

# Formation Automatique et Informatique Industrielle

Master 1 S2

Matière : Systèmes Embarqués et Systèmes  
Temps Réel SE-STR

Par : ATOUI Hamza

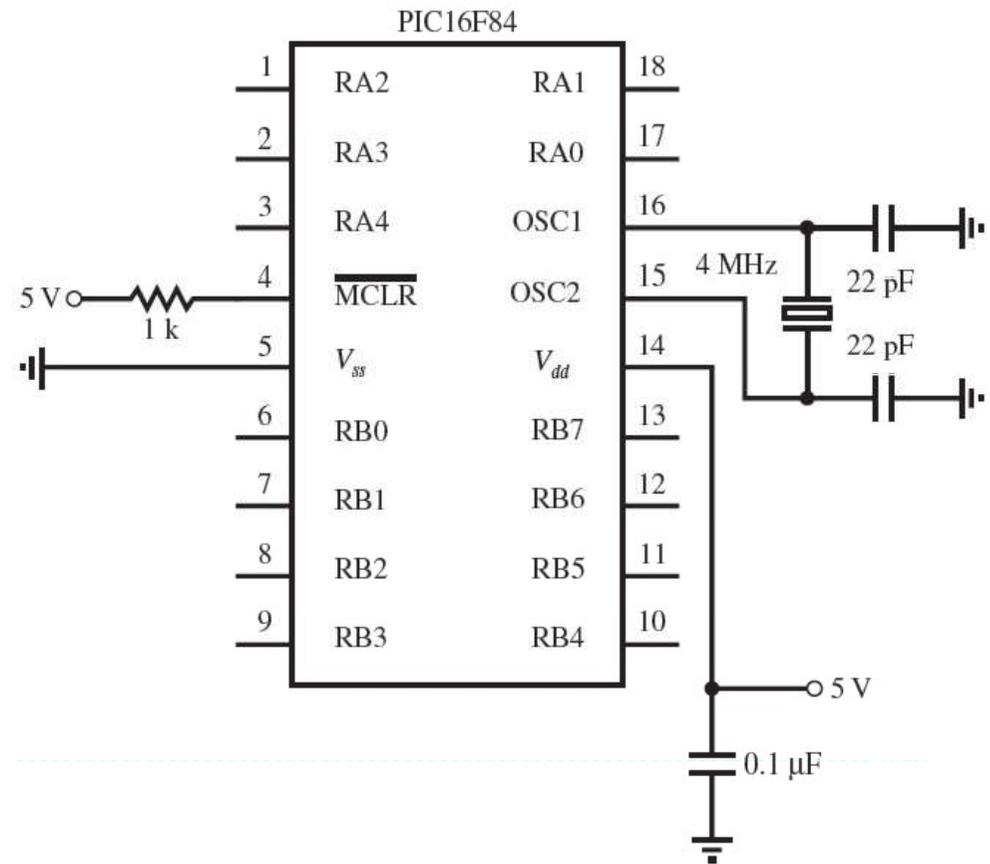
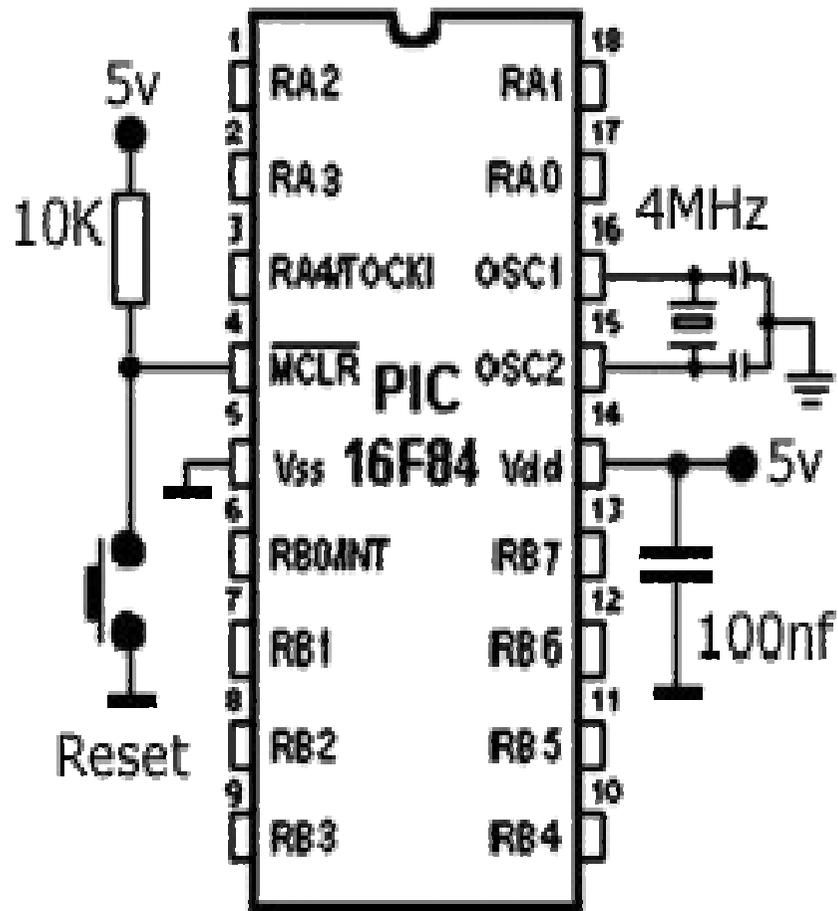
# Plan de TD-TP

