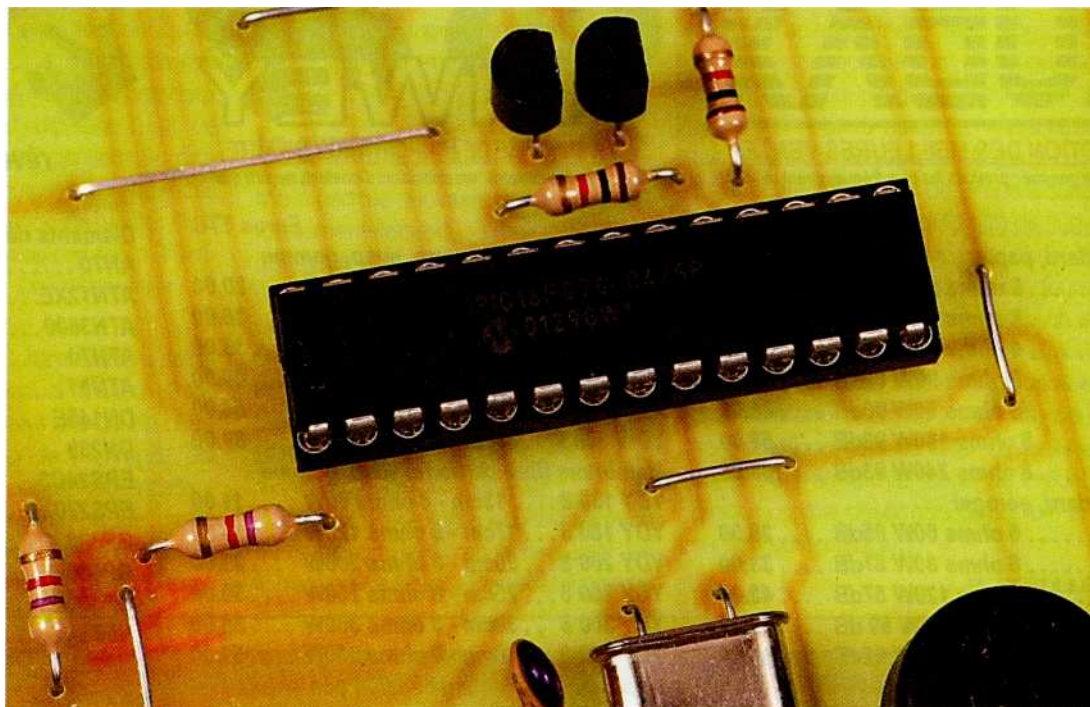


# A la découverte des microcontrôleurs PIC

## (Première partie)



Cette série d'articles que nous vous proposons sur *Electronique Pratique* concerne les microcontrôleurs PIC et, plus particulièrement, la famille 16F84. Vous apprendrez tout au long de ces cours à programmer et à utiliser ce composant très performant.

### Du microprocesseur au microcontrôleur

La découverte du microprocesseur date de près de trente ans. En effet, la fabrication du premier circuit remonte à 1970, année où la société Intel met au point le premier microprocesseur : le 4004. On n'imagine pas, à l'époque, que cette révolution industrielle donnera naissance à l'ordinateur individuel. Depuis, leur puissance de calcul et l'intégration des transistors les constituant n'ont cessé d'évoluer. On retrouve désormais les microprocesseurs dans la plupart des applications, que ce soit pour piloter une centrale à l'intérieur d'un ordinateur ou remplacer le pro-

grammateur d'une machine à laver.

Les microprocesseurs ne sont jamais employés seuls, des circuits périphériques leur sont toujours associés pour pouvoir être intégrés au sein d'une application (**figure 1**).

L'un des avantages d'un montage à base de microprocesseur, par rapport à un montage en logique câblée, réside dans sa souplesse d'emploi.

