

CyBluetool user guide

About this document

Version

1.0.0

Scope and purpose

CyBluetool is a stand-alone, cross-platform tool for testing and debugging AIROC™ Bluetooth® devices. CyBluetool communicates with Bluetooth® devices through a Host/Controller Interface (HCI) and supports UART and USB physical interfaces. CyBluetool can:

- Send HCI commands to Bluetooth® controllers
- Receive HCI events from Bluetooth® controllers
- Program Bluetooth® controllers
- Run ACL and SCO throughput tests for Bluetooth® controllers

Intended audience

This document helps you understand how to use CyBluetool.

Document conventions

Convention	Explanation
Bold	Emphasizes hotkeys and the names of menu items, tabs, buttons, labels and windows
<i>Italics</i>	Denotes file names and paths
<code>Courier New</code>	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
BTP	A preset file used to program AIROC™ Bluetooth® controllers
GUI	Graphical user interface
HCI	Host/Controller interface
SCO	Synchronous connection-oriented link
UART	Universal asynchronous receiver-transmitter
USB	Universal serial bus

Table of contents

Table of contents

- 1 Introduction 3**
- 1.1 System requirements 3
- 1.2 Package contents 3
- 1.3 Supported devices 3
- 2 Installation Procedures..... 5**
- 2.1 Windows 5
- 2.2 Ubuntu Linux 5
- 2.3 mac OS..... 5
- 3 CyBluetool GUI application..... 6**
- 3.1 Open CyBluetool GUI application 6
- 3.2 Command line options..... 6
- 3.3 Main window description..... 7
- 3.4 Select a device 8
- 3.5 Send HCI commands 9
- 3.6 Receive HCI events 10
- 3.7 Reset a device 11
- 3.8 Create a connection between two devices 12
- 3.9 Create an LE connection between two devices 18
- 3.10 ACL throughput tests 20
- 3.11 Create an SCO connection 24
- 3.12 SCO throughput tests..... 26
- 3.13 Program a device..... 28
- 3.14 Log options 29
- 4 CyBluetool console application 30**
- 4.1 Open CyBluetool console application 30
- 4.2 Command line options..... 30
- 4.3 Open a device 31
- 4.4 Close a device 31
- 4.5 List HCI commands 31
- 4.6 Send HCI commands and receive responses 32
- 4.7 Get help..... 33
- 4.8 Exit CyBluetool console application..... 33

Introduction

1 Introduction

1.1 System requirements

CyBluetool supports the following operating systems:

- Windows 10 64-bit
- Ubuntu Linux 18.04 LTS, 20.04
- Mac OS X 10.15, 11

The following system configuration is recommended:

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

1.2 Package contents

The CyBluetool package contains:

- *cybluetoolgui* executable – the CyBluetool GUI application.
- *cybluetoolconsole* executable – the CyBluetool console application.
- Drivers used to communicate with Bluetooth® devices:
 - WICED UART driver – the FTDI VCP driver (Windows only).
 - BTWUSB driver – the USB driver (Windows only).
 - BTUSB driver (source code) – the USB driver (Linux only).
- *CyBluetoolMaintenanceTool* executable – the maintenance tool allowing to uninstall CyBluetool.

1.3 Supported devices

CyBluetool supports the following AIROC™ Bluetooth® devices.

- Stand-alone Bluetooth® chips:
 - CYW20702/5/89071
 - CYW20730
 - CYW20732/6/7
 - CYW20733
 - CYW20703/4/6/7/34/89072
 - CYW20735B0
 - CYW20735B1
 - CYW20819/20
 - CYW20719/21/39
- Combo chips:
 - CYW4330/43303/40183
 - CYW43340

Introduction

- CYW4343W/4343A1/43438
- CYW4343A2/43439
- CYW4339/89335
- CYW43455/4345C0
- CYW4350C0/4354
- CYW4354A2/4356
- CYW4356A3/4354
- CYW43569/43570
- CYW4359D0/89459/5459x
- CYW4373
- CYW4349B1/4359B1/89359
- CYW43012
- CYW5557x/CYW89570

Installation Procedures

2 Installation Procedures

This section describes how to install CyBluetool on different operating systems.

2.1 Windows

1. Run `cybluetool-1.0.0.<build>-win-x64.exe` as Administrator.
2. Follow the installation wizard instructions.

Note: `cybluetool-1.0.0.<build>-win-x64.exe` requires Administrator privileges to install drivers and Microsoft Visual C++ Redistributable.

2.2 Ubuntu Linux

1. Open a terminal and navigate to the folder where `cybluetool-1.0.0.<build>-linux-x64` is located.
2. Provide access rights to `cybluetool-1.0.0.<build>-linux-x64`. For example:

```
chmod +x cybluetool-1.0.0.<build>-linux-x64
```
3. Run the installer:

```
./cybluetool-1.0.0.<build>-linux-x64
```
4. Follow the installation wizard instructions.

2.2.1 Build and install the BTUSB driver

To connect to Bluetooth® devices using the USB transport on Ubuntu Linux, you need to build and install the BTUSB driver. The source code of the BTUSB driver is located in the `drivers/btusb` subdirectory of the CyBluetool 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 ~/Infineon/CyBluetool/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 mac OS

1. Run `cybluetool-1.0.0.<build>-mac-x86_64.dmg`.
2. Follow the installation wizard instructions.

CyBluetool GUI application

3 CyBluetool GUI application

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

3.1 Open CyBluetool GUI application

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

3.1.1 Windows

Navigate to the folder where CyBluetool is installed and run *cybluetoolgui.exe*.

3.1.2 Ubuntu Linux

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

```
cd ~/Infineon/CyBluetool
```

2. Make sure that access rights are provided to *cybluetoolgui.sh*. To provide access rights:

```
chmod +x cybluetoolgui.sh
```

3. Run *cybluetoolgui.sh*. For example:

```
./cybluetoolgui.sh
```

Note: Use *cybluetoolgui.sh* to open CyBluetool GUI. Do not run the *cybluetoolgui* executable directly.

3.1.3 mac OS

Navigate to the folder where CyBluetool is installed and run *cybluetoolgui*.

3.2 Command line options

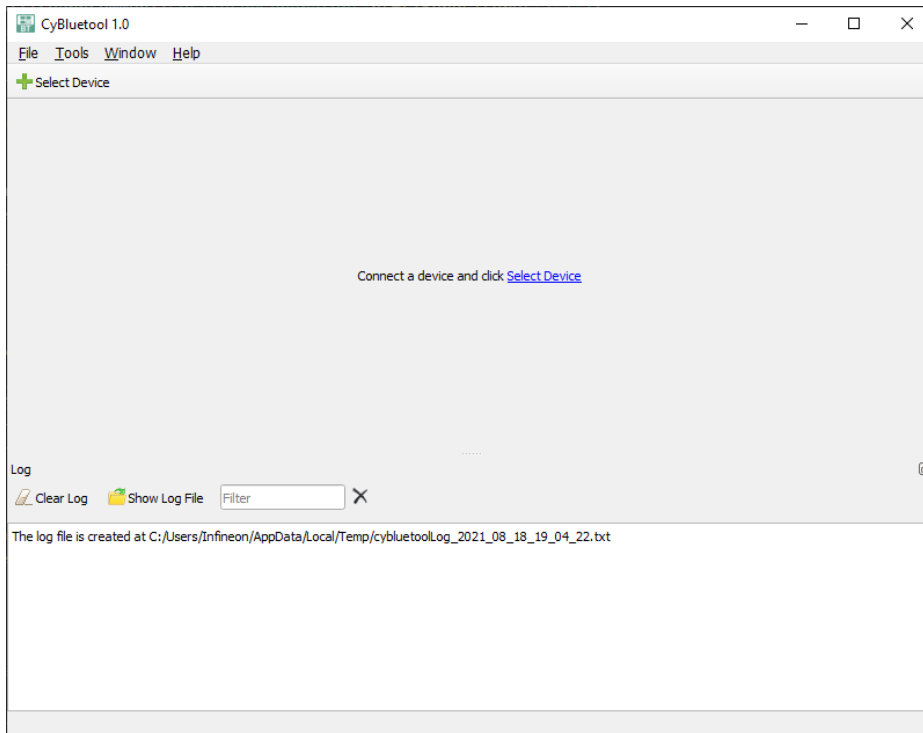
The following table lists command line options which you can use when opening CyBluetool GUI.

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-diagdef-file <filepath>	Overrides the default diagnostic definition file.
--override-lmpdef-file <filepath>	Overrides the default LPM definition file.
--override-hciacldf-file <filepath>	Overrides the default HCI ACL definition file.
--override-hciscodef-file <filepath>	Overrides the default HCI SCO definition file.
--override-hciisodef-file <filepath>	Overrides the default HCI ISO definition file.

CyBluetool GUI application

3.3 Main window description

When you open CyBluetool GUI, the main window opens.