- La mise au point.
- LAB1.
- LAB2.
- LAB3.
- LAB4.
- LAB5.
- LAB6.
- LAB7.
- LAB8.

# Mise au point

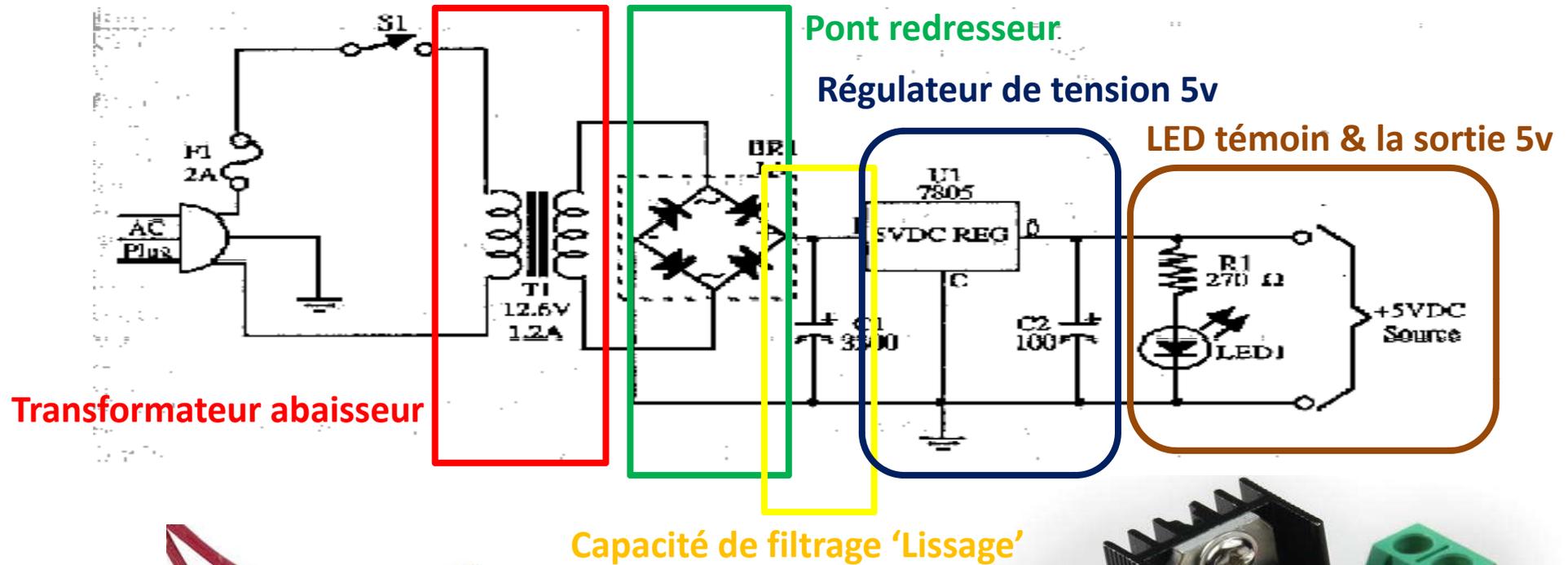
- Quelle est la configuration matérielle de base pour travailler avec le PIC16F84A ???
- Tout simplement, il faut :
  - Une alimentation 5 volts DC.
  - Réalisation de circuit RESET (optionnel).
  - Réalisation de circuit d'horloge (quartz + 2 capacités de 22 pF).

