Procédure d’installation

Poppy-core



## Création de la VM

Avec VMWare, créer une machine virtuelle à l’aide d’une image Debian. A noter que l’on peut se passer de l’environnement bureau.

Une fois installé, pour des raisons de sécurité et de confort, installez « sudo » et un serveur ssh:

$ su

$ [password]

# apt-get install sudo

# apt-get install openssh-server

Sudo doit être configuré, il faut ajouter l’utilisateur au fichier sudoers. Executez la commande « visudo » et ajoutez « %[USERNAME] ALL=(ALL :ALL) ALLL » sous « %sudo ALL=(ALL :ALL) ALLL »

Le pilote de la carte son doit être réglé de préférence sur « ICH AC97 »

La carte réseau de VMware doit être configurée en « accès par pont », connecté sur un réseau sans pare-feu, et configuré sur la carte réseau utilisé (wifi/Ethernet/..).

Un redémarrage est nécessaire pour appliquer les changements de configuration. La commande « ip a » permet de récupérer l’adresse IP de la machine virtuelle, qui est accessible en ssh.

## Installation du projet (sources) et des dépendances

Les dépendances seront installées à l’aide de miniconda, une distribution python peu encombrante. Poppy fonctionnant avec python 2, il faudra utiliser la version 2 de miniconda. Elle est aujourd’hui accessible à l’adresse <https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86.sh>. Exécutez les commandes :

$ sudo apt-get install curl

$ curl –o miniconda.sh https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86.sh

$ sudo chmod +x miniconda.sh

$ ./miniconda.sh

A la question « Do you wish the installer to prepend the Miniconda2 install location to Path » répondez « yes ». Pour mettre à jour le PATH contenant le dossier d’installation de miniconda, exécutez :

$ source ~/.bashrc

Installez ensuite les dépendances du projet :

$ conda install numpy scipy notebook jupyter matplotlib

Sans oublier les libraires python :

$ sudo apt-get install build-essential

$ sudo apt-get install python-pyaudio portaudio19-dev

$ pip install poppy-torso –user –U

$ pip install pypot pygame gtts tts\_watson twisted autobahn speechRecognition boto3

Puis créer le fichier ip.txt, et ajoutez l’ip/port du controller en première ligne de celui-ci :

$ mkdir resources

$ touch ip.txt

Un dossier audio doit aussi être créé dans le doapO  
ssier « resources » avec un fichier list.csv contenant en première ligne « text,file »

$ mkdir audio

$ touch list.csv

Le projet « poppy-core » peut être récupéré à partir de svn :

$ sudo apt-get install subversion

$ svn checkout --username [username] https://coconet-svn-sogfr-01.fr.sogeti.com/svn/repos/projet\_poppy\_core

Pour accéder à la carte USB2AX, l’utilisateur a généralement besoin d’appartenir au groupe « dialout » :

$ sudo usermod –a –G dialout poppy

L’utilisateur devra se déconnecter pour appliquer ces changements.

## Déploiement d’une version

$ python setup.py build && python setup.py install

## Exécution d’une version

Si vous n’avez pas le contrôleur JEE il se situe dans le dépôt du RobotPrésentateur, et peut être téléchargé avec la commande :

$ svn checkout --username [username] https://coconet-svn-sogfr-01.fr.sogeti.com/svn/repos/presentateur\_sogeti

Démarrez le contrôleur JEE et modifier le fichier ip.txt dans le dossier ressources de l’utilisateur, en mettant l’IP/port correspondant au contrôleur.

Une fois dans le répertoire principale (/home/[user]), le script peut être lancé avec les commandes :

$ python

>> from cherry import Cherry

>> cherry = Cherry(use\_http=True, start\_services=True, http\_host="poppy")

Il est aussi possible de créer un script « run-poppy.py » :

from cherry import Cherry

cherry = Cherry(use\_http=True, start\_services=True, http\_host="poppy")

Qui pourra ensuite être lancé avec « python –i run-poppy.py ».