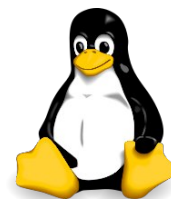


# Formation Pratique

## Yocto pour construire les Systèmes Embarqués



- Présentation** Pour construire un Système Embarqué fonctionnant sous un *Operating System* Linux, il est nécessaire de définir et d'assembler une « distribution » Linux particularisée (noyau et packages) adaptée aussi bien au Hardware qu'aux applications destinées à fonctionner sur ce système.  
Dans l'absolu, cette tâche est d'une extrême complexité considérant les dizaines de milliers de fichiers et les millions de lignes de code que représente aujourd'hui Linux et son environnement.  
C'est pourquoi des méthodologies et des outils ont été développés pour rendre cette tâche accessible, si ce n'est simple.  
**Yocto** est un projet (*Open Source*) qui répond de façon très exhaustive à ces besoins.  
NB : *Buildroot* est une alternative beaucoup moins ambitieuse et plus légère : si vous hésitez entre les deux, n'hésitez pas à consulter notre Expert qui saura vous conseiller.
- Objectifs** Cette **formation de trois jours** apprend concrètement à **maîtriser Yocto** et à **construire des distributions complètes** (toolchains, bootloaders, noyaux, packages...).
- Pour qui ?** Cette formation est **fondamentale** aussi bien pour les **Chefs de Projets** que pour les **Ingénieurs de Développement**.
- Bénéfices** Yocto permet à court terme de **construire et déployer** un projet dans les meilleurs délais. A plus long terme, Yocto permet d'en assurer la **maintenance et les évolutions**.
- Pédagogie** Ce cours s'inscrit dans le cadre d'une action d'acquisition ou de perfectionnement des connaissances. Les **Exercices Pratiques**, qui occupent environ **50 % du temps de la formation** sont des éléments clés du succès de ces formations, et sont essentiel pour l'apprentissage. Ils **assurent** également un **contrôle continu** de l'acquisition des compétences.
- Instructeurs** Les **Instructeurs ALSE** sont aussi et surtout des **Experts** qui utilisent à journée entière les Langages et les Méthodes de Conception qu'ils enseignent, pour concevoir et vérifier des systèmes complexes et performants. Ils savent partager leur savoir-faire avec passion et sont particulièrement appréciés des participants.
- Pré-requis** La participation à cette formation exige une **connaissance solide** de **Linux Embarqué** comme celle qui est enseignée dans notre formation « **Linux Embarqué** ».
- Durée** **Trois (3) jours**, soit **21 heures effectives** de cours.  
Typiquement 9h30 → 18H, le premier jour et 9H → 17h30 les jours suivants.
- Lieu** Dans les locaux d'ALSE, **Paris XIII<sup>ème</sup>**
- Matériel** Pour les cours publics, le matériel (station de travail) est fourni pour la durée du training.

NB : à partir de quatre ou cinq stagiaires, une formation sur site est envisageable.

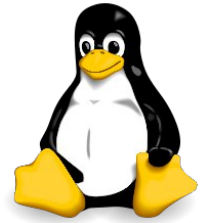
ALSE est un Organisme de Formation Professionnelle Continue déclaré auprès de la DIRECCTE depuis 1996 sous le numéro 26.21.01281.21 et enregistrée dans le Datadock. Cette formation peut donc être prise en charge dans ce cadre.



# Programme de la Formation

## Yocto pour construire les Systèmes Embarqués

(3 jours)



### Yocto Kick-start

- What is Yocto?
- Host setup & system requirements
- Downloading Yocto
- Embedded Linux components
- Using Yocto to build U-Boot, a Linux kernel and a FileSystem
- Booting a Linux board

### Yocto Build System

- Bitbake and OpenEmbedded
- Metadata file types
- User configuration files
- Layers, recipes and classes
- Using Hob to build an image

### Yocto Outputs

- A closer look at the deploy directory
- The sysroot directories
- The working directories
- The Yocto SDKs
- The package feeds

### Manage Custom Layers

- Appending packages to a filesystem image
- Managing new layers using Git
- Image and package recipes
- Using layers to modify recipes
- Creating a new distribution

### Package Management

- Linux package management tools
- Package updates through HTTP
- Package release versioning
- Debug and release packages (split)

.../...

## **The BitBake Tool**

- The global datastore
- Conditional variable assignment
- Functions and tasks

## **Build System Trouble Shooting**

- BitBake log files
- QA error messages
- Using the BuildHistory
- Changing recipe build features

## **Development Workflow**

- Choosing a toolchain mode (TCMODE)
- Developing inside the Yocto build system
- Using a pre-built SDK
- Development and debug image features
- TFTP boot and NFS root filesystems

## **Development Tools**

- Building projects using Makefile, Autotools, Cmake or Qmake
- Debugging an application from the command line (gdbserver)
- Using the Eclipse remote tools (TCF)
- Systems and application profiling tools
- Re-building U-Boot
- Re-building the Linux kernel
- Automated tests

## **Application Integration**

- Creating a new application package layer
- Managing package dependencies
- Masking recipes
- Choosing Yocto graphics back-ends
- Managing licence files

## **Systems Integration**

- Creating a new BSP layer
- Patching existing kernel/U-boot code
- Linux-yocto compatible kernels
- Kernel out-of tree module recipes
- Managing multiple board configurations

## **Additional Yocto Layers**

- Finding new Yocto compatible layers
- Exploring meta-browsers  
Exploring meta-security

--oOo--