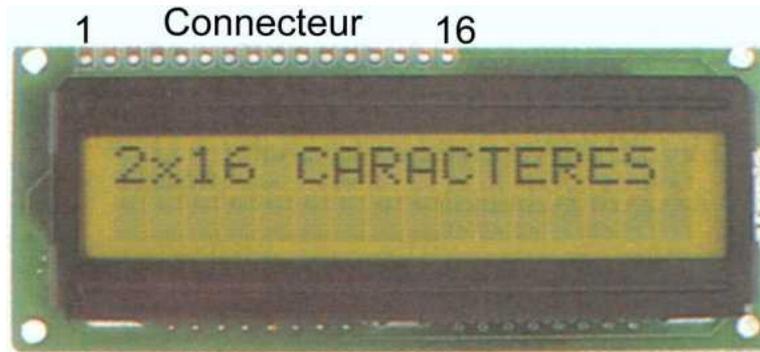


MSP 430

Gestion d'un LCD à interface HITACHI



Au même titre que le port "Centronics" d'un PC, le connecteur et la gestion des LCD dits "intelligents" sont devenus des standards.

En fait, la société HITACHI a développé il y a plusieurs années une famille de circuits intégrés de contrôle et de pilotage d'afficheurs constitués de caractères en matrice de points (5x8 ou 5x10) :

- HD44780 : contrôleur (la nouvelle version HD44780U intègre aussi un pilote permettant de gérer à lui tout seul 1 ligne de 16 caractères ou 2 lignes de 8 caractères).
- HDxxx (ou équivalents chez d'autres fabricants) : pilote chargé du multiplexage. Leur nombre dépend de la taille de l'afficheur.

Ces composants ont eu un succès fulgurant et aujourd'hui tous les fabricants de LCD proposent des modules de tailles diverses (1x8 caractères à 4x40 caractères ou plus) prêt à l'emploi et facile à mettre en œuvre.

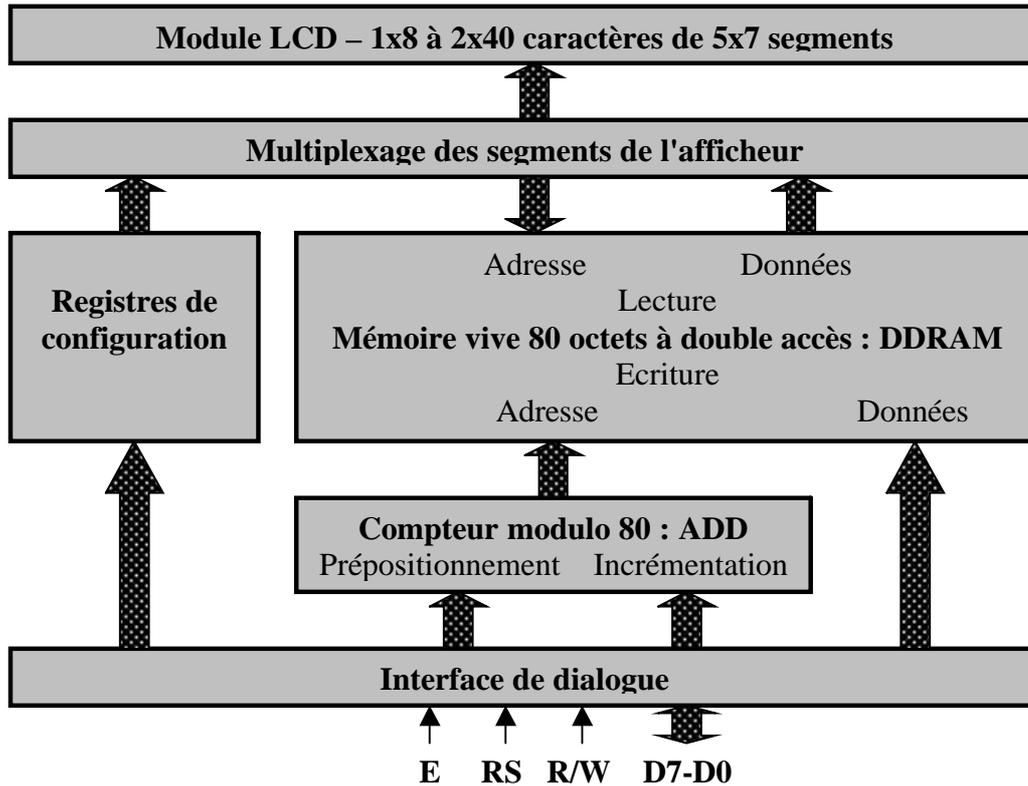
Pour l'utilisateur, un module LCD de ce type se comporte comme une RAM de 80 octets à accès **séquentiel** dont chaque case mémorise le code Ascii du caractère affiché. Ainsi, pour représenter la lettre 'C' sur la 6^o position de la 1^o ligne, il suffit d'affecter la case mémoire d'adresse 5 avec l'octet 43h (code Ascii de 'C').

