IUT de Poitiers – Site de Châtellerault

Départements
Mesures Physiques
Réseaux et Télécommunications

Formation Micro-contrôleur

Généralité Micro-Contrôleur
Utilisation de MPLAB

Les débuts avec MPLAB

1 Préparation à l'utilisation

La première chose à faire est d'aller chercher la version actuelle de MPLAB 7.20 sur le site Microchip : http://www.microchip.com suivre les liens « tools -> MPLAB IDE ». Dans ce cours, les copies d'écrans ont été faites avec MPLAB 7.2, mais les différences devraient être minimes pour les versions plus récentes

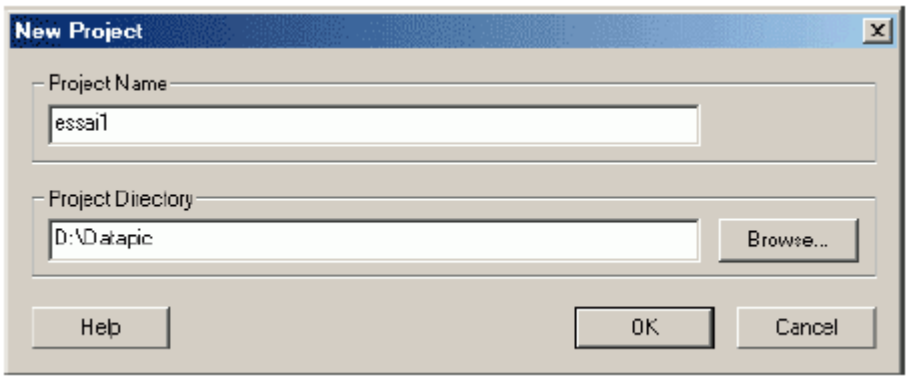
.

Décompactez le fichier et procédez à son installation. Lors de l'installation, vous aurez plusieurs fenêtres explicatives concernant différents outils de Microchip, comme le debugger et le simulateur.

2 Création de notre premier projet

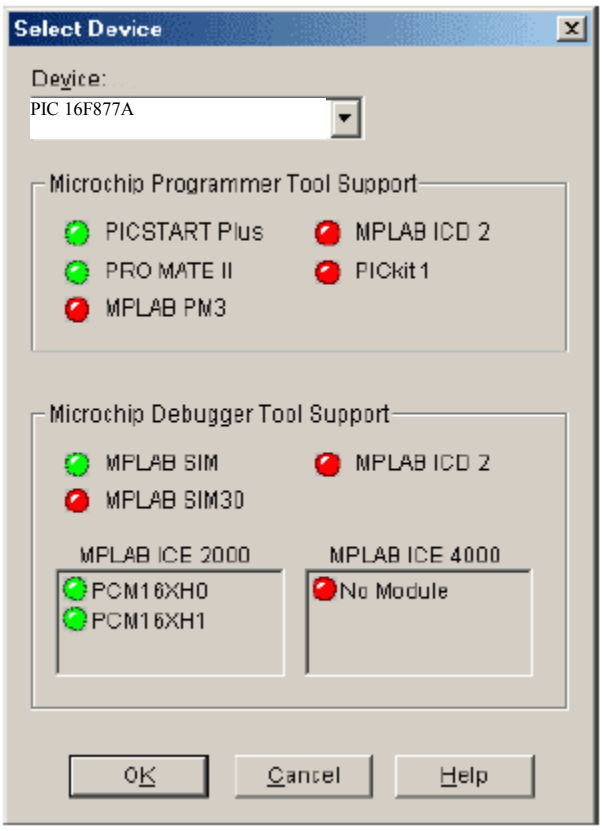
Vous pouvez maintenant lancer MPLAB IDE à partir du menu démarrer ou de l'icône de votre bureau, si vous en avez accepté l'installation. Après quelques instants, vous vous retrouvez avec un écran vide avec menu et barres d'outils. S'il y a des fenêtres ouvertes dans le bureau de MPLAB 7, fermez-les toutes, ainsi vous saurez comment les rouvrir, et tout le monde débutera avec la même configuration.
MPLAB est un logiciel qui est construit sur la notion de projets. Un projet permet de mémoriser tout l'environnement de travail nécessaire à la construction d'un …. projet. Il vous permettra de rétablir tout votre environnement de travail lorsque vous le sélectionnerez.

