|  |
| --- |
| CPNV |
| Troubleshooting latence  P1704 Manette |
| Compte rendu |
|  |
| **MAILLARD Joan** |
| **04/04/2019** |

|  |
| --- |
| [Tapez le résumé du document ici. Il s’agit généralement d’une courte synthèse du document. Tapez le résumé du document ici. Il s’agit généralement d’une courte synthèse du document.] |

Table des matières

[1 But 2](#_Toc5281766)

[2 Conclusion 2](#_Toc5281767)

[3 Documentation 2](#_Toc5281768)

# But

Déterminer la provenance de la latence constatée dans la communication entre la pression d’un bouton et l’arrivée de l’information au récepteur

# Conclusion

Le correctif apporté au problème est le suivant : au lieu d’un envoi répété des trames de données du data gatherer vers l’ESP32, nous les avons synchronisés pour que le data gatherer n’envoie qu’une trame à la fois, sur demande. Le problème de latence est résolu.

# Documentation

3 étapes de communication à analyser :

* Data gatherer -> émetteur manette
* Emetteur manette -> récepteur
* Récepteur -> objet contrôlé (dans ce cas, un Arduino Mega qui allume une LED)

**Selon le rapport de mesure « Mesure latence » :**

Le temps entre la pression du bouton en question et le moment où l’information est livrée au récepteur par le data gatherer a été testé : 3ms

Le temps entre l’écriture de l’information par le récepteur et l’allumage de la LED correspondante a été testé : 3ms

Le temps entre la lecture de l’information par l’émetteur et l’écriture par le récepteur : 30ms

Le temps entre l’écriture de l’information par le data gatherer à l’émetteur manette et l’écriture de l’information par le récepteur pour l’objet contrôlé a été testé : 264ms

Le point problématique se situe donc entre l’entrée dans l’ESP32 émetteur et le récepteur

Une relecture de la documentation de l’ESP32 a déterminé que la taille du buffer matériel UART de l’ESP32 est de 64 bytes, soit 8 trames de données (8 bytes par trame, voir le « Protocole de communication »). En outre, selon le « Test de base Bluetooth », on a 30ms de délai par écriture Bluetooth. Il s’avère donc que le problème se trouve dans le buffering du data provenant du data gatherer : les entrées remplissent le buffer de l’ESP32, qui les traite une par une. Le buffer a une taille de 64 bytes, et une trame, 8 bytes. On a donc 8 trames en buffer, ce qui provoque un délai de 8\*30ms soit à peu près 240ms, ce qui est la majorité de notre délai.

*KAHRIMANOVIC Adel, MAILLARD Joan, le 04.04.2019, Yverdon-les-bains*