



## Manip micro N°3 (Vers :8)

### Centre d'intérêt :

# Interface homme machine (IHM) :

## *Gérer un afficheur LCD et un clavier*

Application :

Systèmes : Perceuse à commande numérique (PCN), Serrure codée (Digicode), Horloge microprogrammée, Agitateur de microplaques, Respirateur artificiel, panneau solaire, turbidimètre...

Présentation:

Centre d'intérêt : Interface Homme Machine. De nombreux systèmes nécessitent une interface homme machine (appelée IHM) afin de permettre l'introduction de consignes et de visualiser les informations utiles pour l'utilisateur.

Ces IHM disposent fréquemment de mini claviers (ou de quelques touches), associés de plus en plus à un afficheur LCD, permettant l'affichage sous forme de menus afin de simplifier l'utilisation.

**Ainsi on désire mettre en oeuvre un afficheur LCD (2 lignes de 16 caractères et un clavier 16 touches, sur une carte Millenium à l'aide du PIC 16F877.**

I) Etape N°1:

On souhaite réaliser dans un premier temps le câblage de l'afficheur LCD et sa mise en œuvre logicielle à l'aide des fonctions proposées par l'environnement MIKROELEKTRONIKA.

On utilisera alors la documentation intégrée au logiciel MikroC et celle de l'afficheur LCD fourni, pour adapter le câblage et le programme, afin de relier l'afficheur LCD au PORTC du PIC16F877.

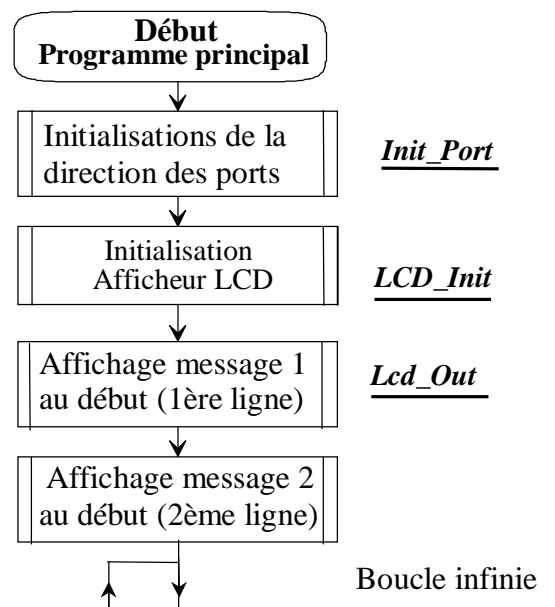
L'afficheur LCD est compatible avec le type d'afficheur LCD alphanumérique "intelligent", référence LTN 211 (référence PHILIPS), comportant 2 lignes de 16 caractères et utilisant le contrôleur Hitachi HD44780, utilisé pour les afficheurs de 1 à 4 lignes de 16 à 40 caractères.

**Organigramme du programme principal de Gestion de l'afficheur LCD**

Utilisez le document source à compléter que vous trouverez sur le serveur F13AXP1, nommé « TP3\_PIC\_IHM\_V1.doc ».

Répondez aux questions et suivez les consignes afin de réaliser le câblage correct du LCD, en rapport avec les fonctions de gestion fournies par MikroC.

Utilisez l'en-tête et la structure de programme fournies dans le compte rendu (par copié collé), et complétez le programme afin d'obtenir l'affichage des deux messages (1 et 2) correspondant à vos prénoms (1 sur chaque ligne), conformément à l'organigramme ci contre.



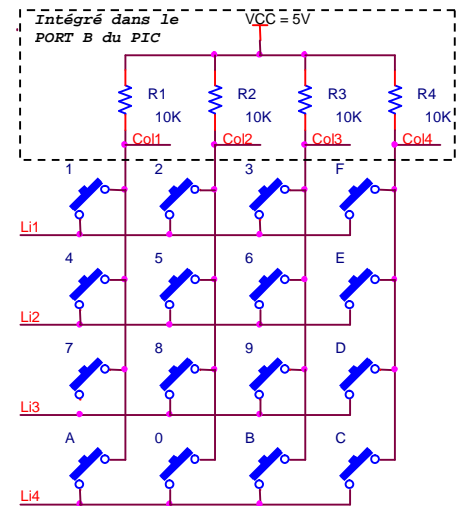
## II) Etape N°2:

On désire procéder au câblage et à la gestion du clavier sur la carte MILLENNIUM en s'aidant des consignes fournies.

Les lignes du clavier (appelées Li1 à Li4 sur le schéma du clavier ci contre) seront directement gérées logiciellement par le portB du pic, et les colonnes seront lues afin d'en déterminer la touche éventuellement enfoncée.

L'objectif est de modifier le programme de l'étape N°1, afin d'afficher vers la fin de la deuxième ligne de l'afficheur LCD (position 14 de la ligne 2), le caractère correspondant à la touche enfoncée sur le clavier.