Exemple : afficheur de 2 lignes de 16 caractères

Colonne	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Ligne 1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ligne 2	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Adresse RAM du module LCD																

Pour les petits afficheurs, certaines cases mémoire "sortent" de l'écran. Elles peuvent toutefois être visualisées par une modification de l'adresse de base de la lecture (initialement cette adresse vaut 0, mais elle peut être affecté avec n'importe quelle valeur entre 1 et 79). C'est un peu comme si on déplaçait une fenêtre de largeur 16 caractères devant un écran virtuel de largeur 40 caractères.

1. Schéma synoptique d'un afficheur



Note : les fonctions de gestion du curseur n'ont pas été représentées car celui-ci se met automatiquement dans la position correspondant à l'état du compteur ADD. L'utilisateur a toutefois la possibilité de changer sa forme (souligné, plein, clignotant).

L'utilisateur de l'afficheur doit s'acquies de 2 ensembles de tâches :

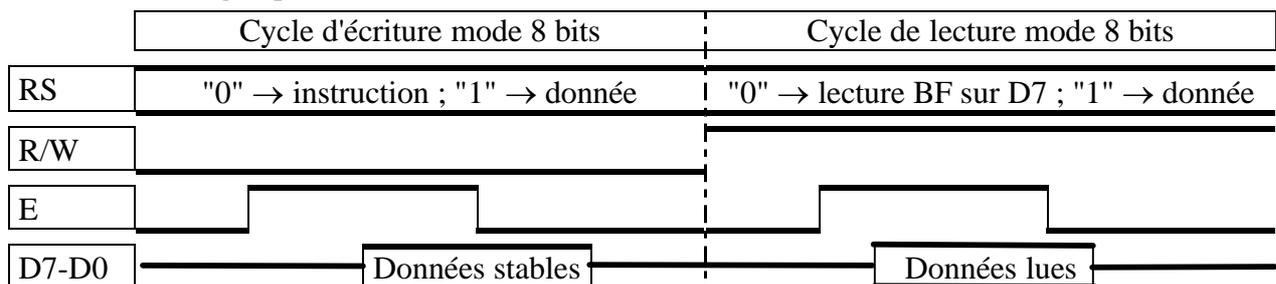
- Affecter les registres de configuration en fonction du format (nombre de lignes par exemple)
- Affecter la mémoire DDRAM avec le message à afficher

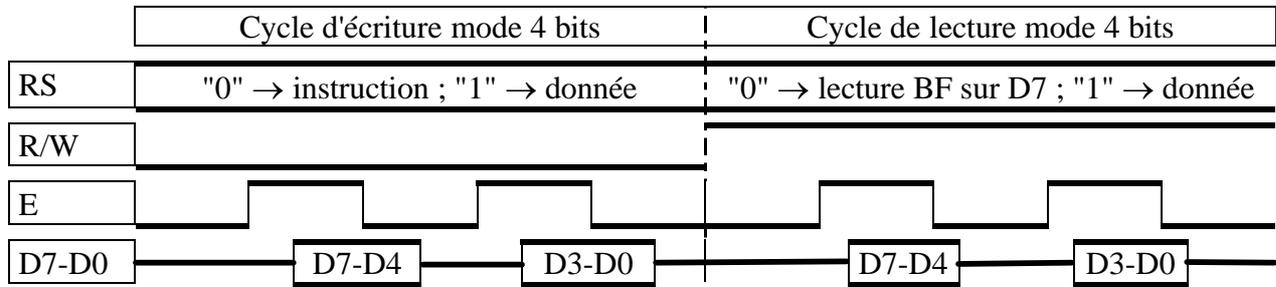
Pour ce faire, on dispose des signaux de contrôle E, RS et R/W et d'un bus de données 8 bits bidir. :