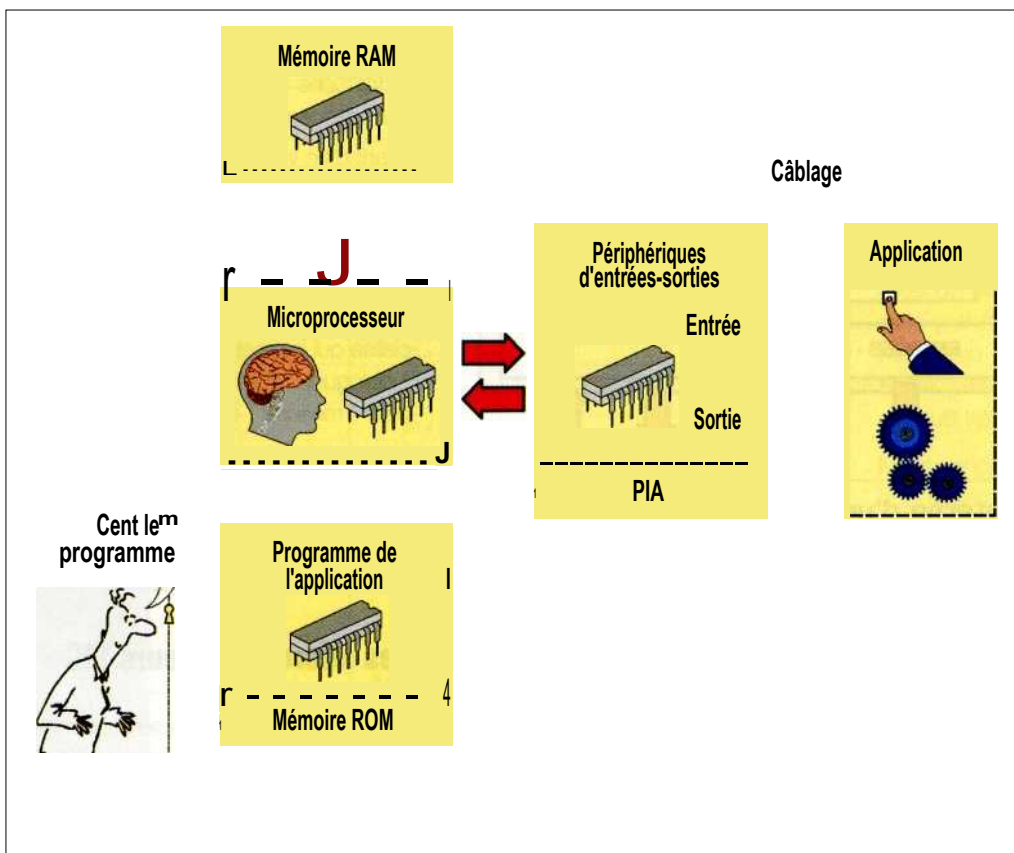
En effet, nous entrons dans le domaine de la logique programmable où le fonctionnement du montage dépend d'un programme logé dans une mémoire. Celui-ci peut être modifié pour changer les équations régissant l'application, sans toutefois entraî-

ner de changement au niveau du câblage des entrées-sorties.

### Les microcontrôleurs

Les microprocesseurs, nous venons de le voir, possèdent un indéniable avantage sur la logique câblée. En effet, pour modifier le fonctionnement d'une application, il suffit de changer le programme sans refaire de câblage.

Les microcontrôleurs possèdent, quant à eux, la puissance d'un microprocesseur, mais présentent un atout supplémentaire : ils possèdent, dans le même boîtier, les périphériques intégrés (**figure 2**). Autrement dit, le programme de l'application est en interne et non plus dans un circuit mémoire



- Une unité centrale, coeur du système, également appelée CPU (Central Processing Unit). Dans cette unité centrale, nous retrouverons plusieurs éléments comme l'unité arithmétique et logique (UAL) que nous détaillerons dans un prochain numéro.

