

BelSD Card-Reader

Ce module sert à pouvoir lire et écrire sur des carte MIFARE via un interface Web.

Pour cela nous avons besoin de faire quelques installations et configurations sur notre Raspberry Pi

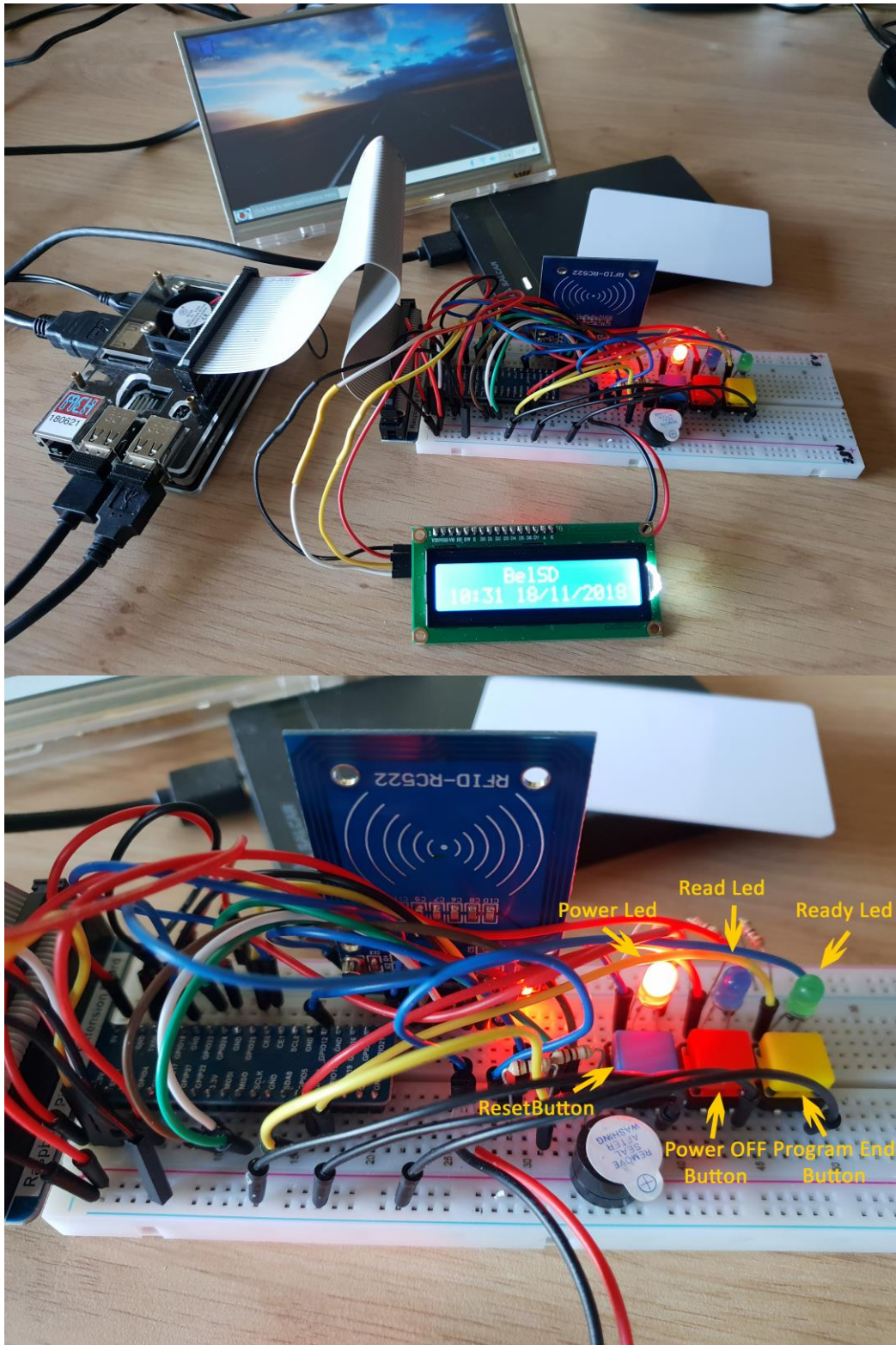


Table des Matières

Table des Matières	2
Montage du Module	3
Liste des composants.....	3
Montage.....	3
Préparation Raspberry	4
Activer I2C sur le Pi	4
Activer I2C dans Raspi-Config	4
Installer I2C-tools et smbus	5
Activer le module RFID RC522 sur le Pi.....	6
Information	7
Installation de la librairie MFRC522.....	7
Installation de Lighttpd avec PHP 7 (PHP-FPM) et MySQL 5.7	7
1 note préliminaire	7
2.Installer MySQL.....	7
3 Installation de Lighttpd	9
4 Installer PHP 7.0.....	9
5 Configuration de Lighttpd et PHP 7.0	9
6 Tester PHP 7.0 / Obtenir des détails sur votre installation de PHP	11
7 Obtenir le support de MySQL en PHP.....	12
8 phpMyAdmin	13
Installer MySQL pour Python	15
Redémarrez.....	15
Installer un serveur ftp sous Raspbian dans votre raspberry	15
Configuration proftpd	16
Test de connexion avec un client ftp comme filezilla	16
Changement du répertoire par défaut	17
Modifier les droits d'accès.....	17
Préparation de la base de données	17
Installation du programme BelSD Card-Reader.....	18
Configuration	18
Démarrage automatique	19
Pour la version Raspbian Stretch	19
Pour la version Raspbian LXDE (Desktop)	19
Conclusion.....	20

