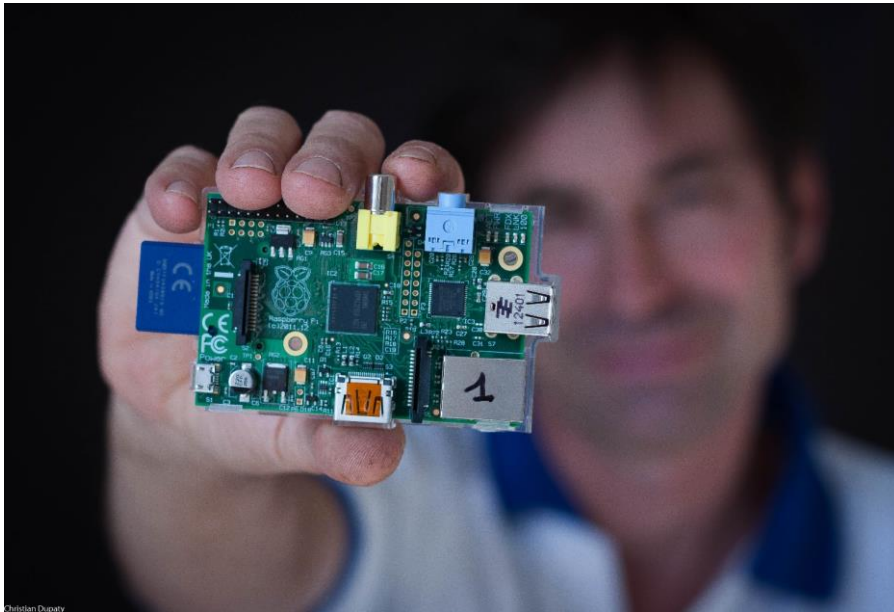


RASPBERRY PI

INSTALLATION-CONFIGURATION

INTERFACES DE COMMUNICATIONS



PRESENTATION-RESSOURCES

Christian Dupaty

BTS Systèmes Numériques

Lycée Fourcade - Gardanne

Académie d'Aix-Marseille



1) Introduction, Objectifs et Intentions

Le BTS SN (Systèmes Numériques) intègre la formation et le développement des composantes électroniques et informatiques des systèmes numériques autour de projets. Les ordinateurs embarqués sous le système d'exploitation Linux sont massivement présents dans les technologies étudiées, BOX multimédia, smartphones, liseuses, appareils photos ...

L'ordinateur Raspberry Pi constitue un support d'enseignement performant, très bon marché et disposant d'un réseau mondial de développeurs. Il possède des entrées/sorties puissantes permettant une connexion avec des capteurs et actionneurs et donc le monde physique.

Les publications sur l'ordinateur Raspberry Pi sont très nombreuses tant sur internet qu'en librairie. Ces publications sont destinées à tous les publics, il est parfois difficile d'isoler les informations recherchées. Les travaux pratiques proposés ont pour objectifs de faciliter la prise en main de l'ordinateur embarqué Raspberry Pi par les enseignants de spécialités en BTS SN et servir éventuellement de ressources pour les étudiants de BTS SN.

Dans le cadre du BTS SN, l'ordinateur Raspberry Pi peut être utilisé dans le cadre d'un mini-projet ou du projet de l'épreuve E62. Il peut également servir de support à l'apprentissage :

- De la gestion et la structure des ordinateurs sous Linux
- De l'administration à distance avec SSH
- Des protocoles SPI, I2C, UART , de la PWM.
- De la mise en œuvre de capteurs, d'actionneurs
- Des langages C, C++ et Python
- ...

Le support Raspberry Pi permet de développer toutes les capacités du référentiel du BTS SN, les TP proposés sont plus particulièrement orientés vers :

(cf : <http://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/textes/formations-bts-bts-systemes-numeriques/3321-referentiel-bts-systemes-numeriques.pdf>)

- C4.1 : câbler et/ou intégrer un matériel (EC+ IR)
ex : TP 1WIRE ou TP I2C
- C4.2 : adapter et/ou configurer un matériel (EC et IR)
ex : TP Afficheur/Camera
- C4.3 : adapter et/ou configurer une structure logicielle (EC)
ex : TP serveur WEB
- C4.3 : installer et configurer une chaîne de développement (IR)
ex : TP installation Raspberry Pi
- C4.4 : fabriquer un sous-ensemble (EC)
ex : TP 1WIRE ou TP I2C
- C4.4 : développer un module logiciel (IR)
TP à développer par le professeur
- C4.5 : tester et valider un module logiciel et matériel
- C4.5 : tester et valider un module logiciel

AVERTISSEMENT : Les TP sont destinés à la réalisation d'évaluations formatives. Ils ne peuvent pas être donnés directement aux étudiants, le professeur doit se les approprier et les adapter au contexte pédagogique particulier de chaque classe, en particulier les opérationnaliser autour de systèmes réels et/ou de cahier des charges répondant à un besoin.

