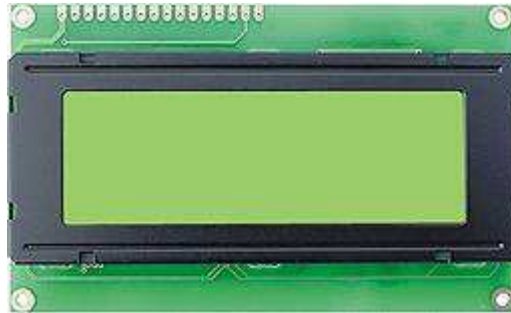


L'afficheur LCD (Light Control Display)

I Rôle d'un afficheur LCD :

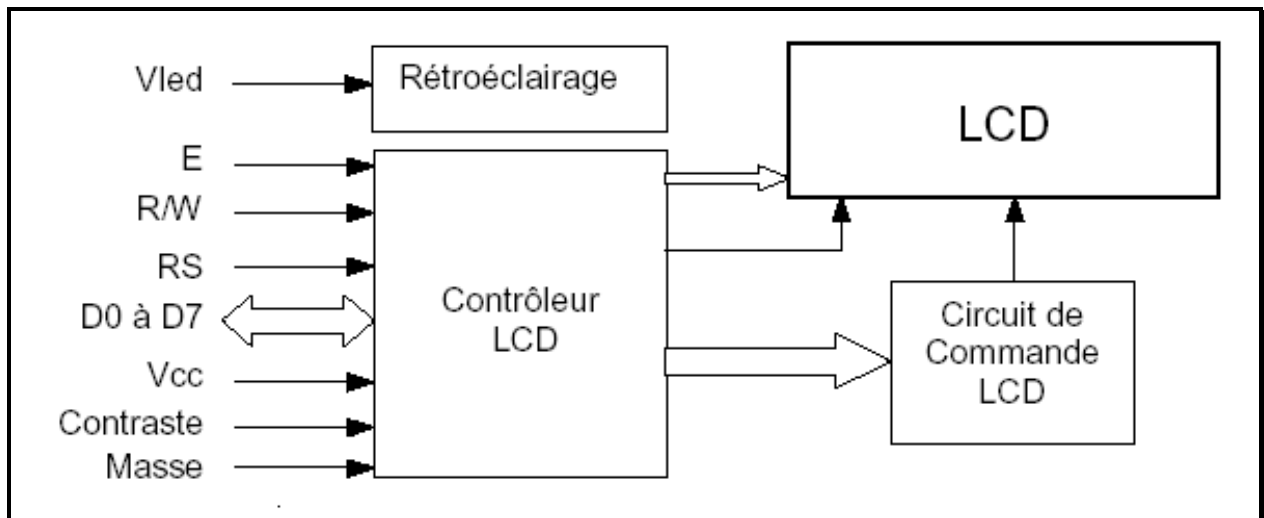
Les **afficheurs LCD** sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage de paramètres de fonctionnement.

Grâce à la commande par un microcontrôleur ces afficheurs permettent de réaliser un affichage de messages aisés. Ils permettent également de créer ses propres caractères.



II Fonctionnement d'un afficheur LCD :

II.1 Schéma fonctionnel :



II.2 Rôle des différentes broches de l'afficheur LCD :

V_{CC}, Masse : alimentation de l'afficheur LCD. Un afficheur LCD s'alimente en **0V-5V**.

Contraste : entrée permettant de régler le contraste de l'afficheur LCD. Il faut appliquer une tension continue réglable (entre **0V** et **5V**) à l'aide d'un potentiomètre.

Vled : différence de potentiel permettant de commander le rétro éclairage.

E : entrée de validation (**ENABLE**), elle permet de valider les données sur un **front descendant**. Lorsque **E=0** alors le bus de données est à l'état haute impédance.

RS : Register Select cette entrée permet d'indiquer à l'afficheur si l'on souhaite réaliser une commande (**RS=0**) par des instructions spécifiques ou écrire une donnée (envoi du code du caractère à afficher) sur le bus (**RS=1**).

R/W : entrée de lecture (**R/W=1**) et d'écriture (**R/W=0**). Lorsqu'on commande l'afficheur LCD il faut se placer en écriture.

D7...D0 : **bus de données bi-directionnel**, il permet de transférer les instructions ou les données à l'afficheur LCD.

L'afficheur LCD (Light Control Display)

II.3 La mise en œuvre d'un afficheur LCD :

Un afficheur LCD sera commandé par un microcontrôleur.

