



EA871

Ambiente de Desenvolvimento Integrado (IDE)

CodeWarrior – Visão Geral

Wu Shin – Ting
DCA – FEEC - Unicamp
Segundo Semestre de 2020

Programação do Microcontrolador

- Dois paradigmas de programação:
 - ***Bare-metal ou board***
 - exposição direta aos registradores (dados, estado e controle) do microcontrolador.
 - exposição indireta dos registradores através de uma camada de funções mais amigáveis, como no Arduino.
 - Em cima de um sistema operacional em tempo real, como MQX-Lite RTOS.
- Aplicativo **IDE** (*Integrated Development Environment*): desenvolver o código num computador com mais recursos, gerar o código de máquina do microcontrolador e transferí-lo para o microcontrolador. Alguns ambientes dispõem de funcionalidades para verificação do fluxo de execução de um programa, passo a passo, no microcontrolador (**depurador**).

Conceitos relacionados a um IDE

- **Áreas de Trabalho** (*Workspaces*): pasta onde contém todos os dados dos projetos desenvolvidos.
- **Bancada de Trabalho** (*Workbench*): janela principal (bancada) que contém todas as subjanelas de trabalho (ferramentas) disponíveis para desenvolvimento de projetos.
- **Barra de menus** (*Menu bar*): barra que contém menus de funções das principais classes de ferramentas.

Conceitos relacionados a um IDE

- **Perspectivas** (*Perspectives*): forma de apresentação da bancada de trabalho.
- **Barra de Ferramentas** (*Toolbar*): barra que contém ícones de acesso aos comandos relacionados a uma perspectiva.
- **Vistas** (*Views*): componentes visuais da bancada de trabalho, organizados dentro de uma perspectiva.
- **Pilhas ou Abas de Vistas** (*View stacks ou tabbed view*): um conjunto de vistas agrupadas.



CodeWarrior IDE Development Suite

Informações Adicionais

- Quick Reference:

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea076/complementos/CWMCUQRCARD.pdf>

- Guia de CodeWarrior:

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/manuais/CWCFUG.pdf>

- Tabela de Teclas de Atalho para CodeWarrior:

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/complementos/eclipseCDT8.0-cheatsheet.pdf>

- Dicas para customizar o ambiente CodeWarrior:

<https://mcuoneclipse.com/2012/07/14/customize-my-workspace-perspectives/>

- Síntese das principais funções do IDE utilizadas nesta disciplina:

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/apostila_C/AmbienteDesenvolvimentoSoftware.pdf



EA871

Ambiente de Desenvolvimento Integrado (IDE)

CodeWarrior – Criar um Projeto

Wu Shin – Ting
DCA – FEEC - Unicamp
Segundo Semestre de 2020

Especificações

1. Nome do projeto
2. Modelo do microcontrolador
3. Tipo de projeto: aplicativo ou biblioteca
4. Conexão para programação do microcontrolador
5. Recursos a serem usados para gerar um aplicativo ou uma biblioteca (*Toolchain*)
6. Uso de funções de mais alto nível (*Processor Expert*)
7. Perspectiva a ser aberta para desenvolvimento

Especificações

1. Nome do projeto
2. Modelo do microcontrolador
3. Tipo de projeto: **aplicativo** ou **biblioteca**
4. Conexão para programação do microcontrolador
5. Recursos a serem usados para gerar um aplicativo ou uma biblioteca (*Toolchain*)
6. Uso de funções de mais alto nível (*Processor Expert*)
7. Perspectiva a ser aberta para desenvolvimento

Aplicativo e Biblioteca

- **Aplicativo:** um arquivo executável no microcontrolador.
- **Biblioteca:** coletânea de funções e rotinas em códigos de máquina (compiladas) que possam ser ligadas com outros códigos de máquina para gerar um aplicativo (um arquivo executável).
 - **Biblioteca estática (.a):** os códigos das rotinas das bibliotecas são adicionados ao aplicativo.
 - **Biblioteca dinâmica (.dll, .so, .dylib):** os códigos das rotinas das bibliotecas estão fora do aplicativo, acessados apenas no tempo de execução.

Aplicativo e Biblioteca

- **Aplicativo:** um arquivo executável no microcontrolador.
- **Biblioteca:** coletânea de funções e rotinas em códigos de máquina (compiladas) que possam ser ligadas com outros códigos de máquina para gerar um aplicativo (um arquivo executável).
 - **Biblioteca estática (.a):** os códigos das rotinas das bibliotecas são adicionados ao aplicativo.
 - **Biblioteca dinâmica (.dll, .so, .dylib):** os códigos das rotinas das bibliotecas estão fora do aplicativo, acessados apenas no tempo de execução.

Aplicativo e Biblioteca

- **Aplicativo:** um arquivo executável no microcontrolador.
- **Biblioteca:** coletânea de funções e rotinas em códigos de máquina (compiladas) que possam ser ligadas com outros códigos de máquina para gerar um aplicativo (um arquivo executável).
 - **Biblioteca estática (.a):** os códigos das rotinas das bibliotecas são adicionados ao aplicativo.
 - **Biblioteca dinâmica (.dll, .so, .dylib):** os códigos das rotinas das bibliotecas estão fora do aplicativo, acessados apenas no tempo de execução.

Especificações

1. Nome do projeto
2. Modelo do microcontrolador
3. Tipo de projeto: aplicativo ou biblioteca
4. Conexão para programação do microcontrolador
5. Recursos disponíveis para gerar aplicativo ou biblioteca para o microcontrolador (***Toolchain***)
6. Uso de funções de mais alto nível (*Processor Expert*)
7. Perspectiva a ser aberta para desenvolvimento

Toolchain

- É um conjunto de ferramentas necessárias para criar um aplicativo a ser executado num dispositivo específico.
- É usado nesta disciplina ***GNU Arm Embedded Toolchain*** para gerar código de máquina de um microcontrolador ARM de 32 *bits*:
 - Bibliotecas padrão de C
 - Pré-processador
 - Compilador **gcc**
 - Montador (*assembler*) **as**
 - Ligador **ld**
 - etc.



CodeWarrior IDE Development Suite

Informações Adicionais

- Bibliotecas estáticas e dinâmicas:

<https://medium.com/swlh/linux-basics-static-libraries-vs-dynamic-libraries-a7bcf8157779>

- GNU ARM *Embedded Toolchain*:

<https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm>



EA871

Ambiente de Desenvolvimento Integrado (IDE)

CodeWarrior – Depurar um Projeto

Wu Shin – Ting
DCA – FEEC - Unicamp
Segundo Semestre de 2020

OpenSDA

- É uma interface entre o computador onde roda o IDE CodeWarrior e o microcontrolador para o qual estamos desenvolvendo o programa. Ela serve para **programar a memória flash do microcontrolador**, depurar o programa em execução no microcontrolador e estabelecer a comunicação serial entre o computador e o microcontrolador.