Montage du Module

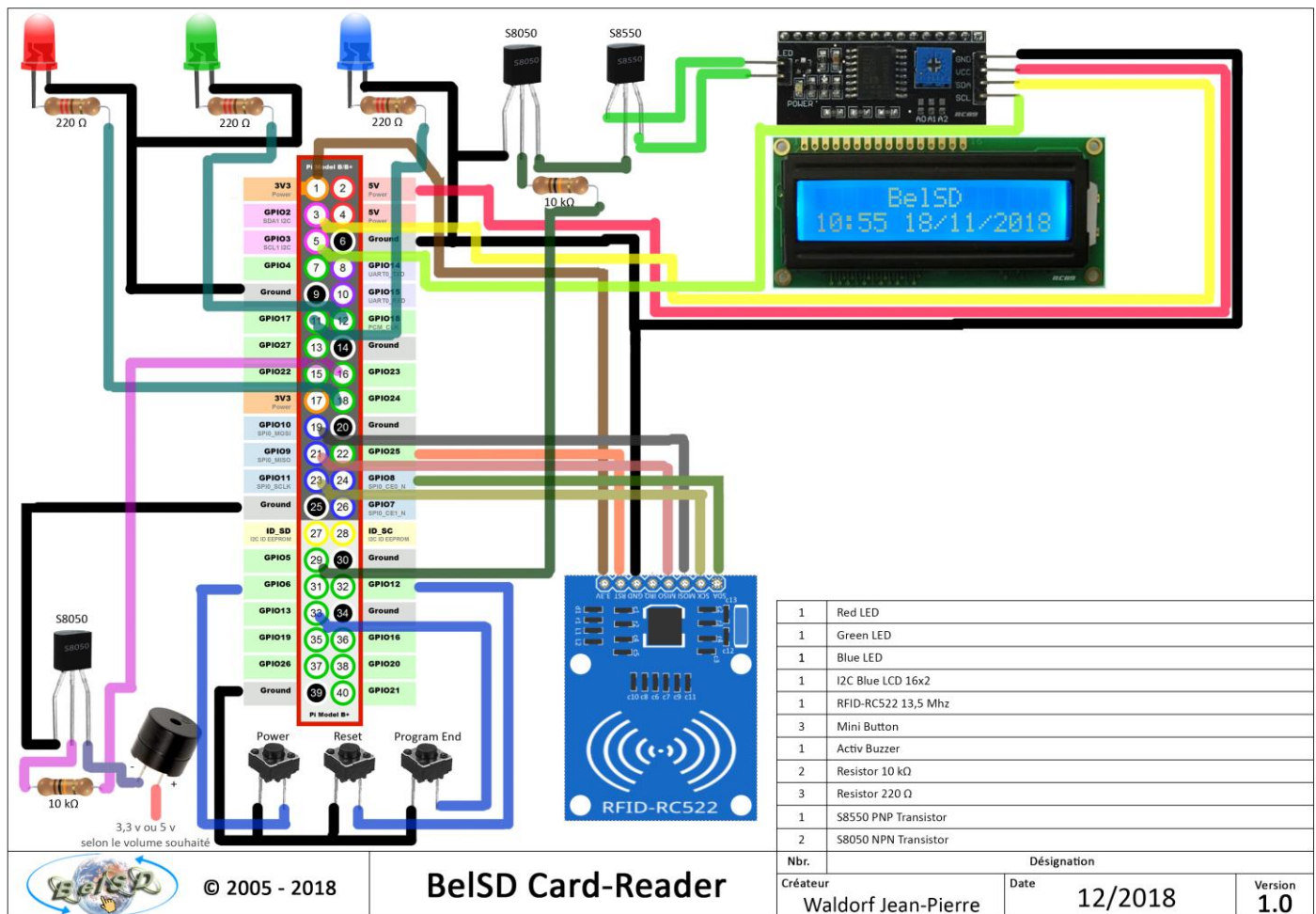
Liste des composants

- 1 x LED Rouge
- 1 x LED Vert
- 1 x LED Bleu
- 1 x I2C Blue LCD 16x2
- 1 x RFID-RC522 (13,5 MHz)
- 3 x Mini bouton poussoir
- 1 x Activ Buzzer
- 2 x Résistance 10 kΩ
- 3 x Résistance 220 Ω
- 1 x S8550 PNP Transistor
- 1 x S8050 NPN Transistor

Montage

Le plan est également téléchargeable à l'adresse suivante :

<https://github.com/BelSD/Card-Reader/blob/master/Documentations/BelSD%20Card-Reader.jpg>



Préparation Raspberry

En tout premier lieu on exécute les 2 commandes suivante !

```
sudo apt-get update  
sudo apt-get upgrade  
sudo reboot
```

Activer I2C sur le Pi

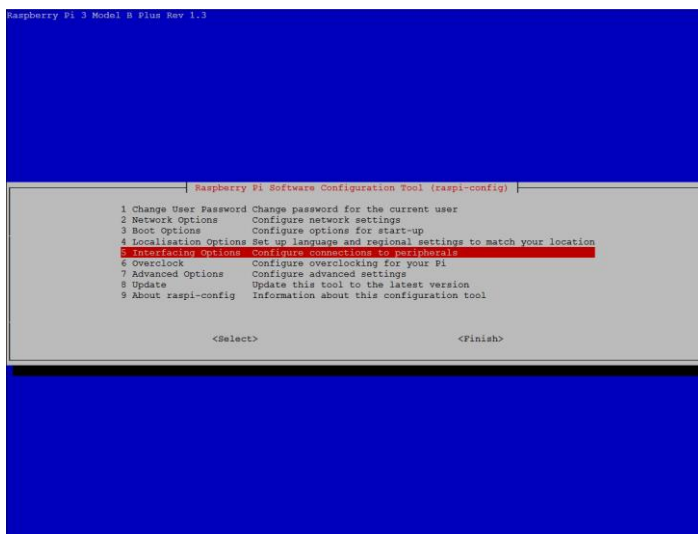
Avant de commencer la programmation, nous devons nous assurer que le module I2C est activé sur le Pi et installer quelques outils qui faciliteront l'utilisation de I2C.

Activer I2C dans Raspi-Config

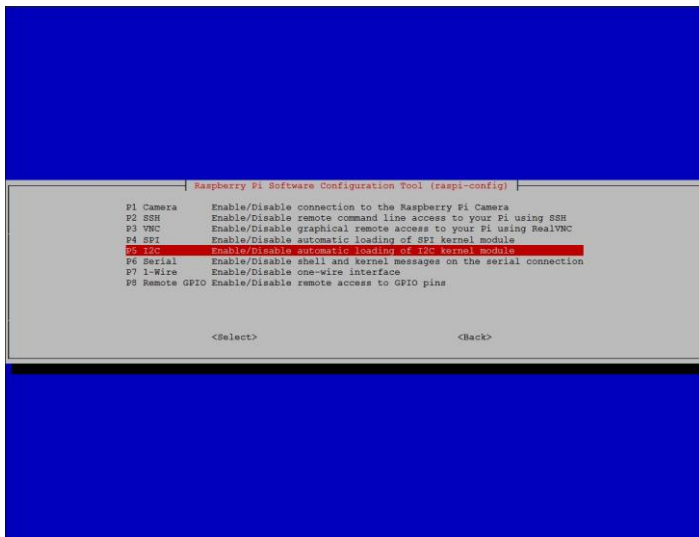
Avant de pouvoir utiliser notre module, il va falloir activer le I2C . Rien de bien compliqué, il suffit d'entrer dans la console :

```
sudo raspi-config
```

Puis flèche vers le bas et sélectionnez «Interfacing Options»:



Maintenant la flèche vers le bas et sélectionnez "I2C Enable/Disable automatic loading of I2C kernel module":



Choisissez «Oui» aux invites suivantes, quittez le menu de configuration et redémarrez pour activer les paramètres.

```
sudo reboot
```

Installer I2C-tools et smbus

Nous devons maintenant installer un programme appelé I2C-tools, qui nous indiquera l'adresse I2C de l'écran LCD lorsqu'il est connecté au Pi. Donc, à l'invite de commande, entrez

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

Ensuite, nous devons installer SMBUS, qui donne à la bibliothèque Python l'accès au bus I2C sur le Pi. À l'invite de commande, entrez

```
sudo apt-get install python-smbus
```

Maintenant, redémarrez le Pi et connectez-vous à nouveau. Avec votre écran LCD connecté, entrez

```
sudo i2cdetect -y 1
```

à l'invite de commande. Cela vous montrera un tableau d'adresses pour chaque appareil I2C connecté à votre Pi:

```
pi@192.168.69.250:~$ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  27  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@192.168.69.250:~$
```

