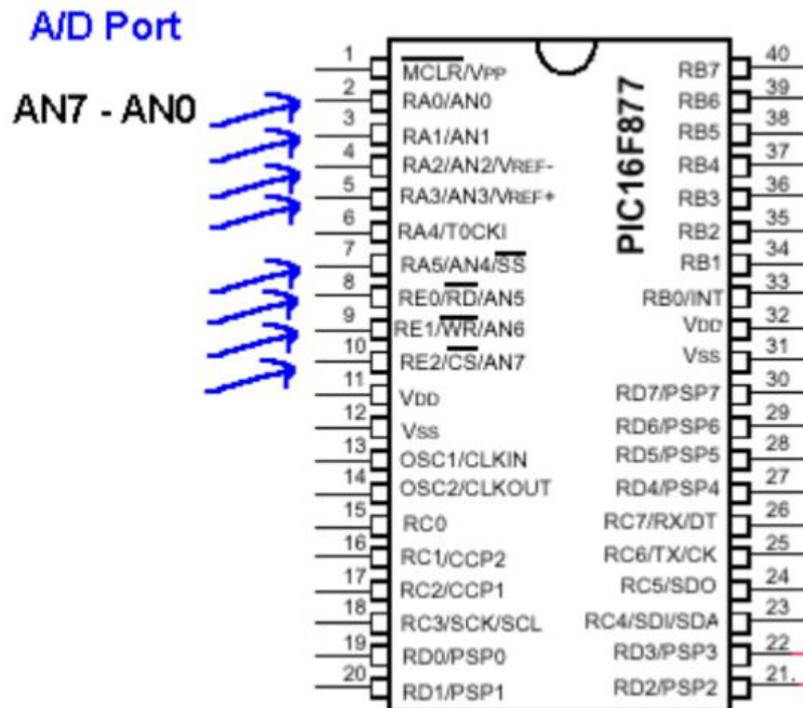


Les Convertisseurs Analogiques Numériques 16F877

Introduction :

Le convertisseur analogique numérique est à approximations successives et il possède une résolution de 10 bits. Il est composé de :

- Un multiplexeur analogique à 8 voies (pour le PIC16F877).
- Un échantillonneur bloqueur.
- Un Convertisseur Analogique Numérique de 10 bits.



Fonctionnement du convertisseur :

La conversion se passe en 2 temps :

- 1er temps le signal à convertir est appliqué sur l'entrée à convertir, ce signal doit être présent au moins pendant le temps **Tacq** (temps d'acquisition environ 20 μ S pour 5V).
- 2ème temps la conversion, approximations successives. Le temps de conversion minimum est de 12 TAD (TAD c'est le temps de conversion dépendant de l'horloge interne, typiquement 1,6 μ S).

Une conversion commence toujours par la mise à 1 du bit GO/DONE du registre ADCON0. Lorsque la conversion est terminée se bit repasse à 0. Donc pour pouvoir lire le résultat dans les registres ADRESL et ADRESH il suffit d'attendre que le bit GO/DONE passe à 0.

La valeur résultante N de la conversion ADRSH: ADRESL est égale à :

N (valeur numérisée) = $((VIN - VREF-) / (VREF+ - VREF-)) * 1023$

Si $VREF+ = VDD = 5V$ et $VREF- = VSS = 0V$ alors

N (valeur numérisée) = $1023 * (VIN / 5)$

Mais avant de réaliser une conversion il faut définir la configuration du convertisseur :

- Le nombre d'entrées analogiques.
- Le nombre d'entrées logiques.
- Le type de tension de référence :
 - Interne $VREF = VDD - VSS$.
 - Externe, soit $VREF = VREF+ - VSS$ ou $VREF = VREF+ - VREF-$.

Cette configuration se fait à travers le registre **ADCON1**

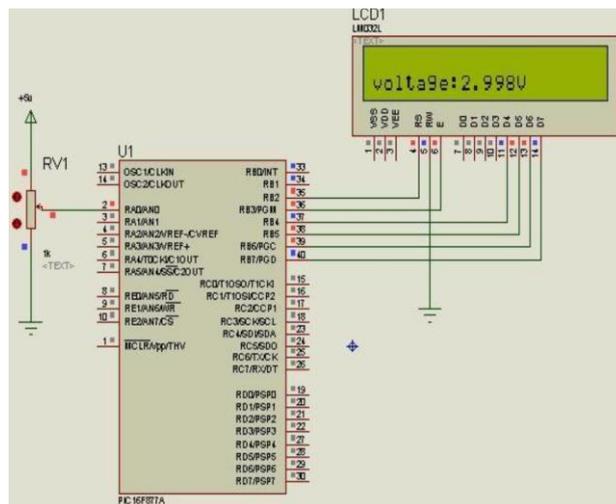
L'entrée analogique est convertie sur 10 bits et enregistrée sur deux registres **ADRESL** and **ADRESH**

- **ADRESH** : Contient les poids fort du résultat de conversion ;
- **ADRESL** : Contient les poids faibles du résultat de conversion ;
- **ADCON0** : Contrôle le registre 0
- **ADCON1** : Contrôle le registre 1

```

unsigned short a;

void main() {
    TRISB=0x00;  PORTB=0x00;
    TRISC=0x00;  PORTC=0x00;
    a=0x00;
    for (;;)
    {
        a=ADC_read(0);
        PORTB=a;
        portc= a >> 8;
    }
}
    
```



Le registre ADCON1 :

Il permet de choisir une configuration parmi les 16 proposées

| ADCON1 REGISTER (ADDRESS 9Fh) | | | | | | | |
|-------------------------------|-------|-----|-----|-------|-------|-------|-------|
| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| ADFM | ADCS2 | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | bit 0 | |

bit 7 ADFM: A/D Result Format Select bit

1 = Right justified. Les 6 poids forts du registre **ADRESH** sont lus comme étant des '0'.

0 = Left justified. Les 6 poids faible du registre **ADRESL** sont lus comme étant des '0'.

Ce bit permet de choisir entre deux types de justification pour le résultat.

bit 7-6 **ADCS1:ADCS0**: A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

| ADCON1 <ADCS2> | ADCON0 <ADCS1:ADCS0> | Clock Conversion |
|-------------------|-------------------------|---|
| 0 | 00 | Fosc/2 |
| 0 | 01 | Fosc/8 |
| 0 | 10 | Fosc/32 |
| 0 | 11 | FRC (clock derived from the internal A/D RC oscillator) |
| 1 | 00 | Fosc/4 |
| 1 | 01 | Fosc/16 |
| 1 | 10 | Fosc/64 |
| 1 | 11 | FRC (clock derived from the internal A/D RC oscillator) |

bit 5-3 **CHS2:CHS0**: Analog Channel Select bits

000 = Channel 0 (AN0)
001 = Channel 1 (AN1)
010 = Channel 2 (AN2)
011 = Channel 3 (AN3)
100 = Channel 4 (AN4)
101 = Channel 5 (AN5)
110 = Channel 6 (AN6)
111 = Channel 7 (AN7)

bit 2 GO/DONE: A/D Conversion Status bit

If ADON = 1:

1 = La conversion A/D est en court.

0 = La conversion A/D n'est pas en court.

bit 1 non implémenté : lu comme étant '0'

bit 0 ADON: A/D On bit

1 = le module de conversion A/D est activé

0 = le module de conversion A/D est désactivé et ne consomme aucun courant