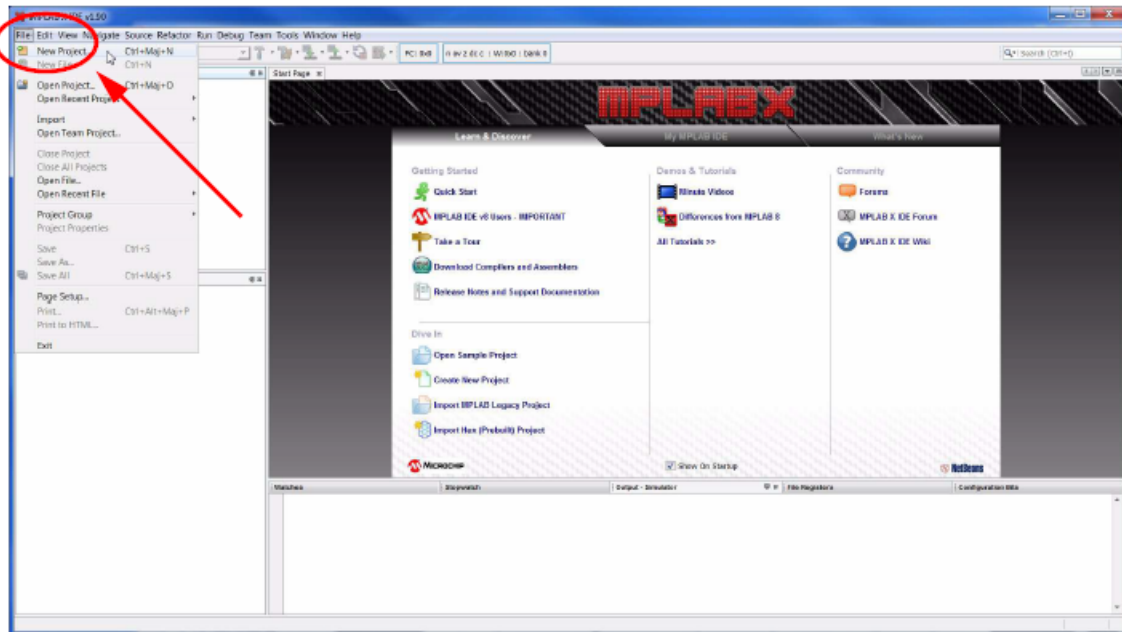


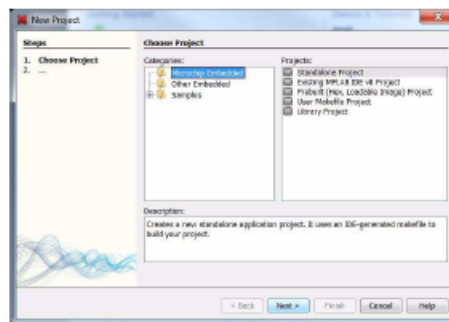
Annexe- Utilisation du logiciel MPLAB X

1 Création d'un projet sous MPLAB X

En démarrant MPLAB X, commencer par créer un nouveau projet : **File → New Project...**



Dans l'étape **Choose Project**, garder les choix par défaut
(Catégories : Microchip Embedded ; Projects : Standalone Project),
puis sélectionner **Next>**.

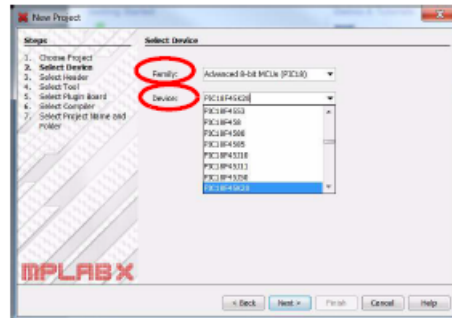


Annexe - Utilisation du logiciel MPLAB X

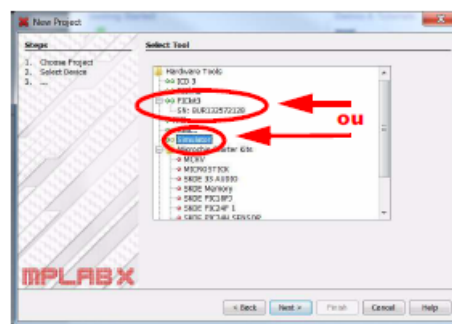
Dans l'étape **Select Device**, sélectionner

