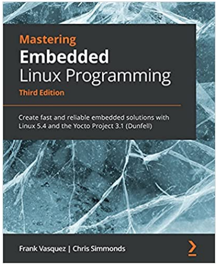
20-years-of-teaching-linux

Lessons I learned from my students

Chris Simmonds

Embedded Linux Conference Europe 2023

About Chris Simmonds



Mastering
Embedded
Linux Programming
Third Edition

Create fast and reliable embedded solutions with
Linux 5.4 and the Yocto Project 3.1 (Dunfell)

Frank Vasquez | Chris Simmonds

- Consultant and trainer
- Author of *Mastering Embedded Linux Programming*
- Working with embedded Linux since 1999
- Android since 2009
- Speaker at many conferences and workshops

"Looking after the Inner Penguin" blog at `https://2net.co.uk/`

Mastodon: @csimmonds@fosstodon.org
`https://fosstodon.org/@csimmonds`

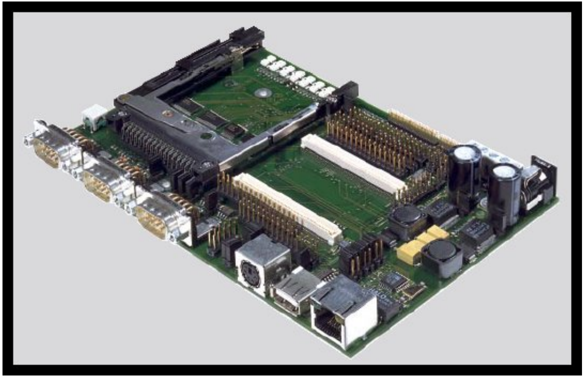
`https://uk.linkedin.com/in/chrisdsimmonds/`

Agenda

- A little history
- Teaching
- Conclusion

2002: The TQM823L

- CPU: 50 MHz MPC 823 (PowerPC)
- RAM: 16 MiB
- Flash: 4 MiB NOR
- Comms: RS-232, 10 MBit Ethernet, USB 1.2
- GPIO: 49 pins



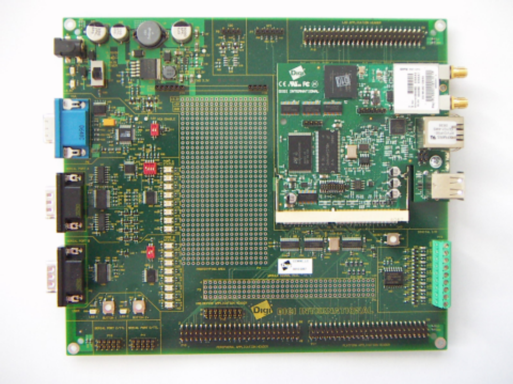
What was happening in 2002?

- I was teaching Embedded Linux and Linux Device Drivers using the TQM board from 2002 to 2007
  - Linux 2.4
  - toolchain from Denx
  - Roll Your Own: cross compile kernel, busybox, libc, etc. (*)
- At the beginning, Embedded Linux was considered with suspicion (how can you maintain quality if anyone can contribute?)
- By 2007, it was mainstream
- Replacement for RTOS (vxWorks, psos), DOS, and Windows CE

(*) Buildroot started in 2001, Open Embedded in 2003. Neither were particularly stable at that time, and supported limited targets

2007: The Digi ConnectCore Wi-9C

- CPU: NS9360 processor (ARM 926EJ-S), 155MHz
- RAM: 64 MiB
- Flash: 128 MiB NAND
- Comms: RS-232, 10/100 MBit Ethernet, USB 2, WiFi 802.11 b/g
- GPIO: 73 pins

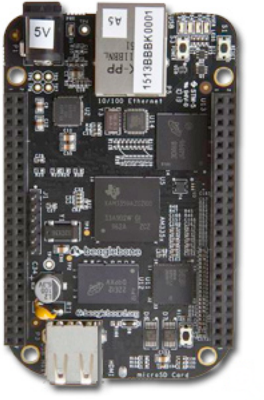


What was happening in 2007?

- I used the Digi board 2007 to 2010
- Linux 2.6, Montavista toolchain, then Angstrom
- Still RYO - some people using Open Embedded
- Teaching silicon vendors, set top box vendors, printer vendors, industrial control

2012: BeagleBone Black

- CPU: TI AM335x ARM Cortex-A8 1GHz
- GPU: Imagination PowerVR SGX530
- RAM: 512 MiB
- Flash: 2 or 4 GiB eMMC
- Mini USB OTG port, also provides power
- Comms: RS-232, 10/100 Ethernet, USB 2.0
- Mini HDMI connector



What was happening in 2012

- I used BeagleBone Black 2012 to 2020
- Mostly using Yocto (Yay!) and Buildroot (also Yay!)
    - I finally got to retire my RYO slides
- But also AOSP (Jellybean 4.1 through to Nougat 7.1)

