

Módulo 4 - ADC

O conversor Analógico para Digital (ADC) é um hardware interno que converte a variação de tensão de um pino em um valor digital para o microcontrolador. Isto permite que você possa criar rotinas de software baseada na variação de tensão de um pino.

Existem muitos cenários na eletrônica e no desenvolvimento de sistemas embarcados onde um conversor analógico para digital (ADC) se faz necessário. Abaixo estão alguns exemplos práticos de situações em que um ADC é utilizado:

1. Sensor de Temperatura:

- Sensores de temperatura, como o termistor, geralmente fornecem saídas analógicas proporcionais à temperatura. Um ADC é utilizado para converter essa informação analógica em um formato digital para que o microcontrolador ou processador possa processar e tomar decisões baseadas na temperatura medida.

2. Sistema de Controle de Motor:

- Em sistemas de controle de motores, sensores de posição e velocidade muitas vezes geram sinais analógicos que precisam ser convertidos para o domínio digital para permitir o controle digital preciso do motor.

3. Leitura de Sensores de Luz:

- Sensores de luz, como fotodiodos ou fototransistores, geram sinais analógicos em resposta à intensidade de luz. Um ADC é utilizado para converter esses sinais analógicos em valores digitais, permitindo que um microcontrolador tome decisões com base na luminosidade ambiente.

4. Sistema de Áudio:

- Em sistemas de áudio, sinais de áudio analógicos provenientes de microfones ou instrumentos musicais precisam ser convertidos para formato digital para processamento digital de sinal (DSP) ou armazenamento.

5. Monitoramento de Tensão de Bateria:

- Em sistemas alimentados por bateria, é comum usar um divisor de tensão para monitorar a carga da bateria. A tensão resultante é então convertida por um ADC para determinar o nível de carga da bateria.

6. Medição de Luz Ambiente em Displays:

- Alguns dispositivos, como smartphones e tablets, ajustam automaticamente o brilho da tela com base na luz ambiente. Sensores de luz ambiente fornecem informações analógicas que são convertidas por um ADC para ajustar a intensidade da luz da tela.

7. Instrumentação Médica:

- Em instrumentação médica, a leitura de sinais biomédicos, como ECG (eletrocardiograma) ou EEG (eletroencefalograma), requer a conversão de sinais analógicos para digitais para análise e exibição.

8. Controle de Processos Industriais:

- Em ambientes industriais, sensores analógicos são frequentemente usados para monitorar variáveis como pressão, fluxo e temperatura. Um ADC é essencial para converter esses sinais analógicos para valores digitais que podem ser processados por controladores industriais.

Nem todos os pinos do microcontrolador suportam essa funcionalidade. No sumário de pinos, presente no datasheet, vemos todos os pinos compatíveis com o Conversor ADC. Geralmente eles são nomeados como AN0, AN1, etc.

A entrada de tensão geralmente é de 0 a 5V (se ultrapassar este valor, pode queimar o pino) e o valor digital gerado é de 0 a 1023 (10 bits). O acesso a este valor é feito por meio de um registrador.

Processo de configuração

Para configurá-lo, precisamos considerar as seguintes funcionalidades:

- A configuração do PORT
- Seleção do canal (pino do microcontrolador)
- Referência de tensão
- Fonte de clock
- Formatação do resultado (8 bits ou 10 bits)

Utilizaremos o pino RA0 como conversor analógico, ele é o canal AN0 do conversor, configuramos ele como entrada e configuramos também o ANSELA.

```
TRISAbits.TRISA0 = 1;
```

```
ANSELA = 0x01;
```

O **ANSEL** é o registrador que habilita as entradas analógicas dos pinos (quando o pino é compatível). Em microcontroladores que não possuem o ANSEL, você utilizará somente os registradores ADCON. Quando um pino recebe ANSEL = 1, o seu circuito digital é desabilitado e ele passa a ser analógico.

Para um PIC18F45K22, os registradores são responsáveis pelas seguintes funções:

- ADCON0 – Seleciona o canal, Inicia a conversão analógica.
- ADCON1 – Define uma referência de tensão.
- ADCON2 – Definem as características do ADC.