The main window contains the following menus:

3.3.1 File menu

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

3.3.2 Tools menu

- **Clear Log (Ctrl+D)** – Clear the log window.
- **Show Log File in Explorer/Show Log File in Finder/Show Log File** – Show the location of the log file.
- **Log Options (Ctrl+L)** – Show the Log Options dialog allowing to set log options.
- **Program Device (Ctrl+P)** – Show the Program Device dialog allowing to program a Bluetooth® controller connected to the computer.

3.3.3 Window menu

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

3.3.4 Help menu

- **View Help (F1)** – Open User Guide.
- **About CyBluetool** – Show the About window displaying information about CyBluetool.

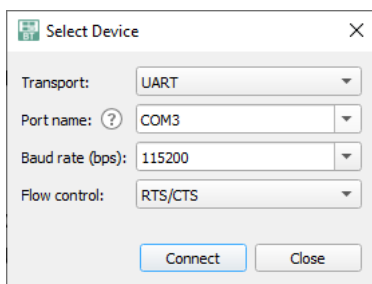
CyBluetool GUI application

3.4 Select a device

Before you can send HCI commands, receive HCI events, and run throughput tests for an AIROC™ Bluetooth® device, you need to select a device. To select a device:

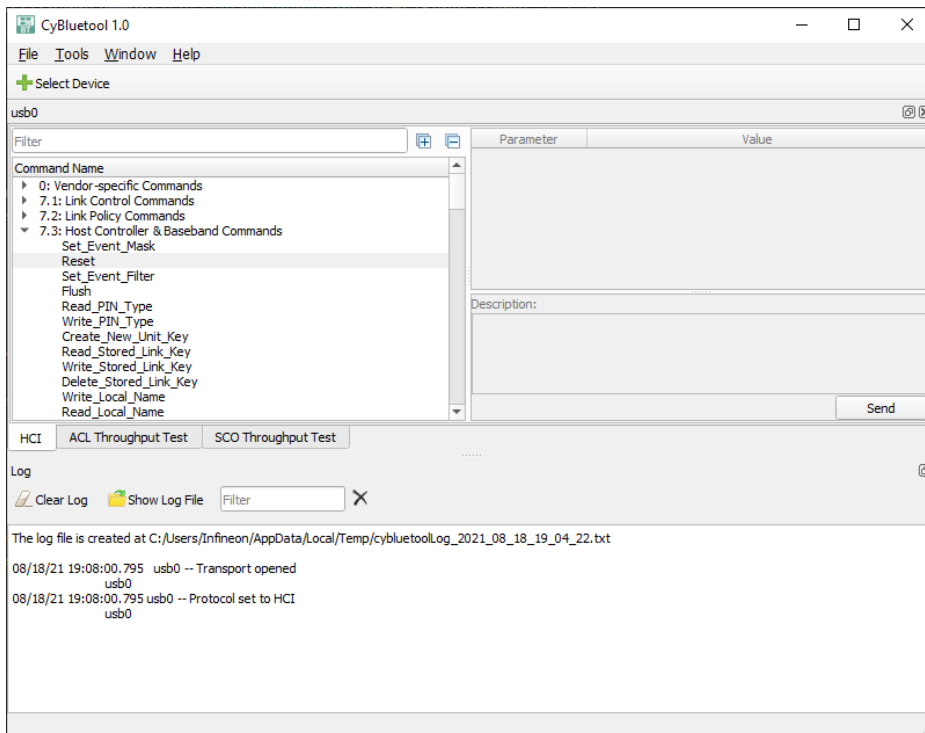
1. Connect the device to your computer and select **File > Select Device** on the main menu.
2. Click **Select Device** on the main toolbar.
3. Click the **Select Device** link on the main window or press **Ctrl+O**.

The Select Device dialog opens.



When the Select Device dialog opens, select **Transport** and **Port name**. If you select UART, select also **Baud rate** and **Flow control**.

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



To check if a 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: To select multiple devices, you can select use the same CyBluetool GUI instance. To select a new device, connect the device to your computer and select **File > Select Device** on the main menu, click **Select Device** on the main toolbar, or press **Ctrl+O**.

CyBluetool GUI application

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 have to use the port with lower number to select the device. Sometimes, the port that has to be used to select the device has "HCI" in port name.

3.4.2 Select port name for USB

If you select USB, the **Port name** drop-down list is not auto-populated and the port name must be entered.

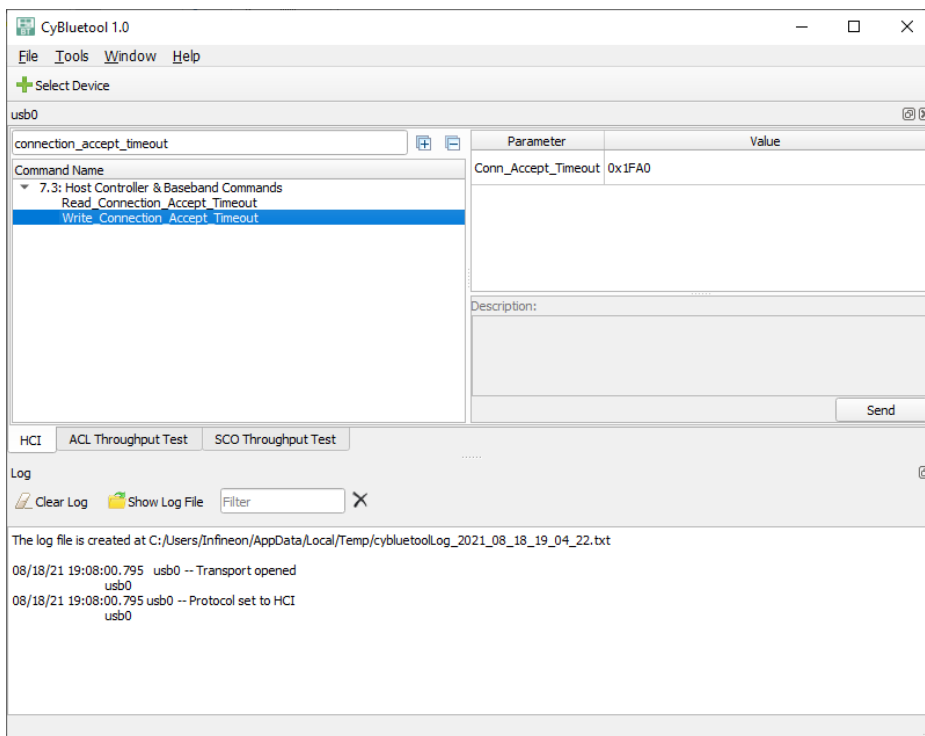
- On Windows, port name has the form **usb<number>**. Port name of the first device that is connected to the computer is usually usb0. Port names of other devices are usually subsequent: usb1, usb2 etc.
- On Linux, port name has the form **btusb<number>**. Port name of the first device that is connected 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 port name of a USB device can change after programming the device.

3.5 Send HCI commands

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

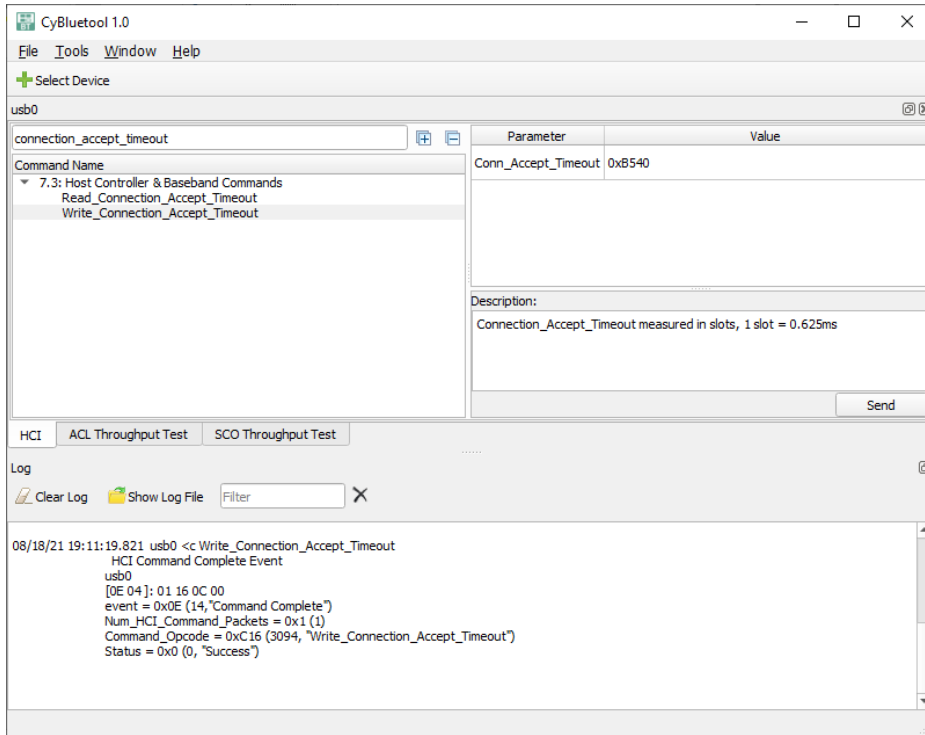
If the selected command has parameters, the list of command parameters is displayed on the right of the device tab or dockable window.



