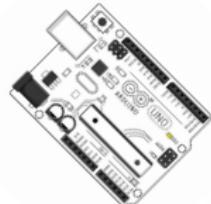


Arduino: Conceitos, Programação e Aplicação Prática



CAMPUS PATO BRANCO

UniversidadequeTransforma

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS PATO BRANCO

AULA
1

Introdução ao Arduino, TinkerCad e IDE

Coordenador
Fábio Favarim

Instrutor
Gabriel S. Folly

CAMPUS PATO BRANCO

UniversidadequeTransforma



Sumário

- Introdução ao Arduino
- Arduino Uno
- Arduino Nano
- Arduino Mega
- Shields para Arduino
- Conectores e Protoboard
- LED
- Conhecendo a IDE
- Laboratório 01

Microcontroladores e sua Importância

- Desempenham papel fundamental na vida cotidiana.
- Cérebro de sistemas eletrônicos: eletrodomésticos, veículos, dispositivos médicos.
- Versatilidade e automação.
- Controle preciso, eficiência e confiabilidade.
- Economia de energia com modos programáveis.
- Papel central na Internet das Coisas (IoT).



Arduino e suas Características

- Plataforma de desenvolvimento de código aberto.
- Criado na Itália em 2003, inspirado em Wiring e Processing.
- Desenvolvido para ser acessível e educacional.
- Impacto significativo na educação em eletrônica e programação.
- Ambiente de desenvolvimento amigável.
- Globalmente reconhecido e utilizado em projetos comerciais e industriais.



CAMPUS PATO BRANCO

UniversidadequeTransforma

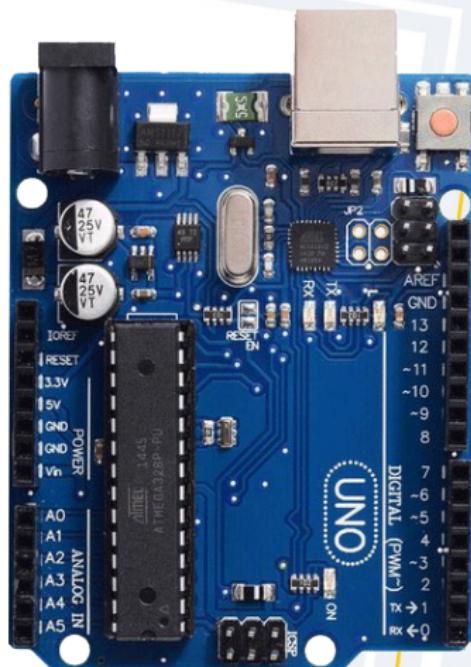
UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS PATO BRANCO

Arduino e suas Características

- Código Aberto: Acesso, modificação e compartilhamento livre.
- Plataforma Acessível: Componentes amplamente disponíveis e microcontroladores baratos.
- Facilidade de Uso: Ambiente amigável e linguagem C/C++ simplificada.
- Ampla Comunidade: Global, ativa e colaborativa.
- Diversos Modelos: Adaptáveis a várias aplicações.
- Versatilidade: Desde robótica até wearables.
- Integração de Hardware: Conexão simples de sensores, motores e displays.
- Suporte a Shields: Placas de expansão para funcionalidades específicas.
- Comunicação: USB, Bluetooth, Wi-Fi, ideal para IoT.
- Ampla Documentação: Tutoriais e recursos online disponíveis.

Componentes Principais da Placa Arduino

- Microcontrolador (ATmega328): Cérebro da placa, executa o código.
- Cristal Oscilador: Fornece referência de tempo (16MHz).
- Portas Digitais (0 a 13): Entradas/saídas digitais para controle.
- Portas Analógicas (A0 a A5): Leitura de sinais analógicos.
- Diversos Modelos: Adaptáveis a várias aplicações.
- Pinos de Alimentação: 5V, 3.3V, GND.
- Botão de Reset: Reinicia a execução do código.
- LEDs (PWR, TX, RX, 13, L): Indicadores visuais e de estado.

