

## Laboratório 2: Programação Bare Metal com Interrupções e Periféricos

### Contexto:

No Lab 2 deve-se utilizar interrupções e periféricos integrados de um microcontrolador para implementar atividades de temporização com uso mínimo de CPU.

### Objetivo:

Escrever um programa em linguagem C (ou C + Assembly) para o kit EK-TM4C1294XL que realize a função de **decodificador PWM** para um sinal digital externo ao kit (a ser fornecido por um gerador de funções), assim como no Lab 1. Entretanto, as especificações para essa nova implementação são:

- 1) A frequência de operação da CPU ficará a critério de cada equipe.
- 2) A medição da largura dos pulsos do sinal externo deverá ser realizada totalmente por hardware (ou seja, por meio do uso de temporizadores internos ao microcontrolador) e com o uso de recursos de interrupção;
- 3) A medição da largura dos pulsos deverá ocorrer tanto para nível alto quanto para nível baixo do sinal externo, permitindo o cálculo do seu ciclo de trabalho e frequência;
- 4) O sistema deverá prever um controle de limite de tempo (*timeout*) de forma a identificar se o sinal externo possui período muito longo e informar essa situação ao usuário (na forma de ciclo de trabalho igual a 0% ou 100% e frequência nula).
- 5) As características medidas (ciclo de trabalho e frequência) do sinal externo deverão ser informadas ao usuário por meio da interface serial (UART disponível na porta USB do *debugger* disponível no kit) e um software emulador de terminal (p.ex.: Tera Term) rodando no PC hospedeiro (opcionalmente, pode-se utilizar também o display TFT LCD do Educational BoosterPack MKII ou os displays de sete segmentos do “BoosterPack do Prof. Peron”);
- 6) A programação dos periféricos do kit de desenvolvimento (SCB, PLL, GPIO, GPTM, UART, display TFT LCD) pode ser realizada utilizando-se a biblioteca de software TivaWare (driverlib, grlib, utils, etc.) ou soluções próprias de cada equipe.

Depois de implementado, o decodificador PWM resultante deve ser avaliado quanto ao seu desempenho e comparado à solução anterior (Lab 1), baseada puramente em software. Para tanto, deve-se idealizar um método para determinar os seguintes parâmetros:

- Estatísticas das medidas (média, desvio padrão e erro médio percentual) de sinais de entrada com ciclos de trabalho e frequências conhecidas em 10 amostras (sugestão: ondas retangulares de frequências 10kHz e 10MHz e ciclos de trabalho de 1%, 25%, 50%, 75% e 99%)
- Resolução do tempo nas medidas da solução implementada.

Qual é o impacto que a frequência de operação da CPU tem sobre os parâmetros de desempenho do decodificador PWM implementado dessa forma (temporização por hardware com interrupções)?

### Metodologia:

- Planejamento do sistema para medição de largura de pulsos por temporização de hardware (GPTM). Responder, de maneira formal e estruturada: Quais são os capítulos, seções, figuras e tabelas relevantes na documentação de hardware e software disponível? Qual é o modo de operação adequado do hardware (GPTM)? Quais são os registradores e funções de bibliotecas relevantes no uso desse modo de operação? Quais pinos do microcontrolador podem ser utilizados como entrada para o sinal digital externo?

- Implementação do sistema planejado;
- Teste e depuração do decodificador PWM, incluindo análises estatísticas dos resultados;
- Apresentação do funcionamento e dos resultados obtidos ao professor.

#### **Cronograma de desenvolvimento (*entregas*):**

26/09/2019 (S11) e 27/09/2019 (S12) – Estudo da plataforma e planejamento da solução.

03/10/2019 (S11) e 04/10/2019 (S12) – Código-fonte parcial desenvolvido até o momento.

10/10/2019 (S11) e 11/10/2019 (S12) – Semana de planejamento.

17/10/2019 (S11) e 18/10/2019 (S12) – Estatísticas de desempenho (planilha).

24/10/2019 (S11) e 25/10/2019 (S12) – Apresentação e demonstração do funcionamento.

#### **Critérios para as entregas:**

Os arquivos das entregas deverão ter os seus nomes codificados da seguinte forma: Sxx\_Gyy\_Lab2.\*, onde Sxx codifica a turma (S11 ou S12) e Gyy codifica a equipe (G01, G02, etc). Entregas deverão ser feitas sempre por e-mail para o endereço [hvieir@gmail.com](mailto:hvieir@gmail.com).

1. A entrega do estudo da plataforma e planejamento da solução deverá ser em formato **pdf** *sem a identificação dos integrantes da equipe* (apenas o nome do arquivo Sxx\_Gyy\_Lab2.pdf vinculará o trabalho à equipe).

O documento a ser elaborado deverá levar em conta, no mínimo, as seguintes diretrizes:

- Explicitar qual é a documentação de referência para o estudo da plataforma (ex: manual do microcontrolador, manual da biblioteca de funções), incluindo quais são os capítulos, seções, figuras e tabelas relevantes (se houver).
- Explicitar qual será o temporizador de hardware (GPTM) que será utilizado dentre os disponíveis no microcontrolador. Explicitar também qual será o pino do microcontrolador que será utilizado para a entrada do sinal digital externo.

2. A entrega do código-fonte da implementação parcial deverá preferencialmente ser em um repositório no GitHub contendo a pasta *completa* do projeto, cujo link deverá ser informado por e-mail. Os nomes dos integrantes devem constar em comentários logo no início dos arquivos de código-fonte de autoria da equipe (separar código principal e *device drivers* dos periféricos em arquivos diferentes). Alternativamente, a entrega do código-fonte poderá ser feita em formato **zip** ou **rar** (Sxx\_Gyy\_Lab2.zip ou Sxx\_Gyy\_Lab2.rar) – nesse caso, o arquivo compactado também deve conter a pasta *completa* do projeto.

3. A entrega das estatísticas de desempenho (planilha) deverá ser em formato **xlsx** ou **ods** *com a identificação dos integrantes da equipe* (nomes dos integrantes devem constar do arquivo Sxx\_Gyy\_Lab2.xlsx ou Sxx\_Gyy\_Lab2.ods). **Importante:** a planilha contendo as estatísticas de desempenho deverá seguir o modelo disponível para download no website da disciplina.

4. A entrega do código-fonte da implementação final deverá seguir os mesmos moldes da entrega do código-fonte da implementação parcial previamente entregue. **Importante:** a estrutura do projeto a ser entregue deve seguir as instruções para configuração de novos projetos no ambiente IAR EWARM fornecidas anteriormente – o projeto a ser entregue deve ser elaborado a partir do zero, inclusive com nome diferente dos fornecidos pelo professor.

#### **Bônus:**

Os integrantes da equipe cuja implementação apresentar melhor desempenho (maior exatidão e estabilidade nas medidas, além da maior frequência de medida possível) receberão um bônus de 0,5 ponto na nota final da disciplina (sujeito às condições estabelecidas anteriormente).

Sugestão (parcial) de arquitetura para a solução:

