

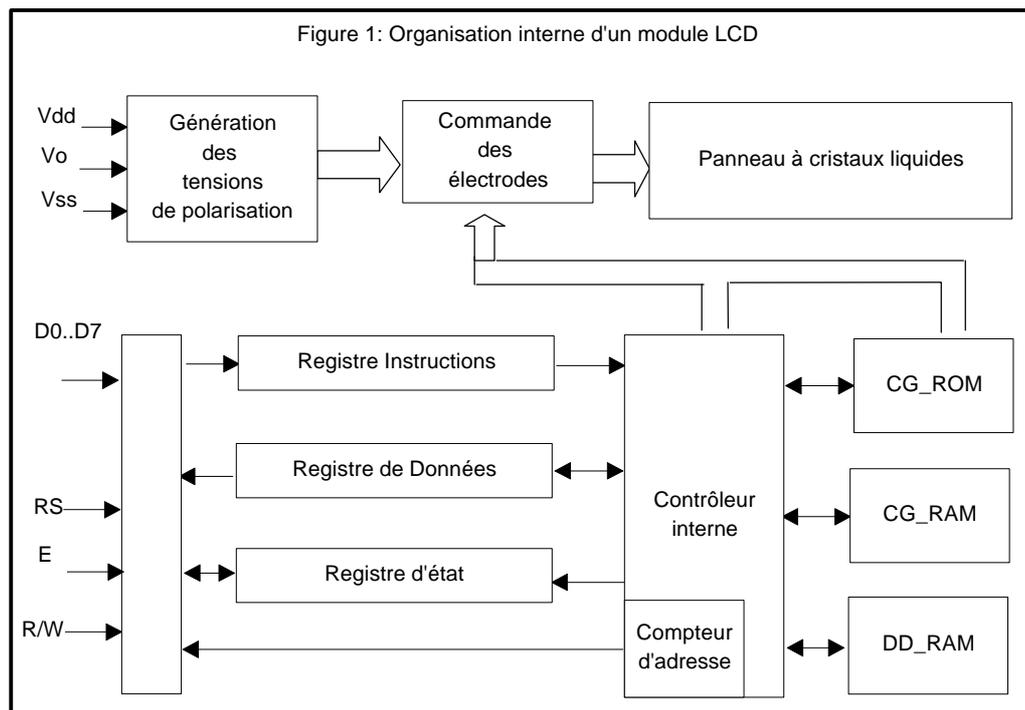
## Mise en œuvre de l'afficheur LCD alphanumérique

Cette afficheur assure simultanément l'affichage de textes et de données numériques, pour une consommation de quelques 10 mW. Cependant, son jeu d'instructions très complet le destine à être piloté par un dispositif microprogrammé.

### Organisation interne d'un module LCD :

L'organisation fonctionnelle du module est indiquée en figure 1. Un port d'entrée/sortie est doté de huit bits de données (D0..D7), associé à trois bits de contrôle (RS, E et R/W).

L'utilisateur a accès directement à trois registres depuis ce port :



- un registre de données, bidirectionnel, qui reçoit les codes ASCII des caractères à afficher. Les codes seront ensuite stockés dans la DD-RAM, d'une capacité de 80 octets. Le registre de données permet aussi de lire le contenu de la DD-RAM.
- Un registre de contrôle, destiné à recevoir les consignes de contrôle, comme effacer l'afficheur, déplacer le curseur, etc.. (accessible uniquement en écriture).
- Un registre d'état qui est destiné à indiquer à l'utilisateur si le processus interne est prêt à recevoir une instruction (accès en lecture seule).

La CG-ROM est une ROM génératrice de caractères qui fournit 192 motifs de caractères différents en matrices de 5 x 7 points. La relation entre le code à transmettre et le motif du caractère est donnée par le tableau 1. La DD-RAM reçoit les codes des caractères à afficher. L'adresse à laquelle est placé un code dans la DD-RAM définit la position du caractère sur le panneau d'affichage. Le pointage d'un élément dans la DD-RAM ou la CG-RAM est déterminé par un compteur d'adresses interne, accessible en lecture par l'utilisateur .

La CG- RAM est une RAM génératrice de caractères grâce à laquelle l'utilisateur peut définir jusqu'à 8 motifs de caractères personnalisés.

