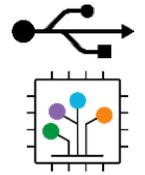
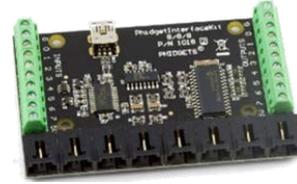


# Systèmes Embarqués Avancé et Logiciels



devicetree  
.org

**Responsables : Stéphane Lavirotte**

**2 ECTS – Seconde période Semestre 9**

## Prérequis :

- Programmation C
- Système d'exploitation Unix
- Pas d'assembleur

## Résumé :

Ce module a pour but de comprendre comment étendre les fonctionnalités offertes par le noyau Linux, en particulier via l'écriture de pilote de périphérique (driver) dans un système d'exploitation pouvant être embarqué. Après une présentation de l'architecture du noyau, le cours mettra tout d'abord l'accent son extension via les modules. L'écriture de modules sera alors appliquée à la mise en œuvre d'un pilote de périphérique purement logiciel pour poursuivre sur l'implémentation d'un pilote pour matériel USB. Ce cours se terminera sur les méthodes pour gérer la variabilité matérielle/logicielle.

Pour rendre les choses concrètes et faire la part belle à l'expérimentation, plus de 2/3 du module sera consacré à la mise en œuvre (travaux dirigés).

## Objectifs :

Le but de ce cours est de présenter l'extension d'un système d'exploitation, pouvant être embarqué, en lui ajoutant de nouvelles fonctionnalités via l'écriture de modules pour prendre en compte de nouveaux périphériques.

*Pour les étudiants IoT-CPS:* comprendre comment étendre un système d'exploitation pour prendre en compte de nouveaux périphériques, avoir des notions sur la communication des périphériques USB.

*Pour les étudiants AL:* étendre une application monolithe (le noyau) de plus de 25 millions de lignes de code. Ce sera l'occasion de vous confronter au plus gros projet sur lequel vous aurez travaillé durant vos études mais aussi de voir une application de la gestion de la variabilité matérielle/logicielle.

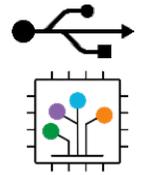
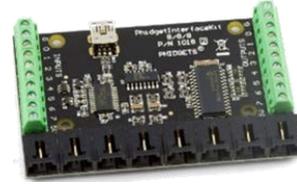
*Pour les étudiants CASPAR:* comprendre les mécanismes de communication des données du périphérique jusqu'aux applications pour mieux investiguer les problèmes de sécurité.

*Pour les étudiants IHM:* Ecrire le(s) pilote(s) de nouveaux dispositifs d'interaction pour les exploiter dans des applications avec une interaction innovante.

## Description du cours et programme :

En vue de doter les étudiants d'une réelle expertise le cours est articulé selon trois axes :

# Systèmes Embarqués Avancé et Logiciels



devicetree  
.org

- **Connaissances**
- **Compétences**

Les séances durent 4h regroupant :

- une heure de cours sur les principaux concepts et éléments techniques concernés par la séance (**Connaissances**)
- trois heures de tutorial dans des environnements logiciels professionnels et sur du matériel grand public (**Compétences**)

## Plan du cours

- Architecture du noyau
- Développement de modules
- Pilotes de périphériques logiciel
- Pilotes de périphériques matériels (matériel USB)
- Prendre en compte de la variabilité matérielle au niveau logiciel (Device Tree)

**Volume horaire** : 32h encadrées et 8h (1h par semaine) de travail personnel non encadré

## Modalités d'Evaluation :

- Contrôle intermédiaire (QCM pour évaluer les connaissances acquises)
- Contrôle terminal (Résolution de problème pour évaluer les compétences acquises).

## Références:

- P. Ficheux, E. Bénard, Linux embarqué. Nouvelle étude de cas - Traite d'OpenEmbedded, Eyrolles, 2012
- Michael Kerrisk, The Linux Programming interface: A Linux and UNIX System Programming Handbook, 2010
- Robert Love, Linux Kernel Development, 2010
- D. Bovet et M. Cesati, Understanding the Linux Kernel (3rd edition), O'Reilly 2006
- J Corbet, A Rubini, G Kroah-Hartman, Linux Device Drivers (3rd Edition), O'Reilly, 2005
- Linux pour l'embarqué: <https://bootlin.com/>

## Acquis

- Compréhension de la communication entre périphérique et logiciel sous Unix. Niveau Expert
- Développement de modules et de pilotes de périphériques. Niveau: Maîtrise
- Découverte de la notion de Device Tree pour la description de plateforme. Niveau: Applications