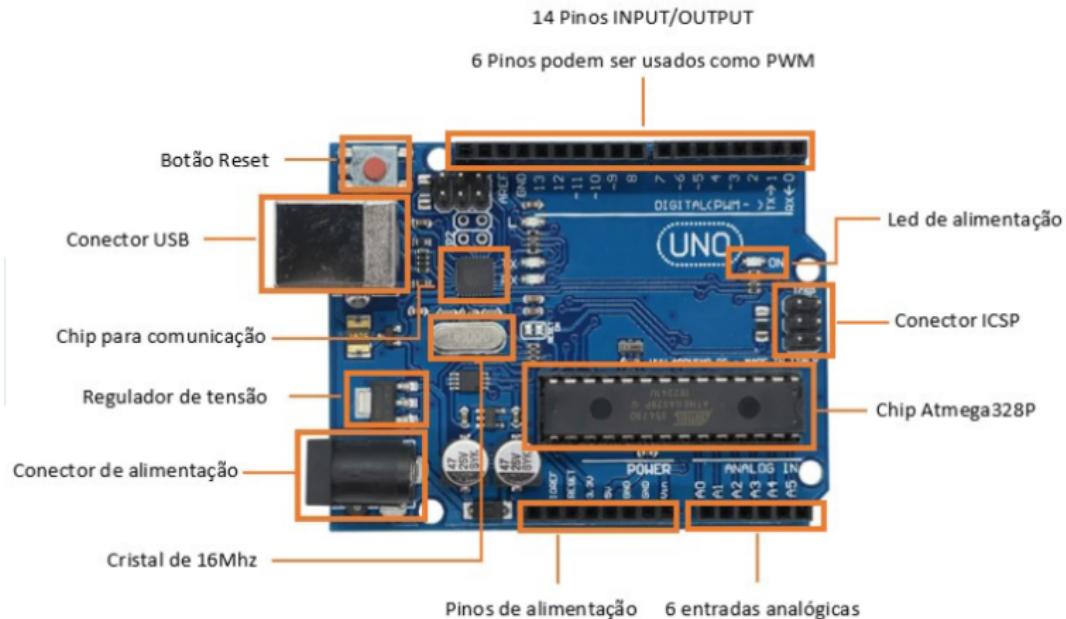


Componentes Principais da Placa Arduino

- Conector USB: Alimentação e transferência de dados.
- Conector ICSP: Programação serial externa.
- Conector de Alimentação Externa: Alimentação via fonte externa.
- Regulador de Tensão (7805): Fornece 5V à placa.
- Cabeçalhos de Conexão: Facilitam conexão de sensores e dispositivos.
- Circuito Integrado USB-to-Serial (ATmega16U2): Facilita comunicação USB com o computador.

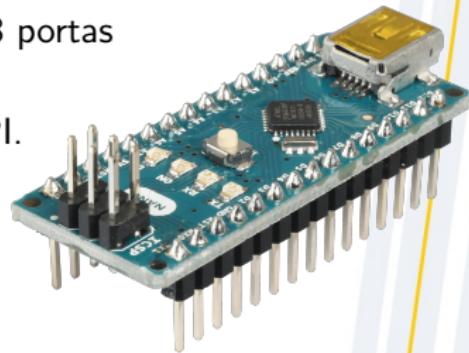


Arduino Uno



Arduino Nano

- Microcontrolador: ATmega328P.
- Dimensões: Compacto e leve, ideal para projetos com espaço limitado.
- Portas de E/S Digitais: 14 portas digitais e 8 portas analógicas.
- Interfaces de Comunicação: UART, I2C e SPI.
- Clock: Frequência de operação de 16 MHz.
- Memória Flash: 32 KB para armazenamento de programas.
- Alimentação: Via USB ou fonte externa de 7-12V.
- Compatibilidade: Suporta shields e placas de expansão, sendo eficiente para projetos compactos.



Arduino Mega

- Microcontrolador: ATmega2560.
- Portas de E/S Digitais: 54, sendo 15 com saídas PWM.
- Portas Analógicas: 16 disponíveis.
- Interfaces de Comunicação: UART, I2C e SPI.
- Clock: Frequência de operação de 16 MHz.
- Memória Flash: 256 KB para armazenamento de programas.
- Alimentação: Via USB ou fonte externa de 7-12V.
- Compatibilidade: Suporta shields, ideal para projetos complexos com muitas portas de I/O.



Características dos Shields

São placas de expansão para pinos de E/S do Arduino. Que adicionam funcionalidades específicas e facilitam a incorporação de hardware. Incluem sensores, módulos de comunicação, displays, relés, etc.

- Conexão Fácil: Encaixe direto nos pinos do Arduino, simplificando a conexão de hardware.
- Empilháveis: Permitem o uso de vários shields em um único projeto.
- Compatibilidade: Desenvolvidos para diferentes modelos de placas Arduino.
- Diversidade de Funções: Comunicação, sensores, controle de motores, exibição, etc.
- Facilidade de Prototipagem: Agiliza a prototipagem para testar e implementar ideias eficientemente.
- Documentação e Suporte: Acompanhados de documentação e bibliotecas de software para desenvolvimento.

Exemplos Comuns de Shields

- Ethernet Shield: Adiciona funcionalidades de conexão à rede.
- Motor Shield: Facilita o controle de motores.
- GPS Shield: Integra módulos de posicionamento global.
- LCD Shield: Adiciona capacidades de exibição ao projeto.



Jumpers de Protoboard

- Definição: Fios condutores temporários para estabelecer conexões em protoboards.
- Utilidade: Conectam componentes eletrônicos sem solda, permitindo ajustes.
- Flexibilidade: Facilitam reorganização rápida de conexões no circuito.
- Cores e Tamanhos: Disponíveis em várias cores e tamanhos para identificação.
- Conexão com Protoboard: Encaixam nos orifícios, proporcionando conexão temporária.
- Facilidade de Uso: Ideal para ambientes educacionais e desenvolvimento rápido de projetos.



Protoboard - Definição e Estrutura

- Definição: Placa de ensaio para desenvolvimento e teste de circuitos eletrônicos.
- Estrutura e Layout: Matriz de furos organizados em linhas e colunas, com condutores internos.
- Finalidade e Aplicações: Amplamente usada em laboratórios, cursos de eletrônica e prototipagem rápida.
- Conexões Temporárias: Componentes eletrônicos conectados por jumpers para testes.
- Organização e Boas Práticas: Crucial para clareza do circuito e facilitar teste e análise.

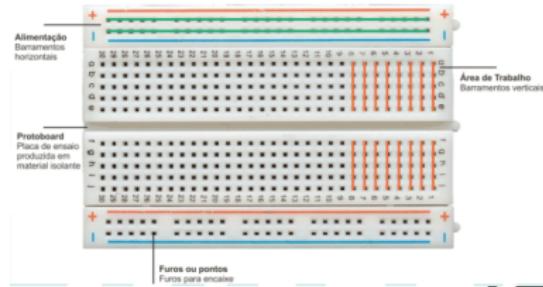
Protoboard - Importância e Versatilidade

- Facilidade de Uso: Ferramenta fundamental no aprendizado e experimentação em eletrônica.
- Importância na Prototipagem: Essencial para validar ideias antes da produção definitiva.
- Versatilidade: Usada em projetos simples educacionais e prototipagem complexa profissional.



Conexões na Protoboard

- Furos Horizontais: Conectam na horizontal, compartilhando alimentação e referência (GND).
- Área de Trabalho: Furos conectados na vertical, permitindo conexão de componentes como LEDs.



Identificação e Conexão de LEDs

- Identificação dos Pólos:
 - Ânodo (Polo Positivo): Mais longo, saliência no invólucro, associado ao lado plano ou chanfro.
 - Cátodo (Polo Negativo): Mais curto, associado ao lado com chanfro ou marcação plana.
- Conexão Correta:
 - Fonte de Alimentação: Apropriada para a tensão e corrente do LED.
 - Resistor (se necessário): Limitar a corrente, calculado conforme a lei de Ohm e especificações do LED.

