

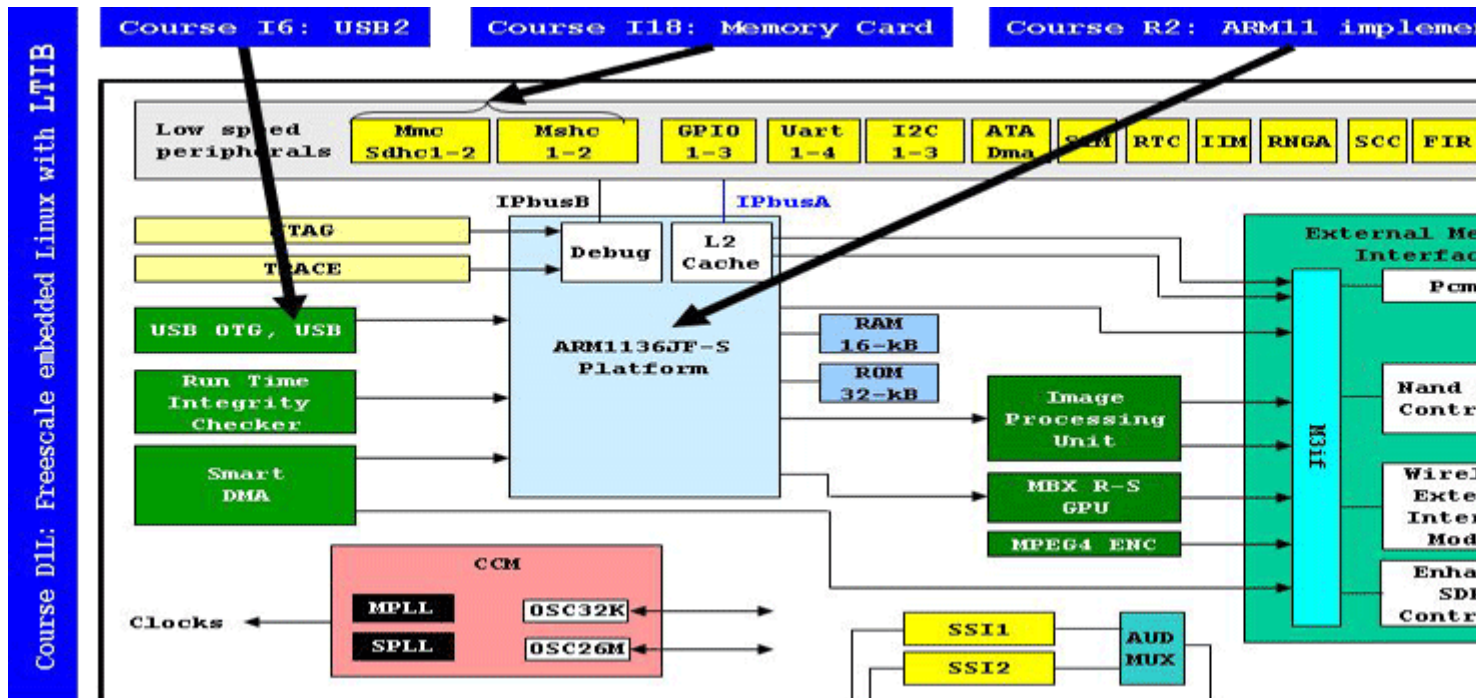
## FA2 - i.MX31 implementation + LTIB

*This course describes the i.MX31 multimedia processor and Linux Target Image Builder tool*

### Objectives

- The course details the hardware implementation of the i.MX31 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 ARM1136 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].
- Products and services offered by ACSYS:
  - ACSYS has developed FFTs (floating-point and fixed-point) optimized for ARM cores, based on SIMD instructions supported by the ARM1136.
  - Contact [training@ac6-training.com](mailto:training@ac6-training.com) to obtain informations about the performance of these FFTs.
  - ACSYS is able to assist the customer by providing consultancies. Typical expertises are done during board bringup, hardware schematics review, software debugging, performance tuning.
  - ACSYS has also an expertise in programming the SDMA, a simple OS-agnostic driver has been developed to explain how to manage scripts.

### Related courses



### Prerequisites

- Knowledge of ARM1136JF-S is recommended, see our course reference R2.
- Knowledge of USB is recommended, see our course reference [IP2 - USB 2.0course](#)
- ACSYS also offer a large set of courses on Linux.

## 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.
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

## Target Audience

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

# Course Outline

## ARCHITECTURE OF i.MX31

### Overview

- Clarifying the internal data paths : AHB bus, peripheral buses
- Highlighting the purpose of the 2 central interconnect units : MAX and M3IF
- Organization of a board based on i.MX31

## ARM11 PLATFORM

### THE ARM1136JF-S CORE

- Presentation of the core, architecture and programming model
- Operating modes : user, system, super, IRQ, FIQ, undef and abort
- ARM vs Thumb instruction sets, interworking
- Branch instructions, implementation of C call and return statements
- Level1 cache operation
- Memory management unit
- C-to-Assembly interface
- Exception mechanism, handler table
- Debug facilities

### THE ARM11 PLATFORM

- MAX parameterizing
- ARM Vector Interrupt Controller
- Level 2 cache operation

## HARDWARE IMPLEMENTATION

### RESET AND CLOCKING

- Clock distribution
- PLL output frequency calculation
- Power-up sequence
- Low power modes, clock gating
- Global reset vs warm reset

- System boot mode selection

## **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]
- Introduction to DDR SDRAM
- Enhanced DDR SDRAM controller
- NAND flash controller, boot from flash

## **STANDARD PARALLEL INTERFACES**

- ATA controller
- MSHC
- SDHC

## **MULTIMEDIA UNITS**

### **SMART DMA CONTROLLER**

- Scheduler
- CRC calculation unit
- SDMA initialisation
- Instruction description

### **VIDEO PROCESSING UNITS**

- Video acquisition
- MPEG4 encoder
- Image Processing Unit
- Graphics accelerator

### **AUDIO RELATED INTERFACES**

- SSI interfaces
- Digital audio multiplexor

### **COMMUNICATION CONTROLLERS**

- 1-wire interface
- Configurable SPI
- I2C interfaces
- UART
- USB

## **LTIB**

### **GENERATING THE LINUX KERNEL IMAGE**

- Introducing the tools required to generate the 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
- Command line options
- Adding a new package
- Other deployment methods
- Creating a new package and integrating it into LTIB