IUT de Poitiers – Site de Châtellerault

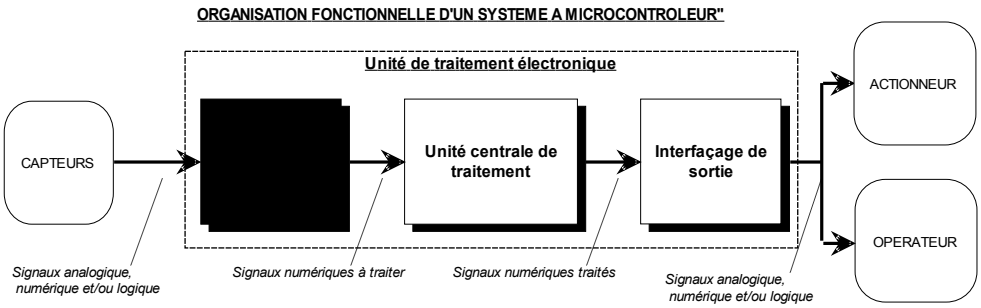
Départements
Mesures Physiques
Réseaux et Télécommunications

Formation Electronique Numérique 3

Généralité Micro-Contrôleur
Cours 2

LE MICROCONTRÔLEUR

1- Mise en situation.



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.

Un microcontrôleur se présente sous la forme d'un circuit intégré réunissant tous les éléments d'une structure à base de microprocesseur. Voici généralement ce que l'on trouve à l'intérieur d'un tel composant :

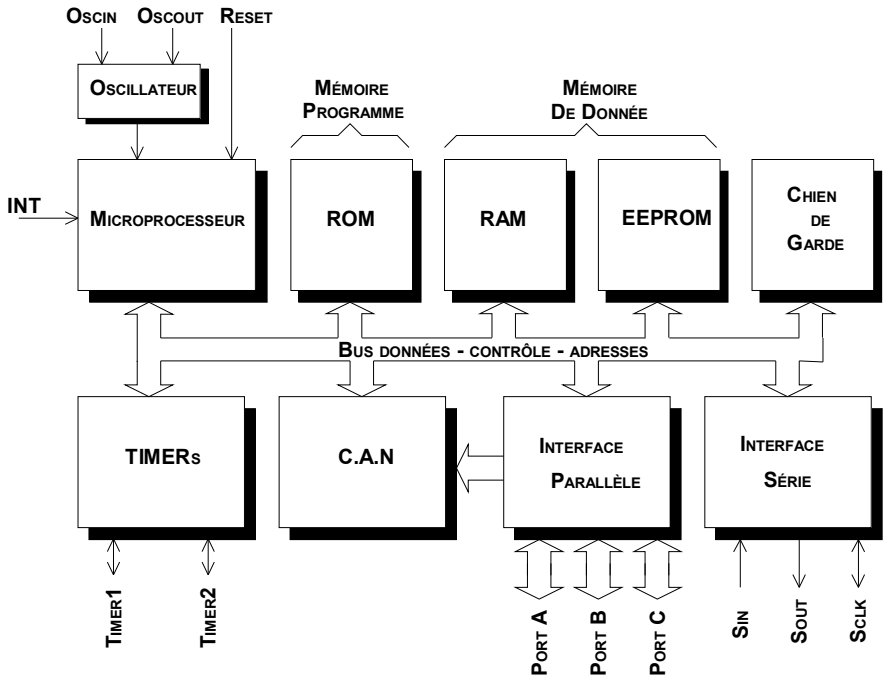
- Un microprocesseur (C.P.U.),
- De la mémoire de donnée (RAM et EEPROM),
- De la mémoire programme (ROM, OTPROM, UVPROM ou EEPROM),
- Des interfaces parallèles pour la connexion des entrées / sorties,
- Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,
- Des timers pour générer ou mesurer des signaux avec une grande précision
- Des convertisseurs analogique / numérique pour le traitement de signaux analogiques.

➢ **Avantages :**

Ÿ Encombrement réduit,
Ÿ Circuit imprimé peu complexe,
Ÿ Faible consommation,
Ÿ Coût réduit.

Ÿ

➢ **Inconvénient :**
Ÿ Système de développement onéreux,
Ÿ Programmation nécessitant un matériel adapté.



Le schéma fonctionnel précédent représente une architecture de “Von Neumann” (Commune à la plupart des microprocesseurs) où la mémoire programme partage le même bus que la mémoire de donnée. L’architecture de “Harvard”, qui dispose de bus distincts pour les données et pour le programme, est plus rarement utilisée.

