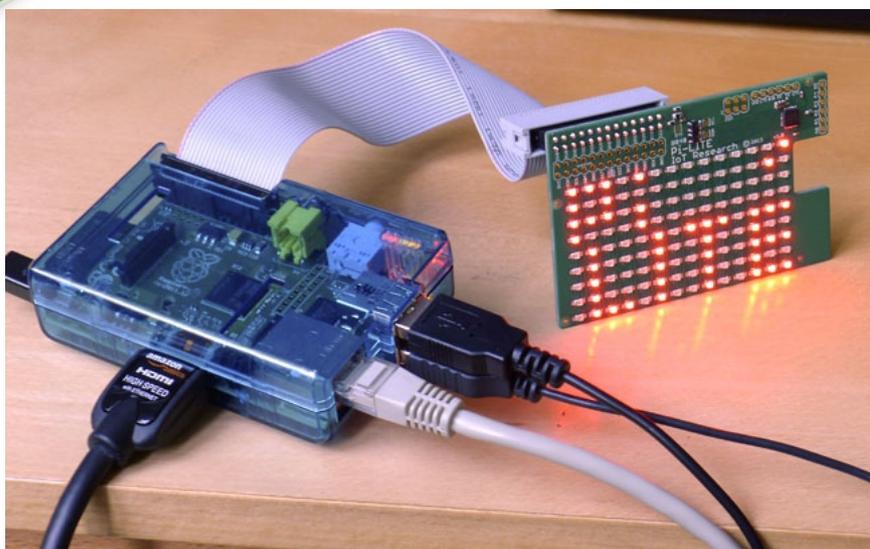


# Activités autour de la carte *Raspberry Pi*

## *Application à un journal lumineux connecté*

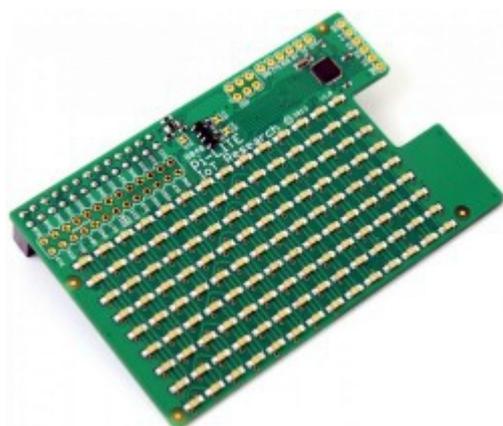
### Sommaire

1. Présentation du support et objectifs pédagogiques.....	2
2. Découverte et première mise en œuvre de la Raspberry Pi.....	3
2.1 Où trouver la Raspberry Pi.....	3
2.2 Préparation de la carte SD.....	3
2.3 Démarrer avec la Raspberry Pi – un miniordinateur à part entière.....	4
2.4 Découverte des caractéristiques principales.....	5
2.5 L'environnement graphique.....	6
2.6 Le système de fichiers.....	8
2.7 Raspberry Pi sur le réseau.....	9
3. Connexion à la Raspberry Pi sur un poste distant.....	11
3.1 Putty.....	12
3.2 FileZilla.....	12
4. Mise en œuvre de la Pi-Lite.....	14
4.1 Communication série Raspberry Pi – Pi-Lite.....	14
4.2 Un premier programme Python « Hello World ».....	16
5. Activité/Projet Raspberry Pi – un journal lumineux connecté.....	17
5.1 Des données au format structuré XML.....	18
5.2 Analyse du fichier XML.....	19
5.3 Exécuter un programme Python à intervalle régulier.....	20
5.4 Le cahier des charges du projet.....	21
6. Bibliographie et sitographie.....	23



# 1. Présentation du support et objectifs pédagogiques

La carte *Raspberry Pi* est une véritable carte mère de mini-ordinateur de la taille d'une carte de crédit à très bas coût (environ 30 € pour le modèle B sorti en février 2012). Conçu dans le cadre de la « Fondation Raspberry », cet outil initialement destiné à l'initiation de la programmation informatique en Angleterre est suffisamment ouvert pour satisfaire de nombreuses exigences en termes d'embarqué. Fournie nue (la carte mère seule, sans clavier ni souris, sans écran, et même sans périphérique de stockage ou système d'exploitation), cette carte sera une occasion d'aborder l'architecture matérielle des ordinateurs avec les élèves.



Afin de compléter les activités de programmation en langage Python (le langage de prédilection à l'origine du « Pi » de Raspberry Pi), on propose d'équiper la carte d'une matrice de 14 x 9 LED du constructeur Ciseco. La Pi-Lite (environ 20 £) s'interface très facilement sur la carte Raspberry Pi. Les fonctionnalités d'affichage proposées par le constructeur (défilement de textes paramétrable, dessins de bargraphes ou vumètres, etc.) et le mode de communication sérielle choisi entre la Pi-Lite et la Raspberry Pi sont autant d'occasions d'aborder à moindre investissement l'informatique physique sur un support original.

<http://www.youtube.com/watch?v=tyS45yYtKJU&hd=1>

Les chapitres 2,3 et 4 ne fournissent pas à proprement parler d'activités clés en main mais doivent permettre de s'approprier le matériel et les logiciels par les professeurs enseignant en ISN. Néanmoins ils sont rédigés de sorte que certaines parties peuvent être reprises pour proposer une ou plusieurs activités à moduler selon le temps et le matériel disponibles. Le chapitre 5 propose une activité plus soutenue et plus détaillée qui peut être proposée comme mini-projet de fin d'année. Ce mini-projet, défini par un besoin, sera alors le point d'entrée de l'activité et sa réalisation fera appel aux ressources des chapitres 2, 3 et 4.

## 2. Découverte et première mise en œuvre de la Raspberry Pi

**Niveau : Classe de Terminale S spécialité ISN**

**Moyens :**

Poste informatique

Raspberry Pi modèle B + connectique + carte SD (4 Go minimum) et lecteur cartes SD + clavier et souris USB.

**Éléments du programme**

Architectures matérielles :

- éléments d'architecture

Réseaux :

- adressage sur un réseau

### 2.1 Où trouver la Raspberry Pi

Il est aisé de trouver une carte Raspberry Pi seule, notamment chez les distributeurs officiels :

<http://www.kubii.fr/fr/>

<http://fr.farnell.com/raspberry-pi>

<http://www.element14.com/community/community/raspberry-pi>

Le modèle B actuel est aux environs de 25-30 € TTC l'unité (sans les frais de port), mais il faut rajouter au minimum :

- une carte SD ;
- un adaptateur secteur (5 V – 1 A) pour alimenter la carte via la prise micro-USB ;
- un adaptateur HDMI-VGA ou HDMI-DVI ;
- un câble Ethernet RJ45.

En fonction du matériel dont vous souhaitez vous équiper, il sera alors pertinent ou non de rechercher des « kits de démarrage » (ou *starter kit*) comprenant la carte embarquée et un minimum d'accessoires.

### 2.2 Préparation de la carte SD

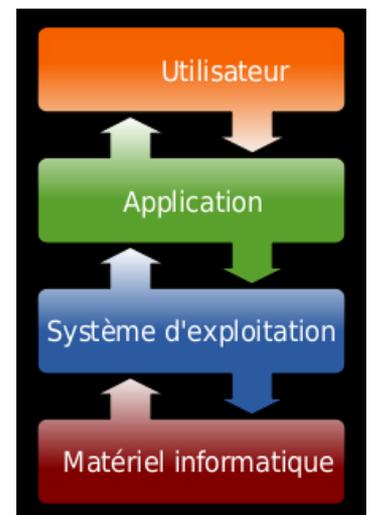
La Raspberry Pi n'a pas de disque dur pour le stockage de masse, mais un lecteur de carte mémoire SD (*Secure Digital*) telles celles que l'on peut trouver équipant les appareils photo numériques.

Avant de démarrer la Raspberry Pi, il faut préparer une carte SD et y installer la totalité du système d'exploitation (ou OS pour *Operating System*).

La Raspberry Pi est conçue pour faire fonctionner le système d'exploitation GNU Linux. La philosophie open source de Linux a permis de porter rapidement l'OS à l'architecture matérielle de la Raspberry Pi.

Il existe à l'heure actuelle plusieurs variantes de Linux utilisables avec le circuit de la Raspberry Pi, appelées distributions Linux (Raspbian, Arch, Pidora, etc.).

Par la suite, nous utiliserons une distribution Debian portée et optimisée pour Raspberry Pi nommée Raspbian.



d'après Wikipedia

**Note :** afin de s'affranchir de configurations parfois fastidieuses, Ciseco propose même d'utiliser une image avec tous les outils nécessaires au bon fonctionnement de la Pi-Lite et disponible à l'adresse :

<http://www.openmicros.org/Download/2013-05-25-wheezy-raspbian-ciseco.img.zip>

(Prendre la dernière version à l'adresse <http://www.openmicros.org/Download/>)

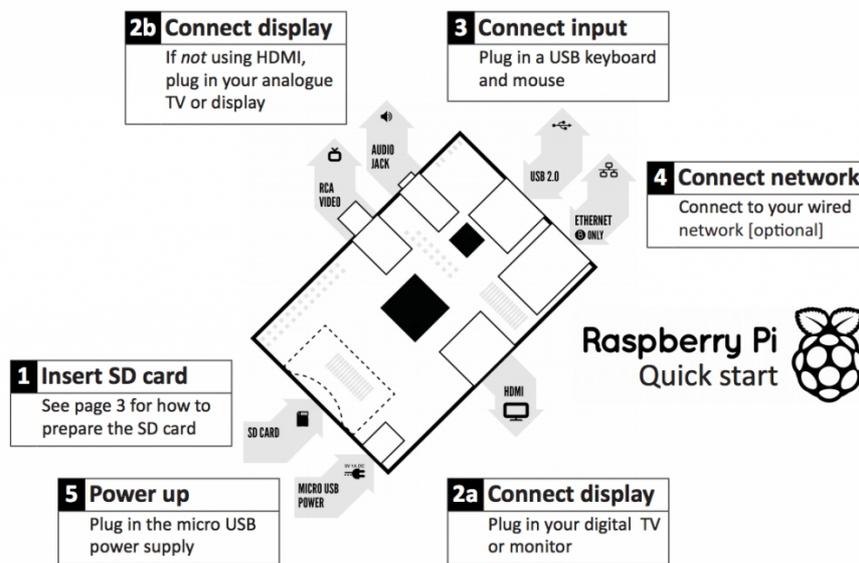
**Préparation de la carte sous Windows :** on peut prendre l'utilitaire *Image Writer for Windows* (voir [Installation de la carte SD du Raspberry Pi](#)).

