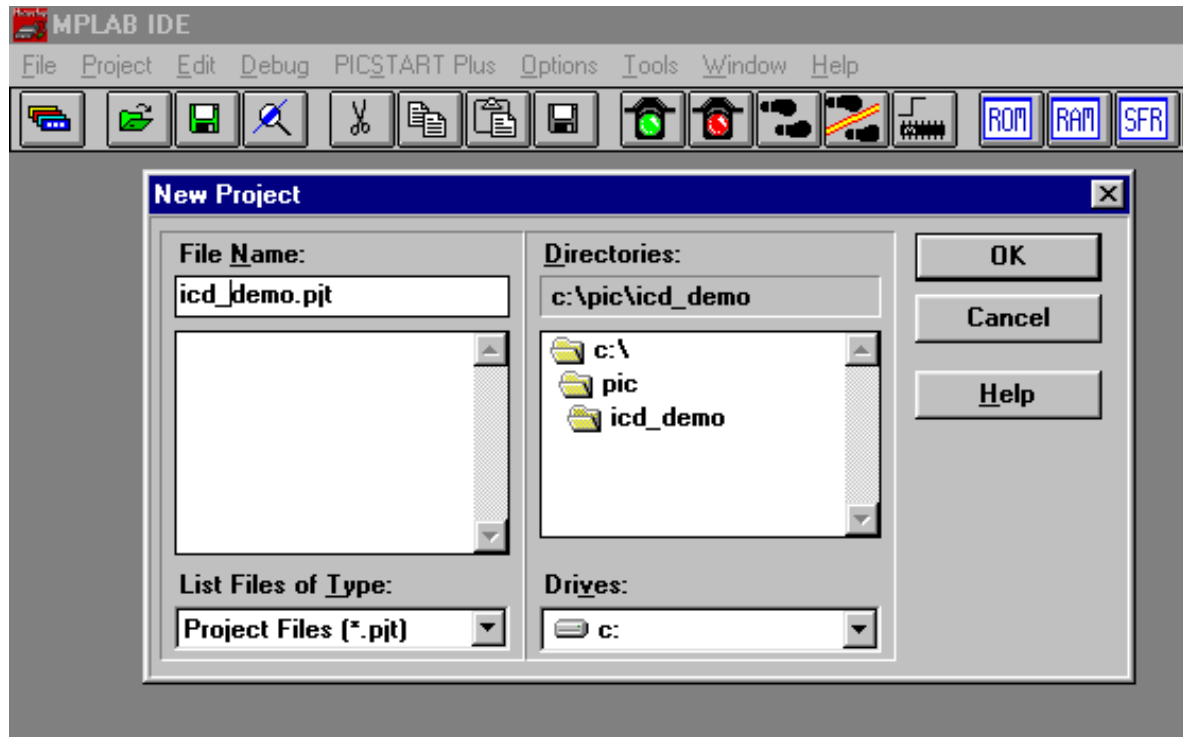


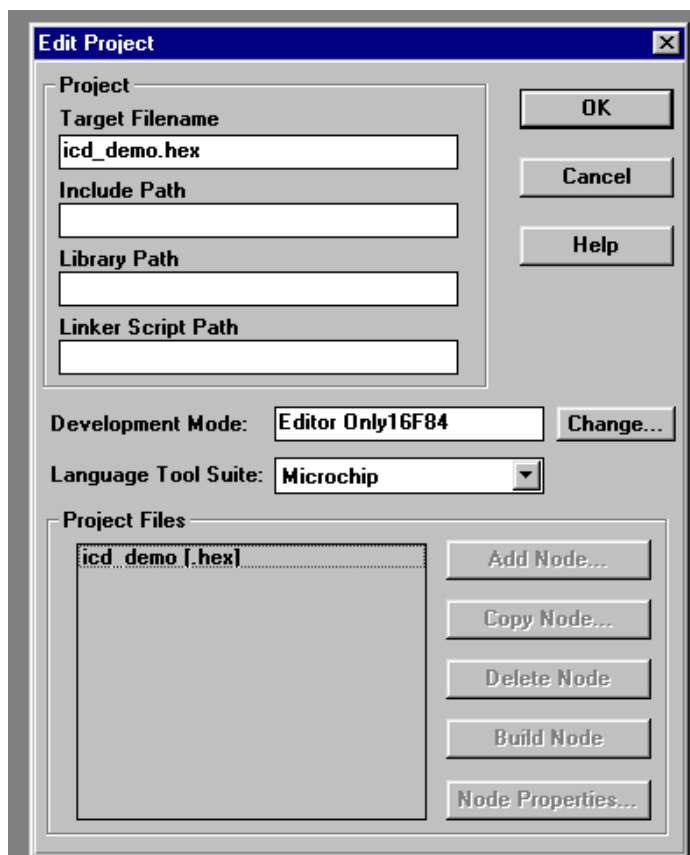
Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

