Fluxo de projeto Eletrônica Embarcada

Slide 3

Rafael Corsi - corsiferrao@gmail.com

January 24, 2016

Instituto Mauá de Tecnologia
EEN251 - Microcontroladores e Sistemas Embarcado

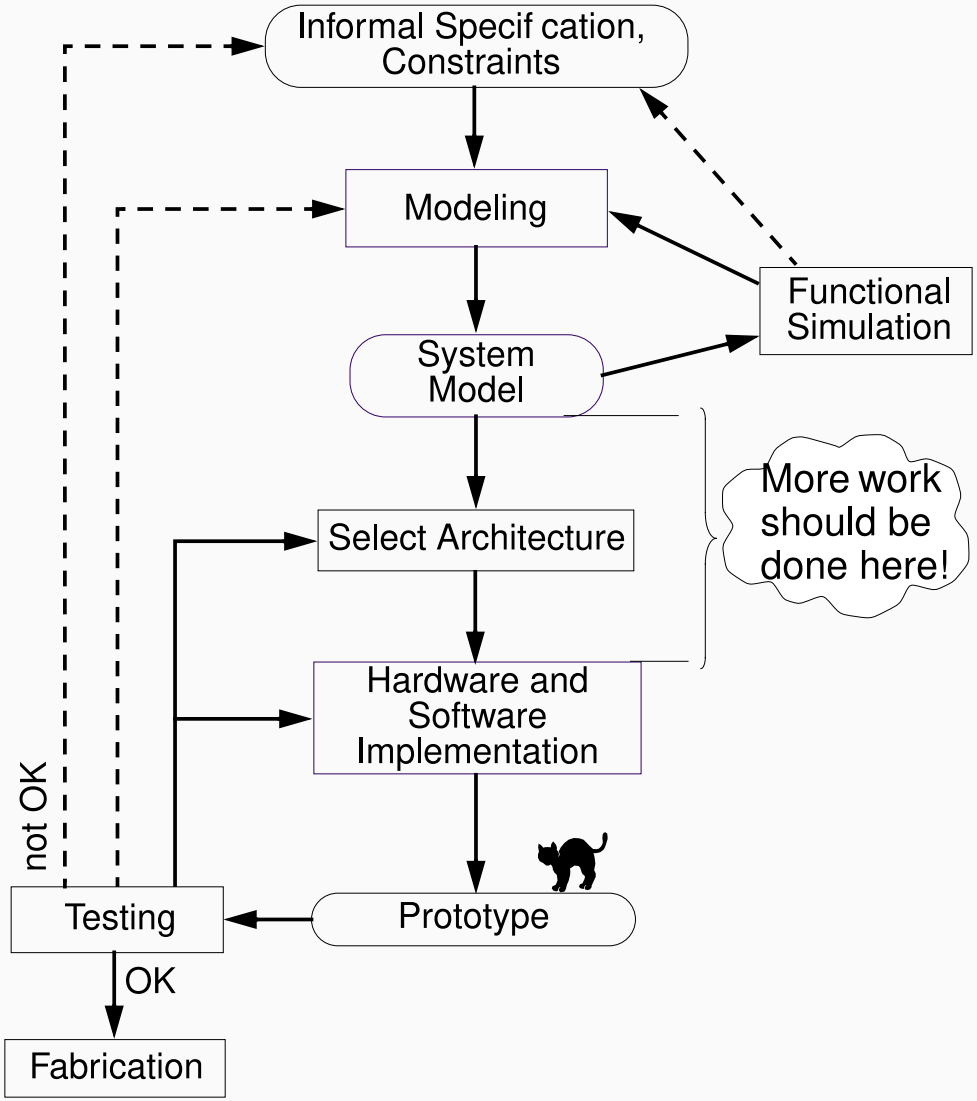
Conteúdo

Especificação

Visão geral

- Especificação
  - Objetivo, custo estimado, time to marketing, tecnologia utilizada, ...
  - Hardware e Software
- Modelagem : Diagrama de blocos, fluxogramas, cronograma
- Implementação : Onde será realizado a implementação tanto do HW quanto do SW

