Raspberry

|  |  |
| --- | --- |
| Adresse du pi raspberry donné par le DNS  : 192.168.1.13  Acces via putty ( SSH config sur le raspberry \* )    Login en pi : raspberry ou en root : frederic |  |
| Note \*: Sudo raspi-config pour acceder à la config et au ssh : | Utilisation de sshfs.exe qui peut monter un disque virtuel d’un répertoire du raspberry sous windows : voilà les 2 outils : Putty et sshfs |
|  |  |

11/07/2015

Reprise du rasp 192.168.1.17 pi /frederic

Tentative d’install de node js a partir de :

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

mise a jour du système avec $ sudo apt update et $ sudo apt full-upgrade.. qui marchent pas apt non reconnu !!

en fait ça marche avec $ sudo apt-get update et $ sudo apt-get upgrade

On fait un curl pour install le package node.js .. en fait le curl fait un download a partir d’une url :

curl -sL https://deb.nodesource.com/setup\_8.x | sudo -E bash –

ici l’url est <https://deb.nodesource.com/setup_8.x>

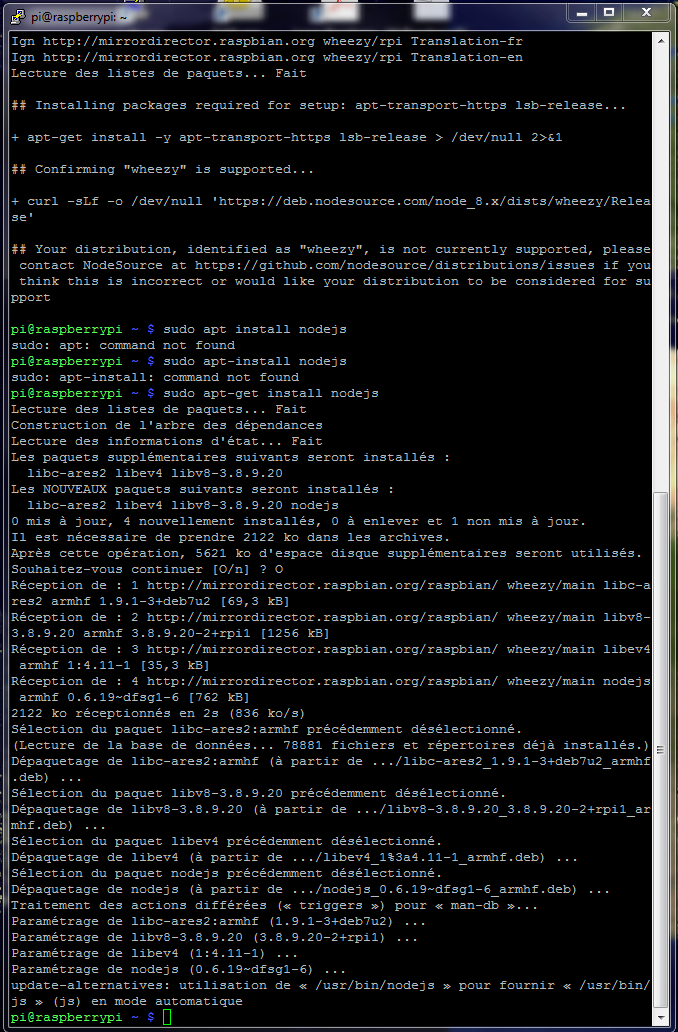
On peut y aller et voir le script..

Bon.. ca marche pas wheezy not supported.. .. faudrait une distribution debian jessie !!

Bref malgrés cela on tente de continuer :

$ sudo apt-get install nodejs

Au final ça a donné ça :



Bref ça marche pas mieux..

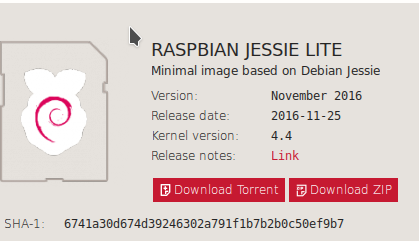
$ node –v n’est pas reconnu !

On est bon pour reinstaller l’OS en version Jessy..

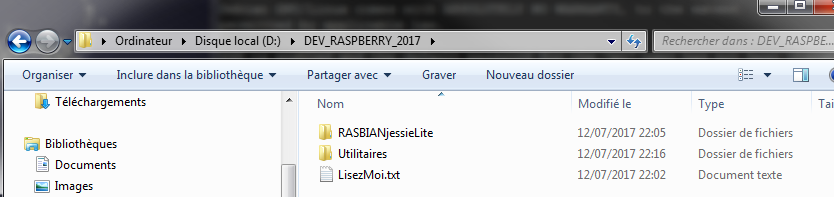
<https://www.nextinpact.com/news/96697-raspberry-pi-raspbian-passe-a-debian-8-jessie-et-integre-libreoffice.htm>

INSTALLATION DE l’OS RASPBIAN JESSIE LITE suivant :

<https://soozx.fr/nextcloud-raspberry-windows-2/installer-raspbian-raspberry-pi-ordinateur-windows/>



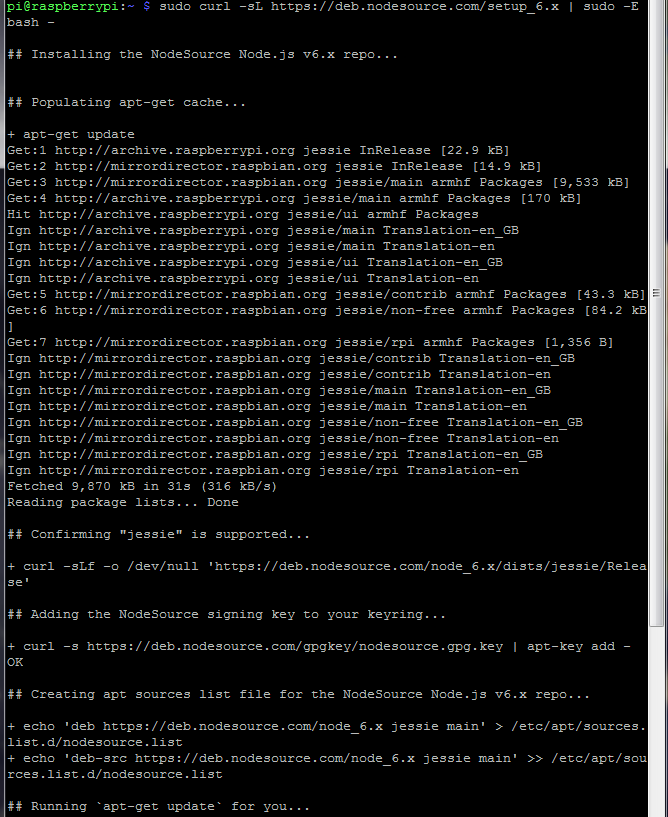
Tout est OK conforme aux instructions : répertoire des images et utilitaires :

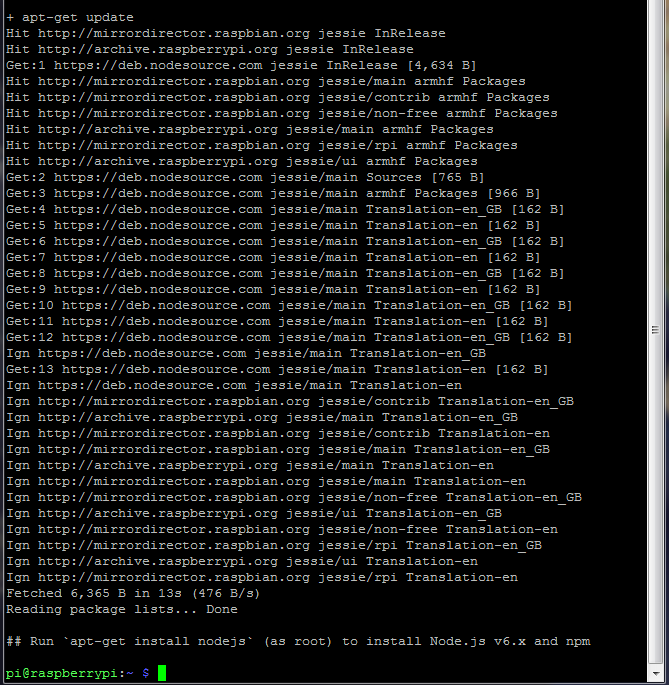


Après avoir fait un petit sudo raspi-config pour essayer de config le clavier en Azerty.. le menu du clavier n’est pas proposé dans la config a partir de putty .. il faut faire ça direct sur le raspberry qui lui voit son clavier et propose ce menu..