- Une mémoire contenant le programme à exécuter par le microcontrôleur, généralement appelée "mémoire morte" ou ROM (Read Only Memory), "mémoire à lecture seule". Cette mémoire a la particularité de sauvegarder en permanence les informations qu'elle contient, même en absence de tension (ce qui est primordial,

sinon il faudrait reprogrammer le microcontrôleur à chaque remise sous tension !).

### Les microprocesseurs ne sont jamais employés seuls

externe et les périphériques d'entrées-sorties sont également intégrés, d'où l'économie de nombreux circuits périphériques.

Cette caractéristique explique que les montages deviennent encore plus simples et la programmation plus aisée. Un système à base de microprocesseur oblige le concepteur à réaliser un décodage d'adresse pour permettre au microprocesseur de ne dialoguer qu'avec un seul périphérique à la fois.

Un microcontrôleur seul peut donc gérer une application, sans faire forcément appel à d'autres circuits associés.

Sur la **figure 2**, le microcontrôleur possède en interne la mémoire programme contenant le programme de l'application, ainsi que le port d'entrées-sorties qui va permettre au microcontrôleur de s'interfacer avec l'application.

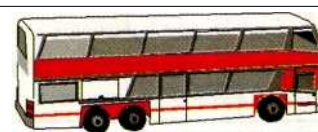
On le voit ici, par rapport au schéma à base de microprocesseur présenté **figure 1**, nous avons encore franchi un degré d'intégration en rassemblant tous les circuits nécessaires au fonctionne-

ment d'une application dans le même boîtier.

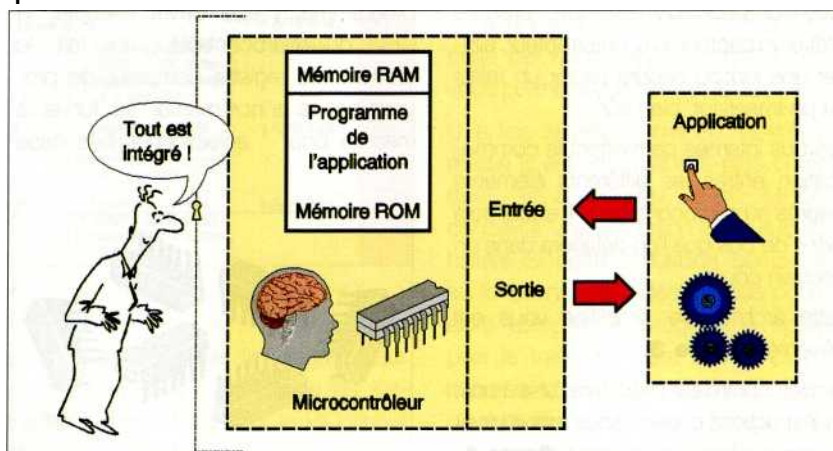
### Structure interne d'un microcontrôleur [figure 3]

Un microcontrôleur, le plus simple qu'il soit, possède au minimum les éléments suivants

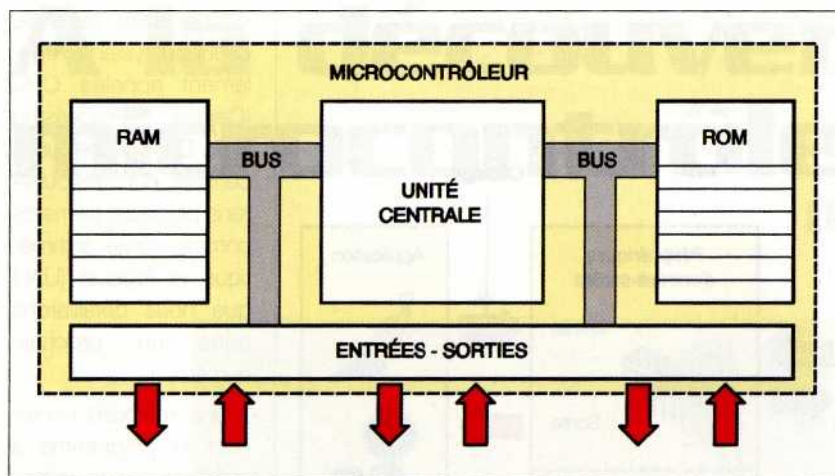
- Une mémoire vive également appelée RAM (Random Access Memory). Cette mémoire permet de sauvegarder temporairement des informations. Il est à



