

# **Remise à niveau Informatique Industrielle**

## **Travaux Pratiques**

Formation au développement d'applications en Assembleur sur un microcontrôleur Microhip PIC 18F4520 en utilisant l'environnement MPLAB IDE.

Contact :

[julien.marot@fresnel.fr](mailto:julien.marot@fresnel.fr)

## 1. Avant de commencer

Ces séances de travaux pratiques (TP) sont destinées à illustrer les notions qui vous ont été présentées en cours. Pour cela, vous aurez à concevoir et à tester un certain nombre de programmes soit en langage assembleur soit en langage C.

**Pour chaque tâche que vous aurez à réaliser**, nous vous invitons à enregistrer votre programme sous un nom particulier pour garder une sauvegarde de travail. Pour faciliter le débogage et assurer la maintenance et la portabilité de vos codes, il est primordial que vous commenciez par rédiger un algorithme, que vous commentiez clairement votre programme et que vous pensiez à le rendre le plus clair possible (*ex* : en choisissant des noms appropriés pour les labels).

**Concernant l'évaluation de votre travail**, nous vous demanderons de nous présenter le programme correspondant à chaque tâche que vous aurez à programmer sur le simulateur ou sur le kit de démonstration (*cf.* plus bas). Nous corrigerons également vos codes en portant une attention particulière à la lisibilité (pensez donc aux commentaires !). Enfin, vous devrez également nous rendre les algorithmes qui correspondent aux codes.

## 2. Le Matériel

### 2.1. Le Kit PICDEM 2 Plus et PIC 18F4520

Le kit PICDEM 2 Plus est un kit simple qui permet de mettre en évidence les capacités de différents micro-contrôleurs du fabricant Microchip à travers divers périphériques et accessoires reproduits sur le schéma de la figure 1.

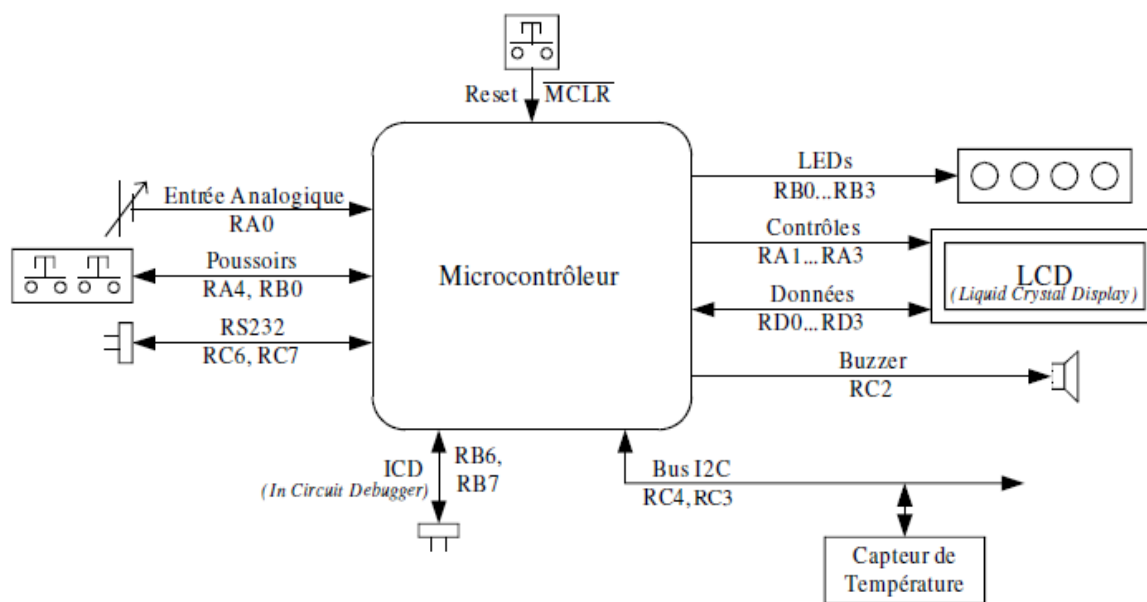


Figure 1 : Le kit PICDEM 2 PLUS, Périphériques et Accessoires

Le micro-contrôleur utilisé est le PIC18F4520 de Microchip. C'est un micro-contrôleur 8 bits disposant d'une mémoire de programme de 32 Ko, de 1536 octets de RAM (Random Access Memory), d'une EEPROM (Electrically Erasable Programmable Read Only Memory) de 256 octets et de 36 entrées/sorties.

