

## CONFIGURATION DE L'ENVIRONNEMENT MPLAB

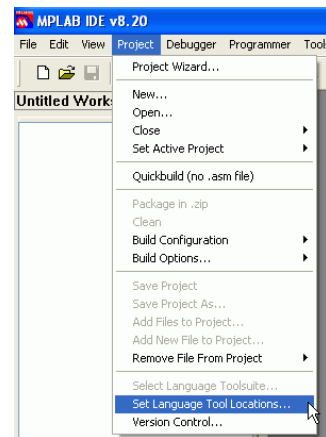
### Objectif :

- ☞ Configurer l'environnement MPLAB pour que la compilation avec le C18 se fasse sans problème

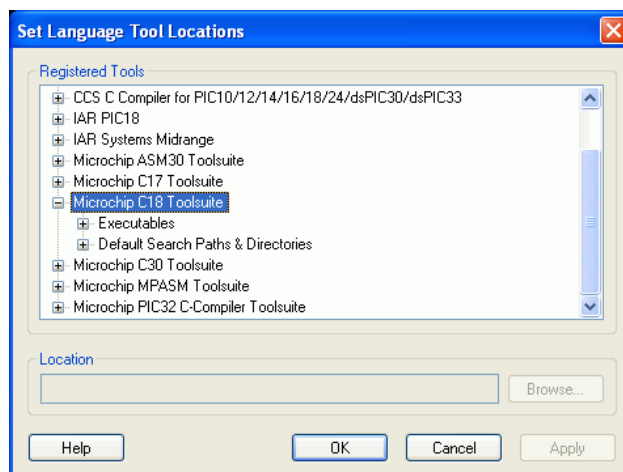
### Etape 1 : Ouvrir un projet MPLAB vierge



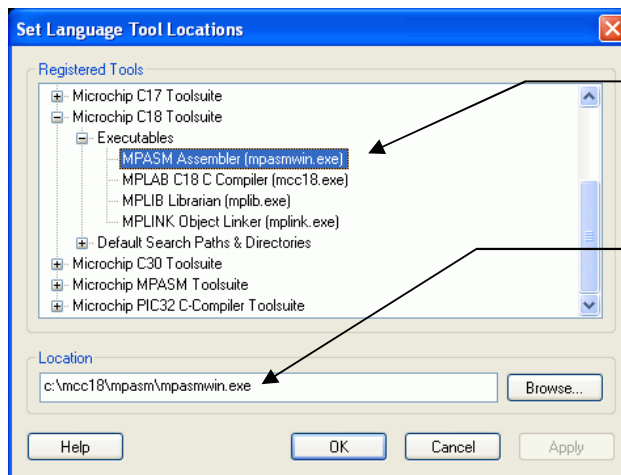
### Etape 2 : Sélectionner le menu "Set Language Tool Locations..."



