

INTERNET OF THINGS
Lab1: Smart building
Master HES-SO

Émilie GSPONER, Grégory EMERY

11 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
3	REST Client API	5
3.1	Lancement de l'application	5
3.1.1	Utilisation de l'exécutable	5
3.2	Utilisation du projet QT	5
3.3	Vues de l'application	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.

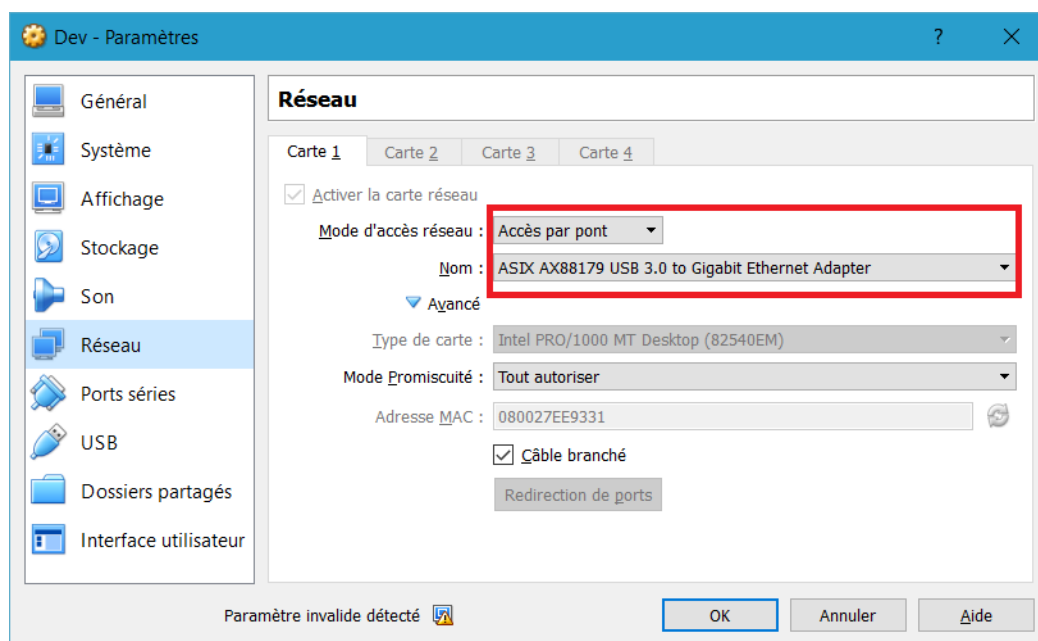


FIGURE 1 – Configuration de la carte réseau

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.

