

Plano de Aula

DESENVOLVIMENTO DE SISTEMAS

Sistemas Embarcados

Função: Desenvolvimento de aplicações para sistemas embarcados	Atribuições e Responsabilidades: Desenvolver sistemas embarcados
Valores e Atitudes: <ul style="list-style-type: none">• Incentivar a criatividade;• Estimular a organização;• Estimular o interesse na resolução de situação-problema.	
Competências <ol style="list-style-type: none">1. Analisar modelos de sistemas embarcados2. Desenvolver aplicações com microcontroladores	Habilidades <ol style="list-style-type: none">1.1 Identificar as características de sistemas embarcados2.1 Programar sistemas para microcontroladores2.2 Executar instruções para microcontroladores

Metodologia

A disciplina de **Sistemas Embarcados** será desenvolvida com foco na integração entre teoria e prática, buscando preparar o estudante para compreender e aplicar conceitos fundamentais em projetos reais.

As estratégias metodológicas incluem:

- **Aulas expositivas dialogadas:** apresentação dos conceitos teóricos sobre arquitetura de sistemas embarcados, microcontroladores, sensores e atuadores.
- **Estudos de caso:** análise de aplicações reais em áreas como automotiva, médica, industrial e IoT.
- **Aprendizagem prática em laboratório:** desenvolvimento de projetos utilizando kits de microcontroladores (Arduino, ESP32, PIC, ARM, etc.), explorando programação em C/C++ e integração com periféricos.
- **Trabalhos em grupo:** incentivo à colaboração e à resolução de problemas complexos de forma coletiva.
- **Projetos integradores:** atividades que relacionam os conteúdos da disciplina com outras áreas do curso, como programação, eletrônica e redes.
- **Uso de recursos digitais:** simulações em softwares específicos, ambientes de prototipagem e plataformas de versionamento de código.

Tipos de Avaliação

A avaliação será contínua e diversificada, considerando tanto o desempenho individual quanto coletivo. Os principais instrumentos avaliativos são:

- **Provas teóricas:** verificação da compreensão dos conceitos fundamentais de sistemas embarcados, arquiteturas e protocolos.
- **Listas de exercícios e atividades práticas:** resolução de problemas e implementação de pequenos programas em microcontroladores.
- **Projetos práticos:** desenvolvimento de sistemas embarcados aplicados a situações reais, avaliando criatividade, funcionalidade e documentação técnica.
- **Apresentações e relatórios técnicos:** comunicação dos resultados obtidos em projetos e experimentos, com foco em clareza e organização.
- **Participação e engajamento:** acompanhamento da dedicação em sala de aula, envolvimento em discussões e colaboração em atividades de grupo.

CrITÉRIOS de Avaliação

- **Domínio conceitual:** capacidade de explicar e aplicar os fundamentos teóricos.
- **Habilidades práticas:** competência na programação e integração de hardware/software.
- **Trabalho em equipe:** cooperação, divisão de tarefas e cumprimento de prazos.
- **Inovação e criatividade:** propostas originais e soluções eficientes para problemas apresentados.
- **Documentação e apresentação:** clareza na escrita técnica e na exposição oral.

Esse modelo tem a missão de garantir que o aluno não apenas compreenda os conceitos de Sistemas Embarcados, mas possa também ser capaz de aplicá-los em projetos reais, desenvolvendo competências técnicas e profissionais essenciais para o mercado de tecnologia.

Bases Tecnológicas

Introdução desenvolvimento de software embarcado

- Hardware open-source;
- Movimento maker e tinkering;
- Internet das coisas;
- Microcontroladores de Sistemas embarcados;
- Linguagem, IDE e simuladores.

Princípios de elétrica e eletrônica

- Práticas de manuseio de componentes;
- Protoboards, LEDs e botões;
- Módulos e shields.

Programação de microcontroladores

- Estrutura de um programa (setup e loop);
- Compilação, gravação e execução;
- Variáveis e tipos de dados;
- Estruturas de decisão e repetição;
- Funções.

Entrada e saída digital

- pinMode, digitalWrite e digitalRead.
- Utilização de controle de tempo
- Timers e contadores;
- millis e micros;
- Delay e delayMicroseconds.

Entrada e saída analógica

- Conversão Analógico-Digital e Digital-Analógico.
- AnalogReference, analogRead e analogWrite.
- Funções matemáticas e trigonométricas e de texto;
- Números aleatórios.

Bibliotecas

- Sensores, sons, interrupções e comunicação.

