oSTG - STM32 + FreeRTOS + LwIP

STM32 Development with FreeRTOS and the LwIP Stack

Objectives

- Get an overview on the Cortex-M architecture
- Understand the Cortex-M software implementation and debug
- Learn how to deal with interrupts
- Get an overview on STM32F4 architecture
- Describing the units which are interconnected to other modules, such as clocking, interrupt controller and DMA controller
- Describing some independent I/O modules like the ADC and GPIOs
- Getting started with the ST Drivers to program STM32 peripherals (The STM32Cube Library or ST Standard Peripheral Library)
- Understand the FreeRTOS architecture
- Discover the various FreeRTOS services and APIs
- Learn how to develop and debug FreeRTOS applications
- Getting started with the LwIP TCP/IP stack (Describing the STM32 Ethernet Controller, having a look on porting, describing the parameterizing, and developing application based on UDP and TCP protocols) (not available for STM32F0 family)
- The peripherals overview presented in this course can be detailed upon request (STR9 - STM32 Peripherals course)

Prerequisites

- Familiarity with C concepts and programming targeting the embedded world
- Basic knowledge of embedded processors
- Basic knowledge of multi-task scheduling

Course environment

- Theoretical course
  - PDF course material (in English)
  - Course dispensed using the Teams video-conferencing system
  - The trainer to answer trainees' questions during the training and provide technical and pedagogical assistance through the Teams video-conferencing system
- Practical activities
  - Practical activities represent from 40% to 50% of course duration
  - One Online Linux PC per trainee for the practical activities
  - One STM32 board connected to the online PC
  - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
- Downloadable preconfigured virtual machine for post-course practical activities.

Course duration

- 5 sessions (30 hours)

Target Audience

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

Course Outline

First Session

Cortex-M Overview

- ARMv7-M Architecture
- Cortex-M4 Architecture
- Registers and Execution States
- Privileges, Mode and Stacks
- Reset Behavior
- Exception and Interrupts
- The System Timer
- Memory Model
- Power Management

Exercise: Create a new project
Exercise: Interrupt Management

STM32F4 MCUs Architecture Overview

- ARM core based architecture
- Description of STM32Fx SoC architecture
- Clarifying the internal data and instruction paths: Bus Matrix, AHB-lite interconnect, peripheral buses, AHB-to-APB bridges, DMAs
- Memory Memory Organization
  - Flash memory read interface
  - Adaptive Real-Time memory accelerator, instruction prefetch queue and branch cache
  - Sector and mass erase
  - Concurrent access to RAM blocks
- SoC mapping
- Flash Programming methods
- Boot Configuration

Second Session

Reset, Power and Clocking

- Reset
  - Reset sources
  - Boot configuration, physical remap
  - Embedded boot loader
- Clocking
  - Clock sources, HSI, HSE, LSI, LSE
  - Integrated PLLs
  - Clock outputs
  - Clock security system
- Power control
  - Power supplies, integrated regulator
  - Battery backup domain, backup SRAM
  - Independent A/D converter supply and reference voltage
  - Power supply supervisor
  - Brownout reset
  - Programmable voltage detector

- Low power modes
  - Entering a low power mode, WFI vs WFE
  - Sleep mode
  - Stop mode
  - Standby mode

DMA

- Dual AHB master bus architecture, one dedicated to memory accesses and one dedicated to peripheral accesses
- 8 streams for each DMA controller, up to 8 channels (requests) per stream
- Priorities between DMA stream requests
- FIFO structure
- Independent source and destination transfer width
- Circular buffer management
- Double buffer mode
- DMA1 and DMA2 request mapping

Exercise: DMA FIFO mode

Hardware Implementation

- Power pins
- Pinout
  - Pin Muxing, alternate functions
- GPIO module
  - Configuring a GPIO
  - Speed selection
  - Locking mechanism
  - Analog function
  - Integrated pull-up / pull-down
  - I/O pin multiplexer and mapping
- System configuration controller
  - I/O compensation cell
  - External Interrupts / Wakeup lines selection
  - Ethernet PHY interface selection
- External Interrupts

