

Introduction

A) Historique et gamme

Nous n'allons pas faire un historique complet et une description détaillée des 4 principaux types de modèles proposés actuellement, mais présenter succinctement les principaux éléments selon notre point de vue. Pour plus de détail on pourra aller sur la documentation officielle : <https://www.raspberrypi.org/help/faqs/>

Le Raspberry pi a vu le jour en 2011 et en 2016, 3 gammes principales de produits existent actuellement (2016) :

- ✓ *Raspberry pi V1 (B+), SOC Broadcom BCM2835 700MHz (processeur ARMv6) , RAM 512Mo, 2xUSB2.0, protection alimentation 5V /2A*
- ✓ *Raspberry pi V2, SOC Broadcom BCM2836 900MHz (processeur ARMv7 – A7- 4 coeurs) ; RAM: 1GB, 4x USB 2.0, protection alimentation 5V /2A,*
- ✓ *Raspberry pi V3, SOC Broadcom BCM2837 1.2GHz (processeur ARMv8 – A53- 4 coeurs) RAM 1GB, 4x USB 2.0, integrated 802.11n wireless LAN, and Bluetooth Low Energy 4.1, protection alimentation 5V /2A,*

En mode normal, le courant consommé (pas de clavier, souris, écran) est de l'ordre de 200mA à 300mA pour les V1 et V2 et grimpe à plus de 500mA pour la V3 (à cause des chipsets wifi/Bluetooth) et du processeur plus gourmand en énergie.

La taille de la carte est la même pour les 3 versions. La version pi zero est deux fois plus petite et consomme moins (un peu moins puissante que la V2).

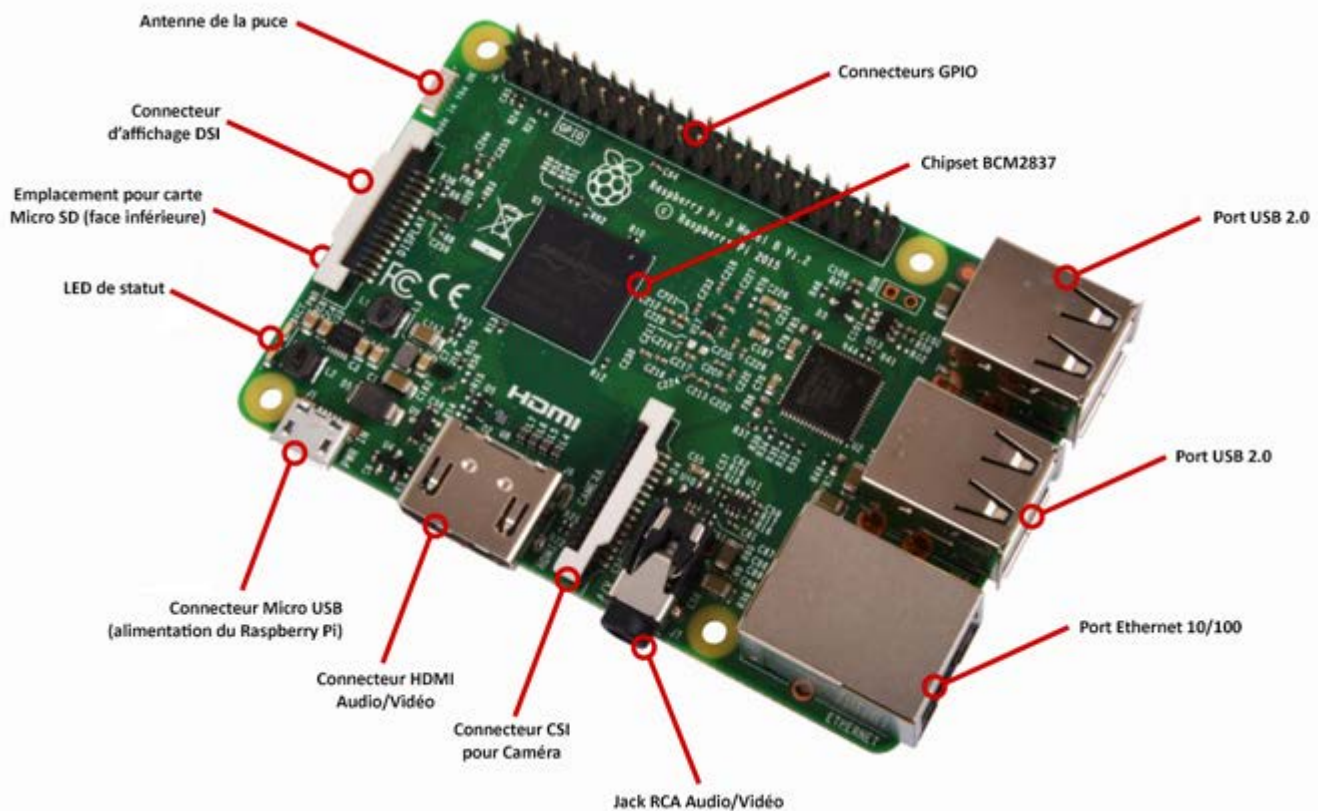
Au niveau des puissances de calculs, on peut résumer en disant que la version 3 est 50% plus rapide que la 2 qui est 2x plus rapide que la version 1 (B+). Si l'on utilise les 4 processeurs (version 2 et 3), la vitesse de calcul augmentera évidemment (compilation par exemple). On peut noter que la version 3 a un processeur qui chauffe, ce qui devrait être corrigé avec la version 4 prévue en 2017.

		Pi1 (B+)	Pi2 B	Pi3 B (Amps)	Zero (Amps)
Boot	Max	0.26	0.40	0.75	0.20
	Avg	0.22	0.22	0.35	0.15
Idle	Avg	0.20	0.22	0.30	0.10
Video playback (H.264)	Max	0.30	0.36	0.55	0.23
	Avg	0.22	0.28	0.33	0.16
Stress	Max	0.35	0.82	1.34	0.35
	Avg	0.32	0.75	0.85	0.23

Figure 1:consommation des différents modèles PI (<https://www.raspberrypi.org/help/faqs/>)

B) Présentation (modèle B+)

La présentation ci-dessous présente les principaux éléments d'une carte Raspberry version B+, mais applicable aussi pour les versions suivantes.



Le Raspberry est un mini-ordinateur (le modèle B+ est équivalent à un pentium 300MHz) possédant donc une sortie vidéo HDMI mais aussi RCA. Depuis le début aucun connecteur VGA n'a été prévu sur les Raspberry. Il est donc possible de connecter au travers du RCA (sortie jaune) un moniteur vidéo.



Le connecteur HDMI peut quant à lui être utilisé pour se connecter à un téléviseur HD, ou bien à un écran d'ordinateur en utilisant un câble HDMI-DVI D. Le connecteur HDMI fournit un signal digital haute définition (Full HD 1920x1080) ainsi qu'une sortie son. Le câble HDMI DVI-D ne prend que le signal vidéo, il faudra alors utiliser le connecteur son (noir) 3,5mm pour le connecter sur des haut-parleurs.



Un connecteur DSI permet quant à lui de brancher un afficheur de type dalle tactile.



Les connecteurs usb, permettent d'ajouter un clavier et une souris, ou bien un dongle pour clavier et souris sans fil permettant à notre Raspberry de se faire passer pour un mini-ordinateur Linux.

Pour terminer sur cette mini-présentation du raspberry, n'oublions évidemment pas l'essentiel : la SD CARD. En effet la Raspberry est fournie sans disque dur ou mémoire non volatile. Un connecteur SD Card est prévu pour accueillir votre OS Linux (Raspbian) qu'il faudra évidemment écrire sur une SD Card de taille minimale 2Go. Le Raspberry est alimenté au travers d'un câble micro-USB.



Raspberry Pi

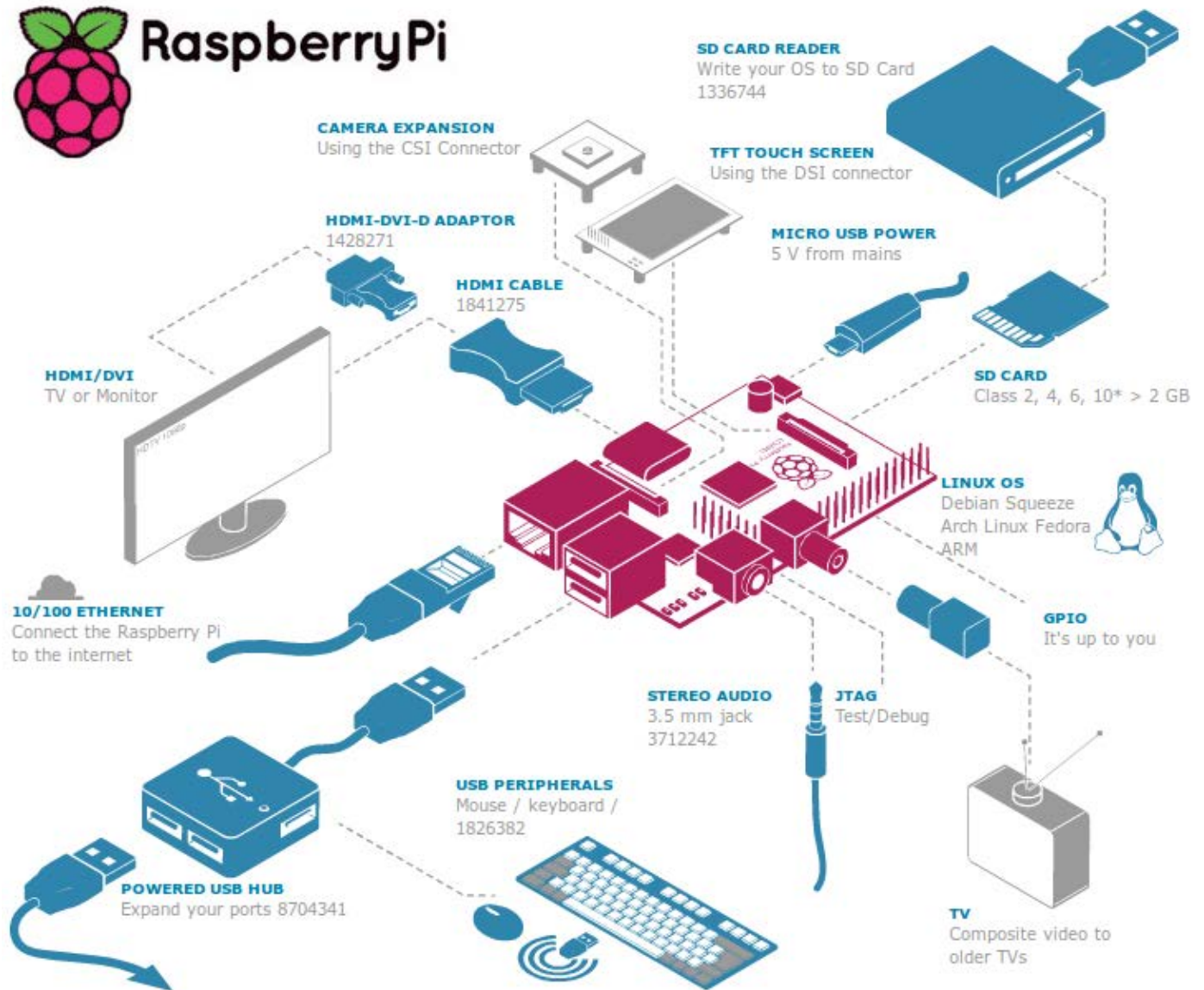


Figure 2: le Raspberry et ses périphériques

Sur la figure ci-dessus, on peut voir les principaux périphériques et leur utilisation pour le Raspberry modèle B. La différence avec le B+ et les modèles 2 et 3 dont nous avons parlé est principalement le nombre de connecteurs USB (au nombre de 2 au lieu de 4) et un connecteur GPIO de 26 broches au lieu des 40 broches pour les modèles B+, 2 et 3. Pour information le modèle B+ a été lancé en 2014.

Pour plus d'information, on pourra aller sur <http://raspbrian-france.fr/choisir-accessoires-raspberry-pi-b-plus/>

C) Premiers pas

Dans cette partie nous allons installer l'image Raspbian sur la SDCard et prendre en main la Raspberry. Le choix fait ici est d'utiliser le minimum de matériel pour le développement d'applications web et de commandes d'E/S déportées. Le Raspberry sera configuré en mode AP (Access Point).

C.1) Matériel nécessaire

Afin de travailler sur la Raspberry, vous aurez besoin :

- ✓ D'une alimentation au minimum 5V/1A mais 5/2A sera préférable (Micro USB Output Power Adapter)
- ✓ Une SDCard 2Go (ou plus)
- ✓ Une clé USB/Wifi compatible linux et capable de supporter le mode AP : **cf-wu720N** , on pourra aller sur le site http://elinux.org/RPi_USB_Wi-Fi_Adapters et choisir en prenant soin de vérifier la colonne AP=YES.
- ✓ Un câble ethernet à brancher sur la Raspberry
- ✓ Une connexion filaire à internet (routeur, box,...)



C.2) Installation sur la sdcard

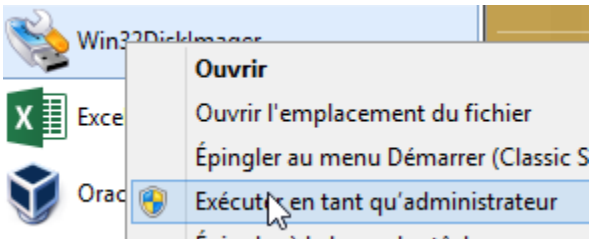
Nous l'avons dit, il n'y a pas d'OS sur la Raspberry, il faut donc télécharger l'image sur la sdcard (2G min). Nous allons utiliser une image déjà préparée (point d'accès, serveur SAMBA, GPIO activés, apache et nodejs installé) qui se trouve sur <https://github.com/jlsalvat/raspberry/tree/master>. Cette image est tirée de <https://www.raspberrypi.org/downloads/raspbian/> image jessie Lite. Pour copier l'image Raspbian, vous aurez besoin de Win32DiskImager et de lancer cette application en mode administrateur.

Télécharger sur <https://github.com/jlsalvat/raspberry/tree/master> le fichier rasp-min.zip



Dézipper le fichier rasp-min.zip et attendre que le fichier image rasp-min.img (taille 1,389Go) soit dézippé

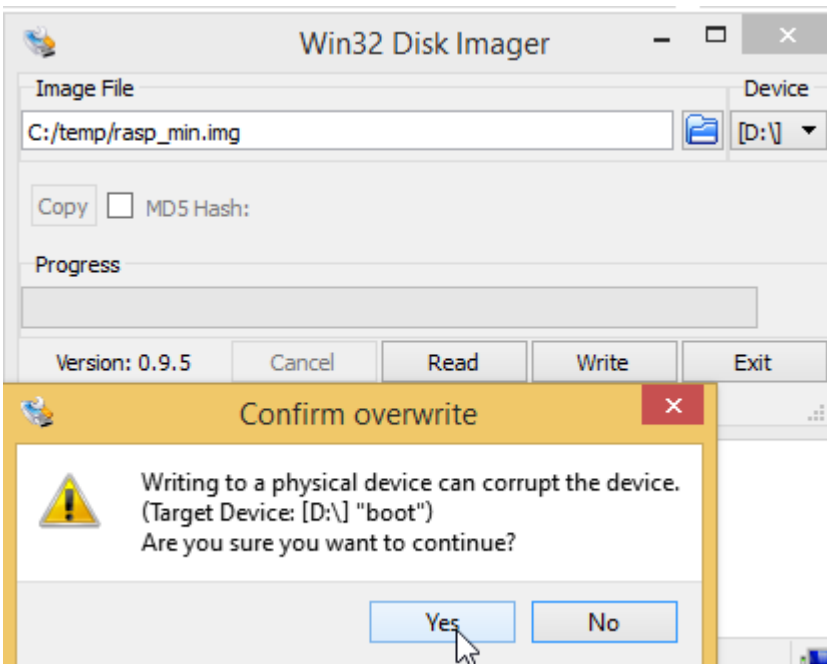
Télécharger Win32DiskImager <http://sourceforge.net/projects/win32diskimager/>

Lancer Win32DiskImager en mode administrateur et copier rasp-min.img sur la SDCard.



