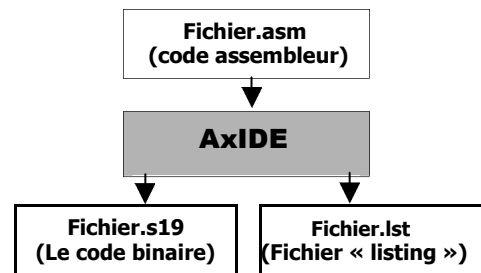


Séance1 : Environnement logiciel d'une carte d'évaluation 68HC11

La carte de développement CME11E9-EVBU est réalisée autour du microcontrôleur 68HC11 de Motorola dont le « langage machine » est l'Assembleur.

Les codes sources peuvent être saisis dans Notepad et assemblés (ou compilés, c'est-à-dire transformé en programme binaire exécutable par le µcontrôleur) avec le logiciel AxIDE (fourni par le fabricant).

Lors de l'opération de l'assemblage, AxIDE génère deux fichiers à partir de la source:



- Le premier fichier représente le code binaire (.s19) à télécharger vers la carte.
- Le second fichier (.lst) établit la relation entre le code source et le code binaire. Il est très utile durant les tests car il avertit des problèmes de compilations.

La liaison série (câble DB9) entre le PC et la carte doit être paramétrée comme suit :

- Port: COM1
- Vitesse : 9600 Bauds
- 8 bits de donnée
- 1 bit de stop
- Pas de parité

Au niveau de la carte, cette communication est gérée par le moniteur BUFFALO 3.4 installé dans la mémoire interne du processeur dont voici les commandes les plus usuelles :

- **LOAD T** : Téléchargement d'un fichier de type .s19 vers la mémoire de la carte
- **RM** : Visualisation et modification des registres du HC11
- **CALL** : Exécution d'un programme .s19
- **T** : Exécution d'un programme pas à pas
- **MD** : Visualisation de la mémoire
- **MM** : Modification de la mémoire

PROGRAMME
DE LA SEANCE

Nous consacrons cette séance à :

- L'apprentissage des manipulations élémentaires autour de notre carte de développement.
- La communication avec un port d'E/S du processeur
- L'écriture de données en mémoire.

DOCUMENTATION
TECHNIQUE ET LIENS

- [CME11E9-EVBU Development Board](#) : L'essentiel sur l'architecture de la carte.
- [BUFFALO MONITOR for HC11 Development Board](#) : Les commandes du Buffalo.
- [M68HC11E Family Technical Data](#) : Données spécifiques aux Acontrôleurs de notre labo.

MODALITES DE REMISE
DES COMPTE-RENDUS

Les élèves devront rédiger un compte-rendu pendant la séance ; pour rendre un rapport final du projet assez complet.

Travail préparatoire : Commenter les programmes de Q4 et de Q6.

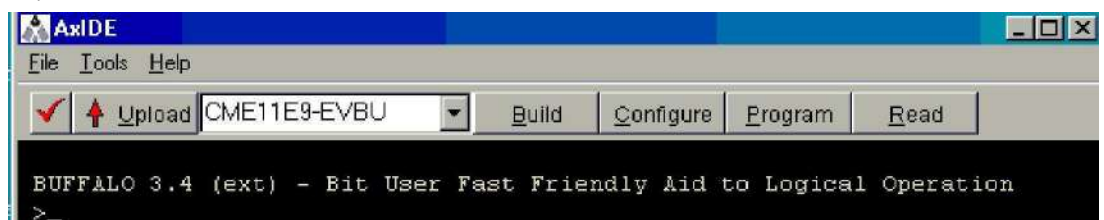
NB : lire attentivement les consignes ; les suivre pas à pas !!!

I. MISE EN ROUTE ET PREMIERS TESTS

LA LIAISON SERIE
SA CONNECTIQUE

Q0. Copier le dossier AxIDE3 de D:\Apps\ vers le dossier (z:\) d'un profil « élèves ».