Family : **Advanced 8-bit MCUs (PIC18)**

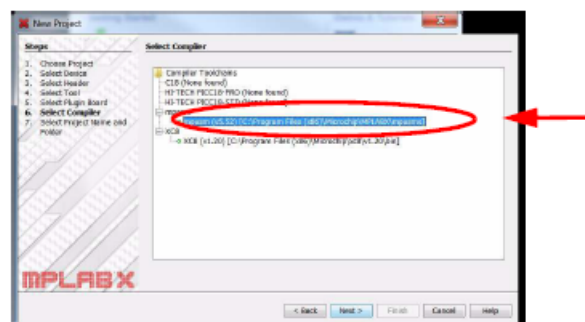
Device : **PIC18F46K22** (ou toute autre référence, en fonction du composant qui sera utilisé)
puis sélectionner **Next>**.



Dans l'étape **Select Tool**, sélectionner **PICKit3** pour débbuger et faire tourner un programme sur un microcontrôleur réel (une carte de prototypage doit dans ce cas être connectée au PC)
puis sélectionner **Next>**.



Dans l'étape **Select Compiler**, sélectionner **XC8 (v2.05)**
puis sélectionner **Next>**.

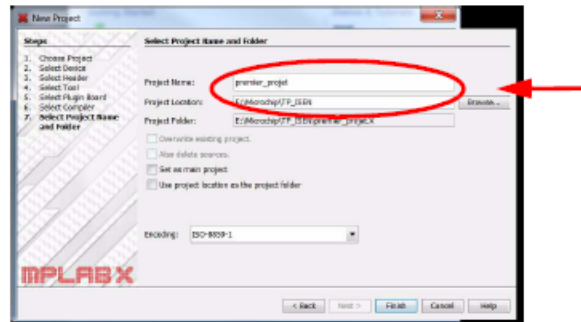


Annexe - Utilisation du logiciel MPLAB X

Dans l'étape **Select Project Name and Folder**, indiquer le nom du projet ainsi que son emplacement (un sous-répertoire xxxxxx.X sera automatiquement créé)

puis cocher la case **Set as main project**
et sélectionner **Next**>.

Remarque importante : pour les noms de projets, répertoires ou fichiers, n'utiliser que les caractères alphanumériques (pas d'espace, d'accent, de &, #, etc...)



2 Conception de projet

Après avoir créé un projet, il faut inclure les fichiers d'entête *.h, les fichiers sources *.c et concevoir le programme principal *main.c*.

Dans la fenêtre **Projet**, développer **Header Files** puis **New/C Header File**. Donner un nom à ce nouveau fichier, sans l'extension .h et cliquer sur **OK**.

Au minimum, il faut inclure le fichier *general.h*. Si nécessaire, on inclura d'autres fichiers, comme *leds.h*, *lcd.h* et *i2c.h*.

Si l'on utilise les fichiers *lcd.h* et/ou *i2c.h*, il faudra également inclure *lcd.c* et/ou *i2c.c* en développant **Sources Files** puis **New/C Sources Files**.

Le programme principal *main.c* sera créé en développant **Sources Files** puis **New/C Main File**.

Le programme principal a toujours un début identique. Par défaut, **MPLABX** ajoute automatiquement deux directives *#include* :

```
#include <stdio.h>
#include <stdlib.h>
```

Annexe - Utilisation du logiciel MPLAB X

Pour les projets, il faudra également inclure le fichier d'entête du compilateur **XC8**, ainsi que les fichiers d'entête dont on a besoin, ce qui donne :

```
#include <stdio.h>
#include <stdlib.h>
#include <xc.h>
#include "general.h"
#include "lcd.h"           //éventuellement
#include "i2c.h"           //idem
...                       //autres...
```

A la suite des directives *#include*, il faut insérer le contenu du fichier *configbits.c* :

```
#pragma config FOSC = INTIO7
#pragma config PLLCFG = OFF
...
#pragma config EBTRB = OFF
```

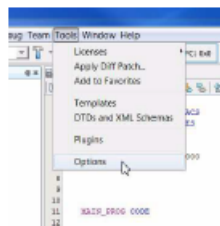
3 Utilisation du "debugger" - Simulation d'un programme

3.1 Lancement du debugger

Après compilation du programme, il est possible de le simuler et de visualiser le contenu des divers registres du microcontrôleur en lançant le debugger.

