

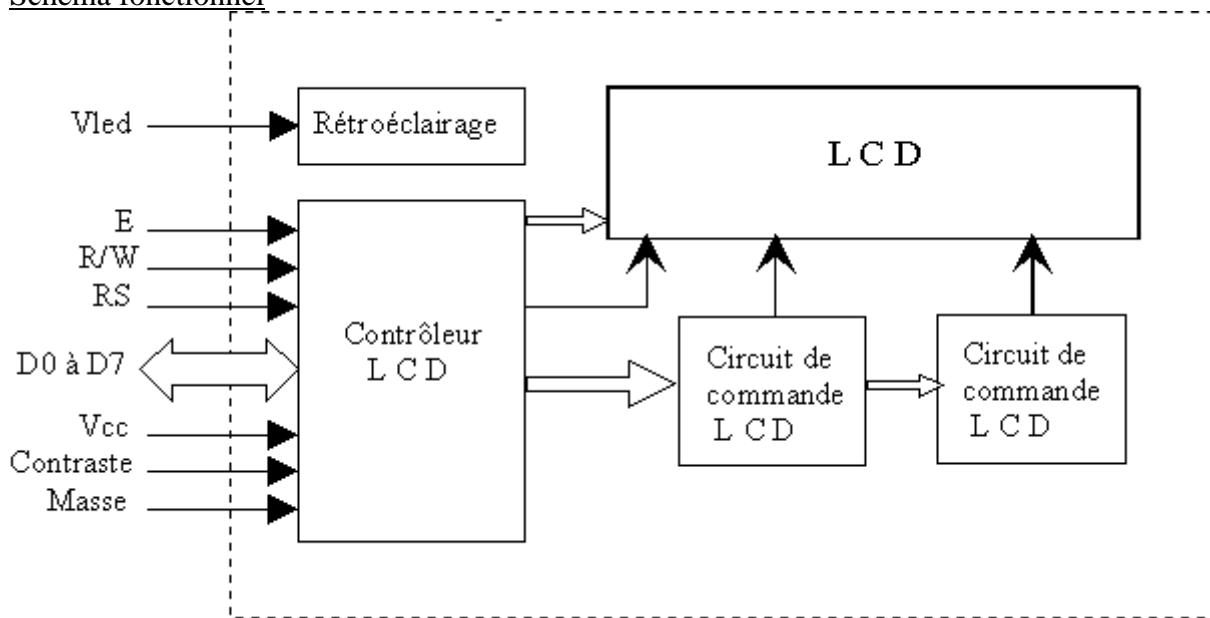
# Afficheur LCD

## Généralités

- Les afficheurs à cristaux liquides sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils sont relativement bons marché et s'utilisent avec beaucoup de facilité.
- Un microprocesseur "pilote" de la famille C-MOS diminue considérablement leur consommation (inférieur à 0.1 mW). Ils sont pratiquement les seuls à être utilisés sur les appareils à alimentation par piles.
- Plusieurs afficheurs sont disponibles sur le marché et ne diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (250 mA max.).

## Principe de fonctionnement

### Schéma fonctionnel



Comme le montre le schéma fonctionnel, l'affichage comporte d'autres composants que l'afficheur à cristaux liquides (LCD) seul. Un circuit intégré de commande spécialisé, le *LCD-controller*, est chargé de la gestion du module. Le "contrôleur" remplit une double fonction: d'une part il commande l'affichage et de l'autre se charge de la communication avec l'extérieur.

## Connexions

Les connexions à réaliser sont simples puisque l'afficheur LCD dispose de peu de broches. Il faut, évidemment, l'alimenter, le connecter à un bus de donnée (4 ou 8 bits) du microcontrôleur, et connecter les broches **Enable** (validation), **Read/Write** (écriture/lecture) et **Register Select** (caractère/commande).