Fluxo

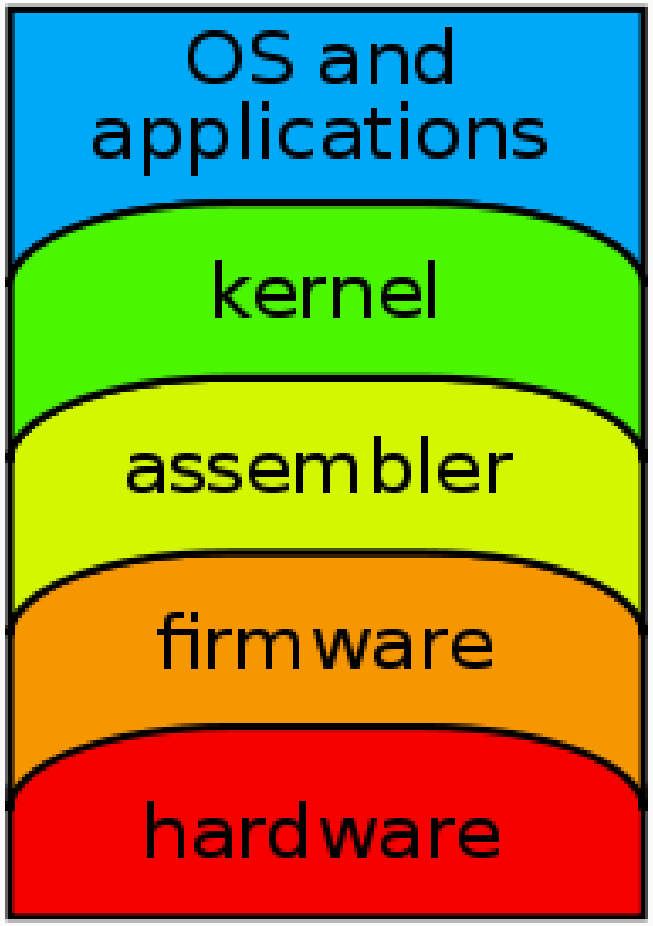

Informal Specif cation, Constraints
Modeling
Functional Simulation
System Model
More work should be done here!
Select Architecture
Hardware and Software Implementation
not OK
Testing
Prototype
OK
Fabrication

Firmware

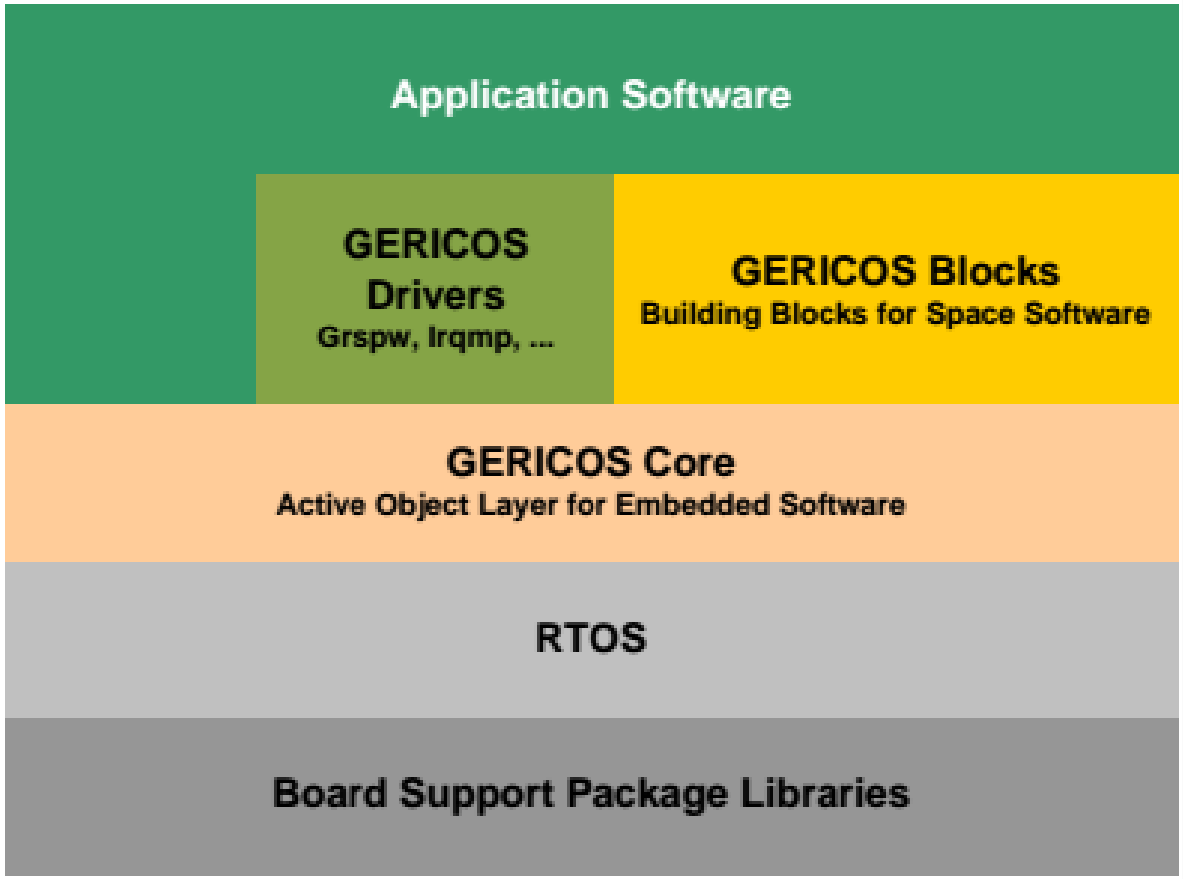
Chamamos de firmware o programa específico para uma eletrônica embarcada, ele se difere de um programa por ser otimizado para o Hardware e trabalharem em um nível mais baixo.

