
Fiche explicative : Convertisseur Analogique Numérique

Periph'Team - INSA de Toulouse

1 Fonction d'un Analogic Digital Converter (ADC)

Un ADC a pour rôle de fournir un nombre entier à l'image d'une tension. Il sera utile par exemple pour :

- opérer un filtrage numérique sur un flot analogique
- mettre en place une régulation numérique
- mesurer de grandeurs physiques
- ...

Une tension possède par définition une infinité de valeurs possibles (grandeur analogique), alors que le nombre entier en sortie est forcément limité en quantité de valeurs.

La caractéristique typique d'un ADC est illustré figure 1 dont on retiendra les différentes définitions suivantes :

- **Le quantum** d'un ADC est l'incrément de tension q , qui fait évoluer le nombre entier d'une valeur à la suivante.
- **L'étendue de mesure** est la plage d'entrée de l'ADC, noté ΔV sur la caractéristique.
- **La résolution** d'un ADC renseigne plus ou moins directement sur la quantité de valeurs entières que peut prendre le nombre de sortie de l'ADC. Les données suivantes renseignent de 3 manière différentes sur la résolution :
 - ADC 1024 points
 - ADC 10 bits
 - ADC dont l'étendue de mesure est $5V$ et le quantum vaut $4,883mV$

2 Structure d'un ADC à une entrée

Un ADC est systématiquement précédé par un **échantillonneur bloqueur**. En effet, pour qu'un ADC puisse fonctionner, il faut que sa **tension d'entrée soit constante** pendant le travail de conversion. Or bien souvent, les signaux mesurés par un ADC sont variables. Comment faire ? C'est l'échantillonneur bloqueur qui va nous tirer d'affaire. Il va prélever la tension à un instant donné, puis la mémoriser. L'élément central est donc le condensateur qui a la propriété, une fois chargé, de conserver son état (du moins si aucun courant ne lui est soutiré).

Le schéma de la figure 2 montre la structure interne d'un ensemble ADC + échantillonneur bloqueur.

Sur ce schéma on distingue :

- Une entrée de déclenchement, *Trigger*. Elle permet de lancer une conversion.

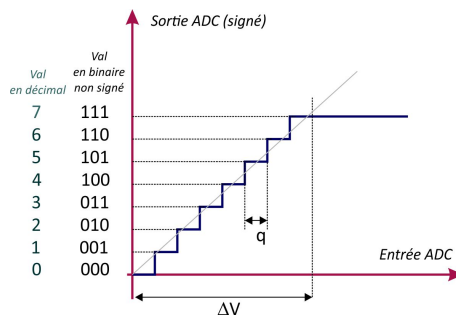


Figure 1: Fonction de transfert et caractéristiques d'un ADC codé sur 3 bits

- Une entrée d'horloge, ADC_clk . Indispensable, elle donne le rythme au séquenceur interne de l'ADC (oui, il y a une logique séquentielle dans tout ADC!)
- Un bloc de synchronisation (un autre séquenceur...) qui permet de commander l'interrupteur de l'échantillonneur au début de la conversion (nous verrons cela en détail juste après).
- Un indicateur de fin de conversion, EOC , qui va permettre à un microprocesseur de venir lire l'information numérique.

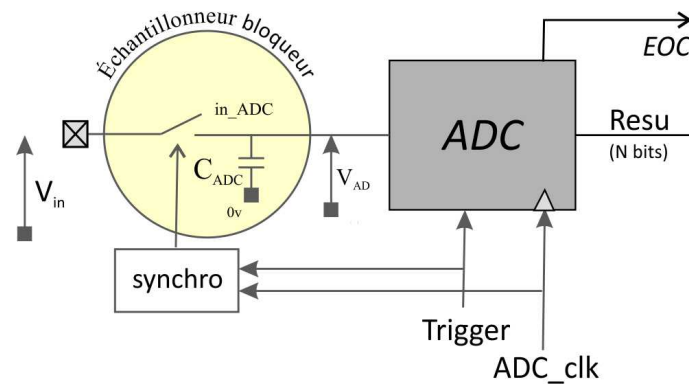


Figure 2: Ensemble échantillonneur bloqueur et ADC

3 L'échantillonneur bloqueur en détail

L'interrupteur de l'échantillonneur, élément indispensable de l'ensemble, est constitué à base de transistors MOS. Sans rentrer dans sa structure (c'est pas l'objet du cours), on gardera à l'esprit que :

- en position fermée, l'interrupteur présente une résistance non nulle, R_{ADC} (c'est un défaut).
- en position ouverte, sa résistance est tellement grande qu'on peut la considérer infinie avec une bonne approximation.

