

Commande d'un servomoteur avec le Raspberry Pi

<http://www.hertaville.com/rpipwm.html>

<http://wiringpi.com/reference/raspberry-pi-specifics/>

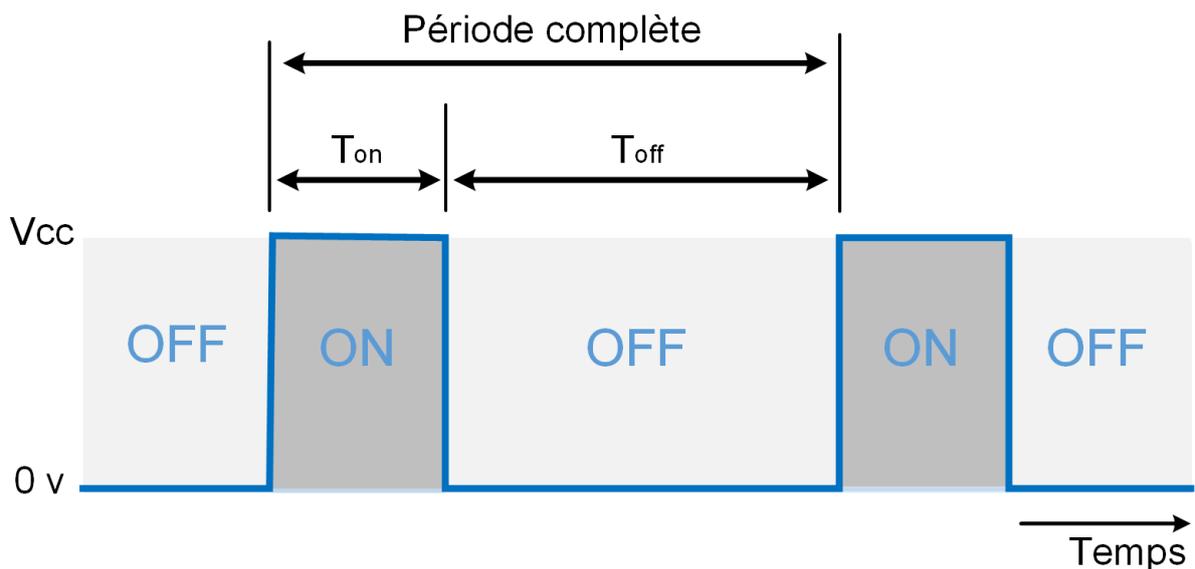
<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-8-using-a-servo-motor/software>

<https://raspberrypi.stackexchange.com/questions/11813/controlling-a-servo-with-hardware-pwm-using-wiringpi>

<https://raspberrypi.stackexchange.com/questions/4906/control-hardware-pwm-frequency>

Le PWM

Le PWM (*Pulse Width Modulation*) ou MLI en français (Modulation de Largeur d'Impulsions) consiste à faire varier le rapport cyclique d'un signal de période fixe.



Sur le signal représenté ci-dessus, de rapport cyclique 25% la période est divisée en T_{on} (25% de la période) et T_{off} (75% de la période).

Servomoteur

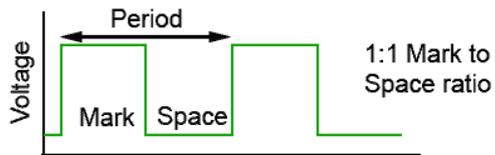


Un servomoteur est un système utilisé dans de nombreuses applications industrielles.

Il existe des servomoteurs de taille réduite, réservés à des utilisations ne demandant pas un couple important. On retrouve cette famille de servomoteurs miniatures dans les modèles réduits (voitures, avions, hélicoptères...)

C'est un de ces modèles que nous allons utiliser avec le Raspberry Pi

Hardware ou software ?

