

**Instituto Tecnológico de Aeronáutica - ITA**

# Laboratório

# 1

Sprint 01

**LUCAS NICOLLI TOSI**

**Agosto de 2019**





## 1 Objetivos

### 1.1 Objetivos gerais

- Se familiarizar com o ambiente de desenvolvimento (KDS, KSDK, *Processor Expert*);
- Exercitar o processo de geração da imagem executável; e
- Exercitar o processo de *debugging* no *target*.

### 1.2 Objetivos específicos

- Construir (*build*) a biblioteca de plataforma no KDS;
- Criar um projeto baseado na placa **FRDM-KL25Z** no KDS;
- Utilizar o *Processor Expert* para criar/configurar uma OSA (RTOS) baseado em **FreeRTOS**; e
- Dominar o processo básico de *debugging* no *target* utilizando o **GDB PEMicro Interface De-bugging**.

## 2 Desenvolvimento

### 2.1 Construir a biblioteca de plataforma no KDS

1. Obter o guia "*KSDK12GSUG - Getting Started with Kinetis SDK (KSDK) v.1.2*" no site da Freescale;
2. Seguir os passos descritos em "*5.3 Build the platform library*" para construir a biblioteca para "**KL25Z4**"; e
3. Construir para as versões "*Debug*" e para "*Release*".

### 2.2 Criar um projeto baseado na placa FRDM-KL25Z no KDS

No KDS:

1. File -> New -> Processor Expert Project
2. Definir um nome para o projeto

Na página "**Devices**":

1. Selecionar a placa "**FRDM-KL25Z**"

Na página "**Rapid Application Development**":

1. Selecionar a localização da KSDK (preferir utilização da variável de ambiente KSDK\_PATH)
2. Marcar a utilização do **Processor Expert**

Na página "*Processor Expert Target Compiler*":

1. Marcar o "*GNU C Compiler*"
2. *Finish*

"*Buildar*" o projeto "*as is*" e anotar as estatísticas:

```
1 'Invoking: Cross ARM GNU Print Size'
2 arm-none-eabi-size --format=berkeley "kdsug\_tutorial.elf"
3 text      data      bss      dec      hex      filename
4 13796     120       3088     17004     426c     kdsug\_tutorial.elf
```

## 2.3 Utilizar o *Processor Expert* para criar/configurar um RTOS baseado em FreeRTOS

Na visão "*Processor Expert*", aba "*Components Library*" adicionar um componente de RTOS (fsl\_os\_abstraction).

Já na aba "*Components*" do KDS, abrir o componente do RTOS criado (dois cliques) e alterar o parâmetro "*OS*" na aba "*Properties*" para "*FreeRTOS*".

Na visão "*Processor Expert*", gerar o código ("*Generate Code*").

"*Buildar*" o projeto e anotar as estatísticas:

```
1 'Invoking: Cross ARM GNU Print Size'
2 arm-none-eabi-size --format=berkeley "kdsug\_tutorial.elf"
3 text      data      bss      dec      hex      filename
```

Na visão "*Processor Expert*", aba "*Components Library*" adicionar um componente de "task OS" (OS\_Task).

Na visão "*Processor Expert*", gerar o código ("*Generate Code*").

"*Buildar*" o projeto e anotar as estatísticas:

```
1 'Invoking: Cross ARM GNU Print Size'
2 arm-none-eabi-size --format=berkeley "kdsug\_tutorial.elf"
3 text      data      bss      dec      hex      filename
```

Comparar as estatísticas e verificar o *footprint* do RTOS/tarefas.

## 2.4 Dominar o processo básico de *debugging* no *target* utilizando o *GDB PEMicro Interface Debugging*

Conectar o kit no *host* utilizando cabo USB na porta "*OpenSDA*" do *target*.

No ícone de "*debug*" do Eclipse, abrir "*Debug Configurations*".

Em "*GDB PEMicro Interface Debugging*", criar uma configuração de *debug*.

Confirmar o seguinte na aba "*Debugger*"

1. "*PEMicro Interface Settings*"
2. Interface: OpenSDA Embedded Debug - USB Port
3. Port: USBx - OpenSDA (xxxxxxx)

Clicar no botão "*Debug*".

