Linaro Connect
MADRID 2024 | MAY 14-17 2024

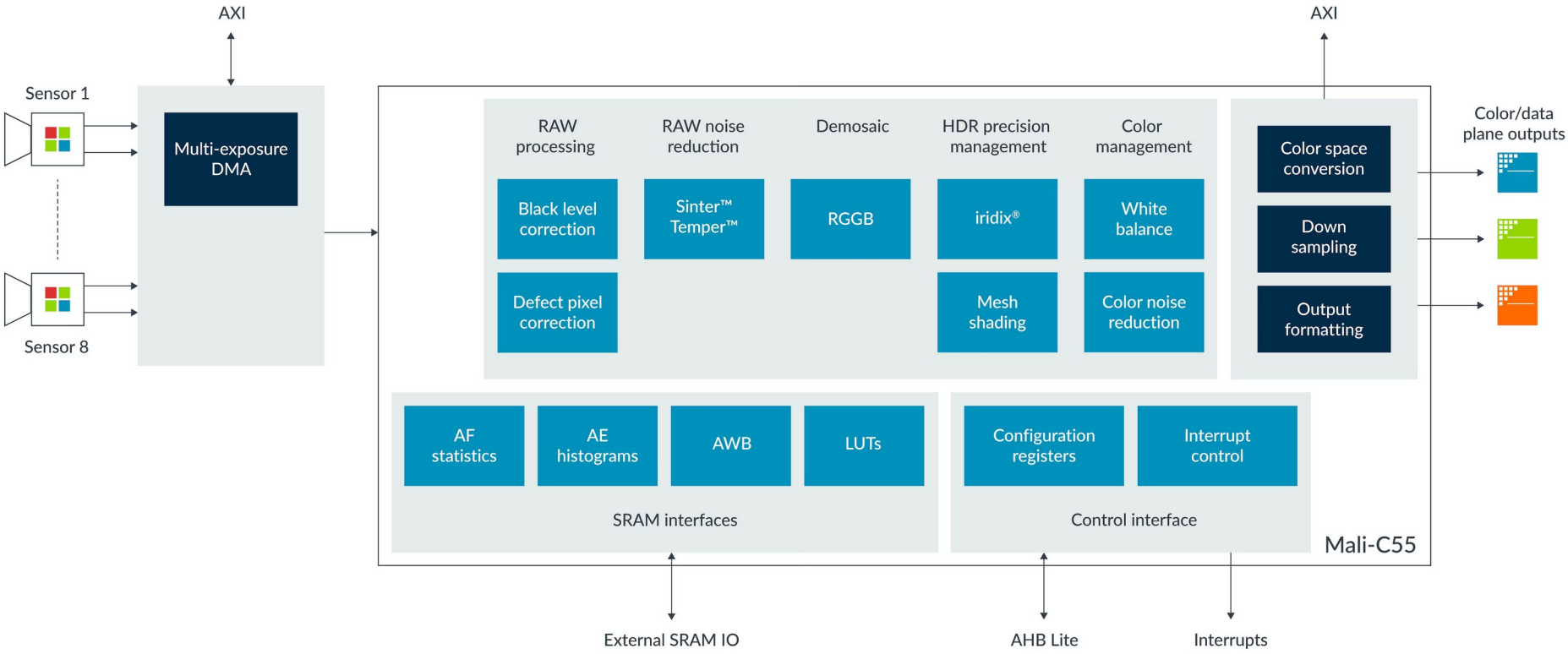
Open source support for Arm's Mali-C55 ISP

Daniel Scally
dan.scally@ideasonboard.com

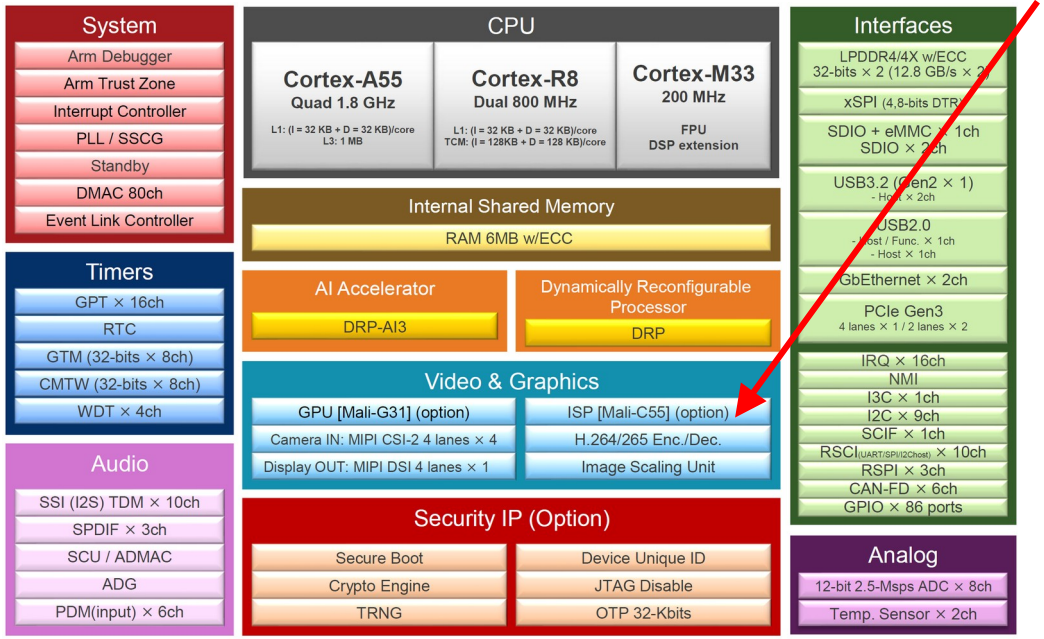
- Introduction
- The hardware
- The kernel
- libcamera
- Future challenges
- How can you use what we’ve done?
- Any questions?

Hi, I’m Dan

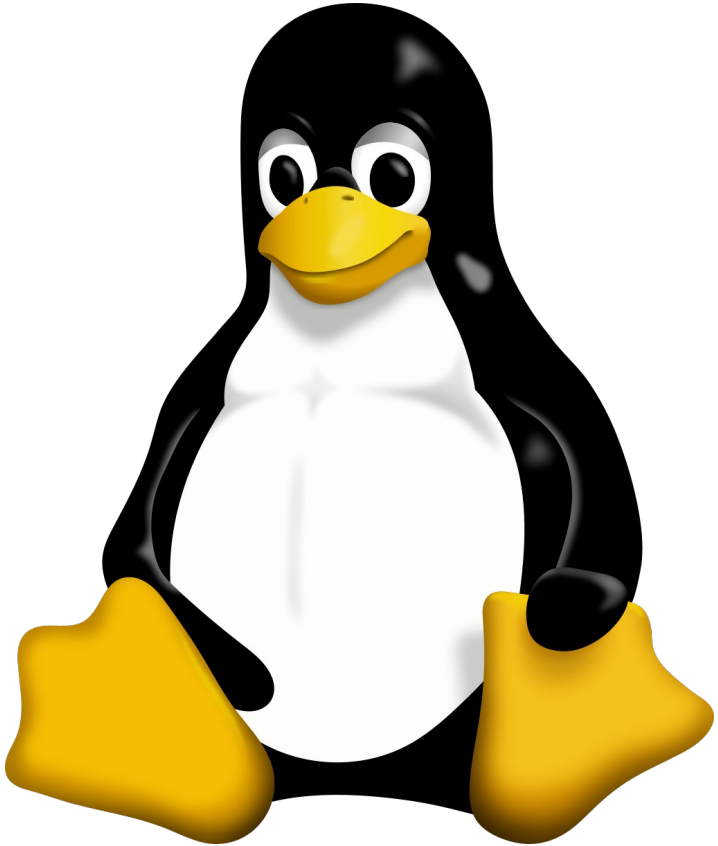
Arm’s Mali-C55 ISP

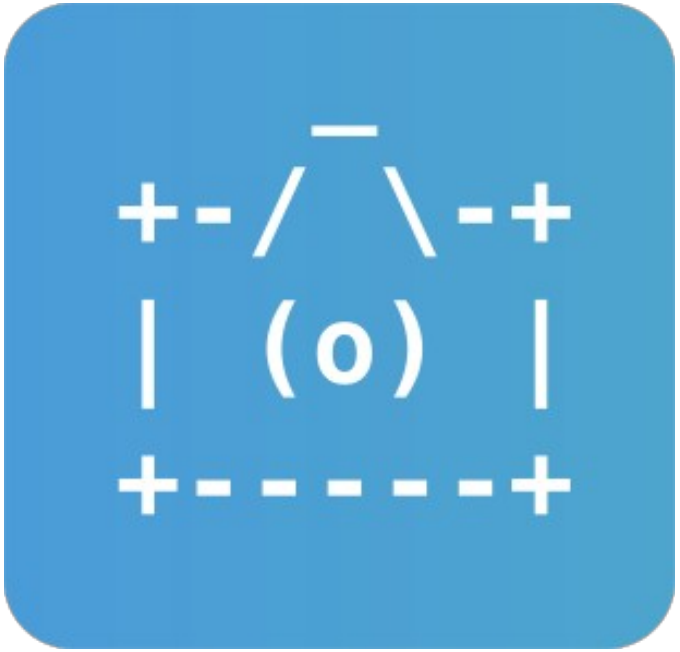


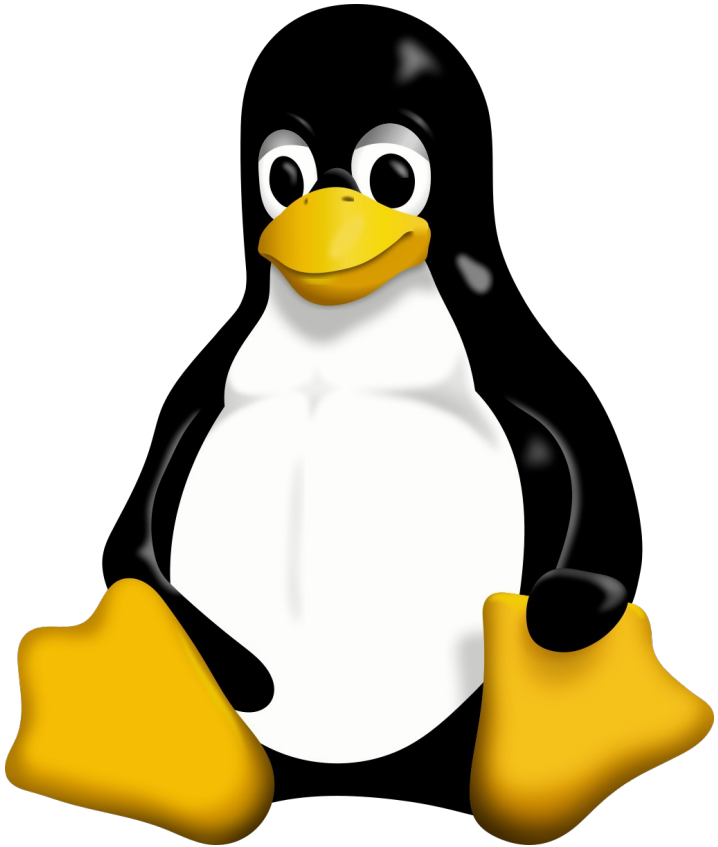
Where can I get one?



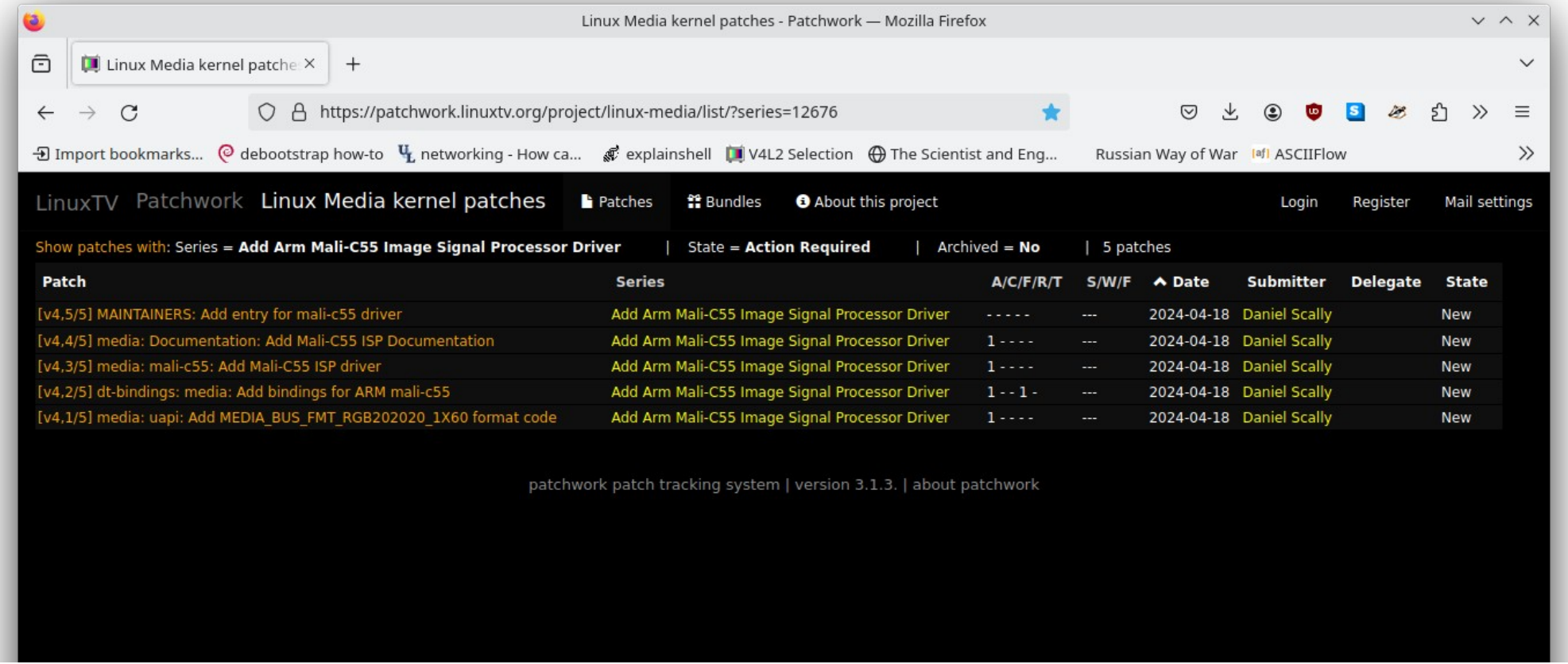
- Renesas RZ/V2H is the SoC you want (but it's an option, so be careful)
- RZ/V2H-EVK has it
- IMDT has a V2H SOM and SBC
- Kakip might have one with a form factor like Rpi in the future




