

Généralités sur les microprocesseurs et les microcontrôleurs

Plan

- I. Les microprocesseurs
 - I.1 Architecture interne
 - I.1.1 l'unité de commande
 - I.1.2 l'unité de traitement
 - I.2 traitement des instructions
 - I.2.1 Principes
 - I.2.2 Modes d'adressage
- II. Les microcontrôleurs
 - II.1 Présentation
 - II.2 constituants d'un microcontrôleur
 - II.3 Programmation

Les systèmes à base de processeur sont des systèmes numériques complexes. Ils sont construits et organisés autour d'un microprocesseur (microcontrôleur).

Ils équipent les micro-ordinateurs, les téléphones, les appareils électroménagers, les appareils hi-fi, les voitures, les avions, etc. Dans l'industrie, ils sont intégrés dans les machines de production et de fabrication comme les robots industriels, les machines à outils (fraiseuse, perceuse, etc.).

Les appareils et les équipements deviennent de plus en plus "intelligents". Ils sont dotés d'un système de traitement d'information complexe et sophistiqué qui nécessite l'usage d'un système à microprocesseur dont la taille n'a pas cessé de diminuer.

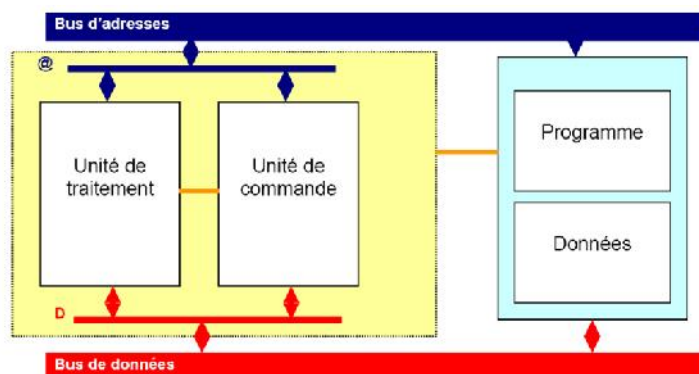


Les microprocesseurs

I.1 Architecture interne

Un microprocesseur est construit autour de deux éléments principaux :

- Une unité de commande
- Une unité de traitement



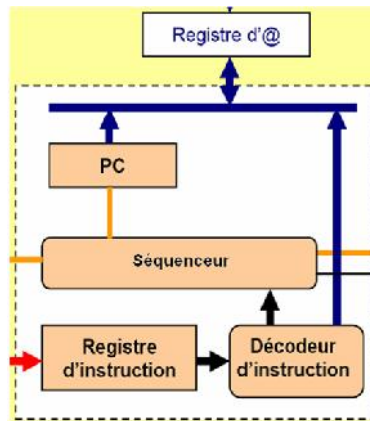
I.1.1 L'unité de commande

Elle permet de séquencer le déroulement des instructions. Elle effectue la recherche en mémoire de l'instruction, le décodage de l'instruction codée sous forme binaire. Enfin elle pilote l'exécution de l'instruction.

Les blocs de l'unité de commande :

- 1. Le compteur de programme (PC : Programme Counter) :** appelé aussi Compteur Ordinal (CO) est constitué par un registre dont le contenu est initialisé avec l'adresse de la première instruction du programme. Il contient toujours l'adresse de la prochaine instruction à exécuter.
- 2. Le registre d'instruction et le décodeur d'instruction :** Chacune des instructions à exécuter est transféré depuis la mémoire dans le registre instruction puis est décodée par le décodeur d'instruction.
- 3. Bloc logique de commande (ou séquenceur) :** Il organise l'exécution des instructions au rythme d'une horloge. Il élabore tous les signaux de synchronisation internes ou externes (bus de commande) du microprocesseur en fonction de l'instruction qu'il a à exécuter. Il s'agit d'un automate réalisé de façon micro-programmée.