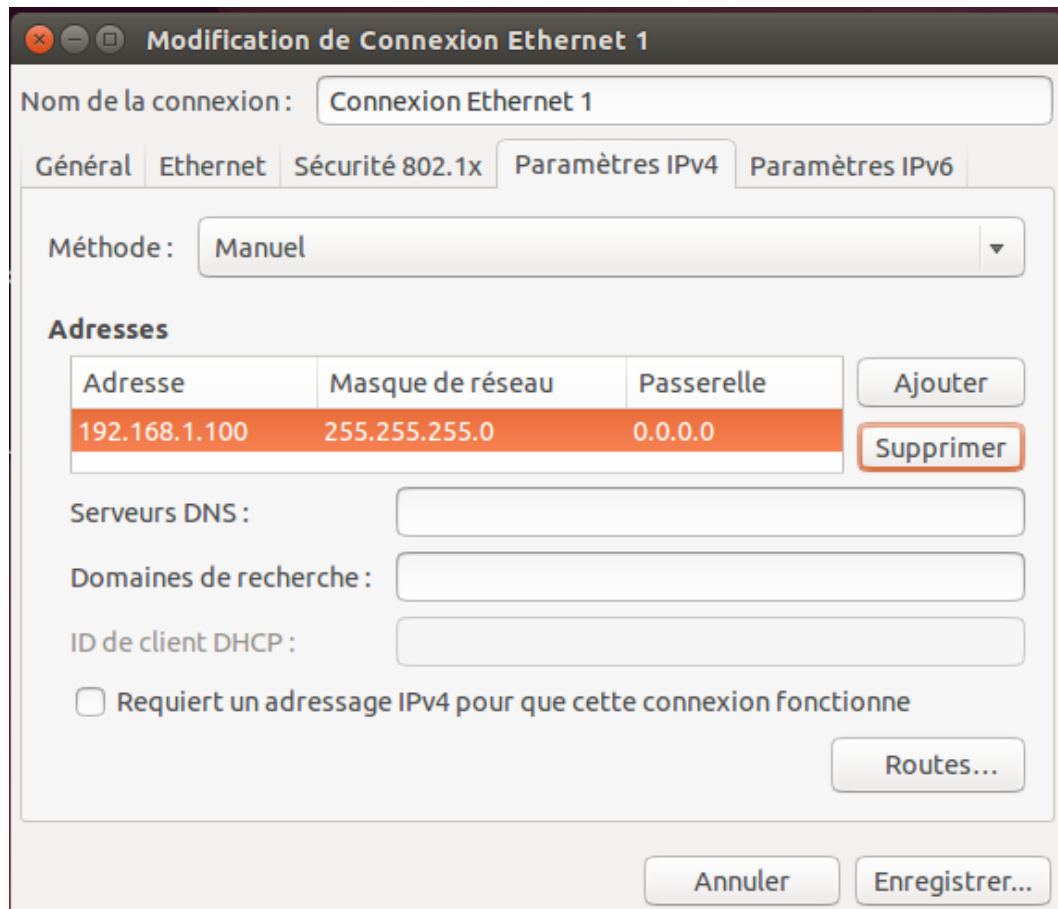


FIGURE 2 – Changement de l'adresse IP

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 devons implémenter les fonctions du groupe 2, soit :

1. `get_luminance`
2. `get_motion`
3. `get_battery_level`
4. `get_all_measures_sensors`
5. `addNode`
6. `removeNode`
7. `get_nodes`

Emplacement du fichier source : */backend.py*

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++.

3.1 Lancement de l'application

3.1.1 Utilisation de l'exécutable

L'application a été conçue pour une distribution Linux. Nous avons mis à disposition un exécutable pouvant être lancé de la manière suivante.

```
1 $ cd executable/  
2 $ ./RESTClient
```

Emplacement de l'exécutable : */executable/RESTClient*

La vue par défaut de l'application s'affiche.

3.2 Utilisation du projet QT

Nous avons également inclus les sources du projet au cas où l'exécutable ne fonctionnerait pas, ou pour compiler le projet pour un autre environnement que Linux.

Emplacement des sources : */RESTClientApp*

Pour utiliser ces fichiers, il suffit d'installer QT Creator et de double cliquer sur le fichier *.pro des sources, cela va ouvrir le projet. La dernière étape est ensuite de lancer l'application avec le bouton *Run*.

3.3 Vues de l'application

Au lancement de l'application, on obtient la vue ci-dessous. La fenêtre a été séparée en trois parties distinctes. La partie centrale va appeler la fonction *get_all_measures_sensors*, tandis que la dernière partie permet de gérer le réseau.

Nous avons également inclu un champ permettant de spécifier l'ID du capteur avec lequel on souhaite communiquer.