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

CyBluetool GUI application

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



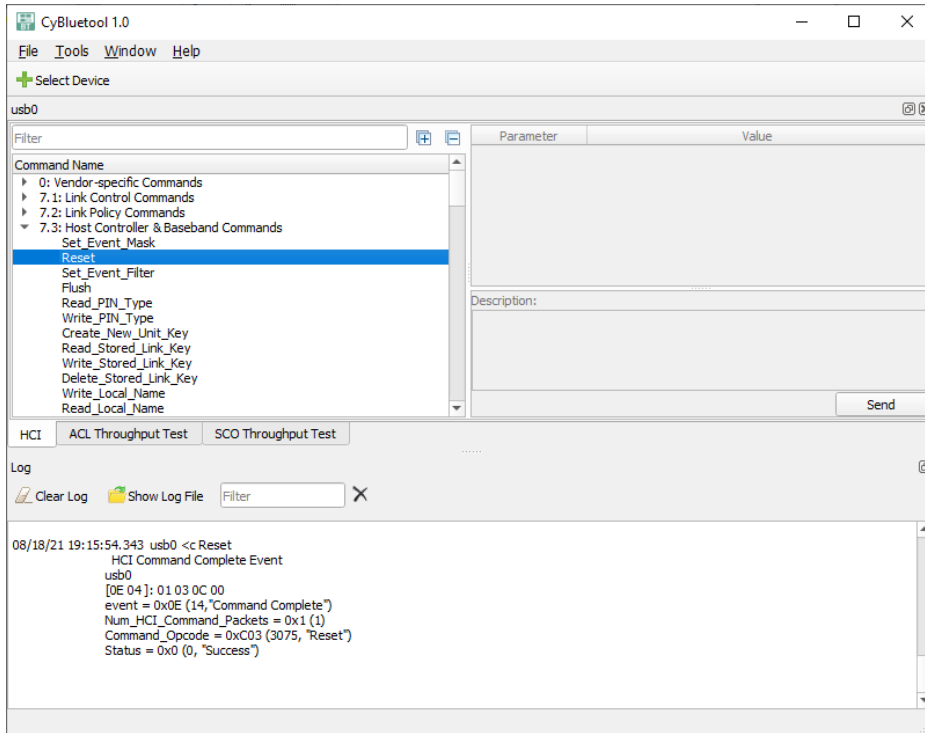
3.6 Receive HCI events

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

CyBluetool GUI application

3.7 Reset a device

To reset a selected device, send the **Reset** command.



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

CyBluetool GUI application

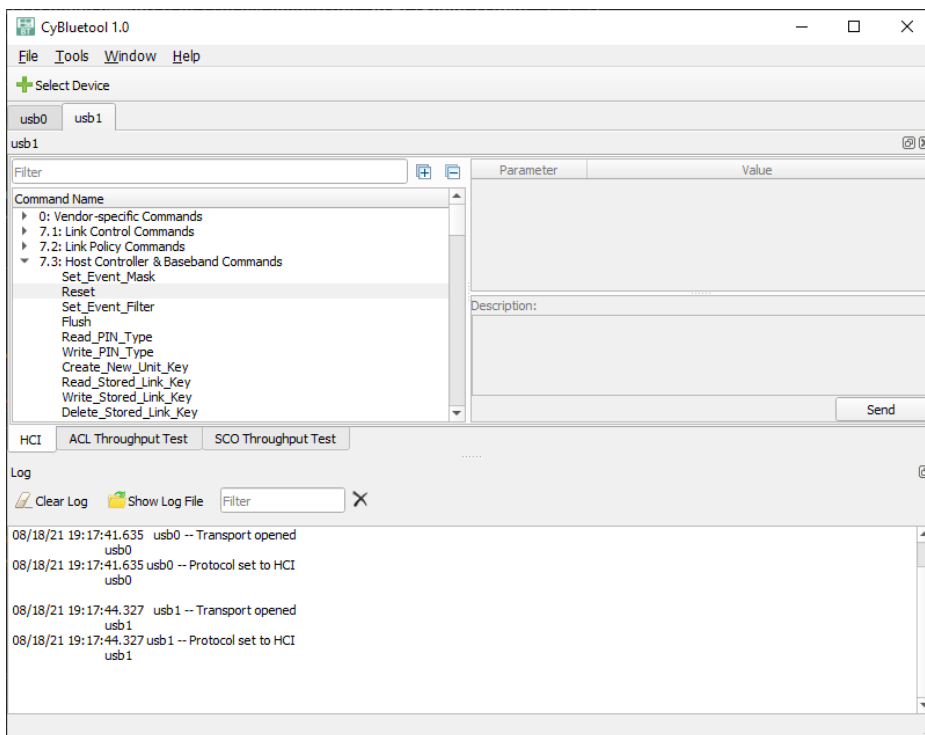
3.8 Create a connection between two devices

To create a connection between two AIROC™ Bluetooth® devices:

1. Connect the devices to your computer.
2. [Select the devices](#) using CyBluetool GUI.
3. [Increase connection accept timeout for the slave device](#) (recommended).
4. [Enable scan for the slave device](#).
5. [Send the **Create Connection** command to the master device](#).

3.8.1 Select the devices

To select devices using CyBluetool GUI, connect the devices to your computer and follow the instructions from the [Select a device](#) section to select the devices. If the devices are connected to the same computer, to select the devices, you can use the same CyBluetool GUI instance.



Note: If you select devices using the same CyBluetool GUI instance, before creating a connection, it is recommended to send the **Read_BD_ADDR** command to both devices. This will auto-populate the device address on next steps of creating the connection.

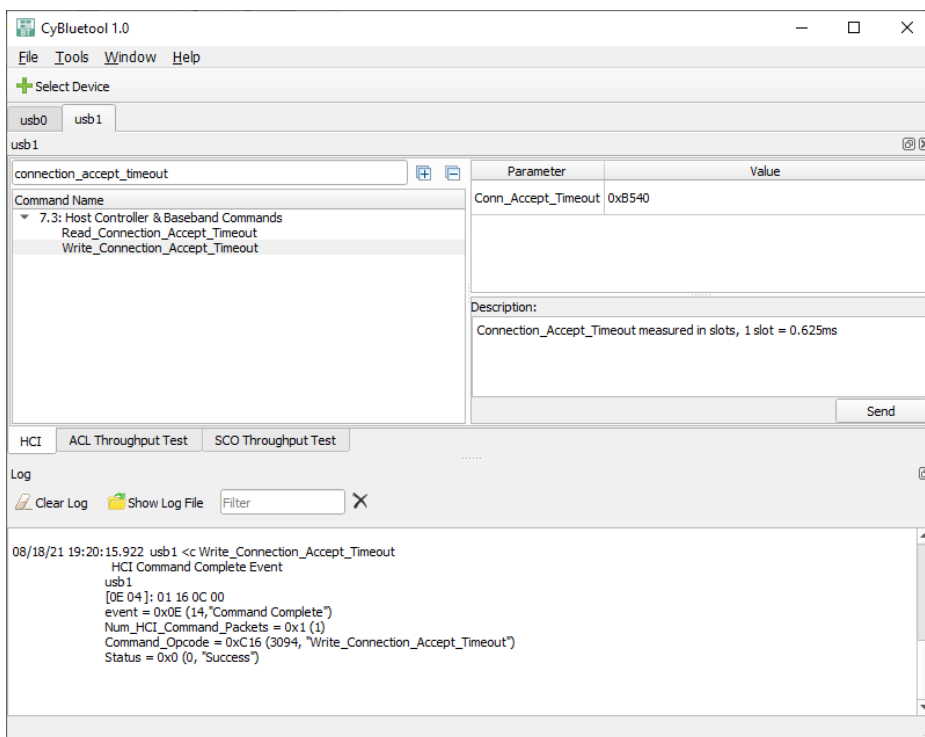
CyBluetool GUI application

3.8.2 Increase connection accept timeout for the slave device

Before creating a connection between the selected devices, it is recommended to increase the connection accept timeout for the slave device. To increase the connection:

1. Accept the timeout for the slave device.
2. Select the **HCI** tab on the device tab or dockable window.
3. Select the **Write_Connection_Accept_Timeout** command.
4. Set the **Conn_Accept_Timeout** parameter value.
5. Click **Send**.

The maximum value of the **Conn_Accept_Timeout** parameter is 0xB540.

