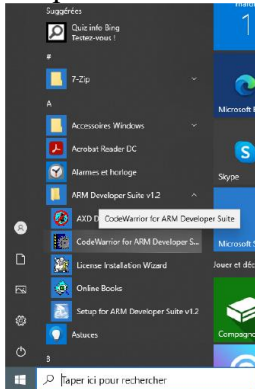


Dans un premier temps, chaque étudiant (ou binôme) devra créer un répertoire de travail portant son nom. C'est uniquement à l'intérieur de ce répertoire que seront sauvegardés vos programmes (projets) des différents TDs et TP de programmation ARM7. Au sein du répertoire portant votre nom, créez un sous-répertoire appelé TD1 dans lequel vous allez travailler aujourd'hui.

Donc, sur le bureau, le chemin sera : **L2\ARM7\votre_nom\TD1**
L2\ARM7\votre_nom\TD2..... L2\ARM7\votre_nom\TP1.....Etc.

Pour lancer le logiciel CodeWarrior, il y a deux manières de le faire :

- (1) Cliquez sur démarrer tout en bas à gauche de l'écran et choisir ARM Developer Suite



- (2) Cliquez sur CodeWarrior qui se trouve sur le bureau.

Exercice I (Programmation de base et prise en main de la carte Evaluator-7T):

Dans cet exercice, vous allez apprendre à compiler et à exécuter un programme en assembleur.

Nous allons saisir un premier programme et créer un projet.

Voici le programme :

Etiquettes

```

start
ENTRY                                ; marque la première instruction
MOV r0, #15                          ; initialise les valeurs
MOV r1, #20
BL firstfunc                         ; Appel sous-programme
MOV r0, #0x18 ;
LDR r1, =0x20026 ;
SWI 0x123456                         ; Termine le programme

firstfunc
ADD r0, r0, r1 ; r0 = r0 + r1
MOV pc, lr                          ; retour du sous-programme
END                                  ; Marque la fin du fichier
  
```

La directive « AREA » est utilisée par l'éditeur de lien. Une application complète peut être composée de plusieurs AREAs (modules). Dans cet exemple un seul champ marqué en lecture seule.

La directive « ENTRY » permet de marquer la première instruction. Dans une application, il n'existe qu'une seule entrée « ENTRY ».

Forme générale d'une instruction assembleur :

label <espace blanc> instruction <espace blanc> ; commentaire

Le label (ou étiquette) peut être n'importe quel nom (tata, titi, toto, ...). Dans l'exemple ci-dessus, « start » est une étiquette qui aurait pu prendre un autre nom « essai, debut, ou autre ».

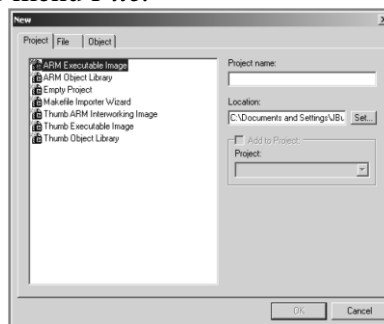
Attention : Les instructions doivent être alignées selon la verticale (colonne) comme montré dans l'exemple ci-dessus.

(a) Création du projet

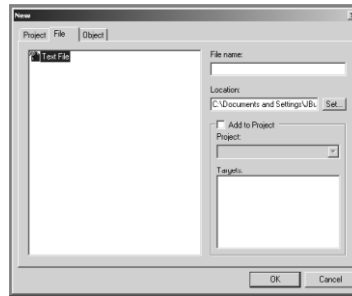
Chacun de vos programmes ARM7 est un projet. Chaque projet peut contenir plusieurs fichiers (programmes assembleur, langage C). Au minimum, un projet doit lui être associé un fichier nompgme.s ou bien nompgme.c. Les noms de fichiers *.s sont pour l'assembleur et les fichiers *.c pour le langage C. Donc, vérifier à chaque fois l'extension du nom de votre fichier.

Dans un premier temps, nous créons le projet puis nous lui associerons un fichier « .s » ou « .c ».

- Lancer CodeWarrior
- Sélectionner *New* du menu *File*.



- Sélectionner l'icône *Projet*
- Cliquer sur *ARM Executable Image* et entrer *TD1* dans le champ *name*. Votre projet s'appelle donc **TD1.mcp**. Tous les projets portent l'extension « mcp ».
- Vérifier le chemin du répertoire (cliquez sur *set* dans la rubrique *location*) de travail et cliquez sur *enregistrer*. Le chemin doit être celui qui correspond à « L2\ARM7\ votre_nom \TD1 » .
- Cliquez sur *Ok*
- Créer le fichier source (le fichier qui va contenir votre code en assembleur ou en c) en sélectionnant dans le menu *File*, *New*
- Cliquer sur l'icône *File*