# Mise au point

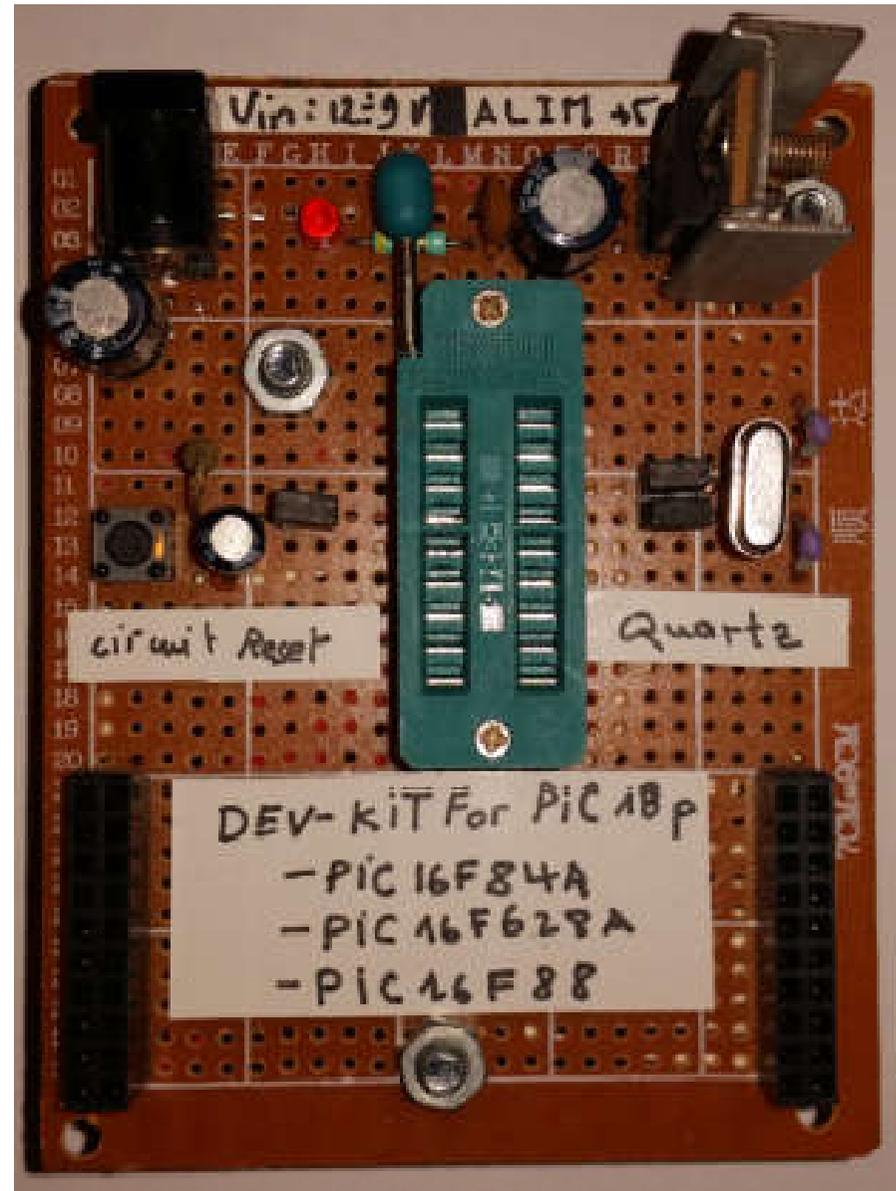
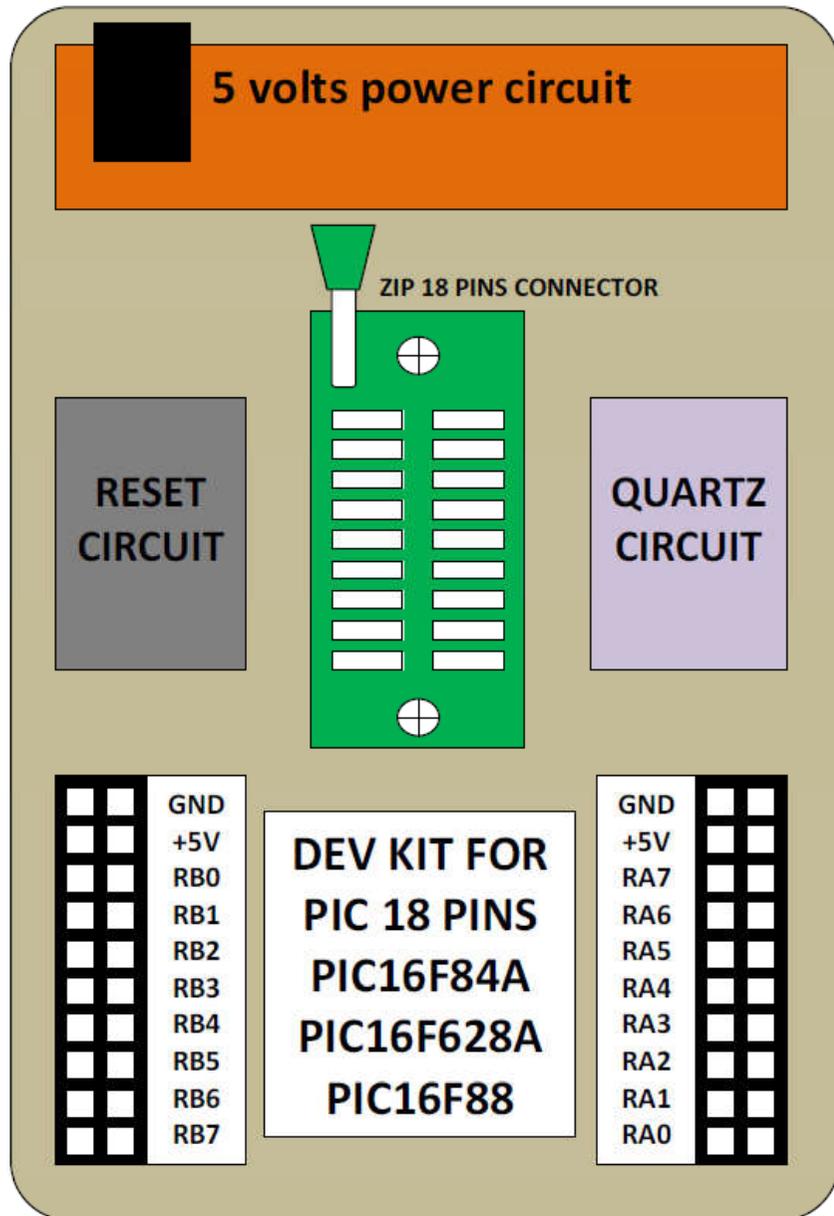


