**École Polytechnique de Montréal**

**Département de génie informatique et logiciel**

**Cours INF3610**

**SYSTÈMES EMBARQUÉS**

**Travail pratique 3**

**Noms:**

**Maxime Pérusse (1685946)**

**Alexis St-Onge (1688297)**

**Gabriel Bilodeau-Périard (1681165)**

**Date:**

**6 Novembre 2015**

**Question 1 - Quel est l’intérêt d’utiliser les modules writer et reader au lieu d’interfacer directement le module bubble au module dataRam ?**

L'intérêt est multiple. Premièrement, ces modules permettent de simplifier le fonctionnement du logiciel en découplant les différentes entités du code. Les modules bubble et dataRam n'ont aucune idée du fonctionnement des modules reader et writer, ils connaissent seulement les interfaces. Ces interfaces permettent donc de cacher l'implémentation à l'utilisateur, celui-ci sait seulement qu'il a accès à certaines fonctions ce qui permet donc de simplifier sa compréhension.

Puisque les modules bubble et dataRam se retrouvent complètement séparés, une modification à l'un ne nécessite pas de modifier l'autre. Il suffit de suivre le standard définis par l'interface des modules lecteur et écrivain puis le fonctionnement sera le même. Ceci rend donc la modification des modules bubble et dataRam plus simple.

De plus, en considérant la question sur un point de vue matériel, l'idée du « reader/writer » peut facilement être vue comme un médium, gestionnaire des accès mémoire entre différents périphériques. Ainsi, plusieurs autres modules auraient pu s’ajouter au « bubble sort » pour communiquer avec la mémoire.

**Question 2 - Aurait-il été possible d’implémenter la communication handshake du channel Reader dans le modèle AT à l’aide d’une sc\_method plutôt que d’un sc\_thread ? Expliquez votre réponse.**

Un SC\_THREAD peut être suspendu et réactivé avec un wait. Il ne peut pas être terminé.

Un SC\_METHOD se termine toujours, il n'y a pas de wait, de boucle infinie ou de suspension d'exécution. Il s'agit d'une routine qui s'exécute en entier lorsqu'un changement sur la liste de sensibilité survient.

Dans le cas du channel Reader, il est possible d’utiliser une SC\_METHOD car il suffit de l’exécuter en activant le requestReader lorsqu’il y a des données à communiquer. Il ne serait alors pas nécessaire de boucler à l’infini sur une request à tous les fronts montant de l'horloge comme nous le faisons présentement avec le SC\_THREAD.