

*L'utilisation croissante de Linux dans les systèmes embarqués va de pair avec un besoin de fiabilité et de pérennité de l'installation. Pour cela **Yocto** est aujourd'hui le système de construction le plus adapté aux nécessités de l'environnement industriel. Héritier d'outils réputés comme Buildroot ou PXTdist, il offre une richesse incomparable en terme d'applications disponibles et des possibilités inégalées pour garantir la pérennité et la portabilité de l'environnement Linux embarqué produit.*

Ce cours vous propose de découvrir les rouages de Yocto, d'en maîtriser l'utilisation courante (génération et installation d'images Linux embarqué) afin de développer des applications pour Linux embarqué.

Les travaux pratiques pour ce cours se déroulent sur des cartes « Raspberry Pi 3 » ou « Beaglebone Black » (une carte par participant) mises à disposition par Logilin.

Organisation

Durée : 4 jours (28 heures).

Pré-requis : Connaissance de Linux (utilisateur), notions de programmation.

Thèmes abordés

- **Linux industriel et Yocto** : système Linux, environnement embarqué, industrialisation de la production.
- **Génération et personnalisation du BSP** : production d'une image standard, découverte et analyse du système, personnalisation de l'image.
- **Développement du code métier** : installation du SDK, compilation du code applicatif, débogage et mise au point, intégration du code métier dans l'image.

.../...

Plan détaillé

I – Linux industriel et Yocto

Le système Linux

Structure du système Linux, versions du noyau, licences libres et implications en environnement industriel, drivers spécifiques.

Environnement embarqué

Caractéristiques et composants spécifiques, outils de développement, chaîne de *cross-compilation*.

Industrialisation de la production

Recherche de pérennité et reproductibilité, nécessité d'un *Build system*, avantages et inconvénients de Buildroot et Yocto.

Travaux pratiques

Téléchargement et installation des packages nécessaires pour utiliser Yocto.

II – Génération et personnalisation du BSP

Production d'une image standard

Configuration, aperçu de Bitbake, compilation, installation de l'image sur la cible.

Découverte et analyse du système

Connexion, système de fichiers, arborescence standard, boot.

Personnalisation de l'image

Administration, configuration du noyau, de Busybox, ajout de *packages*, de *layers*.

Travaux pratiques

Production d'une image complète et personnalisée pour processeur ARM.

III – 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 d'aide à la mise au point (Gprof, Gcov, Valgrind...).

Intégration du code métier dans l'image

Recettes, lancement au démarrage.

Travaux pratiques

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

Conclusion

Discussions libres sur l'ensemble des thèmes abordés.

Travaux pratiques

Expérimentations libres suivant les demandes des participants.