**Contenus de formation :**

- Description de la carte Raspberry Pi
- Installation d'un OS
- Contrôle de la carte à distance par SSH sur PC Windows avec support WIFI
- Les commandes Linux minimum
- Configuration et installation des bibliothèques de gestion des périphériques
- Exemples mettant en œuvre les périphériques embarqués : GPIO, SPI, I2C, UART, PWM, 1WIRE
- Réalisation d'un serveur WEB embarqué avec lecture/commande de périphériques (thermomètre)
- Interface tactile Adafruit
- Acquisition vidéo, gestion d'une caméra.

Python :

Les exemples sont écrits en Python, le lecteur doit avoir des connaissances minimales dans ce langage. Python est le principal langage utilisé par la communauté internet pour le développement d'applications sur Raspberry Pi. Néanmoins le langage C est largement utilisé pour la réalisation des drivers bas niveau de gestion des E/S (<http://wiringpi.com/>).

Un très grand nombre de bibliothèques sont développées en Python pour Raspberry et disponibles gratuitement sur Internet.

Python est un langage interprété (non compilé comme le C) orienté objet de très haut niveau, il dispose d'une syntaxe (assez proche du C) qui permet une écriture et une lecture aisée et structurée des programmes.

Fonction factorielle en C	Fonction factorielle en Python
<pre>/* Fonction factorielle en C */ int factorielle(int x) { if (x < 2) { return 1; } else { return x * factorielle(x-1); } }</pre>	<pre># Fonction factorielle en Python def factorielle(x): if x < 2: return 1 else: return x * factorielle(x-1)</pre>

©Wikipedia

Pour apprendre le langage Python :

<http://fr.openclassrooms.com/informatique/cours/apprenez-a-programmer-en-python>

<http://www.framasoft.net/article1971.html>

<http://python.developpez.com/cours/TutoSwinnen/>

Pour apprendre les commandes Linux (entre autres, il y a énormément de publications sur internet):

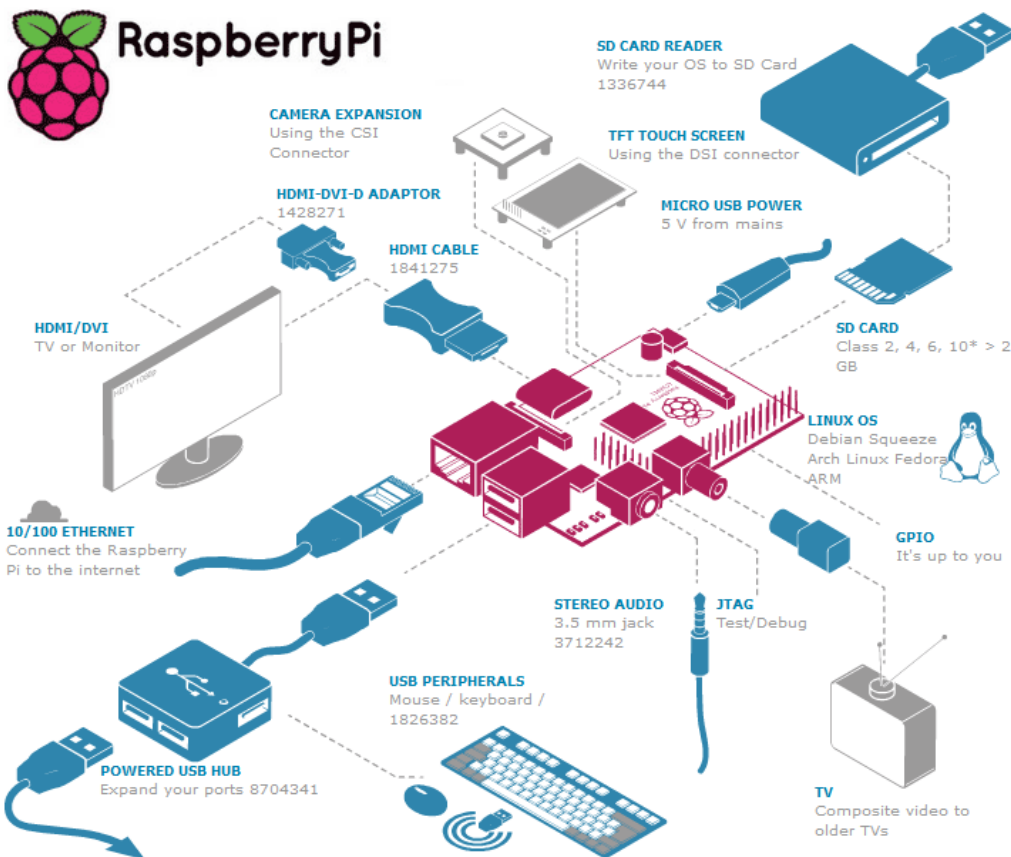
<http://fr.openclassrooms.com/informatique/linux/cours>

