

Rapport de Projet : Contrôle de Servomoteurs sur Raspberry Pi

1. Introduction

Ce projet, réalisé dans le cadre d'un projet d'école, a pour objectif de contrôler plusieurs servomoteurs en simultané à l'aide d'une carte Raspberry Pi. Le C a été le langage de programmation choisi pour sa performance et son accès bas-niveau au matériel.

2. Matériel et Logiciel

2.1 Matériel

Le matériel nécessaire pour ce projet comprend :

- Une carte Raspberry Pi
- Plusieurs servomoteurs

2.2 Logiciel

Le logiciel de ce projet a été développé en C. Pour interagir avec les broches GPIO de la Raspberry Pi, la bibliothèque `libgpiod` a été utilisée. De plus, la bibliothèque `pthread` a été utilisée pour gérer des threads, permettant ainsi de contrôler plusieurs servomoteurs en parallèle.

3. Description détaillée de l'implémentation

Le projet est organisé en plusieurs fichiers source C, avec chaque fichier ayant une fonction spécifique.

Le fichier `servo.c` contient le code pour la gestion d'un servomoteur. Cette gestion est faite en envoyant des signaux PWM (Pulse Width Modulation) aux servomoteurs via les broches GPIO de la Raspberry Pi. Les signaux PWM sont des signaux carrés dont le rapport cyclique est modifié pour contrôler la position du servomoteur. La fréquence de ces signaux est de 50Hz pour les servomoteurs, ce qui correspond à un signal toutes les 20 millisecondes.

La génération de ces signaux est gérée dans un thread séparé pour chaque servomoteur, créé grâce à la bibliothèque `pthread`. La partie "haut" du signal est générée en mettant la broche GPIO à l'état haut et en attendant une durée proportionnelle à la largeur souhaitée pour le signal PWM. La partie "bas" du signal est ensuite générée en mettant la broche GPIO à l'état bas et en attendant le reste de la période de 20ms.

L'interaction avec les broches GPIO est facilitée par la bibliothèque `libgpiod`, qui offre une API simple pour contrôler les broches GPIO à partir d'un programme en C. En particulier, elle est utilisée pour mettre une broche à l'état haut ou bas, ce qui est essentiel pour la génération des signaux PWM.

Le fichier `main.c` est le point d'entrée du programme. Il affiche un prompt où l'utilisateur peut entrer des commandes pour contrôler les servomoteurs.

Le fichier `term_servo.c` fournit une surcouche du module de terminal pour la commande des servomoteurs. Il gère la création, la configuration et l'arrêt des servomoteurs.

Le fichier `term.c` est responsable de l'interprétation des commandes reçues par l'interface de commande. Il dispatche ces commandes aux différentes fonctionnalités implémentées dans `term_servo.c`.

4. Instructions pour l'utilisation

4.1 Branchement des servomoteurs

Chaque servomoteur doit être connecté à une broche GPIO de la Raspberry Pi. Le numéro de cette broche doit être noté car il sera nécessaire pour créer le servomoteur via l'interface de commande.

4.2 Compilation et exécution du programme

Pour compiler le code source, vous devez avoir installé le compilateur GCC pour la Raspberry Pi ainsi que les bibliothèques `libgpiod` et `pthread`. Ces dépendances peuvent être installées à partir des gestionnaires de paquets standard de votre système d'exploitation.

Le fichier `Makefile` inclus dans le projet facilite le processus de compilation. Vous pouvez compiler le code en utilisant la commande `make` dans le répertoire du projet :

```
$make
```

Cela génère un exécutable nommé `projet_trop_style.elf`.

Pour lancer l'exécution du programme, vous pouvez utiliser la commande suivante :

```
$make run
```

Cela compile le code (si nécessaire) et lance l'exécutable généré, vous pouvez aussi lancer le fichier généré avec la commande suivante :

```
$/projet_trop_style.elf
```

4.3 Utilisation de l'interface de commande

L'interface de commande est un prompt où l'utilisateur peut entrer des commandes pour contrôler les servomoteurs. Les commandes disponibles sont :

- `help` : Aide
- `servo-add <pin>` : Ajout d'un servomoteur
- `servo-set <id> <pwm>` : Configuration du rapport cyclique d'un servomoteur
- `servo-stop` : Arrêt de tous les servomoteurs
- `stop` : Arrêt du programme

Attention, il faut bien arrêter les servos moteurs avec la commande `servo-stop` avant de mettre fin au programme avec la commande `stop`.

5. Conclusion

Ce projet a permis de développer un programme en C pour contrôler plusieurs servomoteurs en simultanément à l'aide d'une carte Raspberry Pi. Cette implémentation offre une interface de commande facile à utiliser pour la gestion des servomoteurs, elle est aussi modulaire et la librairie de servo ou de terminal pourra être réutiliser dans un autre projet.

Malgré cela, il y a toujours des opportunités pour de futures améliorations et extensions, par exemple, en intégrant d'autres types de composants ou en élargissant le nombre de commandes disponibles dans l'interface de commande.