



## TP n°1: Périphériques de bases des PIC18F



### Objectifs :

**Savoir utiliser les périphériques PORT E/S, ADC, USART et Timer d'un pic18F.**

Ce tp se fait une la carte PICDEM Mechatronics équipée d'un pic 18F4520 et d'un oscillateur à 4Mhz. La fréquence d'horloge (Fosc) du pic sera portée à **16MHz** par l'utilisation de la pll interne du pic.

La configuration est la suivante :

```
#pragma config OSC = HSPLL
#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config PBDEN = OFF
```

La carte à été modifiée pour pouvoir accepter un 18F4520. Ainsi les pattes RC0,RC1 et RC2 ont été recâblées sur des broches libres (RC4,RC3,RC6 respectivement) du support U2. C'est donc sur le connecteur J6 qu'il faudra se connecter lorsqu'on voudra utiliser les broches RC0, RC1 ou RC2 du 18F4520 monté sur U1.

(L'écran lcd n'est pas utilisable dans cette configuration. Les dialogues se feront par le port série.)

**Attention les erreurs de câblages et les erreurs de programmation (notamment du sens (entrée ou sortie ) peuvent endommager le pic et/ou la carte. Soyez vigilant. Dessinez un schéma avant chaque manipulation. Vérifiez et faites vérifier vos câblages. Attention à la parallaxe.**

### Documents nécessaires :

Cours pic S3 : cours micro pic S3 v > 1.6  
Documentation 18F4520 : Data Sheet DS39631A  
Documentation librairie C18 : MPLAB-C18-Libraries\_51297f  
Documentation carte PICDEM Mechatronics : DS51557B  
Exemples pic18 : <http://arlotto.univ-tln.fr/pic/pic18>

**Dans tous vos programmes, en plus de ce qui est demandé, la led D0 doit clignoter au rythme de la boucle infinie (environ 1/5000). Le clignotement de cette led prouvera que la boucle infinie n'est ni bloquée, ni fortement ralentie.**

## 1) Entrée sorties et port série.

a) On souhaite allumer une led lorsqu'on appuie sur le bouton SW2 et l'éteindre lorsqu'on appuie sur SW3.

Faites un schéma, câbler le et faites le programme correspondant.

b) Lorsque la led s'allume émettre la chaîne de caractères "ON" sur le port série à 19200b/s.  
Émettre "OFF" lorsque la led s'éteint.

c) Implantez dans votre programme le stockage sous interruption d'une chaîne reçue jusqu'à rencontrer un CR. Allumer la led lors de la réception de "MARCHE\r" et l'éteindre lors de la réception de "ARRET\r".

Exemple d'affichage sur un terminal :

```
TP1 v1.0
>MARCHE
ON
>ARRET
OFF
>ZZZ
ERREUR !
>
```

## 2) Conversion analogique/numérique

a) Câblez un potentiomètre sur une entrée analogique. (mettez à jour votre schéma) Ajoutez des commandes (POT? , VPOT?) permettant de connaître la position du potentiomètre à partir du terminal et la tension en millivolt sur l'entrée analogique :

```
>POT?
POT : 75%
>VPOT?
VPOT : 3750 mV
>
```

Nb : Le type float n'est pas nécessaire ici.

C18 accepte le type float mais la fonction printf n'a pas le format %f.

b) Câblez le capteur de luminosité sur une entrée analogique (J4,LIGHT). (mettez à jour votre schéma). Ajoutez une commande permettant de connaître le niveau de luminosité relatif (0-100%).

```
>LUM
LUM 35%
>
```

c) Câblez le capteur de température sur une entrée analogique (J4,TEMP). (mettez à jour votre schéma).

Exprimez la relation entre la température du capteur TC1047A et le résultat de la conversion fournit par le module ADC.

Calculer la résolution de la mesure en °C.

e) Ajoutez une commande permettant de connaître la température en °C avec la meilleure résolution possible : `>TEMP?`

TEMP : 23.7 °C

>

### 3) Éclairage automatique

On va maintenant simuler un éclairage automatique qui s'allume dès que la luminosité descend en dessous d'un seuil. (le capteur d'éclairement est placé hors de la zone couverte par l'éclairage). On suppose que la sortie de la question 1 (câblée sur la led) commande (via une interface de puissance) l'éclairage. Le bouton SW4 permet de passer du mode manuel (question 1 ON SW1/OFF SW2) au mode automatique.

En mode automatique l'éclairage est commandé dès que l'éclairement descend en dessous d'un seuil avec un hystérésis de +/-10%.

L'appui sur SW1 ou SW2 en mode automatique provoque le retour au mode manuel.

a) la valeur du seuil est stockée dans une variable qui est mise à jour par la position d'un potentiomètre. Réalisez le programme d'éclairage automatique.

b) lorsque qu'on reçoit du port série la commande LUM xxx ( $0 \leq xxx \leq 100$ ) la valeur du seuil est fixée à xxx indépendamment de la position du potentiomètre. Le retour à la prise en compte du seuil par potentiomètre se fait par un reset ou la réception de la commande LUM POT.

Exemple : Mise sous tension Seuil = position pot.

`>LUM 63` // Seuil = 63%

`>LUM 234` // non valide Seuil inchangé

Valeur hors limite ( $0 < \text{seuil} < 100$ )

ERREUR !

`>LUM POT` // Seuil = position pot.

c) Modification de la valeur de l'hystérésis, la commande HYST yy permet de faire varier l'hystérésis de l'asservissement entre 5 et 25 %.

d) En plaçant le traitement de l'éclairage automatique dans la boucle principale, on exécute celui-ci à la vitesse maximum de la boucle. Or cela n'est pas nécessaire, une mise à jour toutes les minutes est largement suffisante. A l'aide d'un timer, exécutez le traitement avec une période de une minute. Dans ce cas la valeur de luminosité à prendre à compte est la moyenne de six dernières mesures effectuées toutes les dix secondes.

e) La configuration de l'automatisme (mode,seuil,hystérésis,...) étant sauvée en RAM, celle-ci est réinitialisée après une coupure de courant. Sauver la configuration en mémoire EEPROM pour s'affranchir de cet inconvénient. Prévoir dans ce cas, la possibilité de revenir à une configuration par défaut (dite « configuration usine »).

f) Pour diminuer la consommation du dispositif on peut mettre le pic en veille (par une instruction sleep) entre les périodes de mesure. Pour cela il faut arrêter tout les périphériques (fonctions close()) sauf un timer, puis programmer un réveil par le timer et mettre le PIC en veille. Attention l'utilisation du débbuger n'est pas possible avec le mode veille.