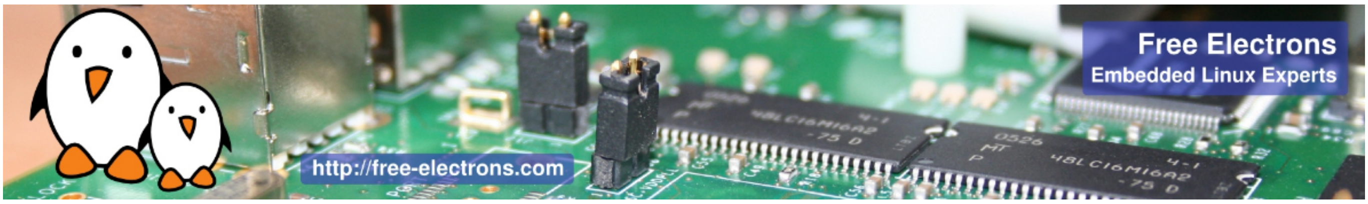
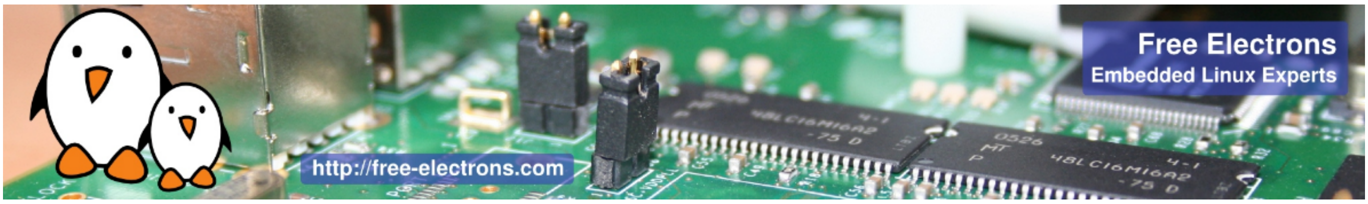


Boot Time Optimization Training

1-day session

Title	**Boot Time Optimization Training**
Overview	Measuring boot time Reducing user space boot time Reducing kernel boot time Bootloader optimizations Advanced techniques and alternatives Practical labs with ARM boards (SAMA5D3x evaluation kits from Atmel).
Materials	Check that the course contents correspond to your needs: `http://free-electrons.com/doc/training/boot-time`.
Duration	**One** day - 8 hours. 50% of lectures, 50% of practical labs.
Trainer	One of the engineers listed on `http://free-electrons.com/training/trainers/`
Language	Oral lectures: English or French. Materials: English.
Audience	People developing embedded Linux systems. People supporting embedded Linux system developers.
Prerequisites	**Knowledge and practice of Unix or GNU/Linux commands** People lacking experience on this topic should get trained by themselves, for example with our freely available on-line slides: `http://free-electrons.com/blog/command-line/` **Knowledge and practice of embedded Linux system development**



Required equipment	**For on-site sessions only.** Everything is supplied by Free Electrons in public sessions. • Video projector • PC computers with at least 4 GB of RAM, and Ubuntu Linux installed in a **free partition of at least 30 GB. Using Linux in a virtual machine is not supported**, because of issues connecting to real hardware. • We need Ubuntu Desktop 12.04 (32 or 64 bit, Xubuntu and Kubuntu variants are fine). We don't support other distributions, because we can't test all possible package versions. • **Connection to the Internet** (direct or through the company proxy). • **PC computers with valuable data must be backed up** before being used in our sessions. Some people have already made mistakes during our sessions and damaged work data.
Materials	Print and electronic copies of presentations and labs. Electronic copy of lab files.

Morning

Lecture - Principles	**Lab - Measuring boot time**
• How to measure boot time • Main ideas	• Flashing the board and accessing its serial line • Measure the initial boot time

Lecture - Userland optimizations	**Lab - Reducing boot time in user space**
• Methodology • Using bootchart • Optimize init scripts • Reduce size (C library, compiler optimizations...) • Use an initramfs • Toolchains	• Regenerate the root filesystem with Buildroot • Use bootchart to measure boot time • Simplify user space scripts

Afternoon

Lecture - Kernel optimizations

- Reducing kernel size
- Choosing the right compression method
- Reducing kernel initialization time

Lab - Reducing kernel boot time

- Recompile the kernel, switching to an initramfs
- Use `initcall_debug` to find the biggest time consumers
- Reduce the number of modules
- Tune kernel command line parameters

Lecture - Bootloader optimizations

- Reducing kernel image loading size
- Removing bootloader features
- Skipping the bootloader

Lab - Reducing bootloader time

- Reduce boot time by using the Barebox bootloader
- Optimize Barebox

Lecture - Advanced techniques

- More kernel tweaks
- More user space techniques
- Using strace
- Using oprofile
- Alternatives: suspend to RAM, hibernating, checkpointing