### 4 Périphériques intégrés







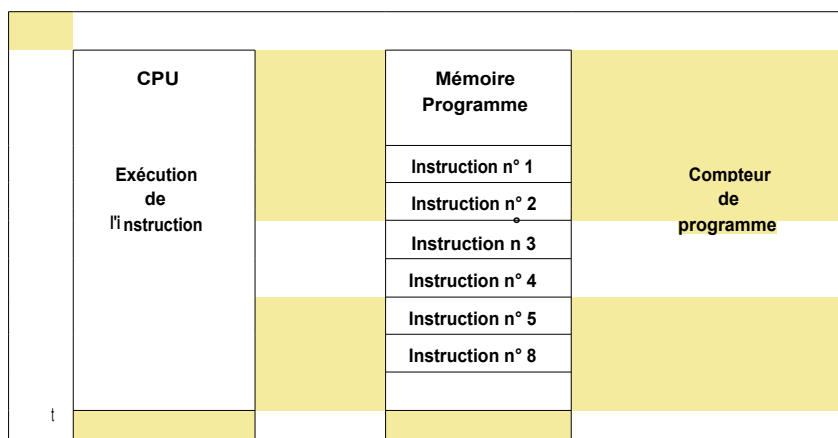
### 3 Structure interne d'un microcontrôleur

noter que le contenu d'une RAM n'est sauvegardé que pendant la phase d'alimentation du circuit. Le microcontrôleur pourra utiliser cette mémoire pour stocker des variables temporaires ou faire des calculs intermédiaires.

- Un port d'entrées-sorties permettant au microcontrôleur de dialoguer avec

En réalité, le microcontrôleur exécutera les instructions que vous avez transférées dans sa mémoire programme.

Sans encore entrer trop dans les détails, sachez qu'un registre spécifique du microcontrôleur (le CP ou "Compteur de programme") est chargé de pointer l'instruction stockée en ROM (c'est votre



### R Le microcontrôleur exécutera, une à une les instructions codées

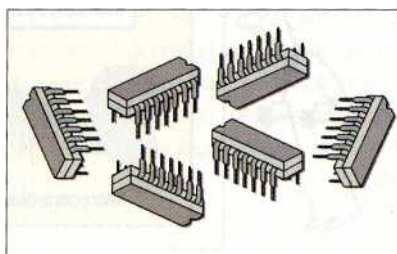
l'extérieur pour, par exemple, prendre l'état d'un capteur, d'un interrupteur, allumer une led ou encore piloter un relais (via un transistor, bien sûr).

Des bus internes permettent la communication entre les différents éléments intégrés au microcontrôleur. Il existe trois sortes de bus que l'on détaillera dans un prochain cours.

Cette architecture simplifiée vous est présentée **figure 3**.

Le microcontrôleur exécutera, une à une, les instructions codées sous forme binaire dans la mémoire programme (**figure 4**).

programme...) que devra exécuter la CPU du microcontrôleur. En fait, le contenu du registre compteur de programme va s'incrémenter au fur et à mesure pour "sélectionner" la case



mémoire suivante. Ainsi, la CPU du microcontrôleur exécutera toutes les instructions que vous avez transférées dans la mémoire du microcontrôleur.

Avant d'en venir au microcontrôleur PIC, on peut dire, en simplifiant, qu'un microcontrôleur est un microprocesseur auquel on a intégré divers périphériques, en particulier la mémoire contenant le programme à exécuter et un circuit spécialisé qui permet au microcontrôleur de "dialoguer" avec l'extérieur, que se soit pour "mesurer" ou "actionner".

