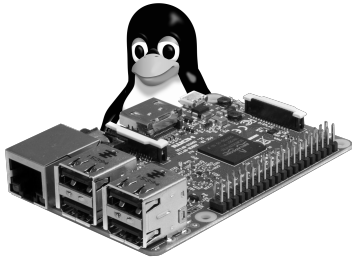


## Linux embarqué avec Buildroot



*L'utilisation de Linux dans les environnements embarqués sur des systèmes restreints nécessite une bonne connaissance des mécanismes sous-jacents, de la configuration du noyau, de l'organisation des bibliothèques et de la mise en œuvre des services et applications utilisateurs.*

*Il existe aujourd'hui deux approches pour construire un système embarqué : l'utilisation de Buildroot ou celle de Yocto Project. Cette formation est orientée autour de Buildroot, plus simple à maîtriser que Yocto.*

*La progression pédagogique est organisée autour d'une construction progressive d'une solution Linux embarqué pour carte à processeur ARM.*

### 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 de Linux (niveau utilisateur).

#### Durée

3 jours (21 heures)

#### Travaux pratiques

Sessions à distance : 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

---

**Création d'un système Linux embarqué** : outils de développement, *cross-compilation*, utilisation de Buildroot.

**Personnalisation d'un système embarqué** : ajustement du système, des services, du réseau, paramétrage du noyau.

**Développement du code métier** : chaîne de compilation, débogage à distance, optimisation.

## Plan détaillé

---

### I - Créer un système Linux embarqué

#### Linux embarqué

Spécificités du développement embarqué, architectures, cross-compilation, build systems, utilisation de Buildroot.

**Travaux pratiques** : création d'un système embarqué standard avec Buildroot

#### Composition d'un système Linux embarqué

Aspects matériels, composants logiciels, noyau, drivers, espace utilisateur.

#### Boot du système

Firmware, bootloader, kernel, processus init, partitionnement, démarrage des services du système.

**Travaux pratiques** : Démarrage de l'émulateur avec le système créé.

### II - Personnalisation d'un système embarqué

#### Découverte du système Linux embarqué

Arborescence standard

**Travaux pratiques** : exploration de l'arborescence des fichiers, éléments produits par Buildroot.

#### Configuration du système

Nom d'hôte et message de connexion, répertoire d'overlay, protection du système de fichiers, scripts de démarrage

**Travaux pratiques** : configuration des utilisateurs, ajout de scripts de remontage du rootfs

#### Ajout de commandes et d'applications

Busybox, applications de Buildroot

**Travaux pratiques** : configuration de Busybox, ajout de package de Buildroot

#### Configuration du réseau

Configuration statique ou utilisation de DHCP, Services SSH, HTTP, NTP.

**Travaux pratiques** : mise en œuvre d'un serveur SSH

#### Noyau Linux

Versions, configuration, licence, drivers

**Travaux pratiques** : Configuration du noyau Linux

### III - Développement du code métier

#### Outils de développement

Chaîne de compilation croisée, extraction et installation. Utilisation des makefiles,

**Travaux pratiques** : utilisation de la toolchain pour compiler du code métier.

#### Débogage distant

Utilisation de GDB et GDBserver

**Travaux pratiques** : débogage à distance avec GDBserver

#### Optimisation et mise au point

Outils gcov, gprof, strace, ltrace, valgrind...

**Travaux pratiques** : utilisation de Valgrind pour vérifier du code métier

### Conclusion

Discussions libres sur l'ensemble des thèmes abordés.

**Travaux pratiques** : Expérimentations libres suivant les demandes des stagiaires.