

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

**Projet d'étude et réalisation :**

**Affichage de grandeurs  
électriques**

Romain MOUCHEL-VALLON  
Ugo FIGARD  
2<sup>ème</sup> année - S2  
Promotion 2005/2007

Enseignants:  
Thierry LEQUEU  
Jacky BRUN

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

Projet d'étude et réalisation :  
**Affichage de grandeurs  
électriques**

Romain MOUCHEL-VALLON  
Ugo FIGARD  
2<sup>ème</sup> année - S2  
Promotion 2005/2007

Enseignants:  
Thierry LEQUEU  
Jacky BRUN

# Table des matières

Introduction	
1.Le micro contrôleur.....	5
1.1.Pourquoi un micro contrôleur ?.....	5
1.2.Quel micro contrôleur choisir?.....	5
1.3.La mise en oeuvre du picbasic-1S.....	8
2.Les afficheurs LCD.....	11
2.1.Qu'est ce qu'un afficheur LCD?.....	11
2.2.Le fonctionnement.....	12
2.3.Le brochage et composition.....	13
2.4.Le choix de notre afficheur LCD.....	16
Conclusion	
Index des illustrations.....	19

# **Introduction**

Dans le cadre du projet d'étude et réalisation de deuxième année, nous avons du choisir un projet. Après nous être concerté, nous nous sommes mis d'accord pour réaliser un module nous permettant d'afficher des grandeurs électriques à partir de capteurs, vers un afficheur L.C.D..

Nous allons développer notre dossier en commençant par une présentation du micro contrôleur que nous avons sélectionné, puis en décrivant l'afficheur et ses caractéristiques ainsi que son fonctionnement.

# 1. Le micro contrôleur

Dans cette partie nous allons voir, en premier lieu, la raison de notre choix d'un micro contrôleur et non un autre circuit intégré programmable, puis les différentes possibilités qui s'offraient à nous et enfin nous verrons notre choix final et sa mise en oeuvre.

## 1.1. Pourquoi un micro contrôleur ?

Le principal avantage des micro contrôleurs par rapport aux microprocesseurs est le fait que la plupart des circuits annexes, comme la mémoire ou autres, est intégrée au circuit. Il n'y a donc que peu de composant à implanter autour d'un micro contrôleur pour le faire fonctionner. Le tracé du typon est donc plus simple et le coût réduit. De plus, certains micro contrôleurs sont programmables "in situ", c'est à dire que l'on peut les programmer directement sur la carte de circuit imprimé par le biais d'un câble de liaison à l'ordinateur sans les déconnecter. Notre choix s'est donc rapidement porté vers l'un d'eux pour leur facilité de mise en oeuvre, leur encombrement réduit, ainsi que leur coût très abordable.

## 1.2. Quel micro contrôleur choisir?

Le fait de connaître le type de composant ne suffit évidemment pas à créer notre carte. En effet, il nous a fallut, ensuite, déterminer quel micro contrôleur utiliser en fonction de nos besoins. Face à la quantité de micro contrôleurs disponibles sur le marché, nous avons du faire un tri et seuls deux d'entre eux ont retenu notre attention.

### 1.2.1. Le C.P.L.D.

Le premier est le C.P.L.D., Complex Programmable Logique Device (système logique programmable complexe). Nous avons en premier lieu pensé à celui-ci car nous nous en sommes servi au cours de la première

année de notre cursus universitaire. Il n'était donc pas très difficile pour nous de nous remémorer nos anciens cours puis de le programmer et ensuite de le mettre en place sur une carte de circuit imprimé avec les composants nécessaires à son bon fonctionnement.

Il se programme, en effet, en vérilog, langage de programmation que nous avons étudié et utilisé dans de nombreux programmes pour le prendre en main. Ce composant ne devrait donc pas nous causer de soucis quant à sa programmation.