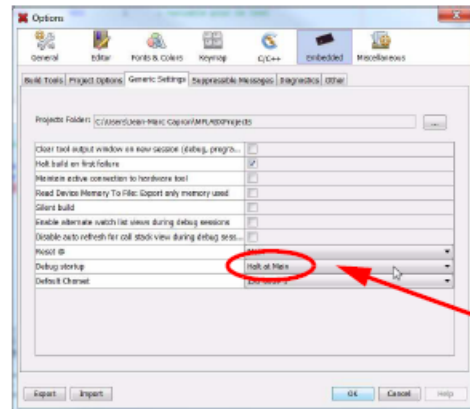
Dans un premier temps, si l'on souhaite lancer la simulation en mode pas-à-pas pour étudier le comportement du programme, il faut éviter que le simulateur n'exécute tout le programme au démarrage. Une option de MPLAB doit pour cela être modifiée.

Tools → Options

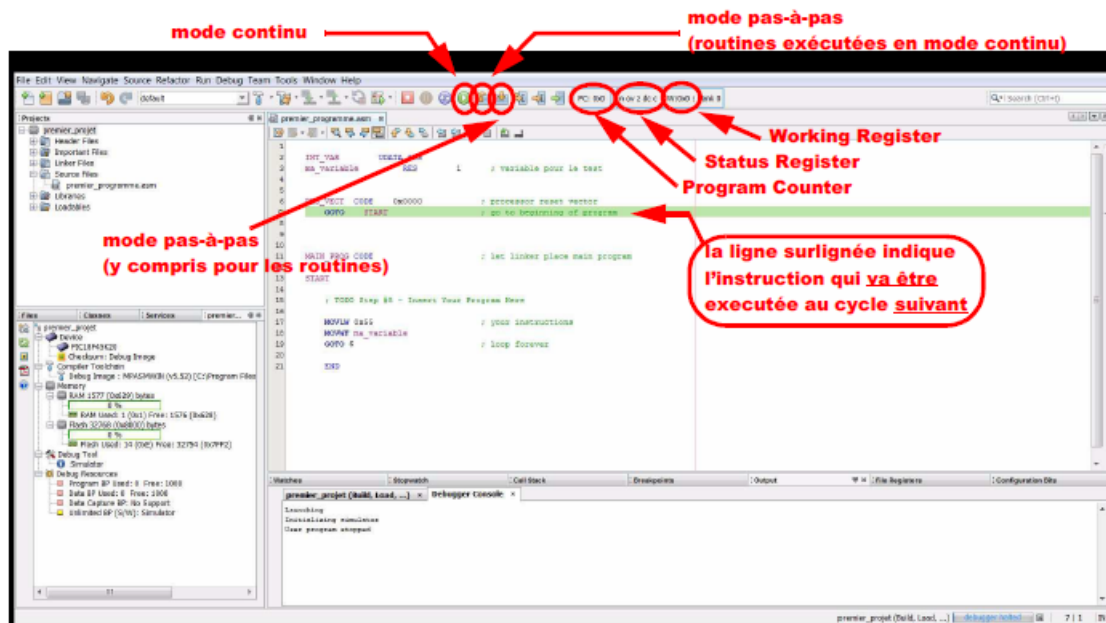


Annexe- Utilisation du logiciel MPLAB X

Embedded → Generic Settings → Debug Startup : Halt at Main.



La fenêtre principale permet ensuite de tester le programme en mode pas-à-pas (Step Into) ou de lancer son exécution (Continue), tout en visualisant le contenu des divers registres.



