

# Yocto Project avancé

**Christophe BLAESS**

[christophe.blaess@logilin.fr](mailto:christophe.blaess@logilin.fr)  
<https://www.blaess.fr/christophe/>



*Ingénierie et formations sur Linux et les logiciels libres*  
<https://www.logilin.fr>

## Avant-propos

Ce support de formation est distribué sous licence **Creative Commons 4.0**



*(Attribution - Partage dans les mêmes conditions).*

Vous êtes libres de copier et partager ce document, en mentionnant son origine. Si vous l'intégrez dans un contenu plus vaste, ce dernier devra être distribué avec les mêmes droits.

Ce cours a été rédigé en utilisant des logiciels libres sur système d'exploitation Linux :

- *LibreOffice Writer* pour le support et la mise en page
- *LibreOffice Draw* pour les dessins vectoriels
- *Gimp* pour les images bitmap
- *Excalidraw* (en ligne) pour certains schémas

« Linux » est un nom déposé par Linus Torvalds et administré par le « Linux Mark Institute ».

Le logo du pingouin « Tux » a été créé par Larry Ewing.

Yocto Project avancé

YPA v. 1.2

<https://www.blaess.fr/christophe/>

<https://www.logilin.fr>

# Introduction

Ce cours est un approfondissement de l'utilisation de **Yocto Project** pour produire un système Linux embarqué et développer le code applicatif.

On considère comme acquis l'utilisation basique de Yocto.

Les démonstrations et **travaux pratiques** se dérouleront sur émulateur Qemu pour les trois premiers chapitres et sur une carte ARM simple (comme le *Raspberry Pi*) pour le chapitre 4.

Le **support de cours** en version PDF est disponible ici :

<https://www.logilin.fr/files/support-YPA.zip>



# Plan de la formation

## I – Rappels sur Yocto Project

- Linux embarqué et *build systems* : concepts, comparatif avec Buildroot.
- Arborescence recommandé : *site.conf*, *layers*, *builds*, *downloads*, *sstate cache*.
- Rappels des commandes : *bitbake*, *bitbake-layer*, *runqemu*, etc.
- Contenu d'un layer typique. Travaux pratiques.

## II – Utilisation de devtool

- Principes, création automatique d'une recette, mise à jour d'une recette existante.
- Tests, édition et validation ou abandon d'une recette en cours de modification.
- Création de patch sur les sources d'un package. Travaux pratiques.

## III – Approfondissement du contenu des recettes

- Variables globales et contextuelles, tableaux, conditions, *overrides*, travaux pratiques
- Classes : fichiers, héritage, utilisations, classes intéressantes, travaux pratiques.
- *Distro* : Rôle, choix de libC et du système d'initialisation, travaux pratiques
- Exécution au démarrage : avec *SysVinit*, avec *Systemd*, travaux pratiques

## IV – Support du matériel et industrialisations

- Machine : fichier de configuration, travaux pratiques.
- *Bootloader*, *kernel* et *device tree* : versions, configuration, travaux pratiques
- Partitionnement : fichiers *WKS*, travaux pratiques.
- Obligations : licences, manifestes, *Software Bill of Materials*, *Cyber Resilience Act*.