Il faut donc penser aux mises en œuvre :

- **matérielle** : connexion des broches du microcontrôleur à l'afficheur LCD,
- **logicielle** : utilisation de sous programmes permettant de commander l'afficheur LCD (initialisation, effacement de l'afficheur, affichage d'un caractère, affichage d'une variable,...).

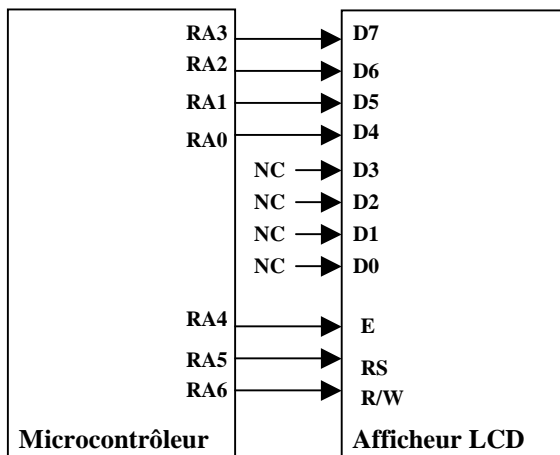
II.3.1 Mise en œuvre matérielle d'un afficheur LCD :

En fonction du mode de commande choisis de l'afficheur LCD la mise en œuvre matérielle sera différente.

Il existe deux modes de commande d'un afficheur LCD :

- commande en **4 bits**,
- commande en **8 bits**.

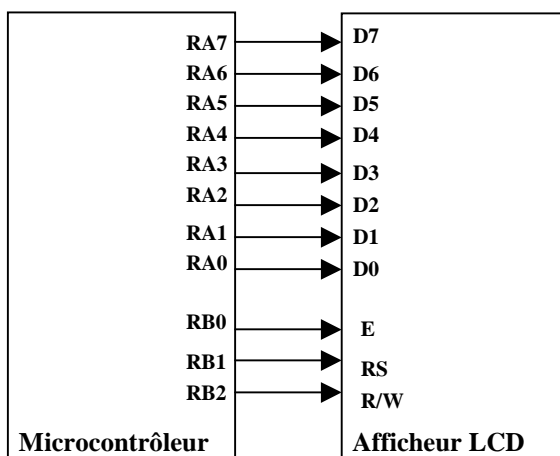
Mise en œuvre matérielle pour la commande en 4 bits :



En mode de commande 4 bits l'intérêt est de limiter le nombre de broches du microcontrôleur.

Il faut utiliser 7 broches des ports d'entrées/sorties du microcontrôleur (configurées en sorties) de manière à commander l'afficheur.

Mise en œuvre matérielle pour la commande en 8 bits :

