

# Formation : Yocto, développer son propre système embarqué sous Linux

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

Prix : 2550 € H.T.

★★★★☆ 4,6 / 5

L'environnement Yocto permet de développer des systèmes embarqués fiables et robustes en isolant le développement applicatif métier des couches basses liées au matériel. Cette formation vous permettra de maîtriser l'utilisation courante de Yocto, de générer et d'installer des images Linux aux contenus personnalisés.

## Objectifs pédagogiques

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

- ✓ Connaître le rôle des différents éléments de Yocto Project
- ✓ Savoir générer une image Linux embarqué standard pour une carte cible
- ✓ Savoir ajouter des packages et personnaliser le contenu d'une image Linux
- ✓ Savoir utiliser le SDK de Yocto pour développer du code applicatif personnalisé

## Public concerné

Architecte ou ingénieur devant porter Linux sur un nouveau matériel ou étendre un système embarqué.

## Prérequis

Avoir des notions de programmation. Connaissance de Linux (niveau utilisateur).

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

## Méthodes et moyens pédagogiques

### Travaux pratiques

Les travaux pratiques sont réalisés sur des serveurs Linux et sur des cartes Raspberry Pi 4 que vous pourrez emporter à l'issue de cette formation.

### PARTICIPANTS

Architecte ou ingénieur devant porter Linux sur un nouveau matériel ou étendre un système embarqué.

### PRÉREQUIS

Avoir des notions de programmation. Connaissance de Linux (niveau utilisateur).

### 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.

## 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 Créer un système Linux embarqué

- Environnement Linux embarqué : concepts, composants, outils de génération, Buildroot et Yocto Project.
- Production d'une image standard : environnement de travail, Poky, configuration, layers spécifiques, outil BitBake...
- Composition d'un système Linux embarqué : matériel, bootloader, noyau Linux, processus init, scripts de démarrage.

#### Travaux pratiques

Installation de Poky et meta-raspberrypi, production d'une image standard.

### 2 Personnalisation du système embarqué

- Découverte et analyse du système : connexion, systèmes de fichiers, arborescence standard, boot.
- Personnalisation de l'image : fichier .bb, syntaxe des recettes, création d'une recette d'image, users et mots de passe.
- Ajout de packages : packages standards de Poky, fournis par OpenEmbedded, configuration de BusyBox.

#### Travaux pratiques

Production d'une image complète et personnalisée pour Raspberry Pi 4 avec des packages fournis par Poky et meta-openembedded.

### 3 Configuration avancée du système

- Extension de recettes : fichiers .bbappend, surcharge de fichiers de recettes, configuration réseau statique.
- Création et application de patches : sur un fichier de donnée d'une recette, sur un fichier source à compiler.
- Noyau Linux et Device Tree : configuration du noyau, création de patch sur le noyau, sur le Device Tree.

#### Travaux pratiques

Extensions et patches pour base-files, init-ifupdown, nano, BusyBox, kernel...

#### 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 Développement du code métier

- Installation du SDK : production, extraction et installation du SDK de Yocto.
- Compilation du code applicatif : examen de la toolchain, scripts d'initialisation, cross-compilation de code personnel.
- Débogage et mise au point : débogage distant avec gdbserver, outils de mise au point (gprof, valgrind, gcov...).
- Intégration du code métier dans l'image : recettes, lancement au démarrage.

#### Travaux pratiques

Écriture, débogage et intégration d'un code métier dans l'image.

### Dates et lieux

#### CLASSE À DISTANCE

2026 : 16 juin, 29 sep., 8 déc.

#### PARIS LA DÉFENSE

2026 : 16 juin, 29 sep., 8 déc.