Third Session

12-bit Analog-to-Digital Converter

- 12-bit, 10-bit, 8-bit or 6-bit configurable resolution
- Regular channel group vs Injected channel group
- Single and continuous conversion modes
- Scan mode
- External trigger option with configurable polarity for both regular and injected conversions
- Discontinuous mode
- Analog watchdog
- Dual/Triple mode (on devices with 2 ADCs or more)
- Configurable delay between conversions in Dual/Triple interleaved mode
- DMA request generation during regular channel conversion

Exercise: Get voltage from the potentiometer using, DMA transfer generation

Introduction to FreeRTOS

- The FreeRTOS Family
- FreeRTOS+Ecosystem
- Why use FreeRTOS

- FreeRTOS Code Structure

Scheduling

- Scheduler
- Schedulability

Task Management

- Creating Tasks
- Task Priorities
- Task States
- The idle task
- Delays
- Changing Task Priority
- Deleting Tasks
- Suspending Tasks
- Kernel Structures
- Thread Local Storage
- Kernel Interrupts on Cortex-M4
- Scheduling Traces
- Visual trace diagnostics using Tracealyzer

Exercise: Task Management
Exercise: Periodic Tasks

Fourth Session

Memory Management in FreeRTOS

- FreeRTOS Memory Managers
- Out of Memory management
- Stack overflow detection

Exercise: Context Switch Measurement

Resource Management

- Mutual Exclusion
- Critical Sections
- Mutexes
- Gatekeeper Tasks
- Lock-Free Data Structures

Exercise: Resource Management

Synchronization Primitives

- Queues
- Queues Sets
- Synchronization
- Events and Event Groups
- The Readers/writer problem
- Using Other Primitives within an ISR

Exercise: Queue Management

Fifth Session

Interrupt Management

- Tasks and Interrupts
- FreeRTOS Binary and Counting Semaphores
- Task Notifications
- Stream Buffers
- Message Buffers
- Interrupt Nesting
- Low Power Support

Exercise: Interrupt Management

Software Timer

- Software Timers
- Deferred Interrupt Handling

STM32 Fast Ethernet Controller Overview

- Architecture of the MAC
- Connection to PHY, RMII / MII
- Transmit and receive FIFO threshold setting
- Multicast and unicast address filtering
- Management interface
- Buffer and Buffer Descriptor organization
- Low level Drivers for STM32

LwIP introduction

- Overview
- Buffer and memory management
- LwIP configuration options
- Network interfaces
- MAC and IP address settings
- IP processing
- UDP processing
- TCP processing
- Interfacing the stack
- Application Program Interface (API)
- Standalone
- Netconn and BSD socket library

Appendix

Timers Overview

- Advanced-control timers TIM1 and TIM8
- 16-bit up, down, up/down auto-reload counter; 16-bit programmable prescaler
- Input Capture, Output Compare, PWM generation, One-pulse mode
- Synchronization circuit/ Controlling Timers external signals / Interconnecting several timers
- Interrupt/DMA generation
- Real Time Clock
- Independent BCD timer/counter; 16-bit programmable prescaler
- Daylight saving compensation programmable by software
- Two programmable alarms with interrupt function

- Automatic wakeup unit
- Reference clock detection / Digital calibration circuit
- Tamper detection

FreeRTOS-MPU

- The Cortex-M MPU
- The FreeRTOS-MPU Port
- Defining MPU Regions
- Creating User and System Tasks
- Practical Usage Tips

CMSIS – RTOS

- Overview
- Kernel Information and Control
- Threads Management
- Generic Wait Functions
- Communication and Resource Sharing
  - Semaphores
  - Mutex
  - Message Queue
  - Signal Events
  - Event Flags
  - Memory Pool
  - Mail Queue
- Timer Management
- Interrupt Service Routines