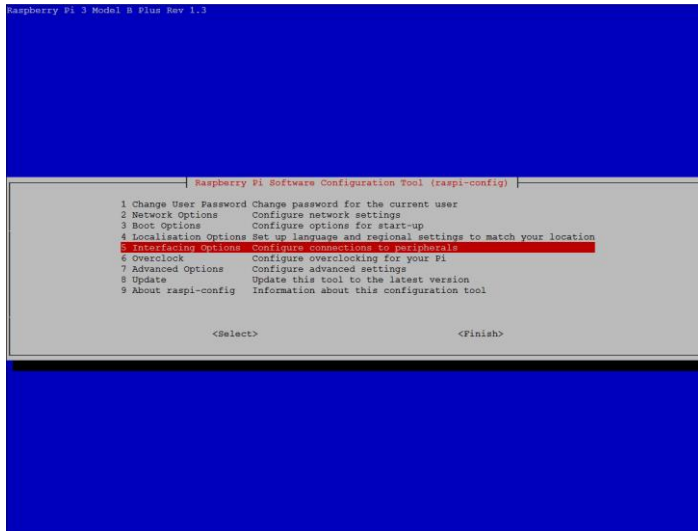

L'adresse I2C de l'écran LCD connecté à mon Pi est 27. Prenez note de l'adresse I2C de votre écran LCD, nous en aurons besoin plus tard.

Activer le module RFID RC522 sur le Pi

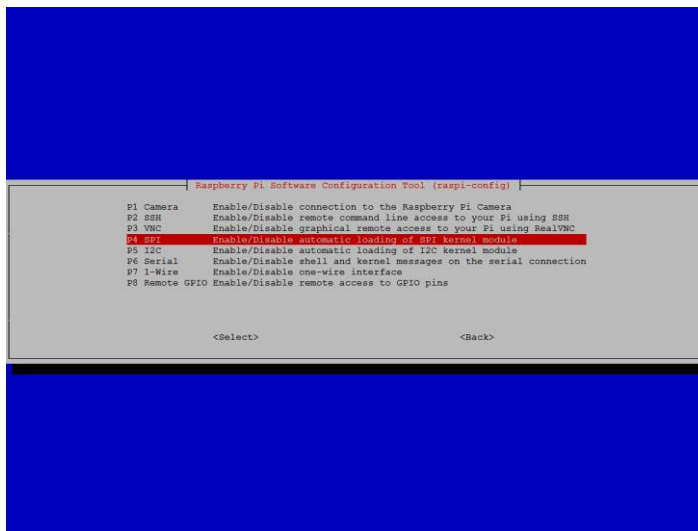
Avant de pouvoir utiliser notre module, il va falloir activer le SPI (*Serial Peripheral Interface*). Rien de bien compliqué, il suffit d'entrer dans la console :

```
sudo raspi-config
```

Puis flèche vers le bas et sélectionnez «Interfacing Options»:



Maintenant la flèche vers le bas et sélectionnez "SPI Enable/Disable automatic loading of SPI kernel module":



Choisissez «Oui» aux invites suivantes, quittez le menu de configuration et redémarrez pour activer les paramètres.

```
sudo reboot
```

Pour vérifier que le module SPI a bien été activé, entrer la commande suivante :

```
lsmod | grep spi
```

Si le module `spi_bcm2835` est listé, alors le module est bien activé.

Information

Module non listé ?

Il se peut que la ligne `spi_bcm2835` ne soit pas dans la liste. Pour remédier à ce problème, entrez la commande suivante :

```
sudo nano /boot/config.txt
```

Dans le fichier ouvert, cherchez la ligne `dtoverlay=spi=on`. Si cette ligne existe, alors enlevez le `#` qui débute la ligne. Sinon, si la ligne n'existe pas, ajoutez la à la fin du fichier. Enregistrez et quittez (Ctrl+X > "Y" > Enter). Redémarrer le Raspberry.

Installation de la librairie MFRC522

Nous pouvons maintenant installer SPY-Py :

```
cd ~
git clone https://github.com/lthiery/SPI-Py.git
cd ~/SPI-Py
sudo python setup.py install
```

Installation de Lighttpd avec PHP 7 (PHP-FPM) et MySQL 5.7

Lighttpd est un serveur Web sécurisé, rapide et conforme aux normes, conçu pour les environnements critiques en termes de vitesse. Ce didacticiel explique comment installer Lighttpd sur un serveur Ubuntu 16.04 prenant en charge PHP 7 (via PHP-FPM) et MySQL 5.7. PHP-FPM (FastCGI Process Manager) est une implémentation alternative de FastCGI avec quelques fonctionnalités supplémentaires utiles pour les sites de toutes tailles, en particulier les sites les plus occupés. J'utilise PHP-FPM dans ce tutoriel à la place de `spawn-fcgi` de Lighttpd.

1 note préliminaire

Dans ce tutoriel, j'utilise le nom d'hôte `server1.example.com` avec l'adresse IP `192.168.69.250`. Ces paramètres peuvent être différents pour vous, vous devez donc les remplacer le cas échéant. J'exécute toutes les étapes de ce tutoriel avec les privilèges root, alors assurez-vous d'être connecté en tant que root:

```
sudo -s
```

2.Installer MySQL

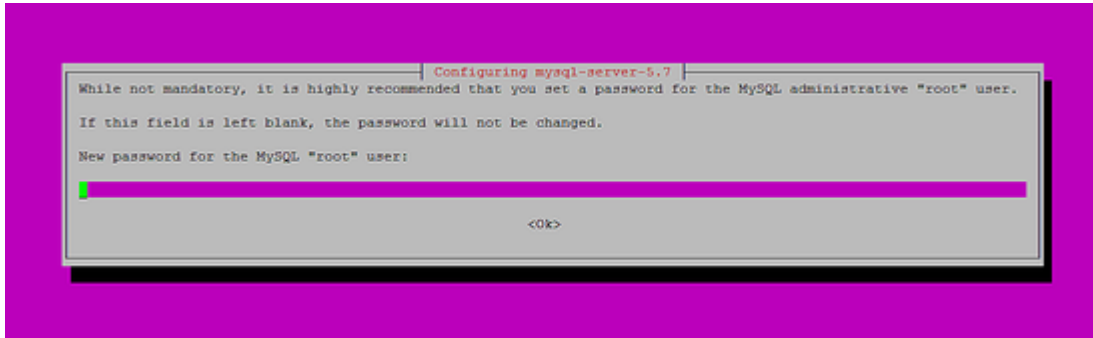
Tout d'abord, nous installons MySQL comme ceci:

```
apt-get -y install mysql-server mysql-client
```

Il vous sera demandé de fournir un mot de passe pour l'utilisateur root MySQL - ce mot de passe est valable pour l'utilisateur *root @ localhost* ainsi que pour *root@server1.example.com* . Il n'est donc pas nécessaire de spécifier ultérieurement un mot de passe root MySQL manuellement. :

Nouveau mot de passe pour l'utilisateur MySQL "root": <- **yourrootsqlpassword**

Répéter le mot de passe pour l'utilisateur MySQL "root": <- **yourrootsqlpassword**



Le programme d'installation a défini un mot de passe root MySQL, mais certains paramètres supplémentaires doivent être modifiés pour une installation sécurisée de MySQL. Cela peut être fait avec la commande `mysql_secure_installation`.

