

# NP1 - LPC21XX/LPC22XX microcontroller implementation

This course covers NXP ARM-based MCU family

## Objectives

- ▶ The course details the hardware implementation of the LPC2294 microcontrollers.
- ▶ The boot sequence and the clocking are explained.
- ▶ The training helps to become familiar with the development environment chosen by the customer.
- ▶ Practical labs on integrated peripherals are based on I/O functions provided by NXP.
- ▶ The course focuses on the low level programming of the ARM7TDMI core.
- ▶ The course provides examples of internal peripheral software drivers.
  
- ▶ Note that ACSYS does not sell emulation probes and IDEs. Consequently this course has not been designed to convince attendees to buy a particular IDE. The unique objective consists in providing sufficient knowledge to attendees so that they can successfully design a system based on LPC21XX/LPC22XX.
  
- ▶ This course has been delivered several times to companies developing embedded systems, such as voltage counters.  
*A lot of programming examples have been developed by ACSYS to explain the boot sequence, the vector table and the operation of embedded peripherals.*
  - *They have been developed with 2 different IDEs : Keil and IAR.*
  - *Consequently for on site course, it is up to the customer to select the IDE under which labs will be run.**A more detailed course description is available on request at [training@ac6-training.com](mailto:training@ac6-training.com)*

## Prerequisites and related courses

- ▶ This course provides an overview of the ARM7TDMI core. Our course reference [R1 - ARM7/9 implementation](#) course details the operation of this core.
- ▶ The following course could be of interest:
  - CAN bus, reference [IA1 - CAN bus](#) course

## Plan

### **INTRODUCTION TO LPC2210 AND LPC2294**

#### Overview

- ▶ ARM core based architecture
- ▶ ARM7 local bus
- ▶ AMBA AHB/APB internal buses
- ▶ The main three blocks : platform, core and input / output peripherals
- ▶ APB Bridges
- ▶ Memory mapping, internal flash (2294) and SRAM

## **THE PROCESSOR CORE**

### **ARCHITECTURE OF THE ARM7TDMI CORE**

- ▶ Presentation of the core, architecture and programming model
- ▶ Operating modes : user, system, super, IRQ, FIQ, undef and abort
- ▶ Pipeline, calculation of the CPI
- ▶ Effects of branches and exceptions on the performance
- ▶ ALU data path

### **SOFTWARE IMPLEMENTATION, V4T SPECIFICATION**

- ▶ Parameterizing the linker to define sections
- ▶ Branch instructions, implementation of C call and return statements, long branch veneers
- ▶ ARM vs Thumb instruction sets, interworking
- ▶ ARM instruction set
- ▶ Inline barrel shifter
- ▶ Access to memory-mapped locations, addressing modes
- ▶ Arithmetical and logic instructions
- ▶ Thumb instruction set, highlighting restrictions with regard to ARM instruction set
- ▶ Compiler hints and tips, optimisations supported by RVCT
- ▶ Stack management
- ▶ Benefits of condition set capability in ARM state
- ▶ C-to-Assembly interface, ATPCS specification

### **EXCEPTION MECHANISM**

- ▶ Reset
- ▶ FIQ vs IRQ
- ▶ Exception return instructions
- ▶ Latency estimation, impact of load and store multiple instructions
- ▶ Organization of the handler table, priority decoder, pre-emption and nesting
- ▶ ISR header and footer routines
- ▶ Development of a generic exception handler

### **INTEGRATED DEBUG FACILITIES**

- ▶ JTAG interface
- ▶ Debug facilities, hardware breakpoint
- ▶ Executing code from RAM to take benefit of software breakpoints

## **PLATFORM**

### **THE VECTORED INTERRUPT CONTROLLER**

- ▶ Assigning a priority to each interrupt source
- ▶ Steering external interrupts and local interrupts to either the core FIQ or IRQ
- ▶ Developing a generic interrupt handler performing nesting according to peripheral priorities defined by the user
- ▶ Integrated timers
- ▶ Using timers to understand the operation of the VIC

### **SYSTEM CONTROL**

- ▶ Pin connect block
- ▶ Clocking

- ▶ Reset and wake-up timer
- ▶ Low power modes
- ▶ Watchdog timer
- ▶ Real-Time clock

### **ON-CHIP FLASH MEMORY (2294)**

- ▶ Organization
- ▶ Erase sequence
- ▶ Program sequence
- ▶ In system programming via serial port
- ▶ On-chip bootloader

### **EXTERNAL MEMORY CONTROLLER**

- ▶ Address decoding
- ▶ Chip-select registers
- ▶ Parameterizing the memory bank registers to support external burst flash

### **INTEGRATED I/Os**

### **SERIAL INTERFACES**

- ▶ I2C basics
- ▶ I2C controller
- ▶ UART controller
- ▶ SPI and SSP interfaces
- ▶ CAN protocol basics
- ▶ CAN controller (2294)

## **Renseignements pratiques**

**Duration : 4 days**  
**Cost : 2370 € HT**