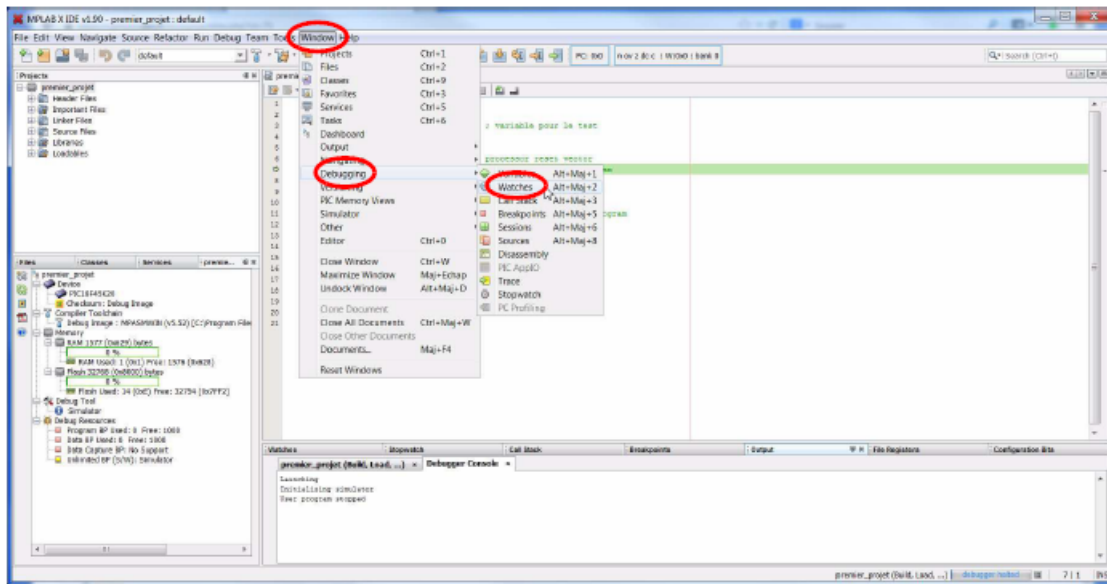
La barre d'outils en haut de la fenêtre principale permet de visualiser :

- le compteur ordinal ou Program Counter (adresse de l'instruction qui sera exécutée au cycle CPU suivant)
- le registre d'état ou Status Register (contient les flags qui renseignent sur le résultat de la dernière opération - une lettre minuscule signifie un flag à 0, une lettre majuscule signifie un flag à 1)
- l'accumulateur ou Working Register
- le numéro de banque courante

Annexe- Utilisation du logiciel MPLAB X

3.2 Visualisation du contenu des registres (contenu de la RAM)

Ouvrir la fenêtre "Watches" par la commande **Window** → **Debugging** → **Watches**



Un clic-droit dans la fenêtre "Watches" permet d'ouvrir la fenêtre "New Watch".



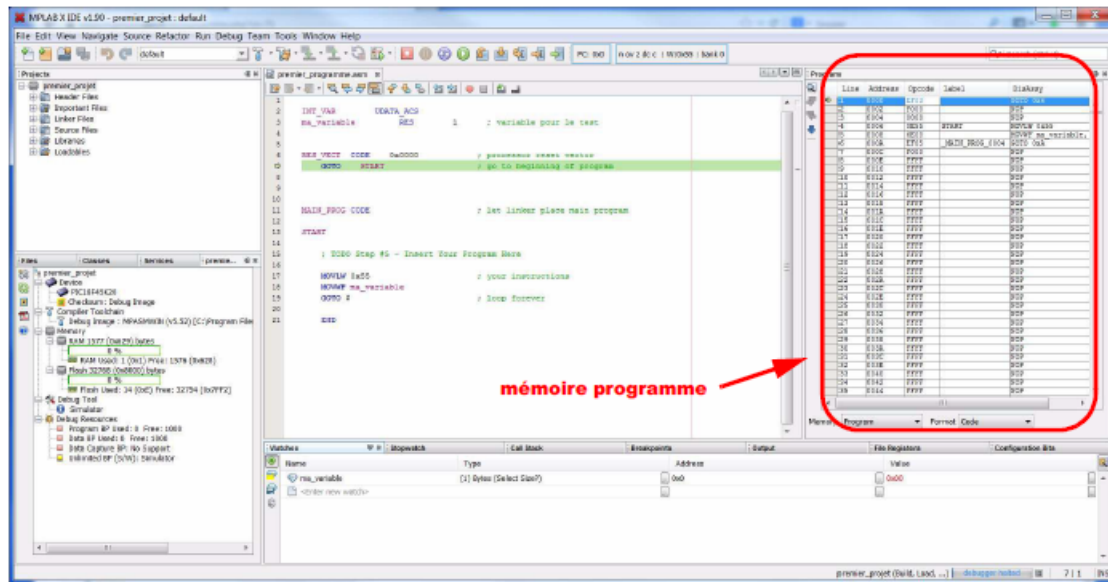
"Global Symbols" permet de visualiser les variables définies par l'utilisateur.