Pour se donner un ordre d'idée, en allant au chapitre "*electrical characteristics*" de la datasheet du *STM103xB*, on trouve les valeurs suivantes :

- $R_{ADC} = 1k\Omega$
- $C_{ADC} = 8pF$

Le schéma d'application est alors celui de la figure 3 avec :

- V_e : c'est par exemple la tension de sortie d'un capteur que l'on mesurerait à vide avec un oscilloscope ou multimètre ayant une grande impédance d'entrée.
- R_{AIN} : résistance de sortie du capteur (ou tout autre circuit électronique). C'est aussi la **résistance de Thévenin** de la source de tension.

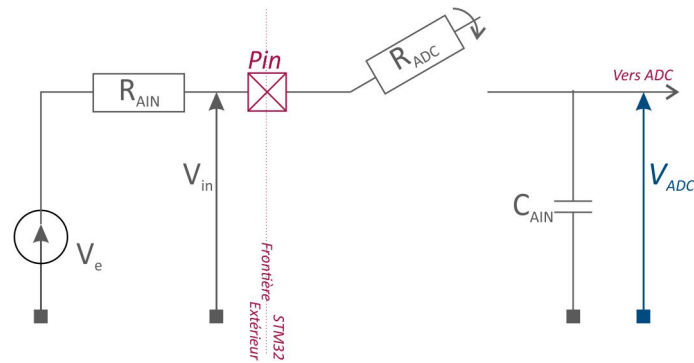


Figure 3: Schéma de l'échantillonneur bloqueur

3.1 Fonctionnement dans un monde parfait, $R_{AIN} = 0$ et $R_{ADC} = 0$

Lorsque l'interrupteur se ferme, le condensateur est instantanément chargé (constante de temps RC considérée nulle). Cela peut provoquer un courant très important, mais on est dans un mode idéal...

La figure ?? illustre ce que cela pourrait donner au niveau de la prise des échantillons.

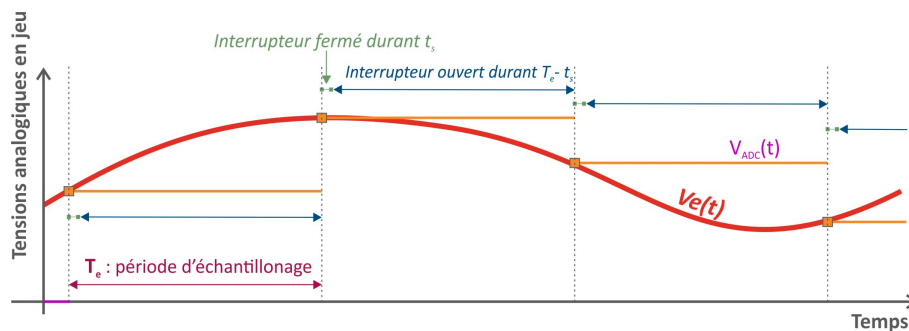


Figure 4: Echantillonnage idéal

Nous voyons ici, que pendant toute la période d'échantillonnage, T_e , la tension V_{ADC} est parfaitement stable (d'où le terme échantillonneur bloqueur) et que l'ADC pourra procéder n'importe quand à la conversion (pour peu que le temps mis pour la conversion rentre dans l'intervalle de temps $T_e - t_s$!). Pour les amateurs de *Matlab-Simulink*, les simulations mixtes continu/discret travaillent dans ce contexte d'instantanéité (le « monde parfait »...). Il n'y a qu'à la mise en œuvre qu'il devient nécessaire de se pencher sur ce problème concret.

