

How to Build an Open Source Embedded Video Playback System

Michael Tretter

m.tretter@pengutronix.de

FOSDEM 2017

Feb 4, 2017



Embedded Video Playback System



By Mw12310 - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=13431076>



Embedded Video Playback System



By Catecardvd - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=32985304>



Embedded Video Playback System



By LG 전자 - 세계 명화가 LG 스마트 TV 속으로 , CC BY 2.0, <https://commons.wikimedia.org/w/index.php?curid=17450806>

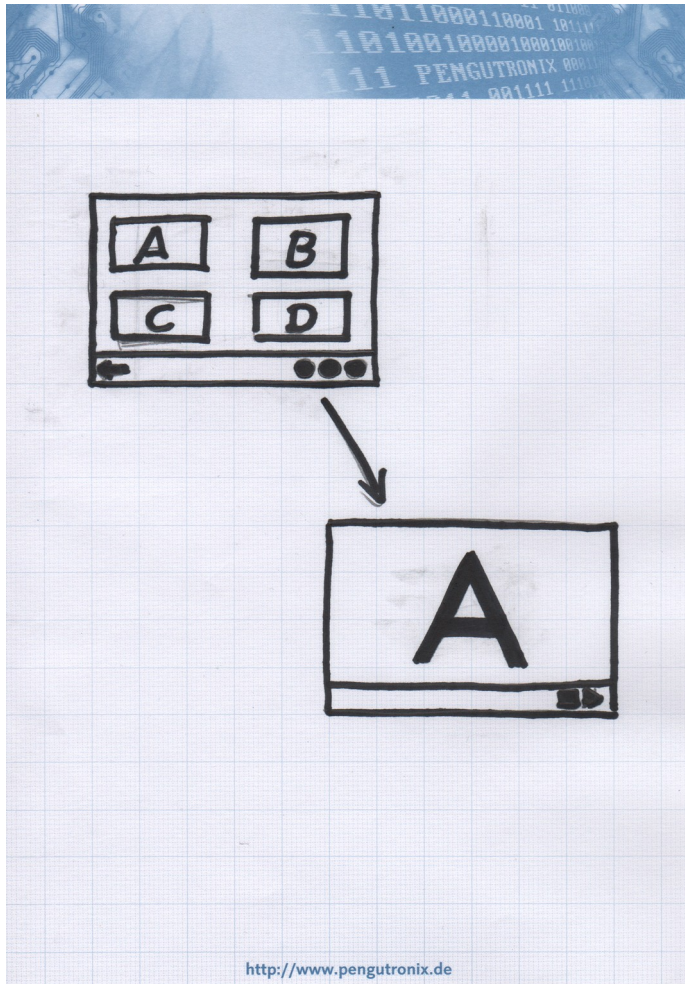


Agenda

- Features
- Status Quo
- Open source
- Future Work

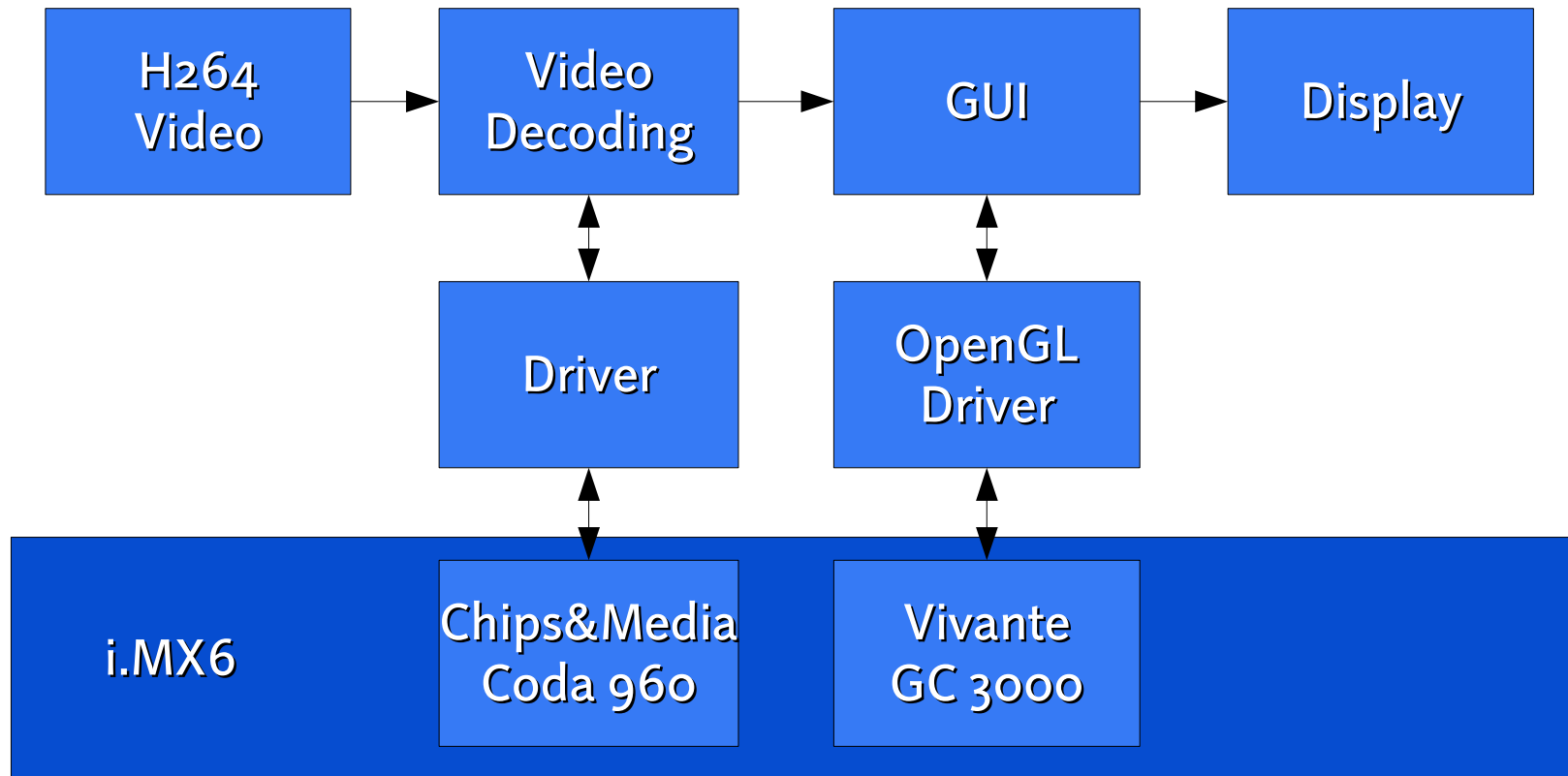


Features



- Interactive GUI
- Preview of multiple videos
- Fullscreen playback of one video
- OpenGL acceleration for GUI

System Architecture



Status Quo

- Vendor supplied board support package
- Linux kernel and userspace
- Patches for vendor selected Linux version (e.g., 3.14.52 / 4.1.15)
- Binary blob drivers for GPU and video decoding
- Obstacles for debugging and maintenance

What can we achieve using open source software from upstream?



User Interface: QML



- Declarative user interface specification and programming language
- Graphics acceleration via OpenGL
- Demo application
150 lines of QML code
- Interface to video decoding
200 lines of C++ code



Compositing: etnaviv

- Reverse-engineered driver for Vivante GPUs
- Available upstream in Mesa (since 17.0) and Linux (since 4.5)
- etnaviv implements OpenGL → Usable from Qt 5
- Video frames can be composited in hardware



Video Pipeline: Gstreamer



- No solution in upstream Gstreamer, yet
- Custom sink “gst-video-item”
- Zero-copy from Gstreamer to QML
- Pipeline auto-plugging with playbin