Camadas (layers)

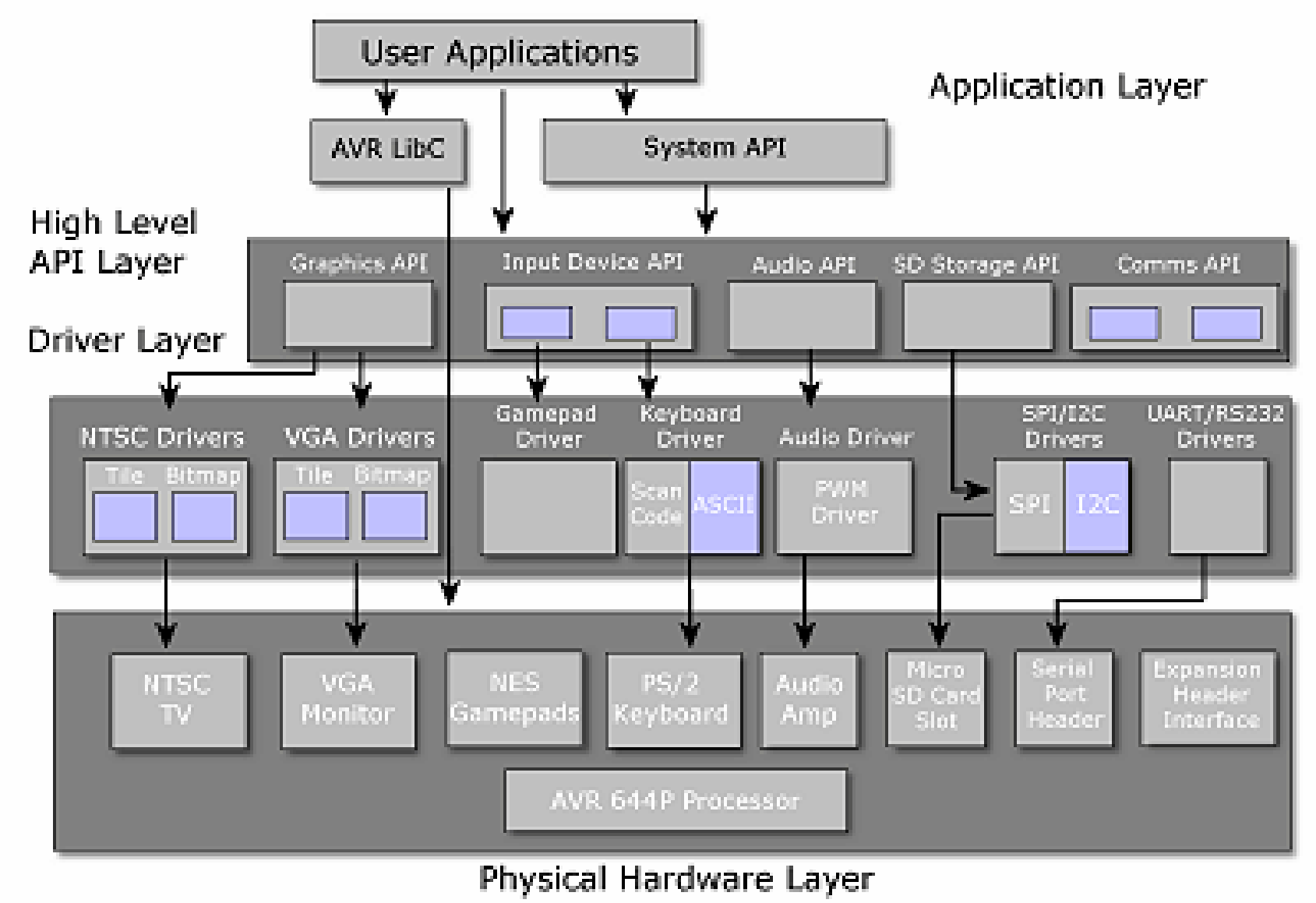
É comum no desenvolvimento de firmwares trabalharmos com o conceito de camadas.