1. Introdução desenvolvimento de software embarcado

1.1. Hardware open-source;

O conceito de hardware open-source refere-se a projetos de equipamentos eletrônicos cujo design, esquemas e documentação são disponibilizados livremente, permitindo que qualquer pessoa possa estudar, modificar, fabricar e distribuir esses dispositivos. Assim como o software livre, o hardware aberto promove colaboração, inovação e acesso democrático à tecnologia.

Início	Fim
___/___/2026	___/___/2026

O que é?

Definição: Hardware open-source (ou hardware livre) é aquele cujo projeto é publicado com licenças que permitem uso, modificação e redistribuição sem restrições proprietárias.

Origem: A ideia surgiu inspirada no movimento de software livre, mas aplicada ao mundo físico dos circuitos, placas e dispositivos.

Documentação aberta: Inclui esquemáticos, layouts de PCB, listas de materiais e até instruções de montagem, garantindo que qualquer pessoa possa reproduzir ou adaptar o projeto.

Características Principais

Transparência: Todos os detalhes técnicos são públicos.

Colaboração: Comunidades podem contribuir com melhorias e novas versões.

Acessibilidade: Reduz barreiras de entrada para estudantes, pesquisadores e empreendedores.

Licenciamento: Geralmente utiliza licenças como CERN OHL (Open Hardware License), que asseguram direitos de uso e compartilhamento

Exemplos de Hardware Open-Source

Arduino: Plataforma de prototipagem eletrônica que disponibiliza seus esquemas e código.

Raspberry Pi (parcialmente open-source): Embora o chip principal não seja aberto, grande parte da documentação e software é livre.

BeagleBone: Placa baseada em ARM, totalmente aberta em hardware e software.

RISC-V: Arquitetura de processador open-source que vem ganhando destaque mundial.

Impacto e Importância

Educação: Facilita o aprendizado prático em cursos de eletrônica e sistemas embarcados.

Indústria: Permite inovação rápida e redução de custos em prototipagem.

Comunidade: Cria ecossistemas colaborativos, semelhantes ao que ocorreu com o software livre.

Soberania tecnológica: Países e empresas podem desenvolver soluções sem depender de fornecedores proprietários.

Futuro do Hardware Open-Source

O movimento cresce com a popularização da IoT, da Indústria 4.0 e da IA embarcada. A tendência é que cada vez mais projetos sejam abertos, permitindo que desenvolvedores e empresas construam soluções sustentáveis e acessíveis.

1. Introdução desenvolvimento de software embarcado

1.2. Movimento maker e tinkering;

O movimento **Maker** e a prática do **Tinkering** representam uma nova forma de aprender e criar tecnologia, baseada na experimentação, na curiosidade e no espírito de inovação. Esses conceitos estão diretamente ligados ao desenvolvimento de sistemas embarcados, pois incentivam o uso de microcontroladores, sensores e atuadores em projetos práticos e acessíveis.

Início	Fim
___/___/2026	___/___/2026

O que é

- **Movimento Maker:** uma cultura que valoriza o “faça você mesmo” (DIY – *Do It Yourself*), estimulando pessoas a projetarem, construírem e compartilharem soluções tecnológicas.
- **Tinkering:** prática de explorar, desmontar, modificar e experimentar dispositivos ou protótipos, sem medo de errar, como forma de aprendizado ativo.

Características Principais

- **Aprendizagem prática:** foco em aprender fazendo, com prototipagem rápida.
- **Colaboração:** incentivo ao trabalho em equipe e à troca de conhecimento em comunidades.
- **Acessibilidade:** uso de ferramentas e plataformas abertas, como Arduino e Raspberry Pi.
- **Criatividade e inovação:** liberdade para criar soluções originais e personalizadas.
- **Integração multidisciplinar:** envolve programação, eletrônica, design e até impressão 3D.

Exemplos

- Criação de **robôs educativos** utilizando Arduino.
- Desenvolvimento de **estações meteorológicas caseiras** com sensores de temperatura e umidade.
- Projetos de **automação residencial** com ESP32 e módulos de comunicação Wi-Fi.
- Protótipos de **wearables** (dispositivos vestíveis) para monitoramento de saúde.
- Impressão 3D de peças personalizadas para sistemas embarcados.