Reinstal de Node js par :

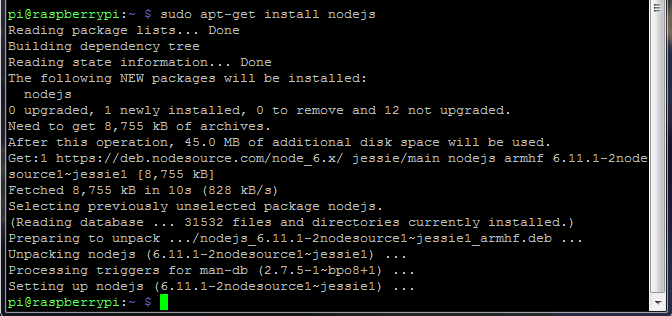
sudo curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -





Jusque là pas de message d’erreur.. la suite…

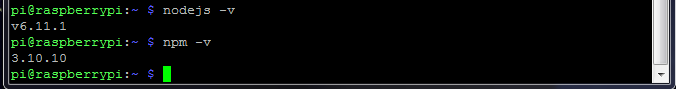
Sudo apt-get install nodejs



On verif les version install :

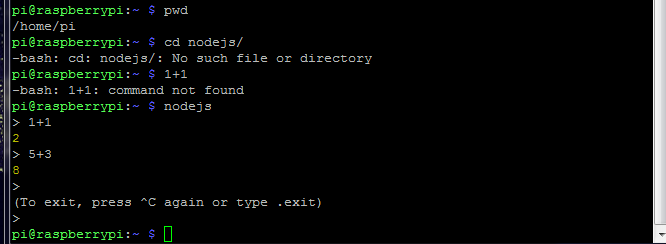
nodejs –v

npm -v

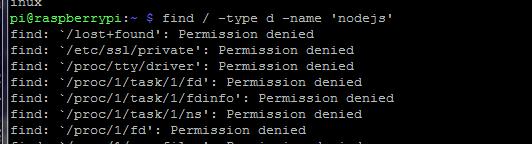


On fait qq tests…. Bien sûr on sait pas ou on est.. mais si on lance nodejs on peut executer du javascript : 1+1 c’est du javascript..

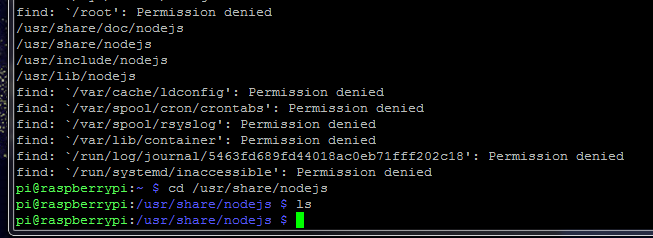
Ctrl-c 2 fois pour sortir



Pour trouver le repertoire nodejs..



Puis miracle un usr/share/nodejs.. on y va, ls pour voir ce qu’il y a .. y a rien..



On s’ouvre une session sshs et on va rajouter ce répertoire.. ca rajoute dans l’explorateur windows mais on peut rien faire dessus.. pas de droits !!

|  |  |
| --- | --- |
|  | En fait y a pas de permission niveau linux sur ce repertoire.. |

Bon..

mkdir testnodejs

cd testnodejs

|  |  |
| --- | --- |
|  | On revient sur son rep racine avec le cd~  Verif avec pwd qu’on est sous /home/pi  Creation d’un repertoire testnodejs puis ajout d’un fichier.. |

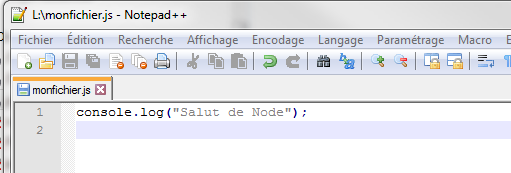
On verif le répertoire :



On revient dans SShs , on monte un disque sur ce rep.. et on voit bien le contenu..

|  |  |
| --- | --- |
|  |  |

Qu’on peut editer avec Notepad par exemple :



Puis on revient sous putty  et on execute notre fichier js sous nodejs..



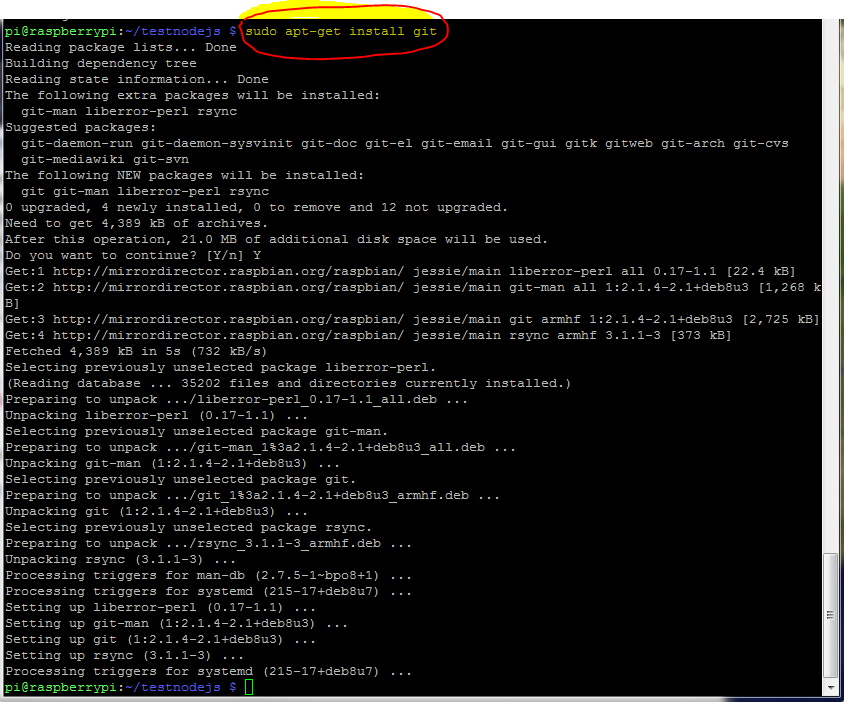
Pour aller plus loin :

<http://www.webondevices.com/install-node-js-on-a-raspberry-pi-and-run-javascript/>

On va essayer la sequence d’install suivante :

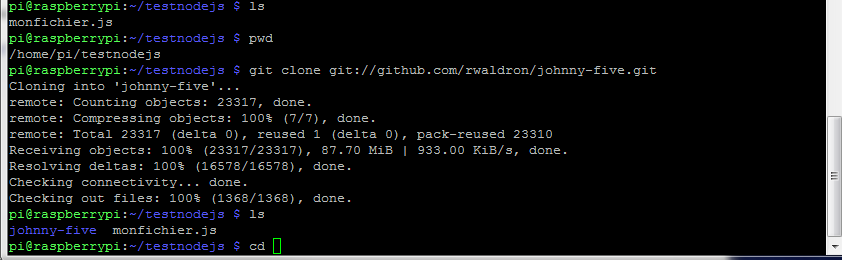


Mais avant faut installer git pour récup de projets :