### Etape 3 : Sélectionner "Microchip C18 Toolsuite"



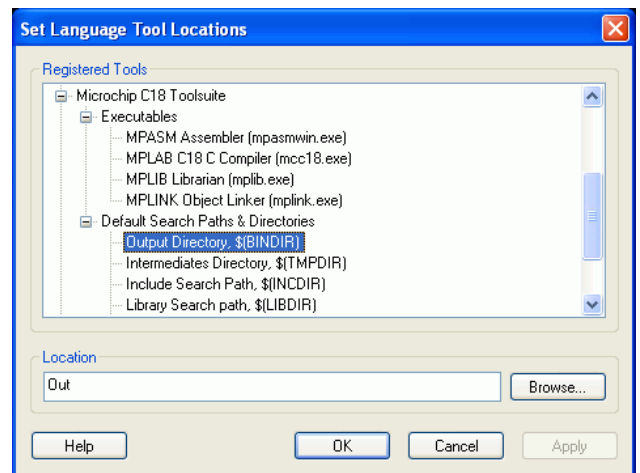
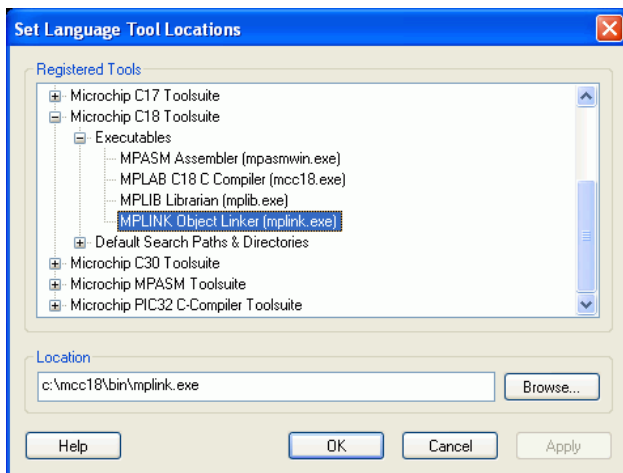
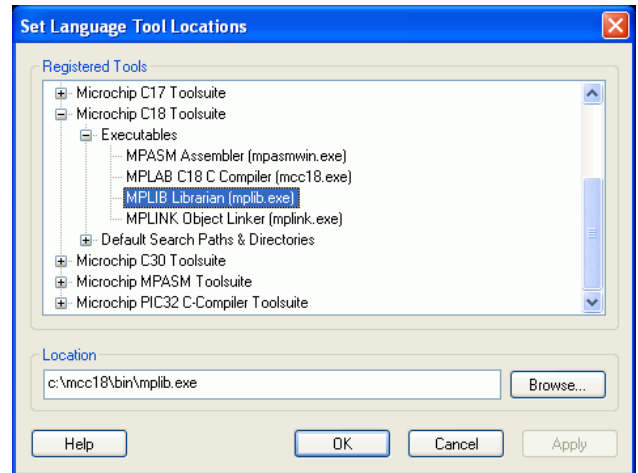
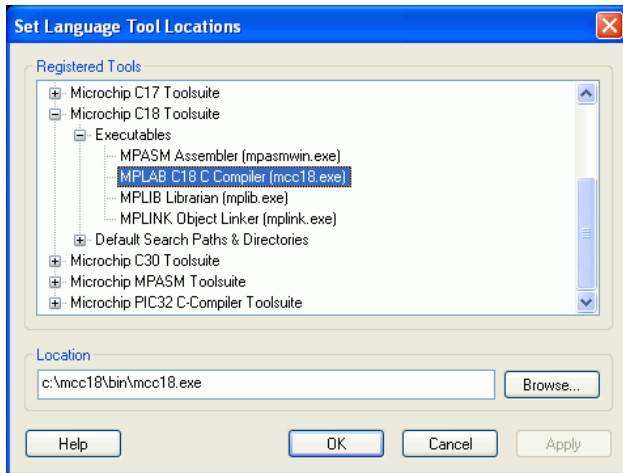
**Etape 4 :** Sélectionner "MPASM Assembler (mpasmwin.exe)" et vérifier si le chemin indiqué dans "Location" correspond bien au chemin de l'application "mpasmwin.exe"