**Note :** utilitaire *Image Writer for Windows*, aller sur le site <https://launchpad.net/win32-image-writer>. Pensez à télécharger le fichier *binaire*.

## 2.3 Démarrer avec la Raspberry Pi – un miniordinateur à part entière

### branchements

La Raspberry Pi est raccordée à un moniteur, un clavier et une souris USB. Pensez à insérer la carte SD avant d'alimenter la Raspberry Pi.



1

d'après <http://www.raspberrypi.org/new-quick-start-guide/>

### démarrage et configuration

Une fois la Raspberry Pi alimentée, des informations de démarrage défilent à l'écran puis quand tout est « [OK] », un login est demandé. Le login par défaut est « pi » :

`login : pi`

le mot de passe est « raspberry » mais si vous utilisez un clavier « azerty », il faut saisir « rqsberry ».

Dans le cas d'un clavier « azerty », il faut en informer le système en entrant dans un utilitaire de configuration avec la commande :

`sudo raspi-config`

```
Raspberry Pi Software Configuration Tool (raspi-config)
Setup Options
1 Expand Filesystem      Ensures that all of the SD card storage is available to the OS
2 Change User Password   Change password for the default user (pi)
3 Enable Boot to Desktop Choose whether to boot into a desktop environment or the command-line
4 Internationalisation Options Set up language and regional settings to match your location
5 Enable Camera          Enable this Pi to work with the Raspberry Pi Camera
6 Add to Rastrack        Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock              Configure overclocking for your Pi
8 Advanced Options       Configure advanced settings
9 About raspi-config     Information about this configuration tool

                <Select>                                <Finish>
```

Les informations locales sont à renseigner dans le menu 4. Il faudra se rendre ensuite dans les sous-menus I1, I2 et I3 :

```
Raspberry Pi Software Configuration Tool (raspi-config)
Internationalisation Options
I1 Change Locale          Set up language and regional settings to match your location
I2 Change Timezone       Set up timezone to match your location
I3 Change Keyboard Layout Set the keyboard layout to match your keyboard

                <Select>                                <Back>
```

Par défaut on choisira la configuration `fr-FR. UTF-8 UTF-8`

Pour prendre en compte les informations, il est recommandé de redémarrer le système avec la commande :  
`sudo reboot`

## 2.4 Découverte des caractéristiques principales

La Raspberry Pi est dorénavant un miniordinateur opérationnel dont on peut étudier les caractéristiques principales à l'aide de quelques commandes à base Unix.

Dans le cadre d'activités de découverte avec les élèves, on peut donner un lien vers l'aide de la commande en ligne Unix, [vcgencmd](#).

Par exemple pour mesurer la fréquence du processeur Broadcom BCM2835 (architecture ARM) :

`vcgencmd measure_clock arm`

```
pi@raspberrypi ~ $ vcgencmd measure_clock arm
frequency (45)=700074000
```

qui donne la fréquence d'environ 700 000 000 Hz, soit 700 MHz qui est la fréquence par défaut (possibilité d'*overclocking*) à l'achat de la Raspberry Pi.

La mémoire vive est constituée d'une puce Samsung visible à la surface de la carte. L'inscription « 4G » représente la quantité de mémoire vive en GigaBits, soit 4 GigaBits = 512 Mo pour la Raspberry modèle B.



Le processeur de la carte est accompagné d'un processeur graphique (ou GPU pour *Graphics Processing Unit*) afin d'accélérer les tâches graphiques (et celui de la Raspberry Pi est même plutôt performant pour accélérer le décodage des vidéos en *full HD*). Une partie des 512 Mo de mémoire vive est spécialement dédiée au GPU. La répartition mémoire peut être obtenue grâce aux commandes :

```
pi@raspberrypi ~ $ vcgencmd get_mem arm
arm=448M
pi@raspberrypi ~ $ vcgencmd get_mem gpu
gpu=64M
```

Avec la commande `sudo fdisk -l`, la carte SD utilisée ici apparaît sous le nom `/dev/mmcblk0` avec deux partitions `p1` et `p2` (ici environ 8 Go au total) :

```
pi@raspberrypi ~ $ sudo fdisk -l

Disk /dev/mmcblk0: 7969 MB, 7969177600 bytes
4 heads, 16 sectors/track, 243200 cylinders, total 15564800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000c7b31

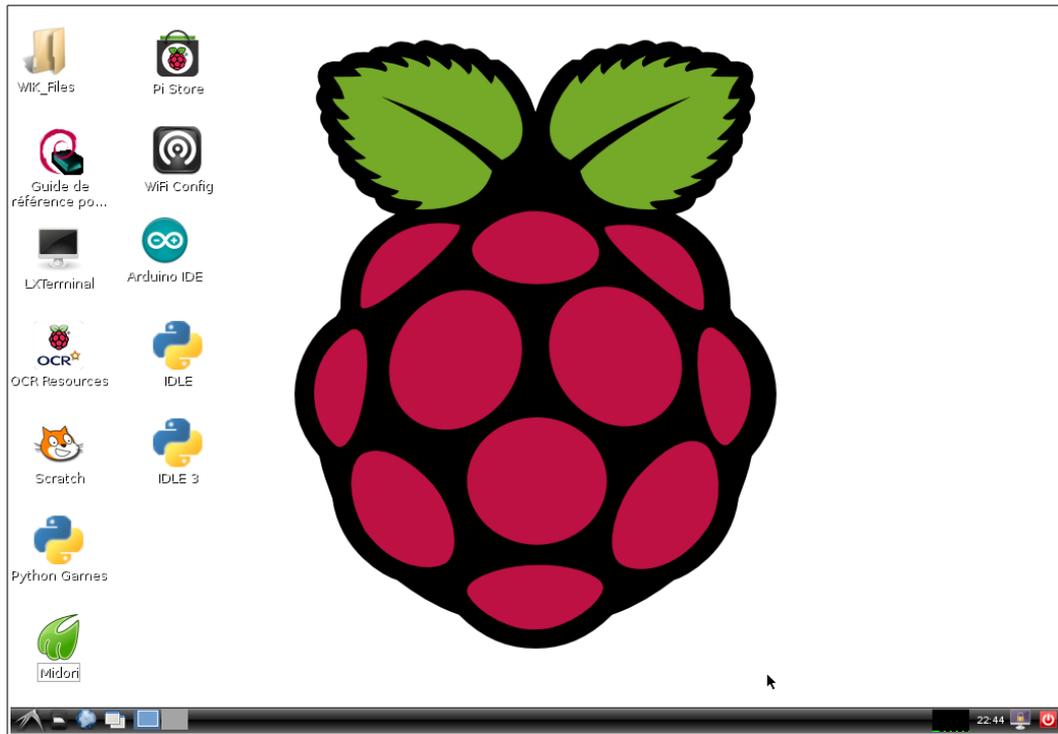
   Device Boot      Start         End      Blocks    Id  System
/dev/mmcblk0p1            8192      122879       57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2       122880      7626751      3751936    83   Linux
pi@raspberrypi ~ $
```

## 2.5 L'environnement graphique

L'environnement graphique ne démarre pas de lui-même dans la plupart des distributions Linux pour Raspberry Pi. Afin de quitter la console en mode texte, on saisit la commande :

```
startx
```

L'environnement de bureau graphique LXDE (*Lightweight X11 Desktop Environment*) propose un dialogue simplifié avec l'utilisateur auquel sont habitués ceux qui utilisent Windows ou Mac OS X.



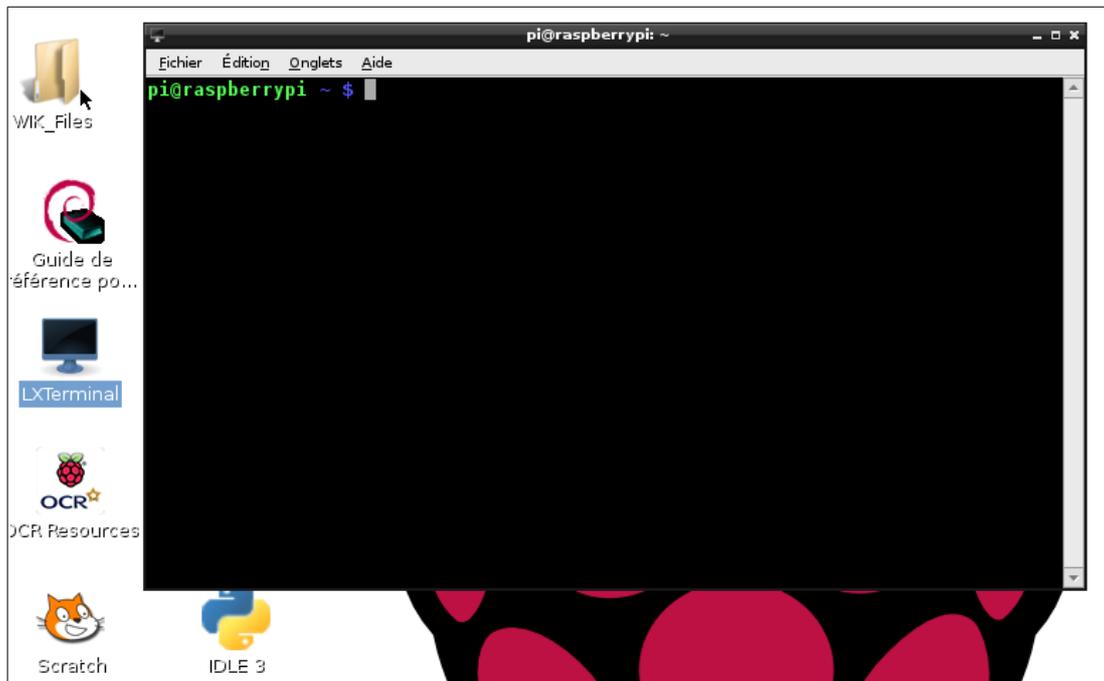
IDLE

À gauche du bureau, on trouve les raccourcis IDLE et IDLE3 vers les ateliers de développement pour le langage Python (respectivement pour les versions 2.7 et 3.x).