A présent nous pouvons vérifier que MySQL fonctionne correctement. Lançons maintenant MySQL:

```
mysql -u root -p votremotdepasse
```

NB: Si aucun mot de passe n'a été mis en place dans les manipulations précédentes, Lançons MySQL de la manière suivante:

```
mysql -u root -p
```

ou encore par:

```
mysql
```

Cette fenêtre devrait s'ouvrir. Grâce à cette invite de commande vous pouvez gérer toutes vos bases de données en ligne de commande.

```
root@raspberrypi:/home/pi# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Fenêtre montrant le bon fonctionnement de MySQL

Nous allons maintenant changer les privilèges de connexion afin de vous rendre super administrateur de MySQL.

```
use mysql;

update user set plugin='' where User='root';

flush privileges;

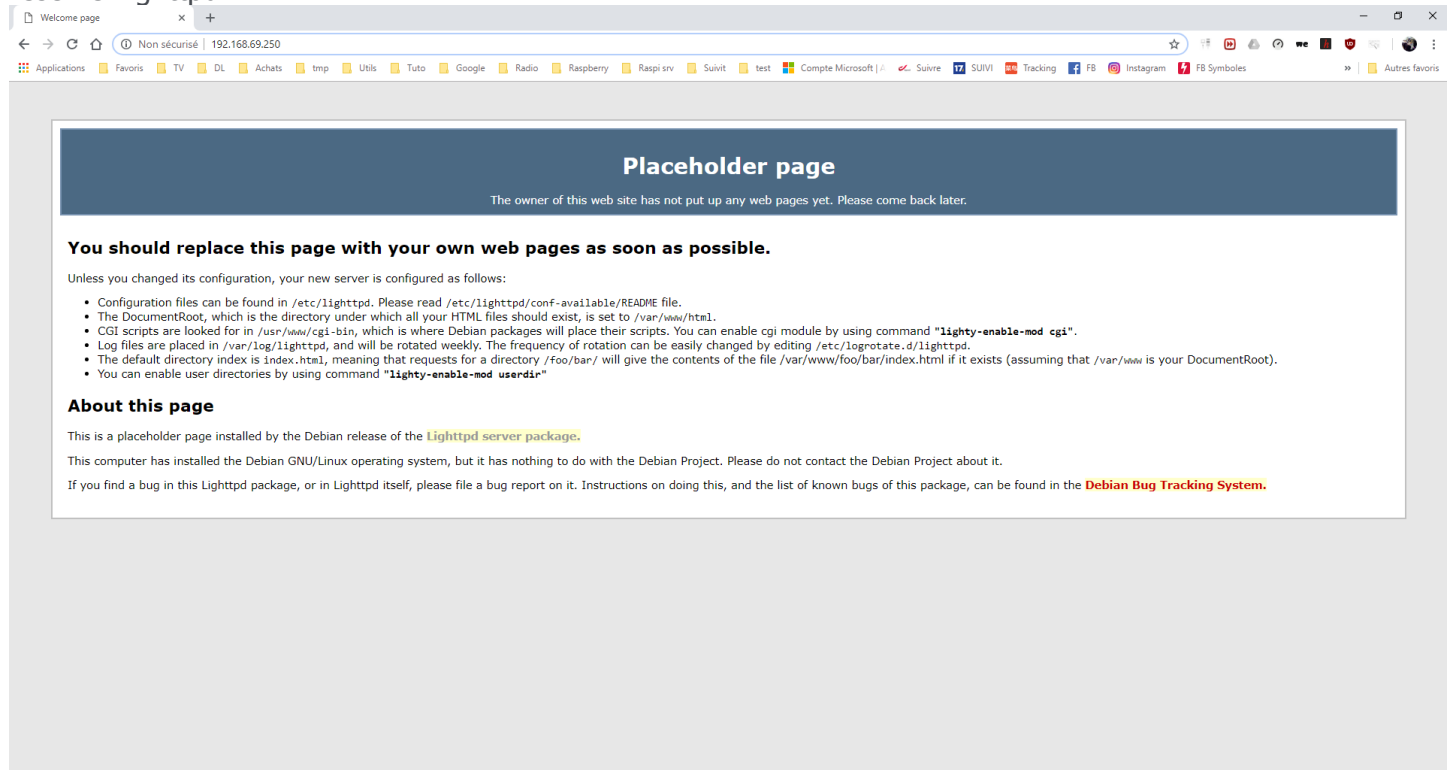
exit;
```


3 Installation de Lighttpd

Lighttpd est disponible sous forme de paquet Ubuntu. Par conséquent, nous pouvons l'installer directement avec apt à partir du référentiel de paquets Ubuntu Xenial Xerus:

```
apt-get -y install lighttpd
```

Dirigez maintenant votre navigateur vers `http://192.168.69.250/` et vous devriez voir la page d'espace réservé Lighttpd:



Le répertoire par défaut de Lighttpd est `/var/www/html` sur Ubuntu et le fichier de configuration est `/etc/lighttpd/lighttpd.conf`. Les configurations supplémentaires sont stockées dans des fichiers du répertoire `/etc/lighttpd/conf-available`. Ces configurations peuvent être activées à l'aide de la commande `lighttpd-enable-mod` qui crée un lien symbolique du répertoire `/etc/lighttpd/conf` à la configuration appropriée. fichier dans `/etc/lighttpd/conf-available`. Vous pouvez désactiver les configurations avec la commande `lighttpd-disable-mod`.

4 Installer PHP 7.0

Nous pouvons faire fonctionner PHP dans Lighttpd via PHP-FPM que nous installons comme ceci:

```
apt-get -y install php7.0-fpm php7.0
```

PHP-FPM est un processus démon (avec le script d'initialisation `php7-fpm`) qui exécute un serveur FastCGI sur le socket `/var/run/php/php7.0-fpm.sock`.

5 Configuration de Lighttpd et PHP 7.0

Pour activer PHP dans Lighttpd, nous devons modifier `/etc/php/7.0/fpm/php.ini` et changer la taille maximum des Upload de 2Mb à 10G à la ligne `post_max_size = 8M` et décommenter la ligne `cgi.fix_pathinfo = 1`:

```
nano /etc/php/7.0/fpm/php.ini
```

```
[...]
; Maximum size of POST data that PHP will accept.
; Its value may be 0 to disable the limit. It is ignored if POST data reading
; is disabled through enable_post_data_reading.
; http://php.net/post-max-size
post_max_size = 10G
[...]
; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting
; of zero causes PHP to behave as before. Default is 1. You should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
cgi.fix_pathinfo=1
[...]
```

Le fichier de configuration Lighttpd pour PHP `/etc/lighttpd/conf-available/15-fastcgi-php.conf` convient à une utilisation avec spawn-fcgi. Cependant, nous souhaitons utiliser PHP-FPM. Par conséquent, nous créons une sauvegarde du fichier (nommé `15-fastcgi-php.conf.bak`) et modifiez le fichier `15-fastcgi-php.conf` comme suit:

```
cd /etc/lighttpd/conf-available/
cp 15-fastcgi-php.conf 15-fastcgi-php.conf.bak
nano 15-fastcgi-php.conf
```

```
# /usr/share/doc/lighttpd-doc/fastcgi.txt.gz
# http://redmine.lighttpd.net/projects/lighttpd/wiki/Docs:ConfigurationOptions#mod_fastcgi-fastcgi

