





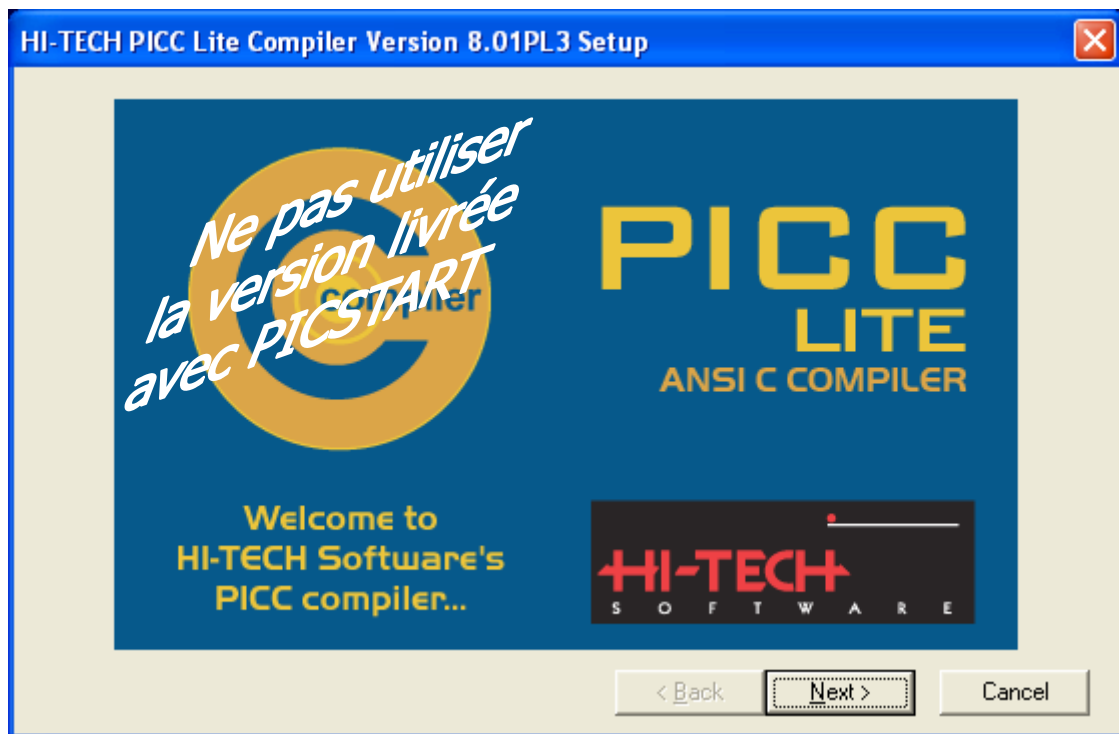
Environnement de développement pour  $\mu$ contrôleur PIC de chez [Microchip](#)



*Note* : Une version de MPLAB se trouve sur le CD de la formation académique 2003 au chemin ...\\M\_CHIP\COMPLEMENT\Mplab613. La dernière version est disponible [ici](#) (v6.30 en juillet 2003)

Cet environnement permet d'éditer un programme en C ANSI. Pour cela il faut associer à l'environnement de MPLAB l'outil « HI-TECH PIC C Lite ». Il s'agit d'un compilateur C (dégradé) proposé par HI-TECH que l'on peut télécharger à l'adresse ci-dessous (v8.01PL3 en juillet 2003).

On installe cette version en conservant si possible les répertoires par défaut.

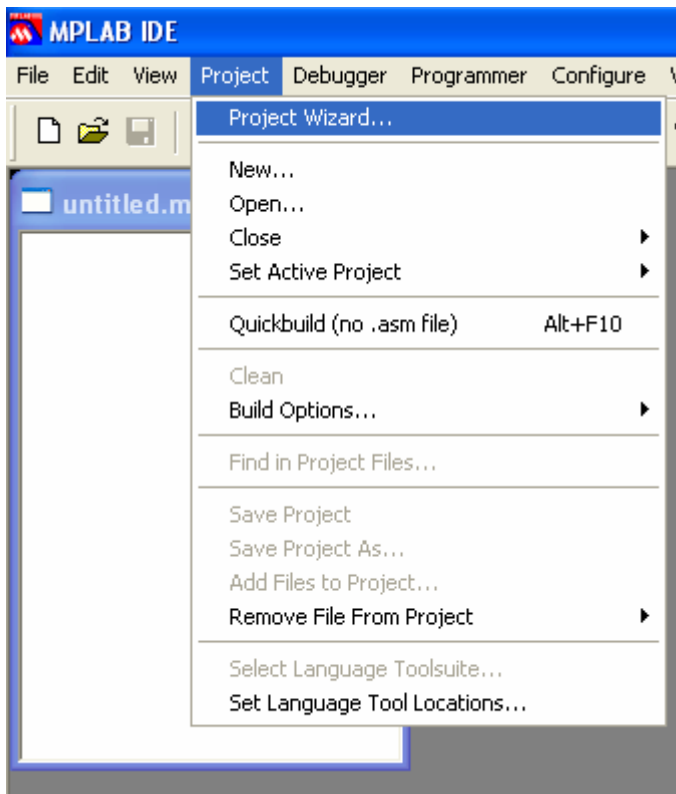


 *Liens :*

<http://www.microchip.com/1010/index.htm>  
<http://www.htsoft.com/products/piclite/>



[Frédéric JOBARD](#)



On débute un projet avec l'assistant de projet...

