Embedded Linux — How To Get Started

Capitole du Libre 2025

Michael Opdenacker

Root Commit

Nov. 16, 2025

Capitole du Libre

root commit

Michael Opdenacker

root commit

Embedded Linux consultant and trainer

- https://rootcommit.com/about/michael-opdenacker/
- Former founder of Bootlin
- New founder of Root Commit
- Offering consulting on embedded Linux, kernel and Yocto
- Offering embedded Linux training courses with a focus on practical activities, interactivity and learning techniques. https://rootcommit.com/training/
- Free Software enthusiast and advocate (member of April.org)

April
promouvoir et défendre le logiciel libre

Embedded Linux is Great for Tinkering

root commit

- Hardware is cheap, sometimes free, like free software 😉
- Software is free (otherwise you're at the wrong conference)
- Power is low
- Connectivity is easy
- Be independent, self reliant

Project Ideas

Present at Capitole du Libre with an Embedded Board

root commit

✅

- Raspberry Pi Zero 2W
- 4x ARM Cortex-A53 CPU at 1 GHz
- 18 EUR at https://kubii.com!
- 512 MB of RAM, enough for many applications!

https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/

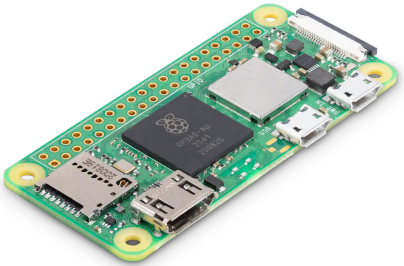
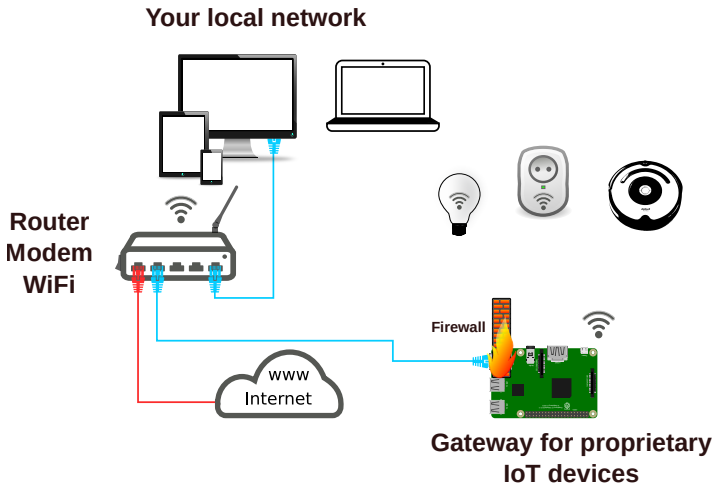
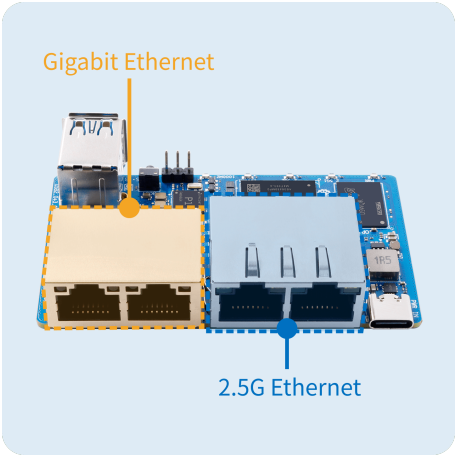


Image credits: https://www.raspberrypi.com/

Project Idea: IoT Gateway



IoT Gateway — Suggested Hardware



OrangePi R2S

- Starts at 32 EUR (2 GB version)
- 8 core RISC-V CPU
  New open-source instruction set!
- Initial support in mainline Linux 6.19
- Will be supported by Yocto soon

https://www.orangepi.org

Image source: https://www.orangepi.org

Project Idea: Break a Device

Image source: Wikipedia

But breaking is not a project!