- E : Enable.
 - Cycle d'écriture : l'état du bus de données est pris en compte au \downarrow de ce signal
 - Cycle de lecture : la donnée est fournie pendant l'état H de ce signal
- R/W : indique le sens du transfert :
 - "0" : écriture vers l'afficheur
 - "1" : lecture depuis l'afficheur
- RS : Register Select. L'état de ce signal indique la destination de la donnée :
 - "0" : registres de contrôle
 - "1" : mémoire DDRAM
- D7-D0 : bus de données bidirectionnel

1.1 Cycles d'écriture et de lecture

L'interface de dialogue peut être utilisée en mode "8 bits" et en mode "4 bits" :





2. Affectation de la mémoire DDRAM

C'est l'opération la plus simple. En effet, il faut produire un cycle d'écriture avec RS="1" et une donnée valide (D7-D0) correspondant au code Ascii du caractère.

IMPORTANT : la donnée fournie est mémorisée à l'adresse ADD courante et le caractère est affiché à la position actuelle du curseur. Le compteur ADD est incrémenté automatiquement à la fin de cette opération (ce qui fait aussi avancer le curseur d'un cran) pour préparer l'écriture du caractère suivant.

3. Registres de configuration

Il s'agit d'affecter les indicateurs de fonctionnement suivants :

- DL** = 1 : interface en mode "8 bits"
0 : interface en mode "4 bits"
- N** = 0 : 1 ligne
1 : 2 lignes
- F** = 0 : caractères en 5x8 points
1 : caractères en 5x10 points. Non autorisé pour les afficheurs à 2 lignes
- D** = 0 : afficheur "éteint", mais le contenu de DDRAM est maintenu
1 : afficheur "allumé"
- C** = 0 : curseur caché
1 : curseur apparent (barre "souligné")
- B** = 0 : caractère du curseur affiché normalement
1 : caractère du curseur clignotant (positif/négatif)
- I/D** = 1 : le compteur d'adresse ADD s'**incrémente** à chaque caractère entré
0 : le compteur d'adresse ADD se **décrompte** à chaque caractère entré
- S** = 0 : le texte déjà affiché reste figé et le curseur se déplace à chaque caractère entré
1 : le curseur reste figé et le texte déjà affiché se déplace à chaque caractère entré

Note : au reset, chaque indicateur est affecté par le premier état décrit.

4. Affectations des registres de configuration et instructions

L'affectation des indicateurs de fonctionnement est réalisée par le biais d'instructions. Des instructions complémentaires permettent de réaliser certaines opérations.

Ecrire une instruction consiste à produire un cycle d'écriture avec RS="0". Les bits de poids fort de la donnée écrite sont utilisés pour identifier l'instruction et les poids faibles contiennent les paramètres.

4.1 Affectation des indicateurs de fonctionnement

INSTRUCTIONS	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Durée d'ex.
Function set	0	0	0	0	1	DL	N	F	*	*	40µS
Display on/off control	0	0	0	0	0	0	1	D	C	B	40µS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	40µS

Les états des bits de données représentés en gras sont utilisés pour identifier les instructions.

La valeur "*" indique un état indifférent.

Exemple : configuration d'un afficheur de 1 ligne de 16 caractères au format 5x8 pixels en mode 4 bits :

- Affecter RS = "0"
- Ecrire **00100000b** : Function set avec DL="0", N="0" et F="0"
- Ecrire **00001110b** : Display On, Curseur On, caractère non clignotant
- Ecrire **00000110b** : Mode Set : incrémentation ADD, texte figé

Attention : il faut attendre la fin du traitement de l'instruction en cours avant d'en écrire une nouvelle (voir § correspondant).

4.2 Autres instructions

INSTRUCTIONS	Durée d'ex.	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
Display clear	1,52mS	0	0	0	0	0	0	0	0	0	1
Cursor home	1,52mS	0	0	0	0	0	0	0	0	1	*
Cursor display shift	40µS	0	0	0	0	0	1	S/C	R/L	*	*
CG RAM address set	40µS	0	0	0	1	ACC					
DD RAM address set	40µS	0	0	1	ADD						

Les états des bits de données représentés en gras sont utilisés pour identifier les instructions.

