



Travaux pratiques de Microprocesseur Présentation de l'environnement MPLAB

I. Introduction

MPLAB est un environnement gratuit fournis par Microchip (www.microchip.com) destiné à la gestion des projets de programmation de toute la gamme des microcontrôleurs PIC. Il comprend principalement :

- Un module d'édition pour l'écriture et la modification des sources de programmes
- Un module destiné à la simulation et à la correction des programmes
- Un module qui permet de charger le programme dans la mémoire du PIC

Pour lancer MPLAB cliquer sur l'icône MPLAB IDE qui se trouve sur le bureau Windows.

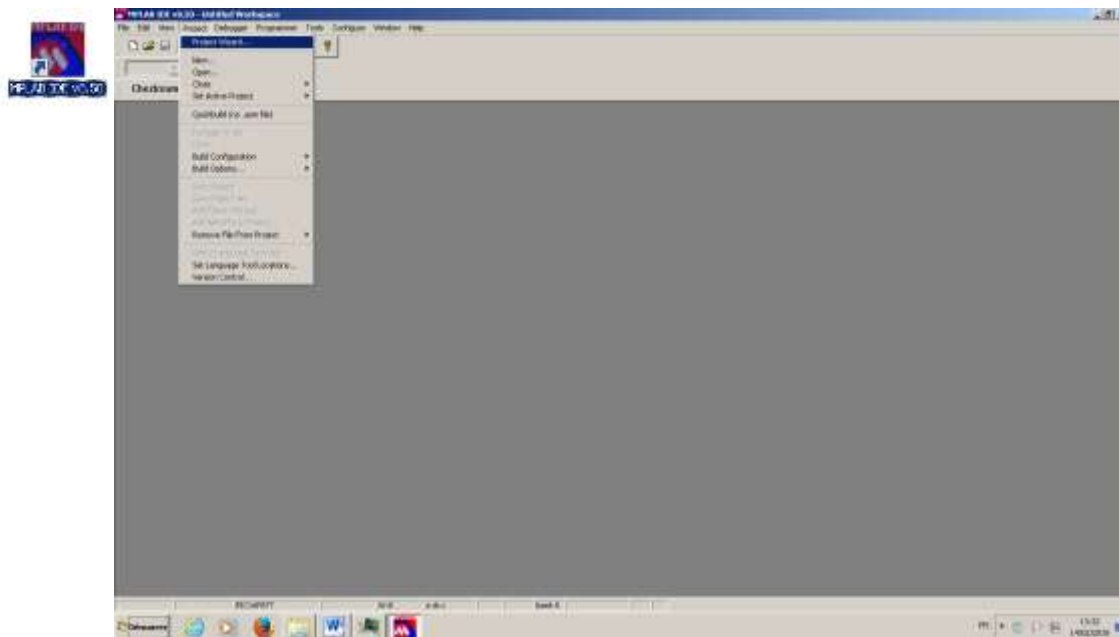


Figure 1

II. Créer un projet

MPLAB travaille avec la notion de projets. Pour créer un nouveau projet, cliquer sur **Project** → **Project Wizard**. Une boîte de dialogue apparaît (Figure 2) cliquer alors sur suivant.

La nouvelle boîte de dialogue vous permet de choisir le type de microcontrôleur sur lequel vous allez travailler (Figure 3). Sélectionner dans la fenêtre déroulante, **Device**, Le type **PIC16F877** puis cliquer sur suivant.



Figure 2



Figure 3

La boîte de dialogue suivante permet de choisir le langage de programmation qui va être utilisé dans le projet. Ici nous allons utiliser le langage assembleur. Dans la fenêtre déroulante, **Active Toolsuite**, choisir **Microchip MPASM Toolsuite** (voir figure 4) puis cliquer sur suivant.

La boîte de dialogue qui apparait ensuite (Figure 5) vous permet de créer un fichier de projet et de lui donner un nom. Dans la zone d'édition, **Create New Project File**, saisir le chemin complet et le nom du projet que vous voulez créer. Vous pouvez aussi vous aider du bouton **Browse...** pour choisir un dossier existant sur le disque. Dans la zone d'édition saisir **c:\microchip\TP0** puis cliquer sur suivant. Si une boîte de dialogue (Figure 6) vous demande si vous voulez créer le dossier **c:\microchip** cliquer sur **ok**.

La boîte de dialogue suivante (Figure 7) vous permet d'ajouter des fichiers existant à votre projet. Ici cliquer simplement sur suivant.

