Annexes :

A. Commandes et redirection

Les commandes Linux acceptent de nombreuses options dont on peut consulter la documentation en tapant **`man ls`** ou **`ls --help`** par exemple pour la commande **`ls`**. Celle-ci comporte des options pour affiche les fichiers cachés **`ls -a`** ou encore pour afficher les détails et permissions d'un fichier : **`ls –l`**

COMMANDE	DESCRIPTION
`cd`	*Change Directory* : se déplacer dans l'arborescence
`ls`	*List* : liste le contenu du répertoire courant
`mkdir`	*Make Directory* : créer un dossier
`pwd`	*Print Working Directory* : affiche le dossier courant
`cp`	*Copy* : copier des fichiers ou répertoires
`mv`	*Move* : déplace un dossier ou un répertoire
`rm`	*Remove* : supprime des fichiers ou des répertoires
`cat`	Permet de visualiser le contenu d'un fichier
`echo`	Affiche un message ou le contenu d'une variable
`touch`	Créer un fichier vide
`chmod`	Modifier les autorisations d'accès à un fichier
`nano`	Editeur de texte. Nécessite d'être installé
`ps`	Afficher les informations des processus en cours
`top`	Gestionnaire de ressources
`kill`	Envoyer un signal au processus, généralement pour l'arrêter
`grep`	Filtrer une sortie en ne gardant que les lignes de contenant un terme
`|` **`(altgr + 6)`**	*Pipe* (tuyau) : permet de rediriger le contenu de la commande de gauche, vers la commande de droite. Exemple : **`ls | grep toto`** : va afficher tous les fichiers du répertoire courant qui contiennent toto

Linux permet également la redirection des sorties de commande vers un fichier par exemple. Dans ce cas, on utilise les chevrons : **`>`** (va remplacer le contenu du fichier) ou **`>>`** (va concaténer le résultat à la fin du fichier destinataire).

Une commande **`echo 'bonjour'`** aura pour sortie « bonjour ». Cette sortie peut être redirigée afin de ne pas être affichée du tout, ou être dirigée vers un fichier texte, ou de log par exemple :

```
$ echo 'bonjour'
bonjour
$ echo 'bonjour' > /dev/null # on ne cherche pas à récupérer le contenu
$ echo 'bonjour' > text.txt
$ cat text.txt
bonjour
```

B. Installer des paquets Linux

Linux fonctionne avec des paquets : des petites applications très spécifiques qui ont un but précis. Les plus basiques et nécessaires sont installées de base. Mais très rapidement, nous allons devoir installer de nouveaux paquets. Prenons l'exemple avec **nano**, une application facile d'utilisation pour la modification de fichiers. Pour ces commandes, il faut un accès à Internet :

```
$ sudo apt-get update #permet de mettre à jour la liste connue des paquets existants
$ sudo apt-get install nano # permet l'installation du paquet nano
```

C. Installer des librairies Python

Dans le cadre du cours, nous utiliserons également Python. De la même manière que Linux fonctionne par paquet, Python fonctionne par librairies. Pour l'installation de nouvelles librairies et pour les utiliser dans les programmes Python, il faut d'abord les installer sur le système avec l'installeur de librairies Python (**pip3**). Afin de faire communiquer un programme Python sur Linux avec une carte Arduino connectée en série, il faut utiliser la librairie **pyserial :**

```
$ pip3 install pyserial
```

Ensuite, pour les utiliser, il faudra les importer dans le fichier Python. Voici un exemple d'importation de librairies dans le cas d'utilisation de notions de temps (sleep, date, heure…). Ces lignes se placent généralement au début d'un script Python :

```
import datetime                 # importe toute la librairie
from datetime import sleep      # importe uniquement sleep de la librairie datetime
```