

# Les afficheurs à cristaux liquides

## Introduction

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité.

Plusieurs afficheurs sont disponibles sur le marché et 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 rétroéclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA).



Ils sont très utilisés dans les montages à microcontrôleur, et permettent une grande convivialité. Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement y afficher les valeurs de différentes variables.

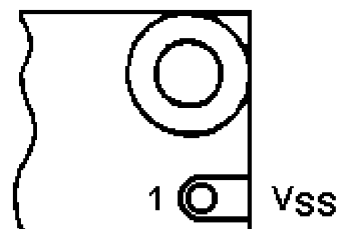
## Principe des cristaux liquides

L'afficheur est constitué de deux lames de verre, distantes de 20  $\mu\text{m}$  environ, sur lesquelles sont dessinées les mantisses formant les caractères. L'espace entre elles est rempli de cristal liquide normalement réfléchissant (pour les modèles réflectifs). L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant. Les caractères apparaissent sombres sur fond clair. N'émettant pas de lumière, un afficheur à cristaux liquides réflectif ne peut être utilisé qu'avec un bon éclairage ambiant. Sa lisibilité augmente avec l'éclairage. Les modèles transmissifs fonctionnent différemment: normalement opaque au repos, le cristal liquide devient transparent lorsqu'il est excité; pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière, comme c'est le cas pour les modèles rétroéclairés.

## Brochage

Un circuit intégré spécialisé est chargé de la gestion du module. Il remplit une double fonction : d'une part il commande l'affichage et de l'autre se charge de la communication avec l'extérieur.

Broche	Nom	Niveau	Fonction
1	Vss	-	Masse
2	Vdd	-	Alimentation positive +5V
3	Vo	0-5V	Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.
			Selection du registre (Register Select) Grâce à cette broche, l'afficheur est capable de faire la



4	RS	TTL	différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée.
5	R/W	TTL	Lecture ou écriture (Read/Write) L : Écriture H : Lecture
6	E	TTL	Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.
7	D0	TTL	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
8	D1	TTL	
9	D2	TTL	
10	D3	TTL	
11	D4	TTL	
12	D5	TTL	
13	D6	TTL	
14	D7	TTL	
15	A	-	Anode rétroéclairage (+5V)
16	K	-	Cathode rétroéclairage (masse)

Les broches 15 et 16 ne sont présentes que sur les afficheurs LCD avec rétroéclairage.

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), et connecter les broches **E**, **R/W** et **RS**.

## La mémoire

L'afficheur possède deux type de mémoire, la DD RAM et la CG RAM. La DD RAM est la mémoire d'affichage et la CG RAM est la mémoire du générateur de caractères.

### La mémoire d'affichage (DD RAM)

La DD RAM est la mémoire qui stocke les caractères actuellement affiché à l'écran. Pour un afficheur de 2 lignes de 16 caractères, les adresses sont définies de la façon suivante :

Ligne	Visible	Invisible
Haut	00H.....0FH	10H.....27H
Bas	40H.....4FH	50H.....67H

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

### La mémoire du générateur de caractères (CG RAM)

Le générateur de caractère est quelque chose de très utile. Il permet la création d'un maximum de 8 caractères ou symboles 5x7. Une fois les nouveaux caractères chargés en mémoire, il est possible d'y accéder comme s'il s'agissait de caractères classiques stockés en ROM.

La CG RAM utilise des mots de 8 bits de large, mais seul les 5 bits de poids faible apparaissent sur le LCD. Ainsi D4 représente le point le plus à gauche et D0 le point le plus à droite. Par exemple, charger un octet de la CG RAM à 1Fh fait apparaître tous les points de cette rangée ; charger un octet à 00h éteint tous ces points. Les 8 lignes d'un caractère doivent être chargées dans la CG RAM.

La CG RAM peut être utilisée pour créer des caractères en vidéo inversée, des caractères avec des accents, etc. La limitation d'un total de 8 caractères peut être contournée en utilisant une bibliothèque de 8 symboles résidant dans le système hôte. Un maximum de 8 caractères peut être affiché à la fois.

La CG RAM peut être rechargée périodiquement en fonction des besoins. Si un caractère de la CG RAM qui est actuellement sur l'afficheur est changé, alors le changement est immédiatement apparent sur l'afficheur.

## Commande d'un afficheur LCD

Deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur (voir plus bas).