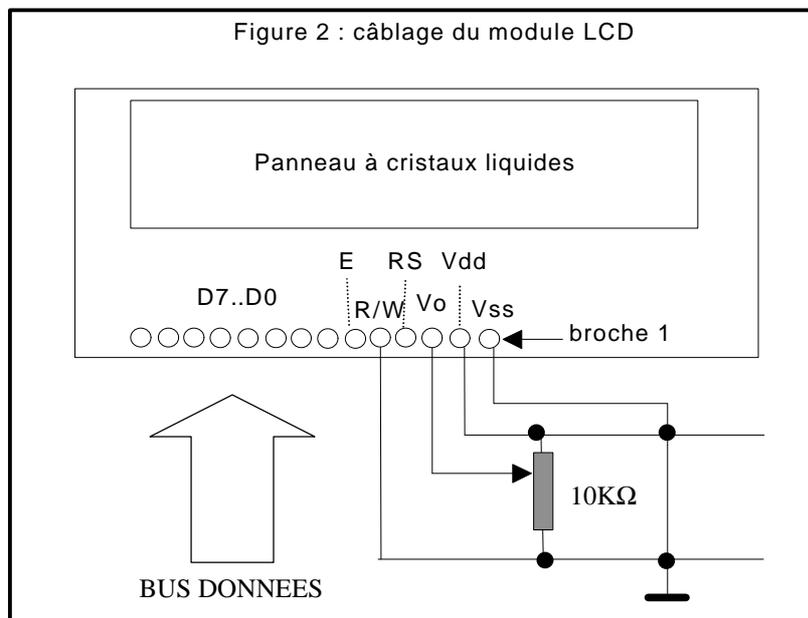
Les adresses de la seconde ligne ne sont pas consécutives à celles de la première ligne de caractères. Quand l’affichage est décalé, les adresses se décalent uniquement vers la droite ou vers la gauche d’une même ligne. Les adresses assignées à chacune des lignes sont les suivantes :

- Ligne n°1 : \$00 à \$27
- Ligne n°2 : \$40 à \$67

Signalons également que le module d’affichage LCD peut fonctionner à partir d’un microprocesseur 4 bits ou 8 bits.

### **Brochage du module :**

La figure 2 montre le câblage du module LCD. Le module dispose de 3 bornes d’alimentation : Vdd, Vss et Vo. Une tension de contrôle du contraste comprise entre 0 V et 5V doit être appliquée sur la broche Vo, par l’intermédiaire d’un potentiomètre de 10 K $\Omega$  (le meilleur réglage sera obtenu entre 0 V et 1 V). Vdd est reliée à l’alimentation de +5 V, et Vss à la masse.



### **Pilotage du module LCD :**

A priori, l’affichage d’un caractère sur le module sera effectué en deux temps : il faut d’abord envoyer une information de positionnement du curseur dans le registre d’instruction, puis le code ASCII du caractère dans le registre de données. Le module affiche alors le motif du caractère à l’emplacement indiqué. En fait, selon le mode d’affichage initialisé par l’utilisateur, le processeur du module peut assurer automatiquement le déplacement du curseur vers la droite ou vers la gauche. Dans ce cas, il suffit d’entrer une succession de codes ASCII pour obtenir l’affichage d’une suite de caractères.

La ligne de contrôle RS permet de préciser au module si la donnée qu’on lui envoie est une instruction (RS = 0) ou un caractère (RS = 1). L’entrée E active sur le front descendant, permet de valider la commande.

### Tableau 1

This character generator ROM (CG ROM) generates a character pattern of 5 x 7 dots from 8-bit character code and provides 192 character patterns.