"SFR" (Special Function Register) permet de visualiser les registres spécifiques au microcontrôleur, qui sont liés aux périphériques et au matériel (port d'entrée/sortie, timers, ADC, etc...). (Se reporter à la datasheet du composant pour connaître la fonction de ces registres.)

Annexe- Utilisation du logiciel MPLAB X

3.3 Visualisation de la mémoire programme (contenu de la Flash)

Ouvrir la fenêtre par la commande **Window → PIC Memory Views → Program Memory**



La fenêtre donne les indications suivantes :

- Line : numéro d'instruction
- Adress : emplacement de l'instruction dans la mémoire programme
- Opcode : instruction en langage machine (sur 16 bits)
- Label : label éventuel donné à une instruction par l'utilisateur
- DisAssy (Dissassembly) : instruction en langage assembleur

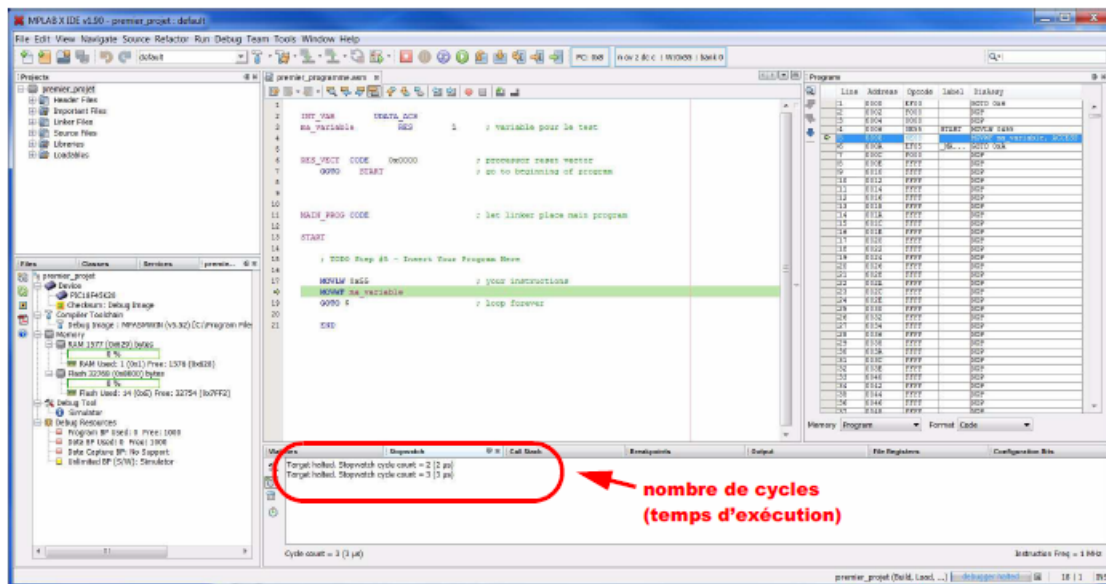
Annexe- Utilisation du logiciel MPLAB X

3.4 Visualisation du nombre de cycles (temps d'exécution)


Il est possible de visualiser le nombre de cycles nécessaire pour exécuter une instruction, une routine, ou une partie de programme.

Ouvrir dans un premier temps la fenêtre "Stopwatch" : **Window** → **Debugging** → **Stopwatch**

En lançant la simulation, que ce soit en mode pas-à-pas ou en mode continu, le nombre de cycles s'affiche dans la fenêtre "Stopwatch".