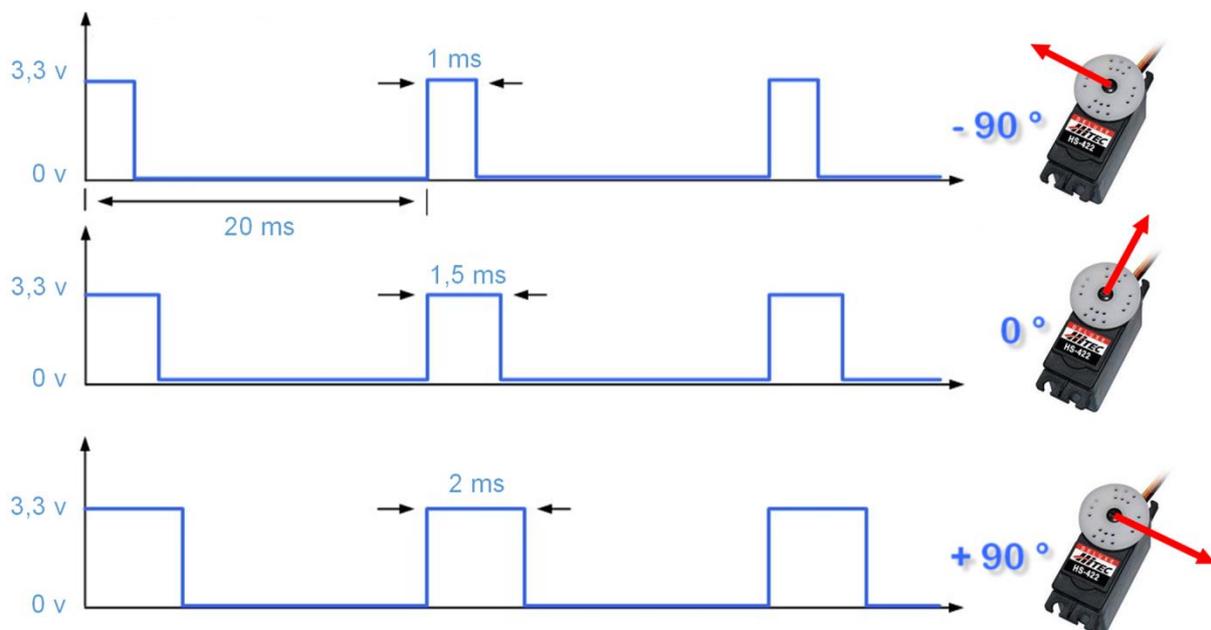


Le Raspberry Pi possède une seule sortie PWM commandée par un système intégré. C'est le GPIO 18 – Broche 12 du connecteur GPIO. Cette sortie assure un fonctionnement normal du PWM, car ce sont des registres qui sont programmés et des compteurs hardware qui fournissent le signal en sortie. Le PWM Hardware possède deux modes de fonctionnement : Balanced et Mark/Space. Ce dernier mode fournit un signal régulier répondant à notre cahier des charges, c'est lui que nous utiliserons.

Il est possible de dédier les autres sorties GPIO au PWM. La gestion est alors assurée par logiciel. Le noyau Linux utilisé n'étant pas « Temps réel », la régularité du signal est perturbée par le partage des ressources au niveau du processeur. On voit alors apparaître du « jitter » (légères variations du signal : fréquence et rapport cyclique) ce qui va se traduire par de petits mouvements ou des vibrations en sortie de servo.

Nous utiliserons donc le mode PWM hardware pour cette réalisation. Si plusieurs servomoteurs doivent être pilotés par le Raspberry Pi, il existe par exemple des cartes capables de piloter 16 servomoteurs en 12 bits (<https://www.adafruit.com/product/815>). Ce type de carte permet de gérer chaque servomoteur avec un PWM hardware, donc sans jitter.

