

# IM420 Sistemas Embarcados de Tempo Real

## Notas de Aula – Semana 03

**Prof. Denis Loubach**

dloubach@fem.unicamp.br

Programa de Pós-Graduação em Eng. Mecânica / Área de Mecatrônica  
Faculdade de Engenharia Mecânica - FEM  
Universidade Estadual de Campinas - UNICAMP



1º Semestre de 2018

# Tópicos

- 1 Motivação
- 2 uC Target
- 3 Ambiente de desenvolvimento
- 4 RTOS
- 5 Referências

# *Cooki: a Desktop Robotic Chef That Does Everything*



Photo: Evan Ackerman/IEEE Spectrum

You'll never ever have to cook ever again, ever.

**Figura:** Lançamento na CES 2016, fonte: <http://spectrum.ieee.org/automation/robotics/home-robots/cooki-a-desktop-robotic-chef-that-does-everything>

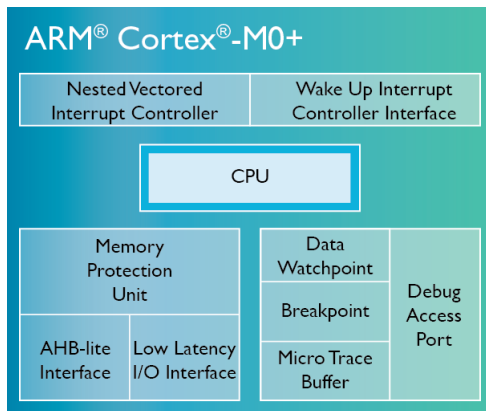
# Visão geral uP

Esta disciplina utilizará um uP da família ARM (*Advanced RISC Machine*), linha Cortex-M: *ARM Cortex-M0+ Core 32-bit*

- Projetado para ser o mais eficiente em consumo de energia da família ARM
- *Core pipeline* de dois estágios
- Consumo de energia: 9.8uW/MHz
- *Debug: Optional JTAG ou Serial-Wire Debug (SWD) ports* com até 4 *Breakpoints* e 2 *Watchpoints*
- 56 instruções
- Arquitetura ARMv6-M
- Arquitetura de memória *Von Neumann* (dados e instruções compartilham um barramento)

Fonte [1]

# Diagrama de blocos do uP



**Figura:** Cortex-M0+ Processor, fonte:

<http://www.arm.com/products/processors/cortex-m/cortex-m0plus.php>

# Kit de desenvolvimento

Esta disciplina utilizará o kit de desenvolvimento **FRDM-KL25Z** que contém um processador ARM Cortex-M0+

Principais características do uC **Freescall KL25Z Kinetis KL2x (MKL25Z128VLK4)**:

- 48MHz, 16KB RAM, 128KB FLASH
- USB (Host/Device)
- SPI (x2)
- I2C (x2)
- UART (x3)
- PWM (Timer/PWM Module - TPM)
- ADC (16 bit)
- DAC (x1 12bit)
- Touch Sensor
- GPIO (x66)

# FRDM-KL25Z

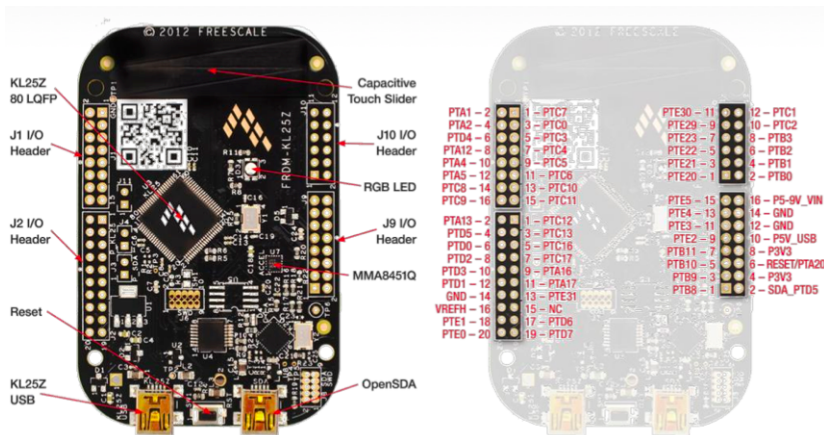


Figura: FRDM-KL25Z fonte: [2]