O código abaixo realiza uma leitura analógica no pino RA0 (AN0):

```
ADCON1 = 0; // Configuração geral
ADCON2 = 0xAA; // Configuração geral
ADCON0 = 0; // Seleciona aqui o canal
ADCON0bits.ADON = 1; // Habilita o conversor
ADCON0bits.GO = 1; // Inicia a conversão
while(ADCON0bits.GO); // Aguarda até terminar
```

A leitura ocorre somente em um instante, ou seja, lê um único valor equivalente à tensão do pino no momento que o bit GO foi colocado em 1. O valor salvo fica em dois registradores: ADRESH e ADRESL. Estes registradores permitem uma formatação do resultado de 10 bits.

Lendo o resultado

A melhor forma de traduzir o valor que está em dois registradores diferentes em somente uma variável é utilizando uma union:

```
union {
    char leitura[2];
    int result;
} adc;

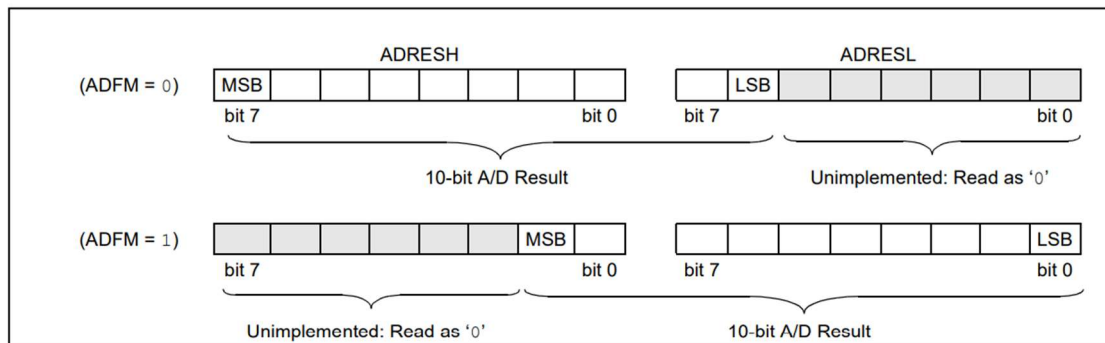
adc.leitura[1] = ADRESH;
adc.leitura[0] = ADRESL;

adc.result; // Valor final
```

Conceitos avançados de ADC

Formatação do resultado

O resultado retorna como uma palavra de 10 bits, ou seja, ele consome um endereço de memória completo e mais 2 bits de outro endereço. É possível formatar este resultado para a direita ou para a esquerda.



O bit ADFM está no registrador ADCON2. A vantagem de utilizar a justificação à esquerda é que os 2 bits menos significativos serão excluídos, tornando a leitura mais precisa. Isso vem ao custo de uma perda de resolução de leitura. Utilizando nesta configuração, basta utilizar o ADRESH como seu resultado final da conversão.

Utilizando a formatação à direita (ADFM = 1) você terá o resultado dividido nos dois registradores. Utilize uma union para transformá-lo em um valor int.

ADCON0

Este registrador pode variar de MCU para MCU. Do caso do 18F45k22, ele define o canal analógico e aciona a conversão.

O acionamento da conversão é feito primeiramente com ADON = 1. Depois, realize o trigger com GO/DONE = 1. Quando a conversão encerrar, o bit GO/DONE é automaticamente resetado.

ADCON1

Este registrador define a referência de tensão do conversor. Por padrão, a referência é a própria alimentação do microcontrolador. Com os bits **PVCFG** e **NVCFG** o programador define a referência de tensão.

ADCON2

Este registrador define a formatação do resultado, a quantidade de TAD's e a frequência do conversor. A quantidade de TAD's geralmente é 12 ou 16 TAD's, alterar diminuir este valor pode tornar o processo mais rápido, porém sem precisão. A frequência depende do cristal oscilador utilizado, seguindo a tabela:

TABLE 17-1: ADC CLOCK PERIOD (T_{AD}) Vs. DEVICE OPERATING FREQUENCIES

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