### Mode 8 bits :

Dans ce mode 8 bits, les données sont envoyées à l'afficheur sur les broches **D0** à **D7**. On place la ligne **RS** à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée. Il faut aussi placer la ligne **R/W** à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture. Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée **E**, pour indiquer que des données valides sont présentes sur les broches **D0** à **D7**. L'afficheur lira la donnée sur le front descendant de cette entrée.

Si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne **R/W** à 1 pour demander une lecture. Les données seront valide sur les lignes D0 à D7 lors de l'état haut de la ligne **E**.

### Mode 4 bits :

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (**D4** à **D7**) de l'afficheur sont utilisées pour transmettre les données et les lire. Les 4 bits de poids faible (**D0** à **D3**) sont alors connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatres bits de poids fort suivi des quatres bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou nibble.

Dans les deux modes, on peut, après chaque action sur l'afficheur, vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela, il faut demander une lecture en mode commande, et tester le flag Busy **BF**. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée.

Il se peut qu'on dispose encore de moins de broches disponibles dans l'application envisagée. Dans ce cas, on peut alors relier la ligne **R/W** à la masse de façon à forcer l'afficheur en écriture. On a alors besoin, hors alimentation de seulement six fils en mode 4 bits, et dix fils en mode 8 bits, pour commander l'afficheur, mais on ne peut alors plus relire l'afficheur. Ceci n'est pas gênant dans la mesure où on sait ce qu'on a écrit sur l'afficheur, mais on ne peut alors plus relire le flag Busy. Il faut alors utiliser des temporisations après chaque écriture sur l'afficheur. On perd alors un peu en temps d'affichage, mais on gagne une broche d'entrée sortie.

## Le jeu de commandes standard

Instructions	Code										Description	Durée
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Efface l'ensemble de la mémoire de donnée sans toucher au générateur de caractères. Ramène le curseur en position « home », à l'adresse 00.	1,64 ms
Return home	0	0	0	0	0	0	0	0	1	X	Ramène le curseur en position « home », à l'adresse 00. Si l'affichage était décalé, il est remis à sa position d'origine : l'adresse 00 se trouve à nouveau en haut à gauche.	1,64 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Définit le sens de déplacement du curseur après l'apparition d'un caractère (vers la gauche si I/D=1, vers la droite si I/D=0) et si l'affichage accompagne le curseur dans son déplacement ou non (S).	40 µs
Display on/off control	0	0	0	0	0	0	1	D	C	B	Met l'affichage en ou hors fonction l'affichage (D), le curseur (C), le clignotement du curseur (B).	40 µs
Cursor and display shift	0	0	0	0	0	1	S/C	R/L	X	X	Deplace le curseur (S/C=1) ou l'affichage (S/C=0) d'une position vers la gauche (R/L=1) ou la droite (R/L=0) sans changer la DD RAM.	40 µs
Function set	0	0	0	0	1	DL	N	F	X	X	Définit la taille de l'interface (DL=0 pour mode 4 bits, DL=1 pour mode 8 bits), le nombre de lignes (NL=0 pour 1 ligne, N=1 pour 2 ou 4 lignes), et la taille des	40 µs

											fontes (F=0 pour des caractères 5x7, F=1 pour des caractères 5x10).	
Set CG RAM address	0	0	0	1	A5	A4	A3	A2	A1	A0	Définit l'adresse de la CG RAM. Les données de la CG RAM sont envoyées après cette commande.	40 µs
Set DD RAM address	0	0	1	A6	A5	A4	A3	A2	A1	A0	Définit l'adresse de la DD RAM. Les données de la DD RAM sont envoyées après cette commande.	40 µs
Read busy flag & address	0	1	BF	A6	A5	A4	A3	A2	A1	A0	Lit le flag busy (BF), et l'adresse de la position du curseur. BF vaut 0 si l'afficheur accepte une instruction, 1 s'il est occupé	1 µs
Write data to CG or DD RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Ecrit des données dans la DD RAM ou la CG RAM.	40 µs
Read data	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Lit les données de la DD RAM ou de la CG RAM.	40 µs

Après avoir défini le sens de déplacement, les caractères apparaissent au dessus du curseur (qu'il soit visualisé ou non). On peut déplacer le curseur en utilisant la commande « **Set DD RAM address** ».

Un défilement des caractères est aussi possible, il suffit pour cela d'envoyer la commande 07H (I/D=1, S=1), sans oublier de positionner **RS** à 0. Mais il faut faire attention car un décalage de l'affichage décale également les adresses de l'afficheur de sorte que l'extrême gauche de la ligne supérieure ne soit plus adressable à l'adresse 00. La commande « **Return home** » permet de repositionner les adresses de l'affichage.