L'unité de commande



I.1.2 L'unité de traitement

Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions. Les blocs de l'unité de traitement :

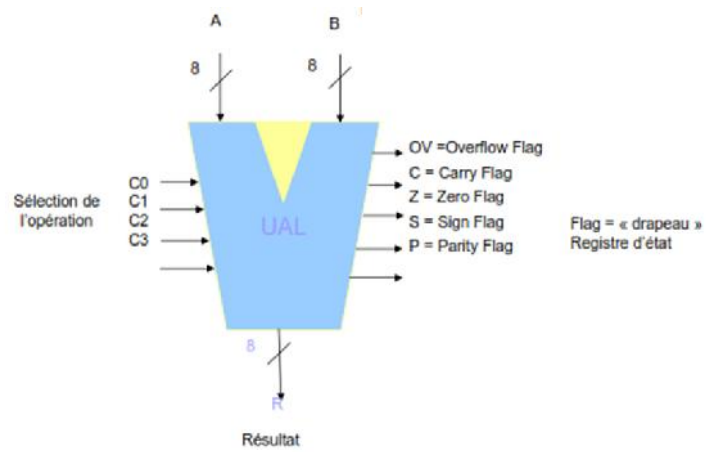
1. **Les accumulateurs sont des registres de travail** qui servent à stocker une opérande au début d'une opération arithmétique et le résultat à la fin de l'opération.
2. **L'Unité Arithmétique et Logique (UAL)** est un circuit complexe qui assure les fonctions logiques (ET, OU, Comparaison, Décalage, etc...) ou arithmétique (Addition, soustraction...).
3. **Le registre d'état** est généralement composé de 8 bits à considérer individuellement. Chacun de ces bits est un indicateur dont l'état dépend du résultat de la dernière opération effectuée par l'UAL. On les appelle *indicateur d'état ou flag ou drapeaux*. Dans un programme le résultat du test de leur état conditionne souvent le déroulement de la suite du programme.

On peut citer par exemple les indicateurs de :

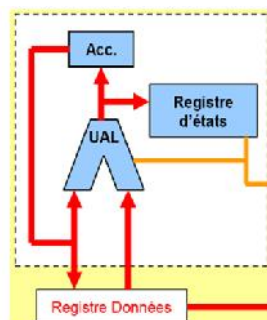
- Retenue (**carry** : C)
- Débordement (**overflow** : OV ou V)
- Zéro (Z)

...

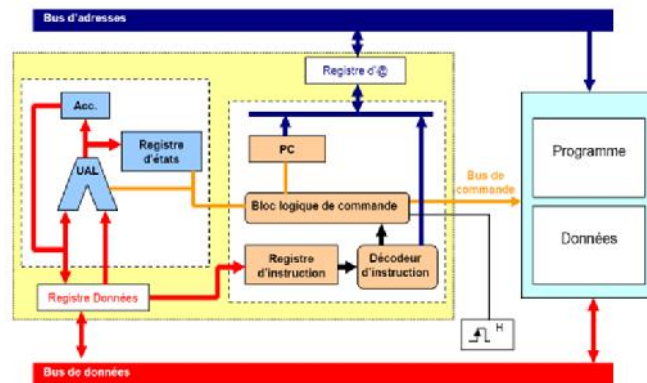
Unité Arithmétique et Logique



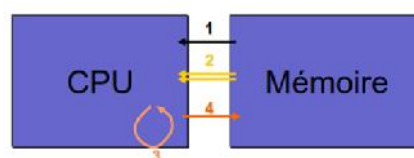
L'unité de traitement



Architecture complète

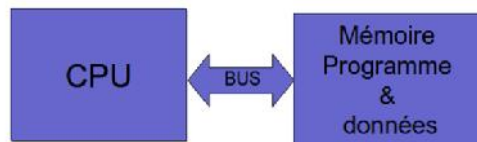


Déroulement d'une opération de calcul



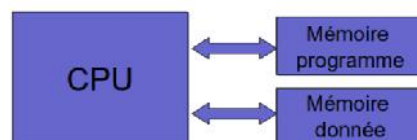
- (1) Charger une instruction depuis la mémoire
- (2) Charger les opérandes depuis la mémoire
- (3) Effectuer les calculs
- (4) Stocker le résultat en mémoire