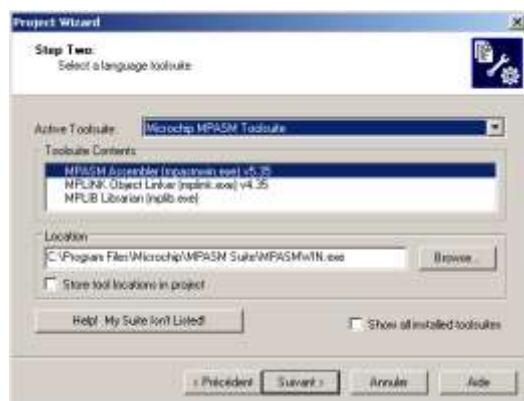


Figure 4

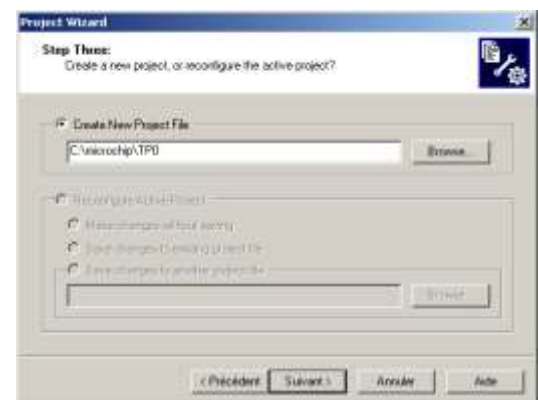


Figure 5

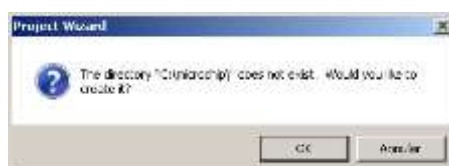


Figure 6

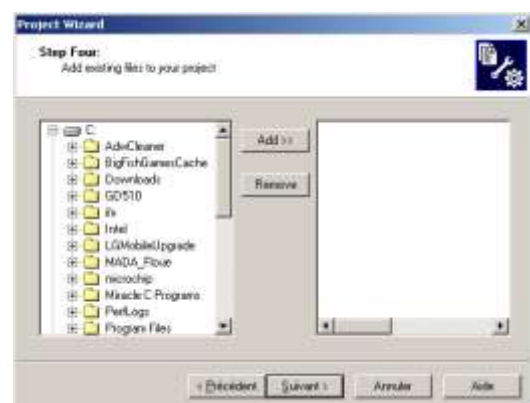


Figure 7

La dernière boîte de dialogue vous permet de voir un résumé des différentes options que vous avez choisies. Noter qu'à chaque étape vous pouvez revenir en arrière et modifier une option. Si le résumé correspond à ce que vous voulez, cliquer sur terminer pour voir apparaître la fenêtre de projet (Figure 8). Si ce n'est pas le cas, cliquer alors sur le menu **View** → **Project** (Figure 9)



Figure 8

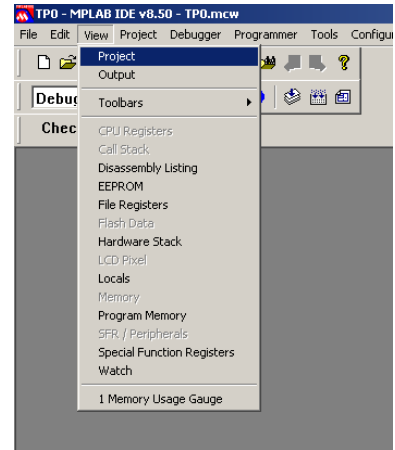


Figure 9

II Ajouter un Fichier source

Dans la fenêtre de projet (**TP0.mcw** dans la figure 8), le projet (**TP0.mcp**) se présente sous la forme d'un dossier contenant plusieurs sous dossiers dont chacun peut contenir un ou plusieurs fichiers, la nature de ces fichiers dépend du sous dossier. L'unique sous dossier que nous allons utiliser dans ce TP est le premier ; **Source Files**.

Le dossier Source File sert à recevoir le fichier source contenant le programme assembleur. Pour ajouter un fichier dans ce dossier, Cliquer avec le bouton droit de la souris puis sélectionner **Add Files...**(Figure 10). Une boîte de dialogue apparaît. Allez dans le dossier **c:\Program Files\Microchip\MPASM Suite\Example** puis Sélectionner le fichier **exemple.asm**. (Figure 11).

Pour supprimer le fichier ajouté, il suffit de le sélectionner à l'aide de la souris puis d'appuyer sur la touche **Suppr (Delete)** au clavier.



Figure 10

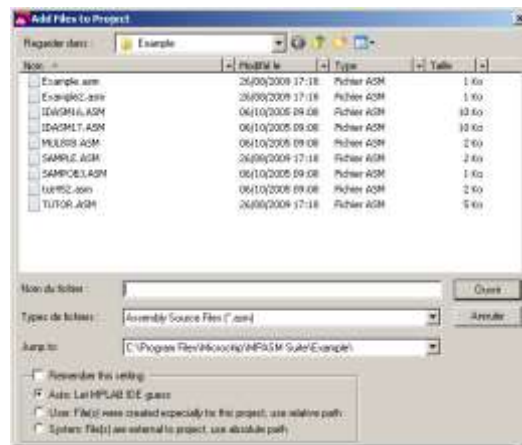


Figure 11

Pour créer un nouveau fichier il faut cliquer sur le menu **File** → **New**. La nouvelle fenêtre qui apparait porte le nom **untitled**, il faut alors l'enregistrer (**File** → **Save as...**) et lui donner un nom avec l'extension **.asm**. il faut veillez à toujours enregistrer vos fichier dans le dossier c :\microchip.