_
+-/ \-+
| (o) |
+-----+



_
+-/ \-+
| (o) |
+-----+


Linux Media kernel patches - Patchwork — Mozilla Firefox
Linux Media kernel patche
https://patchwork.linuxtv.org/project/linux-media/list/?series=12676
Import bookmarks...
debootstrap how-to
networking - How ca...
explainshell
V4L2 Selection
The Scientist and Eng...
Russian Way of War
ASCIIFlow
LinuxTV Patchwork Linux Media kernel patches
Patches
Bundles
About this project
Login
Register
Mail settings
Show patches with: Series = Add Arm Mali-C55 Image Signal Processor Driver | State = Action Required | Archived = No | 5 patches
Patch
Series
A/C/F/R/T
S/W/F
Date
Submitter
Delegate
State
[v4,5/5] MAINTAINERS: Add entry for mali-c55 driver
Add Arm Mali-C55 Image Signal Processor Driver
- - - - -

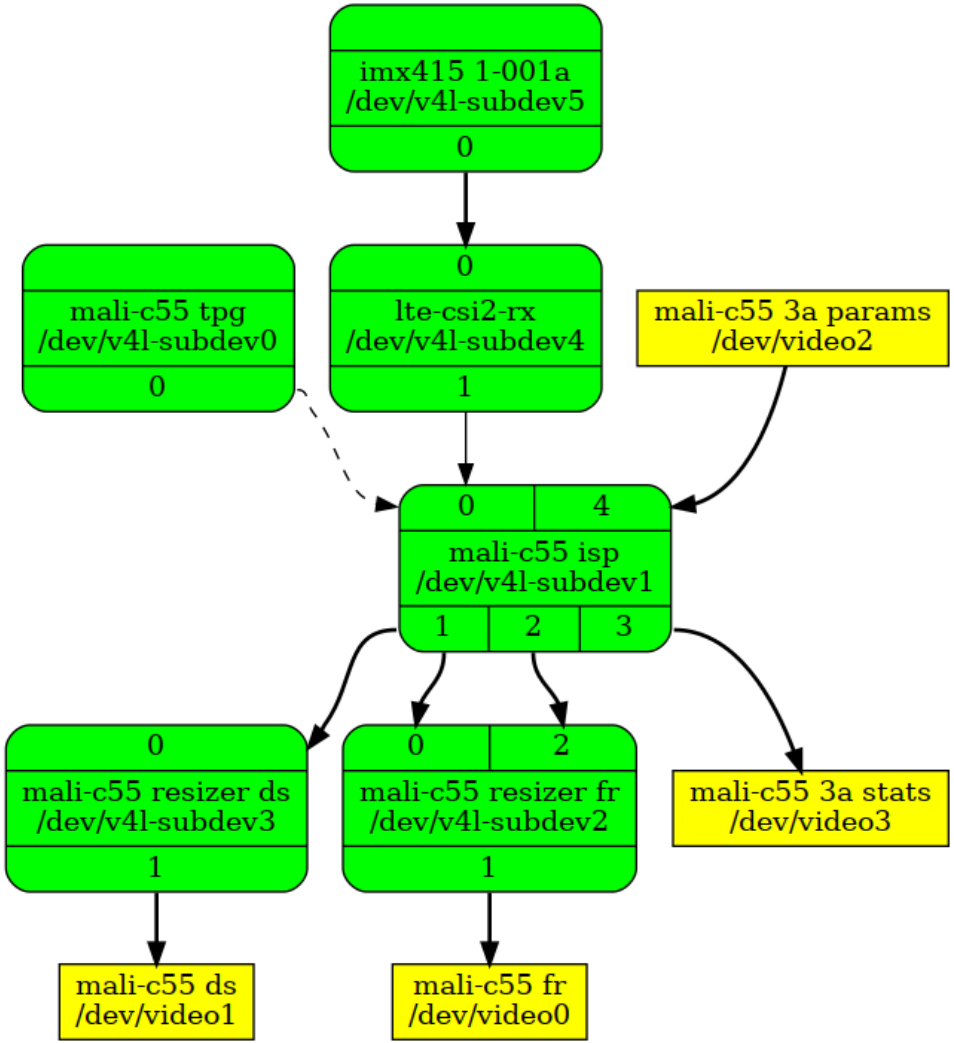
2024-04-18
Daniel Scally
New
[v4,4/5] media: Documentation: Add Mali-C55 ISP Documentation
Add Arm Mali-C55 Image Signal Processor Driver
1 - - - -

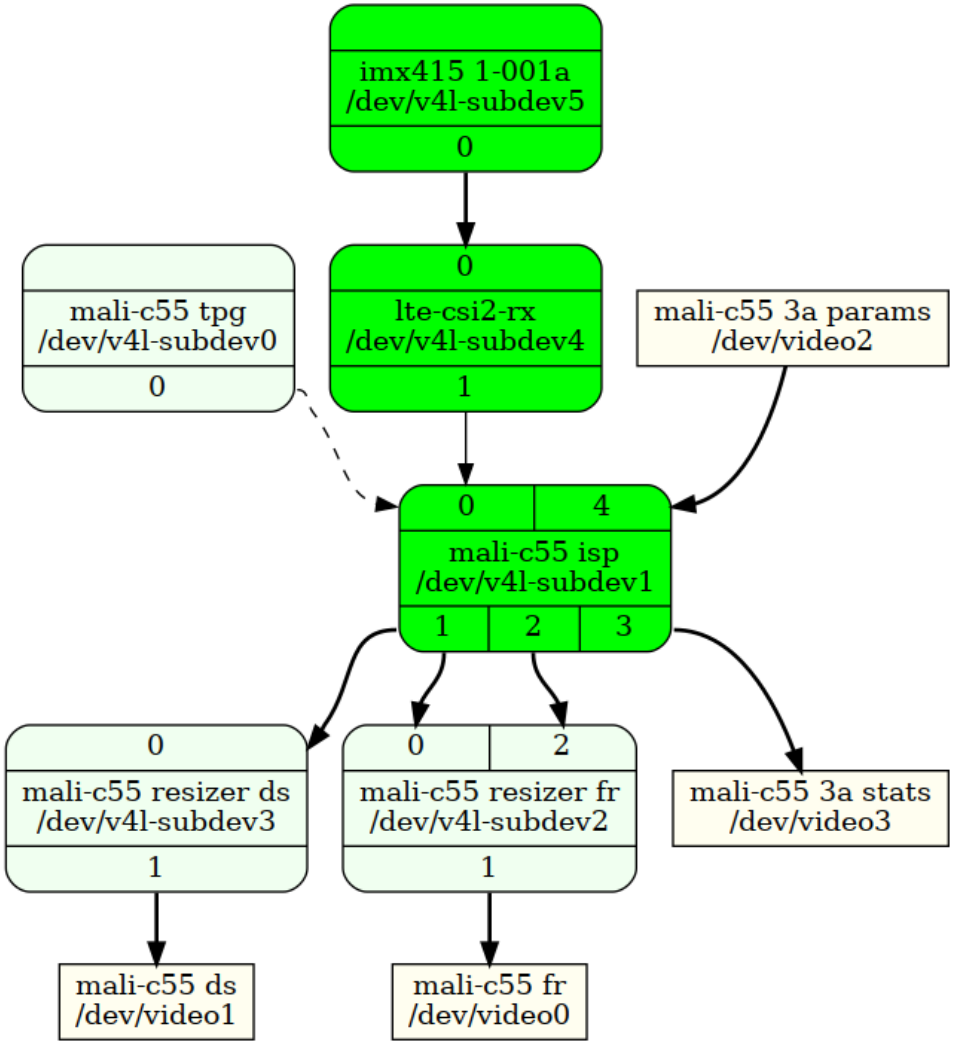
2024-04-18
Daniel Scally
New
[v4,3/5] media: mali-c55: Add Mali-C55 ISP driver
Add Arm Mali-C55 Image Signal Processor Driver
1 - - - -

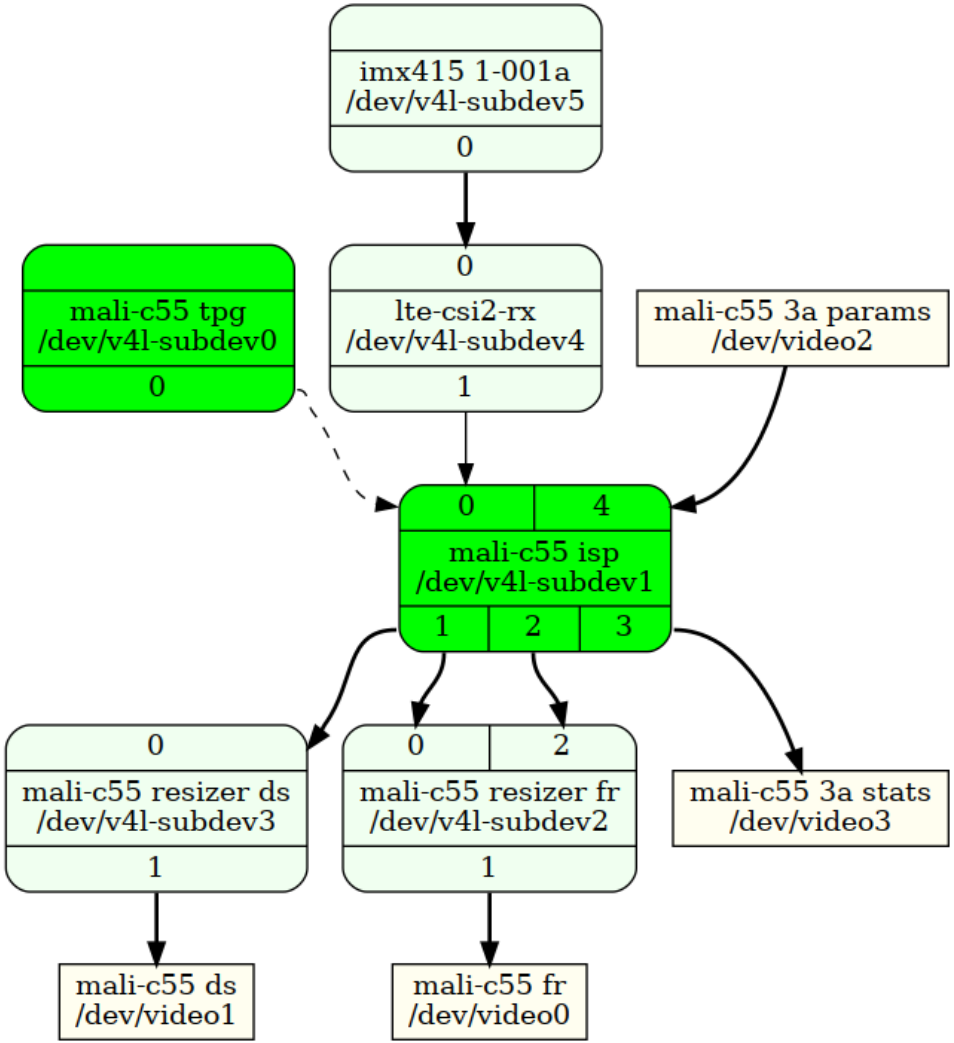
2024-04-18
Daniel Scally
New
[v4,2/5] dt-bindings: media: Add bindings for ARM mali-c55
Add Arm Mali-C55 Image Signal Processor Driver
1 - - 1 -

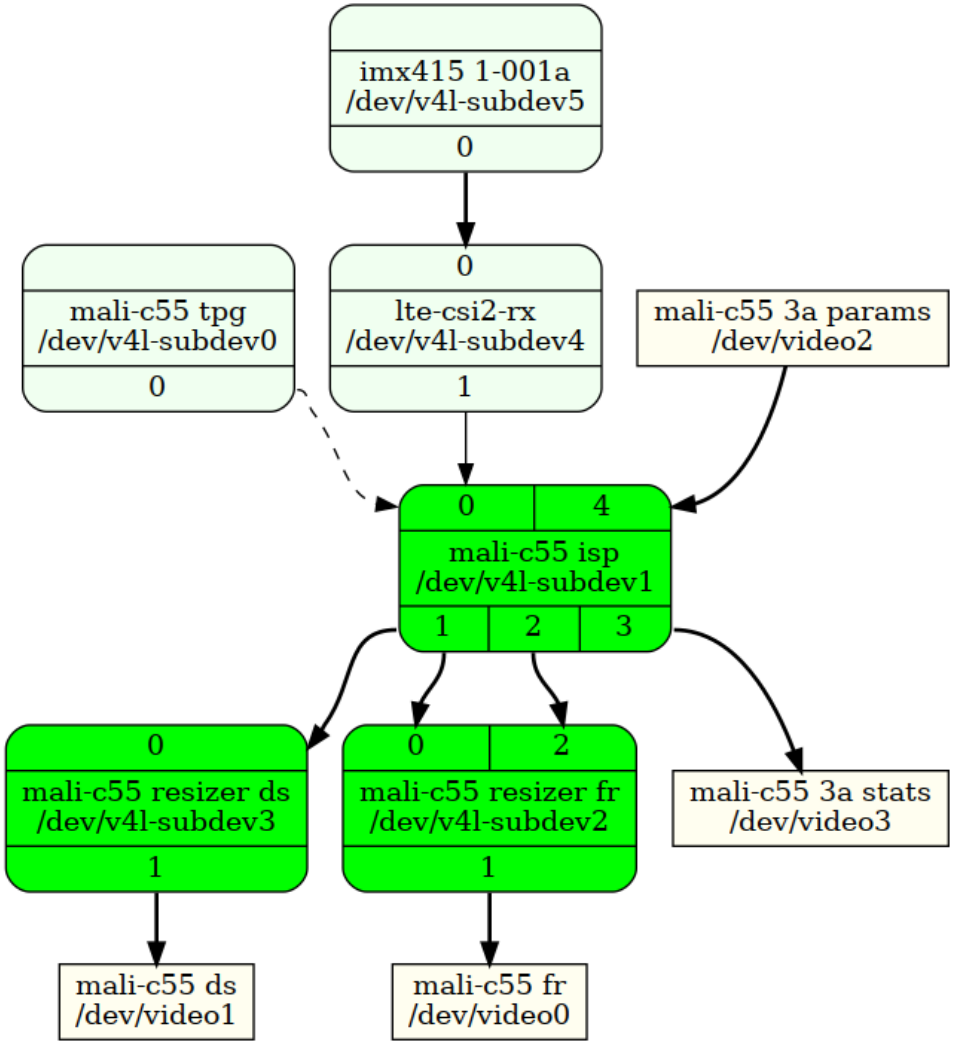
2024-04-18
Daniel Scally
New
[v4,1/5] media: uapi: Add MEDIA_BUS_FMT_RGB202020_1X60 format code
Add Arm Mali-C55 Image Signal Processor Driver
1 - - - -

