Software and hardware images decoding on the Raspberry Pi

X.Org Developers Conference 2020

Maciej Pijanowski
• whoami
• Company profile
• Raspberry Pi Zero hardware
• BCM2835 SoC features
• JPEG encoding/decoding
• OpenMAX Overview
• Using Broadcom OpenMAX IL components
• MMAL API
• Performance results
• Summary
Maciej Pijanowski  
*Embedded Firmware Team Leader*

- @macpijan  
- maciej.pijanowski@3mdeb.com  
- linkedin.com/in/maciej-pijanowski-9868ab120

- over 4 years in 3mdeb  
- interested in:  
  - Embedded Linux  
  - build systems (e.g. Yocto)  
  - system security
3mdeb is a firmware and embedded systems development company founded by Piotr Król and headquartered in Gdańsk, Poland. We combine the collaborative creativity of the open-source community with the reliable strength of an enterprise-grade solutions provider.
Our team is made up of engineers with vast experience working with UEFI/BIOS, coreboot, Linux, Yocto, and more. We create IoT and firmware solutions supporting security and integrity standards; roots of trust, boot integrity, TPM, DRTM, and much more.
Graphics experience

- GPU/VPU acceleration
  - Vivante GPU (i.MX6)
  - Mali GPU (Allwinner, Amlogic, Xilinx, ...)
- Mostly integration work (but not limited to)
- Wayland, Weston, Qt, WebKit, ...

X.Org Developers Conference 2020
CC BY | Maciej Pijanowski
• Broadcom BCM2835 SoC
• $5 for base one
• $10 for the one with WiFi/Bluetooth
• Will remain in production at least until January 2026
- SoC present in:
  - RaspberryPi 1 Model A+/B+
  - RaspberryPi Zero/Zero W
- ARM1176JZF-S
  - 1-core, 1GHz, low power application processor
  - ARMv6 architecture (no NEON instruction set)
- VideoCore IV
  - 2-core multimedia co-processor
  - Hardware video decoding (H.264)
  - Hardware JPEG decoding (baseline JPEG only)
  - more...

https://github.com/hermanhermitage/videocoreiv/wiki/VideoCore-IV---BCM2835-Overview
• Color space transformation (from RGB to YCbCr)
- Human eye is more sensitive to brightness than to colors
- Downsampling (chroma subsampling)
  - effectively reduce resolution in chroma (U, V) channels
  - reduce bits needed to encode pixels

Perform DCT (Discrete Cosine Transform) on blocks
  
  move from spatial to a frequency representation
Human eye is not sensitive for high-frequency brightness changes

Quantization
  - divide DCT coefficients by **some** constants
  - constants are taken from the quantization matrix (matrix form depends on the desired quality level)
  - round the achieved coefficient to integers
  - higher frequencies are eliminated or reduced
  - much fewer bytes are needed to encode pixel data

```
16 11 10 16 24 40 51 61
12 12 14 19 26 58 60 55
14 13 16 24 40 57 69 56
14 17 22 29 51 87 80 62
18 22 37 56 68 109 103 77
24 35 55 64 81 104 113 92
49 64 78 87 103 121 120 101
72 92 95 98 112 100 103 99
```

[https://en.wikipedia.org/wiki/Quantization_(image_processing)#Quantization_matrices](https://en.wikipedia.org/wiki/Quantization_(image_processing)#Quantization_matrices)
• Perform operations inversely than in the encoding case
  • Baseline
    ○ single scan
    ○ target image quality after the first scan
  • Progressive
    ○ several scans
    ○ image quality increases with each scan
- [link](https://libjpeg-turbo.org/)
- 2-6x performance improvement in comparison to the original libjpeg
  - provided that the HW supports SIMD
- Uses SIMD instructions (Single Instruction Multiple Data)
  - operations on data vectors
- Requires hardware support in given CPU
  - x86/x86-64 (MMX, SSE, AVX)
  - ARM (NEON)
  - PowerPC (AltiVec)
- NEON
  - can be available in ARMv7 and ARMv8 architectures only
- Most of the HW used nowadays is supported
  - so we care less about the image decoding acceleration
omxplayer is the well-known hardware accelerated video player for RPi

