

Convertisseur Analogique Numérique

PRESENTATION

Le convertisseur analogique numérique (ADC) des microcontrôleurs PIC fonctionne selon le principe des approximations successives avec une résolution de 10 bits. Le microcontrôleur PIC18F4520 dispose de treize canaux d'entrées analogiques répartis sur les ports A, B et E. Les tensions de référence V_{ref+} et V_{ref-} du module ADC peuvent être choisies de manière logicielle par combinaison des lignes VDD, VSS, RA2 et/ou RA3. Le résultat de conversion de 10 bits est chargé dans deux registres de 8 bits ADRESL et ADRESH. Le schéma bloc de l'ADC est illustré à la Figure 37.

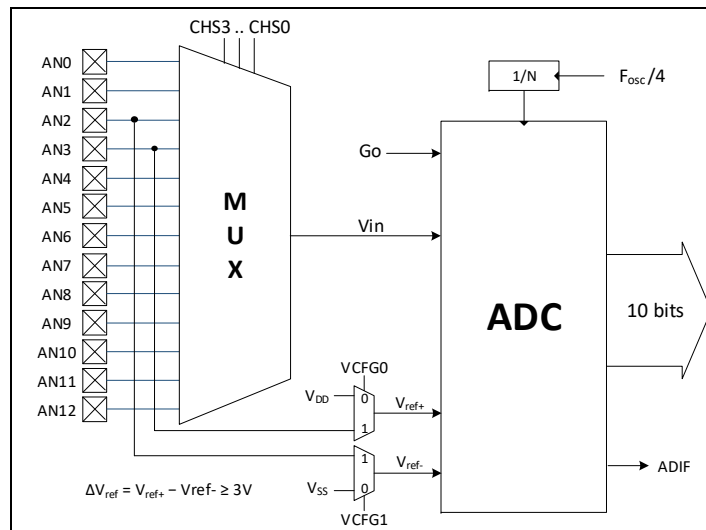


Figure 37 : Schéma bloc de l'ADC

Le convertisseur analogique numérique a la particularité de pouvoir fonctionner lorsque le microcontrôleur est en mode veille. En mode veille, l'horloge de convertisseur doit être dérivée de l'oscillateur RC interne de l'ADC.

Le module ADC dispose de cinq registres :

- ADRESL (A/D RESult Low register)
- ADRESH (A/D RESult High register)
- ADCON0 (A/D CONTrOl register 0)
- ADCON1 (A/D CONTrOl register 1)
- ADCON2 (A/D CONTrOl register 2)

CONVERSION ANALOGIQUE NUMERIQUE

La conversion d'un signal analogique en équivalent numérique passe par deux phases :

- L'échantillonnage blocage (sample and hold). Cette opération consiste à connecter l'entrée à convertir à un condensateur, qui va se charger à travers une résistance jusqu'à la tension appliquée. Le temps nécessaire au charge complet du condensateur est appelé temps d'acquisition T_{ACQ} (Figure 38).
- Une fois le condensateur est chargé, l'ordre de début de conversion permet de déconnecter la tension appliquée, pour procéder à la phase de conversion.

Temps d'acquisition

Le temps d'acquisition dépend de trois paramètres : le temps de charge du condensateur de l'échantillonneur bloqueur à travers les résistances internes et la résistance de la source, le temps de stabilisation de l'amplificateur et d'un temps qui dépend de la variation de la température.

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$

T_{AMP} : temps de stabilisation de l'amplificateur,

T_C : temps de charge du condensateur,

T_{COFF} : temps dépendant du coefficient de température,

D'après le document du constructeur :

$$T_{AMP} = 0,2\mu s$$

$$T_{COFF} = (Temp - 25^\circ C)(0.02 \mu s/^\circ C)$$

$$\text{Pour Temp} = 85^\circ C, T_{COFF} = 1,2\mu s,$$

La résistance interne totale (composée de R_{IC} et R_{SS}), R_{IC} est limitée à $1k\Omega$; R_{SS} varie en fonction de la tension d'alimentation selon l'équation suivante : $R_{SS}(k\Omega) =$

$-V_{DD} + 7$. Microchip recommande que la résistance de la source (R_S) devra être inférieure ou égale à $2,5\text{ k}\Omega$ (Figure 38). Le temps de charge du condensateur est donné par l'expression suivante :

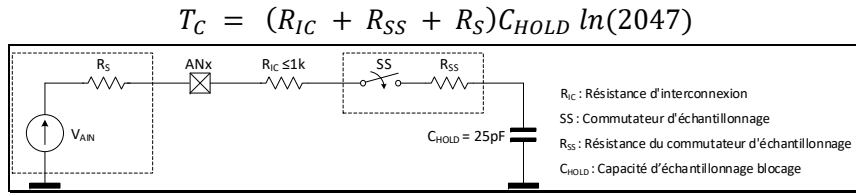


Figure 38

Dans les conditions normales et pour une tension d'alimentation de 5V, le temps d'acquisition $T_{ACQ} = 2,4\mu\text{s}$.