I meant try to break into the device and reverse engineer it!

🤣

Project Idea: Open a Proprietary Device

root commit

More generally, you can contribute to "Degoogling" devices working through a proprietary cloud.

You could join the https://github.com/codykociemba/NoLongerEvil-Thermostat project to keep Google Nest Gen 1 and Gen2 alive.

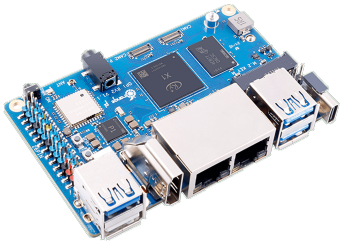
Image source: No Longer Evil Thermostat project

Project Idea: Mainline Support for Device

root commit

Adopt an embedded board
and port it to the mainline Linux kernel!

- Fun and addictive work
- Mostly a matter of describing the hardware with Device Tree and testing.
- Using the "vendor" kernel and board schematics.
- Great learning experience
- Add support to Yocto, U-Boot and Barebox too!
- Help me expand support for the Orange PI RV2 board (8x RISC-V cores, 8 GB, 2x GB Ethernet, 2x PCIe, 54 EUR)



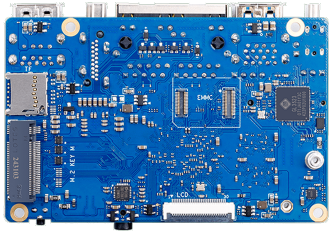


Image source: https://www.orangepi.org

Misc Ideas

root commit

- Home automation: optimize temperature through ventilation control and weather forecasts
- Weather station from a design on the Internet
- Wildlife camera from motion detection or AI camera
- Alarm clock, wake up through progressive lighting
- Implement your own cloud (contacts, photos, media... and be independent)
- "Edge" AI projects
- Lighting manager: LED strips, AI presence detector and light control
- Fridge / freezer temperature monitor
- Toxic gaz / smoke monitor
- You have more ideas that I do...

Hardware Suggestions

Misc Ideas

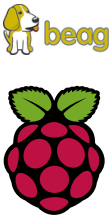
root commit

- Your robot vacuum cleaner (make it free)
- Your coffee machine (make it smarter to make you smarter)
- Any device you could drive
- Electronic boards that are supported by the mainline Linux kernel (check in `arch/<arch>/boot/dts/<cpu>/`)
- Electronic boards that are **not** supported by the mainline Linux kernel 😉

Reputable Hardware Providers

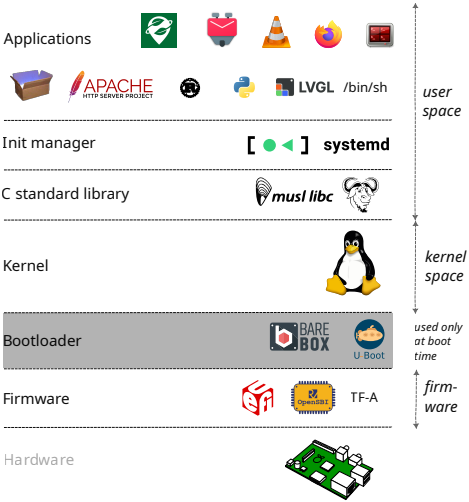
Community and budget friendly ones!

- BeagleBoard.org — https://beagleboard.org
  Open-source hardware boards, great longevity, great community support
- Raspberry Pi — https://raspberrypi.com
  Great longevity, outstanding community, good community support
- Orange Pi — https://www.orangepi.org
  Looking good, powerful, attractive and cheap boards, commitment to make boards as long as the chips are available.



How to Build Your System?

Here's What You Want to Build



A few tens of components to integrate!

How the system is built

Manually

- Build everything from source
- Great for learning (!), manageable for very simple or demo systems
- But not reproducible (!)