2.1. C.P.U. (Microprocesseur).

Un microprocesseur exécute séquentiellement les instructions stockées dans la mémoire programme. Il est capable d’opérer sur des mots binaires dont la taille, en bits, est celle du bus des données (parfois le double pour certains microcontrôleurs). Il est généralement constitué des éléments suivants :

- Un ou plusieurs registres **accumulateurs** contenant temporairement les opérandes ainsi que les résultats des opérations,
- Des **registres auxiliaires** permettant de relayer les accumulateurs,
- Des **registres** d’**index** pour le mode d’adressage indirect,
- Un **compteur programme** pointant l’adresse de la prochaine instruction à exécuter, sa taille est celle du bus des adresses,
- Une unité arithmétique et logique (**ALU**) permettant d’effectuer des opérations entre l’accumulateur et une opérande,
- Un **registre code condition** indiquant certaines particularités en ce qui concerne le résultat de la dernière opération (retenu, zéro, interruption...).

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.

2.2 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 développeur (appelée aussi OTPROM),
- De la EPROM programmable électriquement et effaçable aux U-V (appelée aussi UVPROM),
- De la EEPROM programmable et effaçable électriquement.
-

2.3 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).

2.4 L'interface parallèle.

Ce type d'interface, répartie sur plusieurs ports (maximum 8 bits), permet de prendre en compte des états logiques appliqués en entrée (état de capteurs) ou de générer des signaux binaires en sortie (commande d'actionneurs). Les broches de ces ports peuvent donc être configurées en entrée ou en sortie, avec différentes options (résistances de rappel, sorties collecteurs ouverts, interruption...). La configuration ainsi que l'état logique de ces broches est obtenue par des opérations d'écriture ou de lecture dans différents registres associés à chaque port. On trouve généralement :

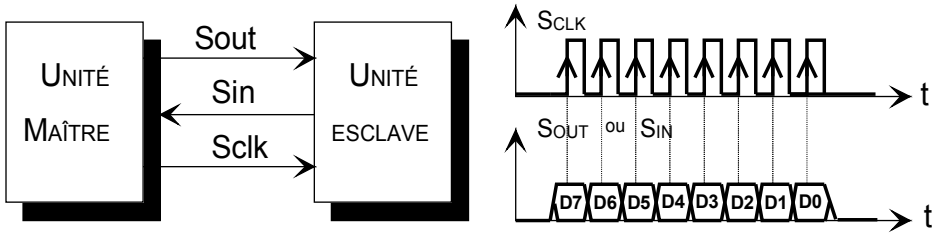
- Un registre de direction pour une configuration en entrée ou en sortie,
- Un registre de donnée recopiant les états logiques de chaquebroche de port,
- Un registre d'option permettant plusieurs configurations en entrée ou en sortie.

2.5 L'interface série.

Ce type d'interface permet au microcontrôleur de communiquer avec d'autres systèmes à base de microprocesseur. Les données envoyées ou reçues se présentes sous la forme d'un succession temporelle (sur un seul bit) de valeurs binaires images d'un mots. Il y a 2 types de liaison série : synchrone et asynchrone.

➢ Liaison série synchrone.

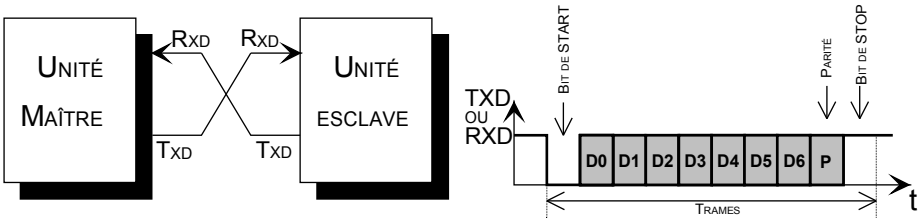
Dans ce dispositif la transmission est synchronisé par un signal d'horloge émis par l'unité maître.



➢ Liaison série asynchrone.

Ce dispositif ne possède pas de signal d'horloge de synchronisation. Les unités en liaison possèdent chacune une horloge interne cadencée à la même fréquence. Lorsqu'une unité veut

émettre un mot binaire, elle génère un front descendant sur sa ligne émettrice. A la fin de l'émission de ce mot, la ligne repasse au niveau haut. La donnée à transmettre peut contenir un bit supplémentaire appelé "parité" et servant à la correction d'erreurs.