3.2 Fonctionnement dans la "vraie vie", $R_{AIN} = 1k\Omega$ et $R_{ADC} \neq 0$

Les choses se compliquent au niveau de la charge du condensateur. Avant la fermeture le condensateur est chargé à une certaine valeur. La tension V_e est différente. Lors de la fermeture, la tension du condensateur V_{ADC} va progressivement atteindre celle de V_e . Cette évolution va se faire de manière exponentielle, régie par la constante de temps $\tau = (R_{AIN} + R_{ADC}) \cdot C_{ADC}$.

La figure 5 présente une analyse plus fine du phénomène d'échantillonnage.

Ainsi, si l'on veut que les valeurs bloquées par l'échantillonneur soient précises (c'est à dire que les valeurs analogiques à convertir soient au plus proche des valeurs V_e réellement produites par le capteur), il faut laisser un temps de fermeture de l'interrupteur, t_s , suffisant pour que le condensateur puisse se charger.

Afin de déterminer t_s , il nous faut partir des paramètres suivants :

- la nature exponentielle de l'évolution de la charge,
- la constante de temps τ
- la résolution de l'ADC, ici $N = 12$ bits : en effet, inutile d'aller chercher le μV de précision si la résolution de l'ADC (son quantum) est de l'ordre du mV !

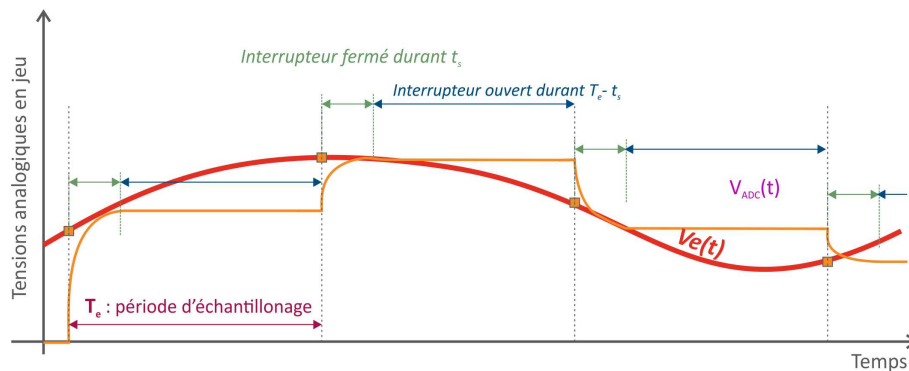


Figure 5: Echantillonnage réel

Dans ces conditions, et si on cherche le $\frac{1}{2}$ quantum de précision, on montre facilement que :

$$t_s = \ln(2^{N+1}) \cdot (R_{AIN} + R_{ADC}) \cdot C_{ADC}$$

Dans la datasheet du STM103xB, on retrouve tous ces éléments.

Il est donc très important de retenir que selon la résistance de sortie du système électronique que l'on connecte sur un ADC, un temps d'échantillonnage minimum doit être respecté pour que la résolution de l'ADC soit atteinte.

Corollaire : *si on a la possibilité, il faut faire en sorte de concevoir le dispositif électronique de sorte à ce que sa résistance de sortie tende vers 0 ! De cette manière, le temps d'échantillonnage peut être minimisé, et on peut ainsi disposer d'une période d'échantillonnage minimale.*

4 Terminologie des divers temps intervenant dans la conversion

Il est très facile de s'embrouiller dans les divers termes associés au fonctionnement de l'ADC. D'une documentation constructeur à une autre, d'un auteur à un autre, d'une langue à une autre, les termes et abréviations changent. De quoi semer le doute...

L'une des sources de confusion se situe dans l'utilisation du mot **échantillonnage** (sampling en anglais). Pour un électronicien, cela fera sans doute penser au processus qui permet de « figer » une tension analogique. Nous pensons bien sûr à l'**échantillonneur bloqueur**. Pour un spécialiste du traitement de signal cette fois, l'échantillonnage fera plutôt référence à la fréquence d'échantillonnage ou période d'échantillonnage : cela ne concerne plus une durée mais un phénomène **périodique**. On pense au *théorème de Shannon*, on pense 24 images/seconde en cinéma, on pense 44.1kHz pour les spécialistes audio...

