
Rapport d'exécutifs temps-réel - Transfert de messages -

ÉCOLE POLYTECH DE NANTES
DÉPARTEMENT ÉLECTRONIQUE ET TECHNOLOGIE DU NUMÉRIQUE

Rédigé par
Mathis BRIARD
Victor DUFRENE

Professeur encadrant :
Olivier PASQUIER

Table des matières

1	Introduction	5
1.1	Contextualisation	5
1.2	Objectifs	5
2	Situation	6
3	Solution classique	7
3.1	Validation du comportement	8
3.2	Asservissement de la boîte aux lettres	9
3.3	Différences liées au type de sémaphore	9
4	Variante	9
5	Conclusion	9

Table des figures

1	Sortie sur la console	6
2	Sortie sur la console	7
3	Sortie sur la console	8
4	Sortie sur la console	9
5	Sortie sur la console	9

Liste des tableaux

1	Attribution des priorités aux différentes tâches	8
---	--	---

Résumé

L'exécutif temps-réel de systèmes électroniques est une discipline essentielle lorsqu'il s'agit, dans le cadre d'une future carrière d'ingénieur, de réaliser un produit répondant à des contraintes de temps. Et ce domaine est d'autant plus indispensable dans un monde où les systèmes numériques sont de plus en plus complexes et rapides, demandant de réaliser des tâches dans une fenêtre de temps bien déterminé. Ce rapport s'inscrit dans la présentation d'une application affichant l'heure sur un écran.

1 Introduction

1.1 Contextualisation

La formation ETN (Électronique et technologies numériques) offerte par l'école polytechnique de l'Université de Nantes propose d'aborder diverses branches de l'électronique, du traitement du signal au systèmes à microprocesseur en passant par l'électronique analogique des hautes-fréquences. Cet ensemble de domaines techniques nécessite des compétences en matière de méthodologie de conception. Ce rapport s'inscrit dans la conception d'un appareil de marquage routier avec la méthode MCSE. La méthode MCSE (Méthode de conception des systèmes électroniques), née à Ireste par l'impulsion de Jean-Paul Calvez, cette méthode a été implantée au sein d'un outil nommée CoFluent rachetée par Intel® depuis 2011. Cette méthode fait désormais partie de la culture de la formation et constitue l'outil de conception premier de l'ingénieur ETN.

Ce rapport se décompose en diverses parties. Il s'agira dans un premier temps de rappeler le cahier des charges de la conception de cet appareil de marquage routier. Dans un second temps la partie spécification sera traitée et pour finir il s'agira de parler de la conception.

1.2 Objectifs

Ce rapport vise à retranscrire les résultats du travail pratique réalisé sur les files (transferts) de messages. Les objectifs des manipulations étaient d'utiliser le mécanisme de boîte aux lettres puis d'estimer l'utilité de l'exclusion mutuelle.

2 Situation

La situation que nous visons à implémenter est une file de message qui permet la communication inter-processus. L'application considérée est celle d'un système qui affiche l'heure dont la structure fonctionnelle est donnée à la figure 1.

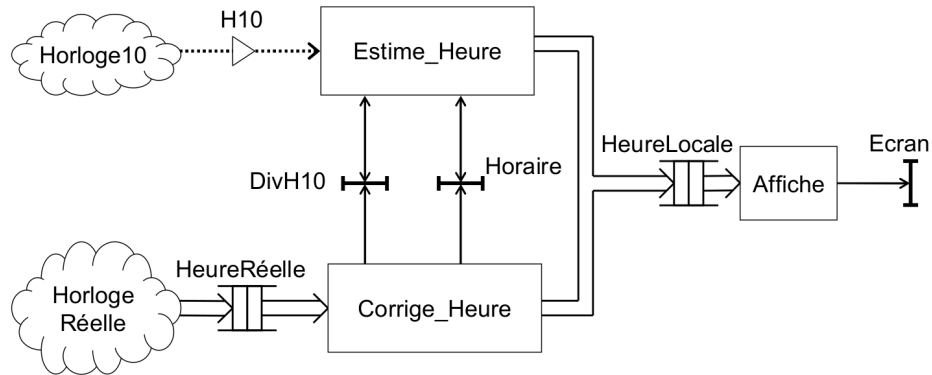


Figure 1: Sortie sur la console

L'application consiste à afficher sur Ecran, l'heure (heure, minute, seconde). L'heure peut être obtenue par estimation à partir d'une horloge à 10Hz (H10) et pour éviter les dérives, l'heure réelle est reçue par l'intermédiaire de HeureRéelle (heure, minute, seconde). Pour information, afin de limiter la longueur des observations, la base de temps de l'environnement est beaucoup plus rapide que la seconde.