Une fois dézippé, l'image rasp_min.img à copier (secteur par secteur) sur la SDCard a une taille de 1,389Go. C'est cette image que l'on va installer sur la SDCard. Attention à bien écrire (write) au bon endroit (la SDCard est vue comme le disque D dans mon cas).

 rasp_min.img	14/05/2016 18:13	Fichier d'image di...	1 389 568 Ko
 rasp_min.zip	15/05/2016 00:58	Dossier compressé	401 523 Ko

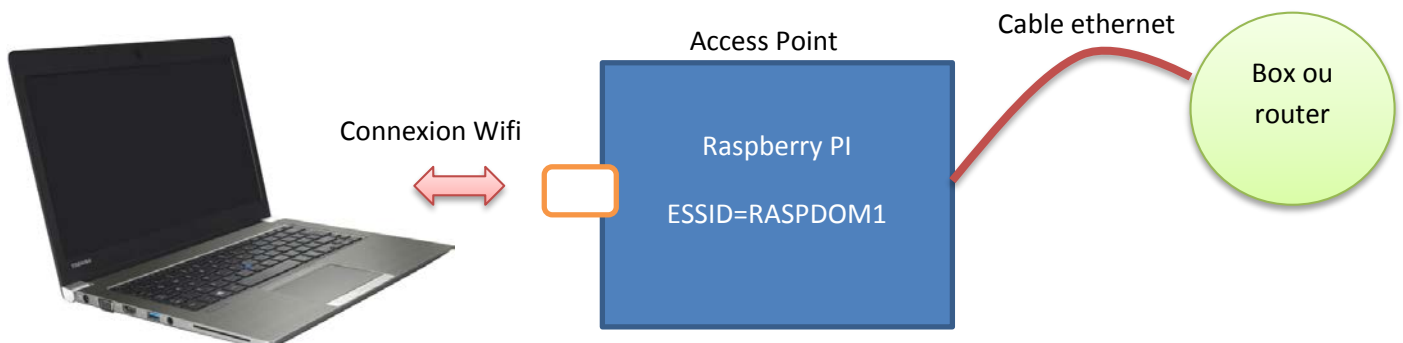


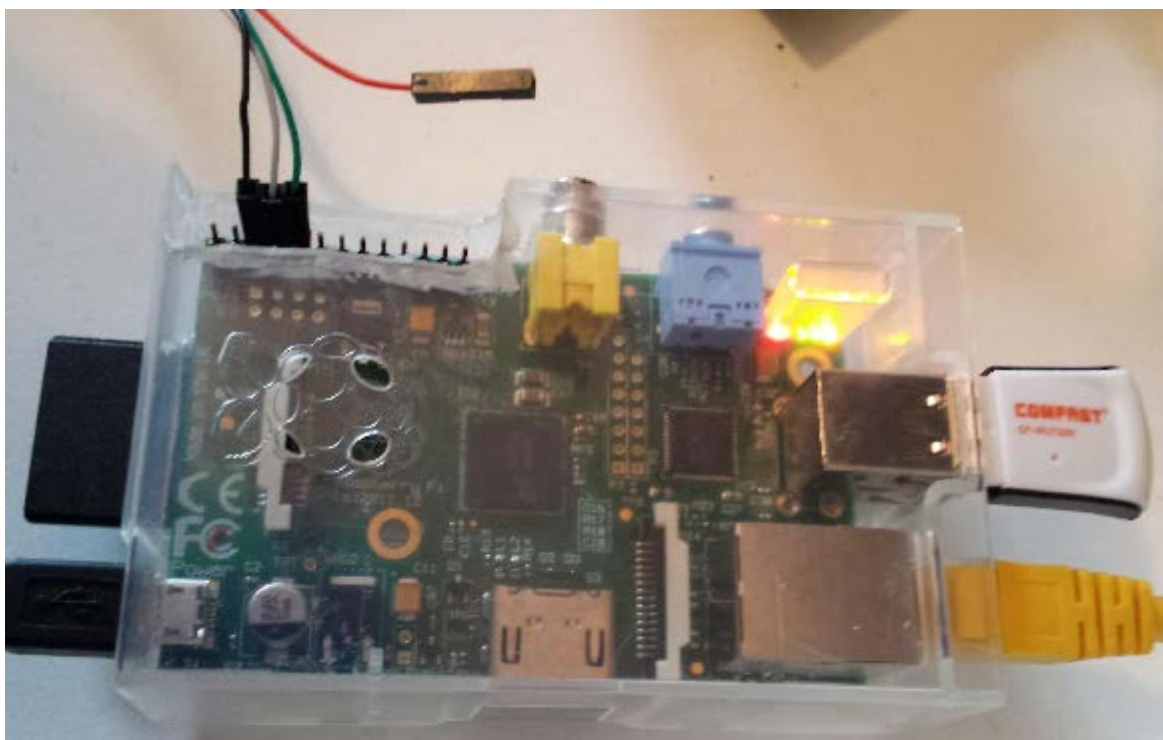
Au moment ou vous cliquez sur Write, une fenêtre apparait, c'est le moment de bien vérifier que l'on écrit au bon endroit !

Une fois l'écriture effectuée, (Write Successful) cliquer sur Exit. Votre SDCard est prête.

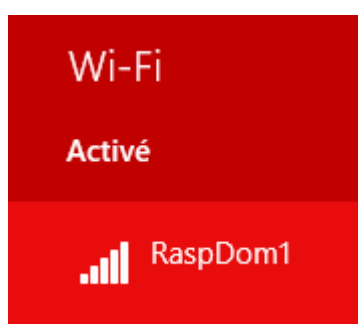
C.3) Premier test

Vous êtes prêt pour tester votre Raspberry. Afin de pouvoir faire les différents tests qui vont suivre, il est nécessaire de brancher le dongle wifi et de brancher le cable ethernet sur la raspberry. Insérez la SDCard et branchez votre alimentation

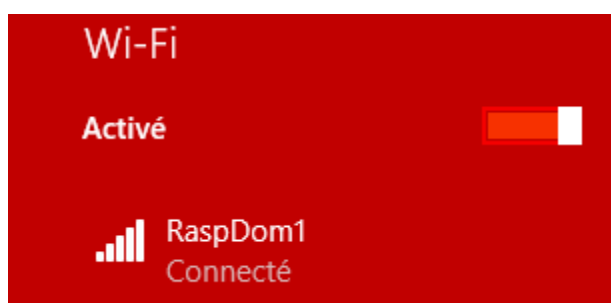




Après un certain temps, vous devriez voir apparaitre un nouveau ESSID sur votre PC.



Connectez-vous, clé wifi : **raspberry**



Vous devriez voir que vous êtes connectés au nouveau réseau.

Et si ça ne marche pas ?

Dans le cas où vous auriez un souci, il est toujours possible de connecter votre PC au routeur et de trouver l'adresse IP du Raspberry puis de se connecter en SSH (avec putty) comme nous allons le faire par la suite.

Une autre solution est de se connecter en série, au moyen d'un câble USB TTL Raspberry comme on peut le voir sur la photo ci-dessus. Dans ce cas il est possible de se connecter sur le Raspberry en téléchargeant TeraTerm et en choisissant la vitesse de 115200bps, comme sur le tutoriel adafruit



<https://learn.adafruit.com/adafruit-raspberry-pi-lesson-5-using-a-console-cable/overview>

Remarque:

Au démarrage après le temps du boot, la led D7 doit clignoter (activité réseau). La led Rouge doit être allumée (carte alimentée).

LED and Led Color	Description
D5 Green	System okay/SD card access
D6 Red	Power okay, 3.3 V
D7 Green	Full duplex; half duplex if the LED is off
D8 Green	Link activity for the LAN

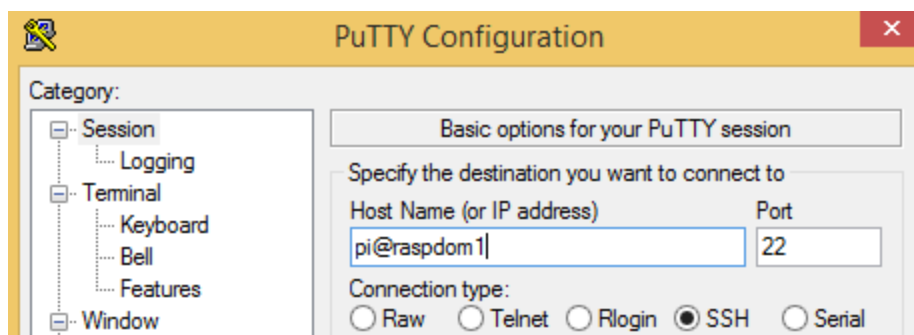
C.4) Première connexion ssh avec putty

Vous êtes connecté en wifi à votre raspberry.

Installez, si vous ne l'avez pas, putty sur votre PC.

<https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Connectez-vous à votre raspberry



Tapez le mot de passe : raspberry

```
pi@raspdom1: ~  
Using username "pi".  
pi@raspdom1's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat May 14 18:27:44 2016  
pi@raspdom1:~ $
```

Quelques explications :

Pour se connecter à votre raspberry vous auriez aussi pu trouver son adresse IP et vous connectez à son adresse IP.

Pour cela il suffit sur votre PC (en bas à droite) de cliquer sur les paramètres réseau, clic droit, Ouvrir le centre de réseau et de Partage et cliquer sur RaspDom1 (le nom de votre point d'accès).

Panneau de configuration > Réseau et Internet > Centre Réseau et partage

Page Outils ?

Afficher les informations de base de votre réseau

Afficher vos réseaux actifs

es de la	RaspDom1	Type d'accès :	Internet
es de	Réseau privé	Groupe résidentiel :	Prêt à créer
		Connexions :	Wi-Fi (RaspDom1)

RaspDom1 2
Accès Internet

14:49
16/05/2016%

État de Wi-Fi

Général

Connexion

Connectivité IPv4

Connectivité IPv6

État du média :

SSID :

Durée :

Vitesse :

Qualité du signal

Détails...

Activité

Octets :

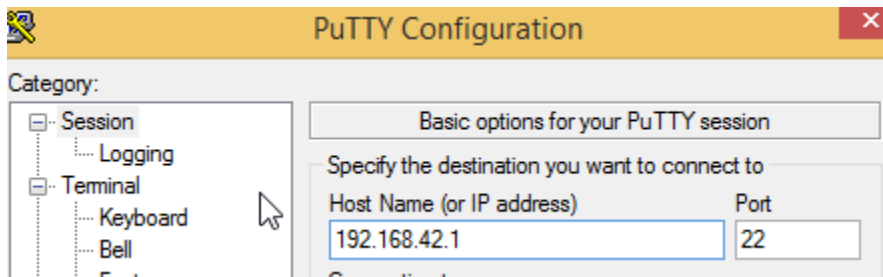
Détails de connexion réseau

Détails de connexion réseau :

Propriété	Valeur
Suffixe DNS propre à la ...	local
Description	Carte réseau Broadcom 802.11n
Adresse physique	68-94-23-4B-56-86
DHCP activé	Oui
Adresse IPv4	192.168.42.11
Masque de sous-réseau ...	255.255.255.0
Bail obtenu	lundi 16 mai 2016 14:13:56
Bail expirant	lundi 16 mai 2016 14:58:55
Passerelle par défaut IPv4	192.168.42.1
Serveur DHCP IPv4	192.168.42.1
Serveurs DNS IPv4	8.8.8.8
	8.8.4.4

Adresse de la raspberry

Vous pouvez donc vous connecter au raspberry en utilisant l'adresse 192.168.42.1.



Par défaut le login est **pi** et le mot de passe **raspberry**.

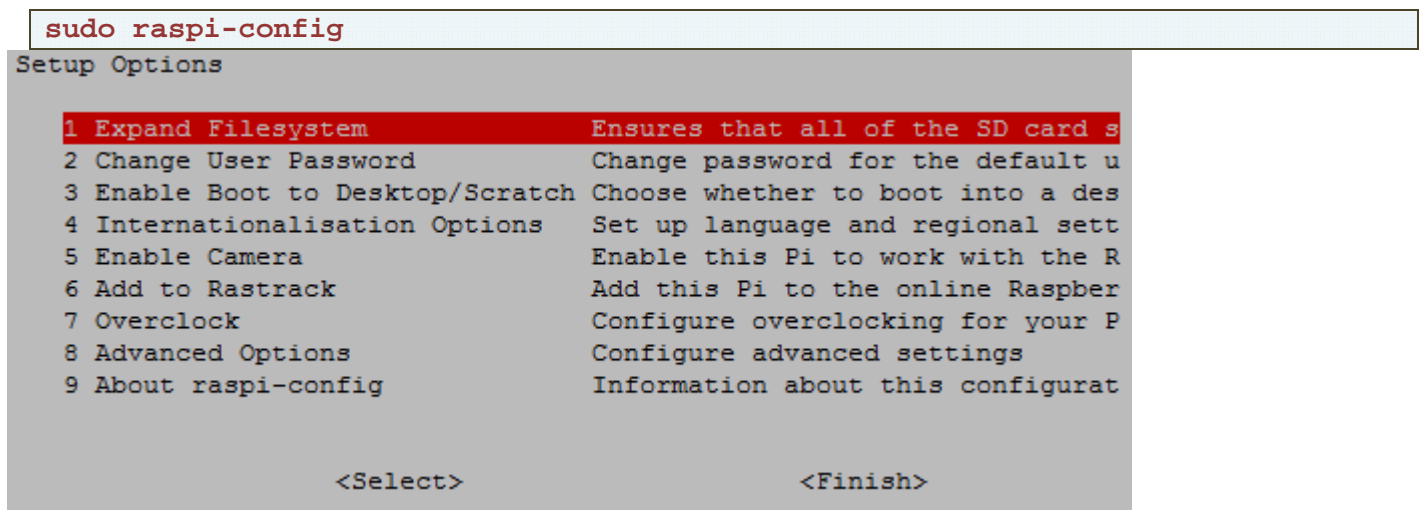
Par rapport à la configuration raspbian, l'utilisateur **root** a été ajouté. On peut donc se connecter aussi avec les privilèges du super utilisateur **root**.



