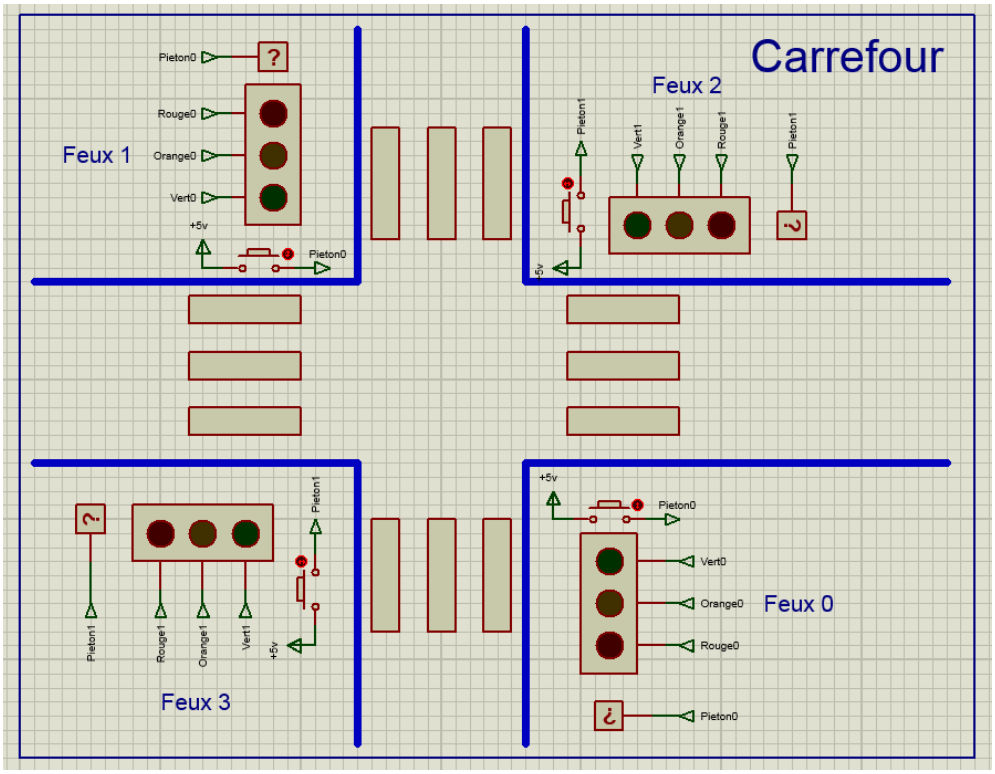
TP	Séquence : Programmation	1ere STI2D SIN
CI : 04 Gestion de l'information	*Activité : Gestion d'un carrefour*	Lycée Louis Rascol

- **Connaissances du référentiel :** Traitement d'une information numérique, Diagramme états-transitions pour un système événementiel, Implémentation d'un programme dans un composant programmable, Architecture client-serveur.

- **Compétences du référentiel :** CO7, CO8, CO9.sin1, CO9.sin4

Dans la même optique que l'activité du chapitre précédent, on souhaite programmer la gestion d'un carrefour. Un **carrefour à feux** est une intersection dont le trafic est réglé par des feux de signalisation lumineux pilotés par un contrôleur. Le réglage des cycles de feux doit permettre d'assurer la sécurité des automobilistes et des piétons tout en permettant un débit maximal.

1. Présentation



Le carrefour étudié possède 4 feux, sachant que les feux opposés par rapport au carrefour ont le même fonctionnement **: Feux0 fonctionne avec Feux1 et Feux2 fonctionne avec Feux3**. **Vous n'avez juste qu'à coder 2 feux, soit 8 signaux.**

Signaux	Pin du uC	Feux associés
Vert0	C0	Feux0-Feux1
Orange0	C1	Feux0-Feux1
Rouge0	C2	Feux0-Feux1
Pieton0	C3	Feux0-Feux1
Vert1	D0	Feux2-Feux3
Orange1	D1	Feux2-Feux3
Rouge1	D2	Feux2-Feux3
Pieton1	D3	Feux2-Feux3

De plus, vous avez à disposition 4 bouton-poussoir symbolisant la demande piéton associé à chaque feu. **Comme précédemment vous n'avez qu'à coder 2 demandes piéton : Pieton0 et Pieton1.**

Signaux	Pin du uC	Feux associés
Pieton0	B0-INT0	Feux0-Feux1
Pieton1	B1-INT1	Feux2-Feux3

2. Fonctionnement

Le fonctionnement attendu est le cycle suivant :

3 sec	3 sec	3 sec	3 sec	3 sec	3sec
V1		O1	R1		
R0			V0		O0

Le cycle reprend en boucle tant que le circuit est alimenté.

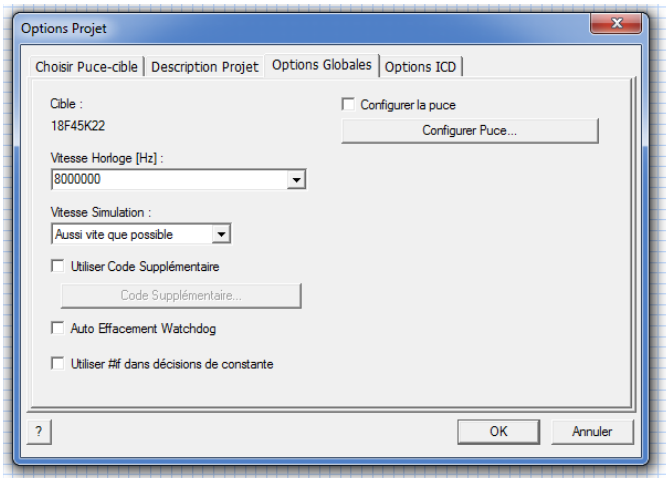
3. Travail à faire

Le TP est noté par validation successives
Appeler l'enseignant pour valider une fois l'algorithme fait et le programme fonctionnant sur ISIS
2 étapes : 3.1 et 3.2
Bon courage !

