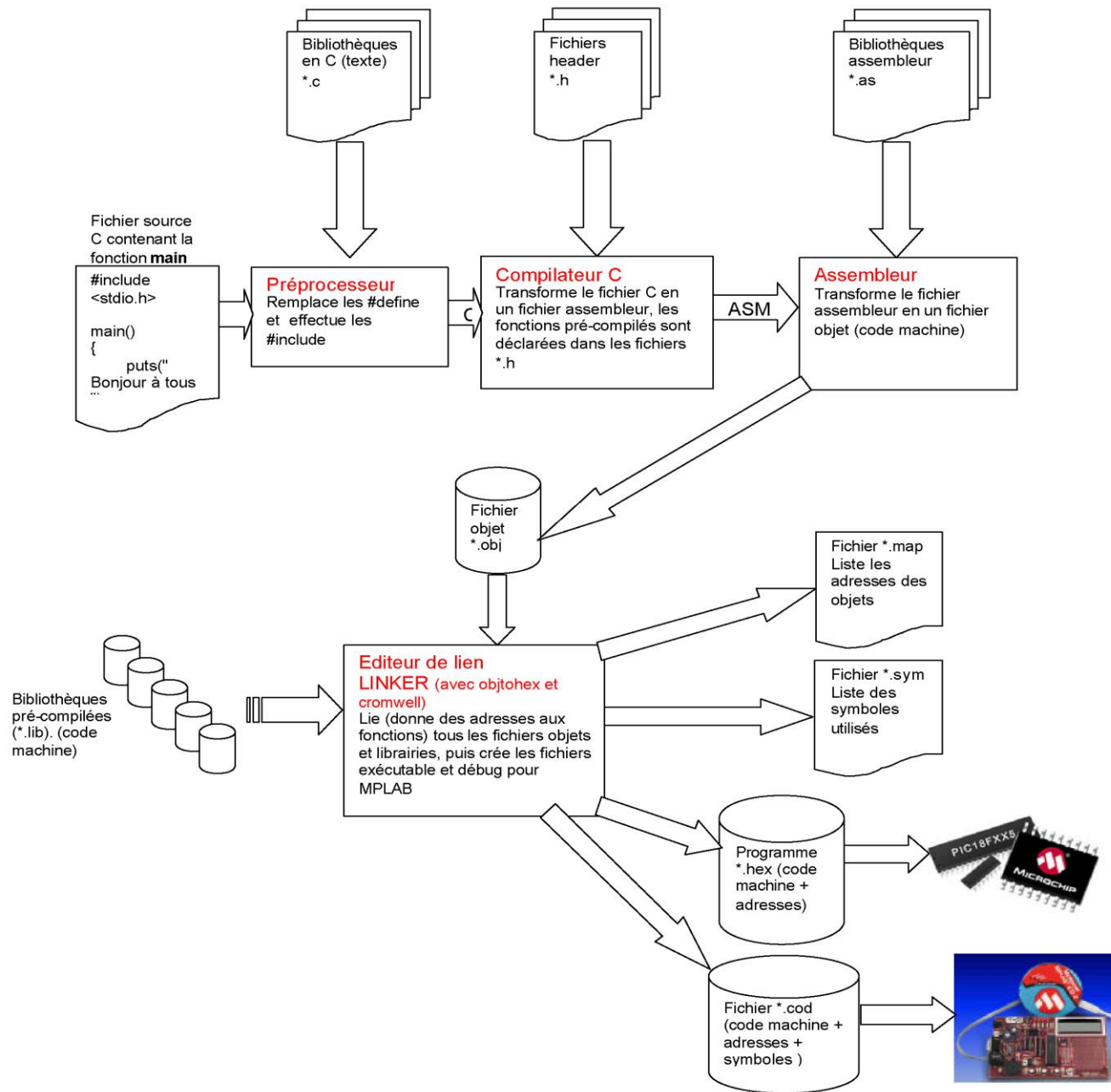


# Utilisation de MPLAB X IDE

## Simulation/Programmation PIC18F en C



# Chap1 :

# Créer un projet

# Méthode

- Ouvrir MPLABX IDE
- Choisir Project Create New (Lancement assistant de création de projet)



LEARN & DISCOVER | [MY MPLAB® X IDE](#) | WHAT'S NEW

## PROJECTS

- ▶ Open Sample
- ▶ **Create New**
- ▶ Import Legacy
- ▶ Import Prebuilt



DATA SHEETS

EMBEDDED  
CODE SOURCE

## MY MPLAB® X IDE

### Recent Projects

<no recent project>

### Microchip Login

#### Already Registered?

E-mail Address:

Password:

LOGIN

[Forgot Password?](#)

#### New to Microchip?

Register now to enjoy the benefits of software update alerts and easy software purchasing, licensing and account viewing.

