Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Eletrônica (DAELN)

SISTEMAS EMBARCADOS

Laboratório 4

Prof. André Schneider de Oliveira
andreoliveira@utfpr.edu.br

Objetivo

Escrever um programa em linguagem C++ Cortex-M para a geração de sinais através do módulo PWM, com 4 tipos de onda: **senoidal, triangular, dente-de-serra e quadrada.**

A configuração dos sinais será realizada através do teclado do computador, via UART, e deve permitir modificar: **a frequência (0-200Hz) e amplitude (0-3.3V).**

Será necessário a implementação de ***devices driver*** para a comunicação serial (UART) e o módulo PWM, que devem funcionar sem a necessidade de nenhum arquivo externo **(.c ou .h).**

Especificações UART

- Para o gerenciamento da UART deve ser desenvolvido um ***device driver*** (UART.c e UART.h) que contenha **TODOS** os recursos para a interface
  - Função de inicialização – initUART
  - Função de escrita - writeUART
  - Handler para o tratamento da interrupção
  - Todas as definições (#define) de endereçamento

*** Todas as linhas de código do *device driver* devem estar comentadas, descrevendo sua funcionalidade em detalhes**

*** Não é permitido o uso de nenhuma biblioteca pronta !!!**

Especificações PWM

- Para o gerenciamento do PWM deve ser desenvolvido um ***device driver*** (PWM.c e PWM.h) que contenha **TODOS** os recursos para a interface
  - Função de inicialização – initPWM
  - Função de envio - sendPWM
  - Todas as definições (define) de endereçamento

*** Todas as linhas de código do *device driver* devem estar comentadas, descrevendo sua funcionalidade em detalhes**

*** Não é permitido o uso de nenhuma biblioteca pronta !!!**

Especificações RTOS

- O recebimento da UART será realizada por meio da interrupção de hardware, que através de um **método de comunicação entre threads**, deve interfacear com a thread de tratamento da UART *(com prioridade maior que as demais threads)*

- A escrita no PWM deverá ser feita através de comunicação entre threads, ou seja, deve interfacear com a thread de tratamento da UART

  * Rever as funções RTOS que podem ser chamadas da ISR

- As seções críticas devem ser especificadas no código por um comentário antes e outro depois **(obrigatório)**

```
// INICIA SECAO CRITICA
.....
// FIM DA SECAO CRITICA
```

Requisitos

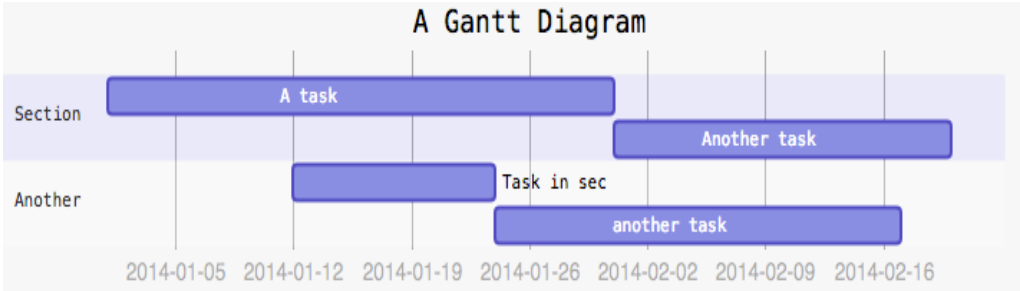
- Deve ser entregue o diagrama de estados e transições da solução proposta, representando a comunicação entre threads

- O desenvolvimento deve seguir ***integralmente*** o DET e sua comprovação será realizada com o Diagrama de Gantt **(a não execução da solução conforme o DET será penalizada)**

- O Diagrama de Gantt deve ser escrito pela placa, através da UART, conforme o exemplo (utilizar o site https://mermaidjs.github.io para gerar gráfico)

```
gantt
    title A Gantt Diagram
    dateFormat  YYYY-MM-DD
    section Section
    A task           :a1, 2014-01-01, 30d
    Another task     :after a1  , 20d
    section Another
    Task in sec      :2014-01-12  , 12d
    another task      : 24d
```

*



Cronograma de Avaliação – Lab 4

- 15/10
  - Entrega do Diagrama de Estados e Transições (DET) da solução proposta
  - A não execução acarretará na **anulação** do laboratório

- 22/10
  - Apresentação do **envio e recebimento** (por interrupção) da UART
  - Apresentação da geração de **algum sinal** através do PWM
  - A não execução acarretará na **perda de 1 ponto** na nota do laboratório

- 29/10
  - Defesa, demonstração e teste