Il existe de nombreux types de microcontrôleurs, qu'ils soient spécifiques pour une fonction donnée ou bien banalisés et configurables pour de nombreuses applications. Quelle famille...

## Les microcontrôleurs PIC

De nombreux fabricants se sont implantés sur le marché des microcontrôleurs. La société américaine Microchip Technologies a mis au point un microcontrôleur CMOS (Complementary Métal Oxyde Semi-conducteur) : **Le PIC**. Ce microcontrôleur, encore très utilisé à l'heure actuelle, constitue un compromis entre simplicité d'emploi et prix de revient. Il fait partie de la famille des circuits RISC (Reduced Instruction Set Computer), caractérisée par leur vitesse d'exécution et leur jeu d'instruction réduit (le PIC 16F84 possède seulement trente-cinq instructions de base).

Il existe de nombreuses versions de PIC possédant chacune des caractéristiques différentes. Des tableaux comparatifs permettent de choisir le PIC le plus adéquat par rapport à l'application envisagée. Un comparatif vous est proposé sur le **tableau 1**. Nous reviendrons ultérieurement sur les caractéristiques des PIC présentés.

## Un minimum de matériel

Ce premier article vous a plu, vous avez décidé de réaliser une application à base de PIC, que vous faut-il pour commencer ?

Dans un premier temps, une fois votre projet établi (nous y reviendrons ultérieurement avec une application concrète), il va falloir écrire un programme en

PIC	programme	RAJA	Commentaires	Tuner	E
Quelques PIC de la série 12					
120509	512x12	25x8		1 x8bits	
12CE518	1024 x 12	41 x 8		1 x 8bits	
12CE518	512 x 12	25 x 8	E2PROM : 16 x 8	1 x 8bits	6
12CE518	1024 x 12	41 x 8	E2PROM : 16 x 8	1 x 8bits	6
	1024 x 14	128 x 8	4 ADC 8 bits	1 x 8bits	6
	2048 x 14	128 x 8	4 ADC 8 bits	1 x 8bits	6
	2048 x 14	128 x 8	E2PROM:16x8-4ADC8bits	1 x 8bits	6
Quelques PIC de la série 16					
	384 x 12	25 x 8		1 x 8bits	12
	512x12	25x8		1 x 8bits	12
	512x12	24x8		1 x8bits	21
	1kx12	32x8		1 x8bits	13
	2k x 12	80x8		1 x 8bits	21
	2kx12	80x8		1 x8bits	21
18C82A	2048 x 14 (flash)	128 x 8		1 x 16bits 2 x 8bits	22
18083	4096 x 14 (flash)	192 x 8		1 x 16bits 2 x 8bits	22
	2048 x 14	128 x 8		1 x 16bits 2 x 8bits	33
	4096x 14 (flash)	192 x 8		1 x 16bits 2 x 8bits	33
	8192 x 14	128 x 8		1 x 16bits 2 x 8bits	33
18071	1024 x 14	4 canaux ADC		1 x 8 bits	13
18072	2048x 14	5 canaux ADC		1 x 16 bits 2 x 8 bits	
18C73A	4096 x 14	5 canaux ADC		1 x16 Ms 2x 8 bits	22
18C74A	4096x14	8canaux ADC		1x16bits 2x8bBSj	33
18076	8192 x 14 (flash)	5 canaux ADC		1 x 16 bits 2 x 8 bits	22
18077	8192 x 14	8 canaux ADC		1x 16 bits 2 x 8bits	33
18F83	512 x 14 (flash)	36 x 8	E2PROM : 64 x 8	1 x 8bits	13
SIC 16084	1024 x 14	68 x 8	E2PROM : 64 x 8	1 x 8bits	13
PIC 16F84	1024 x 14 (flash)	68 x 8	E2PROM : 64 x 8	1 x 8bits	13
PIC 16F827	1024 x 14 (flash)	224 x 8	E PROM :128 x 8 1 UART 2 comparateurs analogiques	31~f1eltilds 3 8/16 bits	16
MPIC 16F628	2048 x 14 (flash)	224 x 8	E PROM :128 x 8 1 UART 2 comparateurs analogiques	3 8/16 bits 3 8/16 bits	16
18F873	4k x 14 (flash)	192 x 8	E PROM :128 x 8 5 ADC 10 bits - 2 PWM	3 8/16 bits 3 8/16 bits	20
16F874	4k x 14 (flash)	192 x 8	E PROM : 128x 8 8 ADC 10 bits - 2PWM	3 8/16 bits 3 8/16 bits	33
18F876	8k x 14 (flash)	368 X 8	E PROM : 256 x 8 5 ADC 10 bits - 2 PWM	3 8/16 bits 3 8/16 bits	20
18F877	8k x 14 (flash)	368 x 8	E PROM : 256 x 8 8 ADC 10 bits - 2 PWM	3 8/16 bits 3 8/16 bits	33
Quelques PIC de la série 17					
17C62	2k (Eprom)	232 x 8	Freq:25MHz - 2 PWM 1 USART- 11 sources IT	4 8/16 bits 4 8/16 bits	33
17C62A	2k (Eprom)	232 x 8	Freq:33MHz - 2 PWM 1 USART- 11 sources IT	4 8/16 bits 4 8/16 bits	33
17C64	8k (Eprom)	454 x 8	Freq:33MHz - 2 PWM 1 USART-11 sources IT	4 8/16 bits 4 8/16 bits	33
Quelques PIC de la série 18					
18F242	16k (flash) + 8k	768 x 8	Freq:40MHz- 2 PWM 1 USART- 5 ADC 10 bits 75 instructions -17 sources 1rE_pran256x8	4 8/16 bits 4 8/16 bits	33
18F442	16k (flash) + 8k	768 x 8	Freq:40MHz-2 PWM 1 ISART- 8ADC 10 bits 75 instructions-18 sources IT E_prom 256 x 8	4 8/16 bits	33
PIC 18F452	32k (flash) + 16k	1536 x 8	Freq:40MHz-2 PWM 1 ISART- 8 ADC 10 bits 75 instructions-18 sources ITE_prom256x8	4 8/16 bits	33