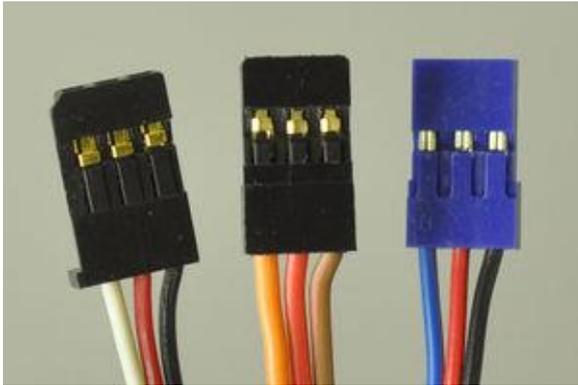
Commande du servomoteur



Les servomoteurs utilisés en radiomodélisme sont commandés par un signal à 50 Hz (période = 20ms). La position de l'axe du servomoteur est asservie à la largeur du créneau positif du signal.

- 1 ms l'axe est en butée dans le sens antihoraire
- 1,5 ms l'axe est positionné au centre de la course
- 2 ms l'axe est en butée dans le sens horaire

Connexion du servomoteur



Les servomoteurs sont équipés de prises à 3 contacts. En fonction de la marque, les couleurs de fils peuvent différer. Il n'y a pas de norme à ce sujet.

Il faut donc se reporter à la documentation du fournisseur.

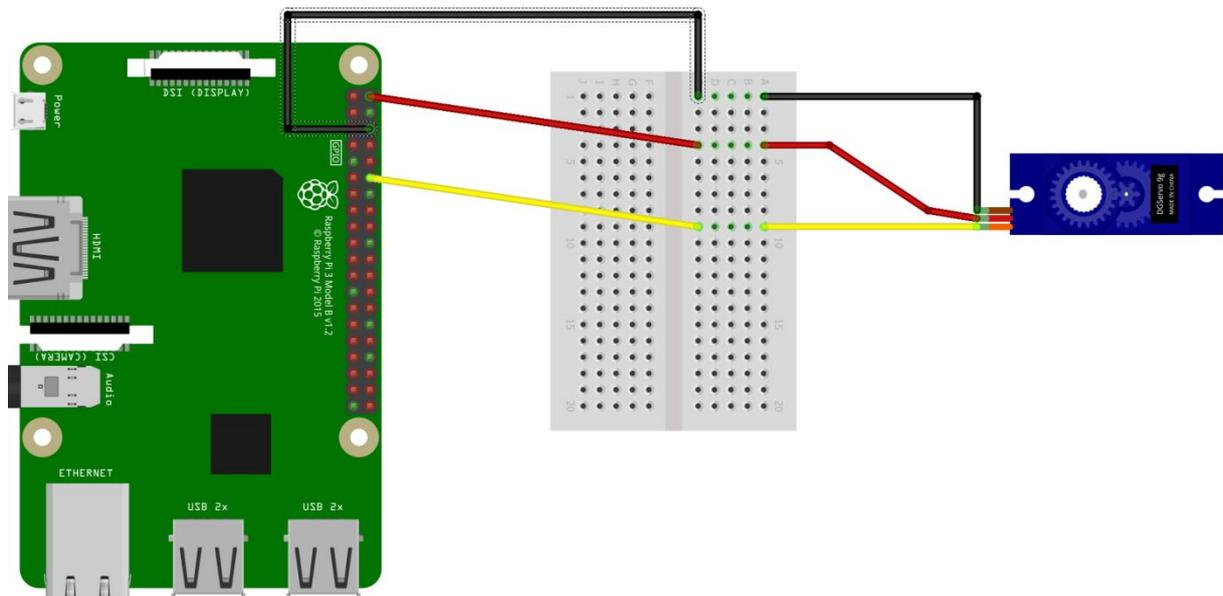
(source mchobby.be)

En général on trouve :