De plus ce composant présente des caractéristiques intéressantes pour notre projet. À savoir, de nombreuses entrées sorties : plus d'une vingtaine, ce qui est un net avantage pour les communications avec les périphériques.

Cependant, cela entraîne automatiquement un gros défaut qui est l'encombrement. En effet, ce composant possède 44 broches, il est donc relativement imposant et très difficile à mettre en place sur une carte de circuit imprimé. D'autant plus que notre projet nécessite des capteurs de tension et de courant. Et qui dit capteur et micro contrôleur dit forcément convertisseur analogique/numérique pour convertir la valeur transmise par les capteur en un mot binaire, codé sur un certain nombre de bits, utilisable et interprétable par le composant. Et, bien sûr, si l'on désire une bonne précision de conversion, il faut d'autant plus de bit en sortie du convertisseur, donc, d'autant plus d'entrées réquisitionnées sur le C.P.L.D., et donc d'autant plus de connexions et de pistes sur notre carte finale.

Or notre projet nécessite deux convertisseurs (tension et courant), ce qui correspond à 2 fois 8 entrées utilisées si l'on désire une précision moyenne de conversion pour les données.

En plus, ce C.P.L.D. ne fonctionne pas tout seul. En effet, il nécessite de nombreux composants autour de lui pour fonctionner, comme un quartz et de nombreux condensateurs et résistances.

Là, on commence à se rendre compte que ce formidable composant va peut-être coûter cher à mettre en place en plus de nécessiter des heures de travail pour réaliser le schéma, ainsi que le typon.

Nous avons finalement tenté de trouver un autre micro contrôleur plus pratique et plus économique.

### 1.2.2. Le picbasic

Après de longues recherches, nous avons fini par trouver le picbasic. C'est un micro contrôleur très pratique proposé à un prix très compétitif par rapport à ses énormes possibilités. En effet, il permet de développer très rapidement une multitude d'applications de la plus simple à la plus complexe. Cependant, c'est un composant qui se programme en basic qui est un langage de programmation que nous n'avons pas étudié au cours de notre cursus universitaire. Il va donc nous falloir apprendre à l'utiliser avant de tenter quoi que ce soit vis à vis de notre projet.

Le modèle que nous avons sélectionné est le plus petit et le moins cher de ceux proposés dans la même gamme, mais est néanmoins performant : c'est le picbasic-1S. Il possède évidemment des caractéristiques très intéressantes pour nous :

- 4 kO de mémoire EEPROM programme/données ainsi que 96 octets de mémoire RAM pour les variables, ce qui est relativement faible mais suffisant pour un programme pas trop conséquent;
- 16 entrées/sorties, beaucoup moins que le C.P.L.D.. Cependant, 5 d'entre elles peuvent être utilisées en entrées/sorties "standard" ou en conversion analogique/numérique 8 bits, ce qui nous sera très utile.

- Nombre de codes traités à la seconde : 1000, très faible devant un microprocesseur, mais amplement suffisant pour notre application.

Le fait d'avoir les convertisseurs analogique/numérique intégrés au composant va, effectivement, nous être d'une grande aide, car il nous épargne l'ajout des convertisseurs avec leurs nombreuses sorties, à notre carte. Il nous suffira donc de connecter la sortie de chaque capteur à une entrée de conversion et, par l'intermédiaire du programme grâce à la fonction correspondante, d'en récupérer la valeur numérique codée sur 8 bits (valeur comprise entre 0 et 255).

De plus, il possède une sortie, appelée "picbus", qui permet de communiquer par liaison série directement avec un afficheur L.C.D.. C'est donc, bien entendu, une caractéristique que nous ne risquons pas d'oublier en pensant à notre projet.

Il est livré sous la forme d'un module "S.I.L.", c'est à dire sur une carte avec toutes les sorties du même côté sur une même rangée, ce qui nous facilite grandement la tâche pour le tracé du circuit imprimé.