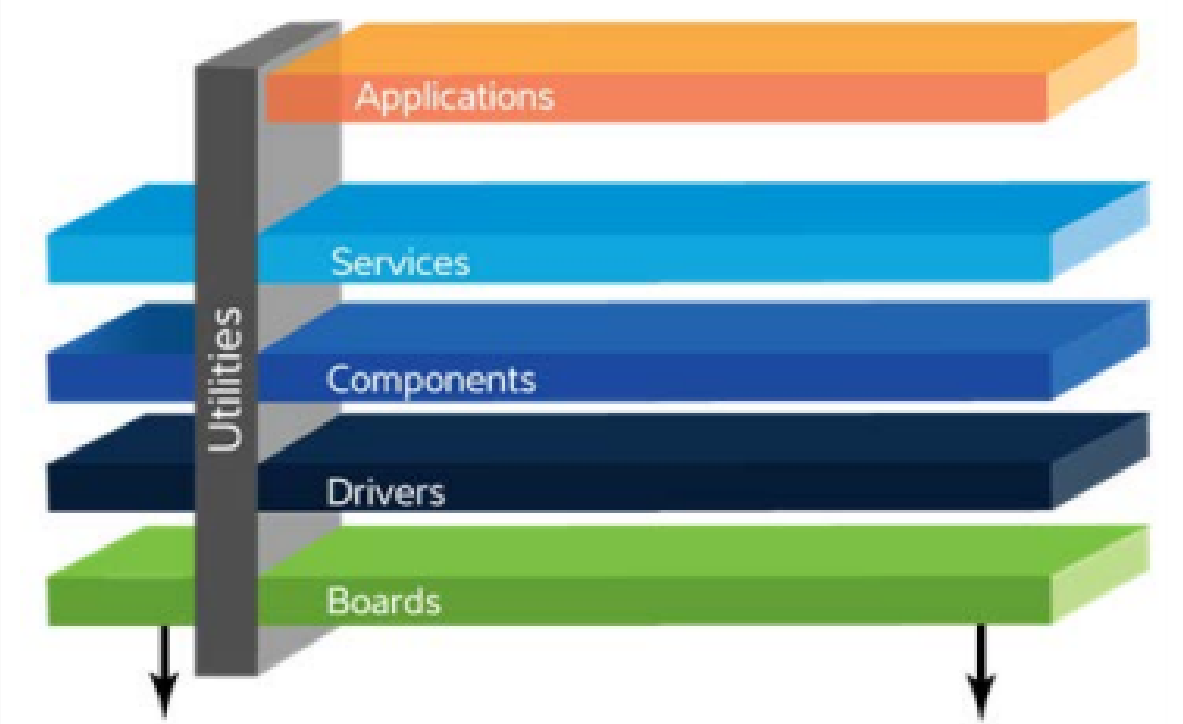
Exemplo camadas



Exemplo camadas



Atmel Software Framework



Vantagens

As principais vantagens de se trabalhar dessa maneira :

- Modulação do firmware
- Portabilidade (placas, uC)
- Fácil manutenção e entendimento do projeto
- Fácil documentação
- Otimização do projeto