Architecture Von NEUMAN



- Un seul chemin d'accès à la mémoire
- Un bus de données (programme et données),
- Un bus d'adresse (programme et données)
- Architecture des processeurs d'usage général
- Goulot d'étranglement pour l'accès à la mémoire

Architecture HAVARD



- Séparation des mémoires programme et données
- Un bus de données programme,
- Un bus de données pour les données,
- Un bus d'adresse programme,
- Un bus d'adresse pour les données.
- Meilleure utilisation du CPU :
- Chargement du programme et des données en parallèle

I.2 Traitement des instructions

I.2.1 Principe

Le microprocesseur ne comprend qu'un certain nombre d'instructions qui sont codées en binaire. Une instruction est composée de deux éléments :

Le code opératoire : C'est un code binaire qui correspond à l'action à effectuer par le processeur

Le champ opérande : Donnée ou bien adresse de la donnée.

La taille d'une instruction peut varier, elle est généralement de quelques octets (1 à 8), elle dépend également de l'architecture du processeur.

- **Instruction Addition :**

Accumulateur = Accumulateur + Opérande

Correspond à l'instruction ADD A,#2

Instruction (16 bits)	
Code opératoire (5 bits)	Champ opérande (11 bits)
ADD A	#2
11001	000 0000 0010

Cette instruction est comprise par le processeur par le mot binaire :

11001 000 0000 0010 = code machine

Phase 1: recherche de l'instruction en mémoire

La valeur du PC est placée sur le bus d'adresse par l'unité de commande qui émet un ordre de lecture.

Après le temps d'accès à la mémoire, le contenu de la case mémoire sélectionnée est disponible sur le bus des données.

L'instruction est stockée dans le registre d'instruction du processeur.

Phase 2: décodage et recherche de l'opérande

L'unité de commande transforme l'instruction en une suite de commandes élémentaires nécessaires au traitement de l'instruction.

Si l'instruction nécessite une donnée en provenance de la mémoire, l'unité de commande récupère sa valeur sur le bus de données.

L'opérande est stocké dans le registre de données.

Phase 3: traitement de l'instruction

Le séquenceur réalise l'instruction.

Les drapeaux sont positionnés (registre d'état).

L'unité de commande positionne le PC pour l'instruction suivante.

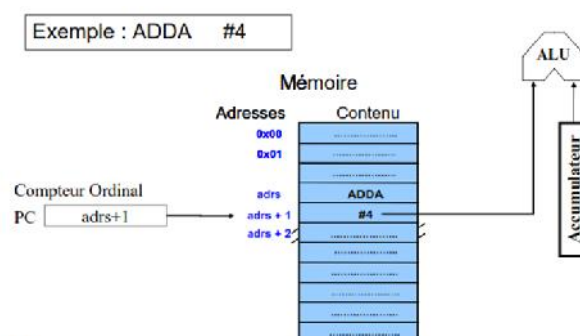
I.2 Traitement des instructions

I.2.2 Modes d'adressage

Ce sont les diverses manières de définir la localisation d'un opérande. Les trois modes d'adressage les plus courants sont :

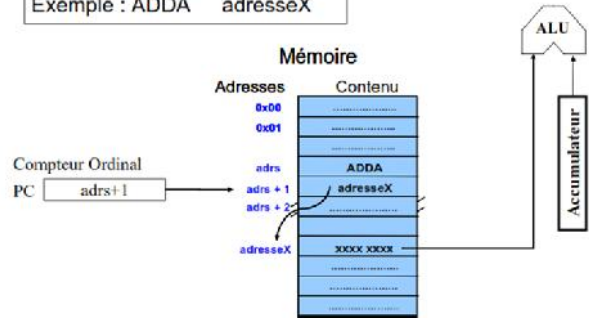
- **Adressage immédiat**
- **Adressage direct**
- **Adressage indirect**