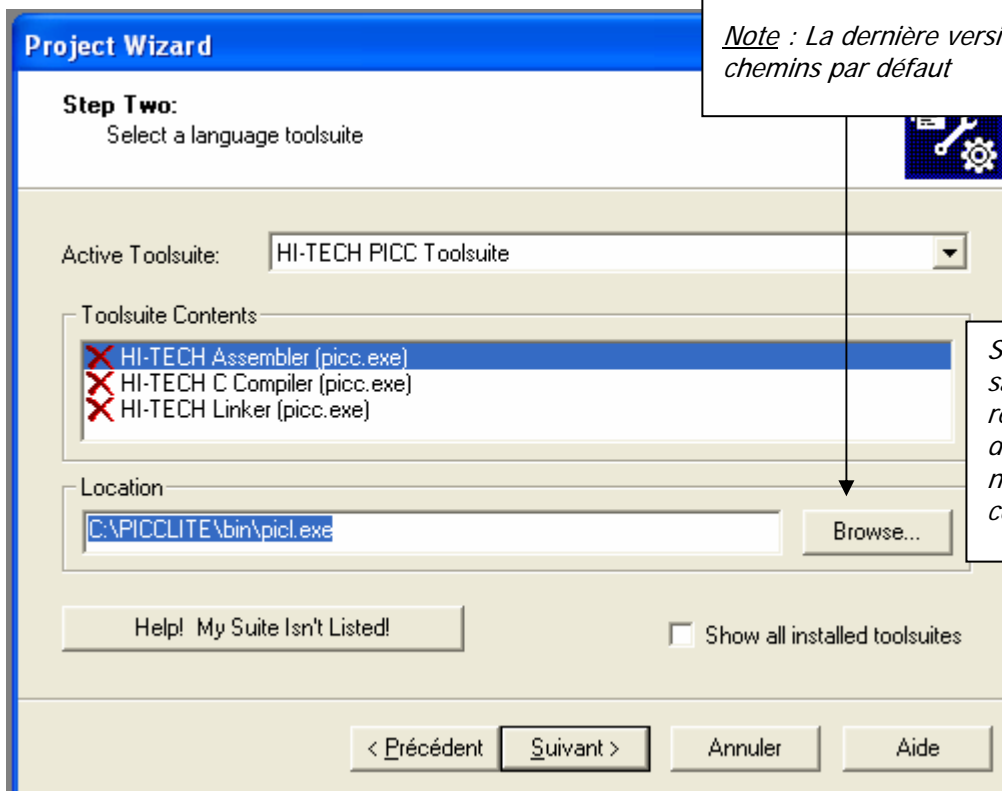
Après le message de bienvenue  
On sélectionne le composant qui convient.  
  
Ici ce sera le PIC16F84

**On active la suite HI-TECH PICC...  
pour une édition en « C »**  
- Pour **chaque outil** on précise  
le chemin **C:\PICCLITE\bin\picl.exe**

**OU BIEN**

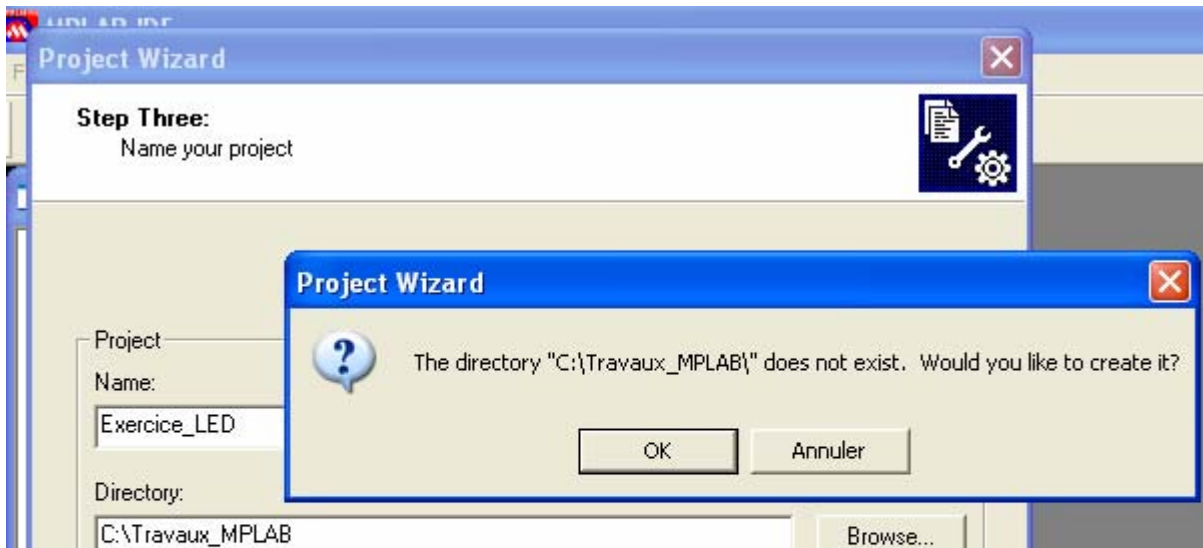
**On active la suite Microchip MPASM  
Toolsuite  
pour une édition en « assembleur »**  
- à **MPASM Assembler** on indique  
le chemin **C:\Program Files MPLAB  
IDE\MCHIP\_Tools\mpasmwin.exe**  
- à **MPLINK Object Linker** on indique  
le chemin **C:\Program Files MPLAB  
IDE\MCHIP\_Tools\mplink.exe**

*Note : La dernière version de MPLAB intègre les  
chemins par défaut*

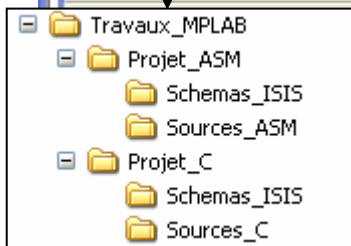


Si vous avez installé PICC Lite  
sans modifier le nom du  
répertoire d'installation par  
défaut les croix rouges  
n'apparaissent pas car MPLAB  
connaît le chemin.

- Puis on nomme le projet, etc...

