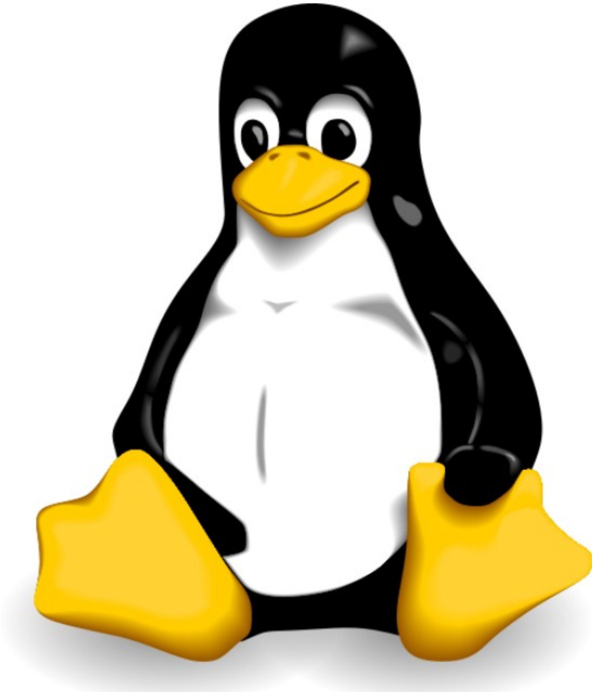
i.MX6 - Linux 1-day training design



Adeneo Embedded
April 2012

Document objectives

- Define the **overall objectives** and **prerequisites** of the training
- Define the **outline** of the training:
    - Lecture
    - Labs
    - Demos
- Estimate the **duration** of the different modules

Training objectives

- **Awareness training**

- Which tools do we want to use?
- How can we get our board up and running?
- How to build the different software components?
- How to write applications for our system?
- How to work efficiently with the i.MX6 processor?
- How do I design a custom platform based on the i.MX6?

- *There will be a strong focus on hands-on labs, preceded by explanatory slides.*

Training agenda

- **Morning**
  - Introduction to the i.MX6 platform (20 mins)
  - Introduction to Linux (45 mins)

  - Community and licensing (15 mins)
  - Development environment (20 mins)
  - The different components of a Linux system: programs, frameworks and libraries (10 mins)
  - Compiling the code using toolchains (10 mins)

  - Building a Linux system (10 mins)
  - Using LTIB (30 mins)
  - Ubuntu on the i.MX6 (20 mins)

Training agenda

- **Afternoon**
    - Working with U-Boot (30 mins)
    - Working with the Linux kernel (60 mins)
    - Application development (60 mins)
    - Adapting Linux for a custom i.MX6 design (20 mins)
    - Going further (10 mins)

Hardware requirements

- Use cheap equipment (easily purchased by trainees)
- **Development laptop**
  - + Power supply
  - VMWare Workstation installed
- **i.MX6 Evaluation board**
  - + Power supply
  - + LCD screen
- **Cables and peripherals**
  - USB to serial port (FTDI chip recommended)
  - USB SD Card reader
  - Network cable (to connect the PC directly to the board)
  - Several SD Cards (Sandisk is preferred as they are faster)

Prerequisites

- Knowledge of embedded systems
- Basic knowledge of C
- Basic understanding of operating systems in the context of embedded systems

Introducing the i.MX6

i.MX6 Intro - Module description

- **Duration**: 20 minutes
- **Objectives**:
  - Describe the i.MX family
  - Describe the i.MX6 line of processors
  - Introduce software support for different operating systems
- **Notes:**
  - This section is common to all the trainings

i.MX6 Intro - Module outline

- **The i.MX family**
- **The i.MX6**
    - **Hardware highlights**: *ARM micro-architecture, multi-core, compatibility, multimedia and graphics acceleration, capture and display, networking, power management, ...*
    - **Reference platforms**: *SABRE Lite, QSB, ...* (to be updated when finalized)
    - **Software support**: *Linux, Android, Windows Embedded Compact, QNX*

Introduction to Linux

Linux Intro - Module description

- **Duration**: 15 minutes (lecture) + 30 minutes (lab)
- **Objectives**:
  - Introduce Linux for embedded systems
  - Give an idea of the big picture: what are the main components that make a Linux system? What happens when we boot?
  - Give an idea of how the different pieces fit together before describing them in detail
  - Show how to start using the board

