

Audio with embedded Linux training

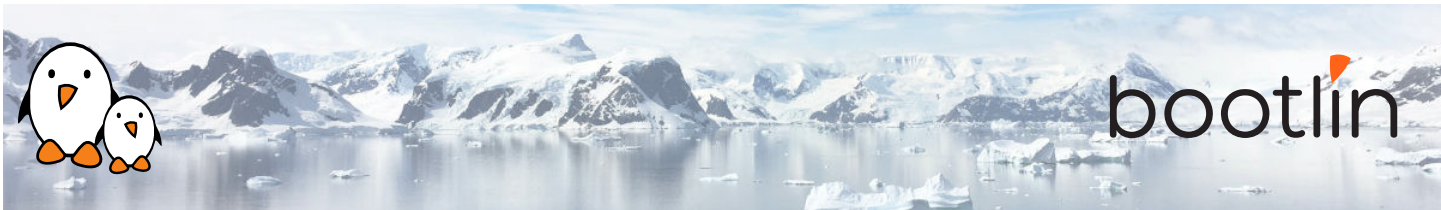
On-site training, 2 days

Latest update: May 08, 2024

Title	Audio with embedded Linux training
Training objectives	<ul style="list-style-type: none">• Be able to understand the basics of audio theory: analog vs. digital, sampling theory, audio formats and audio hardware• Be able to understand the architecture of ASoC, the Linux kernel subsystem for audio devices used in embedded systems, the role of the different drivers and the Device Tree representation• Be able to write a Device Tree describing the topology of audio hardware on an embedded Linux system• Be able to use complex audio routing capabilities, audio power management capabilities, and complex audio components such as auxiliary devices or amplifiers• Be able to configure the ALSA user-space components and ALSA audio widgets• Be able to use the available user-space APIs for playing and capturing audio, as well as the PipeWire audio server and the GStreamer multimedia pipeline library
Duration	Two days - 16 hours (8 hours per day)
Pedagogics	<ul style="list-style-type: none">• Lectures delivered by the trainer: 75% of the duration• Practical demonstrations done by the trainer: 25% of the duration• Electronic copies of presentations, lab instructions and data files. They are freely available at https://bootlin.com/doc/training/audio.
Trainer	One of the engineers listed on: https://bootlin.com/training/trainers/
Language	Oral lectures: English, French. Materials: English.
Audience	Engineers who need a detailed level of understanding of audio concepts, audio hardware components used in typical embedded systems and how the audio stack works in Linux.



Prerequisites	<ul style="list-style-type: none">• Knowledge and practice of UNIX or GNU/Linux commands: participants must be familiar with the Linux command line. Participants lacking experience on this topic should get trained by themselves, for example with our freely available on-line slides at bootlin.com/blog/command-line/.• Minimal experience in embedded Linux development: participants should have a minimal understanding of the architecture of embedded Linux systems: role of the Linux kernel vs. user-space, development of Linux user-space applications in C. Following Bootlin's <i>Embedded Linux</i> course at bootlin.com/training/embedded-linux/ allows to fulfill this pre-requisite.• Minimal English language level: B1, according to the <i>Common European Framework of References for Languages</i>, for our sessions in English. See bootlin.com/pub/training/cefr-grid.pdf for self-evaluation.
Required equipment	<p>For on-site sessions at our customer location, the customer must provide:</p> <ul style="list-style-type: none">• Video projector• Large monitor• Drawing board
Certificate	Only the participants who have attended all training sessions, and who have scored over 50% of correct answers at the final evaluation will receive a training certificate from Bootlin.
Disabilities	Participants with disabilities who have special needs are invited to contact us at training@bootlin.com to discuss adaptations to the training course.



Day 1 - Morning

Lecture - Digital Audio Representation

- What is sound?
- Sampling theory
- Sample size, sample rate
- Audio formats: I2S, LJ, RJ, DSPA, DSPB
- AC97
- IEC 61937 (S/PDIF and HDMI)
- PDM

Introducing the basic notions used for representing audio waveforms.

Lecture - Hardware

- Signals
- CPU Digital Audio Interfaces
- CODEC Digital Audio Interfaces
- Amplifiers
- Clocks and clock providers

Presenting the hardware involved in the audio playback or capture.

Lecture - Linux kernel ASoC subsystem

- ASoC: the ALSA System-on-Chip subsystem in the Linux kernel
- Describing audio cards with Device Tree: *audio-graph-card*, *simple-audio-card*
- Linux kernel drivers for audio cards
- Linux kernel drivers for audio codecs
- Controls
- Linux kernel drivers for CPU audio interfaces

Introducing the Linux kernel subsystem for audio on embedded systems.

