



Paolo Rognoni (Paolino)

LO HAI MAI REALIZZATO CON UN PIC? I PIC E I SEGNALI ANALOGICI: LA CONVERSIONE A/D

8 June 2010

Dall'analogico al digitale

Anche con i **PICMicro**, così come la maggior parte dei microcontrollori, è possibile **acquisire segnali analogici** e gestirli come **valori digitali**. Non voglio dilungarmi sull'importanza di poter trattare segnali analogici in ambito digitale, ma al lettore si richiede un minimo di dimestichezza con questi argomenti di base. In questo articolo si mostra come deve essere configurato il modulo ADC dei PICMicro, andando ad approfondire argomenti spesso considerati ostici. Dopo questa breve premessa, si va ad incominciare!

Il modulo ADC dei PICMicro

Molti PICMicro sono dotati di **convertitore analogico/digitale (ADC, Analog-to-Digital Converter)** tanto gli **8bit** quanto i **16bit** e i **32bit**. Fornire uno schema univoco per capire come funziona il modulo ADC dei PIC è impresa assurda. Tuttavia ritengo che per i neofiti sia interessante approcciarsi ai PICMicro a 8 bit, dai 12F ai 18F, passando attraverso i 16F. Nonostante ci si concentri sulla sola famiglia a 8bit, il ventaglio di possibilità è ancora molto ampio ma è possibile comunque un approccio di tipo sistematico. Sebbene gli esempi e le considerazioni che seguiranno sono specifiche per **PIC16F87Xa** ed (alcuni) PIC12F, senza troppe difficoltà il lettore dovrebbe essere in grado di applicare quanto esposto a tutti i PICMicro a 8 bit dotati di modulo ADC.

Il modello dell'ADC

Il datasheet del PIC16F87xA illustra il modello interno del modulo ADC, secondo quanto esposto in figura 1. Con ANx si intende il generico pin in grado di interfacciarsi al modulo ADC. Obiettivo principale è quello di configurare il modulo ADC e di stabilire i valori da inserire nei registri al fine di eseguire una corretta conversione A/D. I valori dei registri sono influenzati da alcuni parametri, come ad esempio quelli di figura 1:

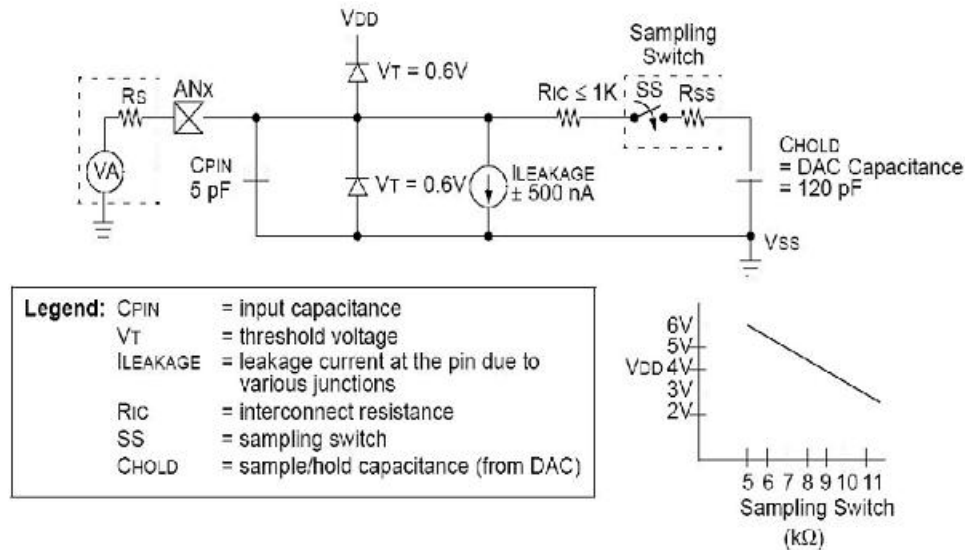


Figura 1

R_S : impedenza della sorgente analogica

R_{IC} : resistenza di interconnessione

R_{SS} : resistenza del dispositivo di commutazione

Calcolo del periodo di campionamento

Il periodo di campionamento è suddiviso in due parti, secondo quanto illustrato in figura 2: Acquisition time: tempo di acquisizione A/D Conversion time: tempo di conversione