Dans cet enseignement, on va proposer une terminologie que l'on suivra impérativement pour bien se comprendre et éviter les confusions :

- **Durée d'échantillonnage (sample time, t_s)** : C'est le temps que met l'échantillonneur bloqueur à figer la tension qui lui est appliquée. On peut comparer ce temps au temps d'exposition pour un appareil photo.
- **Période d'échantillonnage (sampling period, sampling interval, T_e)** : C'est l'intervalle de temps qui sépare deux prises d'échantillon. On parlera aussi de Fréquence d'échantillonnage (sampling rate, sampling frequency)
- **Temps de conversion (conversion time, t_{conv})** : c'est la durée qui sépare la demande de conversion à l'obtention du résultat. C'est donc la somme de la durée d'échantillonnage et de la durée conversion propre à l'ADC.

Rappel théorique : Pour avoir une chance d'être correctement restitué, un signal analogique dont la fréquence maximale est F_{Max} doit être échantillonné avec une fréquence d'échantillonnage :

$$F_e = \frac{1}{T_e} > 2.F_{Max}$$

C'est la première contrainte qui doit venir à l'esprit. Ensuite on pourra faire un choix d'ADC rapide ou non, et se poser la question de la durée d'échantillonnage vis à vis de la résistance de sortie du dispositif amont.

5 Fonctionnement interne de l'ADC, notions

On va oublier un peu l'échantillonneur bloqueur pour se consacrer uniquement au bloc ADC (tension d'entrée constante).

Le bloc de la figure 6 peut être implémenté de diverses manières. Chacune possède ses avantages et inconvénients.

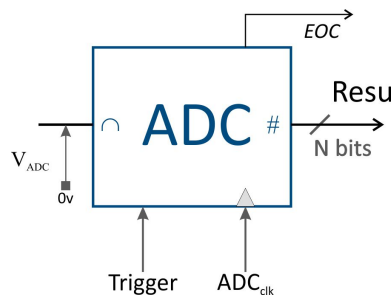


Figure 6: ADC : bloc seul

Avant de balayer quelques technologies existantes, notons tout de suite qu'un ADC inclut forcément un bloc de logique séquentielle. Il aura donc besoin d'une horloge pour fonctionner. C'est d'elle que va systématiquement dépendre le temps mis par le convertisseur pour obtenir le résultat.

Paradoxalement, on trouve souvent à l'intérieur d'un ADC un DAC ! Il est en effet simple à faire. Il suffit de disposer d'un réseau dit $R - 2R$ pour parvenir simplement à une tension analogique en fonction de N entrées binaires dont le poids dépend de l'endroit où le réseau de résistances est attaqué (hors du champs de ce cours).

Voici quelques technologies d'ADC :

- L'ADC simple rampe (Figure 7): simple et précis, mais lent...

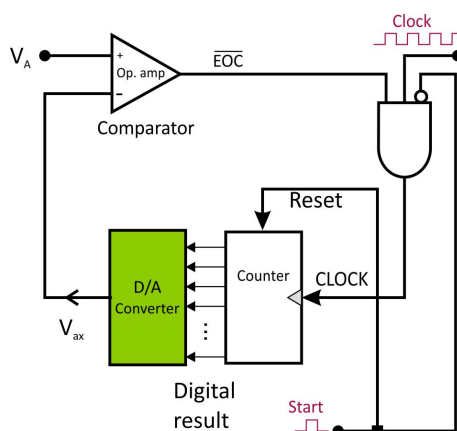


Figure 7: ADC simple rampe

Tant que V_A est supérieure à V_{AX} (sortie du DAC), la porte ET relaie l'horloge, le compteur compte. Dès que la tension V_{AX} atteint V_A , le flag $/EOC$ devient actif ('0') et inhibe l'horloge du compteur. Celui-ci est

alors stoppé. Usuellement un registre N bits prélève les sorties du compteur (c'est le *Data Register*, non représenté ici). Une impulsion *Start* bloque l'horloge, remet à zéro le compteur, qui par cascade invalide le flag */EOC* et le cycle reprend.

Inconvénient : le temps de conversion sera d'autant plus long que la tension V_A est grande.

- **L'ADC à pesées successives (SA) (Figure 7)**: le passe-partout.

