

## TP N°3

**Objectif :** l'étudiant sera capable de

**1\* se familiariser avec les microcontrôleurs PICs et sa programmation avec le logiciel PROTON BASIC et savoir faire la simulation avec ISIS PROTEUS et la possibilité de faire entrer et sortir les données.**

**2\* savoir faire cette simulation en pratique avec un ARDUINO.**

### **1-Faire entrer et sortir des données:**

L'une des caractéristiques les plus importantes du microcontrôleur est le nombre des broches d'entrées et de sorties utilisées pour la connexion avec des périphériques externes (des interrupteurs ou des boutons poussoirs, des Leds, des Lcds, des interfaces, etc... ).

Dans ce TP , on va utiliser un microcontrôleur PIC de Microship (PIC 16F887) comme "hardware"  
Dans l'aspect programmation, on va utiliser un langage évolué "software" qui est Proton Basic" On rappelle que la "DataSheet" d'un microcontrôleur est très indispensable pour faire la programmation et de comprendre les dispositifs interne de ce dernier.

L'objectif de ce TP est de prendre en main le logiciel ISIS PROTEUS en utilisant un PIC, les Leds et les boutons poussoirs et d'écrire un programme évolué nommé Basic sous le logiciel Proton Basic puis faire la compilation de ce programme en langage machine puis l'implanter dans le microcontrôleur PIC pour commander notre système.

### **2-Explication du travail**

1-Faire entrer les données: deux boutons poussoirs sont utilisées et branchées au Pic à travers les Ports B comme suit:

Bouton poussoirs 1 (BP1) au port RB0

Bouton poussoirs 2 (BP2) au port RB1

1-Faire sortir les données: On utilise 8 Leds comme sortie qui sont branchés au port D comme suit:

D1	D2	D3	D4	D5	D6	D7	D8
RD0	RD1	RD2	RD3	RD4	RD5	RD6	RD7

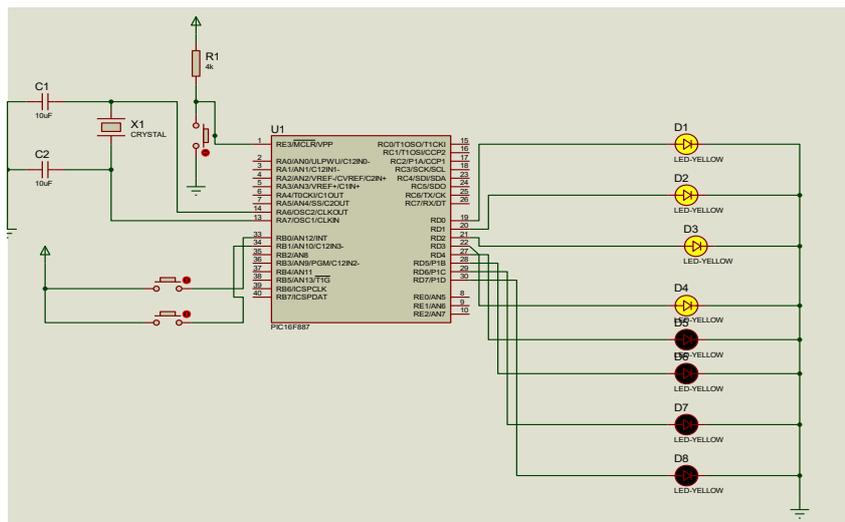
Le programme doit répondre au cahier des charges comme suit:

**Pour le bouton poussoir BP1:**

- Tâche 1: lire l'état du bouton poussoir BP1 broche RB0
- Tâche 1: si le bouton est appuyé, on allume les Leds: D1,D2,D3,D4,D5,D6,D7, D8 pour 3s puis on les éteint.

**Pour le bouton poussoir BP2:**

- Tâche 1: lire l'état du bouton poussoir BP2 broche RB1
- Tâche 1: si le bouton est appuyé,
- Tâche 1: lire l'état du bouton poussoir BP1 broche RB0.
- Tâche 1: si le bouton est appuyé, on allume les Leds:
  - \*D1,D2,D3,D4 pour 0,5 s puis on les éteint pour 0,7 s,
  - \*D5,D6,D7,D8 pour 0,5 s puis on les éteint pour 0,7 s,
  - \*D1,D2,D3,D4,D5,D6,D7,D8 0,6 s puis on les éteint pour 0,9 s à deux reprises.



Voici le programme:

```

Device 16F887           % Définir le uC utilisé,
XTAL 8                  % Définir la fréquence du oscillateur (CRYSTAL)

TRISB=$ff              % Toutes les broches des ports B sont des entrées
TRISD=$00              % Toutes les broches des ports D sont des sorties
ANSEL=$00              % Tous les ports E/S sont configurés comme numériques
ANSELH=$00            % Tous les ports E/S sont configurés comme numériques
Clear PORTD            % Remise à zéro les bascules D du port D

If PORTB.0=1 Then      % Lire l'état du broche "pin" des Ports B , (RB0) bouton poussoir BP1
PORTD=%01010101      %Alumer les leds: D1,D2,D3,D4,D5,D6,D7
DelayMS 3000          % Temporisation de 3000 ms (3 s)
Else
End If                % Fin de la boucle du bouton poussoir BP1

If PORTB.1=1 Then      % Lire l'état du broche"pin" des Ports B , (RB1) bouton poussoir BP2