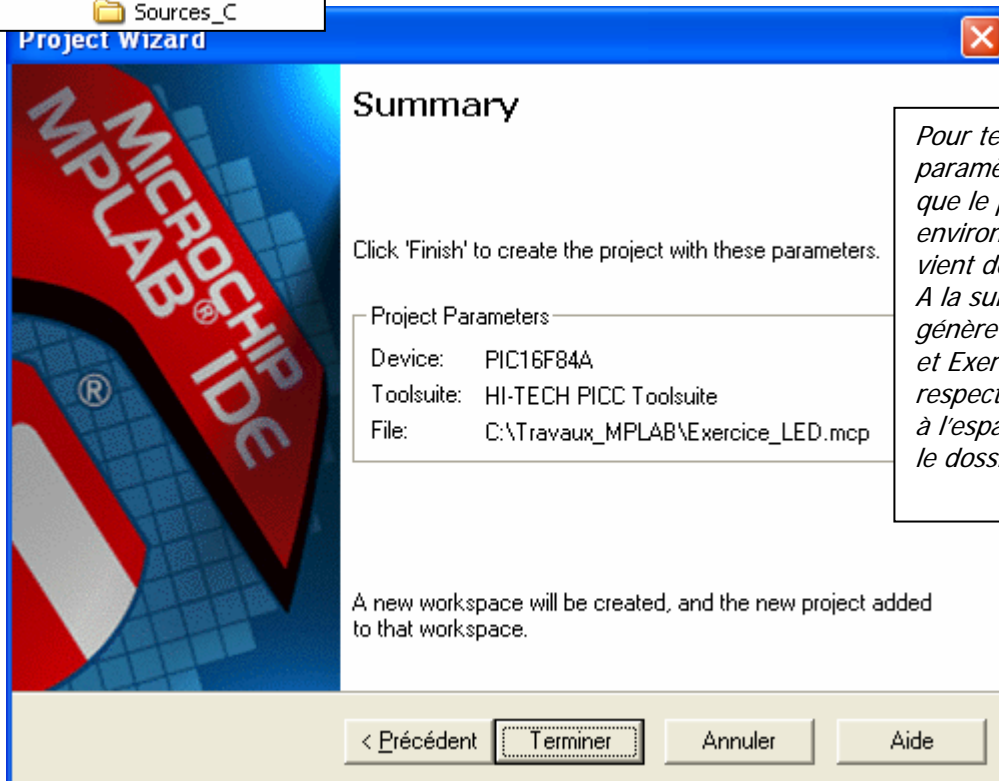


Arborescence conseillée...



*Dans l'exemple le projet se nomme Exercice\_LED et le dossier cible est Travaux\_MPLAB si il n'a pas été préalablement créer l'assistant de MPLAB le crée dans le lecteur spécifié.*

*A l'étape qui suit MPLAB demande les fichiers que vous désirez intégrer au projet en cours de construction. Il suffit alors de trouver le(s) fichier(s), en développant l'arborescence correcte, de l'incorporer par Add puis de cocher la case associée pour en faire une copie dans le répertoire Travaux\_MPLAB.*



*Pour terminer un résumé rappelle les paramètres du projet et nous indique que le projet est intégré au nouvel environnement de travail que l'on vient de créer.*

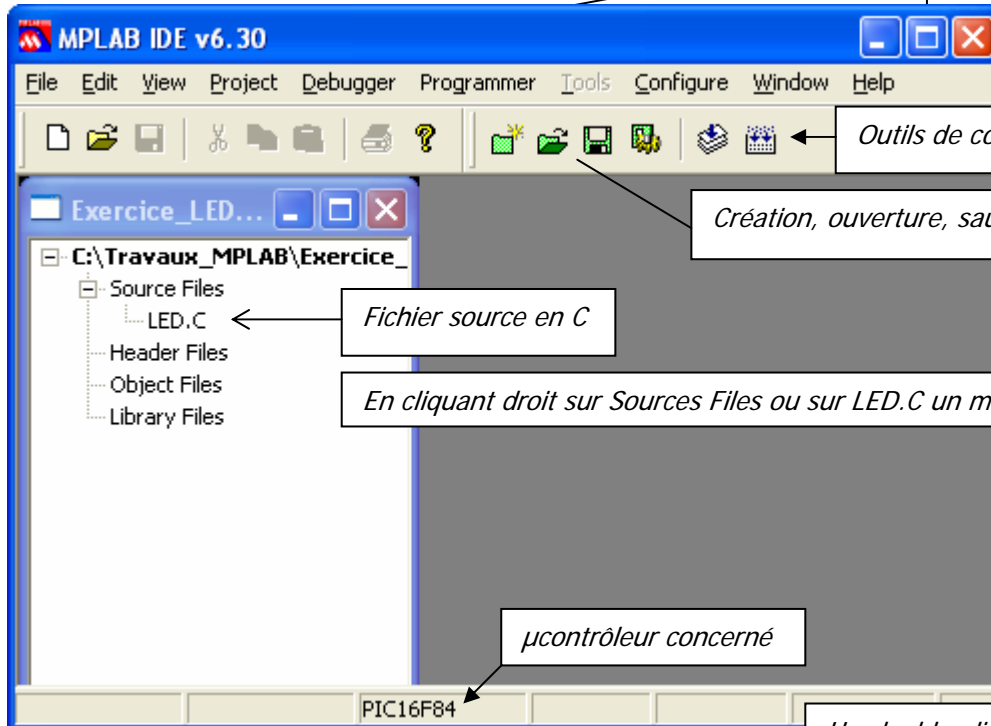
*A la suite de cette étape MPLAB génère 2 fichiers, Exercice\_LED.mcp et Exercice\_LED.mcw qui sont respectivement associés au projet et à l'espace de travail, qu'il place dans le dossier Travaux\_MPLAB*



A ce stade le répertoire de travail peut contenir 3 fichiers :

Exemple : nomduprojet.mcp      Fichier projet  
          nomduprojet.mcw      Fichier espace de travail  
          nomdusource.c        Fichier source en C

Dans le menu View on peut cocher ou décocher l'apparition des fenêtres Project et/ou Output



