

Programação em BASIC para o PIC

Utilizando o AD do PIC

Vitor Amadeu Souza

Introdução

Nesta série de artigos apresentados, veremos neste capítulo como medir uma tensão analógica na entrada do microcontrolador e transmitir este resultado via canal serial de comunicação RS232 o resultado da conversão para o PC. O microcontrolador PIC18F442 possui internamente um *conversor analógico para digital* de 10 bits. A idéia deste exemplo será a de transmitir de 1 em 1 segundo, o valor presente na entrada RA0 e através do Communication Terminal do mikroBASIC poder visualizar o resultado da conversão AD. Como o AD do PIC18F442 é de 10 bits e 2^{10} é igual a 1024, podemos definir a resolução de cada bit nesta conversão. Digamos que o nosso sistema irá variar a tensão de 0 a 5 Vcc. Para acharmos a tensão referente a alteração de uma unidade de AD no microcontrolador, devemos dividir a diferença entre as tensões de referência positiva e negativa pela resolução do AD. Observe o box 1 para verificar melhor esta fórmula.

$$R_{bit} = (V_{ref+} - V_{ref-}) / \text{Resolução do AD}$$

Box 1 – Cálculo da Resolução de AD

Desta forma, conforme a tensão analógica presente na entrada RA0 variar 4,88 mV, o resultado da conversão AD variará 1 bit do seu resultado. Fique claro que quando o resultado da conversão for 1023, teremos na entrada o equivalente a 5 V, pois este é o resultado máximo de conversão e quando o resultado da conversão for 0, teremos 0 Vcc na entrada.

As entradas dos pinos analógicos estão presentes nos pinos RA0, RA1, RA2, RA3, RA5, RE0, RE1 e RE2 e quando eles são usados na forma analógica, nos referimos aos mesmos como AN0, AN1, AN2, AN3, AN4, AN5, AN6 e AN7. Na placa didática PICLAB18F442 temos dois trimpots, sendo o primeiro ligado ao RA0 e o segundo no RA1. De acordo com a tensão presente no pino RA0, este resultado será convertido para digital e enviado em seqüência para o PC através da RS232. Com o resultado que surgir no PC, poderemos saber perfeitamente a tensão que está presente na entrada analógica. Digamos que seja observado o resultado 500 no Communication Terminal. Como sabemos através da fórmula anterior que cada bit equivale a tensão de 4,88 mV, basta multiplicarmos o resultado da conversão por este valor e encontrarmos a tensão real na entrada do microcontrolador. Desta forma, encontraremos a tensão de 2,44 V que é a tensão encontrada neste caso. Apesar do AD do PIC ser de 10 bits, neste exemplo utilizaremos somente 8, de forma com que este valor seja enviado em um único

byte para o PC. A lógica de funcionamento permanece a mesma, porém agora a resolução será 256 passos em função da resolução adotada.

Recursos de Hardware

O esquema elétrico para execução deste experimento pode ser observado na figura 1. Observe que no pino RA0 do microcontrolador está conectado o trimpot que será utilizado para medirmos a tensão analógica ajustada. A parte da comunicação RS232 permanece a mesma do artigo passado, mantendo desta forma a utilização do MAX232 como circuito de conversão.

