

IUT de Poitiers – Site de Châtelleraut

Départements

Mesures Physiques

Réseaux et Télécommunications

Formation Electronique Numérique 3

Principe de la
Conversion Analogique Numérique

Echantillonnage, quantification, Conversion Analogique-Numérique

Dans cette partie on se propose d'étudier le spectre des signaux échantillonnés et la reconstruction analogique des signaux échantillonnés. La quantification des signaux et le bruit de quantification sont également traités.

La théorie de l'échantillonnage est à la base des transmissions numériques, la transmission numérique des signaux permet :

- d'améliorer la qualité de la transmission par une meilleure immunité aux bruits, les niveaux transmis sont des niveaux logiques '0' ou '1'.
- de réaliser des traitements mathématiques sur les signaux (filtrage, codage,...)
- de procéder à un multiplexage temporel, c'est à dire transmettre plusieurs signaux sur une même voie de transmission, cas du téléphone numérique.

Dans une première étape, on aborde l'échantillonnage et la quantification sous l'aspect traitement du signal, c'est à dire sans se soucier des moyens pratiques à mettre en œuvre pour arriver aux résultats. Les techniques de conversion analogique-numérique (CAN) sont étudiées sur une maquette électronique et le principe sera ensuite appliqué au niveau des microcontrôleurs PICS.

I - Aspects pratiques de la conversion Analogique Numérique

a. Filtrage

Le signal à échantillonner occupe une largeur spectrale finie. Pour limiter cette largeur spectrale et éviter les problèmes de « repliement de spectre », il est nécessaire de filtrer le signal à convertir à une fréquence de coupure $F_c < 0.5F_e$. En général, on choisit $F_c < 0.1 F_e$.

b. Blocage

Une fois le signal filtré et échantillonné, il reste à le quantifier. Pour pouvoir réaliser cette fonction, on doit maintenir constant la valeur à quantifier afin de permettre au CAN de traiter l'échantillon et de le numériser. On appelle cette opération, le blocage. Ce blocage doit être d'une durée supérieure au temps de conversion.

En pratique on utilise, souvent un bloqueur d'ordre zéro (un convertisseur analogique numérique est équivalent à un bloqueur d'ordre zéro).

Réaliser un échantillonneur bloqueur consiste à associer un interrupteur à une capacité comme le montre la figure 3.

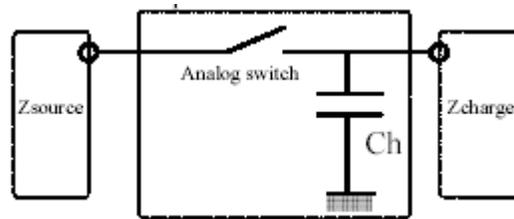


Fig 3 : Bloqueur

La capacité joue le rôle d'élément mémoire, l'interrupteur est là pour réactualiser la valeur mémorisée ou bien l'isoler vis à vis de l'entrée.

Dans le cas idéal :

Interrupteur fermé : $V_{eb}=V_e$

La sortie V_{eb} suit les variations de l'entrée V_e . On transmet directement l'entrée sur la sortie. On dit que l'on est en phase d'échantillonnage (Sample).

Interrupteur ouvert :

$$V_{eb}=Cste$$

La sortie reste constante et égale à la dernière valeur transmise du signal d'entrée. On dit que l'on est en phase de blocage (Hold). La figure suivante montre l'évolution du signal de sortie durant les différentes phases de fonctionnement.

L'utilisation d'un interrupteur et d'une capacité introduisent des limitations en terme de rapidité et de maintien :

Présence d'une résistance d'entrée : R_{on}

Cette résistance représente à la fois la résistance de sortie du montage en amont de l'E/B mise en série avec la résistance d'état passant de l'interrupteur. Cette résistance va limiter la possibilité du suivi de la tension. En effet la capacité se charge au travers de cette résistance. On obtient donc une constante de temps :

$$\tau_{charge} = R_{on} C$$

Présence d'une résistance de sortie : R_{ch}

Cette résistance est due à la résistance d'entrée du montage en aval de l'E/B associée à la résistance modélisant les pertes de la capacité. Cela introduit une limitation du maintien de la tension lors de la phase de blocage due à la décharge de la capacité dans cette résistance :