Il est aussi possible de se connecter à la raspberry au travers de son nom (hostname).

```
root@raspdom1:~# more /etc/hostname
raspdom1
```

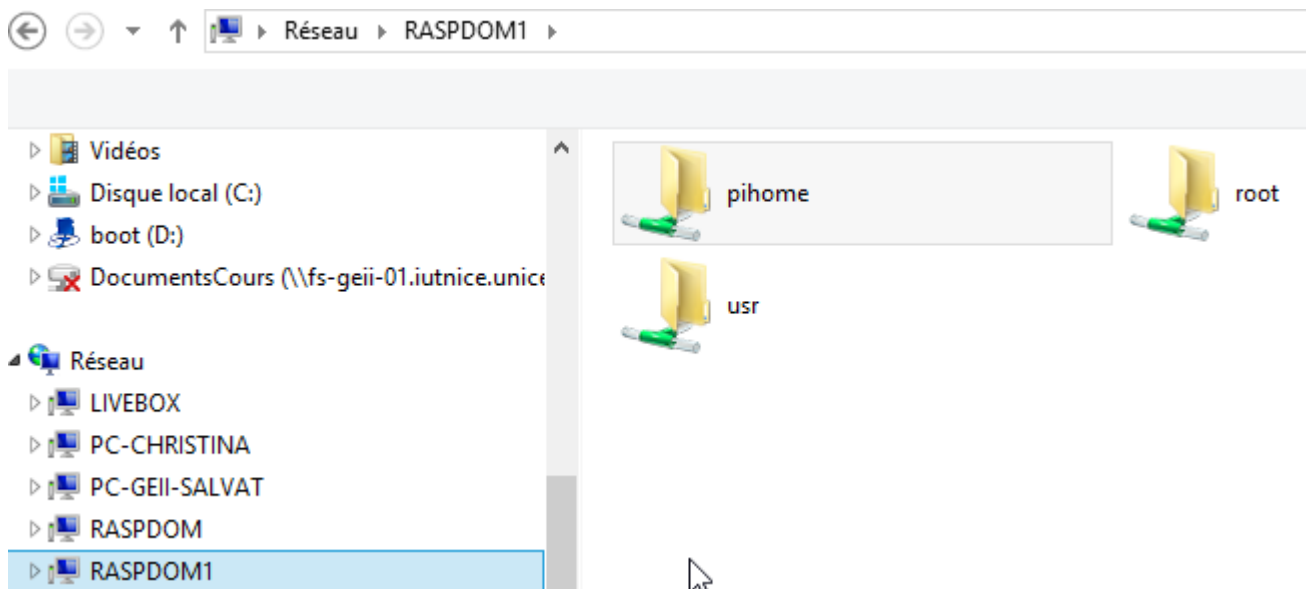
Passez en taille maximale :



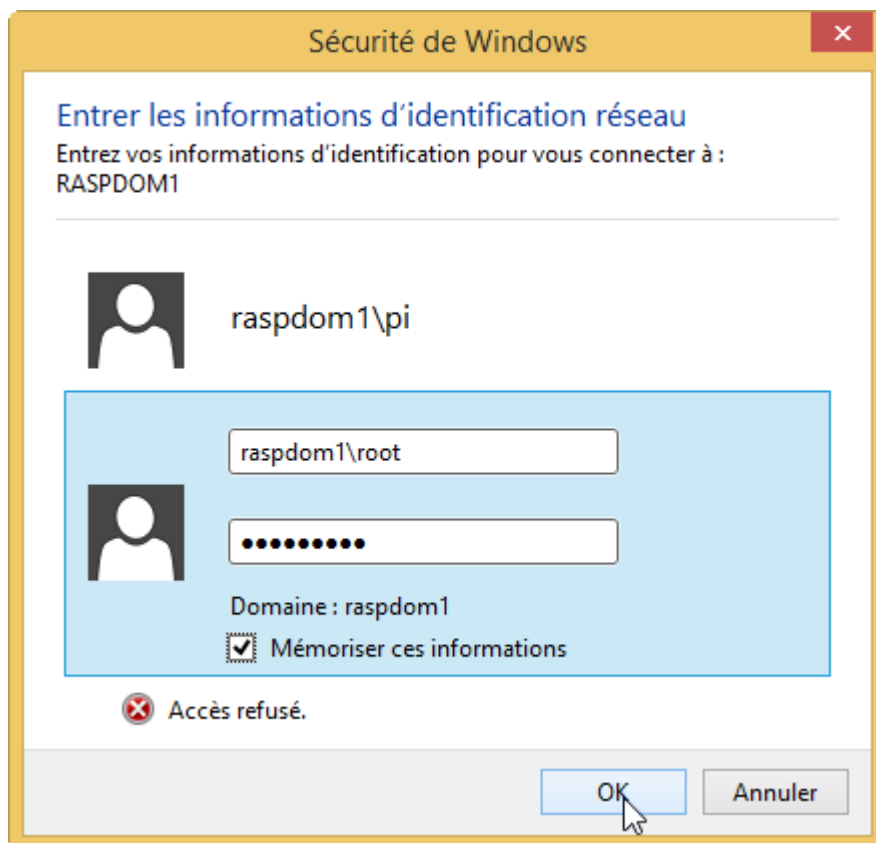
Expand filesystem, Finish, reboot pour prendre en compte les changements.

C.5) Test du partage samba

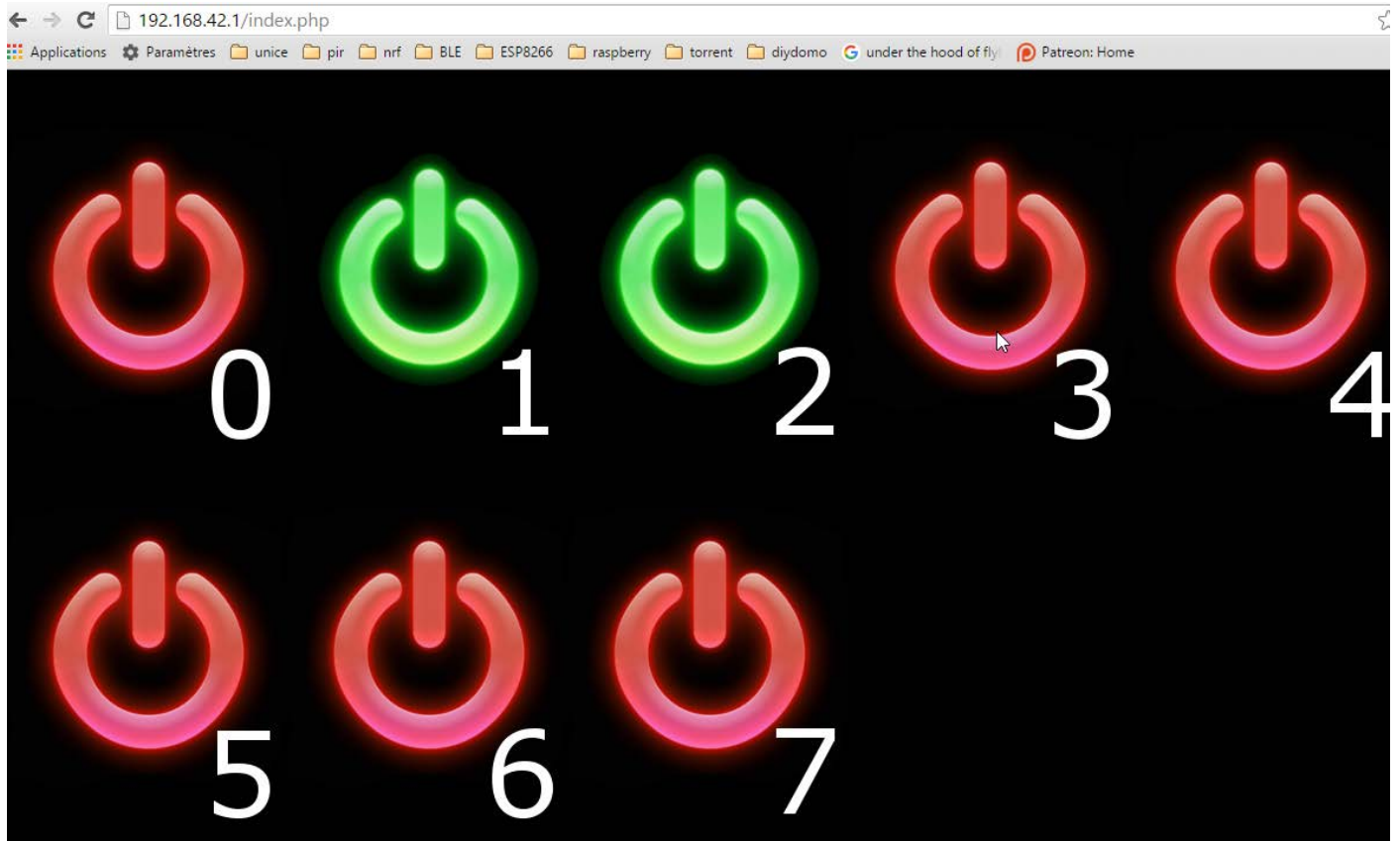
Sur votre pc, vous devriez voir la machine raspdom1 et 3 répertoires partagés. La première fois, il faudra se connecter avec toujours le même login : **pi** et le mot de passe **raspberry**. En cas de problème on pourra essayer connecter un autre compte : **raspdom1\pi** et mot de passe **raspberry**



Pour piHome on se connectera avec le login pi , pour root on se connectera avec le login root.



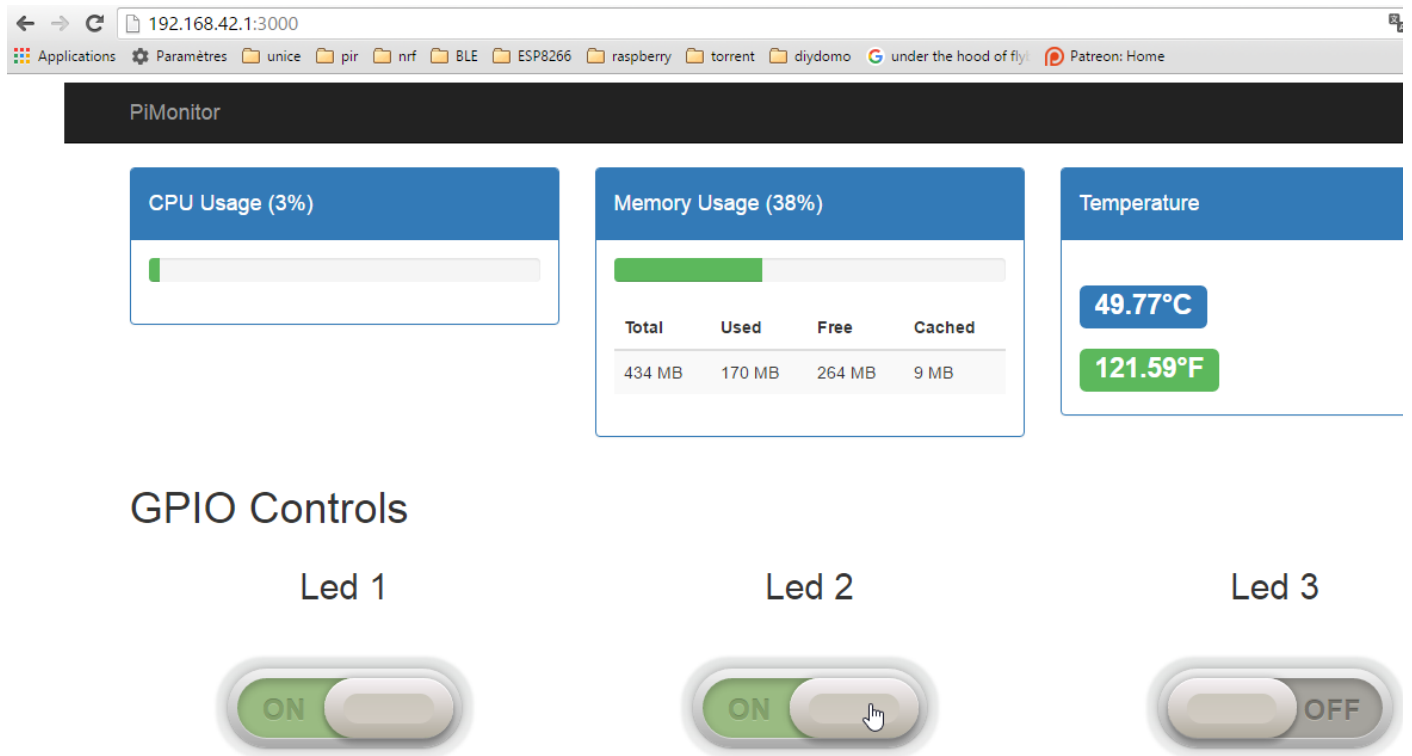
C.6) Test du serveur apache



C.7) Test nodejs

Lancer le serveur node

```
pi@raspdom1:~ $ sudo node pimonitor-master/bin/www
Client Connected
GPIOSwitch5 State Changed: true
GPIOSwitch6 State Changed: true
```



C.8) Allumer ou éteindre des LEDS à partir de ssh

En utilisant directement le driver sur la broche 23

```
pi@raspdom1:~ $ echo 23 > /sys/class/gpio/export
pi@raspdom1:~ $ out > /sys/class/gpio/gpio23/direction
-bash: out : commande introuvable
pi@raspdom1:~ $ echo out > /sys/class/gpio/gpio23/direction
pi@raspdom1:~ $ echo 1 > /sys/class/gpio/gpio23/value
pi@raspdom1:~ $ echo 0 > /sys/class/gpio/gpio23/value
```

Avec wiringpi

P1: The Main GPIO connector						
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO
		3.3v	1	2	5v	
8	Rv1:0 - Rv2:2	SDA	3	4	5v	
9	Rv1:1 - Rv2:3	SCL	5	6	0v	
7	4	GPIO7	7	8	TxD	14
		0v	9	10	RxD	15
0	17	GPIO0	11	12	GPIO1	18
2	Rv1:21 - Rv2:27	GPIO2	13	14	0v	
3	22	GPIO3	15	16	GPIO4	23
		3.3v	17	18	GPIO5	24
12	10	MOSI	19	20	0v	
13	9	MISO	21	22	GPIO6	25
14	11	SCLK	23	24	CE0	8
		0v	25	26	CE1	7
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO

La led 4 s'allume...

```
pi@raspdom1:~ $ gpio mode 4 out
pi@raspdom1:~ $ gpio write 4 1
pi@raspdom1:~ $ gpio write 4 0
```

La pwm

```
pi@raspberrypi /sys/class/gpio $ gpio mode 1 pwm
pi@raspberrypi /sys/class/gpio $ gpio pwm 1 500
```

C.9) Premier programme sur raspberry

Ecrire le programme test.c. Utiliser nano (CTR+X o pour sauver le fichier)

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>

//wiringPI v2
//http://wiringpi.com/download-and-install/
// how to use all the things in the salvat board

void setup(){
//initialisation : Cf http://wiringpi.com/reference/setup/
// This fucntion must be called first to use of wiring Lib
    if(wiringPiSetup()==-1)
        exit(1);
// pin configuration
    pinMode(4,OUTPUT); // in ver2 pin4 = GPIO23
    pinMode(5,OUTPUT); // in ver2 pin5 = GPIO24
    pinMode(6,OUTPUT); // in ver2 pin6 = GPIO25
}

int main()
{
    int i ;
    printf("Hello world!\n");
    setup();
//loop
    while(1)
    {
        for (i=4;i<=6;i++){
            digitalWrite(i,HIGH);
```



```

        delay(50);
        digitalWrite(i,LOW);
        delay(50);
    }

    return 0;
}

```

```

pi@raspdom1: ~
GNU nano 2.2.6      Fichier : test.c
{
    int i ;
    printf("Hello world!\n");
    setup();
//loop
    while(1)
    {
        for (i=4;i<=6;i++){
            digitalWrite(i,HIGH);
            delay(50);
            digitalWrite(i,LOW);
            delay(50);
        }

        return 0;
    }
}

```

Compiler le programme et le lancer en mode root. Regardez le chenillard sur les leds 4 5 6

