

# TP4 OS avancés et embarqués

## Familiarisation avec la carte BeagleBone Black

Dans le cadre de ce cours, nous utiliserons une carte BeagleBone Black (<https://beagleboard.org/black>), la grande sœur de la Raspberry Pi.

Cette carte est munie d'un CPU « Sitara » AM335x de Ti (Cortex-A8 à 1 GHz), une interface Ethernet, 512Mo de RAM, 4Go de Flash eMMC ainsi qu'un emplacement pour une carte Micro SD. Le support noyau Linux pour cette carte est proche du « mainline » mais il existe un noyau dédié sur <https://github.com/beagleboard/linux>.

Plusieurs méthodes peuvent permettre d'interagir avec cette carte et d'y installer un système. Nous allons en investiguer 2.

Nous utiliserons une machine virtuelle (VM) pour le développement sur la carte. Vous pouvez soit :

- Installer une VM (VirtualBox) Ubuntu Desktop en téléchargement l'image sur <https://releases.ubuntu.com/focal/>
- [plus simple] Télécharger la première version de la VM sur le lien donné sur moodle (ou sinon ici : O:\share\CSN3A\ENSEIRB\_Ubuntu20\_OS\_1.ova) . Le login pour cette machine est « enseirb » et le mot de passe « sudo » est « enseirb ». Il s'agit d'une VM Ubuntu version 20.04 avec un noyau Linux version 5.15.

---

## GETTING STARTED !

Sur le site du constructeur, on peut trouver des informations afin de pouvoir commencer à utiliser la carte. L'ensemble des informations relatives à cette section ont été inspirées du site : <https://beagleboard.org/gettingstarted> . Vous pouvez réaliser l'ensemble de ces étapes afin d'avoir très facilement une version d'un Linux embarqué sur la carte Beaglebone. Ces étapes sont à effectuer sur un système Windows (il est possible de faire cela sous GNU Linux, il faut donc remplacer le logiciel de l'étape 2 par un autre).

**Etape 1 :** téléchargement de la dernière version de l'image

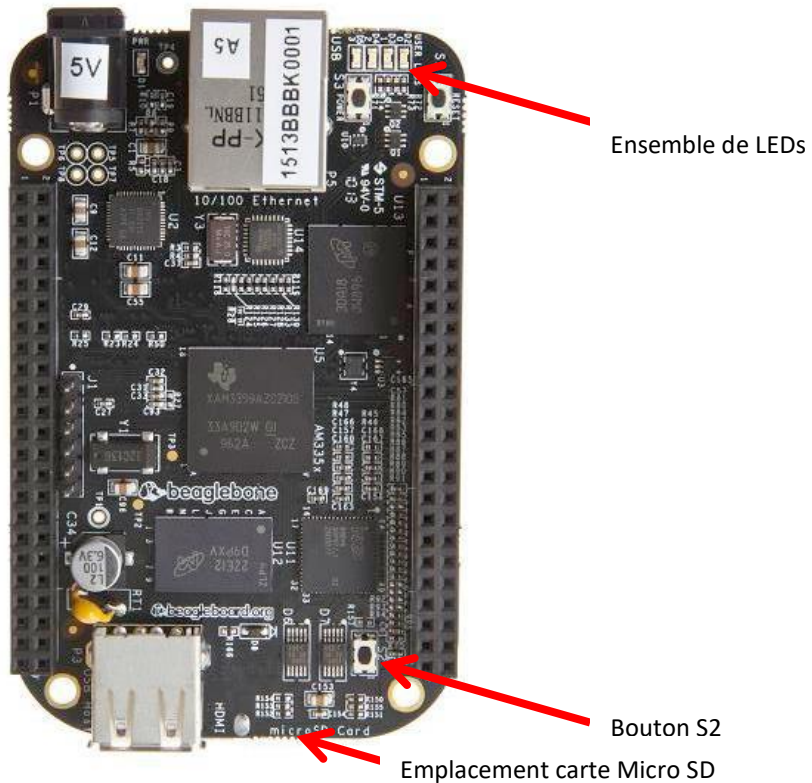
Pour cette première étape, nous utiliserons l'image *Buster IoT (without graphical desktop)* à partir du site [beagleboard.org/latest-images](https://beagleboard.org/latest-images) → O:\share\CSN3A\bone-debian-10.3-iot-armhf-2020-04-06-4gb.img.xz (sinon cf <https://www.beagleboard.org/distros> )

**Etape 2 :** Installer le logiciel de programmation de la carte SD. Il s'agit ici d'installer le logiciel Etcher (<https://etcher.io/>). Nous allons, plus loin, refaire ces étapes en utilisant les commandes Linux (sans outils externe).

**Etape 3 :** Connectez la carte SD à votre ordinateur (via le lecteur de cartes par exemple)

**Etape 4 :** Ecrivez l'image sur votre carte SD en utilisant le logiciel Etcher et éjectez la carte une fois cela réalisé.

**Etape 5** : démarrez la BeagleBone à partir de la carte SD : Une fois la carte SD insérée dans la BeagleBone (en vous assurant qu'elle est débranchée avant), appuyez sur le bouton USER/BOOT (S2) se trouvant à côté de l'interface USB (voir schéma suivant), rebranchez la carte en gardant le bouton appuyé, puis relâchez le bouton.