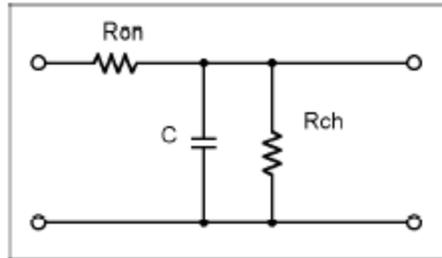


Fig 4 : Modèle équivalent de l'E/B

Après passage dans le bloqueur d'ordre zéro le signal présente des "marches d'escalier", celles-ci se traduisent dans le domaine des fréquences par les raies supplémentaires de la Fig. 5.

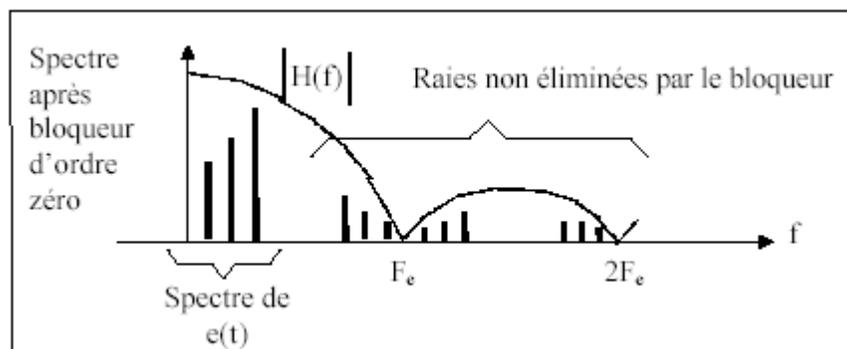


Figure 5 : Spectre du signal après passage dans le bloqueur d'ordre zéro

Application au PIC

La première chose à comprendre, c'est que notre PIC ne contient qu'un seul convertisseur, mais plusieurs pins sur lesquelles peuvent être connectés nos signaux analogiques. Un circuit de commutation sélectionnera donc laquelle parmi ces pins sera reliée au condensateur de maintien interne durant le temps T_{acq} . Ces différentes entrées seront donc des canaux différents d'un seul et même convertisseur.

Corollaire : si vous avez plusieurs canaux à échantillonner, vous devrez les échantillonner à tour de rôle, et donc le temps total nécessaire sera la somme des temps de chaque conversion. Donc, plus vous avez de signaux à échantillonner, moins la fréquence d'échantillonnage pour chaque canal pourra être élevée.

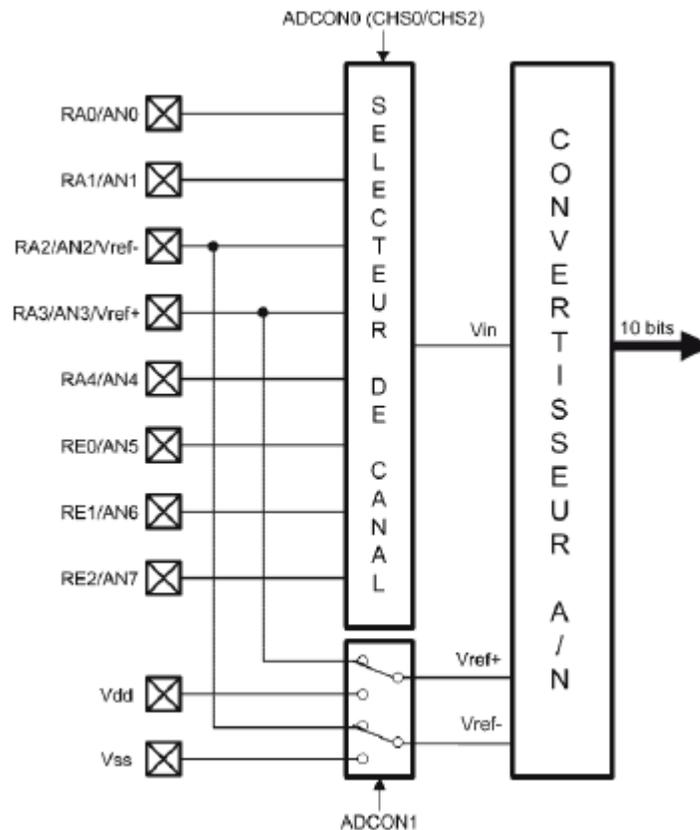
Le 16F877 dispose de 8 canaux d'entrée analogique. Vous pourrez donc échantillonner jusqu'à 8 signaux différents sur les pins AN0 à AN7. Les pins AN0 à AN4 sont les dénominations analogiques des pins RA0 à RA3 + RA5, tandis que les pins AN5 à AN7 sont les dénominations analogiques des pins RE0 à RE2.

