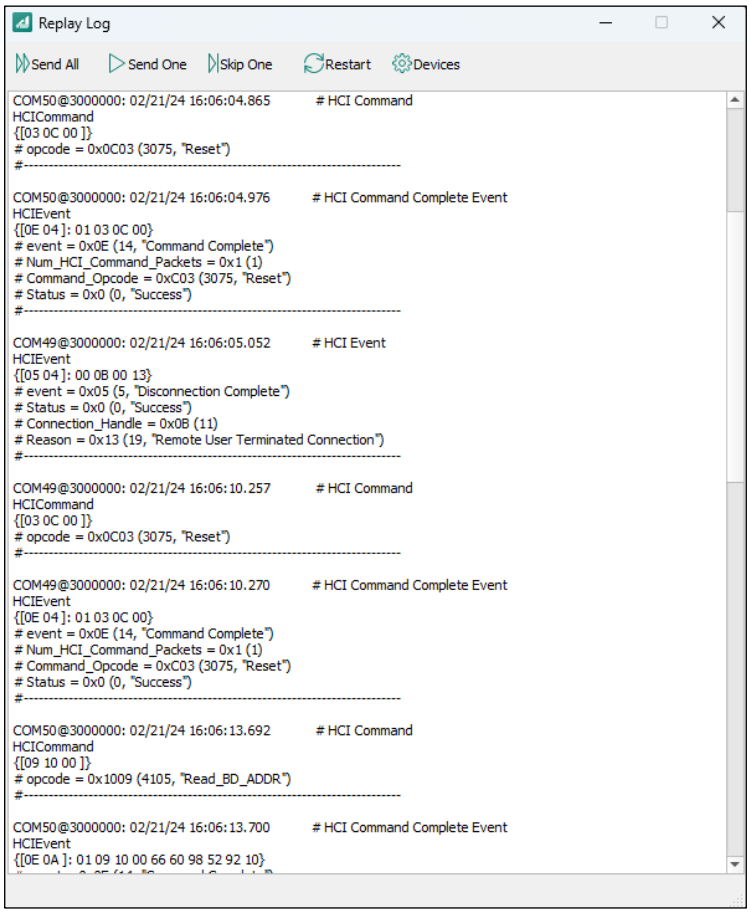
3.22.6 Replay log

To send HCI commands from a previously saved log file, select the devices to which to send the commands. Then, select **Tools > Log > Replay log** on the main menu. In the dialog that opens, select the log file, and click **Open**.



GUI application

The Replay Log dialog opens.



The following table describes toolbar buttons:

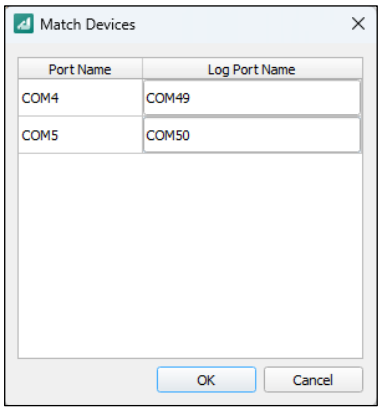
Button	Description
Send All	Send all commands from the selected log file.
Send One	Send the next command from the selected log file.
Skip One	Skip the next command from the selected log file.
Restart	Go to the beginning of the log file.
Devices	Allow the user to match port names in the log file and port names of the selected devices. See Device matching for details.

Commands marking: "sent" – green, not "sent" – red, "skipped" – blue. The status of the latest command is displayed in the status bar.

3.22.6.1 Device matching

You can match port names of the selected devices and port names in the log file. To match the port names, click **Devices**. The "Match Devices" dialog opens.

GUI application

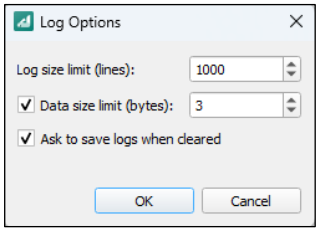


The **Port Name** column contains port names of the selected devices. For each selected device, in the **Log Port Name** column, enter a port name present in the log file. For example, if **Port Name** is COM4 and **Log Port Name** is COM12, the commands, which were sent to COM12 in the log file, will be sent to COM4.

If **Log Port Name** is not set, **Log Port Name** is equal to **Port Name**.

3.22.7 Log options

To edit the log options, select **Tools > Log Options** on the main menu or press **Ctrl+L**. The "Log Options" dialog opens.