IDLE 3

On trouve également le raccourci vers LXTerminal qui ouvre un terminal permettant d'émettre des commandes Linux dans une fenêtre en mode texte sans quitter l'interface graphique.



## 2.6 Le système de fichiers

Les données sont structurées de façon logique grâce à une arborescence de répertoires.

Dans un terminal (icône *LXTerminal*), on peut découvrir en partie cette arborescence avec un jeu d'instructions :

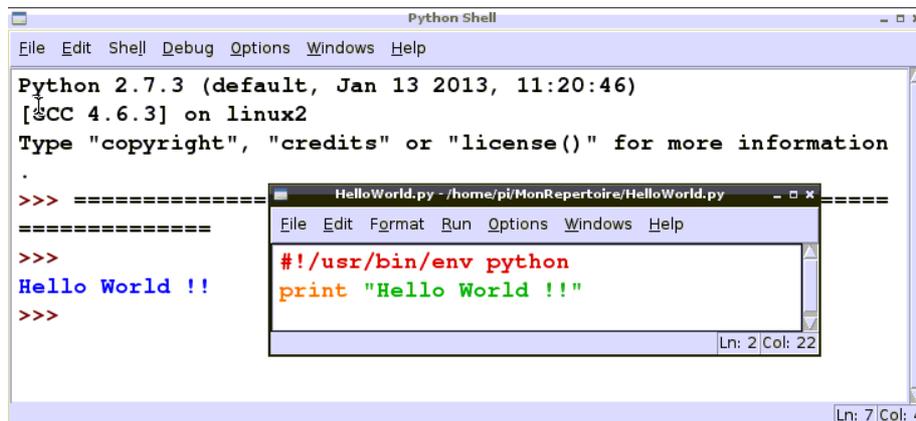
`cd` : (*Change Directory*) permet de naviguer d'un répertoire à un autre dans le système de fichiers  
`ls` : (*LiSting*) affiche le contenu du répertoire courant  
`mkdir` : créer un nouveau répertoire  
`rmdir` : supprimer un répertoire une fois qu'il a été vidé de ses fichiers avec la commande `rm` (*Remove*)

Le répertoire par défaut à l'ouverture du terminal est `/home/pi` (signalé par le `~` dans le prompt).

Un exemple de séquence commentée :

<code>cd /</code>	Aller dans le répertoire racine.
<code>man ls</code>	<code>man</code> pour <i>manuel</i> , aide pour la commande <code>ls</code> . Un bon réflexe pour obtenir de l'aide.
<code>ls</code>	Lister le contenu du répertoire.
<code>ls -l</code>	Avec l'option <code>-l</code> .
<code>cd home</code>	Chaque compte d'utilisateur possède un sous-répertoire dans ce répertoire pour y stocker ses données personnelles.
<code>cd pi</code>	Compte utilisateur <code>pi</code> par défaut.
<code>mkdir MonRepertoire</code>	Créer un répertoire nommé <code>MonRepertoire</code> dans le répertoire courant.
<code>cd MonRepertoire</code>	Aller dans le répertoire <code>MonRepertoire</code> .
<code>ls -l</code>	Lister le contenu du répertoire (vide pour l'instant).

Grâce à l'environnement de développement Python IDLE, on peut créer un premier programme `HelloWorld.py` sauvegardé dans `MonRepertoire` :



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.3 (default, Jan 13 2013, 11:20:46)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information
.>>> =====
>>>
Hello World !!
>>>

HelloWorld.py - /home/pi/MonRepertoire/HelloWorld.py
File Edit Format Run Options Windows Help
#!/usr/bin/env python
print "Hello World !!"
Ln: 2 Col: 22

Ln: 7 Col: 4
```

Dans un terminal, on lance l'exécution du programme avec la commande :

```
python HelloWorld.py
```

```
pi@raspberrypi: ~/MonRepertoire
Fichier  Édition  Onglets  Aide
pi@raspberrypi ~/MonRepertoire $ ls -l
total 4
-rw-r--r-- 1 pi pi 45 janv.  6 22:40 HelloWorld.py
pi@raspberrypi ~/MonRepertoire $ python HelloWorld.py
Hello World !!
pi@raspberrypi ~/MonRepertoire $
```

Pour rendre le fichier HelloWorld.py exécutable, on utilise la commande chmod avec l'option +x de la façon suivante :

```
chmod +x HelloWorld.py
```

```
pi@raspberrypi: ~/MonRepertoire
Fichier  Édition  Onglets  Aide
pi@raspberrypi ~/MonRepertoire $ ls -l
total 4
-rw-r--r-- 1 pi pi 45 janv.  6 22:40 HelloWorld.py
pi@raspberrypi ~/MonRepertoire $ chmod +x HelloWorld.py
pi@raspberrypi ~/MonRepertoire $ ls -l
total 4
-rwxr-xr-x 1 pi pi 45 janv.  6 22:40 HelloWorld.py
pi@raspberrypi ~/MonRepertoire $
```

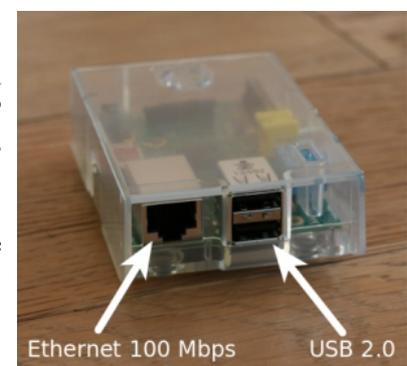
On peut maintenant lancer l'exécution sans invoquer le programme Python grâce à la commande :

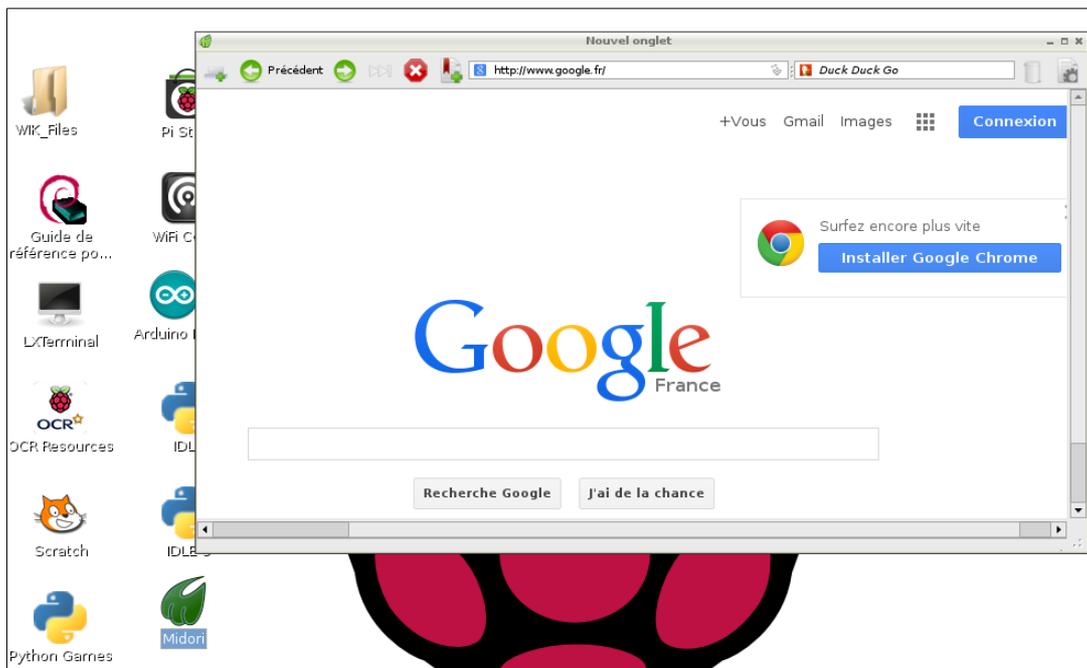
```
./HelloWorld.py
```

```
pi@raspberrypi: ~/MonRepertoire
Fichier  Édition  Onglets  Aide
pi@raspberrypi ~/MonRepertoire $ ls -l
total 4
-rwxr-xr-x 1 pi pi 45 janv.  6 22:40 HelloWorld.py
pi@raspberrypi ~/MonRepertoire $ ./HelloWorld.py
Hello World !!
pi@raspberrypi ~/MonRepertoire $
```

## 2.7 Raspberry Pi sur le réseau

Pour connecter la Raspberry Pi au réseau, on profite du fait que le réseau où sont connectés les ordinateurs de la salle de classe profite d'un service DHCP (*Dynamic Host Configuration Protocol*) dont la fonction est de distribuer des adresses IP aux hôtes connectés. Le client DHCP est activé par défaut sur la distribution Linux proposée et installée sur la carte SD de sorte que vous n'avez plus qu'à connecter votre Raspberry Pi à une prise Ethernet murale disponible de la salle (ou via un *switch*).





Sur la copie d'écran ci-dessus, on a ouvert le navigateur Midori. La Raspberry Pi est connectée au réseau Internet.

Pour voir la configuration réseau, saisir la commande `ifconfig` dans un Terminal :

```
pi@raspberrypi: ~  
Fichier  Édition  Onglets  Aide  
pi@raspberrypi ~ $ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:cf:5e:c9  
          inet adr:192.168.1.11  Bcast:192.168.1.255  Masque:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:202607 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:209750 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 lg file transmission:1000  
          RX bytes:199690397 (190.4 MiB)  TX bytes:169528400 (161.6 MiB)  
  
lo        Link encap:Boucle locale  
          inet adr:127.0.0.1  Masque:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 lg file transmission:0  
          RX bytes:1504 (1.4 KiB)  TX bytes:1504 (1.4 KiB)
```

L'adresse IP, l'adresse de diffusion (*broadcast*) et le masque affectés sont affichés (interface `eth0`).

Même si ce n'est pas utile pour les premières activités, vous pouvez profiter de la connexion Internet pour mettre à jour les logiciels installés grâce au gestionnaire de paquets installé dès le départ sur la distribution Linux. Un paquetage sous Linux est le regroupement d'un logiciel principal avec tous ses fichiers et ses bibliothèques pour fonctionner. L'outil `apt`, disponible en ligne de commande dans un terminal, assure la gestion et le suivi de tous les logiciels installés, des bibliothèques et des dépendances entre applications et/ou bibliothèques.

La mise à jour complète du système s'effectue en deux temps :

```
sudo apt-get update
```

pour actualiser le cache du logiciel `apt`.

