1 Utilisation de l'environnement de programmation et de développement

Description générale

- **Date du TP :** première semaine ;
- **Durée du TP :** 4,5h (3semaines) ;
- **Matériels nécessaires** : Carte d'experimentation, microcontrôleur PIC 16f877, programmateur, des composants électroniques ;
- **Logiciels utilisés :** Complilateur C PCW, Simulateur ISIS, IC Writer ;
- **Public cible :** 4ième niveau d'études de technicien supérieur en informatique industrielle ;
- **Partie du cours en rapport avec le TP :** Classification architecturale des systèmes temps reel, les microcontrôleurs.

Objectifs

- Se familiariser avec les outils de programmation et de développement ;
- Se familiariser avec la programmation en C des microcontrôleurs PIC ;
- Savoir les instructions à suivre pour mettre en en place une application à base de microcontrôleur.
- Maitrise des architectures des microcontrôleurs Microchip PIC16.

Pré requis

Notions générales sur :
- L'électronique ;
- Algorithmique ;
- Programmation en C ;
- Les microcontrôleurs.

1.1 Introduction

Nous avons choisi le compilateur C de CCS, dans sa version PCW, qui est tout à la fois la plus complète et la plus performante, puisqu'elle supporte toutes les familles de PIC et dispose d'un environnement de programmation et de développement intégré.

1.2 Installation et configuration de l'environnement

Si vous avez acheté le compilateur CCS, celui-ci vous a été fourni sur un CD-ROM avec toutes les instructions d'installation necessaires. Dans le cas contraire, vous avez dû télécharger depuis le site de CCS un fichier baptisé demoupd.exe qui est le fichier d'installation. Même si vous n'envisagez pas dans l'immédiat d'utiliser le compilateur C avec MPLAB, ce que nous verrons à la fin de ce TP, nous vous conseillons néanmoins d'installer MPLAB avant d'installer le compilateur C ; cela facilitera ensuite l'intégration de ce dernier dans MPLAB. Pour cela, il vous suffit de télécharger MPLAB depuis le site internet de Microchip (www.microchip.com). Lancez son execution, ce qui aura pour effet d'installer MPLAB sur votre machine. Dans un premier temps, acceptez toutes ses propositions sans les modifier car elles sont compatibles de la majorité des situations. Vous pouvez alors installer le compilateur en exécutant le fichier d'installation préalablement téléchargé depuis le site de CCS. Ici aussi, acceptez tout ce qui vous est proposé par défaut. Lorsque cette installation est terminée, et contrairement à ce qui est indiqué dans une des fenêtres qui peut s'ouvrir à la fin de l'installation, l'intégration dans MPLAB n'est pas encore effective (sauf modification réalisée sur le compilateur depuis la rédaction de ces lignes). À ce stade des operations vous etes prêt à écrire votre premier programme en C, ce que nous allons faire sans plus tarder tout en décrivant les principales fonctionnalités de l'environnement de développement PCW de CCS.

1.3 Notre premier projet en C

Lancez PCW grâce au raccourci que la procedure d'installation a dû placer sur votre bureau. La fenêtre vide qui s'ouvre alors est celle a partir de laquelle va s'effectuer tout votre travail.

1.3.1 Définition du projet

Notre compilateur C travaille avec la notion de projet ; un projet étant en fait un méta fichier qui renferme tous les fichiers d'une même application. La première opération à réaliser est donc de définir ce projet. PCW nous permet de le faire en mode manuel mais dispose également d'un " magicien " le fameux wizard, accessible par le menu " Projet - New- PIC Wizard ". Une fenêtre d'exploration de répertoire s'ouvre alors pour vous permettre de saisir ce qui sera le nom du fichier source principal de votre projet. Choisissez le répertoire de votre choix et donnez un nom à votre fichier ; par exemple

PremierProjet .c puisqu'il s'agit d'un fichier source en C.
Comme le montre la figure 1.1 une nouvelle fenetre apparaît alors. Elle comporte de très nombreux onglets dans sa partie basse correspondant à la définition matérielle L'écran de départ du wizard ou magicien comporte de nombreux onglets de votre application. Avant de les ouvrir afin de les renseigner, il est donc indispensable de disposer du schéma de cette derniere. En effet, comme nous allons le constater dans quelques instants, ces onglets vont vous demander de préciser un certain nombre de connexions réalisées sur le PIC.

