

Prise en main de l'environnement MPLAB 8.56 (IDE Microchip) pour programmer entre autre la carte PICDEM2+

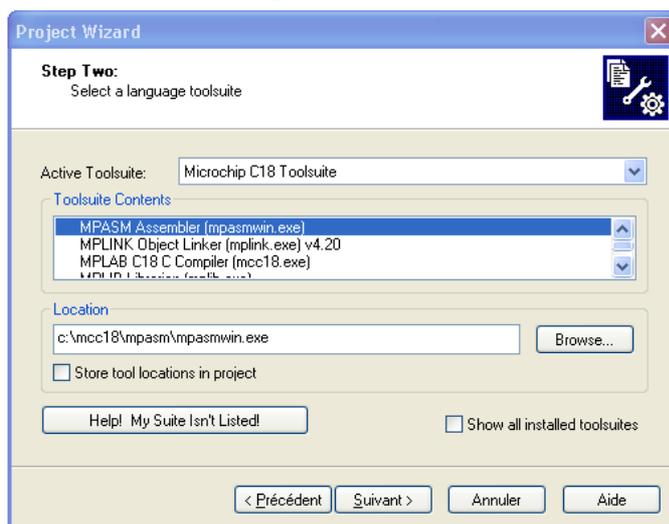
Créez votre premier projet:

Pour prendre en main l'environnement de la carte:

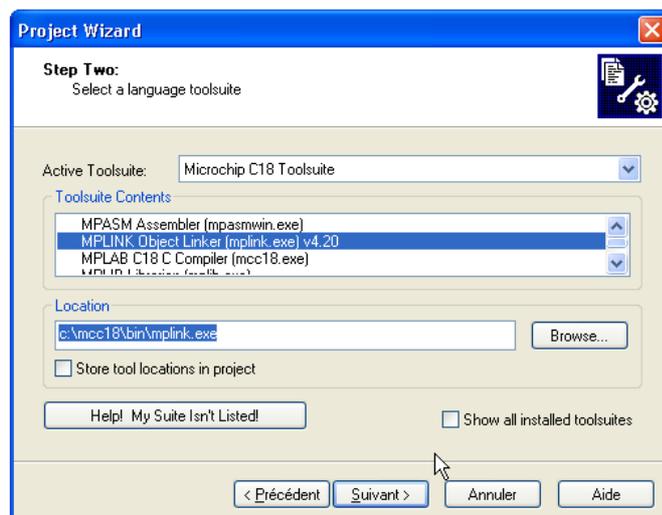
- Rapatriez le fichier *testLED.c* sur *intranet/~lagreze* puis *DocumentationsDiversesPourTP\manuelsPriseEnMain\PriseEnMainMPLAB* dans un répertoire que vous aurez créé à cet effet.

Lancez l'IDE MPLAB par le menu (Démarrer->Programmes->Electronique -> Microchip-> MPLABIDE)

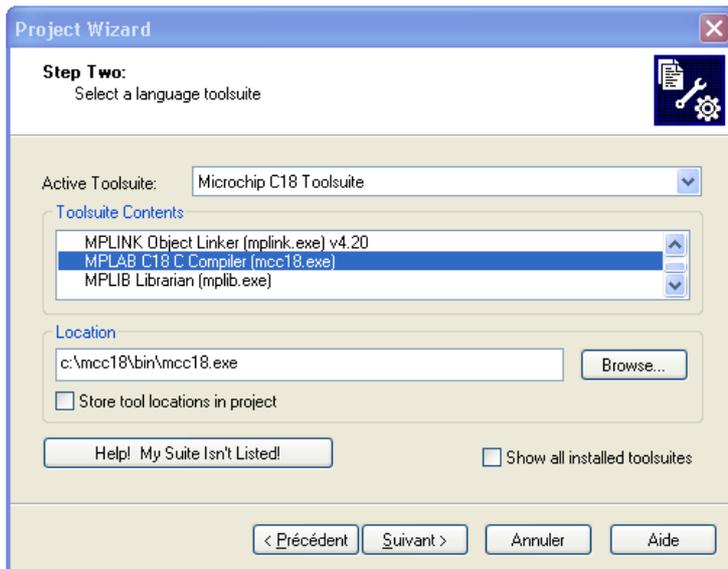
- Créez un nouveau projet par **Project -> Project Wizard :**
 - Device : 18F458
 - choix du compilateur mcc18