Copyright: Dargaud (Lucky Luke)

Using a binary distribution

- Regular desktop/server distributions: **Debian**, Ubuntu, Fedora, OpenSUSE
- Embedded friendly distributions: Alpine

debian

Using automated tools

- Buildroot
- Yocto Project

Most popular solution!

yocto PROJECT

Debian GNU/Linux

root commit

https://debian.org

- ⊕ Great for demos and prototypes. Ready-made images are available.
- ⊕ ARM, ARM64 and RISC-V are supported
- ⊕ Many extra packages are available
- ⊕ You can compile your own applications using native tools
- ⊕ Security updates available
- ⊖ But bigger filesystem
- ⊖ Lots of dependencies you can't remove, increases attack surface
- ⊖ Harder to rebuild from scratch

Between 5 and 15% of all devices?

debian

Buildroot

https://buildroot.org

- ⊕ Simplest solution to build a root filesystem with fixed features
- ⊕ Just containing what's needed!
- ⊕ Run `make menuconfig` and `make` and you have an image to flash.
- ⊕ Great for beginners who want something specific. Great for learning.
- ⊕ A pure community project, no big foundation and members behind
- ⊖ But fewer features, less flexible build engine
- ⊖ Supported by fewer hardware vendors

Between 25 and 45% of market share?

Yocto Project

https://yoctoproject.org

- ⊕ The second project from the Linux Foundation! More engineering and technical resources, supported by paying members.
- ⊕ Allows to create and maintain a custom distribution, but with more advanced features and expandability.
- ⊕ Very smart build engine, never builds the same thing twice.
- ⊕ The most inclusive and welcoming community I know.
  You really feel useful when you contribute.
- ⊕ Lots of third party layers supporting many types of hardware.
- ⊕ Long Term Support releases offering 4 years of updates and fixes.
- ⊖ But steeper learning curve (cliff?).

Between 45 and 70% of market share?

How to Build Your System — Key Takeaways

root commit

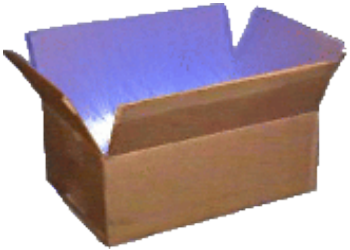
- Start with Debian to build your first prototype. Smooter transition to Embedded Linux.
- Move to Buildroot when ready to build a system with a fixed set of features.
- Switch to Yocto when there are things you can't do with Buildroot, or when you want to manage a family of devices.

What Components to Use?

BusyBox

https://busybox.net

- Lightweight implementation of many command line tools
- Size between a few hundreds of KB to just a few MB
- Just a few missing tools: `ssh`, `rsync`, `evtest`



Other Components

root commit

- Linux Kernel
- Dropbear: lightweight SSH client and server
- Ash, Hush: lightweight shells provided by BusyBox
- Musl C library: about 50% lighter that the full GNU libc
- Das U-Boot: Universal Bootloader
- BareBox: Versatile and flexible bootloader

All others are standard tools

How to Learn?

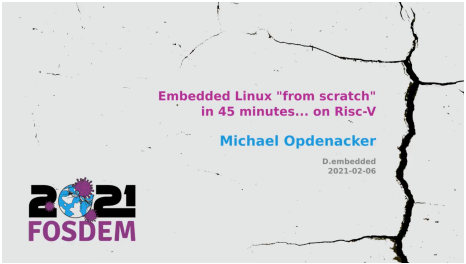
Learning By Doing

Get your hands dirty, but one goal at a time. For example:

- Get access to a serial console, access the bootloader and see the Linux kernel boot.
- Recompile your kernel with a modified configuration, or with a modified version. That's easier than you can think!
- Recompile the root filesystem.
- Customize the root filesystem and add your own applications.

What To Learn First?