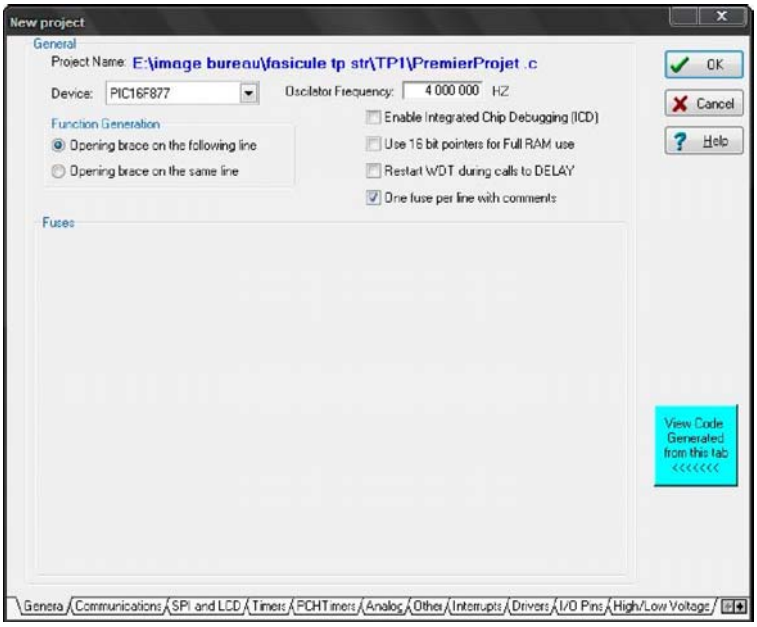

FIG. 1.1 – Fenêtre du PIC Wizard

Pour que cet travail soit interactif, nous vous proposons de réaliser le petit projet fort simple présenté figure 1.2.
Même si cela semble un peu luxueux vu ses ambitions, qui sont de réaliser un petit chronometre avec un affichage en binaire sur de simples LED ; un 16F877 y est utilisé. Ce schéma va nous permettre de renseigner la majorite des différents onglets que nous a présentés le wizard du compilateur, en commençant par l'onglet "General" qui est le premier a s'être ouvert comme vous avez pu le constater en figure 1.1. Cet onglet permet de choisir le type de processeur utilisé, ici un 16F877, ainsi que sa fréquence d'horloge et ses principaux paramètres de configuration. Selectionnez les mêmes informations que nous, visibles 1.1. Notez que vous pouvez faire apparaître le code généré par celles-ci en cliquant sur le bouton bleu vif baptisé "View Code I Generated from this tab", ce que nous avons fait sur cette figure à titre d'exemple. L'onglet suivant, appelé "Communications" permet de valider ou non l'UART interne ainsi que l'eventuel port I2C dont peut être muni le circuit, et de définir un certain nombre de paramètres les concernant l'un et l'autre comme indiqué sur la figure 1.3.
Notez que si vous choisissez d'autres pattes d'entrées/sorties que celles normalement prévues par le circuit pour son UART interne, le compilateur intégrera à votre programme