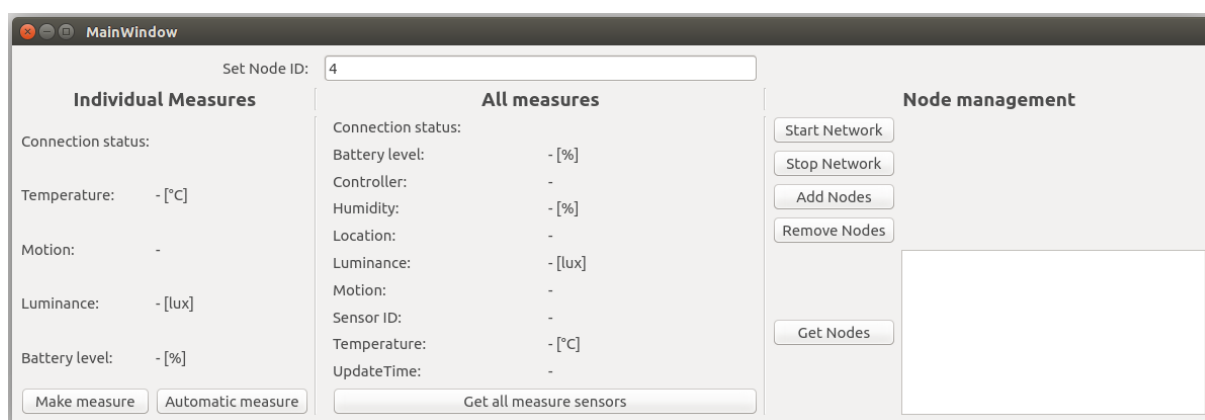


FIGURE 3 – Vue par défaut de l'API

La première partie de la fenêtre va appeler à la suite les différentes méthodes retournant la température, la luminosité, etc...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. Il y a également un champ indiquant si le statut de la communication avec le nœud.

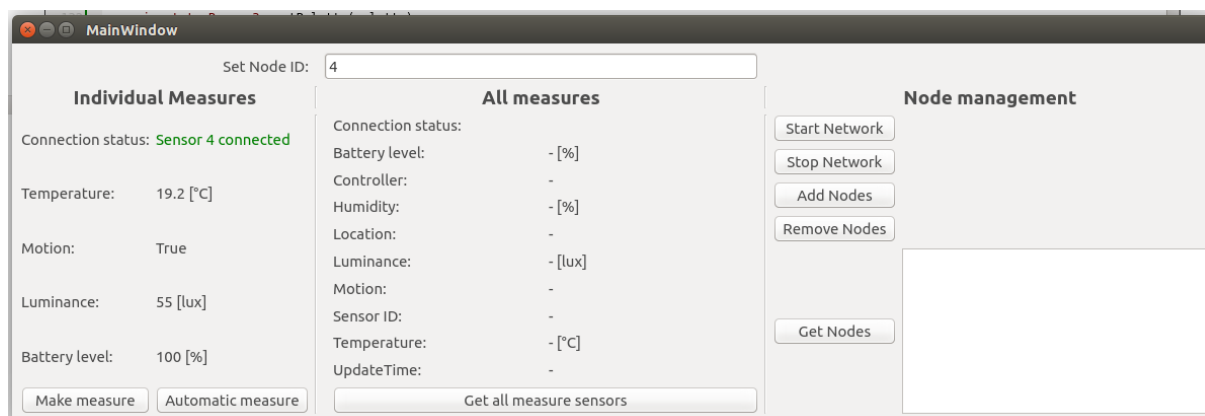


FIGURE 4 – Mesures individuelles

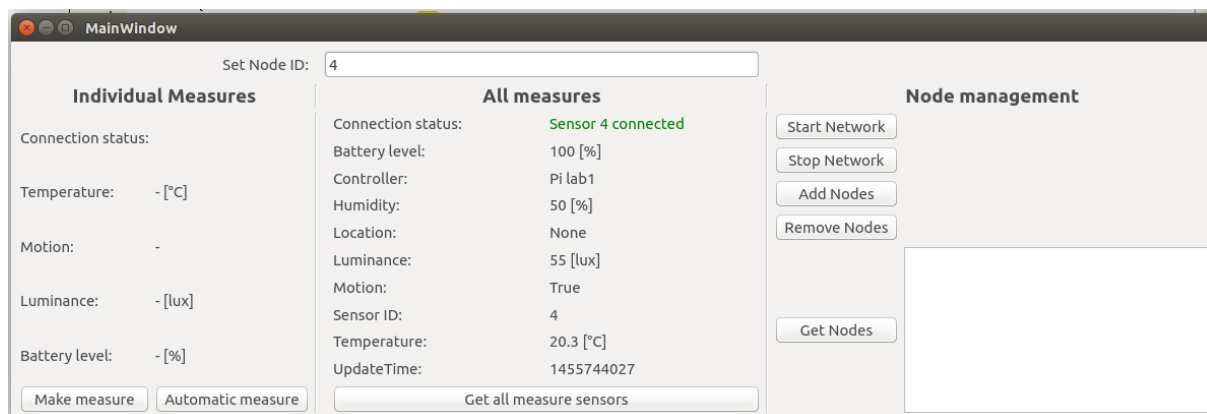


FIGURE 5 – Get all measures

Lorsque l'on appuie sur *Stop network*, le réseau est arrêté et le nœud ne communique plus, un message s'affiche pour informer l'utilisateur.

