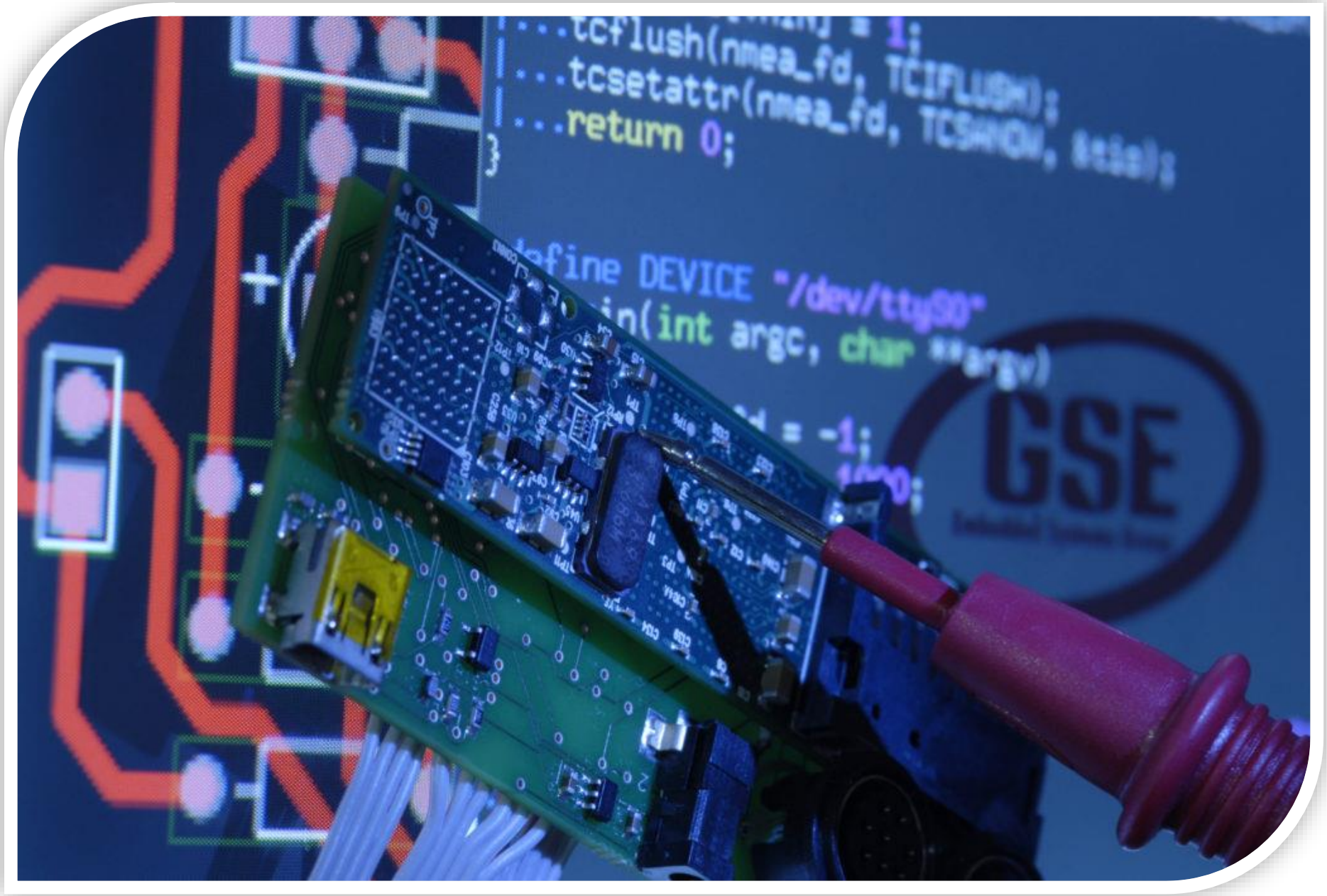
EGM0017 (60h)

Fluxo e metodologias de projeto de Sistemas Embarcados

Prof. **Josenalde Barbosa de Oliveira** – **UFRN**

 josenalde.oliveira@ufrn.br

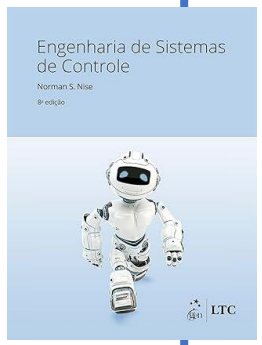
Programa de Pós-Graduação em Engenharia Mecatrônica

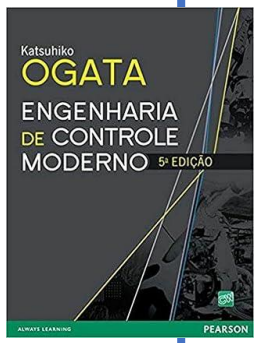


Fonte: Grupo de pesquisa em Sistemas Embarcados -UFSC
http://ppgeel.posgrad.ufsc.br/files/2017/12/foto_artistica_agritec.jpg

Apresentação – componentes relacionados

- Não há pré-requisito cadastrado, contudo entende-se como uma continuidade com foco em projeto/desenvolvimento de protótipo (e produto...) com RTOS

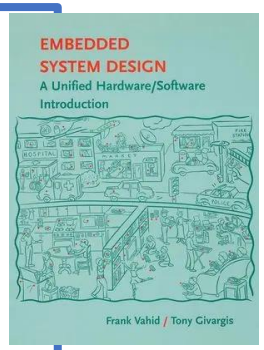


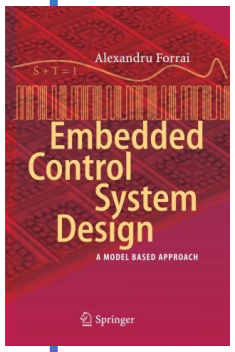


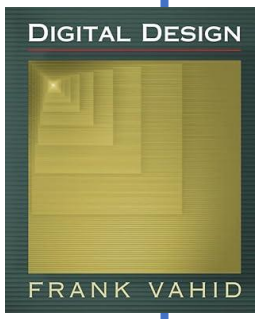
No PPGEMECA:

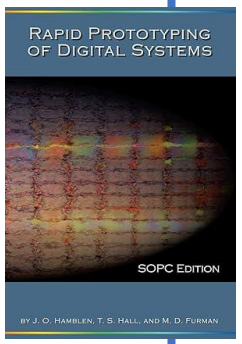
EGM0029 Sistemas Embarcados para Controle e Automação

Introdução aos sistemas de controle e automação; Contexto de controle e automação em sistemas embarcados; Arquitetura de sistemas embarcados para controle e automação; Instrumentação para controle e automação; Sistemas de atuação em controle e automação; Projeto de sistemas embarcados para controle e automação.