### Description de l'afficheur

- L'afficheur comporte 2 lignes de 16 caractères.
- L'électronique est compatible C-MOS et TTL et sa consommation ne dépasse pas 7mW. Ses entrées sont protégées par des diodes.
- Outre les 32 caractères visualisés sur l'affichage, il est possible de mettre en mémoire 48 caractères supplémentaires, caractères que l'on visualisera sur l'affichage au moment voulu.

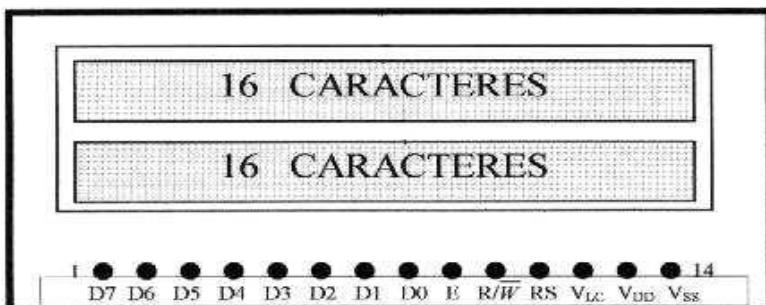
### Possibilités de l'afficheur

- L'afficheur est en mesure de visualiser 192 caractères:
- de \$00 à \$07 : 8 caractères définissables par l'utilisateur
- de \$20 à \$7F : 96 caractères ASCII (majuscules, minuscules, chiffres, signes)
- de \$A0 à \$DF: 64 caractères japonais ( alphabet kana)
- de \$E0 à \$FF : 32 caractères spéciaux: accent, lettres grecques, ...

### Autres commandes

- L'extinction de l'affichage
- Le positionnement du curseur
- Le déplacement des caractères sur l'affichage
- Le choix du caractère à redéfinir
- Le choix du sens du déplacement du curseur ou de l'affichage
- Le clignotement des caractères ou du curseur
- ...

### Brochage du composant



## Niveau des différentes broches

N° DE BROCHE	SIGNAL	NIVEAU
14	VSS	Masse
13	VDD	+ 5 V
12	VLC	$\pm 2,5$ V
11	RS	0 = commande 1 = caractère.
10	R/W	0 = écriture 1 = lecture
9	E	Front descendant
8	D0	Logique positive
7	D1	Logique positive
6	D2	Logique positive
5	D3	Logique positive
4	D4	Logique positive
3	D5	Logique positive
2	D6	Logique positive
1	D7	Logique positive

## Description des différentes broches

- 1 à 8 : D7 à D0 : Bus de données bidirectionnel
- 12 : VLC : Cette tension permet le réglage du contraste de l'afficheur.
- 13 : VDD : + 5 V
- 14 : VSS : Masse.

## Description des différentes broches

- 9 : E : Entrée de validation (ENABLE) ; elle est active sur front descendant. Il est important ici de tenir compte des 2 seuils durées de commutation importantes en pratique: lorsque RS et R/W ont atteint un niveau stable, il doit se passer un intervalle de 140 ns minimum avant que la ligne "E" ne passe au niveau haut. Cette ligne doit ensuite, être maintenue à ce niveau pendant 450 ns au moins et les données doivent rester stables sur le bus de données jusqu'au début du flanc descendant de ce signal. Lorsque E=0 les entrées du bus de l'afficheur sont à l'état haute impédance.
- 10 : R/W : Lecture ou écriture.(READ/WRITE)  
Lorsque R/W est au niveau bas, l'afficheur est en mode "écriture", et lorsque R/W est au niveau haut, l'afficheur est en mode "lecture".
- 11 : RS: Sélection du registre.(REGISTER SELECT)  
Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée(caractère). Un niveau bas indique une commande et un niveau haut indique une donnée.

## Apparition des caractères sur l'afficheur

Après avoir défini le sens de déplacement, les caractères apparaissent au-dessus du curseur (qu'il soit visualisé ou non).

