oRT5 - Zephyr Real Time Programming

Real-time programming applied to the Zephyr operating system

Objectives

- Learn how to develop, configure, debug and trace Zephyr applications
- Discover the real time multitasking concept
- Understand Real Time constraints
  - Determinism
  - Preemption
  - Interrupts
- Understand the Zephyr kernel Services
- Learn communication and synchronization mechanisms
- Interactions with processor architecture features
- Understand Zephyr memory management and data structures

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
  - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
  - Example code, labs and solutions
- Virtual Machine with System Workbench for STM32 with GNU ARM Eclipse QEMU
- Emulated board STM32F4-Discovery

Prerequisites

- Good C programming skills (see our oL2 - C Language for Embedded MCUscourse)

Duration

- Total: 24 hours
- 4 sessions, 6 hours each (excluding break time)
- From 40% to 50% of training time is devoted to practical activities
- Some Labs may be completed between sessions and are checked by the trainer on the next session

Plan

First Session

Real-Time Concepts

- Base real time concepts
- The Real Time constraints
- Multi-task and real time
- Tasks and Task Descriptors
  - Content of the task descriptor
  - List of task descriptors
- Context Switch

Introduction to Zephyr

- Zephyr Project
- Zephyr Ecosystem
- Why use Zephyr
- Install and use Zephyr
- Build and Configuration Systems
  - West
  - CMake
  - Kconfig and configuration overlay
  - Configuration tools: menuconfig and guiconfig

Zephyr Without Threads

- Operation without Threads
- GPIO
- Random Number Generation
- Utilities
- Data Structures
  - Single-linked List
  - Double-linked List
  - Ring Buffers

Exercice: *Hello World from Zephyr, configure and blink LEDs using Zephyr*
Exercice: *Manage Zephyr linked list and understand container_of macro*

Scheduling

- Task Scheduling and Preemption
  - Tick based or tickless scheduling
- Scheduling systems and schedulability proof
  - Fixed priorities scheduling
  - RMA and EDF scheduling
- Scheduling through Zephyr
  - Scheduling Algorithm
  - Cooperative Time Slicing
  - Preemptive Time Slicing

Second Session

Thread Management

- Thread Control Block
- Creating Threads
- Threads Priorities
- Thread States
- Main and Idle Threads
- Delays
- Changing Thread Priority
- Suspending Threads
- Kernel Structures
  - Simple linked-list ready queue
  - Red/black tree ready queue
  - Traditional multi-queue ready queue
- Thread Custom Data
- Scheduling Traces
  - Runtime Statistics
  - User-Defined Tracing
  - Percepio Tracealyzer

Exercice: *Create and manage threads*
Exercice: *Create periodic threads*
Exercice: *Create config overlay for visual trace diagnostics using Tracealyzer*

Memory Management in Zephyr

- Memory Managers
- Dynamic memory managers
  - K_heap
  - System heap
  - Memory Slabs
  - Memory Blocks
- Heap Listeners
- Stack Overflow detection

Exercice: *Understand dynamic memory allocation in Zephyr*
Exercice: *Display threads information and detect stack overflow*

Third Session

Resource Management

- Mutual Exclusion
- Critical Sections
- Mutexes
- Gatekeeper threads
- Atomic
- Lock-Free Data Structures
- SpinLocks

Exercice: *Implement mutual exclusion between threads*

Synchronization Primitives

- Synchronization

- Semaphores
- The Readers/Writer Problem
- Condition variables
- Events and Event Groups
- Polling

Exercice: *The producer-consumer problem, synchronize and avoid concurrent access problems*
Exercice: *Understanding event bit group by synchronizing several threads*

Fourth Session

Data Passing

- Message Queues
- Pipes
- Queues
  - FIFOs
  - LIFOs
- Mailboxes
- Stacks

Exercice: *Create a print gatekeeper thread using message queue*

Interrupt Management

- Threads and Interrupts
- Interrupts in zephyr
- Interrupts on ARM Cortex-M
- Handler thread
- Queue within an ISR
- Workqueue Threads
- Power Management

Exercice: *Understand how to wait on multiple events and interrupt safe APIs*
Exercice: *Understand how to pass data using Queues from an interrupt to a thread*
Exercice: *Create and submit work items from interrupts to custom WorkQueue*

Software Timers

- Timers
  - Defining a Timer
  - Using a Timer Expiry Function
- Timer types
  - One-shot timers
  - Auto-reload timers
- Timer Commands

Exercice: *Understand the use of one-shot and auto-reload timers*

Renseignements pratiques

Duration : 30 hours
Cost : 2930 € HT
Prochaines sessions : du 2 au 6 October 2023 - Ac6 - Courbevoie / Paris (France)