

Faculté Polytechnique



Un robot contrôlé via un Raspberry Pi Projet d'informatique

Rapport de projet

Raphaël LEJEUNE
Maximilien POTTIEZ



Sous la direction de Monsieur le Professeur
Mohammed BENJELLOUN

2015

Table des matières

1	Introduction	2
2	Matériel	3
3	Software	4
3.1	Signaux PWM	4
4	Organisation	5
5	Qui a fait quoi	8
6	Conclusion	9
A	Procédure d'installation	10
A.1	Installation de Raspbian sur le Raspberry Pi	10
A.2	Configurer le WiFi sur le Raspberry Pi	11

Chapitre 1

Introduction

Décrire le but visé, l'utilité du robot.
Faire en français et en anglais !

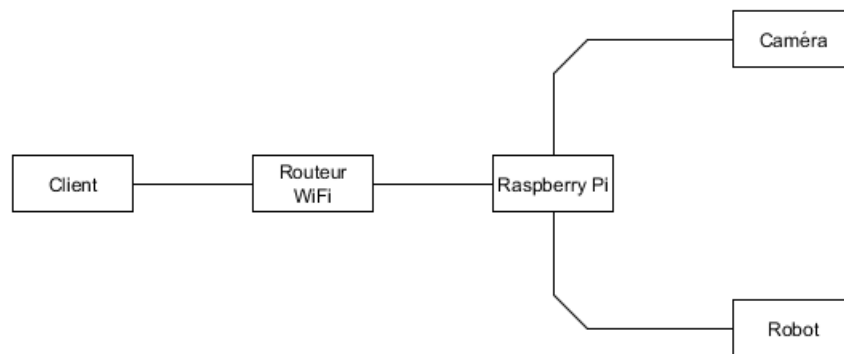


FIGURE 1.1: Matériel

Chapitre 2

Matériel

Lister le matériel utilisé, justifier sa présence

- Raspberry (pourquoi pas un Arduino, par exemple ?) - Moteurs (quel modèle ?) - Contrôleur (expliquer son utilité) - Batterie - Clé wifi - Webcam (caractéristiques techniques) - Réseau Wi-Fi opérationnel - ?

Chapitre 3

Software

Lors de la réalisation du projet, nous avons envisagé plusieurs pistes pour contrôler le robot.

3.1 Signaux PWM

Pour contrôler les moteurs, nous pouvons utiliser les signaux PWM (*Pulse Width Modulation*) : il s'agit d'ondes carrées périodiques.

Pour chaque période, on envoie une tension continue de 5 volts (qui correspond à la tension de base du Raspberry Pi, délivrée sur les ports GPIO) pendant une fraction de la période seulement. La tension « perçue » par le moteur est directement proportionnelle à cette fraction. Si cette fraction vaut par exemple 60 %, le moteur tournera comme si on envoyait 3 V, donc à 60 % de son régime maximum, si sa tension nominale est de 5 V.

Chapitre 4

Organisation

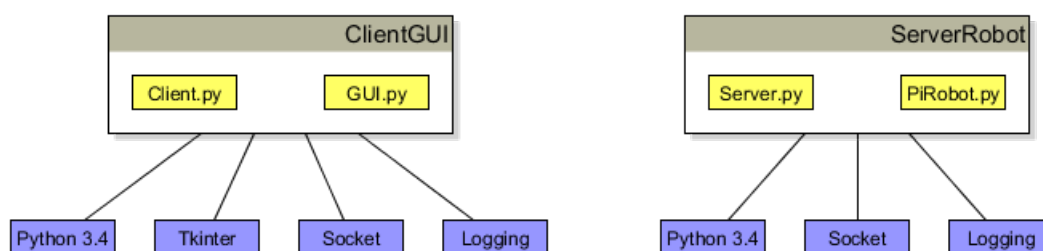


FIGURE 4.1: Logiciel

Notre code est divisé en trois parties :

- La partie client, qui tourne sur un PC (sous Windows, Linux, ...),
- La partie serveur, qui se trouve sur le Raspberry Pi,
- La partie contrôle, qui se trouve sur la carte Arduino.

Client.py sert à communiquer avec le Raspberry Pi. C'est dans ce fichier que sont récupérées toutes les requêtes de l'utilisateur, lorsqu'il appuie sur un bouton, par exemple.

GUI.py sert à tracer l'interface graphique. Nous avons voulu créer des fonctions dans ce fichier et les lier directement au bouton comme ci-après :

```
buttonstop = Button(frameRoot, text="STOP", command=buttonstopclick)
```

Mais cela a entraîné des erreurs de références circulaires (car *Client.py* dépend de *GUI.py* et *GUI.py* dépend de *Client.py*). Nous avons résolu ce problème en ne donnant pas de fonction au bouton. Dans *Client.py*, nous donnons explicitement la commande suivante :

```
GUI.buttonstop.bind("<Button-1>", buttonstopclick)
```

Cette seconde option présente deux avantages : on peut choisir le type d'événement à associer (bouton cliqué ou relâché, ...), et on peut aussi supprimer ce lien (avec la commande `unbind`).