```

pi@raspdom1:~ $ nano test.c
pi@raspdom1:~ $ gcc test.c -o test -lwiringPi
pi@raspdom1:~ $ ./test
Hello world!
wiringPiSetup: Must be root. (Did you forget sudo?)
pi@raspdom1:~ $ sudo ./test
Hello world!

```

D) Programmation sur le PC

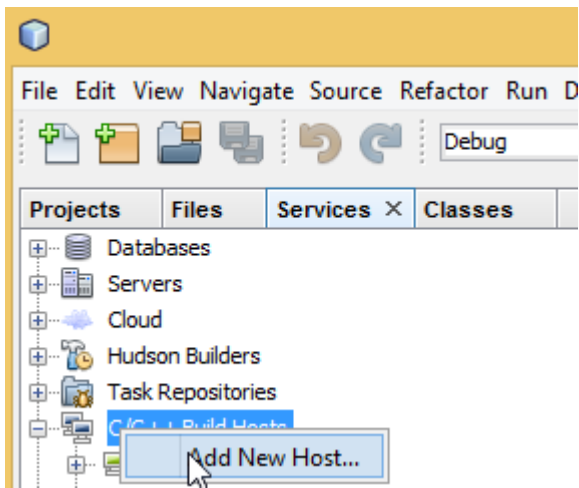
Dans ce chapitre nous allons travailler sur le PC, utiliser netbean comme plateforme de développement et utiliser le compilateur de la raspberry pour la compilation du programme. Netbean intègre les outils permettant une connexion à la cible, de télécharger le programme, de le lancer mais aussi de déboguer le programme sur la cible (gdb + gdbserver sur la cible).

installer jdk et netbean sur le pc

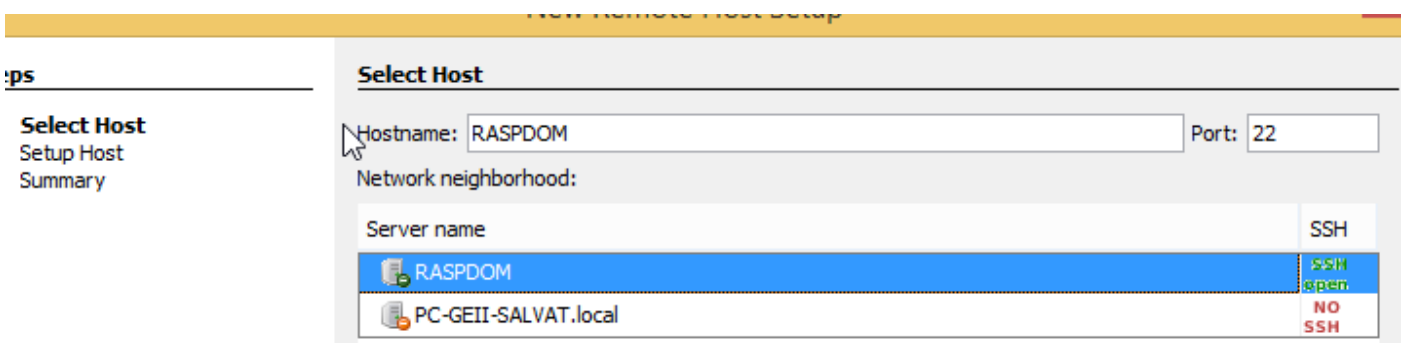
Lancer netbean

La première chose à faire est d'ajouter la cible sur laquelle nous allons travailler.

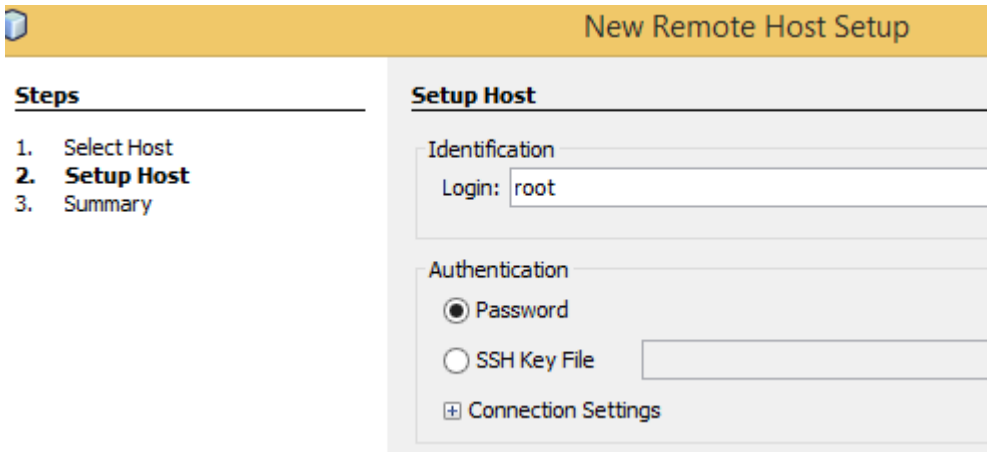
Dans l'onglet service, C/C++ Build Hosts (Add New Host)



Ajouter le nom ou l'adresse de la cible (raspdom1 ou 192.168.42.1)

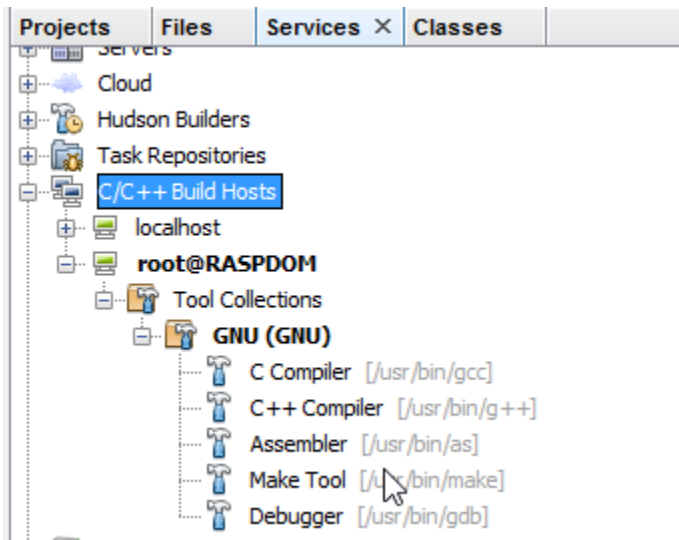


Ensuite ajouter le login root et le mot de passe raspberry pour la connexion à la cible

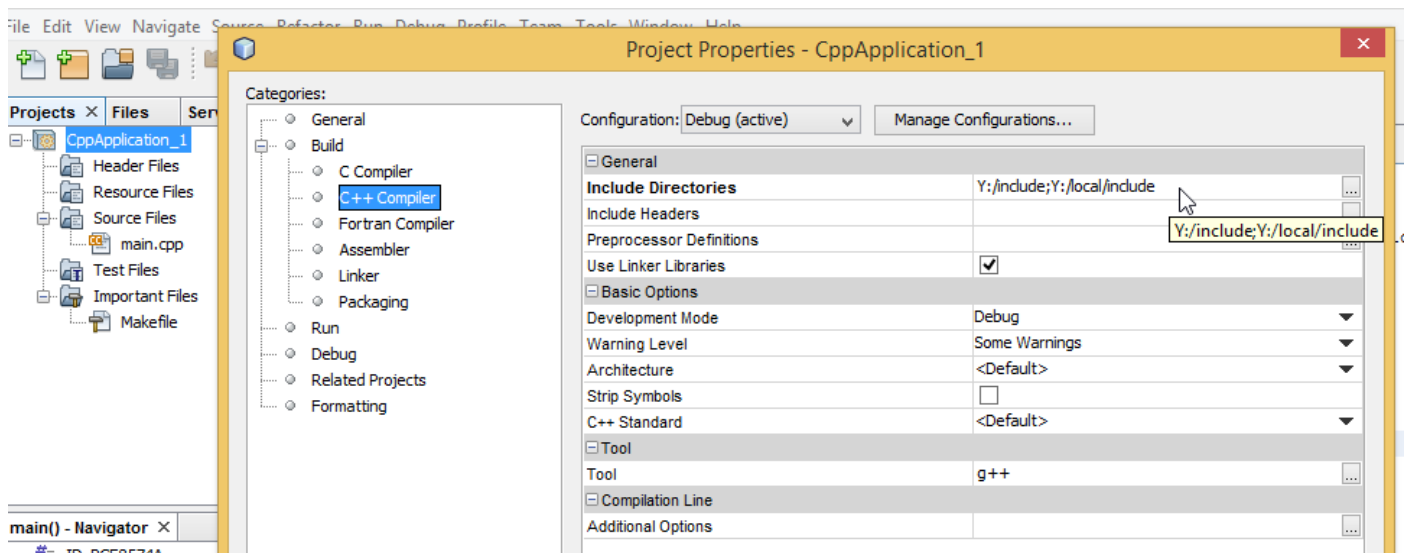


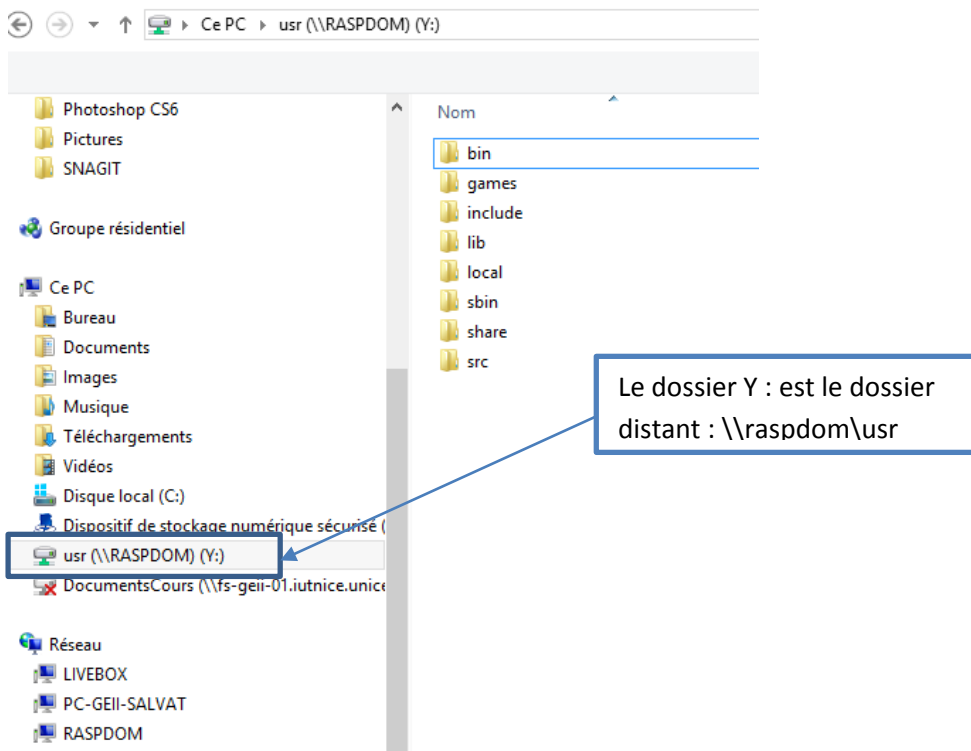
Vous avez terminé la configuration de Netbean. Netbean devrait vous afficher les compilateurs et debugger qu'il a trouvé sur la cible.

Dans nos copies d'écran, la cible s'appelle raspdom

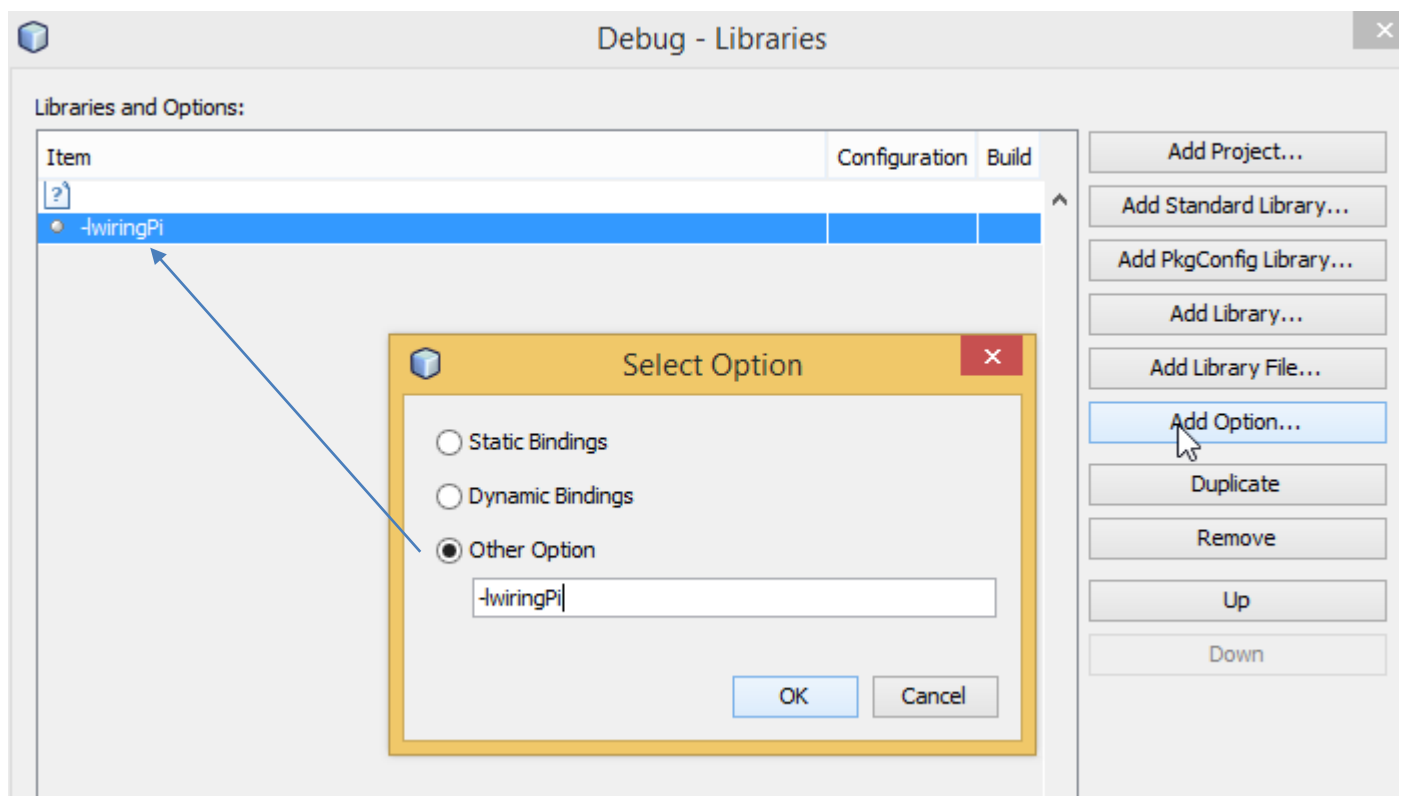


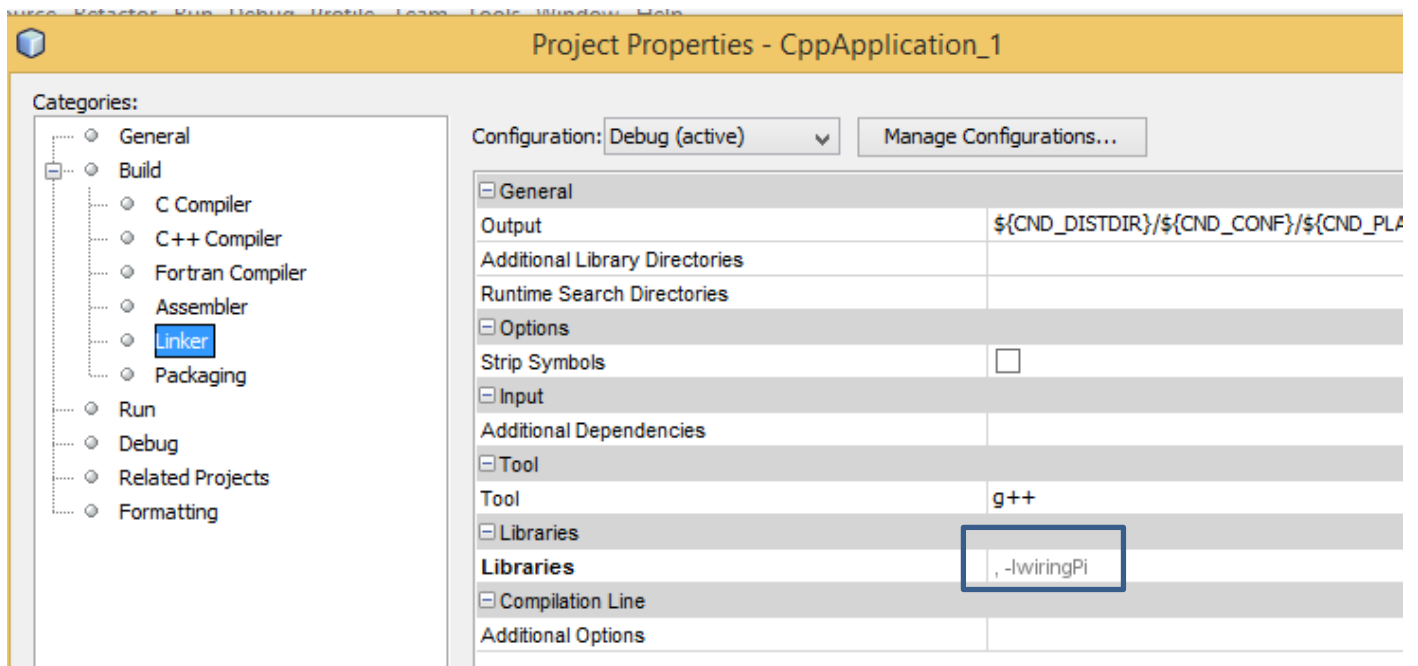
Afin d'avoir l'aide syntaxique il est nécessaire que netbean aient accès aux fichiers de définition sur la cible. Ces fichiers se trouvent dans raspdom1://usr/include et raspdom1://usr/local/include. Ces fichiers sont visibles sur le pc grace au serveur samba sur la cible comme nous l'avons vu précédemment. Vous pouvez dans l'explorateur, ajouter un lecteur réseau directement sur raspdom1://usr et le connecter sur le lecteur réseau Y:/ comme cela a été fait ci-dessous.



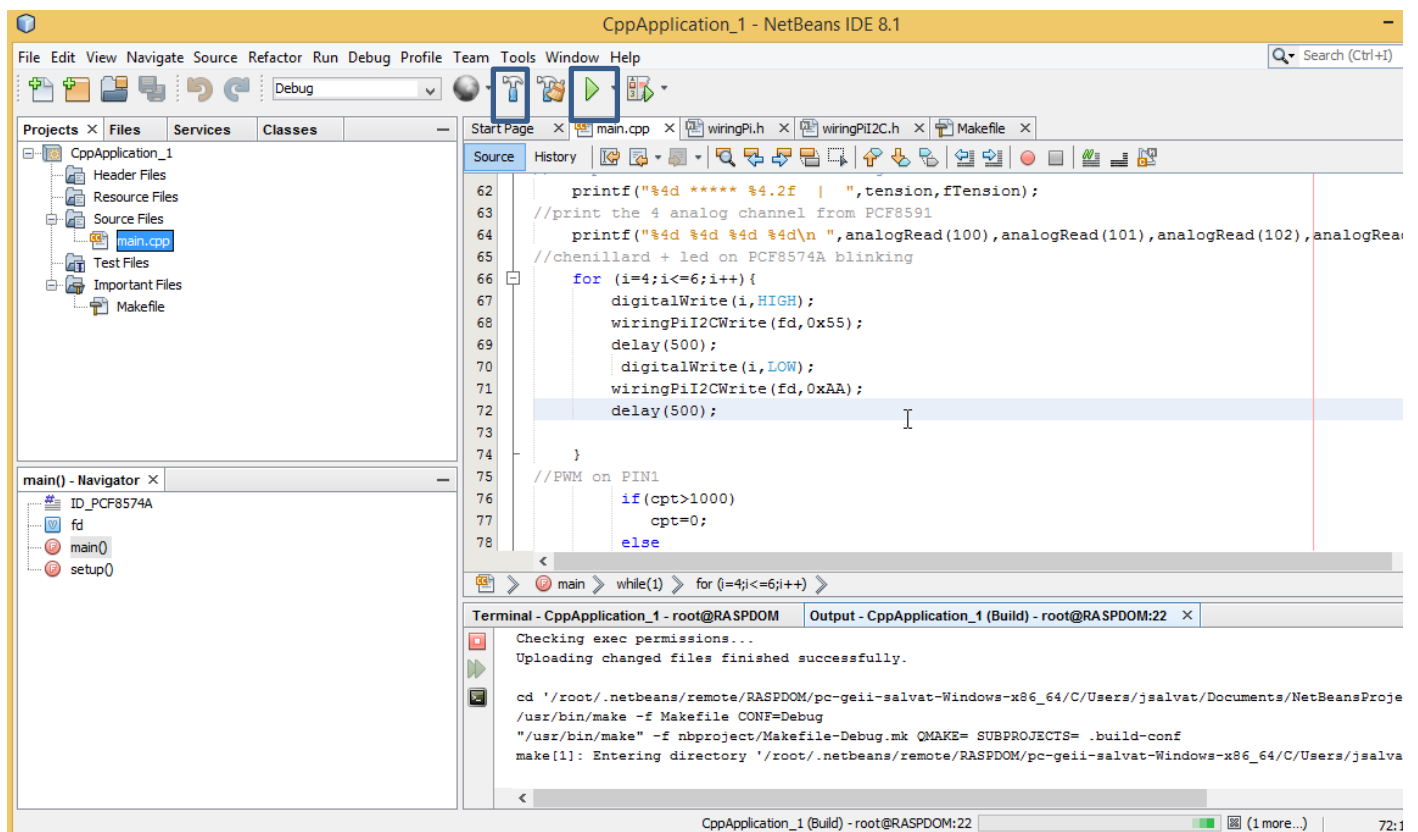


Et pour finir dans linker, on ajoute la compilation avec la bibliothèque wiringPi, comme nous l'avons fait lors de la compilation sur la cible.



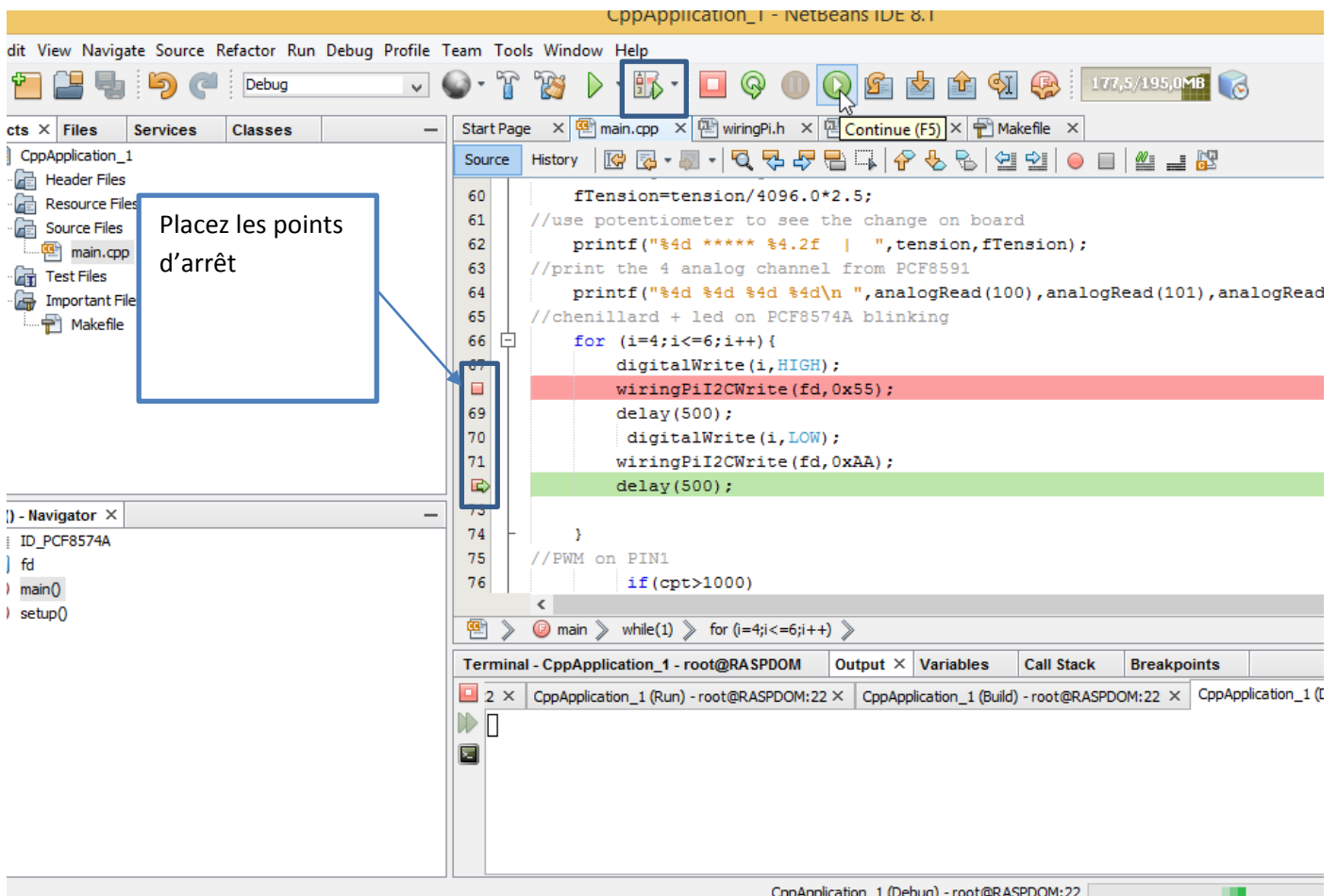


Pour finir créer le fichier main.cpp et mettez le programme que l'on avait tester sur la cible (test.c).



Compiler le programme (le marteau) et lancer le programme sur la cible (flèche verte)

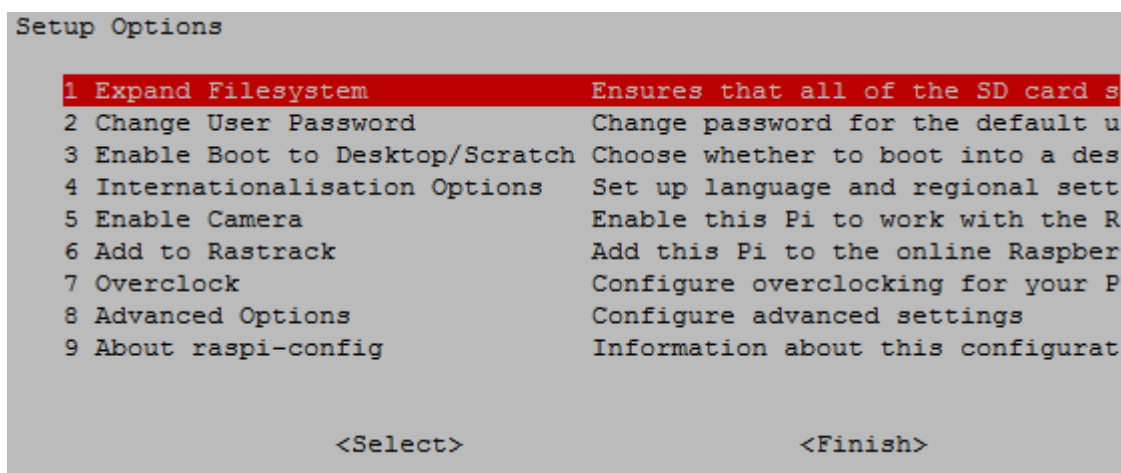
Il est aussi possible de déboguer le programme sur la cible...



E) Astuces diverses

E.1) La commande raspi-config

Les outils fournis par raspi-config sont divers :



On peut choisir un clavier arzetty (fr) un codage en UTF8 et langue en fr_FR.UTF8 pour le codage des caractères dans internationalisation option

Une autre solution est de le faire à la main avec la commande de gestion des paquets

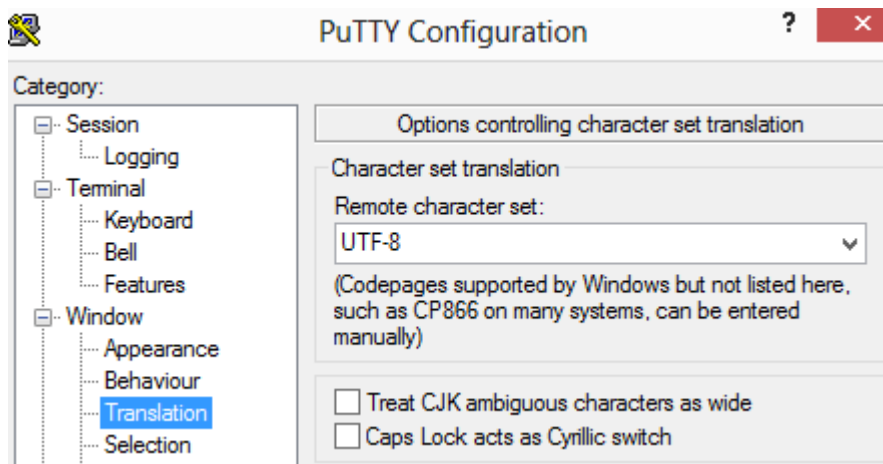
```
sudo dpkg-reconfigure locales
```

Pour la partie clavier azerty ou qwerty

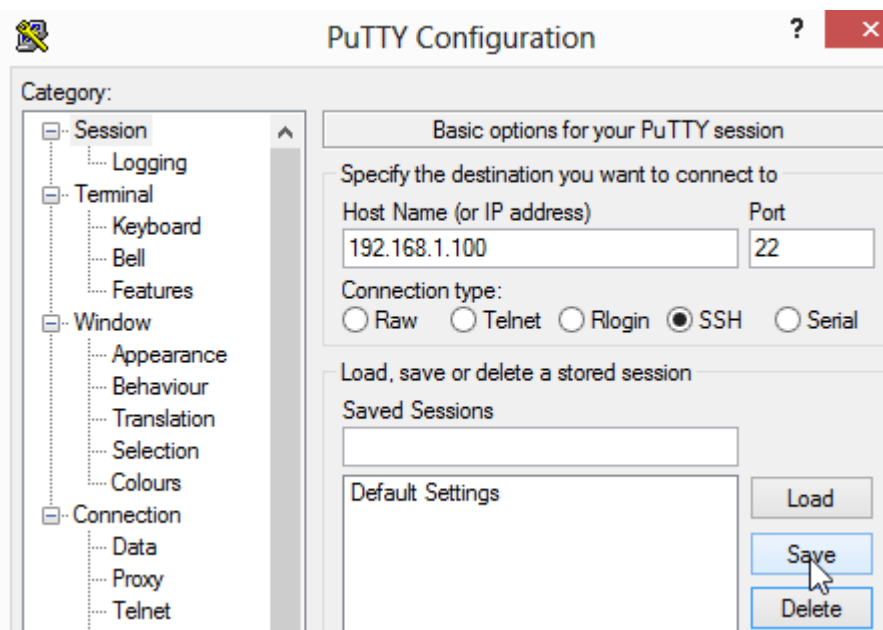
```
sudo dpkg-reconfigure keyboard-configuration
```

E.2) Internationalisation putty et sauvegarde des choix

Lors du lancement de la console pour avoir les bons caractères affichés (on est en UTF8 coté cible).



Puis on va dans Session



Et on sauve pour les prochaines fois...

A chaque fois qu'on ouvrira putty, ce sera avec ces paramètres (Default Settings).

E.3) Création de partition et montage de systèmes de fichier

Il est possible avec la commande fdisk de créer des partitions en plus des 2 partitions déjà existantes. Par exemple, on peut créer une partition accueillant un système de fichier FAT qui sera visible sur un PC lorsque l'on insérera la SDCard sur le PC. Evidemment un montage Samba est plus simple que la manipulation de la Sdcard, comme nous l'avons vu.

Prenons ici un exemple de raspberry avec différentes partitions : En utilisant df -h on peut voir qu'ici il y a une partition de 2,7G en linux et une de 56Mo montée sur boot (en vfat). D'autres partitions temporaires sont montées en mémoire ram et donc n'existent pas sur la Sdcard. La commande fdisk montrent les partitions sur la SDCard. Nous

en avons ici plus de 2. La SDCard est vu sous linux en /dev/mmcblk0 et les différentes partitions se nomment /dev/mmcblk0p1, /dev/mmcblk0p2