Temps de conversion

C'est le temps nécessaire pour convertir le signal analogique en équivalent numérique. Le temps de conversion dépend essentiellement de la période d'horloge du convertisseur T_{AD} (Typiquement $0,7\mu\text{s}$).

Le microcontrôleur PIC18 nécessite un temps T_{AD} avant le démarrage effectif de la conversion, $10 \cdot T_{AD}$ pour la conversion (une période par bit) et un temps supplémentaire T_{AD} à la fin de la conversion. Soit au total $12 \cdot T_{AD}$. Donc dans les meilleures conditions $T_{conv} = 12 \times 0,7\mu\text{s} = 8,4\mu\text{s}$.

Il faut bien noter qu'une attente à $2 \times T_{AD}$ est nécessaire avant d'effectuer une nouvelle conversion.

REGISTRES DE L'ADC

Registre ADCON1

b7 - U-0	b6 - U-0	b5 - R/W-0	b4 - R/W-0	b3 - R/W-0	b2 - R/W-q	b1 - R/W-q	b0 - R/W-q
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

Les treize entrées analogiques sont multiplexées avec les ports d'entrées/sorties. Le nombre des entrées qui doivent être configurées en analogique dépend de l'application. Les quatre bits **PCFG<3:0>** du registre ADCON1, permettent la configuration des broches en analogique ou en numérique. Le Tableau 8 récapitule les combinaisons possibles de configuration des broches de l'ADC.

Tableau 8 : Configuration des entrées de l'ADC

PCFG 3 .. 0	AN12 RB0	AN11 RB4	AN10 RB1	AN9 RB3	AN8 RB2	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A : Analogique

D : Numérique

Par défaut l'ADC est configuré avec des tensions de référence internes ($V_{ref+} = V_{DD}$ et $V_{ref-} = V_{SS}$) ne nécessite aucune liaison externe. Cependant, une référence interne peut ne pas offrir toujours les performances requises. Dans ce cas, les broches RA2 et RA3 peuvent être configurées en références externes. Les bits VCFG1 et VCFG0 permettent de sélectionner les références internes ou externes ou une combinaison de deux.

VCFG1 = 1 $\rightarrow V_{ref-}$ sur AN2	VCFG1 = 0 $\rightarrow V_{ref-} = V_{SS}$
VCFG0 = 1 $\rightarrow V_{ref+}$ sur AN3	VCFG0 = 0 $\rightarrow V_{ref+} = V_{DD}$

Le concepteur doit choisir parmi ces options, la combinaison la plus appropriée à son application.

Pour les microcontrôleurs alimentés sous 5V, la différence entre les tensions de référence doit être supérieure ou égale à 3V.

$$(\Delta V_{ref+} = V_{ref+} - V_{ref-} \geq 3V).$$

Exemple

Supposons qu'une application a besoin de 4 entrées analogiques et une entrée de référence positive externe. Quelle est la valeur à charger dans le registre ADCON1 ?

Solution

D'après le Tableau 8, un choix optimal nécessite l'utilisation des entrées AN0, AN1, AN2 et AN4 (combinaison : 1010); RA3 comme entrée de référence positive $V_{CFG1} = 0$ et $V_{CFG0} = 1$.

$$ADCON1 = 0b00011010.$$

Registre ADCON0

b7 - U-0	b6 - U-0	b5 - R/W-0	b4 - R/W-0	b3 - R/W-0	b2 - R/W-0	b1 - R/W-0	b0 - R/W-0
-	-	CHS3	CHS2	CHS1	CHS0	GO/ \overline{DONE}	ADON