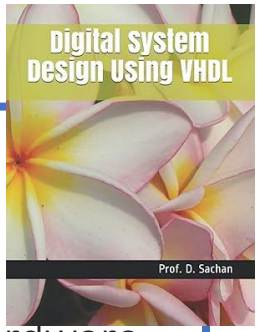


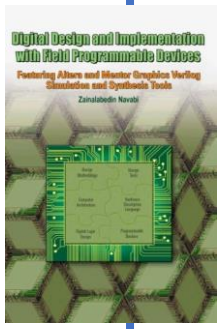


No PPGEMECA:

EGM0018 Projeto e síntese de sistemas digitais

Linguagem de Descrição de Hardware; Metodologia estruturada de projeto de sistemas digitais; Projeto de controladores digitais (MdE), prototipagem; Operadores digitais (componentes datapath); Projeto a nível de transferência de registros (RTL design); Técnicas de Otimização de projetos baseado em PLD; Intro uCs e uPs





Apresentação - conteúdo

- Conteúdo planejado (ementa):

Sistemas operacionais de tempo real (Real Time OS)

Introdução aos sistemas embarcados

Especificação e modelagem de sistemas embarcados

Metodologias de projeto baseado em plataforma

Particionamento entre hw e sw e camada de interfaceamento

Implementação e validação/teste de sistemas embarcados

Apresentação – rtos x bare metal

https://www.osrtos.com/

NuttX – LEV2

Demo LEV2

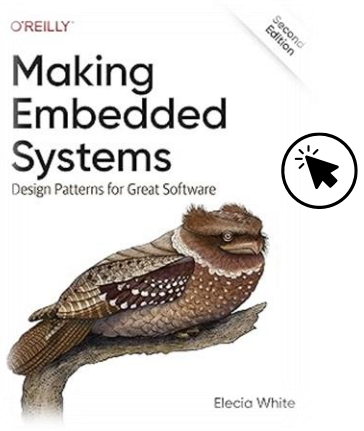
Apresentação - conteúdo

- Conteúdo planejado (ementa):

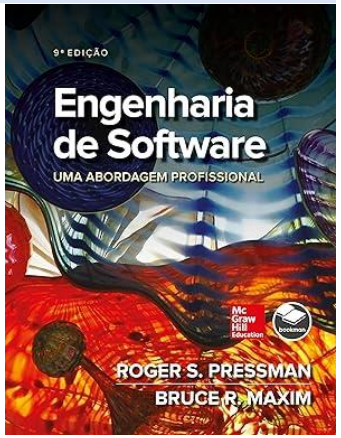
Sistemas operacionais de tempo real (Real Time OS)

- Introdução aos sistemas embarcados
- Especificação e modelagem de sistemas embarcados
- Metodologias de projeto baseado em plataforma
- Particionamento entre hw e sw e camada de interfaceamento
- Implementação e validação/teste de sistemas embarcados

Processo de Desenvolvimento de Software ampliado para **Processo de Desenvolvimento de Sistema Embarcado** (hardware + software embarcado: firmware)



+



Apresentação - conteúdo

- Conteúdo planejado (ementa):

Sistemas operacionais de tempo real (Real Time OS)

Introdução aos sistemas embarcados

Especificação e modelagem de sistemas embarcados

Metodologias de projeto baseado em plataforma

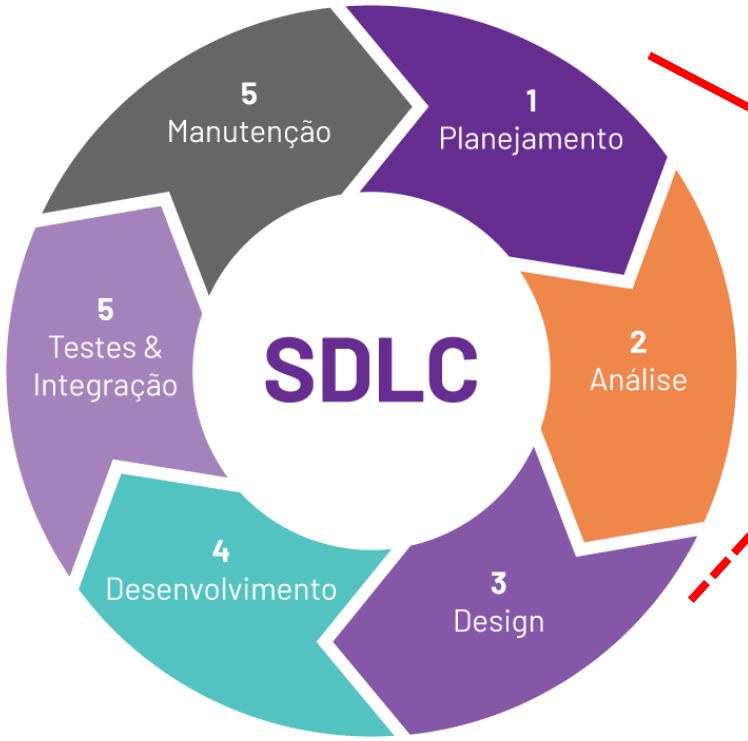
Particionamento entre hw e sw e camada de interfaceamento

Implementação e validação/teste de sistemas embarcados

Processo de Desenvolvimento de Software ampliado para **Processo de Desenvolvimento de Sistema Embarcado** (hardware + software embarcado: firmware)

O firmware é o código compilado residente diretamente no hardware de equipamentos eletrônicos. Também conhecido por “software embarcado” é um tipo de programa residente fundamental para a operacionalidade específica do dispositivo. Na prática, o firmware funciona como uma espécie de “sistema operacional” de aparelhos eletrônicos. Ele fica gravado diretamente no chip de memória ROM de seus hardwares (PROM, EPROM, Flash ROM), podendo ser regravado/atualizado (update)

Apresentação – uma olhada em PDSoftware(E)



Esforço!
- Requisitos
- Viabilidade
- Artefatos/docs
- Protótipos/UI

Apresentação – uma olhada em PDSoftware(E)

Implantação (produção)

1. Planejamento: **diálogo** com stakeholders com brainstorming para ideias de mais alto nível, em busca de delimitar o escopo do projeto em termo de caso(s) de uso(s) – o que deve fazer, para quem (quais usuários), se há interação com outros sistemas. Pode ocorrer pesquisa de mercado, viabilidade, estimativas de custo/prazo. Pode incluir possíveis riscos e um cronograma do projeto.

