



oSEC7 - ARM TrustZone for Cortex-M based devices

Objectives

- Understand the ARM v8-M architecture and its security features
- Learn about ARMv8-M Memory Protection mechanism enhancement
- Configuring the Security Attribution unit
- How to manage Security access faults
- How to build and debug a secure and non-secure software

Course environment

- Students will be given access to a shared filesystem to save and share their work.
- PDF course material
- The labs will use a ARM Cortex-M33 based board

Prerequisites

- Programming skills: Some programming experience, particularly in C
- Basic knowledge of ARM Cortex-M implementations
- Basic understanding of Security Algorithms and Secure coding

Environnement du cours

- Cours 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.
 - Exemples de code, exercices et solutions.
 - Un PC Linux en ligne par stagiaire pour les activités pratiques.
 - Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique.
 - Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque session 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.

Course Outline

First Day

ARM Architecture and Security

- Overview of ARM TrustZone technology
- TrustZone Architecture
 - Overview of the TrustZone architecture
 - TrustZone-enabled processors and their features
 - Secure world and non-secure world
- TrustZone security
 - Overview of TrustZone security model
 - TrustZone-enabled Cortex-M
- Secure Software Design Considerations

ARMv8-M Memory Protection

- Memory types
- Access order
- Memory barriers, self-modifying code
- Memory protection overview, ARM v8 PMSA
- Cortex-M33 MPU and bus faults
- Region overview, memory type and access control
- Setting up the MPU

Exercise : Use the MPU to protect an area of memory against unintended access

Cortex-M TrustZone

- TrustZone-enabled Cortex-M processors and their features
- Security states
- Register banking between security states
- Stacks and security states
- Security Extension and exceptions
- Secure and Non-Secure states interactions
- Exceptions and the Security Extension
 - Handling Secure Exceptions
 - Handling Non-Secure Exceptions while in the Secure state
 - Returning from a Non-Secure exception to the Secure state
- The Security Attribution Unit (SAU)
- The Implementation Defined Attribution Unit (IDAU)
- Debugging TrustZone-enabled Cortex-M processors

Exercise : Implementing a minimal secure monitor

Exercise : Programming and Debugging a TrustZone application example