```
pi@raspberrypi ~ $ df -h
Sys. fich.      Taille Util. Dispo Uti% MontÃ© sur
rootfs          2,7G  2,0G  594M   78% /
/dev/root        2,7G  2,0G  594M   78% /
devtmpfs         212M    0  212M    0% /dev
tmpfs            44M   240K   44M    1% /run
tmpfs            5,0M    0   5,0M    0% /run/lock
tmpfs            88M    0   88M    0% /run/shm
/dev/mmcblk0p1   56M   19M   38M   33% /boot
pi@raspberrypi ~ $ sudo fdisk -l

Disk /dev/mmcblk0: 7822 MB, 7822376960 bytes
4 heads, 16 sectors/track, 238720 cylinders, total 15278080 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: 0x0002c262

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1            8192        122879         57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2       122880       5785599       2831360    83   Linux
/dev/mmcblk0p3         2048           8190         3071+    83   Linux
/dev/mmcblk0p4       5785600       15278079       4746240    5   Extended
/dev/mmcblk0p5       5785616       7738751         976568    c   W95 FAT32 (LBA)
/dev/mmcblk0p6       7738768       15278079       3769656    83   Linux
```

Le montage des systèmes de fichiers se trouvant dans les partitions non montées se fait dans /etc/fstab

Si je veux monter les 2 partitions, il suffit d'ajouter dans /etc/fstab le montage, de créer les répertoires de montage et de faire le montage

```
pi@raspberrypi ~ $ more /etc/fstab
proc            /proc          proc           defaults      0          0
/dev/mmcblk0p1  /boot          vfat           defaults      0          2
/dev/mmcblk0p2  /              ext4           defaults,noatime 0          1
/dev/mmcblk0p5  /media/vfat     vfat           defaults      0          2
/dev/mmcblk0p6  /media/data     ext4           defaults      0          2

