Towards Mainline Video Capture and Camera Support for Recent Rockchip SoCs

Michael Riesch

Recent Rockchip SoCs?

- Rockchip RK356xy
  - RK3566/68
  - RK3588
  - RK3576
  - RK3562
- For camera and video capture we can group
  - RK3566 + RK3568
  - RK3588 + RK3576 + RK3562

Video Capture and Camera Support?

- Capture video frames from companion ICs to RAM
- Camera requires more
  - Process frames (Debayering, 3A, corrections, ...)
  - Auxiliary devices: flash, lens controller, ...

Mainline?

- No explanation required here ;-)
- Rockchip SoCs are well supported in mainline Linux…
- … apart from Video Capture and Camera Support :-/
- Let’s close this gap!

Towards?

- WIP wherever you look
- Some discussions may be required
- Agenda for today:
  - Take a brief look at possible applications
  - Go through the software stack for modern multimedia SoCs
  - Give an overview of the current state of the art
  - Outline the next steps

COLLABORA

Applications?

Applications

- Cameras
  - Capture image data from camera
  - Control camera (brightness, autofocus on/off, zoom in/out, ...)
- Camera Types
  - One sensor, one camera
  - Multiple sensors, multiple cameras
  - Multiple sensors, one camera (3D, stitching, ...)
  - Wide/High Dynamic Range (WDR/HDR)
  - RAW capture, offline image processing

Applications

- Video Capture
  - Capture data
  - Control
- Video Capture Types
  - TV capture cards
  - Digital video inputs (HDMI, DisplayPort, …)
    - EDID control (supported resolutions, …)
    - HDCP support? (en/disable, status, version, …)

COLLABORA

Software Stack for modern multimedia SoCs

Software Stack

- Cameras
  - Linux kernel: provides V4L2/MC entities and graph
  - libcamera: collects V4L2/MC entities, provides cameras, executes IPA
  - PipeWire
  - (Gstreamer?)
  - Applications consume data, control camera

Software Stack

- Does it have to be **that** complicated?
  - Applications could access camera directly using libcamera or GStreamer
  - However, PipeWire allows for multiple consumers, browser integration available, security in sandboxing scenarios, …
  - It already works (exception: camera control???)

Software Stack

- Digital Video Inputs
  - Linux kernel: provides V4L2/MC entities and graph
  - Component that abstracts the input is missing here!
  - Applications consume data (V4L2, PipeWire, Gstreamer, ...) and control input (V4L2)
  - libdvinput? libcamera byproduct? PipeWire plugin for DV inputs?

