Rust in V4L2

Daniel Almeida
Consultant Software Engineer, Collabora
daniel.almeida@collabora.com

Who am I?

COLLABORA

First and foremost, I am working on software support for video codec accelerators

Things I have been working on

- Codec uAPIs for V4L2 stateless codecs
- A virtual V4L2 stateless codec driver
- GStreamer support for V4L2 stateless codecs
- Video codec virtualization on Chromebooks
- cros-codecs/cros-libva

COLLABORA

Now lets talk about Rust in V4L2 proper

What has been done so far?

What we have so far

- Some POD types (yay for syn/quote support!)
- A *very* thin videobuf2 abstraction (you can create a queue)
- Abstractions for some VIDIOC_* ioctls
- The necessary code to get the driver to probe
- A module that prints to the terminal when processing some of the VIDIOC_* ioctls

COLLABORA

Why is the V4L2 subsystem a good candidate for Rust?

Because there are some low-risk areas that we can tackle first

Easy areas to tackle

- Codec libraries, especially the AV1 library
- JPEG parser
- Codec-specific logic in codec drivers (e.g. writing codec metadata to MMIO registers)
- Virtual drivers (we love these, they help in testing)

Why are these easy?

- Self contained
- Do not interact with HW directly
- Easy way for V4L2 maintainers to gauge whether Rust will work for them
- If it does not work, it is not much work to rewrite in C

COLLABORA

I discussed that approach during the Media Summit 2023

COLLABORA

Roadblocks and feedback

Roadblocks and feedback

- V4L2 can’t keep up with the workload as is

Roadblocks and feedback

- V4L2 can’t keep up with the workload as is
- Not enough reviews and maintainers

Roadblocks and feedback

- V4L2 can’t keep up with the workload as is
- Not enough reviews and maintainers
- There are long-standing issues with some C frameworks (e.g. media controller lifetime issues)

Roadblocks and feedback

- V4L2 can’t keep up with the workload as is
- Not enough reviews and maintainers
- There are long-standing issues with some C frameworks (e.g. media controller lifetime issues)
- Huge fear of breaking existing C code

Roadblocks and feedback

- V4L2 can’t keep up with the workload as is
- Not enough reviews and maintainers
- There are long-standing issues with some C frameworks (e.g. media controller lifetime issues)
- Huge fear of breaking existing C code
- **There must be more contributors working on this**

At the very minimum, we need two people working on this

COLLABORA

Also, Collabora is a consultancy, which means…

Business perspective

- Chicken and egg problem
  - Hard to find clients interested in sponsoring infrastructure work
  - Rust looks risky and most companies do not want to take such a gamble
  - Hard (impossible) to provide deadlines

In summary: the value proposition is a bit unclear

COLLABORA

Open questions:

Open questions

- What happens if the C API is changed and it breaks the Rust bindings? Can we detect automatically?
- Will the above delay the C work, as the Rust code will not compile anymore?
- How can we expose a C FFI so that C drivers can use Rust code in practice?
- How should maintainability work?

Quick summary

- Maintainers are not against Rust, but a lot has to happen
- This work needs more people involved
- Hopefully, the subsystem's maintainership issues will be solved by the time we start upstreaming this
- Until then, Collabora can maintain a tree so that people can experiment with Rust in V4L2

COLLABORA

But

Collabora needs sponsors to make this happen

If your company is interested, do let me know

Thank you!