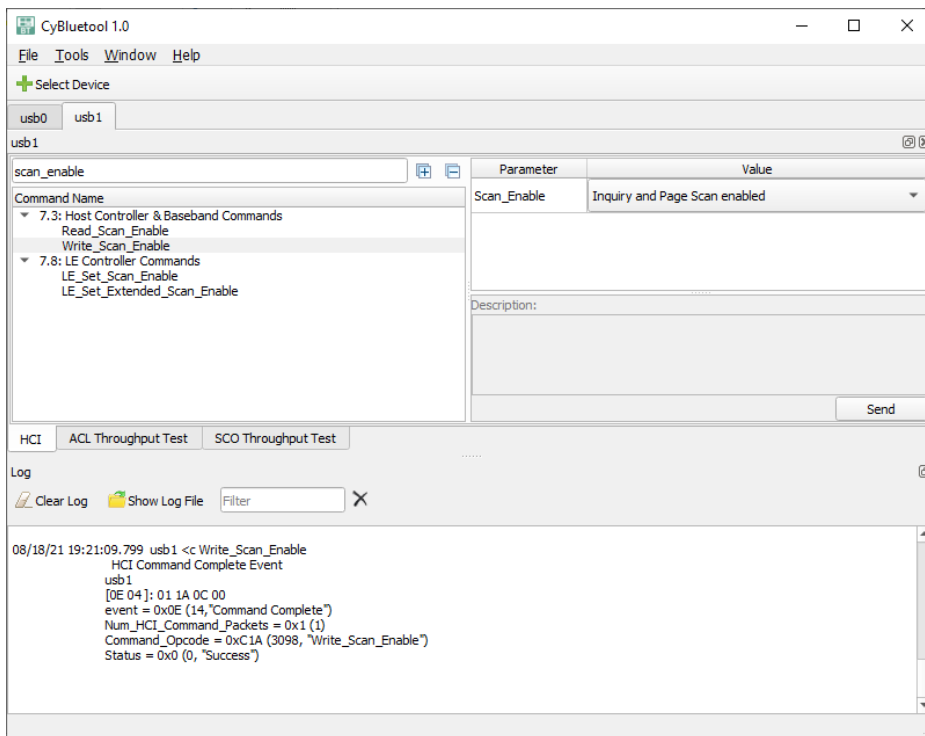


CyBluetool GUI application

3.8.3 Enable scan for the slave device

Before creating a connection between the selected devices, enable scan for the slave device. To enable scan for the slave device:

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



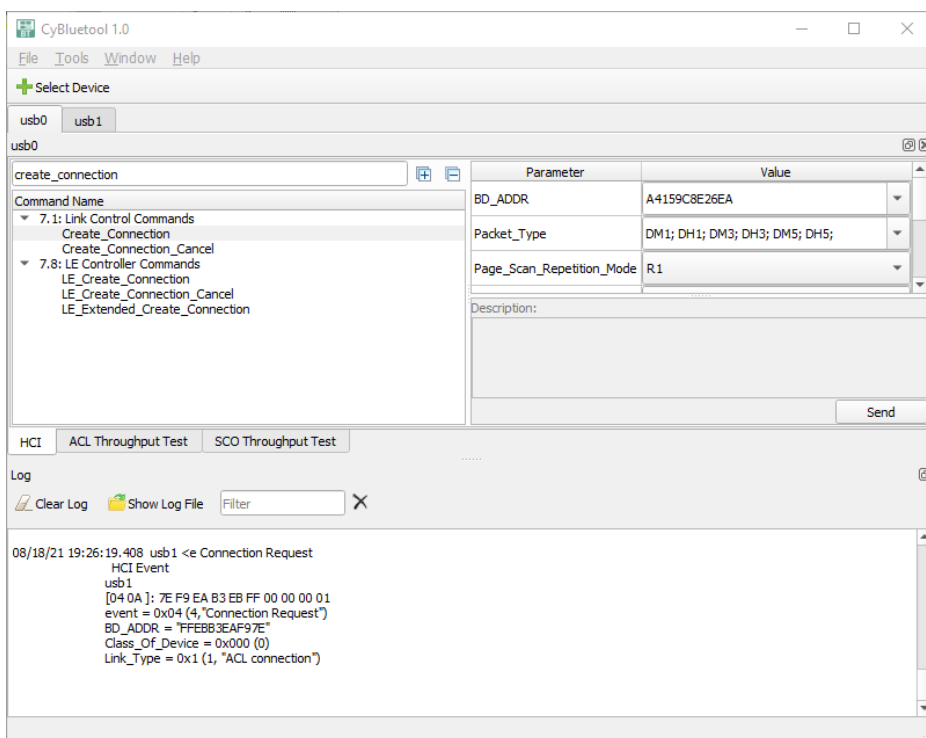
CyBluetool GUI application

3.8.4 Send Create_Connection to the master device

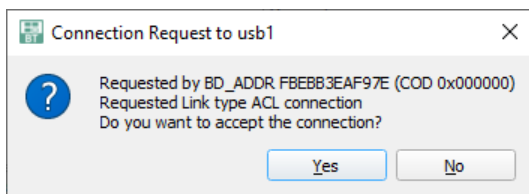
To create a connection between the selected devices, send the **Create_Connection** command to the master device. To send the **Create_Connection** command to the master device:

1. Select the **HCI** tab on the device tab or dockable window.
2. Select the **Create_Connection** command.
3. Set the command parameters.
4. Click **Send**.

Note: The **BD_ADDR** parameter of the **Create_Connection** command must be set to the address of the slave device. Use the **Read_BD_ADDR** command to get the address of the slave device.



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



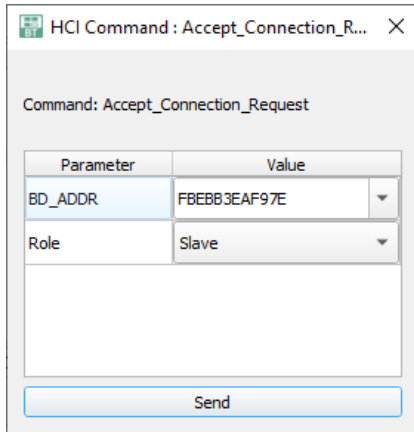
Note: Some AIROC™ Bluetooth® devices may need antennas. If the connection request dialog is not opened, check if the devices need antennas and, if yes, check if the antennas are connected to the devices.

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

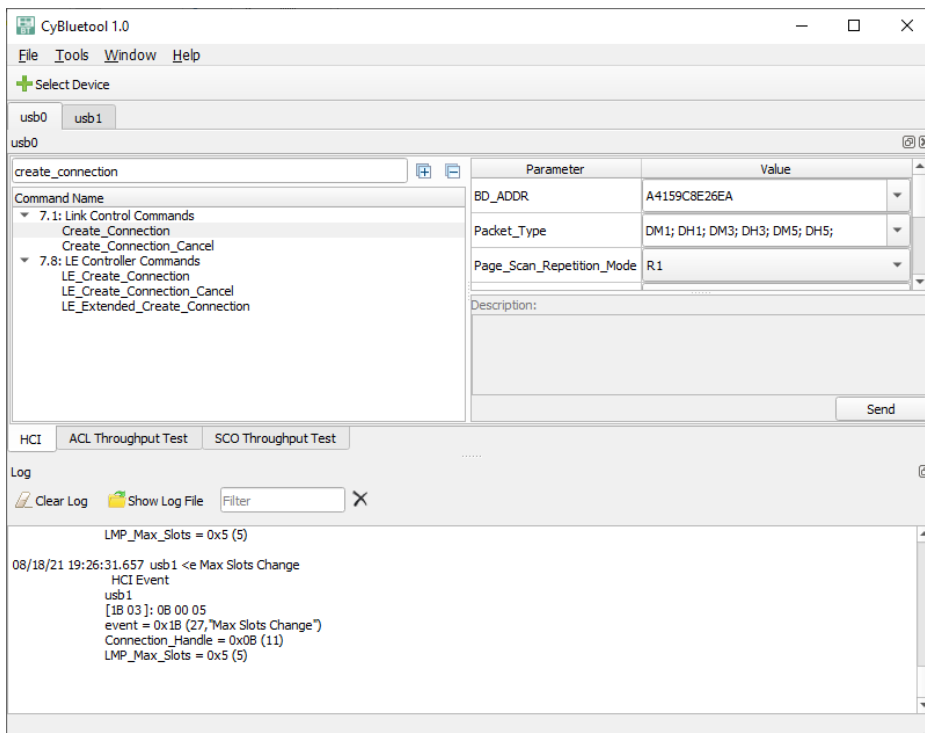
CyBluetool GUI application

On the next dialog that opens, set parameters of the **Accept_Connection_Request** command that must be sent to the slave device within the connection accept timeout, and click **Send**.