<http://jplu.developpez.com/tutoriels/systeme/presentation-linux/>

<http://sylvain.cherrier.free.fr/documentations/coursLinux.pdf>



2) Pour s'équiper :



© Farnell

Equipement recommandé : (environ 150€ HT)

Un PC sous Windows, Linux ou IOS pour le développement à distance par SSH.

Un accès internet, pour la récupération des bibliothèques et logiciels.

Un ordinateur Raspberry Pi type B avec carte SD 8GO (30€)

Une camera Rapsberri Pi (19€ HT) (facultatif)

Un écran tactile TFT adafruit (34€) (facultatif)

Une platine adafruit Proto Plate ADA801 (13€)

Une alimentation micro USB

soit un câble USB-micro USB, l'alimentation se fait par la câble USB d'un PC (3€)

soit une alimentation secteur autonome avec un câble micro USB (5€)

un dongle USB-WIFI pour Raspberry Pi (18€)

Un PACK cables et accessoires (25€)

un clavier USB

une souris USB

câbles et adaptateur HDMI VGA et HDMI DVI-D

un écran de préférence HDMI (*moins de 150€, achat optionnel, l'écran n'est utile que pendant la phase de configuration initiale, le PC n'étant pas utile à ce moment son écran, souris, clavier sont utilisables*)

Quelques composants (tmp102, 18B20 ...) pour les TP (moins de 10€)

