

Conversor Analógico Digital do MSP430G2553

O MSP430G2553 possui um conversor analógico-digital de alta performance. As principais características deste conversor são:

- conversor por aproximação sucessiva de 10 bits;
- taxa de conversão de até 200ksps
- tempo de sample-and-hold programável;
- clock do conversor selecionado por software;
- Possui um clock interno de 5MHz específico para o ADC (ADC10OSC), mas este clock não é muito estável;
- início da conversão por software ou pelo TIMER A;
- tensões de referência positiva e negativa interna ou externa;
- tensão de referência interna selecionadas por software entre 1,5v ou 2,5v;
- 8 entradas analógicas multiplexadas;
- sensor de temperatura interno acoplado à entrada 10;
- O valor convertido por ser representado em binário puro ou em complemento de 2;
- Possui a capacidade de armazenar os valores convertidos diretamente na memória, sem intervenção da CPU (através do módulo DTC – Data Transfer Controller);
- Pode ser colocado em modo de baixa potência independentemente dos outros blocos do MSP430;

Funcionamento

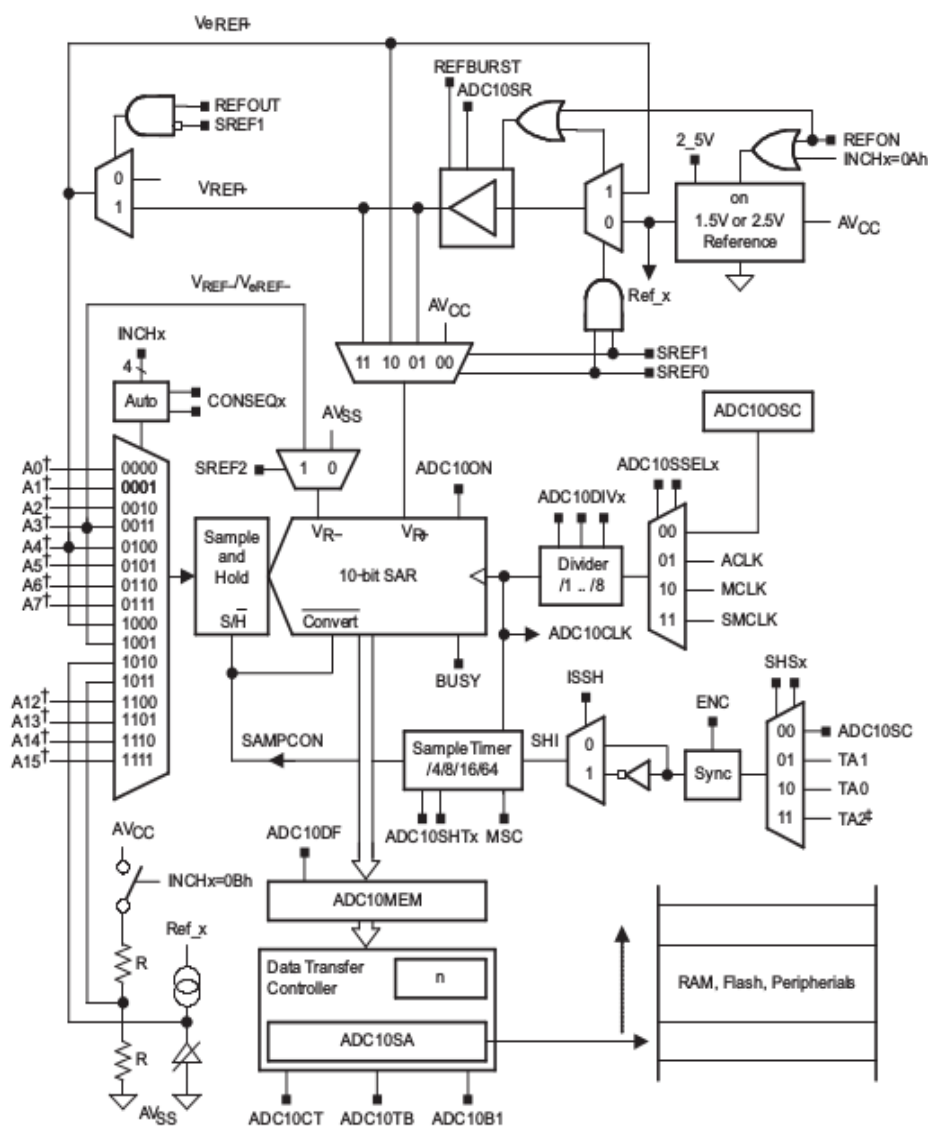
O conversor do MSP430G2553 faz a conversão do valor de tensão analógica presente na entrada selecionada em um valor digital de 10 bits. O resultado pode estar no formato binário ou no formato em complemento de dois. Este resultado estará disponível no registrador ADC10MEM após o término da conversão que é indicada pelo bit ADC10IFG (bit 2) do registrador ADC10CTL0. O bit ADC10BUSY (bit 0) do registrador ADC10CTL1 indica, quando setado, que existe uma conversão em andamento.