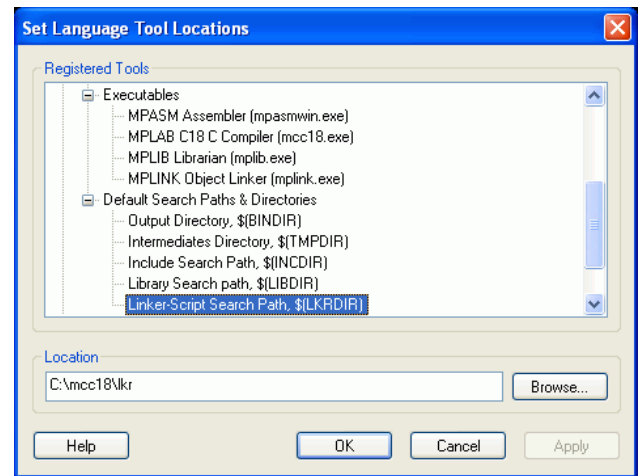
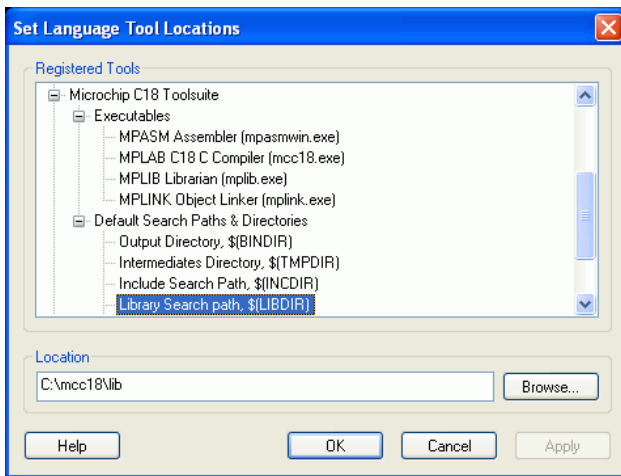
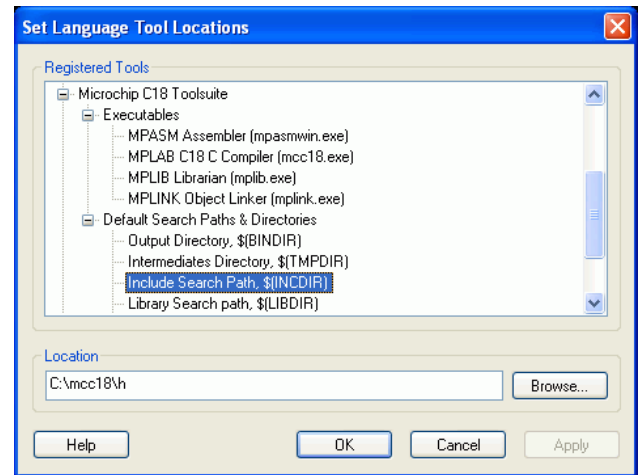
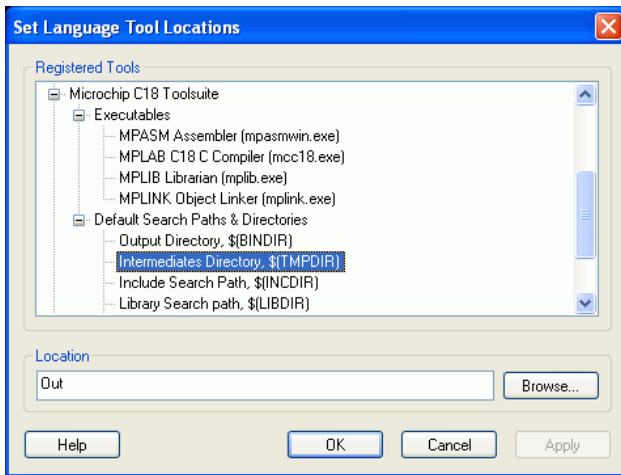


1 - Sélectionner ce menu

2 - Vérifier si le chemin est correct

**Etape 5 :** Faire de même avec les boites suivantes





## Etape 6 : Création d'un projet MPLAB.

Ouvrir le fichier Création Projet MPLAB et réaliser les différentes tâches.



## Etape 7 : STRUCTURE D'UN PROJET :

**Allume Led - MPLAB IDE v8.20 - [MPLAB IDE Editor]**

File Edit View Project Debugger Programmer Tools Configure Window Help

Debug Checksum: 0x8329

**Allume Led.mcp**

**Nom du Projet**

**Compiler le Projet**

**Liste des fichiers C**

**Liste des fichiers H**

```

// Permet l'astuce de compiler Var.h)

//*****
// Declaration des fichiers de parametres, entetes et variables
//*****
#include "../h/Main.h" // Inclus tous les fichiers necessaires pour le bon fonctionnement
#include "../h/Config.h" // Configuration du microcontrôleur
//*****

// Programme principal
//*****
void main(void)
{
    TRISBbits.TRISB0 = 0; // Place le bit RB0 en mode sortie
    TRISBbits.TRISB1 = 0; // Place le bit RB1 en mode sortie

    LATEbits.LATE1 = 1; // Allume la led sur le bit RB1 du microcontrôleur

    RB0 = 1; // RB0 a ete defini dans le fichier Main.h

    while(1); // Fait une boucle sans fin pour bloquer le programme
}
//*****
  
```

**Output**

Build Version Control Find in Files

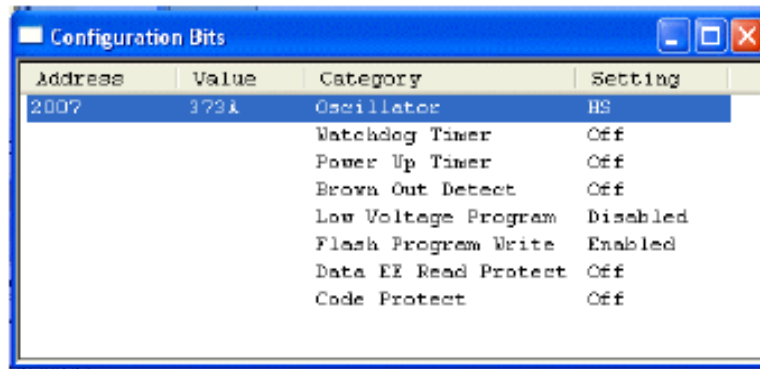
Packaged project in 'E:\Educ\Formation Proteus\Formation Proteus\Tp Micro\TP1\Prog\Allume Led.zip'.

