

INTERNET OF THINGS
Lab1: Smart building
Master HES-SO

Émilie GSPONER, Grégory EMERY

8 mars 2016
version 1.0

Table des matières

1	Mise en place de l'infrastructure	3
1.1	Accès ssh à la raspberry pi	3
1.1.1	Configuration pour VirtualBox	3
1.2	Connexion du capteur	4
2	Fonctions à implémenter	5
2.1	get_luminance	5
2.2	get_motion	5
2.3	get_battery_level	5
2.4	get_all_measures_sensors	5
2.5	addNode	5
2.6	removeNode	5
2.7	get_nodes	5
3	REST Client API	6

1 Mise en place de l'infrastructure

1.1 Accès ssh à la raspberry pi

La raspberry pi a été configurée avec l'adresse IP 192.168.1.2. Il faut configurer l'ordinateur avec une adresse IP 192.168.1.x et le masque réseau 255.255.255.0, cela permet aux deux appareils de communiquer.

1.1.1 Configuration pour VirtualBox

Pour partager la connexion dans une machine virtuelle, il faut régler le réseau en *Accès par pont* comme le montre l'image ci-dessous.

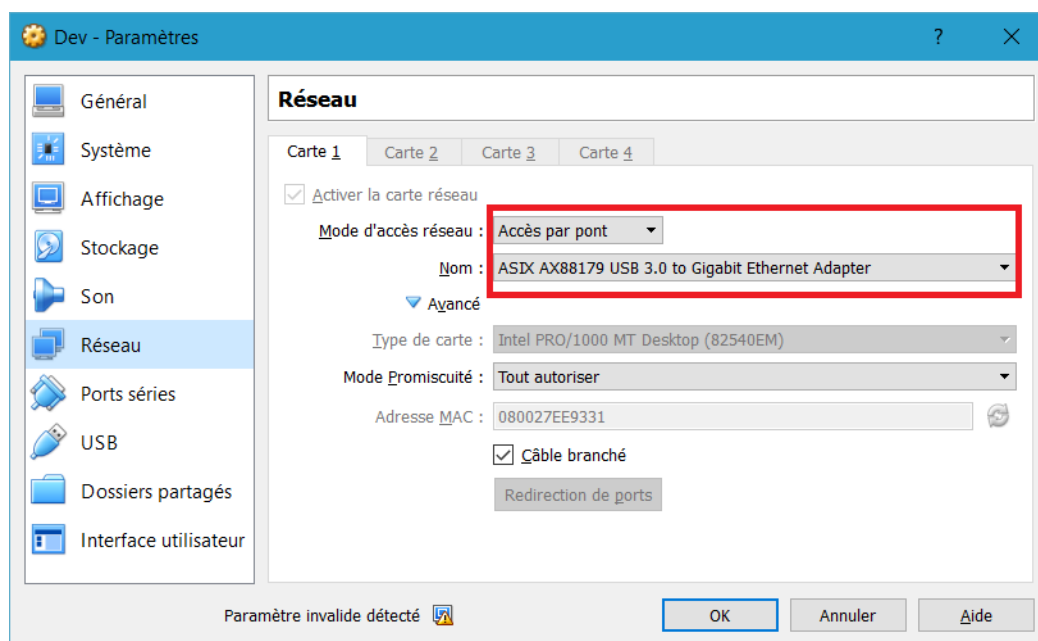


FIGURE 1 – Configuration de la carte réseau

Il faut également couper la connexion wifi de l'ordinateur, cela lui permettra de se brancher sur la connexion Ethernet.

La dernière étape consiste à aller modifier l'adresse IP de la machine virtuelle en changeant les paramètres IPv4. Nous lui avons attribué l'adresse 192.168.1.100.

