# PIC18: ADC in assembly



In questa pagina è descritto come usare il Convertitore Analogico Digitale (Analog to Digital Converter, ADC) interno al PIC18 per misurare una tensione continua compresa tra massa ed alimentazione.

### La struttura interna dell'ADC del PIC18

Il convertitore AD del PIC18 è caratterizzato da:

- Una *risoluzione* di 10 bit, cioè la tensione in ingresso viene convertita in un numero intero compreso tra 0 e 1023 (= 2<sup>10</sup> - 1). Se lo preferite e ricordando la doppia definizione del termine, la *risoluzione* è pari a 3.2 mV se le tensioni di fondoscala sono impostate a 0 e 3,3 V
- 13 ingressi analogici dei quali uno solo alla volta può essere utilizzato
- Un tempo di conversione di circa 10-20 µs (aspetto non approfondito in questa pagina se non per evidenziare come sia *relativamente elevato*)

Di seguito lo schema interno, tratto dai data sheets:

- In verde sono evidenziati gli ingressi analogici, da ANO ad AN12. Questi pin hanno anche altre funzioni e quindi è necessario impostare esplicitamente come *ingressi analogici* quelli che si vogliono utilizzare per misurare tensioni
- În azzurro il multiplexer analogico (A-MUX) la cui funzione è quella di selezionare l'unico ingresso che, in un determinato istante, viene effettivamente utilizzato per la conversione.
- In rosso l'ADC vero e proprio. Si osservino due segnali fondamentali ad esso collegati:

- GO che ha il compito di far partire la conversione
- DONE che segnala quando la conversione è terminata
- ADON che ha il compito di "accendere" il convertitore
- Il risultato necessita di 10 bit, quindi due registri da 8 bit ciascuno, in giallo. I 6 bit non utilizzati possono essere a destra (giustificazione a sinistra) oppure a sinistra (giustificazione a destra), a seconda dell'impostazione di uno specifico flag. Nell'esempio si utilizzerà la seconda impostazione, forse la più *naturale*.



Ciascuno dei registri e dei bit indicati nello schema (CHS, ADON, GO/DONE, ADFM, ADRESH, ADRESL...), corrisponde ad un <u>SFR</u> o ad una sua parte o ad un singolo *flag*.

## Conversione analogico-digitale

Tutte le varie fasi della conversione sono effettuate sequenzialmente dal software (polling) e descritte accuratamente nel paragrafo **19.2.9** A/D **CONVERSION PROCEDURE** dei fogli tecnici. Di seguito il codice che mostra la conversione della tensione analogica collegata a RAO/ANO. Il risultato è memorizzato in una variabile di due byte, da visualizzare inserendo un apposito brakpoint.

### Impostare i canali analogici

La prima operazione da effettuare è quella di selezionare quale è il pin che verrà utilizzato come ingresso analogico; occorre disattivare i buffer digitali corrispondenti, sia in ingresso che in uscita, impostando un bit ciascuno nei due registri TRISA e ANSEL:

```
bsf TRISA, RA0 ; Imposta RA0/AN0 come ingresso
bsf ANSEL, ANS0 ; Imposta RA0/AN0 come ingresso analogico
(disattiva l'ingresso digitale)
```

I dettagli sono ovviamente descritti nei fogli tecnici. Le due figure seguenti mettono in evidenza queste informazioni per il pin RAO/ANO:

Pin	Function	TRIS Setting	I/O	l/O Type	Description
RA0/AN0/C12IN0-	RA0	0	0	DIG	LATA<0> data output; not affected by analog input.
		1	- 1	TTL	PORTA<0> data input; disabled when analog input enabled.
	AN0	AN0 1 I ANA ADC input channel 0. Default input configura affect digital output.		ADC input channel 0. Default input configuration on POR; does not affect digital output.	
	C12IN0-	1	I	ANA	Comparators C1 and C2 inverting input, channel 0. Analog select is shared with ADC.

TABLE 10-1: PORTA I/O SUMMARY

#### REGISTER 10-2: ANSEL: ANALOG SELECT REGISTER 1

	ANS0: RA0 Analog Select Control bit						
bit 7							bit 0
ANS7 <sup>(1)</sup>	ANS6 <sup>(1)</sup>	ANS5 <sup>(1)</sup>	ANS4	ANS3	ANS2	ANS1	ANS0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

ANS0: RA0 Analog Select Control bit

1 = Digital input buffer of RA0 is disabled

0 = Digital input buffer of RA0 is enabled

#### Impostare l'ADC

Occorre quindi configurare l'ADC, attraverso ben tre registri: ADCON0, ADCON1, e ADCON2.

Esamiliamoli:

#### ADCON2

Questo registro è diviso in tre campi evidenziati nella figura seguente da colori diversi, ognuno dei quali ha un diverso significato:

REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0
bit 7	ADFM: A/D Conversion Result Format Select bit 1 = Right justified 0 = Left justified						
bit 6	Unimplemented: Read as '0'						
bit 5-3	ACQT<2:0>: A/D Acquisition time select bits. Acquisition time is the duration that the A/D charge hold- ing capacitor remains connected to A/D channel from the instant the GO/DONE bit is set until conver- sions begins. 000 = 0 <sup>(1)</sup> 001 = 2 TAD 010 = 4 TAD 011 = 6 TAD 100 = 8 TAD 101 = 12 TAD 110 = 16 TAD						D charge hold- et until conver-
bit 2-0	ADCS<2:0>:	A/D Conversio	n Clock Select	t bits			
	000 = Fosc/2 001 = Fosc/8 010 = Fosc/3 011 = FRC <sup>(1)</sup> 100 = Fosc/4 101 = Fosc/6 111 = FRC <sup>(1)</sup>	2 (clock derived 6 4 (clock derived	from a dedicat from a dedicat	ed internal osc ed internal osc	tillator = 600 kH tillator = 600 kH	iz nominal) Iz nominal)	