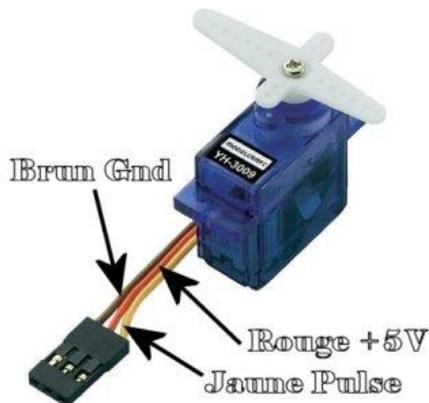
- Brun ou noir = MASSE/GND (borne négative de l'alimentation)
- Rouge = alimentation du servomoteur (Vservo, borne positive de l'alimentation)
- Orange, jaune, blanc ou bleu = Commande de position du servomoteur

Connexion au Raspberry Pi

Le Raspberry Pi dispose d'une seule broche (12 sur le connecteur GPIO) permettant un PWM Hardware.



fritzing



Relier la broche 2 (+5v) du connecteur GPIO au fil V+ du servo

Relier la broche 6 (GND) du connecteur GPIO au fil Masse du servo

Relier la broche 12 (PWM) du connecteur GPIO au fil Commande du servo

Nota 1 : L'alimentation 5v du Raspberry Pi n'est pas prévue pour alimenter un servomoteur. Il faudrait en principe alimenter le servo par une alimentation séparée. Cependant pour un modèle miniature, le débit du 5v est suffisant.

Nota 2 : Le servomoteur est alimenté en 5 v et le signal devrait avoir une tension de 5v également. La sortie GPIO du Raspberry Pi est à 3,3v. Elle est cependant suffisante pour commander le servomoteur. En production, il conviendrait d'intercaler un changeur de niveau.

Test du servomoteur en ligne de commande

Dans un premier temps nous allons vérifier le bon fonctionnement du servo moteur en ligne de commande.

Passer le GPIO 18 en sortie, mode PWM hardware

```
pi@raspberrypi:~ $ gpio -g mode 18 pwm
```

Utiliser le mode PWM Mark/space

```
pi@raspberrypi:~ $ gpio pwm-ms
```

La relation entre les différents paramètres est la suivante :

(page 139 de la doc technique

<http://www.element14.com/community/servlet/JiveServlet/downloadBody/43016-102-1-231518/Broadcom.Datasheet.pdf>)

$pwmFrequency \text{ in Hz} = 19.2 \text{ MHz} / pwmClock / pwmRange$

Configurer pwmClock et pwm Range

```
pi@raspberrypi:~ $ gpio pwmc 192
```

```
pi@raspberrypi:~ $ gpio pwmr 2000
```

soit $19.2 \times 10^6 / 192 / 2000 = 50 \text{ Hz} \Rightarrow 20 \text{ ms}$ (c'est la période attendue par le servomoteur)

Mettre la sortie du servomoteur au centre :

```
pi@raspberrypi:~ $ gpio -g pwm 18 150
```

Mettre la sortie du servomoteur à droite :

```
pi@raspberrypi:~ $ gpio -g pwm 18 50
```

Mettre la sortie du servomoteur à gauche :

```
pi@raspberrypi:~ $ gpio -g pwm 18 200
```

La plage utilisable sans emmener le servomoteur en butée va de 60 à 240.

Le centre du déplacement reste donc $60 + 240 / 2 = 150$

Exercice : Charger le script shell *servo.sh* qui automatise le test du servomoteur en le faisant se déplacer d'une butée à l'autre répétitivement (5 fois par exemple).

Il faut rendre le script exécutable (`chmod 755 servo.sh`) puis le lancer (`./servo.sh`).

Analysez le contenu du script et modifier le nombre de déplacements et/ou la vitesse de rotation.

Les instructions PWM de wiringPi

void pinMode (int pin, int mode) ;

Cette instruction met la broche en entrée (**INPUT**), sortie (**OUTPUT**), ou PWM (**PWM_OUTPUT**). Notez que seule la pin1 de *wiringPi* (BCM_GPIO 18 – broche 12 du connecteur GPIO) supporte la sortie PWM. Référez-vous à la table des N° de broches pour plus d'informations.