MPLAB 7 dispose d'un « wizard » pour la création des projets, qui vous permet de créer automatiquement de nouveaux projets. Cependant, afin de vous permettre de localiser les principales options des projets, je ne m'en servirai pas dans ce petit tutorial. Si par la suite vous désirez l'utiliser, il se sélectionne à l'aide du menu « project->wizard »
Allez dans le menu « Project » et sélectionnez « new… ». La fenêtre qui s'ouvre vous permet d'introduire le nom du projet et le répertoire de travail du dit projet. Pour le répertoire, vous disposez du bouton « browser » qui vous permet de pointer sur le bon répertoire sans risque de vous tromper.
Entrez « essai1 » comme nom de votre nouveau projet, et sélectionnez le répertoire dans lequel vous avez placé votre fichier maquette et votre fichier essai1.asm. J'ai nommé le fichier asm de façon identique au projet, mais ce n'est nullement obligatoire.

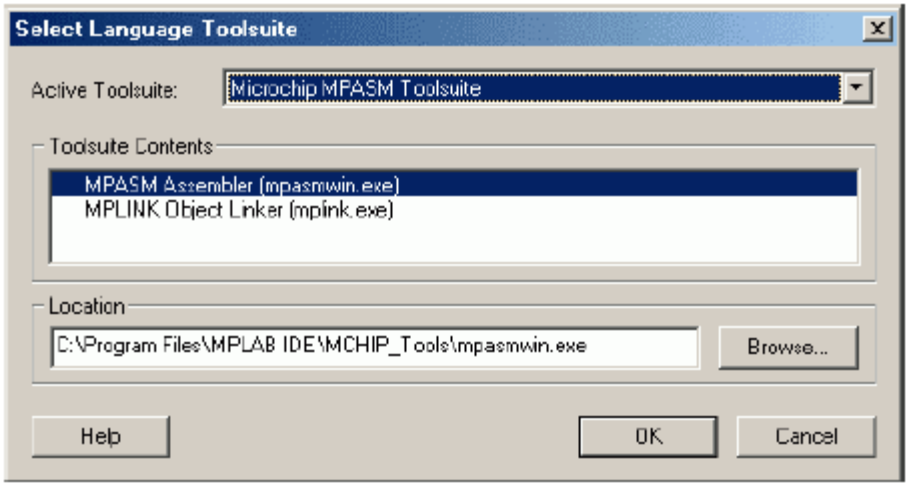


Une fois le bouton <OK> pressé, une nouvelle fenêtre apparaît dans le coin supérieur gauche du bureau de MPLAB IDE.

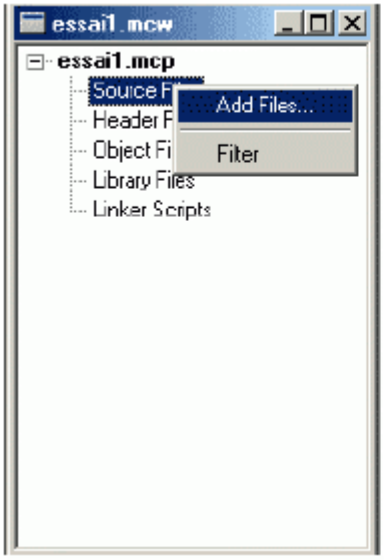
Dans cette fenêtre, vous voyez le nom de votre projet (essai1.mcp), ainsi que les fichiers liés à ce projet. Pour l'instant, vous n'en avez aucun, c'est normal.
Nous allons commencer à préciser les paramètres importants de notre projet, et tout d'abord le type de pic que nous allons utiliser. Sélectionner le menu « configure->select device », une fenêtre apparaît pour vous proposer le choix du pic. Notez que par défaut, Microchip vous propose un pic de la famille 18F, promotion de nouveau produit oblige…
Sélectionnez le 16F877A

