

## Résumé projet Roskate – Ghost in Artem

### Sujet : Réalisation d'un skate contrôlé par une wiimote grâce à une raspberry pi 3

### Réalisation : Tom Mourot-Faraut et Petru Cusnir

Le projet Roskate consiste en la réalisation d'un script de serveur python à faire tourner sur une raspberry pi 3 afin de réceptionner par bluetooth les commandes de vitesse dans le but de faire avancer le skateboard électronique. Une wiimote communique via bluetooth avec le serveur et envoie des commandes de vitesse. Ces commandes réceptionnées par le raspberry pi sont ensuite transmises à un ESC (Electronic Speed Control) via les ports GPIO qui transmet à son tour au moteur électrique afin de faire avancer le skate.

### Script piskate.py :

Le script python utilise les modules GPIO et cwiid. Le code consiste en une première partie de définitions de macros :

-Quelle action pour quelle bouton de la wiimote

-On fixe le mode de GPIO en BCM et on indique quel port utiliser pour la transmission de données à l'ESC.

GPIO.BCM	Function	GPIO.BOARD	Function	GPIO.BCM
<50mA	3V3	1 2	5V	
BCM GPIO02	SDA1 ARM	3 4	5V	
BCM GPIO03	SCL1 ARM	5 6	GND	
BCM GPIO04		7 8	TX	BCM GPIO14
	GND	9 10	RX	BCM GPIO15
BCM GPIO17	SPI1 CE1	11 12	PWM0/SPI1 CE0	BCM GPIO18
BCM GPIO27		13 14	GND	
BCM GPIO22		15 16		BCM GPIO23
<50mA	3v3	17 18		BCM GPIO24
BCM GPIO10	SPI0 MOSI	19 20	GND	
BCM GPIO9	SPI0 MISO	21 22		BCM GPIO25
BCM GPIO11	SPI0 SCLK	23 24	SPI0 CE0	BCM GPIO08
	GND	25 26	SPI0 CE1	BCM GPIO07
BCM GPIO00	SDA0 VC	27 28	SCL0 VC	BCM GPIO01
BCM GPIO05		29 30	GND	
BCM GPIO06		31 32	PWM0	BCM GPIO 12
BCM GPIO13	PWM1	33 34	GND	
BCM GPIO19	SPI1 MISO/PWM1	35 36	SPI1 CE2	BCM GPIO16
BCM GPIO26		37 38	SPI1 MOSI	BCM GPIO20
	GND	39 40	SPI1 SCLK	BCM GPIO21

-On fixe les paramètres qui servent pour l'ESC (Rapport cyclique et période. Plus le rapport cyclique est élevé, plus le moteur tourne vite)

La seconde partie du code correspond au pairing et attend la connexion avec la wiimote grâce à la fonction `cwiid.Wiimote()`.

La troisième partie correspond à la fixation des limites de l'ESC afin de garantir la sécurité du moteur et de l'utilisateur.

Vient finalement la partie `UserLoop` qui écoute en permanence les indications de la wiimote (augmentation/diminution du rapport cyclique et donc de la vitesse et arrêt du moteur au cas où l'utilisateur relâche le bouton b).

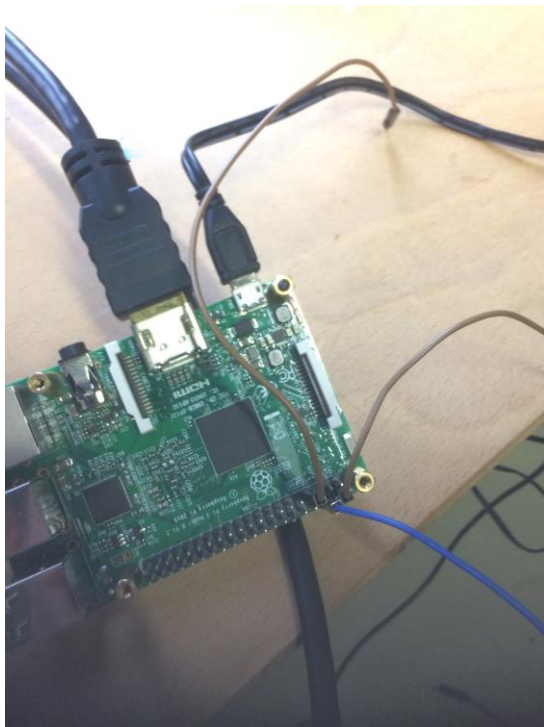
Des prints affichent à chaque instant l'instruction donnée par la wiimote au serveur afin de vérifier le bon fonctionnement du système.

