Supporting a new ARM platform: the Allwinner example

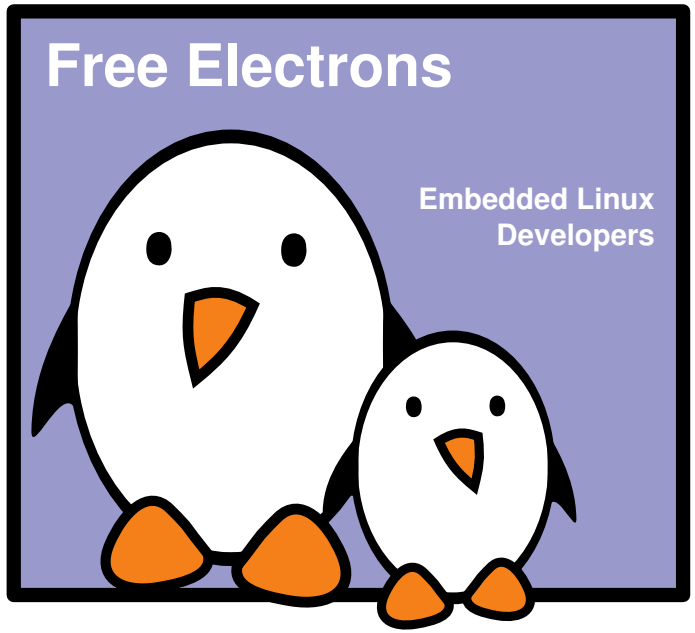
Maxime Ripard
Free Electrons
maxime@free-electrons.com

Maxime Ripard

- Embedded Linux engineer and trainer at Free Electrons
  - Embedded Linux **development**: kernel and driver development, system integration, boot time and power consumption optimization, consulting, etc.
  - Embedded Linux **training**, Linux driver development training and Android system development training, with materials freely available under a Creative Commons license.
  - `http://free-electrons.com`
- Contributions
  - **Kernel support for the sunXi SoCs** from Allwinner
  - Contributor to few open-source projects, **Buildroot**, an open-source, simple and fast embedded Linux build system, **Barebox**, a modern bootloader.
- Living in **Toulouse**, south west of France