Cette fonction n'a pas d'effet en mode Sys : wiringPiSetupSys () ;

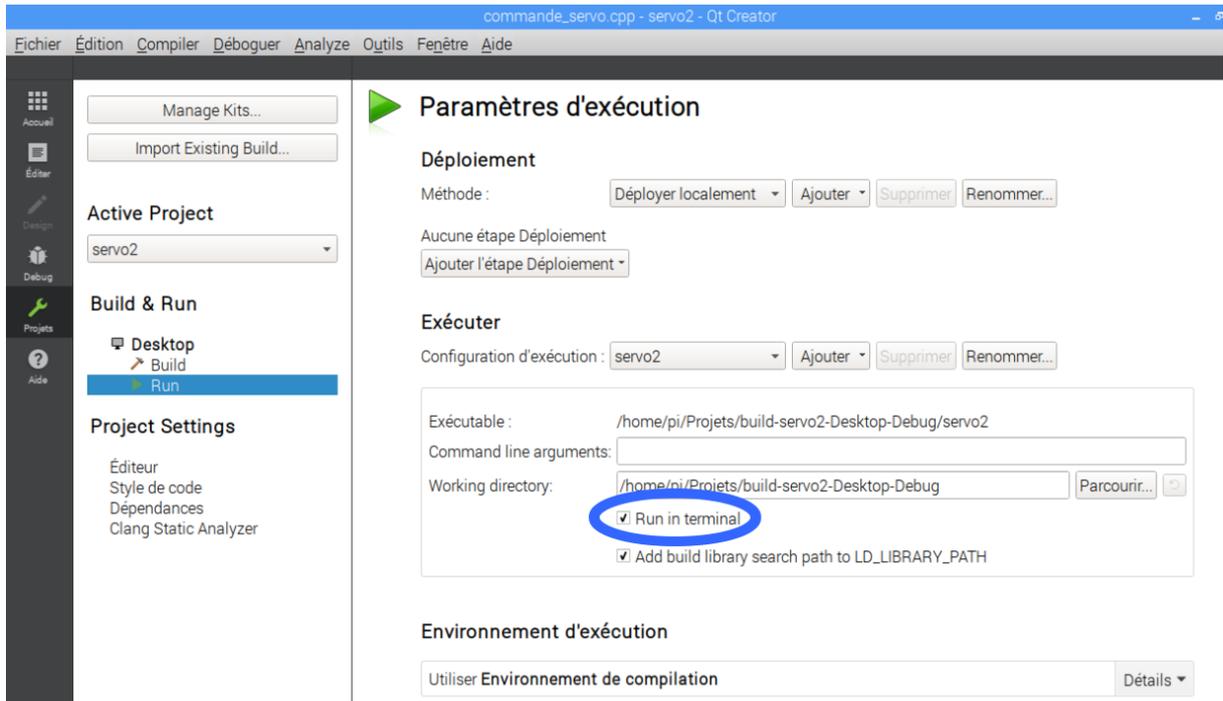
void pwmWrite (int pin, int value) ;

Ecrit la valeur dans le registre PWM pour la pin désignée.register. Cette valeur peut être comprise entre 0 et 1024. Notez à nouveau que seule la pin1 de *wiringPi* (BCM_GPIO 18 – broche 12 du connecteur GPIO) supporte la sortie PWM. Référez-vous à la table des N° de broches pour plus d'informations.