Adresse	gauche	droite	invisible
haut	\$00 .....\$0F	\$10 .....\$27	
bas	\$40 .....\$4F	\$50 .....\$67	

- L'adresse 00 correspond à la ligne du haut à gauche, 0F à droite.
- L'adresse 40 correspond à la ligne du bas à gauche, 4F à droite.
- La zone invisible correspond à la mémoire de l'afficheur.(48 caractères)
- Lorsqu'un caractère est inscrit à l'adresse \$27, le caractère suivant apparaît à la ligne suivante.

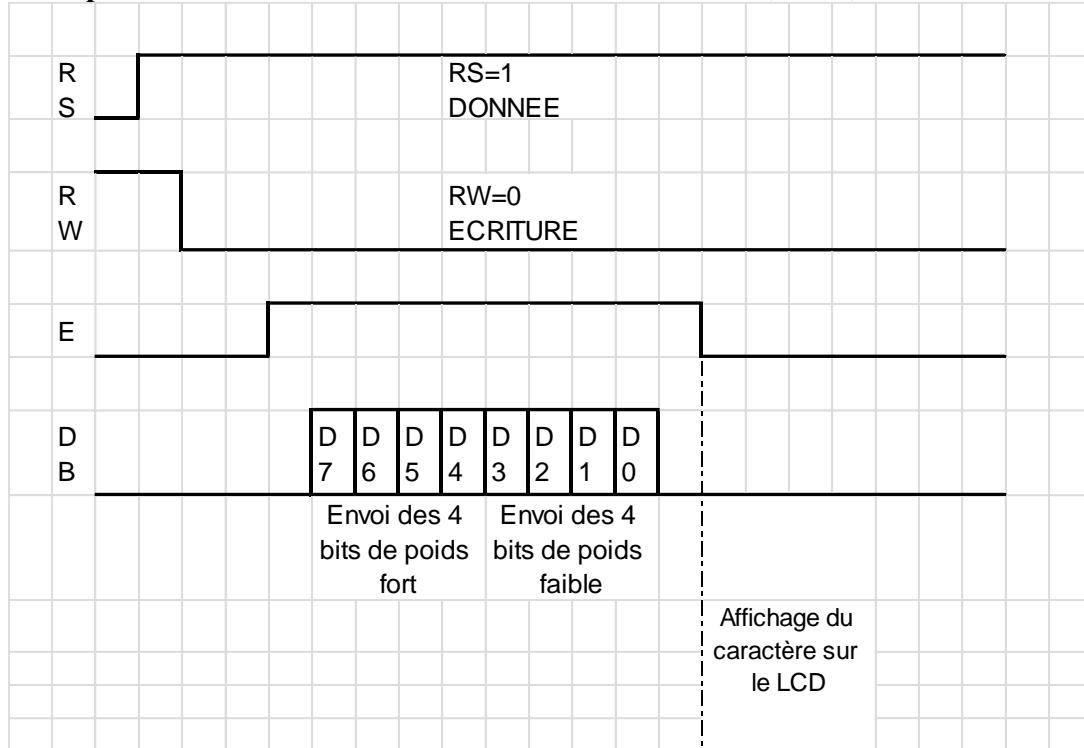
### Principe de fonctionnement

- Pour visualiser un caractère, il suffit de le positionner sur le bus de donnée (codé en ASCII), de mettre RS au niveau haut (caractère), R/W au niveau bas (écriture), et de provoquer un front descendant sur l'entrée de validation de l'afficheur (E).
- **ATTENTION:** Après chaque action sur l'afficheur, il faut vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela il faut aller lire l'adresse de la position du curseur (RS=0, R/W =1) et tester l'indicateur flag "Busy" (BF) (voir tableau ci-dessous). Lorsque BF=1 l'affichage est en cours et lorsque BF=0, l'affichage est terminé.

