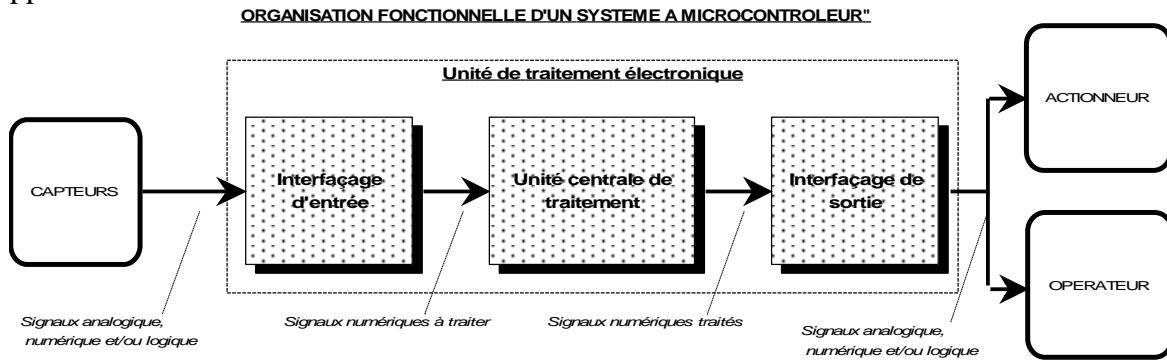


Cours C032_ LES SYSTEMES MICRO-PROGAMMES

1- Mise en situation :

Les systèmes à microprocesseurs ou microcontrôleurs sont omniprésents dans la vie de tous les jours. On les rencontre dans des systèmes complexes (ordinateurs individuels) mais également dans des systèmes simples (machine à laver, pompe à essence, store automatique, ...) où leur grande souplesse est appréciée.

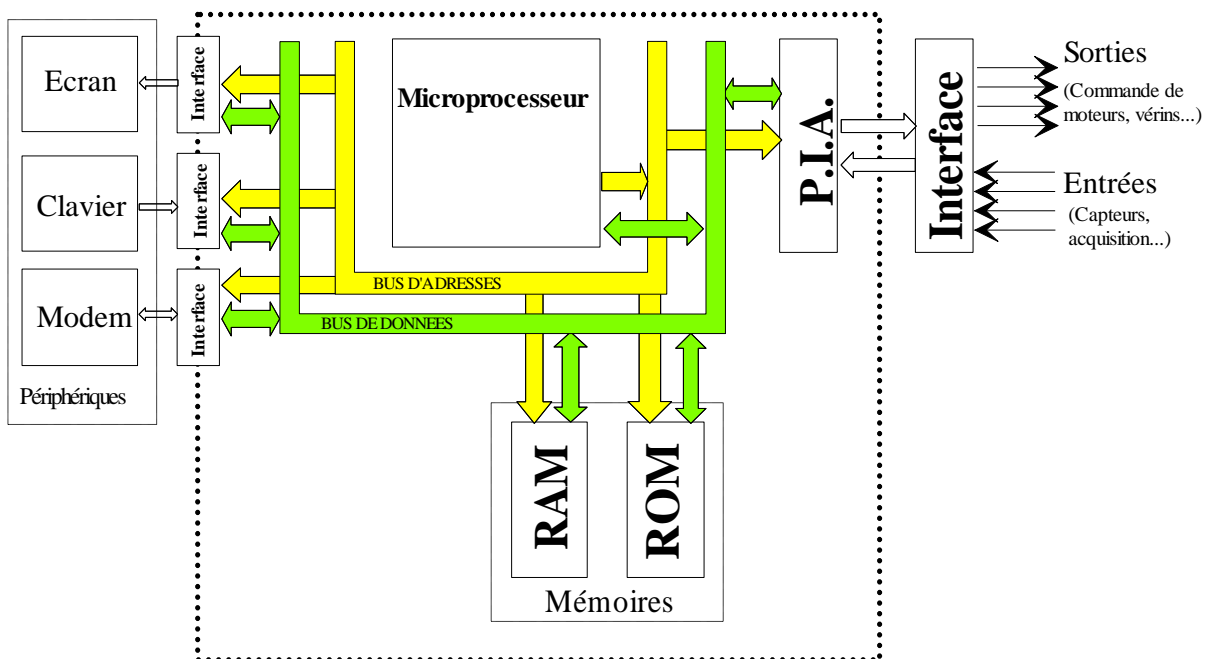


Un objet technique, intégrant de l'électronique, fait souvent apparaître des fonctions ayant pour rôle le traitement d'informations : opérations arithmétiques (addition, multiplication...) ou logiques (ET, OU...) entre plusieurs signaux d'entrée permettant de générer des signaux de sortie.

Ces fonctions peuvent être réalisées par des circuits intégrés analogiques ou logiques. Mais, lorsque l'objet technique devient complexe, et qu'il est alors nécessaire de réaliser un ensemble important de traitements d'informations, il devient plus simple de faire appel à une structure à base de microcontrôleur.