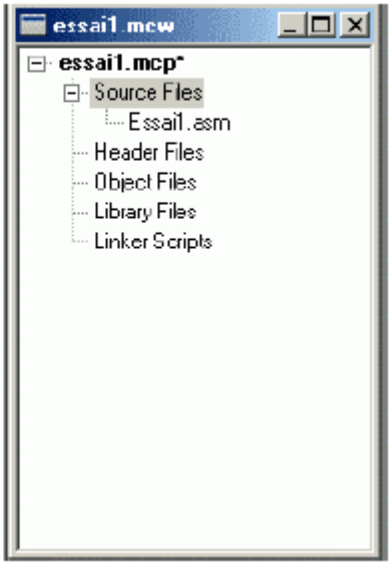


La fenêtre vous montre alors, à l'aide de leds vertes ou rouges quels outils sont supportés par le pic sélectionné. Vous voyez donc que le simulateur intégré (MPLAB SIM) fonctionne avec ce pic, et ainsi de même pour les autres outils de développement disponibles chez Microchip. Une fois cliquée <OK>, la fenêtre se referme.
Nous allons maintenant préciser quel langage nous allons utiliser, sachant que nous travaillons en assembleur, mais que différents compilateurs sont proposés par Microchip et d'autres sociétés. Sélectionnez le menu « project -> Select langage toolsuite ». Dans la fenêtre qui s'ouvre, sélectionnez dans le menu déroulant : « Microchip MPASM toolsuite ». MPASM est en effet l'assembleur par défaut de Microchip.



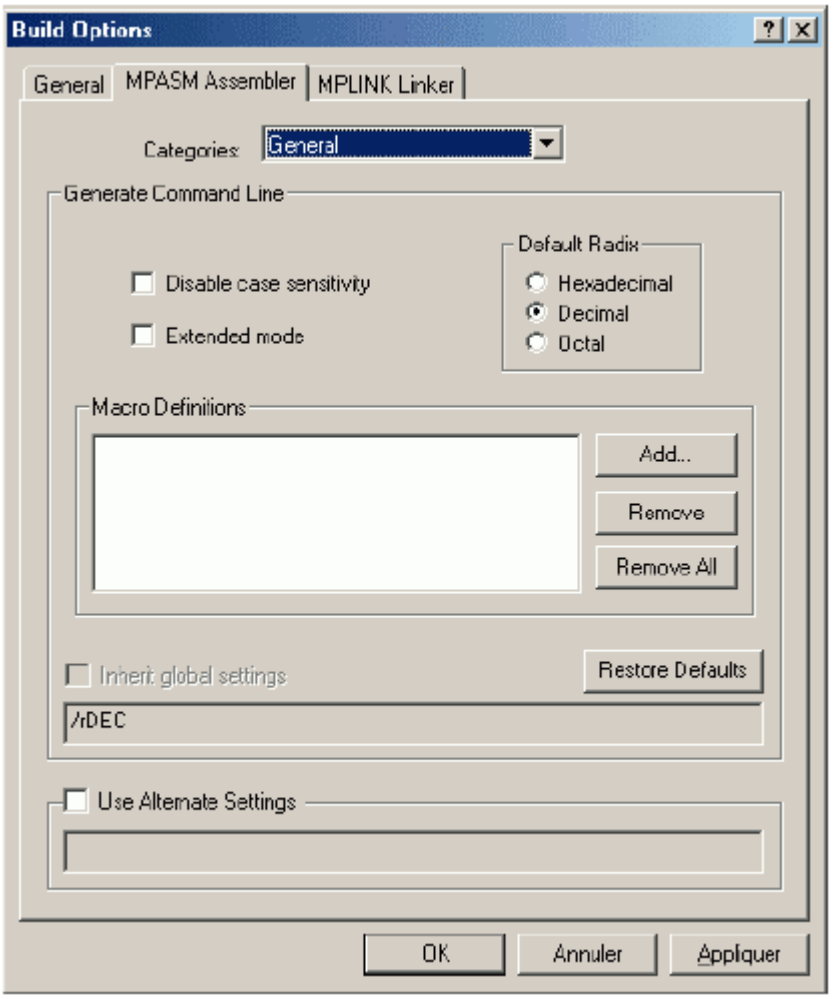
Dans les fenêtres inférieures, vous voyez le nom des exécutables utilisés par MPASM, ne vous en préoccupez pas. Cliquez <OK> pour fermer la fenêtre.
Il nous faut maintenant indiquer à MPASM quel est notre ou nos fichiers source(s). Ceci s'effectue dans la fenêtre « essai1.mcw » qui est restée ouverte dans le coin supérieur gauche. Pour ajouter un fichier source, c'est très simple : cliquez avec le bouton droit sur « source files », puis sélectionnez « Add ». Une fois sélectionnée le fichier, le nom de celui-ci apparaît dans l'arborescence. Notez que MPLAB pointe par défaut sur votre répertoire de travail, précédemment choisi.





