libcamera: Making Complex Cameras Easy

OSS Japan 2019
Tokyo, Japan

Laurent Pinchart
laurent.pinchart@ideasonboard.com
libcamera

Cameras are complex devices that need heavy hardware image processing operations. Control of the processing is based on advanced algorithms that must run on a programmable processor. This has traditionally been implemented in a dedicated MCU in the camera, but in embedded devices algorithms have been moved to the main CPU to save cost. Blurring the boundary between camera devices and Linux often left the user with no other option than a vendor-specific closed-source solution.

To address this problem the Linux media community has very recently started collaboration with the industry to develop a camera stack that will be open-source-friendly while still protecting vendor core IP. libcamera was born out of that collaboration and will offer modern camera support to Linux-based systems, including traditional Linux distributions, ChromeOS and Android.
In the beginning were simple pipelines...

Why?
... and they were simple to control, with a single API.
Then the world became complex ...

Why?
... and application developers were left suffering.
Solutions were proposed... but never implemented.
Then hope came back.
libcamera
libcamera provides a complete userspace camera stack.

The ‘mesa’ of the camera world.
Camera Devices & Enumeration

`libcamera` enumerates cameras...

Diagram showing the interaction between Userspace and Kernel through camera devices and enumeration.
... and exposes their capabilities.
It supports multiple concurrent streams for the same camera...

Streams
Per-Frame Controls

... and per-frame controls.
Image Processing Algorithms (3A)

Image Processing Algorithms are loaded as external modules.
Adaptation layers offer backward compatibility with existing APIs...
Adaptation

... and integrate libcamera with other operating systems.
libcamera
Central to the stack is the Camera object, interfacing to device-specific pipeline handlers.
The camera configuration is backed by device-specific validation from the pipeline handler.
The Camera Manager enumerates media devices and instantiates corresponding pipeline handlers.
The pipeline handlers create and register one or more cameras.
The pipeline handler interfaces with all kernel devices. It abstracts them and exposes video streams to upper layers.
Image Processing Algorithms (IPA) receive statistics from the hardware and compute optimal image parameters.
IPAs are separate modules that don’t access kernel devices directly. They only have access to their pipeline handler through the IPA API.
Closed-source IPAs are sandboxed in a separate process. They communicate with the pipeline handler through IPC.
The IPC is handled in core components, transparently for both the pipeline handler and the IPA.
Many helper classes ease the implementation of pipeline handlers for device vendors.
Native V4L2 applications are supported through a transparent compatibility layer.
A single Android camera HAL module implementation for all devices supported by libcamera.
(o) | libcamera
The camera implements a simple state machine.
The camera generates a configuration template from roles.

The configuration is applied to the camera.

The configuration can be modified, and shall be validated.

Camera Configuration
A request is created on the Camera, populated with a Buffer for each Stream, and queued for capture.
Buffer and request completion are notified separately.

Applications submit new requests to keep the streams going.

Request Completion
libcamera
Contributing

libcamera is developed as a free software project and welcomes contributors. Whether you would like to help with coding, documentation, testing, proposing new features, or just discussing the project with the community, you can join our official public communication channels, or simply check out the code.

Mailing List

We use a public mailing list as our main means of communication. You can find subscription information and the messages archive on the libcamera-devel list information page.

IRC Channel

For informal and real-time discussions, our IRC channel on Freenode is open to the public. Point your IRC client to #libcamera to say hello, or use the #libcamera channel.

Source Code

libcamera is in early stages of development, and no releases are available yet. The source code is available from the project’s git tree, hosted by LinuxTV.

$ git clone git://linuxtv.org/libcamera.git

Documentation

Project documentation is created using Sphinx. Source level documentation uses Doxygen. Please make sure to document all code during development.

Sphinx integration with Doxygen is planned, likely using Breathe and Exhale.

Submitting Patches

Contribute
libcamera-devel@lists.libcamera.org
irc://chat.freenode.net/#libcamera
laurent.pinchart@ideasonboard.com
ご清聴ありがとうございました！