# Diagrama de blocos do uC

Kinetis KL2x MCU Family Block Diagram

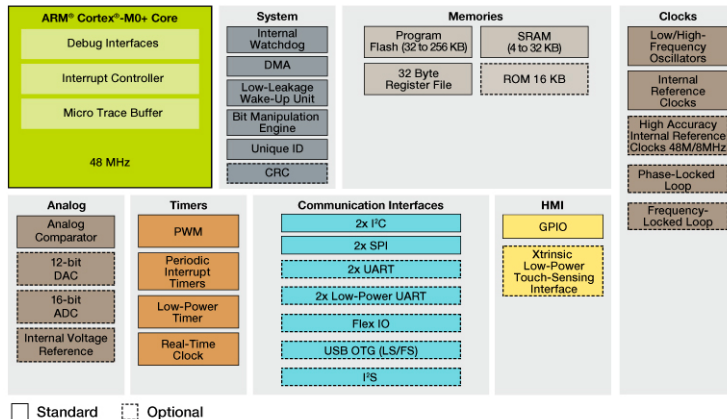


Figura: Família KL2x, fonte:

<http://www.nxp.com/products/microcontrollers-and-processors/arm-processors/kinetis-cortex-m/1-series/kinetis-kl2x-48-mhz-usb-connectivity-ultra-low-power-mcus:KL2x>



# Diagrama de blocos do kit

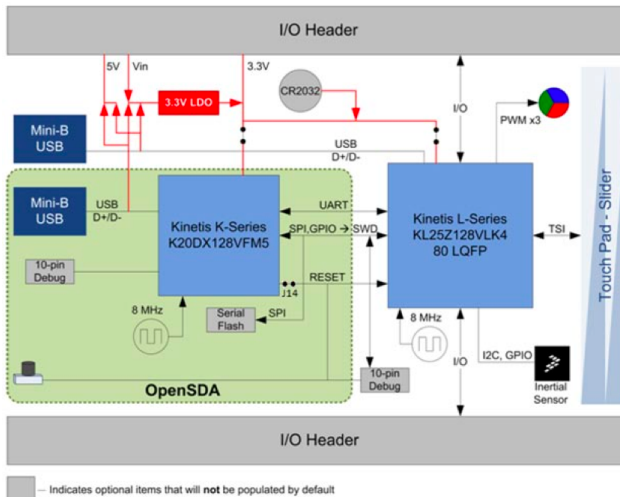


Figura: Digrama de blocos do kit, fonte: [3]

# Detalhes do uC *target*, fonte [3]

## 32-bit ARM Cortex-M0+ core

- até 48 MHz de operação
- porta de acesso de I/O rápida (*single-cycle*)

## Memórias

- 128 kB FLASH
- 16 kB SRAM

## Sistema

- controlador DMA
- *Computer operating properly (COP) Watchdog timer*

## Detalhes do uC *target*, fonte [3] (cont...)

### ***Clock***

- módulo de geração de *clock* com suporte para FLL e PLL para o sistema e para CPU
- *clocks* internos de referência de 4 MHz e 32 kHz
- Oscilador do sistema com suporte para cristal externo e resonador
- Oscilador de baixo-consumo de 1 kHz para RTC e COP *watchdog*
- O uC montado no kit está conectado a um cristal externo de 8 MHz

O uC possui um *on-chip oscillator* compatível com três ranges de frequência de entrada de cristal ou resonador:

- 1 32-40 kHz (modo baixa frequência)
- 2 3-8 MHz (modo alta frequência, baixo range)
- 3 8-32 MHz (modo alta frequência, alto range)

# Detalhes do uC *target*, fonte [3] (cont...)

## Periféricos analógicos

- 16-bit *successive approximation* (SAR) ADC com suporte DMA
- 12-bit DAC com suporte DMA
- Comparadores de alta velocidade

## Periféricos de comunicação

- 2x 8-bit *serial peripheral interfaces* (SPI)
- *USB dual-role controller*
- *USB voltage regulator*
- 2x *inter-integrated circuit* (I<sup>2</sup>C)
- 1x *low-power UART module*
- 2x *standard UART modules*