PARAMÈTRES RENTRANT EN JEU POUR LA NORME RS232 :

- **Longueur des mots :** 7 bits (ex : caractère ascii) ou 8 bits
- **La vitesse de transmission :** elle est défini en bits par seconde ou bauds. Elle peut prendre des valeurs allant de 110 à 115 200 bds.

- **Parité :** le mot transmis peut être suivi ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission.
- **Bit de start :** la ligne au repos est à l'état logique 1 pour indiquer qu'un mot va être transmis la ligne passe à l'état bas avant de commencer le transfert. Ce bit permet de synchroniser l'horloge du récepteur.
- **Bit de stop :** après la transmission, la ligne est positionnée au repos pendant 1, 2 ou 1,5 périodes d'horloge selon le nombre de bits de stop.
- **Niveau de tension :** Un "0" logique est matérialisé par une tension comprise entre 3 et 25V, un "1" par une tension comprise entre -25 et -3 V. Des circuits spécialisés comme le MAX 232 réalise la conversion à partir de niveau TTL.

2.6 Le CAN.

Le CAN intégré dans les microcontrôleurs est généralement du type "Approximations successives". Il possède plusieurs entrées multiplexées accessibles via les broches des ports de l'interface parallèle. Le CAN possède normalement 2 registres :

- Un registre de données contenant le résultat de la conversion,
- Un registre de contrôle permettant de lancer et de surveiller la conversion.

2.7 Le timer.

Le Timer permettent de réaliser les fonctions suivantes :

- Génération d'un signal périodique modulé ou non en largeur d'impulsion,
- Génération d'une impulsion calibrée,
- Temporisation,
- Comptage d'événements.

Plusieurs registres associés au Timer permettent de configurer les différents modes décrits précédemment.

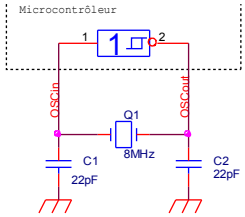
2.8 Le chien de garde.

Ce dispositif est un système anti-plantage du microcontrôleur. Il s'assure qu'il n'y ait pas d'exécution prolongé d'une même suite d'instruction.

Un compteur préchargeable se décrémente régulière au rythme de la fréquence d'horloge. Si aucun préchargement n'est effectué avant qu'il n'atteigne la valeur "0" un Reset est généré relançant ainsi le microcontrôleur. Il faut donc penser à précharger régulièrement ce chien de garde par programme lorsqu'il est activé.

2.9 Les signaux d'horloge.

Le signal d'horloge permet de cadencer le fonctionnement du microcontrôleur. Ce dernier intègre généralement une porte Trigger de Schmitt afin de réaliser un oscillateur. Pour l'obtenir on place un quartz entre les deux broches "OscIn" et "OscOut" comme l'indique le schéma suivant :



3- Mode de fonctionnement.

Le microprocesseur exécute séquentiellement les instructions codées en binaire et présentent dans la mémoire programme. L'initialisation de cette séquence peut se faire de différentes manières selon le mode de fonctionnement.

3.1 Le fonctionnement en interruptions.

Le microcontrôleur, dans son environnement, est destiné à traiter des informations en " temps réel ". L'application est couplée au monde extérieur, par l'échange fréquent de messages et de signaux à des instants prévus. Il est dans l'obligation de changer d'état en fonction des priorités relatives de l'opération en cours et de celle qui lui est demandé. Il interrompt ou non le déroulement normal du programme en fonction d'une demande externe.

Celles-ci sont vues du microcontrôleur comme des demandes d'interruption. On note deux types d'interruption :
- NMI (No Maskable Interrupt) : interruption non - masquable,
- IRQ (Interrupt Request) : interruption masquable.

Quelque soit l'entrée d'interruption activée, le microprocesseur réalise des tâches identiques :
- dans tout les cas, le programme principal est interrompu ;
- le processeur doit sauvegarder le contenu du PC dans la pile ;
- le processeur exécute une séquence privilégiée, reflet du type de traitement d'interruption ;

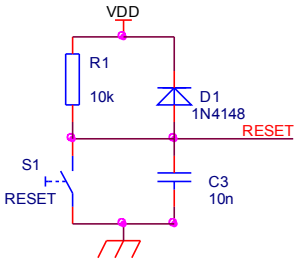
- la prise en compte d'une interruption ne se fait jamais pendant l'exécution d'une instruction.