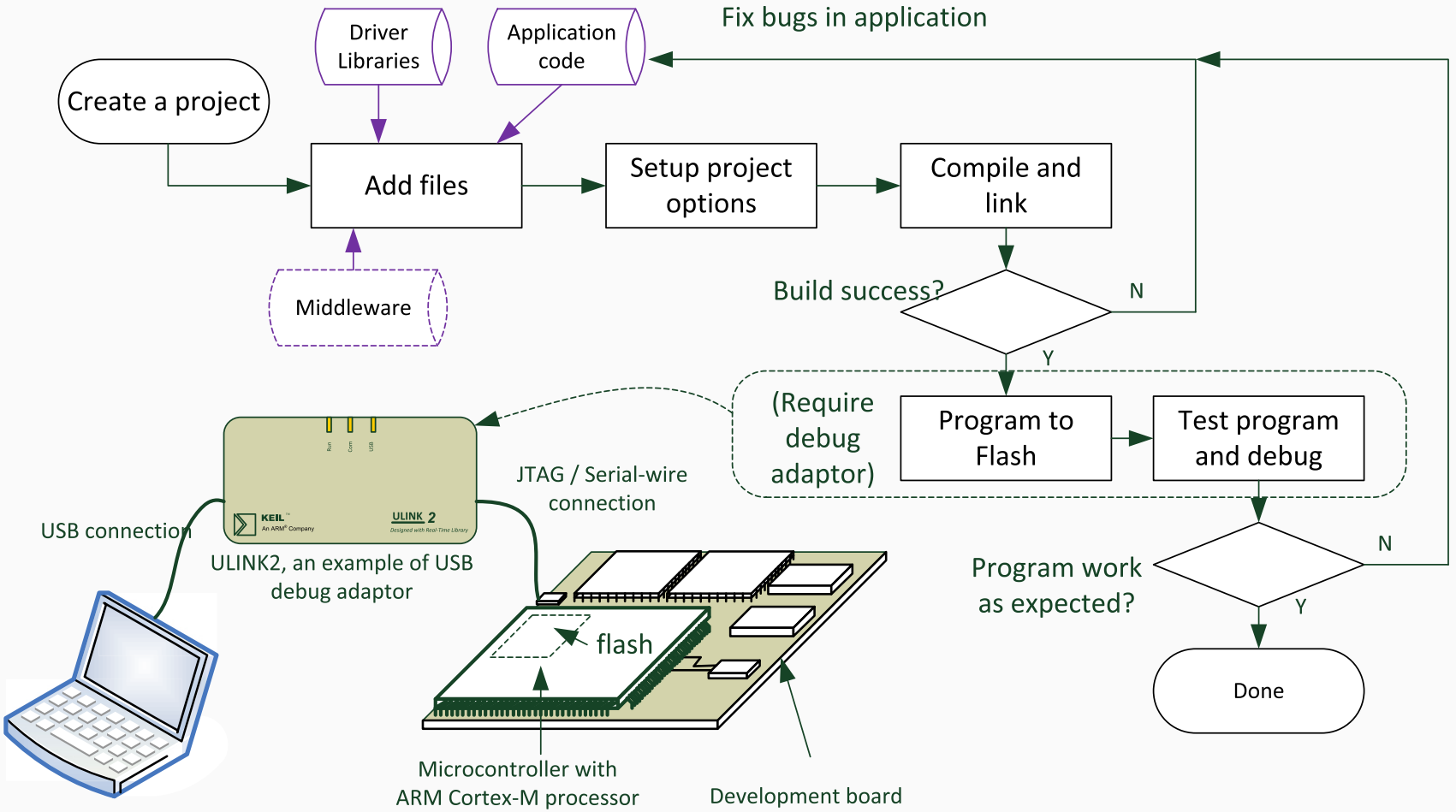
Fluxo de programação

Etapas da programação

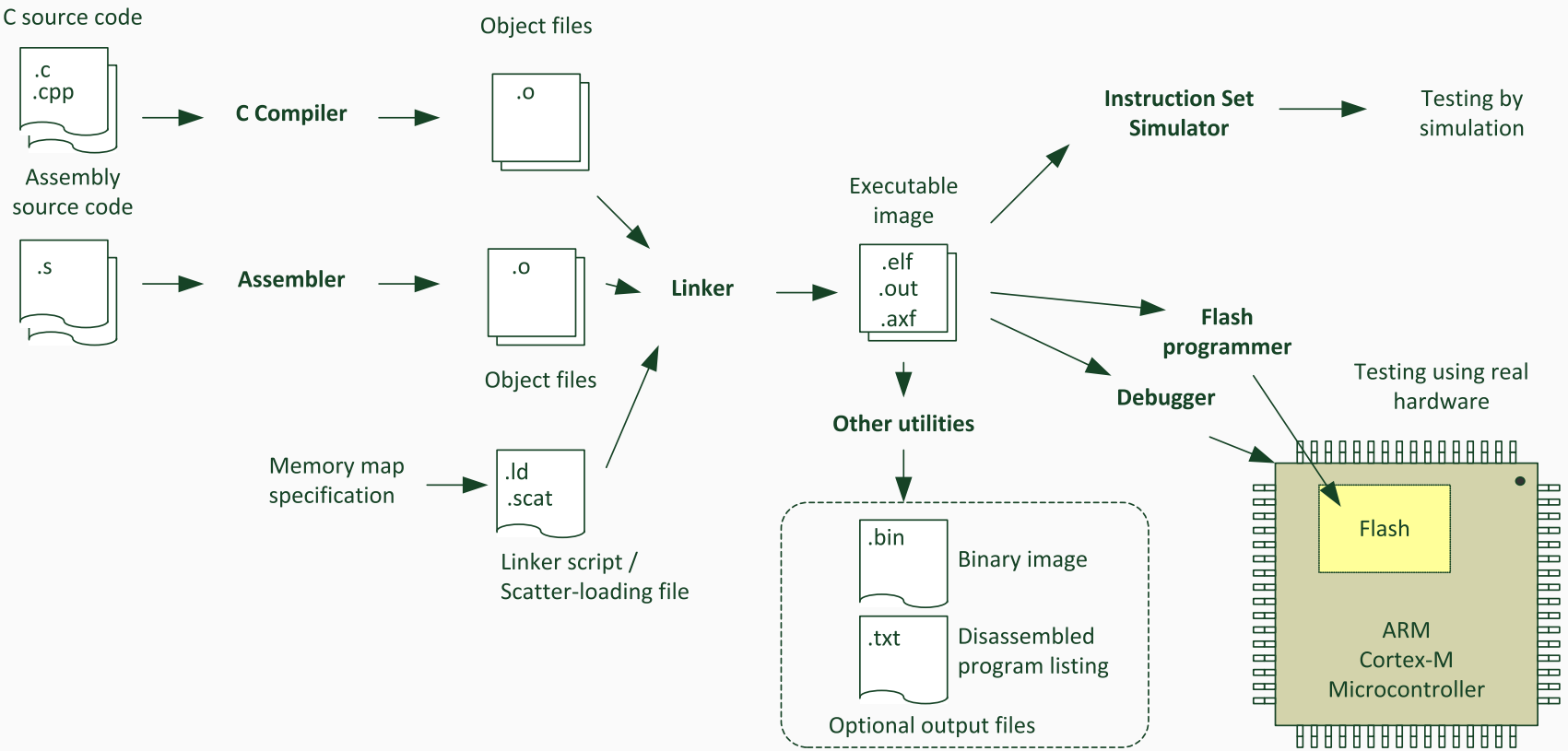
As seguintes etapas são executadas para a programação de um dispositivo embarcada :

1. Cria-se os arquivos contendo o programa (.c e .h)
2. um compilador transforma o código em linguagem de máquina
3. o linker aloca o programa em memória
4. um gravador é utilizado para programar a memória não volátil do microcontrolador
5. um gravador é utilizado para fazer depuração do projeto.