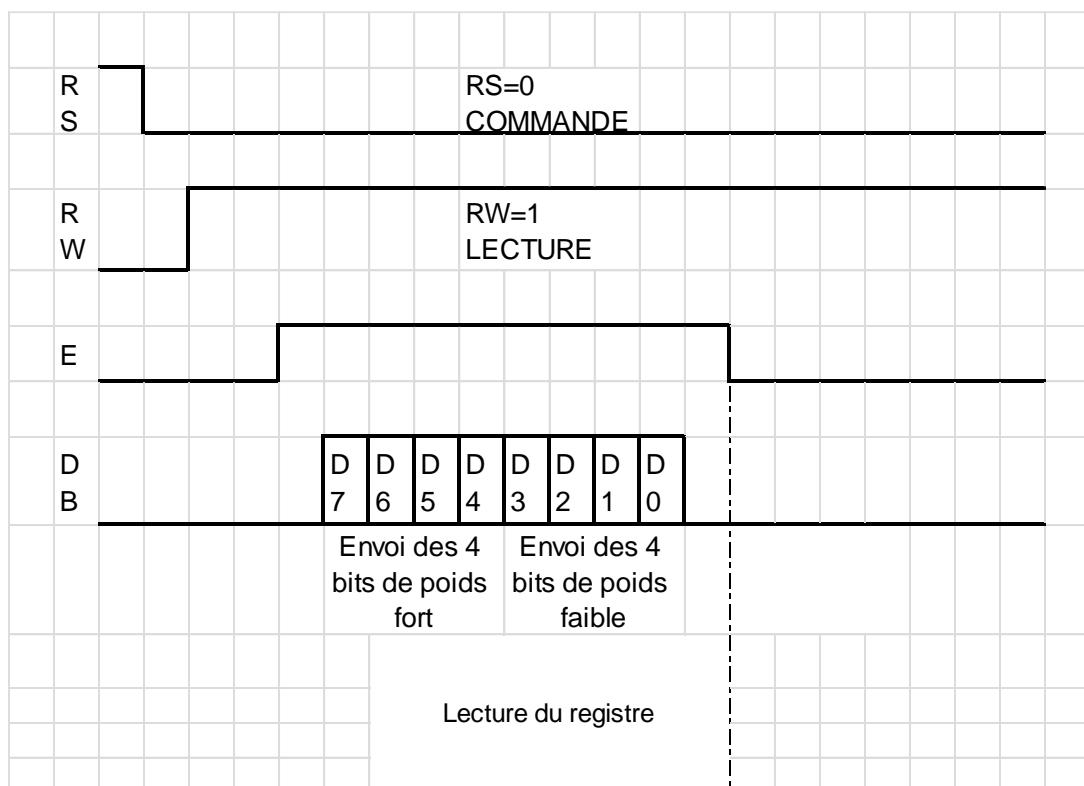
### Application pratique

- Avant toute cette procédure d'écriture, il faut tout d'abord initialiser l'afficheur de façon à définir son mode de fonctionnement (dialogue en 4 ou 8 bits, sens de déplacement, visualisation du curseur, ...). Cette initialisation s'effectue en lui adressant des commandes (voir tableau ci-dessous), c'est à dire en positionnant RS à 0.
- La première commande à envoyer est la commande permettant de définir le mode de dialogue avec l'afficheur (**DL**), le nombre de lignes sélectionnées (**N**), la taille de la matrice(**F=0**). Pour cela, on met RS et R/W au niveau bas et on positionne sur le bus de données le code correspondant au choix de l'utilisateur (cette commande efface la ligne supérieure). Après un front descendant de l'entrée E, l'information sera prise en compte. A partir de maintenant, l'ordre des commandes n'a plus d'importance.
- Envoyons par exemple la commande 01h (effacement de l'afficheur), puis la commande 0Eh, soit D=1, C=1, B=0, pour mettre le curseur en fonction.
- Ensuite, il faut définir le sens de déplacement du curseur, pour cela, envoyons la commande 06h, soit ID=1 et S=0.
- Ceci étant fait, on peut maintenant commencer à envoyer les premiers caractères à afficheur. Il faut faire passer la ligne RS au niveau haut et envoyer sur le bus de données les codes des caractères à afficher. Après chaque front descendant de l'entrée E, le caractère sera affiché à la position du curseur.