Ici, il est important de bien comprendre que le fichier source choisi sera celui qui sera assemblé ou compilé. Autrement dit, si dans ce fichier se trouve une instruction « include » qui inclut un autre fichier, vous ne devez pas le sélectionner en supplément, il sera joint au projet au moment de l'assemblage.
Par contre, si vous désirez assembler simultanément deux fichiers, et que votre premier fichier ne fait pas référence au second, alors vous devez ajouter ce fichier à l'arborescence.

Tous nos projets ne nécessiteront que d'ajouter un seul fichier. Pour le cas où plusieurs fichiers sont nécessaires, la directive « include » sera ajoutée au fichier source afin que l'assembleur intègre automatiquement le ou les autre(s) fichier(s). Vous n'avez besoin de rien d'autre, les autres éléments de l'arborescence ne sont pas nécessaires pour un projet assembleur du type dont nous allons parler.
Il nous reste un élément important à préciser, celui du système de numérotation utilisé par défaut. Sélectionnez le menu : « Project -> build options -> project ». Une fenêtre s'ouvre. Sélectionnez l'onglet « MPASM assembler ».



Comme nous avons décidé de numéroter l'hexadécimal sous la forme 0x, et le binaire sous la forme B''. Tout nombre sans préfixe sera considéré comme décimal. Cochez donc la case « Decimal ». Ne touchez pas aux autres options, vous les utiliserez peut-être lorsque vous serez devenus un « pro » et que vous voudrez adapter certains paramètres. A ce moment, leur rôle vous semblera clair.

3 Débuggage

Pour passer un programme en mode simulation, vous devez, depuis le logiciel MPLAB ouvrir votre projet tel que défini dans les chapitres précédents.
Dans le menu Debugger, sélectionnez Select Tool -> MPLAB SIM

Une fois le mode simulation déclaré, il faut placer les stimuli qui correspondent aux entrées du PIC. Pour ce faire, allez dans le menu « Debugger » et sélectionnez « Stimulus Controler ». Vous devez alors avoir une fenêtre supplémentaire appelée MPlab Stimule**.** C'est dans cette fenêtre que nous allons déclarer les entrées qui seront alors modifiables par un clic de souris.
Dans notre programme « clignotant », nous n'avons en fait qu'une seule entrée à piloter, c'est le bouton marche-arrêt dont l'état permet au promgramme soit de se lancer, soit de se reboucler sur

le test de cet interrupteur. C'est la broche RAO qui a pour rôle de recevoir cet interrupteur, nous allons donc déclarer celle-ci dans la fenêtre de stimuli.
Pour déclarer la broche RAO en entrée de stimuli, cliquez sur le bouton « ADD ROW », puis dans le type, sélectionnez « Asynch », ainsi que la broche concernée 3RAO » ainsi que dans la dernière liste déroulante, l'option « Toggle ». Le bouton « Fire » est là pour valider désormais vos clics de souris. Sachant que nous avons pris l'option « Toggle », chaque appui sur « Fire » fera changer l'état de la broche RAO. Bien sûr, si nous avions plusieurs entrées à simuler, il suffirait de refaire un « Add Row » et de recommencer les étapes détaillées ci-dessus.