- Appeler l'exécutable AxIDE.exe et paramétrer la liaison série dans File/Option/Port.
- Établir cette liaison entre le PC et la carte au moyen du câble DB9.
- Vérifier le bloc d'alimentation (le régler sur 6V) puis alimenter la carte.
- Effectuer si nécessaire un Reset (bouton poussoir au centre) sur la carte puis « Entrée » .



Le prompt ci-contre indique alors que la liaison entre le PC et la carte est bien établie.

LA PROCEDURE D'ASSEMBLAGE

Q1. Créer dans sa partition personnelle, au niveau du répertoire source, un nouveau répertoire « sn_prj » (par exemple). ATTENTION : pas de lettre accentuée, pas d'espace et un chemin de profondeur ≤ 2 répertoires)

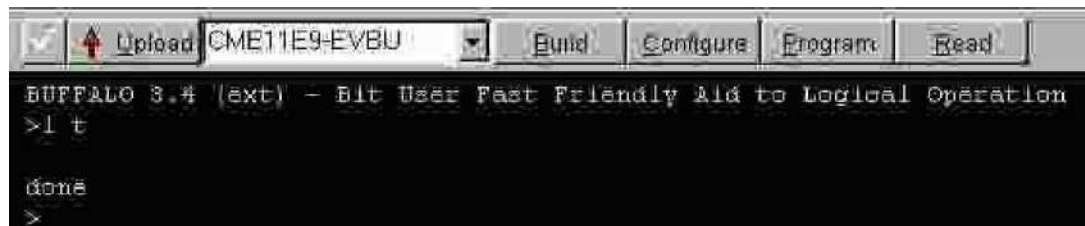
- Copier le fichier hello.asm, fourni par le constructeur dans Z:\AxIDE3\Exemples\hc11, dans le répertoire sn_prj puis l'ouvrir et le comprendre.
- Assembler le fichier hello.asm avec la commande **Build hello.asm**
- Relever le nom des deux fichiers générés dans le répertoire sn_prj.

TELECHARGEMENT ET EXECUTION

Le prompt ci-contre indique alors que le transfert a été effectué correctement.

Q2. Le téléchargement d'un fichier binaire peut s'effectuer de la manière suivante : Taper load t, ou tout simplement l t, pour alerter la cible.

- Télécharger le fichier vers la carte : Upload **hello.s19**



- Exécuter le programme en tapant CALL 2400. Relever la réponse de la carte et justifier la valeur de 2400 comme argument de CALL. **Validation Q2.**

II. ECRITURE D'UNE DONNEE DANS LA MEMOIRE

DIRECTIVES D'ASSEMBLAGE

Q3. On se propose en premier lieu d'écrire la donnée de valeur \$DA dans la RAM avec des directives d'assemblage. Saisir (Notepad), assembler (AxIDE) puis télécharger le programme ci-contre :

org	\$2000
fcb	\$DA
Veiller aux tabulations	

- Relever l'état de la mémoire au moyen de la commande MD 2000 de Buffalo.
- Éteindre l'alimentation de la carte et relever une nouvelle fois l'état de la mémoire à partir de l'adresse 2000 au moyen de la commande (memory display/dump) MD 2000.
- Lire et comprendre les fichiers .lst et .s19 relatifs à ces directives.
- Si nécessaire, reprendre la main avec un reset sur la carte. **Remarque ?**

INSTRUCTIONS PROGRAMMES

Q4. Le programme ci-contre permet le stockage de la même donnée au moyen d'un jeu d'instructions.

- Assembler, télécharger et lancer le programme avec CALL 200A.
- Examiner l'état de la mémoire de l'adresse 2000 à 200F au moyen de la commande MD 2000. Remarque ?
- Si nécessaire, reprendre la main avec un reset sur la carte.

	org	\$2000
val	rmb	1
	org	\$200A
	ldaa	#\$DA
	staa	val

Q5. Relever les fichiers .lst et .s19 relatif à ce programme

- Identifier les colonnes d'adresses, d'instructions et de données dans le fichier. .lst.
- Quel est le code binaire associé aux mnémoniques ldaa et staa ?
- Commenter la trame S19.
- **Validation Q5.**

Q6. Le programme ci-contre doit permettre le remplissage d'une zone de la RAM comprise entre les adresses «head» et «tail».

