

Démarrage PST Lego

N. Biboulet

24 février 2017

Table des matières

1 Installation	1
1.1 VM Ubuntu	1
1.2 Brique Lego EV3	3
1.3 Raspberry	4
1.4 Android	4
2 Outils	5
2.1 Shell de base	5
2.2 Outils réseau de base	5
2.3 Vim	5
2.4 Python	6
3 Démarrage projet	6
3.1 EV3 - ev3dev	6
3.2 Android - Qpython	6
3.3 RPI - Xloborg	6
3.4 Python - Pygame	6
3.5 Python sockets	7

Résumé

Ce document décrit tout d'abord la finalisation des installations des différents éléments pour tous les projets. Ensuite, quelques liens et références sont donnés pour des outils utiles. Enfin, le document se termine par des informations aidant au démarrage (pratique) des projets.

1 Installation

Le but de cette section est de finaliser les installations sur les différents appareils et de récapituler les informations de connexion.

1.1 VM Ubuntu

L'installation est basée sur une **Ubuntu 16.04 LTS** dans une machine virtuelle **VirtualBox 5.1.14**. L'installation par défaut est suffisante à part trois points :

- Le partage d'un répertoire commun avec votre système d'exploitation principal est pratique (un onglet est dédié et sa mise en place est directe).
- Les options de partage de connexion internet sont à définir avec soin, il faut : choisir une connexion **par point**, une 'promiscuité' totale, et partager la connexion **wifi** pour la communication entre ev3-téléphone-RPI. Inversement, la connexion **eth** sera partagée pour l'installation de paquets. Cela apparaît dans la figure 1.
- Si vous souhaitez partager le presse-papier avec votre système d'exploitation principal, l'installation dans votre **VM** du paquet **virtualbox-guest-dkms** est nécessaire ainsi que l'activation dans l'onglet Général/Avancé du partage bidirectionnel.

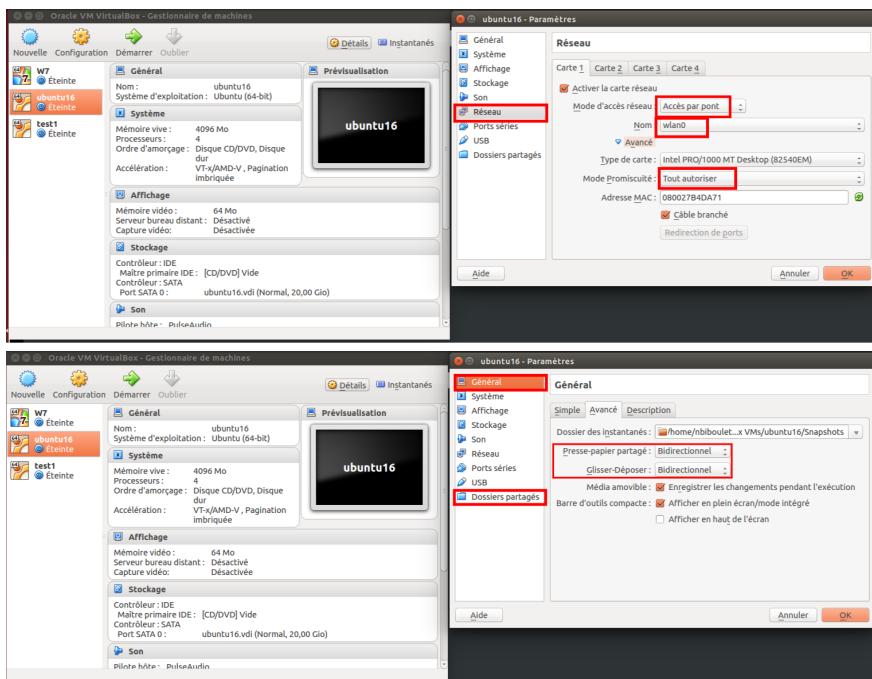


FIGURE 1 – Options VM

```
sudo apt-get install virtualbox-guest-dkms
reboot
sudo adduser LOGIN vboxsf
reboot
```

Le minimum des paquets à installer dans votre VM est :

```
sudo apt-get install python3-numpy python3-scipy
sudo apt-get install python3-matplotlib vim
sudo apt-get install openssh-server
```

Ils permettent d'avoir une gestion des tableaux matrices efficaces ressemblant à Matlab. De plus, vous aurez une base de fonctions scientifiques, des outils graphique 'Matlab-like' et un éditeur de texte dans le terminal, les services ssh.