```
sudo apt-get upgrade
```

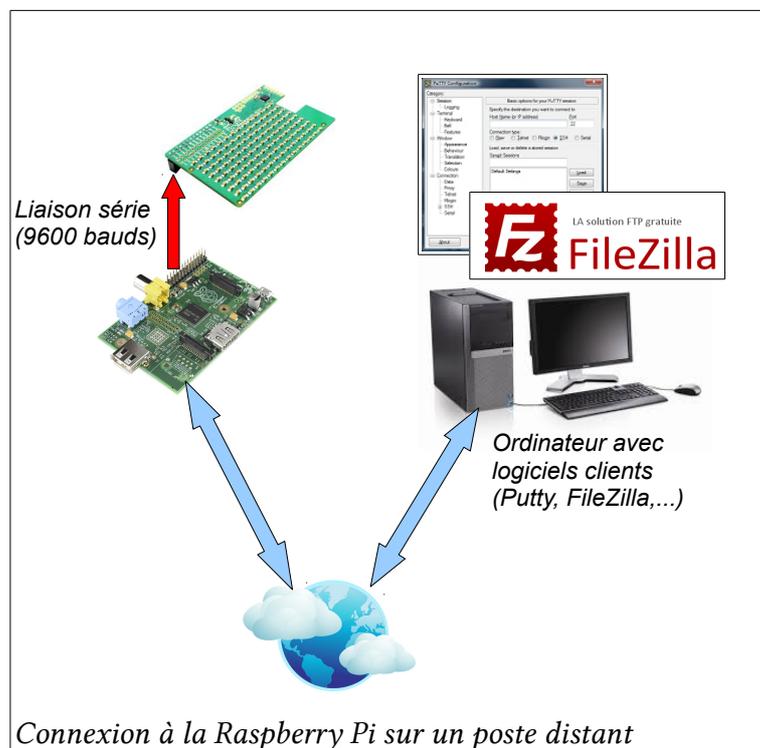
pour mettre à jour toute la distribution.

### 3. Connexion à la Raspberry Pi sur un poste distant

À partir du moment où l'adresse IP de la Raspberry Pi est connue\*, on peut s'y connecter à partir d'un autre poste informatique du réseau pour y effectuer certaines opérations à distance, voire en prendre le contrôle comme si vous travailliez sur la Raspberry Pi. L'écran, le clavier et la souris reliés à la carte embarquée ne sont alors plus utiles.

*\*Note : attention, avec le DHCP, celle-ci est attribuée pour un temps limité et peut changer d'une session à l'autre, ce qui oblige dans ce cas à brancher au moins un écran à chaque démarrage pour visualiser l'IP attribuée. Il existe bien sûr des alternatives :*

- attribuer une IP fixe, à voir avec les services informatiques du lycée. Il vous faudra pour cela modifier le fichier de configuration /etc/network/interfaces (voir par exemple [Attribution d'un IP fixe](#));
- sinon, on peut utiliser un logiciel qui scanne une plage d'adresse IP tel [Advanced IP Scanner](#) mais la recherche peut prendre du temps.



Sur un poste distant sous Windows, une simple commande `ping` tapée dans une console d'exécution permet de vérifier que la connexion est bien établie :

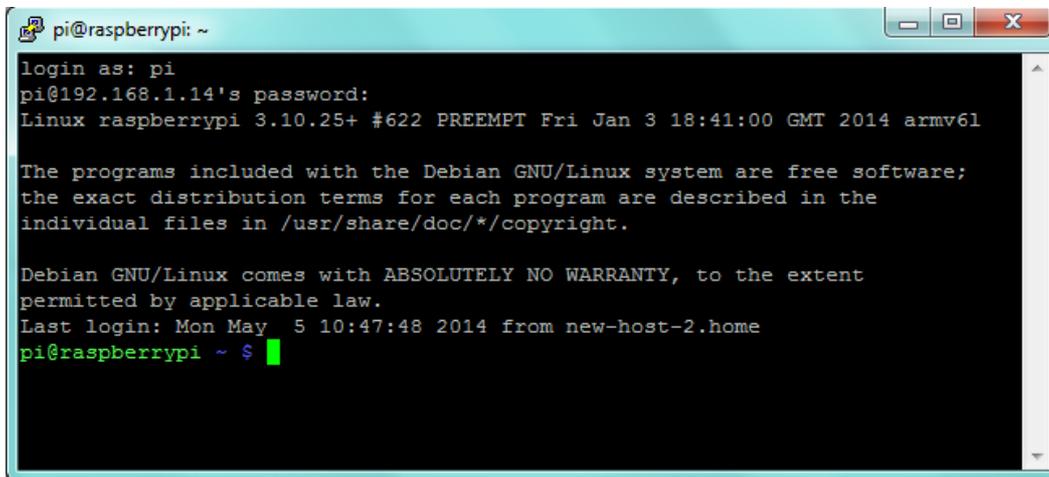
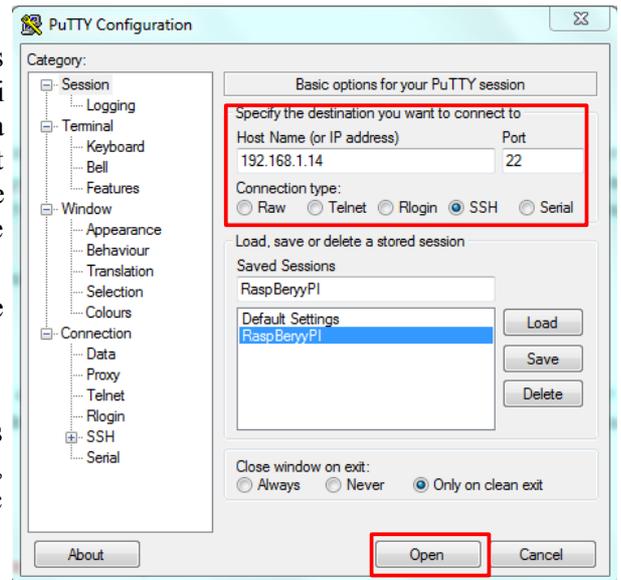
```
C:\Windows\system32\cmd.exe
C:\Users>ping 192.168.1.14
Envoi d'une requête 'Ping' 192.168.1.14 avec 32 octets de données :
Réponse de 192.168.1.14 : octets=32 temps=691 ms TTL=64
Réponse de 192.168.1.14 : octets=32 temps=199 ms TTL=64
Réponse de 192.168.1.14 : octets=32 temps=122 ms TTL=64
Réponse de 192.168.1.14 : octets=32 temps=45 ms TTL=64
Statistiques Ping pour 192.168.1.14:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 45ms, Maximum = 691ms, Moyenne = 264ms
C:\Users>
```

### 3.1 Putty

Putty (<http://www.putty.org/>) est un logiciel Windows qui permet de prendre le contrôle à distance du Raspberry Pi en utilisant le protocole SSH (*Secure Shell*). Si vous utilisez la distribution Linux proposée avec la Pi-Lite, un client SSH est activé par défaut (voir dans l'utilitaire `raspi-config`). Le dialogue s'effectue uniquement dans une console en mode texte.

Il se présente sous la forme d'un simple fichier exécutable (`putty.exe`).

Cet outil peut être intéressant pour exécuter des commandes Linux, exécuter des programmes Python, gérer des fichiers, etc. mais ne permet pas une prise de contrôle avec l'environnement de bureau LXDE en mode graphique.

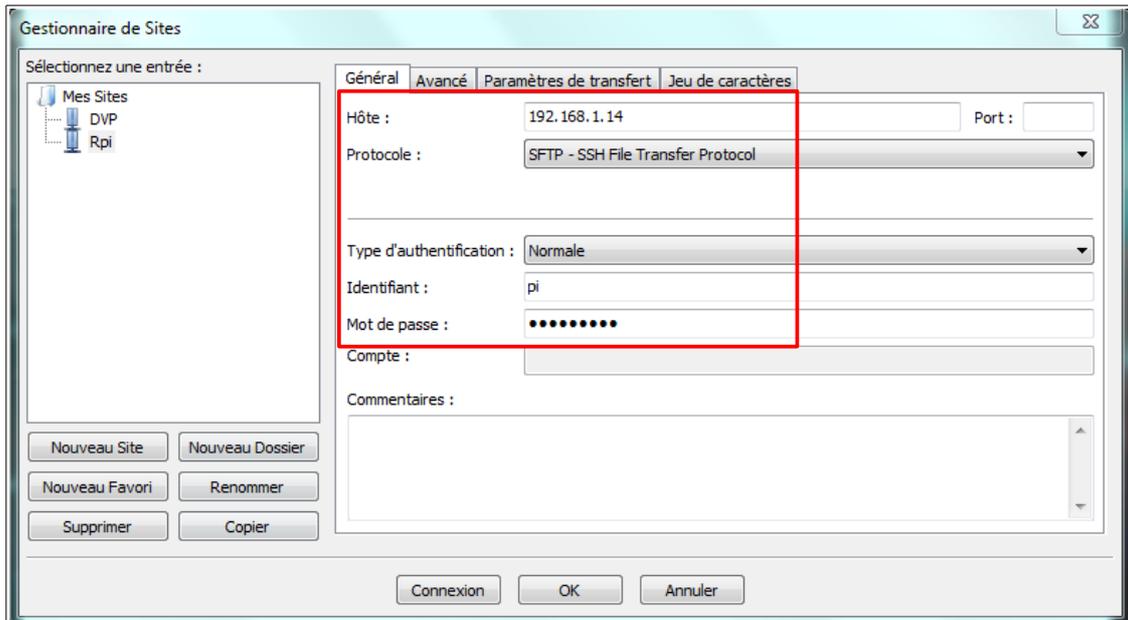


### 3.2 FileZilla

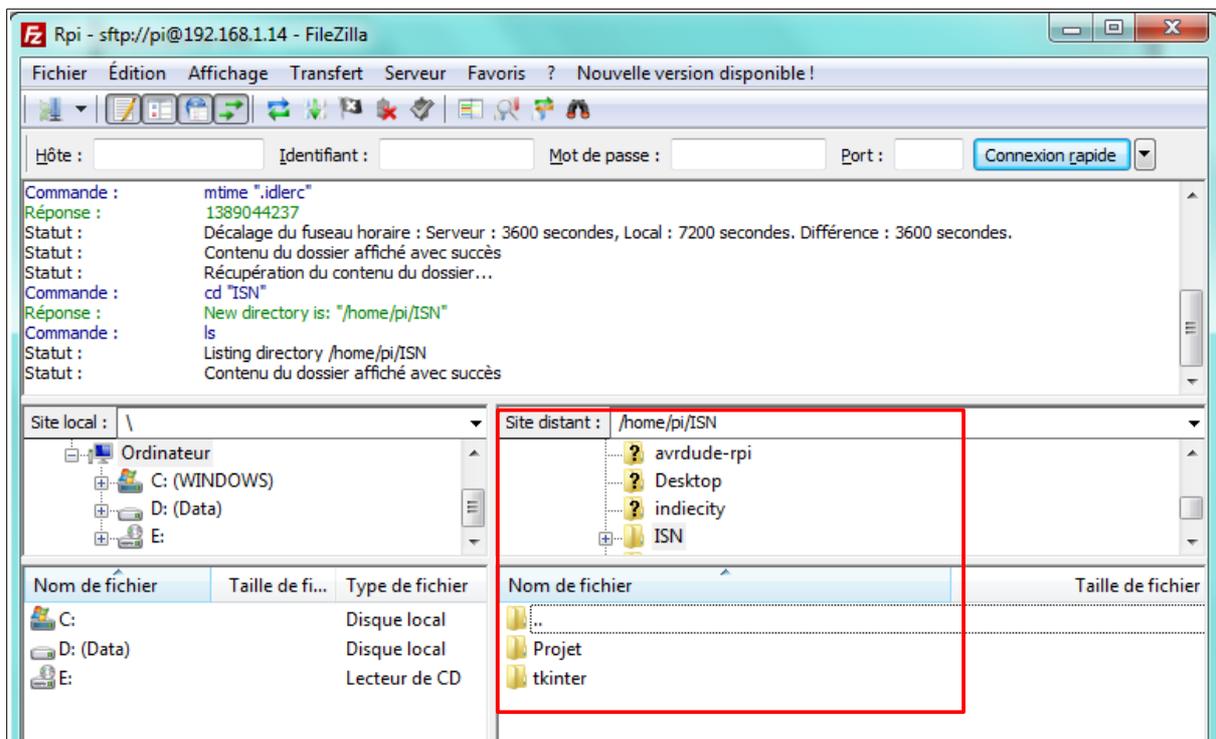
FileZilla est un client FTP (*File Transfer Protocol*) libre qui permet de télécharger des fichiers sur un serveur. Il propose une interface graphique à la façon d'un gestionnaire de fichiers et prend en charge le protocole SSH.