Introduction

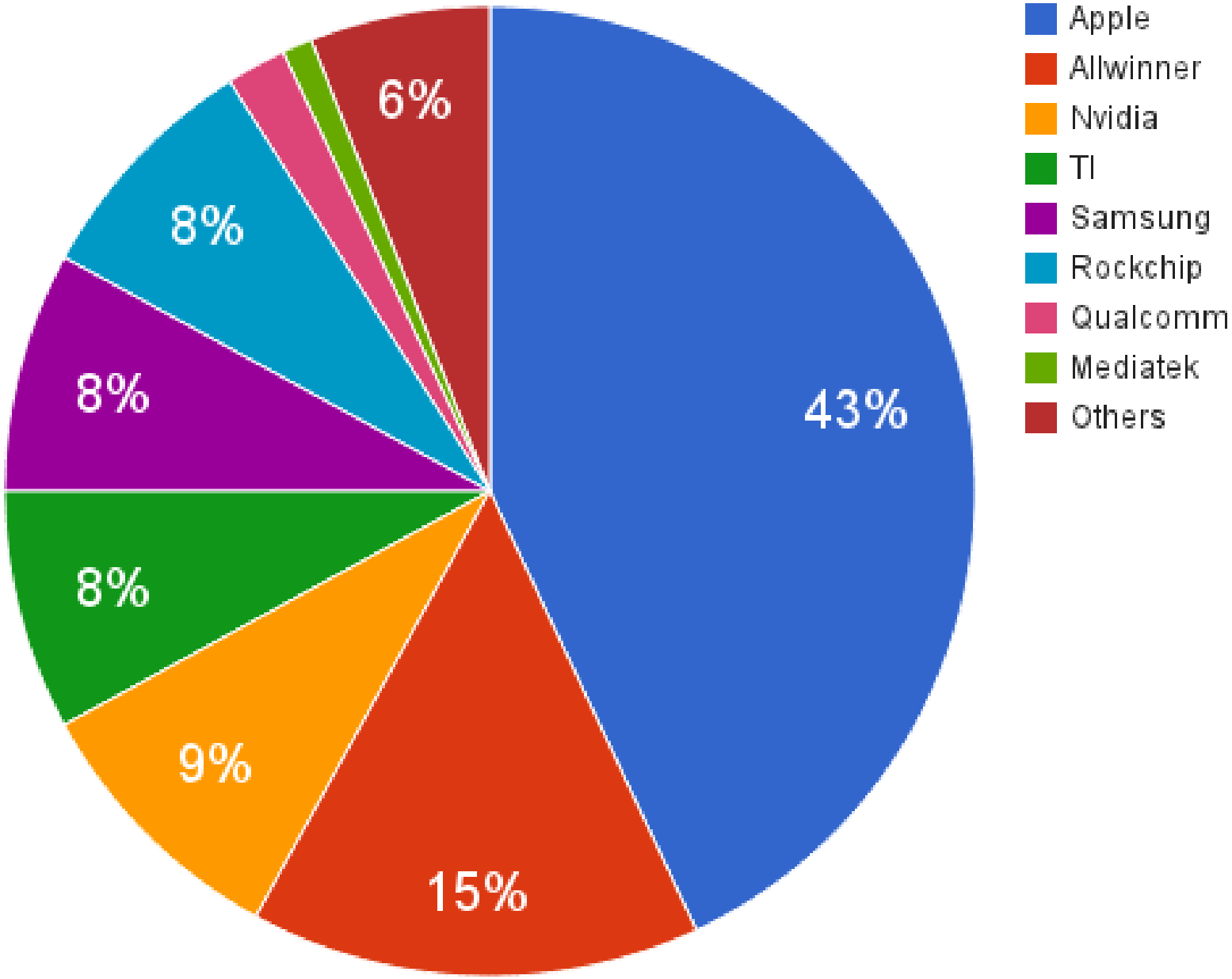
Maxime Ripard
Free Electrons



Allwinner

- Chinese fabless SoC vendor
- Founded in 2007, in Zhuhai
- Always did ARM SoC, starting with ARM9 CPUs
- Mostly doing multimedia SoCs
- 550 employees

Tablet Market Share in 2013

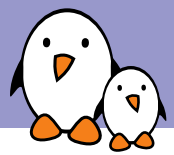


SoC Line-up

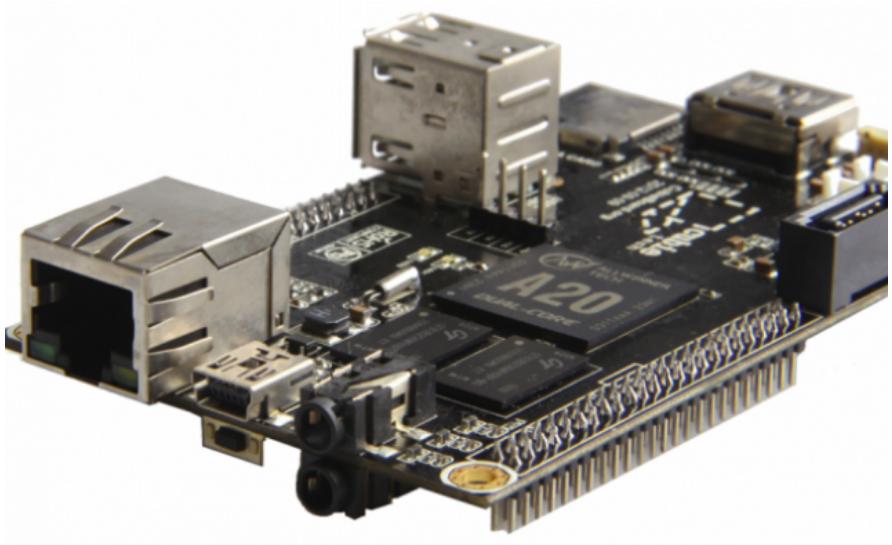
- F-Series
  - ARM9-based SoCs
  - Mostly were running a custom OS called MeliOS, except for the late SoCs in that family
- A10 / A10s / A12 / A13 (sun4i/sun5i), A20 (sun7i)
  - Cortex-A8 based for the sun4i/sun5i
  - Dual Cortex-A7 for the A20
- A31 / A31s (sun6i)
  - Quad Cortex-A7
  - The GPU is Imagination's PowerVR
  - Brand new design, a lot of IPs changed
- A23/A33 (sun8iw3/sun8iw5)
  - Dual/Quad Cortex-A7
  - An hybrid between the A20 and the A31
- A80 (sun9i)
  - big.LITTLE with 4 Cortex A15 and 4 Cortex A7

