## TP afficheur LCD avec PIC18 sur PICDEM2+ (ou autre)

Le TP décrit la procédure de mise en œuvre d'un afficheur LCD alphanumérique sur PIC18 avec le générateur de bibliothèque Microchip MAESTRO,.

- 1) A l'aide du data sheet de l'afficheur (ex : PC1602ARS de POWERTIP), indiquer les fonctions des broches D7-D0, E, RS, RW de l'afficheur LCD
- 2) Repérer sur le schéma de la carte (PICDEM2+) les connexions entre ces broches et la PIC18, tracer un tableau de correspondance
- 3) Lancer Maestro

Available Module	Rev.	Lang	Description	Selected Module			
RTC (Interrupt-driven)	1.0	Asse	RTC for PIC16 family	XLCD for C Languag	je		
10-bit ADC (Interrupt-driven)	1.0	Asse	For PIC18 only				
10-bit ADC (Polled)	1.0	Asse	For PIC18 only				
ADOver	1.00	Asse	Oversampling module for PIC16/PIC				
CANBoot	1.0	Asse	Simple CAN Bootloader for PIC18≫	:			
CAN driver (Interrupt driven)	1.1	С	CAN For PIC18Fxx8				
CAN driver(Interrupt driven)	1.0	Asse	CAN driver with Prioritized transmit b				
G2 DeviceNet Slave	1.00	С	DeviceNet Group 2 Slave for PIC18				
ECAN (Polled)	1.1	С	ECAN Routines PIC18+ECAN				
LIN Master (Interrupt-driven)	1.0	С	EUSART based for 18XXXX family				
I2CMaster (Interrupt-driven)	1.0	Asse	I2CMaster for PIC18/PIC16 family				
12CMaster (Polled)	1.0	Asse	I2CMaster for PIC16/PIC18 family				
I2CSIave (Interrupt-driven)	1.0	Asse	I2CSIave for PIC16/PIC18 family				
SPIMaster (Interrupt-driven)	1.0	Asse	SPIMaster for PIC18/PIC16 family				
SPIMaster (Polled)	1.0	Asse	SPIMaster for PIC16/PIC18 family				
SPISlave (Interrupt-driven)	1.0	Asse	SPISIave for PIC18/PIC16 family	Parameter	Value	Message	
SRALLOC	1.00	С	Simple SRAM Dynamic Memory Allo	lata face and	4 Dit interform	Intestage	
USART (Interrupt-driven)	1.0	Asse	USART for PIC16/18 family	Interrace mode	4 Bit Interrace	Interface with PIL controller	
USART (Interrupt-driven)	1.0	С	USART for PIC18 family	No of display lines	Single line	No of lines	
XLCD	1.0	Asse	LCD routines for PIC18/PIC16 family	Font selection	5x8	Font	
XLCD for C Language	1.0	С	LCD C routines for PIC18 family	Nibble selection	Lower hibble	[Unly in 4 bit mode]Higher	
				Data Port	PURIA	Port selection for data trans	
				LCD RS Pin	RAU	Help Message	
				LCD EN Pin	RAU	Help Message	
				LCD RW Pin	Ground	Help Message	
				BLUCKING	res	BLUCKING	
				Mode	Delay	mode selection for BLUUKI	
				Display Un	Yes	Display on	
				Display Cursor On	Yes	Cursor on	
				Display Blink On	Yes	Blink on	
				Entry mode cursor	Yes	Address and cursor increm	
•		111	•	I Entry mode Displa	No	Display shift during write an	

List of available configurable parameters for selected modules

Selectionner « XLCD for C language » et « tirer » vers la fenêtre « Selected module »

## Compléter les paramètres

Interface mode : indiquer le nombre de bits de communication (4 ou 8)

N° of display lines : indiquer le nombre de lignes de l'afficheur

**Font selection :** en observant un caractère de l'afficheur, compter la taille d'un caractère 5x8 ou 5x10 pixels

**Nibble selection** : si le mode de communication est sur 4 bits, indiquer si les bits utiliser par le PIC sont les quatre bits de poids fort ou les quatre bits de poids faible.

Dans les quatre lignes suivantes indiquer le câblage entre le PIC et l'afficheur.

