

# BeagleBone et Linux Debian

---

## 1 CONSIGNES

- Le travail se fait par groupe de 2 élèves.
- Votre BeagleBone ne doit **EN AUCUN CAS** être branché sur le réseau gris. Seul le réseau bleu est autorisé.

## 2 MODALITÉS D'ÉVALUATION

L'évaluation de ce rapport se fera de manière différente que celle pratiquée jusqu'ici. Vous devez suivre les consignes et essayez d'aller aussi loin que possible dans la manipulation tout en comprenant ce que vous faites.

Une première évaluation prendra la forme d'un questionnaire en ligne. Pour la deuxième évaluation, lors de la dernière leçon, PBY et JBN passeront évaluer l'avancement de votre manipulation et vous poseront des questions sur votre travail. Cette évaluation se fera selon une grille prédéfinie qui vous sera transmise. N'hésitez pas à demander des détails si vous avez des questions à ce sujet.

## 3 GÉNÉRALITÉS

### 3.1 MATÉRIEL : LE BEAGLEBONE GREEN

Vous allez utiliser un système embarqué sous forme de BeagleBone. Le matériel nécessaire est le suivant :

- BeagleBone Green, alimentation USB.
- Capteur de température et humidité SHT40

Le BeagleBone Green (BBG) est un ordinateur bon marché d'une taille comparable à une carte de crédit. Il est basé sur un processeur ARM Cortex A8 (CPU à 1 GHz et 512 MB de RAM pour la version utilisée) et permet l'exécution de différents OS, dont Linux. Les BBG fournis viennent préinstallés avec une version de Linux Debian et une adresse IP fixe pour le réseau bleu, indiquée sur le boîtier.

### 3.2 LOGICIELS UTILISÉS

- Lorsqu'il s'agira de se connecter et d'entrer des lignes de commande sur le BBG, vous pourrez le faire à distance à l'aide d'une connexion SSH, au moyen du logiciel **PuTTY**.
- Vous devrez également accéder à son système de fichiers. Vous pourrez le faire en ligne de commande via le terminal ou avec le logiciel **WinSCP**.
- Pour développer du code en langage Python, Il existe de multiples possibilités.
  - Vous pouvez écrire votre code directement dans le terminal au moyen de l'éditeur **nano**
  - Vous pouvez développer sur votre machine Windows dans un IDE de votre choix et transférer les fichiers manuellement ou grâce aux fonctions intégrées de développement à distance.
    - Exemple IDE Python : PyCharm,, VisualStudio, Sublime text, etc...

## 4 OPÉRATIONS

### 4.1 CONNEXION ET PRISE EN MAIN

Pour commencer à utiliser votre BBG, il vous faut le connecter au réseau bleu puis l'alimenter au moyen de l'alimentation USB fournie.

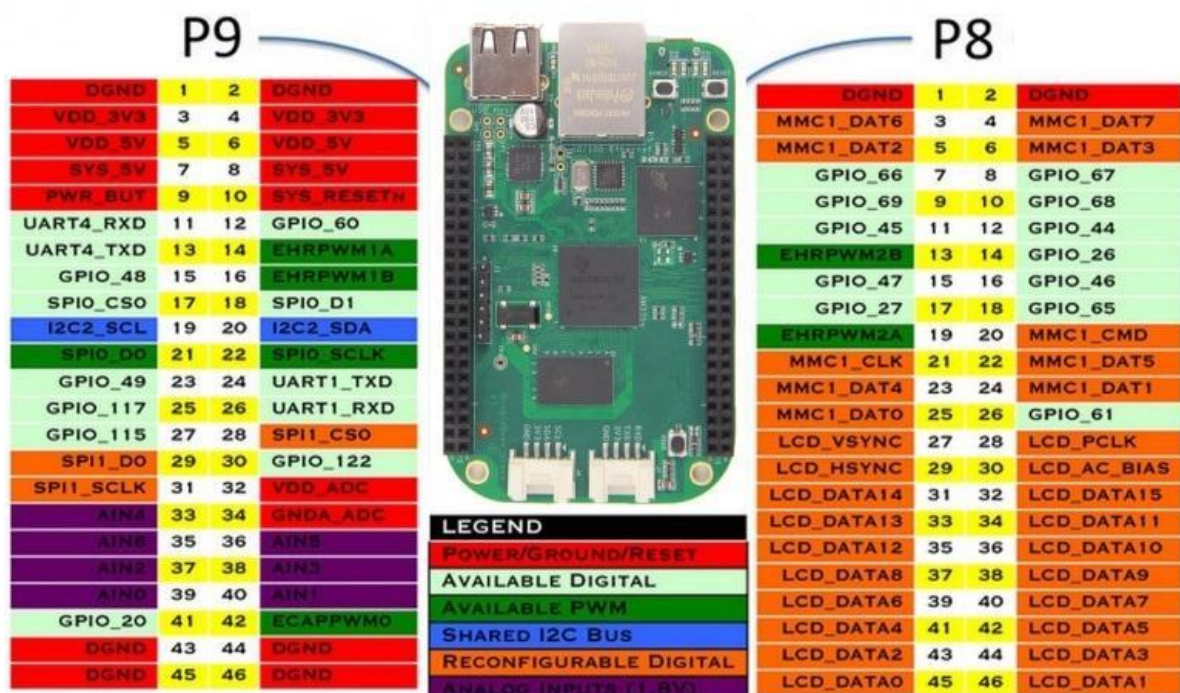
1. Une fois le système démarré (10s. environ) vous pourrez vous y connecter au moyen de SSH
2. Utilisez PuTTY pour démarrer une session SSH
  - Utilisateur : debian
  - Mot de passe : tempwd
3. Au moyen de l'éditeur nano, créez un fichier contenant du texte (vos noms et prénoms par exemple) dans le dossier de l'utilisateur
4. Au moyen d'une session WinSCP, vérifiez que le fichier a bien été créé et vérifiez également son contenu

### 4.2 MESURE DE LA TEMPERATURE

#### 4.2.1 CABLAGE DU CAPTEUR

Il s'agit ici de brancher le capteur au BBG.