PIC18F4520 W:0 n ov z dc c bank 0



## Etape 8 : Définition des bits de contrôle du PIC :

Les microcontrôleurs PIC disposent d'un registre de configuration « Word Configuration » situé à l'adresse 0x2007. La configuration de celui-ci s'effectue via MPLAB. Pour le configurer, sélectionner dans le menu « Configure » l'option « Configuration bits ». La fenêtre ci-dessous s'affiche :



Détails des bits de configuration :

Nom du bit (« Category »)	Valeur (« Setting »)
<b>Oscillator</b>	<b>HS</b> Fréquence du quartz > 8 MHz. <b>XT</b> 200 kHz < Fréquence du quartz < 8 Mhz. <b>LP</b> Fréquence du quartz < 200 kHz. <b>RC</b> Réseau RC.
<b>Watchdog Timer</b> (Activation du chien de garde).	En général sur « <b>Off</b> »
<b>Power Up Timer.</b> Rajoute un délai de 72 mS.	En général sur « <b>Off</b> »
<b>Brown Out Detect.</b>	En général sur « <b>Off</b> »
<b>Low Voltage Program</b>	Toujours sur « <b>Disabled</b> » avec l'ICD2.
<b>Flash Program Write</b> Active la programmation de la mémoire programme.	Toujours sur « <b>Enabled</b> ».
<b>Data EE Read Protect</b> Empêche la lecture de la mémoire EEPROM du PIC.	En général sur « <b>Off</b> »
<b>Code Protect</b> Activation de la protection du code programme.	En général sur « <b>Off</b> »

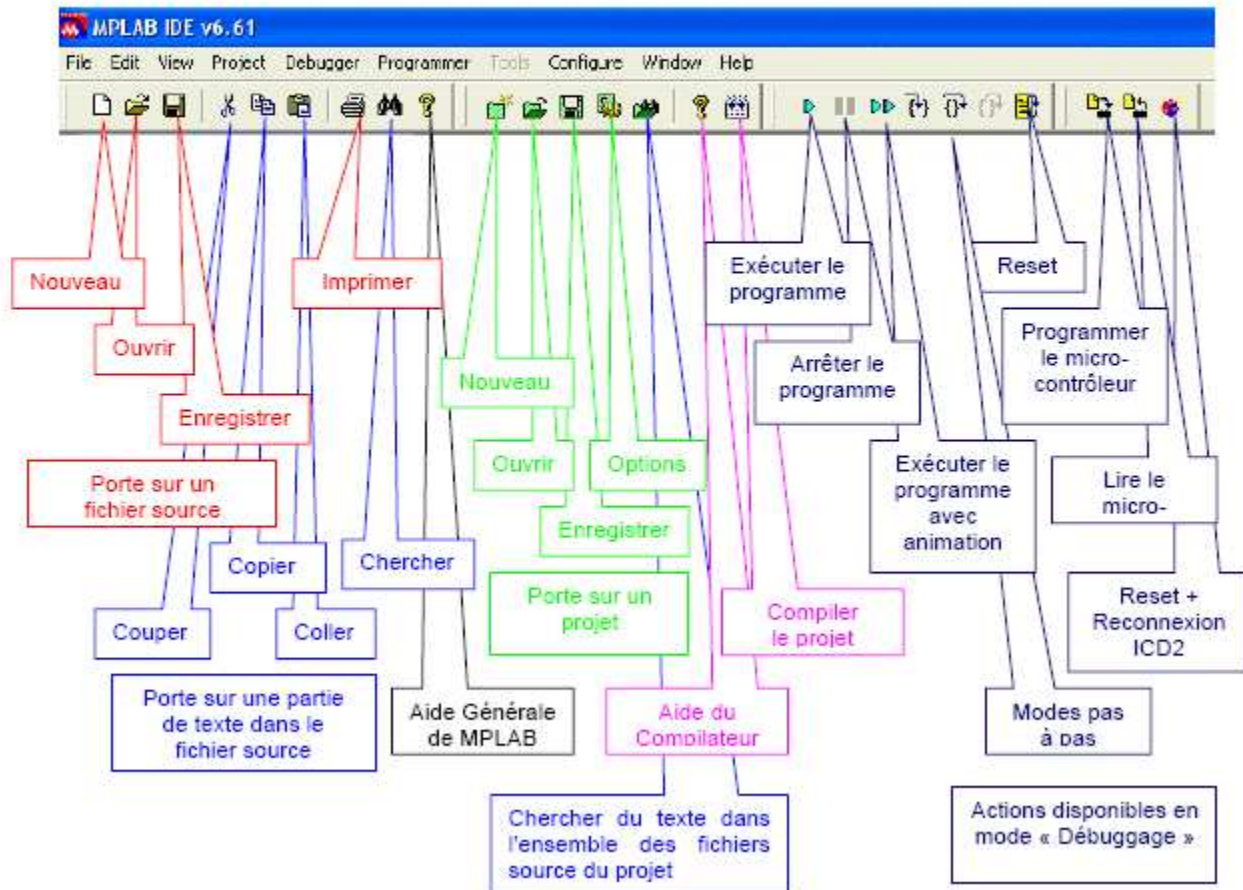
Une fois les différents bits définis correctement, fermer cette fenêtre en cliquant sur la croix en haut à droite.