1/ Pour commencer il faut créer un dossier destiné à contenir tous les fichiers du projet. (Cette opération doit pour l'instant être effectuée à l'extérieur de MPLAB car jusqu'à la version 5, MPLAB reste une application windows 16 bits). Pour ma part, le dossier porte le nom du projet qu'il contient. Ensuite Faire Project->New Project et créer le projet dans ce dossier.



2/ La fenêtre des caractéristiques du projet apparaît alors (vous pourrez la rappeler plus tard par Project -> Edit Project)

Target Filename est le nom du fichier hex (format Intel équivalent au S19 Motorola) qui sera chargé dans le pic. Il porte par défaut le nom du projet (et c'est très bien ainsi).



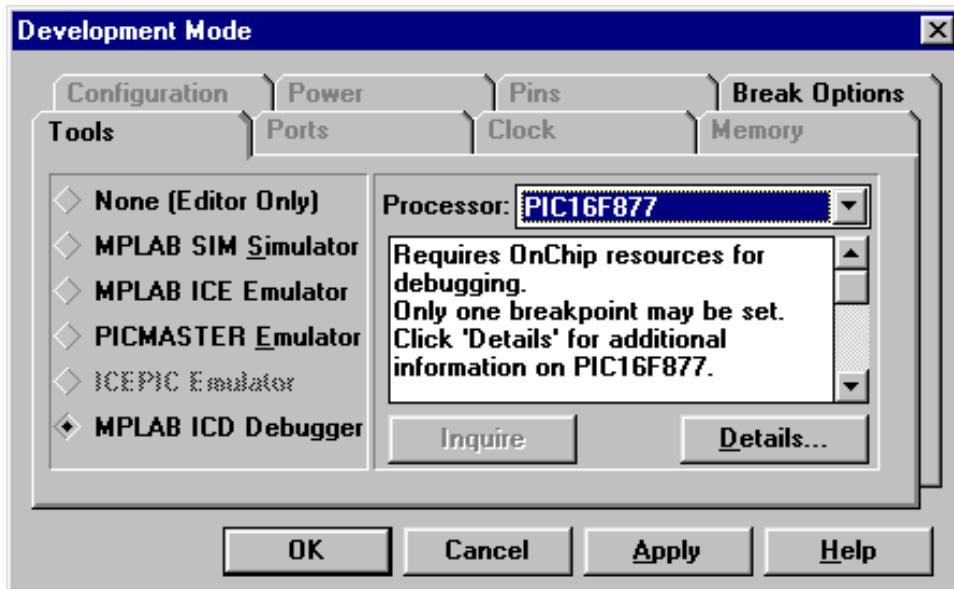
Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

3/ Il faut ensuite choisir le mode de développement et le type de PIC utilisé.

None : Vous pourrez seulement éditer, assembler et compiler pour créer un fichier hex.

MPLAB SIM : Pour développer et simuler un projet.

MPLAB ICD : Pour développer et déboguer avec un ICD (seulement avec des 16F87X)



4/ Si vous avez choisi le mode ICD, en cliquant OK la fenêtre ci-dessous apparaît et MPLAB cherche à communiquer avec l'ICD. Si le dialogue échoue (parce qu'il ICD est hors tension ou non connecté par exemple), vous pourrez par la suite tenter de rétablir la liaison en appuyant sur Reconnect.



5/ Il faut ensuite choisir l'outil de génération de code utilisé. (Language Tool Suite)

Typiquement ce sera :

Microchip pour développer en l'assembleur (tout type de PIC)

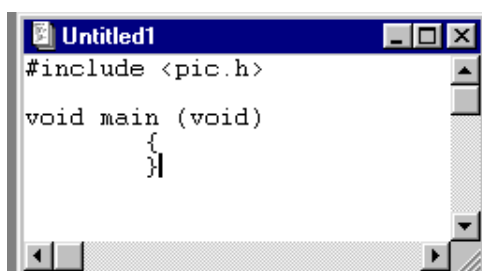
HI-TECH pour développer en C sur les postes où une licence est installée (tout type de PIC 16 ou 12).