En plus des deux interruptions définies précédemment, le ST6 possède une entrée d'initialisation.

3.2 Initialisation : $\overline{\text{RESET}}$.

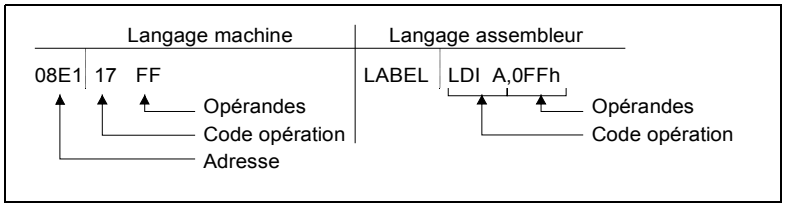
Un niveau bas sur cette entrée entraîne une réinitialisation complète du microprocesseur : l'instruction en cours est arrêtée. D'une façon générale ce signal est activée à la mise sous tension. Un bouton poussoir est souvent rajouté afin qu'une réinitialisation manuelle soit possible.

Lorsque le signal de "RESET" est activé, tous les registres sont initialisé et le compteur programme se place à une adresse spécifique appelée "Vecteur de RESET".



3.3 Instructions et modes d'adressages.

Les instructions contenues dans la mémoire programme sont une suite de mots binaires décodés puis exécutés par le microprocesseur, appelée langage machine. Ces codes sont difficilement compréhensibles par le programmateur. C'est la raison pour laquelle ils sont traduits en différents mots faisant partis du langage assembleur. Le codage d'une instruction s'effectue de la façon suivante :



Les modes d'adressages sont les différents moyens qui permettent au microprocesseur d'accéder à une opérande en vue de tester ou de modifier le contenu d'un registre ou d'une mémoire.

- **Mode d'adressage inhérent ou implicite.**

  L'adressage inhérent concerne les instructions qui ne comportent pas d'opérande, cette dernière étant implicite. Il s'agit généralement des opérations de mise à 0 et d'incrémentation ou de décalage de bits

- **Mode d'adressage immédiat.**

  Ce mode d'adressage permet de charger les registres internes du microprocesseur directement avec la valeur de l'opérande.

- **Mode d'adressage direct.**

  Dans ce mode d'adressage l'opérande correspond à une adresse où est située la donnée de l'opération .

- **Mode d'adressage indexé ou indirect.**

  Ce mode d'adressage s'applique aux registres d'index. Ces derniers contiennent une adresse mémoire dans laquelle est placée la donnée de l'opération.

- **Mode d'adressage relatif.**

  Ce mode d'adressage est réservé pour les instructions de rupture de séquence conditionnel. La condition provient généralement du résultat de l'opération précédente (résultat nul, ayant entraîné une retenue ...) où de l'état d'un bit.

- **Mode d'adressage étendu.**

  Ce mode d'adressage permet d'effectué des ruptures de séquence sans condition afin d'atteindre une adresse non successive dans la mémoire programme.

4- 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'algorigramme. 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".

4.1 L'algorithme.

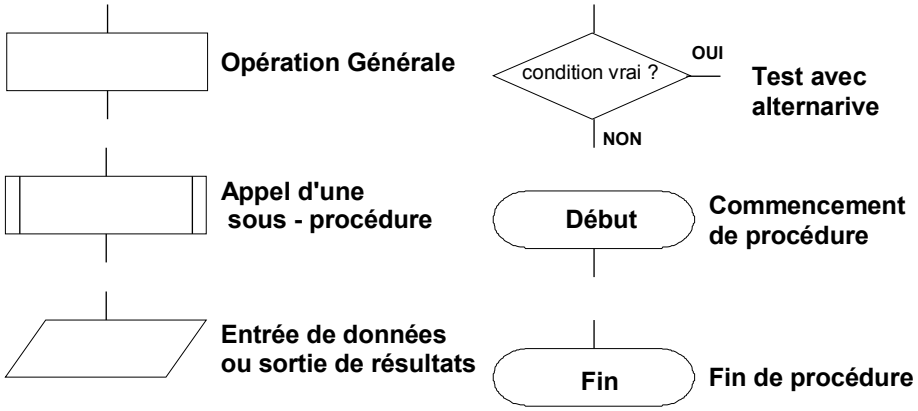
Définition : série des actes ou opérations élémentaires, qu'il faut exécuter en séquence pour accomplir une tache quelconque, en suivant un enchaînement strict.