**BLOCKING :** le logiciel doit il attendre que l'afficheur soit prêt avant de transmettre ? répondre « OUI » , certains afficheurs possèdent une mémoire interne tampon qui leur permet d'enchainer plus rapidement les commandes.

Mode : l'afficheur peut signaler qu'il est prêt à recevoir une donnée par la lecture d'un de ses registre, ce mode ne fonctionne pas sur tout les afficheurs , répondre « delay »
Display : ON, cursor ON, Blink ON, activer ces options rendra l'affichage visible avec un curseur clignotant.

**Entry mode cursor** : « OUI » entraine une incrémentation automatique des adresses lors de l'écriture.

Entry mode display : si « OUI » l'affichage est décalé lors des entrées de caractères, répondre « NON »

Une fois les paramètres renseignés, indiquer sur la ligne du haut la fréquence de l'oscillateur

du PIC puis cliquer Schoisir le dossier du projet, cinq fichiers sont créés Xlcd.h contient les définitions et prototypes de la bibliothèque Xlcd.C, source de la bibliothèque Xlcd.def ce fichier décrit entre autre les connexions entre le PIC et l'afficheur XLCD.P18.ex.txt, fichier de test, on peut renommer ce fichier en XLCD.P18.ex.C pour tester la bibliothèque dans un projet MPLAB

XLCDc.readme.pdf , mode d'emploi et description de la bibliothèque

4) Mise en œuvre et utilisation

Créer dans le projet MPLAB un fichier tst\_aff.c

```
#include <p18f4620.h> // dans le cas d'un PIC18F4620
#include "xlcd.h"
                     // bibliothèque bas niveau pour LCD
#include <stdio.h>
                    // bibliotheque standard , pour fprintf par exemple
// Recopier les lignes de tempo ci-dessous depuis le fichier XLCD.P18.ex.txt
void XLCDDelay15ms (void)
{
    int i;
   for(i=0;i<10000;i++)</pre>
    { Nop(); }
   return;
}
void XLCDDelay4ms (void)
{
    int i;
    for(i=0;i<2500;i++)
    { Nop(); }
   return;
}
void XLCD_Delay500ns(void)
{
    Nop(); Nop();
                   Nop();
}
void XLCDDelay(void)
   {
    int i;
    for(i=0;i<1000;i++)</pre>
                             }
      {
              Nop();
    return;
    }
// user_putc redirige le caractère à sortir vers la fonction d'emission de caractère
//de la bibliotheque xlcd
int _user_putc (char c)
// rechercher dans xlcd.h la fonction qui envoie un caractère vers l'afficheur et
l'exécuter ici
}
void main (void)
{
char a=45;
ADCON1=0x07; // positionne tout le port A en numerique
                                      //initialise LCD
XLCDInit();
                       // curseur en haut à gauche
// Rechercher dans xlcd.h le « define » qui positionne le curseur en haut à gauche
(commande 0x80) et l'exécuter ici
```

fprintf(\_H\_USER,"Test XLCD");

```
// curseur en bas à gauche
// Rechercher dans xlcd.h le « define » qui positionne le curseur en bas à gauche
(commande 0xC0) et l'exécuter ici
```

```
// Commenter et expliquer EN DETAIL la ligne ci dessous
fprintf(_H_USER,"dec:%d hex:%X",a,a);
while(1);
```

## Tester le programme

}



- 5) Utilisation de l'horloge interne du PIC
  Pour activer l'horloge interne du PIC il faut ajouter au projet le fichier init\_horloge\_interne.c
  et exécuter juste après le début de main, init\_horloge\_interne() ;
  L'horloge interne du PIC sera alors à 8MHz, reconfigurer la bibliothèque en conséquence et tester la nouvelle version
- 6) Affichage des nombres réels

A l'aide de la documentation sur les fonctions « fprintf » et « ftoa » ainsi que la bibliothèque math.h ( c:\mcc18\h\math.h )

Déclarer le réel float a=45.0; Calculer et afficher  $a^2$ ,  $\sqrt{a}$ ,  $\log(a)$ , sin ( $a^\circ$ ).

7) Bonus

Réaliser un programme de texte défilant... Le texte sera dans un tableau : char texte[]= « bonjour tout le monde » ; A votre guise...