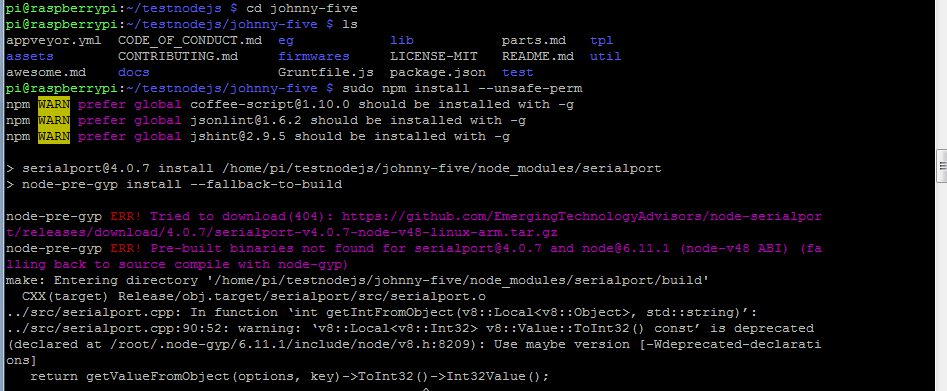
Maintenant on installe la librairie johny-five : <https://github.com/rwaldron/johnny-five>

En se plaçant dans notre rep de travail testnodejs

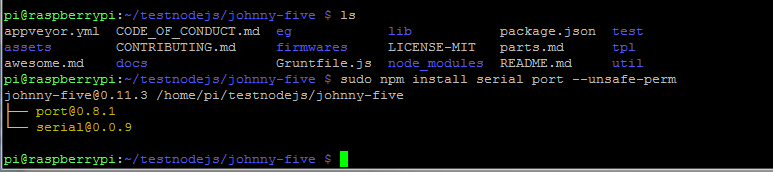


On vient de cloner le dépôt git sous le répertoire dans un repertoire : johnny-five.

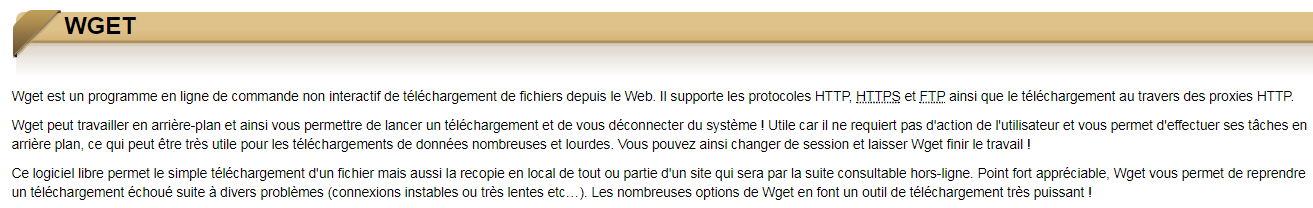
On pousuit l’install : y a qq erreurs..

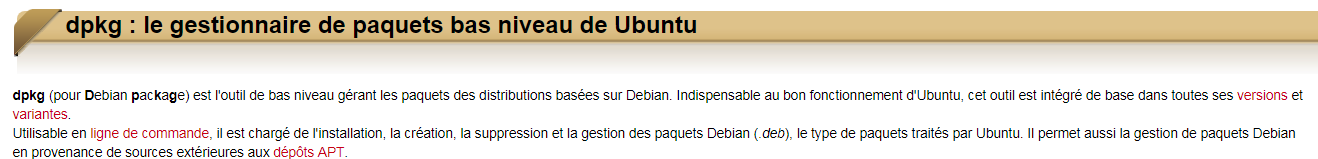


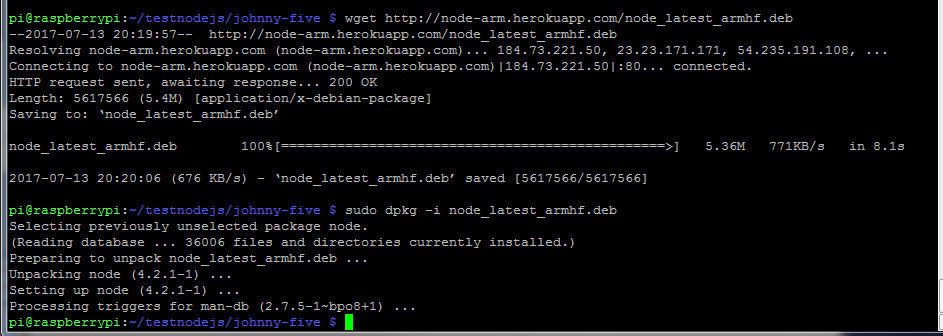
La suite beaucoup plus rapide ..:



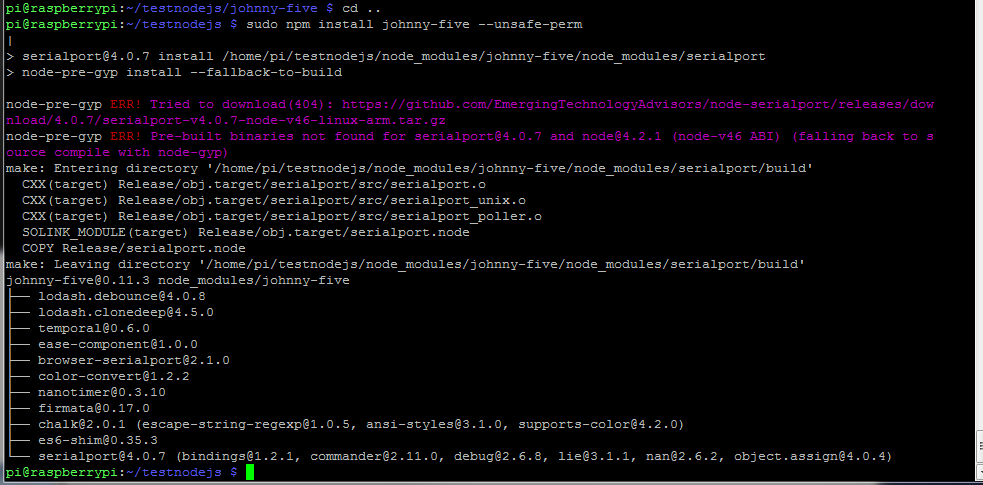
On telecharge (wget) un paquet debian .. puis on (dpkg)



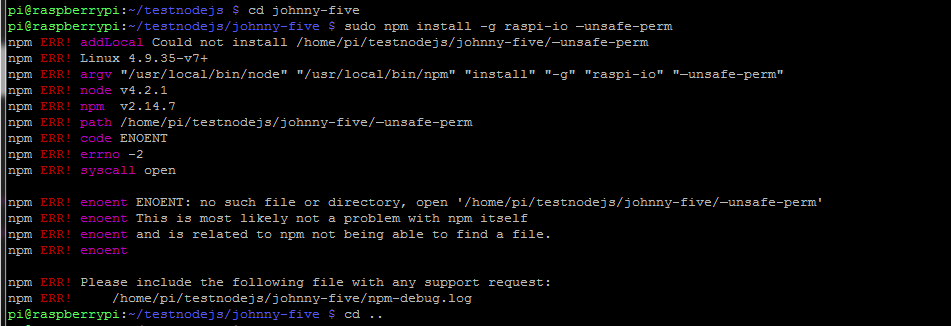




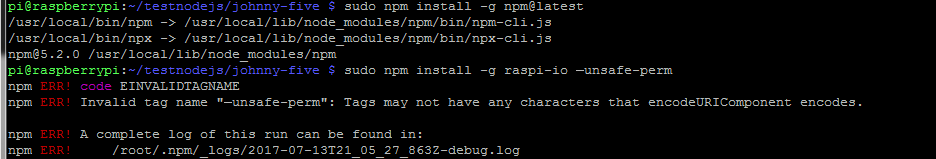
On revient au repertoire parent cd .. et on install jonhnyy-five



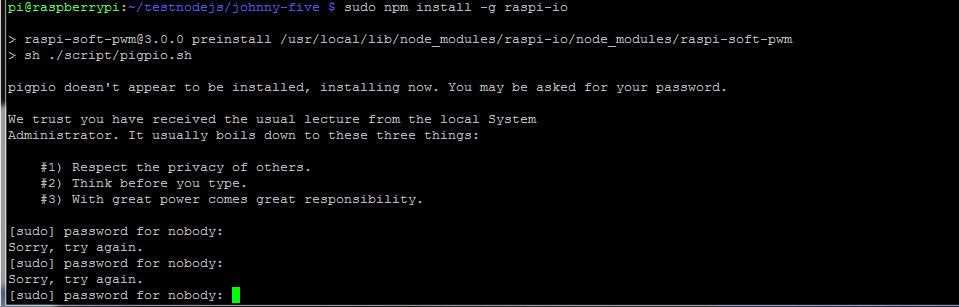
La dernière install semble avoir échoué.



