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.
Why?
Why?
Libcamera Architecture
Camera Stack
Camera Devices & Enumeration
Capabilities & Profiles

Capabilities

Profiles
Streams
Per-Frame Controls
3A & Image Enhancement Algorithms
Adaptation
Adaptation
libcamera architecture
Camera Device
Camera Devices Manager
Camera Devices Manager
Pipeline Handler
Image Processing Algorithms
Image Processing Algorithms

Camera Device <-> API 

3A } |

Image Processing Algorithms 

{ +-----------------+ }

{ Image Algorithms }

{ +-----------------+ }

{ +-----------------+ }

# # # # # # # # # # #
Helpers and Support Classes

- MC & V4L2 Support
- Buffers Allocator
- Sandboxing IPC
- Plugins Manager
- Pipeline Runner
- Debug + Logging
- ...
Native V4L2 Application

<table>
<thead>
<tr>
<th>open()</th>
<th>ioctl()</th>
<th>mmap()</th>
</tr>
</thead>
</table>

LD_PRELOAD=libcamera-v4l2.so

open() {        ioctl() {        mmap()        ...
  ...
}        ...

libcamera

libcamera v API

V4L2 Compatibility
Android Camera HAL
Camera Stack
libcamera architecture status
We need YOU

- Platforms
- Needs, Use cases
- Adoption

Status
Adoption
Contribute

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-dev 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 WebChat.

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

Patches submitted to the libcamera project must be certified as suitable for integration into an open source project. As such libcamera follows the same model as utilised by the Linux kernel, and requires the use of 'Signed-off-by' tags in all patches.

By signing your contributions you are certifying your work in accordance with the Developer's Certificate of Origin.
ขอบคุณ!