Une part importante de la difficulté pour programmer un micro-contrôleur réside dans l'acquisition d'informations. Pour acquérir cette compétence, nous vous invitons à vous référer fréquemment à la documentation technique du micro-contrôleur, ainsi qu'à la documentation technique de la carte d'évaluation, *cf.* les documentations électroniques.

## 2.2. MPLAB Integrated Development Environment

MPLAB IDE est un environnement gratuit de programmation pour les micro-contrôleurs de la famille Microchip. L'environnement intègre un éditeur d'assembleur, un débogueur complet ainsi que tous les outils permettant de gérer les programmeurs de composants. La possibilité d'intégrer un compilateur C dans l'environnement de développement MPLAB s'inscrit directement dans la politique du service qui est d'utiliser des langages de programmation évolués dans les développements en électronique.

## 2.3. Le programmeur ICD 3

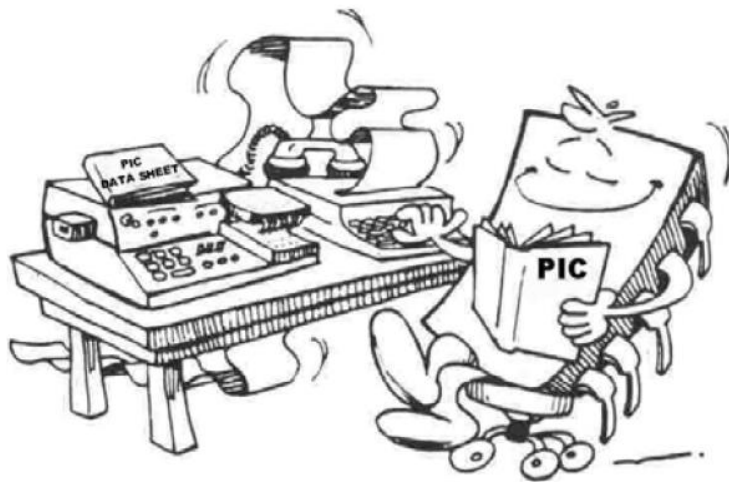
Il permet de transférer le programme directement de MPLAB dans le micro-contrôleur via le port USB d'un PC.

## 3. Gestion de vos fichiers

Avec votre compte vous avez droit à un espace de stockage sur le serveur. Cette espace de stockage apparait dans « *Poste de Travail* » sous le nom de « *voir le nom* ». C'est ici que vous devez stocker tous vos documents afin qu'ils vous soient disponibles sur toutes les machines de TP.

***Les documentations techniques nécessaires au TP sont disponibles à l'adresse:***

<https://sites.google.com/site/infoindusamu/>



## 4. Initiation à MPLAB IDE

Le but de ce TP est de se familiariser avec la suite logicielle « MPLAB Integrated Development Environment » (v8.92) de Microchip afin de pouvoir utiliser les fonctions de bases de cet outil développement à savoir : créer un projet, éditer et compiler un programme en assembleur, simuler son fonctionnement et enfin programmer le microcontrôleur ( $\mu$ C).

### 4.1. Édition et Compilation du programme

Cette partie permet d'apprendre à créer un projet MPLAB, saisir un programme en assembleur et le compiler.

#### 4.1.1. Démarrage



- Avant de commencer, créer le répertoire *tp1* puis dans ce répertoire créer à nouveau un répertoire *interrupteur*.
- Copier dans ce dernier répertoire les fichiers *P18F4520.inc* et *CONFIG.inc*
- Lancer la suite logicielle MPLAB IDE à partir de l'icône qui se trouve sur votre bureau.