Pour certains projets nécessitant la librairie **pygame**, la procédure légèrement adaptée de <http://pygame.org/wiki/CompileUbuntu> est :

```
sudo apt-get install mercurial python3-dev python3-setuptools
python3-numpy python3-opengl libav-tools
libsdl-image1.2-dev libsdl-mixer1.2-dev libsdl-ttf2.0-dev
libsmpeg-dev libsdl1.2-dev libportmidi-dev libswscale-dev
libavformat-dev libavcodec-dev libtiff5-dev libx11-6
libx11-dev fluid-soundfont-gm musescore-soundfont-gm
xfonts-base xfonts-100dpi xfonts-75dpi xfonts-cyrillic
fontconfig fonts-freefont-ttf

sudo apt-get install libfreetype6-dev

hg clone https://bitbucket.org/pygame/pygame
cd pygame
python3 setup.py build
sudo python3 setup.py install
```

1.2 Brique Lego EV3

L'installation est basée sur une **Debian - jessie** <http://www.ev3dev.org/docs/getting-started/>. Son adresse IP est indiquée dans le coin supérieur gauche : 192.168.42.IPEV3 comme visible dans la figure 2.

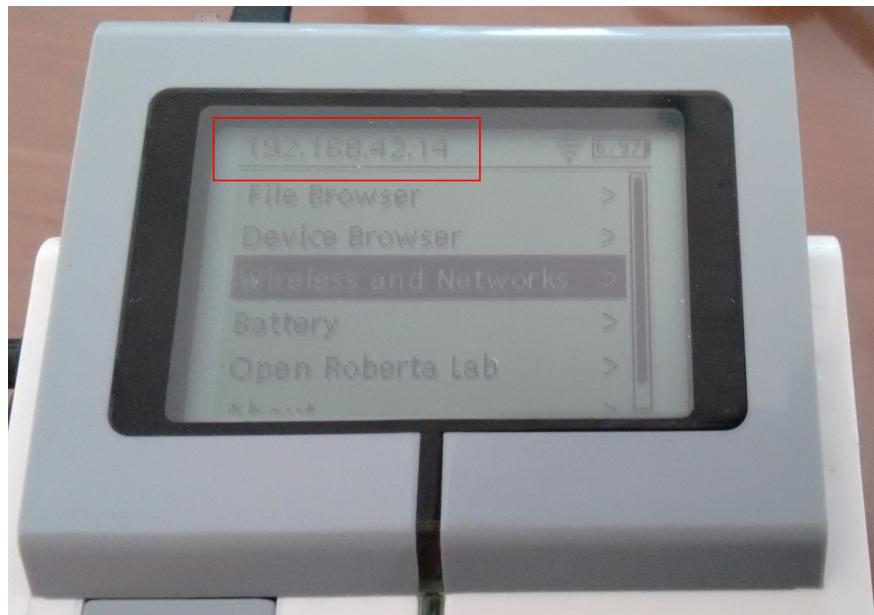


FIGURE 2 – IP ev3

La connexion **ssh** à partir de votre ordinateur se fera grâce à (**robot** est le login et **maker** est le mot de passe) :

```
ssh LOGIN@192.168.42.IPEV3
```

1.3 Raspberry

Le RPI utilise une distribution **Raspian - jessie** <https://www.raspberrypi.org/downloads/raspbian/>. Il est configuré en serveur dhcp (il va distribuer les adresses IP de tous les ‘ordinateurs’ qui s’y connectent). Il émet un wifi nommé **ev3devwN**. Son adresse IP est fixe : 192.168.42.1 le login est **pi**, le mot de passe est **raspberry** (c’est le réglage par défaut, donc il ne faut pas laisser connectés vos appareils en continu chez vous car le réseau est tout sauf sécurisé).

La carte de capteurs **Xloborg** rajoutée au RPI est décrite sur <https://www.piborg.org/xloborg/install>. L’onglet Getting Started décrit la procédure d’installation :

```
mkdir ~/xloborg
cd ~/xloborg
wget http://www.piborg.org/downloads/xloborg/examples.zip
unzip examples.zip
chmod +x install.sh
./install.sh
sudo reboot
```

1.4 Android

Le but de cette section est de permettre la connexion à distance sur votre téléphone. La procédure est en grande partie empruntée de <https://oliverse.ch/tech/2015/11/06/run-an-ssh-server-on-your-android-with-termux.html>. L’installation est basée sur une version **Android 6.0**.

L’installation d’applications et de paquets requiert une connexion internet. Mais, dès qu’une communication entre téléphone et ordinateur doit être établie, elle se fera par l’intermédiaire du wifi local émis par le RPI.

Les applications à installer sont : **Termux**, **Hacker’s Keyboard**, **Qpython3**. Pour avoir respectivement :

- une émulation de terminal Linux,
- un clavier complet,
- un interpréteur **python3** incluant la librairie **SL4A** qui permet d’avoir accès aux capteurs.

```
sudo apt-get update
sudo apt-get install ...
```

Dans **Termux**, il faut installer **netcat**, **vim**, **man**, **openssh** qui permettent de :

- transmettre du texte et de tester son réseau
- d’éditer du texte en mode terminal même si très peu de fichier seront édités directement sur le téléphone
- d’avoir la documentation des commandes
- d’avoir un serveur **ssh** qui sera l’outil pour pouvoir se logger à distance sur le téléphone.