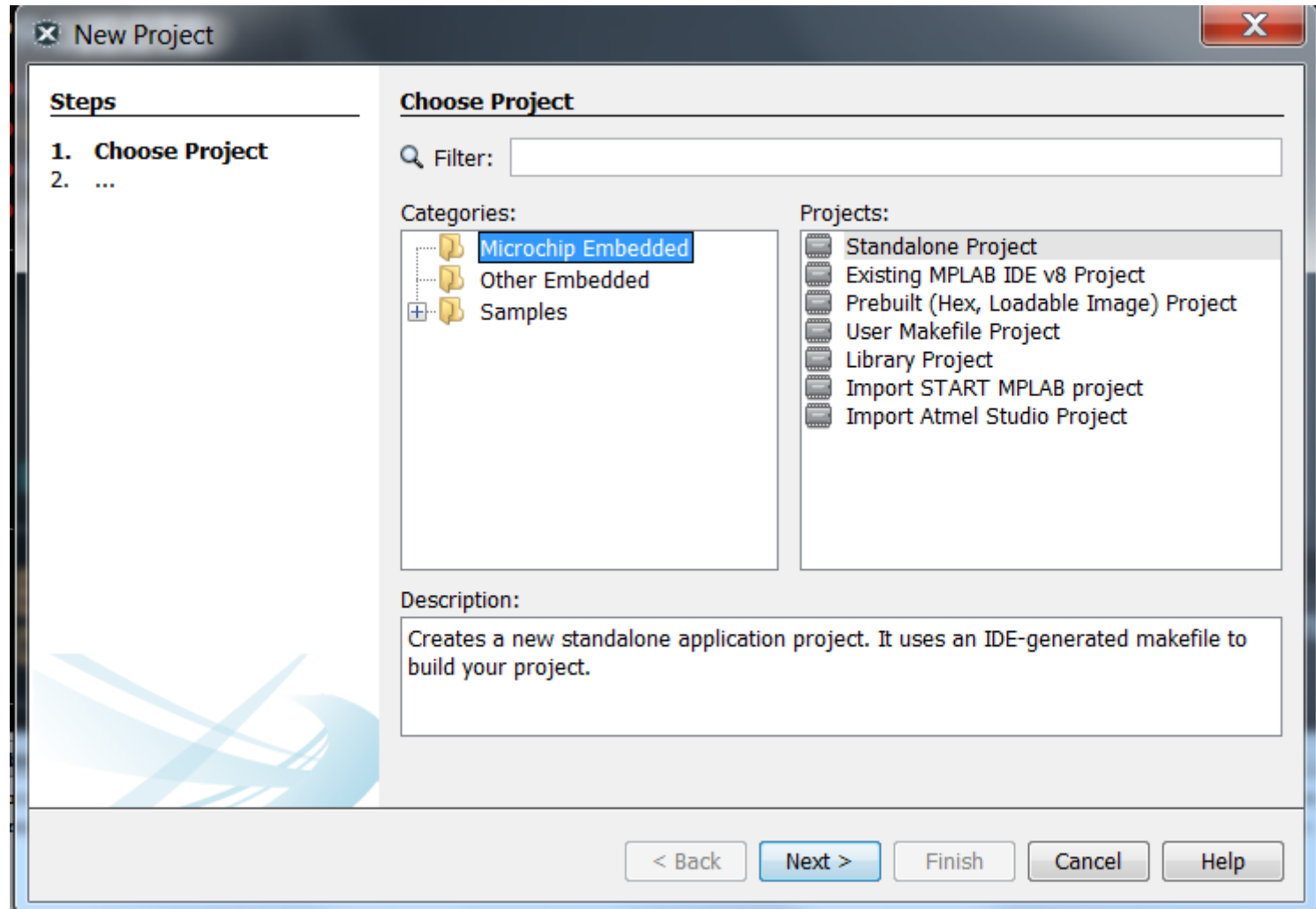
REGISTER NOW

### References & Featured Links

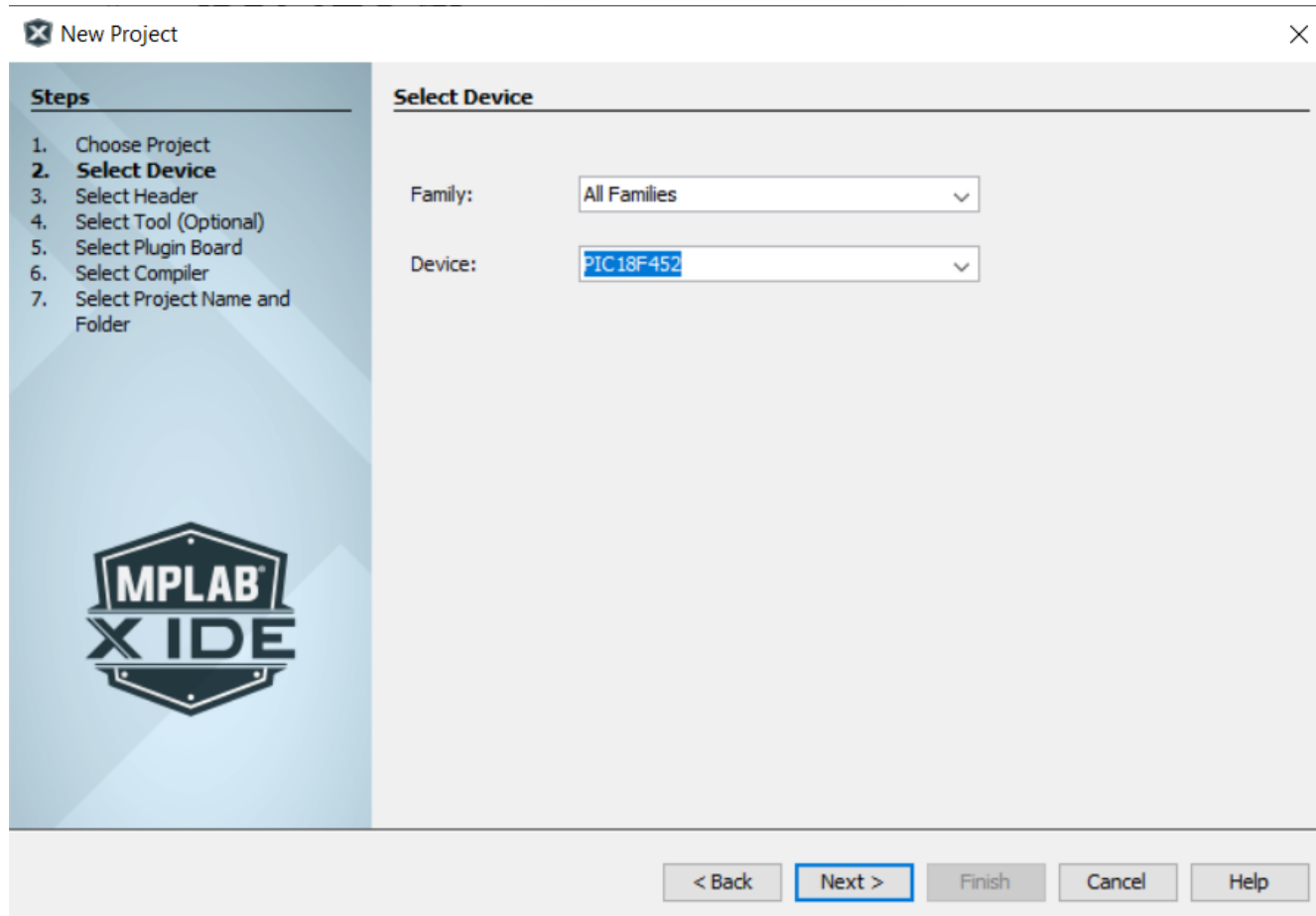
▶ [Errata](#)

▶ [Technical Articles and White Papers](#)

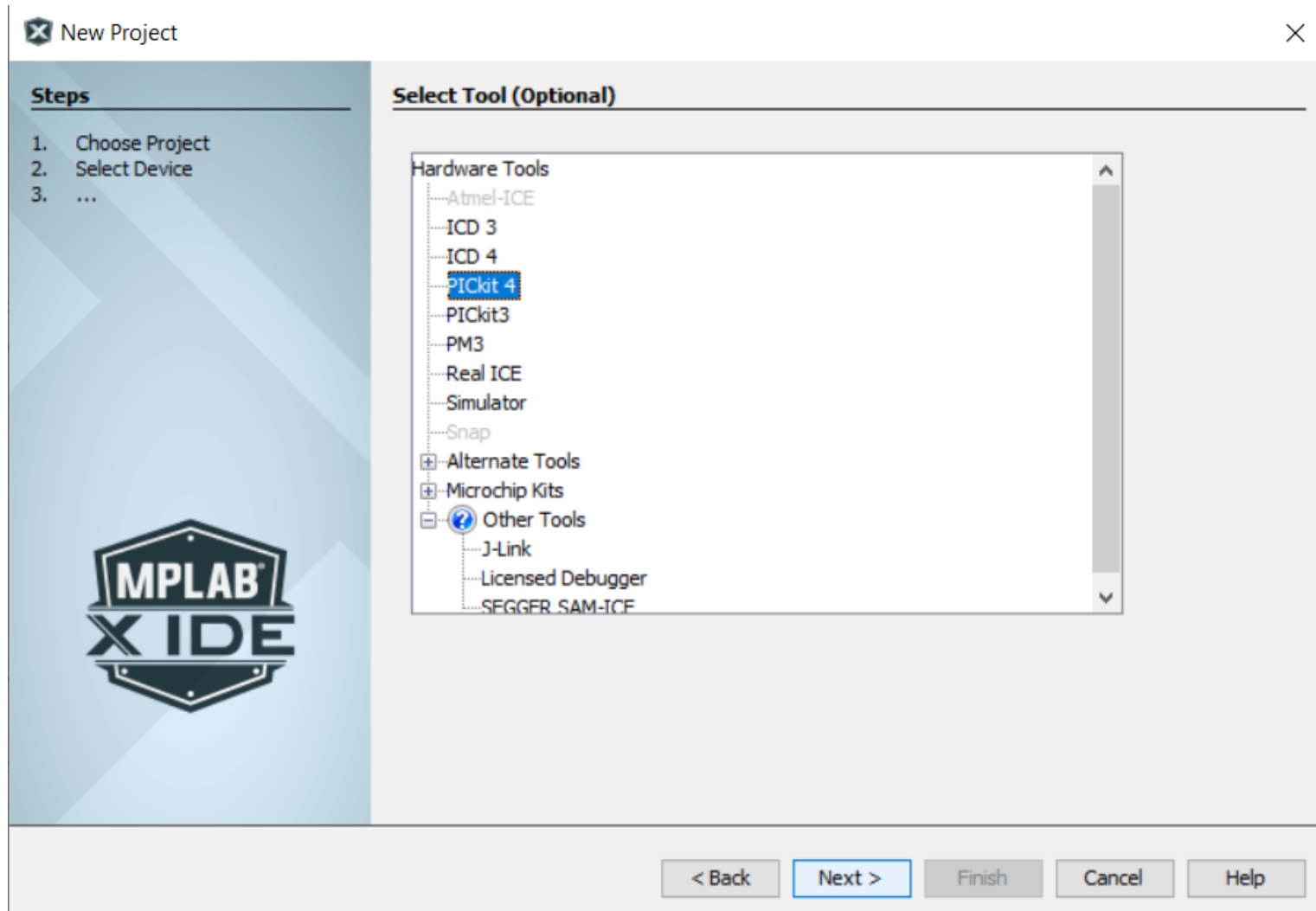
- Choisir catégorie Microchip Embedded et un projet du type Standalone project.
- Puis cliquer sur Next.