Outils de compilation

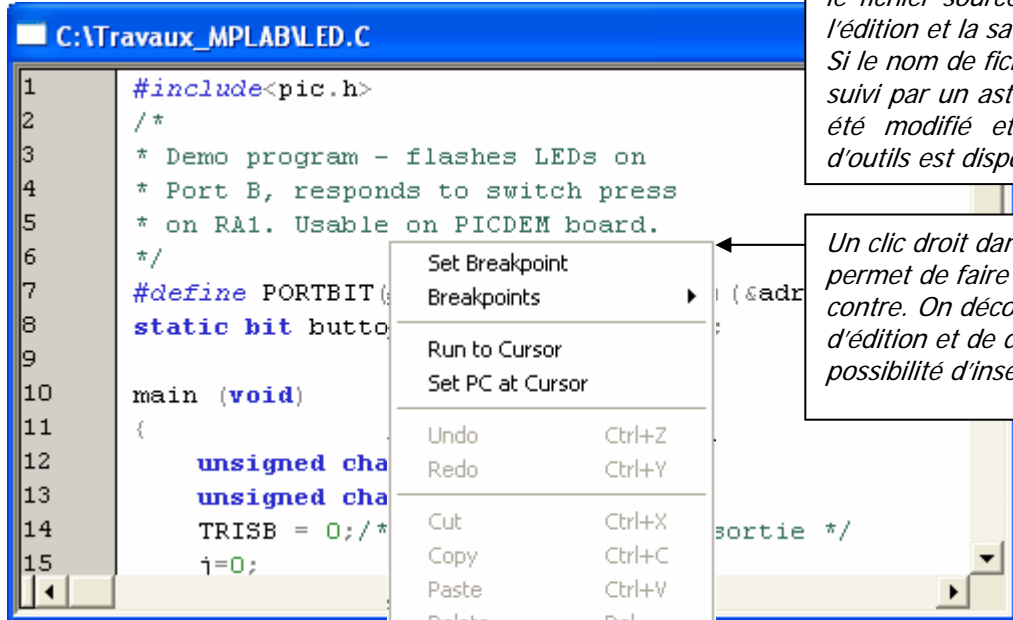
Création, ouverture, sauvegarde de projet

Fichier source en C

En cliquant droit sur Sources Files ou sur LED.C un menu apparaît

µcontrôleur concerné

Un double clic dans la fenêtre du projet sur le fichier source l'ouvre. Dans cette fenêtre l'édition et la sauvegarde sont possible. Si le nom de fichier dans la barre de titre est suivi par un astérisque c'est que le source a été modifié et la disquette de la barre d'outils est disponible pour enregistrer...



Un clic droit dans la fenêtre du fichier source permet de faire apparaître le menu ci-contre. On découvre des possibilités d'édition et de debuggage telle que la possibilité d'insérer un point d'arrêt. Etc...



La compilation peut réussir  
(BUILD SUCCEEDED)  
Elle peut aussi échouer...  
(BUILD FAILED)

Avant de lancer la compilation il faut s'assurer que les options déclarées ci-contre sont correctes...  
FAIRE OK pour fermer la fenêtre ci-contre en conservant les propriétés par défaut des autres onglets.  
Lancer la compilation...

Elle devrait réussir et créer des fichiers du type suivant

nomduprojet.cod  
nomduprojet.ide  
nomduprojet.sym  
nomduprojet.hex

voire

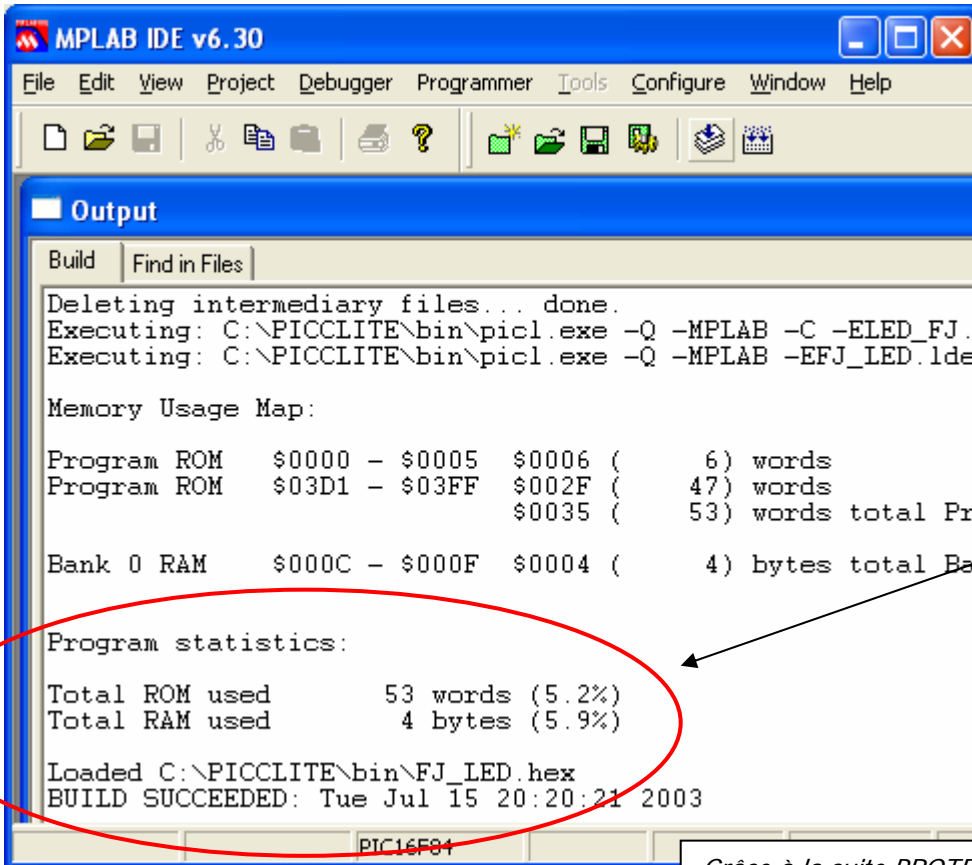
nomduprojet.map

dans le dossier :  
Qui est déclaré à la rubrique **Output Directory, \$(BINDIR):**

dans l'onglet General

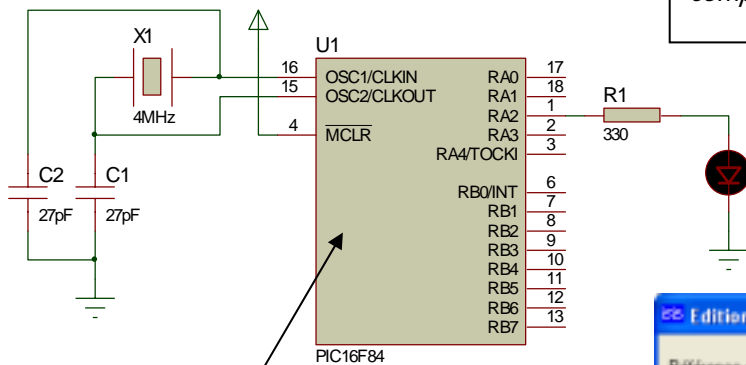
Ici on indiquera le chemin :  
C:\Travaux\_MPLAB\Projet\_C  
Montrer le chemin par le bouton Browse...  
Ne pas faire de copier coller.