La valeur "*" indique un état indifférent.

Display Clear :

- rempli la mémoire DDRAM avec le caractère " " (code Ascii 20h)
- Raz du compteur ADD
- Curseur sur la 1^o position

Cursor home :

- Raz du compteur ADD
- Curseur sur la 1^o position,
- Contenu DDRAM inchangé

Cursor display shift :

- S/C ,R/L =
- 0 0 : le curseur se déplace d'une position à gauche
 - 0 1 : le curseur se déplace d'une position à droite
 - 1 0 : le texte et le curseur se déplacent d'une position à gauche
 - 1 1 : le texte et le curseur se déplacent d'une position à droite

CG RAM address set : prépositionnement du compteur ACC d'adressage de la mémoire CGRAM avec le paramètre (le générateur de caractères personnels n'est pas développé ici).

DD RAM address set : prépositionnement du compteur ADD d'adressage de la mémoire DDRAM avec le paramètre. Le curseur est placé dans la position correspondante au caractère (voir exemple de tableau page 1)

Exemple : afficher le message 'XYZ' à partir de la 5^o position de la 1^o ligne

- Affecter RS = "0"
- Ecrire 10000100b : Prépositionnement de ADD avec la valeur 4. Le curseur est déplacé sur la 5^o position de la 1^o ligne.
- Affecter RS = "0" (affectations DDRAM)
- Ecrire 'X' : affectation de la case mémoire d'adresse 4 de DDRAM avec le code Ascii 58h. ADD s'incrémente et le curseur se déplace d'un cran.
- Ecrire 'Y' : affectation de la case mémoire d'adresse 5 de DDRAM avec le code Ascii 59h. ADD s'incrémente et le curseur se déplace d'un cran.
- Ecrire 'Z' : affectation de la case mémoire d'adresse 6 de DDRAM avec le code Ascii 5Ah. ADD s'incrémente et le curseur se déplace d'un cran.

4.3 Procédure d'initialisation de l'afficheur

Pour initialiser de façon fiable le contrôleur HD44780 (le nouveau modèle HD44790U est plus souple) :

1. Attendre plus de 15mS après la mise sous tension
 2. Ecriture instruction "Function Set" avec DL = "1" (mode 8 bits). Seuls les 4 bits MSB sont pris en compte.
 3. Attendre au moins 4,1mS
 4. Répéter 2 fois les opérations 2. et 3. avec des délais d'attente successifs de 500µS et 100µS.
 5. Si l'interface est en mode 4 bits : écriture instruction "Function Set" avec paramètre DL = "0"
- A partir de ce point, et en mode "4 bits", chaque écriture d'une instruction ou d'une donnée nécessite 2 cycles d'écriture.
6. Ecriture instruction "Function Set" avec les paramètres DL, N et F définitifs

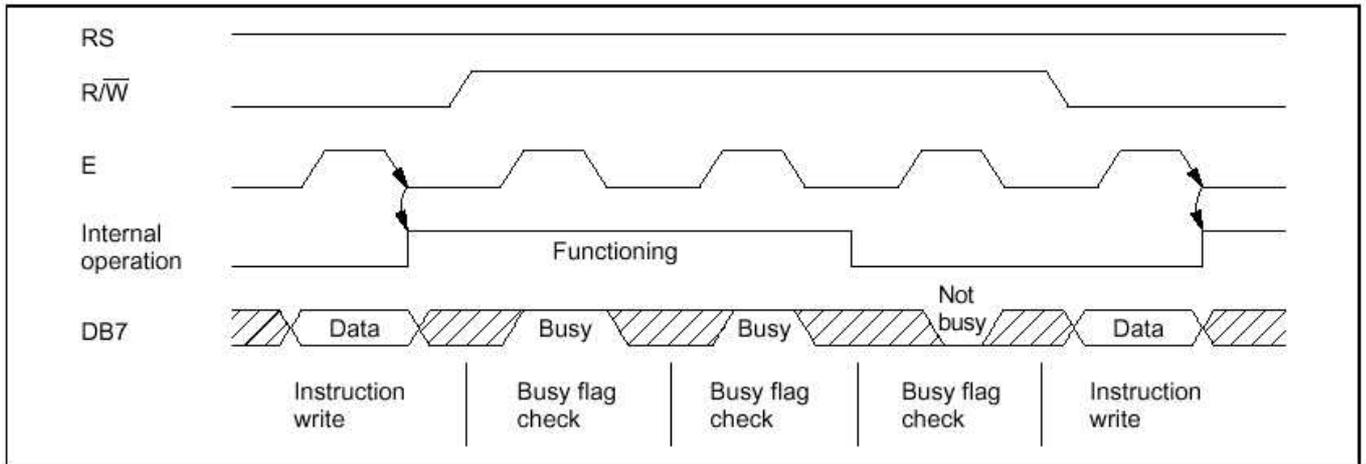
7. Ecriture instruction "Display control" pour affecter les paramètres D, C et B.
8. Ecriture instruction "Display clear"
9. Ecriture instruction "Entry mode set" pour affecter les paramètres I/D et S