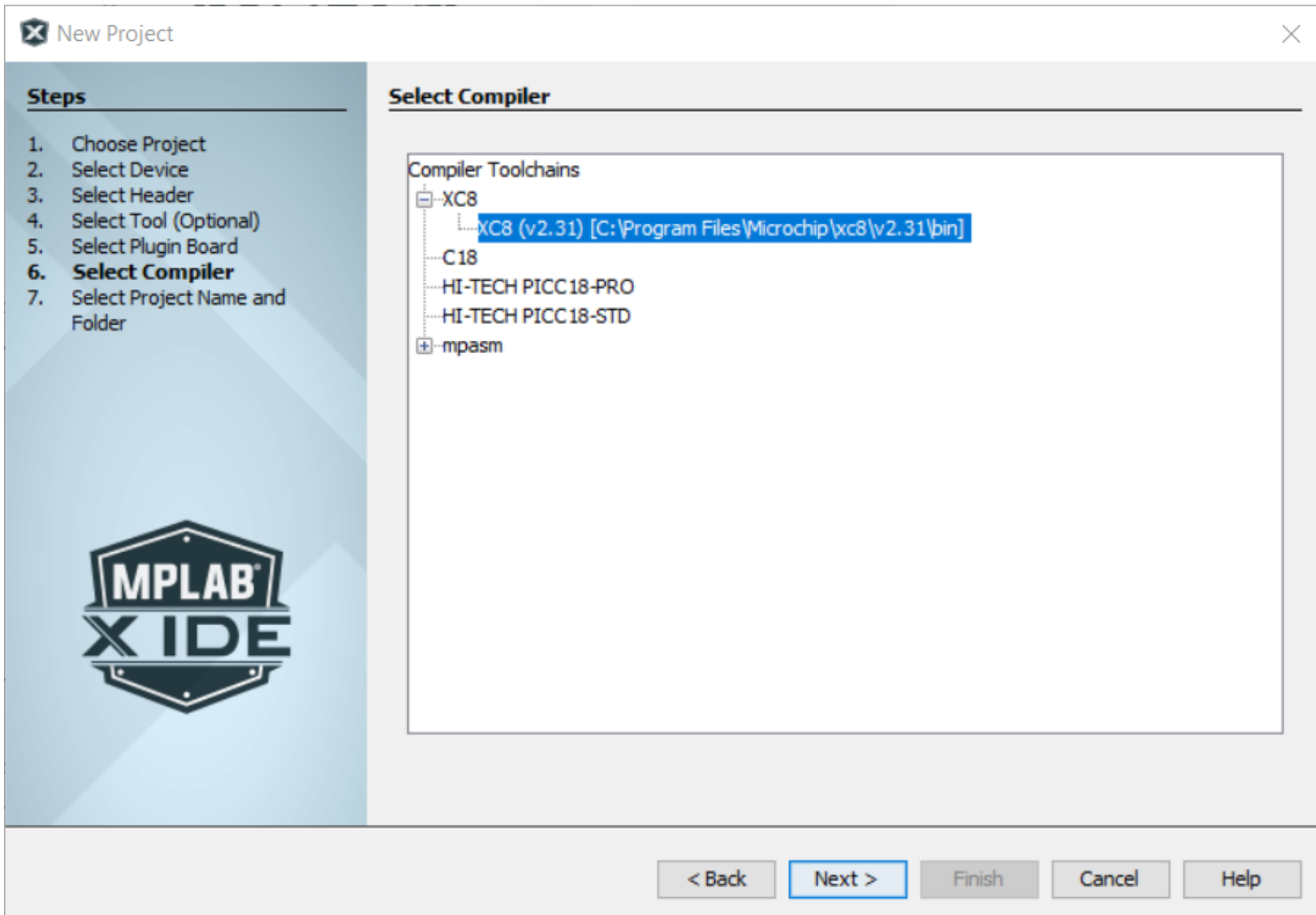
- Choisir le microcontrôleur PIC18F452
- Puis cliquer sur Next.



- Sélectionner PICkit 4.
- Puis cliquer sur Next.

