Logilin formations

Yocto Project avancé

L'outil Yocto Project permet d'obtenir des systèmes Linux embarqués robustes et fiables. Toutefois la structure même de Yocto Project en fait un outil complexe à maîtriser.

Cette formation est destinée à des personnes ayant déjà une première expérience avec Yocto, et souhaitant approfondir leurs connaissances en explorant des aspects avancés de cette puissante plateforme.

Si vous souhaitez une formation de découverte et prise en main de *Yocto, nous vous conseillerons plutôt notre cours « Linux embarqué avec Yocto Project », qui développe les concepts de base considérés ici comme acquis.*

Organisation

Audience

Nous limitons habituellement le nombre de stagiaires dans nos sessions à 4 personnes au maximum pour garantir des échanges fluides et conviviaux.

Les sessions à distance se déroulent sur **plateforme Zoom**. Le seul matériel nécessaire est un ordinateur avec une connexion Internet et un micro. Nous conseillons un ensemble casque + micro pour limiter le bruit de fond. Nous suggérons également l'emploi d'une webcam si l'environnement le permet.

Pré-requis

Connaissance basique de Yocto Project (savoir créer une image standard, connaître les notions de layer, de recettes, d'image, savoir utiliser bitbake).

Durée

3 jours (21 heures)

Travaux pratiques

Les exercices se déroulent sur des PC Linux accessibles à distance (connexion SSH / PuTTY / Tera Term) et émulateur Qemu. Les démonstrations sont présentées sur cartes Raspberry Pi 4.

Thèmes abordés

Rappels sur Yocto Project : *build systems*, arborescence de travail, commandes principales.

Utilisation de *devtool* : recherche d'information, création et maintenance de recette, de *patch*.

Approfondissement des recettes : classes, *distro*, *overrides,* démarrage au *boot*.

Support matériel et industrialisation : *bootloader*, noyau, *device tree*, obligations légales.

Plan détaillé

I – Rappels sur Yocto Project

Linux embarqué et *build systems*

Concepts, comparatif avec Buildroot.

Arborescence de travail recommandé

Fichiers *site.conf*, emplacement des layers, des builds, des répertoires *downloads* et *shared-state cache*.

Rappels des commandes principales

bitbake, *bitbake-layers*, *devtool*, *recipetool*, *runqemu*. Contenu typique d'un layer.

Travaux pratiques : production d'image customisée sur un layer personnel.

II – Utilisation de Devtool

Principes

Recherche de recettes, d'emplacements, de versions, d'extensions de recettes. Layer *workspace*.

Création de recette

Création automatique, mise à jour de recette. Test, édition, validation ou abandon de recette.

Création de patches

Utilisation de *devtool* pour création de *patch*, plusieurs *patches* successifs.

Travaux pratiques : mise à jour de recette, création de recettes pour *Cmake* et *Makefile*. création de *patches* sur *busybox* et le *kernel* Linux.

III – Approfondissement des recettes

Variables

Variables globales et contextuelles, tableaux, conditions, *overrides*.

Classes

Fichiers de classes, héritage, utilisations, classes intéressantes.

Distro

Rôle et choix de *distro*, utilisation des *features*, choix de la *libC* et du système d'*init*.

Exécution au démarrage

Démarrage automatique avec SysVinit et avec Systemd. Projet EarlyInit

Travaux pratiques : création de classe et héritage, création de distro, configuration des *overrides* d'une *distro*, configuration d'option *packageconfig* de *nano*, démarrage automatique au *boot.*

IV – Support du matériel et industrialisation

Machine

Fichier de configuration, variables concernées, types d'image, fichiers *WKS*, partitions.

Bootloader et kernel

Choix de version, configuration, *fragments* et *defconfig*, principe du *device tree*.

Partitionnement

Utilisation des fichiers WKS et WIC.

Obligations légales

Licences, manifeste, *Software Bill of Material*, conformité au *Cyber Resilience Act*.

Travaux pratiques : configuration d'une machine personnalisée, *patch* sur *U-boot* pour ajouter des variables personnalisées, config *kernel* et intégration dans le fichier *machine, device tree* customisé.

Conclusion

Discussions libres sur l'ensemble des thèmes abordés.

Travaux pratiques : expérimentations libres suivant les demandes des stagiaires.

MàJ 05/10/2024