# Detalhes do uC *target*, fonte [3] (cont...)

## Temporizadores

- *1x 6-channel Timer/PWM module*
- *2x 2-channel Timer/PWM modules*
- *2-channel Periodic Interrupt Timer (PIT)*
- *Real time clock (RTC)*
- *Low-power Timer (LPTMR)*
- *System tick timer*

## Interface homem-máquina

- *General purpose input/output controller*
- *Capacitive touch sense input interface hardware module*

## Serial and Debug Adapter (OpenSDA)

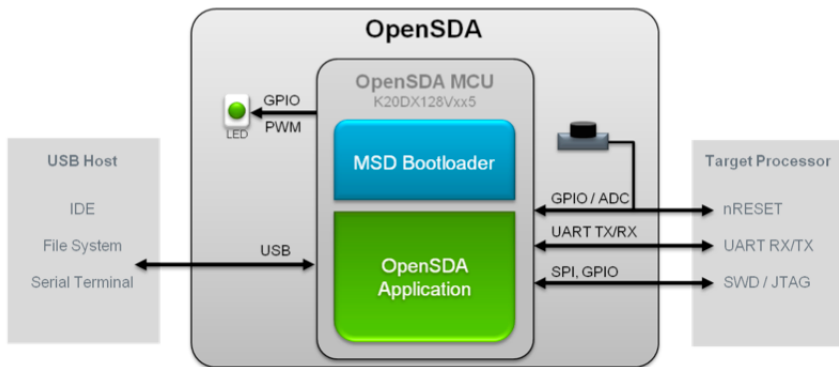
OpenSDA é um adaptador de *debug* e serial de padrão aberto

Tem o objetivo de interligar as comunicações de *debug* e serial de um USB *host* para o processador embarcado *target*

Possui um *mass storage device (MSD) bootloader* para carregar tipos diferentes de aplicações SDA:

- programador de flash
- interfaces de controle de *debug*
- conversor serial-USB

# Diagrama de blocos do OpenSDA



**Figure 5. OpenSDA High-Level Block Diagram**

Figura: Diagrama de blocos do OpenSDA, fonte: [3]

# OpenSDA no kit

Gerenciado por um uC Kinetis K20 contendo um core ARM Cortex-M4

O circuito do OpenSDA conta com um LED de *status* (D4) e um *pushbutton* (SW1)

O SW1 pode ser utilizado para colocar o OpenSDA em modo *bootloader*

Sinais SPI e GPIO fazem interface com a porta SWD do KL25Z

O circuito do OpenSDA é energizado via conector J7 conectado na USB do computador *host*

Uma porta serial para propósitos de *debug* é disponibilizada pelo OpenSDA (*USB Communications Device Class* - CDC)



# Principais documentos de referência

- FRDM-KL25Z Quick Start Package
- FRDM-KL25Z User's Manual
- FRDM-KL25Z Pinouts
- FRDM-KL25Z Schematics
- FRDM-KL25Z Design Package
- OpenSDA User's Guide
- Kinetis KL25 Sub-Family
- KL25 Sub-Family Reference Manual

Documentos podem ser encontrados em

[www.freescale.com/FRDM-KL25Z](http://www.freescale.com/FRDM-KL25Z)

# Visão geral do ambiente de desenvolvimento

Esta disciplina utilizará a ferramenta de desenvolvimento de software denominada *Kinetis Design Studio (KDS)* [4]

Ferramenta GNU e baseada em Eclipse

Suporta os dispositivos Kinetis Cortex-M

Integrado ao *Processor Expert*

Integrado ao *Kinetis Software Development Kit (KSDK)*

Para adaptadores de *debug* suporta:

- *P&E USB Multilink Universal*
- *Cortex Microcontroller Software Interface Standard - Debug Access Port (CMSIS-DAP)*
- outros...

