

Créer un dossier et nommer le judicieusement (PAMMPLABX par exemple).

TOUS LES FICHIERS ISIS8 et MPLABX devront être **placés dans ce dossier**.

1. Créer une carte de test virtuelle avec ISIS8

Ouvrir ISIS8

Fichier -> nouveau projet

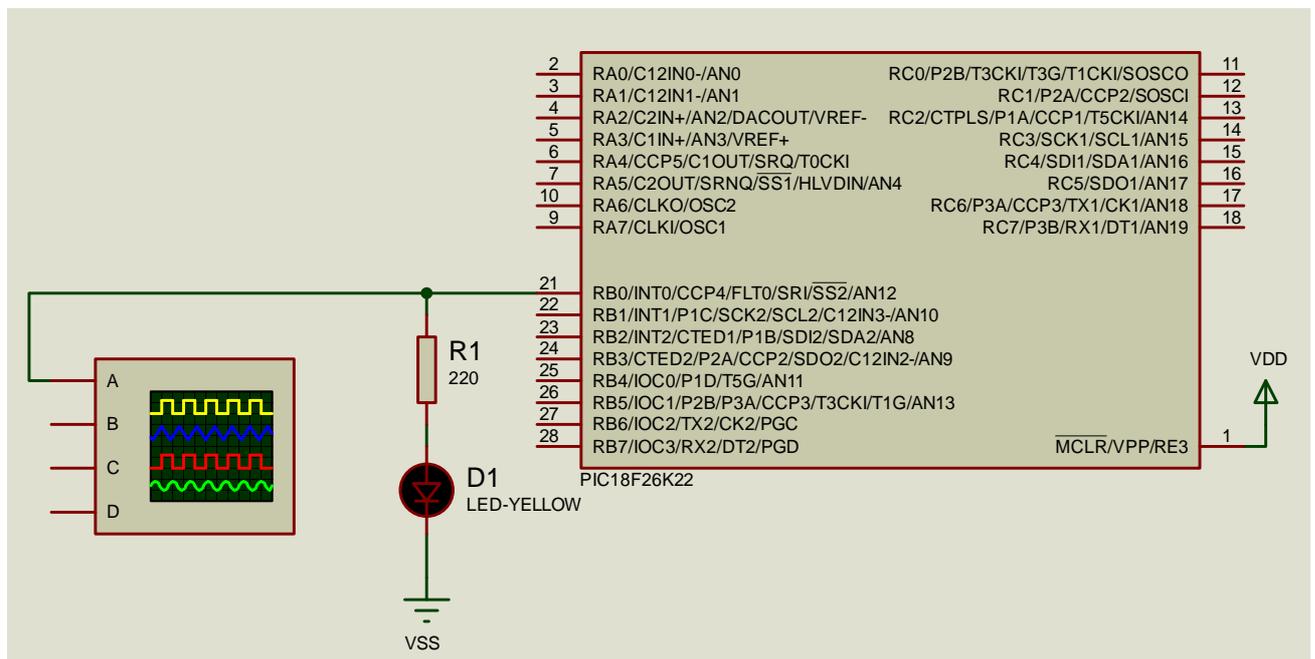
Nommer votre projet, exemple : testISISMPLABX .pdsprj

Sélectionner le dossier crée précédemment.

Créer un schéma à partir du gabarit Landscape A4.

Pas de projet circuit imprimé ni FIRMWARE

Créer ce simple schéma avec un PIC18F26K22 :



Enregistrer le projet ISIS8 **dans le dossier du projet**, fermer ISIS.

Il est très pratique de disposer facilement du datasheet du microcontrôleur. **Télécharger** le datasheet du PIC18F26K22 et **l'enregistrer dans le dossier du projet**

Lancer MPLABX

File -> New Project Microchip Embedded – Standalone Project

Famille PIC18 : sélectionner le PIC18F26K22

Hardware Tool : Proteus VSM Viewer (ou PICKIT3 ou ICD3 si le debug est fait sur le matériel)

Compiler : Eviter le compilateur C18 maintenant obsolète, **choisir la version la plus récente du XC8**.