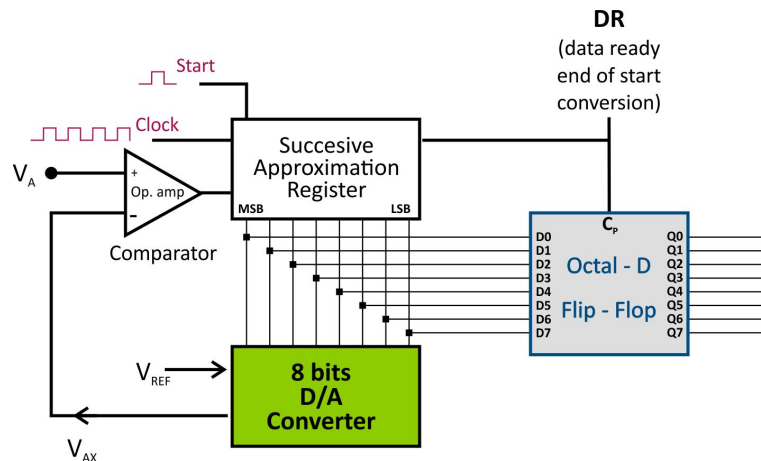


Figure 8: ADC à pesée successive

Le bloc SAR (Successive Approximation Register) n'est pas juste un registre, mais une véritable machine à états. Le principe utilisé est celui de la bonne vieille balance de Mémé : le SAR va placer en sortie le mot 10000000 (juste le poids le plus fort des N bits qu'il peut fournir). Le DAC convertit la valeur en analogique. Celle-ci est comparée et le résultat est renvoyé sur le SAR (machine à états). Elle prend alors la décision (comme le faisait Mémé) d'enlever ce poids trop fort ou de le garder, puis de tester le poids juste plus faible. Le choix donnera donc en sortie du SAR :

- * 01000000 (le poulet pesé est en fait moins lourd...)
- * 11000000 (sacré poulet!)
- * Et ainsi de suite jusqu'à avoir trouvé le poids du poulet (pardon, la valeur analogique la plus proche de celle mesurée). C'est de la dichotomie.

Avantage : quelle que soit la tension à mesurer, le temps de conversion est indépendant du nombre de bits.

- Autre technologies : **Flash** (très rapide, mais moins de résolution), **Sigma Delta** (bon compromis vitesse précision).

6 Et le STM32 alors ?

Le *STM32F103* dispose de deux ADC à pesées successives (on dit aussi approximations successives) d'une résolution de 12 bits, appelés ADC1 et ADC2. Chacun d'eux dispose bien entendu d'un échantillonneur bloqueur mais surtout d'un multiplexeur qui permet d'étendre le nombre de voies possibles à mesurer (jusqu'à 16).

Le périphérique ADC est relativement complexe. Il offre beaucoup de variante de configurations. Nous allons nous limiter dans cette partie à la conversion dite "régulière" (par opposition à "injection"). Nous ne parlerons pas non plus de "analog watchdog".

La partie échantillonneur bloqueur est paramétrée par le registre *ADC_SMPR1* et *ADC_SMPR2*. C'est là qu'on peut agir sur la durée d'échantillonnage.

Le démarrage peut se faire de deux manières différentes. Nous préconisons fortement de passer par le trigger externe, y compris pour un lancement software.

Enfin, les modes de fonctionnement de l'ADC étant assez complexes pour un débutant, nous avons filtré pour vous quelques modes de configurations qui répondent à des applications courantes. C'est l'objet du prochain chapitre qui doit vous aider à appréhender la documentation de référence des périphériques.

7 Les grands modes de fonctionnement

Un schéma simplifié du fonctionnement de l'ADC regroupant les principaux éléments est donnée figure 9