Apresentação – uma olhada em PDSoftware(E)

Implantação (produção)

1. Planejamento: **diálogo** com stakeholders com brainstorming para ideias de mais alto nível, em busca de delimitar o escopo do projeto em termo de caso(s) de uso(s) – o que deve fazer, para quem (quais usuários), se há interação com outros sistemas. Pode ocorrer pesquisa de mercado, viabilidade, estimativas de custo/prazo. Pode incluir possíveis riscos e um cronograma do projeto.

2. Análise: elaboração de um documento de especificação de **requisitos** de software (ERS), que inclui as funções do software (requisitos funcionais), restrições quantitativas a atender (tempo de resposta, consumo energético, plataformas compatíveis), diagramas e descrição de casos de uso (UML/SysML).

Apresentação – uma olhada em PDSoftware(E)

Implantação (produção)

1. Planejamento: **diálogo** com stakeholders com brainstorming para ideias de mais alto nível, em busca de delimitar o escopo do projeto em termo de caso(s) de uso(s) – o que deve fazer, para quem (quais usuários), se há interação com outros sistemas. Pode ocorrer pesquisa de mercado, viabilidade, estimativas de custo/prazo. Pode incluir possíveis riscos e um cronograma do projeto.

2. Análise: elaboração de um documento de especificação de **requisitos** de software (ERS), que inclui as funções do software (requisitos funcionais), restrições quantitativas a atender (tempo de resposta, consumo energético, plataformas compatíveis), diagramas e descrição de casos de uso (UML/SysML).

3. Design (Projeto): **definir** arquitetura do software (componentes/classes e suas relações – UML/SysML) – diagramas comportamentais, estruturais, implantação. Gera Documento de Design de Software (DDS) para equipe de desenvolvimento com MODELOS

Apresentação – uma olhada em PDSoftware(E)

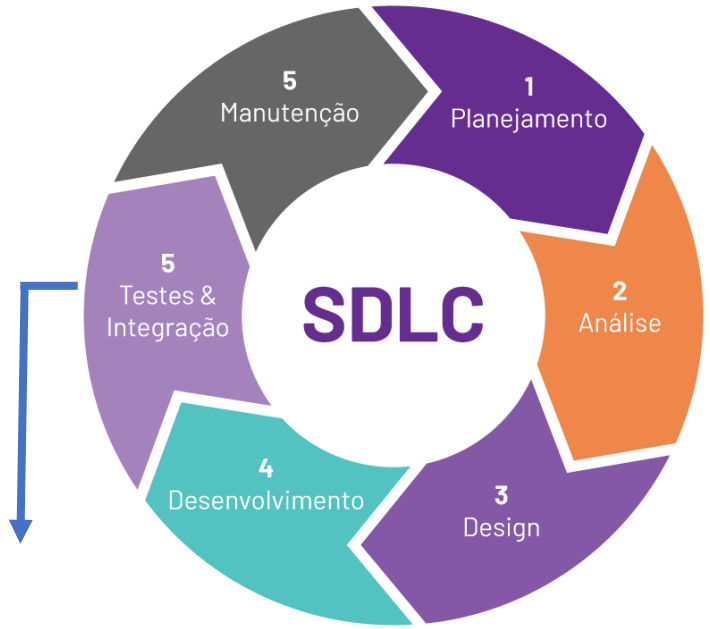
1. Planejamento: **diálogo** com stakeholders com brainstorming para ideias de mais alto nível, em busca de delimitar o escopo do projeto em termo de caso(s) de uso(s) – o que deve fazer, para quem (quais usuários), se há interação com outros sistemas. Pode ocorrer pesquisa de mercado, viabilidade, estimativas de custo/prazo. Pode incluir possíveis riscos e um cronograma do projeto.

2. Análise: elaboração de um documento de especificação de **requisitos** de software (ERS), que inclui as funções do software (requisitos funcionais), restrições quantitativas a atender (tempo de resposta, consumo energético, plataformas compatíveis), diagramas e descrição de casos de uso (UML/SysML).

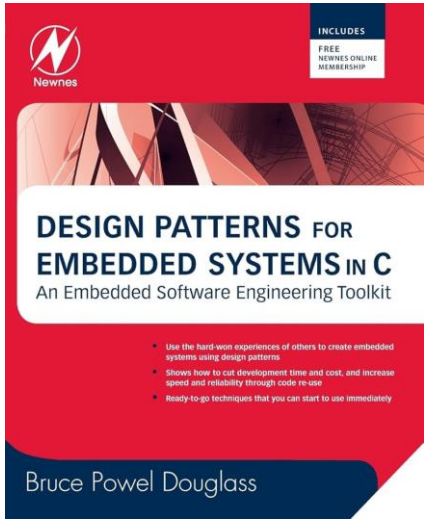
3. Design (Projeto): **definir** arquitetura do software (componentes/classes e suas relações – UML/SysML) – diagramas comportamentais, estruturais, implantação. Gera Documento de Design de Software (DDS) para equipe de desenvolvimento com MODELOS

4. Desenvolvimento: com base no ERS e DDS, definição de linguagem, ambiente de execução (stand-alone, web, mobile), co-design firmware, hardware alvo, software de aplicação, interfaces (APIs)

Apresentação – uma olhada em PDSoftware(E)



Implantação (produção)



1. Planejamento: **diálogo** com stakeholders com brainstorming para ideias de mais alto nível, em busca de delimitar o escopo do projeto em termo de caso(s) de uso(s) – o que deve fazer, para quem (quais usuários), se há interação com outros sistemas. Pode ocorrer pesquisa de mercado, viabilidade, estimativas de custo/prazo. Pode incluir possíveis riscos e um cronograma do projeto.

2. Análise: elaboração de um documento de especificação de **requisitos** de software (ERS), que inclui as funções do software (requisitos funcionais), restrições quantitativas a atender (tempo de resposta, consumo energético, plataformas compatíveis), diagramas e descrição de casos de uso (UML/SysML).