# Mise au point

- Comment réaliser une alimentation 5 volts DC à partir du notre réseau électrique 220 volts AC.



# DEV KIT FOR PIC 18-PINS

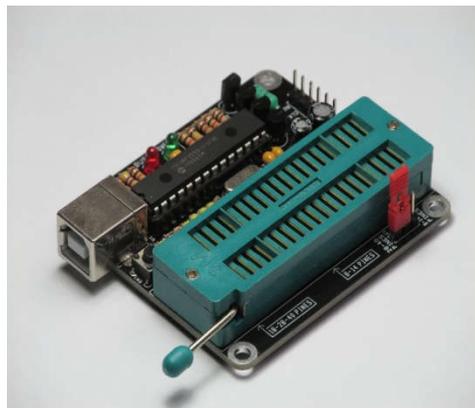
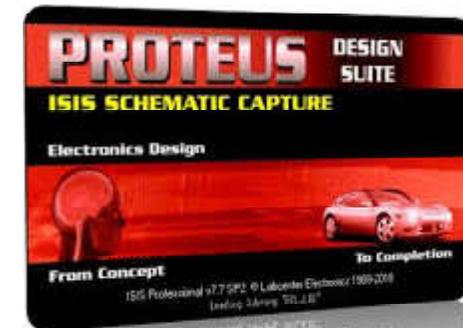


# DEV TOOLS

- Les outils utilisés pour réaliser un projet sont :
- Un environnement de développement
  - MPLAB IDE.
- Un Compilateur
  - Assembleur MPASM
  - HI-TECH C Compiler, MikroC, MikroPascal, PICBasic,....
- Un environnement de simulation (optionnel)
  - PROTEUS (ISIS), MULTISIM,....
- Programmeur SOFT/HARD
  - PICKIT2, PICKIT3, ICPROG, WINPIC800....



everywhere™  
**mikroC**  
PRO for PIC



# LAB1

- Ce LAB présente une démonstration sur l'utilisation des I/O de PIC16F84A.
- Réaliser le schéma et le programme en assembleur d'un bouton relié à la ligne RA0 du PORTA et une LED relié à la ligne RB1 du PORTB, dont le traitement est d'indiquer l'état de bouton sur la LED.

