

Chapitre 4: Architecture et fonctionnement d'un microcontrôleur

1. Introduction :

Au jour d'aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi d'électronique embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le cout de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Pour faire de l'électronique programmée, il faut un ordinateur et un composant programmable. Il existe tout plein de variétés différentes de composants programmables, à noter : les microcontrôleurs.



Un **microcontrôleur** (en notation abrégée **µc**, ou **uc** ou encore **MCU** en anglais) est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire

morte et mémoire vive), unités périphériques et interfaces d'entrées-

sorties. On le programme par le biais d'un ordinateur grâce à un langage informatique, souvent propre au type de microcontrôleur utilisé. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de

fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un

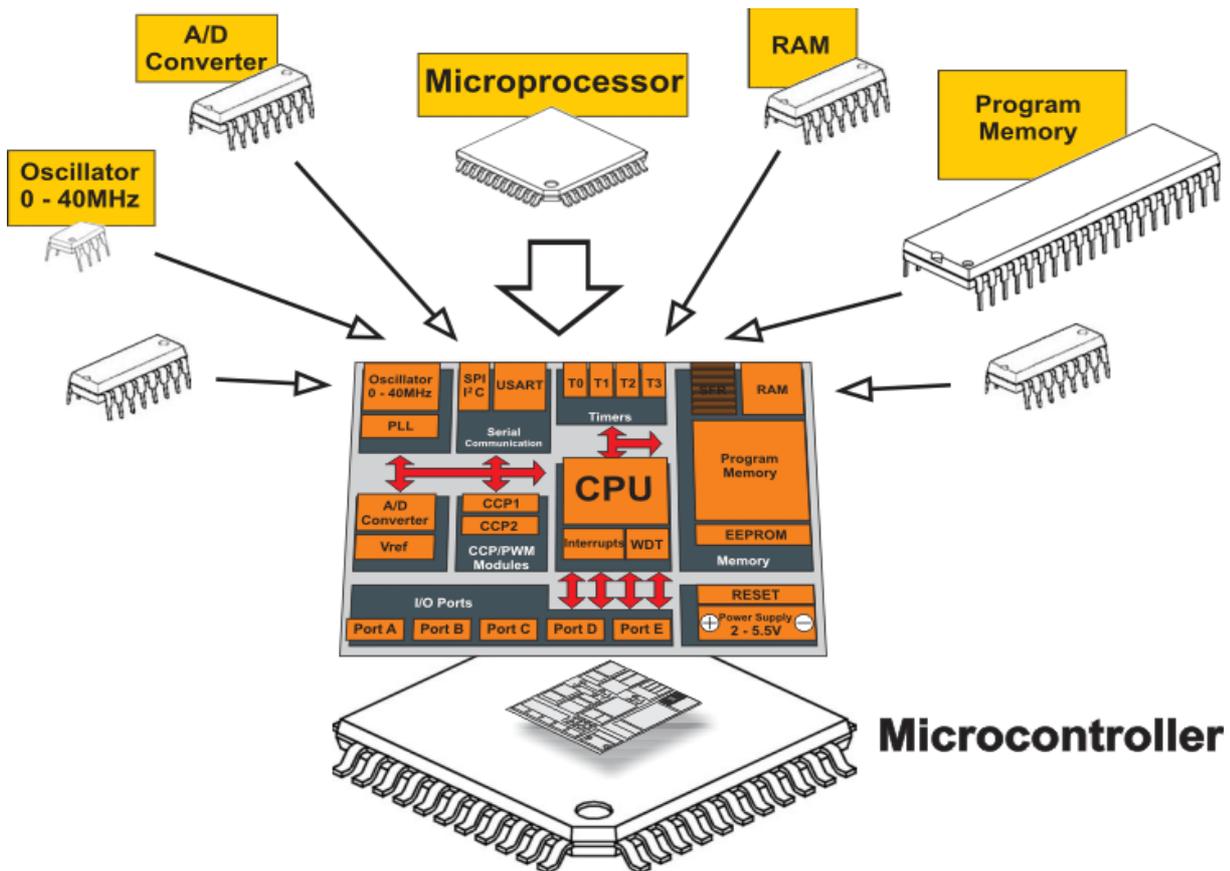
gigahertz¹) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels

2. Microprocesseur et microcontrôleur :

Il y a une différence fondamentale entre un microprocesseur et un microcontrôleur :

- le microcontrôleur intègre dans un même boîtier, un microprocesseur, de la mémoire, et des interfaces entrées/sorties.
- le microprocesseur se présente sous la forme d'un boîtier qui nécessite des éléments externes, comme de la mémoire et des circuits d'interfaces.