## Le jeu de caractères standard

Les afficheurs utilisent tous le même type de contrôleur qui permet d'afficher l'ensemble des caractères ASCII, ainsi que quelques symboles et un jeu de caractères japonais stylisé. Les caractères accentués français ne font pas parti du jeu de ces contrôleurs. Fort heureusement, on peut tout de même afficher tous les caractères désirés grâce à des caractères définissables par l'utilisateur. Ceux-ci sont au nombre de huit et correspondent aux caractères de 0 à 7.

L'afficheur est en mesure d'afficher 200 caractères :

- de 00H à 07H : 8 caractères définissables par l'utilisateur
- de 20H à 7FH : 96 caractères ASCII excepté le caractère \ qui est remplacé par le signe ¥
- de A0H à DFH : 64 caractères japonais (alphabet kana)
- de E0H à FFH : 32 caractères spéciaux (accent, lettres grecques, ...)

Higher 4bit Lower 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000		0	1	2	3	4	5	6	7	8	9	A	B
xxxx0001		C	D	E	F	G	H	I	J	K	L	M	N
xxxx0010		O	P	Q	R	S	T	U	V	W	X	Y	Z
xxxx0011		[	\	^	_	`	a	b	c	d	e	f	g
xxxx0100		h	i	j	k	l	m	n	o	p	q	r	s
xxxx0101		t	u	v	w	x	y	z	{		}	~	
xxxx0110													
xxxx0111													
xxxx1000													
xxxx1001													
xxxx1010													
xxxx1011													
xxxx1100													
xxxx1101													
xxxx1110													
xxxx1111													

## Initialisation

### En mode 8 bits

À la mise sous tension de l'afficheur, la ligne supérieur devrait être totalement sombre, celle du bas complètement claire. Si tel n'était pas le cas, il faudra régler le contraste de l'afficheur en jouant sur la tension de la broche **V<sub>o</sub>**.

Avant de pouvoir utiliser l'afficheur, il faut tout d'abord l'initialiser. Pour cela, la première commande à envoyer, est la commande permettant de définir le mode de dialogue avec l'afficheur (**DL**), et le nombre de ligne sélectionnées (**N**), et l'envoyer plusieurs fois de façon à ce que celle ci soit comprise, que le mode de départ soit quatre ou huit bits. On peut ensuite paramétrer l'afficheur, puis l'effacer.

Voici donc les différentes commandes (RS=0) à envoyer à l'afficheur LCD. Entre chaque valeur, il faut envoyer une impulsion d'au moins 450 ns sur la ligne E :

- **33h, 33h, 33h** : on force le LCD en mode 8 bits (3xh)
- **38h** : mode 8 bits, 2 lignes, caractères 5x7
- **0Ch** : affichage en fonction, pas de curseur
- **06h** : le curseur se déplace vers la gauche
- **01h** : on efface l'afficheur

## En mode 4 bits

Pour l'initialisation d'un afficheur en mode quatre bits, on commence par forcer celui-ci dans le mode huit bits, puis quand on est sûr que celui-ci est valide, on bascule en mode quatre bits. Comme on ne sait pas au début de l'initialisation si l'afficheur est positionné en quatre ou huit bits, il est nécessaire d'envoyer la commande de passage en mode huit bits plusieurs fois de façon à ce que celle-ci soit comprise, que le mode de départ soit quatre ou huit bits. Les données sont écrites ou lues en envoyant séquentiellement les quatres bits de poids fort suivi des quatres bits de poids faible, séparés par une impulsion positive d'au moins 450 ns sur la ligne E.

En résumé, voici sur quatre bits, les commandes (RS=0) à envoyer à l'afficheur LCD. Entre chaque valeur, il faut envoyer une impulsion positive sur la ligne E.

- **0h, 1h, 0h, 0h, 1h** : on commence par effacer l'afficheur (01h)
- **3h, 3h, 3h** : on force le LCD en mode 8 bits (3xh)
- **2h** : on passe en mode 4 bits (20h)
- **2h, 8h** : mode 4 bits, 2 lignes, caractères 5x7 (28h)
- **0h, Ch** : affichage en fonction, pas de curseur (0Ch)
- **0h, 6h** : le curseur se déplace vers la gauche (06h)
- **0h, 1h** : on efface l'afficheur