## Etape 9 : Barre d'outil de MPLAB.

Les différentes icônes disponibles dans la barre d'outils de MPLAB sont définies ci-dessous.

Remarque : en fonction du mode de fonctionnement choisi (mode simple, mode « Débugage », mode programmation), les icônes de la partie droite ne sont pas les mêmes. La barre d'outils représentée ici correspond au mode « Débugage ».



## Etape 10 : Mode de débbugage.

Pour détecter d'éventuelles erreurs de programmation, il est nécessaire de faire exécuter le programme par le microcontrôleur en mode « débbugage ». Ce mode permet les différentes actions suivantes :

- exécution en pas à pas ;
- exécution fonction par fonction ;
- exécution en continu ;
- exécution continue jusqu'à une certaine ligne du programme source (« point d'arrêt »);
- affichage du contenu de registres du microcontrôleur et/ou variables (valeurs actualisées à chaque arrêt du programme).





## Connexion PC / ICD2 / Application.

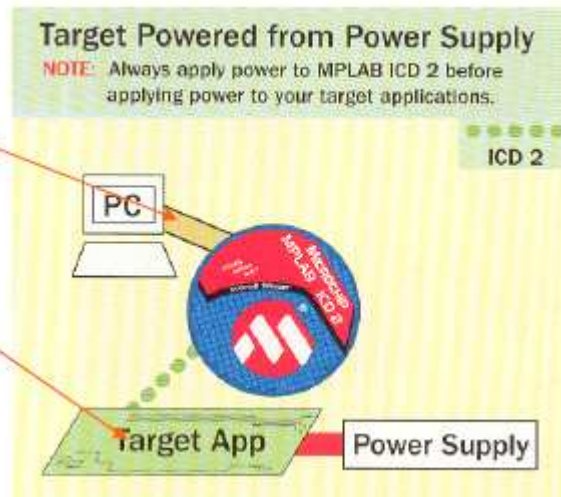
Auparavant, le boîtier ICD2 doit être raccordé, d'une part à l'ordinateur via le bus USB et d'autre part à l'application cible comportant le microcontrôleur à programmer.

Le boîtier ICD2 doit être relié au PC en premier lieu

Ensuite, raccorder le boîtier ICD2 à l'application cible.

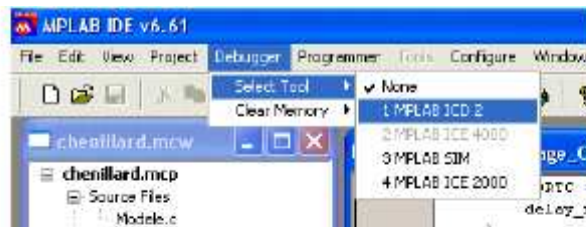
Terminer en alimentant l'application cible.

Le non respect de l'ordre de connexion peut entraîner la destruction du boîtier ICD2.

