

# RDK-C libcamera validation with gst-launch-1.0

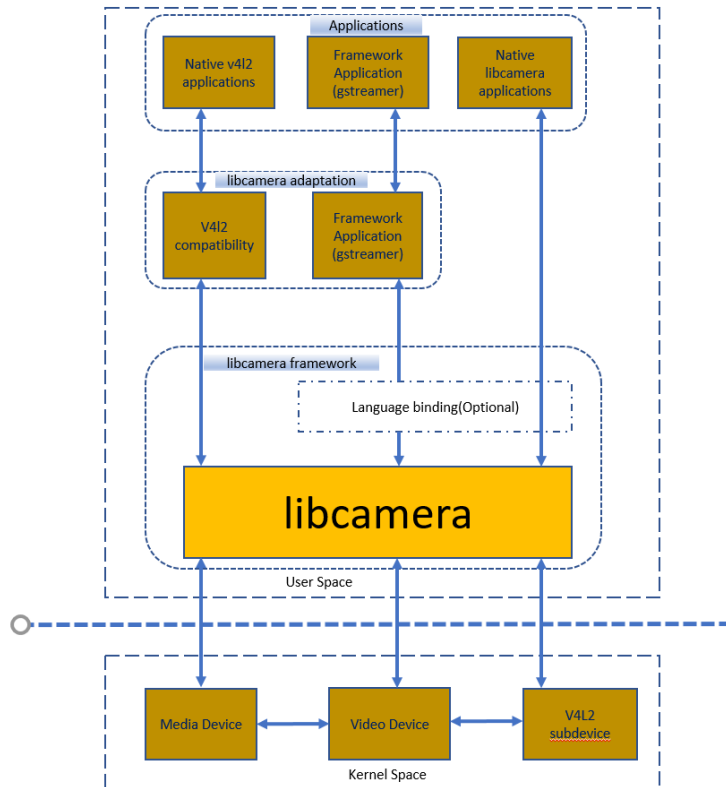
- [Introduction](#)
- [libcamera camera stack](#)
- [libcamera usage](#)
- [Connecting camera to Raspberry pi board](#)
- [Build and Flash Procedure](#)
- [Configuration](#)
- [libcamera video validation procedure](#)
- [Demo Video](#)
- [libcamerasrc.h264](#)

## Introduction

libcamera is an open source camera stack for many platforms with a core user space library, and support from the Linux kernel APIs and drivers already in place. It aims to control the complexity of embedded camera hardware by providing an intuitive API and method of separating untrusted vendor code from the open source core. libcamera aims to encourage the development of new embedded camera applications by limiting the complexity that developers have to deal with. The interface is designed around the way that modern embedded camera hardware works.

This page dedicated to validating raw video captured using libcamerasrc using gst-launch-1.0 in RPI-3 B+ board using imx219 camera.

## libcamera camera stack



The camera stack comprises four software layers. From bottom to top:

- The kernel drivers control the camera hardware and expose a low-level interface to user space through the Linux kernel V4L2 family of APIs (Media Controller API, V4L2 Video Device API and V4L2 Subdev API).
- The libcamera framework is the core part of the stack. It handles all control of the camera devices in its core component, libcamera, and exposes a native C++ API to upper layers. Optional language bindings allow interfacing to libcamera from other programming languages. Those components live in the same source code repository and all together constitute the libcamera framework.
- The libcamera adaptation is an umbrella term designating the components that interface to libcamera in other frameworks. Notable examples are a V4L2 compatibility layer, a gstreamer libcamera element, and an Android camera HAL implementation based on libcamera. Those components can live in the libcamera project source code in separate repositories, or move to their respective project's repository (for instance the gstreamer libcamera element).
- The applications and upper level frameworks are based on the libcamera framework or libcamera adaptation, and are outside of the scope of the libcamera project

## libcamera usage

- Multiple video stream from single camera and multiple camera
- At the same time we will display live streaming on screen, captures stills and record video at different resolution.
- Per frame control
- ISP:
  - AWB ( Automatic white Balance )
  - CCM ( color correction matrix )
  - AGC ( Automatic gain control)
  - ALSC ( Auto lens shading correction)
  - GEQ (green equalization algorithm)
  - SDN (spatial denoise) control algorithm
  - DPC (defective pixel correction) control algorithm
  - Contrast ( Gamma control algorithm)
  - Black level control algorithm
  - Sharpen control algorithm status

## Connecting camera to Raspberry pi board

Refer below link to connect camera to pi board

[connecting camera to Raspberry pi board](#)

## Build and Flash Procedure

Refer below link to build camera image

[RDK-C rdk-next Yocto 3.1 dunfell build for Raspberrypi](#)

## Configuration

### Configuartion step to capture video using libcamerasrc

```
1.Stop rms-launcher and mst-launcher using below command(this are temporary procedure to validate raw video content.)
#systemctl disable rms-launcher
#systemctl disable mst-launcher
2.Add below lines to /boot/config.txt
dtoverlay=imx219
core_min_freq=300
3.Save the file and reboot the target.
```

## libcamera video validation procedure

### STEP 1:

Execute below command inside target.

### libcamera video capture

```
#gst-launch-1.0 libcamerasrc ! video/x-raw,width=1280,height=720,framerate=30/1,format=NV12 ! videoconvert !
v4l2h264enc ! queue ! filesink location=libcamerasrc.h264
```

### STEP 2:

copy the libcamerasrc.h264 file to computer(DESKTOP)

### STEP 3

play video using vlc player

## Demo Video



libcamerasrc.h264