Android System Development Training

4-day session

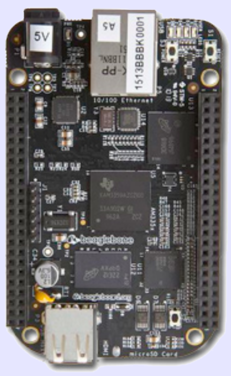
Title	**Android System Development Training**
Overview	Understanding the Android Internals Understanding the Android Build System Customizing Android for a specific hardware Extending the Android framework Practical labs with the ARM-based BeagleBone Black board.
Materials	Check that the course contents correspond to your needs: `https://bootlin.com/doc/training/android`
Duration	**Four** days - 32 hours (8 hours per day). 50% of lectures, 50% of practical labs.
Trainer	One of the engineers listed on `https://bootlin.com/training/trainers/`
Language	Oral lectures: English or French. Materials: English.
Audience	Engineers porting Android to new boards Engineers developing products with Android
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: `https://bootlin.com/blog/command-line/` **Basics of Java programming**

Required equipment	**For on-site sessions only** Everything is supplied by Bootlin in public sessions. • Video projector • PC computers with at least 8 GB of RAM, a CPU at least equivalent to an Intel Core i5 and Ubuntu Linux installed in a **free partition of at least 60 GB. Using Linux in a virtual machine is not supported**, because of issues connecting to real hardware. • We need Ubuntu Desktop 12.04 (Xubuntu and other variants are fine). We don't support other distributions, because we can't test all possible package versions. • **High Speed Connection to the Internet** (direct or through the company proxy), fast enough to download the several gigabytes of Android source code. • **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.

Hardware

The hardware platform used for the practical labs of this training session is the **BeagleBone Black board**, which features:

- An ARM AM335x processor from Texas Instruments (Cortex-A8 based), 3D acceleration, etc.
- 512 MB of RAM
- 2 GB of on-board eMMC storage (4 GB in Rev C)
- USB host and device
- HDMI output
- 2 x 46 pins headers, to access UARTs, SPI buses, I2C buses and more.



Part 1 - Compiling and booting Android

Lecture - Introduction to Android

- History
- Actors involved
- Introduction to the Android architecture

Lab - Setup

- Install the tools required to compile
- Fetch the source code (*If the network bandwidth is not sufficient, we will provide a ready-to-use source code archive*)
- Get used to Android specific tools

Lecture - Android Source Code and Compilation

- How to use git, repo and gerrit to access sources
- How to find one's way in the code base
- How to compile Android (tools, targets, etc.)

Lab - First Compilation

Using the Android Emulator

- Compile a first root filesystem for the emulator

Lecture - Introduction to the Linux kernel

- Role and general architecture of the kernel
- Kernel features
- Understanding the development process.
- Legal constraints with device drivers.
- Kernel user interface (/proc and /sys)
- Kernel configuration.
- Native and cross-compilation. Generated files.

Lab - Compile and Boot an Android Kernel

Using the Android Emulator

- Compile and Boot an Android Kernel
- Extract the patches from the Android Kernel

Part 2 - Porting Android to a New Board

Lecture - Changes introduced in the Android Kernel

- Major functional changes introduced by Google
- Additions to the kernel
- Mainline kernel status of these patches

Lecture - Android Bootloaders

- What is a bootloader
- Bootloader examples
- The fastboot specifications from Android.

Lab - Supporting a board

Using the BeagleBone Black board

- Use the Android's build for the BeagleBone Black
- Boot Android on a real board
- Troubleshoot the glitches on the board

Part 3 - Device Development with Android

Lecture – Developing and debugging with ADB

- Presentation of ADB
- Available commands: transfer files, install packages, executing remote commands, log access, networking... all this done from the development machine.
- Examples of commands and combinations useful to debug

Lab – Use ADB

- Learn how to get the system log, to gain access to a shell on the device, push and pull files, etc.

Lecture – Android filesystem layout

- Know where the various software components are installed and mounted, and why it matters.

Lecture – Android build system

- Concepts introduced in the build system
- Architecture of the Makefiles
- Variables and functions available
- Compilation steps
- Add a new device to the build system

Lab – Add a native library to the build

- Create an external library to control a USB rocket launcher.
- Add this library to the default Android build

Lab - System customization

- Add a device to the build system
- Customize the “About” info, build ID, boot and home screens in your system.

Lecture – Android Native Layer

- Discover the daemons handling the radio, external storage, launching applications, etc.
- Get to know the different components involved in the Android runtime, from the virtual machine to the media framework: StageFright, Flingers, Dalvik...
- Learn how hardware abstraction is done in Android

Lab – Add a native binary to the build

- Get to know the build system and the C library (Bionic) specifics.

Lecture – Android Framework and Applications

- Overview of the services, Content Providers and available applications in a standard Android build
- Structure of a Service / Content Provider
- How to access a native library from a Java app using the Java Native Interface (JNI)

Lab – Develop the Java interface to the native library

- Implement a Java interface to use the previously integrated library

Lecture – Android Application Development

- The application lifecycle
- The various application components
- How to access services
- How to use, access and manage the resources
- How apk packages are built and what do they contain

Lab – Write an app with the SDK

- Learn how to write and distribute an application using the Android SDK and its API.
- Practical case: write an Android application controlling the USB rocket launcher.