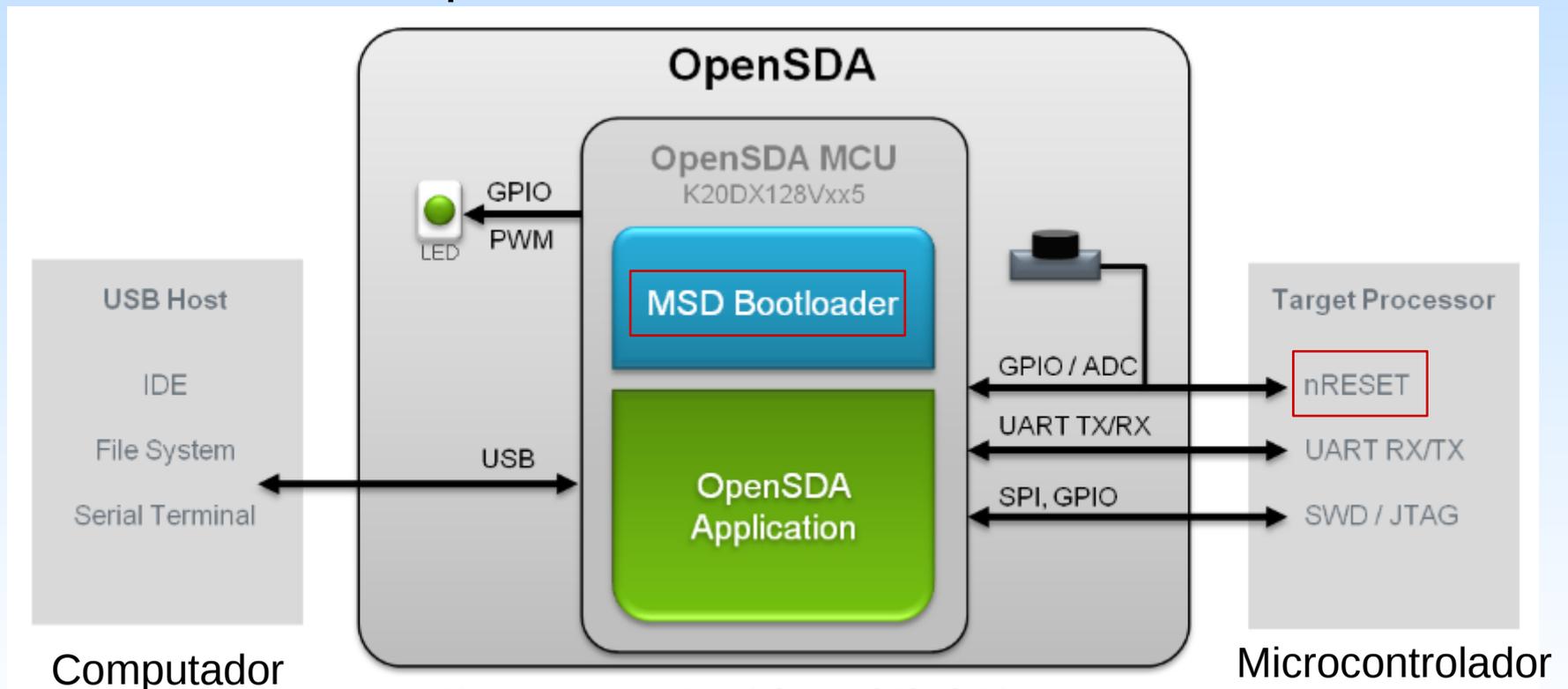


Figure 4. OpenSDA High-Level Block Diagram

Bootloader

- É a primeira sequência de instruções a ser executada quando ocorre o evento “*Reset*”.

0x0000_0000 to 0x0000_03FF

0x0000_0400 to 0x0000_040F

0x0000_0410 to 0x0000_FFFF

0x0000_0410 to 0x0000_FFFF (with printf)