Server.py est le programme qui tourne sur le Raspberry Pi. Son but est de recevoir les messages (depuis le client) et de les interpréter pour envoyer des instructions au robot.

PiRobot.py est une classe qui contient toutes les fonctions d'envoi de commandes au robot, ainsi quand *Server.py* reçoit un message, il n'a qu'à appeler la bonne fonction (par exemple la fonction `Stop`, qui envoie une commande au robot pour arrêter les moteurs).

Python 3.2.3 est la version de l'interpréteur utilisé. Nous précisons ce détail, car certaines fonctionnalités que nous utilisons ne portent pas les mêmes noms dans d'autres versions de Python, ou n'existent tout simplement pas.

Tkinter sert à tracer l'interface graphique. Il permet d'organiser assez facilement les éléments dans la fenêtre graphique.

Socket permet d'ouvrir une connection entre une machine hôte (appelée Serveur) et une ou plusieurs machines (appelées Clients). Une fois la connection établie, les commandes `send` et `recv` permettent d'échanger des données.

Logging sert à générer un fichier `.log`, contenant diverses informations. A titre d'exemple, la commande suivante est appelée dans la fonction `buttonstopclick` :

```
logging.debug('Button STOP click')
```

Dans le fichier `.log`, on verra cette ligne :

```
2015-07-28 18:06:08,051 root DEBUG Button STOP click
```

C'est une alternative au `print('Button STOP click')`, et qui permet de sauvegarder les actions faites lors de l'exécution d'un programme, même s'il est arrêté pour quelque raison que ce soit.

Github est un système de contrôle de révision, et peut être utilisé pour plusieurs choses :

- Garder une copie des codes. C'est la raison principale. Si l'ordinateur ou le Raspberry rencontre un problème, le code est sauvegardé.
- Il permet aussi d'enregistrer plusieurs versions d'un code, en montrant qui l'a mis en ligne, quand cela a été fait, et les différences avec la version précédente. C'est utile si on se rend compte que les modifications apportées à un code rend celui-ci inutilisable, en permettant de retélécharger un code qu'on sait fonctionnel. Sur la figure 4.2, on voit que Sero17 (c'est le pseudo de Raphaël) a modifié le fichier `ClientGUI\GUI.py` il y a dix jours.