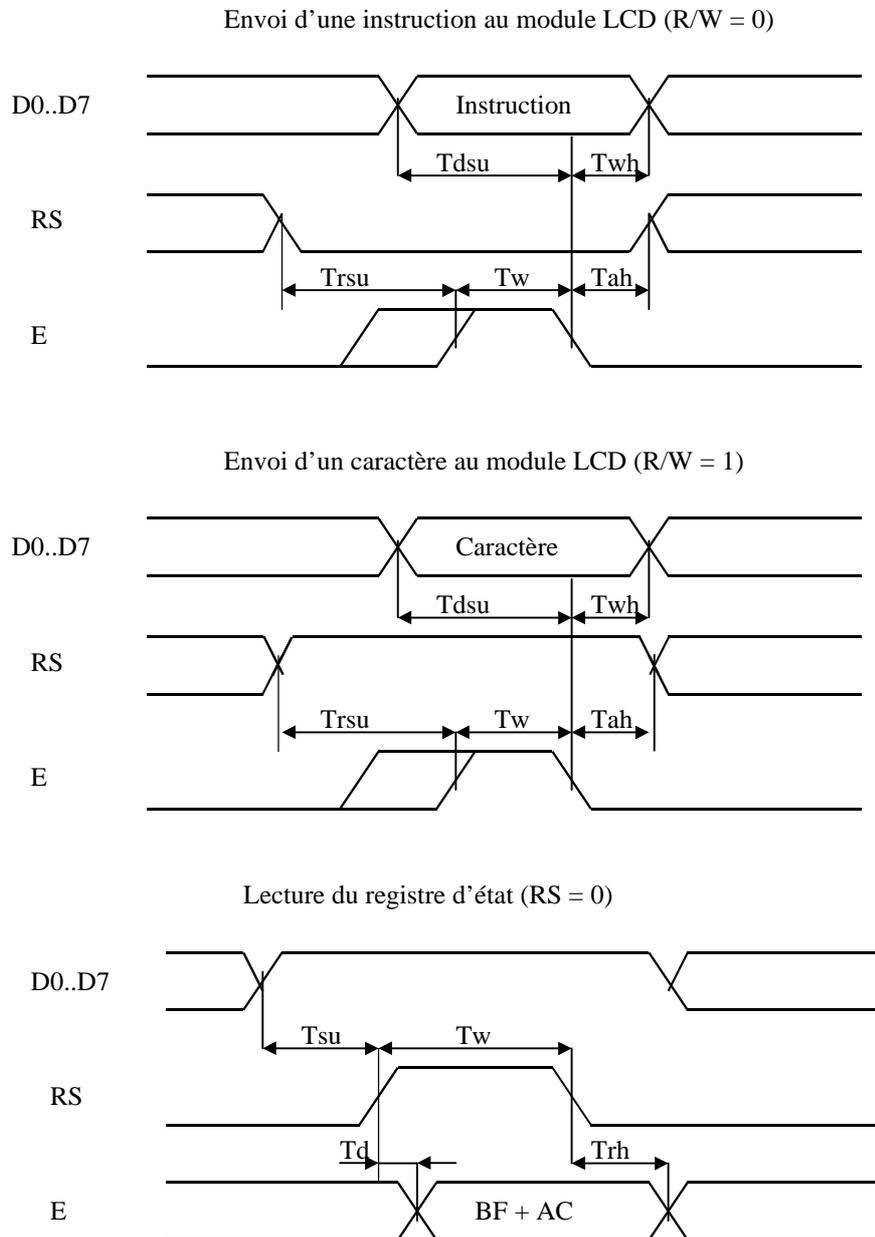
Correspondence between Character Codes and Character Pattern

High Lower 4bit 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	a	P	`	P	-	9	ε	α	ρ	
xxxx0001	(2)	!	1	A	Q	a	q	h	7	7	△	△	q
xxxx0010	(3)	"	2	B	R	b	r	Γ	イ	ツ	×	β	θ
xxxx0011	(4)	#	3	C	S	c	s	∟	ウ	テ	モ	ε	ε
xxxx0100	(5)	\$	4	D	T	d	t	、	エ	ト	ト	μ	α
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	∟	ε	U
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	*	7	G	W	g	w	ア	キ	ヌ	ラ	g	π
xxxx1000	(1)	(	8	H	X	h	x	イ	ウ	ホ	リ	γ	×
xxxx1001	(2)	)	9	I	Y	i	y	ウ	ケ	ル		γ	γ
xxxx1010	(3)	*	:	J	Z	j	z	エ	コ	ン	ク	j	〒
xxxx1011	(4)	+	:	K	[	k	[	オ	サ	ヒ	ロ	%	π
xxxx1100	(5)	,	<	L	∞	l	∞	ホ	シ	フ	ワ	φ	π
xxxx1101	(6)	-	=	M	∩	m	∩	ユ	ズ	ン	ク	ε	÷
xxxx1110	(7)	.	>	N	^	n	^	ヨ	セ	ホ	ン	∩	
xxxx1111	(8)	/	?	O	_	o	_	ホ	ッ	ソ	マ	o	■

\* The user can specify any pattern for character-generator ROM.