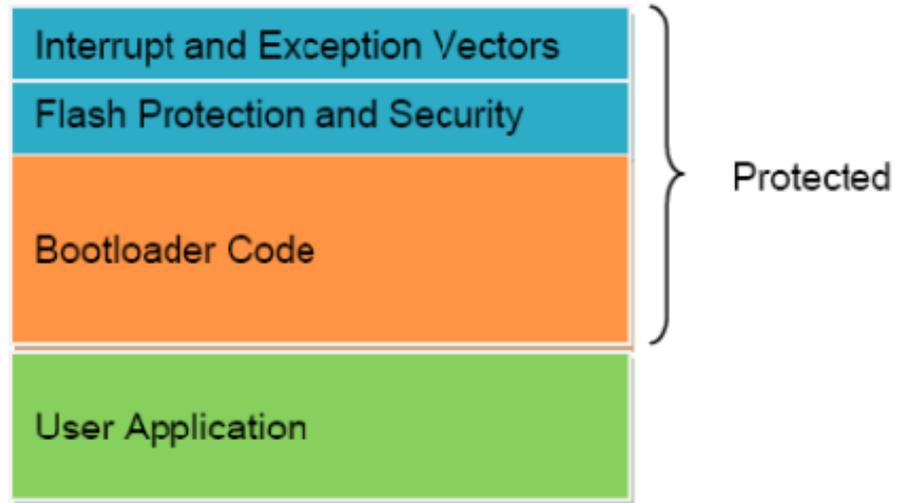


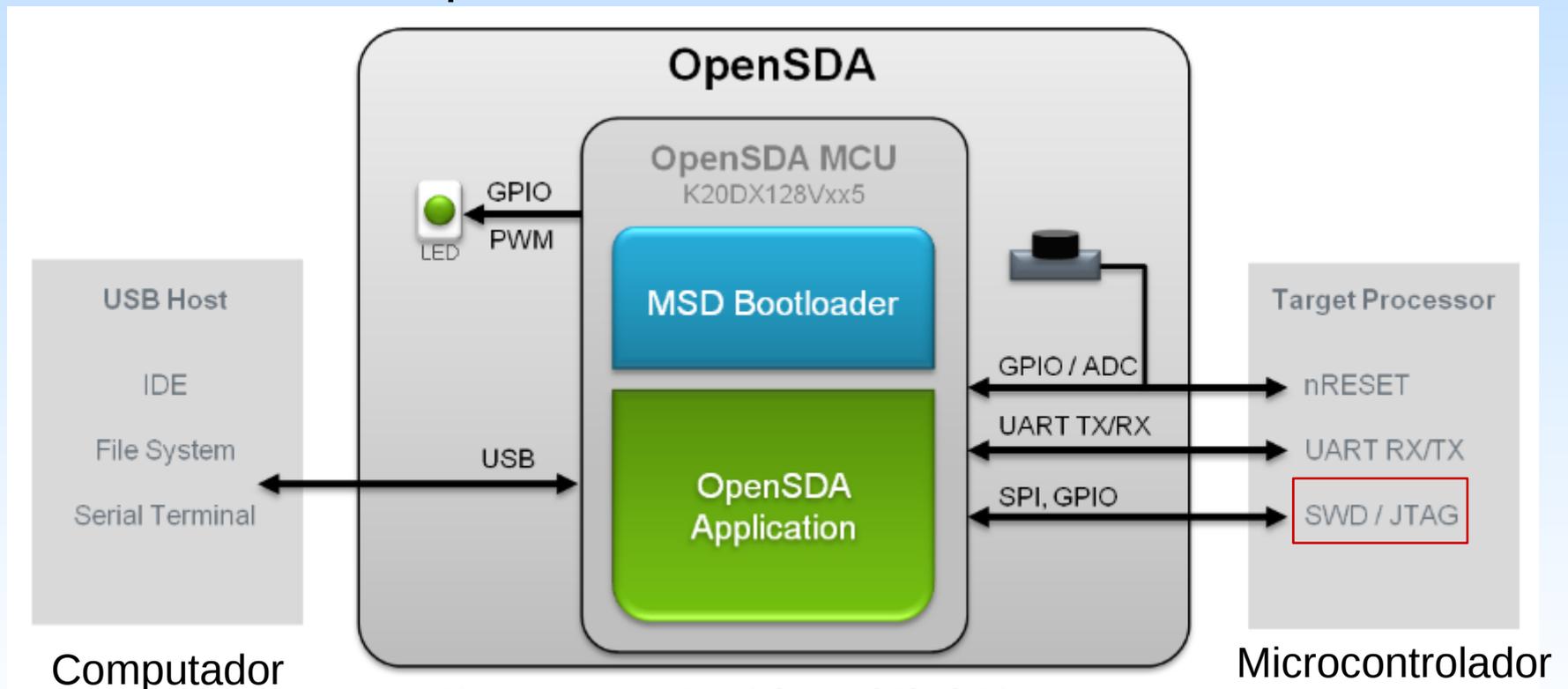
Figure 3. MK60N512VMD100 bootloader memory map

Gravação de um Novo Aplicativo

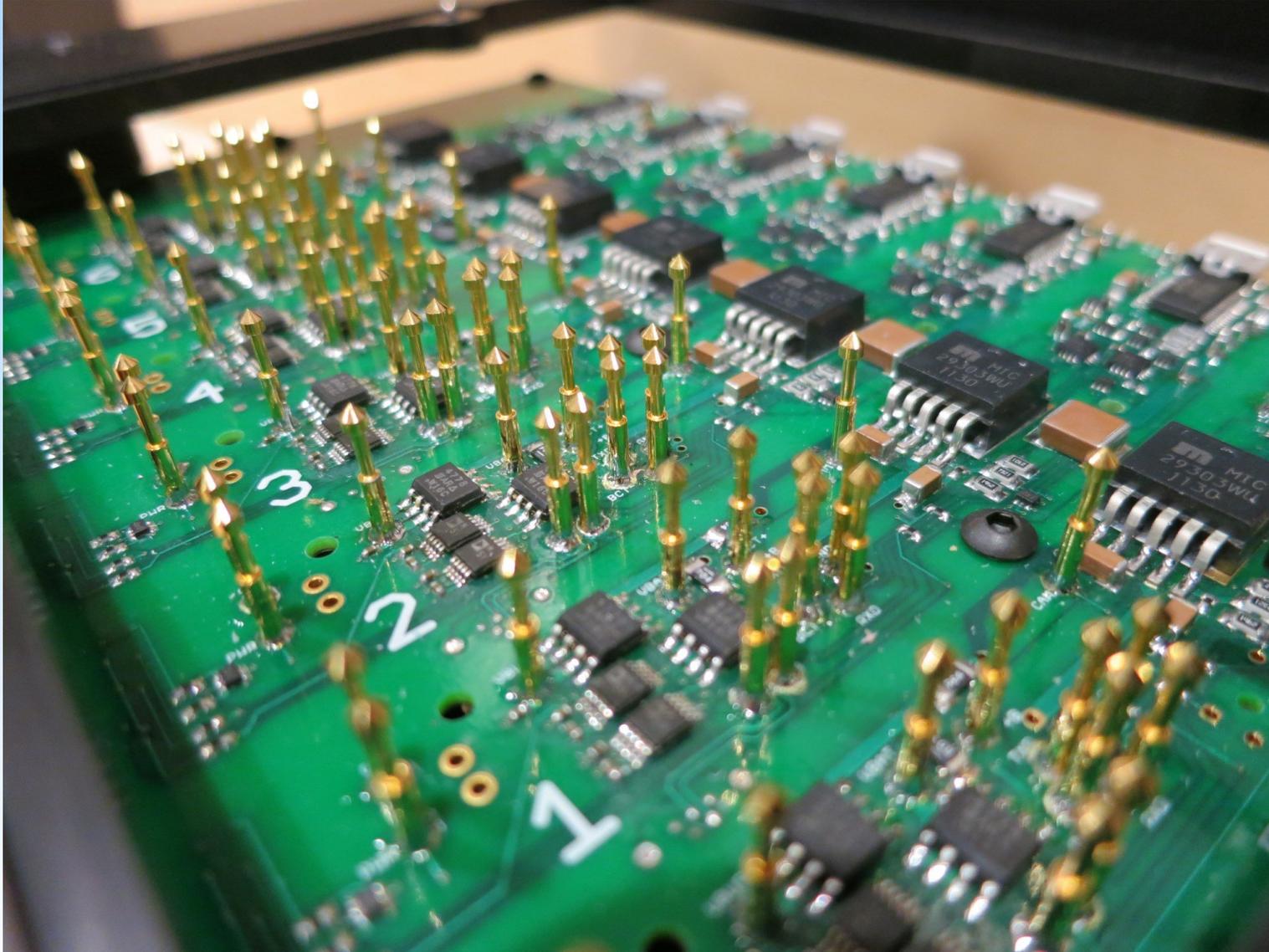
- Caso a porta USB contenha um aplicativo no formato válido (*s-record* ou hex), ele apaga o conteúdo da memória *flash* do microcontrolador e carrega o novo aplicativo com ajuda do ***MSD (mass storage device) bootloader***.

