



## **STR2 - STR750F implementation**

**This course covers STR750 ARM-based MCU family**

### **Objectives**

- The course details the hardware implementation of the STR750F microcontrollers.
- The boot sequence and the clocking are explained.
- The course focuses on the low level programming of the ARM7TDMI core.
- The course provides examples of internal peripheral software drivers and explains how to interact with the software package provided by ST.
  
- 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 STR750.
  
- This course has been delivered several times to companies developing embedded systems, such as communication systems.

*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 [info@ac6-formation.com](mailto:info@ac6-formation.com)*

### **Prerequisites and related courses**

- This course provides an overview of the ARM7TDMI core. Our course reference [R1](#) details the operation of this core.
- The following courses could be of interest:
  - USB Full Speed High Speed and USB On-The-Go, reference [IP2](#)
  - CAN bus, reference [IA1](#)

### **Plan**

#### **INTRODUCTION TO STR750F**

#### **Overview**

- ARM core based architecture
- APB internal busses
- The main three blocks : platform, core and input / output peripherals

## **THE PROCESSOR CORE**

### **ARM7TDMI CORE**

- Presentation of the core, architecture and programming model
- Operating modes : user, system, super, IRQ, FIQ, undef and abort
- Pipeline
- ALU data path
- ARM vs Thumb instruction sets, interworking
- Access to memory-mapped locations, addressing modes
- Stack management
- Branch instructions, implementation of C call and return statements
- Benefits of condition set capability in ARM state
- C-to-Assembly interface
- Exception mechanism, handler table
- Debug facilities

### **PLATFORM**

### **INFRASTRUCTURE**

- APB Bridges, individual peripheral reset control, individual peripheral clock control
- Memory organization, linear 4 GB mapping
- Internal 16 kB SRAM
- Flash memory, bank and sector mapping, burst mode
- Program and erase sequences
- Interrupt controller
- ISR header and footer routines
- External interrupts Unit
- System timers : Real Time Clock, Watchdog timer

### **HARDWARE IMPLEMENTATION**

- Power supplies, external 3.3V, internal generation of 1.8V, related pins
- Low voltage detectors
- Clocking
- Reset causes
- Start-up sequence, fetch of the first instruction
- Boot configuration register
- Low power modes
- External Memory Interface
- Description of the programming interface related to the 4 external chip-selects

### **INTEGRATED I/Os**

### **NON COMMUNICATION ORIENTED INPUT / OUTPUT PERIPHERALS**

- DMA
  - Circular Buffer Management
  - Support for UART, SPI / SSI, Timers and ADC
- Timers
  - 16-bit timers, block diagram, clock selection and prescalers

- Output compare and input capture capabilities, force compare modes
- Output PWM mode, on-the-fly modification of the duty cycle
- Input PWM mode, pulse measurement
- Analog-to-Digital Converter
  - High impedance-analog input configuration
  - ADC features : 10-bit resolution, 0 to 2.5 V range
  - Round-robin or single channel mode
  - Clock timing
  - The Sinc decimation filter
  - Gain and offset errors

## COMMUNICATION CONTROLLERS

- I2C interface
  - I2C protocol basics
  - Slave mode vs master mode
  - Transmit and receive sequences
- SPI / SSI
  - SPI protocol basics
  - Queue mode operation
  - Transfer sequence
- UART
  - Queue operation mode
  - Time-out mechanism
  - LIN capability
  - SmartCard asynchronous protocol
- CAN controller
  - CAN protocol basics
  - CAN controller organization
  - Message objects
  - Filtering received messages
  - FIFO mode management
  - Configuring the bit timing
- USB slave interface
  - USB protocol basics
  - Buffer description block, buffer descriptor table
  - Double buffer usage to support isochronous and high throughput bulk transfers
  - Endpoint initialization

## Renseignements pratiques

**Duration : 4 days**  
**Cost : 1500 € 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: Tue 22 May 2012 10:50:29 CEST

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