#### 4.1.2. Création d'un nouveau projet avec « Project Wizard »

- Cliquer sur *Project >> Project Wizard ...*
- Cliquer sur *Suivant >* puis sélectionner PIC18F4520 dans le menu déroulant *Device*.
- Les fichiers *asmwin*, *asmlib*, *asmlink* doivent ensuite être trouvés (cliquer sur *Browse* à chaque fois) dans le dossier *MPASM SUITE* en remontant dans l'arborescence des dossiers.
- Cliquer sur *Suivant >* puis sélectionner « *Microchip MPASM Toolsuite* » et « *MPASM Assembler* » afin de pouvoir programmer en Assembleur.
- Cliquer sur *Suivant >* puis dans le champ « *Project Name* » saisir le nom du projet (ex : *interrupteur*) et dans le champ « *Project Directory* » aller chercher à l'aide du bouton « *Browse* », le dossier créé au début : *tp1/interrupteur*.
- Cliquer sur *Suivant >* et ajouter les fichiers *P18F4520.inc* et *CONFIG.inc* au projet à l'aide du bouton « *Add* » >>
- Cliquer sur *Suivant >* puis sur *Terminer*.

#### 4.1.3. Edition du programme

Créer un nouveau fichier dans le projet, *File >> Add New Files to Project...*

L'enregistrer dans le répertoire projet avec l'extension *asm* (ex : *prog.asm*)

MPLAB Editor se lance. Saisir un programme simple qui vous permet d'allumer une LED :

```
LIST P=18F4520
#include <P18F4520.inc>
#include <CONFIG.inc>

;----- Programme
org h'0000' ;Adresse de debut du programme sur Reset
goto init

init      clrf    PORTB           ; RAZ PORTB
          movlw  b'00000001'
          movwf  TRISB           ; RB0 en entrée, RB1 à RB7 en sortie
          movlw  h'0F'
          movwf  ADCON1          ; RB0 à RB3 en mode I/O numérique

main
  bsf PORTB,1
  goto main

END
```

## Travaux Pratiques de Microcontrôleur – PIC 18F4520

Penser à ajouter des commentaires au programme pour montrer que vous avez bien compris les différentes instructions. Une ligne de commentaire commence par « ; ».

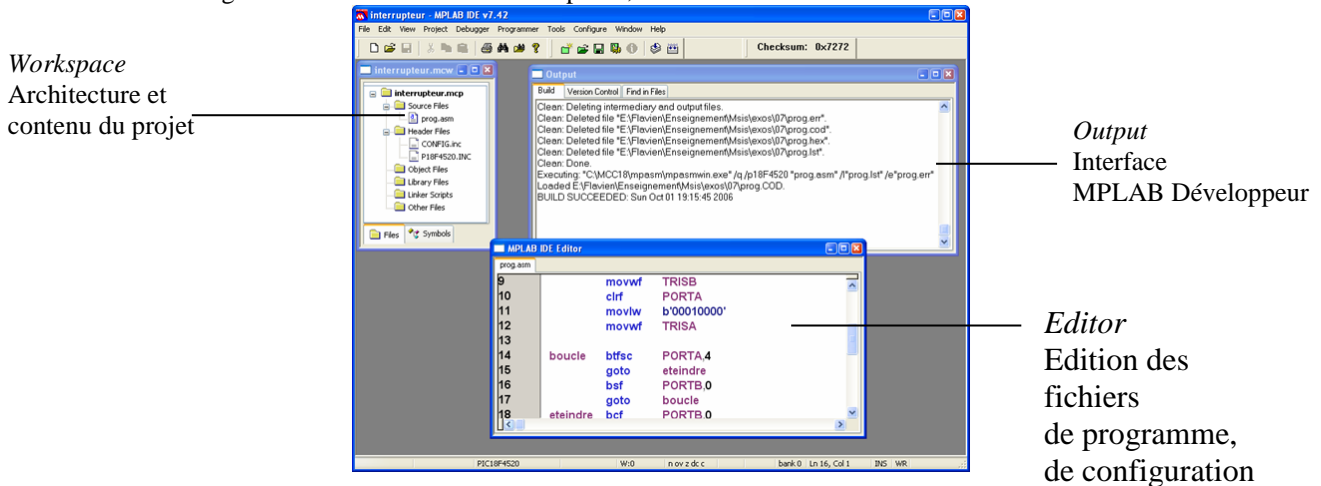




Figure 1: MPLAB IDE, Mode Edition

### 4.1.4. Compilation

- Cliquer sur *Project >> Build All ...* ou sur l'icône  de la barre de menu ou encore *Ctrl+F10* pour compiler le projet complet. Erreurs et warnings sont signalés dans la fenêtre *Output* ainsi que le résultat de la compilation *BUILD SUCCEEDED* ou *BUILD FAILED*.  
*En double-cliquant sur une erreur ou un warning dans la fenêtre Output, vous serez amené directement à la ligne concernée dans votre programme.*

## 4.2. Programmation du microcontrôleur

Cette phase consiste à envoyer le code hexadécimal du programme dans le  $\mu$ C. On utilise le programmeur MPLAB ICD3 pour que le programme soit injecté dans le PIC qui se trouve sur la plaquette de test.