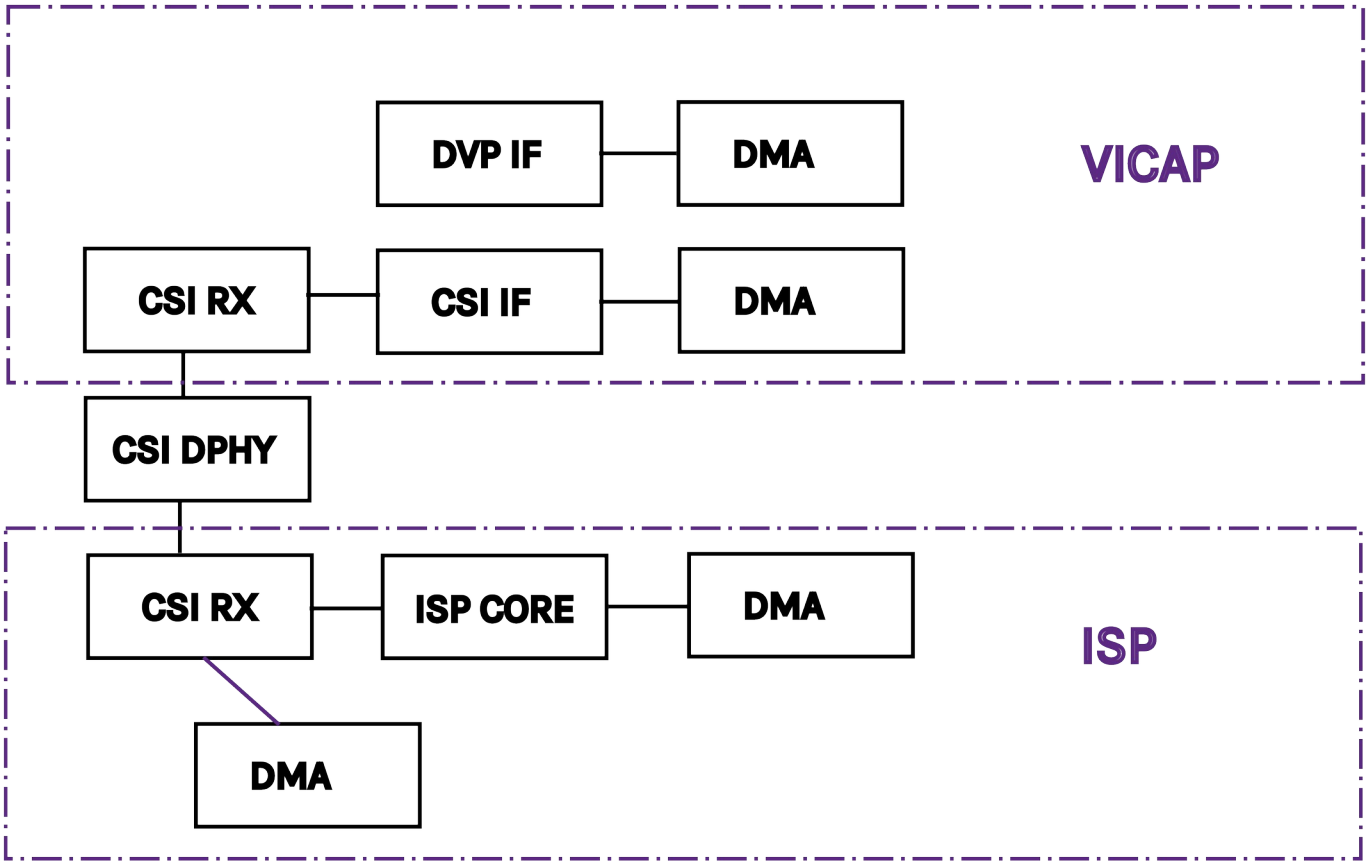
Software Stack

- Let’s focus on the low level from this point
  - Discuss the hardware components and their variants
  - How to expose them to the user space using V4L2/MC?