Nommer le projet MPLABX « **essai_MPLABX_ISIS8** », **sélectionner comme destination le dossier du projet**.

Cocher « Set as main project » et « Use project location as the project folder »

Ne pas modifier "Encoding ISO-8859-1"

Finish ... La carte de simulation de test est terminée, la deuxième étape consiste à créer un programme et le tester sur cette carte

2. Créer un programme sur MPLABX avec simulation sur ISIS8

Ouvrir MPLABX

Créer le programme de test ci-dessous :

File -> New File, C, C source File. Nommer le fichier testISISMPLABX et vérifier qu'il va bien être enregistré dans le dossier du projet.

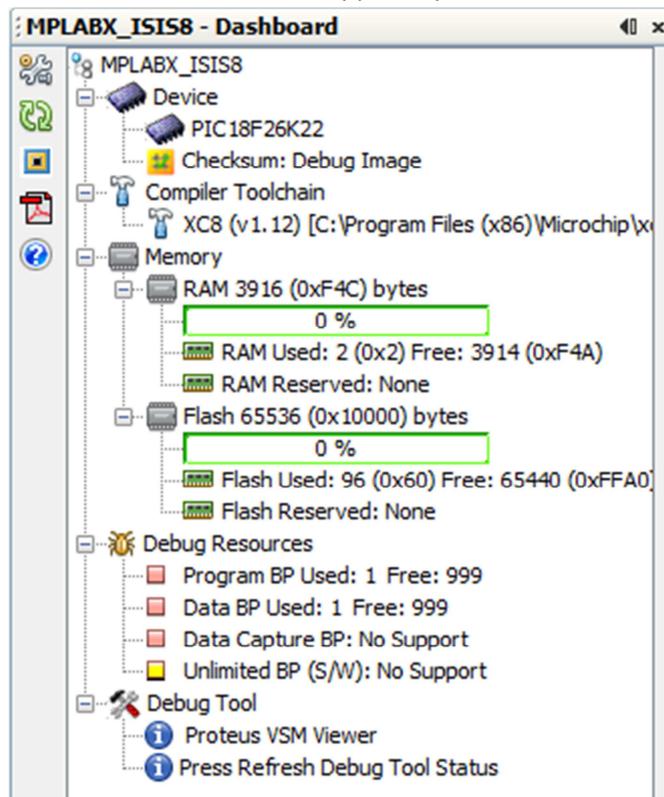
Copier le code :

```

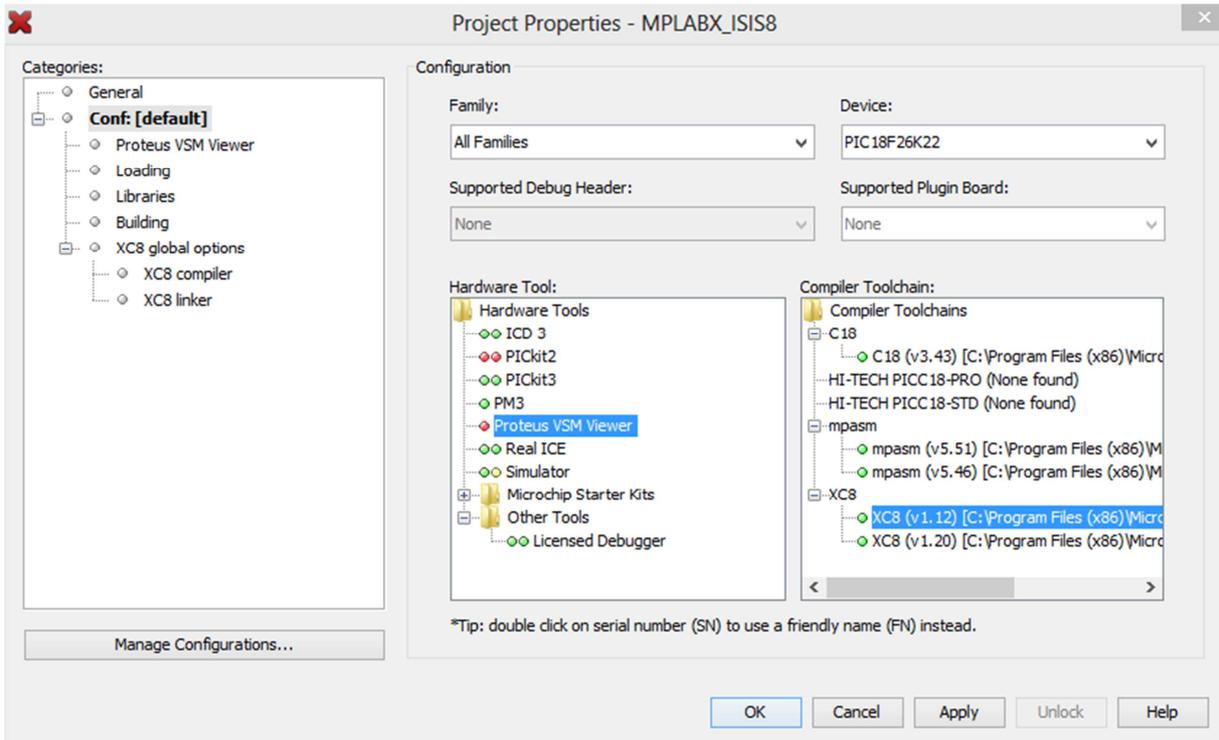
1 // PROGRAMME TEST MPLABX - XC8 - ISIS8
2 // CD 06-2013
3 #include <p18f26k22.h>
4 // horloge interne, sans surveillance, chien de garde désactivé, PORTB numerique
5 #pragma config IESO = OFF, FOSC = INTIO67, PRICLKEN = ON, FCMEN = OFF, WDTEEN=OFF, PBADEN=OFF
6
7 void tempo(void) {
8     unsigned long i;
9     i = 500000;
10    while (i--);
11 }
12
13 void main(void) {
14     OSCCON = 0b01110000; // intosc 16MHz, primary clock ;
15     OSTUNE = 0b01000000; // PLL active
16     TRISBbits.TRISB0 = 0;
17     PORTBbits.RB0 = 1;
18     while (1) {
19         PORTBbits.RB0 = ~PORTBbits.RB0;
20         tempo();
21     }
22 }

