

## Cours Système Embarqué avec $\mu$ Clinux

### Plan de cours sur mesure

Le cours «  $\mu$ Clinux avec la carte STM32F429I » permet de travailler avec les élèves sur des sujets importants et denses de manière ludique **avec un coût matériel par poste inférieur à 50 euro !**

Il comporte dans les premières séances des passages obligés qui servent à leur offrir rapidement une indépendance pour créer leurs premiers essais sans risquer d'abîmer le matériel.

**On peut ensuite le configurer à la demande avec des sujets connexes à vos objectifs**, autour des systèmes embarqués et des particularités de cette carte.

Certains points demandent des pré-requis qui peuvent être aussi l'objet d'autres cours sur mesure.

Pour chaque partie, **nous privilégions l'utilisation des logiciels libres** (Open Source) qui est la condition *sine qua non* de la maîtrise d'un projet.

### Les passages obligés

#### 1°) Installation de la VM de travail (*durée minimum 2 heures*)

- Création d'une VM Linux simple (sans interface graphique) avec tous les paquets, outils de compilation, sources, qui puisse permettre de travailler ensuite sans liaison Internet,
- Configuration et test de l'ensemble pour la compilation croisée,
- Installation des outils de communication et de gestion de source (avec, entre autres, *git*) et récupération des sources du noyau GNU/Linux, de la *busybox*, de U-Boot, ...

#### 2°) Connexion de la carte STM32 à la VM (*durée minimum 4 heures*)

- Rappels sur l'interface JTAG,
- Rappels sur la connexion série (RS232) et installation des programmes avec lesquels on va se connecter à la carte (kermit, nanocom, picocom, etc),
- L'outil openOCD (Open On-Chip Debugger) utilisé pour flasher via l'interface JTAG,
- Rappels sur l'outil *make* et la gestion des arborescences de code,
- L'organisation des sources du noyau GNU/Linux et ses outils de configuration,
- Définition, rôle et interface de configuration de la *busybox*,
- Compilations et assemblage d'une première version fournie et déjà paramétrée,
- Installation (« *flashage* ») de cette première version sur la carte.

# Les sujets liés aux systèmes embarqués

- Rappels sur les boot-loaders et étude de celui utilisé sur la carte : U-Boot,
- RAMFS, Rootfs & initramfs,
- Les systèmes de gestion de fichiers dédiés pour l'embarqué,
- La configuration et la mise en place d'une busybox sur mesure,
- Le paramétrage et la compilation d'un noyau Linux pour l'embarqué (ici 2.6.33),
- Les outils d'émulation de processeurs utilisés dans l'embarqué (ARM, AVR, Coldfire, ...), afin de simuler des plateformes et/ou des interfaces pour tester du code binaire,
- La programmation dans un système sans MMU avec processus simples,
- Comment faire du multi-tâches dans un système sans MMU<sup>(1)</sup>,
- Outils de debugging offrant des services liés à la problématique « MMU-less »<sup>(1)</sup>,
- Ecriture de code en « *user space* » pour accéder aux composants du « *hardware* » le permettant. Sans rien ajouter avec les Leds de la carte, ou en ajoutant des capteurs (de température ou autre),
- Ecriture de code pour exécuter une tâche dans le noyau Linux du système embarqué et exemples de threads dans le noyau<sup>(2)</sup>,
- Ecriture d'un driver pour le noyau Linux du système embarqué <sup>(2)</sup>. Exemples avec le bus I2C<sup>(3)</sup> ou une pseudo-device,
- et bien d'autres choses, la liste n'est pas exhaustive.

## Quelques idées de sujets pour TP ou projets

- Transformation des cartes STM32 en objets connectés communiquant entre eux ou avec un serveur par ondes radio, en utilisant des modules Wifi (TCP/IP) ou nRF24L01 (liaison SPI).
- Intégration de la carte à un réseau TCP/IP via une liaison série. On propose de créer une interface utilisant un protocole comme SLIP, CSLIP, PPP, un réseau Modbus, ou autres.
- Ajout d'une interface graphique afin d'utiliser l'écran tactile et réalisation d'une application de test affichant les courbes de valeurs issues de capteurs reliés au bus I2C<sup>(3)</sup>.
- Conception d'une télécommande pour serrure connectée avec création d'un protocole sécurisé via Bluetooth, Zigbee, ou autres.
- Sujets autour du gyroscope L3GD20.
- Etc, la liste est très longue.



### POUR TOUTE INFORMATION

commerciale : [secretariat@seriane.org](mailto:secretariat@seriane.org)

technique : [pfoubet@seriane.org](mailto:pfoubet@seriane.org)

<sup>(1)</sup> Ces points n'ont d'intérêt que si les élèves maîtrisent déjà le multi-tâches dans un contexte classique avec MMU.

<sup>(2)</sup> Ces points nécessitent que les élèves aient déjà suivi un cours sur la programmation dans le noyau de Linux et l'écriture de drivers.

<sup>(3)</sup> Il est souhaitable d'avoir un capteur à connecter au bus I2C pour valider les tests.