OpenSDA

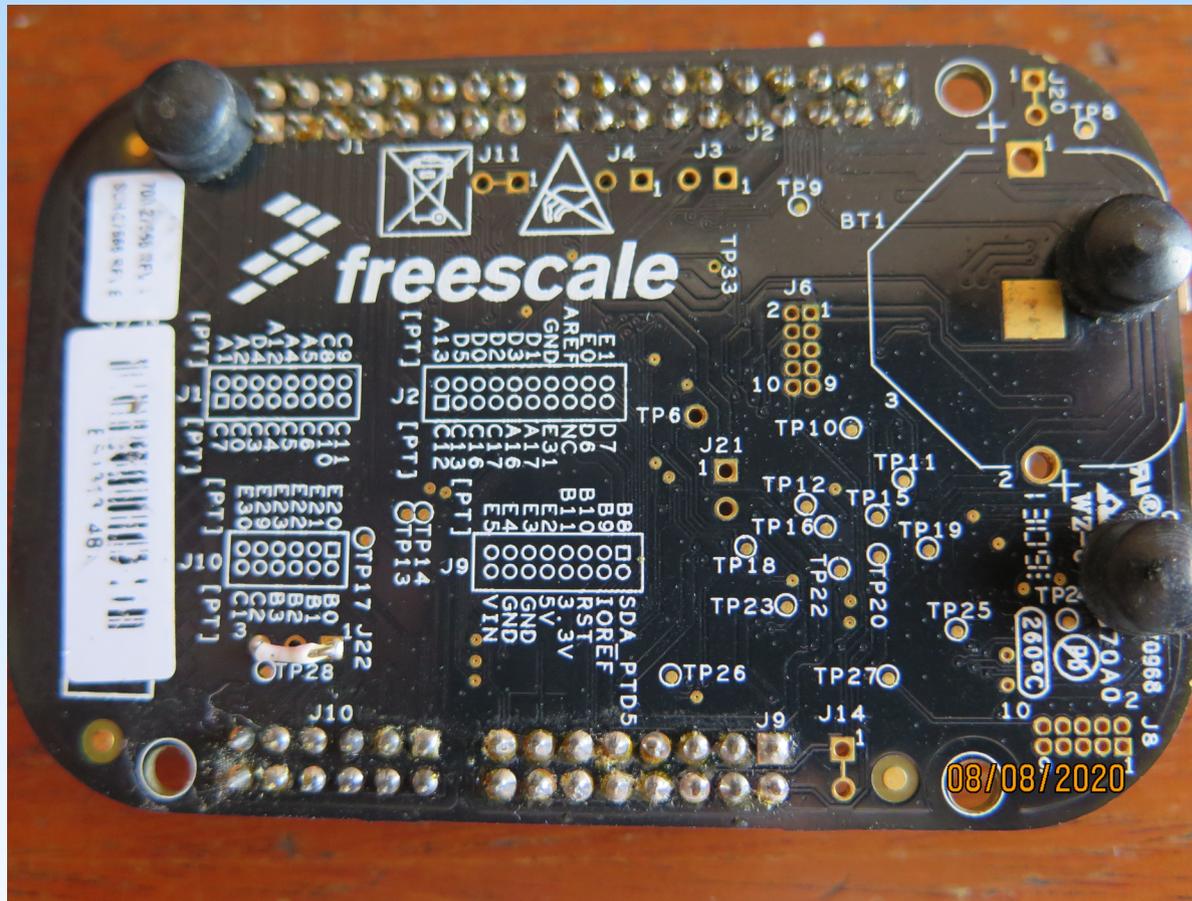
- É uma interface entre o computador onde roda o IDE CodeWarrior e o microcontrolador para o qual estamos desenvolvendo o programa. Ela serve para programar a memória *flash* do microcontrolador, **depurar o programa em execução no microcontrolador** e estabelecer a comunicação serial entre o computador e o microcontrolador.