Outre les fichiers ci-dessus la compilation génère les fichiers nomdusource.lst, nomdusource.cce, nomdusource.obj et nomdusource Package Appfix, dans le dossier :  
Qui est déclaré à la rubrique **Intermediates Directory, \$(TMPDIR):**

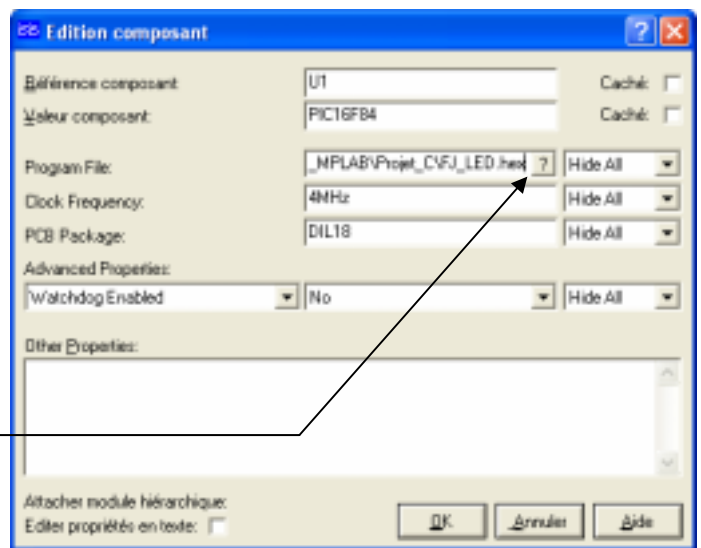


La fenêtre « Output » nous renseigne sur l'occupation et la taille du programme ainsi que sur le résultat de la compilation.

Grâce à la suite PROTEUS on peut éditer un schéma avec ISIS et créer une animation (simulation) qui tourne avec le fichier nomdusource.hex issu de la compilation du programme source.

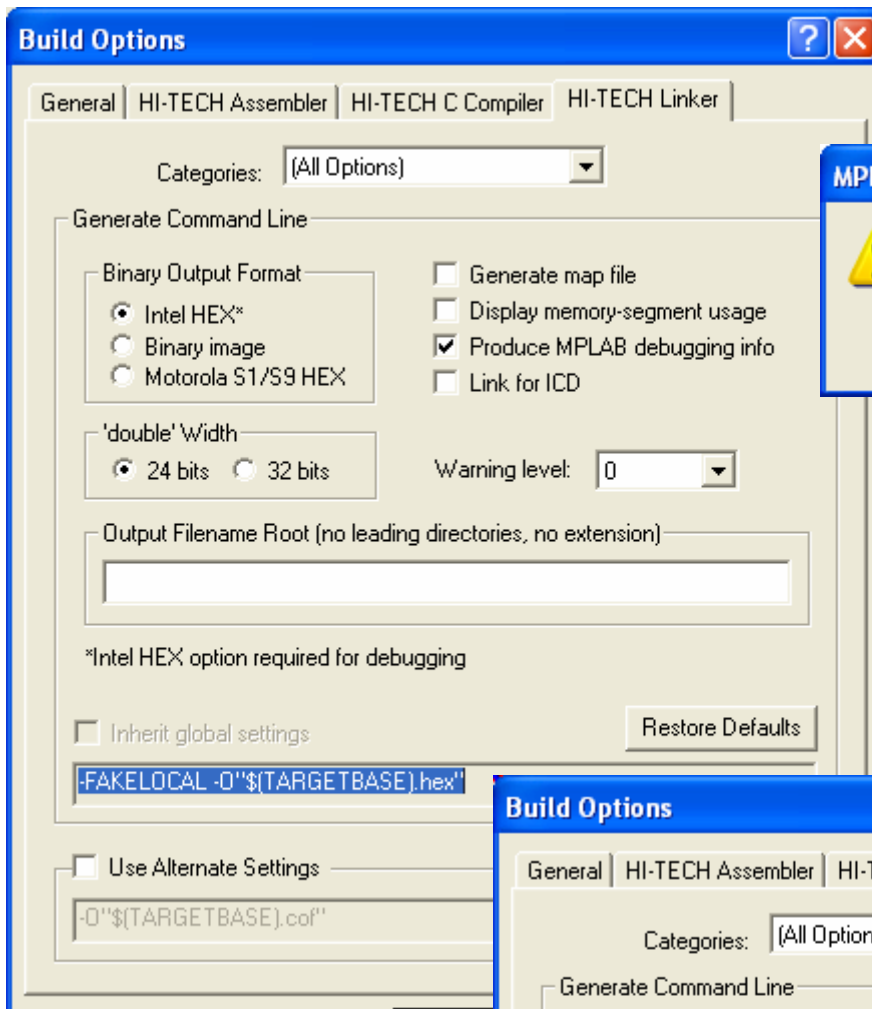


Cette animation fonctionne avec un fichier .hex obtenu par la compilation d'un fichier source édité en C sous MPLAB. Le compilateur C utilisé est celui de HI-TECH dans sa version LITE.

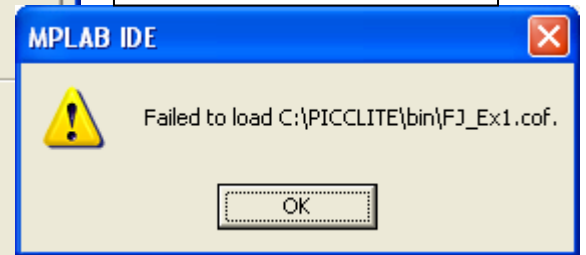


Un clic droit sur le composant programmable fait apparaître la fenêtre ci-contre

Un appui sur le ? permet de montrer le chemin du fichier .hex



La compilation peut hélas échouer...  
Notamment si Isis est ouvert et que le .hex est utilisé