La décision de choisir ce micro contrôleur s'est donc avérée très facile à prendre étant donnée sa facilité de mise en oeuvre, son prix très abordable, et le schéma très léger qui résultera de son utilisation.

### 1.3. La mise en oeuvre du picbasic-1S

L'utilisation du pic ne sera pas des plus facile, notamment pour la programmation. En effet, ce langage de programmation étant nouveau pour nous, nous devons le découvrir entièrement avant de l'utiliser.

Nous devons ensuite chercher comment s'en servir pour afficher ce que l'on souhaite sur un afficheur L.C.D..



### 1.3.1. les instructions de base du basic

Ont été réunies dans cette partie, les instructions de base du langage basic qui pourront nous servir ainsi que leur signification.

**La déclaration des variables** : avant de pouvoir utiliser une variable pour y stocker des informations, il faut la déclarer au module "picbasic" pour qu'il lui réserve de la place au sein de sa mémoire RAM. Il existe deux types de variables : les **"BYTE"** correspondant à un nombre compris entre 0 et 255 et occupant 1 octet de mémoire ; ainsi que les **"INTEGER"** pour un nombre compris entre 0 et 65535 et occupant 2 octets de mémoire. La déclaration se faisant, en général, au début du programme à l'aide de l'instruction **"DIM"**.

**La déclaration de constantes** : une des caractéristique intéressante des picbasic réside dans la possibilité d'attribuer un nom à une constante. Cette déclaration se fait à l'aide de l'instruction **"CONST"**. Cette instruction est très utile notamment si l'on dispose, par exemple, d'une L.E.D. connectée sur la broche 2 du pic. Lors de la déclaration, on indiquera au pic que le terme équivaut à la valeur 2. Dès lors, lorsqu'on voudra allumer la L.E.D., il suffira de faire référence au mot "LED", ce qui améliorera la lisibilité et la compréhension du programme.

**Opération sur les variables et les constantes** : le basic permet d'effectuer toutes sortes d'opérations sur les variables et les constantes à l'aide des opérateurs habituels : +, -, \*, /, **"AND"** "et" logique, **"OR"** "ou" logique, **"XOR"** "ou exclusif" logique, **"MOD"** "Modulo", <, >, <=, >=, << et >> décalage à gauche et à droite.

**"ADIN"** : cette instruction permet de connaître la valeur de la tension analogique présente sur une des broches de conversion du pic. La valeur à lire doit impérativement être comprise entre 0 et 5V sous peine de destruction du port d'entrée. Il est nécessaire de placer, entre parenthèses, à la suite de l'instruction, le numéro de la broche qui reçoit la valeur à mesurer. Le nombre obtenu est directement proportionnel à la tension d'entrée et est compris entre 0 et 255.

On retrouve également, en basic, les fonctions habituelles du langage "C" que l'on utilise en informatique. On retrouve, entre autres : "IF...THEN...ENDIF", "FOR...NEXT", et "GOTO" saut inconditionnel.

### 1.3.2. Les instructions spécifiques aux afficheurs L.C.D.

Le picbasic permet aussi de disposer de toute une panoplie d'instructions pour piloter un afficheur L.C.D..

"LCDINIT", doit impérativement être exécuté au démarrage du programme pour piloter un afficheur L.C.D. via le port "picbus" afin que l'afficheur en question s'initialise correctement.

"CLS", cette instruction permet l'effacement total de l'écran.

"LOCATE" suivi des coordonnées séparées d'une virgule, positionnement du curseur à un endroit précis de l'afficheur.

"PRINT" suivi du texte entre guillemets, permet d'afficher des messages sur l'afficheur.

"PRINT DEC" suivi de la valeur suivie de 2 paramètres entre parenthèses et séparés d'une virgule, permet d'afficher la valeur décimal d'un nombre ou d'une variable sur l'afficheur. Le premier paramètre déterminant le nombre total de caractères affichés (1 à 5), et le deuxième indiquant si le nombre doit être précédé de '0' à la place des caractères non-affichés (0) ou d'espaces (1). Il est également possible d'effectuer des combinaisons afin d'afficher plusieurs types de données sur une même ligne.