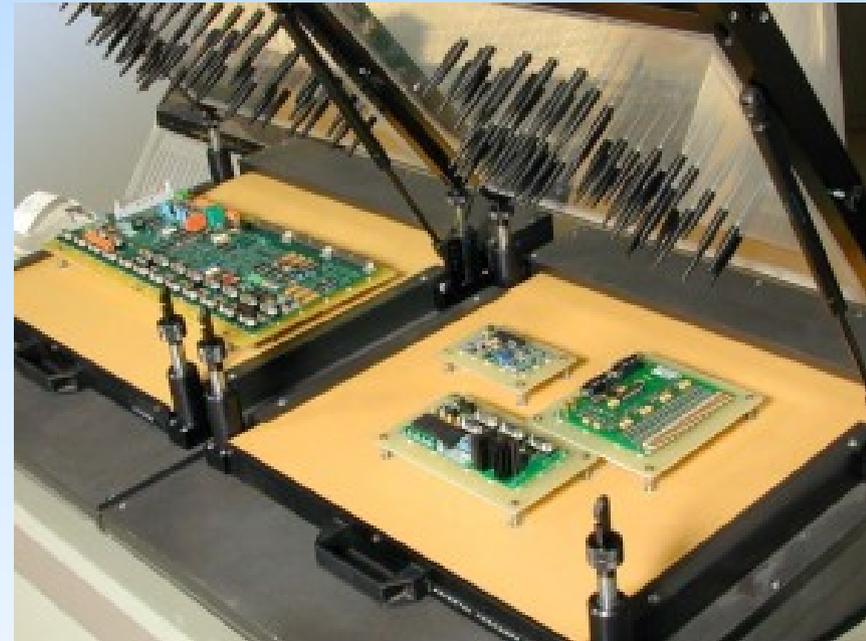
Testes de Placas de Circuito Impresso



Bed of nails



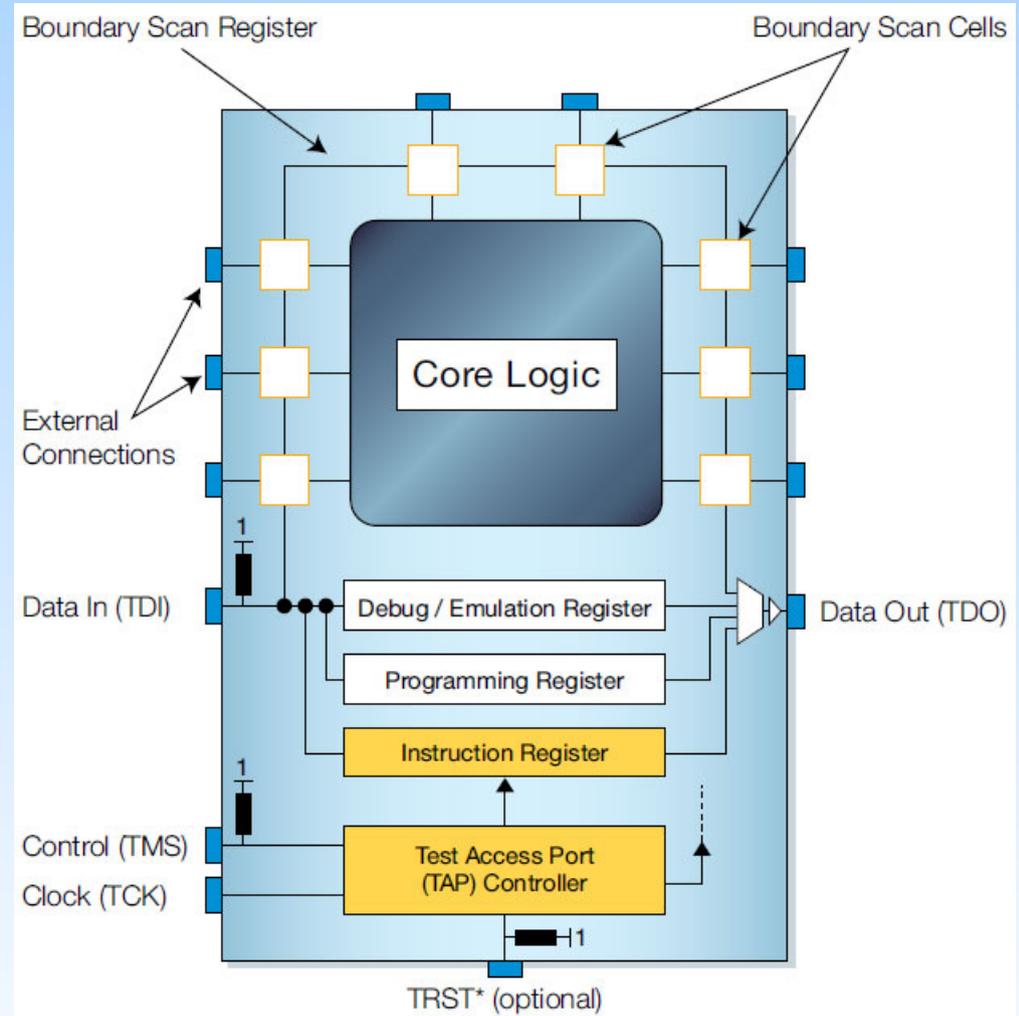
FRDM-kl25Z



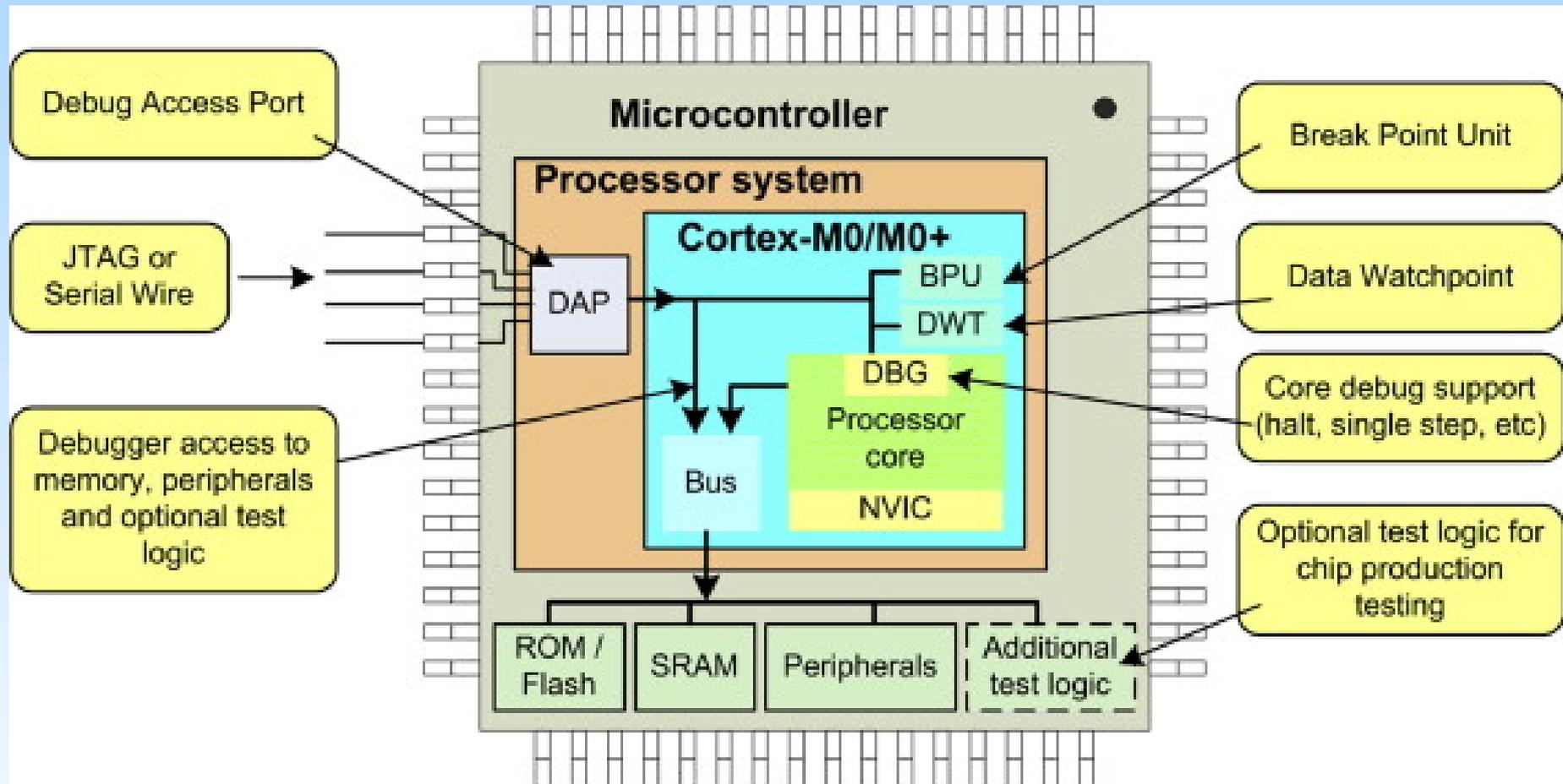
Bed of nails

JTAG

- *Joint Test Action Group* (1985): um padrão de interface (IEEE 1149.1) para testes de placas de circuito impresso por varredura de limites (*boundary scan test*)