Les chronogrammes de la figure 3 indiquent les signaux à envoyer au module dans les deux cas. Le troisième chronogramme correspond à une séquence de lecture du registre d'état (BF) et du compteur d'adresses (AC).

**Figure 3 : Chronogrammes de commande**



Le bit d'état passe à 1 lorsque le module est en cours d'exécution d'une opération interne. Pendant cette période, l'afficheur est incapable de traiter une autre instruction. L'utilisateur peut alors consulter le registre d'état, afin d'envoyer une nouvelle instruction au module dès que le bit D7 revient à 0. Une autre approche consiste à déclencher une temporisation de durée appropriée après chaque sollicitation de l'afficheur. Cette procédure permet d'utiliser le module uniquement en écriture. On remarquera que le contenu du registre d'état est présent sur le bus de données tant que E et R/W sont à l'état haut.

Le tableau 2 indique le timing à respecter pour que les échanges entre le microcontrôleur et le module LCD aient lieu dans des bonnes conditions.

**Tableau 2 : Timing des chronogrammes**

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT
Enable cycle time	Tcyc	1000	-	-	ns
Enable pulse width	Tw	450	-	-	ns
Rise time	Tr	-	-	25	ns
Fall time	Tf	-	-	25	
Register select set-up time	Trsu	140	-	-	ns
Read and write set-up time	Tsu	140	-	-	ns
Data set-up time	Tdsu	195	-	-	ns
Data delay time	Td	-	-	320	ns
Address hold time	Tah	10	-	-	ns
Data hold time write	Twh	10	-	-	ns
Data hold time read	trh	20	-	-	ns

**Jeu d'instructions du module LCD :**

La liste des codes est indiquée dans le tableau 3.

→**Effacer l'affichage (Display clear) :** La DDRAM est remplie avec le code \$20 (Espace). Le compteur d'adresses est remis à zéro. Si l'affichage a été décalé, la position initiale est rétablie. Après l'exécution de cette instruction, l'affichage disparaît et le curseur est ramené à la position située en haut à gauche.

→**Initialiser le curseur (Cursor hom) :** Le compteur d'adresses est remis à zéro. Si l'affichage a été décalé, la position initiale est rétablie. Le contenu de la DD\_RAM n'est pas modifié. Le curseur ou le clignotement du caractère, si activé, est ramené à sa position initiale.

**Tableau 3 : Jeu d'instruction du module LCD**

Instructions	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Time
Display Clear	0	0	0	0	0	0	0	0	0	1	1.64ms
Cursor Home	0	0	0	0	0	0	0	0	1	•	1.64ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	40µs
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	40µs
Curs/Displ Shift	0	0	0	0	0	1	S/C	R/L	•	•	40µs
Function Set	0	0	0	0	1	DL	N	0	•	•	40µs
CG_RAM adress set	0	0	0	1	A-CG					40µs	
DD_RAM adress set	0	0	1	A-DD					40µs		
Busy/adress read	0	1	BF	AC					0		
CG/DD_RAM write	1	0	Write DATA					40µs			
CG/DD_RAM read	1	1	Read DATA					40µs			

→**Initialiser le mode d'entrée des données (Entry mode set) :**

- **I/D (Inc/Dec) :** Lorsqu'on envoie un code de caractère à afficher, le curseur se déplace d'une position d'affichage vers la droite ( $ID = 1$ ) ou la gauche ( $ID = 0$ ). Ce fonctionnement est valable également quand une donnée est lue ou écrite dans la CG\_RAM.

- **S (Shift) :** Quand  $S = 1$ , l'ensemble de l'affichage est décalé vers la gauche ( $ID = 1$ ) ou la droite ( $ID = 0$ ) après écriture d'un code de caractère dans la DD\_RAM. Le curseur garde la même position relative sur l'afficheur. Si  $S = 0$ , l'affichage n'est pas décalé.

- **Remarque :** Quand un décalage d'affichage est exécuté, les 2 lignes sont décalées simultanément. Quand l'opération de l'affichage est répétée, les caractères d'une ligne ne sont pas déplacés vers l'autre ligne mais sont rebouclés sur la même ligne.

→**Allumer/éteindre l'affichage (Display ON/OFF) :**

- **D (Display) :** l'affichage est allumé si  $D = 1$ . Quand  $D = 0$ , l'affichage est éteint, mais les codes de caractères sont conservés dans la DD\_RAM.