3. Design (Projeto): **definir** arquitetura do software (componentes/classes e suas relações – UML/SysML) – diagramas comportamentais, estruturais, implantação. Gera Documento de Design de Software (DDS) para equipe de desenvolvimento com MODELOS

4. Desenvolvimento: com base no ERS e DDS, definição de linguagem, ambiente de execução (stand-alone, web, mobile), co-design firmware, hardware alvo, software de aplicação, interfaces (APIs)

5. Em muitos casos, como em modelos ágeis o teste contínuo (CT) é empregado como parte do modelo e o código é testado durante todo o processo de desenvolvimento (não apenas no final). Testes unitários, de integração, de usabilidade (UI), Jigas de teste (embarcados)

fazer de forma ÁGIL

Pessoas, entrega frequente, melhora contínua

Metodologia ágil é uma forma de conduzir projetos que busca dar maior rapidez aos processos e à conclusão de tarefas. Não apenas isso, mas o agile baseia-se em um fluxo de trabalho mais ágil, flexível, sem tantos obstáculos, com total iteratividade.
Tudo isso em **todo ciclo de vida de um projeto**: da sua concepção até a entrega/produto final.

fazer de forma ÁGIL

Indivíduos e interações mais que processos e ferramentas		Faça da Segurança um pré-requisito
Software em funcionamento mais que documentação abrangente	→	Entregar Valor Continuamente
Colaboração com o cliente mais que negociação de contratos	→	Tornar pessoas sensacionais
Responder a mudanças mais que seguir um plano		Experimentar e Aprender Rapidamente

Novo manifesto ágil: Joshua Kerievsk

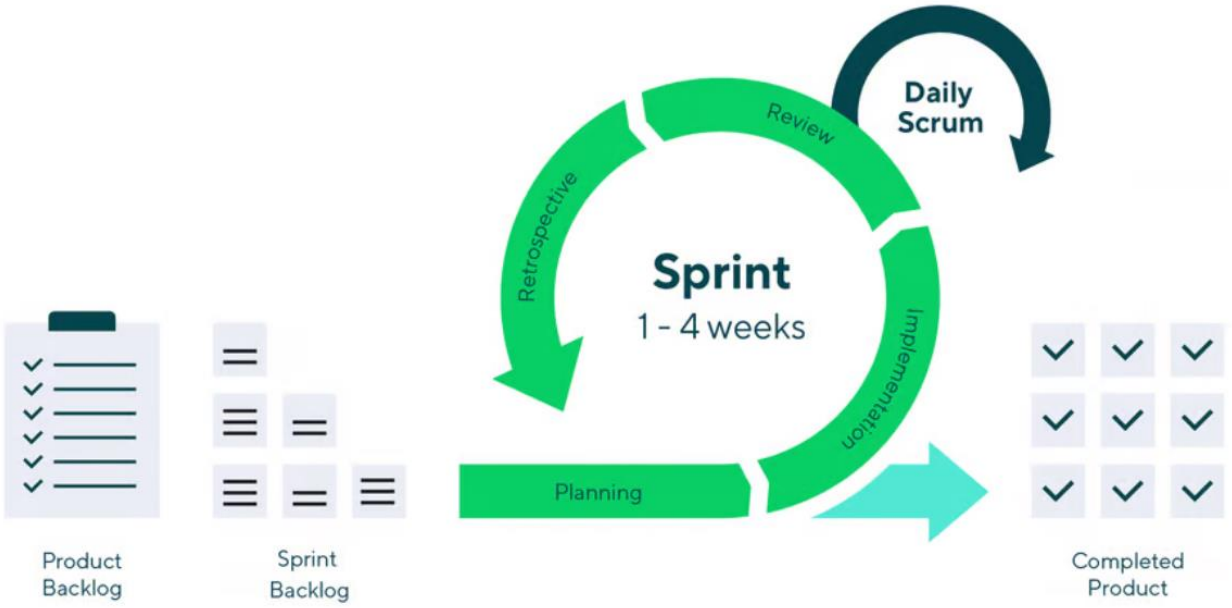
scrum

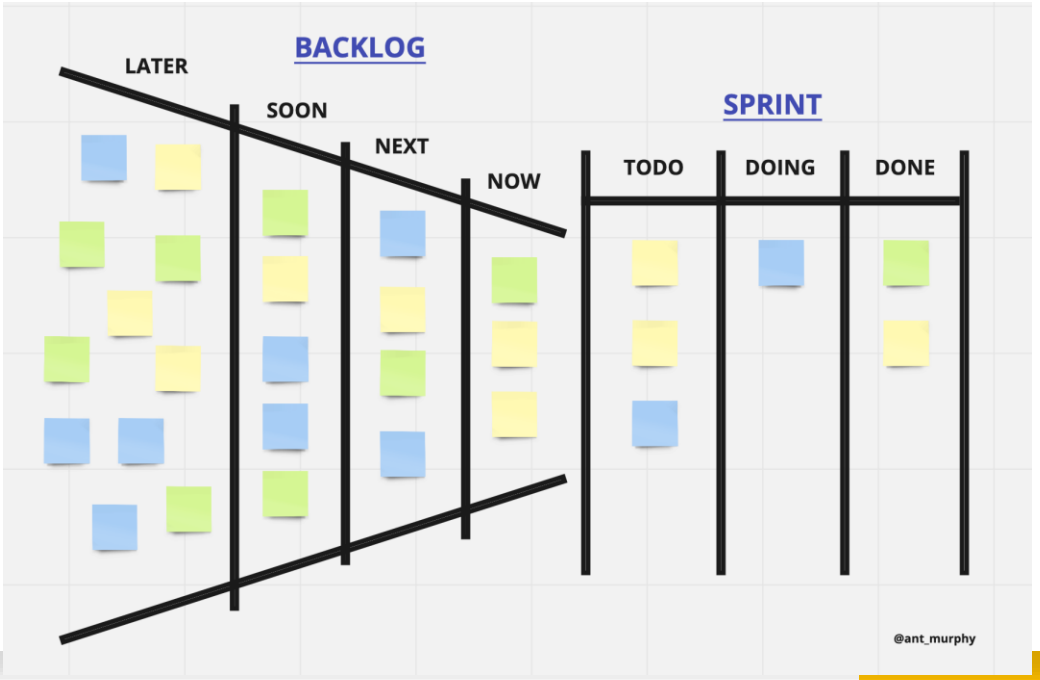
Cronograma dividido em sprints (alguns dias a 2 semanas), com entregáveis específicos.

Esses entregáveis são definidos de acordo com seu valor para o cliente. Quanto maior, mais rapidamente deve ser desenvolvido.

Em caso de atrasos, o projeto segue a dinâmica de prioridades, com as sobras sendo alocadas para sprints futuros.