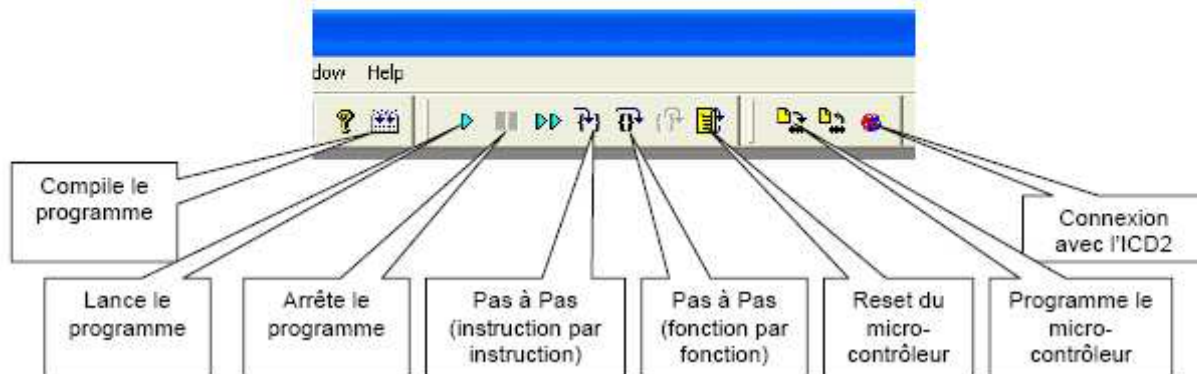


## Passage en mode « Débugage ».

Pour bénéficier du mode de fonctionnement « Débugage », choisir dans le menu « Debugger » l'option « Select Tool » puis « MPLAB ICD 2 ».



Les principales icônes utilisées pour le « débugage » apparaissent :



- ✚ Compiler le programme.
- ✚ Connexions avec iCD2.
- ✚ Faire un reset du PIC.
- ✚ Programmer le PIC.
- ✚ Lancer le programme.

Que constatez-vous ?

- ✚ Exécuter le programme en pas à pas.
- ✚ Réaliser un algorithme du programme main.c.



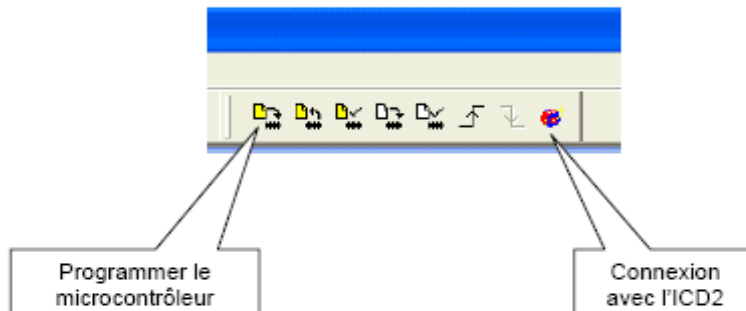
## Etape 10 : Programmation du microcontrôleur.

Pour effectuer la programmation du microcontrôleur afin qu'il fonctionne de façon autonome sur l'application cible, choisir dans le menu « Programmer » l'option « Select Programmer » puis « MPLAB ICD 2 ».

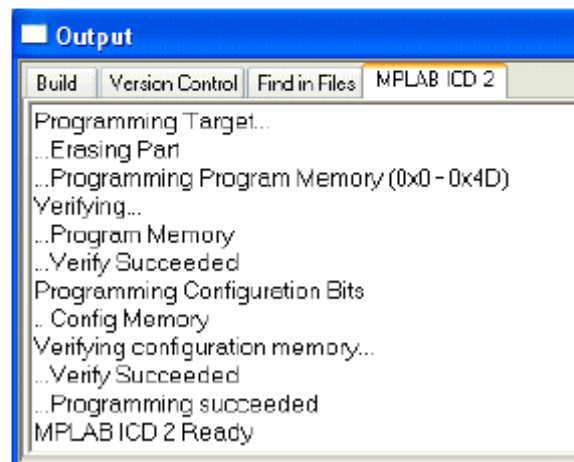


Les icônes présentes dans la partie droite de l'écran pour le mode « Débuggage » sont remplacées par une autre série.

Les 2 principales icônes à utiliser sont, dans un premier temps celle de connexion avec l'ICD2, puis dans un second temps, celle de programmation du microcontrôleur cible.



En cas de succès, la fenêtre « Output » affiche les messages suivants :



Après la programmation, il est indispensable de déconnecter le boîtier ICD2 de l'application cible : celle-ci devient autonome.

 Programmer la cible et effectuer un essai.

## Etape 11: Programmation du microcontrôleur avec un fichier hex.

- ✓ Lancer mplab.
- ✓ Fichier exporter le fichier **P18DEMO.HEX**
- ✓ Programmer le PIC
- ✓ Tester le programme