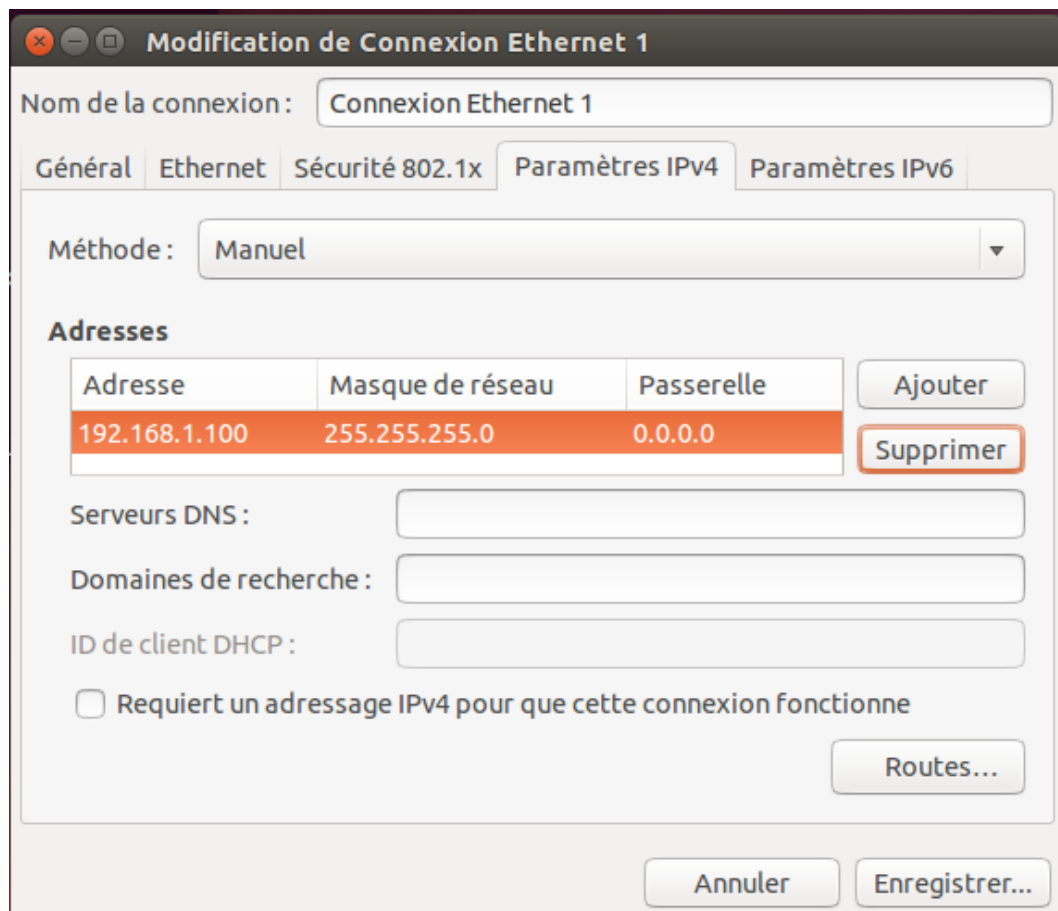


FIGURE 2 – Changement de l'adresse IP

Il est maintenant possible d'accéder à la raspberry pi par ssh :

```

1 $ ssh pi@192.168.1.2
2 The authenticity of host '192.168.1.2 (192.168.1.2)' can't be established.
3 ECDSA key fingerprint is 80:b2:dc:9d:a5:4e:e7:1a:32:60:11:1c:8b:44:39:0e.
4 Are you sure you want to continue connecting (yes/no)? yes
5 Warning: Permanently added '192.168.1.2' (ECDSA) to the list of known hosts.
6 pi@192.168.1.2's password:
7 ...
8 pi@raspberrypi:~ $

```

1.2 Connexion du capteur

Pour connecter le capteur à la clé Z-wave, il faut suivre les étapes suivantes :

1. Déconnecter le Z-Stick de la Raspberry
2. Presser le bouton du Z-stick, la LED doit clignoter lentement en bleu
3. Presser le bouton du capteur