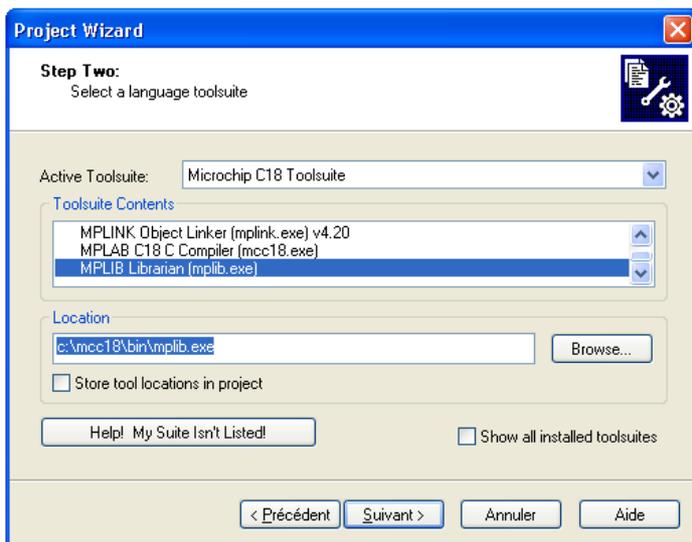
On indique ici le chemin d'accès pour que l'environnement trouve le logiciel *assembleur* qui permet de convertir le langage d'assemblage fourni par le compilateur à partir du code source en langage machine, c'est-à-dire du code binaire directement exécutable par le microcontrôleur.



On indique ici le chemin d'accès à l'*éditeur de liens* (ou *linker*) qui effectue l'assemblage des différents modules du projet (et relie ensemble les identifiants utilisés dans plusieurs modules comme les noms de fonctions ou de variables globales) et produit en final le code binaire qui sera exécuté par le microcontrôleur.



On indique ici le chemin d'accès au *compilateur* qui convertit les *fichiers sources* (suffixés .c) en *fichiers objets* (suffixés .o) qui sont dans un langage intermédiaire proche du binaire et où toutes les adresses des identifiants (variables globales et fonctions) ne sont pas connues (on dit résolues).



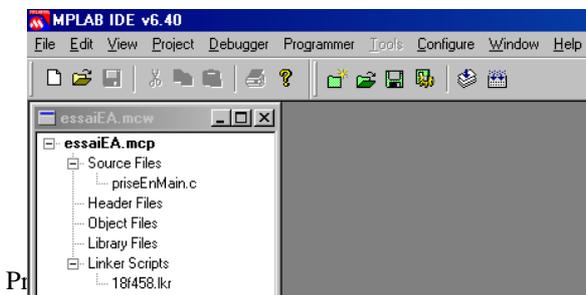
On indique ici le chemin d'accès à l'*utilitaire de librairie* qui permet de rassembler des fichiers compilés (.o) pour en faire une librairie directement utilisable.

N.

- Après avoir cliqué sur *Suivant>* situez votre projet dans le nouveau répertoire que vous venez de créer et donnez lui un nom (*priseEnMain* par exemple). Cliquez sur *Suivant>*. Ne pas renseigner le Step 4 et appuyer sur *suivant>* puis *terminer*.
- Affichez la fenêtre de projet par **Windows -> priseEnMain.mcw**
Ajoutez à votre projet le fichier *testLED.c* par **Project -> Add Files to Project**
Même chose avec le fichier *C:\ProgramFile\mcc18\lkr\18f458i.lkr*. On peut aussi pointer le sous répertoire *Source Files* avec la souris et cliquer sur le bouton droit.

Vous devez obtenir un écran de ce type :

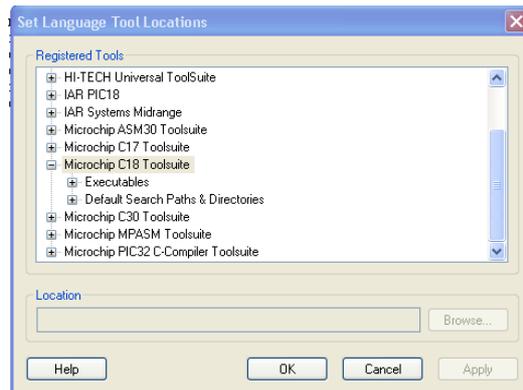
Après avoir terminé la création votre fenêtre projet devra avoir l'aspect suivant :



Pt

Vérifier que vous avez bien les configurations suivantes (les ajouter si besoin) :

- Dans **Project - >Set Language ToolSuite** choisir *MPLAB C18C Compiler* et vérifiez que les chemins d'accès aux exécutables de l'environnement (*mpasmwin, mcc18, mplink, mplib*) sont corrects .
- Dans **Project - >Set Language ToolLocation** choisir *Microchip C18 ToolSuite* et vérifiez que les chemins aux exécutables sont corrects.
Positionnez les chemins de recherche aux différents fichiers (headers .h, librairies .lib, commandes du linker .lkr) dans default *Search Path &Directories* comme suit :

