

# MI43 – Final

Documents autorisés : notes de cours et documentation constructeurs uniquement

Durée 2 h

Tout code non expliqué ne sera pas considéré valable

## **1. Mise ne place d'une interruption système (15pts)**

On souhaite disposer d'une interruption dite « système » permettant d'interrompre les applications toutes les 10 millisecondes pour effectuer des mises à jour de variables système en utilisant le TC0 (Timer counter 0) de l'AT91.

### **1.1. Initialisation du matériel**

1. Donner le code d'initialisation de contrôleur d'interruption AIC pour un fonctionnement en mode autovectorisé, toutes les interruptions masquées. On effectuera également une sorte de « reset » logiciel du composant en effectuant 8 acquittements successifs pour le cas où des interruptions auraient été en attente.
2. Donner le code d'une fonction permettant de configurer le timer TC0 pour qu'une interruption soit générée toutes les 10 millisecondes. Cette fonction enregistrera également sa routine de traitement d'interruption auprès de l'AIC avec une priorité fixée à 1.

### **1.2. Routine assembleur**

La routine de traitement des interruptions sera écrite en assembleur et effectuera :

décrémentation d'une variable globale que l'on nommera *qt* si celle-ci n'est pas à 0

l'appel d'une fonction nommée *system* si cette dernière est à 0 et qu'une variable globale nommée *sc* est égale à 0 également.

## **2. Sauvegarde de contexte (5)**

Il s'agit ici de compléter la routine précédente en vue d'effectuer un changement de tâche. Pour rendre la question indépendante on ne s'occupe ici que de la sauvegarde du contexte de la tâche.

Toutes les tâches sont des tâches en mode svc

Toutes les tâches disposent d'une pile propre

L'exécutif dispose également de sa propre pile.

Donner le code assembleur d'une routine d'interruption qui réalise la sauvegarde complète de tous les registres de la tâche dans sa propre pile. On pourra utiliser ici la pile du système qui est accessible via la variable globale nommée *ssp* pour effectuer des sauvegardes temporaires si nécessaire ou encore la pile *irq*.

Le bout de code se terminera par un appel à la fonction nommée *system* qui prendra cette fois-ci en paramètre l'adresse de la pile de la tâche où est sauvegardé tout le contexte. Lors de cet appel c'est la pile du système qui devra être active.