

Mis à jour le 13/10/2026

S'inscrire

Formation Développement embarqué assisté par l'IA

3 jours (21 heures)

Présentation

Le développement embarqué assisté par l'IA associe les agents de codage aux exigences du micrologiciel et du matériel cible. Une hypothèse plausible peut produire un code compilable, mais incorrect sur la carte. Une démarche fondée sur les preuves techniques permet de contrôler les propositions, de vérifier les comportements et de justifier les choix.

Notre formation Développement embarqué assisté par l'IA vous permettra d'intégrer les assistants à votre environnement de travail sans renoncer à la rigueur technique ni à la responsabilité de validation.

Vous apprendrez à formaliser des spécifications fonctionnelles vérifiables, à préparer le contexte matériel et à construire des prompts adaptés. Vous exploiterez les fiches techniques, les fichiers SVD et les manuels de référence pour encadrer la génération du micrologiciel.

À l'issue de la formation, vous serez en mesure de configurer des agents dans votre IDE et votre terminal, de mobiliser le Model Context Protocol et de réutiliser des compétences spécialisées pour orchestrer vos outils.

Cette formation aborde également les tests sur cible, l'analyse statique, la sécurité des agents et les audits. Vous constituerez un dossier de validation reliant exigences, modifications et résultats mesurés afin de conserver un micrologiciel contrôlé, testé et justifiable.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Structurer une démarche de développement embarqué assisté par l'IA fondée sur des preuves vérifiables
- Rédiger des spécifications fonctionnelles et des prompts intégrant les contraintes matérielles
- Configurer des assistants IDE et CLI avec un contexte projet maîtrisé
- Concevoir des agents spécialisés combinant MCP, compétences réutilisables et orchestration des outils
- Valider le micrologiciel généré par des analyses statiques, des tests sur hôte et des essais sur cible
- Auditer les permissions des agents et documenter la traçabilité des décisions techniques

Public visé

- Développeur embarqué
- Ingénieur firmware
- Ingénieur systèmes embarqués
- Ingénieur validation embarquée
- Architecte logiciel embarqué

Pré-requis

- Expérience de programmation embarquée en C ou C++
- Connaissance des microcontrôleurs, des registres, des interruptions et des périphériques
- Pratique de la compilation croisée, du débogage et du chargement de micrologiciels sur une carte
- Maîtrise des opérations courantes de versionnement avec Git
- Capacité à lire une documentation technique en anglais et à écrire des tests simples

Pré-requis techniques

- Ordinateur sous Windows 11 ou Ubuntu 24.04 LTS, avec processeur quatre cœurs, 16 Go de RAM recommandés et ports USB disponibles, sans carte graphique dédiée requise pour les assistants hébergés
- Installation de Visual Studio Code, Git et d'un assistant compatible IDE et CLI, dans les versions validées pour la session
- Environnement STM32CubeIDE avec compilateur croisé et débogueur opérationnels, selon la configuration communiquée avant la formation
- Carte STM32 Nucleo avec sonde ST-LINK et câble USB de données, référence matérielle commune précisée avant la session
- Connexion Internet stable avec accès autorisé aux services nécessaires, compte GitHub Copilot disposant des fonctionnalités utilisées et droits locaux pour installer les outils approuvés

Programme de notre formation Développement embarqué assisté par l'IA

[Jour 1 - Matin]

Assistants IA et rigueur embarquée

- Introduction à la formation et au projet fil rouge, un micrologiciel de pilotage de périphérique sur carte de développement
- Panorama des outils IA pour ingénieurs embarqués, assistants conversationnels, complétion, agents IDE et agents CLI
- Différences entre code compilable, conformité aux exigences et comportement correct sur le matériel cible
- Principes du prompting embarqué, contraintes mémoire, temporisation, interruptions et critères de réussite
- Méthodologie fondée sur les preuves, séparation des faits documentés, des hypothèses et des vérifications nécessaires
- Atelier pratique : Comparer deux propositions de code pour un périphérique, repérer les hypothèses matérielles et reformuler un prompt exigeant des preuves vérifiables.

