

Qu'est-ce qu'un microcontrôleur ?

La *figure 1* montre l'**architecture** simplifiée d'un **système informatique**, qui comporte quatre éléments:

- une **mémoire morte** (*ROM = Read Only Memory*) qui contient les instructions. Son contenu est permanent : il reste intact lorsque le système n'est plus alimenté.
- un **processeur**, qui exécute les instructions. Il est cadencé par une horloge (H). Il contient un compteur ordinal (*Program Counter, PC*) qui contient l'adresse de l'instruction courante.
- une **mémoire vive** (*RAM = Random Access Memory*), qui permet de stocker des données. Son contenu est perdu lorsque le courant est coupé.
- des **circuits d'Entrée-Sortie** (*I/O = Input/Output*). Ce sont les circuits qui permettent l'interaction avec l'extérieur.

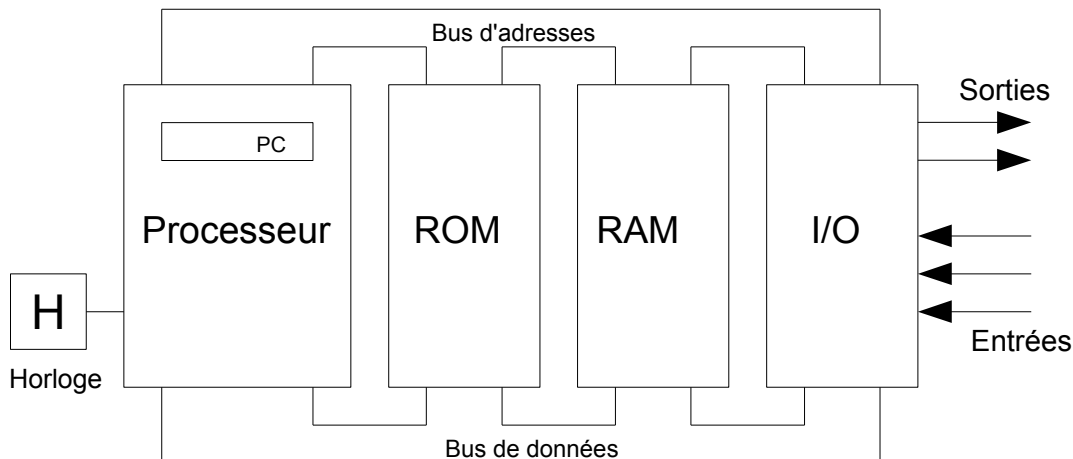


Figure 1

Une **carte-mère** d'un PC possède cette même architecture. La mémoire morte est appelée **BIOS**. C'est le premier programme qui s'exécute lorsqu'on allume le PC.

Le processeur (par exemple un *Pentium 4*) est un circuit intégré contenant des millions de transistors, fonctionnant à une fréquence très élevée. Malgré la très faible consommation de chacun de ces transistors (technologie *C-MOS*), la complexité du circuit et la fréquence élevée (jusqu'à plusieurs GHz) font que le processeur consomme une énergie importante (plusieurs dizaines de Watt pour un Pentium 4). Il est donc nécessaire de le refroidir avec un refroidisseur et un ventilateur.

Les barrettes de mémoire vive (*SD-RAM, DDR, DDR-2, etc*) ont généralement des capacités exprimées en centaines de MB (millions d'octets, plus exactement 2^{20} octets) ou en GB (milliards d'octets, plus exactement 2^{30} octets).

Les circuits d'entrées-sorties correspondent au clavier, à la souris et à l'écran (interface homme-machine), mais aussi aux lecteurs de disques, CD, DVD, etc.

Un **microcontrôleur** est un système informatique contenu dans **un seul circuit intégré**. L'architecture est la même que celle présentée sur la *figure 1*. Par rapport à une carte-mère de PC, les éléments qui constituent un microcontrôleur sont plus simples, moins

puissants, leur capacité est plus limitée :

- la mémoire morte contient généralement de 1 KB à quelques centaines de kB.
- le processeur est cadencé à des fréquences de quelques MHz ou dizaines de MHz. Il ne consomme généralement qu'une fraction de Watt. Son jeu d'instructions est plus simple.
- la mémoire vive est généralement très limitée: de quelques centaines de Bytes, à quelques dizaines de kB, selon les modèles.
- les circuits d'entrée-sortie sont simplement des entrées logiques (pour lire une valeur binaire, par exemple un interrupteur) et des sorties logiques (capables de fournir quelques mA, par exemple pour commander une LED). Certains microcontrôleurs ont aussi des entrées analogiques: des convertisseurs analogiques-numériques (*ADC= Analog to Digital Converter*) et parfois des sorties analogiques: des convertisseurs numériques analogiques (*DAC=Digital to Analog Converter*).