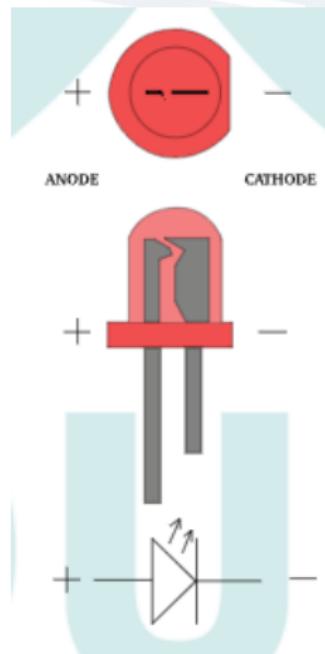


Identificação e Conexão de LEDs

- Procedimento de Conexão:
 - Ânodo para o Positivo: Conectar ao lado positivo da fonte, inserir resistor se necessário.
 - Cátodo para o Negativo: Conectar ao lado negativo da fonte.
 - Respeitar a Polaridade: Assegurar a correta conexão ânodo-cátodo.
- Teste de Funcionamento:
 - Verificação Visual: Confirmar visualmente ânodo e cátodo.
 - Ativação da Fonte: Garantir que o LED acenda; revisar polaridade e conexões.
- Dica Adicional:
 - Multímetro: Utilizar para confirmar a identificação dos pólos do LED antes da conexão.

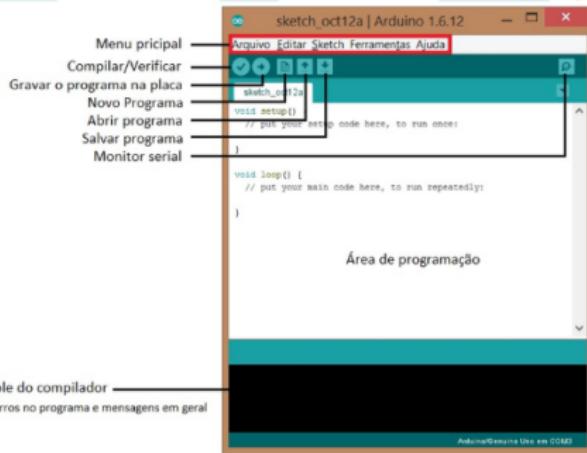
Demonstração da Polaridade do LED

- Identificação: Ânodo (positivo) e cátodo (negativo).
- Círculo Diretamente Polarizado: Corrente flui, LED acende.
- Círculo Reversamente Polarizado: Círcuito aberto, LED não conduz corrente.
- Observação: A correta polaridade é essencial para o funcionamento adequado do LED.



Conhecendo a IDE

- Ambiente amigável baseado em C/C++ para programar placas Arduino.
- Inclui ferramentas de compilação, upload de firmware e monitor serial.
- Amplamente usado em eletrônica, automação e robótica.
- Comunidade oferece suporte extenso e vasta biblioteca de código, tornando a programação de placas Arduino mais acessível e versátil.



LED Piscante

Neste guia, você aprenderá a montar um circuito simples usando um Arduino para fazer um LED piscar a cada 2 segundos. Materiais Necessários:

- Arduino Uno.
- 1 LED.
- 1 Resistor de 220 ohms.
- Cabos jumper (fios de conexão).
- Protoboard.

Solução Laboratório 01

```
1 int LED_VERMELHO = 12;
2
3 void setup() {
4     pinMode(LED_VERMELHO, OUTPUT); //Definindo o LED que esta na porta 12 como OUTPUT (Saída
5 }
6
7 void loop() {
8     digitalWrite(LED_VERMELHO, HIGH); //Liga
9     delay(2000); //2 segundos
10    digitalWrite(LED_VERMELHO, LOW); //Desliga
11    delay(2000); //2 segundos
12 }
13
```