Peut etre pb avec npm.. on charge last version.. puis on relance la commande npm.. qui veut plus de unsafe-perm.. En fait erreur de frappe c’est –unsafe-perm qu’il faut.



Mais là ca demande un mot de passe..

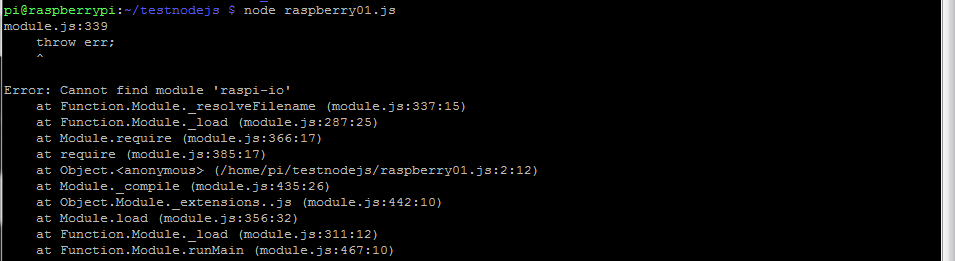


En fait ça passe direct sur le raspberry, non en remote avec putty..

Au final si on teste le code suivant : raspberry.js :

|  |
| --- |
| *var five = require("johnny-five");*  *var Rasp = require("raspi-io");*  *var board = new five.Board({*  *io: new Rasp()*  *});*  *board.on("ready", function() {*  *var led = new five.Led(0);*  *led.blink();*  *});* |

Ca donne ça..

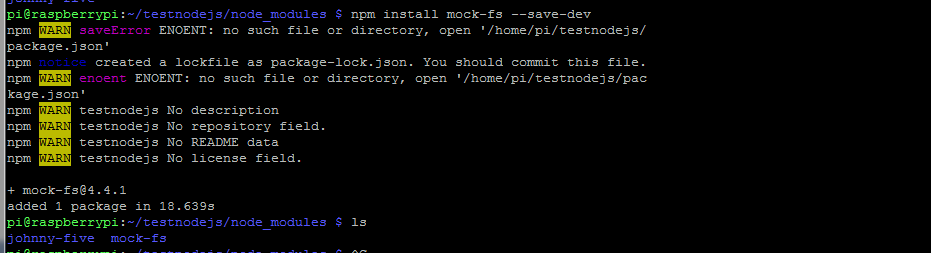


On essaye autre chose :

<https://www.sitepoint.com/getting-started-with-the-raspberry-pi-gpio-pins-in-node-js/>

Faut installer une lib plus légère , on va se servir des propriétés des fichiers sous linux et gérer le port GPIO comme un fichier !! :

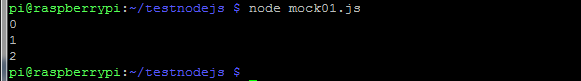
<https://www.npmjs.com/package/mock-fs>



Test sur un fichier js , voir si ca passe sans erreurs..

|  |
| --- |
| *var mock = require('mock-fs');*    *console.log("0");*  *mock({*  *'path/to/fake/dir': {*  *'some-file.txt': 'file content here',*  *'empty-dir': {/\*\* empty directory \*/}*  *},*  *'path/to/some.png': new Buffer([8, 6, 7, 5, 3, 0, 9]),*  *'some/other/path': {/\*\* another empty directory \*/}*  *});*  *console.log("1");*  *// after a test runs*  *mock.restore();*  *console.log("2");* |

Ca donne ça :



Ca a pas l’air de planter..

Aller plus loin vers cette direction : <https://github.com/sitepoint-editors/TddWithGpio>

<https://www.sitepoint.com/getting-started-with-the-raspberry-pi-gpio-pins-in-node-js/>

Autre facon avec module c a rajouter !?? : <https://github.com/darrylhodgins/piface-node>

Pour utiliser du code c++ en module pour nodejs.. : <https://stackoverflow.com/questions/9629677/how-can-i-use-a-c-library-from-node-js>

|  |
| --- |
| 14/07/2017 ENTRE temps petite vérif du fonctionnent de node js  avec l’aspect serveur : |

On va faire un serveur simple en s’inspirant de ce lien pour connecter une camera au raspberry :

<http://www.latechamienoise.com/blog/conferences/un-robot-connecte-avec-nodejs-et-raspberry-pi.html>

Connexion avec putty et sshfs comme d’habitude. On crée sous notre répertoire de test node /home/pi/testnodejs un fichier HelloWord01.js :

|  |  |
| --- | --- |
|  | var http = require('http');  http.createServer(function (req, res) {  res.writeHead(200, {'Content-Type': 'text/plain'});  res.end('Hello World\n'); })  .listen(80);    console.log('Server running'); |

On lance le serveur Node sur le raspberry.. Attention avec les privilèges sudo.. Et on va sur le navigateur a l’adresse du raspberry..

|  |  |
| --- | --- |
|  |  |

Petit test en changeant le port d’ecoute..

|  |  |
| --- | --- |
|  |  |

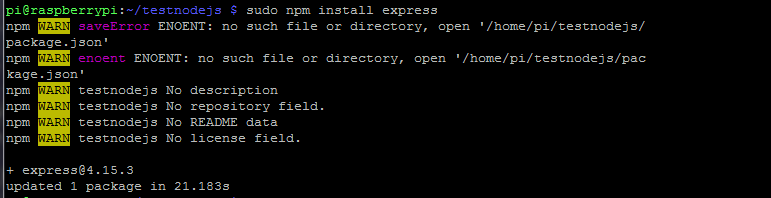
Du coup on va plus loin sur la partie purement serveur de node js  et avec de l’angular en plus..

**CE SITE EST FANTASTIQUE pour les fondements par l’exemple de nodejs. Coté serveur .**

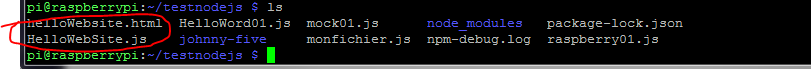
<https://www.element14.com/community/community/raspberry-pi/raspberrypi2/blog/2015/03/25/rpi2-nodejs-expressjs-sockjs-angularjs>

**Installation du package express pour faire des sites..**

**Sudo npm install express**



Premier site avec 2 fichiers un js et un html..

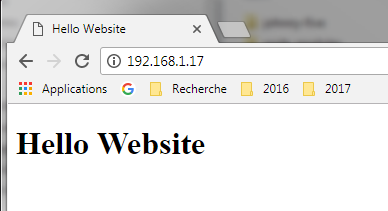


|  |  |
| --- | --- |
| **helloWebSite.js** | **helloWebsite.html** |
| var http = require('http');  var express = require('express');  var app = express();  app.get('/', function (request, response) {  response.sendFile(\_\_dirname + '/helloWebsite.html');  });  var port = 80;  http.createServer(app).listen(port); | <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta charset="utf-8" />  <title>Hello Website</title>  </head>  <body>  <h1>Hello Website</h1>  </body>  </html> |

Lancement du serveur :

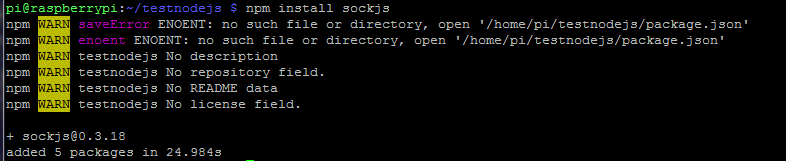


Et résultat



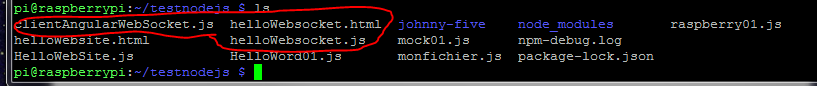
**Installation du package Sock.js pour faire du Websocket..**

**Npm install sockjs**

****

Là on touche au divin.. pour l’instant tout fonctionne..