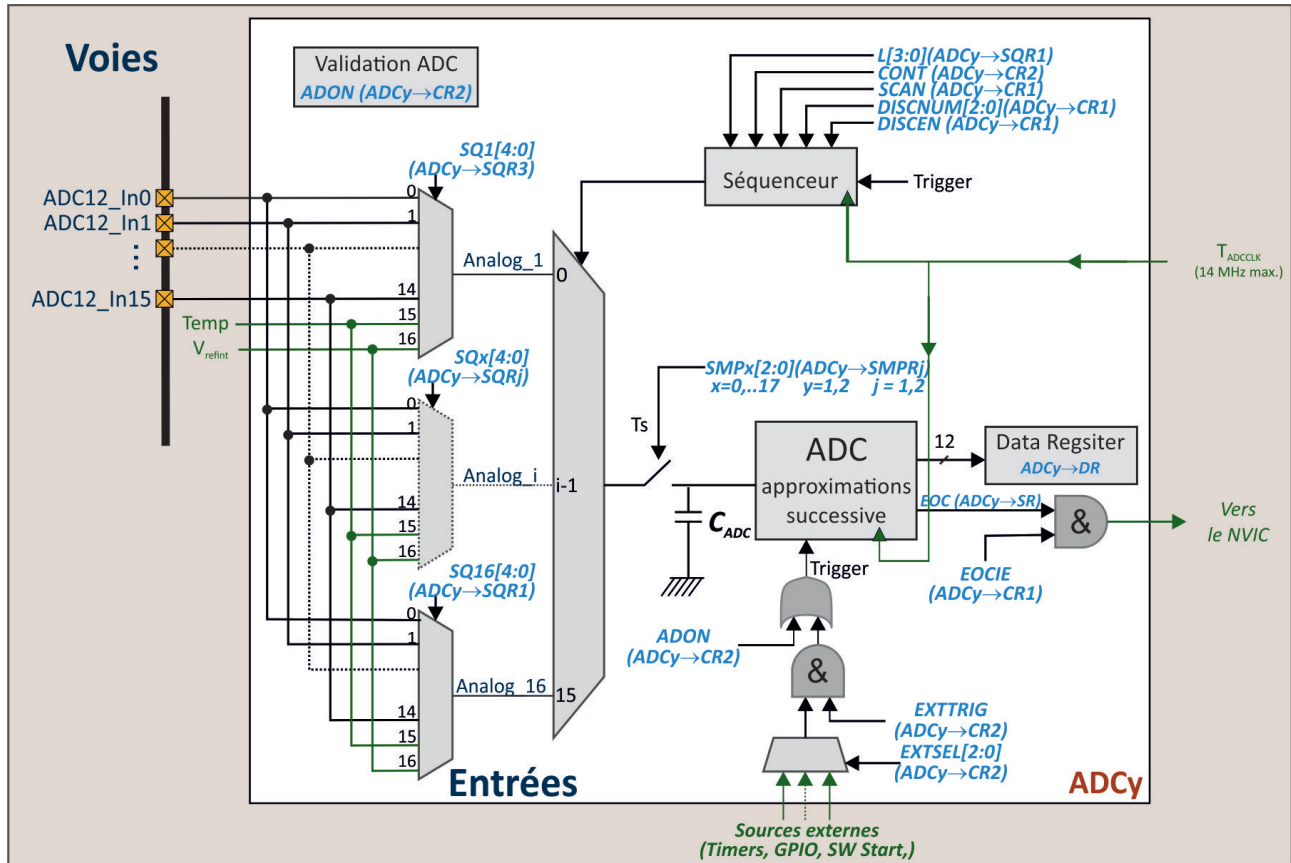


Figure 9: Architecture simplifiée d'un ADC de STM32

7.1 Mode Single not continuous

Contexte habituel pour ce mode : on veut convertir une seule voie.

On veut pouvoir récupérer chacune des données converties pour la traiter en temps réel. On peut travailler en scrutation ou en interruption. Le déclenchement de l'ADC est souhaité avec une périodicité précise (l'entrée de déclenchement ADC, Trig Ext, est alimentée par un signal périodique, de période T_e). La figure 10 présente la configuration du périphérique ADC.

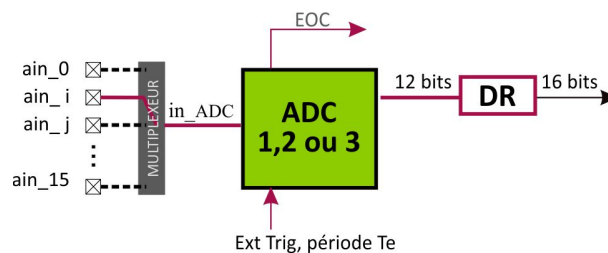


Figure 10: Architecture : "Single mode - not continuous"

Il s'agit de la configuration la plus simple. Une voie est fixée, ici ain_i . Lors d'un déclenchement de l'ADC, celui-ci lance la conversion et range le résultat dans DR (Data Result). A la fin de la conversion, soit le flag EOC

est exploité en scrutation (*polling*) pour savoir si la donnée est prête à être lue, soit il est envoyé vers le *NVIC* pour déclencher une interruption. Si le *NVIC* est bien configuré, celui-ci relaie la demande d'interruption au CPU. Le signal qui déclenche l'ADC (*Ext Trig*) peut provenir de diverses sources au choix.