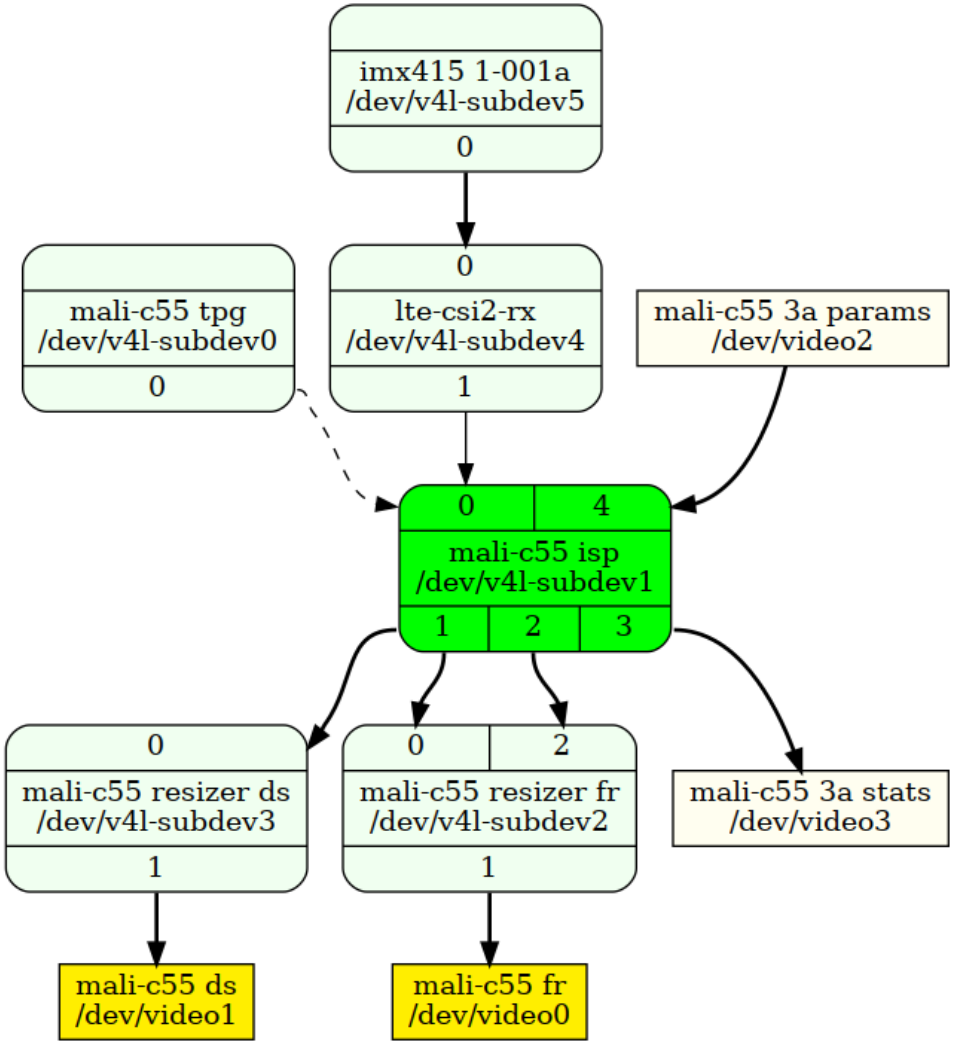
2024-04-18
Daniel Scally
New
patchwork patch tracking system | version 3.1.3. | about patchwork

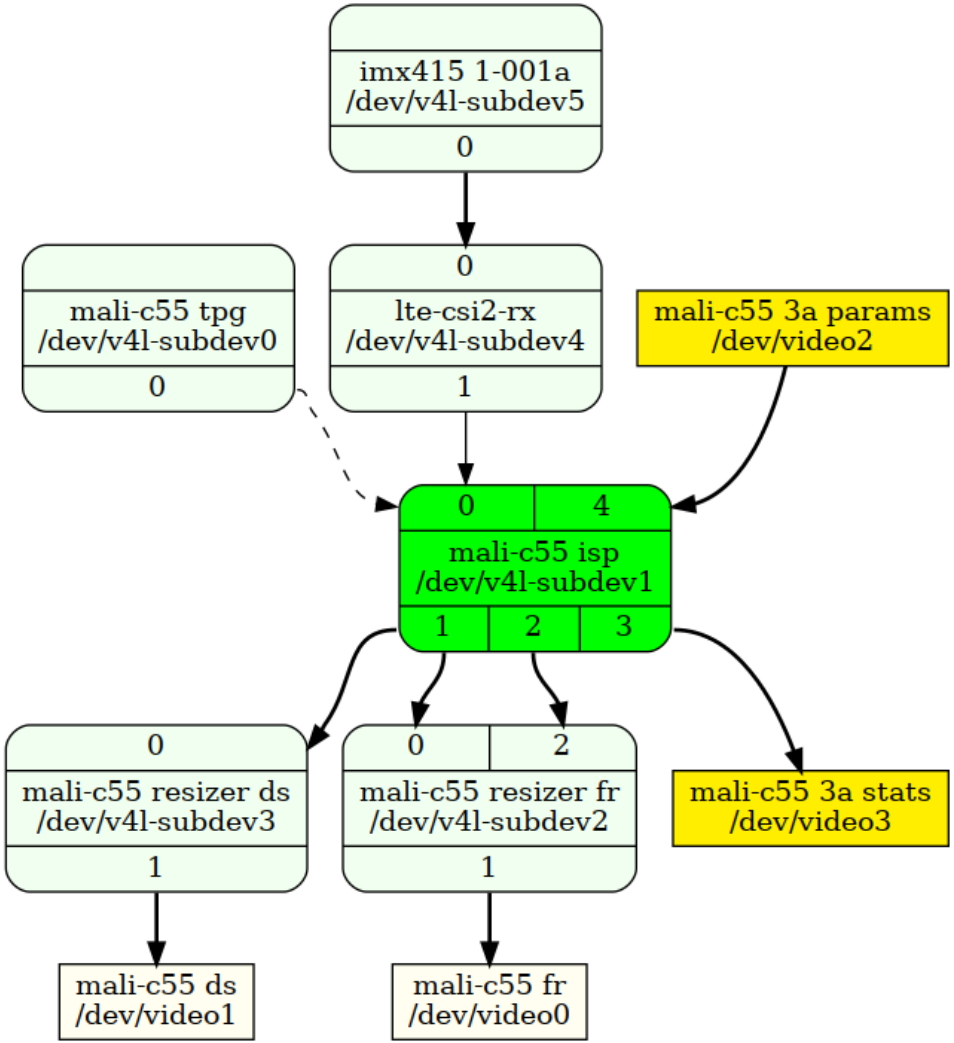

imx415 1-001a
/dev/v4l-subdev5
0
mali-c55 tpg
/dev/v4l-subdev0
0
0
lte-csi2-rx
/dev/v4l-subdev4
1
mali-c55 3a params
/dev/video2
0
4
mali-c55 isp
/dev/v4l-subdev1
1
2
3
0
mali-c55 resizer ds
/dev/v4l-subdev3
1
0
2
mali-c55 resizer fr
/dev/v4l-subdev2
1
mali-c55 3a stats
/dev/video3
mali-c55 ds
/dev/video1
mali-c55 fr
/dev/video0


imx415 1-001a
/dev/v4l-subdev5
0
0
lte-csi2-rx
/dev/v4l-subdev4
1
mali-c55 tpg
/dev/v4l-subdev0
0
mali-c55 3a params
/dev/video2
0
4
mali-c55 isp
/dev/v4l-subdev1
1
2
3
0
mali-c55 resizer ds
/dev/v4l-subdev3
1
0
2
mali-c55 resizer fr
/dev/v4l-subdev2
1
mali-c55 3a stats
/dev/video3
mali-c55 ds
/dev/video1
mali-c55 fr
/dev/video0


imx415 1-001a
/dev/v4l-subdev5
0
0
lte-csi2-rx
/dev/v4l-subdev4
1
mali-c55 tpg
/dev/v4l-subdev0
0
mali-c55 3a params
/dev/video2
0
4
mali-c55 isp
/dev/v4l-subdev1
1
2
3
0
mali-c55 resizer ds
/dev/v4l-subdev3
1
0
2
mali-c55 resizer fr
/dev/v4l-subdev2
1
mali-c55 3a stats
/dev/video3
mali-c55 ds
/dev/video1
mali-c55 fr
/dev/video0


imx415 1-001a
/dev/v4l-subdev5
0
mali-c55 tpg
/dev/v4l-subdev0
0
0
lte-csi2-rx
/dev/v4l-subdev4
1
mali-c55 3a params
/dev/video2
0
4
mali-c55 isp
/dev/v4l-subdev1
1
2
3
0
mali-c55 resizer ds
/dev/v4l-subdev3
1
0
2
mali-c55 resizer fr
/dev/v4l-subdev2
1
mali-c55 3a stats
/dev/video3
mali-c55 ds
/dev/video1
mali-c55 fr
/dev/video0


imx415 1-001a
/dev/v4l-subdev5
0
mali-c55 tpg
/dev/v4l-subdev0
0
0
lte-csi2-rx
/dev/v4l-subdev4
1
mali-c55 3a params
/dev/video2
0
4
mali-c55 isp
/dev/v4l-subdev1
1
2
3
0
mali-c55 resizer ds
/dev/v4l-subdev3
1
0
2
mali-c55 resizer fr
/dev/v4l-subdev2
1
mali-c55 3a stats
/dev/video3
mali-c55 ds
/dev/video1
mali-c55 fr
/dev/video0


imx415 1-001a
/dev/v4l-subdev5
0
mali-c55 tpg
/dev/v4l-subdev0
0
0
lte-csi2-rx
/dev/v4l-subdev4
1
mali-c55 3a params
/dev/video2
0
4
mali-c55 isp
/dev/v4l-subdev1
1
2
3
0
mali-c55 resizer ds
/dev/v4l-subdev3
1
0
2
mali-c55 resizer fr
/dev/v4l-subdev2
1
mali-c55 3a stats
/dev/video3
mali-c55 ds
/dev/video1
mali-c55 fr
/dev/video0

Usually parameters are monolithic

```
struct mali_c55_params_buffer {
        struct {
                __u32 chan00;
                __u32 chan01;
                __u32 chan10;
                __u32 chan11;
                _u8 field_flags;
        } sensor_off_preshading;
        struct {
                __u8 skip_x;
                __u8 offset_x;
                __u8 skip_y;
                __u8 offset_y;
                __u8 scale_bottom;
                __u8 scale_top;
                __u8 plane_mode;
                __u8 tap_point;
                __u8 field_flags;
        } aexp_hist;

        // Plus lots more sub-structs...

        __u8 block_flags
};
```

We've gone for a common header...

```
struct mali_c55_params_block_header {
        enum mali_c55_param_block_type type;
        bool enabled;
        size_t size;
};
```

```
enum mali_c55_param_block_type {
    MALI_C55_PARAM_BLOCK_SENSOR_OFFS,
    MALI_C55_PARAM_BLOCK_AEXP_HIST,
    MALI_C55_PARAM_BLOCK_AEXP_IHIST,
    MALI_C55_PARAM_BLOCK_AEXP_HIST_WEIGHT,
    MALI_C55_PARAM_BLOCK_AEXP_IHIST_WEIGHTS,
    MALI_C55_PARAM_BLOCK_DIGITAL_GAIN,
    MALI_C55_PARAM_BLOCK_AWB_GAINS,
    MALI_C55_PARAM_BLOCK_AWB_CONFIG,
    MALI_C55_PARAM_BLOCK_AWB_GAINS_AEXP,
    MALI_C55_PARAM_MESH_SHADING_CONFIG,
    MALI_C55_PARAM_MESH_SHADING_SELECTION,
    MALI_C55_PARAM_BLOCK_SENTINEL,
};
```

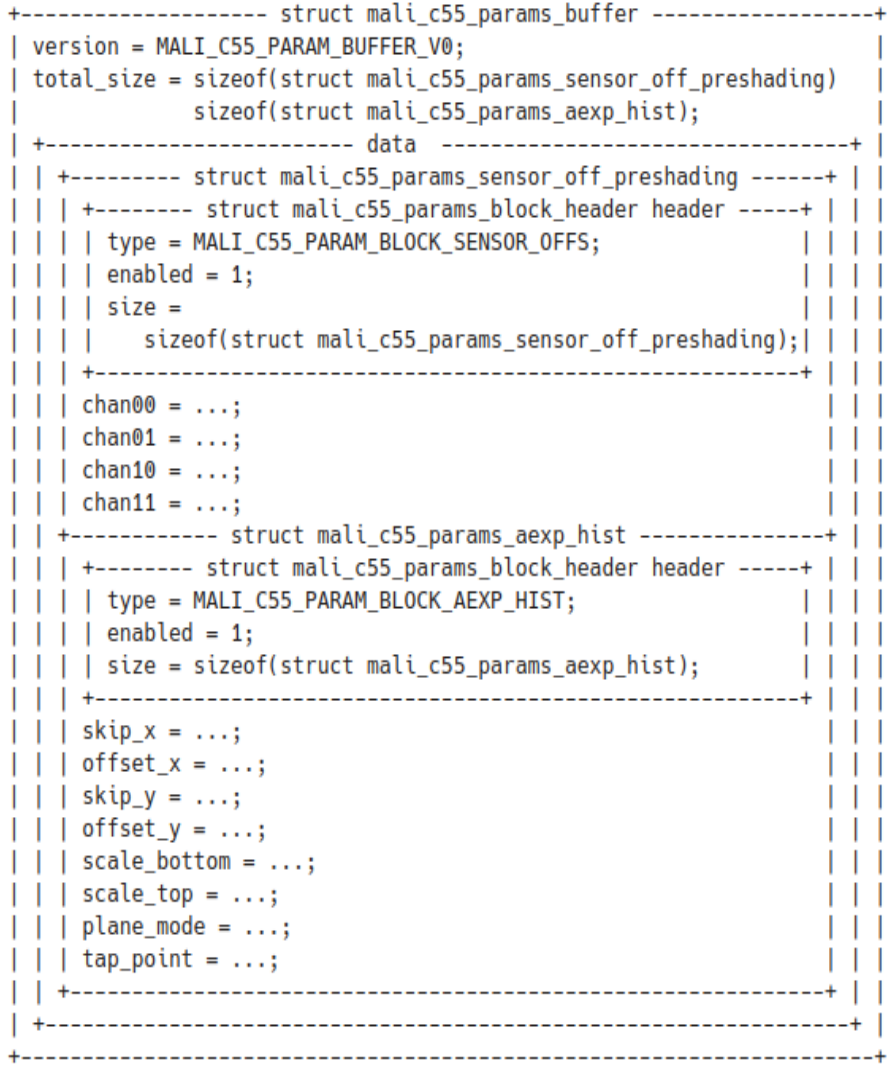
...embedded into structs for each block...

```
struct mali_c55_params_sensor_off_preshading {
    struct mali_c55_params_block_header header;
    __u32 chan00;
    __u32 chan01;
    __u32 chan10;
    __u32 chan11;
};
```

```
struct mali_c55_params_aexp_hist {
    struct mali_c55_params_block_header header;
    __u8 skip_x;
    __u8 offset_x;
    __u8 skip_y;
    __u8 offset_y;
    __u8 scale_bottom;
    __u8 scale_top;
    __u8 plane_mode;
    __u8 tap_point;
};
```

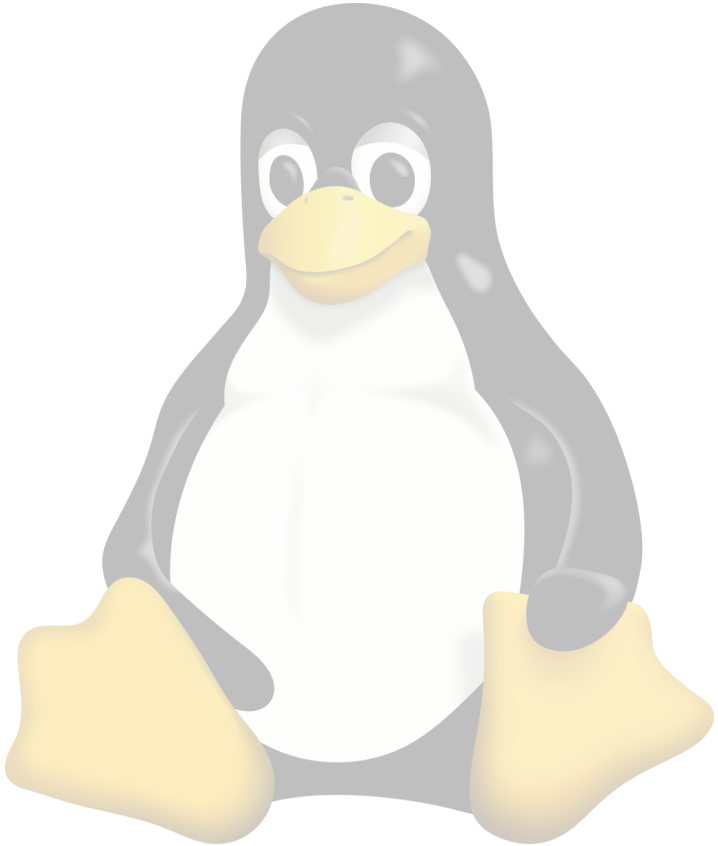
...and then those are packed into .data[]...