HI-TECH PICL pour développer en C sans licence (uniquement pour les 16F84 et 16F627 (donc sans ICD).

Pour cet exemple nous choisissons HI-TECH.

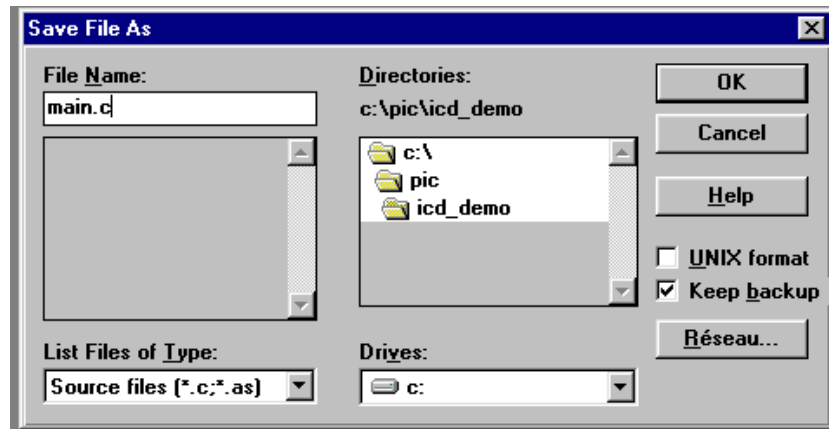
Cliquez sur Ok de la fenêtre Edit Project .

6/ Il faut maintenant créer le fichier source principal de notre projet. Une bonne méthode consiste à créer un fichier minimal (avec un en-tête vierge) qui sera recopié par la suite dans le répertoire de votre projet avant de commencer. Ce fichier minimal pour le C devrait ressembler à ceci (sans l'en-tête) :