*Note: The **BD_ADDR** parameter of the **Accept_Connection_Request** command is auto-populated and must be set to the address of the master device. Use the **Read_BD_ADDR** command to get the address of the master device.*



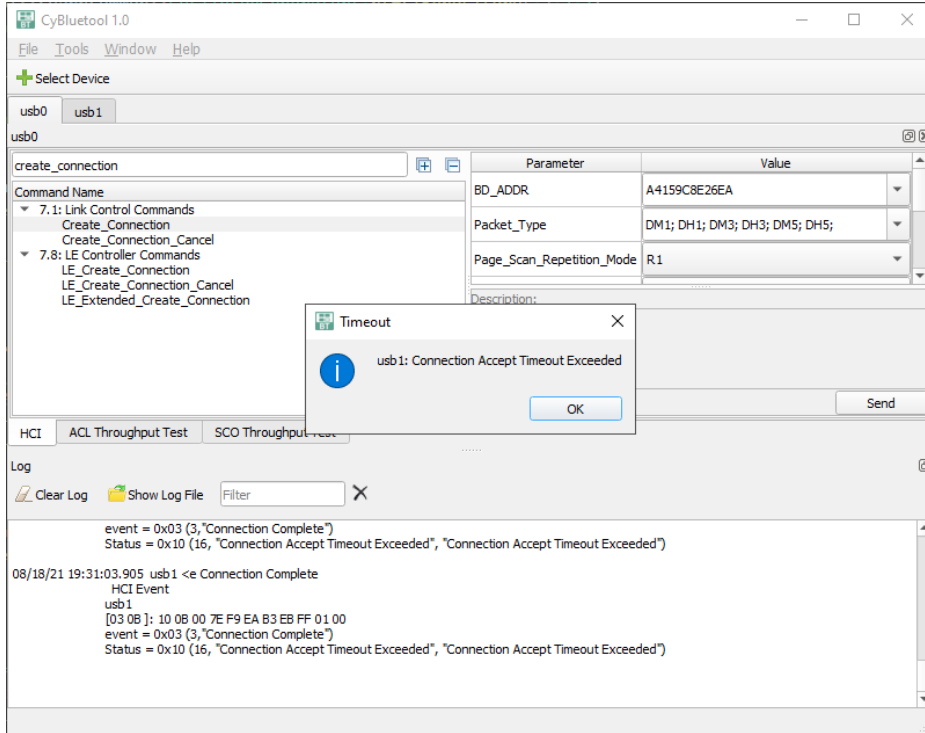
If the connection is created successfully, the **Max Slots Change** HCI event is generated. Event information is printed in the log window. The **Connection_Handle** parameter of the **Max Slots Change** event is the connection handle of the created connection.



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

CyBluetool GUI application

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 that will end with the **Invalid HCI Command Parameters** status.



CyBluetool GUI application

3.9 Create an LE connection between two devices

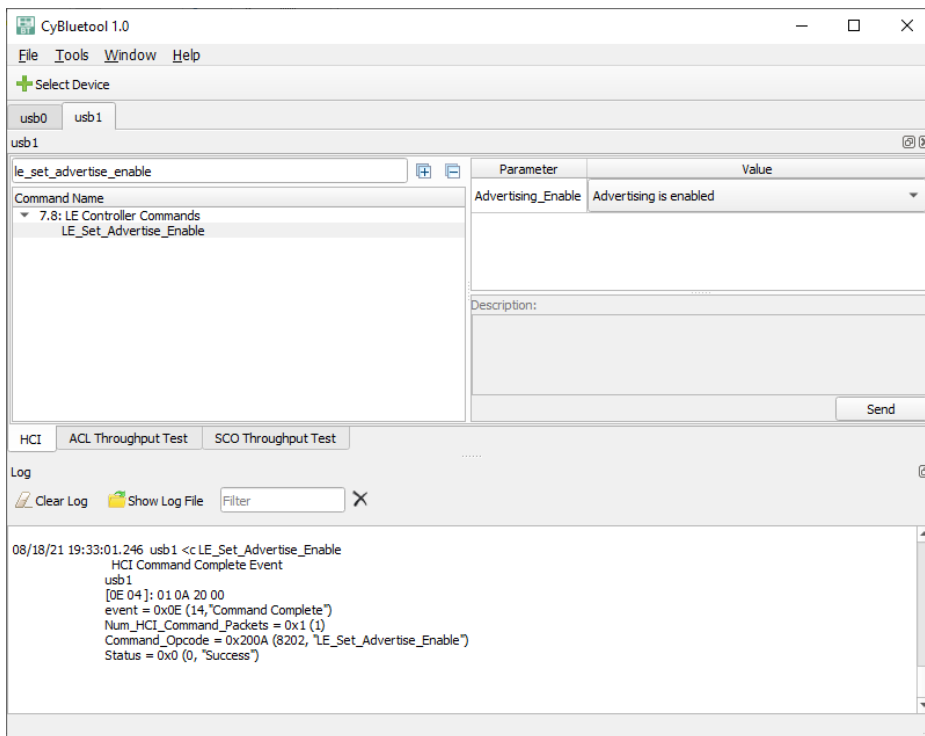
To create an LE connection between two AIROC™ Bluetooth® devices:

1. [Enable advertising on the slave device.](#)
2. [Send the LE Create Connection command to the master device.](#)

3.9.1 Enable advertising on the slave device

Before creating an LE connection between the selected devices, enable advertising for the slave device. To enable advertising for the slave device:

1. Select the **HCI** tab on the device tab or dockable window.
2. Select the **LE_Set_Advertise_Enable** command.
3. Set the **Advertising_Enable** parameter to **Advertising is enabled**.
4. Click **Send**.



Note: You need to enable advertising on the slave device every time when you create an LE connection.

CyBluetool GUI application

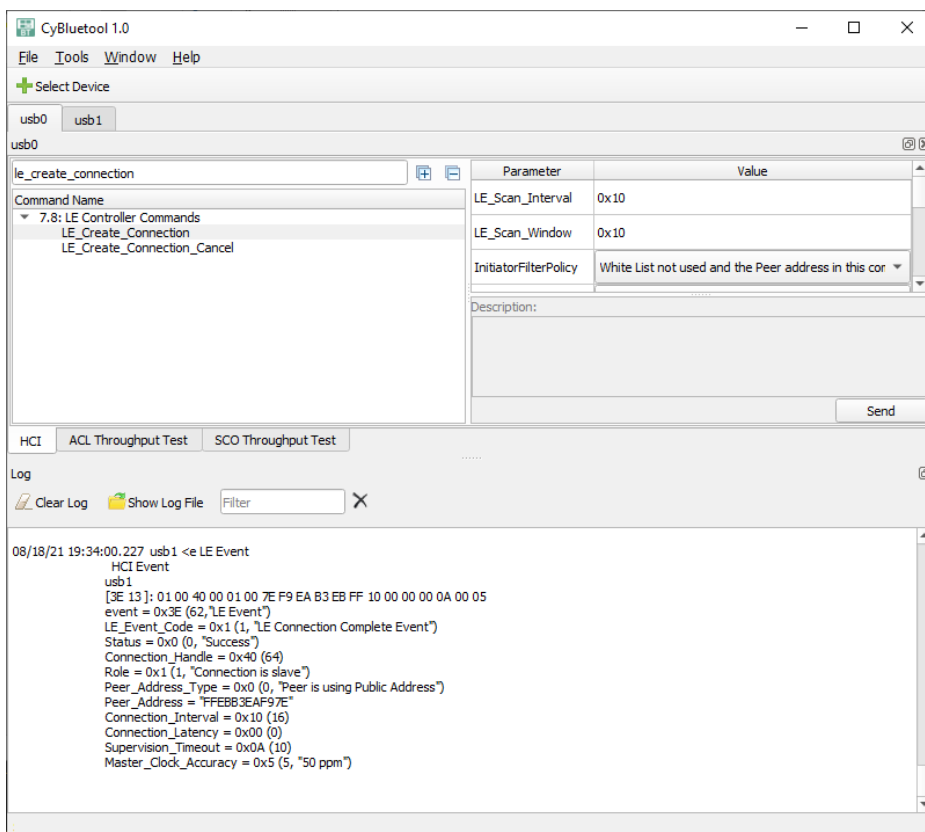
3.9.2 Send LE_Create_Connection to the master device

To create an LE connection between the selected devices, send the **LE_Create_Connection** command to the master device. To send the **LE_Create_Connection** command to the master device:

1. Select the **HCI** tab on the device tab or dockable window.
2. Select the **LE_Create_Connection** command.
3. Set the command parameters.
4. Click **Send**.

Note: The **Peer_Address** parameter of the **Create_Connection** command must be set to the address of the slave device. Use the **Read_BD_ADDR** command to get the address of the slave device.

Note: 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.