- Dans MPLAB, sous le menu *Programmer > Select Programmer* choisir « MPLAB ICD3 ». Un nouvel onglet est alors ajouté à la fenêtre « *Output* ».
- Envoyer votre programme, menu *Programmer > Program* ou en cliquant sur , le transfert est terminé lorsque la LED jaune *Activate* du programmeur est éteinte et que le message « *Programming/Verification completed successfully!* » apparaît dans la fenêtre *Output*.

---

Que pouvez-vous dire de votre programme ? Est-ce que la carte de test réagit comme prévu ?

## 5. Mise en œuvre d'une PWM

Un transducteur piézo-électrique est un composant électronique constitué d'une lamelle de matériau piézo-électrique. Cette lamelle est déformée si une tension alternative est appliquée aux bornes du transducteur piézo-électrique.

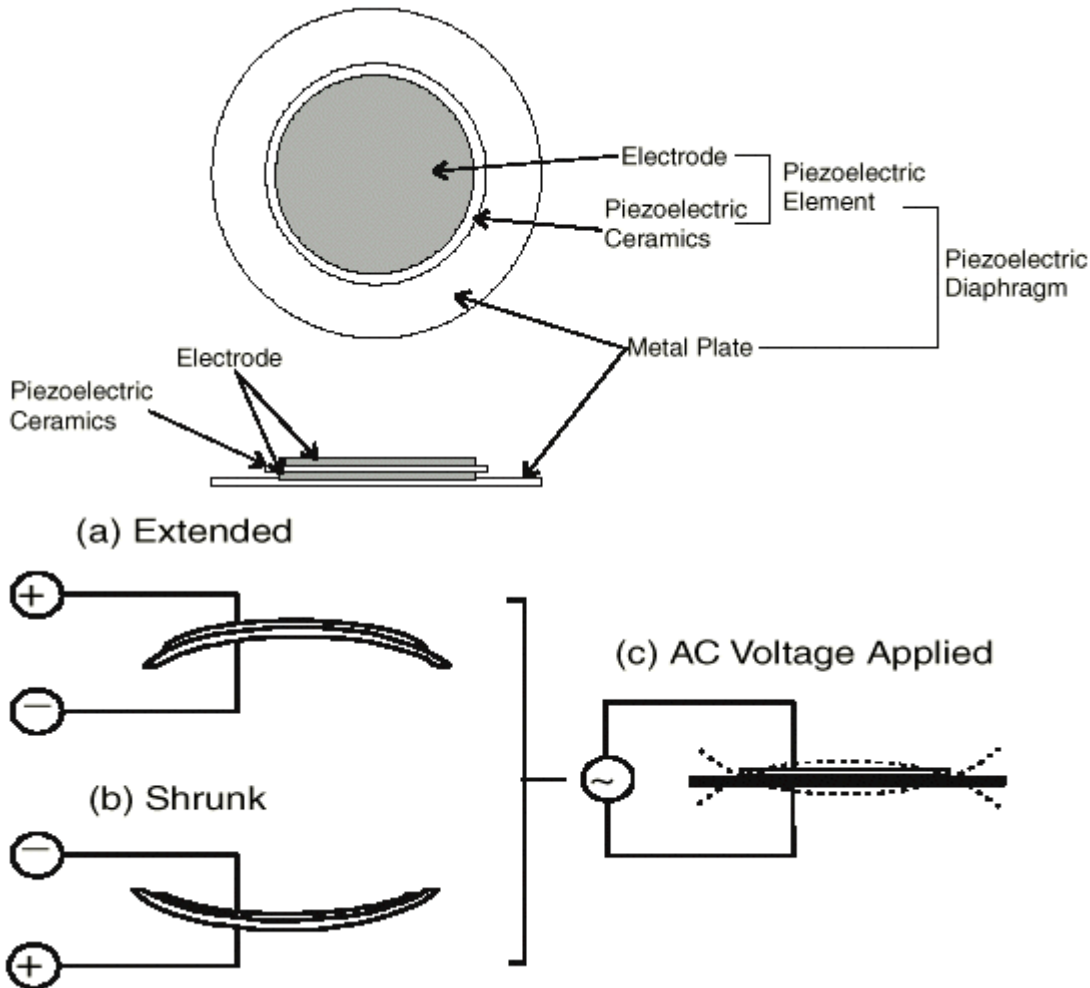


Figure: Le transducteur piézo-électrique

Le mouvement de la lamelle dans l'air génère un son. La fréquence du son émis est directement lié à la nature et à la fréquence du signal appliqué aux bornes du transducteur piézo-électrique.

Sur la carte d'évaluation PICDEM 2 Plus, le transducteur piézo-électrique peut être commandé grâce à la broche RC2 du micro-contrôleur. Le montage électronique utilisé pour commander le transducteur piézo-électrique permet de l'alimenter avec une tension bipolaire (+5V, -5V).

Pour générer un son, on doit donc générer un signal rectangulaire (grâce au module PWM) que l'on appliquera au transducteur piézo-électrique si l'utilisateur appuie sur le bouton poussoir S3. Il faudra initialiser tous les modules nécessaires à la génération d'un signal rectangulaire (module CCP1, TIMER2) (voir p.159 du « data-sheet ») et initialiser le PORTC. À partir du programme précédent, apporter les modifications nécessaires :

## Travaux Pratiques de Microcontrôleur – PIC 18F4520

*lors de l'initialisation :*

- Configurer le PORTC en entrée excepté la broche RC2 en sortie,
- Configurer le module PWM, placer la valeur 103 (en décimal) dans le registre PR2 (fixe la fréquence du signal rectangulaire), régler le duty time (rapport cyclique) à 0.5, on placera la valeur 0x00 dans le registre CCP1CON,
