

Configuration Raspberry

jonathan baillet

GMSI 2021

11/03/2023

Table des matières

1	Préambule	1
1.1	Prérequis	1
1.2	Utilisation de Putty	1
2	Installation pas-à-pas	2
2.1	Installation	2
2.2	Configuration	2
2.3	Utilisation	3

1 Préambule

Dans le cadre de l'atelier robotique et de la conception du labo sécurité, il est nécessaire de mettre en œuvre un Raspberry destiné à communiquer avec un ESP32.

L'objet de cette procédure est de permettre de configurer le Raspberry après une réinstallation.

1.1 Prérequis

Afin de tirer pleinement parti du présent guide, certains prérequis doivent être établis :

- Décompresser les **fichiers joints**
- Avoir installé **Putty**

1.2 Utilisation de Putty

Putty permet de se connecter au Raspberry via un terminal, il permet notamment de coller le contenu du presse papier grâce au clic droit de la souris.

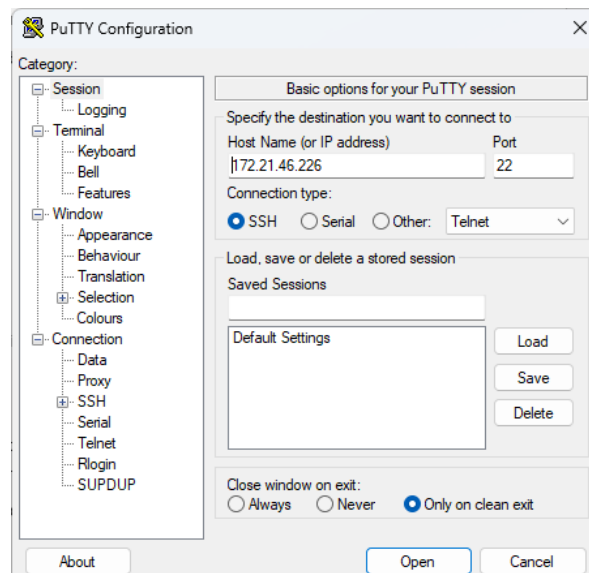
Il est d'abord nécessaire d'autoriser le **SSH** dans le Raspberry :

- **Menu > Préférences > Configuration du Raspberry pi**
- Dans l'onglet interfaces, activer le **SSH**

Il faut ensuite récupérer l'adresse IP du Raspberry pi via le terminal :

- **Ip a**

La rentrer dans putty :



Et s'identifier (Identifiants par défaut : pi / raspberry), la suite de la configuration se déroulera sous putty.

2 Installation pas-à-pas

Assurez-vous d'avoir lu la partie prérequis avant de poursuivre.

Si vous utilisez une carte **SD de 8 go** :

- Désinstaller Libreoffice : **sudo apt purge -y libreoffice**
- Désinstaller Minecraft : **sudo apt purge -y minecraft-pi**
- Ne faites **pas** de mise à jour système

Si vous utilisez une carte **SD de 16 Go** :

- Mettre le système à jour : **sudo apt update && sudo apt upgrade -y**

2.1 Installation

Via Putty :

- **nano install.sh** : y coller le contenu du fichier install.sh fourni
- **dos2unix install.sh**
- **bash install.sh**

Durant l'exécution du script, certaines questions vous seront posées :

- Sélectionner **Apache2** puis **NON**
- Sélectionner **Internet avec un "smarthost"**
- Définir le **nom de domaine** (attention à ne pas utiliser un nom de domaine existant)
- Supprimer le **serveur relais SMTP**

Puis le Raspberry va redémarrer.

2.2 Configuration

Définir le mot de passe root :

- **sudo passwd root**

Sécuriser la base de données :

- **sudo mysql_secure_installation**
- Répondre **Y** à toutes les questions **sauf** au changement de mot de passe root

Créer un utilisateur pour la base de données :

- **sudo useradd** utilisateur
- **sudo passwd** utilisateur

Créer la base de données, les valeurs qui ne sont pas en gras doivent être modifiées, les droits **insert** et **select** suffisent à manipuler la BDD (un résumé des liens ici :

<https://www.liquidweb.com/kb/grant-permissions-to-a-mysql-user-on-linux-via-command-line/>) :

- **sudo mysql -u root**
- **create database** base_de_donnée;
- **grant** droits **on** base_de_donnée.* **to** utilisateur@localhost **identified by** "mot_de_passe";
- **flush privileges;**
- **exit**

2.3 Utilisation

Raspberry Pi : La carte wifi du Raspberry sert de point d'accès sur lequel l'ESP32 peut se connecter. L'adresse IP du Raspberry est configurée en **192.168.0.10/8** et le **DHCP** distribue des adresse sur l'étendue **192.168.0.11/8** à **192.168.0.30/8** et un nouveau bail sera délivré toutes les **24h**. Le Raspberry n'a plus d'accès Internet, il est toutefois possible d'y connecter un téléphone en USB pour partager la connexion.

SSID : RASPBERRY

Mot de passe : Cesi29200

PHPmyadmin : Pour se rendre sur l'API graphique de PHPmyadmin, renseigner dans la barre du navigateur du Raspberry « **localhost/phpmyadmin** » et se connecter avec les identifiants de l'utilisateur de la BDD.

Scripts PHP : Passer **root (su – root)** et copier les scripts PHP vers : « **/var/www/html/** ».

Ces scripts doivent être nommés avec l'extension « **.php** » et sont accessibles via le navigateur à l'adresse « **localhost/nom_du_script.php** »

Pour envoyer un mail via terminal :

echo "corps_du_mail" | mail -s "sujet_du_mail" destinataire@localhost

Le destinataire doit être un utilisateur créé sur le système (voir plus haut la création d'utilisateur).

camera.sh : Ce script est destiné à activer la camera 3 fois à 1 seconde d'intervalle à chaque passage de badge et de sauvegarder les photos prises.

Pour le lancer, envoyer la commande « **bash camera.sh** » dans le terminal à l'emplacement où est stocké le script.