Ils sont présents dans la plupart des systèmes électroniques embarqués ou dédiés à une application unique (exemple : téléphone portable).



Un microcontrôleur est donc un composant autonome, capable d'exécuter le programme contenu dans sa mémoire morte dès qu'il est mis sous tension. Selon les modèles et les conditions de fonctionnement

3. Architecture interne :

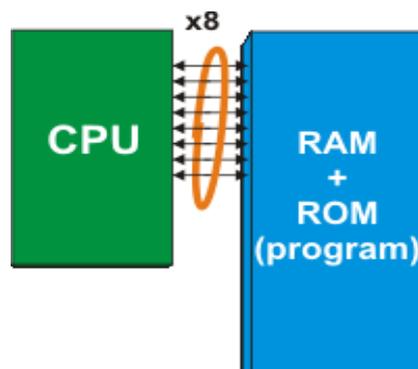
Tous les microcontrôleurs utilisent l'une des 2 architectures nommées Harvard et Von Neumann.

Elles représentent les différentes manières d'échange de données entre le CPU (microprocesseur interne) et la mémoire.

3.1-Architecture Von Neumann :

L'architecture VON NEUMANN employée par la plupart des microcontrôleurs actuels (INTEL80XX, Motorola HC05, HC08 et HC11, ou ZILOG Z80) est basée sur un bus de données unique. Celui-ci véhicule les instructions et les données.

1 bloc mémoire et 1 bus de données sur 8 bits (1 octet). Toutes les



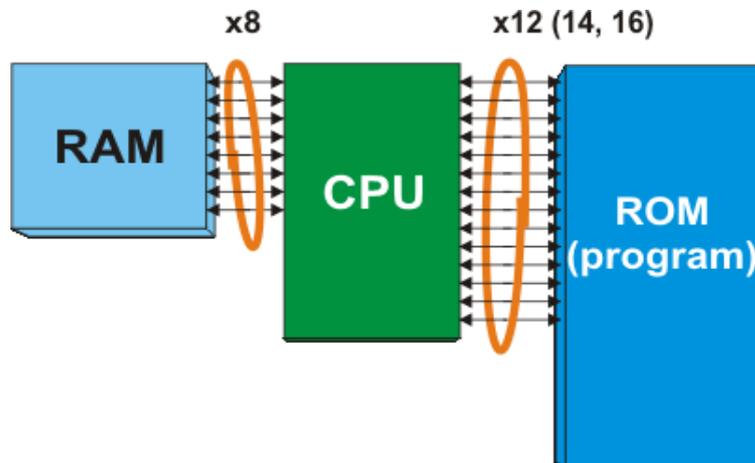
données sont échangées sur ce bus qui, surchargé, rend la communication très lente.

3.2-Architecture de Harvard :

L'architecture HARVARD utilisée par les microcontrôleurs PIC est basée sur deux bus de données. Un bus est utilisé pour les

données et un autre pour les instructions.

1 blocs mémoire distincts et 2 bus différents : 1 bus 8 bits pour communiquer avec la RAM, 1 bus 14 bits pour communiquer avec la ROM, qui contient le programme.



Le CPU peut lire une instruction (en ROM) et accéder à la mémoire de données (en RAM) en même temps.

4. Jeu d'instructions

Toutes les instructions compréhensibles par les microcontrôleurs forment ce que l'on appelle le jeu d'instructions.

On distingue 2 types

- composants dits RISC (Reduced Instructions Set Computer).

Dans ce cas, le microcontrôleur reconnaît et exécute seulement des opérations simples (addition, soustraction, etc.) et des opérations plus complexes sont réalisées en les combinant.

- Composants dits CISC (Complex Instructions Set Computer).

