

TP n°1 module 3 **IUP AISEM 2^{ème} année**

Objectifs du TP :

- Se familiariser avec l'outil de développement et de simulation *AVR Studio4* du microcontrôleur ATMEL AT90S8535 (édition de fichiers et assemblage).
- Débugger et tester un programme écrit en langage assembleur.
- Voir l'évolution des ressources internes du microcontrôleur étudié et les justifier (compteur de programme, pointeur de pile, RAM, registres d'E/S ...).
- Evaluer un temps d'exécution.
- Définir le rôle du pointeur de pile.

Matériel utilisé :

- Un PC équipé du logiciel de développement AVR Studio4 + 1 liaison série
- Un émulateur Atmel ICE200 et sa documentation.

Expérimentation n°1:

Le TP s'effectue en 2 temps :

- une phase de simulation et de vérification du fonctionnement.
- Une phase d'utilisation de l'émulateur ICE200.

- 1) Créer un projet appelé « recopie » (suivre les directives données par l'enseignant)
- 2) Charger le programme « recopie.asm ».
- 3) En ouvrant le fichier « 8535def.inc », justifier la raison de sa présence dans le programme. Ce fichier est-il indispensable ?
- 4) En vous aidant de la documentation technique du circuit AT90S8535, commentez le programme fourni.

Simulation :

- 1) Assembler le programme (directive « build »)
- 2) Observer les résultats de l'assemblage (fenêtre « output »)
- 3) Se mettre en mode « debug » et tester le programme en simulation. (on testera les différentes commandes du simulateur). On prendra soin d'observer comment évoluent les différentes ressources du microcontrôleur. On justifiera la présence d'une instruction à l'adresse \$0.
- 4) Quel est le rôle de ce programme ? Dans ce contexte est-il indispensable d'initialiser le pointeur de pile ?

Mise en œuvre de l'émulateur ICE200 :

- 1) Sélectionner l'émulateur ICE200 et le μ Contrôleur AT90S8535 (Debug => Select platform and Device ...).
- 2) Exécuter le programme sur l'émulateur (Start Debug => Run).
- 3) Vérifier le fonctionnement sur la maquette.
- 4) Réécrire le programme sans faire appel au fichier « 8535def.inc »

5) Assembler et tester.

Expérimentation n°2:

- 1) Créer un projet appelé « compteur ».
- 2) Charger le programme « compteur.asm »
- 3) Exécuter le programme en mode pas à pas. Quelle est l'utilité du sous-programme « delai » ?
- 4) En s'aidant de la documentation des instructions (programme avrasm.htm), **évaluer** l'ordre de grandeur du temps d'exécution de ce sous-programme.
- 5) Tester le programme sur la maquette. Mesurer le temps d'attente correspondant à la boucle « délai ».

Câblage des boutons poussoirs et des leds sur la maquette :

