Linux

Installing, programming and writing drivers

Industrial applications are more and more often performed using an embedded version of Linux. In addition, the very specific environment in which run these systems sometimes make it necessary to adapt the Linux installation to the hardware environment.

Ac6-trainingtrainings not only teach you how to build applications on embedded Linux, but also how to adapt the operating system to your hardware or environment when the need arises.

Main Courses

D0 - Linux user mode programming Programming Embedded Linux Applications for Linux

D1 - Embedded Linux with Buildroot and Yocto Building and installing an embedded and real-time Linux platform Insta lling Linux on an embedded system is a common yet often difficult task. Many Open Source tools are supposed to make things easier, but must be properly controlled to obtain satisfactory results. This training presents you the most common of these tools (crosstool-NG, buildroot, OpenEmbedded, Yocto, System Workbench for Linux ...) and how to use them. The problems due to porting the u-boot bootloader and implementing a Linux BSP are also discussed as well as a porting methodology.

D1S - Embedded Linux with Ac6 System Workbench Implementing Linux on Embedded Systems Installing Linux on an embedded system is a common yet often difficult task. Ac6 System Workbench was designedis to make things easier and to be easily extended. This training presents you the architecture and needs of an Embedded Linux platform and explains how to build it using Sysetm Workbench for Linux. The problems due to porting the u-boot bootloader and implementing a Linux BSP are also discussed as well as a porting methodology.

D1Y - Embedded Linux with Yocto Building embedded Linux platforms using Yocto Installing Linux on an embedded system is a common yet often difficult task. The Yocto project is meant to make things easier, but must be properly controlled to obtain satisfactory results. This training presents you the architecture and needs of an Embedded Linux platform and explains how to build it using Yocto. The problems due to porting the u-boot bootloader and implementing a Linux BSP are also discussed as well as a porting methodology.

D3 - Linux Drivers Writing Linux Drivers This course covers the various techniques needed to write Linux (2.6 and 3.x) drivers, bus management (PCI. ..), hot-plug and auto-configuration of devices as well as the specific problems due to multi-core and advanced processors.

D4 - Real-time Linux Real-time Linux with RT-Preempt patch and Xenomai This course presents the various solutions for a real-time Linux and the tools to measure real-time performances

D5 - Embedded GUI Graphical User Interfaces for Embedded Linux

D7 - Power Management in Linux Drivers Writing drivers with power management support This course delves into the concepts of Linux drivers interaction with power management features of the Linux kernel.

D8 - USB Linux Drivers Writing USB-2.0 and USB-3.0 host and gadget drivers on Linux This course details the Linux driver model, the USB hotplug and power management architecture to write USB host (client) drivers as well as gadget drivers.

Q1 - Embedded GUIs with Qt Embedded GUIs with Qt This course covers the installation and use of Qt to make embedded GUIs

Y1 - Yocto Project Development Building a Linux Embedded image using Yocto Installing Linux on an embedded system is a common yet often difficult task. The Yocto project is meant to make things easier, but must be properly controlled to obtain satisfactory results. This training presents you the architecture of Yocto and how to parameterize it to fit your needs.

Y2 - Yocto Project Expert Advanced Yocto Project usage and adaptation This course expects you to already know how to build a Linux platform using Yocto (see our Y1 - Yocto Project Development course)

Y12 - Comprehensive Yocto Project Usage This course is the combination of the Y1 - Yocto Project Developmentcours e and Y2 - Yocto Project Expert course; it is intended for engineers that need to fully understand the Yocto build environment and be able to tailor it to their needs.

Additional Courses

E1 - Eclipse Utilisation de l'environnement de développement Eclipse pour C, C++ et Java(TM)

G1 - Android Installation Android installation on a hardware platform Installing Android on a new platform is a complex process; you need to port first the Linux kernel then install the Android platform. Even if using an existing Android Open Source Platform, the process to create an usable image is quite complex. This course will explain all the required steps, from building the kernel and the platform from source code to tailoring the boot process and creating test applications.

G2 - Android Programming Programming applications for the Android platform Android was designed to allow quickly creating powerful and ergonomic interfaces for embedded, resource constrained, systems; however, due to th elimitations of the underlying hardware, Android applications are totally different from standard applications. This course will explain how they are structured and how Android allows to combine portability and performance in applications.

G3 - Android Internals Android Frameworks and HAL Implementation Installing Android on a new platform is a complex process requiring a deep understanding of the internals of the Android frameworks and the Hardware Abstraction Layer. This course explains how the frameworks are structured and can be adapted to a platform on which a basic Android port already exist.

G5 - Android for Industrial System Control Building friendly interfaces for industrial systems with Android New industrial systems need sophisticated and ergonomic user interfaces. Building these with traditional GUI toolkits may be cumbersome and difficult. Android may simplify these tasks, allowing industrial application developers to benefit from the tools developped for consumer electronics devices.

RT1 - Real Time and Multi-Core programming Programming Linux real-time and multi-core systems, avoiding common pitfalls Real-time and embedded code, especially targetting multicore processors, cannot be effectively tested; it must be validated before coding. This training help you master mutitask and real-time programming of multi-core processors, understanding how to effectively solve problems using the primitives provided by the underlying Operating System.

SW1 - System Workbench for Linux Building embedded Linux systems using System Workbench Installing Linux on an embedded system is a common yet often difficult task. Ac6 System Workbench was designedis to make things easier and to be easily extended. This training presents you the architecture of Ac6 System Workbench and how to parameterize it to fit your needs.