

STM32 MICROCONTROLLER IMPLEMENTATION

Ref : 004692A

Duration : 4 days

OBJECTIVES

- The course details the hardware implementation of the STM32 microcontroller
- The boot sequence and the clocking are explained
- The course focuses on the low level programming of the Cortex-M3 CPU
- Practical labs on integrated peripherals are based on I/O function package provided by ST
- The various options of communication controllers are explained, particularly the USB and CAN controllers, so that drivers can be adapted to application needs

RELATED COURSES

- USB training (Ref.002606A)
- CAN training (Ref.002601A)
- Ethernet training (Ref.003367A)
- ARM Cortex-M3 System Design (Ref.004581A)

PARTNERS

- MVD Training is a **ST MCU Certified Training Partner**
- MVD Training is an **ARM Approved Training Centre**

PREREQUISITES

- A basic understanding of microprocessors and microcontrollers is recommended
- A basic understanding of digital logic or hardware / ASIC design issues would be useful but not essential
- A basic understanding of assembler or C programming would be useful but not essential

PRACTICAL LABS

- For on-site courses, labs can be run under the following environments : Keil μ Vision, or IAR Workbench
- For open courses, labs are run under IAR Workbench



Contact

Tel : +33 (0)5 62 13 52 32
 Fax : +33 (0)5 61 06 72 60
 training@mvd-fpga.com

Course also available
 customized

Next sessions, see : <http://www.mvd-fpga.com/en/formationsCalend.html>

TOPICS

INTRODUCTION TO STM32

- ARM Cortex-M3 core based architecture
- The main three blocks : platform, core and input / output peripherals

THE CORTEX-M3 CPU

- Presentation of the core, architecture and programming model
- Icode, Dcode and system buses
- Branch prediction mechanism
- Thumb-2 instruction set
- Access to memory-mapped locations, addressing modes
- Conditional execution
- Bit-banding
- Memory map
- C-to-Assembly interface
- Exception mechanism
- System tick timer
- System Control Block
- linker parameterizing
- Embedded software development with Keil
- Debug facilities

INFRASTRUCTURE

- AHB/APB Bridges, split transactions, error handling
- Bus Matrix, round-robin arbitration scheme
- Internal 20 KB SRAM
- 128-KB Flash memory
- Program and erase sequences
- External interrupt/event controller, wake-up event management
- System timers : Real Time Clock, Window Watchdog timer
- Backup registers, tamper detection, RTC calibration

HARDWARE IMPLEMENTATION

- Power supplies, external 3.3V, internal generation of 1.8V, related pins

- Low voltage detectors
- Clocking
- Selection of the boot mode through external pins
- Reset causes
- Start-up sequence, fetch of the first instruction
- Low power modes
- I/O Ports

ADVANCED CONTROL TIMER AND GENERAL PURPOSE TIMER

- 16-bit timers, block diagram, clock selection and prescalers
- Output compare and input capture capabilities, force compare modes
- One pulse mode
- Output PWM mode
- Input PWM mode, pulse measurement
- DMA start event

DMA CONTROLLER

- Request priority management between the 16 channels
- Circular buffer operation
- Memory-to-memory, peripheral-to-memory, memory-to-peripheral and peripheral-to-peripheral transfers
- Error management

ANALOG-TO-DIGITAL CONVERTER

- High impedance-analog input configuration
- ADC features : 12-bit resolution, 0 to 3.6 V range
- One-shot or continuous conversion
- Dual ADC mode
- Conversion on external trigger
- Utilization of a DMA channel

I2C INTERFACE

- I2C protocol basics
- Slave mode vs master mode

- Transmit and receive sequences

SERIAL PERIPHERAL INTERFACE [SPI]

- SPI protocol basics
- Queue mode operation
- Transfer sequence

USART

- Queue operation mode
- Hardware flow control
- Transmit and receive sequences
- LIN mode
- Smartcard interface

CAN

- CAN protocol basics
- CAN controller organization
- Filtering of received messages, acceptance filters
- FIFO mode management
- Test modes : loop back / silent modes
- Configuring the bit timing

USB OTG FULL SPEED INTERFACE

- USB protocol basics
- Buffer description block, buffer descriptor table
- DMA controller used to move data between buffers and EndPoints
- Endpoint initialisation
- Suspend / resume events

FLEXIBLE STATIC MEMORY CONTROLLER

- NOR / PSRAM interface
- NAND / PC CARD interface
- Memory bank mapping
- Address mapping and chip select signals
- LCD module interface

ETHERNET MAC

- Automatic CRC
- Flow control
- DMA transfers
- Address filtering
- IEEE 1588 PTP
- MII interface
- RMON/MIB
- LAN Wakeup frames / Magic packet
- VLAN tags

DIGITAL-TO-ANALOG CONVERTER

- DAC implementation
- DMA capability
- External triggers

SDIO

- Compliance
- Implementation
- DMA requests

DOCUMENTATION

Training manuals will be given to attendees during training in print.