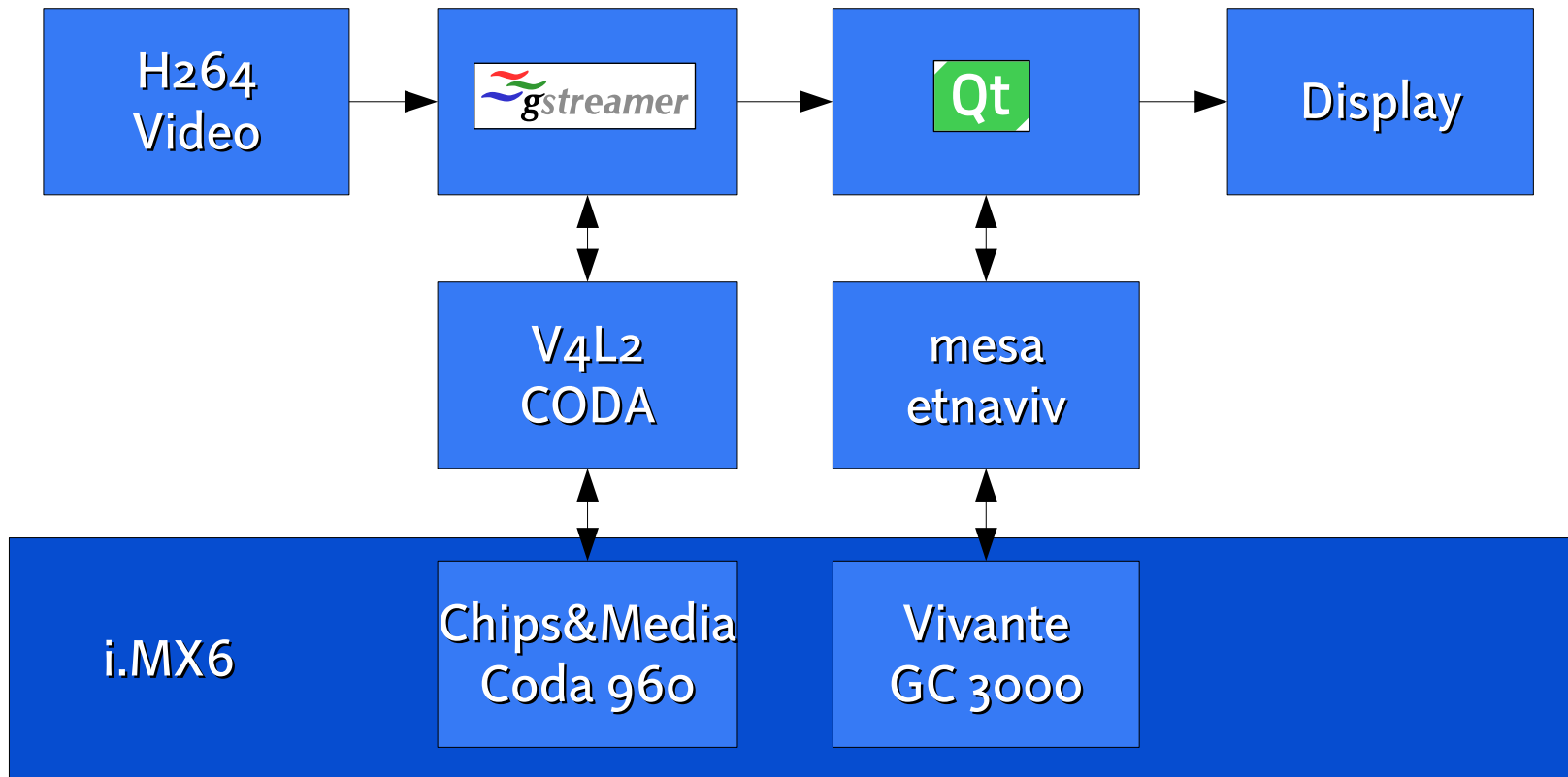


Decoding: CODA Driver

- Linux driver available in mainline (VIDEO_CODA)
- V4L2 mem2mem device → /dev/videoX
- Gstreamer v4l2videoXdec element
- i.MX6 hardware customizations → drivers about to be mainlined
- Unfortunately still closed source firmware



System Architecture



Future Work

- Upstream solution for GStreamer to etnaviv interface (gst-video-item)
- Other compositors for UI, e.g, Wayland compositor
- Adaptive streaming for dynamic adjustment of video quality



Conclusion

- Binary blob drivers → issues with maintenance and debugging
- User interface → etnaviv and QML
- Video decoding → Gstreamer and V4L2
- Future work → Wayland, adaptive streaming

Embedded video playback does not require blob drivers



Thank You!

- GStreamer - <https://gstreamer.freedesktop.org>
- mesa - <http://www.mesa3d.org>
- QML - <https://www.qt.io/qt-quick>