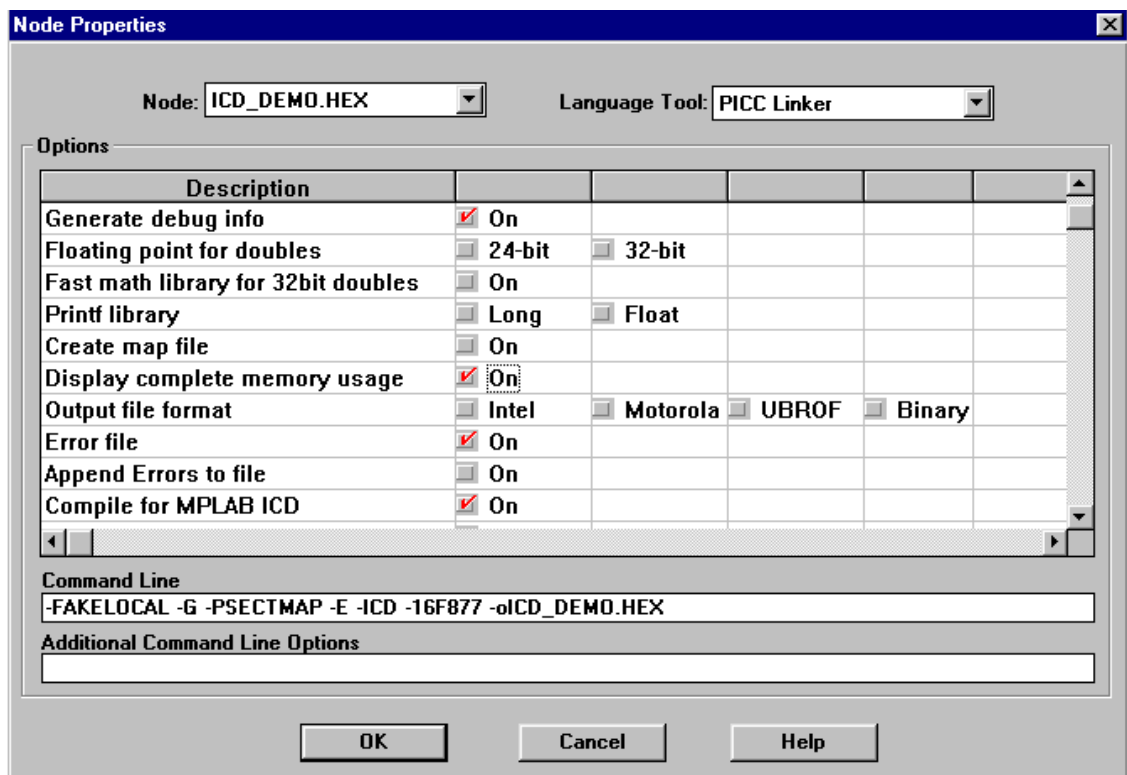


Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

7/ Sauvez le par File->save dans le répertoire du projet. Pour ma part, comme il contient le "main", il s'appelle toujours *main.c*.



8/ Rappeler la fenêtre projet par Project->Edit Project. Sélectionner *icd_demo.pjt* et faire Node Properties... La fenêtre ci-dessous apparaît alors. Le fichier hex est produit par le Linker. Il faut donc sélectionner PIC-C Linker dans la fenêtre Language Tool. Ensuite vous devez cocher les options que vous souhaitez passer à l'éditeur de liens.



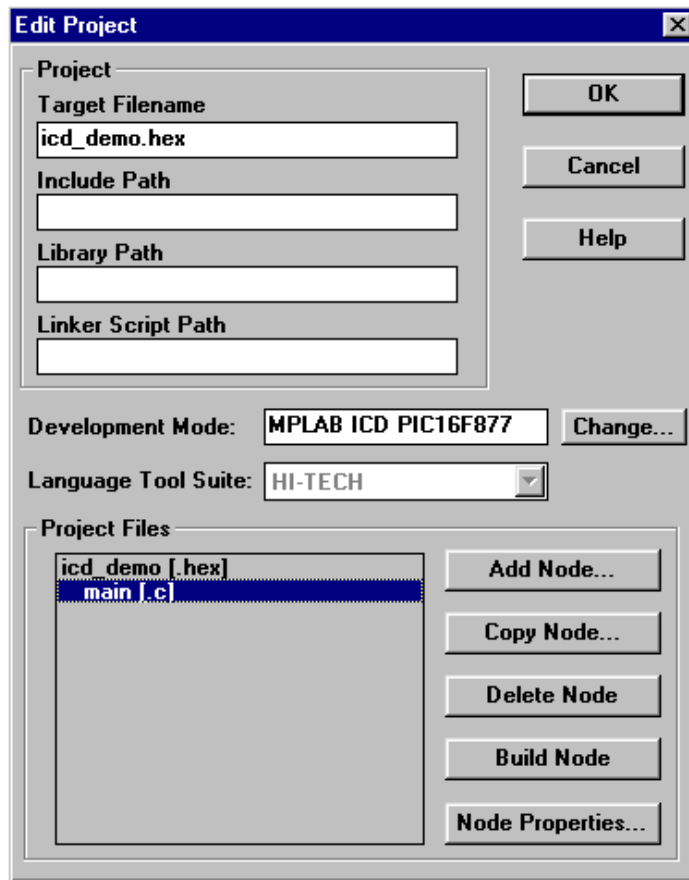
L'option "Generate debug info" est indispensable pour déboguer correctement avec MPLAB.

Pour travailler avec l'ICD, il est important de choisir "Compile for MPLAB-ICD".