"PRINT HEX", même fonctionnement que "PRINT DEC" mais affichant la valeur hexadécimale d'un nombre.

"CRSON" et "CRSOFF", ces fonctions permettent d'activer ou de désactiver l'affichage du curseur.

"**BUSOUT**", la broche série de chaque module picbasic est spécialement conçue pour piloter des afficheurs par le biais d'instructions spécifiques qui envoient une série d'ordres à ce dernier. L'utilisateur a également la possibilité de piloter ces afficheurs série en utilisant cette instruction.

"**SET PICBUS**" suivi de "**HIGH**" ou "**LOW**", cette instruction permet de paramétrer la vitesse de communication du bus spécialisé afin de l'adapter au type d'afficheur à liaison série utilisé.

On s'aperçoit que le pic basic est le mieux adapté à notre projet. En effet, il n'est pas encombrant, pas cher, et, de plus, possède un jeu d'instruction extrêmement complet pour l'utilisation d'un afficheur L.C.D. dans un langage de programmation très simple et facile d'accès. Il constituera donc notre choix définitif pour notre carte finale dont le schéma s'en trouvera très allégé par l'utilisation de ce pic.

## **2. Les afficheurs LCD**

Il existe de nombreuses sortes d'afficheurs L.C.D., c'est pourquoi il nous a fallu faire un tri et choisir celui qui s'est avéré le plus adapté à notre projet.

### **2.1. Qu'est ce qu'un afficheur LCD?**

Un afficheur LCD, Liquid Cristal Display (écran à cristaux liquides), est l'un des dispositifs électroniques les plus utilisés pour afficher ce que l'on veut. Ils sont peu encombrants et ne nécessitent que peu de composants externes pour un bon fonctionnement. De plus, leur utilisation est simple.

Sur le marché, on en trouve de plusieurs types, en effet, les afficheurs diffèrent de par leurs dimensions et leurs caractéristiques électriques. Ainsi on peut trouver des afficheurs qui affichent sur 1, 2, 3, ou 4 lignes avec un nombre de caractères allant de 6 à 80 par lignes.



*Illustration 1 : Trois types d'afficheurs*

On trouve aujourd'hui trois principaux types d'afficheurs LCD:

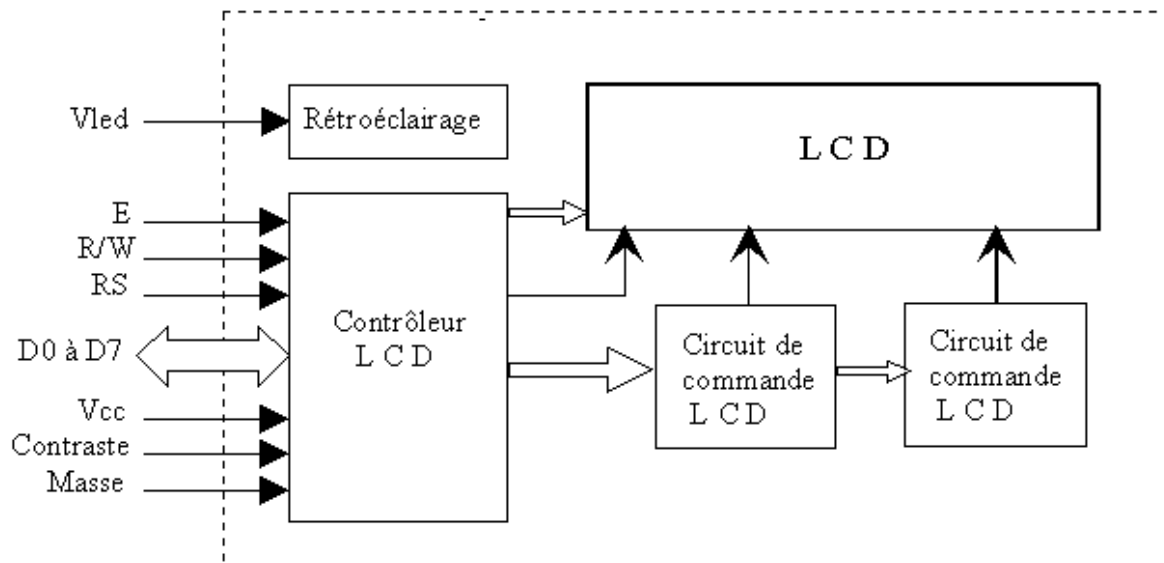
les afficheurs dits « transmissifs », c'est à dire que l'écran est muni d'un retroéclairage ce qui permet à l'utilisateur de s'en servir dans des endroits peu éclairés (utilisation pour les TV,moniteurs informatiques),

