

Formation Pratique

Développement de Logiciel Embarqué en Langage C

Présentation L'immense majorité des systèmes embarqués actuels se programment en Langage C ou C++. Bien que ces langages soient enseignés depuis très longtemps dans les cursus universitaires, force est de constater qu'ils le sont rarement dans le contexte du logiciel embarqué qui implique des pratiques et des exigences particulières. De plus, ce savoir scolaire est parfois un peu éloigné pour certains concepteurs ayant été plus actifs dans l'électronique et la conception numérique. Enfin, les ingénieurs de développement matériel (en langage HDL) ont de plus en plus besoin de pluri-compétences pour l'intégration du matériel avec le logiciel embarqué. Nos formations sont dans tous les cas des aides précieuses pour se (ré)approprier les compétences et les bonnes pratiques de programmation en langage C ou C++ (deux formations différentes) *dans le contexte spécifique du logiciel embarqué*.

La formation « Développement de Logiciel Embarqué en Langage C » de quatre jours est parfaite dans ces contextes.

Ce cours s'inscrit dans le cadre d'une action d'acquisition ou de perfectionnement des connaissances.

Les **Exercices Pratiques**, qui occupent environ **50 % du temps de la formation** sont des éléments clés du succès de ces formations, et sont essentiel pour l'apprentissage. Ils **assurent** également un **contrôle continu de l'acquisition des compétences**.

Les **Instructeurs ALSE** sont aussi et surtout des **Experts** qui utilisent à journée entière les Langages et les Méthodes de Conception qu'ils enseignent, pour concevoir et vérifier des systèmes complexes et performants. Ils savent partager leur savoir-faire avec passion et sont particulièrement appréciés des participants.

Pré-requis La participation à cette formation demande une **connaissance préalable des concepts de programmation**. Une certaine connaissance préalable du langage C est utile sans être absolument indispensable, mais ce cours n'est pas approprié pour les personnes n'ayant aucune connaissance de programmation.

Durée **Quatre (4) jours**, soit **28 heures effectives** de cours ;
Typiquement 9h30 → 18H, le premier jour et 9H → 17h30 les jours suivants.

Lieu Dans les locaux d'ALSE, **Paris XIII^{ème}**

Matériel Pour les cours publics, le matériel (station de travail et plateforme embarquée) est fourni pour la durée du training.

Objectifs pédagogiques

- ◆ Syntaxe & sémantique du langage C pour l'embarqué.
- ◆ Principes du logiciel embarqué et temps réel, Introduction aux outils temps réel.
- ◆ Programmer en C et mettre au point sur une plateforme micro-contrôleur.
- ◆ Accès aux périphériques, la gestion par Interruptions, les drivers
- ◆ Introductions aux OS temps réel, Introduction à l'optimisation de consommation.
- ◆ Les bonnes pratiques de programmation en C embarqué.

NB : à partir de quatre ou cinq stagiaires, une **formation sur site** est envisageable.

ALSE est un Organisme de Formation Professionnelle Continue déclaré auprès de la DIRECCTE depuis 1996 sous le numéro 26.21.01281.21 et enregistrée dans le Datadock. Cette formation peut donc être prise en charge dans ce cadre.

Programme de la Formation

Développement de Logiciel Embarqué en Langage C

Introduction to C

- Characteristics of Embedded Systems • C Language Overview
- Structure of a C Program • Identifiers
- Name Spaces and Scope
- Compilation & Linking
- MCU Boot Process
- C Best Practices for Embedded Systems

Variables, Types and Debugging

- MCU Architecture • Program Execution
- Variables
- Representing Numbers
- Types • Casting
- Debugging Embedded Systems

Operators and Hardware Manipulation

- Understanding Register Maps
- Operators • Bit Manipulation
- Modulus and Shifting
- Memory Addressing • Sizeof
- Ternary Operator
- Precedence Rules
- Best Practices for Embedded Systems

Basic Program Flow Control

- Software Design Cycle • Software Architecture
- UML • Flowcharts
- Round Robin Scheduling
- Statements • For and While Loops • If and Switch statements
- Infinite Loops
- Best Practices for Embedded Systems

Advanced Flow Control

- Introduction to Real-time Concepts
- Interrupt Basics • Interrupt Vector Tables
- Nesting and Priorities
- Software Interrupts
- Volatile keyword • Shared Data Problems and Solutions
- RMA Analysis
- Interrupts Best Practice

Advanced Types, Constants and Expressions

- Enumerations • Derived Types
- Literals
- Expressions and Evaluation
- State Machines • State Charts
- Software Architecture Concepts

.../...

Arrays and Pointer Basics

- Arrays • Multidimensional Arrays
- Strings • String Conversion
- Pointer Types • Pointers and Arrays
- Pointers Operations
- Best Practices for Embedded Systems

More Pointers and Strings

- Pointers to Pointers • Pointers to Constants • Constant Pointers
- String Libraries
- Manipulating Memory
- Best Practices for Embedded Systems

Functions

- Syntax • Variable Scope
- Recursion
- Inline Functions
- Software Metrics • Static Code Analysis
- Testing Techniques
- Best Practices for Embedded Systems

Structures and Unions

- Overview of Structures • Unions
- Driver Design • Defining APIs
- Driver Models
- GPIO Driver Example

Scheduling Techniques

- Arrays of Pointers to Functions
- Function Queue Scheduling
- Cooperative Scheduling • Scheduler Design
- Energy Profiling • Low Power Software Design

Declarations

- Syntax • Storage Class Specifiers
- Global Variables
- Type Qualifiers
- Linkage Identifiers
- Best Practices for Embedded Systems

Preprocessor

- #define • Macros • Precedence
- Conditional Compilation
- Warnings
- #pragma
- Predefined Macros

Real-Time Operating Systems

- Software Concurrency
- Tasks and States
- Task Synchronization
- Mutexes Semaphores & Queues
- GPOS vs RTOS

--oOo--