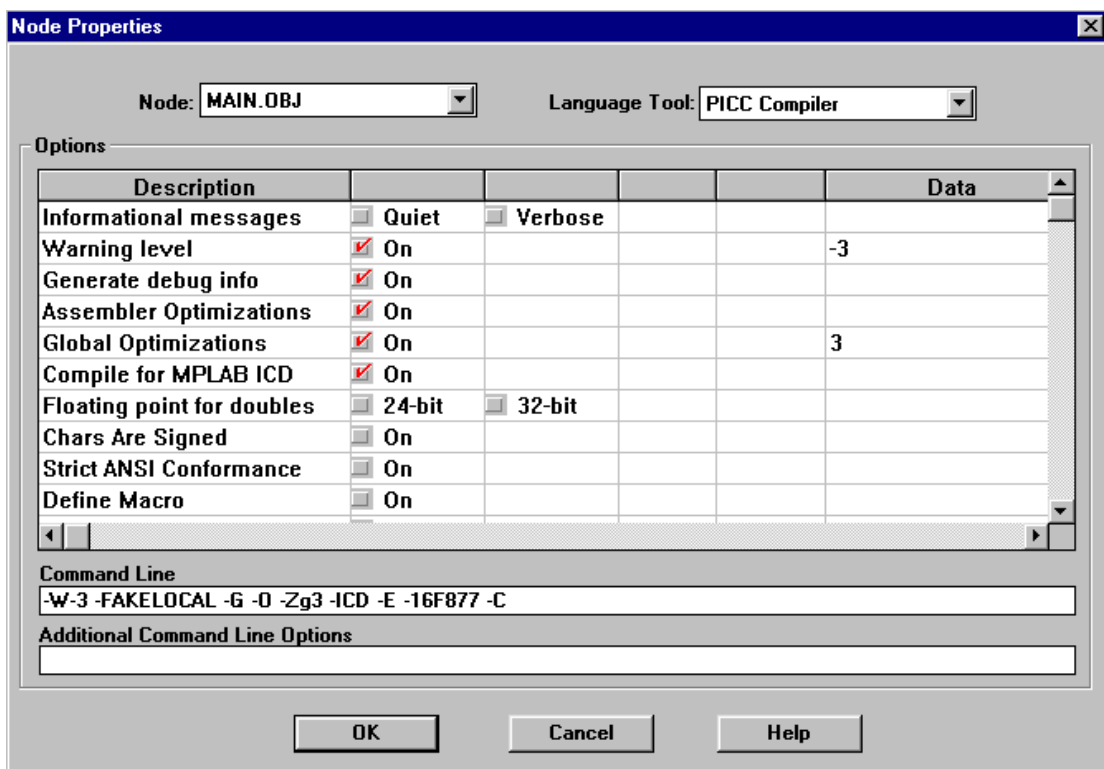
L'option "Display Complete Memory Usage" permet d'afficher les tailles mémoires (données et programme) utilisées lors de chaque génération du programme.

Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

9/ Après avoir fixé les propriétés de l'exécutable, le bouton "Add Node" devient actif. Utilisez ce bouton pour ajouter vos fichiers sources au projet. Ici on ajoute seulement main.c



10/ Il faut ensuite choisir les options de compilation de main.c



Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

Les options importantes sont :

Warning Level : avec le niveau -3 (-W-3) pour rendre les vérifications très complètes.

Generate debug info : indispensable pour déboguer

Compile for MPLAB ICD : indispensable pour utiliser l'ICD.

Pour optimiser le code, vous pouvez cocher les options "Assembler Optimisations" et "Global optimisations" (niveau 3). Le code assembleur généré est alors assez difficile à comprendre.

11/ Vous pouvez maintenant générer le programme par Project->Build All ou en cliquant sur l'icône