Development / Hacker-friendly boards

- Olimex Olinuxino
  - Open Source Hardware
  - Access to most of the SoC pins through 0.1" headers
  - Very supportive towards the community
- Cubieboards
  - Tiny, and cheap
  - A very strong community has been built around it
- A lot of other boards: pcDuino, Marsboard, Banana Pi, etc.

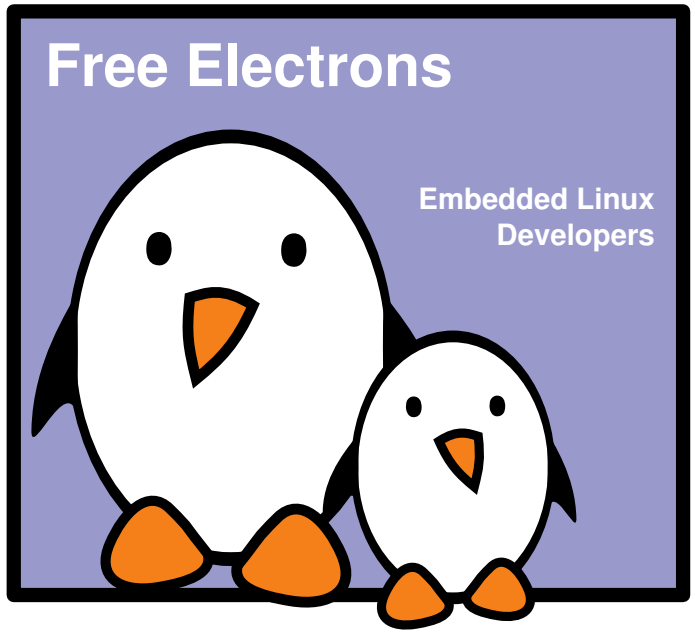


Boards



Where we were

Maxime Ripard
Free Electrons



Allwinner's Kernel

- Allwinner Kernel are mostly based on two kernels:
  - Linux 3.0 for Cortex-A8 based SoCs
  - Linux 3.3/3.4 for Cortex-A7 based SoCs
- Just like most vendor code, their focus is primarily on making that particular kernel version working on one SoC
- There's one different kernel tree for each SoC, even if they are really close
- Apart from that, given the version they're based on, they're in pretty decent shape

Allwinner Kernel Issues

- However, it suffers some major issues
  - By today standards, the code is pretty outdated
    - No Common Clock Framework
    - No pinctrl / GPIOlib
    - No dmaengine
  - There's no consolidation at all between the SoCs code
  - You're pretty much stuck with that kernel version
- ... And there's FEX

FEX

- It is a binary file, compiled from a human-readable text file
- Passed along the boot chain to the various components (bootloaders, kernel)
- Used to get the hardware configuration of the board they're running on, in a generic way
- In the FEX script, you get
  - Which device are enabled
  - The memory timings
  - The clocks frequency
  - The pin muxing
  - The regulators voltage
  - Plus, various hardware and configuration information sq

How it looks like

```
[twi1_para]
twi_used = 1
twi_scl = port:PH16<2><default><default><default>
twi_sda = port:PH17<2><default><default><default>

[mmc0_para]
sdc_used = 1
sdc_detmode = 2
sdc_buswidth = 4
sdc_clk = port:PF02<2><1><2><default>
sdc_cmd = port:PF03<2><1><2><default>
sdc_d0 = port:PF01<2><1><2><default>
sdc_d1 = port:PF00<2><1><2><default>
sdc_d2 = port:PF05<2><1><2><default>
sdc_d3 = port:PF04<2><1><2><default>
sdc_det = port:PA08<6><1><2><default>
```