```

La fenêtre Dashboard indique la configuration du projet ainsi que l'occupation mémoire du microcontrôleur. (Si elle n'apparaît pas : Windows -> Dashboard)



Cliquer sur le logo « clé » pour éditer la configuration.
 Cliquer sur le logo Acrobat Reader et indiquer l'emplacement du fichier du datasheet du microcontrôleur (préalablement téléchargé dans le dossier du projet).



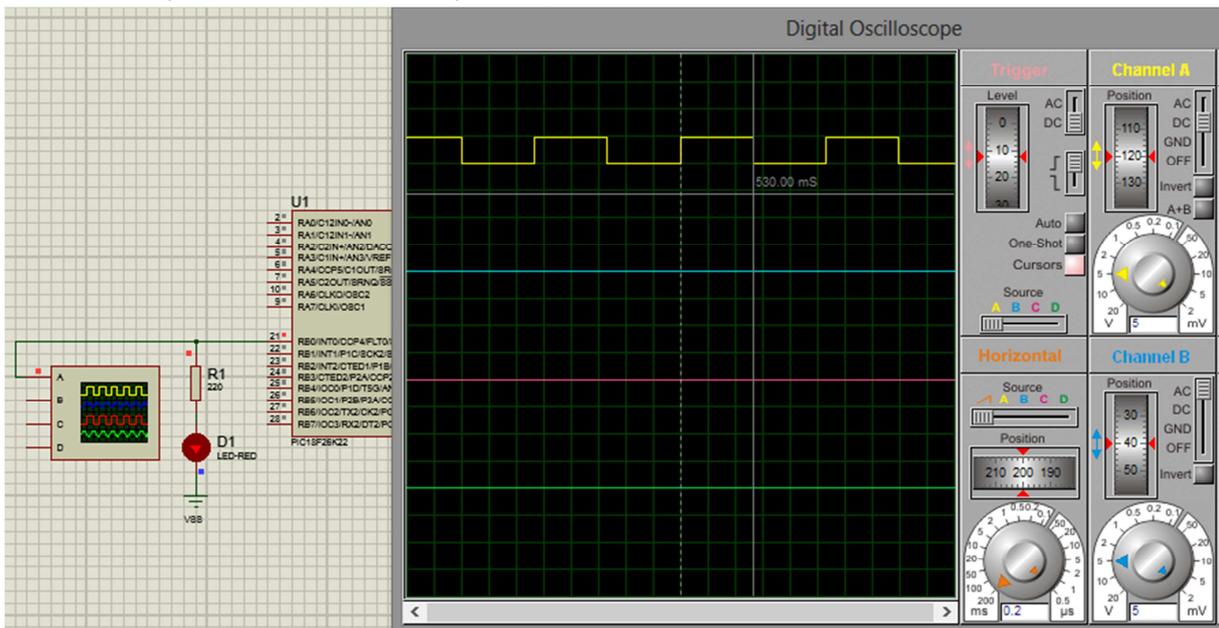
Le schéma ISIS8 doit être indiqué :