# a swapfile is not a swap partition, so no using swapon|off from here on, use
dphys-swapfile swap[on|off] for that
pi@raspberrypi ~ $ cd /media/
pi@raspberrypi /media $ ls
pi@raspberrypi /media $ sudo mount -a
mount: mount point /media/vfat does not exist
mount: mount point /media/data does not exist
pi@raspberrypi /media $ sudo mkdir /media/vfat
pi@raspberrypi /media $ sudo mkdir /media/data
pi@raspberrypi /media $ sudo mount -a
pi@raspberrypi /media $ df -h
Sys. fich.      Taille Util. Dispo Uti% MontÃ© sur
rootfs          2,7G  2,0G  594M   78% /
/dev/root        2,7G  2,0G  594M   78% /
devtmpfs         212M    0  212M    0% /dev
tmpfs            44M   240K   44M    1% /run
tmpfs            5,0M    0   5,0M    0% /run/lock
tmpfs            88M    0   88M    0% /run/shm
/dev/mmcblk0p1   56M   19M   38M   33% /boot
/dev/mmcblk0p5   952M   4,0K  952M    1% /media/vfat
/dev/mmcblk0p6   3,6G   72M   3,3G    3% /media/data
```

Autre solution :La création de nouvelles partitions :

```
sudo cfdisk /dev/mmcblk0
```

```
cfdisk (util-linux 2.20.1)

Disk Drive: /dev/mmcblk0
Size: 7822376960 bytes, 7822 MB
Heads: 4 Sectors per Track: 16 Cylinders: 238720
```

Name	Flags	Part Type	FS Type	[Label]	Size (MB)

			Unusable		1,05
mmcblk0p3		Primary	Linux		3,15*
			Unusable		0,01*
mmcblk0p1		Primary	vfat	[boot]	58,73
mmcblk0p2		Primary	ext4		2899,32
mmcblk0p5		Logical	vfat		1000,02
mmcblk0p6		Logical	ext4		3860,14

```
[ Help ] [ Print ] [ Quit ] [ Units ] [ Write ]
```

En utilisant ps-ef, on peut voir que ssh tourne et donc on peut se connecter en utilisant putty à la cible

Quelques commandes utiles

```
more /proc/cpuinfo
more /proc/partitions
free -m
df -h
du -hs
file /bin/bash
grep find /etc/init.d/* recherche find dans tous les fichiers de init.d
wish ssh
ifconfig
sudo ifup eth0
sudo ifdown eth0
sudo reboot
sudo halt
sudo mkfs.ext4 /dev/mmcblk0 (pour formater une flash).
Sudo apt-get install dosfstools pour récup mkfs.vfat
```

```
pi@raspberrypi /media $ sudo apt-cache search vfat
dosfstools - utilities for making and checking MS-DOS FAT filesystems
dosfstools-dbg - utilities for making and checking MS-DOS FAT filesystems
)
fuse-posixovl - FUSE file system that provides POSIX functionality
udisks - storage media interface
pi@raspberrypi /media $ sudo apt-get install dosfstools
```

Où se trouvent les paramètres réseaux ?

```
more /etc/network/interfaces
```

Les fichiers importants : etc/resolv.conf (DNS), etc/hosts et etc/hostname(nom machine)

```
wget http://www.icrobotics.co.uk/wiki/images/c/c3/Pifm.tar.gz
find /home/pi -name gemgem.py
lsusb
```

E.4) Jouer avec les GPIO en bash

By default the GPIO support that is built into the kernel will not export any GPIO pins to the operating system. After all there is no way for Linux to know what you have connected to each pin. So you are going to need to export your GPIO pin before you can work on it. This is simple: first, you need the number of the GPIO pin (not the physical pin number but the logical GPIO number); in your case this will be GPIO-04. In [Figure 5-9](#) you can see two special writable files called `export` and `unexport`. You're going to need to use the `export` device. The syntax for this special device is just to echo the pin number into it. You will need to run this command to export GPIO-04:

```
# echo 4 > /sys/class/gpio/export
```

Now take a look at `/sys/class/gpio` again. You should now see a directory called `gpio4`. This directory is how you will control the GPIO pin. Take a look at [Figure 5-10](#) and you can see the contents of my `GPIO-04` directory.

You can now work on doing something with your GPIO pin. This directory contains the real business end of the GPIO subsystem. I want to give you a quick overview of each of the main special files in this directory and what value types they may accept. [Table 5-1](#) contains the descriptions of some of the most useful special files.

Table 5-1. A listing of the most useful special files

Name	Accepted Value	Description/Use
Direction	In (Input), out (Output)	Sets the GPIO pin as input or output
Value	0 (low), 1 (high)	Sets the initial pin state high or low and reads the current state
Edge	none, rising, falling	Sets the signal edge type
Active_low	0 (false), 1 (true)	Indicates whether the pin is active when driven low

Given that you need to sense the GPIO pin's state, you're going to need to set it up as an input. From [Table 5-1](#) you can easily see that you need to use the `direction` special file. To set the direction, run this command:

```
# echo in > /sys/class/gpio/gpio4/direction
```

Once you have done that, you can read the state of the GPIO pin. Once again by looking at [Table 5-1](#) you can see that you may want to take a look at the `value` special file. You can do this if you run the following command:

```
# cat /sys/class/gpio/gpio4/value
```

Script pour faire clignoter les leds...

```
#!/bin/bash
# Description : turn on / turn off every second GPIO 23, 24 , 25

# root ?
function isRoot {
if [[ $EUID -ne 0 ]]; then
    echo "you must be root -> use sudo test_leds.sh " 1>&2
    exit 1
fi
}
```



```

# Init
function init_GPIO {
    for i in 23 24 25
    do
        echo out > /sys/class/gpio/gpio$i/direction
    done
}

# if gpio doesn't exist, create
function create_GPIO {
    for i in 23 24 25
    do
        if [ -e /sys/class/gpio/gpio$i ]; then
            echo $i > /sys/class/gpio/export
        fi
    done
}

# Turn on
function on {
    for i in 23 24 25
    do
        /bin/echo 1 > /sys/class/gpio/gpio$i/value
    done
}

# Turn off
function off {
    for i in 23 24 25
    do
        /bin/echo 0 > /sys/class/gpio/gpio$i/value
    done
}

# le script
isRoot
create_GPIO
init_GPIO
while :
do
on
sleep 1
off
sleep 1
done

```

Pour utiliser l'i2C, il faut que le driver soit lancé.

C'est le cas sur l'image raspberry, le driver I2C a été activé au travers de raspi-config dans paramètres avancés

```

modprobe i2c-dev
modprobe -l

```

The i2c Ports on the raspberry pi are not enabled by default. Follow these steps to enable the i2c port. These instructions are for the official Raspbian distro.

First you will need to edit the config file that disables the i2c port by default. This setting is stored in `/etc/modprobe.d/raspi-blacklist.conf`. we will use nano to edit this but you could use any editor you are comfortable with.

```
$sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Once this file is open find this line `blacklist i2c-bcm2708` and comment it out by adding a `#` to the front of it.

```
#blacklist i2c-bcm2708
```

You can now save this file with `Ctrl+O` and pressing enter and exit nano with `Ctrl+X`. Once you are back at the command prompt reboot your raspberry pi.

```
$sudo reboot
```

Once the reboot has finished you need to make the ports active by running this command:

```
$sudo modprobe i2c-dev
```

Now you should be able to see the i2c ports listed in your `/dev/` directory

```
$ls /dev/i2c*
```

You should see `/dev/i2c-0` and `/dev/i2c-1` listed. To be able to use these ports you need to give yourself permission to access them.

```
$sudo chmod o+rw /dev/i2c*
```

Now your i2c ports will be ready for you to use. You will have to run the `modprobe` and `chmod` commands every time you login to activate the ports and apply the permissions. If you wish to automate this process so the commands are run every time you log in then you can edit `/etc/rc.local` and add the commands in.

Mais aussi :

- ✓ *Edit the file `/etc/modules` using the command `sudo nano /etc/modules` and add the following lines to the end of it.*

```
i2c-bcm2708  
i2c-dev
```

```

pi@raspberrypi /etc/modprobe.d $ sudo modprobe i2c_bcm2708
pi@raspberrypi /etc/modprobe.d $ sudo modprobe i2c-dev
pi@raspberrypi /etc/modprobe.d $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  38  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  4c  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --

```

Autre solution , utiliser le prog gpio...

gpio load i2c

gpio i2cdetect

Pour que ces modules soient lancés dès le boot

Il faut mettre dans /etc/modules

I2C-dev

Et enlever le blacklist dans modprobe.d

EN bash

PCF8574A

```

# le script
isRoot
gpio load i2c
while :
do
i2cset -y 1 0x38 0xAA
sleep 1
i2cset -y 1 0x38 0x55
sleep 1
done

```

PCF8591

```

# le script
isRoot
gpio load i2c
echo "chanel1  chanel2  chanel3 chanel4"

while :
do
  for channel in 0 1 2 3
  do
    # selection channel
    i2cset -y 1 0x4c $channel
    #reading first value (nothing)
    i2cget -y 1 0x4c >NUL
    #reading second value (ok) and print value on one line
    echo -n " $(i2cget -y 1 0x4c)"
  done
done

```

```
done
echo
sleep 1

done
```

SPI

Pour le spi il faut aussi insérer le driver spidev et spi_bcm2708 ou tout mettre dans le fichier

E.5) wiringPi

Récup de wiringPi : tout est expliqué sur <http://wiringpi.com/download-and-install/>

On suppose que la bibliothèque de dev i2C a d'abord été installée donc :

```
Sudo apt-get install libi2c-dev
```

Et sinon on peut vérifier que c'est déjà installé

```
pi@raspberrypi /media $ aptitude search libi2C
i  libi2c-dev - userspace I2C programming library development (i) veut
```

dire installé

On peut donc construire wiringPi

Une fois que c'est installé

<https://projects.drogon.net/raspberry-pi/gpio-examples/tux-crossing/gpio-examples-1-a-single-led/>

vérifier que les drivers sont bien installés (gpio i2c spi)

```
pi@raspberrypi /sys/class/gpio $ sudo lsmod
Module              Size  Used by
fuse                 69438  3
i2c_dev              5594  0
snd_bcm2835          16432  0
snd_pcm              77728  1 snd_bcm2835
snd_seq              53482  0
snd_timer            20110  2 snd_pcm,snd_seq
snd_seq_device        6462  1 snd_seq
snd                  58744  5 snd_bcm2835,snd_timer,snd_pcm,snd_seq,s
vice
snd_page_alloc        5169  1 snd_pcm
spidev                5248  0
leds_gpio            2243  0
led_class            3570  1 leds_gpio
spi_bcm2708           4841  0
i2c_bcm2708           3947  0
pi@raspberrypi /sys/class/gpio $ gpio mode 4 out
pi@raspberrypi /sys/class/gpio $ gpio write 4 1
```

La led 4 s'allume...

```
pi@raspdom1:~ $ gpio mode 4 out
pi@raspdom1:~ $ gpio write 4 1
pi@raspdom1:~ $ gpio write 4 0
```

La pwm

```
pi@raspberrypi /sys/class/gpio $ gpio mode 1 pwm
pi@raspberrypi /sys/class/gpio $ gpio pwm 1 500
```

On peut aussi tester l'I2C :

I2cdetect -y 1

```
pi@raspberrypi /sys/class/gpio $ i2cset -y 1 0x38 0xAA
```

Les premiers progs en C

E.1) Utilisation de vnc et codeblock sur raspberry

Dans ce cas il est nécessaire d'ajouter l'interface graphique et codeblock sur la cible

```
apt-get install codeblock
```

Vous allez installer l'interface graphique et un grand nombre de paquet.

Puis télécharger VNC Viewer sur le PC

Download VNC® Viewer

[All downloads](#) » [VNC Viewer](#) » [Windows](#) » VNC View

Your download will begin shortly for:



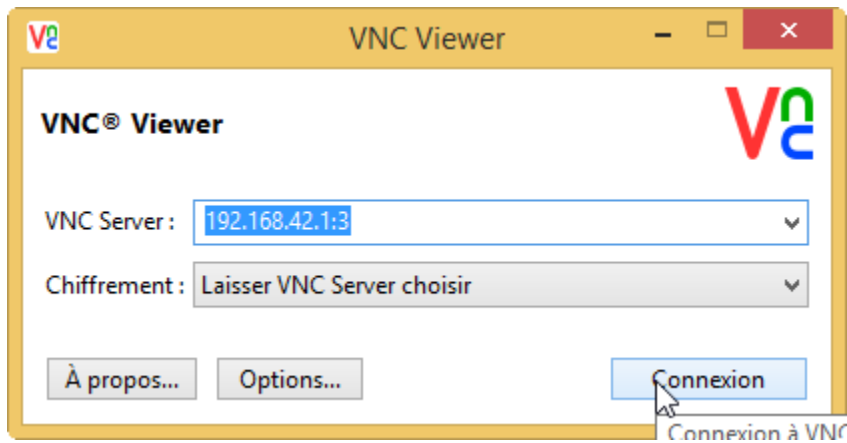
VNC Viewer for Windows (5.3.1)
32-bit (exe) - 3.4 MB

Installer tightvncserver sur la cible

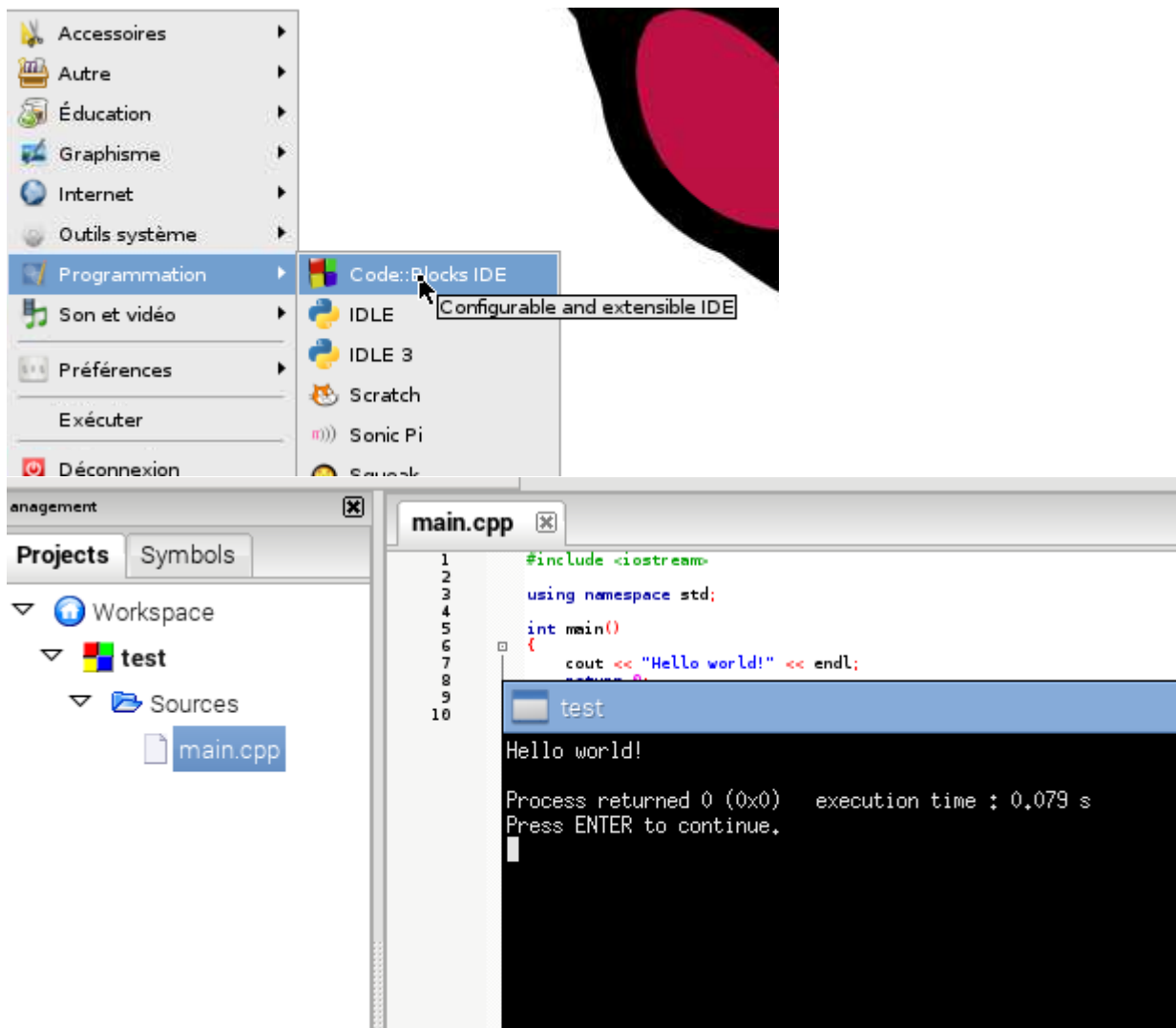
```
sudo apt-get install tightvncserver
```