CoreSight: Arquitetura de Debug



Fonte: Joseph Yiu, in The Definitive Guide to Arm® Cortex®-M0 and Cortex-M0+ Processors (Second Edition), 2015

SWD

- *Serial Wire Debug*

10-pin JTAG/SWD Connector

VCC	1	□	□	2	SDWIO / TMS
GND	3	□	□	4	SWDCLK / TCLK
GND	5	□	□	6	SWO / TDO
KEY	7		□	8	NC / TDI
GNDDetect	9	□	□	10	nRESET

SWD CONNECTOR

SHORTING HEADER ON BOTTOM LAYER

Jumper is shorted by a cut-trace on bottom layer. Cutting the trace will effectively isolate the on-board MCU from the OpenSDA debug interface.

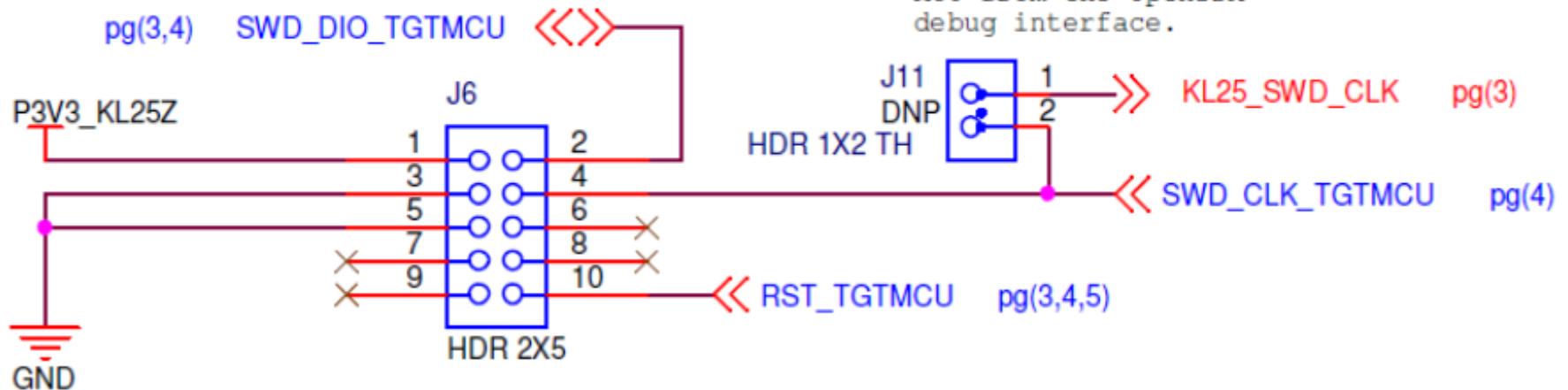


Figure 5. SWD Debug Connector

Conceitos

- **Resume/Continue:** *continuar a execução até o próximo ponto de parada ou próxima pausa.*
- **Step into:** *dar um passo para a próxima linha de instrução, entrando na rotina se houver.*
- **Step over:** *dar um passo para a próxima linha de instrução.*
- **Step out:** *retornar à linha de instrução em que a rotina corrente é chamada.*
- **Pause:** *pausar o fluxo de execução.*
- **Stop:** *parar o fluxo de execução.*

Conceitos

- **Reset:** *retornar ao endereço inicial do fluxo de execução.*
- **Breakpoint:** *linha em que deve parar o fluxo de execução.*
- **Watchpoint:** *variável cuja variação no estado gera parada no fluxo de controle.*
- Escopo de depuração
 - **Variáveis locais:** declaradas na rotina em depuração.
 - **Variáveis globais:** declaradas fora do escopo da rotina em depuração, mas acessíveis pela rotina.



CodeWarrior IDE Development Suite

Informações Adicionais

- *USB Mass Storage Device Host Bootloader*

<https://www.nxp.com/docs/en/application-note/AN4368.pdf>

- JTAG

<https://www.xjtag.com/about-jtag/what-is-jtag/>

- Extraíndo *firmware* com JTAG

<https://sergioprado.org/extraindo-firmware-com-jtag/>

- SWD

<https://www.cnblogs.com/shangdawei/p/4748751.html>

- CodeWarrior – Perspectiva de *Debug*

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/manuais/CWCFUG.pdf>



EA871

Ambiente de Desenvolvimento Integrado (IDE)

CodeWarrior – Construir um Aplicativo

Wu Shin – Ting
DCA – FEEC - Unicamp
Segundo Semestre de 2020

Conceitos

- **Programa:** uma sequência de instruções executadas por um processador em cima de um conjunto de dados, a fim de realizar uma tarefa específica.
- **Registradores de propósito geral:** registradores para armazenar os operandos e os resultados de uma instrução.
- **Contador de Programa (PC):** registrador que contém o endereço da memória cujo conteúdo é o endereço da próxima instrução de um programa. Este conteúdo é atualizado após cada acesso a uma instrução.