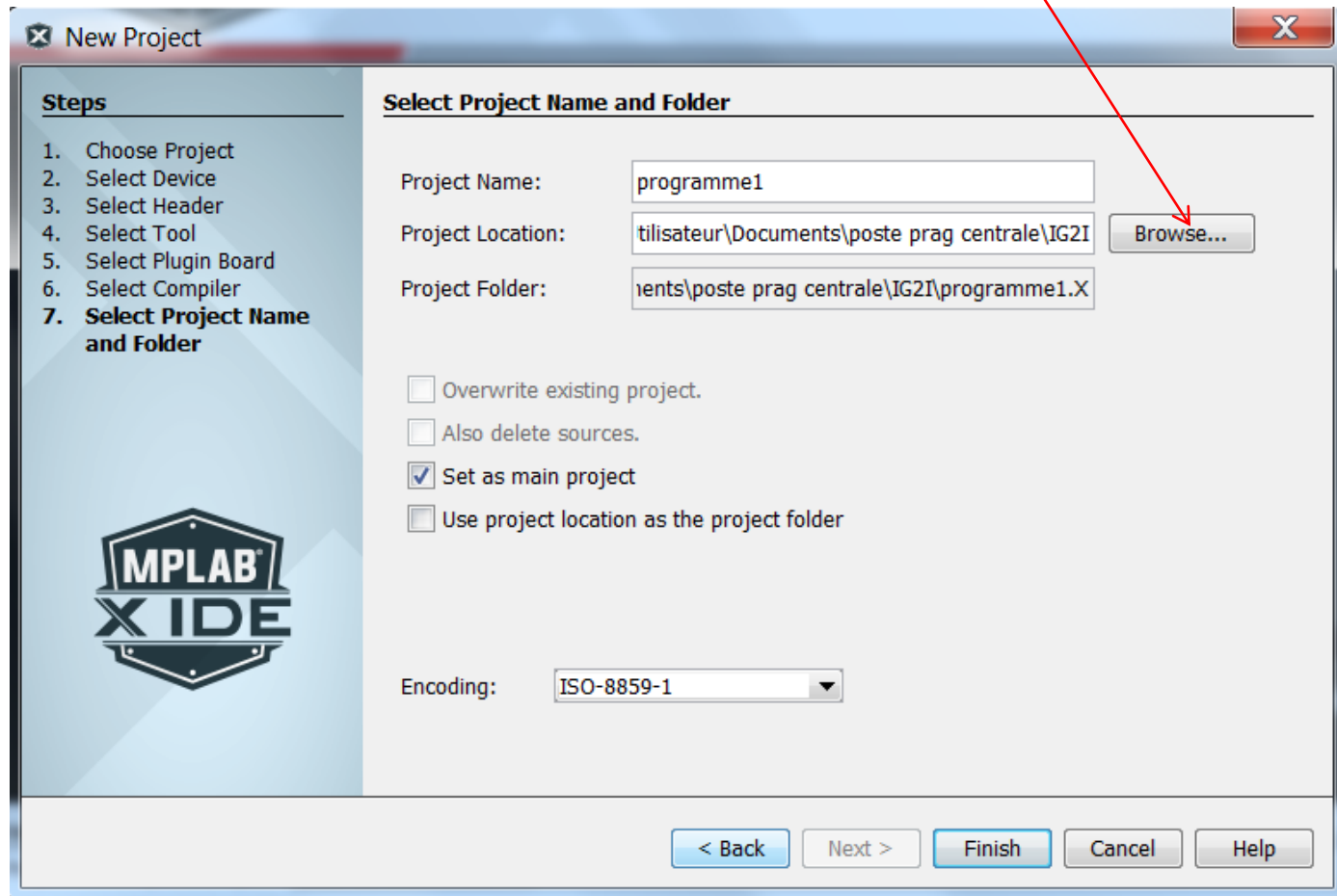


- Sélectionner compiler XC8 en C.
- Puis cliquer sur Next.

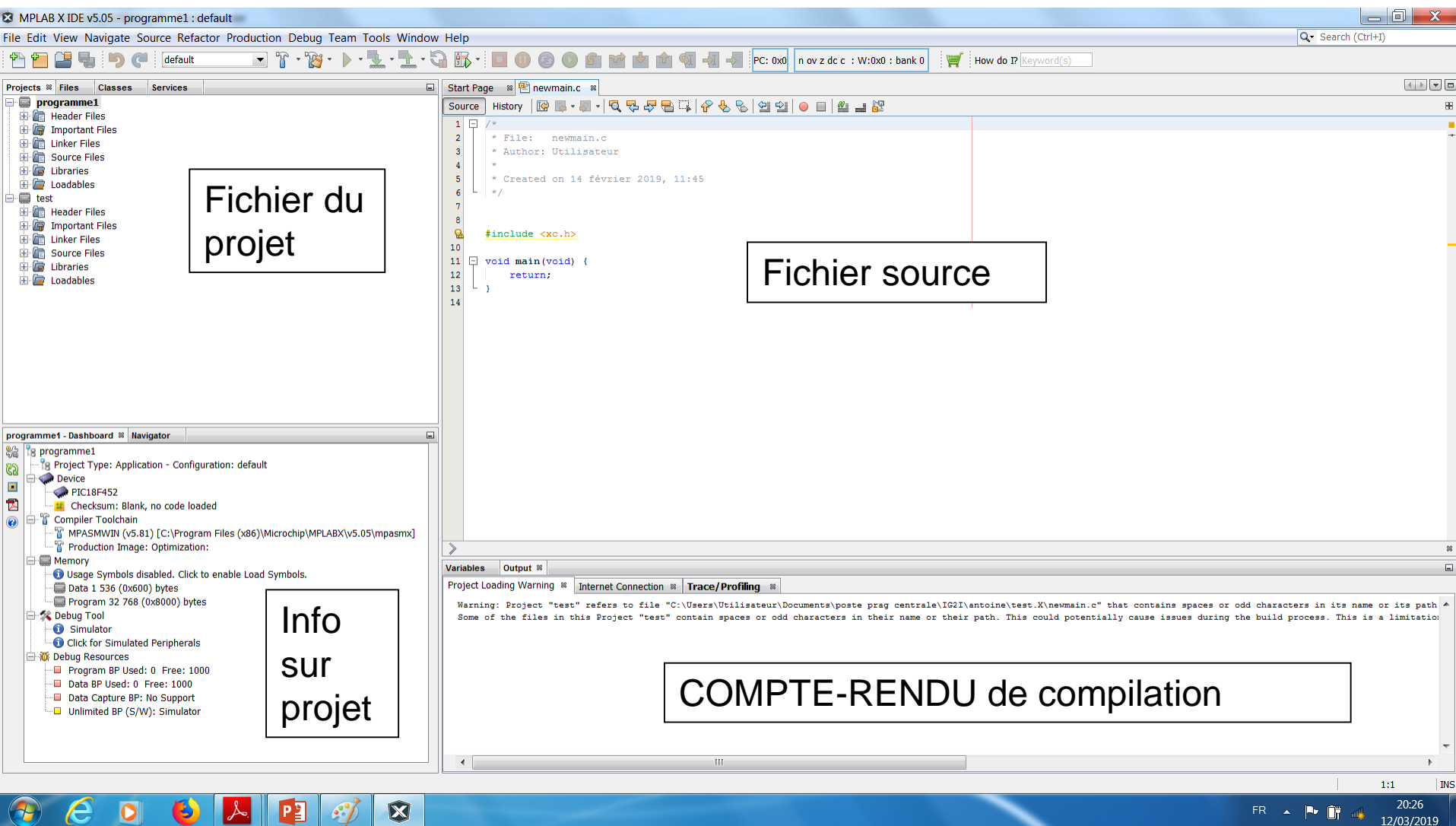




- Donner un nom à votre projet (TPx par exemple)
- Choisir votre répertoire de travail (par exemple dans U :/)
- Puis cliquer sur Finish.



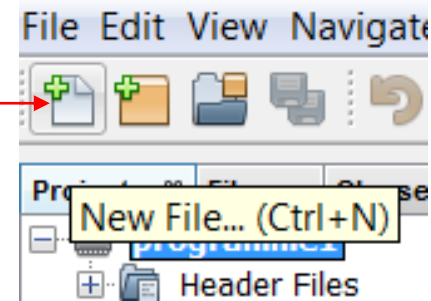
# L'interface graphique de l'environnement de développement MPLABX est représentée en 4 zones ci-dessous.