Adressage immédiat



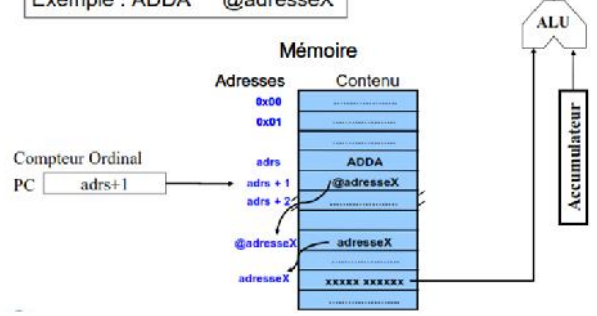
Adressage direct

Exemple : ADDA adresseX



Adressage indirect

Exemple : ADDA @adresseX



II. Les microcontrôleurs

II.1 Présentation

Un microcontrôleur est un circuit qui intègre un maximum de fonctions dans un même boîtier. L'intégration de ces fonctions dans le même environnement permet de créer des applications plus simplement.

La structure interne d'un microcontrôleur comporte typiquement :

- Une unité de calcul et de commande
- Mémoire ROM
- Mémoire RAM
- Un contrôleur d'interruption
- Un compteur/temporisateur (timer)
- Des entrées/sorties parallèles (ports)
- Un UART (port série)

Il peut aussi posséder :

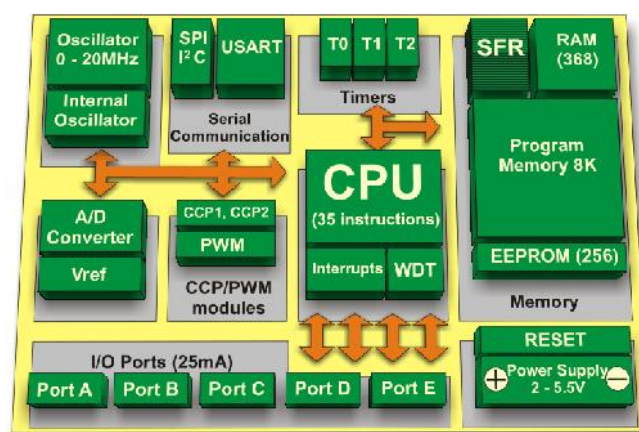
- Un Watchdog : (surveillance du programme)
- Une sortie PWM (modulation d'impulsion)
- Un CAN/CNA (Convertisseur analogique numérique)
- Un interface I²C, CAN...

Avantages des microcontrôleurs

- Diminution de l'encombrement du matériel et du circuit imprimé
- Simplification du tracé du circuit imprimé (plus besoin de tracer de bus !)
- Augmentation de la fiabilité du système nombre de composants connexions composants/supports et composant circuit imprimé
- Intégration en technologie MOS, CMOS, ou HCMOS diminution de la consommation
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux:
 - moins cher que les composants qu'il remplace
 - Diminution des coûts de main d'œuvre (conception et montage)
 - Environnement de programmation et de simulation évolués

II.2 Constituants d'un microcontrôleur

Structure interne d'un microcontrôleur



II.2.1 Les mémoires

ROM memory :

Aussi appelé (à juste titre) program memory. C'est une mémoire Flash qui contient le programme à exécuter.

EEPROM memory:

C'est une mémoire similaire à la mémoire programme. En revanche, le contenu peut être modifié en cours d'utilisation de l'application.

RAM memory :

General Purpose Register : Mémoire RAM classique, utiliser pour stocké des variables. **Exemple :**

```
int i;
```

```
i++; // incrémentation de i depuis la RAM
```

SFR (Special Function Register) : C'est aussi de la mémoire RAM, sauf que les rôles de chacune des cases mémoire (registres) ont été définis par le fabriquant. Chaque registre SFR est connecté à un périphérique matériel spécifique et permet de la contrôler.

