



Experimento 8 – Conversor A/D

OBJETIVO:

Iremos estudar o princípio de conversão analógica-digital e seu uso no MSP430. Aproveitaremos o estudo do conversor para estudar transferência de dados utilizando o periférico DMA.

INTRODUÇÃO:

Conversores A/D são dispositivos de interface entre o mundo analógico e o mundo digital. Eles permitem que um microcontrolador colete dados de transdutores que convertem grandezas físicas (como som, luz, temperatura, pH) no formato de tensão.

Neste roteiro, iremos estudar o princípio da conversão A/D. Para isso, utilizaremos divisores resistivos (na forma de potenciômetros) na entrada do conversor. Dessa forma, iremos realizar leituras de tensão proporcionais à posição do potenciômetro.

Um dispositivo simples que aplica esse princípio é o manche de um joystick. Esse dispositivo é tipicamente composto por dois potenciômetros, um para o eixo x e outro para o eixo y. Os dois terminais de leitura da posição do eixo x e y são na verdade valores de tensão entre uma referência positiva (ex: 3,3V) e uma referência negativa (ex: 0V).

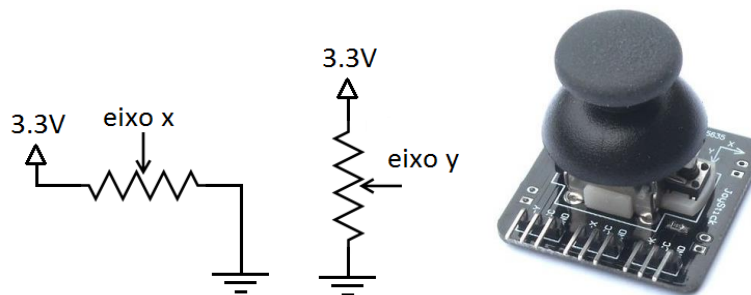
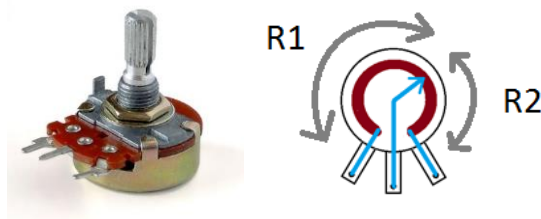


Figura 1 – Joystick com dois potenciômetros, um para cada eixo.

O MSP430 possui um conversor A/D integrado de 12-bits que funciona por aproximações sucessivas. Num primeiro momento, o sinal de entrada é amostrado e a tensão da entrada carrega um capacitor que guarda a informação da tensão. Numa segunda etapa, o conversor testa bit a bit, começando do MSB até o LSB para descobrir qual é o valor binário que representa aquele nível de tensão. A figura 2 mostra as etapas de conversão de um caso simplificado de apenas 3 bits.