## Start an FastCGI server for php (needs the php7.0-cgi package)
fastcgi.server += ( ".php" =>
    (
        "socket" => "/var/run/php/php7.0-fpm.sock",
        "broken-scriptfilename" => "enable"
    )
)
```

Pour activer la configuration de fastcgi, exécutez les commandes suivantes:

```
lighttpd-enable-mod fastcgi
lighttpd-enable-mod fastcgi-php
```

Cela crée les liens symboliques `/etc/lighttpd/conf-enabled/10-fastcgi.conf` qui pointe vers `/etc/lighttpd/conf-available/10-fastcgi.conf` et `/etc/lighttpd/conf-enabled/15-fastcgi-php.conf` qui pointe vers `/etc/lighttpd/conf-available/15-fastcgi-php.conf` :

```
ls -l /etc/lighttpd/conf-enabled
```

```
racine @ serveur1 : / etc / lighttpd / conf-available # ls -l / etc / lighttpd / conf-enabled
total 0
lrwxrwxrwx 1 racine root 33 avril 27 11h26 10-fastcgi.conf -> ../conf-available/ 10-fastcgi.conf
lrwxrwxrwx 1 racine racine 37 27 avril 27 11:26 15-fastcgi-php.conf -> ../conf-available/15-fastcgi-php.conf
lrwxrwxrwx 1 racine racine 42 avril 21 11:10 90-javascript-alias.conf -> ../conf-available/90-javascript-alias.conf
root @ serveur1 : / etc / lighttpd / conf-available #
```

Ensuite, nous rechargeons Lighttpd:

```
service lighttpd force-reload
```

Remarque: Si vous obtenez des erreurs de paramètres régionaux, vous pouvez les supprimer en utilisant

```
apt-get -y installer language-pack-en-base
dpkg-reconfigure locales
```

6 Tester PHP 7.0 / Obtenir des détails sur votre installation de PHP

Le répertoire racine du site Web par défaut est `/var/www/html`. Nous allons maintenant créer un petit fichier PHP (`info.php`) dans ce répertoire et l'appeler dans un navigateur. Le fichier affichera de nombreuses informations utiles sur notre installation PHP, telles que la version PHP installée.

```
nano /var/www/html/info.php
```

```
<?php
phpinfo ();

?>
```


Nous appelons maintenant ce fichier dans un navigateur (par exemple, `http://192.168.69.250/info.php`) :

phpinfo()

Non sécurisé | 192.168.69.250/info.php


Applications Favoris TV DL Achats tmp Utils Tuto Google Radio Raspberry Raspbri Suivit test Compte Microsoft | Suivre SUVI Tracking FB Instagram FB Symboles Autres favoris

PHP Version 7.0.30-0+deb9u1



System	Linux raspberrypi 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l
Build Date	Jun 14 2018 13:50:25
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/fpm
Loaded Configuration File	/etc/php/7.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/fpm/conf.d
Additional .ini files parsed	/etc/php/7.0/fpm/conf.d/10-opcache.ini, /etc/php/7.0/fpm/conf.d/10-pdo.ini, /etc/php/7.0/fpm/conf.d/20-calendar.ini, /etc/php/7.0/fpm/conf.d/20-curl.ini, /etc/php/7.0/fpm/conf.d/20-ctype.ini, /etc/php/7.0/fpm/conf.d/20-exif.ini, /etc/php/7.0/fpm/conf.d/20-fileinfo.ini, /etc/php/7.0/fpm/conf.d/20-ftp.ini, /etc/php/7.0/fpm/conf.d/20-gdlib.ini, /etc/php/7.0/fpm/conf.d/20-iconv.ini, /etc/php/7.0/fpm/conf.d/20-json.ini, /etc/php/7.0/fpm/conf.d/20-ldap.ini, /etc/php/7.0/fpm/conf.d/20-openssl.ini, /etc/php/7.0/fpm/conf.d/20-readline.ini, /etc/php/7.0/fpm/conf.d/20-shmop.ini, /etc/php/7.0/fpm/conf.d/20-sockets.ini, /etc/php/7.0/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.0/fpm/conf.d/20-sysvsem.ini, /etc/php/7.0/fpm/conf.d/20-sysvshm.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API320151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
 with Zend OPcache v7.0.30-0+deb9u1, Copyright (c) 1999-2017, by Zend Technologies



Configuration

calendar

Calendar support	enabled
------------------	---------

Comme vous le voyez, PHP 7.0 fonctionne, et il fonctionne via FPM / FastCGI, comme indiqué dans la ligne *API* du *serveur*. Si vous faites défiler plus bas, vous verrez tous les modules qui sont déjà activés dans PHP5. MySQL n'est pas répertorié ici, ce qui signifie que nous n'avons pas encore de support MySQL en PHP.

7 Obtenir le support de MySQL en PHP

Pour obtenir le support de MySQL en PHP, nous pouvons installer le *paquet* `php7.0-mysql`. C'est une bonne idée d'installer d'autres modules PHP ainsi que vous pourriez en avoir besoin pour vos applications. Vous pouvez rechercher les modules PHP disponibles comme ceci :

```
apt-cache search php7.0
```

Choisissez ceux dont vous avez besoin et installez-les comme ceci :

```
apt-get -y install php7.0-mysql php7.0-curl php7.0-gd php7.0-intl php-pear php-imagick php7.0-imap php7.0-mcrypt php-memcache php7.0-pspell php7.0-recode php7.0-sqlite3 php7.0-tidy php7.0-xmlrpc php7.0-xsl php7.0-mbstring php-gettext
```

APCu est une extension du module PHP Opcache fourni avec PHP 7, elle ajoute des fonctionnalités de compatibilité pour les logiciels prenant en charge le cache APC (par exemple, les plugins de cache Wordpress). APCu peut être installé comme suit :

```
apt-get -y install php-apcu
```

Rechargez maintenant PHP-FPM :