# LAB2

- Dans ce LAB, on va réaliser un émulateur de la bascule S-R par le PIC16F84A.
  - L'entrée S relié à la ligne RA2.
  - L'entrée R relié à la ligne RA3.
  - La sortie Q relié à la ligne RB5.
- On demande de réaliser le schéma et le programme qui assure le bon fonctionnement de l'émulateur de cette bascule.

# LAB3

- Le but du LAB est de réaliser la séquence d'un automatisme [lecture des entrées, Traitement, et mise à jour des sorties] en LADDER (langage contact) par un  $\mu\text{C}$ .
- Ecrire un programme qui réalise le suivant:
  - $\text{RA0} \leftarrow \text{RB0 or RB1}$ .
  - $\text{RA1} \leftarrow \text{RB2 and RB3}$ .
  - $\text{RA2} \leftarrow \text{RB4 xor RB5}$ .
  - $\text{RA3} \leftarrow \text{not(RB6)}$ .
  - $\text{RA4} \leftarrow \text{RB7}$ .

# LAB4

- Le but du LAB est comment programmer une temporisation à base du SOFT et comment faire aussi avec du HARD TIMER0.
- Clignoter une LED relié à la ligne RB0 par une fréquence de 2.5 Hz.
  - La temporisation par SOFT.
  - La temporisation par HARD TIMER0.
- NB: le quartz du PIC est de 4MHz.

# LAB5

- Le comptage est une opération parmi les opérations de base de quelques appareils de mesure et même sur l'axe de l'automatisation industrielle.
- On désire de faire le comptage des impulsions arrivent de la ligne RA4 (impulsion sur franc montant ou descendant). si le nombre d'impulsions arrive à 5 alors on va inverser l'état de la LED relié à la ligne RB7.
  - Faire le comptage par SOFT.
  - Faire le comptage par HARD TIMER0.