les afficheurs dits « réflectifs », ce qui signifie qu'à défaut d'être munis d'un retroéclairage, ils réfléchissent la lumière du soleil. On peut donc les utiliser tout le temps sauf lorsque la lumière est trop faible,

les afficheurs dits « transflectifs », ceux-ci combinent les effets des afficheurs réflectifs et transmissifs.

## 2.2. Le fonctionnement

Le fonctionnement est donc peu complexe, en effet il ne suffit que d'un contrôleur, souvent un micro-contrôleur. Ce micro contrôleur gère la commande de l'affichage ainsi que le traitement des données. Ainsi, on peut faire fonctionner un afficheur LCD en utilisant un montage de ce type:



*Illustration 2 : Schéma fonctionnel*

### 2.3. Le brochage et composition

D'un point de vue électrique, un afficheur LCD doit être, comme le montre le schéma ci dessus, connecté à son micro contrôleur. Pour cela l'afficheur dispose de broches. Il n'y en a que peu : en général 16. En effet, il dispose de 8 broches destinées à recevoir les données à afficher, en fait on en utilise soit 4 soit 8 en fonction du mode de fonctionnement de l'afficheur. Ensuite l'afficheur dispose de 2 broches pour son alimentation +5V-0V. Puis 4 broches qui permettent de régler le contraste, la sélection du registre afin de déterminer si il reçoit une donnée ou une commande, définir si l'afficheur devra écrire ou lire et une quatrième qui permettra de valider les actions. Les broches 15 et 16 ne sont présentes que lorsque l'afficheur est transmissif.

N° DE BROCHE	SIGNAL	NIVEAU
14	$V_{SS}$	Masse
13	$V_{DD}$	+ 5 V
12	$V_{LC}$	± 2,5 V
11	RS	0 = Instruction 1 = caractère.
10	$R/\overline{w}$	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

*Illustration 3 : Brochage de l'affichage avec le nom et l'état d'activation*

L'afficheur possède de la mémoire. En effet, il a de la mémoire appelée DD RAM qui est celle de l'affichage et la mémoire CG RAM qui elle, lui sert à générer des caractères, l'afficheur peut donc créer de nouveaux caractères.

L'afficheur peut donc fonctionner en 2 modes de fonctionnements, le mode 8bits et celui dit 4bits.

Dans le mode 8bits on utilise tous les bits de données disponibles sur l'afficheur, ce fonctionnement est le plus simple car, une fois l'afficheur configuré, c'est à dire lecture/écriture, ou transmission d'une donnée ou d'une commande. Une fois configuré, on lui envoi les données puis il suffit d'activer la broche de validation pendant une certaine durée (précisée dans la documentation technique de l'afficheur) et ainsi l'afficheur inscrit sur l'écran les données désirées.