4.4 Indicateur d'état

L'indicateur BF (pour Busy Flag) est le seul lisible directement par l'utilisateur. Son état indique l'activité du contrôleur de l'afficheur :

- "1" : une opération est en cours. Aucune nouvelle instruction n'est acceptée
- "0" : pas d'opération en cours. Une nouvelle instruction peut être chargée.

L'état de BF est placé sur le bit D7 pendant un cycle de lecture avec RS = "0".



Note : il est possible de se passer de la lecture de cet indicateur si on respecte les temps d'exécution des instructions avant de charger la suivante.

5. Brochage du connecteur

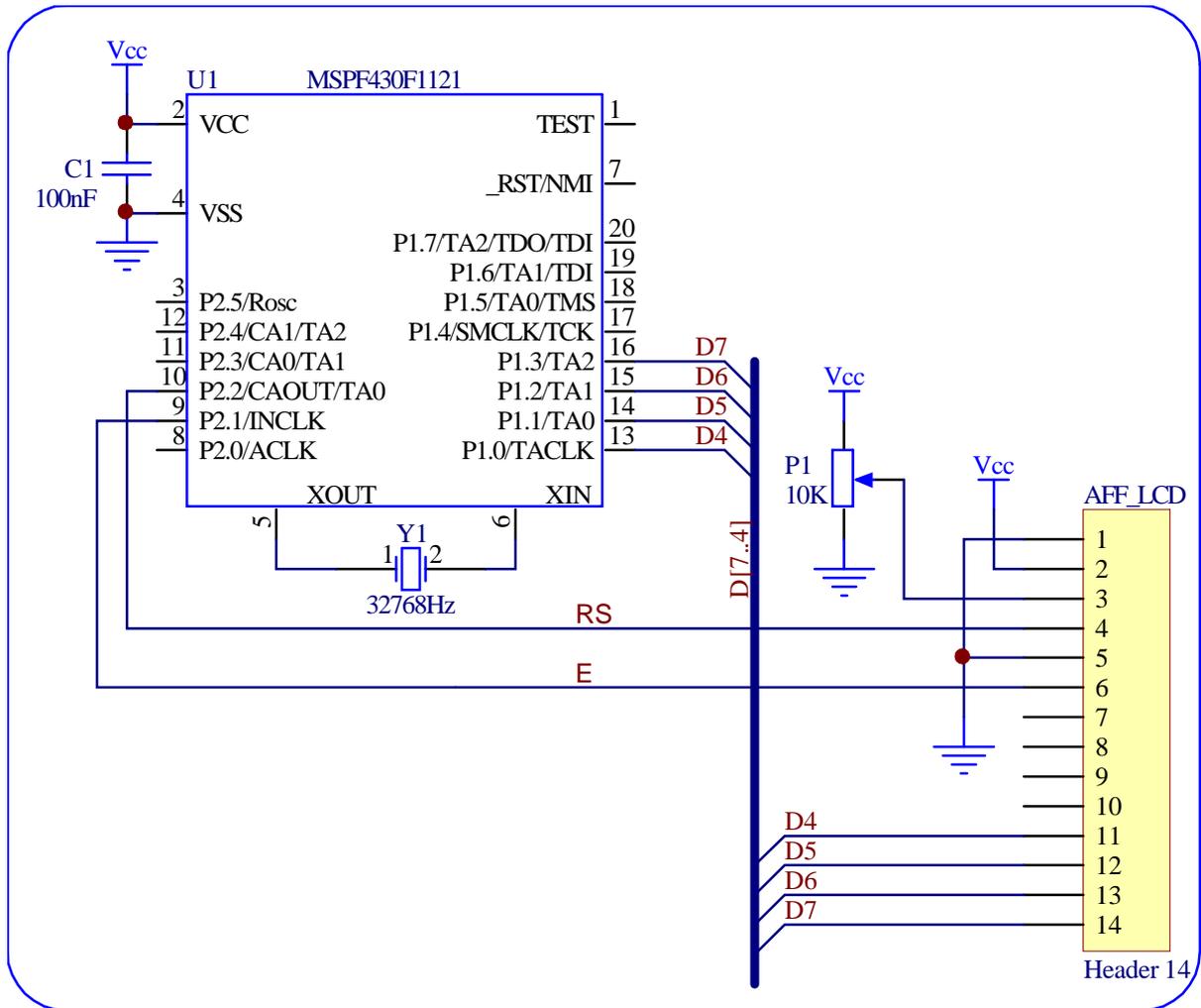
Les connecteurs peuvent être du type SIL (Single In Line) ou DIL (Dual In Line) à 14 (sans rétro-éclairage) ou 16 broches.

