Libcamera: Making Complex cameras Easy!

Umang Jain
email@uajain.com
IRC : uajain
Complexity increases - Image signal processors

- System peripherals that operate on digital images
- DMA, resizing, scaling, cropping, pixel format conversions
- Kernel exposed new API named ‘MediaController’
Emergence of complex ISPs
• Problem statement:
  - Applications are requested to know a lot of details on the underlying hardware.
  - Triggering capture operations requires precise configuration and setup of the whole pipeline (ISP).
  - Recent modern laptops are shipping with powerful ISPs like Intel IPU3.
Subject: Re: Webcams not recognized on a Dell Latitude 5285 laptop
From: Nicolas Dufresne <nicolas@xxxxxxxxxxxx>
Date: Sun, 01 Apr 2018 15:45:16 -0400
In-reply-to: <f69b1c0c-fce3-c1c5-4eeb-7941028991de@univ-grenoble-alpes.fr>

This laptop embeds one of these new "complex" cameras from Intel. They requires IPU3 driver. Though, unlike traditional webcam, you need special userspace to use it (there is no embedded firmware to manage focus, whitebalance, etc, userspace code need to read the stats and manage that). As of now, there is no good plan on how to support this in userspace.
**libcamera** is a complete user space camera stack

- Abstract away from application all interfacing with V4L2 and Media-Controller
- Aims to be compatible with Linux V4L2-based applications, Android and ChromeOS
- Decent on documentation, application developer and pipeline handler guides, sample applications...
• mail based development: mailing list + patchwork
• we use Meson and Ninja, and they’re great.
• Newcomers and contributors’ friendly (believe me!)

• git://linuxtv.org/libcamera.git
• www.libcamera.org
• IRC: irc://chat.freenode.net/#libcamera
• Mailing list: libcamera-devel@lists.libcamera.org
Complex cameras debate

• Why Embedded Cameras are Difficult, and How to Make Them Easy
  Laurent Pinchart, Ideas on Board

• Complex Cameras on Linux
  Mauro Carvalho Chehab, Samsung

• Image Signal Processing (ISP) Drivers & how to merge one upstream
  Helen Koike, Collabora
Camera Stack
Platforms supported by libcamera
libcamera API

- CameraManager: camera enumeration
- Camera Configuration
- Config validation (negotiation with application’s request)
- FrameBuffer allocation
- Frame Capture
CameraManager : camera enumeration

CameraManager *cm = new CameraManager();
cm->start();

for (auto const &camera : cm->cameras())
    std::cout << camera->id() << std::endl;
Camera Configuration

```cpp
std::string cameraId = cm->cameras()[0]->id();
camera = cm->get(cameraId);
camera->acquire(); // exclusive lock

std::unique_ptr<CameraConfiguration> config =
    camera->generateConfiguration({ StreamRole::StillCapture });
```
Validation

- Generated CameraConfiguration has a StreamConfiguration instance for each StreamRole.

StreamConfiguration &streamConfig = config->at(0);

std::cout << "Default StillCapture configuration is: " << streamConfig.toString();
// Default StillCapture configuration is: 1280x720-MJPEG

//custom application’s request
streamConfig.size.width = 640;
streamConfig.size.height = 480;

config->validate();
Framebuffer allocator

FrameBufferAllocator *allocator = new FrameBufferAllocator(camera);

for (StreamConfiguration &cfg : *config) {
    int ret = allocator->allocate(cfg.stream());

    unsigned int allocated = allocator->buffers(cfg.stream()).size();
    std::cout << "Allocated " << allocated << " buffers for stream";
}

Frame Capture

- Happens through object called `Request`.
- `Request` associates at least one stream with FrameBuffer where stream data can be written.
- Request is queued to Camera with `camera::queueRequest()` after starting the camera via `camera::start()`.
- Camera will emit `Camera::requestCompleted` signal; thereafter Frame data can be accessed.
libcamera sample applications

- Cam – swiss knife commandline tool
- Qcam – Qt GUI based application
- Simple-cam – Minimal reference code depicting libcamera usage.

Don’t forget there is already decent documentation and app developers guide in:

src/libcamera/Documentation/guides
Please join! libcamera is a young project.

Read the doc: [http://libcamera.org/docs.html](http://libcamera.org/docs.html)

Read patches: [git://linuxtv.org/libcamera.git](git://linuxtv.org/libcamera.git)

Have a chat: #libcamera on freenode

We welcome inputs from anyone working with cameras!

Thank you!

Questions?