Fluxo de programação



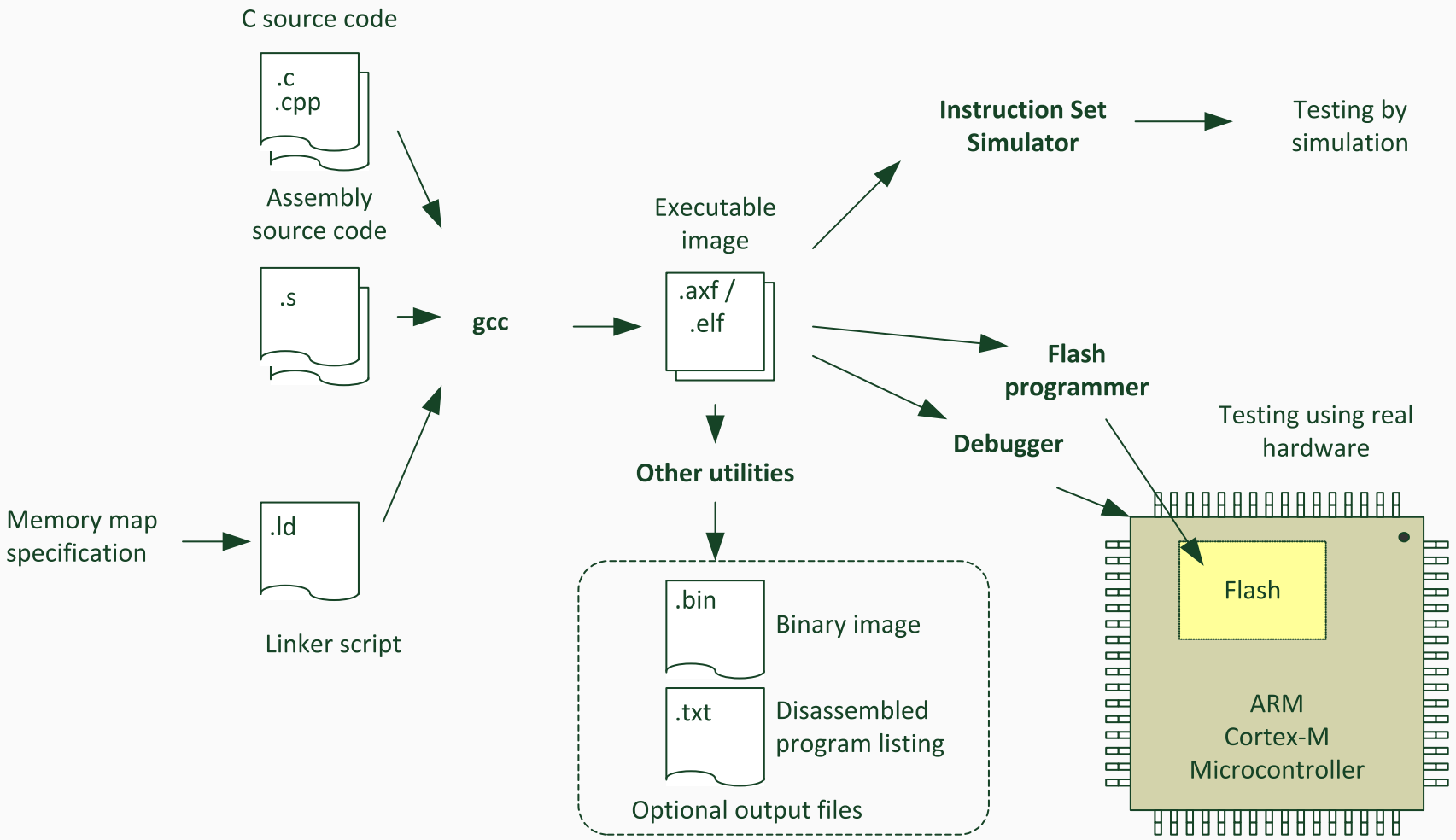
Compilação


C source code
.c
.cpp
C Compiler
Object files
.o
Assembly
source code
.s
Assembler
.o
Object files
Linker
Memory map
specification
.ld
.scat
Linker script /
Scatter-loading file
Executable
image
.elf
.out
.axf
Instruction Set
Simulator
Testing by
simulation
Flash
programmer
Debugger
Testing using real
hardware
Other utilities
.bin
Binary image
.txt
Disassembled
program listing
Optional output files
Flash
ARM
Cortex-M
Microcontroller

GNU compilador - GCC

É um dos principais compiladores existentes no mercado, desenvolvido sob licença de código livre (GNU) é utilizado tanto para compilar o kernel do linux, quanto para eletrônica embarcada.