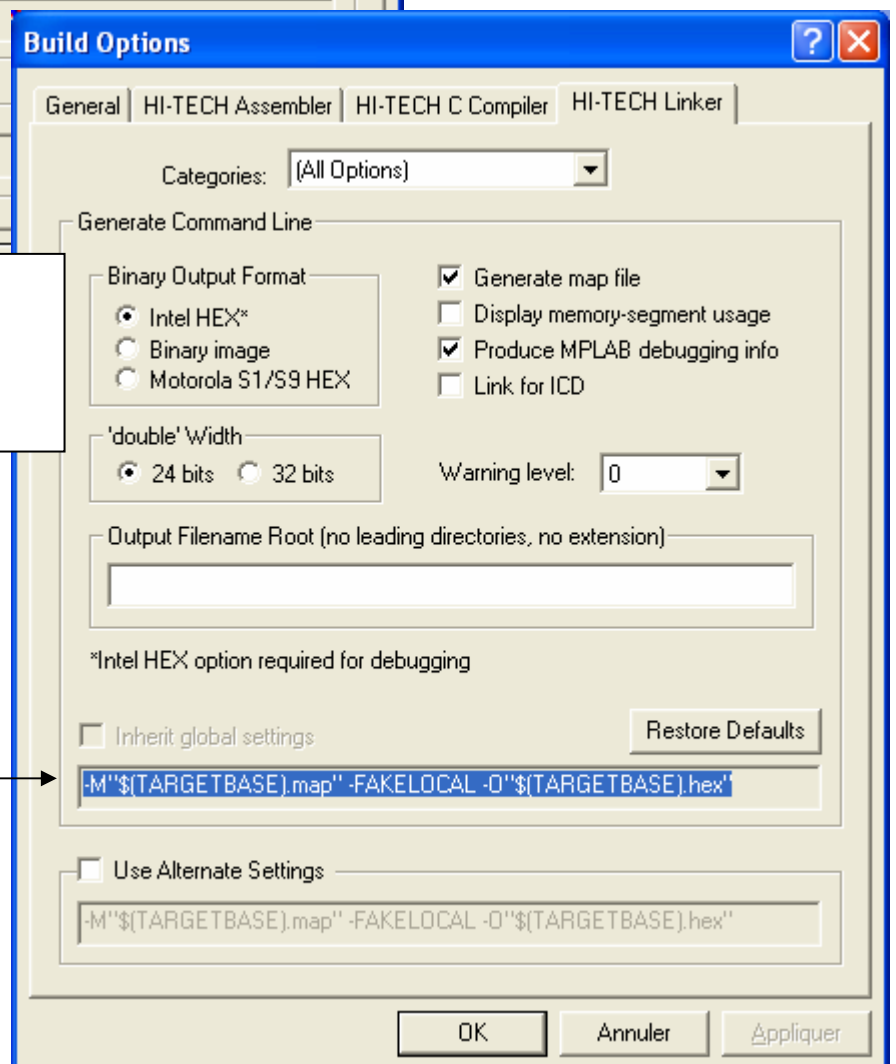


En cochant la case « Generate map file » et en relançant la compilation, celle-ci peut réussir.  
On pourra aussi réessayer en restaurant les propriétés par défaut.

**REMARQUE :**  
La case à cocher  
»Produce MPLAB debugging info »  
Intègre la directive  
-FAKELOCAL

Si aucune case n'est cochée à l'ouverture on a :  
**-O"\${TARGETBASE}.hex"**  
or si on appuie sur le bouton  
« Restore Defaults » on a :  
**-O"\${TARGETBASE}.cof"**

Une fois que la compilation a réussie on peut décocher les cases puis relancer la compilation cela fonctionne...





```

/*****
*      Clignotement LED sur platine BIGONOFF
*      Exercice 1 de source en C
*      Frédéric JOBARD
*****/
#include      <pic1684.h>
#define XTAL_FREQ 4          /*fréquence en MHz (4 MHz)*/
/*****/
void Delay250Us(void);      // déclaration des fonctions écrites
void DelayMs(unsigned int cnt); // après le programme principal
/*****/
/* PROGRAMME PRINCIPAL
*****/
void main(void)
{
    TRISA2 = 0;              //Bit 2 de TRISA en sortie
    while (1>0)             //Boucle sans fin
    {
        RA2 = 1;            //Bit 2 de PORTA à 1
        DelayMs(250);       // appel temporisation avec valeur
        RA2 = 0;            //Bit 2 de PORTA à 0
        DelayMs(100);       // appel temporisation avec valeur
    }
}

/*****/
/* Temporisations*/
/*****/

//délais de 250µs et X fois 250µs
//ATTENTION compiler en 'full optimization' pour obtenir
//l'exécution d'une boucle en 3 cycles d'horloge

void Delay250Us(void)
{
    unsigned char _dcnt;
    _dcnt = 250/(12/XTAL_FREQ);
    while(--_dcnt) continue;
}

void DelayMs(unsigned int cnt)
{
    unsigned char i;
    do
    {
        i=4;
        do
        {
            Delay250Us();
        }
        while(--i);
    }
    while(--cnt);
}

/*****
//
//      FIN
*****/