Au niveau de notre PIC, nous avons 3 modes de fonctionnement possibles :

- Utilisation de Vss (masse du PIC) comme tension Vref- et de Vdd (alimentation positive du PIC) comme tension Vref+. Dans ce mode, les tensions de références sont tirées en interne de la tension d'alimentation. Il n'y a donc pas besoin de les fournir.
- Utilisation de la pin Vref+ pour fixer la tension de référence maximale Vref+, et utilisation de Vss comme tension de référence Vref-. Dans ce cas, la tension Vref+ doit donc être fournie au PIC via la pin RA3.
- Utilisation de la pin Vref+ pour fixer la tension de référence maximale Vref+, et utilisation de la pin Vref- pour fixer la tension de référence minimale Vref-. Dans ce cas, les 2 tensions de références devront être fournies au PIC via RA3 et RA2. Notez que la broche Vref+ est une dénomination alternative de la broche RA3/AN3, tandis que la broche Vref- est une dénomination alternative de la broche RA2/AN2.

Donc, l'utilisation d'une « pin » comme entrée analogique interdit son utilisation comme entrée numérique (pin entrée/sortie « normale »). De même, l'utilisation des références Vref+ et Vref- interdit leur utilisation comme pin entrée/sortie ou comme pin d'entrée analogique.

Notez également que les pins ANx sont des pins d'entrée. Il n'est donc pas question d'espérer leur faire sortir une tension analogique. Ceci nécessiterait un convertisseur numérique/analogique dont n'est pas pourvu notre PIC. Nous pouvons maintenant dessiner le schéma symbolique des entrées de notre convertisseur analogique/numérique. Ce schéma correspond à un 16F877, pour le 16F876, les canaux AN5 à AN7 n'existent bien entendu pas.



On voit très bien sur ce schéma que les pins AN2 et AN3 servent selon la position du sélecteur d'entrée analogique ou de tension de référence. Le sélecteur de canal permet de de 10 bits, dont nous verrons la destination. Donc, notre procédure de numérisation, pour le sélectionner lequel des 8 canaux va être appliqué au convertisseur analogique/digital.

Remarquez que la sélection de la source des tensions de référence dépend de bits du registre ADCON1, tandis que le canal sélectionné pour être numérisé dépend de ADCON0.

Le convertisseur en lui-même, en toute bonne logique, n'a besoin que de la tension d'entrée (la pin ANx sélectionnée), et des 2 tensions de référence. Il sort un nombre numérique cas où on utilise plusieurs canaux, devient la suivante (après paramétrage) :

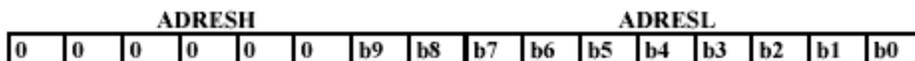
- On choisit le canal à numériser, et on met en route le convertisseur
- On attend Tacq
- On lance la numérisation
- On attend la fin de la numérisation
- On attend 2 Tad
- On recommence avec le canal suivant.