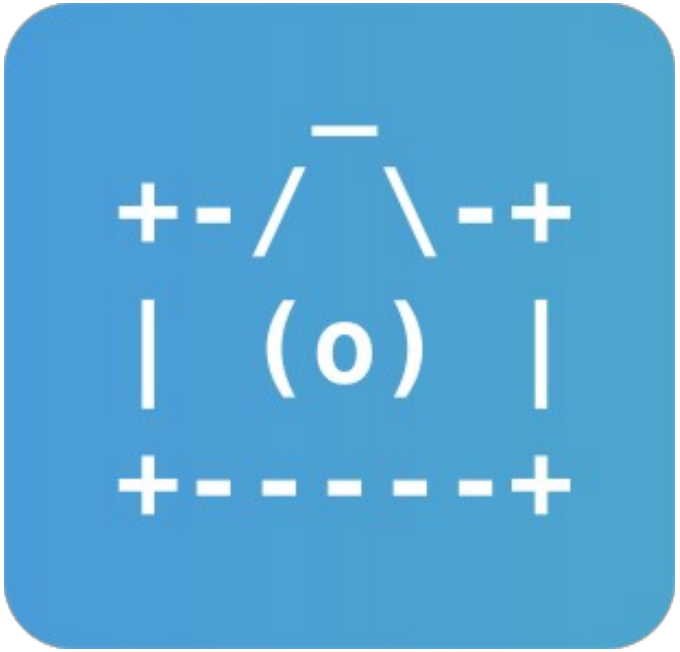
```
struct mali_c55_params_buffer {
        enum mali_c55_param_buffer_version version;
        size_t total_size;
        __u8 data[MALI_C55_PARAMS_MAX_SIZE];
};
```

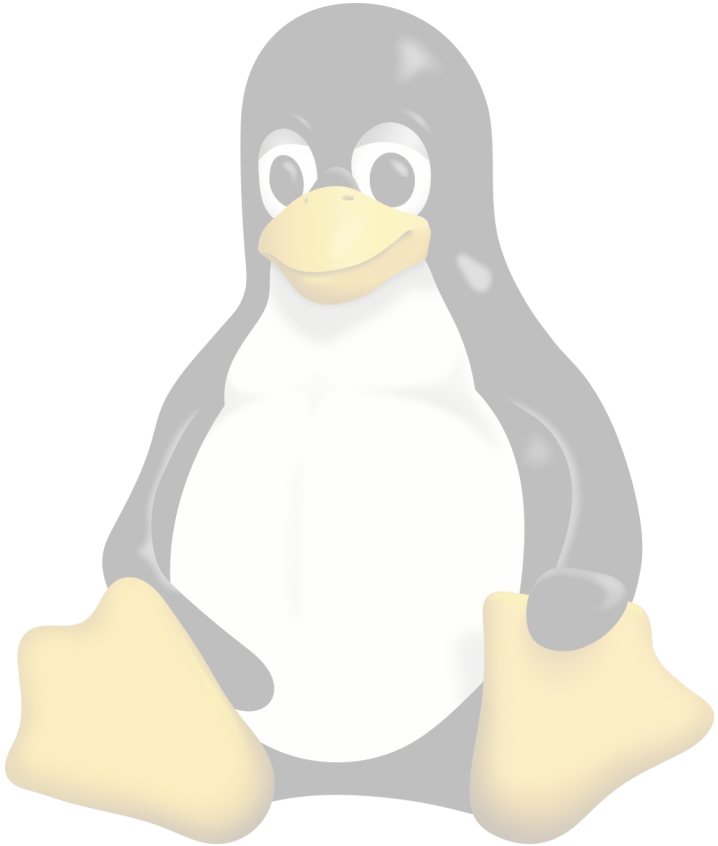


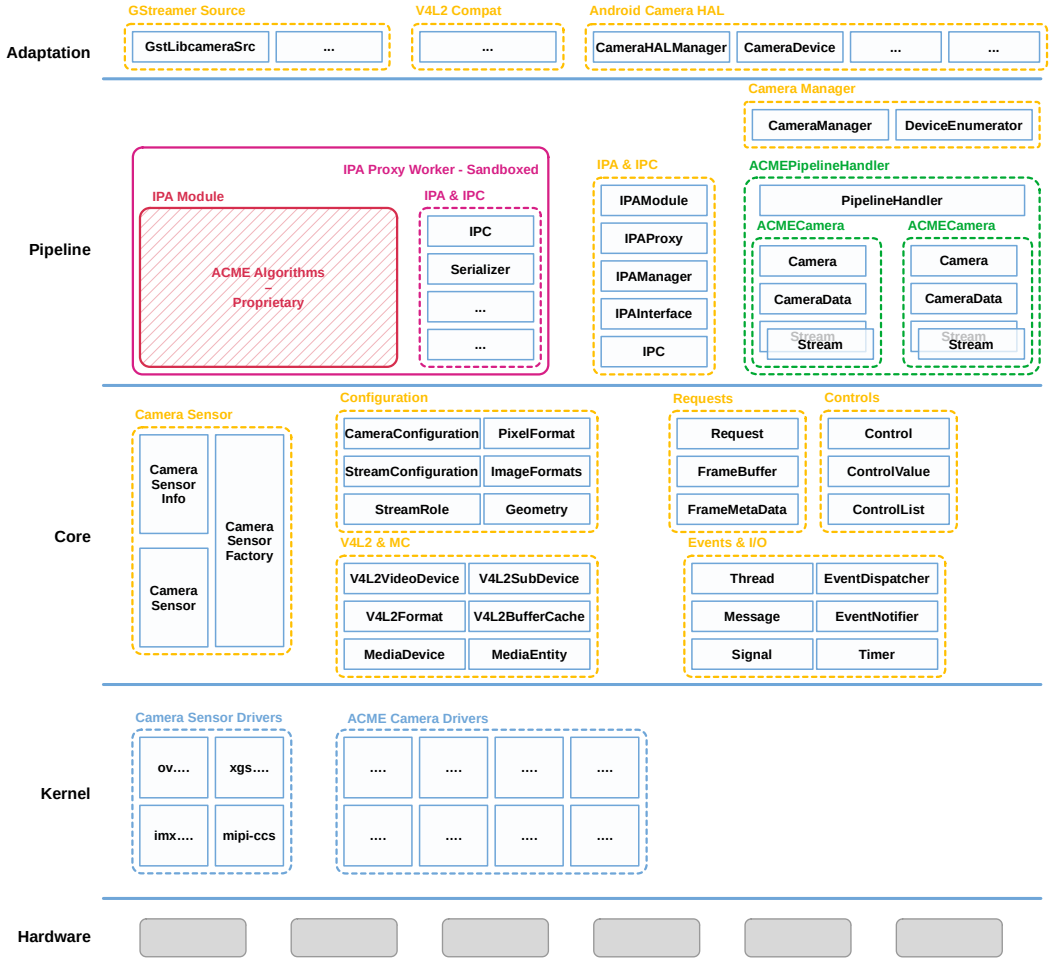
Then we can handle them in the driver with some type punning

```
struct mali_c55_params_buffer *config = vb2_plane_addr(...);
size_t block_offset = 0;

while (block_offset < config->total_size) {
    struct mali_c55_params_block_header *block;

    block = (struct mali_c55_params_block_header *)
            &config->data[block_offset];

    switch (block->type) {
    case MALI_C55_PARAM_BLOCK_SENSOR_OFFS:
            struct mali_c55_params_sensor_off_preshading *sensor_offs =
                    (struct mali_c55_params_sensor_off_preshading *)block;

            handle_sensor_offs(sensor_offs);
            break;
    ...
    }

    block_offset += block_size
}
```

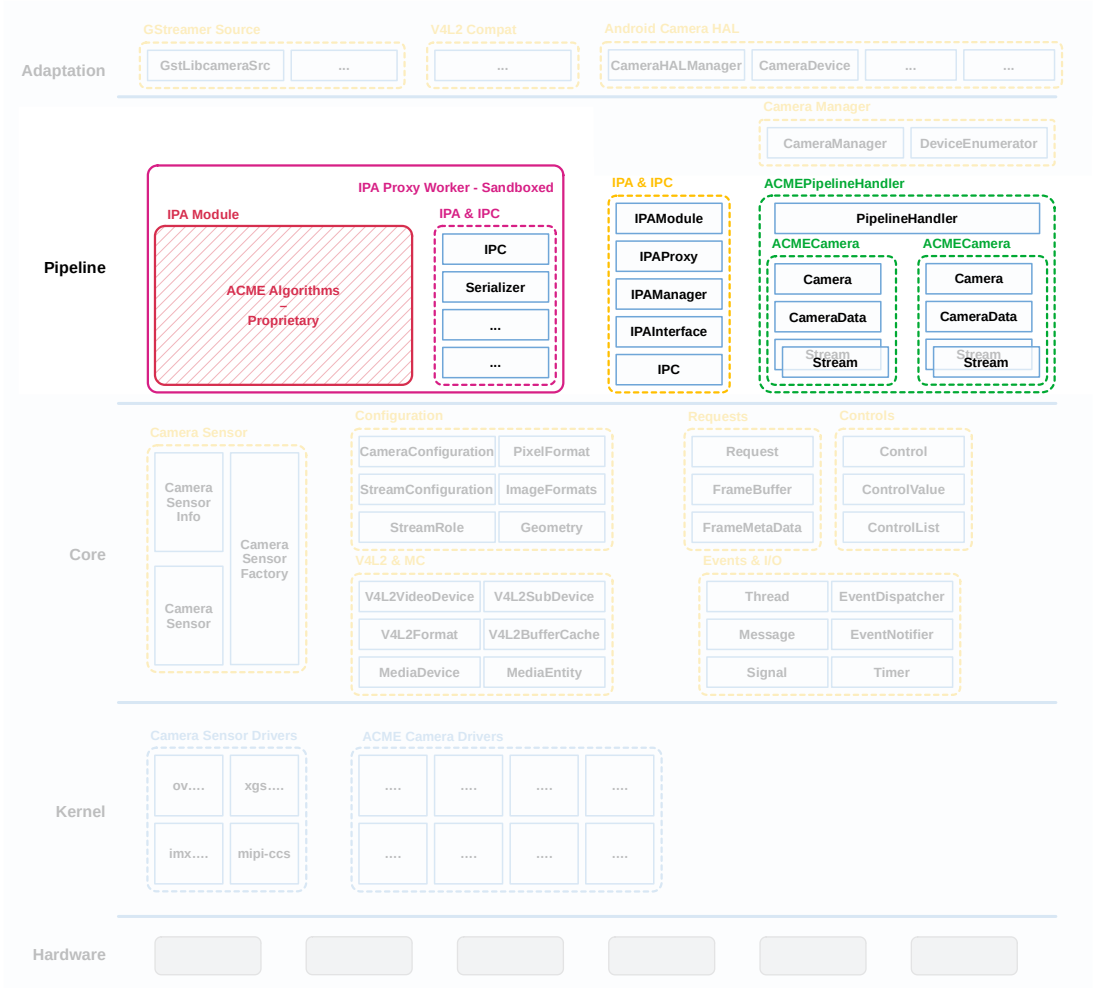



_
+-/ \-+
| (o) |
+-----+





The Camera Stack



The Pipeline

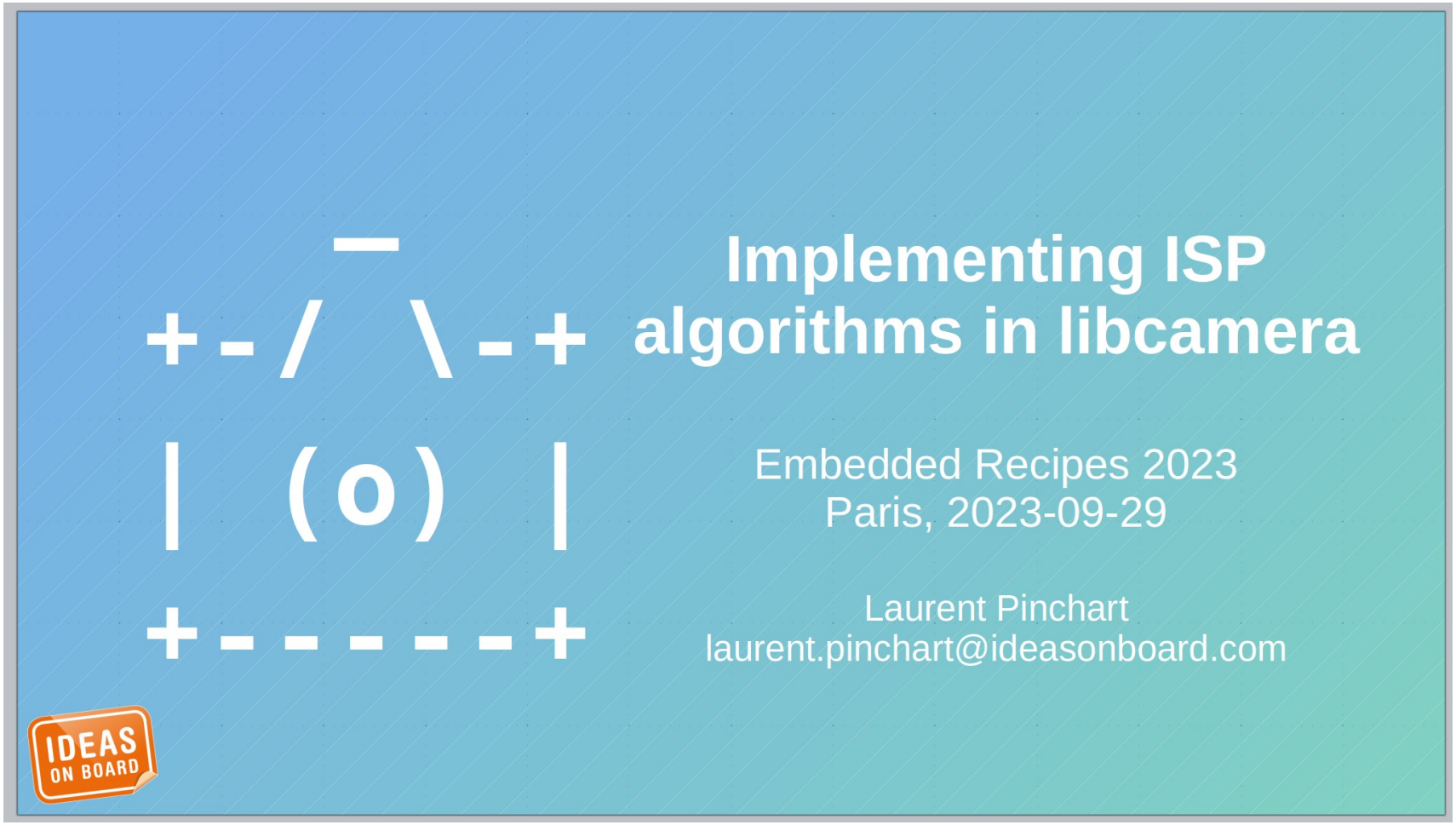
Learn How to Support Your SoC and ISP in Libcamera - Laurent Pinchart, Ideas on Board

