

Master I Commande Electrique et Réseaux électriques

TP 2 Microprocesseurs et Microcontrôleurs:

Programmation des microcontrôleurs par assembleur

I. Objectif

Faire la saisie d'un programme (.asm) sur un logiciel MPLAB et le transfert vers le logiciel ISIS, puis procéder à une animation en temps réel.

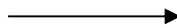
II. Edition, compilation et simulation du programme

Cette partie permet d'apprendre à créer un projet MPLAB, saisir un programme en assembleur, le compiler et le simuler.

II.1. Démarrage

- Avant de commencer, créer le répertoire *tp2* votre bureau
- Lancer le logiciel MPLAB IDE à partir de l'icône qui se trouve sur votre bureau.

Logo du MPLAB IDE



II.2. Création d'un nouveau projet avec « Project Wizard »

- Cliquer sur *Project >> Project Wizard ...*
- Cliquer sur *Suivant >* puis sélectionner PIC16F84 dans le menu déroulant *Device*.
- Cliquer sur *Suivant >* puis sélectionner « *Microchip MPASM Toolsuite* » et « *MPASM Assembler* » afin de pouvoir programmer en Assembleur.
- Cliquer sur *Suivant >* puis dans le champ « *Project Name* » saisir le nom du projet (ex : *exemple*) et dans le champ « *Project Directory* » aller chercher à l'aide du bouton « *Browse* », le dossier créé au début : *tp2/exemple*
- Cliquer sur *Suivant >* puis sur *Terminer*.

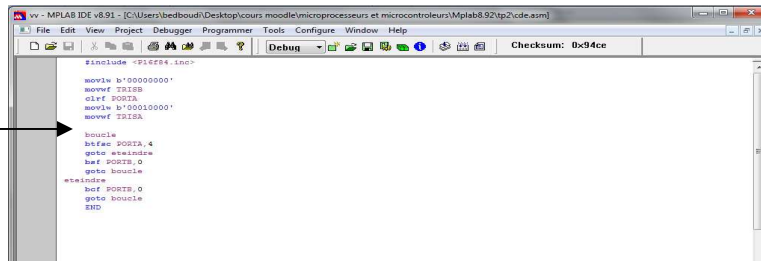
II.3. Edition du programme

- Créer un nouveau fichier dans le projet, *File >> Add New Files to Project...*
- L'enregistrer dans le répertoire projet avec l'extension *asm* (ex : *prog.asm*)

MPLAB Editor se lance

- Saisir le programme «*prog* » et enregistrer.
- Penser à ajouter des commentaires au programme pour montrer que vous avez bien compris les différentes instructions. Une ligne de commentaire commence par « ; ».

Fichier .asm



II.4. Compilation

• Cliquer sur Project >> Build All ... ou sur l'icône de la barre de menu ou encore Ctrl+F10 pour compiler le projet complet. Erreurs et warnings sont signalés dans la fenêtre Output ainsi que le résultat de la compilation BUILD SUCCEEDED ou BUILD FAILED.

En double-cliquant sur une erreur ou un warning dans la fenêtre Output, vous serez amené directement à la ligne concernée dans votre programme.

Bouton de compilation



II.5. Simulation

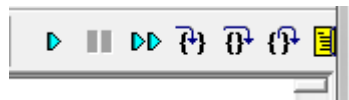
Avant d'envoyer un programme au μC , vous pouvez tester son fonctionnement dans le simulateur (Debugger) MPLAB SIM. Pour lancer le simulateur, cliquer sur Debugger >> Select Tool >> MPLAB SIM. Une nouvelle barre d'outils est ajoutée ainsi qu'un onglet « MPLAB SIM » dans la fenêtre Output. Le simulateur fonctionne selon trois modes :

- le mode *Step By Step* qui permet de faire une exécution pas à pas du programme (vous êtes alors l'horloge du μC).
- le mode *Animate* qui exécute automatiquement le programme mais à une vitesse réduite pour que l'on puisse suivre l'exécution.
- le mode *Run* où le programme est exécuté automatiquement à la vitesse du PC.

Dans les deux premiers modes, nous obtenons des résultats au fur et à mesure de la simulation alors que dans le dernier, les résultats ne sont visibles que lorsque la simulation est stoppée. Dans ce paragraphe, nous nous intéressons aux modes *Animate* et *Step By Step* car ils nous permettent de voir qu'elle est l'influence des entrées sur les sorties.

Le réglage de la fréquence de l'oscillateur utilisé se fait dans Debugger >> Settings Dans notre cas c'est 4 Mhz.

Boutons de simulation



III. Organisation d'un fichier .asm

La syntaxe d'écriture d'un programme en assembleur doit être la suivante:

Etiquette (facultative)	espace	Mnémonique	espace	Opérande ou valeur	,	Bit de destination ou n° de position	espace	;	Commentaire (facultatif)
		BSF		status	,	5		;	mettre le 5 ^{ème} bit (RP0) à 1 du status

- Etiquette (facultative),
- Espace (s) ou tabulation (s),
- Mnémonique ou espace(s),
- Opérande ou la valeur,
- Virgule éventuelle de séparation,
- Bit de destination w ou f ou éventuellement numéro du bit de 0 à 7 si nécessaire,
- Espace(s) ou tabulation(s),
- Point virgule (facultatif si pas commentaire),
- Commentaire (facultatif).

Remarque:

Il est à noter que:

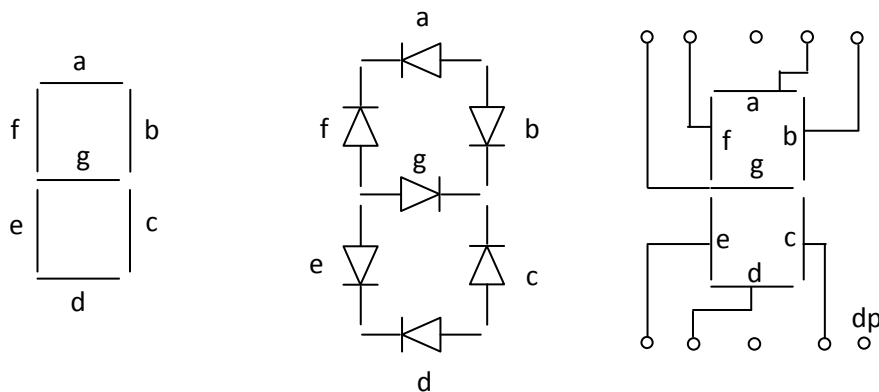
- La mnémonique ne peut pas se trouver en première colonne, et que tout ce qui suit le point virgule est ignoré de l'assembleur.
- La première colonne est réservée pour les étiquettes (repère).

IV. Manipulation

IV.1- Réaliser une commande d'un afficheur 7 segments par bouton poussoir en utilisant le PIC 16F84 qui permet en permanence de:

- Si on appui sur le bouton on affiche le chiffre "1" et si on relâche le bouton on affiche le chiffre "0" en utilisant les deux types d'afficheur 7 segments: à cathode commune et à anode commune.

L'afficheur 7 segments peut être représenté par les schémas suivants:



IV.2- Réaliser une commande d'un moteur à courant continu par bouton poussoir en utilisant le PIC 16F84.

IV.3- Réaliser une commande de l'inversion du sens de marche d'un moteur à courant continu par deux boutons poussoirs en utilisant le PIC 16F84 (un bouton pour la marche avant et l'autre pour la marche arrière).