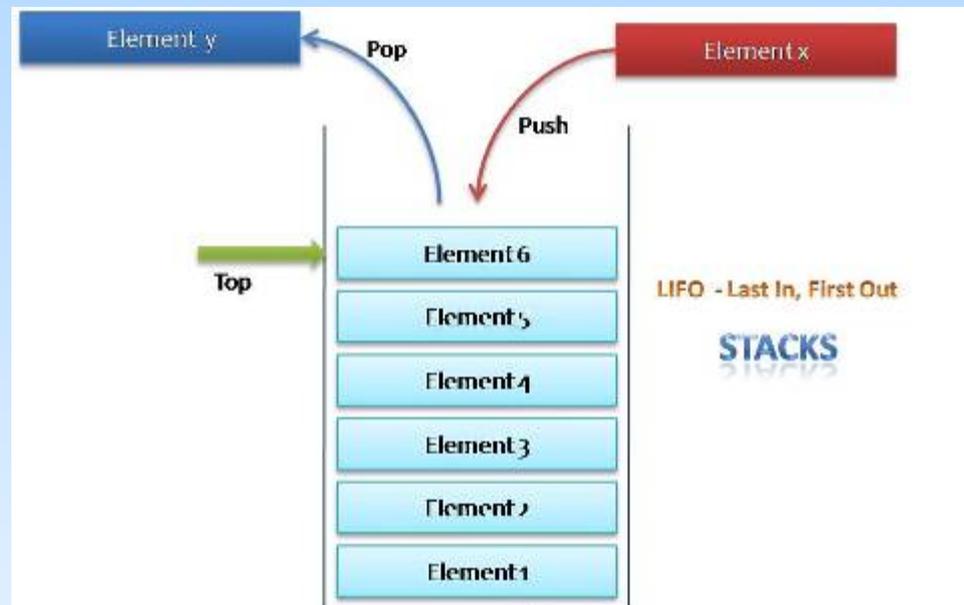
Conceitos

- **Vetor de interrupções:** uma tabela de endereços de memória que contém os endereços iniciais das rotinas de tratamento de interrupção.

Exception number	IRQ number	Offset	Vector
16+n	n	0x0040+4n	IRQn
.	.	.	.
.	.	.	.
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
13		0x0038	Reserved
12			Reserved for Debug
11	-5	0x002C	SVCcall
10			Reserved
9			
8			
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value

Conceitos

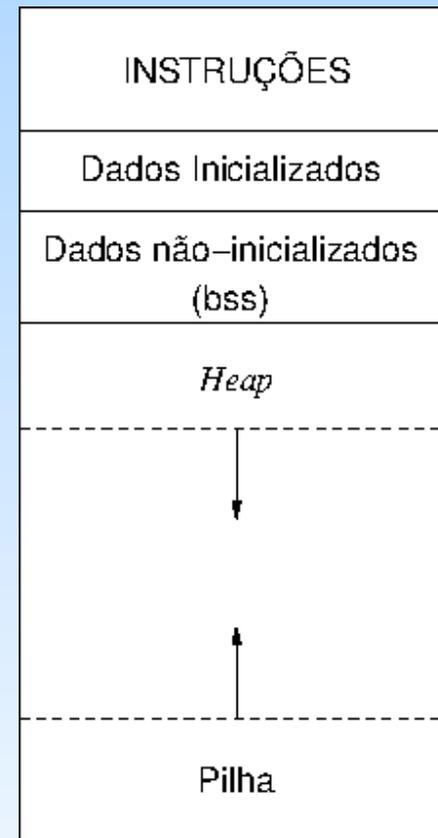
- **Pilha:** uma estrutura de dados em que os seus itens são empilhados (*Last In*) e desempilhados (*First Out*). É muito usada para armazenar dados temporários.



- **Ponteiro à Pilha (SP):** registrador que contém o endereço do topo da pilha. É automaticamente incrementado quando se empilha um novo dado e decrementado quando se desempilha algum dado do topo.

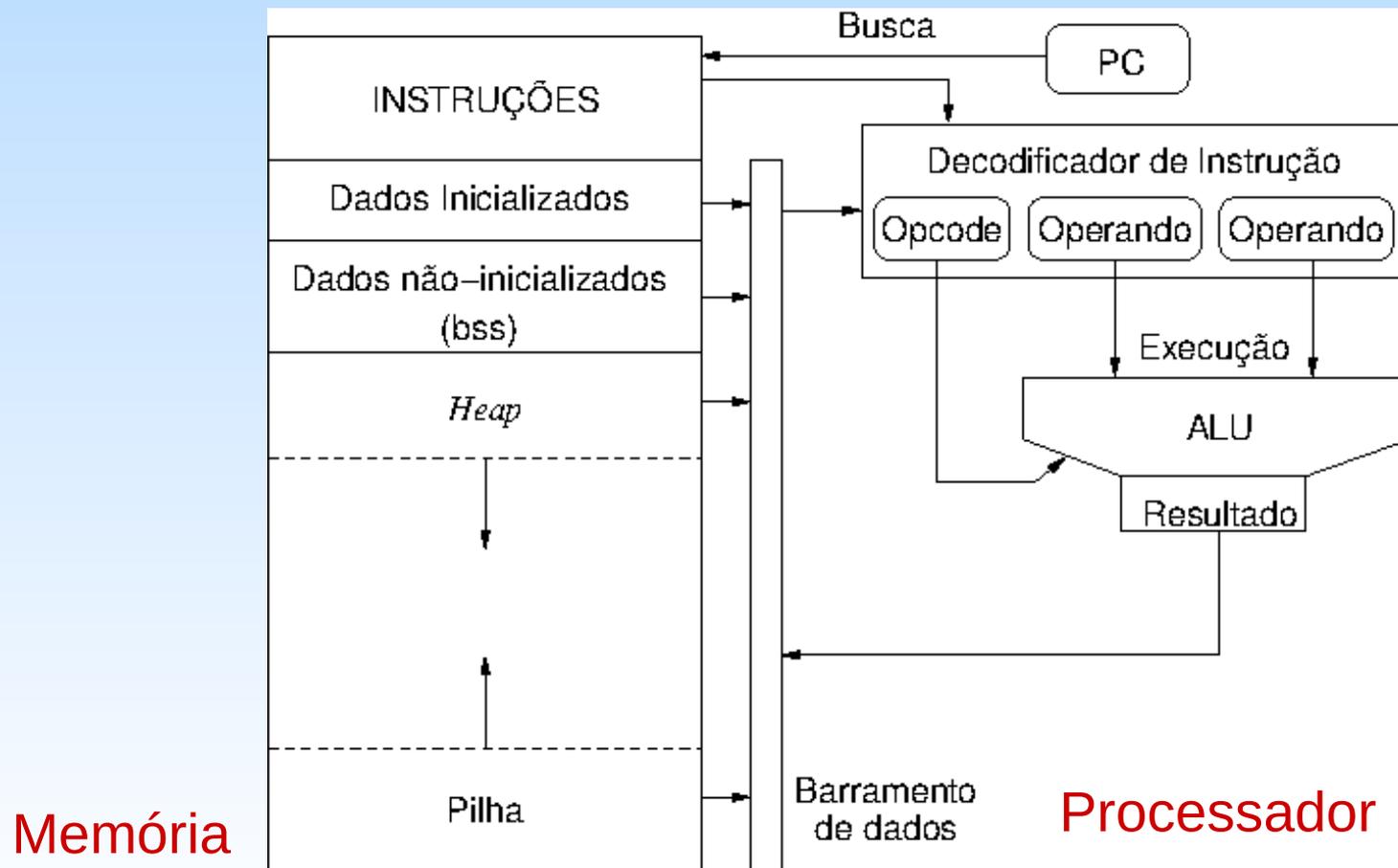
Conceitos

- **Segmento de instruções/.text (R)**: dados decodificáveis em sinais de controle (opcode).
- **Segmento de dados/.data**: operandos
 - Dados inicializados: constantes (R);
estáticos (R/W)
 - Dados não-inicializados (R/W)
 - Dados alocados dinamicamente,
p.ex. malloc ou calloc (*heap*),
durante a execução (R/W)
 - Dados de pilha (R/W)
- **Relocação**: ajuste dos endereços dos segmentos de instrução/dados aos espaços de endereços disponíveis (no microcontrolador).