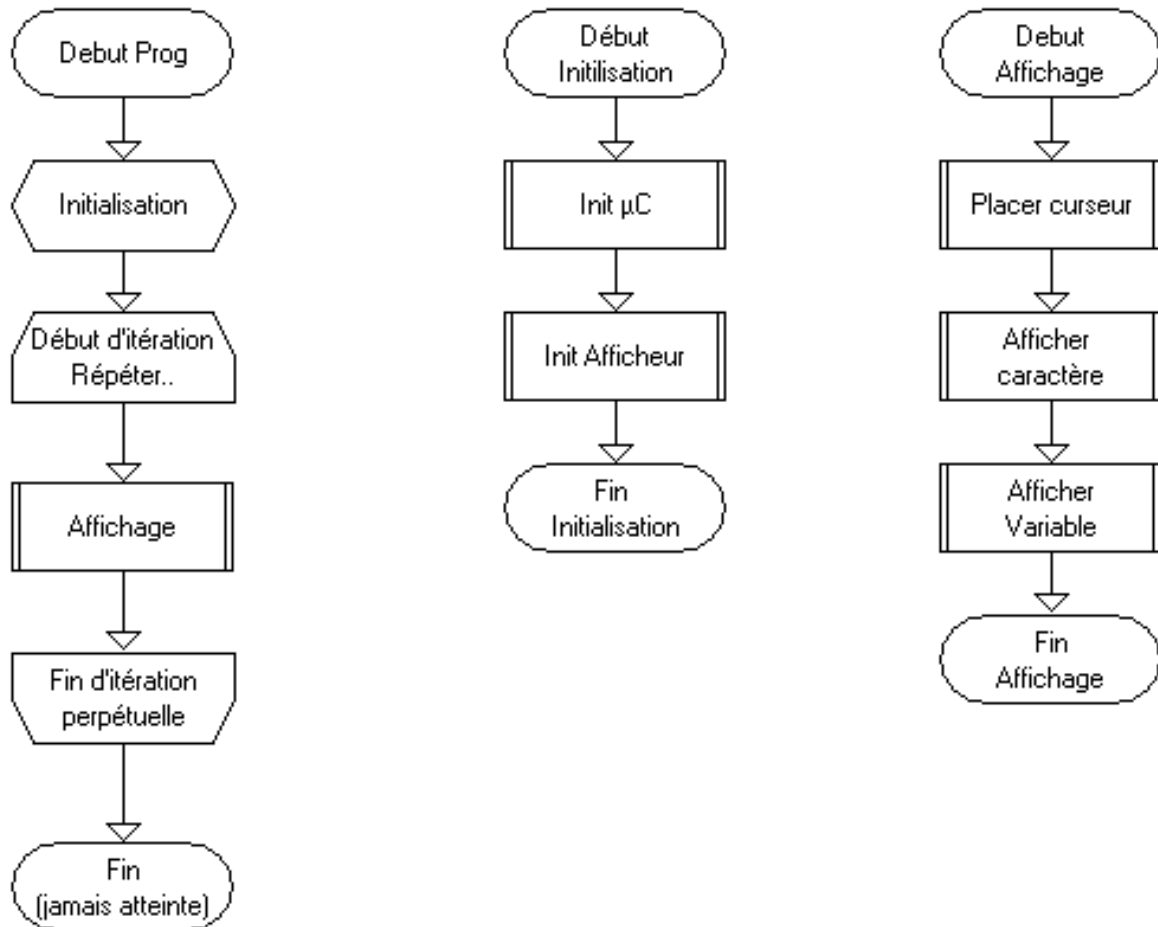


En mode de commande 8 bits on utilise plus de broches du microcontrôleur.

Il faut utiliser 11 broches des ports d'entrées/sorties du microcontrôleur (configurées en sorties) de manière à commander l'afficheur.

L'afficheur LCD (Light Control Display)

II.3.1 Mise en œuvre logicielle d'un afficheur LCD :



Les initialisations :

L'initialisation du microcontrôleur doit permettre de configurer les broches des ports d'entrées/sorties en sorties.

L'initialisation de l'afficheur permet :

- d'**attendre** la mise sous tension de l'afficheur, il faut réaliser une temporisation donnée par la documentation constructeur (10 à 20ms).
- de **préparer** l'initialisation en envoyant des instructions,
- de **configurer** le mode de commande de l'afficheur (8 bits ou 4 bits),
- de **sélectionner** le nombre de lignes et de **configurer** le format du caractère (5x7 ou 5x10 pixels),
- d'**allumer** l'afficheur,
- de **configurer** le curseur (déplacement à gauche ou à droite après affichage, clignotement),
- de **réaliser le reset** de l'affichage.

Les instructions permettant de réaliser ces configurations sont données par la documentation constructeur.

L'afficheur LCD (Light Control Display)

Placement du curseur :

Pour indiquer à quelle position doit se faire l'affichage il faut placer le curseur avant d'afficher.

Pour cela on utilisera un sous-programme qui prendra en paramètres la position où doit se faire l'affichage.

Ce sous programme indique l'adresse de la DDRAM interne à l'afficheur LCD correspondant à la position :

For 16×1 or 8×1 Display																
Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47
For 16×2 or 8×2 Display																
Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
For 16×4 Display																
Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DD RAM Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
For 20×2 Display																
Character	1	2	3	4	5	6	7	8	9	10	17	18	19	20
DD RAM Address	00	01	02	03	04	05	06	07	08	09	10	11	12	13
	40	41	42	43	44	45	46	47	48	49	50	51	52	53
For 20×4 Display																
Character	1	2	3	4	5	6	7	8	9	10	17	18	19	20
	00	01	02	03	04	05	06	07	08	09	10	11	12	13
DD RAM Address	40	41	42	43	44	45	46	47	48	49	50	51	52	53
	14	15	16	17	18	19	1A	1B	1C	1D	24	25	26	27
	54	55	56	57	58	59	5A	5B	5C	5D	64	65	66	67
For 40×2 Display																
Character	1	2	3	4	5	6	7	8	9	10	37	38	39	40
DD RAM Address	00	01	02	03	04	05	06	07	08	09	24	25	26	27
	40	41	42	43	44	45	46	47	48	49	64	65	66	67

Affichage d'un caractère :

