

FRAUNHOFER-INSTITUT FOR INTEGRATED CIRCUITS IIS

LC3PLUS HIGH-RESOLUTION A2DP DEMO GUIDE

Stream LC3plus High-Resolution Audio Over Bluetooth A2DP Using Raspberry Pi

This guide demonstrates how to set up and evaluate audio streaming using two Raspberry Pi units acting as A2DP Source and Sink, transmitting LC3plus High-Resolution encoded audio over a Classic Bluetooth connection.



Fraunhofer-Institute for Integrated
Circuits IIS

Technical inquiries:
Alexander Tschekalinskij
lc3plus-techsupport@iis.fraunhofer.de

Am Wolfsmantel 33
91058 Erlangen, Germany
www.iis.fraunhofer.de/audio

The information in this document is provided 'as is' and, to the extent permitted by law, without any representations or warranties, express or implied, including, without limitation to, warranties of merchantability or fitness for a particular purpose, non-infringement or that the implementation of the contents of this documentation will not infringe any third-party patents, copyrights or other rights.

Fraunhofer will not be liable for any direct, indirect, special or consequential damages arising out of any use of the document or the performance or implementation of the contents thereof. The implementation of any content is done at your own risk.

The information in this document is owned and copyrighted by Fraunhofer-Gesellschaft and may be changed and/or updated at any time without further notice.

The LC3plus High Resolution audio coding scheme is a development of Fraunhofer in cooperation with Ericsson AB, pursuant to the specification issued in clause 5.8 of the ETSI TS 103 634.

This document describes how to transmit LC3plus High Resolution bitstream via Bluetooth LE Audio by way of signaling a vendor specific codec. The LC3plus High Resolution transport over Bluetooth LE Audio described in this document is not part of any Bluetooth SIG specification.

Implementations of the LC3plus High Resolution audio coding scheme are not compliant with any current Bluetooth specification and cannot be qualified pursuant to the Bluetooth qualification process and the LC3plus High Resolution codec is not a compliant portion according to Bluetooth SIG PCLA.

For the purpose of clarity, the LC3plus High Resolution mode is a codec defined by ETSI. It is different from the low complexity communication codec (LC3) specified by Bluetooth. The LC3plus High Resolution mode is not compatible with LC3 specified by Bluetooth.

This documentation doesn't grant any patent license for the use of LC3plus patent licenses for necessary patent claims for the LC3plus High Resolution codec (including those of Fraunhofer) may be obtained from the respective patent owners.

For more information regarding licensing of LC3plus, please visit:
<https://www.iis.fraunhofer.de/en/ff/amm/lizenz/patent.htm>

Table of contents

1	Overview	4
1.1	Operating modes	4
1.2	Hardware Reference	4
1.3	Requirements	4
2	Setup	5
2.1	Flash the Firmware	5
2.2	Find the IP Addresses.....	6
2.3	Credentials.....	6
2.4	SSH Into the Pis.....	6
2.5	Sync Clocks	7
2.6	Clean Restart (Run Before Every Measurement)	7
2.7	Bluetooth Pairing (First Time or After Reset).....	7
2.8	Reconnect (Existing Pairing).....	7
3	System Operation	8
3.1	Start HCI Logging.....	8
3.2	Start Streaming	8
3.3	Check Bluetooth Signal Strength	8
4	Codec Configuration	8
4.1	Default Profile	8
4.2	Custom Bitrate / Multi-Codec	8
5	Collecting Logs	9
6	Troubleshooting	9
7	Safe Shutdown	9

1 Overview

1.1 Operating modes

The demonstration uses a fixed two-device topology:

Source → **Sink**: Pi 1 acts as the A2DP source (transmitter), Pi 2 acts as the A2DP sink (receiver).

1.2 Hardware Reference

This evaluation setup utilizes two Raspberry Pi single-board computers configured in a dedicated Master/Slave relationship. While this guide specifically references the Raspberry Pi 4 Model B hardware layout (detailed below), the installation image and core software architecture are compatible across Raspberry Pi generations 1 through 5. For optimal streaming stability and consistent Bluetooth connection tracking, ensuring proper power distribution via a dedicated USB-C power supply and using wired Ethernet for the local control network is highly recommended.