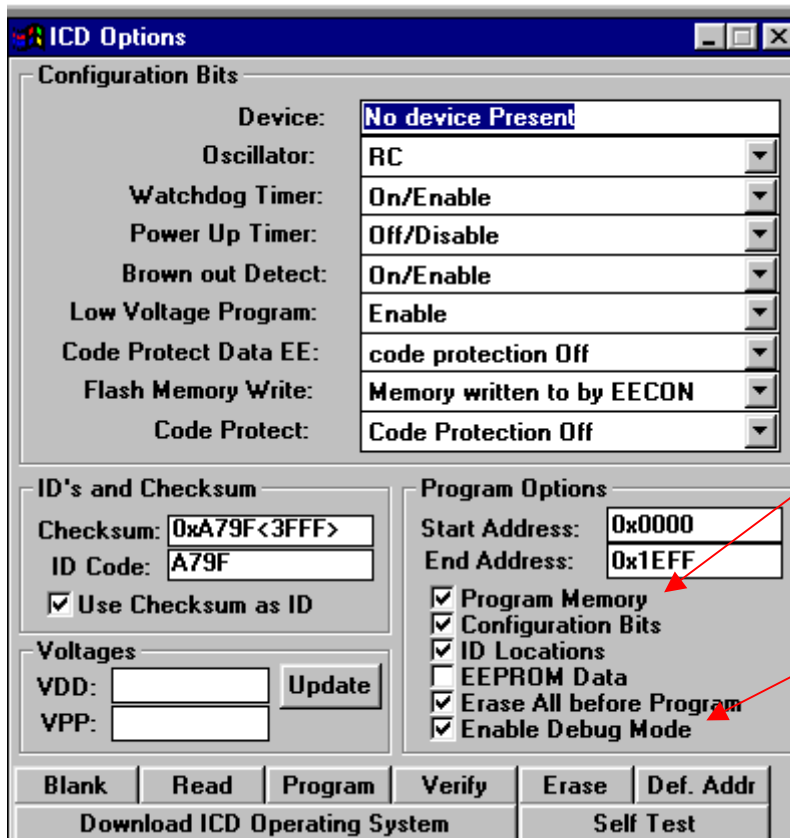


12/ Etoffons un peu notre main.c :

```
/* Ce programme suppose une LED câblée sur RB1 (cathode sur RB1) *
 * et un bouton sur RB2 (imposant un 0 si enfoncé) *
 * il allume la led lorsqu'on appui sur le bouton *
 * et sort en binaire sur le PORTC le nombre d'appuis *
 * le nombre est remis à 0 après 12 appuis */
#include <pic.h>
void main (void)
{
    char Compteur =0 ;
    TRISC = 0 ; /* Port C en sorties */
    TRISB1 = 0 ; /* RB1 en sortie */
    while (1)
    {
        if (RB2==0)
        {
            RB1 = 0 ; /* on allume la led */
            PORTC = ++Compteur ;
            while( RB2==0 ) /* on attend relâchement */
            ;
            RB1 = 1 ; /* on éteind la led */
            if ( Compteur == 12 )
            {
                Compteur = 0 ;
            }
        }
    }
}
```

Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

13/ Après l'avoir compilé, il faut maintenant le charger sur la carte cible et configurer le pic
Cliquez sur le bouton "Options..." de la fenêtre de l'ICD pour faire apparaître l'écran ci-dessous et choisir les bits de configuration en fonction de votre carte et de votre programme.
Pour ce programme qui ne gère pas le watchdog il ne faudra pas oublier de le désactiver.
Le type d'oscillateur doit correspondre à celui câblé sur votre cible.
N'activez pas la protection du code pendant les phases de mise au point.

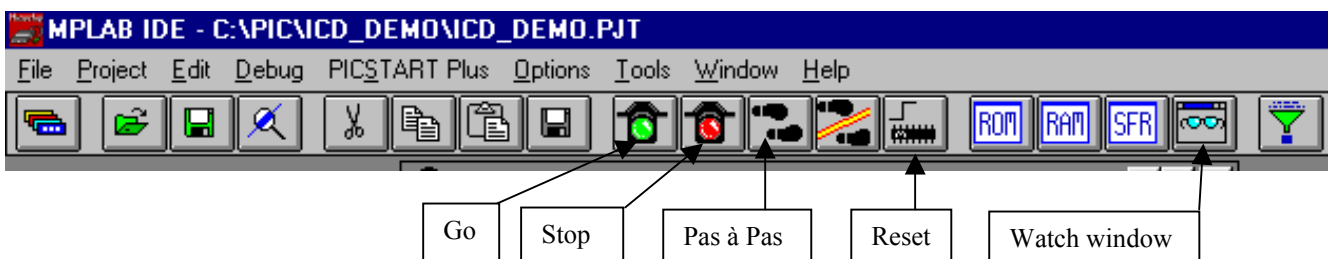


doit être coché pour pouvoir programmer le pic

- doit être coché pour déboguer avec l'ICD
- ne doit pas être coché pour rendre la cible autonome

Chargez le programme dans la carte cible en cliquant sur les boutons "Program" de la fenêtre d'option ou de la fenêtre de l'ICD.

14/ Vous pouvez maintenant lancer votre programme par Debug->Run->Run ou en appuyant sur l'icône du feu vert.



Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

15/ Pour la mise au point vous pouvez créer des fenêtres de visualisation "Watch window" qui vont montrer les variables de votre programme lors de chaque arrêt de celui-ci. Pour cela il faut choisir dans le mode "Minimum & Watch windows" dans la fenêtre de l'ICD (voir 4/).

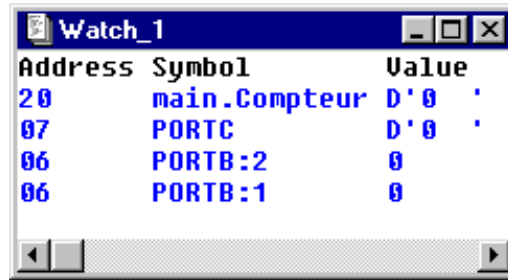
Pour arrêter votre programme, vous pouvez mettre un point d'arrêt en cliquant avec le bouton droit de la souris sur la ligne où vous voulez vous arrêter.