Impacto e Importância

- **Na educação:** promove o aprendizado ativo, tornando os alunos protagonistas do processo.
- **Na indústria:** estimula inovação e prototipagem rápida, reduzindo custos e tempo de desenvolvimento.
- **Na sociedade:** democratiza o acesso à tecnologia, permitindo que qualquer pessoa crie soluções para problemas locais.
- **No curso de Desenvolvimento de Sistemas:** fortalece a prática dos conteúdos teóricos, preparando os alunos para desafios reais do mercado.

Futuro Relacionado

O movimento maker e o tinkering tendem a se expandir ainda mais com:

- **Integração com IoT (Internet das Coisas):** sistemas embarcados conectados em rede para aplicações inteligentes.
- **Inteligência Artificial embarcada:** dispositivos capazes de aprender e tomar decisões localmente.
- **Cidades inteligentes e sustentabilidade:** soluções criadas por makers para mobilidade, energia e meio ambiente.
- **Educação tecnológica inclusiva:** maior acesso a ferramentas de baixo custo, ampliando oportunidades de aprendizado.

Em resumo, o **Movimento Maker** e o **Tinkering** são fundamentais para a disciplina de **Sistemas Embarcados**, pois unem teoria e prática, incentivam a criatividade e preparam os alunos para o futuro da tecnologia.

1. Introdução desenvolvimento de software embarcado

1.3. Internet das coisas;

A **Internet das Coisas (IoT – Internet of Things)** é uma das áreas mais promissoras da tecnologia moderna. Ela conecta objetos físicos ao mundo digital, permitindo que dispositivos embarcados colem, transmitam e processem dados em tempo real. Na disciplina de **Sistemas Embarcados**, a IoT é essencial, pois mostra como microcontroladores, sensores e atuadores podem ser integrados em soluções inteligentes que transformam o cotidiano.

O que é

A IoT consiste em uma rede de dispositivos físicos equipados com sensores, atuadores e sistemas embarcados, capazes de se comunicar entre si e com plataformas na nuvem. Esses dispositivos podem monitorar ambientes, executar ações automáticas e fornecer informações para tomada de decisão.

Características Principais

- **Conectividade:** dispositivos interligados por redes sem fio (Wi-Fi, Bluetooth, ZigBee, LoRa, 5G).
- **Sensoriamento:** coleta de dados ambientais (temperatura, umidade, movimento, localização).
- **Automação:** execução de tarefas sem intervenção humana direta.
- **Integração com sistemas embarcados:** uso de microcontroladores e placas de desenvolvimento (ESP32, Arduino, Raspberry Pi).
- **Escalabilidade:** possibilidade de conectar milhares de dispositivos em uma mesma rede.
- **Interoperabilidade:** comunicação entre diferentes plataformas e protocolos.

Exemplos

- **Casas inteligentes:** lâmpadas, fechaduras e eletrodomésticos controlados por aplicativos.
- **Cidades inteligentes:** monitoramento de tráfego, iluminação pública e coleta de lixo automatizada.
- **Saúde:** dispositivos vestíveis que monitoram batimentos cardíacos e atividade física.
- **Indústria 4.0:** sensores em máquinas para manutenção preditiva.
- **Agronegócio:** sistemas de irrigação automatizados baseados em dados de umidade do solo.

Impacto e Importância

- **Na educação:** amplia o aprendizado prático em sistemas embarcados e redes.
- **Na indústria:** aumenta a eficiência, reduz custos e melhora a produtividade.
- **Na sociedade:** melhora a qualidade de vida com soluções inteligentes e acessíveis.
- **Na segurança:** permite monitoramento remoto e prevenção de falhas.
- **No curso de Desenvolvimento de Sistemas:** prepara os alunos para atuar em uma das áreas mais demandadas do mercado de tecnologia.

Futuro Relacionado

O futuro da IoT está diretamente ligado à evolução dos sistemas embarcados e das redes de comunicação:

- **Expansão do 5G e 6G:** maior velocidade e menor latência para dispositivos conectados.
- **Integração com Inteligência Artificial:** dispositivos capazes de aprender e tomar decisões autônomas.
- **Computação em borda (Edge Computing):** processamento de dados diretamente nos dispositivos, sem depender apenas da nuvem.
- **Sustentabilidade:** soluções IoT voltadas para economia de energia e preservação ambiental.
- **Maior segurança cibernética:** proteção contra ataques e vulnerabilidades em dispositivos conectados.

Resumidamente finalizando, a **Internet das Coisas** é um dos pilares da disciplina de **Sistemas Embarcados**, pois mostra como a tecnologia embarcada pode transformar ambientes físicos em sistemas inteligentes, conectados e eficientes, preparando os alunos para os desafios da era digital.

