1. On remarque que les tâches s’exécutent par ordre de priorité et non pas par leur ordre de création.
2. Le problème est que, malgré que la tâche 1 soit plus prioritaire que la tâche 2, à cause des tics d’attentes imposés après l’écriture de chaque caractère, les tâches s’exécutent en désordre.

Comme le buffer d’écriture est partagé entre les 2 tâches, il faut donc ajouter un mutex pour protéger l’écriture dans ce buffer. Il faut que la tâche 1 finisse d’ajouter tous les caractères dans le buffer ensuite la tache 2 commence.

1. La taille minimum des files sont :

* Pour la file entre controller\_to\_tank\_A : 24, car le temps minimum entre chaque commande est de 5 ticks. Le temps maximum d’exécution d’une commande est de 33 ticks. Par conséquent, dans le pire des cas, au moment de la 1er commande et entre la 7eme-8eme commande envoyé par le contrôleur, un message va être lu par le tank A ce qui résulte a avoir 15 messages dans la file. Ensuite 3 autres message seront envoyés ce qui fera un totale de 24.
* Pour les files controller\_to\_tank\_mixer et controller\_to\_tank\_B, une taille de 1 est suffisante.

1. On peut le faire en initialisant un sémaphore binaire à 1. En initialisant le sémaphore binaire a 1 on peut laisser passer une tache pour la ressource et puis la bloquer une fois utiliser.

Fdjkghsgh fjgf

samerajjoub@hotmail.com