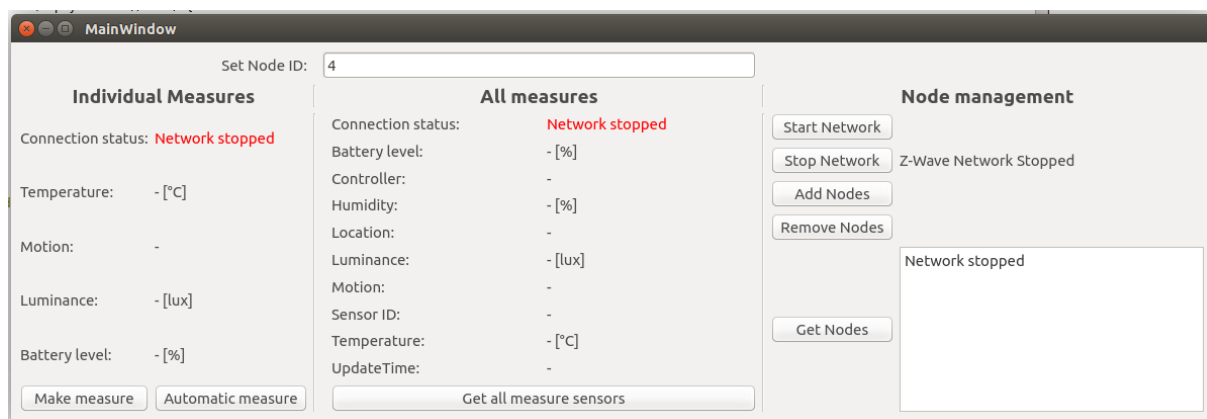


FIGURE 6 – Arrêt du réseau

Le nœud reprend la communication dès que l'on presse *Start network*.

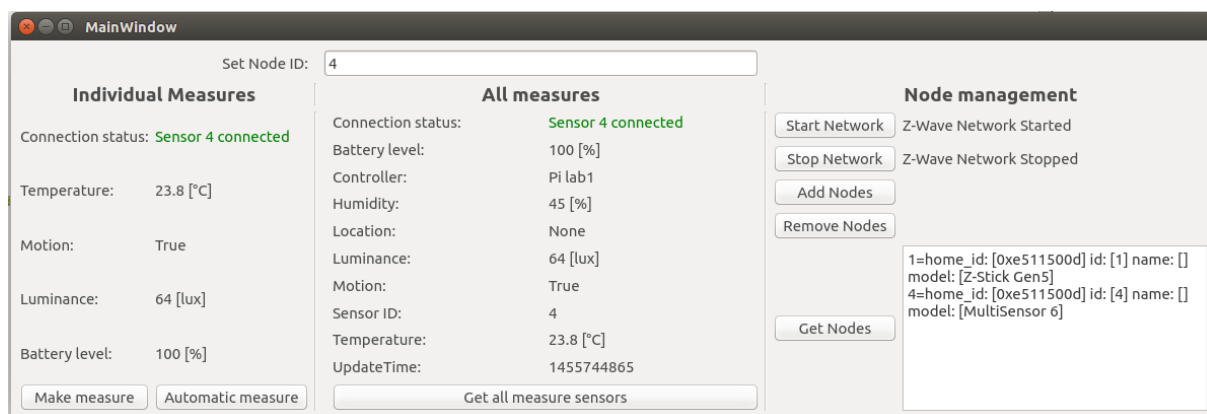


FIGURE 7 – Démarrage du réseau

Pour enlever un noeud du réseau, il faut cliquer sur *Remove node*. Le Z-wave passe en mode exclusion pour 20 secondes. Si l'on appuie sur le bouton du capteur, celui-ci est exclu du réseau. La figure ci-dessous montre ce processus. Le bouton *Get nodes* permet d'afficher à tout instant les nœuds présents dans le réseau.