Dans la suite le symbole ~ désigne le répertoire de base de **Termux** à savoir `/data/data/com.termux/files/home`. Il faut créer une clé d’autorisation (publique et privée) **id_rsa.pub**, **id_rsa** sur votre **téléphone**, puis démarrer le serveur **ssh** :

```
chmod 700 ~/.ssh  
touch ~/.ssh/authorized_keys  
chmod 600 ~/.ssh/authorized_keys  
ssh-keygen  
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
sshd -p 9999
```

Pour transmettre la clé privée, il faut créer un serveur **netcat** sur votre **téléphone**,

```
ifconfig  
netcat -l 8888 < id_rsa
```

puis sur votre **ordinateur**, se connecter au serveur **netcat** et récupérer la clé :

```
cd ~/.ssh  
netcat 192.168.42.IPTEL 8888 > id_rsa  
chmod 600 id_rsa
```

Une session distante **ssh** peut maintenant être ouvert depuis votre **ordinateur** :

```
ssh 192.168.42.IPTEL -p 9999
```

2 Outils

Le but de cette section est de lister les quelques outils qui vous seront utiles au cours du projet.

2.1 Shell de base

Un cours (beaucoup trop complet) est disponible à https://download.tuxfamily.org/linuxvillage/Informatique/Cours-Shell-prog/cours_shell_unix.pdf. Les quelques commandes utiles sont **chmod**, **cp**, **grep**, **ls**, **man**, **mkdir**, **more**, **mv**, **rm**, **tail**, **unzip**, **zip**. L'aide de ces commandes est accessible grâce à **man**.

2.2 Outils réseau de base

Les quelques commandes utiles sont **ifconfig**, **netcat**, **ping**, **scp**, **ssh**, **sshd**.

2.3 Vim

Dans une session distante sans déport graphique (ce qui sera votre cas), il faut pouvoir éditer des fichiers texte dans le terminal grâce à **vim**. Un cours est disponible sur <https://openclassrooms.com/courses/reprenez-le-controle-avec-l-aide-de-linux/vim-l-editeur-de-texte-du-programmeur>

Les commandes utiles sont : **a**, **dd**, **dw**, **Esc**, **i**, **p**, **v**, **y**, **\$**, **:q**, **:w**, **/**, **>>**, **<<**, **"+y**

2.4 Python

Un cours (beaucoup trop complet) est disponible sur <https://openclassrooms.com/courses/apprenez-a-programmer-en-python>. La version du langage utilisée est **python 3.5.X**.

3 Démarrage projet

Cette section récapitule les informations nécessaires au démarrage des projets. Le langage de programmation est le **python3**, les librairies utiles sont : **socket**, **time** , **sys** , **pickle**, **pygame**, **xloborg.XLoBorg**, **ev3dev.ev3**. Vous aurez potentiellement besoin suivant vos choix de **numpy**, **scipy**, **matplotlib.pyplot**.

3.1 EV3 - ev3dev

La documentation concernant la librairie python permettant de contrôle des capteurs et actionneur lego est disponible à <http://python-ev3dev.readthedocs.io/en/latest/>.

3.2 Android - Qpython

Votre session **ssh** et vos programmes seront probablement situés dans un dossier sur votre carte SD de votre téléphone. Vous éditez vos programmes sur votre téléphone via **ssh / vim** à partir de votre ordinateur. Dans la console de **qpython** , il faudra donc se placer dans ce dossier :

```
import os  
os.chdir('CHEMIN')
```

Ensuite, vous pourrez exécuter vos programmes grâce à :

```
exec(open("./FILENAME").read())
```

D'autre part, la librairie **SL4A** permet d'avoir accès aux capteurs et est décrite dans <http://www.mithril.com.au/android/doc/SensorManagerFacade.html>.

3.3 RPI - Xloborg

La carte de capteurs rajoutée au RPI est décrite sur <https://www.piborg.org/xloborg/install>. Les onglets Getting Started et Examples décrivent l'utilisation des capteurs de la carte au sein d'un code python.

3.4 Python - Pygame

Les modules **surface**, **time**, **mouse**, **draw**, **event** seront probablement les premiers que vous testerez. De l'information est disponible sur la page de documentation du site de pygame <http://www.pygame.org/docs/>.

3.5 Python sockets

Il existe de nombreux cours / exemples de l'utilisation des sockets. Les outils de base de la librairie **socket** devraient suffire. Des sockets **TCP** devraient convenir [http:// apprendre-python.com /page-reseaux-sockets -python-port](http://apprendre-python.com/page-reseaux-sockets -python-port).