Compilação GNU



Tipos de implementação

Tipos de implementação

Existem basicamente três maneiras diferentes de implementar um projeto em eletrônica embarcada, são elas :

- Polling - *while(1)*
- Interrupção
- MultiTask (Sistema operacional)

Pollling; super loop; while(1)

É a maneira mais direta de se implementar um projeto porém apresenta diversos pontos negativos.

Nesse método, todo o código é colocado dentro de um loop infinito - *while(1)*, onde executa-se as funções conforme necessário.

Não é possível implementar prioridades no código, sendo difícil sua manutenção.

Pollling; super loop; while(1)

É a maneira mais direta de se implementar um projeto porém apresenta diversos pontos negativos.

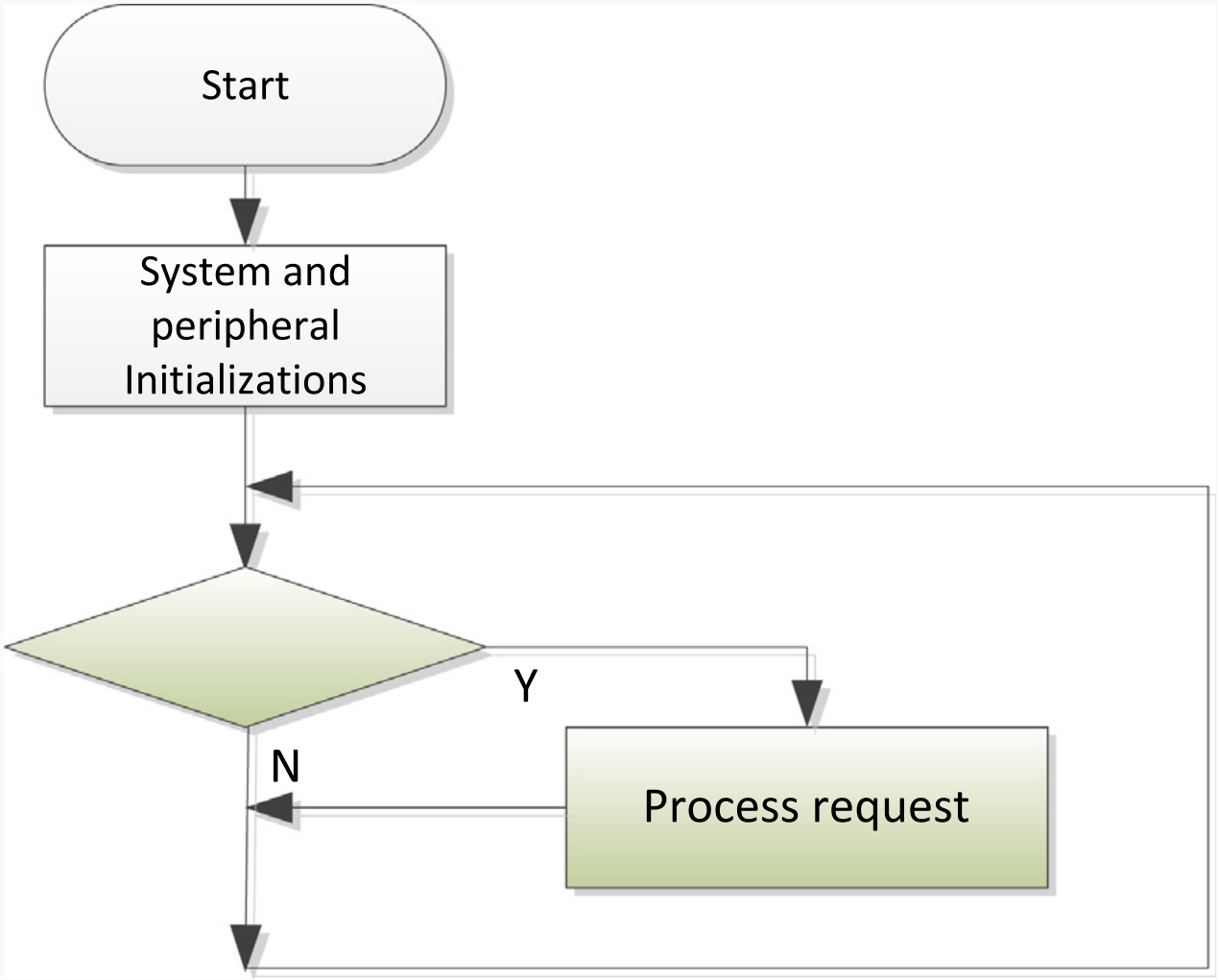
Nesse método, todo o código é colocado dentro de um loop infinito - *while(1)*, onde executa-se as funções conforme necessário.

Não é possível implementar prioridades no código, sendo difícil sua manutenção.