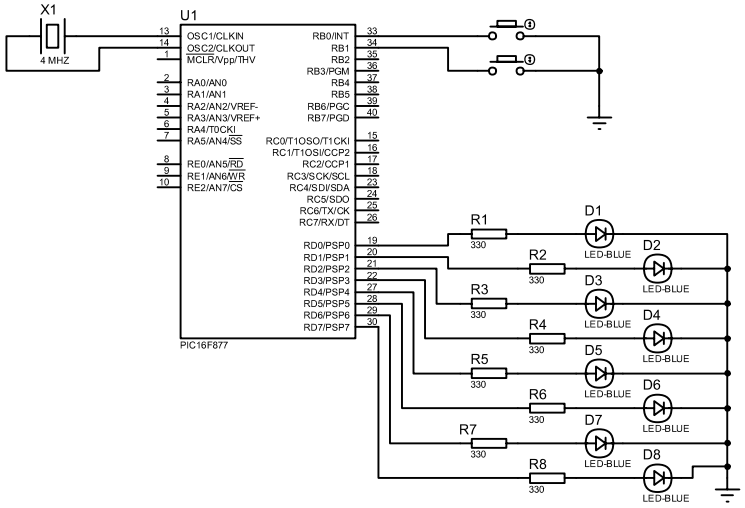

FIG. 1.2 – Schéma du montage utilisé à titre d'exemple

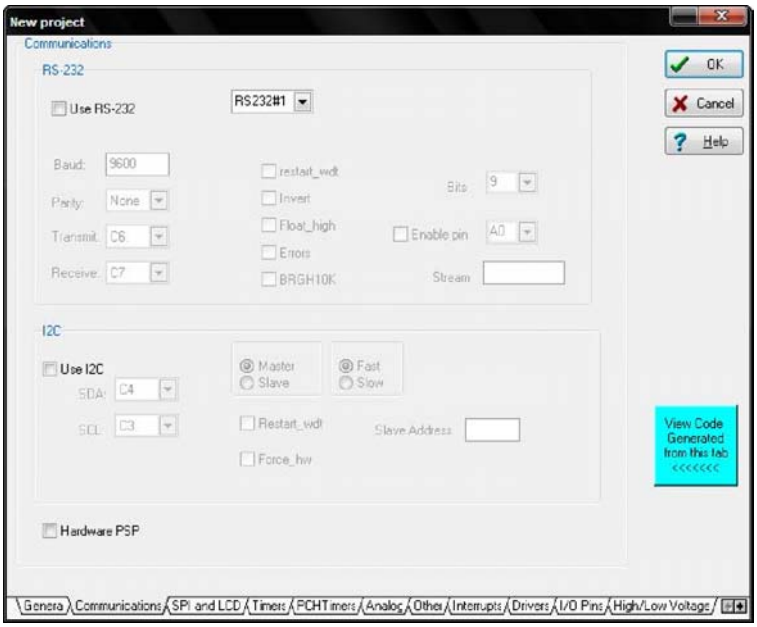

FIG. 1.3 – L'onglet de paramétrage de l'interface série RS 232 et du bus I2C

un UART par logiciel qui ne fera pas alors appel à l'UART interne. C'est également le cas si vous faites appel à un PIC qui ne dispose pas d'un tel UART. Dans le cas de notre exemple, UART et I2C ne sont pas utilisés et les cases correspondantes seront donc décochées, sauf à générer du code inutile.
L'onglet suivant intitule SPI et LCD permet de taire de même mais pour l'interface SPI presente sur de nombreux PIC, ainsi que pour l'interface de pilotage directe d'afficheurs à cristaux liquides présente sur certaines versions spécialisées. Contrairement à ce qui existe

pour l'interface RS 232, l'interface SPI ne peut être ici que matérielle et aucune option de choix des pattes du PIC n'est donc proposée comme vous pouvez le constater figure 1.4