Site officiel : <https://filezilla-project.org/>, suivre les indications après avoir cliqué sur le lien « Download FileZilla Client ».

Rendez-vous dans le « Gestionnaire de Sites » pour créer la connexion :



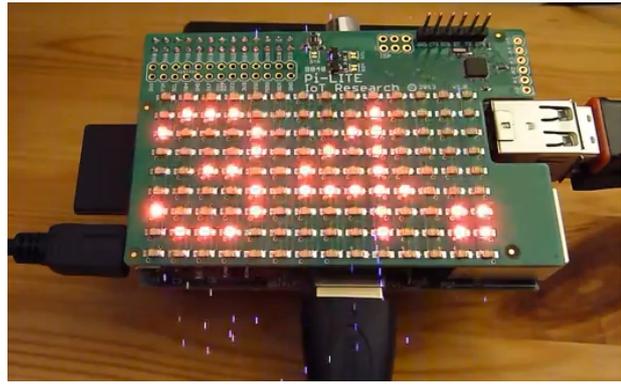
Une fois connecté à la Raspberry Pi, vous pouvez y déposer vos fichiers par simple glisser-déposer à la souris :



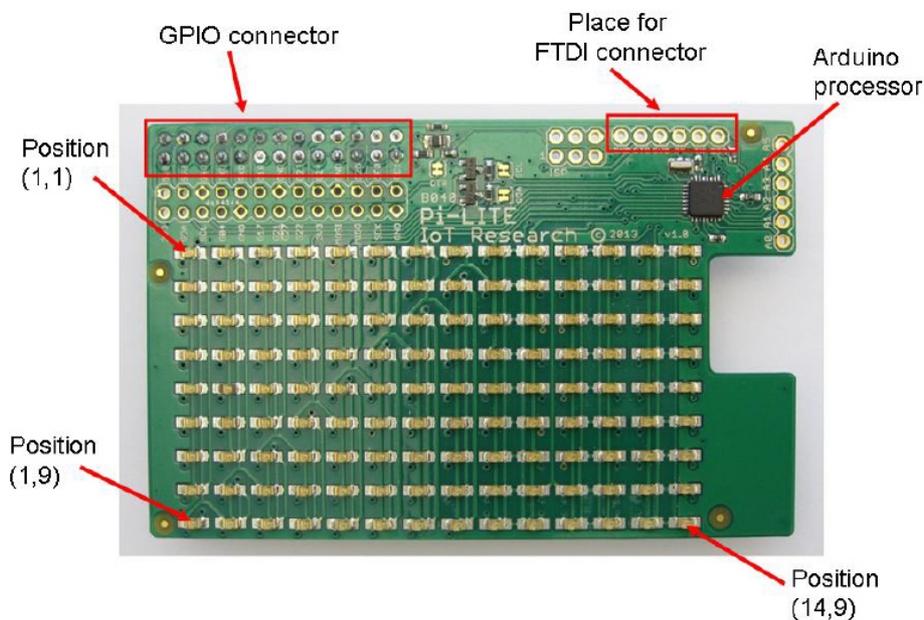
Sur un PC distant, l'élève peut préparer et mettre au point son programme Python avant de le transférer sur la Raspberry grâce à FileZilla. Il peut ensuite lancer son exécution dans une console Putty.

Avec quelques précautions, il est possible d'ouvrir plusieurs sessions avec Putty et donc faire travailler plusieurs élèves à distance sur une même carte Raspberry Pi.

## 4. Mise en œuvre de la Pi-Lite



L'intérêt de ces cartes à microprocesseur embarquées réside également dans ces broches d'entrée/sortie (les GPIO pour *General Purpose Input Output*) qui permettent la connectivité avec le monde extérieur et donc de créer ou prototyper de véritables objets connectés au monde physique pour piloter des systèmes d'éclairage à LED, ampoules ou des actionneurs électriques en robotique, mesurer des températures, etc.



La Pi-Lite, une matrice de LED dédiée à la réalisation de journal lumineux, fait donc partie de ces objets reliés aux GPIO de la Raspberry Pi que nous pouvons piloter par programmation. La distribution Debian sur le site de Ciseco comporte d'emblée logiciel et pilote pour accéder à la liaison série (broches GPIO Tx et Rx) et communiquer les informations à la Pi-Lite. La documentation du constructeur (en anglais) est disponible à partir de l'adresse : <http://shop.ciseco.co.uk/pi-lite-lots-of-leds-for-the-raspberry-pi-0805-red/>

### 4.1 Communication série Raspberry Pi – Pi-Lite

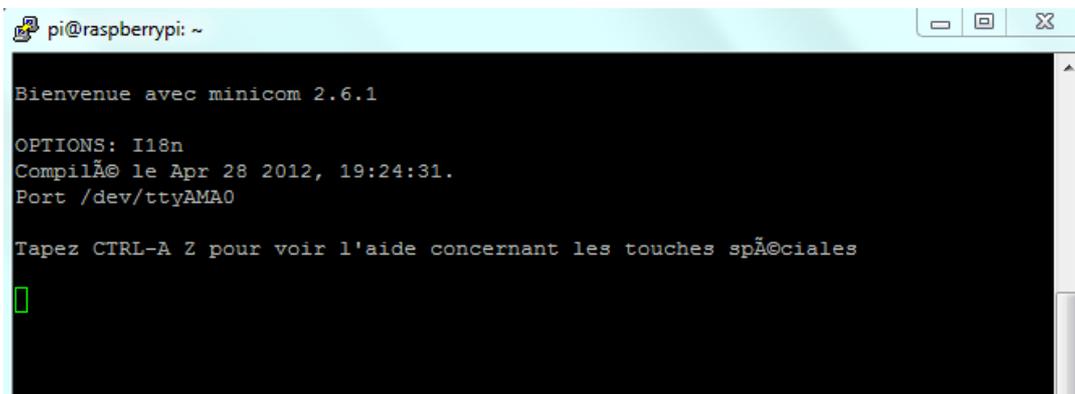
Les signaux numériques envoyés par la Raspberry Pi (via la GPIO nommée Tx) sont compatibles avec le protocole RS-232 (dont les niveaux de tension sont adaptés, si le niveau logique « 0 » est normalement à +12 V et le niveau logique « 1 » à -12 V dans le protocole RS-232, ils sont adaptés à des niveaux de tension compatibles avec le processeur Broadcom à respectivement 0 V et 3,3 V).

Pour plus de détails sur le protocole RS-232, vous pouvez consulter [la ressource ISN sur Eduscol](#) : **Communication par transmission série RS232** et y adapter les activités.

L'outil minicom permet d'envoyer sur le port série les codes ASCII des caractères saisis au clavier.

Dans un terminal (sur un poste avec Putty afin de contrôler le journal lumineux à distance, l'effet est garanti), il faut rentrer la commande :