On va créer 3 fichiers :



|  |  |  |
| --- | --- | --- |
| helloWebsocket.js | helloWebsocket.html | clientAngularWebSocket.js |
|  |  |  |

On lance le serveur et on lance le navigateur .. voilà, on fait de la single page appli avec angular et websocket sur raspberry !! :

|  |  |
| --- | --- |
|  |  |

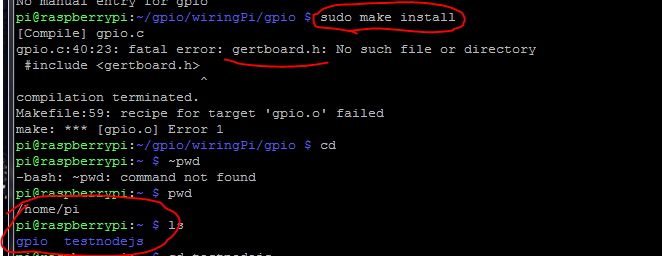
On va jusqu'à la réalisation d’un chat sur Raspbery.. donc voilà un bon exercice pour comprendre la partie serveur pour site web de node js.

19/07/2017 juste après minuit

Juste tentative d’installation de gpio d’après :

<http://www.pobot.org/Entrees-sorties-GPIO-sur-Raspberry.html>

pour s’en servir en ligne de commande et vérif. les câblages.. mais ça plante a la fin de l’install..



Sous le rep de travail on a crée le rep gppio pour faire cette install.. a voir pour effacer et recommencer autre install..

19/07/2017

Après et pendant une inondation de la cabane.. PB sur le boot du raspberry qui check et qui finit par avoir perdu des configs.. repasse en querty, git marche pas bien..

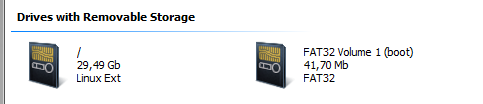
Bref.. on va devoir réinstaller.. Mais il pleut trop et y en a marre d’éponger.

20/07/2017

On regarde ce qu’il y a sur la carte mémoire de notre PI2 de test récemment installé avec Rasbian Jessie.. Sur le PC windows dans le lecteur de carte avec DiskInternalsLinuxReader :



On a ça :



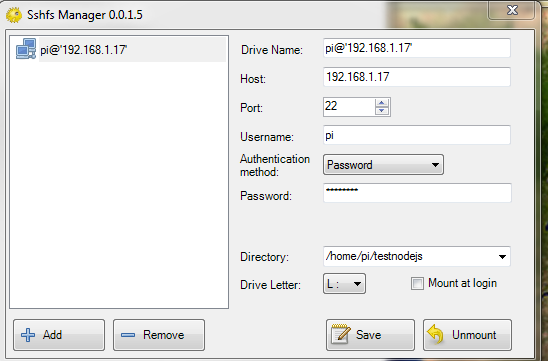
|  |  |
| --- | --- |
|  |  |
|  |  |

Tout a l’air de se passer en partition Linux Ext.

Paradoxe.. quand on va sous home/pi.. y a rien..

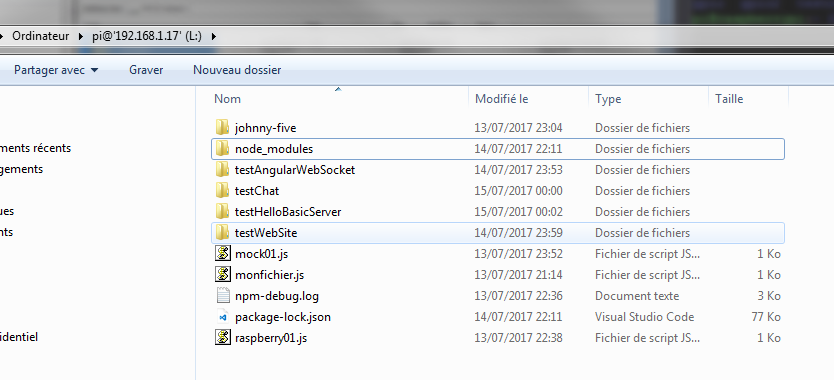
On revient sur le Raspberry avec cette carte SIM on relance.. avec les erreurs de démarrage [FAILED] et le clavier en anglais..

On ouvre une session putty et sshfs pour monter le répertoire /home/pi/testnodejs..



Ca marche toujours !!

Et là on retrouve nos billes :

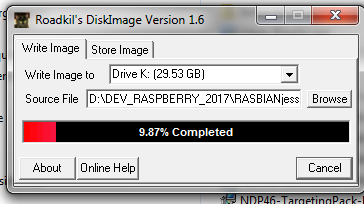


On laisse tomber johnny-five, node\_modules mais on va se récupérer nos fichiers sources de test js..

On repart de zero.. mais Win32Diskne fonctionne plus !!

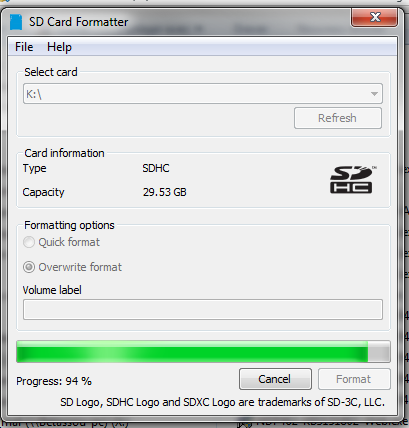
ON va sur : <http://www.roadkil.net/program.php?ProgramID=12>

Mais là ca plante a l’ecriture



On va formatter ..pour ça :

On part la dessus : <https://www.sdcard.org/downloads/formatter_4/eula_windows/index.html>



Des plombes pour formatter le truc…

Bon ben l’ecriture replante sur la fin

…

**26/07/2017 Reprise des hostilités..**

**On repart de zero :**

On revient sur Win32DisImager qui finalement a pu se lancer et écrire la carte SD.

Reintall du système suivant :

<https://soozx.fr/nextcloud-raspberry-windows-2/installer-raspbian-raspberry-pi-ordinateur-windows/>

Avec la carte SD dans le lecteur, on n’oublit pas de faire un cmd > aller sous la carte SD : K : (lettre du lecteur SD)

Puis entrer : type NUL > ssh

On peut ensuite mettre la carte dans le PI puis lancer.

On aura le clavier en Anglais. Faire un sudo raspi-config sur le PI pour modifier le clavier.

Changer le pasword avec un sudo passwr, sinon on est avertit tout le temps du pb de sécurité.

Puis on va sous putty. On se retrouve sous home/pi

On installe node js :

sudo curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash –

sudo apt-get install nodejs // cela installe node js et la gestion de package de node : npm

Verif des version :

nodejs –v

npm -v

Test en lancant nodejs : on peut executer du javascript : 1+1 c’est du javascript..

Ctrl-c 2 fois pour sortir

On va créer un rep de travail :

mkdir testnodejs

cd testnodejs // on se déplace sous ce répertoire

On créer un disque partagé avec sshfs sur home/pi/testnodejs , on peut alors synchroniser des fichier entre Windows et le PI.

On copie / colle les repertoires de source js.. on teste..