Par exemple une demande logicielle, ou matérielle via un Timer directement relié à l'entrée *Ext Trig*.

7.2 Mode Single continuos

Contexte habituel pour ce mode: on veut toujours convertir une seule voie. On ne veut pas spécialement maîtriser la période d'échantillonnage de l'ADC. On va le laisser tourner au maximum de sa vitesse, quitte à perdre des données (écrasement de l'ancienne valeur dans *DR*). On vient lire la valeur dans le *DR* quand on le souhaite, de manière asynchrone du travail de l'ADC.

La figure 11 présente la configuration du périphérique ADC.

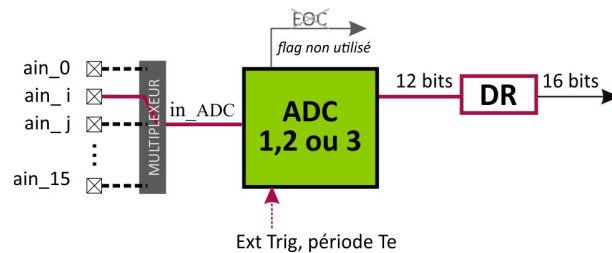


Figure 11: Architecture : "Single mode - continuous"

Le déclenchement de l'ADC se fera une seule fois. Dès qu'une conversion est finie, l'ADC lance la prochaine. Le flag *EOC* ne devrait pas être utilisé, les IT encore moins. La lecture se fera à la volée.

7.3 Mode Single, not continuous et DMA

Contexte habituel pour ce mode : on veut toujours convertir une seule voie, mais ce qui va nous intéresser ici, c'est de travailler sur un bloc de donnée (typiquement pour faire une FFT).

Pendant l'acquisition du bloc, on veut pouvoir libérer le CPU pour qu'il procède à une autre tâche.

La figure 12 présente la configuration du périphérique ADC.

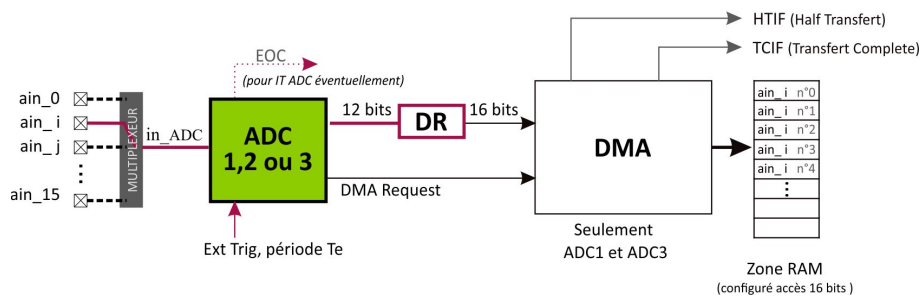


Figure 12: Architecture : "Single mode - not continuous - DMA"

Dans ce cas, il est classique de maîtriser la période d'échantillonnage (signal périodique sur *Ext Trig*). En effet, les données du futur bloc doivent généralement être parfaitement datées. Ainsi, on ne se placera pas en mode continu. La *DMA* (*Direct Memory Access*) va nous aider à réaliser cela. Bien configurée, elle permet de prendre automatiquement la donnée fraîchement convertie et de la ranger dans un tableau défini en RAM. Chaque donnée écrite en RAM est suivie par une incrémentation d'un compteur d'adresse pour que la suivante puisse être rangée à son tour. Si la *DMA* est configurée en mode circulaire, la donnée qui provoquerait un débordement du tableau se retrouve en tête dans ce tableau, et le processus se répète. Si elle ne l'est pas, la *DMA* s'arrête (même si l'ADC continue à produire des données). Enfin on notera que l'ADC peut éventuellement être configuré en interruption. On peut alors réaliser un filtre numérique en temps réel par exemple. La *DMA* déclenche une interruption à la fin du remplissage de la table, ou à la moitié du parcours.