```
minicom -b 9600 -o -D /dev/ttyAMA0
```



Le nom du port est `/dev/ttyAMA0`, la vitesse de transmission est de 9600 bauds.

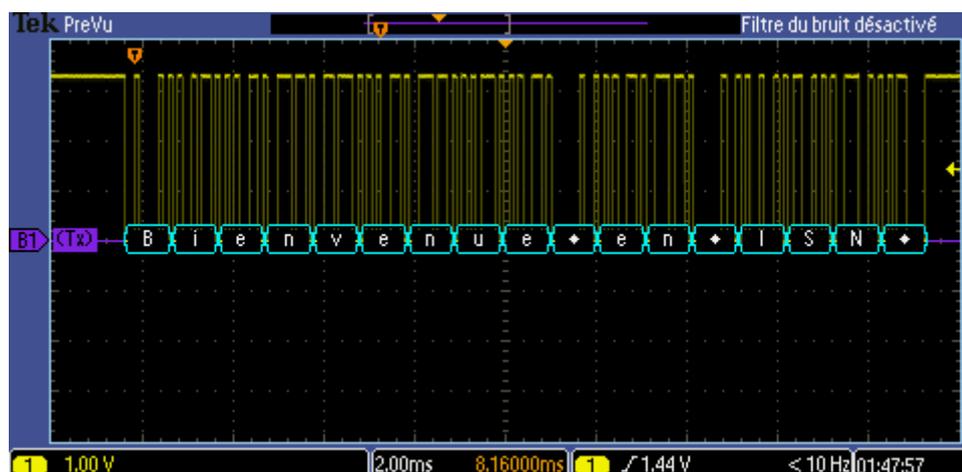
On peut quitter à tout moment l'utilitaire par la combinaison de touches : CTRL-A et x.

À partir de là, la saisie de caractères dans le terminal se fait à l'aveugle mais tout caractère envoyé sur le port série (hormis quelques caractères spéciaux dont les lettres accentuées) est affiché sur la matrice Pi-Lite en défilant de droite à gauche.

En fait, c'est le logiciel implanté dans le microcontrôleur (un AVR de chez ATMEL) de la carte Pi-Lite qui prend en charge les codes ASCII réceptionnés pour contrôler les LED et l'animation de l'affichage, ceci en toute transparence pour l'utilisateur. Par défaut, le traitement programmé du microcontrôleur consiste donc à faire défiler les caractères reçus sur sa broche de réception Rx à la façon d'un journal lumineux.



Des activités de décodage de trames RS-232 peuvent être réalisées. La figure ci-dessous montre à l'oscilloscope (équipé d'un module de décodage de trames) la trame réceptionnée après envoi de la séquence « Bienvenue en ISN » :



Bien entendu, des traitements plus complexes peuvent être réalisés par la Pi-Lite en envoyant des séquences de codes ASCII spécifiques sur le port série.

La séquence commence alors toujours par `$$$`, suivi de la commande puis d'un appui sur <Entrée> (ou <CR> pour *Carriage Return*, code ASCII=13) pour valider la séquence.

La documentation du constructeur donne l'ensemble des commandes disponibles. Par exemple, en tapant à l'aveugle dans minicom les séquences suivantes :

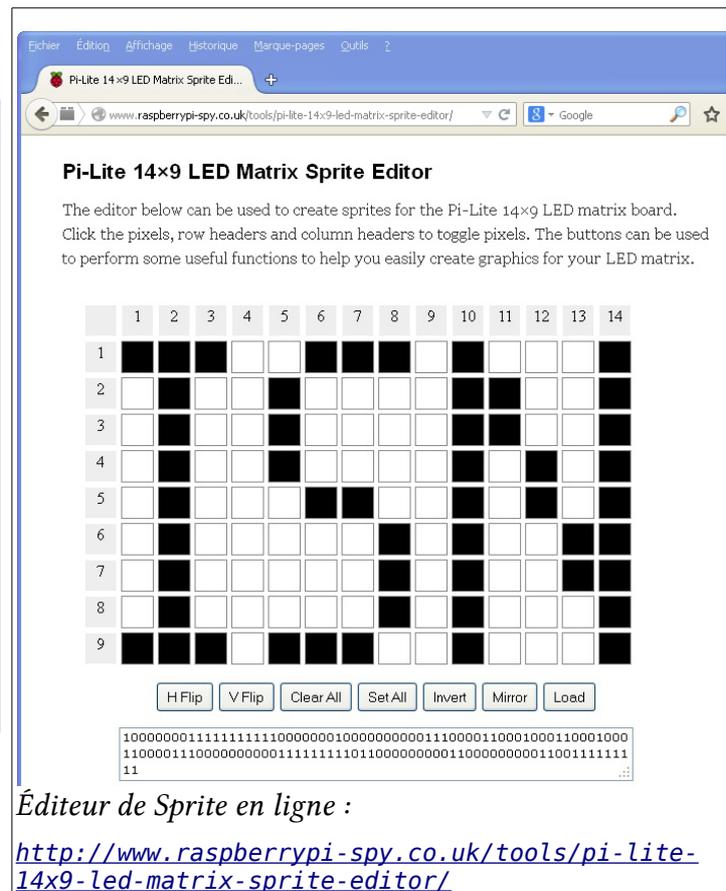
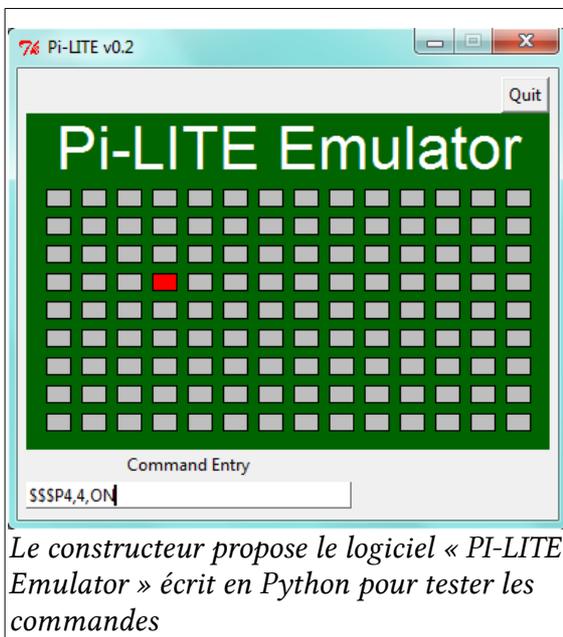
- `$$$SPEED200`, puis dans la foulée vous appuyez sur la touche <Entrée> pour terminer la séquence, vous réglez la vitesse de déroulement du texte (1=très rapide, 1000=très lent, 80 par défaut) ;

- `$$$ALL,ON`, suivi d'un <Entrée> permet d'allumer toutes les LED de la matrice. `$$$ALL,OFF`, suivi d'un <Entrée> permet de les éteindre ;

- `$$$P4,4,ON`, suivi d'un <Entrée> permet d'allumer le pixel de coordonnées (4,4). `$$$P4,4,OFF`, suivi d'un <Entrée> permet de l'éteindre ;

- `$$$F100000001111111111100000001000000000111000011000100011000100011000011100000000000111111111011000000000110000000001100111111111`, suivi d'un <Entrée> dessine un [sprite](#) « ISN » ;

- etc.



## 4.2 Un premier programme Python « Hello World »

Le premier programme Python (en 2.7) de démonstration ci-dessous montre comment utiliser la librairie standard serial et la fonction `serial.write()` pour envoyer des messages au port série et piloter l'affichage de la matrice de LED.

