

Laboratório 1: Programação Bare Metal Monotarefa (sem uso de interrupções)

Objetivo:

Implementar uma aplicação que utilize temporização por software (repetição em laços) e investigar a influência de fatores como frequência de operação da CPU e nível de otimização do compilador no seu desempenho em termos de exatidão e precisão.

Tarefa:

Escrever um programa em linguagem C (ou C + Assembly) para o kit EK-TM4C1294XL que gere um sinal PWM a ser aplicado no LED D4, de forma a controlar a intensidade da luz percebida pelo usuário. A configuração dos periféricos do sistema (SCB, PLL, GPIO, UART, display LCD) poderá ser realizada utilizando-se a biblioteca TivaWare (driverlib, grlib, utils, etc.) ou soluções próprias. As características temporais do sinal PWM (período e ciclo de trabalho) deverão ser monitoradas com o auxílio de um osciloscópio digital.

Especificações de funcionalidade:

- 1) O ciclo de trabalho deverá ser controlado pelo usuário por meio dos botões SW1 e SW2, que servirão para diminuir ou aumentar, respectivamente, a intensidade de luz do LED D4. Mudanças no ciclo de trabalho deverão ocorrer em passos de 10% cada vez que um dos botões seja pressionado e solto.
- 2) O valor do ciclo de trabalho deverá ser reportado ao usuário por meio de uma das seguintes alternativas a escolher: a) UART (*debugger*) para visualização em um emulador de terminal no PC hospedeiro; b) displays de sete segmentos ou display LCD usando o BoosterPack do Prof. Peron; c) display LCD usando o Boosterpack Educacional MKII.

Especificações de desempenho:

- 1) A frequência de operação da CPU deverá ser fixada inicialmente em 24MHz.
- 2) O nível de otimização do compilador deverá ser fixado inicialmente em baixo (*low*).
- 3) A intensidade de luz do LED D4 deverá ser percebida como contínua (sem piscar).
- 4) As características temporais do sinal PWM (período e ciclo de trabalho), monitoradas com o auxílio do osciloscópio, deverão sofrer o mínimo possível de interferência das funcionalidades da interface com o usuário (botões, UART, displays), sendo que a precisão do ciclo de trabalho deve ter maior prioridade do que a precisão do período do sinal. **Perguntas direcionadoras:**
a) Qual é o menor período que o código implementado permite para o sinal gerado? b) Qual é o maior período do sinal gerado que ainda permite a percepção de intensidade de luz contínua? c) Em qual dos dois casos há menor interferência da interface com o usuário nas características temporais do sinal gerado?

Testes a serem realizados e demonstrados:

- 1) Medir com o osciloscópio o período e o ciclo de trabalho do sinal gerado para todos os 11 passos de controle possíveis (0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% 90% e 100%). Há algum erro sistemático no período ou no ciclo de trabalho do sinal gerado?
- 2) Repetir as medições anteriores mantendo um dos botões pressionados. Ocorrem variações no período ou no ciclo de trabalho do sinal gerado? Há diferenças nos resultados das medições quando botões diferentes (SW1 ou SW2) são mantidos pressionados?
- 3) Repetir as medições anteriores para frequências de operação da CPU diferentes da que foi inicialmente determinada. Depois repetir as medições anteriores para níveis de otimização do compilador diferentes do que foi inicialmente determinado. Qual é o impacto destes fatores sobre o desempenho do gerador de sinal PWM implementado dessa forma (temporização por software *bare metal* sem interrupções)?

Metodologia:

- Estudo/revisão dos conceitos sobre modulação em largura de pulso (PWM);
- Elaboração de um diagrama de atividades para planejamento do algoritmo de controle e geração do sinal PWM (a ser entregue ao professor);
- Implementação do algoritmo planejado;
- **Teste e depuração** do gerador de sinal PWM, incluindo **análise e otimização** dos resultados (a serem entregues ao professor);
- Apresentação do funcionamento do sistema e dos resultados obtidos ao professor.

Cronograma de desenvolvimento (**entregas**):

12/03/2020 (S11) e 13/03/2020 (S12) – Repositório no GitHub contendo a estrutura do projeto.

19/03/2020 (S11) e 20/03/2020 (S12) – Diagrama de atividades da solução planejada.

26/03/2020 (S11) e 27/03/2020 (S12) – Código-fonte parcial desenvolvido até o momento.

02/04/2020 (S11) e 03/04/2020 (S12) – Apresentação e demonstração do funcionamento.

Critérios para as entregas:

1. A entrega do diagrama de atividades da solução planejada deverá ser em formato **pdf sem a identificação dos integrantes da equipe** (o nome do arquivo Sxx_Gyy_Lab1.pdf vinculará o trabalho à equipe, em que Sxx codifica a turma e Gyy codifica a equipe) – enviar por e-mail para hvieir@gmail.com.

O diagrama a ser elaborado deverá levar em conta as seguintes diretrizes:

- O detalhamento do algoritmo deverá ser equilibrado, não sendo genérico demais a ponto de não fornecer nenhum detalhe de implementação, nem específico demais a ponto de praticamente corresponder ao código-fonte a ser implementado.
- O detalhamento deverá ser maior em aspectos mais relevantes da implementação, aqueles que serão cruciais para o funcionamento desejado, incluindo aspectos de desempenho.
- Deverá ser utilizada a notação UML 2.x.

Sugestão de material de referência a respeito de diagramas de atividades:

- <https://www.lucidchart.com/pages/uml-activity-diagram>

Sugestão de ferramentas na nuvem para elaboração dos diagramas:

- <https://www.draw.io/>
- <http://www.umlet.com/umletino/umletino.html>

2. As entregas do código-fonte da implementação parcial e final deverão ser realizadas 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. **Importante:** a estrutura do projeto deve seguir as instruções para configuração de novos projetos no ambiente IAR EWARM – a estrutura do projeto deve ser elaborada a partir do *zero*, inclusive com nomes diferentes dos fornecidos nos exemplos de código.

Bônus:

O estudante/equipe cuja implementação apresentar as maiores precisões no ciclo de trabalho e no período do sinal PWM gerado, *devidamente documentadas em uma planilha de dados estatísticos das medições realizadas*, receberá um bônus de 0,5 pontos na nota final da disciplina (sujeito às condições estabelecidas anteriormente).