Utiliza a biblioteca de *runtime newlib-nano C* (ajuda a reduzir o *footprint* de memória para software embarcado)

## Tool-chain utilizada

- *Cross ARM GCC (GNU Compiler Collection)*
- Utilizar como *builder* o *GNU Make Builder*
- Utilizar a extensão `eabi` (*embedded-application binary interface*)

# Application Programming Interface (API)

## Kinetis SDK API

Kinetis SDK Block Diagram

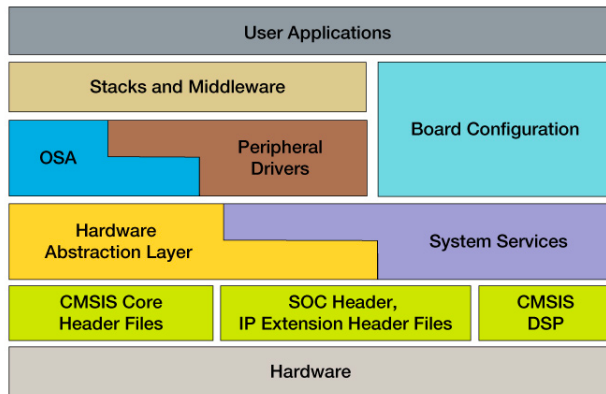


Figura: Digrama de blocos do KSDK, fonte: [5]

# Kinetis SDK

Suite contendo:

- interface de hardware
- camadas de abstração de hardware
- *drivers* de periféricos
- abstrações de RTOS
- pilhas
- *middleware*

Inclui código fonte completo (*open-source license*) para todas:

- abstrações de hardware
- *driver de periféricos*

Não é "caixa-preta"!

# Kinetis SDK (cont...)

## **CMSIS**

Arquivos de cabeçalho (*header files*) que proveem acesso direto aos registradores e bits dos periféricos

## ***Hardware abstraction layer (HAL)***

*Driver* simples e sem estados com uma API encapsulando as funções de baixo nível do periférico

## ***System Services***

Dedicado para recursos centralizados incluindo os gerenciadores de *clock*, interrupções, *low-power* e temporizadores implementados em hardware

# Kinetis SDK (cont...)

## ***High-level peripheral drivers***

Construídos com base no HAL, podem utilizar um ou mais *system services*.  
Utilizados para manipulação do periféricos, em "alto nível" de abstração

## ***Operating system abstraction (OSA)***

Camada para adaptar a aplicação para utilização de RTOS ou *bare-metal*

## ***Pilhas e middleware***

- USB *device, host, OTG*
- CMSIS-DSP
- Sistema de arquivos
- entre outros

## Kinetis SDK (cont...)

O HAL, *peripheral drivers*, e *system services* podem ser utilizados entre vários produtos Kinetis sem modificações

Os itens configuráveis para cada *driver* (em todos os níveis de abstração) encontram-se encapsulados em estruturas de dados escritas em linguagem C

Os *mainline releases* incluem suporte para a coleção de uC Kinetis

Os *standalone releases* incluem suporte para um ou poucos uC



# Principais documentos de referência

- KSDK12APIRM - Kinetis SDK v.1.2 API Reference Manual, Rev. 0, April 2015
- KSDK12GSUG - Getting Started with Kinetis SDK (KSDK) v.1.2, Rev. 0, April 2015
- KDSUG - Kinetis Design Studio V3.0.0- User's Guide, Rev. 1, April 2015