Broche N°	Nom	Niveau	Fonction
1	V _{ss}	0V	Alimentation
2	V _{cc}	2,7V à 5,5V	Alimentation
3	V ₀	-5V à 5V	Contraste
4	RS	H/L	0 = instruction 1 = caractère
5	R/W	H/L	0 = écriture 1 = lecture
6	E	H --> L	Enable (front descendant)
7	D0	H/L	Donnée bidirectionnelle LSB
8	D1	H/L	Donnée bidirectionnelle
9	D2	H/L	Donnée bidirectionnelle
10	D3	H/L	Donnée bidirectionnelle
11	D4	H/L	Donnée bidirectionnelle
12	D5	H/L	Donnée bidirectionnelle
13	D6	H/L	Donnée bidirectionnelle
14	D7	H/L	Donnée bidirectionnelle MSB
15	A		Anode - LED de rétro éclairage
16	K		Cathode - LED de rétro éclairage

Attention : sur certains modèles (notamment de marque OPTREX), les rôles des broches 1 et 2 sont inversés !

Les niveaux H et L en entrées et sorties sont conformes à la technologie CMOS série 4000.

6. Mise en œuvre avec un MSP430F1121



Choix techniques :

- Interface en mode "4 bits" pour réduire le nombre d'E/S du microcontrôleur dédiés au LCD
- R/W (broche 5) connecté à GND pour la même raison. Il sera donc impossible de produire des cycles de lecture, notamment pour lire l'indicateur BF.
- P1 règle le niveau L des tensions d'alimentation des cristaux liquides (le niveau H étant Vcc) et par conséquent le contraste de l'afficheur.

6.1 Déclaration des constantes liées au LCD

```
E      equ 00000010b  ;E = P2.1
RS     equ 00000100b  ;RS = P2.2
```

```
* Instructions du HD44780 de HITACHI
*****
```

```
CLRLCD equ 00000001b ;Ecrit des espaces ($20) ds toutes les positions, place
*          le compteur à 0 et le curseur ds la 1ere position. Mode
*          incrémentation (1,6mS)
HOMLCD  equ 00000010b ;Idem, mais contenu RAM non modifié (1,6mS)
RIGHTC  equ 00000110b ;Le curseur se déplace vers la droite à chaque caractere
*          entré (40µS)
LEFTC   equ 00000100b ;Idem, mais vers la gauche (40µS)
RIGHTD  equ 00000101b ;Mode insertion. Le curseur reste sur place et la ligne
*          se déplace vers la droite (40µS).
LEFTD   equ 00000111b ;Idem, mais vers la gauche (40µS).
DISON   equ 00001100b ;Display On          +
```

```

DISOFF equ 00001000b ;Display Off | A combiner dans une
CURON equ 00001010b ;Curseur On +- seule instruction
CUROFF equ 00001000b ;Curseur Off | (40µS).
CARBLK equ 00001001b ;Caractere clignotant |
CARNBL equ 00001000b ;Caractere non clignotant +
SHFTCR equ 00010100b ;Curseur déplacé à droite. AC incrémenté (40µS).
SHFTCL equ 00010000b ;Curseur déplacé à gauche. AC décrémenté (40µS).
SHFTDR equ 00011100b ;Décalage affichage et curseur vers la droite (40µS).
SHFTDL equ 00011000b ;Décalage affichage et curseur vers la gauche (40µS).
FUNSET4 equ 00100000b ;Mode 4 bits. 1 ligne en 5x7 dots (40µS).
FUNSET8 equ 00111000b ;Mode 8 bits. 1 ligne en 5x7 dots (40µS).
CGRAM equ 01000000b ;Les 6 bits Lsb spécifient l'adresse de la RAM
* g n rateur de caracteres (40µS).
DDRAM equ 10000000b ;Les 7 bits Lsb sp cifient l'adresse de la RAM  cran
* (0   39 = 1ere ligne, 64   103 = 2ieme ligne) (40µS).
    
```