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

CyBluetool GUI application

3.10 ACL throughput tests

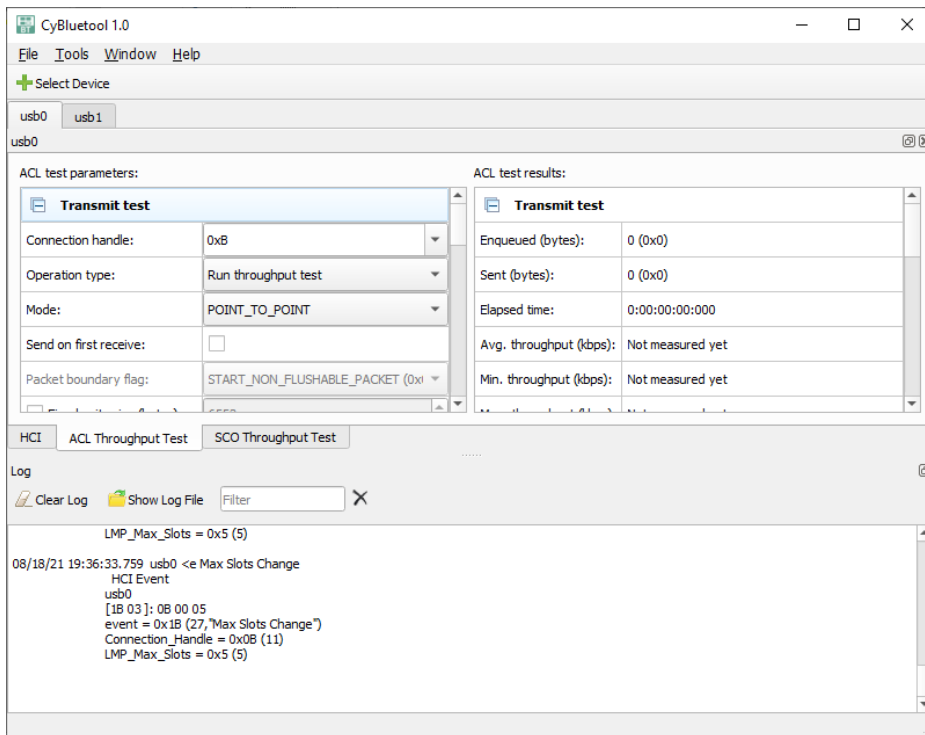
When a connection between two devices is created, you can run ACL throughput tests to check how fast ACL data is transferred between the devices.

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

3.10.1 ACL transmit test

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

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



The following table lists parameters of the ACL transmit test.

Parameter	Description	Values
Connection handle	The connection handle of the connection.	A valid connection handle. When the connection is created using CyBluetool GUI, the drop-down list with connection handles is auto-populated.
Operation type	Defines the operation type.	Run throughput test Send ACL data
Mode	Defines the data transfer mode.	POINT_TO_POINT ACTIVE_SLAVE_BROADCAST PARKED_SLAVE_BROADCAST
Send on first receive	If checked, data transmission starts only when the first byte is received from another device.	Checked or not checked

CyBluetool GUI application

Parameter	Description	Values
Packet boundary flag	Defines the packet boundary flag. Available only if Operation type is set to Send ACL data .	START_NON_FLUSHABLE_PACKET CONTINUATION_PACKET START_OF_PACKET
Fixed write size	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	Defines the interval between data blocks. If not set, the next block is sent when the Number of Completed Packets event is generated by the controller.	Integer values between 1 and 65535
Data source	Defines the source of the data that will be transmitted. Enabled only if Operation type is set to Run throughput test .	Data pattern Data file
Data pattern	Sets the data that will be transmitted. Enabled only if Data source is set to Data pattern .	If ASCII is not checked, a space-separated sequence of bytes. If ASCII is checked, a valid ASCII string.
Data file	A file from which the data to transmit will be read. Enabled only if Data source is set to Data file .	The absolute path to an existing text file.
Count	Defines how many times the data pattern or data file must be transmitted.	Integer values between 1 and 2147483647
Total bytes (read-only)	The total number of bytes to be transmitted.	Integer values
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. Parameters of the transmit and receive tests are synchronized.	A valid port name of a selected device

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

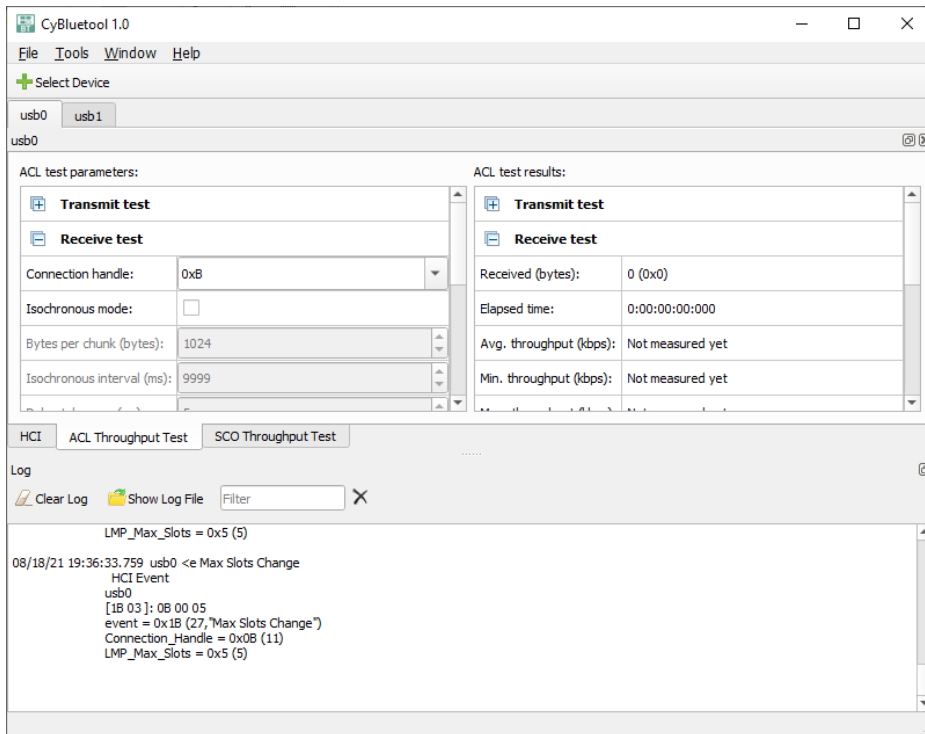
Parameter	Description
Enqueued (bytes)	The number of bytes sent by CyBluetool to the controller.
Sent (bytes)	The number of bytes completed (transmitted or flushed) by the controller.
Elapsed time	Duration of the test.
Average throughput (kbps)	The total number of bytes completed (transmitted or flushed) by the controller divided by time elapsed since the first byte was completed (in kilobytes per second)
Minimum throughput (kbps)	The minimum throughput value in kilobytes per second. Throughput is calculated as the number bytes newly completed (transmitted or flushed) by the controller divided by the time since the previous Number of Completed Packets event.
Maximum throughput (kbps)	The maximum throughput value in kilobytes per second. Throughput is calculated as the number bytes newly completed (transmitted or flushed) by the controller divided by the time since the previous Number of Completed Packets event.
First byte time	Indicates when the first byte was sent to the controller.
Last byte time	Indicates when the last byte was sent to the controller.

CyBluetool GUI application

3.10.2 ACL receive test

The ACL receive test checks how fast ACL data is received by a Bluetooth® controller. The receive test compares the data received by the controller and the data that is expected to be received. To run the ACL receive test:

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



The following table lists parameters of the ACL receive test.

Parameter	Description	Values
Connection handle	Connection handle of the ACL connection.	A valid connection handle. When the ACL connection is created using CyBluetool GUI, the dropdown list of connection handles is auto-populated.
Isochronous mode	Defines the expected interval between data blocks. Should be enabled if Isochronous interval is enabled for the transmit test.	Checked or unchecked.
Bytes per chunk	Enabled only if Isochronous mode is checked. Defines the expected size of a data block.	Integer values between 1 and 65535.
Isochronous interval	Enabled only if Isochronous mode is checked. Defines the expected interval between data blocks.	Integer values between 1 and 65535.
Delay tolerance	Enabled only if Isochronous mode is checked. Defines the expected tolerance for Isochronous interval .	Integer values between 1 and 65535.
Data source	Defines the source to get the expected data to receive from.	Data pattern Data file

CyBluetool GUI application

Parameter	Description	Values
Data pattern	Sets the expected data to receive. Enabled only if Data source is set to Data pattern .	If ASCII is not checked, a space-separated sequence of bytes. If ASCII is checked, a valid ASCII string.
Data file	A file from which the expected data to receive will be read. Enabled only if Data source is set to Data file .	The absolute path to an existing text file.
Count	Defines how many times the data pattern or data file is expected to be received.	Integer values between 1 and 2147483647.
Total bytes (read-only)	The total number of bytes to be transmitted.	
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 a 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 received by the controller.
Elapsed time	Duration of the test.
Average throughput (kbps)	The total number of bytes received by the controller divided by time elapsed since the first byte was completed (in kilobytes per second)
Minimum throughput (kbps)	The minimum throughput value in kilobytes per second. Throughput is calculated as the number bytes newly received by the controller divided by the time since the previous ACL data event.
Maximum throughput (kbps)	The maximum throughput value in kilobytes per second. Throughput is calculated as the number bytes newly received by the controller divided by the time since the previous ACL data event.
First byte time	Indicates when the first byte was received by the controller.
Last byte time	Indicates when the last byte was received by the controller.
Bit errors	The total number of bits that were received incorrectly.
Dropped bytes	The total number of bytes that are missed in the received data.
Delayed packets	The total number of packets that were delayed by more than Delay tolerance value.
Average delay	The average delay of a packet.
Maximum delay	The maximum delay of a packet.