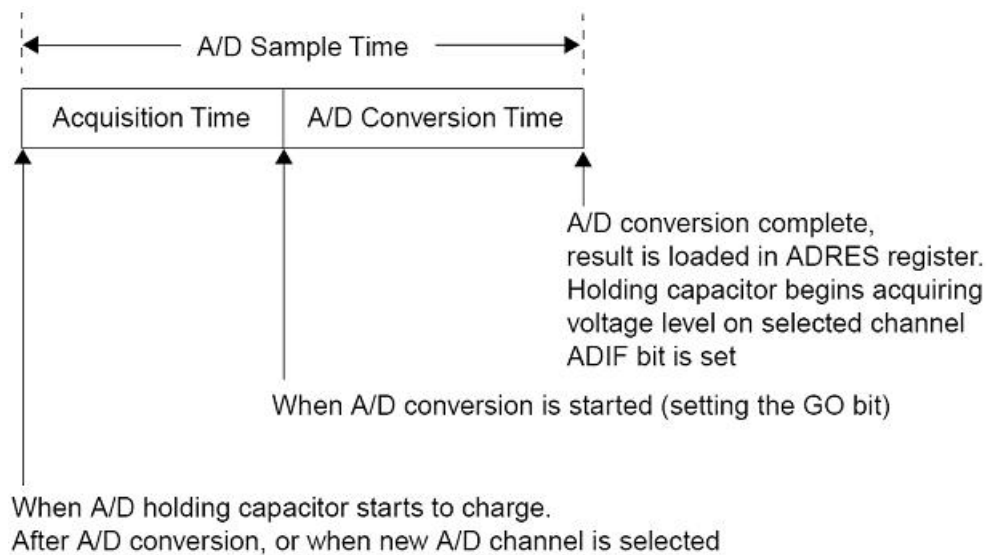


Figura 2

Il periodo di acquisizione T_{acq} , è ottenuto come somma dei contributi:

$$T_{acq} = T_{amp} + T_c + T_{coff}$$

dove:

T_{amp} = Amplifier Settling time

T_c = Hold capacitor Charging time

T_{coff} = Temperature coefficient

Il datasheet, illustra il metodo di calcolo:

$$T_{amp} = 2\mu s \text{ (sembra un dato fisso)}$$

$$T_c = CHOLD * (RIC + RSS + RS) \ln(1 / 2047)] \text{ (Eq. 1)}$$

$$T_{coff} = 0.05 * (T - 25) \text{ (Eq. 2)}$$

Nell'espressione di **T_c** , compaiono i valori **Ric** , **Rs** , **Rss** menzionati in precedenza. I valori di queste resistenze sono così stabiliti: **Rs** : è il valore dell'impedenza di ingresso che non deve superare il valore indicato sul datasheet (a seconda del PIC, può variare da 2.5 kohm a 10kohm)

Ric : si assume il valore massimo, cioè 1k

Rss : la resistenza dell'interruttore di commutazione è calcolata a partire dalla curva di figura 3, che mostra, per il PIC16F87xA, l'andamento del valore della resistenza in funzione della tensione di alimentazione. Eseguendo un'interpolazione lineare tra due punti per ricavare l'equazione della retta, è possibile determinare qualsiasi valore di **Rss** , fornito **V_{dd}** come valore di input.