# Lista de versões das ferramentas utilizadas

Kinetis Design Studio Installation Details

Installed Software | Installation History | Features | Plug-ins | Configuration

Name	Version	Id	Provider
Component Development Environment	1.8.0.RT7_b1530-1106	com.freescale.processorexpert.cde.feature...	Freescale Semiconductor, Inc.
Component Development Environment Documentation	1.1.0.RT7_b1530-1106	com.freescale.doc.cde.feature.feature.gro...	Freescale Semiconductor, Inc.
Documentation for Kinetis Design Studio V3.0.0	4.0.0	com.freescale.kds.coredocs.feature.group	Freescale Semiconductor, Inc.
Embedded Systems Register View (SFR)	0.2.4.r178	org.eclipse.cdt.embsysregview.feature.fea...	
FSL Eclipse Updater Custom Actions Feature for Eclipse 4.4.x	2.0.12.FSL_150303	com.freescale.updater.customActions.feat...	Freescale Semiconductor, Inc.
GNU ARM C/C++ Cross Compiler	2.1.1.201512141335	ilg.gnuarmclipse.managedbuild.cross.fea...	Liviu Ionescu
GNU ARM C/C++ J-Link Debugging	3.2.1.201512141335	ilg.gnuarmclipse.debug.gdbjtag.jlink.feat...	Liviu Ionescu
GNU ARM C/C++ OpenOCD Debugging	3.2.1.201512141335	ilg.gnuarmclipse.debug.gdbjtag.openoc...	Liviu Ionescu
GNU ARM PEMicro Interface Debugging Support	2.2.9.201512221411	com.pemicro.debug.gdbjtag.pne.feature.f...	PEMicro Microcomputer Systems
> Kinetis Design Studio	3.1.0.201512141710	com.somniumtech.kds	
> KSDK 1.2.0 Eclipse Update	1.0.1.b150603	com.freescale.processorexpert.ksdk-1.2.0.s...	Freescale Semiconductor, Inc.
> Processor Expert for Kinetis	3.0.2.b151214	com.freescale.processorexpert.kinetis.feat...	Freescale Semiconductor, Inc.
Project of Projects	1.1.8.201504141312	com.freescale.projectofprojects.eclipse.fe...	Freescale Semiconductor, Inc.

Update... Uninstall... Properties Close

# Visão geral do RTOS FreeRTOS

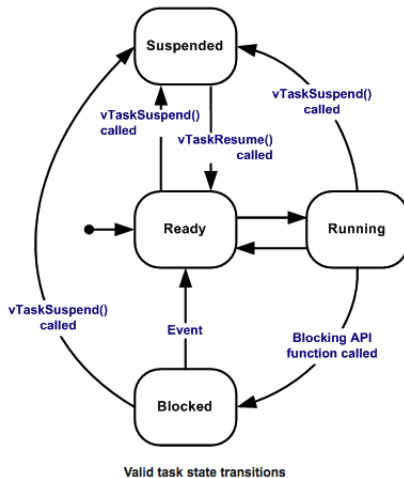
Esta disciplina utilizará o RTOS FreeRTOS para a parte de laboratório

# Visão geral do RTOS FreeRTOS

- Opção de escalonamentos preemptivo, colaborativo, *round robing* com *time slicing*
- 5~12 kB *footprint* de FLASH
- ~250 bytes de *footprint* de RAM
- *Mutexes* com herança de prioridade
- *Mutexes* recursivos
- Semáforos dos tipos contador e binário
- Fila de mensagens
- O código fonte do *core* conforma com o padrão de codificação MISRA
- Detecção de *stack overflow*
- Seleção de esquema de alocação de memória

Fonte: <http://www.freertos.org>

# Tarefas no FreeRTOS



**Figura:** Digrama de estados da tarefa no FreeRTOS, fonte:  
<http://www.freertos.org/RTOS-task-states.html>

# Descrição dos estados

- **Running** - tarefa encontra-se em execução no processador
- **Ready** - tarefa está pronta para ser executada
- **Blocked** - tarefa está bloqueada por conta de algum evento temporal ou externo (*e.g.*, fila, semáforo)
- **Suspended** - tarefa suspensa não disponível para o escalonamento, após ser explicitamente comandada para este estado por um `TaskSuspend` e só volta após um `TaskResume`

# Principais documentos de referência

- **Site oficial** <http://www.freertos.org/> [6]
- **Tutoriais** <http://mcuoneclipse.com/2012/09/29/tutorial-freedom-with-freertos-and-kinetis-l/>

# Informação ao leitor

Notas de aula baseadas nos seguintes textos:



ARM Ltd., “Cortex-m0+ processor  
<http://www.arm.com/products/processors/cortex-m/cortex-m0plus.php>.”



Freescale Semiconductor, *Quick Start Guide for FRDM-KL25Z*, 1 ed., 2012.



Freescale Semiconductor, *FRDMKL25ZUM - FRDM-KL25Z User's Manual*, 2 ed., 2013.



Freescale Semiconductor, *KDSUG - Kinetis Design Studio V3.0.0- User's Guide*, 1 ed., 2015.



Freescale Semiconductor, *KSDK12APIRM - Kinetis SDK v.1.2 API Reference Manual*, 0 ed., 2015.



Real Time Engineers Ltd., “Freertos - <http://www.freertos.org>.”