Pour cela on doit installer des package de node avec les commande ( sudo ) sous le repertoire testnodejs :

**sudo npm install express**

sudo n**pm install sockjs**

**Tout est re testé et fonctionne …**

**Maintenant que faire avec le GPIO … c’est là ou ça a merdé dans la première install avec les différentes manières d’install..**

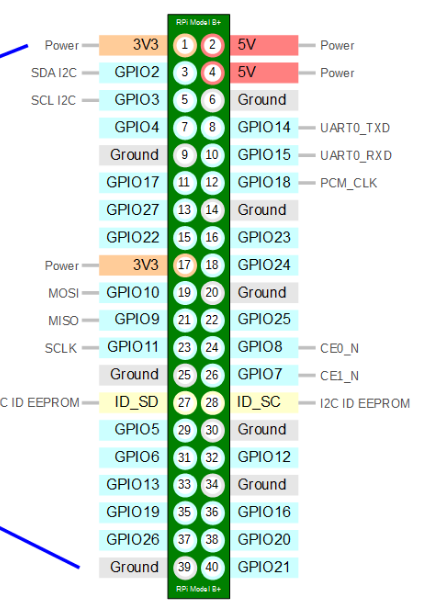
**Bon on va essayer un module qui es fourni par npm.. le plus simple possible :**

**https://www.npmjs.com/package/rpi-gpio**

npm install rpi-gpio

Un petit essai avec le pb de confusion du pin out du GPIO..

On a le GPIO#17 de cablé ..pour l’atteindre avec le code : faut adresser la pin 11..



Petite modif du programme bonjour.js :

|  |
| --- |
| var gpio = require('rpi-gpio');  // gpio.setup(11, gpio.dir\_out, write);  // function write(){  // gpio.write(11, true, function(err){  // if(err){  // console.log("erreur"+ err);  // }  // else{  // console.log("ok");  // }  // });  // } |

ET MIRACLE LA LED S’ALLUME !!!

01/08/2017 Entre temps mise en route d’un routeur WIFI

ENTRE TEMPS CONFIG D’UN ROUTEUR REPETEUR WIFI :



|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

OK pour la tablette en WIFI sur le point d’acces TRENDnet637..

Mais des pb avec la com internet du PC !!

|  |  |
| --- | --- |
| SANS LE ROUTEUR | AVEC ROUTEUR |
|  |  |
|  | PB de ping sur 192.168.1.1  Aleatoire.. |

BON CA deconne complet.. Une grosse m.. ce routeur.. qui veut même pas se reseter et se reconfigurer.. va falloir en changer

BON pour se détendre !! on va se remettre au C..

|  |  |
| --- | --- |
|  |  |

Voilà un hello word program en c : compil gcc puis exécution par ./hello

02/08/2017 :

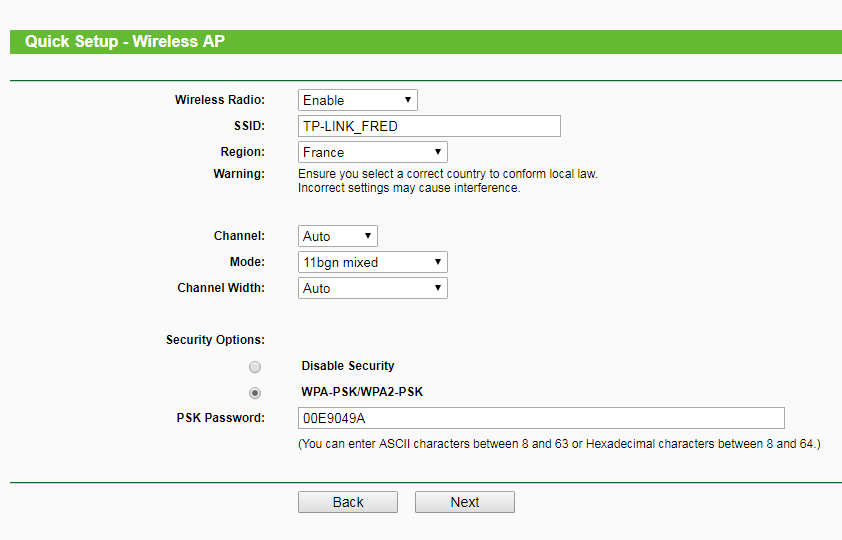
Retour sur WIFI..

TPLINK :

Pour arriver a cette page de config il faut changer l’IP du pc dans la plage 192.168.0.x

Puisque par defaut le TPLink est en 192.168.0.254

Login : admin , password : admin



On le met en mode AP : Access point.. puis changement d’adresse en 192.168.1.254… et là plus personne..

Bon.. je reviendrais la dessus demain..

03/08/2017 :

Install de fritzing pour faire des PCB de circuit imprimé..

Install de la camera Raspberry suivant :

<https://www.raspberrypi.org/documentation/configuration/camera.md>

c’est OK, on récupère bien une pic en jpeg sous le répertoire pi/home :



Ben voilà le résultat..

Puis .. utilisation de la camera via nodejs sur un client Web… suivant..

<http://thejackalofjavascript.com/rpi-live-streaming/>

et ça marche !!! mieux que le chien ..

Lancement de fritzing.. manque des dll de windows… faut faire mise a jour… des plombes..

Résultats après mise a jour de windows.. fritzing ça marche mais sshfs ne marche plus !!!

Même après reinstallation :

<https://code.google.com/archive/p/win-sshfs/downloads>

marche plus.. MERCI WINDOWS..

04/08/2017 :

Puisque sshfs ne marche plus sous win7.. c’est un signe pour passer a linux..

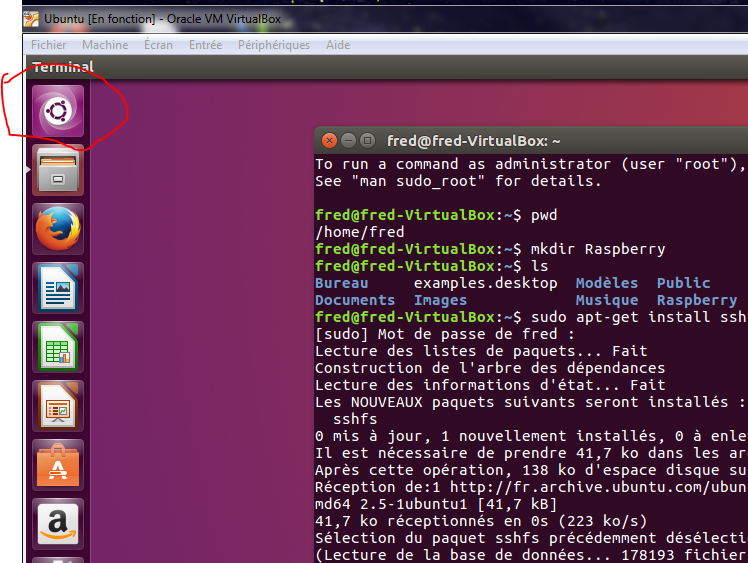
Installation d’une machine virtuelle sous Oracle Virtual Box en suivant :

<https://perso.univ-rennes1.fr/pierre.nerzic/SYS1A/Installation%20VirtualBox%20Ubuntu.pdf>

Maintenant on a linux sous raspberry en ligne de commande .. et ubuntu sur le PC..

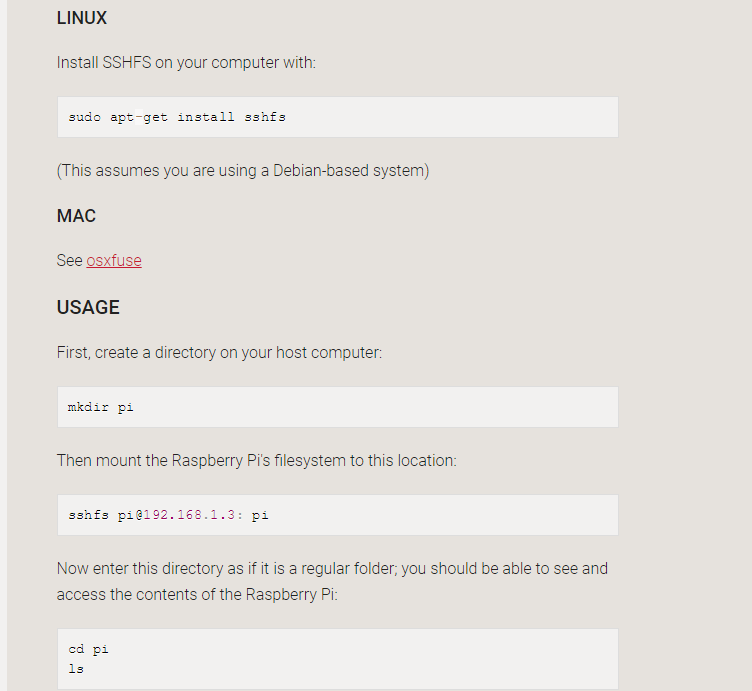
Première chose installer putty et monter un disque pour le raspberry..