• ADFM (in rosso) - Allineare il risultato a destra: nel risultato avranno significato solo gli otto bit presenti in ADRESL e i due meno significativi di ADREH, come evidenziato nella seguente figura:



- ACQT <0:2> (in giallo) Attendere un certo tempo prima di iniziare la conversione, sufficiente per acquisire il segnale. Il calcolo di questo tempo (T<sub>ACQ</sub>) non è banale e richiede qualche conoscenza di elettronica analogica nonché delle caratteristiche della sorgente di tensione; è descritto dalla EQUATION 19-1: ACQUISITION TIME EXAMPLE. Nell'esempio è impostato un valore medio (ACQT<0:2> = 100), adeguato in molti casi (nota 5)
- ADCS <0:2> (in azzurro) Impostare la corretta frequenza di clock da applicare al convertitore, né troppo elevata né troppo bassa. La scelta dipende dalla frequenza del processore, in questo esempio lasciata al valore più basso di 1 MHz

ADC Clock	Period (TAD)	Device Frequency (Fosc)					
ADC Clock Source	ADCS<2:0>	64 MHz	16 MHz	4 MHz	1 MHz		
Fosc/2	000	31.25 ns <sup>(2)</sup>	125 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 μs		
Fosc/4	100	62.5 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	1.0 μs	4.0 μs <sup>(3)</sup>		
Fosc/8	001	400 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 μs	8.0 μs <sup>(3)</sup>		
Fosc/16	101	250 ns <sup>(2)</sup>	1.0 μs	4.0 μs <sup>(3)</sup>	16.0 μs <sup>(3)</sup>		
Fosc/32	010	500 ns <sup>(2)</sup>	2.0 μs	8.0 μs <sup>(3)</sup>	32.0 μs <sup>(3)</sup>		
Fosc/64	110	1.0 μs	4.0 μs <sup>(3)</sup>	16.0 μs <sup>(3)</sup>	64.0 μs <sup>(3)</sup>		
FRC	x11	1-4 μs <sup>(1,4)</sup>	1-4 μs <sup>(1,4)</sup>	1-4 μs <sup>(1,4)</sup>	1-4 μs <sup>(1,4)</sup>		

TABLE 19-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES

Legend: Shaded cells are outside of recommended range.

### ADCON1

Questo registro permette di scegliere le due tensioni di riferimento alta e bassa, nell'esempio sono impostate a VDD e VSS (alimentazione e massa), la situazione più semplice:

U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_		VCFG1	VCFG0	—	_	_	—
bit 7							bit 0
bit 5	VCFG1: Nega 1 = Negative 0 = Negative	ative Voltage R voltage referer voltage referer	eference selection ace supplied examplied international international technologies in the selection of the	t bit ternally throug ternally by Vss	h VREF- pin.		
bit 4	VCFG0: Positive Voltage Reference select bit 1 = Positive voltage reference supplied externally through VREF+ pin. 0 = Positive voltage reference supplied internally by VDD						

#### REGISTER 19-2: ADCON1: A/D CONTROL REGISTER 1

#### **ADCONO**

Anche questo registro è diviso in tre campi evidenziati nella figura seguente da colori diversi, ognuno dei quali ha un diverso significato:

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0
bit 5-2	CHS<3:0>: Analog Channel Select bits 0000 = AN0						
bit 1	<ul> <li>GO/DONE: A/D Conversion Status bit</li> <li>1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle. This bit is automatically cleared by hardware when the A/D conversion has completed.</li> <li>0 = A/D conversion completed/not in progress</li> </ul>						
bit 0	ADON: ADC Enable bit 1 = ADC is enabled 0 = ADC is disabled and consumes no operating current						

REGISTER 19-1:	ADCON0: A/D CONTROL REGISTER 0
REGIOTER IO II.	ABGGHGI AB GGHTHGE REGIGTER G

- CHS <3:0> (in giallo) Scegliere il canale da utilizzare attraverso il multiplexer impostando, quattro bit . Ovviamente è necessario che tale canale sia stato precedentemente impostato come analogico
- ADON (in rosso) Alimentare l'ADC, settando in ADCONO il flag ADON (in rosso)
- GO/DONE (in azzurro) Dalla doppia funzione di avviare la conversione e verificare la sua fine. Questo flag è descritto nel prosieguo del paragrafo

Tutte queste operazioni di configurazioni vengono spesso fatte una sola volta, all'inizio del programma. Ovviamente è possibile una loro modifica durante l'esecuzione del programma, per esempio per leggere la tensione da un altro canale, oppure per spegnere l'ADC quando non serve e risparmiare quindi energia.

#### Leggere una tensione

La prima operazione è avviare la conversione, settando il flag GO:

bsf ADCON0, GO

... e attendere la sua conclusione, verificando il valore del flag  ${\tt DONE}~$ ed attendendo il suo azzeramento

```
attendi
btfsc ADCON0, DONE ; Conversione terminata?
bra attendi ; No, attendi ancora
```

Entrambe queste operazioni utilizzano un unico flag di ADCONO, con *doppio nome* e già evidenziato in azzurro nella precedente figura.

# A questo punto occorre leggere il risultato dai due registri ADRESH e ADRESL (due byte) e salvarlo nella variabile appositamente allocata.

Per visualizzare il risultato all'interno di MPLABX è necessario usare un breakpoint, per esempio in corrispondenza dell'istruzione sleep.