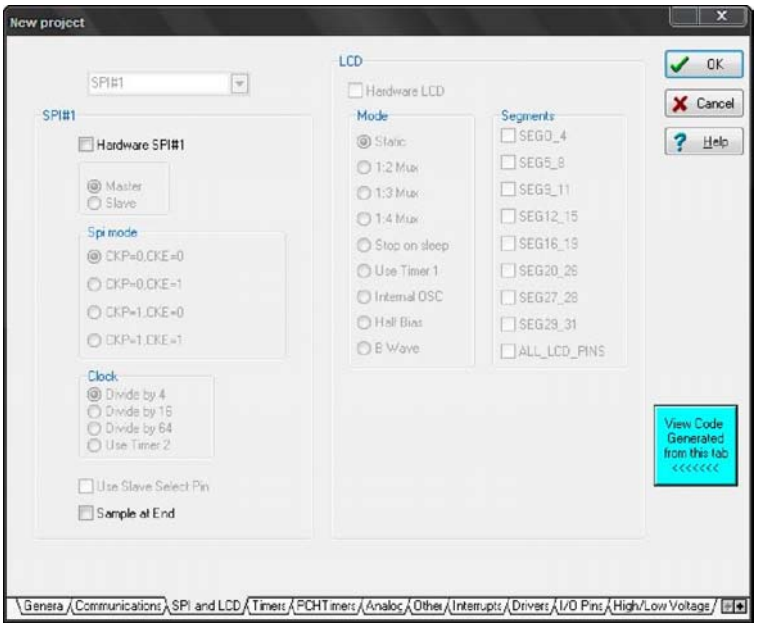

FIG. 1.4 – L'onglet de paramétrage de l'interface SPI et de l'interface LCD optionnelle

L'onglet suivant, baptisé "Timers", est très riche dans le cas du 16F877 car il permet de configurer tous les timers inclus dans le circuit ainsi que le timer chien de garde ou watchdog si celui-ci est utilisé. La signification des cases a cocher se comprend d'elle-même pour peu que vous connaissiez la structure des timers du PIC utilisé. Comme vous pouvez le constater à l'examen de la figure 1.5, cet onglet indique en outre, au niveau du libellé " overflow " et en fonction de la résolution choisie pour tel ou tel timer, quelle valeur atteindra son registre de comptage avant de déborder. Pour notre projet, nous laisserons les timers 1 et 2 non activés et le timer 0 avec les options proposées par défaut.
L'onglet "Analog", visible figure 1.6, permet quant à lui de définir le comportement du convertisseur analogique/digital intégré. Toutes les possibilites offertes par le circuit choisi apparaissent sous forme de cases à cocher dont le libellé est parfaitement explicite pour qui connaît les diverses options proposées par ces convertisseurs
L'onglet "Other" donne accès quant à lui à "d'autres" ressources du circuit comme son nom l'indique. Y figurent ainsi, comme vous pouvez le voir figure 1.7, la définition du mode de fonctionnement des modules de capture et de comparaison ainsi que, le cas échéant (mais il n'y en a pas dans le 16F877) celle du comparateur analogique. Contrairement à ce qu'indique la figure 1.7 sur laquelle nous avons validé une fonction de capture pour les besoins de l'exemple, vous laisserez la case CCPx vide de tout libellé.
L'onglet suivant baptisé "Interrupts" permet de définir les sources d'interruption qui seront autorisees dans le cadre du projet. En effet, le compilateur C de CCS gère intégralement ces dernieres et il lui faut donc connaître les differentes sources qu'il devra utiliser. Il suffit pour cela de cocher la ou les cases appropriées visibles figure 1.8.