The Linux Foundation
170K subscribers
Subscribe

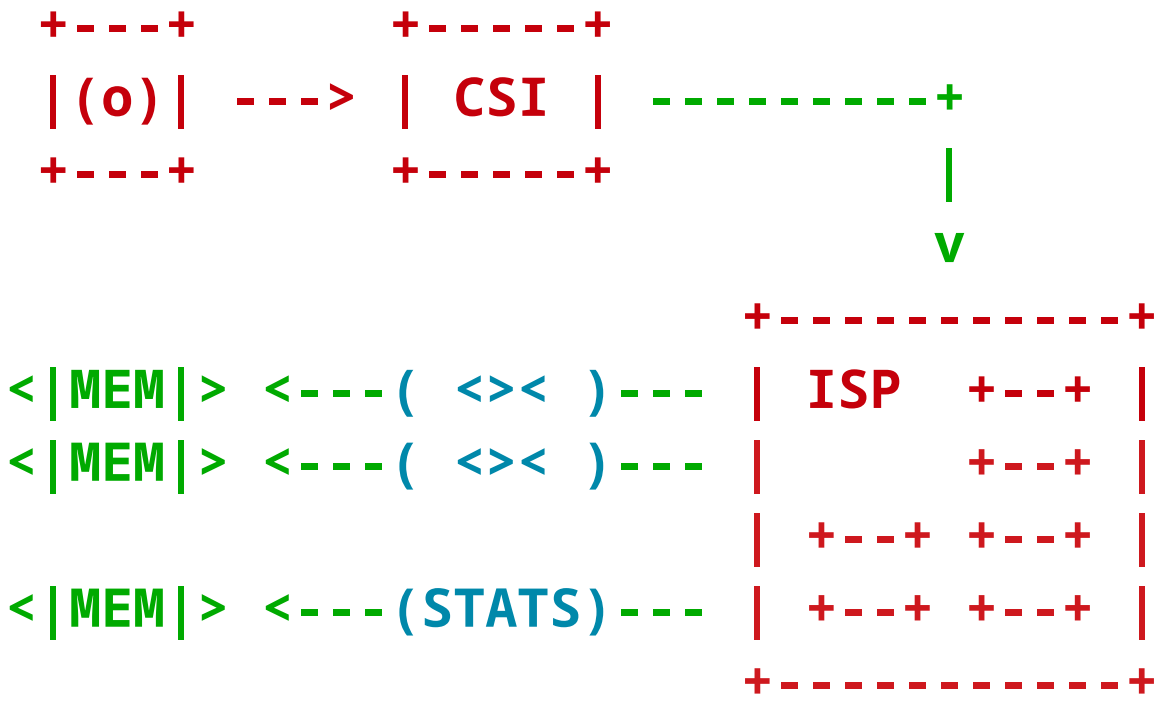
5 Share Save

IDEAS ON BOARD

https://www.youtube.com/watch?v=x9ndjdvXPl0

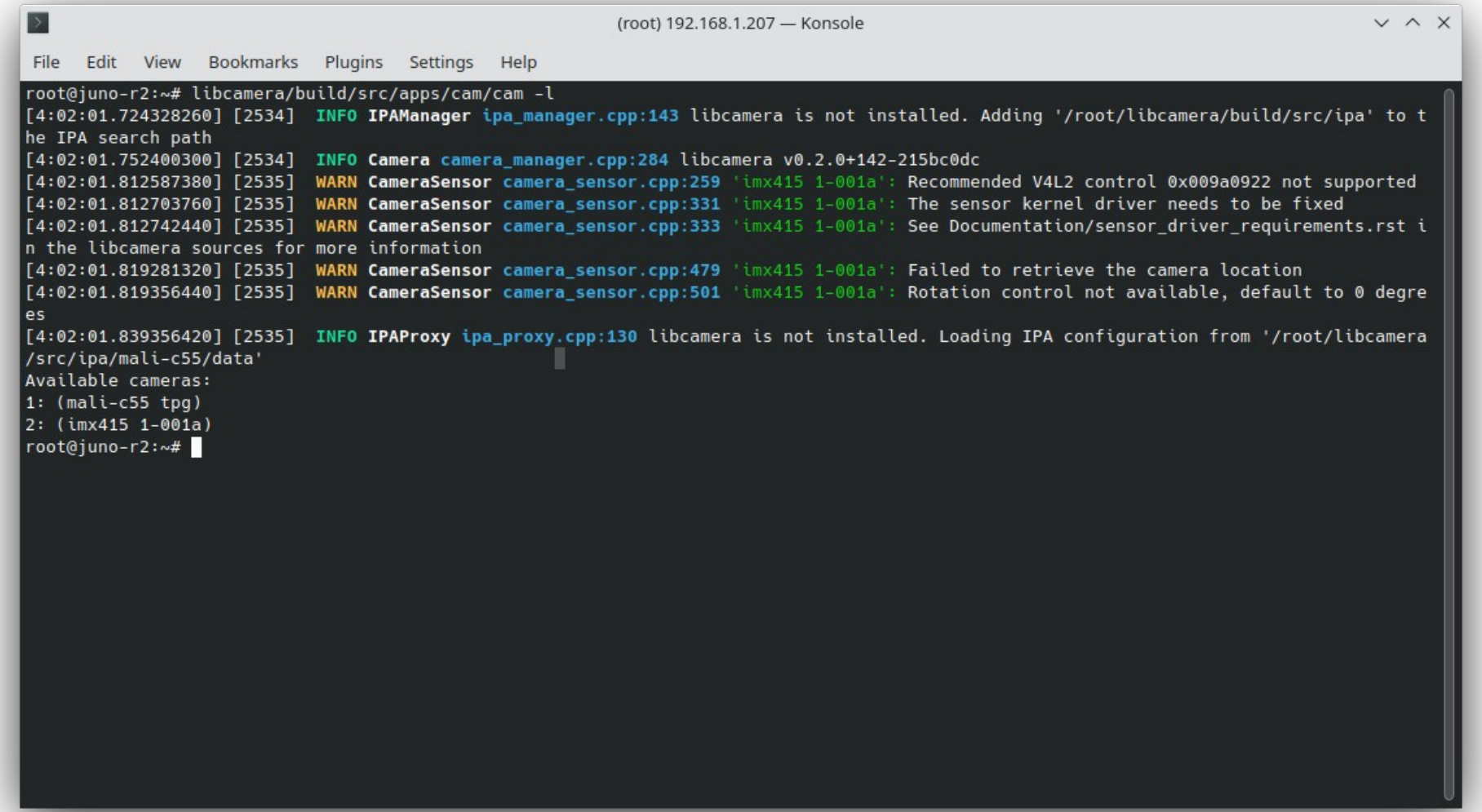


https://www.youtube.com/watch?v=RrHcgBa2f0U

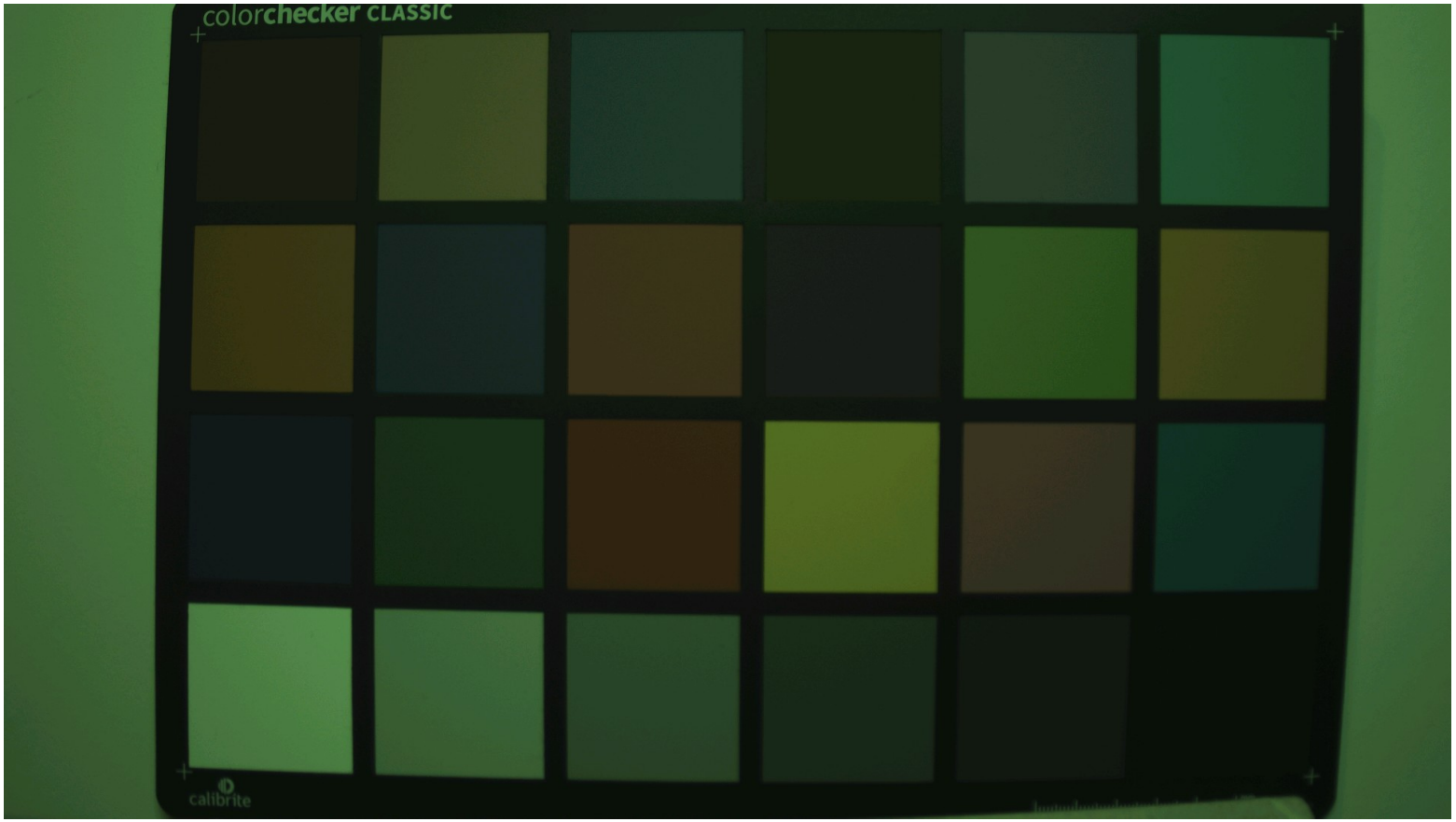


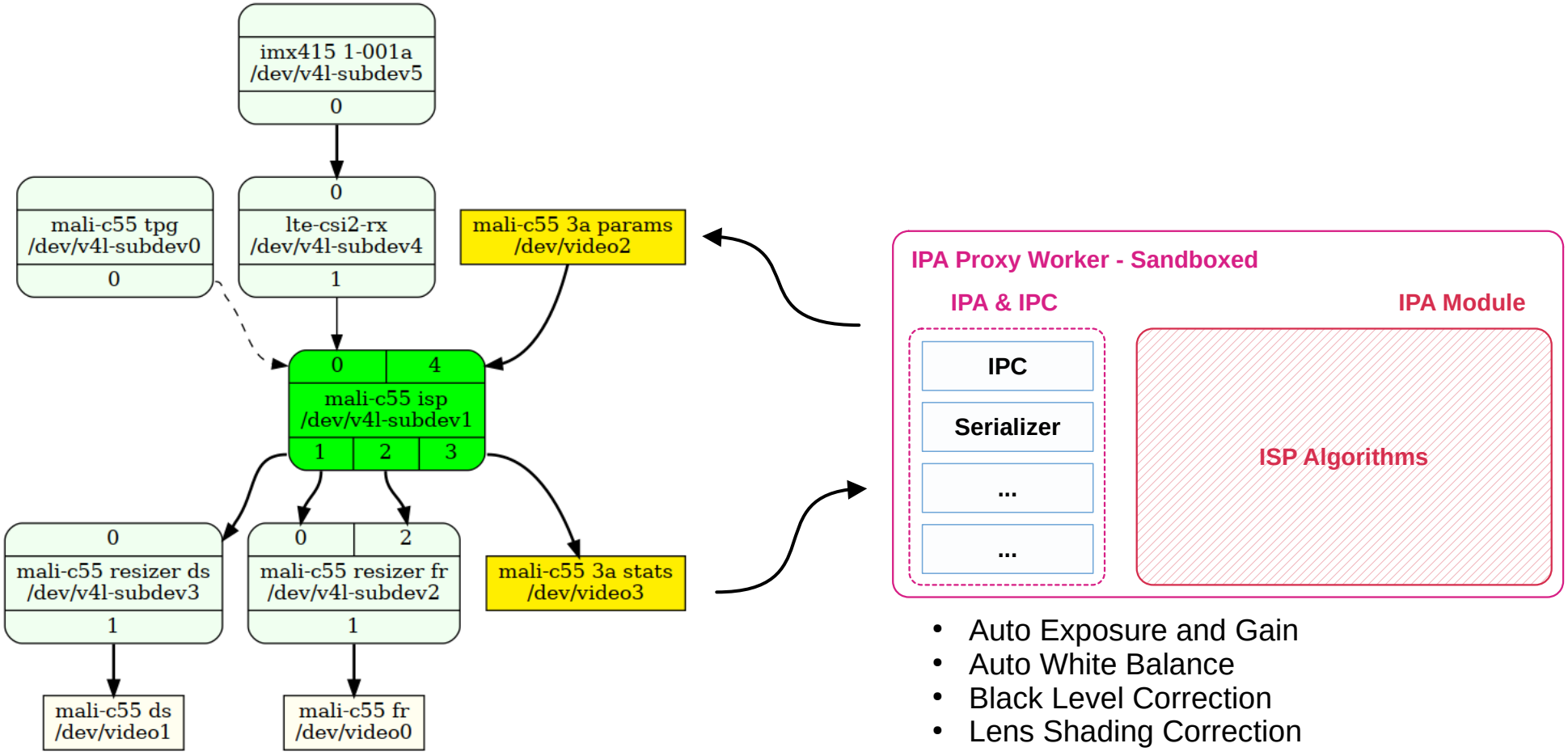
The pipeline handler interfaces with all kernel devices. It abstracts them and exposes video streams to upper layers.