6.2 Sous-programme d'initialisation

Les ports du microcontroleur ont  t  pr alablement configur s :

- P2.1 et P2.2 en "sortie" et affect s   "0"
- P1.3   P1.0 en "sortie". Aucun conflit avec le port du LCD n'est possible car R/W = "0".

Le sous-programme propos  respecte l'algorithme recommand  par HITACHI pour un afficheur de 1 ligne de 16 caract res.

Attention : l'indicateur BF ne pouvant  tre lu, il faut imp rativement respecter le temps d'execution de la derni re instruction charg e dans le LCD avant d'y  crire la suivante.

Les sous-programmes propos s respectent ces d lais si l'horloge du CPU est configur e comme suit :

- MCLK = DCOCLK
- Rsel = DCO = MOD = 0

Dans ces conditions, la fr quence de MCLK est de 80kHz environ.

```

* INIT_LCD_4
*****
* Fonction : Initialisation de l'afficheur LCD
*           - mode 4 bits
*           - texte immobile, curseur vers la droite
*           - afficheur vide
* Registres R4 et R5 modifi s
INIT_LCD_4 bic.b #(E bitor RS),&P2OUT ;RS="0" car  criture d'instructions
            mov #465,r5
            call #WAIT_TIME ;D lai 20mS

            mov #FUNSET8,r4 ;Function Set. Mode 8 bits
            call #WR_LCD_QH ;Ecriture du quartet H dans le LCD
            mov #116,r5
            call #WAIT_TIME ;D lai 5mS

            mov #FUNSET8,r4 ;Function Set. Mode 8 bits
            call #WR_LCD_QH ;Ecriture du quartet H dans le LCD
            mov #11,r5
            call #WAIT_TIME ;D lai 500µS

            mov #FUNSET8,r4 ;Function Set. Mode 8 bits
            call #WR_LCD_QH ;Ecriture du quartet H dans le LCD
            mov #3,r5
            call #WAIT_TIME ;D lai 130µS

            mov #FUNSET4,r4 ;Function Set. Mode 4 bits
            call #WR_LCD_QH ;Ecriture du quartet H dans le LCD
            mov #3,r5
            call #WAIT_TIME ;D lai 130µS

            mov #FUNSET4,r4 ;Function Set. Mode 4 bits. 1ligne, 5x7 dots 1/16
            call #LCDWI_4 ;Ecriture de l'instruction en mode 4 bits
    
```

MSP430 – Gestion d'un LCD HITACHI

```

mov    #(DISON bitor CURON bitor CARBLK),r4 ;Afficheur allumé et curseur
                                             ;clignotant
call   #LCDWI_4      ;Ecriture de l'instruction

mov    #CLRLCD,r4   ;Afficheur vide. Curseur sur 1er caractere
call   #LCDWI_4     ;Ecriture de l'instruction

mov    #RIGHTC,r4   ;Texte immobile, curseur vers la droite
call   #LCDWI_4     ;Ecriture de l'instruction
ret

```

* WAIT_TIME

* Fonction : délai temporel

* Parametre : registre r5 = délai. Résolution 43µs (MCLK=DCOCLK avec Rsel=DCO=MOD=0)

```

WAIT_TIME  dec r5      ;
            jne WAIT_TIME ;
            ret        ;

```

* WR_LCD_QH et WR_LCD_QL

* Fonction :