- Git: for collaboration and project discipline
- Programming language: Rust (if you don't know C)
- Understand how a system boots
- Check Bootlin's free training materials: https://bootlin.com/docs/



https://www.youtube.com/watch?v=cIkTh3Xp3dA

Accelerate Your Learning

root commit

- Participate to a technical course
- Participate to an embedded Linux Conference.
  For example in Europe:
  - FOSDEM
  - Embedded Linux Conference Europe
  - Embedded Recipes
- Create an Anki flash card
  everytime you want to remember something you learned
  https://en.wikipedia.org/wiki/Anki_(software)

Anki flash cards

Root Commit's Yocto course materials with a majority of time spent on practical labs.

Interact With Your Peers

Getting Help

Understand people help you in their own interest. You are more likely to get help if:

- You show you have "done your homework",
  showing you're committed to helping find a solution.
- You ask your questions publicly
- You built a reputation as a reliable contributor.

Connect With Others

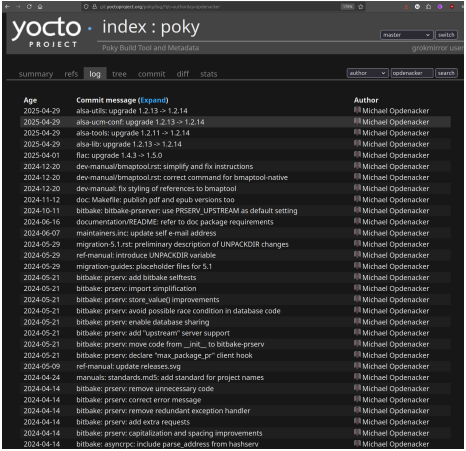
- Try to attend events in your area, country or region.
- Meeting people with shared interests boosts your learning and motivation.
- Don't hesitate to ask questions to speakers.

Yocto user and contributor meeting in 2023

Contribute

- Help others on mailing lists and chat.
- Share what you learn: blog posts, social media, conferences.
- Investigate and report issues you face.
- Implement your ideas or propose improvements.

Do all this publicly!



Yocto Project's cgit interface showing contributions from a specific person.

Going Further Than Embedded Linux

Zephyr®

Questions and Discussion

- Your projects?
- What's your favorite hardware?
- Share your experience

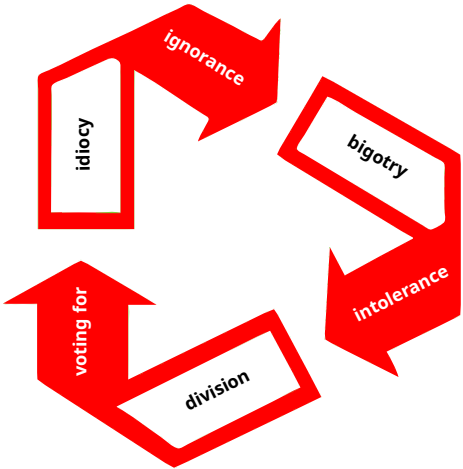
What To Remember

1. Pick a fun personal project
2. Choose hardware matching your requirements
3. Choose a solution to build your root filesystem
4. Learn by doing, one atomic step at a time
5. Accelerate your learning:
   reading, videos, conferences, training
6. Connect with other people sharing interests
7. Contribute to the projects that you use

This will give visibility to your skills, lead to more interesting job or business opportunities, providing funding to get back to 1.

Against Today's Popular Trend....

root commit



Invest in knowledge and be open to the ideas of others. You will stand out.

Thanks

root commit

To people who sent corrections and suggestions:

Lionel Orry

Questions? Comments?

- mo@rootcommit.com
- https://fosstodon.org/@MichaelOpdenacker
- XMPP: omichael@conversations.im
- Signal: rootcommit.01
- Slides available under the CC-By-SA 4.0 license
  https://rootcommit.com/pub/conferences/2025/cdl/embedded-linux-getting-started/
- Original blog post on this topic:
  https://rootcommit.com/2025/seven-steps-to-grow-your-embedded-linux-skills/
- Sources (LaTeX):
  https://gitlab.com/rootcommit/embedded-linux-getting-started/

Capitole du Libre