The following table describes the available log options and their descriptions.

Option	Description
Log size limit	The maximum number of lines in the log window.
Data size limit	The maximum number of data bytes that can be printed in the logs.
Ask to save logs when cleared	Defines if a dialog asks to save the content of the log window to a file when you clear the log window.

*Note: Large values of log size limit and data size limit can slow down the GUI. **Log size limit** affects the log window only and does not affect the log files. **Data size limit** affects both the log window and log files.*

4 Console application

This section describes the AIROC™ test and debug tool console application and how to open it on different operating systems

4.1 Open the console application

This section describes how to open the console application on different operating systems.

4.1.1 Windows

Navigate to the folder where the tool is installed, and run the *airocdebugtoolconsole.exe* executable.

4.1.2 Ubuntu Linux

1. To open the console application, open a terminal and navigate to the folder where the tool is installed. For example:

   ```
   cd /opt/Tools/AirocDebugTool/<version>
   ```

2. To open the console application, run the *airocdebugtoolconsole* executable. For example:

   ```
   ./airocdebugtoolconsole
   ```

4.1.3 mac OS

Open a terminal, navigate to the folder where the tool is installed, and run the *airocdebugtoolconsole* executable. For example:

```
/Applications/AirocDebugTool/<version>/airocdebugtoolconsole.app/Contents/MacOS/airocdebugtoolconsole
```

4.2 Command line options

The following table lists command line options which you can use when opening the console application.

Option	Description
`--help, -h, -?`	Displays help on command line options.
`--help-all`	Displays help on command line options including Qt specific options.

4.3 Open device

Before you can send commands to an AIROC™ device, open the device using the console application. To open a device, connect the device to your computer and type the **Open** command and specify the port name. For example:

```
open usb0
```

Note: In the console application, only one device can be opened at time. To open another device, close the device, which is already opened.

4.4 Close a device

When the device is opened using the console application, to disconnect the device, type "close". For example:

```
(usb0)$ close
```

Console application

```
$
```

Then, you can open another device.

4.5 List HCI commands

When the device is opened using the console application, you can check the list of HCI commands that can be sent to the device.

The HCI commands are grouped in sections. To list all section names, type:

```
list --section
```

For example:

```
(usb0)$ list -section
host_controller_baseband_commands
informational_parameters
le_controller_commands
link_control_commands
link_policy_commands
status_parameters
testing_commands
vendor_specific_commands
```

To list all commands belonging to a specified section, type:

```
list -filter-by-section <section name>
```

For example:

```
(usb0)$ list -filter-by-section informational_parameters
read_local_version_information
read_local_supported_commands
read_local_supported_features
read_local_extended_features
read_buffer_size
read_country_code
read_bd_addr
read_local_supported_codecs
```

To list all commands, type:

```
list
```

4.6 Send HCI commands and receive responses

To send an HCI command without parameters, type:

```
send <command name>
```

If the command generates HCI events, the events information is printed. For example:

```
(usb0)$ send read_bd_addr
08/16/21 13:33:23.409  usb0 c> Read_BD_ADDR
          HCI Command
        usb0
        [09 10 00 ]
        opcode = 0x1009 (4105, "Read_BD_ADDR")

(usb0)$ (usb0)$ 08/16/21 13:33:23.411  usb0 <c Read_BD_ADDR
          HCI Command Complete Event
        usb0
```

Console application

```
[0E 0A ]: 01 09 10 00 DA 26 8E 9E 15 A4
event = 0x0E (14,"Command Complete")
Num_HCI_Command_Packets = 0x1 (1)
Command_Opcode = 0x1009 (4105, "Read_BD_ADDR")
Status = 0x0 (0, "Success")
BD_ADDR = "A4159E8E26DA"
```

To check an HCI command parameters, type:

```
send <command name> --help
```

For example:

```
(usb0)$ send write_bd_addr --help

Usage: send write_bd_addr [options]
This command will write the value for the device's Bluetooth device address.

Options:
  -?, -h, --help       Displays help on command line options.
  --help-all           Displays help including Qt specific options.
  --bd_addr <bd_addr>  BD_ADDR, The Bluetooth device address
```

To send an HCI command with parameters, type:

```
send <command name> <parameters>
```