How it is used

- It is then compiled by `fexc` to a binary file
- Then, Linux will have hardcoded devices, and each matching device will
  - Be probed
  - Lookup the FEX to get if it is allowed to probe
  - Then retrieve whatever parameter it is interested in
- Even though it is a lot like the device tree, the logic is completely backward

Pros / Cons

- Pros
  - Most of the DT Pros
  - Allows to have a completely generic kernel image
  - Separates the hardware description from the code
  - Allows a very short time-to-boot, even if you're not really a kernel developer
- Cons
  - Completely non-standard
  - Only works for a single platform
  - Not generic enough

FEX moment of glory

so.

coming back to what you said earlier: I'm formulating what to say to allwinner [and need to pre-send something by monday so that they can consider it before the meeting]. so far, it consists of:

** device-tree is what the linux kernel community has come up with, it is equivalent to FEX.*

** the linux kernel community would like to apologise for not consulting with you (allwinner) on the decision to only accept device tree*

(Luke Kenneth Casson Leighton, 6/7/2013)

FEX moment of glory

Thanks to their script.fex system when i said they only need to develop one kernel and one u-boot i really wasn't kidding around: they really have got to the point which everyone else dreams of with device-tree

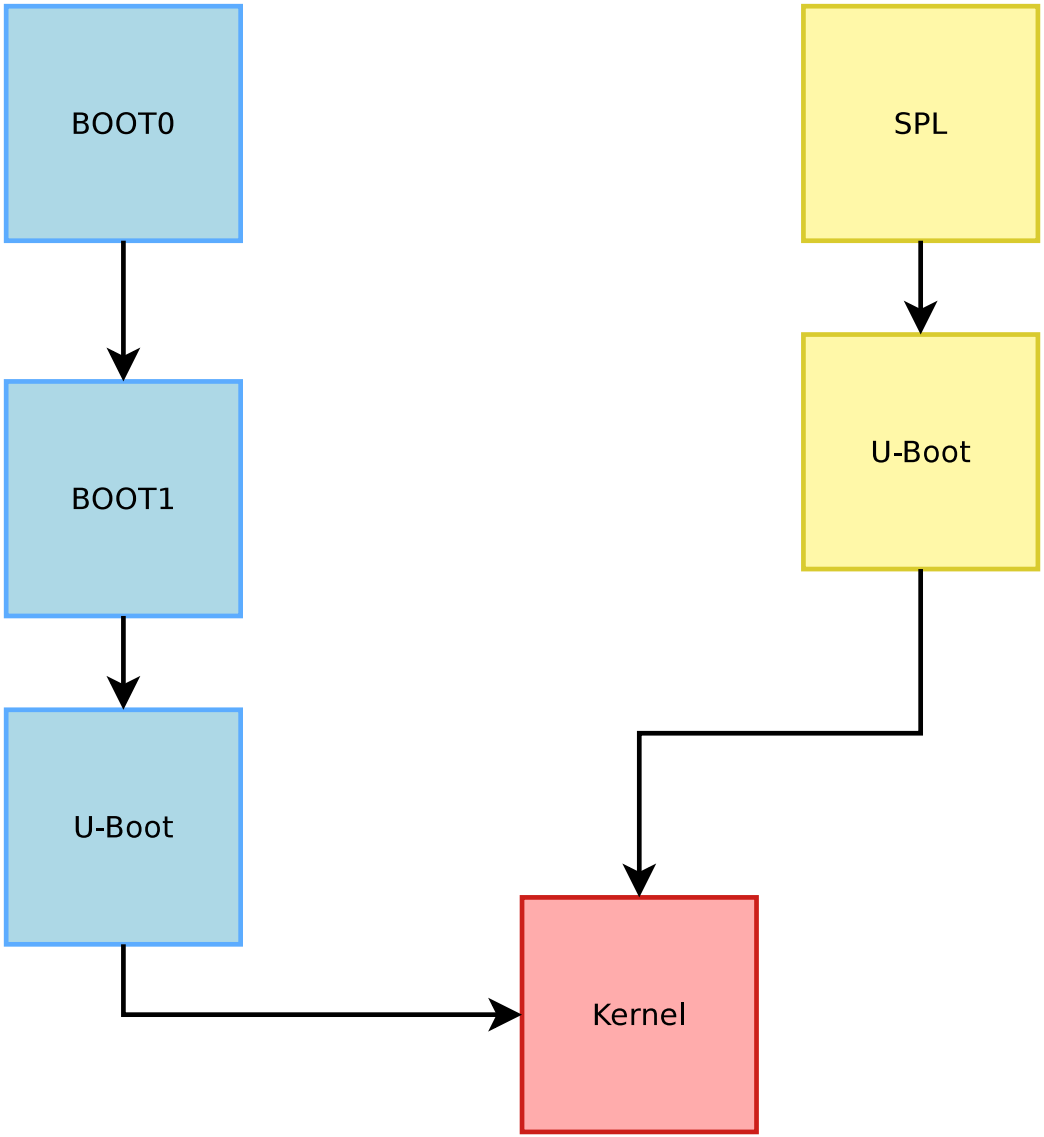
*(*Luke Kenneth Casson Leighton, 6/7/2013*)*