Device	Configuration
Pi 1	Source
Pi 2	Sink

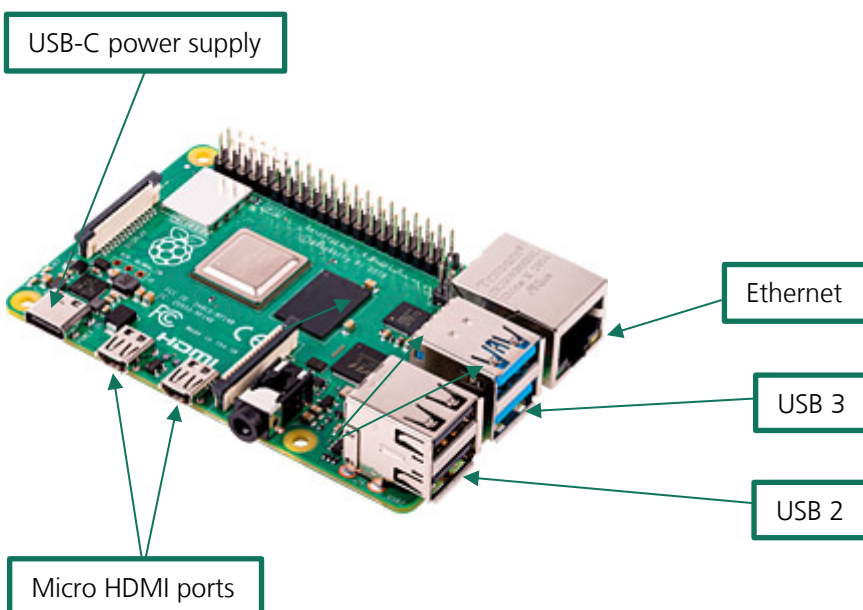


Figure 1: Raspberry Pi 4 (Source: Raspberry — [raspberrypi.com](https://www.raspberrypi.com))

1.3 Requirements

- 2× Raspberry Pi units with BlueALSA (LC3plus build) installed
- Both Pis connected to the same local network (WiFi or Ethernet)
- Host computer (Mac or Windows) on the same network
- **Mac only**: sshpass installed (via Homebrew: `brew install sshpass`)

- **Windows only:** PuTTY or Windows Terminal with OpenSSH enabled; pscp for file transfer
- Audio test file: ABBA_48000_1ch.wav present on Pi 1