Fonction Estime_Heure

Action activée sur H10, événement périodique apparaissant 10 fois par seconde. Elle incrémente la variable DivH10 si elle est inférieure à 9, sinon elle met la variable à 0 et la variable Horaire est incrémentée d'une seconde. Un message est transmis dans HeureLocale. Le message indique que la valeur transmise est une valeur estimée et contient la nouvelle valeur de Horaire.

Fonction Corrige_Heure

Action activée sur HeureRéelle. Elle copie la valeur reçue dans Horaire, met la variable DivH10 à 0 et transmet un message dans HeureLocale. Le message indique que la valeur transmise est une valeur réelle et contient la nouvelle valeur de Horaire.

Fonction Affiche

Cette fonction copie sur Ecran l'heure reçue par l'intermédiaire de HeureLocale en indiquant s'il s'agit de l'heure estimée ou de l'heure réelle (c'est la seule tâche ou vous pouvez faire un printf).

Le réseau de Pétri qui décrit le comportement des fonctions Estime_Heure et Corrige_Heure est donné à la figure 2 ci-dessous.

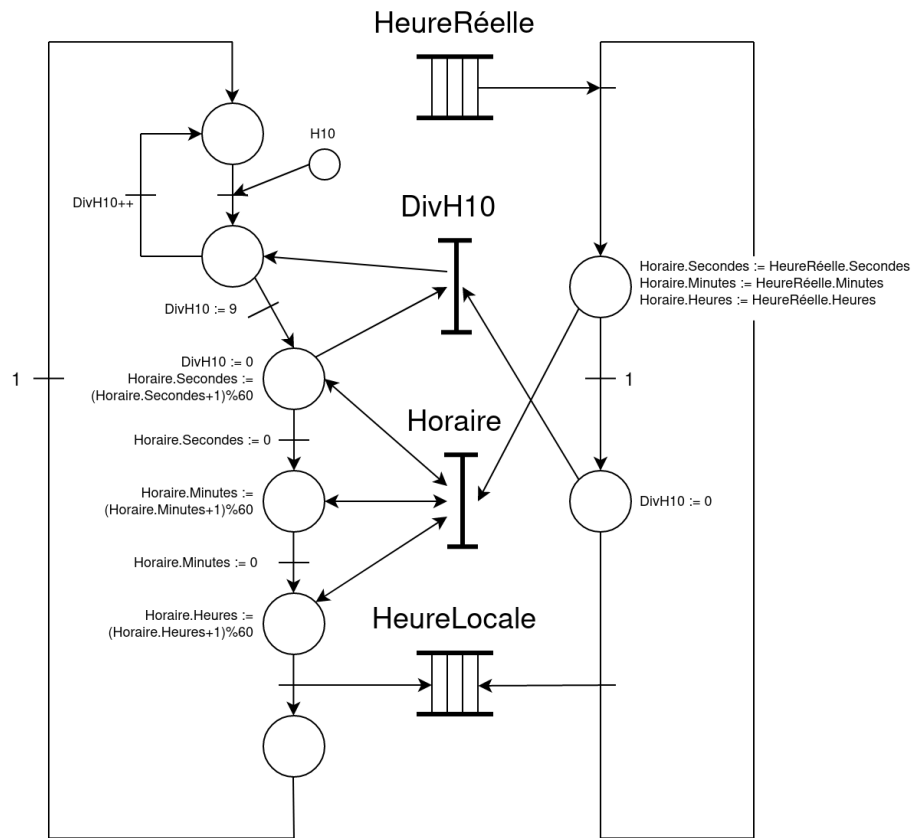


Figure 2: Sortie sur la console

Dans ce réseau on est en présence de deux variables partagées **Div10** et **Horaire**. Une question logique à ce poser est : doit on mettre des sémaphores pour éviter des exclusions mutuelles ?