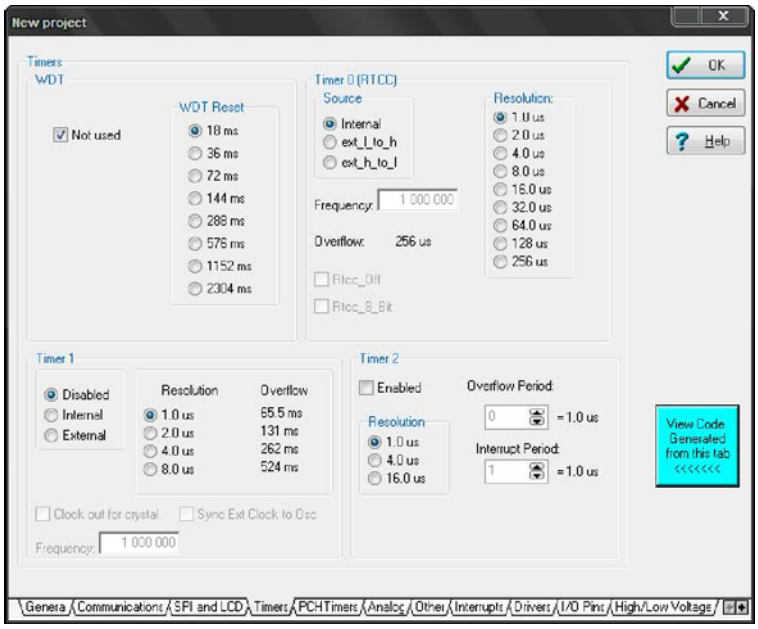

FIG. 1.5 – L'onglet très complet de paramétrage des timers

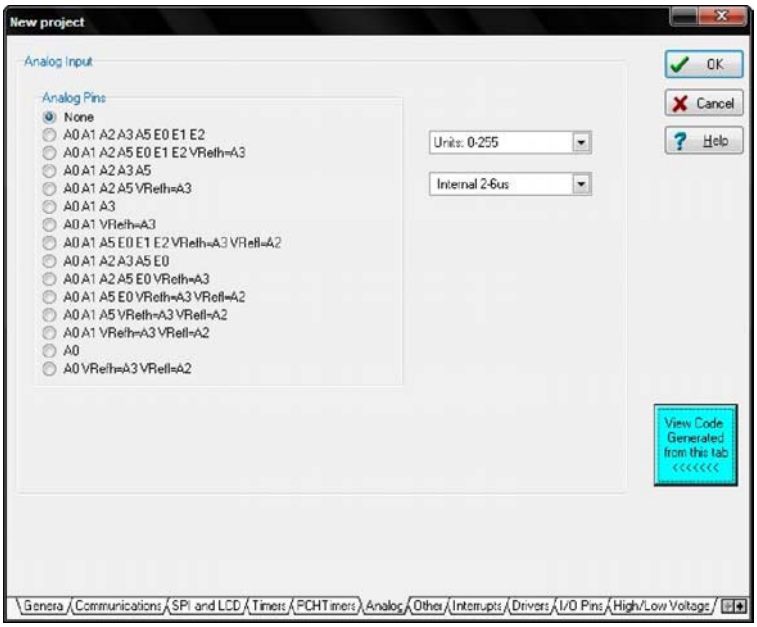

FIG. 1.6 – L'onglet de paramétrage du convertisseur analogique/digital permet de définir les entrées utilisées

L'onglet "Drivers" quant à lui permet d'inclure dans votre projet les fichiers sources destinés à un certain nombre de circuits ou de composants externes, généreusement fournis par CCS. Il suffit pour cela de cocher la ou les cases désirées étant entendu que la version de démonstration, pour des raisons de taille du fichier de téléchargement, ne propose que

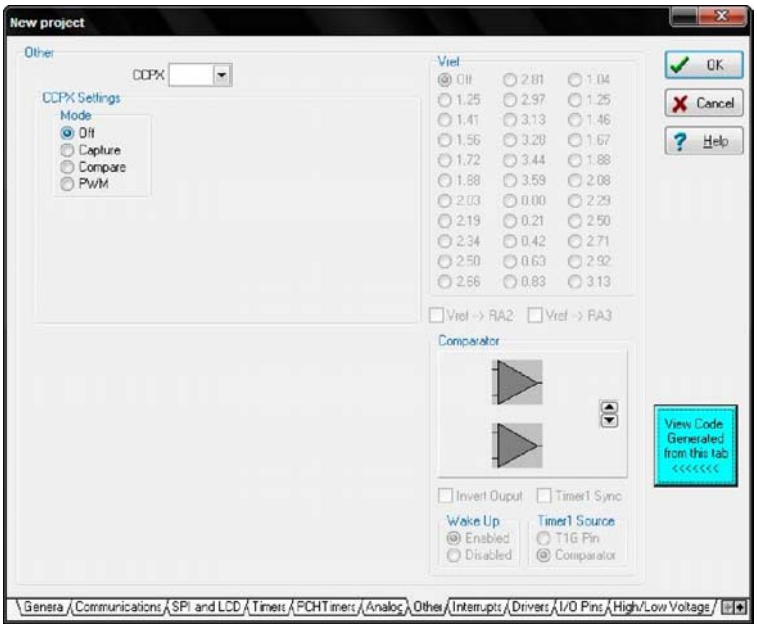

FIG. 1.7 – L'onglet des modules comparaison et capture et du comparateur analogique