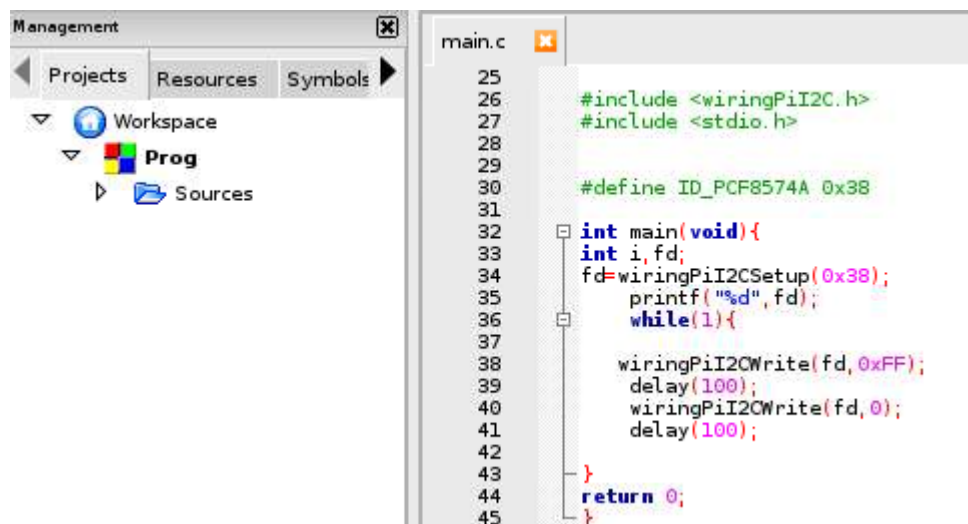
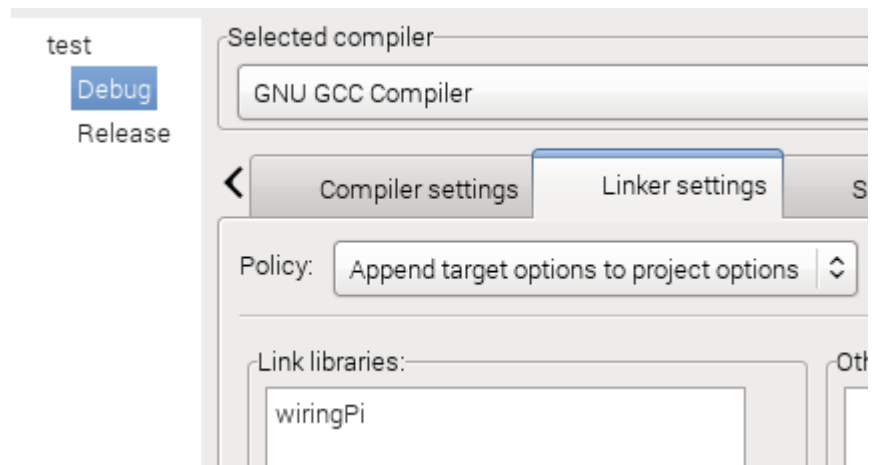
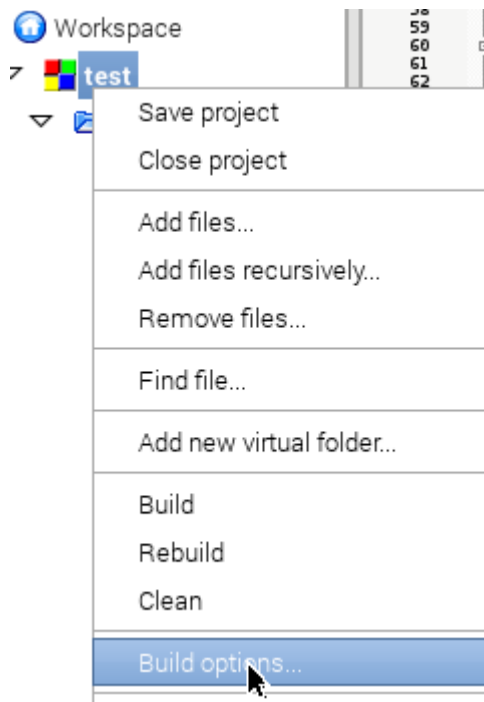
Lancer tightvncserver

```
pi@raspdom ~ $ tightvncserver
New 'X' desktop is raspdom:3
Starting applications specified in /home/pi/.
Log file is /home/pi/.vnc/raspdom:3.log
```

Lancer codeblock et tester





Autre solution :

```
#include <pcf8574.h>
#include <wiringPi.h>

int main(){
    pcf8574Setup(100, 0x38);

    while(1)
    {
        digitalWrite(100, 1);
        delay(100);
        digitalWrite(100, 0);
        delay(100);
    }

    return 1;
}
```

E.2) Programme wiringpi complet

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <wiringPiI2C.h>
#include <pcf8591.h>
```

```

#include <wiringPiSPI.h>

#define ID_PCF8574A 0x38

//wiringPI v2
//http://wiringpi.com/download-and-install/
// how to use all the things in the salvat board

void setup(){
//initialisation : Cf http://wiringpi.com/reference/setup/
// This fucntion must be called first to use of wiring Lib
    if(wiringPiSetup()==-1)
        exit(1);
// init I2C - PCF8574A
//to see the address of I2C slave : i2cdetect -y 1
    fd=wiringPiI2CSetup(0x38);
//init SPI
    wiringPiSPISetup(0,1000000);
//init ADC PCF8591 I2C slave
//100 is a number>64 and this number is used for analogRead function...
//Cf http://wiringpi.com/extensions/i2c-pcf8591/
    pcf8591Setup (100, 0x4C) ;
// pin configuration

    pinMode(4,OUTPUT); // in ver2 pin4 = GPIO23
    pinMode(5,OUTPUT); // in ver2 pin5 = GPIO24
    pinMode(6,OUTPUT); // in ver2 pin6 = GPIO25
    pinMode(7,INPUT); // in ver2 pin7 = GPIO7
    pinMode(1,PWM_OUTPUT); //in ver2 pin1=GPIO18
}

int main()
{
    int i,pushButton,pushButton_av=1,cpt,fd;
    int tension;
    float fTension;
    unsigned char spiData[2];
    printf("Hello world!\n");
    setup();
//loop
    while(1)
    {
//reading SPI MCP3201 ADC 12 bits
        wiringPiSPIDataRW(0, spiData, 2) ;
//converting to integer [0 .. 4095]
        tension=((spiData[0]&0x1F)<<7) + (spiData[1]>>1);
//converting to voltage : Vref=2.5V
        fTension=tension/4096.0*2.5;
//use potentiometer to see the change on board
        printf("%4d ***** %4.2f | ",tension,fTension);
//print the 4 analog channel from PCF8591
        printf("%4d %4d %4d %4d\n ",analogRead(100),analogRead(101),analogRead(102),analogRead(103));
//chenillard + led on PCF8574A blinking
        for (i=4;i<=6;i++){
            digitalWrite(i,HIGH);
            wiringPiI2CWrite(fd,0xFF);
            delay(50);
            digitalWrite(i,LOW);
            wiringPiI2CWrite(fd,0);
            delay(50);
        }
//PWM on PIN1
        if(cpt>1000)
            cpt=0;
        else
            cpt=cpt+100;
        pwmWrite(1,cpt);
//every 300ms test pushbutton
    }
}

```

```
        pushButton=digitalRead(7);
        if(pushButton>pushButton_av)
        {
            puts("released");
        }
        pushButton_av=pushButton;
    }
    return 0;
}
```

F) A Finir

Installation from scratch

```
sudo apt-get update && sudo apt-get upgrade
```

```
sudo apt-get install apache2 apache2-utils
```

```
sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache php5-mysql php5-curl php5-gd
```

That's it, PHP 5 is now installed. We can test it to make sure it's working by creating a index.php file and calling PHP Info. Change the directory to the default document root:

```
cd /var/www/
```

```
sudo echo "<?php phpinfo(); ?>" | sudo tee index.php
```

POUR créer le fichier de test index.php

Et pour finir

```
sudo apt-get install mysql-server
```

On pourra ensuite installer phpmyadmin

Et créer un lien ...

```
pi@raspdom /var/www $ ls -l
total 8
-rw-r--r-- 1 pi pi 177 mai  9 18:55 index.html
-rw-r--r-- 1 pi pi  20 mai  9 20:15 index.php
lrwxrwxrwx 1 root root  21 mai  9 20:46 phpmyadmin -> /usr/share/phpmyadmin
```

Pour installer i2c, spi et onwire

sudo raspi-config

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem          Ensures that all of the SD card s
2 Change User Password       Change password for the default u
3 Enable Boot to Desktop/Scratch Choose whether to boot into a des
4 Internationalisation Options Set up language and regional sett
5 Enable Camera              Enable this Pi to work with the R
6 Add to Rastrack            Add this Pi to the online Raspber
7 Overclock                  Configure overclocking for your P
8 Advanced Options           Configure advanced settings
9 About raspi-config         Information about this configurat
```

```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan                  You may need to configure oversca
A2 Hostname                   Set the visible name for this Pi
A3 Memory Split               Change the amount of memory made
A4 SSH                        Enable/Disable remote command lin
A5 Device Tree                Enable/Disable the use of Device
A6 SPI                        Enable/Disable automatic loading
A7 I2C                        Enable/Disable automatic loading
A8 Serial                     Enable/Disable shell and kernel m
A9 Audio                      Force audio out through HDMI or 3
A0 Update                     Update this tool to the latest ve
```

Ca met à jour le fichier /boot/config.txt

```
dtoverlay=spi=on

# Uncomment this to enable
#dtoverlay=lirc-rpi

# Additional overlays and p

dtoverlay=i2c arm=on
```

Le point d'accès wifi

Le fichier /etc/default/ifplugd

```
INTERFACES="eth0"
HOTPLUG_INTERFACES="eth0"
ARGS="-q -f -u0 -d10 -w -I"
SUSPEND_ACTION="stop"
```

Le fichier /etc/default/locale

```
LANG=fr_FR.UTF-8
LANGUAGE=fr_FR:fr
LC_ALL=fr_FR.UTF-8
```

/etc/default/isc-dhcp-server

INTERFACES="wlan0"

/etc/network/interface

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet static
address 192.168.42.1
netmask 255.255.255.0

#auto wlan1
#allow-hotplug wlan1
#iface wlan1 inet manual
#wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

up iptables-restore < /etc/iptables.ipv4.nat
```

/etc/iptables.ipv4.nat

Generated by iptables-save v1.4.14 on Sun May 8 18:58:36 2016

*filter

:INPUT ACCEPT [7:900]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [3:408]

-A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT

-A FORWARD -i wlan0 -o eth0 -j ACCEPT

COMMIT

Completed on Sun May 8 18:58:36 2016

Generated by iptables-save v1.4.14 on Sun May 8 18:58:36 2016

*nat

:PREROUTING ACCEPT [2:524]

:INPUT ACCEPT [2:524]

:OUTPUT ACCEPT [0:0]

:POSTROUTING ACCEPT [0:0]


```
-A POSTROUTING -o eth0 -j MASQUERADE
```

```
COMMIT
```

```
# Completed on Sun May 8 18:58:36 2016
```

```
/etc/dhcp/dhcpd.conf
```

```
authoritative;
```

```
subnet 192.168.42.0 netmask 255.255.255.0 {
```

```
range 192.168.42.10 192.168.42.50;
```

```
option broadcast-address 192.168.42.255;
```

```
option routers 192.168.42.1;
```

```
default-lease-time 600;
```

```
max-lease-time 7200;
```

```
option domain-name "local";
```

```
option domain-name-servers 8.8.8.8, 8.8.4.4;
```

```
}
```

```
/etc/hostapd/hostapd.conf
```

```
interface=wlan0
```

```
wpa_passphrase=raspberry
```

```
ssid=raspberryDomotique
```

install de samba et <http://www.framboise314.fr/prenez-la-main-a-distance-sur-votre-raspberry-pi-avec-vnc/>

http://www.linux-france.org/~eprigent/Samba_user_HTML/page3.html (comment installer samba)

```
pi@raspdom1:/var/log/samba $ sudo smbpasswd -a pi
New SMB password:
Retype new SMB password:
Added user pi.
```

Compilation croisée dans ubuntu

```
git clone https://github.com/raspberrypi/tools
```

<http://stackoverflow.com/questions/19162072/installing-raspberry-pi-cross-compiler>

compilation distribuée : à tester

<http://jeremy-nicola.info/portfolio-item/cross-compilation-distributed-compilation-for-the-raspberry-pi/>

Compilation sous windows

Netbean

Avoir un compte root avec lequel on peut se connecter pour lancer les programmes

```
sudo passwd root
```

Now you can log into your pi as the root user so give NetBeans the username 'root' and the password you set.

Now check the ssh config

```
sudo nano /etc/ssh/sshd_config
```

Search for PermitRootLogin and change it to yes (the default "without-password" won't work – change it to "yes").

Use CTRL+X to exit and then reboot.

To undo logging in as root later, use this command