La réponse est non car l'environnement est synchrone. Les tâches **Estime_Heure** et **Corrige_Heure** sont prêtes en même temps puis s'activent en même temps. Cela implique une gestion simple des priorités. C'est pourquoi il est possible d'éviter la mise en place de sémaphore.

3 Solution classique

Cette partie vise à mettre en place la solution décrite précédemment. Pour l'implantation sur VxWorks, il est nécessaire d'utiliser les éléments suivants :

- une tâche **Estime_Heure**;
- une tâche **Estime_Heure**;
- une tâche **Estime_Heure**;
- un sémaphore **H10**;
- une file de message **FMHeureLocale**;
- une file de message **FMHeureReelle**;
- une variable partagée **DivH10**;
- une variable partagée **horaire**.

On précise que la variable `horaire` est d'un type prédéfini `type_heure` qui contient les secondes, les minutes et les heures dans une structure.

Pour le bon fonctionnement du programme, il est nécessaire d'attribuer des priorités aux différentes tâches. Dans l'environnement VxWorks, la tâche de priorité la plus élevée est celle qui dispose du numéro de priorité le plus faible. La tâche de priorité 0 est la plus prioritaire et la tâche de fond est la tâche de priorité 255. On évite d'attribuer les priorités comprises entre 0 et 10, réservée à l'environnement VxWorks. Les priorités suivantes sont affectées aux tâches :

Tâche	Priorité
Estime_Heure	13
Corrige_Heure	14
Affiche	15

Table 1: Attribution des priorités aux différentes tâches

3.1 Validation du comportement

La figure 5 montre la sortie de la console pour une exécution classique du programme.

```

value = 0 = 0x0
-> start
Demarrage de l'estimation de l'horloge
Demarrage de la correction de l'horloge
Demarrage de l'affichage
Fin de start
value = 0 = 0x0
-> Origine du message : Corrige
l'Heure est: 0 Heure, 1 min, 2 sec
Origine du message : Corrige
l'Heure est: 0 Heure, 2 min, 3 sec
Origine du message : Corrige
l'Heure est: 0 Heure, 2 min, 4 sec
Origine du message : Corrige
l'Heure est: 0 Heure, 2 min, 5 sec
Origine du message : Estime
l'Heure est: 0 Heure, 2 min, 6 sec
Origine du message : Corrige
l'Heure est: 1 Heure, 2 min, 3 sec
Origine du message : Corrige
l'Heure est: 1 Heure, 3 min, 4 sec
Origine du message : Corrige
l'Heure est: 1 Heure, 3 min, 5 sec
Origine du message : Corrige
l'Heure est: 1 Heure, 3 min, 6 sec
Origine du message : Estime
l'Heure est: 1 Heure, 3 min, 7 sec

```

Figure 3: Sortie sur la console

Une fois l'initialisation des tâches effectués dans le `start`, nous observons un affichage de l'heure provenant des deux tâches. On note que l'heure affiché provient beaucoup plus souvent de `Corrige_Heure` que de `Estime_Heure`.