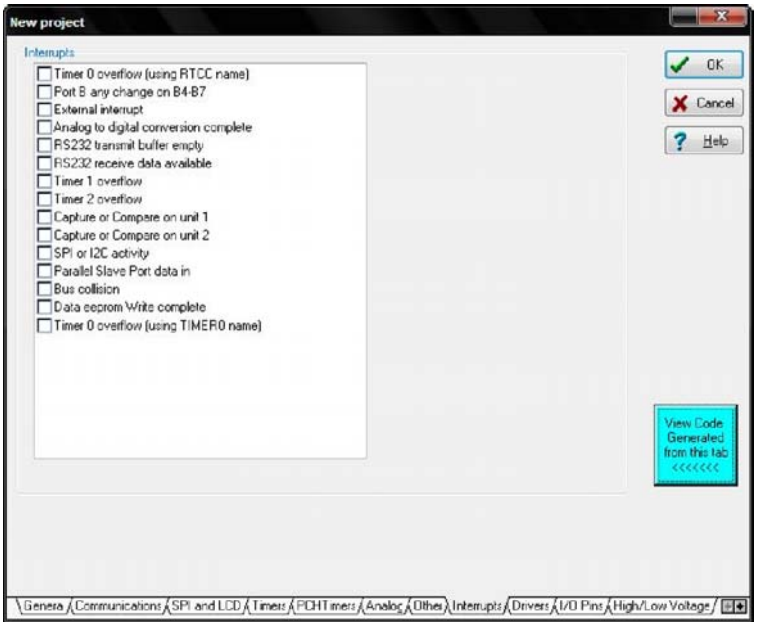

FIG. 1.8 – L'onglet de définition des interruptions

quelques-uns de ces fichiers comme le montre la figure 1.9.
L'onglet "I/O Pins" permet quant à lui de definir le sens de fonctionnement des différentes lignes de port du PIC choisi. En outre, lorsque ces lignes sont utilisées par une des ressources définies dans un des onglets précédents, elles sont automatiquement placées dans le sens de fonctionnement requis et le nom correspondant apparaît dans la case de droite

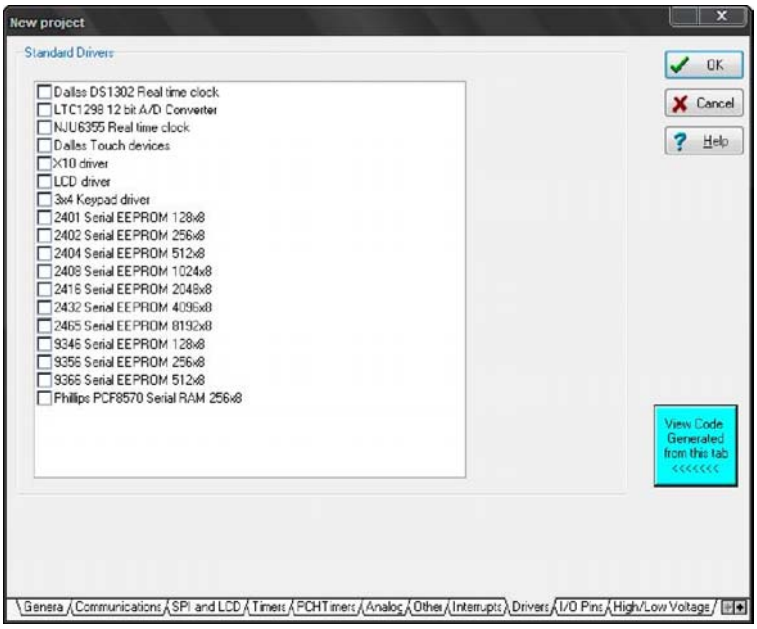

FIG. 1.9 – L'onglet de sélection des "drivers" externes, très incomplet dans le cas de la version de démonstration

de la ligne concernée, comme vous pouvez le voir en figure 1.10 . Vous validerez également les résistances de tirage (" pull up ") du port B, ce qui nous dispensera ainsi de toute résistance de rappel au positif de l'alimentation des poussoirs P1 et P2.