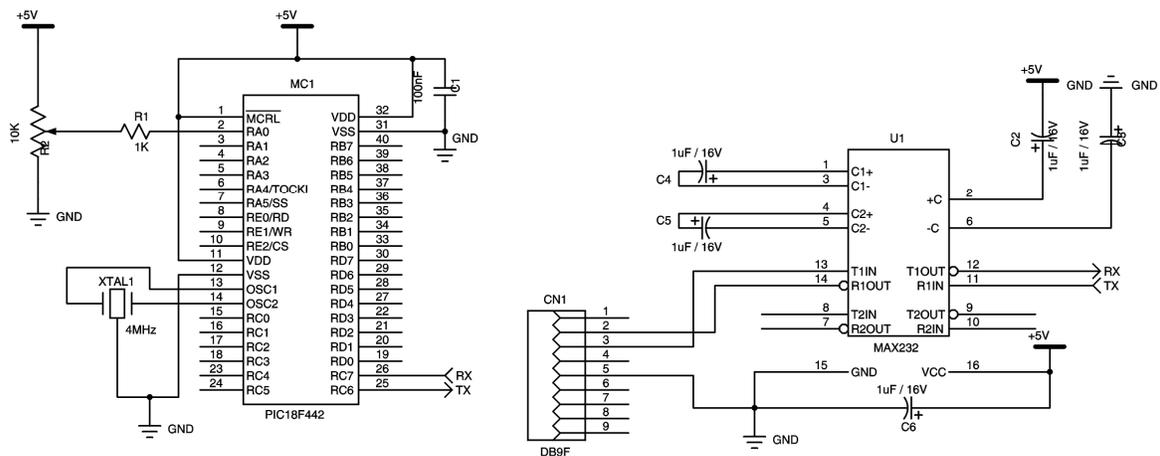


Figura 1 – Esquema Elétrico do Exemplo

Fluxograma

A lógica de funcionamento deste exemplo está apresentado na figura 2. Note que primeiramente a USART é configurada e após este processo, de 1 em 1 segundo o programa fica enviando pelo canal serial o resultado da conversão AD.

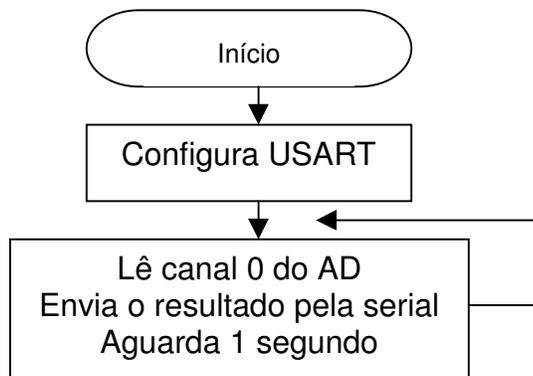


Figura 2 – Fluxo de funcionamento do exemplo

Recursos de Software

Neste exemplo veremos uma nova função disponibilizada pelo mikroBASIC que permite a medição de uma entrada analógica. Esta função está apresentada na tabela 1.

`adc_read (canal a ser lido)`

Tabela 1 – Função de leitura de AD

Quando esta função é chamada, devemos passar como parâmetro o canal de AD a ser lido. No PIC18F442 existem 8 canais que variam de 0 a 7 referentes aos pinos AN0 a AN7. No nosso caso, esta função deverá ser chamada da forma apresentada na tabela 2 permitindo assim a leitura do canal 0.

`adc_read (0)`

Tabela 2 – Chamando a função de leitura de AD

Software

O software completo que permitirá o teste deste exemplo está apresentado no box 2.

```
program ad
main:
    usart_init(9600)
repete:
    usart_write(adc_read(0)/4)
    delay_ms(1000)
    goto repete
end.
```

Box 2 – Listagem do Programa

Logo no início do programa, é configurada a taxa de comunicação utilizada pelo exemplo que neste caso é de 9600 bps. Em seguida, podemos visualizar que o resultado da conversão AD é enviado pelo canal serial através da função `usart_write`. Veja que a leitura do canal de AD é dividida por 4 antes do valor ser enviado. Isto é necessário pelo fato do resultado retornado pela função de leitura ser em 10 bits e como somente os oito bits mais significativos serão enviados pelo canal serial, houve a necessidade de retirar os dois bits menos significativos do resultado da conversão. Isto foi feito dividindo o valor retornado por esta função

por 4 e logo em seguida enviando o mesmo via RS232. Na próxima linha de programa encontramos um retardo de programa através da função `delay_ms`. Finalmente o programa entra em loop, repetindo este processo infundavelmente através do comando `goto`.

Após escrever este programa e compilar o mesmo, transfira o arquivo hex gerado pelo compilador para a placa didática PICLAB18F442. Abra neste momento o Communication Terminal e veja o resultado da conversão analógica chegar via RS232.

Conclusão

Apesar de toda a digitalização em que vivemos hoje, não podemos esquecer que no mundo onde vivemos nos deparamos todos os dias com as variáveis analógicas. Pressão, temperatura, nível, vazão, luminosidade são apenas algumas das variáveis analógicas disponíveis. Trabalhar com transdutores corretos, que transformem estas grandezas em algum equivalente elétrico e conseqüentemente a conversão deste valor para uma forma digital em que um microcontrolador consiga processar já é algo completamente possível, pelo fato dos modernos microcontroladores já terem embutidos em sua própria pastilha esta periférico.