Exemple :

ADCON0 register (adresse 9Fh) permet de piloter le convertisseur A/D

II.2.2 Les oscillateurs

Quartz

Très bonne Stabilité (10 ppm)

Résonateur céramique

Stabilité (0.5%)

Moins couteux que le quartz

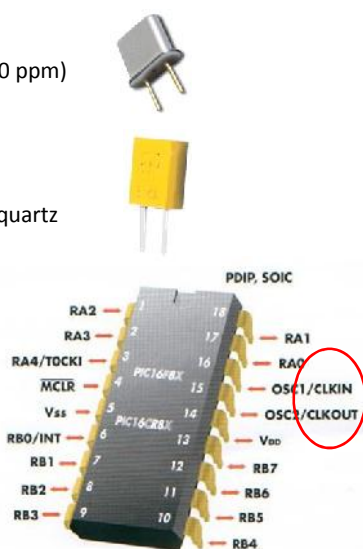


Figure 3

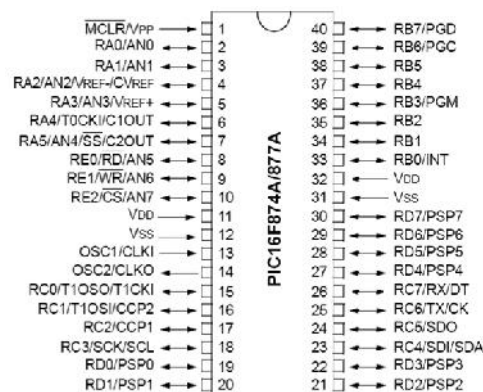
II.2.3 Les Timers

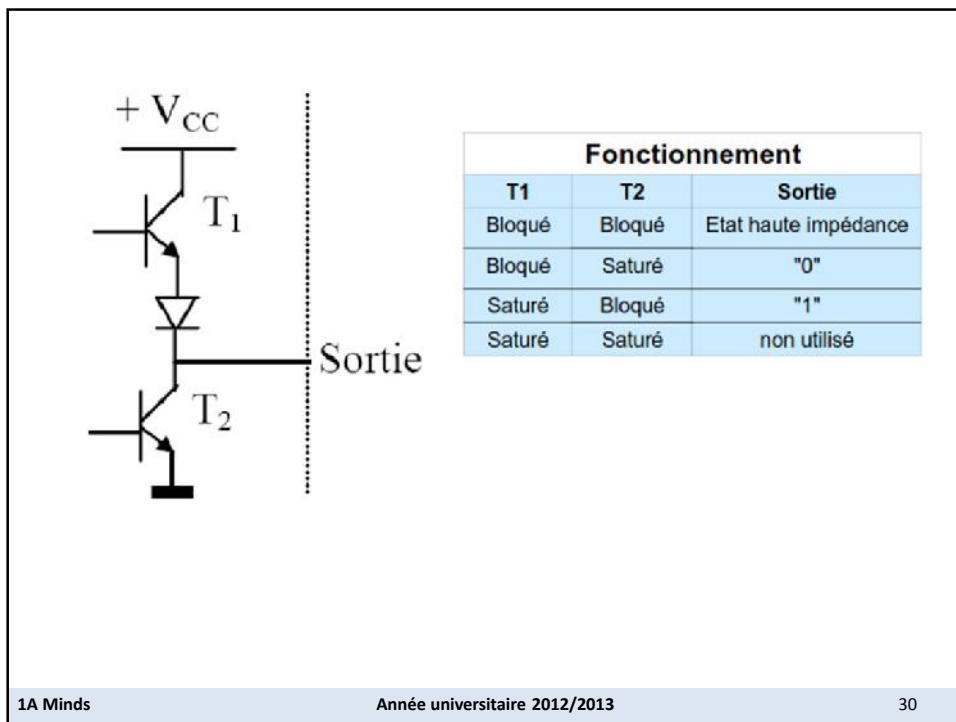
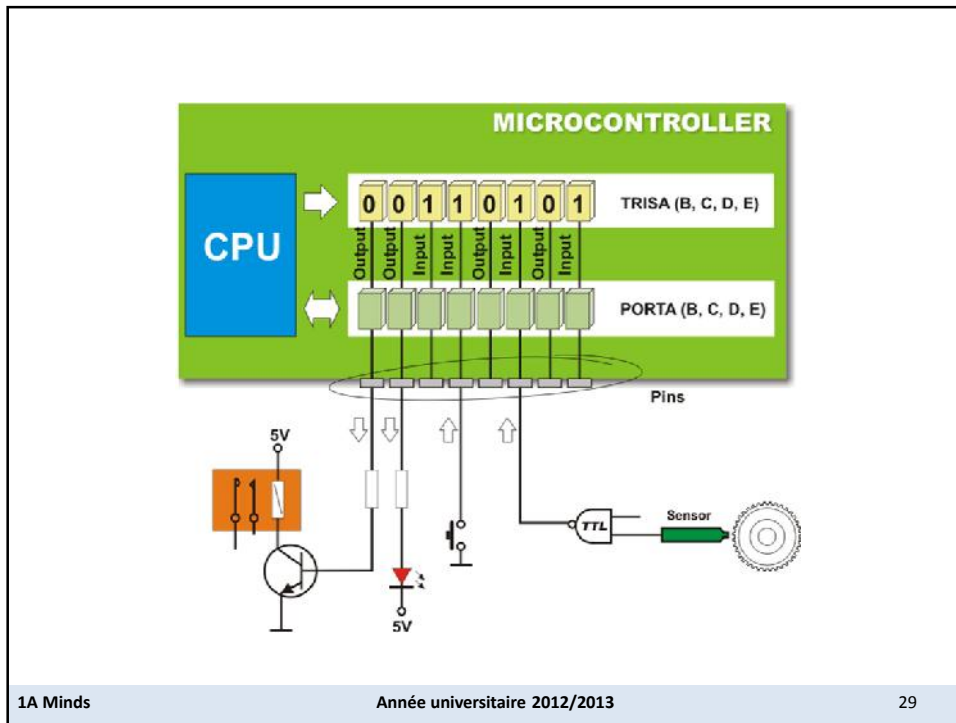
Un timer est le nom courant de compteur /temporisateur . Il sert à :