Les colonnes du clavier (Col1 à Col4) seront lues par des broches de port configurées en entrées. Il est indispensable d'utiliser des résistances de tirage (ici pull up) pour distinguer l'état de repos (ici = '1'), de l'état actif ou l'on appuie sur un bouton (=> '0' en pensant à mettre la ligne à '0').



Le clavier est ici volontairement connecté au PORTB afin de bénéficier des résistances de pull up intégrées sur le portB des PIC 16Fxxx.

Attention : Pour le PORTB uniquement les sorties peuvent être configurées avec les résistances de pull up, en mettant dans un registre appelé « OPTION », le bit 7 nommé /RBPU (= `portB Pull Up` bar) à 0 (car ce bit est actif à l'état bas). Voir commande ci-dessous.

```
OPTION_REG.F7=0; //Car il faut /RBPU = OPTION_REG.F7 à 0
```

II.1) Réalisez le câblage du clavier sur le PORTB pour avoir la correspondance ci -dessous:

RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
Col4	Col3	Col2	Col1	Li4	Li3	Li2	Li1

II.2) Adaptation du programme.

Modifiez le programme d'affichage précédent en ajoutant un sous programme de gestion du clavier sous forme de fonction.

Un sous programme appelé (fonction) nommé **scrut\_clav**, assurera la scrutation du clavier, et fournira le code ASCII de la touche enfoncée sur le clavier. Si aucune touche n'est enfoncée on devra renvoyer le code ASCII du caractère ' ? '.

Pour utiliser un sous programme en tant que fonction il suffit de renvoyer une valeur en définissant cette valeur par son type au début de la ligne de déclaration de la fonction.

Ainsi au lieu d'avoir :                    **void scrut\_clav (void)**

Nous aurons :                            **char scrut\_clav (void)**

Pour ainsi indiquer que le résultat sera de type char.

Il faudra alors placer le résultat dans une variable (Ex : variable « touche ») et utiliser l'instruction : « **return touche ;** » à la fin de la fonction **scrut\_clav**, avant l'accolade finale.

L'utilisation des fonctions se fait comme des variables. Ex : si l'on a créé une variable appelée « caract » de même type que le résultat de la fonction (ici : char),

il suffit d'écrire : **caract = scrut\_clav( )** ; pour récupérer dans « caract » la valeur de la touche détectée et décodée par la fonction **scrut\_clav**.

**Remarque importante :**

**Pour mettre le code ASCII d'un caractère dans la variable touche, il suffit d'utiliser l'une des syntaxes suivantes (Exemple pour les caractères 0, 3, 9, A, F, ?):**

**a) Soit chercher la valeur du caractère en hexadécimal ou en décimal à l'aide d'une table de correspondance caractères <=>hexa ou décimal.**

**touche = 0x30 ;**

**touche = 0x33 ;**

**touche = 0x39 ;**

**b) Pour les chiffres il suffit d'ajouter 0x30 ou le caractère '0', qui correspond justement à 0x30.**

**touche = 0x30 ou '0'**

**touche = 0x33 ; ou 3 + '0'**

**touche = 0x39 ; ou 9 + '0'**

**c) utiliser le caractère entre apostrophe**

**touche = 0x30 ou '0'**

**touche = 0x33 ; ou '3'**

**touche = 0x39 ; ou '9'**

**touche = 0x41 ; ou 'A'**

**touche = 'F'**

**touche = '?'**

**ALGORITHME à respecter pour la fonction gestion clavier : « scrut\_clav » :**

- a) Mettre le code par défaut dans touche : touche = '?' (si aucune touche n'est enfoncée).
- b) **Sélectionner la 1<sup>ère</sup> ligne** (remarque la ligne sélectionnée est mise à 0, les autres à 1).
- c) Lire les colonnes (quartet de poids fort du port B).  
*Rem : Si une touche est enfoncée, la colonne correspondante passe à 0, autrement elle est maintenue à 1 par sa résistance de pull up).*
- d) Pour chaque colonne, si la colonne est à 0, attribuer à la variable touche la valeur du code ASCII du caractère correspondant ('1', '2', '3', 'F' pour les touches 1,2,3,F de la première ligne).
- e) **Sélectionner la 2<sup>ème</sup> ligne**. (ne pas sélectionner les autres).
- f) Lire les colonnes (quartet de poids fort du port B).
- g) Pour chaque colonne, si la colonne est à 0, attribuer à la variable touche la valeur du code ASCII du caractère correspondant ('4', '5', '6', 'E' pour les touches 4,5,6,E de la deuxième ligne).
- h) **Faire de même avec la 3<sup>ème</sup> ligne**.
- i) **Faire de même avec la 4<sup>ème</sup> ligne**.
- j) Placer le résultat de la fonction (= code de la touche, ou '?' si aucune touche n'a été détectée).