oSTG - STM32 + FreeRTOS + LwIP

Développement STM32 avec FreeRTOS et la pile réseau LwIP

Objectifs

- Obtenir un aperçu de l'architecture Cortex-M
- Comprendre l'implémentation et le débogage de logiciels sous l'architecture Cortex-M
- Apprendre à gérer les interruptions
- Obtenir une vue d'ensemble de l'architecture STM32F4
- Décrire les unités qui sont interconnectées à d'autres modules, comme l'horloge, le contrôleur d'interruption et le contrôleur DMA
- Décrire certains modules d'E/S indépendants comme l'ADC et les GPIOs
- Démarrer avec les pilotes ST pour programmer les périphériques STM32 (la bibliothèque STM32Cube ou la bibliothèque standard de périphériques ST)
- Comprendre l'architecture FreeRTOS
- Découvrir les différents services et APIs de FreeRTOS
- Apprendre à développer et déboguer des applications FreeRTOS
- Démarrer avec la pile LwIP TCP/IP (Décrire le contrôleur Ethernet STM32, jeter un coup d'oeil sur le portage, décrire le paramétrage et développer des applications basées sur les protocoles UDP et TCP) (non disponible pour la famille STM32F0)
- L'aperçu des périphériques présenté dans ce cours peut être détaillé sur demande (STR9 - Cours sur les périphériques STM32)

Pré-requis

- Familiarité avec les concepts C et la programmation visant le monde embarqué
- Connaissance de base des processeurs embarqués
- Connaissance de base de l'ordonnancement multi-tâches

Environnement du cours

- Cours théorique
  - SuppoCours théorique
  - Support de cours au format PDF (en anglais)
  - Cours dispensé via le système de visioconférence Teams
  - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
  - Les activités pratiques représentent de 40% à 50% de la durée du cours
  - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
  - Un PC Linux en ligne par stagiaire pour les activités pratiques
  - Une carte STM32 connectée au PC en ligne
  - Exemples de code, exercices et solutions
  - Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
- Une machine virtuelle préconfigurée téléchargeable pour les activités pratiques après le cours
- Au début de chaque demi-journée une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessairert de cours en PDF (en anglais)
  - Cours dispensé via le système de visioconférence Teams
  - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
  - Les activités pratiques représentent de 40% à 50% de la durée du cours
  - Un PC Linux en ligne par stagiaire pour les activités pratiques

    - Une carte STM32 connectée au PC en ligne
    - Le formateur a accès aux PC en ligne des stagiaires pour une assistance technique et pédagogique
- Une machine virtuelle préconfigurée téléchargeable pour les activités pratiques après le cours
- Au début de chaque demi-journée une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus

Durée

- Totale: 30 heures
- 5 sessions de 6 heures chacune (hors temps de pause)

Course Outline

Première 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

Deuxième 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

Troisième 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

Quatrième 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

Cinquième 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

Annexes

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