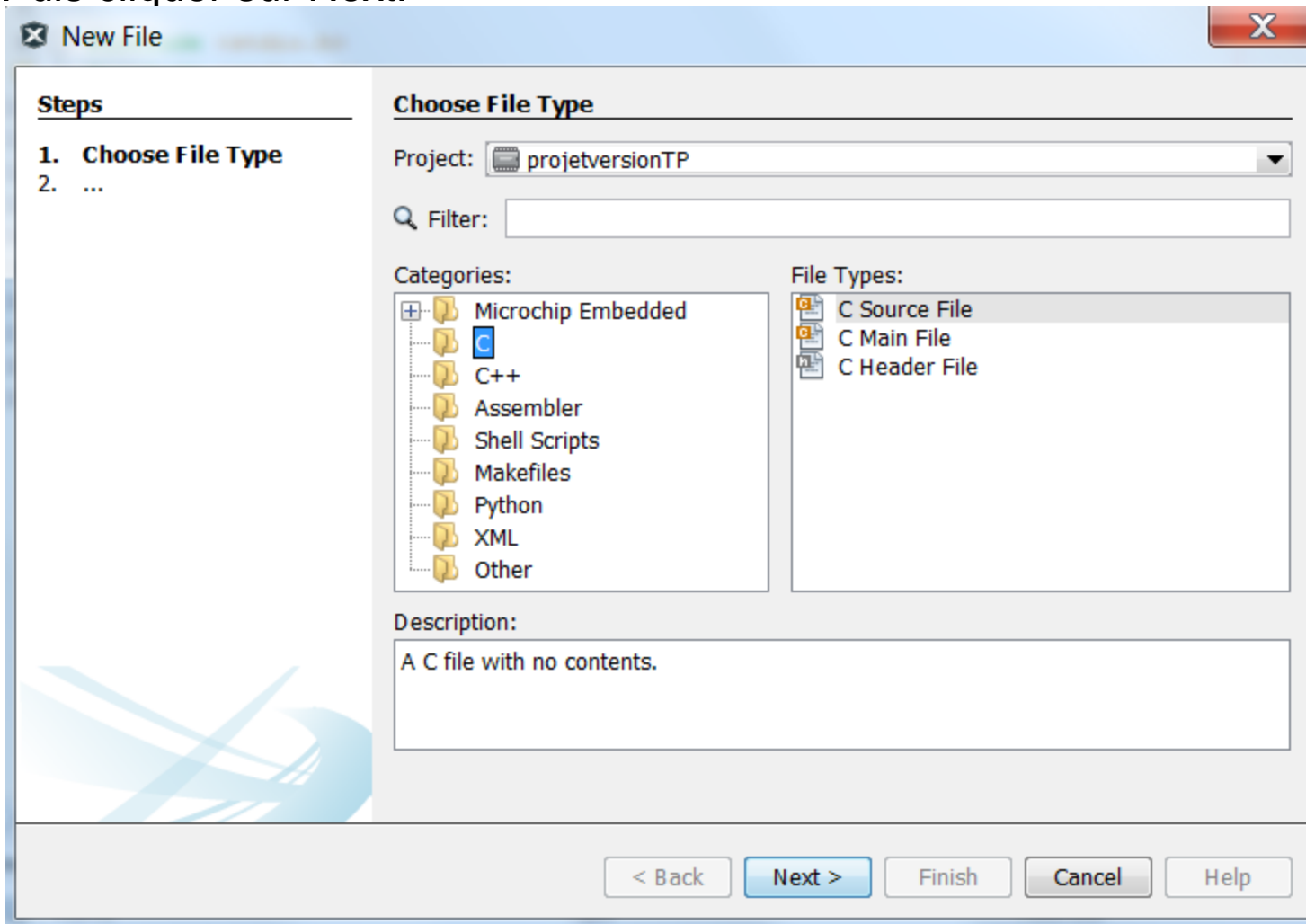
# Chap2 :

## Ecrire un nouveau code source

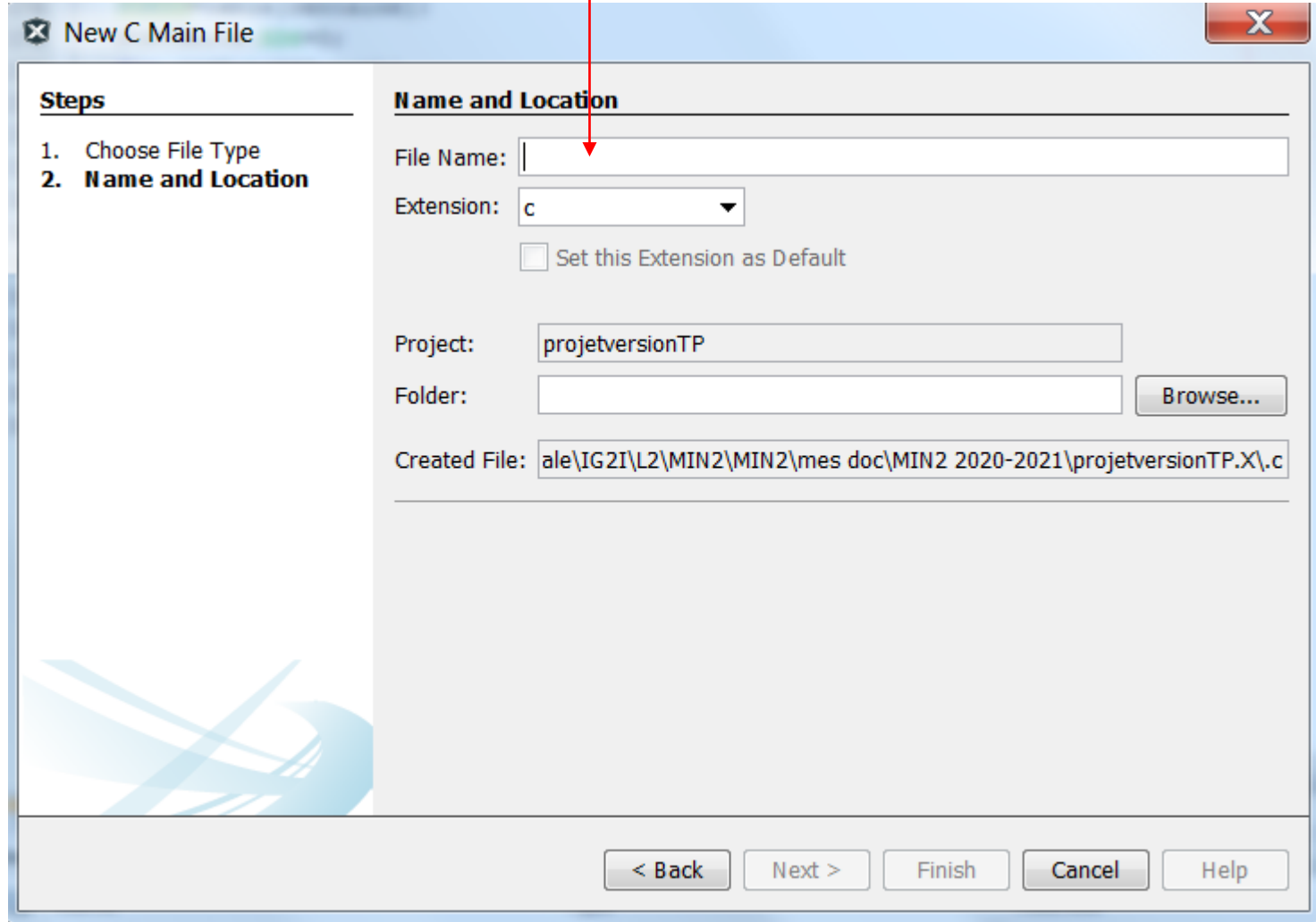
- Ouvrir le fichier source:
  - File / New File ou le raccourci



- Choisir C et le type de fichier C Source File.
- Puis cliquer sur Next.



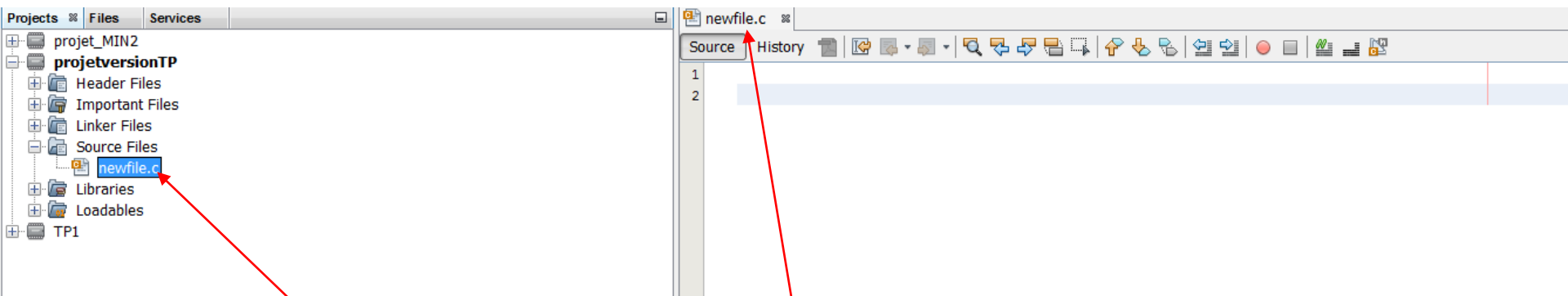
- Donner un nom à votre fichier source
- Cliquer sur Finish.