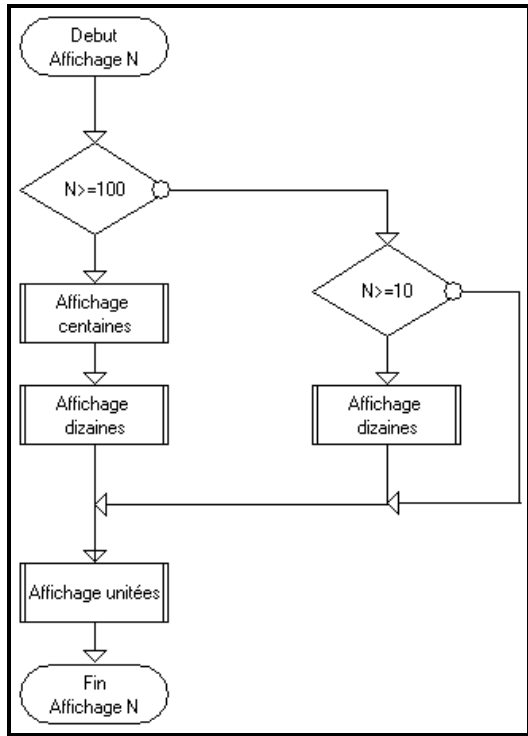
Le principe de fonctionnement est simple, pour visualiser un caractère, il suffit de le positionner sur le bus de donnée (codé en ASCII), de mettre RS au niveau haut (caractère), R/W au niveau bas (écriture), et de provoquer un front descendant sur l'entrée de validation de l'afficheur (E).

L'afficheur LCD (Light Control Display)

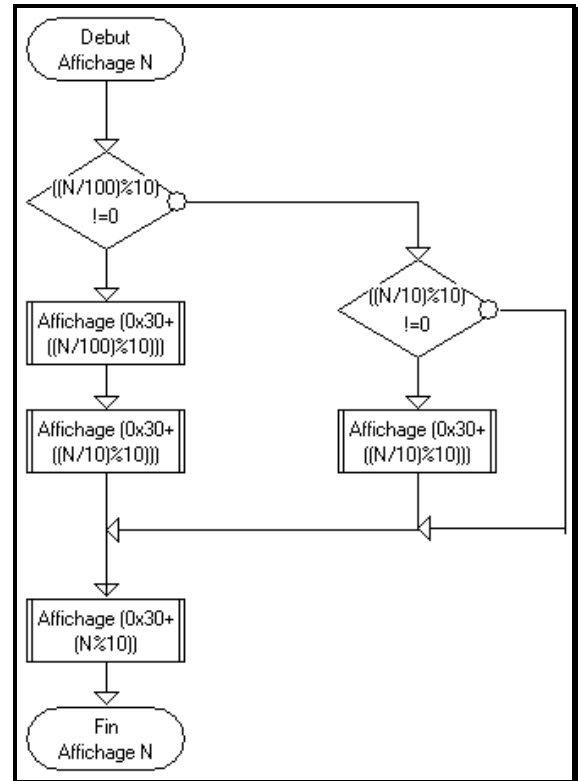
Affichage d'une variable :

Le principe utilisé est d'utiliser les résultats et les restes d'une division successive de la variable **N** par **100** puis par **10**.

Principe utilisé pour afficher un variable N :



Calculs réalisés :



L'afficheur LCD (Light Control Display)

Table de caractères :

	Lower 4 bit	Upper 4 bit	0	2	3	4	5	6	7	A	B	C	D	E	F
			0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
0	XXXX0000		CG RAM '1'		0	A	P	'	P		-	9	E	a	p
1	XXXX0001	'2'	!	1	A	0	a	9	.	7	7	4	ä	g	
2	XXXX0010	'3'	"	2	B	R	b	r	'	/	W	x	p	è	
3	XXXX0011	'4'	#	3	C	S	c	s	_	0	7	e	s	æ	
4	XXXX0100	'5'	\$	4	D	T	d	t	\	I	t	h	µ	o	
5	XXXX0101	'6'	%	5	E	U	e	u	.	7	7	1	o	ü	
6	XXXX0110	'7'	&	6	F	V	f	v	7	7	2	3	p	z	
7	XXXX0111	'8'	'	7	G	W	g	w	7	7	7	9	g	π	
8	XXXX1000	'1'	(8	H	X	h	x	4	7	7	7	7	7	
9	XXXX1001	'2')	9	I	Y	i	y	7	7	7	7	7	7	
A	XXXX1010	'3'	*	:	J	Z	j	z	7	7	7	7	7	7	
B	XXXX1011	'4'	+	;	K	C	k	c	(7	7	7	7	7	
C	XXXX1100	'5'	,	<	L	*	l	*	7	7	7	7	7	7	
D	XXXX1101	'6'	-	=	M	I	m	i)	7	7	7	7	7	
E	XXXX1110	'7'	.	>	N	^	n	^	7	7	7	7	7	7	
F	XXXX1111	'8'	/	?	O	_	o	_	7	7	7	7	7	7	