Tableau 1

assembleur (pour débiter). Un programme en assembleur est constitué par une suite d'ordres (mnémoniques) que devra exécuter le microcontrôleur. Cela sup-

pose, bien sûr, que le programmeur (c'est-à-dire vous !) connaisse les instructions disponibles pour le PIC choisi et également la façon d'utiliser ces ins-

tructions (pas d'affolement, nous verrons toutes les instructions du 16F84).

Pour écrire notre programme en assembleur, nous pouvons utiliser n'importe quel éditeur de texte, par exemple notepad.exe, logiciel fourni avec Windows. Une autre possibilité, plus conviviale, consistera à utiliser l'éditeur fourni dans l'outil de développement MPLAB disponible gratuitement sur le site internet de Microchip ([www.microship.com](http://www.microship.com)). Un article sera ultérieurement consacré à cet outil très performant.

Nous venons d'écrire notre programme en assembleur (également appelé 'code source'), il faut maintenant compiler ce fichier. La compilation consiste à remplacer les ordres mnémoniques du fichier assembleur par des codes binaires compréhensibles par le microcontrôleur. En effet, les microprocesseurs, tout comme les microcontrôleurs, ne comprennent que les "0" et les "1". Pour effectuer cette compilation, il faut disposer d'un compilateur. L'intégré MPLAB cité plus haut permet la compilation.