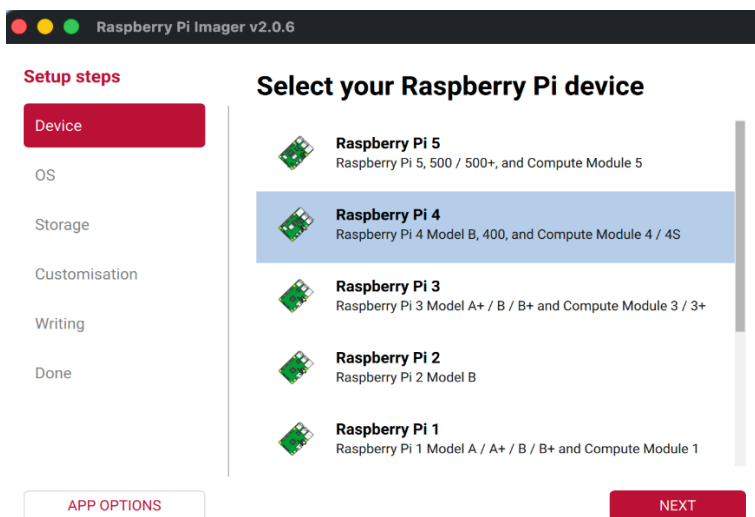
2 Setup

2.1 Flash the Firmware

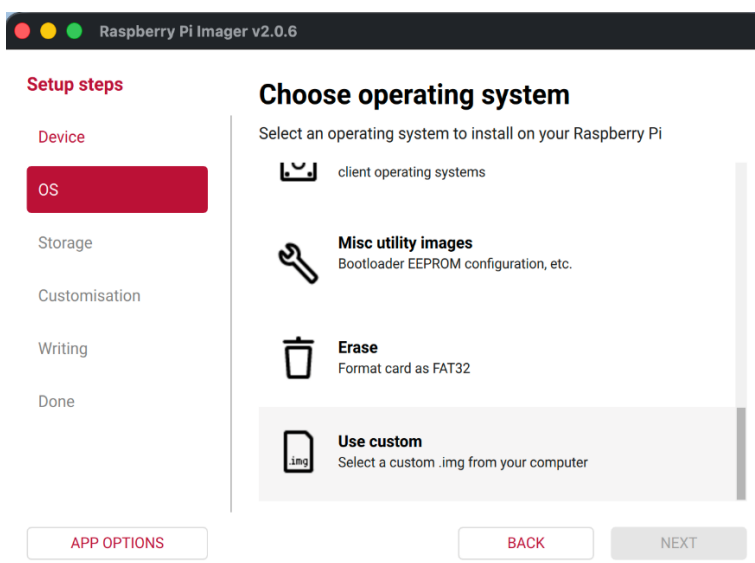
Both Pis must be flashed with the custom Raspberry Pi image before proceeding. This image includes all required software (BlueALSA, LC3plus build, and configuration).

Tools needed: [Raspberry Pi Imager](#) - available for Mac and Windows.

1. Download and install Raspberry Pi Imager on your host machine.
2. Insert the SD card of the first Pi into your computer.
3. Open Raspberry Pi Imager and click **Device** → **your Raspberry device**



4. Choose OS → Use custom, then select Rasp1.img



5. Click **Choose Storage** and select the SD card.
6. Click **Write** and wait for the process to complete.
7. Repeat steps 2–5 for the second Pi's SD card.
8. Insert the flashed SD cards back into the respective Pis and power them on.

NOTE: Both Pis use the same image (Raspi1.img). The source/sink roles are not defined by the image but by the BlueALSA commands run at measurement time (see Section 2.6).

2.2 Find the IP Addresses

The Pis connect to WiFi and receive IP addresses automatically from the router (DHCP), so the addresses will differ depending on the network. After each Pi is connected and booted, find its current IP by running directly on the Pi:

```
hostname -I
```

Note down both addresses — you will need them for every SSH command in this guide.

2.3 Credentials

The SSH username and password are set during the Raspberry Pi OS initial configuration and can be changed at any time (passwd to change the password, usermod to rename the user). The values used during development were:

- **Username:** alex
- **Password:** raspi!b&o!airoha

Replace these with the current credentials if they have been changed.

2.4 SSH Into the Pis

Open two terminal windows and connect to each Pi.

Mac / Linux:

Terminal 1:

```
ssh <pi1-user>@<pi1-ip>
```

Terminal 2:

```
ssh <pi2-user>@<pi2-ip>
```

Windows:

Option A — Windows Terminal / PowerShell (OpenSSH must be enabled):

Window 1:

```
ssh <pi1-user>@<pi1-ip>
```

Window 2:

```
ssh <pi2-user>@<pi2-ip>
```

Option B — PuTTY: open two sessions, enter the IP address of each Pi, port 22, connection type SSH.

2.5 Sync Clocks

Run on **both Pis** to ensure timestamps are aligned for log merging:

```
sudo timedatectl set-ntp true
```

2.6 Clean Restart (Run Before Every Measurement)

```
sudo killall bluealsa
sudo rm -rf /var/lib/bluealsa/*
```

Pi 1:

```
rm -f source.log
sudo bluealsa -p a2dp-source -c LC3plus 2>&1 | tee
source.log
```

Pi 2:

```
rm -f sink.log
sudo bluealsa -p a2dp-sink -c LC3plus 2>&1 | tee sink.log
```

2.7 Bluetooth Pairing (First Time or After Reset)

Both Pis - clear the pairing cache:

```
sudo rm -rf /var/lib/bluetooth/*
sudo systemctl restart bluetooth
```

Pi 2 first - make discoverable:

```
bluetoothctl
power on
discoverable on
pairable on
```

Pi 1 - scan, pair, and connect:

```
bluetoothctl
power on
scan on
pair <pi2-mac>
trust <pi2-mac>
connect <pi2-mac>
scan off
```

Finding Pi 2's MAC address:

run hciconfig on Pi 2 and note the BD Address value

2.8 Reconnect (Existing Pairing)

If the devices are already paired, skip section 2.7.