7.4 Mode Scan, not continuous et DMA

Contexte habituel pour ce mode : Dans cette situation, on veut convertir n canaux. Chaque entrée souhaitée va être traitée tour à tour grâce au multiplexeur d'entrée. Une séquence à programmer permet de définir quelles entrées sont concernées et quel est l'ordre de balayage. Il est fortement conseillé dans ce cas de coupler l'ADC avec la DMA. Dans ce cas, chaque case du tableau en RAM correspondra à une mesure donnée.

La figure 13 présente la configuration du périphérique ADC.

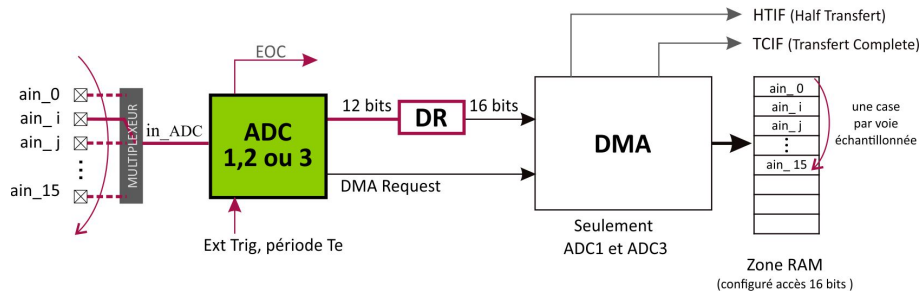


Figure 13: Architecture : "Mode Scan - not continuous - DMA"

A chaque fois que l'ADC est déclenché, la voie d'entrée évolue. Sous condition que la taille du tableau de la DMA corresponde bien au nombre de voies souhaitée, chaque canal sera aiguillé vers sa case mémoire. On utilisera les interruptions DMA pour venir lire au bon moment les n valeurs souhaitées.

7.5 Mode Scan, continuous et DMA

Contexte habituel pour ce mode : Il s'agit ici d'un mode qui ressemble beaucoup au *Single Continuous Mode* mais sur plusieurs canaux. Le but est le même : sur-échantillonner les tensions (en laissant l'ADC tourner à plein régime) et venir piocher de temps en temps les valeurs dans les bonnes cases du tableau de DMA. La configuration est donc la même que précédemment sauf l'ADC est déclenché une seule fois et les interruptions de DMA deviennent inutiles.

7.6 Résumé des modes de fonctionnement présentés

	Flot non continu	Flot continu
Une seule voie	<ul style="list-style-type: none"> • Mode Single not continuous <ul style="list-style-type: none"> – IT – ou polling en fin de conversion – Résultat dans DR • Mode Single, not continuous and DMA <ul style="list-style-type: none"> – IT en fin de DMA – ou IT à mi chemin – IT ADC si besoin – Résultat en RAM, 1 case par échantillon 	<ul style="list-style-type: none"> • Mode Single continuous <ul style="list-style-type: none"> – lecture résultat "à la volée" – EOC non utilisé – Pas d'IT – Résultat dans DR
MultiVoie	<ul style="list-style-type: none"> • Mode Scan, not continuous and DMA <ul style="list-style-type: none"> – IT en fin de DMA – Résultat en RAM, une case par voie 	<ul style="list-style-type: none"> • Mode Scan, continuous and DMA <ul style="list-style-type: none"> – Résultat "à la volée" – EOC non utilisé – Pas d'IT – Résultat en RAM, une case par voie

8 Résumé des connaissances

Nous avons vu :

- ▶ le rôle d'un *ADC* et ses principales caractéristiques,
- ▶ le rôle particulier de l'échantillonneur bloqueur, ainsi que ses contraintes de réglage,
- ▶ la terminologie associée aux grandeurs temporelles et fréquentielles. Vous savez maintenant de quoi on parle !
- ▶ quelques exemples de fonctionnement interne d'un *ADC*. Ce n'est plus magique, vous savez comment s'opère la transformation analogique numérique pour certain type d'*ADC*,
- ▶ les grands modes applicatifs d'un *ADC* que l'on doit pouvoir configurer sur un *STM32*