

# PPE Raspberry

Meryeme ITTO-HEMAD  
Jules HATCHUEL  
Anthony RUGGERI

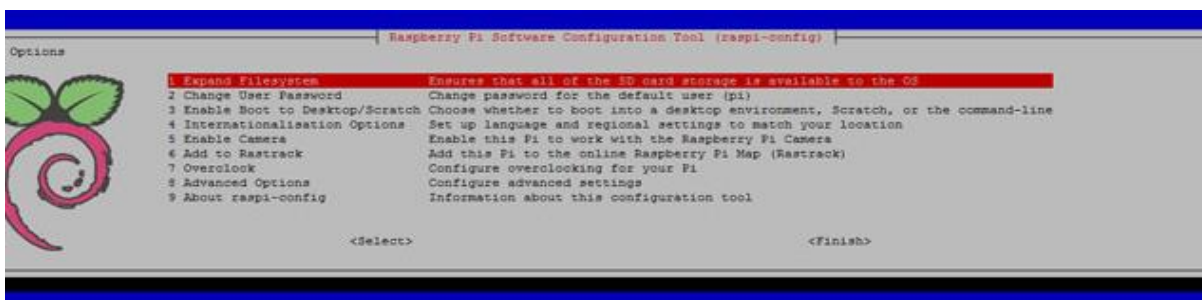
Le Raspberry PI est un nano ordinateur de la taille d'une carte de crédit que l'on peut brancher à un écran et utiliser comme un ordinateur standard. Sa petite taille, et son prix intéressant font du Raspberry PI un produit idéal pour tester différents domaines de l'informatique en abordant aussi bien les aspects matériels que logiques, et notamment la création d'un serveur Web, caméra IP, contrôleur de composants électroniques etc. Le système d'exploitation utilisé est Raspbian, une adaptation de Debian nécessitant moins de puissance.

Le Raspberry PI ne dispose pas d'un disque dur interne (cela augmenterait grandement sa taille) en lieu et place il utilise une carte microSD. Par défaut, le Raspberry pi, est nu, ou vendu avec la carte mémoire comprise par là qu'il est vendu sans accessoires. Pour pouvoir l'utiliser il faut donc avoir :

- Une carte micro SD compatible avec le modèle Raspberry pi que vous avez choisi ([http://elinux.org/RPi\\_SD\\_cards](http://elinux.org/RPi_SD_cards))
- Un câble d'alimentation micro USB standard
- Un câble RJ45 pour se connecter au réseau
- Un câble HDMI afin de connecter le Raspberry pi un écran ou une télévision.
- Un clavier pour taper les commandes

Le premier problème que l'on rencontre est donc l'installation d'un système d'exploitation. Heureusement pour nous cette opération est relativement facile grâce à l'utilisation de la carte SD. La première étape sera donc de préparer la carte micro SD en y installant un système d'exploitation. On peut installer différentes distributions, la plus connue étant Raspbian (Debian pour RPi) que l'on peut télécharger directement depuis le [site Internet de Raspberry pi](#).

Une fois la carte micro SD préparée, il vous suffira de l'insérer dans votre Raspberry pi pour faire fonctionner automatiquement le système. Suivant la distribution que vous avez choisie vous devriez tomber sur l'écran de configuration de votre Raspberry pi.



**Motion** est un logiciel de détection vidéo et qui permet de diffuser un flux vidéo via internet par le protocole HTTP, en envoyant des photos successives au client.