- Mesurer du temps (compter le nombre de coup d'horloge) > **Mode temporisateur**
- Compter le nombre d'évènement sur une broche (exemple : Nombre d'appuis sur un bouton poussoir > **Mode compteur**)

II.2.4 Les ports d'E/S

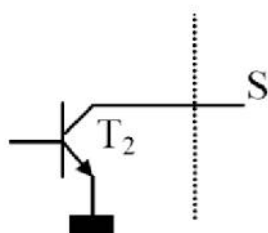
Ce sont des ports parallèles. Ils permettent de recueillir des informations ou de piloter des modules sur l'environnement extérieur. Ils sont souvent bidirectionnels (configurable en entrée ou sortie).





Pour générer le niveau "1", une résistance extérieure est nécessaire (résistance de tirage // pull-up). Plusieurs sorties "collecteur ouvert" peuvent être reliées entre elles, cela réalise un "ET logique"

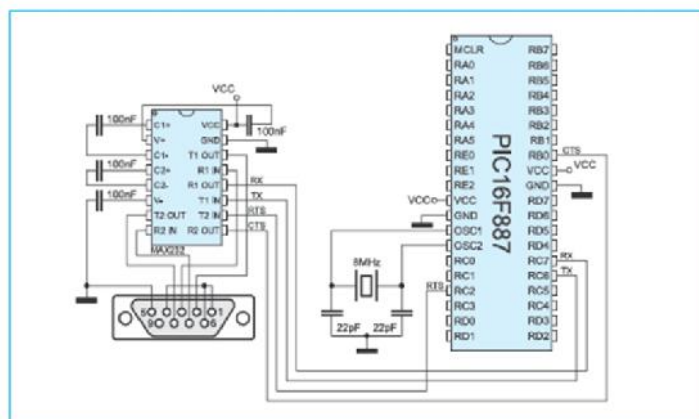
Une sortie « collecteur ouvert » peut commander une charge sous une tension différente de la tension d'alimentation.