```
#!/usr/bin/env python

import serial
import time
import sys

# Definition des messages
```

```

# \r=Carriage Return
message1 = "Hello World!\r"
message2 = "Bienvenue en ISN!\r"

# Configuration du port serie
s = serial.Serial()
s.baudrate = 9600
s.timeout = 0
s.port = "/dev/ttyAMA0"

try:
    # Ouverture du port serie
    s.open()
except serial.SerialException, e:
    # Le port serie ne s'ouvre pas
    sys.stderr.write("Impossible d'ouvrir le port serie %r: %s\n" % (s.port, e))
    sys.exit(1)

print "Port serie OK"

# Mettre toutes les LED de la matrice à OFF
s.write("$$$ALL,OFF\r")

# Envoyer le 1er message sur le port serie
print message1
s.write(message1)

# delai, le temps que le message deroule
time.sleep(6)

# Envoyer le 2e message sur le port serie
print message2
s.write(message2)

# delai, le temps que le message deroule
time.sleep(10)

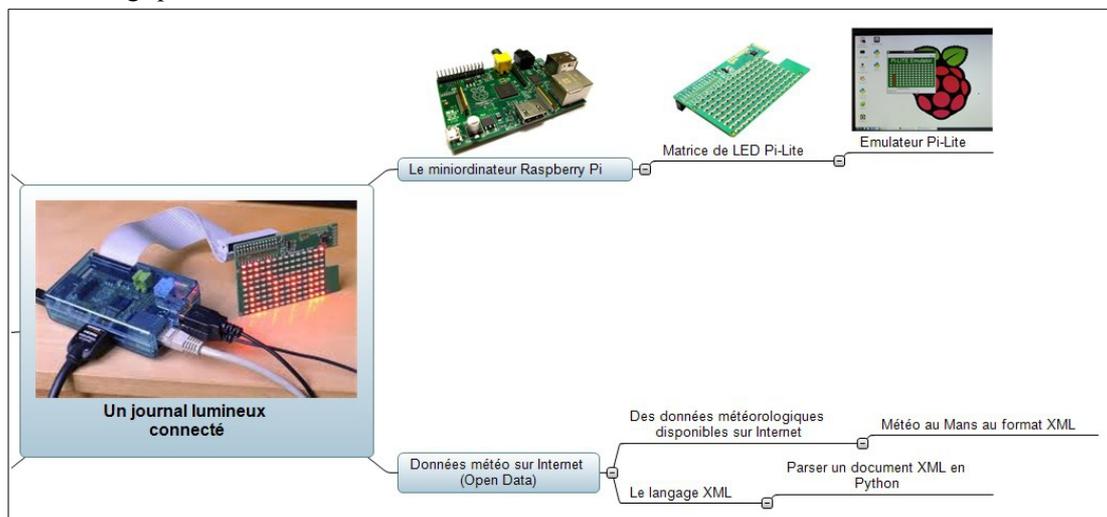
print "Fin des messages, au revoir !"

```

## 5. Activité/Projet Raspberry Pi – un journal lumineux connecté

[openweathermap.org](http://openweathermap.org) est un service libre et gratuit qui offre des données météorologiques et même des prévisions pour les développeurs de services web et d'applications mobiles. Il s'inspire en cela du modèle [OpenStreetMap](http://OpenStreetMap) en ce qui concerne les données cartographiques.

Les sources de ces données météorologiques sont un agrégat provenant de différentes sources officielles de diffusion météorologique.

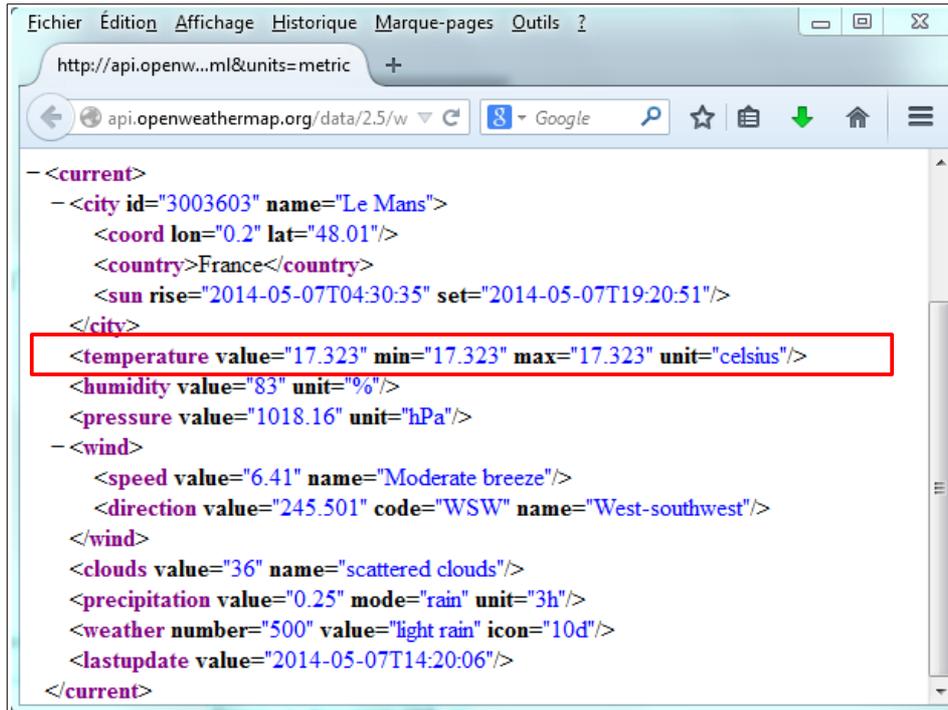


Ce projet consiste à mettre au point un prototype de journal lumineux affichant des données météorologiques (température, état du ciel, etc. d'une ou plusieurs villes par exemple) en continu à partir de données disponibles et régulièrement mises à jour sur Internet.

## 5.1 Des données au format structuré XML

L'API proposée par OpenWeatherMap permet de récupérer les données météorologiques de n'importe quelle ville dans le monde, notamment au format structuré XML. La ville peut être choisie grâce à un identifiant, ses coordonnées géographiques ou simplement par son nom. Par exemple, pour obtenir les dernières données de la ville du Mans, on saisira l'adresse suivante dans un navigateur :

<http://api.openweathermap.org/data/2.5/weather?q=Le-Mans&mode=xml&units=metric>



Voici comment, en ligne de commande, récupérer le code XML en langage Python 2.7 avec la librairie `urllib2` :

```
>>> import urllib2
>>> url = 'http://api.openweathermap.org/data/2.5/weather?q=Le-Mans&mode=xml&units=metric'
>>> file = urllib2.urlopen(url)
>>> data = file.read()
>>> file.close()
>>>
>>> print (data)
<?xml version="1.0" encoding="utf-8"?>
<current>
  <city id="3003603" name="Le Mans">
    <coord lon="0.2" lat="48.01"/>
    <country>France</country>
    <sun rise="2014-05-07T04:30:35" set="2014-05-07T19:20:51"/>
  </city>
  <temperature value="17.323" min="17.323" max="17.323" unit="celsius"/>
  <humidity value="83" unit="%"/>
```

```

<pressure value="1018.16" unit="hPa"/>
<wind>
  <speed value="6.41" name="Moderate breeze"/>
  <direction value="245.501" code="WSW" name="West-southwest"/>
</wind>
<clouds value="36" name="scattered clouds"/>
<precipitation value="0.25" mode="rain" unit="3h"/>
<weather number="500" value="light rain" icon="10d"/>
<lastupdate value="2014-05-07T14:35:09"/>
</current>

```

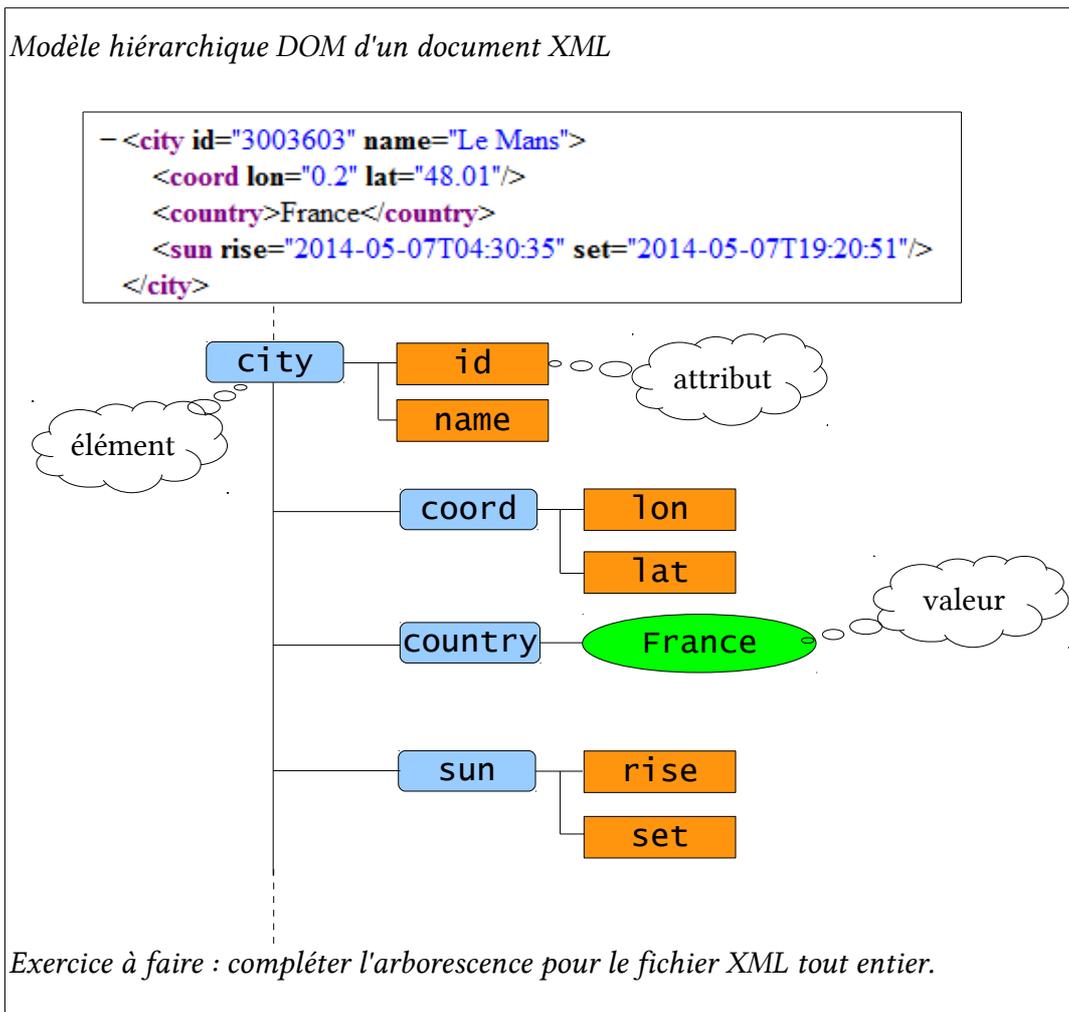
La variable `data` est une chaîne de caractères difficilement exploitable dans un premier temps.

## 5.2 Analyse du fichier XML

Afin d'exploiter la donnée texte brute en langage XML, nous exploitons le DOM (ou *Document Object Model*) [recommandé par le W3C](#) afin d'analyser le XML (de *parser* le XML dans le jargon informatique) et y extraire les données pertinentes comme la température.

Un « parseur » exploitant la technologie DOM, quel que soit son implémentation dans un langage de programmation, représente un document XML de façon hiérarchique sous forme d'un arbre. Le « parseur » propose alors des fonctionnalités pour parcourir l'arborescence afin d'y extraire des données ou manipuler l'arbre pour y rajouter des nœuds, etc.

Une ressource intéressante pour débiter : [L'arbre XML](#)



Une implémentation minimale en langage Python est bien sûr disponible avec la librairie [xml.dom.minidom](#)

On peut faire quelques tests en ligne de commande dans la console Python, toujours en partant de la variable `data` précédente contenant le document XML sous forme de chaîne de caractères.