Le mode 4 bits est un peu plus complexe, en effet, on n'utilise cette fois-ci que quatre bit de données de l'afficheur (par convention on utilise les quatre bits de poids fort et les bits de poids faible sont mis à zéro), ainsi une fois l'afficheur configuré comme voulu, il nous faut envoyer à l'afficheur notre donnée, pour cela on lui envoie les quatre bits de poids fort puis ceux de poids faibles, tout en validant à chaque fois.

Un afficheur est capable d'afficher un grand nombre de caractères du code ASCII plus des caractères japonais, des accents, des lettres grecques, et l'on peut en créer 8. Le tableau qui suit le montre bien.

Higher Lower 4bit 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000		0	a	P	^	P		-	9	3	0	0	
xxxx0001		!	1	A	Q	a	9	w	7	7	4	3	q
xxxx0010		"	2	B	R	b	r	r	4	U	x	p	0
xxxx0011		#	3	C	S	c	s	u	0	T	E	e	*
xxxx0100		\$	4	D	T	d	t	\	I	k	k	u	a
xxxx0101		%	5	E	U	e	u	=	7	+	1	0	0
xxxx0110		&	6	F	V	f	v	9	0	2	3	0	2
xxxx0111		'	7	G	W	g	w	7	7	7	7	q	π
xxxx1000		(	8	H	X	h	x	4	0	*	U	r	2
xxxx1001		)	9	I	Y	i	y	0	7	U	U	'	U
xxxx1010		*	:	J	Z	j	z	2	0	n	k	i	7
xxxx1011		+	:	K	L	k	l	*	7	0	0	*	7
xxxx1100		,	<	L	*	l	l	k	3	7	7	*	7
xxxx1101		-	=	M	I	m	i	2	2	\	0	0	+
xxxx1110		.	>	N	^	n	^	3	0	0	'	n	
xxxx1111		/	?	O	_	o	_	+	w	U	7	0	■

Illustration 4 : Tableau des caractères de l'afficheur

## 2.4. Le choix de notre afficheur LCD

Dans notre projet nous avons besoin d'un afficheur basique que l'on peut trouver facilement. Le but de notre projet est de pouvoir afficher des grandeurs électriques à savoir un courant et une tension Nous aimerions



pouvoir afficher ces grandeurs pour tout types d'appareil, à l'IUT ce serait pour le panneau solaire ou encore pour le kart.

Du fait que l'on ait deux grandeurs à afficher nous opteront pour un afficheur 16x2.

Les afficheurs disponibles sont dits, soit ALCD, soit ELCD, la différence entre les deux est simplement les instructions qui nous permettent de les contrôler qui sont plus simples pour un afficheur ELCD. De plus on utilisera un afficheur série, ce qui nous permet d'envoyer chaque caractère à la fois.

Par contre, nous tomberons sur un inconvénient au niveau de l'alimentation de l'afficheur, en effet il ne peut être alimenté par plus de 5 volts pour un courant très faible, il nous faudra donc en tenir compte lorsque nous réaliseront notre carte électronique.

Finalement, notre afficheur sera un ELCD 162.

## **Conclusion**

En conclusion, nous pouvons dire que la conception du schéma de notre carte sera, finalement, très simplifié grâce à l'utilisation de composants simples et faciles à mettre en oeuvre. Le temps passé à la recherche de ceux-ci ne sera donc pas perdu mais entièrement compensé par le temps gagné lors de la réalisation du montage.

La seule difficulté réside, en fait, dans la découverte du langage basic, qui s'avère relativement simple d'accès avec des instructions dédiées spécialement à l'usage d'afficheur L.C.D..

## **Index des illustrations**

Illustration 1 : Trois types d'afficheurs.....	12
Illustration 2 : Schéma fonctionnel.....	13
Illustration 3 : Brochage de l'affichage avec le nom et l'état d'activation.....	14
Illustration 4 : Tableau des caractères de l'afficheur.....	16