Linux Intro - Module outline

- **General Introduction**
    - History
    - Facts and figures
    - The advantages of using Linux
- **Overview of a Linux system**:
    - Bootloader
    - Kernel
    - Root filesystem (and how the files are organized)
- **Description of the boot process**

Linux Intro - Module outline

- **Lab: Flashing prebuilt Linux images on the development board.**
    - Getting familiar with the development environment, basic shell usage
    - Plugging the equipment
    - Configuring the boards
    - Flashing a prebuilt image
    - Booting the board
    - Understanding the layout of the SD Card

Community and Licensing

Community - Module description

- **Duration**: 15 minutes (lecture)
- **Objectives**:
  - Describe the Linux ecosystem, understand where the different pieces come from
  - Know where to find support and what to expect
  - Understanding the concept of “mainline”
  - Understanding the rights and obligations of free software. Dispel some misconceptions

Community - Module outline

- **Linux ecosystem: players and interactions**
    - Open-source communities
    - Freescale
    - Linaro (and how they are not equivalent to Freescale)
    - Other third-parties (e.g. Timesys)
- **Freescale BSPs**
    - Where to find documentation
    - iMXCommunity, support

Community - Module outline

- **Example of a Linux community: the kernel**
    - What is the “mainline”?
    - How does it compare to the Freescale BSP?
    - How to contribute?
- **Licensing:**
    - Rights and obligations of open-source software
    - Licenses primer: *GPL, LGPL, Apache, BSD, proprietary*
    - How to keep your intellectual property
    - More resources

Development Environment

Dev Env - Module description

- **Duration**: 20 minutes (lecture)
- **Objectives**:
    - Show how to set up a machine for Linux development
    - How to use a virtual machine
    - Introduce useful hardware and software tools for:
        - Coding
        - Debugging
        - Managing the code

Dev Env - Module outline

- **Choosing a Linux distribution**
- **Using a Virtual Machine**: benefits, recommendations and pitfalls
- **Hardware and software tools**:
  - Serial connection
  - TFTP and NFS (how they can dramatically increase your productivity)
  - JTAG: major vendors (Lauterbach, BDI3000, ARM, ...), OpenOCD
  - USB gadgets
  - QEMU

Dev Env - Module outline

- **Editors and IDEs**
  - Standard editors: vi, emacs
  - Eclipse
- **Version control:**
  - Why version control?
  - Quick introduction to Git

The different components of a Linux system

Components - Module description

- **Duration**: 10 minutes (lecture)
- **Objectives**:
  - Highlight one of the main advantages of Linux: reusability
  - Learn about the different components that can be reused when creating a Linux system
  - Get to know the major projects used in the embedded world

Components - Module outline

- Why **reusing** open-source components? How to choose?

- **Basic utilities** (includes Busybox)
- **Networking** tools
- **Graphical interfaces** (Qt, GTK, X/Windows, DirectFB)
- **Multimedia** (gstreamer, ALSA, OpenGL, ...)
- **Web Browsers** (and how to support HTML5)

Compiling the code using toolchains

Toolchains - Module description

- **Duration**: 10 minutes (lecture)
- **Objectives**:
  - Describe cross-compilation
  - See how to get and use a toolchain optimized for the i.MX6

Toolchains - Module outline

- Why cross-compiling?
- The components of a toolchain: *compiler, libc, binutils, ...*
- Getting a toolchain:
    - From source
    - Prebuilt
    - From the Freescale BSP
- Using cross-toolchains?
- Optimizing the compilation for the i.MX6

Building a Linux system

System - Module description

- **Duration**: 10 minutes (lecture)
- **Objectives**:
  - See what it takes to build a system
  - Understand how to manage the build process of a system through different options
  - Compare the different solutions

System - Module outline

- Building a system **manually**
- **Distributions**: *Ubuntu, Debian, ...*
- **Build systems** (introduction):
    - LTIB
    - OpenEmbedded, Yocto
    - Buildroot

- Comparison of the different solutions

Using LTIB

LTIB - Module description

- **Duration**: 10 minutes (lecture) + 20 minutes (lab)
- **Objectives**:
  - Build images quickly using LTIB
  - Know where to look for basic operations
- **Notes:**
  - LTIB might be replaced by OpenEmbedded or Yocto depending on what will be officially supported at the time of delivery