No momento em que as entregas são feitas, elas são revisadas pela equipe do projeto e também pelo cliente. (versões alpha, beta etc.)





scrum

PRODUCT BACKLOG EXAMPLE						
ID	As a…	I want to be able to…	So that…	Priority	Sprint	Status
1	Administrator	see a list of all members and visitors	I can monitor site visits	Must	1	Done
2	Administrator	add new categories	I can allow members to create engaging content	Must	1	Done
3	Administrator	add new security groups	security levels are appropriate	Must	1	Done
4	Administrator	add new keywords	content is easy to group and search for	Must	1	Done
5	Administrator	delete comments	offensive content is removed	Must	1	Done
6	Administrator	block entries	competitors and offenders cannot submit content	Must	1	Done
7	Administrator	change site branding	the site is future-proofed in case brand changes	Could	1	Done
8	Member	change my password	I can keep secure	Must	1	Done
9	Member	update my contact details	I can be contacted by Administrators	Must	2	Work in Progress
10	Member	update my email preferences	I'm not bombarded with junk email	Should	2	Work in Progress
11	Member	share content to social networks	I can promote what I find interesting	Could	2	Work in Progress
12	Visitor	create an account	I can benefit from member discounts	Must		To be started
13	Visitor	login	I can post new entries	Must		To be started
14	Visitor	add comments	I can have a say	Must		To be started
15	Visitor	suggest improvements	I can contribute to the site usability	Should		To be started
16	Visitor	contact the Administrators	I can directly submit a query	Could		To be started
17	Visitor	follow a member's updates	I'm informed of updates from members I find interesting	Should		To be started
18	Visitor	view a member's profile	I can know more about a member	Must		To be started
19	Administrator	generate incoming traffic report	I can understand where traffic is coming from	Must		To be started

Apresentação – Exemplo embarcado

Baseado em padrão arquitetural em camadas

Requisitos funcionais:

1) O sistema deve acender o LED6 (azul) da placa quando o botão de usuário for pressionado, e apagá-lo quando o mesmo botão estiver desacionado;
2) O sistema deve tratar comandos seriais. Segue a lista:
   a) LEDCCN: Apaga/Acende um LED, onde CC é o comando (00 apaga e 01 acende) e N é número do LED (3, 4, 5 ou 6);
   b) ACLRDN: Leitura da aceleração de um eixo do acelerômetro, onde N é o código do eixo (X, Y ou Z). A resposta deve ser no formato ACLNVVVVV, onde N é o eixo e VVVVV é o valor da leitura;
   c) VER: Leitura da versão do firmware. A resposta deve ser no formato VERMMNNNBBB, onde MM é o campo major, NNN é o campo minor e BBB é o campo build da versão do firmware.

O projeto deve ser implementado na placa STM32F4Discovery

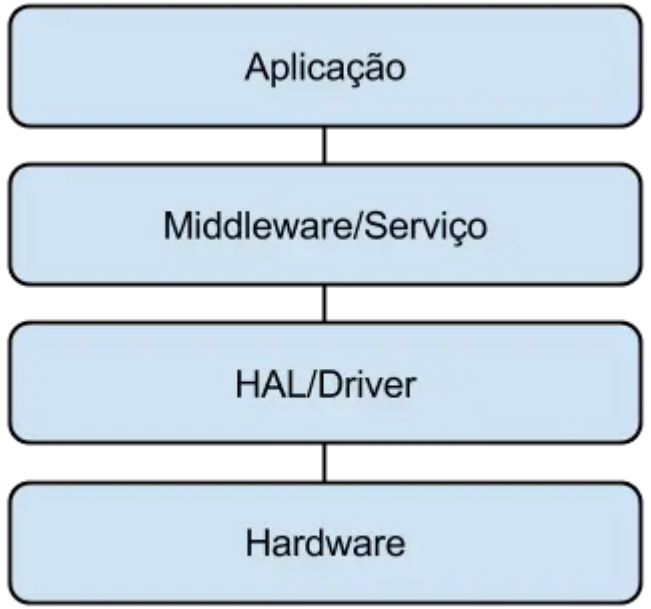


Figura 1 – Exemplo de sistema em camadas.

Fonte: ROSSI, H. **Arquitetura de software em sistemas embarcados**, 2015.
Disponível em: https://embarcados.com.br/arquitetura-de-software-em-sistemas-embarcados/

Apresentação – Exemplo embarcado

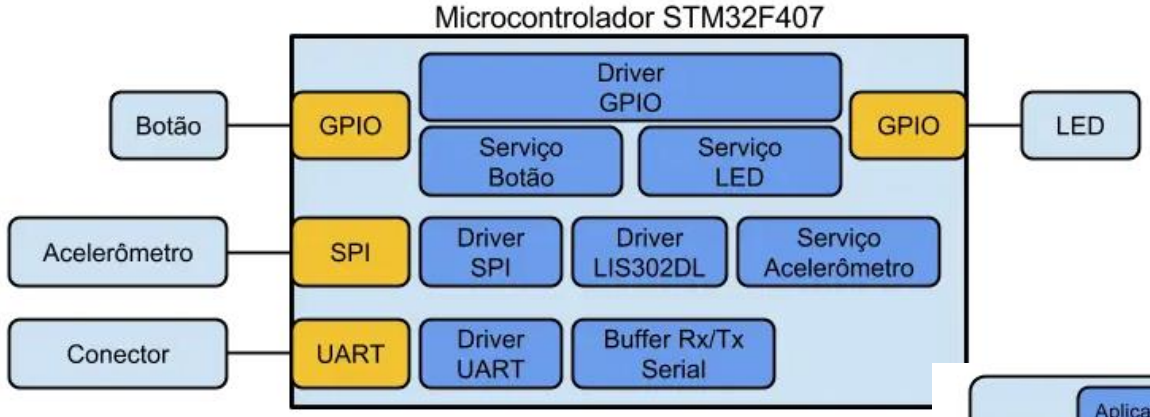


Figura 3 – Diagrama de software.

