

## L70 - Real time project management

### Conduire un projet temps réel embarqué en C, C++ ou Java(TM).

Java est une marque déposée de Sun Microsystems

#### Objectifs

- Découvrir les concepts de base du multi-tâches et du temps réel
- Découvrir la méthode de conception temps-réel UML-RT
- Comprendre le fonctionnement d'une chaîne de compilation
- Maîtriser les difficultés de la programmation concurrente
- Connaître les standards applicables
- Découvrir les contraintes temps réel (déterminisme, interruptions, préemption...)
- Comprendre les implications des architectures des processeurs en contexte temps-réel (cache, pipeline,...)

*Cette formation est particulièrement adaptée aux personnes ayant à conduire des projets d'informatique embarquée et temps-réel, ou devant participer à toutes les étapes de ces projets, de l'analyse et la conception à la réalisation.*

*Les personnes désireuses d'un cours plus orienté vers le développement peuvent regarder également notre cours référence [L71 - Programmation temps réel](#) course*

#### Matériel

- Un PC et une carte ColdFire par binôme
- Chaîne de compilation croisée et sonde d'émulation BDM
- Machine virtuelle Java
- Manipulations et exercices en environnements natif et croisé
- Un support de cours ainsi que la disquette contenant les exemples

#### Pré-requis

- Connaissance de la programmation en C, C++ ou Java (niveau cours L2, L3 ou L4)
- Connaissance d'un microprocesseur souhaitée
- Connaissance de la programmation embarquée utile

#### Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
    - ▶ One Online Linux PC per trainee for the practical activities.
    - ▶ The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
    - ▶ QEMU Emulated board or physical board connected to the online PC (depending on the course).
    - ▶ Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
    - ▶ One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.

- ▶ One PC for two trainees when there are more than 6 trainees.
- For onsite trainings:
  - ▶ An installation and test manual is provided to allow preinstallation of the needed software.
  - ▶ The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

## Target Audience

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

## Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the trainee in his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verify that the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites, in agreement with their company manager if applicable.

## Plan

### Introduction au temps réel

- concepts temps réel de base
- contraintes particulières du temps réel
- programmation structurée et objet
- apports des techniques objets

### L'approche temps réel avec UML

- genèse d'UML
- modèles UML standards
- cycle de développement Objet
- contraintes liées à l'interprétation des diagrammes
- interprétation des diagrammes
- définition de nouveaux diagrammes
- cycle de développement avec RT UML

### Le langage de modélisation UML

- modélisation statique
  - cas d'utilisation
  - modèles de classes
- modélisation dynamique
  - diagrammes de Séquence
  - diagrammes de Collaboration
  - diagrammes Etats Transitions

### Les extensions pour le temps réel

- environnement / diagramme de contexte système
- contraintes / diagramme de contraintes
- comportement / diagramme d'états

- timings / diagramme de séquence étendu
- parallélisme / diagramme architecture logicielle
- architecture / diagramme architecture matérielle

### **Analyse des éléments constitutifs d'une chaîne de compilation**

- Explication des étapes du processus de génération de code en natif et en croisé
- Rôle du compilateur, de l'assembleur et du linker
- Paramétrage en fonction d'un mapping mémoire
- Découpage d'une application en fichiers distincts
- Le préprocesseur
- Les instructions define et include
- Ecriture de macros
- Précautions à prendre dans les headers pour éviter les redéclarations de variables
- Notion de projet, réalisation de bibliothèques

### **Particularités de la programmation dans le contexte embarqué**

- Les tableaux de pointeurs
- Accès aux champs d'une structure
- Déclaration de variables et de pointeurs sur type structuré
- Les formats big et little endian
- Les structures à champ de bits : modélisation des périphériques
- Les unions : une même zone mémoire peut être envisagée de différentes manières
- Utilité des tableaux de pointeurs sur des fonctions

### **Principe de fonctionnement d'un système Temps réel et embarqué**

- Notion de tâche
- Cadencement des tâches selon leur priorité, préemption
- Sauvegarde de contexte
- Nécessité d'un tick temps réel pour déclencher les commutations de tâches

### **Les standards du temps réel**

- Les standards d'environnements temps réel
  - POSIX
- Les langages à sémantique temps-réel
  - Java
  - Ada

### **Les traitements concurrents**

- Recensement des champs d'un descripteur de tâche
- Réalisation d'une structure chaînée des tâches en attente d'exécution
- Insertion d'un nouveau descripteur lors du chargement d'une nouvelle tâche
- Exemple de règles d'ordonnement: priorité évoluant en fonction du temps
- Réorganisation de la file lors de l'invocation de l'ordonnateur: préemption

### **Mise au point**

- Communication avec la cible
- Les différents niveaux de mise au point : C, assembleur
- Les fenêtres du debugger : source, mémoire, pile et variables
- Positionnement de points d'arrêt
- Analyse de la pile et extraction des stacks frames correspondant aux fonctions imbriquées

### **La programmation dans le contexte multi-tâches**

- Structures de données:
  - Listes simplement chaînées

- Listes doublement chaînées
- Listes circulaires
- Files d'attentes
- Piles

*Exercice: réalisation de listes chaînées utilisables en contexte multi-tâches*

- Gestion des accès concurrents
  - Variable simple
  - Structure de données
  - Entre tâches
  - Entre tâches et routines d'interruption

*Exercice: synchronisation et communication entre tâches*

- Gestion de la mémoire
  - Algorithmes
  - Gestion des fuites mémoire

*Exercice: mise en évidence et détection de fuites mémoire*

## Renseignements pratiques

**Duration : 4 days**

**Cost : 1850 € HT**