2- Description et structure interne

L'environnement du microcontrôleur :



Le CPU (microprocesseur) :

C'est un circuit complexe abritant plusieurs millions de transistors dans un espace de quelques cm³. Ce composant est le cerveau d'un système microprogrammé : il est chargé d'effectuer toutes les opérations nécessaires. On peut noter qu'il existe 2 catégories de microprocesseur : les CISC et les RISC.

- **CISC** (Complex Instruction Set Computer) : Ce microprocesseur possède un nombre important d'instructions. Chacune d'elles s'exécute en plusieurs périodes d'horloges.
- **RISC** (Reduced Instruction Set Computer) : Ce microprocesseur possède un nombre réduit d'instructions. Chacune d'elles s'exécute en une période d'horloge.

Les bus :

Les bus sont des liaisons permettant de relier entre eux les différents organes du système. Concrètement, ce sont des ensembles de fils électriques qui permettent de transmettre les signaux aux composants du système. On distingue trois sortes de bus :

- ☞ **Le bus de données** : il permet de transférer les données, c'est à dire les informations traitées par le microprocesseur. Ces données sont de type binaire.
- ☞ **Le Bus d'adresse** : il permet d'établir la liaison proprement dite entre le microprocesseur et un composant. Chaque composant réagit dans un domaine d'adresse précise (de la même manière que lorsque quelqu'un vous téléphone, c'est votre appareil qui sonne et non pas celui de votre voisin).
- ☞ **Le Bus de contrôle** : il n'est pas représenté sur le schéma général (afin de simplifier celui-ci). Son rôle est d'informer le microprocesseur sur l'état du système et de le prévenir en cas de dysfonctionnement.

Mémoires de données :

Ce dispositif permet de mémoriser temporairement les données générées par le microprocesseur pendant les différentes phases du traitement numérique (résultats d'opérations, états des capteurs...). Ces mémoires sont accessibles en écriture et en lecture.

On en trouve 2 types :

- ☞ De la **mémoire vive (RAM)** volatile (données perdues en cas de coupure de l'alimentation) ayant un temps de lecture et écriture assez court (quelques ns),
- ☞ De la **mémoire morte (EEPROM)** non-volatile (données conservées en cas de coupure de l'alimentation) ayant un temps d'écriture assez élevé (quelques ms) par rapport au temps de lecture qui est assez faible (quelques ns).

Mémoires programmes :

Ce dispositif contient les instructions du programme que doit exécuter le microprocesseur. Ce type de mémoire (appelée mémoire morte), est uniquement accessible en lecture. Sa programmation nécessite une procédure particulière et un matériel adéquate. Il en existe différents types selon leur mode de programmation :

- ☞ De la **ROM** dont le contenu est programmé lors de sa fabrication,
- ☞ De la **PROM** programmable électriquement une seule fois par le,
- ☞ De la **EPROM** programmable électriquement et effaçable aux U-V (appelée aussi UV PROM),
- ☞ De la **EEPROM** programmable et effaçable électriquement.

Le PIA:

C'est un périphérique de sortie parallèle.

Il permet la communication avec l'extérieur (aussi bien en entrée qu'en sortie) de manière parallèle. Il est très utilisé car il permet de contrôler beaucoup de systèmes du fait du grand nombre d'entrées et sorties disponibles.

Les interfaces :

Les interfaces permettent de relier plusieurs constituants n'étant pas forcément compatibles entre eux. Cet étage va donc réaliser une adaptation entre les différents signaux présents en entrée et en sortie. Par exemple le microprocesseur peut donner l'ordre d'ouverture d'une vanne mais il ne dispose pas d'assez d'énergie pour effectuer cette action. L'interface fournira l'énergie nécessaire.

Les périphériques :

Ce sont des organes extérieurs au système microprogrammé. Sans périphériques, le système fonctionnerait mais il ne servirait à rien puisqu'il n'échangerait aucune information avec l'extérieur. Parmi les périphériques les plus connus, on peut citer : l'écran, le clavier, la souris, l'imprimante, les haut-parleurs, le scanner, la clé USB, ...

3- Description et programmation de la tâche à accomplir

On définit dans le cahier des charges l'ensemble des commandes ou actions que doit réaliser le microcontrôleur. Ces définitions sont données dans un ordre spécifique : il s'agit d'un fonctionnement séquentiel. Avant d'en générer le code machine binaire compréhensible par le microcontrôleur, des étapes intermédiaires sont nécessaires afin de réduire les risques d'erreurs et les difficultés. On fait d'abord apparaître le cycle de fonctionnement à l'aide de deux représentations normalisées :

l'algorithme et l'algorithme (voir cours sur les algorithmes). On peut ensuite écrire le programme associé en utilisant différents langages :

- ☞ Un langage assembleur spécifique au type du microprocesseur qui est traduit en code machine lors de l'opération d'assemblage.
- ☞ Un langage évolué (C, Pascal et Basic) utilisant des procédures adaptées au type de microprocesseur et qui est traduit en code machine par une opération de compilation effectuée par un "Cross - compilateur".