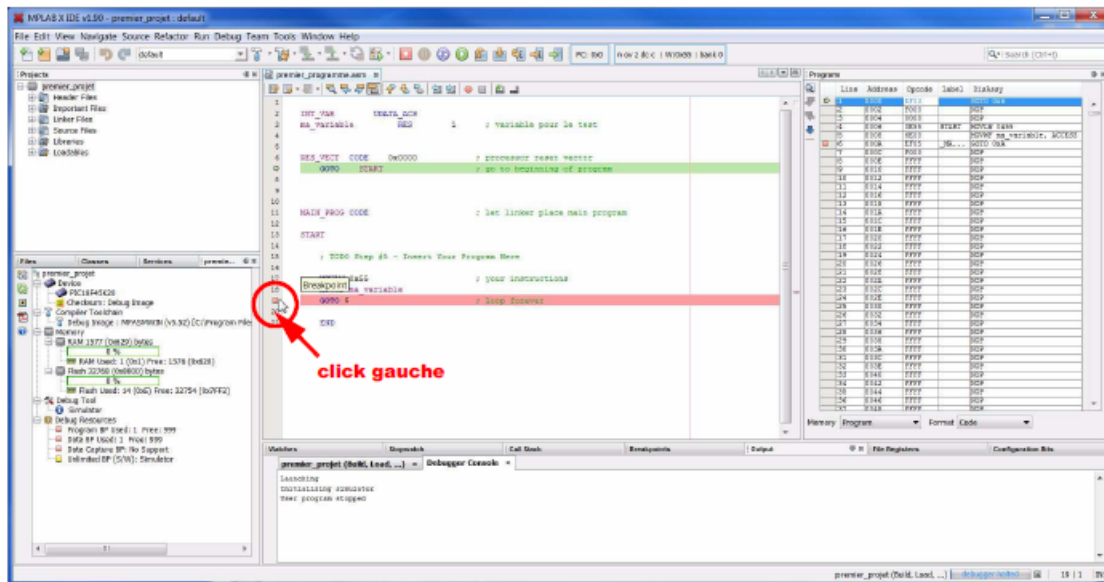
Remarque :

La simulation en mode pas-à-pas peut rapidement s'avérer fastidieuse lorsque les programmes sont conséquents. Il peut dans ce cas être utile de simuler directement toute une partie de programme, jusqu'à une instruction donnée. Il suffit pour cela de placer un point d'arrêt (**breakpoint**) sur l'instruction concernée, en cliquant dans la marge, puis de lancer la simulation en mode continu .

L'instruction où est placé le point d'arrêt est surlignée en rouge.

Annexe- Utilisation du logiciel MPLAB X

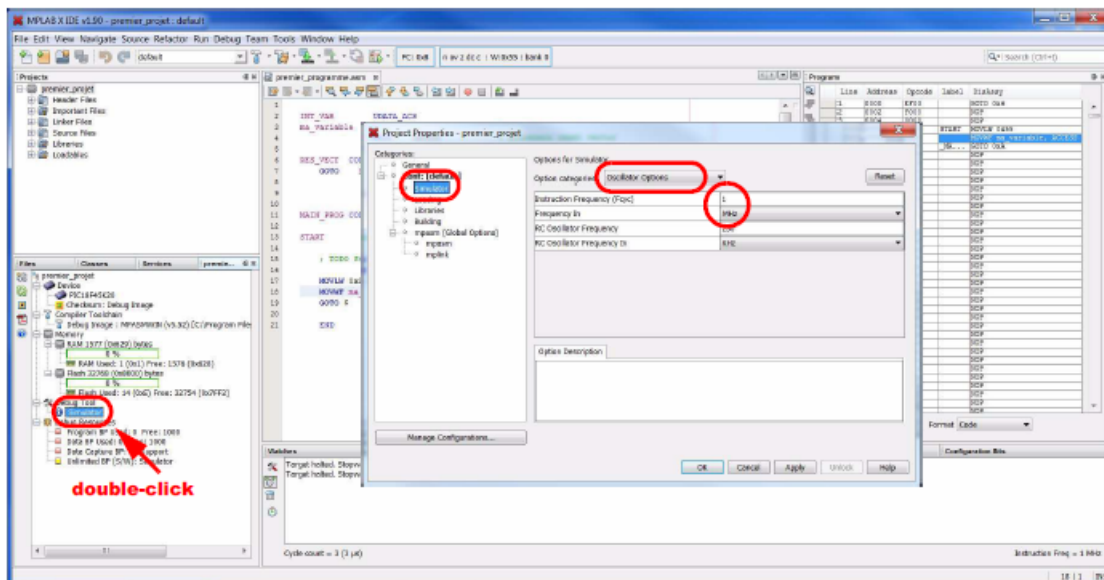
Un nouveau clic dans la marge supprime de la même manière le point d'arrêt.