3. Tests et mise au point

Cliquer ici  pour compiler et lancer le debug

MPLABX indique le résultat de la compilation, lance ISIS8 et débute la simulation.



Il est possible de contrôler la simulation



Respectivement :

Fermer le debugger

Mettre en pause

Réinitialiser le compteur de programme à l'adresse 0x0000 (RESET)

Lancer le programme (RUN)

Pas à pas par-dessus les fonctions

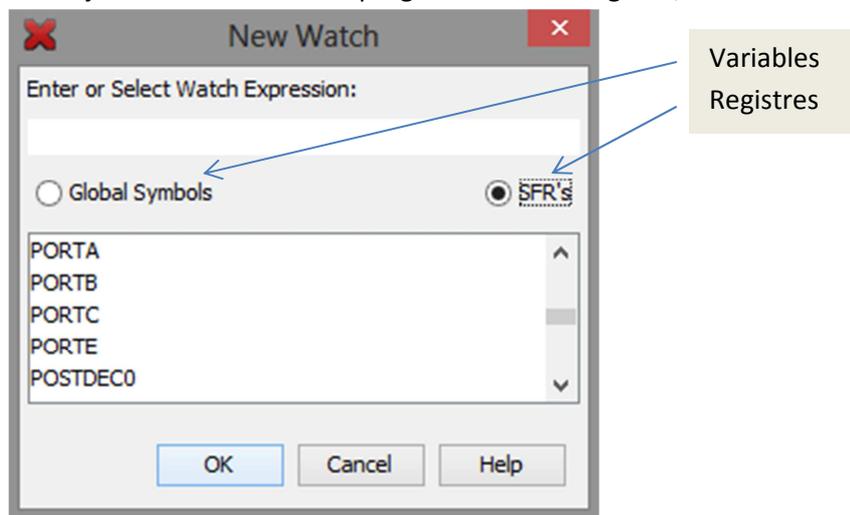
Pas à pas avec passage dans les fonctions.

Run jusqu'au curseur

Pour visualiser variables et constantes, Windows -> Debugging-> Variables

Output		Call Stack		Breakpoints		Variables		Watches	
Name	Type	Address	Value	Decimal	Binary				
PORTB	SFR	0xF81	0x00	0	00000000				
TRISB	SFR	0xF93	0xFE	254	11111110				
<Enter new watch>									
PORTBbits	struct	0xF81							
u1	struct	0xF81							
RB0	unsigned char	0xF81	0x00	0	00000000				
RB1	unsigned char	0xF81	0x00	0	00000000				
RB2	unsigned char	0xF81	0x00	0	00000000				

Pour ajouter une variable du programme ou un registre, clic-droit New Watch ?



Pour placer un point d'arrêt, cliquer sur le numéro de ligne pour faire apparaître un carré rouge.

```

void tempo(void) {
    unsigned long i;
    i = 500000;
    while (i--);
}
    
```

De même pour le retirer