```

```

PORTD=%00001111      %Allumer les leds: D5,D6,D7,D8
DelayMS 500           % Temporisation de 500 ms      (Allumer durant 0,5 s)
PORTD=%00000000      % Eteindre tous les leds
DelayMS 700           % Temporisation de 700 ms      (éteindre durant 0,7 s)
PORTD=%11110000      %Alumer les leds: D1,D2,D3,D4
DelayMS 500           % Temporisation de 500 ms      (Allumer durant 0,5 s)
PORTD=%00000000      % Eteindre tous les leds
DelayMS 700           % Temporisation de 700 ms      (éteindre durant 0,7 s)
PORTD=%11111111      % Alumer tous les leds
DelayMS 500           % Temporisation de 500 ms      (Allumer durant 0,5 s)
PORTD=%00000000      % Eteindre tous les leds
DelayMS 600           % Temporisation de 600 ms      (éteindre durant 0,6 s)
PORTD=%11111111      % Alumer tous les leds
DelayMS 900           % Temporisation de 900 ms      (éteindre durant 0,9 s)
Else
End If                % Fin de la tâche s'il n'y a pas un autre ordre

```

## On demande de réaliser pour afficher à travers l'afficheur à 7 segments:

### 1)- Simulation avec Pic

- **BP1: Afficher à travers l'afficheur à 7 segments** des lettres en majuscule de « A » jusqu'à « F » avec temporisation de 1.35 secondes entre les lettres

- **BP2: Afficher à travers l'afficheur à 7 segments** des lettres en majuscule de « 0 » jusqu'à « 5 » avec temporisation de 1.5 secondes entre les lettres

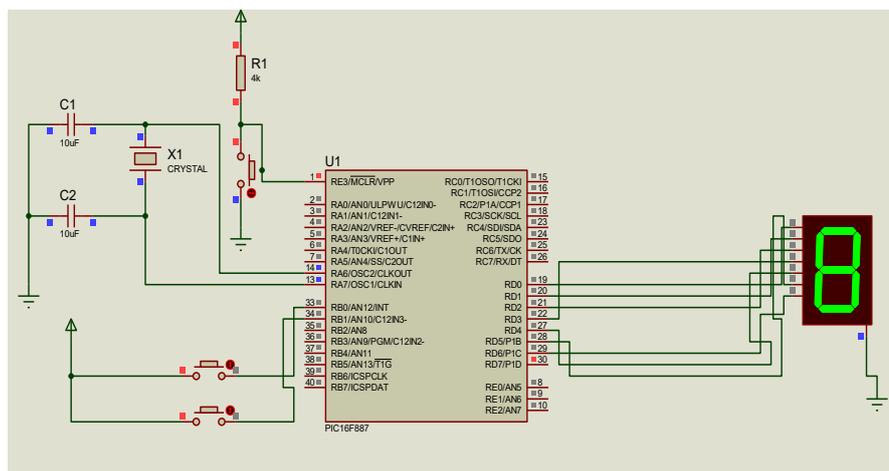
### 2)- Pratique avec arduino

- **BP1: Afficher à travers l'afficheur à 7 segments** des lettres en majuscule de « A » jusqu'à « F » avec temporisation de 1.35 secondes entre les lettres

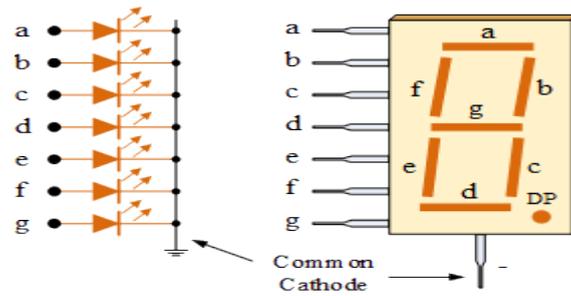
- **BP2: Afficher à travers l'afficheur à 7 segments** des lettres en majuscule de « 0 » jusqu'à « 5 » avec temporisation de 1.5 secondes entre les lettres

**On donne:**

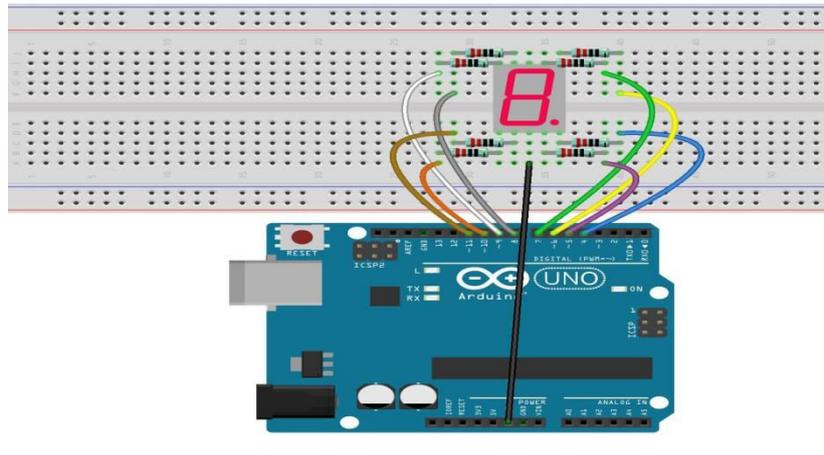
**-Schéma du montage de simulation du Pic**



## -Configuration de l'afficheur 7 segments



## -Schéma du montage de pratique d'un arduino UNO



## Remarque:

Chaque groupe doit faire la visualisation de leur travail demandé en classe avec des questions probables sur la façon de programmation et la connexion du montage à chaque étudiant du groupe.

Bonne chance