4. Quand le capteur est inclu/détecter par le Z-Stick, la LED des deux éléments reste figée (vert pour le capteur, bleu pour la clé) quelques instants
5. Le stick peut ensuite être branché à la Raspberry

2 Fonctions à implémenter

Nous nous sommes basés sur la documentation de cette adresse pour implémenter les fonctions, https://python-openzwave.googlecode.com/hg-history/c2d93bd3574e73b828dd86da40d80bbcd8c/docs/_build/html/index.html

2.1 get_luminance

2.2 get_motion

2.3 get_battery_level

2.4 get_all_measures_sensors

2.5 addNode

2.6 removeNode

2.7 get_nodes

3 REST Client API

Même si nous n'avions aucune connaissance préalable du REST (discuté lors du premier cours), nous avons tout de même implémenté le client.

Nous avons choisi d'utiliser QtCreator et créé une petite application en C++.

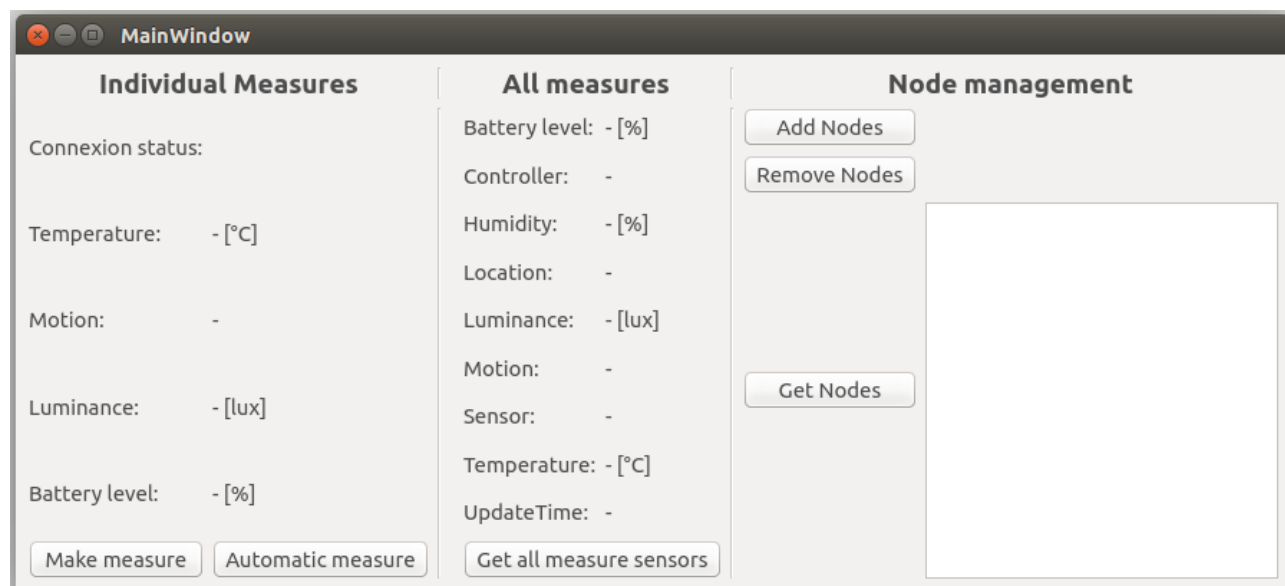


FIGURE 3 – Vue par défaut de l'API

L'API essaie de communiquer avec le nœud numéro 3. Nous n'avons implémenter la possibilité de changer le numéro.

La vue de l'application a été séparée en trois parties distinctes. La première va appeler à la suite les différentes méthodes retournant la température, la luminosité, etc... La partie centrale va appeler la fonction *get_all_measures_sensors*, tandis que la dernière partie permet d'ajouter/supprimer et voir les nœuds du réseau.

La première partie possède un champ indiquant si le nœuds répond ou non aux requêtes. Elle contient également un bouton enclenchant le mode automatique, les mesures sont répétées toutes les 10secondes, même si nous avons vu que les valeurs se mettent uniquement à jour chaque 10 minutes.