- Saisir ce code dans notepad. L'assembler et le télécharger.
- Quelle est l'adresse du début du programme ? Procéder à son exécution. Relever l'état de la mémoire entre «head» et «tail». **Remarque ?**

Q7. Lors de l'exécution pas à pas d'un programme, l'état des registres ainsi que celui du compteur de programme peuvent être examinés dans la fenêtre de AxIDE au moyen de la commande TRACE (T) de Buffalo.

- Modifier si nécessaire l'état du compteur de programme P comme cela est indiqué dans la notice de Buffalo (Commande RM).
- Procéder à l'exécution pas à pas avec T.
- Commenter le programme en examinant l'état des registres du processeur.
- Dresser un organigramme de ce programme pour décrire son exécution.

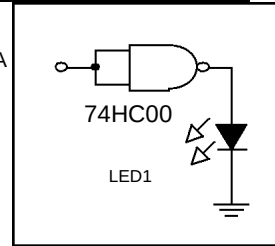
head	equ	\$2000
tail	equ	\$2009
	org	\$200A
	ldaa	#\$01
	ldx	#head
loop	staa	0,x
	inca	
	inx	
	cpx	#tail
	bne	loop
	end	

III. COMMUNICATION AVEC UN PORT D'E/S

PROGRAMMATION DU PORT A EN SORTIE

L'exercice consiste à utiliser le port A du μ contrôleur pour l'allumage/l'extinction d'une diode électroluminescente. Afin de protéger le port d'E/S d'éventuels court-circuits, nous demandons aux élèves de réaliser tout d'abord le petit montage de la figure ci-contre. (La documentation du 74HC00 est disponible sur le web).

Port_A
PA4



Q8. Désactiver l'alimentation de la carte. Réaliser ce montage et attaquer l'inverseur (buffer) avec la sortie PA4 du port A.

Faire vérifier le montage par un enseignant avant d'alimenter à nouveau la carte.

Q9. Consulter la section 6 de la documentation technique du 68HC11E et noter la fonction d'E/S de chaque bit du port A.

- Vérifier la fonction par défaut de sortie en PA4 ?
- Exécuter et commenter le petit programme ci-contre.

```
porta equ $1000
      org $2000
start ldaa #$10
      staa porta
```

PROGRAMMATION DU PORT A EN ENTREE

Q10. Le programme ci-contre illustre d'une manière très simple l'emploi du port A en entrée et en sortie.

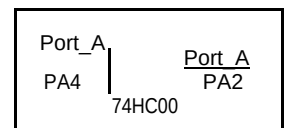
- Rappeler tout d'abord la fonction de PA2 par défaut et commenter chaque ligne de ce programme.
- Activer l'option de comptage des cycles machine au moment de l'assemblage comme cela est indiqué sur la fenêtre ci-dessous.

```
porta equ $1000
      org $2000
start ldaa porta
      asla
      asla
      staa porta
      bra start
```

- Consulter les données techniques relatives à la chronométrie du HC11E (section 2.5 : quartz/horloge) et en déduire la durée théorique de chaque boucle du programme.



- Déplacer la DEL et attaquer PA2 comme cela est représenté sur la figure ci contre. Lancer le programme de Q10.
- **Q11.** Observer le signal PA4 à l'oscilloscope et mesurer sa période. Commentaires et justification (cf Q10).



Q12. Modifier le programme de Q10 en déportant les instructions asla dans le programme sspro : Remplir les champs pointillés.

- Assembler ce programme et relever le nombre de cycles nécessaires à chaque boucle à partir du fichier .lst.
- En déduire la durée théorique de chaque boucle du programme.
- Lancer le programme et mesurer la période du signal PA4 à l'oscilloscope. Commentaires.

```
porta equ $1000
      org $2000
start ldaa porta
      jsr .....
      ..... porta
      bra .....

sspro asla
      asla
      rts
```

**RESTITUER LE MATERIEL
AVANT DE QUITTER LE
LABORATOIRE.**