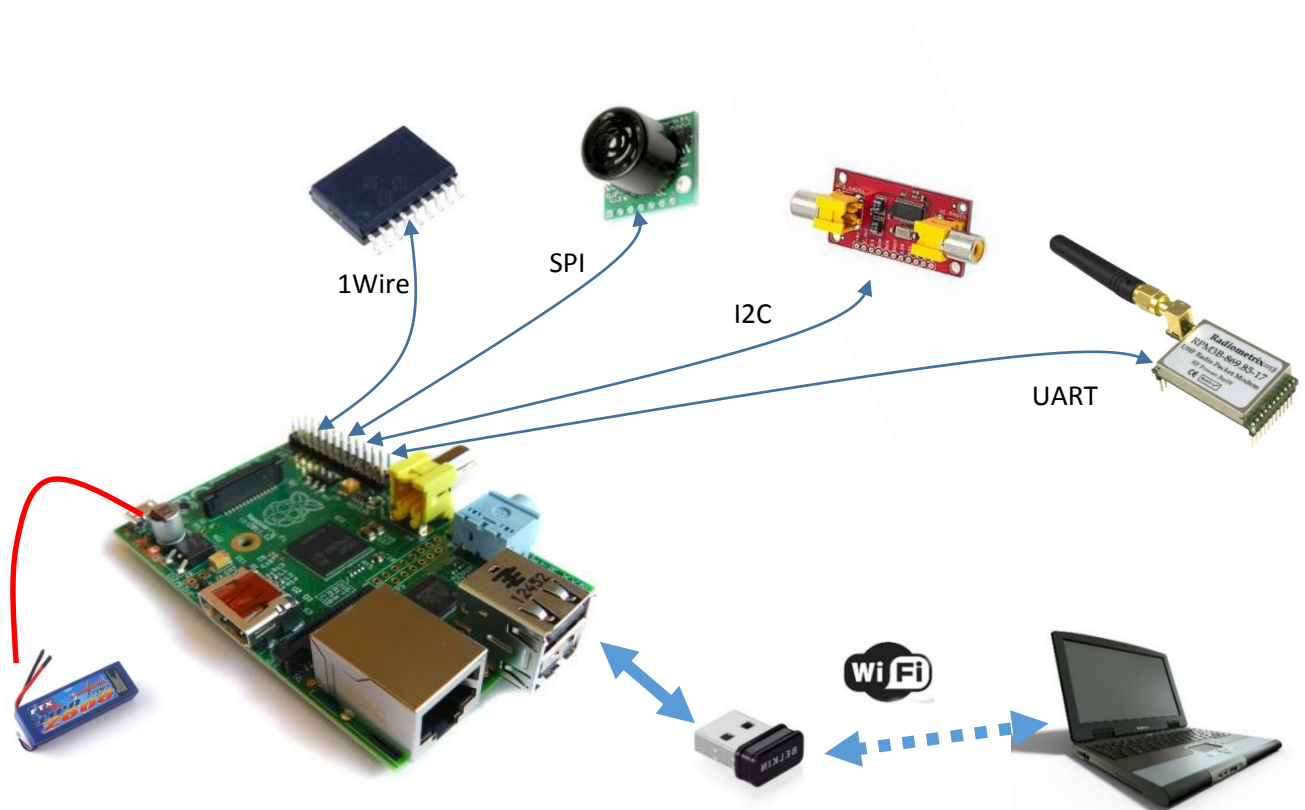


3) La carte Raspberry Pi

Raspberry Pi est un petit ordinateur sous le système d'exploitation Linux sur carte SD destiné à des applications d'informatique embarquée. Le cœur de l'ordinateur est un FPGA (Broadcom 2835) intégrant un processeur ARM11 cadencé à 700MHz et de nombreux périphériques.

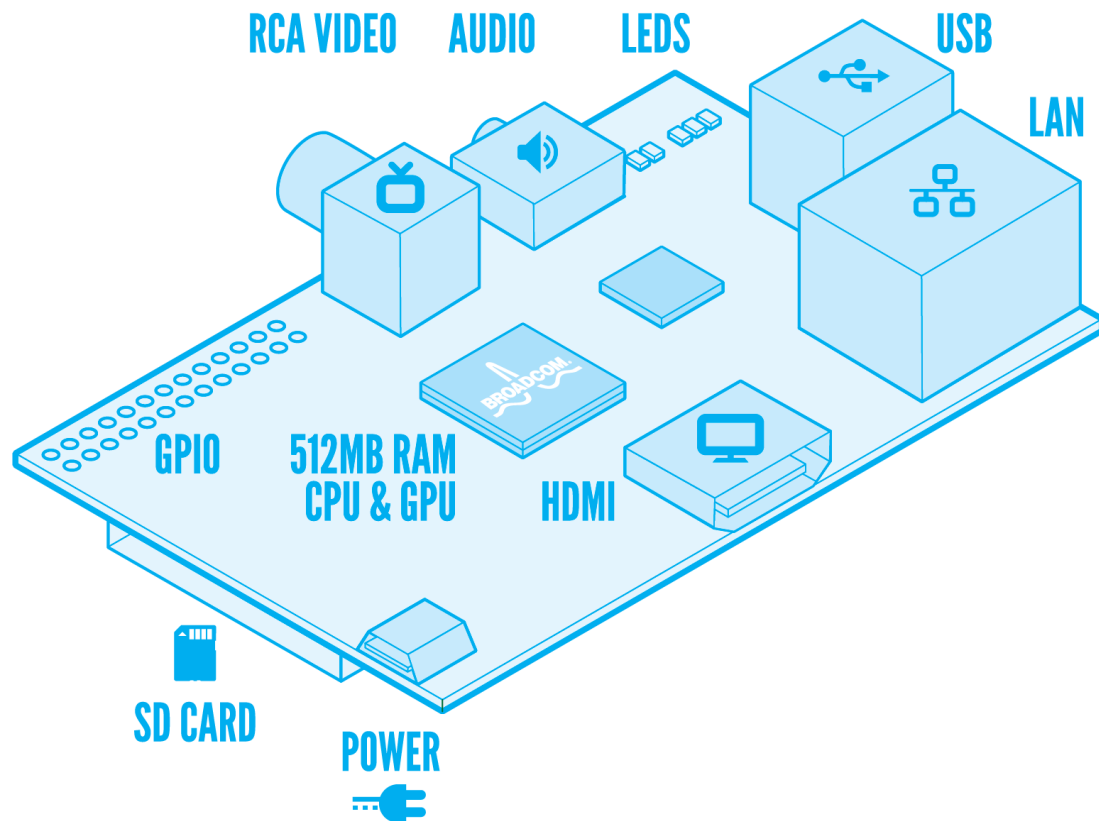
Raspberry Pi peut être directement connecté à une IHM classique, souris/clavier/écran HDMI ou vidéo composite, cependant comme tout ordinateur Linux, Raspberry Pi peut intégrer ses propres outils de développement et une interface homme-machine reposant sur SSH contrôlable depuis un autre ordinateur par Ethernet ou WIFI.

