

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 1. Analisar modelos de sistemas embarcados 2. Desenvolver aplicações com microcontroladores	Habilidades 1.1 Identificar as características de sistemas embarcados 2.1 Programar sistemas para microcontroladores 2.2 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.

Sistemas Embarcados

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

Transparéncia: 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.

Sistemas Embarcados

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.

Sistemas Embarcados

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.

Sistemas Embarcados

Bibliografia

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

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

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

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

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