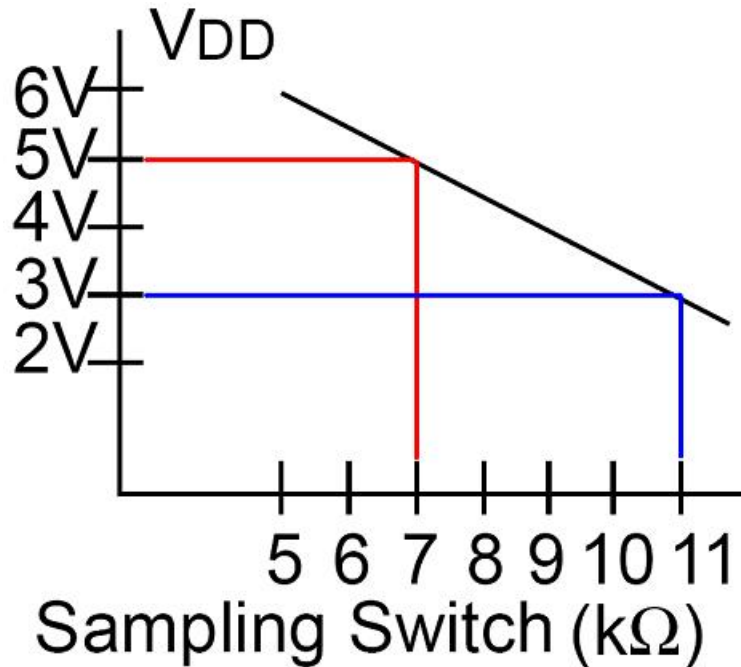


Figura 3

Attenzione: il valore di T_{coff} è richiesto solamente per temperature superiori a 25°C.

Se consideriamo i punti del grafico di figura 3, e indicando con X i termini in ohm e in Y i termini in volt, si ha che l'equazione della retta passante per $P = (11,3)$ $Q = (7,5)$ diventa:

$$x = -2y + 17$$

e sostituendo i termini dei parametri:

$$R_{ss} = -2V_{dd} + 17 \text{ (Eq. 3)}$$

Il secondo termine, quello richiesto per la conversione A/D, per ogni singolo bit viene denominato **Tad**; per una conversione a 10bit, il periodo totale richiesto è pari a **11.5Tad**. Per il PIC16F87xA, è possibile scegliere tra diversi valori per Tad:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC

- Internal A/D module RC oscillator (2-6 μ s)

Il tempo di conversione va stabilito ricordando che deve essere garantito un tempo minimo pari a **1.6 μ s**. In funzione del valore di Fosc (e quindi di T_{osc}) è possibile determinare il valore minimo.

Il minimo periodo di campionamento pertanto vale:

$$T_{smp} = T_{acq} + 11.5 * T_{ad} \text{ (Eq. 4)}$$

Un esempio di calcolo

Supponiamo di avere i seguenti dati:

- Fosc = 4MHz
- Vdd = 5V
- R_s = 2 kohm
- Temperatura di esercizio: 40°C

Secondo quanto espresso dal datasheet, e secondo le formule riportate, il tempo di acquisizione vale:

$$T_{acq} = 10.984\mu s$$

Con Fosc pari a 4MHz, per garantire un $T_{ad} \geq 1.6\mu s$ è necessario selezionare il divisore pari a 8T_{osc}.

Una possibile routine che configura AN0 per la conversione A/D è il seguente:

```
ADC_INIT
banksel ADCON1
clrf ADCON1
bsf ADCON1,PCFG0 ;solo AN0 configurato come ingresso analogico

banksel ADCON0
clrf ADCON0
bsf ADCON0, ADCS0 ;Conversion clock pari a 8Tosc
bsf ADCON0, ADON ;avvio del modulo ADC
```

Foglio di calcolo dei parametri A/D dei PIC

Ho realizzato un semplice foglio di calcolo per aiutarmi nella determinazione dei parametri principali da considerare per il calcolo del periodo di campionamento. Il foglio di calcolo è scaricabile [a questo link](#). Un esempio di come vengono inseriti i dati e proposti i risultati è mostrato in figura 4.