O valor analógico de entrada vem de um circuito multiplexador analógico de 12 entradas, sendo que oito destas entradas estão ligadas aos pinos do microcontrolador e as outras quatro entradas estão ligadas a fontes internas, como por exemplo o sensor de temperatura interno ao dispositivo.

Os valores de tensão de referência do conversor analógico/digital são dados pelas tensões V_{R+} e V_{R-} , que são selecionadas pelos bits SREF X (bits 13, 14 e 15) do registrador ADC10CTL0. Na página 553 de (MSP430X2XX) estão discriminadas as oito opções possíveis para estes valores.

O sinal de clock é usado tanto para realizar a conversão como para controlar o tempo de amostragem do sinal analógico. O clock pode ser selecionado entre as fontes SMCLK, MCLK, ACLK e o clock interno (ADC10OSC) e pode ter o seu valor dividido por um valor entre 1 e 8. O módulo DTC (Data Transfer Controller) permite enviar o valor obtido na conversão (presente no registrador ADC10MEM) diretamente para uma posição de memória interna sem a necessidade de intervenção da CPU. Com isso é possível realizar a conversão de diversos valores analógicos e armazená-los na memória para processamento em blocos.

O diagrama em blocos do conversor analógico-digital pode ser visto na figura 1.



†Channels A12-A15 are available in MSP430F22xx devices only. Channels A12-A15 tied to channel A11 in other devices. Not all channels are available in all devices.
‡TA1 on MSP430F20x2, MSP430G2x31, and MSP430G2x30 devices

Figura 1. Diagrama em blocos do conversor analógico-digital do MSP430. Fonte (MSP430X2XX).

O conversor analógico-digital do MSP430 possui 4 modos de operação:

1. conversão em apenas um canal, sem repetição: realiza uma única conversão, após ser solicitado pela CPU. Quando a conversão termina, o conversor permanece parado;
2. conversão em uma sequência de canais sem repetição: uma única série de diferentes entradas analógicas será realizada;
3. conversão em apenas um canal repetidamente: a conversão é realizada em um único canal, mas de maneira contínua. Quando uma conversão termina, outra é iniciada em sequência
4. conversão em uma sequência de canais repetidamente: a sequência de conversões nas diversas entradas é realizada de maneira contínua. Quando uma sequência termina outra inicia automaticamente.

Configuração do Conversor Analógico-Digital

O conversor analógico-digital do MSP430 possui dois registradores de controle: ADC10CTL0 e ADC10CTL1 e dois registradores de habilitação das entradas analógicas: ADC10AE0 e ADC10AE1. Observe que para o ADC10, os registradores P1SEL e P1SEL2 não são necessários. Somente o registrador ADC10AE0 é necessário para fazer a conexão entre o pino e a entrada do multiplex do conversor AD (veja configuração dos pinos no datasheet do MSP430G2553).

Existem mais 3 registradores para controle da transferência automática dos resultados da conversão do registrador ADC10MEM para uma região de memória RAM do MSP430. Estes registradores são: ADC10DTC0, ADC10DTC1 e ADC10SA. Consultar o documento MSP430x2xx Family User's Guide – SLAU144J (MSP430X2XX), pág. 552 e seguintes, para ver os detalhes dos registradores de configuração e uso do MSP430.

Exemplo:

Faça um programa em C para realizar a conversão analógica para digital do valor presente no pino P1.5 do MSP430G2553. Apresente o resultado em quatro faixas diferente conforme a tabela 1:

Tabela 1: Faixa de valores do ADC.