```

*Ci-contre le source en C du programme de clignotement d'une LED.*

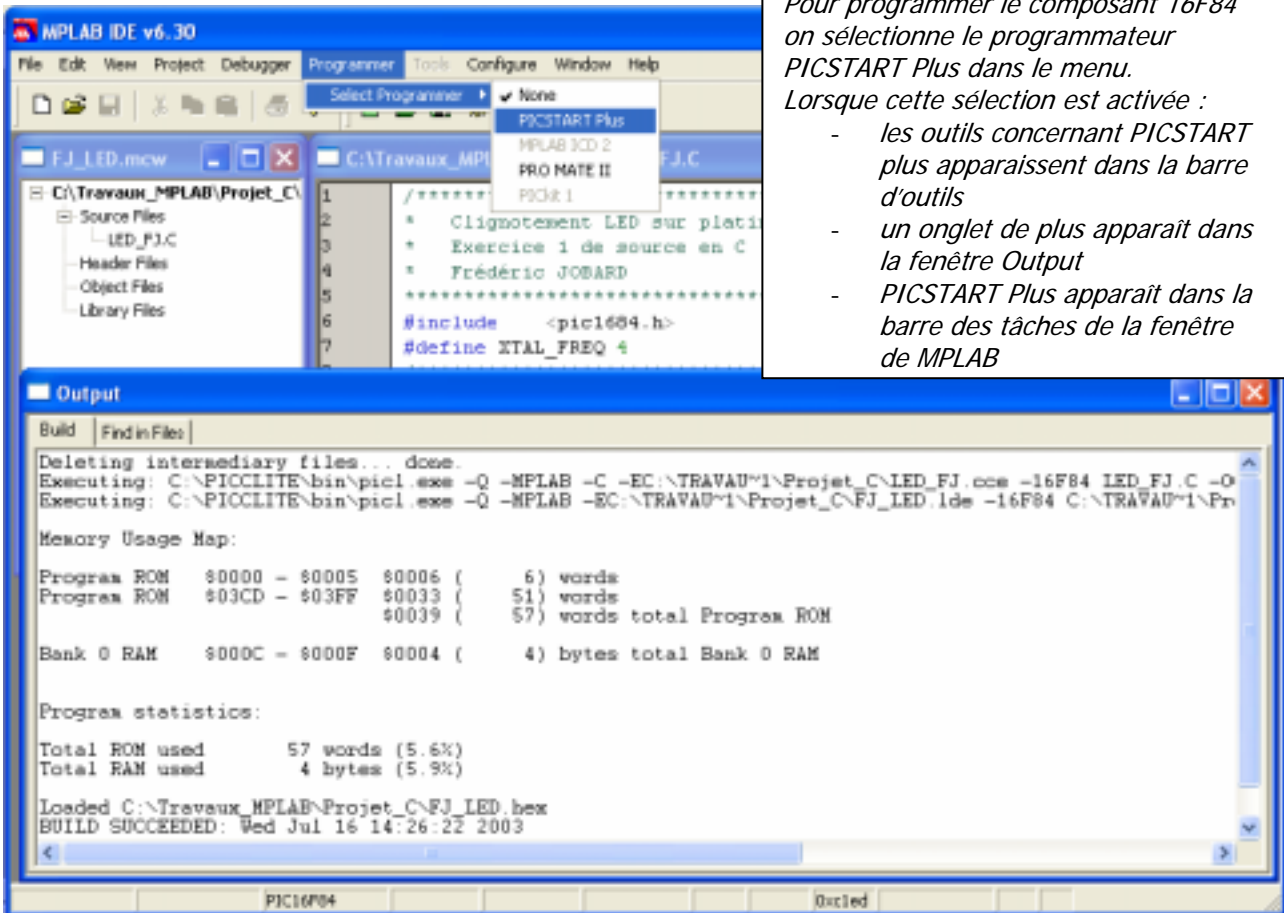
*Cet exercice est inspiré des travaux diffusés sur le Net par [BIGONOFF](http://www.abcelectronique.com/bigonoff/index.php)*

*<http://www.abcelectronique.com/bigonoff/index.php>*

*Le source de cet exercice est inspiré des travaux diffusés également sur le Net par [Claude DRESCHEL](http://microcontrolleur.fr.st/).*

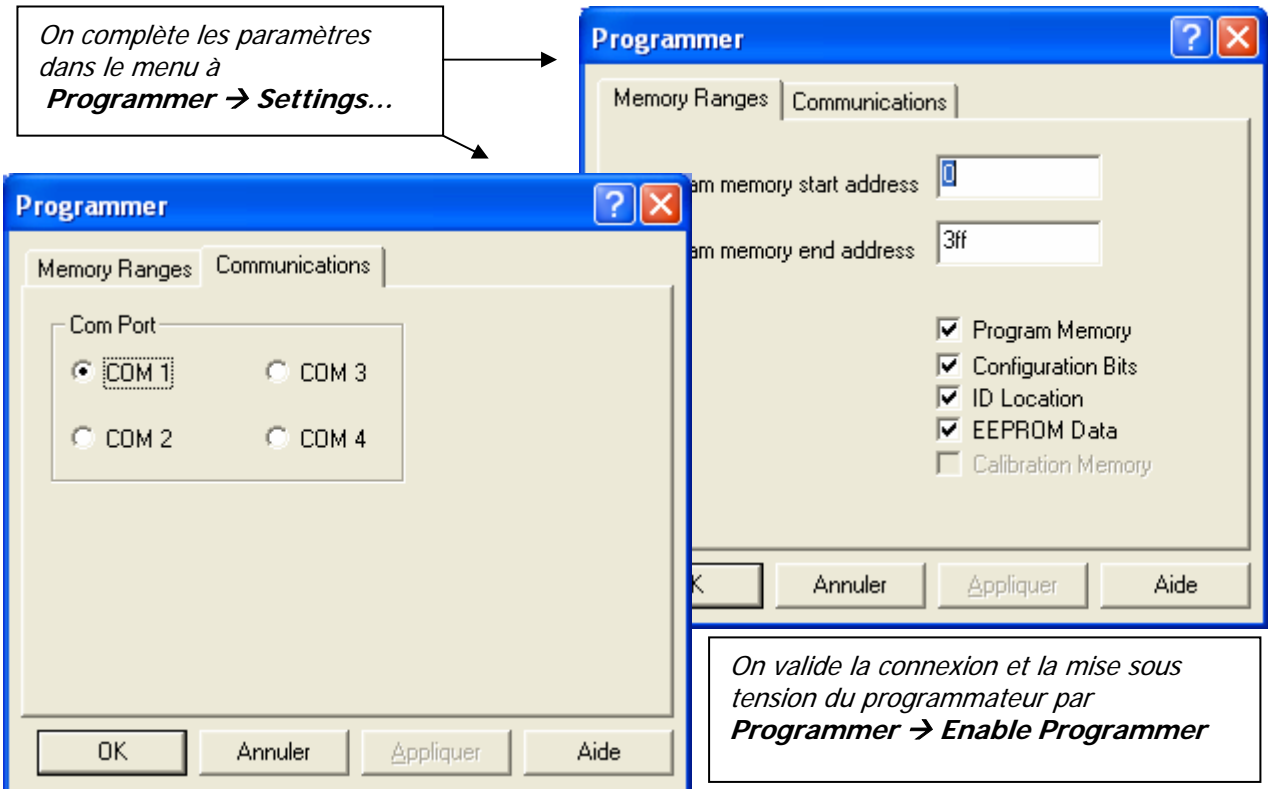
*<http://microcontrolleur.fr.st/>*

*Le debuggage d'un programme édité en C n'est pas aisé avec MPLAB. Il sera utile d'utiliser le compilateur HI-TECH sous son environnement DOS. On le trouve au chemin C:\PICCLITE\bin\htlpic.exe*



On connecte physiquement sur un port série du PC le programmeur PICSTART Plus. Alimentation éteinte, on place le PIC sur le support dans la partie « 18 broches ». On met sous tension (Voyant POWER allumé).