Pi 2:

```
bluetoothctl
power on
discoverable on
pairable on
```

Pi 1:

```
bluetoothctl
power on
connect <pi2-mac>
```

3 System Operation

3.1 Start HCI Logging

Start packet capture before streaming. Run on each Pi:

Pi 1:

```
sudo btmon -w source_hci.log
```

Pi 2:

```
sudo btmon -w sink_hci.log
```

3.2 Start Streaming

Run in this order:

Pi 2 - start recording:

```
arecord -D bluealsa -f s16_le -c 2 -r 48000 capture.wav
```

Pi 1 - stream audio:

```
aplay -D bluealsa ABBA_48000_1ch.wav
```

Pi 2 - play back the recording locally (optional verification):

```
aplay -D hw:0,0 capture.wav
```

3.3 Check Bluetooth Signal Strength

While connected on Pi 1:

```
watch -n 1 "hcitool rssi <pi2-mac>"
```

4 Codec Configuration

4.1 Default Profile

The default setup streams LC3plus only:

Source

```
sudo bluealsa -p a2dp-source -c LC3plus 2>&1 | tee source.log
```

Sink

```
sudo bluealsa -p a2dp-sink -c LC3plus 2>&1 | tee sink.log
```

4.2 Custom Bitrate / Multi-Codec

To override the bitrate or enable fallback codecs:

Source - LC3plus at 192 kbps, with SBC and AAC disabled

```
sudo /usr/bin/bluealsa -p a2dp-source \  
-c LC3plus \  
-c -SBC \  
-c -AAC \  
--lc3plus-bitrate=192000
```

Source - alternative bitrate (96 kbps)

```
sudo /usr/bin/bluealsa -p a2dp-source \  
-c LC3plus
```

```
-c -SBC
-c -AAC
--lc3plus-bitrate=96000
```

Sink - matching configuration

```
sudo /usr/bin/bluealsa -p a2dp-sink \
  -c LC3plus
  -c -SBC
  -c -AAC
```

5 Collecting Logs

After measurement, copy logs from both Pis to your host machine and merge by timestamp.

Mac:

```
cd <your-working-directory>
scp <pi1-user>@<pi1-ip>:source.log .
scp <pi1-user>@<pi1-ip>:source_hci.log .
scp <pi2-user>@<pi2-ip>:sink.log .
scp <pi2-user>@<pi2-ip>:sink_hci.log .
```

```
# Merge application logs sorted by timestamp
sort -m -k1,2 source.log sink.log > combined.log
```

Windows (PowerShell with pscp):

```
cd <your-working-directory>
pscp <pi1-user>@<pi1-ip>:source.log .
pscp <pi1-user>@<pi1-ip>:source_hci.log .
pscp <pi2-user>@<pi2-ip>:sink.log .
pscp <pi2-user>@<pi2-ip>:sink_hci.log .
```

NOTE: The sort -m merge command is a Unix utility and is not available natively on Windows. To merge logs on Windows, use Git Bash, WSL, or copy the files to a Mac/Linux machine for merging.

6 Troubleshooting

Symptom	Action
BlueALSA fails to start	Run <code>sudo killall bluealsa</code> first, then retry
Pairing fails	Clear cache: <code>sudo rm -rf /var/lib/bluetooth/*</code> , restart bluetooth, retry from Pi 2
No audio recorded	Confirm BlueALSA is running on both Pis and connection is established before <code>arecord</code>
Logs missing timestamps	Re-run <code>timedatectl set-ntp true</code> on both Pis
SSH not working on Windows	Ensure OpenSSH client is enabled: Settings → Apps → Optional Features → OpenSSH Client
pscp not found on Windows	Install PuTTY from https://www.putty.org — pscp is included
SD card errors after crash	Always run <code>sudo shutdown now</code> before unplugging

7 Safe Shutdown

Always shut down properly before unplugging to prevent SD card corruption:

```
sudo shutdown now
```