COLLABORA

Hardware Components in Rockchip RK35xy SoCs

RK3566/RK3568



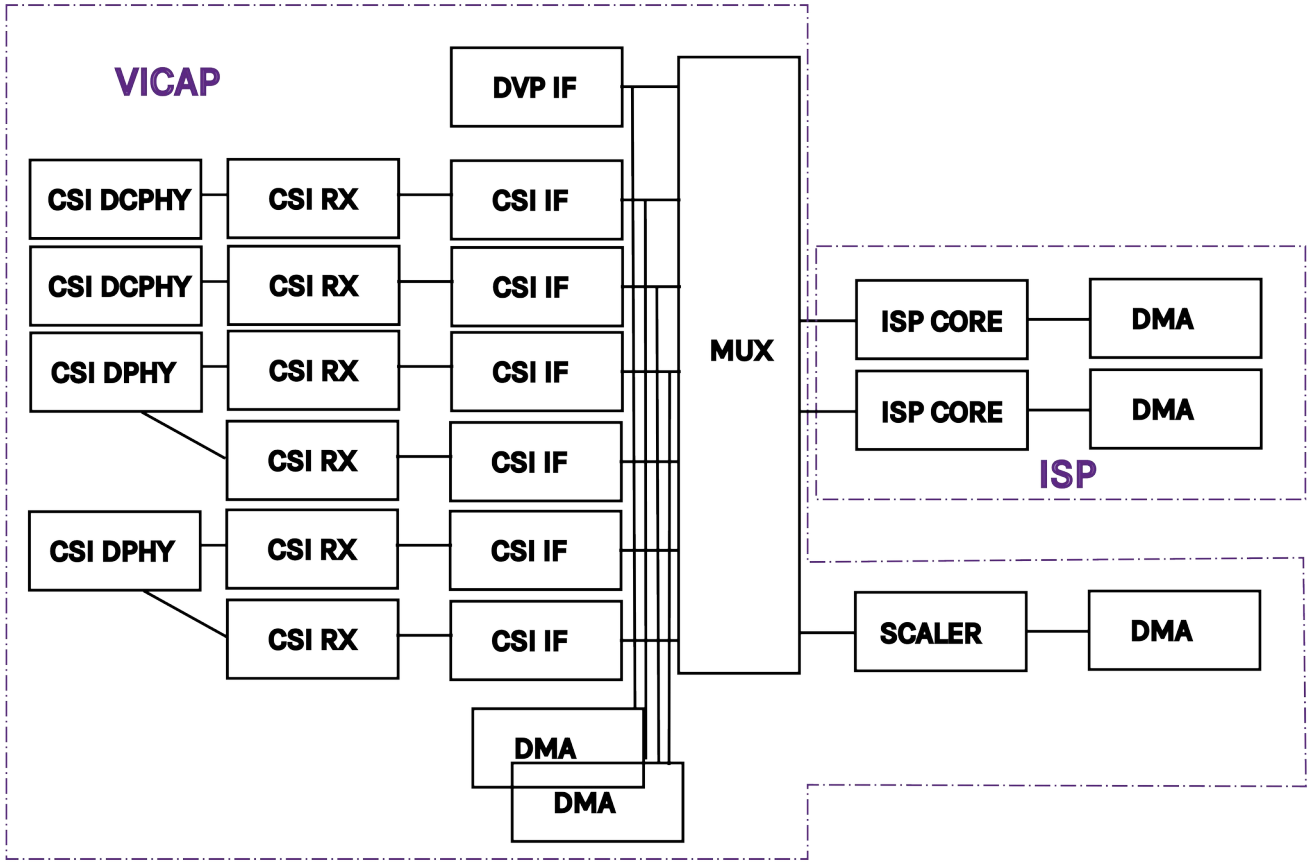
RK3566/RK3568

- 1x MIPI CSI-2 DPHY
  - 1x4 lanes or 2x2 lanes (split mode)
  - Data can be passed to ISP and/or VICAP
- 1x ISP v2.1
  - inline mode using integrated MIPI CSI-2 receiver
  - mem2mem mode, raw capture, HDR

RK3566/RK3568

- 1x MIPI CSI-2 receiver (coupled to VICAP)
- 1x VICAP
  - Digital Video Port (parallel 8/16 bit, BT.656/BT.1120) interface
  - 1x MIPI interface
  - 4x DMA engine for DVP (BT.1120 IDs?)
  - 4x DMA engine for MIPI (CSI VC)

RK3588/RK3576/RK3562



RK3588/RK3576/RK3562

- 2x MIPI CSI-2 DPHY
  - 1x4 lanes or 2x2 lanes (split mode), all to VICAP
- 2x/1x/0x MIPI CSI-2 DCPHY, no split mode, all to VICAP
- 2x/1x/1x ISP *v3.0/v3.9/v3.2 (?)*
  - inline mode from VICAP MUX/TOISP
  - mem2mem mode, ~~raw capture~~, HDR, unite mode

RK3588/RK3576/RK3562

- 6x/5x/4x MIPI CSI-2 receiver (coupled to VICAP)
- 1x VICAP
  - 1x/1x/0x Digital Video Port interface, 6x/5x/4x MIPI interfaces
  - 4x/4x/0x DMA engine for DVP (BT.1120 IDs?)
  - 24x/20x/16 DMA engine for MIPI (CSI VC)
  - MUX/TOISP block, SCALER (4x/1x/1x DMA engine)
- 1x/0x/0x HDMI RX (already well supported in mainline)

