+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 20/01/2026

S'inscrire

Formation Rust Embarqué avec Kernel Linux

3 jours (21 heures)

Présentation

Rust est un langage système moderne qui apporte des garanties fortes en matière de sécurité mémoire, de fiabilité et de performances, particulièrement adaptées aux systèmes embarqués industriels. Dans des environnements où la robustesse et la maîtrise des ressources sont critiques, Rust s'impose progressivement comme une alternative crédible au C et au C++ pour le développement de composants système et applicatifs embarqués.

Cette formation Rust embarqué vous permettra de comprendre comment utiliser Rust dans des architectures embarquées modernes, depuis l'interaction avec le matériel jusqu'au développement de composants système robustes.

Vous apprendrez à compiler, déployer et exécuter des applications Rust sur des plateformes embarquées, à dialoguer avec les périphériques et à structurer des applications fiables et maintenables.

L'approche est résolument pratique et orientée production, avec un accent mis sur les usages industriels réels : communication matérielle, fiabilité, performances et exploitation. À l'issue de cette formation, vous serez en mesure d'intégrer Rust dans vos projets embarqués et de concevoir des solutions robustes adaptées aux contraintes terrain.

Comme toutes nos formations, celle-ci s'appuie sur la dernière version stable de Rust.

Objectifs

- Utiliser Rust dans des systèmes embarqués industriels
- Développer des composants fiables et performants
- Interagir avec le matériel et les périphériques
- Structurer et déployer des applications embarquées robustes

Public visé

- Ingénieurs systèmes embarqués
- Développeurs firmware et Linux embarqué
- Développeurs C/C++ souhaitant utiliser Rust

Pré-requis

- Maîtrise du C ou C++
- Notions de systèmes embarqués
- Connaissances de base Linux

Formation Rust embarqué avec Kernel Linux

[Jours 1 - Matin]

Environnements embarqués modernes et positionnement de Rust

- Panorama des systèmes embarqués industriels actuels
- Différences entre firmware, Linux embarqué et applicatif
- Positionnement de Rust face au C/C++
- Cas d’usage industriels : drivers, services système, IoT
- Contraintes matérielles : CPU, mémoire, performances
- Atelier pratique : Préparation de l’environnement Rust embarqué.

[Jours 1 - Après-midi]

Rust embarqué : toolchain et compilation croisée

- Comprendre la cross-compilation
- Cibles Rust pour systèmes embarqués
- Gestion des dépendances avec Cargo
- Différences std / no_std
- Organisation d’un projet Rust embarqué
- Atelier pratique : Compiler et déployer un binaire embarqué.

Architecture système et interaction avec le matériel

- Architecture matérielle des systèmes embarqués
- Accès aux périphériques et mémoire
- Notions de drivers et d’abstraction matérielle
- Interaction entre Rust et le système
- Gestion des erreurs et robustesse

- Atelier pratique : Premier accès matériel depuis Rust.

[Jours 2 - Matin]

Rust côté système et espace utilisateur

- Rôle de Rust dans un système embarqué
- Développement Rust en userland
- Interaction avec le noyau et les périphériques
- Appels système et accès bas niveau
- Sécurité mémoire et fiabilité
- Atelier pratique : Application Rust dialoguant avec un périphérique.

Drivers et communication matériel

- Principes des drivers embarqués
- Bus matériels : SPI, I2C, UART
- Accès aux périphériques depuis Rust
- Gestion des interruptions et événements
- Gestion des erreurs matérielles
- Atelier pratique : Développer une application Rust qui communique avec un périphérique matériel.

[Jours 2 - Après-midi]

Performance, sécurité et fiabilité

- Performances Rust en environnement embarqué
- Gestion de la mémoire et des ressources
- Sécurité mémoire appliquée aux systèmes
- Isolation, permissions et accès matériel
- Bonnes pratiques industrielles
- Atelier pratique : Analyse et optimisation d’un composant Rust embarqué.

Rust et Kernel Linux embarqué

- Architecture du kernel Linux en environnement embarqué
- Rôle du noyau et des drivers dans un système embarqué
- État actuel du support de Rust dans le kernel Linux
- Interaction entre kernel space et user space
- Cas d’usage industriels et limites actuelles

- Atelier pratique : Charger et utiliser un module kernel simple interagissant avec une application Rust

[Jours 3 - Matin]

Structuration et architecture logicielle embarquée

- Organisation modulaire des composants Rust
- Séparation logique système / matériel
- Gestion des dépendances et des versions
- Patterns d’architecture embarquée
- Maintenabilité et évolutivité
- Atelier pratique : Structuration d’une application embarquée complète.

[Jours 3 - Après-midi]

Debug, tests et validation sur cible

- Debug embarqué et diagnostic système
- Logs, traces et instrumentation
- Tests unitaires et tests système
- Validation sur matériel réel
- Gestion des pannes et anomalies
- Atelier pratique : Débogage et correction sur cible embarquée.

Déploiement, exploitation et cycle de vie

- Déploiement et mise à jour des composants embarqués
- Gestion des versions et compatibilité
- Surveillance et exploitation en production
- Bonnes pratiques industrielles
- Limites et perspectives du Rust embarqué
- Atelier pratique : Déploiement final et validation complète.

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d’auto-évaluation nous permettant d’apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs

personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.