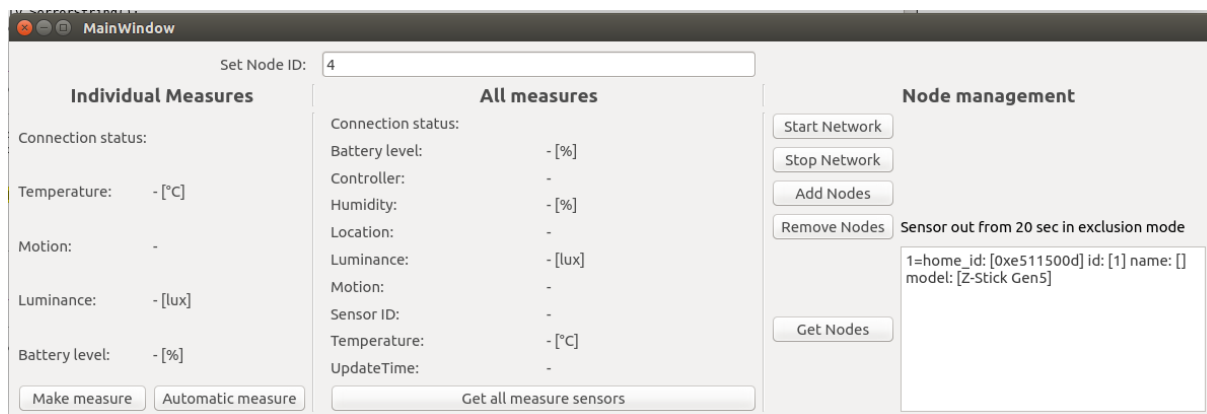


FIGURE 8 – Exclusion du capteur

Le bouton *Add nodes* permet de rajouter un nœud dans le réseau. Le Z-wave passe en mode inclusion pour 20 secondes. Si on appuie sur le bouton du capteur, il rejoindra le réseau. Vu que notre nœud a été précédemment exclu, il est rajouté dans le réseau avec un ID incrémenté de un. Notre capteur a maintenant l'ID 5.

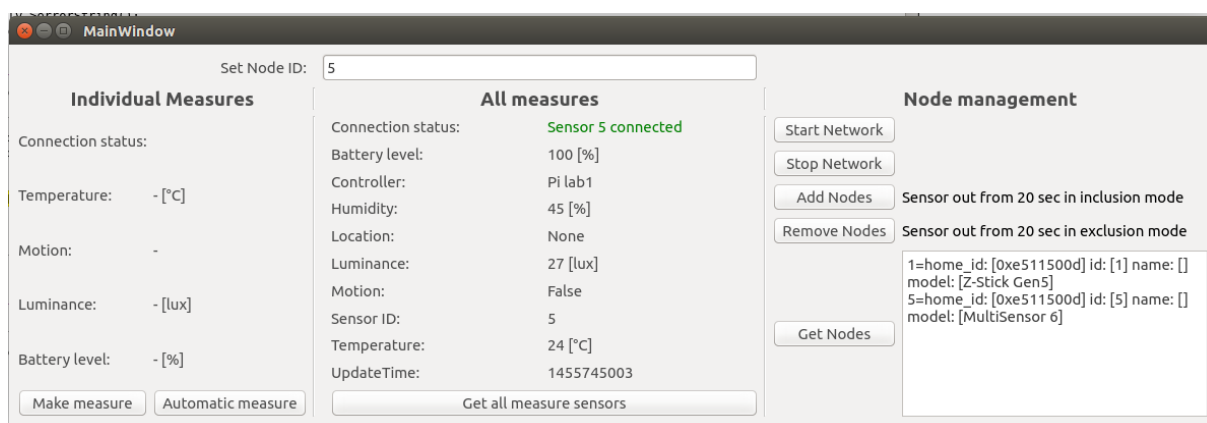


FIGURE 9 – Inclusion du capteur