FAIXA	SAÍDA
De 0v a 0,9v (0x0000 a 0x00FF)	Led vermelho e led verde apagados
De 1,0v a 1,8v (0x0100 a 0x01FF)	Led vermelho apagado led verde aceso
De 1,9v a 2,7v (0x0200 a 0x02FF)	Led vermelho aceso e led verde apagado
Acima de 2,9v (0x0300 a 0x03FF)	Led vermelho e led verde acesos

Resolução

Na função main, na **parte de configuração** do MSP430:

1. configurar os pinos dos LEDs como saída e colocar nível lógico 0 para apagar os LEDs;

2. configurar a função secundária do pino P1.5, para que seja a entrada do conversor AD (setar o bit correspondente do registrador ADC10AE0).

```
ADC10AE0 |= BIT5; // configura o pino P1.5 como entrada do ADC10
```

3. configurar o registrador ADC10CTL1 para:

- selecionar entrada 5 do conversor AD (INCH_x);
- selecionar *sample-and-hold* controlado por software (SHS_x);
- determinar o fator de divisão do clock (entre 1 e 8, neste caso pode ser 8 - ADC10DIV_x);
- selecionar a fonte de clock do ADC (ADC10SSEL_x);
- selecionar o modo de operação do ADC (neste caso como entrada única e conversão simples, isto é sem repetição - CONSEQ_x);

```
ADC10CTL1 |= INCH_5 + SHS_0 // configura entrada 5 do ADC  
          + ADC10DIV_7      // seleciona divisao do clock por 8  
          + ADC10SEL_0      // seleciona SMCLK  
          + CONSEQ_0;      // seleciona modo 0
```

4. configurar o registrador ADC10CTL0 para:

- definir a tensão de referência como sendo VCC e VSS (SREF_x);
- determinar o tempo de amostragem para 4 pulsos de clock (sample-and-hold - ADC10SHT_x);
- ligar o ADC (ADC10ON);
- ativar a geração de interrupção pelo ADC (ADC10IE);

```
ADC10CTL0 = SREF_0      // seleciona Vcc como VR+ e GND como VR-  
          + ADC10SHT_0 // tempo de sample-and-hold de 4 ciclos de  
                      // clock  
          + ADC10ON     // ativa o canversor  
          + ADC10IE;    // habilita interrupcao do ADC10
```

5. habilitar as interrupções (ativar GIE);
6. Iniciar a primeira conversão (ativar os bits ENC e ADC10SC de ADC10CTL0 simultaneamente).

```
ADC10CTL0 |= ENC + ADC10SC; // inicia a conversao
```

No loop infinito:

1. nada é feito neste caso.

Na rotina de interrupção:

1. definir o vetor de interrupção como ADC10_VECTOR (em C: `#pragma vector = ADC10_VECTOR`);
 2. dentro da função de interrupção:
 1. realizar a leitura do valor convertido (presente no registrador ADC10MEM);
 2. verificar em qual das faixas o valor da conversão se enquadra e acender ou apagar os LEDs conforme a tabela 1;
 3. Iniciar uma nova conversão (ativar os bits ENC e ADC10SC de ADC10CTL0 simultaneamente como feito no item 6).
-

Exercício (valendo a AP2):

Faça um programa em C que realize a conversão analógica para digital do valor presente no pino P1.5 do MSP430G2553 e gere um sinal PWM, no pino P1.2, com período de 20ms e nível alto **proporcional** ao valor convertido, dentro da faixa de 1ms a 2ms (veja as figuras 2, 3, 4 e 5 abaixo), para controlar um servo motor de aeromodelismo:

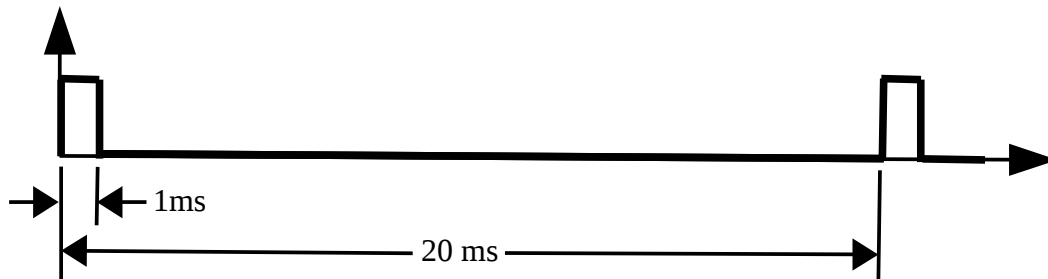


Figura 2: Forma de onda para o valor do ADC = 0x0000.

