

27 mai 2015

## Serveur domotique

TERRIE Corentin  
CROS Bastien  
MONNIER Matthias

## Table des matières

<b>1</b>	<b>Presentation Projet</b>	<b>2</b>
<b>2</b>	<b>Partie Android (client)</b>	<b>2</b>
<b>3</b>	<b>Partie Raspberry (serveur)</b>	<b>2</b>
3.1	Première version du serveur : Connexion à un seul client . . . . .	3
3.2	Seconde version du serveur : gestion de plusieurs clients . . . . .	4
3.3	Troisième version du serveur : échanges de trame au format JSON	4
3.4	Quatrième version du serveur : lancement de tâche différentes selon le type de connexion . . . . .	4
<b>4</b>	<b>Protocoles domotiques</b>	<b>4</b>
4.1	Les protocoles grand public . . . . .	4
4.2	Les protocoles de rupture technologique . . . . .	4
<b>5</b>	<b>Annexes</b>	<b>6</b>

# 1 Présentation Projet

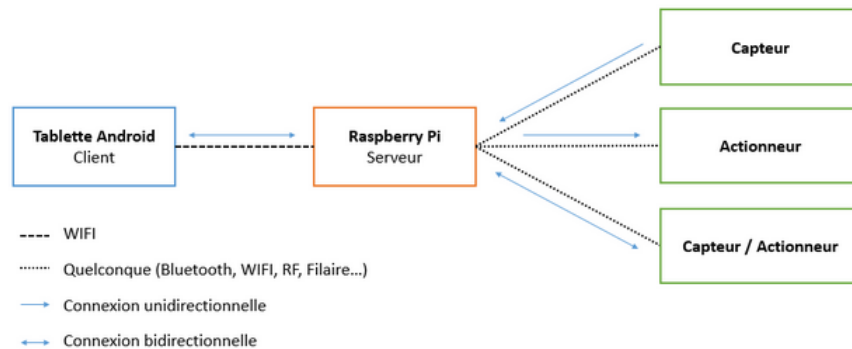


FIGURE 1 – Schéma montrant les différentes connexion du réseau.

## 2 Partie Android (client)

## 3 Partie Raspberry (serveur)

Le serveur servira a centralisé tout le réseau domotique. Il aura pour tâches :

- Possibilité de stocker les donnés des capteurs si liaison Tablette-Rasberry coupée.
- Gestion des données différentes en fonction de la connexion à un client Android ou un capteur/ actionneur.
- Si connexion à un Android :
  - Envoie des informations concernant les capteurs, les actionneurs, de métrologie, etc. (à définir).
  - Réception des ordres de la tablette : allumer tel lampe, se connecter à tel capteur, etc. (à définir).
- Si connexion à un capteur, simplement lire les informations.
- Si connexion à un actionneur, le commander (selon les ordres reçu de la tablette) et gérer son état.
- Calcul de métrologie pour chaque connexion.

Pour implémenter le serveur, nous avons choisi d'utiliser une Raspberry B+, embarquant un système dérivé de Débian : Raspbian. Cela nous as permit d'implémenter un serveur linux, utilisant les fonctions unix.

### 3.1 Première version du serveur : Connexion à un seul client

Le premier serveur implémenté sur la Raspberry permettait seulement de gérer une connexion client/serveur la plus simple possible : le client se connecte sur le port du serveur (ici 8888) et grâce à l'adresse IP du serveur à celui-ci, et le serveur notifie sur le terminal que la connexion a bien été effectuée. Ensuite le client demande à l'utilisateur de taper un message à l'écran, que le serveur lui renvoie. Finalement le client affiche dans son terminal la réponse du serveur ce qui permet de vérifier que le serveur a bien reçu le message.

**Connexion au client Linux** Le premier test a été effectué en implémentant un client linux sur la Raspberry. Ainsi, nous pouvions valider le serveur grâce au client, avant de tester la connexion avec le client Android.

Les premiers résultats ont été concluants en terme de connexion, c'est à dire que le serveur reçoit la demande de connexion du client et renvoie le message du client. Le client, quant à lui, reçoit et affiche le message renvoyé par le serveur. Cela nous a permis de valider la création du serveur et la connexion au client. Par contre, nous avons rencontré un problème au niveau du buffer avec le client. En effet, le message affiché à l'écran était bien celui initialement envoyé, mais le client affiche aussi tous les caractères présents dans le buffer.

```
baba@baba:~/Documents/Projet_Domotique/Programme$ ./serveur
Socket created
bind done
Waiting for incoming connections...
Connection accepted
```

0.3  
(a)  
b

FIGURE 2 – connexion au client.

```
baba@baba:~/Documents/Projet_Domotique/Programme$ ./client
Socket created
Connected

Enter message : test
Server reply :
test
Enter message : test 2
Server reply :
test
Enter message : Server reply :
2est
Enter message : test 3
Server reply :
test
Enter message : Server reply :
3est
```

0.3  
(a)  
b

FIGURE 3 – connexion au client.