```
service php7.0-fpm reload
```

Rechargez maintenant « `http://192.168.1.100/info.php` » dans votre navigateur et accédez à nouveau à la section modules. Vous devriez maintenant trouver beaucoup de nouveaux modules, y compris les modules `mysqli` et `mysqlnd`:

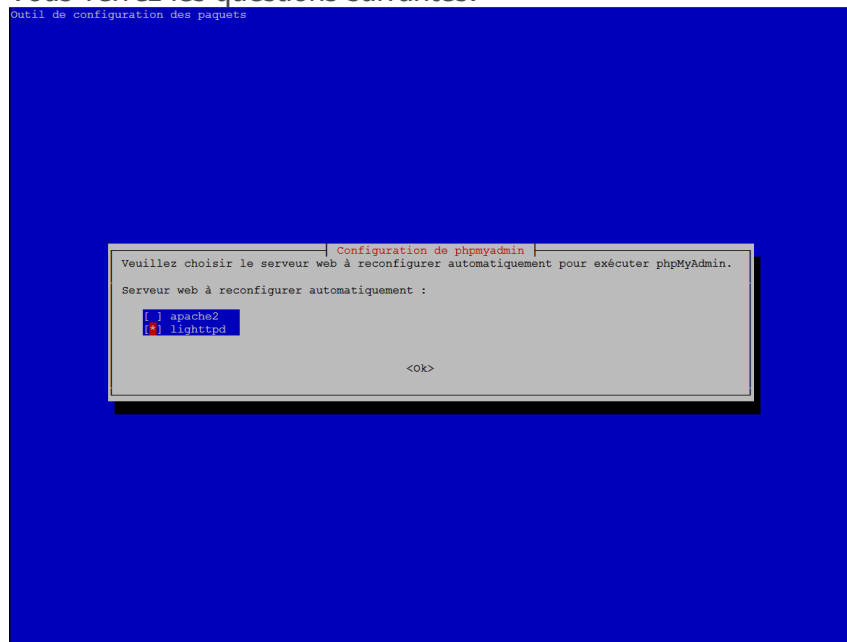
mysqli			
mysqli Support		enabled	
Client API library version	mysqli 5.0.12-dev - 20150407 - \$Id: b5c5906d452ec590732a93b051f3827e02749b83 \$		
Active Persistent Links	0		
Inactive Persistent Links	0		
Active Links	0		
Directive	Local Value	Master Value	
mysqli.allow_local_infile	On	On	
mysqli.allow_persistent	On	On	
mysqli.default_host	no value	no value	
mysqli.default_port	3306	3306	
mysqli.default_pw	no value	no value	
mysqli.default_socket	no value	no value	
mysqli.default_user	no value	no value	
mysqli.max_links	Unlimited	Unlimited	
mysqli.max_persistent	Unlimited	Unlimited	
mysqli.reconnect	Off	Off	
mysqli.rollback_on_cached_plink	Off	Off	
mysqlnd			
mysqlnd		enabled	
Version	mysqlnd 5.0.12-dev - 20150407 - \$Id: b5c5906d452ec590732a93b051f3827e02749b83 \$		
Compression	supported		
core SSL	supported		
extended SSL	supported		
Command buffer size	4096		
Read buffer size	32768		
Read timeout	31536000		
Collecting statistics	Yes		
Collecting memory statistics	No		
Tracing	n/a		
Loaded plugins	mysqlnd.debug_trace,auth_plugin_mysql_native_password,auth_plugin_mysql_clear_password,auth_plugin_sha256_password		
API Extensions	mysqli_pdo_mysql		
mysqlnd statistics			
bytes_sent	0		
bytes_received	0		

8 phpMyAdmin

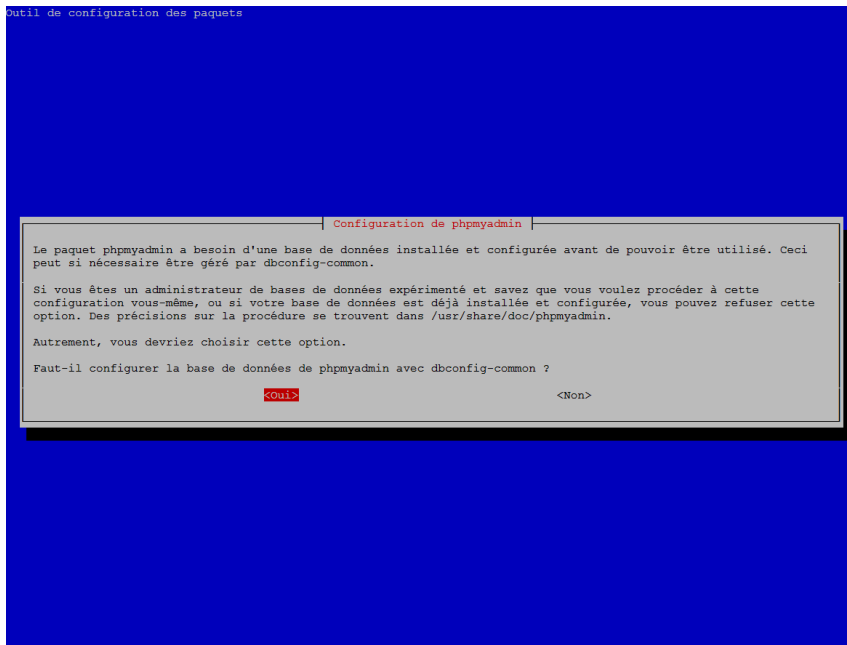
[phpMyAdmin](#) est une interface Web à travers laquelle vous pouvez gérer vos bases de données MySQL. C'est une bonne idée de l'installer:

```
apt-get -y install phpmyadmin
```

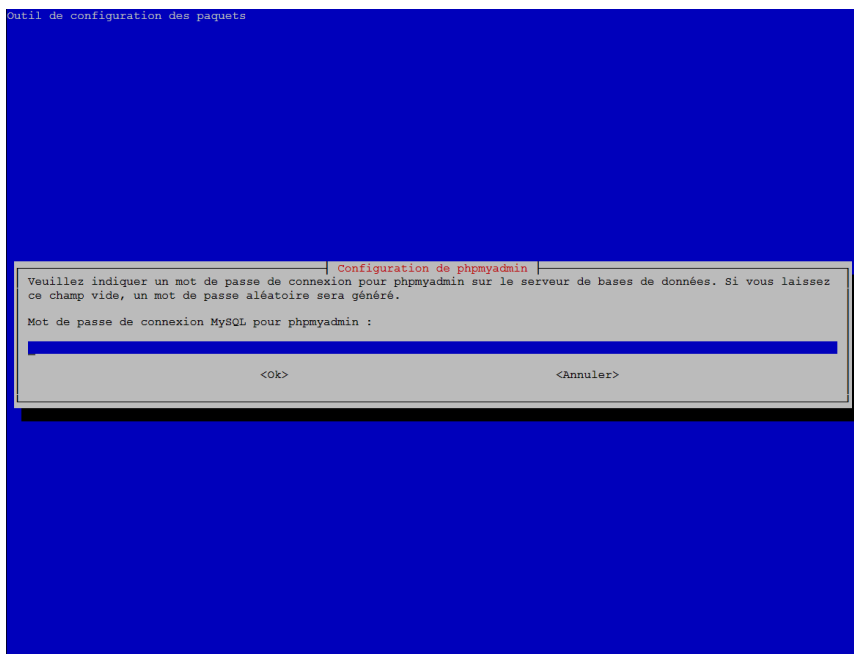
Vous verrez les questions suivantes:



Serveur web à reconfigurer automatiquement: **<-- lighttpd**



Faut-il configurer la base de données de phpadmin avec dbconfig-common? **<-- Oui**



Mot de passe de connexion MySQL pour phpmyadmin: **<-- Pousser Enter**

Si vous obtenez l'erreur suivante:

```
Run /etc/init.d/lighttpd force-reload to enable changes
dpkg: error processing package phpmyadmin (--configure):
subprocess installed post-installation script returned error exit status 2
E: Sub-process /usr/bin/dpkg returned an error code (1)
```

Puis lancez ces commandes:

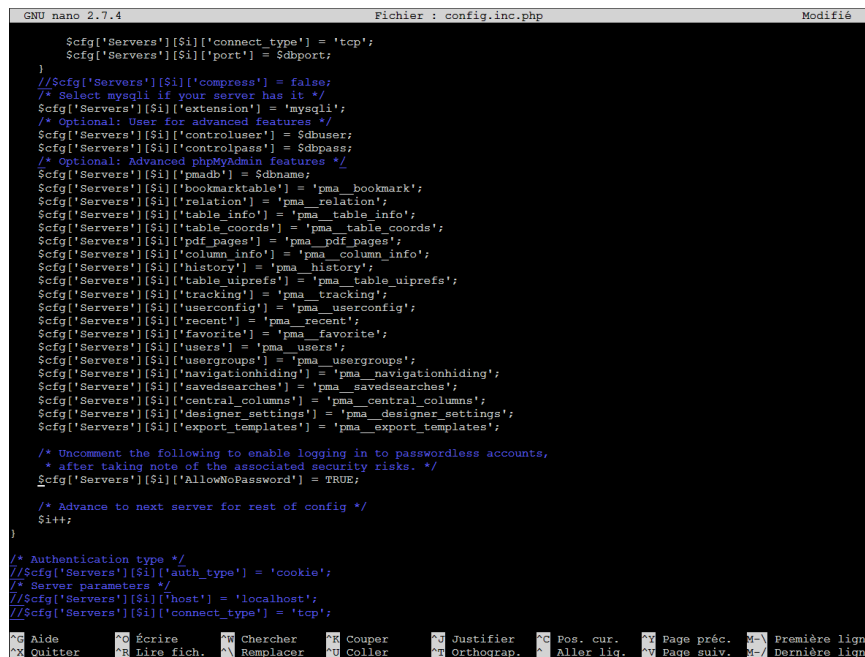
```
/etc/init.d/lighttpd force-reload
apt-get -y install phpmyadmin
```

Rendez-vous ensuite dans le répertoire phpmyadmin :

```
cd /etc/phpmyadmin
```


Il va falloir modifier le fichier « config.inc.php » de la façon suivante de telle sorte que « `$cfg['Servers'][$i]['AllowNoPassword'] = TRUE;` » soit décommentée. Cette ligne se trouve vers la fin du fichier.

`nano config.inc.php`



```
GNU nano 2.7.4 Fichier : config.inc.php Modifié
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['port'] = $dbport;
}
// $cfg['Servers'][$i]['compress'] = false;
/* Select mysqli if your server has it */
$cfg['Servers'][$i]['extension'] = 'mysqli';
/* Optional: User for advanced features */
$cfg['Servers'][$i]['controluser'] = $dbuser;
$cfg['Servers'][$i]['controlpass'] = $dbpass;
/* Optional: Advanced phpMyAdmin features */
$cfg['Servers'][$i]['pmadb'] = $dbname;
$cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
$cfg['Servers'][$i]['relation'] = 'pma_relation';
$cfg['Servers'][$i]['table_info'] = 'pma_table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma_column_info';
$cfg['Servers'][$i]['history'] = 'pma_history';
$cfg['Servers'][$i]['table_uiprefs'] = 'pma_table_uiprefs';
$cfg['Servers'][$i]['tracking'] = 'pma_tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma_userconfig';
$cfg['Servers'][$i]['recent'] = 'pma_recent';
$cfg['Servers'][$i]['favorite'] = 'pma_favorite';
$cfg['Servers'][$i]['users'] = 'pma_users';
$cfg['Servers'][$i]['usergroups'] = 'pma_usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma_navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma_savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma_central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma_designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma_export_templates';

/* Uncomment the following to enable logging in to passwordless accounts,
 * after taking note of the associated security risks. */
$cfg['Servers'][$i]['AllowNoPassword'] = TRUE;

/* Advance to next server for rest of config */
$i++;
}

/* Authentication type */
// $cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
// $cfg['Servers'][$i]['host'] = 'localhost';
// $cfg['Servers'][$i]['connect_type'] = 'tcp';

⌘ Aide ⌘ Écrire ⌘ Chercher ⌘ Couper ⌘ Justifier ⌘ Pos. cur. ⌘ Page préc. ⌘ Première ligne
⌘ Quitter ⌘ Lire fich. ⌘ Remplacer ⌘ Coller ⌘ Orthograp. ⌘ Aller lig. ⌘ Page suiv. ⌘ Dernière ligne
```

Installer MySQL pour Python

`apt-get install python-mysqldb`

Redémarrez

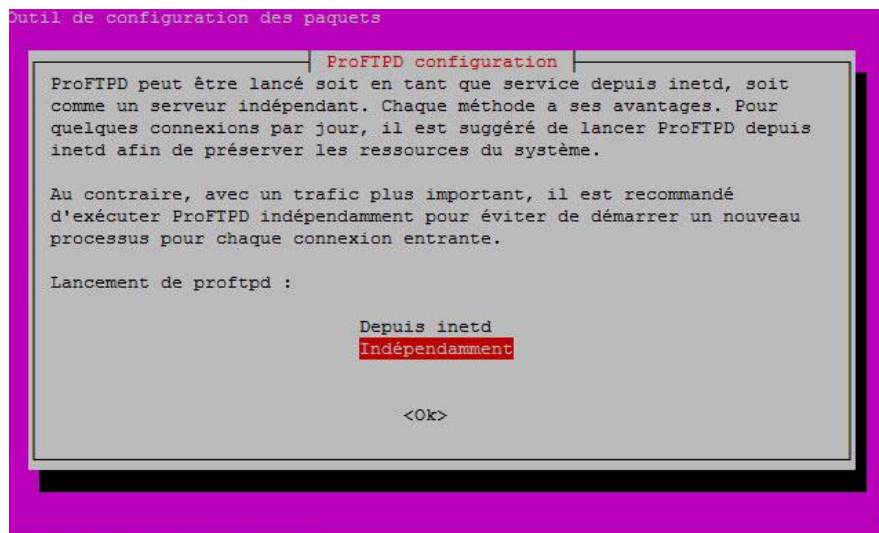
`reboot`

Installer un serveur ftp sous Raspbian dans votre raspberry

Installer proftpd, avec la commande suivante :

`sudo aptitude install proftpd`

Message d'avertissement sur le choix sur comment va être lancé proftpd



Pour un serveur avec quelques connexion, choisir « *inetd* » ou au contraire beaucoup de connexions alors choisir « *Indépendamment* »

Configuration proftpd

Modifier le fichier « *proftpd.conf* » situé dans le répertoire « */etc/proftpd/* »

Édition du fichier à modifier en mode console avec la commande suivante :

```
sudo nano /etc/proftpd/proftpd.conf
```

Fichier proftpd

```
#activation ou pas de l'IPv6
UseIPv6                                off

#le port d'écoute
Port                                    21
```

Redémarrer le serveur ftp, avec la commande suivante :

```
sudo service proftpd restart
```

Tester si le service à bien été démarré, avec la commande suivante :

```
sudo ps aux | grep proftpd
```

```
pi@192.168.69.250:~$ sudo ps aux | grep proftpd
proftpd  633  0.0  0.3 14568 3436 ?        Ss   07:26   0:00 proftpd: (accepting connections)
pi      1614  0.0  0.4  6096 3956 pts/0    Ss   09:15   0:00 bash -i -c sudo ps aux | grep proftpd
pi      1626  0.0  0.0  4372  536 pts/0    S+   09:15   0:00 grep --color=auto proftpd
pi@192.168.69.250:~$
```

Test de connexion avec un client ftp comme filezilla

Avec l'adresse IP de votre Raspberry

L'identifiant de votre Raspberry « *pi* » si vous ne l'avez pas changé
Mot de passe de votre Raspberry « *raspberrypi* » si vous ne l'avez pas changé
Port numéro « *21* » par défaut si vous ne l'avez pas changé

Vous devriez arriver sur le répertoire « */home/pi/* » par défaut

Changement du répertoire par défaut

Si vous voulez arriver sur le répertoire « */var/www/* », ajouter la ligne suivante dans le fichier

« *DefaultRoot* */var/www* » comme ci-dessus.

Fichier *proftpd*