## Hardware :

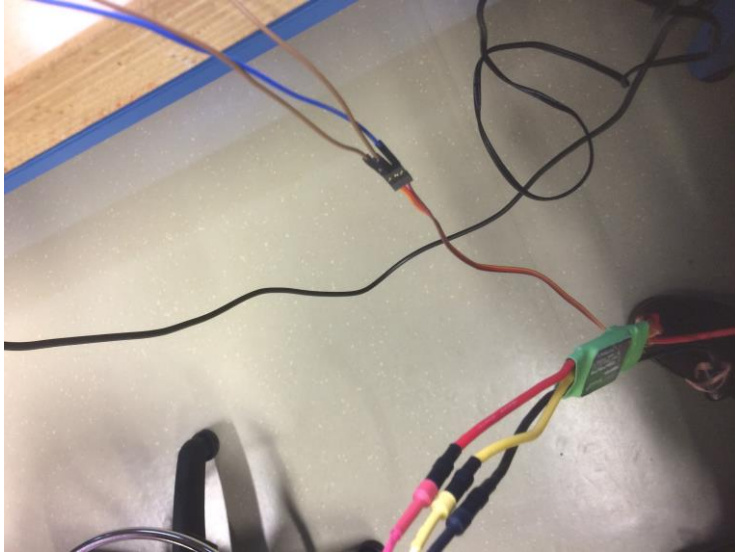
Matériel :

- Wiimote
- Raspberry pi 3
- Skateboard conversion kit de Turnigy (comprend 4 roues, 2 trucks, ESC, Moteur électrique, courroie, roulements etc...)
- Deck (à fabriquer car le deck original vient d'un penny board et n'est pas aux bonnes dimensions par rapport au kit de skate électronique)
- Générateur
- Fils / pinces croco

Nous n'avons pas eu le temps de construire le deck pour y fixer la raspberry pi. Cependant nous avons codé le script sur ce dernier. 3 fils sortent des ports GPIO et viennent se connecter à l'ESC



Raspberry pi connecté en hdmi à un écran, et via usb à une souris et un clavier. On voit les trois fils sortant des ports GPIO et allant se connecter à l'ESC du moteur.



**connexion entre les trois fils sortant du GPIO et l'ESC connecté au moteur électrique**



**Wiimote utilisée pour le contrôle du moteur**

Les ports GPIO utilisés sont (mode BCM) :

- le port n°2 : sortie 5V
- le port n°6 : GND
- le port n°7 : Sortie GPIO4 qui envoie les commandes

## Difficultés rencontrées :

Nous avons déjà passé les premières séances à essayer de transformer un raspberry pi en passerelle wifi mais l'ensemble des tutoriels que nous avons trouvé ne nous a pas permis de rendre un rendu fonctionnel. Nous avons donc perdu beaucoup de temps sur cette partie. A ce titre nous avons suivi des tutoriels tels que <https://dadarevue.com/raspberry-pi-routeur-wifi-hotspot-hostapd/> qui consiste à créer un point d'accès wifi sur son raspberry avec hostapd, et ensuite de créer un pont entre l'entrée Ethernet de la raspberry et le wifi en sortie grâce à de l'Ip forwarding du NAT et au service IPTables de linux. Cependant les choses ne se sont pas déroulées comme prévu sur notre OS, nous avons dû le réinstaller plusieurs fois et avons mélangé des approches de plusieurs tutoriels en vain. Nous n'arrivions notamment jamais à avoir l'Ethernet et le wifi activés en même temps. La modification des fichiers de paramétrage réseau s'est avéré très délicate et sans succès.

Le montage du bloc moteur sur le skateboard a été une autre difficulté rencontrée. En effet nous n'avions aucune notice livrée avec le moteur, de faite le montage c'est fait de manière empirique. Par ailleurs nous avons pu constater dans un premier temps des problèmes liés aux frottements entre différents composants du bloc moteur que nous avons rectifié par la suite. A présent le bloc moteur est correctement monté et fonctionnel.



**Moteur avec son controleur et sa courroie de transmission monté sur les 2 roues motrices**

Enfin le paramétrage du contrôleur a été un point bloquant. En effet nous n'avions aucune notice avec le contrôleur ou des informations sur son paramétrage. De faite nous n'avons pas réussi à trouver les bonnes fréquences pour le contrôler. Nous avons donc essayé de changer ses paramètres à l'aide de logiciels prétendument compatibles avec ce type d'ESC mais là encore, par manque de connaissances dans le domaine nous n'avons pas réussi.