



## oL3 - Programmation C++ embarqué

### Objectifs

- Maîtriser les bases du langage C++
- Intégrer les templates C++ (code générique) dans les systèmes embarqués
- Maîtriser les aspects avancés du C++ tels que le polymorphisme, l'héritage simple et l'héritage multiple
- Redéfinir les opérateurs C++ d'allocation dynamique de mémoire pour l'embarqué
- Rendre les objets C++ persistants flashables et romables
- Gérer les exceptions en C++ pour sécuriser les applications embarquées
- Utiliser des objets C++ pour gérer la transmission/réception série de chaînes de caractères

Les travaux pratiques sont effectués sur une carte ARM émulée par QEMU

### Pré-requis

- Compétences en programmation C (voir notre cours [oL2 - Langage C pour les MCUs embarqués](#))

### 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.
  - Un PC Linux en ligne par stagiaire pour les activités pratiques
  - Exemples de code, exercices et solutions
  - 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 les activités pratiques après le cours
- Au début de chaque demi-journée 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

### Durée

- Totale : 18 heures
- 3 sessions de 6 heures chacune (hors temps de pause)
- De 40% à 50% du temps de formation est consacré aux activités pratiques
- Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante

# Course Outline

## Première session

### Introduction to C++ for industrial systems

- Introduction to object oriented programming
- History and definition
- Overview on C++98/C++03/C++11/C++14/C++17/C++20
- Modern C++ objectives
- Switch from C to C++
- Embedded C++
- How to write optimized embedded code

**Exercise :** Understand function mangling

**Exercise :** Function inlining

**Exercise :** Volatile variable handling

### C++ and embedded systems

- Object Oriented Programming in C++
  - Encapsulation
  - Classes and objects
  - Attributes and member functions
  - Object construction and destruction
  - Construction parameters
  - Copy constructor
  - Object composition and container
  - Scope qualifier operator

**Exercise :** Declaring classes and methods

**Exercise :** Working with default, copy and parameterized constructors

**Exercise :** Understand the differences between composition and aggregation

## Deuxième session

### Operator Overloading

- Optimizing parameter object passing
- Overloading operators by member functions
- Overloading operators by friend functions
- Memory management operators overloading

**Exercise :** The assignment operator

**Exercise :** overloading operators

### Simple Inheritance

- Specialization by addition and substitution
- Derivation and access rules
- Construction during inheritance
- Inheritance polymorphism
- Virtual methods

**Exercise :** Understand inheritance

## Persistent and flashable objects

- Constant and partially constant objects
- Persistent objects
- Flashable objects

**Exercise :** Creating constant, mutable, persistent and ROMable objects

## Enhancing security with exceptions

- Launching, capturing and handling exceptions
- Retriggering exception
- Exceptions specifications
- Handling unexpected exception
- Exception objects of the C++ standard library

**Exercise :** Handle errors using exceptions

**Exercise :** Unexpected exceptions management

## Troisième session

### C++ advanced techniques

- Member pointers
- Generic objects and templates
  - Classes and generic functions
  - Templates overloading
  - Specializing templates
  - STL (Standard Template Library)
  - Templates in embedded systems
- Polymorphic objects
- Virtual objects and abstract classes
- Specializing objects by simple inheritance
  - Building derivate objects
  - Access control rules for inherited objects
  - Specializing objects by multiple inheritance
  - Conflicts resolution by scope operator
  - Virtual inheritance

**Exercise :** Generic classes and functions

**Exercise :** Understand virtual methods by subclassing a generic Device class

**Exercise :** Understand multiple inheritance and virtual bases