### Exemple d'écriture d'un caractère. Données sur 8 bits (DL=1)



### Exemple de lecture d'une commande. Données sur 8 bits (DL=1)



**Tableau des différentes commandes de l'afficheur**

TYPE DE COMMANDE	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	DESCRIPTIF				
<b>EFFACER L'AFFICHAGE</b>	0	0	0	0	0	0	0	0	0	1	Efface l'ensemble de la mémoire de données sans toucher au générateur de caractère. Met le curseur en position Home, à l'adresse 00.				
<b>CURSEUR EN POSITION HOME</b>	0	0	0	0	0	0	0	0	1	*	Met le curseur en position Home. Si l'affichage a été décalé, il est remis à sa position d'origine: l'adresse 00 se trouve à nouveau en haut à gauche.				
<b>MANIERE DE VISUALISER LES CARACTERES</b>	0	0	0	0	0	0	0	1	ID	S	Détermine le sens de déplacement du curseur après apparition d'un caractère ( <b>ID</b> ) et le déplacement collectif d'une position de l'ensemble de l'affichage ( <b>S</b> ).				
<b>MARCHE/ARRET DE L'AFFICHAGE DU CURSEUR</b>	0	0	0	0	0	0	1	D	C	B	Met l'affichage en ou hors fonction ( <b>D</b> ). Met le curseur en ou hors fonction ( <b>C</b> ). Fait clignoter le caractère situé au-dessus du curseur ( <b>B</b> ), clignotement se traduisant par une alternance du caractère et du caractère FF (rectangle noir)				
<b>DECALAGE</b>	0	0	0	0	0	1	S/C	R/L	*	*	Déplace le curseur ou l'ensemble de l'affichage sans modifier le contenu de la mémoire.				
<b>FONCTION</b>	0	0	0	0	1	DL	N	F	*	*	Indique la largeur du bus de données. Indique s'il ne faut utiliser que la ligne du haut ou que celle du bas. ( <b>F</b> ) : matrice				
<b>ADRESSE DU GENERATEUR DE CARACTERES</b>	0	0	0	1	Caractère			Rangée			Définit l'adresse de la mémoire du générateur de caractères. Les données suivantes correspondent à la matrice du caractère concerné.				
<b>ADRESSE DE LA MEMOIRE DE DONNEES</b>	0	0	1	Adresse							Définit l'adresse de la mémoire de données. Les données suivantes correspondent au caractère ASCII à visualiser.				
<b>INDICATEUR BUSY LECTURE D'ADRESSE</b>	0	1	BF	Adresse							Lit l'indicateur Busy ( <b>BF</b> ) pour vérifier que l'afficheur est en mesure de traiter la commande suivante. Lit l'adresse de la position du curseur.				
<b>ECRITURE DE DONNEES</b>	1	0	Données									Ecrit des données respectivement dans la mémoire de données ou le générateur de caractères.			
<b>LECTURE DE DONNEES</b>	1	1	Données									Lit les données respectivement de la mémoire de données ou le générateur de caractères.			

## Description des différentes commandes.

	<b>0</b>	<b>1</b>
<b>ID</b>	Déplacement vers la gauche	Déplacement vers la droite
<b>S</b>	L'affichage ne bouge pas	L'affichage est décalé
<b>D</b>	Absence affichage	Visualisation affichage
<b>C</b>	Absence du curseur	Visualisation du curseur
<b>B</b>	Absence de clignotement du caractère	Clignotement du caractère
<b>S/C</b>	Déplacement du curseur	Déplacement de l'affichage
<b>R/L</b>	Décalage vers la gauche	Décalage vers la droite
<b>DL</b>	4 bits	8 bits
<b>N</b>	Ligne du haut	2 lignes validées
<b>F</b>	Matrice 5 x 8 points	Matrice 5 x 11 points