For example:

```
(usb0)$ send write_bd_addr -bd_addr FFFFFFFFFFFF
08/16/21 13:41:12.067  usb0 c> Write_BD_ADDR
          HCI Command
        usb0
        [01 FC 06 ]: FF FF FF FF FF FF
        opcode = 0xFC01 (64513, "Write_BD_ADDR")
        BD_ADDR = "FFFFFFFFFFFF"

(usb0)$ (usb0)$ 08/16/21 13:41:12.070  usb0 <c Write_BD_ADDR
          HCI Command Complete Event
        usb0
        [0E 04 ]: 01 01 FC 00
        event = 0x0E (14,"Command Complete")
        Num_HCI_Command_Packets = 0x1 (1)
        Command_Opcode = 0xFC01 (64513, "Write_BD_ADDR")
        Status = 0x0 (0, "Success")
```

4.7 Program device

To program the connected device, type:

```
program <parameters>
```

Different devices may require different parameters. The following table lists the parameters for CYW207xx, CYW20819/20, CYW943xx, and their descriptions.

Parameter	Description
`--target`	The target device (CYW207xx, CYW20819/20 or CYW943xx).
`--port`	The port name.
`--baudrate`	The baud rate (required for UART devices).
`--btp`	The path to the BTP file.
`--config`	The path to the config file.

Console application

Parameter	Description
`--firmware`	The path to the firmware file.
`--minidriver`	The path to the minidriver file.
`--dlminidriver`	Send the download minidriver command.
`--bdaddr`	New BD_ADDR of the device.

The following table lists the parameters for CYW20829, CYW555xx, CYW89570, and their descriptions.

Parameter	Description
`--target`	The target device (CYW20829, CYW555xx or CYW89570).
`--port`	The port name.
`--baudrate`	The baud rate.
`--minidriver`	The path to the minidriver file.
`--minidriverlaunchaddress`	The minidriver launch address.
`--firmware`	The path to the firmware file.
`--firmwarelaunchaddress`	The firmware launch address.

The following are examples of commands to program a device:

```
program –target CYW555xx –port COM6 –baudrate 921600 –firmware <path to firmware.hcd>

program –target CYW207xx –port COM6 –baudrate 3000000 –btp <path to the BTP file> -
minidriver <path to minidriver.hex> --config <path to config.hex> --bdaddr
20706A203323 –dlminidriver>
```

4.8 Print help

To get help on the commands of the console application, type `help`. For example:

```
$ help

Commands supported by the application.
Type <command> -h or <command> --help to get more information about a command

Commands:
  exit   Close the console and exit
  send   Send a command to the connected device. Type "send <command> --help"
           to get more information about a command.
  List   Lists supported commands
  close  Disconnect from COM port
  open   Connect to COM port
```

To get help on a specific command:

```
<command name> -help
```

For example:

```
$ list –help
Usage: list [options]
Lists supported commands

Options:
  -?, -h, --help                          Displays help on command line
                                          options.
  --help-all                              Displays help including Qt specific
                                          options.
  --section                               list only the section / group name
```

```
                                          of the commands
  --filter-by-section <filter-by-section>  filter commands by section/group
                                          Name
```

4.9 Exit console application

To exit the console application, type `exit`.

5 Troubleshooting

The following table lists possible issues with AIROC™ test and debug tool and their solutions.

Issue	Possible solution
A UART device does not respond after being selected.	• Check if you have selected the correct port. Some AIROC™ Bluetooth® devices have two UART ports, the correct port is usually the one with the lower port number or the one which has HCI in the port name. • Make sure that you are using the correct baud rate.
On Windows, MSVCP140_1.dll is not found.	• Download and install Microsoft Visual C++ Redistributable. • Reboot the computer after the installation of Microsoft Visual C++ Redistributable.
On Windows, a USB device does not respond after programming.	• Try another port name for the same device. For example, if the selected device was **usb0** before programming, try **usb1**.
On Windows 11, a BTWUSB device (e.g., CYW4373) does not respond.	Turn off memory integrity and assign the BTWUSB driver to the connected device. To turn off memory integrity: 1. Tap the Windows **Start** button. 2. Type **windows security** and select the option from the search results. 3. Select **Device security**. 4. Select **Core isolation**. 5. Turn off the **Memory integrity** setting. 6. Reboot the computer. To assign the BTWUSB driver to the connected device: 1. Tap the Windows **Start** button. 2. Type **device manager** and select it from the results. 3. Locate the connected device which is often found under **Other devices**. 4. Select **Properties** from the context menu. 5. Click **Update driver**. 6. Click **Browse my computer for drivers**. 7. Navigate to the AIROC™ test and debug tool installation folder and select the **drivers** subfolder. 8. Select **Next**.
On Ubuntu Linux, a BTUSB device (e.g., CYW4373) does not respond.	• Reinstall the BTUSB driver and reboot the computer.
On Ubuntu Linux, when opening a device, the error 0xe0010050 occurs. After changing the baud rate for a UART device, the error 0xe0010050 occurs.	• Provide permissions by running `sudo chmod 777 /dev/ttyUSB*` (replace * by the correct port number)