Bare metal – sem Sistema Operacional

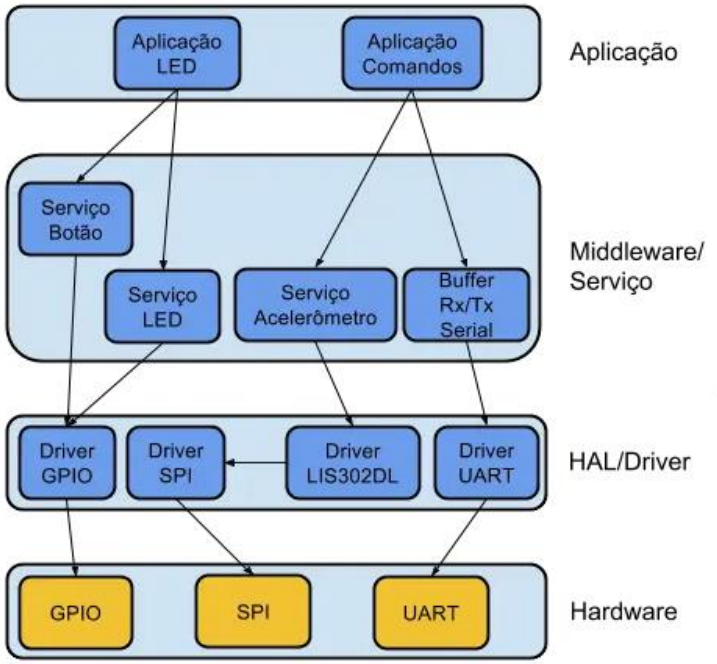


Figura 4.a – Diagrama de uso de software sem RTOS.

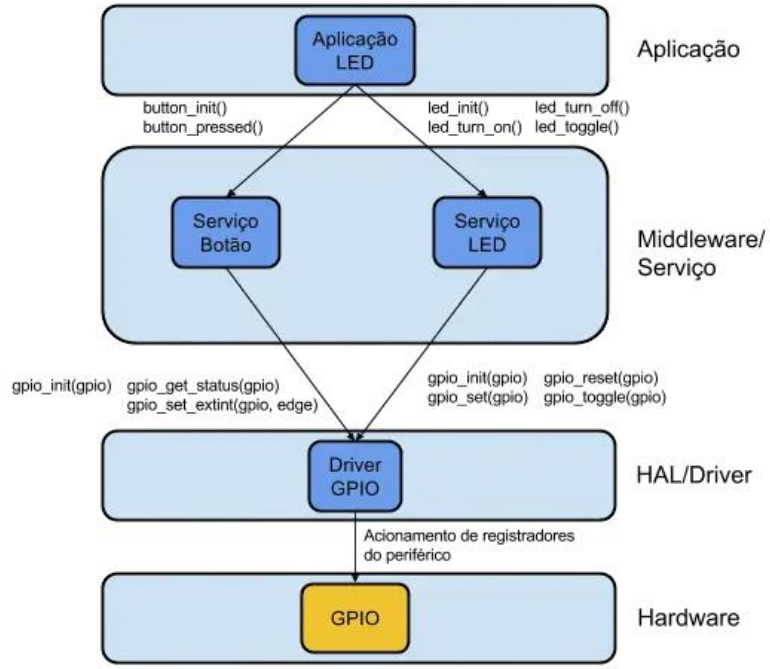


Figura 5 – Interface dos módulos responsáveis pelo tratamento do LED e Botão (sem RTOS).

Fonte: ROSSI, H. **Arquitetura de software em sistemas embarcados**, 2015.
Disponível em: https://embarcados.com.br/arquitetura-de-software-em-sistemas-embarcados/

Apresentação – Exemplo embarcado

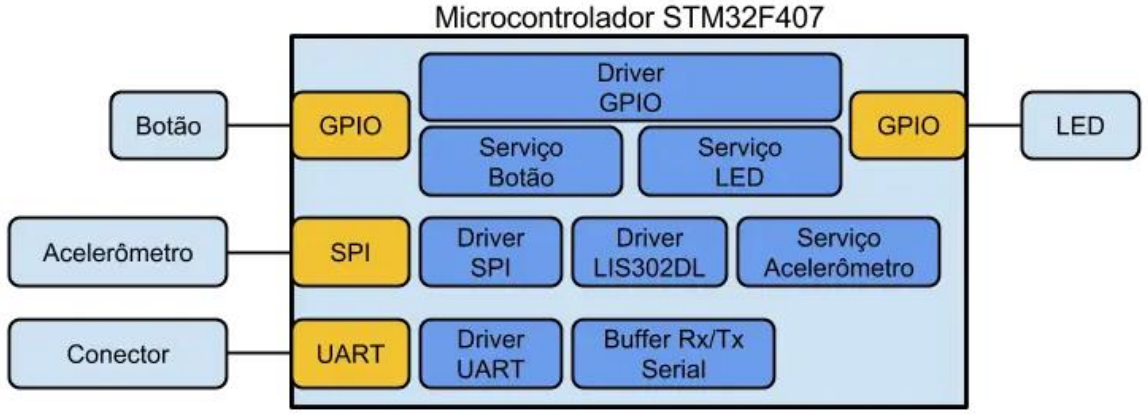


Figura 3 – Diagrama de software.

Bare metal – com Sistema Operacional

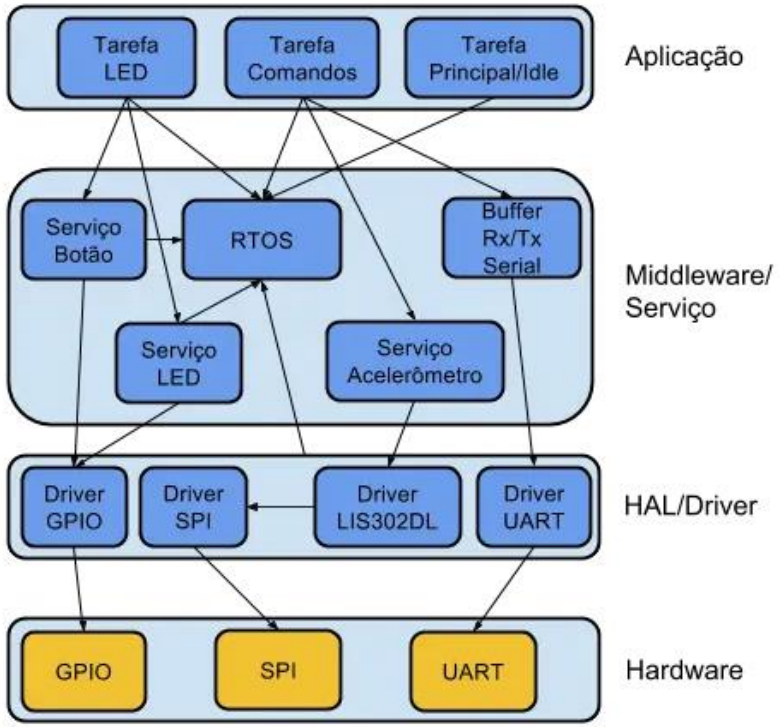


Figura 4.b – Diagrama de uso de software com RTOS.