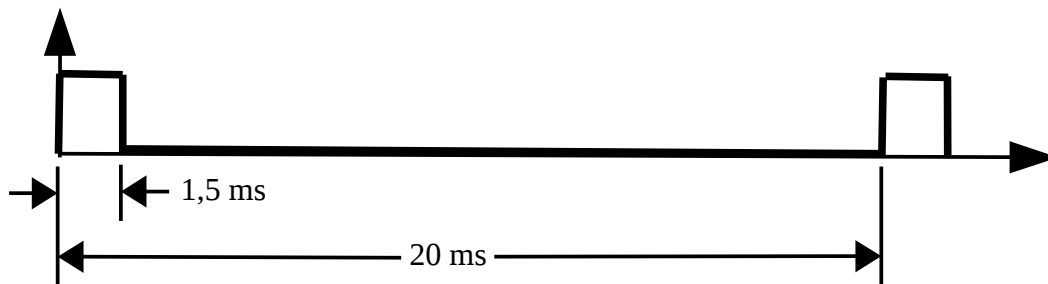


Figura 3: Forma de onda para o valor do ADC = 0x01FF.

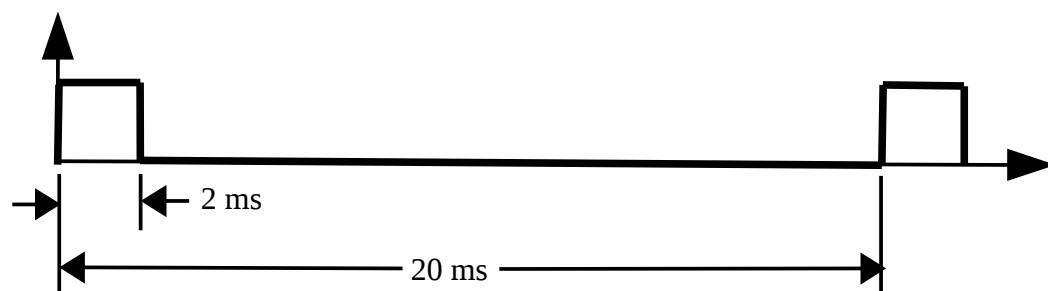


Figura 4: Forma de onda para o valor do ADC = 0x03FF.

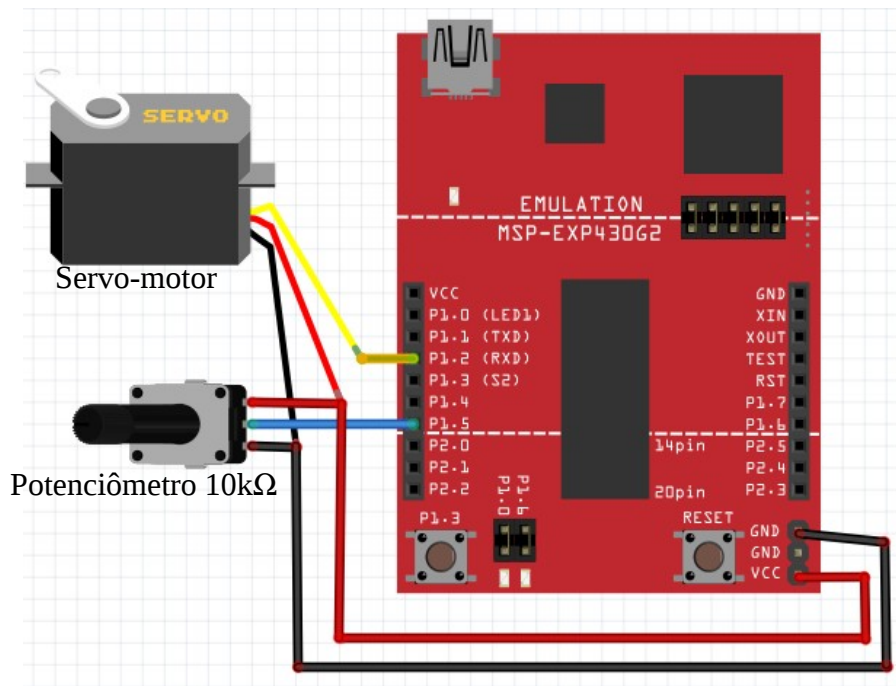


Figura 5. MSP-EXP430G2 controlando um servo-motor.

Referências:

- (MSP430X2XX) MSP430x2xx family user's guide. Disponível em:
<<http://www.ti.com/lit/ug/slau144j/slau144j.pdf>>. Acessado em 17 nov. 2014.
- Hitec HS-300 Servo Specification and Reviews. Disponível em:
<<http://www.servodatabase.com/servo/hitec/hs-300>>. Acessado em 04 maio de 2015.