Le connecteur d'extension supporte les entrées/sorties parallèles ainsi que la plupart des bus de communication. C'est un support particulièrement économique et puissant qui peut être facilement mis en œuvre dans de petits systèmes nécessitant un accès au monde physique par des capteurs/actionneurs disposants d'interfaces numériques.





RASPBerry PI MODEL B



© <http://www.raspberrypi.org/>

- Une carte SD faisant office de disque dur contient le système d'exploitation Linux, ainsi que les programmes et fichiers de données
- Les informations techniques de la carte sont disponibles sur :
<http://www.raspberrypi.org/>
http://elinux.org/RPi_Hub
- Les TP permettent de configurer la Raspberry Pi et développer des applications de contrôle des entrées/sorties (GPIO) en langages C et Python, depuis un terminal SSH embarqué sur un ordinateur (PC, LINUX, MAC) et communiquant par Ethernet ou WIFI.



4) Le connecteur d'extension de Raspberry Pi :

L'essentiel des TP utilisent le connecteur d'extension de la carte Raspberry Pi. Il est nécessaire de câbler correctement les périphériques sur ce connecteur.

Connecteur GPIO

Le connecteur GPIO supporte les GPIO (entrées/sorties binaires) mais également les sorties PWM, les périphériques de communication (UART, I2C, SPI) et les alimentations 5v et 3V3.

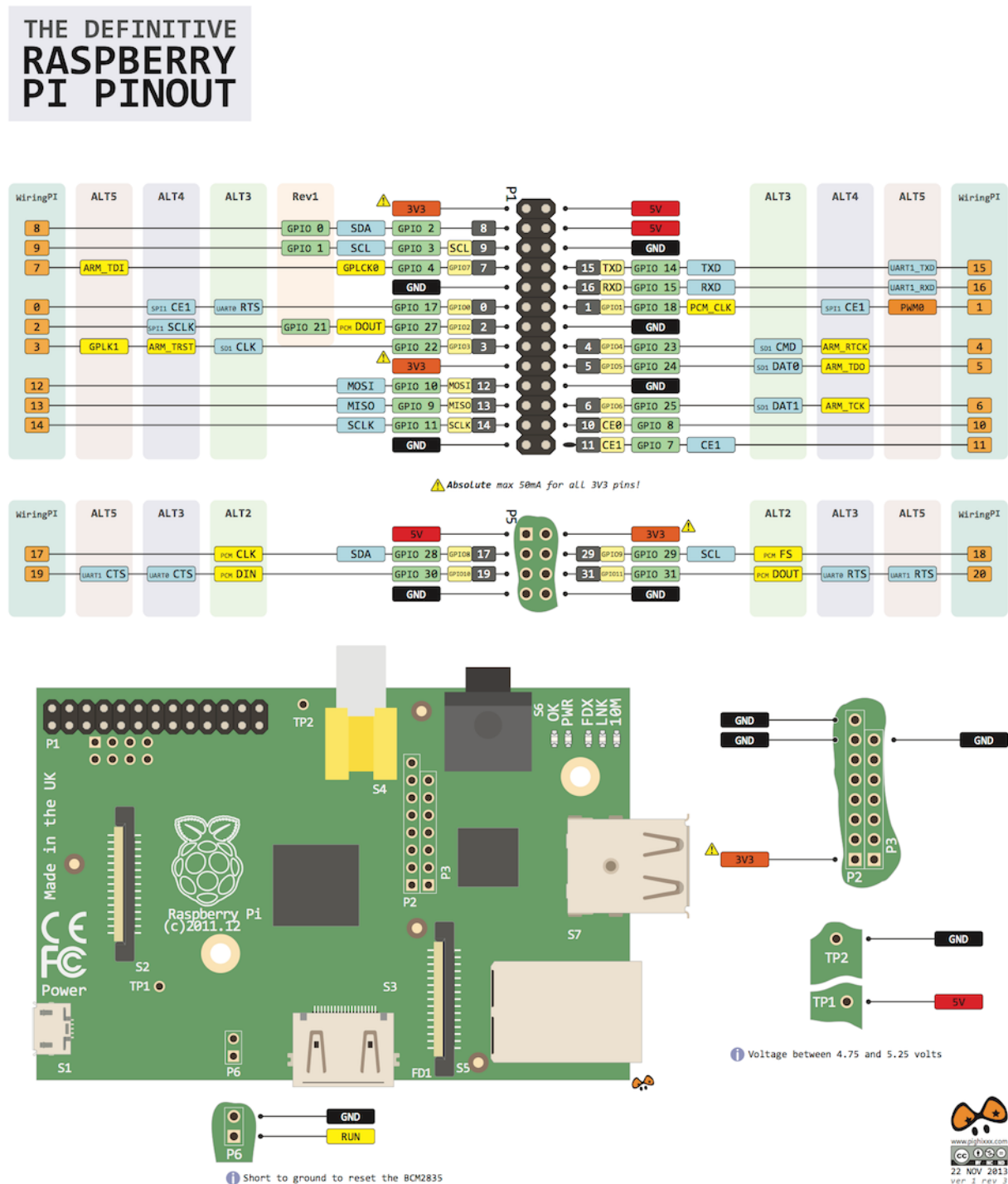
Les broches peuvent avoir des fonctions différentes suivant qu'elles sont activées en tant que GPIO ou périphérique de communication. Certaines possèdent des résistances de pull-up donnant un bit à 0 dominant et un bit à 1 récessif. (ex I2C)