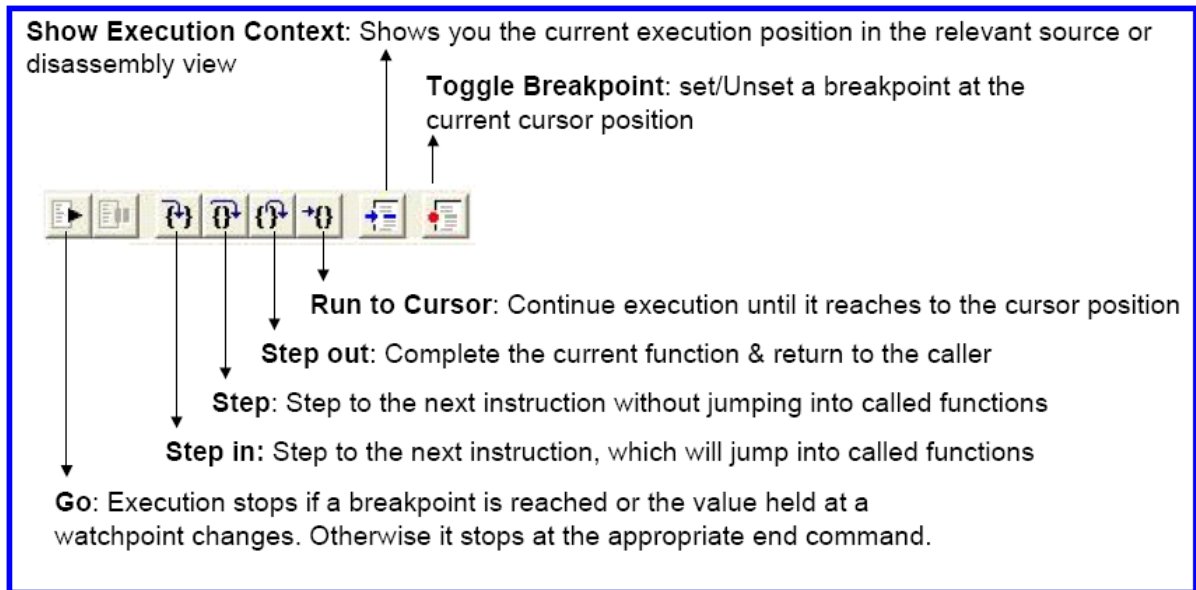
- Cliquer sur *Text File* et entrer *exemple1.s* comme nom dans *File name*
- Cliquer sur *Add to Project* et valider (cocher) toutes les options proposées.
- Cliquer sur *ok*
- Entrer (tapez au clavier) le programme ci-dessus dans la nouvelle fenêtre qui vient de s'ouvrir.
- Une fois terminé, sauvegarder ce fichier en cliquant sur *save* du menu *File*
- Sélectionner *Make* du menu *Project*. Ceci permet la génération d'une image binaire du code (exécutable) que vous venez d'entrer. Le fichier exécutable s'appelle ***exemple1.axf***. L'exécutable a donc l'extension « *.axf* » pour ARM7.

(b) Exécution du programme

- Lancer le Debugger ARM (il s'appelle **AXD**). Pour ceci, suivez la même procédure que pour lancer CodeWarrior sauf vous sélectionner AXD.
- Sélectionner *Load image* du menu *File* dans **AXD**.
- Dans la fenêtre de dialogue, chercher le fichier que vous venez de créer en spécifiant le chemin d'accès. Il s'agit du fichier *exemple1.axf* et cliquer sur *open*
- Une fenêtre s'ouvre, elle contient le source que vous venez de taper.
- Nous allons ajouter *un point d'arrêt* dans le programme pour examiner les contenus des registres, de la mémoire et des flags. Cliquer 2 fois sur la ligne

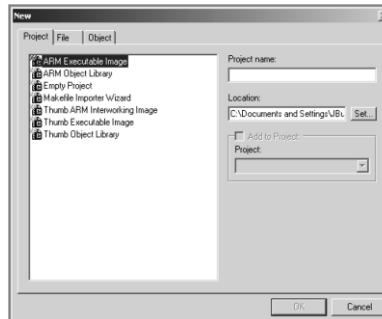
ADD r0, r0, r1 ; r0 = r0 + r1

- Démarrer le programme en cliquant sur *Go* du menu *Execute*. Le programme se lance jusqu'à ce qu'il croise *le point d'arrêt*.
- Voulez maintenant modifier le contenu des registres en cliquant sur le menu *View* puis *Current Mode* du menu *Registers*.
- Double cliquer sur le registre r0 et changer sa valeur à *0x00001234*
- Double cliquer sur le registre r1 et changer sa valeur à *0x87650000*
- Exécuter l'instruction ADD en sélectionnant *Step* du menu *Execute*
- Vérifier la valeur de r0
- Afficher le résultat sur différents formats : hexadécimal, décimal, octal, binaire et ASCII

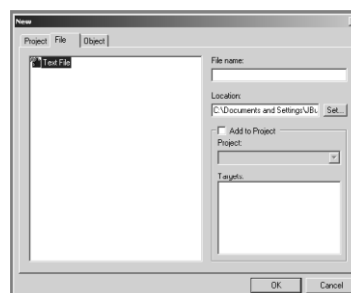


(c) Compiler et exécuter un programme en « c »

- Sélectionner *New* du menu *File*.



- Sélectionner l'icône *Projet*
- Cliquer sur *ARM Executable Image* et entrer *TDC* dans le champ *Project name*
- Vérifier le chemin du répertoire (le vôtre) de travail et cliquer sur *ok*
- Créer le fichier source en sélectionnant dans le menu *File, New*
- Cliquer sur l'icône *File*



- Cliquer sur *Text File* et entrer *program.c* comme nom dans *File name*
- Cliquer sur *Add to Project* et valider toutes les options proposées.
- Cliquer sur *ok*
- Entrer le programme ci-dessous dans la nouvelle fenêtre qui vient de s'ouvrir.
- Une fois terminé, sauvegarder ce fichier en cliquant sur *save* du menu *File*
- Sélectionner *Project* du menu *Make*. Ceci permet la génération d'une image binaire du code que vous venez d'entrer.
- Lancer le Debugger ARM
- Aller dans le menu *Execute* et faire *Go*

```
//Program.c
```

```
#include <stdio.h>
```

```
int main()
{
    int a, b, c;
    printf("hello, world\n");
    a = 3;
    b = 4;
    c = a + b;
    printf("%d + %d = %d\n", a, b, c);
}
```