Les microcontrôleurs de ce type possèdent 200 instructions différentes et peuvent réaliser de nombreuses opérations à grande vitesse. Les instructions sont plus complexes.

5. Les éléments de choix d'un μ C

5.1. Architecture

- ALU (8, 16, 32, 64 bits)
- Structure du processeur (Harvard, Von Neumann)
- Type de processeur (RISC, CISC)
- Taille des mémoires programme et donnée
- Nombre de ports d'entrée/sortie

5.2. Fonctionnalité

- Fonctions *analogiques* : CAN, CNA, Comparateur, etc.
- Fonctions de *timing* : Timer, Watchdog, etc.
- Fonctions de *communication* : UART (Communication série), USB, I2C, etc.
- Facilité de programmation : In-Circuit Serial Programming, Self Programming, etc.

5.3. Fréquence d'horloge

- Tensions d'alimentation
- Consommation d'énergie, modes faible consommation d'énergie, etc.

5.4. Caractéristique physiques :

5.4.1. Type de boîtier : DIL, PLCC, etc.

6. LA PROGRAMMATION :

Le microcontrôleur exécute le programme chargé dans sa mémoire FLASH. Les mots binaires sont considérés par le CPU comme une commande.

Mais programmer en binaire ou même en hexadécimal est impossible. Les programmeurs utilisent des langages comme l'assembleur ou le C, qui utilise des abréviations.

Le passage de ce code plus lisible en code binaire prêt à être mis en mémoire s'appelle la compilation.

Remarque :

- L'assembleur est un langage bas niveau étroitement lié au type de microcontrôleur.
- Le C est un langage de plus haut niveau que l'assembleur et permet en théorie d'être portable, donc de s'adapter à n'importe quel microcontrôleur.

Ses modèles :

Il en existe de nombreux modèles :

- 68HC11 de Motorola,
- 8051 de Intel,
- et les PIC de Microchip.
- les ARM qui sont très utilisés en électronique embarquée
- Atmel AVR (utilisée par des cartes Wiring et Arduino)

Généralités sur les AVR :

AVR est une famille de microcontrôleurs développée depuis 1996 par Atmel , acquise par Microchip Technology en 2016. Il s'agit de microcontrôleurs monopuce RISC 8 bits à architecture modifiée de Harvard . AVR a été l'une des premières familles de microcontrôleurs à utiliser la mémoire flash sur puce pour le stockage de programmes, par opposition à la ROM programmable unique , l' EPROM ou l' EEPROM utilisée par d'autres microcontrôleurs à l'époque.

Spécifications

Le microcontrôleur basé sur RISC AVR 8 bits d' Atmel combine :

- 32 Ko de mémoire flash ISP avec des capacités de lecture pendant l'écriture,
- 1 Ko d'EEPROM , 2 Ko de SRAM

- 23 lignes d'E / S à usage général,
- 32 registres de travail à usage général,
- Trois temporisateurs flexibles / compteurs avec modes de comparaison,
- Interruptions internes et externes, USART programmable en série,
- interface série à 2 fils orientée octets, port série SPI,
- convertisseur A / N à 6 canaux à 10 bits (8 canaux dans les packages TQFP et QFN / MLF),
- programmable horloge de surveillance avec oscillateur interne et cinq modes d'économie d'énergie sélectionnables par logiciel.

L'appareil fonctionne entre 1,8 et 5,5 volts. L'appareil atteint un débit approchant 1 MIPS par MHz.

Familles de base

megaAVR - la série ATmega

- Mémoire de programme de 4 à 256 Ko
- Boîtier 28–100 broches
- Ensemble d'instructions étendu (multipliez les instructions et les instructions pour gérer des mémoires de programme plus importantes)
- Ensemble périphérique étendu

*L' ATmega328 est un microcontrôleur monopuce créé par Atmel dans la famille megaAVR

Applications

Depuis 2013, l'ATmega328 est couramment utilisé dans de nombreux projets et systèmes autonomes où un microcontrôleur simple, de faible puissance et à faible coût est nécessaire. L'implémentation la plus courante de cette puce se trouve peut-être sur la plate-forme de développement populaire à savoir le modèle arduino UNO.