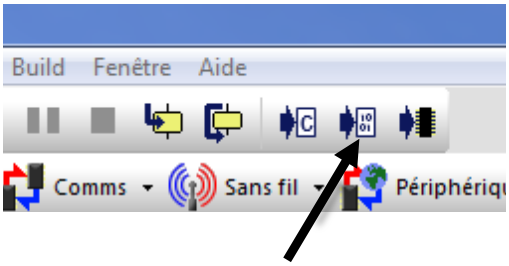
3.1. Codage du système sans demande piéton

En vous aidant de la description précédente utilisez le logiciel Flowcode afin de faire fonctionner le carrefour :

- Ouvrez Flowcode PIC et choisissez le microcontrôleur **PIC 18F45K22**
- Dans les paramètres : Build->Options Projet, **réglez la vitesse d'horloge à 8 Mhz**

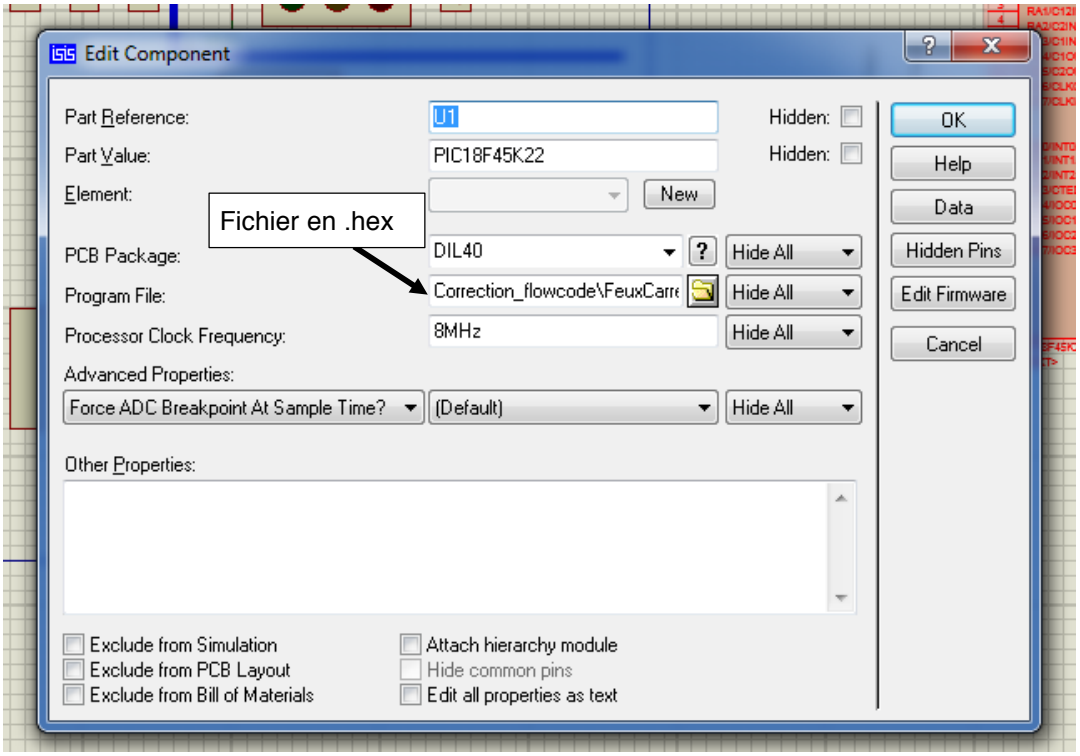


- Réalisez votre programme, en utilisant les pins du microcontrôleur cités précédemment.
- Compilez le programme en .hex



Compilation en .hex

- Ouvrez le fichier projet ISIS : **Feux_de_carr_flowcode.pdsprj** et insérez votre code en .hex (situé dans le même dossier que le fichier de projet Flowcode) dans le microcontrôleur. Clic droit sur celui-ci -> Edit properties -> Program File.



- Vérifiez le bon fonctionnement du système en appuyant sur Play.

3.1. Codage du système avec demande piéton

En prenant pour base le programme précédent ajoutez les demandes piétons :

- Lors d'un appui sur le bouton de demande piéton, le programme enregistre la demande et attend la fin du cycle**, il met tous les feux à l'orange durant 3 sec et au rouge durant 9 sec, puis le cycle recommence au début.**
- Pour mémoriser la demande piéton nous allons utiliser 2 interruptions : INT0 sur B0 et INT1 sur B1

Fonctionnement d'une interruption

Une interruption sur un microcontrôleur est déclenchée par un évènement interne (fin de comptage d'un timer …) ou externe (évènement sur une pin d'interruption…). Cette interruption arrête le programme à l'endroit où il est et déclenche le sous programme d'interruption associé. Une fois le sous programme d'interruption achevé le programme reprend à l'endroit où il c'était arrêté.

- Dans un premier temps réalisez une macro passage_pieton qui **met tous les feux à l'orange durant 3 sec et au rouge durant 9 sec, puis met à 0 une variable de type Bool : demande_pieton**
- Créez les 2 sous programmes d'interruption **Interrupt0** et **Interrupt1,** qui mettent à 1 la variable : **demande_pieton**
- Rajoutez dans la branche principale l'initialisation des 2 interruptions
- Rajoutez la condition sur la variable : **demande_pieton**
- Compilez en .hex et testez le fonctionnement sur ISIS