L'intérêt des microcontrôleurs est leur coût très faible (à partir d'un demi US\$, soit quelques centaines de CFA), leur faible consommation de courant (quelques dizaines de mA) et leur taille très réduite (un seul circuit intégré, ayant de 6 à quelques centaines de pattes). Ils sont donc utilisés dans de très nombreuses applications.

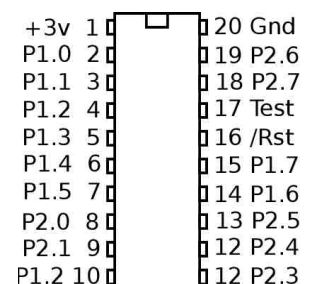
Bien que des microcontrôleurs existent depuis les années 1970, ils se sont développés de manière spectaculaire depuis quelques années. Alors qu'il était encore complexe et coûteux de mettre en œuvre un microcontrôleur au début des années 2000, cette tâche est maintenant beaucoup plus simple et ne nécessite que du matériel très peu coûteux, en plus d'un simple PC utilisé comme système de développement. Les microcontrôleurs sont donc devenus des composants électronique incontournables, modifiant profondément la manière de concevoir les circuits électroniques.

Plusieurs fabricants proposent des microcontrôleurs (*Microchip, Atmel, Texas-Instrument, FreeScale, NXP, ST-micro, Cypress, etc*). Chaque fabricant propose souvent plusieurs familles de microcontrôleurs (*PIC et dsPIC* chez *Microchip*; *AVR, AVR32 et ARM* chez *Atmel*, etc). Chaque famille comporte souvent des dizaines de modèles, qui diffèrent principalement par les tailles mémoires (RAM et ROM) et le nombre de pattes d'entrée-sortie.

A titre d'exemple, le processeur **MSP430G2253**, de **Texas Instrument**, contient:

- un processeur 16 bits, avec une centaine d'instructions, cadencé entre 1 et 16MHz , avec 16 registre de 16 bits.
- une mémoire morte de type *flash* (technologie similaire à celle des clés USB) de 16 kB. Cette mémoire peut être effacés et écrite à nouveau un très grand nombre de fois.
- une mémoire vive de 512 octets.
- 16 pattes d'entrée-sorties. Huit d'entre elles peuvent être connectées à un convertisseur analogique-numérique de 10 bits de résolution. Certaines de ces pattes ont également d'autres fonctions spécifiques (capture d'événements, utilisation d'un quartz, etc). Le MSP430G est disponible dans plusieurs boîtiers, dont le DIL20 (dual in line 20 pins), facile à mettre en oeuvre. Son brochage est donné ci-contre.

D'autres microcontrôleurs de cette même famille ont un boîtier DIL14. Dans ce cas, les pattes P2.0 à P2.5 ne sont pas disponibles. La gamme des MSP430 est très vaste, la grande majorité des modèles étant disponibles uniquement en boîtiers SMD (Surface Mounted Device = Composants Montés en

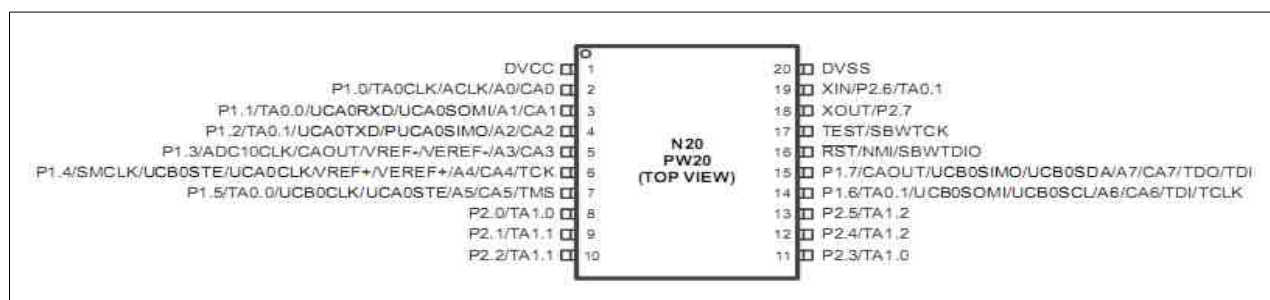


Surface).

