

Contrôle d'un moteur sur Beagle Bone Black en C

Cette partie a pour but de vous donner des indications quant à l'utilisation d'un moteur (brushless, servo..) avec la Beagle Bone. Le fonctionnement est simple, la Beagle est capable de générer des signaux PWM (pulse width modulation) pour diriger un moteur. Dans notre projet, nous avons travaillé avec un moteur brushless driver par un ESC (electro speed controller) dans lequel nous envoyons nos PWM. Pour ce faire nous avons utilisé en premier lieu la librairie Robot Control Library disponible par défaut sur la Beagle et disponible sur :

<https://beagleboard.org/static/librobotcontrol/index.html>

Première étape : Installation

Conseil : Utiliser l'image officielle de BeagleBone

Les images officielles de BeagleBoard.org sont livrées avec le paquet librobotcontrol préinstallé. Après avoir flashé une nouvelle image, reconfigurez simplement le package pour définir l'arborescence des périphériques :

```
sudo dpkg-reconfigure librobotcontrol
```

BeagleBoard.org héberge gracieusement le dernier paquet stable librobotcontrol dans ses dépôts. Sur un produit BeagleBoard, le package peut facilement être mis à jour :

```
sudo apt update && sudo apt upgrade librobotcontrol
```

Deuxième étape : La version (update disponible sur le site)

Vous pouvez vérifier quelle version du package est actuellement installée avec le programme rc_version. L'image stable 2018-06-17 est livrée avec 0.4.4

Troisième étape : Le project template

Nous vous suggérons fortement d'utiliser ou du moins d'étudier le rc_project_template disponible sur https://github.com/StrawsonDesign/librobotcontrol/tree/master/rc_project_template

En effet, l'intérêt de ce template est la création d'un Makefile qui va permettre la compilation et l'exécution de l'ensemble de vos programmes.

Vous pouvez également copier le projet directement de la manière suivante :

```
cp -r /usr/share/robotcontrol/rc_project_template ~/nouveau_nom_projet
cd nouveau_nom_projet
```

Ainsi, il suffira de modifier le Makefile de votre projet avec la commande nano pour lui donner la cible de compilation, en l'occurrence le nom choisi pour votre dossier contenant la template.

Enfin, parce qu'un code n'est jamais parfait du premier coup, vous pourrez facilement compiler et nettoyer votre dossier à l'aide des commandes respectives **make** et **make clean**.

Remarque : Je vous conseille néanmoins de coder sur un friendly IDE comme Vscode ou notepad afin de voir rapidement vos erreurs et prendre en compte l'ensemble du code. Attention ! Cela sera normal de ne pas pouvoir compiler car il vous manquera sûrement les bibliothèques incluses dans rc_lib_control.

Quatrième étape : La création de votre code

L'utilisation de la bibliothèque Robot Control Library permet une écriture rapide et une exécution simplifiée de certaine commande basique comme la commande de servo moteurs ou la mise en marche d'un brushless. Cependant, l'utilisation de module comme la fonction [rc_servo_send_esc_pulse_normalized](#) essentiel à la création d'un PWM, rend totalement opaque la création de celui-ci et donc la compréhension du code dans sa globalité.

Malgré tout, je vous encourage à vous exercer vous-même et à essayer de nouvelles choses !

Ressources : (exemple dans la librairie susceptible de vous aider)

Commande sudo devant si jamais problème de PRU.

- rc_test_escs : commande d'un moteur brushless équipé d'un ESC
- rc_test_servos : commande d'un servo moteur
- rc_test_encoders : récupérer la data d'un encodeur incrémental
- rc_version : donne la version de la librairie installée sur la carte