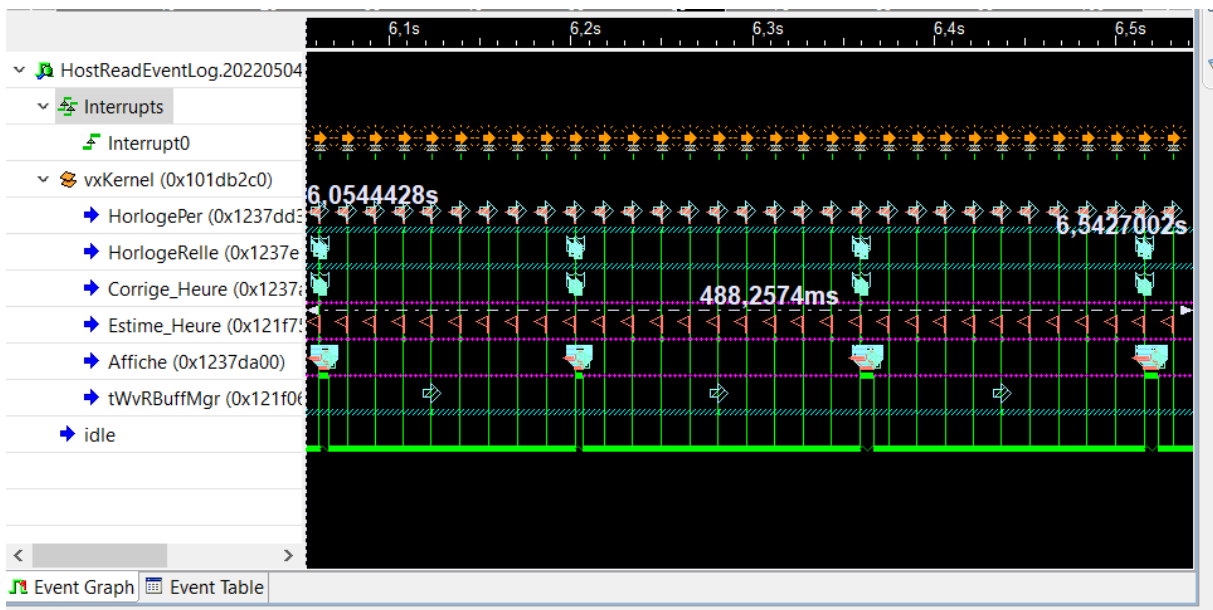


Figure 4: Sortie sur la console

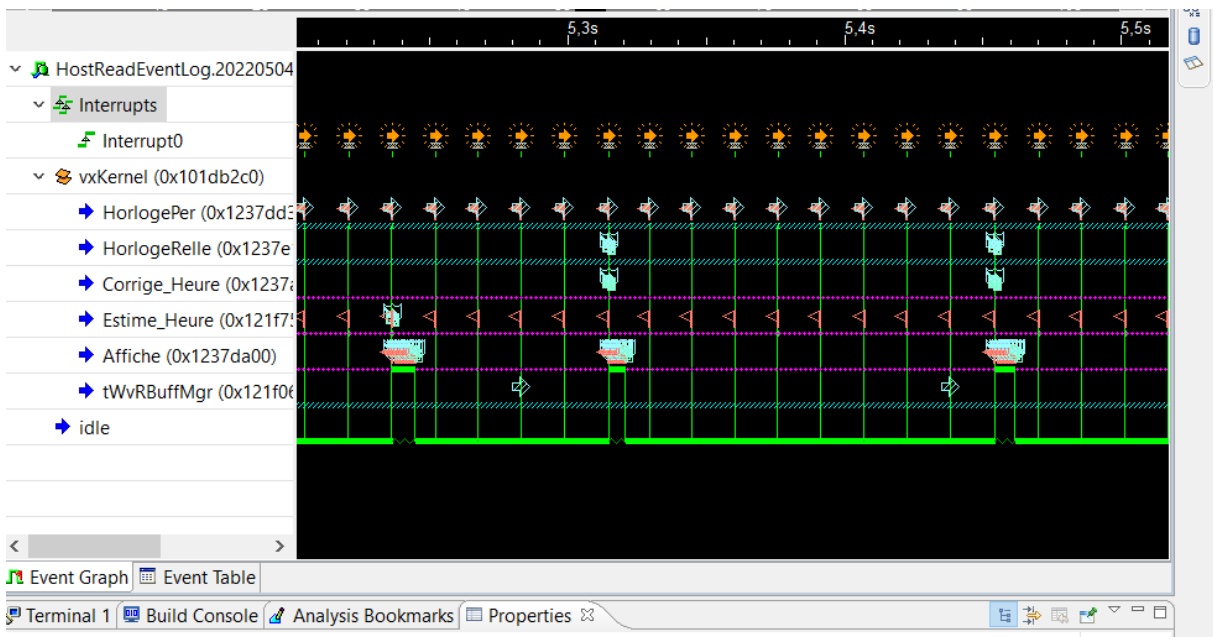


Figure 5: Sortie sur la console

Le code commenté de la solution à un utilisateur pour les priorités ci-dessus est disponible en Annexe 1. La simulation de la situation est lancée et analysée ci-dessous.

3.2 Asservissement de la boîte aux lettres

3.3 Différences liées au type de sémaphore

4 Variante

5 Conclusion