PSI3441 – Arquitetura de Sistemas Embarcados

Configuração do DMA no PE

Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – grehder@lme.usp.br

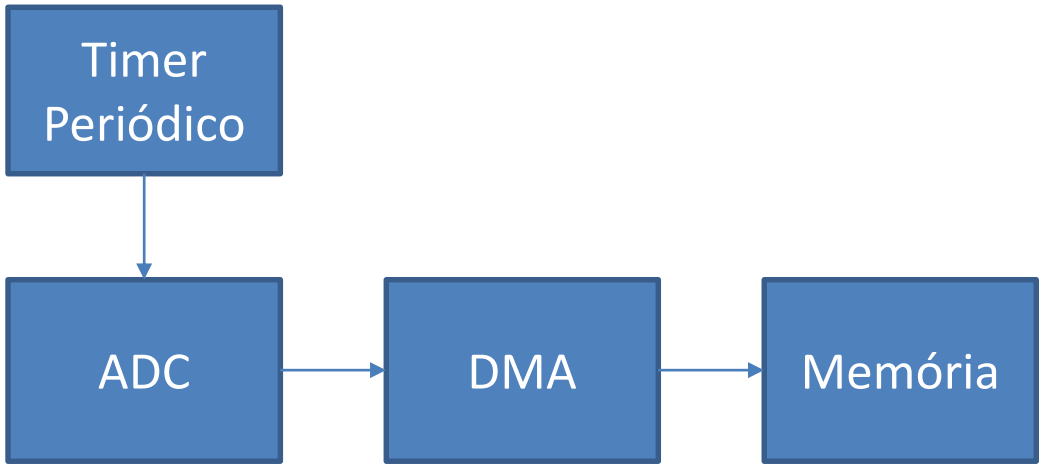
Primeiro Semestre de 2017

Objetivo

- Configurar o DMA para transmitir os dados da aquisição analógica (ADC) para a memória.

Procedimento

- ADC realiza a conversão sincronizado (triggered) pelo timer periódico.
- O ADC ao terminar a conversão, requisita a transferência de dados ao DMA
- O DMA executa a transferência para a memória
- Ao terminar a transferência, os flags são limpados e o número de bits resetado



Procedimento

- Configure um timer e certifique-se que ele está funcionando
- Configure um canal de ADC e certifique-se que ele está sendo disparado pelo timer. Não esqueça de inicializar o trigger (AD1_EnableIntChanTrigger();)
- Configure o DMA

Configuração do ADC

- Configurar canal
- Tempo de conversão
- Configurar Trigger (TPM0 overflow, por exemplo)
- Para habilitar a requisição de transferência ao controlador do DMA coloque a expressão no main (não em no loop infinito do main):

```
ADC0_SC2 |= ADC_SC2_DMAEN_MASK;    // DMA Enable
```

Obs: Habilitar o DMA pelo componente ADC resulta em erro!

Configuração do DMA

Utilize o componente Init_DMA

- Habilite o clock (clock gate for DMA e clock gate for DMA for Multiplexor)
- Inicialize o Canal 0
  - Transfer mode: cycle steal
  - Data Source:
    - Address: &ADC0_RA
    - Transfer Size: 16 bits
  - Data Destination:
    - External Object Declaration: #include “meu.h”
    - Address: &var
    - Transfer Size: 16 bits
  - Byte Count: 2
  - Pin/Signal – DMA Mux settings:
    - Channel State: Enable
    - Channel Request: ADC0_DMA_Request

Configuração do DMA

- Interrupt – DMA transfer done interrupt
  - Interrupt Request: Enable
  - ISR Name: Dê um nome da interrupção Ex. DMA_done
  - DMA transfer interrupt: Enable
- Initialization – External Request : Enable

Configuração do DMA

- Crie um header chamado meu.h
  - Coloque a definição: extern uint16_t var
- Declare no main a variável var: uint16 var
- No Events.c, crie a função da interrupção:

```
PE_ISR(DMA_done)
{
DMA_DSR0 |= DMA_DSR_BCR_DONE_MASK;      // Clear Done Flag
DMA_DSR_BCR0 |= DMA_DSR_BCR_BCR(2);     // Set byte count register
}
```

- Uma vez que o DMA funcione, tente configurar o buffer circular do DMA