FEX moment of glory

I have a Cubieboard and I have a pca9532 on my desk. Now I want to attach this pca9532 to the Cubieboard so I wire them together on I2C. How is the Allwinner kernel going to load the driver for the pca9532? The mainline pca9532 driver does not understand fex so it can't read the necessary initialization data. (Jon Smirl, 6/6/2013)

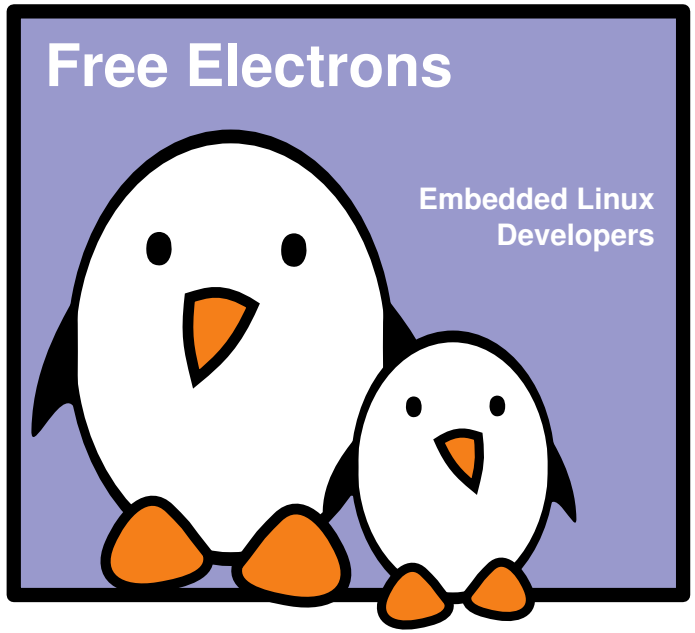
You're immediately outside of the target market for which allwinner designed and deployed script.fex.
(Luke Kenneth Casson Leighton, 6/6/2013)

Boot Process



Community

Maxime Ripard
Free Electrons



linux-sunxi

- A community started around rhombus-tech and their EOMA-68 standard in 2010
- It forked off in 2011, and created `http://linux-sunxi.org`
- It's very active these days, with around 600 subscribers on the ML, and around 150 persons on IRC (idlers included)
- Around 10 very active developers, working on different areas of the SoC support: bootloader, kernel, VPU and GPU reverse engineering, distribution support, etc.
- Libre Software Extremists

Achievements

- Documentation of the hardware
- Documentation of the BSP
- Implementation of needed tools
- Maintainance of linux and u-boot
- Bring up of Distributions
- Reverse engineering of the SoC (NAND, VPU, etc.)