Tableau 9 : Sélection de l'entrée à convertir

CHS3	CHS2	CHS1	CHS0	Canal	Broche
0	0	0	0	0	RA0/AN0
0	0	0	1	1	RA1/AN1
0	0	1	0	2	RA2/AN2
0	0	1	1	3	RA3/AN3
0	1	0	0	4	RA5/AN4
0	1	0	1	5	RE0/AN5
0	1	1	0	6	RE1/AN6
0	1	1	1	7	RE2/AN7
1	0	0	0	8	RB2/AN8
1	0	0	1	9	RB3/AN9
1	0	1	0	10	RB1/AN10
1	0	1	1	11	RB4/AN11
1	1	0	0	12	RB0/AN12
1	1	0	1	Non implémentés	
1	1	1	0		
1	1	1	1		
1	1	1	1		

Après avoir défini les broches appropriées dans les ports spécifiques comme des entrées analogiques, l'entée à convertir est sélectionnée par les bits CHS<3:0> du registre ADCON0. Le Tableau 9 illustre les différentes combinaisons pour la sélection de l'entrée à convertir.

Après la mise sous tension, l'ADC est désactivé, ceci permet de réduire la consommation. Le bit d'activation ADON doit être mis à 1 lorsque le module ADC est utilisé.

La mise à 1 du bit GO/ \overline{DONE} de manière logicielle, démarre la conversion. La fin de la conversion est signalée par le retour à zéro du bit GO/ \overline{DONE} et la mise à 1 du bit indicateur d'interruption ADIF (PIR1<6>). Une interruption pourra être générée si le bit

ADIE (PIE1<6>) est défini. Le résultat de conversion est placé dans les registres ADRESH et ADRESL.

Le code suivant illustre un exemple de lancement d'une opération de conversion analogique numérique :

```
ADCON0bits.GO = 1 ; // début de conversion
while(ADCON0bits.GO == 1) continue ; // Attente de fin de conversion
AdcVal = ADRESH<<8 + ADRESL ; // lecture de la valeur convertie
```

Registre ADCON2

La fonction principale de ce registre est de permettre à l'utilisateur de programmer la source d'horloge et le temps d'acquisition de l'ADC.

b7 - R/W-0	b6 - U-0	b5 - R/W-0	b4 - R/W-0	b3 - R/W-0	b2 - R/W-0	b1 - R/W-0	b0 - R/W-0
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0

Sélection de l'horloge de l'ADC

La fréquence d'horloge de l'ADC F_{AD} est dérivée de la fréquence principale du microcontrôleur F_{osc} . Pour des raisons électroniques, la période d'horloge de l'ADC est limitée à 0,7µs, soit une fréquence maximale $F_{AD} = 1,428MHz$. Le diviseur intégré est programmé pour diviser la fréquence F_{osc} par 2, 4, 8, 16, 32 et 64. La sélection de l'une de ces valeurs de division est spécifiée dans le champ de bits ADCS<2:0> (ADCON2<2:0>). Lorsque la source d'horloge est un circuit RC, T_{AD} est typiquement égale à 1,2µs. Le Tableau 10 illustre le choix de la période de l'ADC au moyen des bit ADCS<2:0>.

Tableau 10 : choix de T_{AD} en fonction de F_{osc}

ADCS2	ADCS1	ADCS0	Période d'horloge T_{AD}	F_{osc} Maximale
0	0	0	2 T_{osc}	2,86 Mhz
1	0	0	4 T_{osc}	5,71 Mhz
0	0	1	8 T_{osc}	11,43 Mhz
1	0	1	16 T_{osc}	22,86 Mhz
0	1	0	32 T_{osc}	40 Mhz
1	1	0	64 T_{osc}	40 Mhz
X	1	1	RC interne	1 Mhz

Exemple

Déterminer la valeur à charger dans le champ de bits ADCS<2 :0> pour une fréquence $F_{osc} = 8MHz$.

Solution :

Cherchons la valeur de division N la plus appropriée :

$$T_{AD} = N \times T_{osc} \Rightarrow F_{osc} = N \times F_{AD} = \frac{N}{T_{AD}} \text{ d'où } N = F_{osc} \times T_{AD}$$

Cherchons la valeur minimale N_{min} pour la valeur limite $T_{ADmin} = 0,7\mu s$.

$$N_{min} = F_{osc} \times T_{ADmin} = 8 \cdot 10^6 \times 0,7 \cdot 10^{-6} = 5,6$$

Toute valeur de N supérieure à 5,6 est acceptable, cependant, la valeur optimale est celle la plus proche de N_{min} , d'après le Tableau 10, $N = 8$. La valeur de $T_{AD} = N \times T_{osc} = \frac{N}{F_{osc}} = \frac{8}{8 \cdot 10^6} = 1\mu s$.