The Pipeline Handler


(root) 192.168.1.207 — Konsole
File Edit View Bookmarks Plugins Settings Help
root@juno-r2:~# libcamera/build/src/apps/cam/cam -l
[4:02:01.724328260] [2534] INFO IPAManager ipa_manager.cpp:143 libcamera is not installed. Adding '/root/libcamera/build/src/ipa' to the IPA search path
[4:02:01.752400300] [2534] INFO Camera camera_manager.cpp:284 libcamera v0.2.0+142-215bc0dc
[4:02:01.812587380] [2535] WARN CameraSensor camera_sensor.cpp:259 'imx415 1-001a': Recommended V4L2 control 0x009a0922 not supported
[4:02:01.812703760] [2535] WARN CameraSensor camera_sensor.cpp:331 'imx415 1-001a': The sensor kernel driver needs to be fixed
[4:02:01.812742440] [2535] WARN CameraSensor camera_sensor.cpp:333 'imx415 1-001a': See Documentation/sensor_driver_requirements.rst in the libcamera sources for more information
[4:02:01.819281320] [2535] WARN CameraSensor camera_sensor.cpp:479 'imx415 1-001a': Failed to retrieve the camera location
[4:02:01.819356440] [2535] WARN CameraSensor camera_sensor.cpp:501 'imx415 1-001a': Rotation control not available, default to 0 degrees
[4:02:01.839356420] [2535] INFO IPAProxy ipa_proxy.cpp:130 libcamera is not installed. Loading IPA configuration from '/root/libcamera/src/ipa/mali-c55/data'
Available cameras:
1: (mali-c55 tpg)
2: (imx415 1-001a)
root@juno-r2:~#

Hurray – you can take a picture

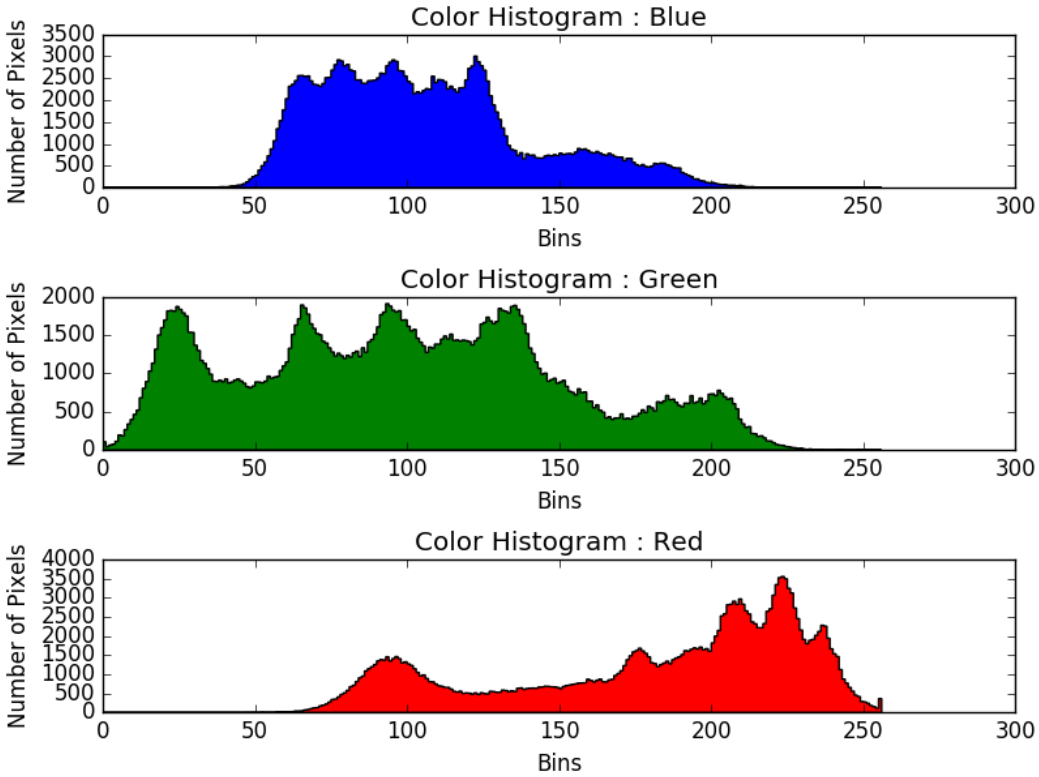




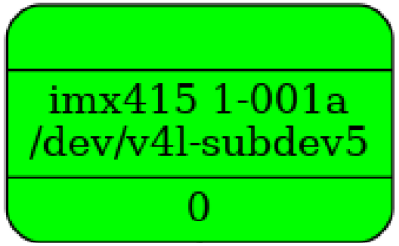
- Auto Exposure and Gain
- Auto White Balance
- Black Level Correction
- Lens Shading Correction

The IPA Interface

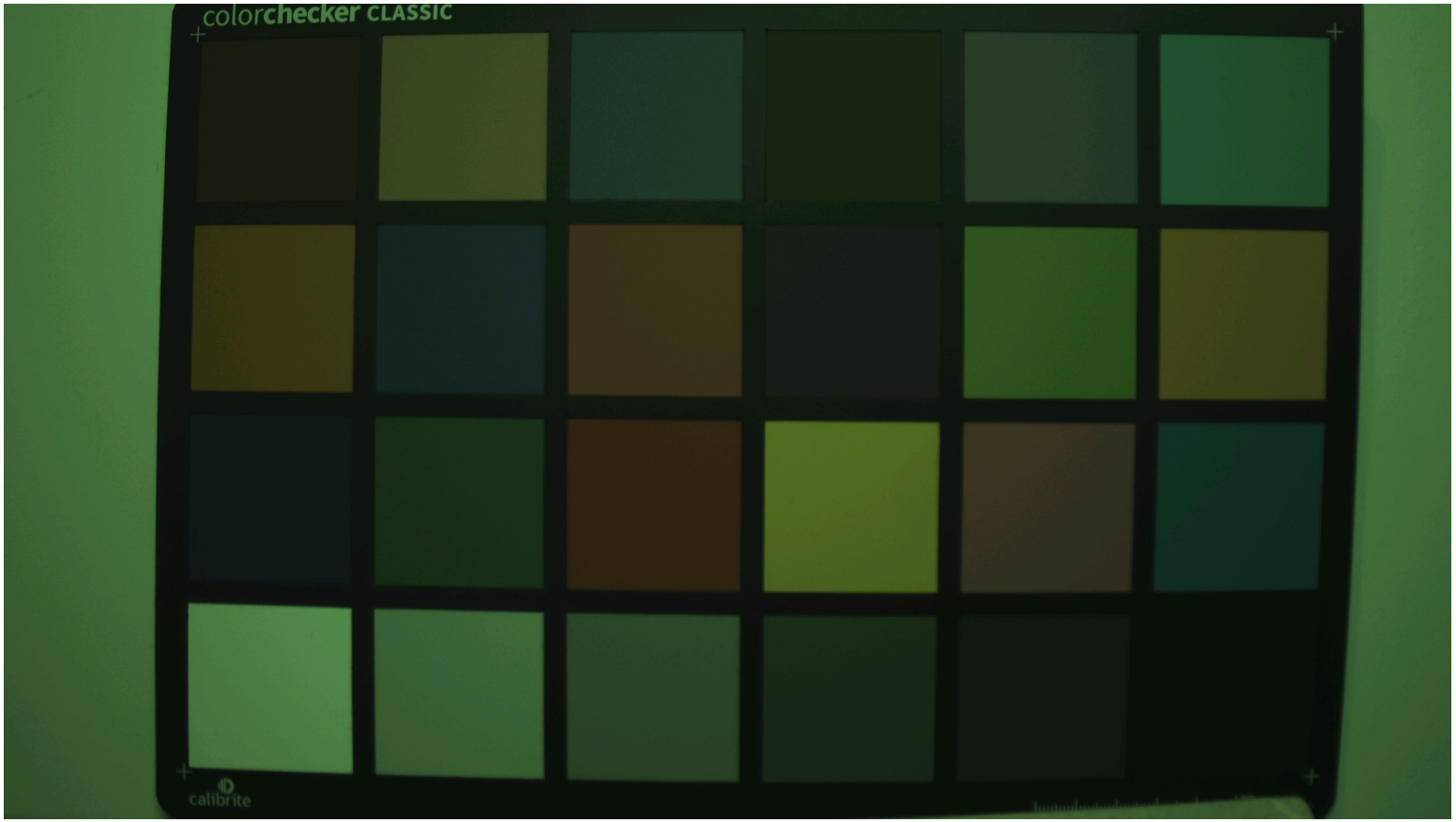
AEGC Histograms



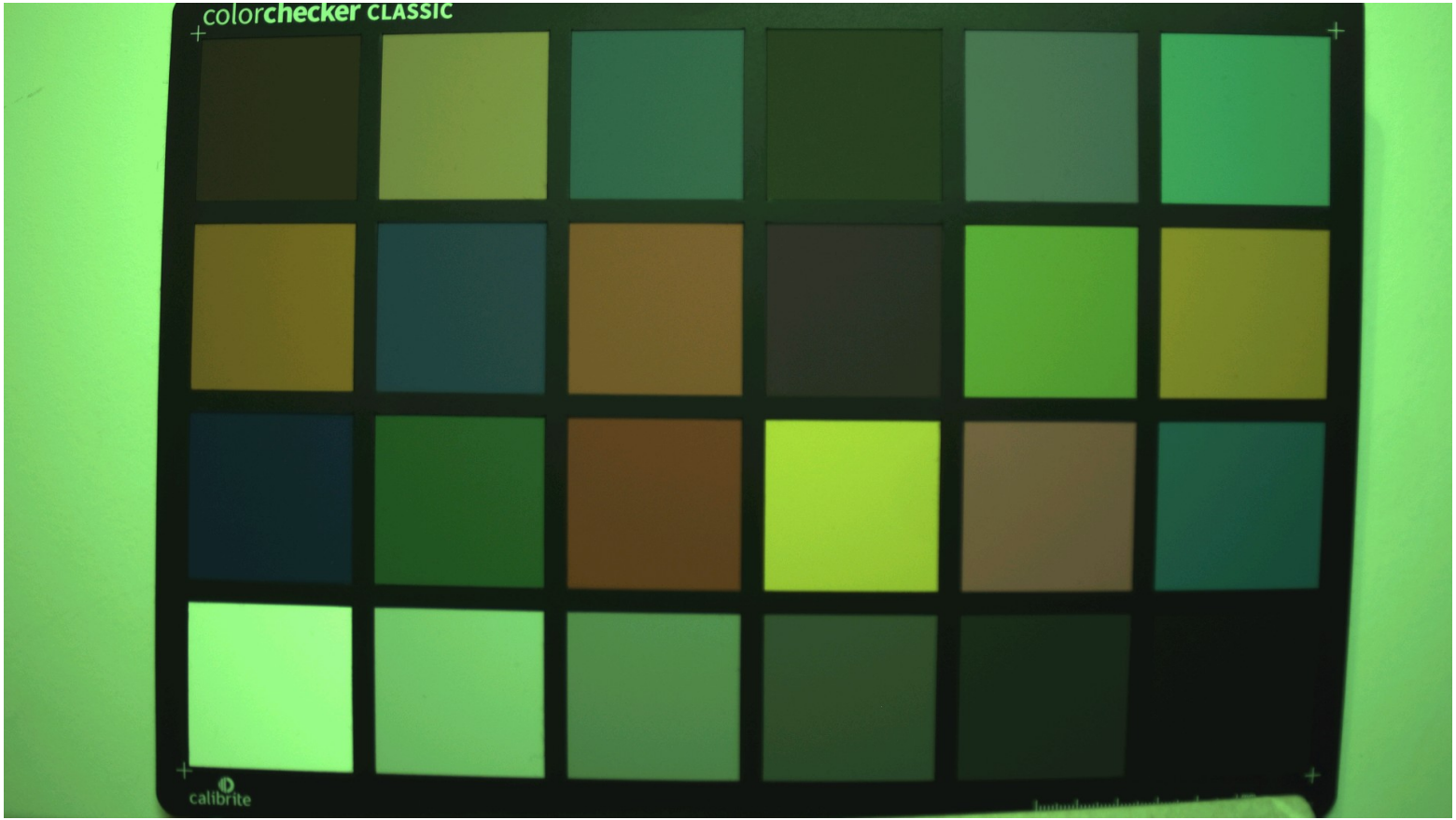

```
double yAvg = (rAvg *.299) + (gAvg * .587) + (bAvg * .114);

Duration effectiveExposure = thisFramesExposure * thisFramesGain;
Duration targetExposure = effectiveExposure * (yTgt / yAvg);

newExposure, newGain = splitUpExposure(targetExposure);
```