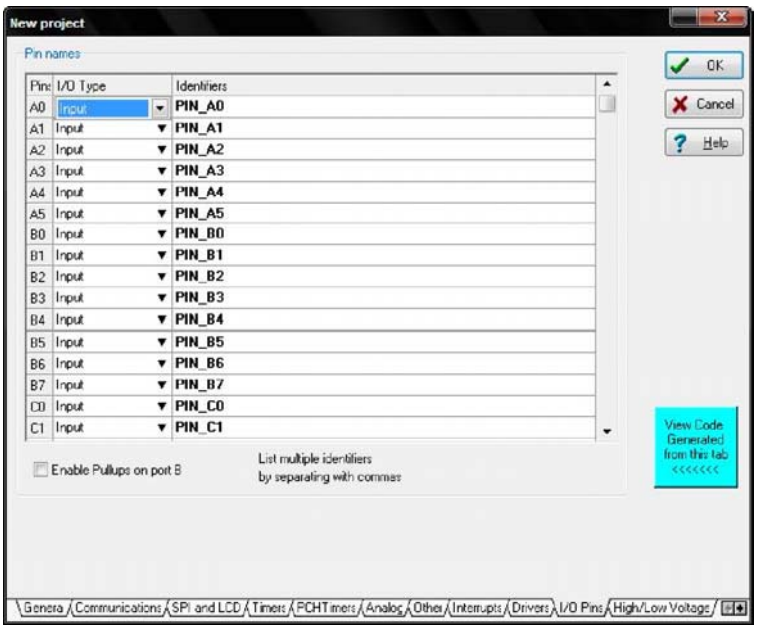

FIG. 1.10 – L'onglet " I/O Pins "

L'onglet "Header File" permet d'inclure dans le listing source de votre projet un certain

nombre de fichiers d'en-tête standards du langage C. En ce qui nous concerne, nous ne l'utilisons pas car il nous semble difficile, à ce stade du projet, de savoir desquels on va avoir réellement besoin. En outre, leur ajout manuel en cours d'écriture du programme ne pose vraiment aucun problème. Par contre, comme indiqué figure 11, nous cochons systématiquement la case ZERO_RAM qui fait ce que son nom indique au début de l'exécution du programme.

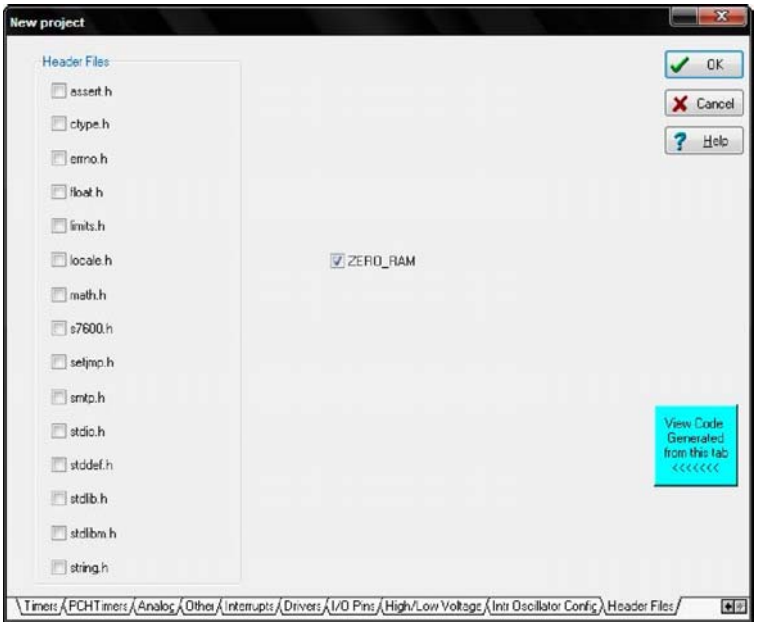

FIG. 1.11 – L'onglet de selection des fichiers d'en-tête à inclure dans le programme