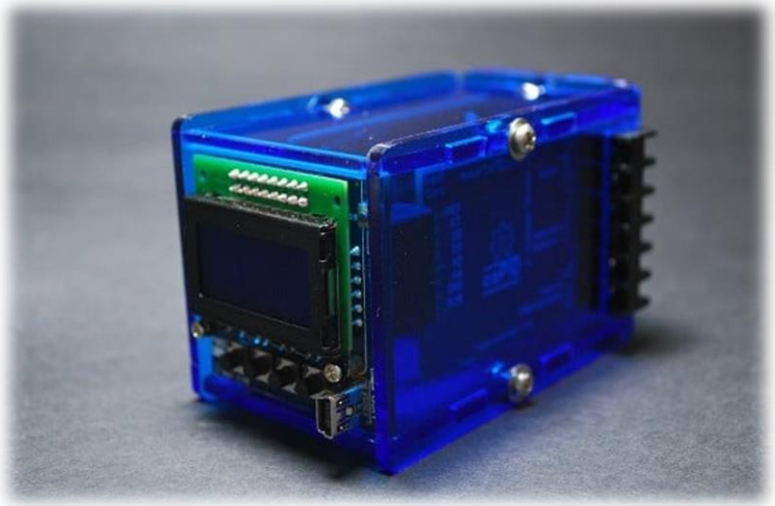
Fonte: ROSSI, H. **Arquitetura de software em sistemas embarcados**, 2015.
Disponível em: https://embarcados.com.br/arquitetura-de-software-em-sistemas-embarcados/

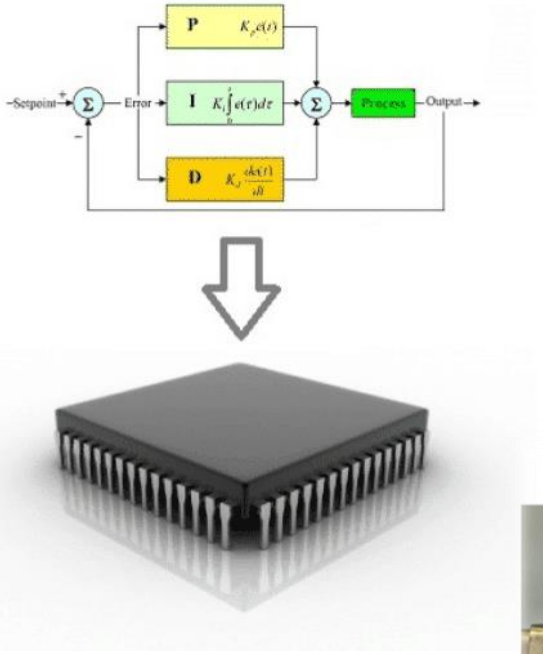
Apresentação – exemplos embarcados

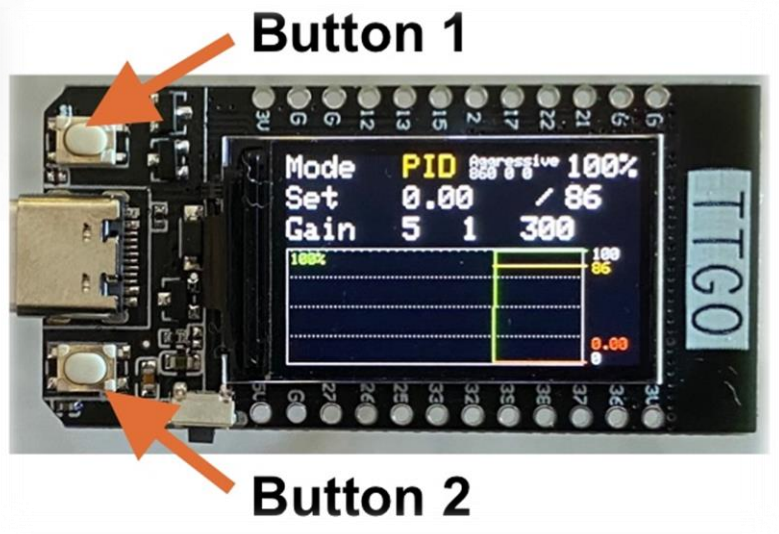
Especificação:

Tipo de item: Controlador de temperatura PID
Modelo do produto: BEM-C700-2Z-18A-3P+N-CT
Tensão nominal: 380 Vca
Potência total nominal: 7,5 kW
Corrente nominal: 18A
Faixa de temperatura: 0-400°C
Especificação do termopar: medidor de rosca M6 K tipo 2
PID do display digital: REX-C700 (O próprio PID neste caso está embarcado em produto industrial)
Características do produto: Instalação e depuração integradas, podem ser usadas diretamente (disjuntor integrado, contator, termopar, controlador de temperatura, relé de tempo digital, interruptor, indicador)

Apresentação – exemplos embarcados







Você pode criar o firmware do seu algoritmo controlador e embarcar

Apresentação – exemplos embarcados

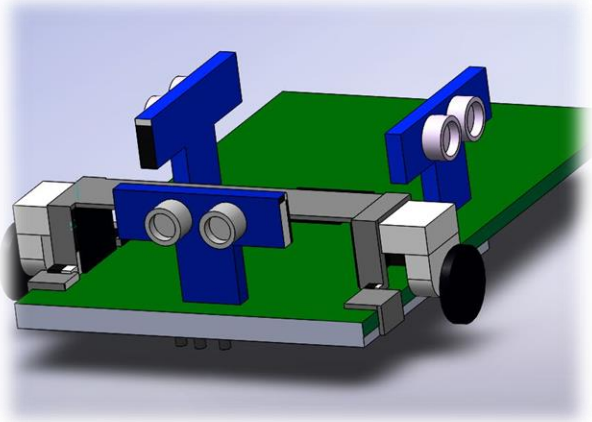
MICRO MOUSE

https://www.youtube.com/watch?v=07_MSPoP8hI

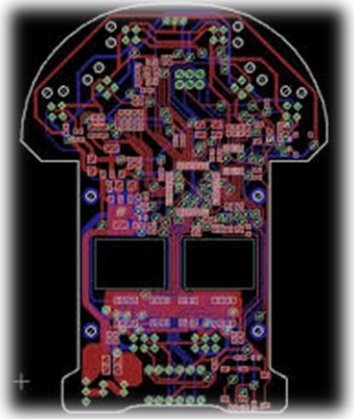
https://micromouse.utad.pt/pages/hard_overview.html