On complète les paramètres dans le menu à **Programmer → Settings...**





**Un clic droit sur la fenêtre PICSTART puis Clear pour effacer.**

Dans l'onglet PICSTART de la fenêtre Output des messages vous informent...

Traductions...


A newer version of the PICSTART Plus firmware operating system is available, and may be required for programming some devices. The newer version is located in your MPLAB IDE main directory. Refer to the topic, 'Upgrading the PICSTART Plus Operating System' in the PICSTART Plus Help file for instructions.

MPLAB IDE main directory. Refer to the topic,

Une version plus récente du PICSTART Plus le logiciel d'exploitation est disponible, elle peut être nécessaire pour programmer certains composants. Une version plus récente est située dans le répertoire MPLAB IDE. Mettez à jour le logiciel d'exploitation de votre PICSTART Plus. Référez-vous au fichier d'aide pour plus d'informations.

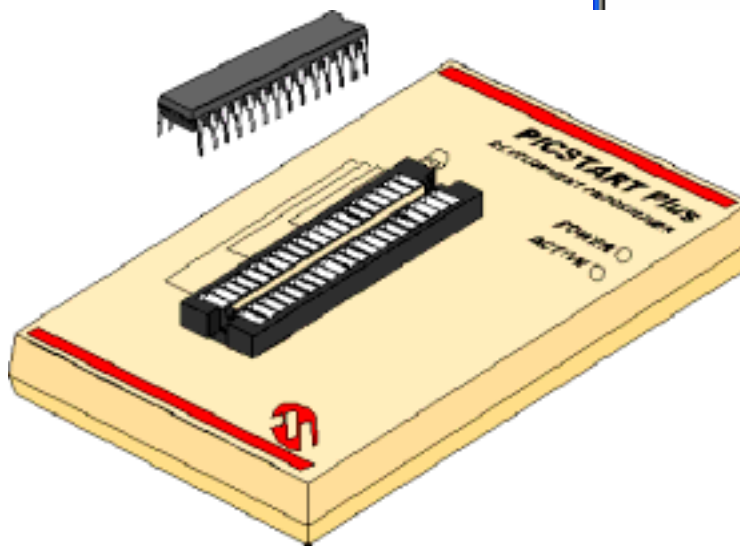
Cannot Transmit. Please verify connection and retry the previous operation.

Ne peut pas Transmettre. Veuillez vérifier le raccordement et réessayez l'opération précédente.

 Avant la compilation et la programmation il faut vérifier l'état du registre de configuration. Si le setting des bits n'est pas celui qui convient il faudra reconfigurer ce registre en passant par le menu Configure → Configurations Bits...

Address	Value	Category	Setting
2007	3FFF	Oscillator	RC
		Watchdog Timer	On
		Power Up Timer	Off
		Code Protect	Off

Address	Value	Category	Setting
2007	3FF9	Oscillator	XT
		Watchdog Timer	Off
		Power Up Timer	Off
		Code Protect	Off



On lance la programmation du PIC par **Programmer → Program**

Ou Par l'icône de la barre d'outil

A ce moment là le téléchargement débute... Une animation apparaît en bas à gauche de la fenêtre de MPLAB. Le voyant jaune « ACTIVE » situé sur le programmeur PICSTART Plus s'allume pendant la durée du transfert.

Un message nous informe que la programmation a réussi. Et que la vérification a été accompli avec succès.

Le composant est programmé. On peut l'insérer sur son application et procéder aux essais.

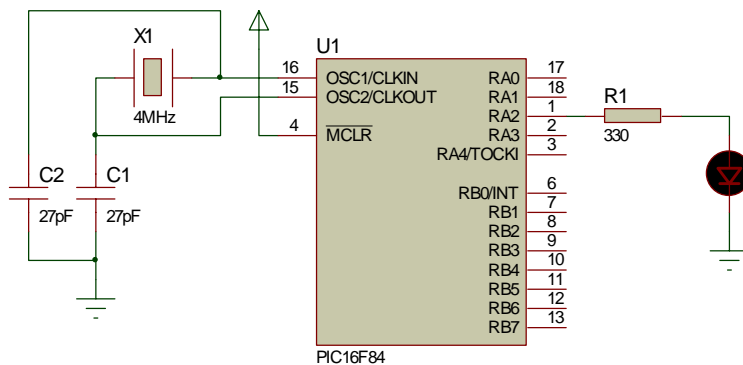


En ce qui concerne l'édition d'un programme en assembleur on peut utiliser l'Assistant de MPLAB comme précédemment à la condition de spécifier les outils suivants :

- MPASM Assembler (mpasmwin.exe)
- MPLINK Object Linker (mplink.exe)

J'ai expérimenté cette partie, en reprenant un exercice de BIGONOFF Led\_cli.asm. Ce programme fait clignoter une LED connectée sur la ligne de port PA2.

Après compilation on peut tester le simulateur livré avec MPLAB en visualisant une fenêtre watch. La compilation génère quelques fichiers dont Led\_cli.hex. J'ai connecté PICSTART Plus et la programmation s'est bien passée. J'ai placé le pic sur la platine et cela a marché...



Pour que la simulation fonctionne il faut mettre en commentaire la ligne ci-dessous du programme source puis compiler et animer.  
\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_PWRTE\_ON & \_HS\_OSC

Enfin je me suis servi de cet exercice pour tester une animation sous ISIS.

Cela a marché à partir du moment où j'ai fait ignorer la ligne de programme concernant le registre spécial de programmation

```
; __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _HS_OSC
```

; ' \_\_CONFIG' précise les paramètres encodés dans le processeur au moment de

; la programmation du processeur. Les définitions sont dans le fichier include.