FA1 - i.MX27 implementation + LTIB

This course describes the i.MX27 multimedia processor and Linux Target Image Builder tool

Objectives

- The course details the hardware implementation of the i.MX27 microcontroller.
- The boot sequence and the clocking are explained.
- The course explains all parameters that affect the performance of the system in order to easily perform the final tuning.
- A description of all internal peripherals is provided.
- An overview of the ARM926EJ-S core helps to understand issues caused by cache and MMU.
- The course ends with practical labs explaining how to generate a Linux image as well as a Root File System, by using a tool called LTIB [Linux Target Image Builder].

This course has been delivered to several companies developing multimedia equipments.

A lot of programming examples have been developed by ACYSYS to explain the boot sequence and the operation of complex peripherals.

- They have been developed with GNU compiler and are executed under Lauterbach debugger.
- Furthermore, a host desktop running Fedora Linux is used to generate Linux image and Root File System during labs on LTIB.

A more detailed course description is available on request at formation@ac6-formation.com

Prerequisites

- This course provides an overview of the ARM926 core. Our course reference course R1 - ARM7/9 implementation details the operation of this core.
- The following courses could be of interest:
  - USB Full Speed High Speed and USB On-The-Go, reference course IP2 - USB 2.0
  - Ethernet and switching, reference course N1 - Ethernet and switching

Plan

ARCHITECTURE OF i.MX27

Overview

- ARM core based architecture
- Clarifying the internal data paths
- Highlighting the purpose of the 2 central interconnect units: MAX and M3IF
- Organization of a board based on i.MX27
- Mapping
**CORE PLATFORM**

**THE ARM926EJ-S CORE**
- Presentation of the core
- Operating modes
- Pipeline
- ARM vs Thumb instruction sets, interworking
- Branch instructions
- C-to-Assembly interface
- Exception mechanism
- Debug facilities

**THE ARM9 PLATFORM**
- AHB slave device latencies
- MAX parameterizing
- ARM Interrupt Controller [AITC]

**HARDWARE IMPLEMENTATION**

**RESET AND CLOCKING**
- Clock distribution
- Power-up sequence
- Low power modes, clock gating
- System boot mode selection
- Bootstrap mode operation

**SYSTEM CONTROL**
- GPIO module
- General Purpose Input interrupt request capability
- Signal description

**ACCESSING EXTERNAL MEMORIES**
- Description of the Master Arbitration and Buffering [MAB] unit
- Description of the M3IF arbitration [M3A]
- Enhanced DDR SDRAM controller
- NAND flash controller, boot from flash
- Programming the chip-selects

**STANDARD PARALLEL INTERFACES**
- ATA controller
- PIO mode
- Ultra DMA mode
- FIFO receive and FIFO transmit alarms
- MSHC
- Transfer protocol
- Error management
- SDHC
MULTIMEDIA UNITS

DMA CONTROLLER

VIDEO PROCESSING UNITS

AUDIO RELATED INTERFACES

SECURITY MODULES

COMMUNICATION CONTROLLERS
### Transfer sequence
- I2C interfaces
- I2C protocol basics
- Master vs slave
- Transfer sequence
- UART
- IrDA modulation / demodulation
- Support for Smart Card
- Flow control
- USB
- Explaining what is OTG
- High-speed operation
- EHCI support
- Full speed operation
- Endpoint configuration
- Fast Ethernet Controller [FEC]
- Buffer management, based on Buffer Descriptors
- Incoming frame filtering mechanisms
- VLAN support

---

### LCD CONTROL
- LCDC
- LCD screen format
- Standard panel interface for common LCD drivers
- Graphic window on screen
  - SLCDC
- Interface to an external display controller
- Transferring images and controls from DDR to the external controller

---

### LTIB

---

### GENERATING THE LINUX KERNEL IMAGE
- What is required on the host before installing LTIB
- Common package selection screen
- Common target system configuration screen
- Building a complete BSP with the default configurations
- Creating a Root Filesystems image
- Re-configuring the kernel under LTIB
- Selecting user-space packages
- Setup the bootloader arguments to use the exported RFS
- Debugging Uboot and the kernel by using Trace32
- Adding a new package
- Other deployment methods
- Creating a new package and integrating it into LTIB

---

### Renseignements pratiques

**Durée :** 4 jours  
**Prix :** 0 € HT