

# Travaux Pratiques

## DSP – SHARC (L. Kessal – ENSEA)

### 1) Tutoriel - Environnement de développement VisualDSP :

- ☐ Simulation (Exemple : Filtre RIF en assembleur, sans E/S)
  - TP1 : Dossier de travail **rif\_ASM**
- ☐ Simulation (Exemple : Filtre RII en C, sans E/S)
  - TP2 : Dossier de travail **rri\_C**

1 séance 1/2

### 2) Travaux pratiques (Kit de développement : EZ-KIT)

- ☐ Gestion des interruptions + E/S
  - ☐ TP3 : Filtre RIF en assembleur (temps réel) ➔ Dossier de travail **rifes\_ASM**
  - ☐ TP4 : Filtre RIF en C (temps réel) ➔ Dossier de travail : **rifes\_C**

1 séance 1/2

### 3) Travail à réaliser (avec compte rendu)

- ☐ Mixage du C/Assembleur : Filtre récursif en temps réel
  - Dossier de travail : **Der1D**

3 séances

❑ *L'objectif des deux tutoriaux est de bien maîtriser l'environnement de développement des processeurs SHARC (VisualDSP++ d'Analog Devices. Le premier tutorial, écrit exclusivement en langage assembleur, simule un filtre non récursif (RIF pour Réponse Impulsionnelle Finie) et le second, écrit en langage C, simule un filtre récursif (RII pour Réponse Impulsionnelle Infinie). Ce dernier exemple vous fera découvrir des fonctionnalités supplémentaires très importantes de l'environnement VisualDSP++ (mixage du C & ASM, Profiling, ...)*

❑ *Les deux TP suivants abordent les E/S et la gestion des interruptions en langage C et en assembleur par le Cœur du processeur SHARC.*

❑ *Dans le dernier thème, les différents mécanismes rencontrés vous permettront d'implanter un filtre récursif temps réel en mixant du C et de l'assembleur.*

🚧 *Pour chaque projet, les signaux à traiter et les coefficients du filtre (fichiers \*.dat) doivent être placés dans le même dossier que le projet.*

🚧 *Voici l'arborescence à créer et à rendre à la fin des 4 séances (sous forme d'une archive).*

#### **TP\_SHARC**

```
|  
|__LDF      (ADSP-21262_ASM.LDF, ADSP-21262.LDF)  
|  
|__rif_ASM  
|  
|__rii_C  
|  
|__rifes_ASM  
|  
|__Der1D
```