The screenshot shows a dialog box titled "New C Main File". On the left, under the "Steps" section, step 2 "Name and Location" is selected. The main area contains the following fields and controls:

- Name and Location** section:
  - File Name:** An empty text input field. A red arrow points to this field.
  - Extension:** A dropdown menu currently showing "c".
  - ☐ Set this Extension as Default
  - Project:** A text input field containing "projetversionTP".
  - Folder:** An empty text input field with a "Browse..." button to its right.
  - Created File:** A text input field showing the path "ale\IG2I\L2\MIN2\MIN2\mes doc\MIN2 2020-2021\projetversionTP.X\c".

At the bottom of the dialog, there are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help".



Vous devriez obtenir un fichier \*.c dans la zone  
fichier source

Si il n'apparait pas, cliquez droit sur Source Files et vous l'ajoutez avec « add existing  
Item »

Vous pouvez saisir votre programme.

Vous avez à disposition un fichier squelette.  
Vous pouvez l'utiliser en le renommant

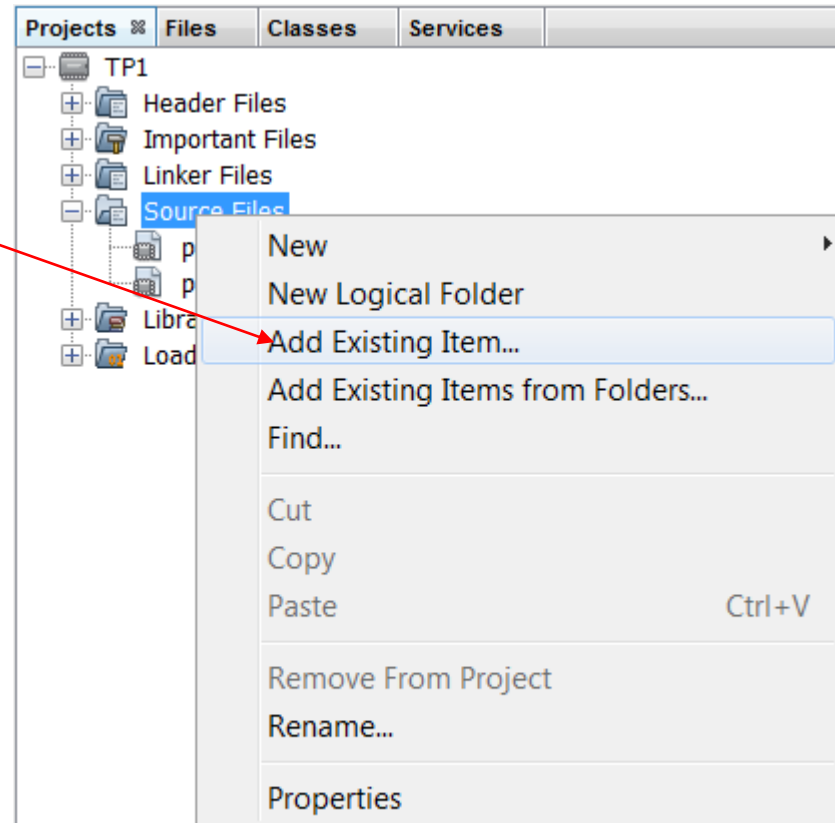
# Chap3 :

## Changer code source



# Changer code source

- Vous pouvez créer un autre code source ou en ajouter un existant

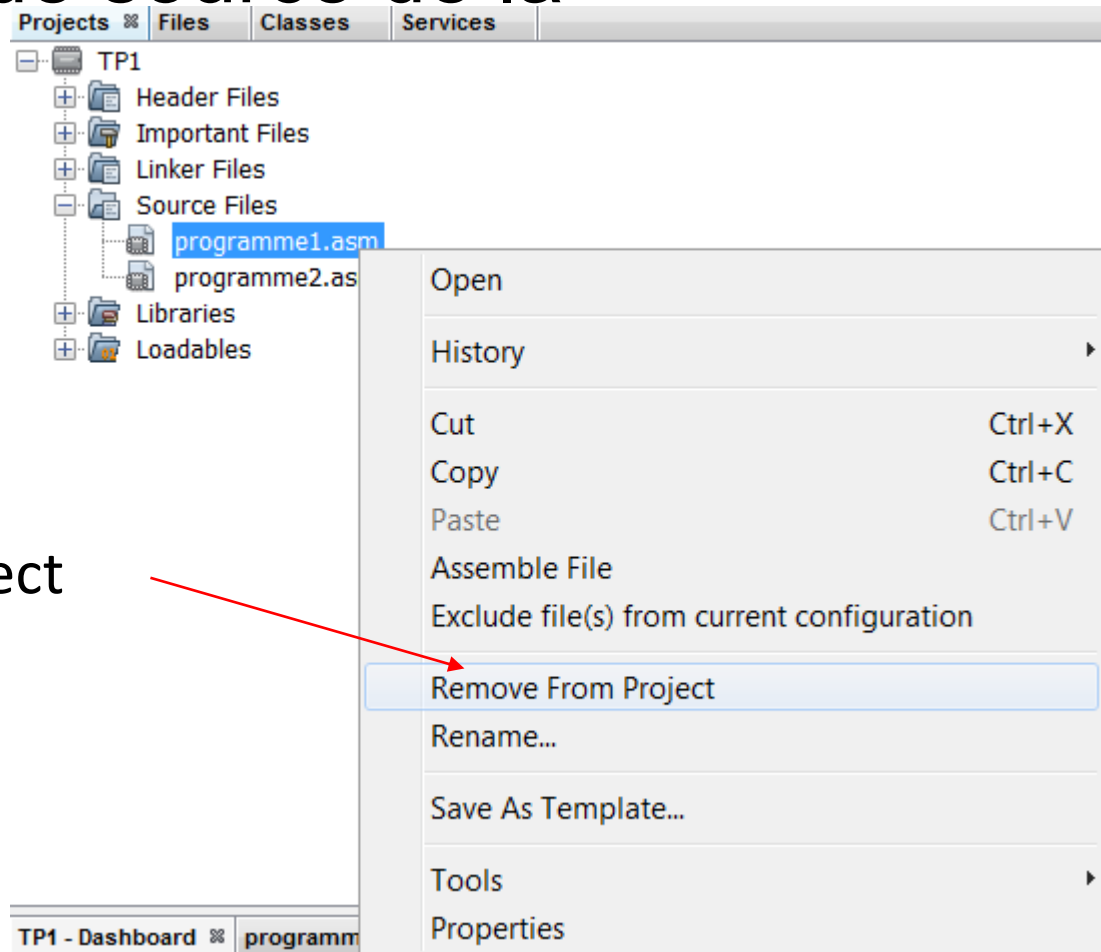


# Changer code source

- Pour l'instant, vous ne pouvez travailler que sur un seul code source. Il faut donc enlever l'autre code source de la configuration.

- Sélectionner le programme à enlever
- Puis cliquer droit
- Cliquer sur

Remove From Project



# Chap4 :

# Travailler sur maquette

# Travailler sur maquette

- Connecter le boîtier PICkit4 au PC par le port USB
- Connecter le boîtier PICkit4 à la carte
- Alimenter la carte et mettre sur ON.