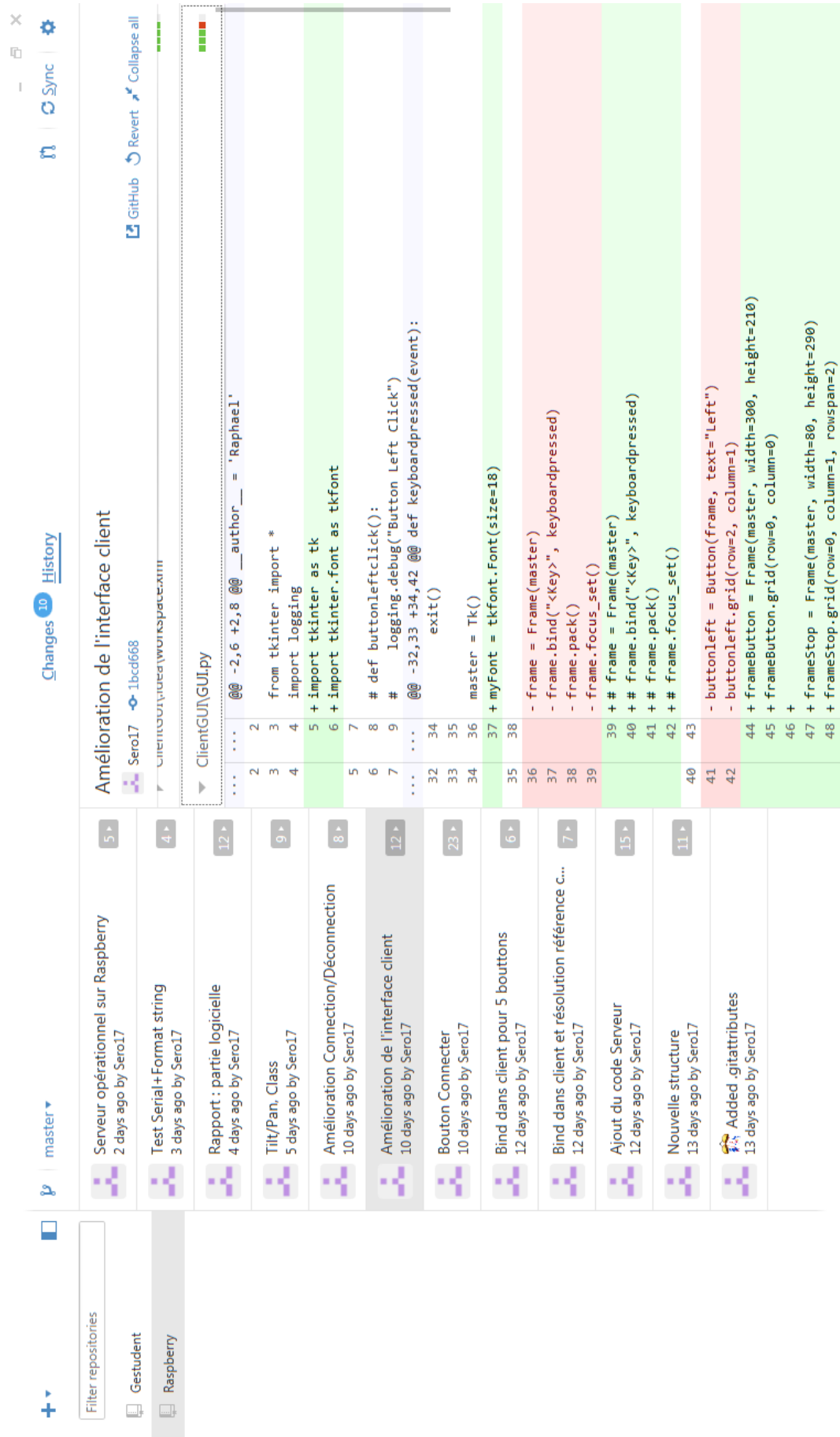


FIGURE 4.2: Interface de GitHub sous Windows

Chapitre 5

Qui a fait quoi

Raphaël

Recherches sur connection socket

Programme client, interface, programme serveur sur raspberry, connection serial avec arduino Backup régulier des codes sur GitHub, ...

Chapitre 6

Conclusion

Difficultés rencontrées, limitations, améliorations possibles, ...

Annexe A

Procédure d'installation

A.1 Installation de Raspbian sur le Raspberry Pi

Voici la procédure à suivre pour installer Raspbian sur le Raspberry :

- Téléchargez NOOBS sur <https://www.raspberrypi.org/downloads/>, et extrayez les fichiers du zip.
- Formatez la carte SD (avec https://www.sdcard.org/downloads/formatter_4/). Sélectionnez la carte SD et dans les options choisissez *FORMAT SIZE ADJUSTMENT ON*, puis cliquez sur *Format*.
- Copiez les fichiers extraits de NOOBS sur la carte SD, puis éjectez la carte et insérez-la dans le Raspberry Pi.
- Connectez le Raspberry à un écran avec un câble HDMI, et connectez aussi un clavier et une souris sur les ports USB (un récepteur sans fil convient aussi).
- Alimentez le Raspberry. Attention il est déconseillé de brancher/débrancher des câbles lorsque le Raspberry est sous tension. Si vous souhaitez changer l'écran ou un autre périphérique, éteignez d'abord le Raspberry et débranchez-le.
- Sélectionnez Raspbian comme OS et la langue, ainsi que la configuration du clavier, et cliquez sur *Installer*.
- Après l'installation (compter environ 25 minutes), redémarrez le Raspberry Pi.
- Dans les options proposées, choisissez *Enable Boot to Desktop/Scratch*, et *Desktop Log in as user 'pi'*. Ensuite dans *Advanced Options*, choisissez *SSH*, et *Enable*. Allez sur *Finish* (avec Tabulation) et redémarrez le Raspberry.
- Si vous utilisez un écran avec un câble VGA et un adaptateur, il faut modifier `\boot\config.txt`. Il faut décommenter ou ajouter les lignes suivantes :

```
hdmi_force_hotplug=1
hdmi_group=2
hdmi_mode=69
hdmi_drive=2
```

Le numéro 69 correspond à une résolution d'écran de 1920x1200, et une fréquence de 60 hz. Pour voir quel numéro correspond, voir <https://www.raspberrypi.org/documentation/configuration/config-txt.md>.

- Configurez le WiFi (voir section suivante), et mettez à jour le système en entrant ces deux commandes :

```
sudo apt-get update
sudo apt-get upgrade
```

A.2 Configurer le WiFi sur le Raspberry Pi

Nous allons configurer le Raspberry pour se connecter au routeur, et lui assigner une IP fixe.

- Lorsque le Raspberry est hors tension, insérez la clé WiFi dans un port USB.
- Toutes les informations concernant le réseau peuvent être retrouvées grâce à un autre appareil connecté à ce réseau : entrez dans un terminal ces deux commandes :

```
sudo route -n
ifconfig
```

- On édite un premier fichier de configuration : entrez cette commande :

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Ajoutez les paramètres du réseau Wi-Fi, et sauvegardez :

```
network={
    ssid="<SSID du réseau local>"
    psk="<Clé de sécurité>"
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
}
```

- On édite un autre fichier : entrez cette commande :

```
sudo nano /etc/network/interfaces
```

Ajoutez les paramètres du réseau Wi-Fi, et sauvegardez :

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
auto wlan0

iface wlan0 inet static
address 192.168.1.50
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.1
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

L'adresse 192.168.1.50 est l'adresse IP que j'ai choisie pour le Raspberry. Il faut choisir une adresse en dehors de la plage d'adresse réservée pour le dhcp.

- Réinitialisez le réseau avec ces deux commandes, ou redémarrez le Raspberry.

```
sudo ifdown wlan0
sudo ifup wlan0
```