TP

**Démarrage et connexion au Raspberry PI (RPI)**

Avant de se connecter il faut s'assurer avant de le brancher à l'alimentation :

- De bien connecter le module de camera
  - o Le port où se connecte la nappe de connexion doit être ouvert (soulever la petite partie grise qui débloque le port camera)
  - o La fiche de connexion doit être insérée dans le bon sens (1 chance sur deux) ou plutôt les connecteurs doivent être vers la prise HDMI
- Connecter au réseau du BTS 125 par le câble Ethernet (le Wifi est désactivé sur ce RPI)

Puis on branche le transformateur en micro USB à notre RPI pour l'alimenter. Le RPI démarre dès qu'il est branché. Il faut lui laisser quelques minutes puis on se connecte. Il est préconfiguré et on peut s'y connecter par l'adresse 125.125.2.3 en SSH en suivant ces étapes :

- Ouvrir l'invite de commande
- Entrer la commande
  - o `ssh -l pi 125.125.2.3` (entrée)
  - o Mot de passe : `admin` (entrée)

Pour tester si la camera est fonctionnelle on peut exécuter une commande qui va prendre une photo (avec un timer de 5sc pour faire un joli sourire)

- `raspistill -o image01.jpg -t 5000`

## Python

Pour effectuer un test de performances et pour vérifier que Python fonctionne sur ce Raspberry nous allons créer un script Python qui utilise la caméra. L'objectif est de prendre 5 photos à la suite en optimisant le code, en effet, pour ne pas que les photos soient remplacées par les suivantes on doit pouvoir changer leur nom de façon automatisée et ainsi créer par exemple : « photo1.jpg, photo2.jpg, etc. jusqu'à 5).

Ce dont on a besoin :

- Une boucle qui va répéter une exécution un certain nombre de fois
- La/les commandes à exécuter dans cette boucle
- Une variable contenant le numéro de photo à donner aux noms de fichiers
- Un petit timer pour nous laisser 1sc entre chaque photo

Voici la syntaxe de base du code dont nous avons besoin :

```
import os, time
```

(ici on importe les librairies dont python a besoin : os permet d'intégrer facilement avec Linux, time donne plein de fonctions pratiques en rapport avec le temps et les dates)

```
variable=0
```

(initialisation d'une variable et attribution d'une valeur = 0)

```
variable+=1
```

(on ajoute 1 à la valeur de variable)

```
for i in range (0, 100):
```

(cette boucle sera exécutée 100 fois)

```
os.system("raspistill -w 1024 -h 768 -t 500 -o ma_photo.jpeg")
```

(cette commande permet de prendre une photo de dimension 1024/768px après un laps de temps de 500ms et dont le nom est ma\_photo.jpeg)

**-w** : width ou largeur en px

**-h** : height ou hauteur en px

**-t** : time ou temps d'attente afin que le module se règle tout seul en luminosité

**-o** : output ou nom du fichier de sortie

```
time.sleep(1)
```

(cette commande met l'exécution du script en pause pendant 1 seconde)

```
str(variable)
```

(cette commande retourne en chaîne de caractères la valeur numérique de variable)

Le script doit faire 6 lignes si tout va bien.

### **Création d'un script Python :**

Sous linux, passez en mode super utilisateur si ce n'est pas déjà fait (`sudo su`)

puis placez-vous dans votre dossier utilisateur (`cd /home/pi`)

créez un fichier python contenant notre script (`nano mon_script.py`)

L'éditeur de texte crée alors un fichier vide qu'il faut compléter avec le code, à vous de jouer.

### **Lancement d'un script Python :**

```
python mon_script.py
```

### **Vérifier si les fichiers sont créés :**

```
ls
```

**Récupérer les fichiers pour les visualiser sur Windows :**

Il faut avoir Filezilla d'installé puis renseigner les trois champs en haut à gauche :

Hôte : `sftp://192.168.2.3`

Utilisateur : `pi`

Mot de passe : `admin`

Transférez ensuite en glissant les fichiers voulu vers votre bureau (filezilla vous place par défaut dans votre dossier utilisateur : `/home/pi`)

### Optionnel : code pour une vidéo

```
import os, picamera, time

camera = picamera.PiCamera()
nomVideo = time.strftime("%d.%m.%Y-%Hh%Mm%Ss")
dureeVideo = 10

camera.start_recording(nomVideo+".h264")
print("REC...")
time.sleep(dureeVideo)
camera.stop_recording()

commande = "MP4Box -add "+nomVideo+".h264 " + nomVideo+".mp4"

os.system(commande)
```