linux-sunxi kernel

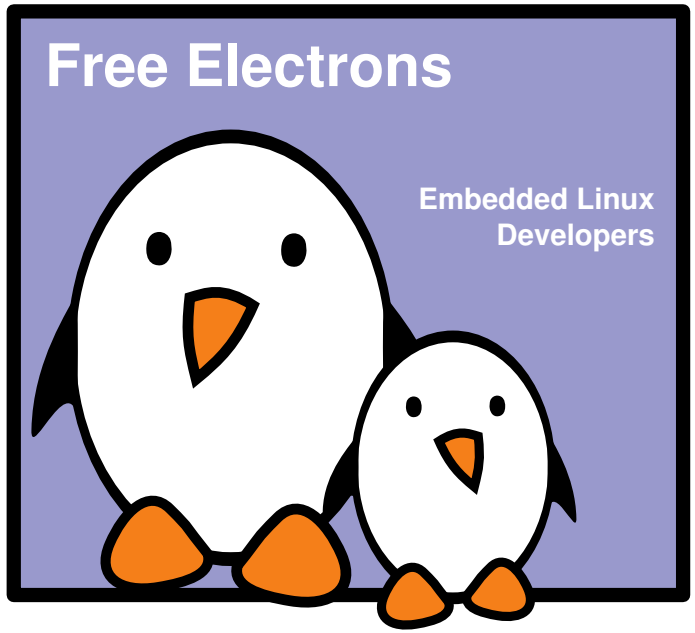
- Maintains a fork of Allwinner's kernel
- Does all the maintenance work
  - Support most of the SoCs made by Allwinner in one single kernel tree
  - Factorizes and cleans up the code
  - Fix bugs
- Based on a 3.4 (stable) kernel

Community Kernel: Pros and Cons

- Pros
  - Maintain the code that Allwinner doesn't
  - Allows to run your own Android/Fedora/Buildroot on your cheap tablet
  - Slightly better than the Allwinner code
  - Works well for the hobbyist that just wants to play with his new toy
- Cons
  - Stuck in the past
  - Doesn't think about the long-term view
  - Only keeps hacking more without thinking about upstreaming changes to lessen their burden
  - Not *much* better than Allwinner code
  - Support for new "generation" SoCs?

