



## Building embedded Linux systems using System Workbench

### Objectifs

- Creating Embedded Linux platforms using System Workbench
- Using and customizing System Workbench

*Labs are conducted on target boards, that can be:*

*Dual Cortex/A7 and M4F "STM32MP15-DISCO" boards from STMicroelectronics.*

*Quad Cortex/A9-based "SabreLite" boards from NXP.*

*Quad Cortex/A53 and M4F "imx8q-evk" boards from NXP.*

*We use the last Ac6 System Workbench for Linux "Classic Edition" version, using a recent Linux kernel.*

### Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
    - ▶ One Online Linux PC per trainee for the practical activities.
    - ▶ The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
    - ▶ QEMU Emulated board or physical board connected to the online PC (depending on the course).
    - ▶ Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
    - ▶ One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - ▶ One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - ▶ An installation and test manual is provided to allow preinstallation of the needed software.
    - ▶ The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

### Prerequisite

- Good C programming skills
- Knowledge of Linux user programming (see our [D0 - Linux user mode programming](#) course)
- Knowledge of Linux Embedded systems (see our [D1 - Embedded Linux with Buildroot and Yocto](#) course)
  - For those without a prior knowledge of Embedded Linux, see our [D1S - Embedded Linux with Ac6 System Workbench](#) course
- Preferably knowledge of Linux kernel and driver programming (see our [D3 - Linux Drivers](#) course)

### Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

## Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the trainee in his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verify that the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites, in agreement with their company manager if applicable.

## Plan

## Introduction to Ac6 System Workbench

- Overview
  - Eclipse
  - Kernel and modules
  - Platforms and Root file-systems
- The build system architecture
  - Building individual packages
  - Building platforms
  - Building Root file-systems
- Developing with System Workbench
  - Creating an application
  - Building the application
  - Debugging

*Exercise: Building a root file system using a pre-defined platform template*

## Developing applications with System Workbench

- Creating a Linux program
- Creating a library
  - Static library
  - Shared library
- Debugging on the target
  - Using an SSH connection
  - Debugging shared libraries

*Exercise: Create a small program, with a custom shared library, and debug it on the target*

## Creating a Linux Platform

- Creating a platform project
  - Importing a pre-configured platform
  - Creating a platform from scratch
- Configuring the platform
  - Source and installation directories
  - Link to a target Rootfs
  - Build configurations

*Exercise: Create and configure a minimum platform from scratch, using library packages*

- Populating the build environment
  - Import packages in the build environment
  - Build individual packages
  - Build the whole platform

*Exercise: Build the platform, manually building some packages*

- Adding packages to a platform

- From a library
- From an existing Eclipse project

*Exercise: Add the previously developed application to the platform*

- Creating a new package
  - Specifying the source
  - Patching the official sources
  - Adding package-specific resources
  - Adding package configuration directives

*Exercise: Add a new open-source package to the platform*

*Exercise: Compiling and customizing the kernel*

## Compiling and customizing the kernel

- Kernel projects
  - Creating a kernel project
  - Selecting the architecture and configuration
  - Customizing the configuration
  - Compiling the kernel

*Exercise: Configure and compile the kernel in the platform*

- Module projects
  - Creating a module project
  - Linking it to a kernel project
  - Creating and building modules

*Exercise: Add and configure an external module*

## Creating a Root File-System

- Creating a rootfs project
  - Creating the rootfs structure
  - Add files to the base structure
- Edit standard configuration files
  - File systems
  - Initialization
  - Starting applications
- Creating and populating the root filesystem
  - Linking the file system to the platform
  - Installing platform components
  - Installing libraries

*Exercise: Create the root filesystem for the platform just built*

## Managing Package Libraries

- Creating a Library
  - Adding packages from a platform
  - Creating packages in the Library
- Importing a library
- Exporting a library

*Exercise: Create a library with the kernel, module and application created*

## Renseignements pratiques

**Inquiry : 1 day**