LTIB - Module outline

- What is **LTIB**?
- **Installing** LTIB
- **Configuring** LTIB
- **Building** the different components
- **Customizing** the filesystem image (e.g. adding applications)
- **Extending** the functionality of LTIB

- **Lab: Building and customizing an image with LTIB**

Ubuntu on the i.MX6

Ubuntu - Module description

- **Duration**: 20 minutes (lecture)
- **Objectives**:
  - Use a complete distribution and understand how it differs from a custom-built system
  - Show nice-looking demos (if available)
- **Notes:**
  - The lab might be replaced by a demo for the trainer if flashing the SD Cards takes too much time

Ubuntu - Module outline

- **Advantages and drawbacks** of using a prebuilt distribution on a system
- **Customizing** Ubuntu

- **Lab: Flashing and using Ubuntu on the i.MX6**
  - **Note:** Might require to use pre-flashed SD Cards since it takes a while to flash Ubuntu
  - Will show 3D and video capabilities (depending on software availability)

Working with U-Boot

U-Boot - Module description

- **Duration**: 10 minutes (lecture) + 20 minutes (lab)
- **Objectives**:
  - Getting the sources, building and flashing U-Boot
  - Introduction to the command-line interface
  - Know where to look when porting U-Boot (quick introduction)

U-Boot - Module outline

- **Getting** the sources
- **Navigating** through the sources:
  - Explanation of the different directories (and where to look if you want to adapt the bootloader to a new platform).
- **Lab: Building U-Boot (standalone and using LTIB)**
- **Lab: Flashing and customizing U-Boot**

Working with the Linux kernel

Kernel - Module description

- **Duration**: 30 minutes (lecture) + 30 minutes (lab)
- **Objectives**:
    - Identify the main kernel subsystems and their role
    - Getting the sources, building and flashing the kernel
    - Know where to look when porting the kernel (quick introduction)
    - Introduction to kernel advanced features: debugging, real-time, multi-core

Kernel - Module outline

- **Kernel architecture**
- **Getting** the sources
  - The different suppliers (mainline, Linaro, Freescale)
- **Navigating** through the sources:
  - Explanation of the different directories (and where to look if you want to adapt the kernel to a new platform).
  - Where do you find drivers?

Kernel - Module outline

- **Lab: Configuring the kernel**
  - Adding support for different drivers, removing unused ones
- **Lab: Building the kernel**
  - Using LTIB
  - Standalone
- **Lab: Flashing the kernel**
  - Loading from SD Card
  - Loading from TFTP

Kernel - Module outline

- **Debugging**:
  - Traces
  - Kernel debugging features: *tracers, kernel hacking options*
  - JTAG, KGDB
- **Real-time** support for Linux
- **Multi-core** support

Application development

Application - Module description

- **Duration**: 20 minutes (lecture) + 40 minutes (lab)
- **Objectives**:
  - Identify the tools that can be used for easy and productive application development
  - Cross-compile applications
  - Using Eclipse

Application - Module outline

- Development **languages**
- Development **environment**

- **Lab: Compiling an application**
  - Writing a simple simple Makefile
  - Using a cross-toolchain
- **Lab: Using NFS to speed up development**
- **Lab: Using Eclipse to work with applications.**

  Debugging from the IDE

Adapting Linux for a custom i.MX6 design

Adapting - Module description

- **Duration**: 20 minutes (lecture)
- **Objectives**:
  - Identify the areas of interest when porting Linux to a custom i.MX6 design
  - Leverage Freescale/Adeneo's experience of porting Linux. Lessons learned
  - Optimize code/design for the i.MX6

Adapting - Module outline

- **Software tools** for hardware design (e.g. pinmux)
- **Porting activities:**
  - Bootloader
  - Kernel
  - Filesystem
- **Optimizing** applications to run on the i.MX6 (using hardware acceleration, multicore capabilities).
- **Manufacturing** tool
- **Lessons learned**

Going further

Going further - Module description

- **Duration**: 10 minutes (lecture)
- **Objectives**:
  - Provide more resources (books, websites) for additional information
  - Highlight Freescale's documentation and community website

Quizz (TBD)