* WR_LCD_QH : écriture du quartet de poids fort (H) de r4.b dans le LCD

* WR_LCD_QL : écriture du quartet de poids faible (L) de r4.b dans le LCD

* Paramètre : WR_LCD_QH : bits 7 à 4 de r4

* WR_LCD_QL : bits 3 à 0 de r4

* Note : RS doit être préalablement affecté

```

WR_LCD_QH  rra.b r4
            rra.b r4
            rra.b r4
            rra.b r4
WR_LCD_QL  and.b #00001111b,r4      ;Raz des 4 bits de poids fort
            mov.b r4,&P1OUT          ;Sortie du quartet
            bis.b #E,&P2OUT          ;
            bic.b #E,&P2OUT          ;Ecriture ds le LCD
            ret

```

* LCDWI_4

* Fonction : Envoi d'une instruction en mode 4 bits

* Parametre : registre r4 = code de l'instruction

* Note : compte tenu de la vitesse du CPU, le temps d'exécution du programme

* entre 2 appels successifs de LCDWI_4 respecte largement le délai de 40µs

* Le sous-programme WAIT_TIME n'est appelé que pour les instructions

* CLRLCD ou HOMLCD

```

LCDWI_4    mov    r4,r5      ;Sauvegarde dans r5
            bic.b #RS,&P2OUT ;Car instruction
            call  #WR_LCD_QH ;Ecriture du quartet de poids fort de l'instruction
            mov    r5,r4     ;Restitution de l'instruction dans octet de poids faible
            call  #WR_LCD_QL ;Ecriture du quartet de poids faible de l'instruction
            mov    r5,r4     ;Restitution de l'instruction dans octet de poids faible
            mov    #40,r5    ;Délai long à priori
            cmp.b #CLRLCD,r4 ;
            jeq  WAIT_DELI4 ; |
            cmp.b #HOMLCD,r4 ; > Instructions CLRLCD ou HOMLCD -> délai 1,7mS
            jne  FIN_LCDWI  ;/
WAIT_DELI4 call  #WAIT_TIME ;Attendre fin de l'opération
FIN_LCDWI  ret

```


6.4 Exemple

L'exemple proposé affiche la phrase "Lycee Louis Couffignal – 11, route de la Federation – 67100 STRASBOURG" sur un afficheur de 1 ligne de 16 caractères, puis la fait défiler de droite à gauche.

```

ORG    DEBROM
* RESET
*****
RESET  mov    #WDTPW+WDTHOLD, WDTCTL ;
        mov    #FINRAM+1, SP        ; initialisation du stack pointer

* Configuration de l'horloge CPU
        mov.b  #00000110b, &BCSCTL2 ;MCLK=DCOCLK, SMCLK=DCOCLK/8, DCO:rés. interne
        mov.b  #10110000b, &BCSCTL1 ;XT2 off, XT1:low, Rsel=0:fréq. Nom. la + basse
        mov.b  #00000000b, &DCOCTL  ;DCO=0 et MOD=0
        mov    #0000000010100000b, R2 ;SCG1=1 : SMCLK inhibée
*
*                                     SCG0=0 : DCO validé
*                                     OscOff=1 : XT1 inhibé
*                                     CPUOff=0 : CPU opérationnelle
* Configuration des ports
        bis.b  #0Fh, &P1DIR          ;Port1.0 à P1.3 en sortie
        bis.b  #(E bitor RS), &P2DIR ;E et RS en sortie
        bic.b  #(E bitor RS), &P2OUT ;E et RS à 0
        bic.b  #0Fh, &P1OUT          ;P1.0=D4 à P1.3=D7 à 0

* Initialisation du LCD
        call   #INIT_LCD_4           ;Initialisation du LCD en mode 4 bits

* Message de test
        mov    #MSG_TEST, r6
        mov    #0, r4                ;1° ligne, 1° position
        call   #LCDWRT

* Boucle principale
*****
BOUCLE  mov    #5000, r5
        call   #WAIT_TIME           ;Délai de 180mS environ
        mov    #SHFTDL, r4          ;Décalage affichage et curseur d'un cran vers la droite
        call   #LCDWI_4             ;Ecriture de l'instruction
        br     #BOUCLE              ;

MSG_TEST db    'Lycee Louis Couffignal - 11, route de la Federation - 67100
STRASBOURG '
        db    EOT                    ;ATTENTION : le nb d'octets total doit être pair

```