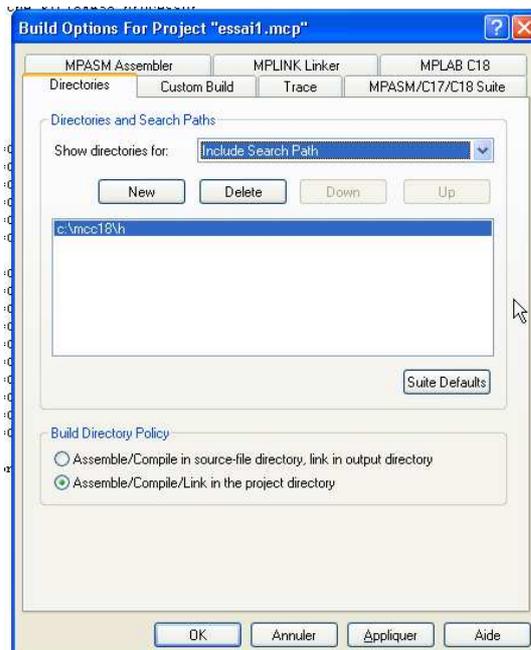


Include search path -> c:\mcc18\h C'est là que le compilateur ira chercher les fichiers headers (.h) déclarés par #include <.....h>

Library search path -> c:\mcc18\lib chemin de recherche des librairies du langage C qui contiennent le code correspondant à toutes les fonctions systèmes.

Linker-script search path -> c:\mcc18\lkr

Ne pas oublier de cliquer sur **Apply** avant de sortir par **OK**.



- Dans **Project - >Build Options** choisir **Project** . Dans l'onglet **Directories**, cliquez sur **Suite Defaults** puis vérifiez avec la liste déroulante que les chemins d'accès aux fichiers .h,.lib et .lkr sont bien positionnés. Si ce n'est pas le cas, faites le.

- Dans **Debugger-> Select Tool** choisir **MPLAB ICD2** qui est le « In Circuit Debugger ». Un message vous avertit que le boîtier de débogage n'est pas connecté.
- Dans **Configure -> Select Device** vérifiez que le PIC18F458 est bien le microcontrôleur choisi (simple vérification puisque cela a été renseigné dans l'assistant de création de projet) .

- Dans **Configure -> Configuration Bits** après avoir décoché la case *Configuration Bits Set In Code* choisir
 - Oscillator **-EC-OS2 as RA6**
 - Oscillator Switch Enable- **Disable**
 - Power Up Timer – **Disable**
 - Brown Out Detect **-Disable**
 - Watch Dog Timer—**Disable**
 - Low Voltage Programming-**Disable**
 - Stack Overflow Reset **-Disable**
 Tous les autres bits de configuration doivent être Disable

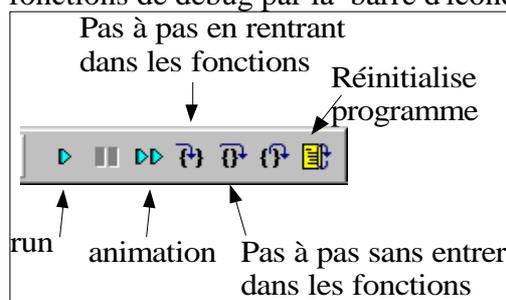
Recocher la case *Configuration Bits Set In Code*.

- Sauvegardez votre projet par **Project -> Save Project**.
- Connectez l'ICD2 (boîtier rond) à un des ports USB (fil long) de votre PC et à la carte (fil court) et connectez l'alimentation à la carte puis faire **Debugger -> Connect** et attendre le message de connexion.
- Puisqu'on va travailler dans un premier temps avec le debugger, il faut adjoindre au code de l'application qui sera chargé dans la mémoire, un code supplémentaire permettant la communication avec la sonde ICD2 et offrant des fonctions de mise au points (points d'arrêt, pas à pas ...). Pour cela choisissez debug dans la liste représentée si dessous :



Compilez ensuite votre projet par **Project ->Build All** ou l'icône 

- Téléchargez le programme exécutable par **Debugger -> Program**. Lancez le programme par **Debugger -> Run** (ou F9). Constatez le bon fonctionnement. Arrêtez le programme par **Debugger -> Halt** puis relancez le par **Debugger -> Run**
 - Posez un point d'arrêt au début du main (click droit sur la ligne), lancez le programme (**Debugger -> Run-ou F9 ou logo**). Exécutez pas à pas le programme et regardez PORTB évoluer au cours du pas à pas. (**View -> Watch** puis choisir PORTB dans liste déroulante)
- On a également accès aux fonctions de debug par la barre d'icônes



- Lorsque le programme est au point après une longue séance de débogage, on peut programmer la mémoire flash de la carte avec la l'application uniquement (sans la partie debugger) et rendre ainsi la carte autonome. Pour ce faire, passer en mode *release* dans la barre d'outils :



puis faire **Programmer -> Select Programmer** puis **Programmer -> Program**. Vérifiez alors que la carte peut fonctionner indépendamment de l'outil de mise au point (boîtier ICD2) en débranchant la sonde. Débranchez et rebranchez l'alimentation de la carte et constatez que la programmation de la mémoire flash est bien permanente.