Le terminal..pas dans les raccourcis.. on va le chercher par là en tapant terminal :



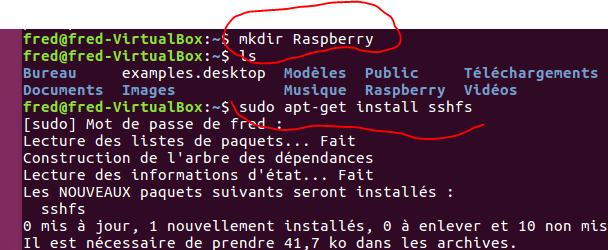
Pour sshfs voir :

<https://www.raspberrypi.org/documentation/remote-access/ssh/sshfs.md>



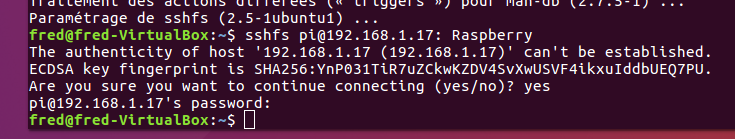
La variante utilisé ici est : création d’un répertoire de partage Raspberry

Puis l’install de sshfs

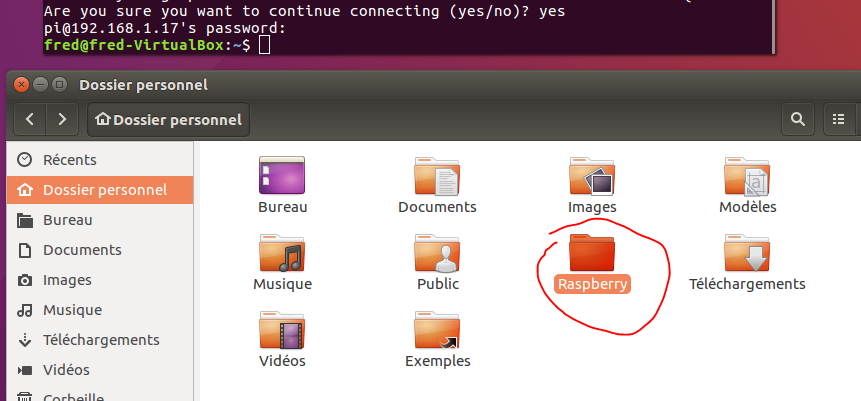


Ensuite on monte le disque par sshfs [pi@192.168.1.17:Raspberry](mailto:pi@192.168.1.17:Raspberry)

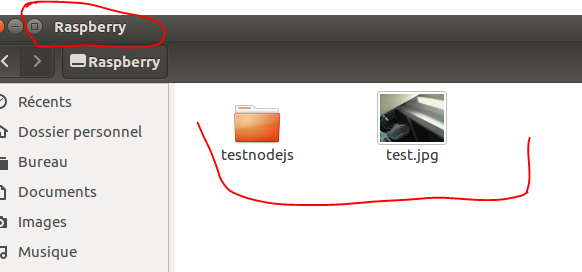
Y a quelques messages d’insultes.. mais on valide, ca demande le password du raspberry..



Et..

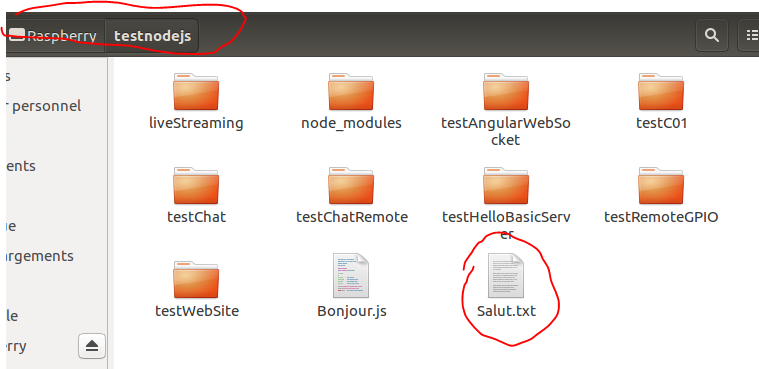


On retrouve notre arborescence du raspberry :

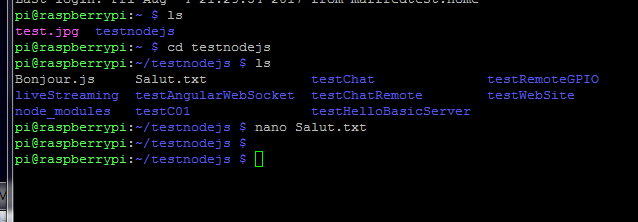


Tout comme quand ca marchait encore sous windows on peut ajoute run fichier : ici Salut.txt.

On peut l’éditer et le sauver..

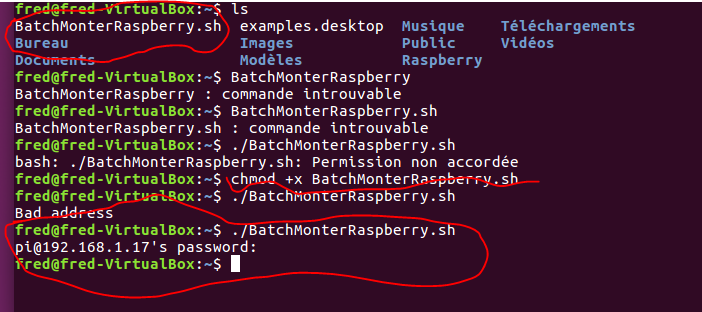


Si on revient sous le raspberry : on retrouve notre fichier un petit nano Salit.txt et on verif ce qu’il contient..



Bon on a au moins le même environnement de base sous linux Ubuntu que celui qu’on avait sous windows avant la mise a jour.

On s’essai meme a faire un fichier batch pour monter le disque..



A améliorer et lire la doc pour les fichier bach..

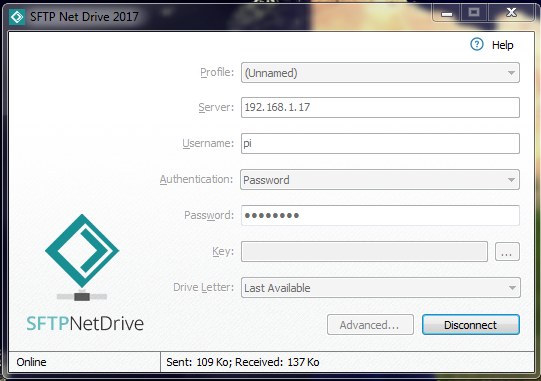
<https://openclassrooms.com/courses/reprenez-le-controle-a-l-aide-de-linux/introduction-aux-scripts-shell>

|  |
| --- |
| 06/08/2017 : |
| * Install outils ARDUINO * Install alternative sshfs |

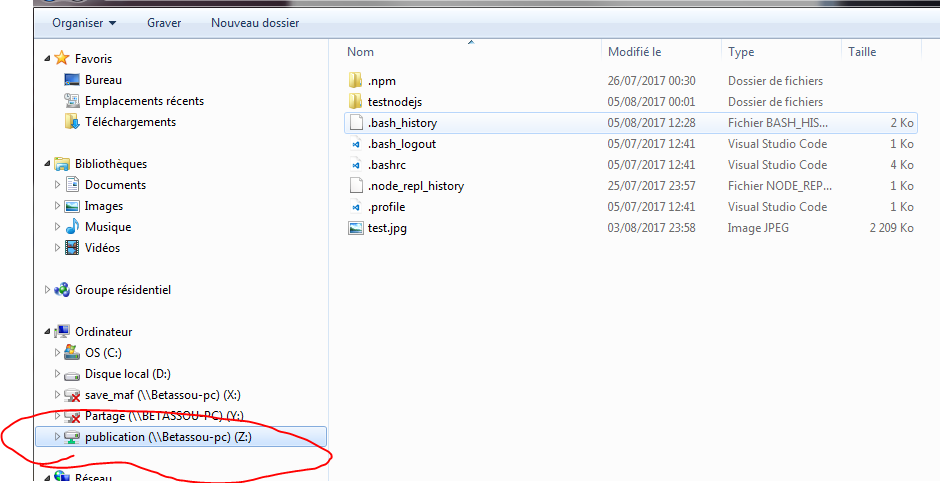
Après un petit tour sous ubuntu… pour remplacer le disfonctionnement de sshfs sous windows..