FIGURE 4 – Serveur Linux

*images/test\_reponse\_client\_impl.png* Comme solution nous avons pensé à gérer la longueur du buffer en fonction du message. Nous avons choisi de parser JSON, que nous détaillerons plus loin.

**Connexion avec le client Android** Ensuite il a fallu tester cette solution avec le client sous Android : la connexion a bien marché, mais nous avons eu un premier problème au moment de recevoir le message sur le périphérique Android.

### **3.2 Seconde version du serveur : gestion de plusieurs clients**

### **3.3 Troisième version du serveur : échanges de trame au format JSON**

### **3.4 Quatrième version du serveur : lancement de tâche différentes selon le type de connexion**

## **4 Protocoles domotiques**

### **4.1 Les protocoles grand public**

**X10** Le X10 est un vieux protocole (développé en 1975) par courant porteurs. Les modules X10 peuvent être piloté par des télécommandes radio (433MHz).

#### **Les plus :**

- Protocole le moins cher dans le domaine des automatismes résidentiels
- Communauté d'utilisateurs très active
- Bonne distribution des produits

#### **Les moins :**

- Gros problème de sécurité (toute personne possédant l'accès à une partie de l'installation électrique peut envoyer des ordres X10)
- Incompatibilité entre les réseaux électriques des différents pays
- Pas de retour d'état des modules et collisions non gérées

Le X10 est vieillissant et par conséquent de nombreux constructeurs le délaissent. Il comporte comme seul avantage d'être simple et n'est donc plus conseillé.

## **OREGON**

## **OREGON**

### **4.2 Les protocoles de rupture technologique**

Deux protocoles seront détaillés dans cette partie : Z-Wave et EnOcean. Ils partagent de nombreuses similitudes, ils sont tous deux :

- des protocoles domotiques
- des technologies sans fil
- des technologies disponibles via plusieurs centrales domotiques
- des technologies mises en œuvre par plusieurs constructeurs de périphériques.

**Z-Wave** Le Z-Wave est un protocole radio conçu pour la domotique. Il utilise la fréquence 848.42MHz. Ses caractéristiques sont qu'il est relativement sécurisé, à double sens (chaque composant est à la fois récepteur et émetteur) et qu'il utilise un réseau maillé.

**Les plus :**

- Grande richesse fonctionnelle.
- Topologie maillée du réseau sans fil (contrainte de portée levée car les périphériques relayent les informations entre eux).
- Retour d'état des périphérique (acquiescement de commande, retour d'état de la batterie etc...).
- Grande liberté de choix sur les périphériques.

**Les moins :**

- Complexité de la mise en place.
- Prix élevé des modules.
- Consommateur de pile.

Le Z-Wave est un protocole de choix pour les actionneurs mais moins pour les capteurs. Voici le résumé graphique :

**EnOcean** Le EnOcean est également un protocole radio utilisant la fréquence 848.42MHz. Ce protocole à l'avantage d'être sans fils et sans piles. En effet les périphériques puisent leur source d'énergie de leur environnement direct. Il utilise notamment l'effet photovoltaïque (transforme la lumière en électricité), l'effet piezo-électrique (transforme un choc ou une forte pression en électricité) ainsi que l'effet Peltier ou effet thermoélectrique (transforme une différence de température constaté à un instant donné en électricité).

Lorsqu'un interrupteur EnOcean utilise des cristaux piezo-électrique pour produire son électricité, il communique son ordre ON/OFF par voie hertzienne en utilisant l'énergie fournie par la pression mécanique de celui qui actionne l'interrupteur. Il peut ainsi générer le peu de courant nécessaire pour envoyer l'information à la centrale.

**Les plus :**

- Flexibilité élevée lors de la mise en oeuvre mais également en cas de modification de l'installation.
- Pas besoin de changer les piles.
- Pas de coût de consommables.

**Les moins :**

- Coût des modules plus important que certaines autres technologies.
- Design peut attrayant de certains modules.
- Retour d'état non disponible pour certains périphériques (seuls ceux ayant une batterie de stockage de l'énergie le peuvent).
- Le réseaux n'est pas maillé. Mais il existe des relayeurs graces auxquels le réseaux peut rayonner sur une portée de 200m.

Voici le résumé graphique :

Les deux graphiques précédent sont issues du site [www.abavala.com](http://www.abavala.com). Elles ne sont pas issues de mesures scientifiques mais d'avis personnels exprimés librement. Voici le graphique qui regroupe ces deux protocoles :

## **5 Annexes**