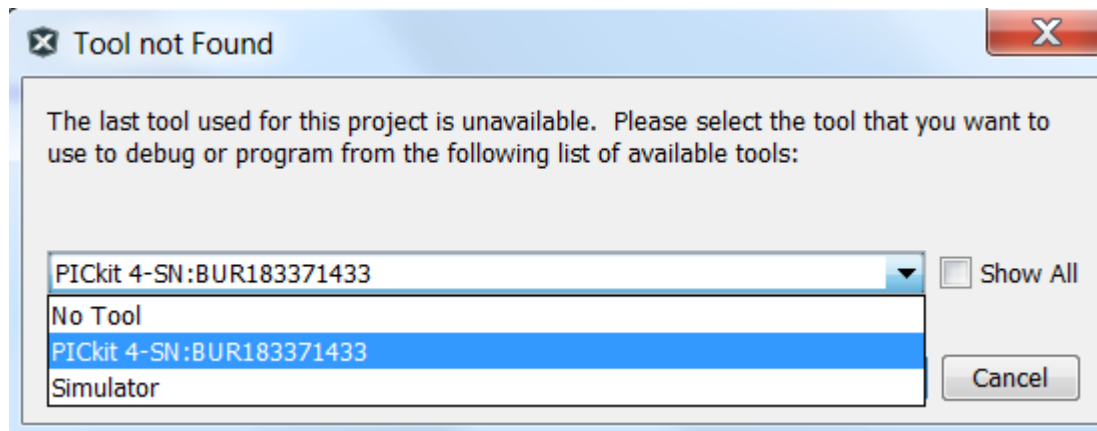
# Travailler sur maquette

- Cliquer sur l'onglet Debug main project



# Travailler sur maquette

Vous devez l'option PICkit4 pour programmer une carte électronique.



# Travailler en mode Debug

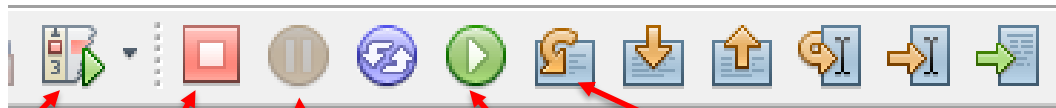
- Pour compiler:



La compilation se fait en cliquant sur l'icône de l'interface .

Si la compilation ne retourne pas d'erreurs, un fichier \*.hex est alors généré dans le répertoire /dist/default/production/ du répertoire de travail

- Sélectionner le mode Debug:(Debug Main Project)



Mode Debug

Fin Debug

Pause

Reset

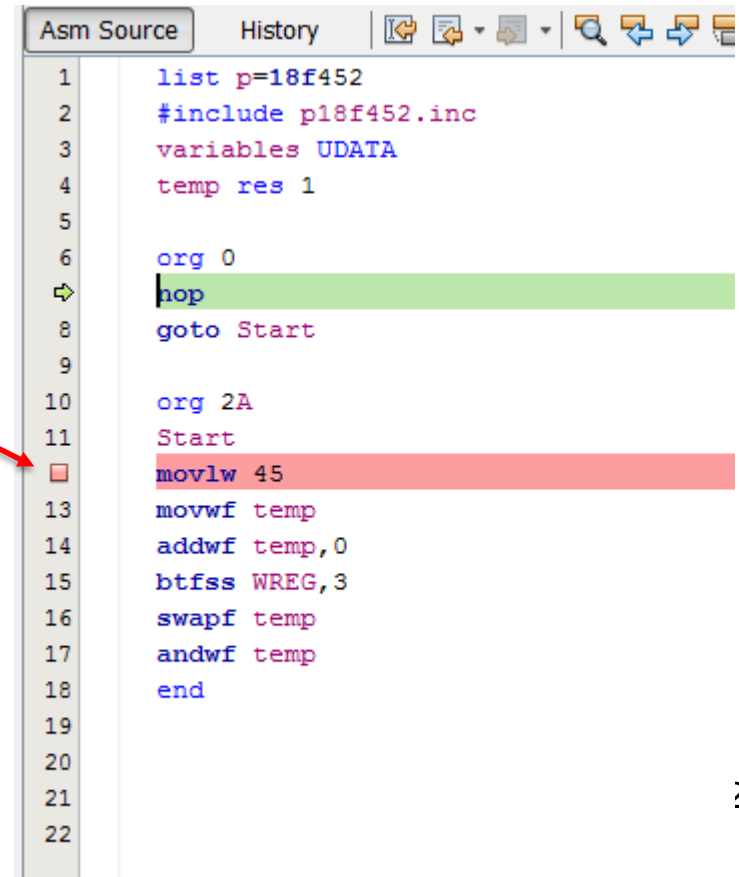
Run

Pas à pas

# Travailler en mode Debug

- Vous pouvez créer des points d'arrêt dans votre programme pour regarder les grandeurs , débbuger....

Double clique sur la ligne  
où on veut fixer le point  
d'arrêt



```
Asm Source    History
1  list p=18f452
2  #include p18f452.inc
3  variables UDATA
4  temp res 1
5
6  org 0
7  nop
8  goto Start
9
10 org 2A
11 Start
12 movlw 45
13 movwf temp
14 addwf temp,0
15 btfss WREG,3
16 swapf temp
17 andwf temp
18 end
19
20
21
22
```

