

# Formation : Systèmes embarqués, programmation assembleur ARM

Formation pratique - 4j - 28h00 - Réf. ARM

Prix : 2550 € H.T.



Ce cours intensif vous apprendra à mettre en oeuvre l'assembleur ARM, programmer les coprocesseurs des SoC "System on Chip" ARM9 et Cortex-A9 et optimiser les librairies. Il vous apportera tous les éléments nécessaires au développement de logiciels en utilisant ce type de technologies.

## Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- ✓ Maîtriser l'environnement de développement, le debugger, le compilateur, l'éditeur de liens
- ✓ Mettre en oeuvre des programmes en assembleur ARM
- ✓ Intégrer de routines assembleur dans un programme C
- ✓ Programmer les coprocesseurs des SoC ARM9 et CortexA9
- ✓ Optimiser les librairies

## Public concerné

Les informaticiens désireux d'acquérir une formation complète sur la programmation assembleur ARM et la mise en oeuvre de SoC ARMv5 à ARMv7.

## Prérequis

Connaissances de base de la programmation et des micro-processeurs. La connaissance du système Linux facilitera la mise en oeuvre des travaux pratiques.

Vérifiez que vous avez les prérequis nécessaires pour profiter pleinement de cette formation en faisant [ce test](#).

### PARTICIPANTS

Les informaticiens désireux d'acquérir une formation complète sur la programmation assembleur ARM et la mise en oeuvre de SoC ARMv5 à ARMv7.

### PRÉREQUIS

Connaissances de base de la programmation et des micro-processeurs. La connaissance du système Linux facilitera la mise en oeuvre des travaux pratiques.

### COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

### MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Méthodes et moyens pédagogiques

### Travaux pratiques

Les nombreux travaux pratiques seront effectués sous Linux, sous émulateur ARM ou sur une carte avec un processeur ARM.

### Méthodes pédagogiques

Présentation théorique des différents aspects de la programmation entrecoupée d'exercices pratiques permettant la mise en oeuvre immédiate des concepts.

## Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Programme de la formation

### 1 Introduction à ARM

- Présentation de l'architecture ARM : de v5 à v7.
- Le processeur ARM, fonctionnement, les registres.
- Les instructions ARM. L'architecture load/store.
- Architecture ARMv5 : TCM, Jazelle, domaines, MMU.
- Architecture ARMv7 : caches, mode monitor, trustzone.

#### Travaux pratiques

Prise en main de l'environnement de développement. Compilateur, éditeur de liens et simulateur. Ecriture d'un programme ARM.

### 2 Présentation de l'Assembleur ARM

- Les opérations de base de l'assembleur ARM.
- L'exécution conditionnelle et les branches.
- Les opérations arithmétiques et logiques.
- Accès à la mémoire, la gestion des piles.
- Les opérations atomiques.

#### Travaux pratiques

Ecriture de différents petits programmes en assembleur ARM.

### 3 Interface C et assembleur ARM

- Utiliser l'assembleur ARM pour optimiser les applications.
- Intégration de routines assembleur dans un programme C.
- Bibliothèques de fonctions assembleur.

#### Travaux pratiques

Ecriture d'une bibliothèque de synchronisation de threads en assembleur.

### MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

### MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

### ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr).

#### 4 Gestion des exceptions et des périphériques

- La gestion des exceptions et des interruptions par le processeur ARM.
- Le vecteur d'interruption.
- Les différentes façons de gérer une interruption.
- Présentation de quelques périphériques standard, PIC, timer, UART.

##### Travaux pratiques

Mise en oeuvre d'une interruption timer sur une carte versatile.

#### 5 Programmation système ARM9

- La gestion du processeur par un OS.
- Modes de fonctionnement. Passage en mode superviseur (SWI et SMC).
- Gestion du CPSR et des coprocesseurs.
- Le coprocesseur système CP15.
- La gestion de la MMU, des domaines et des caches.
- Les barrières, la Tiny Coupled Memory, le Fast Context Switch.

##### Travaux pratiques

Passage en mode superviseur et retour en mode utilisateur. Construction d'un mini-OS, activation de la protection mémoire par domaines.

#### 6 Programmation système CortexA9

- Le Cortex A9 pour sécuriser les applications sensibles.
- Mise en oeuvre de la protection par trustzone.
- Mise en oeuvre d'un moniteur sécurisé.
- Gestion des coprocesseurs, de la MMU et des caches L1 et L2.
- FCSE et ASID.

##### Travaux pratiques

Passage en mode moniteur et gestion d'un mini OS. Activation de la protection mémoire par trustzone.