

Centro Paula Souza
Etec Vasco Antonio Venchiarutti - Jundiaí-SP

Pesquisa - Arduino

Felipe Teixeira

Gabriel Medeiros

Paulo Roberto

Samuel Henrique

Vinicius Bonturi

1. Introdução

O Arduino é uma plataforma de prototipagem eletrônica de código aberto, composta por hardware e software fáceis de usar.

Ela foi criada para permitir que artistas, designers, estudantes e entusiastas construam projetos interativos de forma acessível e intuitiva. A principal característica do Arduino é sua capacidade de conectar componentes eletrônicos físicos (como sensores e atuadores) a programas que controlam seu comportamento por meio de código.

2. Estrutura do Hardware

O hardware do Arduino consiste em uma placa de circuito impresso com um microcontrolador — o 'cérebro' do sistema — responsável por executar instruções contidas no código carregado. Existem diversos modelos de placas, sendo os mais comuns o Arduino Uno, Mega e Nano.

Principais componentes do hardware:

- **Microcontrolador:** geralmente da família ATmega (como o ATmega328P).
- **Pinos digitais:** usados para entrada e saída de sinais (0V ou 5V).
- **Pinos analógicos:** permitem leitura de valores variáveis, como sensores de luz ou temperatura.
- **Pinos de alimentação:** fornecem energia (5V, 3.3V e GND).
- **Porta USB:** usada para conectar o Arduino ao computador e enviar programas.

- **Regulador de tensão:** estabiliza a energia elétrica que alimenta a placa.

3. Software e Linguagem de Programação

O ambiente de desenvolvimento integrado (IDE) do Arduino é utilizado para escrever, compilar e enviar códigos para a placa. A linguagem de programação é baseada em C/C++, com algumas simplificações que tornam o aprendizado mais acessível.

Principais funções da linguagem Arduino:

- **setup()**: executada uma vez ao iniciar o programa; usada para configurações iniciais.
- **loop()**: executada continuamente após o setup(); usada para o funcionamento repetitivo do sistema.
- **digitalWrite(pino, valor)**: envia um sinal (HIGH ou LOW) a um pino.
- **digitalRead(pino)**: lê o estado de um pino digital.
- **analogRead(pino)**: lê valores de sensores analógicos (0–1023).
- **delay(ms)**: pausa o programa por um tempo em milissegundos.

4. Aplicações do Arduino

O Arduino é amplamente utilizado em projetos educacionais, industriais e de pesquisa. Por ser versátil, ele permite o desenvolvimento de sistemas de automação, robótica e Internet das Coisas (IoT).

Exemplos de aplicações práticas:

- Controle de iluminação automática.
- Medidores de temperatura e umidade.
- Estações meteorológicas caseiras.
- Sistemas de irrigação automática.
- Robôs controlados por sensores.
- Impressoras 3D e drones.

5. Vantagens e Limitações

Vantagens	Limitações
<ul style="list-style-type: none">- Facilidade de uso e programação.	<ul style="list-style-type: none">- Baixo poder de processamento comparado a microcontroladores avançados.
<ul style="list-style-type: none">- Grande comunidade de suporte.	<ul style="list-style-type: none">- Memória limitada para projetos muito grandes.
<ul style="list-style-type: none">- Compatibilidade com diversos sensores e módulos.	<ul style="list-style-type: none">- Não indicado para aplicações críticas ou industriais de alta complexidade.
<ul style="list-style-type: none">- Custo acessível.	

6. Conclusão

O Arduino revolucionou a forma como a eletrônica é ensinada e aplicada, tornando o desenvolvimento de sistemas interativos acessível a todos. Com uma comunidade global ativa e inúmeros recursos disponíveis, o Arduino é uma ferramenta essencial para quem deseja aprender sobre eletrônica, programação e automação de forma prática e criativa.

7. Referências

<https://victorvision.com.br/blog/o-que-e-arduino/>

<https://arqia.com.br/post/arduino/>

<https://www.redbull.com/br-pt/arduino-e-suas-multiplas-aplicacoes>