CyBluetool GUI application

3.11 Create an SCO connection

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

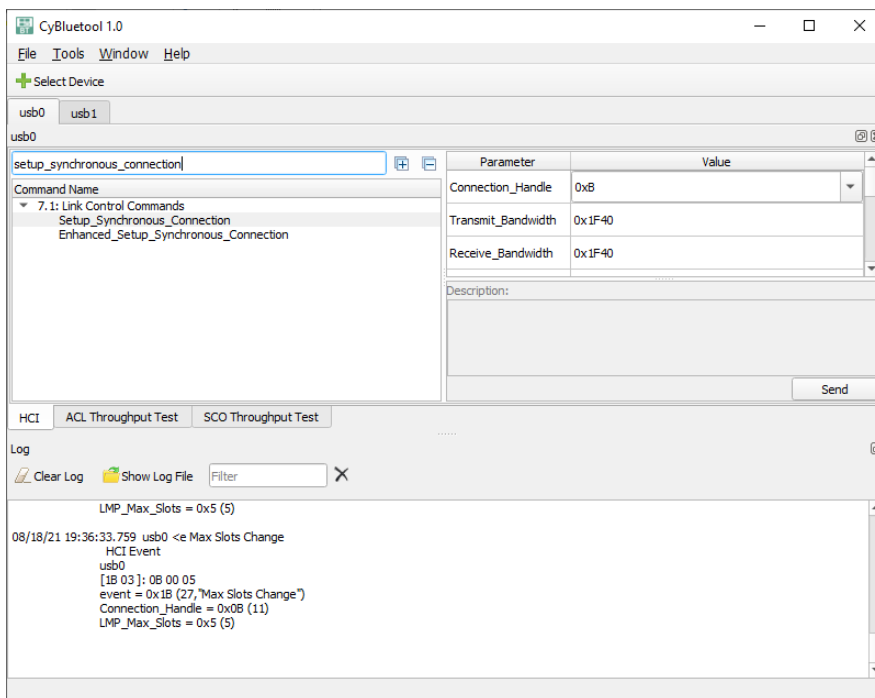
1. [Create a connection between the devices.](#)
2. [Send the **Setup Synchronous Connection** command to the master device.](#)

3.11.1 Send Setup_Synchronous_Connection to the master device

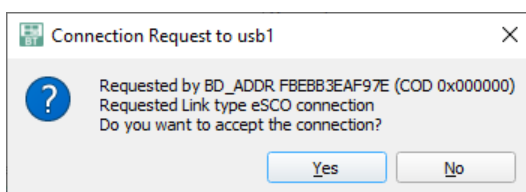
When the ACL connection is created, to create an SCO connection, send the **Setup_Synchronous_Connection** command to the master device. To send the **Setup_Synchronous_Connection** command to the master device, on the device tab or dockable window:

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

Note: The **Connection_Handle** parameter must be set to the connection handle of the ACL connection. Other parameter values must correspond to the audio data that will be transmitted.



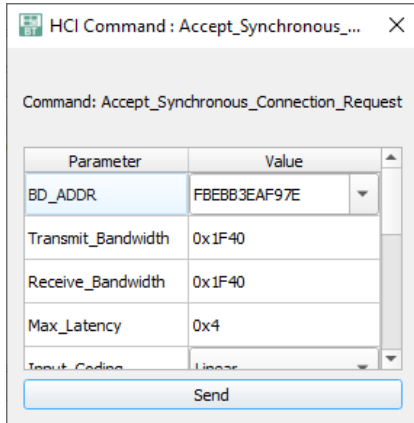
When the slave device gets the synchronous connection request from the master device, a dialog opens to ask if you want to accept the connection request.



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

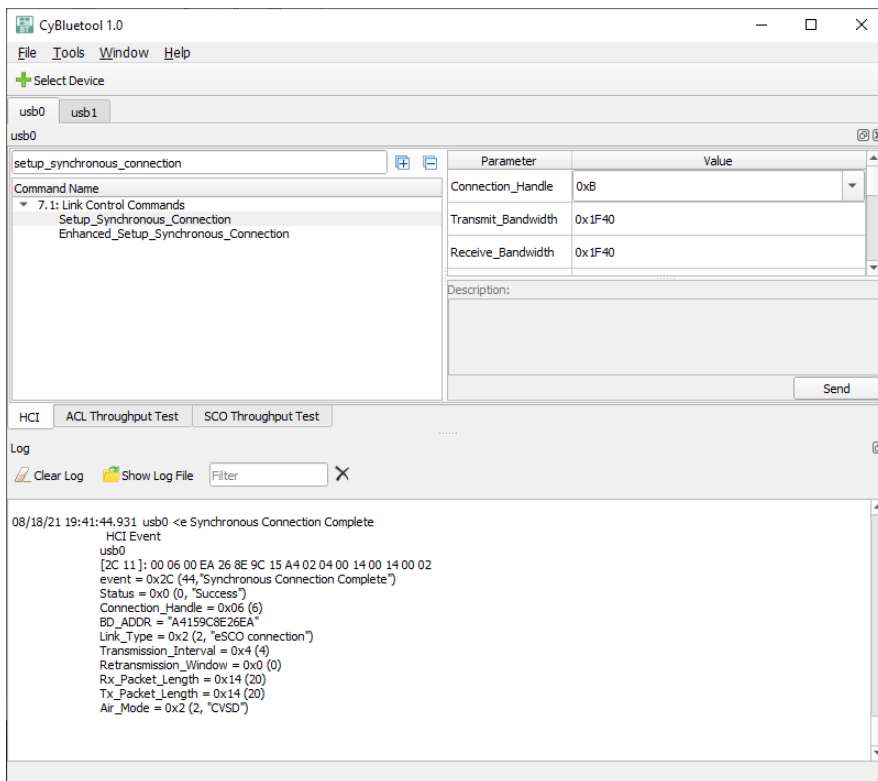
CyBluetool GUI application

On the next dialog that opens, set parameters of the **Accept_Synchronous_Connection_Request** command that has to be sent to the slave device within the connection accept timeout.



Note: *The parameters of the **Accept_Synchronous_Connection_Request** command must match the parameters of the synchronous connection between the devices. The **BD_ADDR** parameter of the **Accept_Synchronous_Connection_Request** command must be set to the address of the master device. Use the **Read_BD_ADDR** command to get the address of the master device.*

If the SCO connection is created successfully, the **Synchronous Connection Complete** HCI event is generated. 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: *When the synchronous connection accept request is received, if you click **No** to reject the connection request, the **Reject_Synchronous_Connection_Request** command must be sent to the slave device within the connection accept timeout.*

CyBluetool GUI application

3.12 SCO throughput tests

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

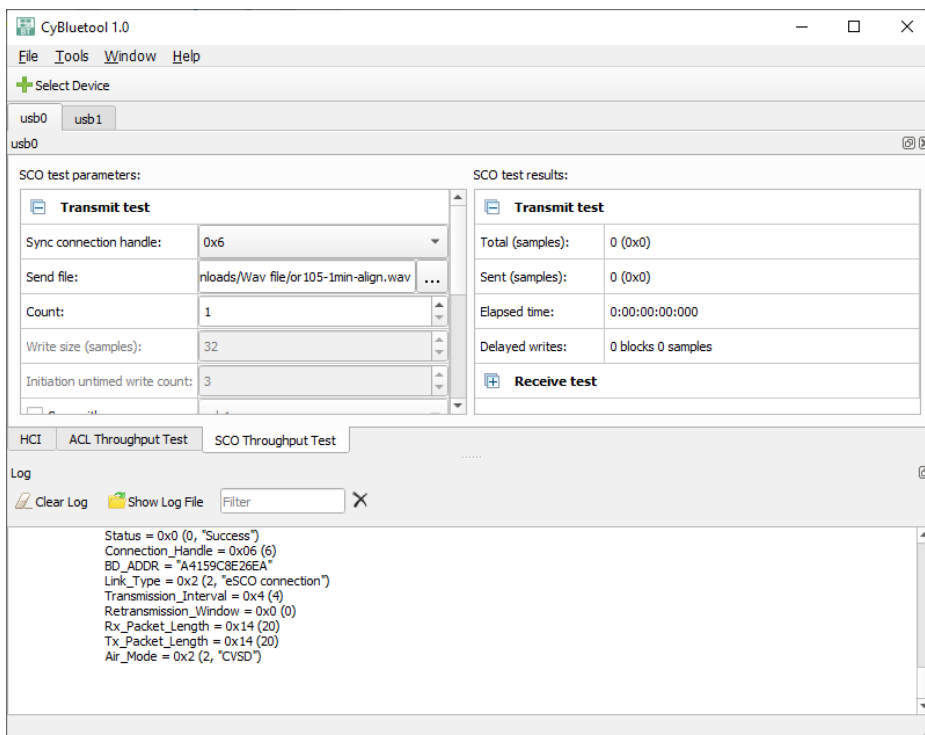
To run an SCO throughput test, on a device tab or dockable window, select the **SCO Throughput Test** tab.