1. Câblez un capteur de température et d'humidité SHT40 sur le BBG afin qu'il communique en I2C sur le bus n°2. Aidez-vous des datasheets du capteur et du BBG pour effectuer vos connexions.



#### Connexions sur le BBG

Dans votre rapport, dessinez le schéma de votre connexion.



**Soyez vigilants ! Vous vous connectez directement sur le processeur. Il n'y a aucune protection.**

Prix BBG Mes: 45.-

Prix SHT40 : 5.-

### 4.2.2 PYTHON : MESURE ET AFFICHAGE CONSOLE

Il s'agit ici de lire la température du capteur via un script en langage Python.

Pour ce faire, vous pouvez utiliser un squelette de script « tp\_emsy.py » que vous trouverez dans les documents du TP. Ce fichier contient la structure, et les énoncés de certaines fonctions. Le script n'est volontairement pas complet. A vous d'écrire et de combler aux manquements.

1. Concevez un script Python qui exécute les actions suivantes :
  - Lecture de la température
  - Lecture de l'humidité
  - Calcul du point de rosée
  - Affichage des trois valeurs dans la console



#### Le point de rosée (Dew point), qu'est-ce que c'est ?

Le point de rosée, ou point de condensation est la température à laquelle l'air doit être refroidi pour qu'il se forme de la condensation. Ce phénomène est lié à la saturation de l'air en eau.

Le point de rosée se calcule au moyen de la température et de l'humidité de l'air.

Cette valeur est utilisée en météorologie, en aviation (pour prédire la formation de glace sur la voilure ou un éventuel carburateur par exemple) et dans les systèmes de ventilation et de climatisation pour éviter la formation d'eau.

Le point de rosée peut se calculer de plusieurs manières. La formule ci-dessous en est une, qui permet d'obtenir des résultats satisfaisants entre -45°C et 60°C.

Calcul du point de rosée :

$$Dp(T, RH) = \frac{\lambda \cdot \left( \ln\left(\frac{RH}{100}\right) + \frac{\beta \cdot T}{\lambda + T} \right)}{\beta - \left( \ln\left(\frac{RH}{100}\right) + \frac{\beta \cdot T}{\lambda + T} \right)}$$

Avec :

$$\begin{aligned} Dp &= \text{point de rosée } [^{\circ}\text{C}] \\ T &= \text{température } [^{\circ}\text{C}] \\ RH &= \text{humidité relative } [\%] \\ \beta &= 17.62 [\text{Coefficient de Magnus}] \\ \lambda &= 243.12 [\text{Coefficient de Magnus}] \end{aligned}$$

2. Documentez votre code et son exécution dans la console Linux.

A partir de ce point, vous pouvez réaliser les parties suivantes dans l'ordre que vous souhaitez, selon vos intérêts personnels. Les parties réalisées seront évaluées lors de la revue de projet. L'ensemble des points suivants comprend des notions qui vous seront utiles dans la compréhension de Python et des systèmes Linux en général. Ces connaissances vous seront précieuses pour la deuxième année et vos vies professionnelles en général.

## 5 ENREGISTREMENT CSV

Enregistrement des trois valeurs dans un fichier CSV :

La valeur lue doit être ajoutée au fichier `/home/debian/TempLog.csv` , avec le format suivant : `<date>,<heure>,<température>,<humidité>,<point de rosée>`

Exemple : `07.04.2014,17:51,23.5,54.1,10.2`

Chaque nouvelle valeur est mise sur une nouvelle ligne.

- Exécutez votre script et vérifiez le contenu du fichier `.csv` généré

## 6 ENVOI D'EMAIL

Envoi d'un email d'alarme lorsqu'une valeur limite est dépassée (28°C). Pour ce faire, aidez-vous du squelette de fonction « `send_email.py` » situé dans le dossier du TP.

- Complétez le fichier `tp_emsy.py` en vous inspirant de `send_email.py`
- Exécutez le script et testez son fonctionnement
- Les paramètres smtp pour l'envoi de l'email fonctionnent à travers un compte "`@etml-es.ch`" (Infomaniak).

## 7 EXÉCUTION AUTOMATIQUE

Lancez votre script à intervalle régulier de manière automatique (tâche planifiée). Sous Linux Debian, on parle de tâche "crontab". Ajoutez une tâche au système qui lance votre script toutes les 15 minutes.

Pour afficher la liste des crontab déjà en place, utilisez la commande `crontab -l`. Pour éditer la liste, utilisez `crontab -e`. Vous devrez alors rajouter votre ligne à la liste, sous la forme :

`mm hh jj MMM JJJ <commande>`

Avec respectivement la minute, l'heure, le jour du mois, le mois et le jour de la semaine auquel(les) votre tâche doit s'exécuter :

### Exemples

- `30 23 * * * /usr/bin/python3 /debian/script.py # tous les jours à 23h30`
- `30 23 1 * * /debian/test.sh # tous les 1er du mois à 23h30`
- `15,30,45,0 * * * * /debian/test.sh # tous les quarts d'heure`
- `@reboot /debian/test.sh # à chaque démarrage de l'ordinateur`

Vérifiez le fonctionnement (fichier log et alerte e-mail).