Explorar o código fonte do projeto gerado:

- Criação das tarefas

- Inicialização do sistema
- Adicionar *breakpoints* no código
- Explorar as opções de fluxo de *debug* (entrar nas funções, passar direto, etc)

## 3 Avaliação

### 3.1 Avaliação do resultado do laboratório

Essa atividade de laboratório não necessita de entrega de relatório.

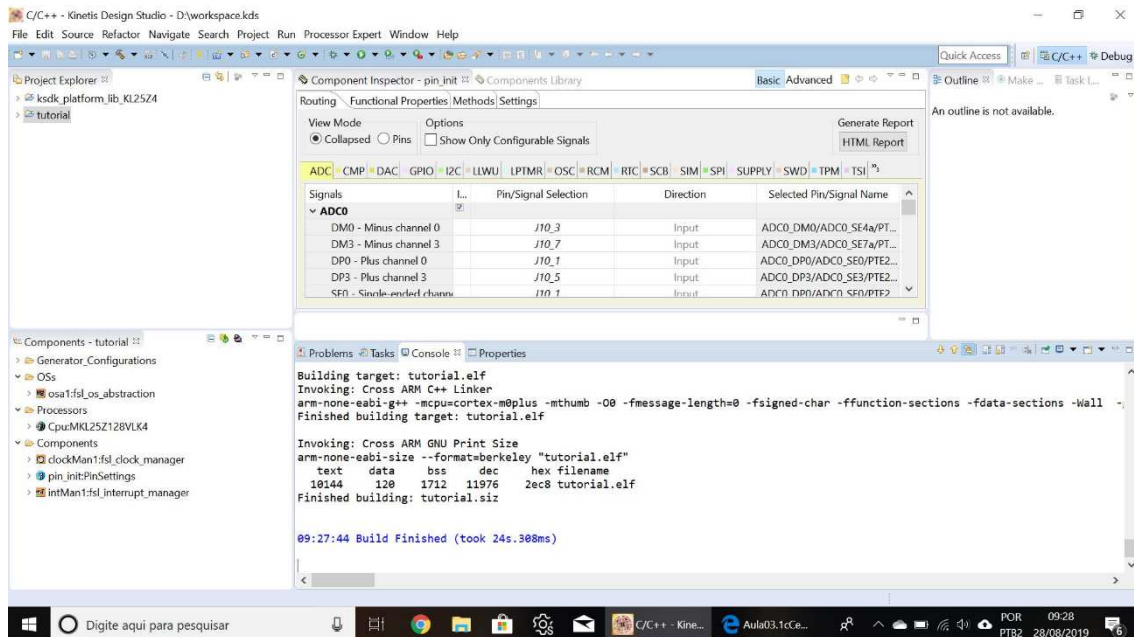
Ao término da atividade, o aluno deverá apresentar os resultados ao professor no período da aula.

## 4 Referências

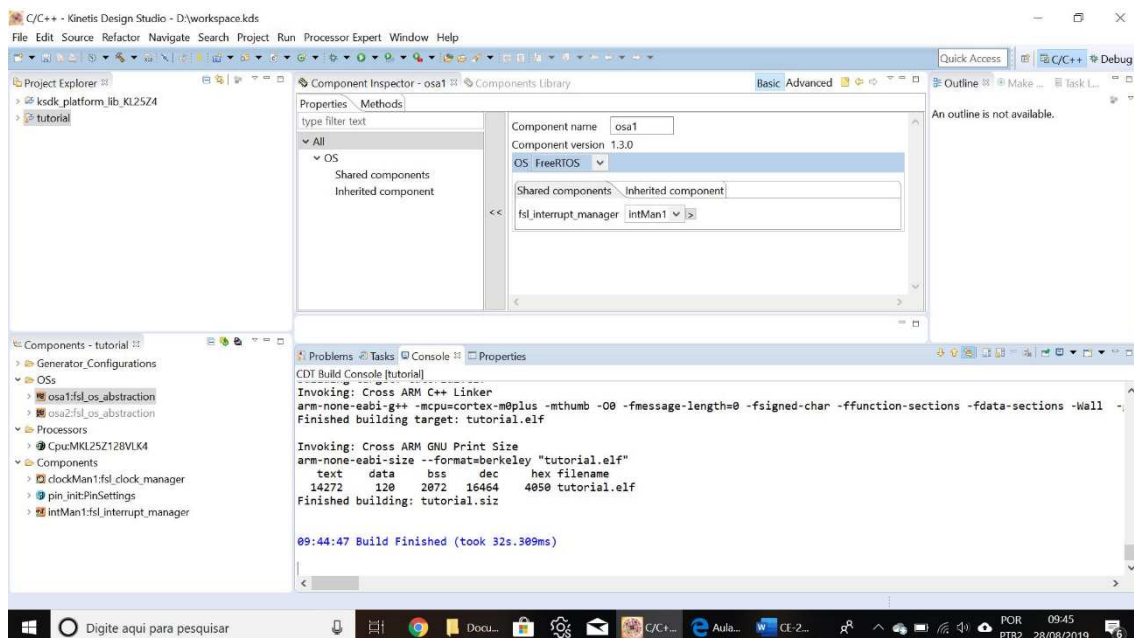
Notas de aula baseadas nos seguintes textos:

- [1] Real Time Engineers Ltd., “Freertos - <http://www.freertos.org>.”
- [2] Freescale Semiconductor, *KSDK12GSUG - Getting Started with Kinetis SDK (KSDK) v.1.2*, 0 ed., 2015.

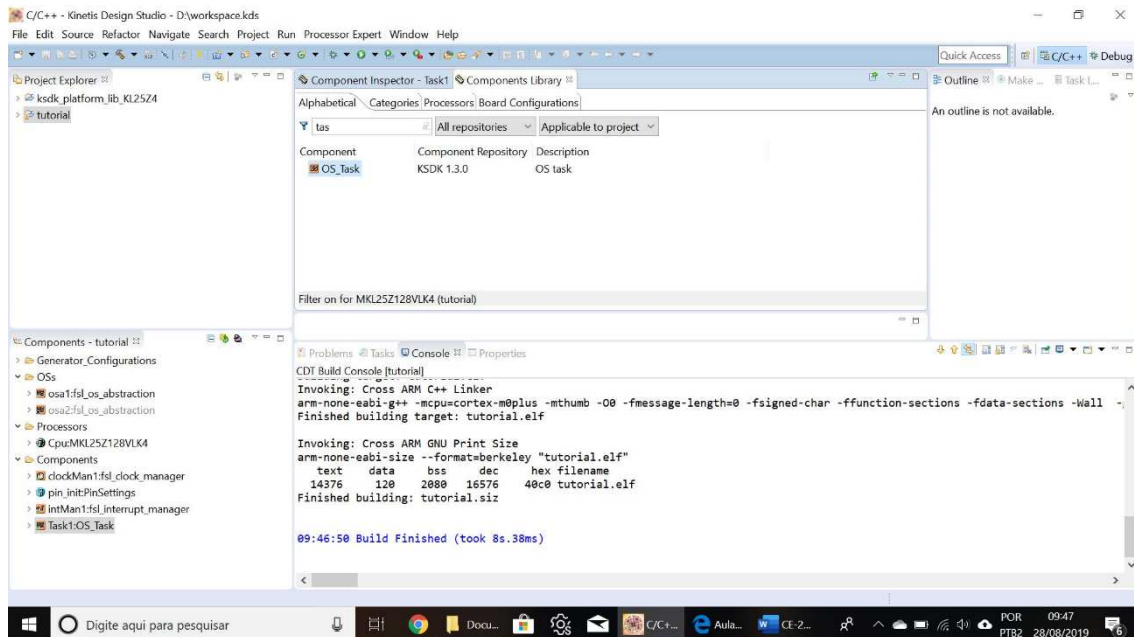
## 1. Build do Projeto "As is". Espaço em disco utilizado **11976 Bytes**:



## 2. Build do Projeto com o FreeRTOS. Espaço em disco utilizado **16464 Bytes**:



3. Build do Projeto com a Task. Espaço em disco utilizado **16576** Bytes:



4. Upload do LAB1 no GitHub: <https://github.com/Intosi/CE-235>