https://embarcados.com.br/micromouse/

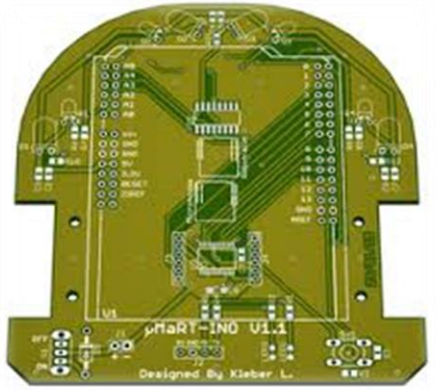
https://kleberufu.wixsite.com/micromousebrasil



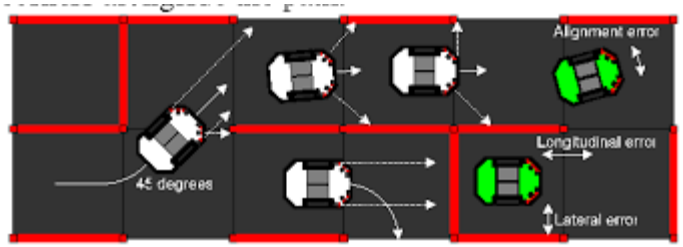
CAD



PCD design



PCB production



potiprot

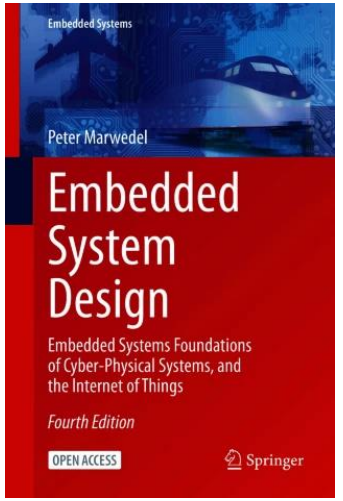
Apresentação – referências

- Referências
  - MARWEDEL, Peter. **Embedded System Design**. 4ª ed. Springer. 2021 (IoT)

    (Open Access: https://link.springer.com/content/pdf/10.1007/978-3-030-60910-8.pdf)

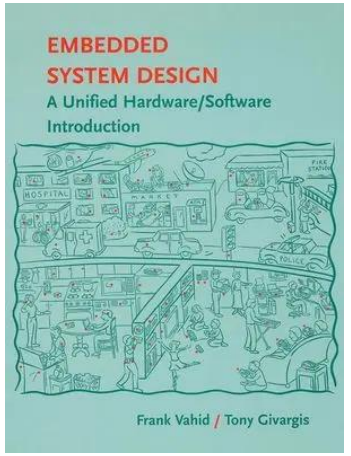
    Vídeos baseados no livro do Prof. Marwedel: https://www.youtube.com/watch?v=8HLJFlncMls

    https://www.youtube.com/@cyphysystems



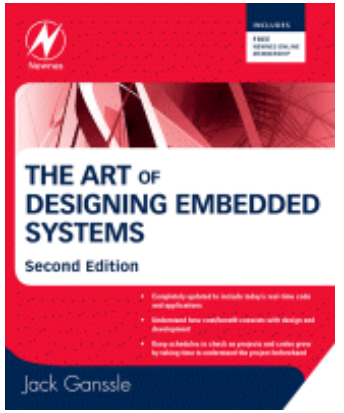
  - VAHID, Frank and GIVARGIS, Tony. **Embedded System Design**: A Unified Hardware/Software Approach. Wiley. 2002

    [url]: http://esd.cs.ucr.edu/



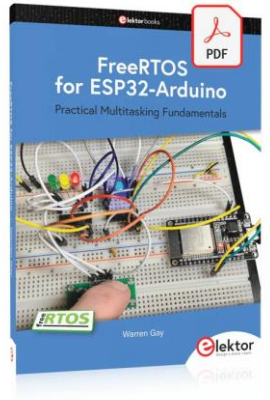
Apresentação – referências

- Referências
  - Ganssle. J. **The Art of Designing Embedded Systems**. Newnes (Elsevier),
    2a Ed. USA. 2008



  - Gay, Warren. **FreeRTOS for ESP32-Arduino**: practical multitasking fundamentals. Elektor, 2020

    https://www.elektor.com/products/freertos-for-esp32-arduino-e-book
  - Referência oficial FreeRTOS (AWS)



Real-time operating system microcontrollers and small microprocessors

FreeRTOS is a market-leading embedded system RTOS supporting 40+ processor architectures with a small memory footprint, fast execution times, and cutting-edge RTOS features and libraries including Symmetric Multiprocessing (SMP), a thread-safe TCP stack with IPv6 support, and seamless integration with cloud services. It's open-source and actively supported and maintained.

Apresentação – referências projetos

- Repo projetos
  - https://randomnerdtutorials.com/
  - https://embarcados.com.br/
  - Canal RTOS DigiKey: https://www.youtube.com/watch?v=F321087yYy4
  - TinyML (Machine Learning) no Arduino: https://www.digikey.com/en/maker/projects/intro-to-tinyml-part-1-training-a-model-for-arduino-in-tensorflow/8f1fc8c0b83d417ab521c48864d2a8ec
- Lista de livros associados: https://bookauthority.org/books/best-embedded-development-books

Apresentação – ferramentas

- Ferramentas simulação (malhas de controle): Matlab/Simulink, Scilab, Octave (.m), Python, R, Julia etc.
  - https://octave.org/ (https://octave.sourceforge.io/control/)
    - pkg install –forge control; pkg load control
- Ferramentas simulação uC (esp32 etc.): wokwi.com
  - extensão no VS CODE para simular códigos compilados
  - extensão no VS CODE Arduino (Microsoft)
- IDE Arduino
- Projeto com platformIO
- Ferramentas esquemas: fritzing, kicad, eagle, proteus, etc.

Apresentação – datas

- Datas chave propostas para *deadline* de instrumentos avaliativos
  - 18 segundas-feiras (17.03.2025 – 21.07.2025)
    - 28.04 (entrega#1), 09.06 (entrega#2), 14.07 (apresentação projeto final – docs+proto)