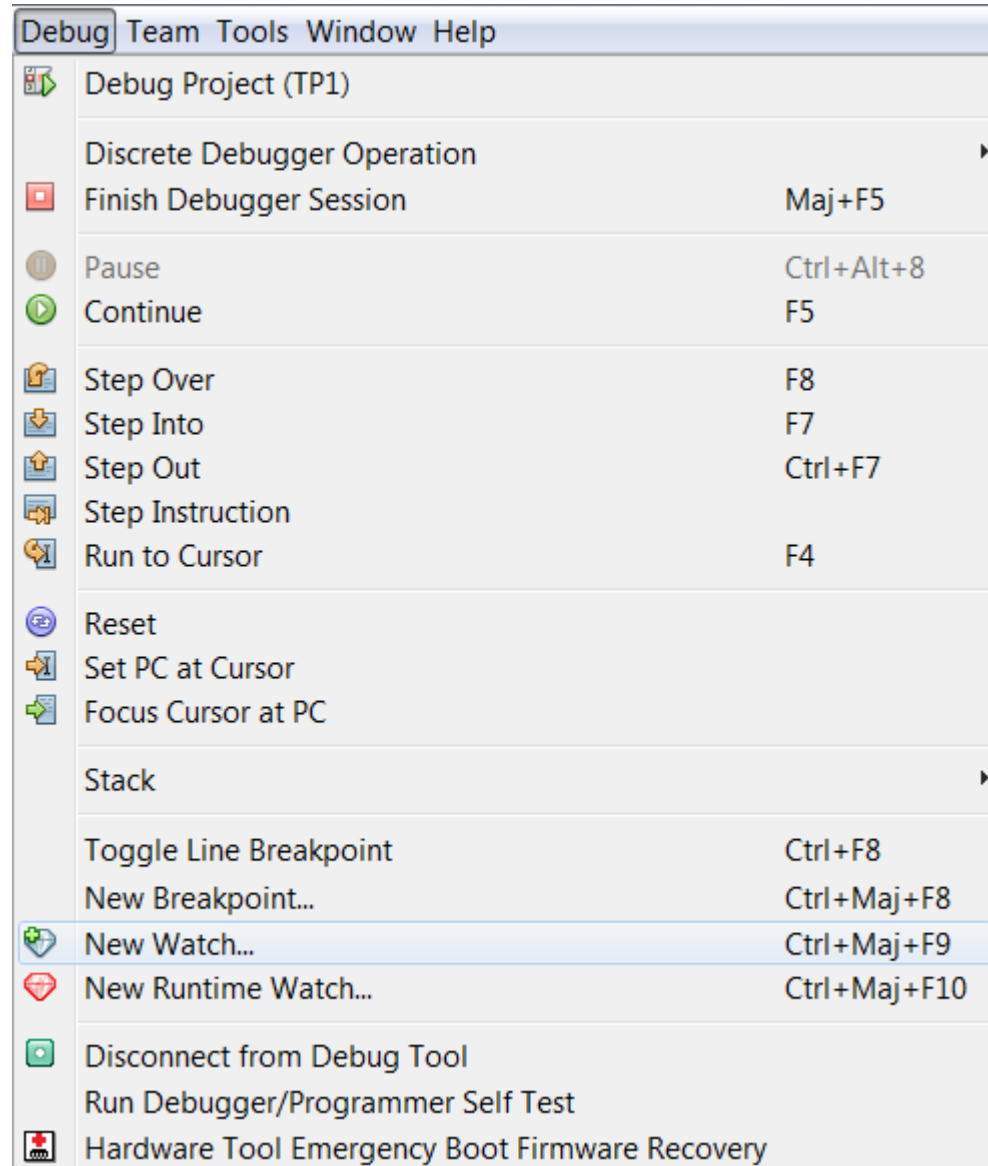


# Chap5 :

# Visualisation des grandeurs

# Visualiser des variables

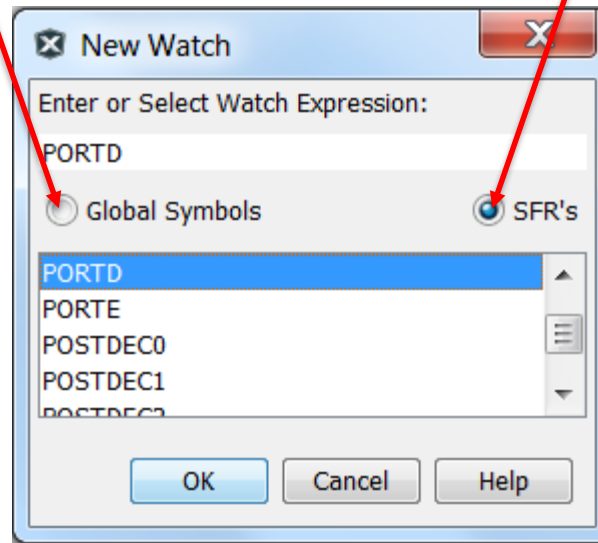
- Sélectionner Debug
- Puis cliquer sur New Watch



# Visualiser des variables

Vous pouvez sélectionner :  
une variable globale ou un registre réservé

variable globale:  
créer par vous en  
début de programme



Registre réservé:  
Fixé par le PIC par  
exemple WREG pour  
l'accumulateur

# Visualiser des variables

Vous pouvez visualiser les variables dans la zone de travail

Name	Type	Address	Value
ADRES	SFR	0xFC3	0x0000
temp	(1) Bytes (Select Size?)	0x80	0x00
PORTD	SFR	0xF83	0x00

<Enter new watch> (SFR) 0x00

Nom

Type

Adresse de la variable dans la mémoire de données

Valeur de la variable

Rmq: en cliquant droit sur Value, vous pouvez ajouter ou changer la base d'affichage de la donnée : binaire, hexadécimal, décimal

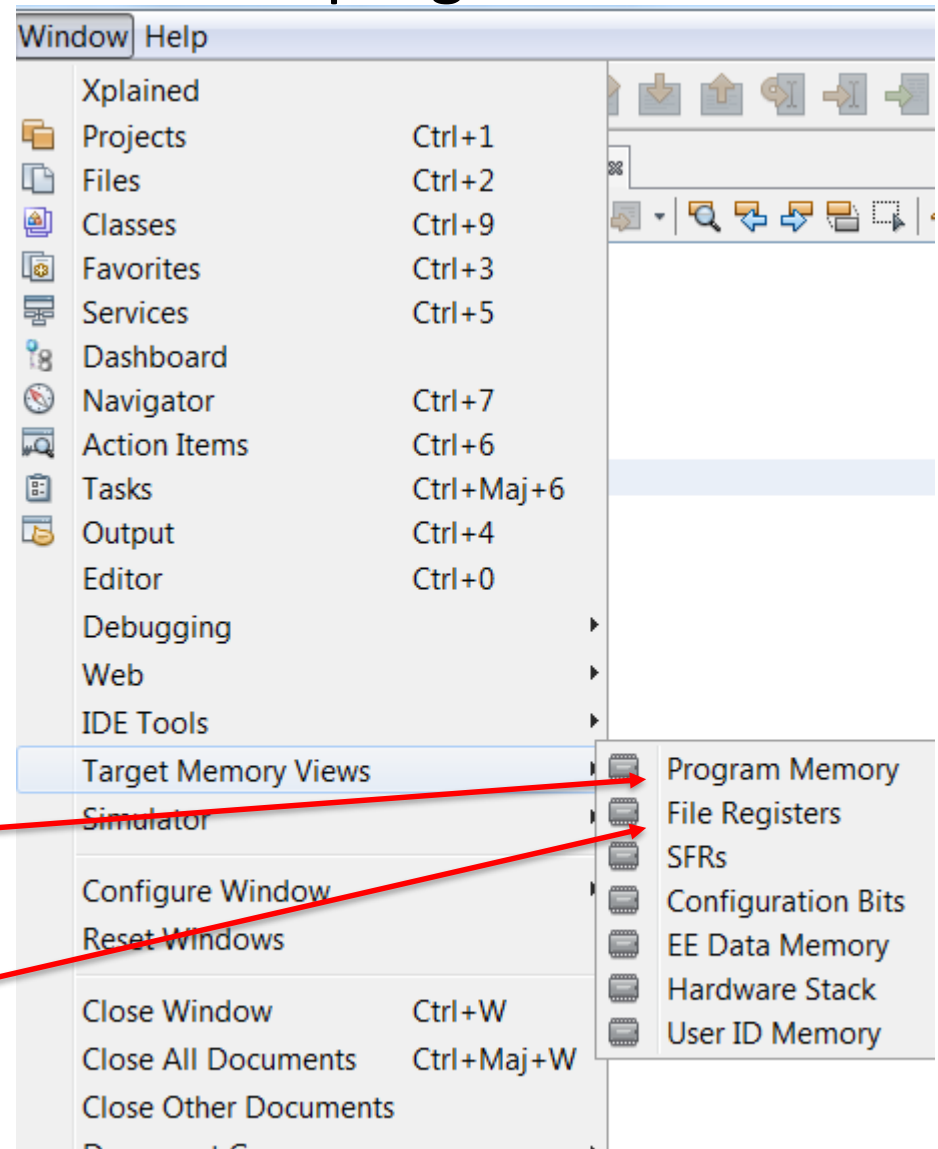
# Visualiser des mémoires

Vous pouvez visualiser les mémoires de programmes et de données.

- Sélectionner Window
- Puis cliquer sur Target Memory Views

Mémoire de programme

Mémoire de données



# Visualiser des mémoires

Vous pouvez visualiser les mémoires dans la zone de travail


Mémoire de données

Variables

Output - Project Loading Warning

Program Memory

File Registers %



Address

00

01

02

03

04

05

06

07

08

09

0A

0B

0C

0D

0E

0F

ASCII

000

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

010

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

020

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

030

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

040

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

050

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

060

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

070

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

080

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

090

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

0A0

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

00

.....

Memory

File Registers


Format

Hex

Variables

Output - Project Loading Warning

Program Memory %



	Line	Address	Opcode	Label	DisAssy
➡	1	0000	FFFF		NOP
	2	0002	FFFF		NOP
	3	0004	FFFF		NOP
	4	0006	FFFF		NOP
	5	0008	FFFF		NOP
	6	000A	FFFF		NOP
	7	000C	FFFF		NOP
	8	000E	FFFF		NOP
	9	0010	FFFF		NOP
	10	0012	FFFF		NOP
	11	0014	FFFF		NOP

Memory

Program Memory ▼

Format

Code ▼

Mémoire de programme