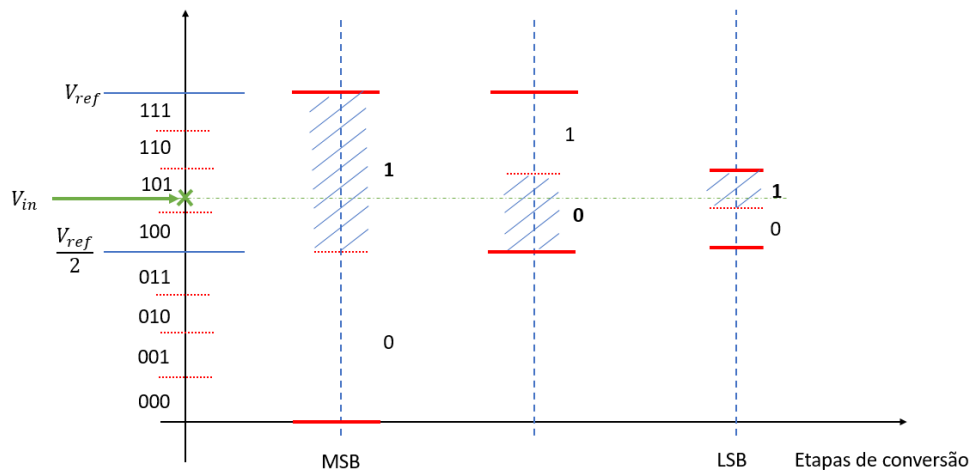


Figura 2 – Etapas de conversão de um conversor por aproximações sucessivas

No exemplo da figura 2, a tensão de entrada V_{in} está acima da metade da referência. Assim, na primeira etapa de conversão, o conversor A/D atribui o valor “1” ao MSB. Em seguida, o conversor divide a escala da etapa anterior (região hachurada) em duas e verifica se a entrada se encontra na parte superior ou inferior. No exemplo a tensão se encontra na parte inferior da segunda etapa e então o segundo bit é “0”. Por fim, repete-se esse procedimento até que todos os bits sejam definidos. No exemplo, a palavra binária que representa a tensão V_{in} é **101**.

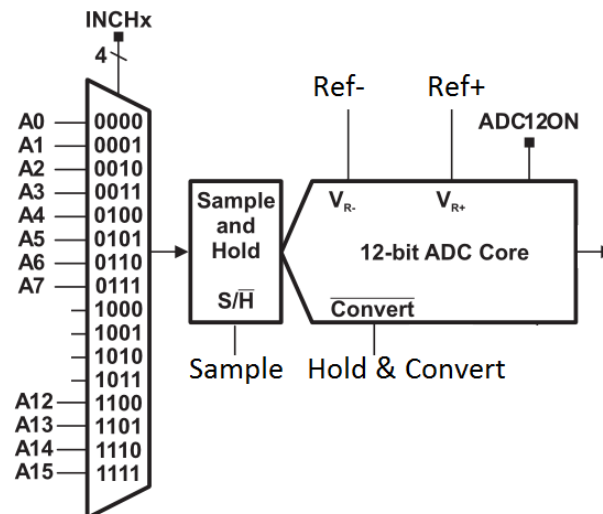


Figura 3 – Conversor AD do MSP430

Para que essa conversão seja feita em etapas sucessivas, é necessário que a entrada V_{in} esteja estável durante todo o processo de conversão. Isso é garantido através de um capacitor na entrada do conversor. Devido à natureza desse elemento, o capacitor leva um certo tempo para acumular carga, então é necessária uma etapa de amostragem do sinal antes de iniciar a conversão. Esse processo é conhecido como *sample and hold*. A figura 4 modela a entrada do conversor A/D. O resistor R_s representa o multiplexador de entrada e o capacitor C_i representa o bloco “Sample and Hold” da figura 3.

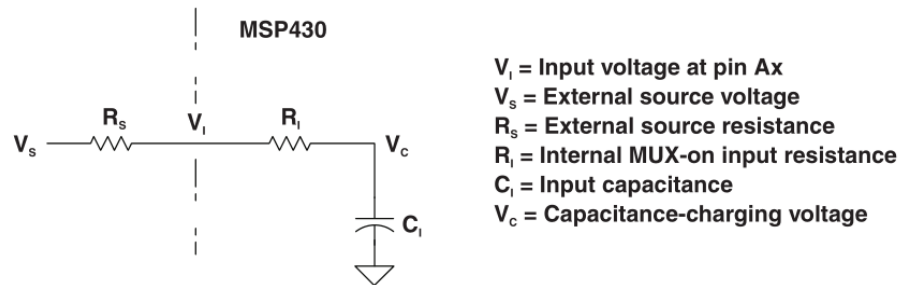


Figura 4 – Amostragem da entrada

A resistência de entrada externa ao MSP430 na porta do conversor AD influencia o tempo de amostragem, ou seja, o tempo que leva para o capacitor da entrada carregar. Inicie o seu projeto calculando o tempo de carga do capacitor de entrada do conversor AD seguindo a seguinte expressão:

$$t_{sample} > (R_s + R_i) \times \ln(2^{n+1})C_i + 800ns$$

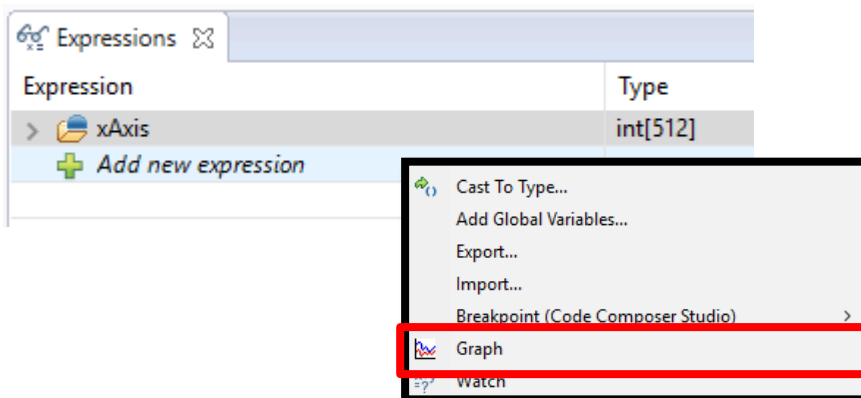
Considere a resistência interna como sendo 1,8k (resistência do mux de entrada do conversor) e o valor da capacitância como sendo no máximo 25pF. A resistência R_s deve ser o valor do potenciômetro no pior caso de medida, ou seja, em 50%.

$$R_s = \frac{R_{pot}}{2} \parallel \frac{R_{pot}}{2} = \frac{R_{pot}}{4}$$

PEDIDOS:

Programa 23:

Configure o módulo do conversor A/D para amostrar a tensão de entrada que representa apenas um eixo da posição do joystick (ou de um potenciômetro qualquer). Não se esqueça de ajustar os limites superior e inferior da tensão de referência para maximizar a precisão da leitura. Configure um timer para realizar essa amostragem a cada 10ms. Grave num vetor de memória os últimos 5 segundos de amostragem. Você pode plotar o resultado da leitura no próprio CCS colocando o vetor na aba de expressões e clicando com o botão direito, selecione a opção "Graph".





Programa 24:

Em muitas situações, copiar regiões de memória de uma posição à outra consome tempo que a CPU poderia estar utilizando para realizar outras operações. O periférico DMA (Direct Memory Access) permite que cópias sejam realizadas sem intervenção do processador. DMA é a versão em hardware da função `memcpy()` muitas vezes encontradas em programas C ou C++.

O conversor A/D tem uma memória interna de até 16 posições para leituras consecutivas dos canais de entrada. Para este programa iremos ler simultaneamente (ou quase) o valor da posição dos dois eixos do potenciômetro. Os dois valores devem ser em seguida movidos para uma posição na memória. Utilize o DMA para não ocupar o processador com essa cópia (mesmo que não seja uma cópia tão grande assim).

Mantenha a mesma taxa de amostragem do exercício anterior. Configure o conversor AD para realizar a leitura de dois canais consecutivos de uma só vez. Ao final da conversão dos dois valores de entrada, configure um trigger que irá iniciar o DMA a realizar a cópia dessas leituras para a memória.

Programa 25:

Use a informação da posição dos dois potenciômetros para controlar a luminosidade dos dois LEDs da launchpad. Se a posição do joystick estiver no centro, os dois LEDs devem estar em 50% da luminosidade. Use o eixo X para controlar a luminosidade do LED vermelho (LED1) e o eixo Y para controlar a luminosidade do LED verde (LED2). Trabalhe com um sinal PWM de 100 passos (de 0% até 100%) e corrigido para a sensibilidade do olho humano e com período de 20 ms. Note que você pode usar o gerador de PWM do MSP para acionar o LED2 (P4.7), mas precisará construir por software o PWM para acionar o LED1 (P1.0).

RELATÓRIO

O relatório é individual, e deve ser entregue impresso (ou feito à mão). Em hipótese alguma será admitida a entrega do relatório de forma eletrônica.

Questão 1 (2,5 pontos)

Mostre como você calculou o tempo necessário para amostrar o sinal de entrada e consequentemente carregar o capacitor de entrada do conversor AD. Qual é a configuração do conversor AD que reflete esse resultado.

Questão 2 (2,5 pontos)

Como você configurou a taxa de amostragem do sinal de entrada? Se estivéssemos trabalhando com um sinal de voz (até 3kHz), o que seria necessário modificar no seu programa?

Questão 3 (2,5 pontos)

Explique o funcionamento do periférico DMA. É possível que a CPU e o DMA escrevam na memória RAM ao mesmo tempo?

Questão 4 (2,5 pontos)

O que é a configuração “round robin” dos canais do periférico DMA? Qual o objetivo de usar tal configuração?