Mainlining

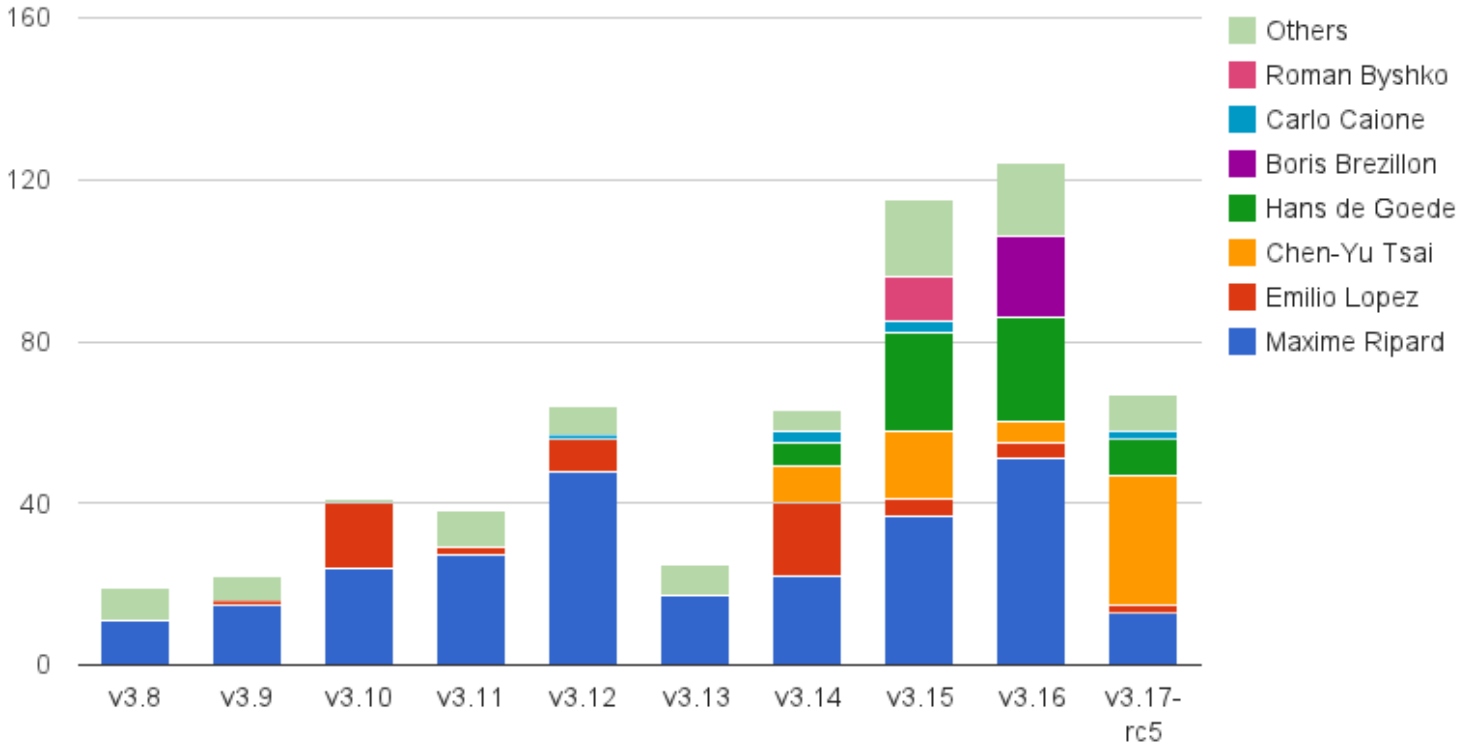
Maxime Ripard
Free Electrons



How it started

- Started in late 2011
- First code merged in Linux 3.8, for the Allwinner A13
- Was not doing much at the time
- Was only able to boot to an initramfs
- Other SoCs and features quickly followed