Les registres ADRESL et ADRESH

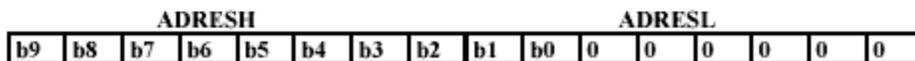
Le convertisseur donne un résultat sur 10 bits, et donc que ce résultat devra donc obligatoirement être sauvegardé dans 2 registres. Ces registres sont tout simplement les registres ADRESL et ADRESH.

Comme 2 registres contiennent 16 bits, et que nous n'en utilisons que 10, Microchip vous a laissé le choix sur la façon dont est sauvegardé le résultat. Vous pouvez soit justifier le résultat à gauche, soit à droite.

La justification à droite complète la partie gauche du résultat par des « 0 ». Le résultat sera donc de la forme :



La justification à gauche procède bien évidemment de la méthode inverse :



La justification à droite sera principalement utilisée lorsque vous avez besoin de l'intégralité des 10 bits de résultat, tandis que la justification à gauche est très pratique lorsque 8 bits vous suffisent. Dans ce cas, les 2 bits de poids faibles se trouvent isolés dans ADRESL, il suffit donc de ne pas en tenir compte. Cette approche est destinée à vous épargner des décalages de résultats. Merci qui ? Merci Microchip.

Le choix de la méthode s'effectue à l'aide du bit 7 de ADCON1, registre dont je vais maintenant vous parler.

ATTENTION : Le registre ADRESH se situe en banque 0, alors que ADRESL se trouve en banque 1.

Le registre ADCON1

Ce registre permet de déterminer le rôle de chacune des pins AN0 à AN7. Il permet donc de choisir si une pin sera utilisée comme entrée analogique, comme entrée/sortie standard, ou comme tension de référence. Il permet également de décider de la justification du résultat.

Notez déjà que pour pouvoir utiliser une pin en mode analogique, il faudra que cette pin soit configurée également en entrée par TRISA et éventuellement par TRISE pour le 16F877.

Le registre ADCON1 dispose, comme tout registre accessible de notre PIC, de 8 bits, dont seulement 5 sont utilisés :

ADCON1

- b7 : ADFM : A/D result ForMat select
- b6 : Inutilisé : lu comme « 0 »
- b5 : Inutilisé : lu comme « 0 »
- b4 : Inutilisé : lu comme « 0 »
- b3 : PCFG3 : Port ConFiGuration control bit 3
- b2 : PCFG2 : Port ConFiGuration control bit 2
- b1 : PCFG1 : Port ConFiGuration control bit 1
- b0 : PCFG0 : Port ConFiGuration control bit 0

Le bit ADFM permet de déterminer si le résultat de la conversion sera justifié à droite (1) ou à gauche (0).

Nous trouvons dans ce registre les 4 bits de configuration des pins liées au convertisseur analogique/numérique. Ces bits nous permettent donc de déterminer le rôle de chaque pin.

Comme nous avons 16 combinaisons possibles, nous aurons autant de possibilités de configuration (en fait, vous verrez que nous n'en avons que 15).

Je vous donne le tableau correspondant à ces combinaisons pour le 16F877:

PCFG 3 à 0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	Vref- Vss	Vref +	A/D/R
0000	A	A	A	A	A	A	A	A	Vss	Vdd	8/0/0
0001	A	A	A	A	Vref+	A	A	A	Vss	RA3	7/0/1
0010	D	D	D	A	A	A	A	A	Vss	Vdd	5/3/0
0011	D	D	D	A	Vref+	A	A	A	Vss	RA3	4/3/1
0100	D	D	D	D	A	D	A	A	Vss	Vdd	3/5/0
0101	D	D	D	D	Vref+	D	A	A	Vss	RA3	2/5/1
0110	D	D	D	D	D	D	D	D	Vss	Vdd	0/8/0
0111	D	D	D	D	D	D	D	D	Vss	Vdd	0/8/0
1000	A	A	A	A	Vref+	Vref-	A	A	RA2	RA3	6/0/2
1001	D	D	A	A	A	A	A	A	Vss	Vdd	6/2/0
1010	D	D	A	A	Vref+	A	A	A	Vss	RA3	5/2/1
1011	D	D	A	A	Vref+	Vref-	A	A	RA2	RA3	4/2/2
1100	D	D	D	A	Vref+	Vref-	A	A	RA2	RA3	3/3/2
1101	D	D	D	D	Vref+	Vref-	A	A	RA2	RA3	2/4/2
1110	D	D	D	D	D	D	D	A	Vss	Vdd	1/7/0
1111	D	D	D	D	Vref+	Vref-	D	A	RA2	RA3	1/5/2