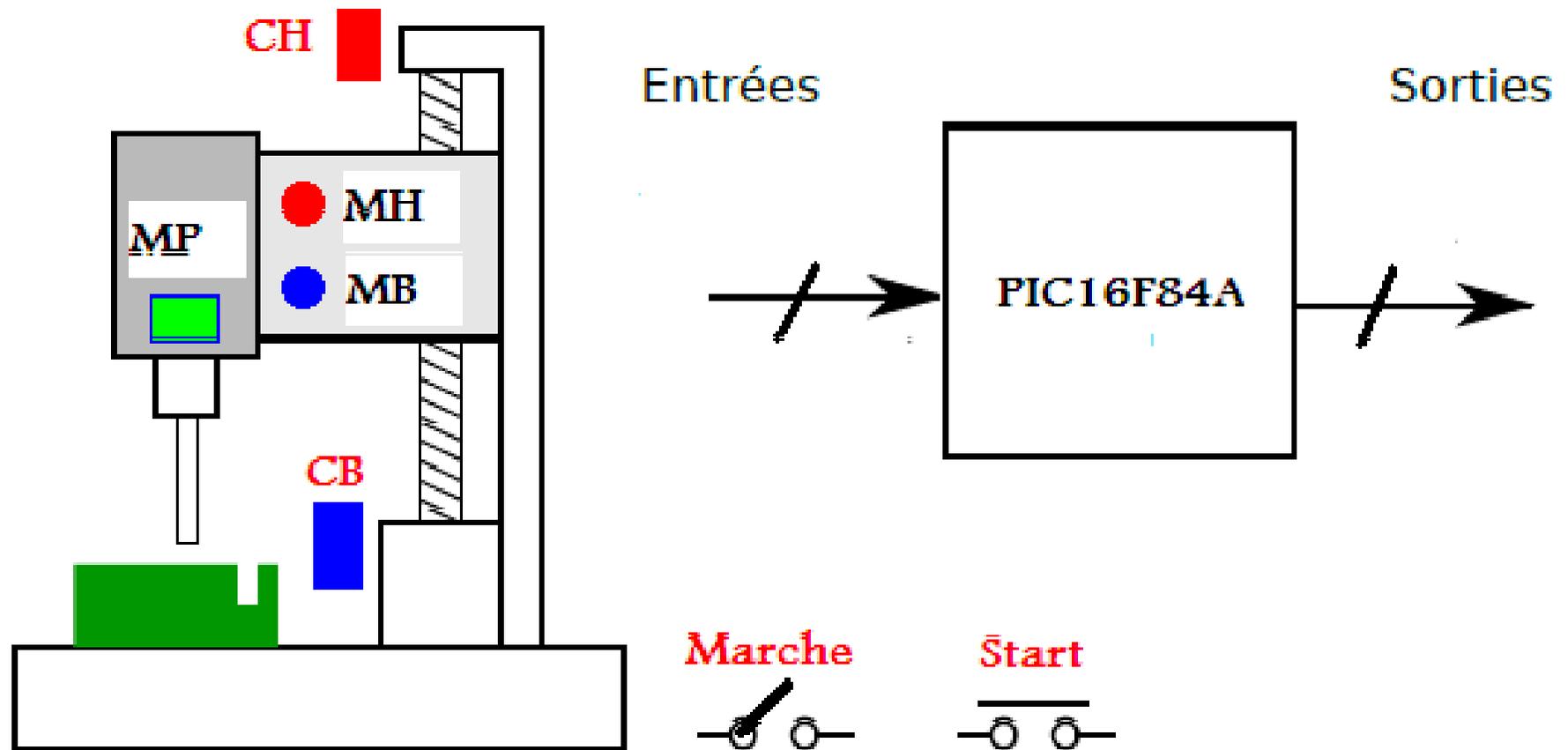
# LAB6

- Le but du LAB est comment gérer les interruptions générées par les périphériques de PIC16F84A.
- Les interruptions qu'on va gérer sont :
  - L'interruption externe de la ligne RB0/INT.
  - Le débordement de TIMER0.
- Au moment d'appuyer sur le bouton relié à la ligne RB0 (présence d'un niveau bas), ce dernier déclenche une interruption avec son traitement est d'inverser une LED relié à ligne RA0.
- Le TIMER0 fait le comptage des impulsions négatives par la ligne RA4. Si le nombre d'impulsions comptées est égale à 10, ce dernier déclenche une interruption avec son traitement est d'inverser une LED relié à ligne RB4.
- La priorité est donnée à la source RB0/INT.

# LAB7

- Le but du LAB est de gérer un afficheur BCD7SEG par le PIC16F84A en utilisant le Look Up Table (LUT).
- Réaliser le montage et le programme d'un compteur Modulo 10 libre.
- Affichage se fait sur un afficheur BCD7SEG anode commune puis sur un afficheur BCD7SEG cathode commune.
- La fréquence du compteur est 2Hz.

# LAB8



La perceuse à automatiser

# LAB8

- On cherche à automatiser le mouvement descente/montée d'une perceuse .
- Un moteur tourne dans les deux sens (**MH** : Moteur vers le Haut, **MB** : Moteur vers le Bas).
- Un bloc perceuse composé d'un moteur avec une mèche de perçage; le moteur tourne si la présence d'un niveau haut sur **MP** (Moteur Perceuse).
- Deux capteurs pour déterminer la position du bloc perceuse (**CH** : Capteur position Haute, **CB** : Capteur position Basse).
- Un Interrupteur « **Marche** » pour démarrer le système.
- Un bouton poussoir « **Start** » pour démarrer le cycle de perçage.

# LAB8

- **Cahier des charges:**
- Le  $\mu$ C attend la fermeture de l'interrupteur **Marche** pour lancer le processus d'initialisation de la perceuse (si le capteur **CH** ni pas actif le moteur des/mont marche vers le haut jusqu'au positionnement de **CH** ; le moteur du bloc perceuse **MP** est à l'arrêt).
- Après le processus de l'initialisation, le  $\mu$ C attend à chaque fois la présence d'une impulsion sur le poussoir **Start** pour démarrer le cycle de perçage (**cycle de perçage** : le bloc perceuse descend vers le bas jusqu'à la fin de course **CB** puis remonte vers le haut jusqu'à la fin de course **CH**).
- faire tourner le moteur de perçage **MP** durant le cycle de perçage.
- L'arrêt d'urgence est assuré par l'ouverture de l'interrupteur **Marche**. Dans cette situation tous les moteurs sont à arrêt.

# LAB8

- **Travail demandé:**
- Définir les entrées/sorties du système.
- Designer la carte à PIC16F84A avec les I/O de la perceuse.
- Ecrire l'organigramme et le programme en MPASM sous MPLAB.
- Simuler le système par le logiciel PROTEUS (ISIS).