Créez une fenêtre de visualisation pour y mettre la variable Compteur, le PORTC et les bits 1 et 2 du PORTB par la commande Window->Watch windows->New Watch Window ...

Tapez le début de nom d'un symbole pour le faire apparaître puis appuyer sur Add pour l'ajouter à la fenêtre.

Vous pouvez modifier le mode d'affichage par le bouton Properties. En particulier, pour faire apparaître uniquement certains bits choisir Size bit et cocher les bits voulus dans Display bit.

Votre fenêtre devrait ressembler à celle-ci :



Remarquez que la variable Compteur qui est locale à la fonction main est nommée main.Compteur.

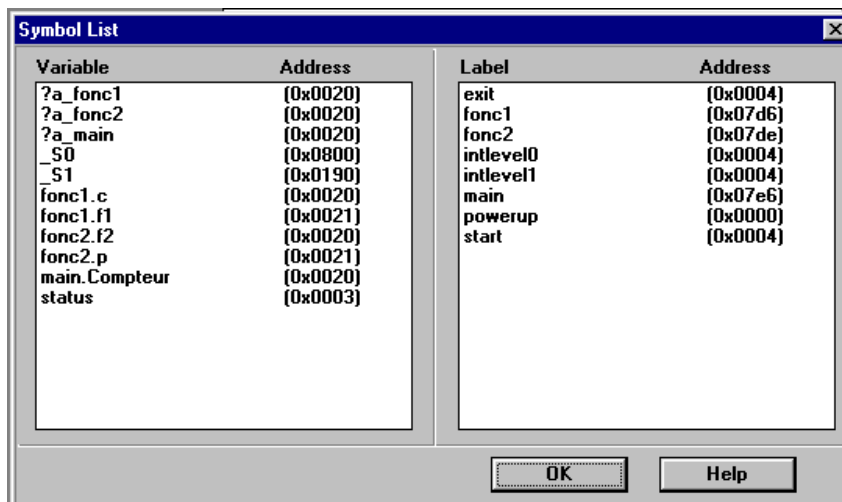
Elle se trouve à l'adresse 0x20 qui est le début de la RAM GP. Elle a une adresse fixe bien qu'elle soit locale !

En fait le PIC n'a pas de pile l'espace mémoire alloué de manière statique aux variables locales peut être réutilisé lors que l'on sort de la fonction. Pour s'en convaincre il suffit de rajouter les deux fonctions ci-dessous à votre programme.

```
char fonc1( char c)
{
    char f1=2;
    return f1 + c ;
}
char fonc2( char p)
{
    char f2=3;
    return f2 + p ;
}
```

Compilez-le et faites Windows->Show Symbol List pour voir tous les symboles :

Vous remarquez que c et f1 de la fonction fonc1 ont les mêmes adresses que f2 et p de fonc2, ce qui montre bien la réutilisation de l'espace mémoire local qui est pourtant alloué statiquement !



Démarrer avec MPLAB, l'ICD et le compilateur C de Hi-Tech

Installation de MPLAB 5.70 avec HT-PICC 7.87PL2

1/ Installer MPLAB 5.70

2/ Installer HT-PICC 7.87PL2

3/ Ensuite NE PAS suivre la procédure indiquée par HITECH sur leur site mais :

Extraire les fichiers `picc_lkr.mtc`, `picc_cmp.mtc` et `picc_asm.mtc` du fichier `htmplab4.exe` et remplacer les fichiers de même nom existants dans le répertoire d'installation de `mplab` par ceux-ci.

4/ Lancer MPLAB et Faire Project->Install Language tool

Pour Language Tool : HI-TECH PICC mettre le compilateur sur le chemin de `PICC.EXE` (qui devrait être `C:\HT-PIC\BIN\PICC.EXE`) en mode "Command-line".

Recommencer avec le linker et l'assembleur en les faisant pointer sur le même exécutable `PICC.EXE`.

Rq : Si `htmplab4.exe` n'est pas sur le cdrom de l'iut, vous pouvez le récupérer sur www.htsoft.com