La première colonne contient les 16 combinaisons possibles des bits de configuration PCFG3 à PCFG0. Remarquez déjà que les valeurs 0110 et 0111 donnent les mêmes résultats. Vous avez donc en réalité le choix entre 15 et non 16 combinaisons. Les colonnes « AN7 à AN0 » indiquent le rôle qui sera attribué à chacune des pins concernée. Un « A » dans une de ces colonnes indique que la pin correspondante est configurée comme entrée analogique (ne pas oublier TRISx), un « D » indiquera que la pin est en mode Digital, c'est-à-dire qu'elle se comportera comme une pin d'entrée/sortie « classique ».

La colonne « AN3/RA3 » peut contenir également la valeur « Vref+ » qui indiquera que cette pin devra recevoir la tension de référence maximale. Il en va de même pour AN2/RA2 qui pourra contenir « Vref- », qui indiquera que cette pin doit recevoir la tension de référence minimale.

La colonne « Vref- » indique quelle tension de référence minimale sera utilisée par le convertisseur. Il ne pourra s'agir que de la tension d'alimentation Vss ou de la pin RA2. Cette colonne est donc liée au contenu de la colonne « RA2 ».

Raisonnement identique pour la colonne « Vref+ », liée à « RA3 ». Cette colonne indique quelle sera la tension de référence maximale. De nouveau, il ne pourra s'agir que de la tension d'alimentation Vdd ou de la tension présente sur la pin RA3.

La dernière colonne « A/D/R » résume les colonnes précédentes. Le premier chiffre représente le nombre de pins configurées en tant qu'entrées analogiques, le second en tant qu'entrées/sorties numériques, et le dernier le nombre de pins servant à l'application des tensions de référence.

Comme il y a 8 pins concernées pour le 16F877, la somme des 3 chiffres pour chaque ligne sera bien entendu égale à 8.

Vous voyez que si vous avez le choix du nombre de pins configurées en entrées analogiques, vous n'avez cependant pas le choix de leur attribution. Par exemple, si vous avez besoin de configurer ces ports pour disposer de 3 entrées analogiques et de 5 entrées/sorties numériques, vous devez chercher dans la dernière colonne la ligne « 3/5/0 ». Cette ligne vous indique que vous devez configurer les bits PCFGx à 0100, et que les pins utilisées comme entrées analogiques seront les pins RA0, RA1, et RA3. Vous devez donc en tenir compte au moment de concevoir votre schéma. Une fois de plus, logiciel et matériel sont étroitement liés.

Vous voyez également que lors d'une mise sous tension, les bits PCFGx contiennent 0000. Le PORTA et le PORTE seront donc configurés par défaut comme ports complètement analogiques. Ceci vous explique pourquoi l'utilisation de ces ports comme ports d'entrées/sorties classiques implique d'initialiser ADCON1, avec une des valeurs des 2 lignes complètement en bleu dans le tableau.

Le registre ADCON0

Ce registre est le dernier utilisé par le convertisseur analogique/numérique. Il contient les bits que nous allons manipuler lors de notre conversion. Sur les 8 bits de notre registre, 7 seront utilisés.

- b7 : ADCS1 : A/D conversion Clock Select bit 1
- b6 : ADCS0 : A/D conversion Clock Select bit 0
- b5 : CHS2 : analog Channel Select bit 2
- b4 : CHS1 : analog Channel Select bit 1
- b3 : CHS0 : analog Channel Select bit 0
- b2 : GO/DONE : A/D conversion status bit
- b1 : Inutilisé : lu comme « 0 »
- b0 : ADON : A/D ON bit

Nous avons parlé à maintes reprises de diviseur, afin de déterminer l'horloge du convertisseur en fonction de la fréquence du quartz utilisé. Vous pouvez choisir ce diviseur à l'aide des bits ADCSx.