Il était possible de déterminer la valeur de N directement à partir du Tableau 10. La fréquence $F_{osc} = 8MHz$ correspond à la ligne dont la fréquence maximale est 11,43MHz, d'où $T_{AD} = 8 \times T_{osc}$.

$$T_{AD} = 8 \times T_{osc} \text{ alors } ADCS < 2 : 0 > = 001.$$

Programmation du temps d'acquisition

Pour les microcontrôleurs plus récents tel que le PIC18F4520, un temps d'acquisition automatique peut être programmé à travers les bits ACQT<2:0> (ADCON2<5:3>). Lorsque le bit GO/\overline{DONE} est positionné, le condensateur de l'échantillonneur bloqueur reste connecté pendant le temps d'acquisition sélectionné. Une fois le temps d'acquisition programmé est épuisé, le condensateur de C_{HOLD} sera déconnecté et la conversion effective commence. Le condensateur de maintien sera reconnecté à l'entrée analogique à la fin de la conversion (Figure 39). Le temps d'acquisition étant programmé, il n'est pas nécessaire d'attendre un temps d'acquisition entre la sélection d'un canal et la mise à 1 du bit GO/\overline{DONE} . Le Tableau 11 présente les différentes valeurs possibles du temps d'acquisition en fonction de la période T_{AD} .

Tableau 11 : choix du temps d'acquisition

ACQT2	ACQT1	ACQT0	Temps d'acquisition (T_{Acq})
1	1	1	20 T_{AD}
1	1	0	16 T_{AD}
1	0	1	12 T_{AD}
1	0	0	8 T_{AD}
0	1	1	6 T_{AD}
0	1	0	4 T_{AD}
0	0	1	2 T_{AD}
0	0	0	0 T_{AD} (mode compatibilité)

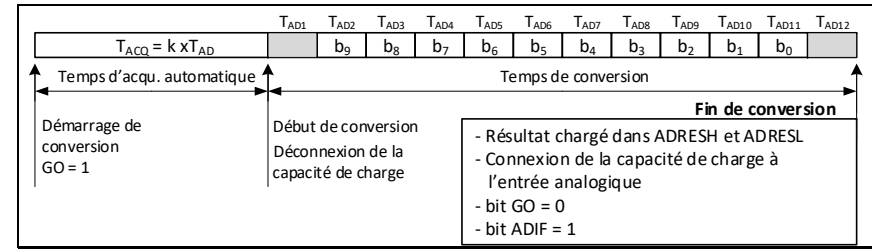


Figure 39 : Cycles d'une opération A/N

Exemple

Déterminer la valeur à charger dans le champ de bits ACQT<2:0> pour un temps d'acquisition de 2,4μs ; on admet que $T_{AD} = 1\mu s$

Solution :

Le temps d'acquisition est fonction de la période d'horloge de l'ADC : $T_{ACQ} = K \times T_{AD}$, d'où $K = T_{ACQ}/T_{AD} = 2,4\mu s/1\mu s = 2,4$, d'après le Tableau 11, la valeur la plus proche $T_{ACQ} = 4 \times T_{AD}$ d'où ACQT<2:0> = 010.

Alignement des données

Le module ADC a besoin de deux registres pour un résultat de conversion de 10 bits. La capacité totale des registres $ADRESH$: $ADRESL$ étant de 16 bits, il existe deux manières d'aligner les dix bits. Généralement c'est l'alignement à droite qui est utilisé, cependant, des nombreuses applications nécessitent uniquement une résolution de 8 bits. Dans ce cas, l'alignement à gauche se trouve plus adapté et les deux derniers bits peuvent être ignorés. Le bit **ADFM** du registre **ADCON2** définit la façon de charger le résultat de conversion dans les registres $ADRESL$ et $ADRESH$ (Figure 40).

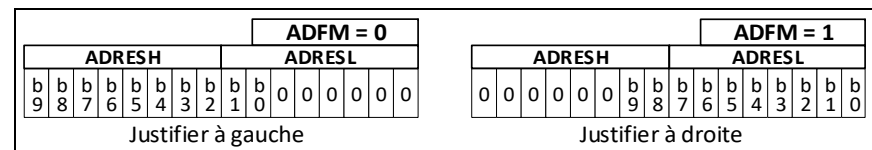


Figure 40 : format de donnée