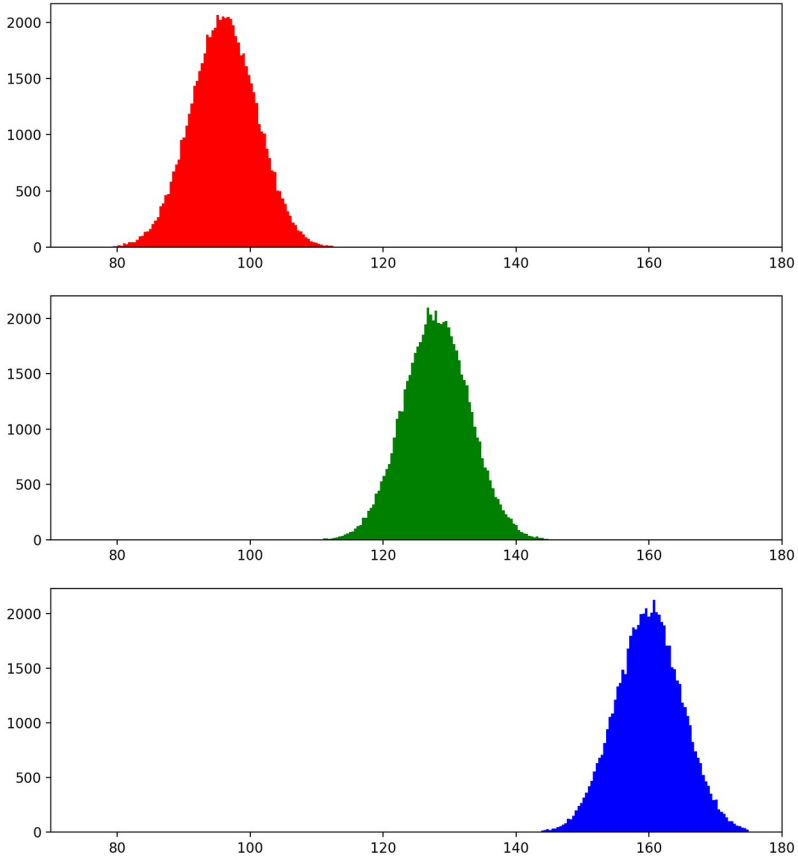
We go from a dim image



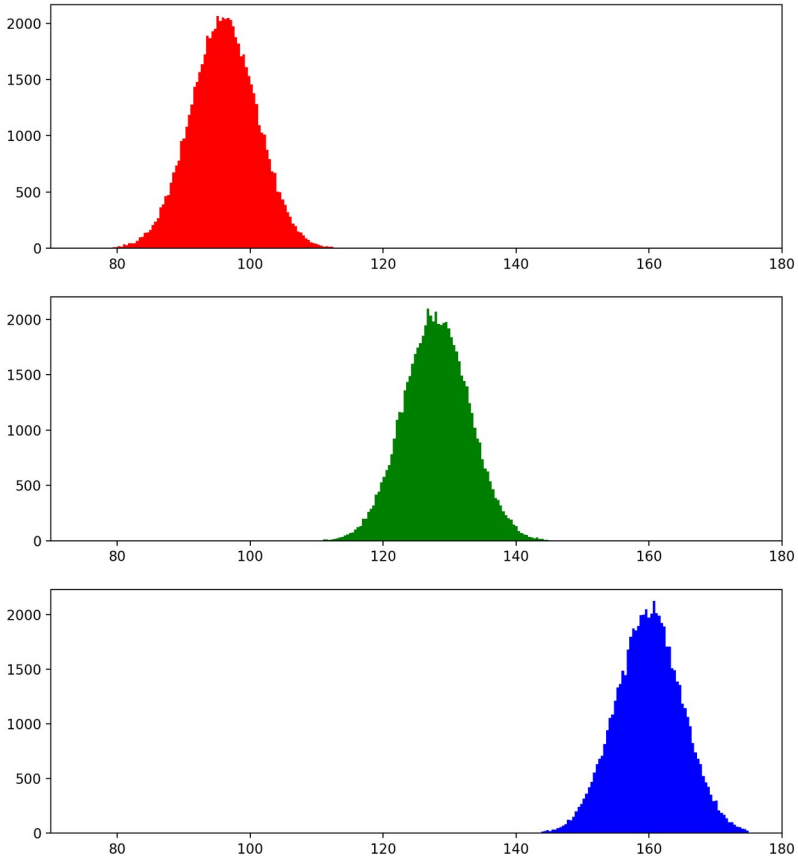
To a brighter one

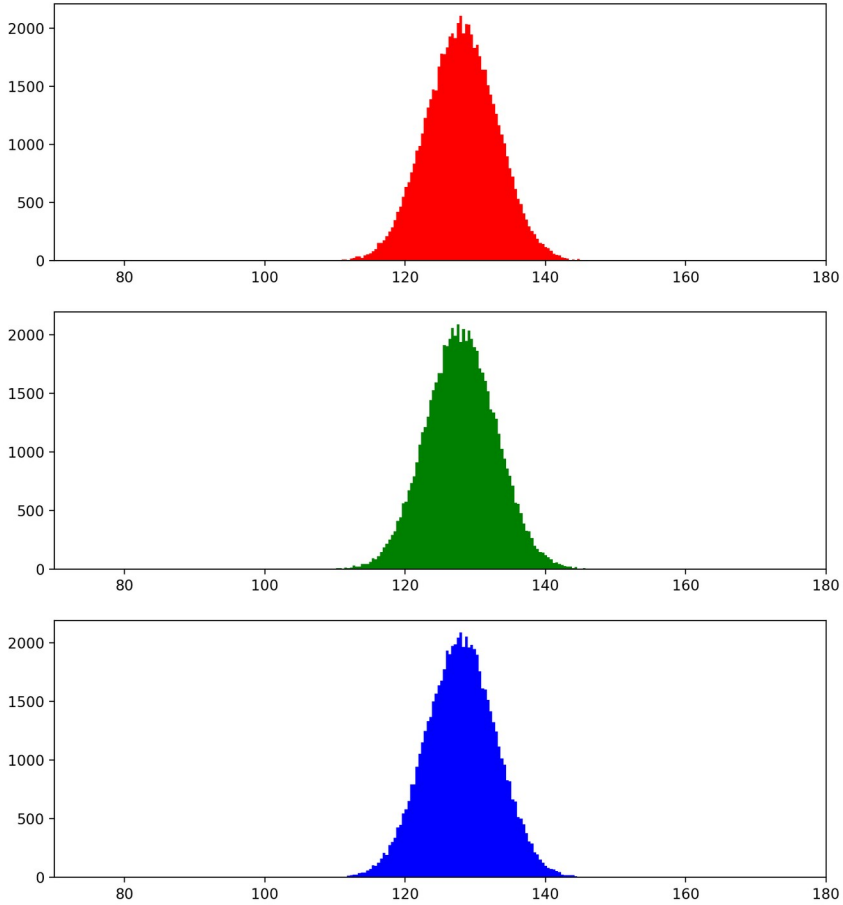


We aim to achieve a “grey world”



We aim to achieve a “grey world”





And the ISP makes it easy

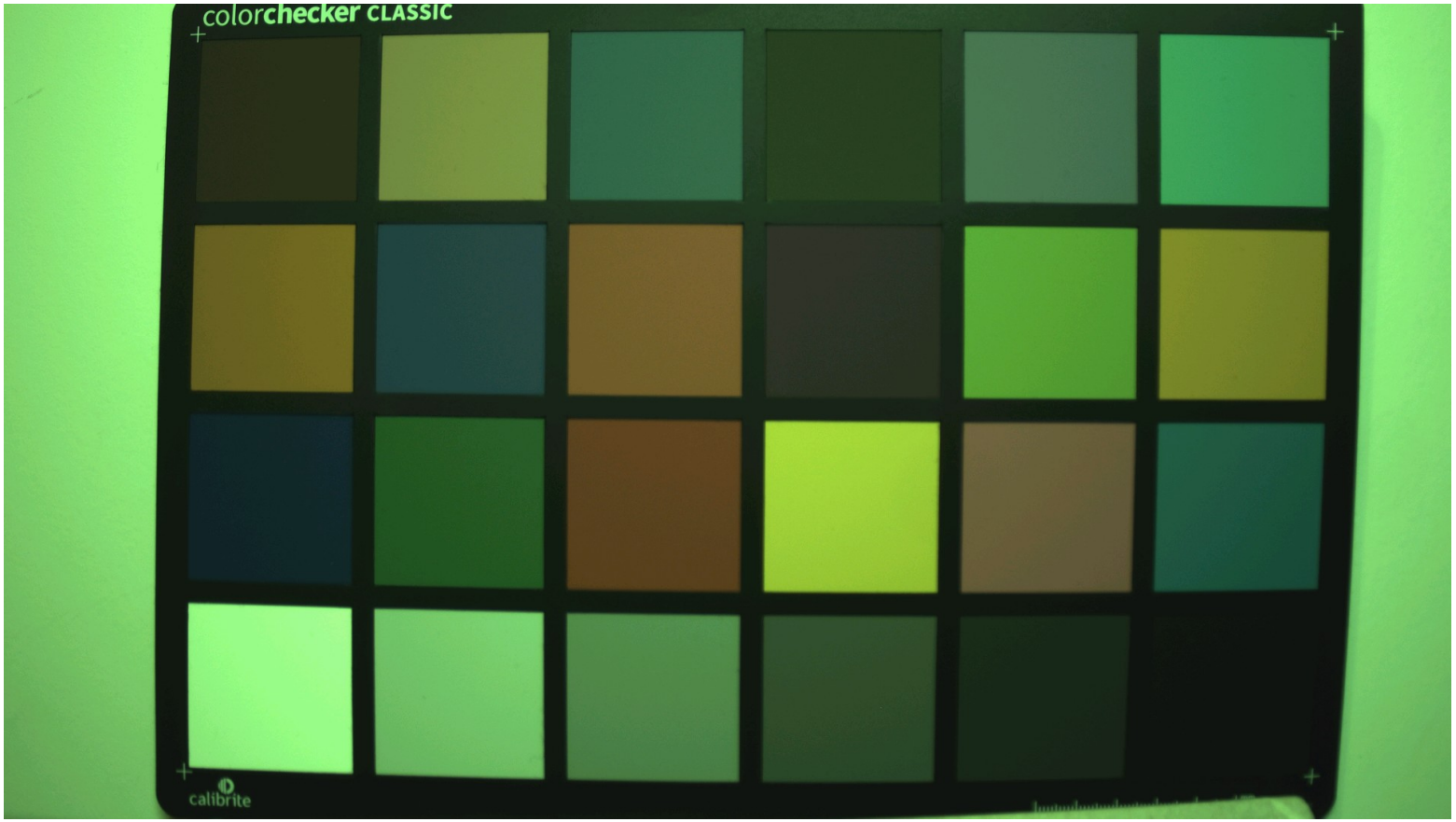
```
struct mali_c55_awb_average_ratios {
        __u16 avg_rg_gr;
        __u16 avg_bg_br;
        __u32 num_pixels;
}


/*
 *      gain00 = R
 *      gain01 = Gr
 *      gain10 = Gb
 *      gain11 = B
 */

struct mali_c55_params_awb_gains {
        struct mali_c55_params_block_header header;
        __u16 gain00;
        __u16 gain01;
        __u16 gain10;
        __u16 gain11;
};
```

And the ISP makes it easy

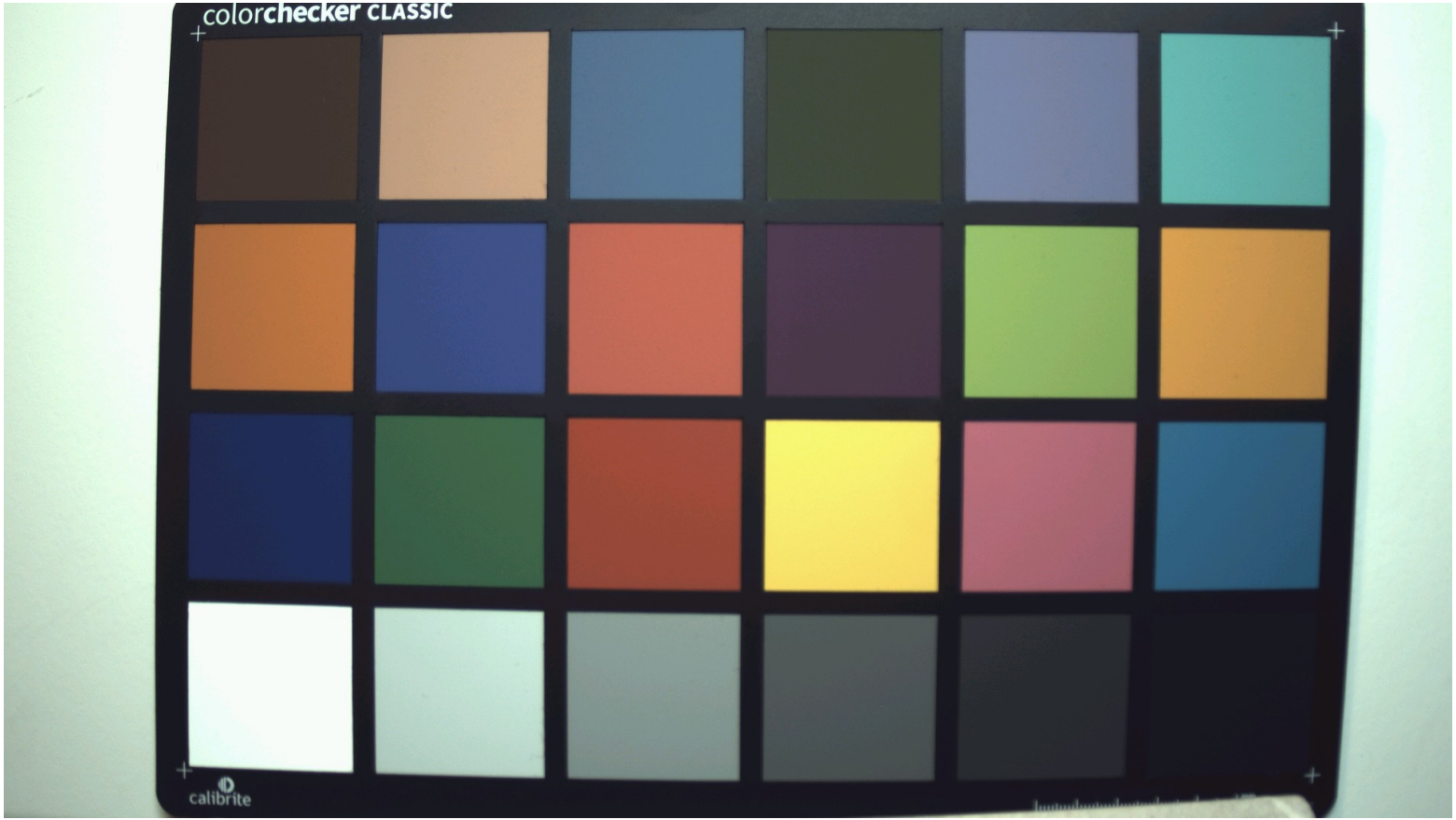
```
struct mali_c55_awb_average_ratios {
        __u16 avg_rg_gr;
        __u16 avg_bg_br;
        __u32 num_pixels;
}


/*
 *      gain00 = R
 *      gain01 = Gr
 *      gain10 = Gb
 *      gain11 = B
 */

struct mali_c55_params_awb_gains {
        struct mali_c55_params_block_header header;
        __u16 gain00;
        __u16 gain01;
        __u16 gain10;
        __u16 gain11;
};
```

```
void MaliC55Awb::process(...)
{

    ...

    params->gain00 = 1 / stats->avg_rg_gr;
    params->gain01 = 1;
    params->gain10 = 1;
    params->gain11 = 1 / stats->avg_bg_gr;

    ...

}
```