The equivalent for static images could be the omxiv:
  - [https://github.com/HaarigerHarald/omxiv](https://github.com/HaarigerHarald/omxiv)
  - makes use of the hardware baseline JPEG decoding
  - software decoding support for other formats (progressive JPEG, PNG, BMP, TIFF)
  - not widely used
  - uses the OpenMAX IL library through the ilclient library
  - not actively maintained
  - not packaged for Raspbian

We have experimented with the code a little:
  - converted app into a basic library
  - added zoom/crop feature
  - [https://github.com/3mdeb/omxiv/commits/lib](https://github.com/3mdeb/omxiv/commits/lib)
• Practically deprecated for a few years already
• The RPi supports only the IL layer

https://www.khronos.org/openmaxil
Using Broadcom OpenMAX IL components

- Documentation in the `raspberrypi/firmware` repository
Using Broadcom OpenMAX IL components

- **image_decode**
  - takes the encoded image on the input port
  - decodes it into raw pixels on the output port

- **resize**
  - resizes raw pixel image
  - the input image can be additionally cropped prior resizing

- **video_render**
  - can render both single frame and video
Multi-Media Abstraction Layer
An alternative way (to OpenMAX) to access the hardware blocks of the VC4
C library
  ◦ designed by Broadcom for use with the VideoCore IV GPU
  ◦ the aim was to replace the OpenMAX IL
  ◦ specific to the Broadcom SoC (RPi devices, really)
MMAL API documentation
Similar high-level design to OpenMAX IL
  ◦ uses components
  ◦ components have ports
It is supposed to be easier to use
• Is suggested to use for new projects instead of the OpenMAX IL
• omxplayer - deprecated project already
  ○ resources moved to VLC development
  ○ MMAL-based codec patch being developed: https://kutt.it/rs8mQ0
  ○ VLC - not quite suitable for embedded systems
• No image viewer using MMAL for hardware-accelerated image decoding
• Limited MMAL examples
  ○ jpeg example on userland fork: https://tkutt.it/bGkn91
In our case time to display was important
  - total time from typing command to having an image on the screen
  - the decoding time (just in memory) would be slightly lower

omxiv was used as an image viewer

```bash
$ ./omxiv.bin image baseline.jpg
$ ./omxiv.bin --soft image baseline.jpg
$ ./omxiv.bin image progressive.jpg
```
Performance measurements

X.Org Developers Conference 2020
CC BY | Maciej Pijanowski
Summary

- Hardware decoding of baseline JPEG decoding performance is almost identical for both platforms - this is not surprising since the same hardware accelerating block is used.
- It takes longer to decode progressive image rather than baseline.
- The gain from using hardware acceleration is more visible for images with higher resolution.
- The difference is less on the RPi3, but the hardware accelerator still outperforms the CPU in JPEG decoding, even if the libjpeg-turbo was used.
libjpeg-turbo is here for around 10 years already and we sometimes assume that software JPEG decoding is enough.

There is still some older hardware which cannot benefit from it. In such cases, usage of the hardware JPEG decoding (when available) is beneficial.

The hardware blocks may limit the supported JPEG formats - in such case software decoding fallback is needed.

OpenMAX IL is practically deprecated.
  - but projects using it (like omxplayer) will likely be used for years.

The MMAL should be used, but
  - limited usage examples
  - no lightweight image/video player
We hope that X.Org Developers Conference will finally take place in our beautiful city one day.
We are open to cooperate and discuss

- contact@3mdeb.com
- facebook.com/3mdeb
- @3mdeb_com
- linkedin.com/company/3mdeb
- https://3mdeb.com
- Book a call
- Sign up to the newsletter
• JPEG/JFIF format description
• Programming AudioVideo on the Raspberry Pi GPU
• Khronos OpenMax Overview
• VideoCore IV Architecture Reference Guide
• BCM2835 ARM Peripherals Documentation
• ARM1176JZF-S Technique Reference Manual
• Decode JPEG stream using MMAL or OpenMAX
• MMAL-based image viewer
• Video acceleration on the Raspberry Pi 4
• 3mdeb website
Q&A