- **C (Cursor) :** le curseur est affiché à la position courante du compteur d'adresse si  $C = 1$ . Quand  $C = 0$ , le curseur n'est pas visible.

- **B (Blink) :** si  $B = 1$ , le caractère à la position du curseur clignote à 1.2 Hz environ.

→**Décaler le curseur/l'affichage (Cursor/display shift) :** L'affichage et/ou le curseur sont décalés vers la droite ou la gauche. Pour un afficheur à 2 lignes, le curseur se déplace de la 40<sup>ème</sup> position de la ligne du haut à la première position de la seconde ligne (adresse \$67), le curseur ne retourne pas à l'adresse \$00 mais à la première position de la seconde ligne (adresse \$40).

- **S/C (Shift/Cursor) :** déplacement du curseur si  $S/C = 0$ . Lorsque  $S/C = 1$ , c'est tout l'affichage, avec le curseur, qui est décalé.

- **R/L (Right/Left) :** décalage vers la gauche si  $R/L = 0$ , vers la droite si  $R/L = 1$ .

→**Initialiser le mode de fonctionnement (Function Set) :**

- **DL (Data Length)** sélectionne la largeur du bus de données : 8 bits ( $DL = 1$ ) ou 4 bits ( $DL = 0$ ).

- **N (Number of lines)** initialise l'affichage sur une ligne si  $N = 0$ , et deux lignes si  $N = 1$ .

- **Remarque :** cette instruction doit être exécutée au début du programme du microprocesseur, avant toute autre instruction (à l'exception de l'instruction de lecture du bit d'état et du compteur d'adresses).

→**Positionner l'adresse de la CG\_RAM (CG\_RAM adresse set) :** cette instruction permet au compteur d'adresses de pointer une adresse dans la CG\_RAM. Un transfert de données pourra ensuite être réalisé entre le micro et la CG\_RAM.

→**Positionner l'adresse de la DD\_RAM (DD\_RAM adresse set) :** cette instruction permet au compteur d'adresses de pointer une adresse dans la DD\_RAM. Un transfert de données pourra ensuite être réalisé entre le micro et la DD\_RAM.

→**Lire le registre d'état et le compteur d'adresses (Busy flag/address read) :** Le bit BF ne peut être que lu et indique si le module est occupé par le traitement d'une instruction.  $BF = 1$  indique l'état occupé (opération interne) et une nouvelle instruction ne sera pas acceptée tant que BF ne sera pas égal à 0. Cette instruction lit aussi le contenu du compteur d'adresses, exprimé par un nombre binaire de 7 bits. Lors d'une lecture, le compteur d'adresses va

contenir une adresse de la DD\_RAM ou de la CG\_RAM selon la dernière instruction d'adressage exécutée.

→**Ecrire une donnée dans la CG\_RAM/DD\_RAM (CG\_RAM/DD\_RAM data write) :**

Une donnée de 8 bits est écrite dans la CG\_RAM ou la DD\_RAM selon l'adresse spécifiée par le compteur d'adresses. Après écriture de la donnée dans la RAM, le compteur d'adresses est incrémenté ou décrémenté selon le mode d'entrée prédéfini.

→**Lire une donnée dans la CG\_RAM/DD\_RAM (CG\_RAM/DD\_RAM data read) :**

Une donnée de 8 bits est lue dans la CG\_RAM ou la DD\_RAM selon la dernière instruction d'adressage exécutée, à l'emplacement indiqué par le compteur d'adresses. Cette instruction doit être précédée par une instruction de positionnement d'adresse dans l'une des deux RAM, ou une instruction de décalage du curseur.

→**Initialisation du module à la mise sous tension :** A la mise sous tension, le module exécute une procédure d'initialisation automatique qui assure les fonctions suivantes :

- Effacement de la DD\_RAM,
- Extinction de l'afficheur,
- Format des données : 8 bits,
- Affichage sur une seule ligne,
- Curseur fixe,
- Déplacement du curseur vers la droite à chaque écriture du caractère.

Pendant cette phase d'initialisation, qui dure environ 15 ms, le bit d'état est mis à l'état haut.

Il est toutefois préférable d'assurer l'initialisation du module par programmation avant d'envoyer les premiers caractères au module.