



## **RM0 - Cortex-M0 implementation**

**This course covers the Cortex-M0 ARM core**

### **Objectives**

- This course is split into 3 important parts:
  - Processor architecture
  - Software implementation
  - Hardware implementation.
- A tutorial has been developed by ACSYS to facilitate the understanding of Cortex-M0 low level programming, therefore labs can be replayed after the course.
- The course explains how to design a SoC based on Cortex-M0, clarifying the operation of the interconnect and the debug facilities integrated in the CPU.

*Labs are run under RVDS*

*A more detailed course description is available on request at [info@ac6-training.com](mailto:info@ac6-training.com)*

### **Prerequisites**

- Basic knowledge of processor or DSP.

### **Plan**

#### **First day**

#### **CORTEX-M0 ARCHITECTURE**

- Instruction pipeline, single-cycle multiplier
- Internal bus matrix, fixed memory map
- Deterministic instruction execution timing
- Highlighting the differences between Cortex-M0 and Cortex-M3
- Architecture of a SOC based on Cortex-M0 : the NXP LP1100

#### **ARM V6-M PROGRAMMING**

- Program registers, xPSR format
- Writing the whole code in C language
- Thumb 16-bit instruction set
  - Direct and indirect branches

- Arithmetical instructions
- Load and store instructions
- Load and store multiple instructions
- RVDS library functions, divide
- System instruction, Thumb-2 16-bit instructions
- Process stack pointer, supervisor call instruction
- A tutorial developed with RVDS4.0 will be used to allow attendees to become familiar with Cortex-M0 low level programming

## **DEBUG**

- Coresight overview
- CPU-dependent coresight units, breakpoints, watchpoints
- Serial Wire Debug, extrat functionality over JTAG using 2 wire interface
- Optional Serial Wire Trace port (SWV)

## **Second day**

### **EXCEPTION MECHANISM AND LOW POWER MODES**

- Exception vs interrupt
- Automatic state saving on exception entry and exit, CISC approach
- Nested Vectored Interrupt Controller
- Interrupt priority levels, nesting
- Tail-chaining and late arriving interrupts

### **LOW POWER MODES**

- Standby and deep sleep with state retention
- Event vs interrupt
- Optional wake-up interrupt controller
- Non-Maskable interrupt
- SysTick hardware timer

### **EMBEDDED SOFTWARE DESIGN**

- Application startup
- Placing code, data, stack and heap in the memory map, scatterloading
- Reset and initialisation
- Placing a minimal vector table
- Further memory map considerations, 8-byte stack alignment in handlers
- Long branch veneers
- A tutorial developed with RVDS4.0 will be used to allow attendees to become familiar with ARM IDE

### **HARDWARE IMPLEMENTATION**

- Bus architecture, von Neuman operation
- Single-data transactions, zero-latency 32-bit interface
- Address pipelining
- Sequential transfers
- AHB-lite specification

## Renseignements pratiques

**Duration :** 2 days  
**Cost :** 1250 € HT



SARL au capital de 15400€ - SIRET 449 597 103 00026 - RCS Nanterre - NAF 722C - Centre de Formation : 19, rue Pierre Curie - 92400 Courbevoie  
Siège social et administration : 21, rue Pierre Curie - 92400 Courbevoie - Tél. 01 41 16 80 10 - Fax. 01 41 16 07 78

Last site update: Mon 13 Feb 2012 02:28:18 PM CET

<http://www.ac6-formation.com/>