	A	B	C	D	E
3					
4		Frequenza oscillatore	Fosc [Mhz]	4	
5		Tensione di alimentazione	VDD [V]	5	
6		Impedenza della sorgente	Rs [kΩ]	2	
7		Impedenza del sampling switch	Rss [kΩ]	6	
8		Resistenza di interconnessione	Ric [kΩ]	1	
9		Capacità di sample/hold	CHOLD [pF]	120	
10		Temperatura	TEMP [°C]	40	
11					
12					
13		Amplifier Settling Time	TAMP [μs]	2	
14		Hold Capacitor Charging	Tc [μs]	8,234	
15		Temperature Coefficient	Tcoff [μs]	0,750	
16		Acquisition Time	TACQ [μs]	10,984	
17					
18					
19		Tempo di conversione	Tad [μs]	Tempo di conversione [μs]	
20					
21		2Tosc	0,500	5,5	Non valido
22		4Tosc	1,000	11	Non valido
23		8Tosc	2,000	22	Valore accettabile
24		16Tosc	4,000	44	Valore accettabile
25		32Tosc	8,000	88	Valore accettabile
26		64Tosc	16,000	176	Valore accettabile
27					

Figura 4

Interrupt su conversione A/D

Il modulo ADC permette di acquisire i segnali in polling oppure su risposta all'interrupt al termine della conversione. In questo caso, una volta abilitato l'interrupt, il PIC setta il bit ADIF del registro PIR1 quando è possibile prelevare il dato dai registri di conversione. Nell'esempio che segue, si mostra la routine di abilitazione dell'interrupt per la gestione della conversione A/D

```
; Abilitazione interrupt
INTR_INIT
    banksel PIR1
    bcf PIR1,ADIF
    banksel PIE1
    bsf PIE1,ADIE

    banksel INTCON
    bsf INTCON,PEIE
    bsf INTCON,GIE

    return
```

Il risultato della conversione

L'ADC del PIC16F87xA è a 10 bit, il che significa che un registro a 8 bit è ovviamente insufficiente per contenere il risultato della conversione. Per risolvere il problema, Microchip ha predisposto due registri **ADRESL** e **ADRESH**, rispettivamente per contenere la parte bassa e la parte alta del risultato della conversione. Ma c'è qualcosa in più. Benché possa sembrare poco pratico, è possibile scegliere se allineare i 10 bit della conversione a partire dal bit 0 oppure dal bit 15, come illustrato in figura 5.

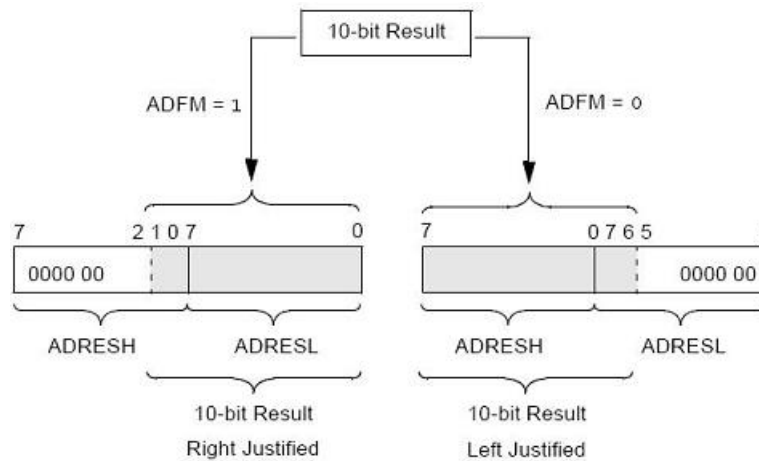


Figura 5

Per operare l'allineamento a destra o a sinistra è necessario agire sul bit **ADFM** del registro **ADCON1**.

Esempio di funzionamento

Per la configurazione del modulo ADC, l'avvio della conversione e la lettura dei risultati, è consigliabile seguire la guida che il datasheet mostra e che riporto qui nel seguito:

1. Configurazione del modulo ADC:

- Configurare i pin deputati alla conversione A/D e gli I/O digitali;
- Selezionare il canale A/D dal quale acquisire (registro ADCON0);
- Selezionare il valore opportuno per il conversion clock (registro ADCON0);
- Avviare il modulo ADC (registro ADCON0).

2. Configurazione dell'interrupt per l'ADC (se desiderato):

- a) pulire il bit ADIF;
- b) settare il bit ADIE;
- c) settare il bit PEIE;
- d) settare il bit GIE.