Day 1 - Afternoon

Lecture - Linux kernel helpers

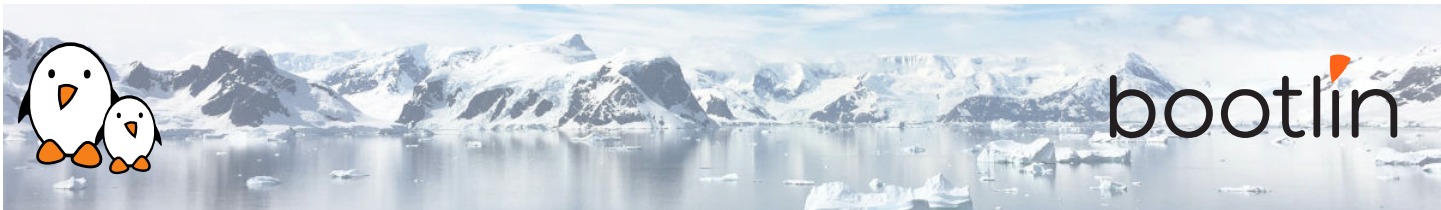
- *regmap*, *regcache*
- DMA handling

Presenting the common helper APIs.

Lecture - More audio components

- Auxiliary devices, amplifiers, muxing
- Jack detection
- Asynchronous Sample Rate Converter

Presenting more components of the sound cards.



Lecture - Routing

- Routing audio
- DAPM: Dynamic Audio Power Management

Presenting the audio routes and power management.

Day 2 - Morning

Lecture - Userspace, hardware configuration

- ALSA plugins
- `asound.conf`
- Sound card configuration

Configuring the userspace audio paths and effects.

Demo - Card configuration examples

- Reordering channels
- Splitting channels
- Resampling
- Mixing
- LADSPA

Exercising the ALSA plugins.

Lecture - Userspace, controls configuration

- `amixer`
- `alsamixer`
- Userspace API
- Saving state: `alsactl`, `asound.state`

Configuring the userspace audio paths and effects.

Demo - Configuring controls

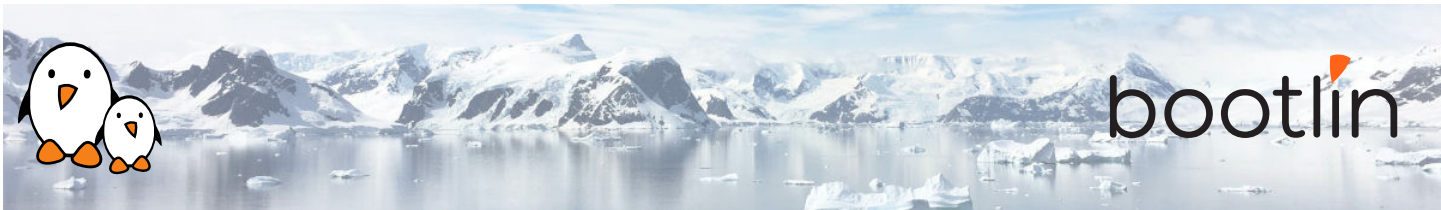
- `alsamixer` demonstration
- `asound.state` examples
- Custom application

Configuring the sound card controls.

Lecture - Userspace, playing and capturing audio

- Userspace ALSA API

Playing and capturing audio samples.



Day 2 - Afternoon

Lecture - Troubleshooting

- Debugging
- *vizdpm*

Presenting how to debug common issues.

Demo - Debugging

- Examples of common issues and their resolutions

Lecture - Pipewire

- Pipewire introduction
- Pipewire configuration
- Pipewire tools (pwdump, pw-cli, ...)
- Pipewire session and policy management
- Pipewire modules and filtering
- *WirePlumber*

Using Pipewire as the sound server.

Demo - Pipewire

- Running pipewire on the target
- Inspecting the configuration and setting properties
- Dynamic routing and patchbay
- Using modules and Filter-Chain

Running Pipewire and exercising advanced configuration.

Lecture - The GStreamer multimedia framework

- *GStreamer*
- GStreamer pipelines
- GStreamer plugins

Using Gstreamer for audio capture and playback.

Demo - GStreamer

- Running gstreamer on the target
- Creating multiple pipelines

Running Gstreamer using different audio pipelines.