ADCS1 ADCS0 Diviseur Fréquence maximale du quartz

0	0	Fosc/2 1,25Mhz
0	1	Fosc/8 5Mhz
1	0	Fosc/32 20 Mhz
1	1	Osc RC Si > 1MHz, uniquement en mode « sleep »

Vous avez vu que vous pouvez configurer, via ADCON1, plusieurs pins comme entrées analogiques. Vous avez vu également que vous ne pouvez effectuer la conversion que sur une pin à la fois (on parlera de canal). Vous devez donc être en mesure de sélectionner la canal voulu. Ceci s'effectue via les bits CHSx.

CHS2	CHS1	CHS0	Canal	Pin
0	0	0	0	AN0/RA0
0	0	1	1	AN1/RA1
0	1	0	2	AN2/RA2
0	1	1	3	AN3/RA3
1	0	0	4	AN4/RA5
1	0	1	5	AN5/RE0 (Uniquement pour 16F877)
1	1	0	6	AN6/RE1 (Uniquement pour 16F877)
1	1	1	7	AN7/RE2 (Uniquement pour 16F877)

Le bit ADON permet de mettre en service le convertisseur. Si le canal a été correctement choisi, le positionnement de ce bit permet de démarrer la charge du condensateur interne, et donc détermine le début du temps d'acquisition.

Quant au bit *Go/DONE*, il sera placé à « 1 » par l'utilisateur à la fin du temps d'acquisition. Cette action détermine le début de la conversion en elle-même, qui dure 12 T_{ad} .

Une fois la conversion terminée, ce bit est remis à 0 (« Done » = « Fait ») par l'électronique du convertisseur. Cette remise à 0 est accompagnée du positionnement du flag ADIF du registre PIR1. Ce bit permettra éventuellement de générer une interruption. Vous disposez donc de 2 façons pratiques de connaître la fin de la durée de conversion :

- Si votre programme n'a rien d'autre à faire durant l'attente de la conversion, vous bouclez dans l'attente du passage à 0 du bit GO/Done.
- Si votre programme continue son traitement, vous pouvez utiliser l'interruption générée par le positionnement du flag ADIF.

Attention : Si vous arrêtez manuellement la conversion en cours, le résultat ne sera pas transféré dans les registres ADRESL et ADRESH. Vous n'obtenez donc aucun résultat, même partiel. De plus, vous devrez quand même respecter un temps d'attente de $2T_{ad}$ avant que ne redémarre automatiquement l'acquisition suivante.

La conversion analogique/numérique et les interruptions

Cette partie ne comporte aucune difficulté particulière. En effet, l'interruption générée par le convertisseur est une interruption périphérique, et doit donc être traitée comme telle. Les différentes étapes de sa mise en service sont donc :

- Positionnement du bit ADIE du registre PIE1
- Positionnement du bit PEIE du registre INTCON
- Positionnement du bit GIE du registre INTCON

Moyennant quoi, toute fin de conversion analogique entraînera une interruption. Il vous suffira de remettre à « 0 » le flag ADIF après traitement de cette interruption.

L'utilisation pratique du convertisseur

Voici un résumé des opérations concrètes à effectuer pour échantillonner votre signal :

1. Configurez ADCON1 en fonction des pins utilisées en mode analogique, ainsi que les registres TRISA et TRISE si nécessaire.
2. Validez, si souhaitée, l'interruption du convertisseur
3. Paramétrez sur ADCON0 le diviseur utilisé
4. Choisissez le canal en cours de digitalisation sur ADCON0
5. Positionnez, si ce n'est pas déjà fait, le bit ADON du registre ADCON0
6. Attendez le temps T_{acq} (typiquement $19,7\mu s$ sous 5V)
7. Démarrez la conversion en positionnant le bit GO du registre ADCON0
8. Attendez la fin de la conversion
9. Lisez les registres ADRESH et si nécessaire ADRESL
10. Attendez un temps équivalent à $2T_{ad}$ (typiquement $3,2\mu s$)
11. Recommencez au point 4