3. Attendere il tempo di acquisizione necessario.

4. Avviare la conversione:

- a) settare il bit GO/DONE (registro ADCON0)

5. Attendere il termine della conversione A/D, utilizzando:

- a1) il polling sul bit GO/DONE (a fine conversione, si riporta al valore zero)

oppure

- a2) attendere il termine della conversione A/D come risposta all'interrupt. a3) gestire la conversione come ISR e pulire il bit ADIF.

6. Leggere la coppia dei registri (ADRESH:ADRESL), pulire il bit ADIF se usato interrupt.

7. Avviare la conversione successiva partendo dal punto 1-a per selezionare un nuovo canale di acquisizione ed eventualmente eseguire anche il passo 1-b qualora l'impedenza della sorgente applicata al nuovo canale A/D sia diversa dalla precedente e richieda un valore diverso per il tempo di conversione.

Il progetto TestADC, realizzato in ambiente MPLAB, è scaricabile al seguente link:

[Progetto TestADC](#)

Sorgenti analogiche con impedenze alte

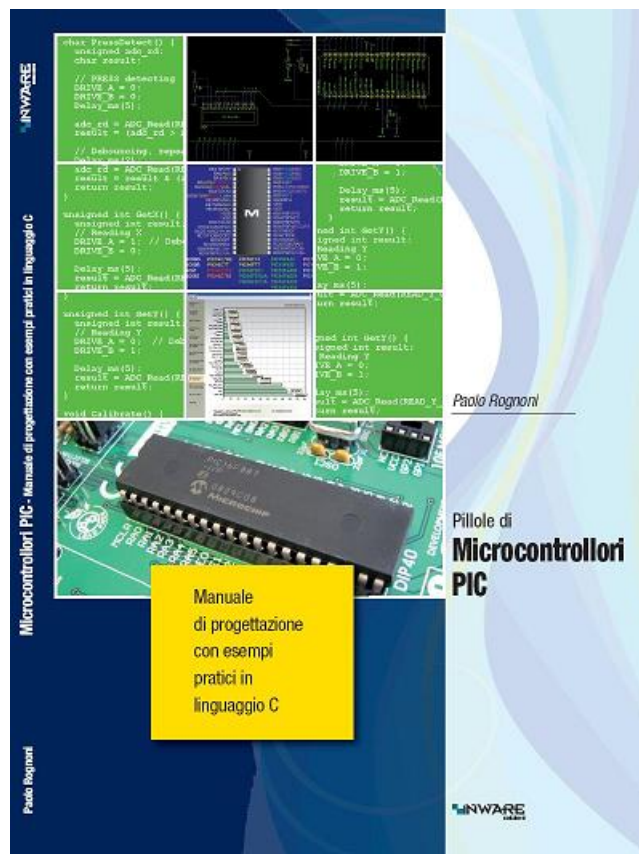
Il data sheet del PIC16F87xA indica chiaramente che non è opportuno collegare al PICMicro sorgenti analogiche con impedenze di uscita superiori a 2.5 kohm. Ma nella realtà potrebbe capitare di non poterne fare a meno. Come fare, in questo caso? Due sono le soluzioni di compromesso:

- 1. impiegare un buffer (ad esempio un OpAmp in configurazione inseguitore di tensione) interposto tra sorgente analogica e PIC;

2. utilizzare un tempo di acquisizione molto lungo, per dare modo al condensatore del modulo ADC di caricarsi.

Riferimenti

- **Ambiente di sviluppo MPLAB:** <http://www.microchip.com/mplab>
- **Datasheet PIC16F87xA:** <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>
- **Sito web PicExperience**
- **Collana "LO HAI MAI REALIZZATO CON UN PIC?":**
 - [Il contamarce](#)
 - [Una sorpresa musicale per Babbo Natale](#)
 - [Una tecnica antirimbalzo](#)
 - [Il dado elettronico](#)
 - [Un approccio ai timer dei PICMicro](#)
- [Pillole di microcontrollori PIC:](#)



Pillole di microcontrollori PIC

Estratto da ["http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:segnalianalogici_1_conversione_ad"](http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:segnalianalogici_1_conversione_ad)