```
sudo passwd -l root
```

Changer smb.conf pour avoir /usr dans samba

[usr]

comment= usr

path=/usr

browseable=Yes

writable=no

only guest=no

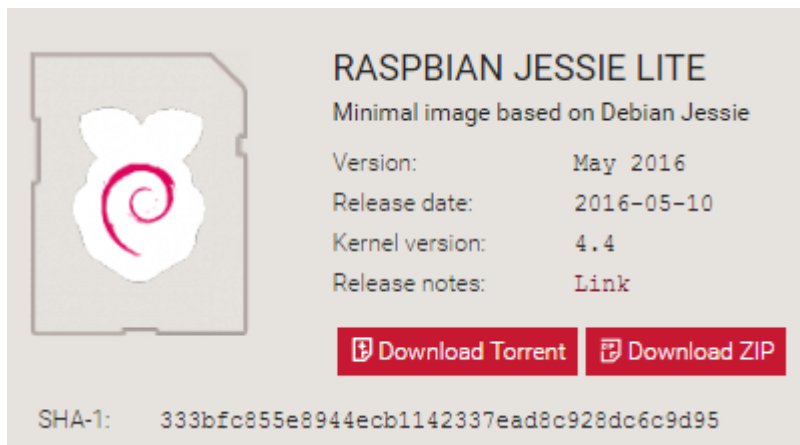
create mask=0777

directory mask=0777

public=no

From scratch

Télécharger l'image lite de Jessie



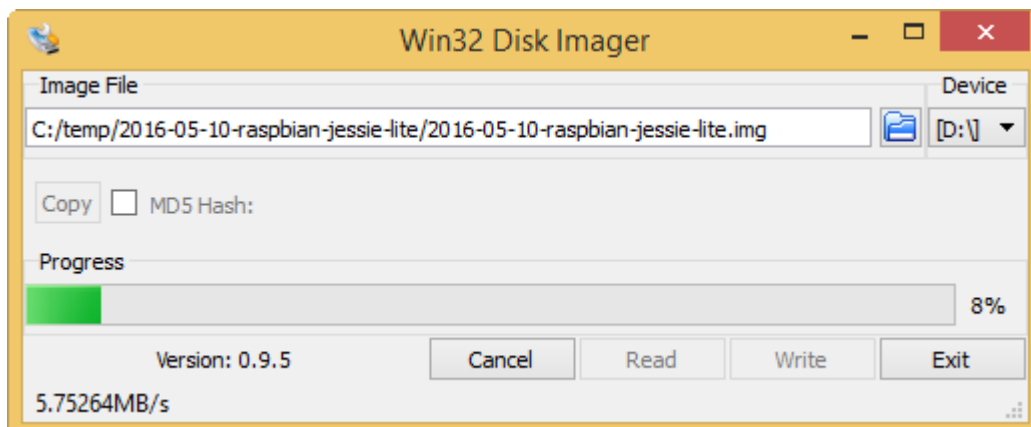
Télécharger windiskimager

Puis lancer en mode administrateur

Dézipper l'image de raspbian

Puis ouvrir l'image dans windiskimager

Cliquer sur write



Clic sur exit

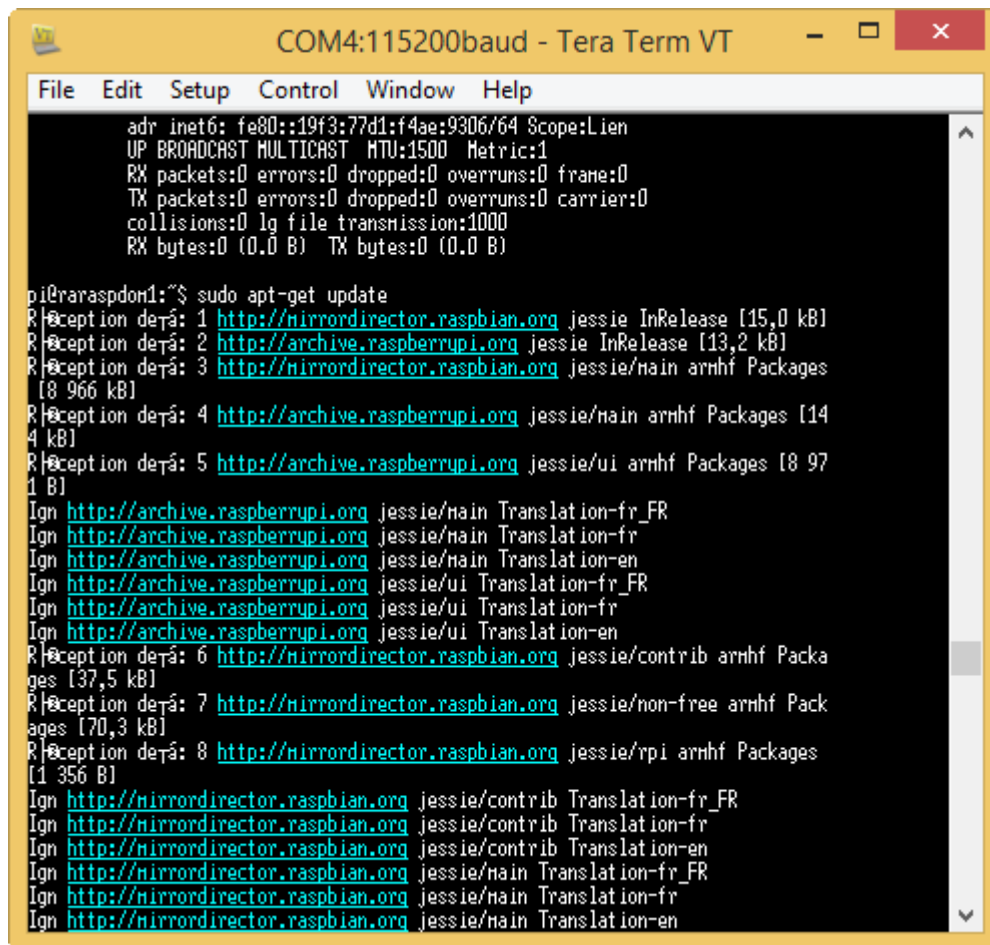
Prendre la flash et la placer sur la raspberry, démarrer

Analysons la flash : Il y a 2 partitions (une partition Vfat sur laquelle se trouve le boot et les fichiers de config que l'on peut modifier sous windows + le noyau) puis il y a le système de fichiers sur une deuxième partition linux.

Dans advanced options (i2c, onewire et spi), internationalisation (fr-UTF8)

On pourra passer en 900MHz

Sudo apt-get update



```
COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
adr inet6: fe80::19f3:77d1:f4ae:9306/64 Scope:Lien
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

pi@raspbdom1:~$ sudo apt-get update
Rception de: 1 http://mirrordirector.raspbian.org jessie InRelease [15,0 kB]
Rception de: 2 http://archive.raspberrypi.org jessie InRelease [13,2 kB]
Rception de: 3 http://mirrordirector.raspbian.org jessie/main armhf Packages
[8 966 kB]
Rception de: 4 http://archive.raspberrypi.org jessie/main armhf Packages [14
4 kB]
Rception de: 5 http://archive.raspberrypi.org jessie/ui armhf Packages [8 97
1 B]
Ign http://archive.raspberrypi.org jessie/main Translation-fr_FR
Ign http://archive.raspberrypi.org jessie/main Translation-fr
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-fr_FR
Ign http://archive.raspberrypi.org jessie/ui Translation-fr
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Rception de: 6 http://mirrordirector.raspbian.org jessie/contrib armhf Packa
ges [37,5 kB]
Rception de: 7 http://mirrordirector.raspbian.org jessie/non-free armhf Pack
ages [70,3 kB]
Rception de: 8 http://mirrordirector.raspbian.org jessie/rpi armhf Packages
[1 356 B]
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-fr_FR
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-fr
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-fr_FR
Ign http://mirrordirector.raspbian.org jessie/main Translation-fr
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
```

```
COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-fr_FR
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-fr
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
9 257 ko réceptionnés en 47s (193 ko/s)
Lecture des listes de paquets... Fait
pi@rarspdom1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:59:c2:e5
          inet addr:192.168.1.17  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::6999:f004:a116:df17/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6656 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3729 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:9654728 (9.2 MiB)  TX bytes:320622 (313.1 KiB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:HHte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:200 errors:0 dropped:0 overruns:0 frame:0
          TX packets:200 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1
          RX bytes:16656 (16.2 KiB)  TX bytes:16656 (16.2 KiB)

vuland0   Link encap:Ethernet  HWaddr 00:0f:13:48:1b:22
          adr inet6: fe80::19f3:77d1:f4ae:9306/64 Scope:Lien
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@rarspdom1:~$
```

```
pi@rarspdom1: ~
Using username "pi".
pi@192.168.1.17's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 14 00:15:12 2016
pi@rarspdom1:~$ sudo apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
pi@rarspdom1:~$
```

```
sudo apt-get install git-core

git clone git://git.drogon.net/wiringPi

cd wiringPi
```

```
./build
```

```
sudo apt-get install i2c-tools
```

```
pi@rarasdom1:~/wiringPi $ sudo apt-get install i2c-tools
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Paquets suggérés :
  libi2c-dev python-smbus
Les NOUVEAUX paquets suivants seront installés :
  i2c-tools
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 51,3 ko dans les archives.
Après cette opération, 227 ko d'espace disque supplémentaires seront utilisés
Réception de : 1 http://archive.raspberrypi.org/debian/ jessie/main i2c-tools
mhcf 3.1.1+svn-2 [51,3 kB]
51,3 ko réceptionnés en 0s (147 ko/s)
Sélection du paquet i2c-tools précédemment désélectionné.
(Lecture de la base de données... 31932 fichiers et répertoires déjà installés)
Préparation du dépaquetage de .../i2c-tools_3.1.1+svn-2_armhf.deb ...
Dépaquetage de i2c-tools (3.1.1+svn-2) ...
Traitement des actions différées (« triggers ») pour man-db (2.7.0.2-5) ...
Paramétrage de i2c-tools (3.1.1+svn-2) ...
/run/udev or .udevdb or .udev presence implies active udev. Aborting MAKEDEV
vocation.
pi@rarasdom1:~/wiringPi $ i2c
i2cdetect i2cdump i2cget i2cset
pi@rarasdom1:~/wiringPi $ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  38  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

<http://www.framboise314.fr/une-interface-web-simple-et-intuitive-pour-le-raspberry-pi-12/>

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

```
cd /var/www/
```

```
sudo chown -R pi /var/www
```

```
sudo nano index.php
```

```
<?php
phpinfo ();
?>
```

```
CTR+X o
```

Installation de pimonitor, nodejs

Download Node.js source Raspberry Pi Model A, B, B+ and Compute Module

```
wget https://nodejs.org/dist/v4.0.0/node-v4.0.0-linux-armv6l.tar.gz
```

```
tar -xvf node-v4.0.0-linux-armv6l.tar.gz
```



```
cd node-v4.0.0-linux-armv6l
```

Raspberry Pi 2 Model B

```
wget https://nodejs.org/dist/v4.0.0/node-v4.0.0-linux-armv7l.tar.gz  
tar -xvf node-v4.0.0-linux-armv7l.tar.gz  
cd node-v4.0.0-linux-armv7l
```

Copy to /usr/local

```
sudo cp -R * /usr/local/
```

ok

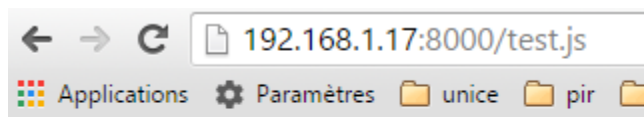
<http://www.slidequest.com/q/70ang>

<http://thisdavej.com/beginners-guide-to-installing-node-js-on-a-raspberry-pi/>

<http://revryl.com/2014/01/04/nodejs-raspberry-pi/>

```
wget http://node-arm.herokuapp.com/node_latest_armhf.deb
```

```
pi@rarpdom1:~ $ mkdir nodejs  
pi@rarpdom1:~ $ cd nodejs  
pi@rarpdom1:~/nodejs $ nano test.js  
pi@rarpdom1:~/nodejs $ node test.js  
Web Server running at http://127.0.0.1:8000
```



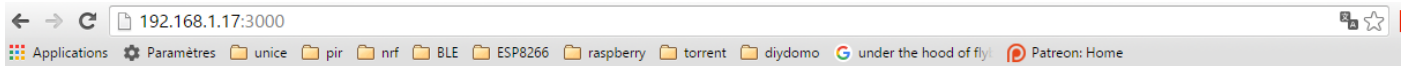
Hello World!

Git clone <https://github.com/sidwarkd/pimonitor.git>

Git clone <https://github.com/ever3001/RaspberrypiMonitoring.git>

```
pi@rarspdom1: ~/RaspberrypiMonitoring
bin      npm-debug.log      public      views
client   package.json             README.md
pi@rarspdom1:~/RaspberrypiMonitoring $ rm Étapes\ projet\ RaspberryPi.pdf
pi@rarspdom1:~/RaspberrypiMonitoring $ ls
app.js  client      package.json  public      routes
bin      npm-debug.log  pinode_stats.js  README.md  views
pi@rarspdom1:~/RaspberrypiMonitoring $ node bin/www
module.js:339
    throw err;
    ^

Error: Cannot find module 'debug'
    at Function.Module._resolveFilename (module.js:337:15)
    at Function.Module._load (module.js:287:25)
    at Module.require (module.js:366:17)
    at require (module.js:385:17)
    at Object.<anonymous> (/home/pi/RaspberrypiMonitoring/bin/www:2:13)
    at Module._compile (module.js:435:26)
    at Object.Module._extensions..js (module.js:442:10)
    at Module.load (module.js:356:32)
    at Function.Module._load (module.js:311:12)
    at Function.Module.runMain (module.js:467:10)
pi@rarspdom1:~/RaspberrypiMonitoring $ npm install
```



PiMonitor

CPU Usage (3%)

Memory Usage (57%)

Total	Used
434 MB	268 MB

GPIO Controls

Led 1

ON

Led 2

ON

OFF

```
pi@rarspdom1: ~/RaspberrypiMonitoring
pi@rarspdom1:~/RaspberrypiMonitoring $ node bin/www
Client Connected
GPiOSwitch6 State Changed: true
GPiOSwitch6 State Changed: false
GPiOSwitch5 State Changed: true
GPiOSwitch5 State Changed: false
GPiOSwitch4 State Changed: true
GPiOSwitch4 State Changed: false
^Cpi@rarspdom1:~/RaspberrypiMonitoring $ sudo node bin/www
Client Connected
GPiOSwitch6 State Changed: true
GPiOSwitch6 State Changed: false
GPiOSwitch5 State Changed: true
GPiOSwitch5 State Changed: false
GPiOSwitch4 State Changed: true
GPiOSwitch4 State Changed: false
GPiOSwitch1 State Changed: true
GPiOSwitch1 State Changed: false
GPiOSwitch2 State Changed: true
GPiOSwitch2 State Changed: false
GPiOSwitch6 State Changed: true
GPiOSwitch5 State Changed: true
```

Sudo apt-get install samba samba-common-bi

<http://www.framboise314.fr/partager-un-repertoire-sous-jessie-avec-samba/>

Sudo Nano /etc/samba/smb.conf

```
[pihome]
comment = Pi Home
path = /home/pi
read only = No
```

```
create mask = 0777
directory mask = 0777

[root]
comment = root Home
path = /root
read only = No
create mask = 0777
directory mask = 0777

[usr]
comment = usr Home
path = /usr
read only = No
create mask = 0777
directory mask = 0777
```

puis écrire dans /etc/default

```
# Defaults for samba initscript
```

```
# sourced by /etc/init.d/samba
```

```
# installed at /etc/default/samba by the maintainer scripts
```

```
#
```

```
#
```

```
# This is a POSIX shell fragment
```

```
#
```

```
# How should Samba (smbd) run? Possible values are "daemons"
```

```
# or "inetd".
```

```
RUN_MODE="daemons"
```

```

root@raspdom1:/etc# service smbd restart
root@raspdom1:/etc# service smbd status
● smbd.service - LSB: start Samba SMB/CIFS daemon (smbd)
   Loaded: loaded (/etc/init.d/smbd)
   Active: active (running) since lun. 2016-05-16 16:57:28 CEST; 6s ago
 Process: 11899 ExecStop=/etc/init.d/smbd stop (code=exited, status=0/SUCCESS)
 Process: 11908 ExecStart=/etc/init.d/smbd start (code=exited, status=0/SUCCESS)
    CGroup: /system.slice/smbd.service
            └─11918 /usr/sbin/smbd -D
              └─11920 /usr/sbin/smbd -D

mai 16 16:57:28 raspdom1 smbd[11908]: Starting SMB/CIFS daemon: smbd.
mai 16 16:57:28 raspdom1 systemd[1]: Started LSB: start Samba SMB/CIFS daemon (smbd).
root@raspdom1:/etc# smbpasswd -a pi
New SMB password:
Retype new SMB password:
Added user pi.

```

Pour lancer au démarrage

```
systemctl enable smbd.service nmbd.service
```

Fichier /etc/default/ifplugd

This file may be changed either manually or by running dpkg-reconfigure.

#

N.B.: dpkg-reconfigure deletes everything from this file except for

the assignments to variables INTERFACES, HOTPLUG_INTERFACES, ARGS and

SUSPEND_ACTION. When run it uses the current values of those variables

as their default values, thus preserving the administrator's changes.

#

This file is sourced by both the init script /etc/init.d/ifplugd and

the udev script /lib/udev/ifplugd.agent to give default values.

The init script starts ifplugd for all interfaces listed in

INTERFACES, and the udev script starts ifplugd for all interfaces

listed in HOTPLUG_INTERFACES. The special value all starts one

ifplugd for all interfaces being present.

INTERFACES="eth0"

HOTPLUG_INTERFACES="eth0"

ARGS="-q -f -u0 -d10 -w -l"

SUSPEND_ACTION="stop"

sudo apt-get install isc-dhcp-server

sudo apt-get install hostapd

sudo nano /etc/default/

INTERFACE = « wlan0 »

```
sudo passwd root
```

Now you can log into your pi as the root user so give NetBeans the username 'root' and the password you set.

Now check the ssh config

```
sudo nano /etc/ssh/sshd_config
```

Search for PermitRootLogin and change it to yes (the default "without-password" won't work – change it to "yes").

```
zcat /usr/share/doc/hostapd/examples/hostapd.conf.gz | sudo tee -a /etc/hostapd/hostapd.conf
```

<http://www.molivier.com/isc-dhcp-server-debian.html>

<http://elinux.org/RPI-Wireless-Hotspot> (hostpad)

Le bon ?

<http://www.daveconroy.com/using-your-raspberry-pi-as-a-wireless-router-and-web-server/>

Resizing image file

<http://www.aoakley.com/articles/2015-10-09-resizing-sd-images.php>

Ca fonctionne , mais il faut copier le fichier image dans /tmp (ext4fs).