III. Programmer en assembleur

Ouvrir le fichier **Example.asm**. Ce fichier contient un programme écrit en langage assembleur pour PIC. Dans ce fichier source il existe plusieurs sections, chacune est identifiée par une couleur propre :

a. Les commentaires

Les commentaires sont identifiés par la couleur verte et sont précédés par '**;**' ils peuvent être écrits à n'importe quelle point du fichier. Les commentaires sont destinés au lecteur et sont ignorés par le compilateur.

b. Les directives

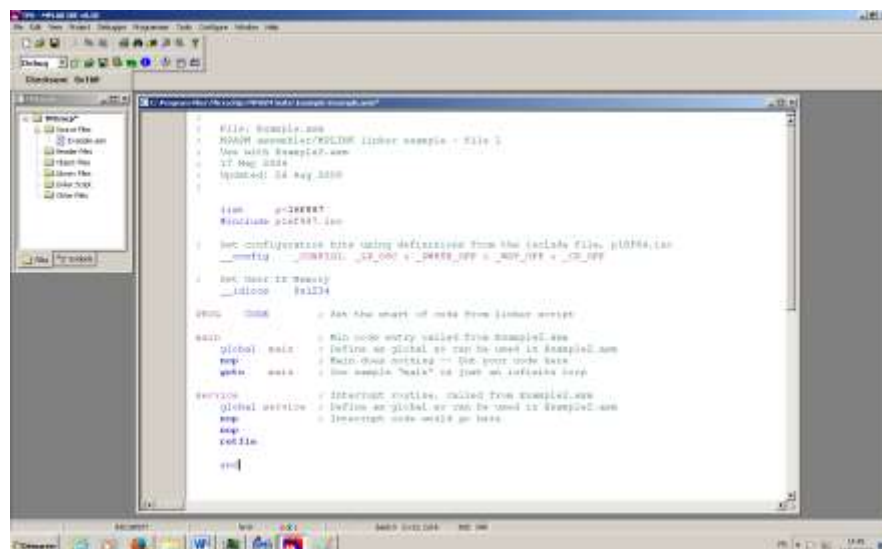
Ce sont des pseudo-instructions qui sont destinées au compilateur et l'aide dans l'opération de création du programme final (langage machine). Elles sont indiquées par la couleur bleue. Les directives ne doivent pas être écrites en première colonne du fichier. Un exemple est la directive **end** qui indique au compilateur que le fichier est terminé en ce point.

c. Les instructions

Les instructions sont la seule partie du fichier qui va être traduite en code machine. Chaque microcontrôleur possède son propre jeu d'instruction. Elles sont indiquées par la couleur bleu foncé. Les instructions ne doivent pas être placées en première colonne du fichier et il doit y avoir une seule instruction par ligne.

d. Les étiquettes

Ce sont des mots libres choisis par le programmeur pour indiquer l'emplacement d'une instruction. Les étiquettes doivent obligatoirement être placées en première colonne avant l'instruction ciblée sur la même ligne ou sur la ligne d'avant. Les étiquettes sont indiquées par la couleur rouge.



La programmation avec MPLAB doit respecter ces quelques contraintes

- La première colonne du fichier est réservée soit pour un commentaire soit pour une étiquette
- Il est préférable de spécifier le type de composant utilisé avec la directive **List p =**
- On doit spécifier le fichier d'entête qui contient le mapping mémoire du composant choisit avec la directive **#include**. Tout autre fichier utilisé doit être indiqué avec cette directive
- Le code introduit (le programme) doit être précédé par la directive **org xxxx** pour que le compilateur sache à quelle adresse mémoire le programme doit être chargé.

III. compiler et exécuter un programme

Une fois le code entièrement écrit, il peut être testé en simulation avant qu'il ne soit chargé dans le PIC. Pour cela, sélectionner dans le menu **Debugger** → **Select Tools** → **MPLAB SIM** cela vous permet de tester et de simuler votre code avant de le charger dans la mémoire du PIC.

Pour compiler le projet, sélectionner dans le menu **Project** → **Build all**. Une boîte de dialogue s'ouvre alors et vous indique toutes les opérations qui se font sur le fichier source (la compilation, l'assemblage, la liaison,...) s'il n'y a pas d'erreurs la barre de progression s'affiche en vert et une flèche verte pointe sur la première instruction du code. Le programme est alors prêt pour être exécuté.

Vous avez plusieurs options pour exécuter le code

- Soit entièrement en sélectionnant **Debugger** → **Run (F9)**
- Soit pas à pas en sélectionnant **Debugger** → **Step into**

A chaque fois qu'une instruction est exécutée la barre d'état (en bas) vous indique l'état du PIC (PC, registre d'état, ...)

Le menu **View** vous permet

- d'afficher le contenu de la mémoire programme
- d'afficher le contenu de la mémoire donnée
- d'afficher le contenu de l'Eeprom
- d'afficher le contenu de la pile
- d'ajouter une observation sur une variable (watch)

Essayer de découvrir toutes les autres options de l'environnement ; vous en aurez besoin !