```
sudo nano /etc/proftpd/proftpd.conf

#répertoire de destination par défaut
DefaultRoot                      /var/www
```

Redémarrer le serveur ftp, avec la commande suivante :

```
sudo service proftpd restart
```

Modifier les droits d'accès

Pour pouvoir faire des modifications, uploads, etc... Vous devez IMPÉRATIVEMENT donner les droits d'accès au répertoire « */var/www* » et « */var/www/html* »

Pour cela exécutez les commandes suivantes :

```
sudo chown pi /var/www
sudo chown pi /var/www/html
```

Cela donnera un accès total à l'utilisateur « *pi* » sur ces deux répertoires.

Préparation de la base de données

Sous phpMyAdmin, créez un nouveau compte utilisateur avec « *créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base* ».

Connectez-vous en tant que « *root* » sur phpMyAdmin
Allez dans l'onglet « *comptes utilisateurs* »

Indiquez le nom d'utilisateur

Nom d'hôte -> *localhost*

Le mot de passe

Ne PAS oublier de cocher la case devant « *créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base* ».

Déconnectez-vous et reconnecter vous avec votre nouveau nom d'utilisateur.
Sélectionner la base de données comportant votre nom d'utilisateur puis allez dans l'onglet « SQL » et copier les lignes suivantes :

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
CREATE TABLE `card_reader` (
  `id` int(11) NOT NULL,
  `valeur` varchar(50) DEFAULT NULL,
  `info` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `card_reader` (`id`, `valeur`, `info`) VALUES
(1, 'BelSD', 'Instruction au Card Reader'),
(2, '', 'UID de la carte'),
(3, '', 'Date création'),
(4, '', 'Nom'),
(5, '', 'Prénom'),
(6, '', 'Nr client'),
(7, '', 'Points'),
(8, '', 'Date dernière visite');
ALTER TABLE `card_reader`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `card_reader`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
```

Installation du programme BelSD Card-Reader

En mode SSH, tapez les lignes suivantes :

```
cd /var/www
sudo git clone https://github.com/BelSD/Card-Reader.git
cd Card-Reader
```

Configuration

Allez dans le répertoire « `\var\www\Card-Reader` » et éditez le fichier « `config.py` »

```
sudo nano config.py
```

Changez les données « *Db_User_Login* », « *Db_Password* » et « *Db_Data_base* » avec vos données introduites à la configuration de la préparation de la base de données

Changez également les données dans la « *class LCD* »

ADDRESS -> changez le numéro avec celui obtenu dans la partie « Installation I2C »

Démarrage automatique

Pour que le module « *BelSD Card-Reader* » démarre directement après le lancement de notre Raspberry Pi.

Pour la version Raspbian Stretch

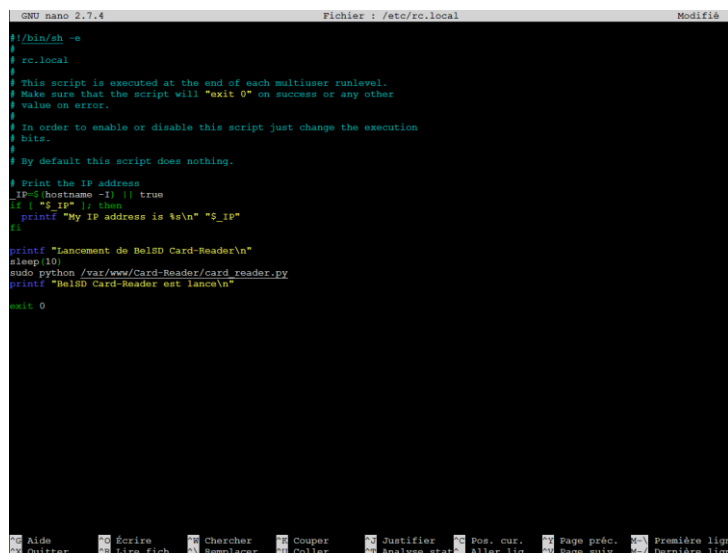
Il faut éditer le fichier « *rc.local* ».

Pour cela tapez la ligne suivante :

```
sudo nano /etc/rc.local
```

Il faut maintenant y rajouter les lignes suivantes avant la ligne « *exit 0* » :

```
printf "Lancement de BelSD Card-Reader\n"
sleep(10)
sudo python /var/www/Card-Reader/card_reader.py
printf "BelSD Card-Reader est lancé\n"
```



```
GNU nano 2.7.4 Fichier : /etc/rc.local Modifié
#!/bin/sh -s
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

printf "Lancement de BelSD Card-Reader\n"
sleep(10)
sudo python /var/www/Card-Reader/card_reader.py
printf "BelSD Card-Reader est lancé\n"

exit 0
```

Pour la version Raspbian LXDE (Desktop)

Il faut éditer voir créer le fichier « *python.desktop* »

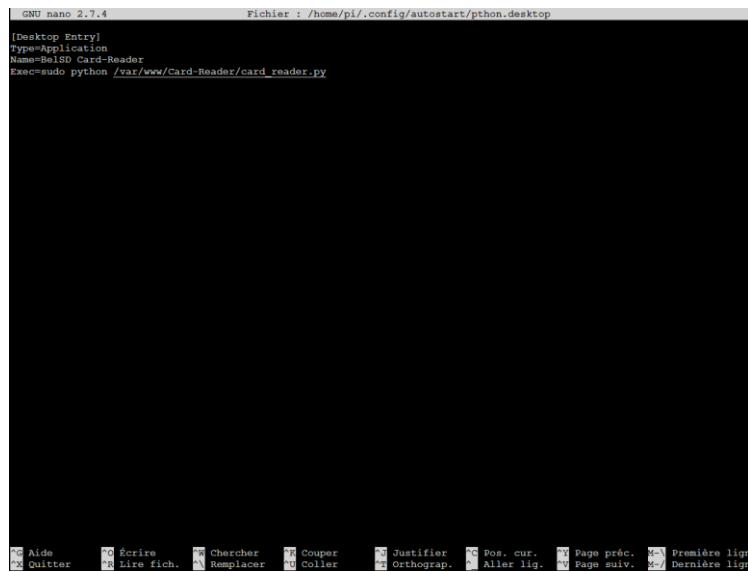
Pour cela tapez la ligne suivante :

```
sudo nano ~/.config/autostart/python.desktop
[Desktop Entry]
```

```
Type=Application
```

```
Name=BelSD Card-Reader
```

```
Exec=sudo python /var/www/Card-Reader/card_reader.py
```



Conclusion

Voilà votre BelSD Card-Reader est prêt à fonctionner avec des instructions via une page php.

Dans le répertoire « `/var/www/Card-Reader/Exemples` » ce trouve un simple exemple comment cela fonctionne avec php et MySQL