Il s’avère que le lancement de la machine virtuelle linux n’est pas sans douleur pour le PC et son temps de réponse.

EN cherchant un peu plus ce qui serait une alternative a sshfs on trouve STP Net Drive :



Qui de la même façon que sshfs monte un disque pour le raspberry directement visible sous l’explorateur :



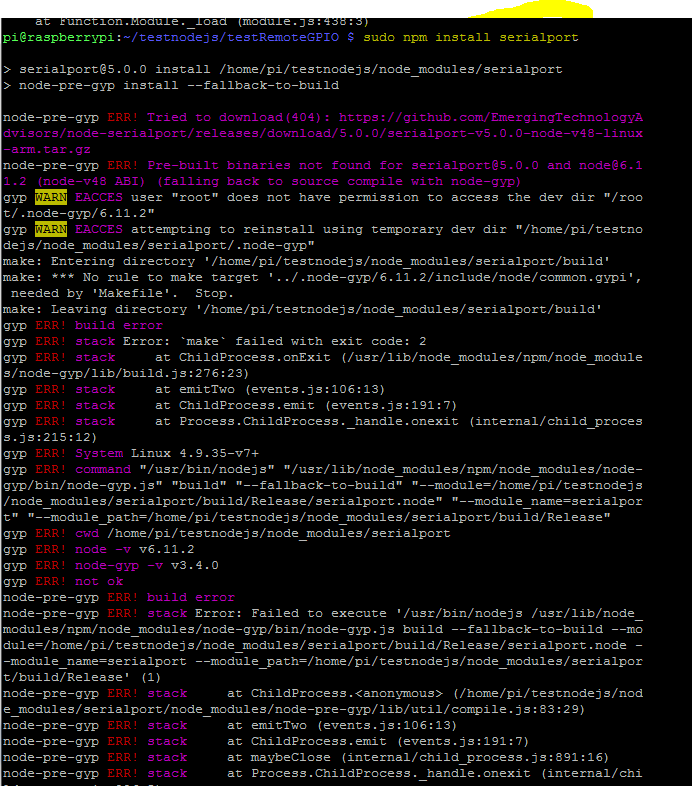
Inspiration de :

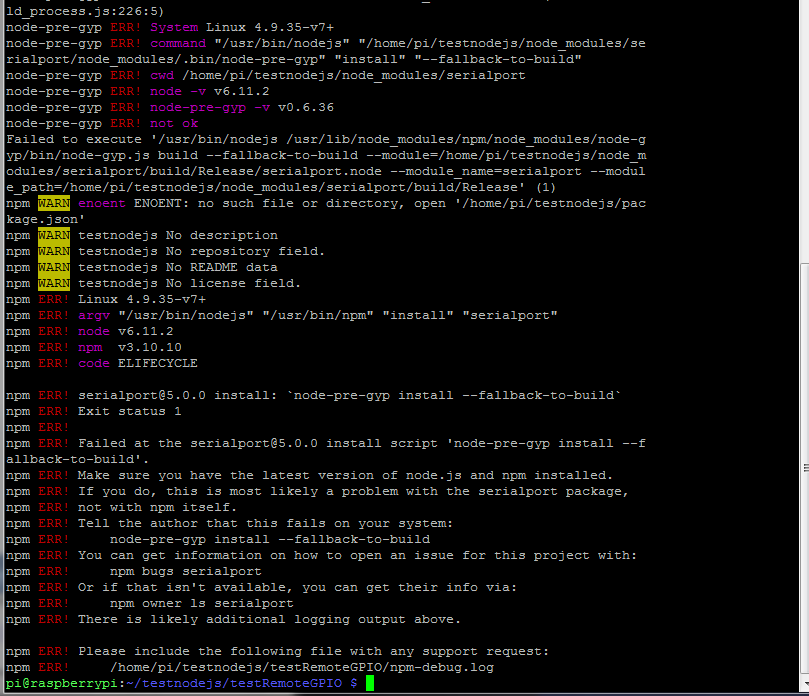
<https://alexandre.alapetite.fr/doc-alex/raspberrypi-nodejs-arduino/>

pour faire une com arduino - Raspberry

On doit installer le package serialport…

Et voilà que l’install plante..





07/08/2017 :

Test ventilo 12v

Faudra voir pour les addon c/C++ dans nodejs.. mais ca a l’air touchy.. donc il faut faire ça en pleine forme.

En attendant voir comment améliorer l’environnement de dev ou C..

Voir : <http://www.i-programmer.info/programming/hardware/9525-raspberry-pi-and-the-iot-in-c.html?start=1>

Pour l’install de NetBeans en version remote.

.. bon faut installer java ..etc..

A demain..

09/08/2017 :

On revient après un petit jour de OFF..

Après un détour sur les techno RAID0, RAID1 qui est très bien, utilisé par le NAS mais aussi utilisé sur les workstations HP Z600 etc.. et super outils Intel Storage management.. pour configurer son PC en RAID 1 a chaud.. !!! A exploiter dans les plus brefs délais..

reprise de l’install de NetBeans cette fois seulement la version C/C++ .. plus de message pour Java.

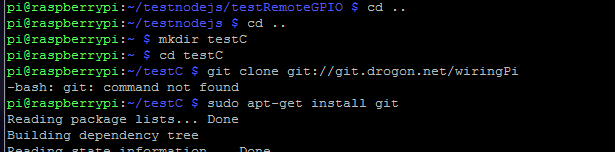
L’IDE se lance.. puis Pb avec NVidia cette fois.. incompatibilité de drivers..

Bon on revient sur le C.. a l’ancienne.

Pour installer la librairie wiringPi :

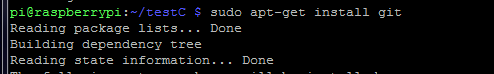
Bon on va laisser tomber notre environnement NetBeans et revenir a notre chaine de compil classique sous C.

Tentative d’install via git..



Pas de git..

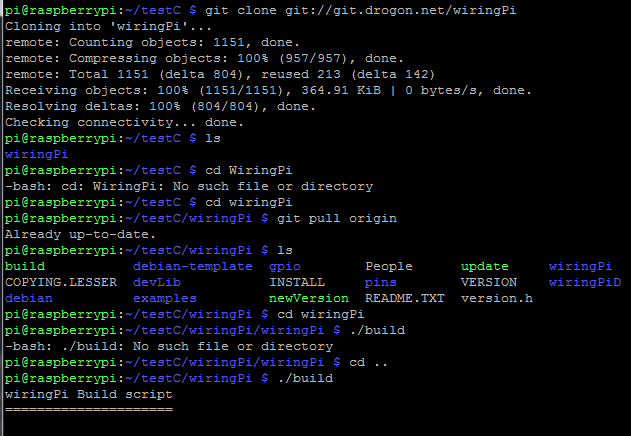
Donc install de git :



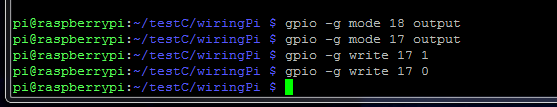
Puis reinstall de wiring pi via git :

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

Puis on build..



Et ça marche : on peut tester a la ligne de commande pour allumer notre port 17



Pour la suite de l’exploitation de cette lib via C

Voir : <https://learn.sparkfun.com/tutorials/raspberry-gpio>