Lorsque toutes ces définitions sont terminees, il suffit de cliquer sur " OK " au niveau du dernier onglet pour voir apparaıtre dans la fenêtre d'édition le code source généré par les différentes options précédemment choisies. Dans le cas de notre exemple.
Le wizard ou magicien a terminé son travail et a écrit pour nous le début du programme. Remarquez que tout ce que nous avons défini ne figure pas dans ce listing mais qu'un fichier externe s'y trouve inclus au niveau de sa premiere ligne. Ce fichier porte le même nom que notre fichier source mais avec le suffixe.h. C'est donc un fichier d'en-tête tout à fait classique en C. Vous pouvez le visualiser au moyen du menu "File - Open". Il apparaît alors dans un autre onglet de la fenêtre d'édition et vous pouvez basculer immédiatement du fichier.c au fichier.h en cliquant sur l'onglet correspondant. Cette façon de faire reste vraie pour tous les fichiers que vous pourrez être amenés à ouvrir dans l'environnement, ce qui permet de passer très vite de l'un à l'autre et d'y realiser sans difficulté des copier-coller si nécessaire.
Notez que le fichier premierProjet.h contient les informations qui "manquaient" sur l'écran de PremierProjet.c avec, en particulier, la définition des bits ou fusibles de configuration du PIC.

1.3.2 Édition du programme

Vous pouvez maintenant editer votre programme dans la fenêtre de PremierProjet.c. Nous vous proposons pour cela le listing 1 qui réalise un chronomètre simplifié (notez que ce listing est présenté ici une fois saisi et intégré dans le code source déjà généré par le compilateur PCW).

```
//****************Listing 1**********************************
#include "E:\image bureau\fasicule tp str\TP1\PremierProjet .h"
#ZERO_RAM
int i ;
void main() {
port_b_pullups(TRUE);
setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_OFF);
setup_psp(PSP_DISABLED);
setup_spi(FALSE);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1 (T1_DISABLED) ; setup_timer_2(T2_DISABLED,0,1);
while (1)
{
while (input(PIN_B0)); // Attente de mise au niveau bas de RBO
for (i=0;i<=255;i++)
{
output_D(i); // Affichage du comptage
delay_ms(100); // L'unité est le 1/10 de seconde
if (!input(PIN_B1)) // Si RB1 au niveau bas, arrêt du comptage
break; } } }
//****************fin du listing**********************************
```

Notre programme commence par attendre que l'on appuie sur le poussoir connecté sur B0 pour déclencher le comptage, avec une résolution d'un dixième de seconde, dont l'évolution est visualisée en binaire sur les LED connectees au port D. L'appui sur Bl arrête ce comptage et permet donc de prendre connaissance du temps écoulé entre l'action sur B0 et l'action sur Bl. C'est évidemment rudimentaire mais, pour prendre en main l'environnement de développement, il etait inutile de cumuler les difficultés en faisant appel à un programme plus complexe. Remarquez que l'éditeur dispose de diverses fonctions destinées a vous faciliter la saisie des programmes en C. Il propose en effet la coloration syntaxique et il met automatiquement en surbrillance de couleur verte les parenthèses ou les accolades qui, pour lui, sont appairées. Vous n'avez donc plus aucune excuse si vous en oubliez une ce qui, soit dit en passant, élimine de nombreuses erreurs de programmation bêtes propres au langage C. Toutes les autres fonctions d'un éditeur a fenêtre sous Windows sont évidemment disponibles mais elles sont plus classiques et nous vous laissons le soin de les découvrir au sein du menu " Edit ".

1.4 Travail demandé

1. Suivez les étapes, indiquées ci-dessus, pour créer votre premier projet en C.
2. compiler le programme avec PCW par la suite vérifier son bon fonctionnement à l'aide de la simulation par ISIS.
3. expliquer puis commenter le programme présenté dans le listing.
4. Charger le programme dans le microcontrôleur, situé dans la carte d'expérimentation.