Principais desvantagens :

- Pode travar o programa
- não é energeticamente eficiente
- difícil de escalonar

Polling

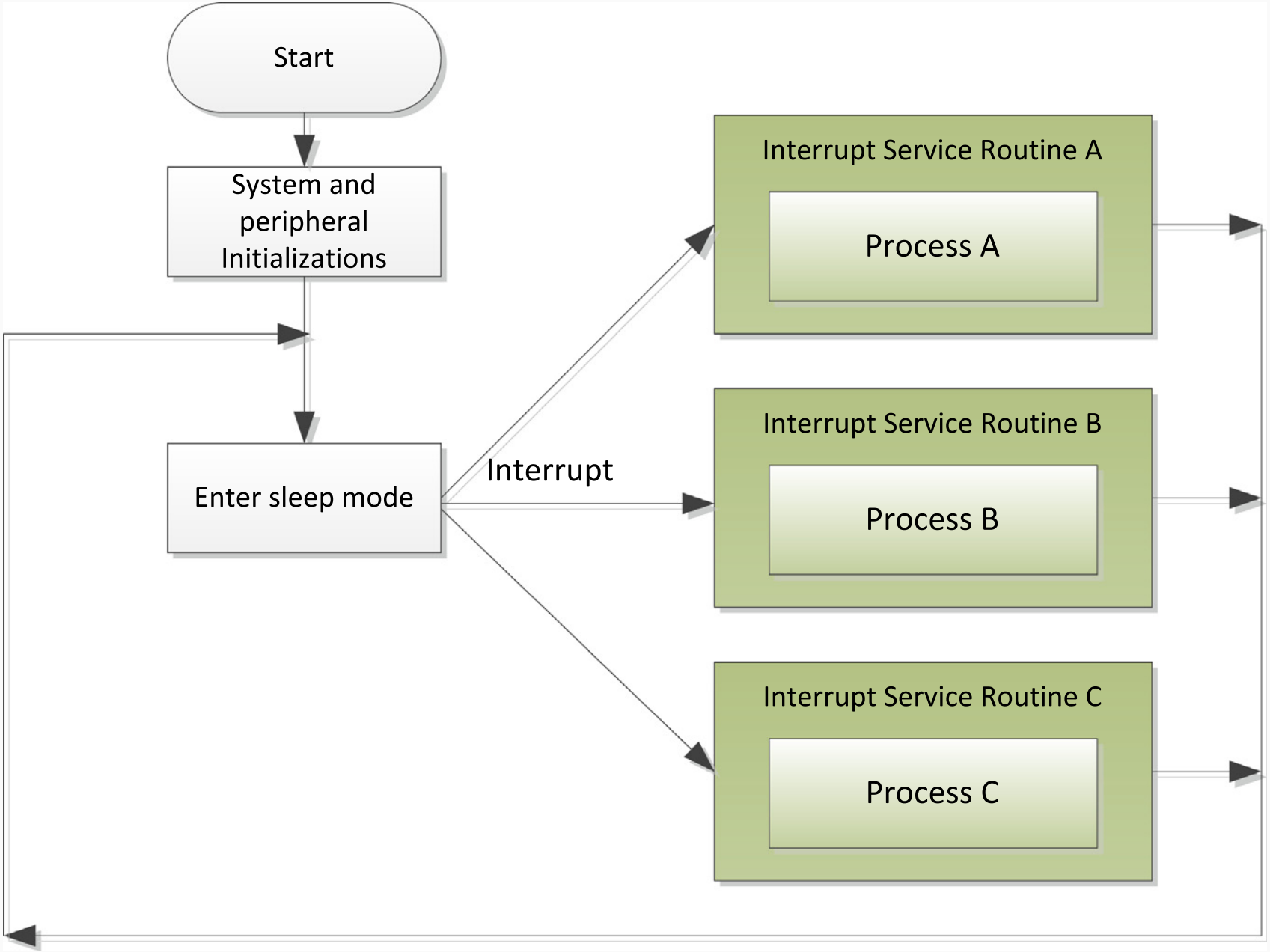

Start
System and peripheral Initializations
Peripheral requires processing?
Y
N
Process request

Interrupção

É a maneira mais eficiente de se implementar um projeto em eletrônica embarcada. Configura-se os periféricos para gerarem uma interrupção quando um estado pré determinado foi alcançado.

O processador fica em modo sleep e somente é acordado quando ocorre uma interrupção, então executa-se as ações atreladas a essa interrupção e retorna para o modo de baixo consumo energético.

Interrupção



Misto - Polling + Interrupção

As vezes é necessário processar uma grande quantidade de dados que não seria possível dentro da interrupção, portanto utiliza-se uma forma hibrida que mistura os dois métodos:

1. O uC fica em modo sleep
2. ocorre uma interrupção
3. os dados referentes a interrupção são processados no super loop
4. retorna-se para o modo sleep

Sistema Operacional

Utiliza-se nesse caso um programa auxiliar (OS) que possibilita a implementação de *tarefas*.

Um tipo de sistema operacional comummente utilizado em eletrônica embarcada é o Real Time Operational System (RTOS).

Linux, Android e iOS são sistemas operacionais para eletrônica embarcada.

Sistema Operacional