```
>>> from xml.dom.minidom import parseString
>>> dom = parseString(data)
>>> xmlTag = dom.getElementsByTagName('temperature')[0]
>>> temperature = xmlTag.getAttribute('value')
>>>
>>> print (temperature)
17.323
```

**getElementsByTagName(*tagname*)** : rechercher les descendants avec un nom d'élément particulier, `getElementsByTagName(tagname)[0]` permet de se positionner sur le premier descendant trouvé.

**getAttribute(*Name*)** : retourne la valeur de l'attribut de nom `Name`.

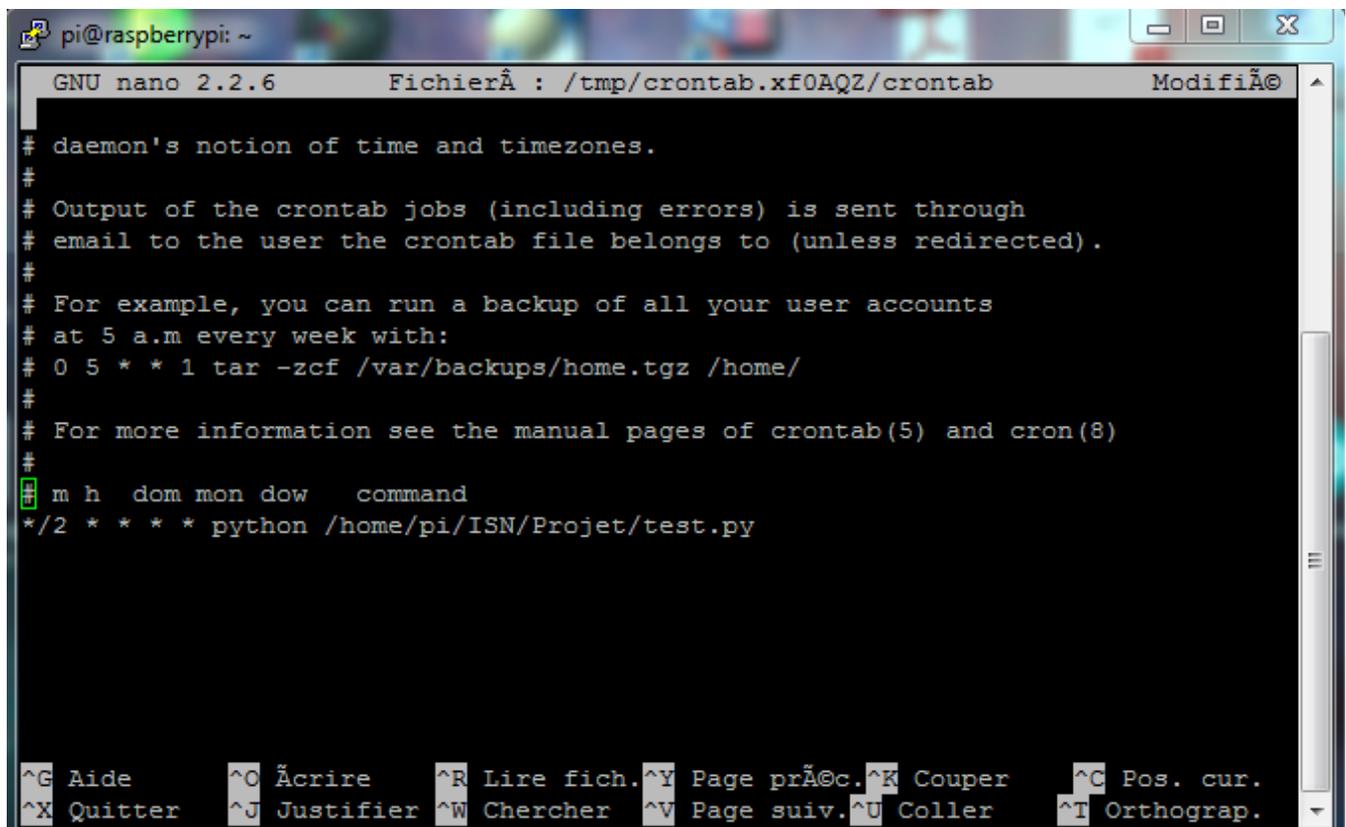
### 5.3 Exécuter un programme Python à intervalle régulier

Dans le cas d'un journal lumineux, les informations doivent être mises à jour régulièrement et défiler sur la matrice de LED à intervalle régulier. On peut faire cela en programmant des temporisations dans une boucle infinie du programme Python. Le programme est exécuté au démarrage et boucle en permanence.

Pour économiser la ressource, on préférera utiliser le service `cron`, un service Unix qui permet de lancer des tâches à des dates et heures spécifiées à l'avance ou selon des cycles périodiques.

Voir [Comment programmer l'exécution d'une tâche à intervalle régulier ?](#)

Dans une console, on lance la commande : `sudo crontab -e`



```
pi@raspberrypi: ~
GNU nano 2.2.6      Fichier : /tmp/crontab.xf0AQZ/crontab      Modifié : @
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/2 * * * * python /home/pi/ISN/Projet/test.py

^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper     ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv.^U Coller     ^T Orthograp.
```

crontab est le programme pour éditer les tables de configuration qui gèrent le service cron. La commande avec l'option -e va lancer l'édition de la table de l'utilisateur courant (Pi) avec l'éditeur de texte GNU nano. Une fois l'éditeur quitté, la table est automatiquement mise à jour selon les modifications saisies (voir l'aide avec la commande `man crontab`).

La syntaxe de la dernière ligne :

```
* /2 * * * * python /home/pi/ISN/Projet/test.py
```

permet par exemple d'évoquer le lancement du programme Python test.py (il faut indiquer son chemin complet) à intervalle régulier, toutes les deux minutes.

Il est à noter que le service cron est automatiquement exécuté au démarrage de la Raspberry Pi. La Raspberry Pi devient ainsi autonome, vous alimentez la carte reliée au réseau et le journal lumineux est fonctionnel.

## 5.4 Le cahier des charges du projet

Sous forme de projet de fin d'année, il faut bien sûr prendre en compte le temps de découverte des ressources et du matériel. Si le groupe d'élèves découvre la Raspberry Pi pour la première fois, il faut compter une à deux séances pour découvrir la carte et les outils (Chapitres 1 à 4).

Les paragraphes précédents fournissent alors les briques nécessaires à la réalisation du projet d'affichage d'informations météorologiques extraites à partir de données en langage XML proposées par le site OpenWeatherMap.

À cette étape, l'élève ou le groupe d'élèves a pu découvrir une architecture matérielle riche et originale (carte à microprocesseur embarquée) et communicante (mise en réseau, communication sérielle avec un périphérique) qu'il conviendra de mettre en valeur lors de la restitution (dossier et présentation orale).

Il convient également d'apporter de la complexité en termes de programmation Python. On propose les pistes suivantes :

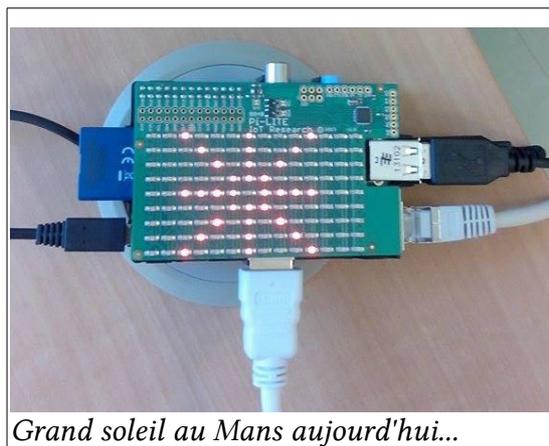
- le programme fournit les données météorologiques (température) parmi une liste des grandes capitales du monde dans un ordre aléatoire ;
- le site OpenWeatherMap retourne également un code relatif aux conditions météorologiques (ciel clair, nuageux, pluie, orage, neige, etc., voir [Weather Condition Codes](#)) :

Weather Condition Codes		
Thunderstorm		
ID	Meaning	Icon
200	Thunderstorm with light rain	[[file:11d.png]]
201	Thunderstorm with rain	[[file:11d.png]]
202	Thunderstorm with heavy rain	[[file:11d.png]]
210	Light thunderstorm	[[file:11d.png]]
211	Thunderstorm	[[file:11d.png]]
212	Heavy thunderstorm	[[file:11d.png]]
221	ragged thunderstorm	[[file:11d.png]]
230	Thunderstorm with light drizzle	[[file:11d.png]]
231	Thunderstorm with drizzle	[[file:11d.png]]
232	Thunderstorm with heavy drizzle	[[file:11d.png]]

Icon list		
Day	Night	
01d.png 	01n.png 	sky is clear
02d.png 	02n.png 	few clouds
03d.png 	03n.png 	scattered clouds
04d.png 	04n.png 	broken clouds
09d.png 	09n.png 	shower rain
10d.png 	10n.png 	Rain

```
Fichier Édition Affichage Historique Marque-pages Outils ?
http://api.openw...ml&units=metric +
api.openweathermap.org/data/2.5/w
<current>
<city id="3003603" name="Le Mans">
  <coord lon="0.2" lat="48.01"/>
  <country>France</country>
  <sun rise="2014-05-07T04:30:35" set="2014-05-07T19:20:51"/>
</city>
<temperature value="17.323" min="17.323" max="17.323" unit="celsius"/>
<humidity value="83" unit="%"/>
<pressure value="1018.16" unit="hPa"/>
<wind>
  <speed value="6.41" name="Moderate breeze"/>
  <direction value="245.501" code="WSW" name="West-southwest"/>
</wind>
<clouds value="36" name="scattered clouds"/>
<precipitation value="0.25" mode="rain" unit="3h"/>
<weather number="500" value="light rain" icon="10d"/>
<lastupdate value="2014-05-07T14:20:06"/>
</current>
```

La matrice de LED doit également afficher un pictogramme correspondant à l'ID retourné dans le XML.



- le programme donne également l'heure (locale), récupérée sur internet (voir [Mettre à l'heure le Raspberry Pi](#)) ;
- etc.

## 6. Bibliographie et sitographie

### **Pour débiter sur Raspberry Pi :**

- Raspberry Pi : déballage et installation : <http://nicolargo.developpez.com/tutoriels/raspberry-pi/deballage-installation/>
- Léa-Linux : Raspberry Pi : [http://lea-linux.org/documentations/Raspberry\\_Pi](http://lea-linux.org/documentations/Raspberry_Pi)
- Qick Start Guide : <http://www.raspberrypi.org/new-quick-start-guide/>
- Raspberry Pi : Le guide de l'utilisateur : <http://www.amazon.fr/Raspberry-Pi-Le-guide-lutilisateur/dp/2744025798>

### **Fournisseurs de matériels Raspberry Pi :**

- Kubii : <http://www.kubii.fr/fr/>
- Farnell : <http://fr.farnell.com/raspberry-pi>
- Element14 : <http://www.element14.com/community/community/raspberry-pi>

### **Matrice de LED Pi-Lite :**

- Ciseco Pi-Lite - Lots of LEDs for the Raspberry : <http://shop.ciseco.co.uk/pi-lite-lots-of-leds-for-the-raspberry-pi-0805-red/>
- **Distribution Linux Raspbian optimisée pour Raspberry Pi et Pi-Lite :** <http://www.openmicros.org/Download/> (prendre l'image la plus récente)
- Pi-Lite User Guide : <http://www.openmicros.org/index.php/articles/94-ciseco-product-documentation/raspberry-pi/280>