Dernière étape, la compilation a réussi (ouf, il n'y a pas d'erreur de syntaxe dans le code source), un fichier binaire comportant l'extension .hex (en général) vient d'être créé par le compilateur. Il reste à transférer ce fichier présent sur le disque dur de votre ordinateur vers la mémoire programme de votre microcontrôleur PIC. De nombreux kits de programmeurs de PIC proposés par les annonceurs d'Electronique Pratique (pour 45 euros environ) permettent de transférer le fichier binaire vers la mémoire du PIC. Un mini programmeur est proposé dans cette même revue. Si toutefois vous possédez le programmeur proposé par Microchip, le logiciel MPLAB vous permettra de réaliser le transfert.

Une fois le programme en assembleur réalisé (c'est finalement le plus compliqué), vous pourrez, avec MPLAB, concevoir une simulation (si vous le souhaitez). Si cette simulation correspond au fonctionnement désiré, vous pourrez alors compiler le programme source, puis le transférer vers la mémoire du microcontrôleur. Ces quelques étapes devront être exécutées à chaque nouveau programme.

## Pour conclure

Pour terminer cette première approche autour d'un microcontrôleur PIC, on peut déjà écrire qu'avec un minimum de moyens (un programmeur de PIC), on pourra réaliser une application simple et peu coûteuse ne comportant que très peu de composants. Il faudra, bien évidemment, se familiariser (petit à petit !) avec le langage assembleur qui paraît, au premier abord, assez compliqué et surtout avec les Instructions (il n'y en a que 35!), ainsi qu'à la façon de les utiliser.

Dans notre prochain cours, nous "voyagerons" au coeur d'un PIC en détaillant les différents blocs qui constituent son architecture interne.



P. MAYEUX

## PROJET

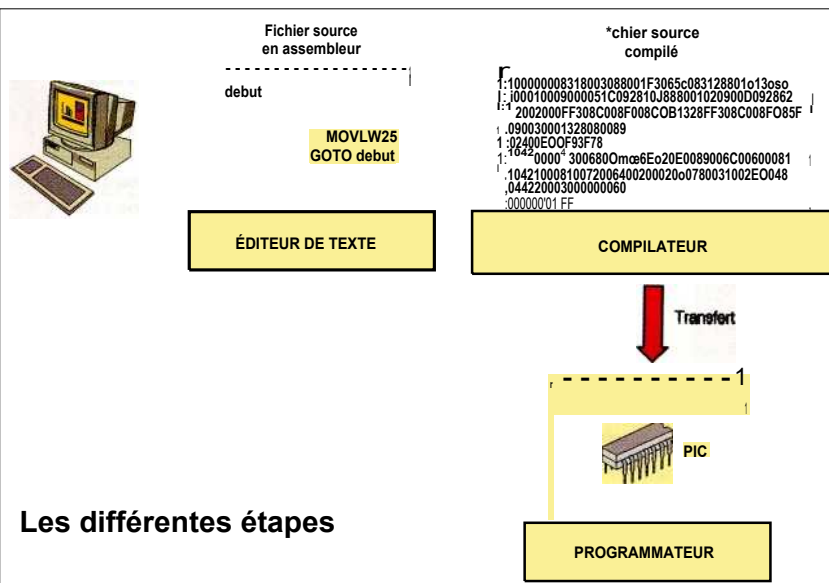
Application à base de PIC



Écriture du programme sain



## Le transfert



Les différentes étapes

816 pages, tout en couleurs



Envoi contre 5,00€ (10 timbres-poste à 0,50€ ou chèque)

**NOUVEAU**

# Catalogue Général

# Sele n n n n n ronic

Connectique, Electricité.  
Outils. Librairie technique.  
Appareils de mesure.  
Robotique. Etc.

Plus de 15.000 références

Coupon à retourner à : **Selectronic B.P 513 59022 LILLE Cedex**

☐ OUI, je désire recevoir le **Catalogue Général 2004 Selectronic** à l'adresse suivante (ci-joint 5,00€ en timbres-poste (10 timbres de 0,50€) ou chèque) :

Mr./Mme : ..... Tél : .....  
N° ..... Rue : .....  
Ville : ..... Code postal : .....