Fonctionnement	
T2	S
Saturé	"0"
Bloqué	Dépend du montage

II.2.5 Le ports série

La liaison série USART (*Universal Synchronous Asynchronous Receiver transmitter*) est le mode le plus répandu pour communiquer (et aussi le plus vieux).



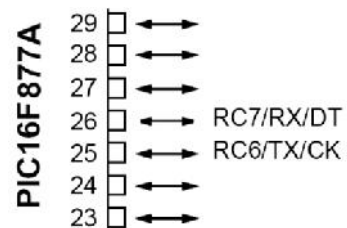
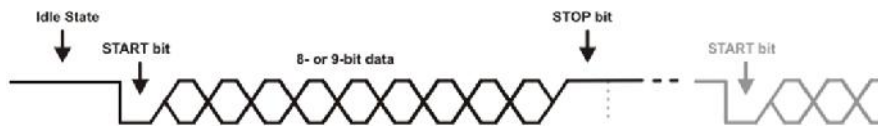
Une liaison série permet de transmettre des données sur un nombre limité de fils. Cette liaison peut être Half duplex (liaison synchrone) ou full duplex (liaison asynchrone).

Exemple :

Sur le microcontrôleur PIC16F877A, la liaison série asynchrone peut être mise en œuvre à l'aide des pins RxD et TxD.

Elle est réalisée par un USART :

- RxD, signal de réception de l'USART
- TxD, signal de transmission de l'USART

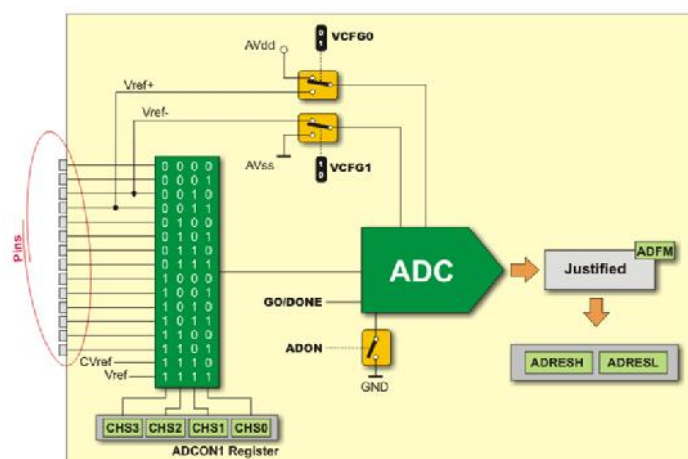


- 1^{ère} étape : Configuration de la transmission
 2^{ème} étape : Envoyer et recevoir des données

II.2.6 Le convertisseur Analogique Numérique

CAN : Dans les microcontrôleurs, les voies de conversion analogique/numérique sont souvent multiplexées. Ceci signifie que la fréquence maximale de conversion analogique numérique est divisée par le nombre de voies utilisées.

Très souvent, il faut configurer les entrées de conversion en « entrée analogique » car celles-ci peuvent aussi être utilisées en entrée numérique.



II.2.7 Les interruptions

Un système informatique n'est utile que s'il communique avec l'extérieur. L'objectif est de pouvoir prendre connaissance que le périphérique sollicite le processeur. Cette sollicitation arrive de façon totalement asynchrone.

Deux modes sont possibles :

- La scrutation (polling) permet d'interroger régulièrement les périphériques afin de savoir si une nouvelle donnée est présente.
- Une interruption permet au périphérique lui-même de faire signe au processeur de sa présence.

Scrutation (polling)

- Coûteux en temps (multiplier par le nombre de périphérique à interroger)
- Implémentation : Appel classique à une fonction dans le programme

Interruption

- Demande à l'initiative du périphérique
- Prise en compte rapide de l'évènement
- Interruption asynchrone d'un programme puis retour au même endroit à la fin du traitement

Une interruption est un arrêt temporaire de l'exécution normale d'un programme informatique par le microprocesseur afin d'exécuter un autre programme (appelé routine d'interruption).

II.3 Programmation des microcontrôleurs