We go from a greenish image



To a more balanced one

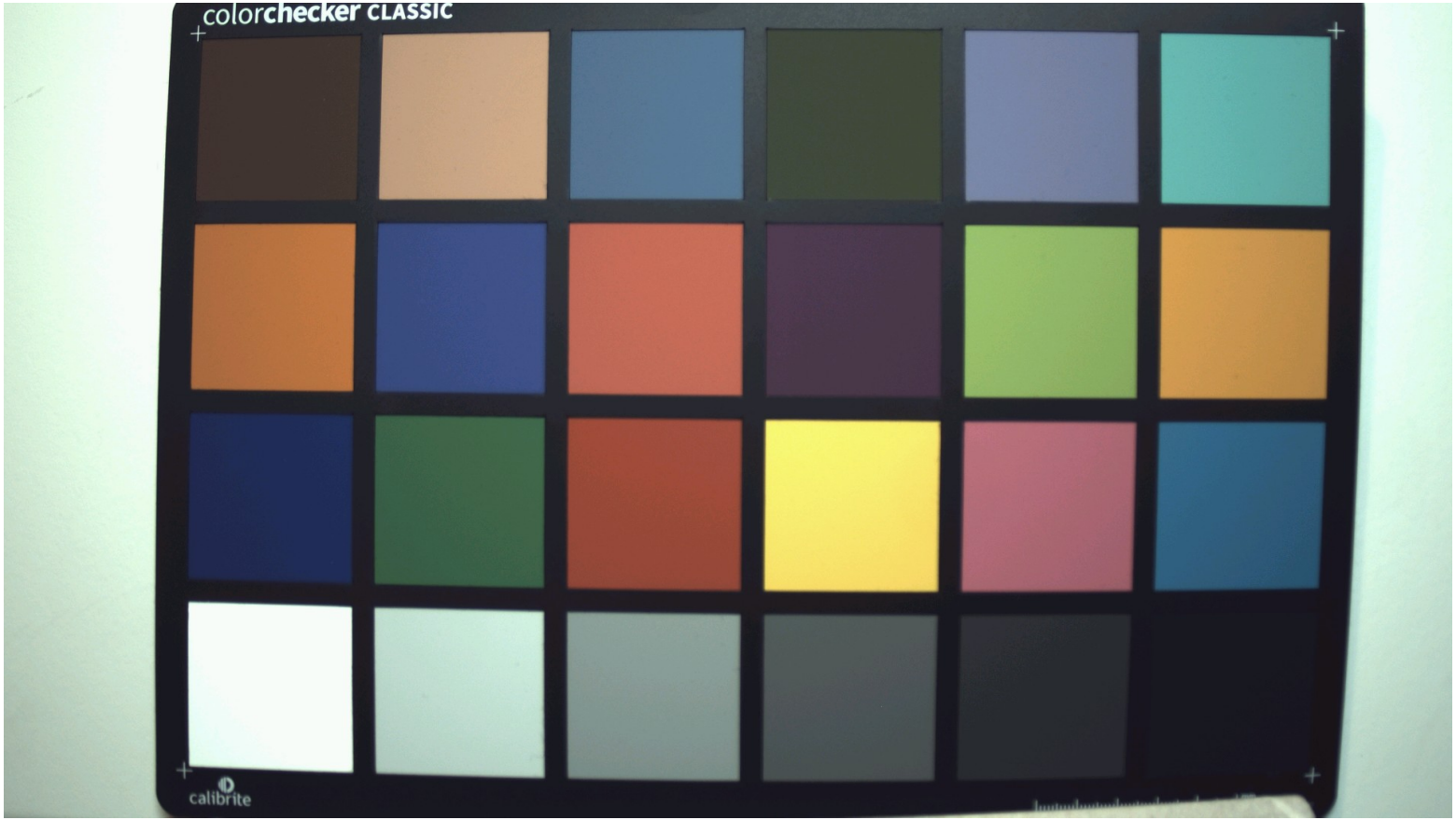


Some values we just parse from tuning files

```yaml
# SPDX-License-Identifier: CC0-1.0
%YAML 1.1
---
version: 1
algorithms:
  - Agc:
  - Awb:
  - BlackLevelCorrection:
      offset00:  51200
      offset01:  51200
      offset10:  51200
      offset11:  51200
...
```

```cpp
int BlackLevelCorrection::init([[maybe_unused]] IPAContext &context,
                               const YamlObject &tuningData)
{
        offset00 = tuningData["offset00"].get<uint32_t>(256);
        offset01 = tuningData["offset01"].get<uint32_t>(256);
        offset10 = tuningData["offset10"].get<uint32_t>(256);
        offset11 = tuningData["offset11"].get<uint32_t>(256);

        if (offset00 > kMaxOffset || offset01 > kMaxOffset ||
            offset10 > kMaxOffset || offset11 > kMaxOffset) {
                LOG(MaliC55Blc, Error) << "Invalid black level offsets";
                return -EINVAL;
        }

        tuningParameters_ = true;

        LOG(MaliC55Blc, Debug)
                << "Black levels: 00 " << offset00 << ", 01 " << offset01
                << ", 10 " << offset10 << ", 11 " << offset11;

        return 0;
}
```

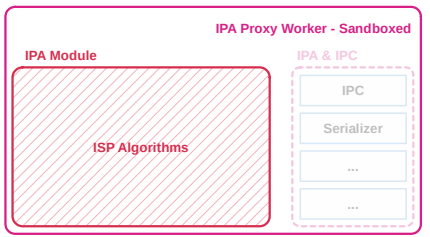
We go from noisy “black”



To well calibrated black

We need to tackle vignetting

```yaml
# SPDX-License-Identifier: CC0-1.0
%YAML 1.1
---
version: 1
algorithms:
  - Agc:
  - Awb:
  - BlackLevelCorrection:
      R:  256
      Gr: 256
      Gb: 256
      B:  256
  - Lsc:
      meshScale: 4
      sets:
        - ct: 2500
          r: [
              21, 20, 19, 17, 15, 14, 12, 11,  9,  9,  9,  9,  8,  8,  9,  9,
               9,  9,  9,  9,  9,  9, 13, 13, 10, 10, 13, 16, 17, 18, 21, 22,
              ...
              ]
          g: [
              19, 18, 17, 15, 13, 12, 10,  9,  8,  8,  7,  7,  7,  7,  7,  7,
               7,  7,  7,  7,  7,  7, 12, 12,  9,  9, 11, 15, 15, 16, 19, 20,
              ...
              ]
...
```



src/ipa/mali-c55/data/imx415.yaml

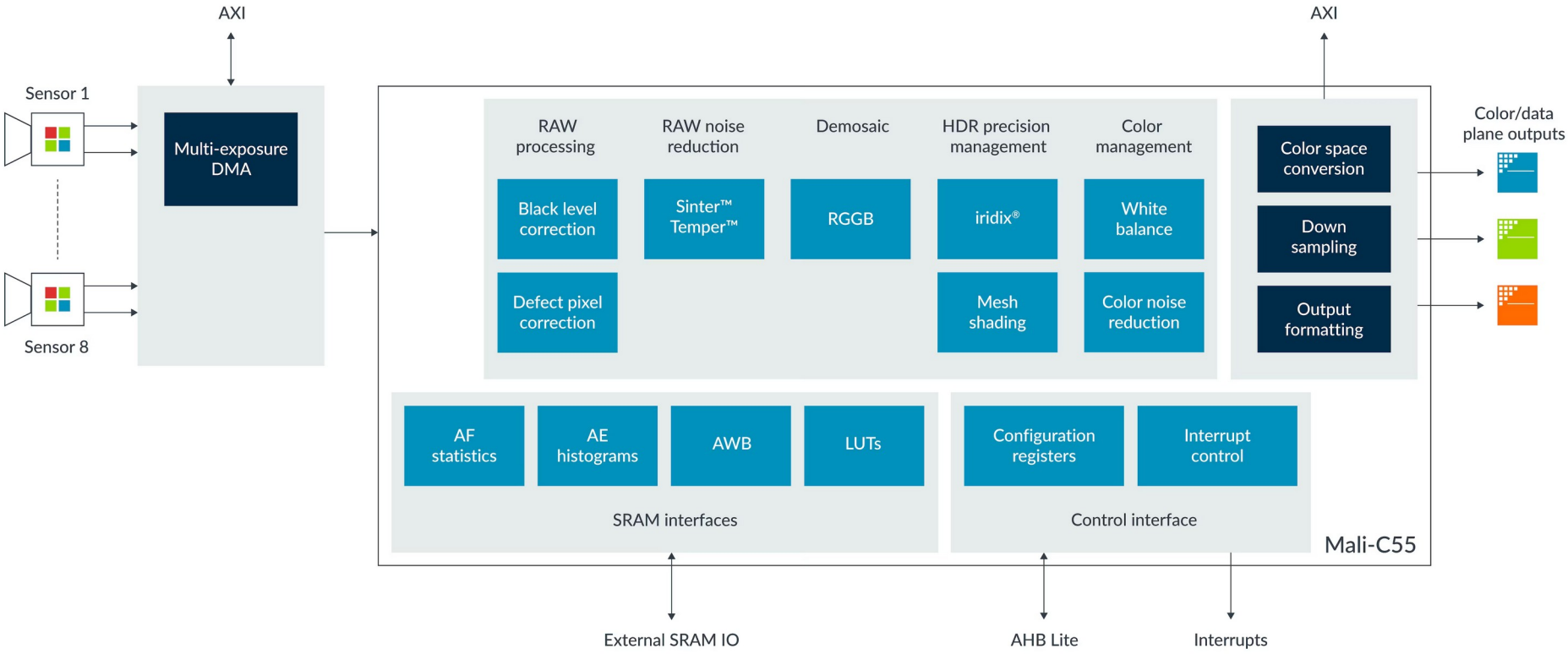
Tuning File

Corrected black levels

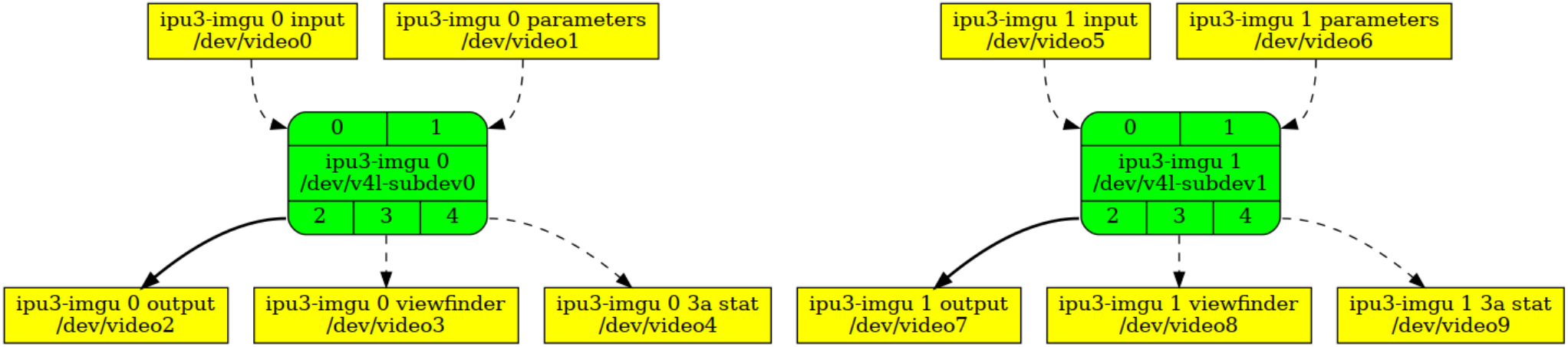
Brightened peripheries

- Introduction
- The hardware
- The kernel
- libcamera
- Future challenges
- How can you use what we've done?
- Any questions?

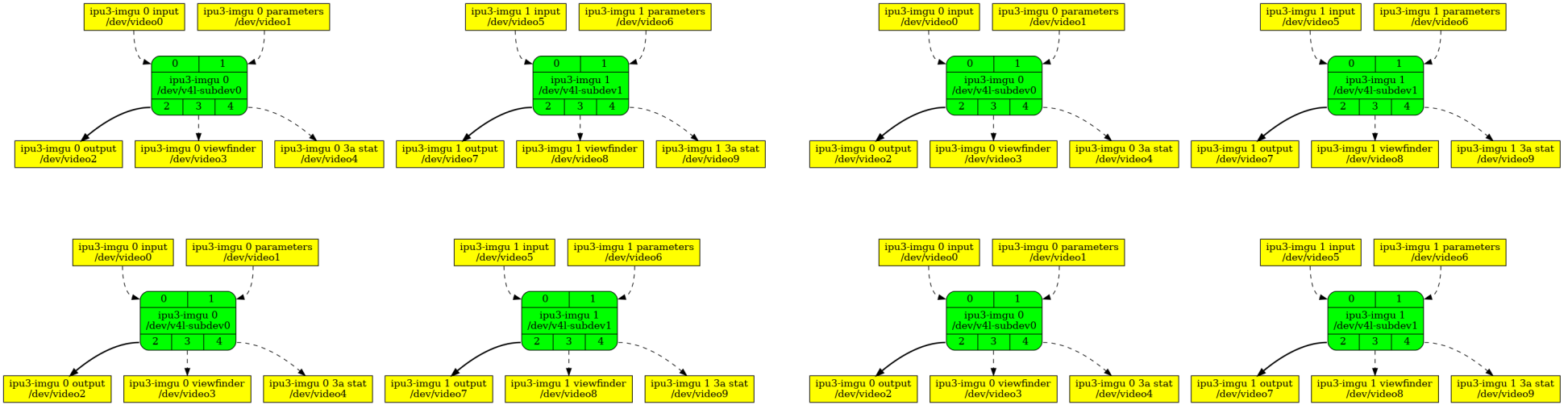
The ISP has other image quality blocks



Multi-context in the kernel today...



...but does it scale?

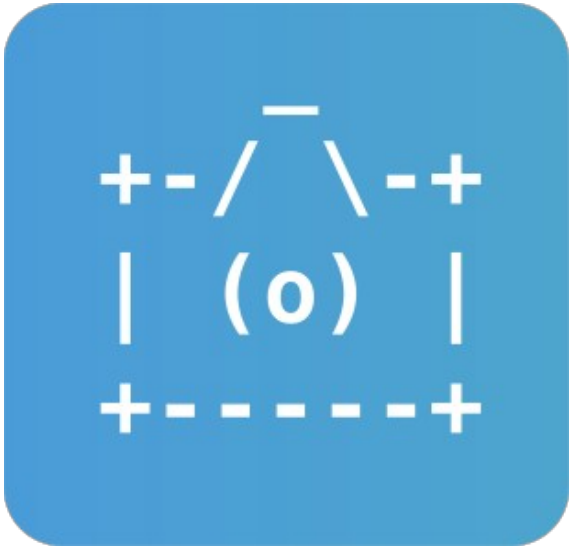


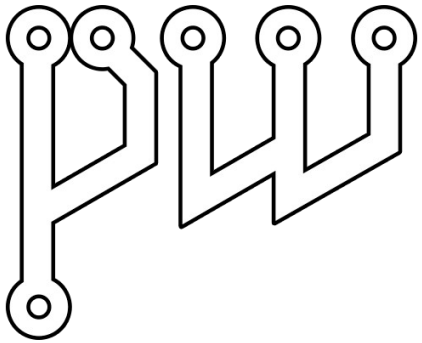
- Introduction
- The hardware
- The kernel
- libcamera
- Future challenges
- How can you use what we've done?
- Any questions?

Well, you can use the V4L2 APIs of course...

```
# Set formats on the TPG and ISP
media-ctl -V "'mali-c55 tpg':0[fmt:SRGGB16_1X16/1920x1080]"
media-ctl -V "'mali-c55 isp':0[fmt:SRGGB16_1X16/1920x1080]"
media-ctl -V "'mali-c55 isp':1[fmt:SRGGB16_1X16/1920x1080]"

# Set routing on the FR resizer
media-ctl -R "'mali-c55 resizer fr'[0/0->1/0[1],2/0->1/0[0]]"

# Set format on the resizer, must be done AFTER the routing.
media-ctl -V "'mali-c55 resizer fr':1[fmt:RGB121212_1X36/1920x1080]"

yavta -f RGB565 -s 1920x1080 -c10 /dev/video0
```

But you should just use libcamera





Any questions?

LinaroConnect
MADRID 2024 | MAY 12-17 2024

Thank you