1. Introdução desenvolvimento de software embarcado

1.4. Microcontroladores de Sistemas embarcados;
Microcontroladores em sistemas embarcados são circuitos integrados que concentram CPU, memória e periféricos em um único chip, projetados para executar tarefas específicas de controle em dispositivos eletrônicos. Eles são a base da maioria dos sistemas embarcados modernos, permitindo que equipamentos realizem funções dedicadas de forma eficiente e com baixo consumo de energia.

Início	Fim
___/___/2026	___/___/2026

O que são?

- Definição: Diferente de um microprocessador, que depende de componentes externos, o microcontrolador já possui memória, portas de entrada/saída e periféricos integrados.
- Função: Controlar sensores, atuadores e sistemas eletrônicos em tempo real.
- Exemplo: O ATmega328P, usado no Arduino Uno, integra CPU, memória Flash, RAM e periféricos como ADC e timers.

Características

- Integração: CPU + memória + periféricos em um único chip.
- Baixo consumo: Ideal para dispositivos portáteis e IoT.
- Custo reduzido: Muito mais acessível que sistemas baseados em microprocessadores.
- Tempo real: Capacidade de responder rapidamente a eventos externos.

Aplicações em Sistemas Embarcados

- Automação residencial: Controle de iluminação, temperatura e segurança.
- Automotivo: Sistemas de injeção eletrônica, ABS, airbags.
- Dispositivos médicos: Monitores cardíacos, bombas de insulina.
- IoT: Sensores inteligentes conectados à internet.

Famílias de Microcontroladores:

- AVR (Atmel/Microchip): Base do Arduino.
- PIC (Microchip): Muito usado em aplicações industriais.
- ARM Cortex-M: Popular em sistemas embarcados modernos.
- ESP32/ESP8266: Microcontroladores com Wi-Fi e Bluetooth integrados.

Importância nos Sistemas Embarcados:

- Flexibilidade: Permitem criar soluções personalizadas para diferentes áreas.
- Eficiência: Executam apenas a tarefa para a qual foram programados.
- Escalabilidade: Podem ser usados desde projetos educacionais até aplicações industriais complexas.

Tendências Futuras

- RISC-V: Arquitetura aberta que vem ganhando espaço em microcontroladores.
- IA embarcada: Chips capazes de executar algoritmos de inteligência artificial localmente.
- Edge Computing: Processamento próximo à fonte de dados, reduzindo latência.

1. Introdução desenvolvimento de software embarcado

1.5. Linguagem, IDE e simuladores.

Linguagens, IDEs e simuladores são ferramentas fundamentais no desenvolvimento de sistemas embarcados, pois permitem programar, testar e validar o funcionamento de microcontroladores e plataformas de hardware. Elas formam o ecossistema que conecta o desenvolvedor ao dispositivo físico, garantindo eficiência e confiabilidade no processo de criação.

Início	Fim
___/___/2026	___/___/2026

O que são?

- Linguagens: São os meios de comunicação entre o programador e o hardware.
- IDEs (Ambientes de Desenvolvimento Integrados): Reúnem editor de código, compilador e ferramentas de depuração em um só lugar.
- Simuladores: Permitem testar o código e o circuito sem precisar de hardware físico, reduzindo custos e tempo de prototipagem.
- Função: Facilitar o desenvolvimento, depuração e validação de sistemas embarcados.
- Exemplo: O Arduino IDE permite escrever código em C/C++, compilar e enviar diretamente para a placa Arduino, enquanto simuladores como Proteus ou Tinkercad permitem verificar o funcionamento do circuito virtualmente.

Características Principais

Linguagens:

- C e C++: Controle direto do hardware e eficiência.
- Python: Prototipagem rápida em placas mais robustas (ex.: Raspberry Pi).
- Rust: Segurança de memória e desempenho em sistemas críticos.
- Assembly: Programação de baixo nível para otimização máxima.

IDEs:

- Integração de ferramentas em um único ambiente.
- Suporte a bibliotecas e drivers.
- Depuração em tempo real.

Simuladores:

- Testes sem necessidade de hardware físico.
- Visualização gráfica de circuitos e sinais.
- Redução de custos e tempo de desenvolvimento.