Nous venons de mettre en place notre entrée de validation RAO et nous pouvons dès maintenant commencer la simulation. Pour ce faire, et afin de visualiser l'action sur le bouton « Fire » qui simulera notre entrée RAO, nous allons ajouter une vue d'écran. Cliquez dans le menu » View », puis sélectionnez « Special Function Registers », afin de visualiser l'état du port A (pour visualiser l'entrée RAO entre autre).

Sélectionnez maintenant le menu « Debugger » puis cliquez sur « Animate » dans la fenêtre où se trouve la source du programme. Il apparaît une flèche verte qui se positionne au rythme de l'horloge sur l'instruction en cours d'éxécution. Si vous n'avez pas appuyé encore sur le bouton « Fire », la flèche oscille entre les deux instructions :
Btfss PORTA, inter0 ; interrupteur 0 (marche) appuyé ?si oui on continue sinon on retourne à l'étiquette début
Goto début Appuyez maintenant sur le bouton « Fire » et notez au passage le changement d'état du registre PORTA de la fenêtre Special Function Registers qui passe de 0000 0000 à 0000 0001 puisque la broche RAO vient de passer à 1 par le clic de souris. A ce moment, le programme continue de se dérouler. L'appui sur le bouton marche-arrêt vient d'être simulé : pratique non ?
Pour arrêter le mode animation, appuyez sur la touche F5 ou bien allez dans le menu « Debugger » puis cliquez sur « Halt ».
Pour faire un reset de l'animation, appuyez sur la touche F6 ou bien allez dans le menu « Debugger » puis cliquez sur « Reset » et sur « Processor Reset ».
Depuis le menu view, vous pouvez ajouter des écrans composés de différents registres internes du PIC, y compris ceux que vous avez déclarés dans votre programme. Par exemple, depuis le menu « View », cliquez sur « Watch », puis dans la liste déroulante « Add Symbol » sélectionnez le registre « retard 1 » et validez sur le bouton « Add Symbol ». Comme vous l'avez constaté, vous pouvez visualiser tous les registres internes au PIC ce qui est très pédagogique pour en comprendre le fonctionnement.

Nous voici arrivés au stade où nous avons écrit notre programme, nous l'avons simulé et tout se passe bien : la simulation correspond à nos attentes…

Nous pouvons maintenant transférer le fichier compilé(clignotant.hex) vers la mémoire du PIC.

Phase de programmation

Trois signaux sont nécessaires pour programmer un PIC, en effet la programmation nécessite un signal d'horloge. Celui-ci sera appliqué sur la broche RB6 du PIC, quant aux données, elles transiteront sur la broche RB7.

Il est à noter que cette broche sera bidirectionnelle puisque l'on pourra également lire le programme contenu dans la mémoire flash du PIC ainsi que dans la mémoire EEPROM.

Lors de la programmation, une tension de 12 V est appliquée par le programmateur sur la broche MCLR/ indiquant à la logique interne que le mode programmation est demandé. Un reste du PIC pour initialiser le compteur ordinal est alors effectué.

Programmation ICP

La programmation du PIC nécessite soit d'enlever le PIC du montage sur lequel il est implanté et de l'insérer dans le programmateur, soit de réaliser la programmation in-situ (directement sur la platine d'application). On parlera alors de programmation ICP (In-circuit Programming). Le mode ICP est très pratique puisque l'on ne doit pas enlever le PIC du montage pour le programmer ce qui est appréciable. Lorsque l'on fait de la mise au point, il y a quand même quelques précautions à prendre, notamment l'isolation des broches RB6 et RB7 qui sont utilisées lors de la phase de programmation.

Les fils de connexions entre la platine et le programmateur devront être le plus court possible pour éviter les effets de capacité parasite qui pourraient nuire à la programmation. Il est à noter que la masse doit être reliée entre le programmateur et la platine ainsi que le +5V.

La programmation de la plaquette qui est à votre disposition est réalisée à partir du programme PicFlash.exe.