Remarque :

Le simulateur prend par défaut $1\mu s$ comme temps d'exécution d'une instruction (soit une fréquence d'oscillateur de 4 MHz).

Il est possible de modifier ce paramètre de la manière suivante :



Annexe- Utilisation du logiciel MPLAB X

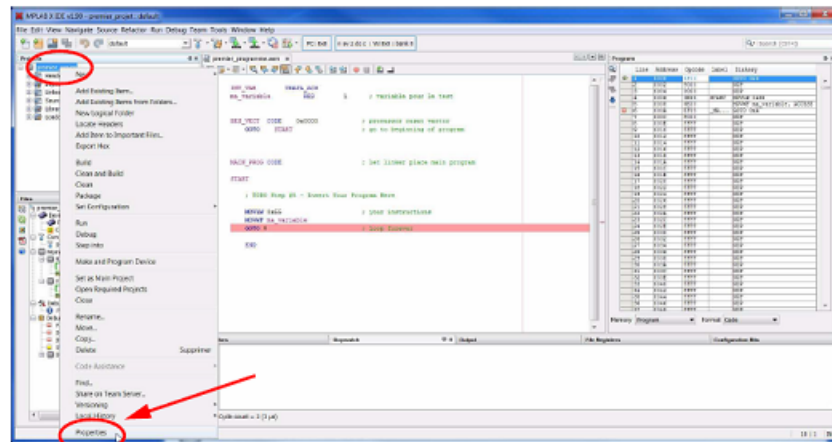
4 Utilisation du "debugger" - Exécution d'un programme sur le composant

4.1 Utilisation du PICKit3

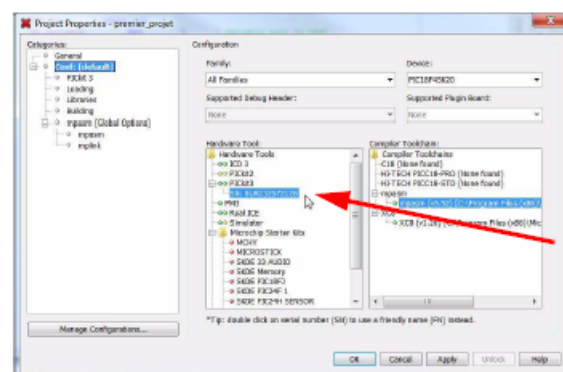
Pour exécuter le programme sur le composant réel, et non pas en simulation pure, un outil de debug (par exemple PICKit3) doit être connecté entre le PC et le microcontrôleur.

Si c'est le cas, lors de la création du projet (cf. annexe 1) il faut choisir comme outil de debug PICKit3 au lieu de Simulator.

Remarque : ceci peut toujours être par la suite dans les propriétés de projet. Clic-droit sur le nom du projet, dans la fenêtre Projects → Properties



puis Conf :[default] → Hardware Tool.



le numéro de série
doit correspondre à celui
inscrit derrière le PICKit3

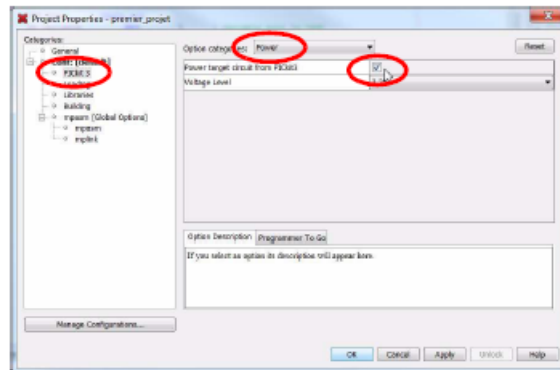
Annexe - Utilisation du logiciel MPLAB X

4.2 Alimentation de la carte de prototypage par le PICkit3

Lorsque la carte de prototypage ne dispose pas d'une source d'alimentation dédiée, il est possible de l'alimenter à partir du PICkit3, à condition qu'elle consomme moins de 30 mA.

Pour utiliser cette option, éditer les propriétés du projet (cf. page précédente).