Un des atouts de la famille des MSP430 est sa consommation extrêmement faible.

Premier schéma

Beaucoup de mes étudiants se sont montrés perplexes en regardant le brochage d'un microcontrôleur, tel qu'il trouve sur l'une des premières pages de la description du circuit. Voici par exemple celui du MSP430G2553 (fichier *MSP430G2x53-13.pdf*) :



Chaque patte est associée à un ou plusieurs noms, tous un peu incompréhensibles pour un débutant. Essayons donc d'y voir plus clair et d'identifier les pattes indispensables à la réalisation d'un premier montage fonctionnel à microcontrôleur.

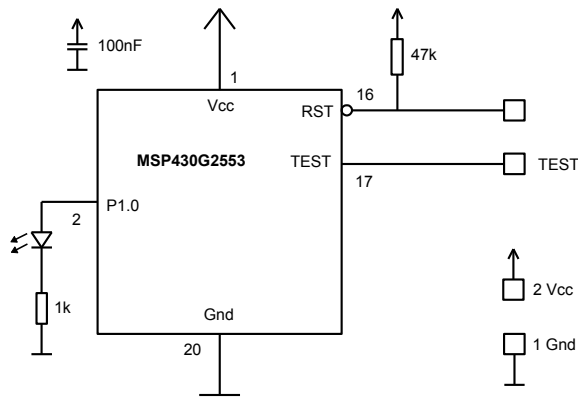
Nous pouvons identifier trois groupes de pattes: les **alimentations**, les pattes de **programmation** et les pattes d'**entrées-sorties** pour l'application proprement dite.

Les microcontrôleurs sont réalisés en technologie **CMOS** et nécessitent une alimentation unique. Les **MSP430** acceptent une tension comprise entre 1.8 et 3.6 volt. Deux piles AA ou AAA 1.5 volt en série peuvent donc convenir. La borne négative de l'alimentation est souvent désignée par **Gnd** (*Ground* = masse). C'est la patte 14 du **MSP430G2302**, qui s'appelle **DVSS**. La borne positive est appelée **DVcc**, c'est la patte 1.

La patte **RST** ou **RESET** est la patte 16. Ignorons pour le moment le fait que cette patte a aussi d'autres noms. Le **RST** est une entrée du circuit. Notez la barre sur son nom : cela signifie que l'entrée est active à zéro. On le note aussi parfois avec le signe / (*slash*) précédent le nom. C'est une bonne habitude de noter chaque fois qu'un signal est actif à zéro. Il sera donc nécessaire de fixer à l'état 1 la patte **RST** pour le fonctionnement normal du microcontrôleur. C'est une résistance reliée au *Vcc* qui va jouer ce rôle. On parle de résistance de tirage (en anglais, *pull-up*: tirer vers le haut). Une valeur d'environ 47 kOhm convient dans ce cas.

La programmation des **MSP430** peut se faire par l'intermédiaire de deux signaux: le **RST** et un signal nommé **TEST** (patte 17). Il n'est pas nécessaire de comprendre le rôle exact de ce signal. Il est généré, ainsi que le **RST**, par le logiciel de programmation qui tourne sur le PC (par exemple Code Composer Studio ou Energia). Il est transmis par le programmeur, par exemple celui inclus dans le LaunchPad MSP430.

Les autres pattes du **MSP430G2302** sont des pattes d'entrée-sortie. Elles sont regroupées en 2 **ports**: **P1** et **P2**. Ces deux ports sont complets : ils comportent chacun 8 pattes. D'autres versions du MSP430 sont proposées en boîtier à 14 pattes (DIL 14). Dans ce cas, les pattes P2.0 à P2.5 n'existent pas.



Il nous reste maintenant à choisir notre application. Prenons à titre d'exemple la plus simple : une seule patte du microcontrôleur sera utilisée, pour y brancher une diode lumineuse (*LED*) avec une résistance de limitation du courant reliée en série vers la masse. Nous avons donc maintenant le schéma complet de notre montage (ci-contre).

Notez la présence d'un condensateur de découplage entre le + et le -. Les électroniciens savent qu'il est utile pour filtrer l'alimentation. Appréciez la

simplicité : seulement 5 composants électroniques. Un connecteur permet d'apporter l'alimentation à notre montage, un autre les signaux de programmation. Ce connecteur est utile dans la phase de développement du programme correspondant à l'application.