Aplicações em Sistemas Embarcados

- Educação: Ensino de programação e eletrônica com Arduino IDE e Tinkercad.
- Indústria: Desenvolvimento de sistemas críticos com STM32CubeIDE e simuladores avançados.
- IoT: Prototipagem rápida de dispositivos conectados usando Python e simuladores online.
- Pesquisa: Testes de arquiteturas novas com QEMU e linguagens modernas como Rust.

Exemplos de Ferramentas

- Linguagens: C, C++, Python, Rust, Assembly.
- IDEs: Arduino IDE, PlatformIO, MPLAB X, STM32CubeIDE.
- Simuladores: Proteus, Tinkercad Circuits, SimulIDE, QEMU.

Importância nos Sistemas Embarcados

- Produtividade: Acelera o ciclo de desenvolvimento.
- Segurança: Permite depuração e testes antes da implementação física.
- Acessibilidade: Facilita o aprendizado e democratiza o acesso à tecnologia.
- Escalabilidade: Suporta desde projetos educacionais até aplicações industriais complexas.

Tendências Futuras

- Cloud IDEs: Ambientes de desenvolvimento baseados em nuvem, acessíveis de qualquer lugar.
- IA embarcada: Simuladores integrando algoritmos de inteligência artificial.

Sistemas Embarcados

- Integração com Edge Computing: Ferramentas que simulam processamento distribuído próximo à fonte de dados.

Resumidamente podemos dizer que linguagens, IDEs e simuladores são o tripé essencial do desenvolvimento embarcado, permitindo que ideias se transformem em protótipos funcionais e soluções reais de forma ágil e acessível.

2. Princípios de elétrica e eletrônica

Os **princípios de elétrica e eletrônica** são fundamentais para o estudo e aplicação de sistemas embarcados. Eles fornecem a base para compreender como os componentes eletrônicos funcionam, como são interligados e como podem ser utilizados em protótipos e projetos reais. Nesta etapa da disciplina, o aluno aprende a manusear componentes básicos, utilizar protoboards e explorar módulos e *shields*, desenvolvendo habilidades práticas essenciais para a integração entre hardware e software, com foco em:

- Práticas de manuseio de componentes;
- Protoboards, LEDs e botões;
- Módulos e shields.

O que é

Este tópico aborda os conceitos básicos de eletricidade e eletrônica aplicados ao desenvolvimento de sistemas embarcados. Inclui:

- **Manuseio de componentes eletrônicos** (resistores, capacitores, transistores, LEDs, botões).
- **Uso de protoboards** para montagem de circuitos sem necessidade de solda.
- **Aplicação de módulos e shields** que expandem as funcionalidades de microcontroladores (Wi-Fi, Bluetooth, sensores, displays).

Características Principais

- **Segurança no manuseio:** cuidados ao lidar com energia elétrica e componentes.
- **Modularidade:** uso de protoboards e shields que permitem rápida prototipagem.
- **Interatividade:** LEDs e botões como elementos básicos de entrada e saída.
- **Flexibilidade:** possibilidade de testar diferentes configurações sem montagem definitiva.
- **Integração com microcontroladores:** conexão direta com placas como Arduino e ESP32.

Exemplos

- Montagem de um circuito simples com **LED e resistor** para aprender sobre corrente e tensão.
- Uso de **botões** para controlar funções em um microcontrolador.
- Protótipos com **protoboard** para testar sensores de temperatura ou movimento.
- Aplicação de **shields de comunicação Wi-Fi** em projetos de IoT.
- Integração de **módulos de display LCD** para mostrar informações coletadas por sensores.

Impacto e Importância

- **Na educação:** permite que o aluno compreenda na prática conceitos abstratos de eletrônica.
- **Na prototipagem:** facilita a criação rápida de projetos sem necessidade de soldagem.
- **Na indústria:** desenvolve habilidades essenciais para manutenção e inovação tecnológica.
- **Na disciplina de Sistemas Embarcados:** conecta teoria de programação com prática de hardware, formando profissionais completos.

Futuro Relacionado

O futuro dos princípios de elétrica e eletrônica aplicados a sistemas embarcados aponta para:

- **Protoboards inteligentes:** com sensores integrados e monitoramento digital.
- **Componentes miniaturizados:** cada vez menores e mais eficientes.
- **Módulos avançados:** integração com inteligência artificial e conectividade 5G.
- **Educação prática digitalizada:** simulações virtuais complementando o uso físico de componentes.
- **Sustentabilidade:** uso de componentes de baixo consumo energético e recicláveis.