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

3.12.1 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. Expand the **Transmit test** section of the **SCO test parameters** column.
2. Set test parameters.
3. Click **Start test**.



The following table lists parameters of the SCO transmit test.

Parameter	Description	Values
Sync connection handle	Connection handle of the SCO connection.	A valid connection handle. When the ACL connection is created using CyBluetool GUI, the dropdown list of connection handles is auto-populated.
Send file	A WAV file containing the data to send.	The absolute path to an existing WAV file.
Count	Defines how many times the file must be sent.	POINT_TO_POINT ACTIVE_SLAVE_BROADCAST PARKED_SLAVE_BROADCAST

CyBluetool GUI application

Parameter	Description	Values
Write size	Defines how many samples in a transaction slot can be filled. In case of USB transport, enabled only on Linux.	Integer values between 1 and 127.
Initiation untimed write count	The number of blocks of Write size to write to the device at the time of test initiation.	Integer values between 1 and 255.
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.	A valid port name of a selected device.

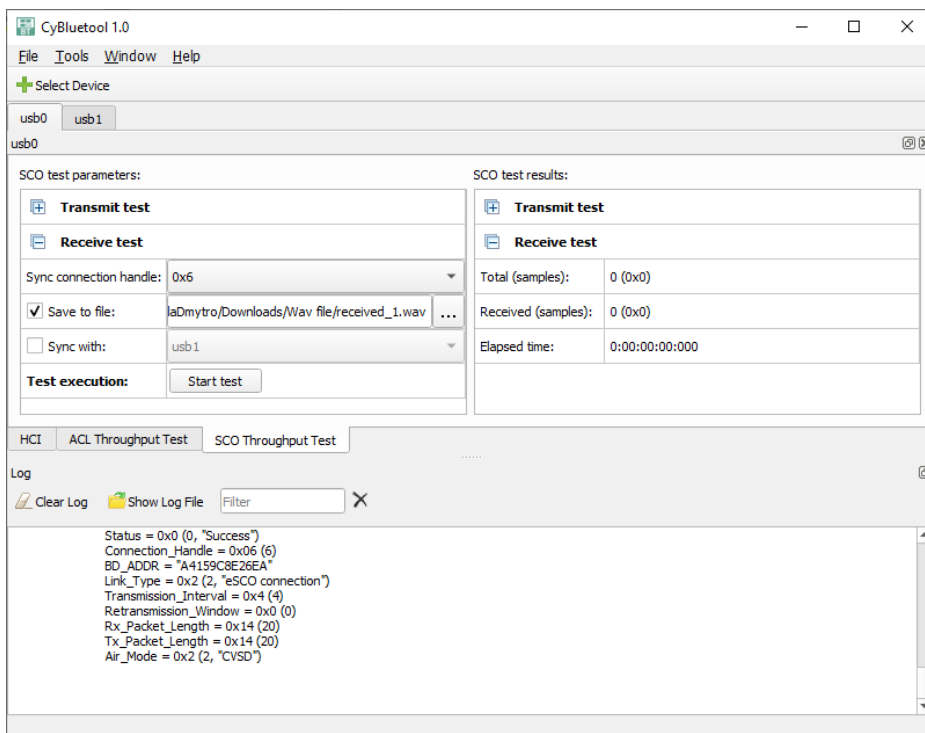
To check transmit test results, expand the Transmit test section of the SCO test results column. The following table lists test result parameters.

Parameter	Description
Total (samples)	The total number of samples to be transmitted.
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.

3.12.2 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. Expand the **Receive test** section of the **SCO test parameters** column.
2. Set test parameters.
3. Click **Start test**.



The following table lists parameters of the SCO receive test.

CyBluetool GUI application

Parameter	Description	Values
Sync connection handle	Connection handle of the SCO connection.	A valid connection handle. When the ACL connection is created using CyBluetool GUI, the dropdown list of connection handles is auto-populated.
Save to file	The file to save the received data to.	The absolute path to WAV file to create.
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 a selected device.

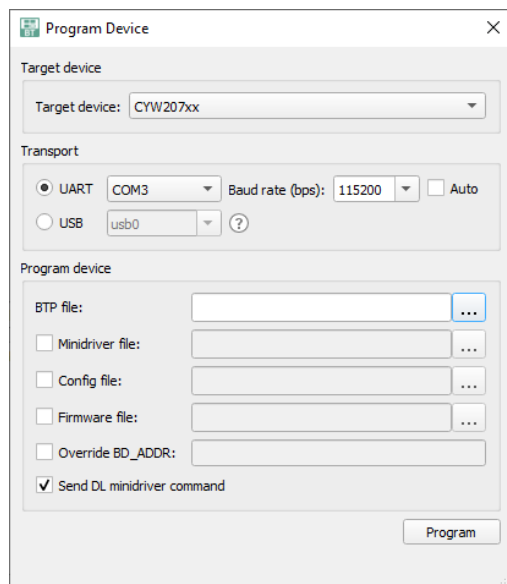
To check test results, expand the **Receive test** section of the **ACL test results** column. The following table lists test result parameters and explains their meanings.

Parameter	Description
Total (samples)	The number of samples to receive. Defined if the test is synchronized with a transmit test using the same CyBluetool GUI instance.
Received (samples)	The number of received samples.
Elapsed time	Duration of the test.

Note: The SCO receive test always has to be stopped manually.

3.13 Program a device

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



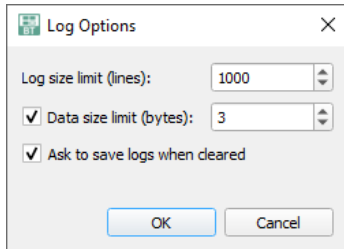
Select **Target device** to match the device that you want to program and set parameter values. Different target devices can have different parameters. When the parameters are set, click **Program** to program the device.

Note: If the device is selected in CyBluetool GUI when the programming starts, the device is disconnected from CyBluetool. In case of the USB transport, the port name of the programmed device can change after programming.

CyBluetool GUI application

3.14 Log options

CyBluetool GUI prints information about HCI commands, HCI events and throughput tests to the log window and to the log file. To edit 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 CyBluetool 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 the log files.

CyBluetool console application

4 CyBluetool console application

This section describes how to open and use the CyBluetool console application.

4.1 Open CyBluetool console application

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

4.1.1 Windows

Navigate to the folder where CyBluetool is installed and run *cybluetoolconsole.exe*.

4.1.2 Ubuntu Linux

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

```
cd ~/Infineon/CyBluetool
```

2. Make sure that access rights are provided to *cybluetoolconsole.sh*. To provide access rights:

```
chmod +x cybluetoolconsole.sh
```

3. Then, to open the CyBluetool console application, run *cybluetoolconsole.sh*. For example:

```
./cybluetoolconsole.sh
```

Note: Use *cybluetoolconsole.sh* to open the CyBluetool console application. Do not run the *cybluetoolconsole* executable directly.

4.1.3 mac OS

Open a terminal, navigate to the folder where CyBluetool is installed and run *cybluetoolconsole*. For example:

```
/Applications/CyBluetool/cybluetoolgui.app/Contents/MacOS/cybluetoolconsole
```

4.2 Command line options

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

Option	Description
<code>--help, -h, -?</code>	Displays help on command line options.
<code>--help-all</code>	Displays help on command line options including Qt specific options.
<code>--disable-console-log</code>	Disables logging to the console.
<code>--enable-log-server</code>	Enables logging to the log server.
<code>--server-port <server-port></code>	Log server port number.

CyBluetooth console application

4.3 Open a device

Before you can send commands to an AIROC™ Bluetooth® device, you have to open the device using the CyBluetooth 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 CyBluetooth 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 CyBluetooth console application, to disconnect from the device, type close. For example:

```
(usb0)$ close  
$
```

Then, you can open another device.

4.5 List HCI commands

When the device is opened using the CyBluetooth 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:

```
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:

```
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:

```
list
```

CyBluetool console application

4.6 Send HCI commands and receive responses

To send an HCI command without parameters:

```
send <command name>
```

If the command generates HCI events, event 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
[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 parameters of an HCI command:

```
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:

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

For example:

```
(usb0)$ send write_bd_addr --bd_addr FFFFFFFFFFFFFFFF
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 = "FFFFFFFFFFFFFFF"

(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")
```


CyBluetool console application

4.7 Get help

To get help on the commands of CyBluetool 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
```

4.8 Exit CyBluetool console application

To exit the CyBluetool console application, type `exit`.

Revision history

Revision history

Revision	Date	Description
**	12/18/2018	New document
*A	09/20/2021	Document completely updated for new version of CyBluetool.

Trademarks

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

Edition 2021-09-22

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2021 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

002-25714 Rev. *A

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffheitsgarantie").

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

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

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

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

WARNINGS

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

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