- Activer le TIMER2, placer le « *postscaler* » à 1:1, placer le « *prescaler* » à 16,

*dans le programme principal :* on implantera une boucle d'attente simple (avec les instructions goto, et nop)

**Attention :** *Le Duty Time est le temps pendant lequel le signal est à l'état haut. Il faut adapter ce temps en fonction de la fréquence du signal généré pour garder un rapport cyclique égale à 0,5. Pour cela lisez la documentation technique et trouver la relation qu'il doit exister entre les valeurs des registres PR2 et CCP1L:CCP1CON<5:4> pour que le rapport cyclique reste égale à 0.5.*

Donner la relation qu'il doit exister entre les valeurs des registres PR2 et CCP1L:CCP1CON<5:4> pour que le rapport cyclique reste égale à 0.5 et proposer un partie de programme (le plus simple possible) en langage Assembleur qui permet d'adapter le rapport cyclique (contrôlé par les registres CCP1L:CCP1CON<5:4>) en fonction de la fréquence (contrôlée par le registre PR2) du signal généré.

Réaliser l'algorithme du programme demandé,

Écrire le programme sur ordinateur avec MPLAB, simuler son fonctionnement et le valider sur la carte d'évaluation.

*Dans un premier temps, modifier le programme précédent pour qu'il génère un signal rectangulaire (en utilisant le module PWM) sur la broche RC2. Puis vous prendrez en compte la gestion du bouton RB0. En séparant les deux problèmes, la difficulté est réduite.*

En partant du programme précédent, on souhaite maintenant ajouter une fonctionnalité : à chaque appui sur le bouton poussoir S3 connecté à RB0, on souhaite activer le transducteur piézo-électrique (pendant la durée de l'appui). Le transducteur piézo-électrique peut être commandé grâce à la broche RC2 du micro-contrôleur. On mettra en œuvre pour cela une interruption externe par bouton poussoir.

- Dans le cas où RB0 est à l'état haut on activera le transducteur piézo-électrique en agissant sur le registre CCP1CON,
- Dans le cas où RB0 est à l'état bas on désactivera le transducteur piézo-électrique en agissant sur le registre CCP1CON,
- Dans le cas où la valeur sur la broche RB0 est égale à la dernière valeur acquise on ne fera rien.