COLLABORA

How to expose the HW (RK3588) using V4L2/MC

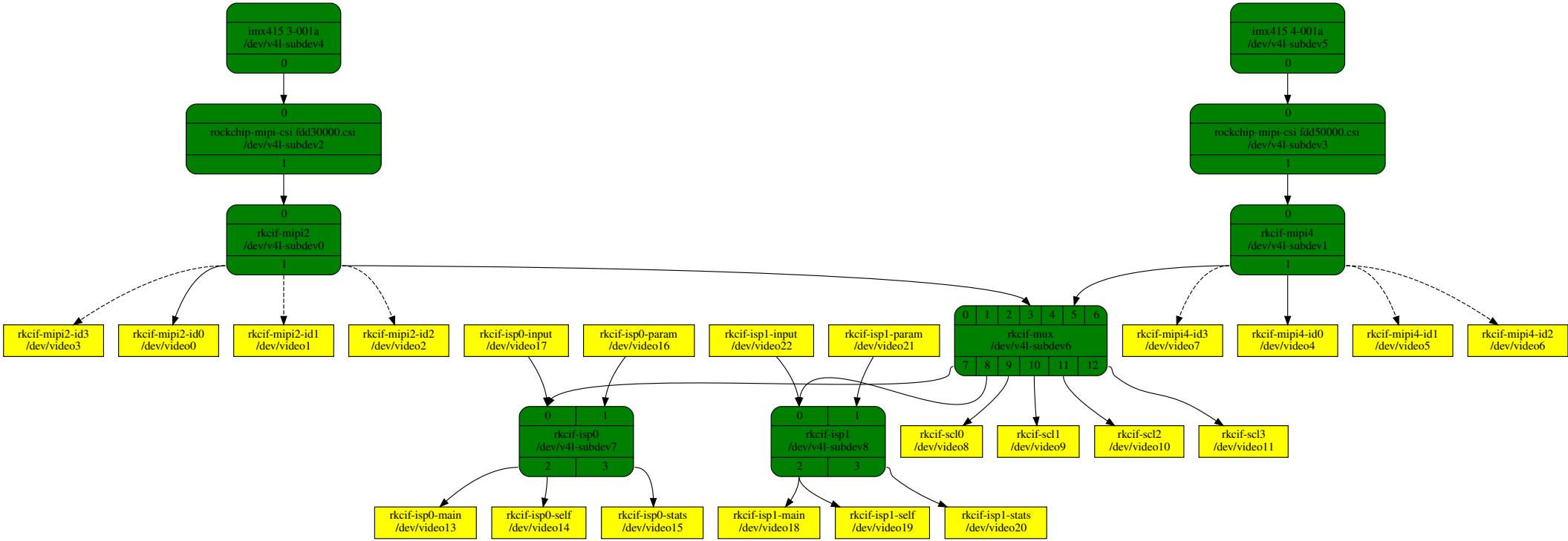
Expose the HW using V4L2/MC

- Use cases
  - Multiple sensors/companion ICs for multiple cameras and video inputs (at a certain number, this requires time division multiplex including buffering frames in RAM and processing it in mem2mem mode)
  - Two sensors in sync for 3D camera
  - Two sensors in sync for stitching: both ISPs write to the same frame buffer
  - One sensor with multiple streams for HDR camera (buffer two or three frames in RAM)

Expose the HW using V4L2/MC

- Goals
  - Leave the use case selection to the user space
    - No use case selection in device tree (DT describes HW, not policy)
    - Expose the HW connections and components in the media graph
    - Many entities, complex and large media graph!
  - No magic BSP beans, use what V4L2/MC offers and extend it as required
    - Handle MIPI CSI-2 VCs with streams API
    - Multicontext MC (in discussion) for time-division multiplex

Expose the HW using V4L2/MC



Expose the HW using V4L2/MC

- Questions
  - Will multicontext MC allow for different user space components working on the same media graph?
  - Should the two ISPs be exposed as one dual-core entity or two single-core entities? Who will be in charge of scheduling?
  - Should there be a scheduler framework in V4L2/MC?

“How V4l2 Transformed To Support Embedded Cameras”
Laurent Pinchart, Tue 11:00 CEST
https://sched.co/25Vql

COLLABORA

Current state of HW support in Linux

Status matrix 1/2

SoC	RK3566/68	RK3588	RK3576	RK3562
MIPI CSI-2 DPHY	DONE	PATCH v2	TODO	TODO
MIPI CSI-2 DCPHY	n/a	TODO	TODO	n/a
MIPI CSI-2 receiver	PATCH v10	WIP	TODO*	TODO*
VICAP MIPI capture	PATCH v10	WIP	TODO*	TODO*
VICAP DVP capture	PATCH v10	TODO*	TODO*	n/a
VICAP MUX/TOISP	n/a	TODO	TODO	TODO
VICAP SCALER	n/a	TODO	TODO	TODO

* should be easy™

Status matrix 2/2

SoC	RK3566/68	RK3588	RK3576	RK3562
ISP MIPI CSI-2 RX	WIP	n/a	n/a	n/a
ISP main/self path	WIP	TODO	TODO	TODO
ISP stats/params/3A	WIP	TODO	TODO	TODO
ISP raw capture	TODO	TODO	TODO	TODO
ISP mem2mem	TODO	TODO	TODO	TODO
ISP HDR	TODO	TODO	TODO	TODO
ISP dual support	n/a	TODO	n/a	n/a

* nothing easy™ here

COLLABORA

Next steps

Next Steps

1) Complete RK3588 VICAP support
   - Most importantly, MUX/TOISP support
   - Optionally: SCALER and DVP support (for the latter, find HW?)
2) Minimal RK3588 ISP support
   - Main path/self path with debayering only, no processing
   - Statistics, parameters, processing
3) ???

Next Steps for Y'all

- If you are a developer
  - Go forth and test :-)
  - Play around with different setups, different use cases.
- If you are a CEO, CTO, ...
  - Hire us, you'll get to choose with ??? is :-)
- If you are a linux-media maintainer
  - Pleeeease merge v10 so we can move forward :-)

Next Steps for Y’all

- And if you don’t want wait for the ISP support…
  - …consider giving the software ISP a try
  - Capture raw frames and process them on the CPU/GPU

“Software ISP FOSS Support for MIPI Cameras”
Hans de Goede & Bryan O’Donoghue, Wed 09:00 CEST
https://sched.co/25VuA

Thank you!

We are hiring 👷
col.la/careers