Em resumo, o estudo dos **Princípios de Elétrica e Eletrônica** é essencial para que o aluno domine a base prática dos sistemas embarcados, aprendendo a manipular componentes, montar circuitos e integrar módulos que expandem as possibilidades de seus projetos.

3. Programação de microcontroladores

A **programação de microcontroladores** é o coração dos sistemas embarcados. Ela permite que dispositivos eletrônicos executem tarefas específicas, controlando sensores, atuadores e módulos de comunicação. Ao aprender a programar microcontroladores, o aluno desenvolve a capacidade de transformar circuitos eletrônicos em sistemas inteligentes e funcionais, aplicando lógica de programação diretamente em hardware, onde temos:

- Estrutura de um programa (setup e loop);
- Compilação, gravação e execução;
- Variáveis e tipos de dados;
- Estruturas de decisão e repetição;
- Funções.

O que é

Programar microcontroladores significa escrever instruções que serão gravadas em sua memória e executadas para controlar entradas e saídas. Essa programação geralmente é feita em linguagens como **C/C++**, utilizando ambientes como **Arduino IDE** ou **PlatformIO**, e envolve etapas como:

- Estruturação do programa em **setup()** (configuração inicial) e **loop()** (execução contínua).
- **Compilação** do código para transformar instruções em linguagem de máquina.
- **Gravação** do programa na memória do microcontrolador.
- **Execução** automática do código pelo dispositivo.

Características Principais

- **Estrutura básica:** funções setup() e loop() organizam o fluxo do programa.
- **Variáveis e tipos de dados:** permitem armazenar e manipular informações.
- **Estruturas de decisão:** comandos como if/else controlam o fluxo lógico.
- **Estruturas de repetição:** laços como for e while permitem execução contínua ou repetida.
- **Funções:** organizam o código em blocos reutilizáveis, facilitando manutenção e clareza.
- **Interação com hardware:** leitura de sensores e acionamento de atuadores.

Exemplos

- **Piscar um LED** em intervalos definidos (programa clássico de introdução).
- **Leitura de um sensor de temperatura** e exibição dos dados em um display LCD.
- **Controle de um motor** utilizando PWM (modulação por largura de pulso).
- **Sistema de alarme** que aciona um buzzer quando um botão é pressionado.
- **Automação residencial** com acionamento de lâmpadas via microcontrolador conectado à rede Wi-Fi.

Impacto e Importância

- **Na educação:** desenvolve raciocínio lógico e habilidades práticas de programação.
- **Na indústria:** permite criar soluções embarcadas para automação, robótica e IoT.
- **Na sociedade:** viabiliza dispositivos inteligentes que facilitam o cotidiano.
- **No curso de Desenvolvimento de Sistemas:** conecta teoria de programação com prática de hardware, formando profissionais preparados para o mercado tecnológico.

4. Entrada e saída digital

- pinMode, digitalWrite e digitalRead.

Sistemas Embarcados

5. Utilização de controle de tempo

- Timers e contadores;
- Millis e micros;
- Delay e delayMicroseconds.

Sistemas Embarcados

6. Entrada e saída analógica

- Conversão Analógico-Digital e Digital-Analógico.
- AnalogReference, analogRead e analogWrite.
- Funções matemáticas e trigonométricas e de texto;
- Números aleatórios.

7. Bibliotecas

- Sensores, sons, interrupções e comunicação.

Bibliografia

BOLZANI, Carlos A. **Sistemas embarcados: hardware e software na prática**. São Paulo: Erica, 2014.

COSTA, André Luiz. **Sistemas embarcados e Internet das Coisas: fundamentos e aplicações**. São Paulo: Atlas, 2020.

Macedo, Glener Diniz. **Plano de Aula para disciplina de Sistemas Embarcados do curso de Desenvolvimento de Sistemas**. Osvaldo Cruz: Link:< <https://github.com/gdmacedo/Docencia/blob/main/PlanoDeAulas/5051-SISTEMAS%20EMBARCADOS.pdf> >Disponível em: 24 nov 2025.

PEREIRA, Marcelo Leite. **Sistemas embarcados: projeto de aplicações com Arduino e ARM**. Rio de Janeiro: LTC, 2016.

SILVA, Edson Luiz da; OLIVEIRA, José Carlos de. **Microcontroladores PIC: teoria e prática de sistemas embarcados**. 2. ed. São Paulo: Novatec, 2012.

SOUZA, Sérgio Tenório de. **Projeto de sistemas embarcados com FPGA**. São Paulo: Blucher, 2018.