Représentation normalisée : l'algorithme s'établit par une succession de phrases simples.

4.2 L'algorigramme

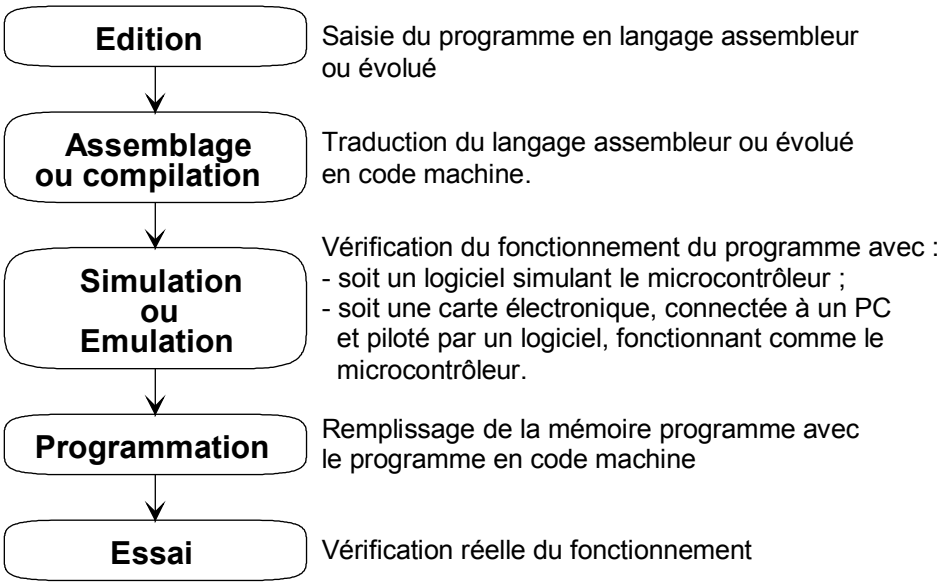
Définition : il s'agit d'une représentation graphique et normalisée de l'algorithme.

Représentation normalisée : il s'agit de dessiner une suite de symboles définit comme il suit :



4.3 Développement du programme et mise au point.

Que ce soit dans un langage assembleur ou évolué, l'écriture du programme ainsi que sa mise au point doivent suivre le diagramme suivant :



L'émulateur est un dispositif (assez onéreux) qui permet de remplacer le microcontrôleur afin d'effectuer la vérification et la mise au point du programme. C'est une carte électronique comportant, d'un coté un connecteur compatible avec le boîtier du microcontrôleur qu'il émule, et de l'autre, une connexion de type RS232 (ou autre) reliée à micro-ordinateur.

La mise au point peut alors se faire en pas à pas ou avec des points d'arrêt permettant ainsi de cerner très rapidement les "bugs" de certaines routines.

Le simulateur , beaucoup moins coûteux, permet la même chose mais de façon beaucoup moins efficace car il ne tient pas ou très peu compte de l'environnement du microcontrôleur

4.4 Structure d'un programme.

La saisie d'un programme, que ça soit en langage assembleur ou évolué, doit suivre la structure suivante.

1- Directives d'assemblage ou de compilation

2- Déclaration des constantes

3- Déclaration des variables

4- Sous-programmes

5- Programme principal

6- Programme d'interruption

- *Les directives indiquent généralement les liens vers d'autres fichiers contenant des définitions de registres ou de sous-programmes.*
- *Les constantes correspondent généralement à des adresses mémoires.*
- *Les variables sont des zones de la mémoire de données.*

5- Exemples de microcontrôleurs.

Référence	Fabricant	Vitesse	RAM	ROM / EPROM / FLASH	EEPROM	E / S Logiques	Timer	Entrées Analogiques	Particularité
8051	Intel	12 Mhz	128 o	4 Ko	X	32	2	0	
16C71	Microchip	20 Mhz	36 o	1Kx14	X	13	1	4	RISC
6805 S2	Motorola	4 MHz	64	1 Ko	X	16	2	8	
68HC11 A1	Motorola	8 MHz	256 o	X	512	22	1	8	Etendu
AT90S 8515	Atmel	20 MHz	512 o	4 Ko	512	32	3	8	RISC
ST 6265	Thomson	8 MHz	128 o	4 Ko	64 o	21	2	13	