PICkit3 → Power → Power target circuit from PICkit3

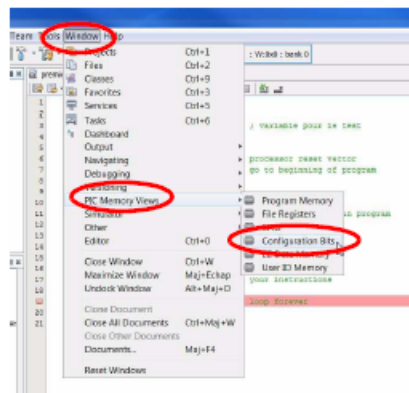


4.3 Édition des bits de configuration (Configuration Bits)

Avant de lancer le debugger sur la carte de prototypage, il est nécessaire d'éditer les bits de configuration qui vont spécifier les options matérielles du microcontrôleur.

Ouvrir la fenêtre "Configuration Bits".

Window → PIC Memory Views → Configuration Bits

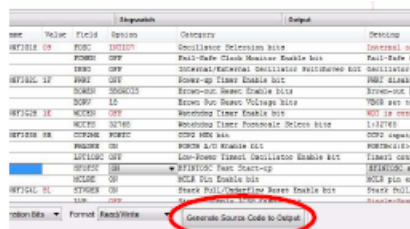


Annexe- Utilisation du logiciel MPLAB X

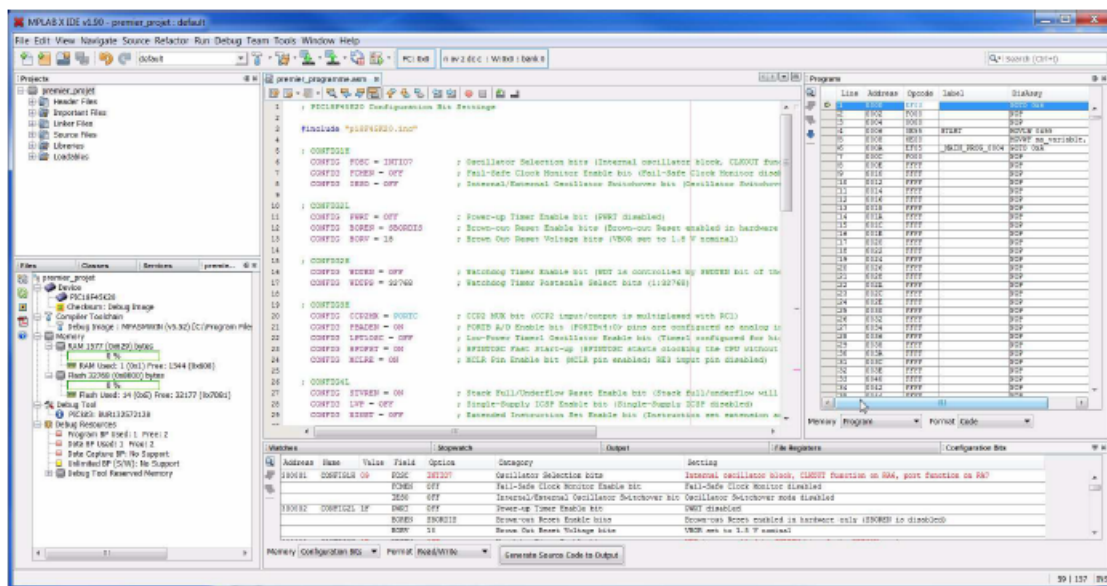
Dans la fenêtre "Configuration Bits", un certain nombre de bits de configuration doivent être modifiés :

FOSC	INTIO7	Oscillator Selection bits	Internal oscillator block, CLKOUT function on OSC2
WDTEN	OFF	Watchdog Timer Enable bits	Watch dog timer is always disabled. SWDTEN has no effect.
PBADEN	OFF	PORTB A/D Enable bit	PORTB<5:0> pins are configured as digital I/O on Reset
LVP	OFF	Single-Supply ICSP Enable bit	Single-Supply ICSP disabled

Sélectionner ensuite "Generate Source Code to Output"



Copier-coller le code ainsi généré au début du programme.



Le debugger peut ensuite être lancé comme précédemment, mais le programme sera réellement exécuté sur le microcontrôleur, et tous les registres observés dans l'IDE (données ou programme) refléteront le contenu réel du microcontrôleur.