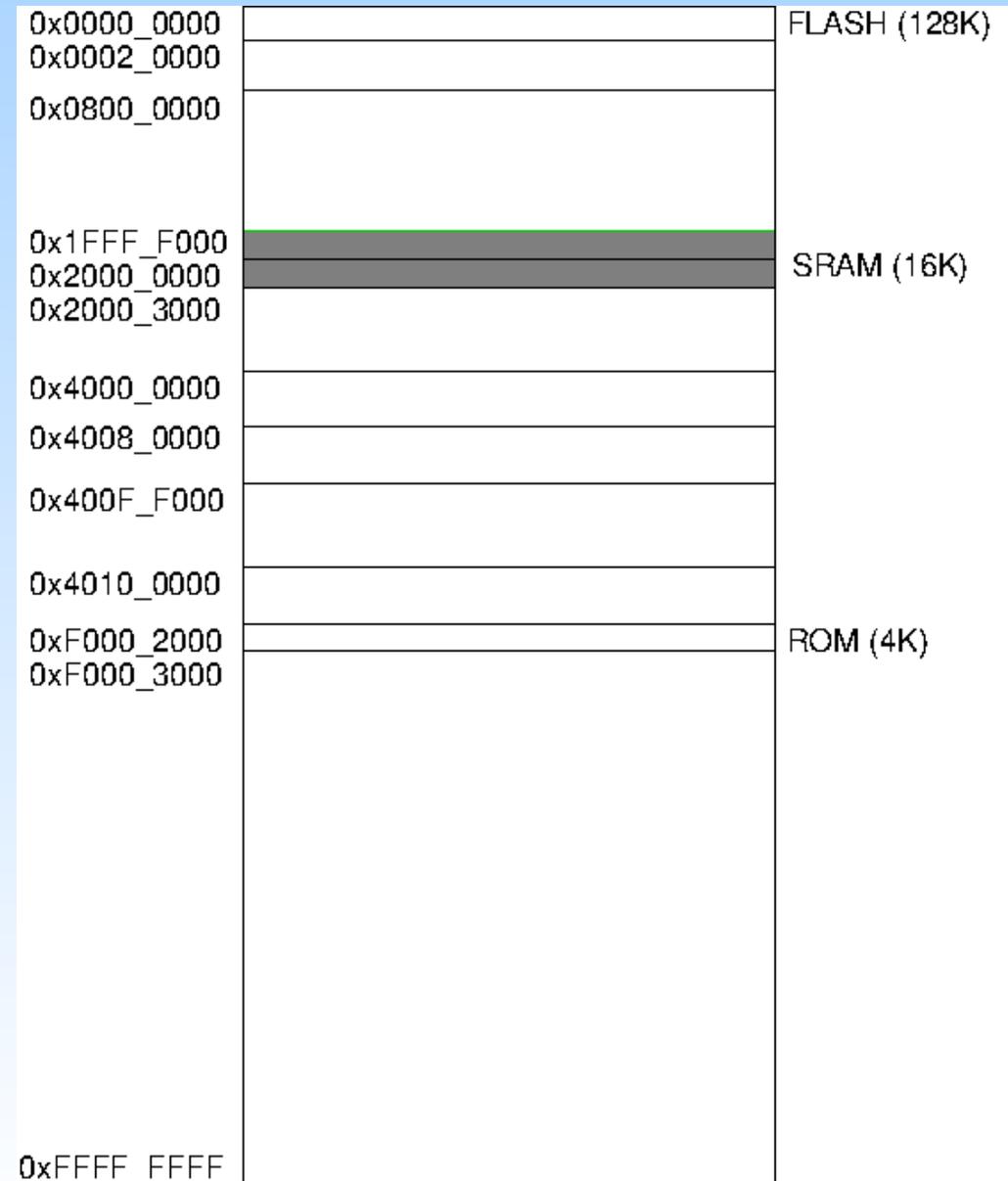
Conceitos

- **Ciclo de instrução:** é o intervalo de tempo que um processador leva para buscar, decodificar e executar uma instrução em código de máquina.



Mapa de Memória do Microcontrolador

- 32 *bits* → 2^{32} endereços distintos.
- Tipos de memória:
 - Flash: 0x0000_0000
128KB
 - SRAM: 0x1fff_f000
16KB
 - ROM: 0xf000_2000
4K



Relocação dos segmentos de instruções e de dados

- Qual é o espaço de endereços da memória em que se pode relocar segmentos de instruções (somente de leitura) e de dados?
- Qual deve ser o conteúdo do contador de programa para iniciar a execução de um programa?
- Qual é o endereço inicial do topo da pilha?

Estes endereços são configurados no script `MKL25Z128_flash.ld`, levando em conta que:

- `0x0000_0000`: Vetor de interrupções
 - `0x0000_0000`: SP
 - `0x0000_0004`: PC

Inicialização do microcontrolador

- Inicializar registradores de propósito geral
- Desabilitar as interrupções
- Desabilitar watchdog
- Inicializar o segmento .bss de dados não inicializados

Esta inicialização é feita pelas rotinas no
arquivo `_arm_start.c`

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KLQRUG.pdf> (Seção 1.1.4)

Inicialização do microcontrolador

- Definir os endereços das rotinas de serviço

Esta inicialização é feita pelas rotinas no
arquivo `kinetis_sysinit.c`



CodeWarrior IDE Development Suite

Informações Adicionais

- Mapa de memória (Seção 4.2, pg. 105)

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0RM.pdf>

- Kinetis L Peripheral Module Quick Reference:

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KLQRUG.pdf>