Contributions



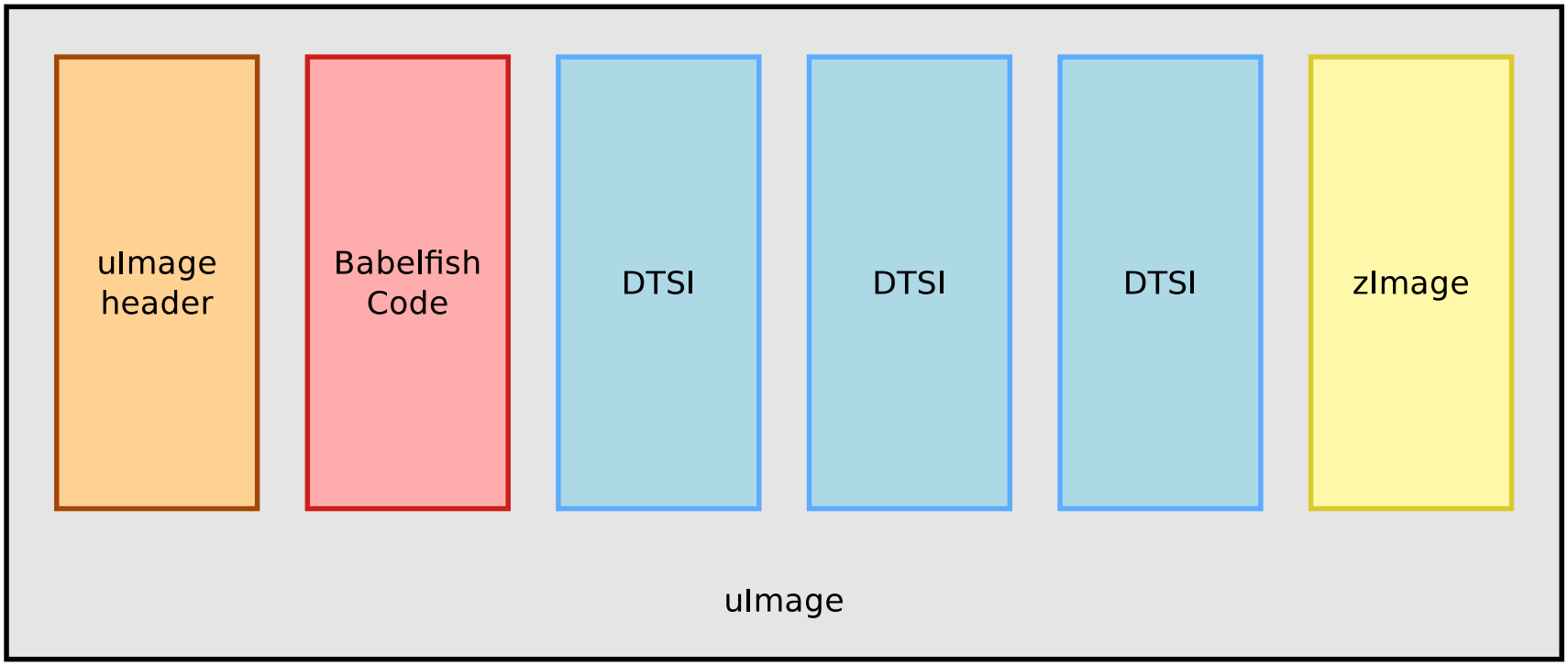
Challenges

- Rewrite most of the core drivers to use the new subsystems: CCF, pinctrl, dmaengine
- Move away from FEX, to DT
- Clean up the code to make it fit for submission
- Get the code accepted

Babelfish

- Couldn't boot a generic kernel on every board out there anymore
- Worked on sunxi-babelfish, a runtime FEX-to-DT translator
- Behaves like a regular u-boot kernel image, and embeds DTSI and the DT-based kernel image
- At runtime, will lookup the data from the FEX file, and construct the DT from the skeletons it has, and pass it to the DT kernel
- Enables to run a DT-based kernel on every cheap tablet out there, without modifying the bootloader.
- Not very used, and support only a few devices
- Only to be seen as a legacy solution

Babelfish

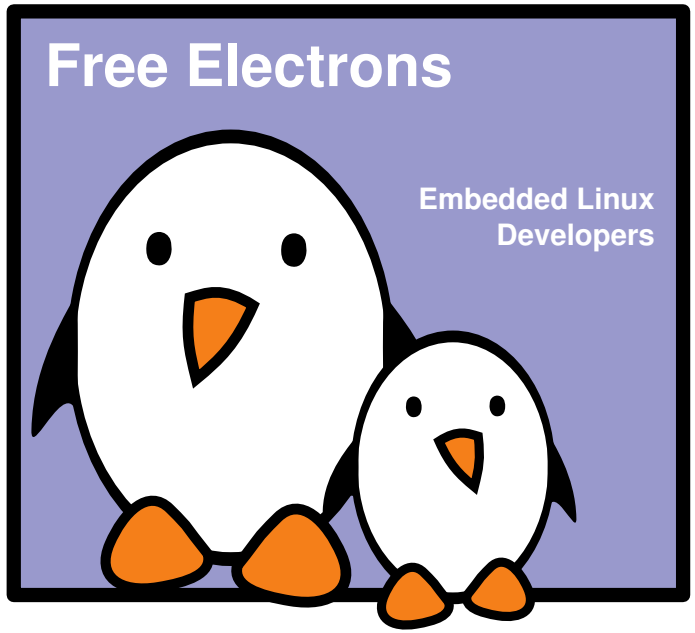


Current Status

- All the core stuff works, only DMA is missing on the older SoCs
- Most of the latest fancy stuff: PSCI, Virtualization, etc.
- Network works
- Storage start to works too: SATA and MMC are merged, MMC is pending
- Pretty much everything you need for a headless system works nowadays.
- The only things missing are the user-friendly one: Display, Audio, touchscreen, etc.

What's next

Maxime Ripard
Free Electrons



Allwinner's usage of the code

- Latest kernel still based on Linux 3.4
- They are using some of our code though (clock, pinctrl)
- Are switching to standard APIs (dmaengine, ASoC)
- But are still using FEX
- Joined Linaro

TODO-List

- Pretty much all the hard work now
- NAND is working, but is going (slowly) through the mainlining process
- Audio is working for some SoCs, and is in the cleanup phase
- Video is still an issue
- Work on newer SoCs (A33, A80, A83)
- Plenty of things to do, you're welcome to join!

Lessons Learned

- Take your time
- Know when to start from scratch and when to clean up
- You want documentation
- Have as much hardware as you can
- You don't want to be alone in there

Questions?

Maxime Ripard

maxime@free-electrons.com

http://free-electrons.com/pub/conferences/2014/elc/ripard-mainlining-of-out-of-tree-socs/