Troubleshooting

Issue	Possible solution

On Ubuntu Linux, cannot connect to a device (e.g., CYW5557x/CYW89570).	Add access rights for the group to which the device belongs. To add access rights:
	• Run one of the following commands (depending on the port name) and check the group name:
	`ls -l /dev/ttyUSB*`
	`ls -l /dev/ttyS*`
	• Run the following command to check if the user has access rights:
	`getent group <group-name>`
	• Run the following command to add access rights:
	`sudo adduser <user-name> <group-name>`
	• Check if access rights are added successfully:
	`getent group <group-name>`
	• Reboot the computer.

On Ubuntu Linux, cannot load the Qt platform plugin “xcb”.	• Make sure that a monitor is connected to your computer.

On Ubuntu Linux or MacOS, when programming a device, the error “Permission denied” occurs.	Provide access rights to ChipLoad and cgs executables. To provide access rights:
	• Open Terminal.
	• Navigate to the folder where the ChipLoad and cgs executables are located. On Ubuntu Linux, they are located in the installation folder. On MacOS, they are located inside the application bundle, e.g. `airocdebugtoolgui.app/Contents/MacOS/`.
	• Execute the following commands:
	`sudo chmod 777 ChipLoad`
	`sudo chmod 777 cgs`

On an embedded Linux platform, cannot open a shared object file (e.g., libQt5SerialPort.so.5).	• Run the application using the `*.sh` script instead of running the executable directly.
On MacOS, a device cannot be opened.	• Remove the lock file by running `sudo rm -rf /var/spool/uucp/*` (replace `*` by the correct port name)
Connection between the selected devices cannot be created.	• Check if the selected devices need antennas. If the selected devices need antennas, connect antennas to the selected devices. • Make sure that the central and peripheral devices have different BD_ADDR. • Enable scan on the peripheral device using the **Write_Scan_Enable** command and send the **Inquiry** command to the central device to see if the peripheral device is detected. • For BIS, make sure that all necessary LE events are enabled.
Throughput values are lower than expected. Packet delays are larger than expected. A received WAV file sounds broken.	• For UART devices, increase the baud rate to 3000000 bps. • Uncheck the **Enable logging** checkbox on the log window. • Make sure that you are using correct connection settings for your devices. • Make sure that you are using the latest firmware for your devices.

Troubleshooting

Issue	Possible solution
SCO receive throughput test does not receive any data.	• You might need to send the **Write_SCO_PCM_Int_Param** command to both devices.
ISOC receive throughput test does not receive any data.	• Make sure that you have sent **LE_Setup_ISO_Data_Path** to both devices with correct **Data_Path_Direction** parameter values.
ISOC receive throughput test does not receive all SDU.	• Make sure that you are using correct connection settings for your devices.
On Ubuntu Linux, the GUI behaves unexpectedly when the user clicks Connect or Disconnect.	• Make sure that you have selected the correct port name. Some AIROC™ Bluetooth® devices have two UART ports. Usually, you have to select the port with the lower number.

Revision history

Revision	Date	Description
**	2018-12-18	New document
*A	2021-09-20	Document completely updated for new version of the tool.
*B	2022-09-30	Renamed the tool and updated version to 1.1.0.
*C	2023-04-13	Updated version to 1.2.0.
*D	2023-09-25	Updated version to 1.3.0.
*E	2024-04-11	Updated version to 1.4.0. Added What's New? dialog.
*F	2024-07-03	Updated version to 1.4.1.
*G	2024-10-29	Updated version to 1.4.2.
*H	2026-01-29	Updated version to 1.5.0.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc., and any use of such marks by Infineon is under license.

Edition 2026-01-29
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2026 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
002-25714 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.

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.