© <http://www.raspberrypi.org/>





Schéma ressource pour les connexions, les connecteurs de la Rasperry Py :



DataSheets et documentation :

http://elinux.org/RPi_Low-level_peripherals

http://elinux.org/RPi_BCM2835_GPIOs



5) LINUX mini (les commandes indispensables)

(<http://www.misfu.com/commandes-unix.html>)

- Raspberry Pi est un ordinateur fonctionnant sous le système d'exploitation LINUX-DEBIAN adapté à la Raspberry Pi sous le nom RASPBIAN, il est indispensable pour le configurer et l'utiliser de connaître quelques commandes Linux en mode console
- Une aide exhaustive pour chaque commande est disponible sous linux par :
man nom-de-la-commande

sudo (substitute user do)	Entête donnant les droits de super utilisateur (root) Les programmes PYTHON doivent être lancés avec sudo
shutdown -r now	redémarrer
shutdown -h now	éteindre
ls	voir le contenu du dossier
ls -l	voir le contenu du dossier et les droits sur les fichiers
pwd	afficher le nom du dossier courant
cd dossier	descendre dans l'arborescence
cd ..	remonter dans l'arborescence
nano nom_fichier	éditer un fichier
exit	se déconnecter de la machine
cp source destination	copier un fichier
mkdir nom	créer un dossier
mv source destination	déplacer un fichier/dossier
mv nom1 nom2 (dans le même dossier)	renommer un fichier
rm fichier et rm -r dossier	supprime un fichier ou un dossier
chmod +x nom_du_fichier	Rendre un fichier exécutable
./nom_du_fichier	Exécuter un fichier

- Pour éviter de taper à chaque fois **sudo**, passez en mode administrateur : **sudo -i**
passer en shell administrateur : **sudo -s** (déconseillé) pour sortir du mode shell : **exit**
- Linux utilise beaucoup les scripts, de petits fichiers de type texte qui permettent l'enchaînement de commandes systèmes. Il existe plusieurs types de scripts systèmes.
Les fichiers BASH sont très utilisés sur Raspberry Pi

Exemple : `sudo nano essai`

```
# !/bin/bash    indique un fichier de type bash
echo « Bonjour, voici le dossier courant »
ls
```

ctrl-x Y pour enregistrer et quitter. Il suffit de lancer le fichier essai pour l'exécuter.

Exemple d'écriture d'un script en PYTHON :

```
sudo nano monscript.py
```

Lancer l'exécution d'un script PYTHON

```
sudo python monscript.py (ctrl-c pour l'arrêter)
```

Pour rendre un programme PYTHON exécutable sans avoir à lancer la commande python, le script doit contenir la ligne `#!/usr/bin/python3.2` (ici pour indiquer qu'il s'agit d'un script Python 3.2).

```
sudo chmod +x monscript.py
```

L'exécuter

```
sudo ./monscript.py
```