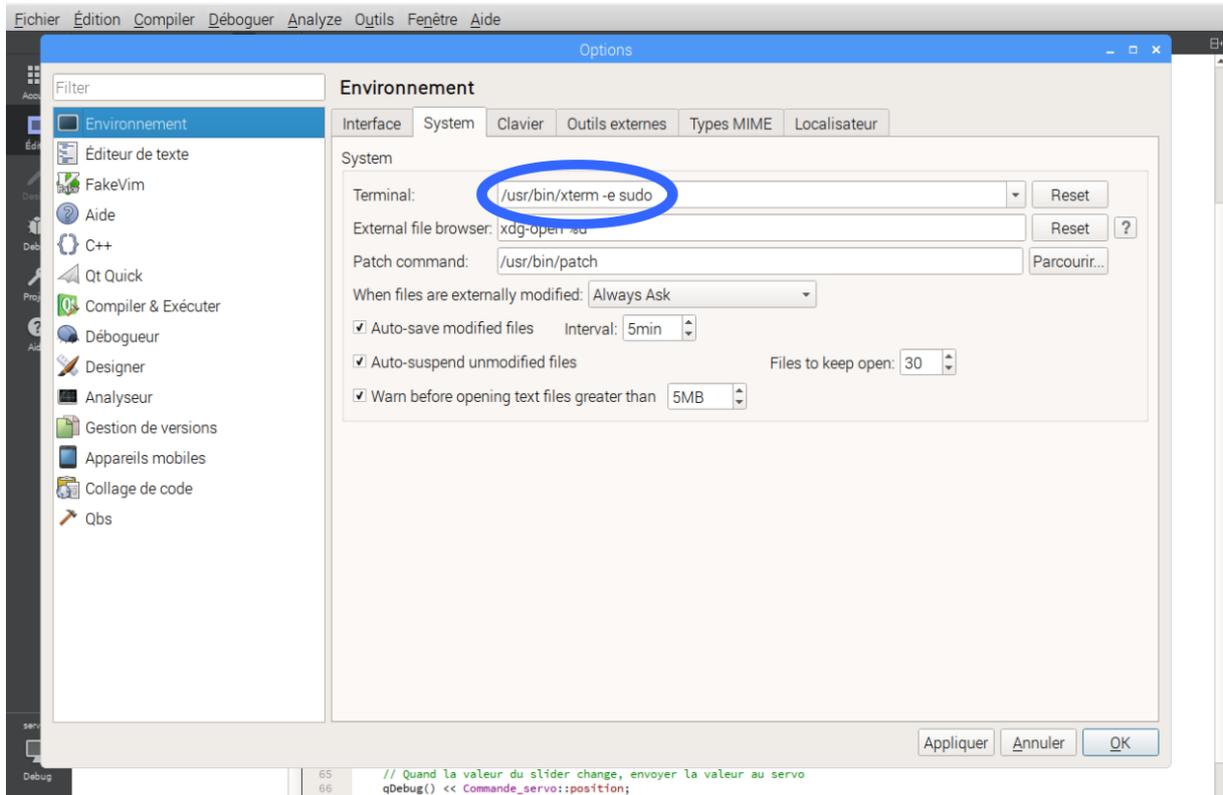
Cette fonction n'a pas d'effet en mode Sys : wiringPiSetupSys () ;

Exécuter un exécutable Qt en admin

L'exécution d'un exécutable dans Qt peut nécessiter des droits d'admin pour accéder au matériel. Il faut faire exécuter le programme dans un terminal :



Après avoir lancé le terminal en administrateur (*sudo*)



Cahier des charges

Commande manuelle de servo

Réalisez avec Qt un programme qui affiche une fenêtre comportant 3 boutons :

Au démarrage le servomoteur se positionne en milieu de course

Boutons :

Droite Gauche QUIT

Droite fait tourner le servo en sens antihoraire (CCW)

Gauche fait tourner le servo en sens horaire (CW)

QUIT ferme le programme

Bonus : Ajouter un slider ☺ pour commander le servomoteur et un afficheur pour indiquer la position.

On peut coupler l'action des boutons au slider.

Thermomètre numérique à affichage analogique

Réalisez avec Qt un programme qui affiche une fenêtre comportant 2 boutons :

Mesure température et **QUIT**

QUIT ferme le programme

Mesure température lance la mesure de température. La valeur s'affiche toutes les 5 secondes dans la fenêtre graphique.

De plus un servomoteur connecté au Raspberry Pi affiche la température sur une échelle graduée allant de 0 à 40 degrés Celsius.

