TP n°1:

Objectifs :

Etre capable de créer un projet en C avec MPLAB
Savoir charger et exécuter un programme avec l'ICD2
Connaître la signification des bits de configurations du 16F877
Savoir rendre la carte cible autonome
Savoir utiliser une "Watch Window" pour voir les variables
Savoir utiliser les fonctions de la librairie biosdem2
Savoir utiliser les entrées/sorties logiques des ports A et B du 16F877.
Faire le lien entre le programme et le schéma de câblage de la carte.

1/Création d'un projet

- En utilisant le document annexe "Créer un projet pour la carte PICDEM avec MPLAB v6 et L'ICD 2", créer un projet C contenant le programme `tp1.c` fourni en annexe.
- Charger le programme sur la carte cible et exécutez le.

2/ Examiner et modifier les variables avec une "Watch Window"

Dans le programme tp1.c, la variable `cpt` est incrémentée lors de chaque appui sur le bouton. (La fenêtre `Watch` apparaît avec `View->Watch…`)

- Ajoutez les registres PORTB,TRISB ainsi que les variables `cpt` et `b` dans la fenêtre Watch. Les ports d'entrée/sorties seront observés en binaire tandis que la variable `cpt` sera visualisée en décimal.
- Exécutez le programme en pas à pas (Reset puis Step Over) et observer l'évolution des variable dans la fenêtre. Vérifier la cohérence avec le programme C exécuté.
- Modifiez la valeur de `cpt` pour mettre une valeur égale à 254 et reprendre l'exécution du programme. Que constatez vous après deux appuis sur le bouton poussoir ?
- Expliquer le rôle de la variable `b`.
- Commenter chaque ligne du programme.

3/ Rendre la carte cible autonome

La macro `__CONFIG( )` fixe la valeur du mot de configuration du µC.

- Expliquez le rôle de chaque bit.
- Avec le mode "Background Debug" validé, le pic attend un ordre de lancement venant de l'ICD2 pour exécuter son programme. La carte n'est donc pas autonome. Pour le vérifier, débranchez l'alimentation de la carte et la liaison vers l'ICD2 et constatez que le programme de démarre pas lors de la remise sous tension.
- Modifiez le programme source pour changer la valeur du mot de configuration afin de désactiver le mode "Background Debug". (Vous pouvez vous aider de la fenêtre "Configuration Bits"). Sélectionner l'ICD2 en tant que programmateur (`Programmer->Select Programmer`) et chargez le programme modifié, débranchez la liaison ICD et vérifiez qu'il démarre dès la mise sous tension.
- Revenir à la configuration initiale pour la suite (sinon le débug est impossible).

TP n°1:

4/ Un microcontrôleur pour allumer une LED !

On souhaite allumer la LED câblée sur la sortie RB3 lorsque l'on appuie sur le bouton câblé sur RA4 (S2).

- Cherchez sur le schéma électrique de la carte "PICDEM 2 PLUS" quel est l'état de la broche RA4 en fonction de l'état appuyé/relâché de S2. Cherchez également quel doit être l'état de la sortie RB3 pour allumer la LED correspondante (le cavalier J6 étant en place).

- Comment est configurée la broche RB3 au Reset ? Comment doit elle être configurée pour notre programme ? Quelles sont les instructions permettant cette configuration ?

- Ecrire le programme d'allumage de la LED et essayez-le.

5/ Affichage du nombre d'appuis sur les bits RB0 à RB4.

On souhaite maintenant afficher le nombre d'appuis sur S2 en binaire sur les bits de poids faible du PORTB en incrémentant un compteur à chaque appui. Il suffira simplement de mettre la valeur du compteur dans le PORTB. Pour éviter un défilement trop rapide, il faut attendre que l'on relâche S2 avant de prendre en compte un nouvel appui. Pour cela, utilisez un mécanisme similaire à celui utilisé dans tp1.c

- Quelle valeur doit on mettre dans TRISB ?

- Ecrire et tester le programme ? Que se passe-t-il au quinzième appui ?

-Modifiez votre programme pour que le compteur repasse à 0 après 10 appuis.

- Noter la taille mémoire utilisée par votre programme (ROM et RAM).

Pour améliorer l'affichage, on va utiliser les fonctions de la librairie biosdem2 pour écrire sur l'afficheur lcd. Pour cela, il faut copier les fichier biosdem221.h et biosdem221.lib dans votre répertoire de travail et ajouter le fichier biosdem221.lib à votre projet. Ajouter l'inclusion de biosdem221.h dans votre fichier source par `#include "biosdem221.h"`.
(les " " indiquent que le fichier est à rechercher dans votre répertoire courant)

- En vous aidant de la documentation de la librairie biodem2, afficher le nombre d'appui en décimal sur la première ligne de l'écran lcd.

- Noter la taille mémoire du programme, conclure.

- Afficher en plus le nombre d'appui en hexadécimal au milieu de la deuxième ligne.

- Modifier votre compteur à l'aide de la watch window pour mettre la valeur 254. Vérifier que votre affichage marche bien après quelques appuis. Sinon corriger votre programme.

6/ Deux boutons et une LED

- Ecrire un programme qui allume la LED RB2 lorsqu'on appui à la fois sur les boutons S2 et S3.

7/ Un compteur/décompteur avec affichage sur l'écran lcd

-Ecrire un programme conforme aux spécifications suivantes :

Le bouton S2 (RA4) incrémente un compteur.
Le bouton S3 (RB0) le décrémente.
L'affichage de la valeur du compteur est réalisé en décimal sur l'écran lcd
Lorsque le compteur arrive à -12, S3 devient inactif.
Lorsque le compteur arrive à +12, S2 devient inactif.

8/ Un thermomètre Celsius/Fahrenheit

En utilisant les fonctions de la librairie biosdem, réalisez un thermomètre qui affiche la température en °C sur la deuxième ligne de l'écran lcd. Un appui sur le bouton RA4 modifiera l'affichage pour le faire passer en Fahrenheit. Un nouvel appui provoquera le retour en °C.

La loi de conversion est :

$$t_{(°C)} = \frac{5}{9} \cdot (t_{(°F)} - 32)$$

```
/* tp1.c : à commenter */
#include <pic.h>
__CONFIG(0X3739);

void main(void)
     {
          unsigned char cpt = 0 ;
          unsigned char b=0 ;
          TRISB3 = 0 ;
          TRISB2 = 0 ;
          RB3 = 0 ;
          RB2 = 0 ;

          for ( ; ; )
               {
                 if ( RB0 == 0 )
                    {
                     RB3 = 1 ;
                     if ( b==0 )
                       {
                        cpt = cpt + 1 ;
                        b = 1 ;
                       }
                    }
                 else
                    {
                     RB3 = 0 ;
                     b = 0 ;
                    }
                 if ( cpt >= 12 )
                    {
                     RB2=1;
                    }
               }

     }
```