

Compte-rendu séance 11 Téo Baillot d'Estivaux :

Objectifs de la séance :

L'objectif de cette séance est de régler le problème des pertes de paquets lorsqu'on envoie des données de la carte émettrice à la carte réceptrice de façon à pouvoir contrôler les mouvements du bras robotique de façon beaucoup plus fluide. Actuellement, le taux de pertes de paquets est de 50%, on envoie de façon périodique 2 paquets puis les 2 paquets suivants sont perdus. Ceci nous permet de contrôler le bras robotique mais pour améliorer le contrôle du bras robotique il faudrait réduire ce taux de perte qui est tout de même très élevé.

Les différentes solutions essayées pour régler le problème :

Pour régler ce problème, j'ai essayé de nombreuses solutions :

- Tout d'abord, j'ai essayé d'augmenter le délai d'envoi entre les paquets car peut être que le problème était lié à un délai trop court qui pourrait surcharger ESP-NOW lors de la transmission. Cependant cette solution n'a absolument rien changé au problème.
- J'ai par la suite découvert qu'il était possible d'augmenter la puissance du module Wifi de l'ESP32 en écrivant la commande suivante dans le setup :
`esp_wifi_set_max_tx_power(78);`
J'ai donc essayé cette solution qui a l'air d'avoir augmenté la fréquence d'envoi des paquets mais qui a eu pour effet d'avoir des mouvements très fluides sur une certaines période car aucun paquet n'était perdu puis plus aucun mouvement pendant une autre période car tous les paquets étaient perdus.
- Puis, j'ai appris que ESP-NOW fonctionne mieux sans certaines fonctionnalités Wifi actives comme la connexion à un routeur dont on ne se sert pas. J'ai donc essayé de désactiver ces fonctionnalités en écrivant la commande suivante dans le setup :
`WiFi.setSleep(false);`
Toutefois, ça n'a rien changé au problème, ça l'a même empiré car je n'arrivais plus à lire les données de mon accéléromètre.
- Après, je me suis dit que peut-être que le problème venait de la surcharge de la communication I2C, j'ai donc ajouté un léger délai entre le moment où on lit les données et le moment où on les envoie dans la boucle principale mais ça n'a absolument rien changé.
- J'ai ensuite tester si la taille de la structure correspond à la taille de 2 float (représentant les données de l'accéléromètre) et un int (représentant les données du capteur flex). Si ce n'est pas le cas ça pourrait expliquer la perte de paquets mais après test on a bien la correspondance donc le problème ne vient pas de là.
- Puis, j'ai modifié ma fonction OnDataSent() pour faire en sorte d'essayer de renvoyer le paquet une seconde fois à chaque fois que je le perds. La fonction est donc devenue :
`void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
 if (status == ESP_NOW_SEND_SUCCESS) {`

```
Serial.println("Données envoyées avec succès");  
resendRequired = false;  
} else {  
    Serial.println("Échec de l'envoi, nouvelle tentative programmée...");  
    resendRequired = true; // Demande un réessai  
}  
}
```

J'ai évidemment modifié d'autres petites parties du code afin de pouvoir appeler et gérer correctement cette nouvelle fonction.

Malheureusement, je n'ai encore une fois vu aucune différence.

Lors de cette séance, j'ai donc consacré tout mon temps à effectuer des recherches pour essayer de trouver des solutions pour résoudre le problème de pertes de paquets lors de la transmission des données (ce qui m'a pris beaucoup de temps) et à tester toutes les solutions que j'ai trouvées mais je n'ai malheureusement pas eu de résultats concluants.

[Objectifs de la prochaine séance :](#)

Pour la prochaine séance, je vais essayer de modifier la puissance d'émission Wifi pour trouver la puissance idéale dans notre cas car après réflexion comme la carte réceptrice est très proche de la carte émettrice, une puissance trop élevée pourrait avoir pour conséquence de saturer le signal ce qui expliquerait pourquoi pendant une certaine période on ne reçoit plus aucuns paquets. Je vais également continuer à chercher d'autres solutions chez moi et essayer de modifier le code existant afin de pouvoir faire des tests de nouveaux codes dès le début de la prochaine séance. Je passerais donc également la prochaine séance à essayer de régler ce problème qui est crucial pour améliorer les performances de notre bras robotique.