

# Projet de développement Gopiwan

”Manuel Utilisateur”



Groupe 4 : David B., Guillaume C., Nicolas B., Nailya B.

12 mai 2015

## Table des matières

0.1	Mise en marche . . . . .	3
0.1.1	Se connecter depuis un appareil mobile . . . . .	3
0.2	Scripts python . . . . .	3
0.3	Client . . . . .	4
0.4	Serveur . . . . .	4
0.5	Nota Bene . . . . .	4

## 0.1 Mise en marche

La mise en route du robot contrôlé à distance se passe normalement sans que l'utilisateur n'ait à se préoccuper d'aucun paramétrage.

Lors de la livraison du robot, il y aura tout le nécessaire pour que le robot fonctionne "out-of-the-box", c'est à dire :

Des scripts auront été placés sur le pi pour initialiser le raspberry pi en mode hotspot dès qu'il sera sous tension.

Le serveur jetty sera également lancé automatiquement.

Pour utiliser le robot, il ne suffira que de connaître l'adresse ip du raspberry :

---

`ifconfig`

---

L'interface à laquelle nous nous intéresserons est wlan0.

### 0.1.1 Se connecter depuis un appareil mobile

Utilisez simplement l'interface d'accès un point wifi sur l'appareil client, il est important de se connecter en utilisant une adresse statique sur le sous-réseau fourni par le hotspot.

Voici un exemple d'adresse statique attribuable : IP de l'interface wlan0 : 192.168.42.1 : : : IP statique de la tablette : 192.168.42.42.

puis, une fois que l'appareil mobile est enregistré sur le raspberry pi, saisir l'adresse de l'interface wlan0 dans un navigateur, suivi du champ /interface, vous arriverez sur la page d'accueil du client et aurez accès aux contrôles du robot ainsi qu'un affichage de la caméra du pi.

---

`http://192.168.42.1/interface`

---

## 0.2 Scripts python

Lors du développement de notre application, nous avons été confrontés à des problèmes techniques : matériels, mais aussi logiciels ;

en effet, au début du projet, les librairies Java n'existaient pas encore, on a donc opté pour l'utilisation des librairies python Gopigo déjà existantes, qu'on appelle par le biais de scripts python.

Deux types d'appel différents :

On lance un démon python (serveur) qui se charge de recevoir des appels qu'on lui envoie et se charge de lancer les commandes correspondantes sur le pi

On peut lancer les scripts python en appelant les scripts directement depuis java, mais c'est plus coûteux en mémoire (et un problème à considérer pour le Raspberry pi première génération).

### 0.3 Client

Le client Gopiwan, n'est ni plus ni moins qu'une page web, facilement éditable et substituable.

Nous avons utilisé html et ajax pour donner des ordres au serveur Jetty, qui sont deux technologies très largement utilisés sur le web.

Il est possible de changer la taille d'affichage de la camera du pi, la fréquence d'affichage des images.

### 0.4 Serveur

Le serveur utilisé est jetty (écrit en Java), et démarre à la mise sous tension comme dit précédemment. Ce dernier fournit une interface web basique (mais ô combien efficace pour contrôler le robot) par le biais d'un fichier html du package client. Il est agrémenté de l'API Java Jersey conforme à au modèle de programmation REST, qui utilise un système d'annotations et de paths permettant d'obtenir un code combinant lisibilité et élégance et adaptable aux souhaits du développeur.

Le serveur est fourni sous la forme d'une archive jar exécutable.

### 0.5 Nota Bene

À l'heure où ce manuel est écrit, la librairie gopigo Java est sortie, mais manque encore de maturité, c'est pourquoi nous avons choisi de rester sur la déclinaison python, mais le passage d'une librairie à l'autre est somme toute, assez simple, il faudra juste adapter légèrement certaines classes, et retirer les classes devenues obsolètes ; je ne rentre pas dans le détail, mais tous les fichiers mentionnant ".py" peuvent être remplacés ou supprimés et ainsi, l'application serait donc intégralement Java (à l'exception du client bien évidemment).