[Jour 1 - Après-midi]

Spécifications et contexte matériel

- Développement piloté par les spécifications, comportements attendus, interfaces, limites et critères d'acceptation
- Rédaction d'un document de spécification fonctionnelle, états, erreurs, délais et exigences observables
- Exploitation des fiches techniques, manuels de référence, schémas de carte et errata du constructeur
- Lecture des fichiers CMSIS-SVD, registres, champs de bits et limites de leur description du comportement matériel
- Construction du contexte projet, référence exacte du composant, révisions documentaires, code existant et registre des hypothèses
- Atelier pratique : Rédiger la spécification du périphérique du projet fil rouge et constituer un dossier de contexte reliant chaque exigence à sa documentation matérielle.

[Jour 2 - Matin]

Assistants dans l'IDE et le terminal

- Configuration des assistants IDE et CLI, périmètre du dépôt, accès aux fichiers et approbation des commandes
- Instructions projet réutilisables, conventions de code, architecture, commandes de compilation et règles de validation
- Décomposition d'une demande en plan de modifications, tâches limitées et points de contrôle humains
- Génération et revue d'un pilote embarqué, initialisation, registres, types, accès mémoire et gestion des erreurs
- Exécution contrôlée des outils, lecture des différences Git et collecte des journaux de compilation
- Atelier pratique : Configurer un assistant dans l'IDE et le terminal, générer une modification limitée du pilote et vérifier ses changements avant compilation.

[Jour 2 - Après-midi]

Agents spécialisés et orchestration MCP

- Architecture du Model Context Protocol, hôte, clients, serveurs, ressources, outils et prompts
- Connexion contrôlée aux ressources du projet et aux outils de compilation ou de test, avec permissions minimales
- Conception d'un agent embarqué spécialisé, rôle, entrées, outils autorisés, livrables et conditions d'arrêt
- Création de skills réutilisables, instructions, scripts déterministes et modèles de compte rendu
- Orchestration multi-outils, enchaînement génération, compilation et tests, gestion des échecs et validation humaine avant chargement sur cible
- Atelier pratique : Configurer un agent spécialisé avec un serveur MCP local préparé et une skill de validation, puis vérifier ses permissions et ses comptes rendus sur le projet fil rouge.

[Jour 3 - Matin]

Validation du micrologiciel généré

- Construction d'une matrice exigences et tests, critères mesurables et indépendance des résultats vis-à-vis des affirmations de l'agent
- Analyse statique et diagnostics du compilateur, conversions, débordements, accès mémoire et comportements indéfinis
- Tests unitaires sur hôte, doublures de périphériques, cas limites et limites de la simulation
- Tests sur cible, débogage, inspection des registres, traces et mesure des temporisations
- Recherche des écarts matériels, volatile, atomicité, concurrence avec les interruptions et consommation mémoire
- Atelier pratique : Diagnostiquer un micrologiciel qui compile mais échoue sur la carte, corriger la cause et conserver les résultats des tests de non-régression.

[Jour 3 - Après-midi]

Sécurité et dossier de preuves

- Sécurité des agents IA, injections de prompt dans les documents, outils non fiables et commandes dangereuses
- Protection du code confidentiel, des secrets et des données projet, sélection des contextes transmissibles
- Audit des permissions, serveurs MCP, dépendances et configurations, isolation des outils et maîtrise des accès matériels
- Constitution d'un dossier de preuves, versions, exigences, sources matérielles, modifications et résultats reproductibles
- Critères d'acceptation et revue humaine finale, couverture des exigences, limites connues et décision de livraison
- Atelier pratique : Auditer le workflow du projet fil rouge, tester ses garde-fous et présenter un dossier justifiant l'acceptation ou le rejet du micrologiciel généré.

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.