



Systèmes Embarqués 1 & 2
Objectifs du travail écrit n° 3

§Conditions d'examen :

Le seul document à disposition est le « 05_ARM_Instruction_Set_Summary.pdf »
(jeu d'instructions du processeur ARM)

Les étudiant-e-s devront être capables :

Interruptions

- d'expliquer les différentes phases de traitement d'une interruption (séquence)
- de classer les types d'interruptions et d'exceptions
- d'expliquer la fonctionnalité de la table des vecteurs d'interruptions sur les processeurs ARM, ainsi que l'implémentation spécifique du processeur AM335x
- d'expliquer comment l'on peut passer du mode superviseur au mode utilisateur et vice versa et de concevoir le code assembleur permettant ces passages
- de déterminer le mode de fonctionnement du à partir de son registre de statut
- de décrire la fonction des pointeurs de piles des différents modes du processeur ARM et de concevoir le code permettant de les initialiser
- de décrire le concept de commutation de contexte, de latence et de gigue
- de décrire le système d'interruption des processeurs ARM et du processeur AM335x (μ P (core), INTC, GPIO, etc.)
- d'expliquer les mécanismes d'activation et de désactivation des interruptions matérielles (μ P (core), INTC, GPIO, etc.)
- de décrire le principe de niveaux de priorité (μ P, INTC et GPIO)
- de décrire la procédure de reconnaissance d'interruption multiple (scrutation, priorisée, vectorisée)
- de décrire et de concevoir un gestionnaire d'interruption (du μ P à l'application)
- de programmer une application utilisant une interruption
- de décrire et de concevoir des opérations atomiques



Systèmes Embarqués 1 & 2
Objectifs du travail écrit n° 3

Systèmes d'exploitation

- de citer quelques techniques et méthodes de développement pour des systèmes embarqués
- de décrire les différents types de systèmes multitâches (systèmes d'exploitation)
- de décrire les différentes composantes d'un noyau
- de décrire les ressources globales, partagées et privées d'un thread
- de décrire les états principaux d'un thread
- de décrire les éléments et structures nécessaires à la gestion d'un thread
- de décrire la commutation de contexte entre deux threads
- de concevoir un programme de transfert de contexte entre deux threads
- de concevoir l'initialisation du contexte d'un thread
- de concevoir un ordonnanceur (scheduler) élémentaire
- de concevoir un mécanisme de synchronisation simple (sémaphore, message queue)

Travail pratique PI.2

- de programmer une sous-routine en assembleur ARM respectant les conventions pour le passage des arguments
- de coder dans les règles de l'art l'appel de routines développées en C depuis un code en assembleur ARM
- de simuler des interruptions logicielles et des exceptions

Travail pratique PI.3

- de décrire le fonctionnement des DMTimers et du GPIO en mode interruptif
- de décrire la détection du sens de rotation de l'encodeur rotatif en mode interruptif
- de décrire le concept de « listener » et le mettre en œuvre (réalisation)