2017: Raspberry Pi 3B

- CPU: BCM2837 4 x Cortex-A53 ARMv8 64-bit @ 1.2GHz
- RAM: 1 GiB
- Flash: none, have to use micro SD card
- Comms: 10/100 Ethernet, 4 x USB 2.0, WiFi 802.11 a/b/g/n (2.4 GHz)
- Bluetooth 4.2/BLE
- HDMI video output
- 40-pin header for HATs

What was happening in 2017

- The Pi was mostly for AOSP training, but also some Yocto
    - I never got Oreo 8.0 to work on the BBB
- Linux dominates the mid to high end embedded space
- Via Android, totally dominates mobile
- Automotive as well

- A little history
- Teaching
- Conclusion

How did I become a teacher?

- A consulting job where Linux was the obvious embedded OS
- A chance encounter at a conference with someone from a training company
- Some people who wanted to know about Embedded Linux

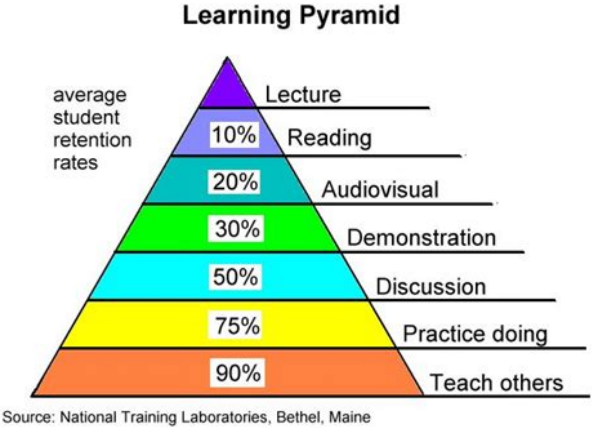
Was it easy at the start?

- No
  - writing materials for a 5 day course takes a long time
  - getting the timings right is tricky: first time I ran out after 4 days, day 5 was a recap of the first 4
  - constant fear that the people I was teaching knew more than me
    - not an issue, they would not be taking the course if they did
  - constant fear that the people I was teaching were smarter than me
    - of course they are smarter than me, get used to it!

What do people want to know?

- basic tech
- hands-on experience - labs are more important than slides
- confidence that this is a viable way to go

Rates of information retention



Source: National Training Laboratories, Bethel, Maine

Ways that people learn

- Top down
  - work from general principles to specifics
  - **Deductive**
- Bottom up
  - work from specifics to general principles
  - **Inductive**
- Best to combine both: begin with general principles, then specific examples then more general stuff, then ...

Learn by doing

- Hands-on labs have a much greater impact than a presentation alone
  - 75% vs 10%
- It’s OK if students make mistakes in labs
  - actually it makes a better learning experience if they do
  - if you make no mistakes, you are not learning things
- Gives students opportunities to explore: if there is time, see what happens if you vary things a bit

Live demos: good and bad

- Live demos and live coding are OK, but no substitute to the student doing the tasks themselves
- Problems with live demos
    - people miss-remember what you typed
    - and miss-understand the objective of the demo
- Live demos work well for
    - short demos, "OK, this is what it looks like, this is how the system will react" ... e.g. this command takes longer to run that you might think
    - also OK if the session is recorded: people can rewind and pause

Questions are good

- Opportunity from the teacher to find out student’s interests
- If they missed something I said earlier, that’s my fault, not theirs
- There are no bad questions
- If you don’t know the answer, don’t make one up
  "That’s a really good question, but I don’t have a really good answer.
  I’ll have to get back to you on that"

Learn from your students

- The unexpected question
  "Hmm: nobody asked that before, let me do some research and get back"
- The unexpected result of a lab "I have never seen that error message before. What did you do?"
- Be open to comments and criticism - there are probably people in the room who know more that me on specific topics
- Listen, invite them to stand up and talk to the rest of the class
- Be happy when someone points out an error on your slide

Fun things happen

- The exercise was to flash and LED on and off
  the solution was a slow fade from off to on and back
- The exercise was to control a toy missile launcher with commands to move up/down left/right
  The solution was to use opencv for face recognition and to track someone and then fire

- A little history
- Teaching
- Conclusion

What are my takeaways?

- I encourage you to become teachers, to lead discussions, to guide workshops
  - it’s fun
  - it’s a great way to learn
  - you help your colleagues
  - you help the community
- We need to spread the word - otherwise we keep making the same mistakes

Call to action

- Give talks in your company
- Attend local meetings
- Upstream something
- Contribute at meetings ... ask (good) questions, be engaged
- **Teach others**

Questions?

Slides at
`https://2net.co.uk/slides/elc/20years-teaching-csimmonds-elce-2023.pdf`

Mastodon: @csimmonds@fosstodon.org
`https://fosstodon.org/@csimmonds`

`https://uk.linkedin.com/in/chrisdsimmonds/`