Une fois votre BeagleBone démarrée, vous verrez 4 LEDs clignoter. Il s'agit des LED USR0, USR1, USR2, et USR3 (D2, D3, D4, D5). Par défaut, ces LEDs clignotent comme suit :

- USR0 est utilisé pour clignoter au rythme du battement de cœur
- USR1 est utilisé pour clignoter lors d'accès à la carte SD
- USR2 est utilisé pour clignoter lors d'activité du CPU
- USR3 est utilisé pour clignoter lors d'accès à la mémoire flash eMMC

Une fois le port USB connecté, un adaptateur réseau devrait apparaître sur votre ordinateur (cela peut prendre quelques minutes). En effet, la distribution installée permet une connexion réseau via l'interface USB. Elle exécute un serveur DHCP qui donne une adresse ip de 192.168.7.1 (ou 192.168.6.1) en fonction du système d'exploitation (plus précisément du type d'adaptateur USB réseau utilisé par le système). La carte se voit réserver l'adresse 192.168.7.2 (ou 192.168.6.2).

Nous allons rendre l'interface USB de la BeagleBone accessible sur la machine virtuelle, pour ce faire, allez sur le menu périphérique de VirtualBox (fenêtre de la machine virtuelle) → USB et activez l'interface de la carte BeagleBone.

	<b>TP4 – OS Avancés et embarqués</b> <b>Utilisation d'une distribution sur étagère</b>	jalil.boukhobza@ensta-bretagne.fr
---	---	-----------------------------------

Une cela réalisé, vous pouvez explorer son contenu en utilisant un explorateur internet (Firefox sur la machine virtuelle) sur l'adresse suivante : <http://192.168.7.2>

## ACCES A LA CARTE A PARTIR D'UNE VM

Nous allons apprendre à accéder à la carte à partir d'une VM.

Afin de pouvoir communiquer avec la carte, nous allons installer un terminal série, « gtkterm » dans notre cas.

Lancer la commande permettant de se faire : `$ sudo apt-get install gtkterm`

Au cas un problème de dépôt se pose, lancez un : `$ sudo apt-get update`

Une fois cela réalisé, branchez la carte, allez sur le menu « Périphérique » puis « USB », vous devriez trouver une nouvelle entrée « BeagleBoard.org ... », cochez cette case.

Maintenant sur la VM Linux, lancez gkterm à partir d'un terminal: `$ sudo gkterm`

Dans gkterm, nous allons configurer le terminal série afin de communiquer avec la carte. Allez sur le menu « configuration », puis « port ». Vous devriez avoir un nouveau port nommé « /dev/ttyACM0 », sélectionnez-le et mettez la vitesse de transmission à « 115200 ». Vous pouvez aussi modifier l'affichage avec le menu « configuration » → « Main Window » afin de modifier la taille et la couleur de l'affichage par exemple.

Une fois la configuration faite, vous pouvez la sauvegarder pour une utilisation future, pour ce faire, allez sur « configuration » → « Sauve la configuration » et donnez un nom.

Maintenant, vous devriez avoir une fenêtre de terminal de login sur votre système Linux sur la carte.

Le login et le mot de passe sont affichés, vous pouvez vous loguez (normalement debian:temppwd).

Explorez votre système Linux.

---


## CREATION D'UN PROGRAMME ET EXECUTION SUR LA CARTE

Nous souhaitons à présent créer et exécuter un programme C.

La distribution que nous venons de porter sur la carte inclut un environnement de développement. Plus précisément, elle inclut un compilateur, nous pouvons donc éditer et compiler notre code directement sur la carte, sans chaine de compilation croisée.

On peut donc écrire notre programme directement en utilisant la ligne de commande du terminal série.

Une autre alternative est de se connecter en ssh sur la carte. Cette dernière est plus simple.

	<b>TP4 – OS Avancés et embarqués</b> <b>Utilisation d'une distribution sur étagère</b>	jalil.boukhobza@ensta-bretagne.fr
---	---	-----------------------------------

Faite à partir de la machine virtuelle un : `$ ssh -X debian@192.168.7.2`

Rappelez-vous que 192.168.7.2 est l'adresse ip de la carte (utilisant l'interface USB pour la connexion).

Une fois cela réalisé, utilisez l'éditeur en ligne de commande de votre choix pour créer un programme simple en C (par exemple vi). Sauvegardez-le et quittez l'éditeur. Vous pouvez à présent compiler votre programme (toujours sur la BeagleBone) et l'exécuter.

Vous venez d'exécuter votre premier code sur un Linux embarqué, félicitations.

Vous trouvez cela peu pratique ?, vous avez un peu raison. Cela pose peu de problèmes pour un développeur chevronné, mais il est difficile de faire des projets d'envergure avec ce mode de développement, finalement peu utilisé pour les plateformes embarquées.

A présent, nous allons tenter d'exécuter les programmes que vous avez déjà développés dans le cadre du premier TP du cours (par exemple) afin de pouvoir les exécuter sur la carte. Pour ce faire nous avons besoin d'envoyer les fichiers de la VM à la carte (après les avoir mis dans la VM). Nous utiliserons le protocole SCP (pour Secure Copy) pour le transfert de fichier. Ce protocole établit une connexion SSH, puis un serveur SCP est lancé (vous pouvez vous documenter sur ce protocole).

L'utilisation de SCP est la suivante :

```
$ scp filename username@ipaddress:/directory/
```

Vous pouvez donc envoyer votre fichier sur la carte de la façon suivante (à partir du terminal de la VM):

```
$ scp fichier.c debian@192.168.7.2:/home/debian
```

Vous retrouverez votre fichier dans le répertoire indiqué. Vous pouvez les compiler et les exécuter.

Aussi, vous pouvez parcourir le répertoire /proc