

NOTAS DE AULAS: MICROCONTROLADORES
Conversor Analógico Digital do PIC16F873

Prof. João Perea Martins
Dep. De Computação, FC–UNESP
E-mail: joao.perea@unesp.br

Registadores envolvidos no tópico:

1. ADCON0
2. ADCON1
3. ADRESH
4. ADRESL

1) REGISTRADOR ADCON0 (ENDEREÇO: 1FH)

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
							bit 0

ADCS1:ADCS0 *(selecionam o clock do AD)*

- 00 = FOSC/2
- 01 = FOSC/8
- 10 = FOSC/32
- 11 = Clock interno – Oscilador RC

O Tempo de conversão é de pelo menos 12 TAD

Configuração	ADCS1:ADCS0	Frequência de cálculo	TAD (us)
2TOSC	00	1.25MHz	0,8
8TOSC	01	5MHz	0,2
32TOSC	10	20MHz	0,05
RC	11	Varia	1-2

CHS2:CHS0 *(Selecionam o canal ou entrada de conversão)*

- 000 = canal 0, (RA0/AN0)
- 001 = canal 1, (RA1/AN1)
- 010 = canal 2, (RA2/AN2)
- 011 = canal 3, (RA3/AN3)
- 100 = canal 4, (RA5/AN4)
- 101 = canal 5, (RE0/AN5)(1)
- 110 = canal 6, (RE1/AN6)(1)
- 111 = canal 7, (RE2/AN7)(1)

Obs. As entradas 5,6 e 7 só estão disponíveis nos modelos PIC16F873/876

GO/DONE *(Controle e verificação da conversão)*

- 1 = A conversão do AD está em progresso
- 0 = O AD não está efetuando conversões

Obs. Quando esse bit é colocado em H por software, uma conversão é iniciada. Quando a conversão terminar, o microcontrolador coloca automaticamente esse bit em L

ADON: A/D On bit

1 = O A/D é ativado

0 = O A/D é desativado e isso diminui o consumo de energia.

2) REGISTRADOR ADCON1 (ENDEREÇO 9FH)

ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
------	---	---	---	-------	-------	-------	-------

ADFM (Determina o formato de ajuste do dado numérico dos 10 bits saída)

1 = Ajusta a direita

0 = Ajusta a esquerda

ADRESH								ADRESL							
MSB									LSB	0	0	0	0	0	0

ADRESH								ADRESL							
0	0	0	0	0	0	MSB									LSB

Tabela relativa aos bits PCFG3/PCFG0

PCFG3: PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VR +	VR-	CHAN
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VR+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VR+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	D	A	D	A	VDD	VSS	3/0
0101	D	D	D	D	VR+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VR+	VR-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VR+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VR+	VR-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VR+	VR-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VR+	VR-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	D	VDD	VSS	1/0
1111	D	D	D	D	VR+	VR-	D	A	RA3	RA2	1/2

A = entrada analógica

D = entrada Digital

CHAN = Número de pinos entradas de A/D e pinos para voltagem de referencia (VR)

Os pinos de entrada analógica devem ser configurados como Input (I) usando o TRISB;
Após uma conversão deve ser esperado um tempo de pelo menos 2 TAD para se iniciar uma nova conversão. Na prática, por segurança, colocamos um tempo bem maior.

Tad = 8*Tosc = 8*250ns (P/ osc de 4MHz) = 2Us

; ESTRUTURA BÁSICA DE PROGRAMAÇÃO DO ADC

```
; ***** CONFIGURO ADCON0
      BANKSEL ADCON0
AD0:  MOVLW      B'01000001'
      MOVWF      ADCON0      ; Bit 7,6: 8*Tosc / Bit 5,4,3: AN0 / Bit 2: GoDone
                                   ; Bit 1,0: Módulo AD ligado
                                   ; Após conversão o bit GoDone fica '1'
```

```
; ***** CONFIGURO ADCON1
      BANKSEL ADCON1
ADX:  MOVLW      B'10000000'      ; CONTINUO A CONFIGURAÇÃO
      MOVWF      ADCON1      ; Bit 7: ajusta posicao dos 10 bits
                                   ; Bit 6,5,4: NC
                                   ; Bit 3,2,1,0: portas (R0 como AD, Verf=Vdd)
```

```
; ----- FIM DA CONFIGURAÇÃO
```

; CONVERSÃO DE UM SINAL

```
      MOVLW      D'200'      ; INICIO DO DELAY
      MOVWF      DELAY
DLY1: NOP
      DECFSZ     DELAY,1
      GOTO      DLY1
      NOP      ; FIM DO DELAY
```

```
      BANKSEL ASCON0
      BSF      ADCON0,2      ; GoDone =1, inicia conversão
WAD1: BTFSC     ADCON0,2      ; CONVERSION DONE (bit=1) ?
      GOTO      WAD1      ; Não, então testa de novo
```

```
      BANKSEL ADRESL
      MOVF      ADRESH,W      ;Leitura do 2 bits MSB
      xxxxxx      ; Salvar/Usar o conteúdo de W
```

```
      BANKSEL ADRESL
      MOVF      ADRESL,W      ;Leitura dos outros 8 bits
      xxxxxx      ; Salvar/Usar o conteúdo de W
```

```
      MOVLW      D'200'      ; INICIO DO DELAY
      MOVWF      DELAY
DLY:  NOP
      DECFSZ     DELAY,1
      GOTO      DLY
      NOP      ; FIM DO DELAY
```